CS 87/187, Spring 2016

RENDERING ALGORITHMS

Markov Chain Monte Carlo

Prof. Wojciech Jarosz

wojciech.k.jarosz@dartmouth.edu

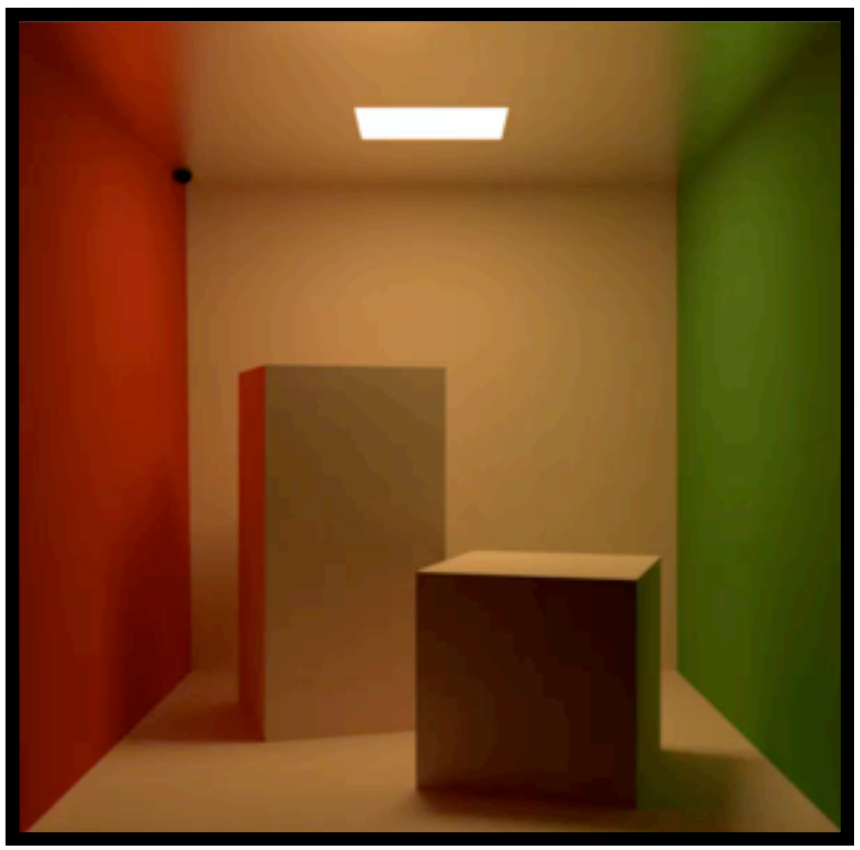
(most slides courtesy of Dr. Wenzel Jakob)

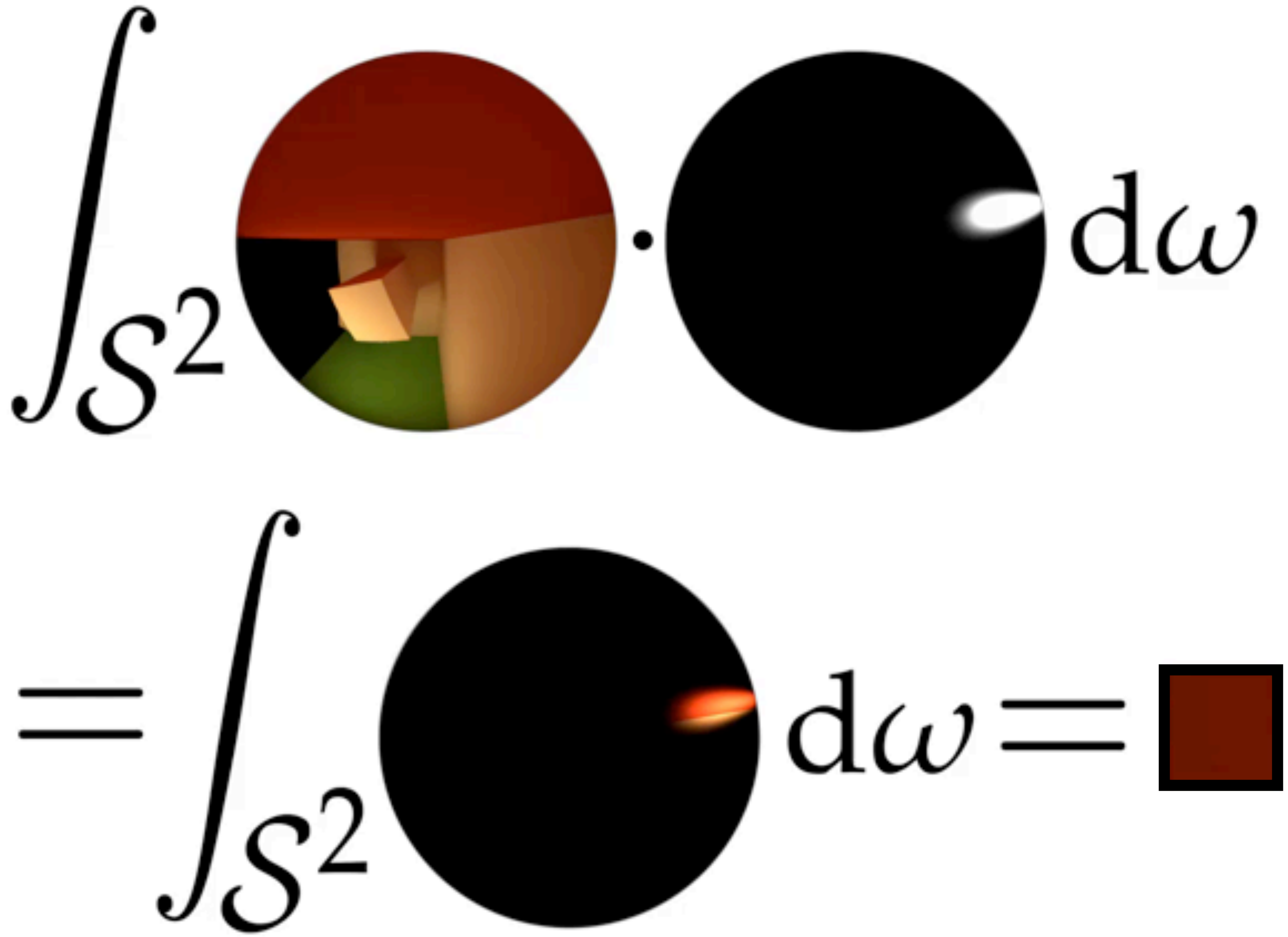
Dartmouth

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

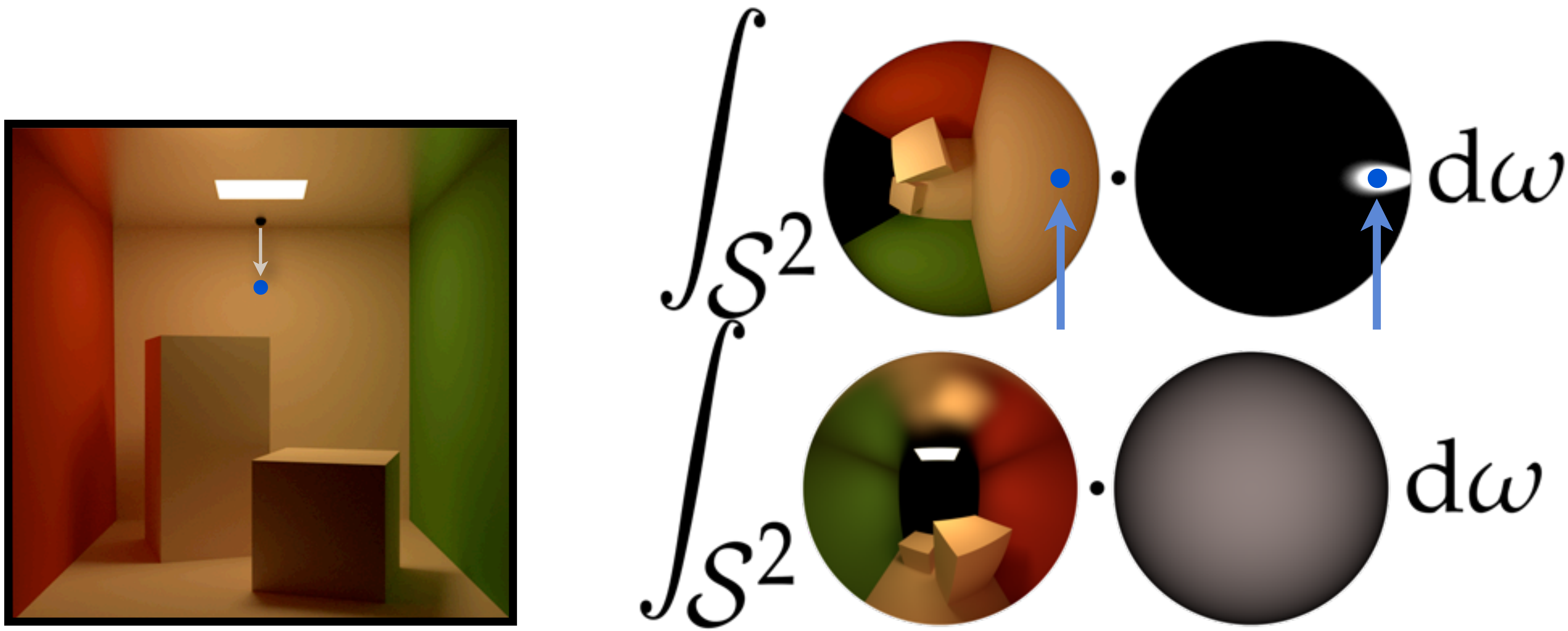
Path tracing review





Final pixel color

Path tracing review

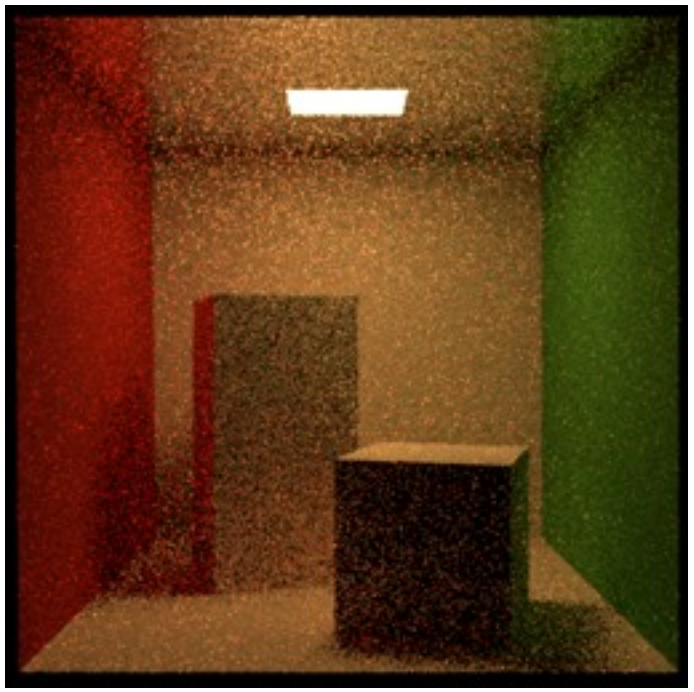


Path tracing pseudocode

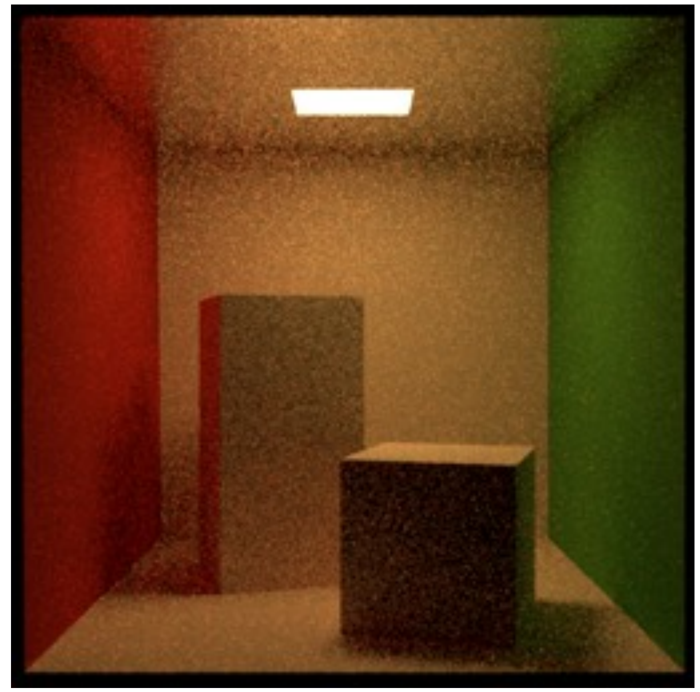
$\langle L_o(\mathbf{x}, \omega_o) \rangle :=$

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

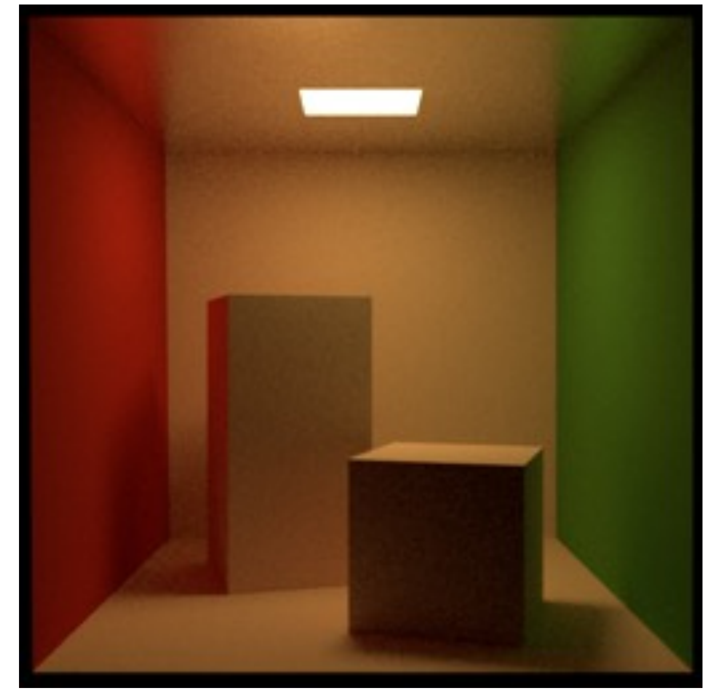
Path tracing convergence



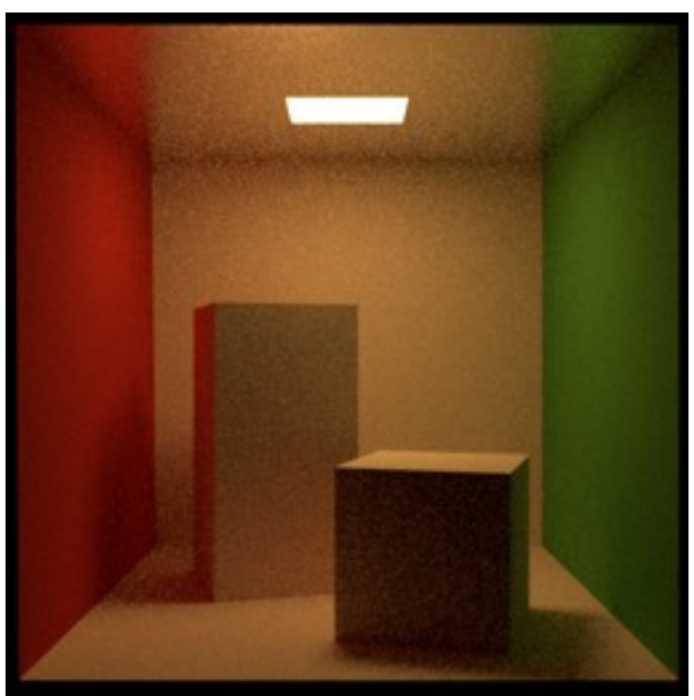
1 sample



4 samples

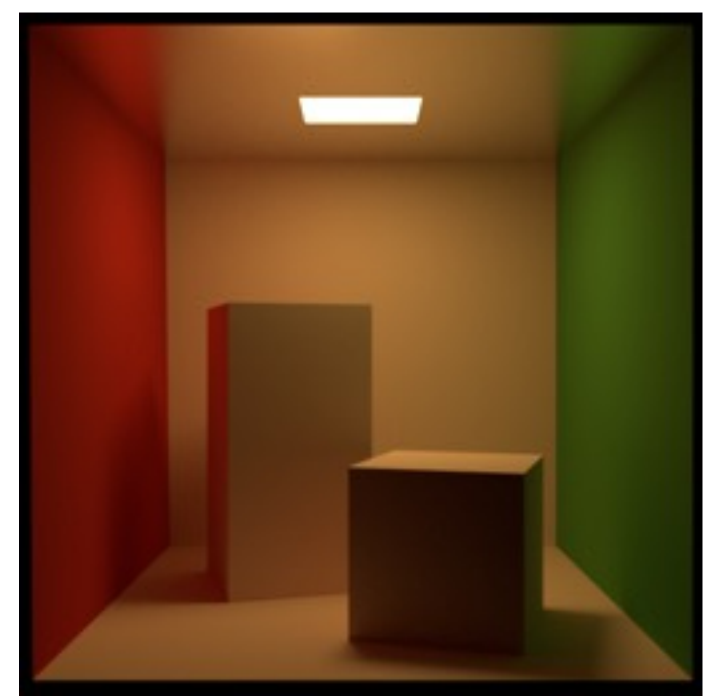


16 samples



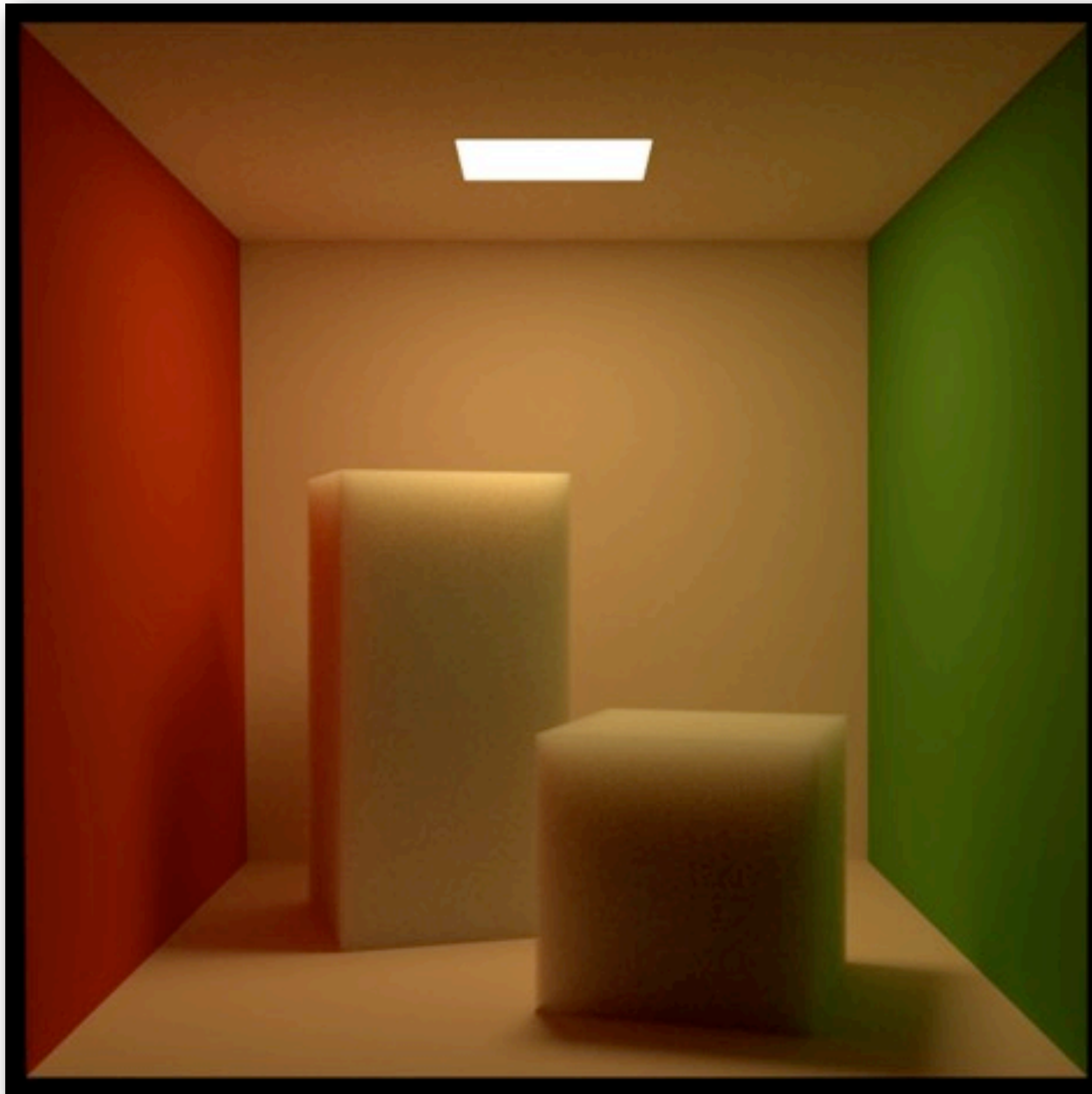
64 samples

256 samples



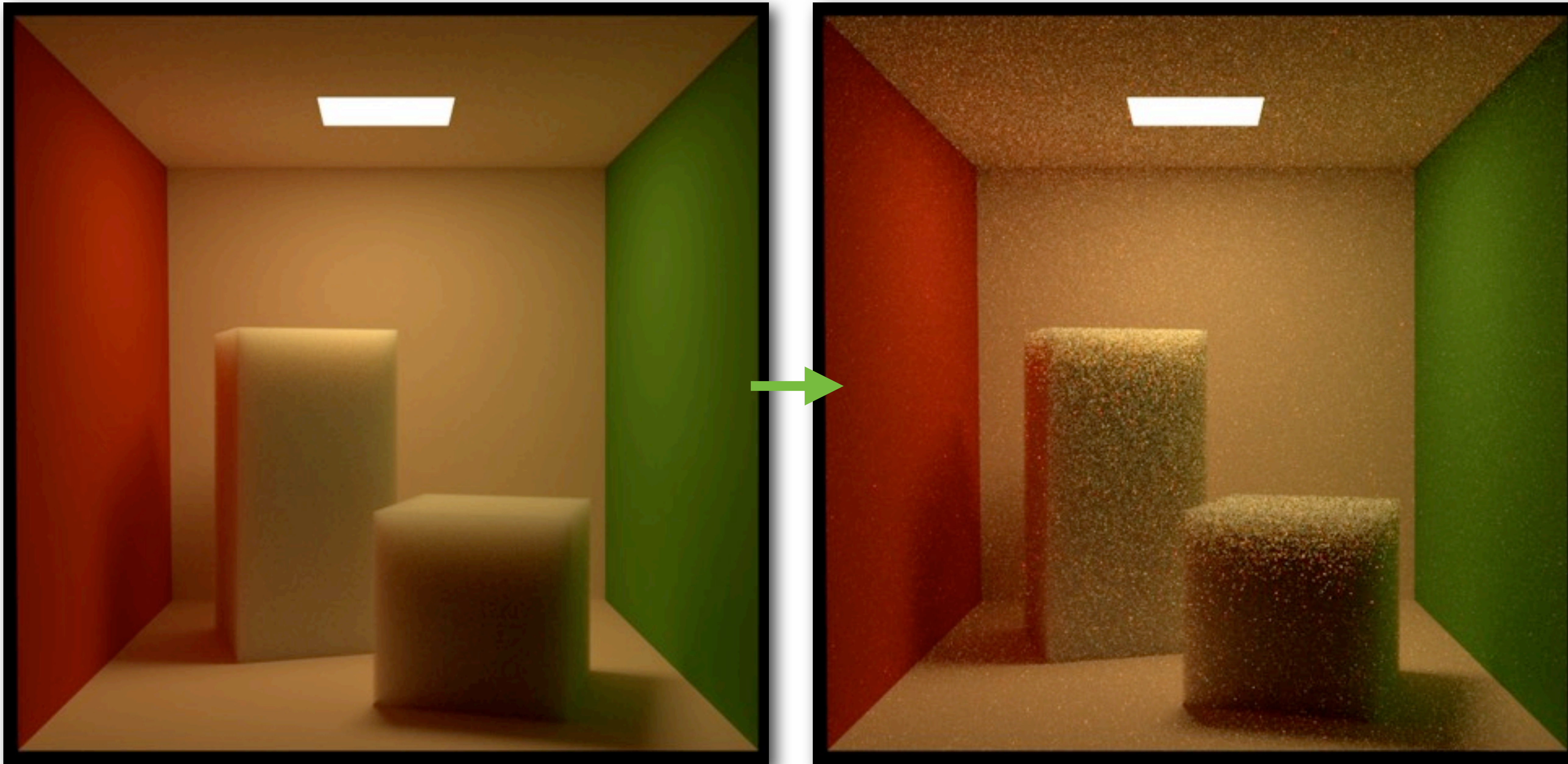
1024 samples

A simple example



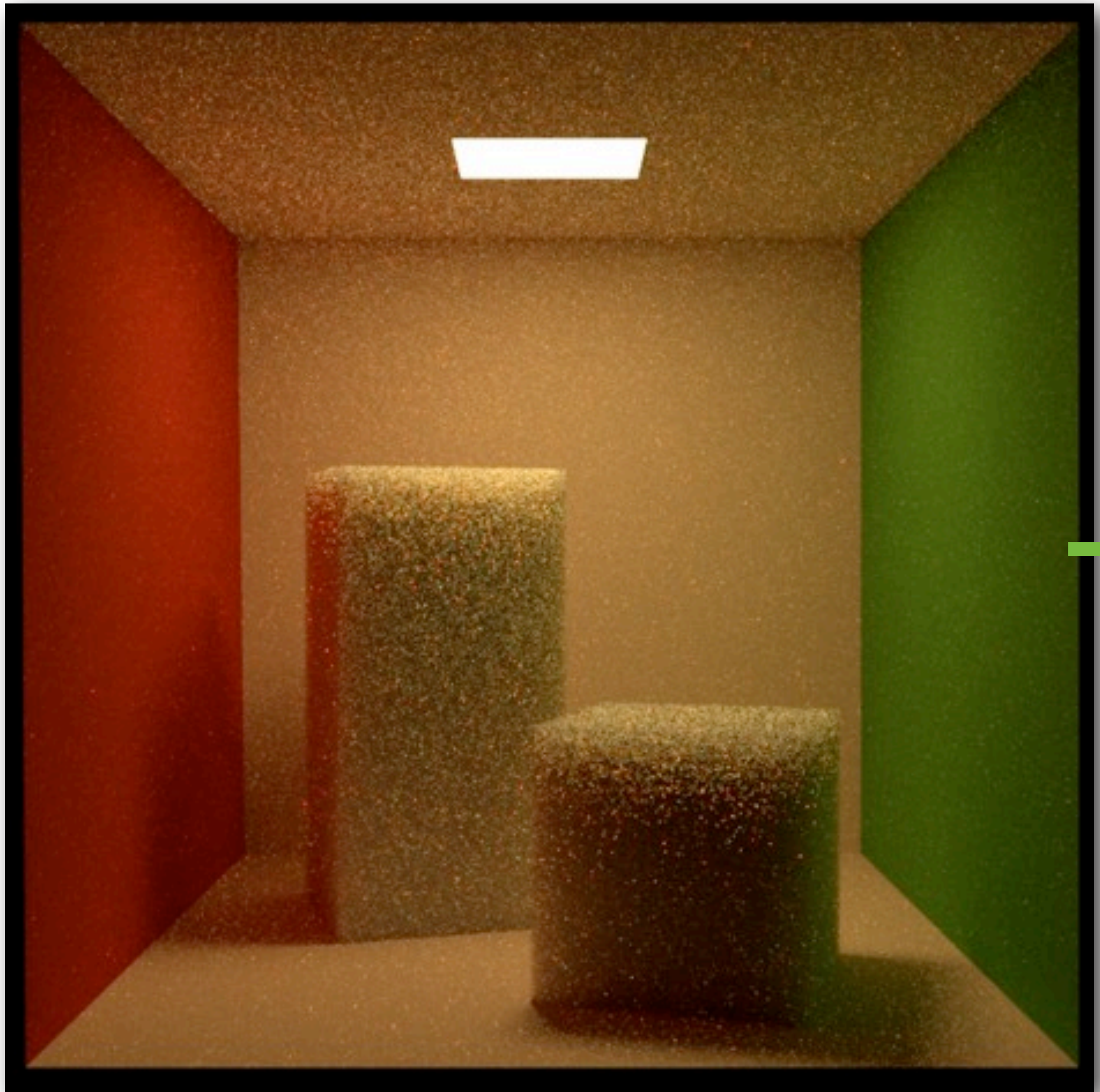
Cornell box with 2 random media, 256 spp.

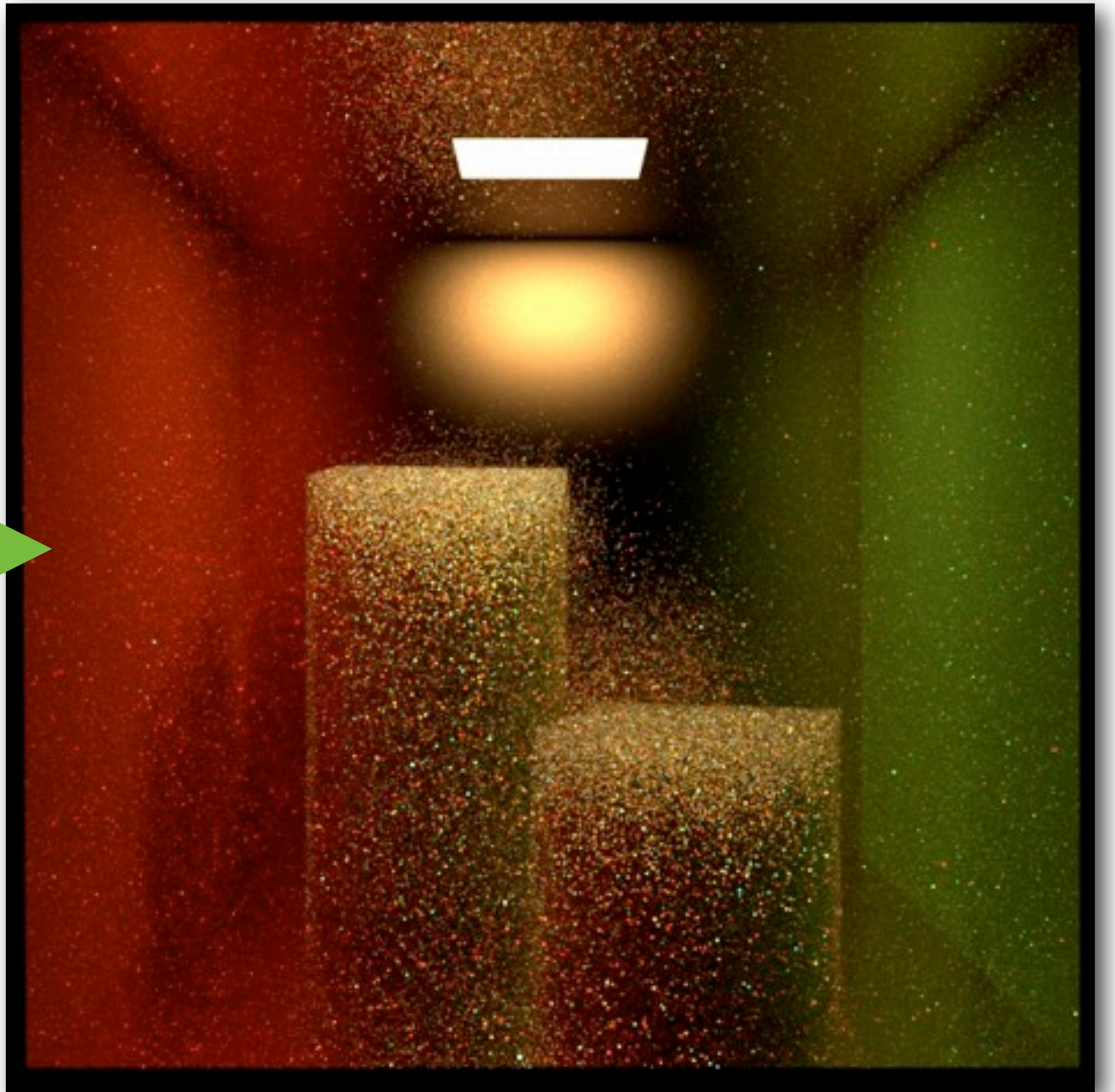
+ dielectric boundaries



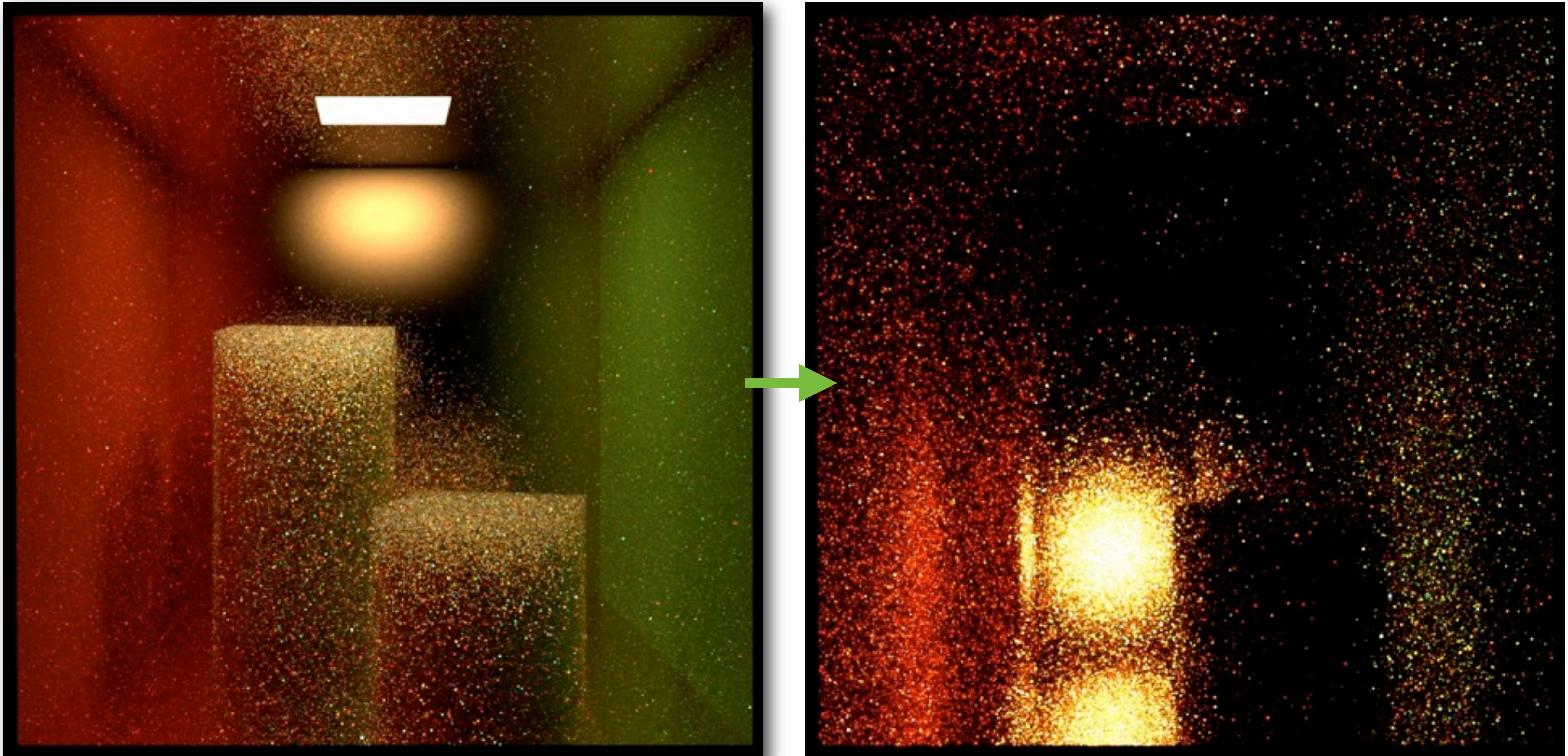
$$\eta = 1 \qquad\qquad \eta = 1.00001$$

+ glossy materials





+ complex lighting



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

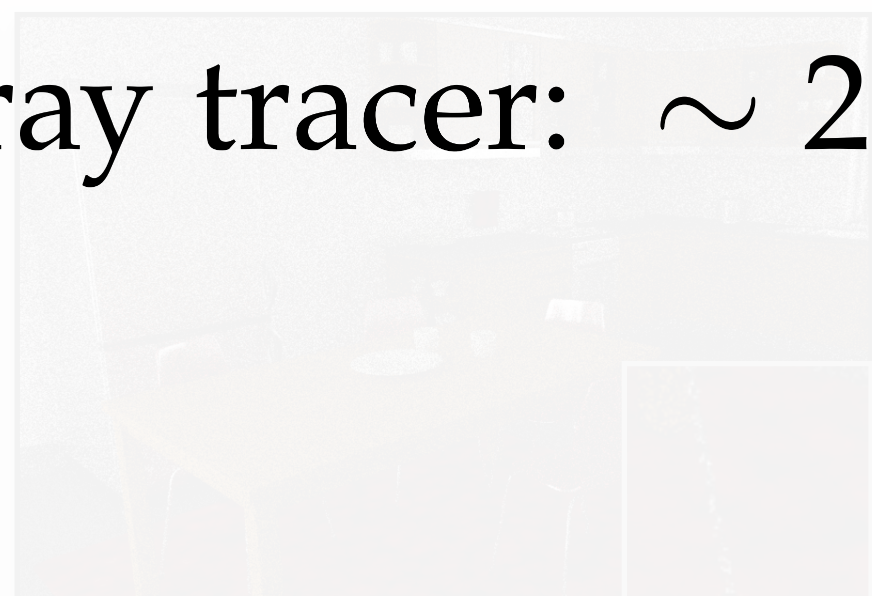
$$\text{Fastest GPU ray tracer: } \sim 2 \cdot 10^{8} \text{ / second}$$

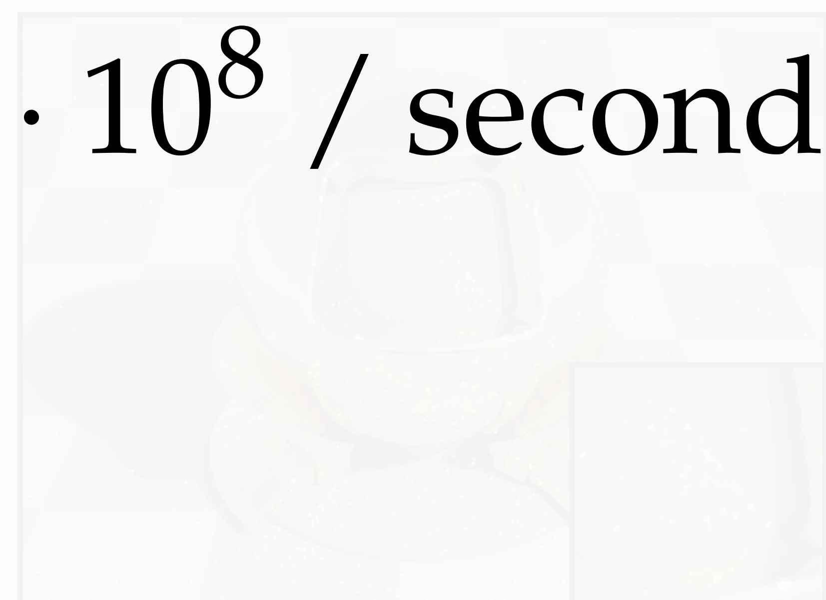

Path tracing


Path tracing

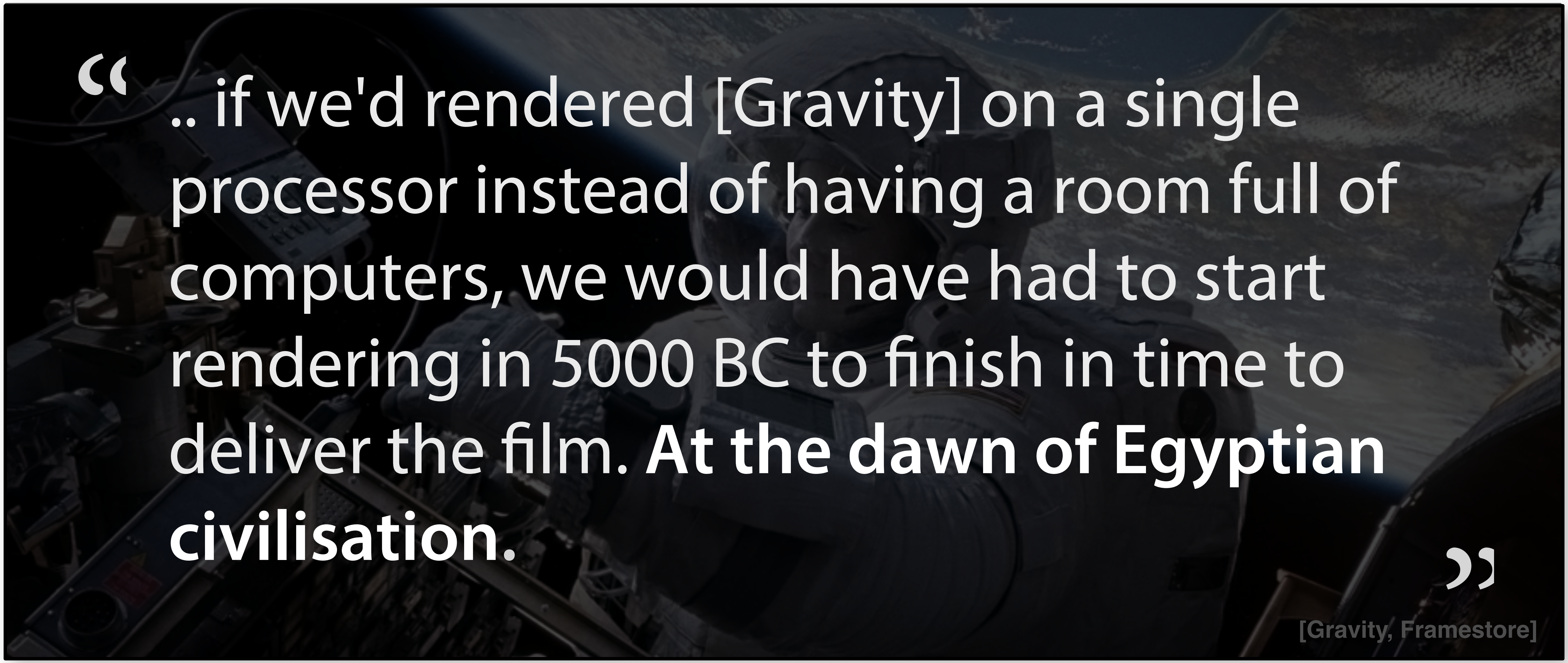

Volumetric path tracing [Georgiev et al.]


Bidirectional path tracing


Progressive photon mapping


Natural sun & sky illumination

Costs of rendering



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)

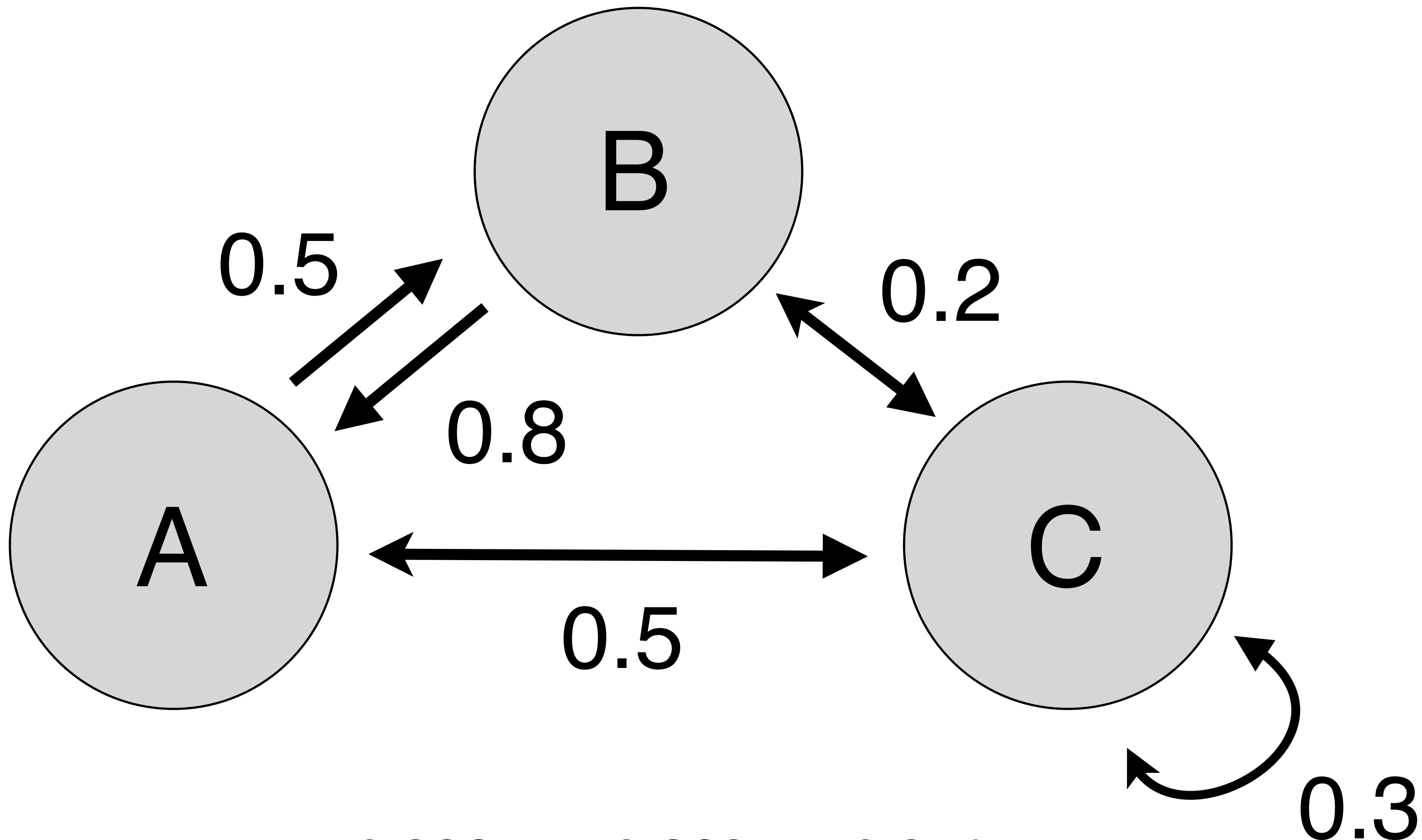
2. Markov Chain Monte Carlo ("MCMC")

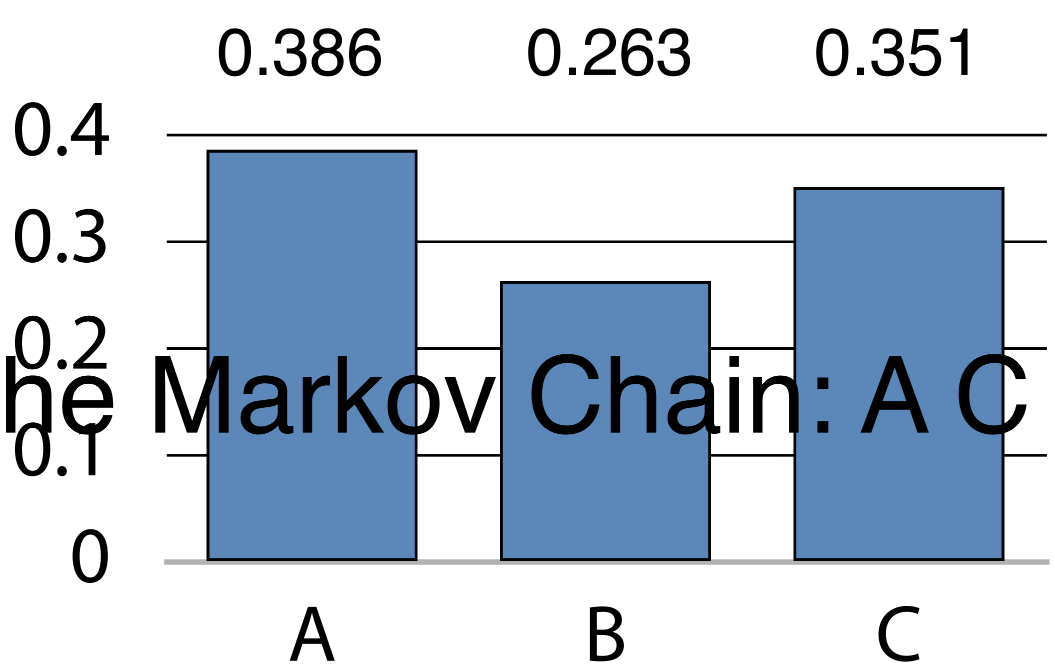
The use of special Markov Chains to create Monte Carlo methods.

Classic example: **Metropolis-Hastings algorithm**

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

Markov Chain review





Simulating the Markov Chain: A C C A B A C ....

Metropolis-Hastings review

Given

1. State space Ω (discrete or continuous)

2. A probability distribution $f : \Omega \to \mathbb{R}$

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

MCMC Pseudocode

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

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

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

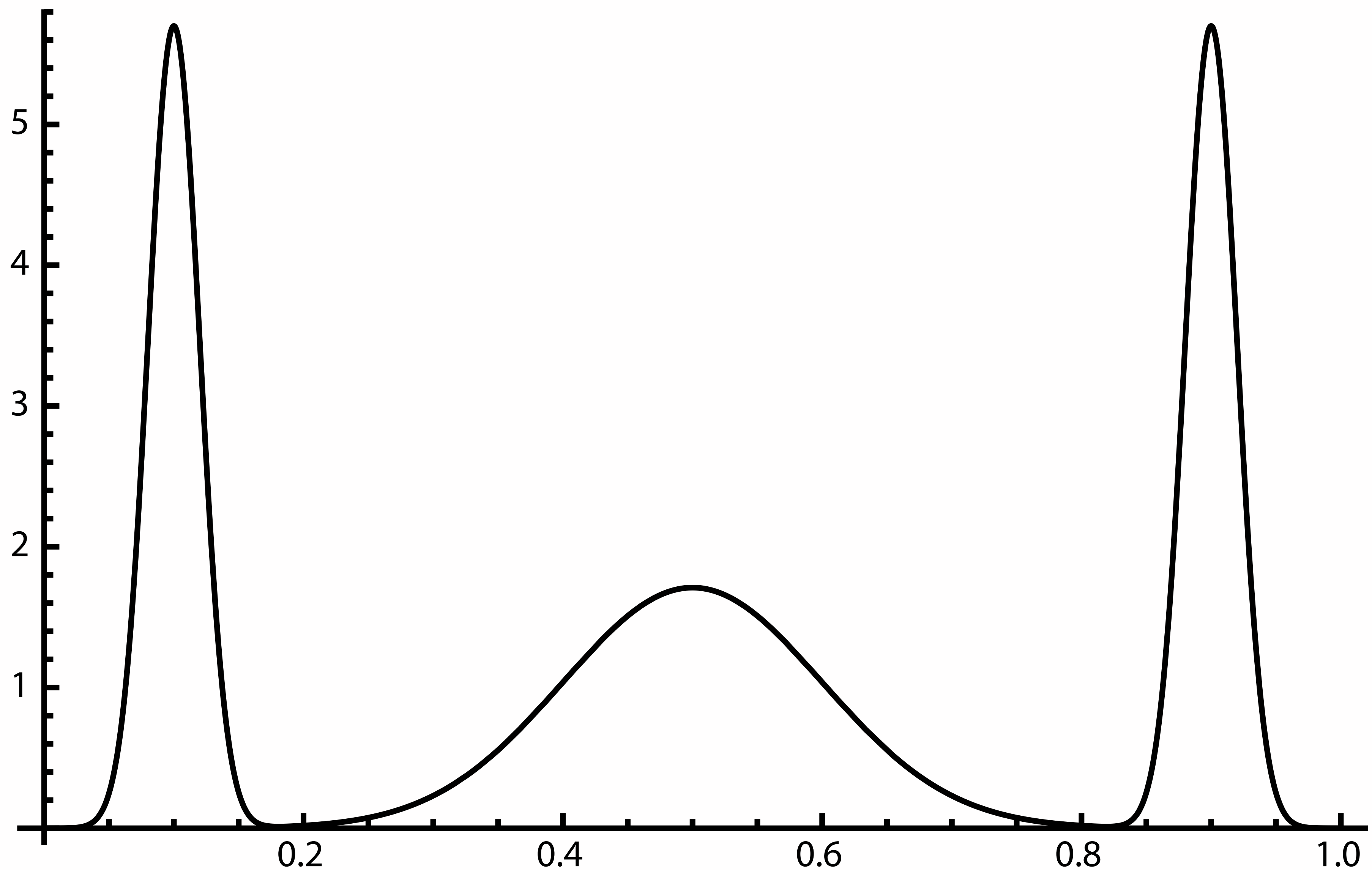
$$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i & \text{with probability } a(\mathbf{x}_{i-1} \rightarrow \mathbf{x}'_i) \\ \mathbf{x}_{i-1} & \text{otherwise} \end{cases}$$

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

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

Example application

Draw samples from a distribution and build a histogram



MCMC Pseudocode

Let $\mathbf{x}_0 =$ 0.5

For $i =$ 1, .., 1000 **do**

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

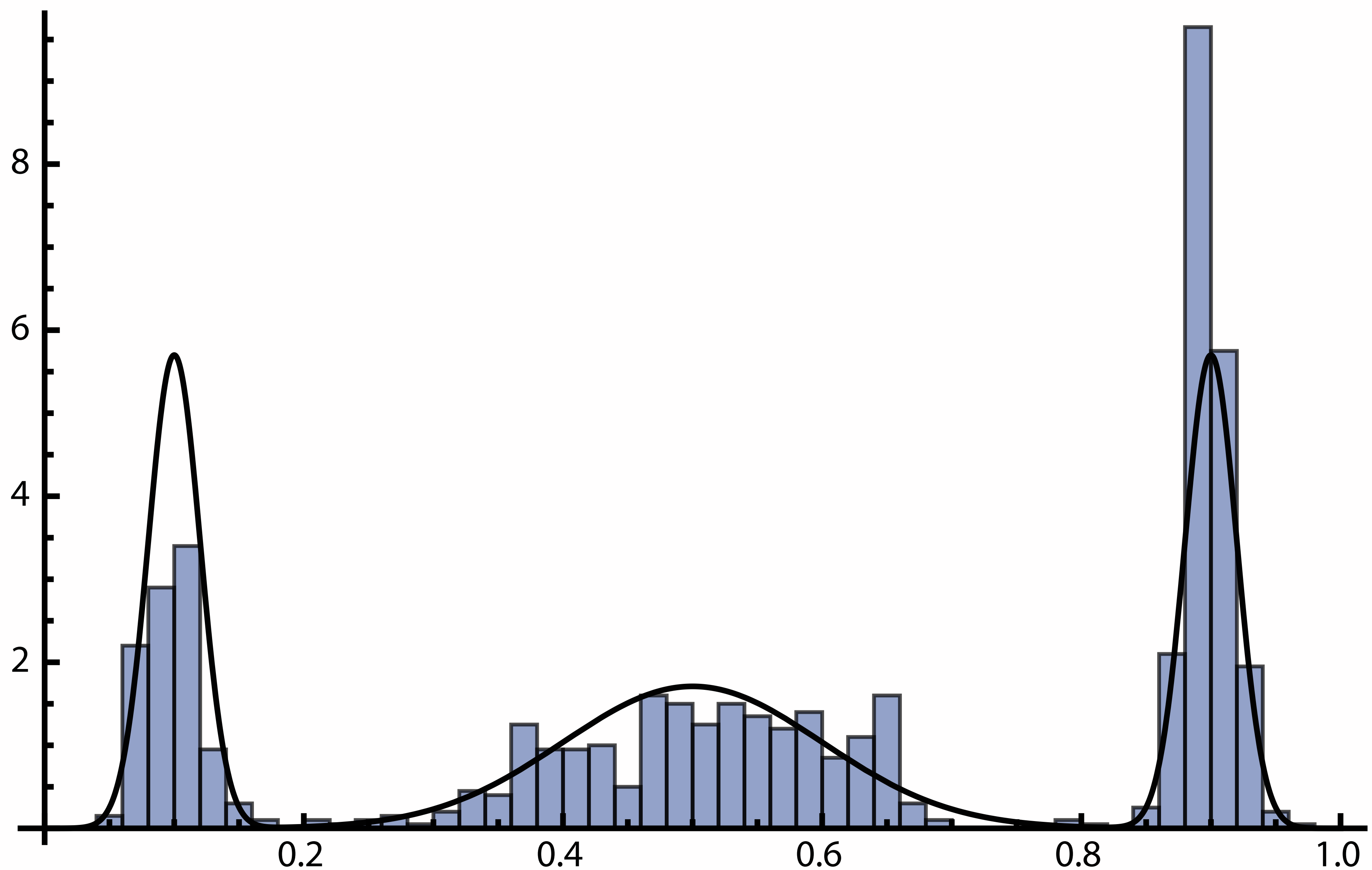
$$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i & \text{with probability } a(\mathbf{x}_{i-1} \to \mathbf{x}'_i) \\ \mathbf{x}_{i-1} & \text{otherwise} \end{cases}$$

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

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

Example application (mutation)

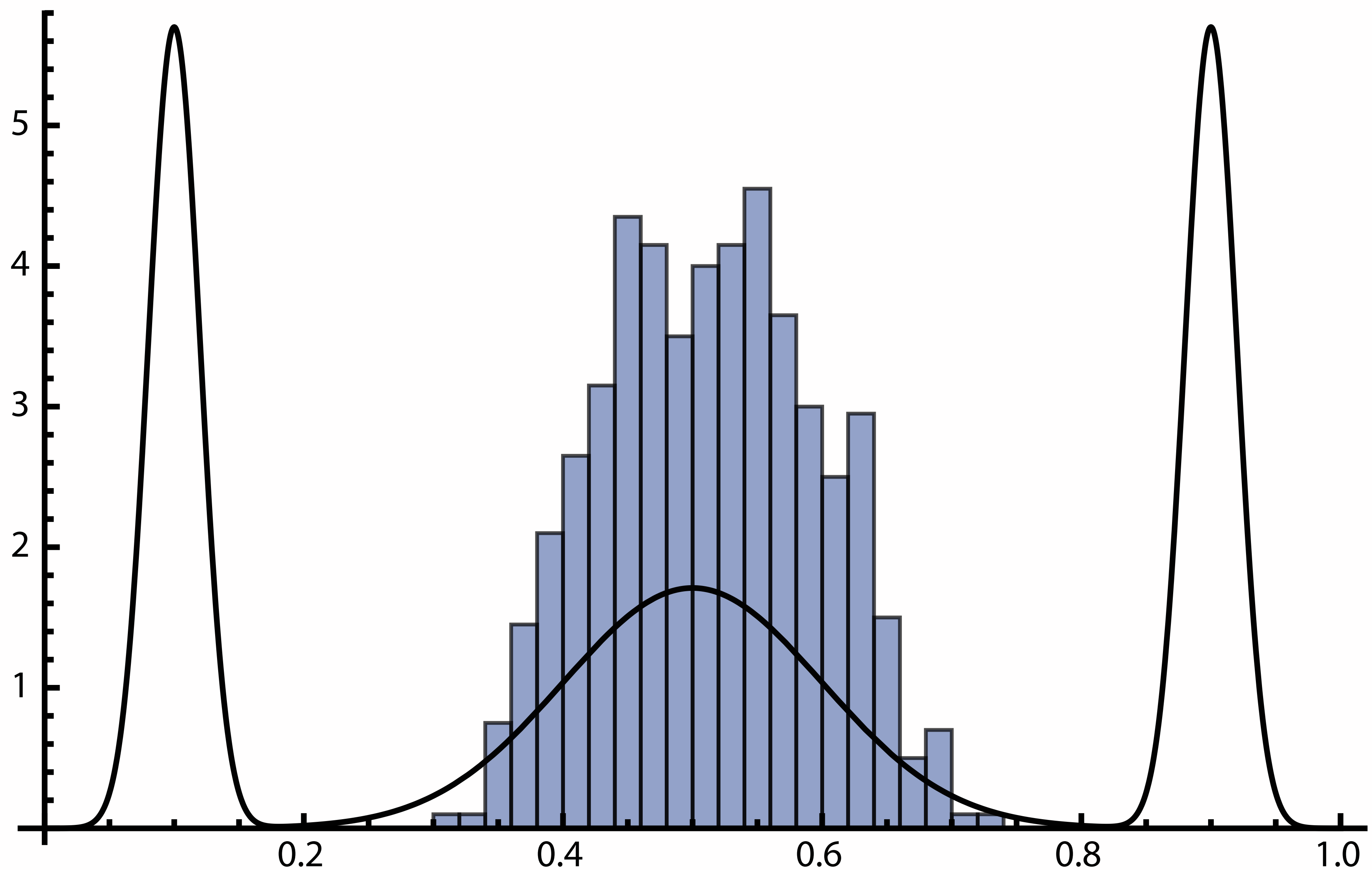
$\text{mutate}_1(\mathbf{x})$ = uniform sample on [0, 1]



acceptance rate: 35%

Example application (perturbation)

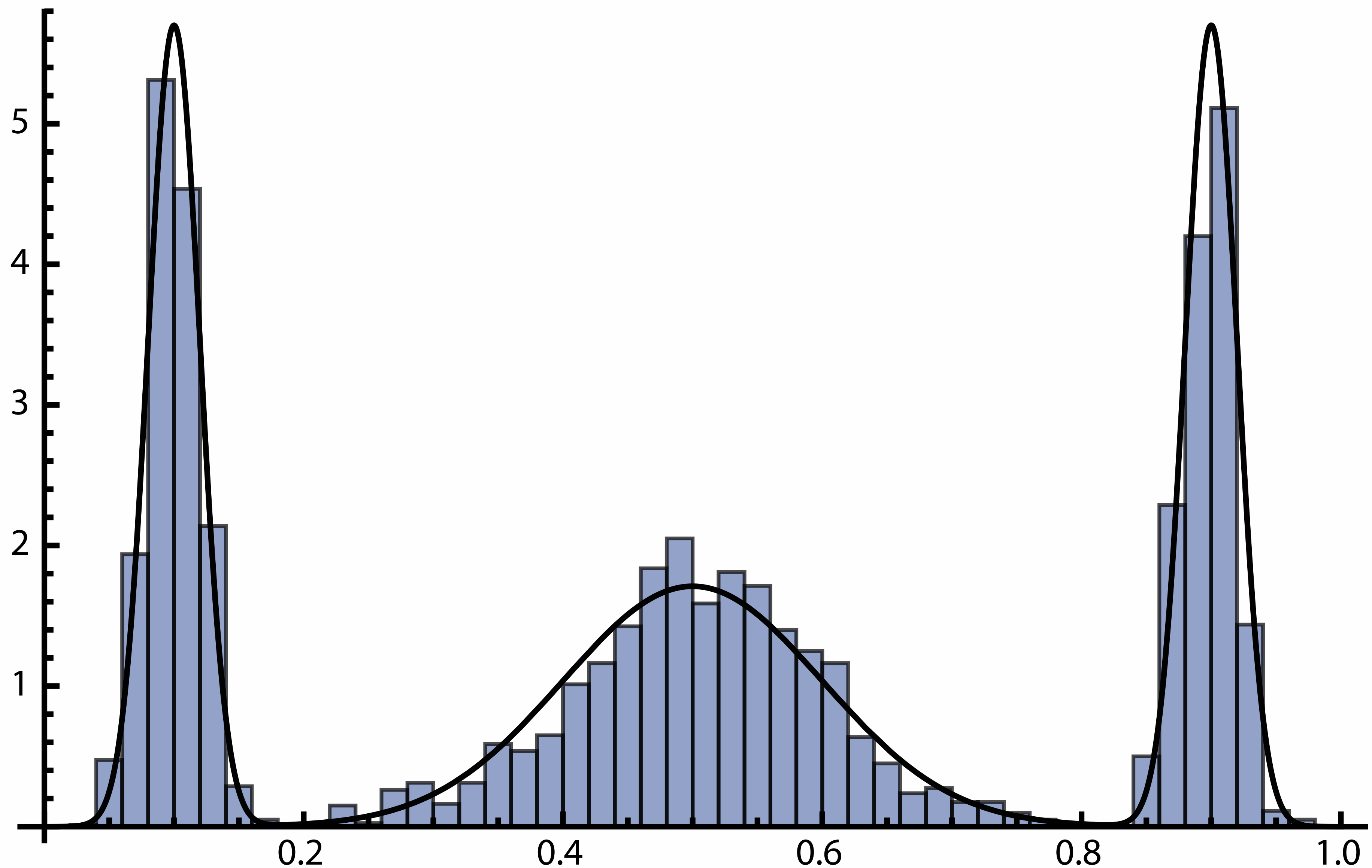
mutate$_2$($\mathbf{x}$) = uniform sample on [$\mathbf{x}$ - 0.05, $\mathbf{x}$ + 0.05]



acceptance rate: 91%

Example application (both!)

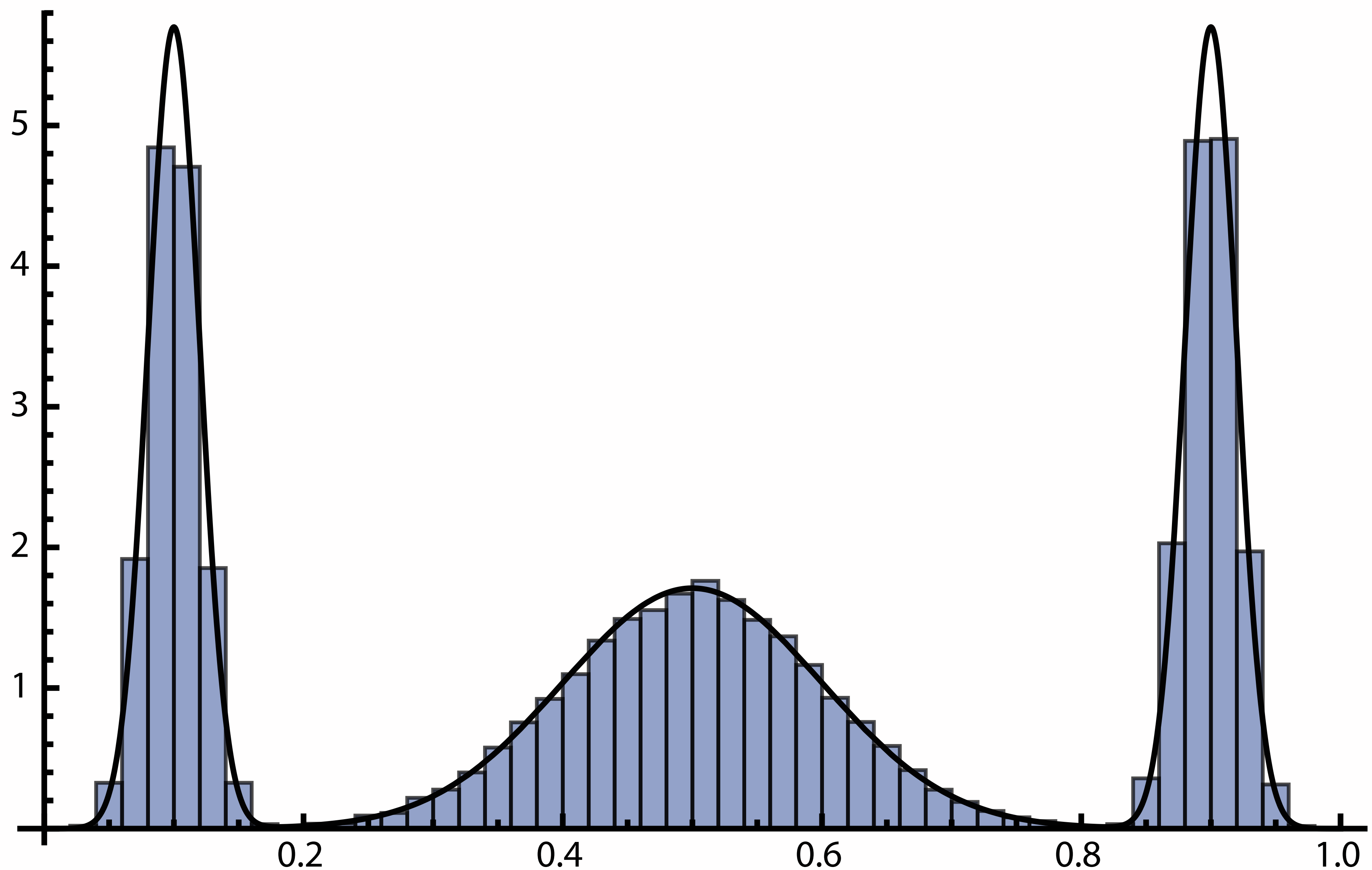
$\text{mutate}_3(\mathbf{x}) =$ randomly choose between mutate_1 and mutate_2



acceptance rate: 63%

Example application (more samples)

$\text{mutate}_3(\mathbf{x}) =$ randomly choose between mutate_1 and mutate_2



acceptance rate: 63%

Metropolis Hastings summary

1. Technique for sampling according to an arbitrary PDF

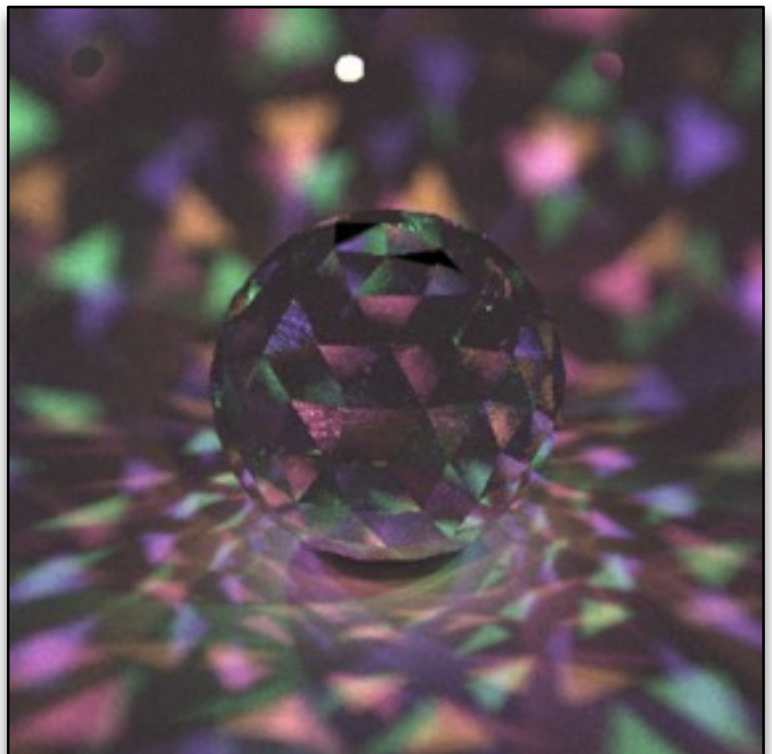
   (which must only be known up to a scale factor)

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

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

1. Return 0 with probability α

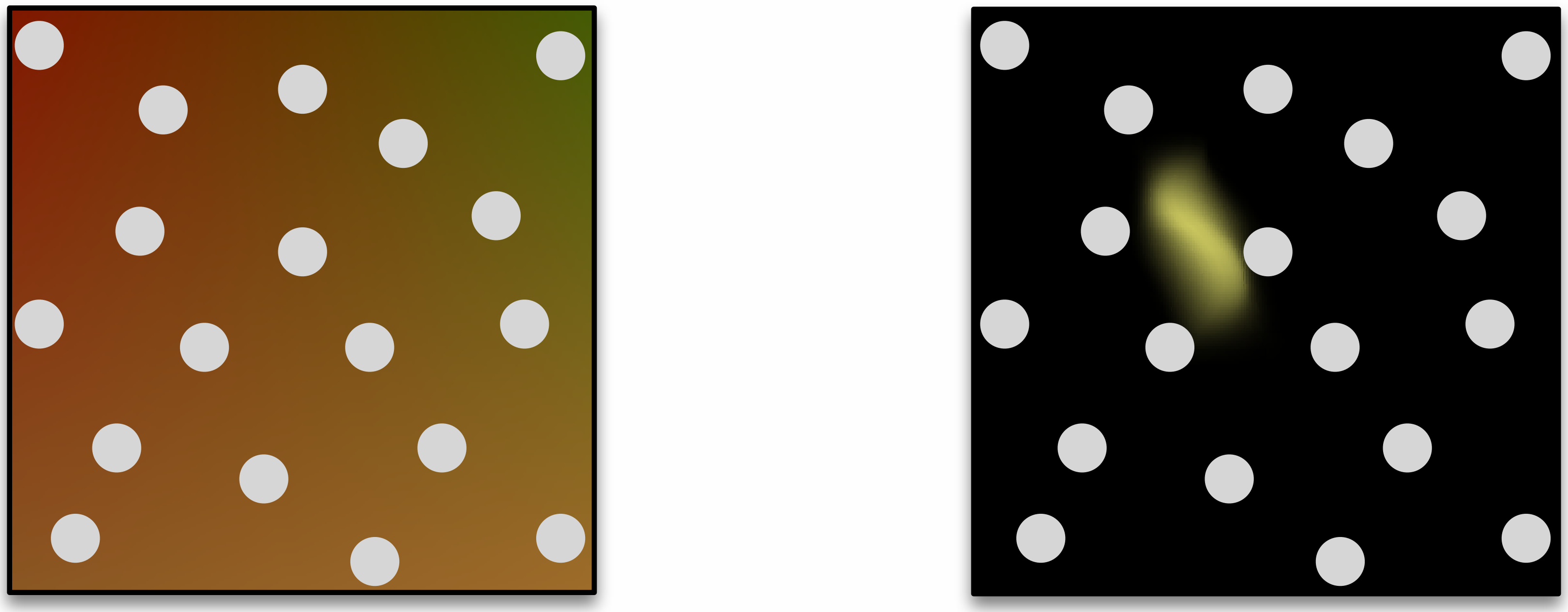
2. Sample a direction ω_i proportional to $f_s(\mathbf{x},\cdot,\omega_o)$

3. Set $\mathbf{x}' := r(\mathbf{x},\omega_i)$

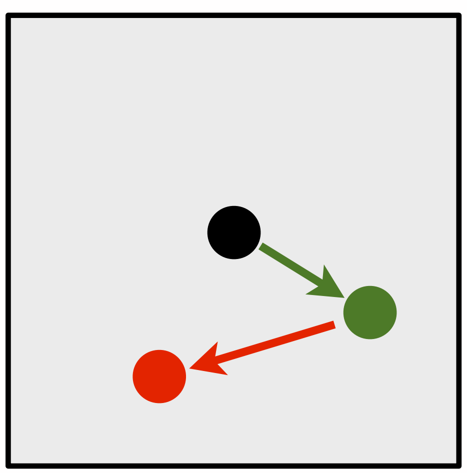
4. Return $\dfrac{1}{1-\alpha}\left(L_e(\mathbf{x},\omega_o) + P_{f_s}\langle L_o(\mathbf{x}',-\omega_i)\rangle\right)$

Primary sample space interpretation

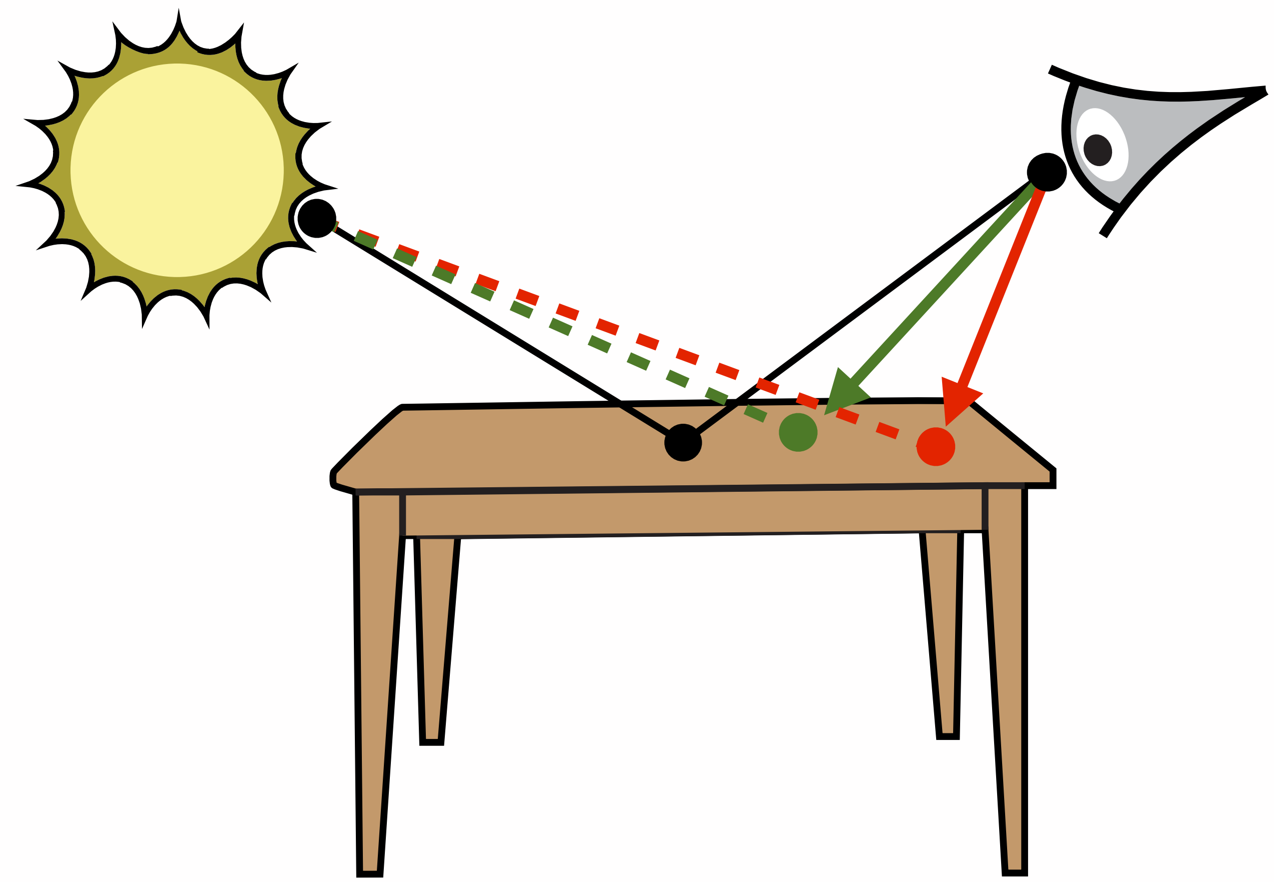
$$L = \int_0^1 \int_0^1 \cdots \; f(\xi_0, \xi_1 \ldots, \ldots) \, \mathrm{d}\xi_0 \, \mathrm{d}\xi_1 \cdots$$



Primary sample-space MLT



Primary sample-space view



Path-space view

MCMC Pseudocode

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

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

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

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

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

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

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

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

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

$$= \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) + \int_A f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) L_e(\mathbf{x}_2, \mathbf{x}_1) + \int_A f(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_1) G(\mathbf{x}_2, \mathbf{x}_3)\, L_e(\mathbf{x}_3, \mathbf{x}_2) + \int_A \cdots\, d\mathbf{x}_4 d\mathbf{x}_3 d\mathbf{x}_2 d\mathbf{x}_1 d\mathbf{x}_0$$

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

Emission

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

Direct illumination (3 vertices)

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

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

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

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.

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

Emission

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

Direct illumination (3 vertices)

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

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

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

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)

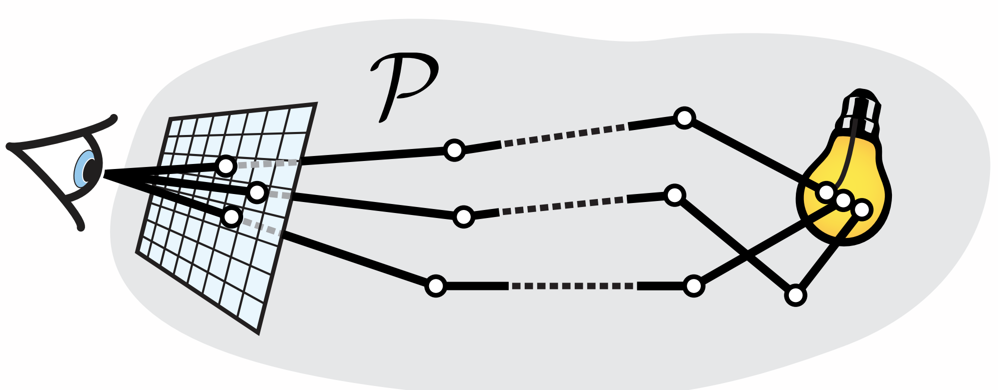
$$+ \int_{\mathcal{P}_2} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_2, \mathbf{x}_1) T(\bar{\mathbf{x}}_2)\, d\bar{\mathbf{x}}_2 + \cdots$$

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

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

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.

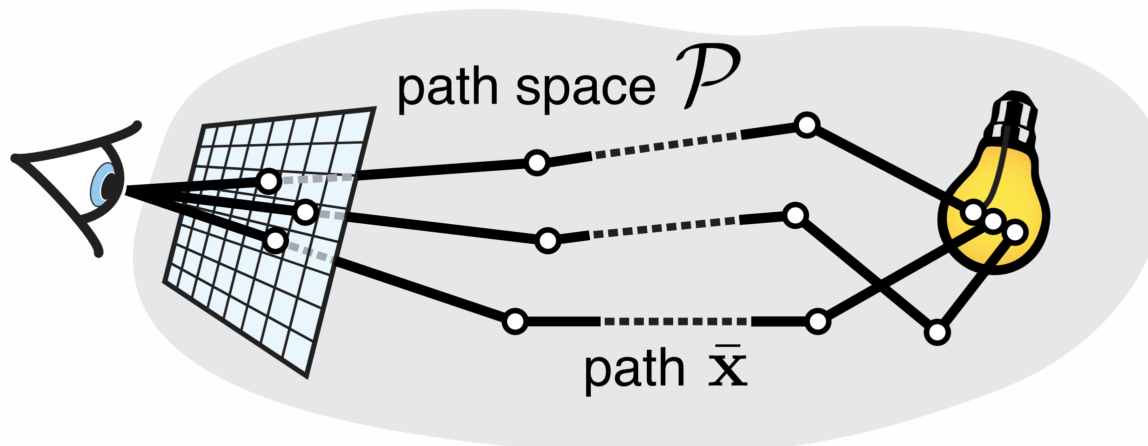
$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0) \, d\mathbf{x}_1 d\mathbf{x}_0$$

global illumination (all paths of all lenghts)

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

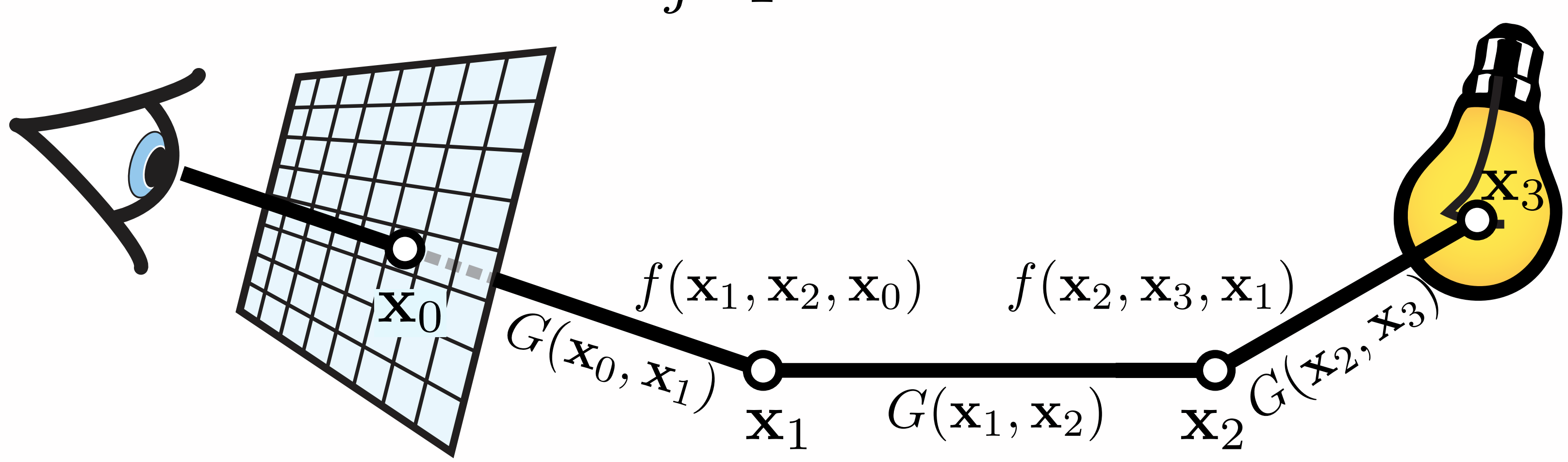
Path Integral Form of Measurement Eq.

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



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

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

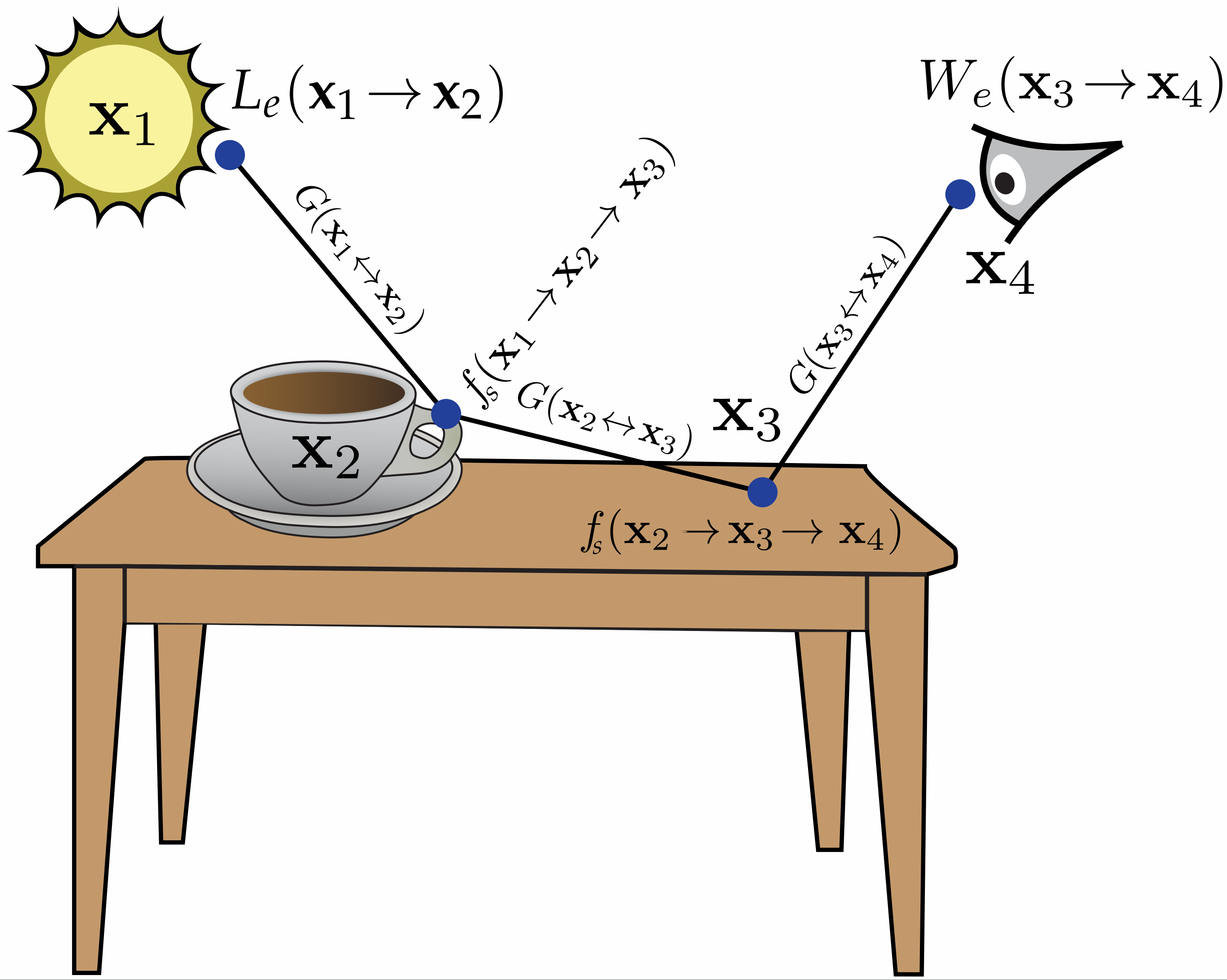
- Monte Carlo estimator:

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

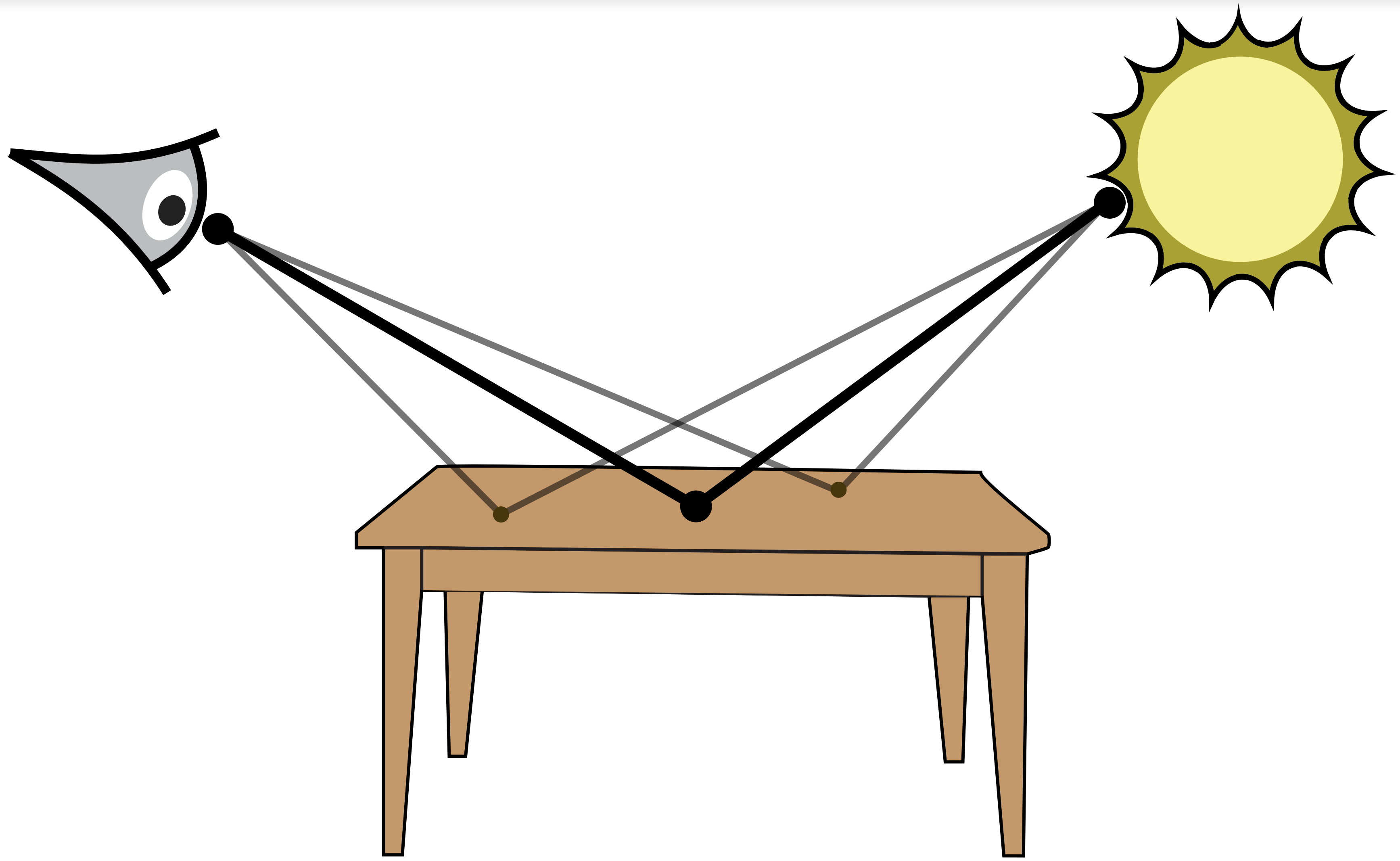
$$\underbrace{p(\bar{\mathbf{x}})}_{\text{path PDF}} = \underbrace{p(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_k)}_{\text{joint PDF of path vertices}}$$

Path space integration: an example

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



Transition to Path Space



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

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

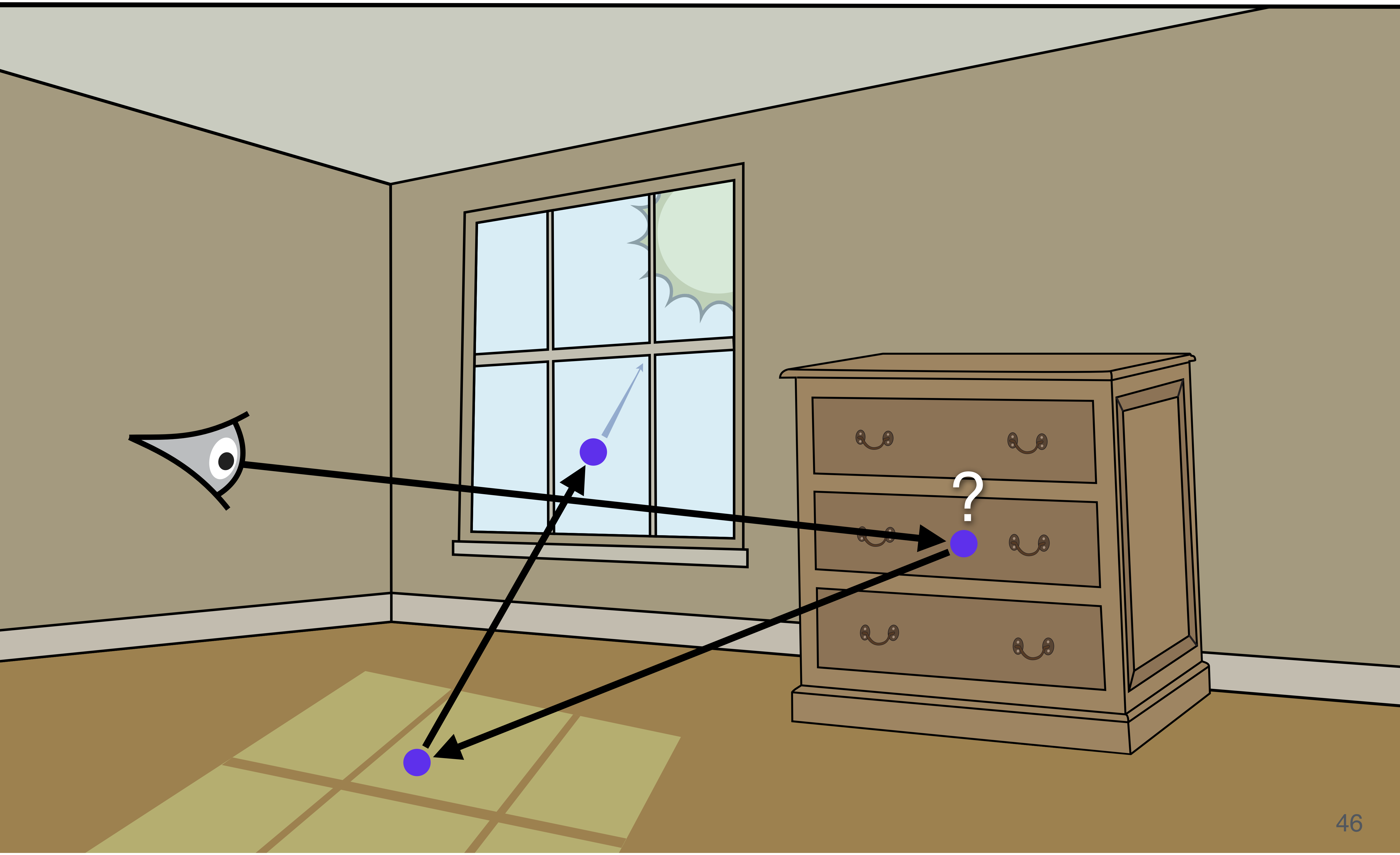
Bidirectional Path Tracing

- 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 $i = 1, \ldots, n$ do

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

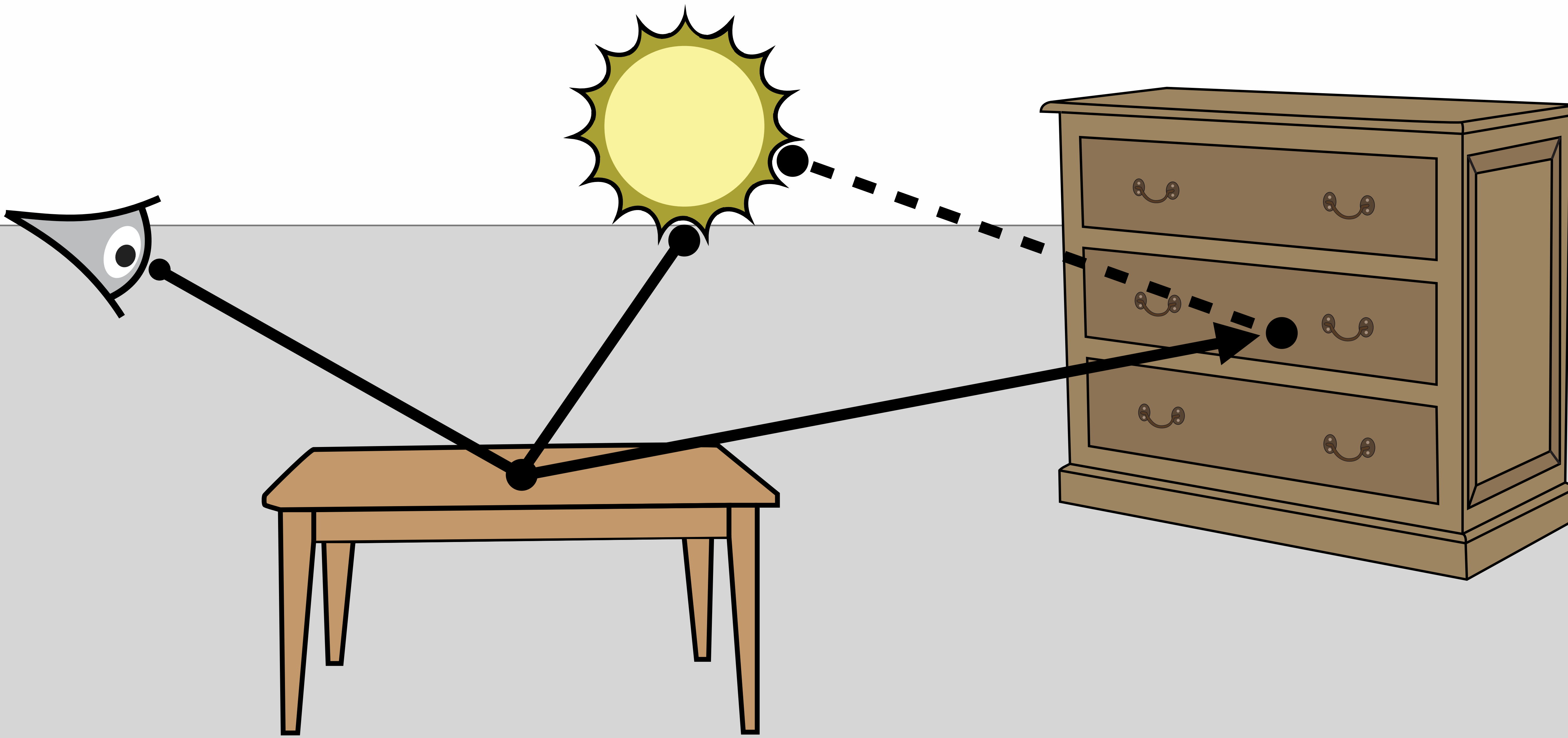
$$\mathbf{x}_i = \begin{cases} \mathbf{x}'_i & \text{with probability } a(\mathbf{x}_{i-1} \to \mathbf{x}'_i) \\ \mathbf{x}_{i-1} & \text{otherwise} \end{cases}$$

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

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

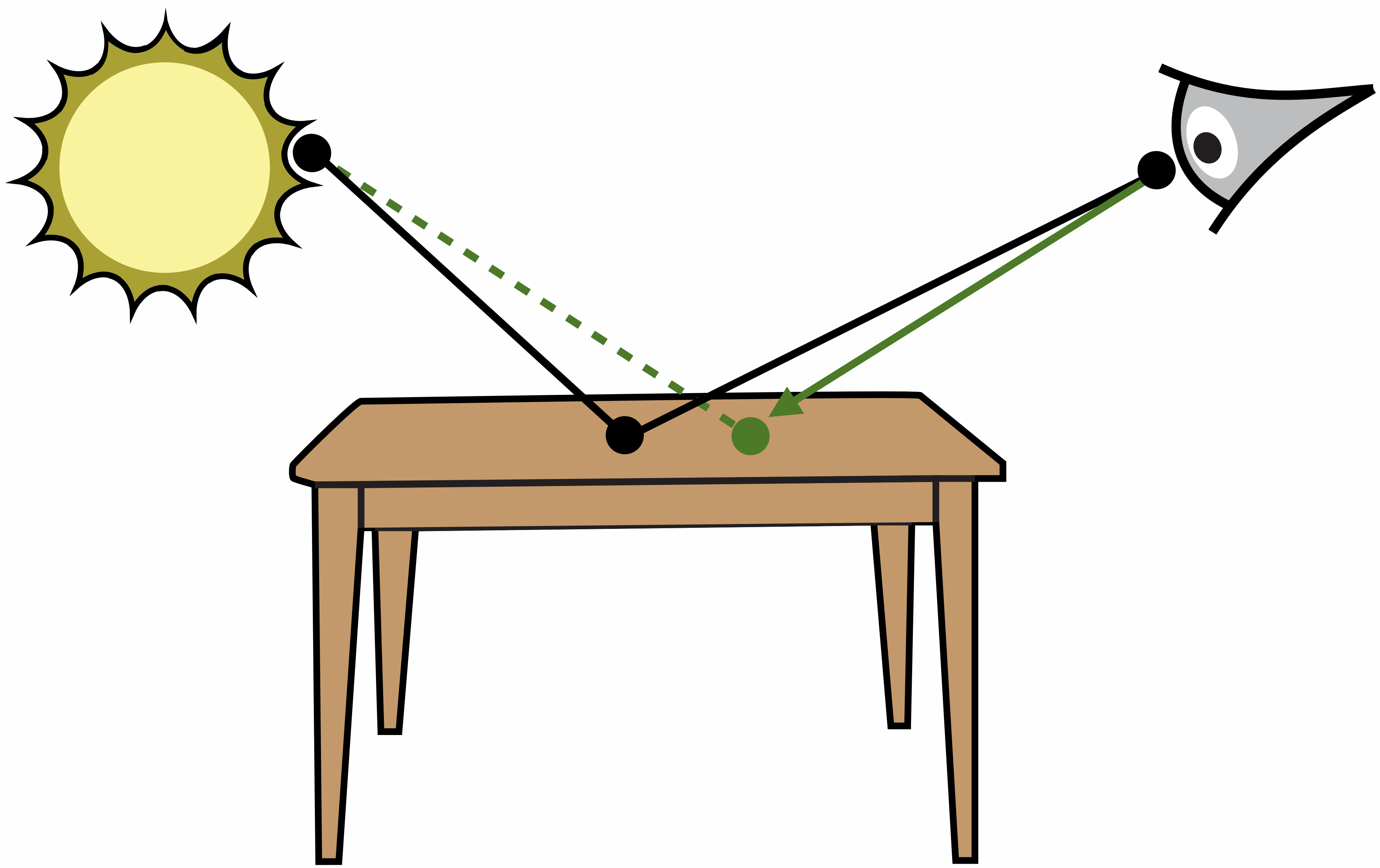
Metropolis Light Transport

Bidirectional mutation



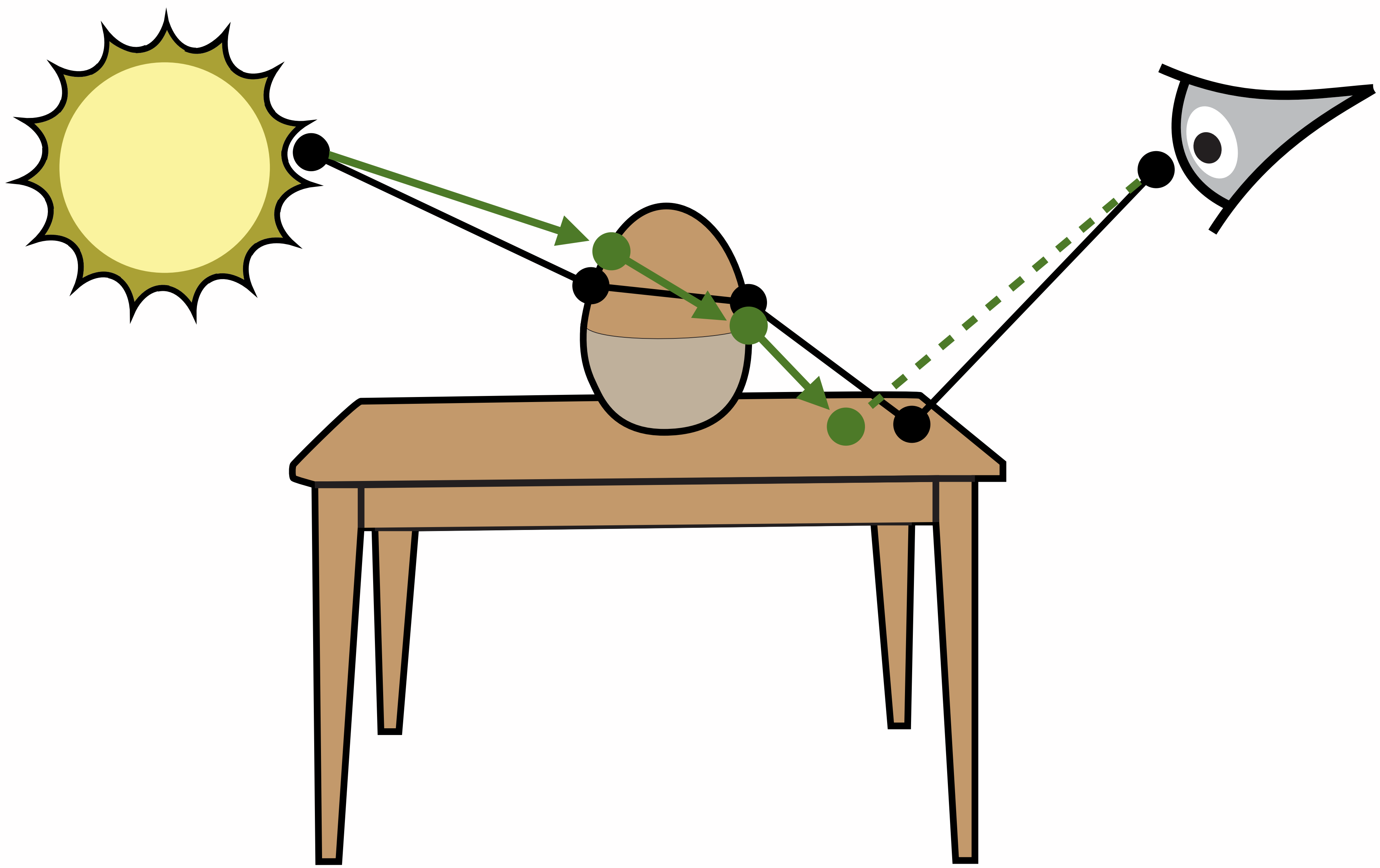
Metropolis Light Transport

Lens perturbation



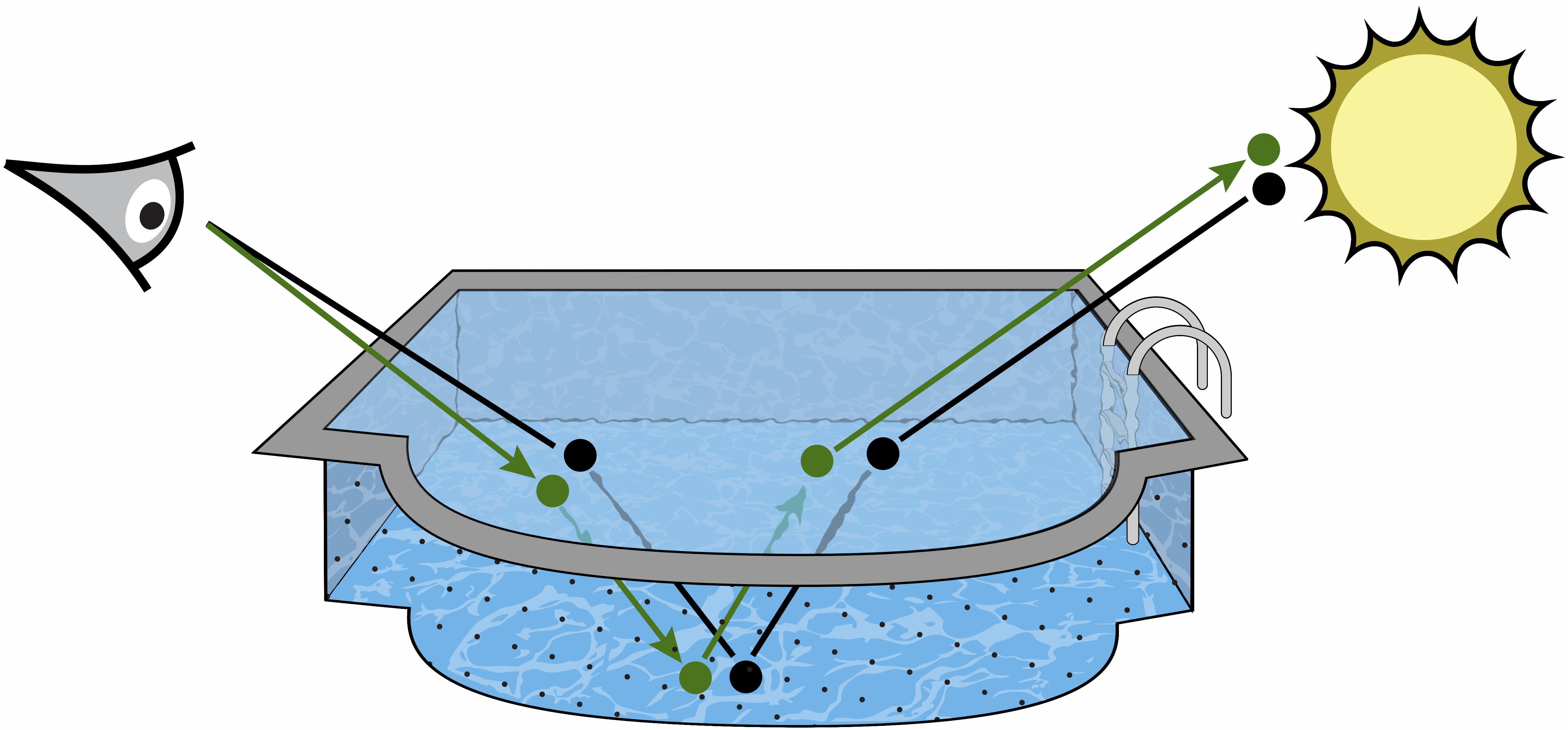
Metropolis Light Transport

Caustic perturbation

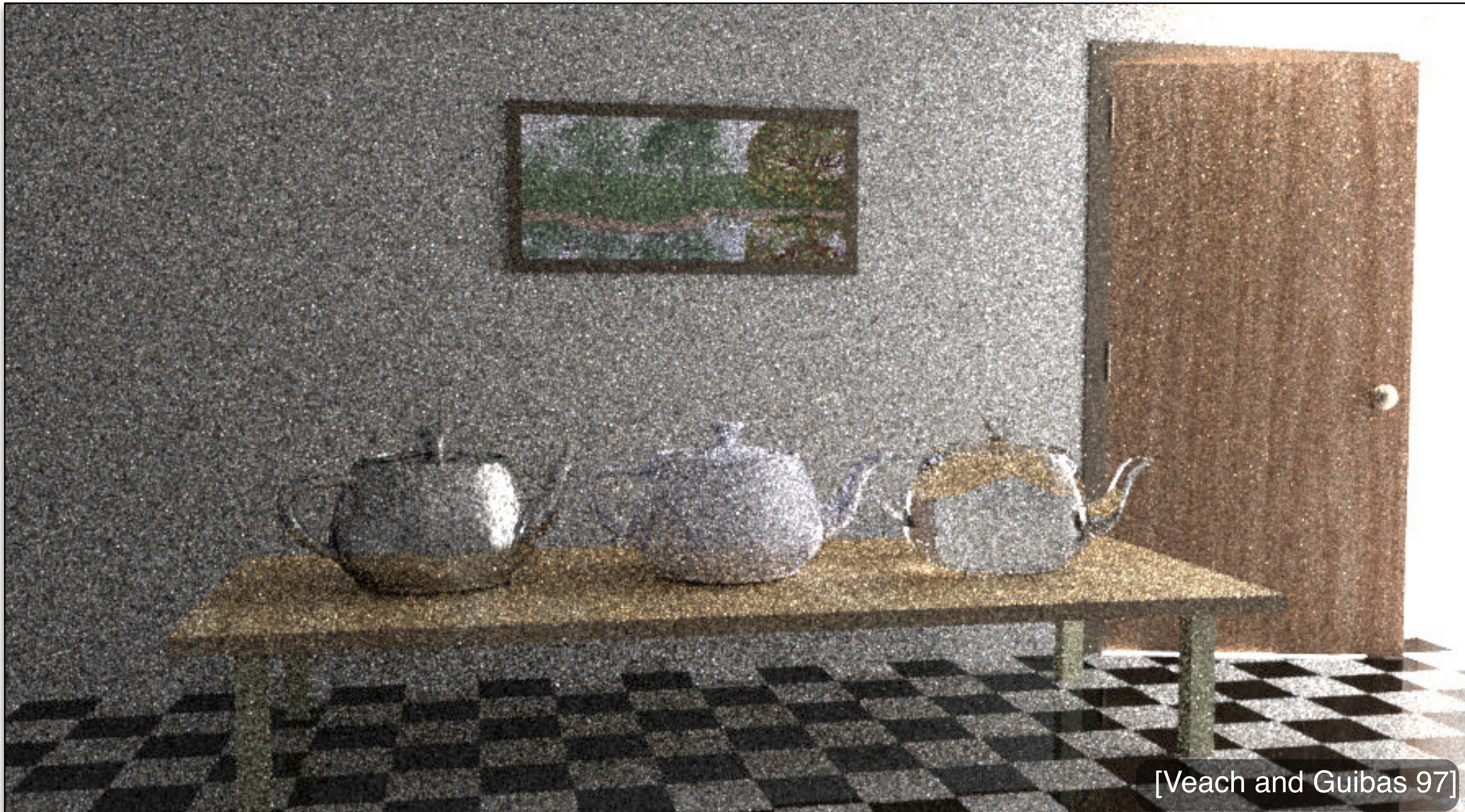


Metropolis Light Transport

Multi-Chain perturbation

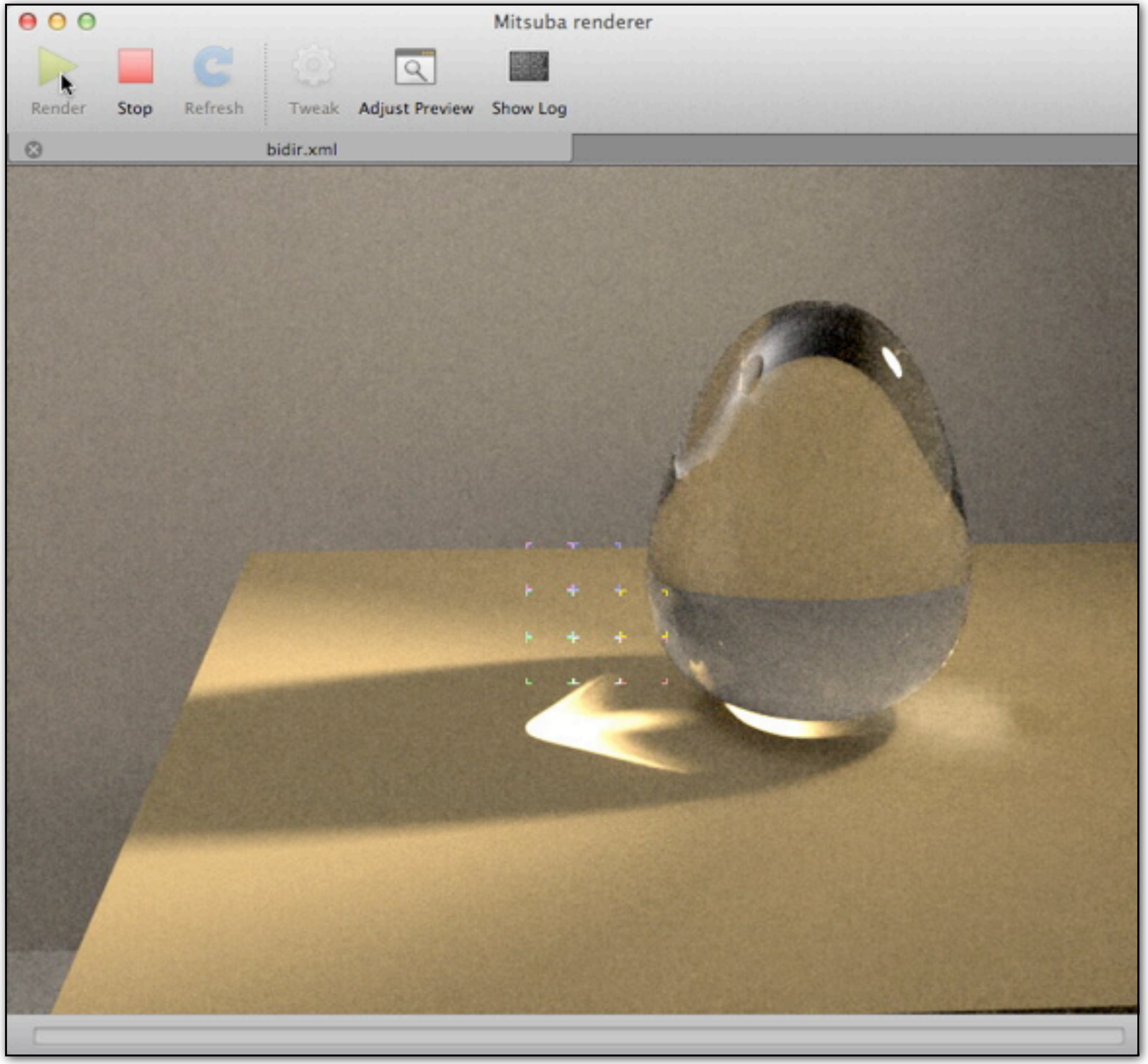


Door scene (BDPT)



Door scene (MLT)

Caustic demo

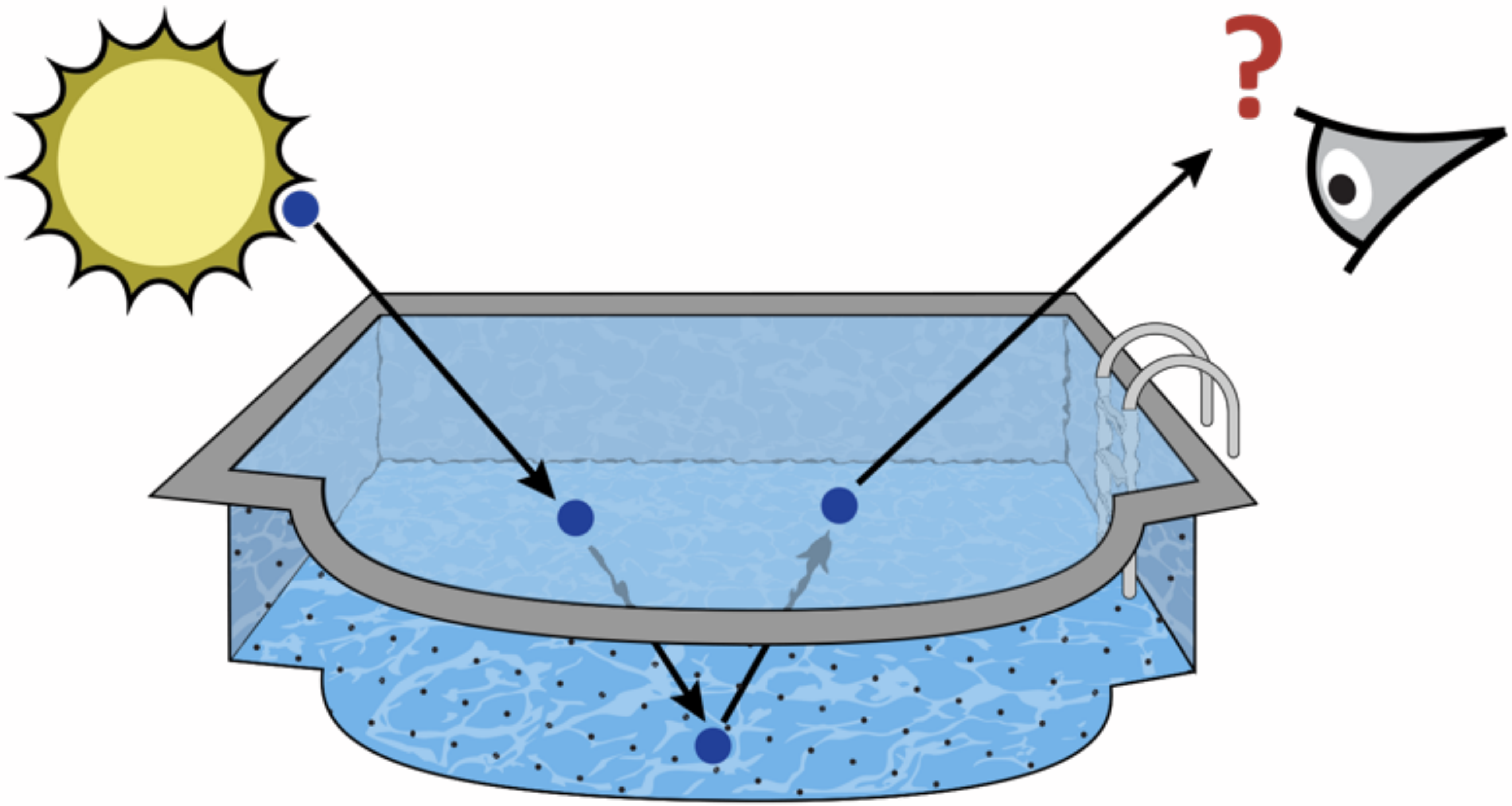


8x timelapse

Light path visualization

green = current state gray = proposal state

Related: Specular paths



Path tracing from
the light source

An observation in flatland

Light source

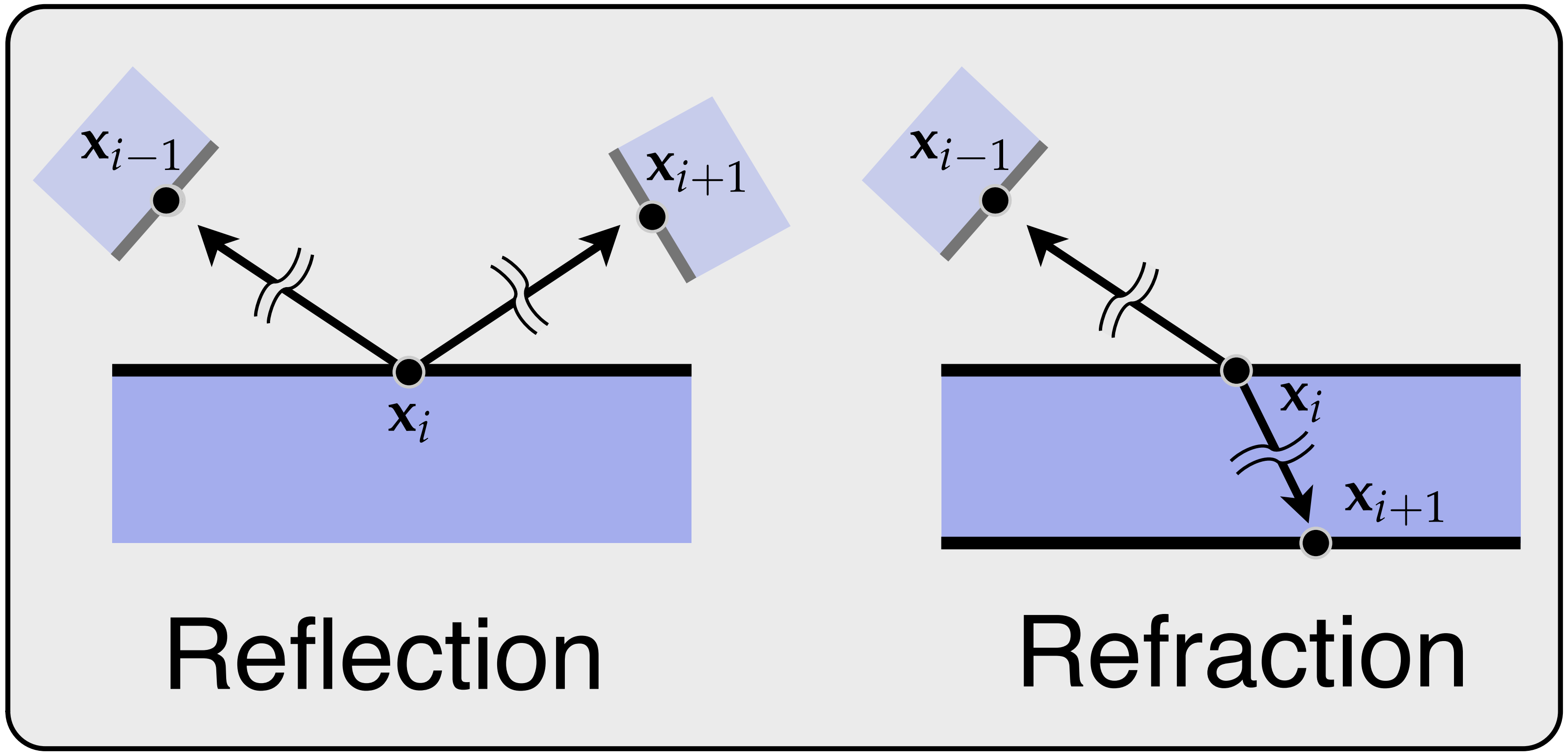
Sensor

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:

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

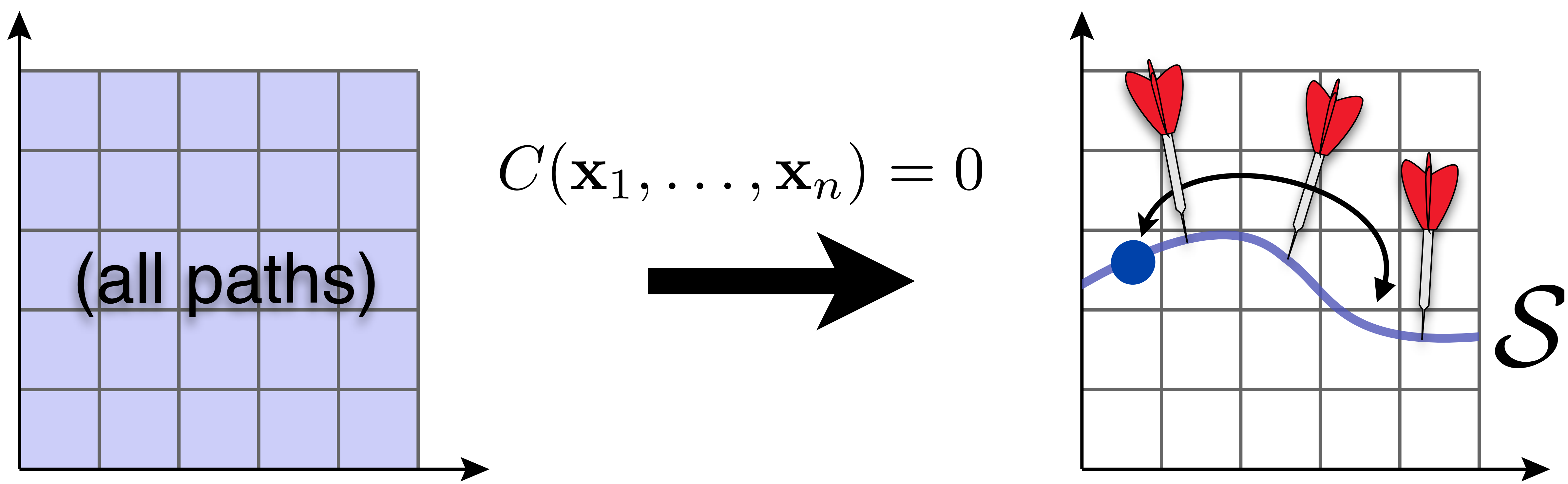
Set satisfying all constraints:

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

More formally

Set satisfying all constraints:

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

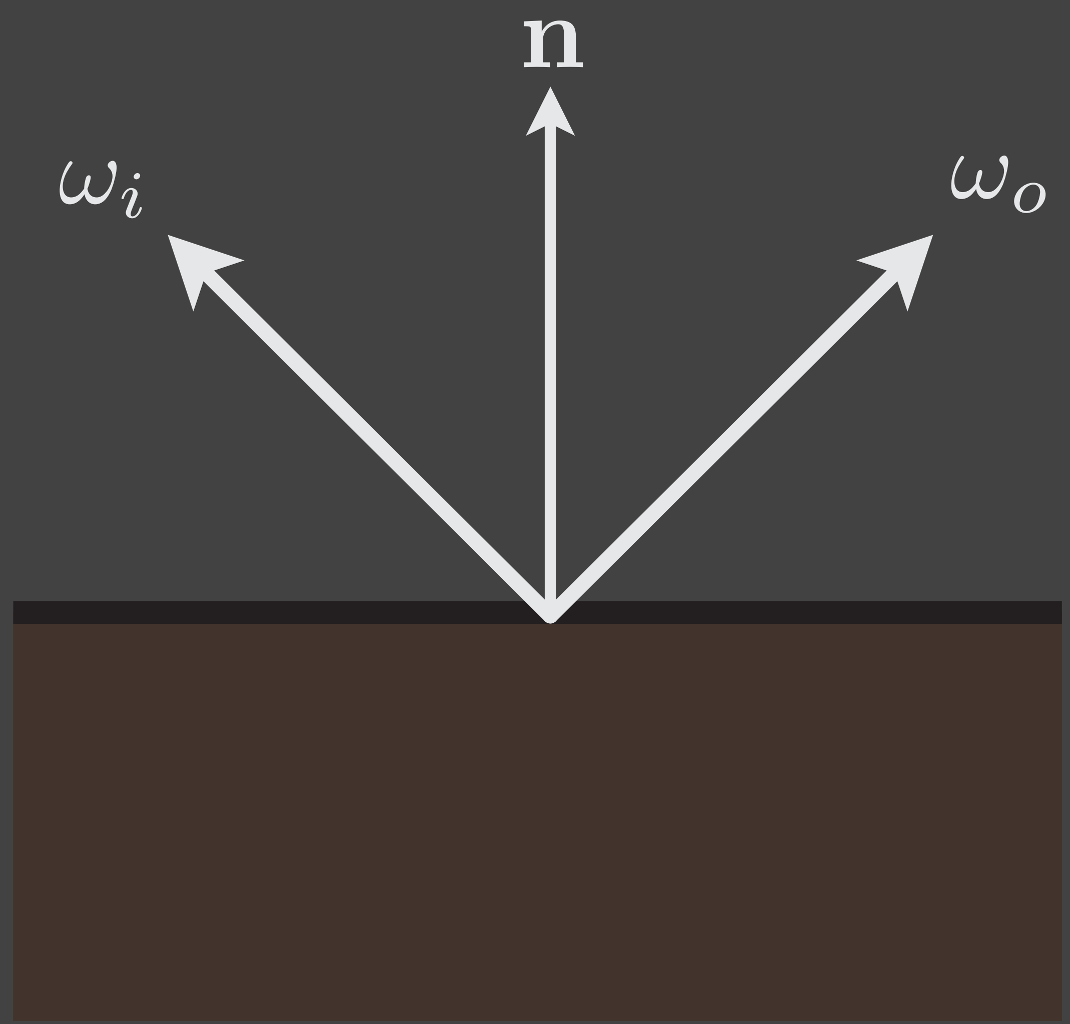


Manifold Exploration

Manifold Exploration: A Markov Chain Monte Carlo technique for rendering scenes with difficult specular transport

[Jakob and Marschner 2012]

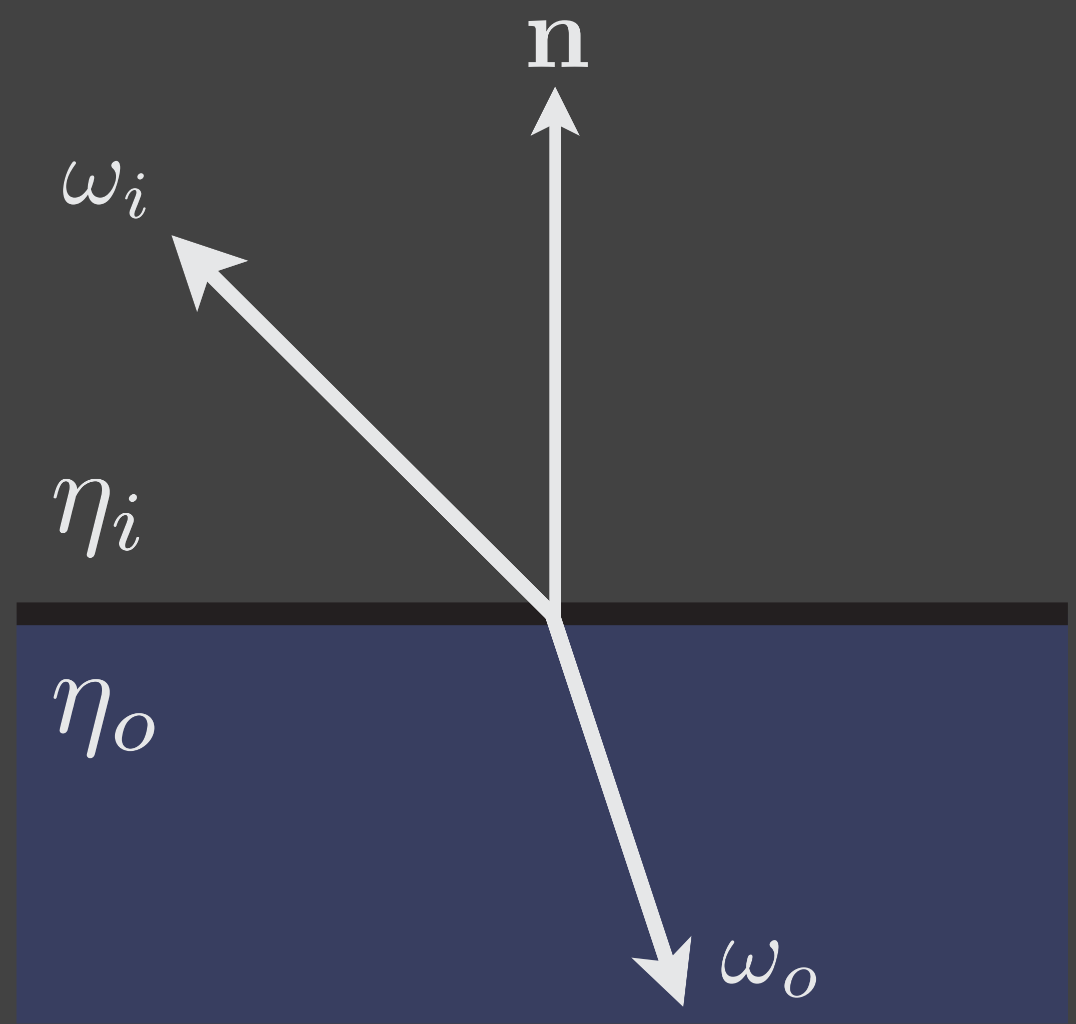
Specular path constraint



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

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

Specular path constraint

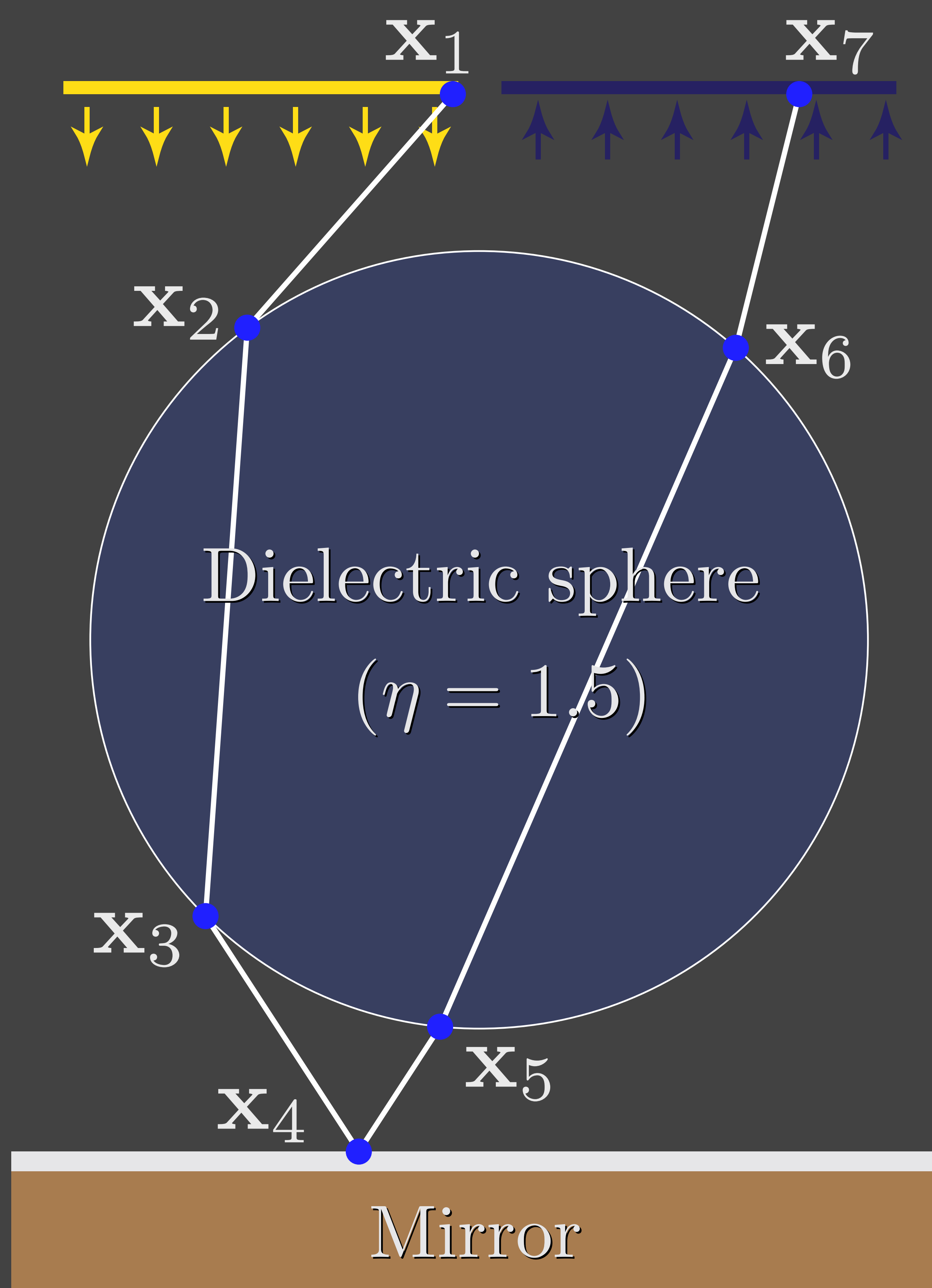


Constraint: $\mathbf{n} = H(\omega_i, \omega_o)$

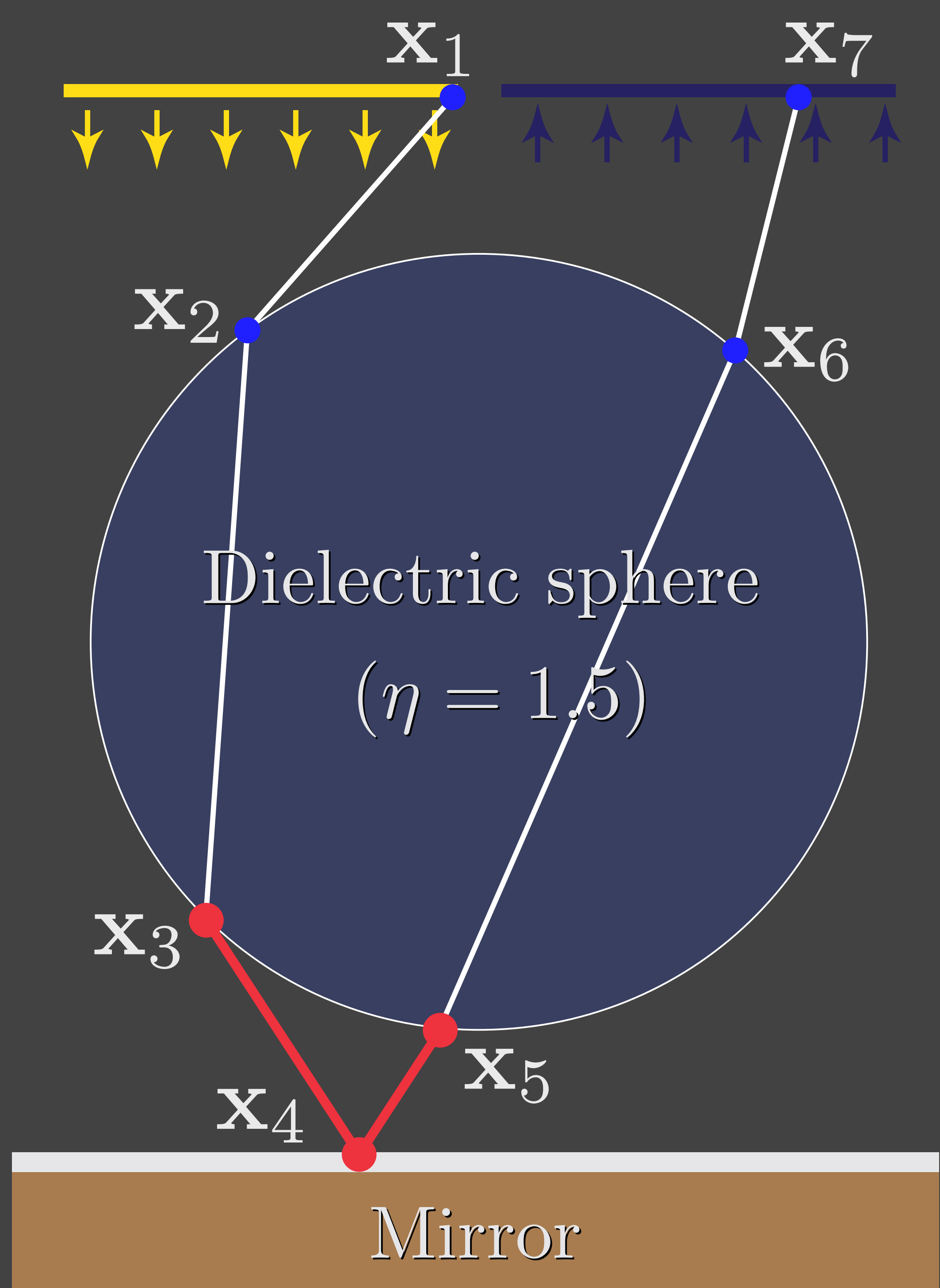
where $$H(\omega_i, \omega_o) := \frac{\eta_i \omega_i + \eta_o \omega_o}{\left\| \eta_i \omega_i + \eta_o \omega_o \right\|}$$

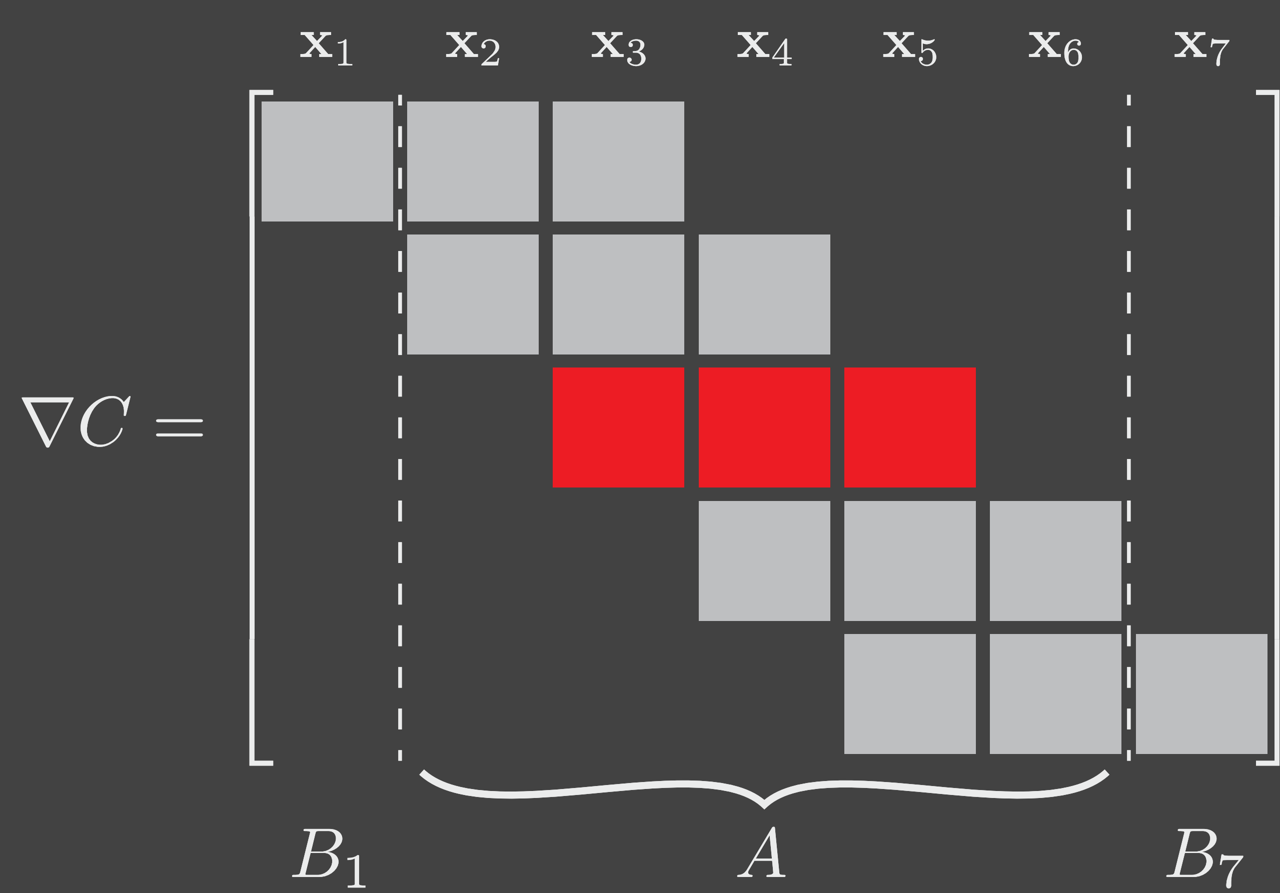
Generalized half vector [Sommerfeld and Runge 1911]

Manifold walks

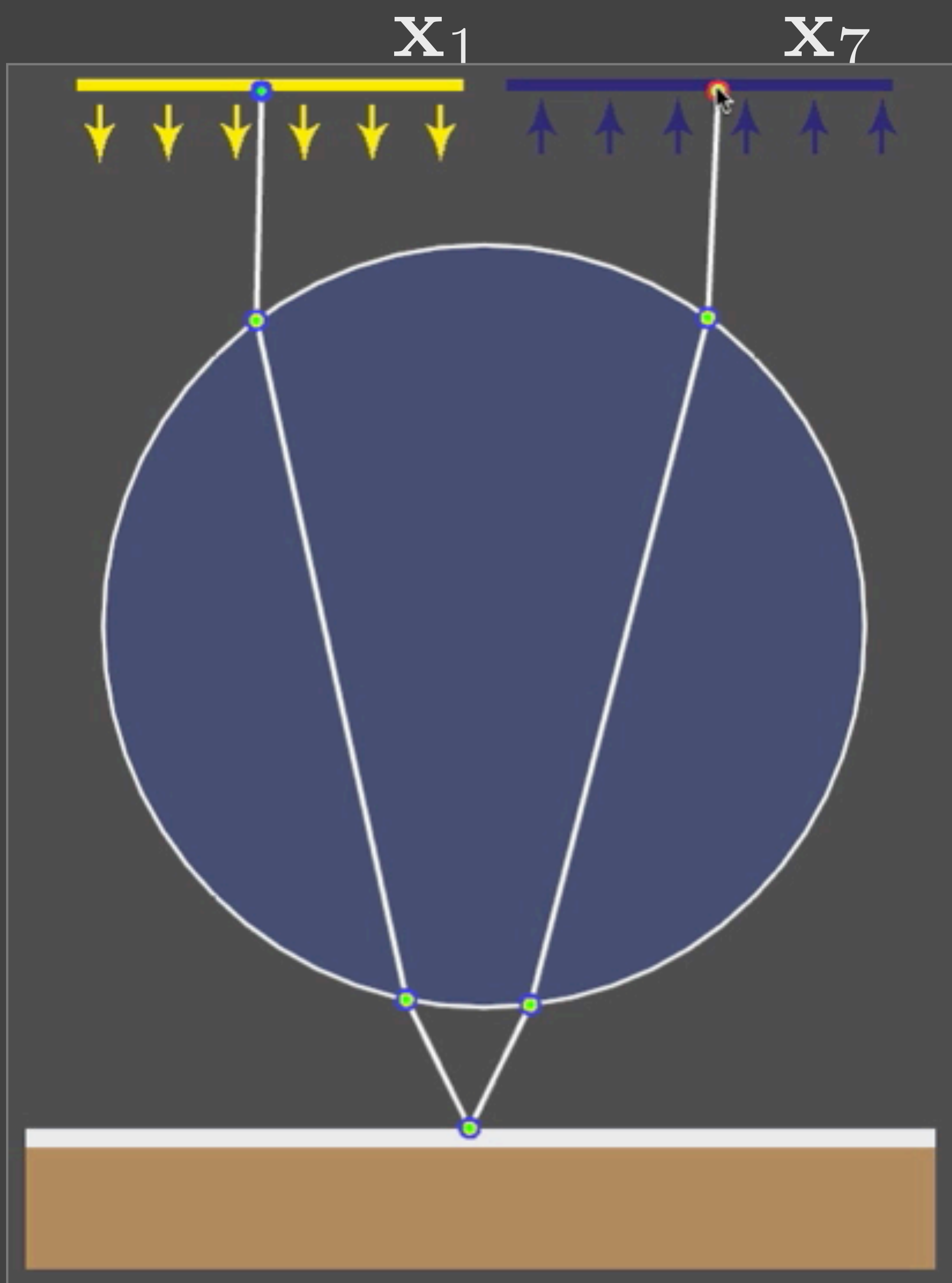


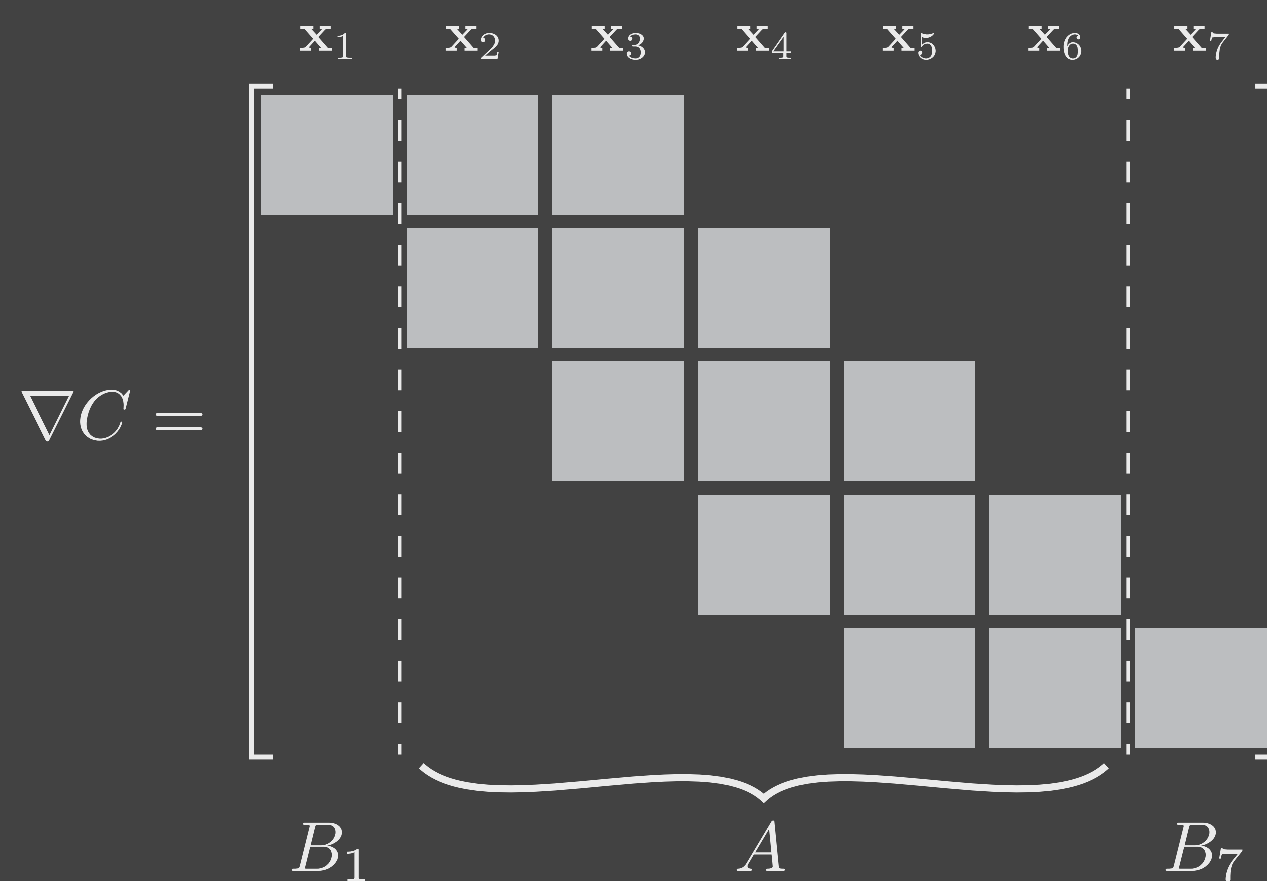
Manifold walks





Manifold walks





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

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

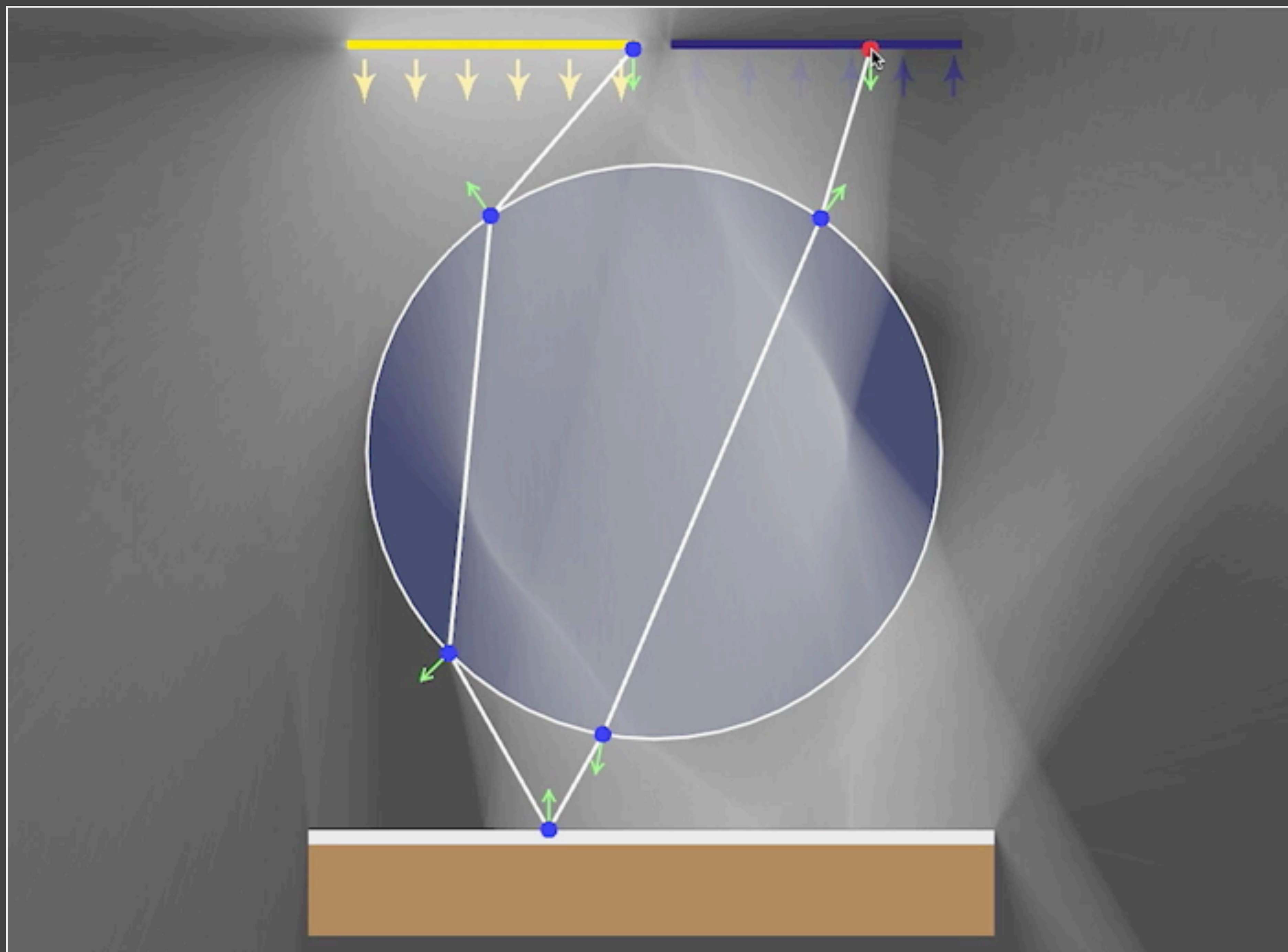
Basic idea:

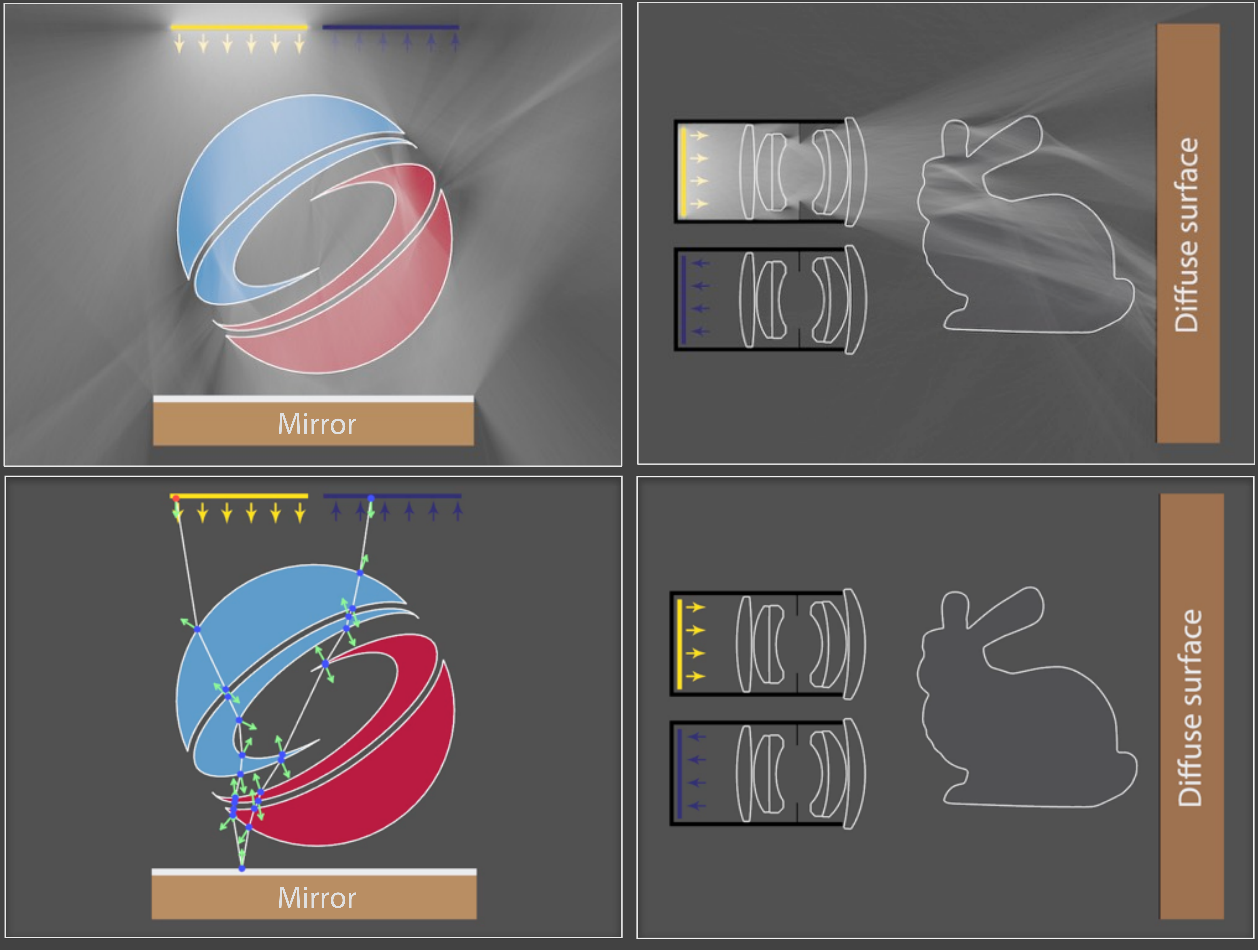
while not there yet:

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

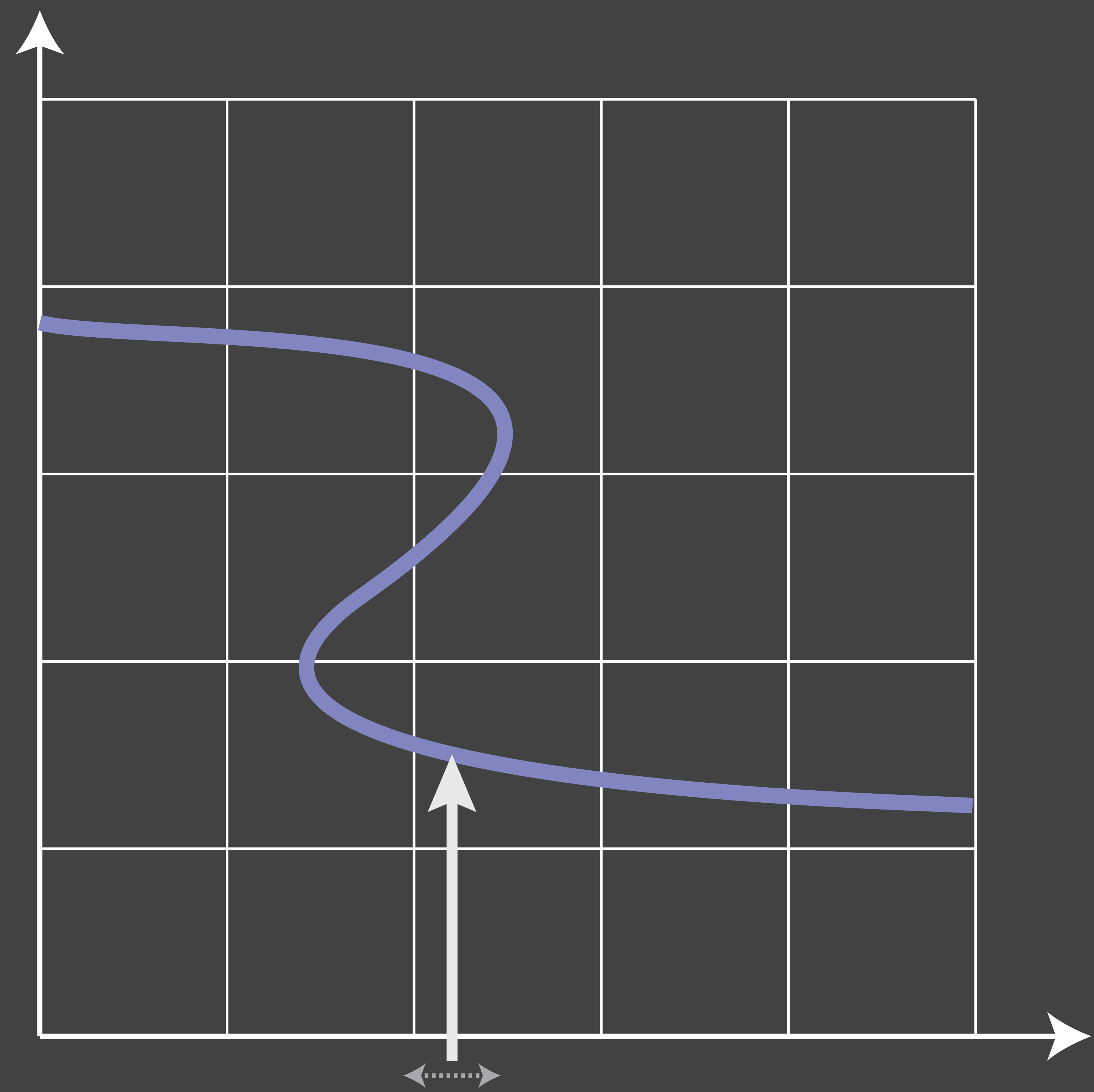
path space

Manifold walk algorithm — Demo

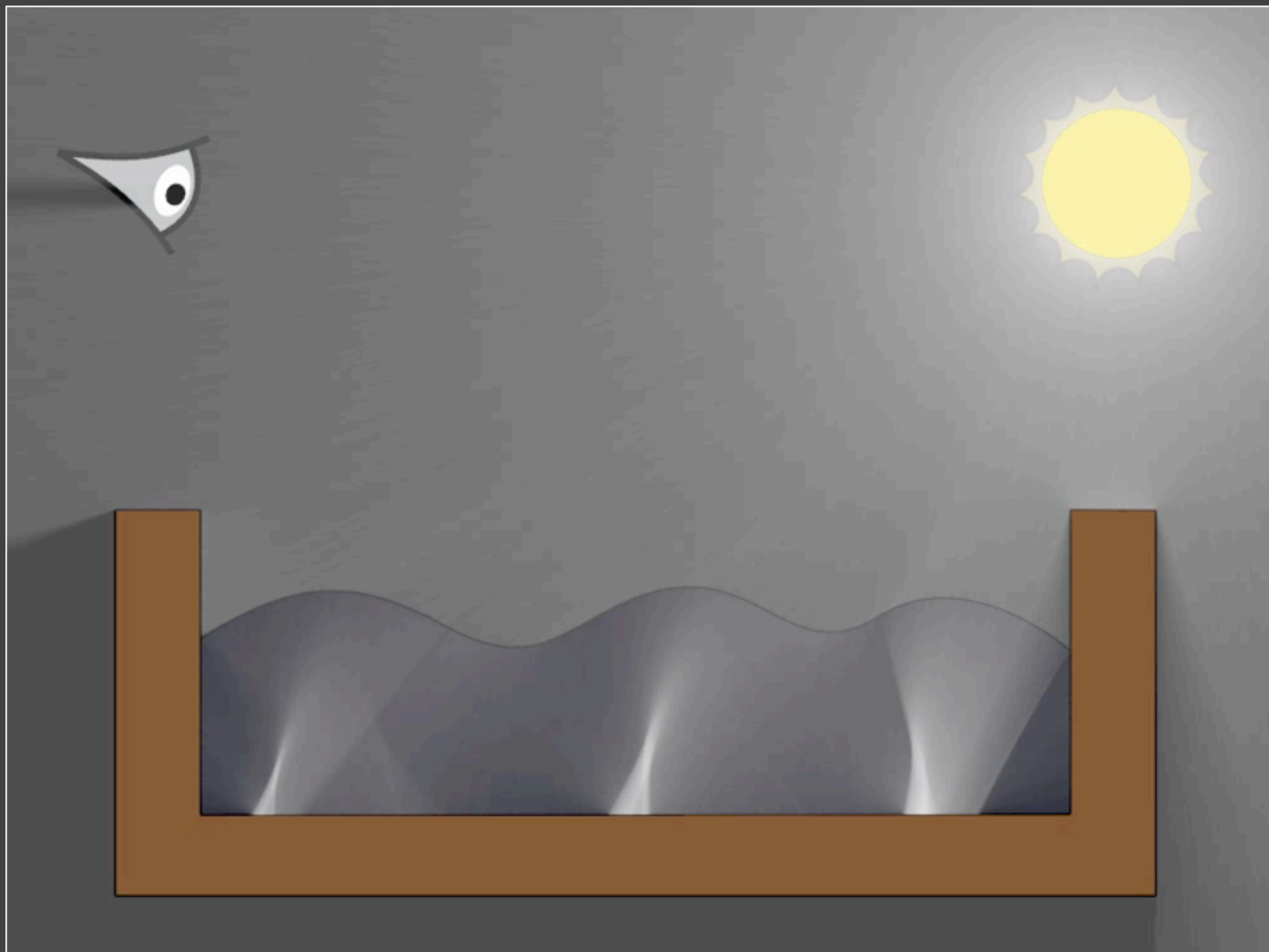



Mirror
Diffuse surface
Mirror
Diffuse surface

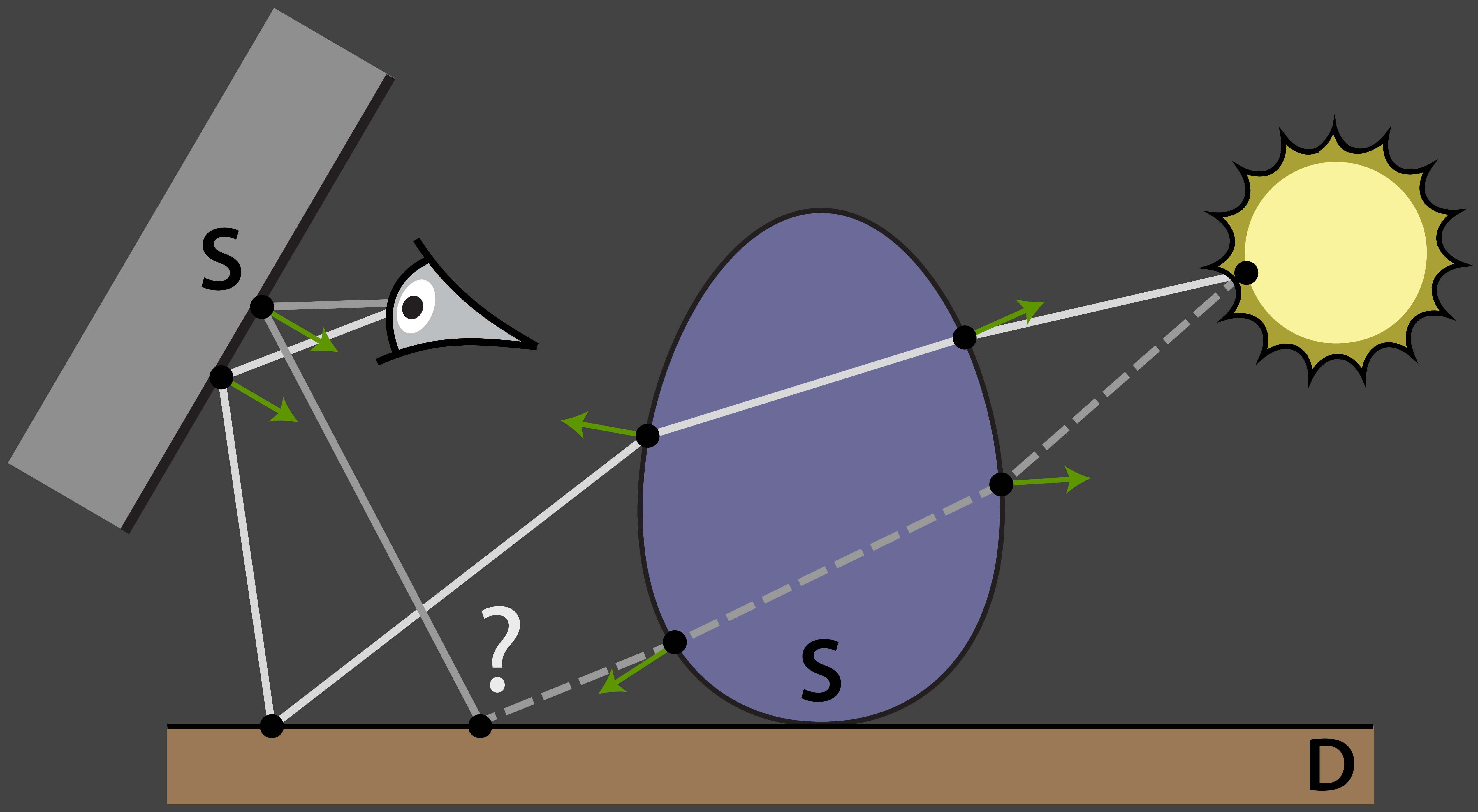
Parameterization failures

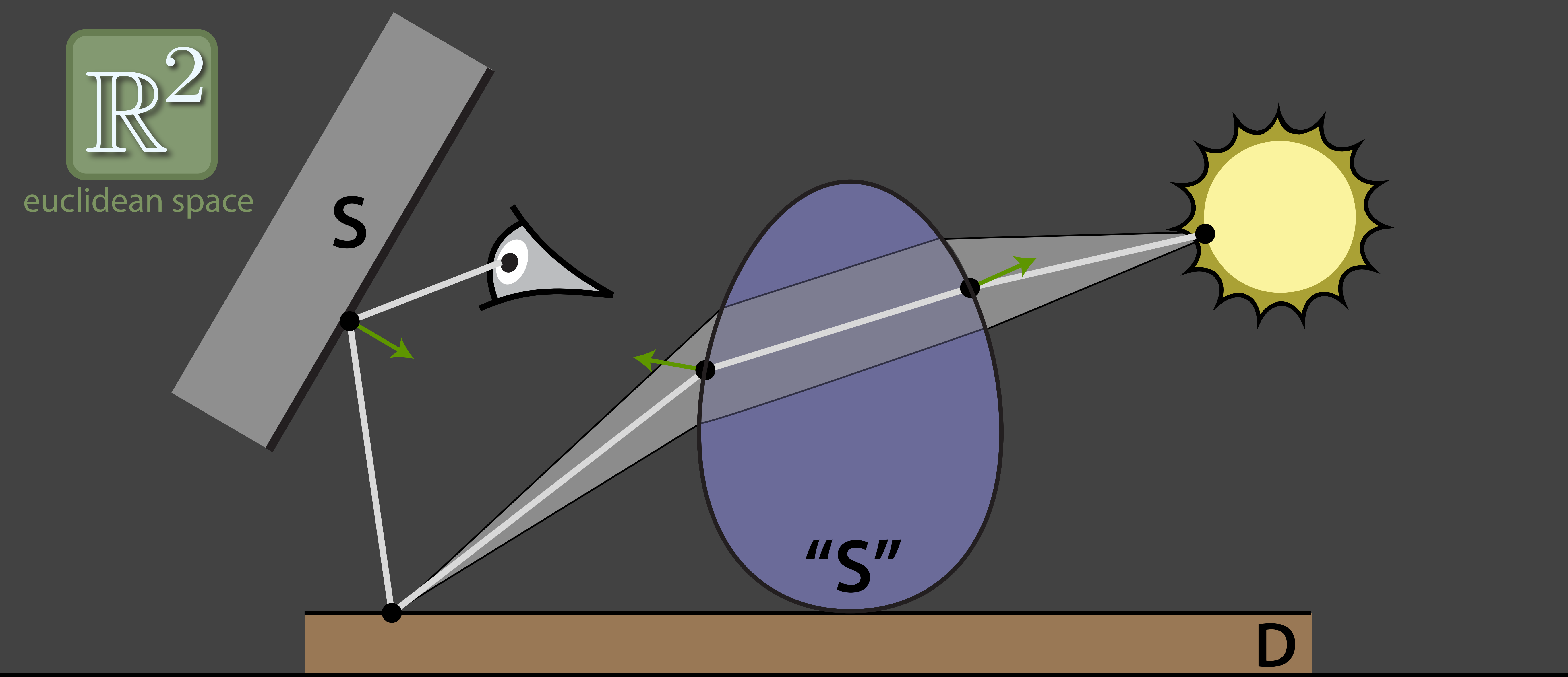


Parameterization failures

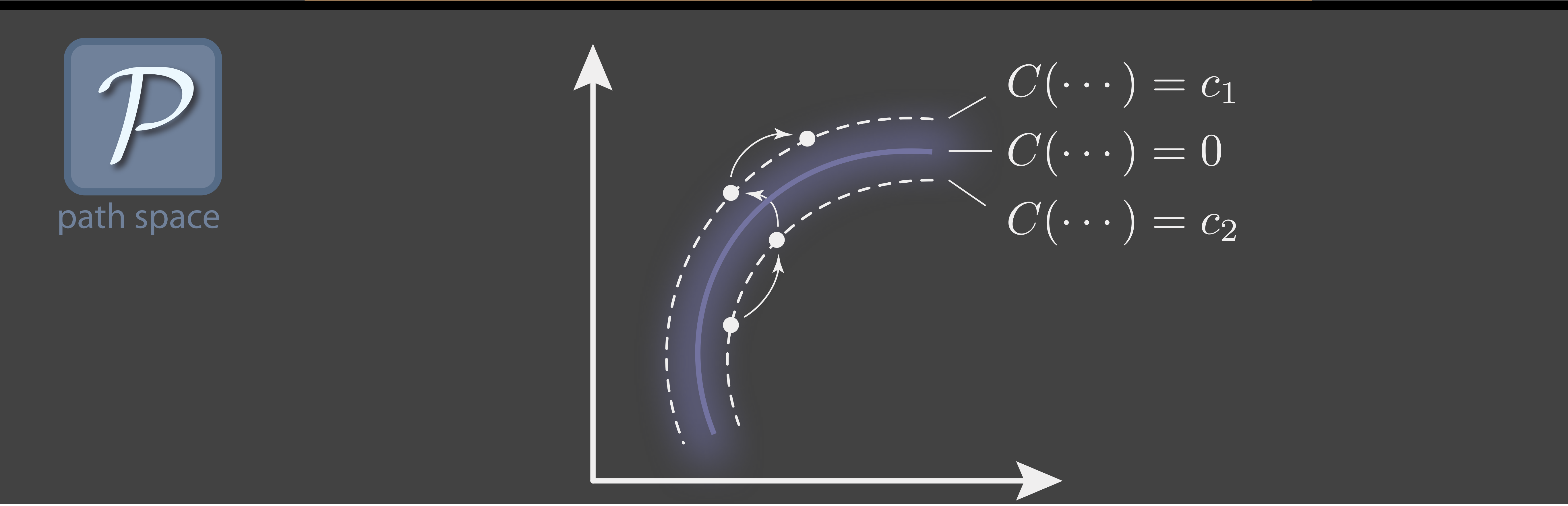


Manifold perturbation



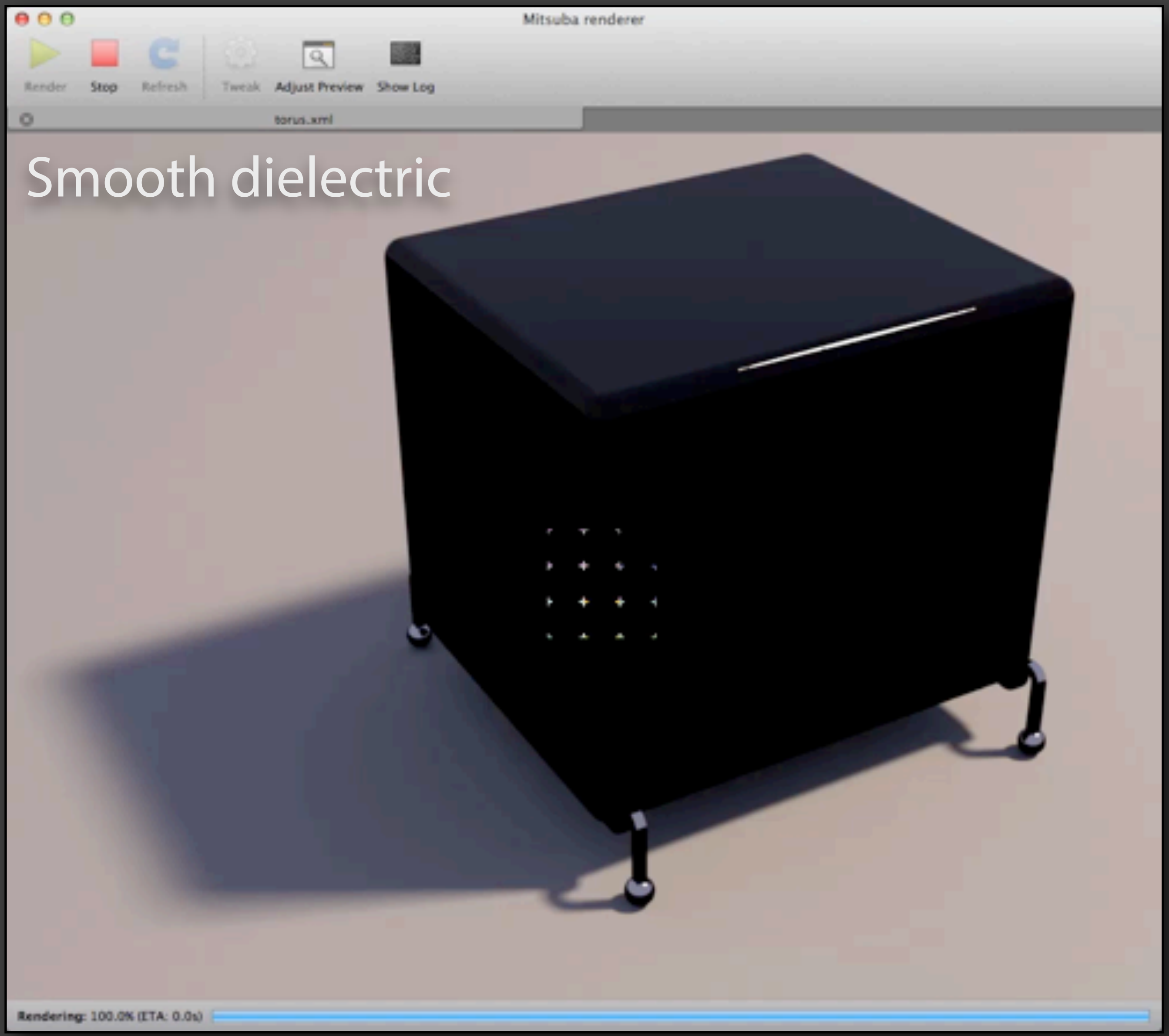


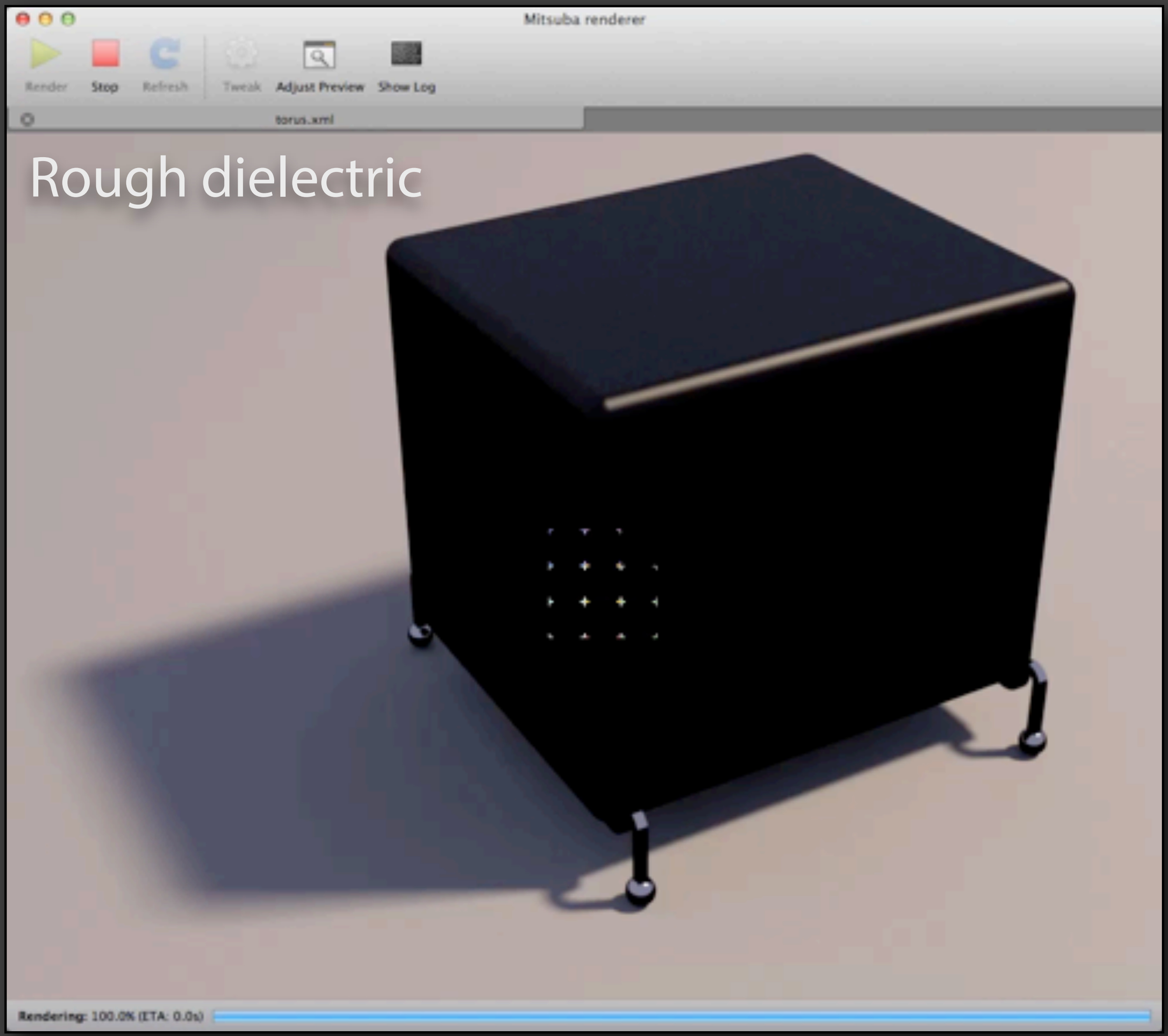
$\mathbb{R}^2$
euclidean space
S
"S"
D



$\mathcal{P}$
path space
$C(\cdots) = c_1$
$C(\cdots) = 0$
$C(\cdots) = c_2$

Manifold Exploration Path Tracing





[15x time-lapse]

Rendering is a hard problem

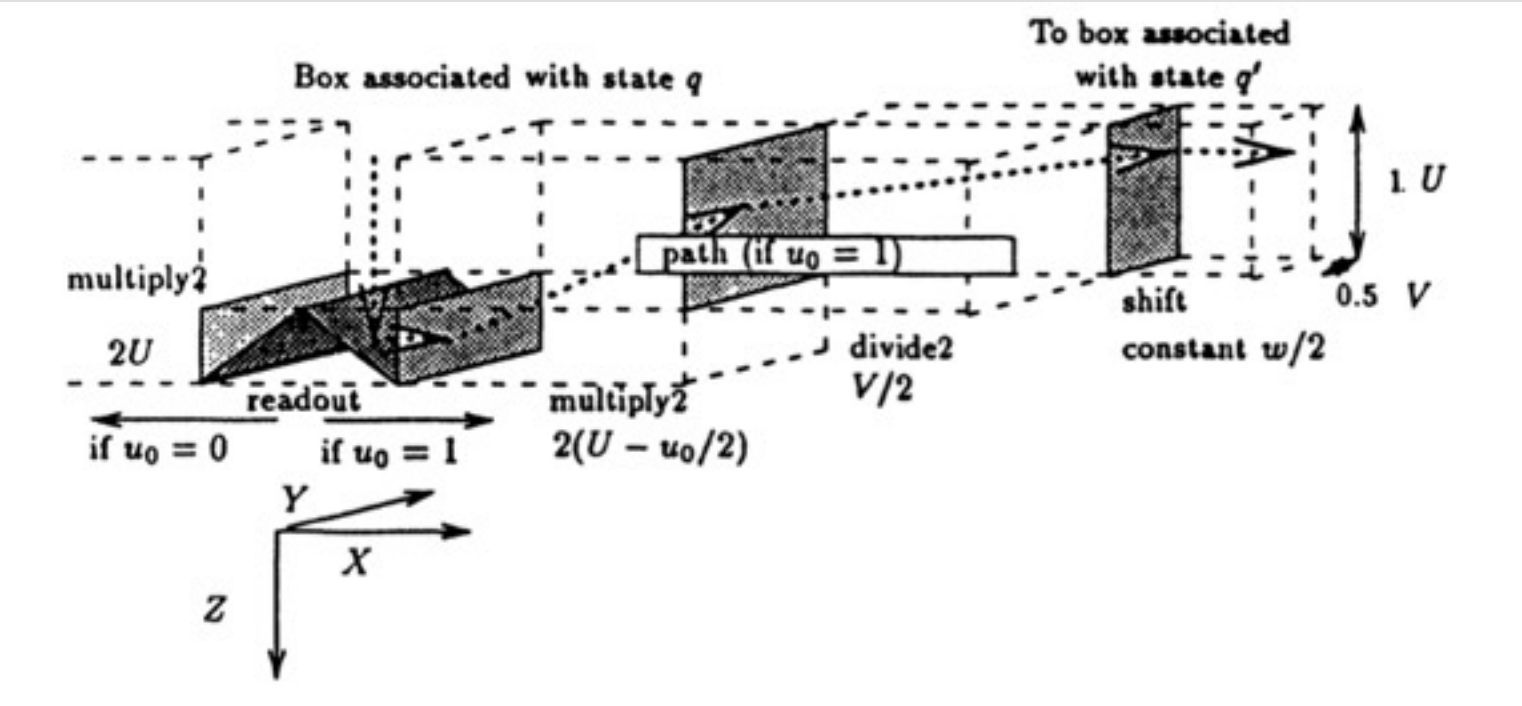
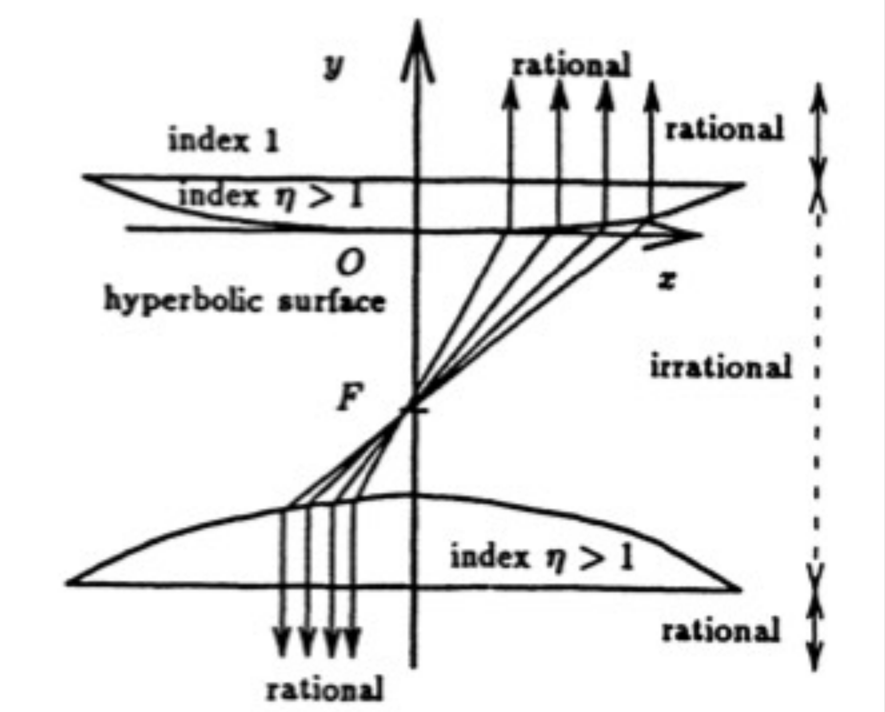


Fig. 8. Implementing $\delta(q, c) = (q', c', L)$.

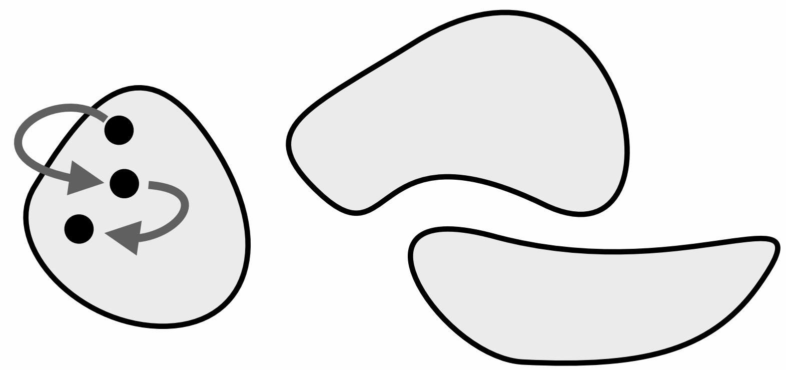


Computability and Complexity of Ray Tracing (Reif et al 1990)

Friday, May 27, 16

Current issues

- **Initialization:** unpredictable convergence (islands)



- **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