Variance Reduction

Computer Graphics
CMU 15-462/15-662, Fall 2016

Last time: Rendering Equation

- Recursive description of incident illumination
- Difficult to integrate; *tour de force* of numerical integration
- Leads to lots of sophisticated integration strategies:
  - sampling strategies
  - variance reduction
  - Markov chain methods
  - ...
- Today: get a glimpse of these ideas
- Also valuable outside rendering!
  - E.g., innovations coming from geometry processing/meshing

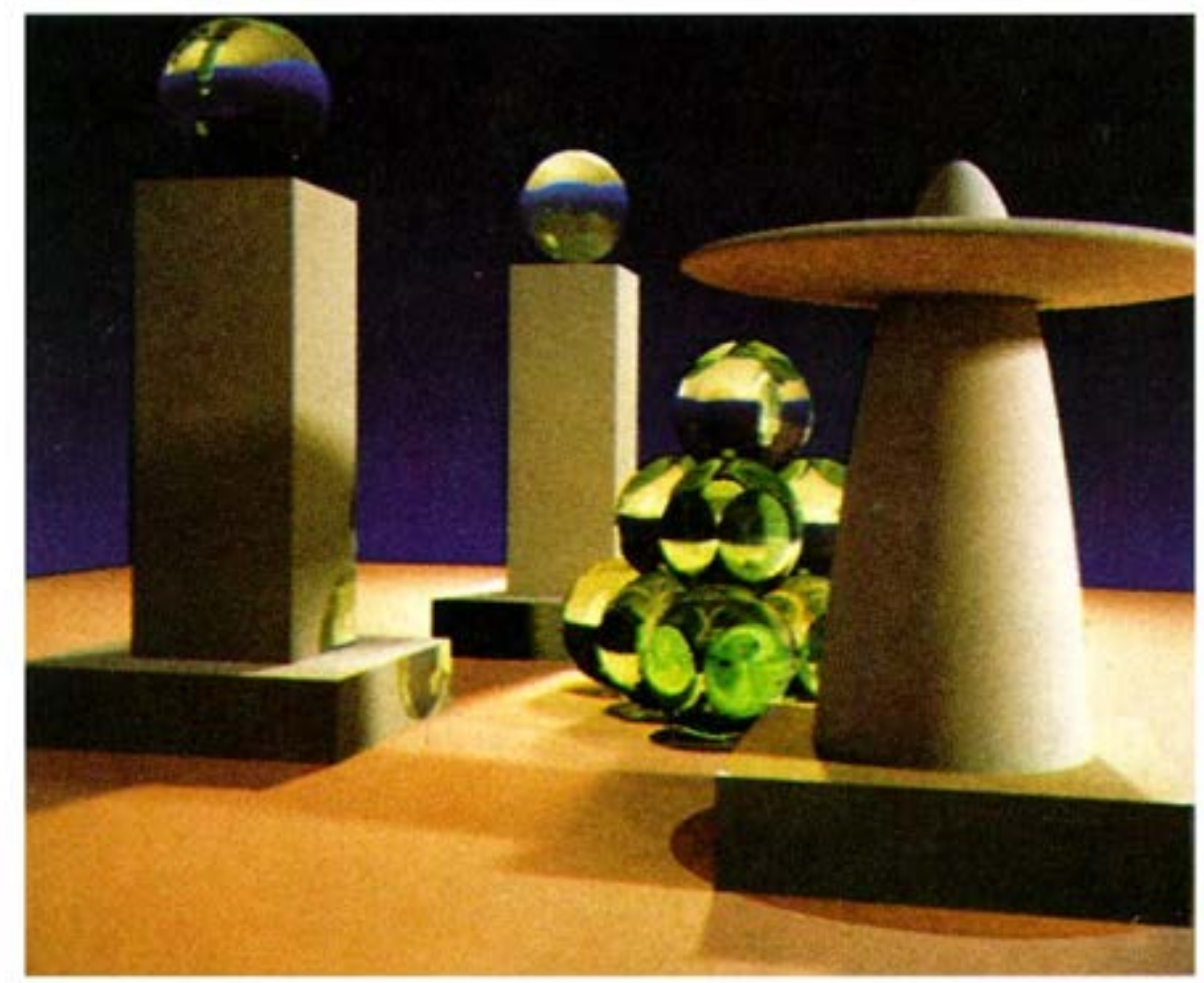


Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.

$$L_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}) = L_e(\mathbf{x}, \omega_{\mathrm{o}}) + \int_{\Omega} f_r(\mathbf{x}, \omega_{\mathrm{i}}, \omega_{\mathrm{o}})\, L_{\mathrm{i}}(\mathbf{x}, \omega_{\mathrm{i}})\, (\omega_{\mathrm{i}} \cdot \mathbf{n})\, \mathrm{d}\,\omega_{\mathrm{i}}$$

Review: Monte Carlo Integration

Want to integrate:

any function

$$I := \int_{\Omega} f(x)\; dx$$

any domain

(*Not* just talking about rendering here, folks!)

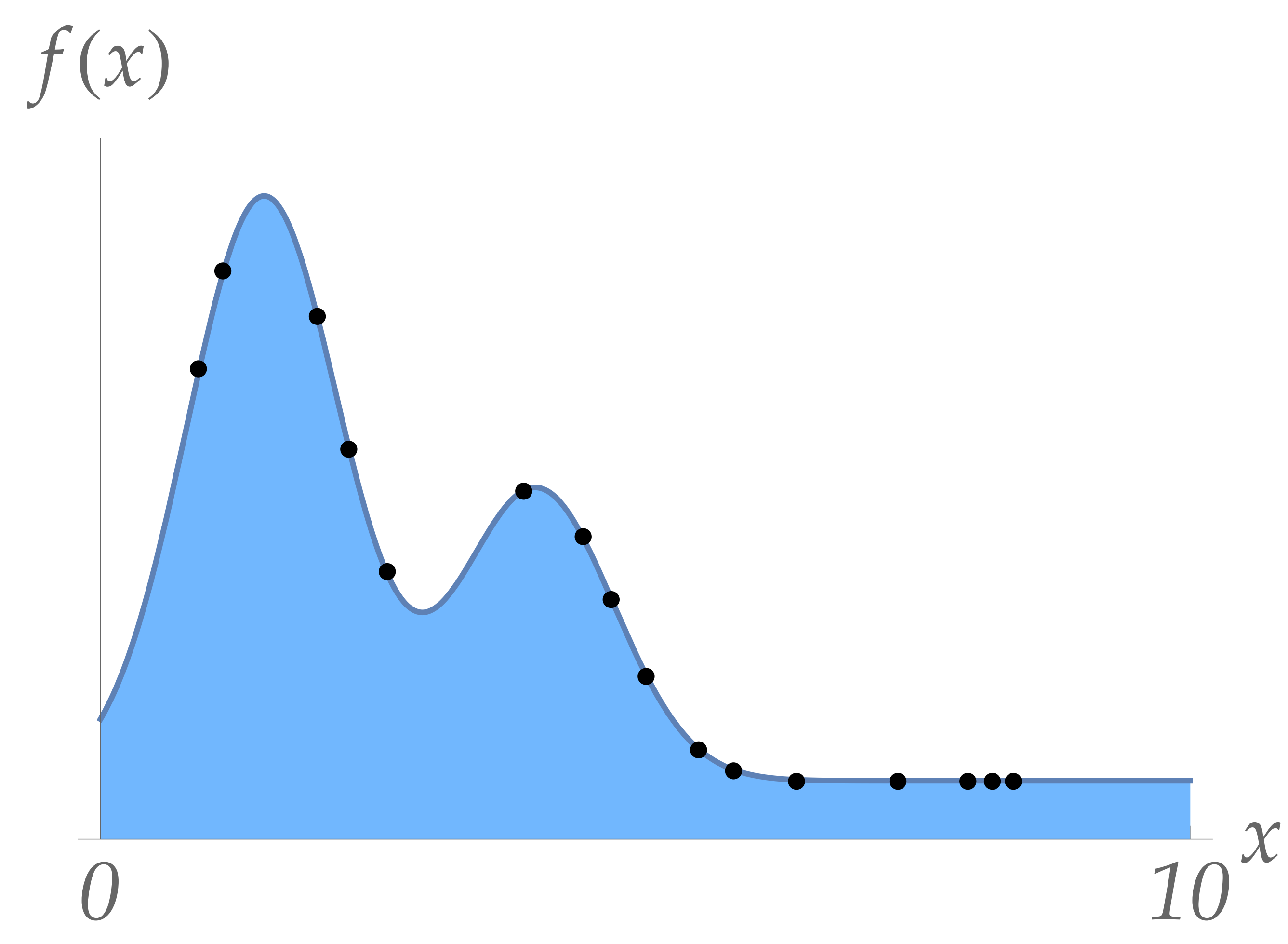
General-purpose hammer: Monte-Carlo integration

$$I = \lim_{n \to \infty} V(\Omega) \frac{1}{n} \sum_{i=1}^{n} f(X_i)$$

under mild conditions on *f*

volume of the domain

uniformly random samples of domain



Review: Expected Value (DISCRETE)

A *discrete* random variable X has n possible outcomes x_i, occuring w/ probabilities $0 \leq p_i \leq 1$, $p_1 + \ldots + p_n = 1$

probability of event i

$$E(X) := \sum_{i=1}^{n} p_i x_i$$

value of event i

expected value

(just the "weighted average"!)

E.g., what's the expected value for a fair coin toss?

$p_1 = 1/2$
$x_1 = 1$

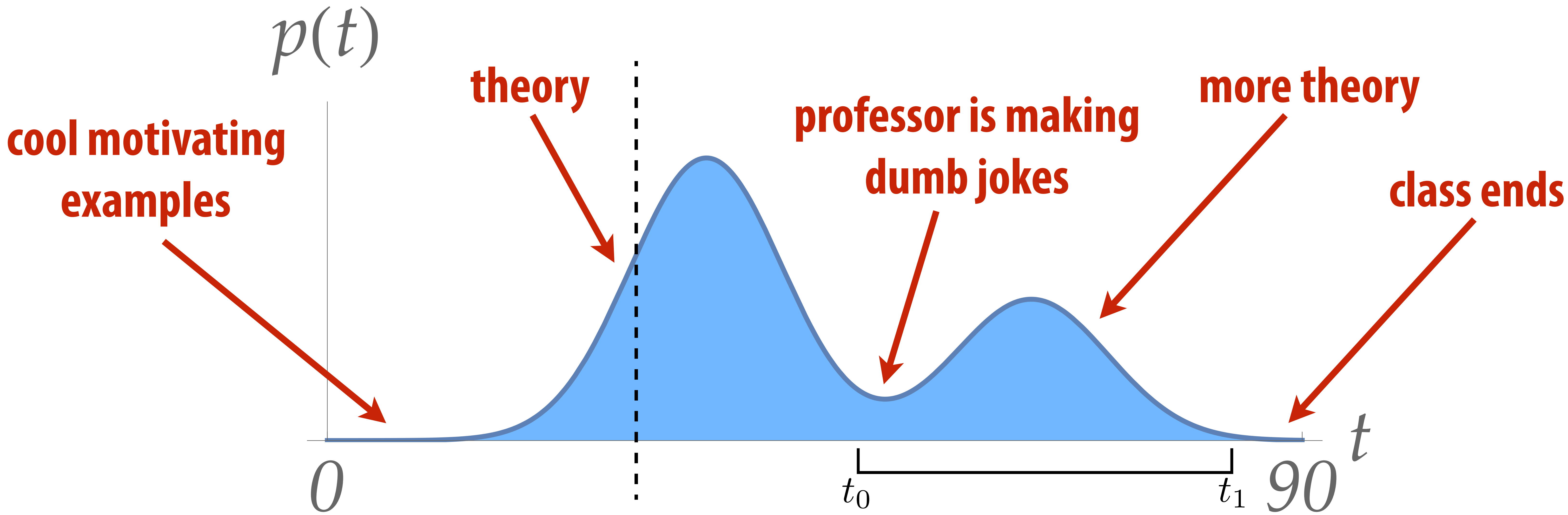
$p_2 = 1/2$
$x_2 = 0$

Review: Continuous Random Variables

A *continuous* random variable *X* takes values *x* anywhere in a set Ω

Probability density *p* gives probability x appears in a given *region*.

E.g., probability you fall asleep at time *t* in a 15-462 lecture:



probability you fall asleep *exactly* at any given time t is ZERO!

can only talk about chance of falling asleep in a given *interval* of time

$$\int_{t_0}^{t1} p(t)\ dt$$

Review: Expected Value (CONTINUOUS)

Expected value of continuous random variable again just the "weighted average" with respect to probability *p*:

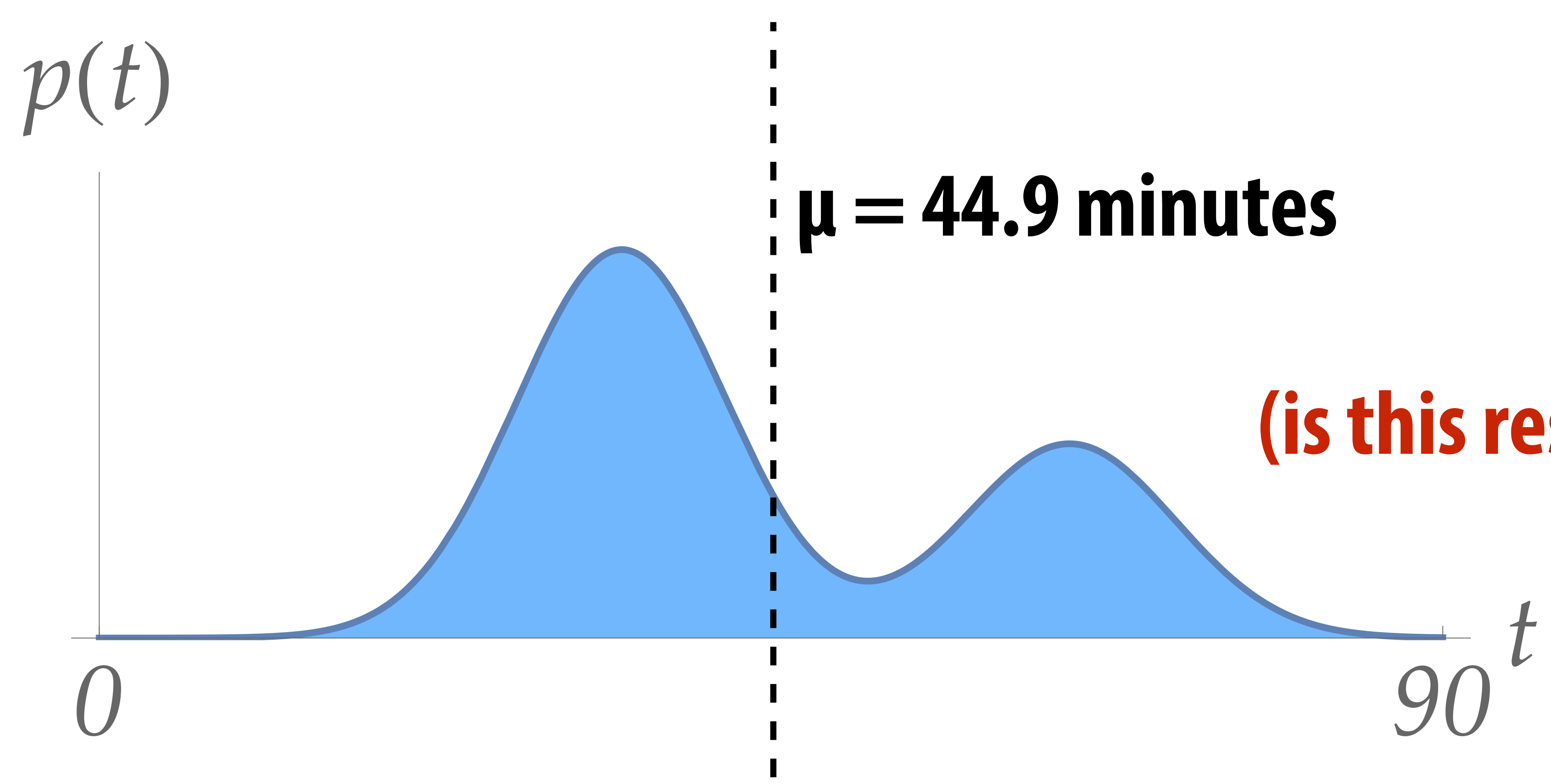
probability density at point x

$$E(X) := \int_\Omega x p(x)\; dx$$

expected value

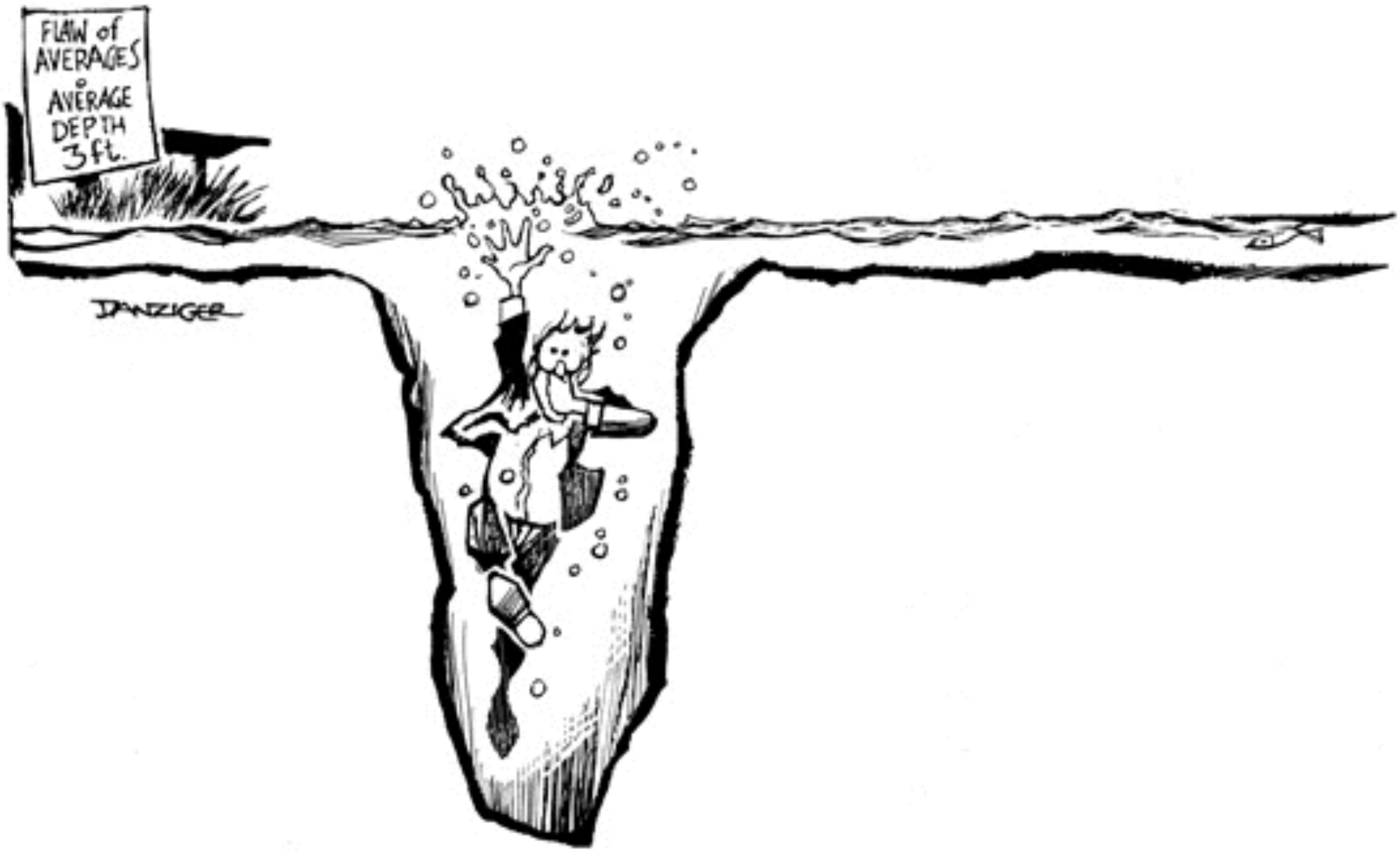
sometimes just use "μ" (for "mean")

E.g., expected time of falling asleep?



(is this result counter-intuitive?)

Flaw of Averages



Review: Variance

- **Expected value is the "average value"**
- **Variance is how far we are from the average, *on average!***

$$\mathrm{Var}(X) := E[(X - E[X])^2]$$

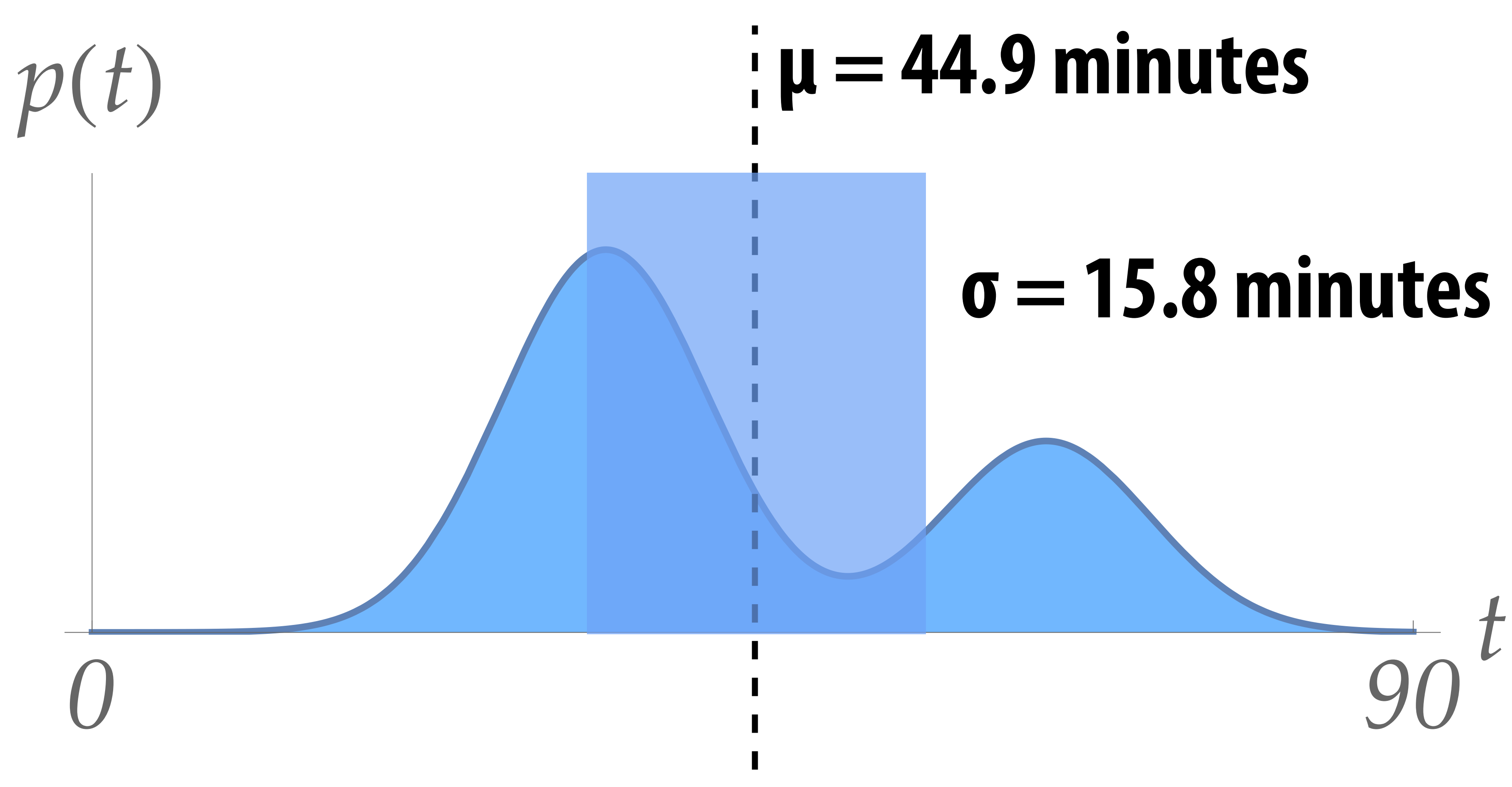
DISCRETE

$$\sum_{i=1}^{n} p_i (x_i - \sum_j p_j x_j)^2$$

CONTINUOUS

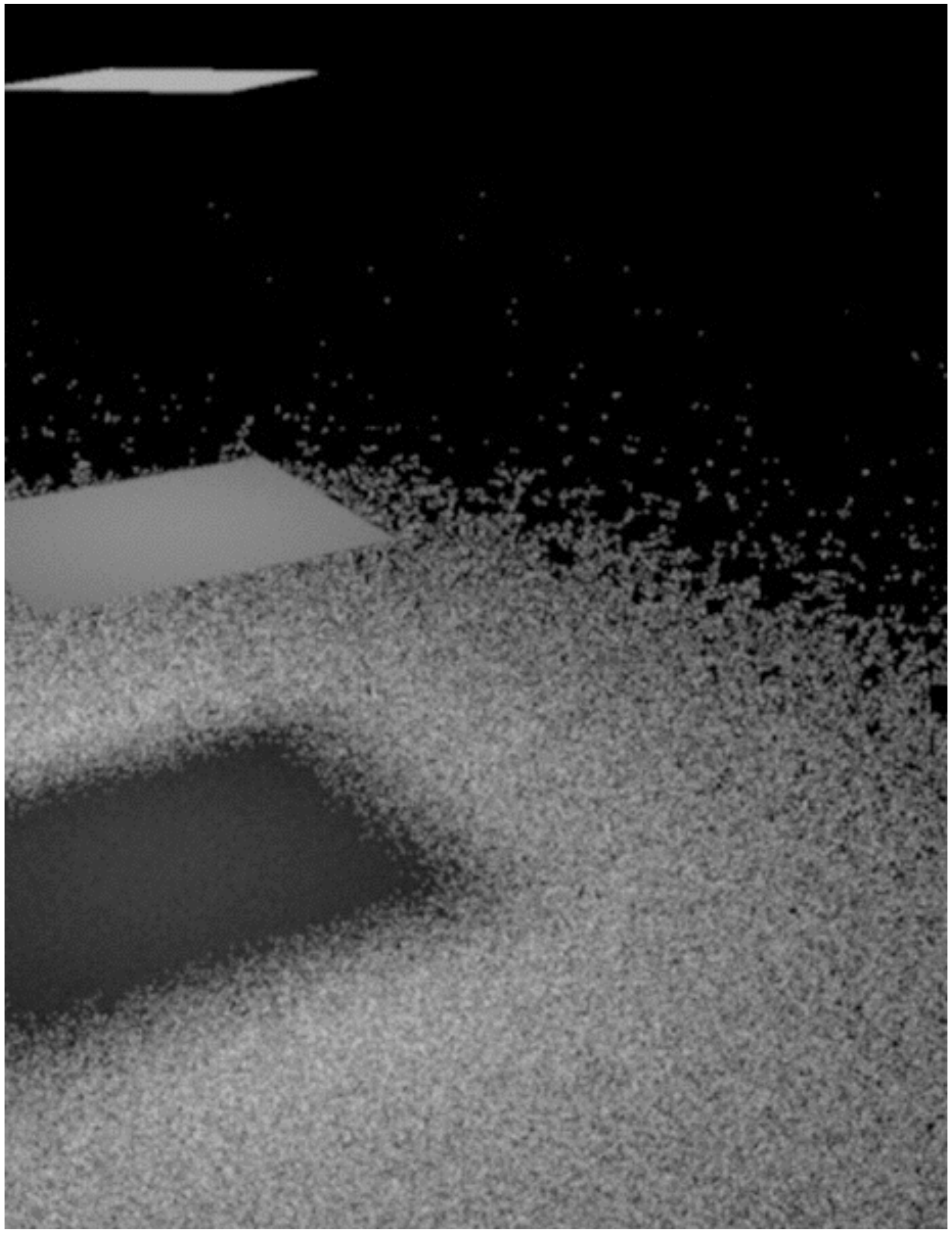
$$\int_\Omega p(x)(x - \int_\Omega y p(y)\ dy)^2\ dx$$

- ***Standard deviation* σ is just the square root of variance**

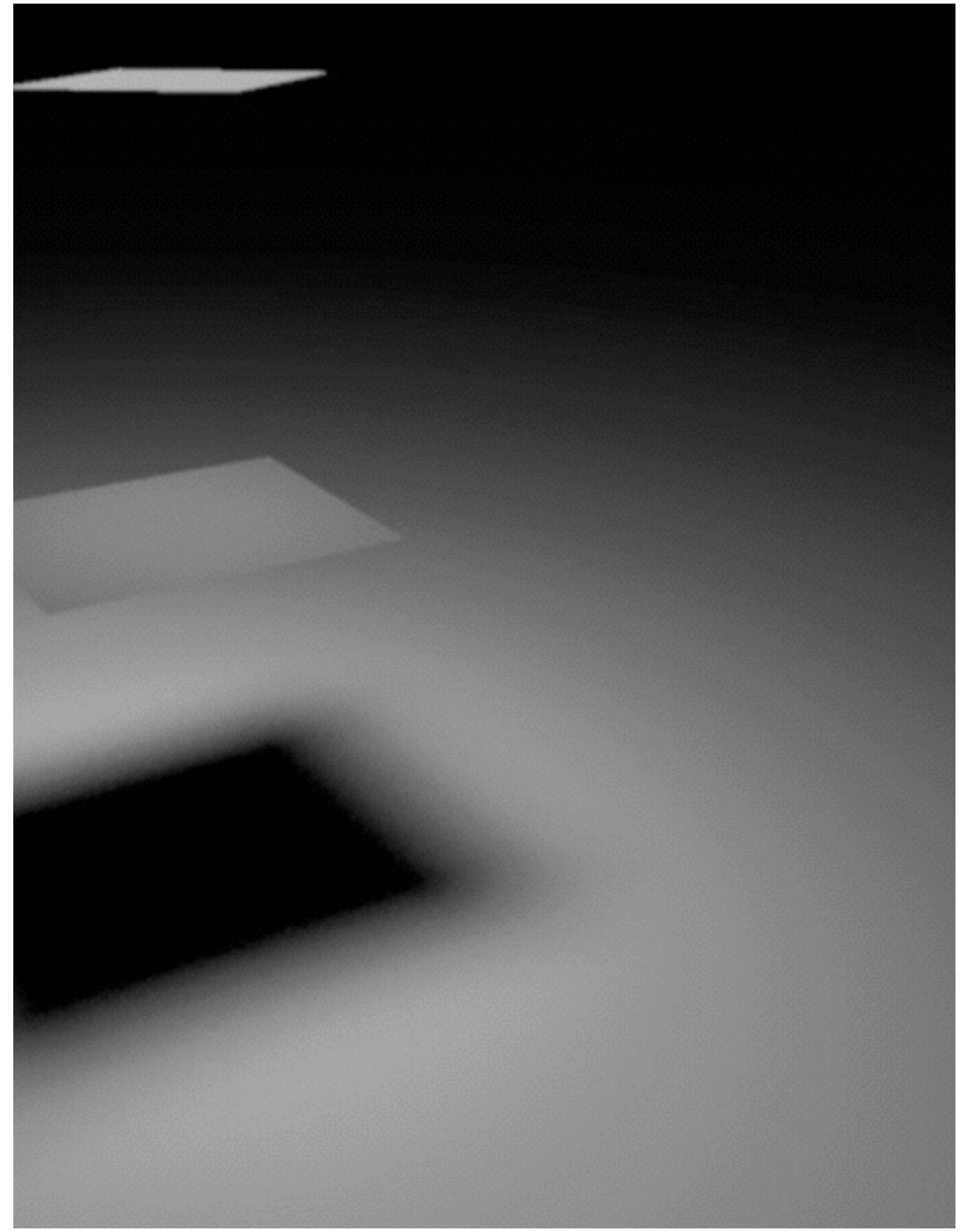


(any more intuitive?)

Variance Reduction in Rendering



higher variance



lower variance

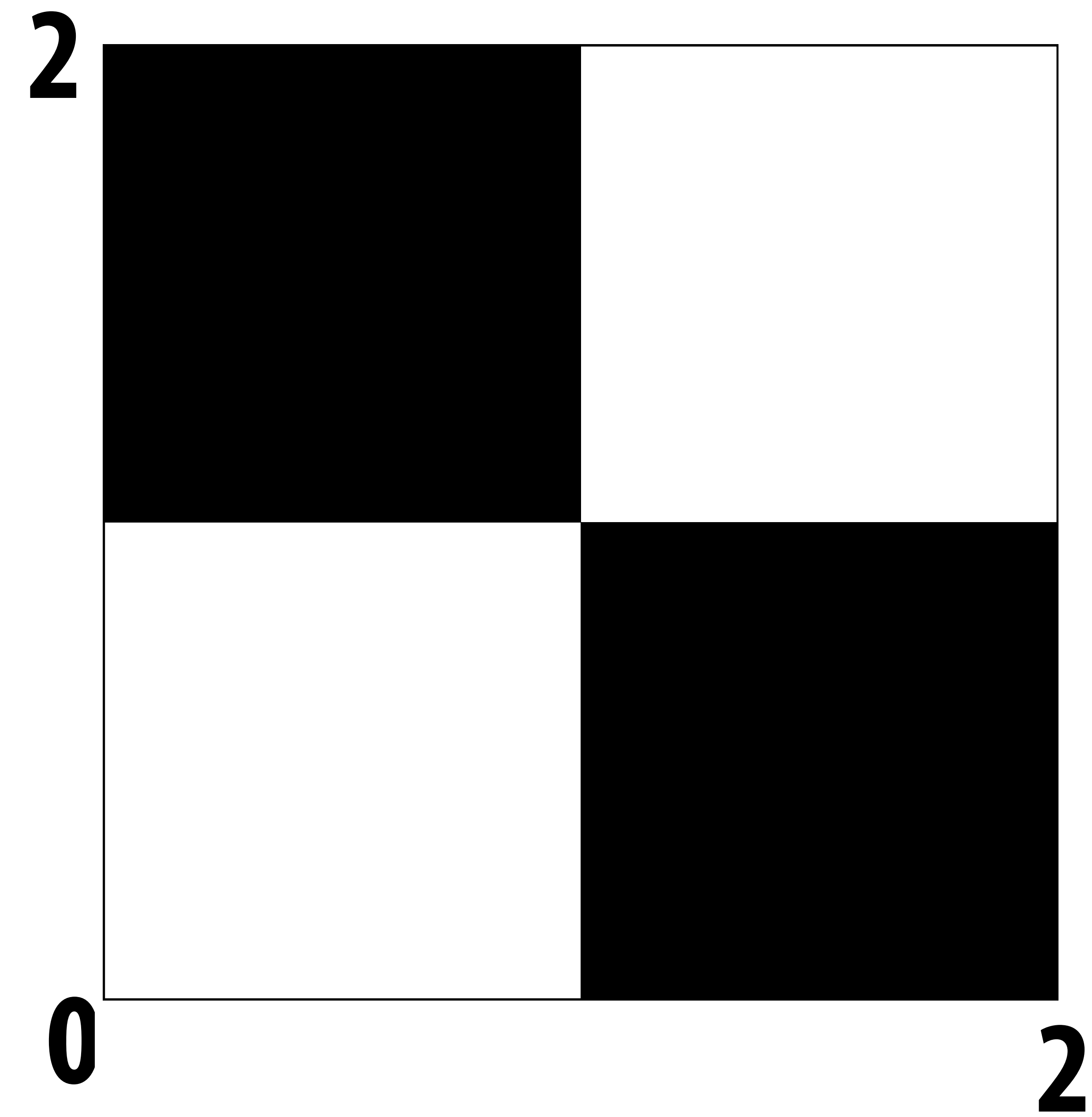
Q: How do we reduce variance?

Variance Reduction Example

$$\Omega := [0,2] \times [0,2]$$

$$f(x,y) := \begin{cases} 1 & \lfloor x \rfloor + \lfloor y \rfloor \text{ is even,} \\ 0 & \text{otherwise} \end{cases}$$

$$I := \int_{\Omega} f(x,y)\ dxdy$$



Q: What's the expected value of the integrand *f*?

A: Just by inspection, it's 1/2 (half white, half black!).

Q: What's its variance?

A: $(1/2)(0-1/2)^2 + (1/2)(1-1/2)^2 = (1/2)(1/4) + (1/2)(1/4) = 1/4$

Q: How do we reduce the variance?

That was a trick question.

You can't reduce variance of the *integrand!*
Can only reduce variance of an *estimator*.

Variance of an Estimator

- An "estimator" is a formula used to approximate an integral
- Most important example: our Monte Carlo estimate:

$$I = \int_\Omega f(x)\ dx \qquad\qquad \hat{I} := V(\Omega)\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

true integral — Monte Carlo estimate

- Get different estimates for different collections of samples
- Want to reduce variance of *estimate* across different samples
- Why? Integral itself only has one value!
- Many, many (many) techniques for reducing variance
- We will review some key examples for rendering

Bias & Consistency

- Two important things to ask about an estimator
  - Is it *consistent*?
  - Is it *biased*?
- Consistency: "converges to the correct answer"

estimate

$$\lim_{n\to\infty} P(|I - \hat{I}_n| > 0) = 0$$

true integral

# of samples

- Unbiased: "estimate is correct on average"

$$E[I - \hat{I}_n] = 0$$

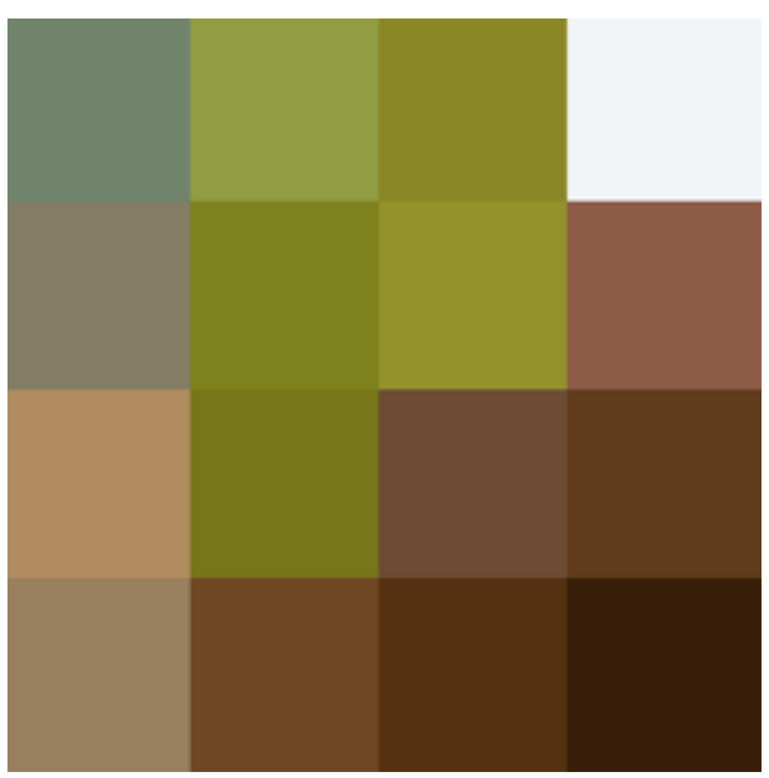
expected value

...even if n=1! (only one sample)

- Consistent does not imply unbiased!

Example 1: Consistent or Unbiased?

- My estimator for the integral over an image:
  - take $n = m \times m$ samples at fixed grid points
  - sum the contributions of each box
  - let m go to ∞



$m = 4$

$m = 16$

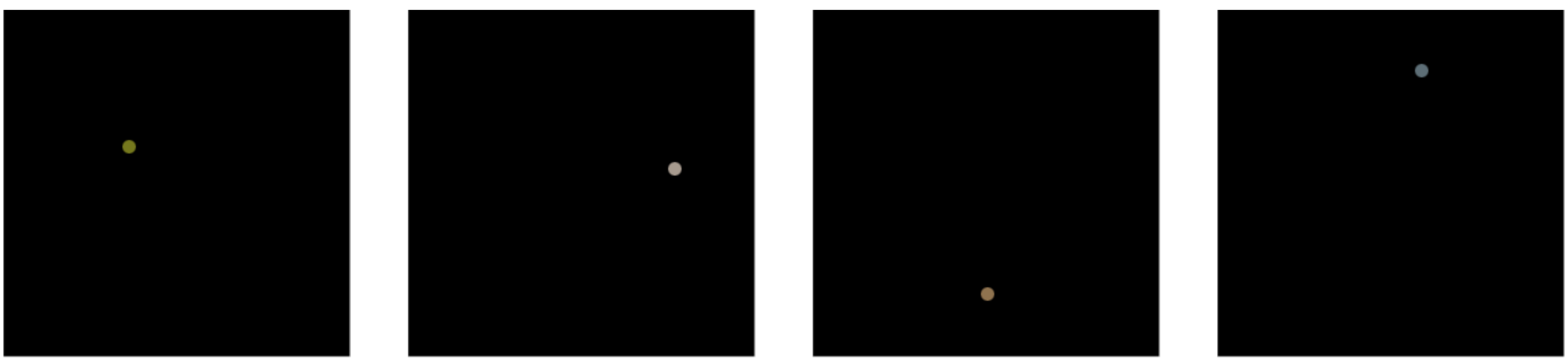
$m = 64$

$m = \infty$

Is this estimator consistent? Unbiased?

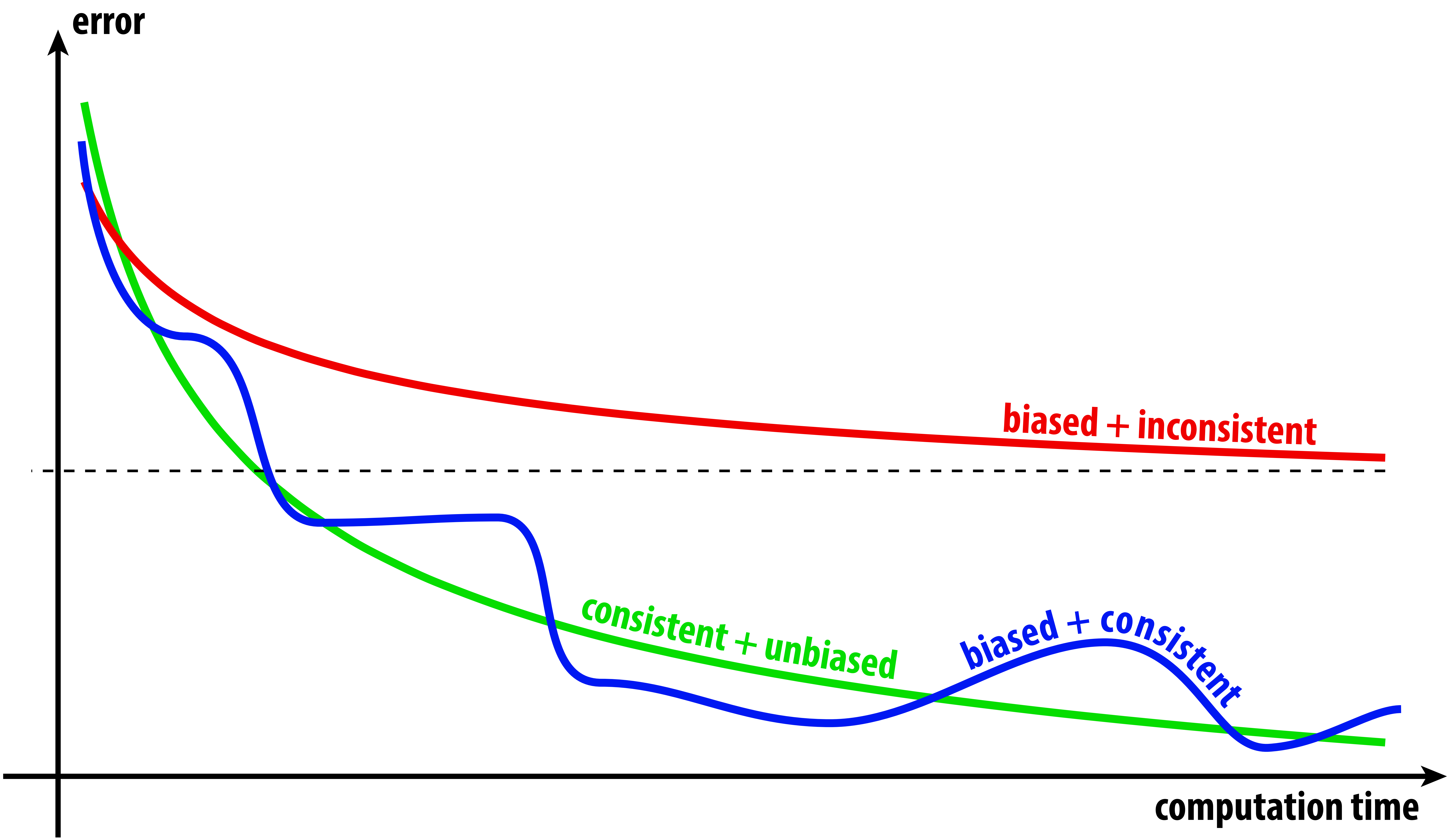
Example 2: Consistent or Unbiased?

- My estimator for the integral over an image:
  - take only a **single** random sample of the image (n=1)
  - multiply it by the image area
  - use this value as my estimate



Is this estimator consistent? Unbiased?
(What if I then let n go to ∞?)

Why does it matter?



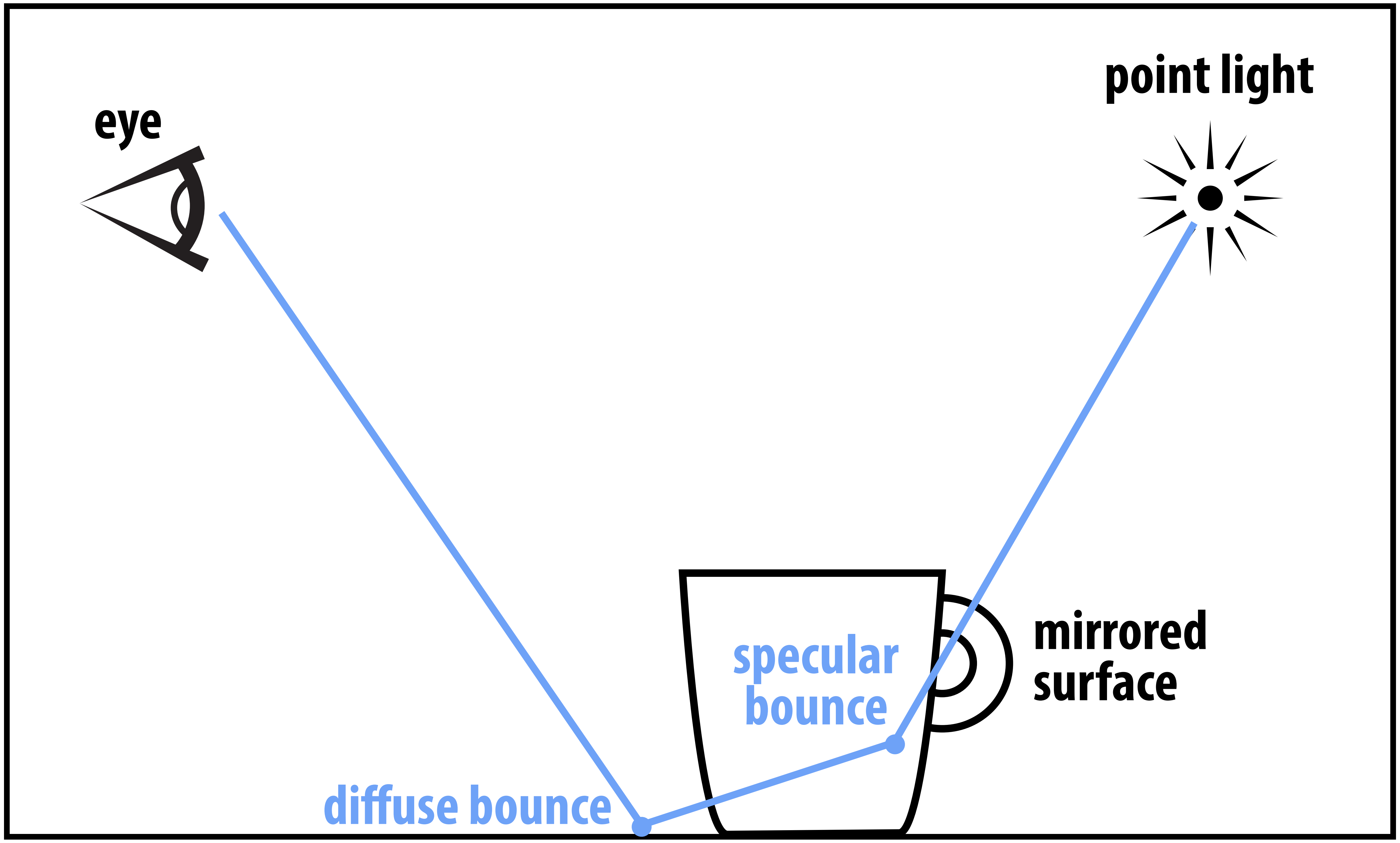
Rule of thumb: unbiased estimators have more predictable behavior / fewer parameters to tweak to get correct result (which says nothing about *performance*...)

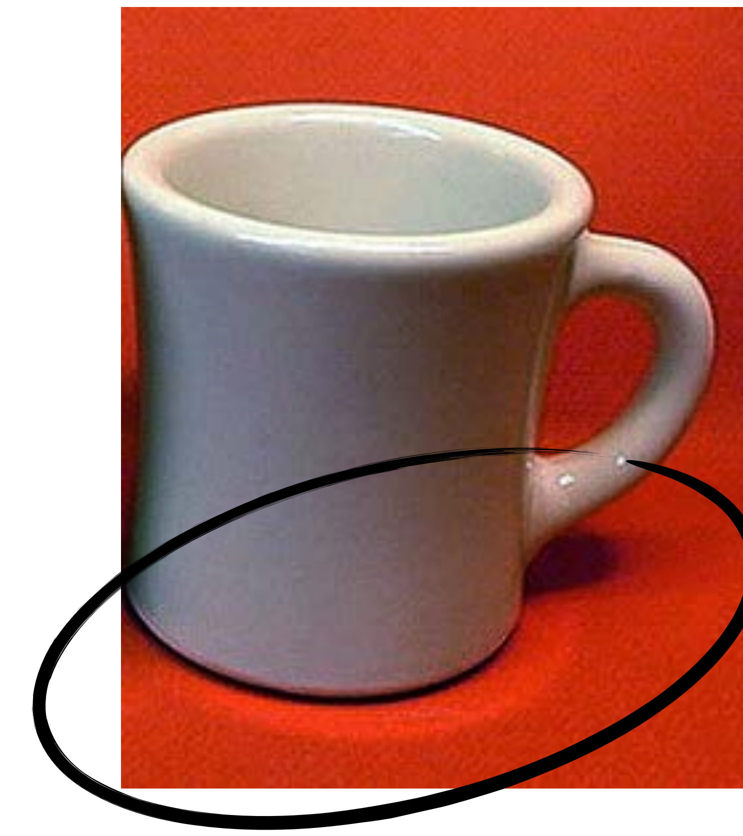
Consistency & Bias in Rendering Algorithms

method	consistent?	unbiased?
rasterization*	NO	NO
path tracing	ALMOST	ALMOST
bidirectional path tracing	???	???
Metropolis light transport	???	???
photon mapping	???	???
radiosity	???	???

*But *very* high performance!

Naïve Path Tracing: Which Paths Can We Trace?





"caustic" (focused light) from reflection

Q: What's the probability we sample the reflected direction?

A: ZERO.

Q: What's the probability we hit a point light source?

A: ZERO.

Naïve path tracing misses important phenomena!
(Formally: the result is *biased*.)

...But isn't this example pathological?
No such thing as point light source, perfect mirror.

Real lighting can be close to pathological

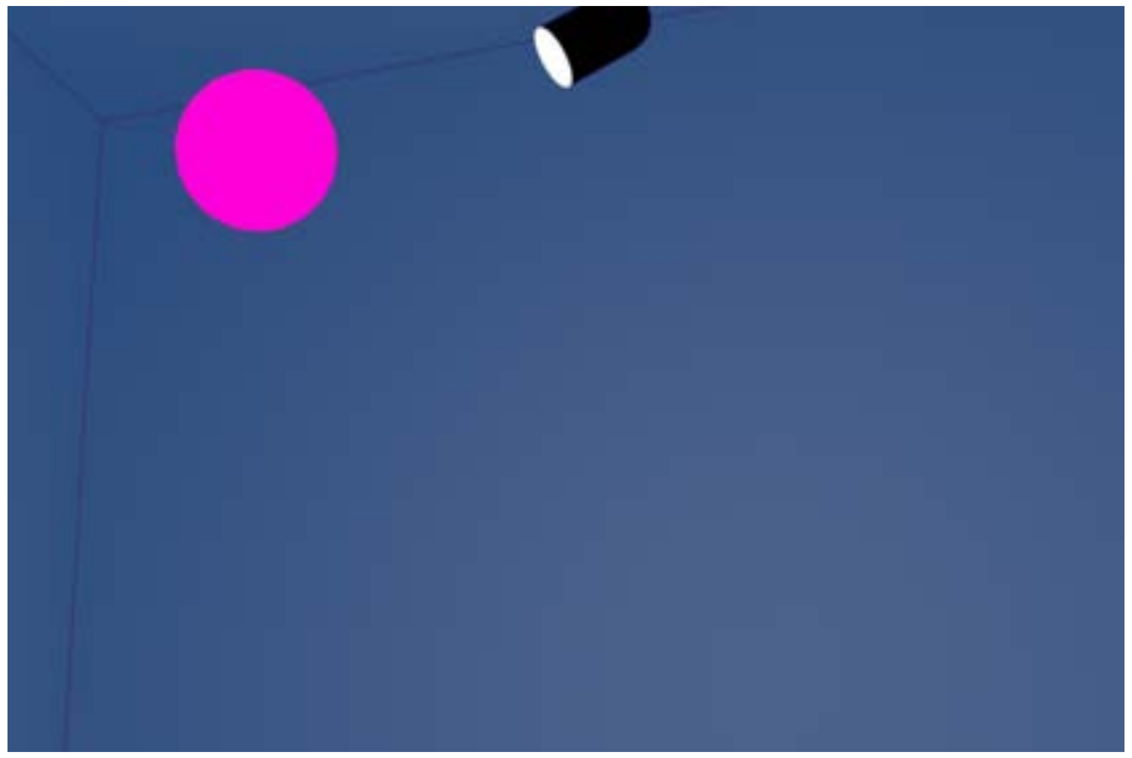
small directional
light source

near-perfect mirror

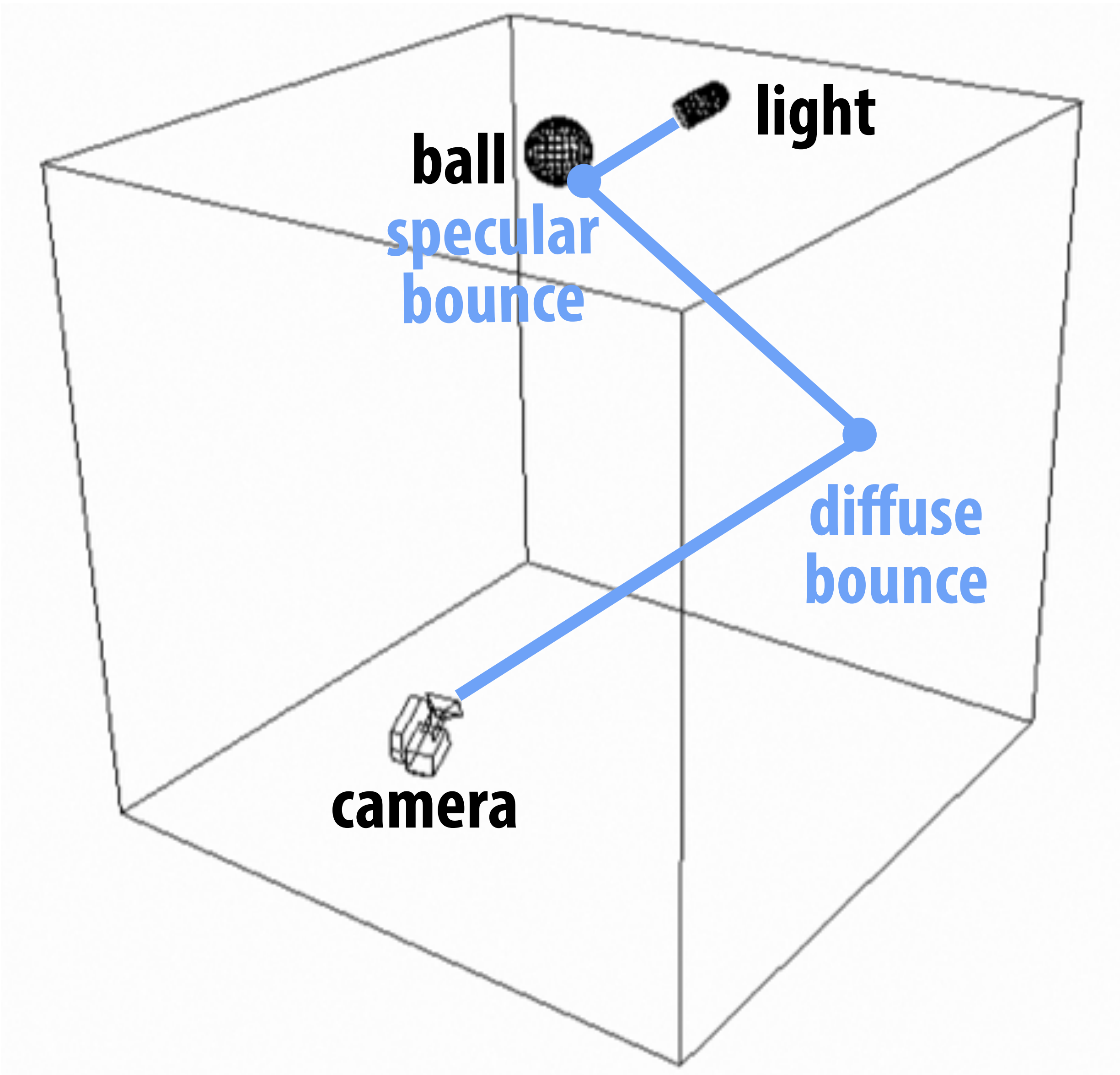
Still want to render this scene!

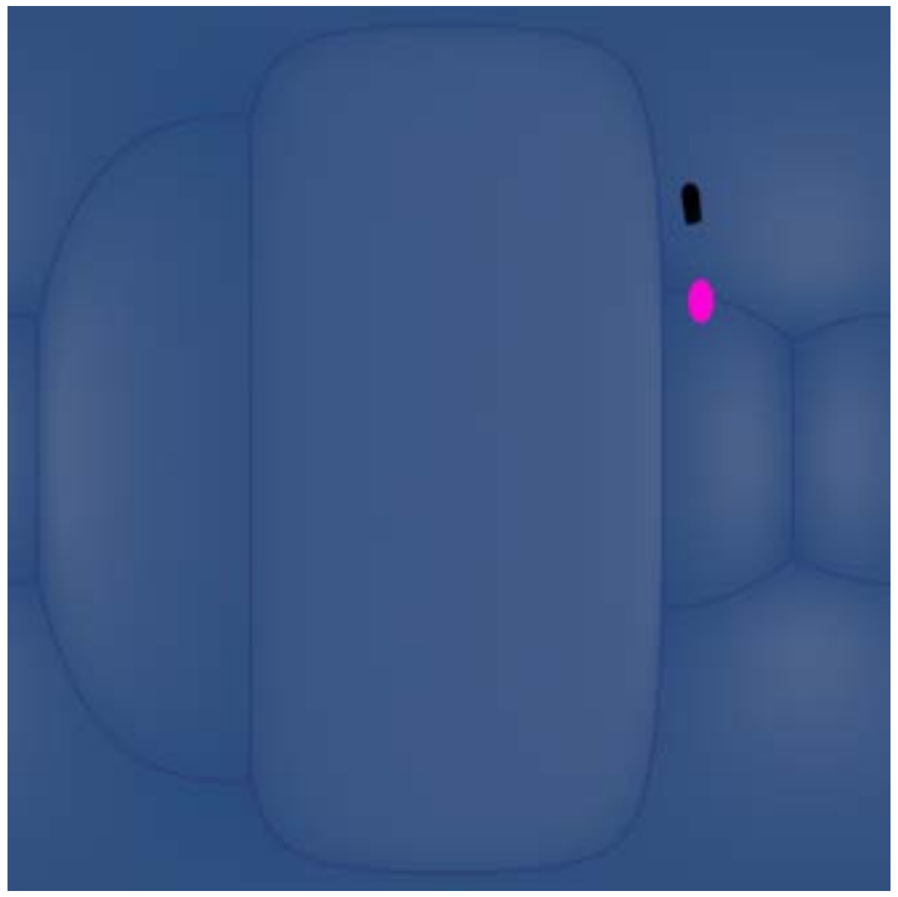
Light has a very "spiky" distribution

- Consider the view from each bounce in our disco scene:

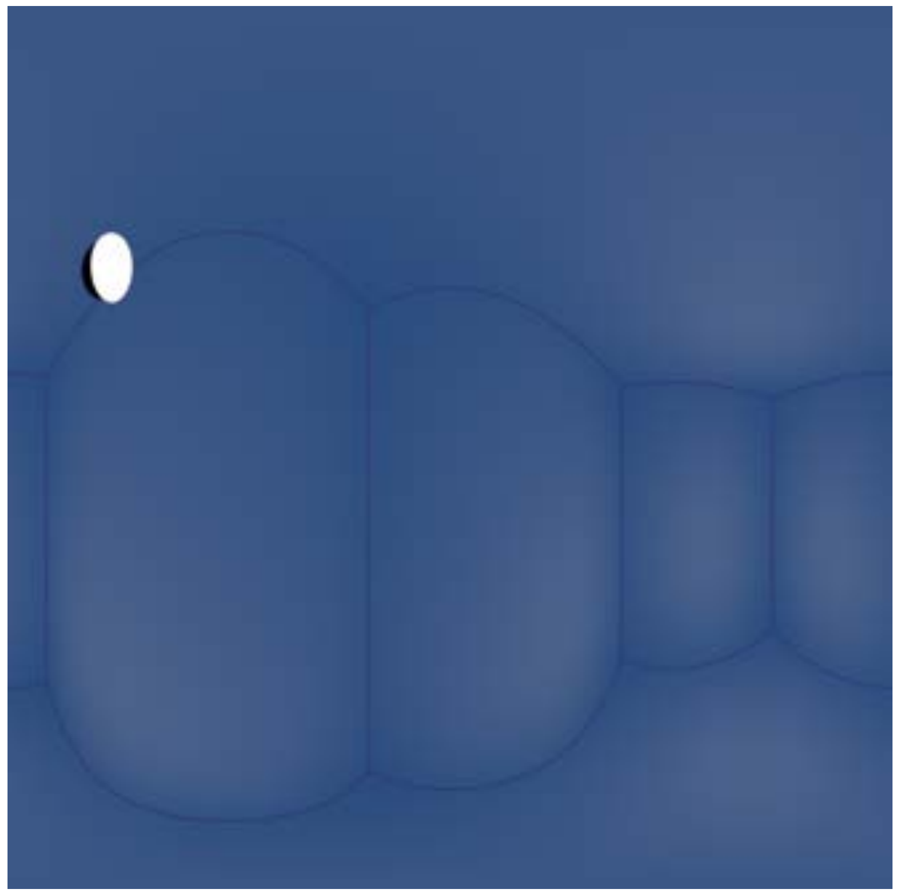


view from camera





view from diffuse bounce
mirrored ball (pink) covers small percentage of solid angle

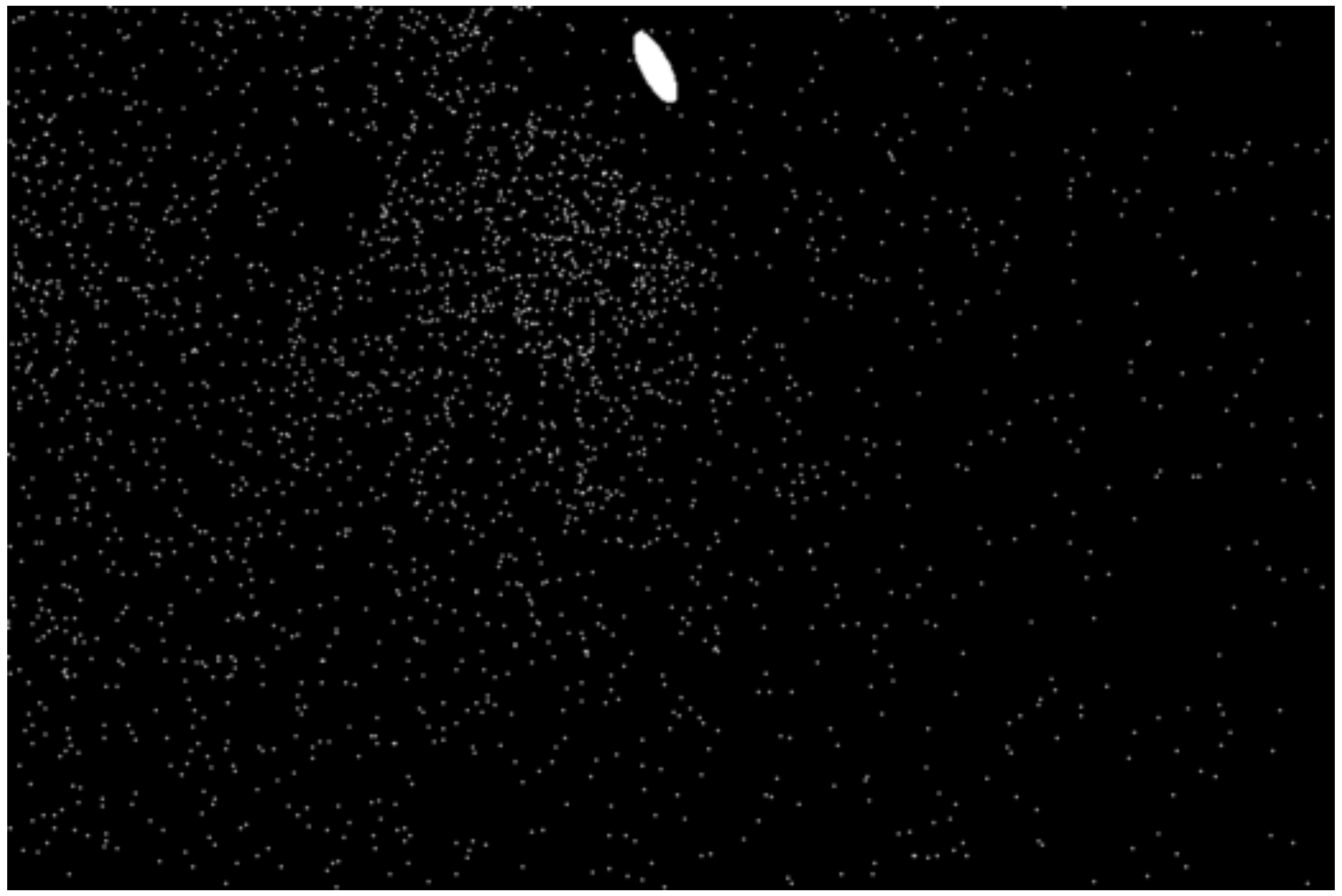


view from specular bounce
area light (white) covers small percentage of solid angle

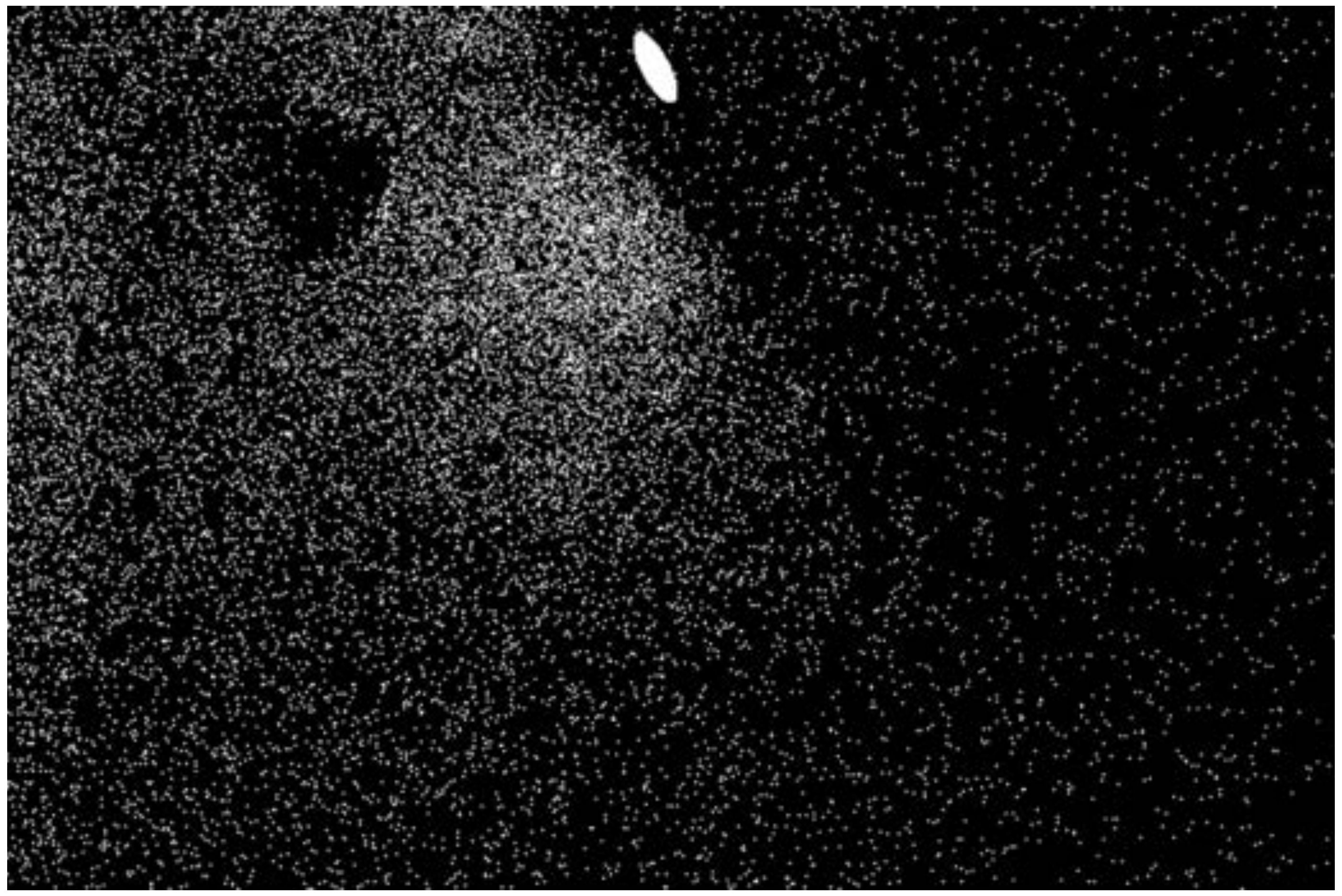
Probability that a uniformly-sampled path carries light is the *product* of the solid angle fractions. (Very small!)

Then consider even more bounces...

Just use more samples?



path tracing - 16 samples/pixel



path tracing - 128 samples/pixel

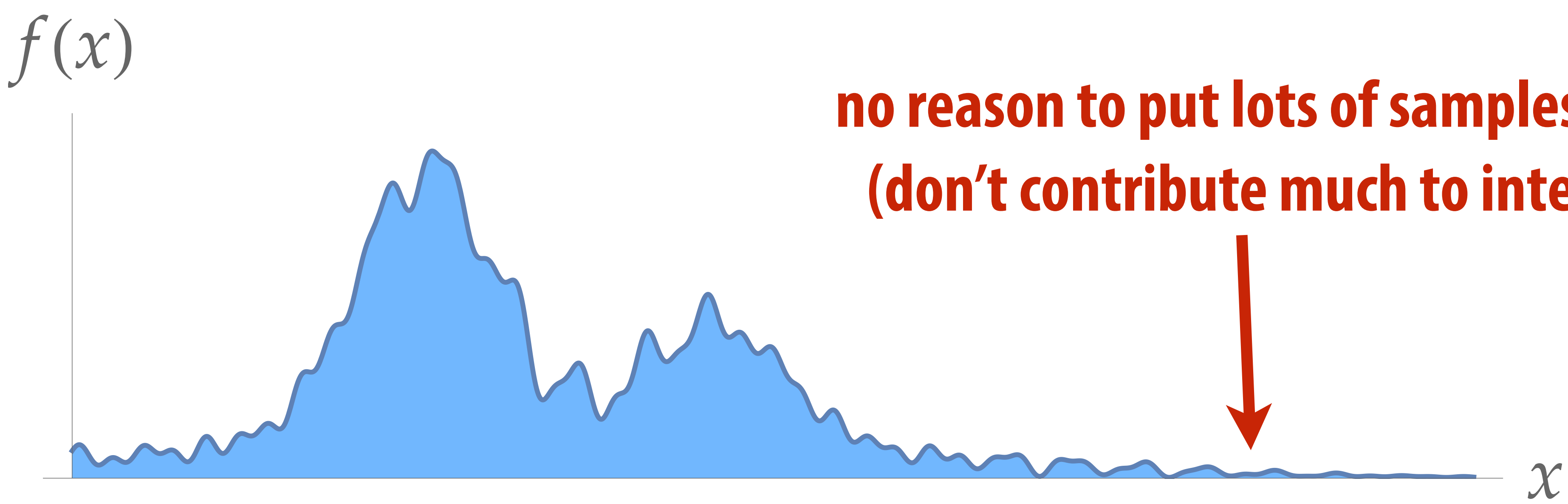
path tracing - 8192 samples/pixel

how do we get here? (photo)

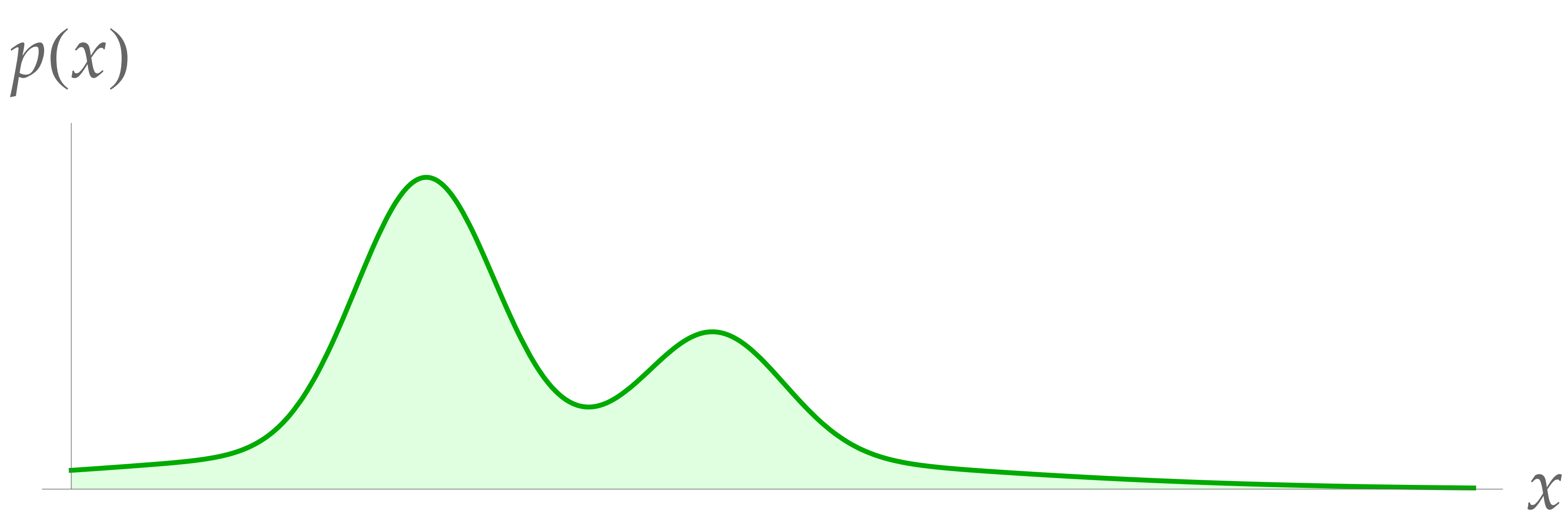
We need better sampling strategies!

Review: Importance Sampling

- **Simple idea: sample the integrand according to how much we expect it to contribute to the integral.**



complicated integrand



our best guess for where the integrand is "big"

naïve Monte Carlo:

$$V(\Omega)\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

(x_i are sampled *uniformly*)

importance sampled Monte Carlo:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p(x_i)}$$

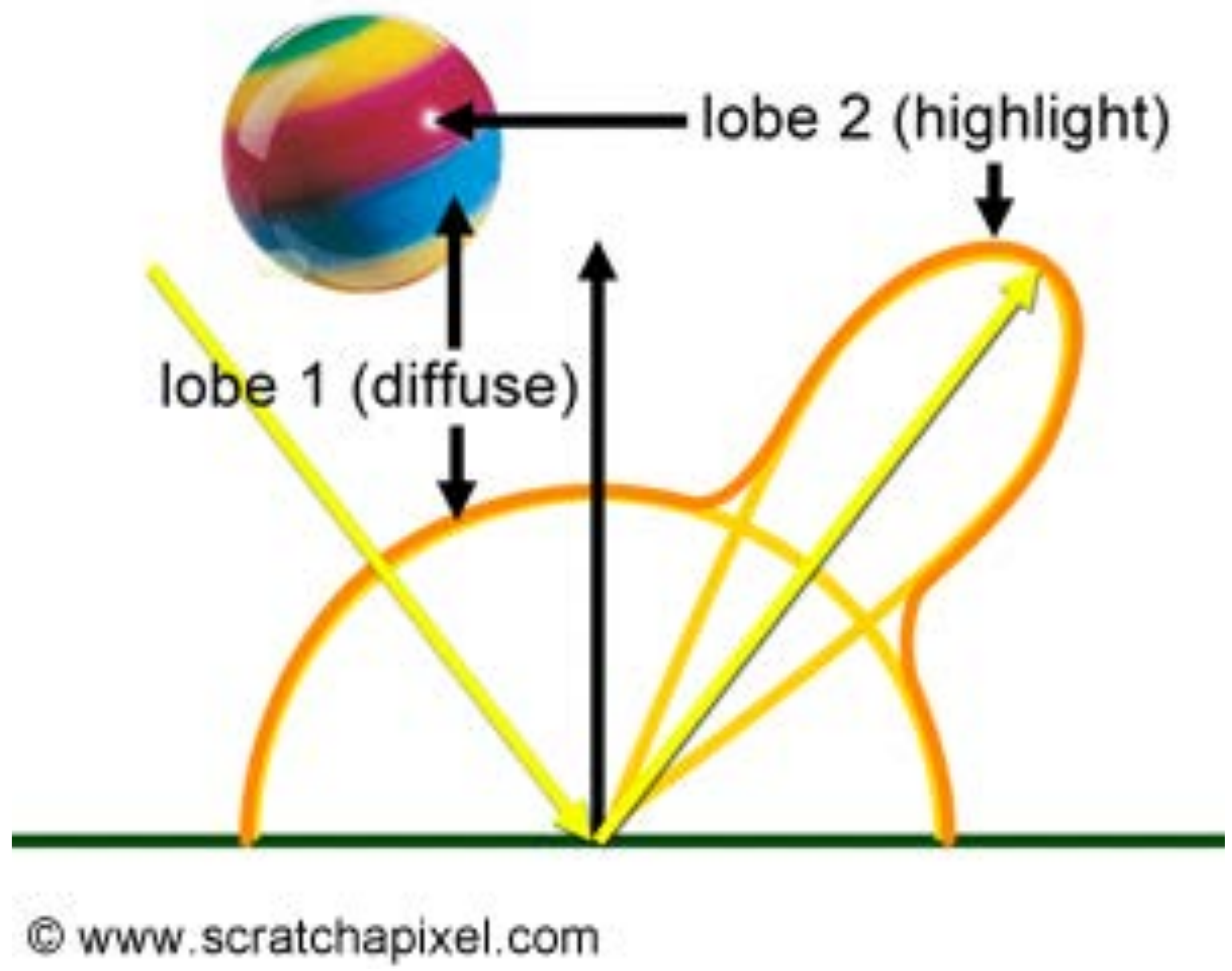
(x_i are sampled proportional to *p*)

"If I sample x more frequently, each sample should count for less; if I sample x less frequently, each sample should count for more."

Q: What happens when *p* is proportional to *f* (*p* = *cf*)?

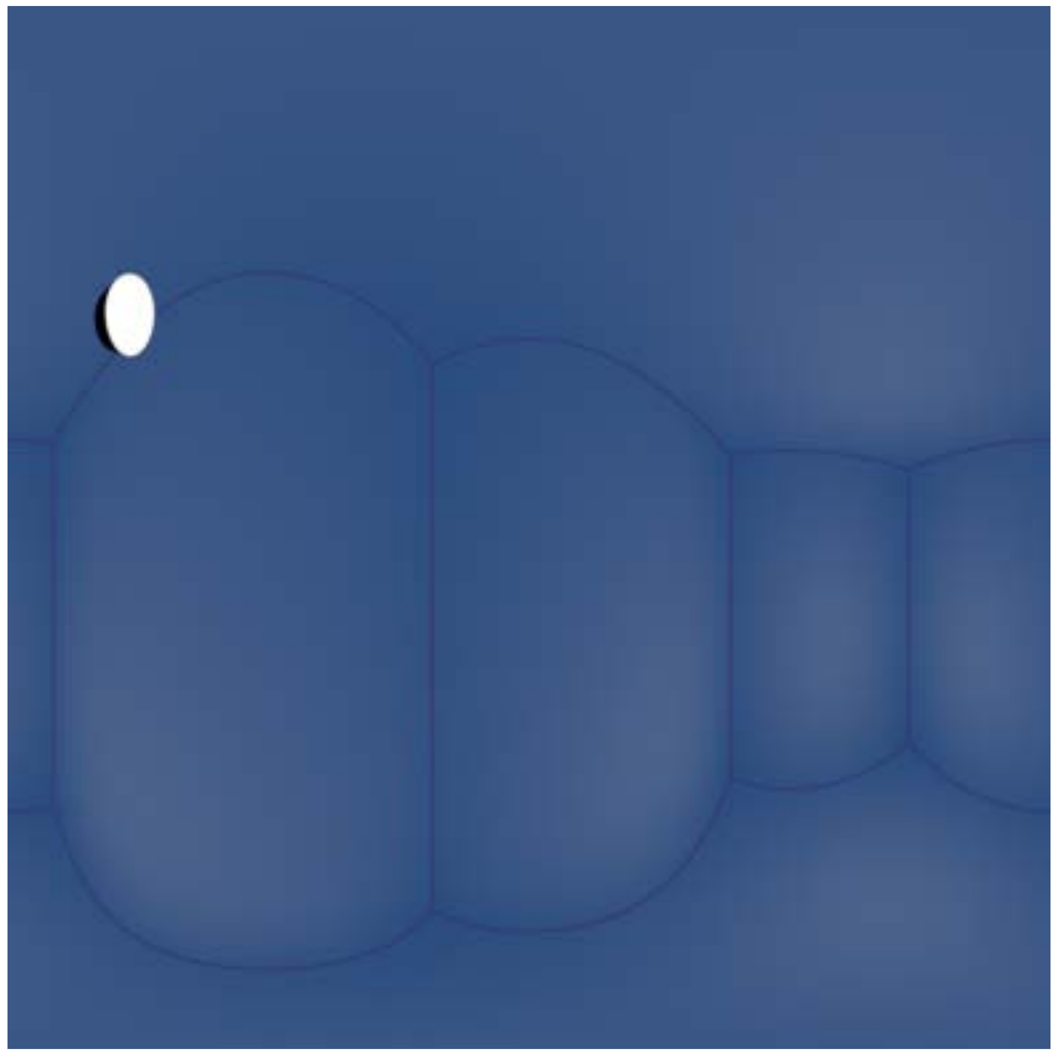
Importance Sampling in Rendering

materials: sample important "lobes"



(important special case: perfect mirror!)

illumination: sample bright lights



Q: How else can we re-weight our choice of samples?

Path Space Formulation of Light Transport

- **So far have been using recursive rendering equation:**

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o)\, L_i(\mathbf{x}, \omega_i)\, (\omega_i \cdot \mathbf{n})\, d\omega_i$$

- **Make intelligent "local" choices at each step (material/ lights)**
- **Alternatively, we can use a "path integral" formulation:**

how much "light" is carried by this path?

$$I = \int_\Omega f(\bar{x}) d\mu(\bar{x})$$

how much of path space does this path "cover"

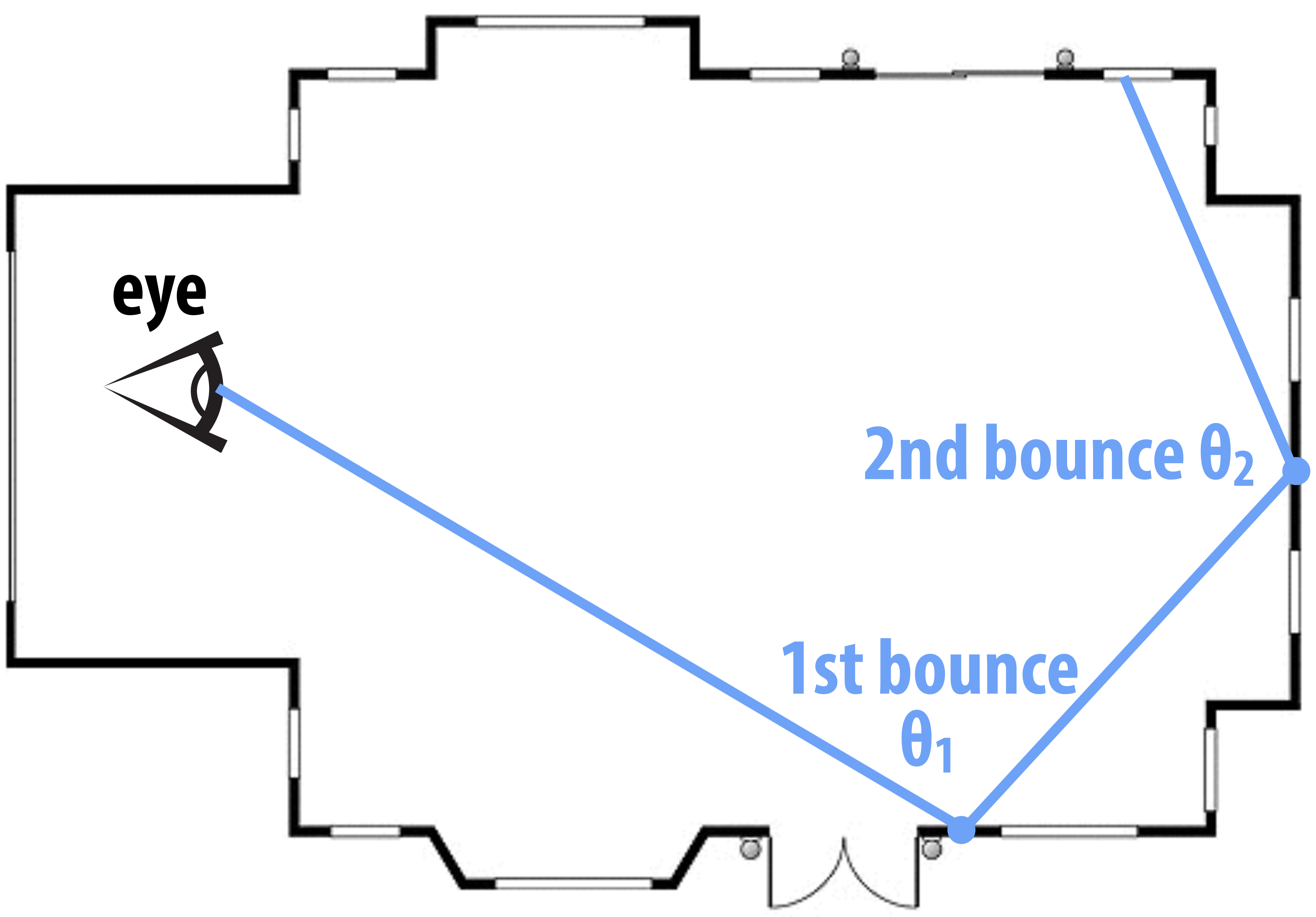
one particular path

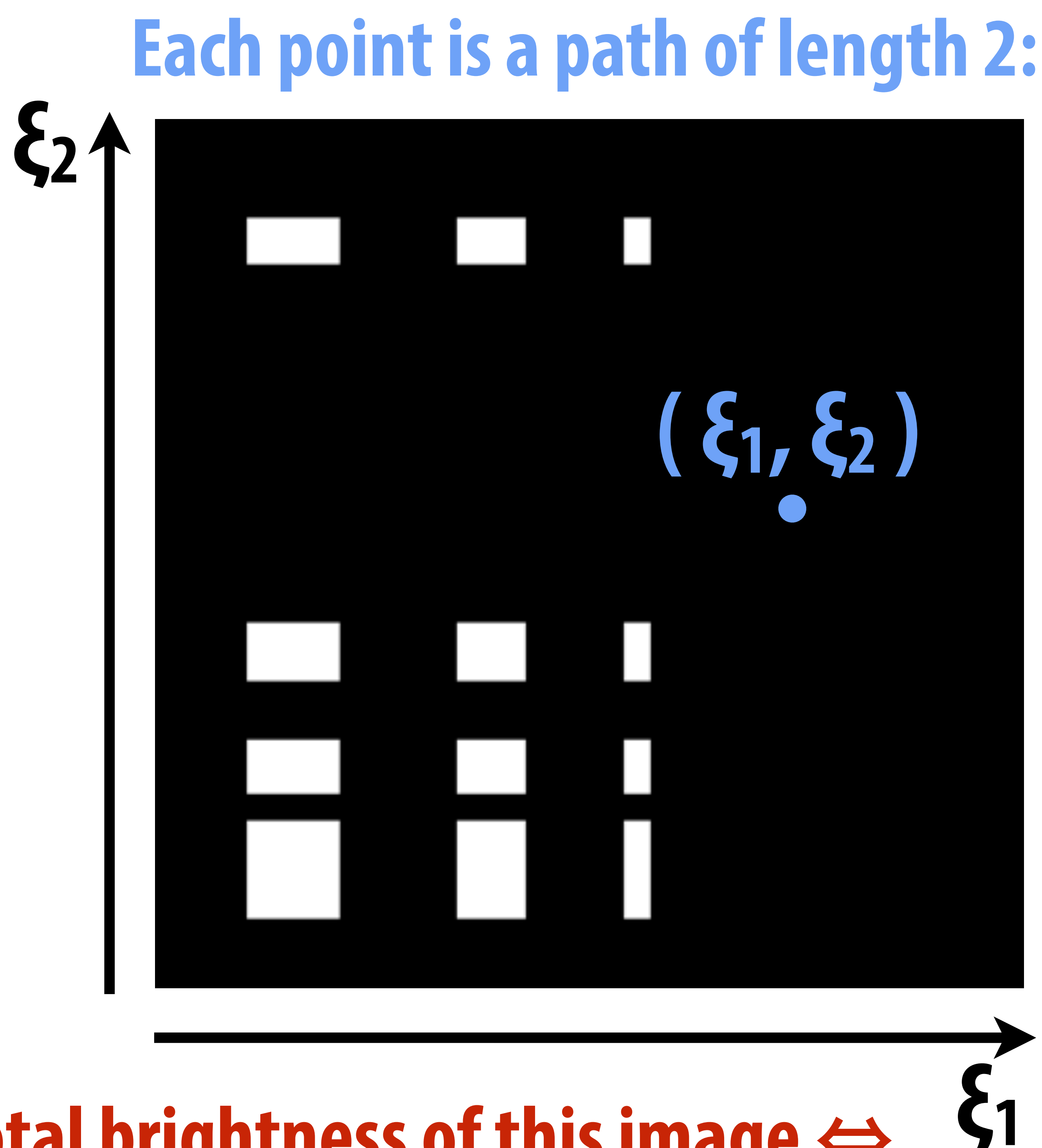
all possible paths

- **Opens the door to intelligent "global" importance sampling. (But still hard!)**

Unit Hypercube View of Path Space

- Paths determined by a sequence of random values ξ in [0,1]
- Hence, path of length k is a point in hypercube $[0,1]^k$
- "Just" integrate over cubes of each dimension *k*
- E.g., two bounces in a 2D scene:





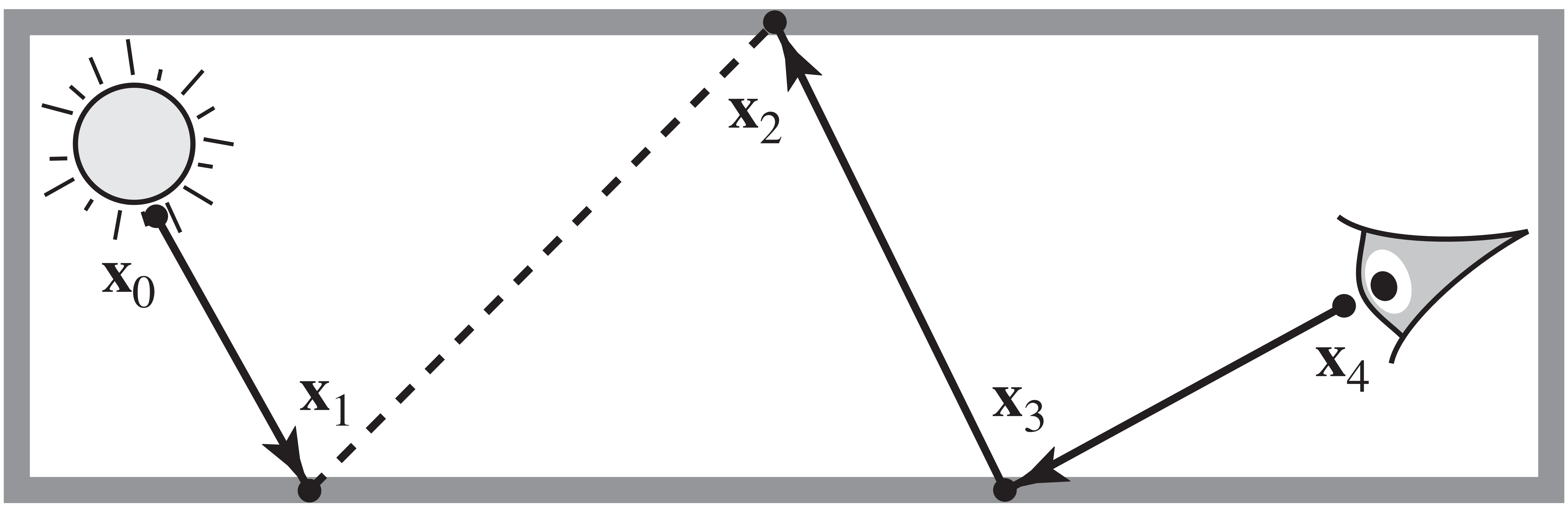
Total brightness of this image ⇔ total contribution of length-2 paths.

each bounce: $\xi \in [0,1] \mapsto \theta \in [0,\pi]$

How do we choose paths—and path *lengths*?

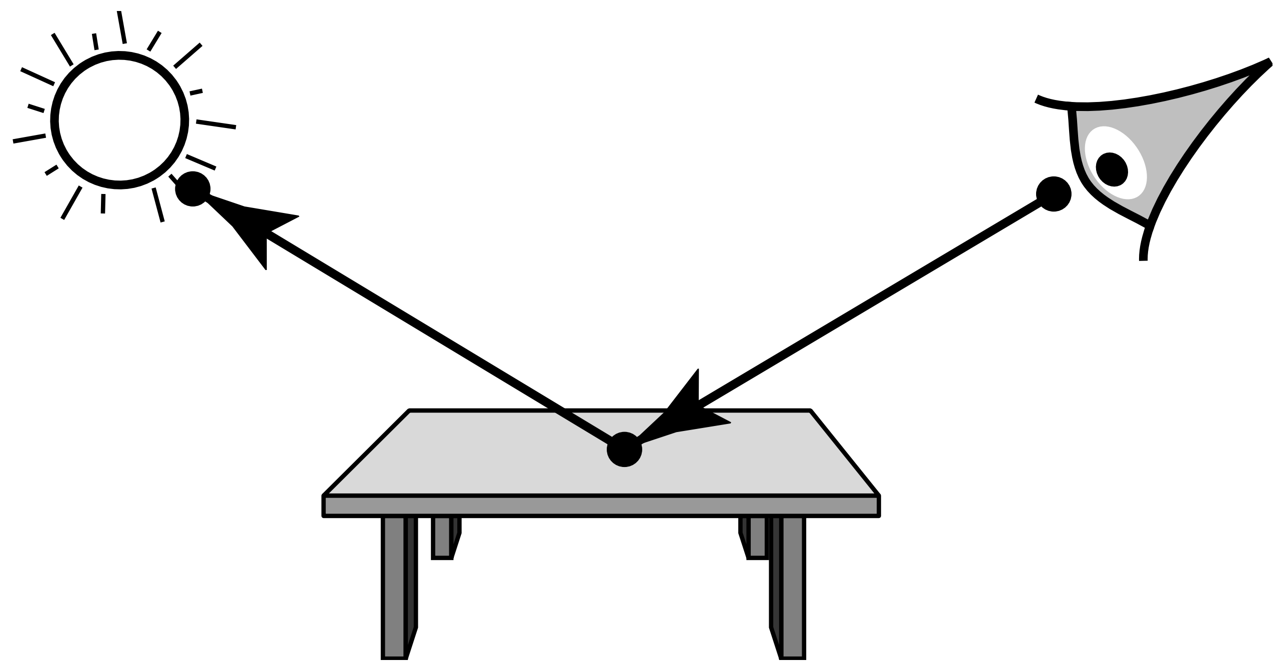
Bidirectional Path Tracing

- **Forward path tracing: no control over path length (hits light after *n* bounces, or gets terminated by Russian Roulette)**
- **Idea: connect paths from light, eye (*"bidirectional"*)**



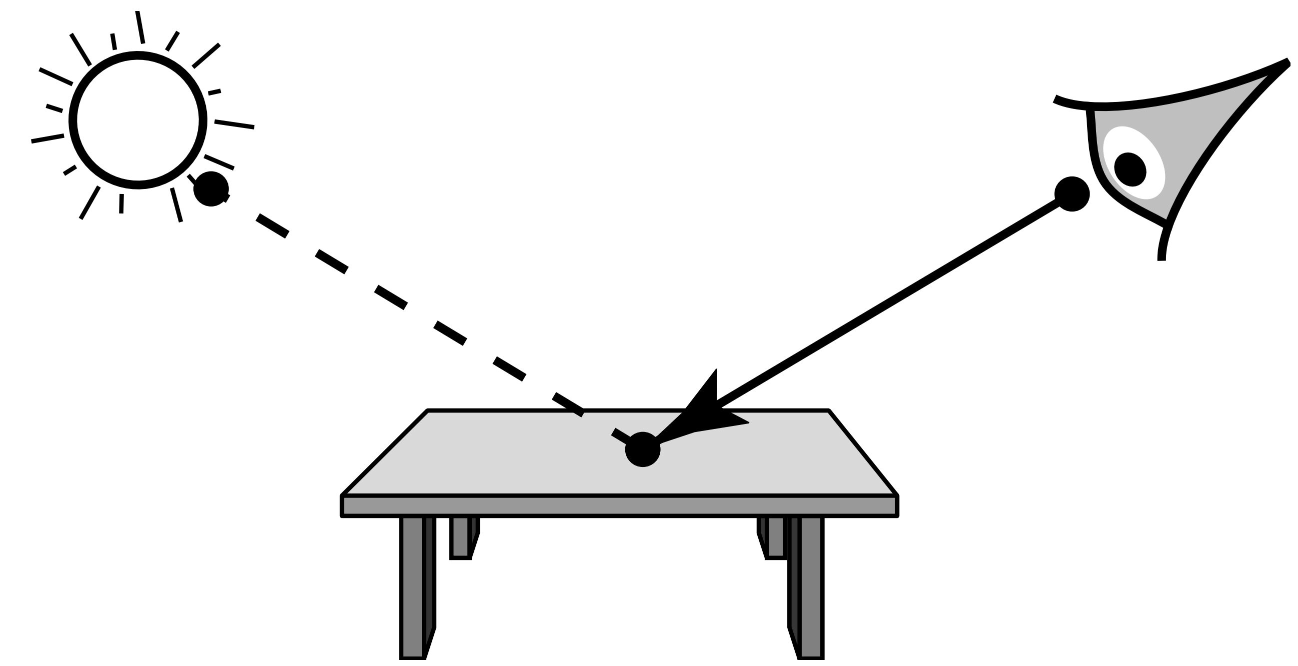
- **Importance sampling? Need to *carefully* weight contributions of path according to sampling strategy.**
- **(Details in Veach & Guibas, "Bidirectional Estimators for Light Transport")**

Bidirectional Path Tracing (Path Length=2)

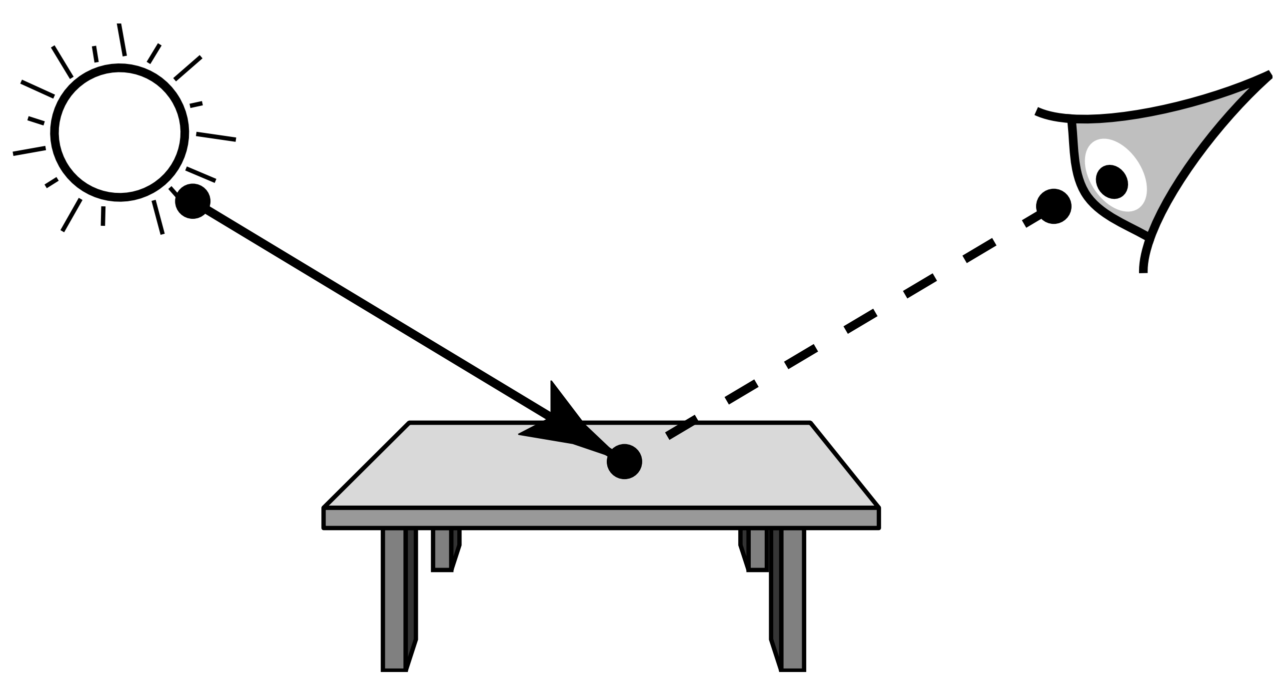


standard (forward) path tracing

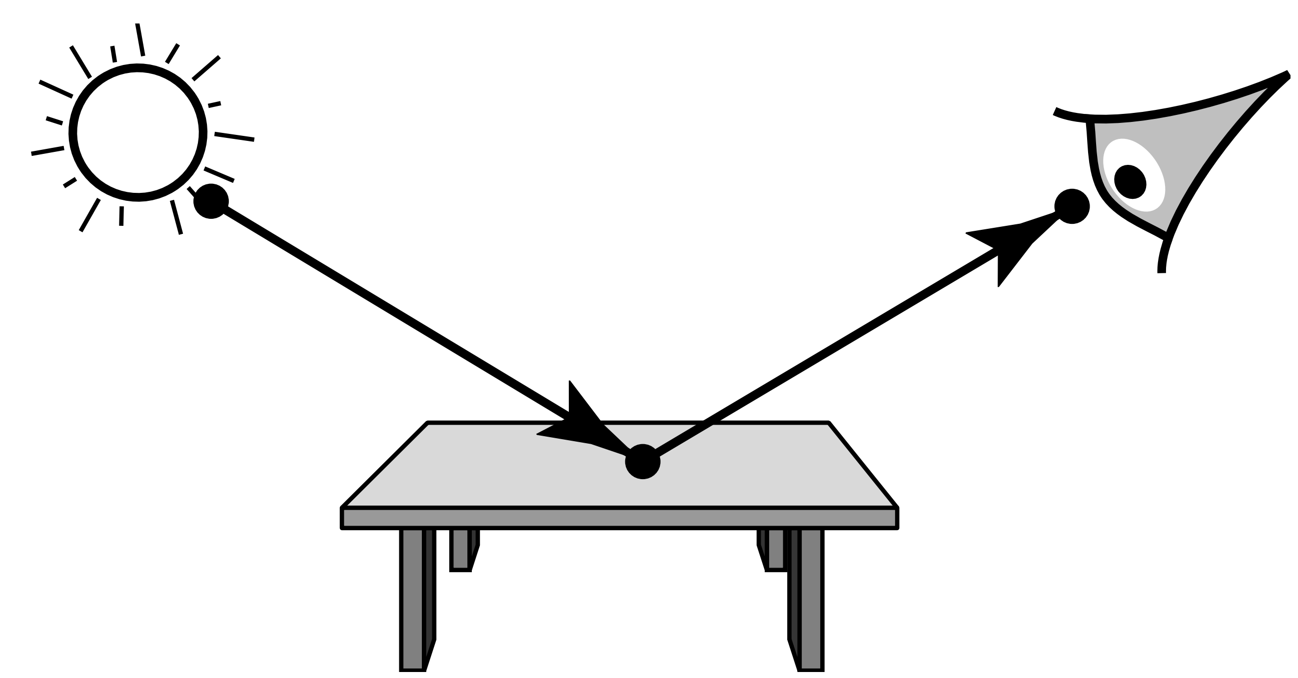
fails for point light sources



direct lighting



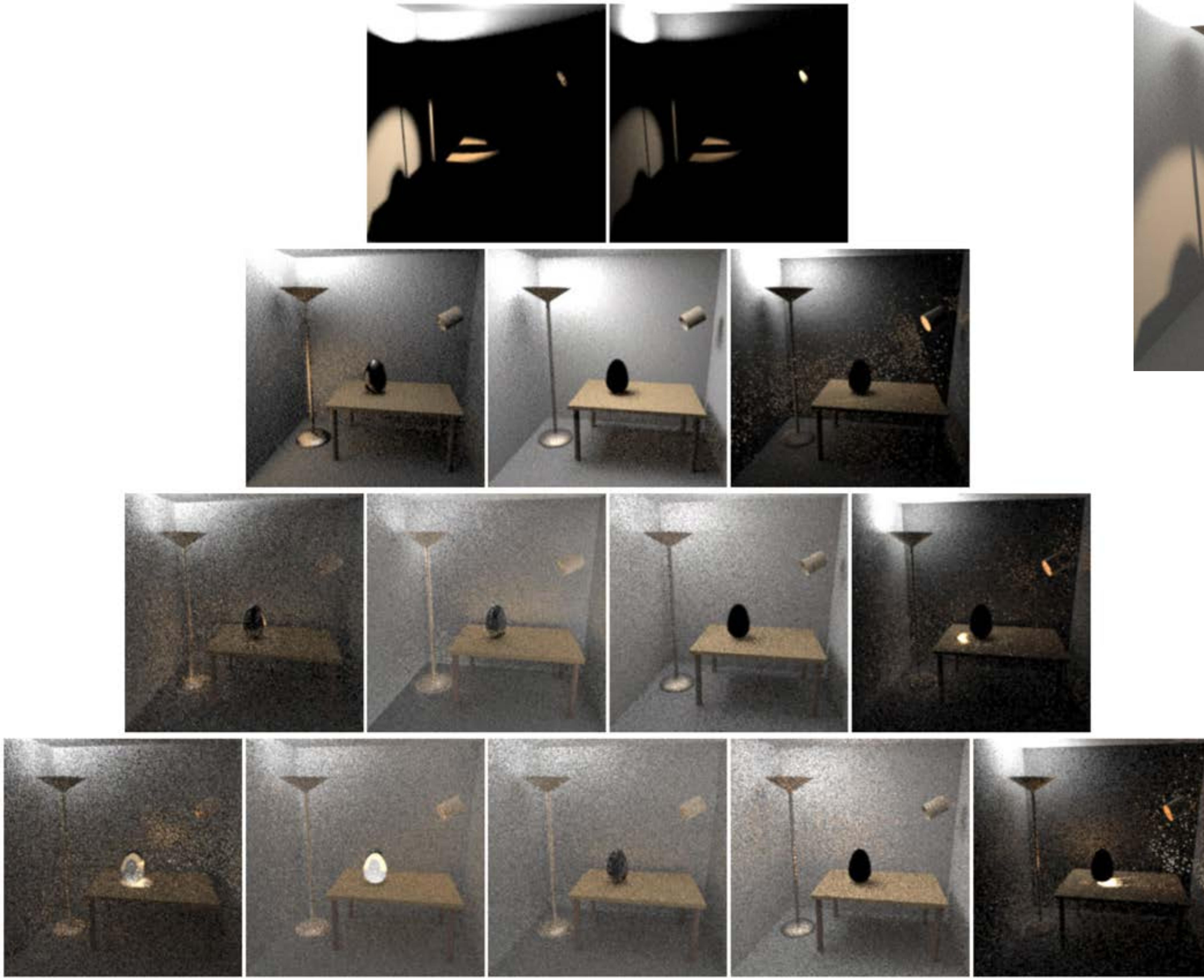
visualize particles from light



backward path tracing

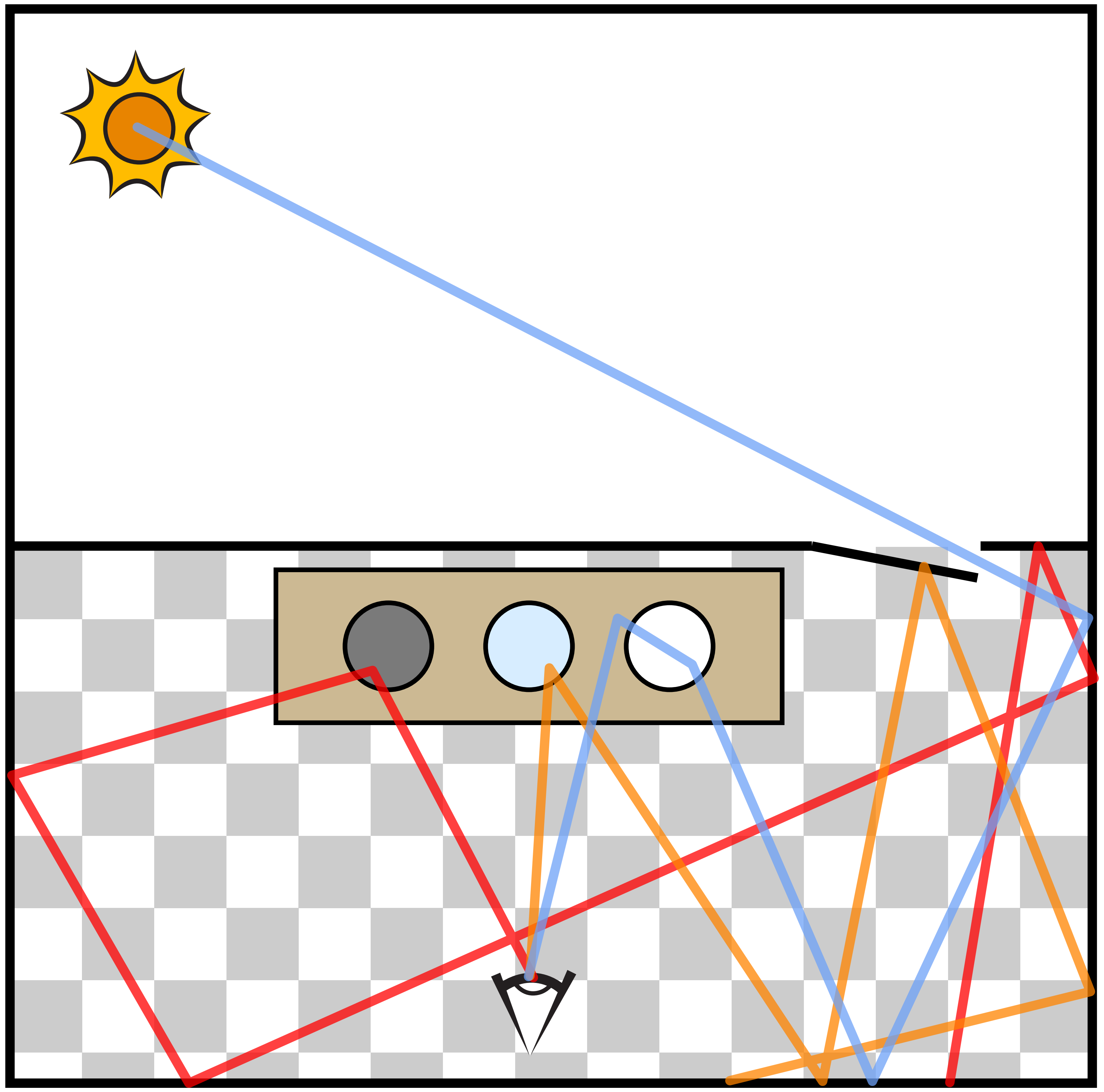
fails for a pinhole camera

Contributions of Different Path Lengths



final image

Good paths can be hard to find!



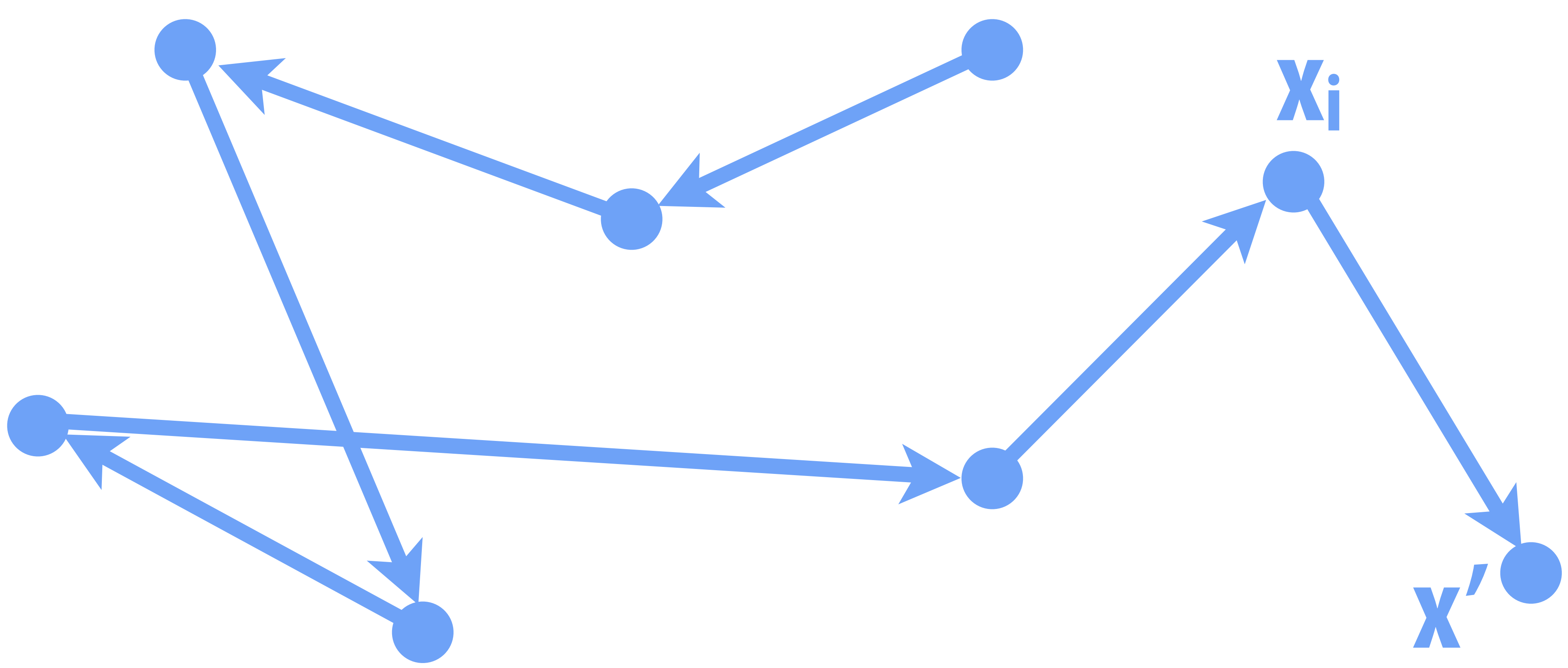
Idea:
Once we find a good path, perturb it to find nearby "good" paths.

bidirectional path tracing

Metropolis light transport (MLT)

Metropolis-Hastings Algorithm (MH)

- Standard Monte Carlo: sum up independent samples
- MH: take random walk of dependent samples ("mutations")
- Basic idea: prefer to take steps that increase sample value



$\alpha := f(x') / f(x)$ "transition probability"

if random x_i in $[0,1] < \alpha$:

$x_{i+1} = x'$

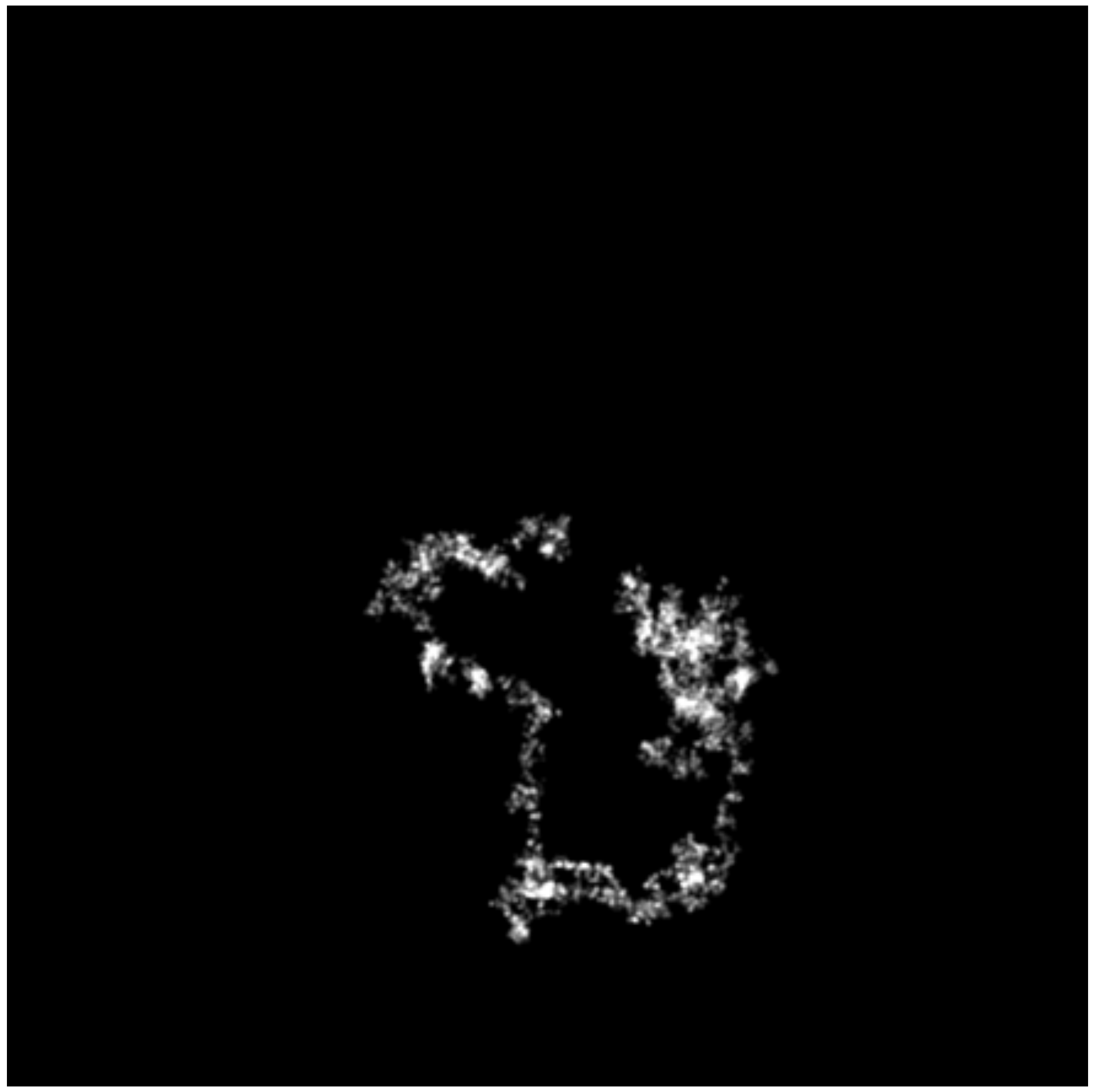
else:

$x_{i+1} = x_i$

- If careful, sample distribution will be proportional to integrand
  - make sure mutations are "ergodic" (reach whole space)
  - need to take a long walk, so initial point doesn't matter ("mixing")

Metropolis-Hastings: Sampling an Image

- Want to take samples proportional to image density f
- Start at random point; take steps in (normal) random direction
- Occasionally jump to random point (ergodicity)
- Transition probability is "relative darkness" $f(x')/f(x_i)$

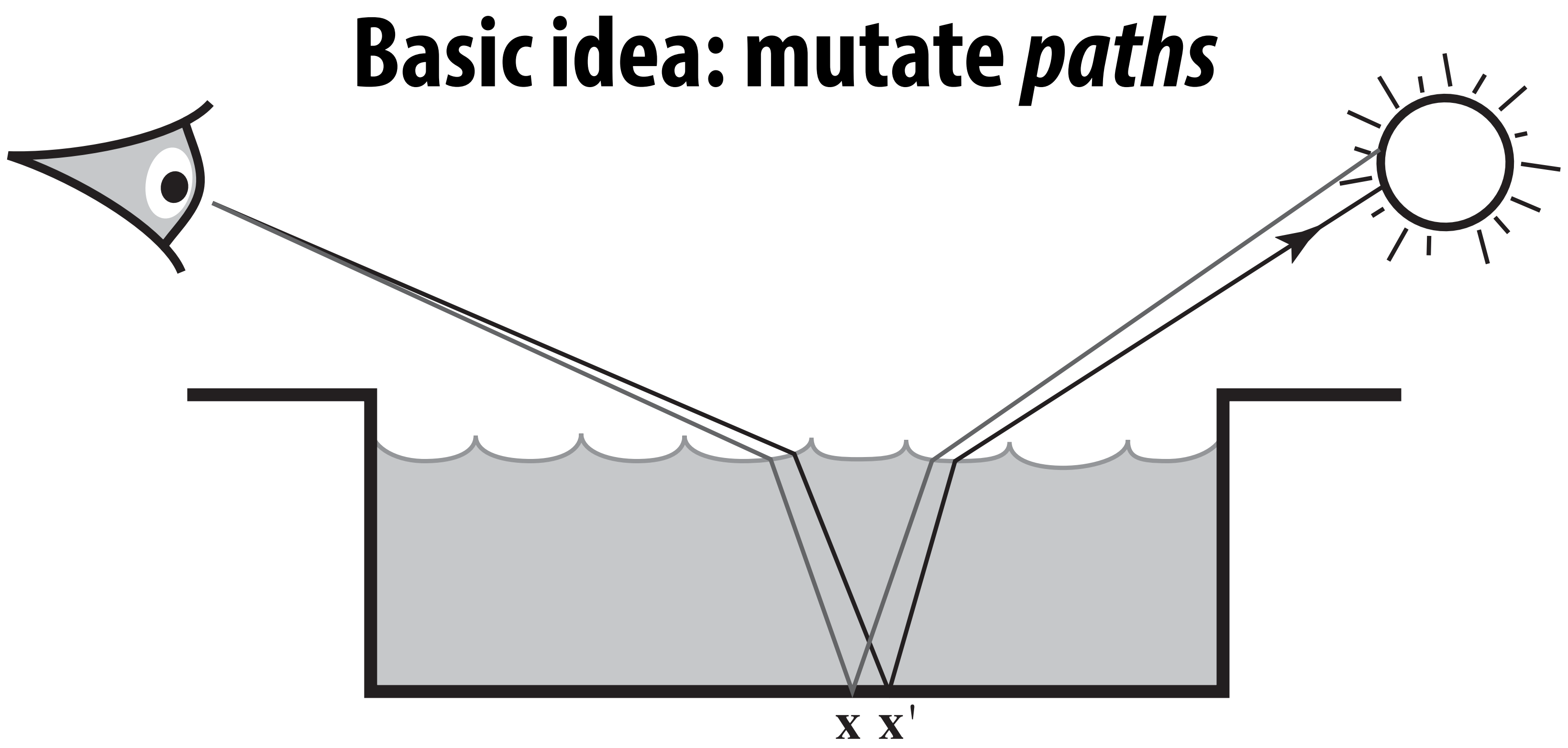


short walk

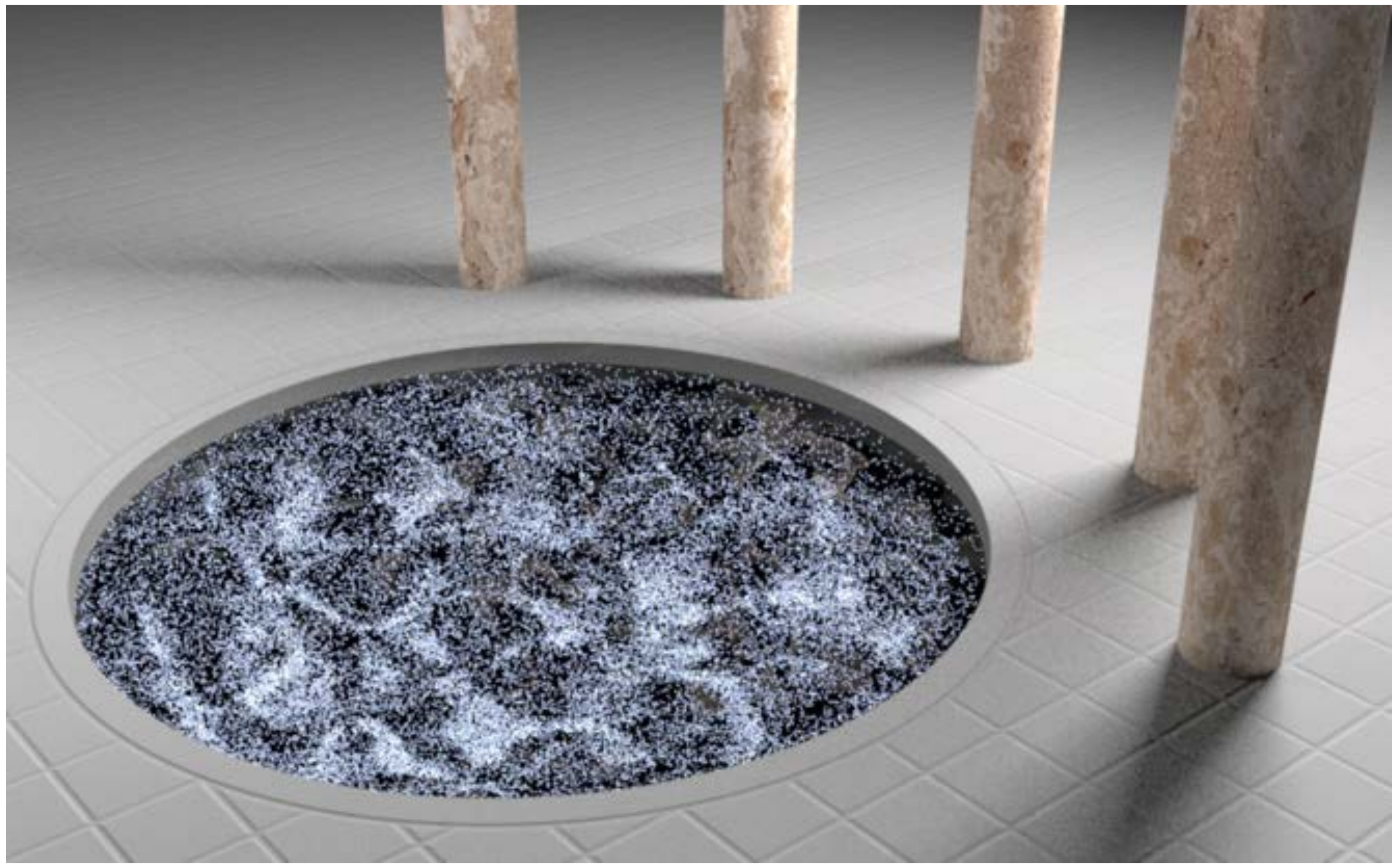
long walk

(original image)

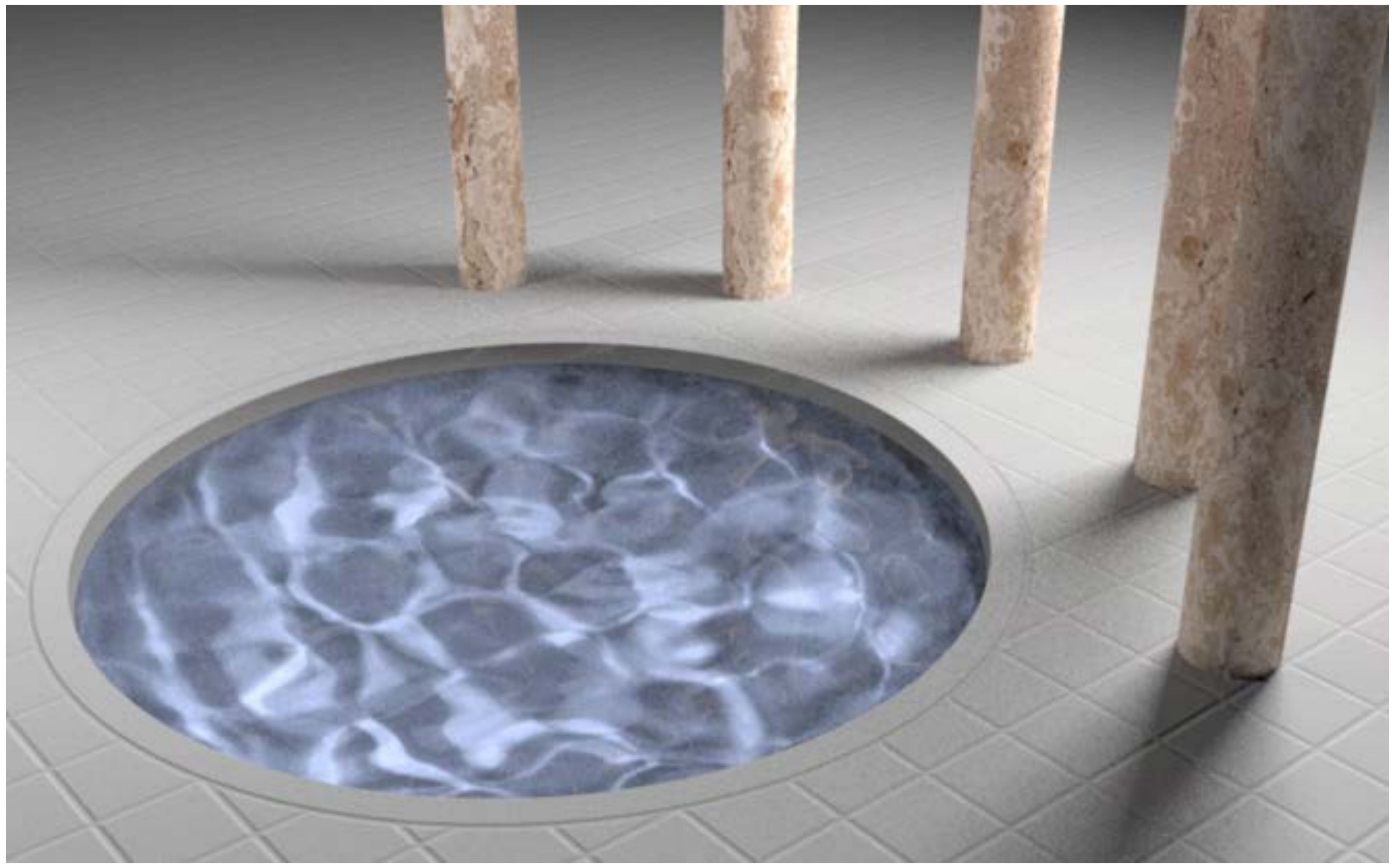
Metropolis Light Transport



(For details see Veach, "Robust Monte Carlo Methods for Light Transport Simulation")



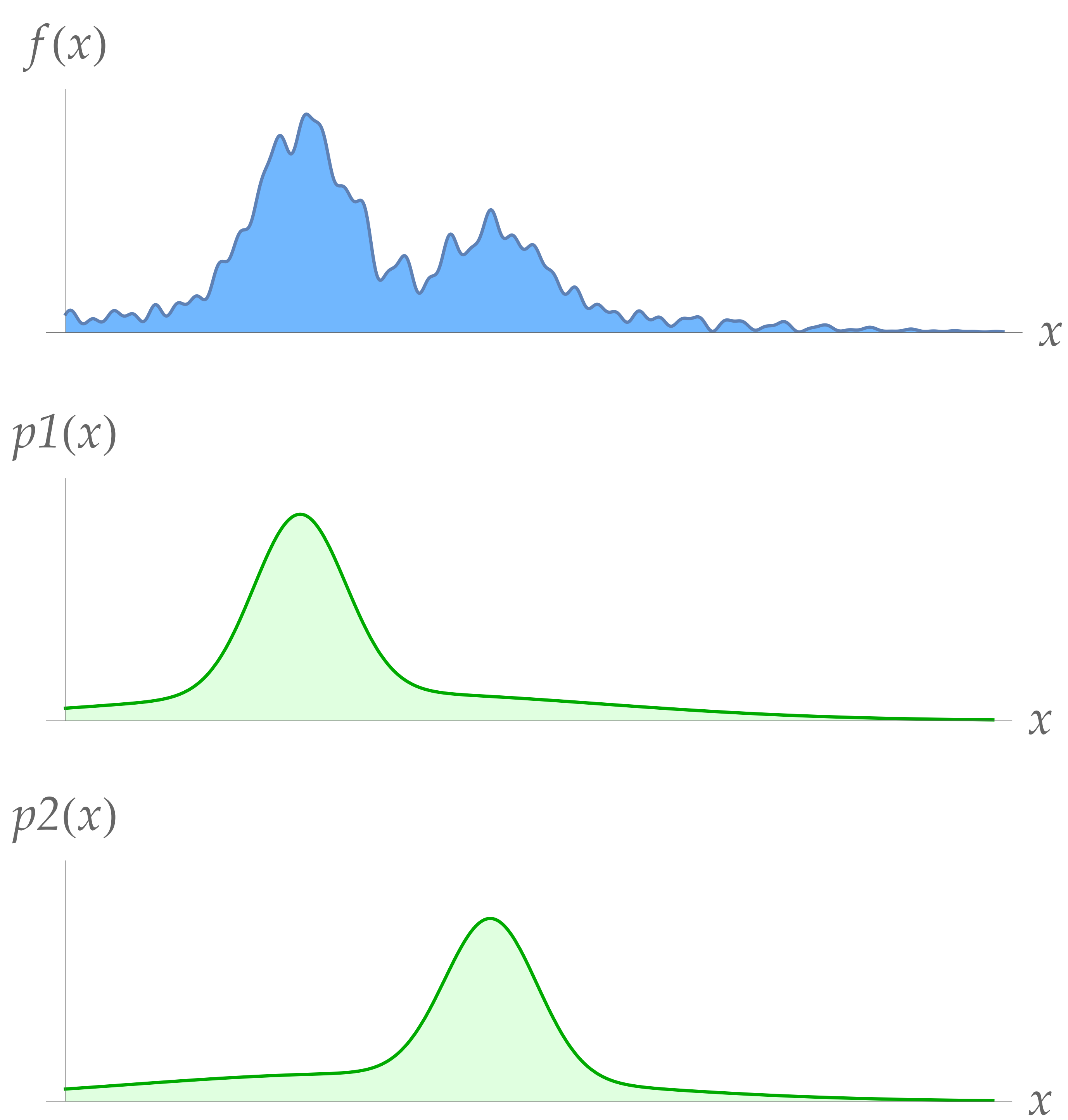
path tracing



Metropolis light transport (same time)

Multiple Importance Sampling (MIS)

- Many possible importance sampling strategies
- Which one should we use for a given integrand?
- MIS: *combine* strategies to preserve strengths of all of them
- *Balance heuristic* is (provably!) about as good as anything:



sum over strategies

sum over samples

jth sample taken with ith strategy

$$\frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\frac{f(x_{ij})}{\sum_k c_k p_k(x_{ij})}$$

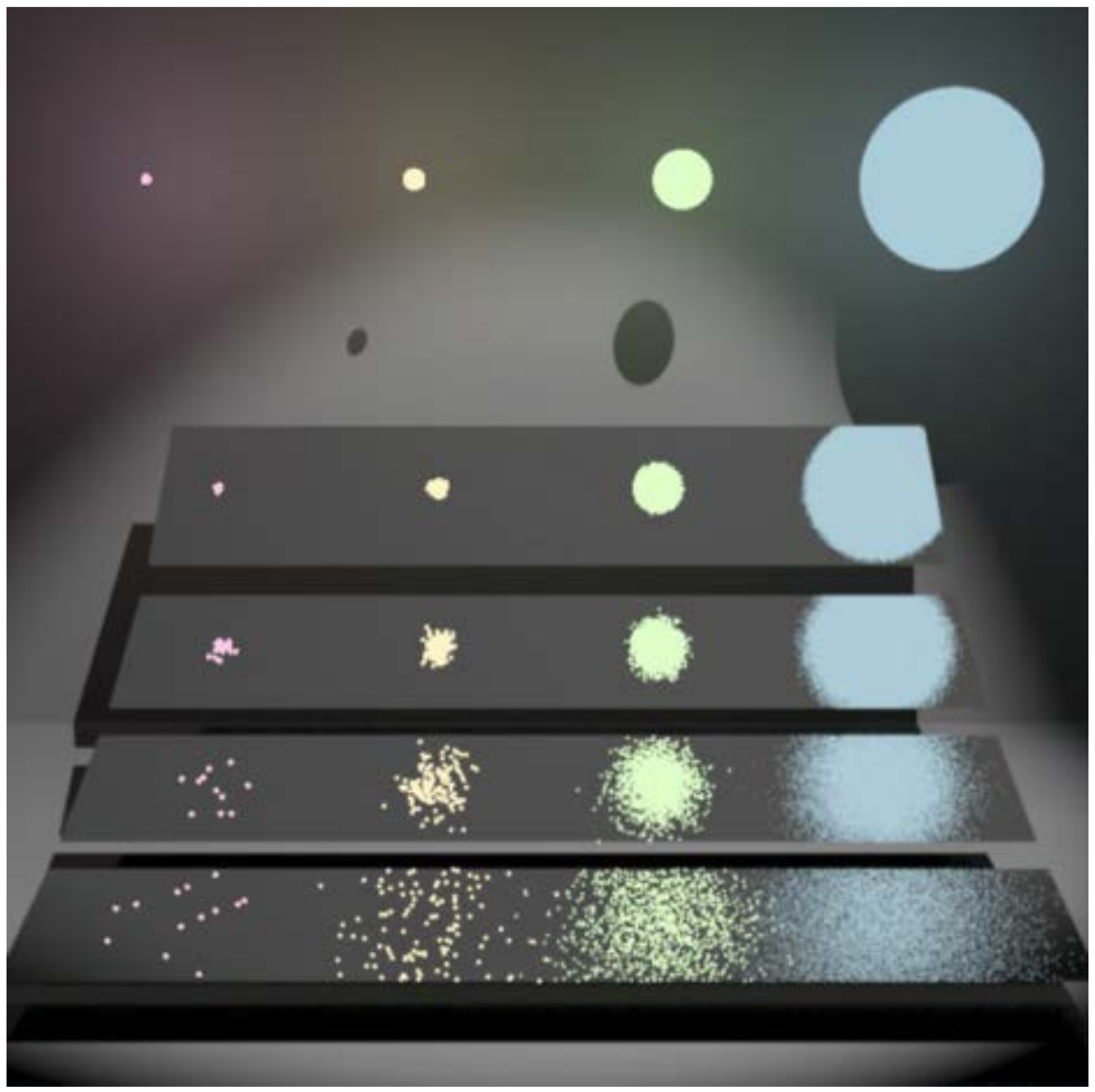
total # of samples

fraction of samples taken w/ kth strategy

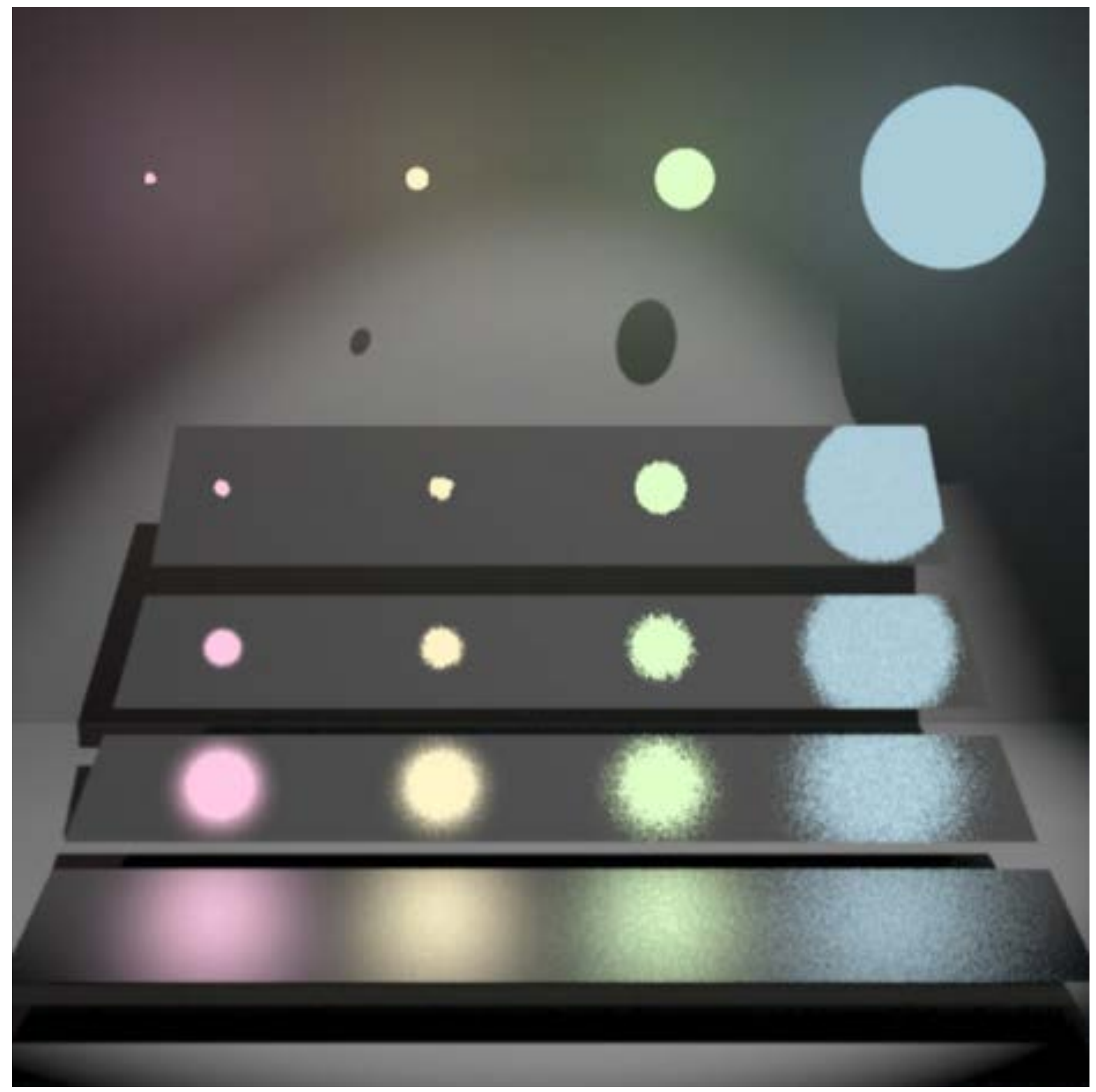
*k*th importance density

Still, several improvements possible (cutoff, power, max)—see Veach & Guibas.

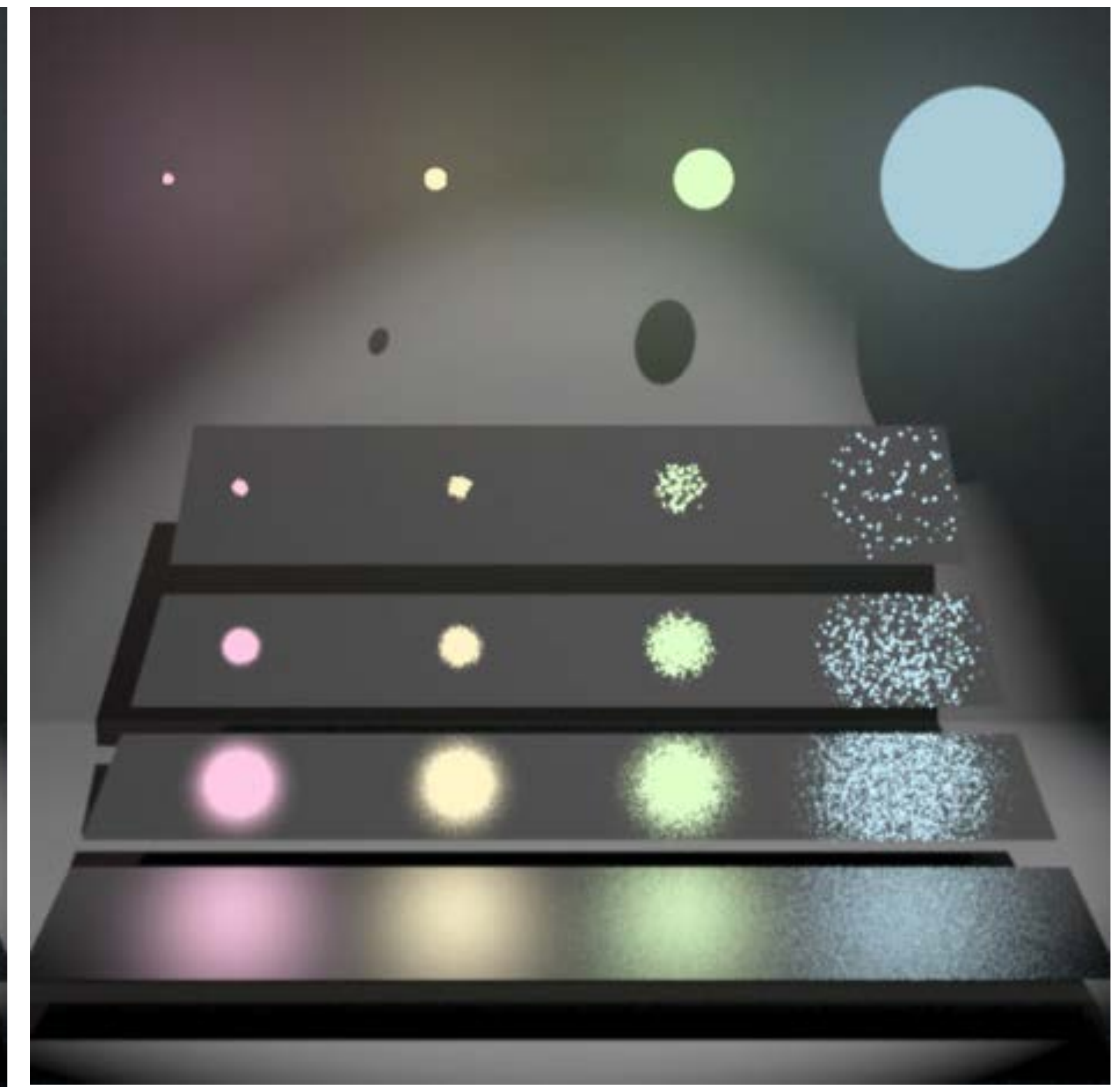
Multiple Importance Sampling: Example



sample materials



multiple importance sampling
(power heuristic)



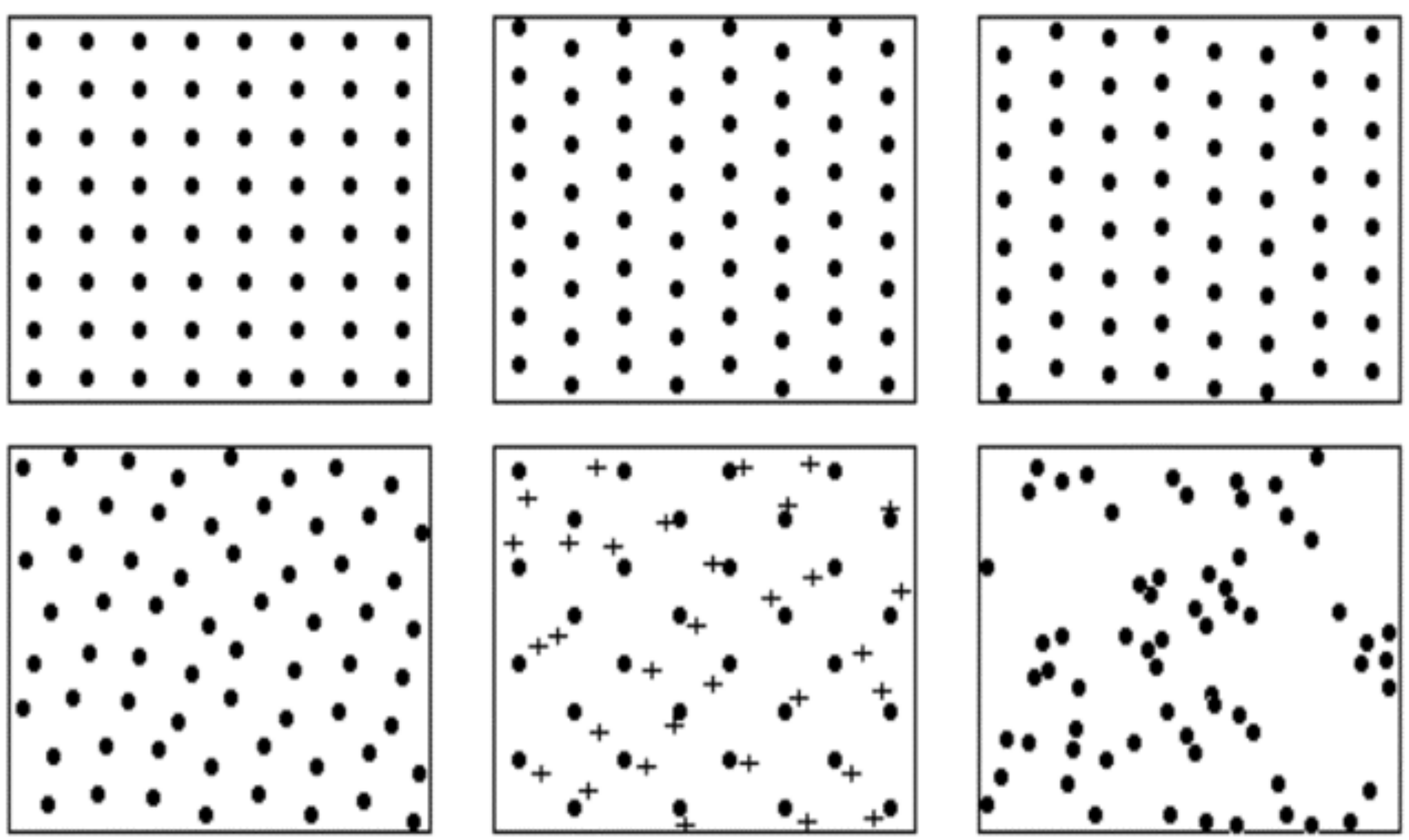
sample lights

Ok, so importance is important.

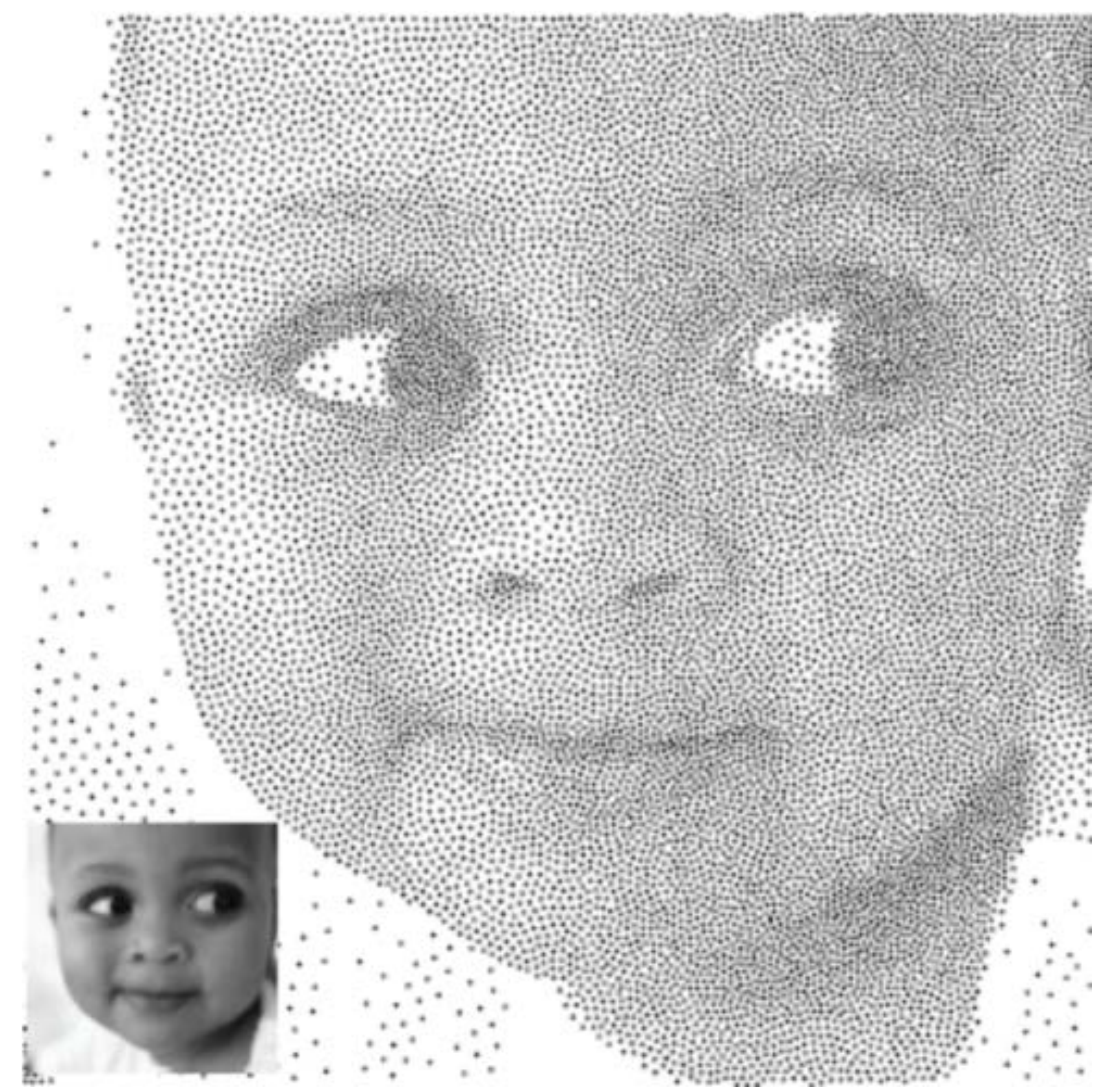
But how do we sample our function in the first place?

Sampling Patterns & Variance Reduction

- Want to pick samples according to a given density
- But even for uniform density, lots of possible sampling patterns



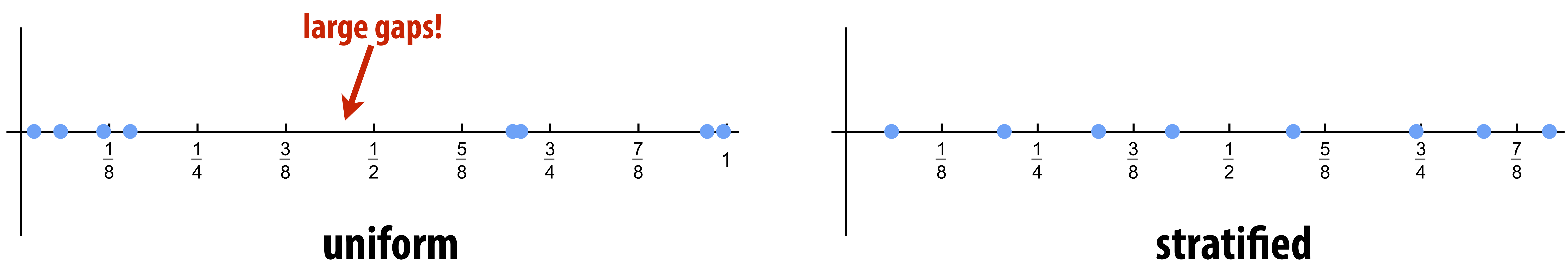
uniform sampling density



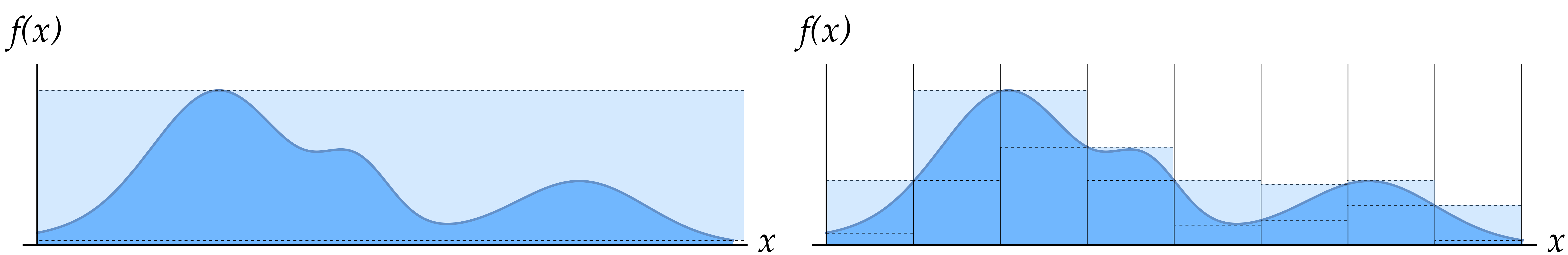
nonuniform sampling density

Stratified Sampling

- How do we pick *n* values from [0,1]?
- Could just pick *n* samples uniformly at random
- Alternatively: split into *n* bins, pick uniformly in each bin



- FACT: stratified estimate never has larger variance (often lower)

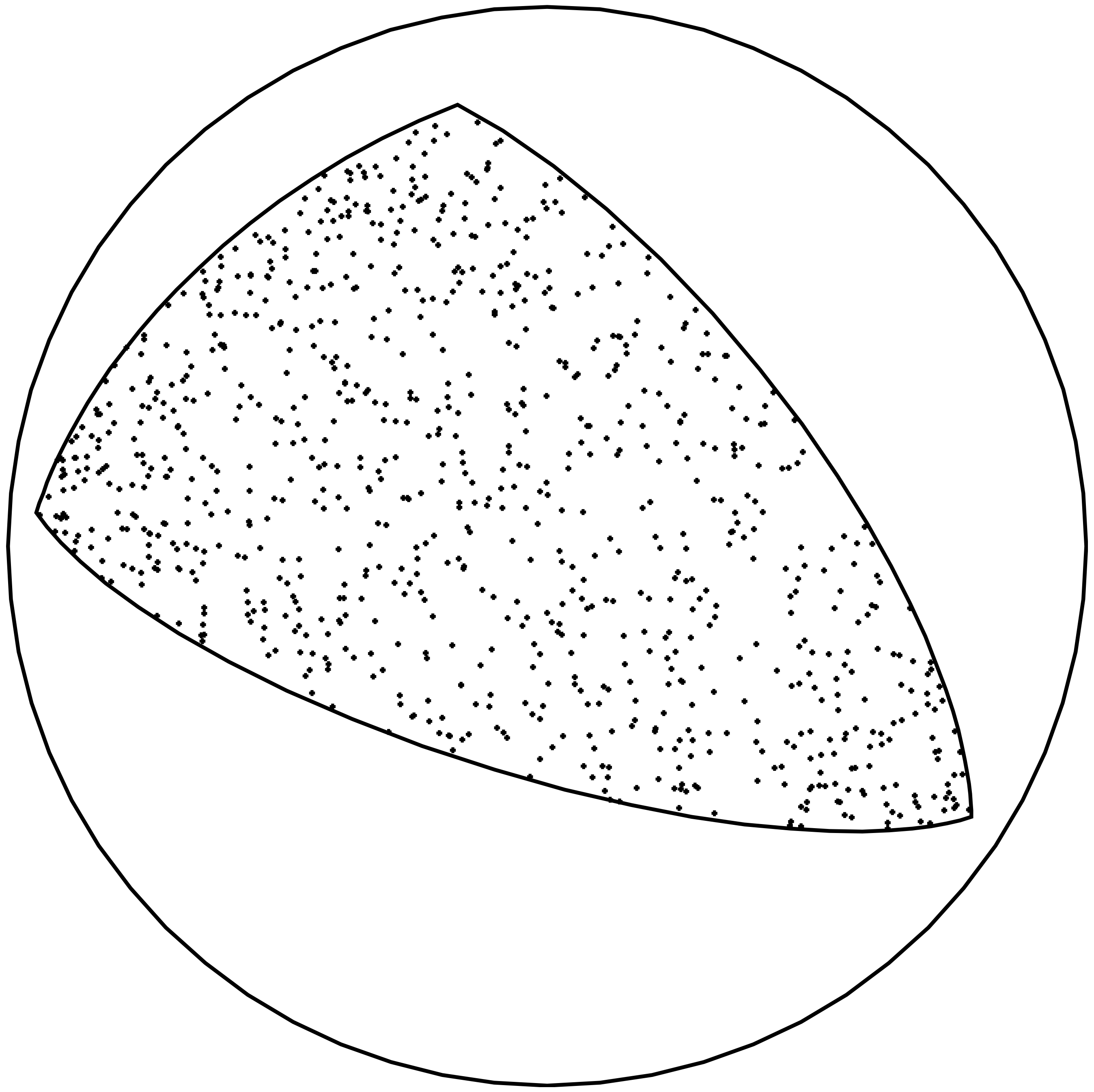


Intuition: each stratum has smaller variance. (Proof by linearity of expectation!)

Stratified Sampling in Rendering/Graphics

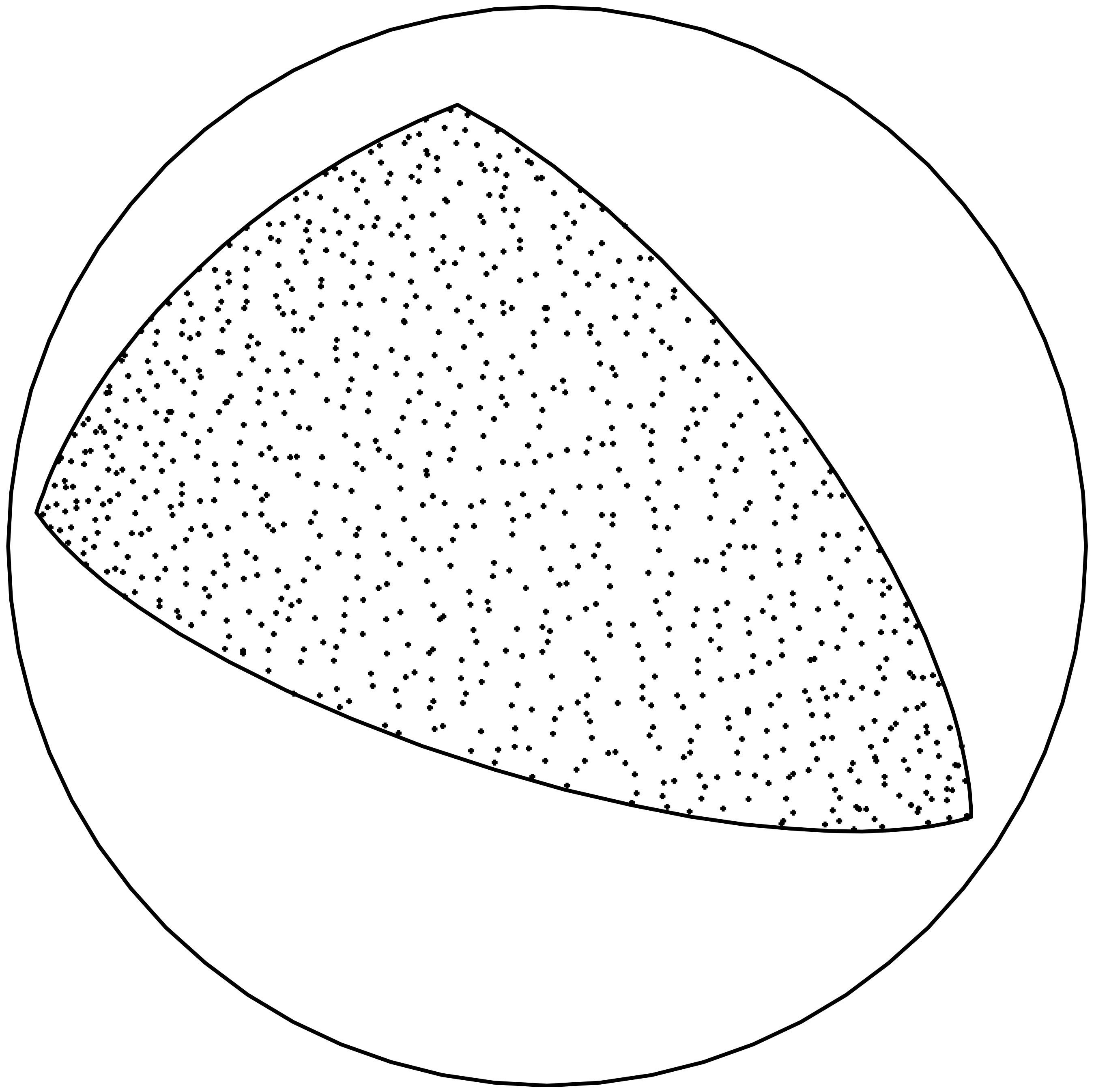
- Simply replacing uniform samples with stratified ones already improves quality of sampling for rendering (…and other graphics/visualization tasks!)

uniform



"more clumpy"

stratified



"more even"

See especially: Jim Arvo, "Stratified Sampling of Spherical Triangles" (SIGGRAPH 1995)

Low-Discrepancy Sampling

- "No clumps" hints at one possible criterion for a good sample:
- *Number of samples should be proportional to area*
- *Discrepancy* measures deviation from this ideal

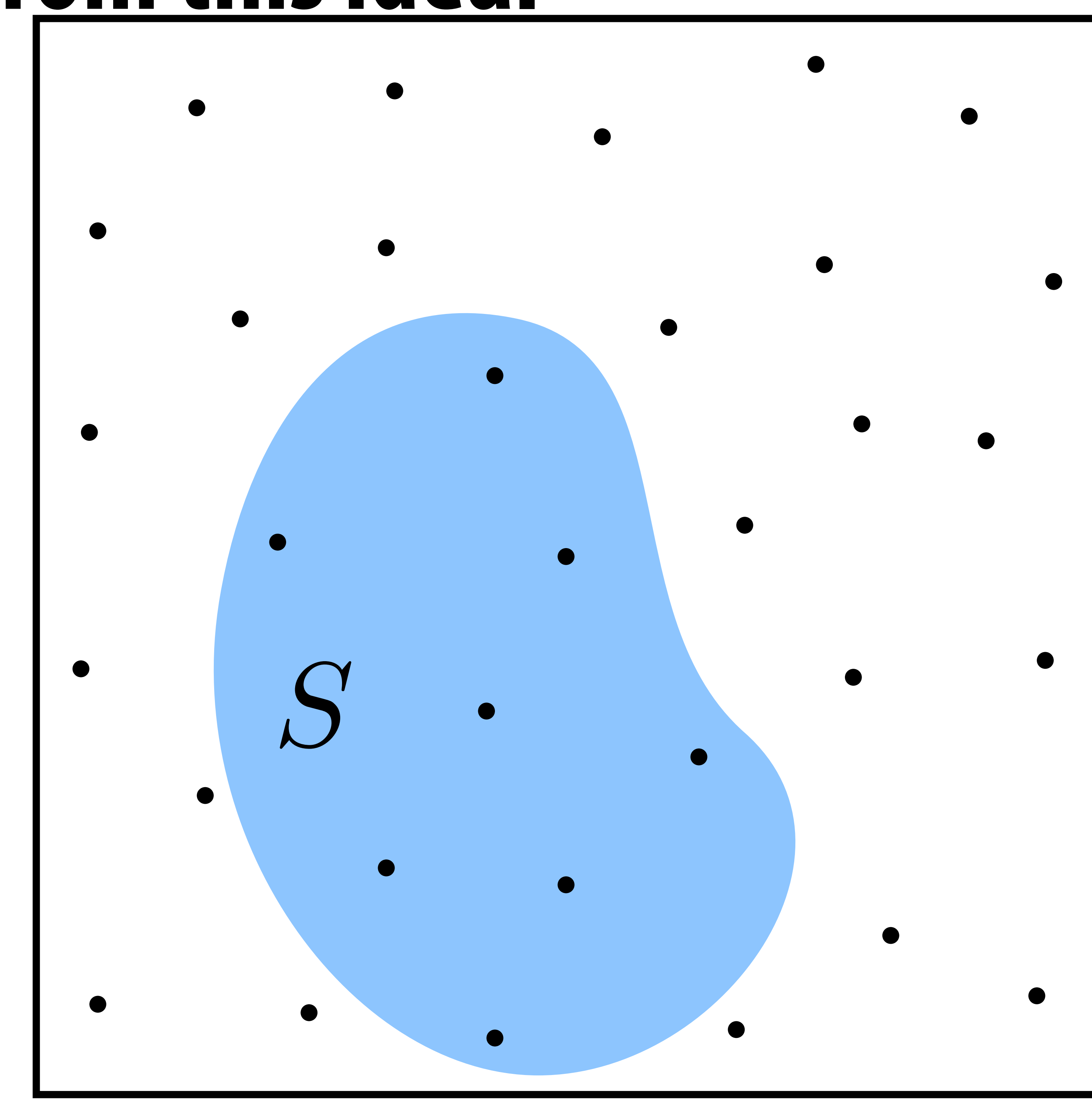
discrepancy of sample points X over a region S

number of samples in X covered by S

$$d_S(X) := \left| A(S) - \frac{n(S)}{|X|} \right|$$

area of S

total # of samples in X



overall discrepancy of X

$$D(X) := \max_{S \in F} d_S(X)$$

(*ideally* equal to zero!)

some family of regions S (e.g., boxes, disks, ...)

See especially: Dobkin et al, "Computing Discrepancy w/ Applications to Supersampling" (1996)

Quasi-Monte Carlo methods (QMC)

- Replace truly random samples with low-discrepancy samples
- Why? *Koksma's theorem:*

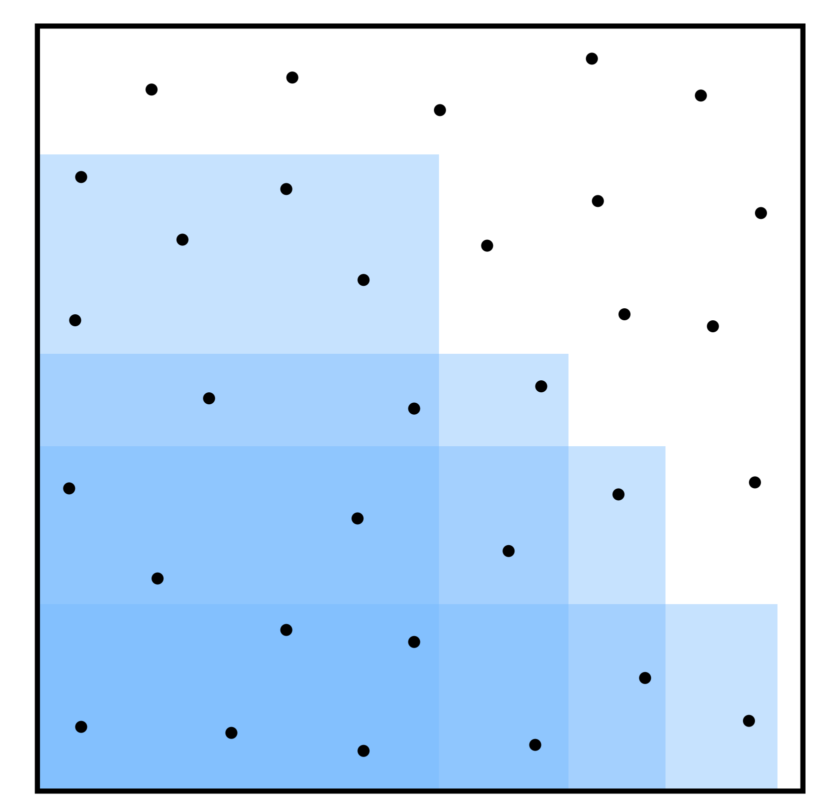
total variation of f (integral of |f′|)

discrepancy of sample X

sample points in X

$$\left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \int_0^1 f(x)\ dx\right| \leq \mathcal{V}(f)D(X)$$

- I.e., for low-discrepancy X, estimate approaches integral
- Similar bounds can be shown in higher dimensions
- **WARNING:** total variation not always bounded!
- **WARNING:** only for family F of *axis-aligned* boxes S!
- E.g., edges can have arbitrary orientation (coverage)
- Discrepancy still a very reasonable criterion in practice



F

Hammersley & Halton Points

- Can easy generate samples with *near-optimal* discrepancy
- First define *radical inverse* $\varphi_r(i)$
- Express integer *i* in base *r*, then reflect digits around decimal
- E.g., $\varphi_{10}(1234) = 0.4321$
- Can get *n Halton points* $x_1, \ldots, x_n$ in k-dimensions via

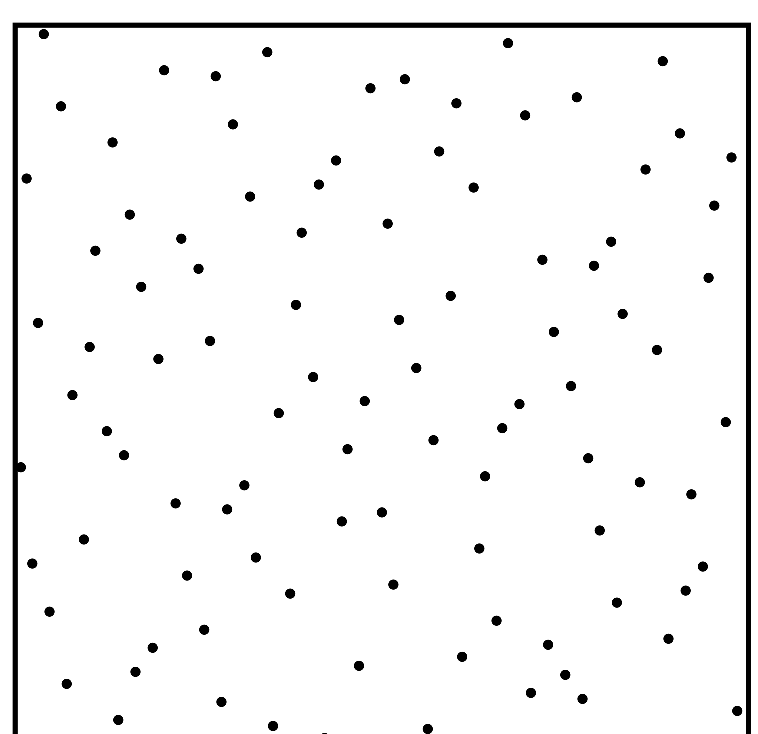
$$x_i = (\phi_{P_1}(i), \phi_{P_2}(i), \ldots, \phi_{P_k}(i))$$

kth prime number

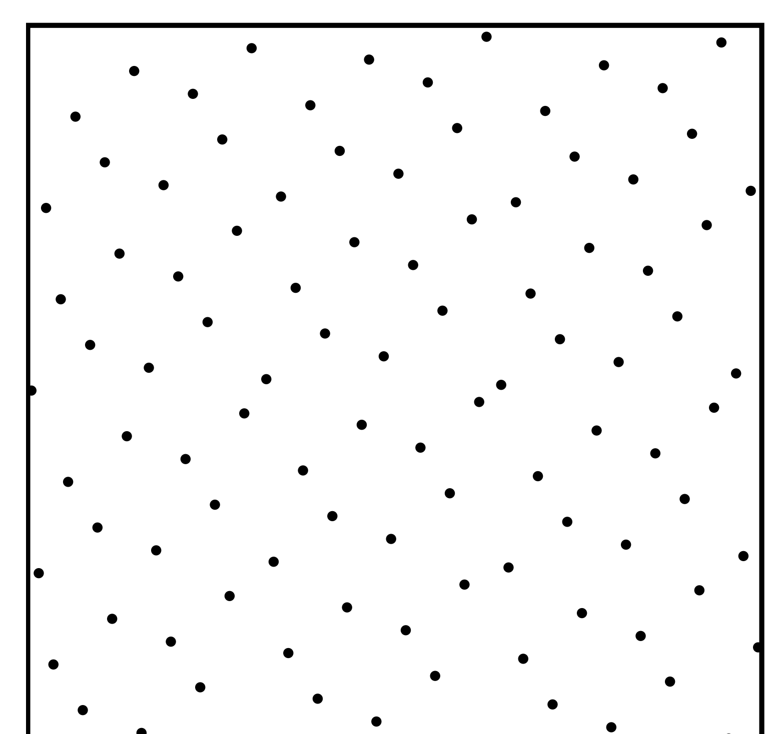
- Similarly, *Hammersley sequence* is

$$x_i = (i/n, \phi_{P_1}(i), \phi_{P_2}(i), \ldots, \phi_{P_{k-1}}(i))$$

n must be known ahead of time!

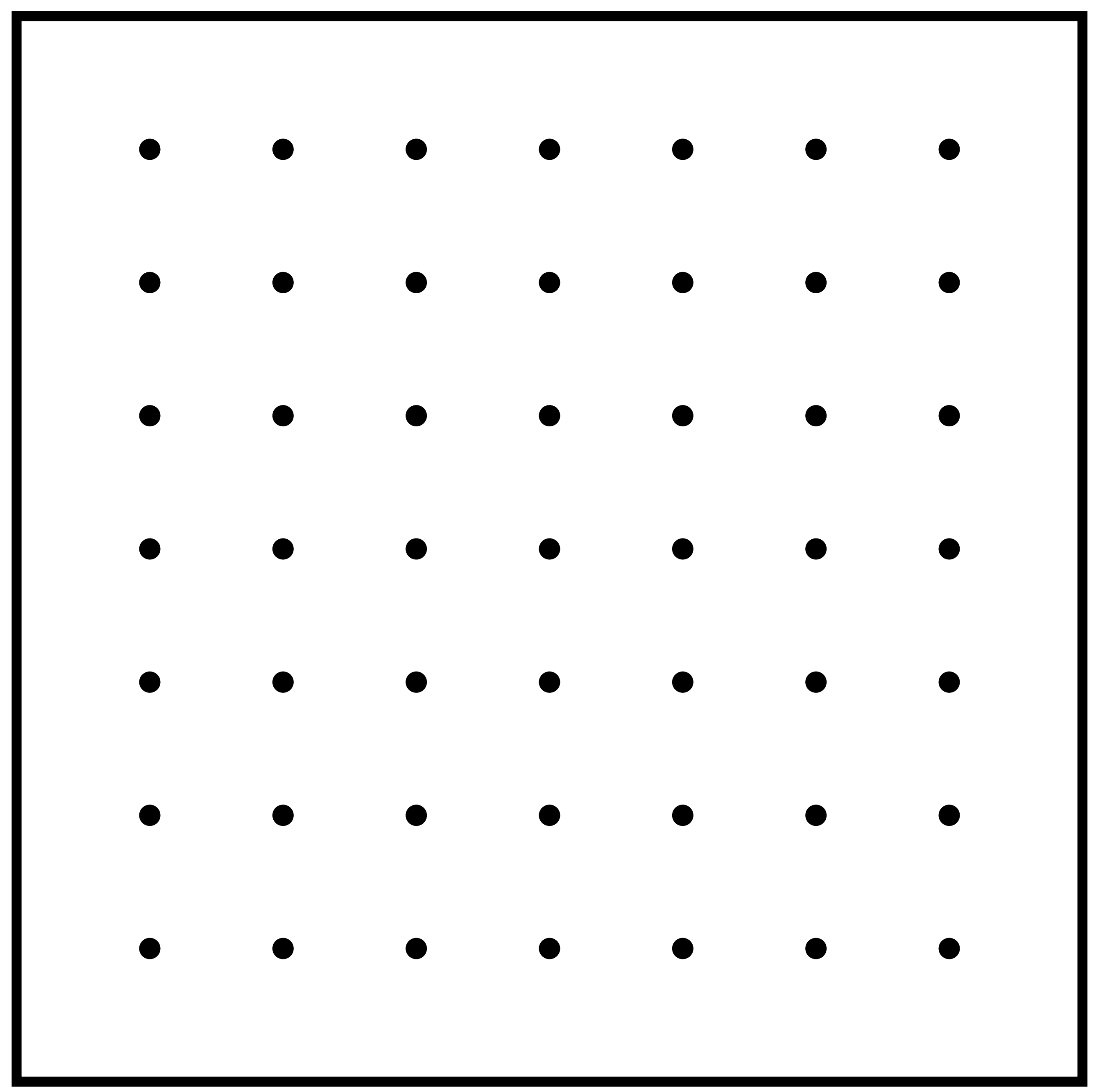


Halton



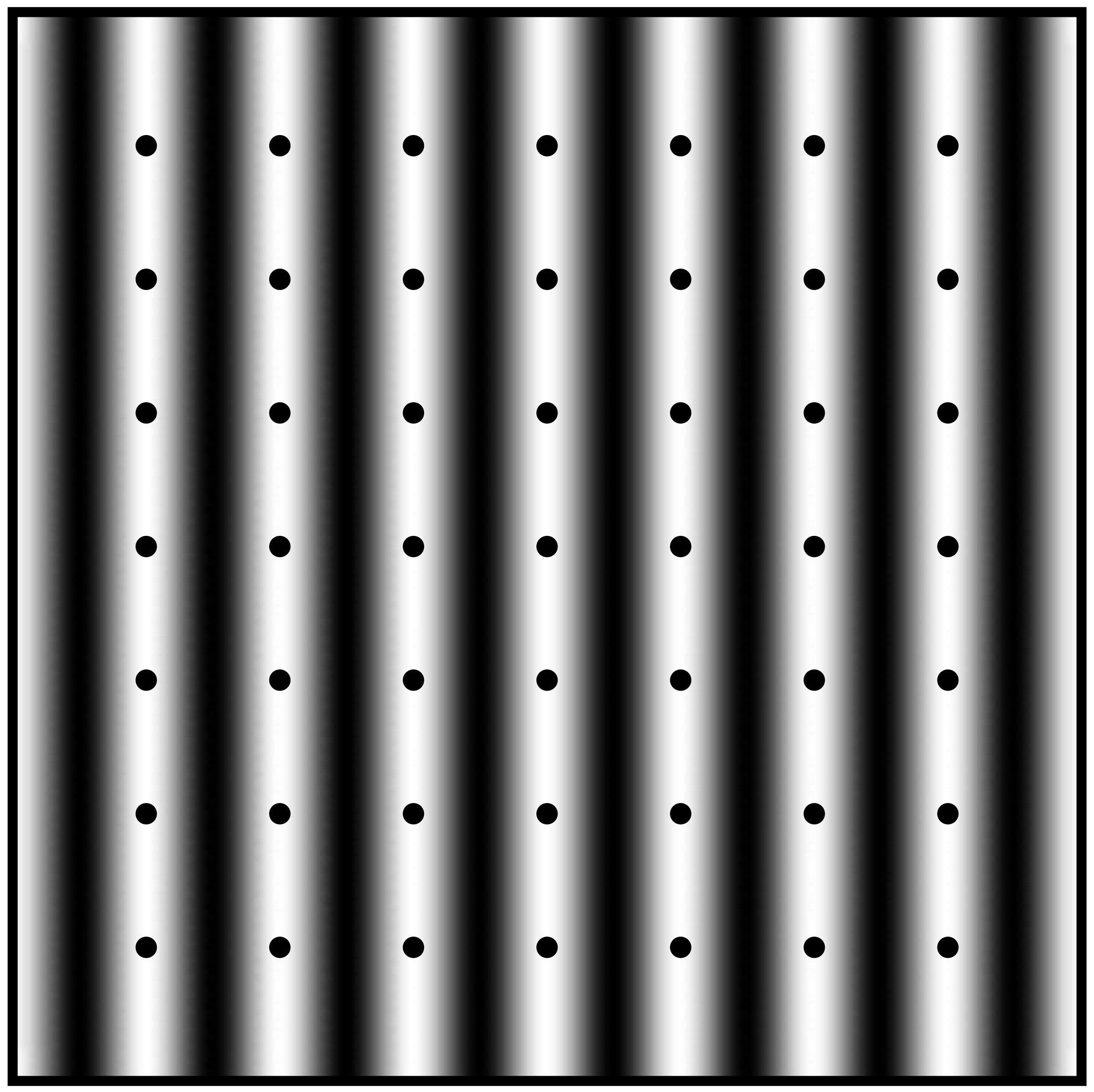
Hammersley

Wait, but doesn't a regular grid have really low discrepancy...?

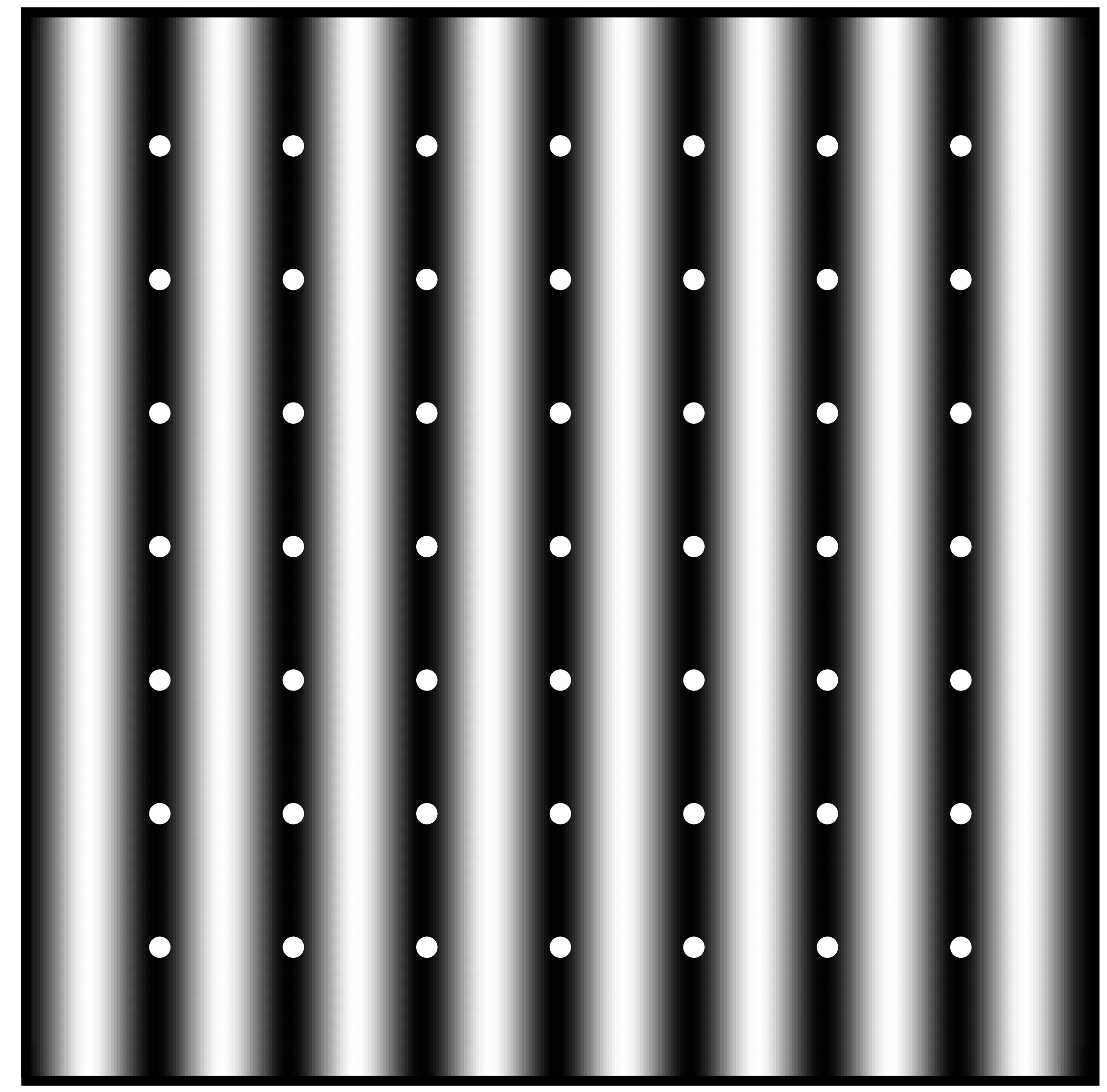


There's more to life than discrepancy

- **Even low-discrepancy patterns can exhibit poor behavior:**



$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) = 1$$



$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) = 0$$

- **Want pattern to be *anisotropic* (no preferred direction)**
- **Also want to avoid any preferred *frequency* (see above!)**

Blue Noise - Motivation

- Yellott observed that monkey retina exhibits *blue noise*

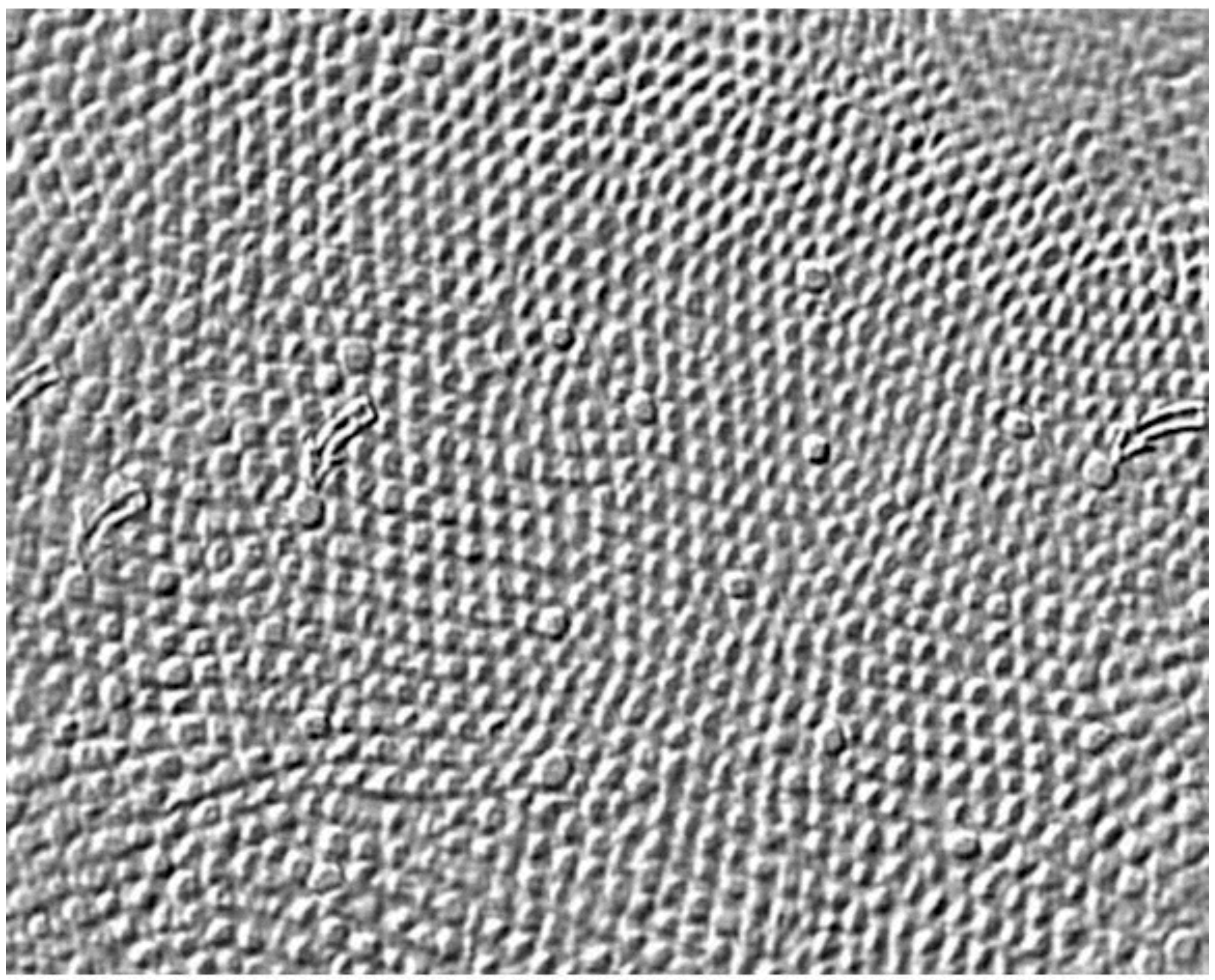
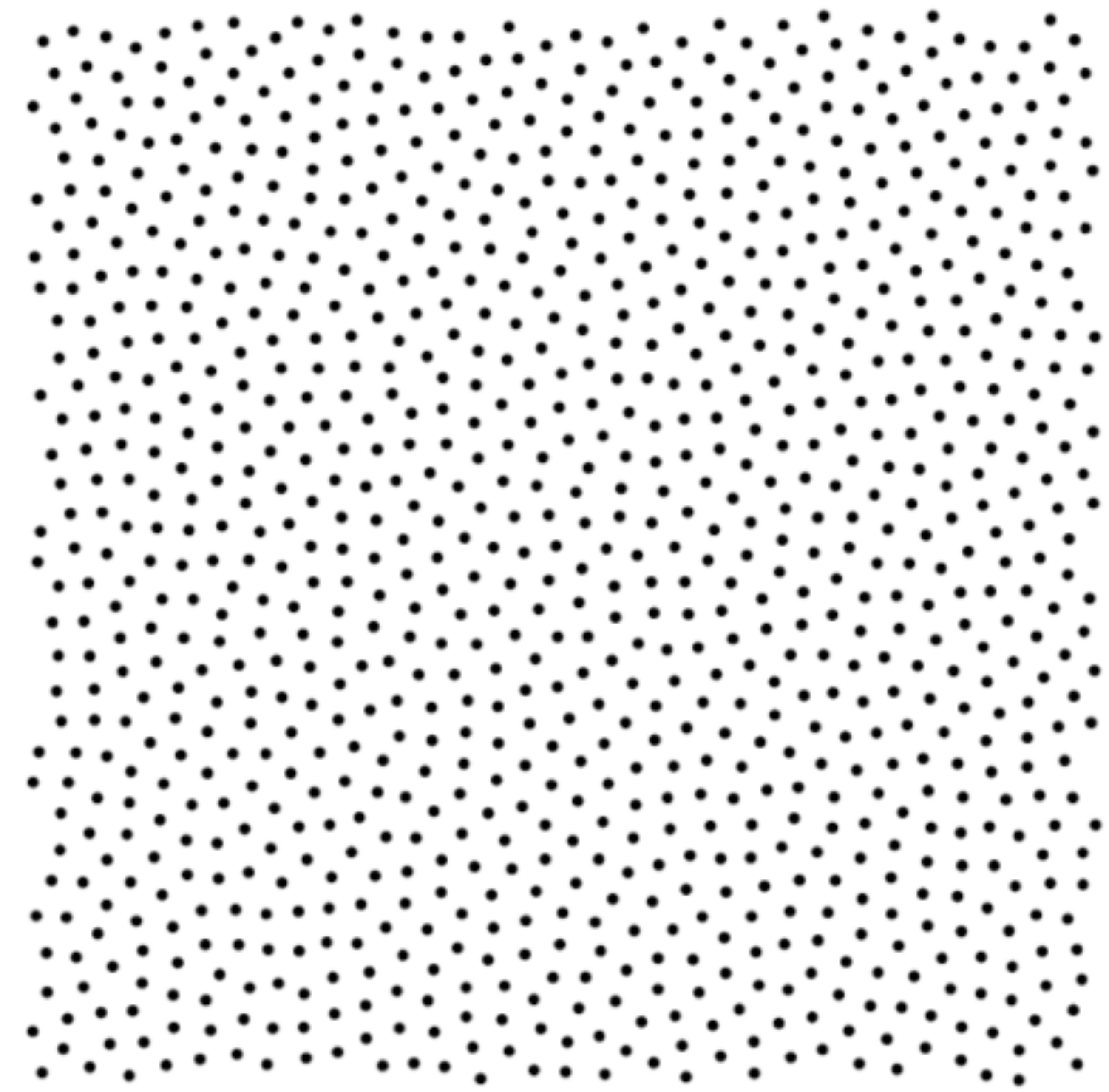


Fig. 13. Tangential section through the human fovea. Larger cones (arrows) are blue cones. From Ahnelt et al. 1987.

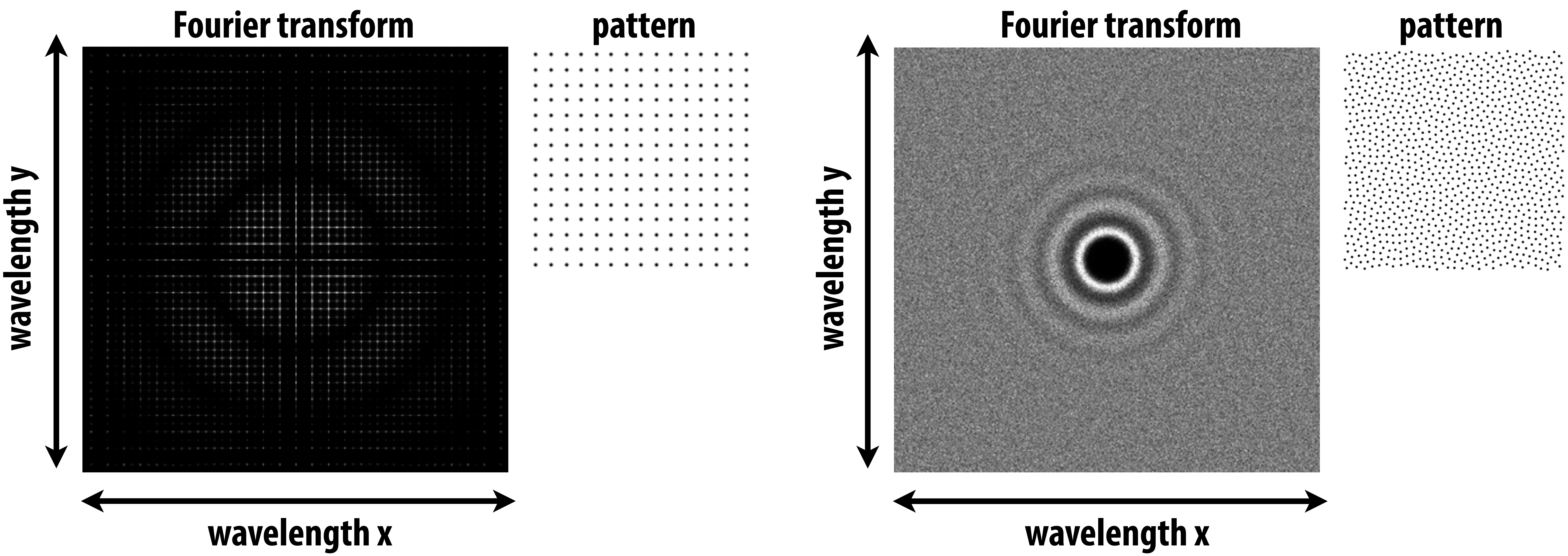


"blue noise"

- No obvious preferred directions (anisotropic)
- What about frequencies?

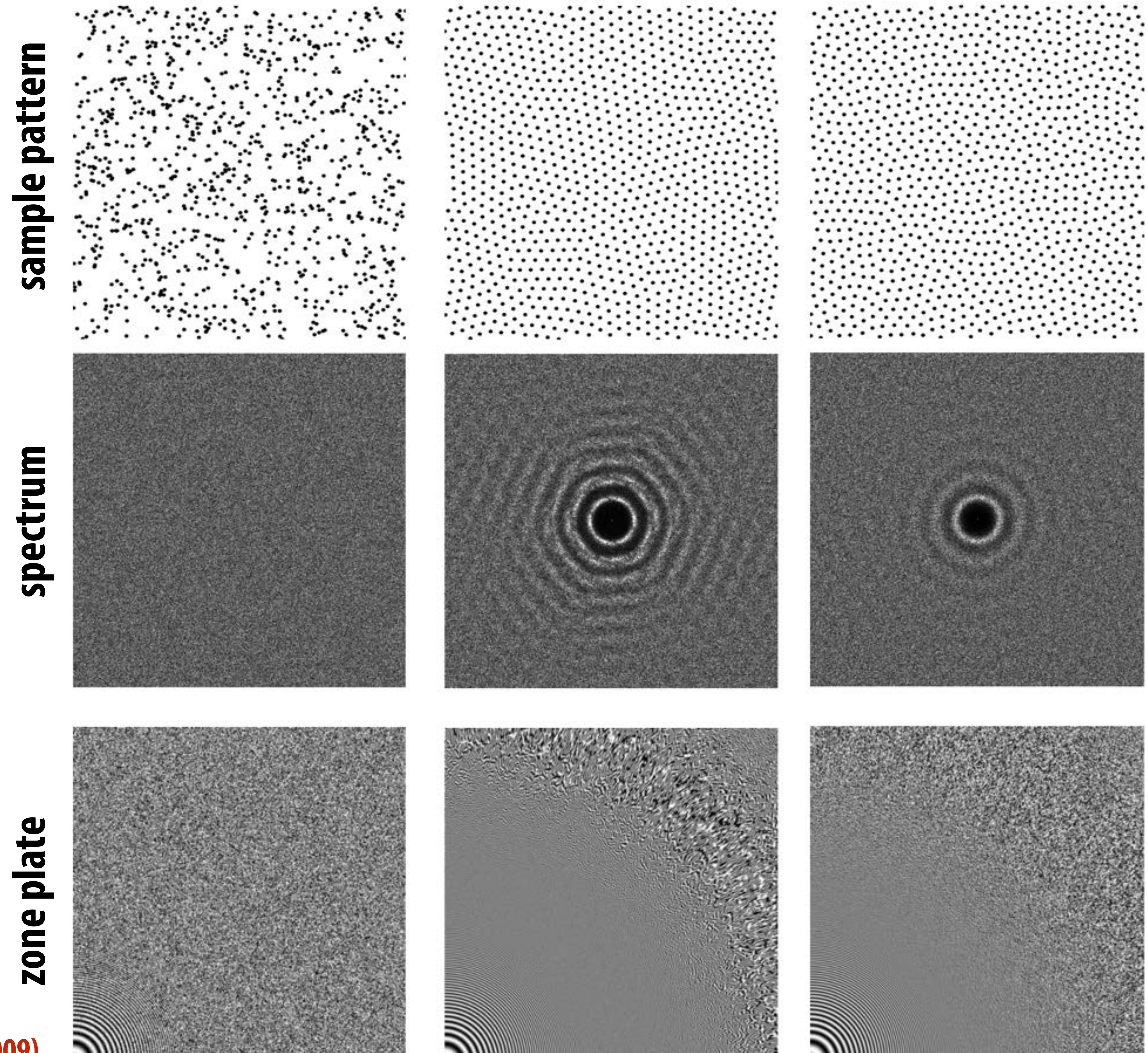
Blue Noise - Fourier Transform

- Can analyze quality of a sample pattern in *Fourier domain*



- Regular pattern has "spikes" at regular intervals
- Blue noise is spread evenly over all frequencies in all directions
- bright center "ring" corresponds to sample spacing

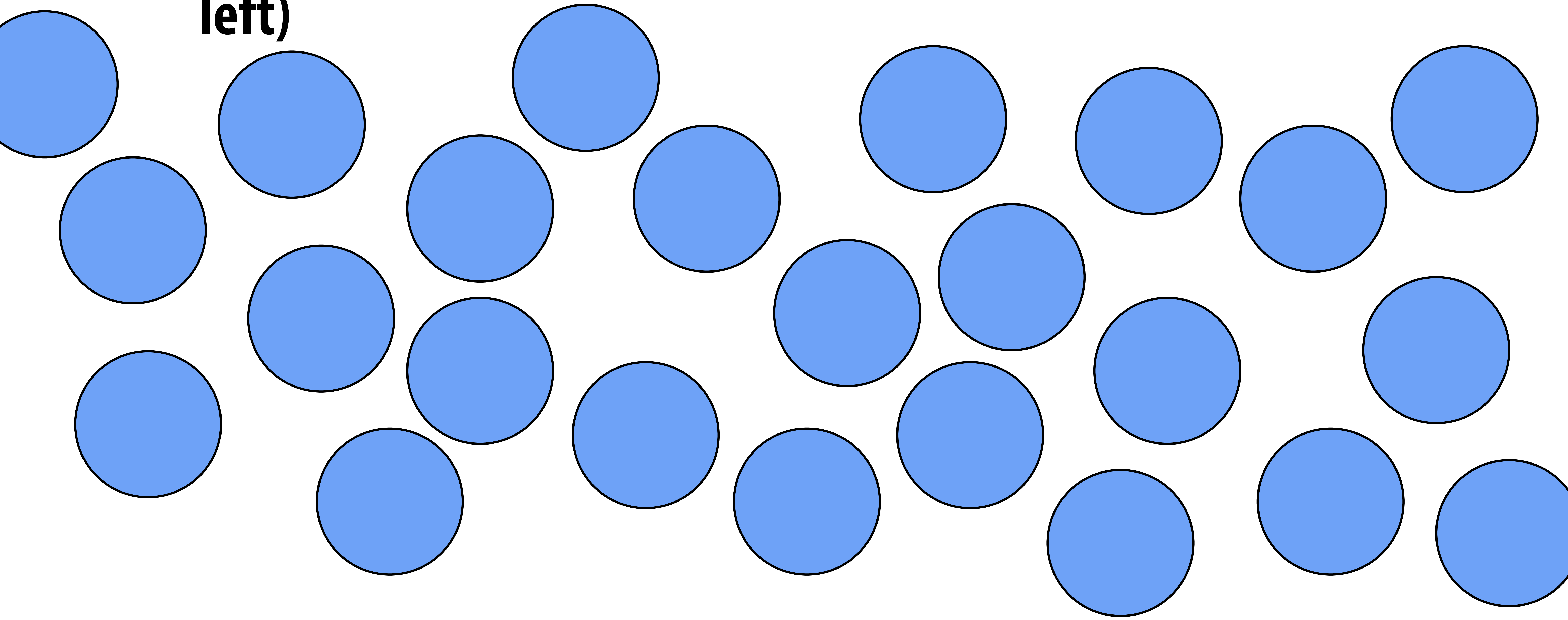
Spectrum affects reconstruction quality



(from Balzer et al 2009)

Poisson Disk Sampling

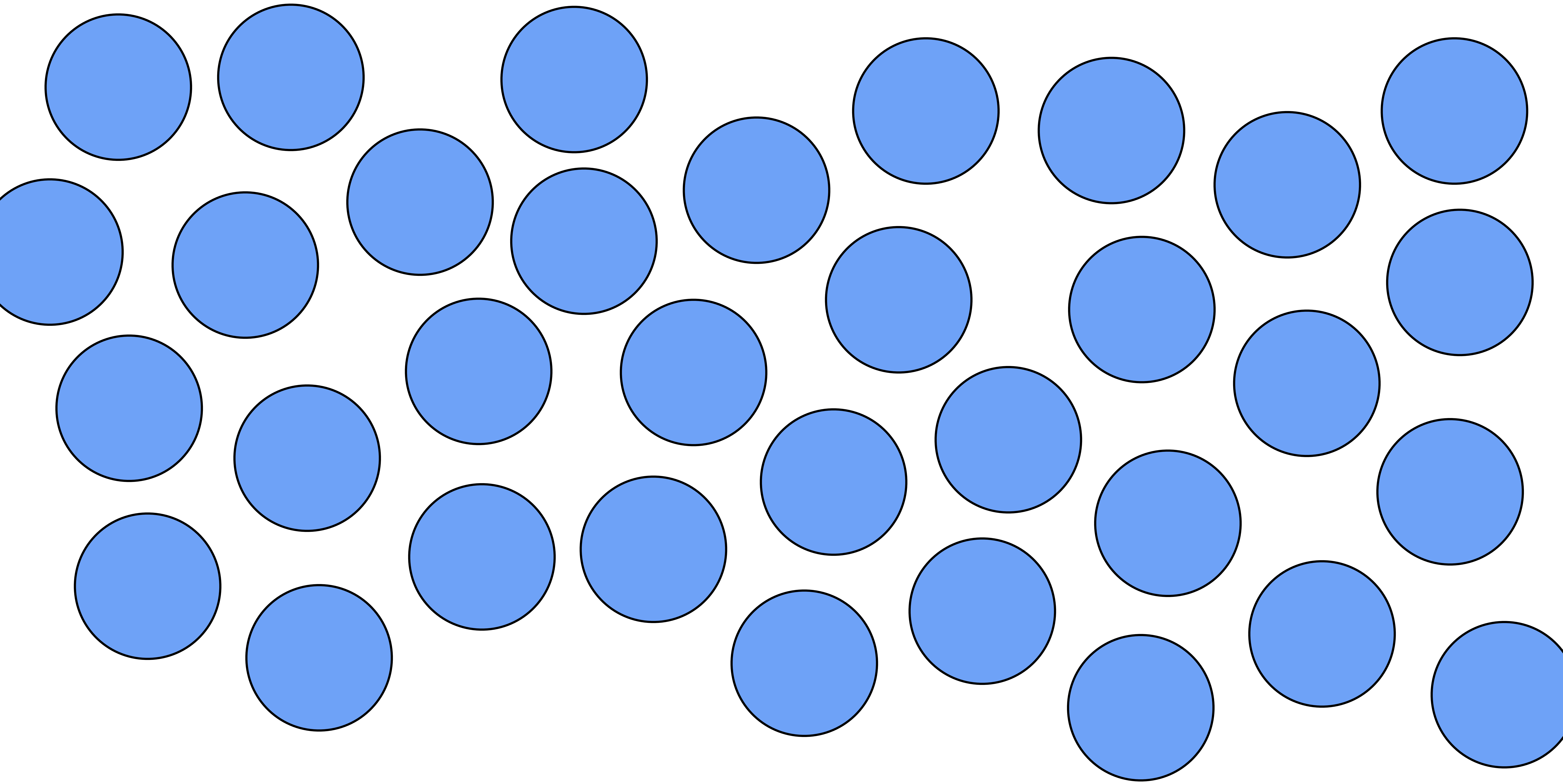
- **How do you generate a "nice" sample?**
- **One of the earliest algorithms: *Poisson disk sampling***
- **Iteratively add random non-overlapping disks (until no space left)**



Decent spectral quality, but we can do better.

Lloyd Relaxation

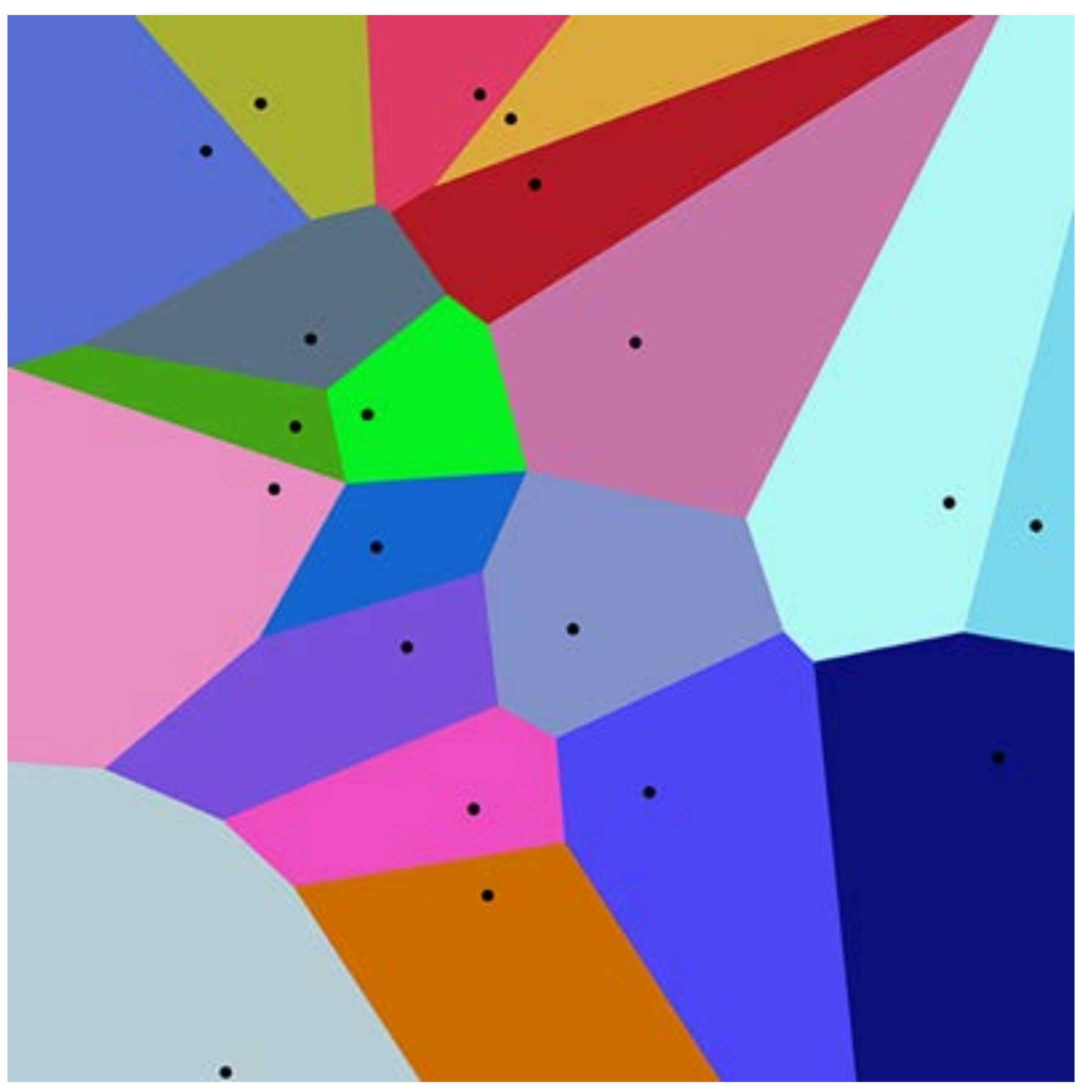
- Iteratively move each disk to the center of its neighbors



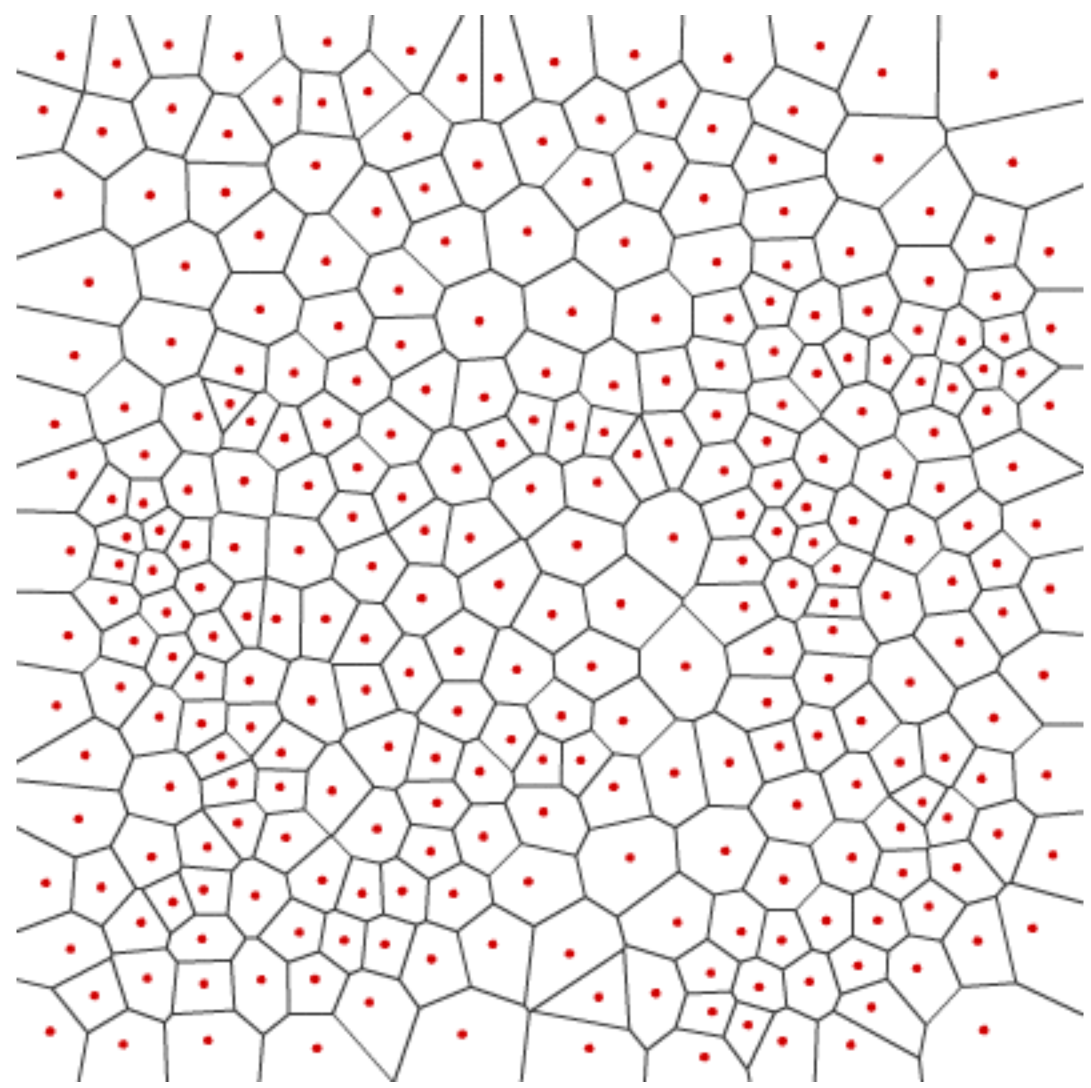
Better spectral quality, slow to converge. Can do better yet...

Voronoi-Based Methods

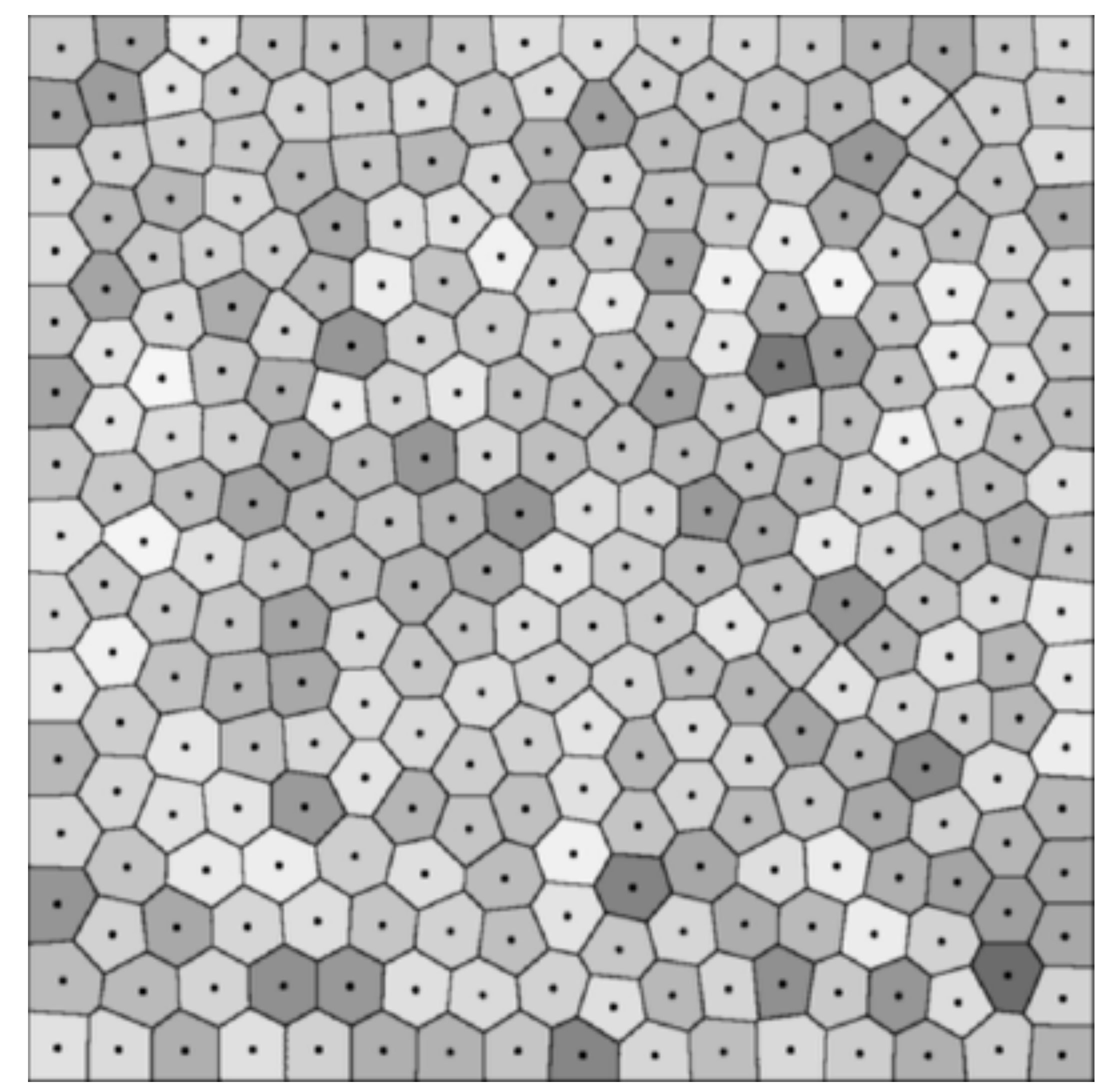
- Natural evolution of Lloyd
- Associate each sample with set of closest points (*Voronoi cell*)
- Optimize qualities of this *Voronoi diagram*
- E.g., sample is at cell's *center of mass*, cells have same area,



Voronoi



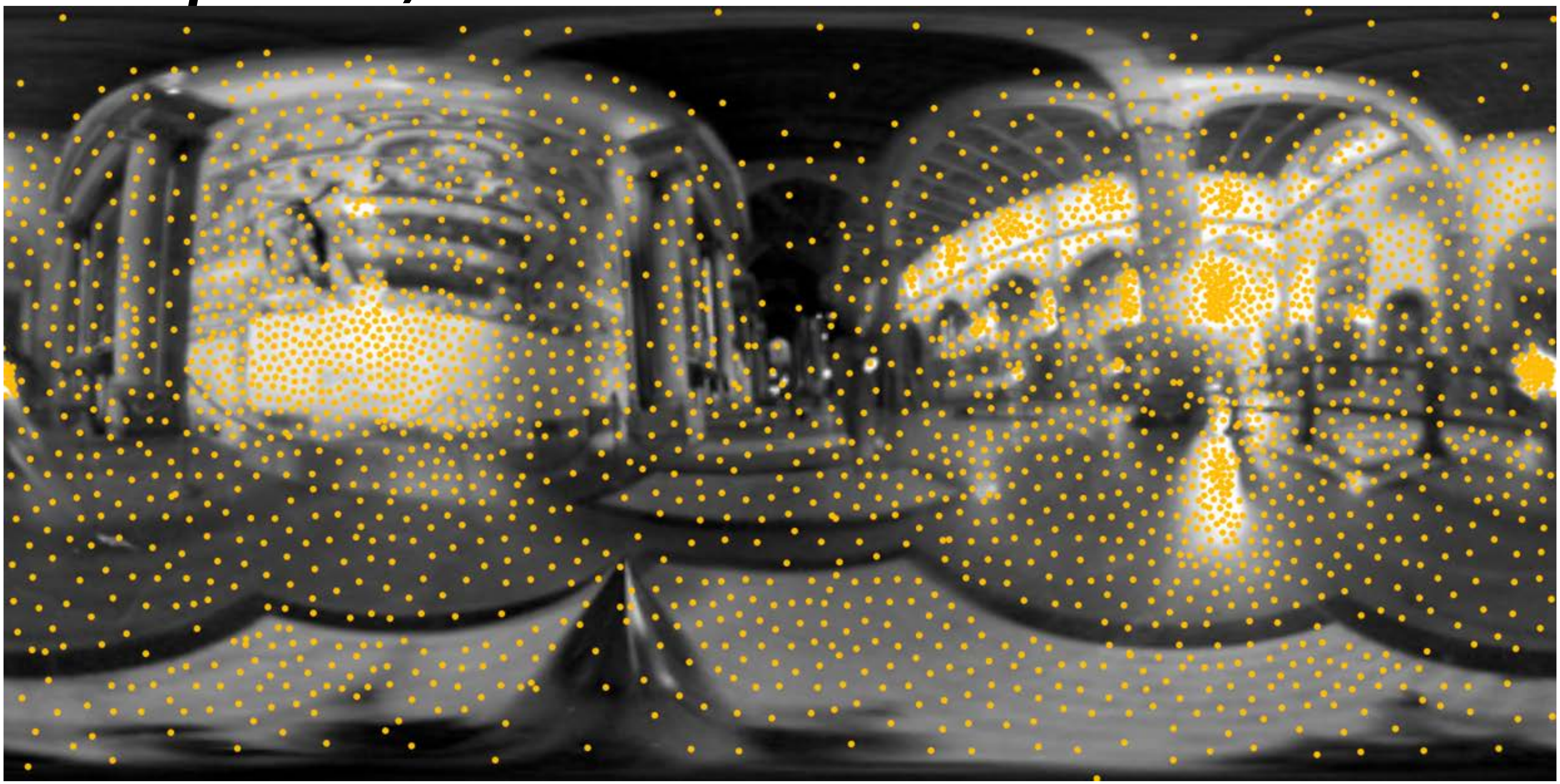
centroidal



equal area

Adaptive Blue Noise

- Can adjust cell size to sample a given density (e.g., *importance*)

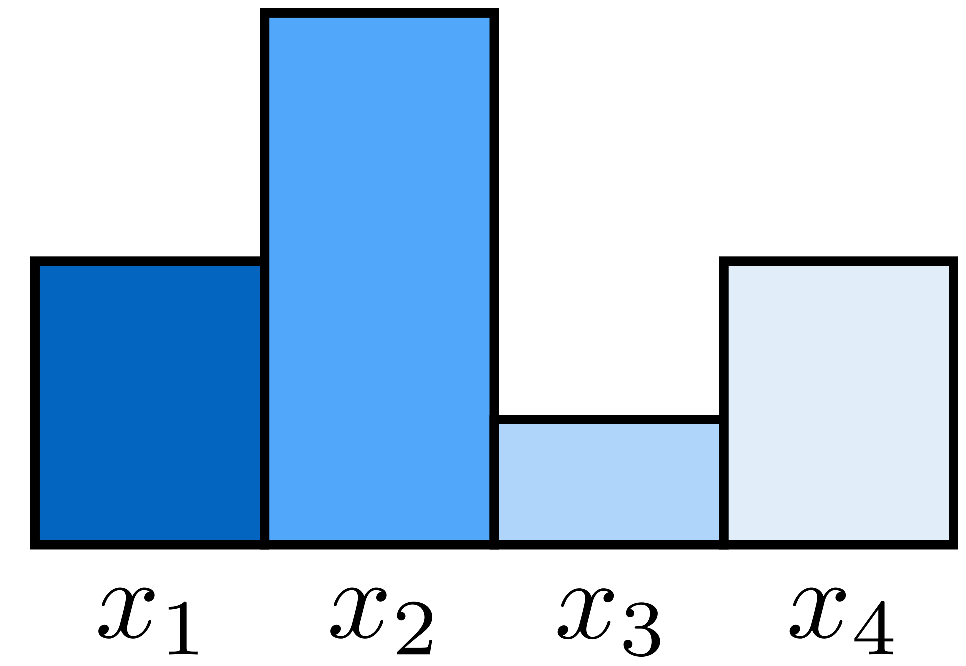


Computational tradeoff: expensive* precomputation / efficient sampling.

*But these days, not *that* expensive...

How do we efficiently sample from a large distribution?

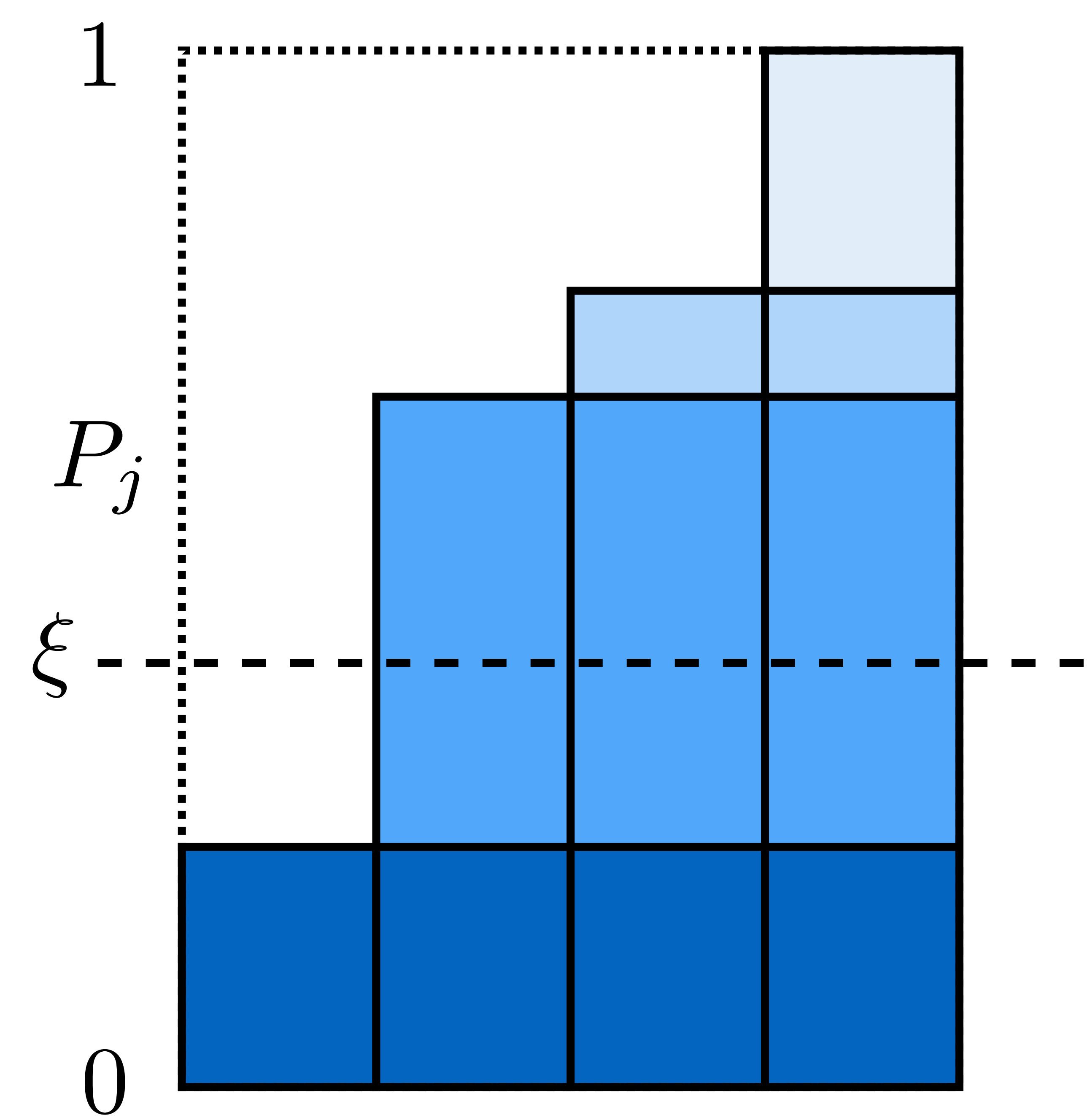
Sampling from the CDF



To randomly select an event, select x_i if

$$P_{i-1} < \xi < P_i$$

Uniform random variable $\in [0, 1]$

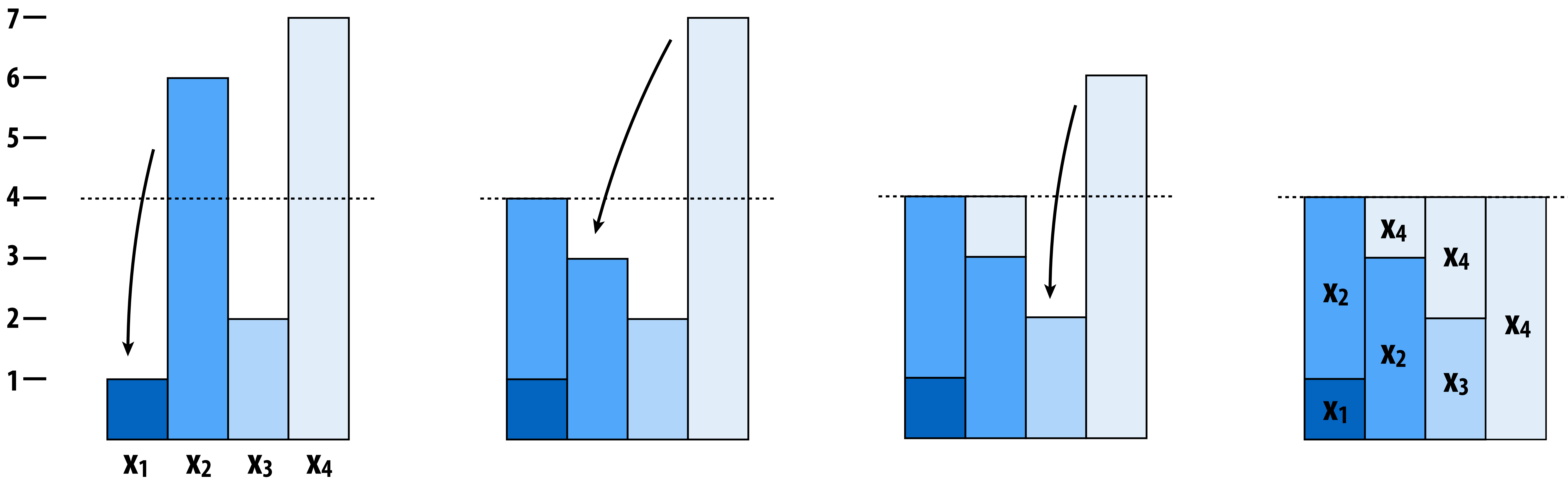


e.g., # of pixels in an environment map (big!)

Cost? O(n log n)

Alias Table

- Get amortized O(1) sampling by building "alias table"
- Basic idea: rob from the rich, give to the poor (O(n)):

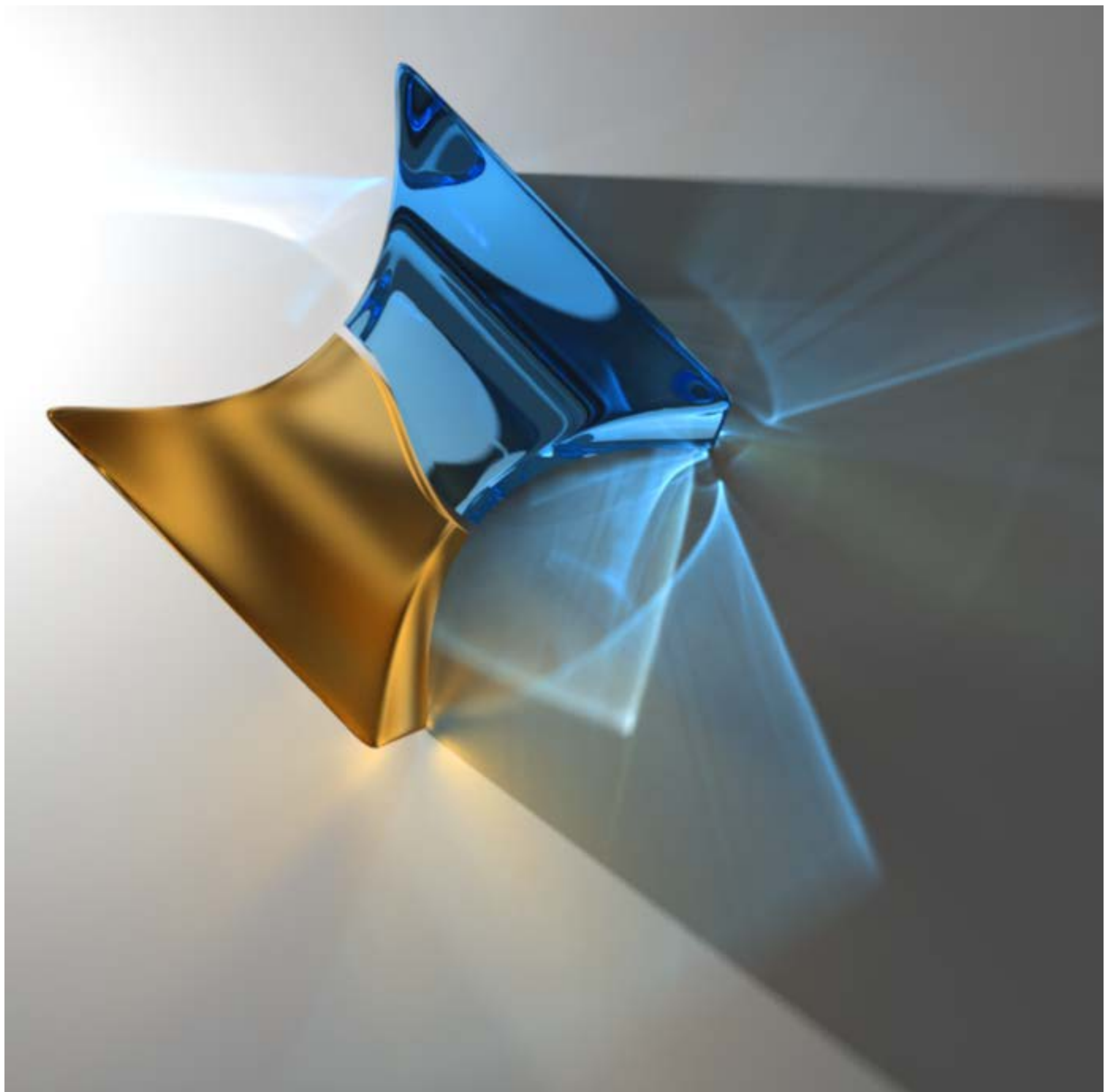


- Table just stores two identities & ratio of heights per column
- To sample:
  - pick uniform # between 1 and *n*
  - biased coin flip to pick one of the two identities in *n*th column

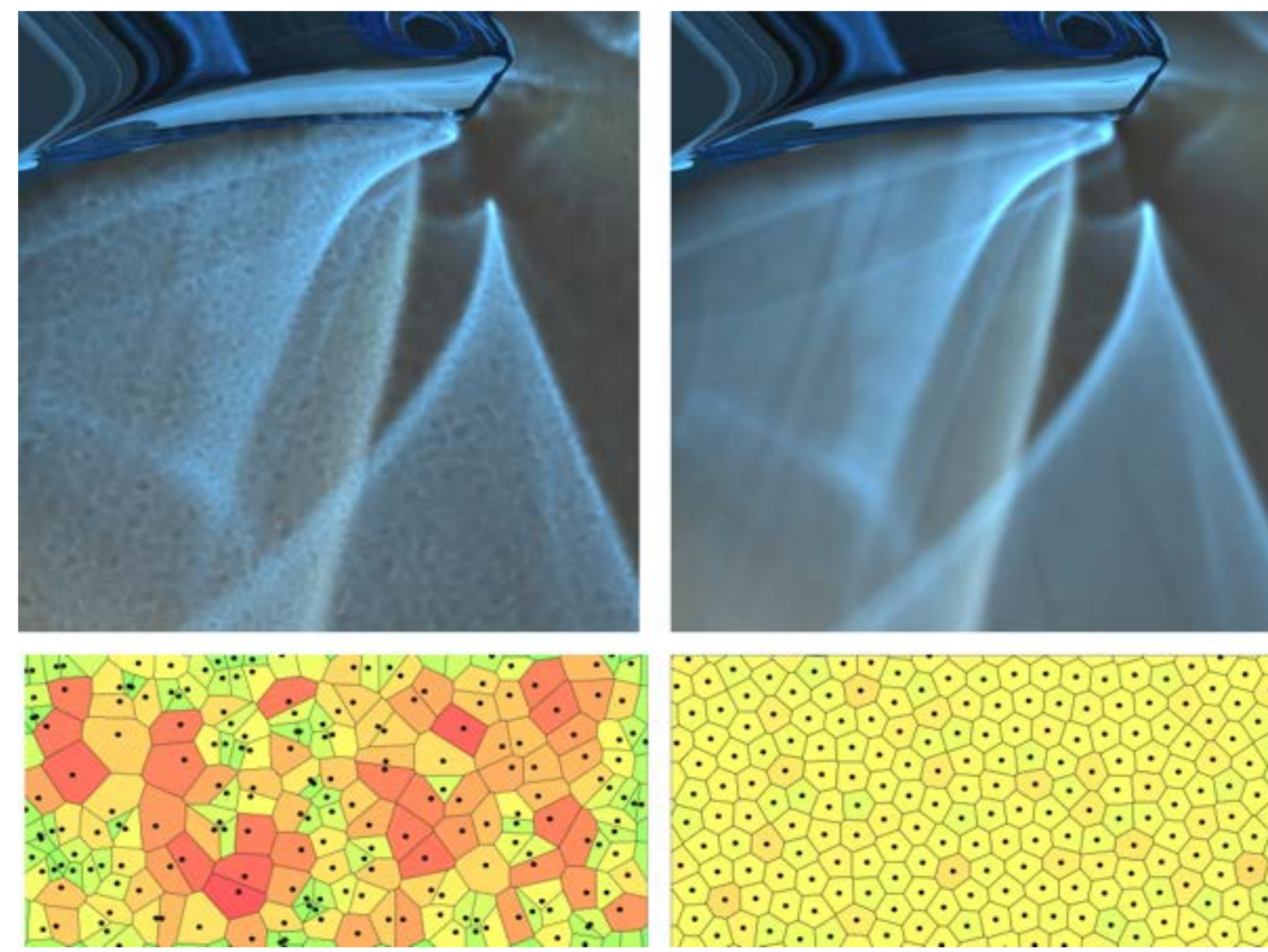
Ok, great!
Now that we've mastered Monte Carlo rendering, what other techniques are there?

Photon Mapping

- Trace particles from light, deposit "photons" in kd-tree
- Especially useful for, e.g., caustics, participating media



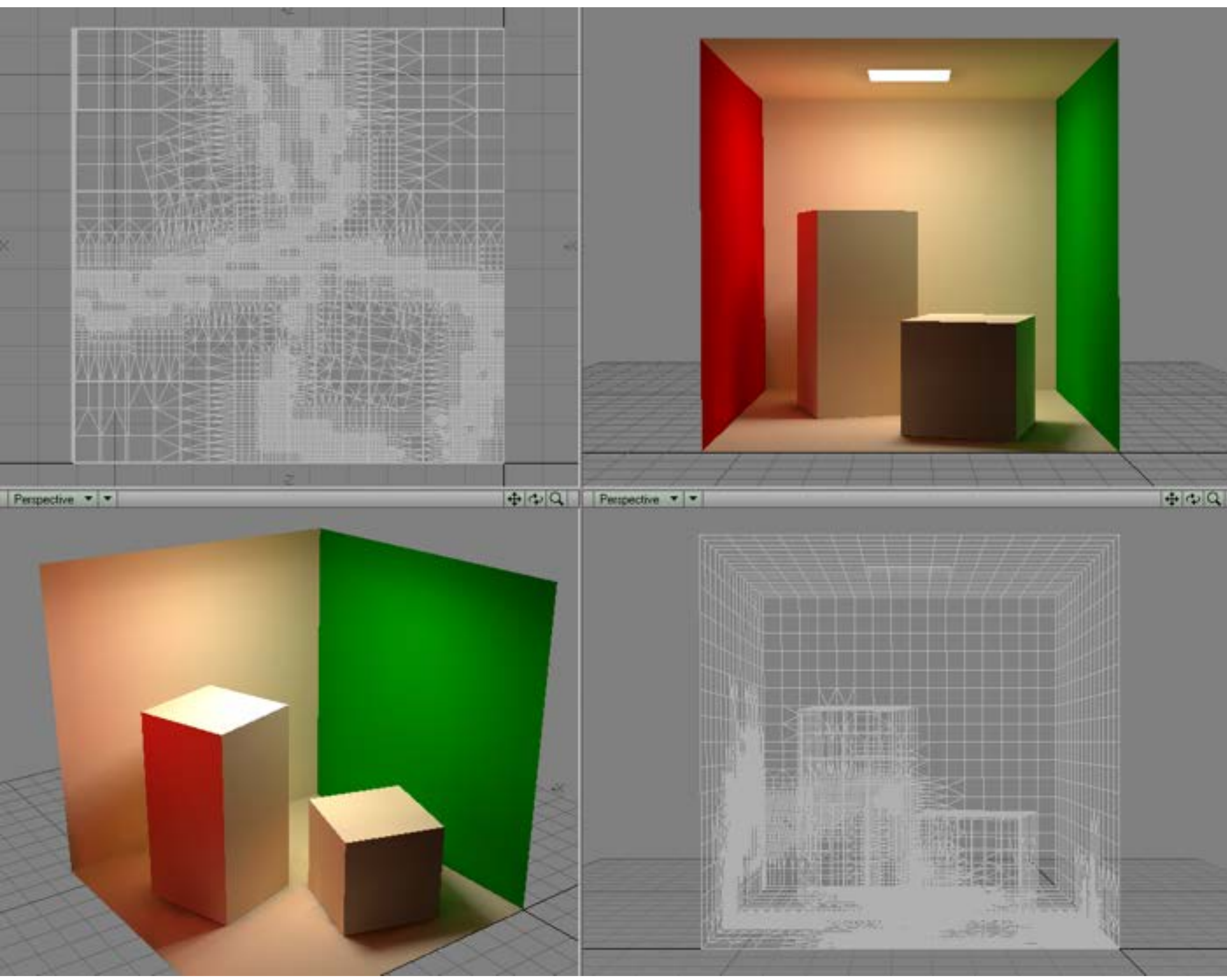
Interestingly enough, Voronoi diagrams also used to improve photon distribution!



(from Spencer & Jones 2013)

Finite Element Radiosity

- Very different approach: transport between patches in scene
- Solve large linear system for equilibrium distribution
- Good for diffuse lighting; hard to capture other light paths

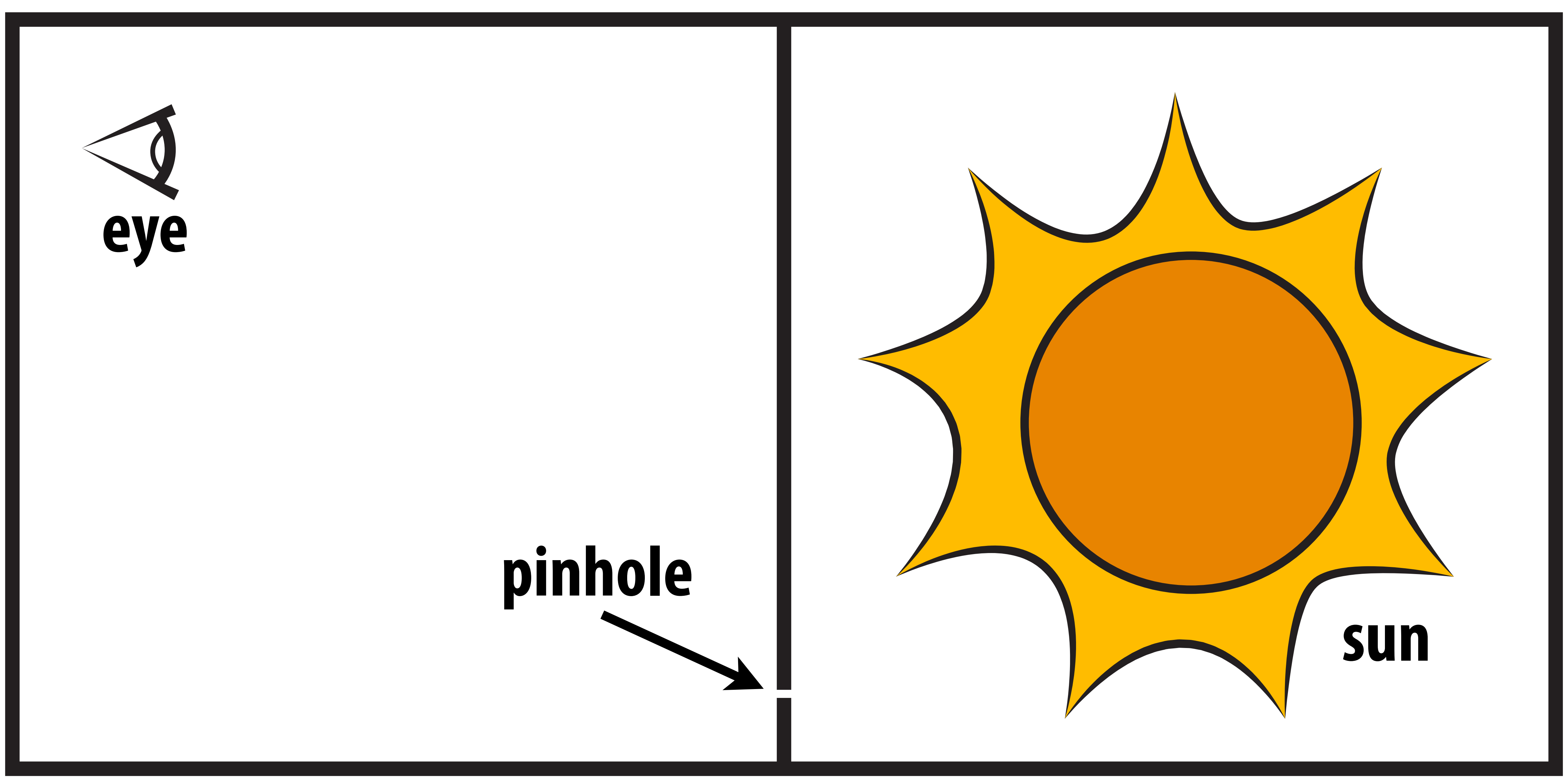


Consistency & Bias in Rendering Algorithms

method	consistent?	unbiased?
rasterization	NO	NO
path tracing	ALMOST	ALMOST
bidirectional path tracing	YES	YES
Metropolis light transport	YES	YES
photon mapping	YES	NO
radiosity	NO	NO

Can you certify a renderer?

- Grand challenge: write a renderer that comes with a *certificate* (i.e., provable, formally-verified guarantee) that the image produced represents the illumination in a scene.
- Harder than you might think!
- Inherent limitation of sampling: you can never be 100% certain that you didn't miss something important.



Can always make sun brighter, hole smaller...!

Moment of Zen