Metropolis Light Transport

15-468, 15-668, 15-868
Physics-based Rendering
Spring 2022, Lecture 17

http://graphics.cs.cmu.edu/courses/15-468

Course announcements

- Talk by Yifan Peng on AR/VR today.
- Take-home quiz 9-10 posted, due 4/26.
- Feedback for final project proposals posted.

Slide credits

Most of these slides were directly adapted from:

- Shuang Zhao (UC Irvine).

Today's Lecture

- Metropolis light transport (MLT)
  - A Markov Chain Monte Carlo (MCMC) framework implementing the Metropolis-Hastings method first proposed by Veach and Guibas in 1997
  - Capable of efficiently constructing "difficult" transport paths
  - Lots of ongoing research long this direction
- MLT is capable of solving both the rendering equation (RE) and the radiative transfer equation (RTE). We will focus on the former

Metropolis-Hastings Method

- A Markov-Chain Monte Carlo technique

- Given a non-negative function f, generate a chain of (correlated) samples X_1, X_2, X_3, … that follow a probability density proportional to f

- **Main advantage:** f does not have to be a PDF (i.e., unnormalized)

Metropolis-Hastings Method

- Input
  - Non-negative function *f*
  - Probability density $g(y \to x)$ suggesting a candidate for the next sample value *x*, given the previous sample value *y*

- The algorithm: given current sample X_i
  1. Sample X' from $g(X_i \to X')$
  2. Let $a = \frac{f(X')}{f(X_i)} \frac{g(X' \to X_i)}{g(X_i \to X')}$ and draw $\xi \sim U(0, 1]$
  3. If $\xi \leq a$, set X_{i+1} to X'; otherwise, set X_{i+1} to X_i
- Start with arbitrary initial state X_0

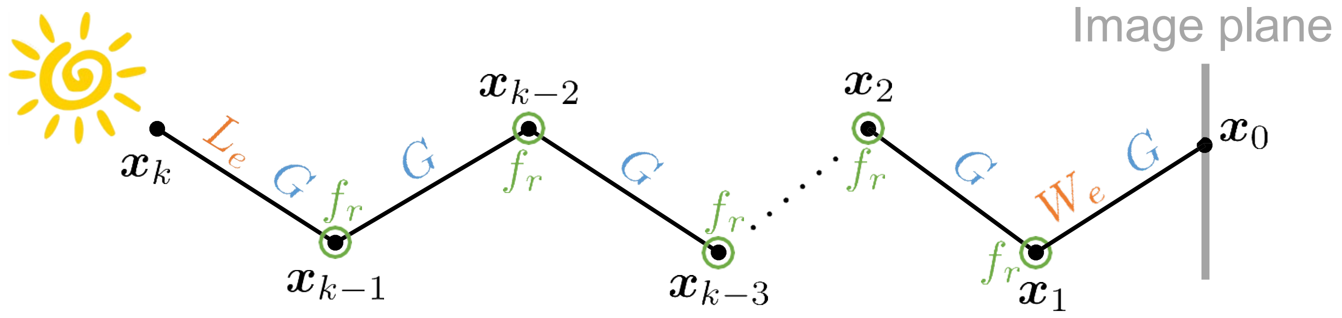
The Problem

- We focus on estimating the pixel values of a virtual image where intensity $I^{(j)}$ of pixel j is

$$I^{(j)} = \int_\Omega f^{(j)}(\bar{x})\,\mathrm{d}\bar{x} = \int_\Omega h^{(j)}(\bar{x})\,f(\bar{x})\,\mathrm{d}\bar{x}, \text{ where}$$

$$h^{(j)}(\bar{x}) = W_e^{(j)}(\boldsymbol{x}_1 \to \boldsymbol{x}_0)$$

$$f(\bar{x}) = L_e(\boldsymbol{x}_k \to \boldsymbol{x}_{k-1}) \left[\prod_{j=0}^{k-1} G(\boldsymbol{x}_{j+1} \leftrightarrow \boldsymbol{x}_j)\right] \left[\prod_{j=1}^{k-1} f_r(\boldsymbol{x}_{j+1} \to \boldsymbol{x}_j \to \boldsymbol{x}_{j-1})\right]$$



The Problem

- We focus on estimating the pixel values of a virtual image where intensity $I^{(j)}$ of pixel j is

$$I^{(j)} = \int_\Omega f^{(j)}(\bar{x})\,\mathrm{d}\bar{x} = \int_\Omega h^{(j)}(\bar{x})\,f(\bar{x})\,\mathrm{d}\bar{x}, \text{ where}$$

$$h^{(j)}(\bar{x}) = W_e^{(j)}(\boldsymbol{x}_1 \to \boldsymbol{x}_0)$$

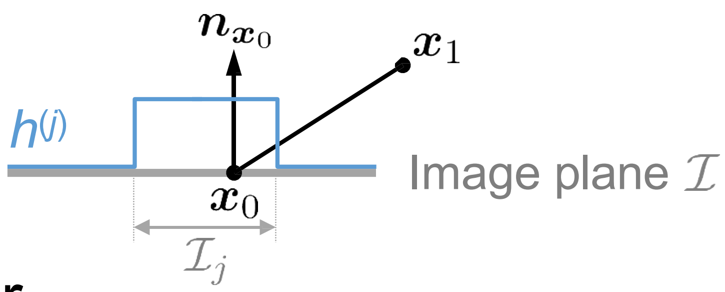
$$f(\bar{x}) = L_e(\boldsymbol{x}_k \to \boldsymbol{x}_{k-1}) \left[\prod_{j=0}^{k-1} G(\boldsymbol{x}_{j+1} \leftrightarrow \boldsymbol{x}_j)\right] \left[\prod_{j=1}^{k-1} f_r(\boldsymbol{x}_{j+1} \to \boldsymbol{x}_j \to \boldsymbol{x}_{j-1})\right]$$

- $h^{(j)}$ varies per pixel and is called the *filter function*
- f stays identical for all pixels

Example Filter Functions

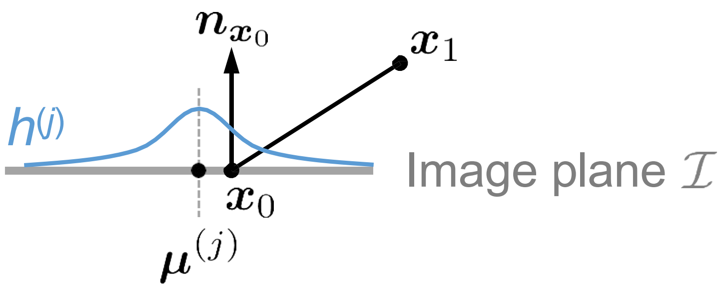
- Box Filter

$$h^{(j)}(\bar{x}) = \frac{\mathbb{1}[\boldsymbol{x}_0 \in \mathcal{I}_j]}{|\mathcal{I}_j|} \mathbb{1}[\langle \boldsymbol{x}_0 \to \boldsymbol{x}_1, \boldsymbol{n}_{\boldsymbol{x}_0} \rangle > 0]$$



- Gaussian Filter

$$h^{(j)}(\bar{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\boldsymbol{x}_0 - \boldsymbol{\mu}^{(j)}\|^2}{2\sigma^2} \right) \mathbb{1}[\langle \boldsymbol{x}_0 \to \boldsymbol{x}_1, \boldsymbol{n}_{\boldsymbol{x}_0} \rangle > 0]$$



Estimating Pixel Values

$$I^{(j)} = \int_\Omega h^{(j)}(\bar{x})\, f(\bar{x})\, \mathrm{d}\bar{x}$$

- We have seen that if we can draw *N* path samples $\bar{x}_1, \ldots, \bar{x}_N$ according to some density function *p*, then

$$I^{(j)} = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{h^{(j)}(\bar{x}_i)\, f(\bar{x}_i)}{p(\bar{x}_i)}\right]$$

- Particularly, if we take $p \propto f$, namely $p(\bar{x}) = f(\bar{x})/b$ with *b* being the normalization factor, then

$$I^{(j)} = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} b\, h^{(j)}(\bar{x}_i)\right]$$

Estimating Pixel Values

$$I^{(j)} = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} b\, h^{(j)}(\bar{x}_i)\right]$$

- Challenges
  - How to obtain $b = \int_{\Omega} f(\bar{x})\, \mathrm{d}\bar{x}$?

    Monte Carlo integration

  - How to draw samples from $p(\bar{x}) = f(\bar{x})/b$?

    Metropolis-Hastings method

$$I^{(j)} = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} b\, h^{(j)}(\bar{x}_i)\right]$$

Metropolis Light Transport (MLT)

- Overview
  - **Phase 1:** initialization (estimating *b*)
    - Draw *N'* "seed" paths $\bar{x}_1^{\text{seed}}, \ldots, \bar{x}_{N'}^{\text{seed}}$ from some known density p_0 (e.g., using bidirectional path tracing)
    - Set $\langle b \rangle = \frac{1}{N'} \sum_{i=1}^{N'} \frac{f(\bar{x}_i^{\text{seed}})}{p_0(\bar{x}_i^{\text{seed}})}$
    - Pick a small number (e.g., one) of representatives from $\bar{x}_1^{\text{seed}}, \ldots, \bar{x}_{N'}^{\text{seed}}$ and apply Phase 2 to each of them
  - **Phase 2:** Metropolis
    - Starting with a seed path, apply the Metropolis-Hastings method to generate samples according to *f*

Metropolis Phase

- Overview (pseudocode)

```
Metropolis_Phase(image, x^seed):
    x = x^seed
    for i = 1 to N:
        y = mutate(x)
        a = acceptanceProbability(x → y)
        if rand() < a:
            x = y
        recordSample(image, x)
```

Path Mutations

- The key step of the Metropolis phase

- Given a transport path $\bar{x}$, we need to define a transition probability $g(\bar{x} \to \bar{y})$ to allow sampling mutated paths $\bar{y}$ based on $\bar{x}$
  - Given this transition density, the acceptance probability is then given by

$$a(\bar{x} \to \bar{y}) = \min \left\{ 1, \frac{f(\bar{y})}{f(\bar{x})} \frac{g(\bar{y} \to \bar{x})}{g(\bar{x} \to \bar{y})} \right\}$$

Desirable Mutation Properties

- High acceptance probability
  - $a(\bar{x} \to \bar{y})$ should be large with high probability

- Large changes to the path

- Ergodicity (never stuck in some-region of the path space)
  - $g(\bar{x} \to \bar{y})$ should be non-zero for all $\bar{x},\ \bar{y}$ with $f(\bar{x}) > 0,\ f(\bar{y}) > 0$

- Low cost

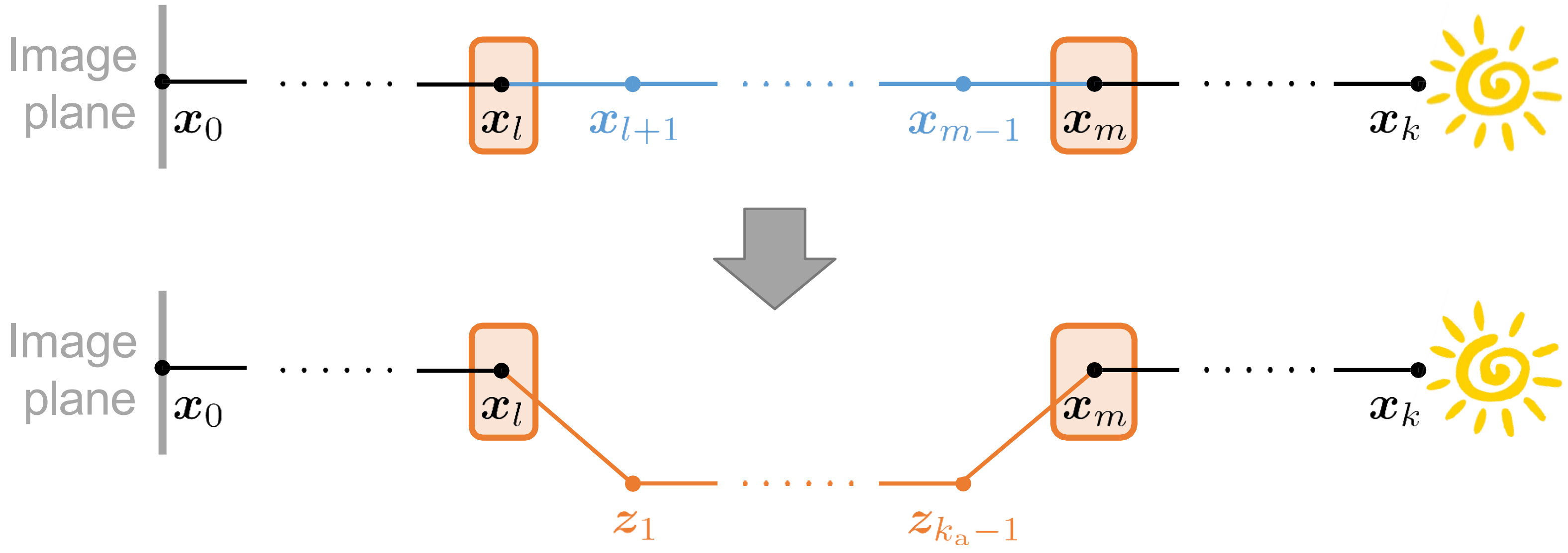
Path Mutation Strategies

- [Veach & Guibas 1997]
  - Bidirectional mutation
  - Path perturbations
  - Lens sub-path mutation
- [Jakob & Marschner 2012]
  - Manifold exploration
- [Li et al. 2015]
  - Hamiltonian Monte Carlo

…

Bidirectional Path Mutations

- Basic idea
  - Given a path $\bar{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_k)$, pick *l*, *m* and replace the vertices $\boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{m-1}$ with $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{k_a-1}$
    - *l* and *m* satisfies $-1 \leq l < m \leq k+1$

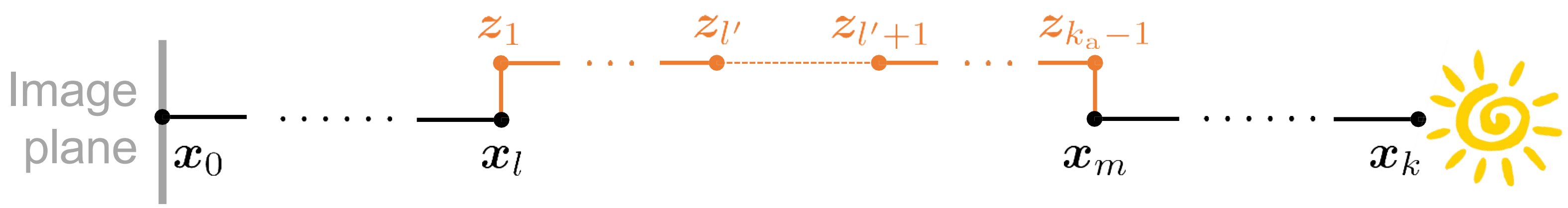


Deletion Probability

- l and m are sampled as follows:
  - Draw integer k_d from some probability mass function $p_{d,1}[k_d]$. This number captures the length of deleted sub-path (i.e., $m - l$)
  - Draw l from another probability mass function $p_{d,2}[l \mid k_d]$ to avoid low acceptance probability and set m to $l + k_d$ (more on this at the end of today's lecture)
  - The joint probability p_d for drawing (l, m) is

$$p_{\mathrm{d}}[l, m] = p_{\mathrm{d},1}[m - l]\, p_{\mathrm{d},2}[l \mid m - l]$$

Addition Probability



- The deleted sub-path is then replaced by adding *l'* and *m'* vertices on each side. To determine *l'* and *m'*:
    - Draw integer k_a from $p_a[k_a]$. This integer determines the new sub-path length (i.e., $k_a = l' + m' + 1$)
    - Draw *l'* uniformly from {0, 1, …, k_a - 1} and set *m'* to $k_a - 1 - l'$
    - Let $p_a[l', m']$ denote the joint probability for drawing (*l'*, *m'*)
- After obtaining *l'* and *m'*, the two corresponding sub-paths are generated via local path sampling, yielding the new path $\bar{y} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_l, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{k_a-1}, \boldsymbol{x}_m, \ldots, \boldsymbol{x}_k)$

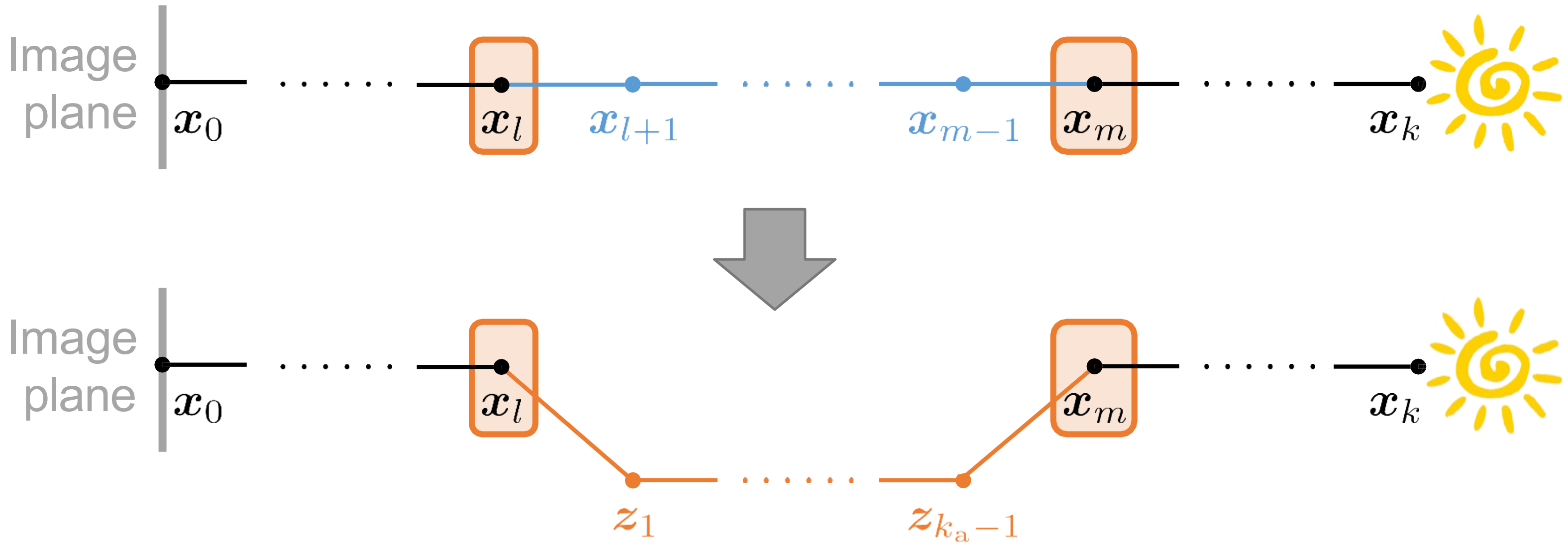
Parameter Values

- Veach [1997] proposed the following parameters:

- Deletion parameters
  - $p_{d,1}[1] = 0.25$, $p_{d,1}[2] = 0.5$, $p_{d,1}[k] = 2^{-k}$ for $k > 2$ (before normalization)
  - $p_{d,2}[l \mid k_d]$ to be discussed later

- Addition parameters (given k_d)
  - $p_{a,1}[k_d] = 0.5$, $p_{a,1}[k_d \pm 1] = 0.15$, $p_{a,1}[k_d \pm j] = 0.2(2^{-j})$ for $j > 2$ (before normalization)

Evaluating Transition Probability

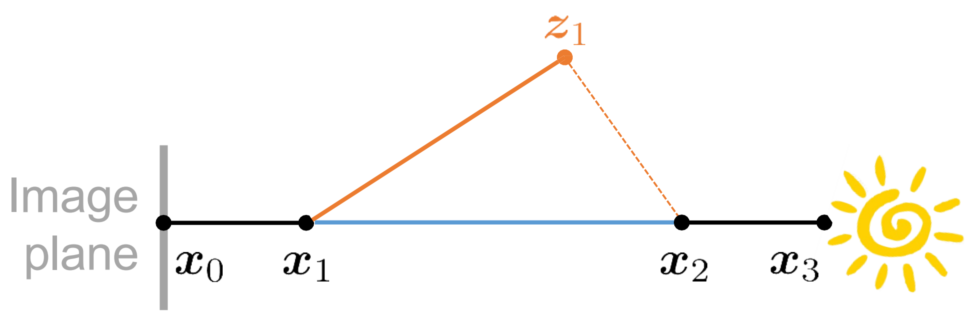


- The probability for transitioning from $\bar{x}$ to $\bar{y}$ is

$$g(\bar{x} \to \bar{y}) = p_d[l, m] \sum_{l'=0}^{k_a-1} p_a[l', k_a - 1 - l'] \, p((z_1, \ldots, z_{l'})) \, p((z_{k_a-1}, \ldots, z_{l'+1}))$$

$$= p_d[l, m] \sum_{l'=0}^{k_a-1} p_a[l', k_a - 1 - l'] \left(\prod_{i=1}^{l'} p_A(z_i)\right) \left(\prod_{i=k_a-1}^{l'+1} p_A(z_i)\right)$$

Bidirectional Mutation: Example

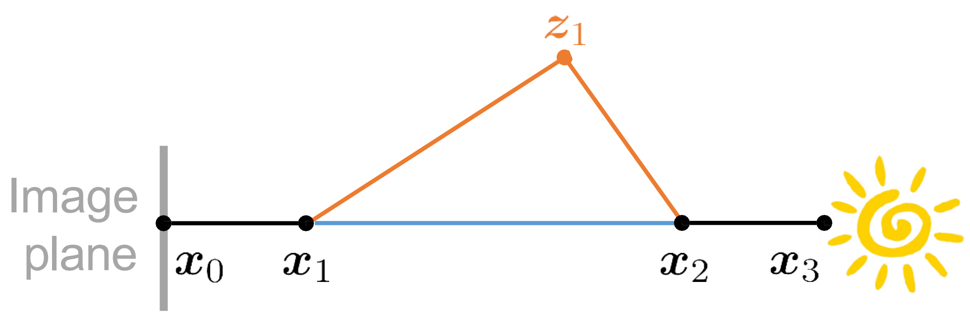


- Original path: $\bar{x} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$
- Mutation parameters:
  - *l* = 1, *m* = 2 (deletion); *l'* = 1, *m'* = 0 (addition)
- Mutated path: $\bar{y} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{z}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$
- The probability to accept $\bar{y}$ equals

$$a(\bar{x} \to \bar{y}) = \min\left\{1, \frac{f(\bar{y})}{f(\bar{x})} \frac{g(\bar{y} \to \bar{x})}{g(\bar{x} \to \bar{y})}\right\} = \min\left\{1, \frac{q(\bar{y} \to \bar{x})}{q(\bar{x} \to \bar{y})}\right\},$$

where $q(\bar{u} \to \bar{v}) := g(\bar{u} \to \bar{v}) / f(\bar{v})$

Bidirectional Mutation: Example



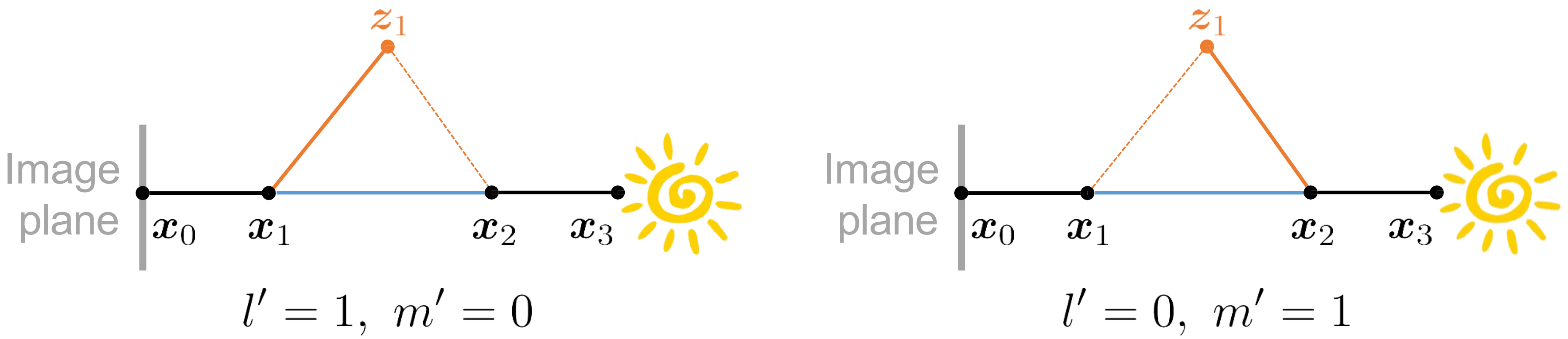
- $q(\bar{x} \to \bar{y}) := g(\bar{x} \to \bar{y})/f(\bar{y})$, where

$$f(\bar{y}) = G(\boldsymbol{x}_0 \leftrightarrow \boldsymbol{x}_1)\, f_s(\boldsymbol{z}_1 \to \boldsymbol{x}_1 \to \boldsymbol{x}_0)\, G(\boldsymbol{x}_1 \leftrightarrow \boldsymbol{z}_1)\, f_s(\boldsymbol{x}_2 \to \boldsymbol{z}_1 \to \boldsymbol{x}_1)$$
$$G(\boldsymbol{z}_1 \leftrightarrow \boldsymbol{x}_2)\, f_s(\boldsymbol{x}_3 \to \boldsymbol{x}_2 \to \boldsymbol{z}_1)\, G(\boldsymbol{x}_2 \leftrightarrow \boldsymbol{x}_3)\, L_e(\boldsymbol{x}_3 \to \boldsymbol{x}_2)$$

- Recall that $f(\bar{y})$ does not involve $W_e(\boldsymbol{x}_1 \to \boldsymbol{x}_0)$ as it is captured by the filter function $h^{(j)}$

Bidirectional Mutation: Example

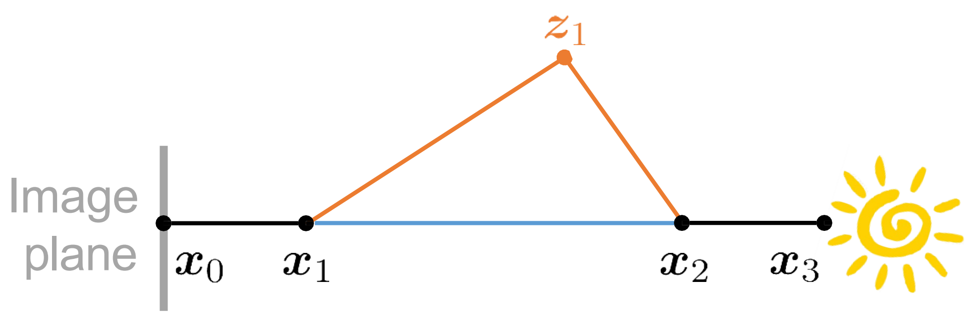
- $\bar{y}$ can be generated from $\bar{x}$ in two ways



- Thus,

$$\begin{aligned} g(\bar{x} \to \bar{y}) &= p_{\mathrm{d}}[1,2]\big(p_{\mathrm{a}}[1,0]\, p_A(\boldsymbol{z}_1 \mid \boldsymbol{x}_1) + p_{\mathrm{a}}[0,1]\, p_A(\boldsymbol{z}_1 \mid \boldsymbol{x}_2)\big) \\ &= p_{\mathrm{d}}[1,2]\big(p_{\mathrm{a}}[1,0]\, p_{\sigma^\perp}(\boldsymbol{x}_1 \to \boldsymbol{z}_1)\, G(\boldsymbol{x}_1 \leftrightarrow \boldsymbol{z}_1) + \\ &\qquad p_{\mathrm{a}}[0,1]\, p_{\sigma^\perp}(\boldsymbol{x}_2 \to \boldsymbol{z}_1)\, G(\boldsymbol{x}_2 \leftrightarrow \boldsymbol{z}_1)\big) \end{aligned}$$

Bidirectional Mutation: Example



- $q(\bar{y} \to \bar{x}) := g(\bar{y} \to \bar{x}) / f(\bar{x})$, where

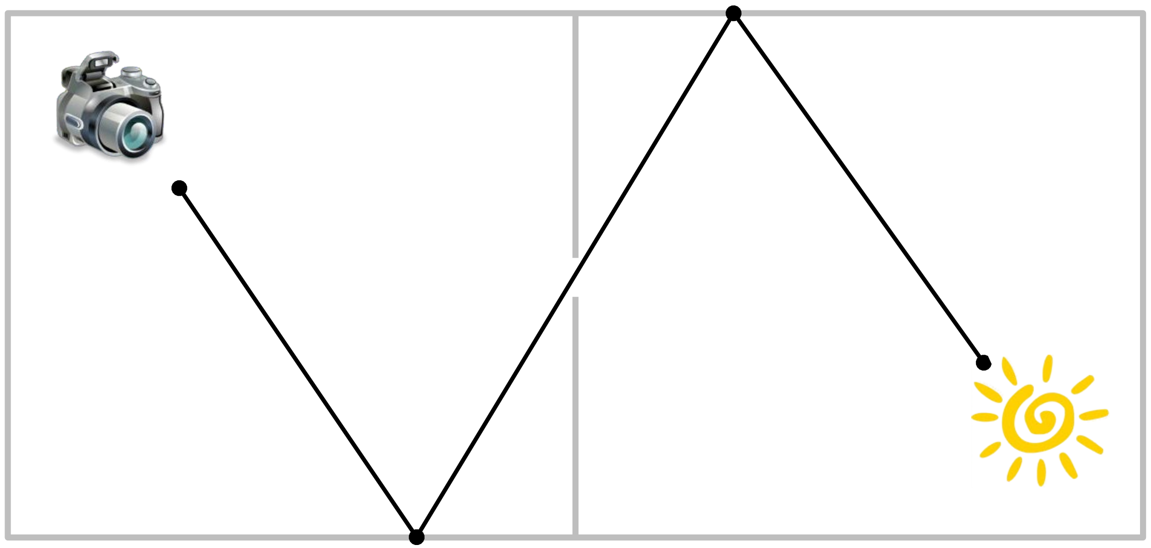
$$f(\bar{x}) = G(\boldsymbol{x}_0 \leftrightarrow \boldsymbol{x}_1)\, f_s(\boldsymbol{x}_2 \to \boldsymbol{x}_1 \to \boldsymbol{x}_0)\, G(\boldsymbol{x}_1 \leftrightarrow \boldsymbol{x}_2)\, f_s(\boldsymbol{x}_3 \to \boldsymbol{x}_2 \to \boldsymbol{x}_1)$$
$$G(\boldsymbol{x}_2 \leftrightarrow \boldsymbol{x}_3)\, L_e(\boldsymbol{x}_3 \to \boldsymbol{x}_2)$$

- To obtain $\bar{x}$ from $\bar{y}$ using bidirectional path mutation, we need *l* = 1, *m* = 3 and *l'* = *m'* = 0. Thus,

$$g(\bar{y} \to \bar{x}) = p_{\mathrm{d}}[1, 3]\, p_{\mathrm{a}}[0, 0]$$

Path Perturbations

- “Smaller” mutations
- Useful for finding “nearby” paths with high contribution



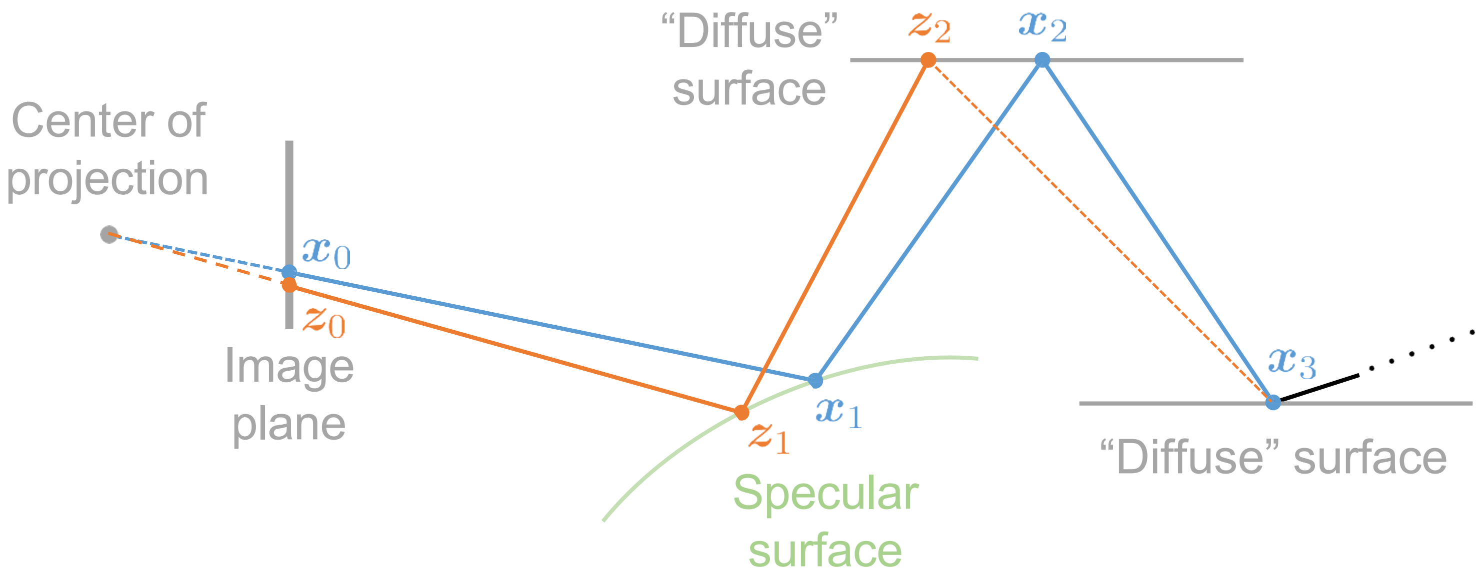
Path Perturbations

- Basic idea: choosing a sub-path and moving the vertices slightly

- Three types of perturbations
  - Lens
  - Caustic
  - Multi-chain

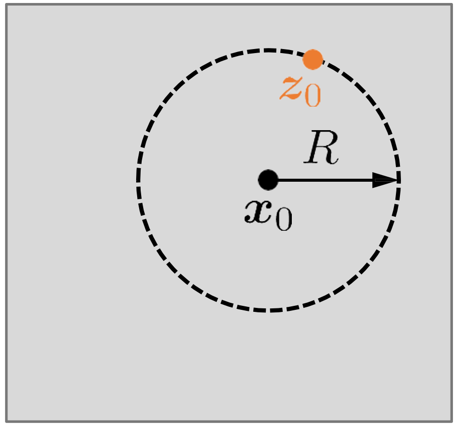
Path Perturbation: Lens

- Replace sub-paths ($\boldsymbol{x}_0, \ldots, \boldsymbol{x}_m$) of the form ES*D(D|L)
- Randomly move the endpoint $\boldsymbol{x}_0$ on the image plane to $\boldsymbol{z}_0$
- Trace a ray through $\boldsymbol{z}_0$ to form the new sub-path



Path Perturbation: Lens

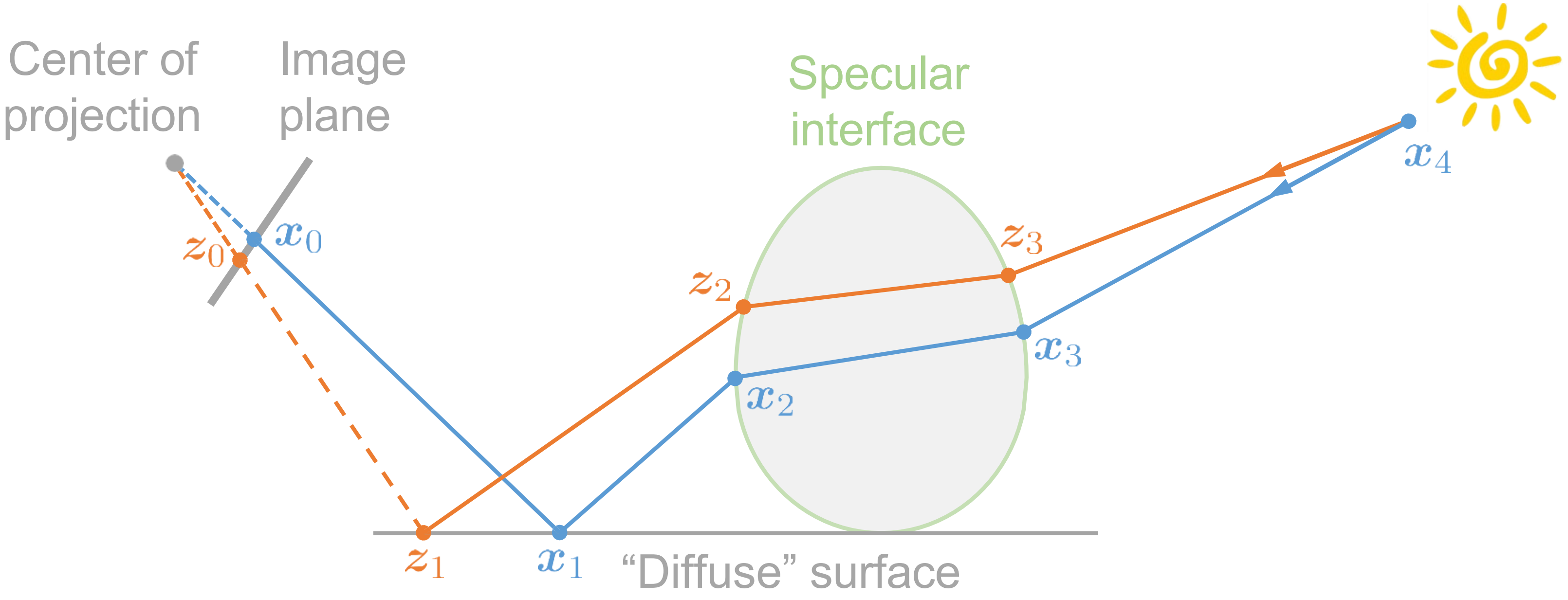
- To draw z_0:
  - First, sample a distance R using
    $R = r_2 \exp(-\log(r_2/r_1)\,\xi)$ where $\xi \sim U(0,1)$
  - Then, uniformly sample z_0 from the circle which is center at x_0 and has radius R



- The mutation is immediately rejected if ray tracing through z_0 fails to generate a new sub-path with exactly the same form (i.e., ES*D(D|L))
- Otherwise, the acceptance probability is evaluated in a way similar to the bidirectional mutation case

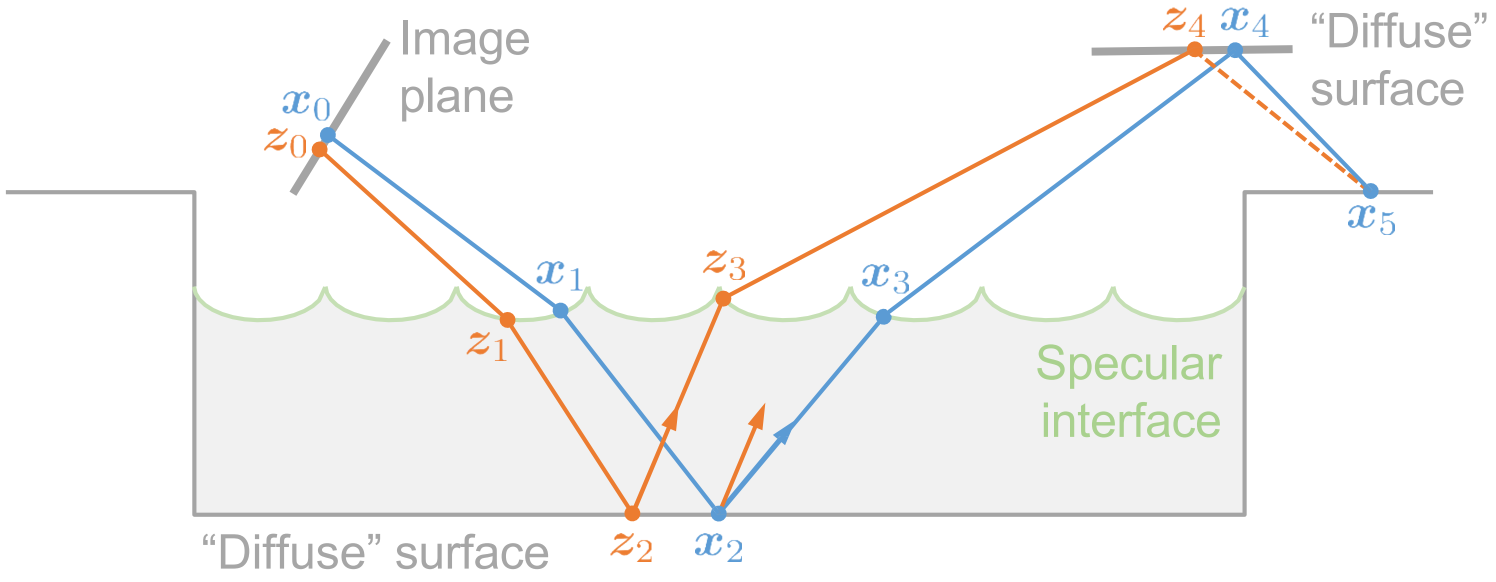
Path Perturbation: Caustic

- Replace sub-paths ($\boldsymbol{x}_0, \ldots, \boldsymbol{x}_m$) of the form EDS*(D|L)
- Slightly modify the direction $\boldsymbol{x}_m \rightarrow \boldsymbol{x}_{m\text{-}1}$ (at random)
- Trace a ray from $\boldsymbol{x}_m$ with this new direction to form the new sub-path



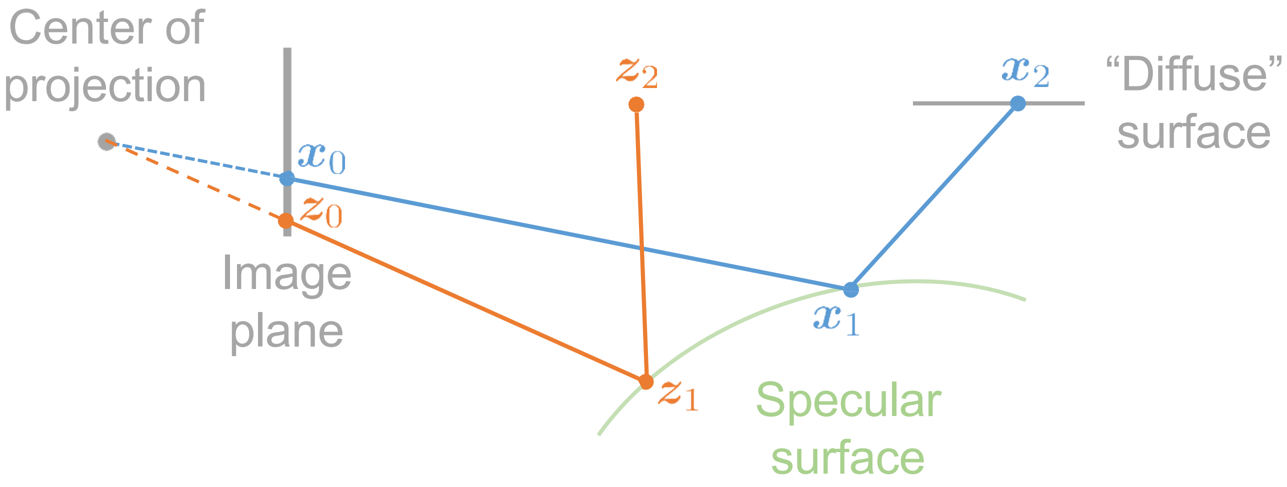
Path Perturbation: Multi-Chain

- Replace sub-paths of the form $ES^+DS^+D(D|L)$
- Lens perturbation is applied for ES^+D
- The direction of the DS^+ edge in the original sub-path is perturbed
- The new direction is then used to complete the $DS^+D(D|L)$ segment of the new sub-path (using ray tracing)



Lens Sub-Path Mutation

- Used to stratify samples over the image plane
  - Each pixel should get enough sample paths
- Replace lens sub-paths of the form ES*(D|L)
  - Similar to lens perturbation, but draw z_0 from a different density



Selecting Between Mutation Types

- Path mutations strategies introduced so far:
  - Bidirectional mutation
  - Lens, caustic, multi-chain perturbations
  - Lens sub-path mutation

- Choose one randomly in each iteration

Refinements

- Direct lighting
  - It is more efficient to estimate direct illumination with standard methods (e.g., area & BSDF sampling combined using MIS) and apply MLT only for indirect illumination

- Importance sampling for mutation probabilities
  - For increasing the average acceptance probability $a(\bar{x} \to \bar{y})$

Improving Acceptance Rates

- Recall:

$$a(\bar{x} \to \bar{y}) = \min\left\{1,\ \frac{q(\bar{y}\to\bar{x})}{q(\bar{x}\to\bar{y})}\right\},\ \text{where } q(\bar{u} \to \bar{v}) := \frac{g(\bar{u}\to\bar{v})}{f(\bar{v})}$$

- **Observation:** given a path $\bar{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_k)$ and $-1 \leq l < m \leq k+1$, $q(\bar{y} \to \bar{x}) = g(\bar{y} \to \bar{x})/f(\bar{x})$ can be partially evaluated without constructing $\bar{y}$
- $f(\bar{x})$ can be fully evaluated
- $g(\bar{y} \to \bar{x})$ can be partially evaluated

Improving Acceptance Rates

- Let $k_a = m - l - 1$, then

$$g(\bar{y} \rightarrow \bar{x}) = p_{\mathrm{d}} \sum_{l'=0}^{k_{\mathrm{a}}-1} p_{\mathrm{a}} \, p((\boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{l+l'})) \, p((\boldsymbol{x}_{m-1}, \ldots, \boldsymbol{x}_{l+l'+1}))$$

Unknown Known

- Set the unknown term to one and get a weight $w_{l,m}$ for each mutation

$$w_{l,m}(\bar{x}) = \frac{\sum_{l'=0}^{k_{\mathrm{a}}-1} p((\boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{l+l'})) \, p((\boldsymbol{x}_{m-1}, \ldots, \boldsymbol{x}_{l+l'+1}))}{f(\bar{x})}$$

Improving Acceptance Rates

$$w_{l,m}(\bar{x}) = \frac{\sum_{l'=0}^{k_\mathrm{a}-1} p((\boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{l+l'}))\, p((\boldsymbol{x}_{m-1}, \ldots, \boldsymbol{x}_{l+l'+1}))}{f(\bar{x})}$$

- Given a path $\bar{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_k)$, we can evaluate the weights for several possible mutation strategies and use these weights to sample one
- Can be used to obtain $p_{\mathrm{d},2}$ for bidirectional mutations
  - Given k_d, simply make $p_{\mathrm{d},2}[l \mid k_\mathrm{d}] \propto w_{l,l+k_\mathrm{d}}(\bar{x})$

Results

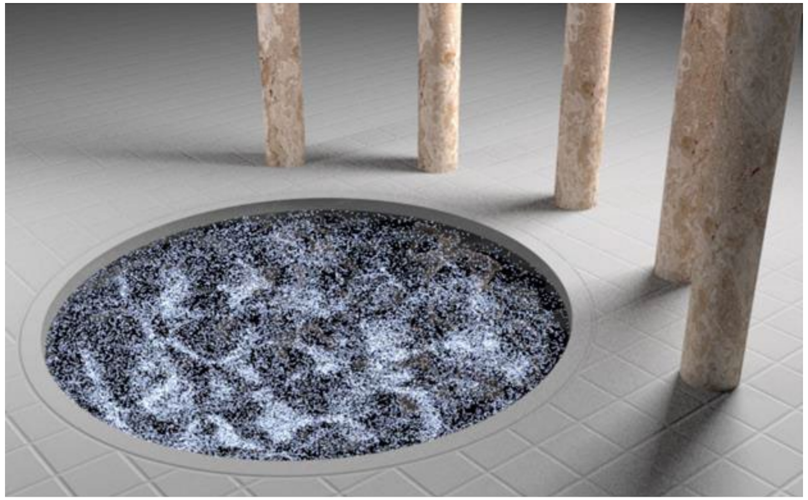
[Veach 1997]

BDPT

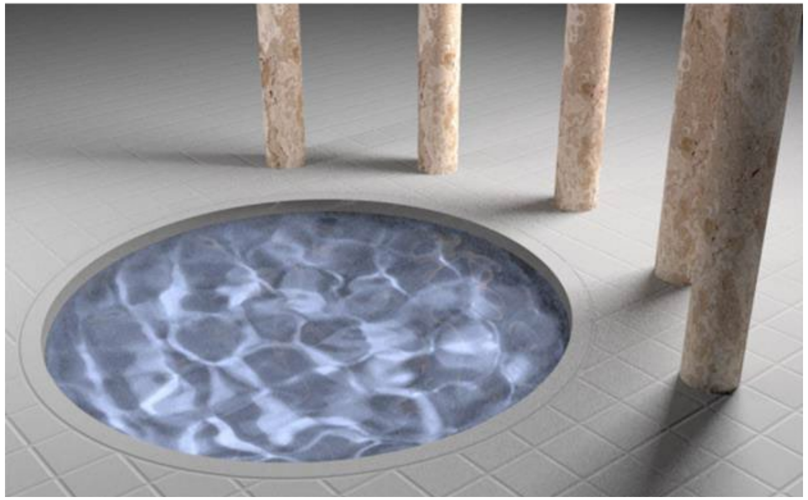
MLT
(equal-time)

Results

[Veach 1997]



BDPT



MLT
(equal-time)