COL781: Computer Graphics

23. The Rendering Equation

James Kajiya

Ray tracing

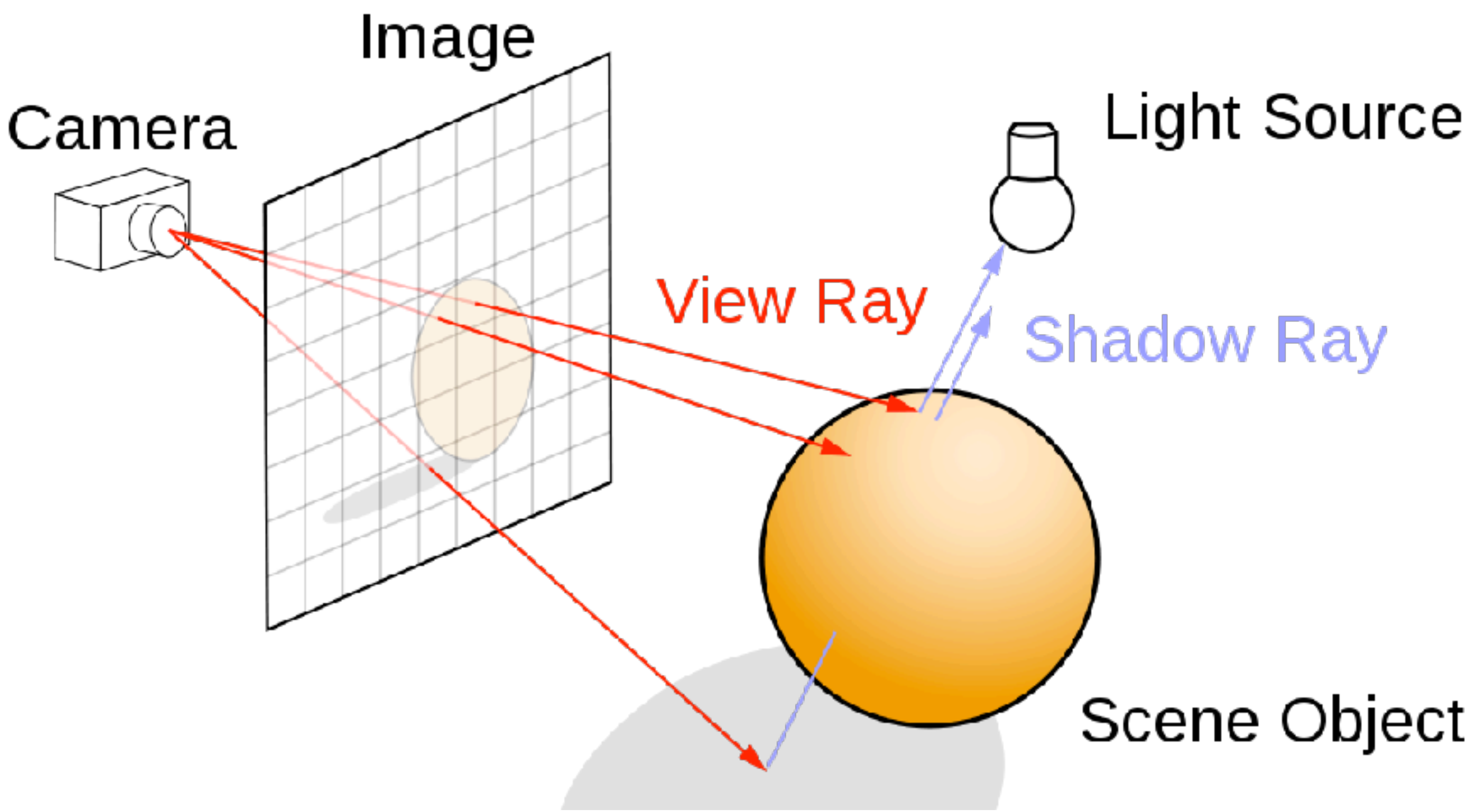
For each sample:

Cast a ray into the scene

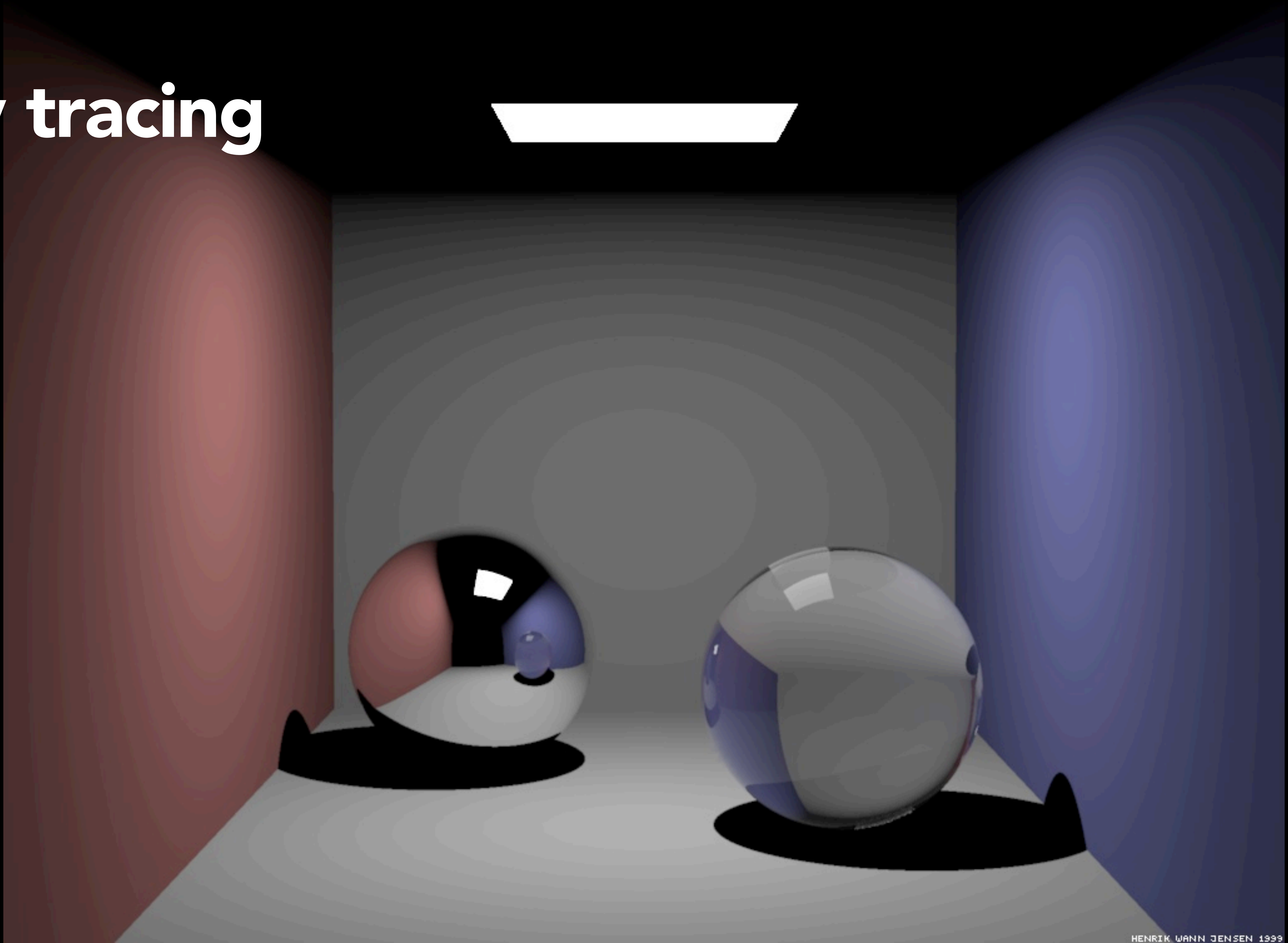
Find the closest intersection

Get shaded colour at intersection point

Set sample colour to it

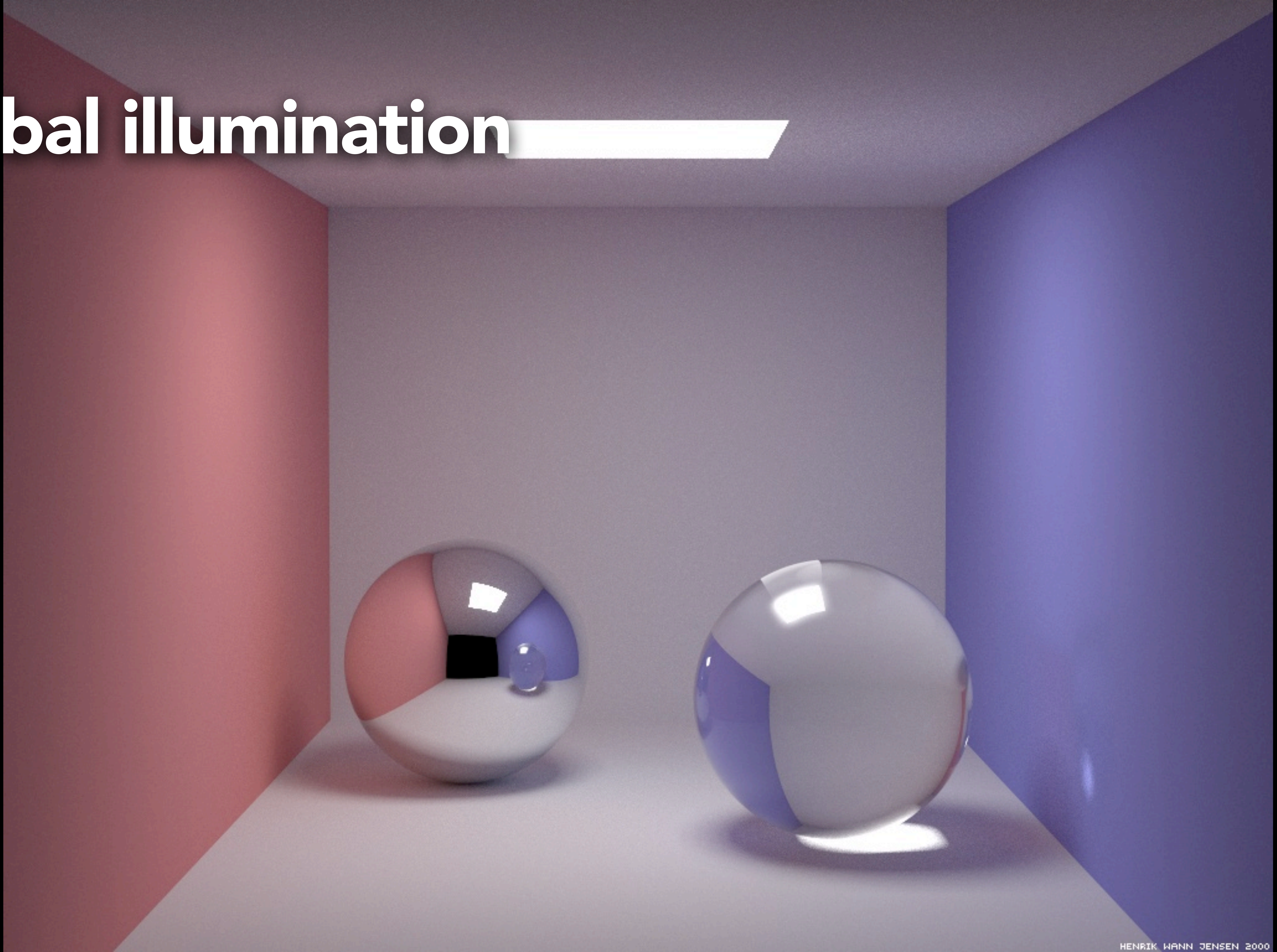


Ray tracing



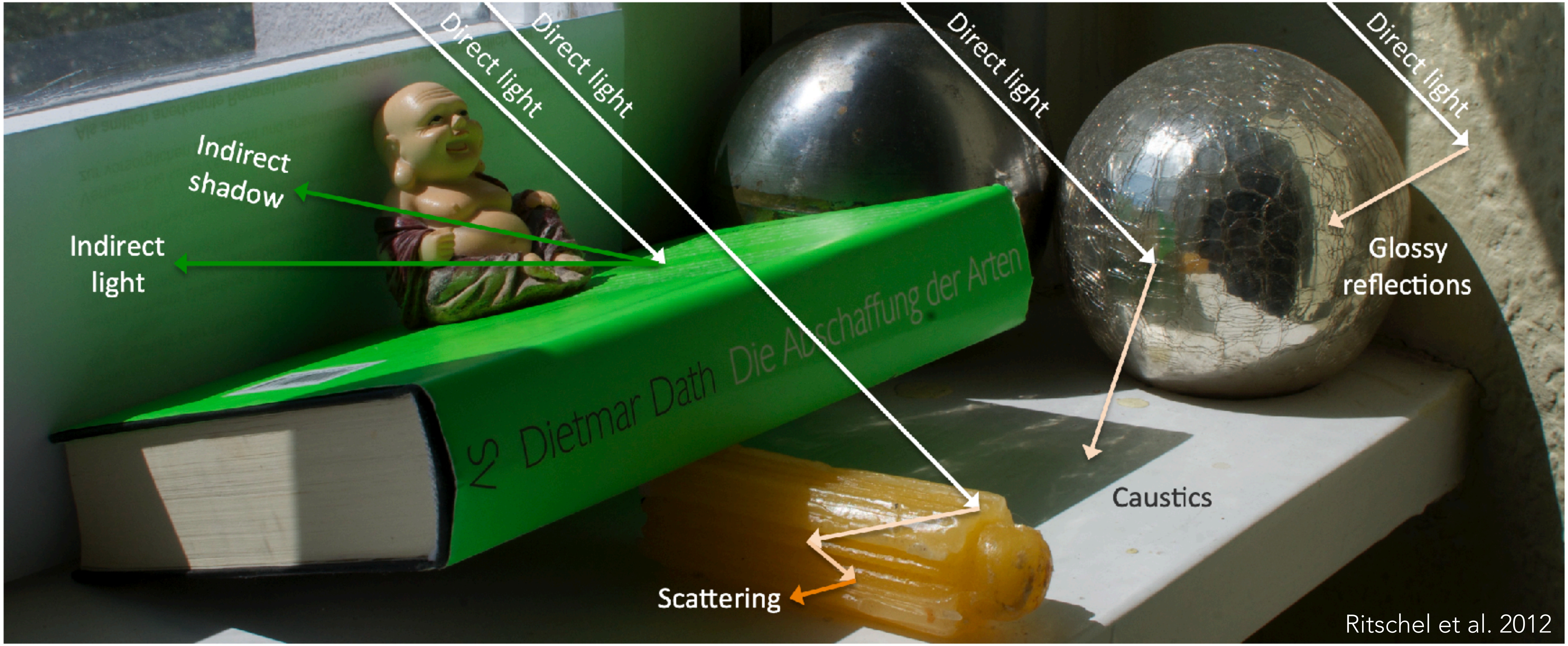
Henrik Wann Jensen

Global illumination



Henrik Wann Jensen

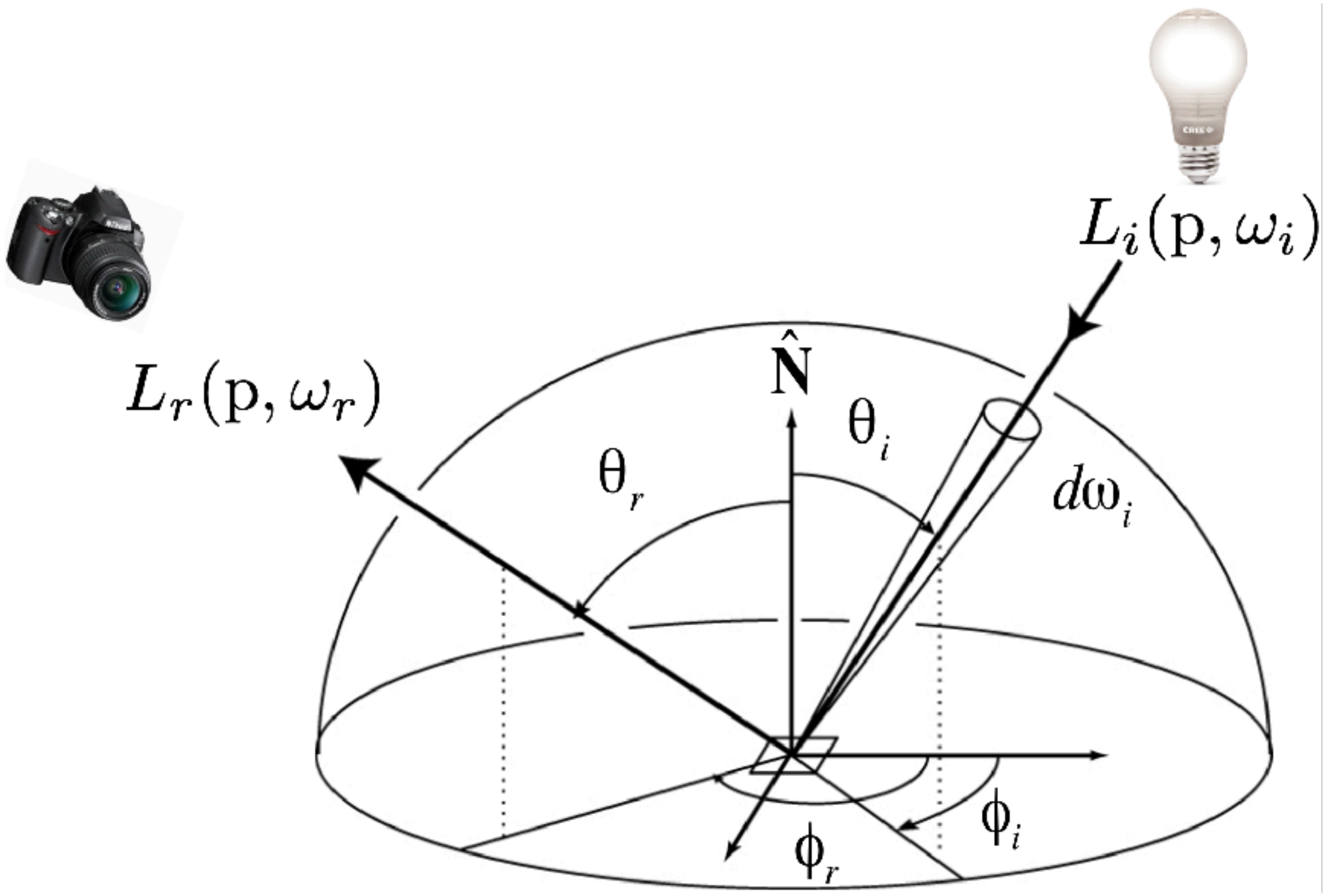
Global illumination



Ray tracing revisited

For each sample:

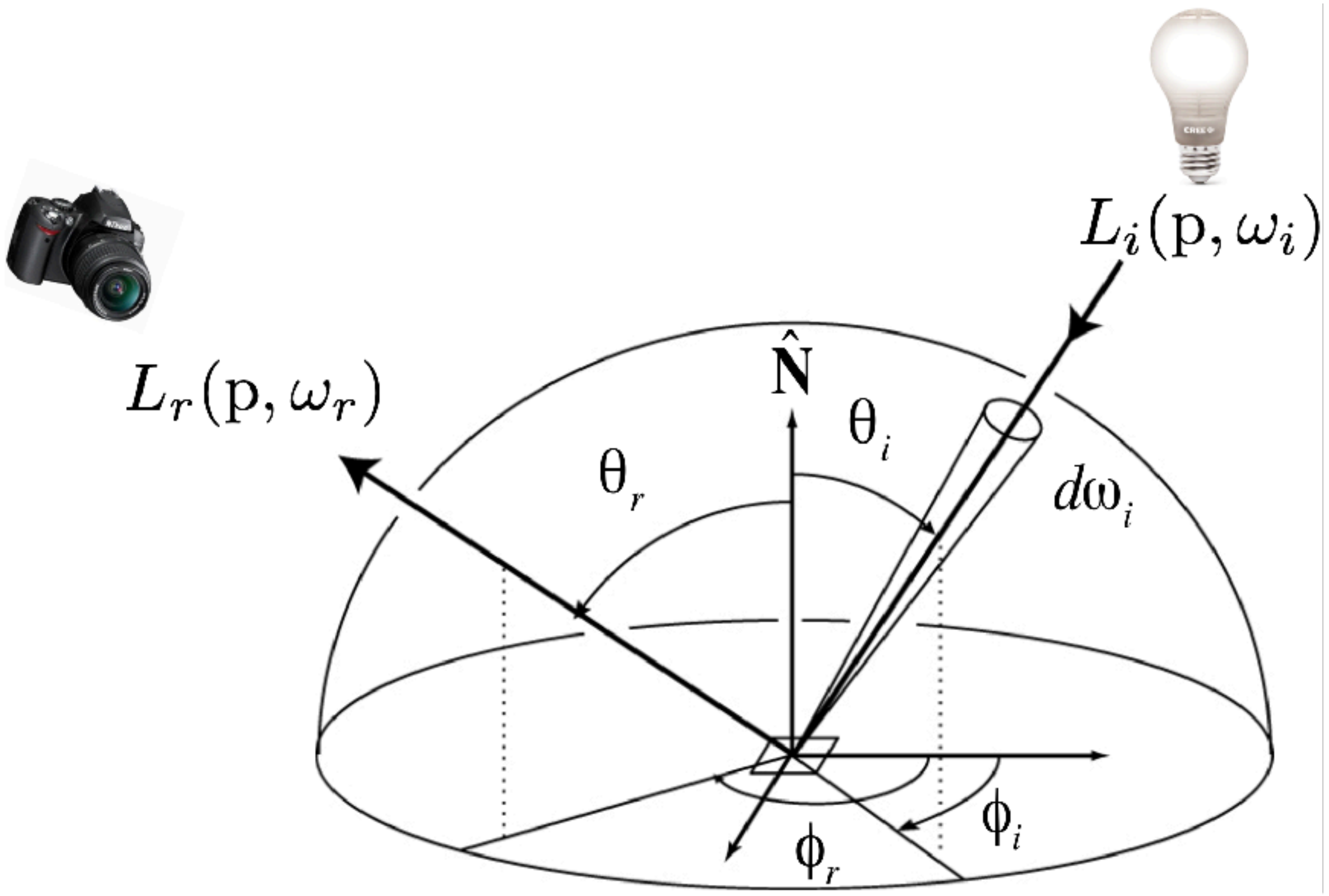
- Cast a ray into the scene
- Find the closest intersection
- Get exitant radiance at intersection point
- Set sample colour to it



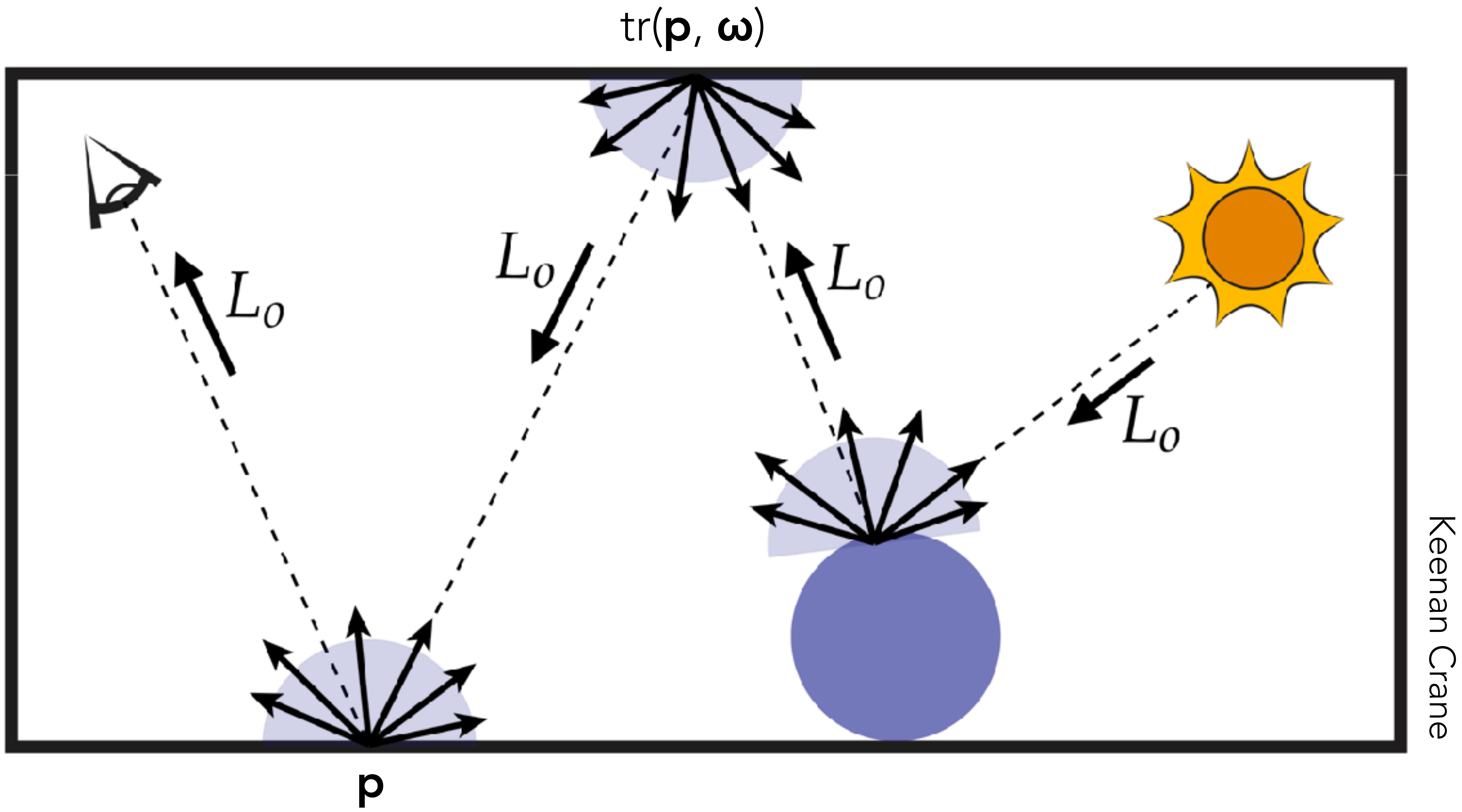
$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) = L_e(\mathbf{p}, \boldsymbol{\omega}_o) + \int_{H^2} f_r(\mathbf{p}, \boldsymbol{\omega}_i \to \boldsymbol{\omega}_o)\, L_i(\mathbf{p}, \boldsymbol{\omega}_i) \cos(\theta_i)\, \mathrm{d}\boldsymbol{\omega}_i$$

$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) = L_e(\mathbf{p}, \boldsymbol{\omega}_o) + \int_{H^2} f_r(\mathbf{p}, \boldsymbol{\omega}_i \rightarrow \boldsymbol{\omega}_o)\, L_i(\mathbf{p}, \boldsymbol{\omega}_i) \cos(\theta_i)\, d\boldsymbol{\omega}_i$$

- How to evaluate incident radiance from any direction (not just light sources)?
- How to compute the integral over a hemisphere?



What is $L_i(\mathbf{p}, \boldsymbol{\omega}_i)$? Simply exitant radiance from somewhere else!



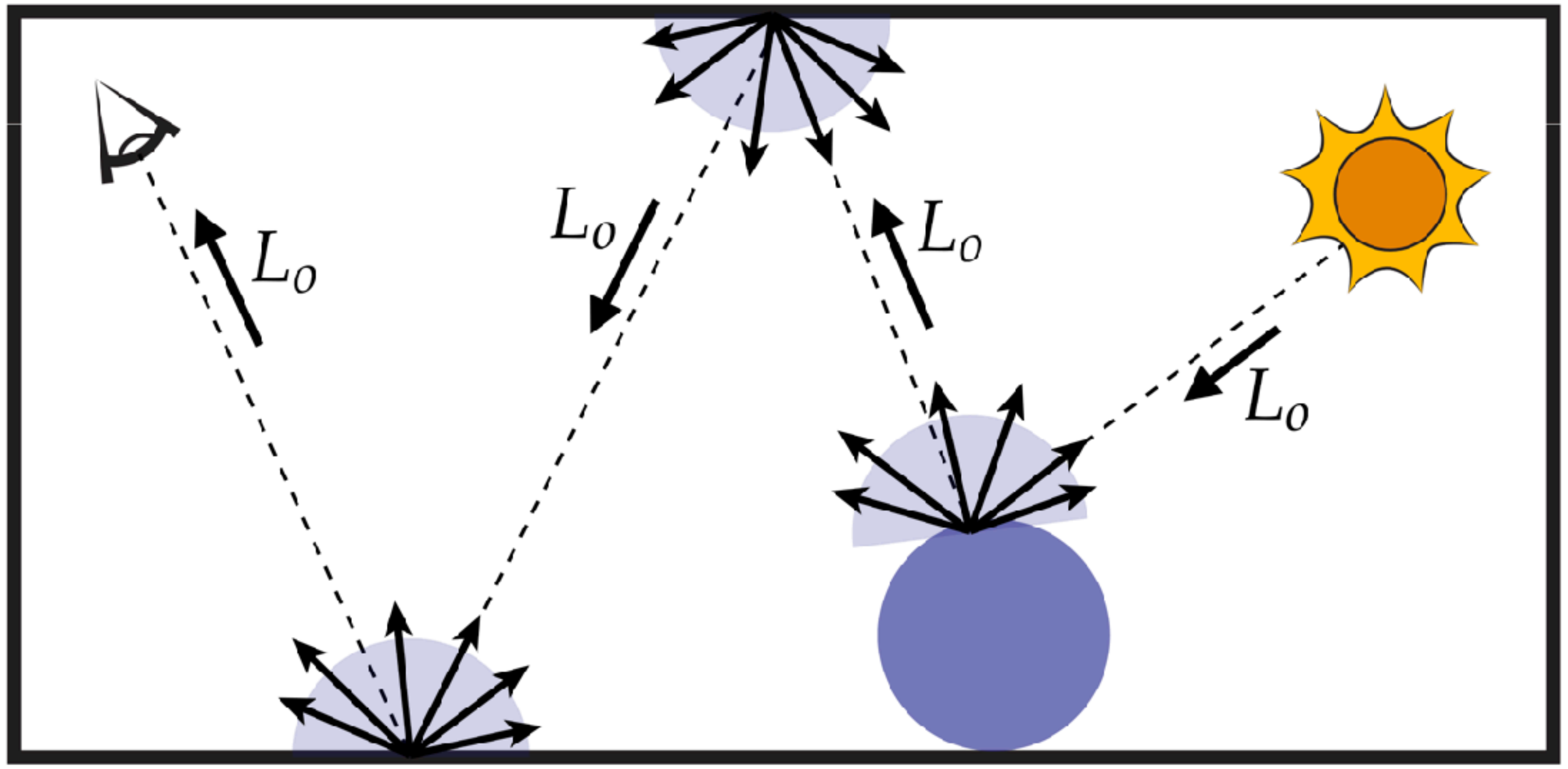
Define $\mathrm{tr}(\mathbf{p}, \boldsymbol{\omega})$ as the first surface point hit by the ray $\mathbf{p} + t\boldsymbol{\omega}$.

$$L_i(\mathbf{p}, \boldsymbol{\omega}_i) = L_o(\mathrm{tr}(\mathbf{p}, \boldsymbol{\omega}_i), -\boldsymbol{\omega}_i)$$

$$L_o(\mathbf{p}, \boldsymbol{\omega}_o) = L_e(\mathbf{p}, \boldsymbol{\omega}_o) + \int_{H^2} f_r(\mathbf{p}, \boldsymbol{\omega}_i \rightarrow \boldsymbol{\omega}_o)\, L_o(\mathrm{tr}(\mathbf{p}, \boldsymbol{\omega}_i), -\boldsymbol{\omega}_i) \cos(\theta_i)\, d\boldsymbol{\omega}_i$$

This is an integral equation!
Unknown quantity L_o on both sides

Like ray tracing, we'll evaluate it recursively



Numerical integration

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

If I know how to compute $f(x)$, how can I compute its integral?

- Analytical / symbolic
- Numerical quadrature
- Monte Carlo methods

Analytical integration

$$\int x^3 \, \mathrm{d}x = \frac{1}{4}x^4 \qquad \int x \cos x \, \mathrm{d}x = x \sin x + \cos x$$

$$\int e^{-x^2} \, \mathrm{d}x = ? \qquad \int \lceil \sin x^2 \rceil \, \mathrm{d}x = ?$$

Closed-form formulas only possible in very special cases.

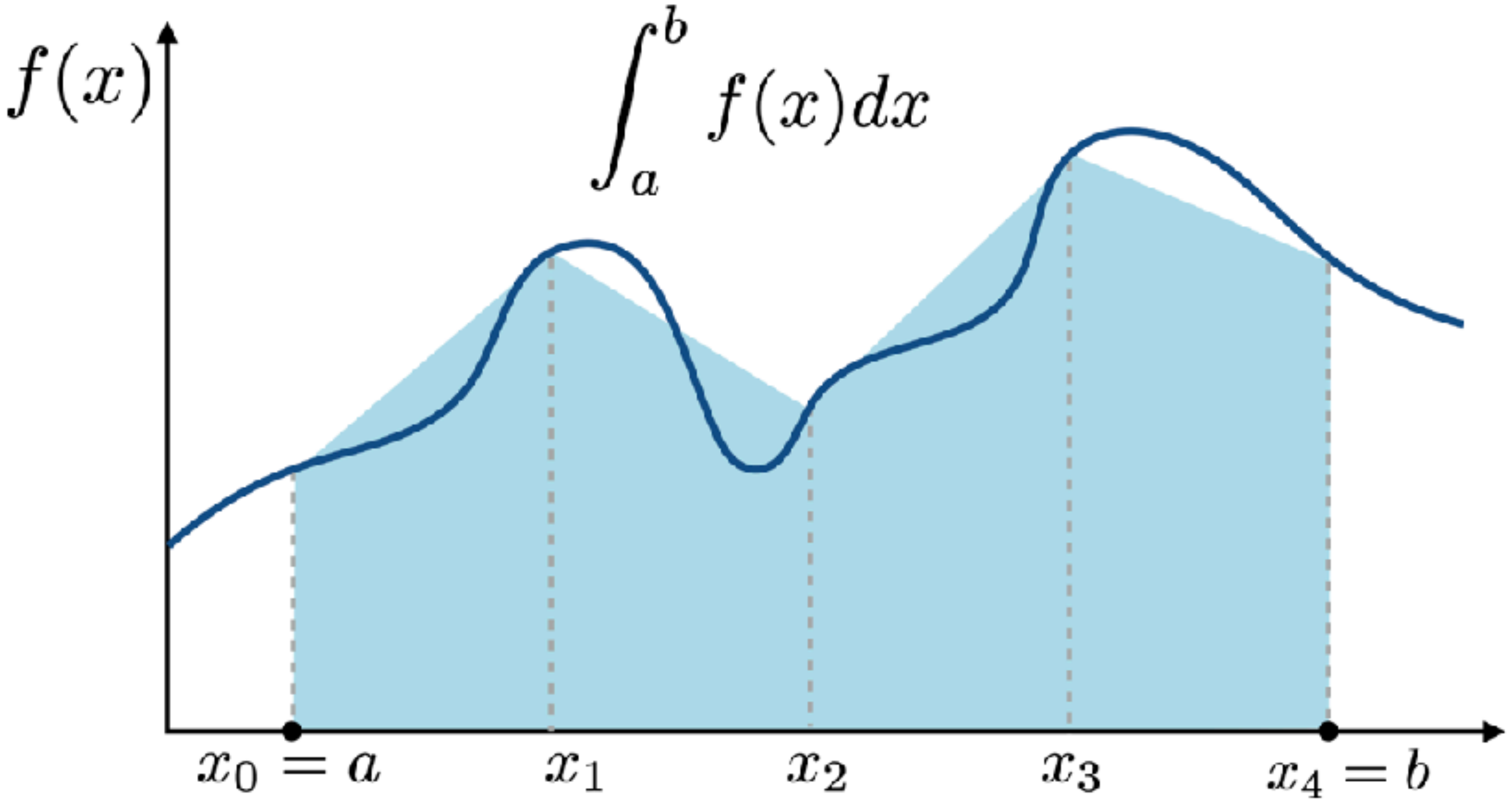
In rendering, integrand is very complicated! Depends on visibility, texture, BRDF, …
No chance of analytical solution.

Numerical quadrature

Sample function at various points, estimate integral as weighted sum

e.g. trapezoidal rule:

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} \left(\frac{f(x_{i-1}) + f(x_i)}{2} \right) \Delta x_i$$



If integrand is smooth, error decreases as $O(n^{-2})$

Many higher-order accurate methods
e.g. Gaussian quadrature, Simpson's rule, etc.

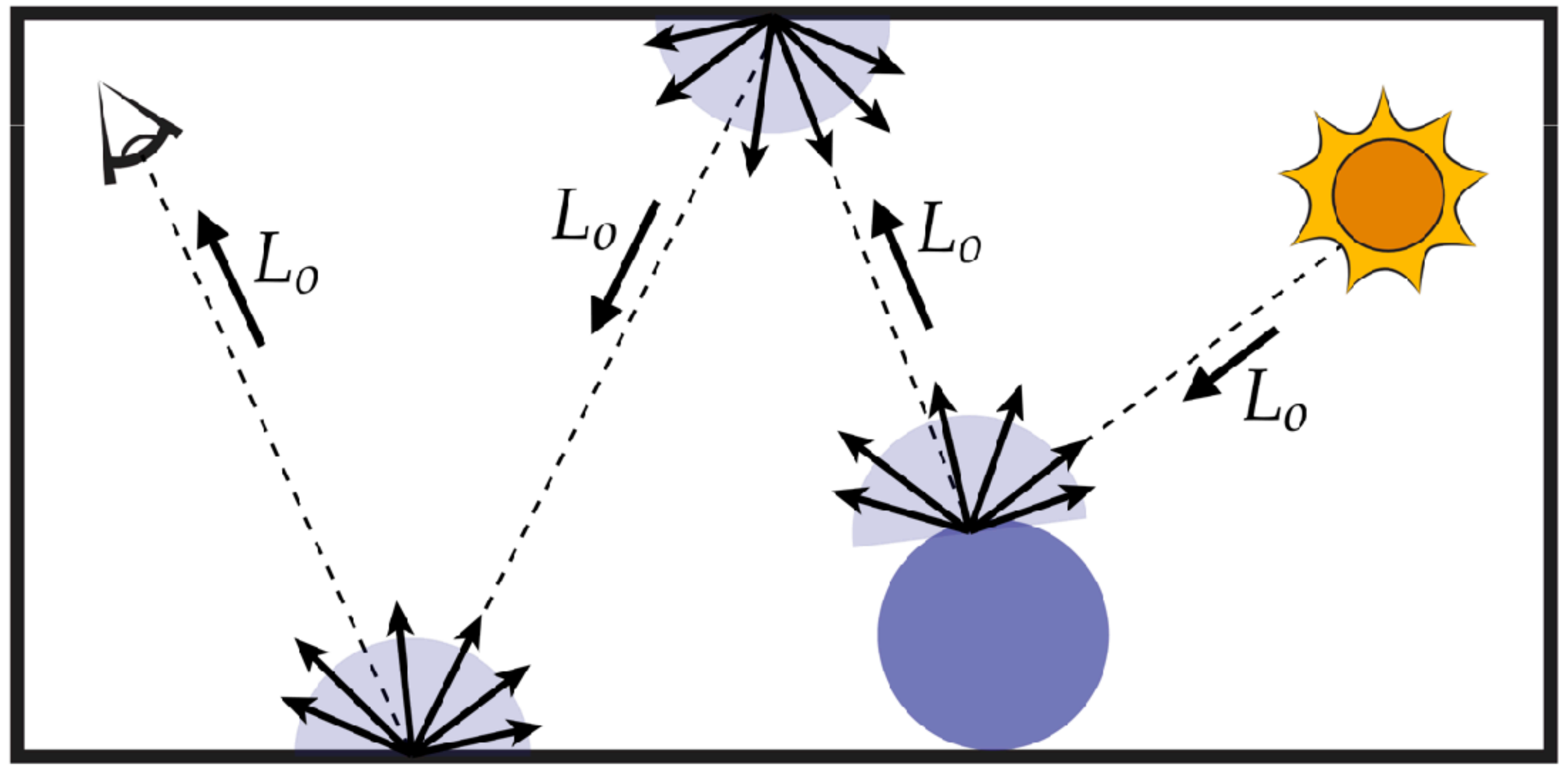
Why not use quadrature?

- Integrand is not smooth! e.g. incident radiance from area light

  Error might decrease at only $O(n^{-1})$

- Integral is high-dimensional! e.g. k-bounce illumination requires integral over k hemispheres

  If n samples per hemisphere,
  computational cost increases as $O(n^k)$.
  Error still decreasing at same rate w.r.t. n

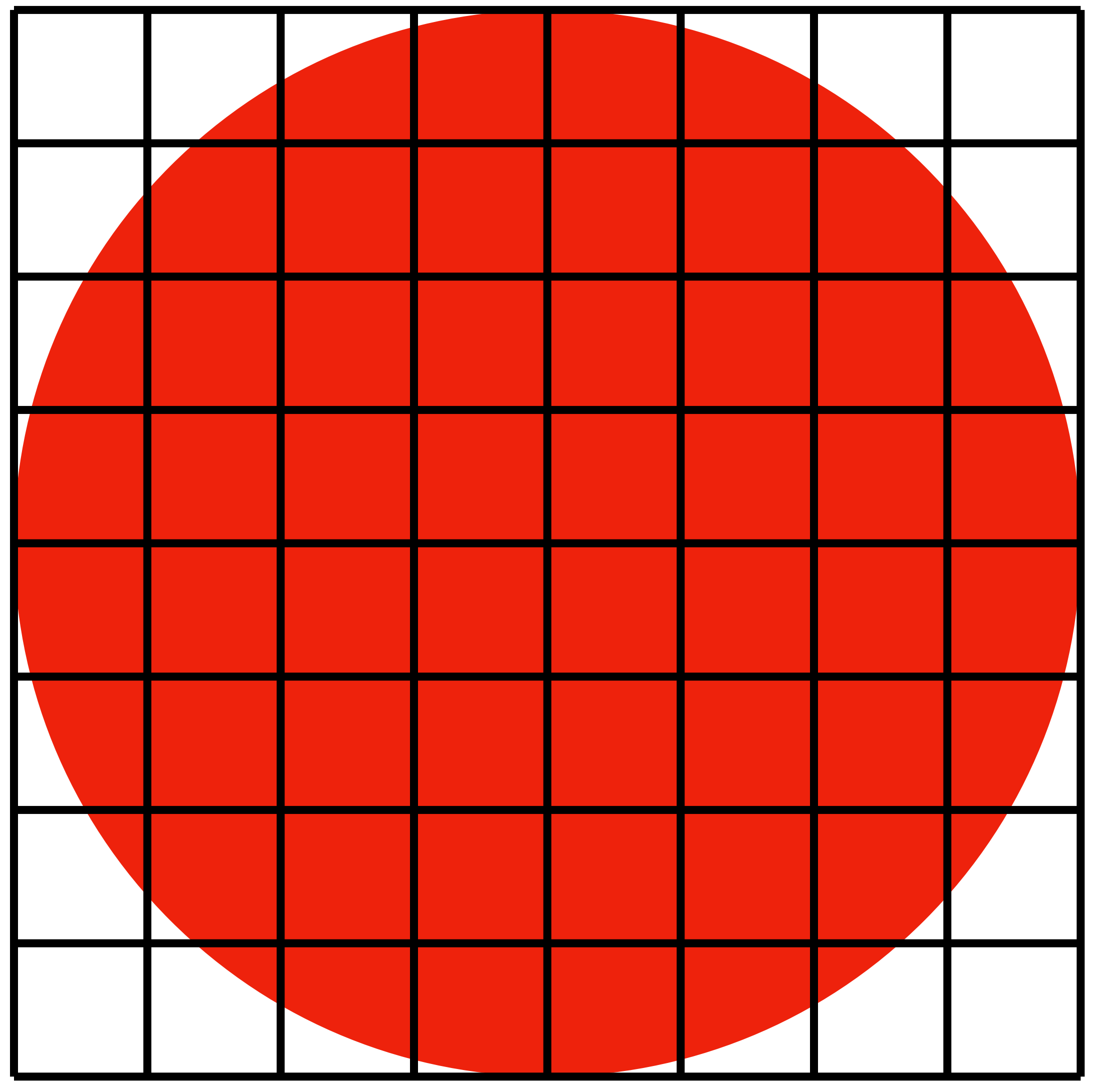


Example: area of a disk

Suppose you want to estimate π by computing the area of the region $\{(x, y): x^2 + y^2 \leq 1\}$.

$$f(x,y) = \begin{cases} 1 & \text{if } x^2 + y^2 \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

$$A = \int_{-1}^{1} \int_{-1}^{1} f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$



With trapezoidal rule:

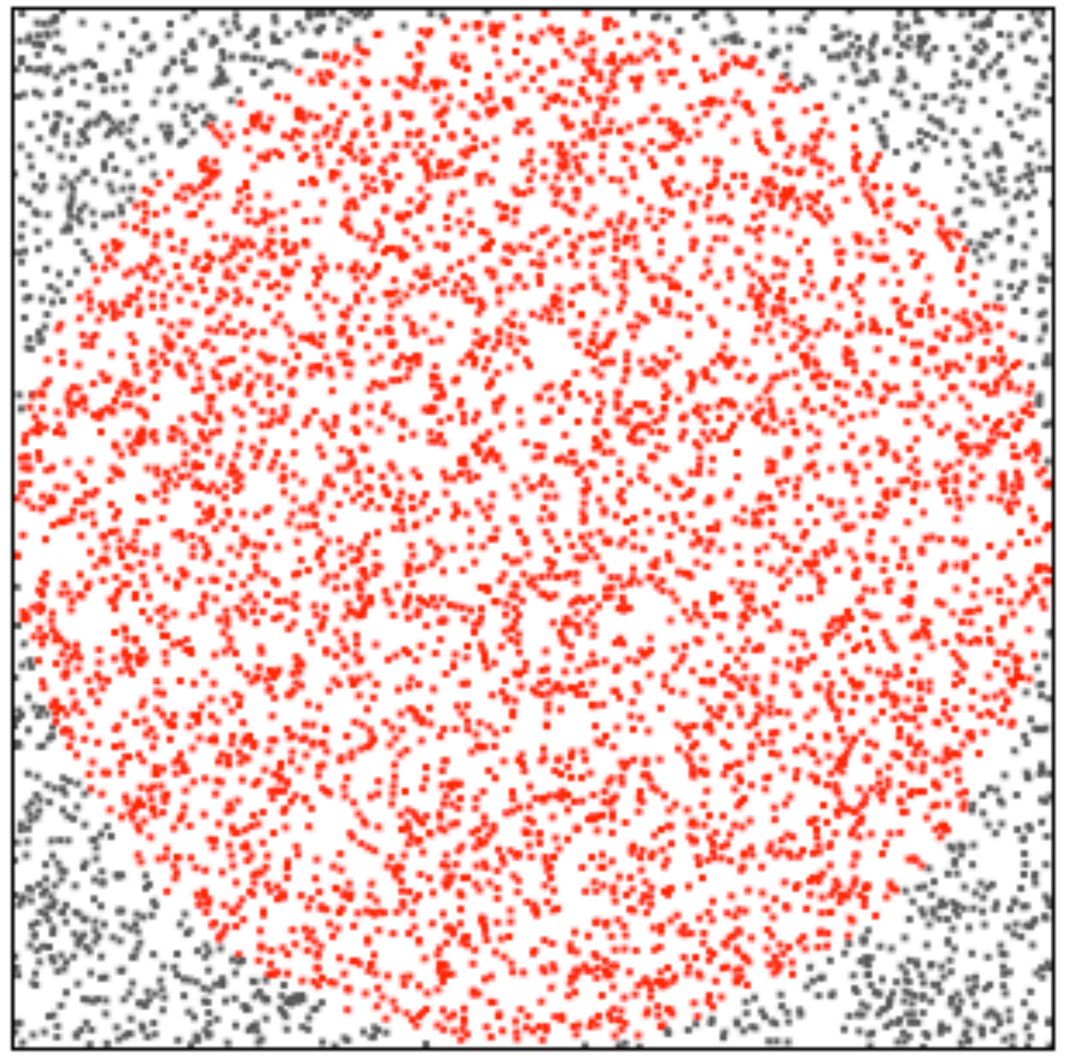
- $O(n)$ samples in x and y each → $N = O(n^2)$ total samples
- Discontinuous integrand → error decreases slowly

What about finding the volume of a k-dimensional ball? 😬

A randomized algorithm

Pick N random points uniformly distributed in $[-1, 1]^2$, count how many land in the disk.

Let M = number of points with $x^2 + y^2 \leq 1$.



- Probability of a point landing in the disk = $A/4$
- Expected number of points: $E[M] = NA/4$

So, estimated area = $4M/N$.

What is the likely error in the estimate?

Quick probability recap

If X is a random variable with probability distribution $p(x)$, its **expected value** or **expectation** is

$$E[X] = \sum x_i \, p_i \qquad \text{(discrete)}$$

$$E[X] = \int x \, p(x) \, \mathrm{d}x \qquad \text{(continuous)}$$

Expectation is linear:

- $E[X_1 + X_2] = E[X_1] + E[X_2]$
- $E[aX] = a \, E[X]$

Variance = average squared deviation from expected value

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

Variance is **not** linear, but it is additive for **independent** random variables:

- If X_1 and X_2 are independent, then $V[X_1 + X_2] = V[X_1] + V[X_2]$
- $V[aX] = a^2\, V[X]$

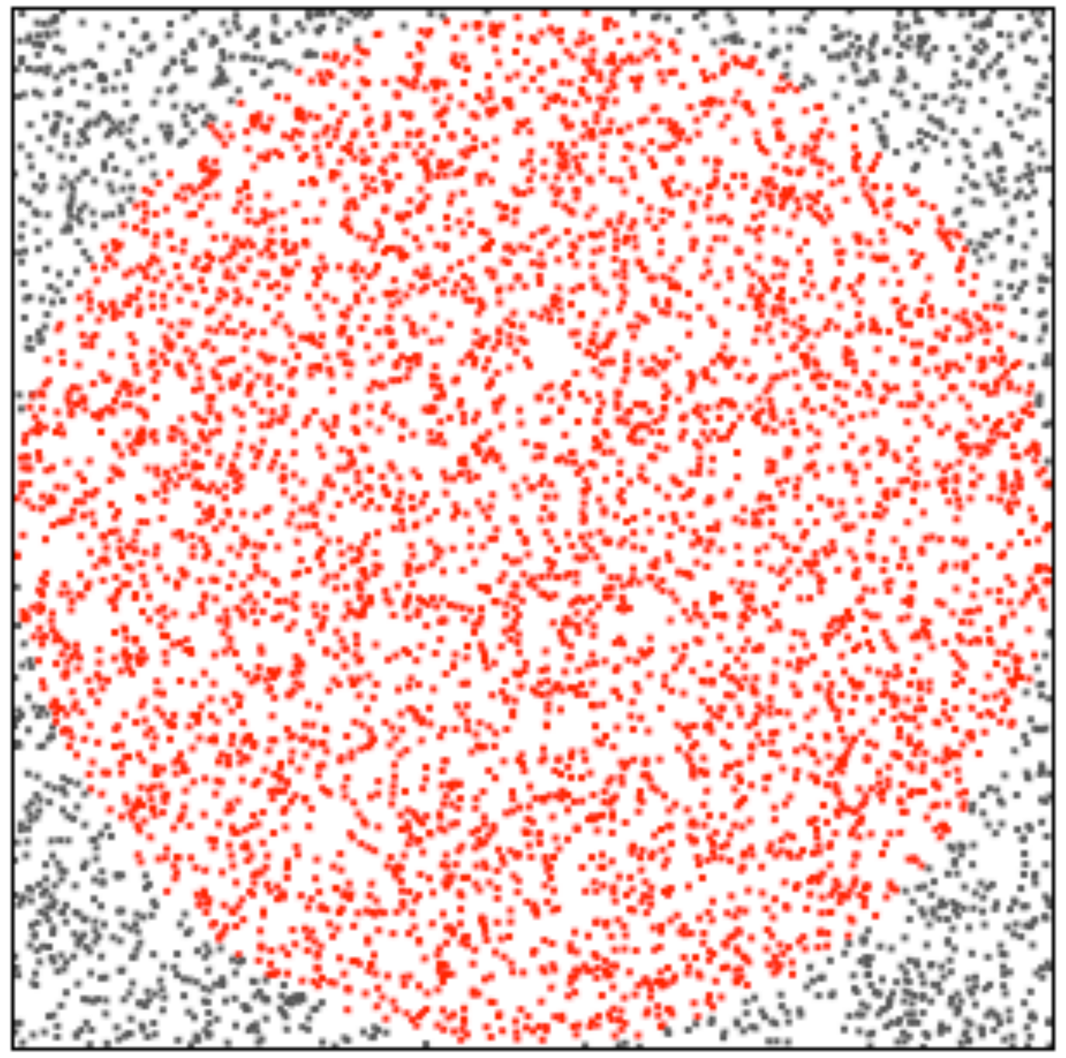
So if I take the mean of N i.i.d. random variables,

$$V\left[\frac{1}{N}\sum X_i\right] = \frac{1}{N^2} V\left[\sum X_i\right] = \frac{1}{N^2}\sum V[X_i] = \frac{1}{N} V[X]$$

Randomized area estimation

Pick N random points X_i independently and uniformly distributed in $[-1, 1]^2$.

Let $Y_i = f(X_i)$, so number of points in disk is $M = \sum Y_i$.



- What are $E[Y_i]$ and $E[M]$?
- What are $V[Y_i]$ and $V[M]$?

Variance of estimated area = $O(N^{-1})$

What about in k dimensions? Estimated volume = $2^k\, M/N$, variance still $O(N^{-1})$!

The basic Monte Carlo method

If X is uniformly distributed in $[a, b]$, then

$$E[f(X)] = \frac{1}{b-a}\int_a^b f(x)\,\mathrm{d}x$$

So, if I take N independent samples of X,

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) \approx E[f(X)] = \frac{1}{b-a}\int_a^b f(x)\,\mathrm{d}x$$

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

Interpretation:
Integral = average value × domain size

Basic Monte Carlo estimation of an integral $\int_a^b f(x)\,\mathrm{d}x$:

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

where $X_i \sim U(a, b)$ with probability density $p(x) = 1/(b - a)$.

- F_N is an **unbiased** estimator: $E[F_N] = \int_a^b f(x)\,\mathrm{d}x$ for any N.
- Variance decreases linearly: $V[F_N] = \frac{(b-a)^2}{N} V[f(X)]$
- Standard deviation $= \sqrt{V[F_N]} = O(N^{-1/2})$

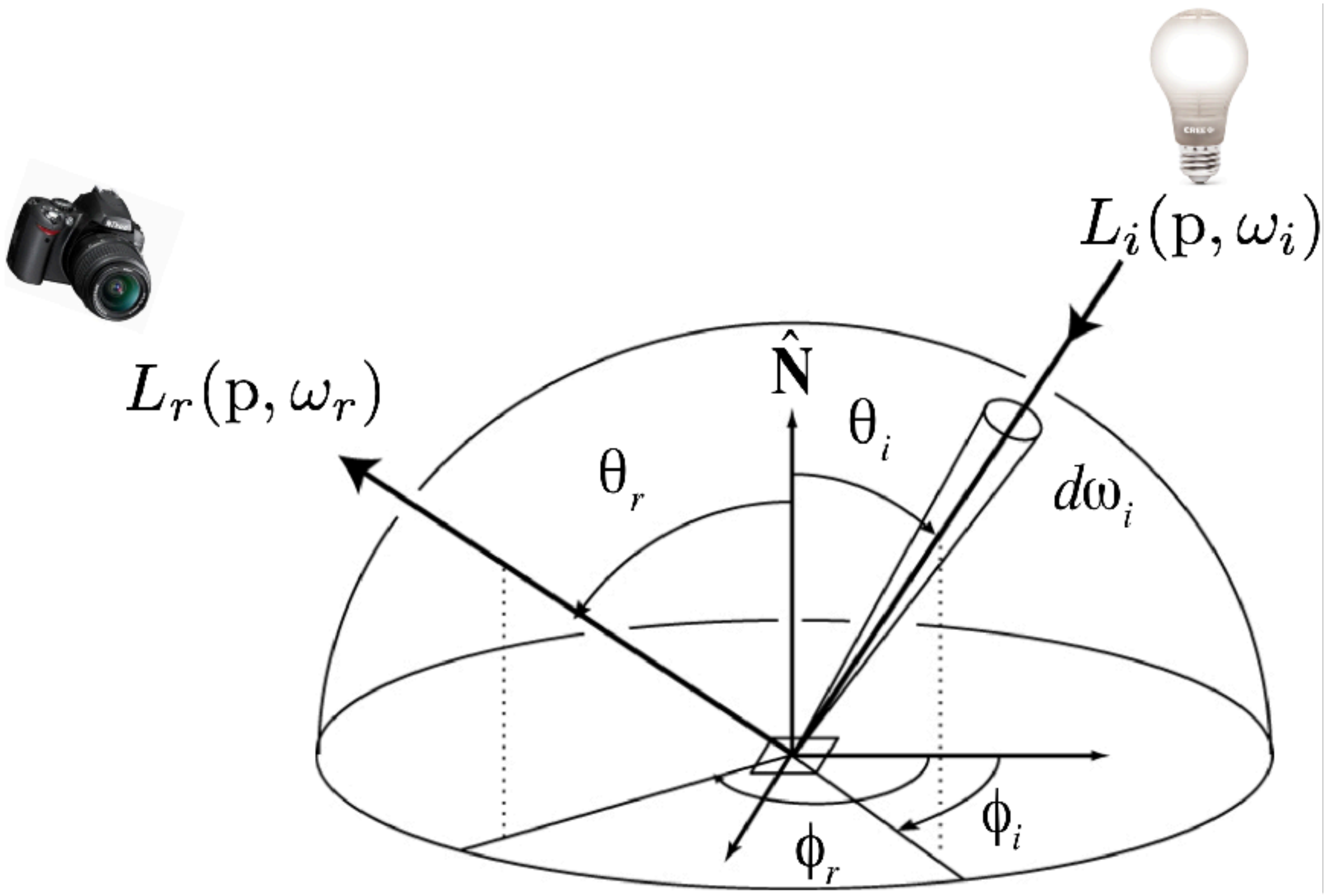
Back to rendering

Monte Carlo rendering

We need to estimate the reflectance integral $\int_{H^2} f_r(\mathbf{p}, \boldsymbol{\omega}_i \to \boldsymbol{\omega}_o)\, L_i(\mathbf{p}, \boldsymbol{\omega}_i) \cos(\theta_i)\, d\boldsymbol{\omega}_i$

With Monte Carlo, it's easy:

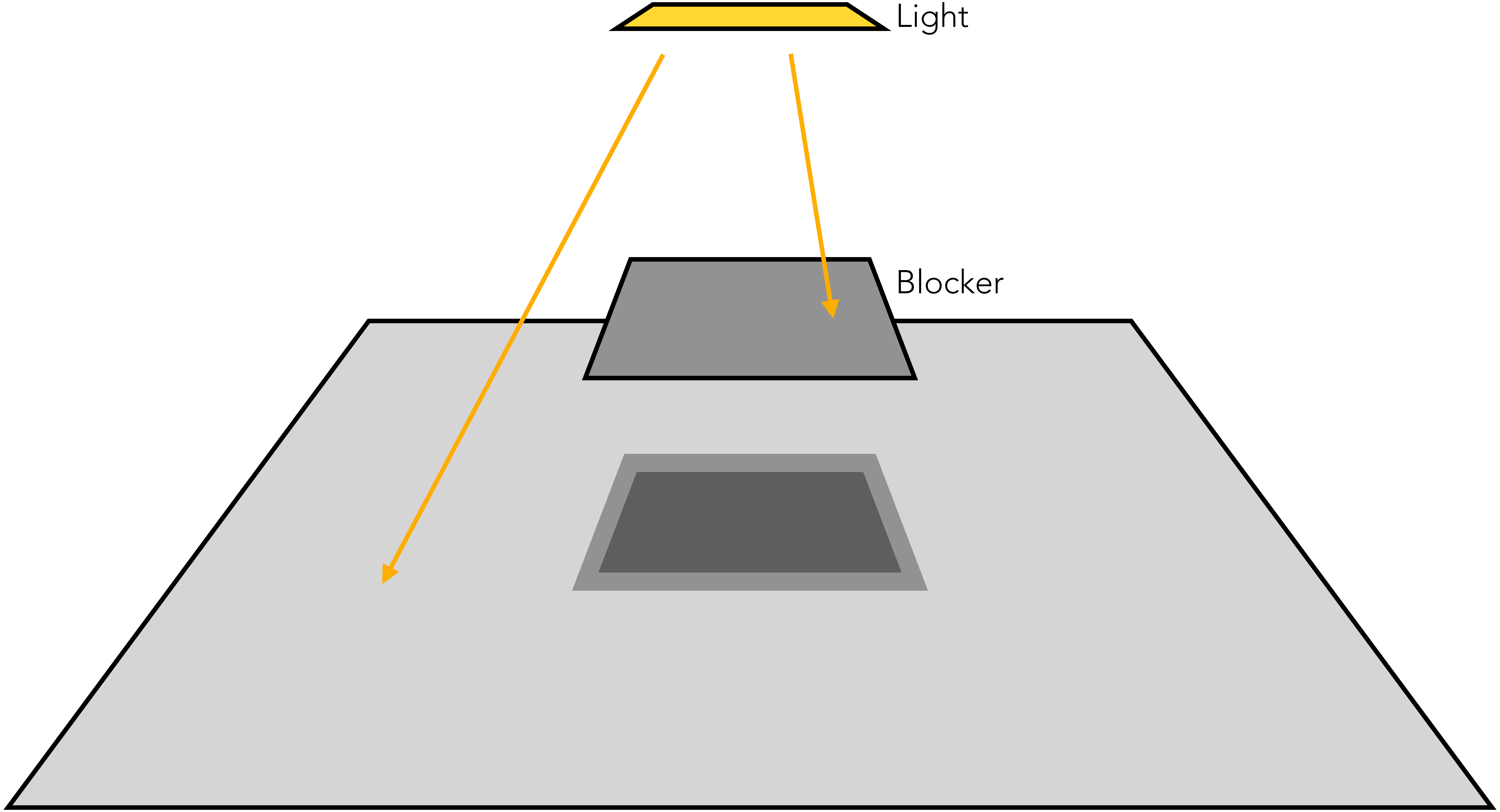
- Uniformly sample hemisphere of incident directions: $\mathbf{X}_i \sim U(H^2)$, probability density $p(\boldsymbol{\omega}) = 1/(2\pi)$
- Evaluate integrand $Y_i = f_r(\mathbf{p}, \mathbf{X}_i \to \boldsymbol{\omega}_o)\, L_i(\mathbf{p}, \mathbf{X}_i) \cos(\theta_i)$
- MC estimator is simply $F_N = 2\pi/N \sum Y_i$

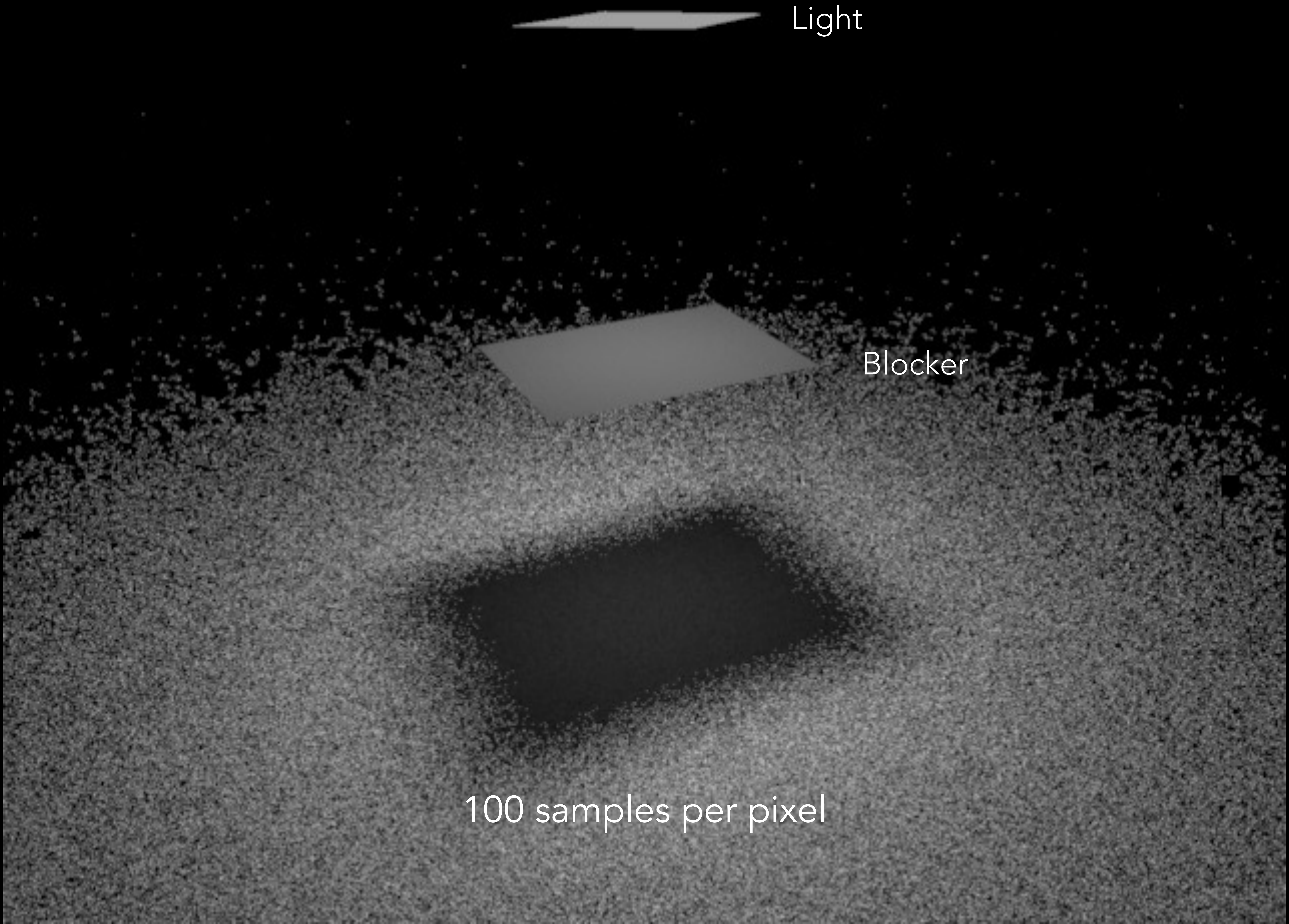


```
incidentRadiance(x, ω):
   p = intersectScene(x, ω)
   L = p.emittedLight(−ω)
   for i = 1, …, N:
      ωi = sampleDirection(p.normal)
      L += incidentRadiance(p, ωi) * p.BRDF(ωi, −ω) * cos_θi * 2π / N
   return L
```

Two problems:

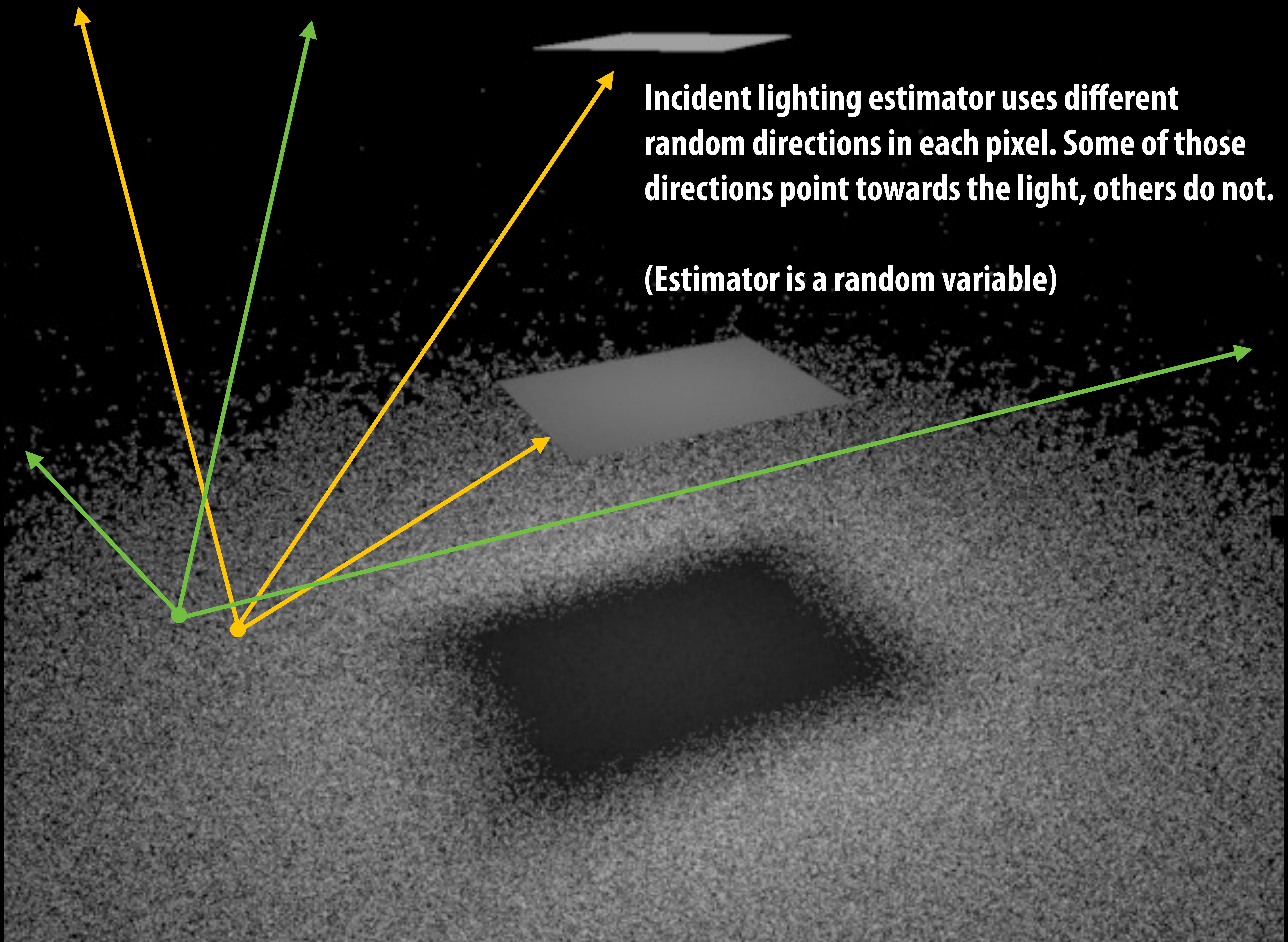
- Exponential increase in #samples: after k bounces, we're tracing N^k rays
- Don't know when/how to stop recursion


Light
Blocker


Light
Blocker
100 samples per pixel

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)



Keenan Crane

Three problems:

- Exponential increase in #samples: after *k* bounces, we're tracing N^k rays
- Don't know when/how to stop recursion
- Results are noisy!