

CS 87/187, Spring 2016

# RENDERING ALGORITHMS

## Markov Chain Monte Carlo



Prof. Wojciech Jarosz

[wojciech.k.jarosz@dartmouth.edu](mailto:wojciech.k.jarosz@dartmouth.edu)

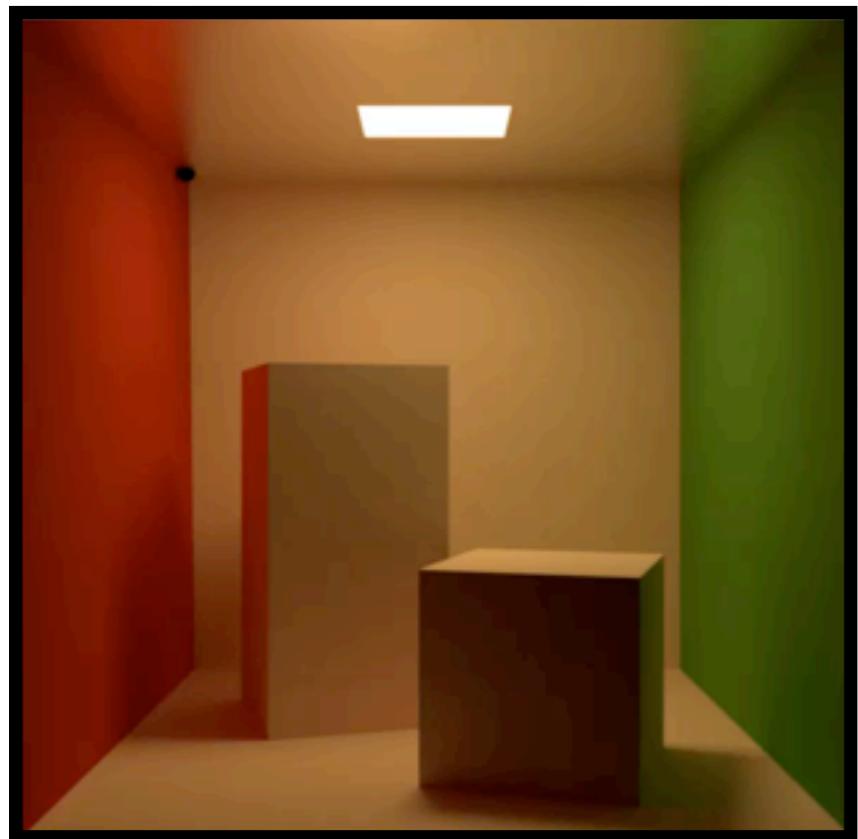
(most slides courtesy of Dr. Wenzel Jakob)

# Today's menu

---

1. Review of surface light transport & path tracing
2. “Metropolized” path tracing
3. Path space integration
4. Path space MCMC methods
5. Conclusion

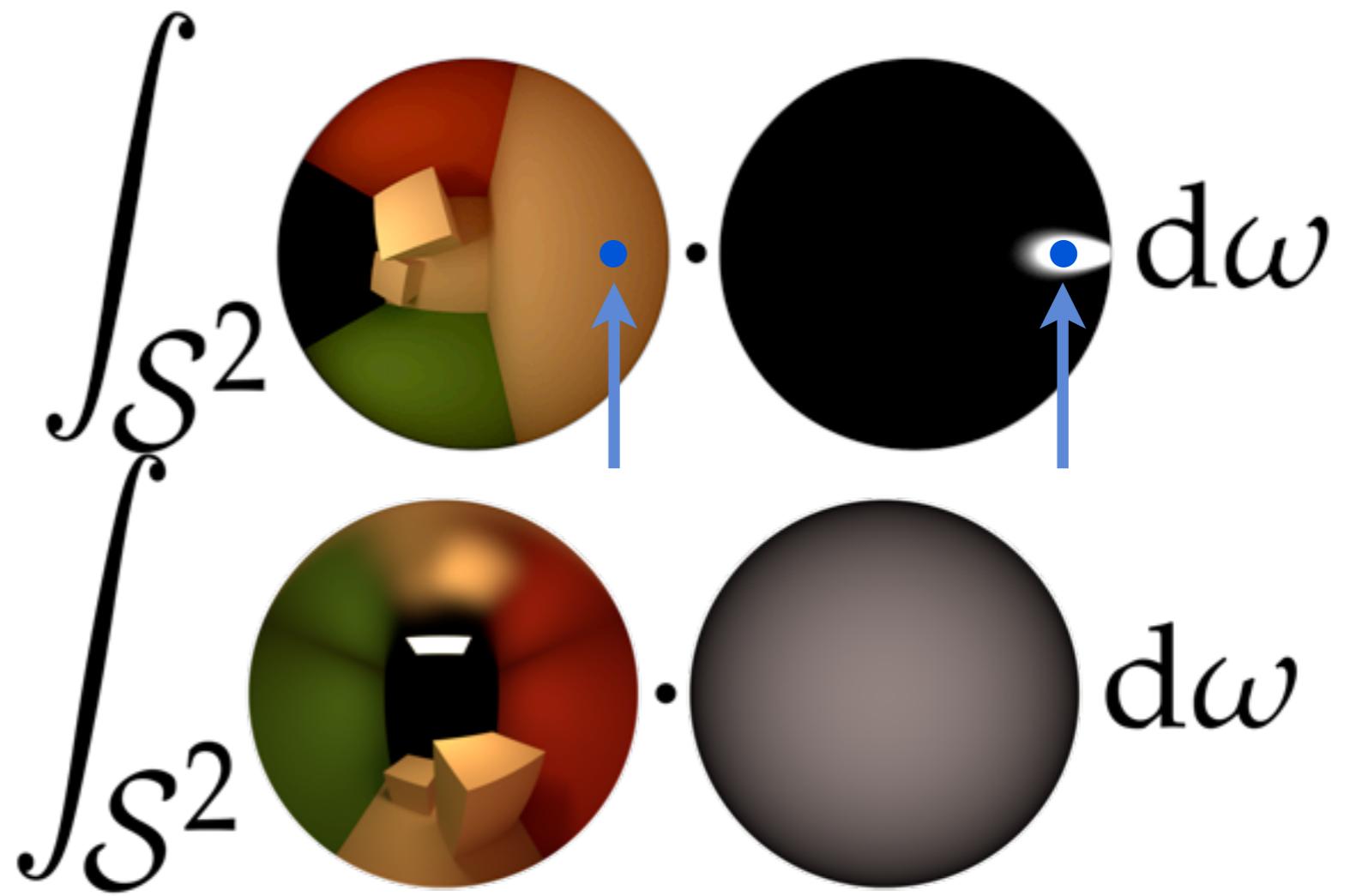
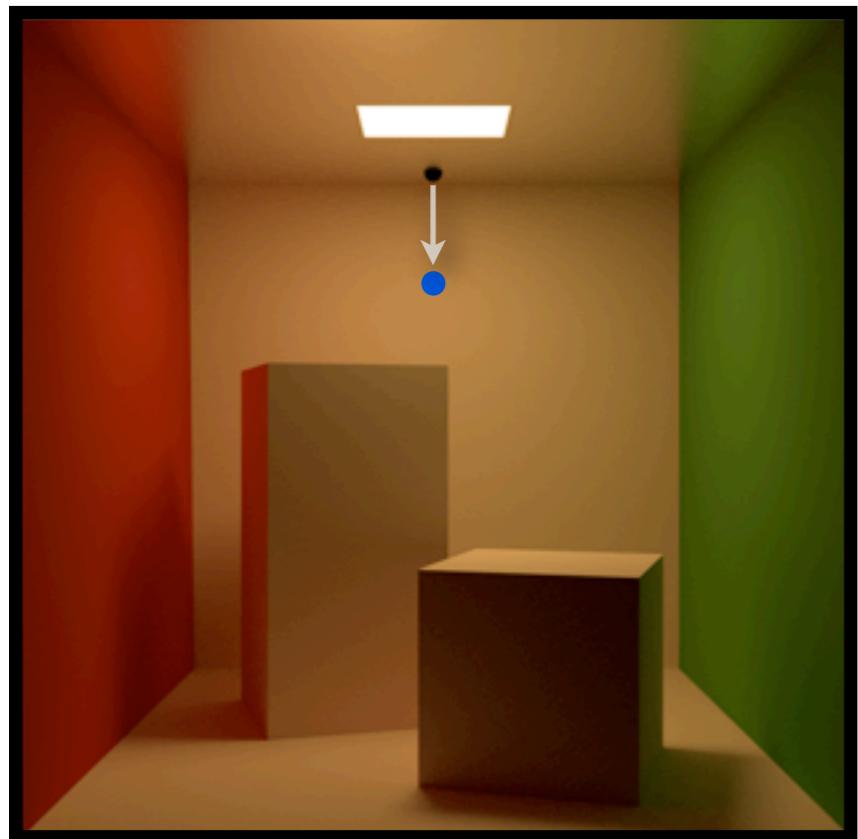
# Path tracing review



$$\int_{S^2} \cdot \cdot d\omega = \int_{S^2} \cdot d\omega = \boxed{\text{Final pixel color}}$$

The diagram illustrates the path tracing equation. On the left, a 3D scene is shown with a red wall, a green wall, and a brown floor. A small brown cube sits on the floor. The lighting is soft, suggesting a global illumination path tracer. To the right, the path tracing equation is shown. It consists of two parts. The top part shows a hemisphere divided into four quadrants: red (top), black (bottom), orange (left), and green (right). A small yellow cube is visible at the center. A dot product operation is performed between the hemisphere and a black circle representing a light source. The result is labeled  $d\omega$ . The bottom part shows a large black circle with a small orange dot representing the light source. This is followed by another dot product operation with a black circle, resulting in the final pixel color, represented by a red square with a black border.

# Path tracing review

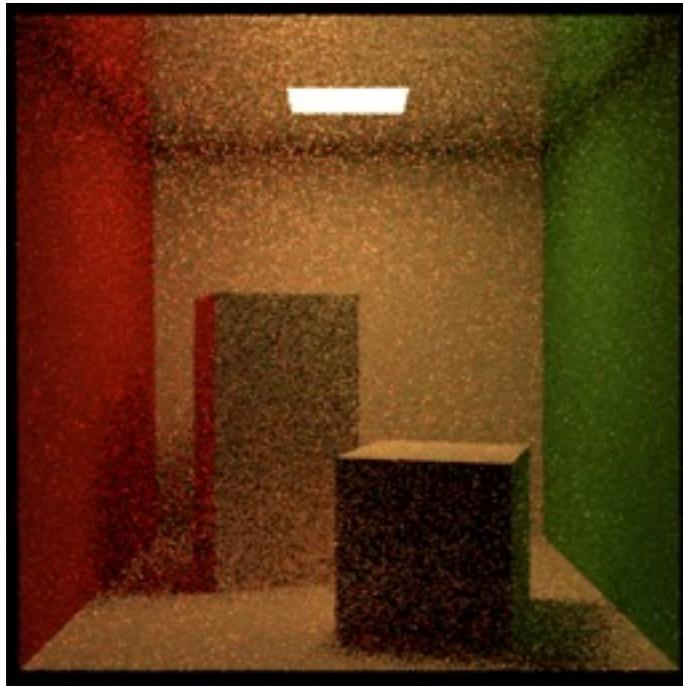


# Path tracing pseudocode

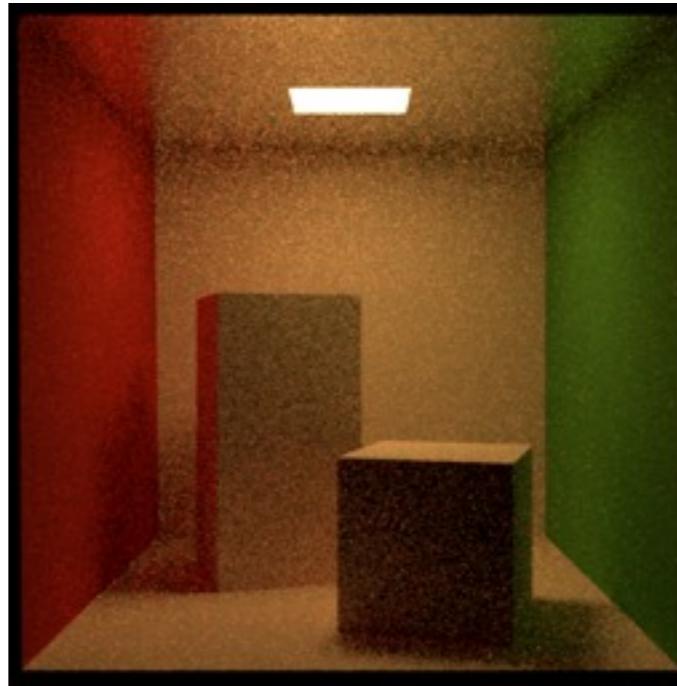
$\langle L_o(\mathbf{x}, \omega_o) \rangle :=$

1. Return 0 with probability  $\alpha$
2. Sample a direction  $\omega_i$  proportional to  $f_s(\mathbf{x}, \cdot, \omega_o)$
3. Set  $\mathbf{x}' := r(\mathbf{x}, \omega_i)$
4. Return  $\frac{1}{1 - \alpha} \left( L_e(\mathbf{x}, \omega_o) + P_{f_s} \langle L_o(\mathbf{x}', -\omega_i) \rangle \right)$

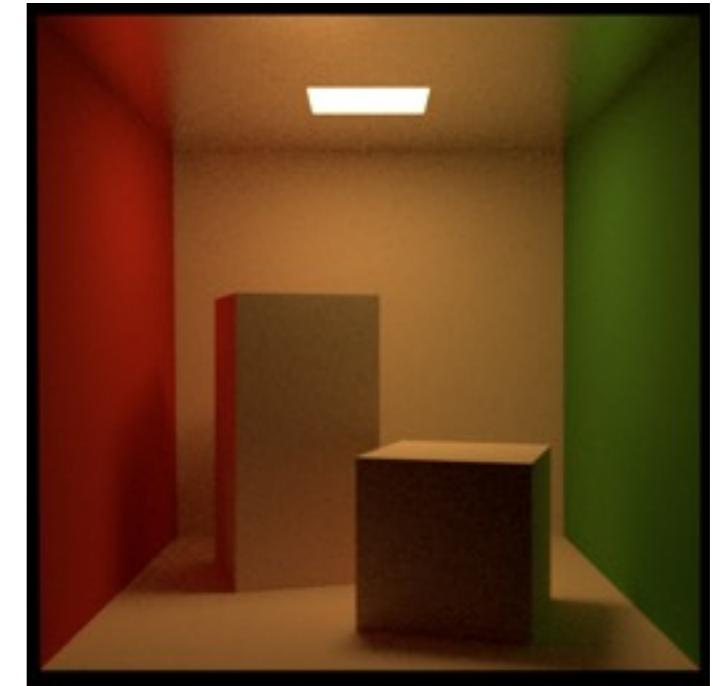
# Path tracing convergence



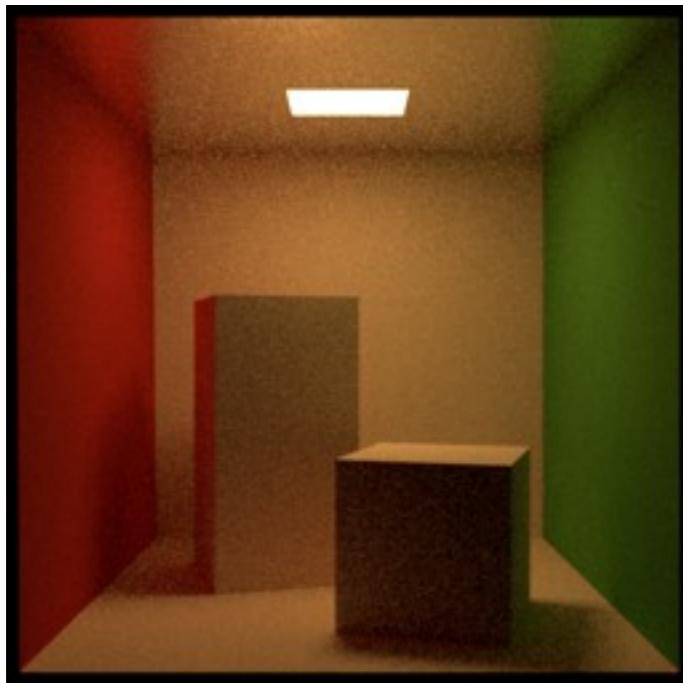
1 sample



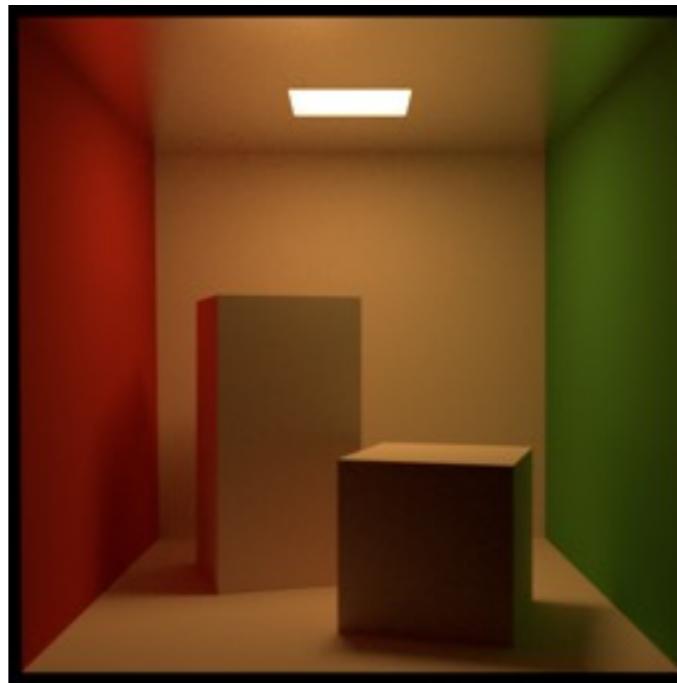
4 samples



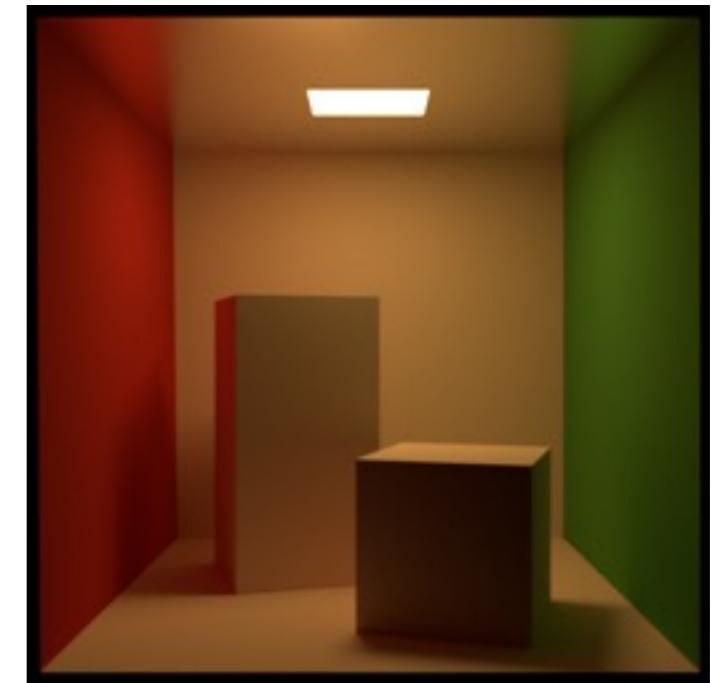
16 samples



64 samples



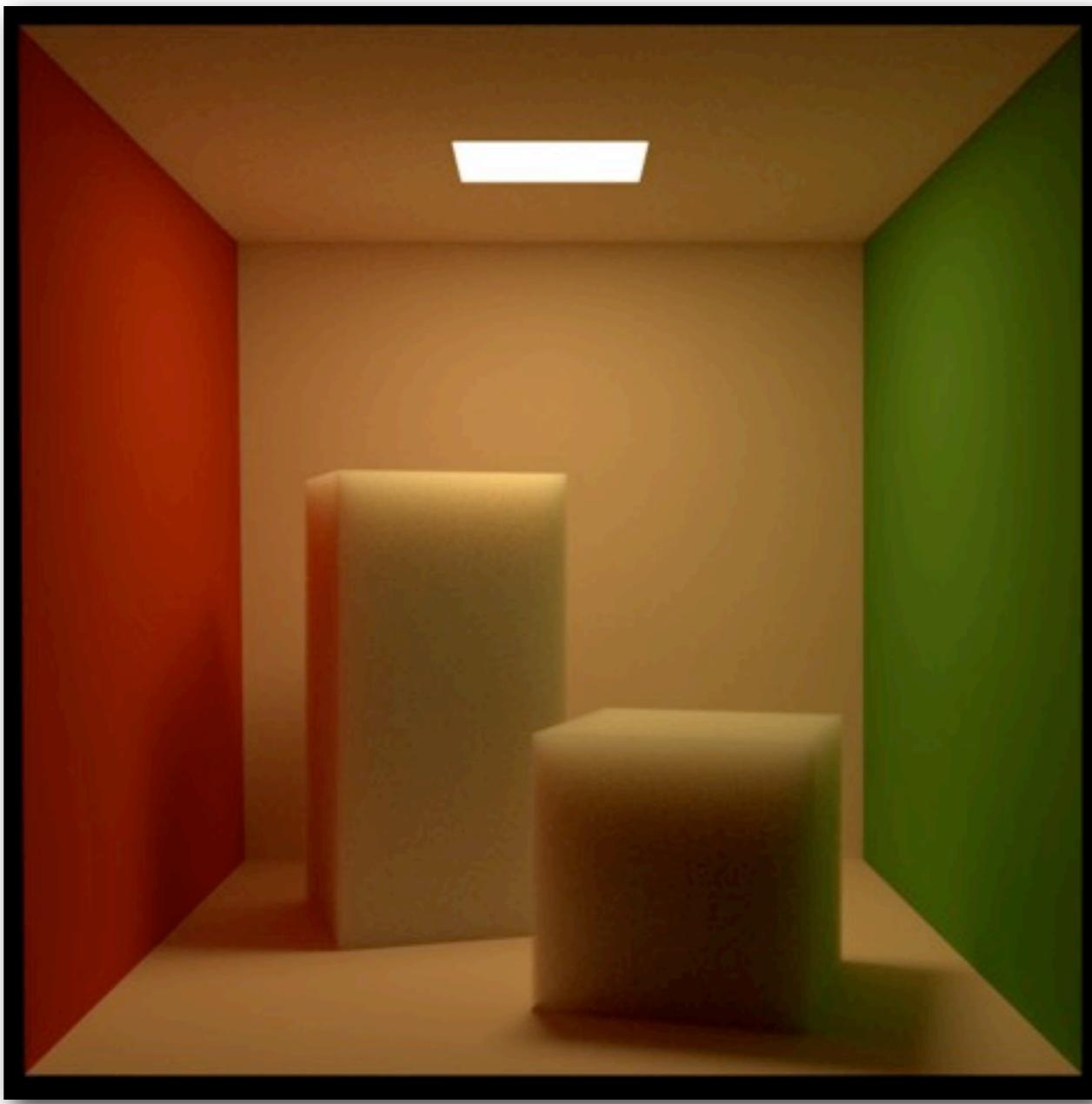
256 samples



1024 samples

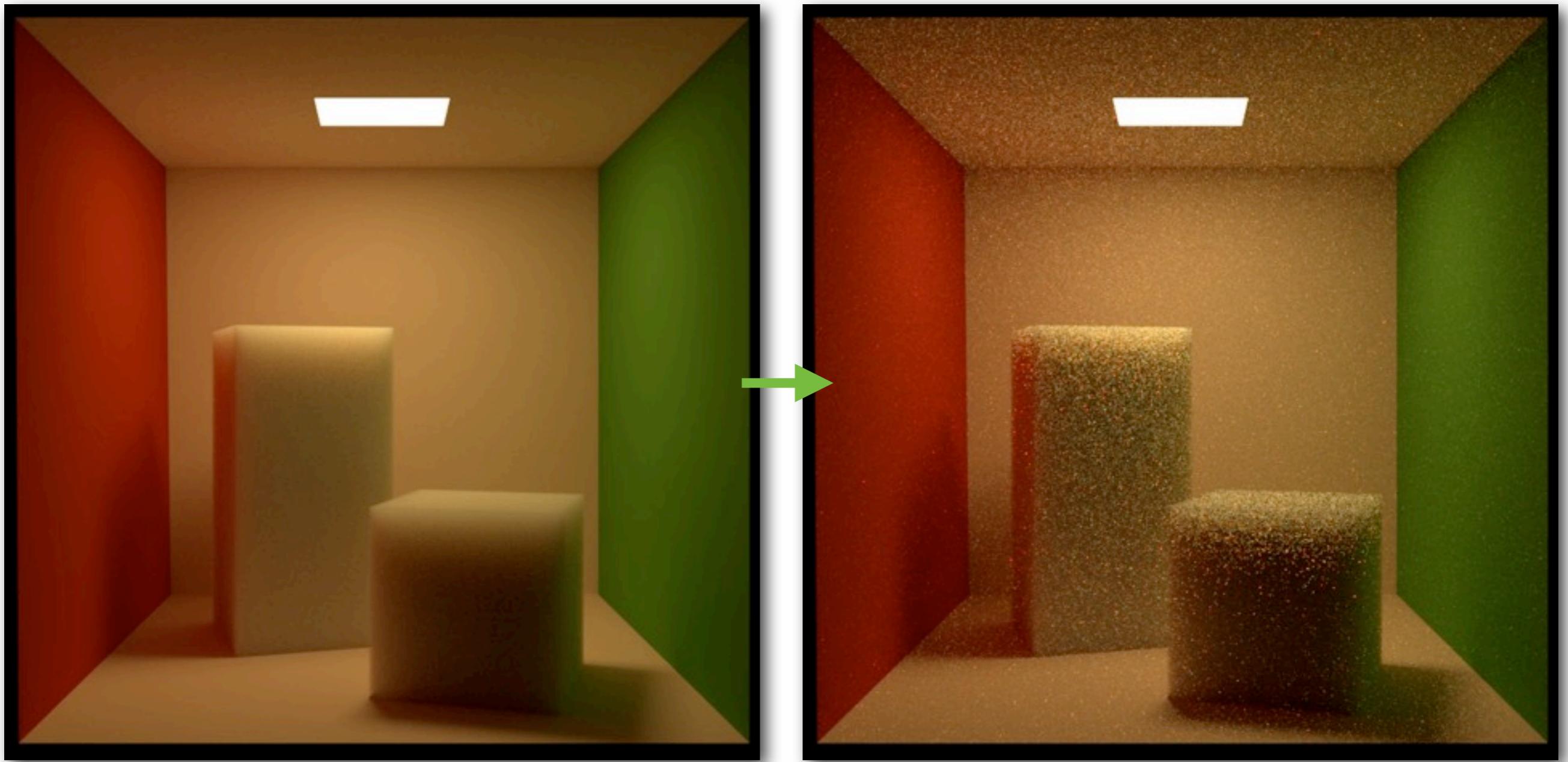
# A simple example

---



Cornell box with 2 random media, 256 spp.

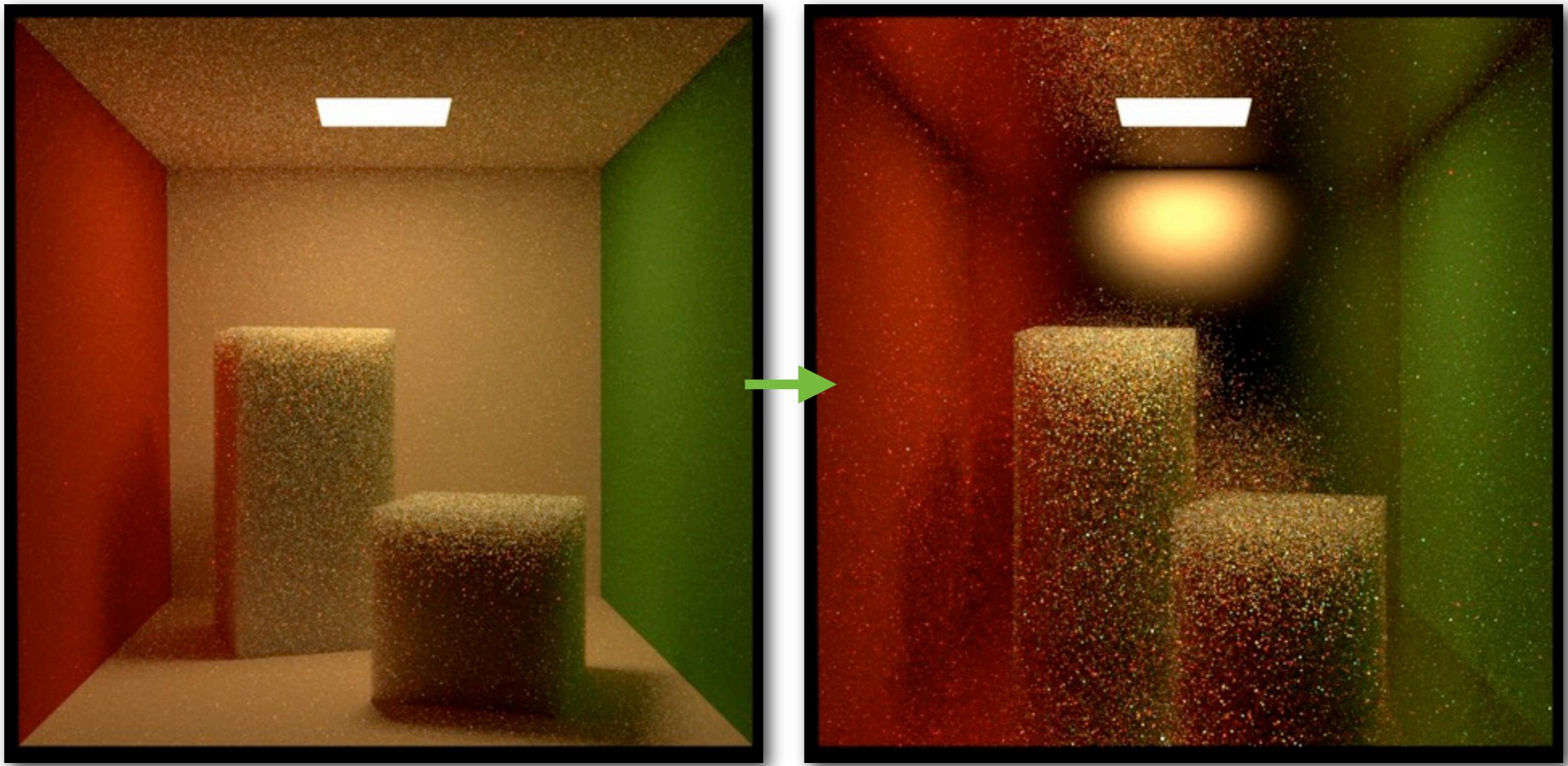
# + dielectric boundaries



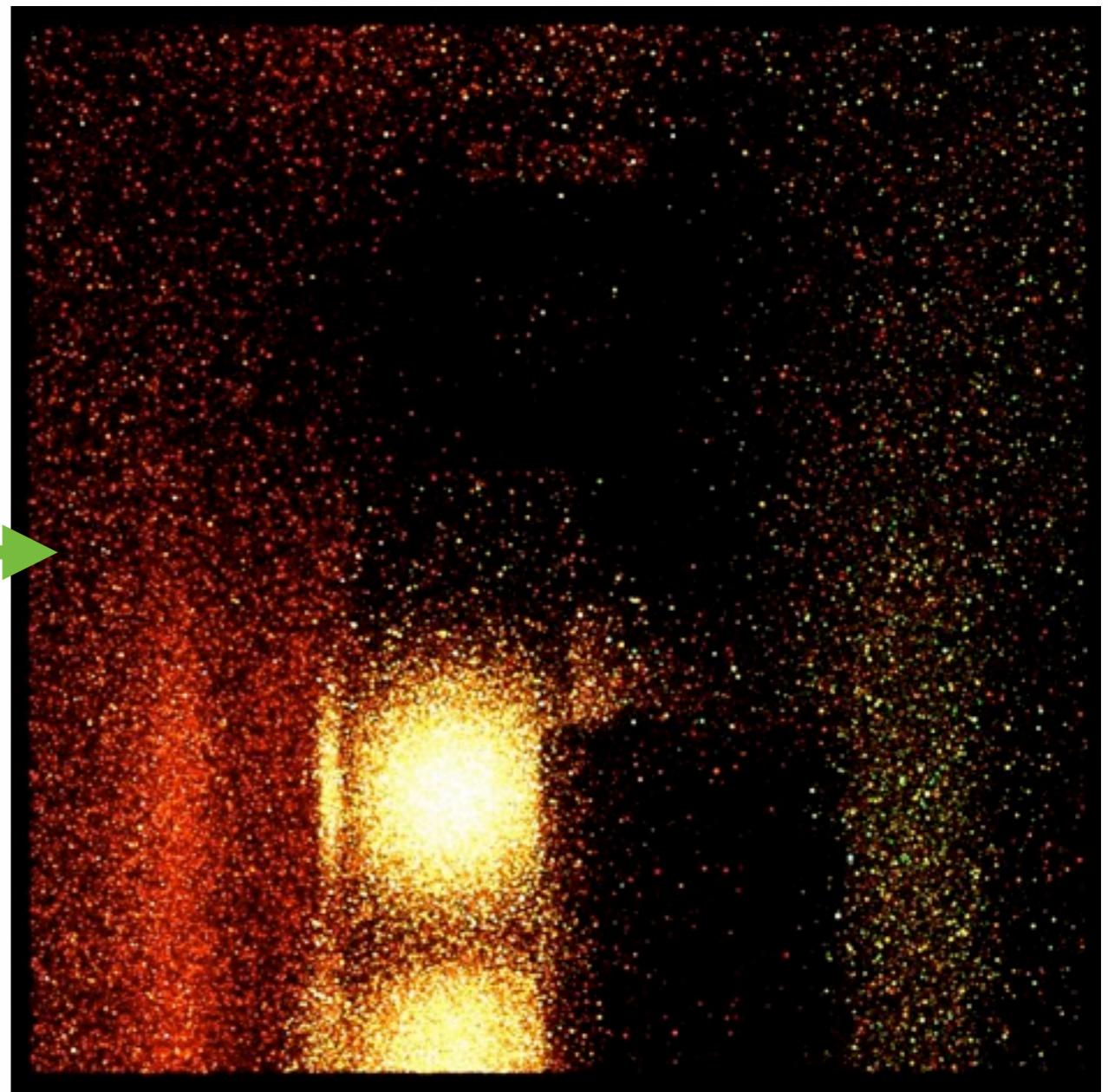
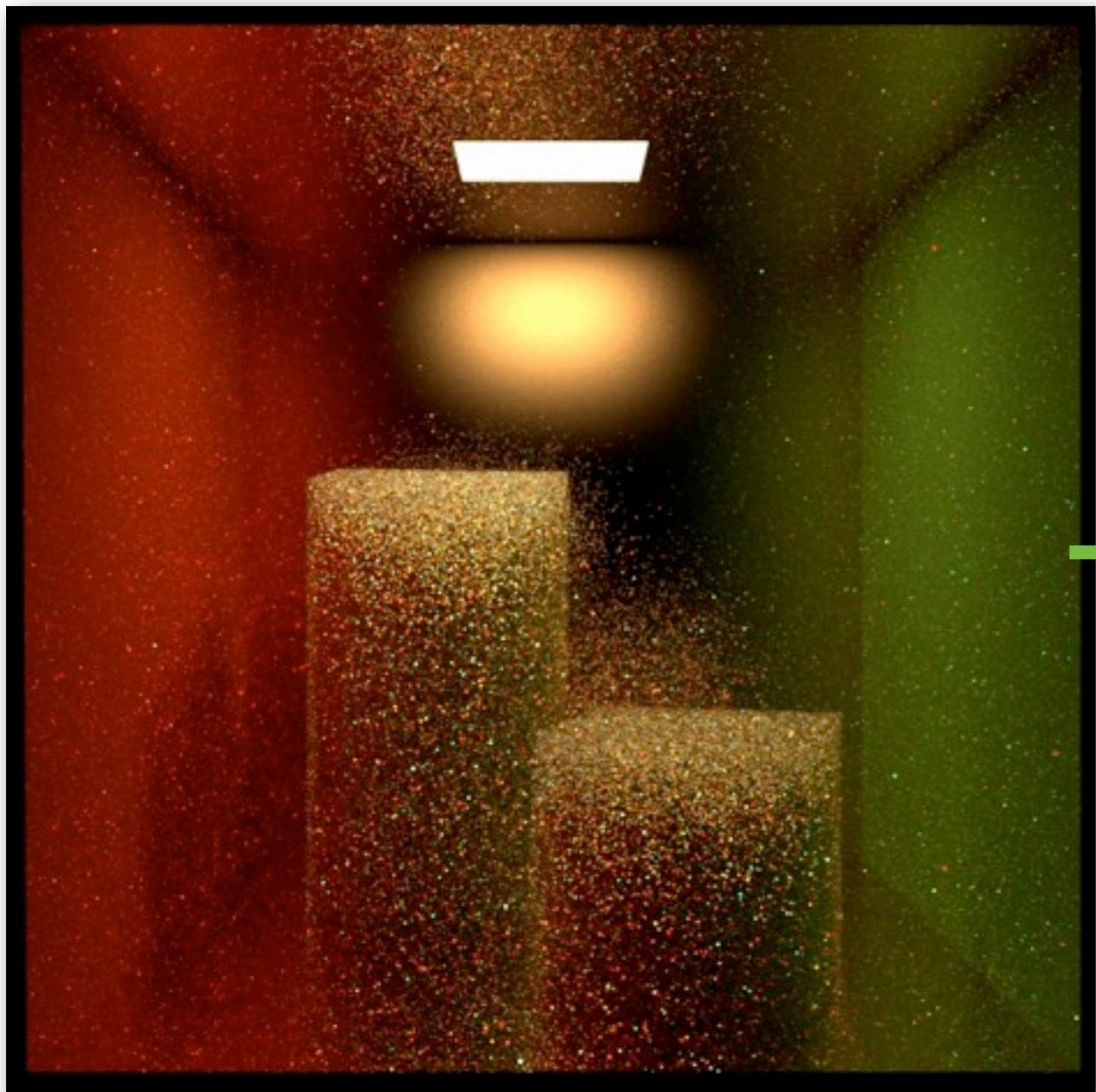
$$\eta = 1$$

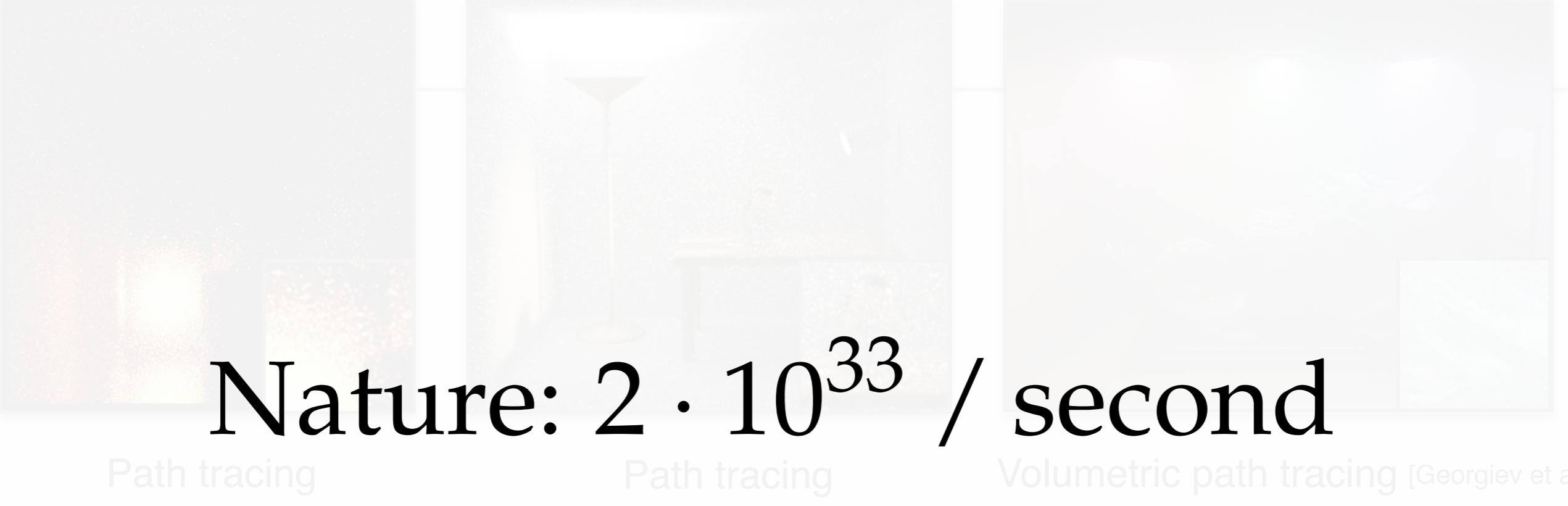
$$\eta = 1.00001$$

# + glossy materials



# + complex lighting



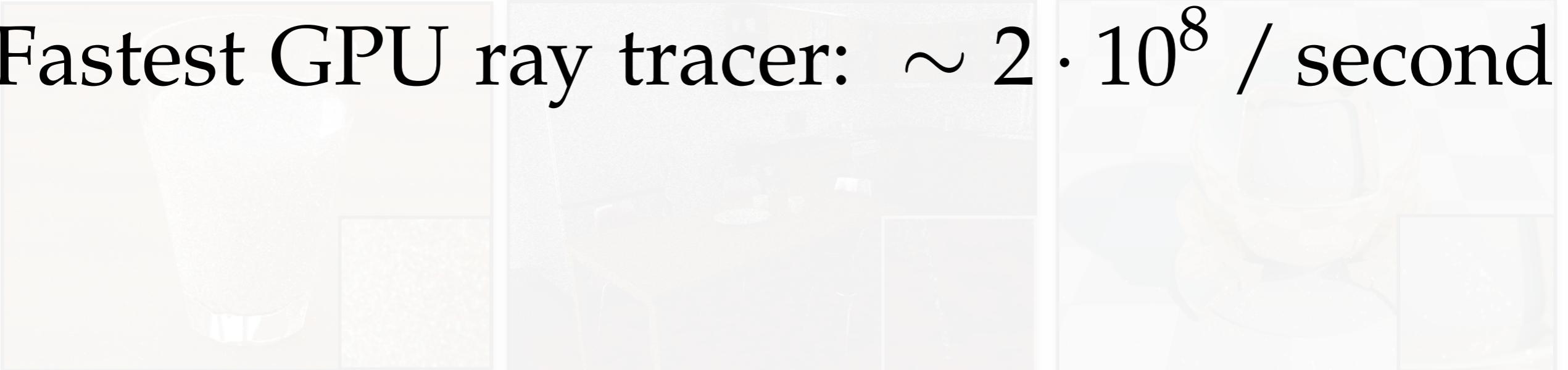


Nature:  $2 \cdot 10^{33}$  / second

Path tracing

Path tracing

Volumetric path tracing [Georgiev et al.]



Fastest GPU ray tracer:  $\sim 2 \cdot 10^8$  / second

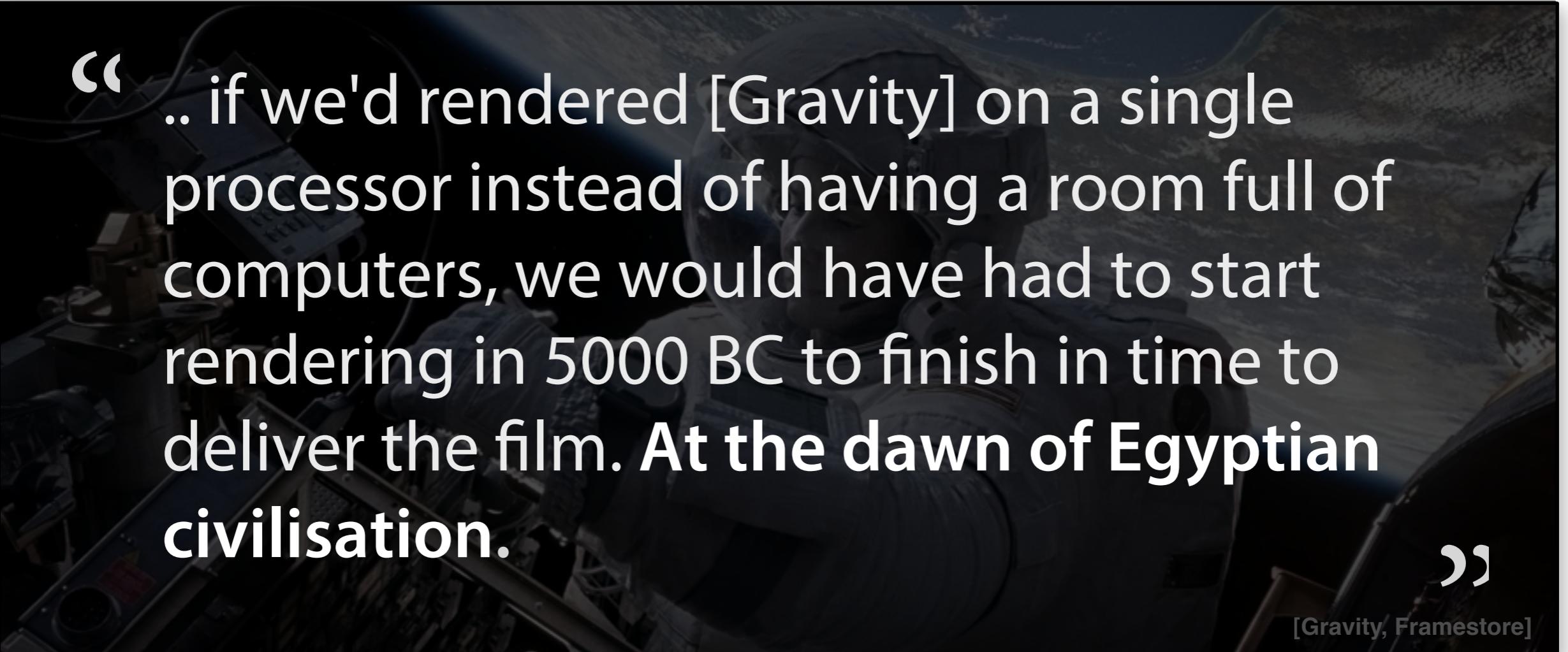
Bidirectional path tracing

Progressive photon mapping

Natural sun & sky illumination

# Costs of rendering

---



“ .. if we'd rendered [Gravity] on a single processor instead of having a room full of computers, we would have had to start rendering in 5000 BC to finish in time to deliver the film. **At the dawn of Egyptian civilisation.** ”

[Gravity, Framestore]

Tim Webber, Gravity VFX supervisor

# Key issues

---

- **Performance:** convergence takes prohibitively long
- **Independent samples:** tricky to reuse information
- **Robustness:** impractical for some combinations of lighting and materials
- **General issue with path sampling:** Can importance sample individual factors, but one term is always missing

# Definitions

---

## 1. Markov Chain

A random process that jumps from state to state.

The probability distribution of the next state only depends on the current state (but no previous ones)

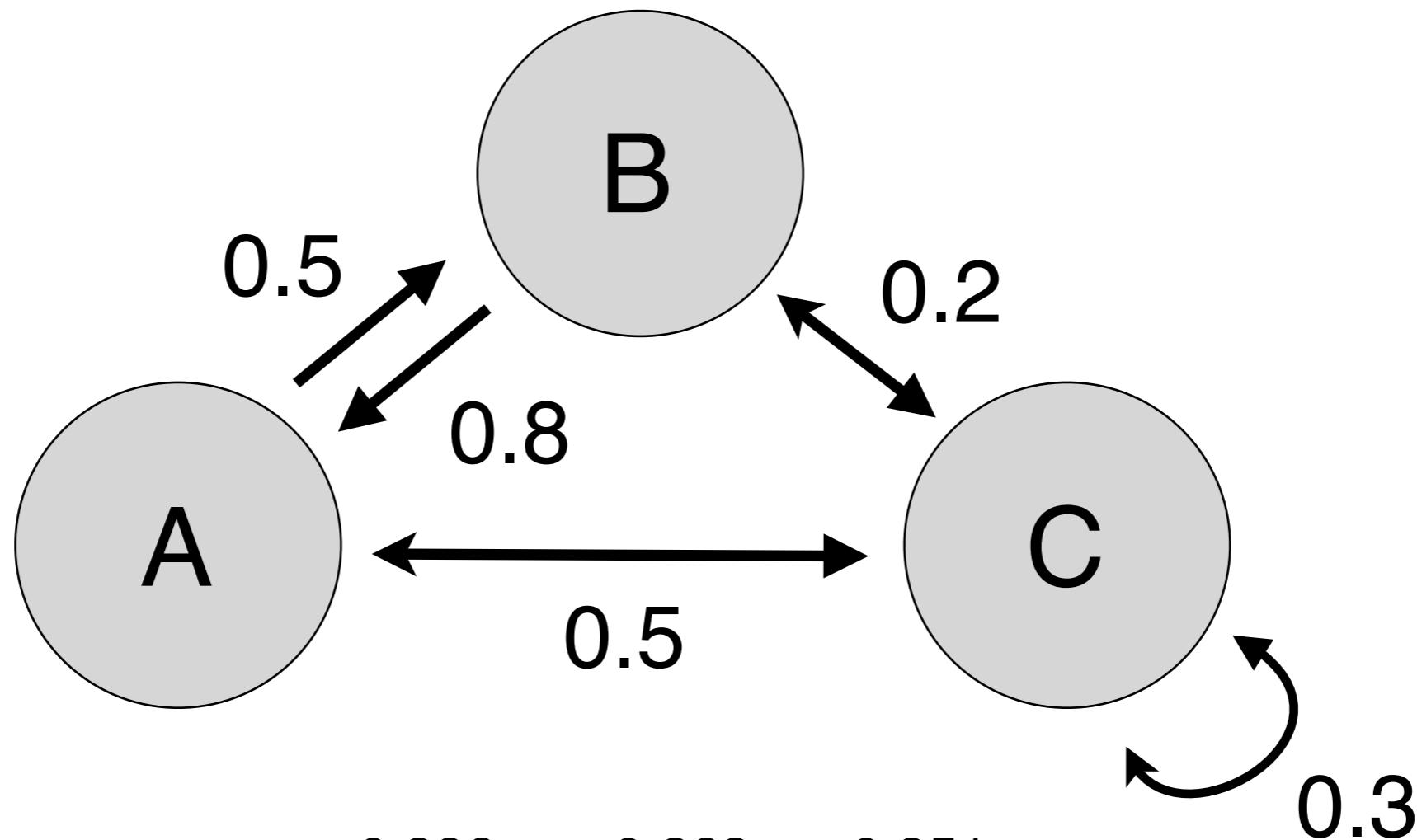
## 2. Markov Chain Monte Carlo (“MCMC”)

The use of special Markov Chains to create Monte Carlo methods.

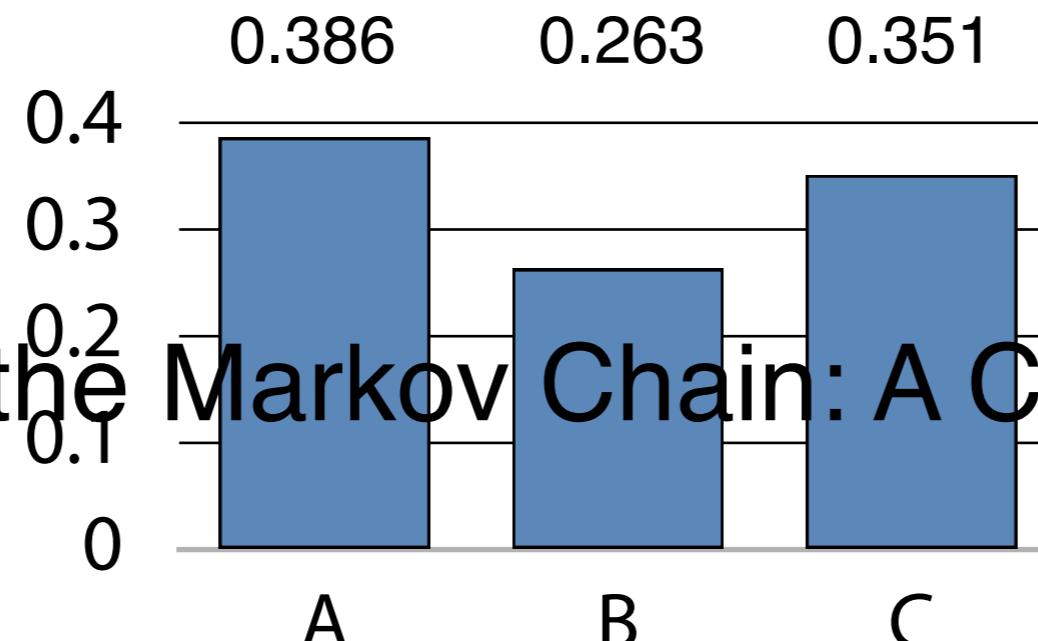
Classic example: **Metropolis-Hastings algorithm**

[Metropolis et al. 1953], [Hastings 1970]

# Markov Chain review



Simulating the Markov Chain: ACCCABAAC...



# Metropolis-Hastings review

---

Given

1. State space  $\Omega$  (discrete or continuous)
2. A probability distribution  $f : \Omega \rightarrow \mathbb{R}$

Generate a sequence  $\mathbf{x}_0, \mathbf{x}_1, \dots \sim f$ .

# MCMC Pseudocode

Let  $\mathbf{x}_0$  = initial state

For  $i = 1, \dots, n$  do

$\mathbf{x}'_i = \text{mutate}(\mathbf{x}_{i-1})$

$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i & \text{with probability } a(\mathbf{x}_{i-1} \rightarrow \mathbf{x}'_i) \\ \mathbf{x}_{i-1} & \text{otherwise} \end{cases}$

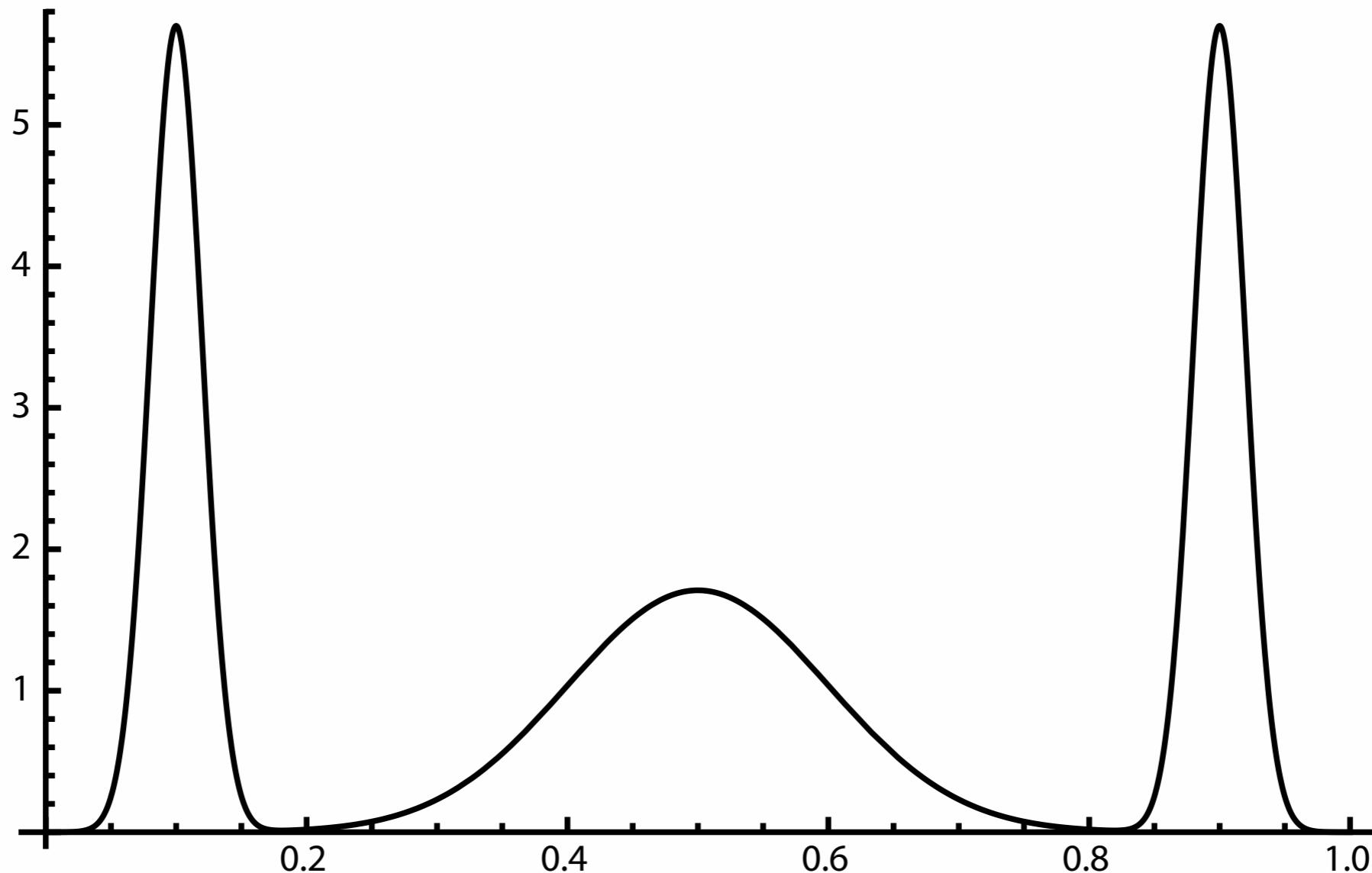
record( $\mathbf{x}_i$ )

where  $a(\mathbf{x} \rightarrow \mathbf{x}') := \min \left( 1, \frac{f(\mathbf{x}') T(\mathbf{x}' \rightarrow \mathbf{x})}{f(\mathbf{x}) T(\mathbf{x} \rightarrow \mathbf{x}')} \right)$

# Example application

---

Draw samples from a distribution and build a histogram



# MCMC Pseudocode

---

Let  $\mathbf{x}_0 = 0.5$

For  $i = 1, \dots, 1000$  do

$\mathbf{x}'_i = \text{mutate}(\mathbf{x}_{i-1})$

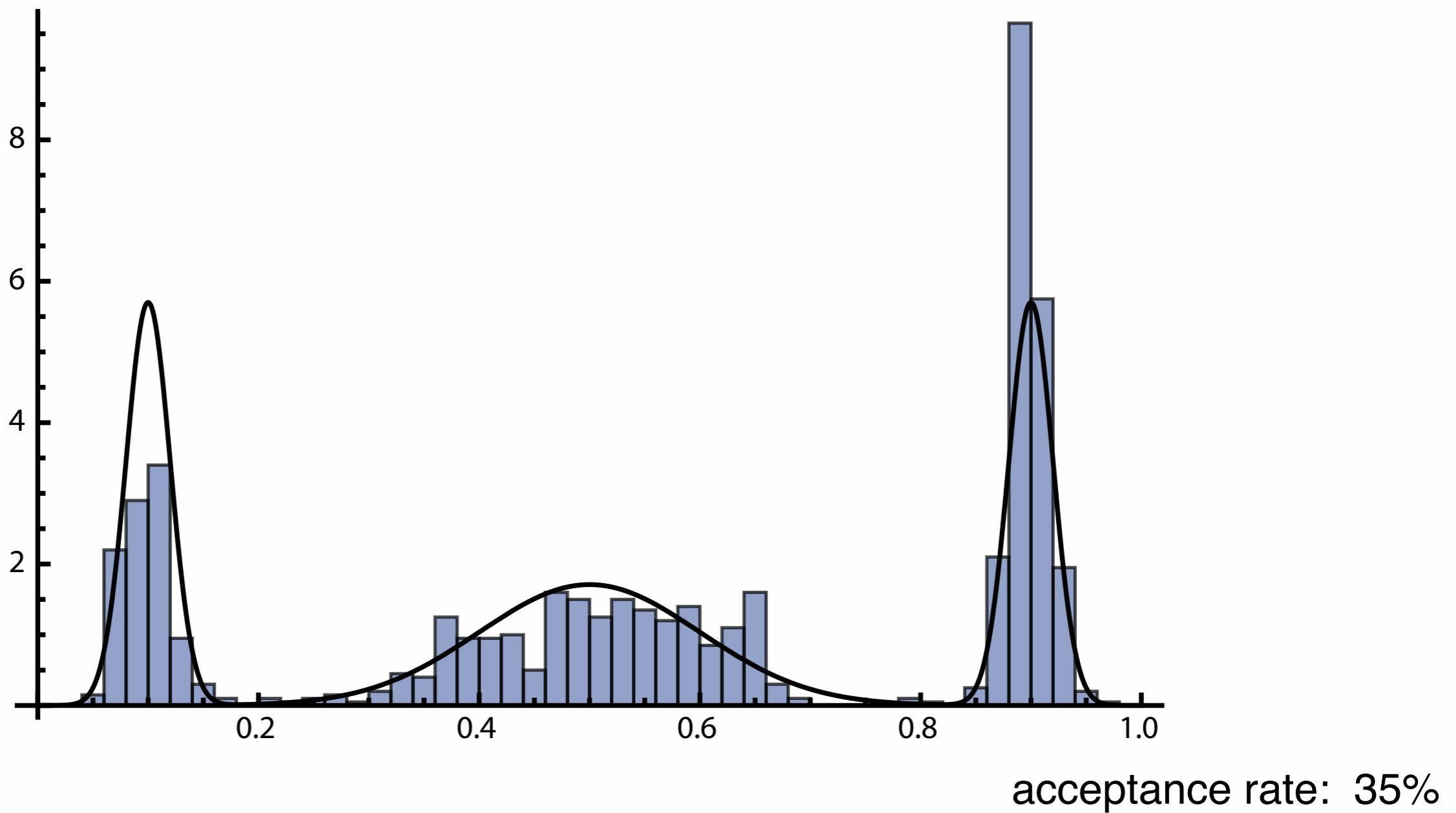
$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i & \text{with probability } a(\mathbf{x}_{i-1} \rightarrow \mathbf{x}'_i) \\ \mathbf{x}_{i-1} & \text{otherwise} \end{cases}$

record( $\mathbf{x}_i$ )      (increase bin count in histogram)

where  $a(\mathbf{x} \rightarrow \mathbf{x}') := \min \left( 1, \frac{f(\mathbf{x}') T(\mathbf{x}' \rightarrow \mathbf{x})}{f(\mathbf{x}) T(\mathbf{x} \rightarrow \mathbf{x}')} \right)$

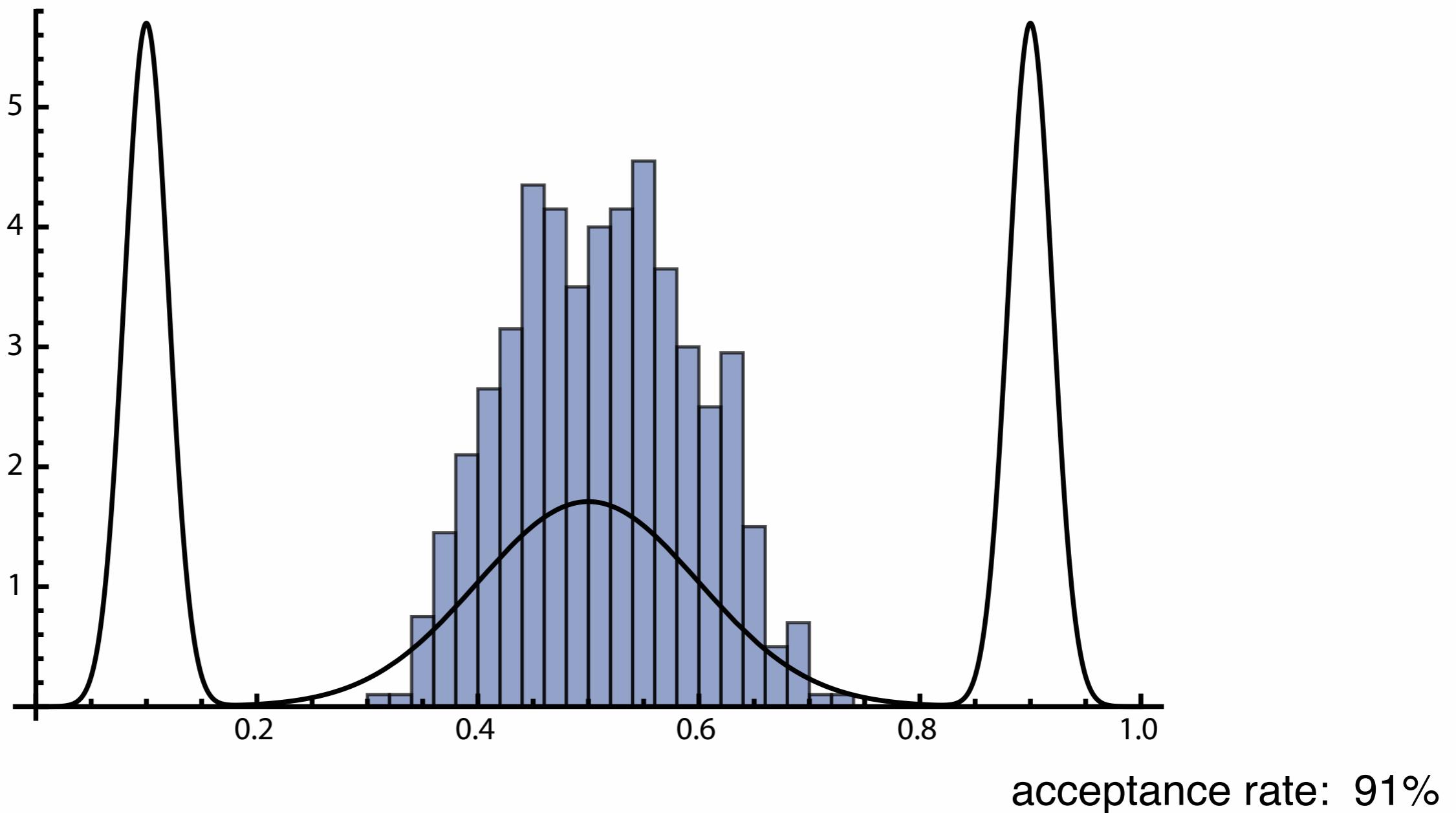
# Example application (mutation)

$\text{mutate}_1(\mathbf{x}) = \text{uniform sample on } [0, 1]$



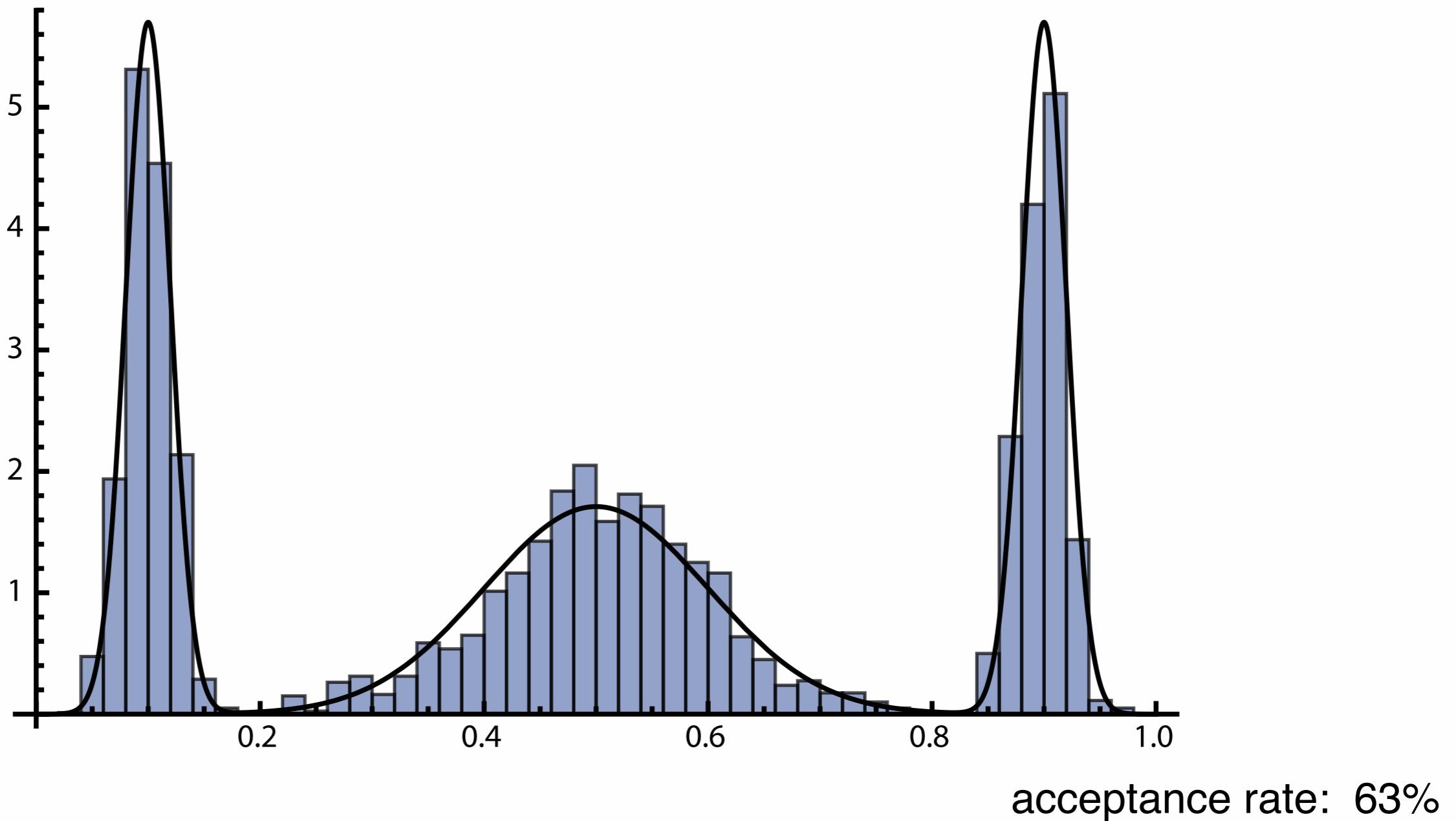
# Example application (perturbation)

`mutate2(x ) = uniform sample on [x - 0.05,x + 0.05]`



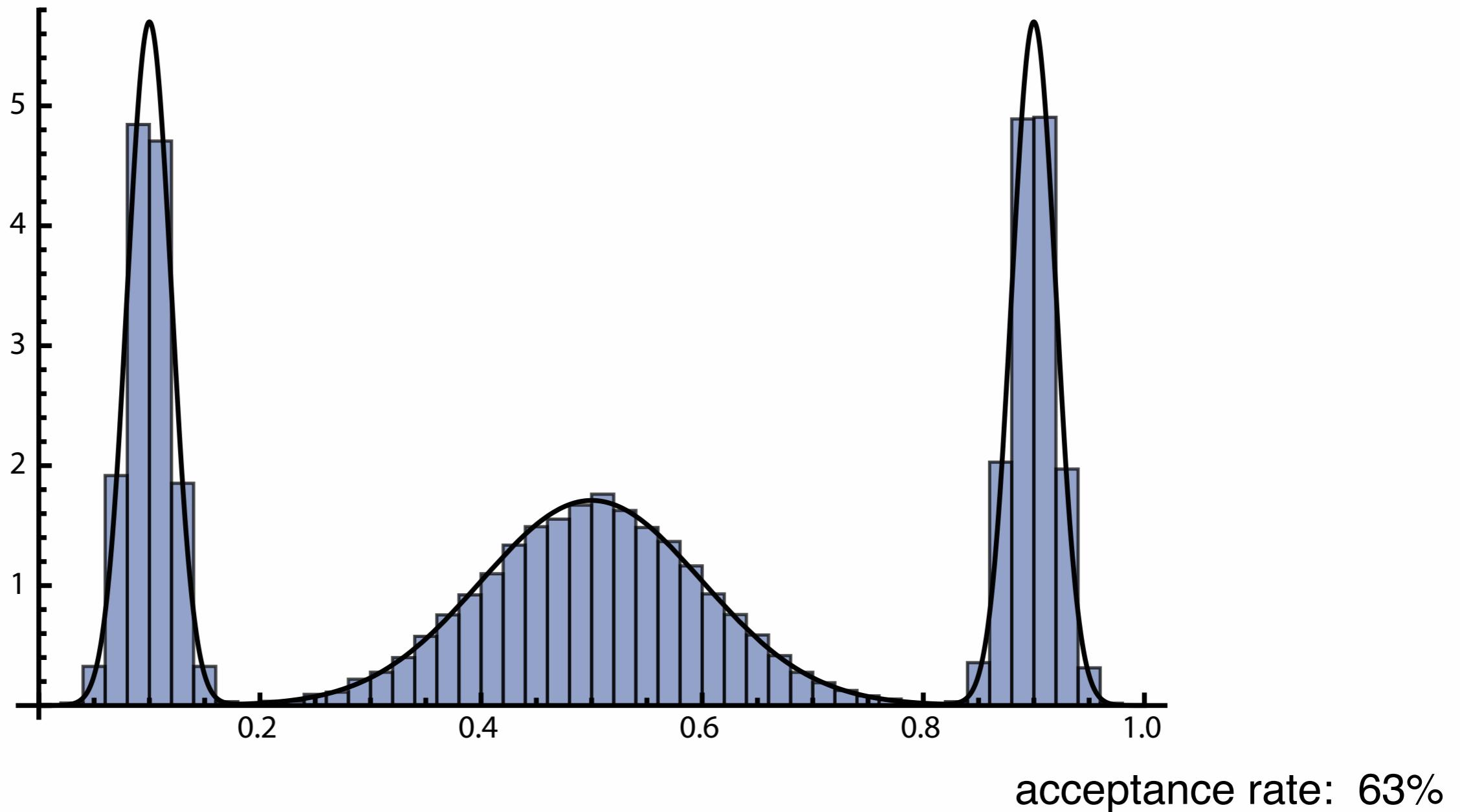
# Example application (both!)

`mutate3(x )` = randomly choose between `mutate1` and  
`mutate2`



# Example application (more samples)

`mutate3(x )` = randomly choose between `mutate1` and  
`mutate2`



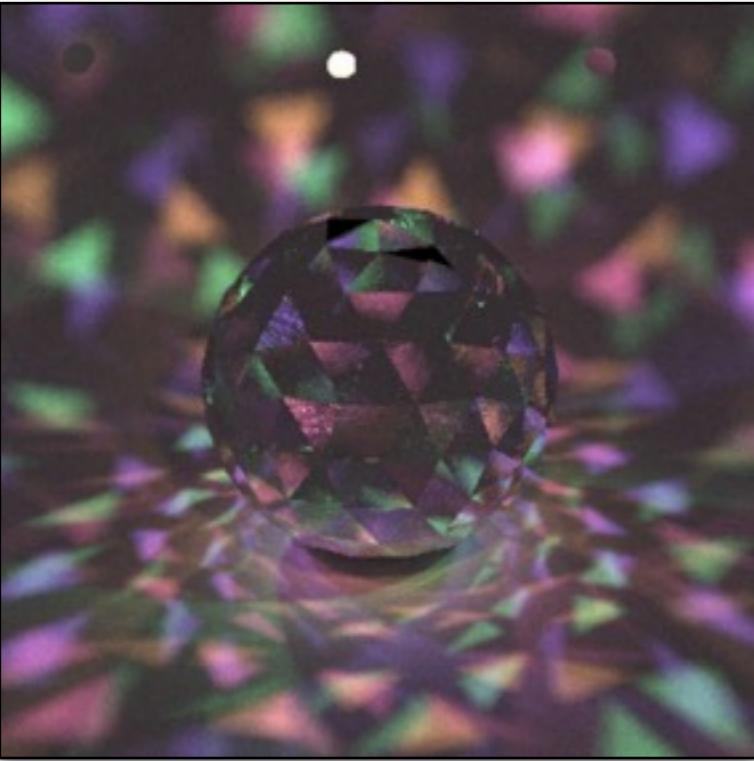
# Metropolis Hastings summary

---

1. Technique for sampling according to an arbitrary PDF  
(which must only be known up to a scale factor)
2. Samples are **correlated**
3. Degree of correlation related to efficiency

This is also known as the *mixing ratio*.

# “Metropolized” path tracing



*A Simple and Robust Mutation Strategy for  
the Metropolis Light Transport Algorithm*

[Kelemen et al. 2002]

# Path tracing pseudocode

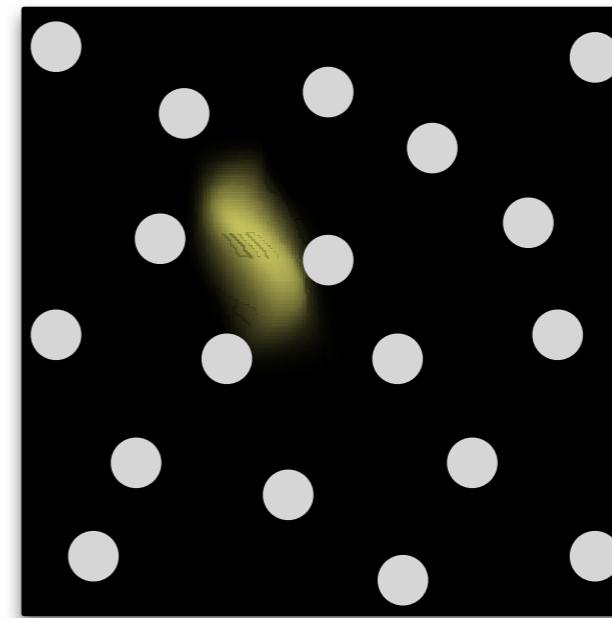
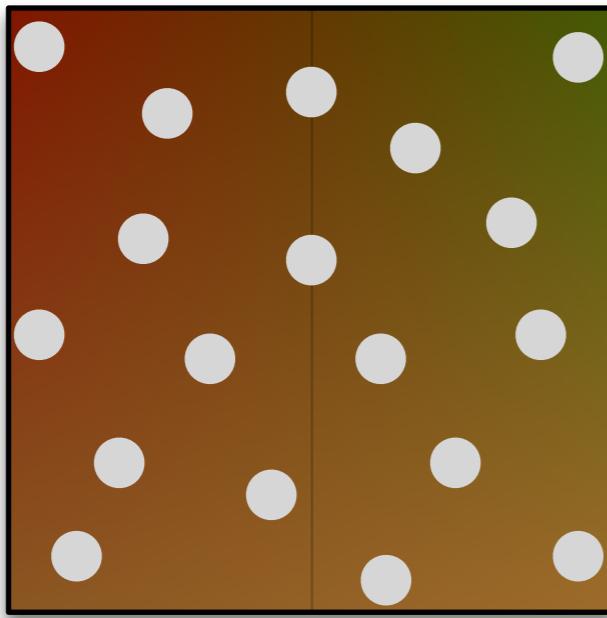
---

$\langle L_o(\mathbf{x}, \omega_o, \xi_0, \xi_1, \dots) \rangle :=$

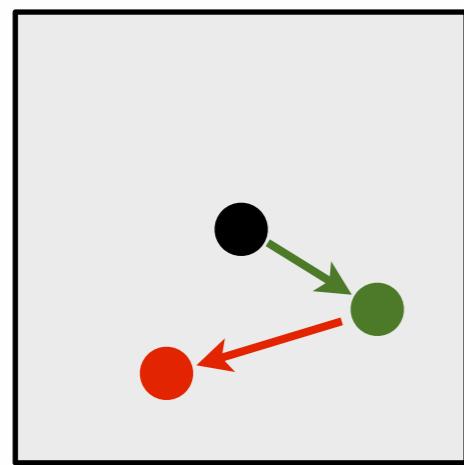
1. Return 0 with probability  $\alpha$
2. Sample a direction  $\omega_i$  proportional to  $f_s(\mathbf{x}, \cdot, \omega_o)$
3. Set  $\mathbf{x}' := r(\mathbf{x}, \omega_i)$
4. Return  $\frac{1}{1 - \alpha} \left( L_e(\mathbf{x}, \omega_o) + P_{f_s} \langle L_o(\mathbf{x}', -\omega_i) \rangle \right)$

# Primary sample space interpretation

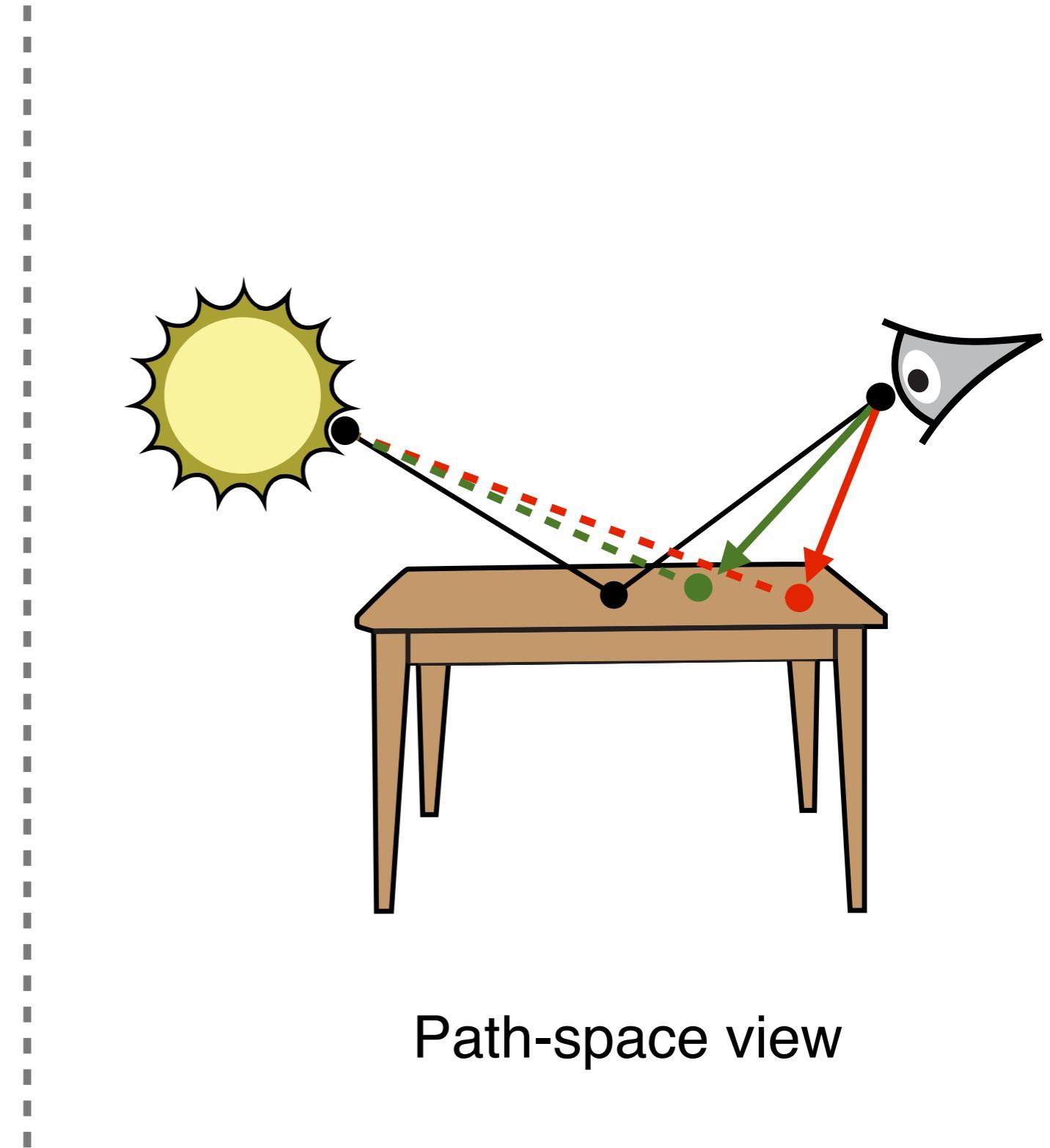
$$L = \int_0^1 \int_0^1 \cdots f(\xi_0, \xi_1, \dots, \dots) d\xi_0 d\xi_1 \cdots$$



# Primary sample-space MLT



Primary sample-space view



Path-space view

# MCMC Pseudocode

Let  $\mathbf{x}_0$  = initial state

For  $i = 1, \dots, n$  do

$\mathbf{x}'_i = \text{mutate}(\mathbf{x}_{i-1})$

$$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i & \text{with probability } a(\mathbf{x}_{i-1} \rightarrow \mathbf{x}'_i) \\ \mathbf{x}_{i-1} & \text{otherwise} \end{cases}$$

$\text{record}(\mathbf{x}_i)$

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

# Primary sample-space MLT

- Generally an improvement, but cannot do miracles.



Path tracer



“Metropolized” path tracer

# Today's menu

---

1. Review of surface light transport & path tracing
2. “Metropolized” path tracing
- 3. Path space integration**
4. Path space MCMC methods
5. Conclusion

# Path Integral Framework

# Recall: Measurement Equation

$$\begin{aligned} I_j &= \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0) d\mathbf{x}_1 d\mathbf{x}_0 \\ &= \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) + \int_A f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) L_o(\mathbf{x}_2, \mathbf{x}_1) d\mathbf{x}_2 d\mathbf{x}_1 d\mathbf{x}_0 \\ &= \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) + \int_A f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) L_e(\mathbf{x}_2, \mathbf{x}_1) + \int_A f(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_1) G(\mathbf{x}_2, \mathbf{x}_3) L_e(\mathbf{x}_3, \mathbf{x}_2) + \int_A \cdots d\mathbf{x}_4 d\mathbf{x}_3 d\mathbf{x}_2 d\mathbf{x}_1 d\mathbf{x}_0 \end{aligned}$$

Hard to concisely express arbitrary light transport with all the nested integrals



Let's find a better way

# Path Integral Form of Measurement Eq.

$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0) d\mathbf{x}_1 d\mathbf{x}_0$$

possibly add illustr

$$= \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) G(\mathbf{x}_0, \mathbf{x}_1) d\mathbf{x}_1 d\mathbf{x}_0$$

$$+ \int_A \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_2, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 d\mathbf{x}_1 d\mathbf{x}_0 + \dots$$

$$+ \int_A \cdots \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1}) d\mathbf{x}_k \cdots d\mathbf{x}_0 + \dots$$

introduce:  $\mathcal{P}_k = \{\bar{\mathbf{x}} = \mathbf{x}_0 \cdots \mathbf{x}_k; \mathbf{x}_0 \cdots \mathbf{x}_k \in A\}$   
space of all paths with  $k$  segments

# Path Integral Form of Measurement Eq.

$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0) d\mathbf{x}_1 d\mathbf{x}_0$$

$$= \int_{\mathcal{P}_1} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) G(\mathbf{x}_0, \mathbf{x}_1) d\bar{\mathbf{x}}_1$$

$$+ \int_{\mathcal{P}_2} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_2, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) d\bar{\mathbf{x}}_2 + \dots$$

$$+ \int_{\mathcal{P}_k} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1}) d\bar{\mathbf{x}}_k + \dots$$

introduce:  $T(\bar{\mathbf{x}}_k) = G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1})$   
*throughput of path  $\bar{\mathbf{x}}_k$*

# Path Integral Form of Measurement Eq.

$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0) d\mathbf{x}_1 d\mathbf{x}_0$$

**Emission**

$$= \int_{\mathcal{P}_1} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) T(\bar{\mathbf{x}}_1) d\bar{\mathbf{x}}_1$$

**Direct illumination (3 vertices)**

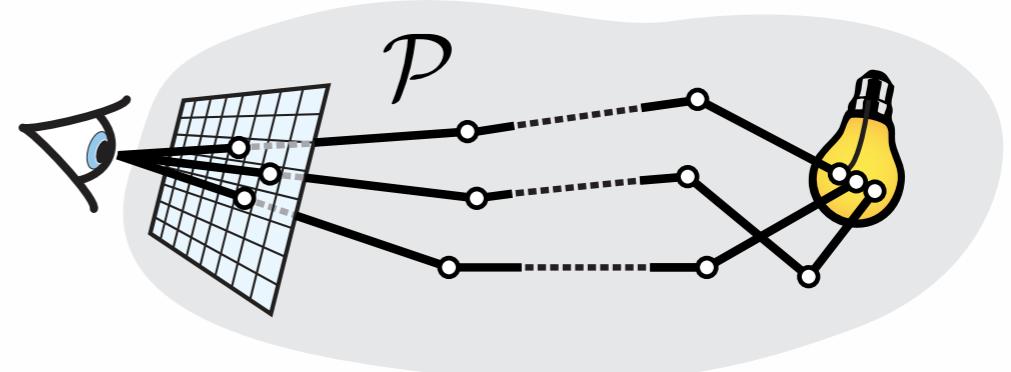
$$+ \int_{\mathcal{P}_2} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_2, \mathbf{x}_1) T(\bar{\mathbf{x}}_2) d\bar{\mathbf{x}}_2 + \dots$$

**(k-2)-bounce illumination (k vertices)**

$$+ \int_{\mathcal{P}_k} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}_k) d\bar{\mathbf{x}}_k + \dots$$

introduce:  $\mathcal{P} = \bigcup_{k=1}^{\infty} \mathcal{P}_k$

the *path space*, i.e. the space  
of all paths of all lengths



# Path Integral Form of Measurement Eq.

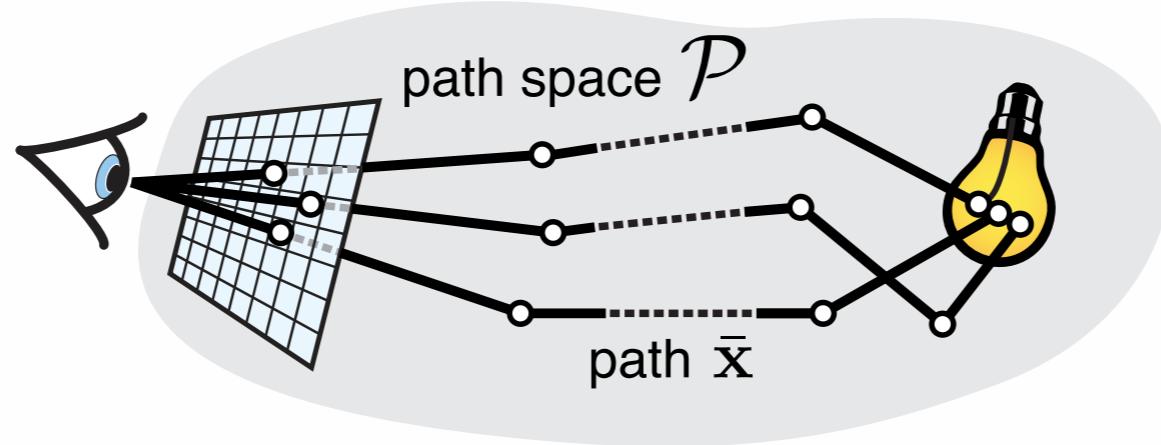
$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0) d\mathbf{x}_1 d\mathbf{x}_0$$

$$= \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}) d\bar{\mathbf{x}}$$

global illumination (all paths of all lengths)

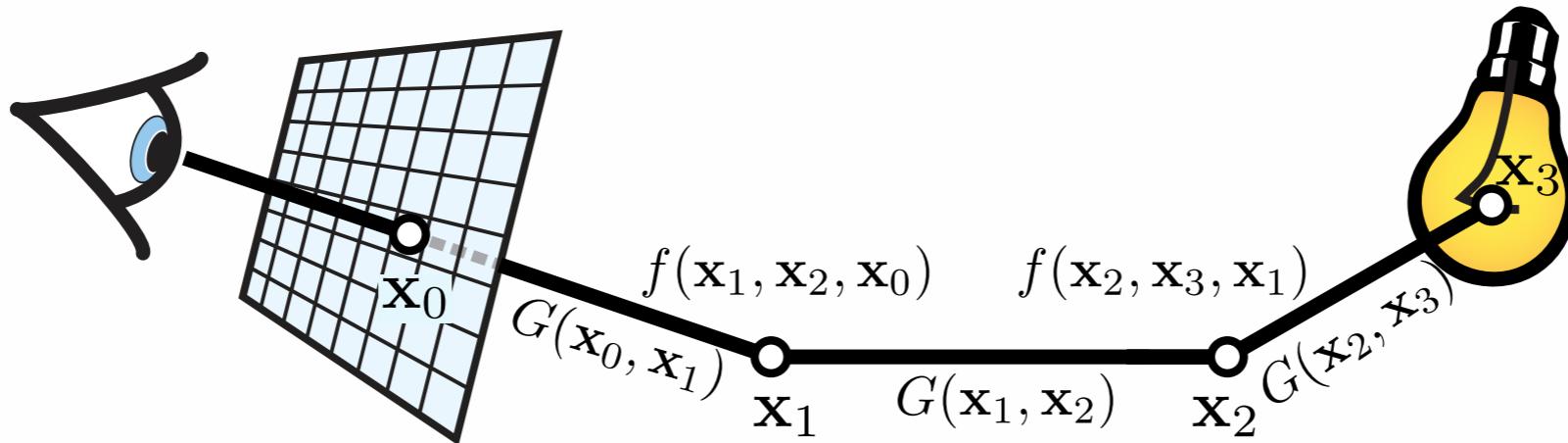
# Path Integral Form of Measurement Eq.

$$I_j = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}) d\bar{\mathbf{x}}$$



path throughput

$$T(\bar{\mathbf{x}}) = G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1})$$



# Path Integral Form of Measurement Eq.

---

$$I_j = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}) d\bar{\mathbf{x}}$$

- Advantages:
  - no recursion, no “nasty” nested integrals
  - emphasizes symmetry of light transport
  - easy to relate different rendering algorithms
  - emphasizes path geometry abstracting from incident and outgoing quantities
  - MC estimator on path space looks much simpler

# Monte Carlo Estimator

---

$$I_j = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}) d\bar{\mathbf{x}}$$

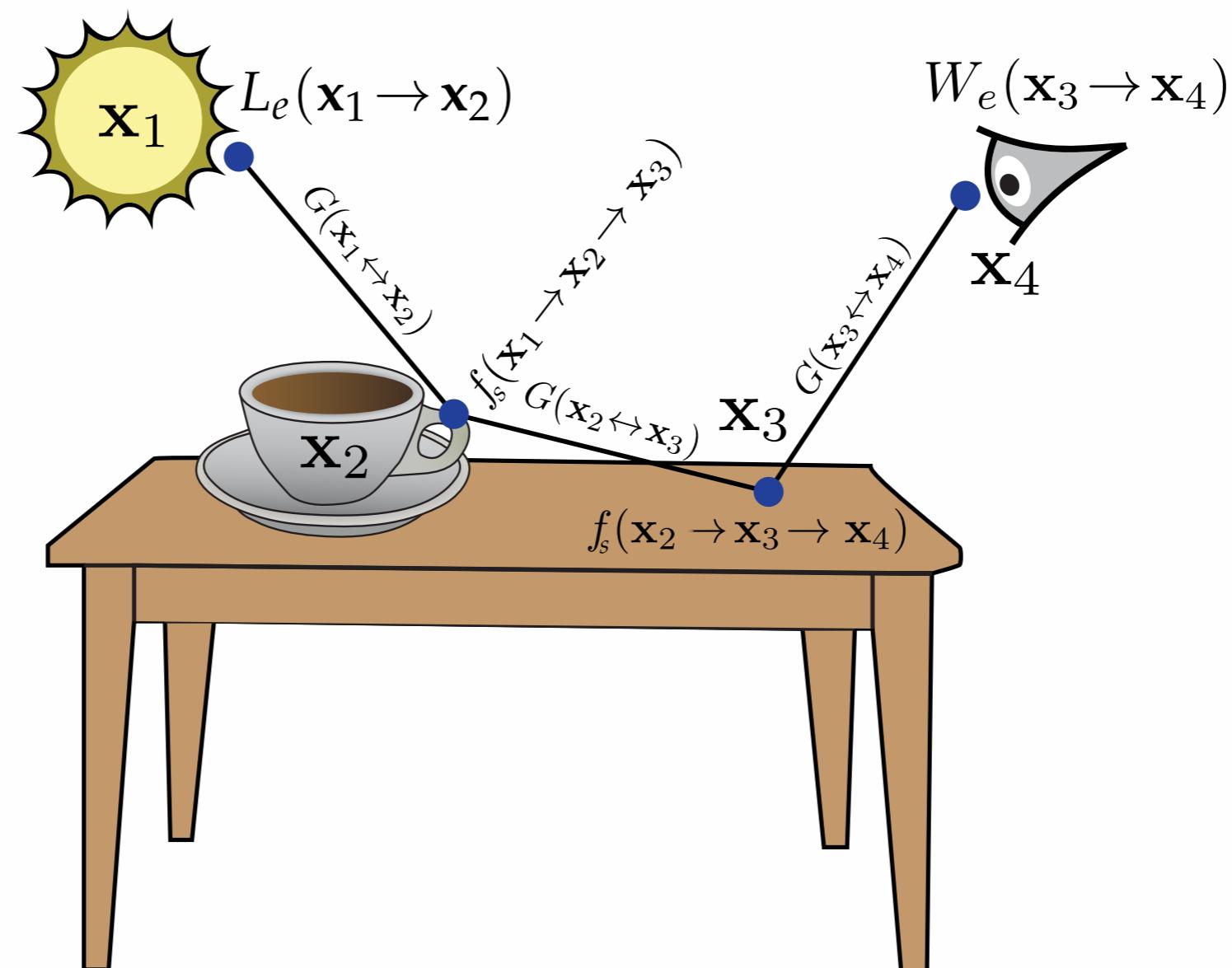
- Monte Carlo estimator:

$$I_j \approx \frac{1}{N} \sum_{i=1}^N \frac{W_e(\mathbf{x}_{i,0}, \mathbf{x}_{i,1}) L_e(\mathbf{x}_{i,k}, \mathbf{x}_{i,k-1}) T(\bar{\mathbf{x}}_i)}{p(\bar{\mathbf{x}}_i)}$$

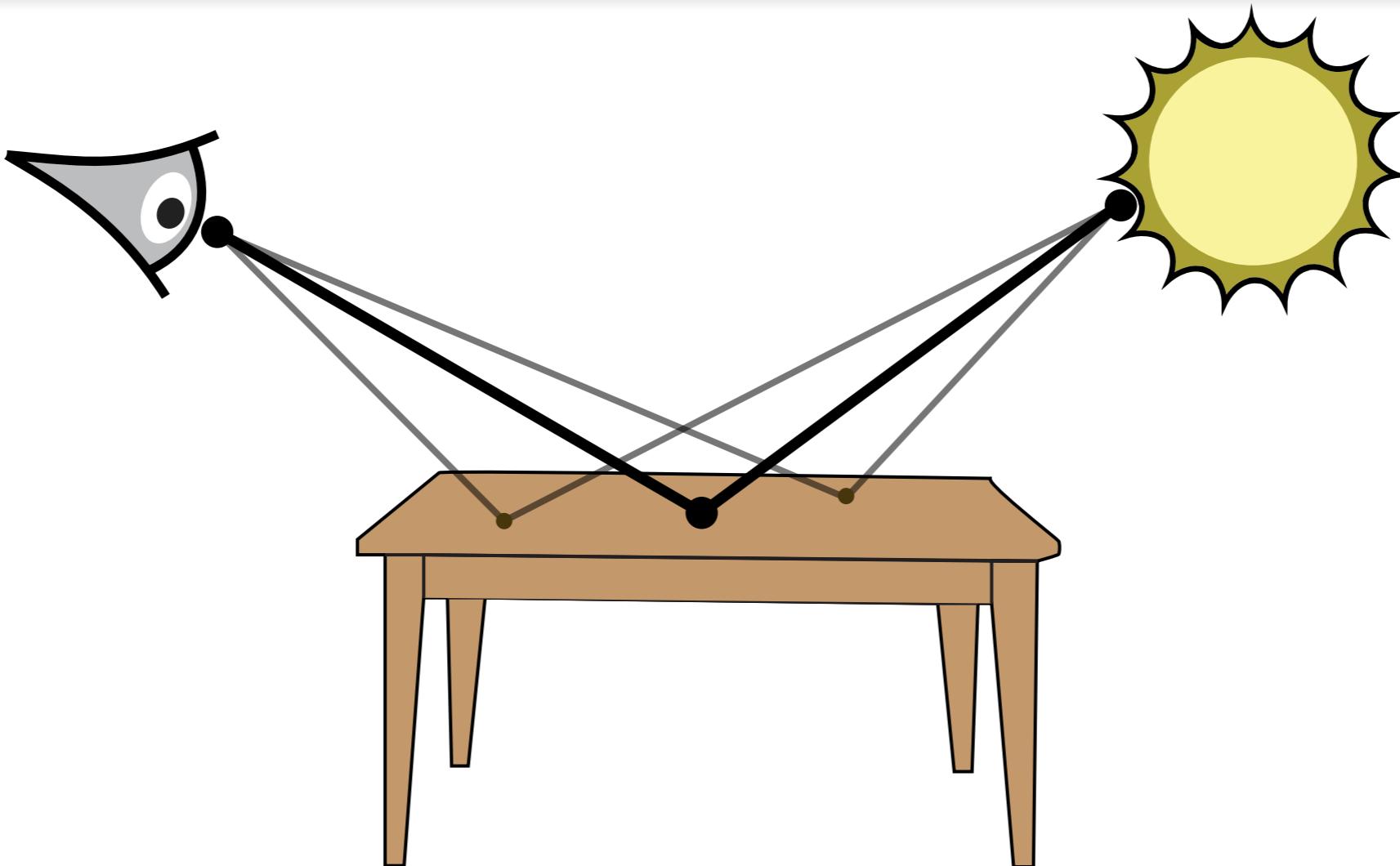
$$p(\bar{\mathbf{x}}) = p(\underbrace{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{k-1}, \mathbf{x}_k}_{\text{path PDF}}, \underbrace{\mathbf{x}_k}_{\text{joint PDF of path vertices}})$$

# Path space integration: an example

$$f(\bar{\mathbf{x}}) = L_e(\mathbf{x}_1 \rightarrow \mathbf{x}_2) G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2) f_s(\mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \mathbf{x}_3) G(\mathbf{x}_2 \leftrightarrow \mathbf{x}_3)$$
$$f_s(\mathbf{x}_2 \rightarrow \mathbf{x}_3 \rightarrow \mathbf{x}_4) G(\mathbf{x}_3 \leftrightarrow \mathbf{x}_4) W_e(\mathbf{x}_3 \rightarrow \mathbf{x}_4)$$



# Transition to Path Space



$$I = \int_{\mathcal{P}} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}})$$

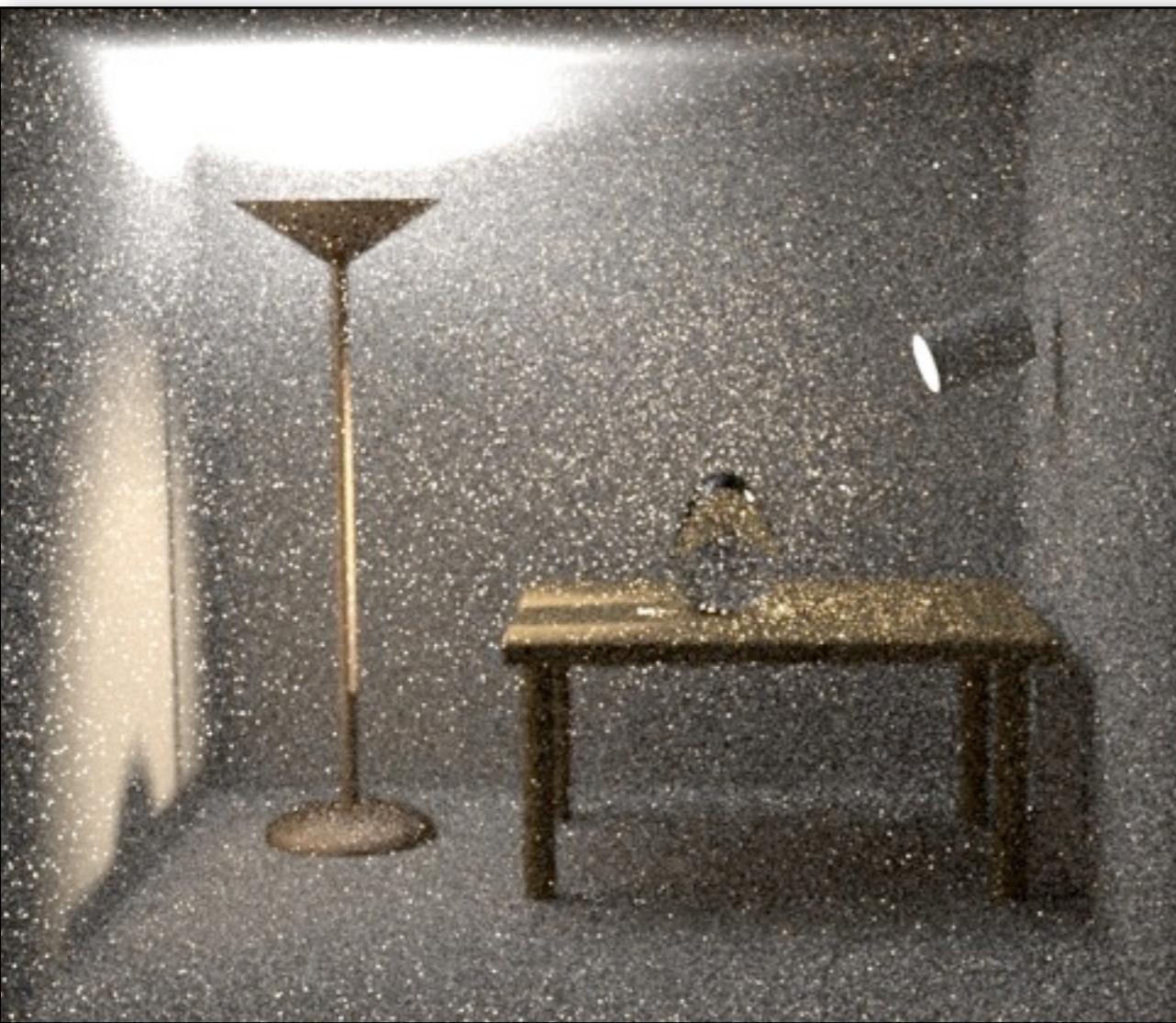
$\mathcal{P}$  Path space

# Path space summary

---

1. Alternative equation (but exactly the same solution)
2. *Explicit* formulation as an integral over paths
3. Enables new sampling techniques compared to naive recursive path tracing

# Bidirectional Path Tracing



(Unidirectional) path tracing



Bidirectional path tracing

# Bidirectional Path Tracing

- Some recent extensions combine density estimation and bidirectional path tracing

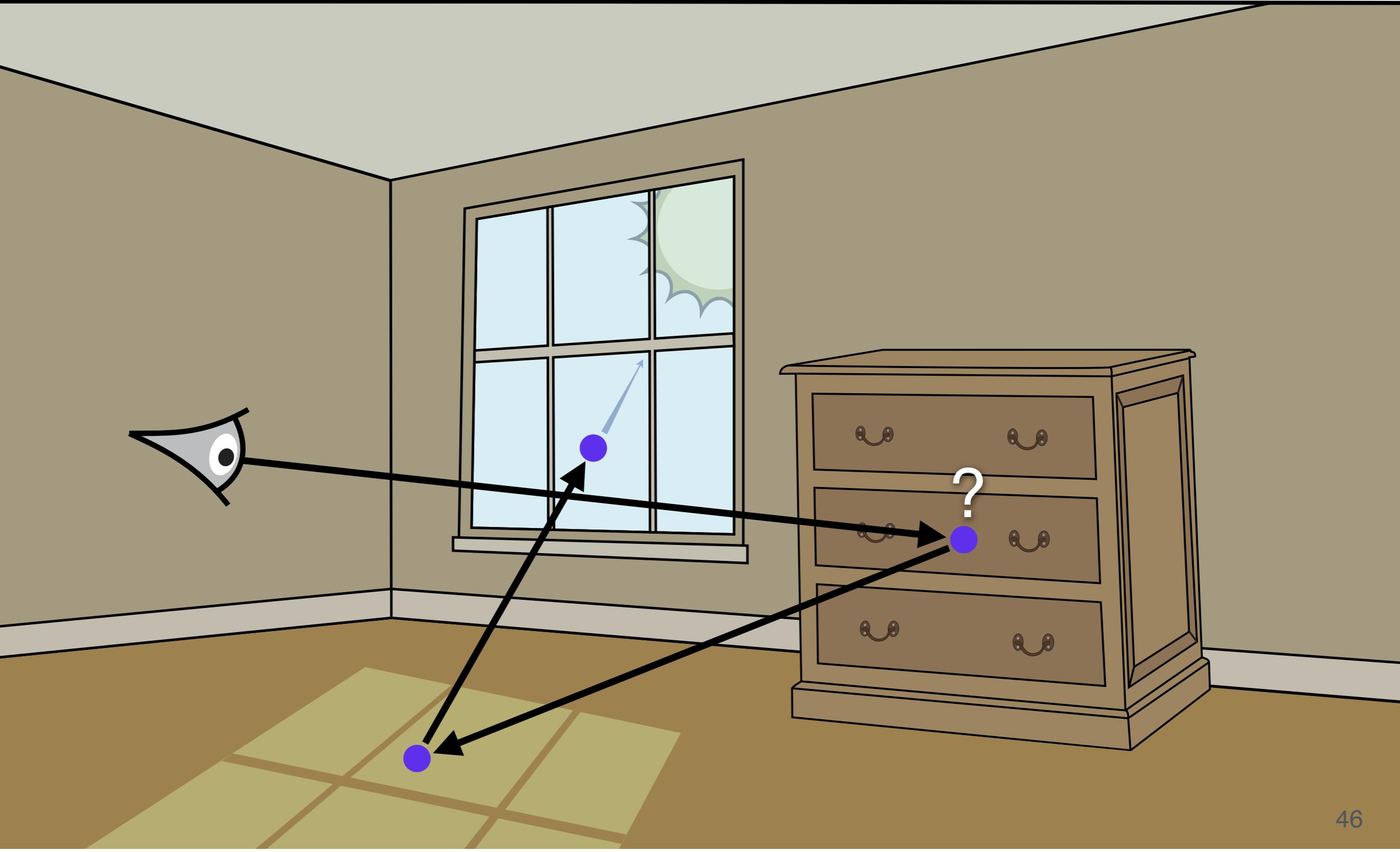


[Hachisuka et al. 2012]



[Georgiev et al. 2012]

# Problems of Bidirectional Path Tracing



# Today's menu

---

1. Review of surface light transport & path tracing
2. “Metropolized” path tracing
3. Path space integration
- 4. Path space MCMC methods**
5. Conclusion

# Metropolis Light Transport Algorithm



*Metropolis Light Transport*  
[Veach and Guibas 1997]

Related: *On solving integral equations using Markov chain Monte Carlo methods* [Doucet et al. 2010]

# MCMC Pseudocode

---

Let  $\mathbf{x}_0$  = initial state

For  $i = 1, \dots, n$  do

$\mathbf{x}'_i = \text{mutate}(\mathbf{x}_{i-1})$

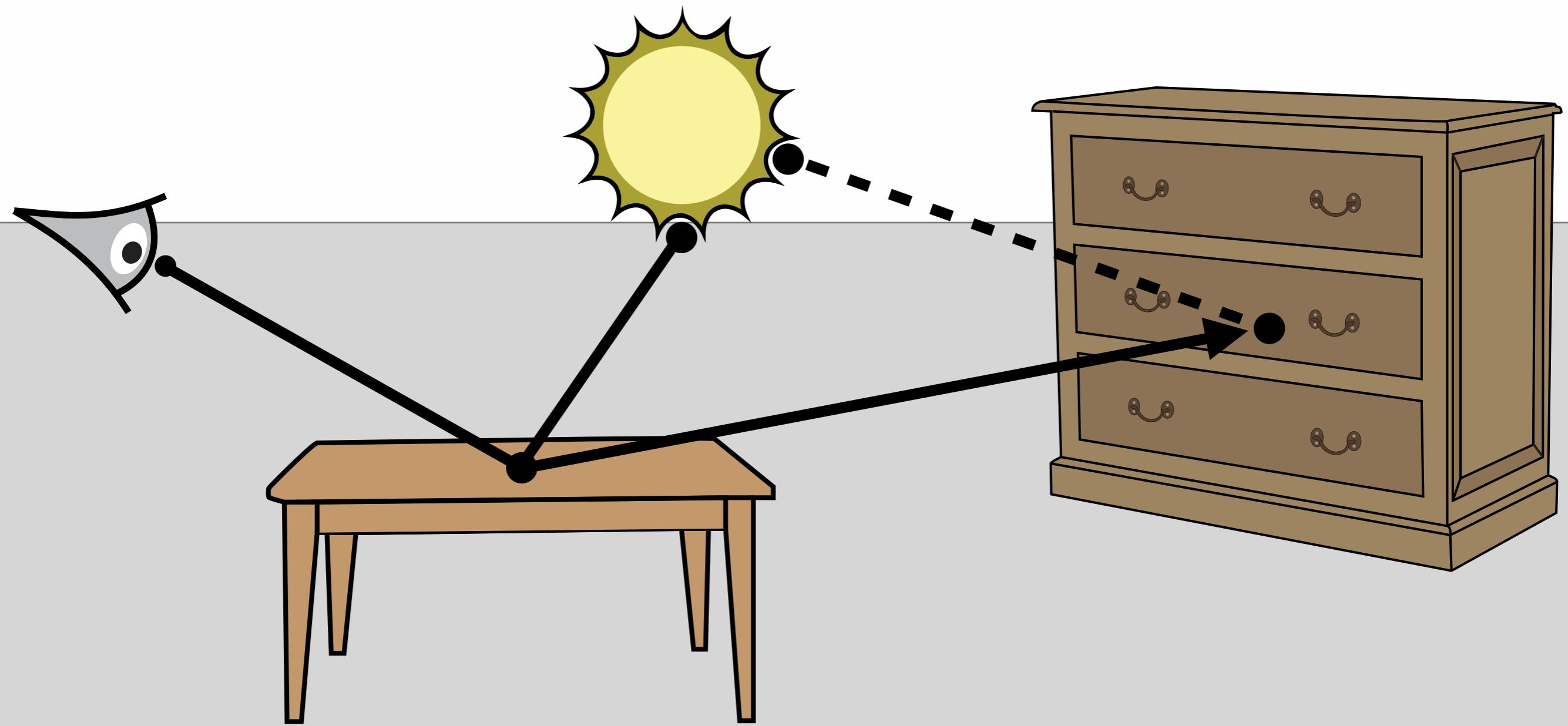
$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i & \text{with probability } a(\mathbf{x}_{i-1} \rightarrow \mathbf{x}'_i) \\ \mathbf{x}_{i-1} & \text{otherwise} \end{cases}$

record( $\mathbf{x}_i$ )

where  $a(\mathbf{x} \rightarrow \mathbf{x}') := \min \left( 1, \frac{f(\mathbf{x}') T(\mathbf{x}' \rightarrow \mathbf{x})}{f(\mathbf{x}) T(\mathbf{x} \rightarrow \mathbf{x}')} \right)$

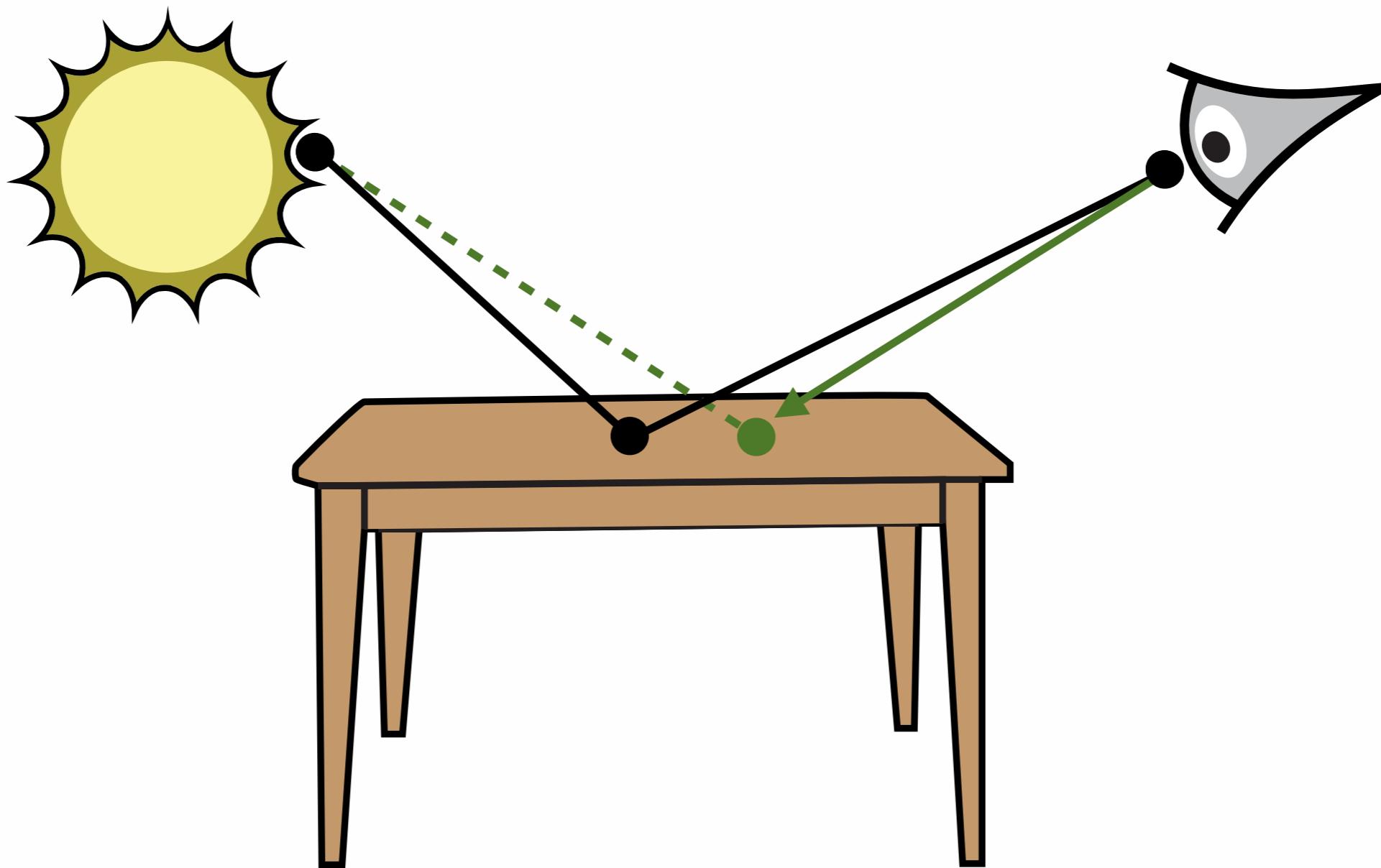
# Metropolis Light Transport

Bidirectional mutation



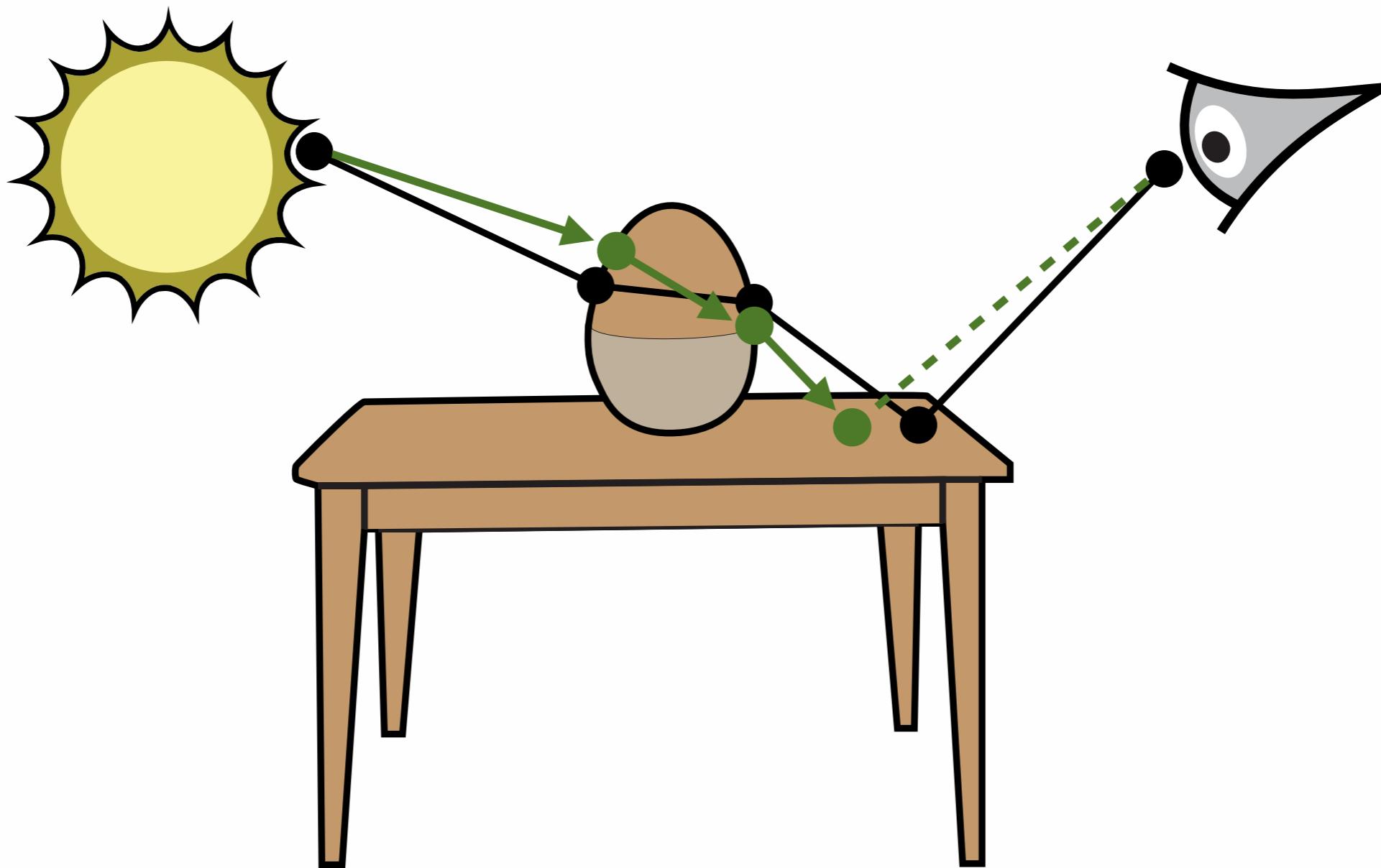
# Metropolis Light Transport

Lens perturbation



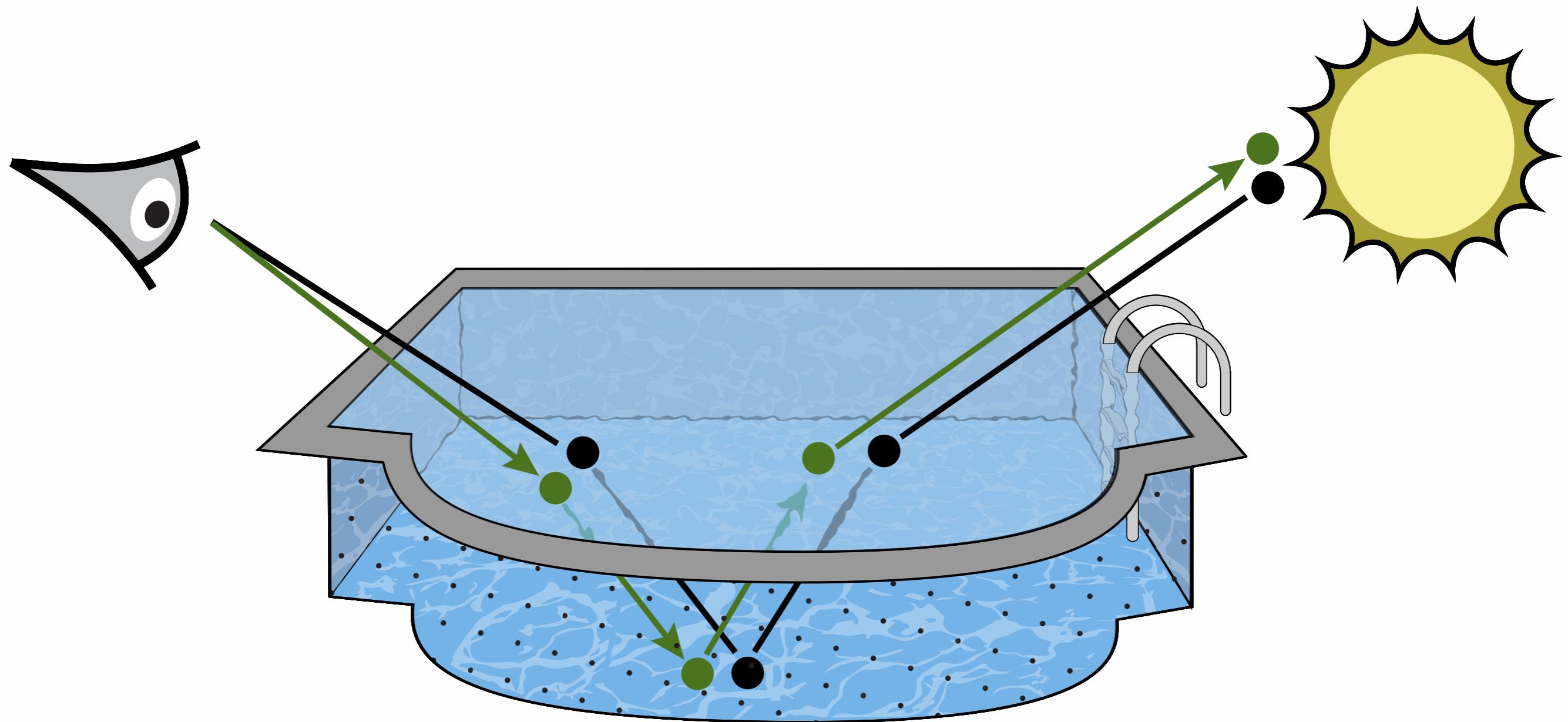
# Metropolis Light Transport

Caustic perturbation

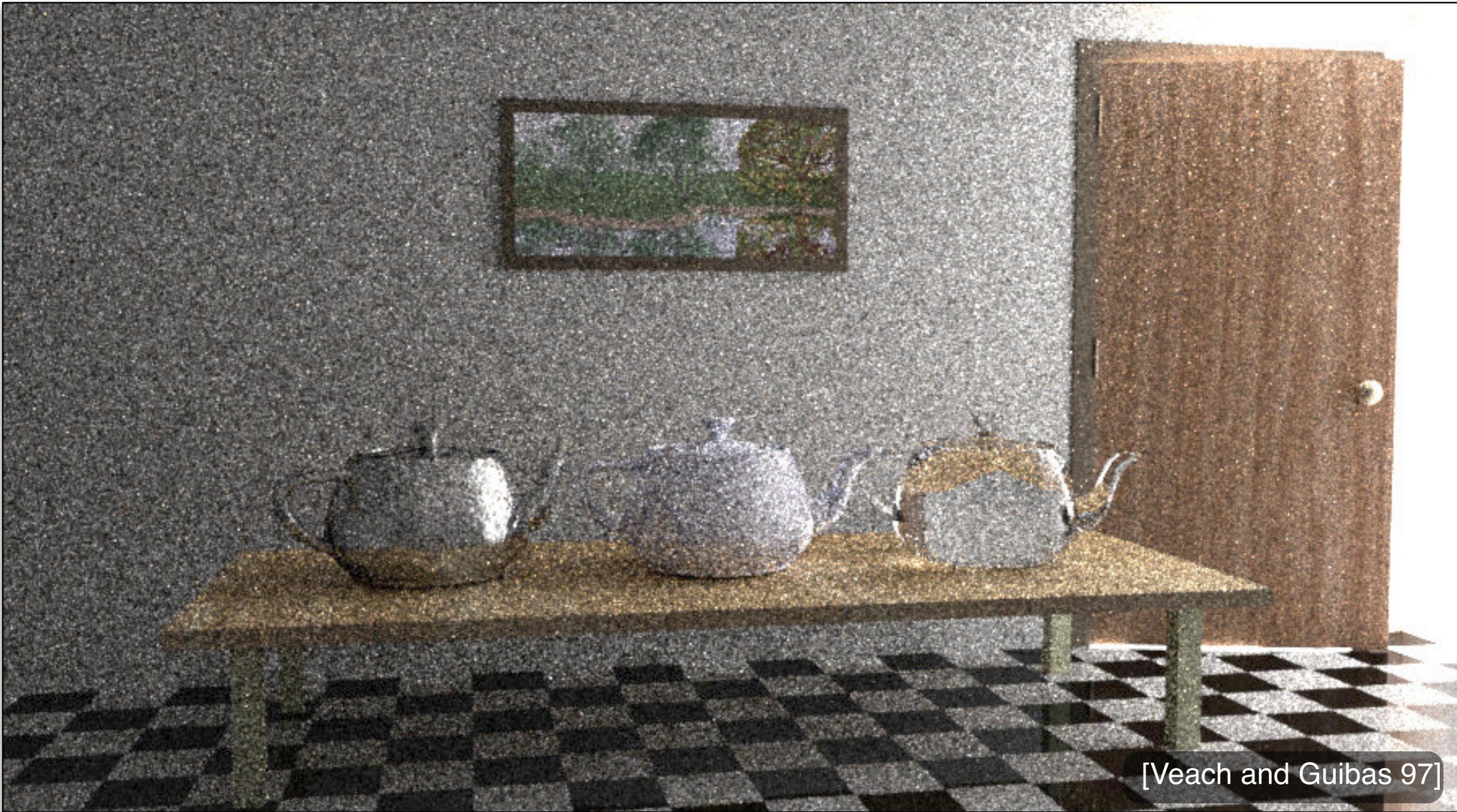


# Metropolis Light Transport

Multi-Chain perturbation



# Door scene (BDPT)



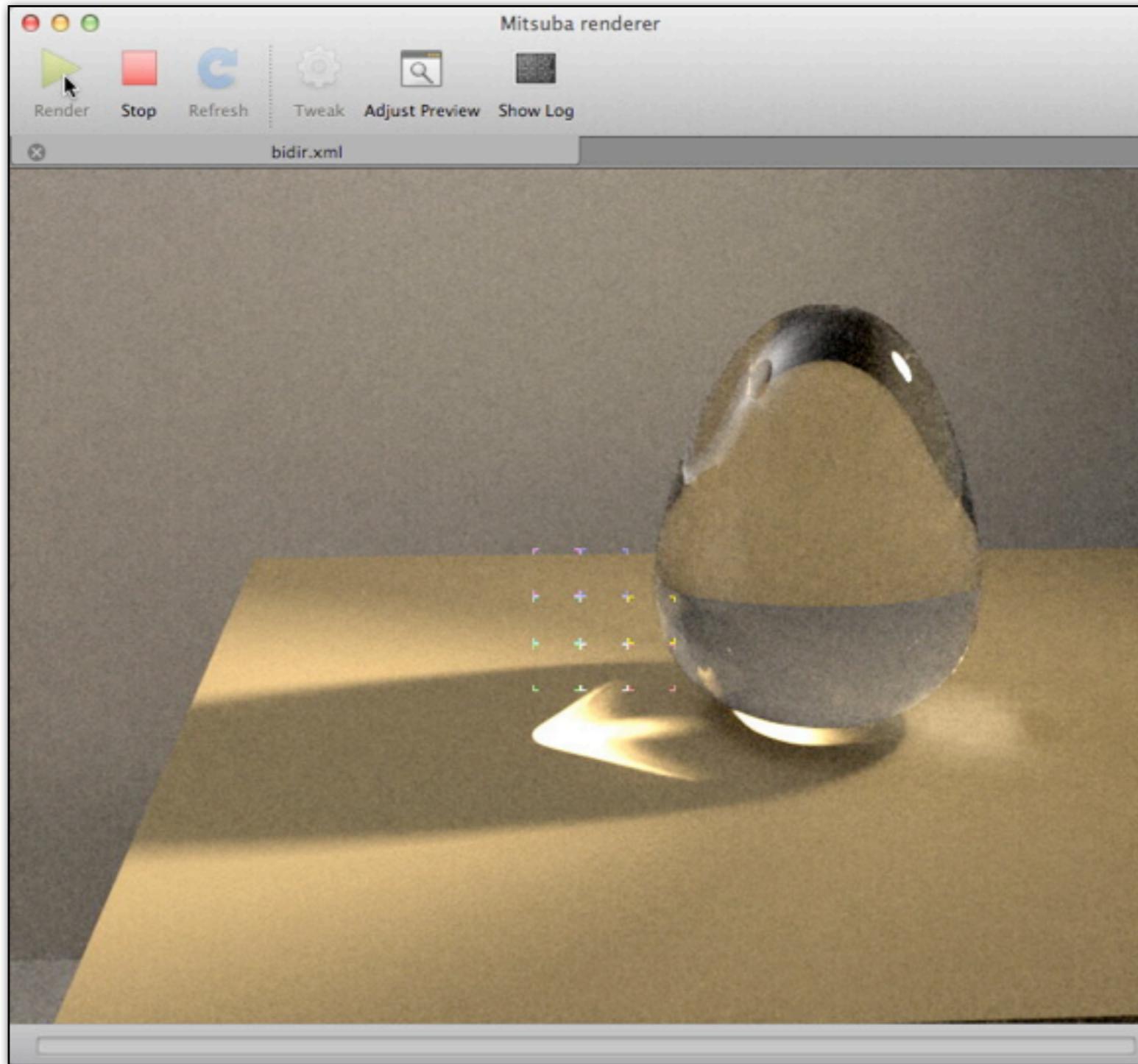
[Veach and Guibas 97]

# Door scene (MLT)



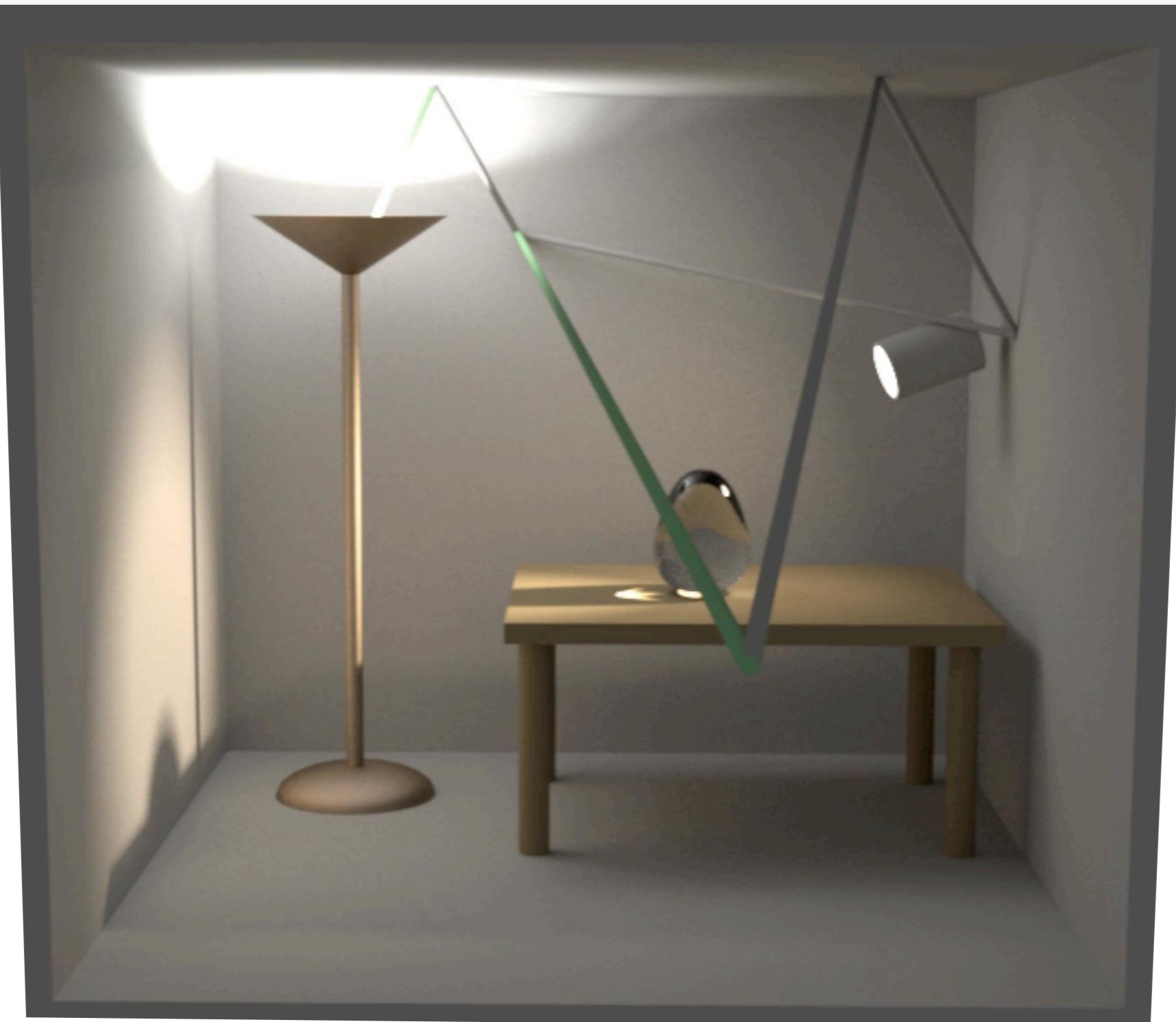
[Veach and Guibas 97]

# Caustic demo



8x timelapse

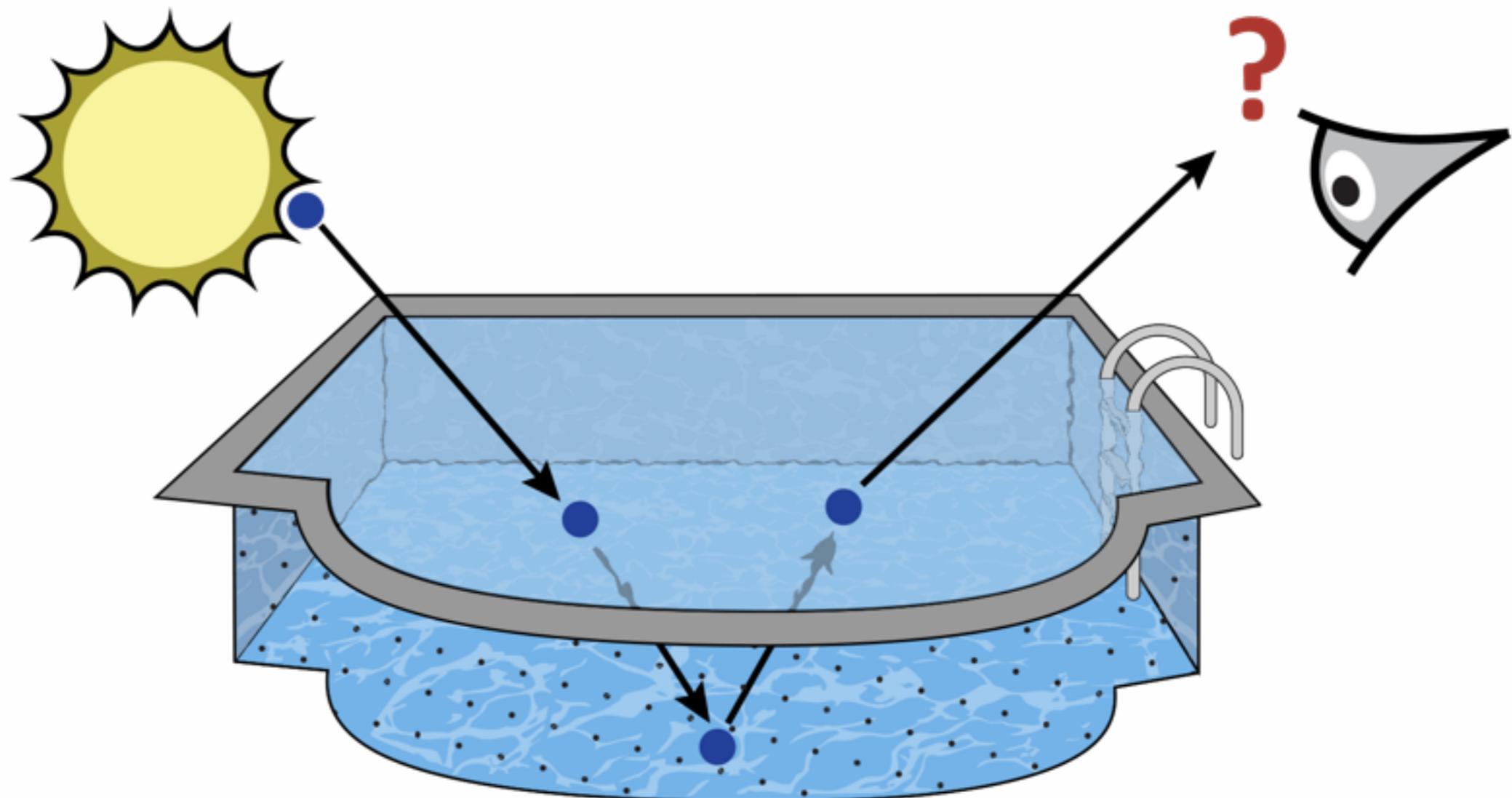
# Light path visualization



green = current state

gray = proposal state

# Related: Specular paths



Path tracing from  
the light source

# An observation in flatland

Light source

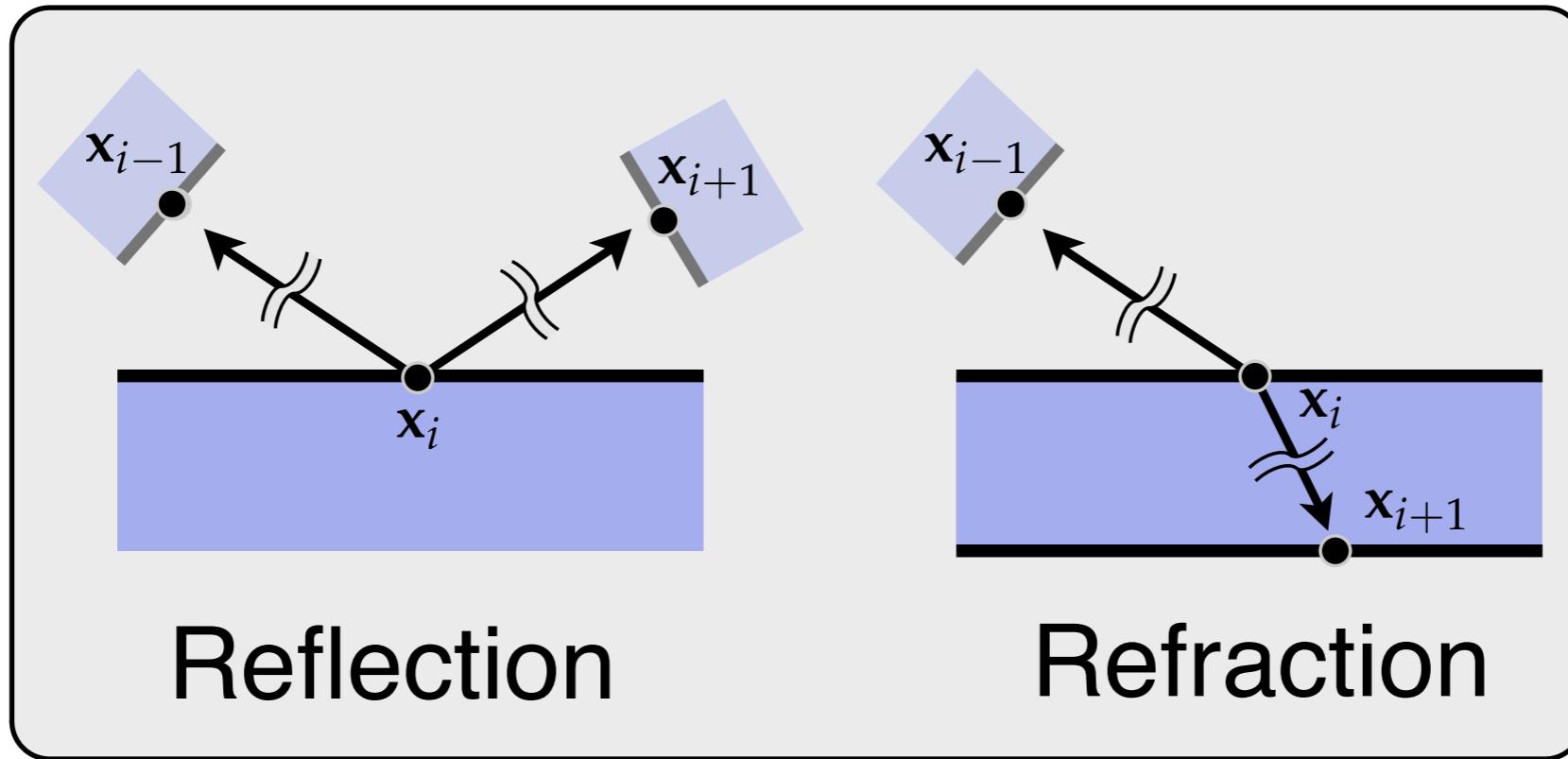
Sensor

The set of paths undergoing specular reflection or refraction is *lower in dimension* than the entire path space.

Mirror

$$x_2 = \frac{1}{2}(x_1 + x_3)$$

# More formally



**Express as constraint:**

$$c_i(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}) = 0$$

**Set satisfying all constraints:**

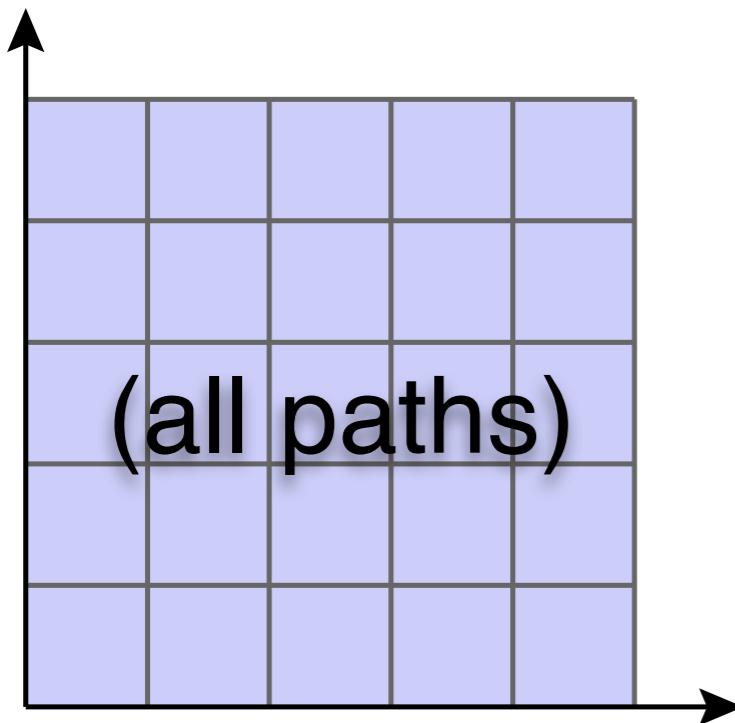
$$\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n \in \Omega \mid C(\mathbf{x}_1, \dots, \mathbf{x}_n) = 0\}$$

# More formally

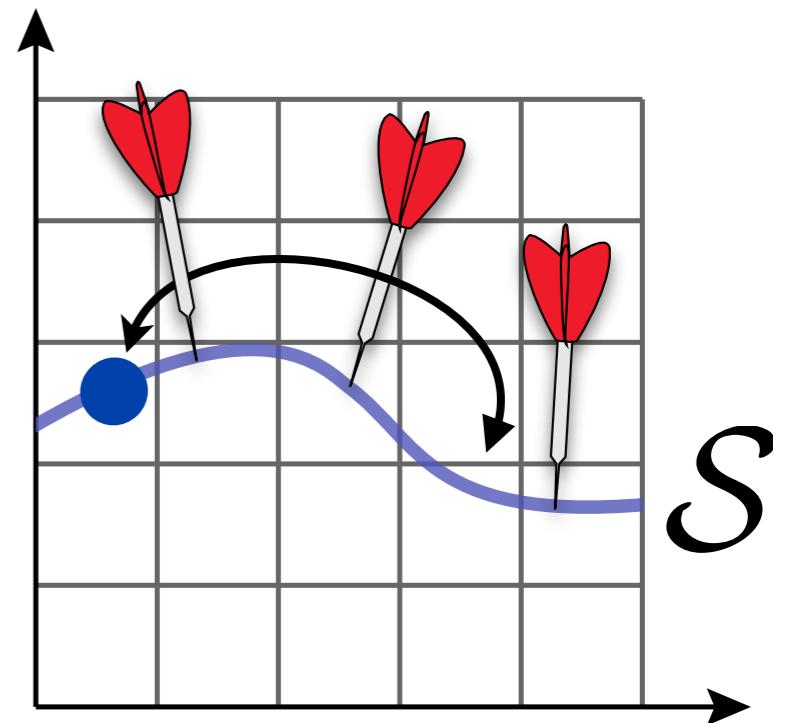
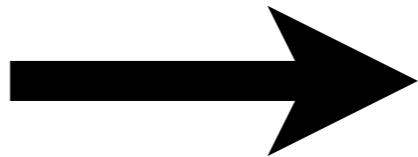
---

Set satisfying all constraints:

$$\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n \in \Omega \mid C(\mathbf{x}_1, \dots, \mathbf{x}_n) = 0\}$$



$$C(\mathbf{x}_1, \dots, \mathbf{x}_n) = 0$$

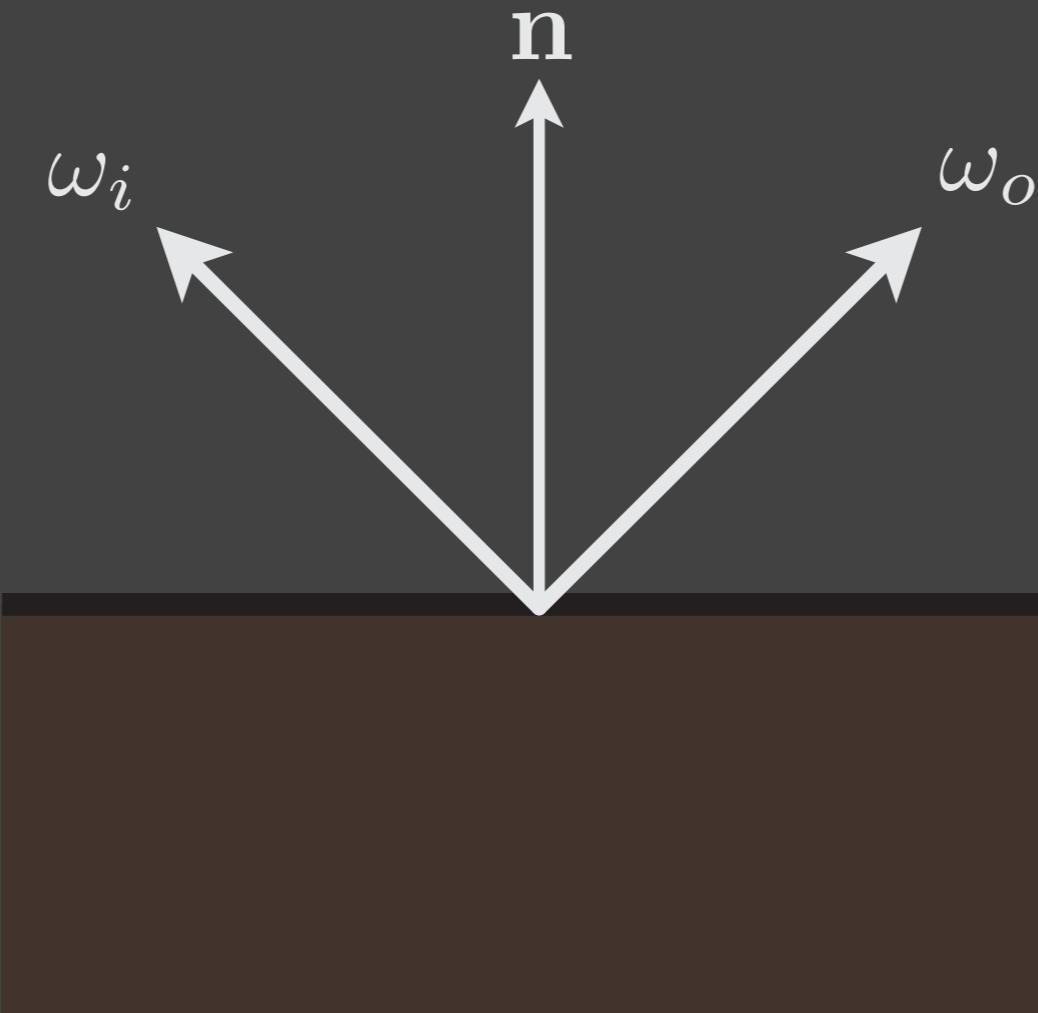


# Manifold Exploration



*Manifold Exploration: A Markov Chain Monte Carlo technique  
for rendering scenes with difficult specular transport*  
[Jakob and Marschner 2012]

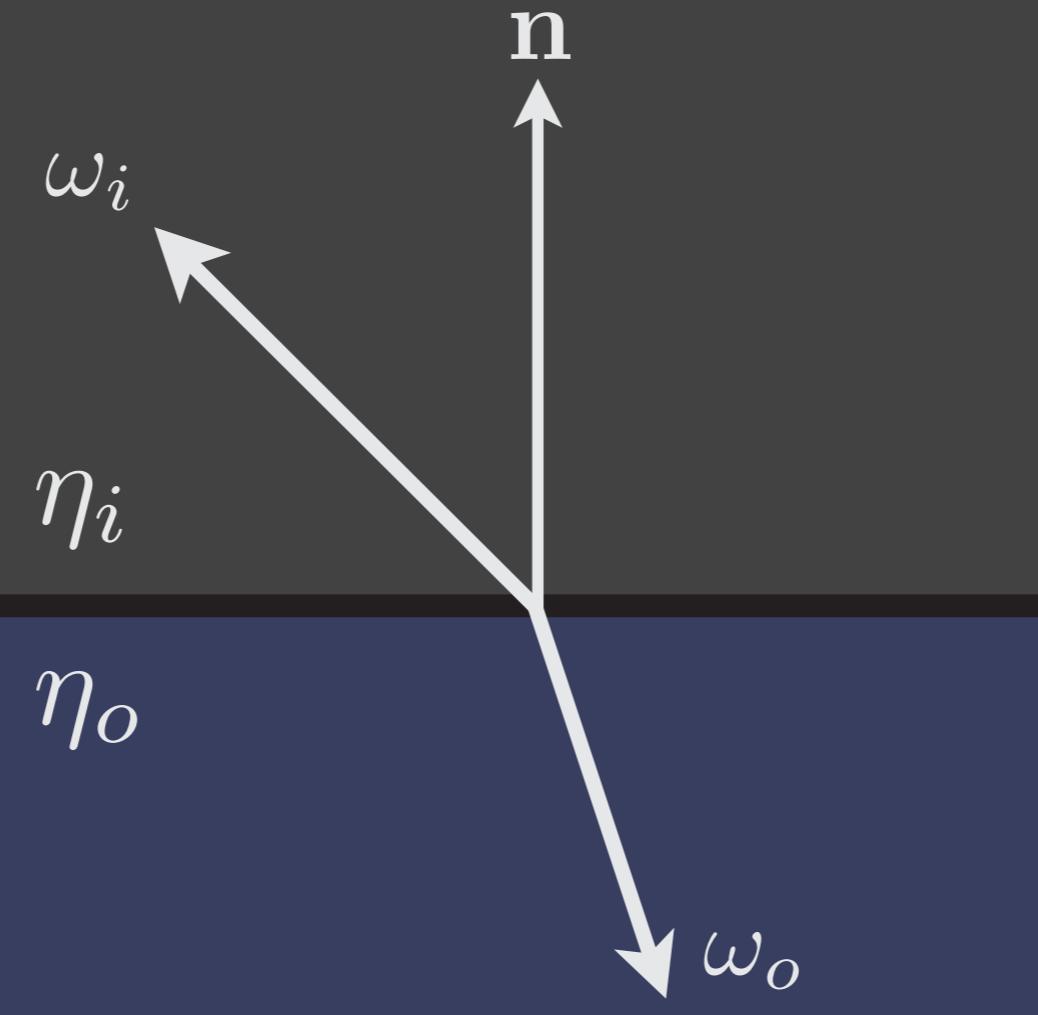
# Specular path constraint



Constraint:  $\mathbf{n} = H(\omega_i, \omega_o)$

$$\text{where } H(\omega_i, \omega_o) := \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|}$$

# Specular path constraint

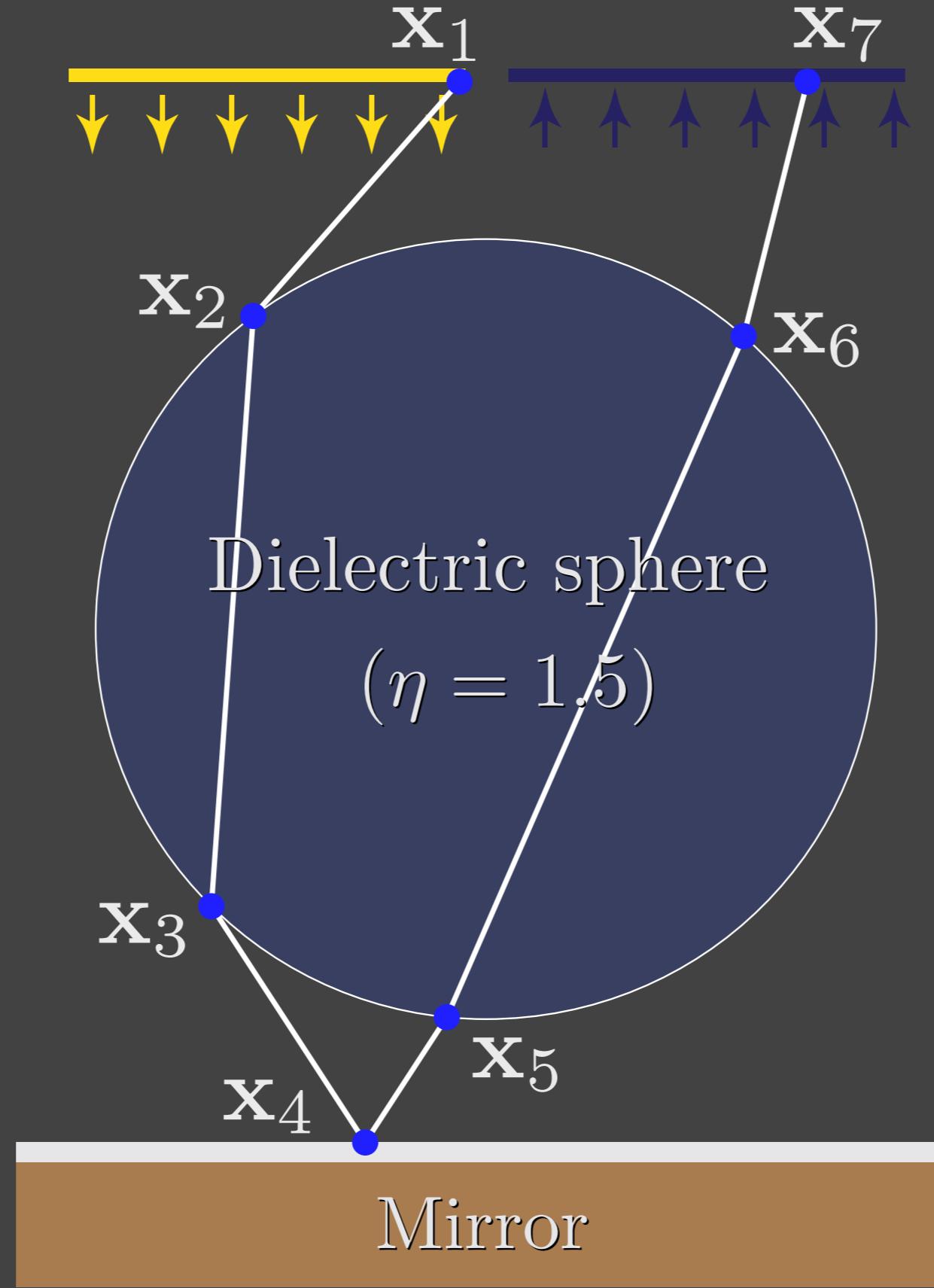


Constraint:  $\mathbf{n} = H(\omega_i, \omega_o)$

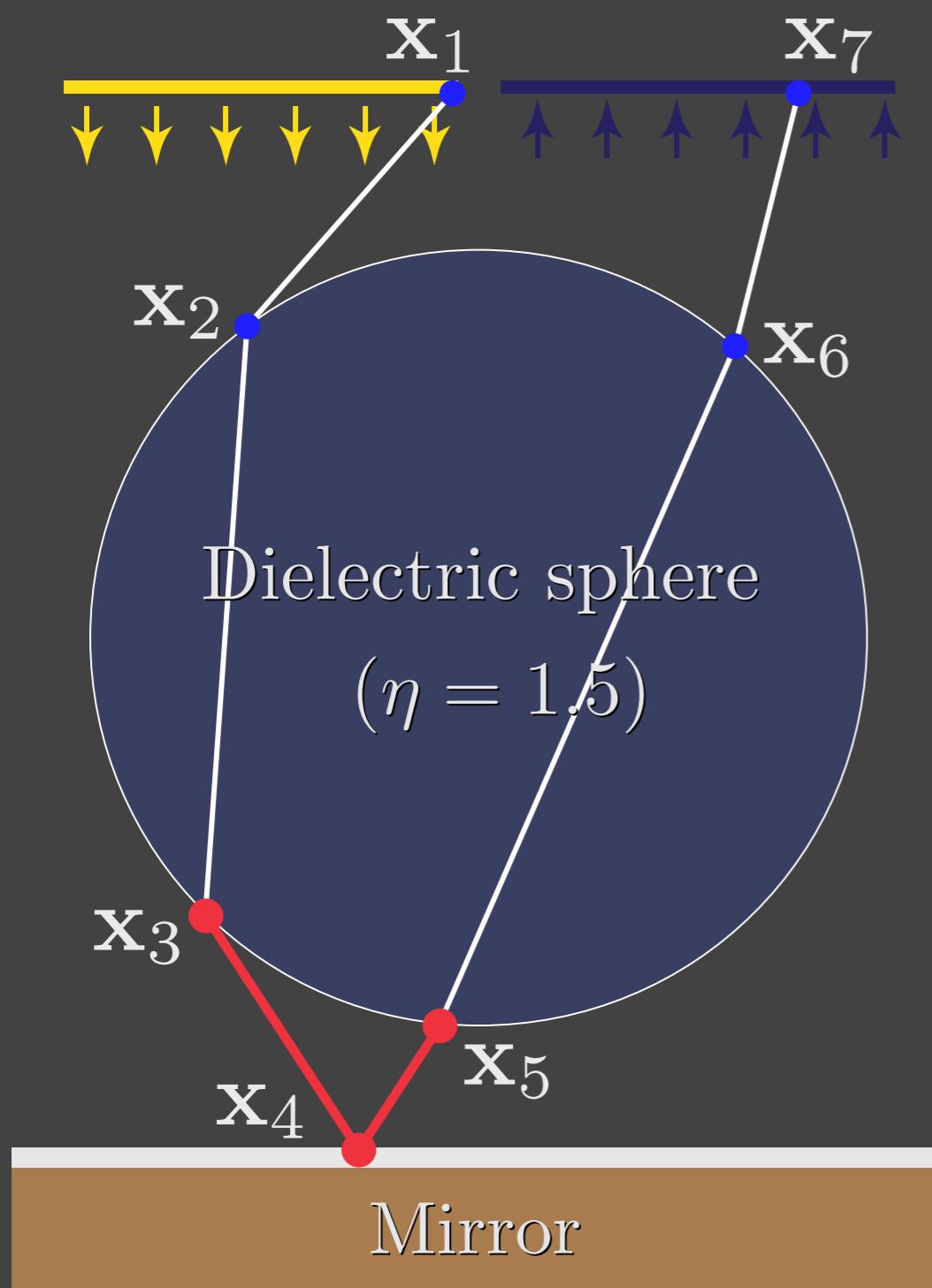
$$\text{where } H(\omega_i, \omega_o) := \frac{\eta_i \omega_i + \eta_o \omega_o}{\|\eta_i \omega_i + \eta_o \omega_o\|}$$

Generalized half vector [Sommerfeld and Runge 1911]

# Manifold walks



# Manifold walks

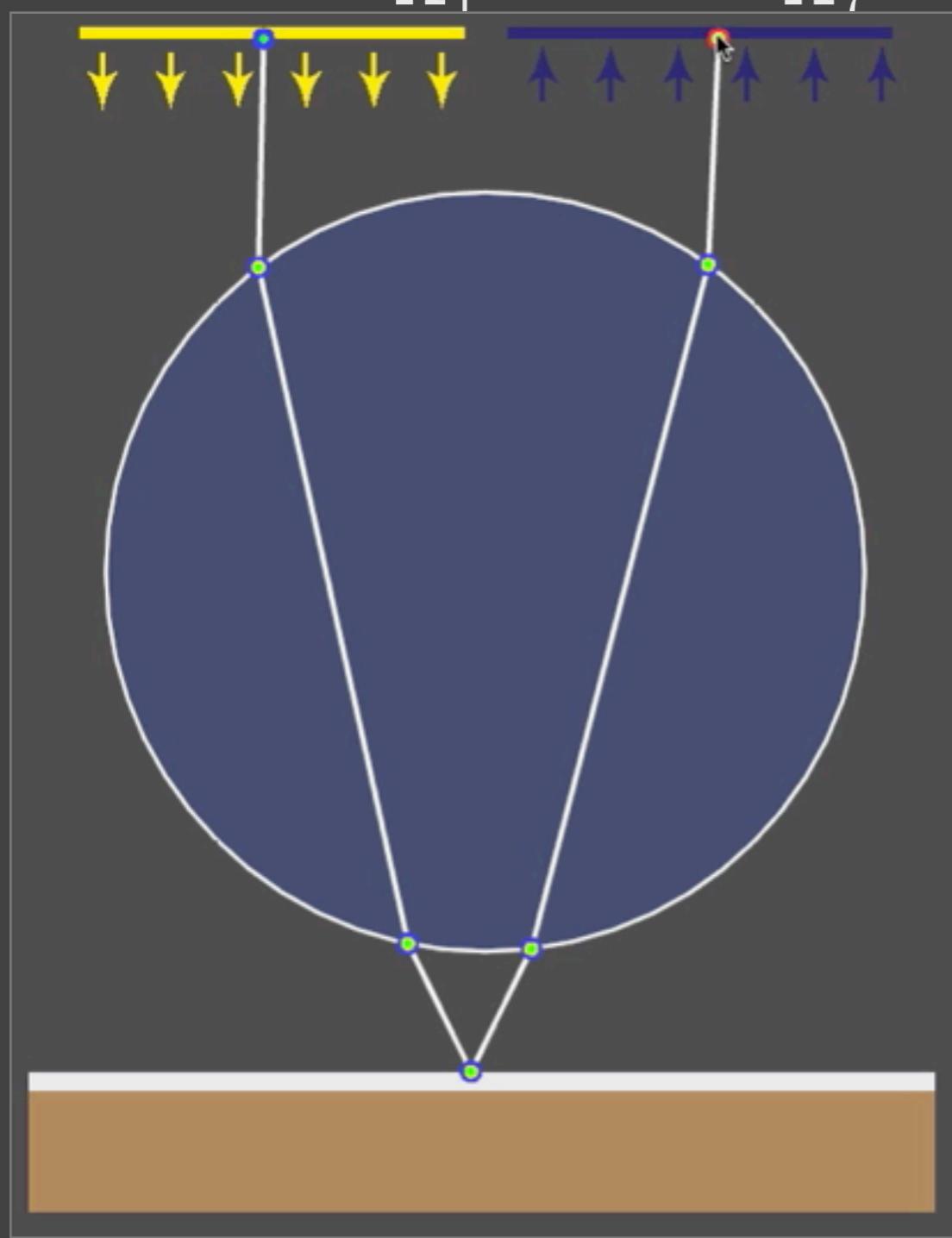


$$\nabla C = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 \end{bmatrix}$$

The matrix  $\nabla C$  is a 7x7 grid. The columns are labeled  $\mathbf{x}_1$  through  $\mathbf{x}_7$ . The matrix is partitioned into three regions:  $B_1$  (leftmost column),  $A$  (central 3x3 block), and  $B_7$  (rightmost column). The matrix contains the following values:

Region	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
$B_1$	Gray	Gray	Gray	Red	Red	Red	Gray
$A$	Gray						
$B_7$	Gray						

# Manifold walks



$$\nabla C = \begin{bmatrix} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 \\ & \vdots \\ & B_1 & & & & & & B_7 \end{bmatrix}$$

The matrix  $\nabla C$  is a 7x7 block matrix. It consists of seven diagonal blocks labeled  $B_1, B_2, B_3, B_4, B_5, B_6, B_7$ . The blocks are arranged such that  $B_1$  is at the top-left,  $B_7$  is at the bottom-right, and the other blocks  $B_2, B_3, B_4, B_5, B_6$  are in the middle. The blocks  $B_1, B_2, B_3, B_4, B_5, B_6, B_7$  are all identical 2x2 identity matrices. The labels  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7$  are positioned above each column of the matrix.

$$\frac{\partial \begin{bmatrix} \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_6 \end{bmatrix}}{\partial \mathbf{x}_7} = -A^{-1}B_7$$

# Manifold walks

Only works for *infinitesimal* displacements

1. New path is not a valid specular path anymore
2. Proposed vertex positions not on the surface!

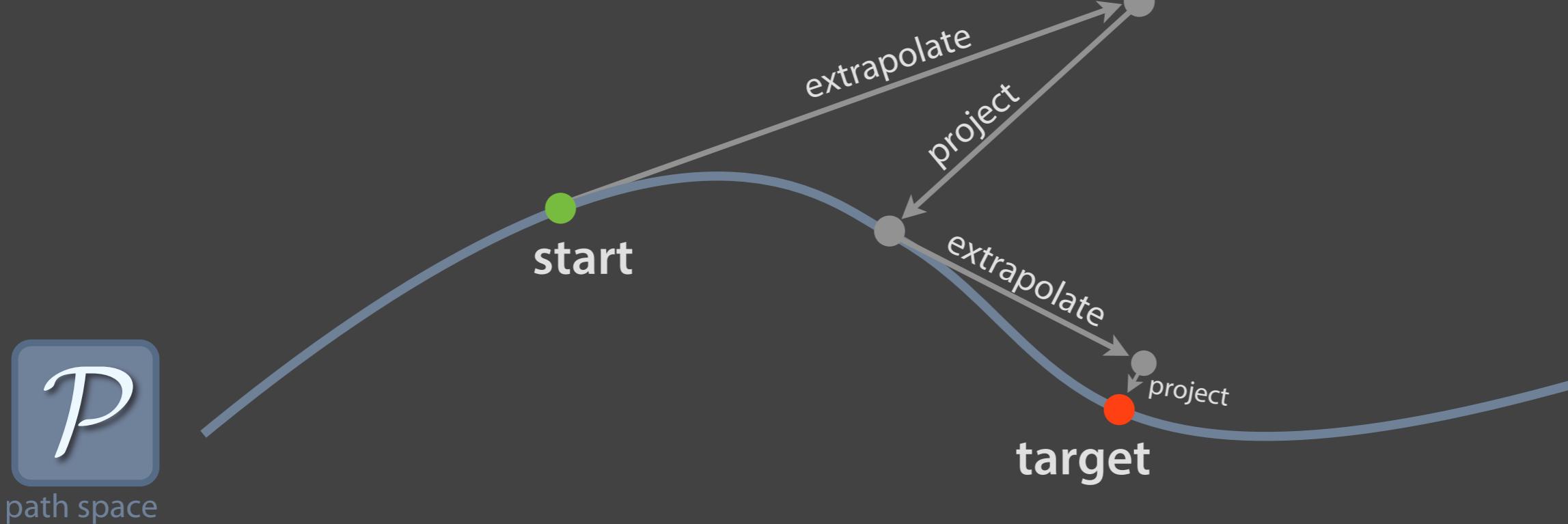
Need a way to stay on the manifold

# Manifold walk algorithm

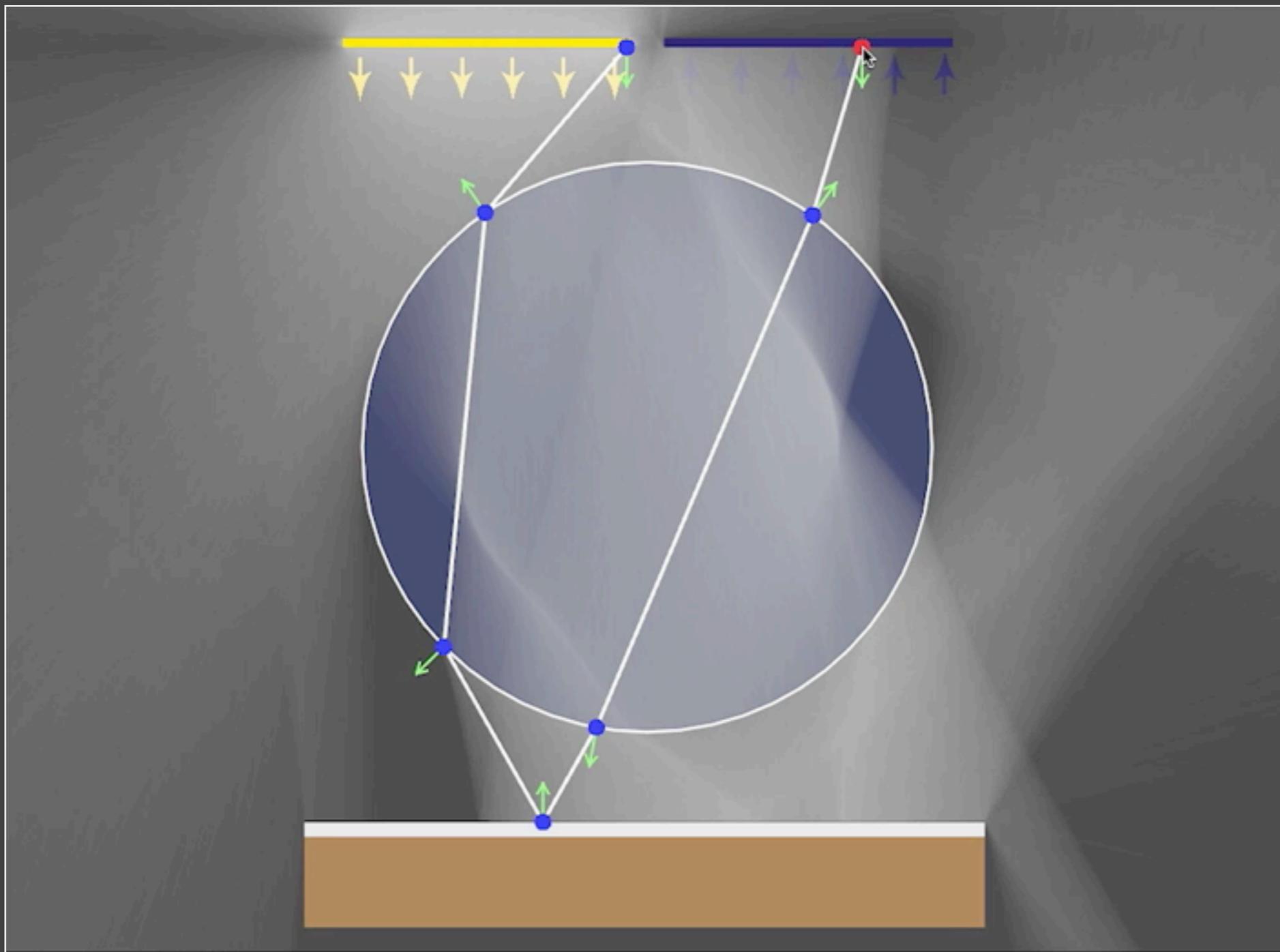
**Basic idea:**

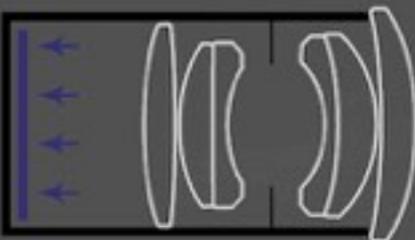
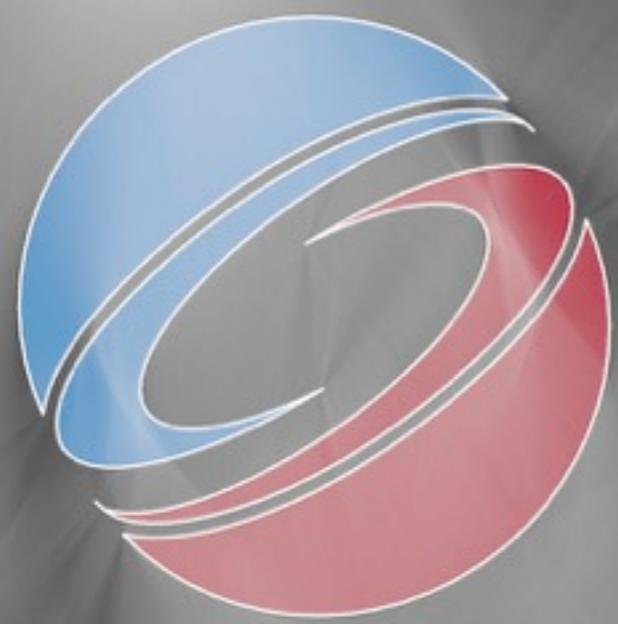
**while** not there yet:

1. **EXTRAPOLATE**: Perturb vertices using manifold tangents
2. **PROJECT**: re-trace extrapolated path

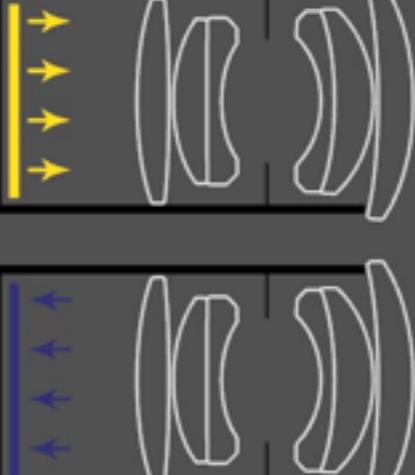
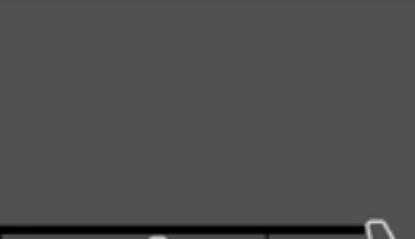
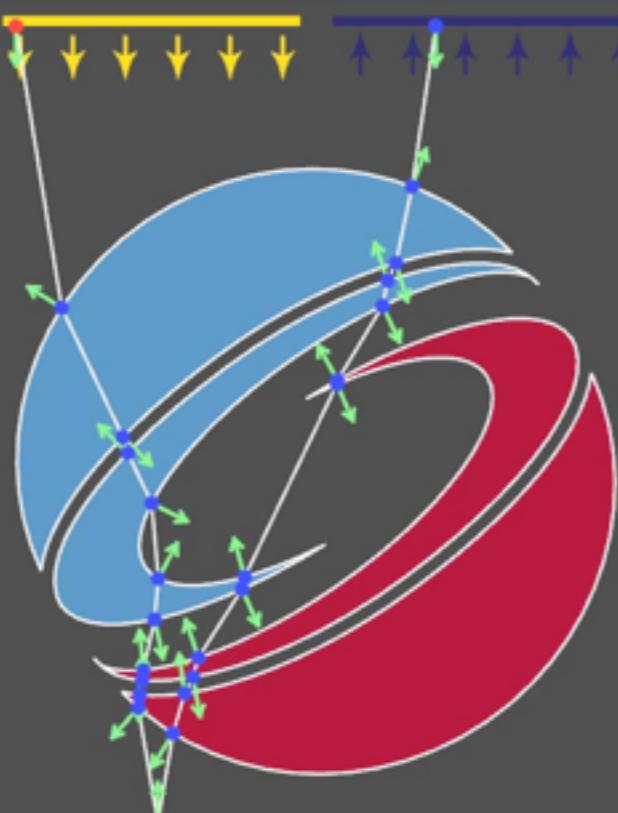


# Manifold walk algorithm — Demo



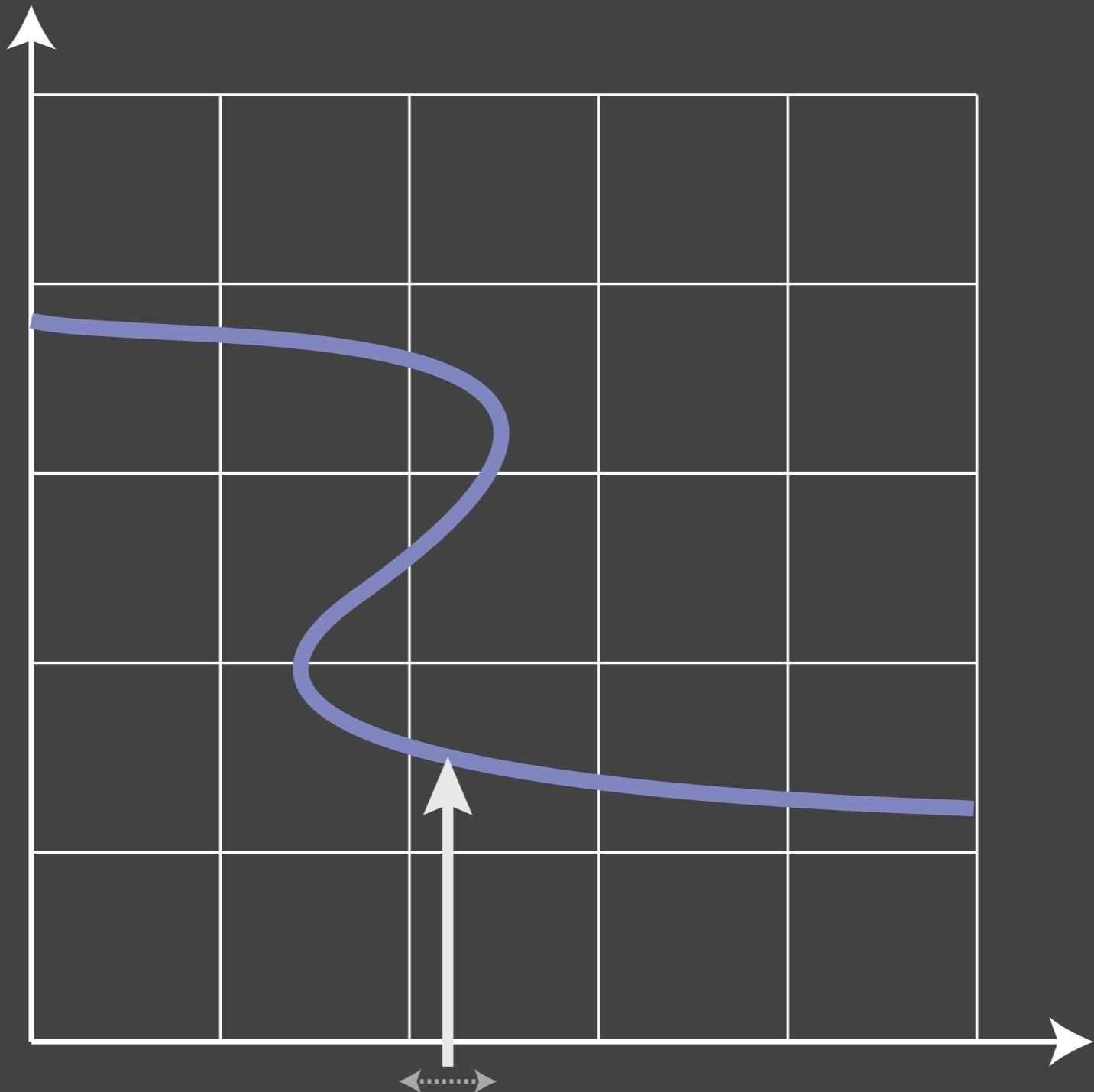


Diffuse surface

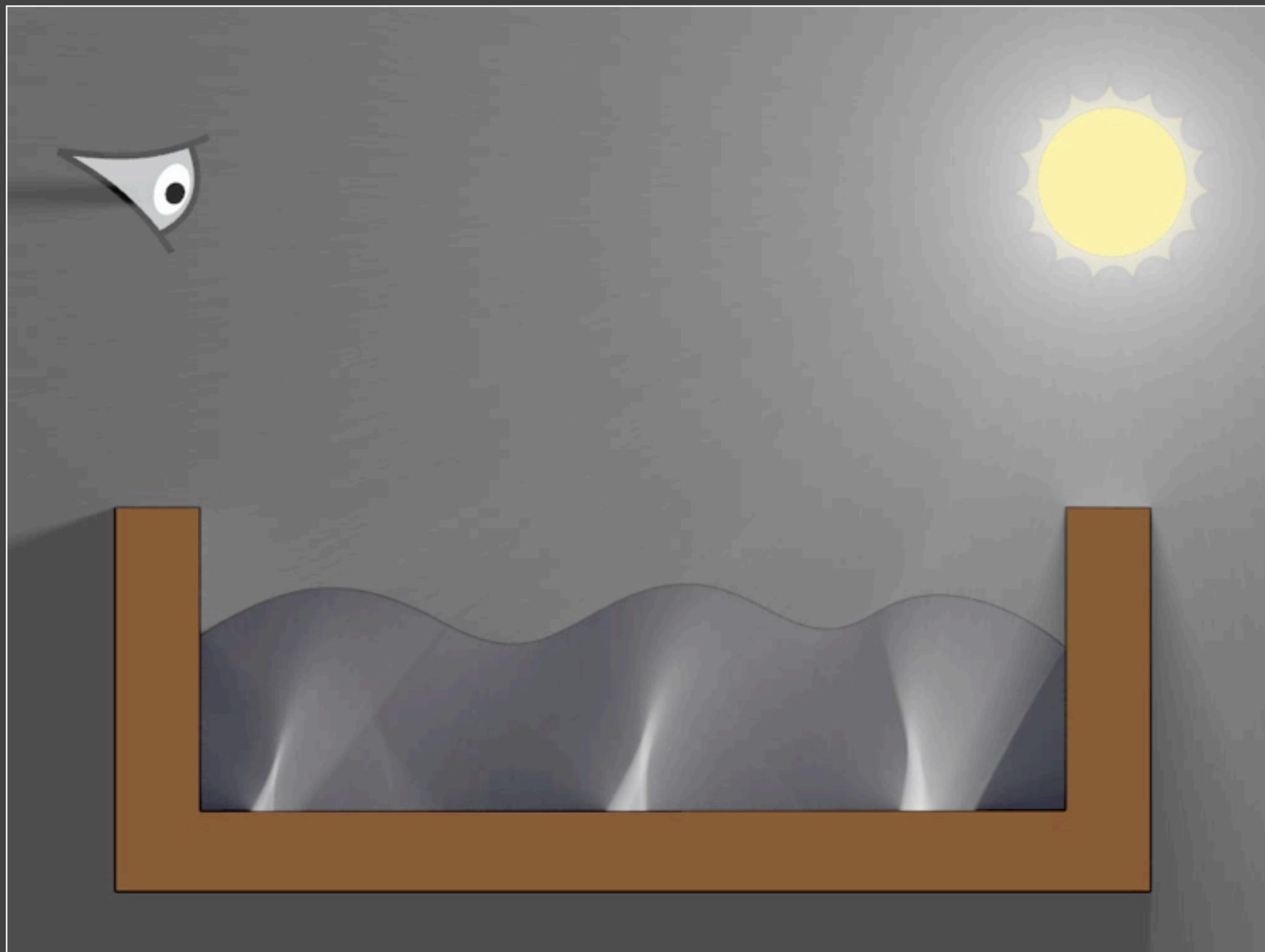


Diffuse surface

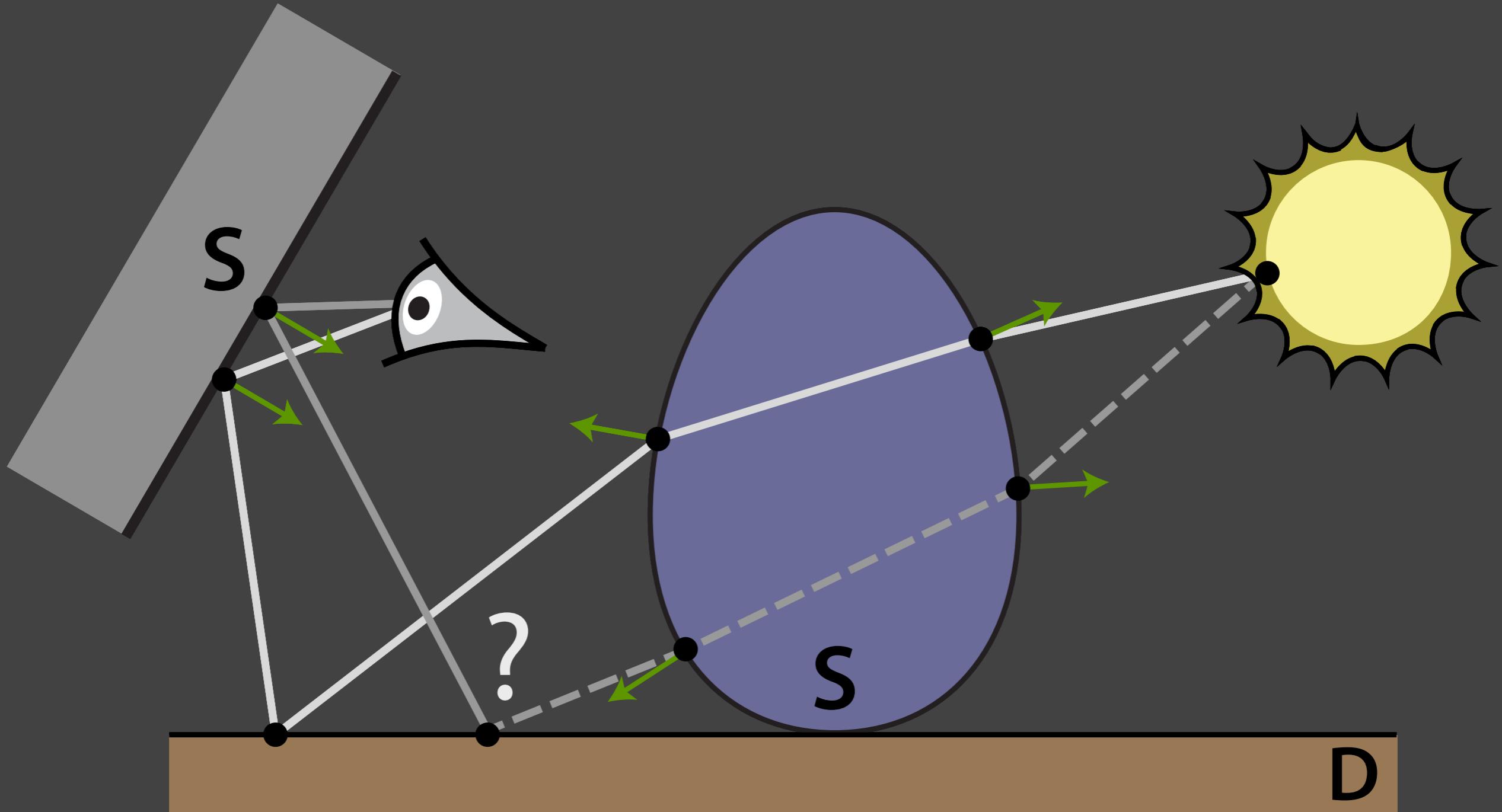
# Parameterization failures



# Parameterization failures

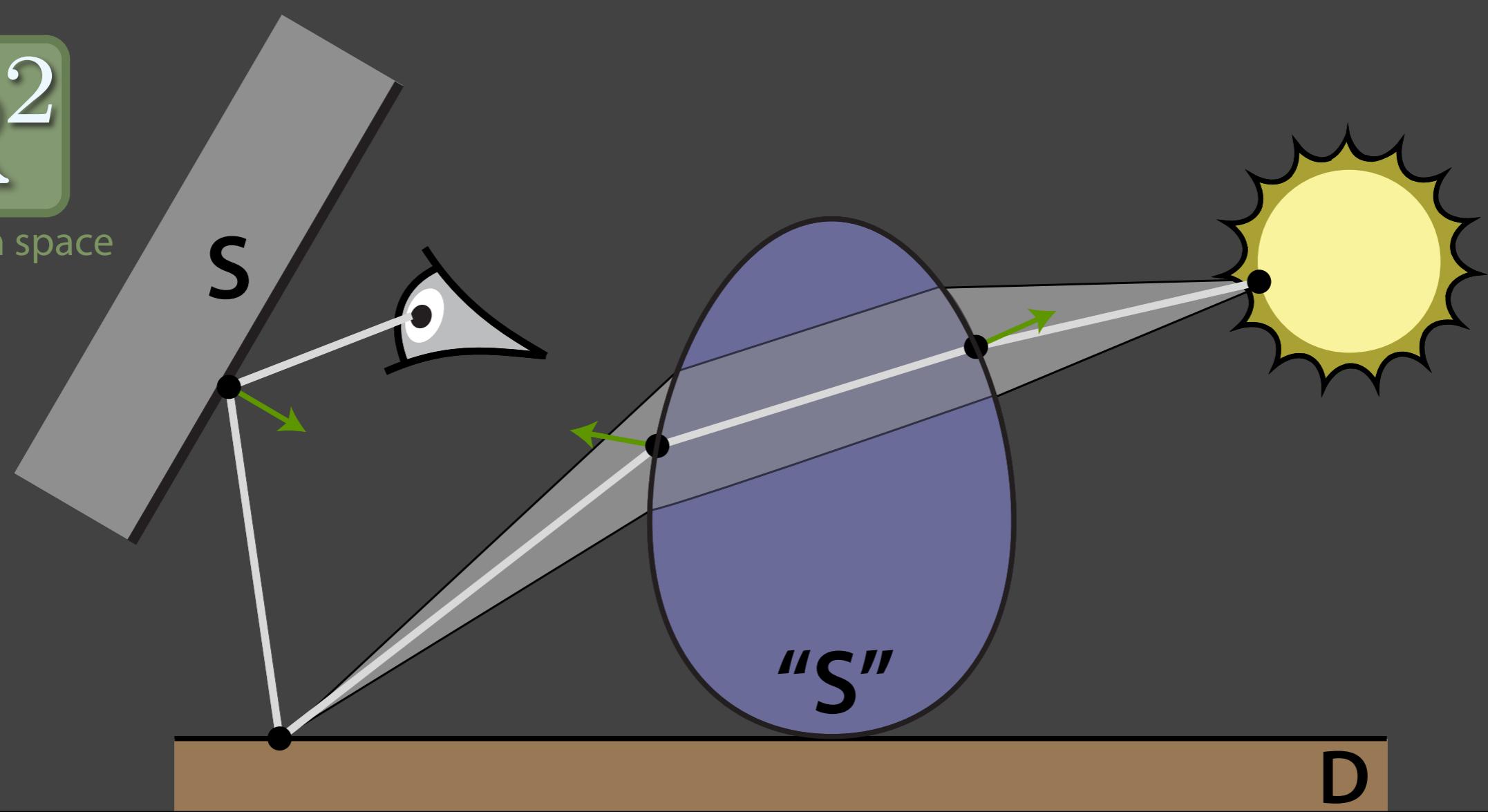


# Manifold perturbation

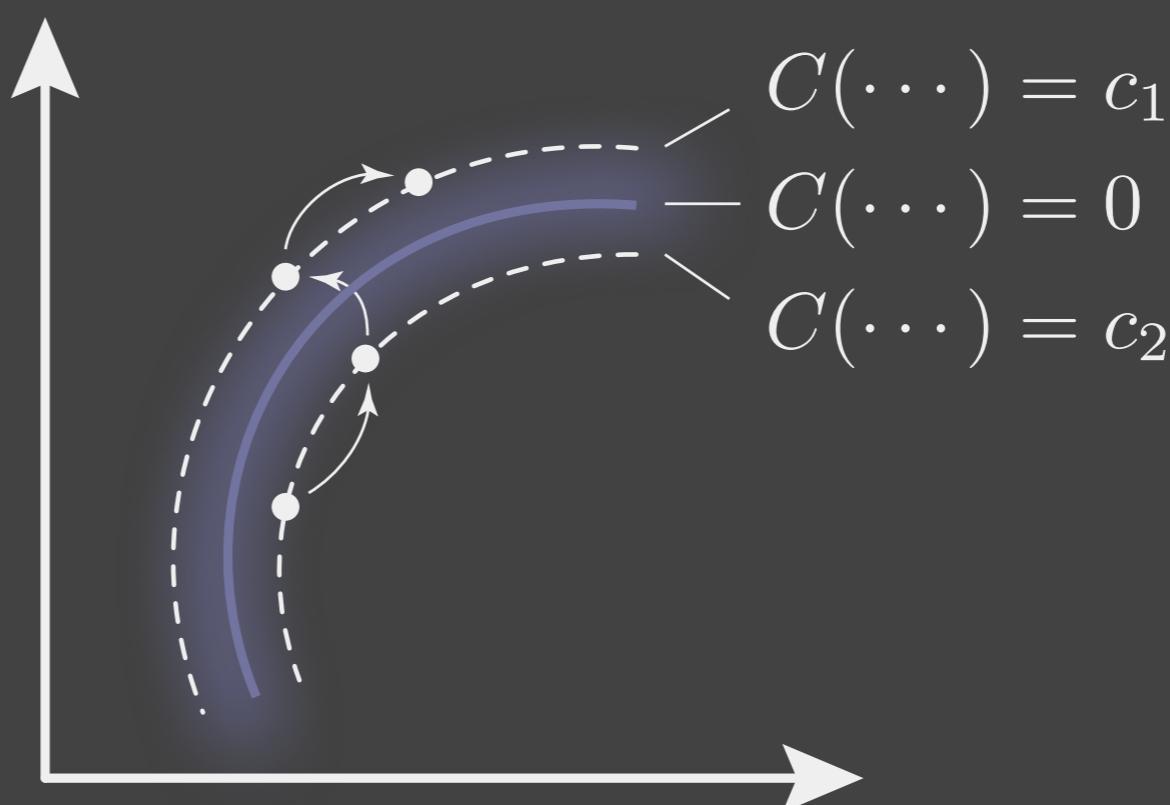


$\mathbb{R}^2$ 

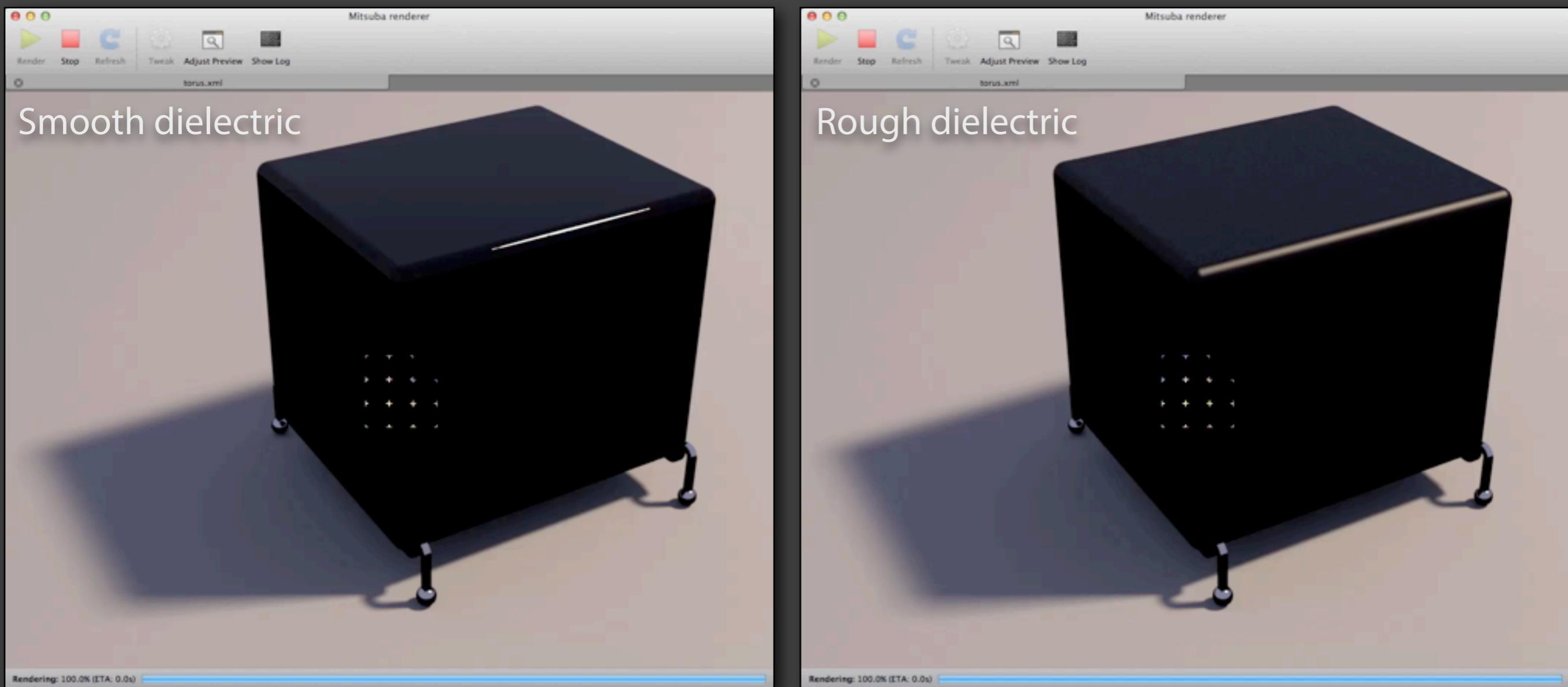
euclidean space

 $\mathcal{P}$ 

path space

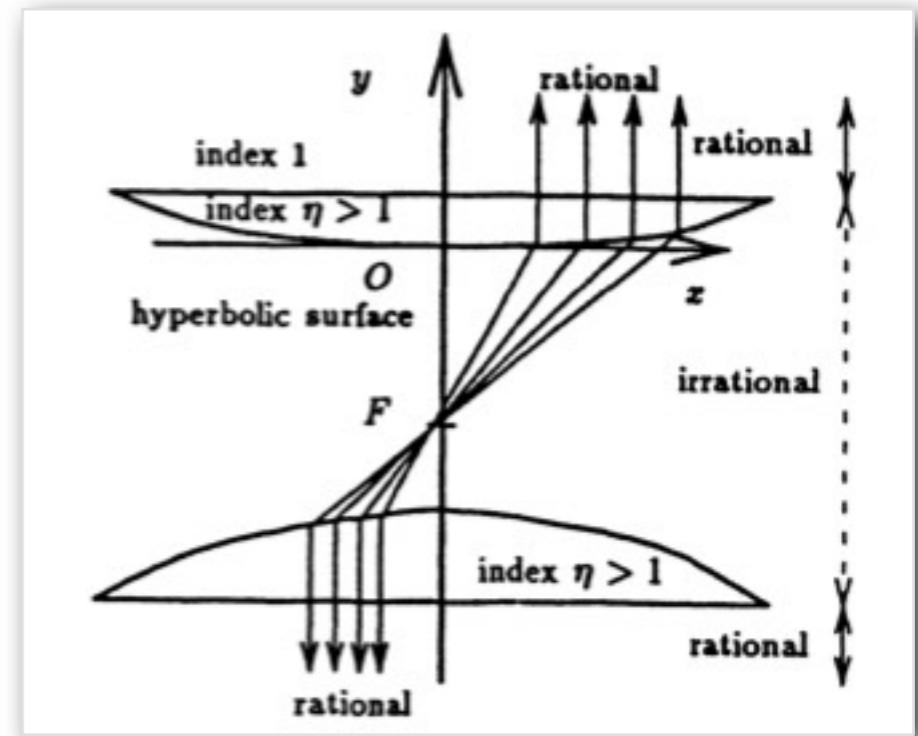
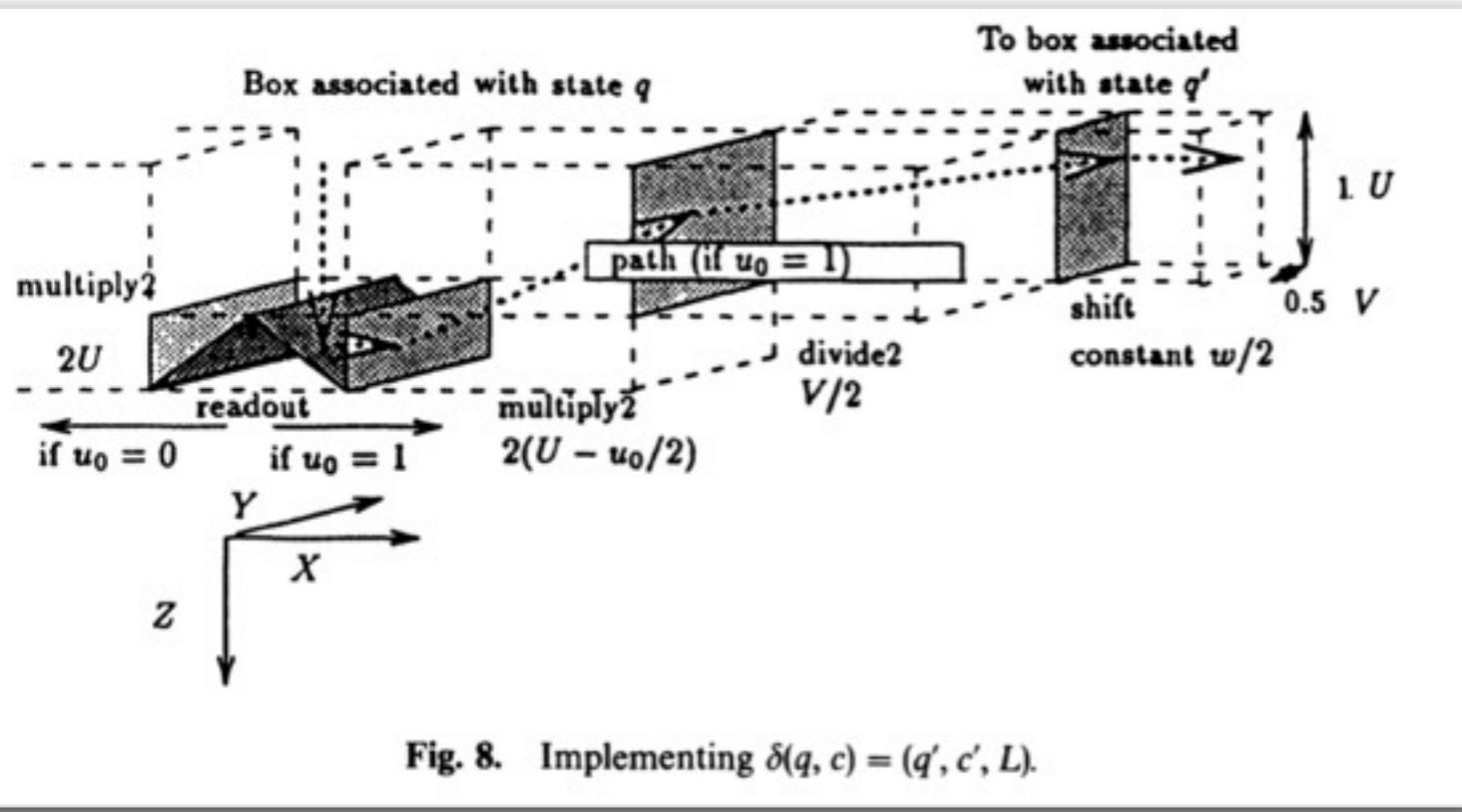


# Manifold Exploration Path Tracing



[15x time-lapse]

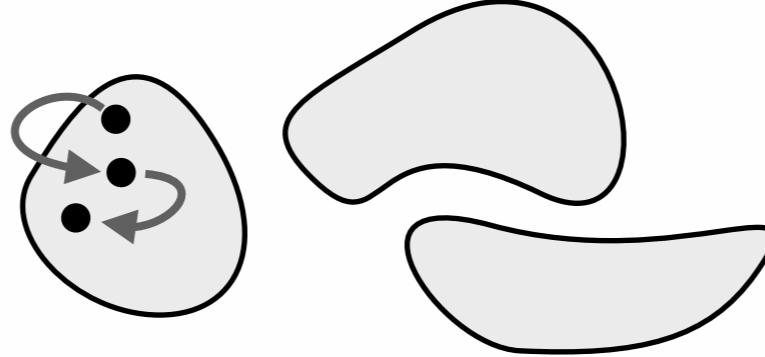
# Rendering is a hard problem



*Computability and Complexity of Ray Tracing* (Reif et al 1990)

# Current issues

---

- **Initialization:** unpredictable convergence (islands)
- A diagram showing three entities. On the left, a circular region contains three black dots connected by curved arrows forming a cycle, representing a local optimum or 'island'. To its right are two separate, irregularly shaped regions, each containing a single black dot, representing other global optima.
- **Stratification:** how to use “good” random numbers?
- **Realization:** algorithms are more challenging to implement and debug
- **Generality:** Need better mutations that work well for arbitrary input