# Path Integral Methods for Light Transport Simulation: Theory & Practice

## Introduction to Markov Chain and Sequential Monte Carlo

# Markov Chains

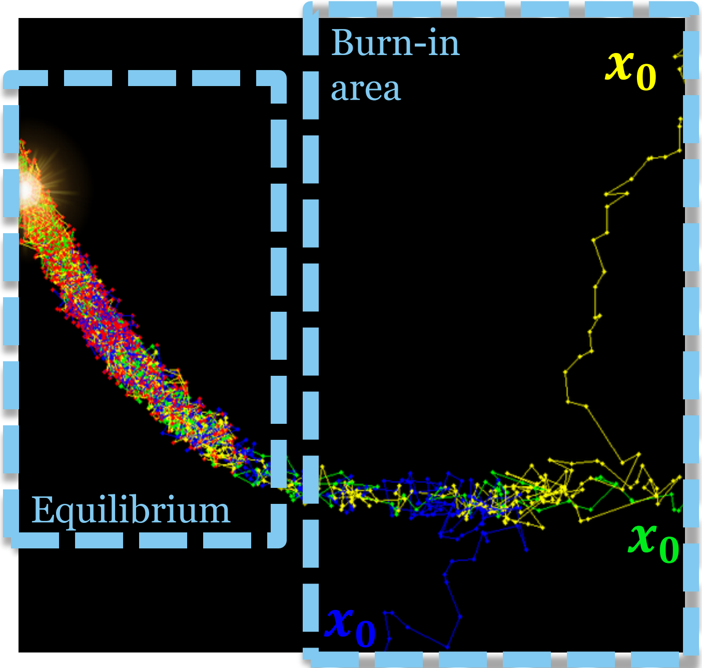
# Markov Chain

- Random walk implies a *transition probability* for each move
$$P(x_{n+1} = j | x_n = i) \equiv P_{i\to j}$$
- At each move the chain forms a *posterior distribution* over state space
  - A histogram of all visited states up to move $n$
- *Detailed balance* defined as $P_{i\to j} = P_{j\to i}$

# Markov Chain

- Posterior converges to the *target* distribution *if* the detailed balance obeyed and all states are reachable (*ergodicity*)

- With "bad" initial state $x_0$ the *start-up bias* (burn-in phase) can be significant



# Metropolis-Hastings Algorithm

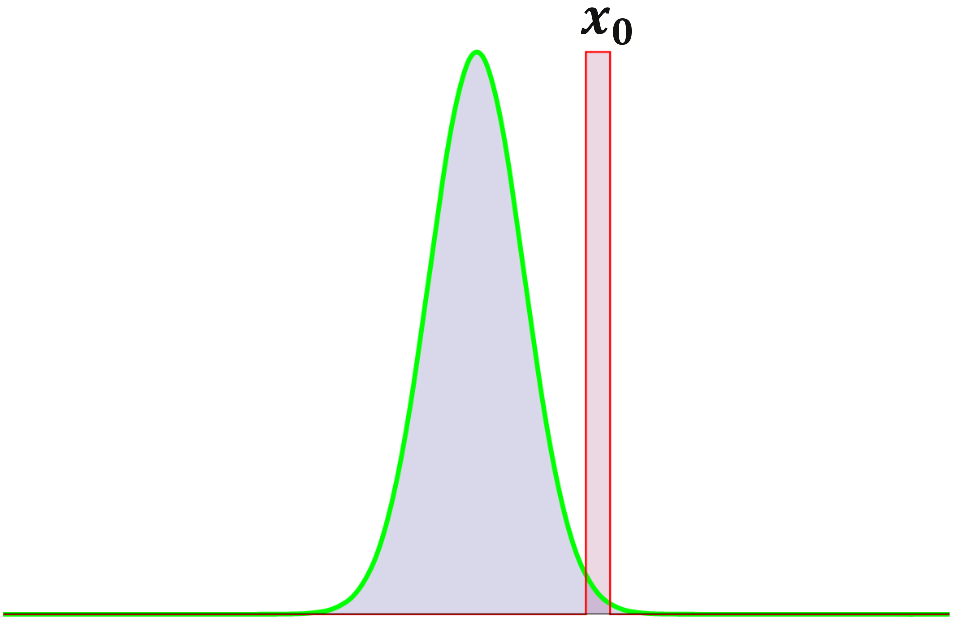
# Metropolis-Hastings (MH) Algorithm

- Goal: Random walk according to a desired function $f$
- Define conditional rejection sampling probability

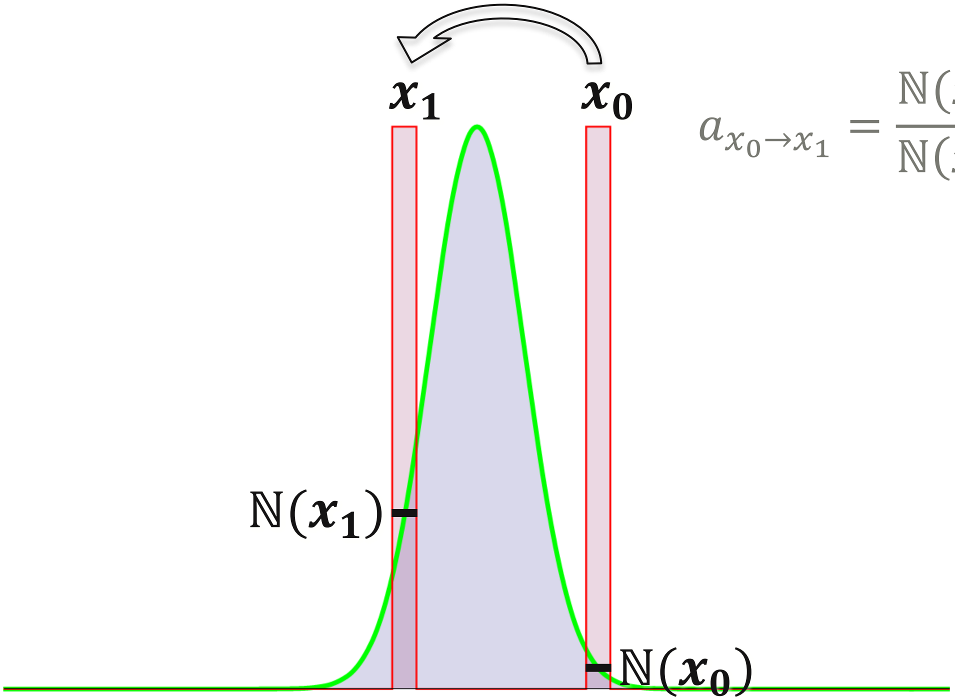
$$a_{i\to j} = \frac{f(x_j)}{f(x_i)} = \frac{f_j}{f_i}$$

  - $a_{i\to j}$ is *acceptance probability* at state $i$ for proposal state $j$
- Detailed balance is affected as $a_{i\to j} P_{i\to j} = a_{j\to i} P_{j\to i}$
- Posterior distribution is then proportional to $f$
  - Accurate to a scaling factor = normalization constant

# Metropolis-Hastings: Example

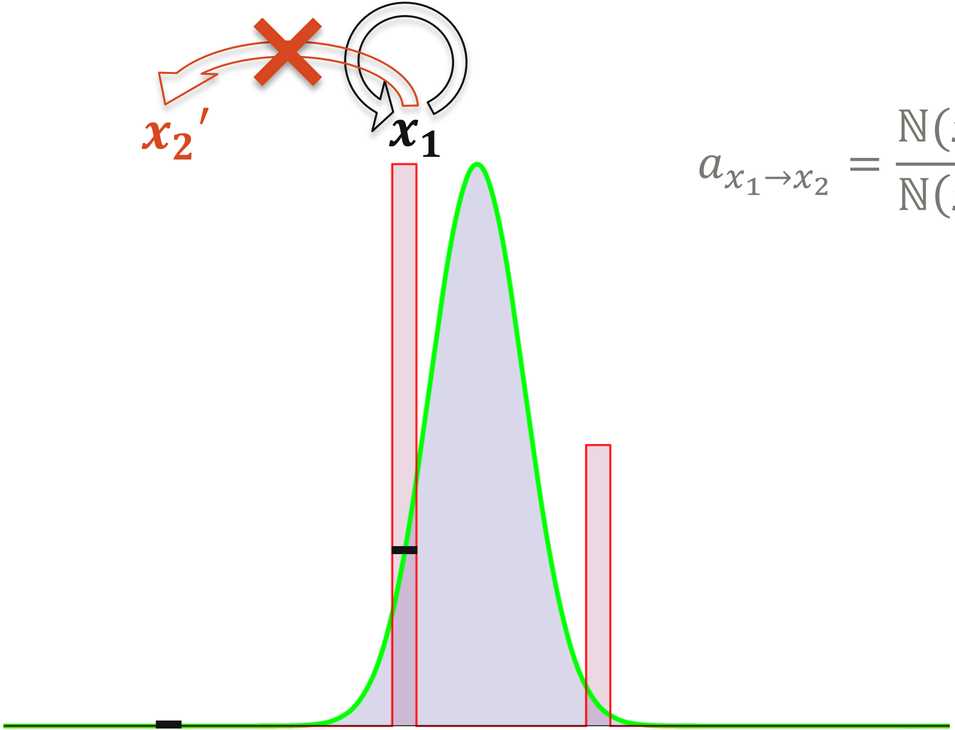


# Metropolis-Hastings: Example



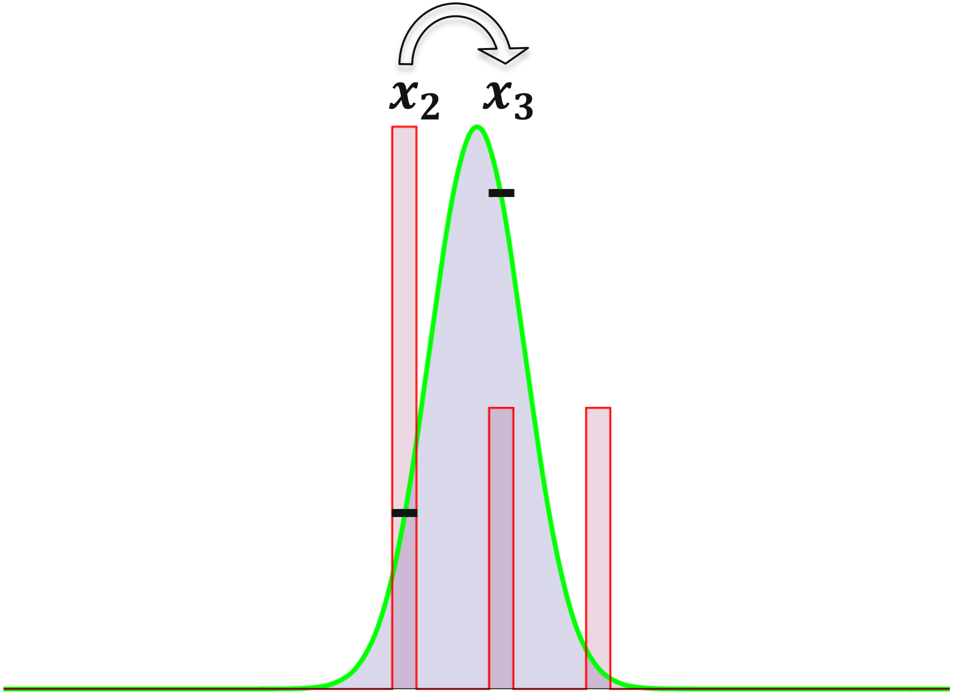

$$a_{x_0 \to x_1} = \frac{\mathbb{N}(x_1)}{\mathbb{N}(x_0)} > 1$$

# Metropolis-Hastings: Example

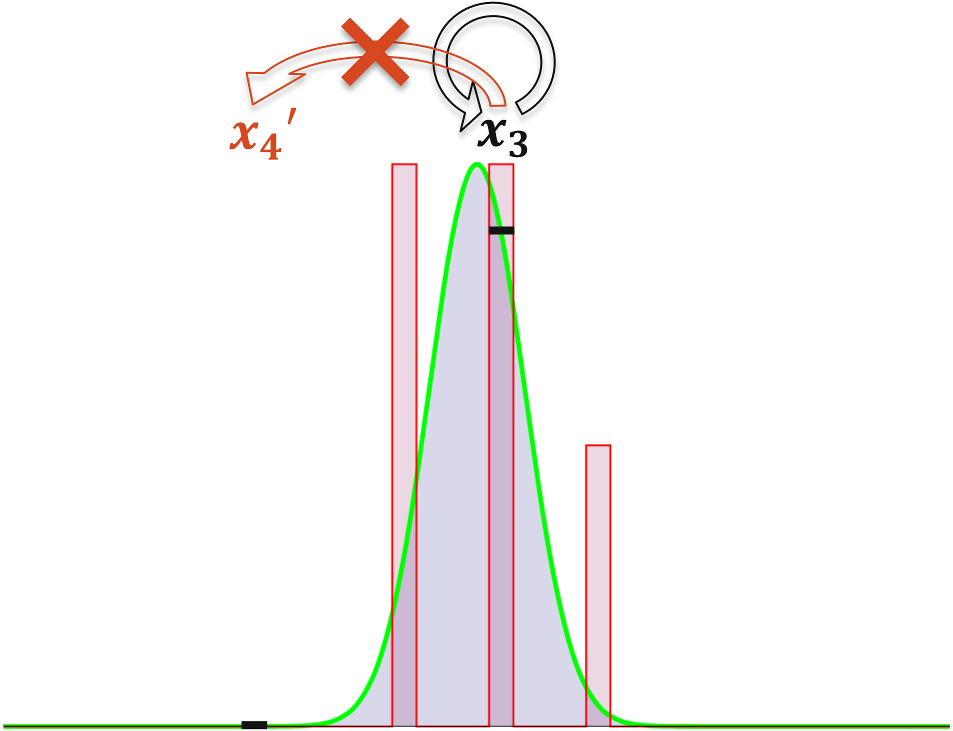



$$a_{x_1 \to x_2} = \frac{\mathbb{N}(x_2)}{\mathbb{N}(x_1)} \ll 1$$

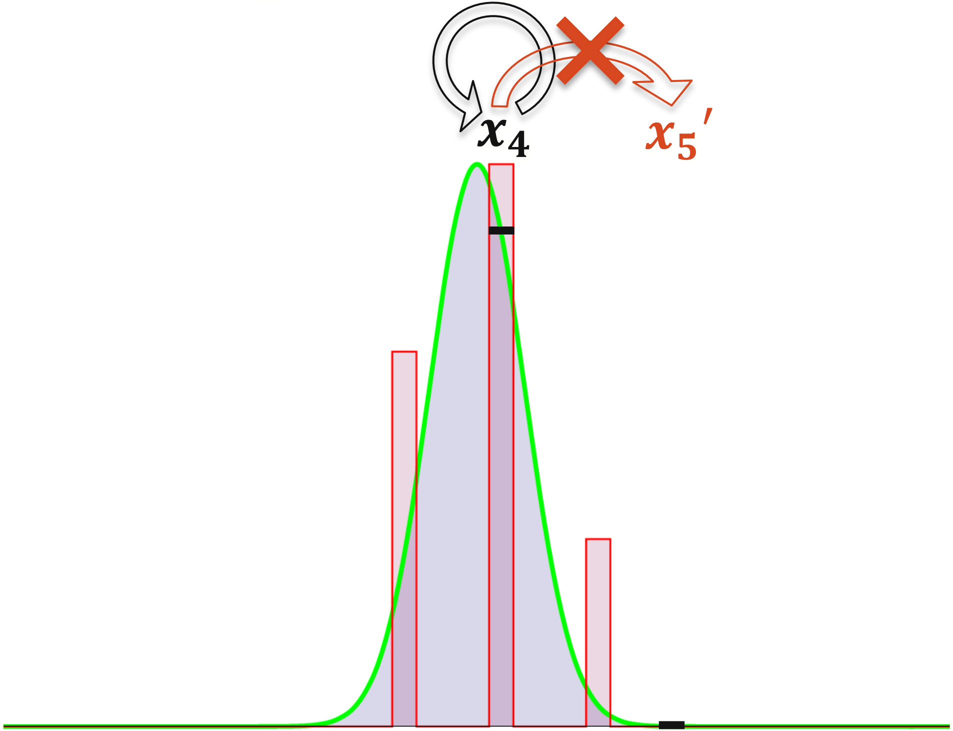
# Metropolis-Hastings: Example



# Metropolis-Hastings: Example

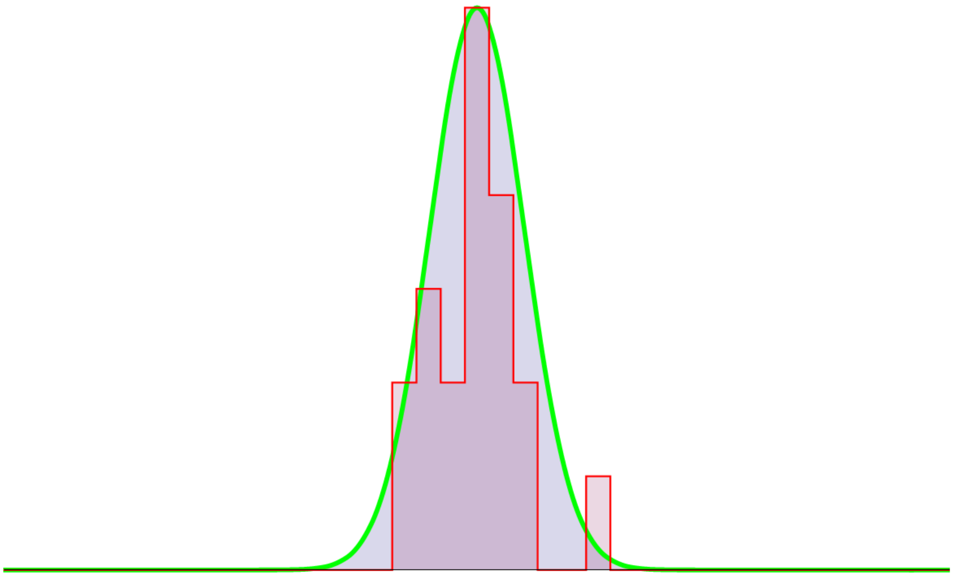


# Metropolis-Hastings: Example



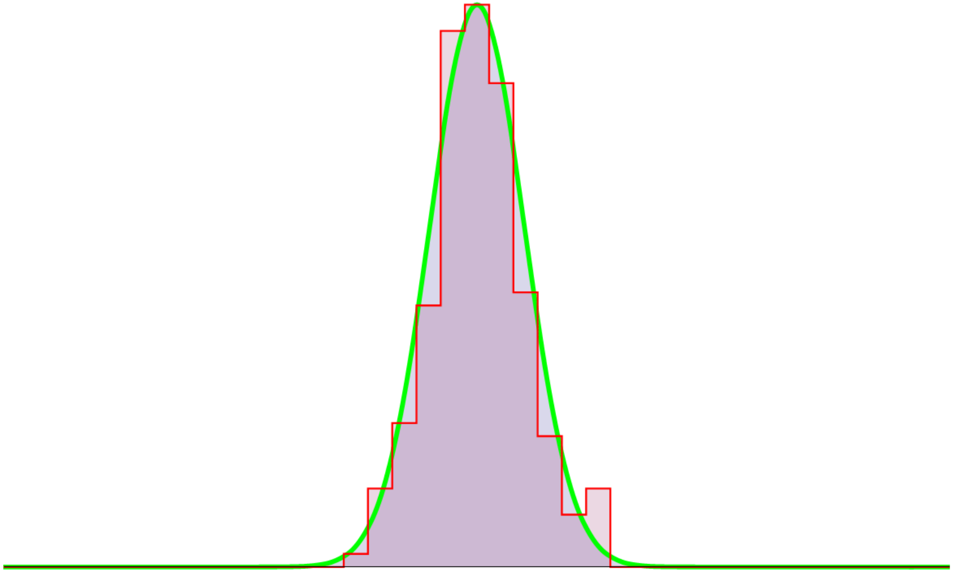
# Metropolis-Hastings: Example

$n = 20$



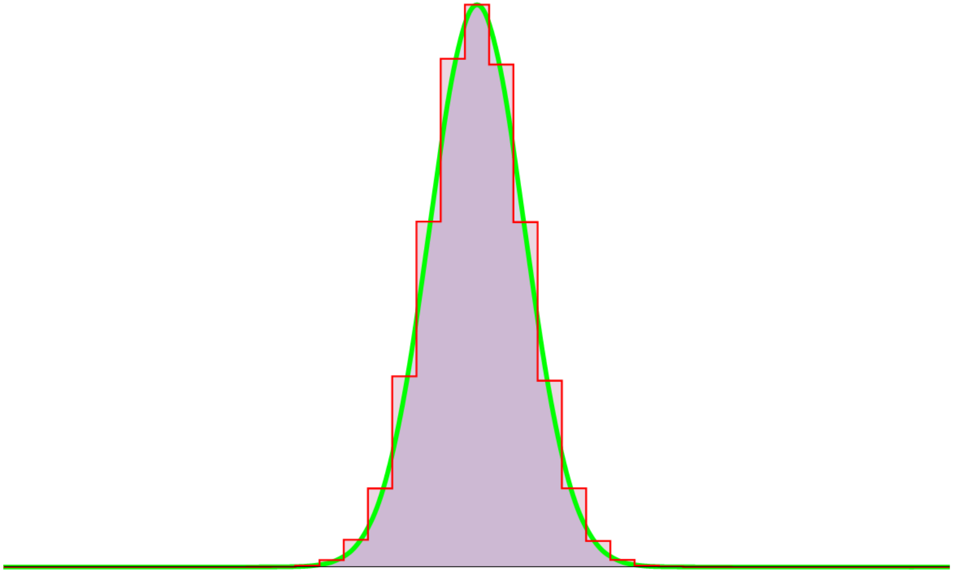
# Metropolis-Hastings: Example

$n = 200$



# Metropolis-Hastings: Example

$$n = 2000$$



# Importance Sampling for M-H

- Cannot fetch proposals directly from $f$
- Generate a proposal $j$ from some *proposal distribution* $T$
  - Similar to importance sampling in Monte Carlo
  - $T$ can depend on the current state $i$: $T_{i \to j}$
  - New transition probability $P_{i \to j} = a_{i \to j} T_{i \to j}$
- Acceptance probability is then (from detailed balance):

$$a_{i \to j} = \left( \frac{f_j}{T_{i \to j}} \right) / \left( \frac{f_i}{T_{j \to i}} \right)$$

# Correspondence Table

| Ordinary Monte Carlo | Markov chain Monte Carlo |
|---|---|
| Convergence rate, usually $O(\frac{1}{\sqrt{N}})$ | Mixing rate, depends on multiple factors, can be geometric $O(\gamma^N), \gamma \in (0;1)$ |
| Convergence to an expected value | Convergence of the posterior to the target distribution (e.g., in total variation) |
| Importance sampling distribution $p(x)$ | Proposal distribution $T_{i \to j}$ |
| Variance of the estimate | Acceptance rate, correlation of samples |
| Number of samples | Number of moves (mutations) |

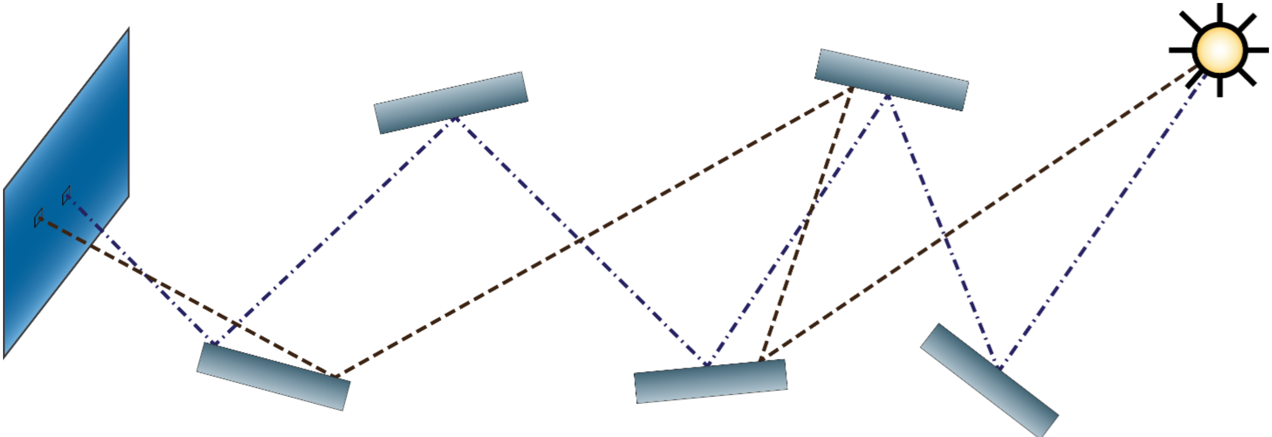
# Metropolis Light Transport

# Image Generation

- Reduce per-pixel integrals to a single integral
  - Each pixel has an individual filter function then
- Compute the distribution over the image plane
  - Bin this distribution into corresponding pixels
- Walk over the image plane

# Metropolis Light Transport

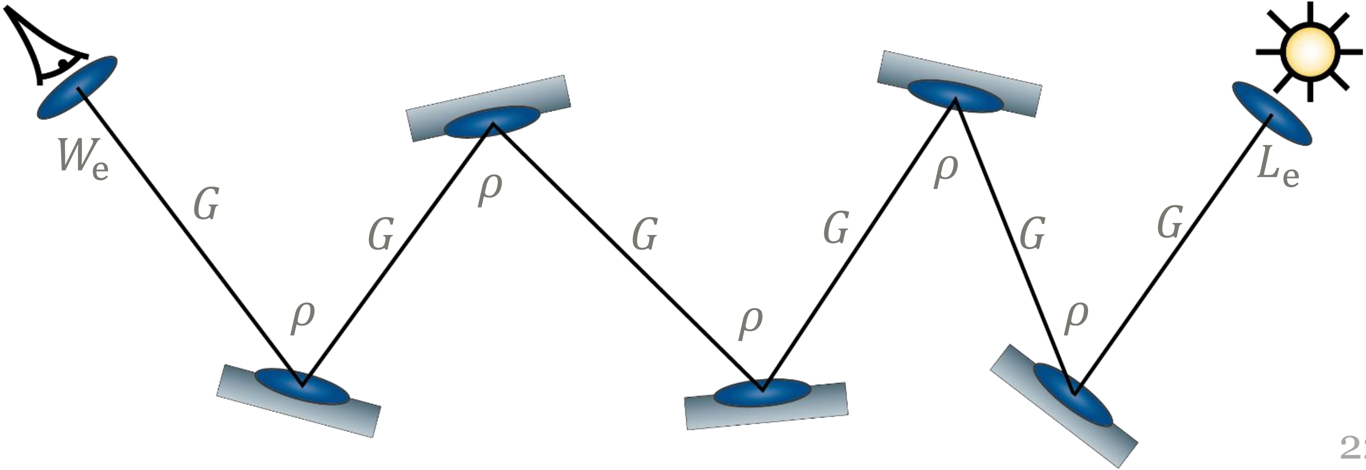
- State space = space of full paths, *path space*
- What is the function $f$ for light transport?
- Interested in flux arriving at image plane



# Measurement Contribution

- Measurement contribution $f$ for $k$-length path

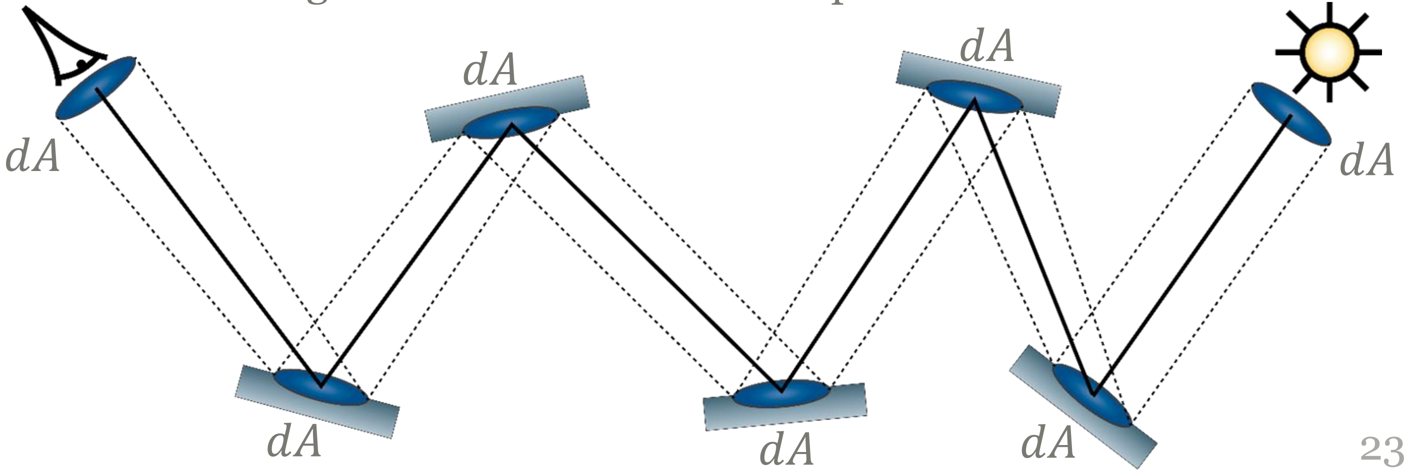
$$f(\bar{x}_k) = L_e G \left( \prod_{k-1} \rho_k G_k \right) W_e$$



# Measurement Contribution

$$f(\bar{x}_k) = \prod_k \frac{dQ}{dA_k} = \frac{dQ}{d\mu_k} \qquad [W/(m^2)^k]$$

- Flux through all differential areas of a path



# Comparing Paths

- MH needs to compare two states (paths)
  - Use flux through the infinitesimal path beam
- Directly comparable for equal-length paths
  - Compare flows of energy through each path
- For different lengths the measure is different
  - Always compare fluxes going through each path

# Path Integral

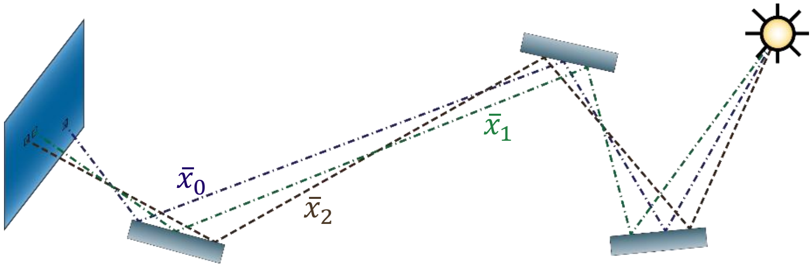
- For path of length $k$: $I_k = \int_{\Omega_k} f(\bar{x}) d\mu_k(\bar{x})$
- Combine all path lengths into a single integral
  - Use unified measure for all paths

$$d\mu(D) = \sum_{k=1}^{\infty} d\mu_k(D \cap \Omega_k)$$

  - Compare paths of different length
  - Compare groups of paths
- Use $f$ in Metropolis-Hastings!

# Metropolis Light Transport

1. Generate initial path $\bar{x}_0$ using PT/BDPT
2. Mutate with some proposal distribution $T_{\bar{x}_i \to \bar{x}_j}$
3. Accept new path $\bar{x}_j$ with probability $a_{\bar{x}_i \to \bar{x}_j}$
4. Accumulate contribution to the image plane
5. Go to step 2

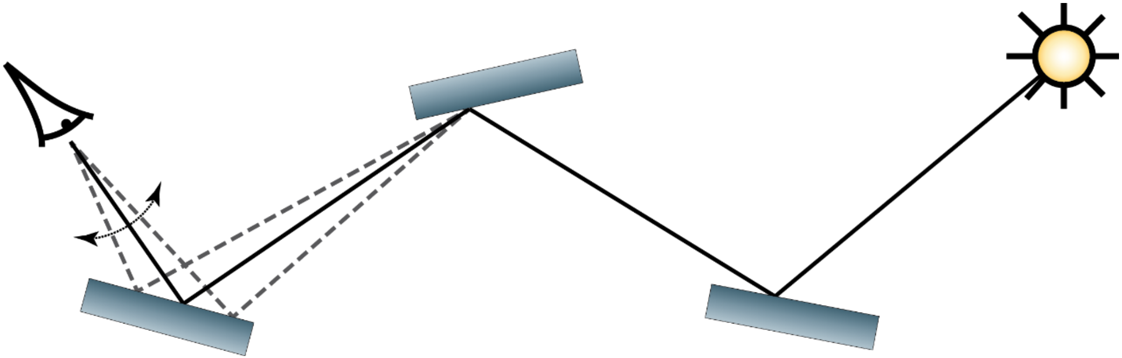


# Advantages

- More robust to complex light paths
  - Remembers successful paths
- Utilizes coherence of image pixels
  - Explores features faster
- Cheaper samples
  - Correlated
- Flexible path generators (mutations)

# Energy redistribution path tracing [Cline05]

- Run many short Markov chains for each seed
- Adaptive number of chains according to path energy
- In spirit of Veach's lens mutation



# Normalization and Start-up Bias in MLT

# Differences to MCMC

- We *do* have a good alternative sampler
  - Path tracer / bidirectional path tracer
  - Easy to compute normalization constant
- No start-up bias, start within the equilibrium
  - Start many chains stratified over path space
  - Scales well with massively parallel MLT

# Mutation Strategies and Their Properties

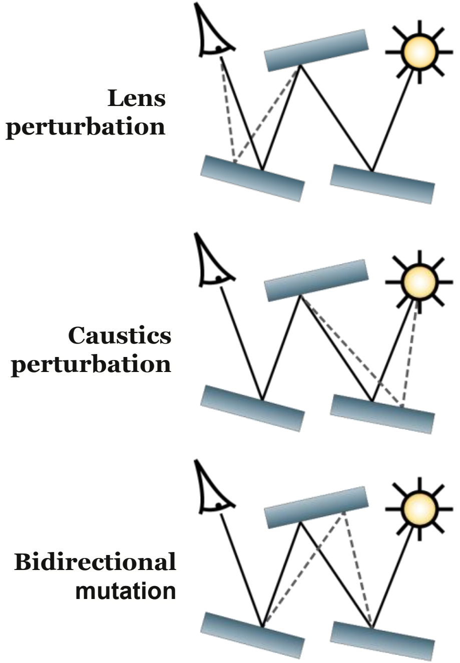
# Good Mutation Criteria

- Lightweight mutation: change a few vertices
- Low correlation of samples
  - Large steps in path space
- Good stratification over the image plane
  - Hard to control, usually done by re-seeding
- It’s OK to have many specialized mutations

# Existing Mutation Strategies

# Veach Mutations

- Minimal changes to the path
  - Lens, caustics, multi-chain perturbations
- Large changes to the path
  - Bidirectional mutation
    - BDPT-like large step
  - Lens mutation
    - stratified seeding on the image plane

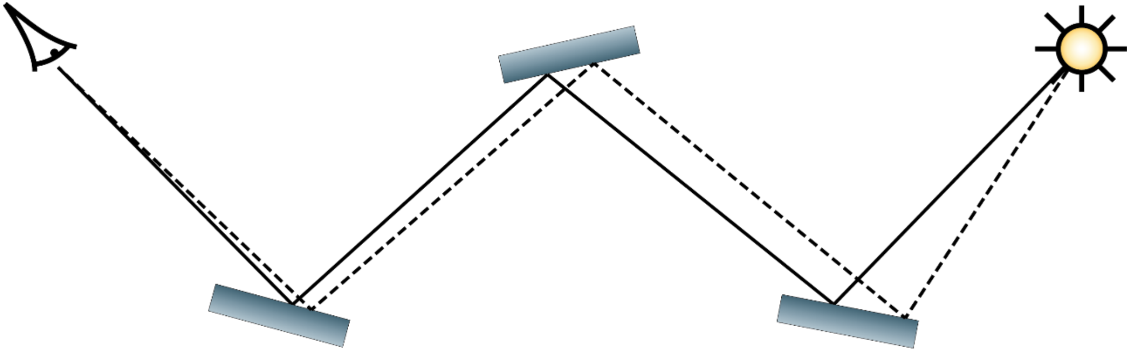


# Kelemen Mutation

- Mutate a "random" vector that maps to a path
- Symmetric perturbation of "random" numbers
- Use the "random" vector for importance pdfs
  - Primary space: importance function domain
  - Assume the importance sampling is good

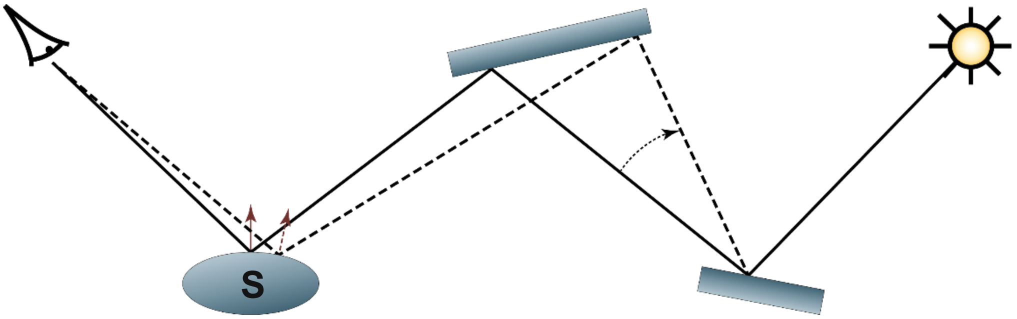
# Kelemen Mutation, Part II

- Acceptance probability $a_{i \to j} = (f_j / p_j) / (f_i / p_i)$
  - Easy to compute: just take values from PT/BDPT
- Large step: pure PT / BDPT step
  - Generate primary sample (random vector) anew



# Manifold Exploration Mutation

- Works in the local parameterization of current path
- Can connect through a specular chain
- Freezes integration dimensions
  - Tries to keep *f* constant by obeying constraints



# Combinations

- Manifold exploration can be combined
  - With Veach mutation strategies in MLT
  - With energy redistribution path tracing
- Combine Kelemen's and Veach's mutations?
  - Possible, yet unexplored option

# Population Monte Carlo Light Transport

# Population Monte Carlo Framework

- Use a *population* of Markov chains
  - Can operate on top of Metropolis-Hastings
- Rebalance the workload
  - Weakest chains are eliminated
  - Strongest chains are forked into multiple
- Use mixture of mutations, adapt to the data
  - Select optimal mutation on the fly

# Population Monte Carlo ERPT [Lai07]

- Spawn a population of chains with paths
  - Do elimination and reseeding based on path energy
- Use many mutations with different parameters
  - Reweight them on-the-fly based on the efficiency
  - Lens and caustics perturbations in the original paper
- We will show PMC with manifold exploration

Thank You for Your attention.

# Part one questions?