Rendering Operators

Formal solution to the rendering equation

$$R = \iint_{\Lambda\, T} \iint_{d\omega \bullet dA} \left\{ \sum_{k=0}^{\infty} \int_{M_2} \int_{M_2} \ldots \int_{M_2} K(x_0, x_1, \ldots, x_k) L_e(x_0, x_1)\, dS_0\, dS_1 \ldots dS_k \right\} d\vec{A} \bullet d\vec{\omega}\, dt\, d\lambda$$

$$f_r(x_k, \omega_k, \omega_k')G(x_k, x_{k-1}) \ldots f_r(x_1, \omega_1, \omega_1')G(x_1, x_0)$$

Lots of darn integrals:

- **Spectral response**
- **Motion blur**
- **Antialiasing and depth of field**
- **Multiple paths between lights and camera**
  - **Reflection and transmission**
  - **Transport between surfaces**

Robust (and efficient) integration technique: Monte Carlo

Monte Carlo

Basic idea: Use a stochastic process to compute a ensemble average (integration)

- **Direct simulation of a stochastic physical process**
  Brownian motion, quantum mechanics, queing
- **Stochastic simulation of a deterministic process**
  Definite integrals, solution of linear equations

History (Los Alamos)

- **Fermi: Neutron diffusion and moderation**
- **Ulam (coined the term) and Metropolis**
- **Albert and Spanier**
  Applied to integral equations
- **von Neumann, Kahn, Ulam**
  Variance reduction; 1950's

Advantages

Random sampling

- **Convert aliases to noise**
- **Statistical view**

Quasi-random sampling

- **Discrepancy: Asymptotically fast in high dim.**

- ■ **Easy to think about (but deceptive!)**
- ■ **Easy to program**
- ■ **Handle complicated systems**
- ■ **Robust; point sampling process**
- ■ **Solution at selected points**

Monte Carlo Integration

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

Rejection

```
score=0
for(i=0;i<n;i++)
  if( f(rand()) < rand() )
    score++
return score/n;
```

Random sample

```
score=0
for(i=0;i<n;i++)
  score += f(rand());
return score/n;
```

Discrete Probability Distributions

Discrete events e_i with probabilities p_i

Probabilities

1. $p_i \geq 0$
2. $\sum_i p_i = 1$

Construction of samples

To randomly select an event,

Select e_i if

$$p_0 + \cdots + p_{i-1} < r \leq p_0 + \cdots + p_{i-1} + p_i$$

Continuous Probability Distributions

1. **P(x) is the Prob{r<x}**
2. **P(b)-P(a) is the Prob(a<r<=b}**
3. **P(x) monotonic, 0<=P(x)<=1**
4. **p(x)=P′(x) is the probability distribution function**

Construction of samples

Solve for $x=P^{-1}(r)$

Must know:

1. **The integral of p(x)**
2. **The inverse function $P^{-1}(x)$**

Rejection Methods

Algorithm

Pick r1 and r2

Accept r1 if r2 < p(r1)

1. Easy

2. Wasteful? Efficiency = Area / Area of rectangle

Uniform Random Directions

1. Continuous distribution

Pick a random angle

Set direction

2. Rejection

Pick a random point (r1,r2)

If r...

Efficiency?

Trick:

Uniform Random Rotations

1. 3 random angles -> Euler angles
2. Random direction and random angle

Caution **[Delthiel 1926]:** $$M(R) = \frac{1}{4\pi^2} \iiint \sin^2 \psi \sin\theta \, d\theta \, d\varphi \, d\psi$$

3. Two random directions
   - Orthogonalize
   - Two reflections -> rotation(?)

$$\vec{\mathbf{x}} = \vec{\mathbf{r}}_1$$
$$\vec{\mathbf{y}} = \vec{\mathbf{r}}_2 - \vec{\mathbf{r}}_1(\vec{\mathbf{r}}_1 \bullet \vec{\mathbf{r}}_2)$$
$$\vec{\mathbf{z}} = \vec{\mathbf{x}} \times \vec{\mathbf{y}}$$

Criteria for a uniform random rotation:

Distribution invariant under rotation

Uniform Random Lines

Criteria for a uniform distribution of lines:

Invariant under rotations and translations

2D: $M(L_2) = \int\int dr \, d\theta$ **Poincare 1912**

3D: $M(L_3) = \int\int d\vec{\omega} \bullet d\vec{A}$ **-> throughput!**

Algorithm 1
1. Pick a random direction
2. Pick ...

Algorithm 2
1. Pick a random point on the xy-plane
2. Pick ...

Form Factors

Generate cosine weighted distribution

Probabilistic interpretation of form factors

- **Generate an ensemble of random lines**
- **Record N_i, the number of lines through element i**
- **Record N_{ij}, the number of lines through i and j**
- $F_{ij} = N_{ij} / N_i$

Monte Carlo and Computer Graphics

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

- ***Sample distributions during ray tracing***
- ***Distribute samples amongst dimensions***
- **1. Sample x-y: antialiasing**
- **2. Same time: motion blur**
- **3. Sample lense: depth of field**
- **4. Sample area lights: penumbra**
- **5. Sample specular reflections: gloss**

Path tracing [Kajiya, 1986]

Path Sampling

Monte Carlo solutions to integral equations

Sample path space to generate trajectories

Need not introduce finite state space

$$\sum_{k=0}^{\infty} \int_{M_2} \int_{M_2} \ldots \int_{M_2} f_r(x_k, \omega_k, \omega_k') G(x_k, x_{k-1}) \ldots f_r(x_1, \omega_1, \omega_1') G(x_1, x_0) L_e(x_0, x_1)\, dS_0\, dS_1 \ldots dS_k$$

- **All paths must be sampled**
- **The sampling process must not introduce biases**
- **Typically compute finite set of numbers**

Photon Paths