More Variance Reduction

Computer Graphics
CMU 15-462/15-662

Review: Path Space Formulation of Light Transport

- We had been using the recursive rendering equation:

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

- Importance sampling can be done by making 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

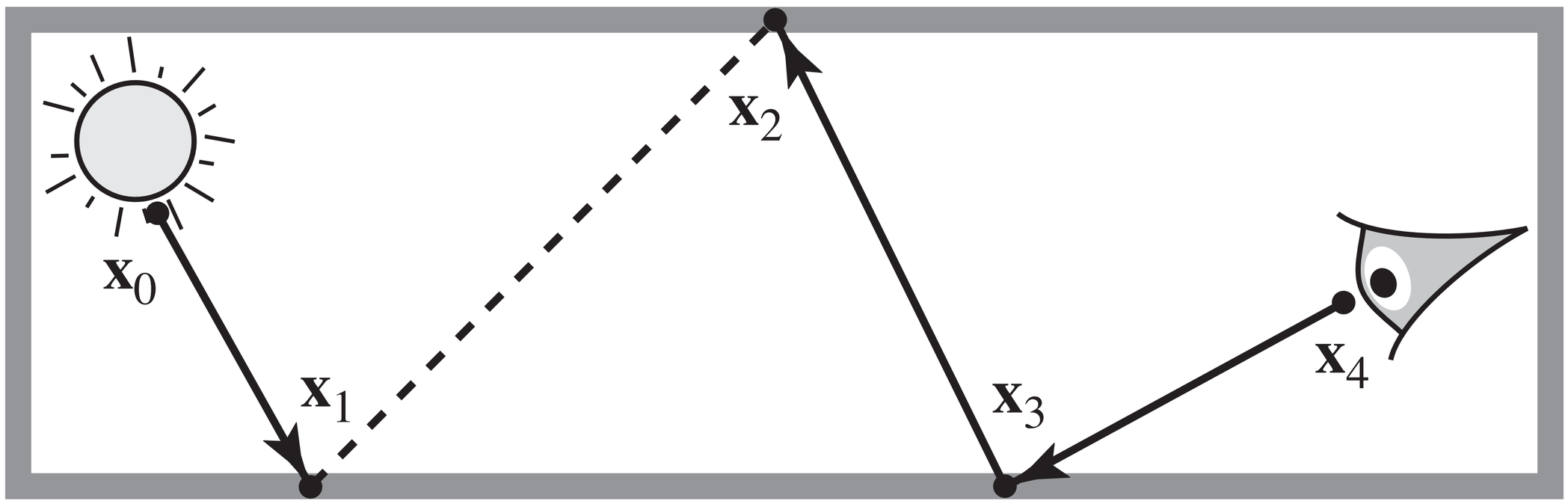
all possible paths

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

How do we choose paths—and path lengths?

Bidirectional Path Tracing

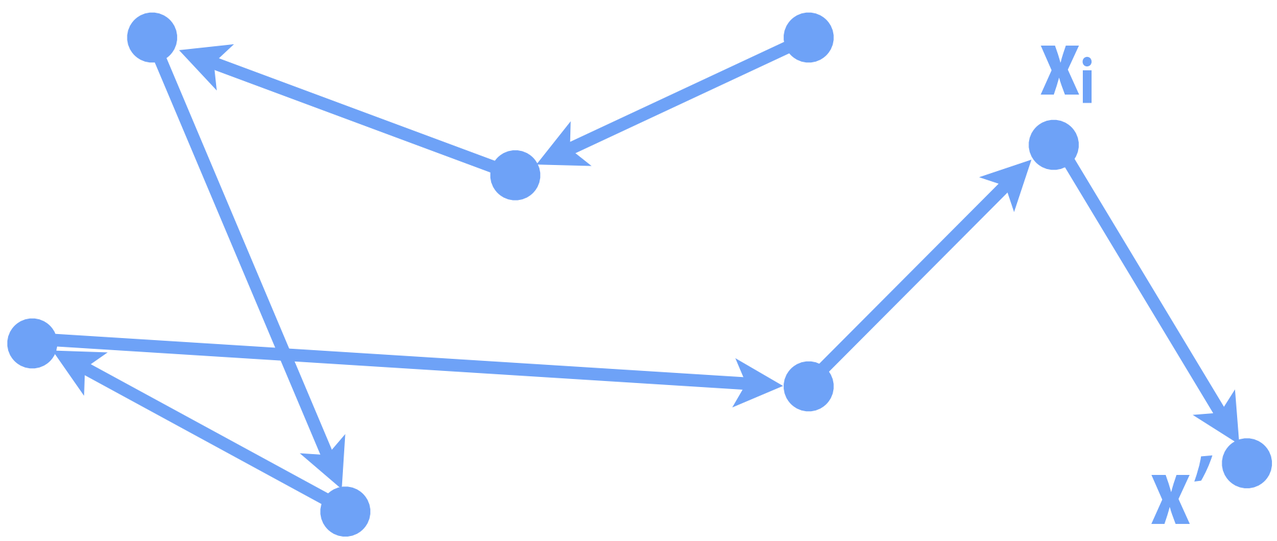
- Forward path tracing: no control over path length (hits light after n bounces, or gets terminated by Russian Roulette)
- Idea: connect paths from light, eye ("bidirectional")



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

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



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

if random # in [0,1] < α:

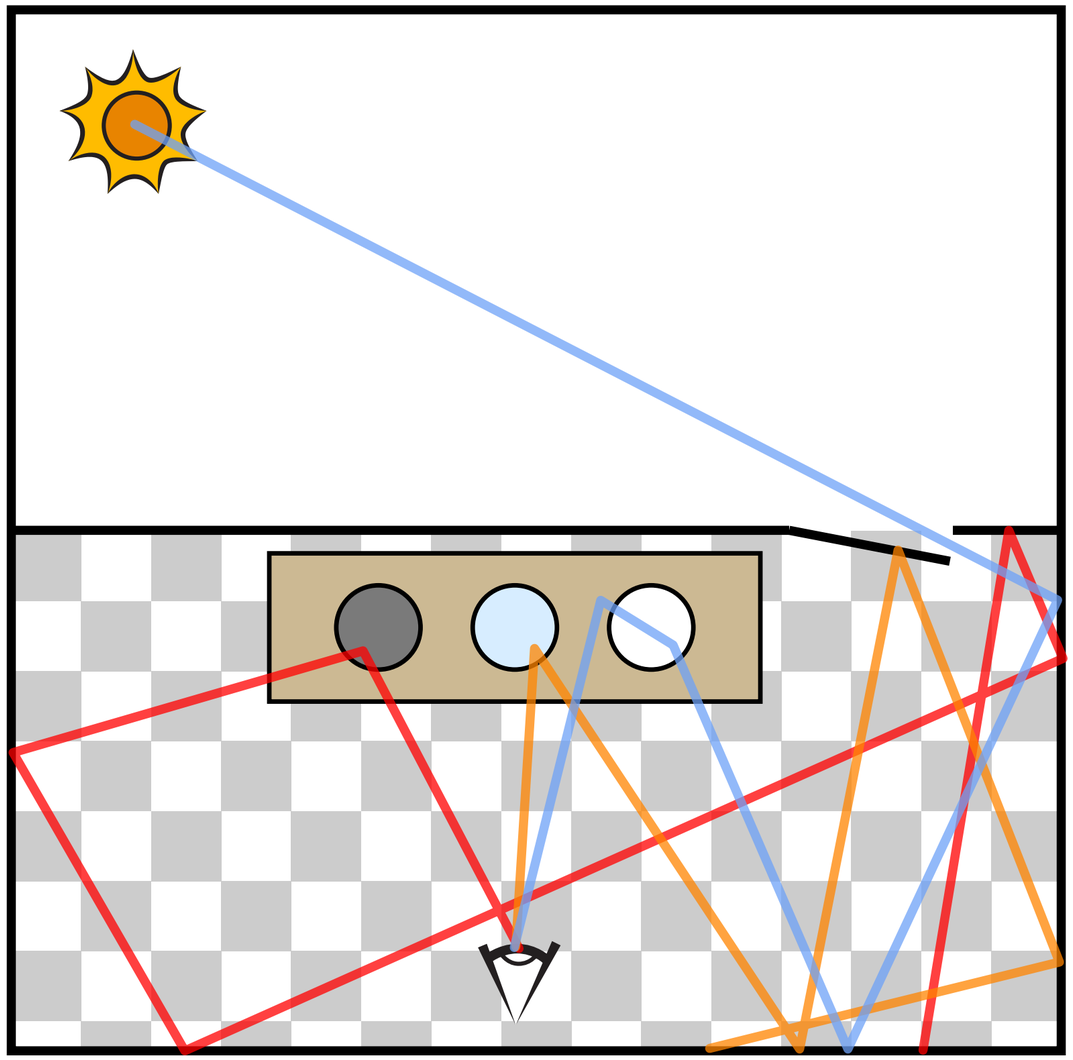
$x_{i+1} = x'$

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")

Good paths can be hard to find!



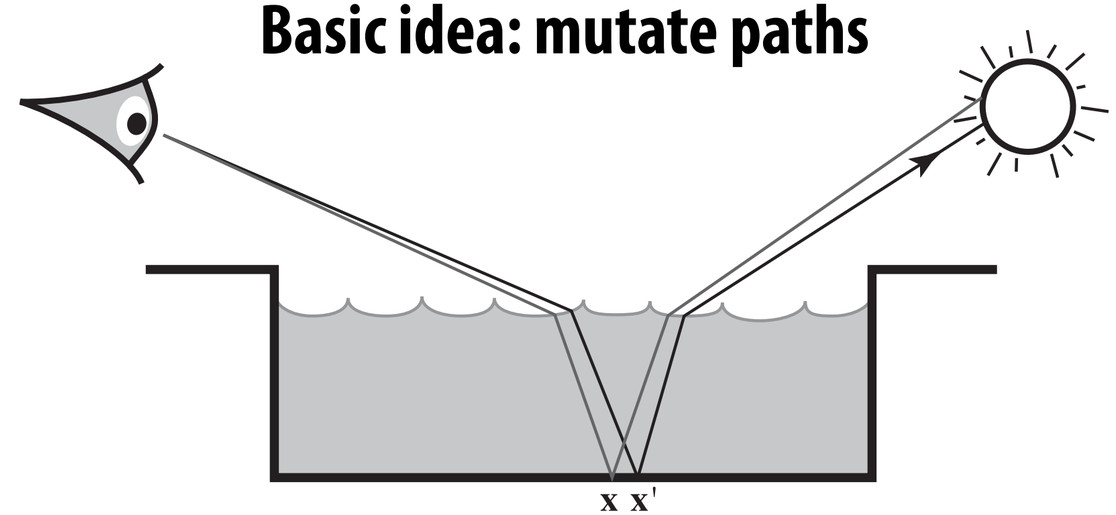
bidirectional path tracing

Metropolis light transport (MLT)

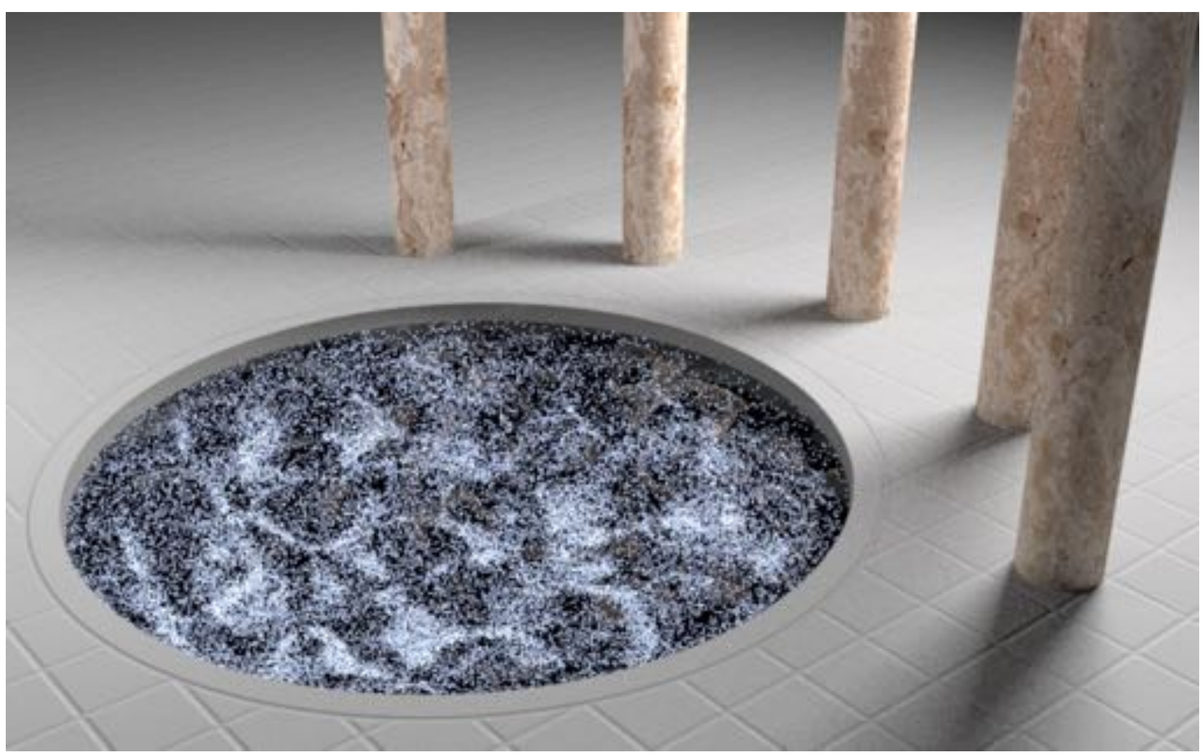
Idea:
Once we find a good path, perturb it to find nearby "good" paths.

Metropolis Light Transport

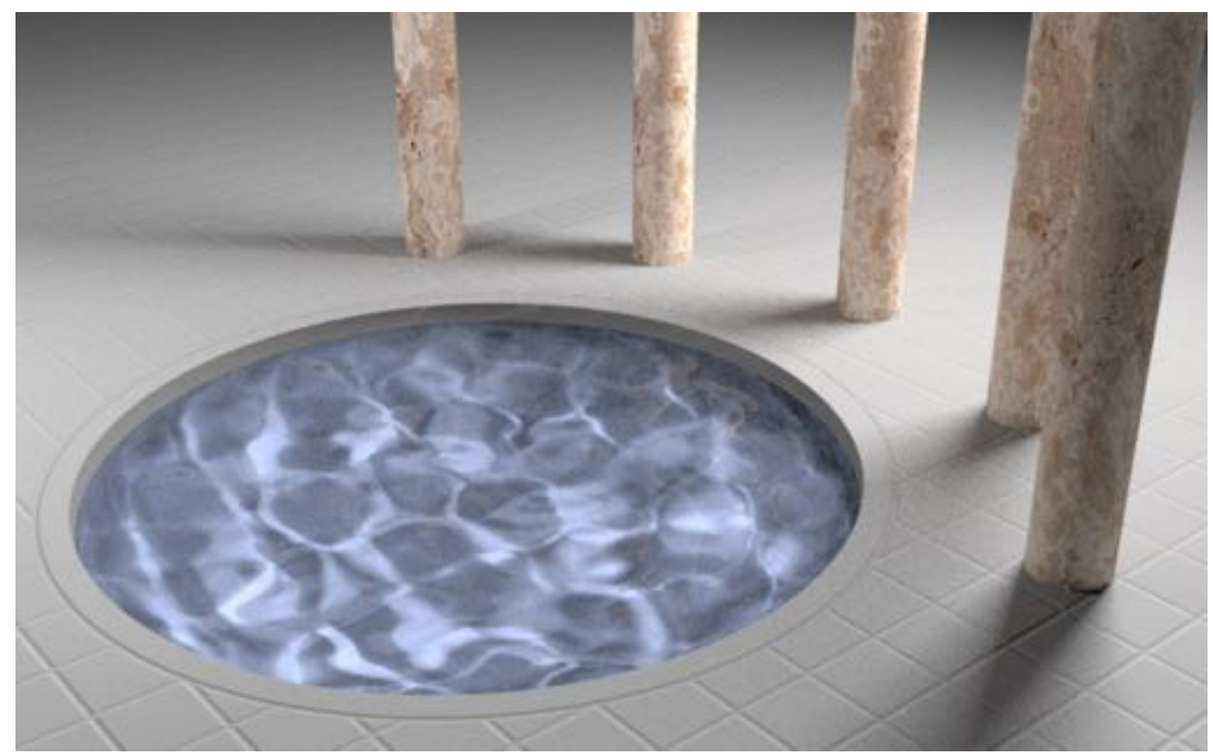
Basic idea: mutate paths



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



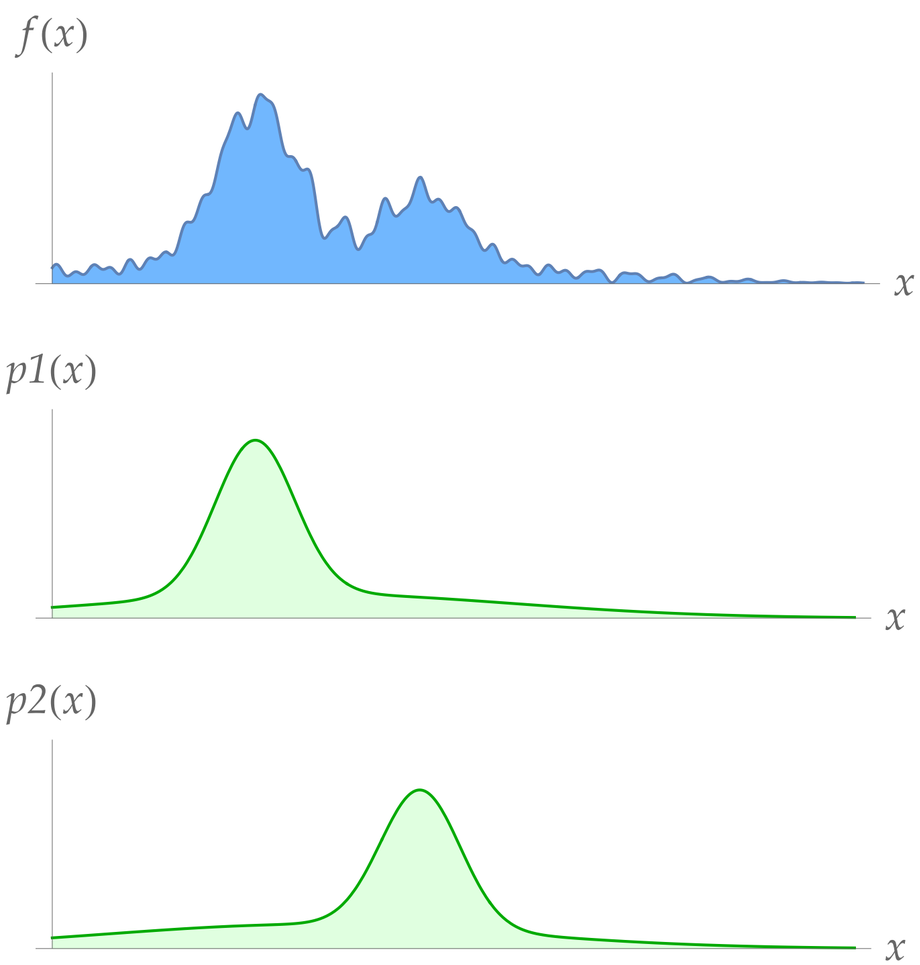
path tracing



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



sum over strategies

sum over samples

jth sample taken with ith strategy

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

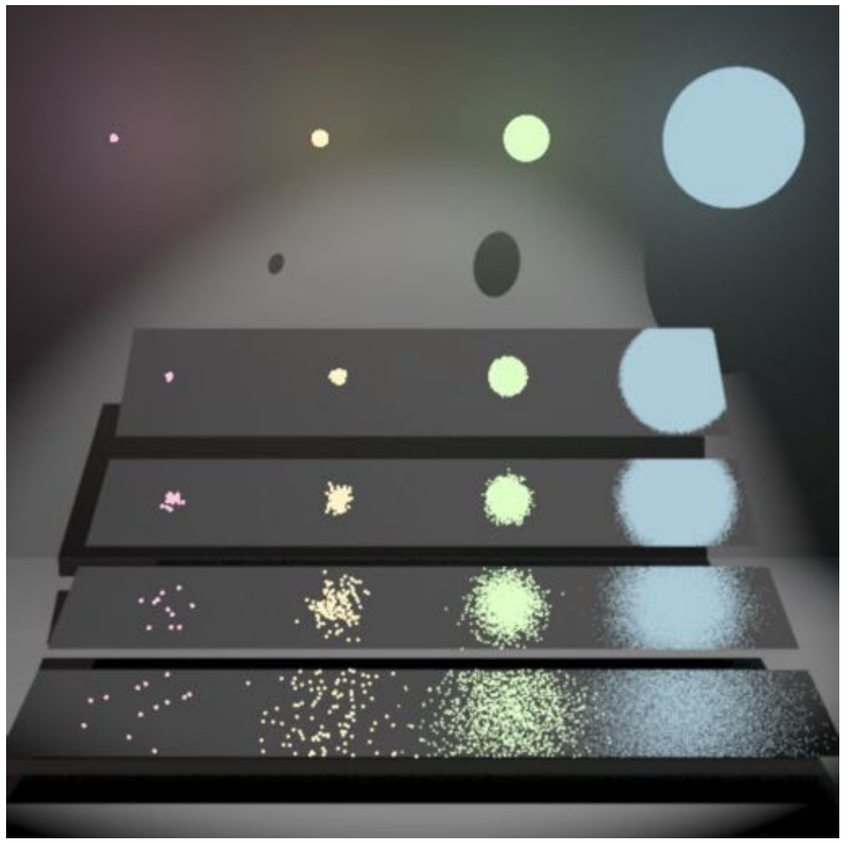
total # of samples

fraction of samples taken w/ kth strategy

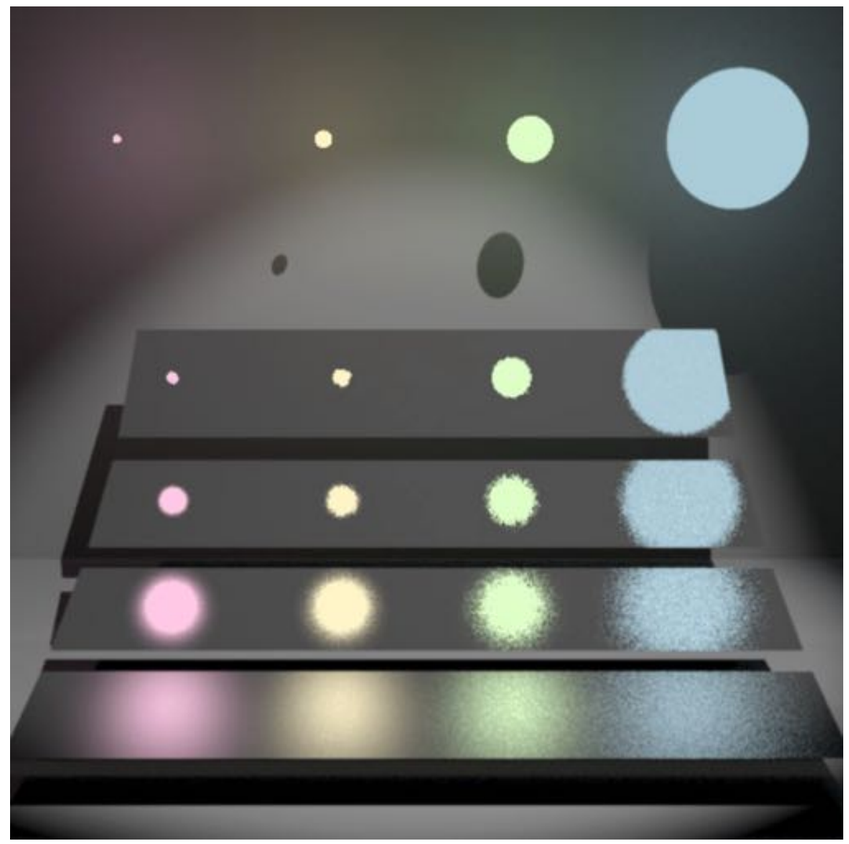
kth importance density

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

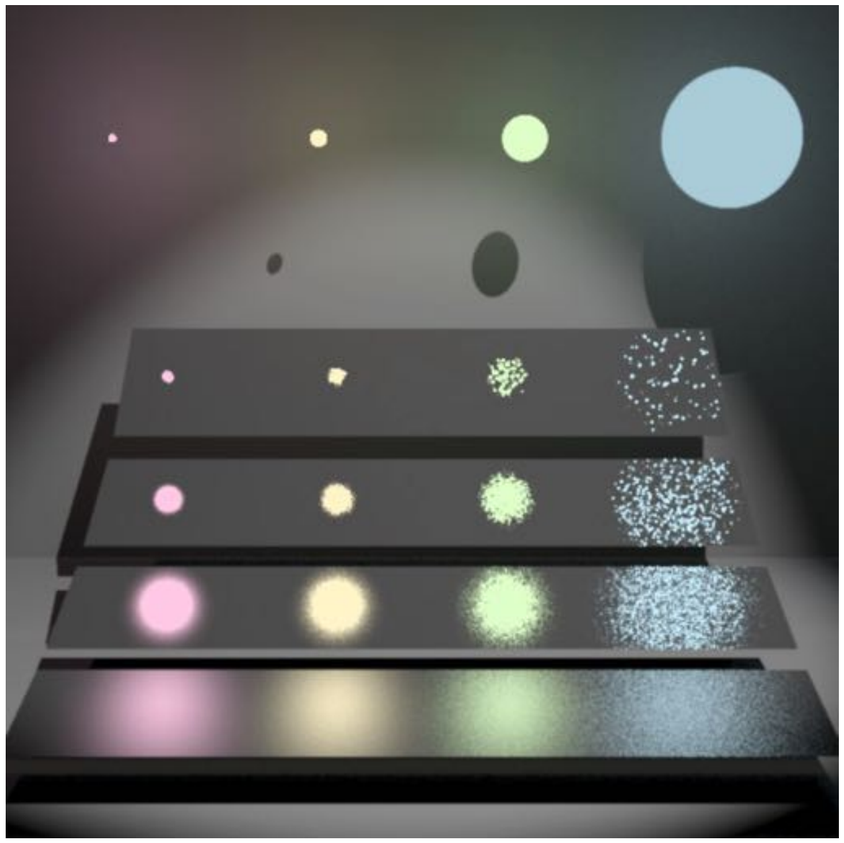
Multiple Importance Sampling: Example



sample materials



multiple importance sampling
(power heuristic)



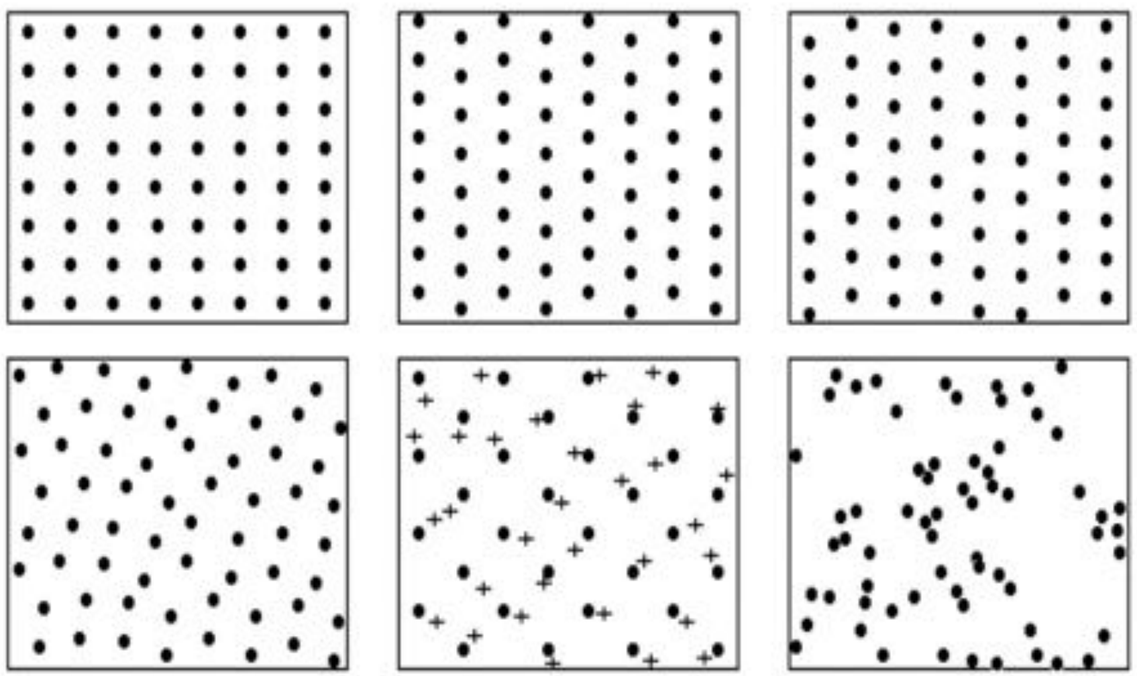
sample lights

Ok, so importance is important.

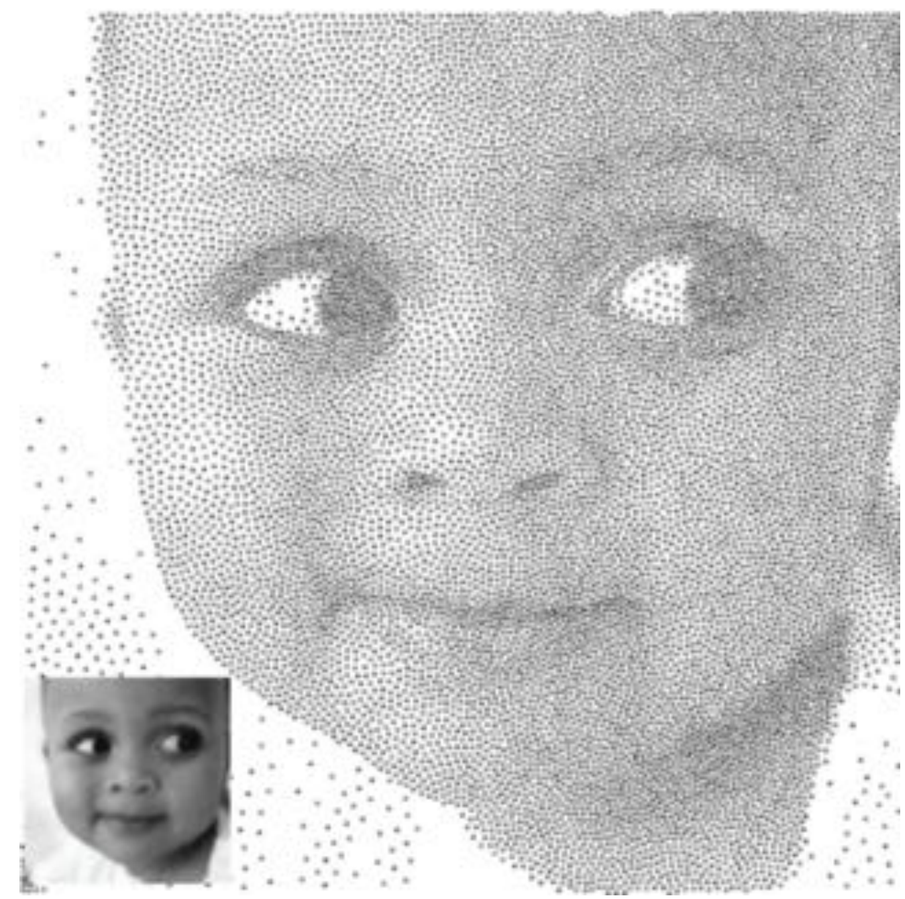
But how do we sample our function in the first place?

Sampling Patterns & Variance Reduction

- Want to pick samples according to a given density
- But even for uniform density, lots of possible sampling patterns
- Sampling pattern will affect variance (of estimator!)



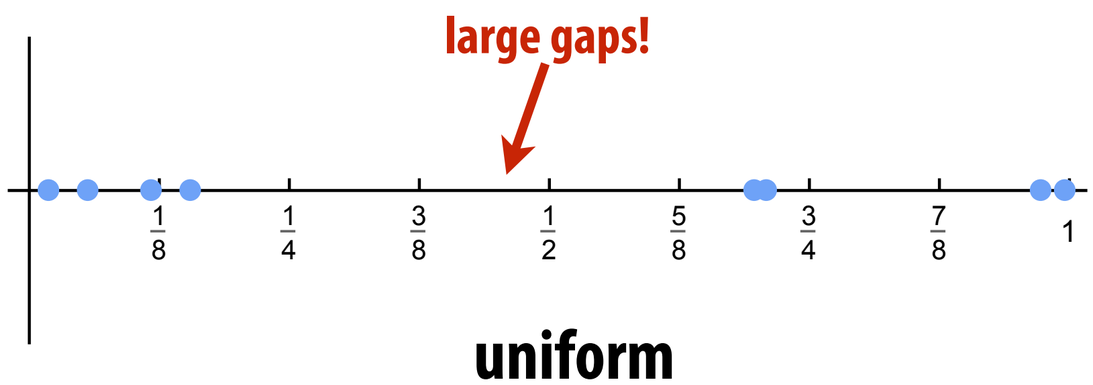
uniform sampling density

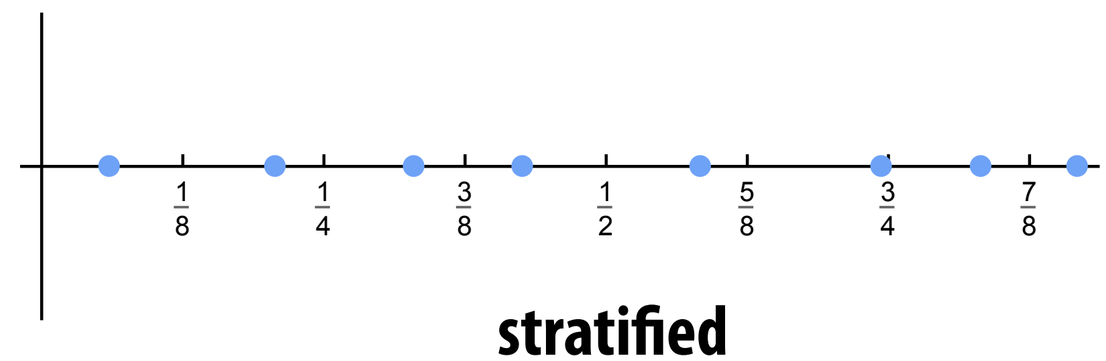


nonuniform sampling density

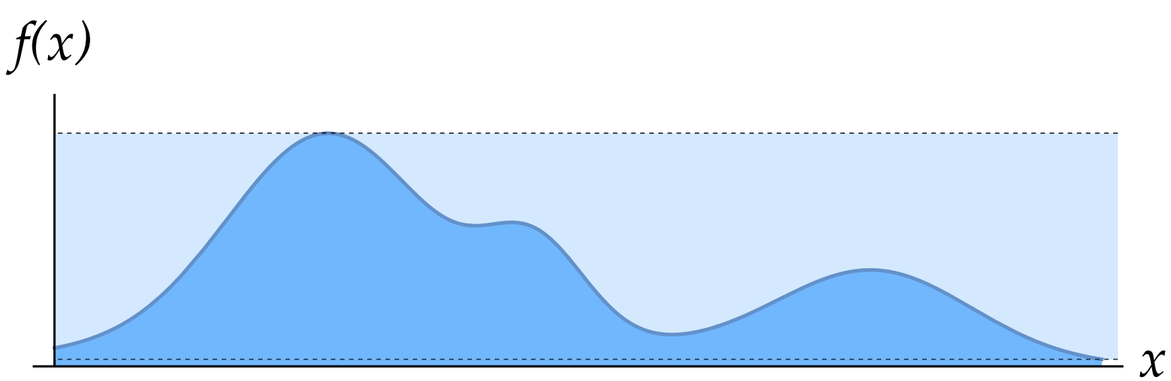
Stratified Sampling

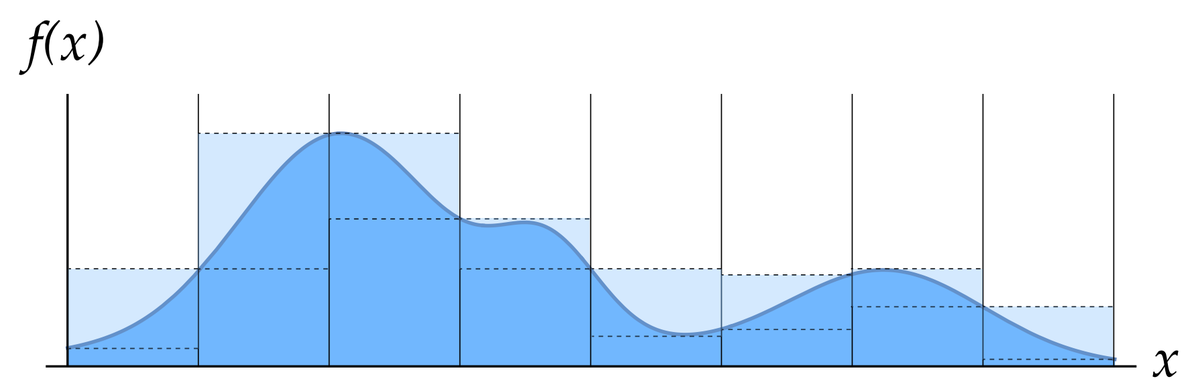
- How do we pick n values from [0,1]?
- Could just pick n samples uniformly at random
- Alternatively: split into n bins, pick uniformly in each bin





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



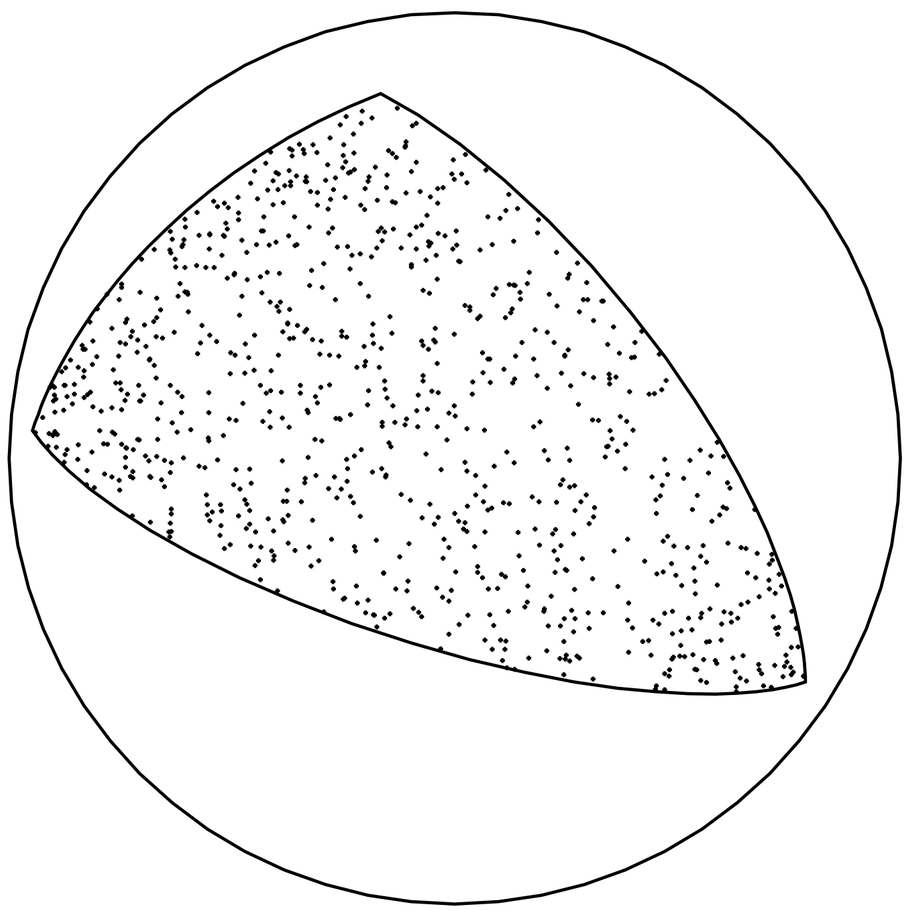


Intuition: each stratum has smaller variance. (Proof by linearity of expectation!)

Stratified Sampling in Rendering/Graphics

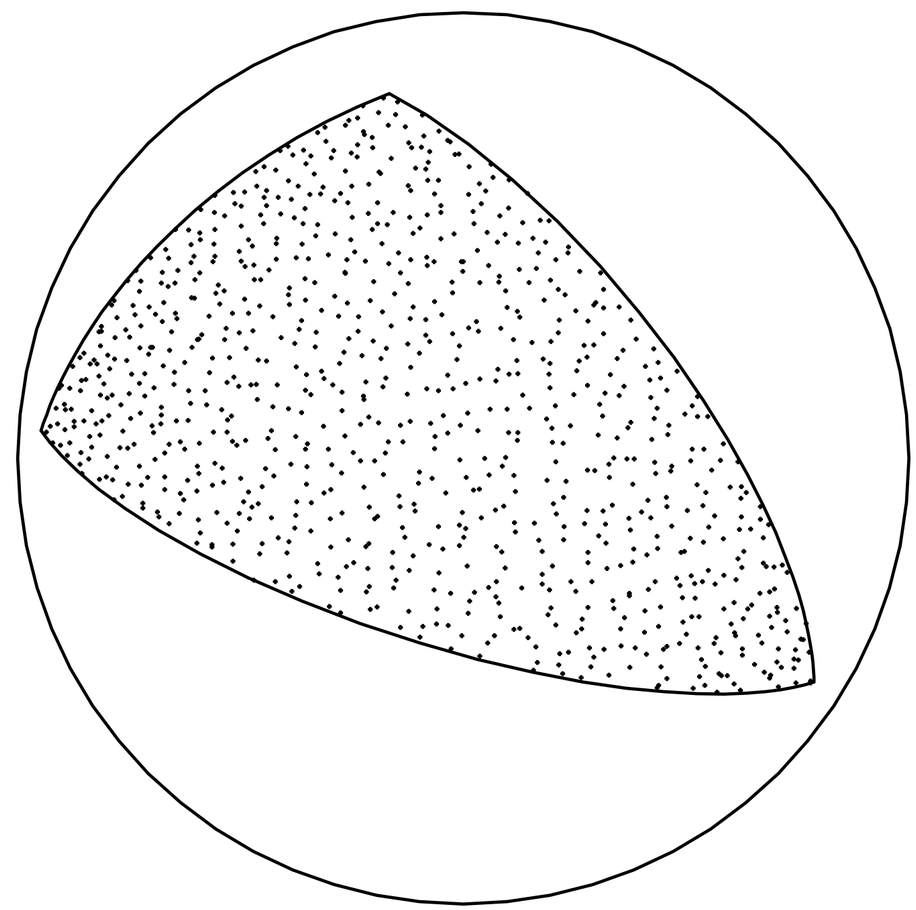
- Simply replacing uniform samples with stratified ones already improves quality of sampling for rendering (…and other graphics/visualization tasks!)

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

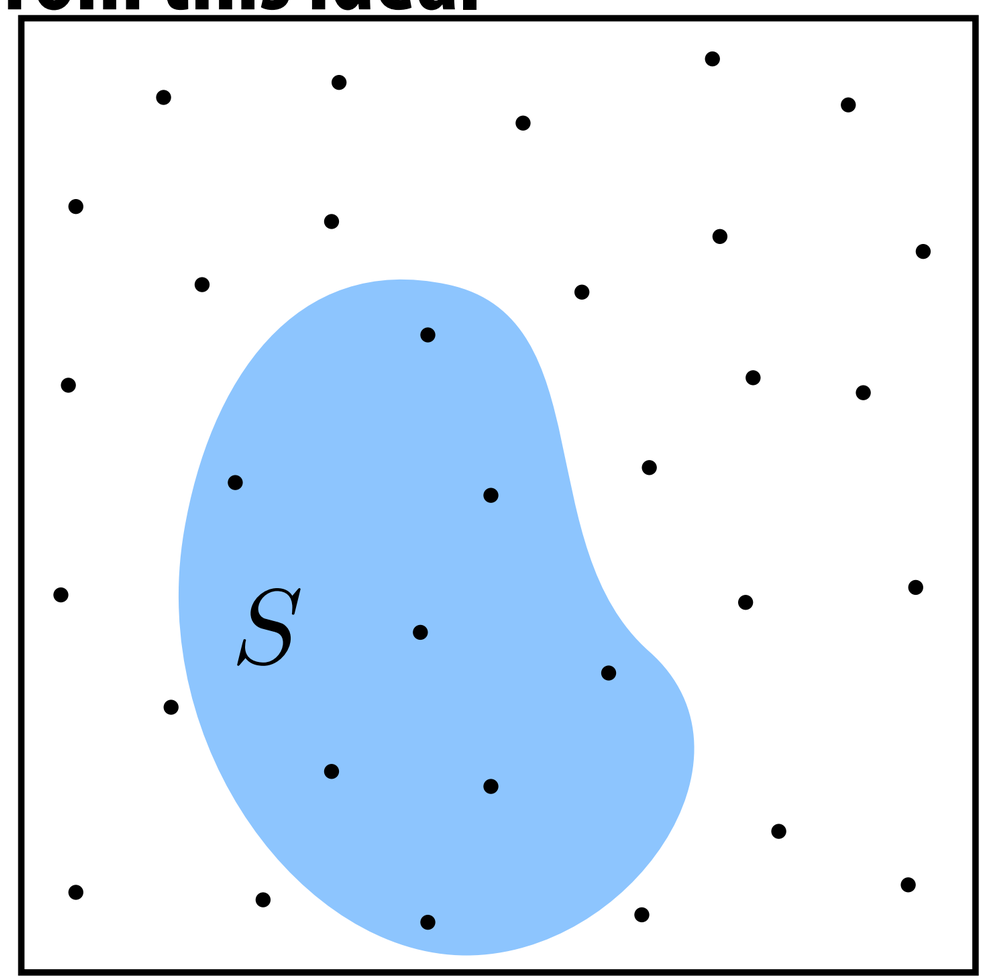
discrepancy of sample points X over a region S

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:

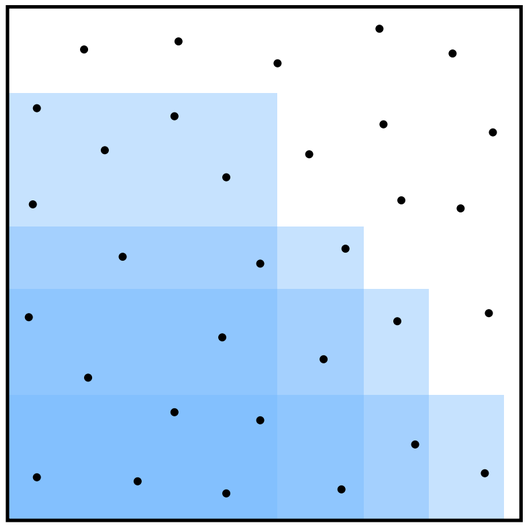
sample points in X

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

discrepancy of sample 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!
- 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

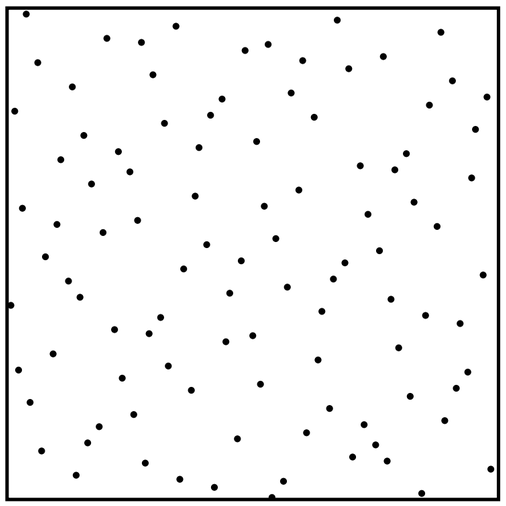
$$x_i = (\phi_{P_1}(i), \phi_{P_2}(i), \ldots, \phi_{P_k}(i))$$

kth prime number

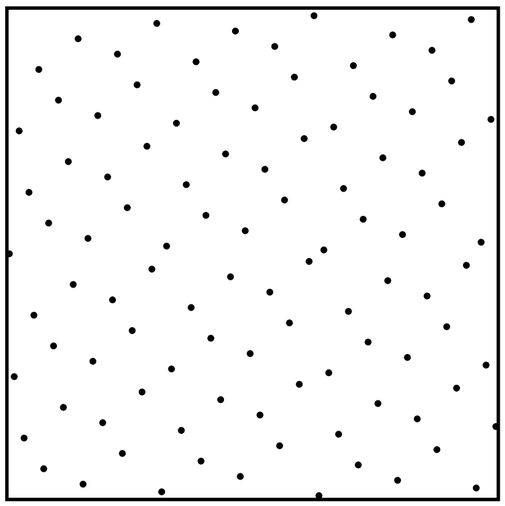
- Similarly, Hammersley sequence is

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

n must be known ahead of time!

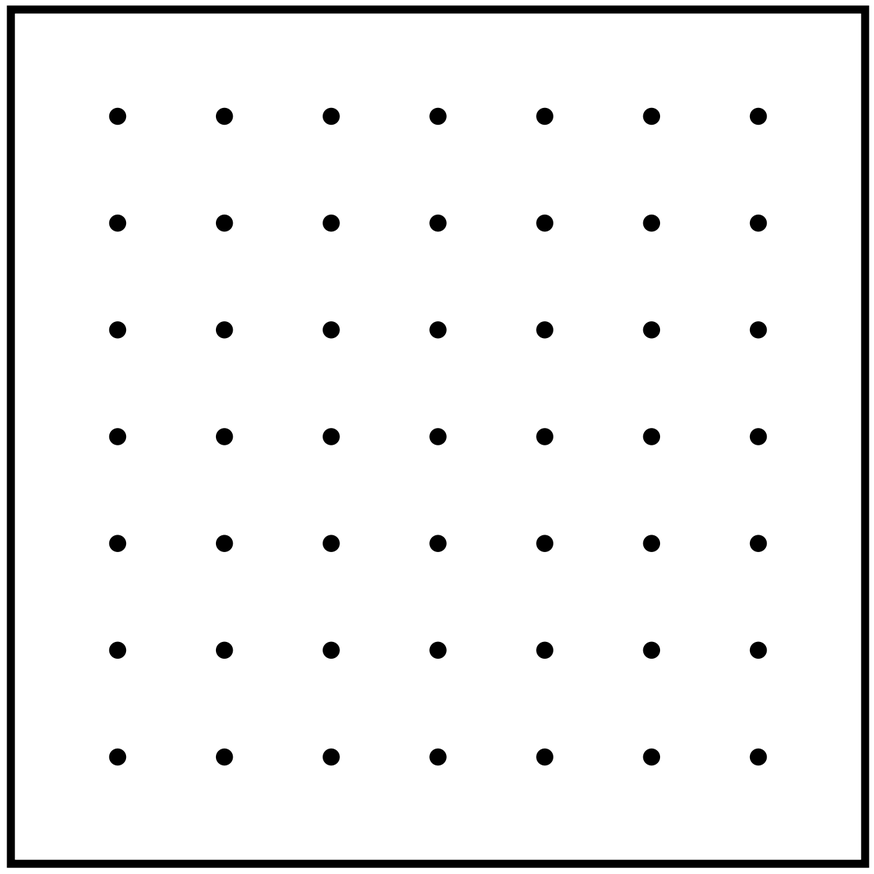


Halton



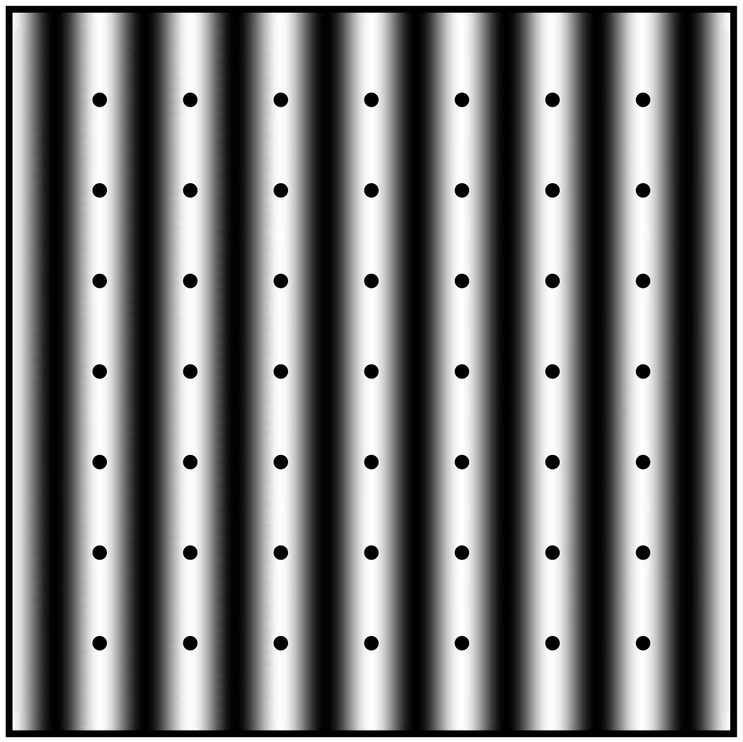
Hammersley

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

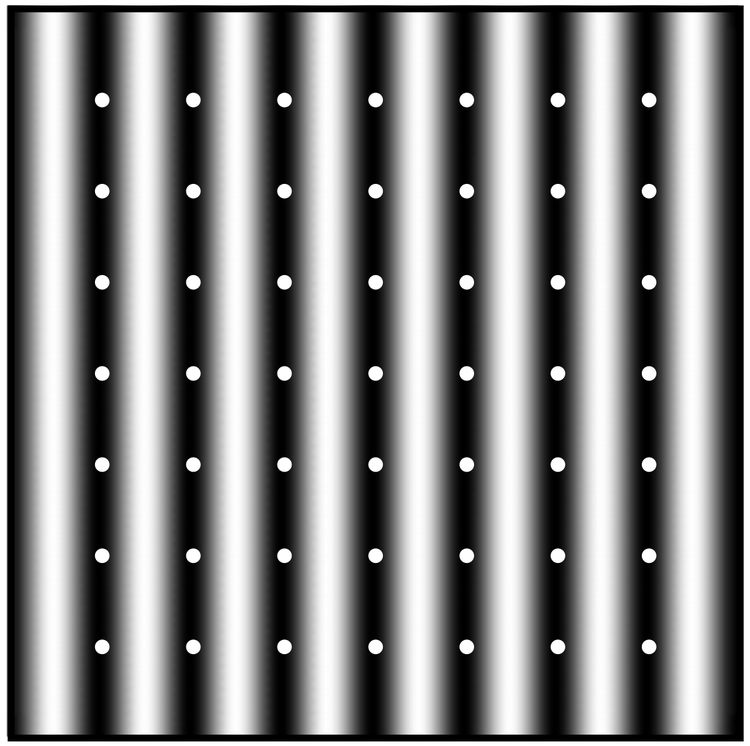


There's more to life than discrepancy

- 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

- Can observe that monkey retina exhibits blue noise pattern [Yellott 1983]

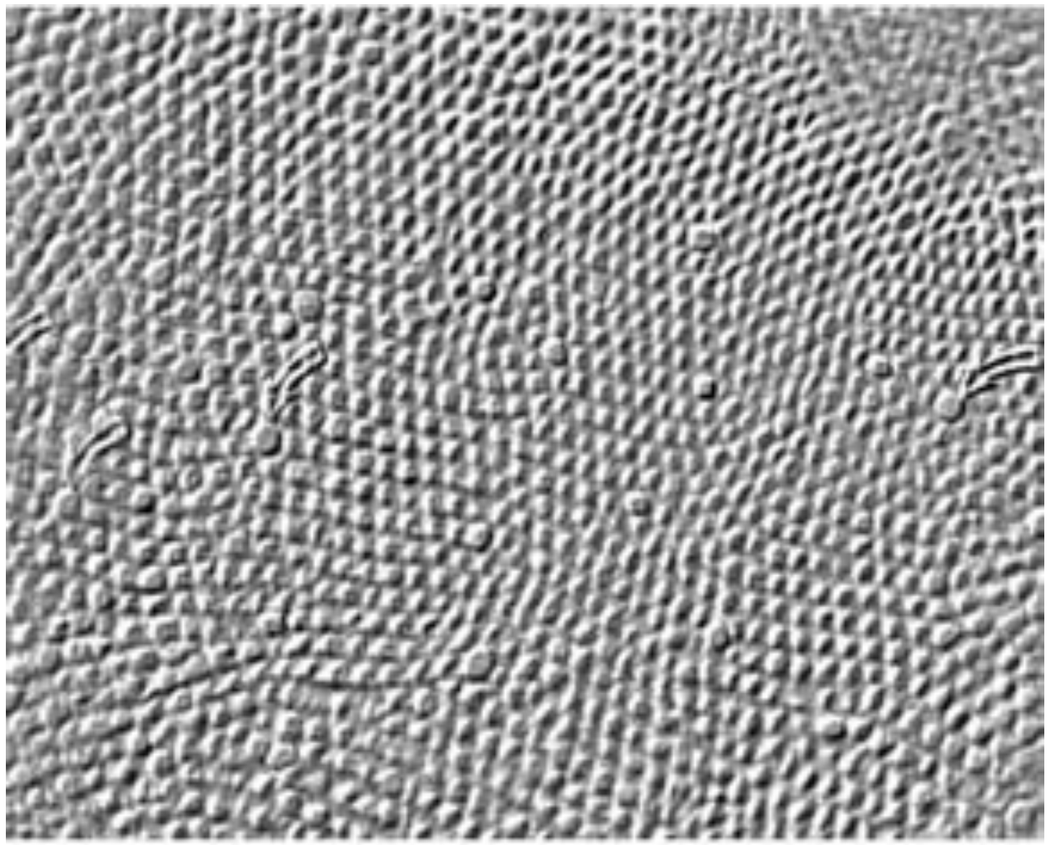
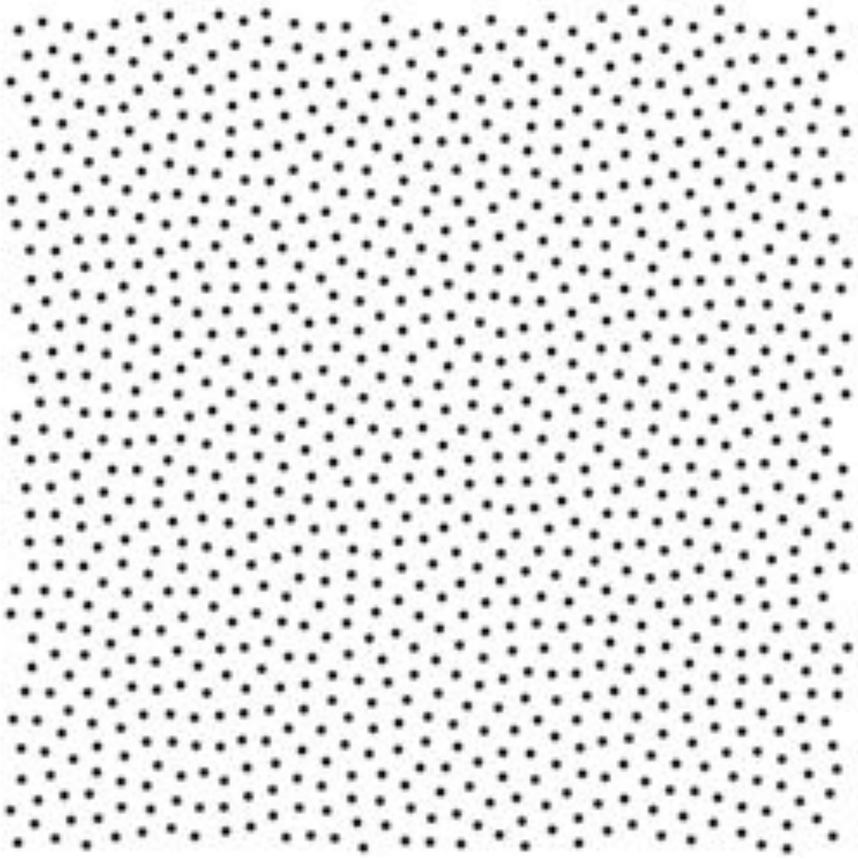


Fig. 13. Tangential section through the human fovea. Larger cones (arrows) are blue cones. From Ahnelt et al. 1987.

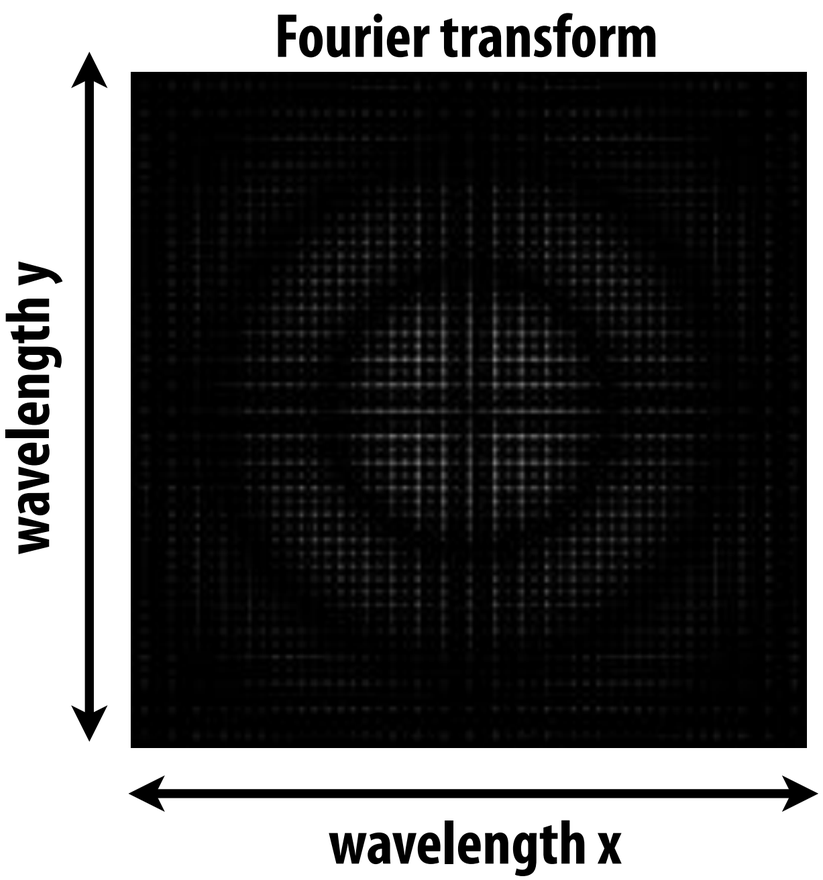


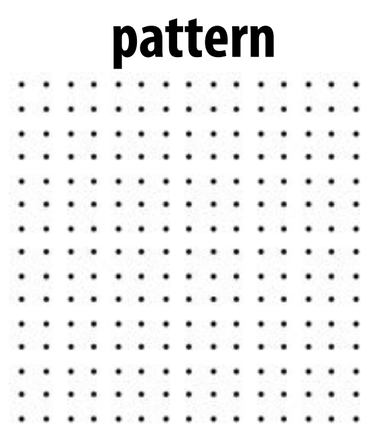
"blue noise"

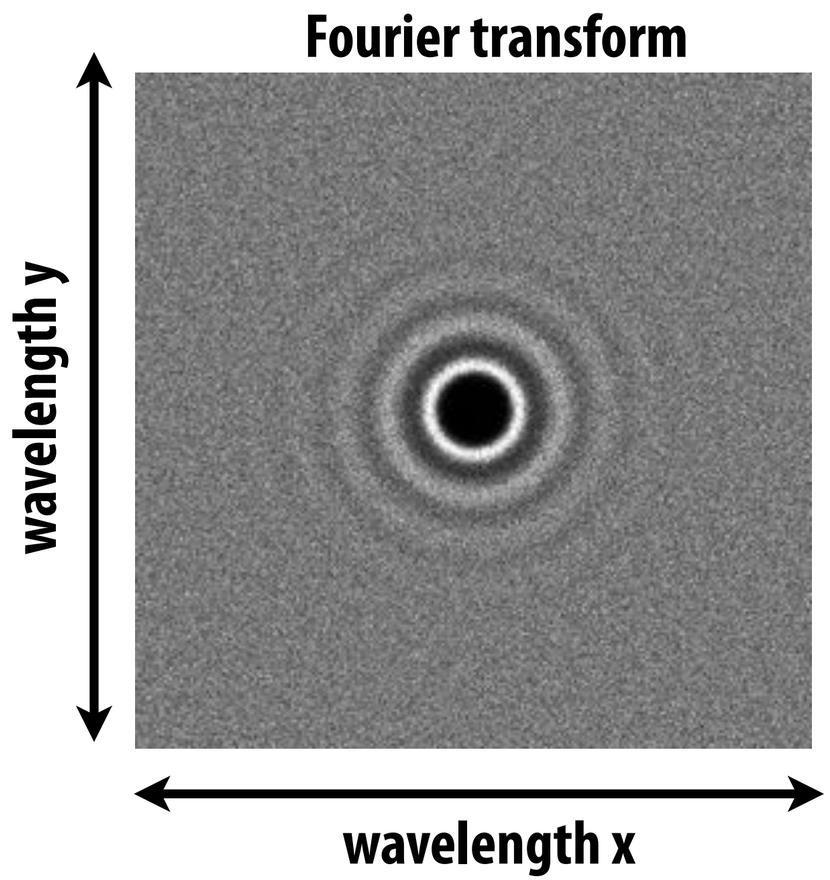
- No obvious preferred directions (anisotropic)
- What about frequencies?

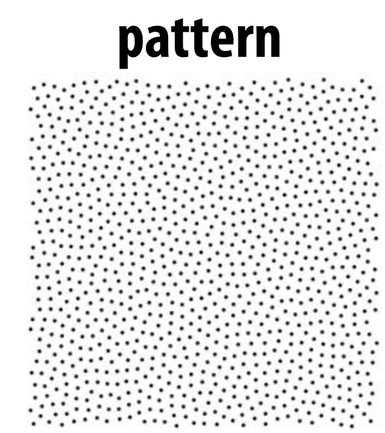
Blue Noise - Fourier Transform

- Can analyze quality of a sample pattern in Fourier domain



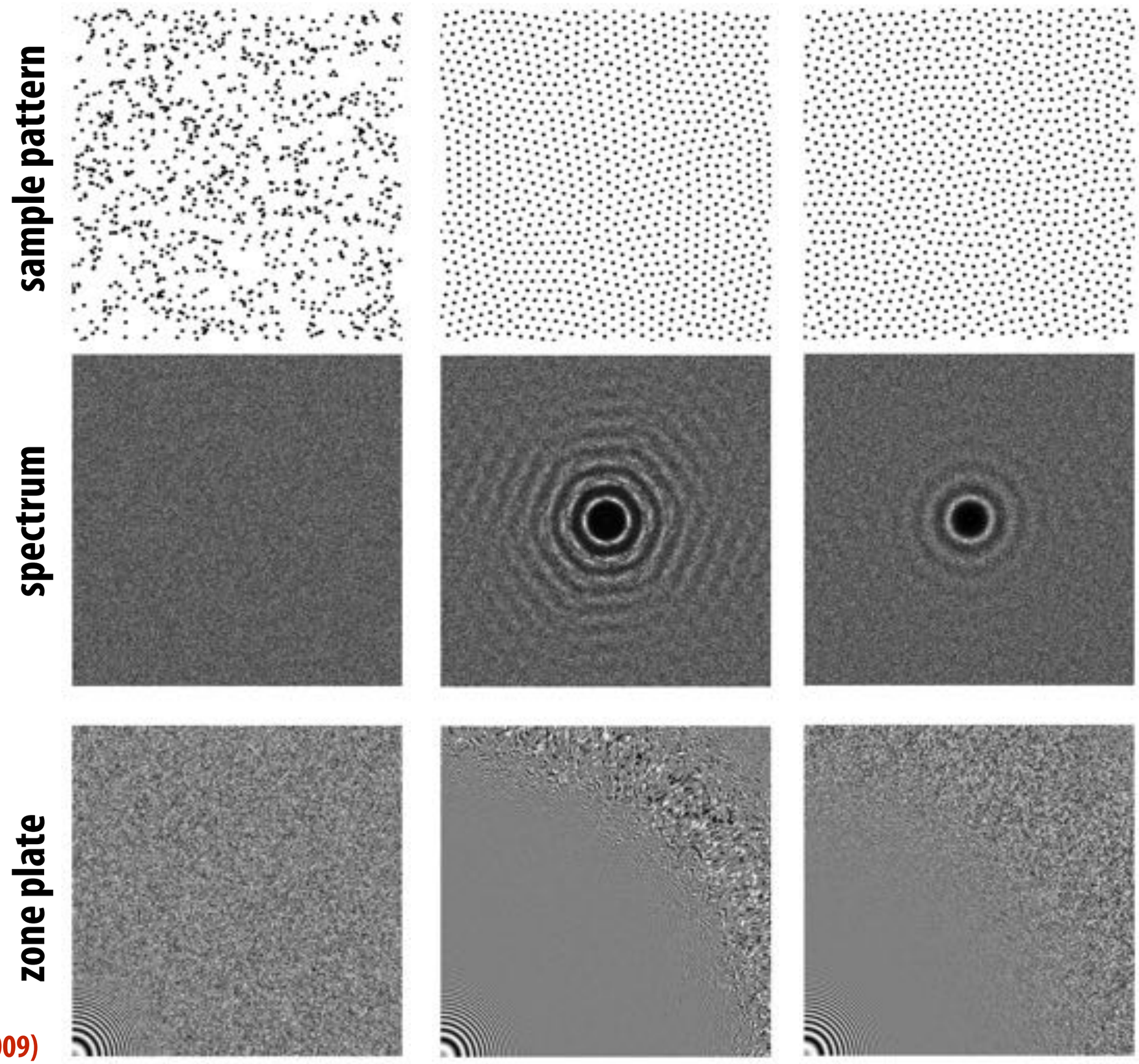






- Regular pattern has "spikes" at regular intervals
- Blue noise is spread evenly over all frequencies in all directions
- bright center "ring" corresponds to sample spacing

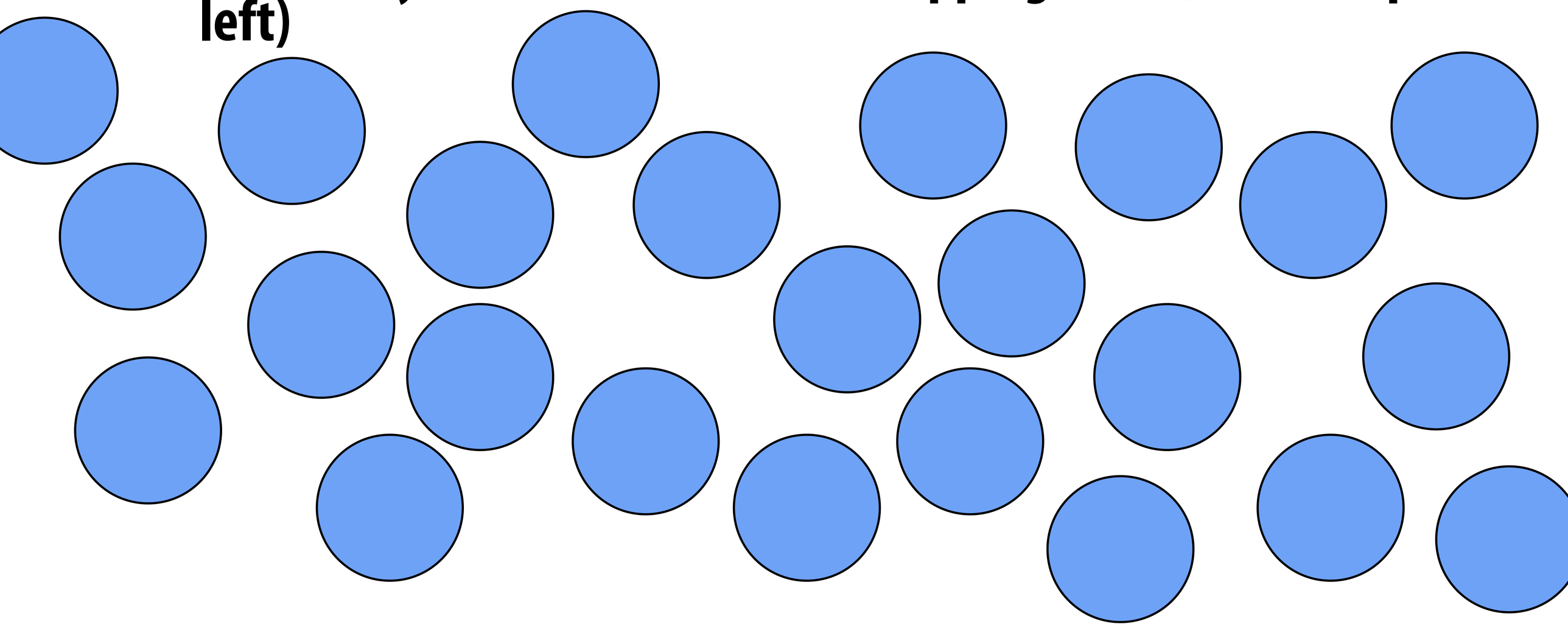
Spectrum affects reconstruction quality



(from Balzer et al 2009)

Poisson Disk Sampling

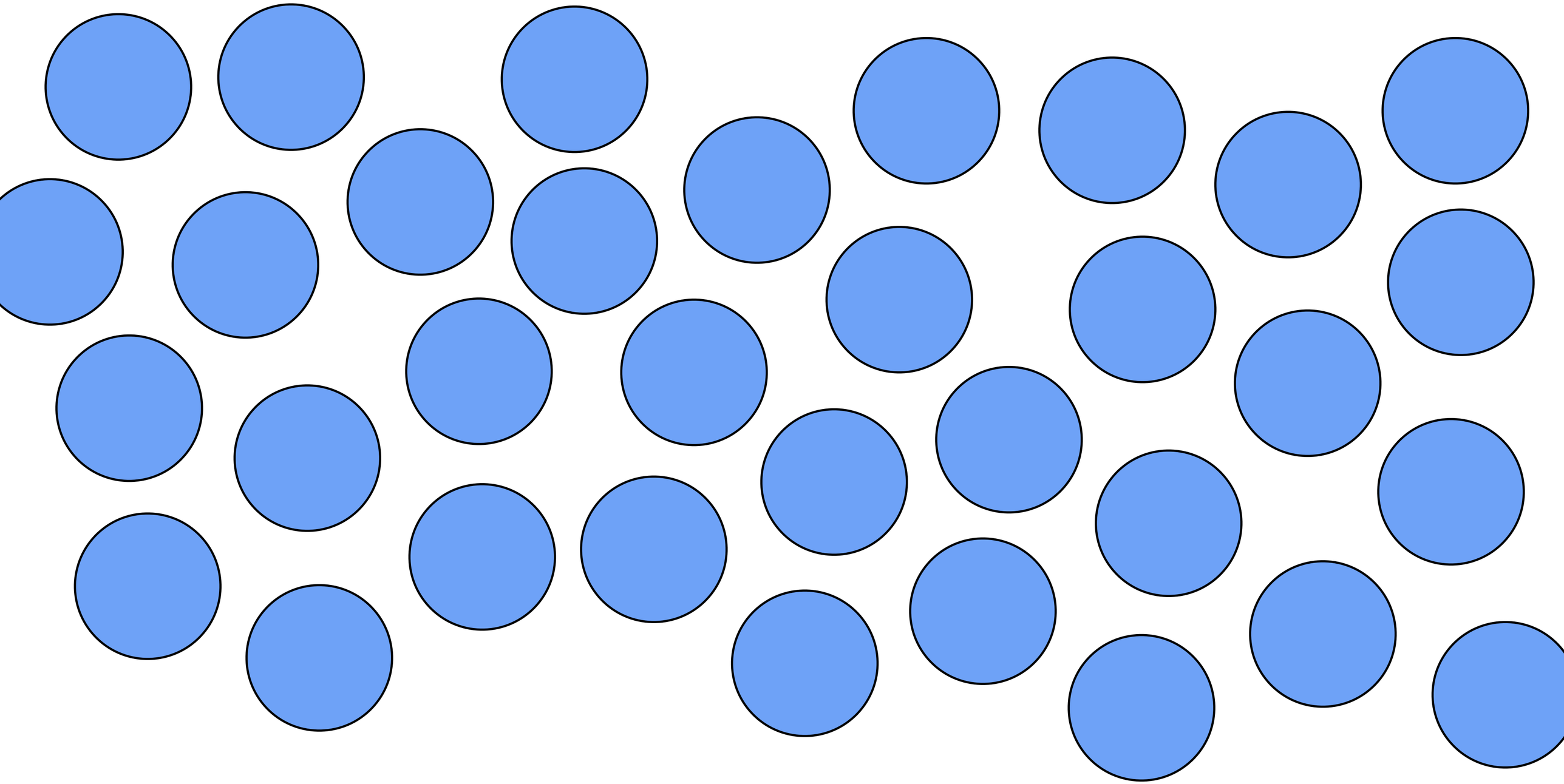
- How do you generate a "nice" sample?
- One of the earliest algorithms: Poisson disk sampling
- Iteratively add random non-overlapping disks (until no space left)



Decent spectral quality, but we can do better.

Lloyd Relaxation

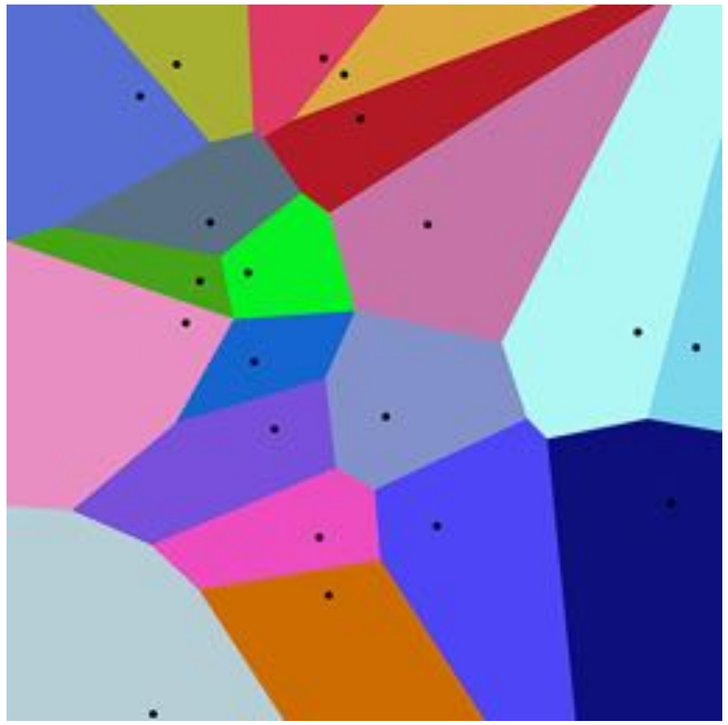
- Iteratively move each disk to the center of its neighbors



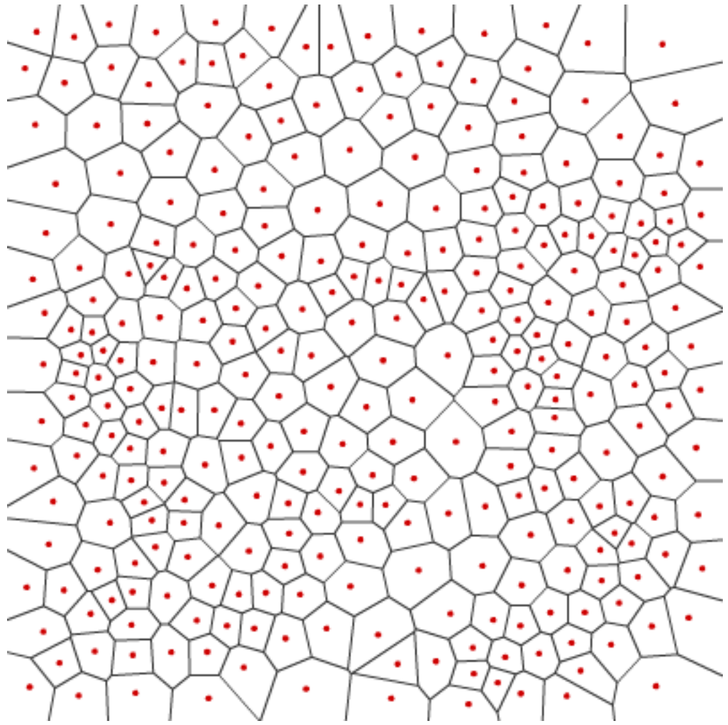
Better spectral quality, slow to converge. Can do better yet...

Voronoi-Based Methods

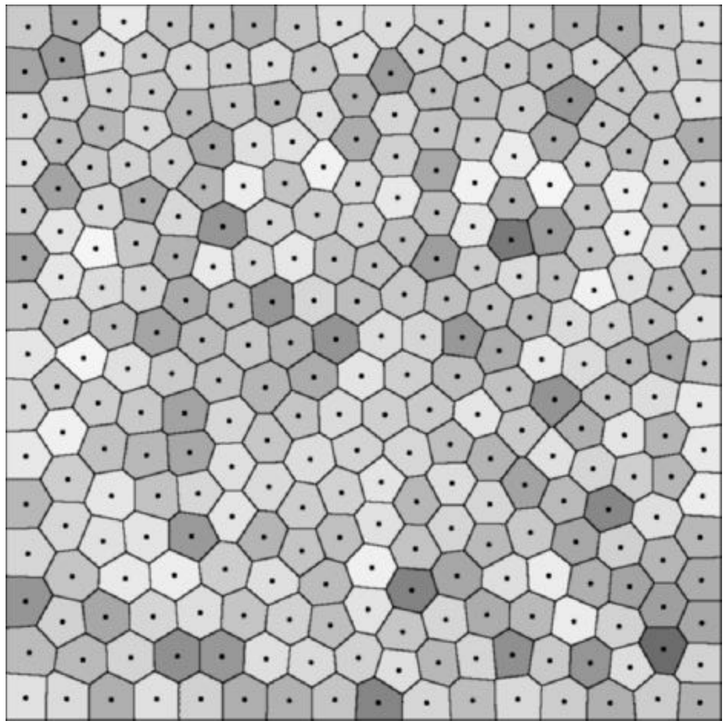
- Natural evolution of Lloyd
- Associate each sample with set of closest points (Voronoi cell)
- Optimize qualities of this Voronoi diagram
- E.g., sample is at cell's center of mass, cells have same area, etc.



Voronoi



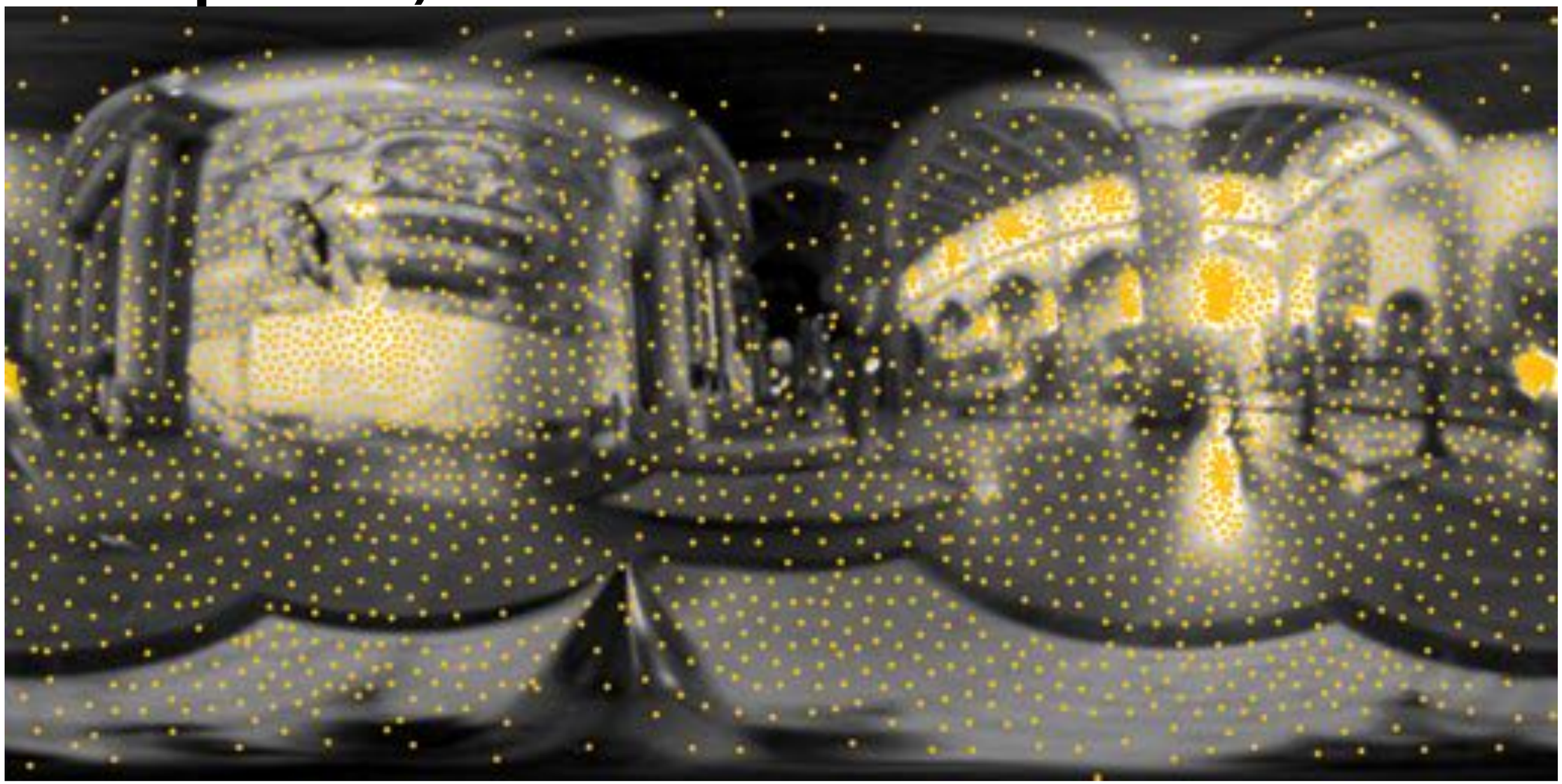
centroidal



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

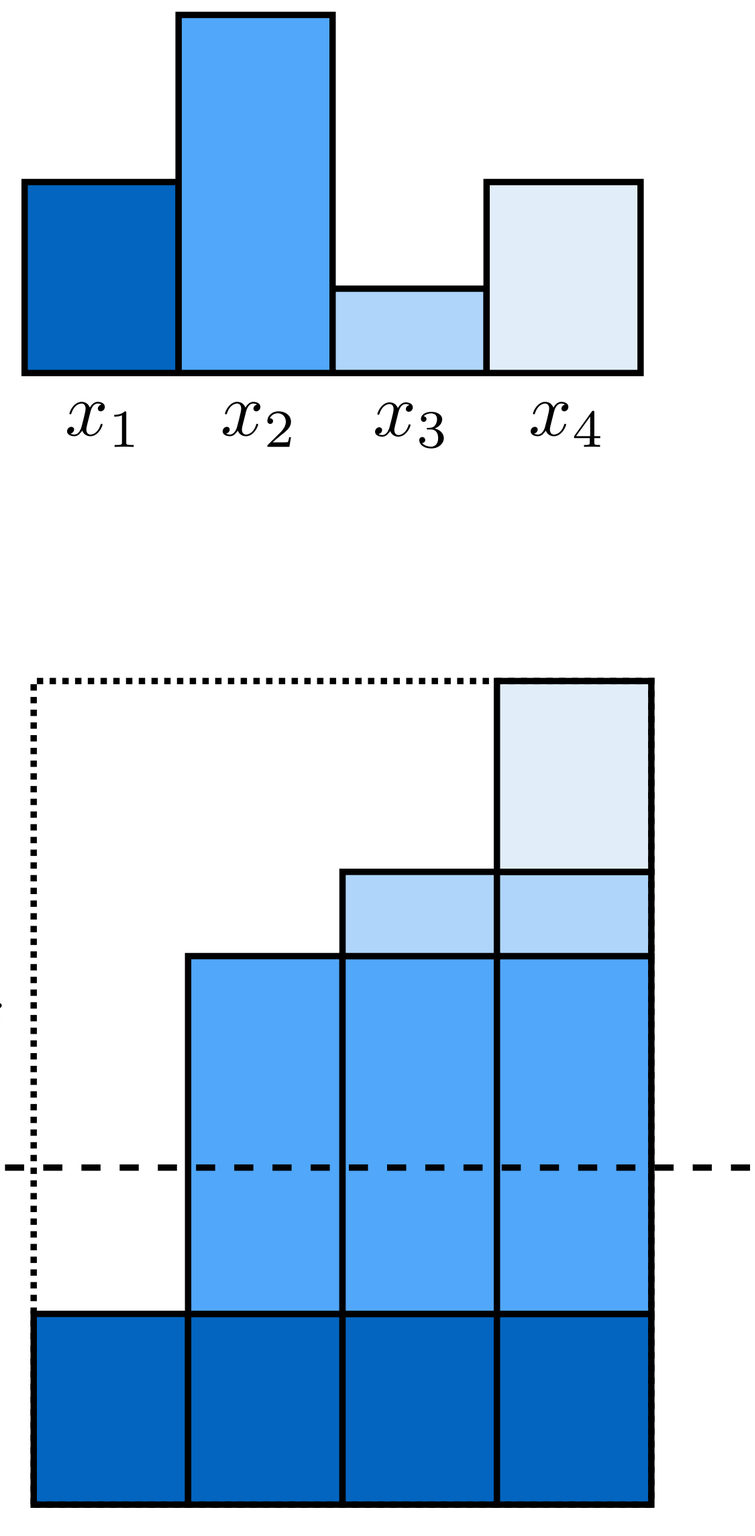
To randomly select an event, select x_i if

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

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

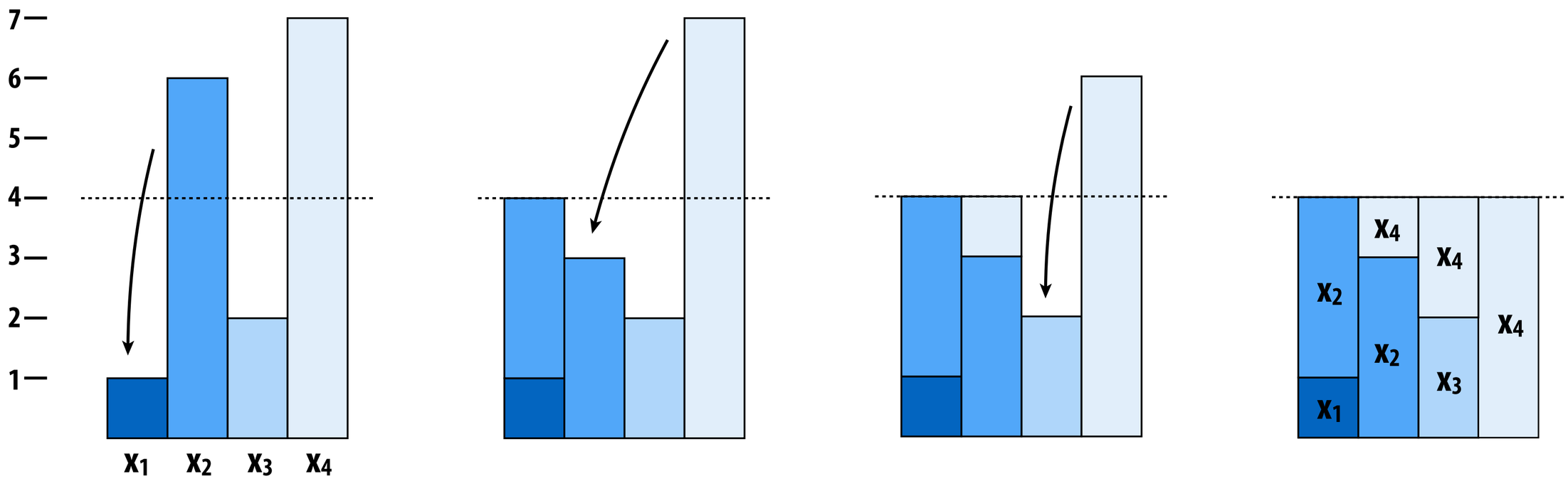
e.g., # of pixels in an environment map (big!)

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)):



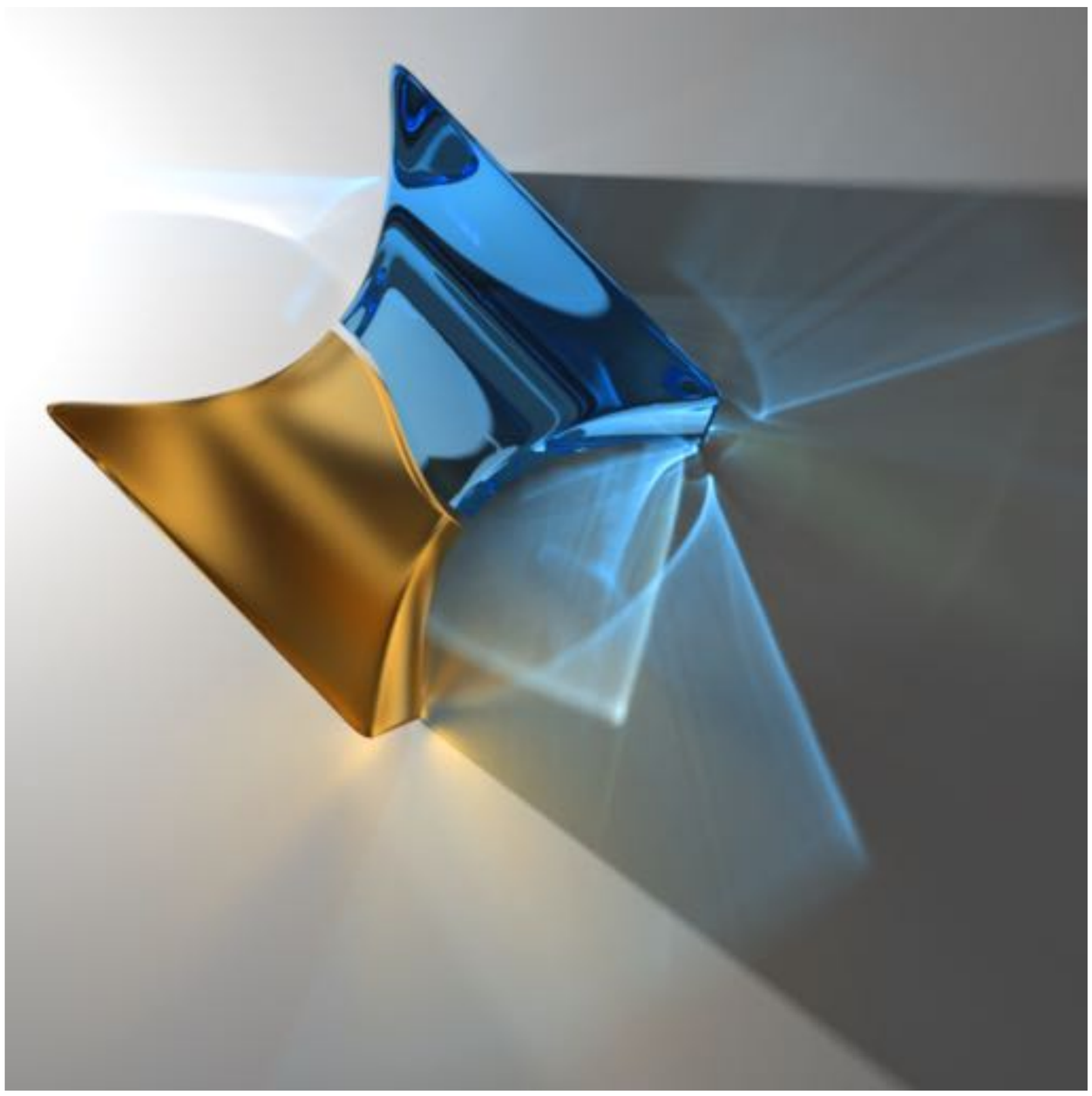
- Table just stores two identities & ratio of heights per column
- To sample:
  - pick uniform # between 1 and n
  - biased coin flip to pick one of the two identities in nth column

Ok, great!
Now we've wrapped up sampling
and we've reached the end of our discussion
of Monte Carlo path tracing

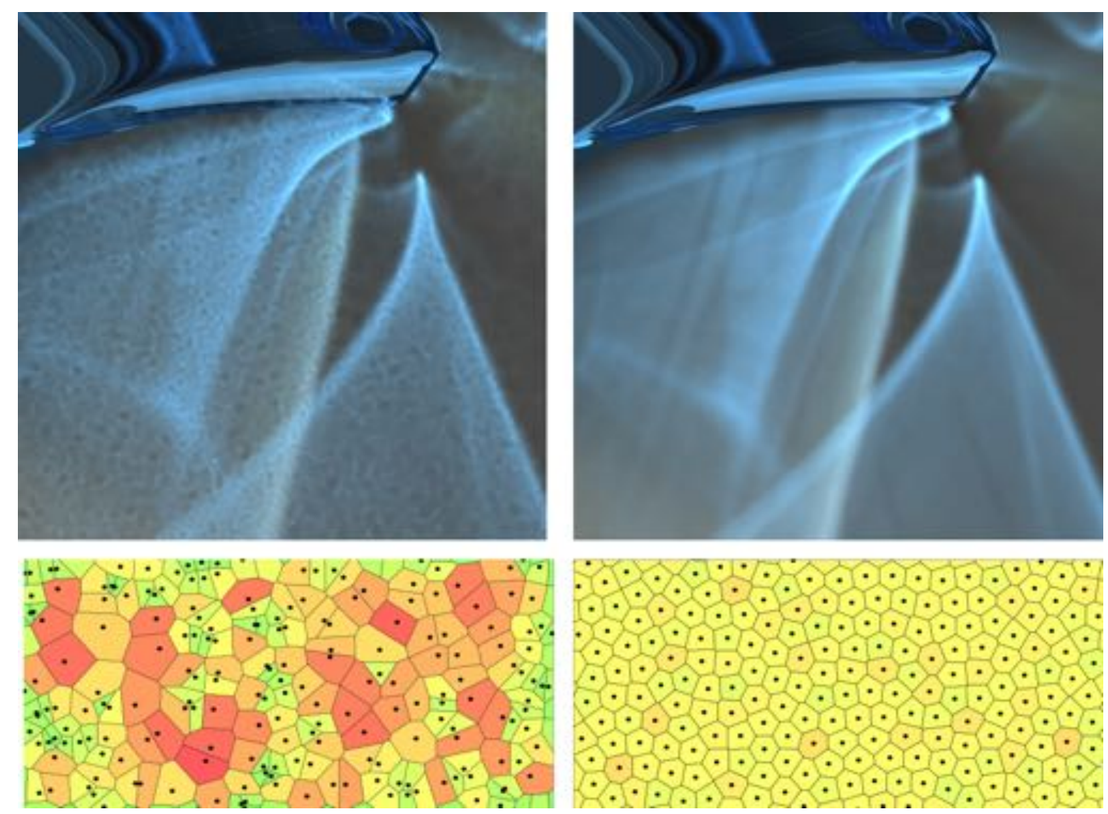
What other techniques are there?

Photon Mapping

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



Voronoi diagrams can be used to improve photon distribution



(from Spencer & Jones 2013)

Photon mapping

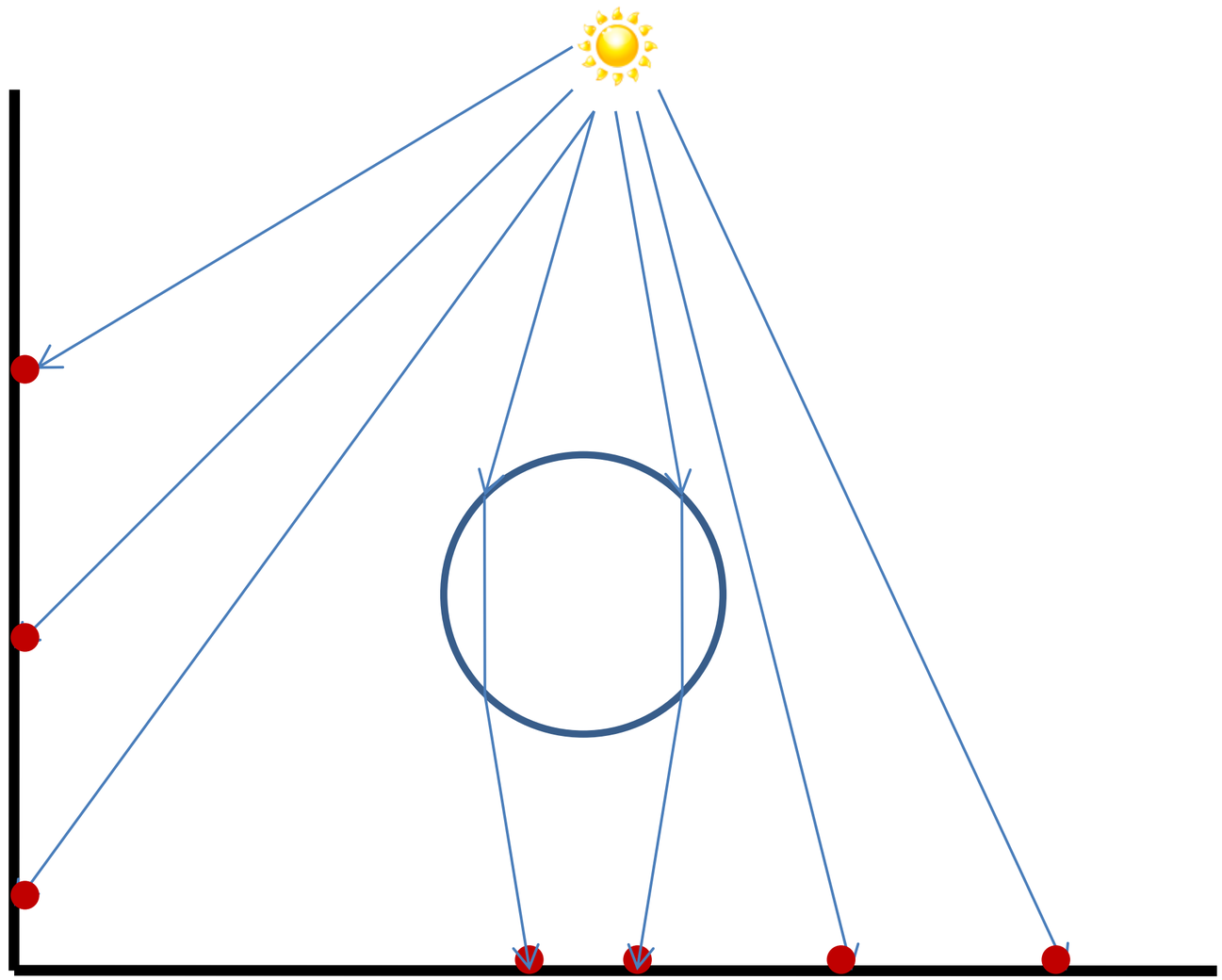
A two-pass method

Pass 1: Build the photon map (photon tracing)

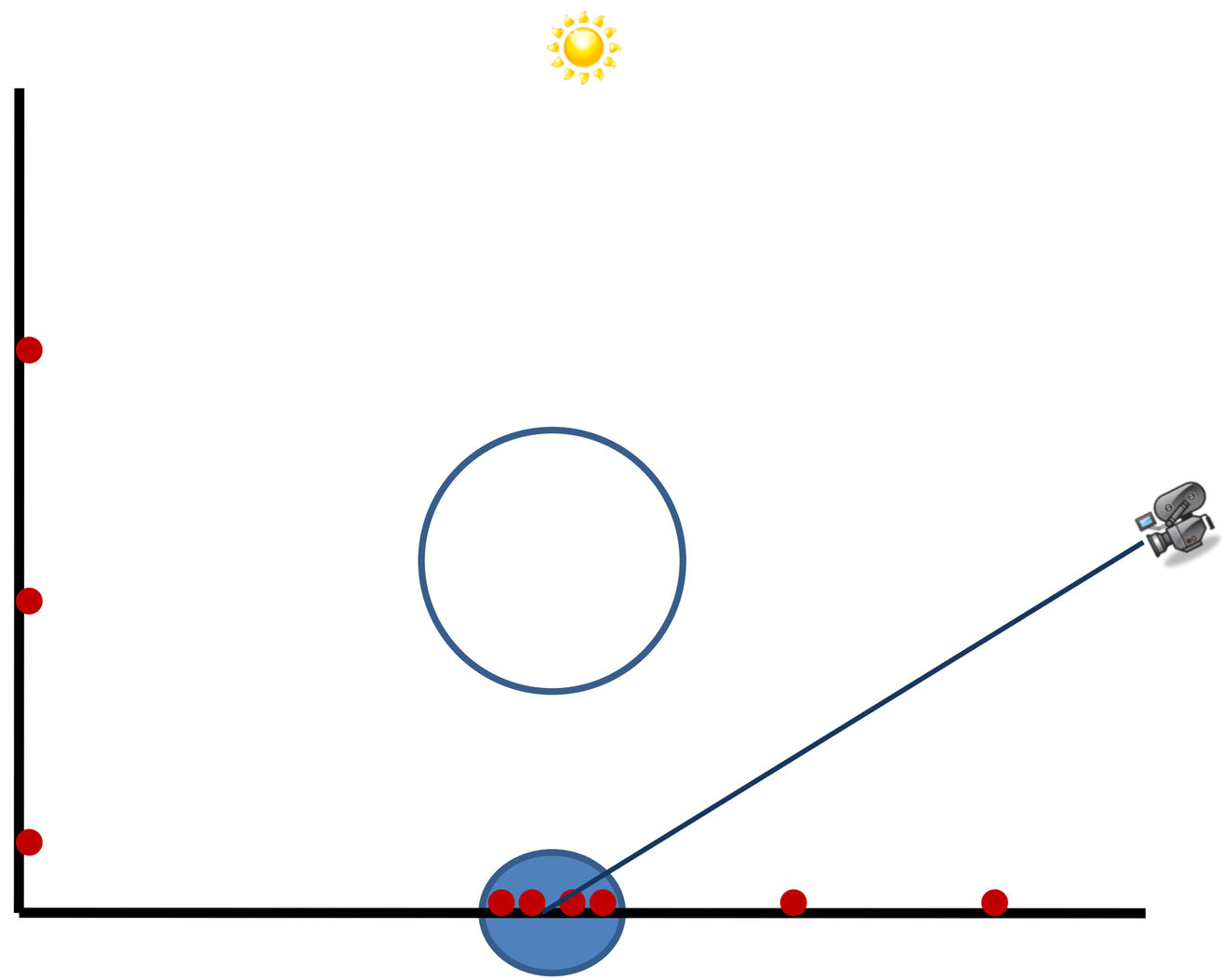
Pass 2: Render the image using the photon map

Thanks to Henrik Wann Jensen, UCSD

Building the photon map: Photon tracing



Rendering using the photon map



What is a photon?

- Flux (power) - not radiance!
- Collection of physical photons
  - A fraction of the light source power
  - Several wavelengths combined into one entity

Photon emission

Given Φ Watt lightbulb.
Emit N photons.
Each photon has the power $\frac{\Phi}{N}$ Watt.

- Photon power depends on the number of emitted photons. Not on the number of photons in the photon map.

Diffuse point light

Generate random direction
Emit photon in that direction

```
// Find random direction
do {
  x = 2.0*random()-1.0;
  y = 2.0*random()-1.0;
  z = 2.0*random()-1.0;
} while ( (x*x + y*y + z*z) > 1.0 );
```

Example: Diffuse square light

- Generate random position p on square
- Generate diffuse direction d
- Emit photon from p in direction d

```
// Generate diffuse direction
u = random();
v = 2*π*random();
d = vector( cos(v)√u, sin(v)√u, √(1 − u) );
```

Surface interactions

The photon is

- Stored (at diffuse surfaces) and
- Absorbed (A) or
- Reflected (R) or
- Transmitted (T)

$$A + R + T = 1.0$$

Storing the photon

```
struct photon {
  float x,y,z;        // position
  char p[4];          // power packed as 4 bytes
  char phi,theta;     // incident direction
  short flag;         // flag used for kd-tree
}
```

Memory overhead: 20 bytes/photon.

Photon scattering

The simple way:

Given incoming photon with power Φ_p

Reflect photon with the power $R * \Phi_p$

Transmit photon with the power $T * \Phi_p$

- Risk: Too many low-powered photons - wasteful!
- When do we stop (systematic bias)?
- Photons with similar power is a good thing.

Russian Roulette

- Statistical technique
- Known from Monte Carlo particle physics
- Introduced to graphics by Arvo and Kirk in 1990

Terminate un-important photons and still get the correct result.

Russian Roulette Example

Surface reflectance: $R = 0.5$
Incoming photon: $\Phi_p = 2$ W

```
r = random();
if ( r < 0.5 )
  reflect photon with power 2 W
else
  photon is absorbed
```

Reflect 100 photons with power 2 Watt instead of 200 photons with power 1 Watt.

Russian Roulette Example 2

Surface reflectance: R = 0.2
Surface transmittance: T = 0.3
Incoming photon: Φ_p = 2 W

```
r = random();
if ( r < 0.2 )
  reflect photon with power 2 W
else if ( r < 0.5 )
  transmit photon with power 2 W
else
  photon is absorbed
```

Sampling a BRDF

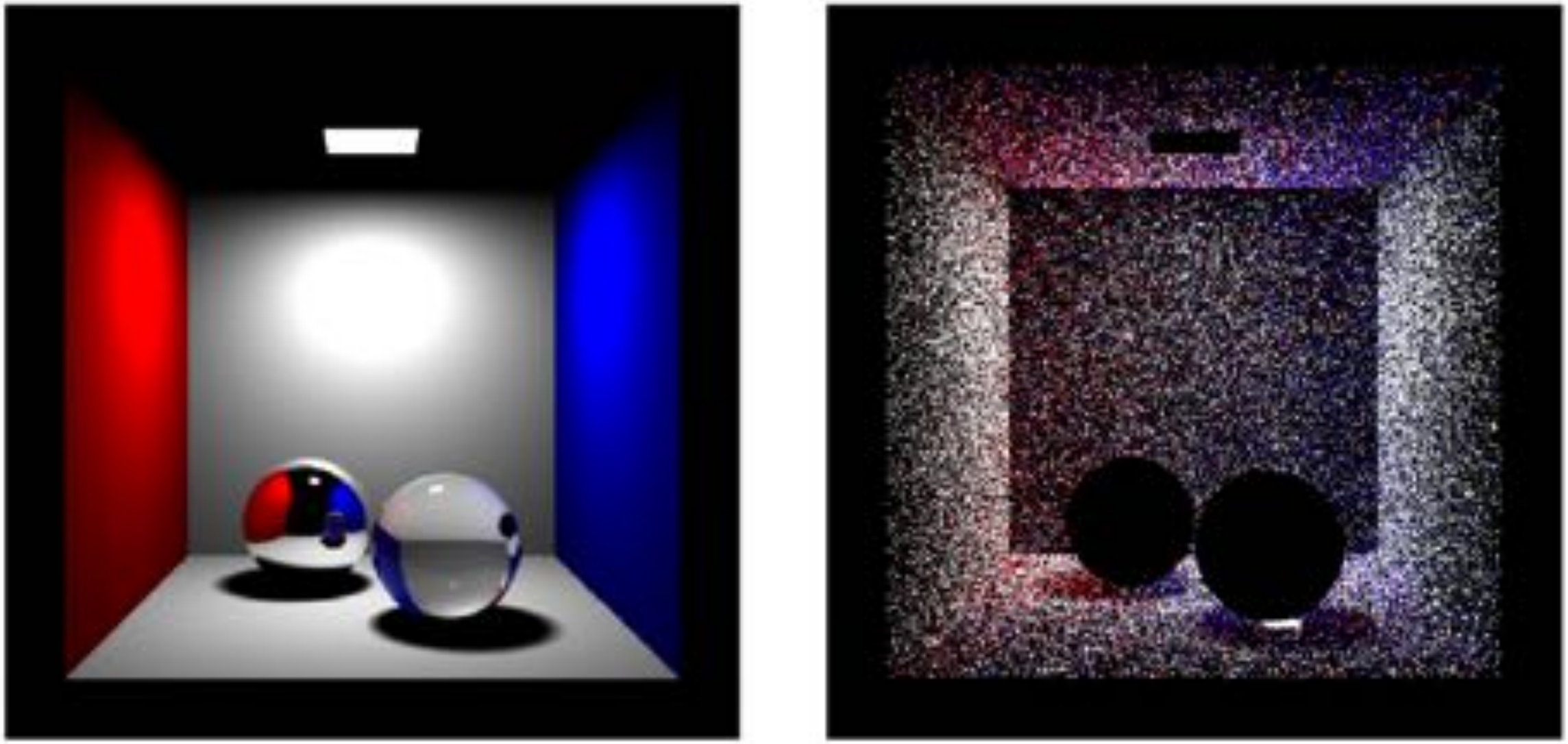
$$f_r(x, \vec{\omega}_i, \vec{\omega}_o) = w_1 \cdot f_{r,d} + w_2 \cdot f_{r,s}$$

```
r = random()·(w1 + w2);
if ( r < w1 )
  reflect diffuse photon
else
  reflect specular
```

Photon tracing

Overview:

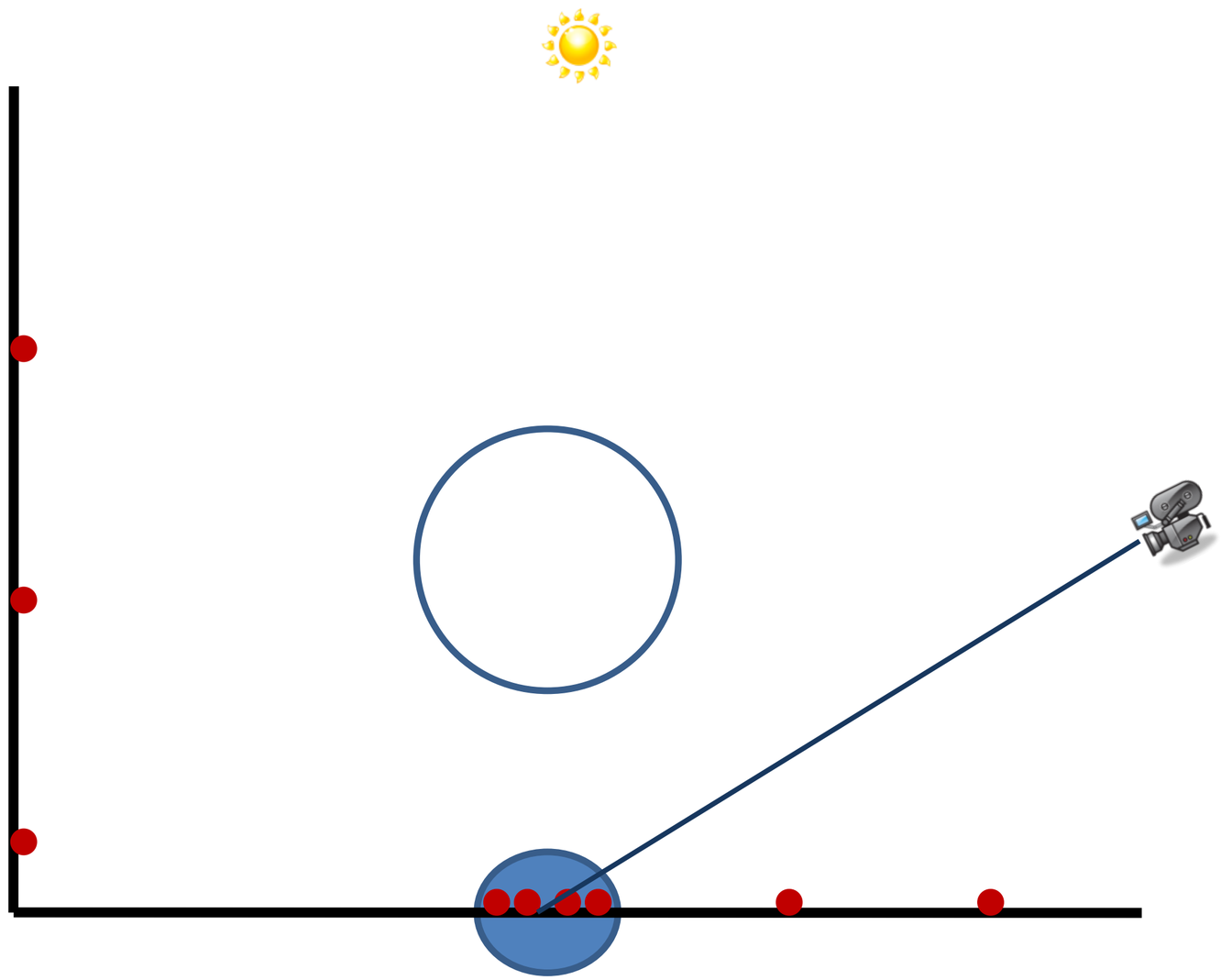
```
While (we want more photons) {
  Emit a photon
  while (photon hits a surface) {
   Store photon
   Use Russian Roulette to scatter photon
  }
}
Build balanced kd-tree
```



(a) (b)

Figure 4.4: "Cornell box" with glass and chrome spheres: (a) ray traced image (direct illumination and specular reflection and transmission), (b) the photons in the corresponding photon map.

Rendering using the photon map



Rendering

We want a Radiance value, L, per pixel.

The photon map stores flux/power.

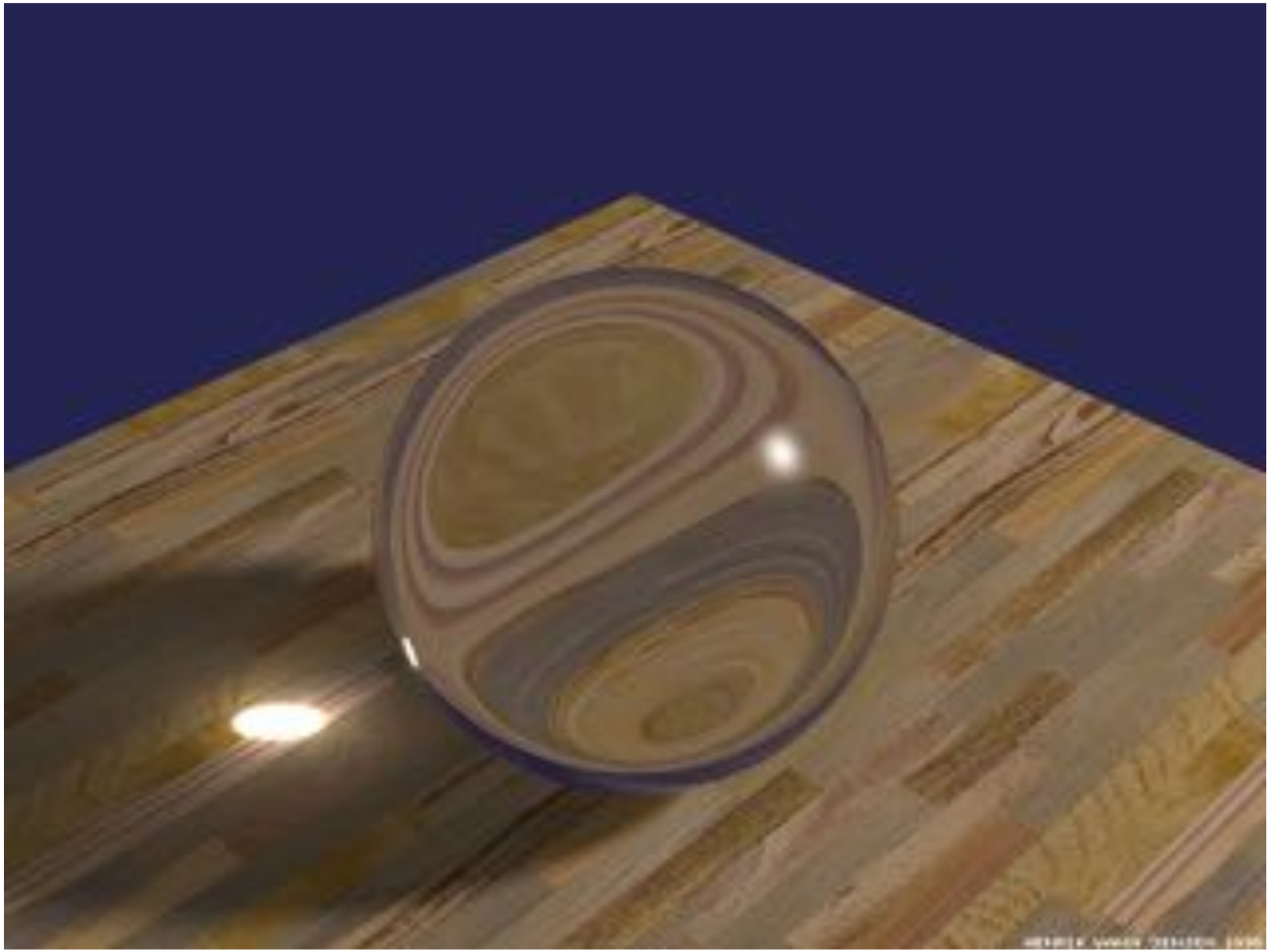
Radiance is the differential flux per differential solid angle per differential cross-sectional area:

$$L(x, \vec{\omega}) = \frac{d\Phi^2(x, \vec{\omega})}{d\omega \, \cos\theta \, dA}$$

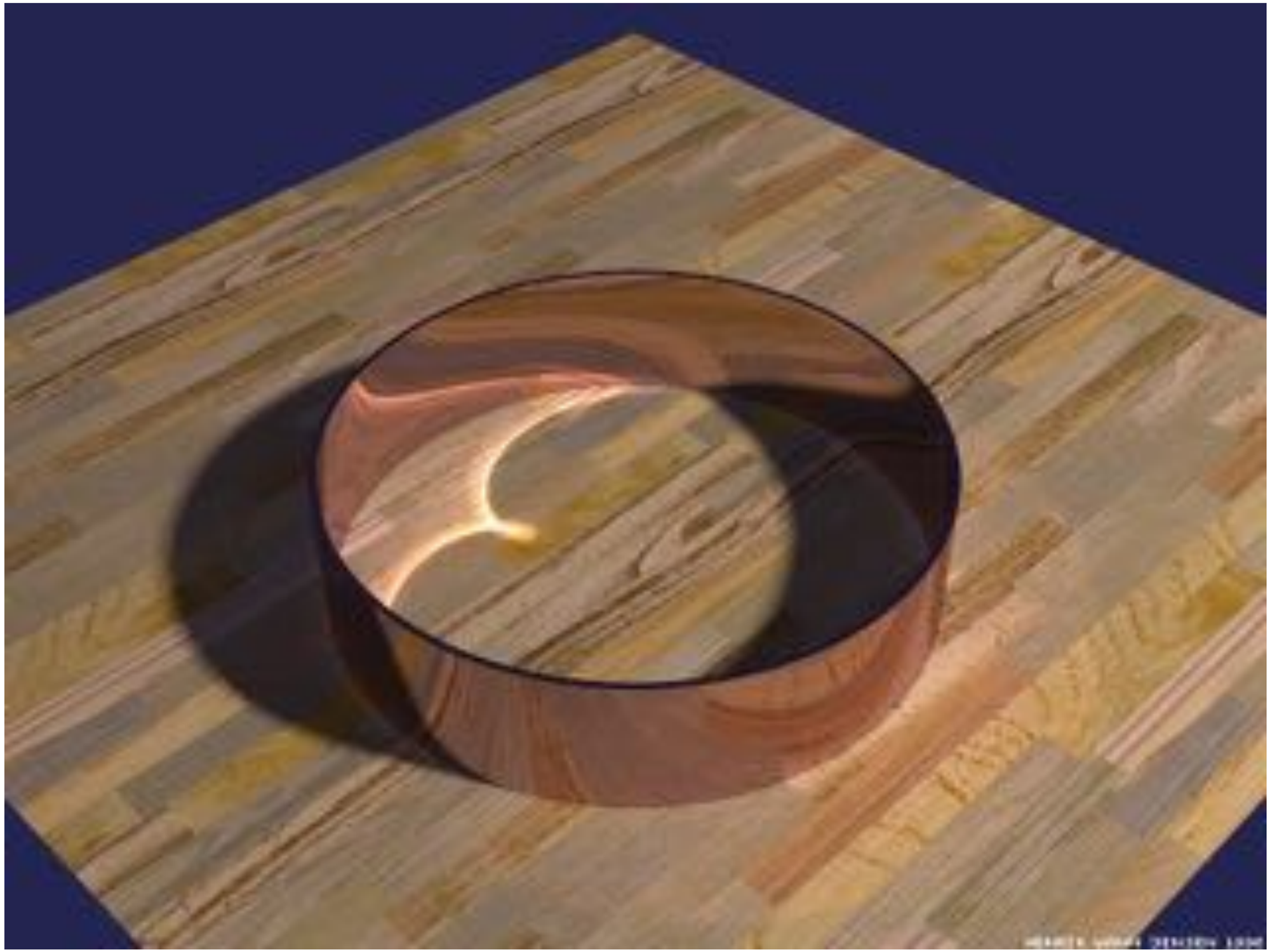
Radiance estimate

$$
\begin{aligned}
L(x,\vec{\omega}) &= \int_{\Omega} f_r(x,\vec{\omega}',\vec{\omega}) L'(x,\vec{\omega}') \cos\theta' \, d\omega' \\
&= \int_{\Omega} f_r(x,\vec{\omega}',\vec{\omega}) \frac{d\Phi'^2(x,\vec{\omega}')}{d\omega' \cos\theta' \, dA} \cos\theta' d\omega' \\
&= \int_{\Omega} f_r(x,\vec{\omega}',\vec{\omega}) \frac{d\Phi'^2(x,\vec{\omega}')}{dA} \\
&\approx \sum_{p=1}^{n} f_r(x,\vec{\omega}'_p,\vec{\omega}) \frac{\Delta\Phi_p(x,\vec{\omega}'_p)}{\Delta A}
\end{aligned}
$$

Radiance estimate



30000 photons / 50 photons in radiance estimate



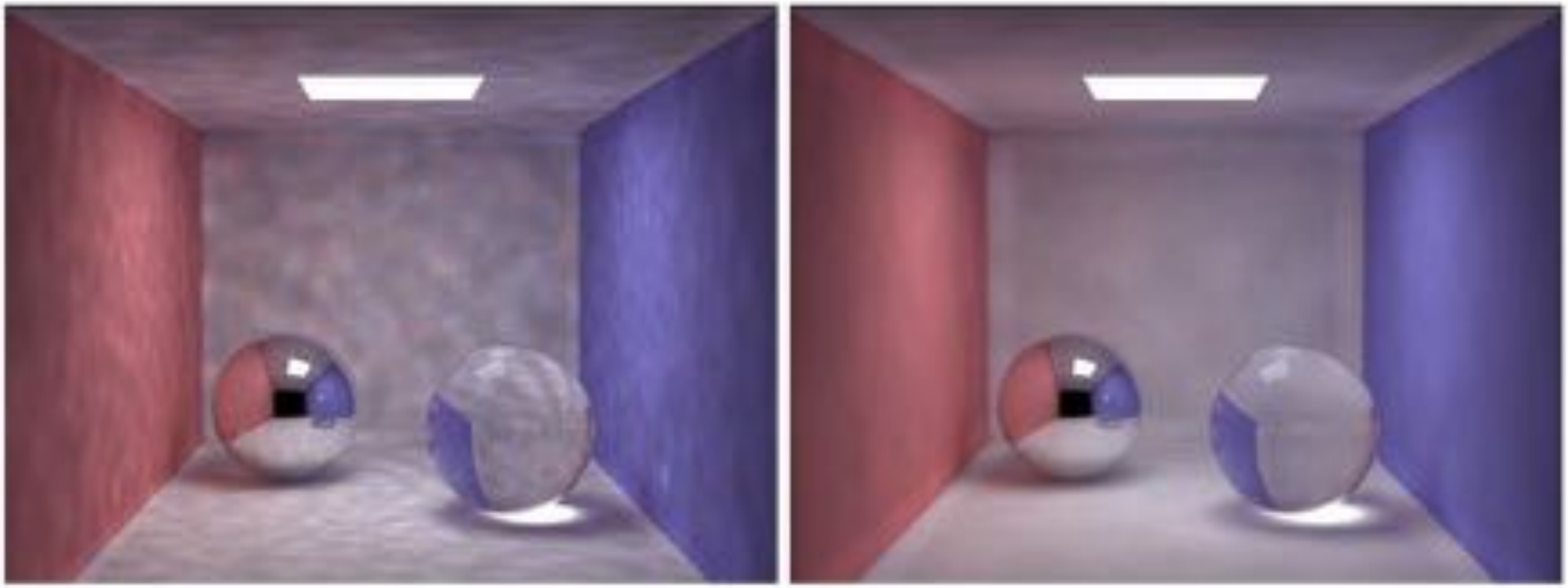


Figure 4.20: Global photon map radiance estimates visualized directly using 100 photons (left) and 500 photons (right) in the radiance estimate.

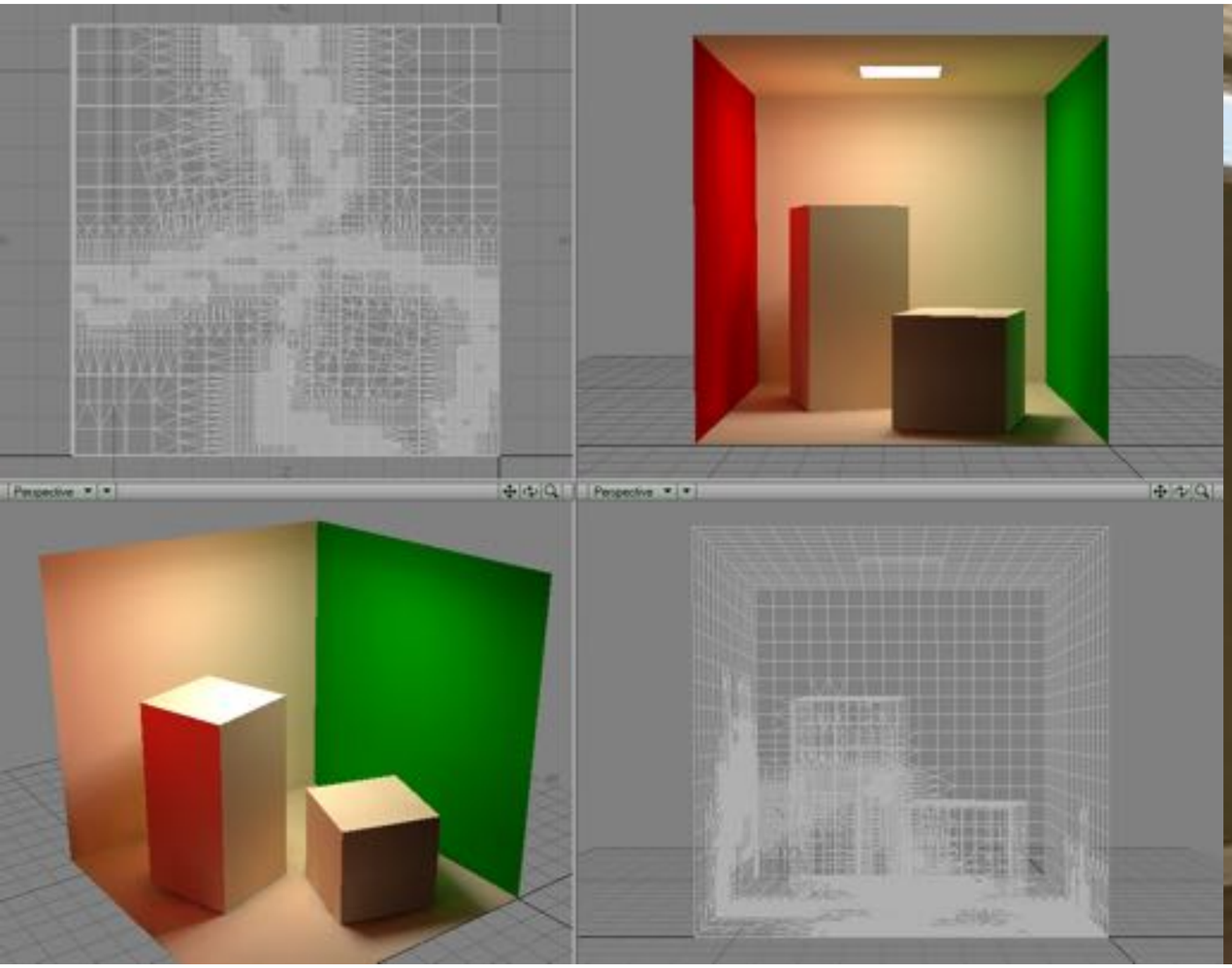
SIGGRAPH COURSE NOTES

The SIGGRAPH course notes I showed in class can be found here:
http://www.seanet.com/~myandper/abstract/sig07c08.htm

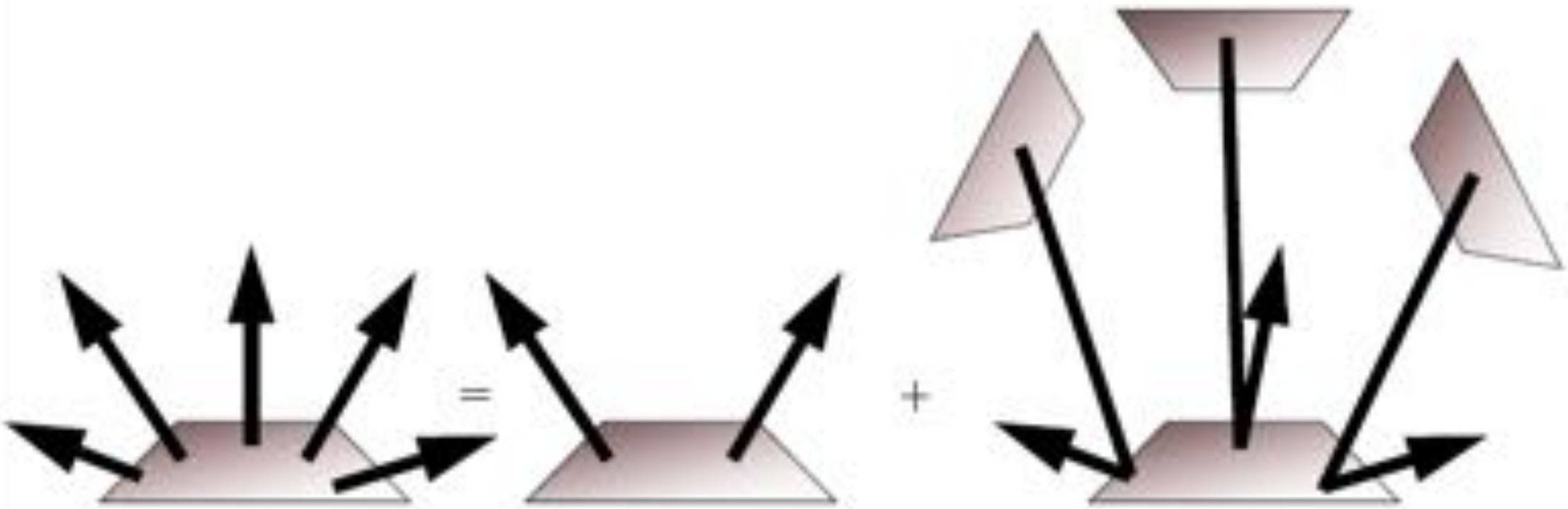
This is not the most recent SIGGRAPH course, but it may be the most useful for understanding the basic issues.

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



Conservation of Energy



Emitted power = self-emitted power + received & reflected power

© Kavita Bala, Computer Science, Cornell University

Diffuse Interreflections - Radiosity

- Consider lambertian surfaces and sources.
- Radiance independent of viewing direction.
- Consider total power leaving per unit area of a surface.
- Can simulate soft shadows and color bleeding from diffuse surfaces.
- Used abundantly in heat transfer literature

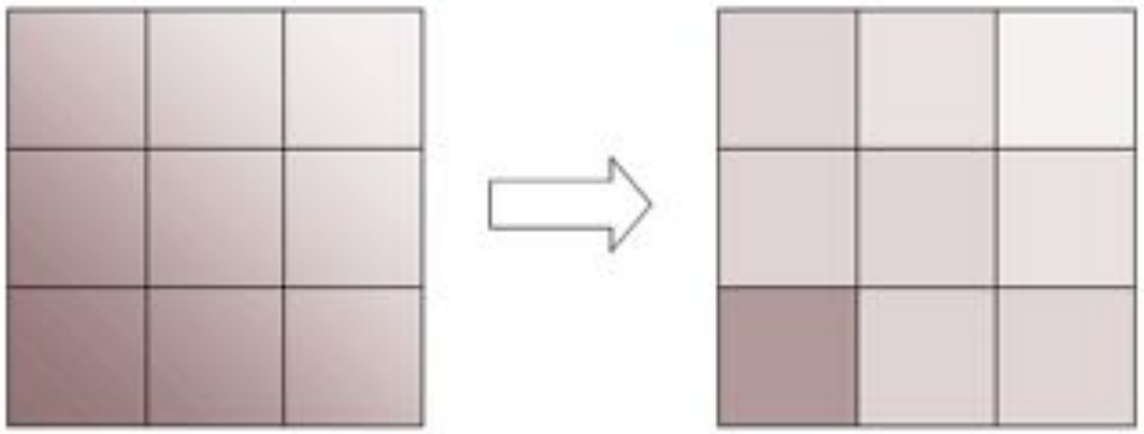
Irradiance, Radiosity

- Irradiance E is the power **received** per unit surface area
  - Units: W/m^2

- Radiosity
  - Power per unit area **leaving** the surface

Planar piecewise constancy assumption

- Subdivide scene into small “uniform” polygons



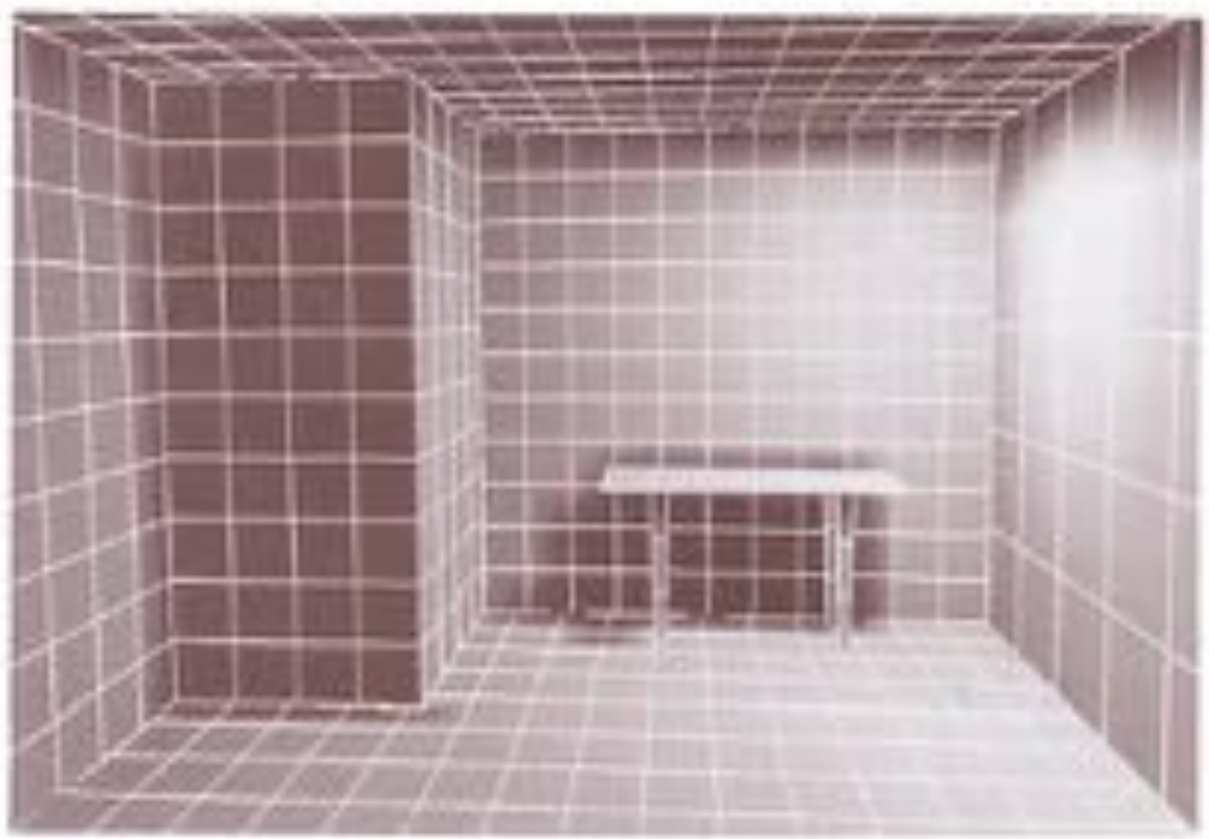


Table in room sequence from Cohen and Wallace

Power Equation

- Power from each polygon:

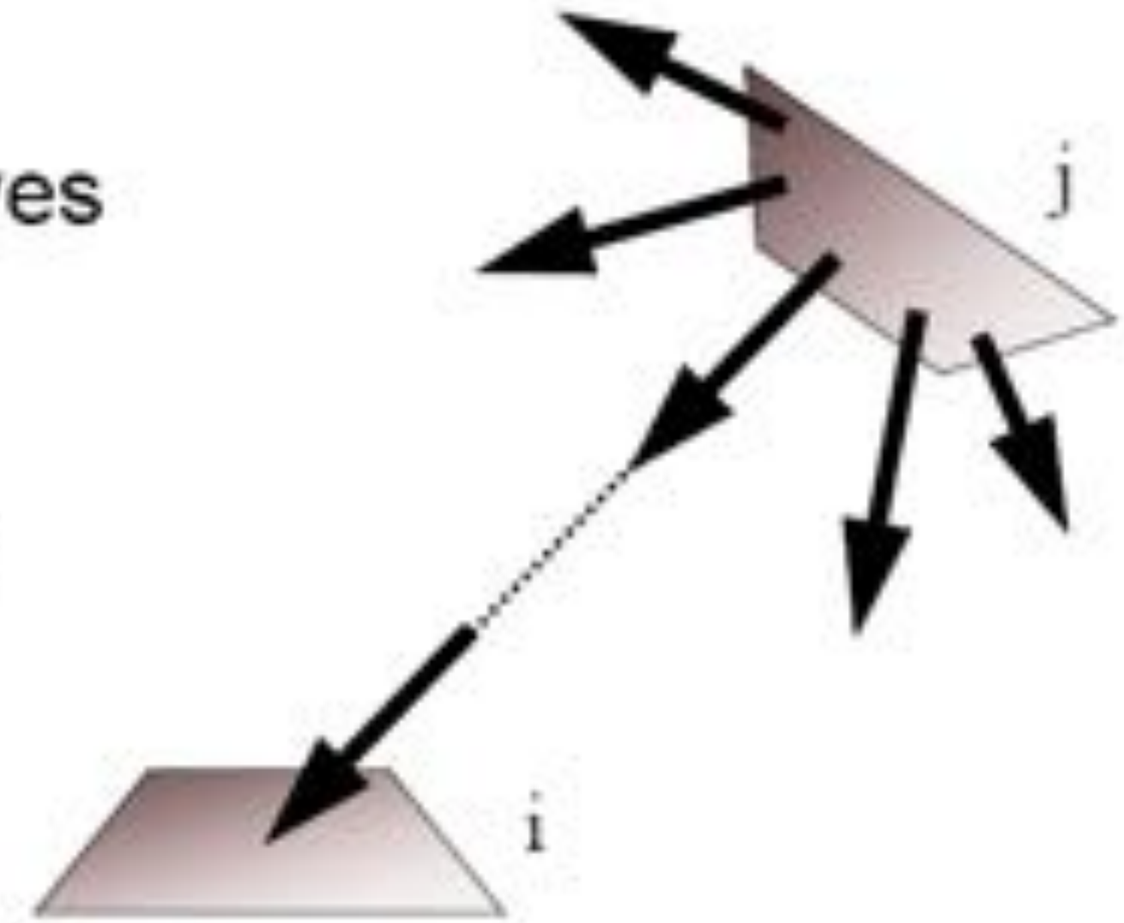
$$\forall i : \Phi_i = \Phi_{ei} + \rho_i \sum_{j=1}^{N} \Phi_j \, F(j \rightarrow i)$$

- Linear System of Equations:

- Φ_i : power of patch i (unknown)
- $\Phi_{e,i}$: emission of patch i (known)
- ρ_i : reflectivity of patch i (known)
- $F(j \rightarrow i)$: form-factor (coefficients of matrix)

Form Factor

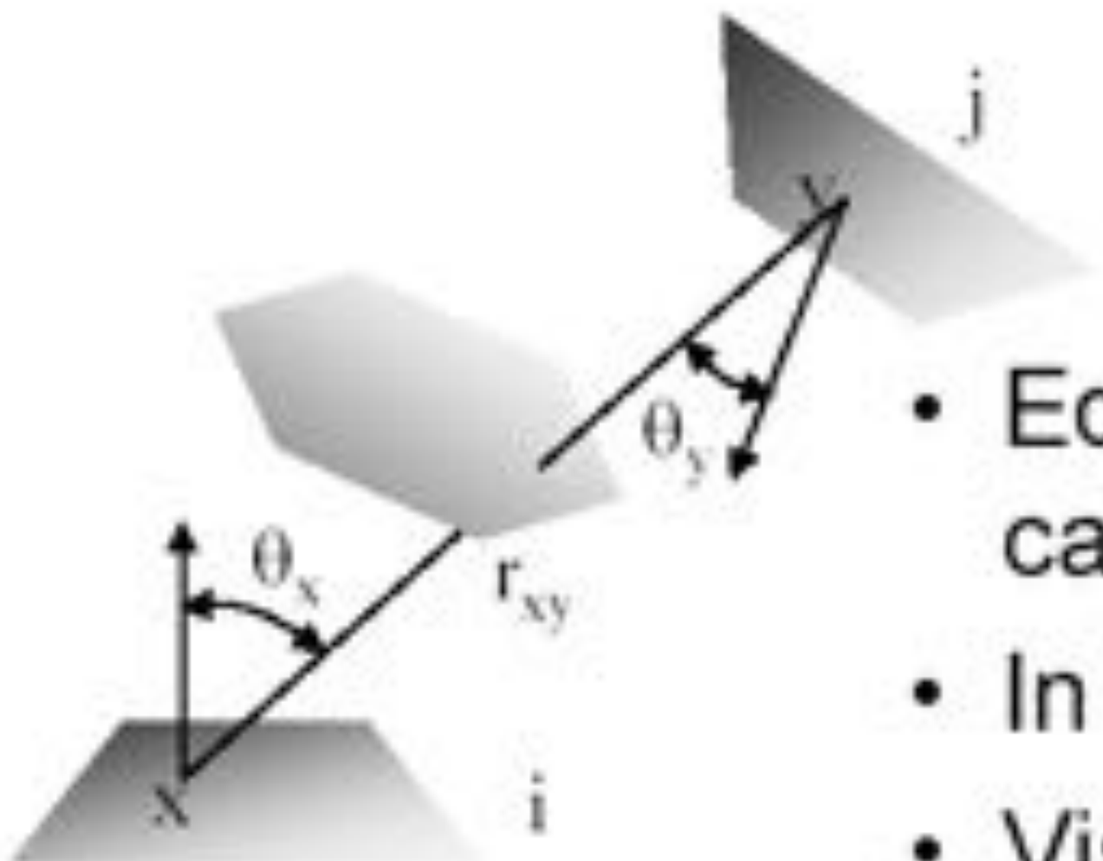
- $F_{j \to i}$ = the fraction of power emitted by j, which is received by i
- <u>Area</u>
  - if i is smaller, it receives less power
- <u>Orientation</u>
  - if i faces j, it receives more power
- <u>Distance</u>
  - if i is further away, it receives less power



© Kavita Bala, Computer Science, Cornell University

Form Factor

$$F(j \rightarrow i) = \frac{1}{A_j} \int_{A_i} \int_{A_j} \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} V(x,y)\, dA_y\, dA_x$$



- Equations for special cases (polygons)
- In general hard problem
- Visibility makes it harder

© Kavita Bala, Computer Science, Cornell University

Power →Radiosity

$$\Phi_i = \Phi_{e,i} + \rho_i \sum_{j=1}^{N} \Phi_j F(j \to i)$$

Divide by A_i

$$\frac{\Phi_i}{A_i} = \frac{\Phi_{e,i}}{A_i} + \rho_i \sum_{j=1}^{N} \frac{\Phi_j F(j \to i)}{A_i}$$

$$B_i = B_{e,i} + \rho_i \sum_{j=1}^{N} \frac{\Phi_j \dfrac{F(i \to j) A_i}{A_j}}{A_i}$$

$$B_i = B_{e,i} + \rho_i \sum_{j=1}^{N} \frac{\Phi_j F(i \to j)}{A_j}$$

$$B_i = B_{e,i} + \rho_i \sum_{j=1}^{N} B_j F(i \to j)$$

© Kavita Bala, Computer Science, Cornell University

Linear System of Radiosity Equations

$\forall$patches i: $$B_i = B_{ei} + \rho_i \sum_j F_{i \to j} B_j$$

$$\begin{bmatrix} 1-\rho_1 F_{1\to 1} & -\rho_1 F_{1\to 2} & \cdots & -\rho_1 F_{1\to n} \\ -\rho_2 F_{2\to 1} & 1-\rho_2 F_{2\to 2} & \cdots & -\rho_2 F_{2\to n} \\ \cdots & \cdots & \cdots & \cdots \\ -\rho_n F_{n\to 1} & -\rho_n F_{n\to 2} & \cdots & 1-\rho_n F_{n\to n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdots \\ B_n \end{bmatrix} = \begin{bmatrix} B_{e1} \\ B_{e2} \\ \cdots \\ B_{en} \end{bmatrix}$$

Known

Unknown

Known

- Matrix Inversion to Solve for Radiosities.

Iterative approaches

- Jacobi iteration
- Start with initial guess for energy distribution (light sources)
- Update radiosity/power of all patches based on the previous guess

$$B_i = B_{e,i} + \rho_i \sum_{j=1}^{N} B_j F(i \rightarrow j)$$

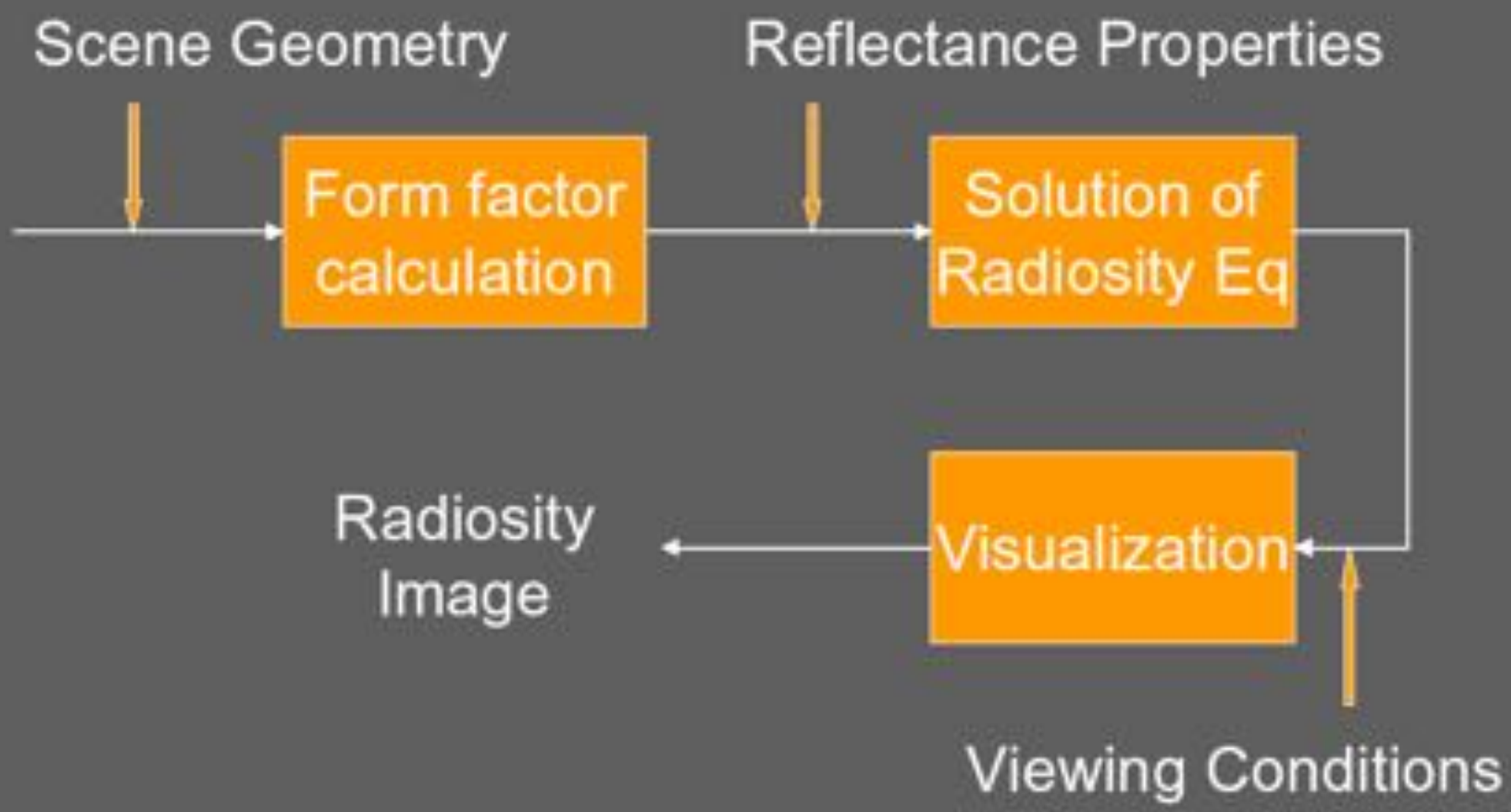
new value

old values

- Repeat until converged

© Kavita Bala, Computer Science, Cornell University

Radiosity "Pipeline"



Sample Scenes

Sample Scenes

From Cohen, Chen, Wallace and Greenberg 1988

Sample Scenes

Sample Scenes

Sample Scenes

Radiosity Summary

Classic radiosity = finite element method

Assumptions

- Diffuse reflectance
- Usually polygonal surfaces

Advantages

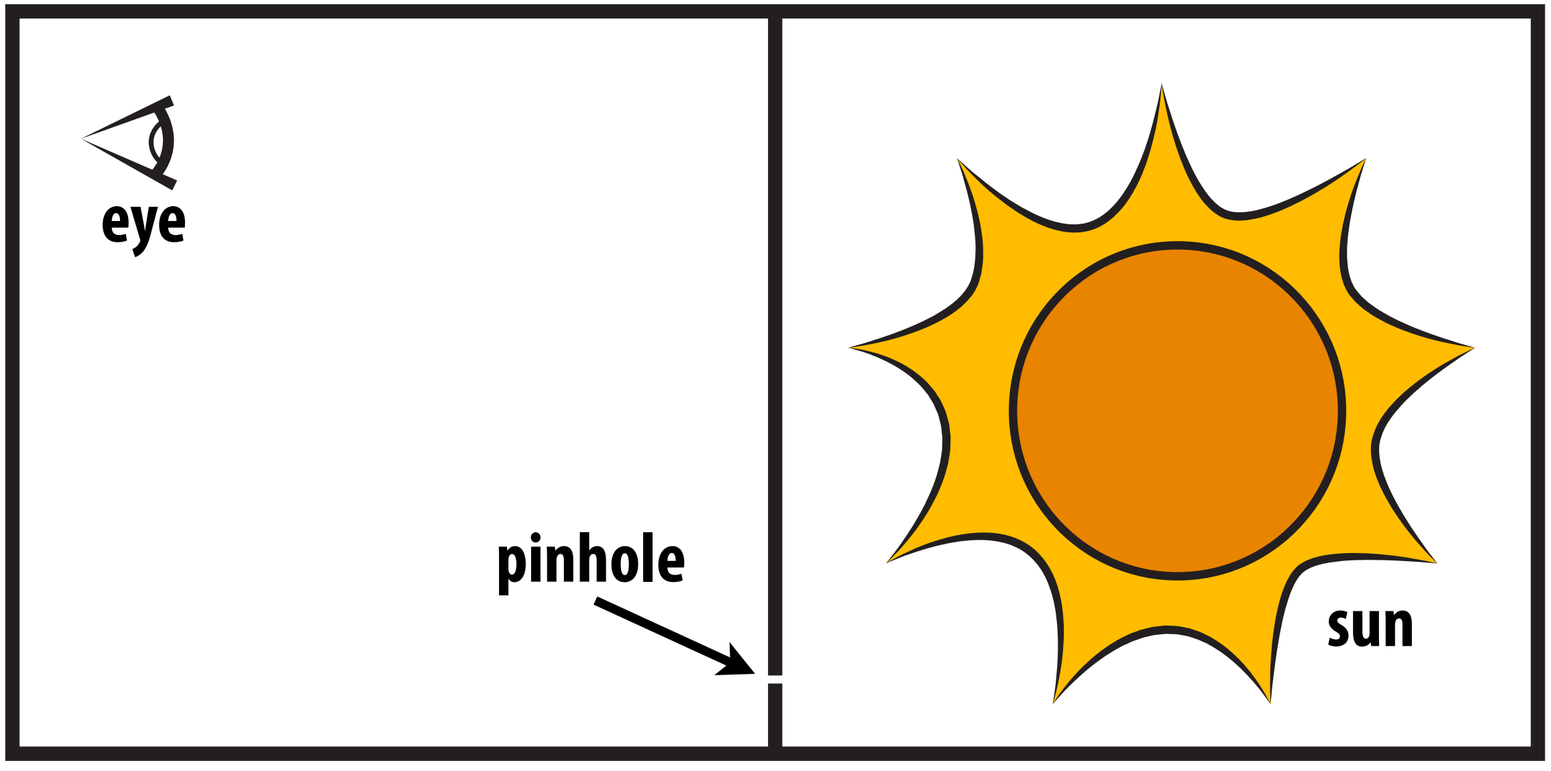
- Soft shadows and indirect lighting
- View independent solution
- Precompute for a set of light sources
- Useful for walkthroughs

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...!