Lecture 11:

Monte Carlo Evaluation of the Reflection Equation

Interactive Computer Graphics
Stanford CS248A, Winter 2026

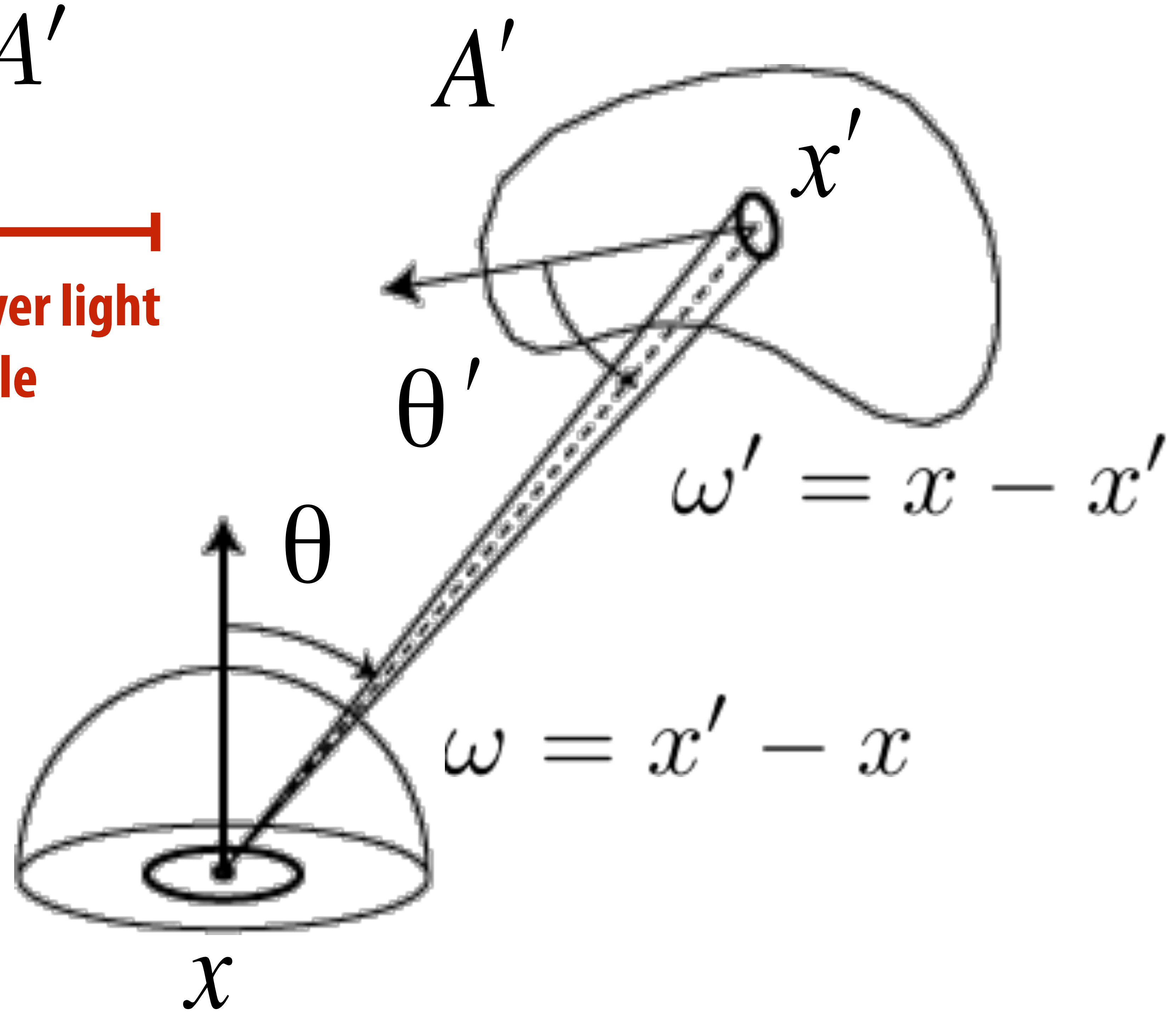
Review: irradiance at point X from uniform area source

Assume area light emits radiance L from all directions from all points on surface.

$$E(x) = \int_{H^2} L_i(x,\omega)\,\cos\theta\, d\omega = \int_{A'} L \frac{\cos\theta\cos\theta'}{|x-x'|^2} dA'$$

Integrate over solid angle

Reparameterization: now integrate over light source area, instead of solid angle



Integral reparameterization:

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

Radiance leaving light from x′ in direction ω′ is the same as radiance arriving at x from ω:

$$L_i(x,\omega) = L_o(x',\omega') = L$$

Review:
Q. How do we estimate the value of these integrals?
(A. Monte Carlo integration)

Review: Monte Carlo integration

- **Definite integral**

  What we seek to estimate

$$\int_a^b f(x)dx$$

- **Random variables**

  X_i **is the value of a random sample drawn from the distribution** $p(x)$

  Y_i **is also a random variable.**

$$X_i \sim p(x)$$
$$Y_i = f(X_i)$$

- **Expectation of *f***

$$E[Y_i] = E[f(X_i)] = \int_a^b f(x)\, p(x)\, \mathrm{d}x$$

- **MC estimator: (assuming samples X_i are drawn from uniform random sampling of domain) ***

Monte Carlo estimate of $\int_a^b f(x)dx$ **is given by** $F_N = \frac{b-a}{N} \sum_{i=1}^{N} Y_i$

*** We'll relax this assumption shortly.**

Basic Monte Carlo estimator

$$E[F_N] = E\left[\frac{b-a}{N}\sum_{i=1}^{N} Y_i\right] = \int_a^b f(x)\,\mathrm{d}x$$

Where:

$$Y_i = f(X_i)$$

$$X_i \sim U(a,b)$$ ← Uniform distribution over domain [a,b]

$$p(x) = \frac{1}{b-a}$$

Note: Even though my notation suggests this is integration over 1D domain [a,b], this holds for Uniform sampling over any integration domain, such as the 2D hemisphere of solid angles on the previous slide.

Why this works…

Unbiased estimator: Expected value of estimator is the integral we wish to evaluate.

$$E[F_N] = E\left[\frac{b-a}{N}\sum_{i=1}^{N} Y_i\right]$$

$$= \frac{b-a}{N}\sum_{i=1}^{N} E[Y_i] = \frac{b-a}{N}\sum_{i=1}^{N} E[f(X_i)]$$

$$= \frac{b-a}{N}\sum_{i=1}^{N} \int_a^b f(x)\,p(x)\mathrm{d}x$$

Uniform density, so:

$$p(x) = \frac{1}{b-a}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \int_a^b f(x)\,\mathrm{d}x$$

$$= \int_a^b f(x)\,\mathrm{d}x$$

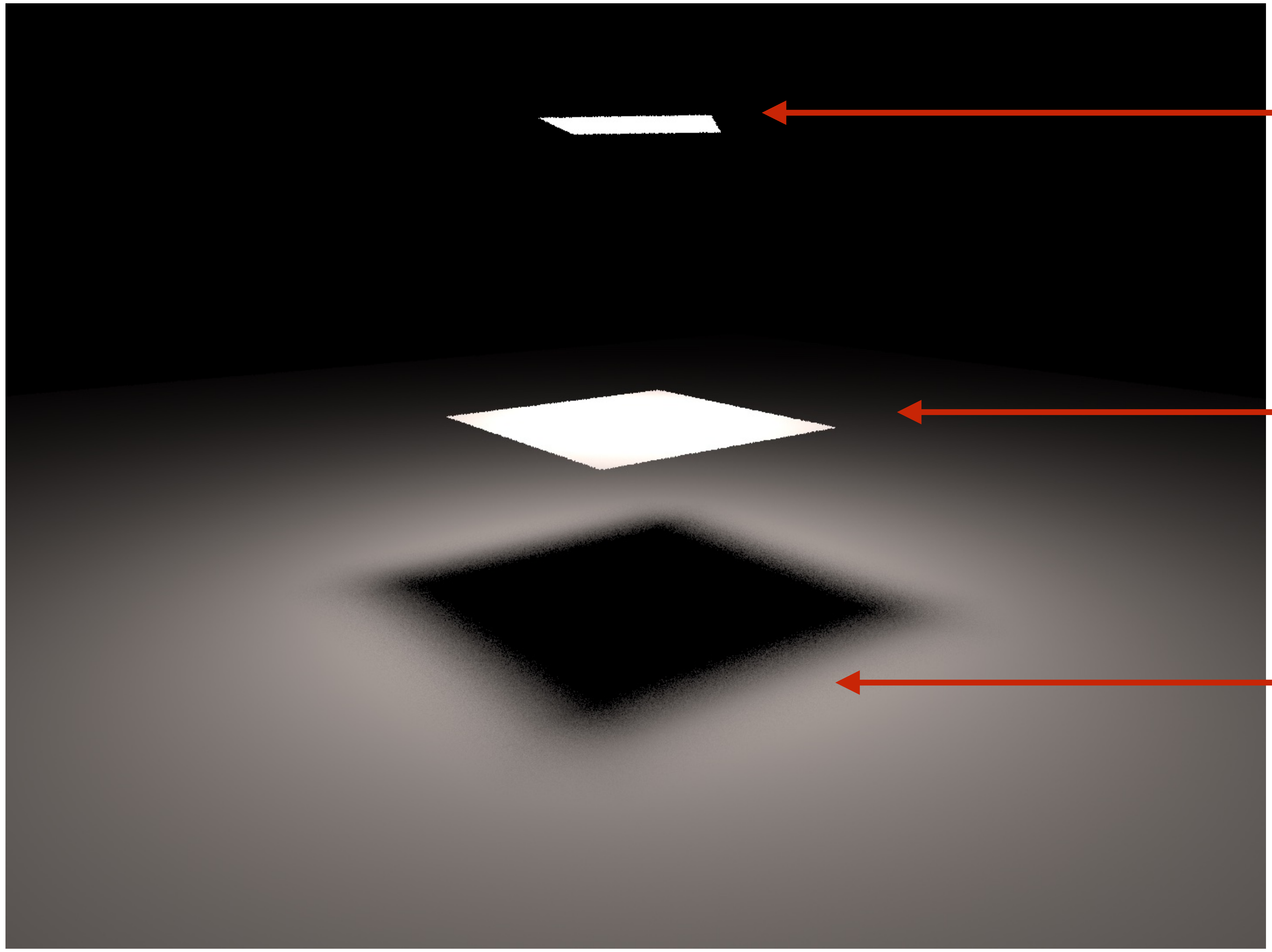
Properties of expectation:

$$E\left[\sum_i Y_i\right] = \sum_i E[Y_i]$$

$$E[aY] = aE[Y]$$

Note: Even though my notation suggests this is integration over 1D domain [a,b], this proof holds for any integration domain, such as the 2D hemisphere of directions on the previous slide.

Direct lighting example: light from an area light source



Light source

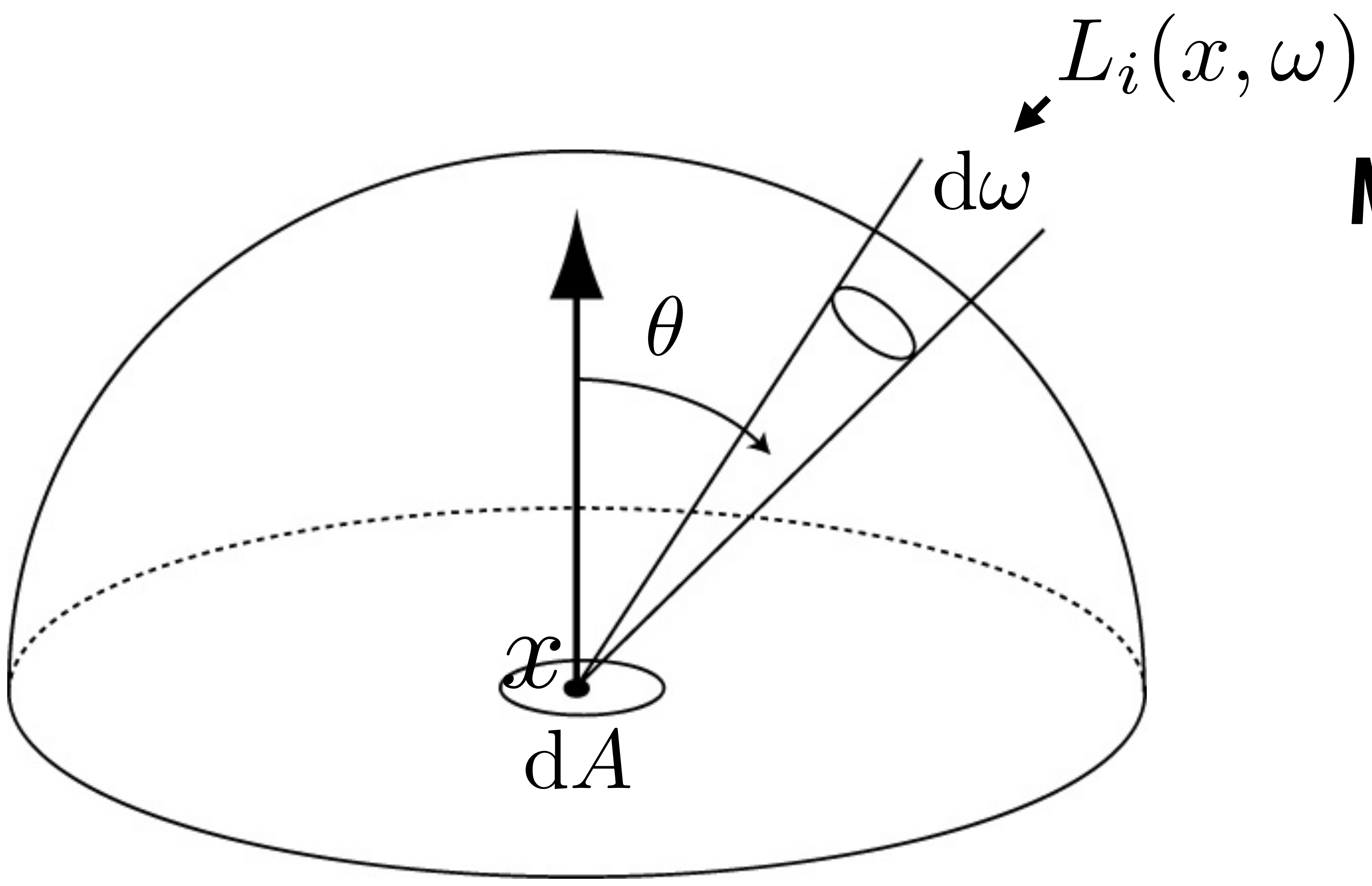
Occluder (blocks light)

Ground receives shadow

Monte Carlo integration applied to illumination (hemisphere sampling)

$$E(x) = \int_{H^2} L_i(x,\omega)\cos\theta \, \mathrm{d}\omega$$

We want to estimate this integral
(total incident irradiance at surface point x)



Monte Carlo estimator:

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

We sample directions (aka rays) uniformly from the hemisphere of directions
(a ray direction is a random variable)

$$Y_i = f(X_i)$$

$$Y_i = L_i(x,\omega_i)\cos\theta_i$$

For each ray we compute the incident differential irradiance.

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

We average all these samples, and scale by the size of the domain we are sampling from.
(The hemisphere has 2π steradians)

Then the expected value of the result is the value of the integral.

Direct lighting estimate

Uniformly-sample hemisphere of directions with respect to solid angle

$$E(\mathrm{x}) = \int_{H^2} L(\mathrm{x}, \omega) \cos\theta \, d\omega$$

Given surface point *x*:

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

For each of N samples:

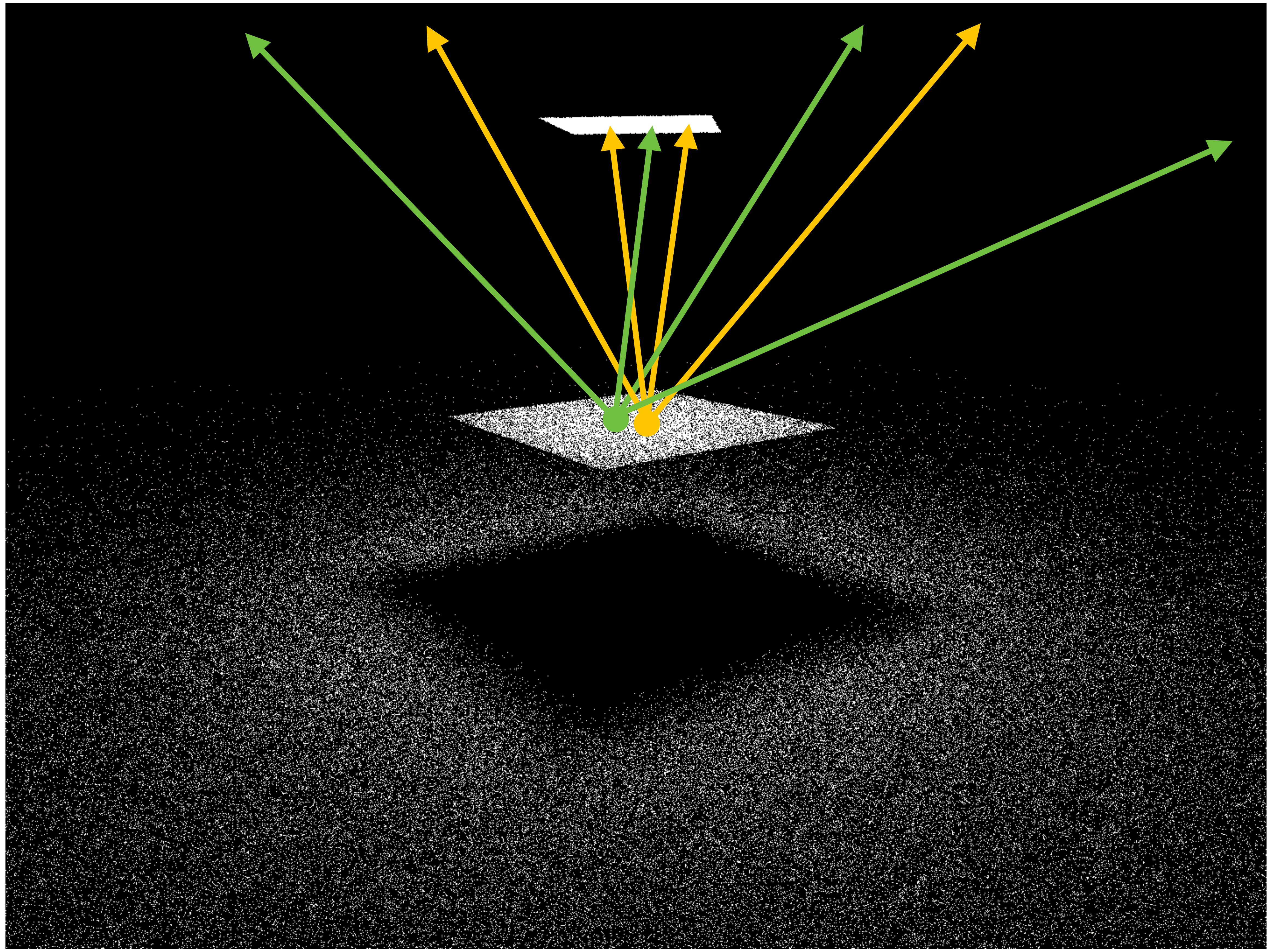
- **Generate random direction:** ω_i **(from uniform distribution over hemisphere) ***
- **Compute incoming radiance arriving** L_i **at *p* from direction:** ω_i
- **Compute incident irradiance due to ray:** $dE_i = L_i cos\,\theta_i$
- **Accumulate** $\frac{2\pi}{N} dE_i$ **into estimator**

*** We will relax the uniform probability restriction soon…**

Direct lighting: hemisphere sampling

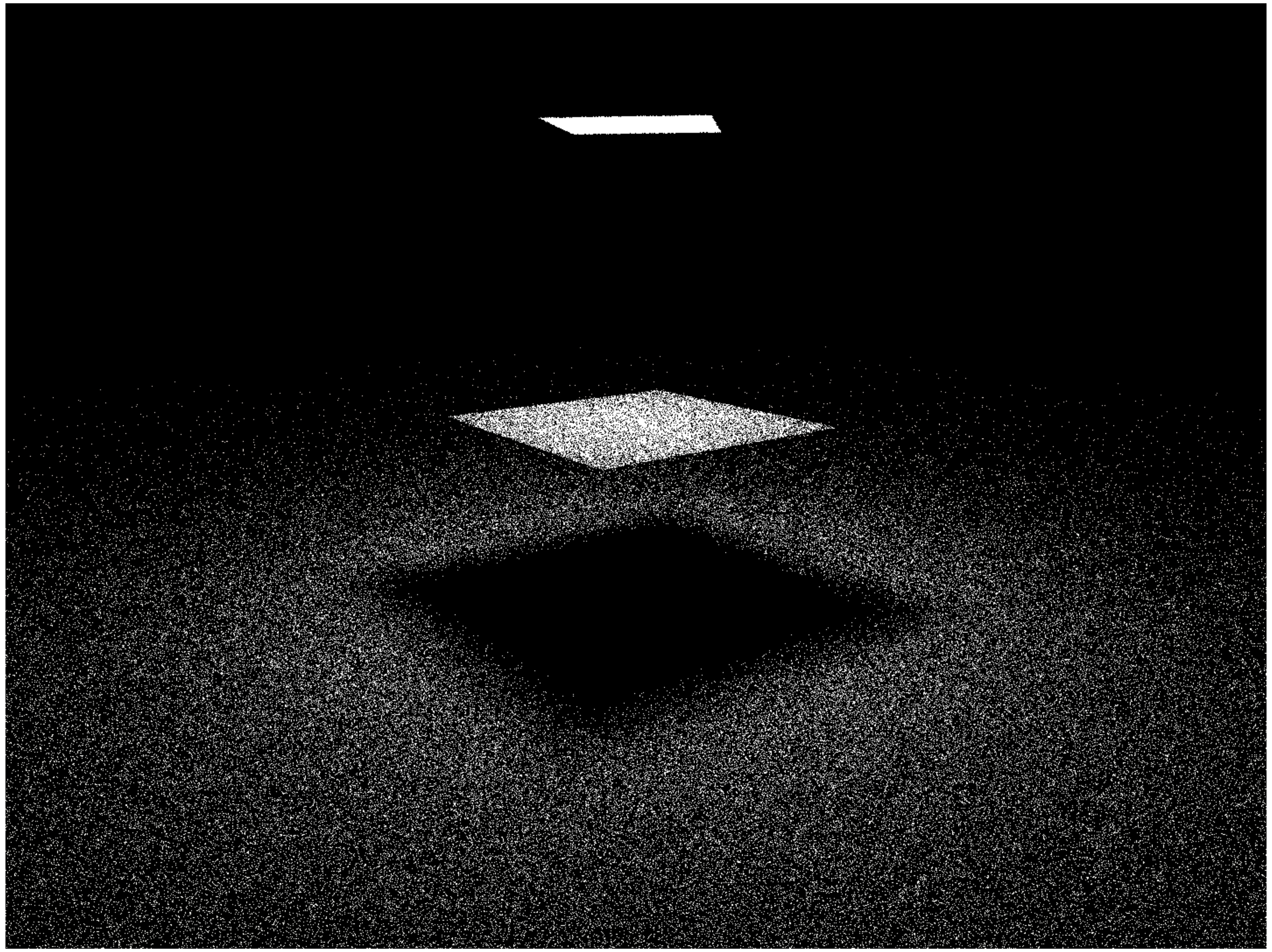
Incident lighting estimator uses random directions when computing incident lighting for different points. Some of those directions hit the light (and contribute illumination, some do not)

(The estimator is a random variable!)



16 light samples
(Uniformly sampled from hemisphere)

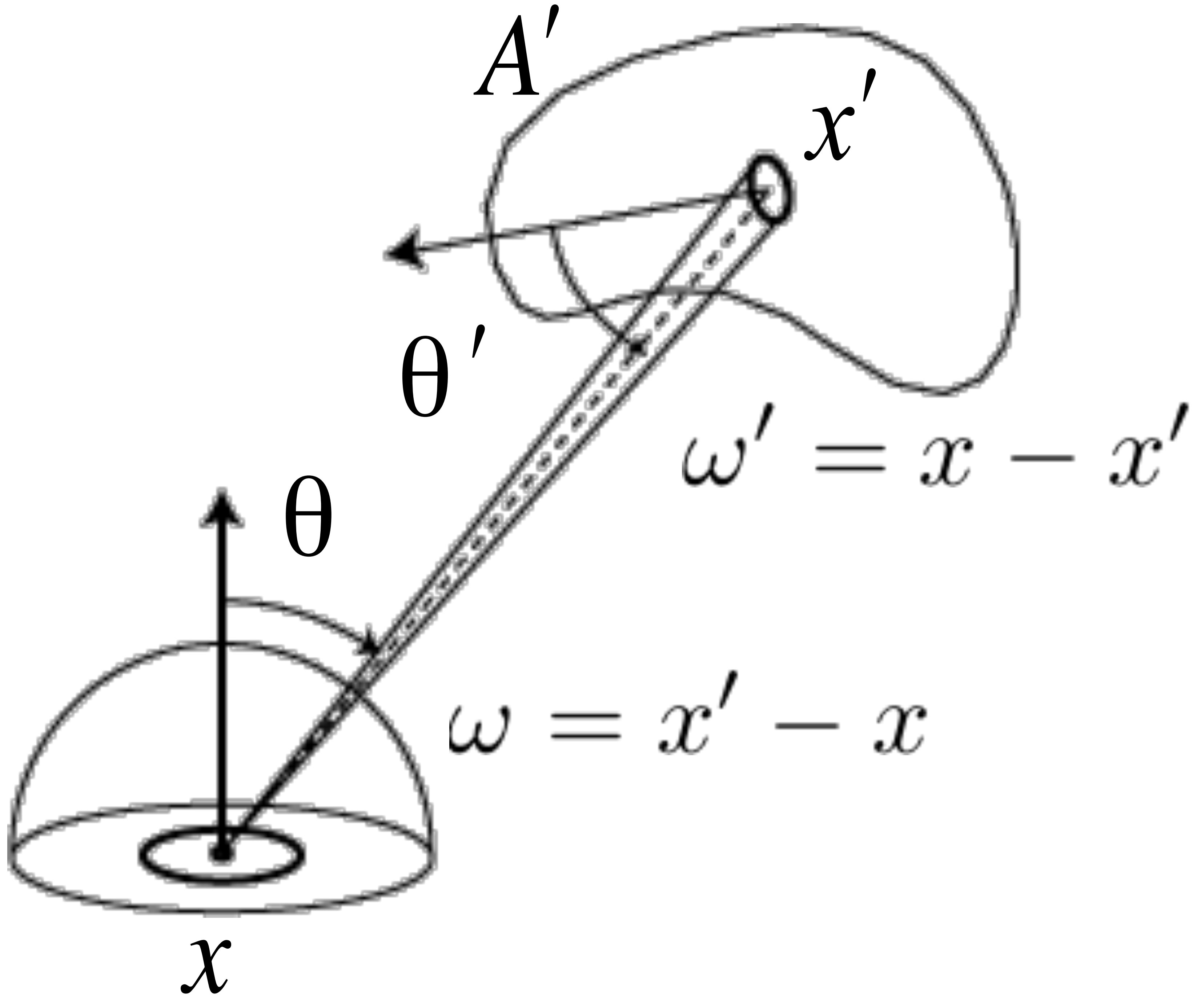
Back to the Monte Carlo integration story…



Now let's reparameterize the integral as integration over the light

$$E(x) = \int_{H^2} L_i(x,\omega)\, \cos\theta \, d\omega = \int_{A'} L \frac{\cos\theta \cos\theta'}{|x - x'|^2} dA'$$

← We want to estimate this integral (total incident irradiance at surface point x)



Monte Carlo estimator:

$$X_i \sim p(x') = \frac{1}{A'}$$

← We sample points on the light source uniformly with respect to area (a point on the light is a random variable)

$$Y_i = f(X_i)$$

$$Y_i = L \frac{\cos\theta \cos\theta'}{|x - x'|^2}$$

← We compute the incident differential irradiance from the sampled point on the light to surface point x.

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

← We average all these samples, and scale by the size of the domain we are sampling from. (The light has area A')

Then the expected value of the result is the value of the integral. →

* Assume area light emits radiance L from all directions from all points on surface.

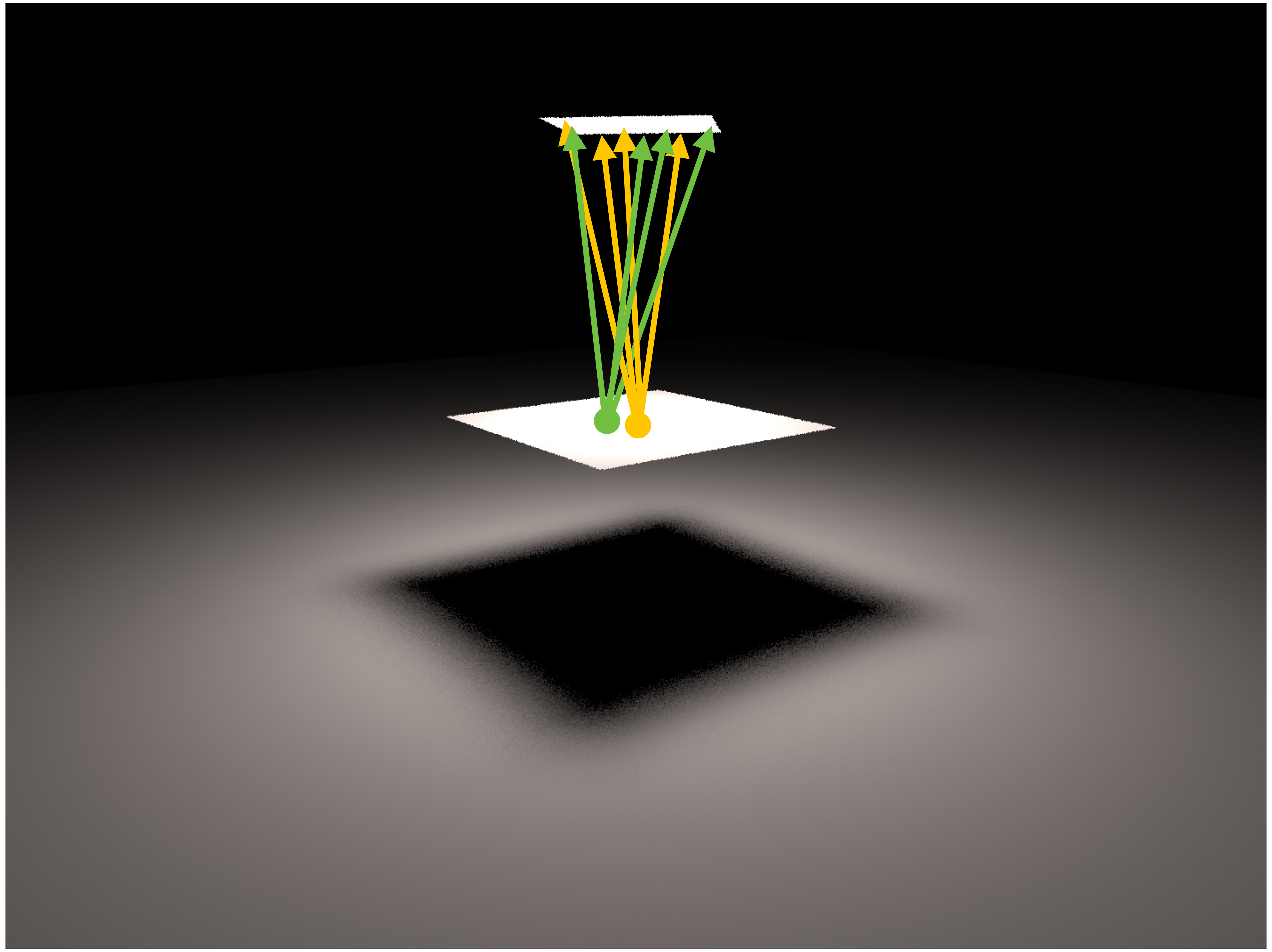
Direct lighting estimate (area sampling light with area A')

Given surface point *x*

For each sample i of N samples:

- **Generate random point x' on area light, compute direction from x to x':** ω_i
- **Compute incident irradiance due to ray from x' to x as** $dE_i = L\frac{\cos\theta\cos\theta'}{|x-x'|^2}$
- **Accumulate** $\frac{A'}{N}dE_i$ **into estimator**

Direct lighting: area sampling

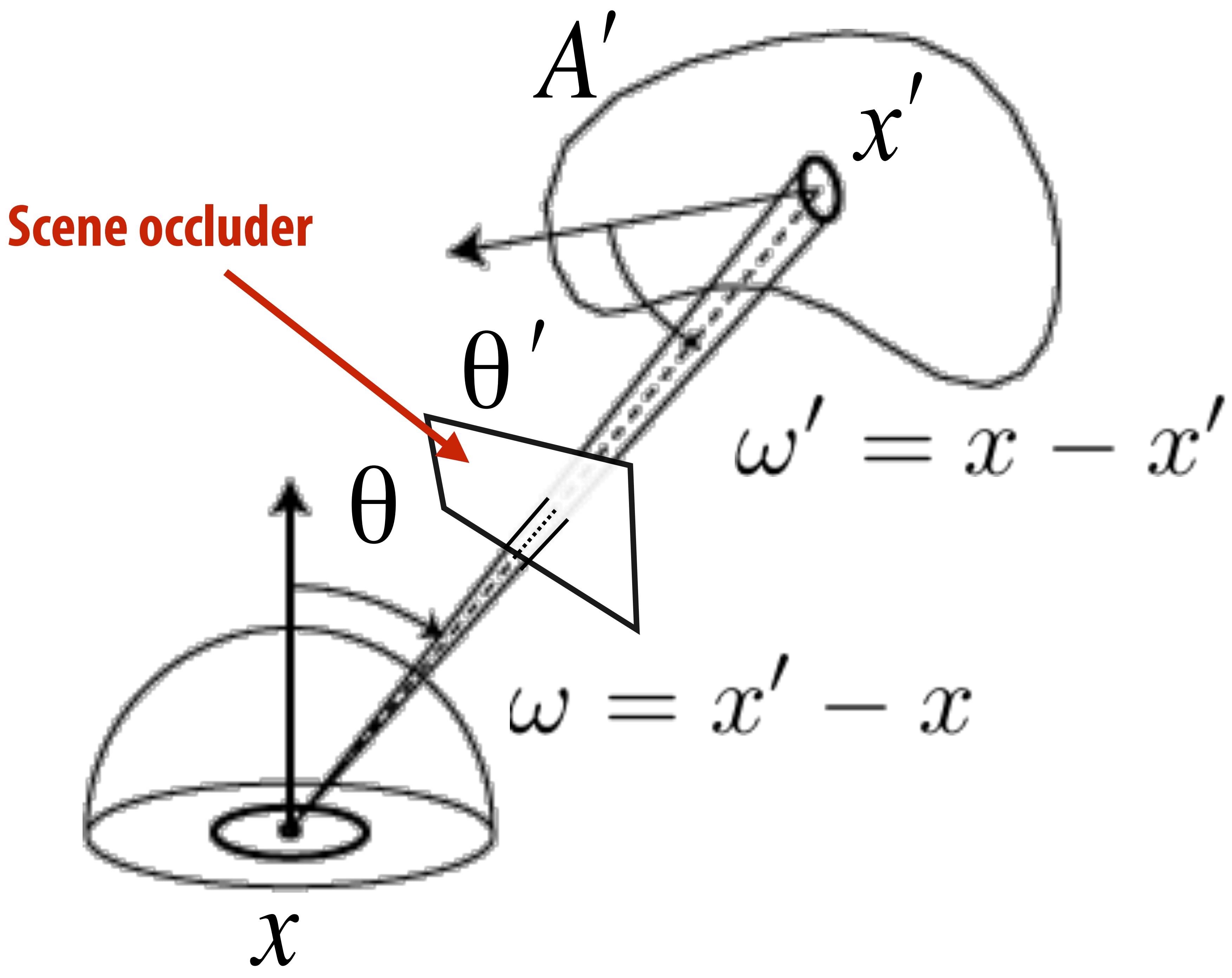


16 light samples (16 shadow rays)

Wait… how do we compute the shadows in this photo?

Shadowed light area sampling

$$E(x) = \int_{A'} V(x, x') L \frac{\cos\theta \cos\theta'}{|x - x'|^2} dA'$$



Note new: visibility term:

$$V(x, x') = \begin{cases} 0 & \text{blocked} \\ 1 & \text{visible} \end{cases}$$

Monte Carlo estimator:

$$X_i \sim p(x') = \frac{1}{A'}$$

$$Y_i = f(X_i)$$

$$Y_i = V(x, x') L \frac{\cos\theta \cos\theta'}{|x - x'|^2}$$

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

* Assume area light emits radiance L from all directions from all points on surface.

Direct lighting estimate (area sampling light with area A')

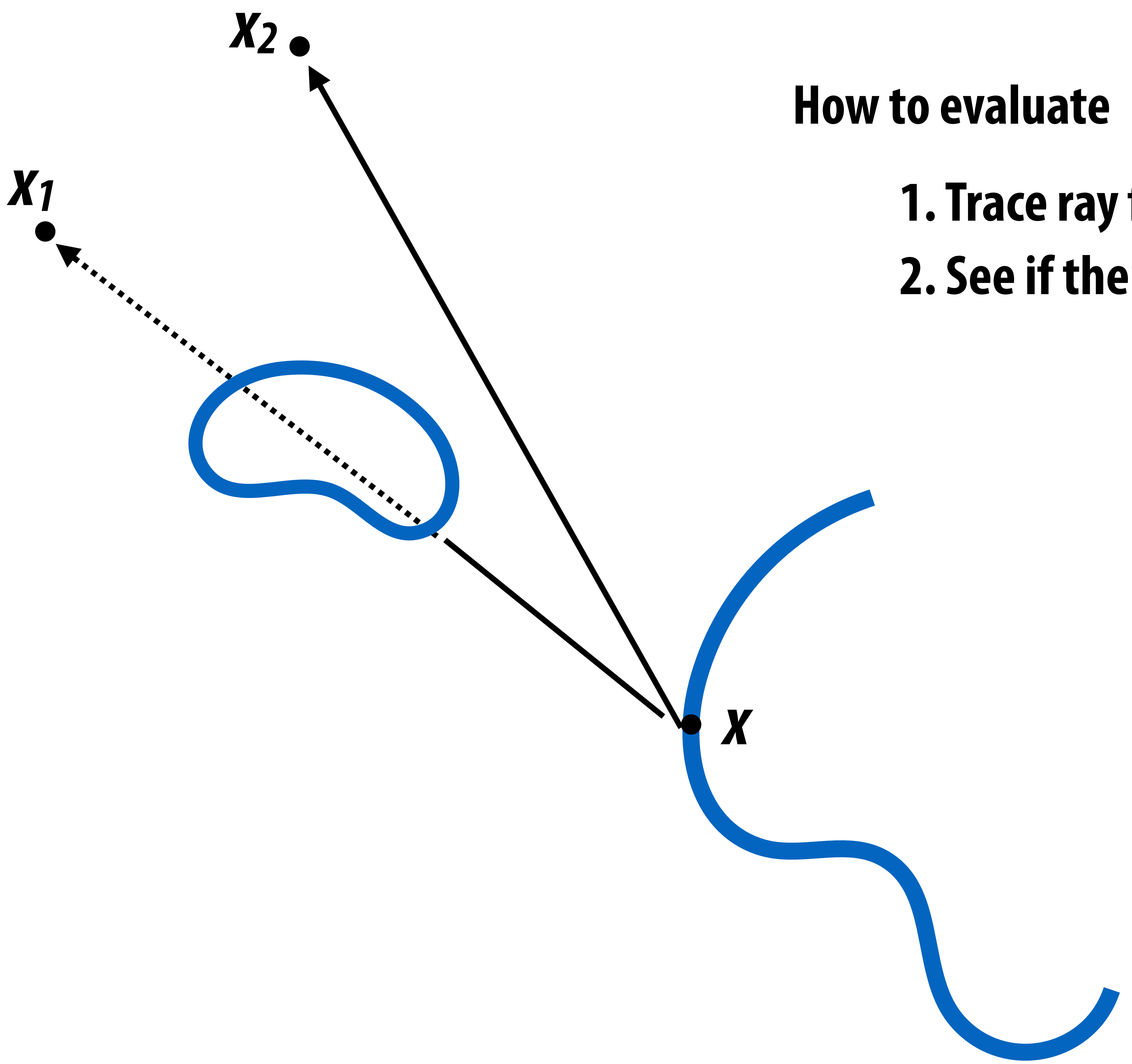
Given surface point *x*

For each of N samples:

- **Generate random point x' on area light, compute direction from x to x':** ω_i
- **Compute incident irradiance due to ray from x' to x as** $dE_i = L_o(x', -w_i)V(x, x')\frac{\cos\theta_i \cos\theta_o}{|x - x'|^2}$
- **Accumulate** $\frac{A'}{N} dE_i$ **into estimator**

How do you evaluate V()?

How to compute if point is visible from another point?



How to evaluate $V(x, x')$ using ray tracing:

1. Trace ray from x toward x'
2. See if there is any hit with scene geometry closer to x than $|x - x'|$

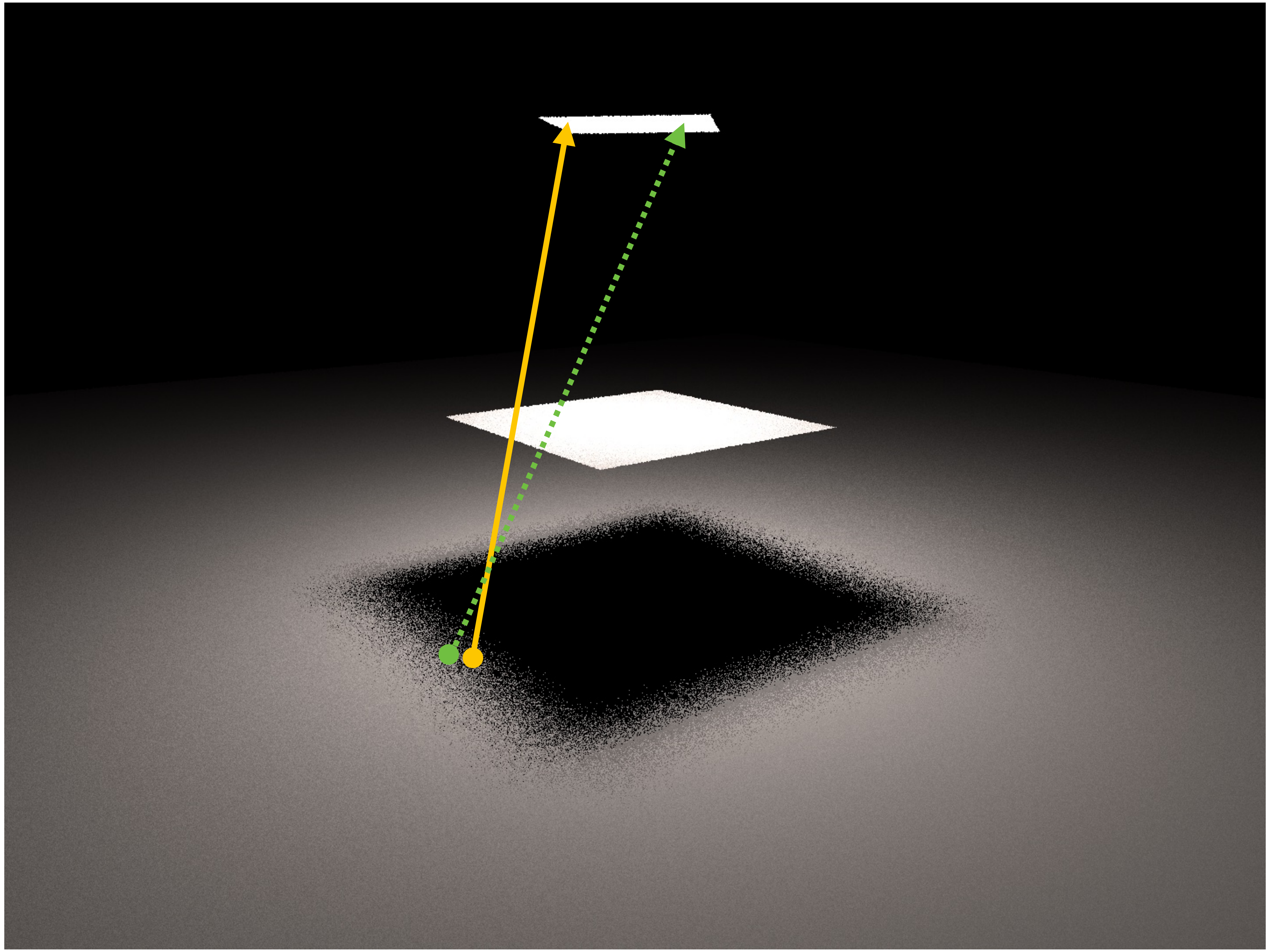
Shadowed direct lighting estimate (area sampling light with area A')

Given surface point *x*

For each sample *i* of N samples:

- **Generate random point x' on area light, compute direction from x to x':** ω_i
- **Compute incident irradiance due to ray from x' to x as** $dE_i = L\frac{\cos\theta\cos\theta'}{|x - x'|^2}$
- **Trace shadow ray from x in direction** ω_i **.**
- **If shadow ray does not hit geometry before x', accumulate** $\frac{A'}{N}dE_i$ **into estimator**

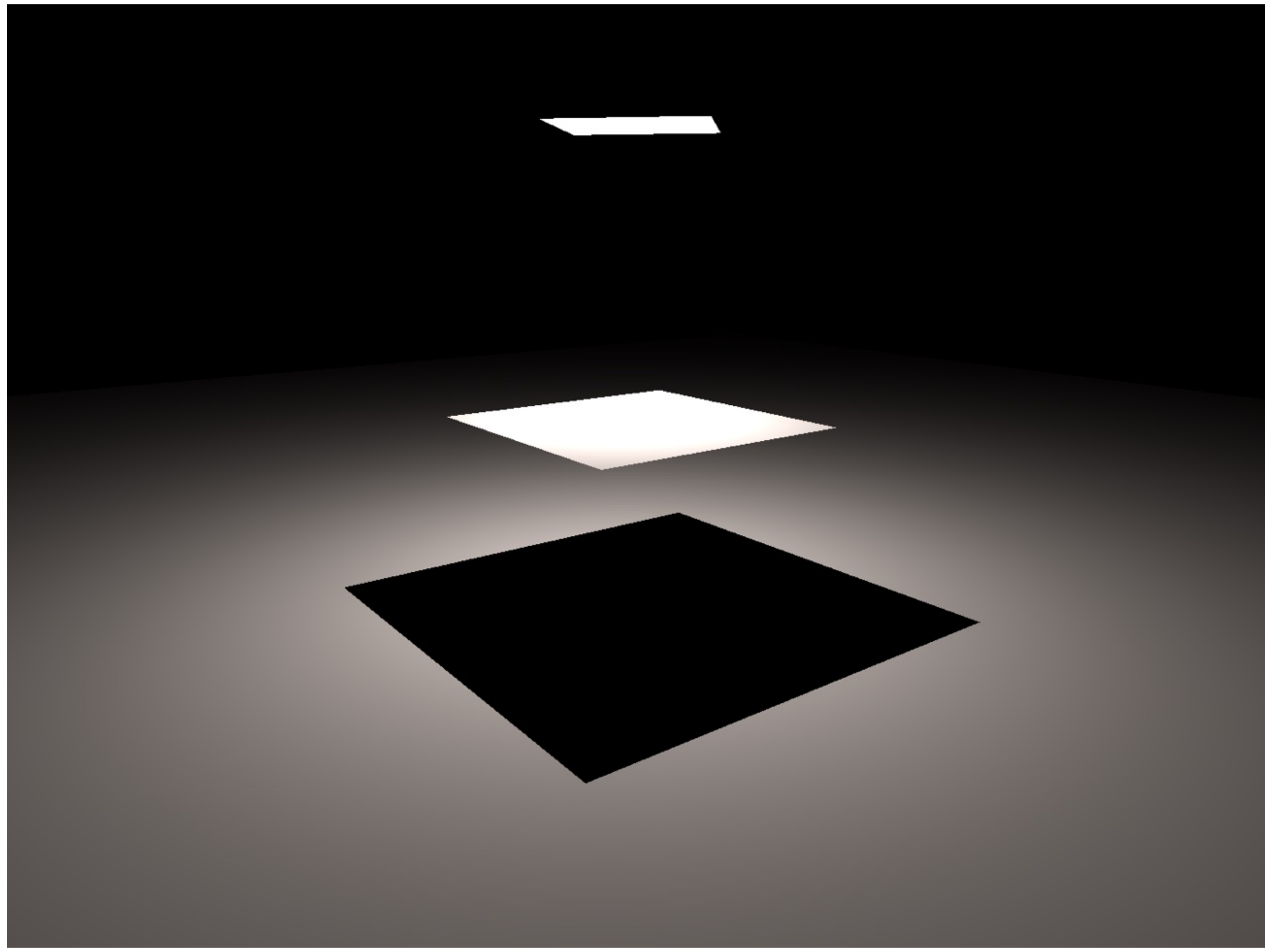
Random sampling introduces noise



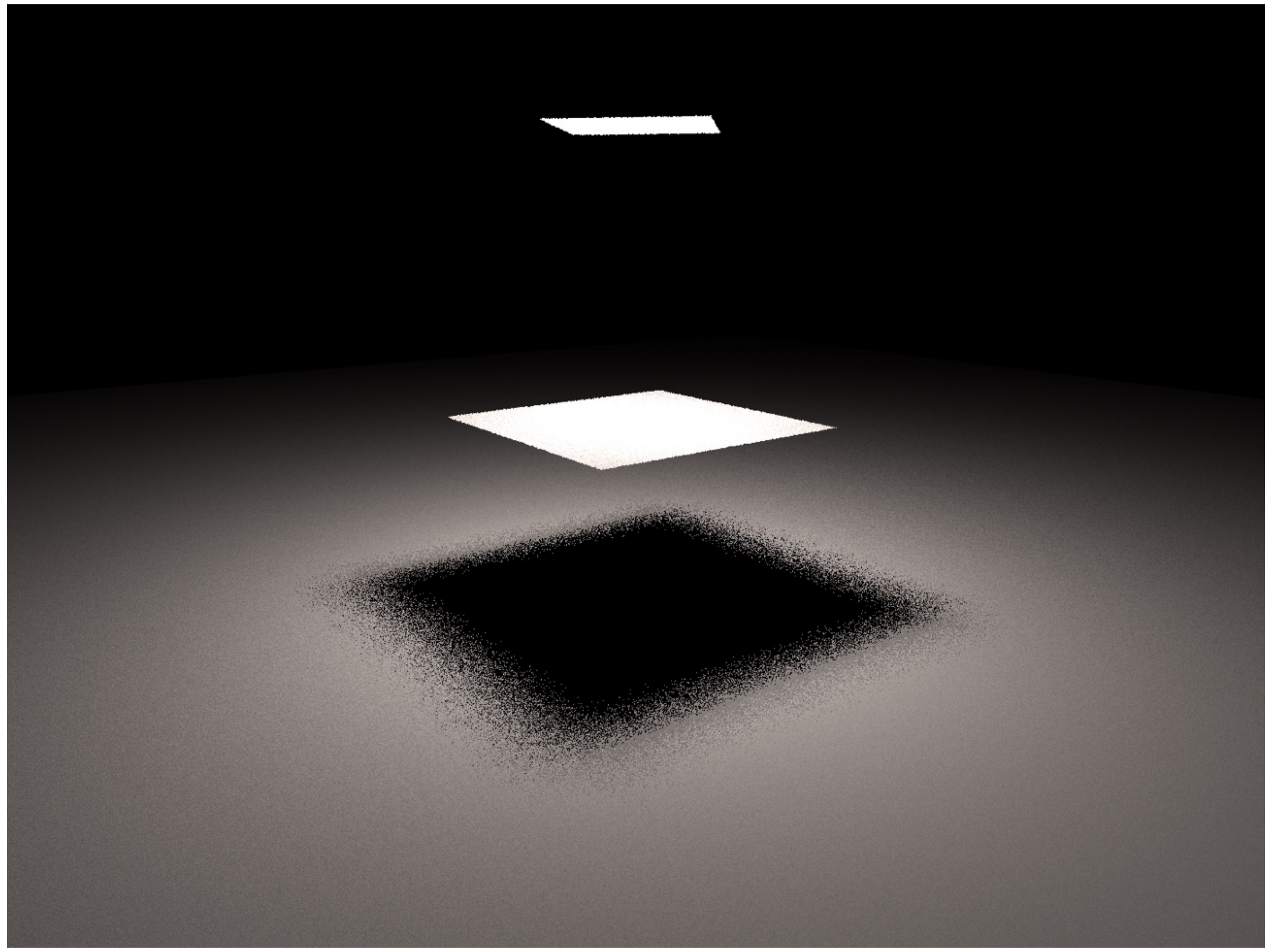
Incident lighting estimator uses different random directions when computing incident lighting for different points. Some of those directions are occluded, some are not!

(The estimator is a random variable!)

Random sampling introduces noise



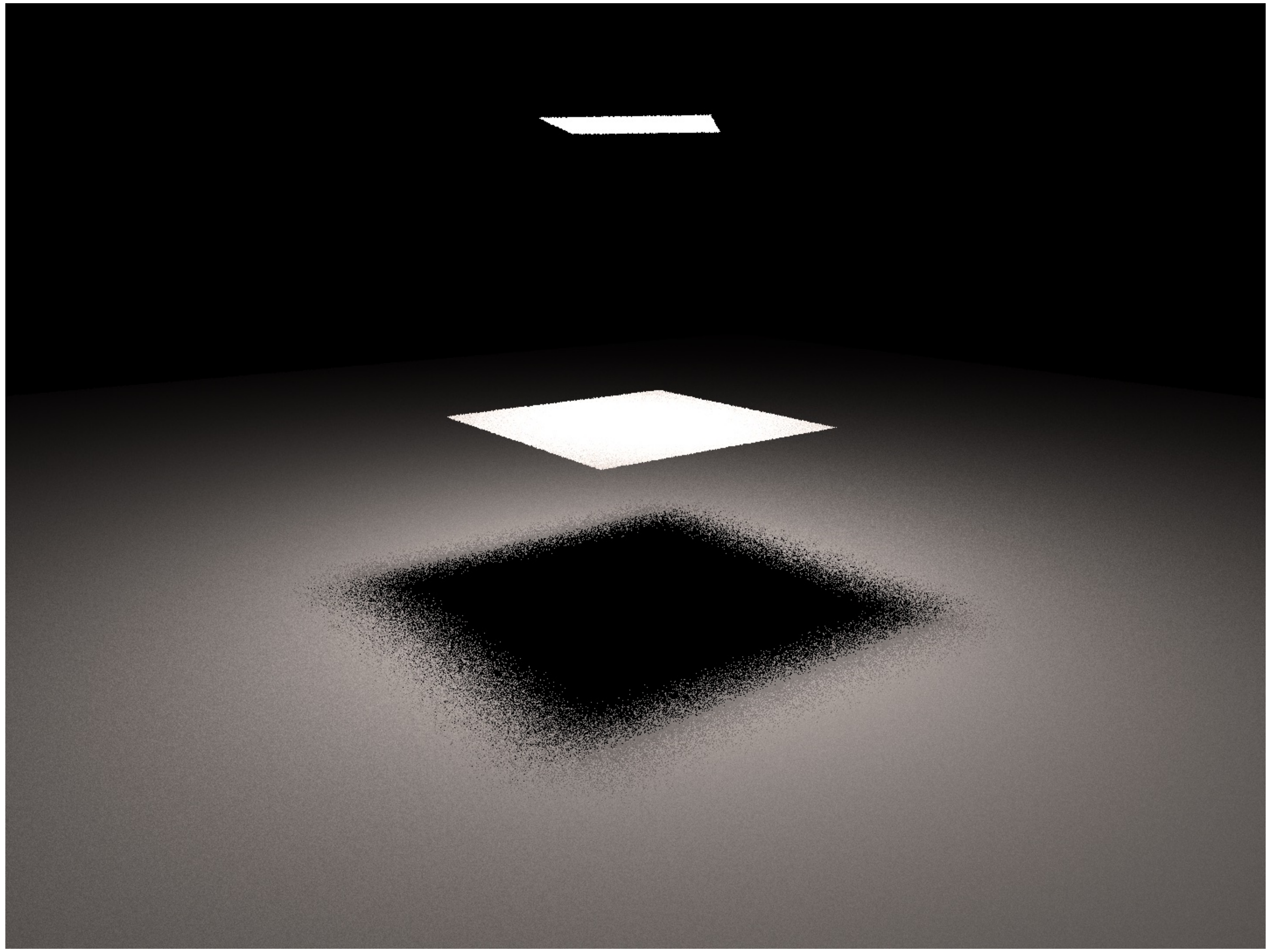
Always sample light center



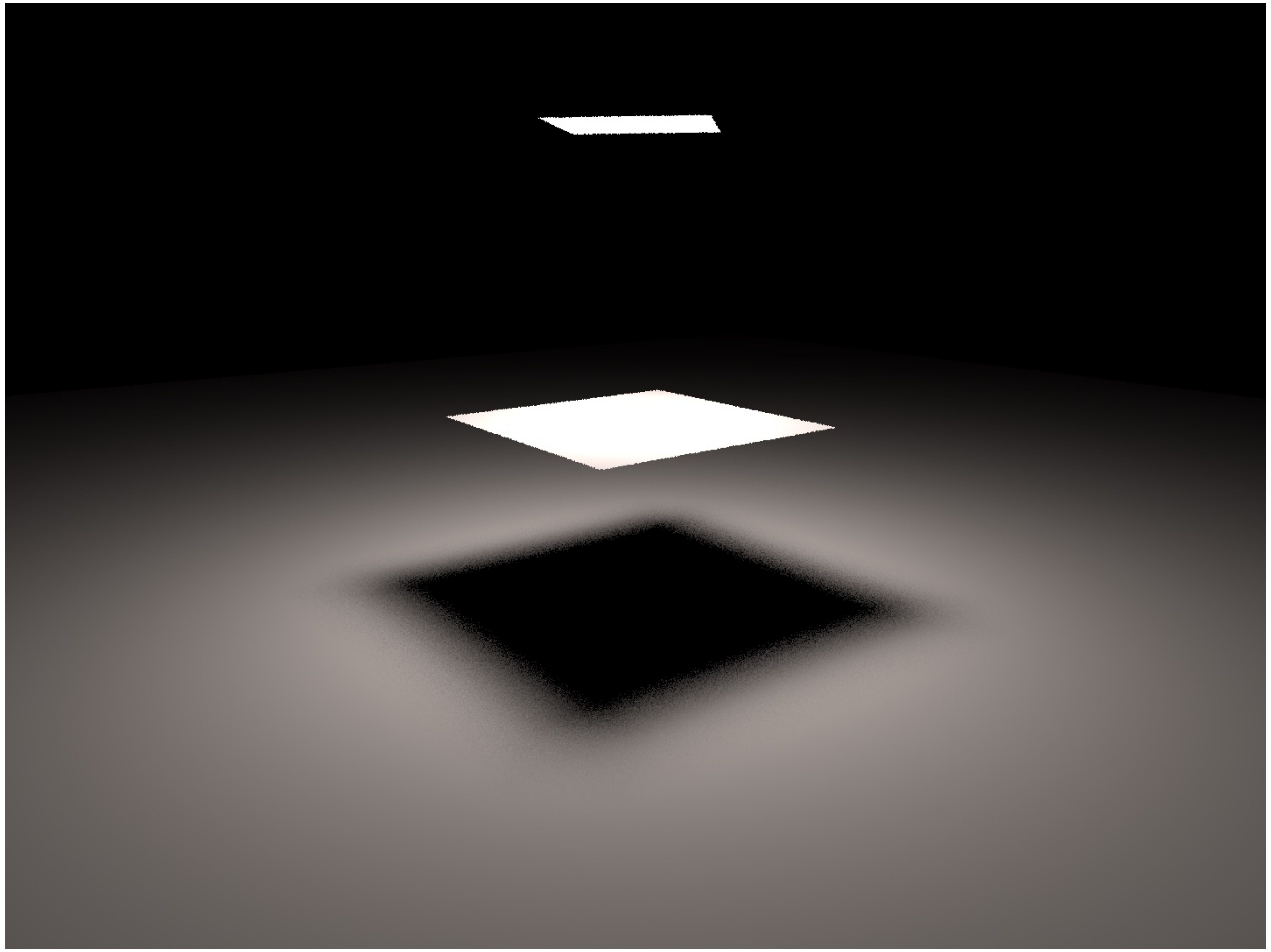
Random area sampling

1 shadow ray per eye ray

Quality improves with more samples (more shadow rays)

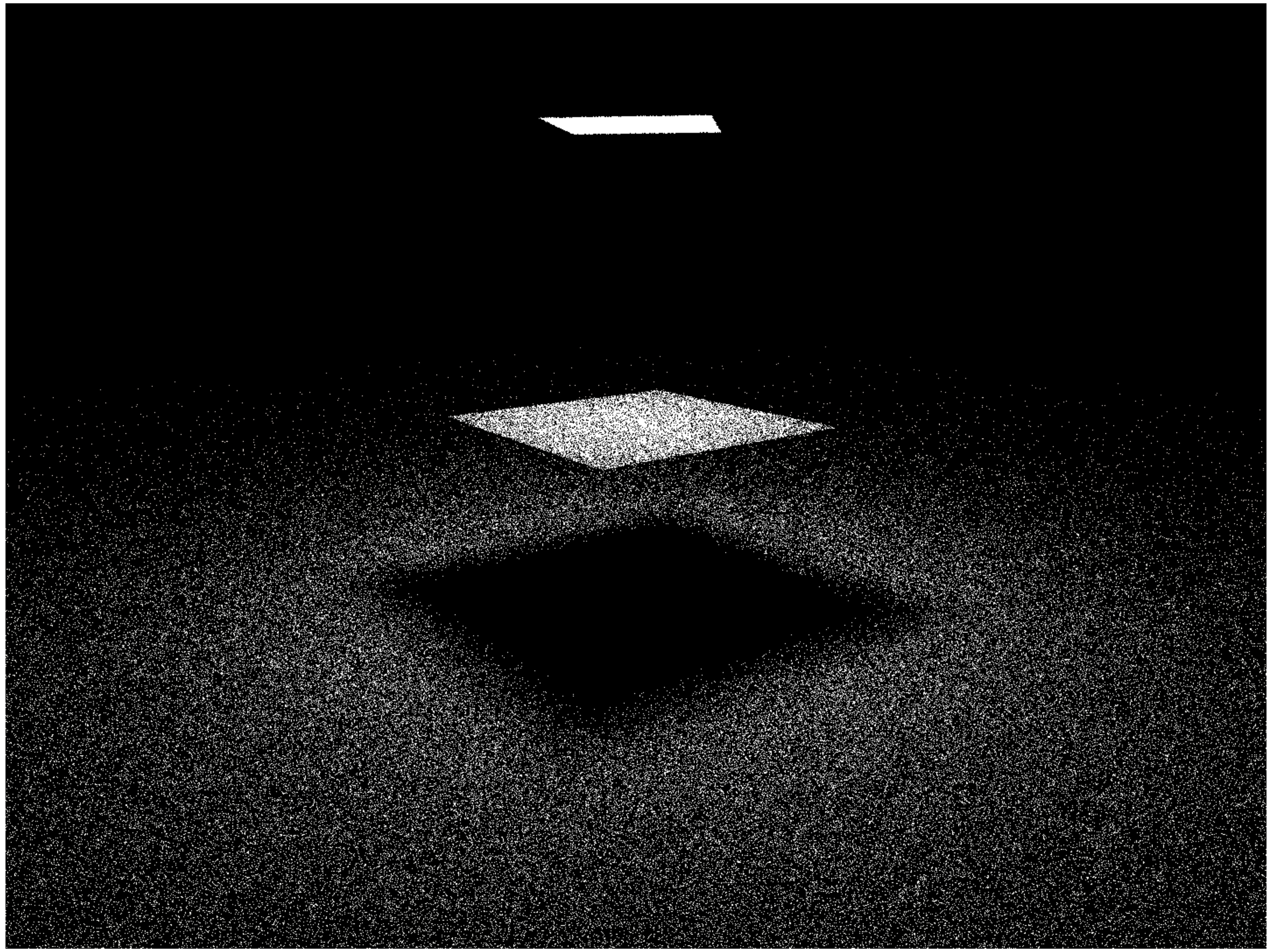


Uniform area sampling, 1 shadow ray

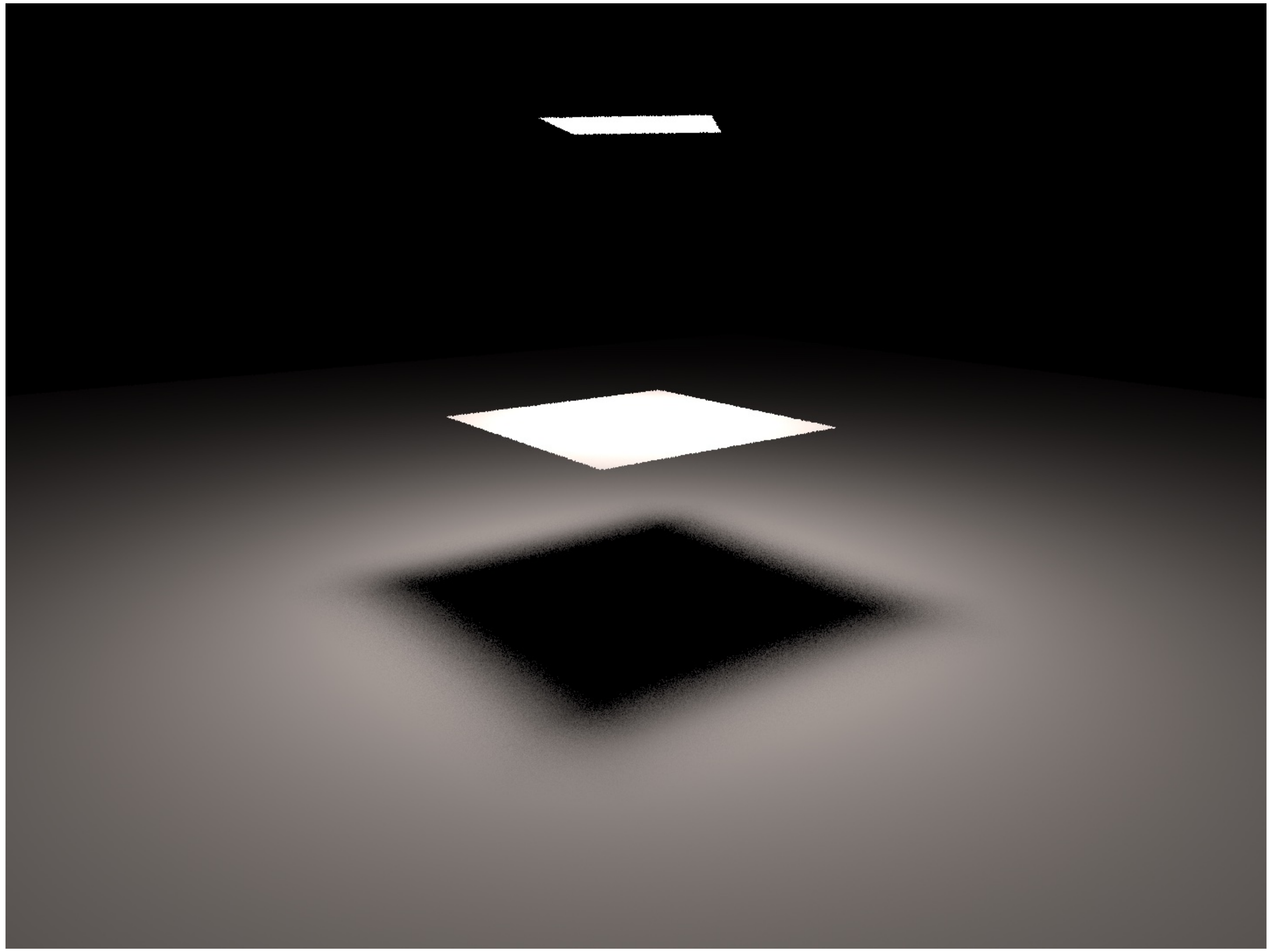


Uniform area sampling, 16 shadow rays

Why is area sampling better than hemisphere sampling?



Uniform hemisphere sampling
16 shadow rays



Uniform area sampling
16 shadow rays

Variance

- **Definition**

Variance is expected squared deviation from mean

$$V[Y] = E[(Y - E[Y])^2]$$
$$= E[Y^2] - E[Y]^2$$

- **Variance decreases linearly with number of samples**

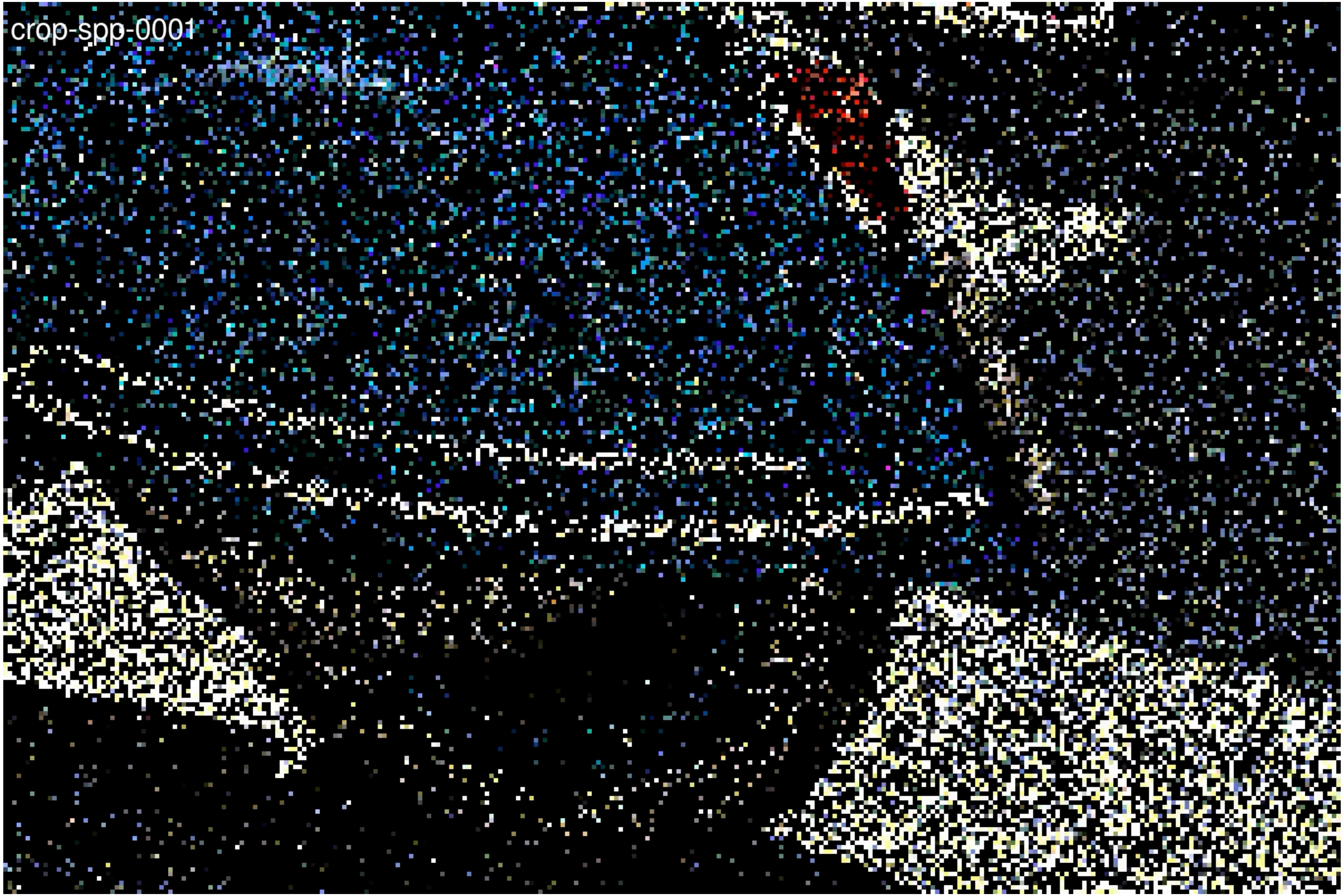
$$V\left[\frac{1}{N}\sum_{i=1}^{N} Y_i\right] = \frac{1}{N^2}\sum_{i=1}^{N} V[Y_i] = \frac{1}{N^2} N\, V[Y] = \frac{1}{N} V[Y]$$

Properties of variance:

$$V\left[\sum_{i=1}^{N} Y_i\right] = \sum_{i=1}^{N} V[Y_i]$$

$$V[aY] = a^2\, V[Y]$$

Samples vs. error



For video see: http://pharr.org/matt/assets/bistro-spp.mp4

“Biasing” sample selection

- We previously used a uniform probability distribution to generate samples in our estimator
  - Example 1: uniform over directions, Example 2: uniform over light surface area
- Idea: change the distribution–bias the selection of samples

$$X_i \sim p(x)$$

- However, for estimator to remain unbiased, must change the estimator to:

$$Y_i = \frac{f(X_i)}{p(X_i)}$$

- Note: “biasing” selection of random samples is different than creating a biased estimator
  - Biased estimator: expected value of estimator does not equal integral it is designed to estimate (not good!)

General unbiased Monte Carlo estimator

$$E[F_N] = E\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}\right] = \int_a^b f(x)\,\mathrm{d}x$$

$$X_i \sim p(x)$$

Consider the special case where X_i is drawn from uniform distribution:

$$F_N = \frac{b-a}{N}\sum_{i=1}^{N} f(X_i)$$

$$X_i \sim U(a,b)$$

$$p(x) = \frac{1}{b-a}$$

Biased sample selection, but unbiased estimator…

$$
\begin{aligned}
E[F_N] &= E\left[\frac{1}{N}\sum_{i=1}^{N} Y_i\right] \\
&= \frac{1}{N}\sum_{i=1}^{N} E[Y_i] = \frac{1}{N}\sum_{i=1}^{N} E\left[\frac{f(X_i)}{p(X_i)}\right] \\
&= \frac{1}{N}\sum_{i=1}^{N} \int_a^b \frac{f(x)}{p(x)}\, p(x)\mathrm{d}x \\
&= \frac{1}{N}\sum_{i=1}^{N} \int_a^b f(x)\,\mathrm{d}x \\
&= \int_a^b f(x)\,\mathrm{d}x
\end{aligned}
$$

The sequence above boils down to…

$$
\begin{aligned}
E[Y_i] &= \int_a^b \frac{f(x)}{p(x)}\, p(x)\mathrm{d}x \\
&= \int_a^b f(x)\mathrm{d}x
\end{aligned}
$$

Properties of expectation:

$$E\left[\sum_i Y_i\right] = \sum_i E[Y_i]$$

$$E[aY] = aE[Y]$$

**Note: Even though my notation suggests this is integration over 1D domain [a,b],
this proof holds for any integration domain, such as the 2D hemisphere of directions on the previous slide.**

Importance sampling

Idea: bias selection of samples towards parts of domain where function we are integrating is large ("the most useful samples")

Draw samples according to magnitude of f(x)

$$\tilde{p}(x) = cf(x)$$ ← Normalization to make a pdf

$$c = \frac{1}{\int f(x)dx}$$

$$\tilde{f}(x) = \frac{f(x)}{\tilde{p}(x)}$$ ← Generalized MC estimator

Recall definition of variance

$$V[f] = E[f^2] - E^2[f]$$

$$\begin{aligned} E[\tilde{f}^2] &= \int \left[\frac{f(x)}{p(x)}\right]^2 p(x)\,\mathrm{d}x \\ &= \int \left[\frac{f(x)}{cf(x)}\right]^2 cf(x)\,\mathrm{d}x \\ &= \int \frac{f(x)}{c}\,\mathrm{d}x \\ &= \frac{1}{c}\int f(x)\,\mathrm{d}x \\ &= \frac{1}{c^2} \\ &= \left[\int f(x)\mathrm{d}x\right]^2 \end{aligned}$$

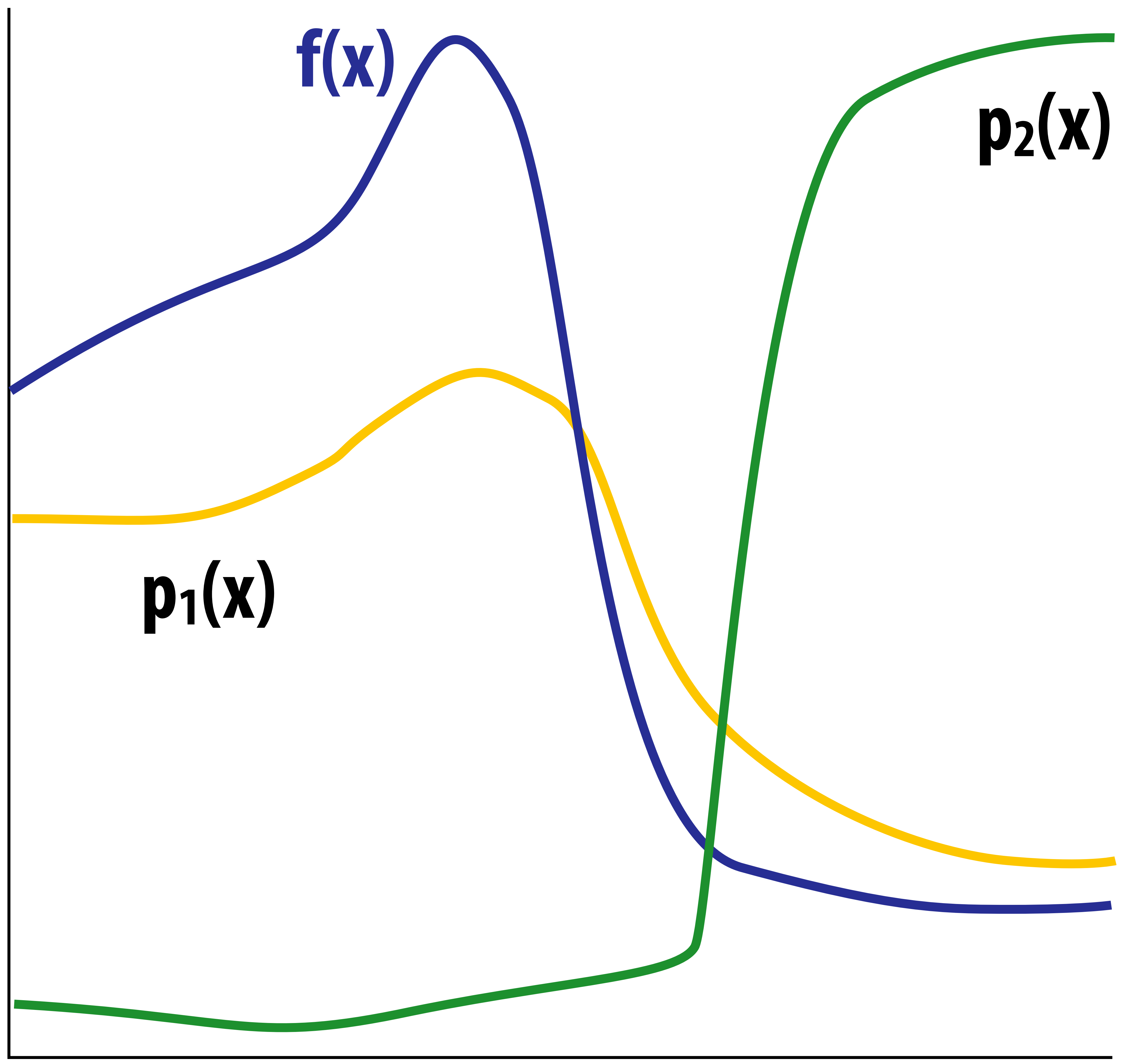
$$\begin{aligned} E^2[\tilde{f}] &= \left[\int \left[\frac{f(x)}{cf(x)}\right] \tilde{p}(x)\,\mathrm{d}x\right]^2 \\ &= \left[\int \left[\frac{f(x)}{cf(x)}\right] c\,p(x)\,\mathrm{d}x\right]^2 \\ &= \left[\int f(x)dx\right]^2 \end{aligned}$$

Zero variance!

$$V[\tilde{f}] = E[\tilde{f}^2] - E^2[\tilde{f}] = 0$$

What's the gotcha??

Effect of sampling distribution "Fit"



What is the behavior of $f(x)/p_1(x)$? $f(x)/p_2(x)$?

How does this impact the variance of the estimator?

Environment map light sources

Texture(u,v) defines incoming radiance from a direction:

$$L(\omega) = L(\phi, \theta)$$

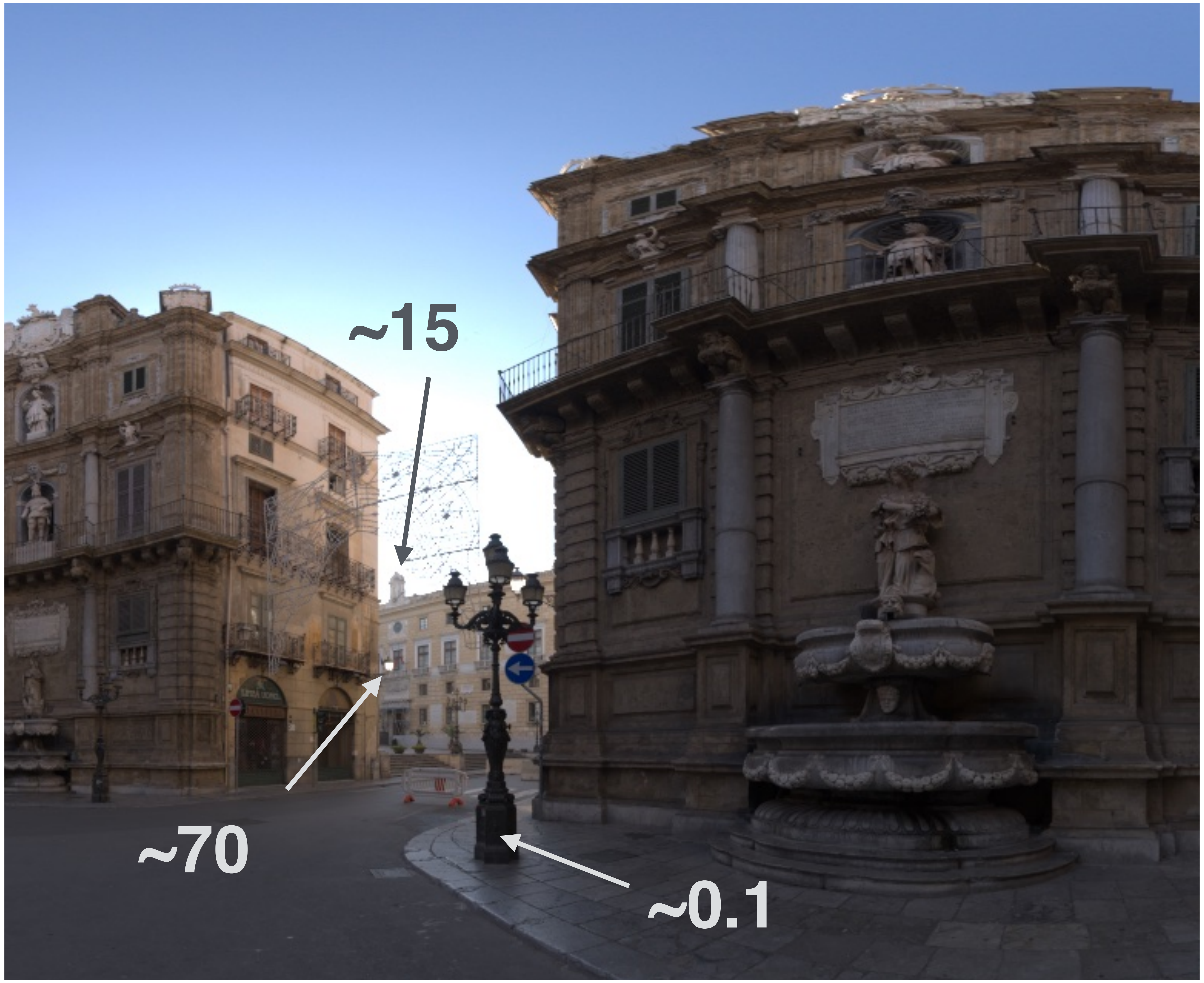
Capturing an environment map

[ILM 2009]

Consider this environment map

HDRI Haven: Quattro Canti

Real world lighting: large differences in incoming radiance



HDRI Haven: Quattro Canti

Importance sampling environment map lights

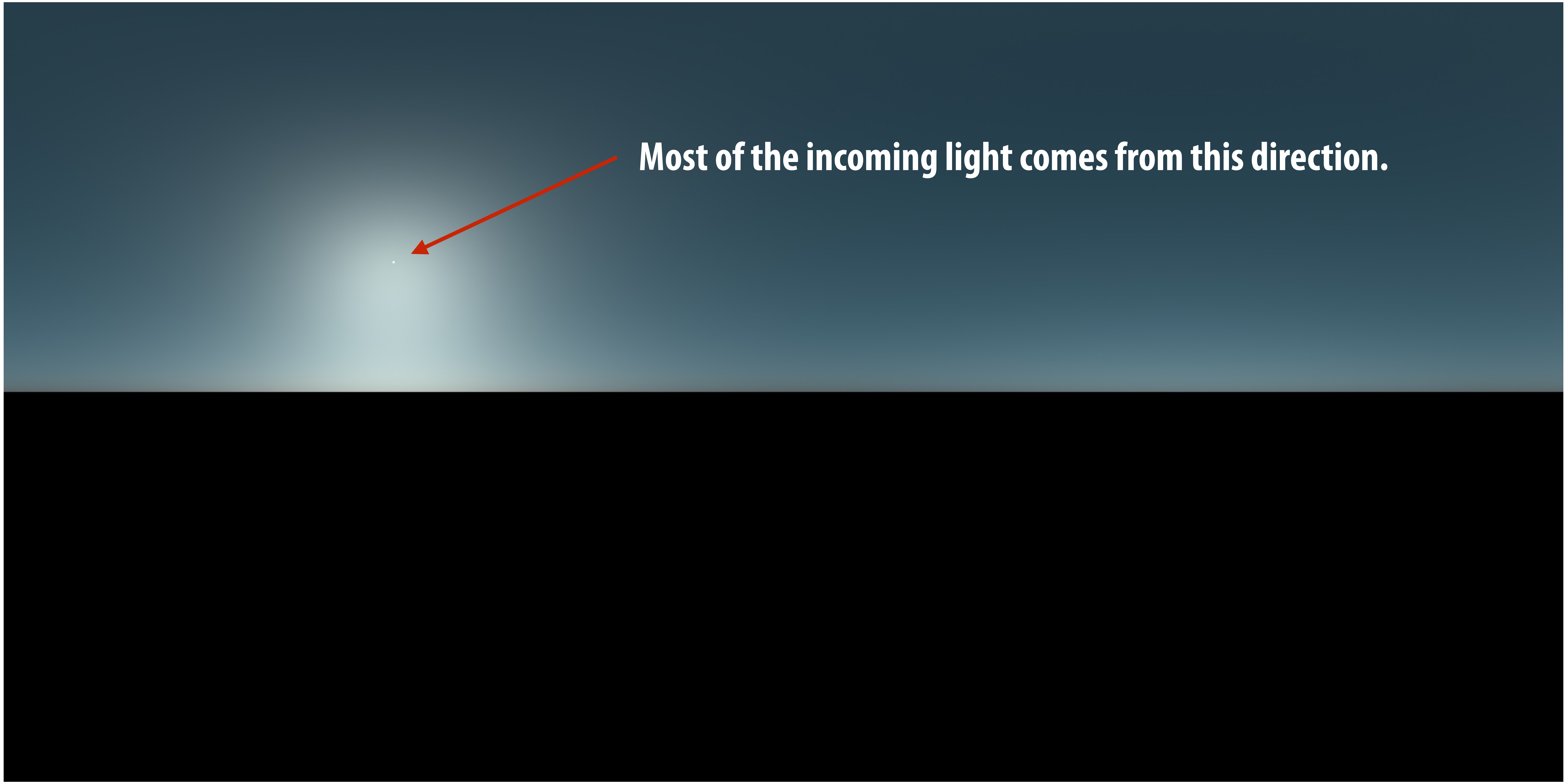
Idea: sampling incident lighting directions proportional to luminance
(prioritize directions that contribute the most)

Luminance map

HDRI Haven: Quattro Canti

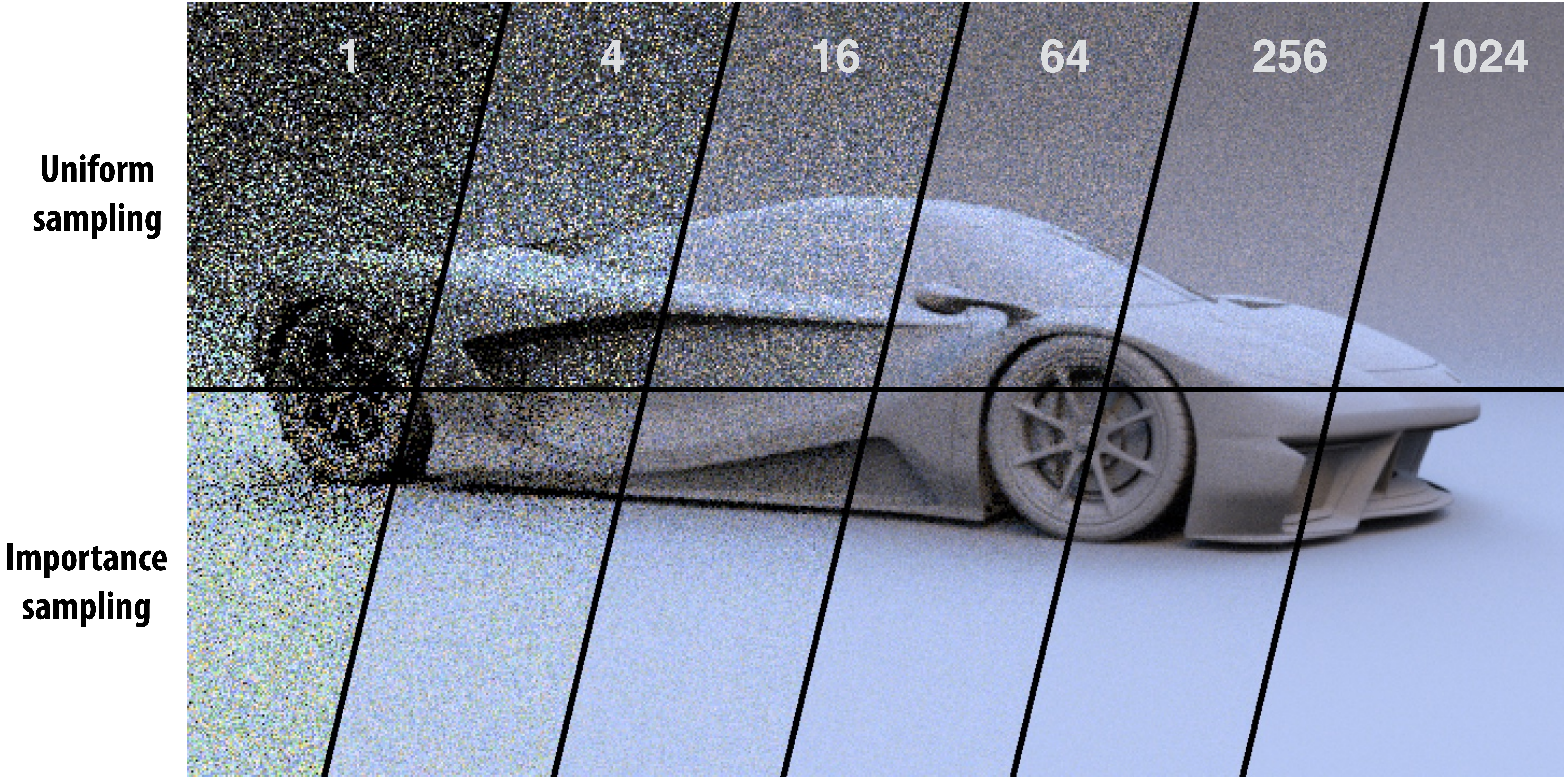
Sky environment map with a bright sun

$\theta \in [0, \pi]$



$\phi \in [0, 2\pi]$

Uniform vs. importance sampling the environment light



Comparing different techniques

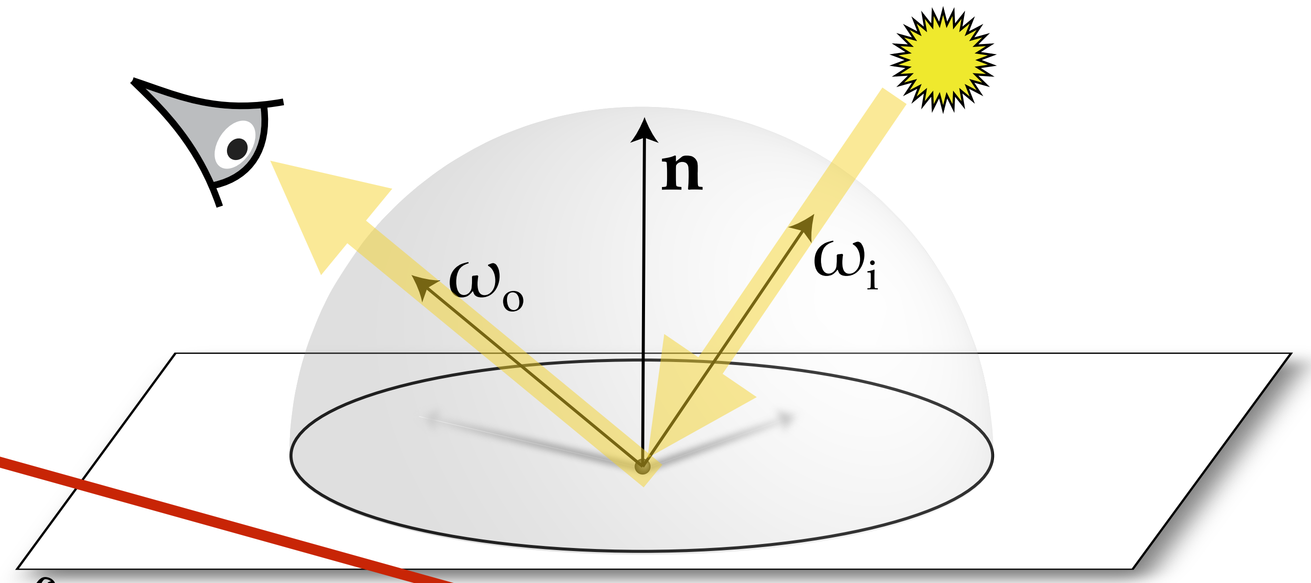
- **Variance in an estimator manifests as noise in rendered images**
- **Estimator efficiency measure:**

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

- **If one integration technique has twice the variance as 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**

Putting it all together: reflectance due to direct lighting

Estimating reflectance off surface point x in direction ω_o due to incident illumination from multiple area light sources



Radiance from light l

Sum over lights

$$L_o(x,\omega) = \sum_l \int_{A'_l} f(\mathrm{x},\omega_i,\omega_o) L_{(l)}(\mathrm{x},\mathrm{x}') V(\mathrm{x},\mathrm{x}') \frac{\cos\theta\cos\theta'}{|\mathrm{x}-\mathrm{x}'|^2} dA'$$

dE_i

Given surface point *x*

For each area light *l*:

For each sample *i* of N samples:

Generate random point x' on light l according to $p(x)$ for light, compute direction from x to x': ω_i

Evaluate BRDF $f(\mathrm{x},\omega_i,\omega_o)$

Compute incident irradiance due to ray from x' to x: as dE_i

Trace shadow ray from x in direction ω_i.

If shadow ray does not hit geometry before x', accumulate $\frac{1}{N}\left[\frac{f(\mathrm{x},\omega_i,\omega_o)\,dE_i}{p(x')}\right]$ into estimator

Cost???

7M light sources

Pixar, Coco

Tens of thousands of lights...

[Bitterli et al. 2020]

Multiple light sources

- **Now we need a Monte Carlo estimate for a finite sum of terms** $\sum_{i=1}^{N} f_i$ **(not an integral)**

- **Define a discrete probability over terms)** p_i

$$\sum_i p_i = 1$$

- **Draw** N **samples** $j \sim p_i$

- **Estimator:**

$$\frac{1}{N} \sum \frac{f_j}{p_j}$$

Multiple light sources

- **Consider drawing a single sample:**
  - Draw one sample $j \sim p_i$ ← Choose a light
  - Compute f_j
  - Estimator: f_j / p_j

Incoming radiance from light j

$$f_j = \int f(\mathrm{x}, \omega_i, \omega_o)\, L_{\mathrm{i}(j)}(\omega_{\mathrm{i}})\, \cos\theta_{\mathrm{i}}\, \mathrm{d}\omega_{\mathrm{i}}$$

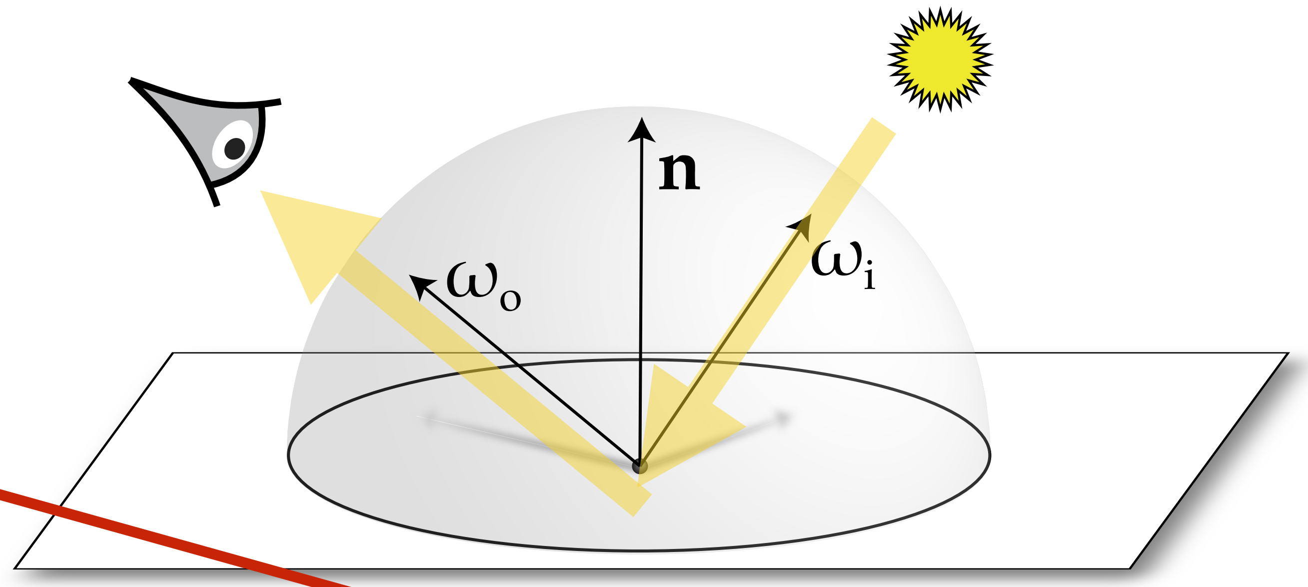
- **Expected value:**

$$E\left[\frac{f_j}{p_j}\right] = \sum_i p_i \frac{f_i}{p_i} = \sum_i f_i$$

- **What's a good discrete distribution p_i for choosing lights? (uniform?)**

Putting it all together: reflectance due to direct lighting

Estimating reflectance off surface point x in direction ω_o due to incident illumination from multiple area light sources



Given surface point *x*…

For all K chosen lights:

- Select area light *l* with probability p_l
- For each sample *i* of N samples:
  - Generate random point x' on light l according to $p(x)$ for light, compute direction from x to x': ω_i
  - Evaluate BRDF $f(\mathrm{x}, \omega_i, \omega_o)$
  - Compute incident irradiance due to ray from x' to x: as dE_i
  - Trace shadow ray from x in direction ω_i.
  - If shadow ray does not hit geometry before x', accumulate $\frac{1}{KN}\left[\frac{f(\mathrm{x}, \omega_i, \omega_o)\, dE_i}{p_l\, p(x')}\right]$ into estimator

Radiance from light *l*

Sum over only the sampled lights *l*

$$L_o(x,\omega) = \frac{1}{K}\sum_l \frac{1}{p_l}\int_{A'_l} f(\mathrm{x},\omega_i,\omega_o)\underbrace{L_{(l)}(\mathrm{x},\mathrm{x}')V(\mathrm{x},\mathrm{x}')\frac{\cos\theta\cos\theta'}{|\mathrm{x}-\mathrm{x}'|^2}dA'}_{dE_i}$$

Zero day scene (beeple@)
Very large number of lights

Uniform sampling (16 spp)

Choosing 16 lights (K=16, uniform probability across lights), tracing one ray to random point on each light (N=1)

Importance sampling: sampling lights proportional to light power (16 spp)

Choosing 16 lights (K=16, light probability proportional to its power), tracing one ray to random point on each light (N=1)

(12.4x lower mean squared error than uniform sampling)

Summary: Monte Carlo integration

- **Monte Carlo estimator**
  - Estimate integral by evaluating function at random sample points in domain

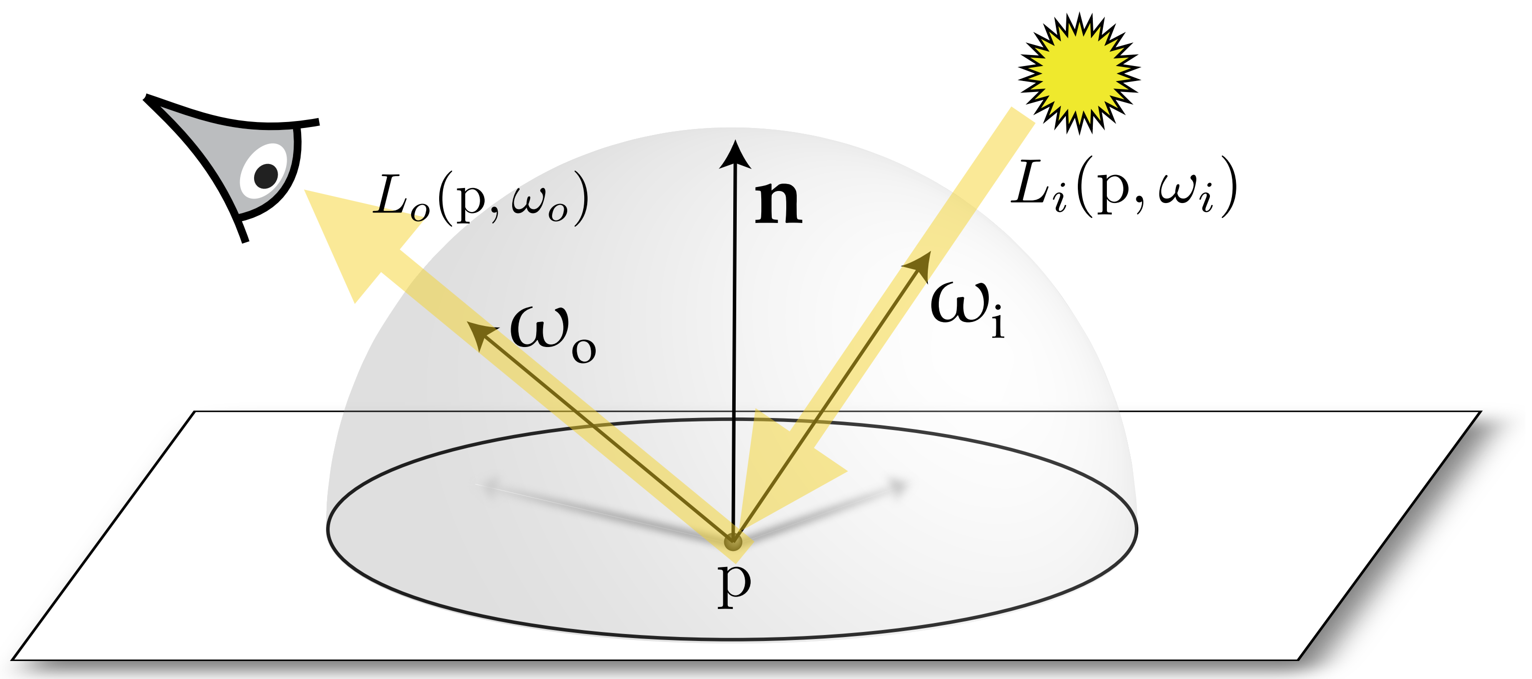
$$E[F_N] = E\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}\right] = \int_a^b f(x)\,\mathrm{d}x$$

- **The function (the estimator) is computed by a ray tracer!**

- **Useful in rendering due to estimate high dimension integrals**
  - Faster convergence in estimating high dimensional integrals than non-randomized methods
  - But it's still slow…
  - Suffers from noise due to variance in estimate (need many samples to produce good quality images)

- **Importance sampling**
  - Reduce variance by biasing choice of samples to regions of domain where value of function is large
  - Intuition: pick samples that will "contribute most" to estimate
  - Intelligent sampling matters A LOT!

But wait.... We know how to numerically estimate the reflection equation, but that depends on $L_\mathrm{i}(\mathrm{p}, \omega_j)$



Reflection equation in terms of integration over solid angles:

$$L_\mathrm{o}(\mathrm{p}, \omega_\mathrm{o}) = \int_{H^2} f_r(\mathrm{p}, \omega_\mathrm{i} \rightarrow \omega_\mathrm{o})\, L_\mathrm{i}(\mathrm{p}, \omega_\mathrm{i})\, \cos\theta_\mathrm{i}\, \mathrm{d}\omega_\mathrm{i}$$

Monte Carlo estimate via sampling solid angle according to the PDF: $p(\omega)$

Sample: $\omega_j \sim p(\omega)$

Estimate: $$\frac{1}{N} \sum_{j=1}^{N} \frac{f_r(\mathrm{p}, \omega_j \rightarrow \omega_\mathrm{o})\, \boxed{L_\mathrm{i}(\mathrm{p}, \omega_j)}\, \cos\theta_j}{p(\omega_j)}$$

We talked about how get this radiance from a light source in the scene, or from an environment map

But consider incoming light at any point in this scene

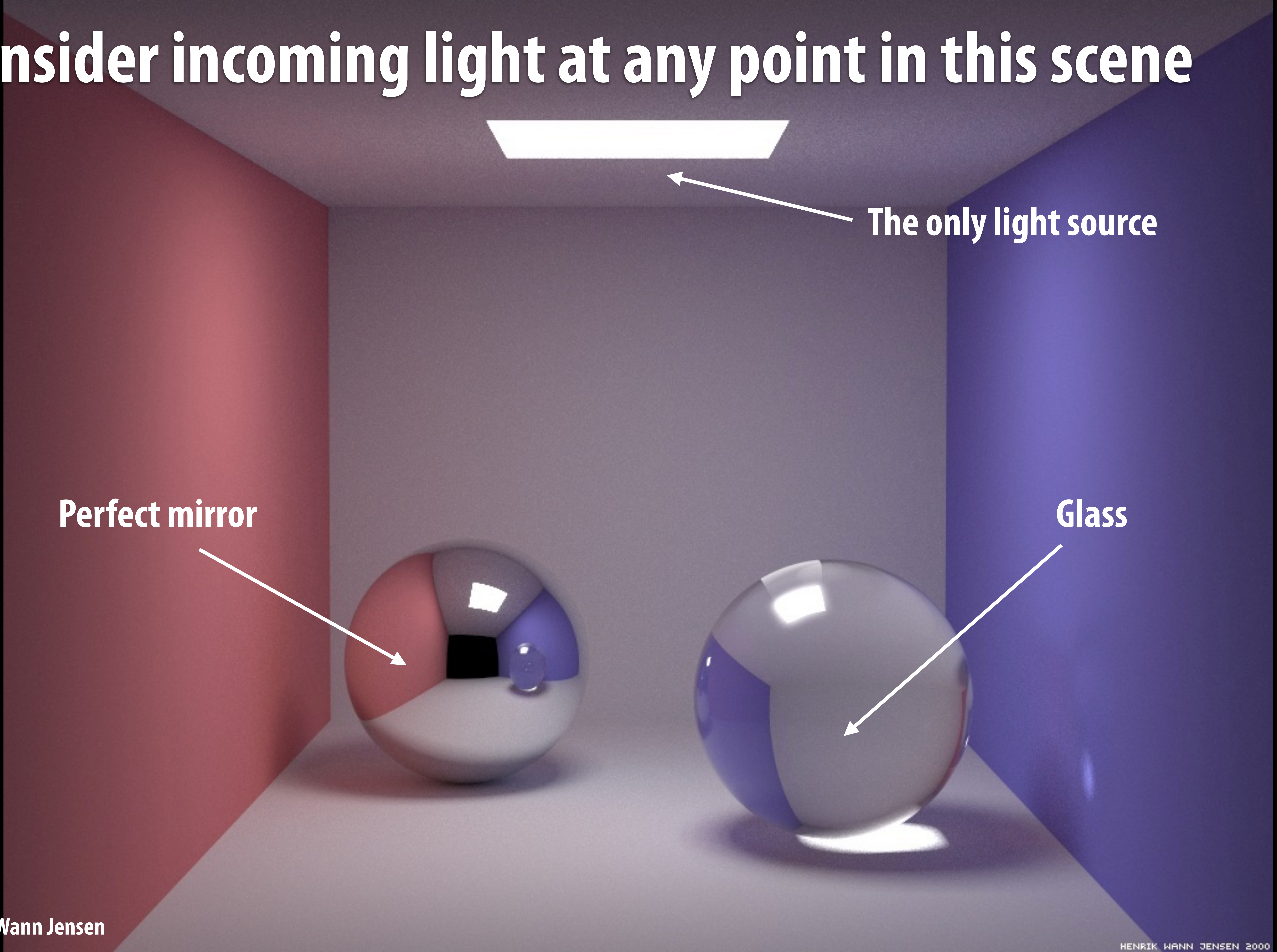


Image credit: Henrik Wann Jensen

To compute realistically lit images,
we have to account for many bounds of light in a scene.
(Next time…)

Acknowledgements

- Thanks to Keenan Crane, Ren Ng, Pat Hanrahan and Matt Pharr for presentation resources