Monte Carlo Path Tracing

Today

- Path tracing
- Random walks and Markov chains
- Eye vs. light ray tracing
- Bidirectional ray tracing
- Adjoint equations

Next

- Irradiance caching
- Photon mapping

Light Transport

Integrate over all paths

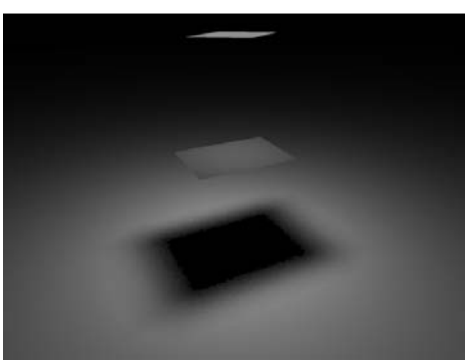
$$L(x_{k-1}, x_k) = \sum_{k=1}^{\infty} \int_{M^2} \cdots \int_{M^2} L_S(x_0, \cdots, x_{k-2}, x_{k-1}, x_k) dA(x_0) \cdots dA(x_{k-2})$$

Questions

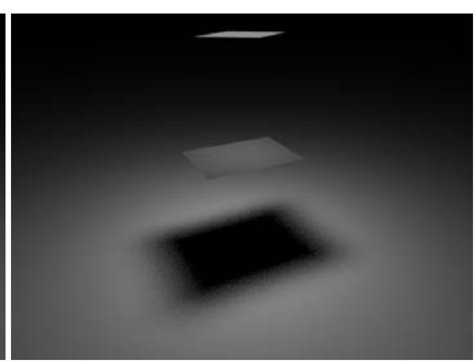
- How to sample space of paths
- Find good estimators

Path Tracing

Penumbra: Trees vs. Paths



4 eye rays per pixel
16 shadow rays per eye ray



64 eye rays per pixel
1 shadow ray per eye ray

Path Tracing: From Camera

```
Step 1. Choose a camera ray r given the
   (x,y,u,v,t) sample
      weight = 1;
Step 2. Find ray-surface intersection
Step 3. Randomly choose to compute Le or Lr
   Step 3a. If Le,
      return weight * Le();
   Step 3b. If Lr,
      weight *= reflectance(r);
      Choose new ray r' ~ BRDF pdf
      Go to Step 2.
```

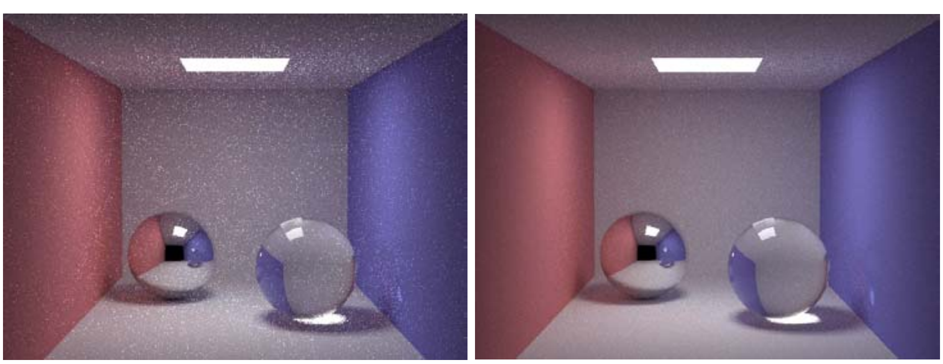

Path Tracing: From Camera

```
Step 1. Choose a camera ray r given
   (x,y,u,v,t)
      weight = 1;
Step 2. Find ray-surface intersection
Step 3. Randomly choose to absorb or reflect
   Step 3a. If absorb,
      return weight * Lr;
   Step 3b. If reflect,
      weight *= reflectance(r);
      Choose new ray r' ~ BRDF pdf
      Go to Step 2.
```

M. Fajardo Arnold Path Tracer

CS348B Lecture 16 Pat Hanrahan, Spring 2004

Cornell Box: Path Tracing



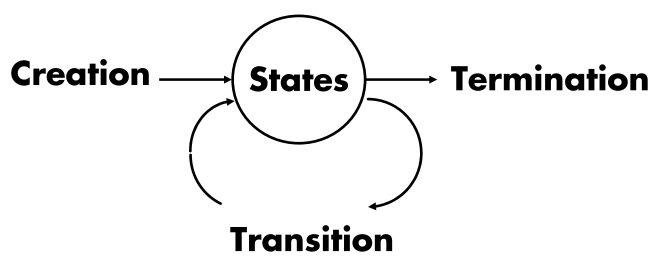
10 rays per pixel 100 rays per pixel

From Jensen, Realistic Image Synthesis Using Photon Maps

CS348B Lecture 16 Pat Hanrahan, Spring 2004

Discrete Random Walk

Discrete Random Process



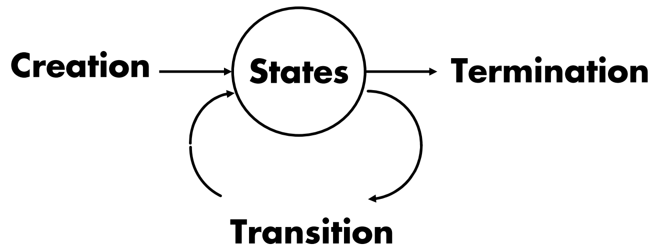
Assign probabilities to each process

p_i^0 : probability of creation in state i

$p_{i,j}$: probability of transition from state $i \rightarrow j$

p_i^* : probability of termination in state i $\quad p_i^* = 1 - \sum_j p_{i,j}$

Discrete Random Walk



1. **Generate random particle paths from source.**
2. **Undertake a discrete random walk.**
3. **Count how many terminate in state i**

[von Neumann and Ulam; Forsythe and Leibler; 1950s]

Monte Carlo Algorithm

Define a random variable on the space of paths

Path: $\alpha_k = (i_1, i_2, \ldots, i_k)$

Probability: $P(\alpha_k)$

Estimator: $W(\alpha_k)$

Expectation:

$$E[W] = \sum_{k=1}^{\infty} \sum_{\alpha_k} P(\alpha_k) W(\alpha_k)$$

Monte Carlo Algorithm

Define a random variable on the space of paths

Probability: $P(\alpha_k) = p^0_{i_1} \times p_{i_1,i_2} \cdots p_{i_{k-1},i_k} \times p^*_{i_k}$

Estimator: $W_j(\alpha_k) = \dfrac{\delta_{i_k,j}}{p^*_{i_k}}$

Estimator

Count the number of particles terminating in state *j*

Unbiased:

$$E[W_j] = \sum_{k=1}^{\infty} \sum_{i_k} \cdots \sum_{i_1} (p^0_{i_1} p_{i_1,i_2} \cdots p_{i_{k-1},i_k} p^*_{i_k}) \frac{\delta_{i_k,j}}{p^*_j}$$
$$= \left[p^0 \right]_j + \left[M p^0 \right]_j + \left[M^2 p^0 \right]_j \cdots$$

$$M_{i,j} = p_{i,j}$$

Equilibrium Distribution of States

Total probability of being in states P

$$P = \left(I + M + M^2 + \cdots\right) p^0$$

Which is the same as the Monte Carlo estimator for a discrete random walk

Note that this is the solution of the equation

$$(I - M)P = p^0$$

Variations

Particle interaction:

- **Reflect with probability R**
- **Absorb with probability A = 1-R**

Variation 0:

Set w *= R;

Unbiased, but never terminates

Variation 1:

Terminate with probability A

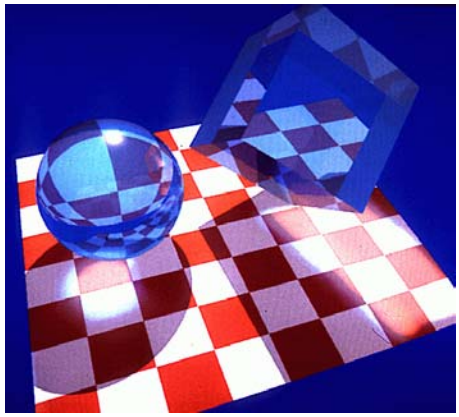
Unbiased, but never terminates if R = 1

Russian Roulette

```
if( w < threshold ) {
   r = rand();
   if( r < P )
      terminate;
   else
      w /= (1-P);
}
```


Light Ray Tracing

Examples



Backward ray tracing, Arvo 1986

Path Tracing: From Lights

```
Step 1. Choose a light ray

   Choose a light source according to the
   light source power distribution function.

   Choose a ray from the light source
   radiance (area) or intensity (point)
   distribution function

   w = 1;

Step 2. Trace ray to find surface intersect

Step 3. Interaction
```

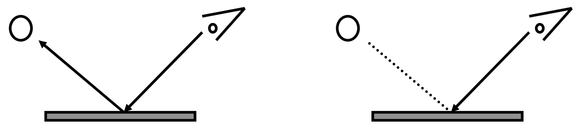
Path Tracing: From Lights

```
Step 1. Choose a light ray
Step 2. Find ray-surface intersection
Step 3. Interaction
  u = rand()
  if u < reflectance
    Choose new ray r ~ BRDF
    goto Step 2
  else
    terminate on surface; deposit energy
```

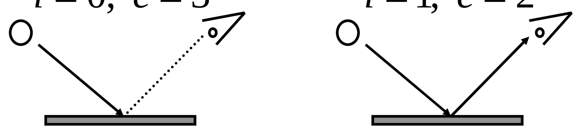

Bidirectional Path Tracing

Bidirectional Ray Tracing

$$k = l + e$$



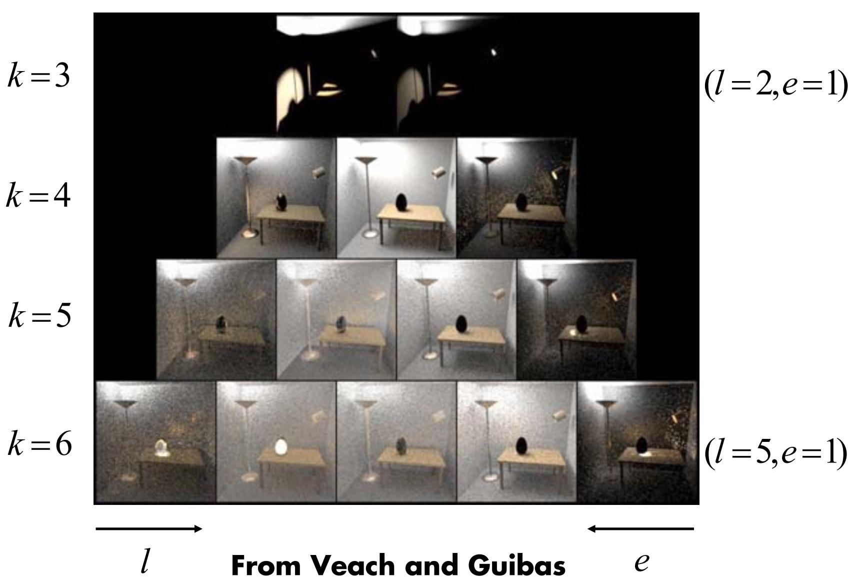
$l = 0,\ e = 3$ $\quad$ $l = 1,\ e = 2$



$l = 2,\ e = 1$ $\quad$ $l = 3,\ e = 0$

$$k = 3$$

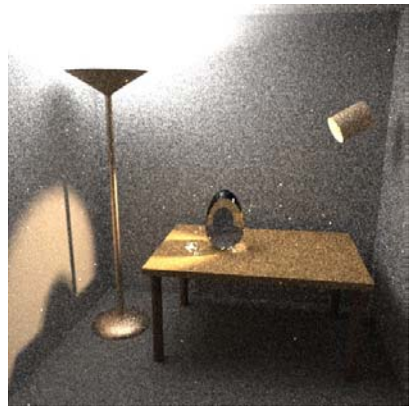
Path Pyramid



From Veach and Guibas

Comparison

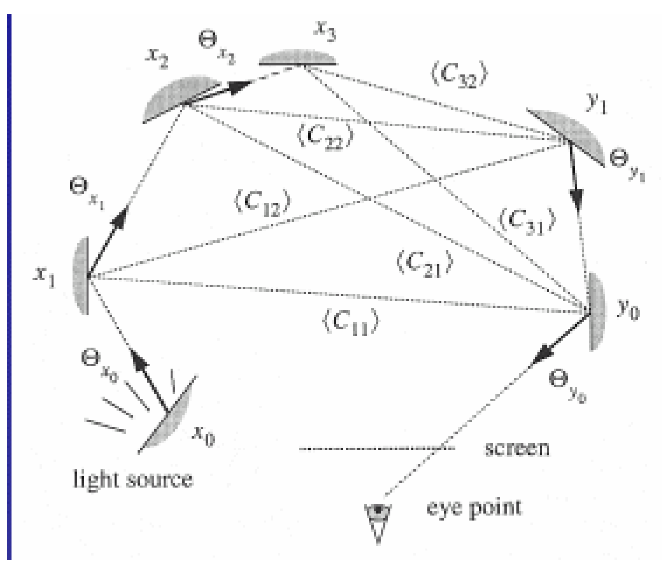
Bidirectional path tracing



Path tracing

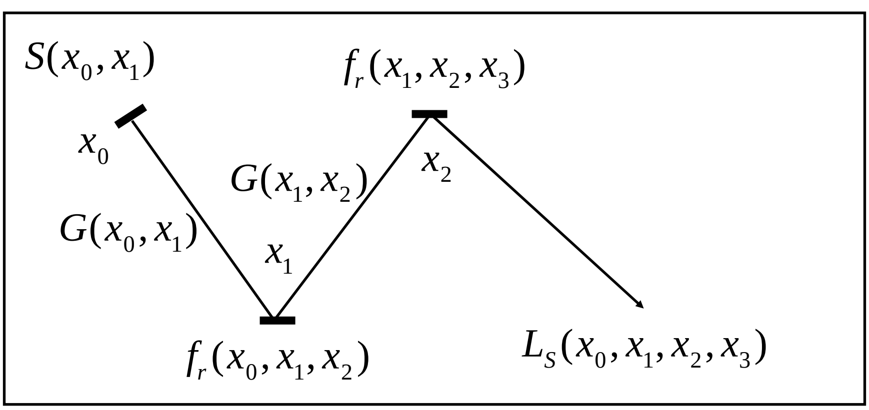
From Veach and Guibas

Generating All Paths



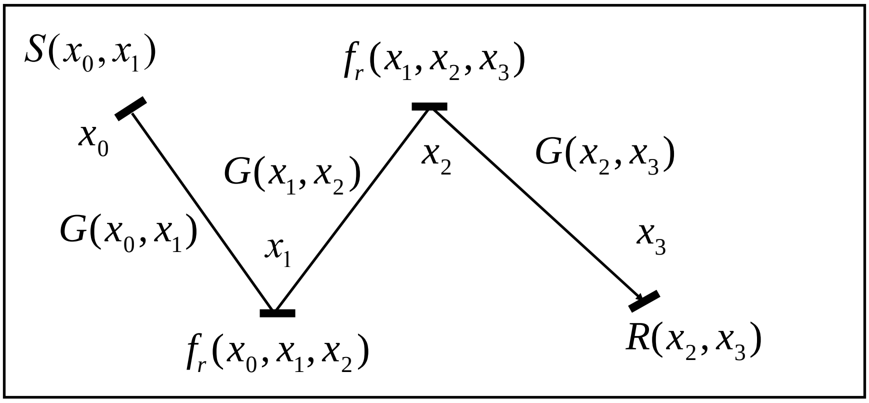
Adjoint Formulation

Light Path



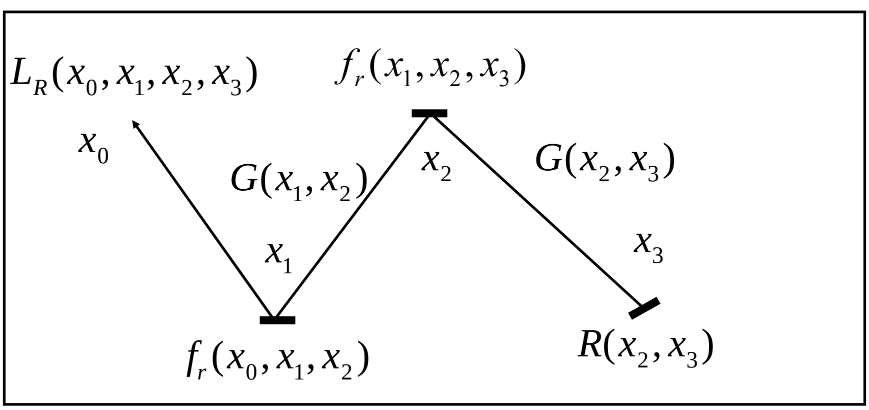
$$L_S(x_0, x_1, x_2, x_3) = S(x_0, x_1) G(x_o, x_1) f_r(x_0, x_1, x_2) G(x_1, x_2) f_r(x_1, x_2, x_3)$$

Symmetric Light Path



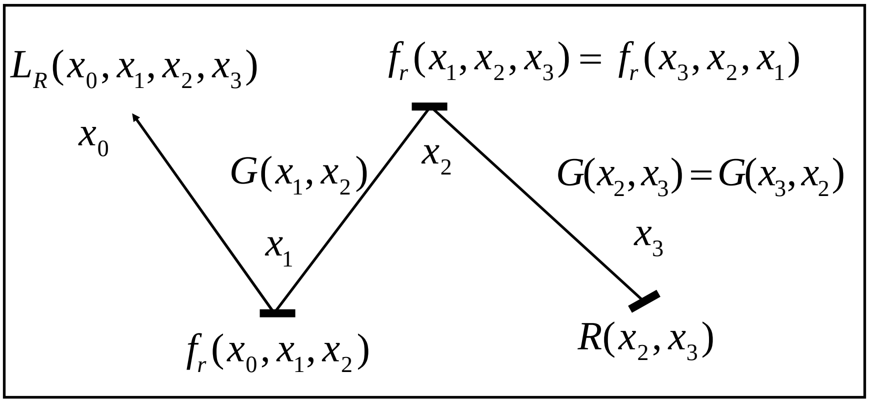
$$M = S(x_0, x_1) G(x_o, x_1) f_r(x_0, x_1, x_2) G(x_1, x_2) f_r(x_1, x_2, x_3) G(x_2, x_3) R(x_2, x_3)$$

Symmetry



$$L_R(x_0, x_1, x_2, x_3) = f_r(x_0, x_1, x_2) G(x_1, x_2) f_r(x_1, x_2, x_3) G(x_2, x_3) R(x_2, x_3)$$

Symmetry



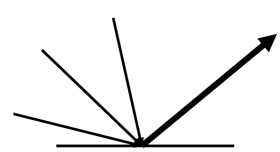
$$
\begin{aligned}
L_R(x_0, x_1, x_2, x_3) &= f_r(x_0, x_1, x_2) G(x_1, x_2) f_r(x_1, x_2, x_3) G(x_2, x_3) R(x_2, x_3) \\
&= R(x_3, x_2) G(x_3, x_2) f_r(x_3, x_2, x_1) G(x_2, x_1) f_r(x_2, x_1, x_0)
\end{aligned}
$$

Adjoint Equations

Original equation

$$K \circ f = \int K(x, y) f(y)\, dy$$

Forward direction

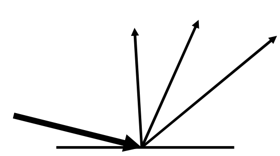


Out-Scatter

Adjoint equation

$$K^{+} \circ f = \int K(x, y) f(x)\, dx$$

Backward direction



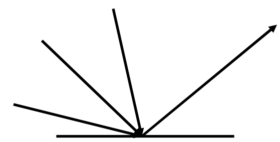
In-Scatter

Self-Adjoint Equations

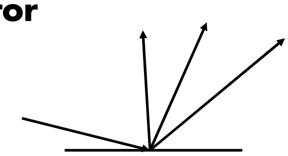
Self-adjoint

$$K^{+} = K$$

Forward = backward operator



Out-Scatter



In-Scatter

Forward = Backward Estimate

$$< f,g >= \int f(x)g(x)\,dx$$

$$\begin{aligned} < f,K \circ g > &= \int f(x)\left(\int K(x,y)g(y)\,dy\right)dx \\ &= \int\left(\int f(x)K(x,y)dx\right)g(y)\,dy \\ &=< K^{+} \circ f,g > \end{aligned}$$

Three Consequences

1. Forward estimate equal backward estimate
   - May use forward or backward ray tracing

2. Adjoint solution
   - Importance sampling paths

3. Solve for small subset of the answer

Example: Linear Equations

Solve a linear system $Mx = b$

Solve for a single x_i?

Solve the adjoint equation

Source x_i

Estimator $< (x_i + Mx_i + M^2 x_i + \cdots), b >$

More efficient than solving for all the unknowns [von Neumann and Ulam]