Metropolis Sampling

Matt Pharr
cs348b

May 20, 2003

Introduction

- Unbiased MC method for sampling from functions' distributions
- Robustness in the face of difficult problems
- Application to a wide variety of problems
- Flexibility in choosing how to sample
- Introduced to CG by Veach and Guibas

Overview

- For arbitrary $f(x) \rightarrow \mathbb{R}$, $x \in \Omega$

Overview

- For arbitrary $f(x) \rightarrow \mathbb{R}$, $x \in \Omega$
- Define $\mathbf{I}(f) = \int_\Omega f(x) \mathrm{d}\Omega$ so $f_{\text{pdf}} = f/\mathbf{I}(f)$

Overview

- For arbitrary $f(x) \rightarrow \mathbb{R}$, $x \in \Omega$
- Define $\mathbf{I}(f) = \int_{\Omega} f(x) \mathrm{d}\Omega$ so $f_{\mathrm{pdf}} = f / \mathbf{I}(f)$
- Generates samples $X = \{x_i\}$, $x_i \sim f_{\mathrm{pdf}}$
- *Without needing to compute* f_{pdf} *or* $\mathbf{I}(f)$

Overview

- Introduction to Metropolis sampling
- Examples with 1D problems
- Extension to 3D, motion blur
- Overview of Metropolis Light Transport

Basic Algorithm

- Function $f(x)$ over state space Ω, $f : \Omega \longrightarrow \mathbb{R}$.
- Markov Chain: new sample x_i using x_{i-1}

Basic Algorithm

- Function $f(x)$ over state space Ω, $f : \Omega \rightarrow \mathbb{R}$.
- Markov Chain: new sample x_i using x_{i-1}
- New samples from *mutation* to $x_{i-1} \rightarrow x'$
- Mutation accepted or rejected so $x_i \sim f_{\mathrm{pdf}}$
- If rejected, $x_i = x_{i-1}$
- Acceptance guarantees distribution of x_i is the *stationary distribution*

Pseudo-code

```
x = x0
for i = 1 to n
    x' = mutate(x)
    a = accept(x, x')
    if (random() < a)
        x = x'
    record(x)
```

Expected Values

- Metropolis avoids parts of Ω where $f(x)$ is small
- But e.g. dim parts of an image need samples
- Record samples at both x and x'
- Samples are weighted based on $a(x \to x')$
- Same result in the limit

Expected Values – Pseudo-code

```
x = x0
for i = 1 to n
    x’ = mutate(x)
    a = accept(x, x’)
    record(x, (1-a) * weight)
    record(x’, a * weight)
    if (random() < a)
        x = x’
```

Mutations, Transitions, Acceptance

- Mutations propose x' given x_i
- $T(x \to x')$ is probability density of proposing x' from x
- $a(x \to x')$ probability of accepting the transition

Detailed Balance – The Key

- By defining $a(x \to x')$ carefully, can ensure $x_i \sim f(x)$

$$f(x)\,T(x \to x')\,a(x \to x') = f(x')\,T(x' \to x)\,a(x' \to x)$$

- Since f and T are given, gives conditions on acceptance probability
- (Will not show derivation here)

Acceptance Probability

- Efficient choice:

$$a(x \to x') = \min\left(1, \frac{f(x')\,T(x' \to x)}{f(x)\,T(x \to x')}\right)$$

Acceptance Probability – Example I

- If $\Omega = a, b$ and $f(a) = 9$, $f(b) = 1$
- If

$$\text{mutate}(x) = \begin{cases} a & : \quad \xi < 0.5 \\ b & : \quad \text{otherwise} \end{cases}$$

- Then transition densities are

$$T(\{a, b\} \rightarrow \{a, b\}) = 1/2$$

Acceptance Probability – Example I

- It directly follows that

$$a(a \to b) = \min\left(1, f(b)/f(a)\right) = .1111\ldots$$

$$a(a \to a) = a(b \to a) = a(b \to b) = 1$$

Acceptance Probability – Example I

- It directly follows that

$$a(a \to b) = \min\left(1, f(b)/f(a)\right) = .1111\ldots$$

$$a(a \to a) = a(b \to a) = a(b \to b) = 1$$

- Recall (simplified) detailed balance

$$f(a)\, a(a \to b) = f(b)\, a(b \to a)$$

Acceptance Probability - Example II

- If

$$\text{mutate}(x) = \begin{cases} a & : \quad \xi < 8/9 \\ b & : \quad \text{otherwise} \end{cases}$$

- Then transition densities are

$$T(\{a, b\} \to a) = 8/9$$

$$T(\{a, b\} \to b) = 1/9$$

Acceptance Probability - Example II

- Acceptance probabilities are

$$a(a \rightarrow b) = .9/.9 = 1$$

$$a(b \rightarrow a) = .9/.9 = 1$$

- Better transitions improve acceptance probability

Acceptance Probability – Goals

- Doesn't affect unbiasedness; just variance
- Maximize the acceptance probability $\rightarrow$
  - Explore state space better
  - Reduce correlation (image artifacts...)
- Want transitions that are likely to be accepted
  - i.e. transitions that head where $f(x)$ is large

Mutations: Metropolis

- $T(a \rightarrow b) = T(b \rightarrow a)$ for all a, b

$$a(x \rightarrow x') = \min\left(1, \frac{f(x')}{f(x)}\right)$$

- Random walk Metropolis

$$T(x \rightarrow x') = T(|x - x'|)$$

Mutations: Independence Sampler

- If we have some pdf p, can sample $x \sim p$,
- Straightforward transition function:

$$T(x \to x') = p(x')$$

- If $p(x) = f_{\mathrm{pdf}}$, wouldn't need Metropolis
- But can use pdfs to approximate parts of f...

Mutation Strategies: General

- Adaptive methods: vary transition based on experience
- Flexibility: base on value of $f(x)$, etc. pretty freely
- Remember: just need to be able to compute transition densities for the mutation
- The more mutations, the merrier
- Relative frequency of them not so important

1D Example

- Consider the function

$$f^1(x) = \begin{cases} (x - .5)^2 & : \quad 0 \leq x \leq 1 \\ 0 & : \quad \text{otherwise} \end{cases}$$

- Want to generate samples from $f^1(x)$

1D Mutation #1

$$\begin{aligned} \text{mutate}_1(x) &\rightarrow \xi \\ T_1(x \rightarrow x') &= 1 \end{aligned}$$

- Simplest mutation possible
- Random walk Metropolis

1D Mutation #2

$$\text{mutate}_2(x) \;\rightarrow\; x + .1 * (\xi - .5)$$

$$T_2(x \rightarrow x') \;=\; \begin{cases} \frac{1}{0.1} & : \quad |x - x'| \leq .05 \\ 0 & : \quad \text{otherwise} \end{cases}$$

- Also random walk Metropolis

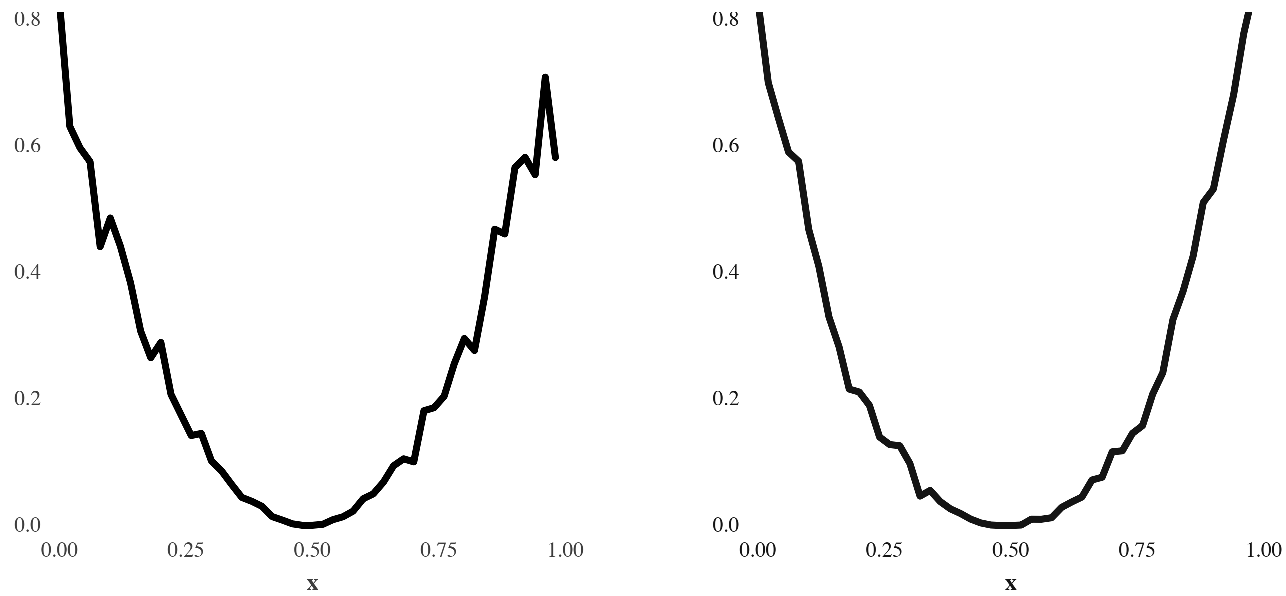
1D Mutation #2

- mutate_2 increases acceptance probability

$$a(x \to x') = \min\left(1, \frac{f(x')\,T(x' \to x)}{f(x)\,T(x \to x')}\right)$$

- When $f(x)$ is large, will avoid x' when $f(x') < f(x)$
- Should try to avoid proposing mutations to such x'

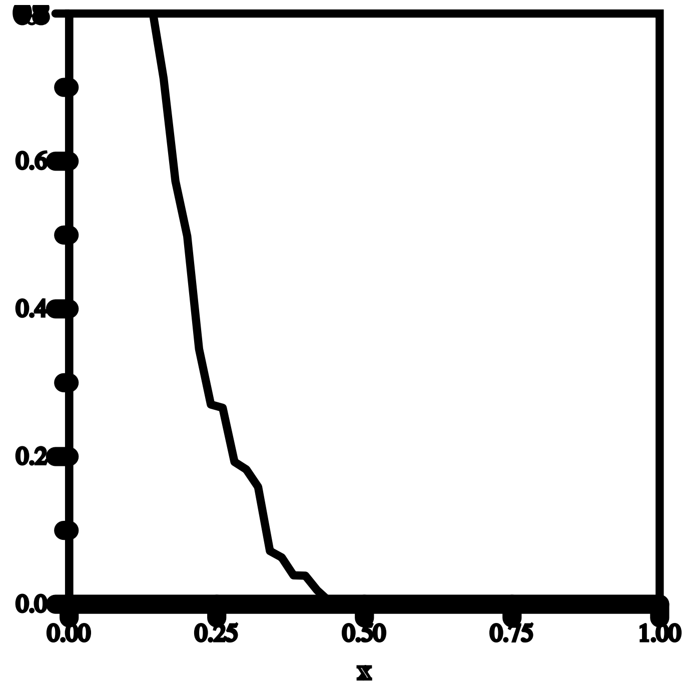
1D Results - pdf graphs



- Left: mutate_1 only
- Right: a mix of the two (10%/90%)
- 10,000 mutations total

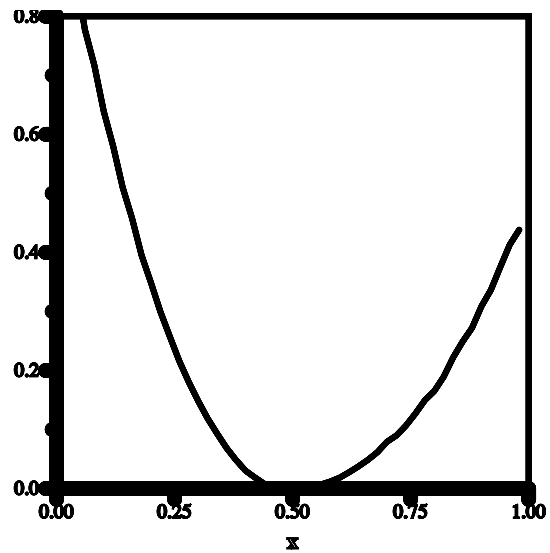
Why bother with mutate_1, then?

- If we just use the second mutation ($\pm .05$)...



Ergodicity

- Need finite prob. of sampling x, $f(x) > 0$
- This is true with mutate_2, but is inefficient:



- Still unbiased in the limit...

Ergodicity – Easy Solution

- Periodically pick an entirely new x
- e.g. sample uniformly over Ω...

Motion Blur

- Onward to a 3D problem
- Scene radiance function $L(u, v, t)$ (e.g. evaluated with ray tracing)
- $L = 0$ outside the image boundary
- Ω is $(u, v, t) \in [0, \mathrm{u}_{\mathrm{max}}] \times [0, \mathrm{v}_{\mathrm{max}}] \times [0, 1]$

Application to Integration

- Given integral, $\int f(x)g(x)\mathrm{d}\Omega$
- Standard Monte Carlo estimator:

$$\int_{\Omega} f(x)g(x)\,\mathrm{d}\Omega \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)g(x_i)}{p(x_i)}$$

- where $x_i \sim p(x)$, an arbitrary pdf

Application to Integration

$$\int_{\Omega} f(x)g(x)\,\mathrm{d}\Omega \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)g(x_i)}{p(x_i)}$$

Application to Integration

$$\int_\Omega f(x)g(x)\,\mathrm{d}\Omega \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)g(x_i)}{p(x_i)}$$

- Metropolis gives $x_1, \ldots, x_N$, $x_i \sim f_{\mathrm{pdf}}(x)$

$$\int_\Omega f(x)g(x)\,\mathrm{d}\Omega \approx \left[\frac{1}{N}\sum_{i=1}^{N} g(x_i)\right] \cdot \mathbf{I}(f)$$

- (Recall $\mathbf{I}(f) = \int_\Omega f(x)\mathrm{d}\Omega$)

Image Contribution Function

- The key to applying Metro to image synthesis

$$I_j = \int_\Omega h_j(u,v)\, L(u,v,t)\, \mathrm{d}u\, \mathrm{d}v\, \mathrm{d}t$$

- I_j is value of j'th pixel
- h_j is pixel reconstruction filter

Image Contribution Function

- So if we sample $x_i \sim L_{\text{pdf}}$

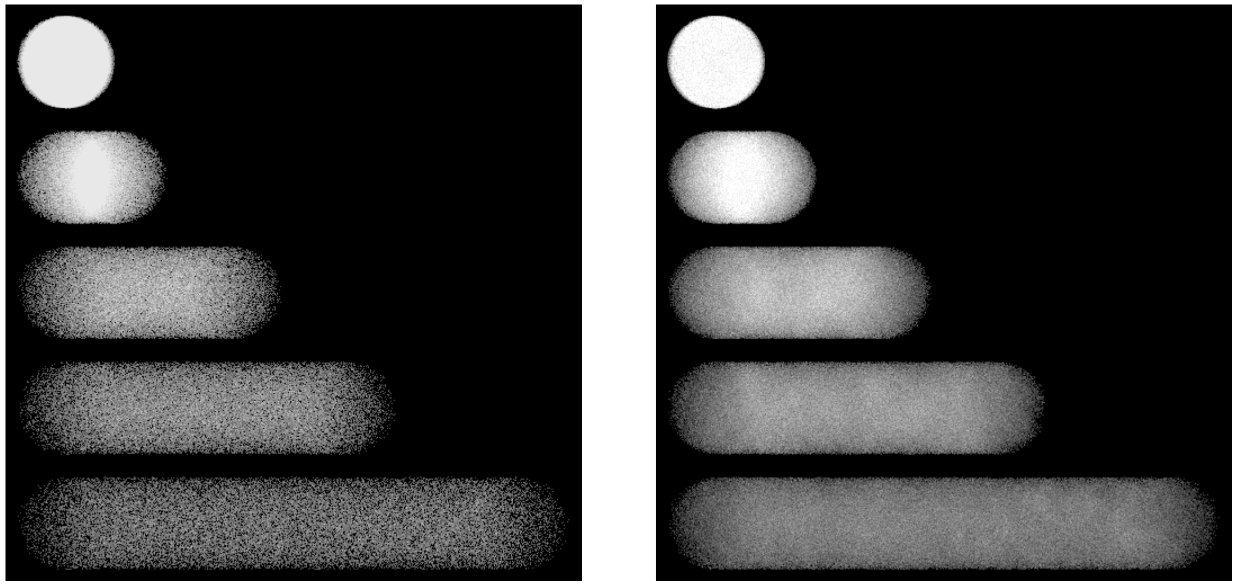
$$I_j \approx \frac{1}{N} \sum_{i=1}^{N} h_j(x_i) \cdot \left(\int_\Omega L(x) \, \mathrm{d}\Omega \right),$$

- *The distribution of x_i on the image plane forms the image*
- Estimate $\int_\Omega L(x) \, \mathrm{d}\Omega$ with standard MC

Two Basic Mutations

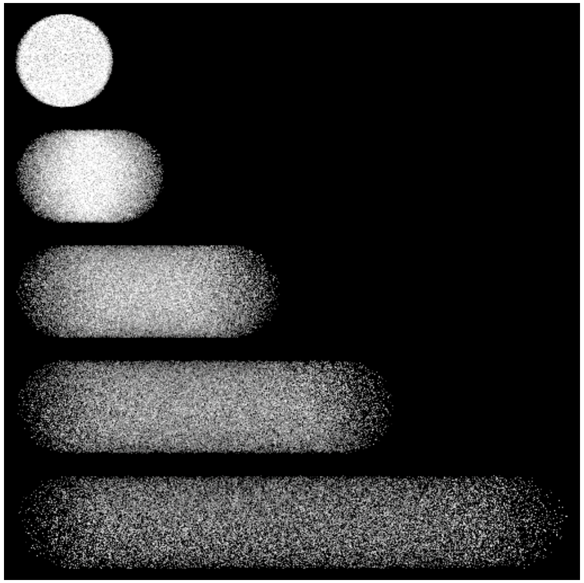
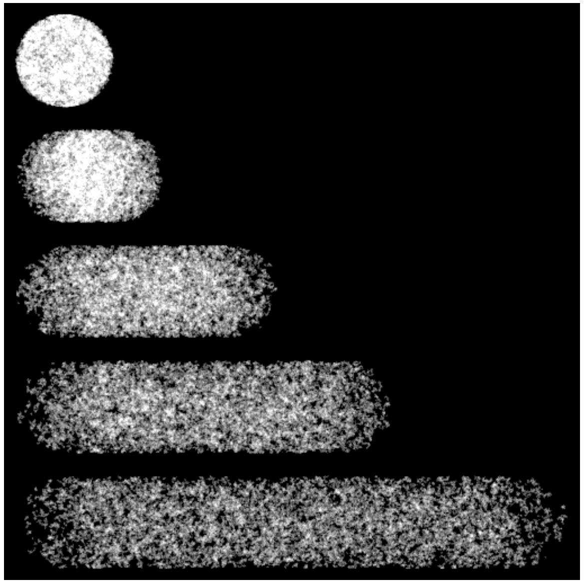
- Pick completely new (u, v, t) value
- Perturb u and v ± 8 pixels, time $\pm .01$.
- Both are symmetric, Random-walk Metropolis

Motion Blur – Result



- Left: Distribution RT, stratified sampling
- Right: Metropolis sampling
- Same total number of samples

Motion Blur – Parameter Studies

- Left: ± 80 pixels, $\pm .5$ time. Many rejections.
- Right: ± 0.5 pixels, $\pm .001$ time. Didn't explore Ω well.

Exponential Distribution

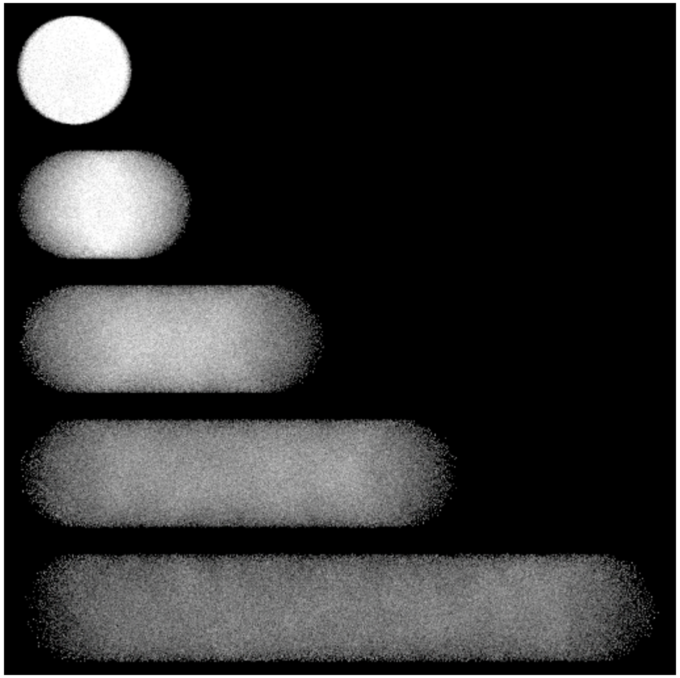
- Vary the scale of proposed mutations

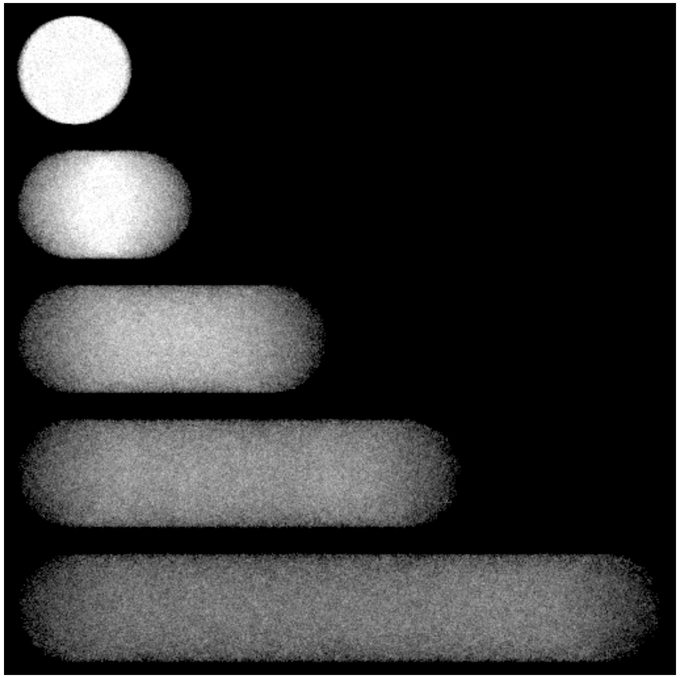
$$r = r_{\max} \, \mathrm{e}^{-\log(r_{\max}/r_{\min})\xi}, \quad \theta = 2\pi\xi$$

$$(du, dv) = (r \sin\theta, r\cos\theta)$$
$$dt = t_{\max} \, \mathrm{e}^{-\log(t_{\max}/t_{\min})\xi}$$

- Will reject when too big, still try wide variety

Exponential distribution results



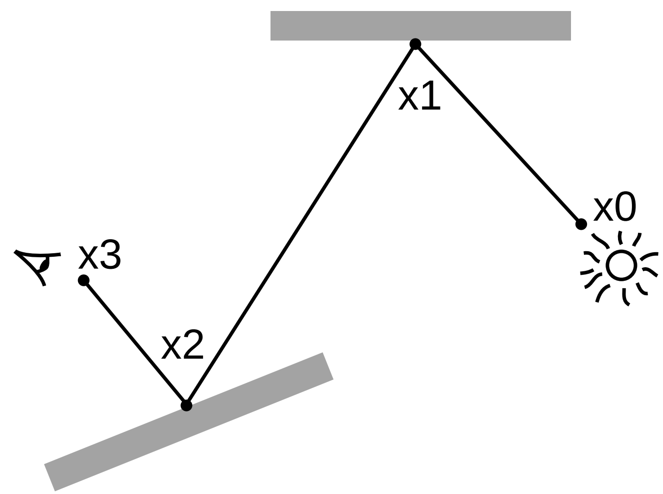


Light Transport

- Image contribution function was key
- $f(x)$ over infinite space of paths
- State-space is light-carrying paths through the scene–from light source to sensor
- Robustness is particularly nice–solve difficult transport problems efficiently
- Few specialized parameters to set

Light Transport – Setting

- Samples x from Ω are sequences $v_0 v_1 \ldots v_k$, $k \geq 1$, of vertices on scene surfaces



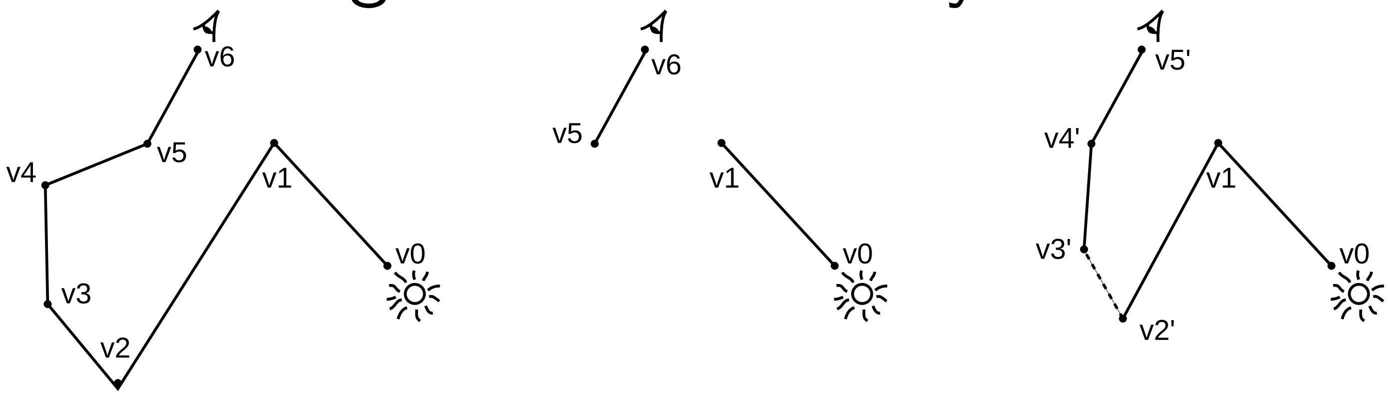
- $f(x)$ is the product of emitted light, BRDF values, cosines, etc.

Light Transport – Strategy

- Explore the infinite-dimensional path space
- Metropolis's natural focus on areas of high contribution makes it efficient
- New issues:
  - Stratifying over pixels
  - Perceptual issues
  - Spectral issues
  - Direct lighting

Bidirectional Mutation

- Delete a subpath from the current path
- Generate a new one
- Connect things with shadow rays



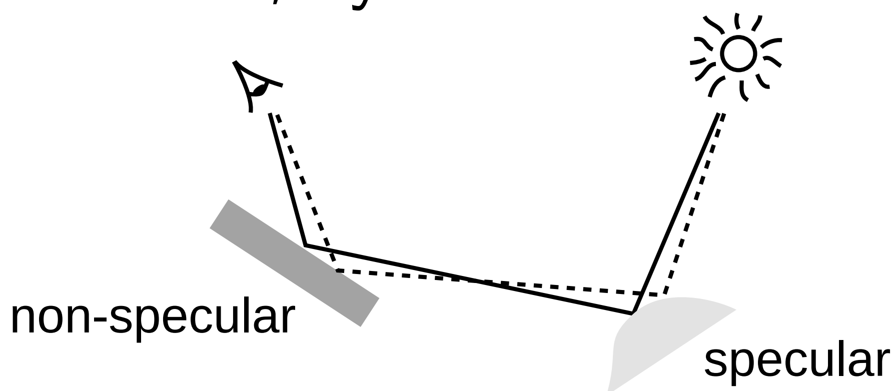
- If occluded, then just reject

Bidirectional Mutation

- Very flexible path re-use
- Ensures ergodicity–may discard the entire path
- Inefficient when a very small part of path space is important
- Transition densities are tricky: need to consider *all* possible ways of sampling the path

Caustic Perturbation

- Caustic path: one more more specular surface hits before diffuse, eye



- Slightly shift outgoing direction from light source, regenerate path

Lens Perturbation

- Similarly perturb outgoing ray from camera
- Keeps image samples from clumping together

Why It Works Well

- Path Reuse
  - Efficiency–paths are built from pieces of old ones
  - (Could be used in stuff like path tracing...)
- Local Exploration
  - Given important path, incrementally sample close to it in Ω
  - When f is small over much of Ω, this is extra helpful

Conclusion

- A very different way of thinking about integration
- Robustness is highly attractive
- Implementation can be tricky