Monte Carlo Rendering

Computer Graphics
CMU 15-462/15-662

TODAY: More on Monte Carlo Rendering

- How do we render a photorealistic image?
- Put together many of the ideas we've studied:
  - color
  - materials
  - radiometry
  - numerical integration
  - geometric queries
  - spatial data structures
  - rendering equation

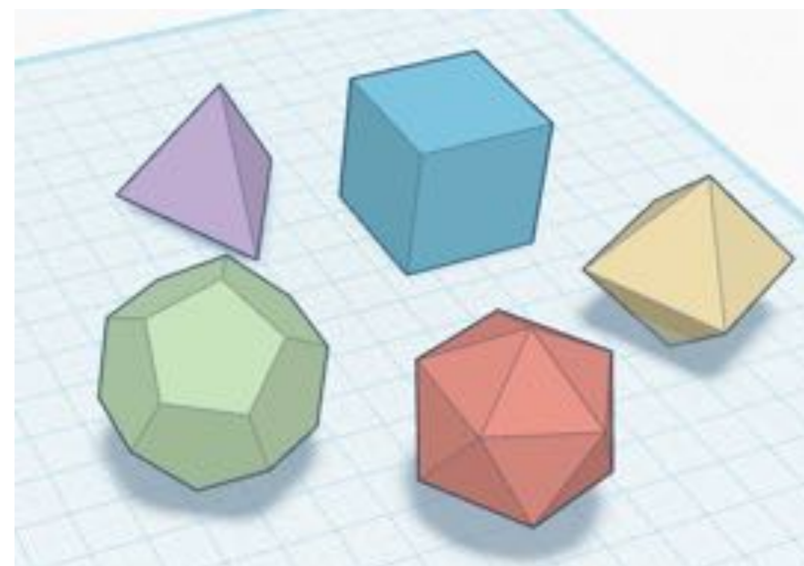
- Combine into final Monte Carlo ray tracing algorithm
- Alternative to rasterization, lets us generate much more realistic images (usually at much greater cost…)

Photorealistic Rendering—Basic Goal

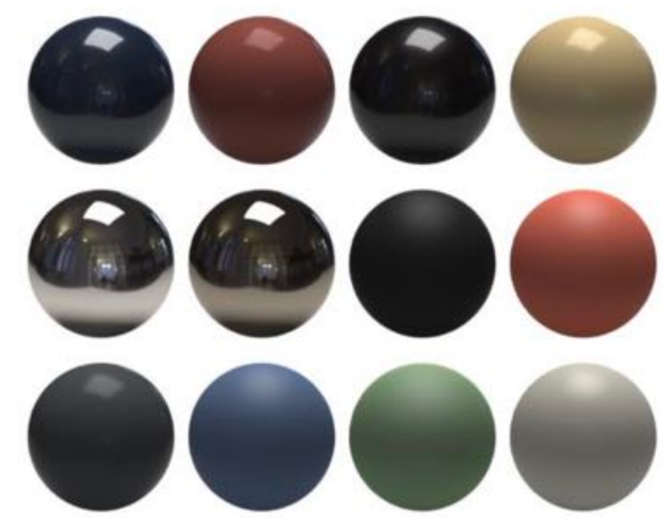
What are the INPUTS and OUTPUTS?

camera

geometry



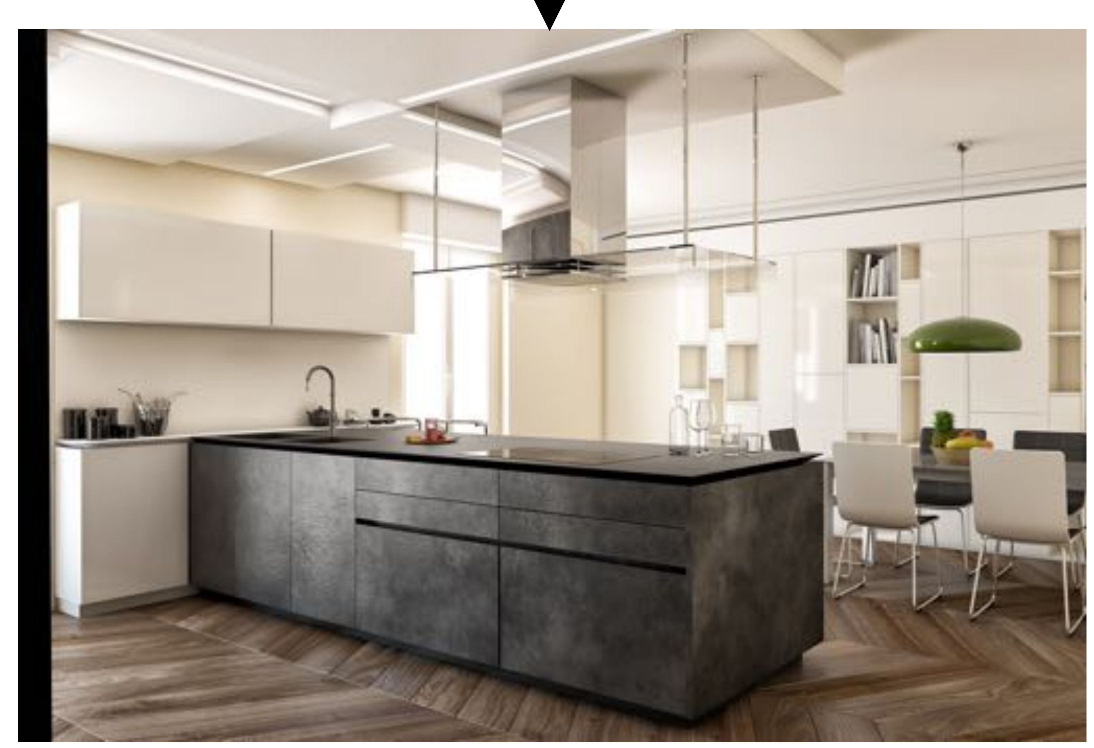
materials



lights

(“scene”)

Ray Tracer



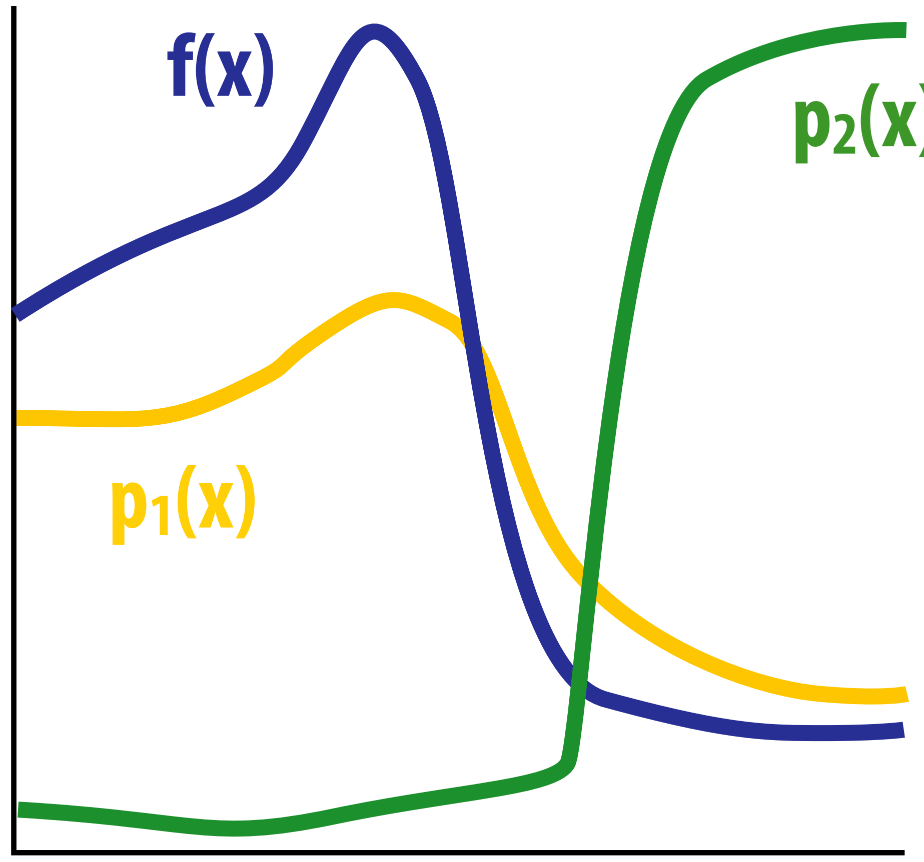
image

Monte Carlo Integration

$$\int_\Omega f(x)\, dx \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

Importance sampling

Q: Where is the best place to take samples?

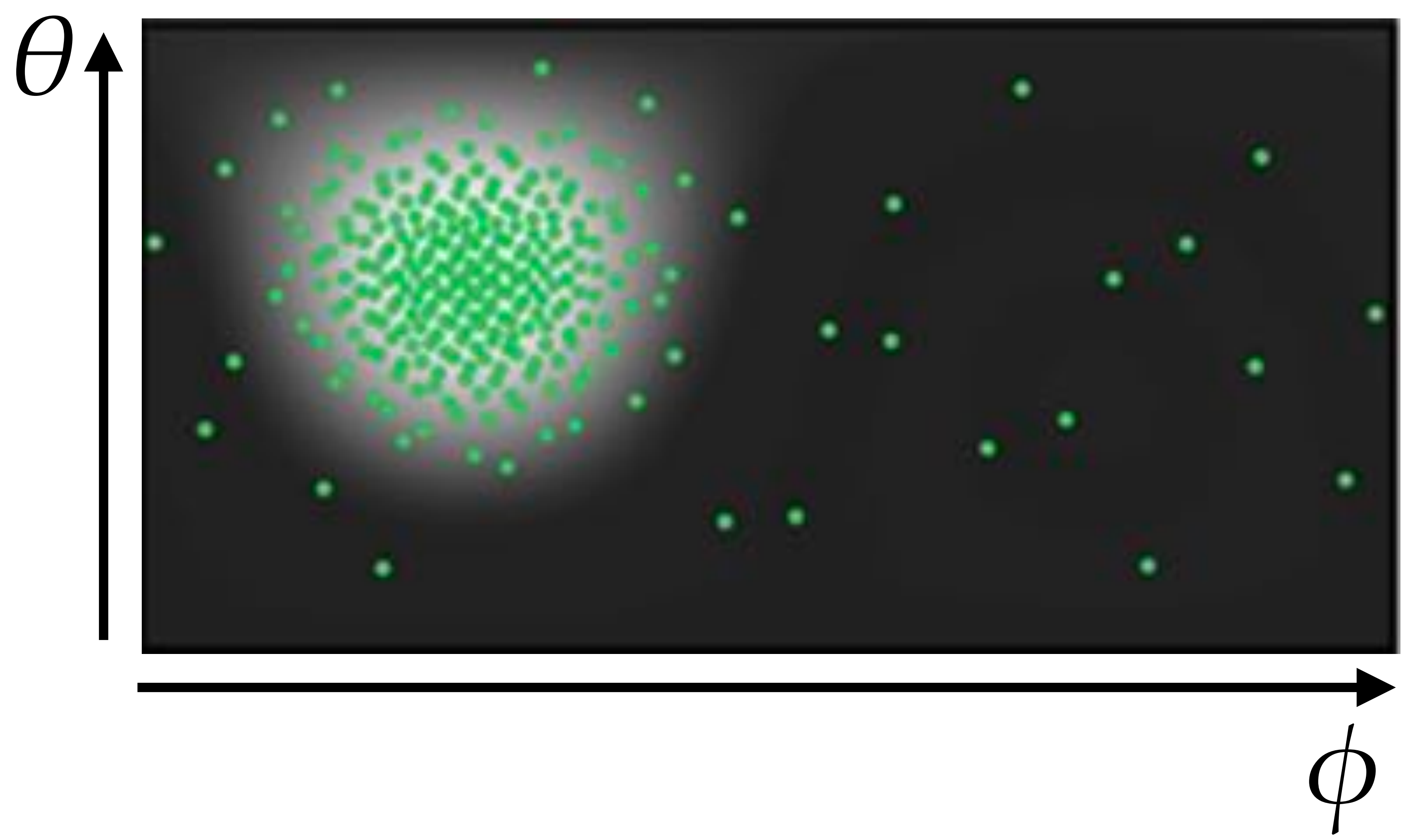


Think:

- What is the behavior of $f(x)/p_1(x)$? $f(x)/p_2(x)$?
- How does this impact the variance of the estimator?

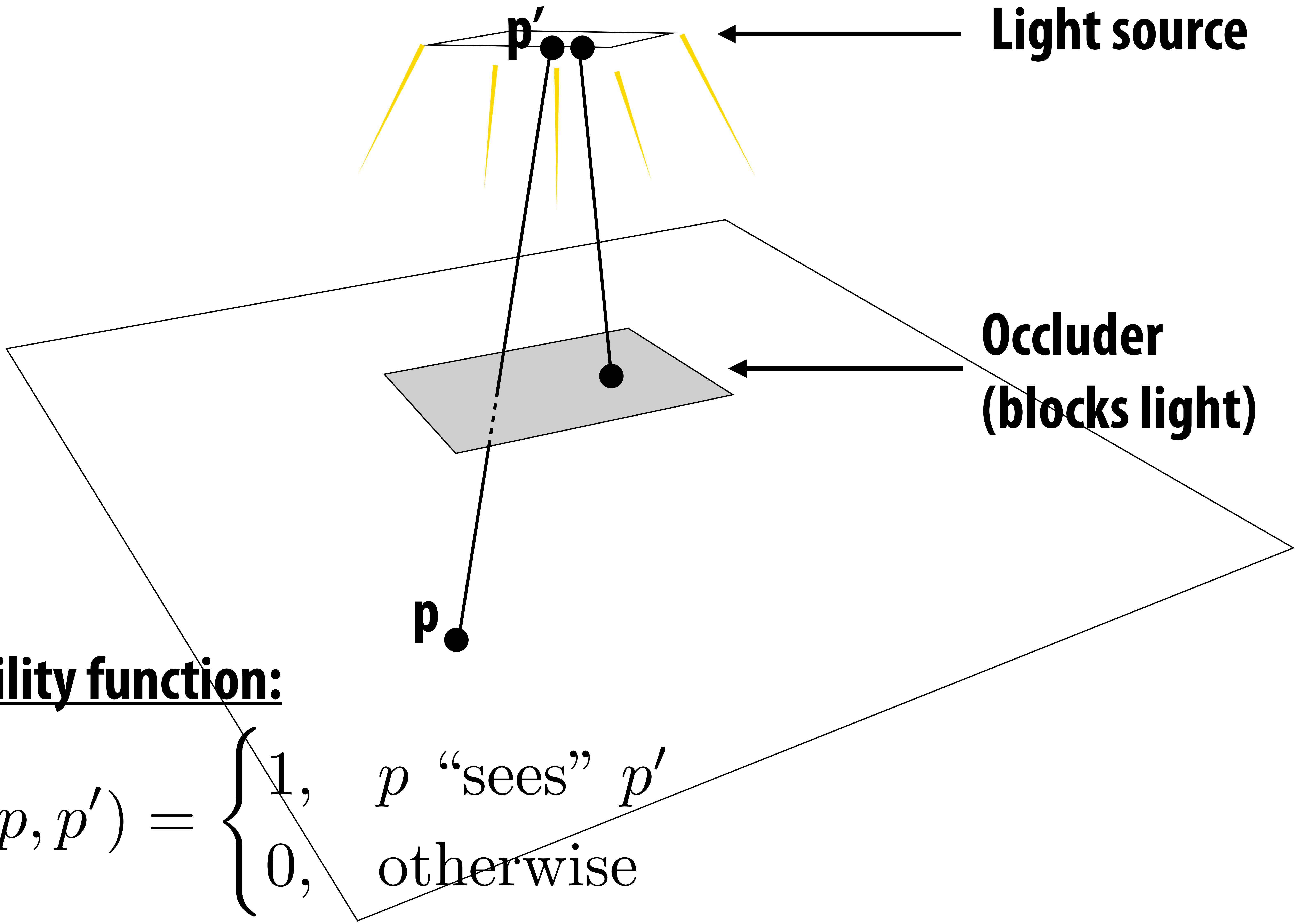
Idea: put more where integrand is large ("most useful samples"). E.g.:

(BRDF)



(image-based lighting)

Example: Direct Lighting



Visibility function:

$$V(p, p') = \begin{cases} 1, & p \text{ “sees” } p' \\ 0, & \text{otherwise} \end{cases}$$

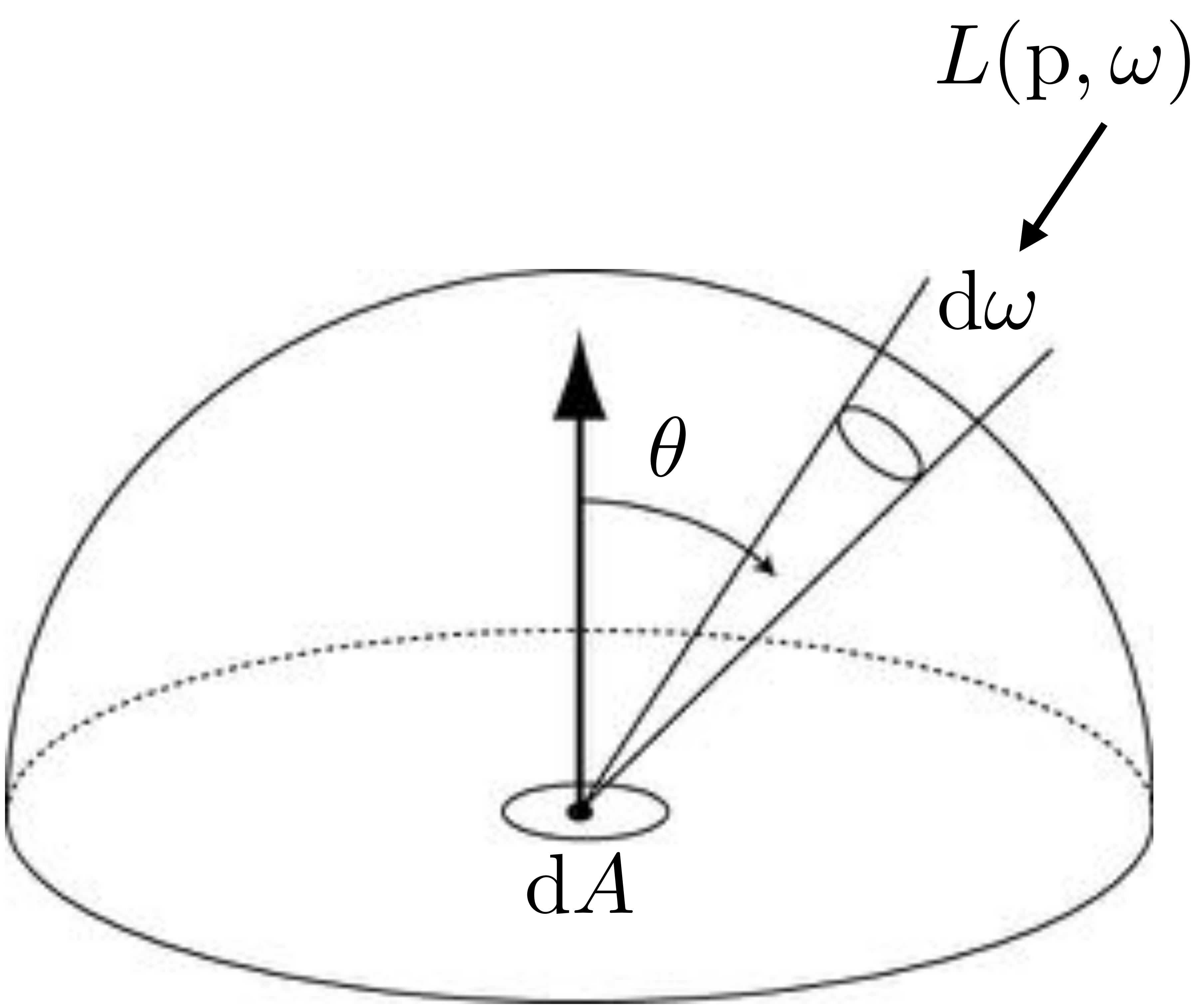
How bright is each point on the ground?

Direct lighting—uniform sampling

Uniformly-sample hemisphere of directions with respect to solid angle

$$p(\omega) = \frac{1}{2\pi}$$

$$E(\mathrm{p}) = \int L(\mathrm{p}, \omega)\ \cos\theta \,\mathrm{d}\omega$$



Estimator:

$$X_i \sim p(\omega)$$

$$Y_i = f(X_i)$$

$$Y_i = L(\mathrm{p}, \omega_i) cos\,\theta_i$$

$$F_N = \frac{2\pi}{N} \sum_{i=1}^{N} Y_i$$

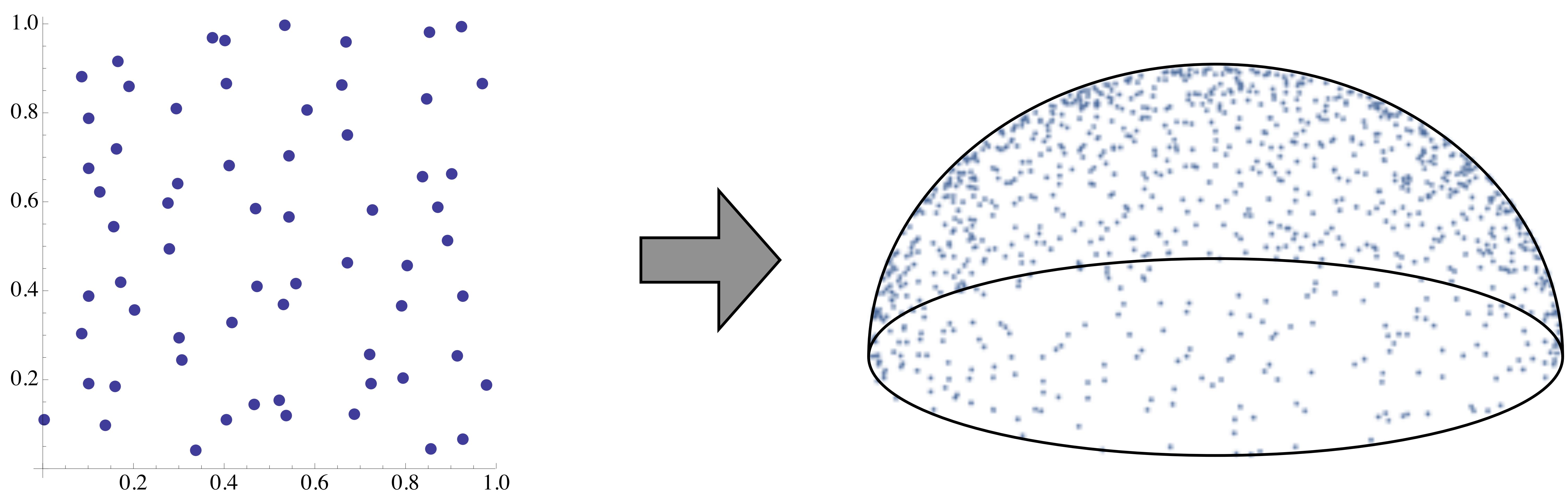
Aside: Picking points on unit hemisphere

How do we uniformly sample directions from the hemisphere?

One way: use rejection sampling. (How?)

Another way: “warp” two values in [0,1] via the inversion method:

$$(\xi_1, \xi_2) = (\sqrt{1-\xi_1^2}\cos(2\pi\xi_2), \sqrt{1-\xi_1^2}\sin(2\pi\xi_2), \xi_1)$$



Exercise: derive from the inversion method

Direct lighting—uniform sampling (algorithm)

Uniformly-sample hemisphere of directions with respect to solid angle

$$p(\omega) = \frac{1}{2\pi}$$

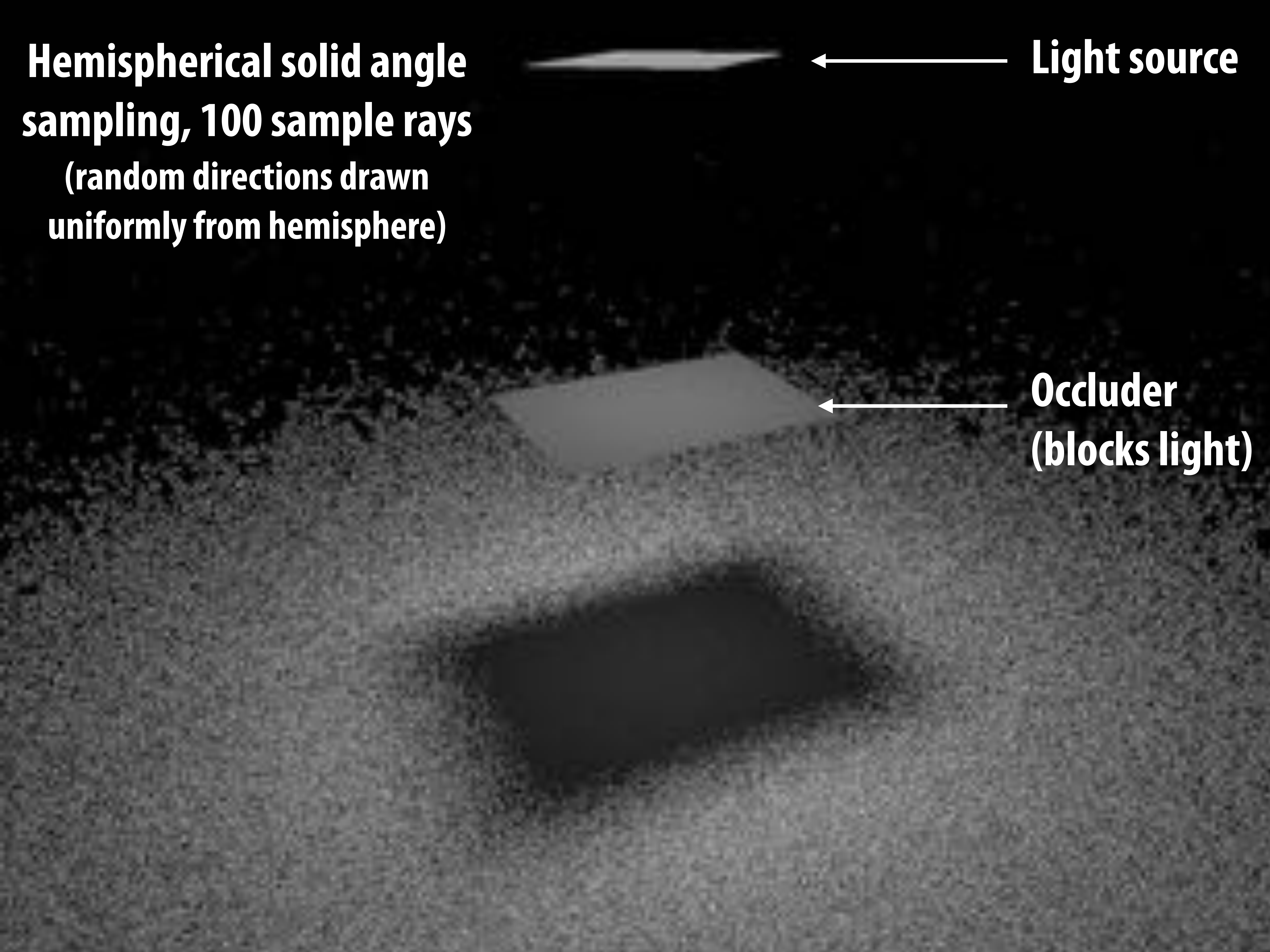
$$E(\mathrm{p}) = \int L(\mathrm{p}, \omega) \cos\theta \, \mathrm{d}\omega$$

Given surface point p

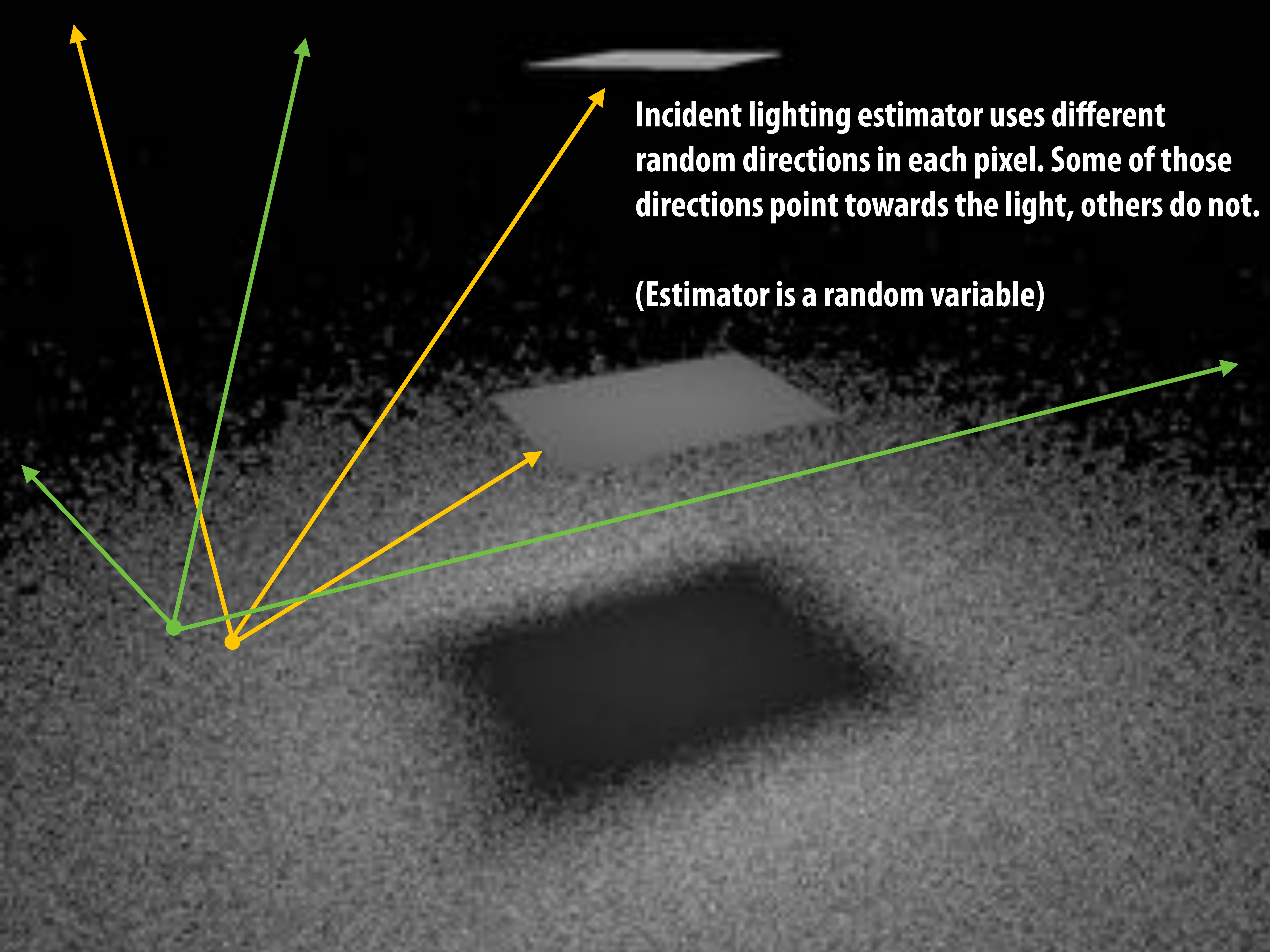
A ray tracer evaluates radiance along a ray (see Raytracer::trace_ray() in raytracer.cpp)

For each of N samples:

- **Generate random direction:** ω_i
- **Compute incoming radiance arriving** L_i **at p from direction:** ω_i
- **Compute incident irradiance due to ray:** $dE_i = L_i cos\, \theta_i$
- **Accumulate** $\frac{2\pi}{N} dE_i$ **into estimator**


Hemispherical solid angle
sampling, 100 sample rays
(random directions drawn
uniformly from hemisphere)
Light source
Occluder
(blocks light)

Why is the image in the previous slide “noisy”?



Incident lighting estimator uses different random directions in each pixel. Some of those directions point towards the light, others do not.

(Estimator is a random variable)

How can we reduce noise?

One idea: just take more samples!

Another idea:

- Don't need to integrate over entire hemisphere of directions (incoming radiance is 0 from most directions).
- Just integrate over the area of the light (directions where incoming radiance is non-zero)and weight appropriately

Direct lighting: area integral

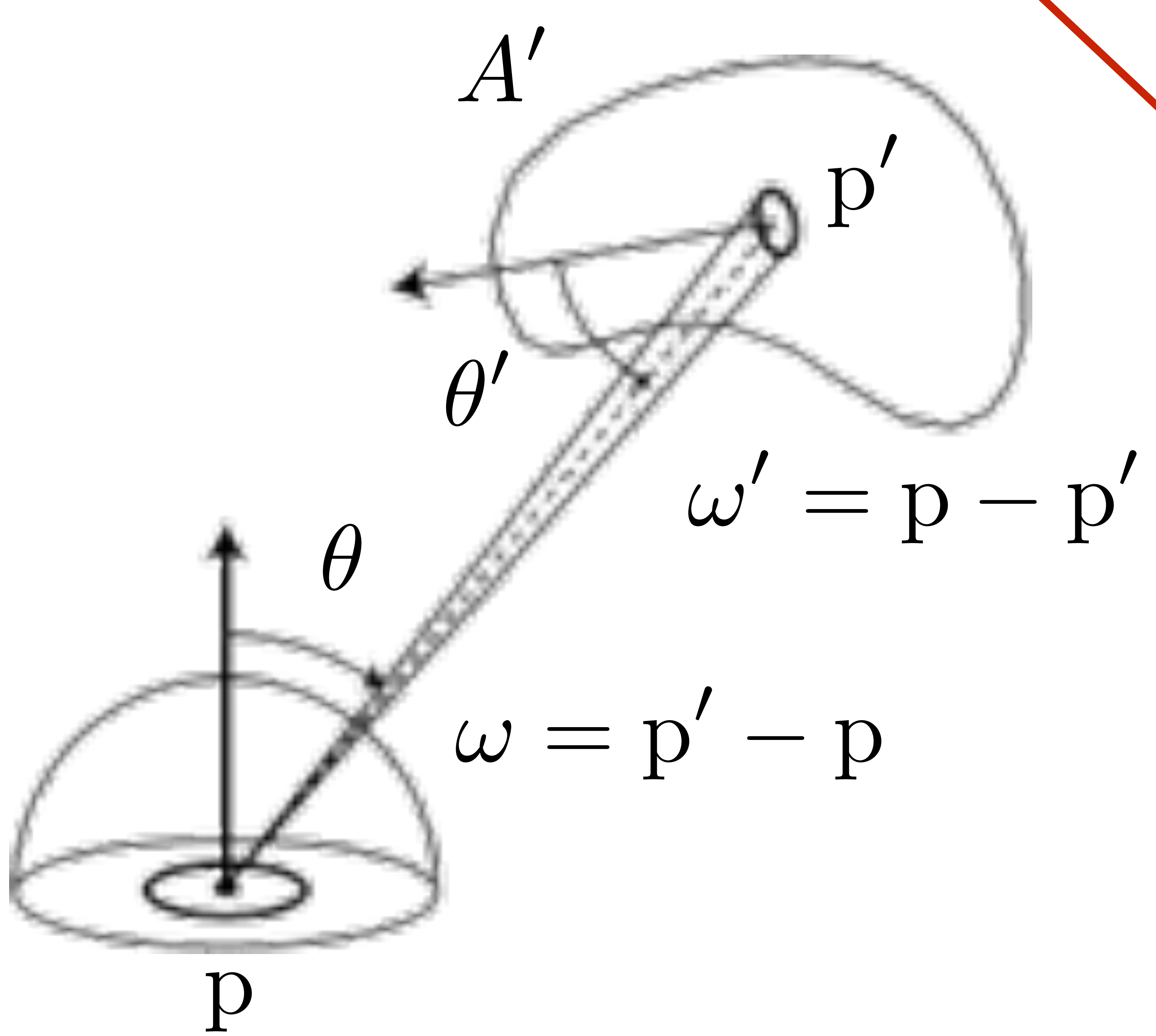
$$E(\mathrm{p}) = \int L(\mathrm{p}, \omega)\ \cos\theta\, \mathrm{d}\omega$$

← Previously: just integrate over all directions

$$E(\mathrm{p}) = \int_{A'} L_o(\mathrm{p}', \omega')\, V(\mathrm{p}, \mathrm{p}')\, \frac{\cos\theta \cos\theta'}{|\mathrm{p} - \mathrm{p}'|^2}\, \mathrm{d}A'$$

← Change of variables to integrate over area of light

$$d\omega = \frac{dA}{|\mathrm{p}' - \mathrm{p}|^2} = \frac{dA' \cos\theta'}{|\mathrm{p}' - \mathrm{p}|^2}$$

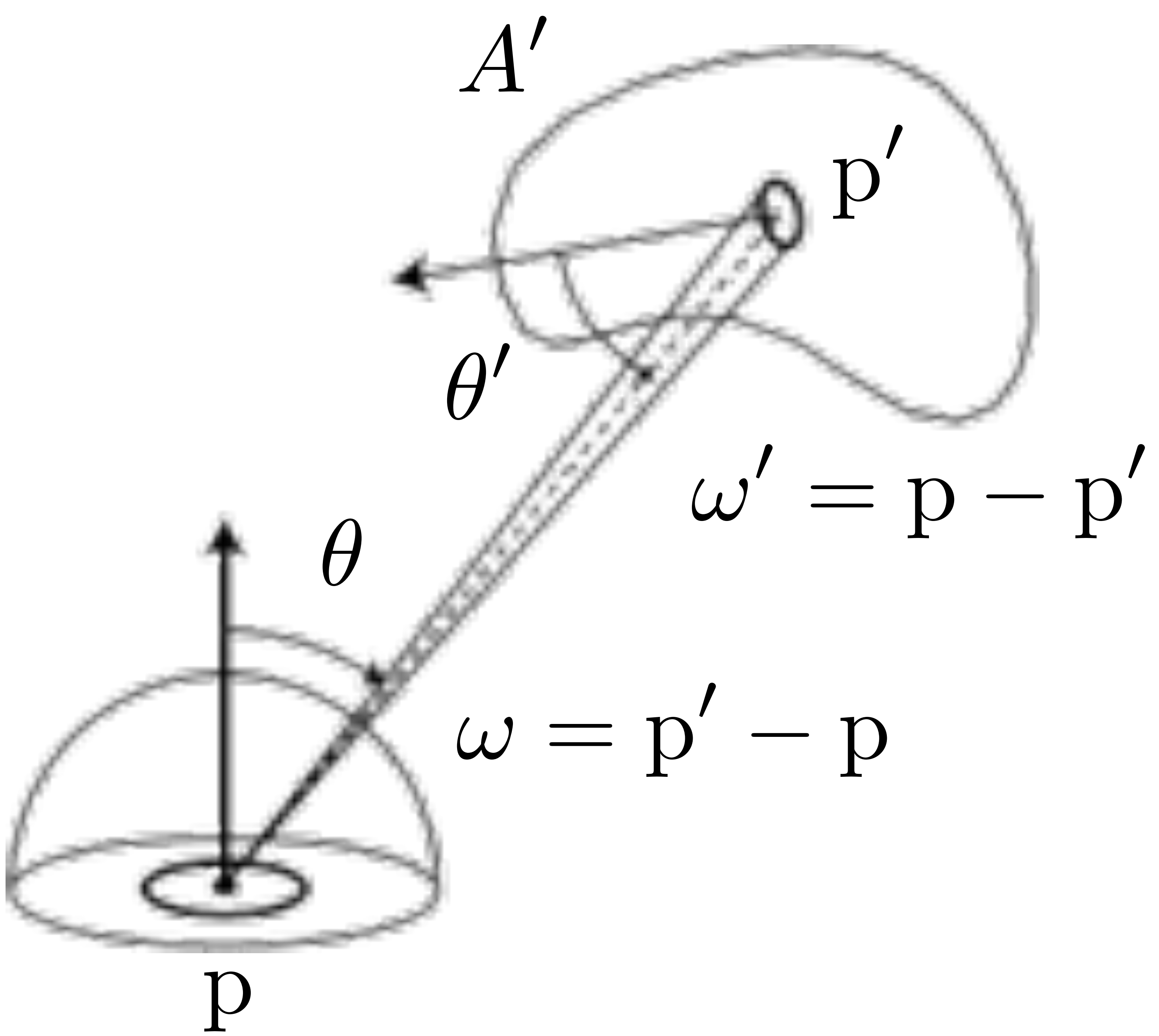


$V(\mathrm{p}, \mathrm{p}')$: Binary visibility function: 1 if p′ is visible from p, 0 otherwise (accounts for light occlusion)

$L_o(\mathrm{p}', \omega')$: Outgoing radiance from light point p′, in direction w′ towards p

Direct lighting: area integral

$$E(\mathrm{p}) = \int_{A'} L_o(\mathrm{p}', \omega')\, V(\mathrm{p}, \mathrm{p}')\, \frac{\cos\theta \cos\theta'}{|\mathrm{p} - \mathrm{p}'|^2}\, \mathrm{d}A'$$



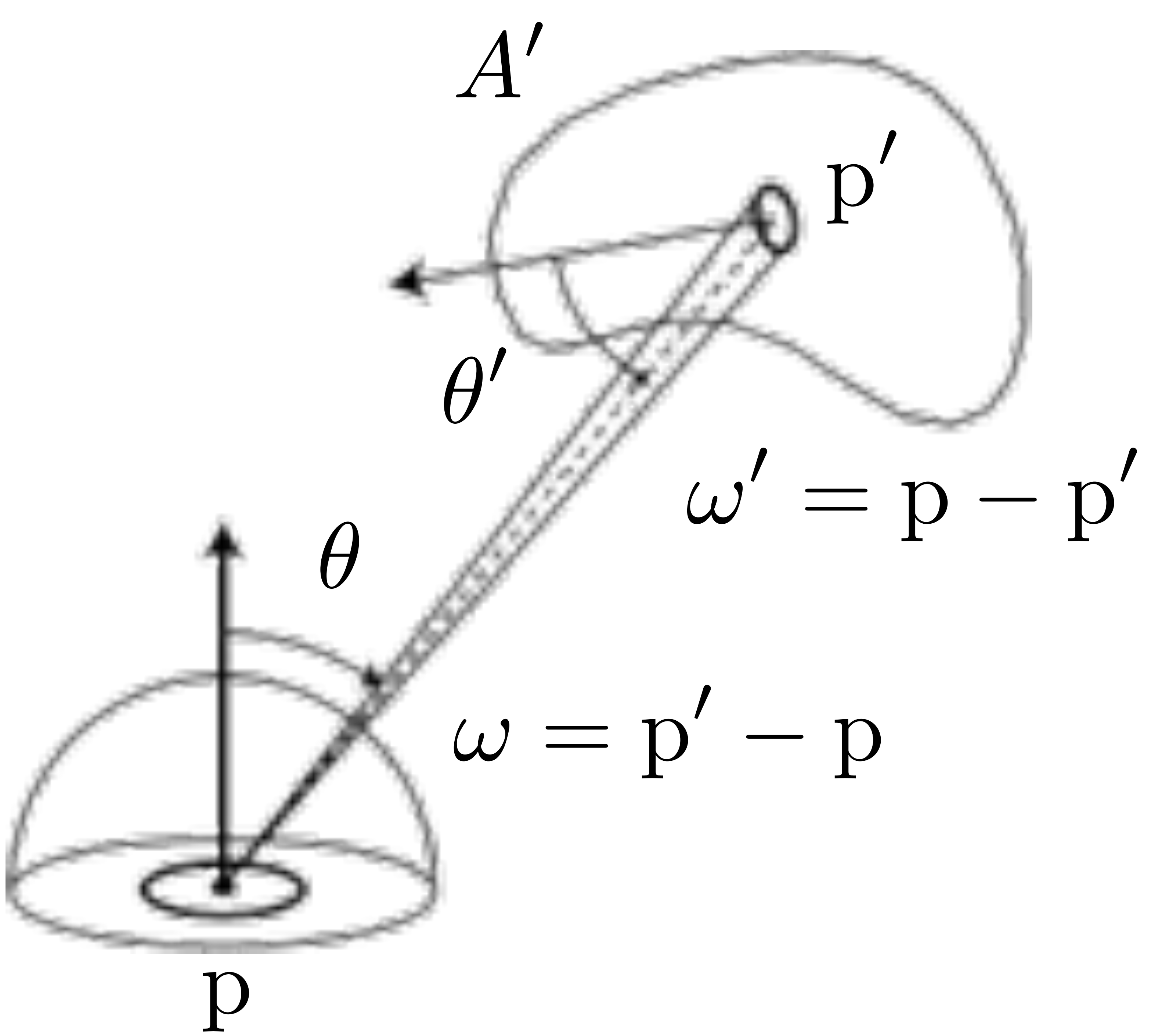
Sample shape uniformly by area A'

$$\int_{A'} p(\mathrm{p}')\, \mathrm{d}A' = 1$$

$$p(\mathrm{p}') = \frac{1}{A'}$$

Direct lighting: area integral

$$E(\mathrm{p}) = \int_{A'} L_o(\mathrm{p}', \omega')\, V(\mathrm{p}, \mathrm{p}')\, \frac{\cos\theta \cos\theta'}{|\mathrm{p} - \mathrm{p}'|^2}\, \mathrm{d}A'$$



Probability:

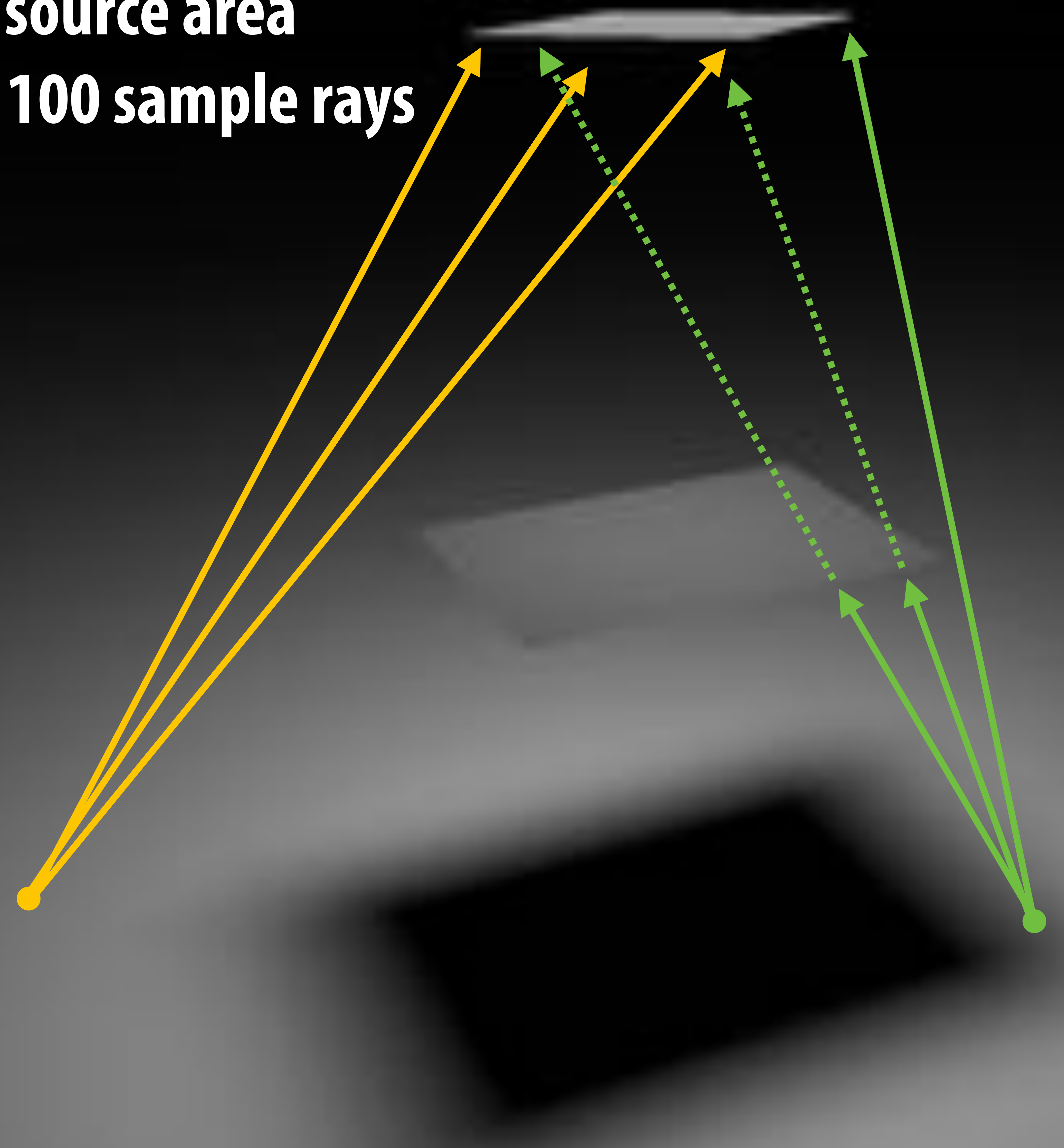
$$p(\mathrm{p}') = \frac{1}{A'}$$

Estimator

$$Y_i = L_o(\mathrm{p}'_i, \omega'_i)\, V(\mathrm{p}, \mathrm{p}'_i)\, \frac{\cos\theta_i \cos\theta'_i}{|\mathrm{p} - \mathrm{p}'_i|^2}$$

$$F_N = \frac{A'}{N} \sum_{i=1}^{N} Y_i$$

Light source area sampling, 100 sample rays

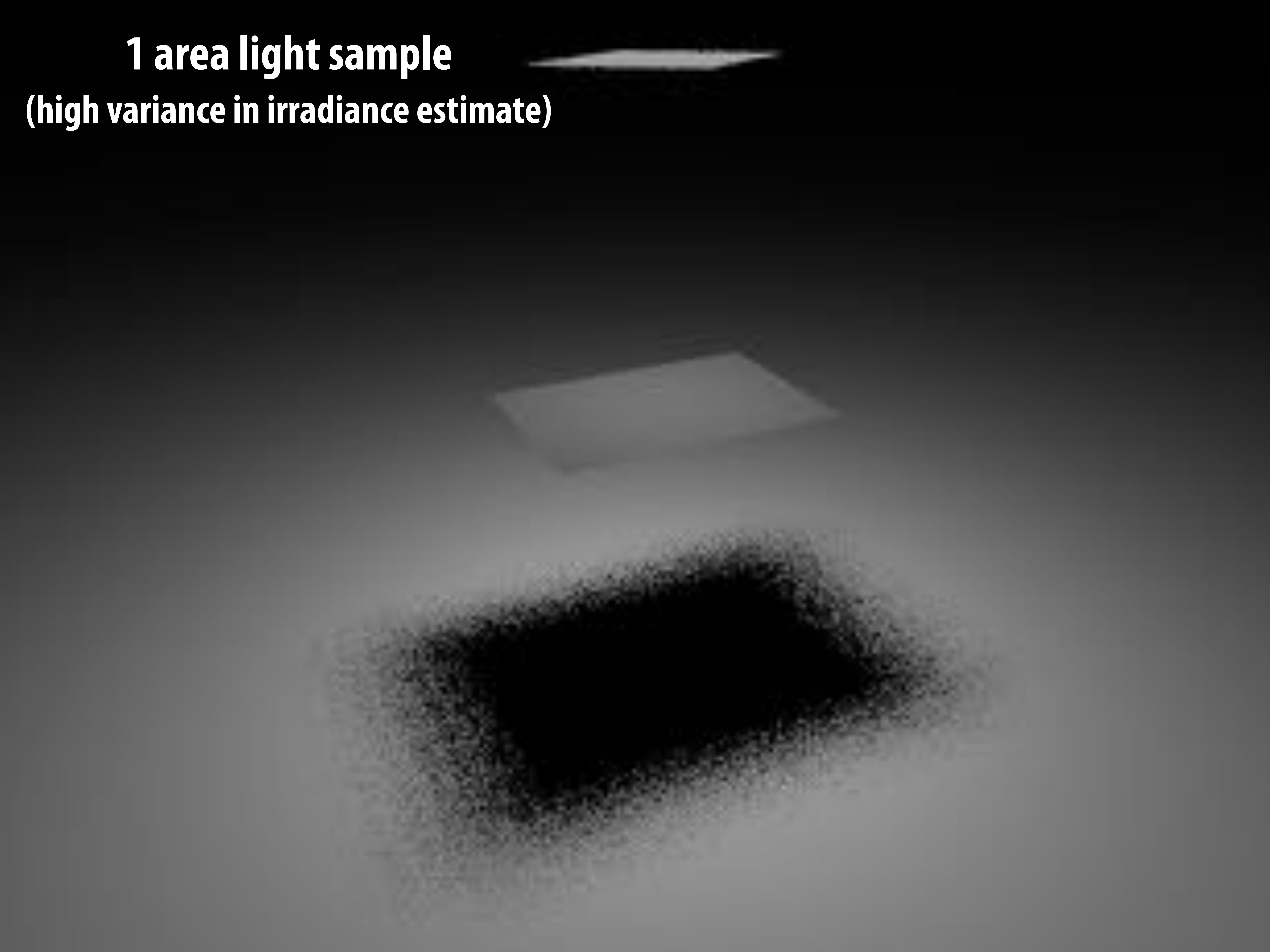


If no occlusion is present, all directions chosen in computing estimate “hit” the light source.

(Choice of direction only matters if portion of light is occluded from surface point p.)

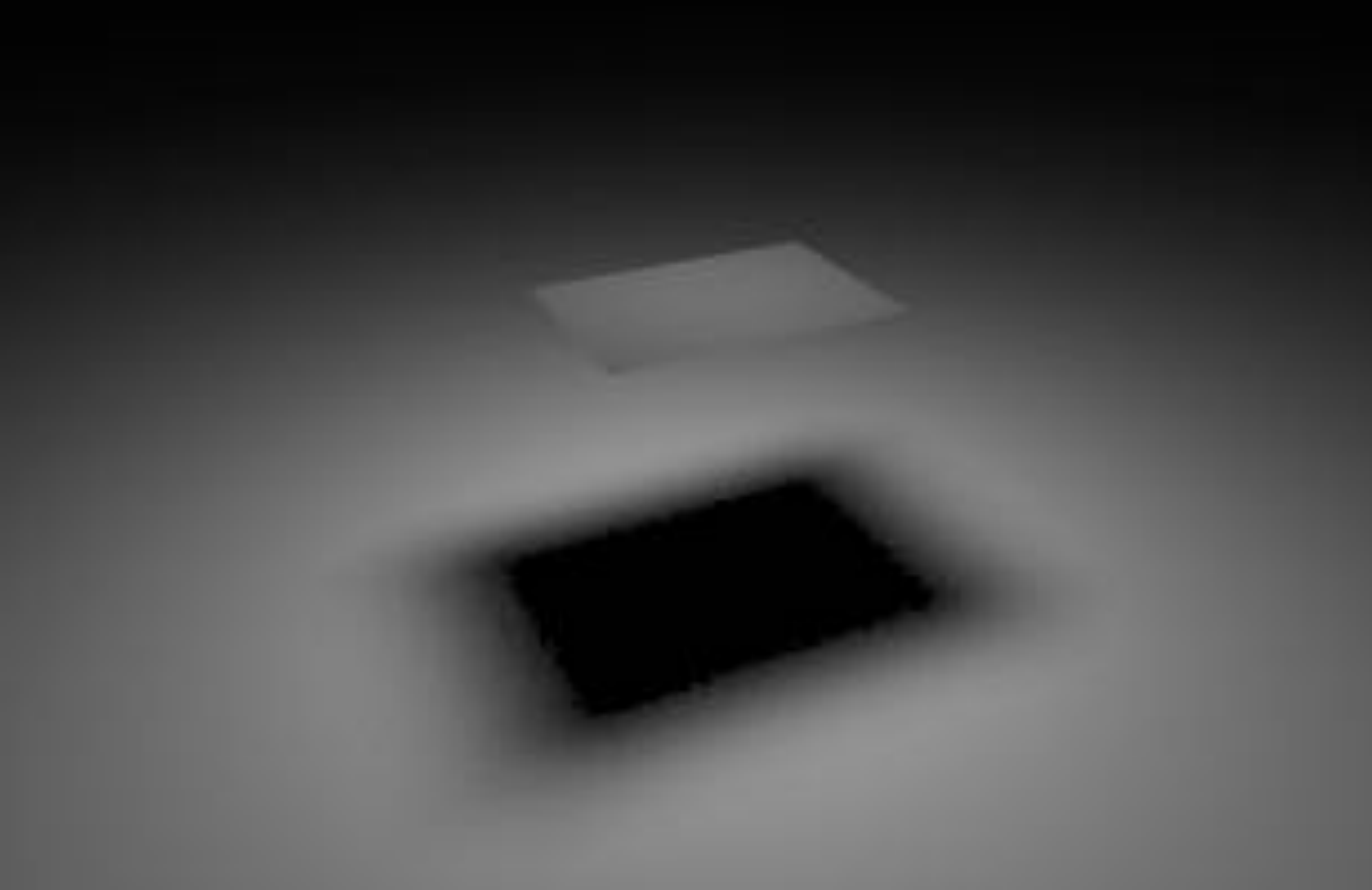
1 area light sample

(high variance in irradiance estimate)

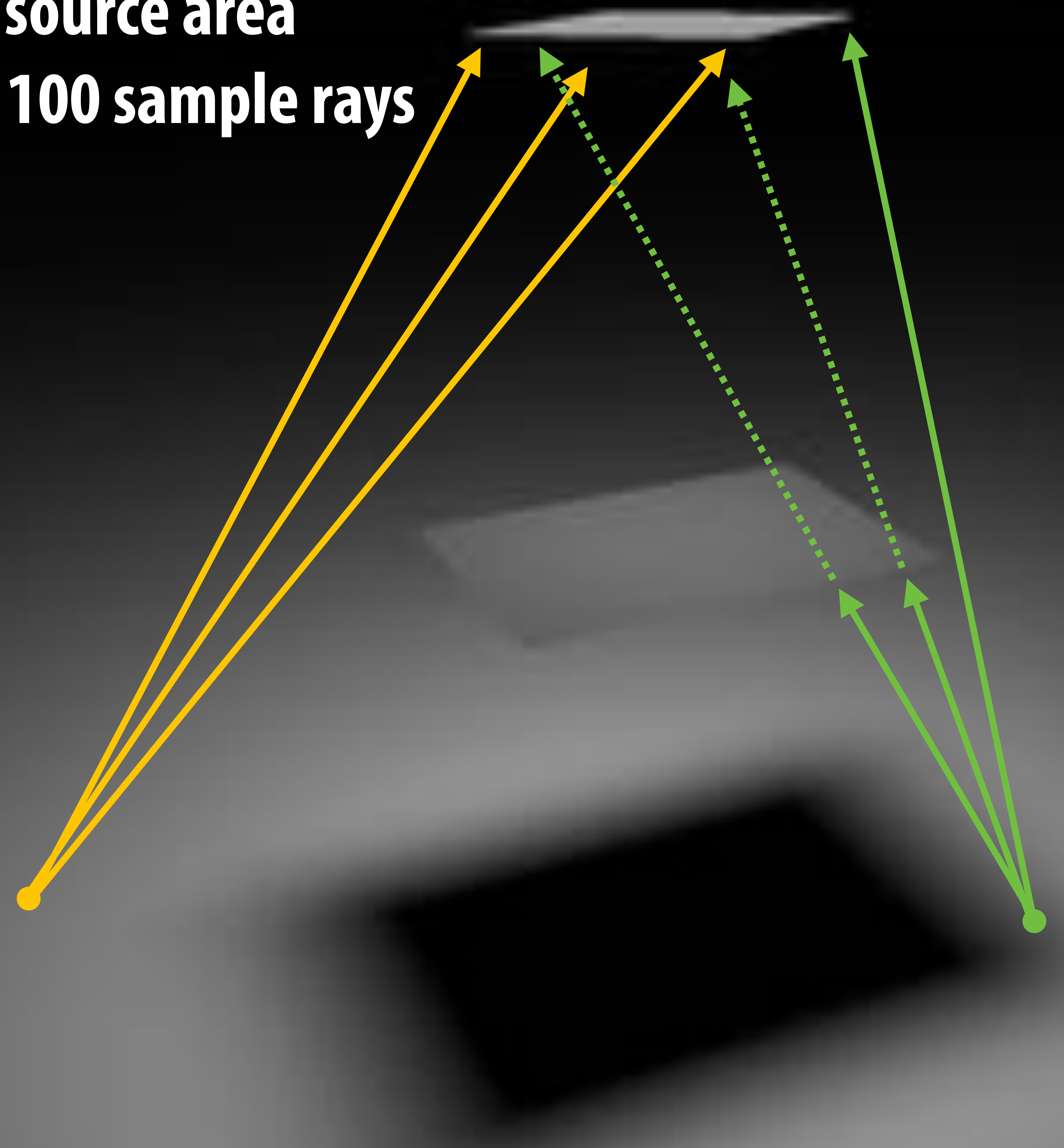


16 area light samples

(lower variance in irradiance estimate)



Light source area sampling, 100 sample rays



If no occlusion is present, all directions chosen in computing estimate “hit” the light source.

(Choice of direction only matters if portion of light is occluded from surface point p.)

Comparing different techniques

- **Variance in an estimator manifests as noise in rendered images**
- **Estimator efficiency measure:**

$$\text{Efficiency} \propto \frac{1}{\text{Variance} \times \text{Cost}}$$

- **If one integration technique has twice the variance of another, then it takes twice as many samples to achieve the same variance**

- **If one technique has twice the cost of another technique with the same variance, then it takes twice as much time to achieve the same variance**

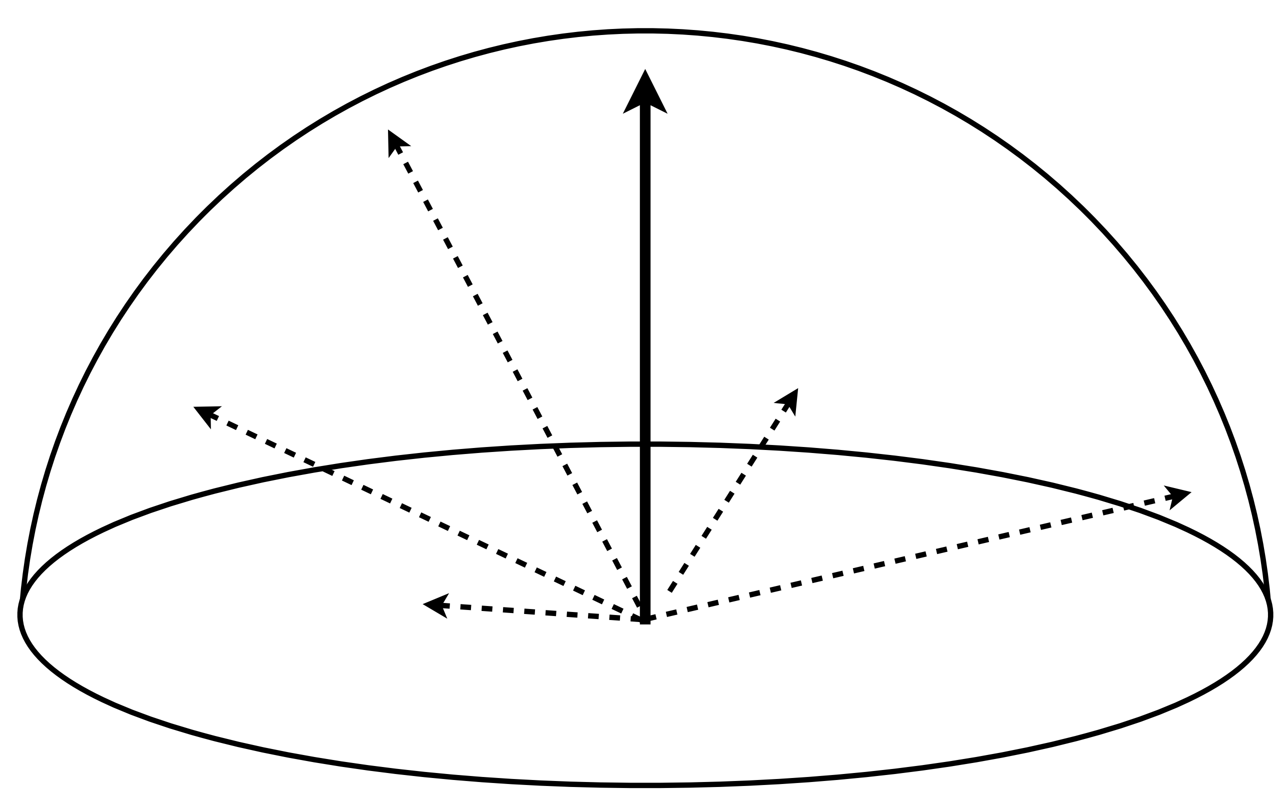
Example—Recall Uniform Hemisphere Sampling

Consider uniform hemisphere sampling in irradiance estimate:

$$f(\omega) = L_i(\omega)\cos\theta \qquad p(\omega) = \frac{1}{2\pi}$$

$$(\xi_1, \xi_2) = (\sqrt{1-\xi_1^2}\cos(2\pi\xi_2), \sqrt{1-\xi_1^2}\sin(2\pi\xi_2), \xi_1)$$

$$\int_\Omega f(\omega)\,\mathrm{d}\omega \approx \frac{1}{N}\sum_i^N \frac{f(\omega)}{p(\omega)} = \frac{1}{N}\sum_i^N \frac{L_i(\omega)\ \cos\theta}{1/2\pi} = \boxed{\frac{2\pi}{N}\sum_i^N L_i(\omega)\cos\theta}$$



Example—Cosine-Weighted Sampling

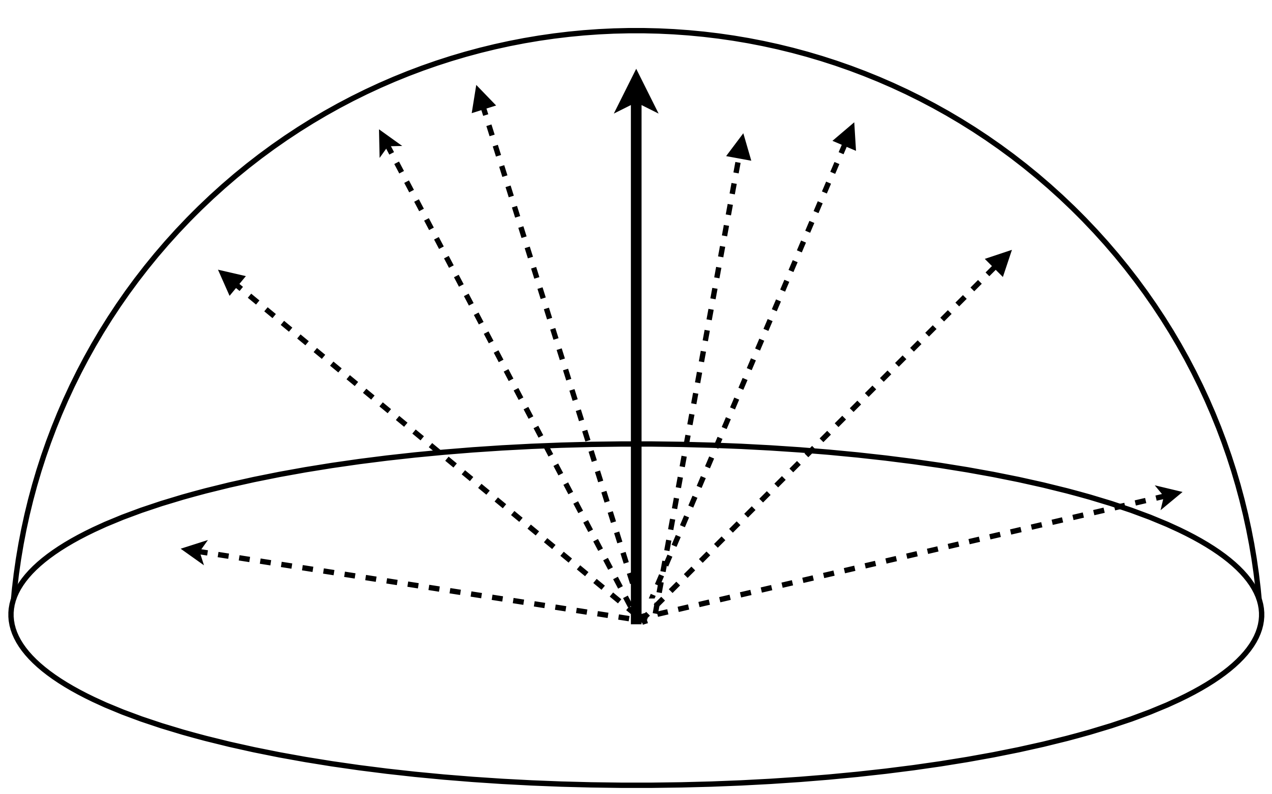
Cosine-weighted hemisphere sampling in irradiance estimate:

$$f(\omega) = L_i(\omega)\cos\theta \qquad\qquad p(\omega) = \frac{\cos\theta}{\pi}$$

$$\int_\Omega f(\omega)\,\mathrm{d}\omega \approx \frac{1}{N}\sum_i^N \frac{f(\omega)}{p(\omega)} = \frac{1}{N}\sum_i^N \frac{L_i(\omega)\ \cos\theta}{\cos\theta/\pi} = \boxed{\frac{\pi}{N}\sum_i^N L_i(\omega)}$$

Idea: bias samples toward directions where $\cos\theta$ is large

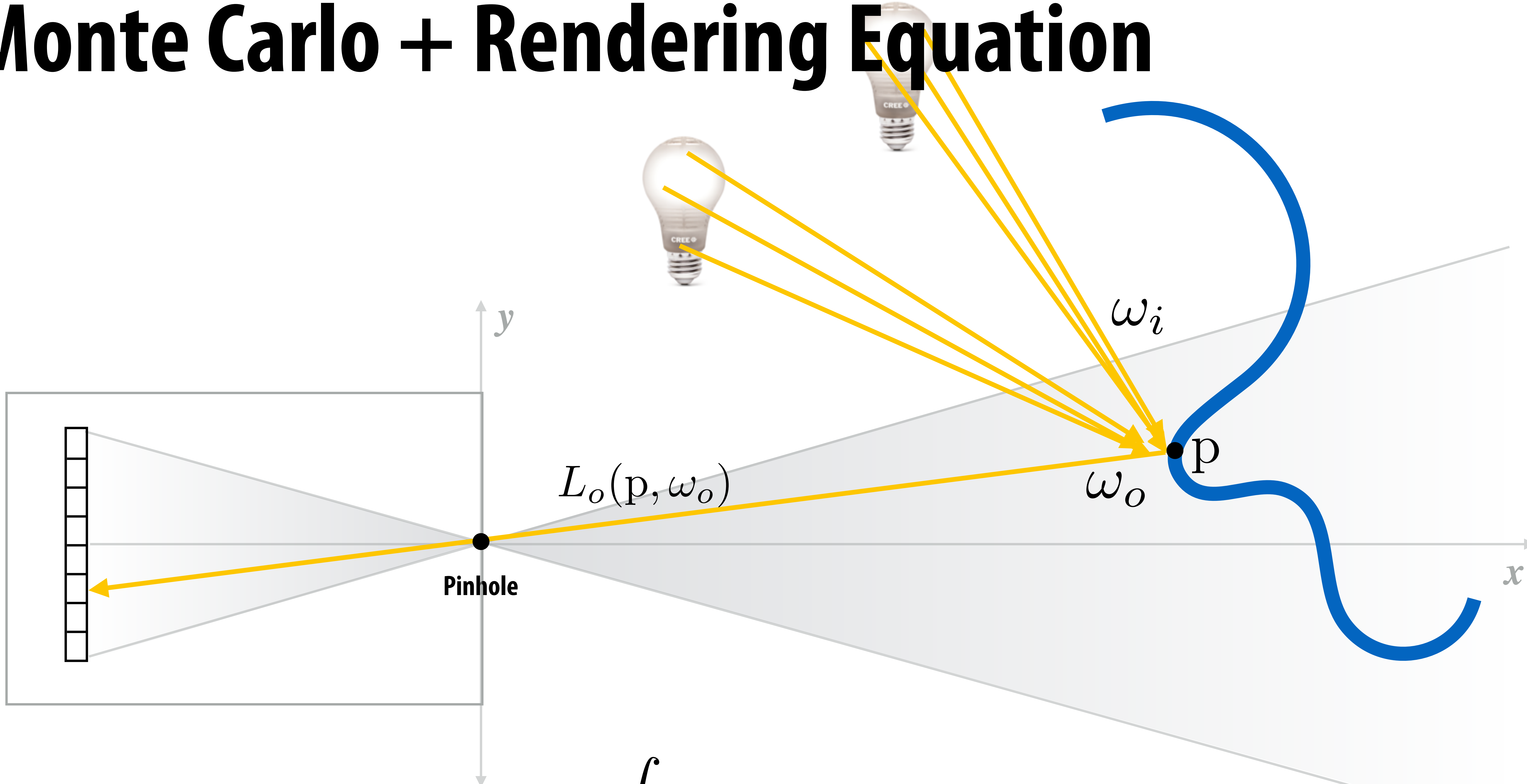
(if L is constant, then these are the directions that contribute most)



So far we've considered light coming directly from light sources, scattered once.

How do we use Monte Carlo integration to get the final color values for each pixel?

Monte Carlo + Rendering Equation

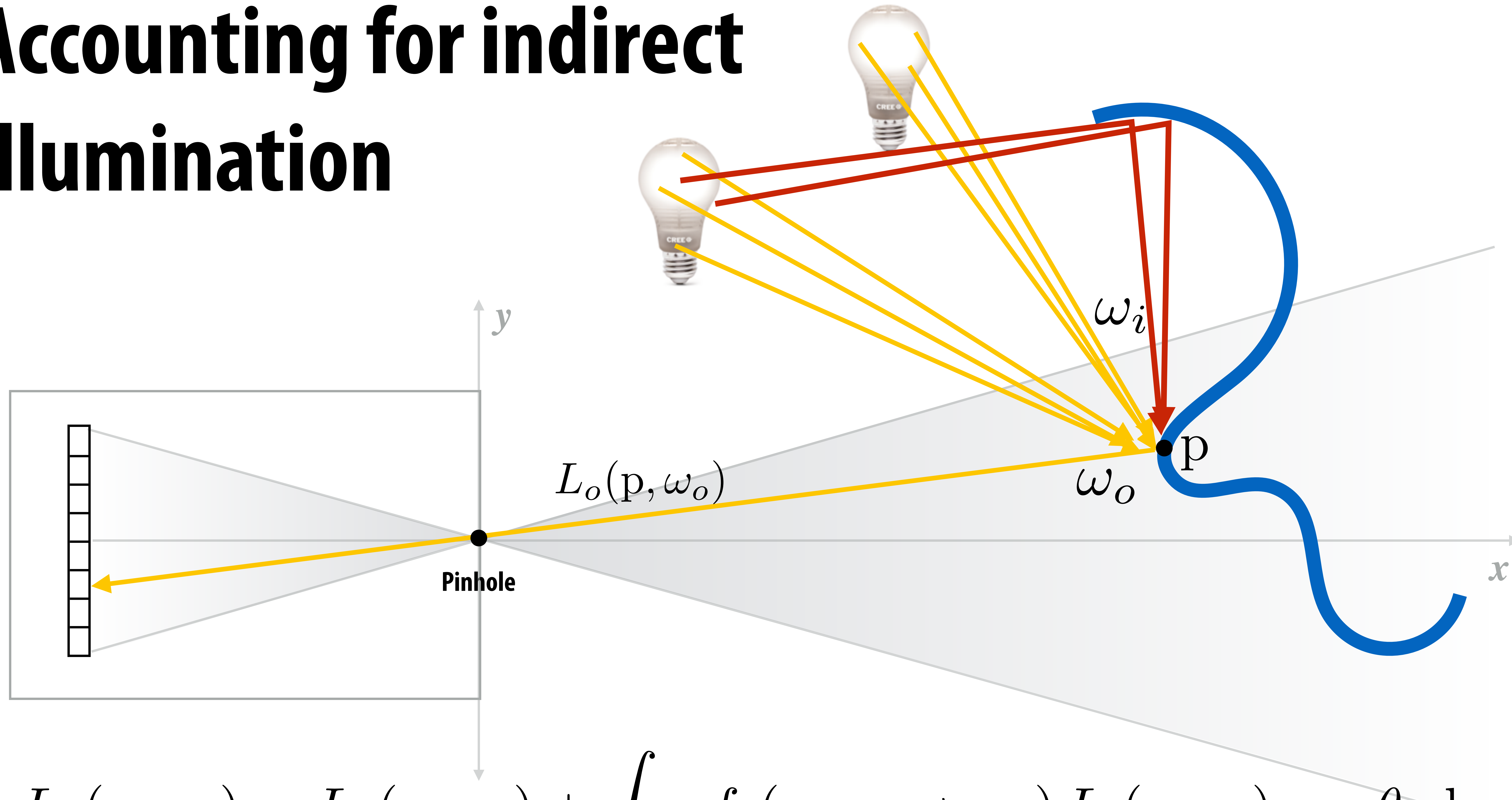


$$L_o(\mathrm{p}, \omega_o) = L_e(\mathrm{p}, \omega_o) + \int_{H^2} f_r(\mathrm{p}, \omega_i \to \omega_o)\, L_i(\mathrm{p}, \omega_i)\, \cos\theta_i \, \mathrm{d}\omega_i$$

Need to know incident radiance.

So far, have only computed incoming radiance from scene light sources.

Accounting for indirect illumination



$$L_o(\mathrm{p}, \omega_o) = L_e(\mathrm{p}, \omega_o) + \int_{H^2} f_r(\mathrm{p}, \omega_i \to \omega_o)\, L_i(\mathrm{p}, \omega_i)\, \cos\theta_i \, \mathrm{d}\omega_i$$

Incoming light energy from direction ω_i may be due to light reflected off another surface in the scene (not an emitter)

Path tracing: indirect illumination

$$\int_{H^2} f_r(\omega_i \to \omega_o)\, L_{o,i}(tr(\mathrm{p}, \omega_i), -\omega_i)\, \cos\theta_i\, \mathrm{d}\omega_i$$

- **Sample incoming direction from some distribution (e.g. proportional to BRDF):**

$$\omega_i \sim p(\omega)$$

- **Recursively call path tracing function to compute incident indirect radiance**


p
Direct illumination

One-bounce global illumination

Two-bounce global illumination

Four-bounce global illumination


p
Eight-bounce global illumination

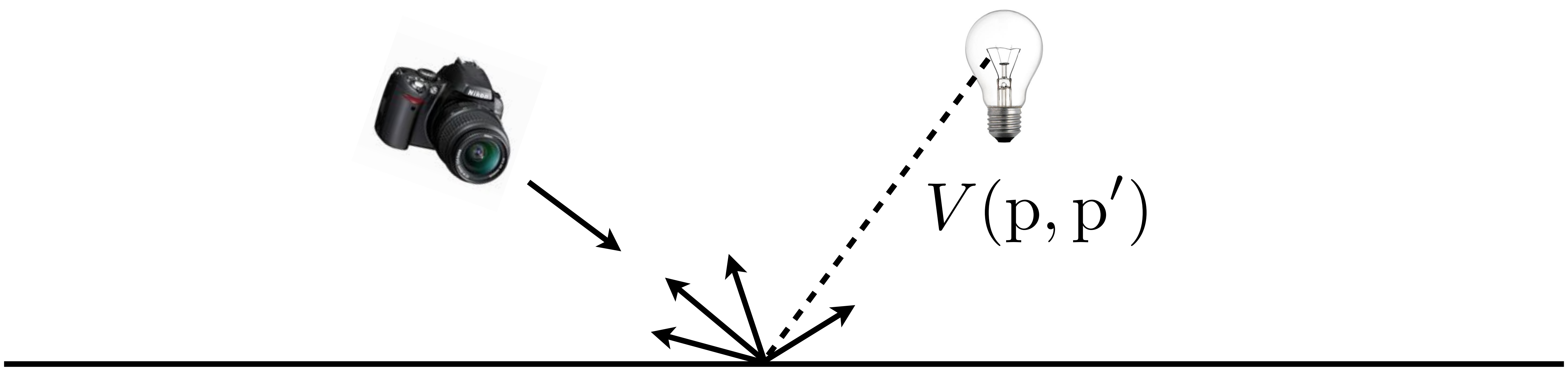
•p
Sixteen-bounce global illumination

Wait a minute…
When do we stop?!

Russian roulette

- **Idea: want to avoid spending time evaluating function for samples that make a small contribution to the final result**
- **Consider a low-contribution sample of the form:**

$$L = \frac{f_r(\omega_i \to \omega_o)\, L_i(\omega_i)\, V(\mathrm{p}, \mathrm{p}')\ \cos\theta_i}{p(\omega_i)}$$



Russian roulette

$$L = \frac{f_r(\omega_i \to \omega_o)\, L_i(\omega_i)\, V(\mathrm{p}, \mathrm{p}')\ \cos\theta_i}{p(\omega_i)}$$

$$\downarrow$$

$$L = \left[\frac{f_r(\omega_i \to \omega_o)\, L_i(\omega_i)\ \cos\theta_i}{p(\omega_i)}\right]\ V(\mathrm{p}, \mathrm{p}')$$

- **If tentative contribution (in brackets) is small, total contribution to the image will be small regardless of** $V(\mathrm{p}, \mathrm{p}')$
- **Ignoring low-contribution samples introduces systematic error**
  - **No longer converges to correct value!**
- **Instead, randomly discard low-contribution samples in a way that leaves estimator unbiased**

Russian roulette

- **New estimator: evaluate original estimator with probability p_{rr}, reweight. Otherwise ignore.**
- **Same expected value as original estimator:**

$$p_{rr} E\left[\frac{Y}{p_{rr}}\right] + E\left[(1 - p_{rr})0\right] = E\left[Y\right]$$

No Russian roulette: 6.4 seconds

Russian roulette: terminate 50% of all contributions with luminance less than 0.25: 5.1 seconds

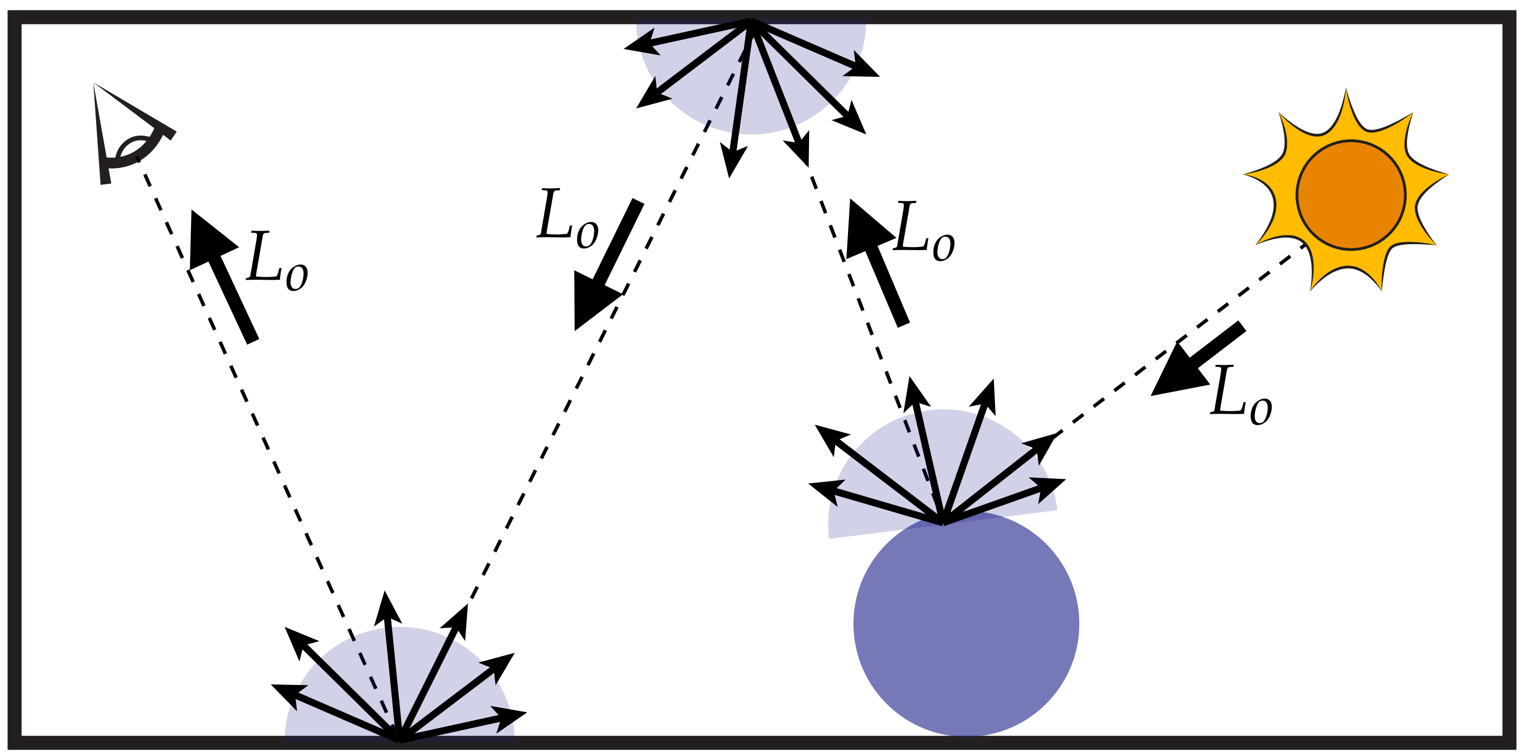
Russian roulette: terminate 50% of all contributions with luminance less than 0.5: 4.9 seconds

Russian roulette: terminate 90% of all contributions with luminance less than 0.125: 4.8 seconds

Russian roulette: terminate 90% of all contributions with luminance less than 1: 3.6 seconds

Monte Carlo Rendering—Summary

- Light hitting a point (e.g., pixel) described by rendering equation
  - Expressed as recursive integral
  - Can use Monte Carlo to estimate this integral
  - Need to be intelligent about how to sample!



$$L_o(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_{\mathcal{H}^2} f_r(\mathbf{p}, \omega_i \to \omega_o) L_i(\mathbf{p}, \omega_i) \cos\theta \, d\omega_i$$

Next time:

- **Variance reduction—how do we get the most out of our samples?**