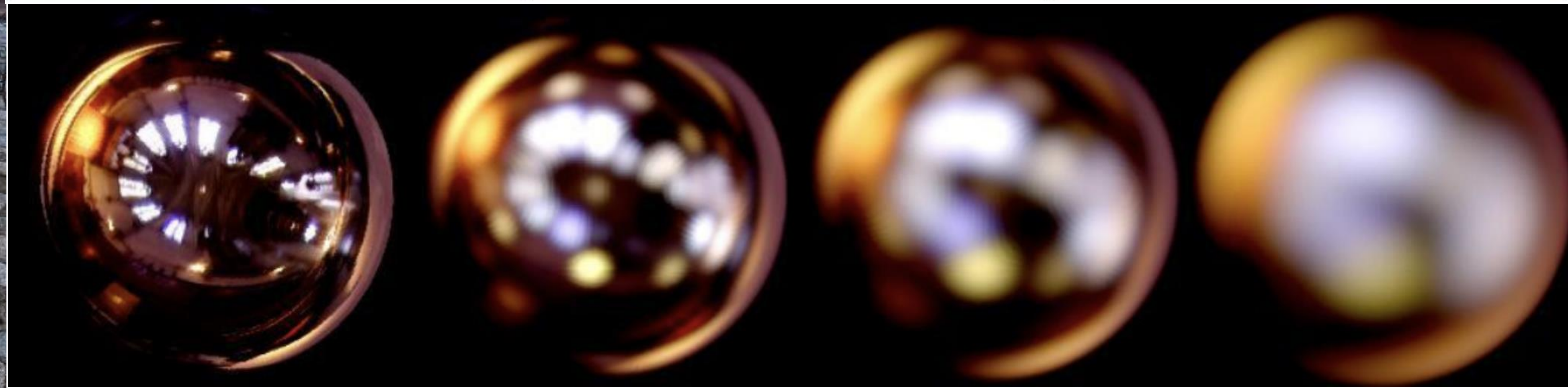
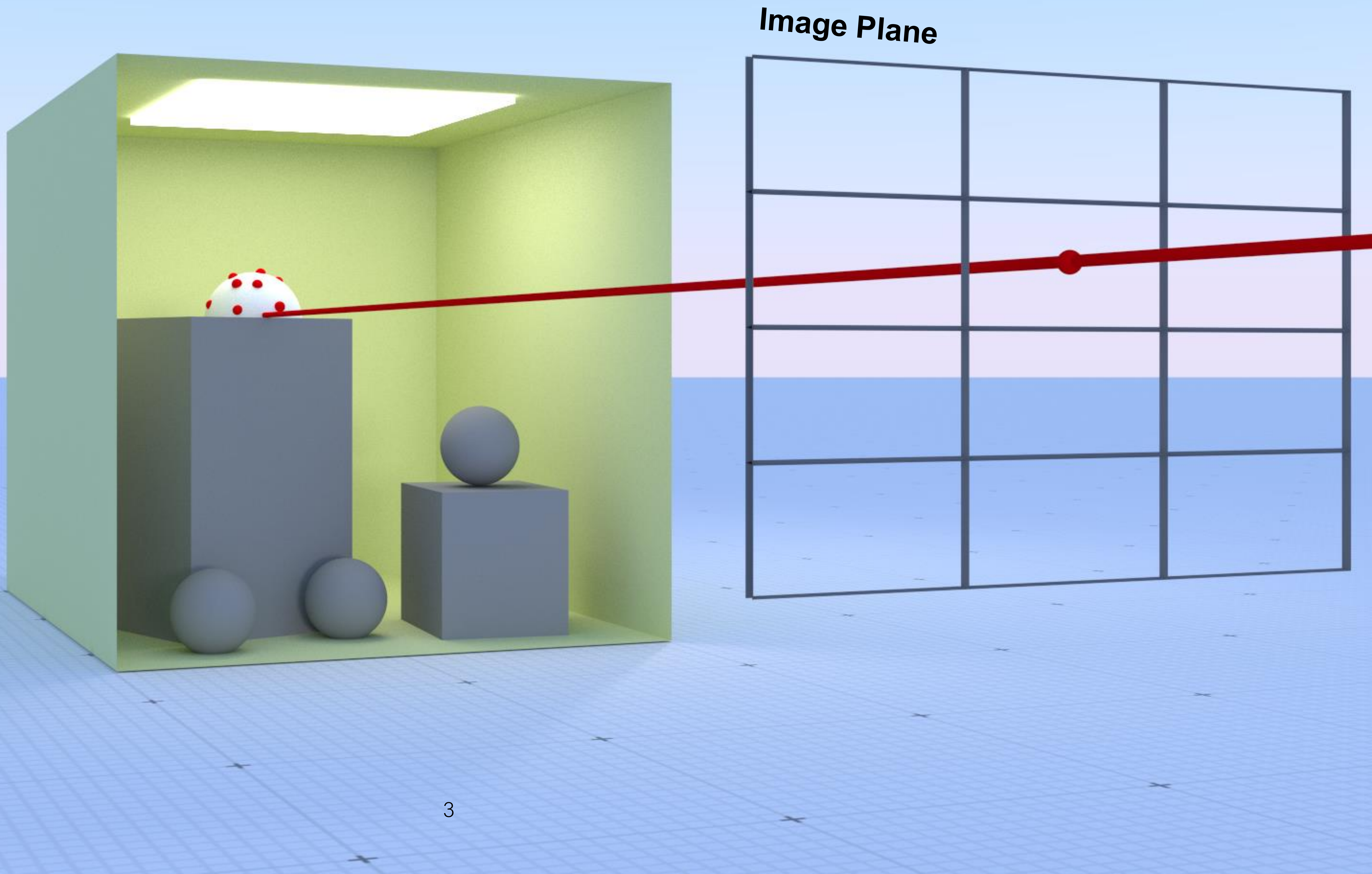


Monte Carlo Integration, Path Tracing & Microfacet BSDFs

Gurprit Singh

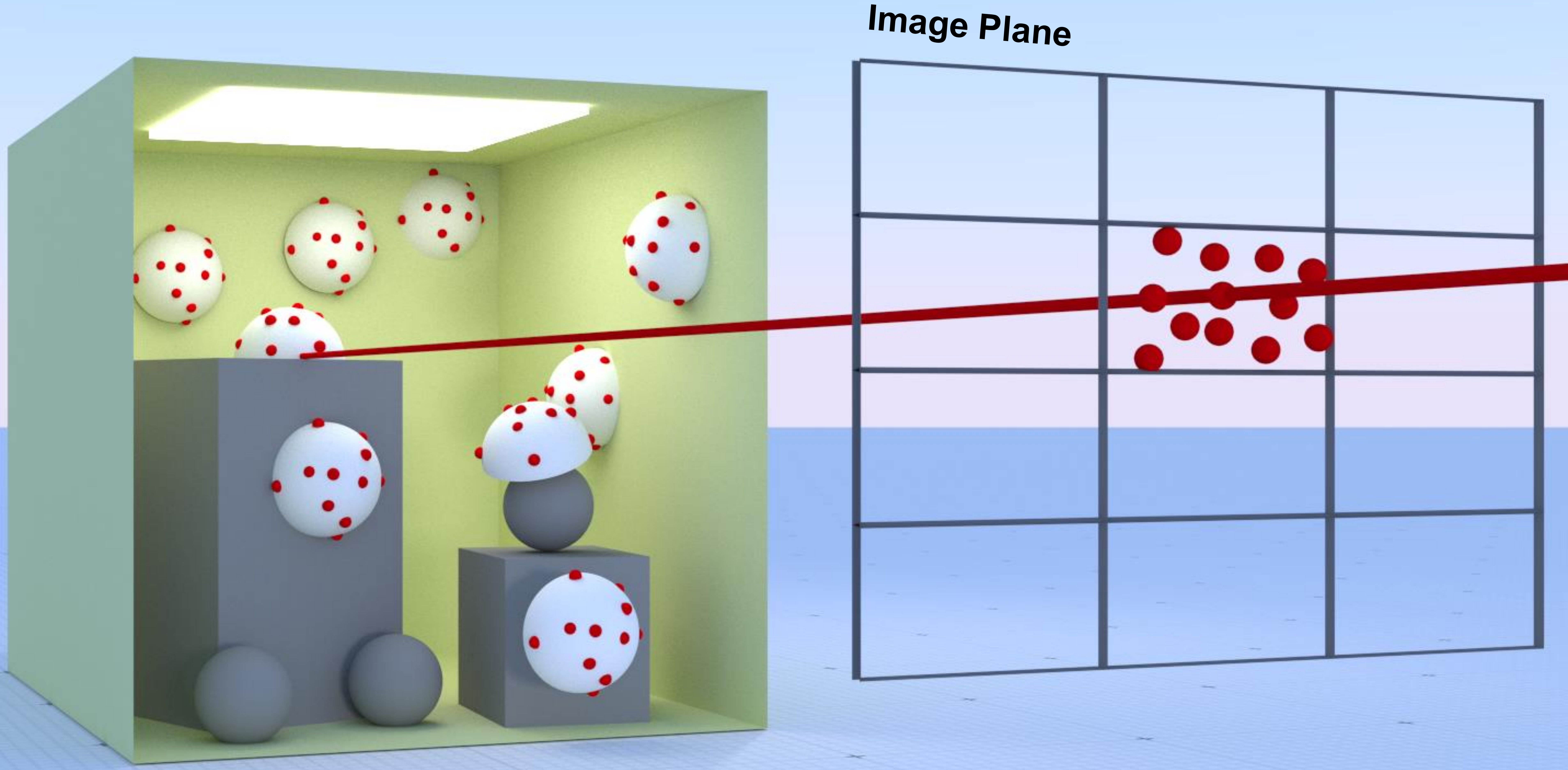
Ray Tracing

Ray Tracing



Scene designed by David Coeurjolly

Ray Tracing



Scene designed by David Coeurjolly

Direct Illumination

4 spp

Image rendered using PBRT

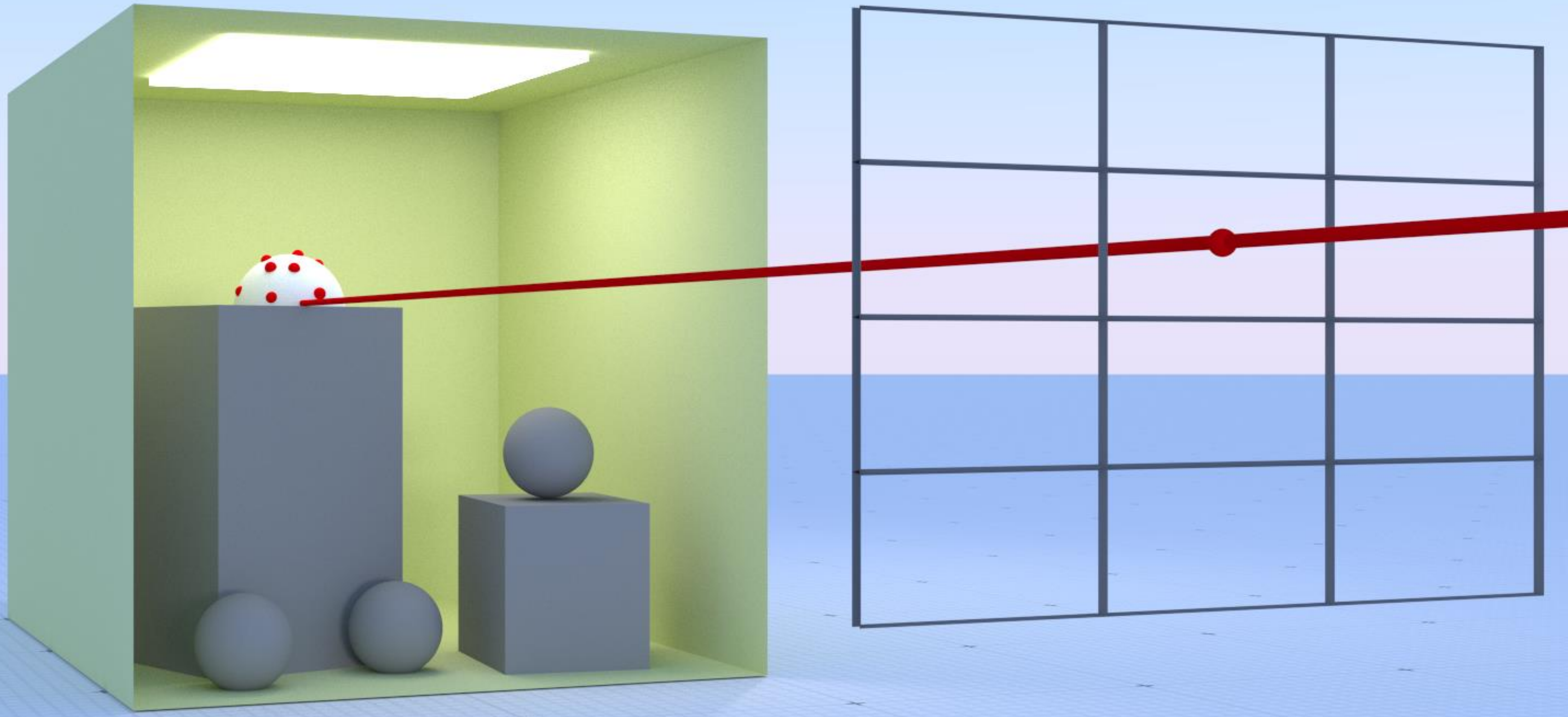
Direct Illumination

Direct and Indirect Illumination

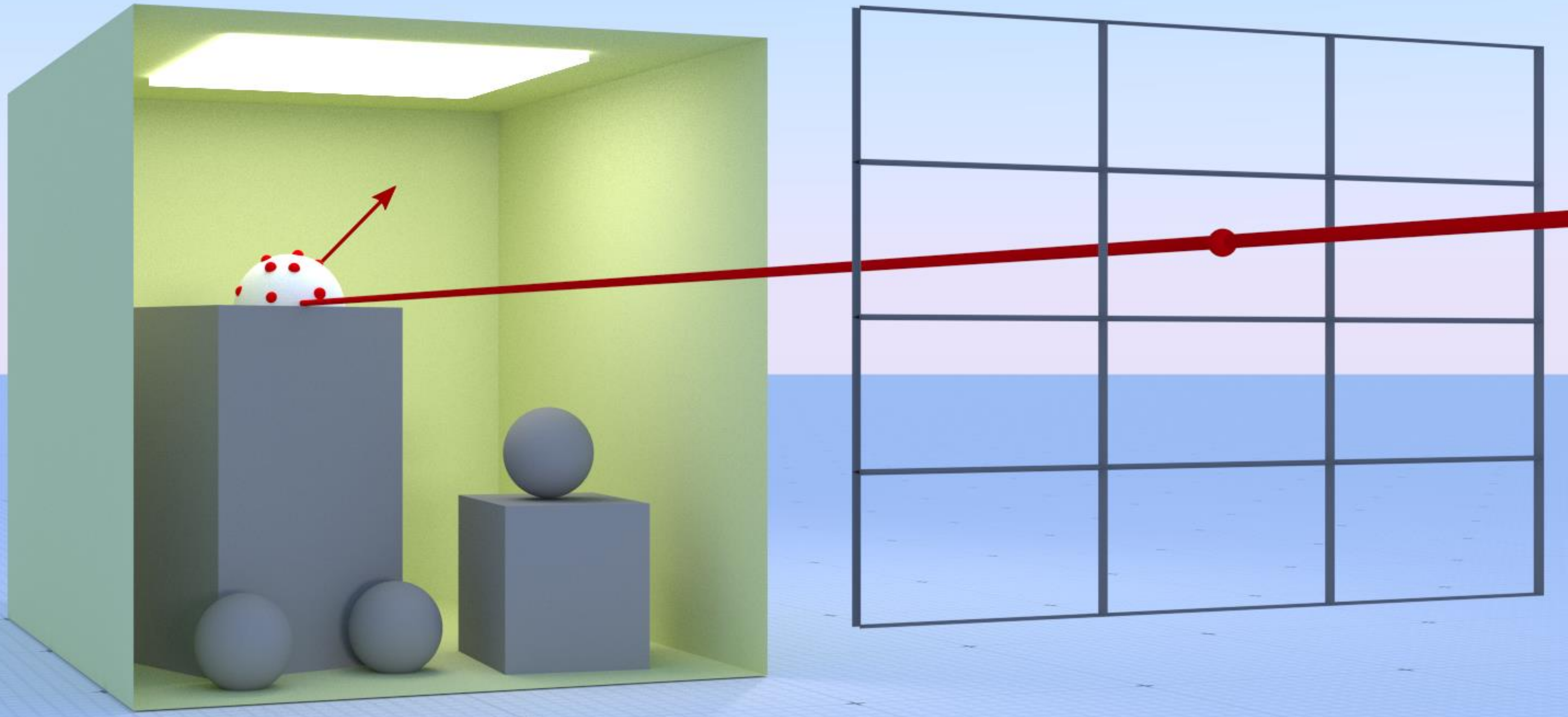
4096 spp

Image rendered using PBRT

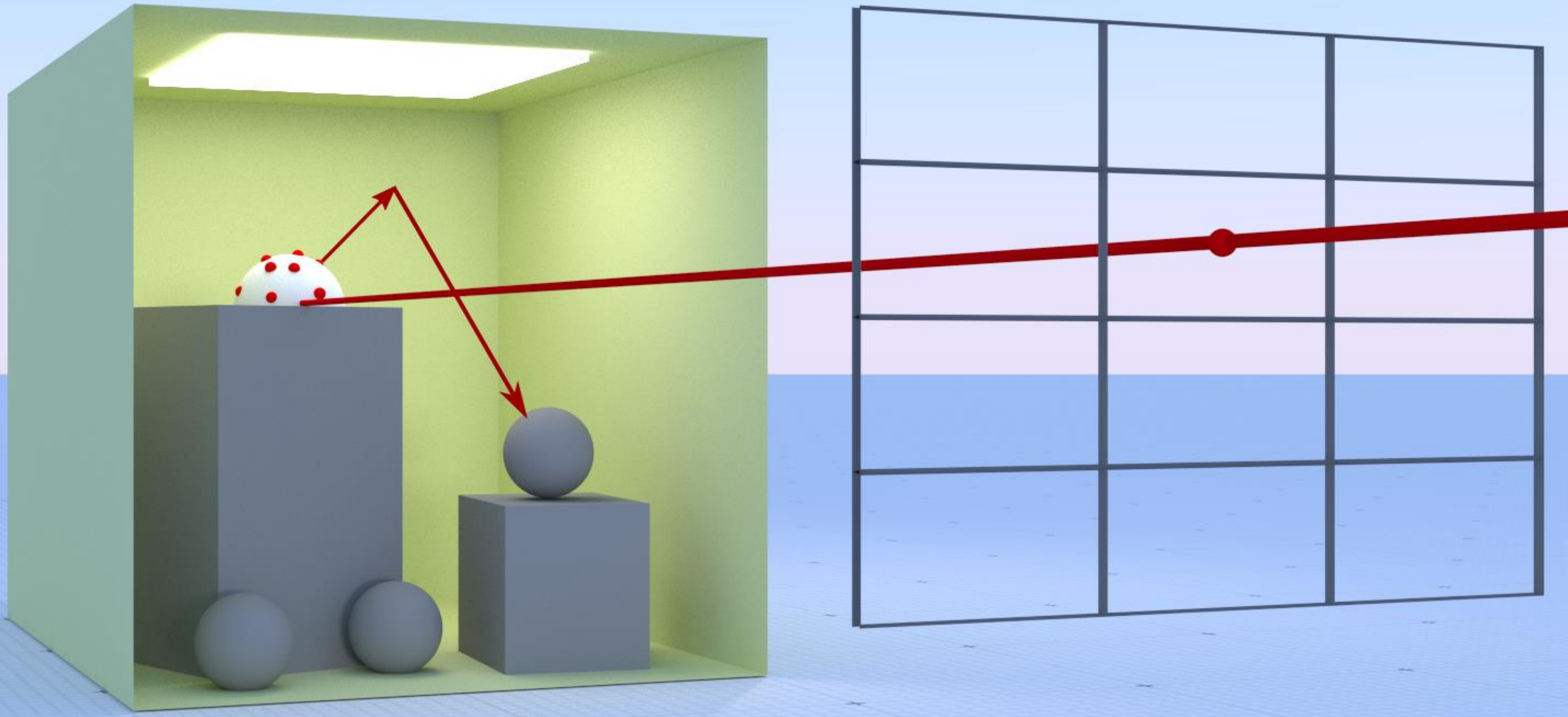
Path Tracing



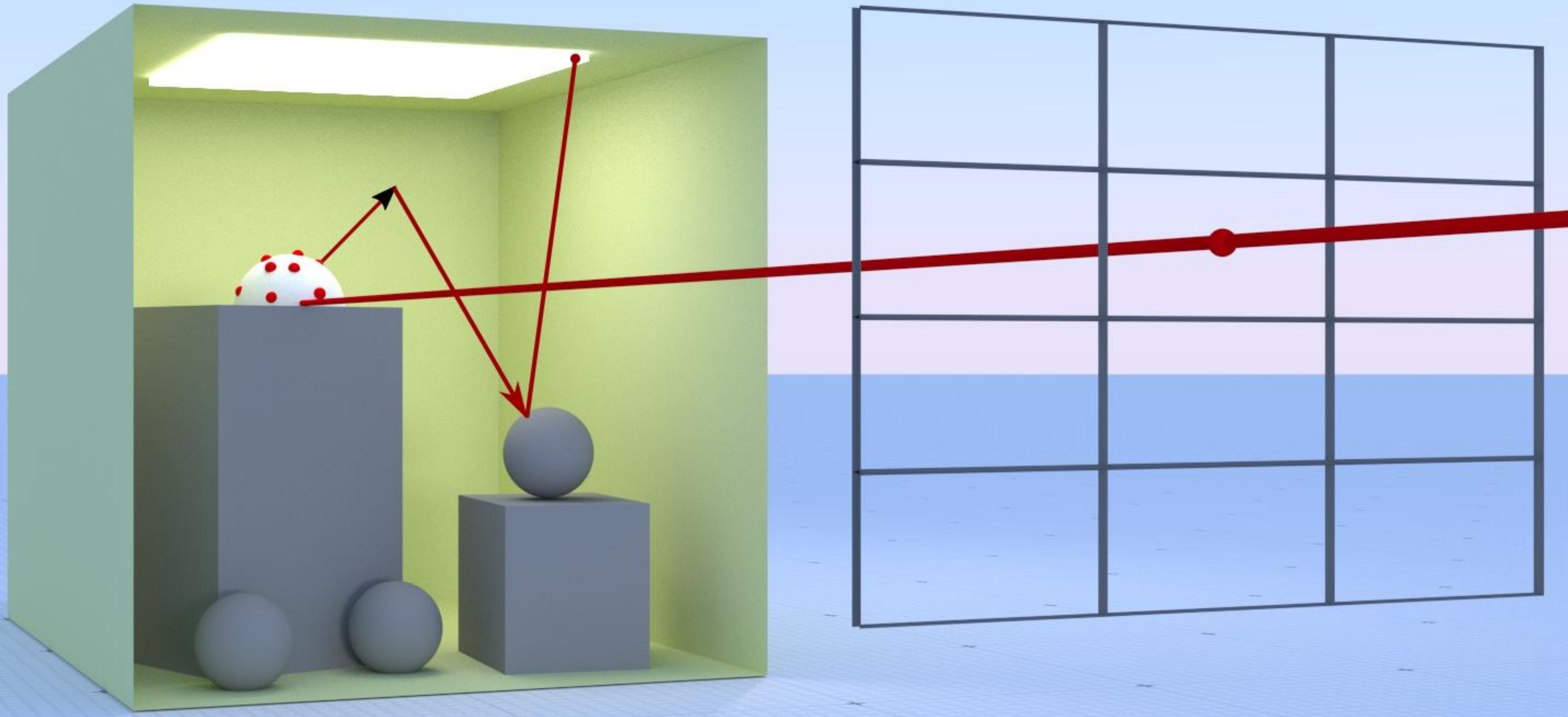
Path Tracing



Path Tracing



Path Tracing

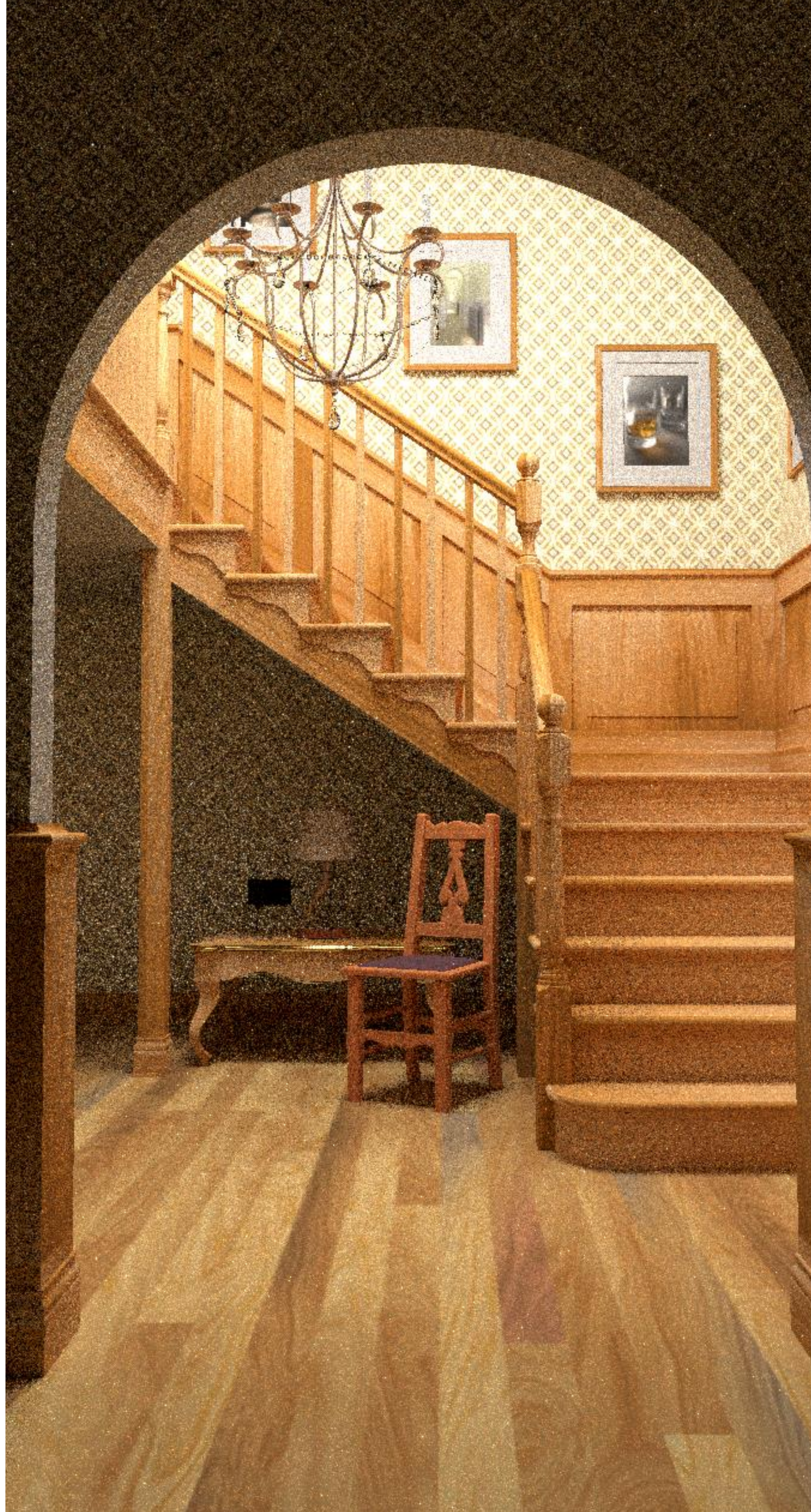


Direct and Indirect Illumination

4 spp

Image rendered using PBRT

Noise

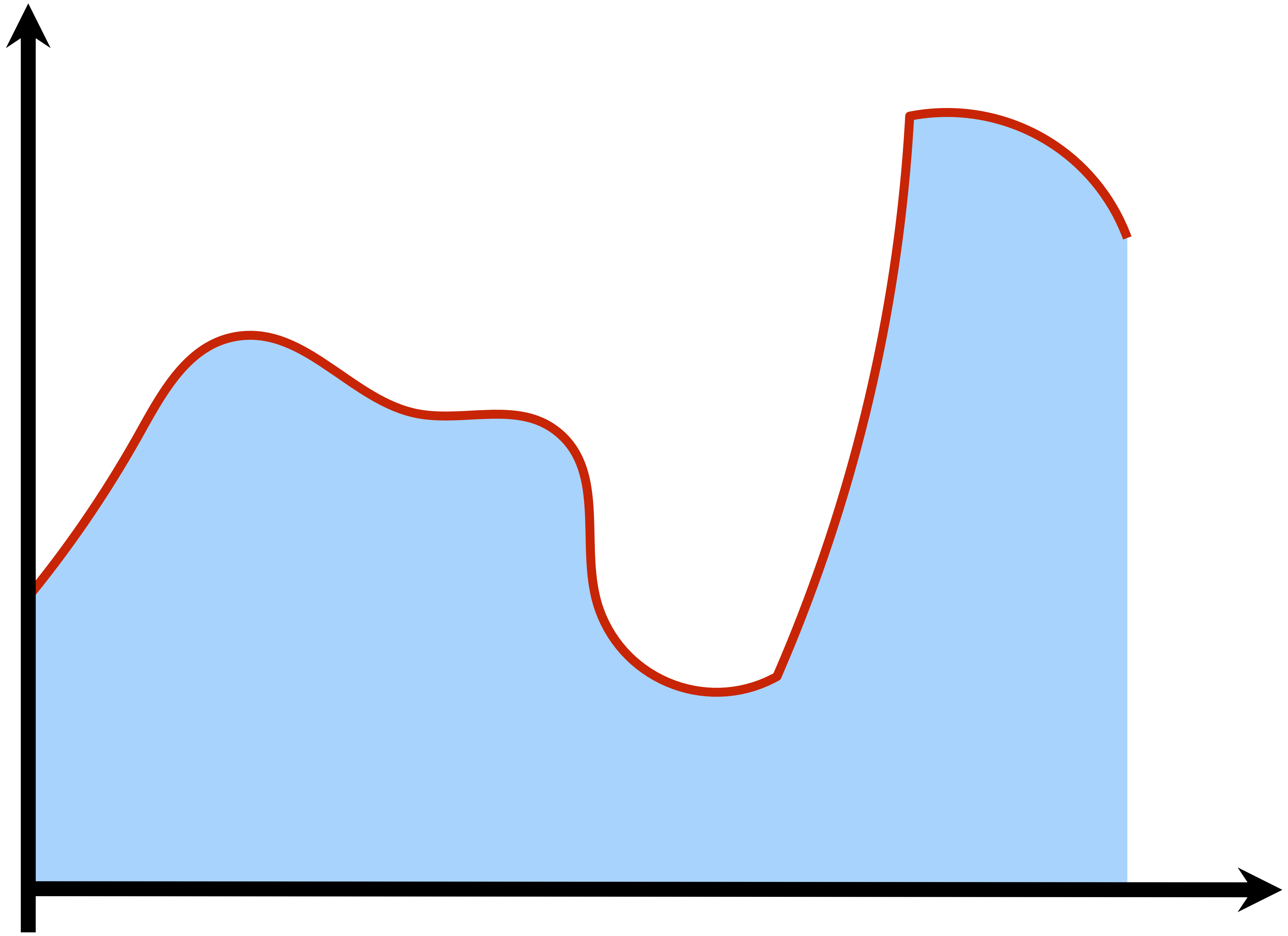


Casino de Monte-Carlo

Copyright Monte Carlo SBM

Monte Carlo Integration

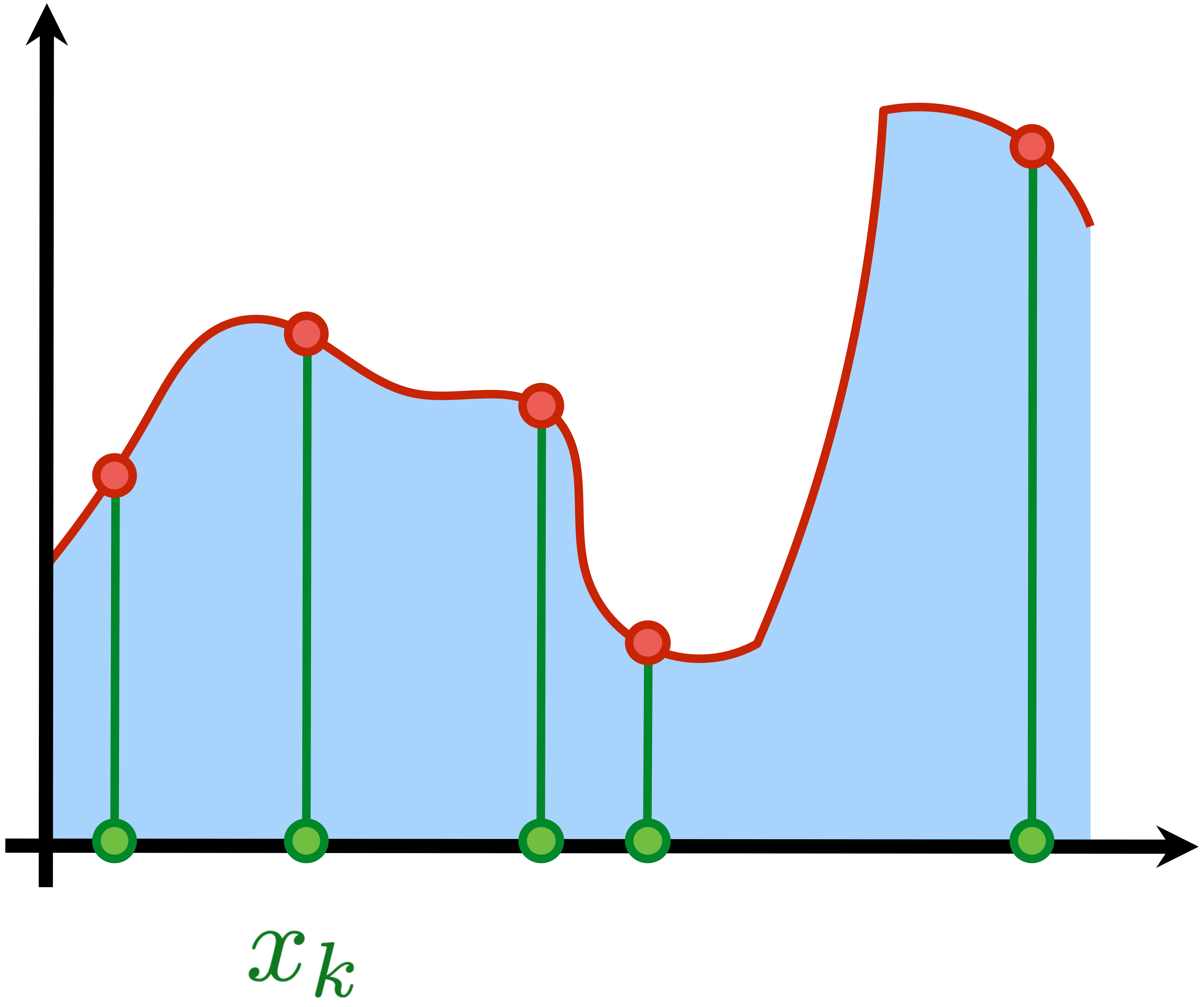
$$I = \int_D f(x)\,\mathrm{d}x$$



Slide after Wojciech Jarosz

Monte Carlo Integration

$$I = \int_D f(x)\,\mathrm{d}x$$

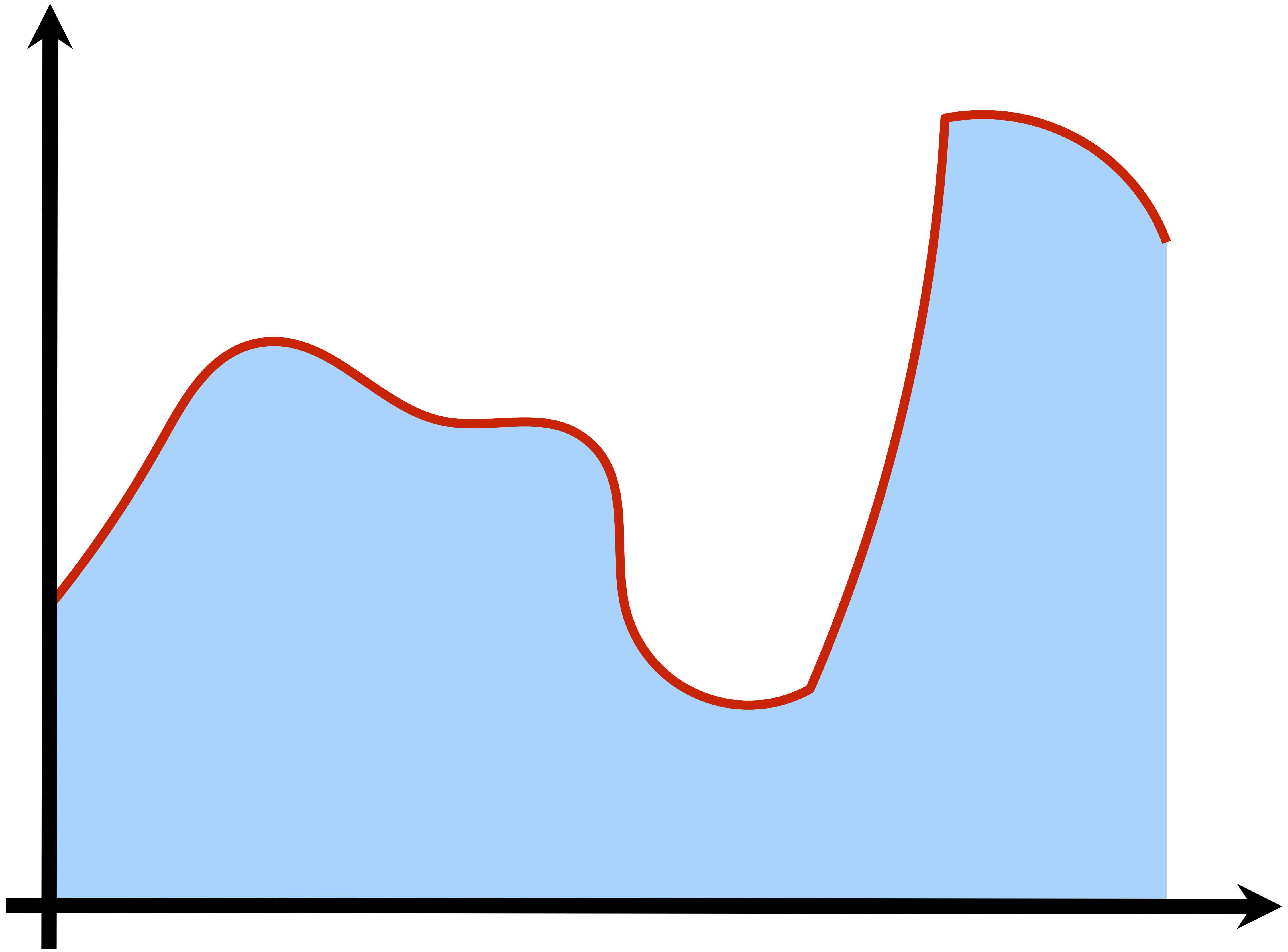


Slide after Wojciech Jarosz

Numerical Integration

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

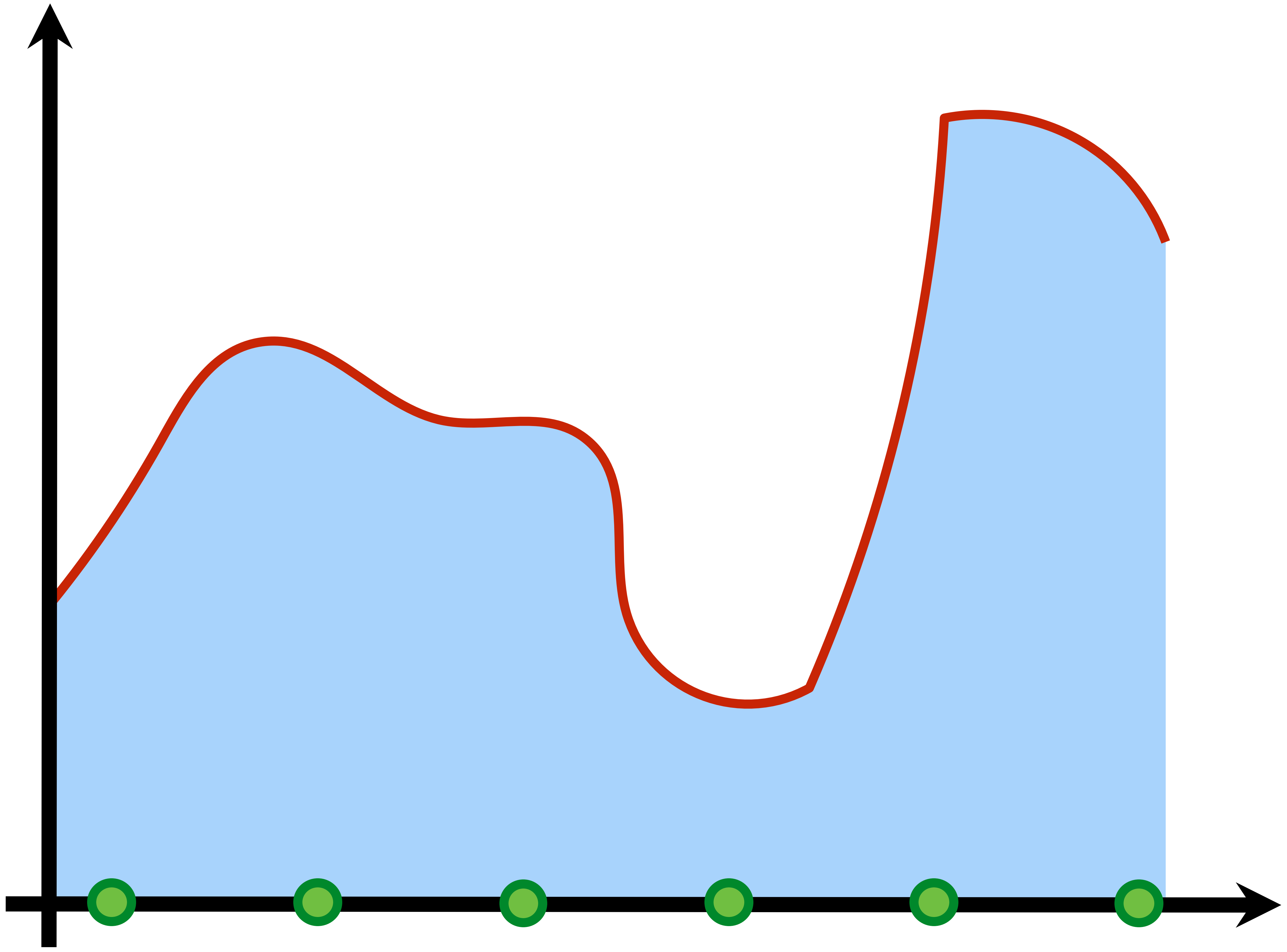
- Analytic evaluation: accurate and fast



Numerical Integration

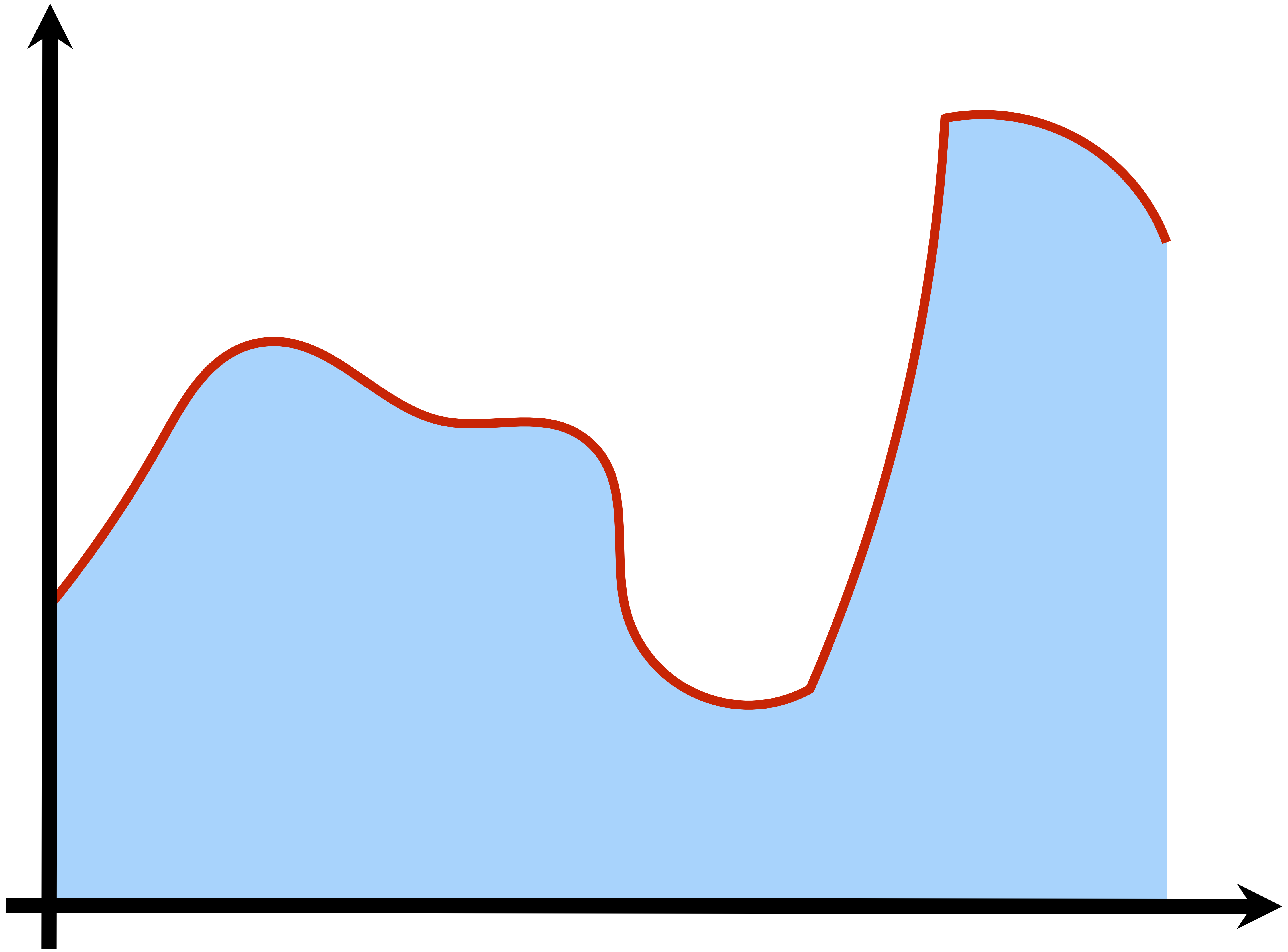
$$\int_a^b f(x)dx$$

- Numerical evaluations:
  - Provide only approximate solutions,
  - Rate of convergence is important
  - Often involves evaluations only at selected locations



Numerical Integration

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



- Numerical quadrature: designed for 1D integrals
- Cubature/Quadratures: for higher dimensions

Numerical Integration

- Hybrid methods: First transform the integral analytically for simpler numerical handling

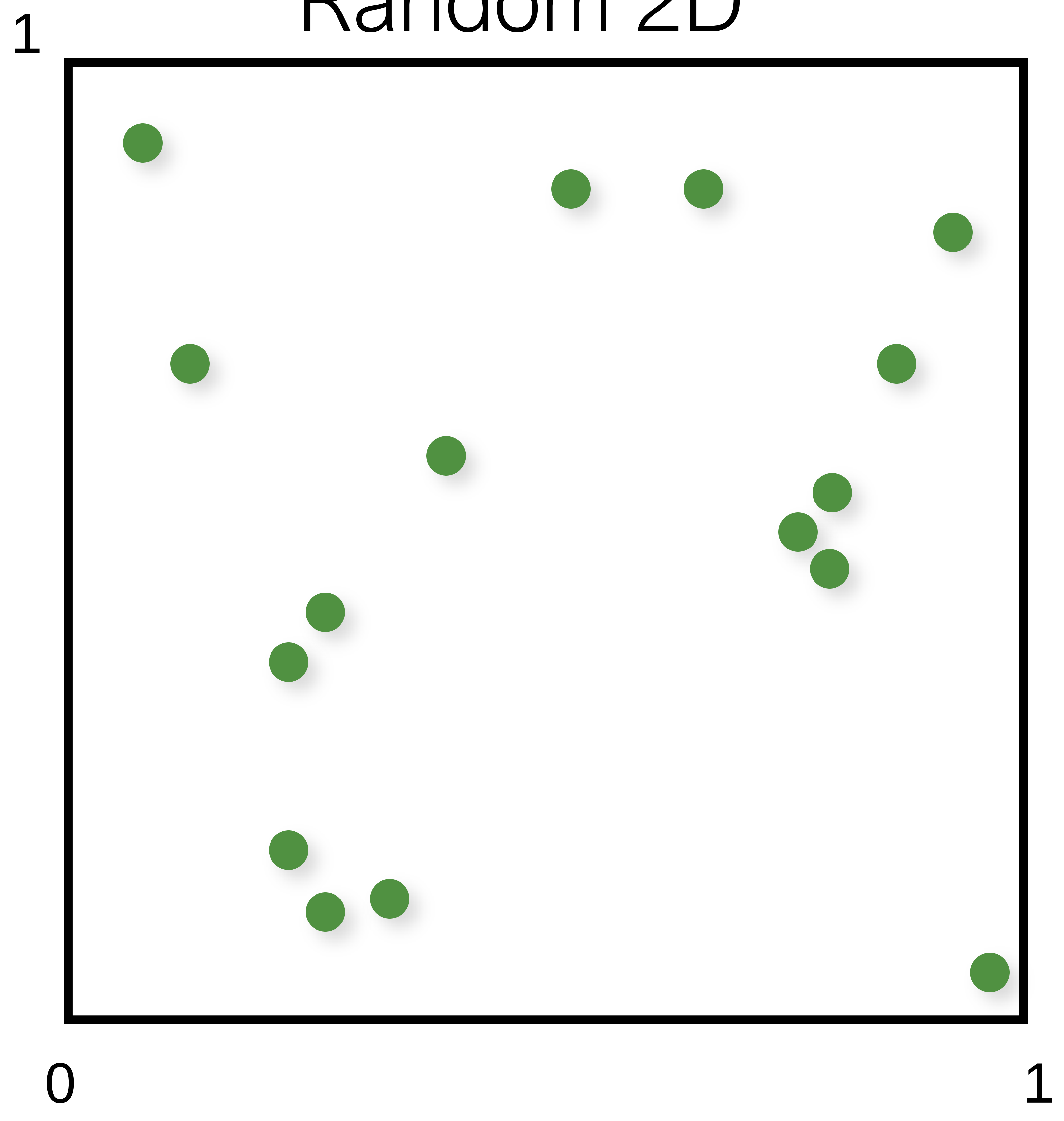
Variance Reduction Techniques

- Correlated Sampling
- Importance Sampling

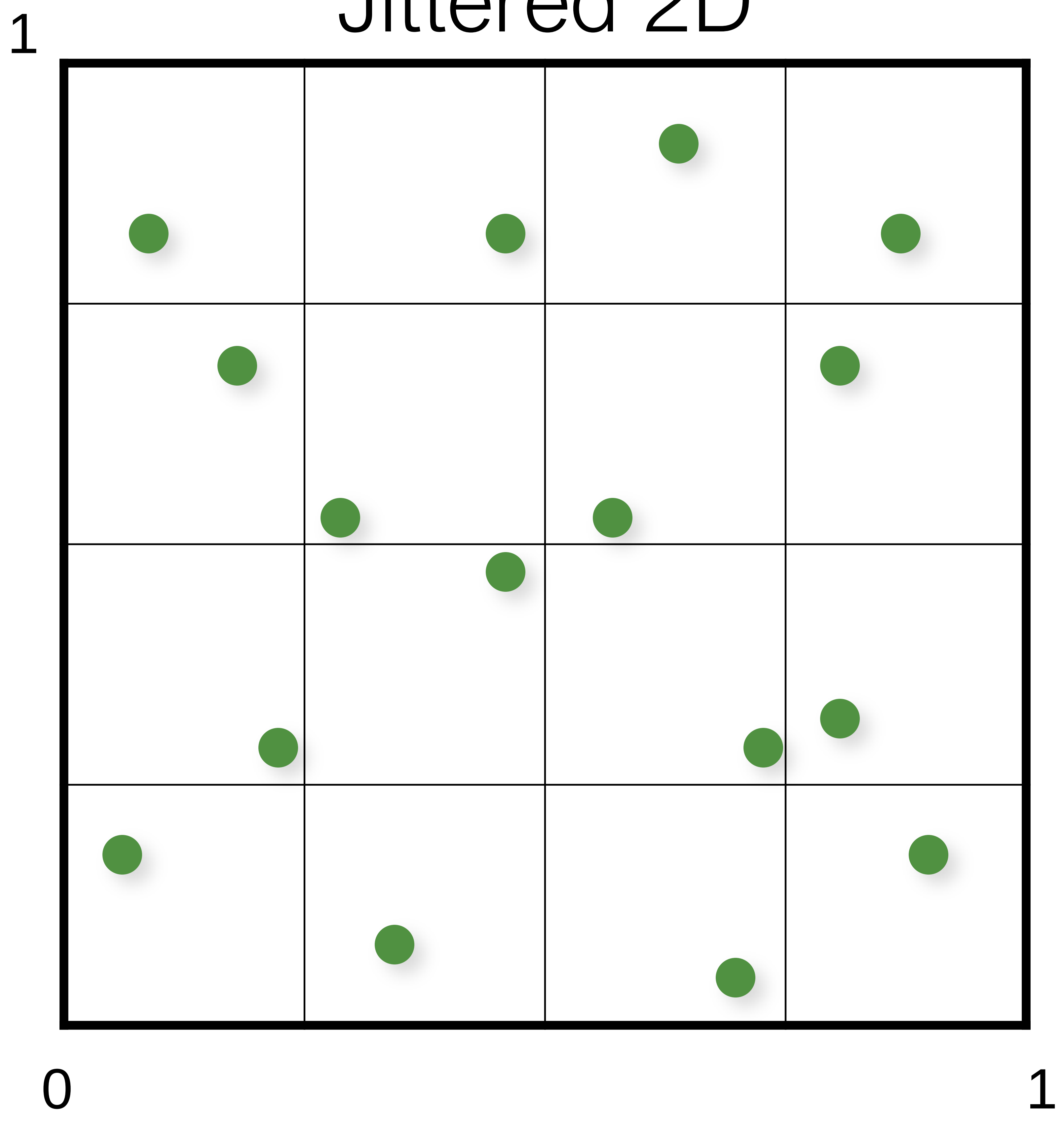
Correlated Sampling: Jittered Sampling

Variance reduction: Stratified Sampling

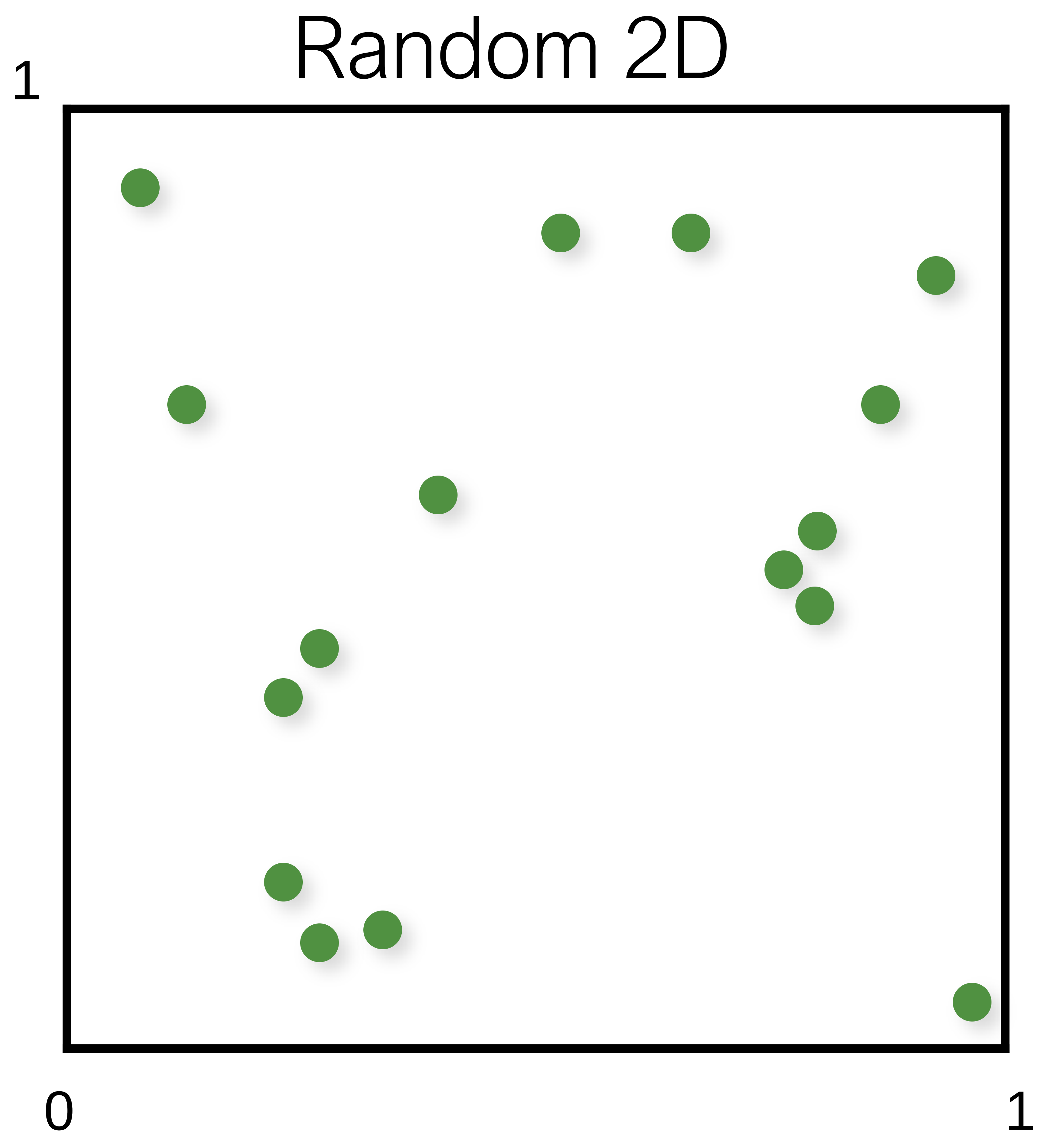
Random 2D

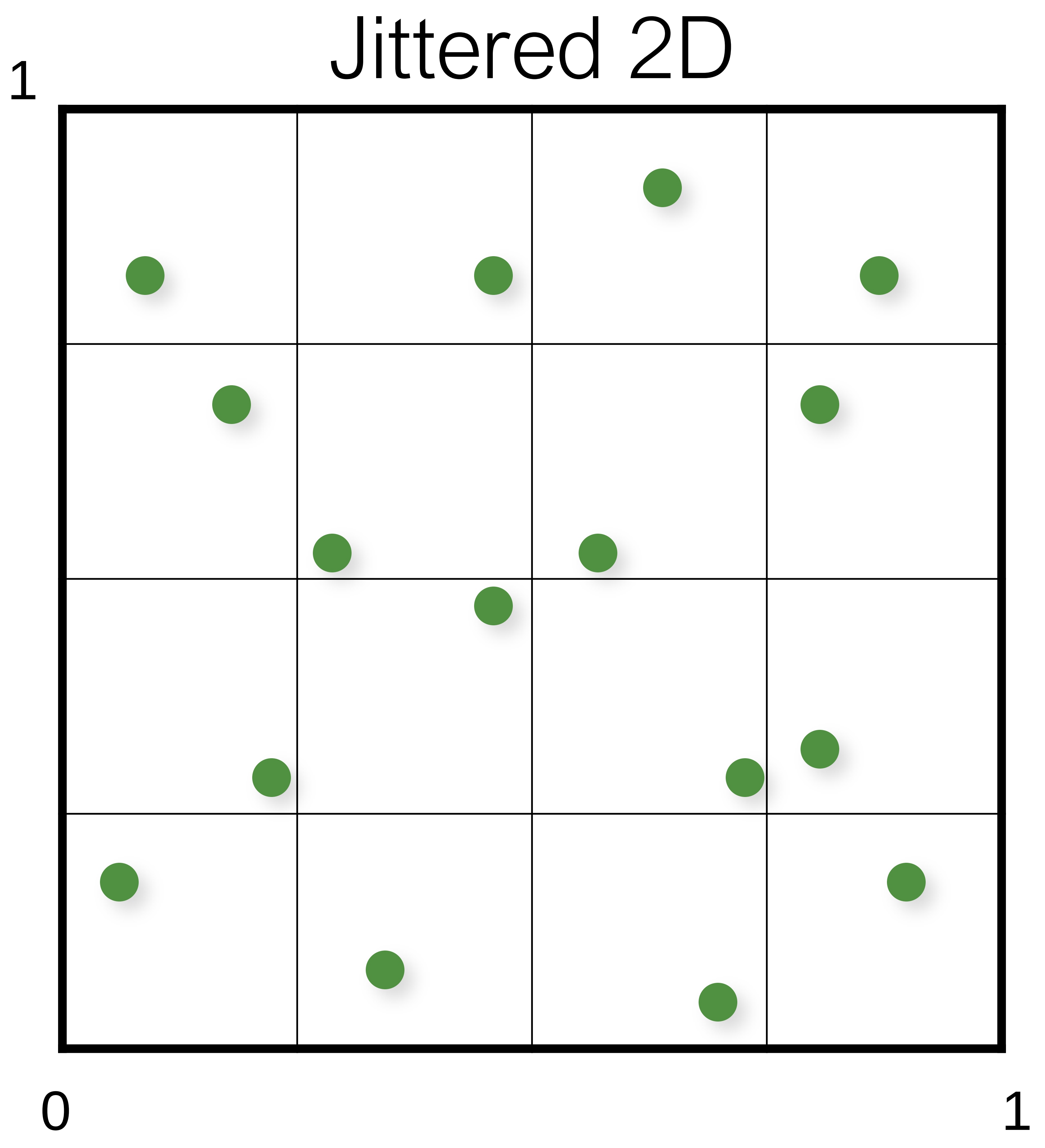


Jittered 2D

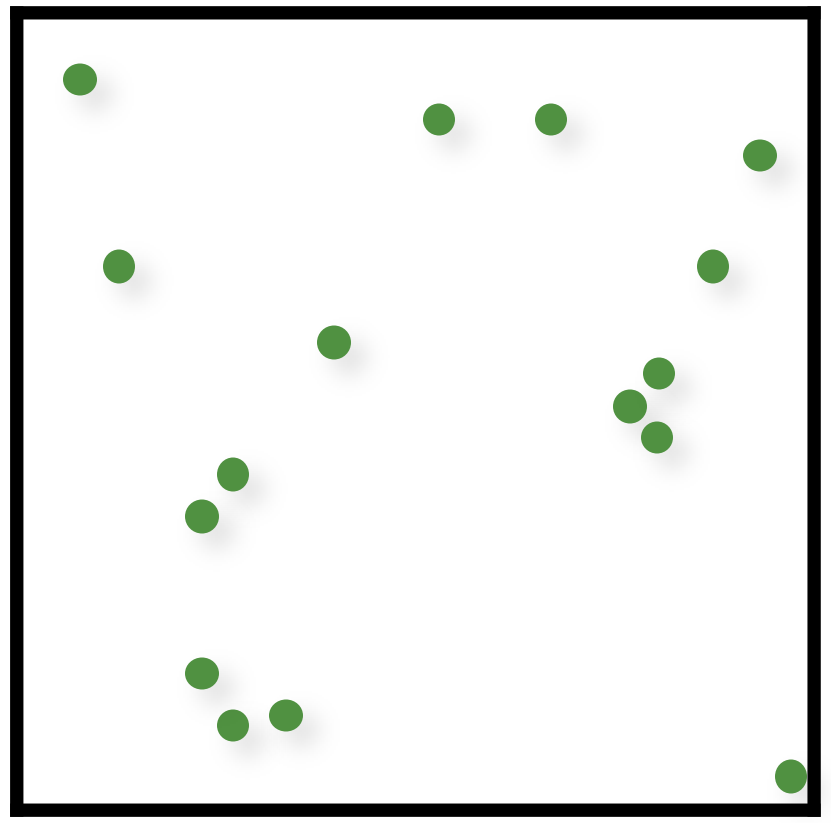


Variance reduction: Stratified Sampling

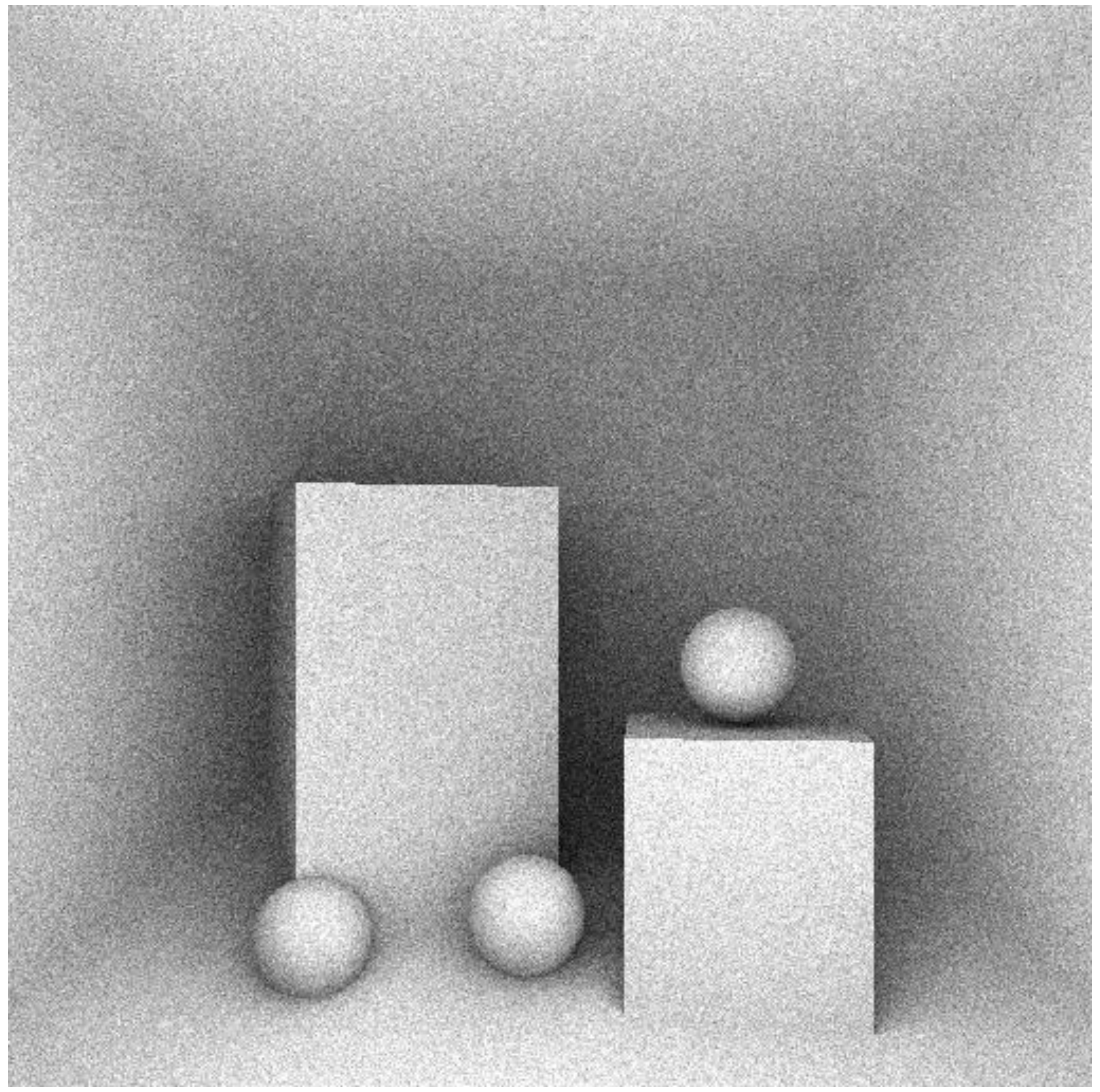




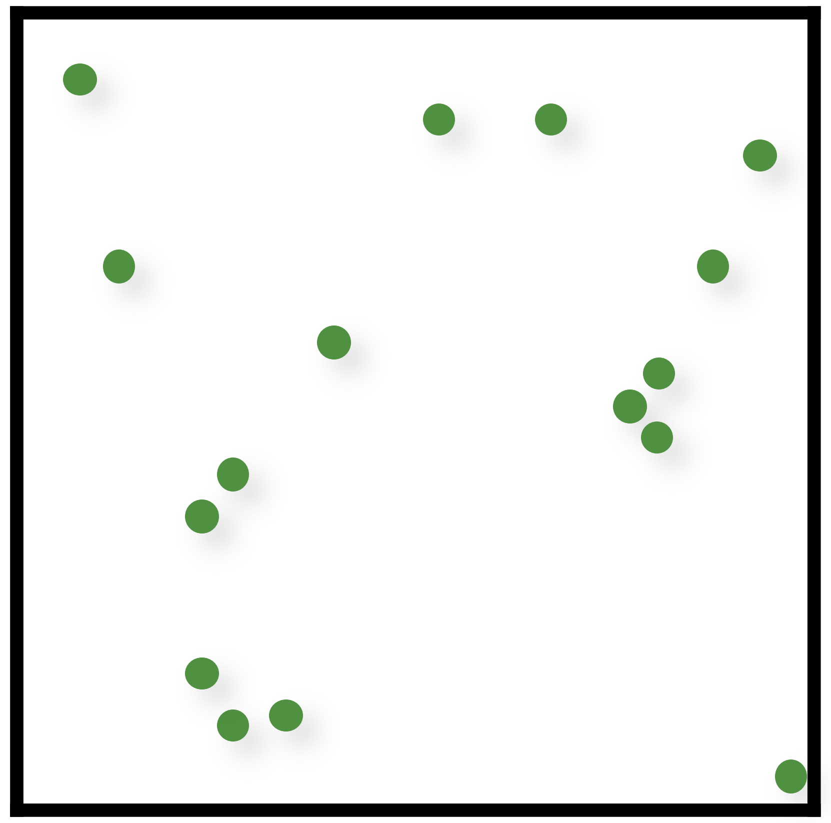
Random vs. Stratified Sampling



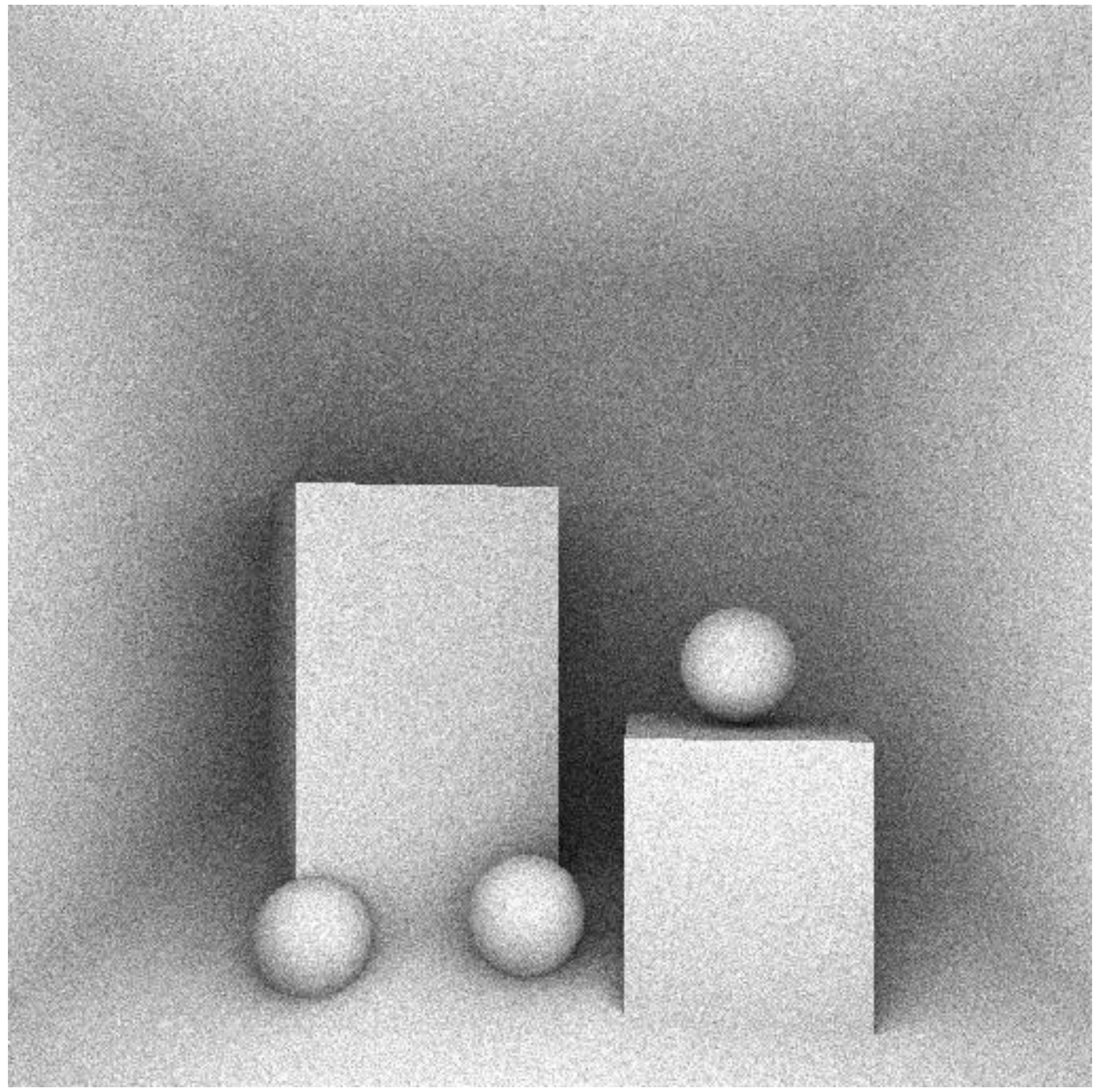
Random Samples



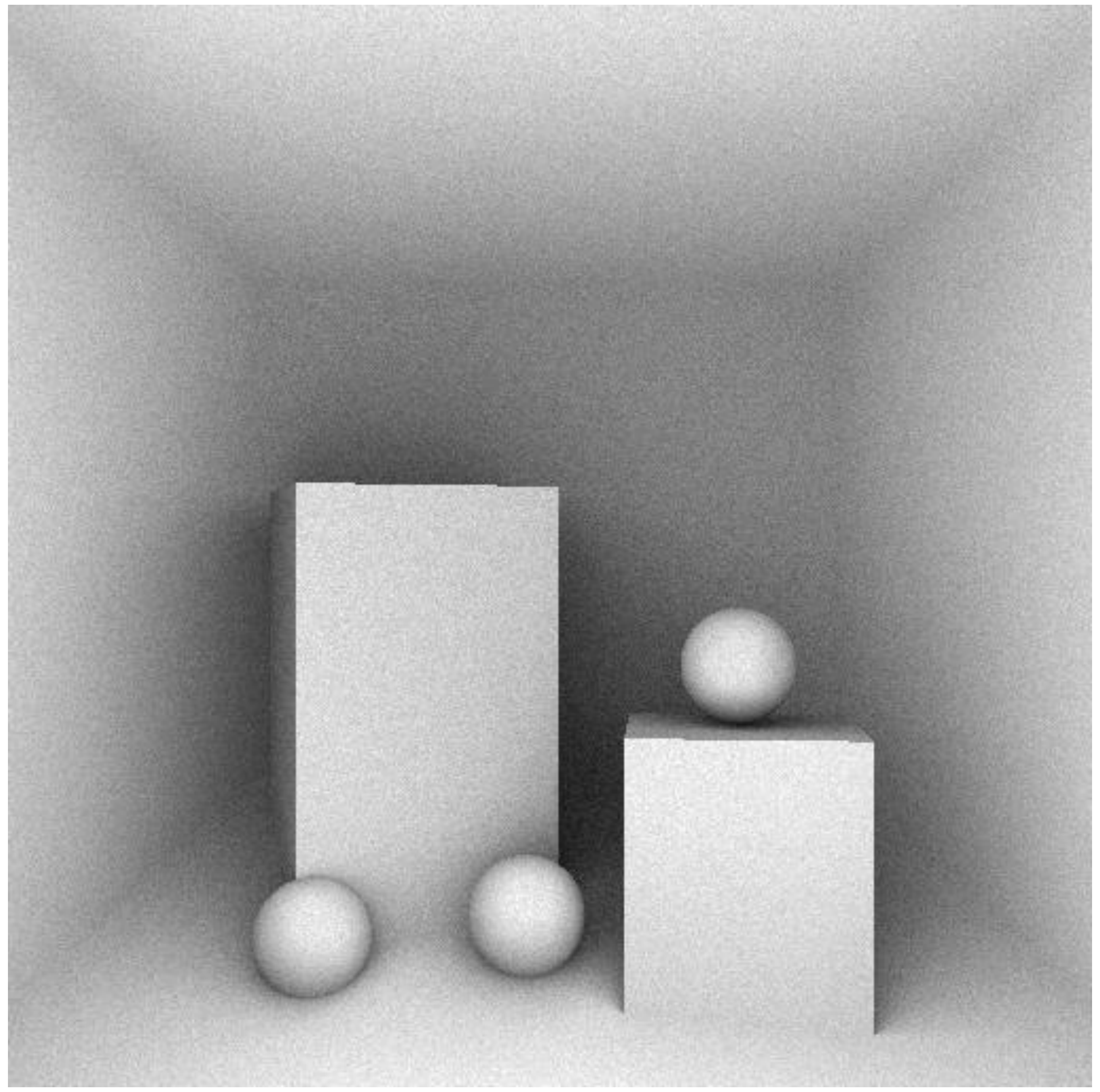
Random vs. Stratified Sampling

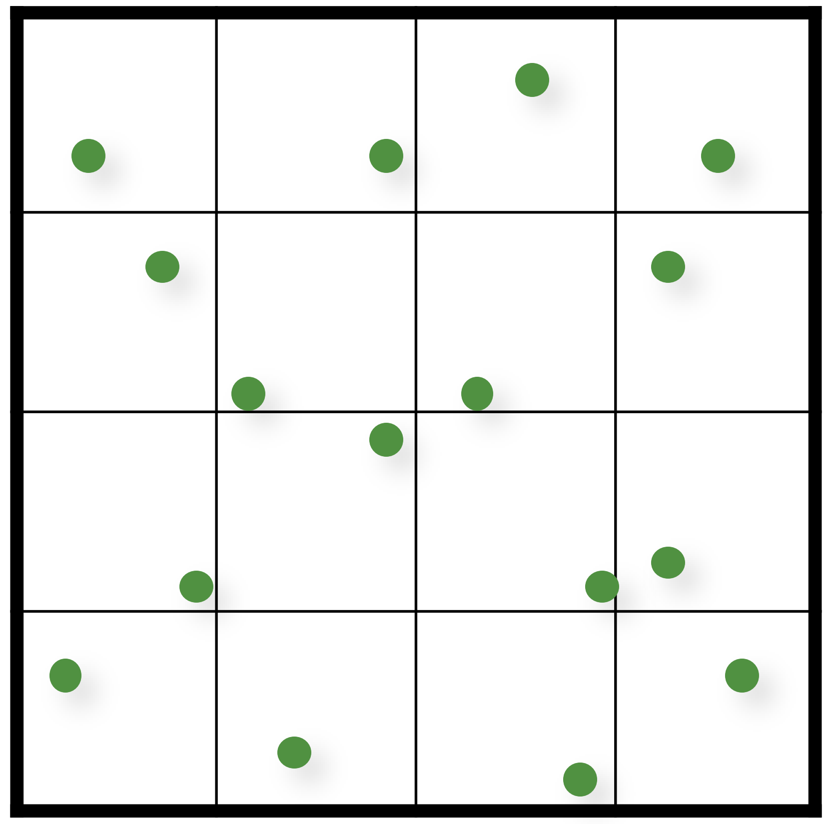


Random Samples



Jittered Samples



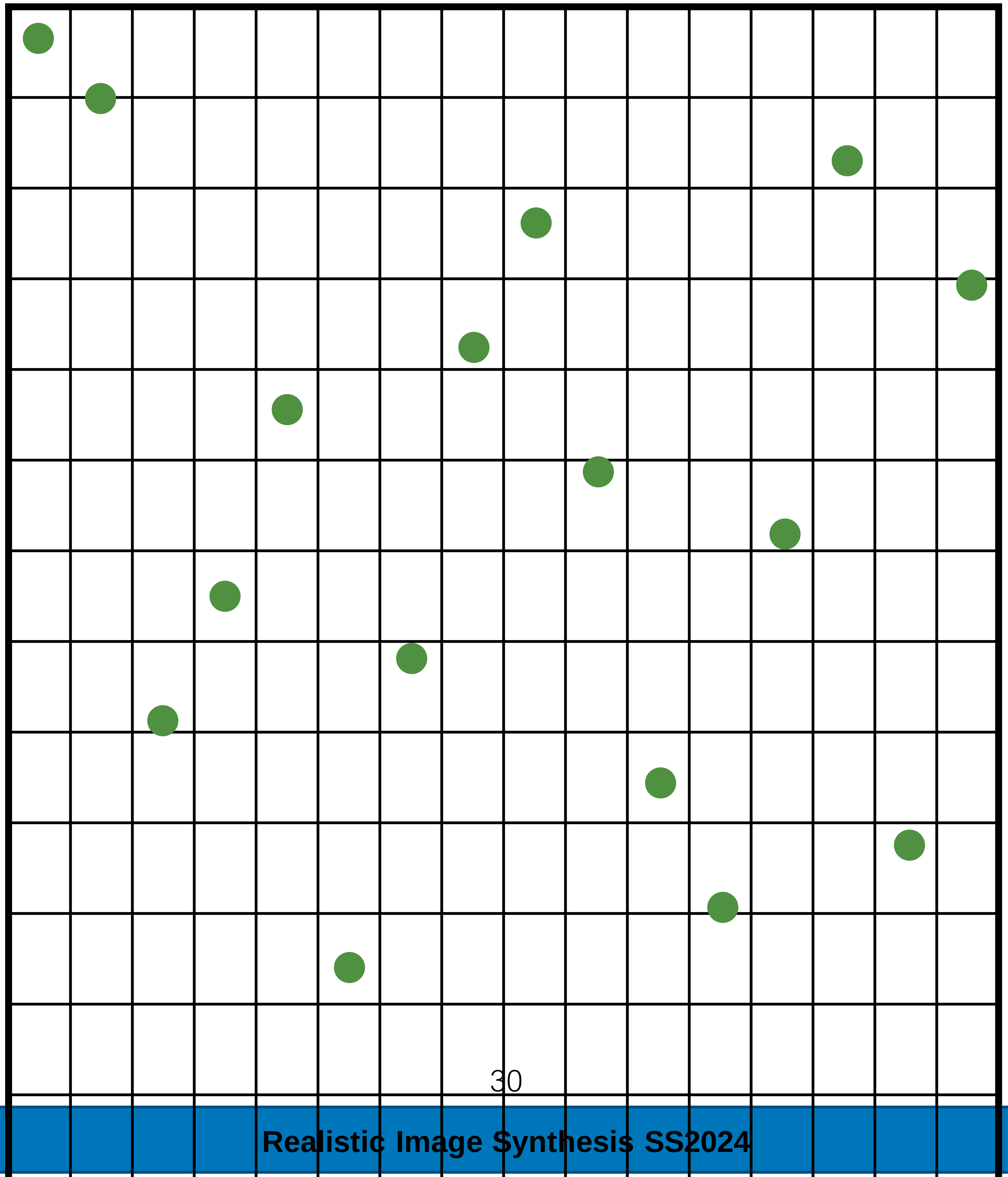


N = 64 spp

Stratified sampling suffers from the curse of dimensionality

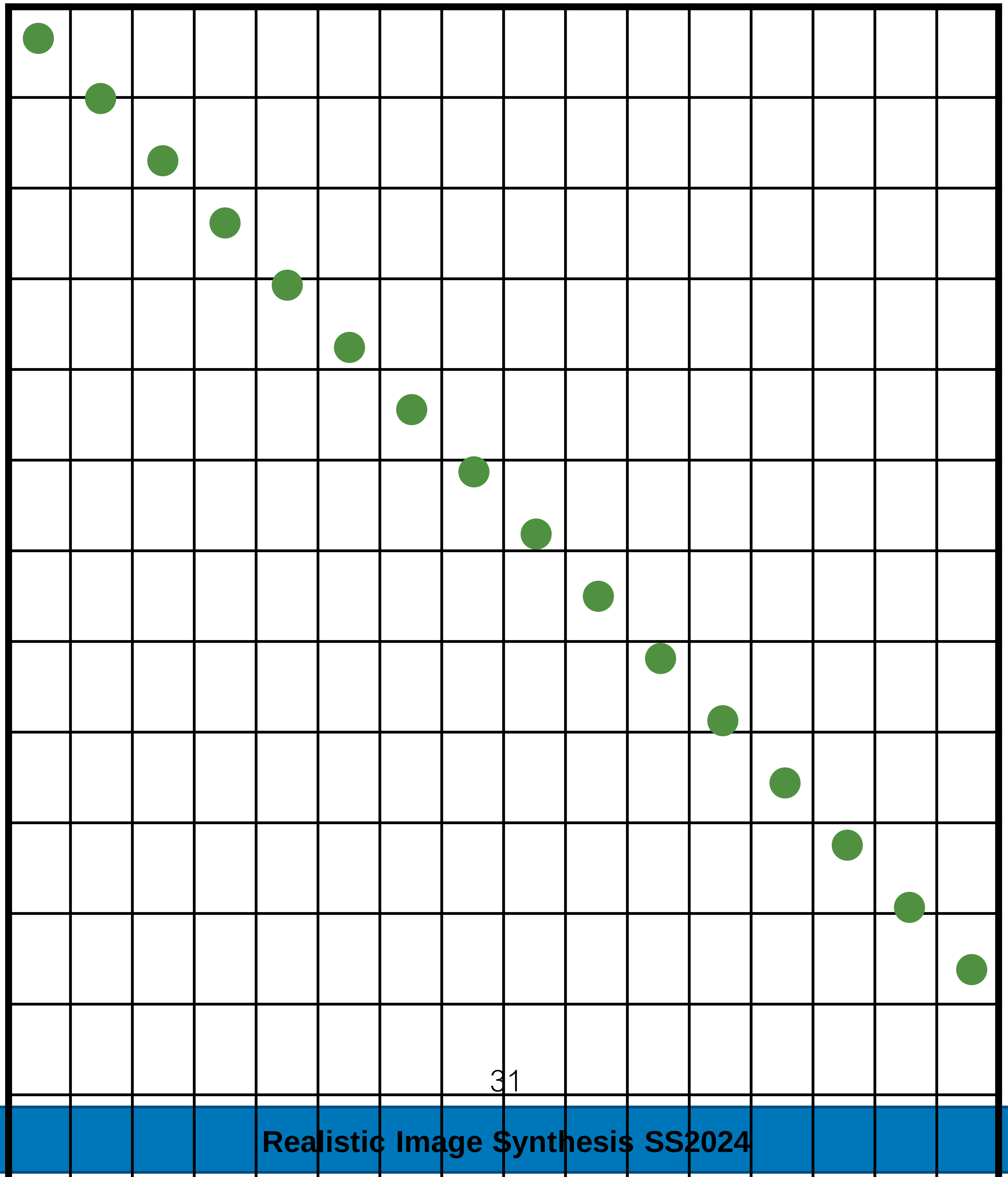
Correlated Sampling: Latin Hypercube Sampling

Latin Hypercube Sampler (N-rooks)



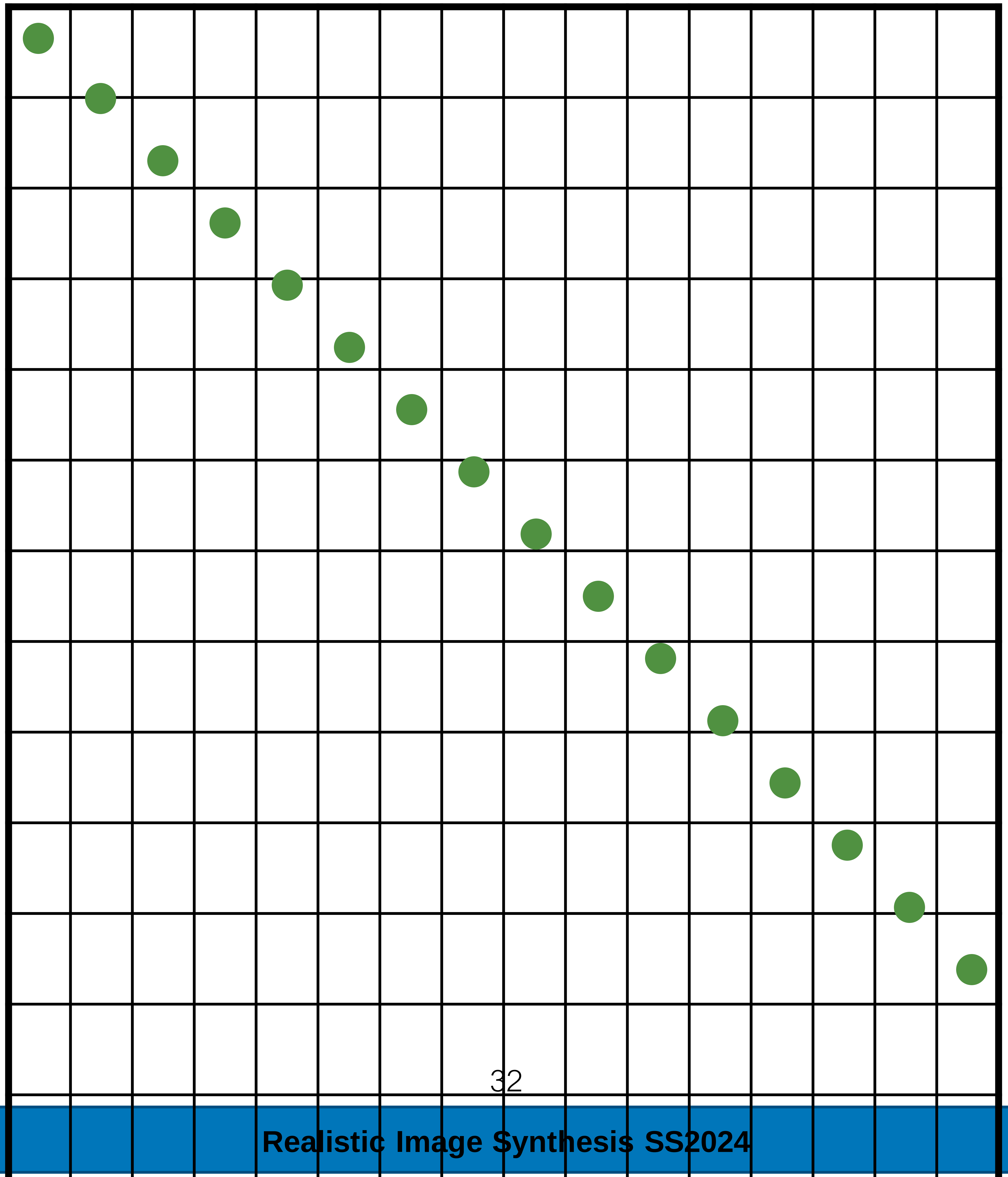
Latin Hypercube Sampler (N-rooks)

Initialize

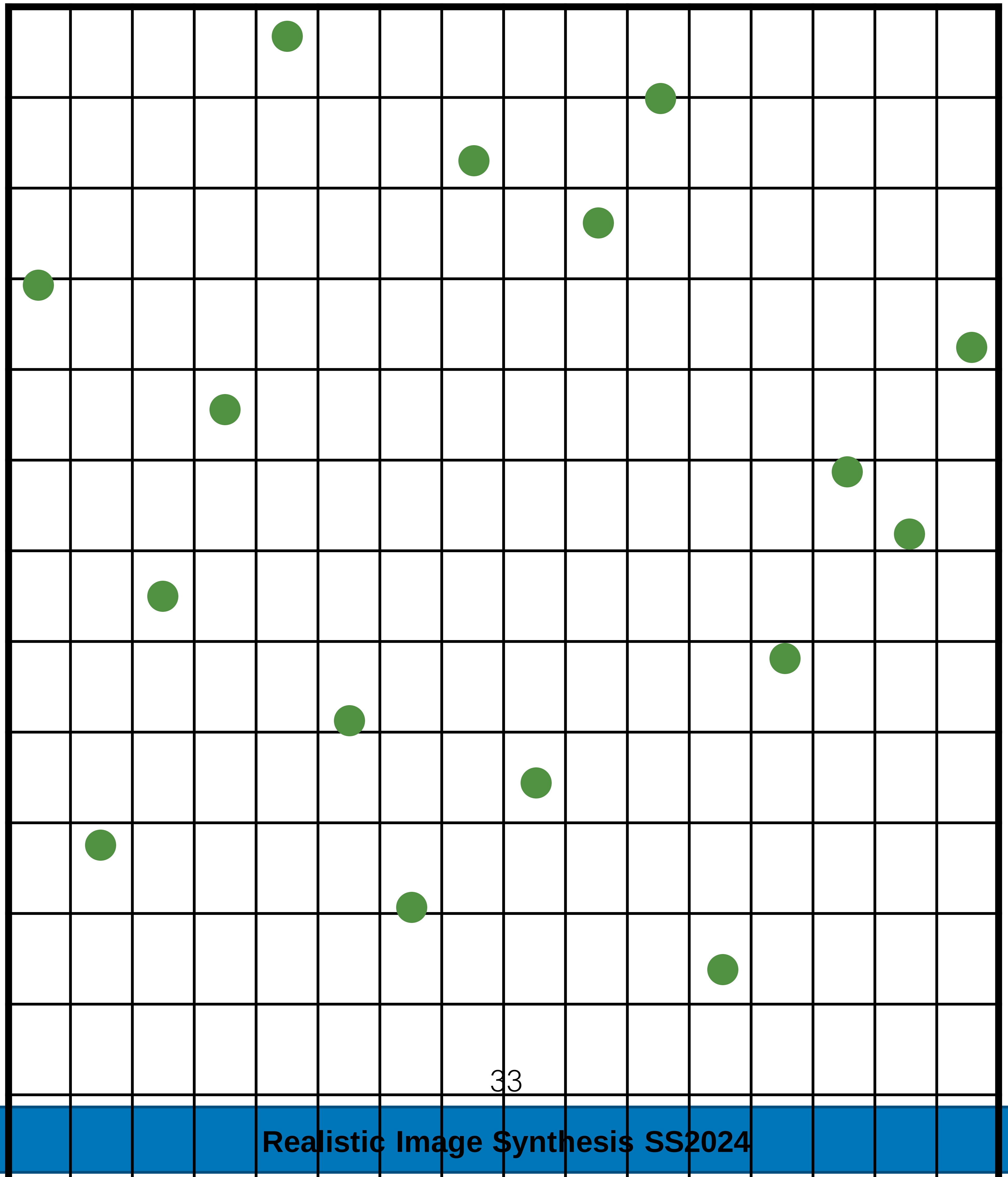


Latin Hypercube Sampler (N-rooks)

Shuffle rows

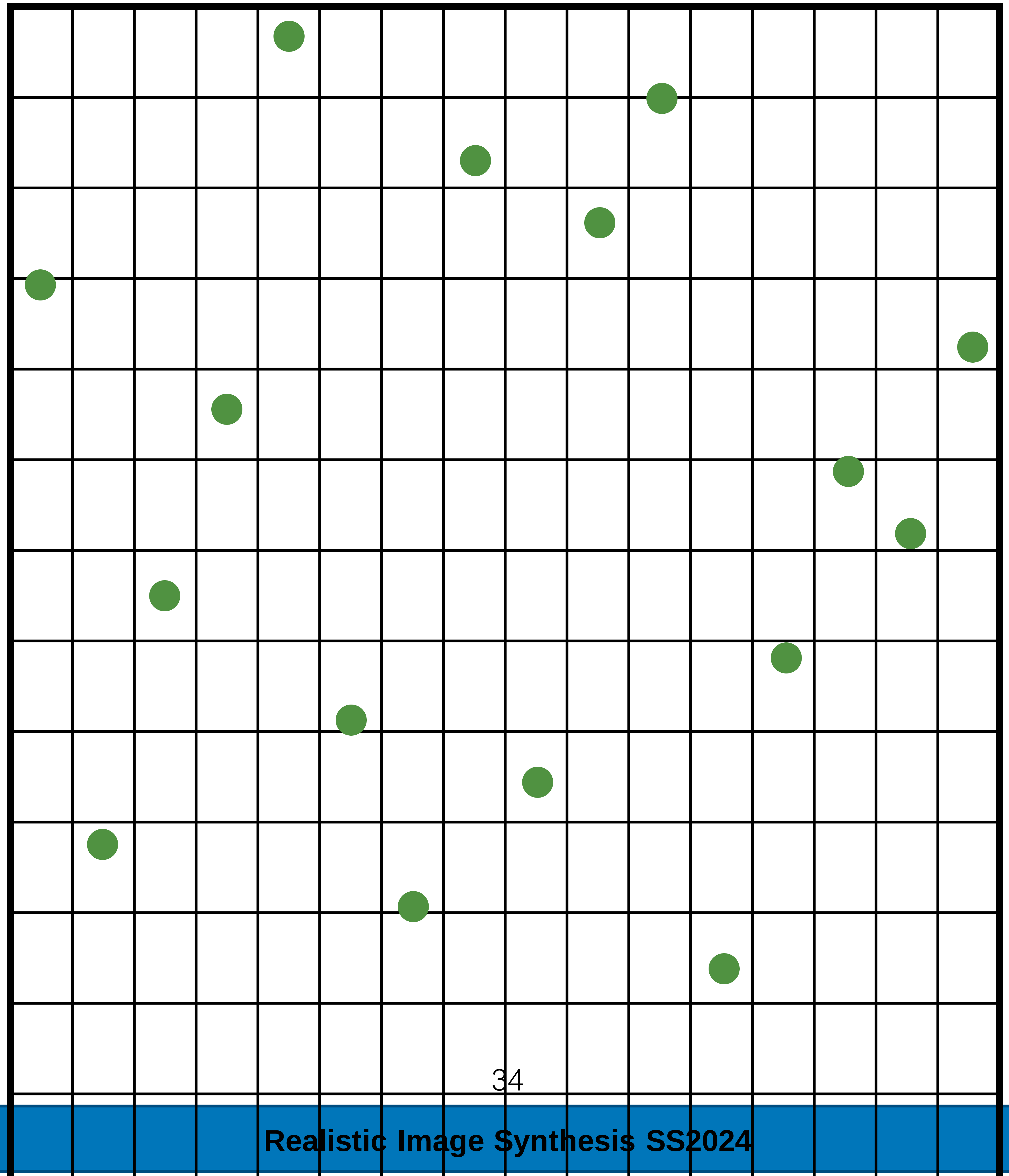


Latin Hypercube Sampler (N-rooks)

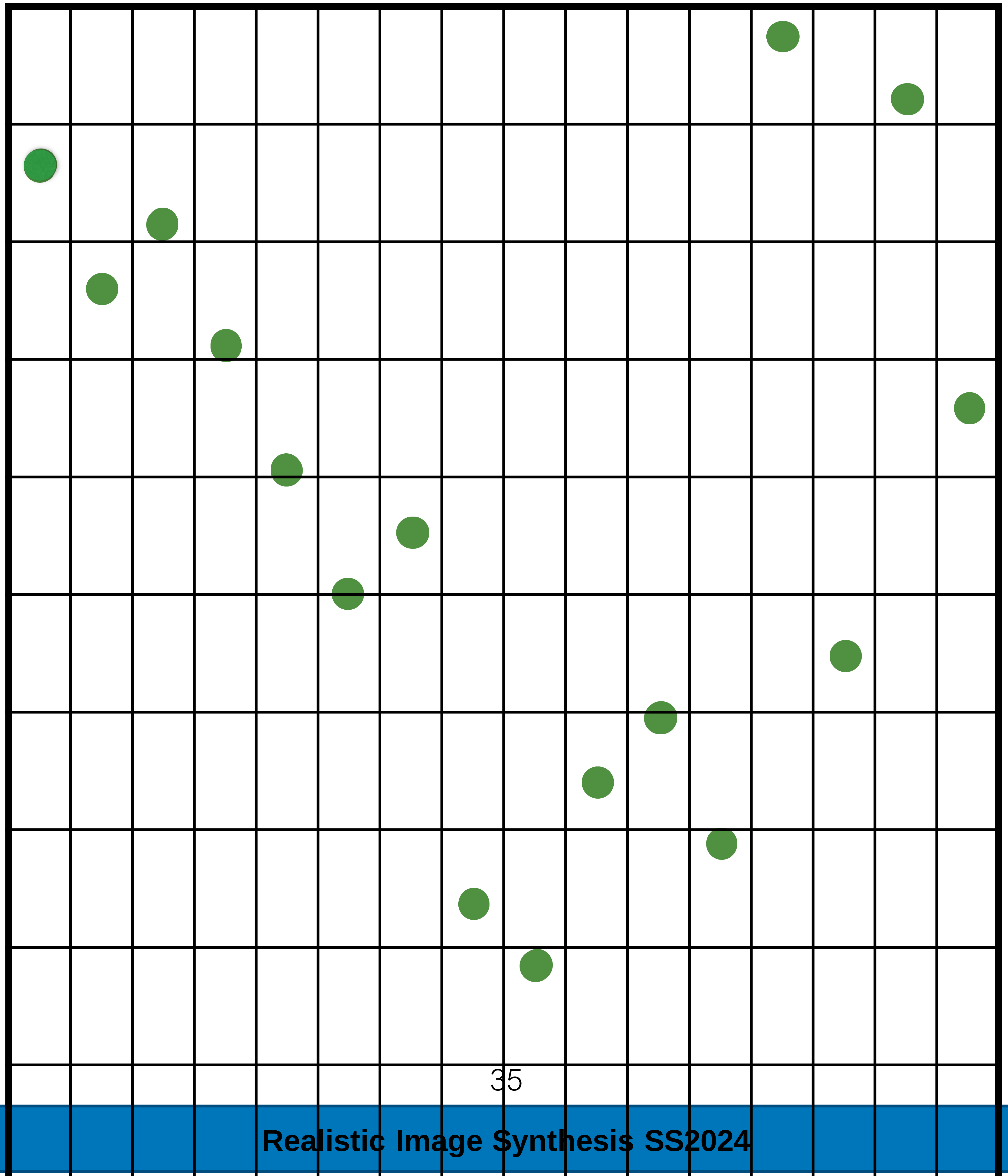


Latin Hypercube Sampler (N-rooks)

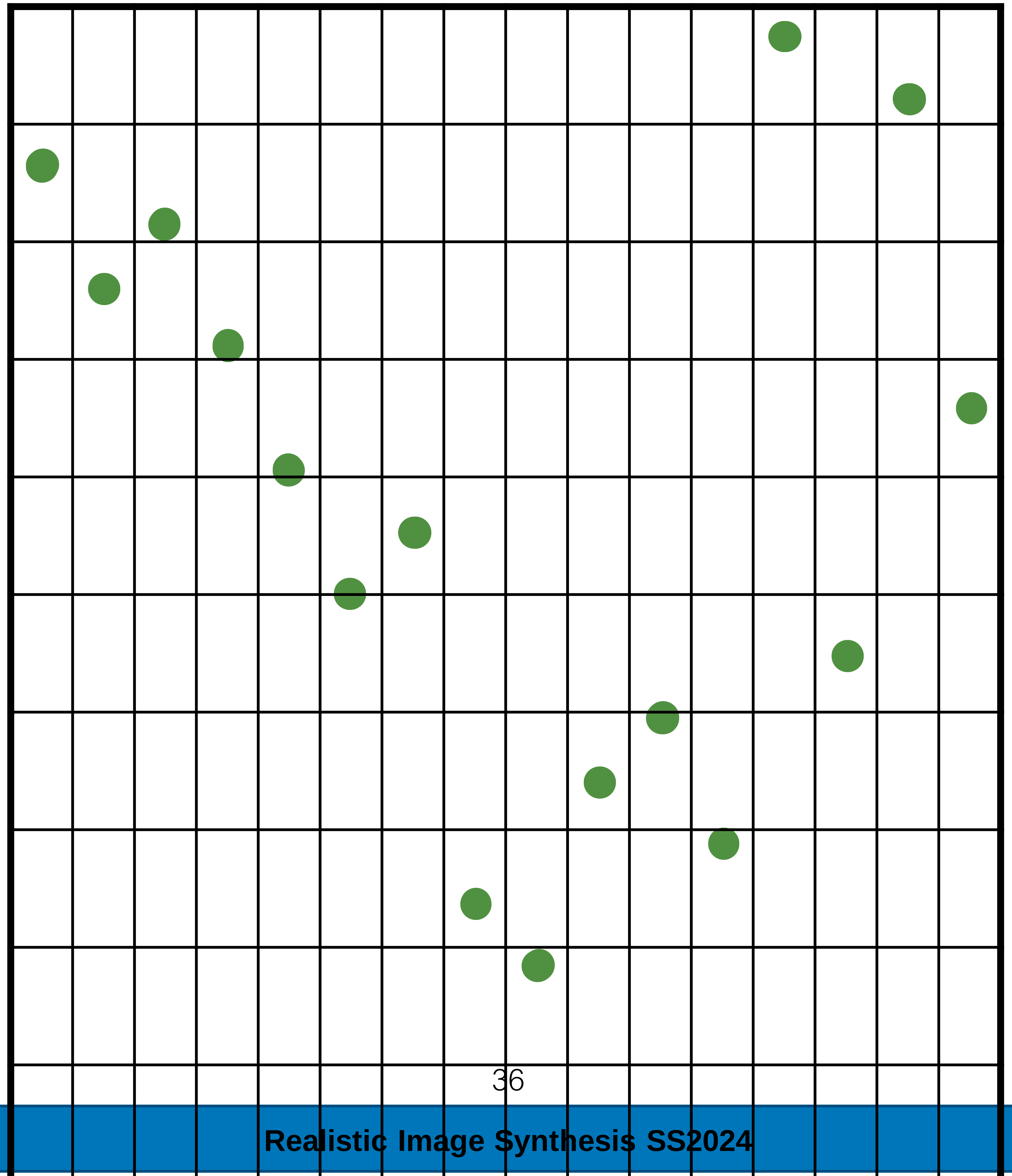
Shuffle columns



Latin Hypercube Sampler (N-rooks)



Latin Hypercube Sampler (N-rooks)



Variants of stratified sampling

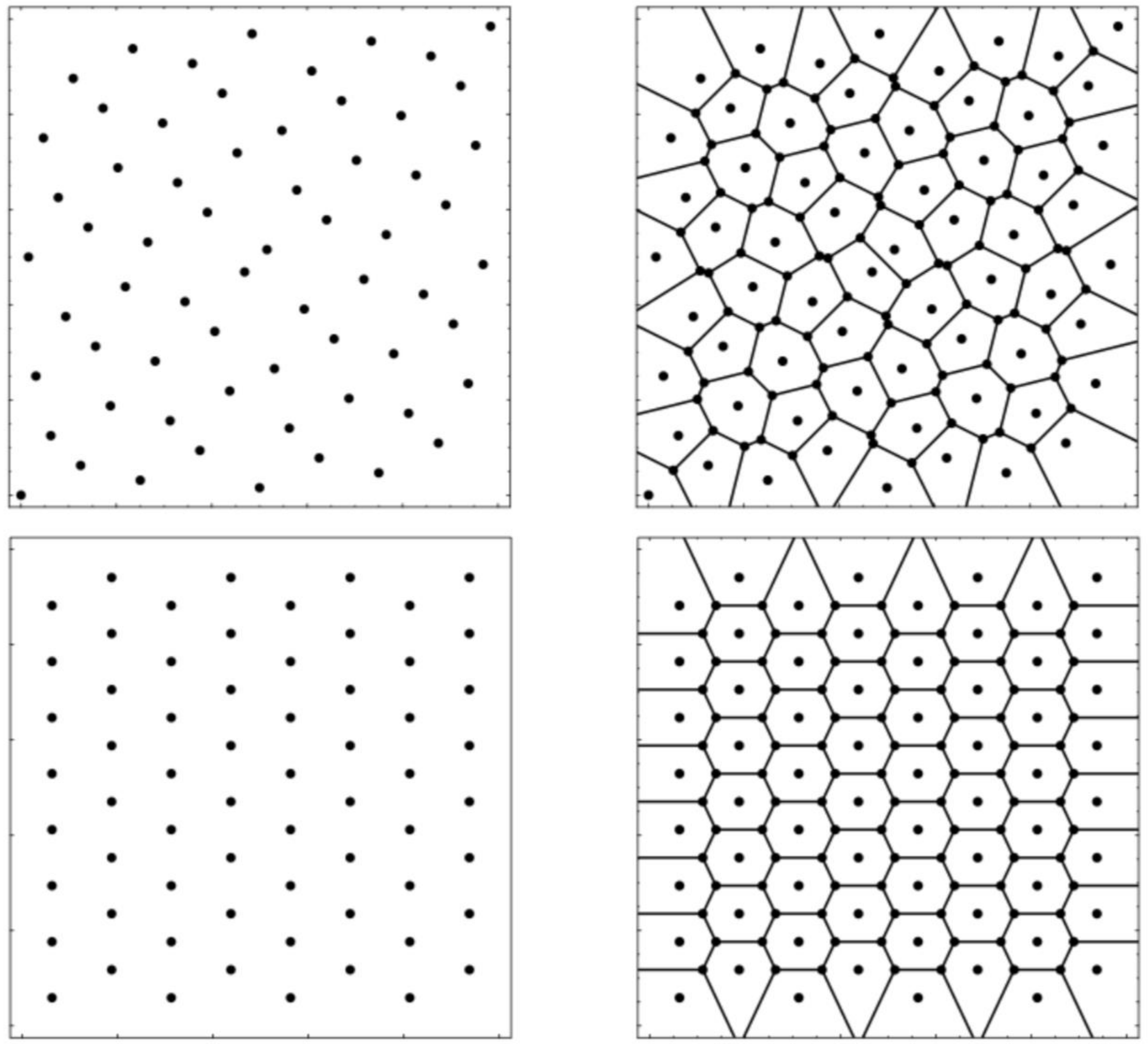


Figure 2.25: Stratification of I^2 with Voronoi diagrams. (a) 64-element Hammersley point set; (b) Voronoi diagram implied through (a); (c) 64-element hexagonal grid; (d) Voronoi diagram implied through (c).

Slide from Philipp Slusallek

Correlated Sampling:
Quasi-Monte Carlo Integration

Quasi-Monte Carlo Integration

- Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible

- The success of any Monte Carlo procedure stands or falls with the quality of these random samples

- If the distribution of the sample points is not uniform then there are large regions where there are no samples at all, which can increases the error

- Closely related to this is the fact that a smooth function is evaluated at unnecessary many locations if samples are clumped

Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions
- Based on number-theoretic approaches
- Samples with good uniform properties can be generated in very high dimensions.
- Sample generation is pretty fast: (almost) no pre-processing

Quasi-Monte Carlo Integration

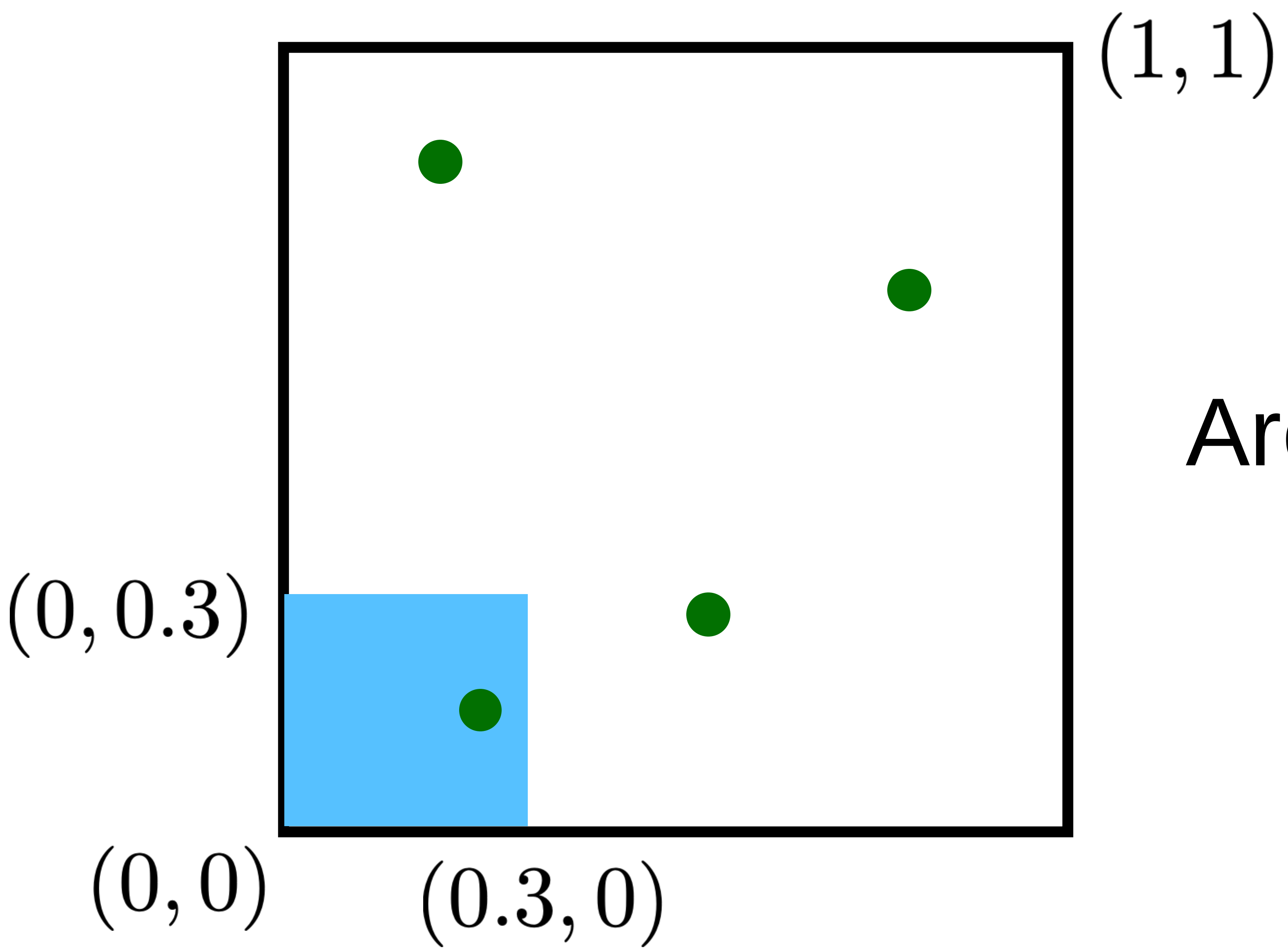
- Low discrepancy sequences
  - Halton and Hammerslay sequences
  - Scrambled sequences
- Discrepancy

Discrepancy: Basic idea

- The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution

Discrepancy: Basic idea

- The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution

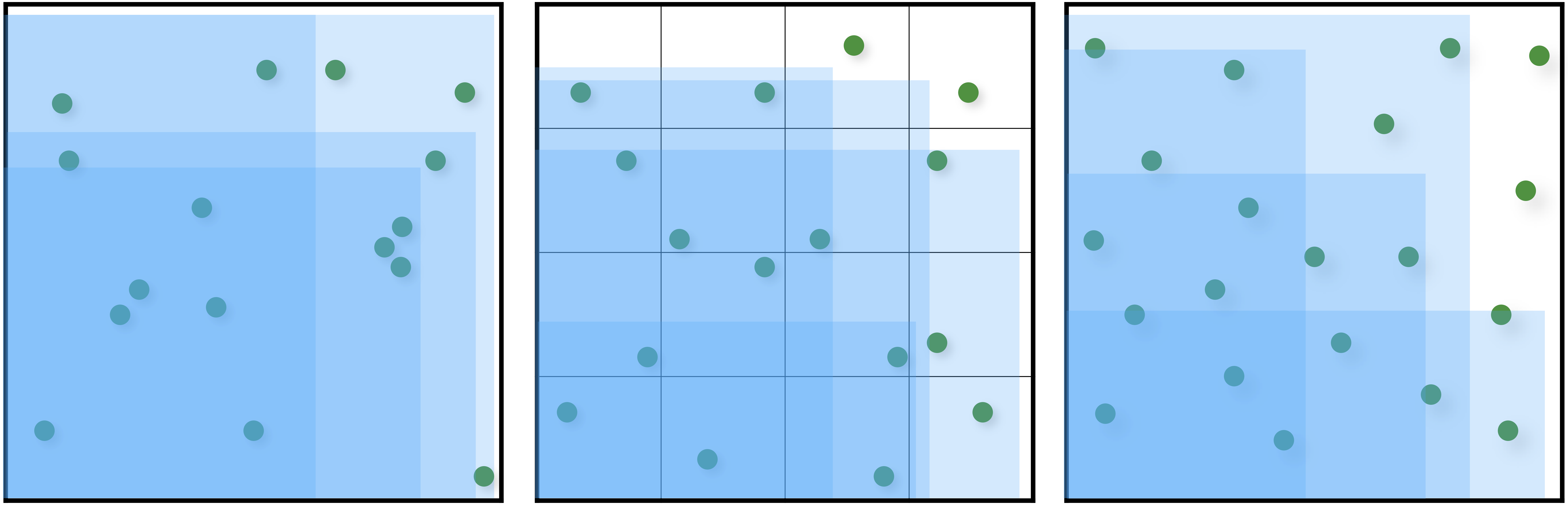


Area of the blue box: 0.09

Area associated to each sample: 0.25

Discrepancy: 0.25 - 0.09 = 0.16

Spatial Statistics: Discrepancy



Random Jitter Poisson Disk

Discrepancy = BoxArea - FractionSamples

Star Discrepancy

Discrepancy

DEFINITION 2.1 (Discrepancy) *Let* $\mathbf{P} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ *with* $\mathbf{x}_i \in \mathbf{I}^s, i = 1, \ldots, N$ *be a point set. The* discrepancy *of* $\mathbf{P}$*, denoted as* $D_N(\mathbf{P})$*, is a measure for the deviation of a point set from its ideal distribution. The discrepancy of* $\mathbf{P}$ *is defined as*

$$D_N(\mathbf{P}) \equiv D_N(\mathbf{P}, \mathcal{B}) \stackrel{\text{def}}{=} \sup_{\mathbf{B} \in \mathcal{B}} \left| \frac{\#(\mathbf{P} \cap \mathbf{B})}{N} - \mu^s(\mathbf{B}) \right|,$$

where $\mathcal{B}$ *corresponds to a Lebesgue measurable family of subsets of* $\mathbf{I}^s$*,* $\#$ *corresponds to the counting measure over* $\mathcal{B}$ *with respect to* $\mathbf{P}$*,* μ^s *is, as usual, the Lebesgue measure and* $\mathbf{B}$ *refers to a non empty subset of* $\mathcal{B}$*.*

Slide from Philipp Slusallek

Radical Inverse

Techniques based on a construction called as **radical inverse**

Any integer can be represented in the form:

$$n = \sum_{i=1}^{\infty} d_i b^{i-1}$$

n	Binary	$\Phi_b(n)$
1	1	
2	01	
3	11	
4	001	
5	101	

Radical Inverse

Techniques based on a construction called as **radical inverse**

Any integer can be represented in the form:

$$n = \sum_{i=1}^{\infty} d_i b^{i-1}$$

Radical inverse:

$$\Phi_b(n) = 0.d_1 d_2 ... d_m$$

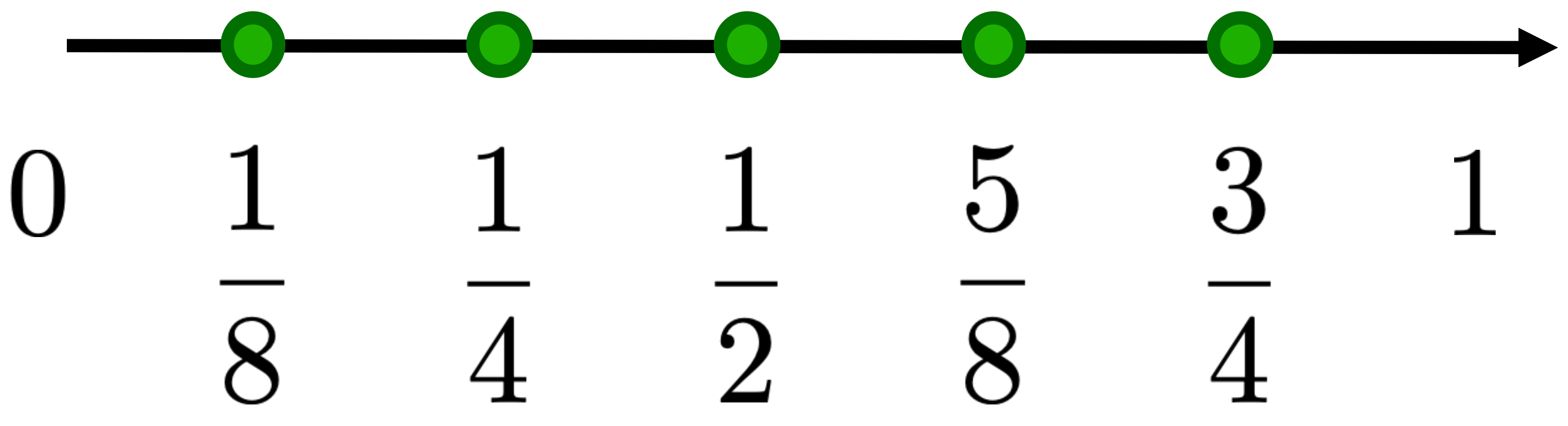
n	Binary	$\Phi_b(n)$
1	1	0.1
2	01	0.01
3	11	0.11
4	001	0.001
5	101	0.101

Radical Inverse

Techniques based on a construction called as **radical inverse**

Radical inverse:

$$\Phi_b(n) = 0.d_1 d_2 ... d_m$$



n	Binary	$\Phi_b(n)$
1	1	0.1 = 1/2
2	01	0.01 = 1/4
3	11	0.11 = 3/4
4	001	0.001 = 1/8
5	101	0.101 = 5/8

Halton and Hammerslay Sequence

Techniques based on a construction called as **radical inverse**

Radical inverse: $\Phi_b(n) = 0.d_1 d_2 ... d_m$

Halton Sequence: For n-dimensional sequence, we use different base b for each dimension

$$x_i = (\Phi_2(i), \Phi_3(i), \Phi_5(i), \ldots, \Phi_{p_n}(i))$$

Halton and Hammerslay Sequence

Techniques based on a construction called as **radical inverse**

Radical inverse: $\Phi_b(n) = 0.d_1 d_2 ... d_m$

Halton Sequence: For n-dimensional sequence, we use different base b for each dimension

$$x_i = (\Phi_2(i), \Phi_3(i), \Phi_5(i), \ldots, \Phi_{p_n}(i))$$

Hammerslay Sequence: All except the first dimension has co-prime bases

$$x_i = \left(\frac{i}{N}, \Phi_{b_1}(i), \Phi_{b_2}(i), \ldots, \Phi_{b_n}(i)\right)$$

Halton and Hammerslay Sequence

Techniques based on a construction called as **radical inverse**

Radical inverse: $\Phi_b(n) = 0.d_1 d_2 ... d_m$

Halton Sequence:

$$x_i = (\Phi_2(i), \Phi_3(i), \Phi_5(i), \ldots, \Phi_{p_n}(i))$$

Hammerslay Sequence:

$$x_i = \left(\frac{i}{N}, \Phi_{b_1}(i), \Phi_{b_2}(i), \ldots, \Phi_{b_n}(i)\right)$$

Hammerslay has slightly **lower** discrepancy than Halton

Halton 4spp


Halton 8spp


Sobol 4spp

Sobol 8spp


Random 8spp

Visualizing samples

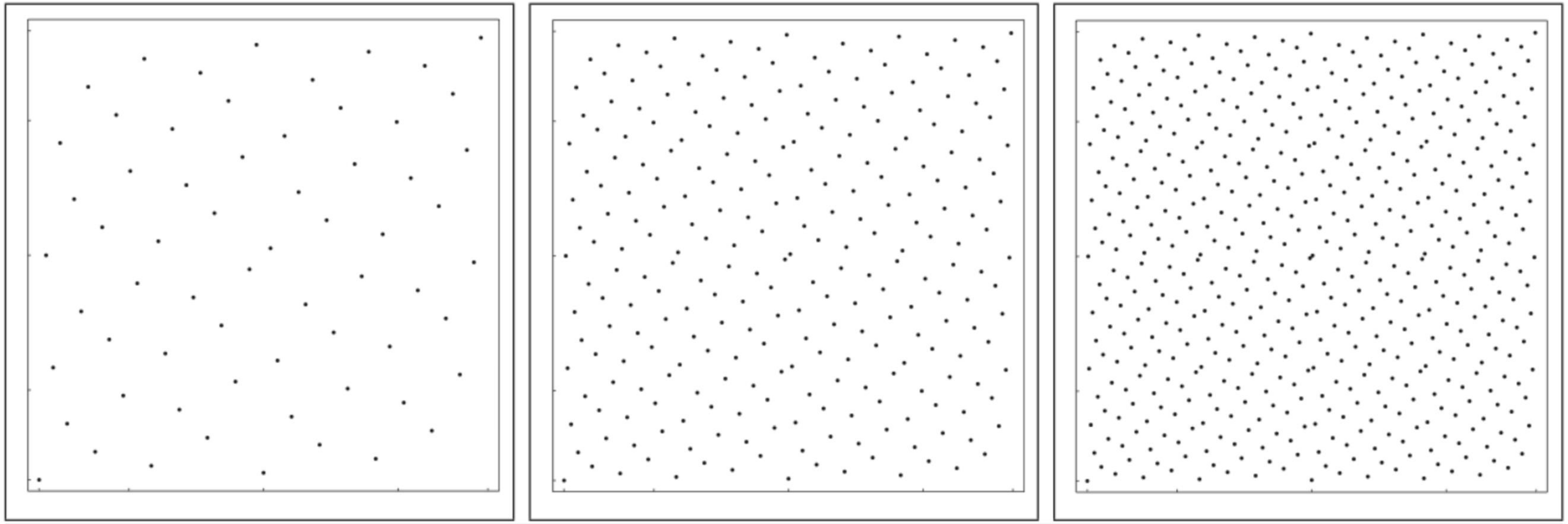


Figure 2.7: Hammersley Point Set on the 2D Plane. Three 2-dimensional Hammersley point sets $\mathbf{P}^2_{HAM} = \left(\frac{i}{N}, \Phi_2(i)\right)_{i \in (0,\ldots,N-1)}$ of sizes N = 64-element, N = 256-element and N = 512-element.

Slide from Philipp Slusallek

Visualizing samples

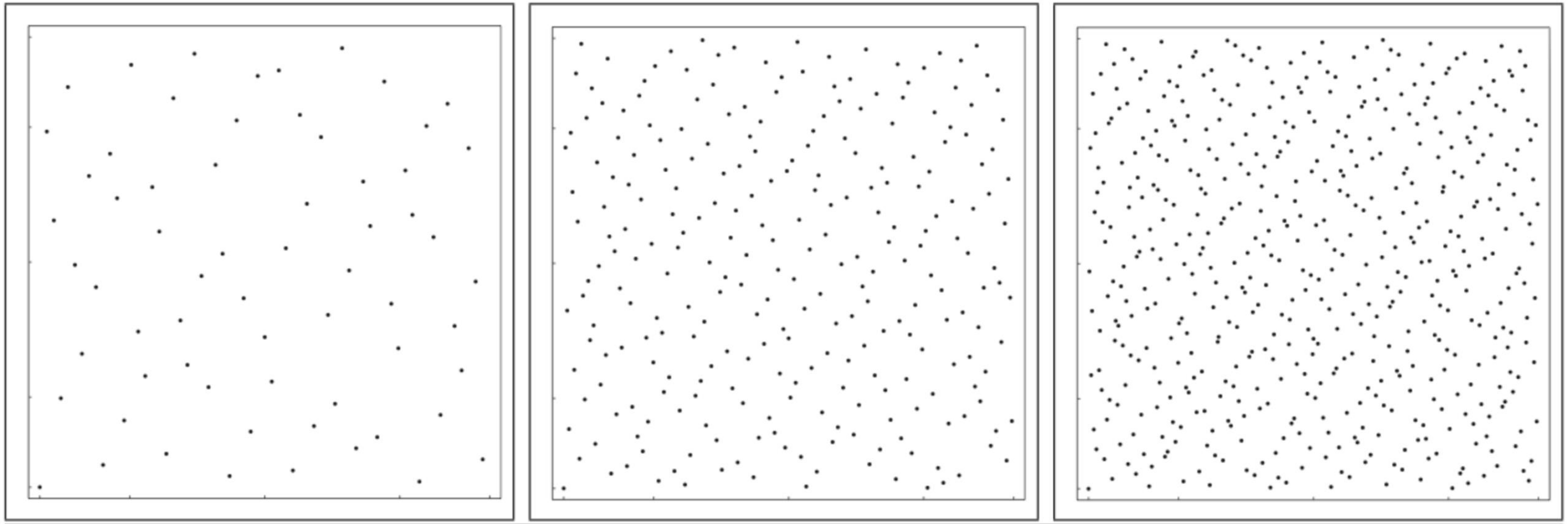
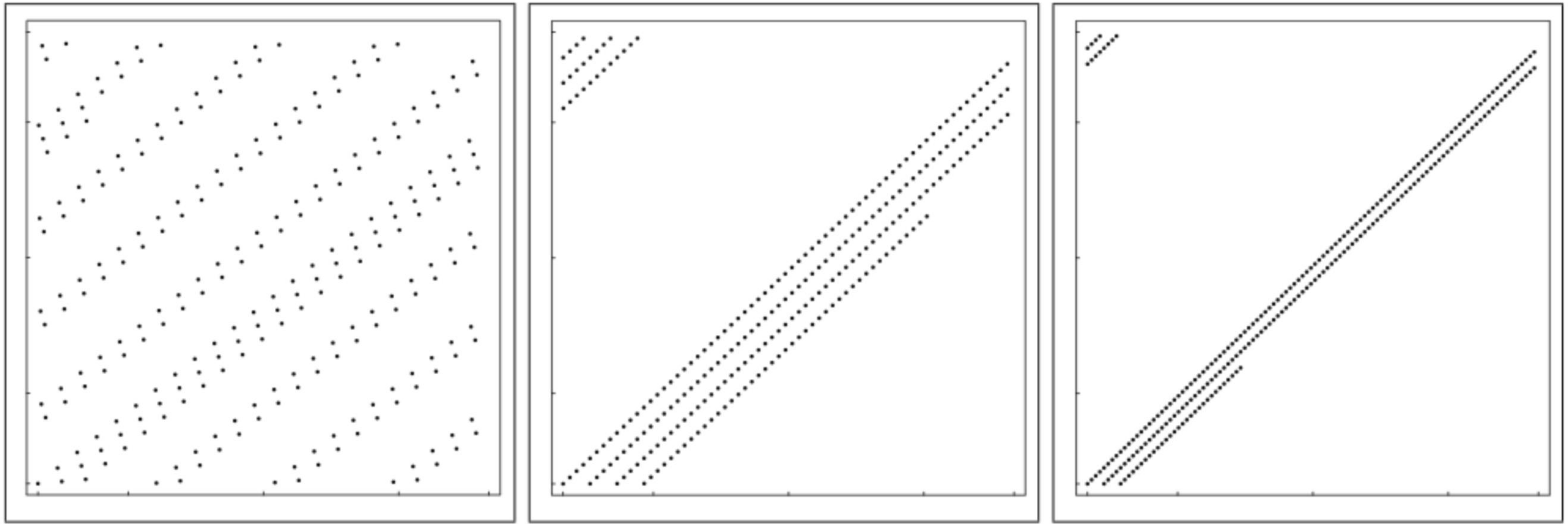


Figure 2.5: Halton sequence. The first 64, 256, and 512 points of the 2-dimensional Halton Sequence $\mathbf{P}^2_{\mathrm{HAL}} = (\Phi_2(i), \Phi_3(i))_{i \in \mathbb{N}_0}$.

Slide from Philipp Slusallek

Visualizing samples

Projection: (9,10) Projection: (19,20) Projection: (29,30)



Halton Sequence

Slide from Philipp Slusallek

Faure's permutation

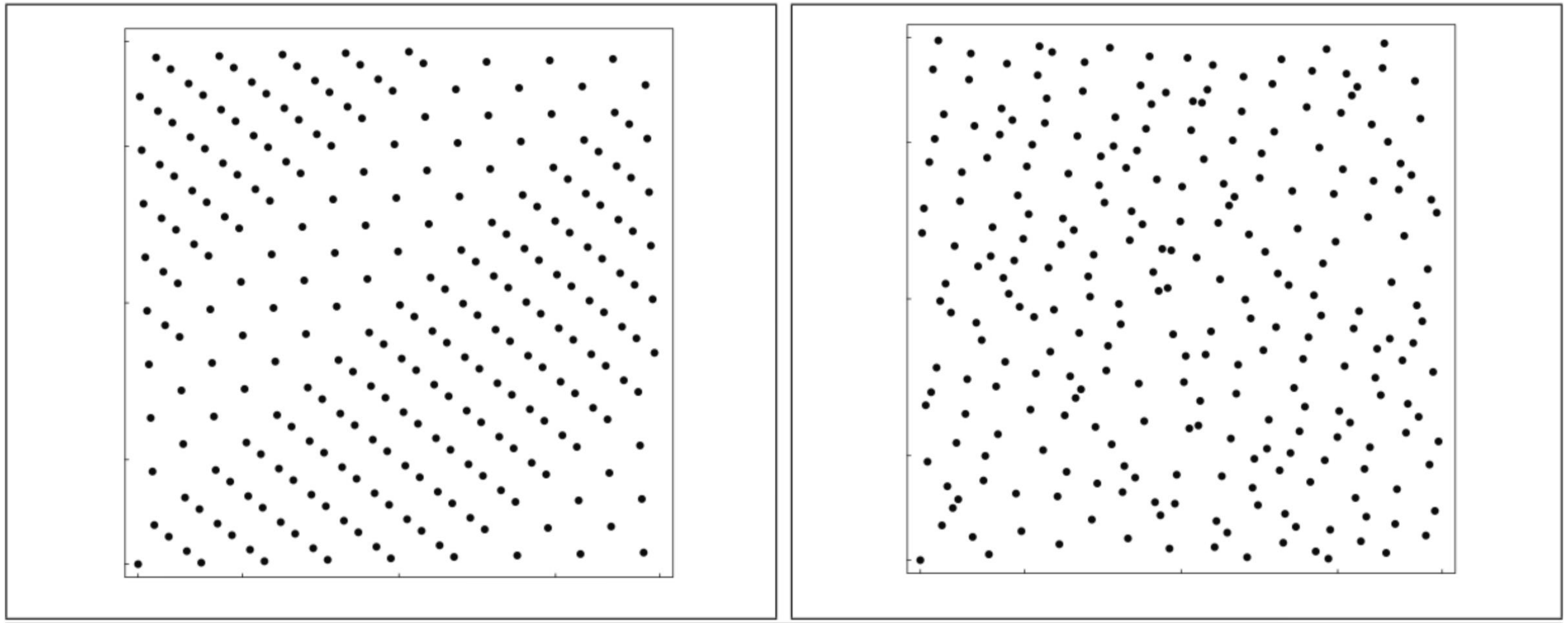


Figure 2.12: Halton Sequence and Scrambled Halton Sequence, Dimensions 7 and 8. **(a)** The first 256 elements of the 2-dimensional Halton sequence $\mathbf{P}^2_{\mathrm{HAL}} = \left(\Phi_7(i), \Phi_8(i)\right)$ and the scrambled versions of dimension 7 and 8 generated according to procedure of Faure.

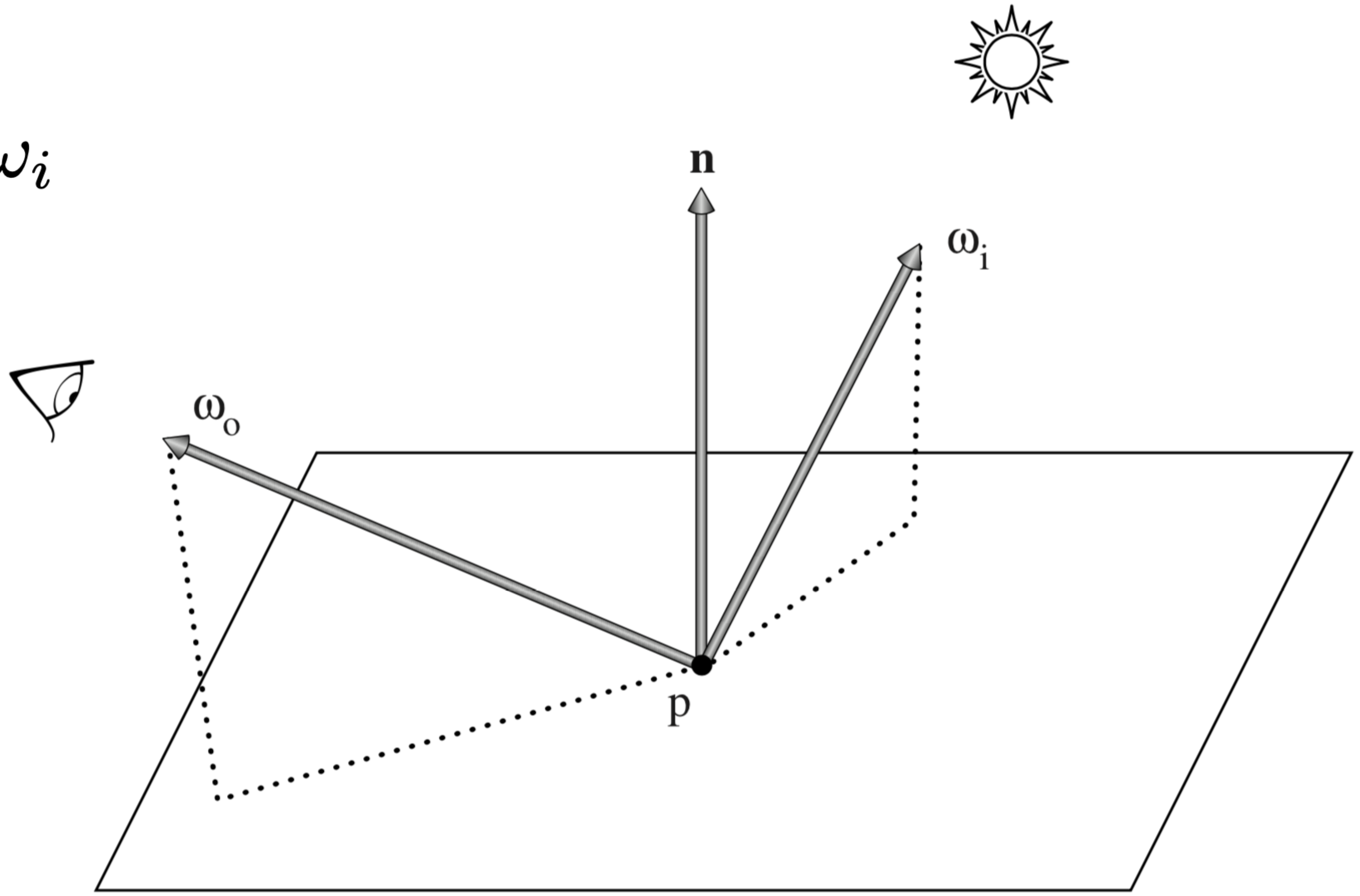
Slide from Philipp Slusallek


Questions?
Gaussian Material Synthesis by Zsolnai-Feher, Wonka, Wimmer [SIGGRAPH 2018]

Importance Sampling

Importance Sampling

$$L_o(p,\omega) = \int_{\mathcal{H}^2} f_r(x,\omega_0,\omega_i) L_i(x,\omega_i) |\cos\theta_i| d\omega_i$$

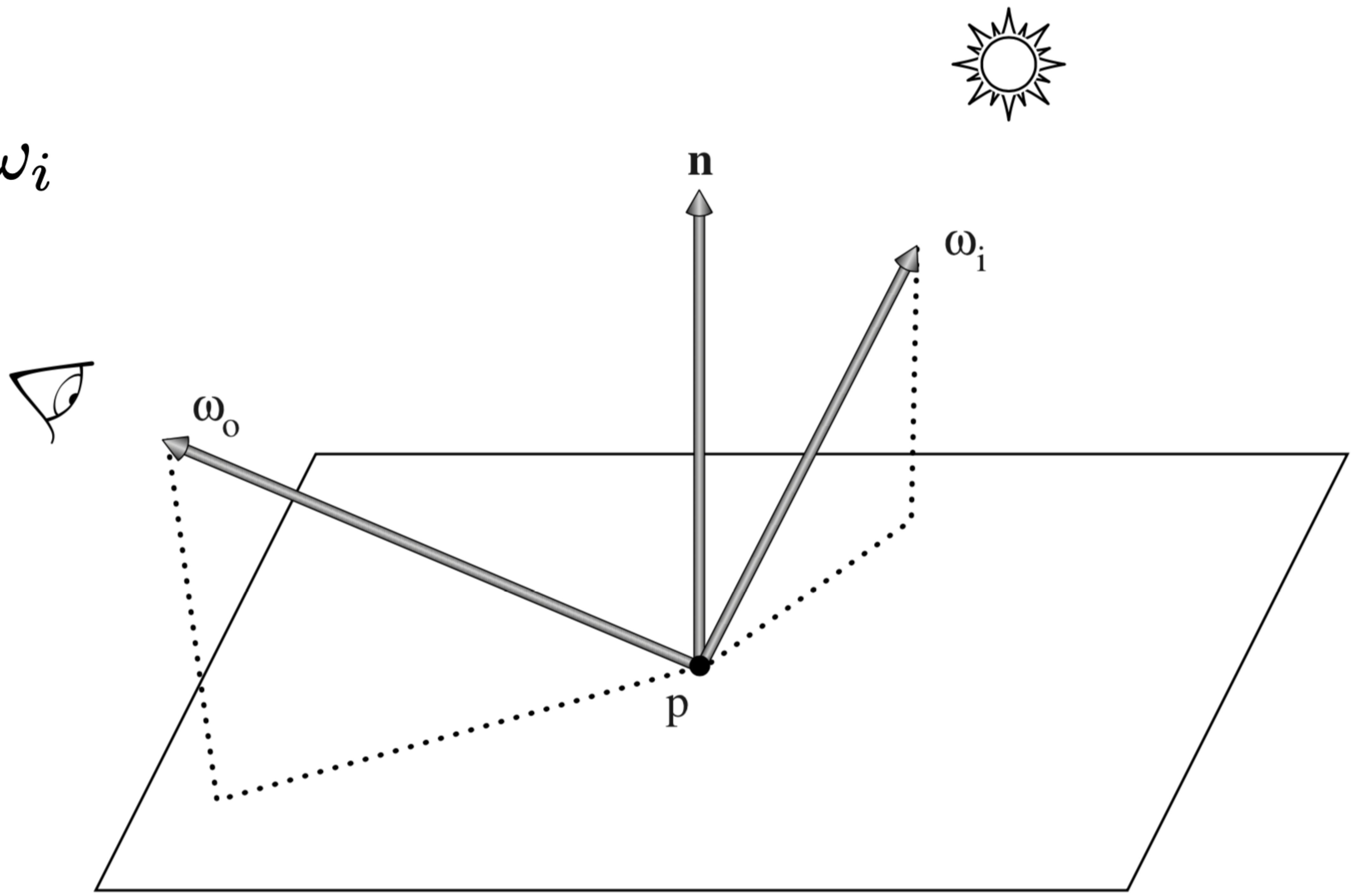


What terms can we importance sample?

- BSDF
- Incident radiance
- cosine term

Importance Sampling: Cosine term

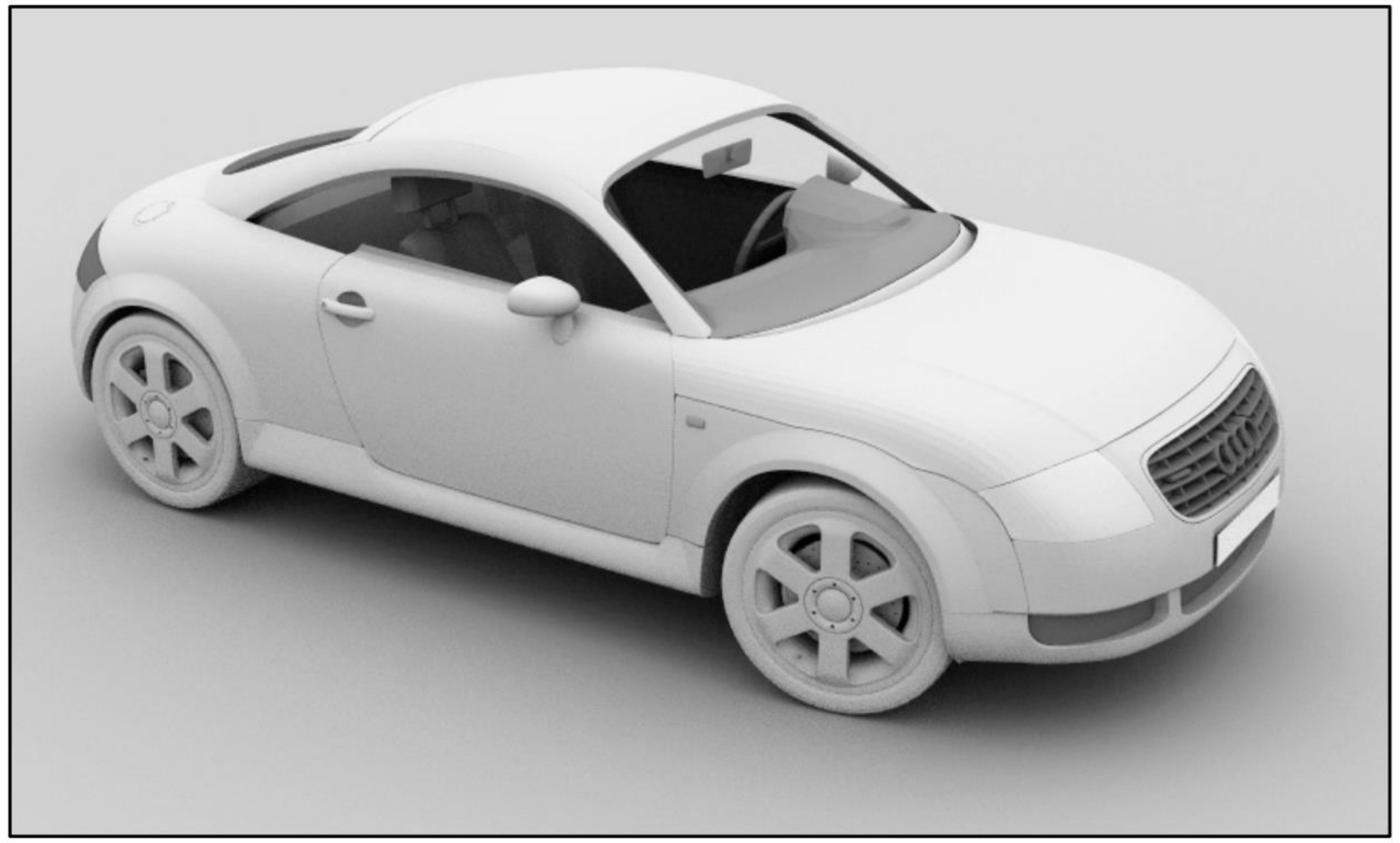
$$L_o(p,\omega) = \int_{\mathcal{H}^2} f_r(x,\omega_0,\omega_i) L_i(x,\omega_i) |\cos\theta_i| d\omega_i$$

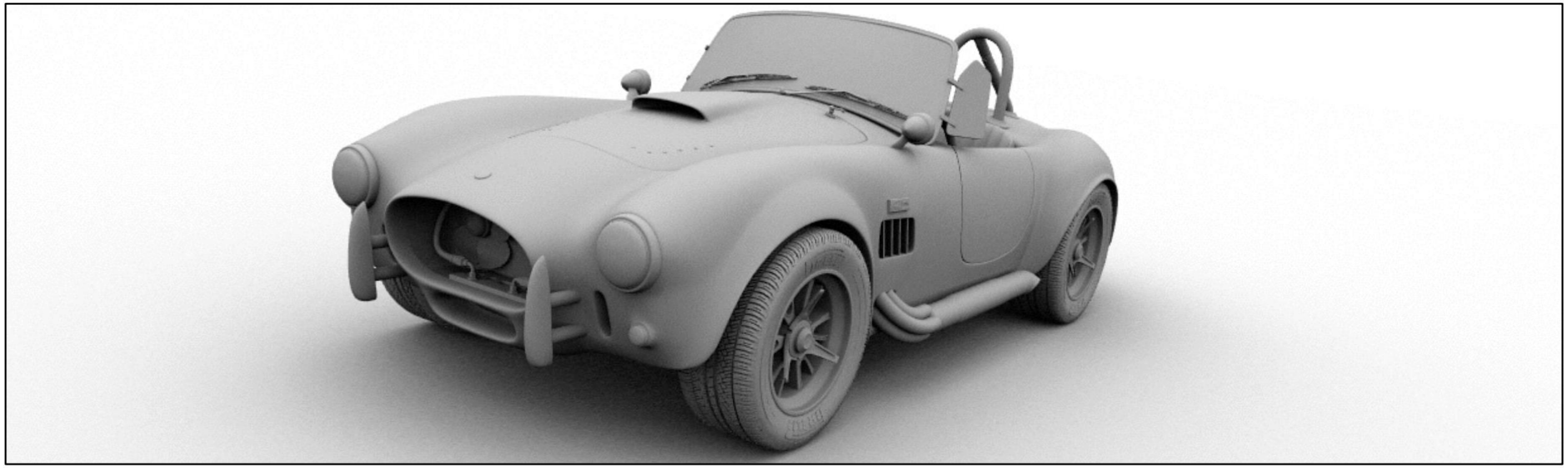


What terms can we importance sample?

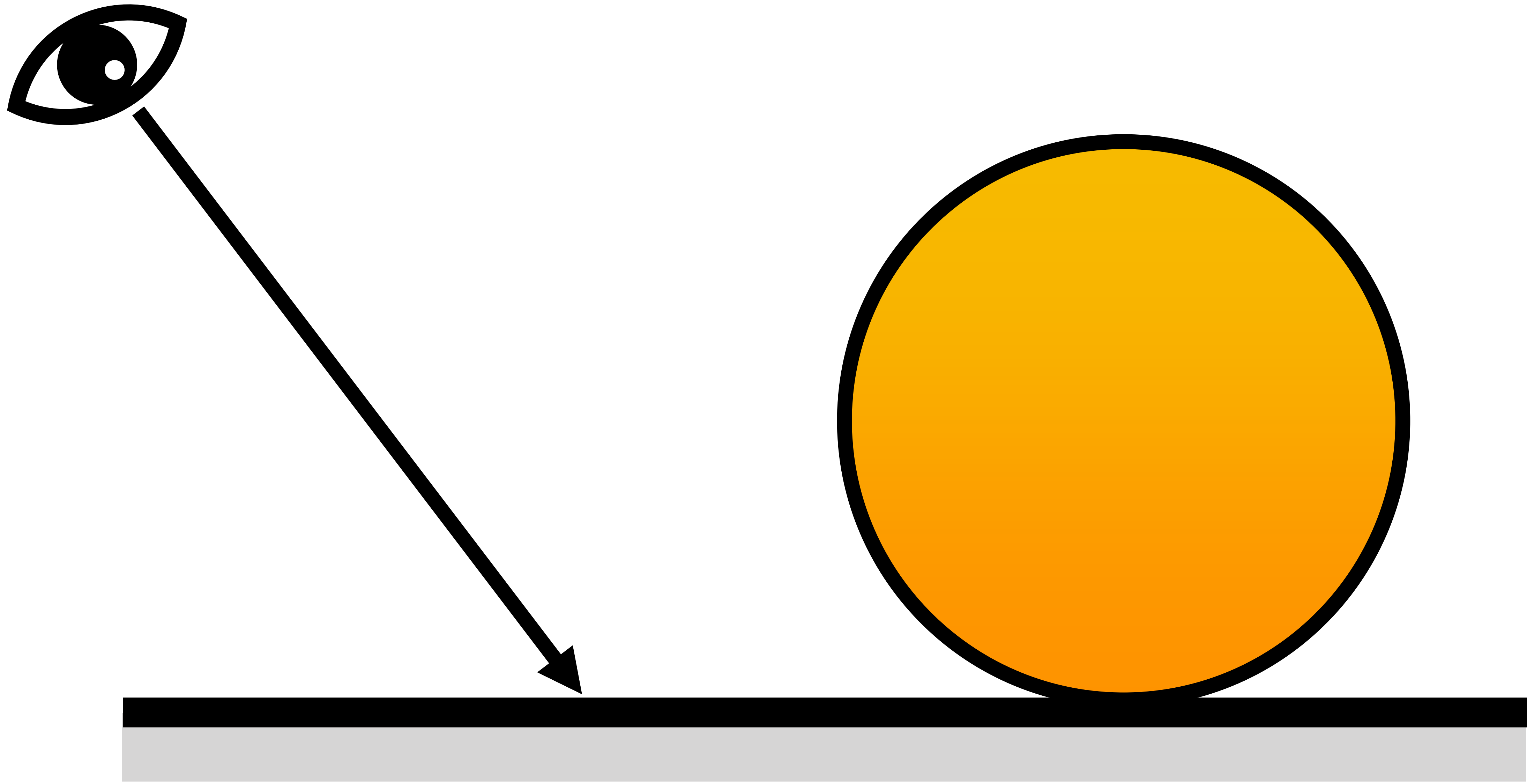
- BSDF
- Incident radiance
- cosine term

Example: Ambient Occlusion





Example: Ambient Occlusion

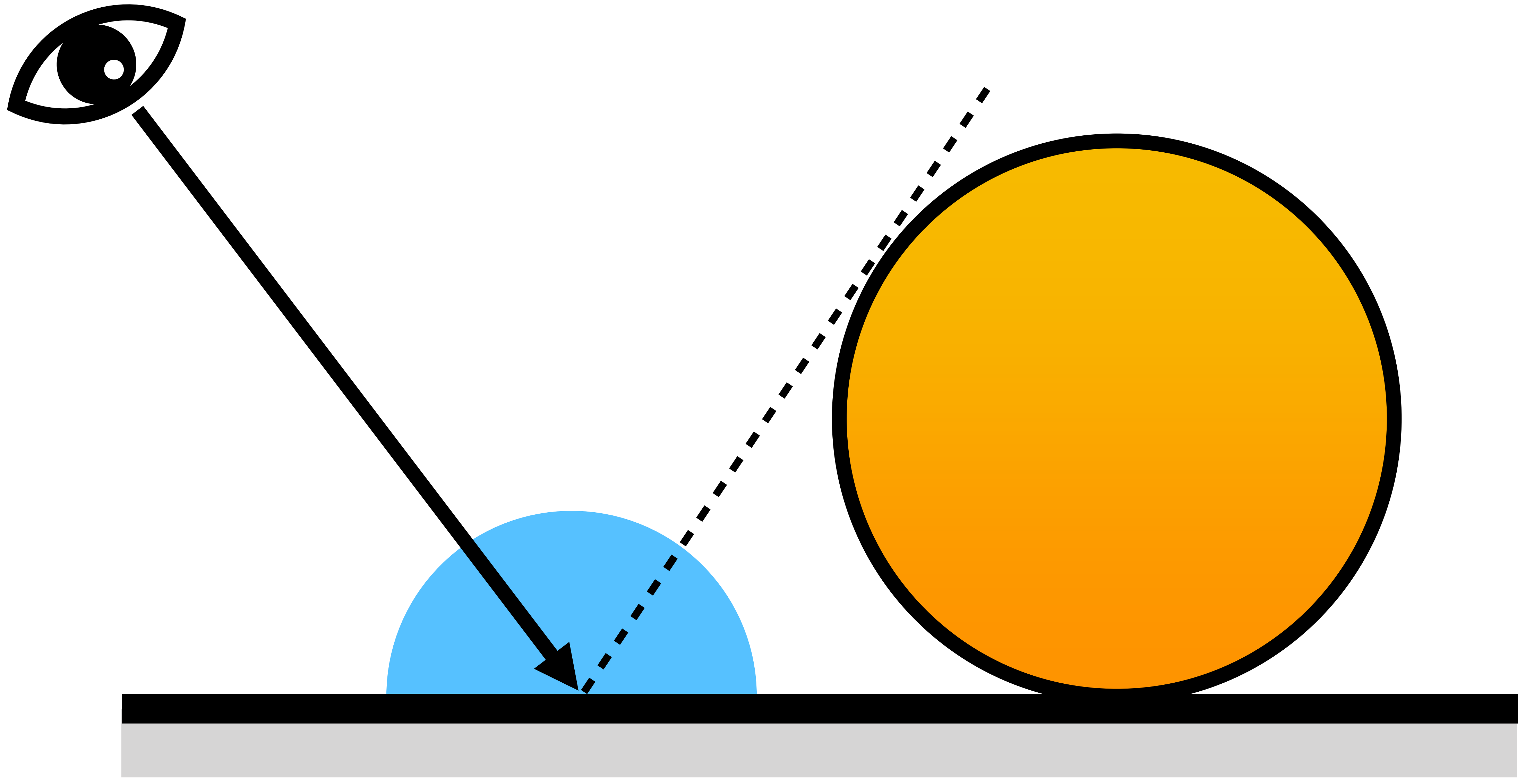


$$L_o(p,\omega) = \int_{\mathcal{H}^2} f_r(x,\omega_0,\omega_i) L_i(x,\omega_i) |\cos\theta_i| d\omega_i$$

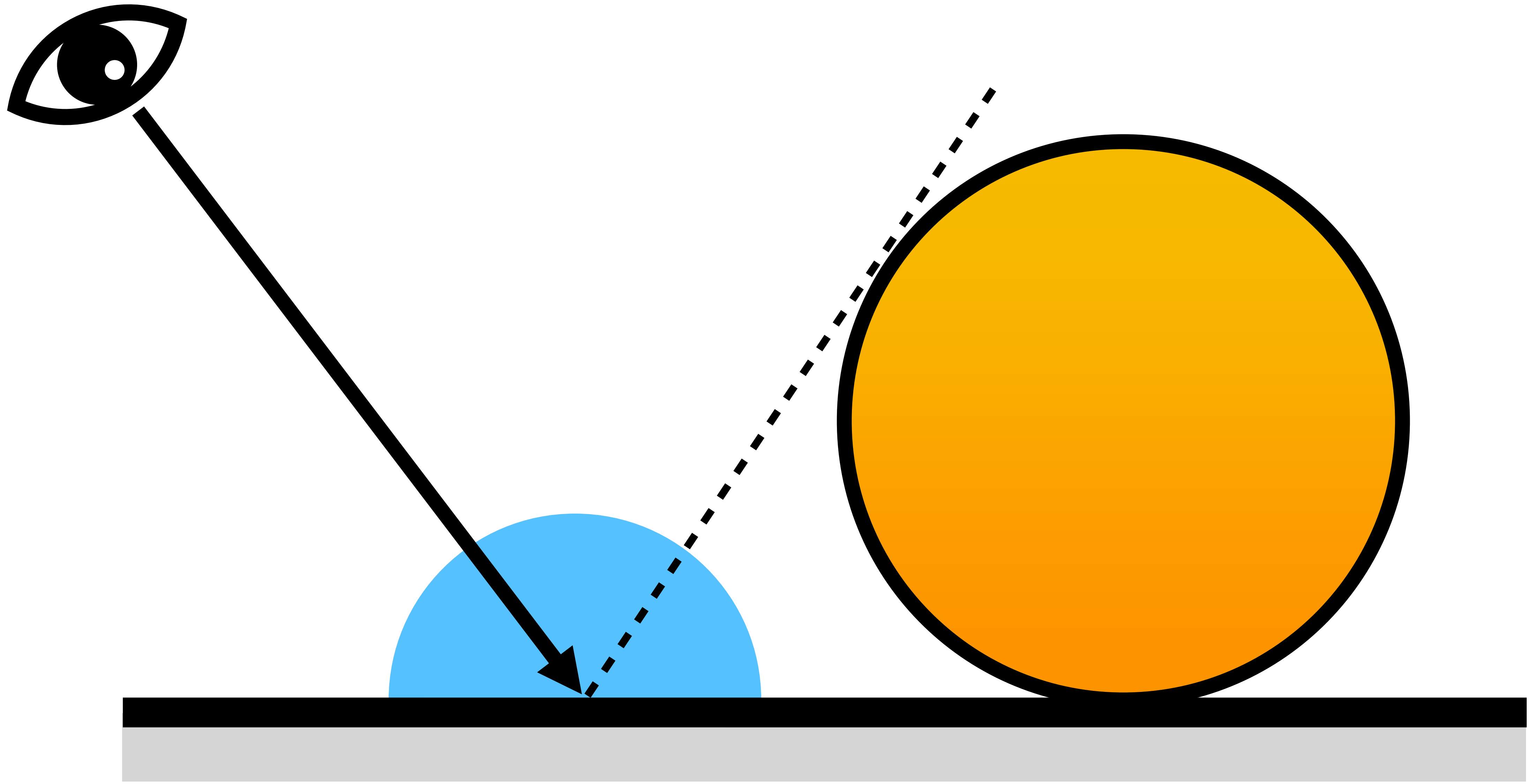
$$L_o(p,\omega) = \frac{\rho}{\pi} \int_{\mathcal{H}^2} V(x,\omega_i) |\cos\theta_i| d\omega_i$$

Example: Ambient Occlusion

$$L_o(p,\omega) = \frac{\rho}{\pi}\int_{\mathcal{H}^2} V(x,\omega_i)|\cos\theta_i|d\omega_i$$



Example: Ambient Occlusion

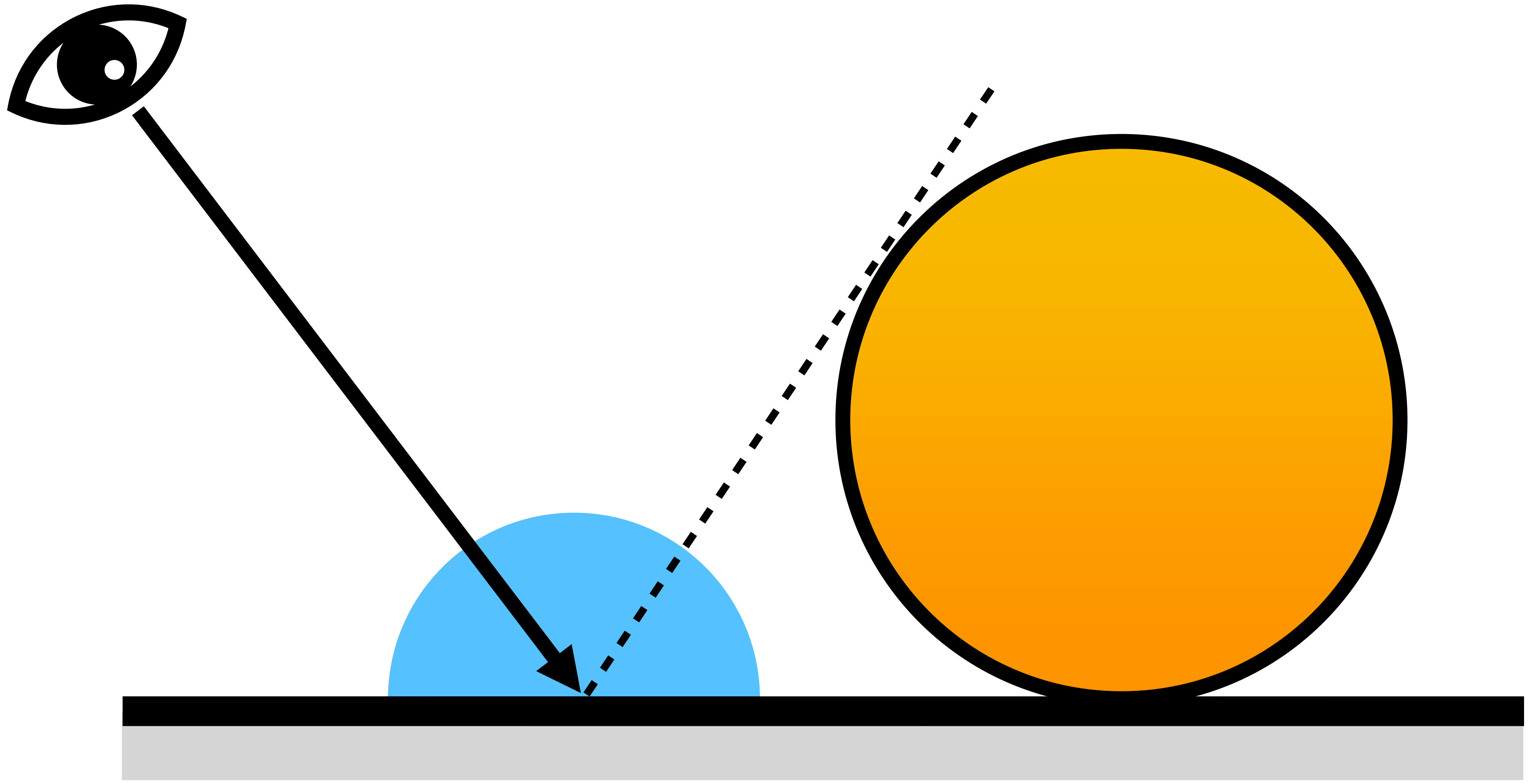


$$L_o(p,\omega) = \frac{\rho}{\pi}\int_{\mathcal{H}^2} V(x,\omega_i)|\cos\theta_i|d\omega_i$$

$$L_o(p,\omega) = \frac{\rho}{\pi}\frac{1}{N}\sum_{k=1}^{N}\frac{V(x,\omega_{i,k})|\cos\theta_{i,k}|}{p(x,\omega_{i,k})}$$

Example: Ambient Occlusion

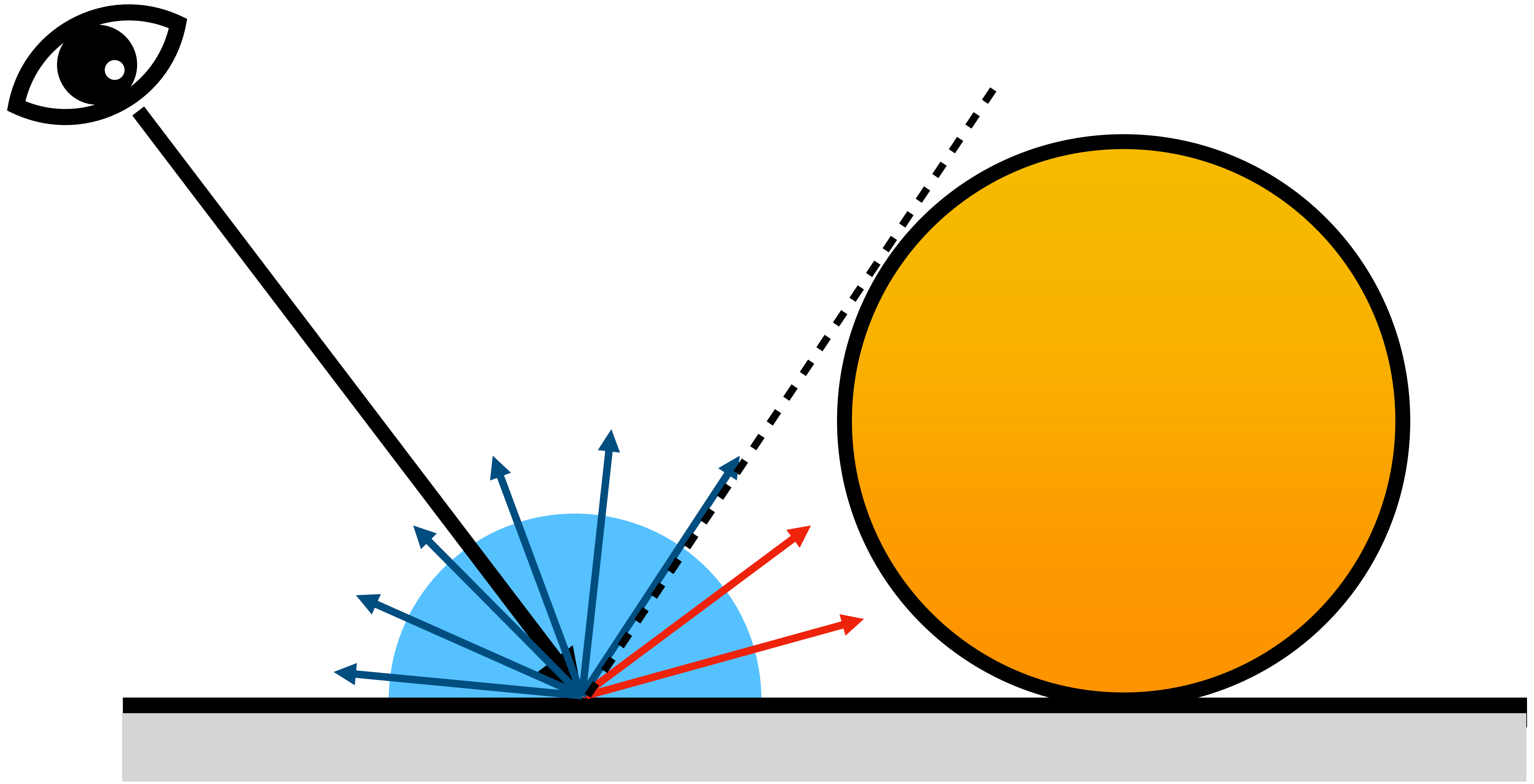
$$L_o(p,\omega) = \frac{\rho}{\pi}\frac{1}{N}\sum_{k=1}^{N}\frac{V(x,\omega_{i,k})|\cos\theta_{i,k}|}{p(x,\omega_{i,k})}$$



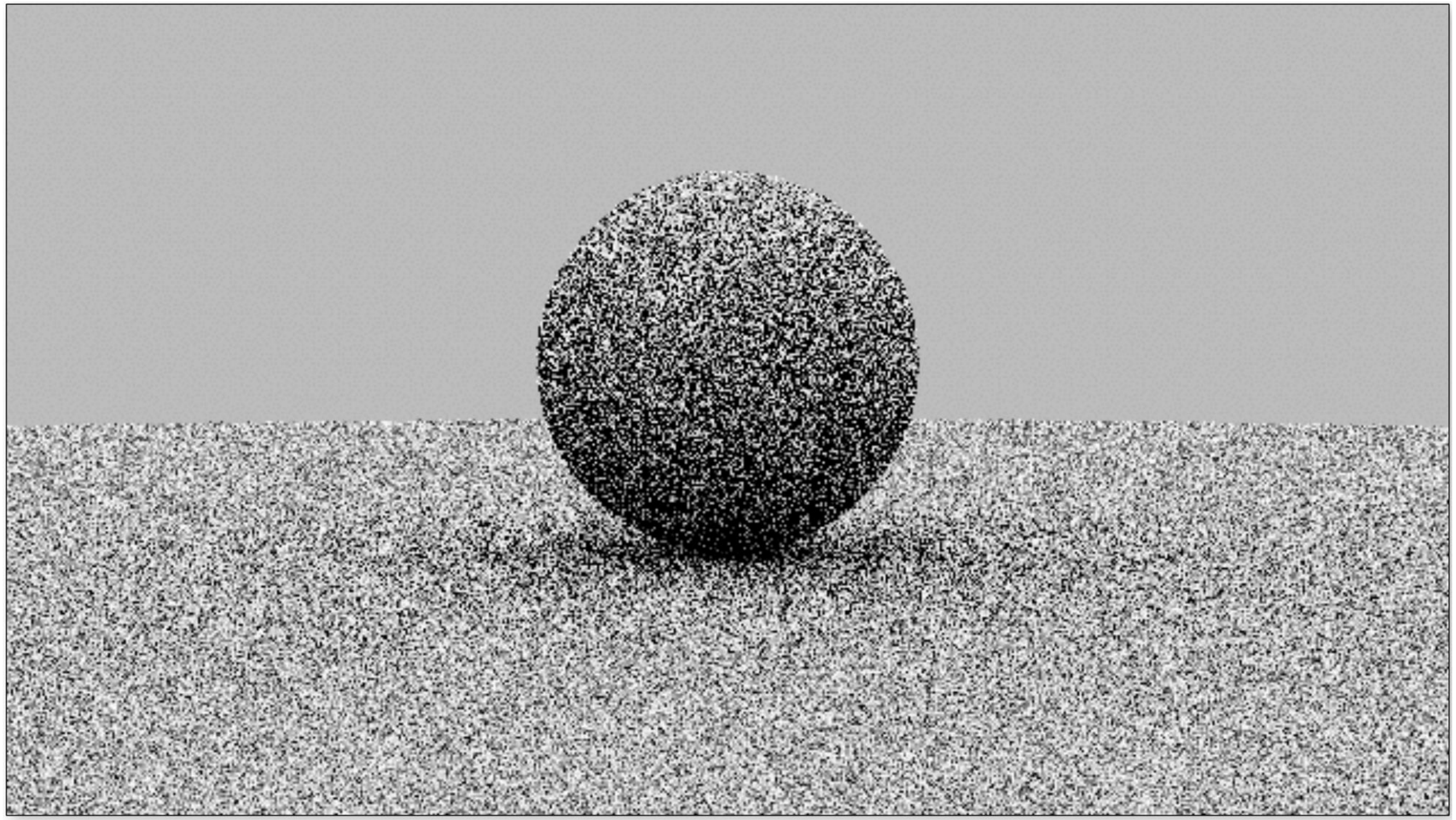
$$p(x,\omega_{i,k}) \propto \;\; ???$$

Hemispherical Sampling: Constant PDF

Uniform Hemispherical Sampling

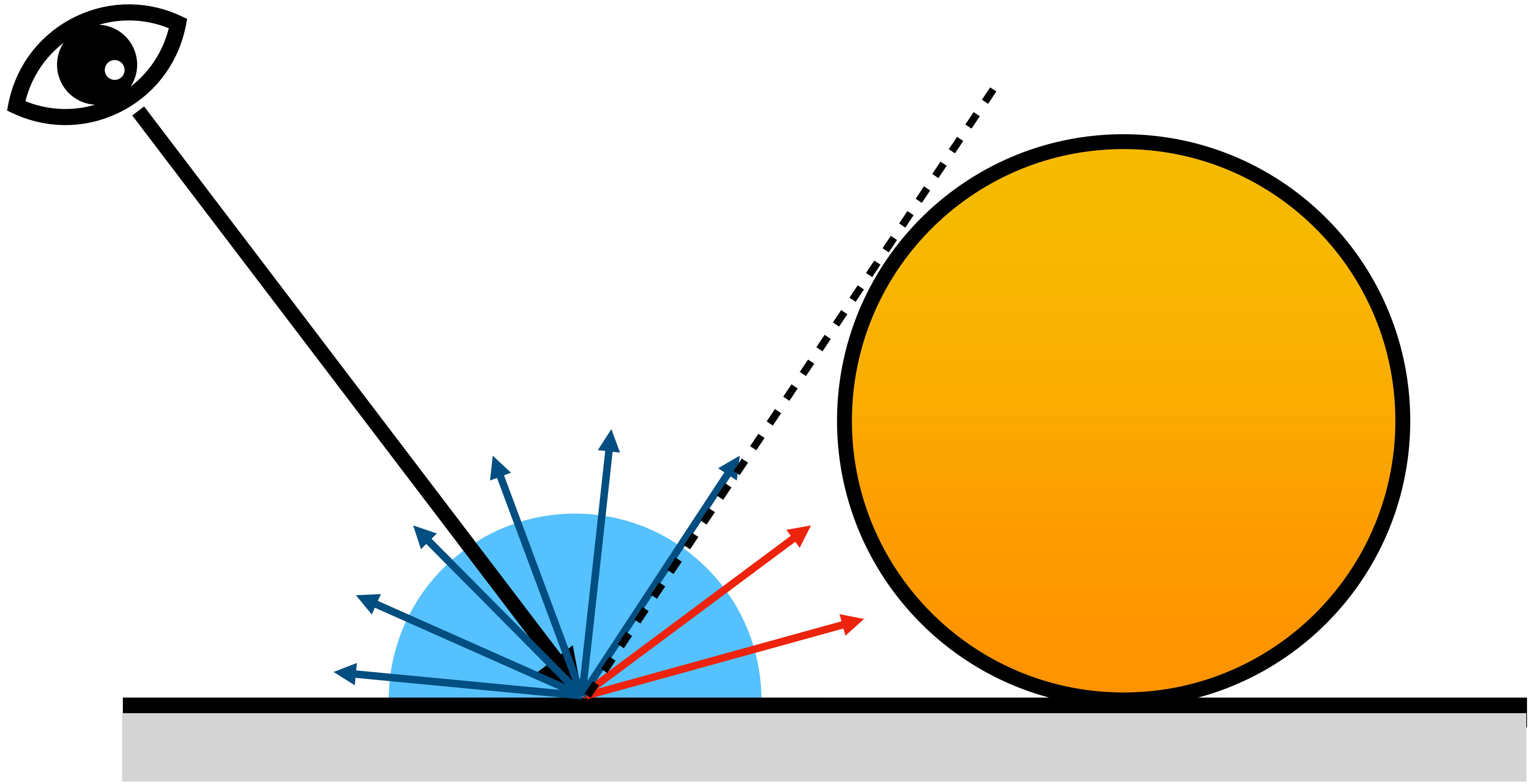


(1 Sample)

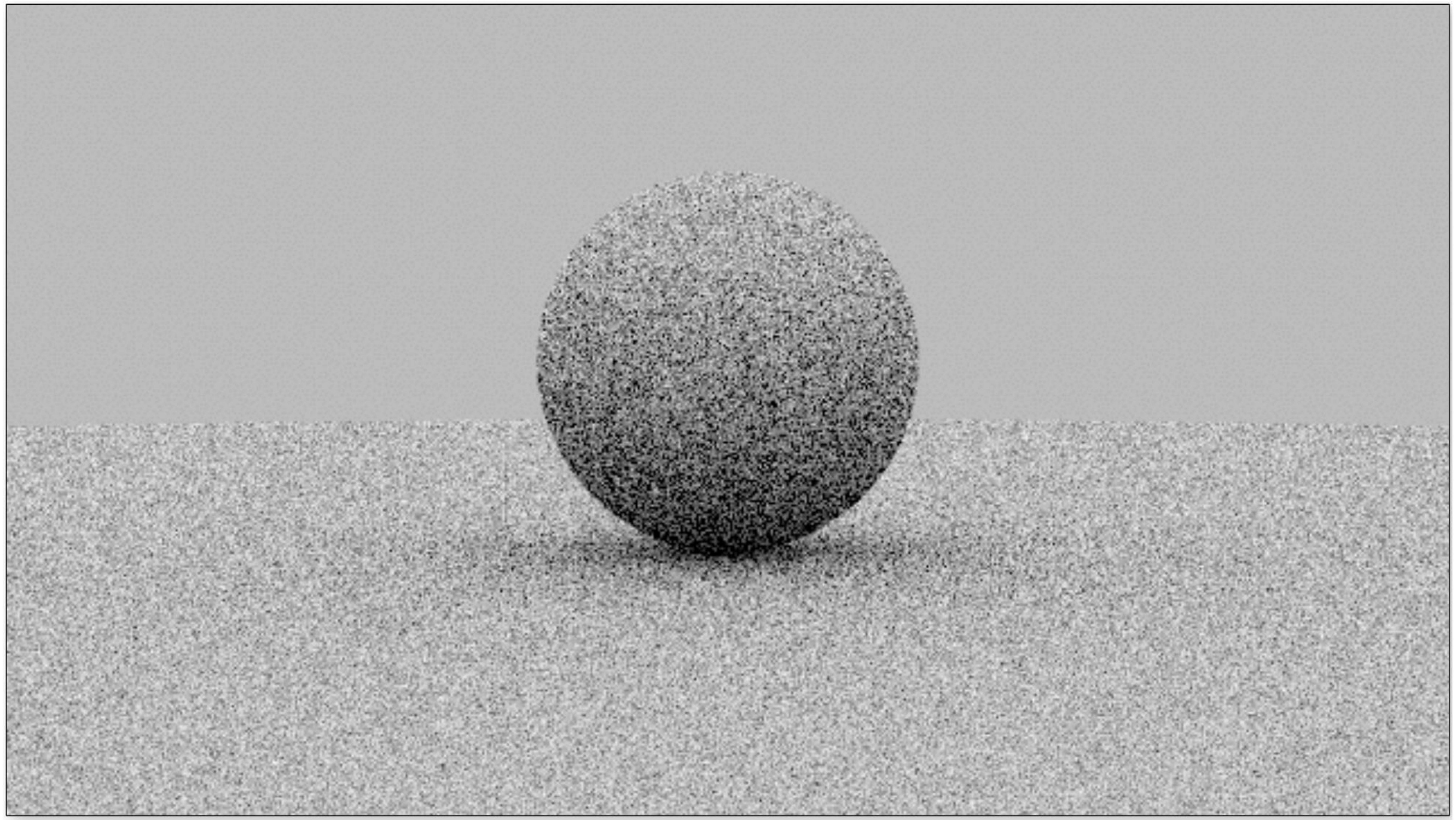


Hemispherical Sampling: Constant PDF

Uniform Hemispherical Sampling



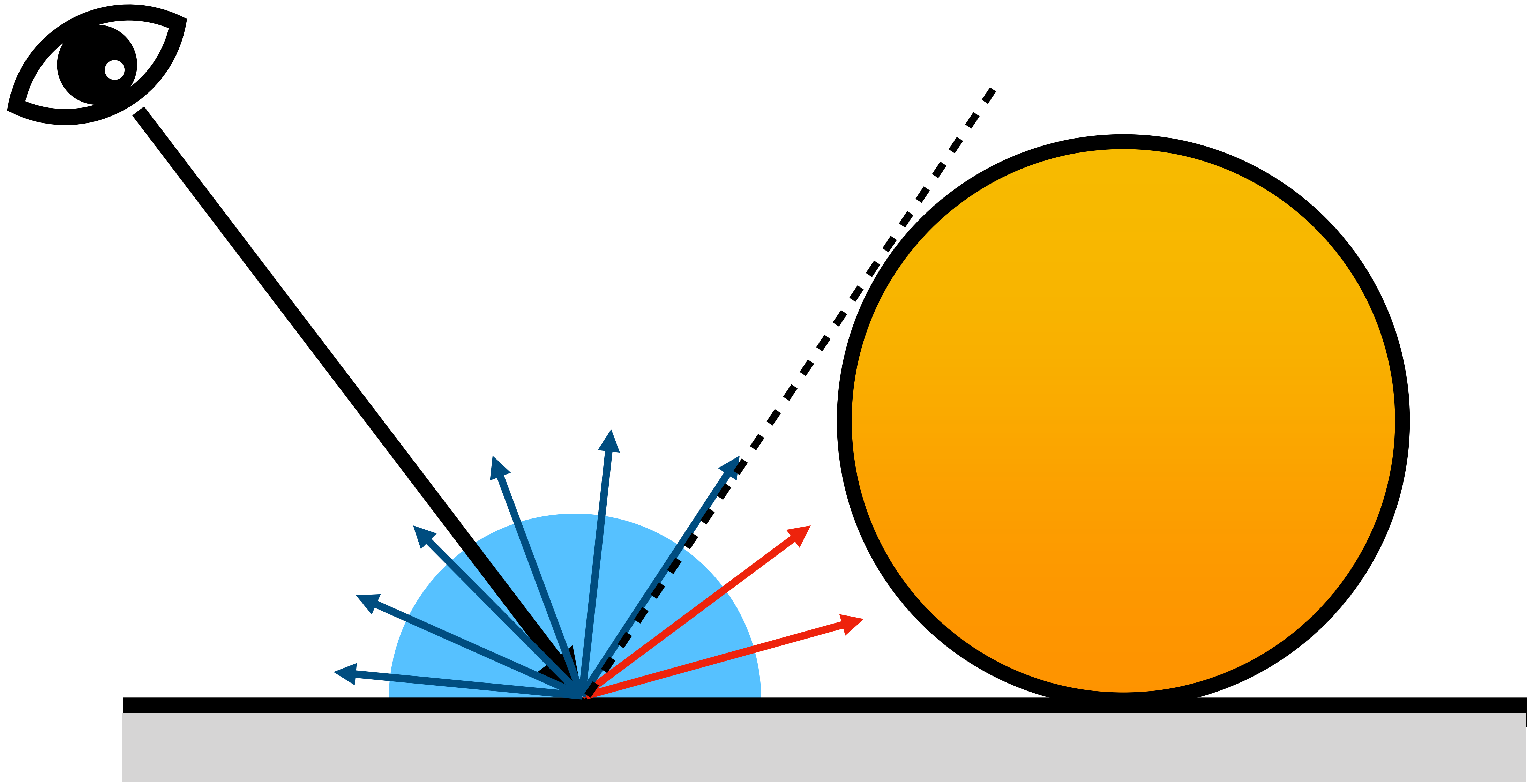
(4 Samples)



$$p(x, \omega_i) = \frac{1}{2\pi}$$

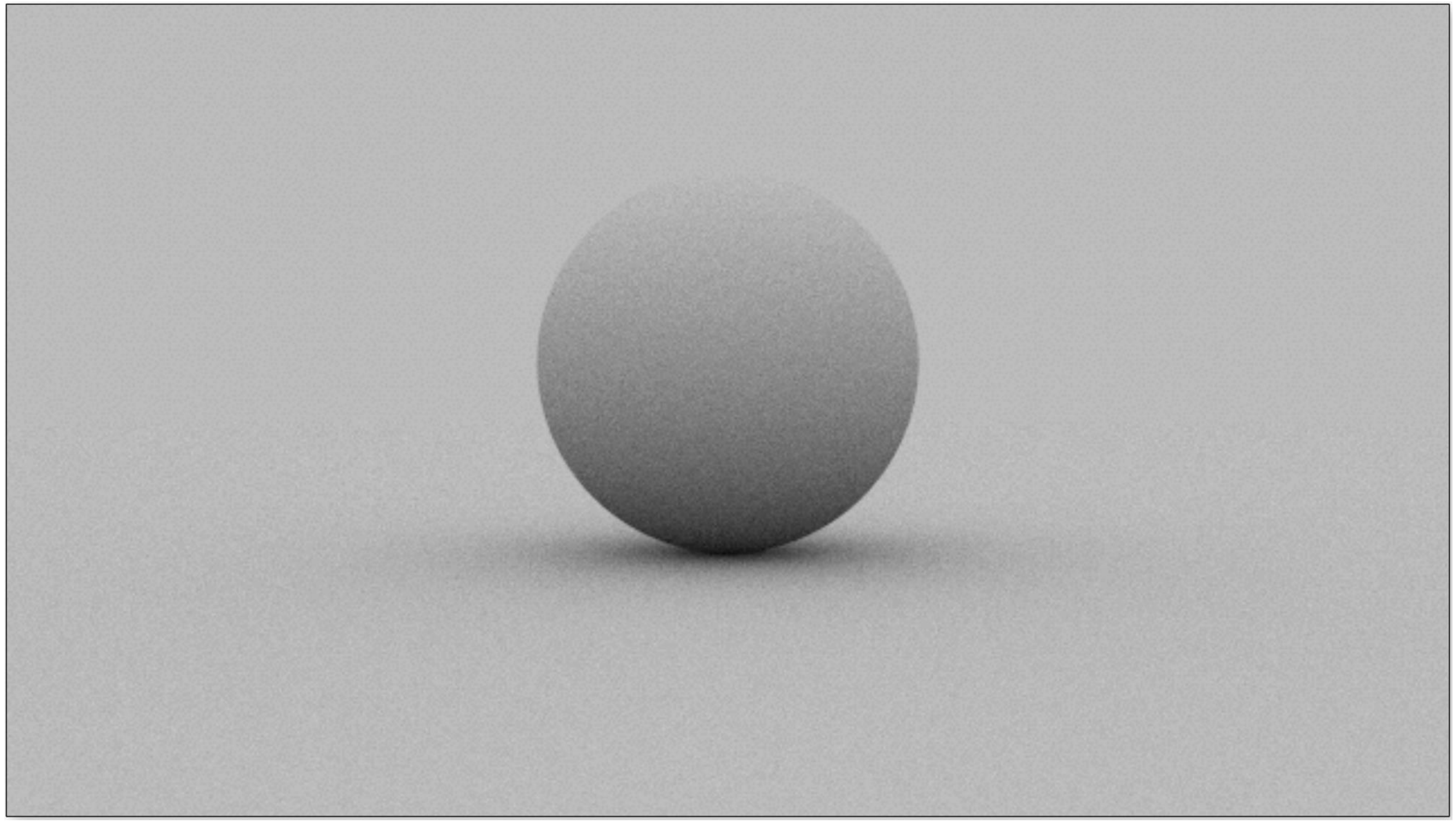
Hemispherical Sampling: Constant PDF

Uniform Hemispherical Sampling



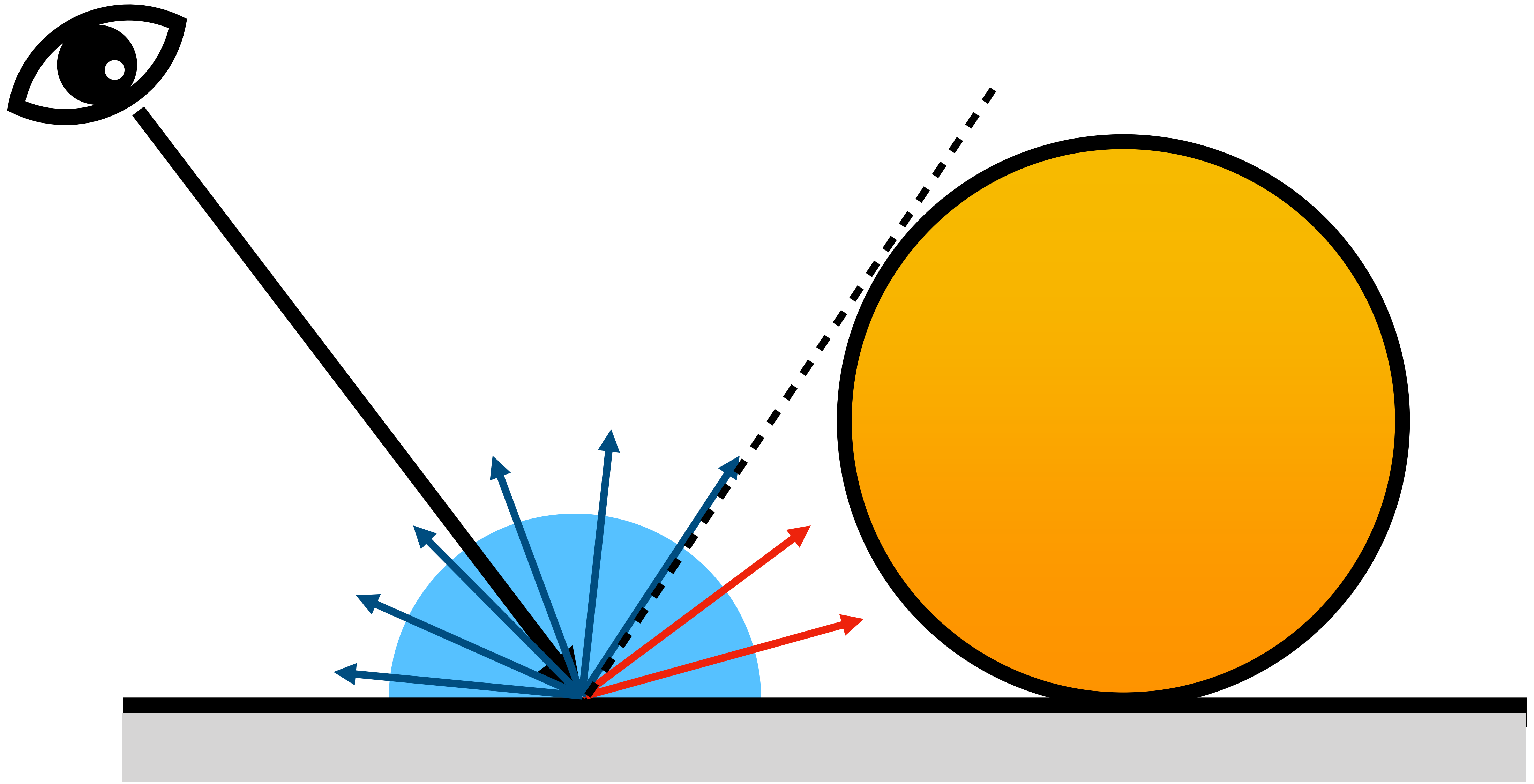
$$p(x, \omega_i) = \frac{1}{2\pi}$$

(256 Samples)



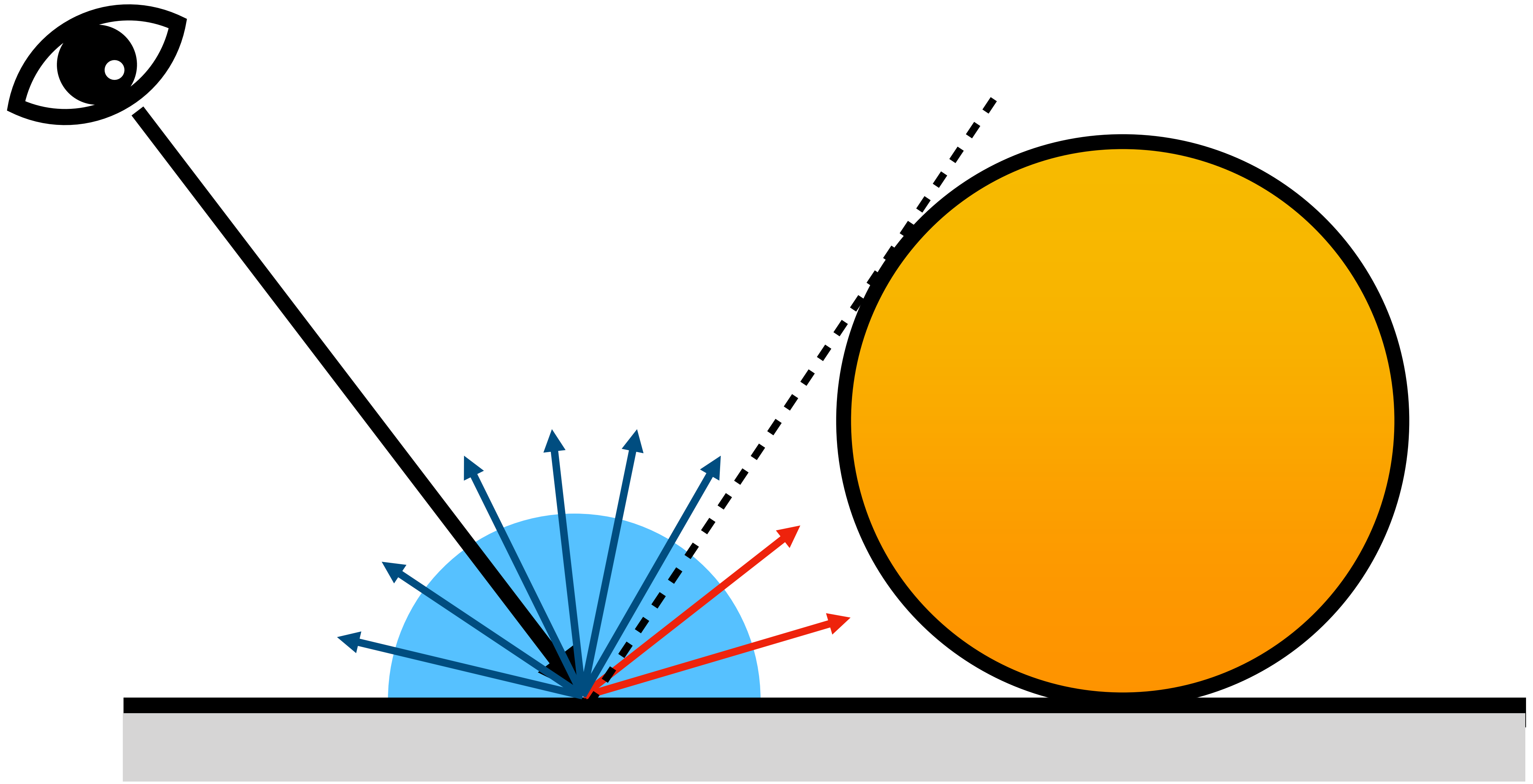
Importance Sampling: Cosine term

Uniform Hemispherical Sampling



$$p(x, \omega_i) = \frac{1}{2\pi}$$

Cosine-weighted Importance Sampling



$$p(x, \omega_i) = \cos\theta_i$$

Uniform hemispherical sampling 1 sample/pixel **Cosine-weighted importance sampling**

Slide from Wojciech Jarosz

Uniform hemispherical sampling

4 sample/pixel

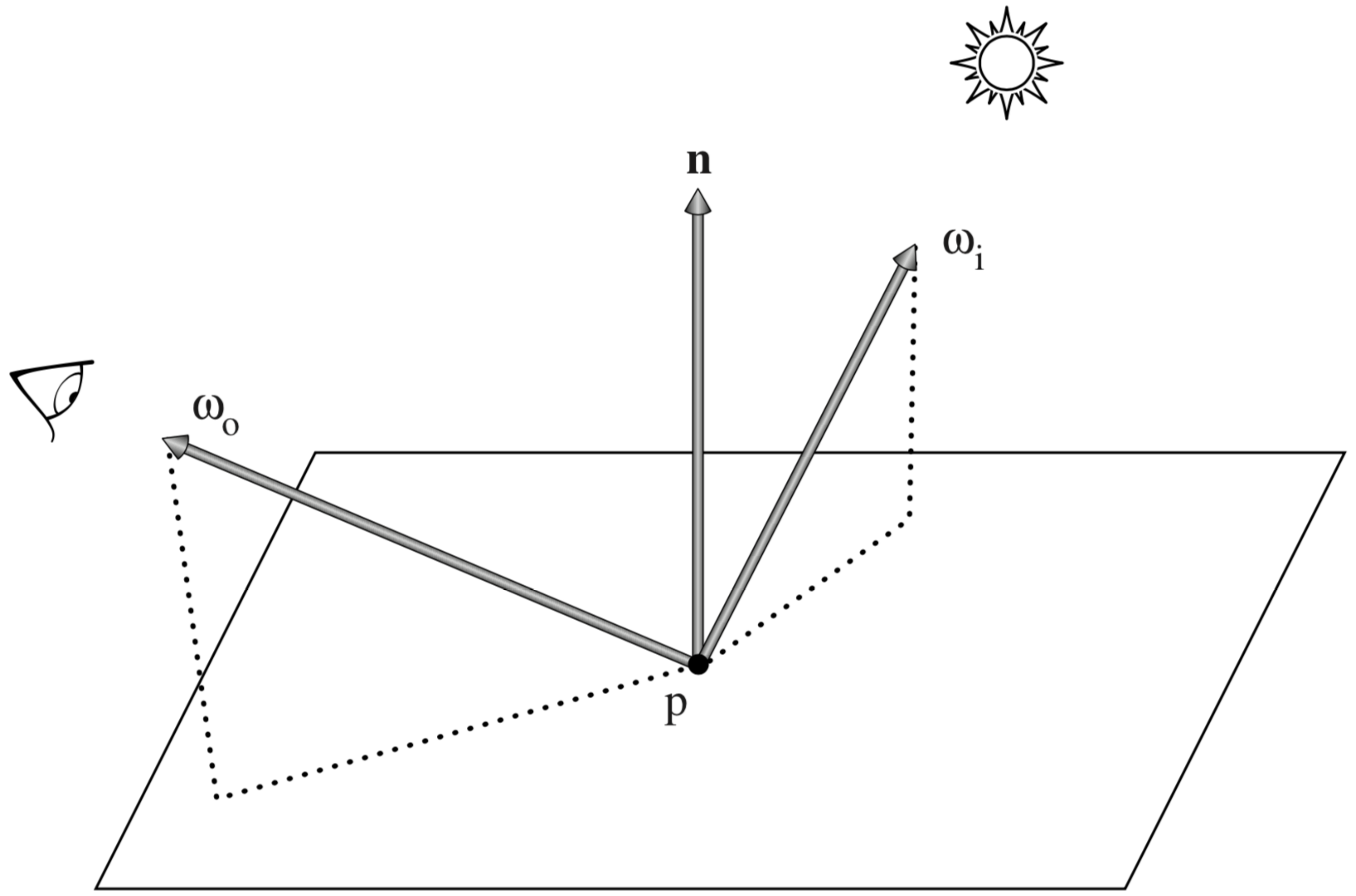
Cosine-weighted importance sampling

Slide from Wojciech Jarosz


Uniform hemispherical sampling
16 sample/pixel
Cosine-weighted importance sampling
Slide from Wojciech Jarosz

Importance Sampling: Incident Radiance

$$L_o(p,\omega) = \int_{\mathcal{H}^2} f(p,\omega_0,\omega_i) L_i(x,\omega_i) |\cos\theta_i| d\omega_i$$



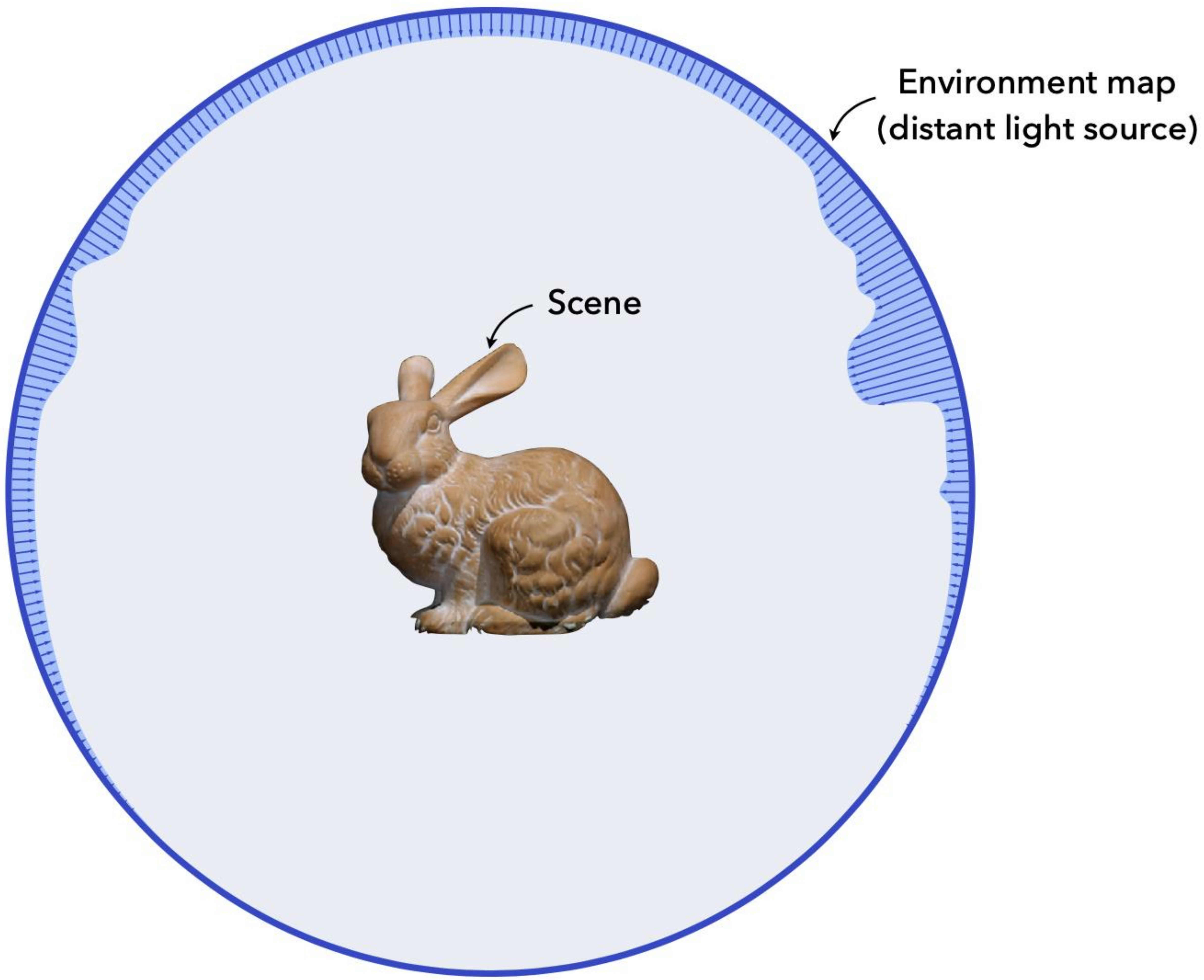
What terms can we importance sample?

- BSDF
- Incident radiance
- cosine term

Example: Environment Lighting

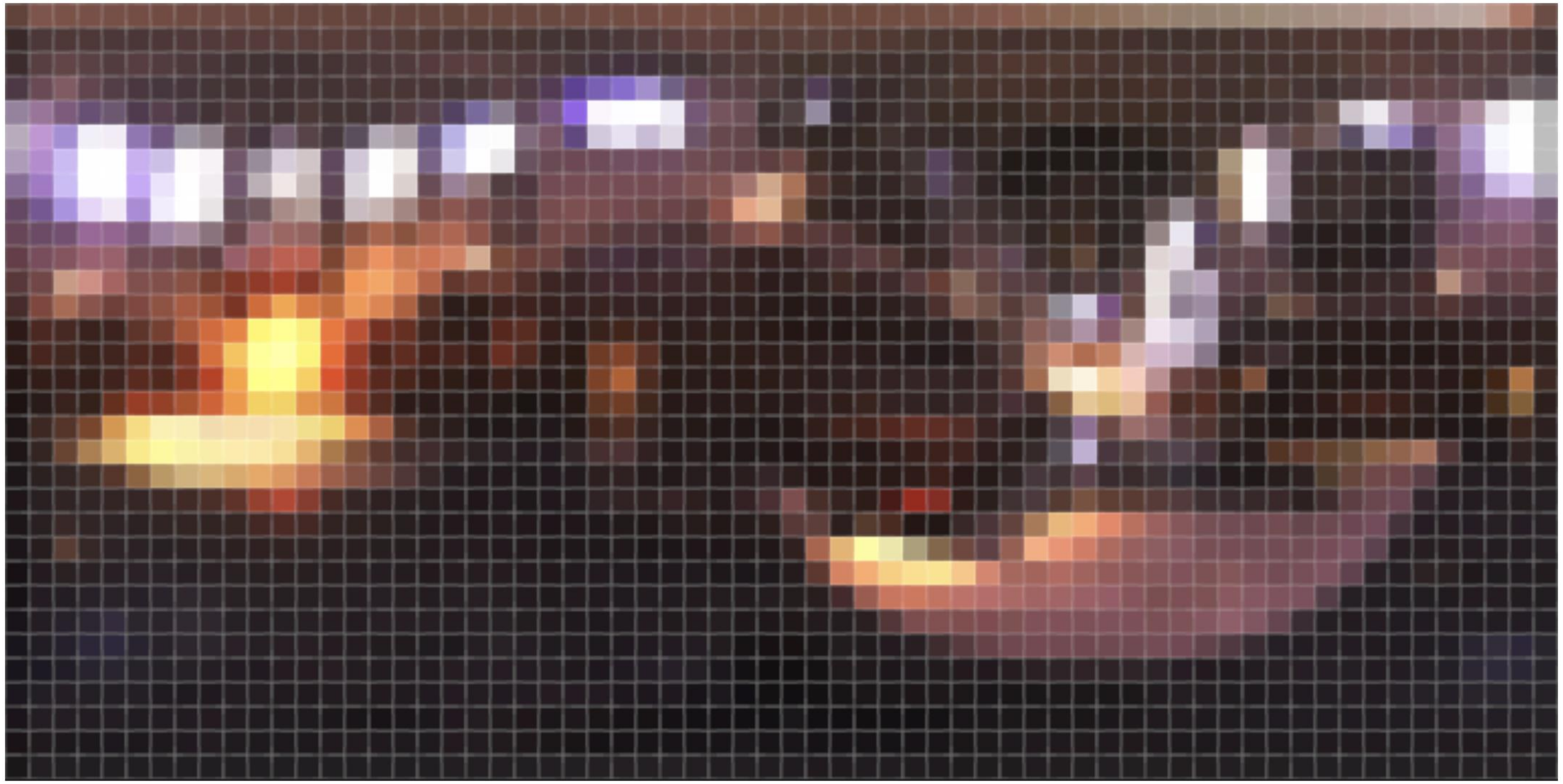
Example: Environment Lighting

Environment Lighting



Slide after Wojciech Jarosz

Importance function



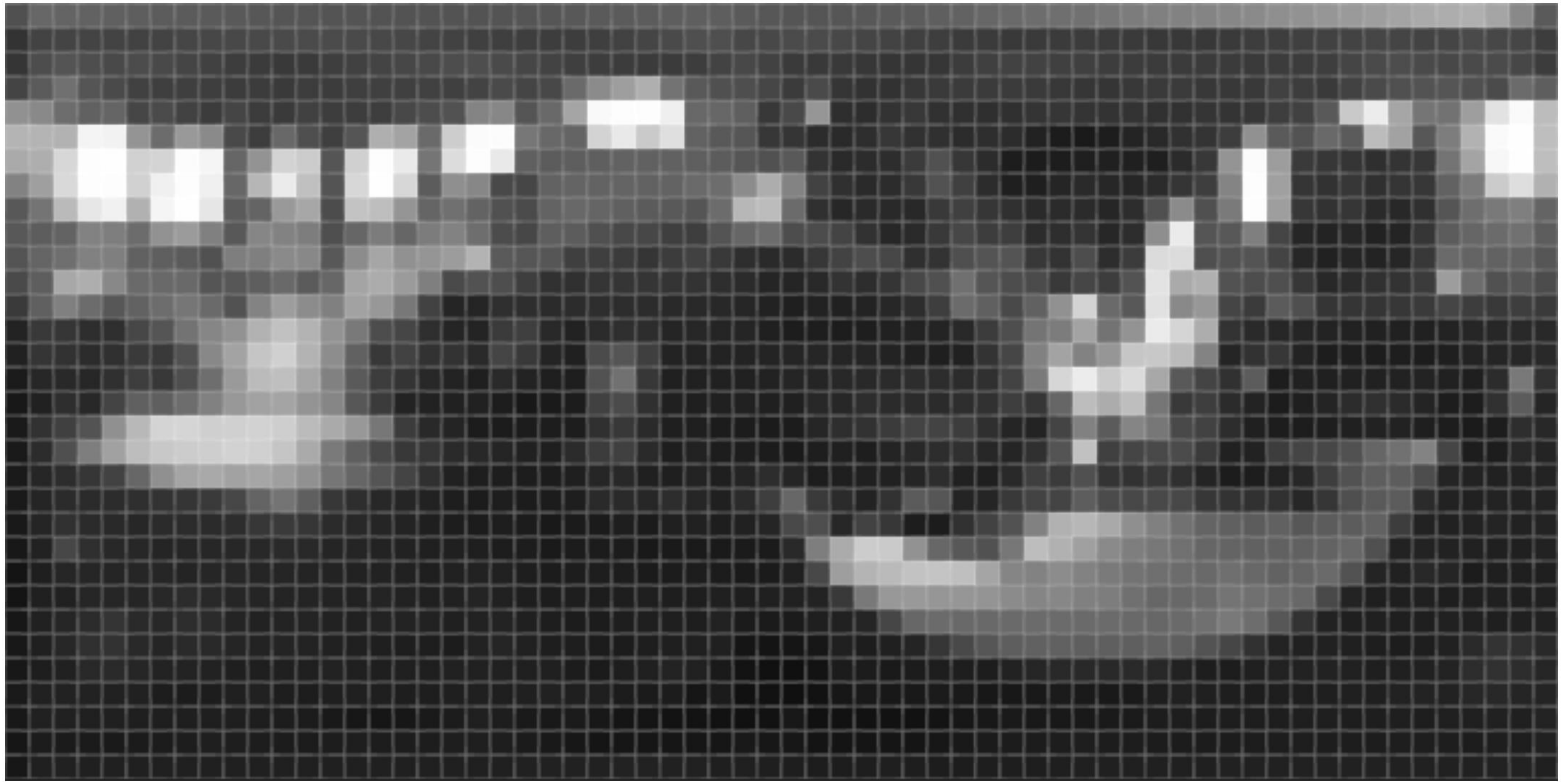
θ

ϕ

Slide after Wojciech Jarosz

Importance function

Scalar version e.g., luminance channel only



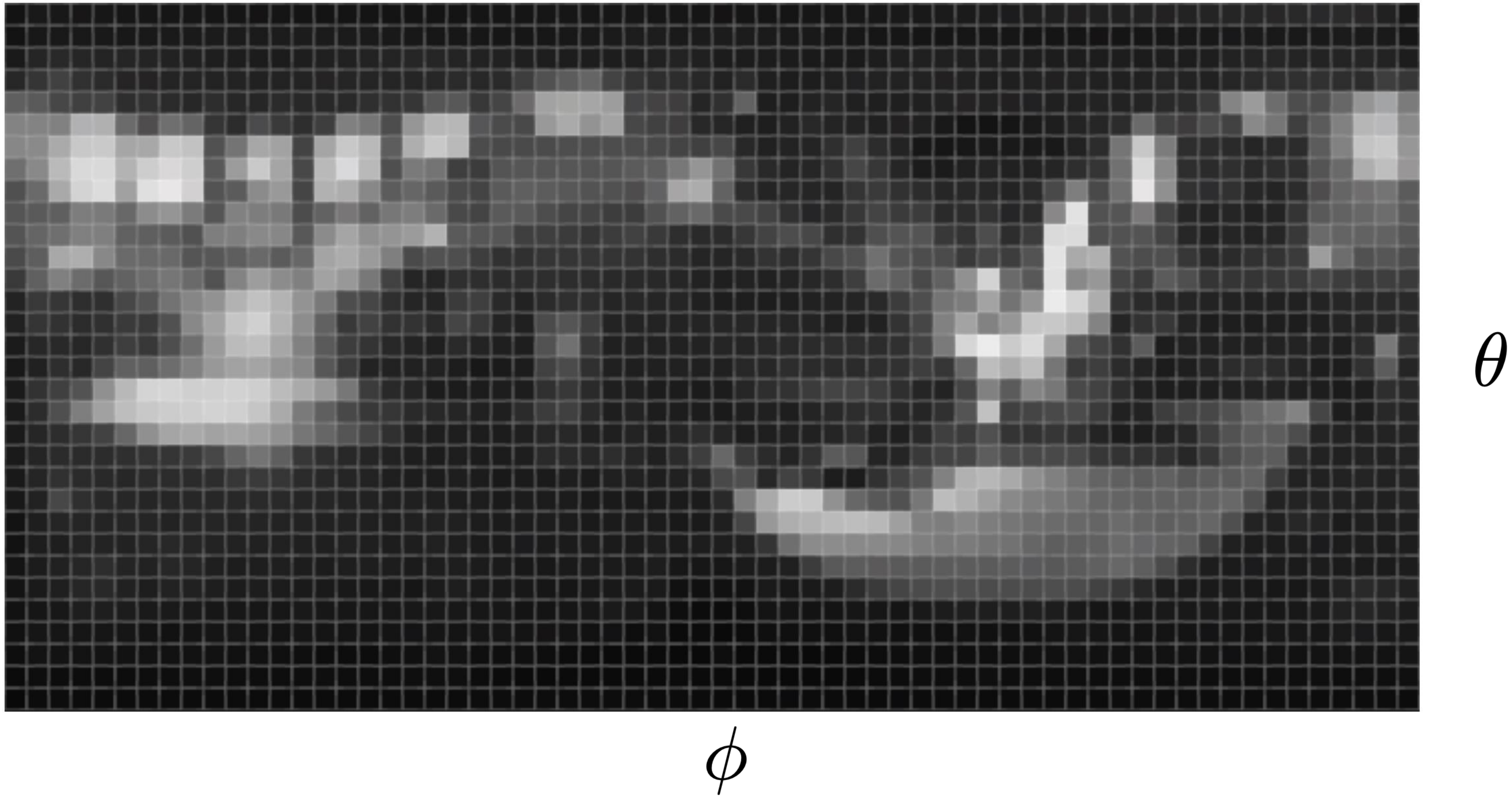
θ

ϕ

Slide after Wojciech Jarosz

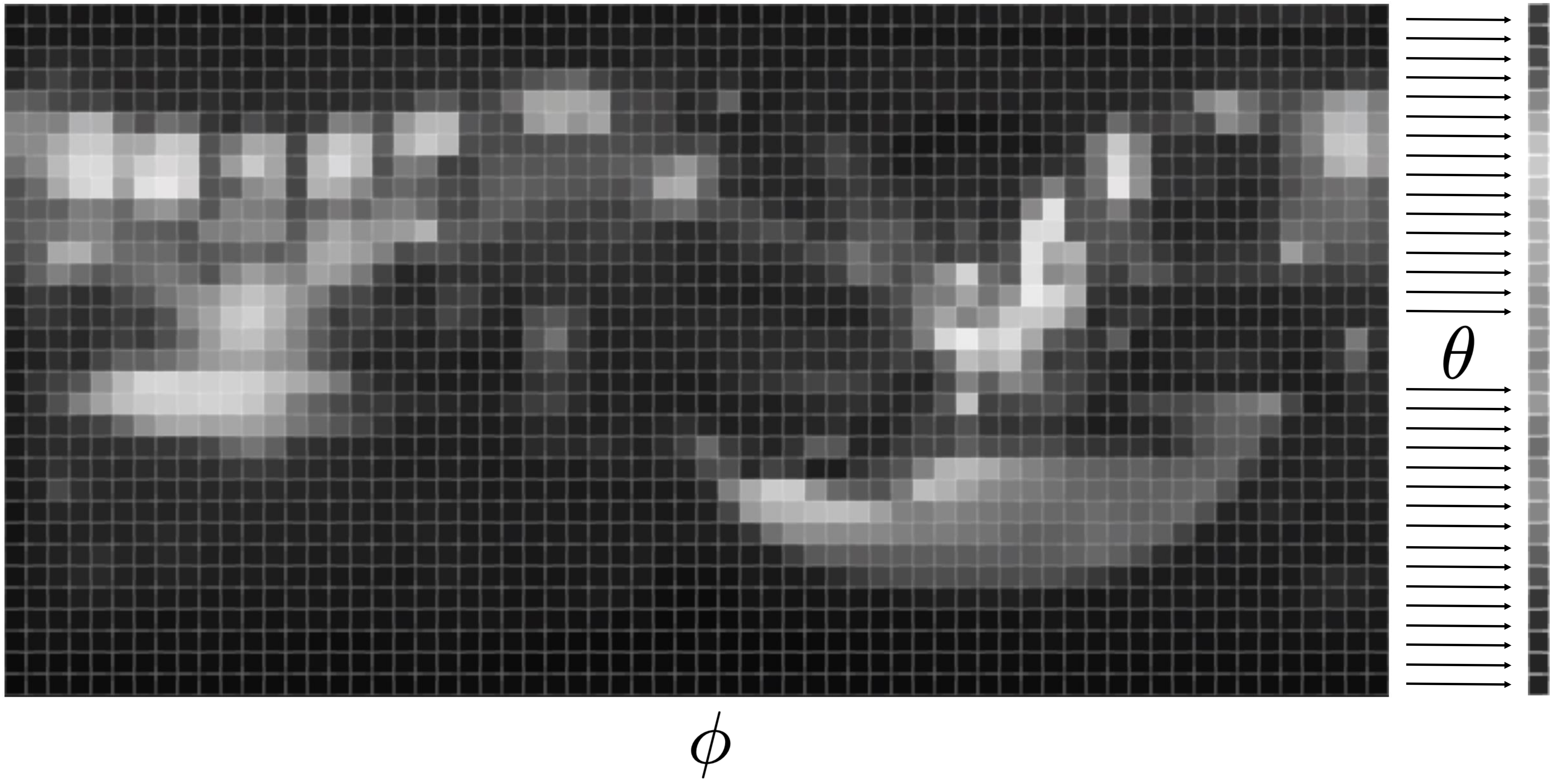
Importance function: Scalar function

Multiplication with $\sin\theta$



Slide after Wojciech Jarosz

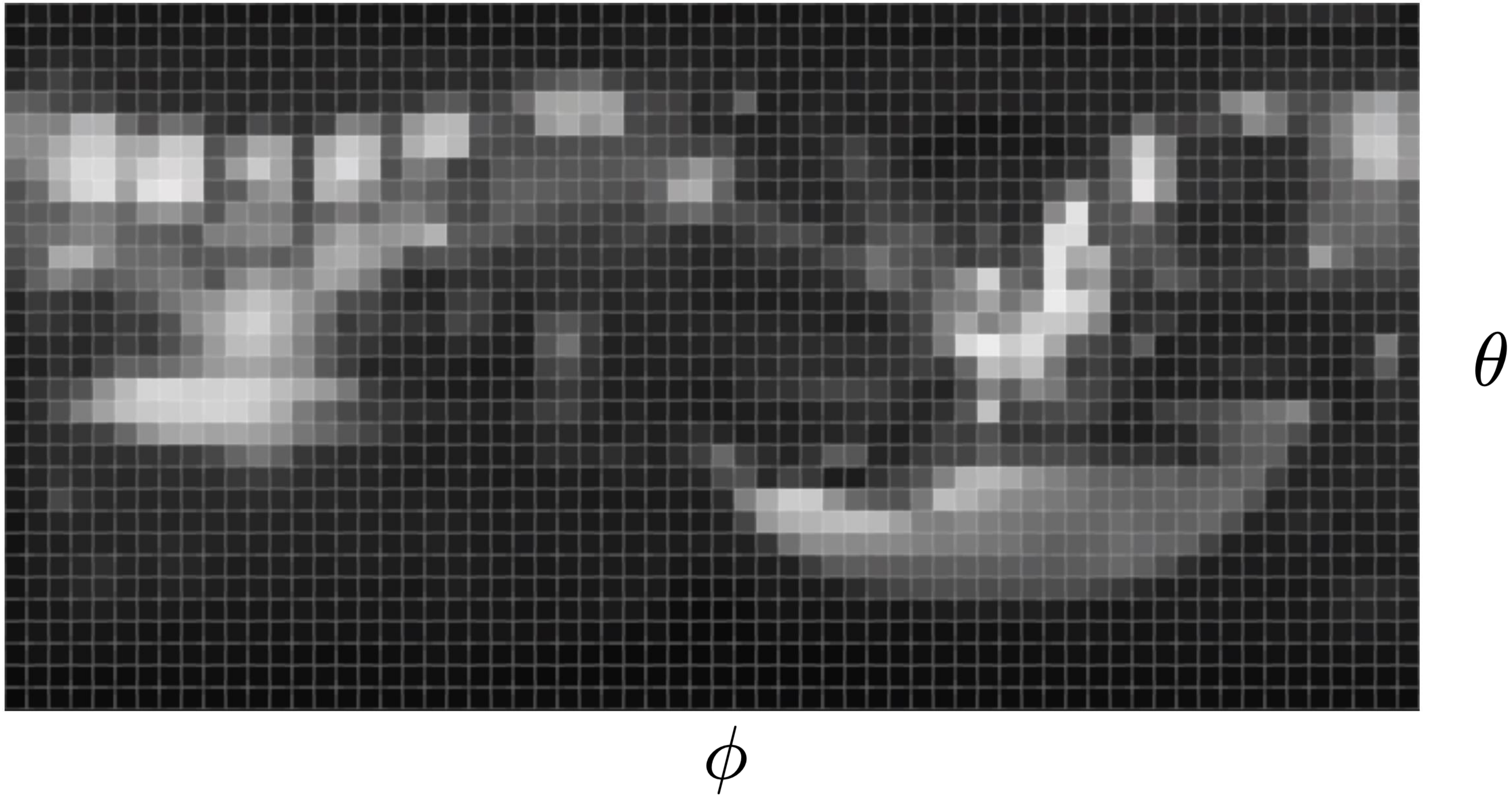
Importance function: Marginalization



Slide after Wojciech Jarosz

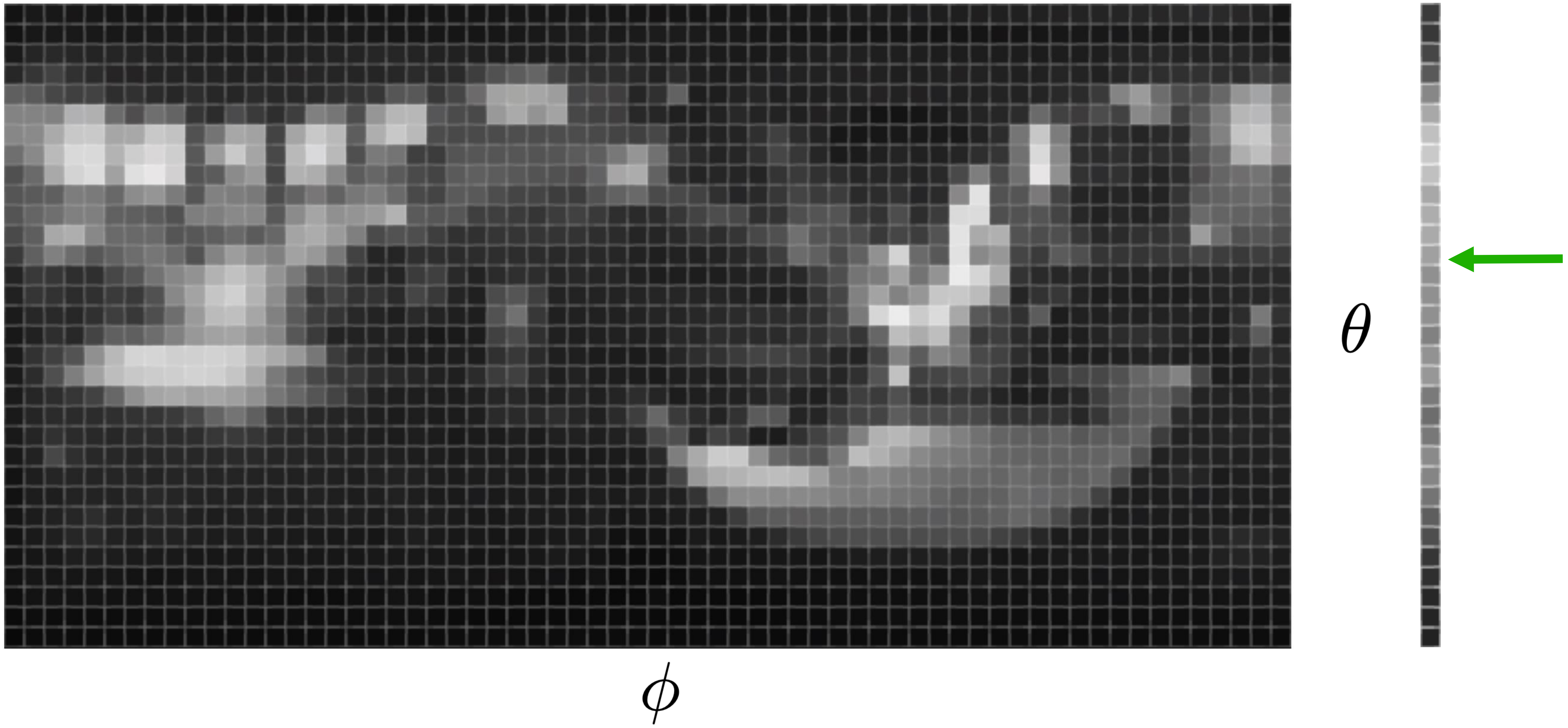
Importance function: Conditional PDFs

Once normalized, each row can serve as the conditional PDF



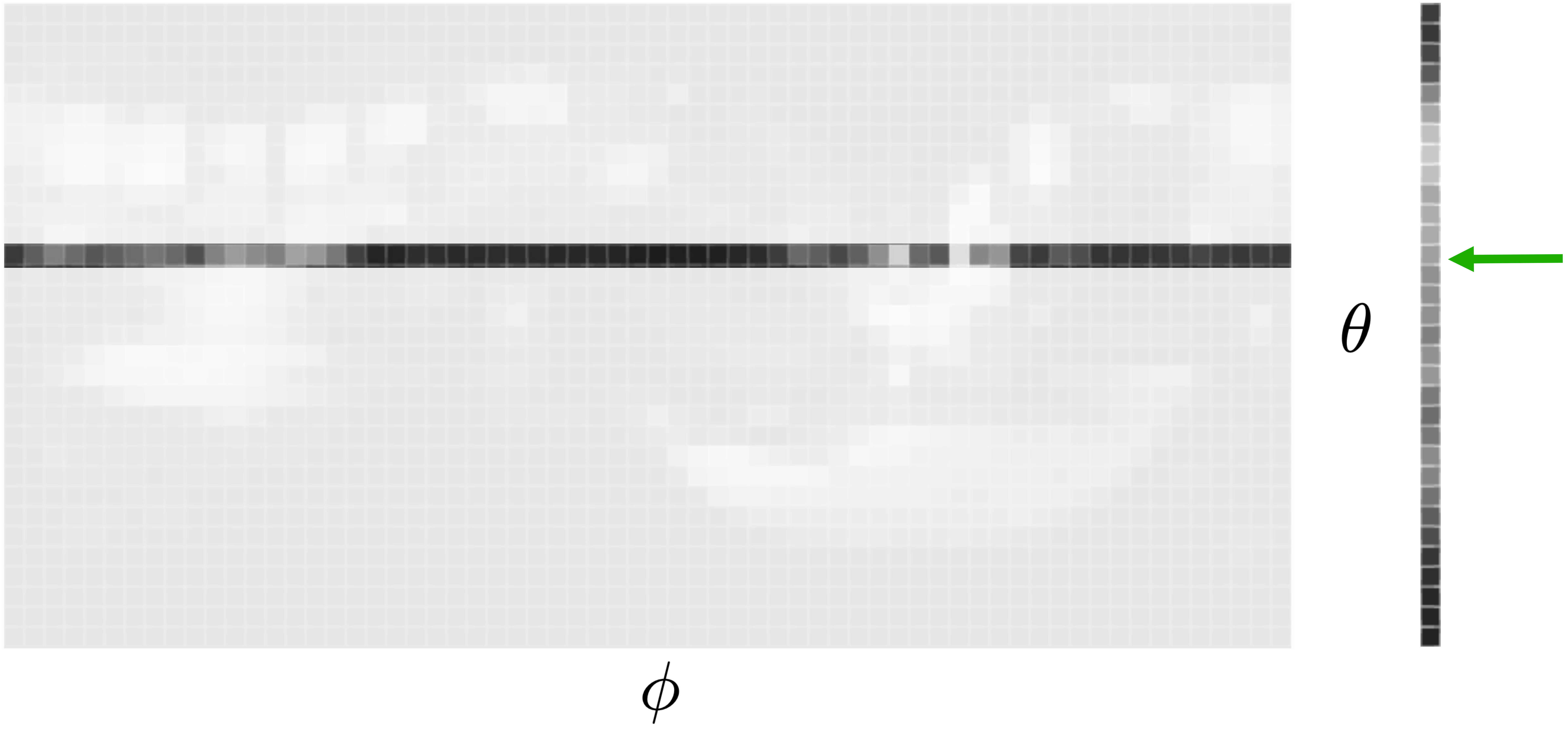
Slide after Wojciech Jarosz

Importance function: Sampling



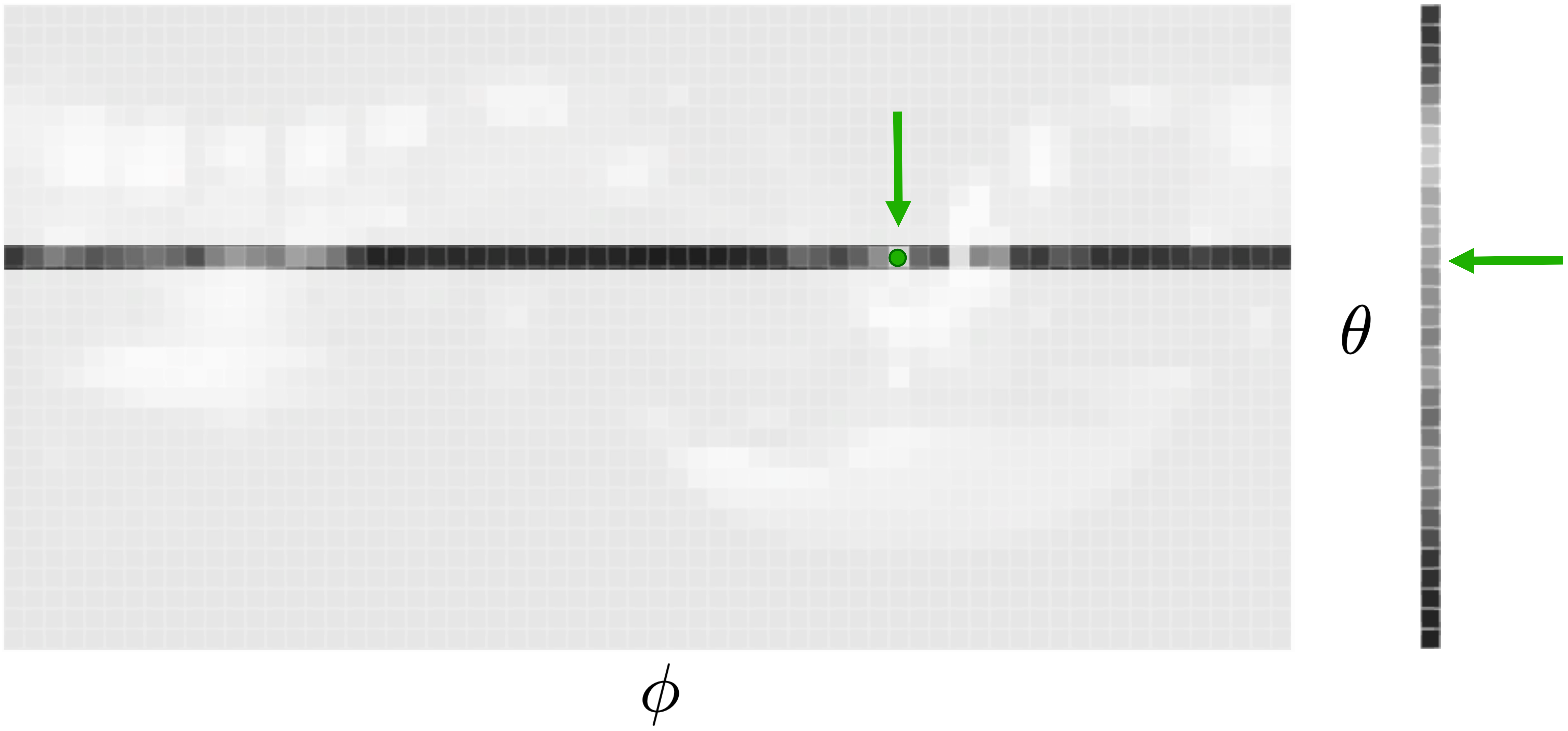
Slide after Wojciech Jarosz

Importance function: Sampling



Slide after Wojciech Jarosz

Importance function: Sampling



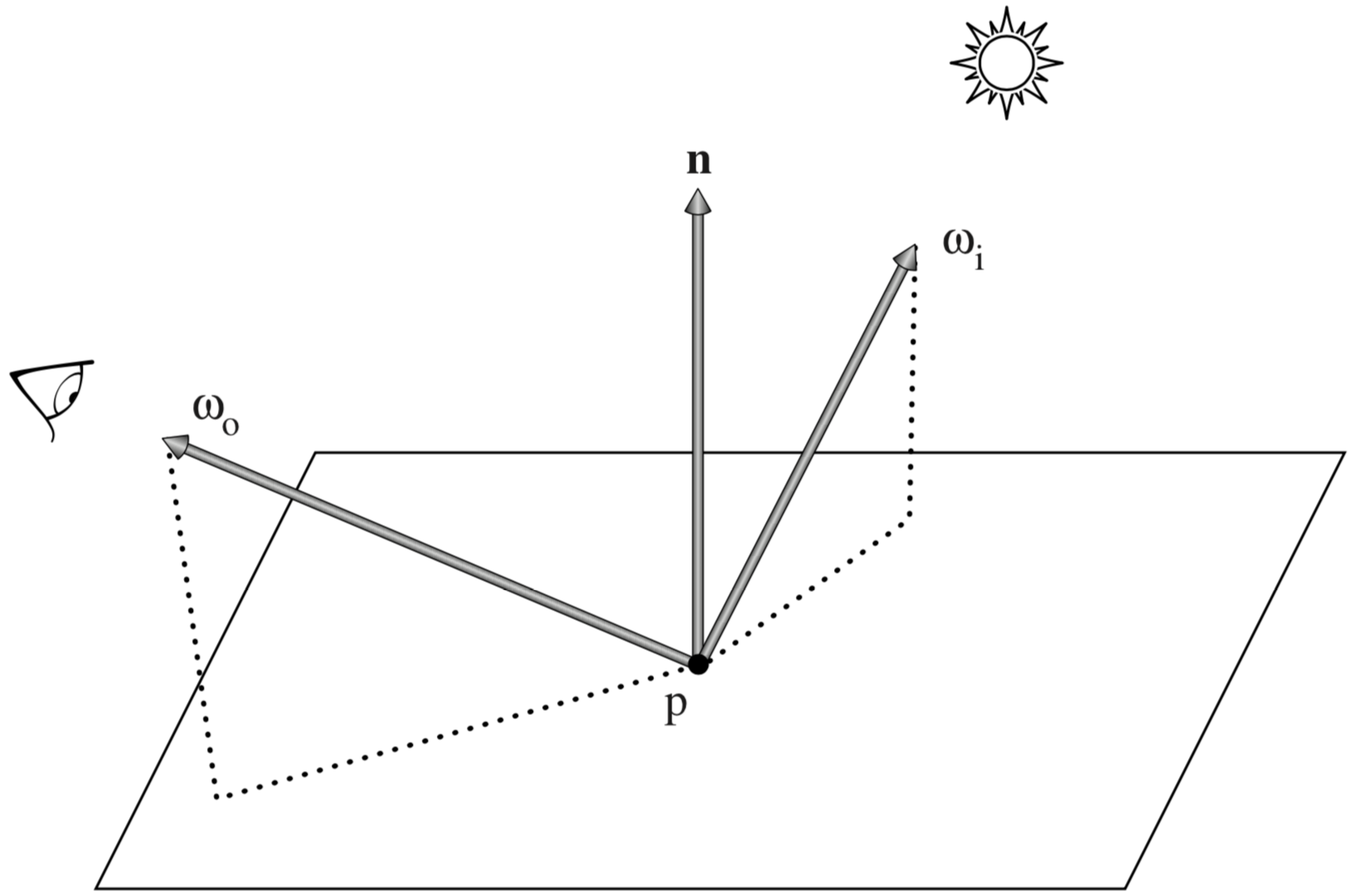
Slide after Wojciech Jarosz

Importance Sampling

For more details, see PBRTv3: 13.2 and 13.6.7

Importance Sampling

$$L_o(p,\omega) = \int_{\mathcal{H}^2} f(p,\omega_0,\omega_i) L_i(x,\omega_i) |\cos\theta_i| d\omega_i$$



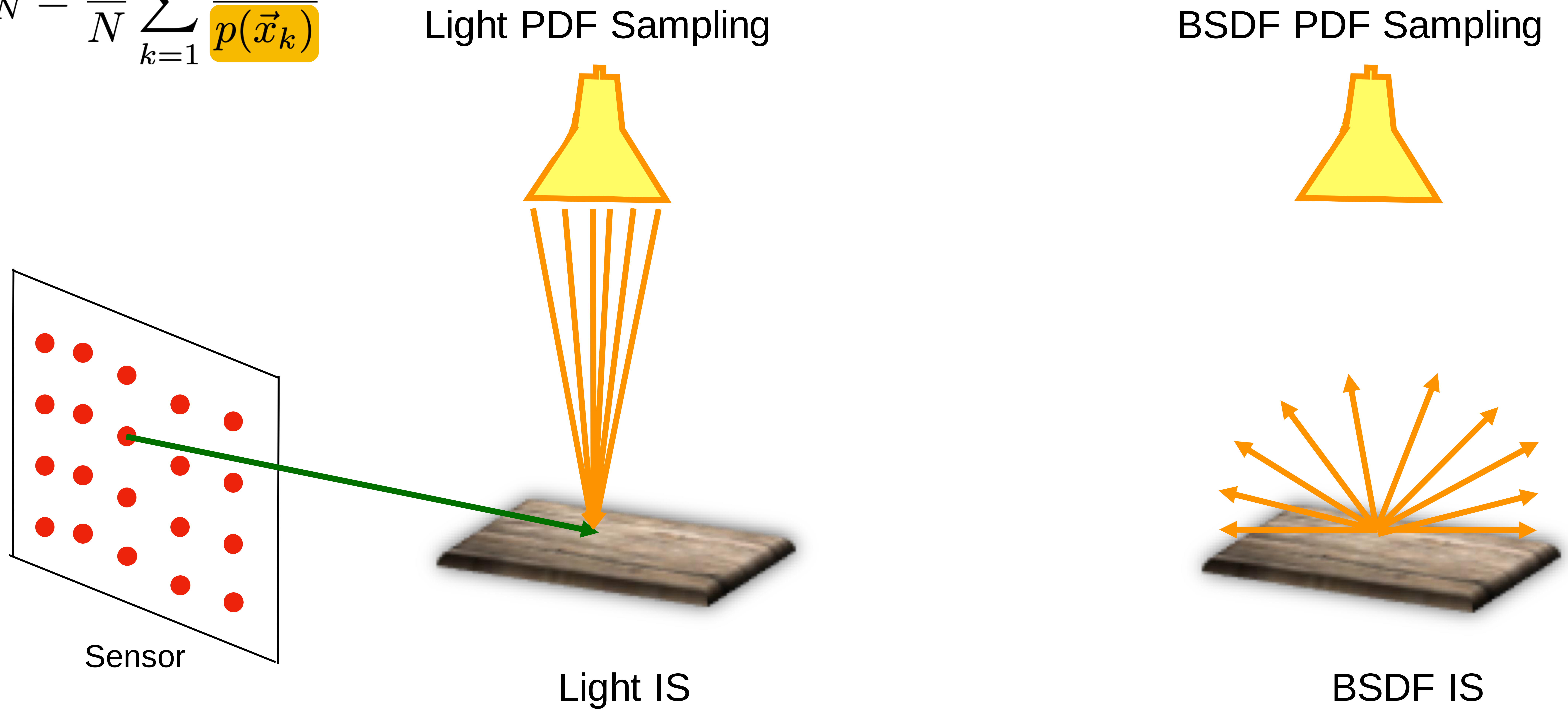
What terms can we importance sample?

- BSDF
- Incident radiance
- cosine term

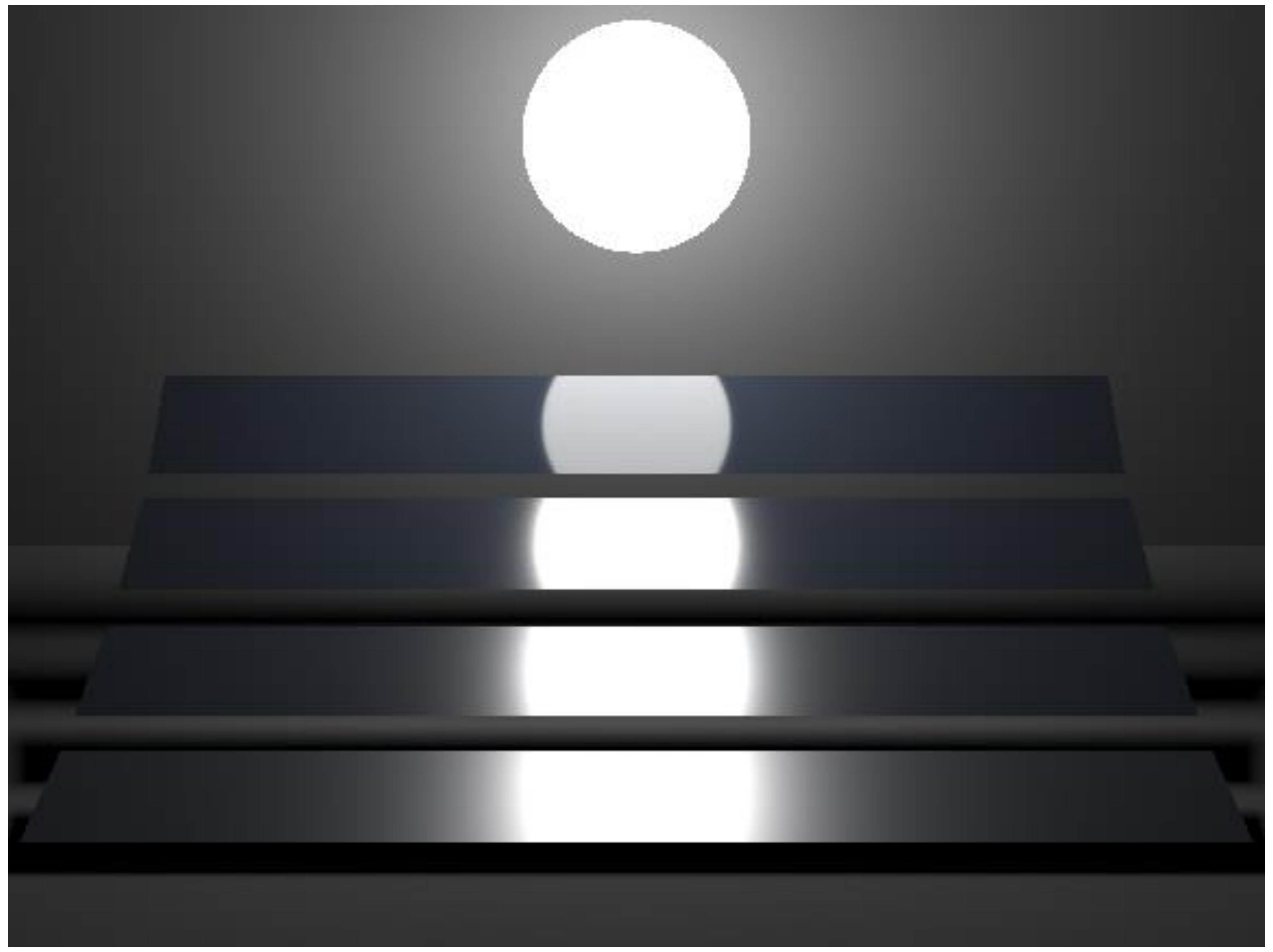
To handle this, we will introduce Microfacet BSDF theory in the later part of the lecture.

Light vs. BSDF Importance Sampling

$$I_N = \frac{1}{N} \sum_{k=1}^{N} \frac{f(\vec{x}_k)}{p(\vec{x}_k)}$$

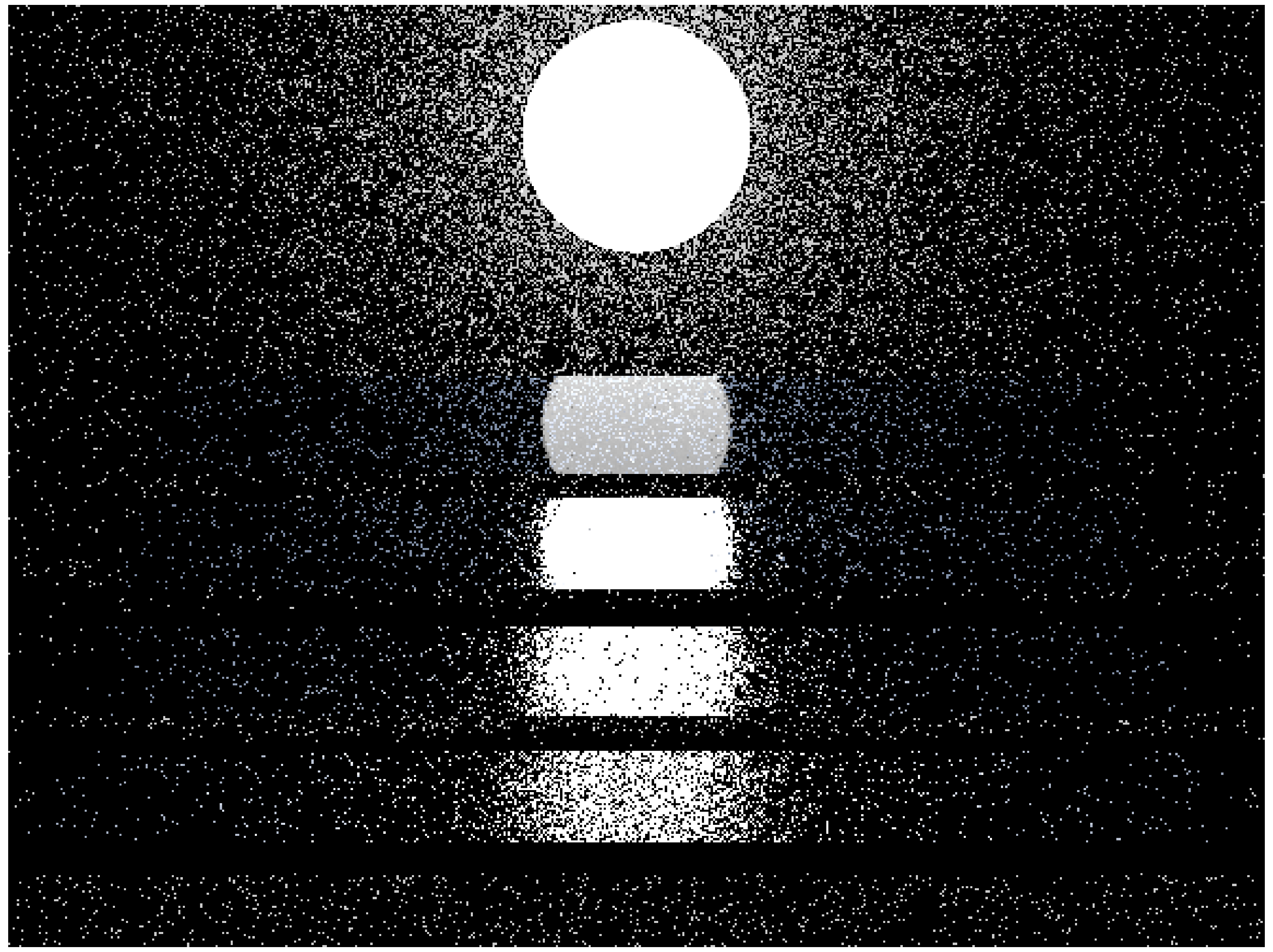


Variance reduction: Importance sampling



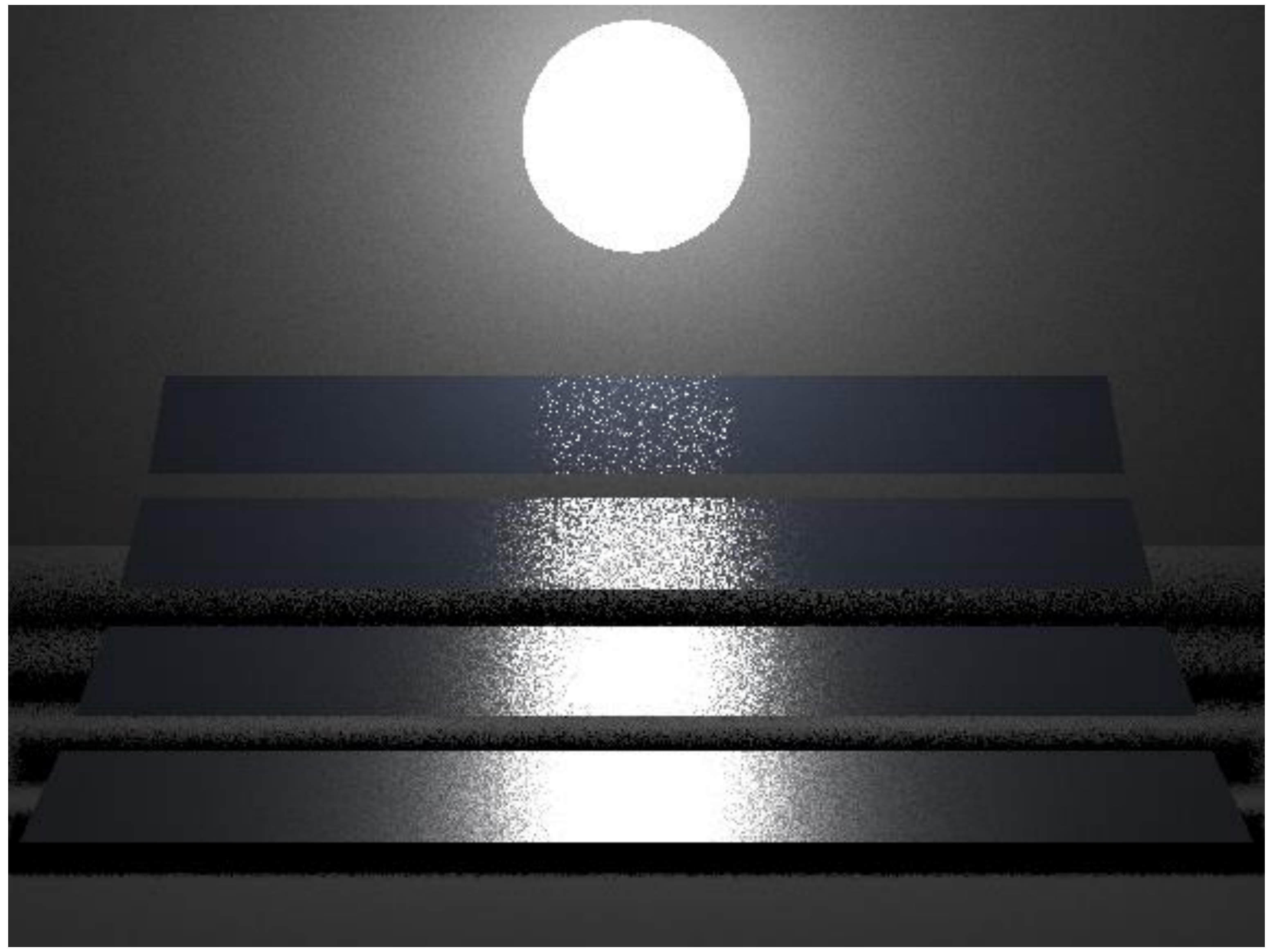
Reference image

N = 1024 spp



BSDF importance sampling

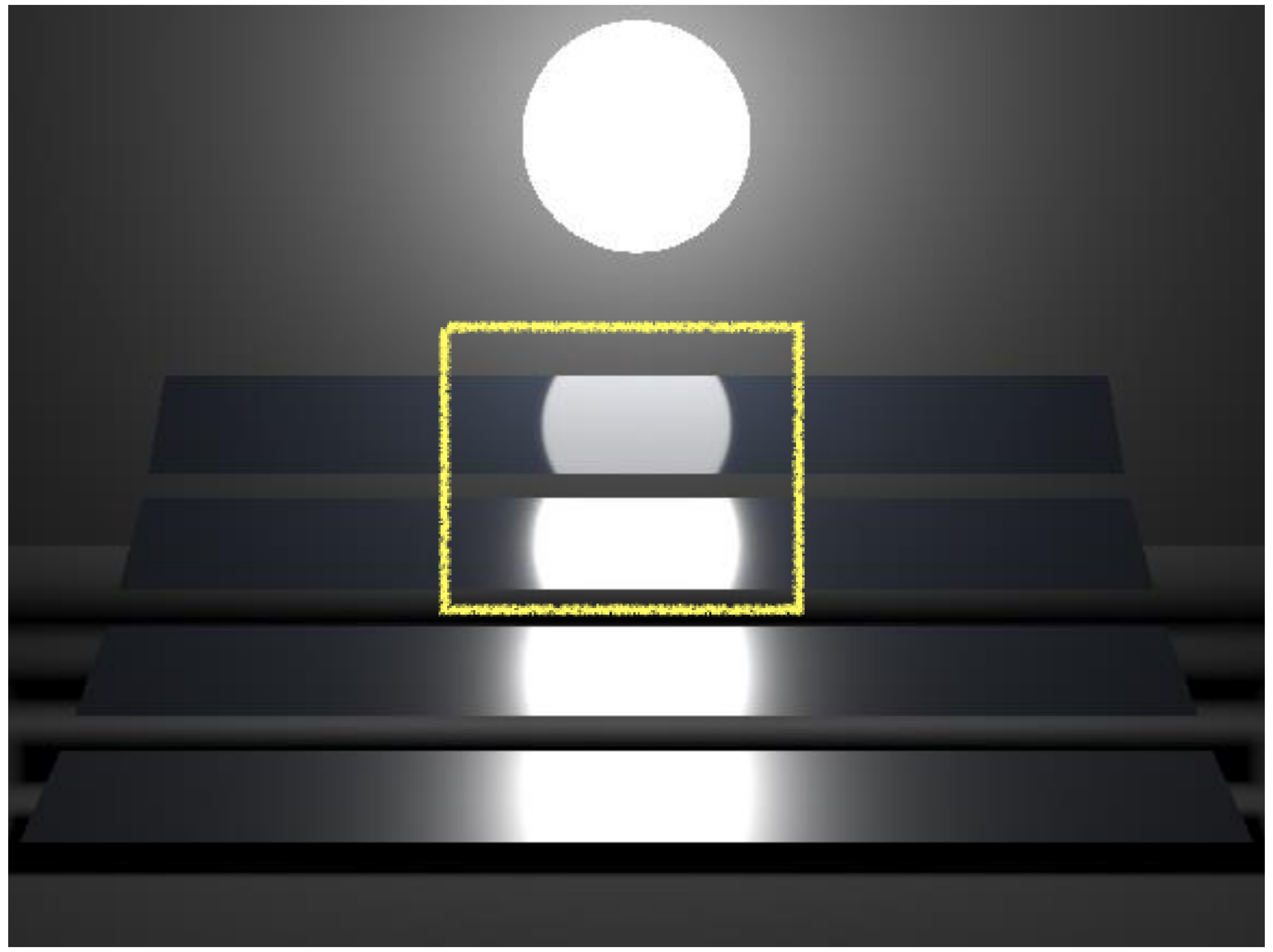
N = 4 spp



Light importance sampling

N = 4 spp

Variance reduction: Importance sampling

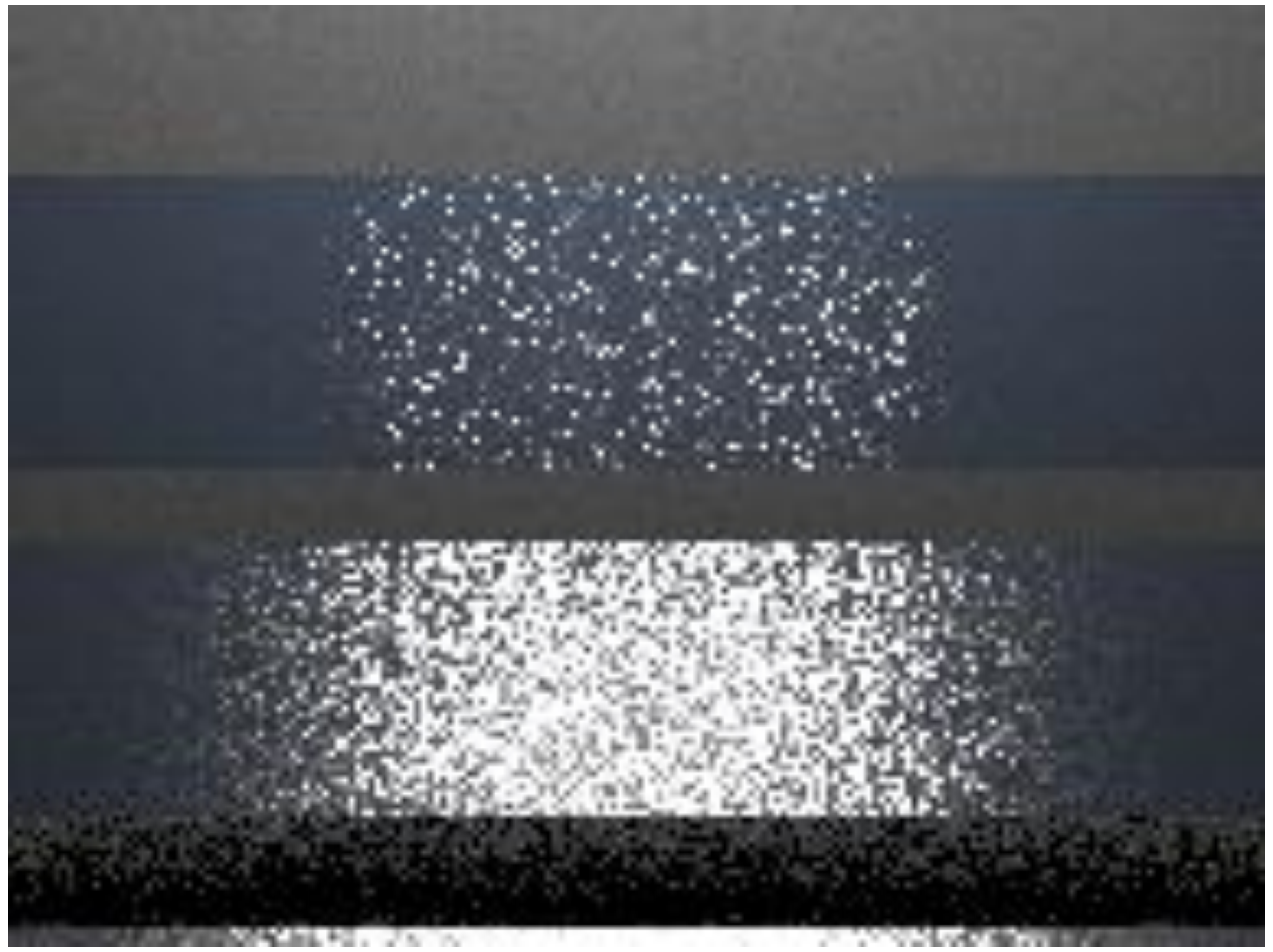


Reference image

N = 1024 spp

BSDF importance sampling

N = 4 spp

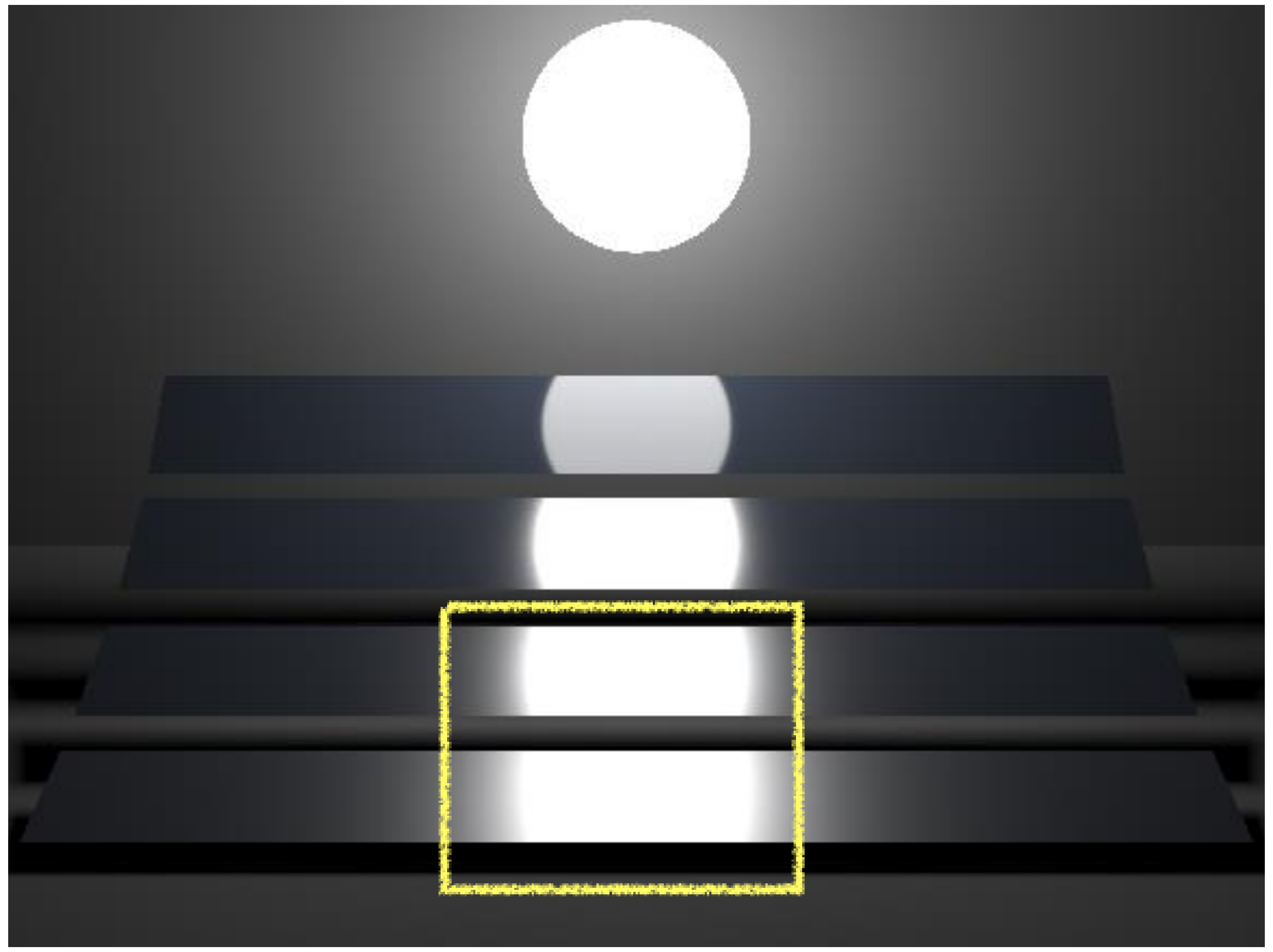


Light importance sampling

N = 4 spp

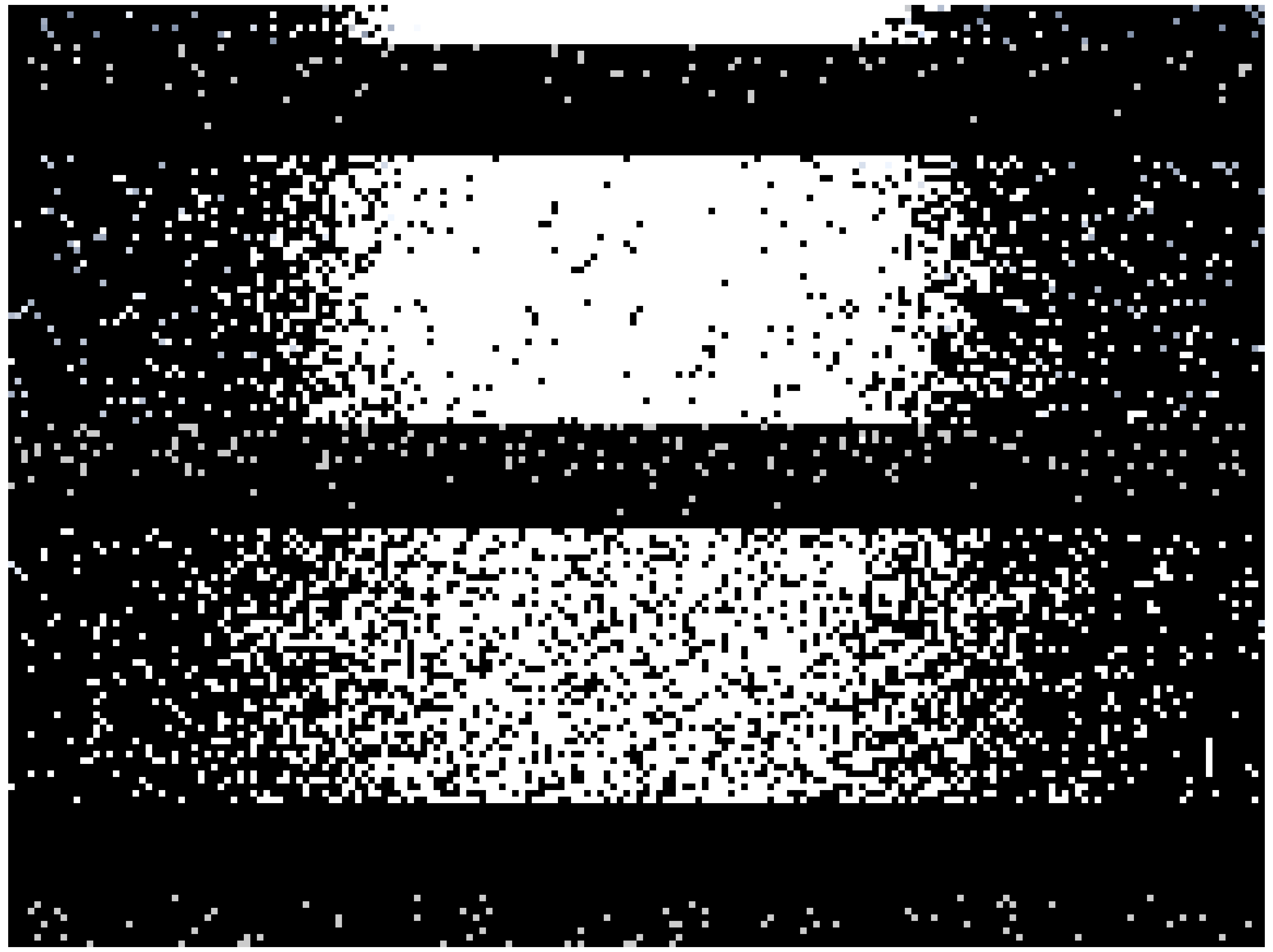
BSDF sampling is better in some regions

Variance reduction: Importance sampling



Reference image

N = 1024 spp



BSDF importance sampling

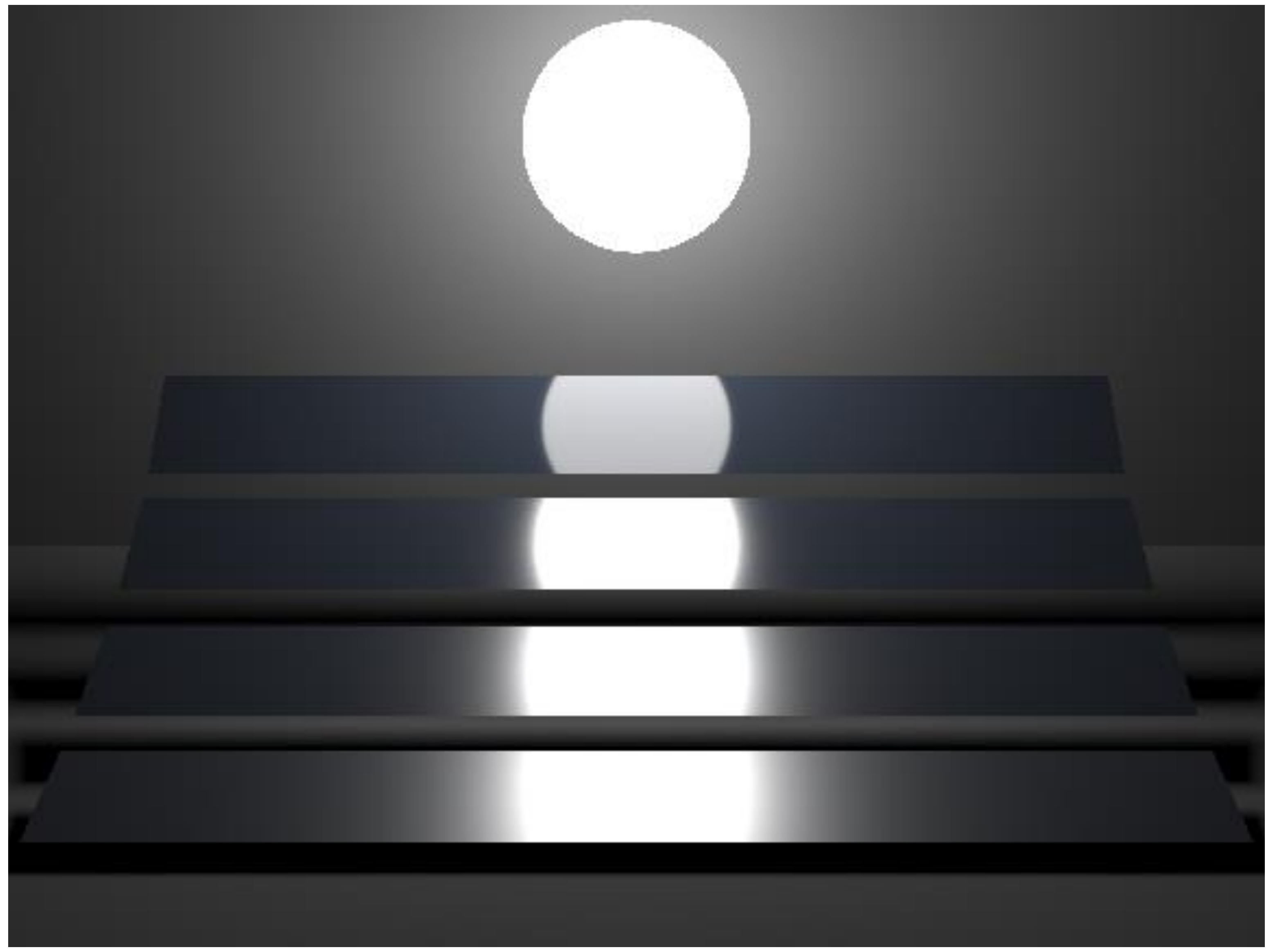
N = 4 spp

Light importance sampling

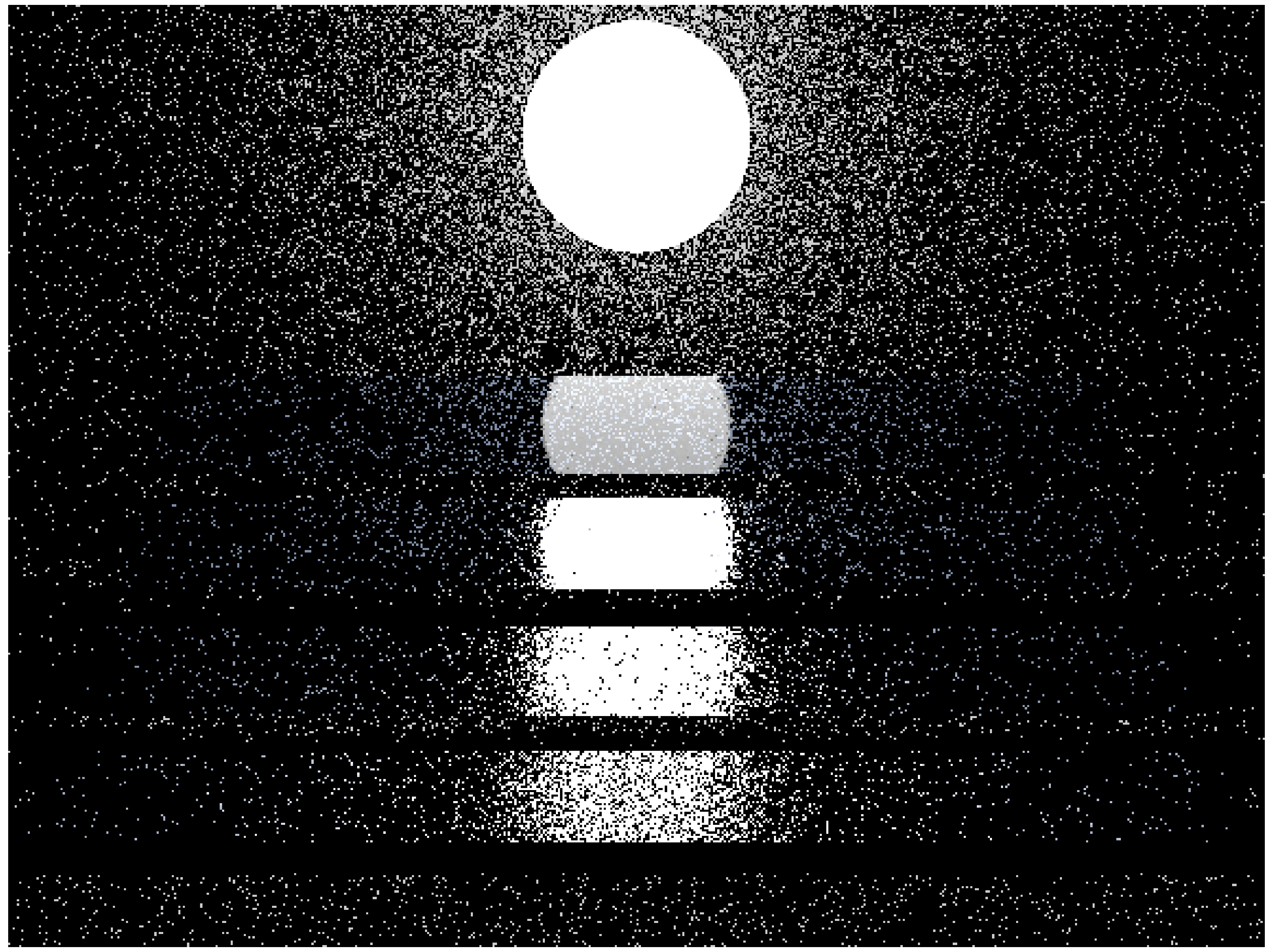
N = 4 spp

Light sampling is better in other regions

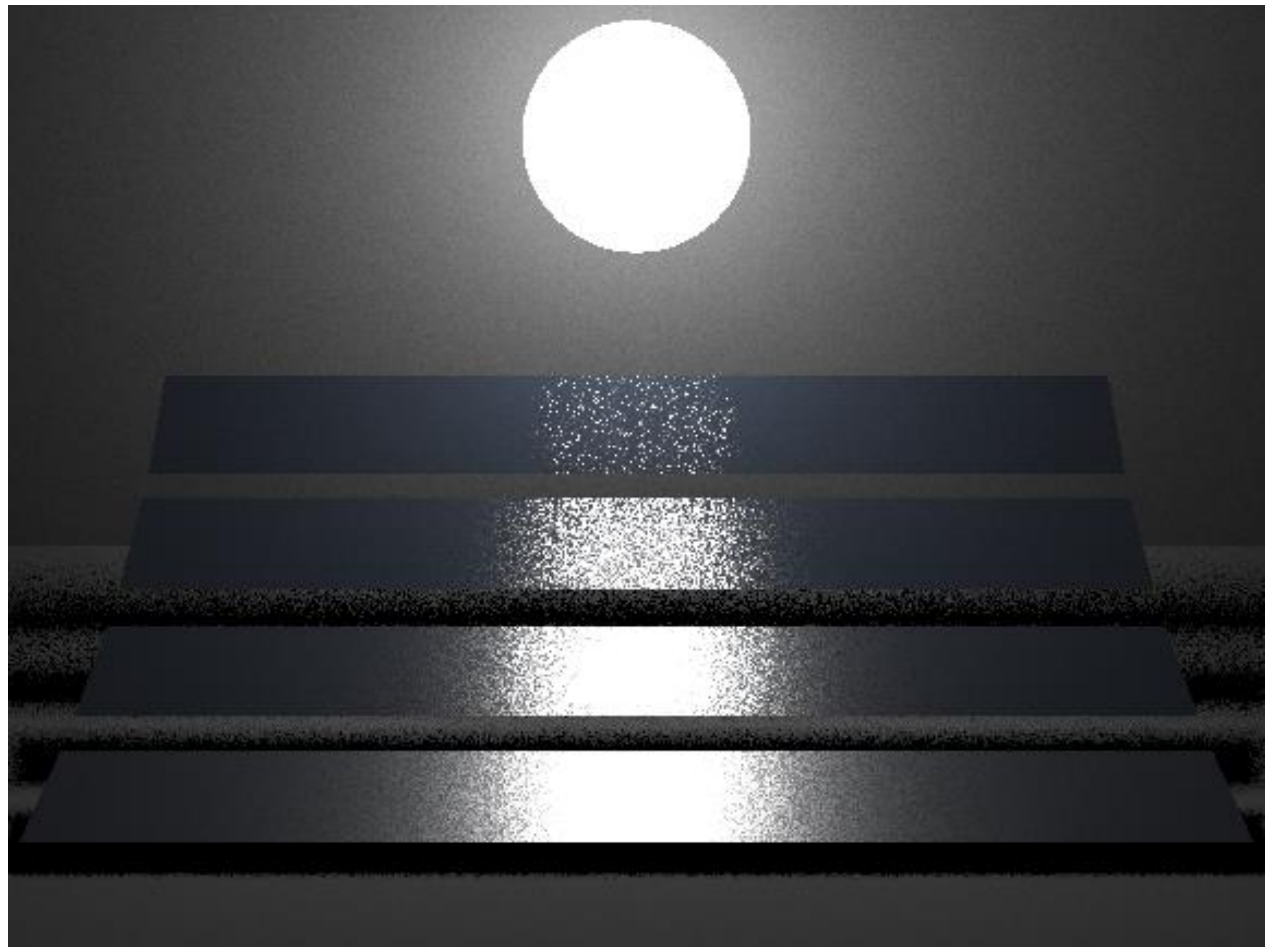
Variance reduction: Importance sampling



Reference image



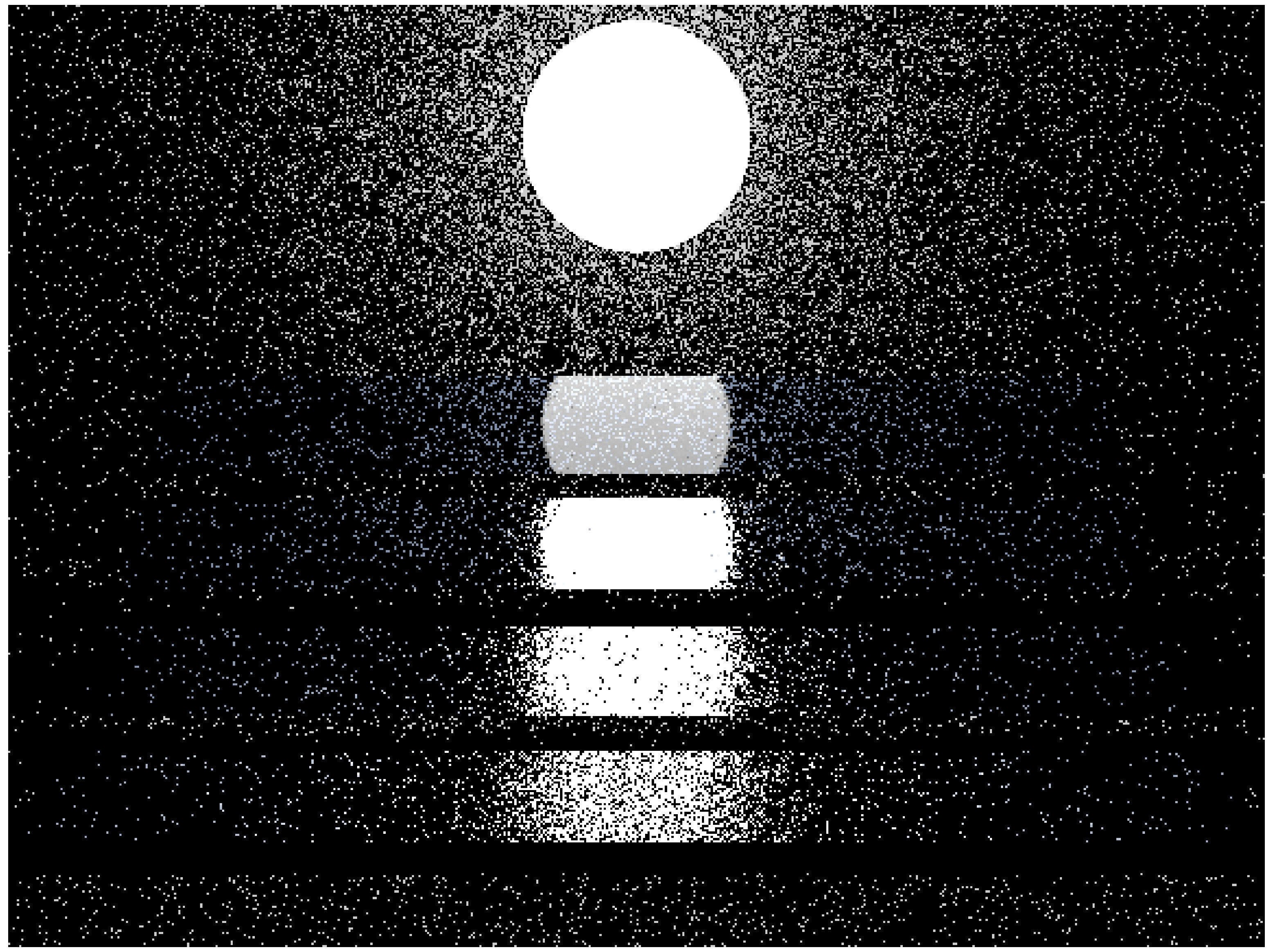
BSDF importance sampling

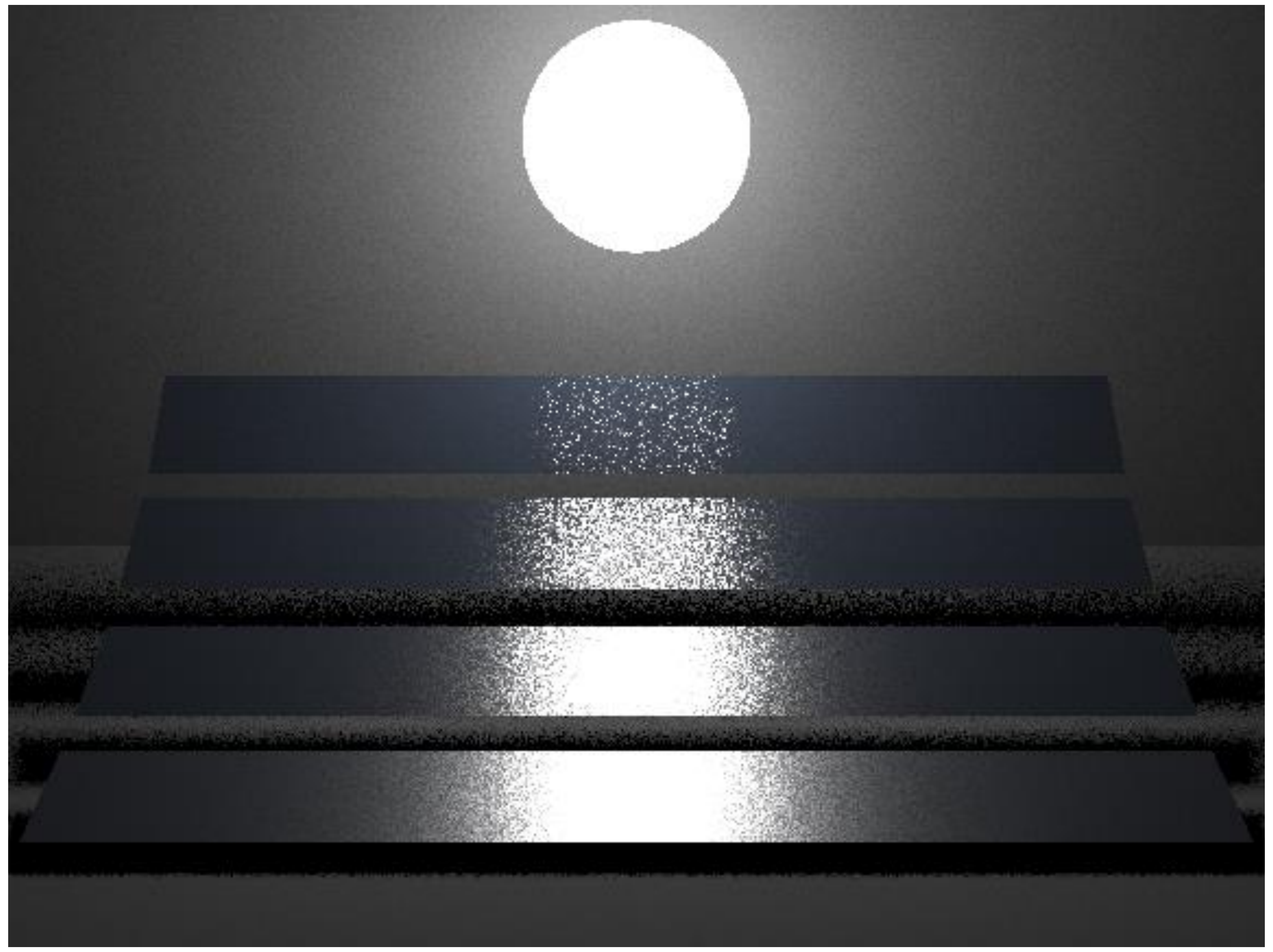


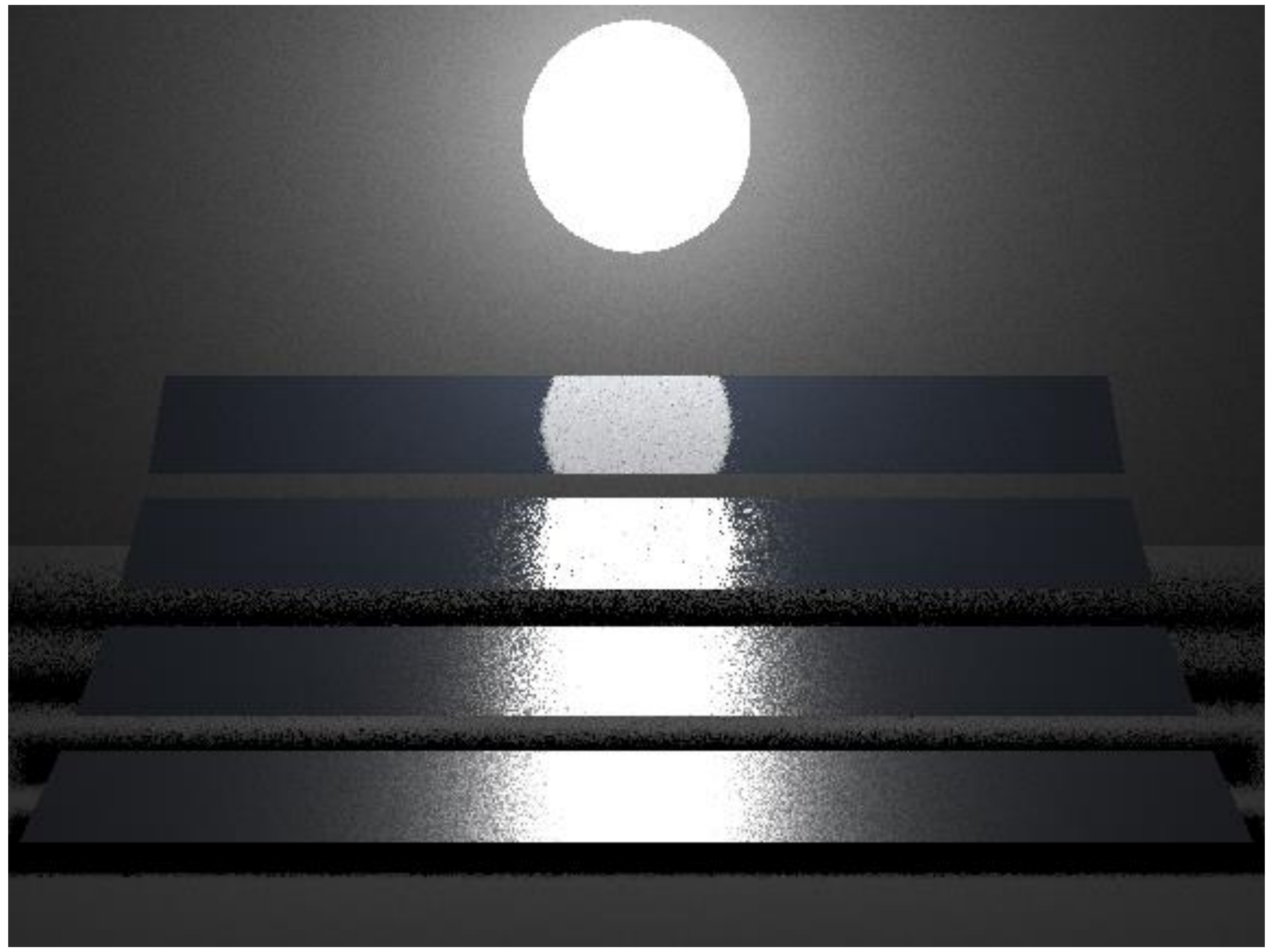
Light importance sampling

Can we combine the benefits of different PDFs ? **Yes!**

Variance reduction: Importance sampling







BSDF importance sampling | Light importance sampling | **Multiple Importance Sampling**

Can we combine the benefits of different PDFs ? **Yes!**

Variance reduction: Multiple Importance sampling

Multiple Importance Sampling

$$I_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x)g(x)}{p(x)}$$

$$p(x) \propto ???$$

$$\mathbf{I}_N = \frac{1}{n_f}\sum_{i=1}^{n_f}\frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)} + \frac{1}{n_g}\sum_{j=1}^{n_g}\frac{f(Y_j)g(Y_j)w_g(Y_j)}{p_g(Y_j)}$$

Variance reduction: Multiple Importance sampling

Multiple Importance Sampling

$$\mathbf{I}_N = \frac{1}{n_f}\sum_{i=1}^{n_f}\frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)} + \frac{1}{n_g}\sum_{j=1}^{n_g}\frac{f(Y_j)g(Y_j)w_g(Y_j)}{p_g(Y_j)}$$

Balance heuristic: $$w_s(x) = \frac{n_s p_s(x)}{\sum_i n_i p_i(x)}$$

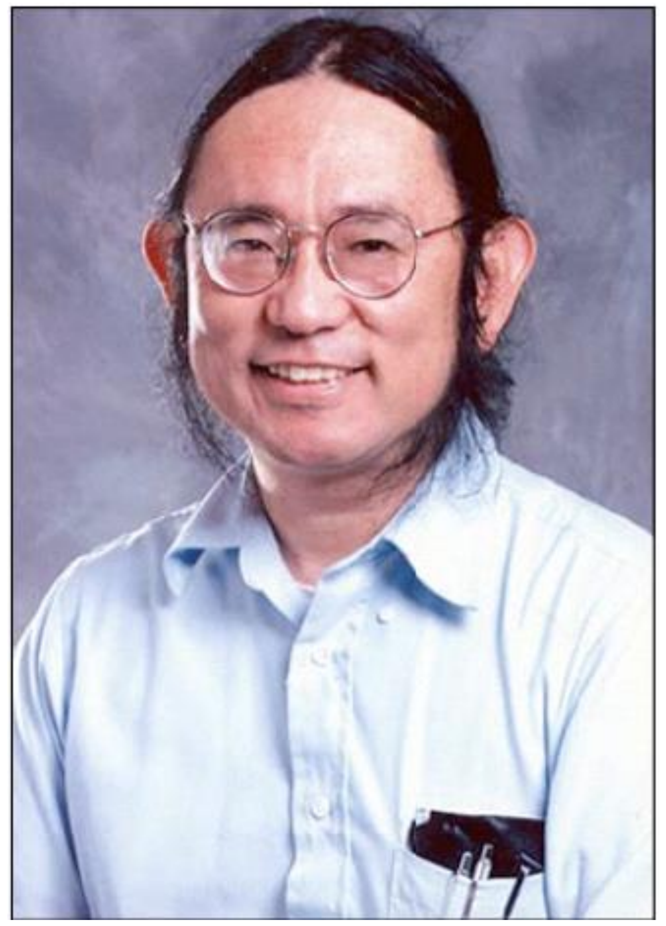
Power heuristic: $$w_s(x) = \frac{(n_s p_s(x))^\beta}{\sum_i (n_i p_i(x))^\beta}$$ $\beta = 2$

Questions?

Gaussian Material Synthesis by Zsolnai-Feher, Wonka, Wimmer [SIGGRAPH 2018]

Rendering Equation

Rendering Equation



$$L_o(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o)$$

Outgoing emitted reflected

James Kajiya, The Rendering Equation, SIGGRAPH 1986

Rendering Equation

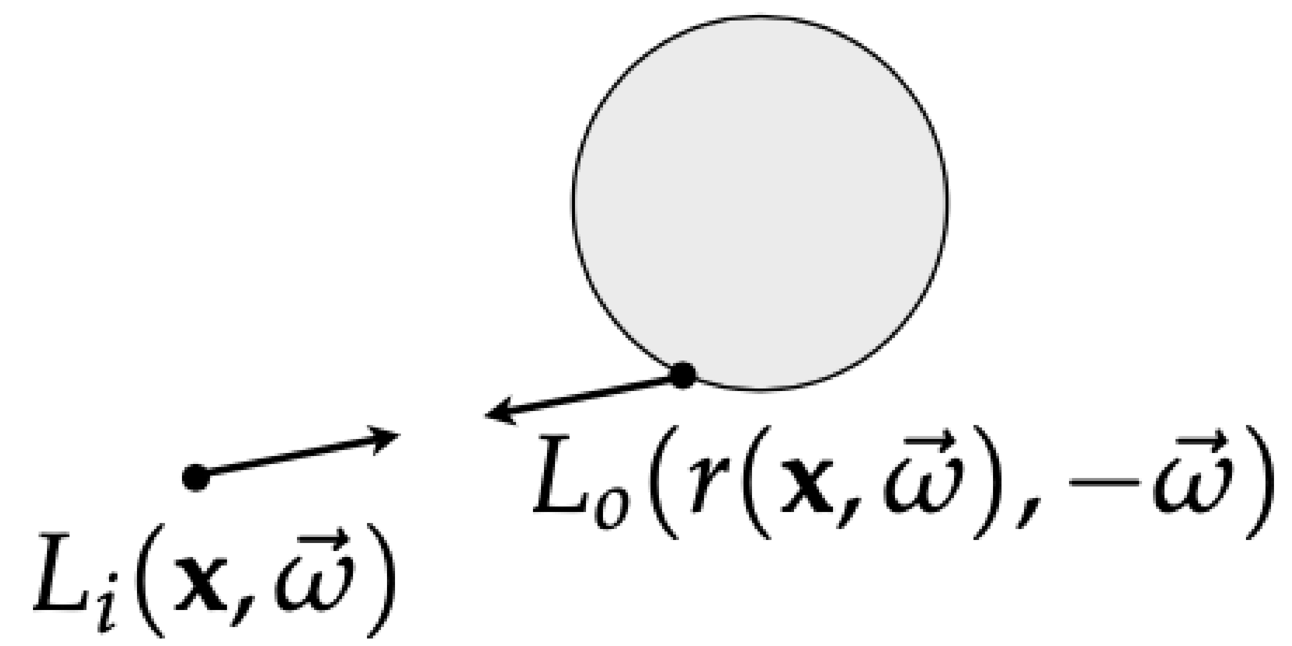
$$L_o(x,\omega_o) = L_e(x,\omega_o) + \int_{\mathcal{H}^2} f_r(x,\omega_0,\omega_i) L_i(x,\omega_i) |\cos\theta_i| d\omega_i$$

Outgoing emitted reflected

Rendering Equation: Light Transport

In vaccum, radiance is constant along rays

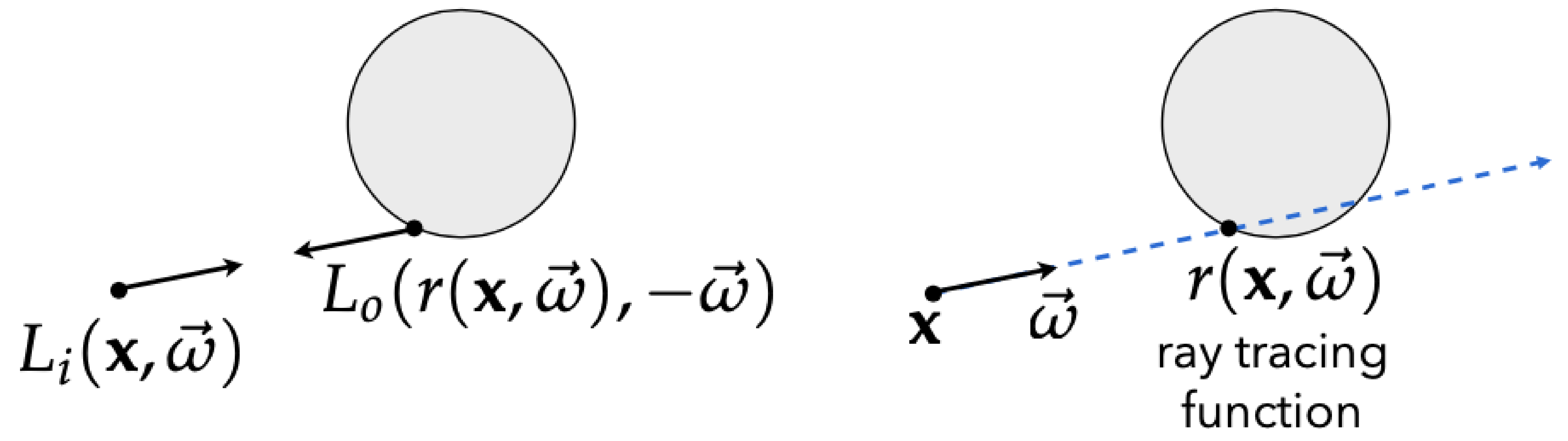
We can relate out-going radiance to the incoming radiance



Rendering Equation: Light Transport

In vaccum, radiance is constant along rays

We can relate out-going radiance to the incoming radiance



Rendering Equation

$$L_o(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L_i(x,\omega) |\cos\theta'| d\omega'$$

Rendering Equation

ray tracing function

$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega), -\omega') |\cos\theta'| d\omega'$$

Only outgoing radiance on both sides

- we drop the "o" subscript

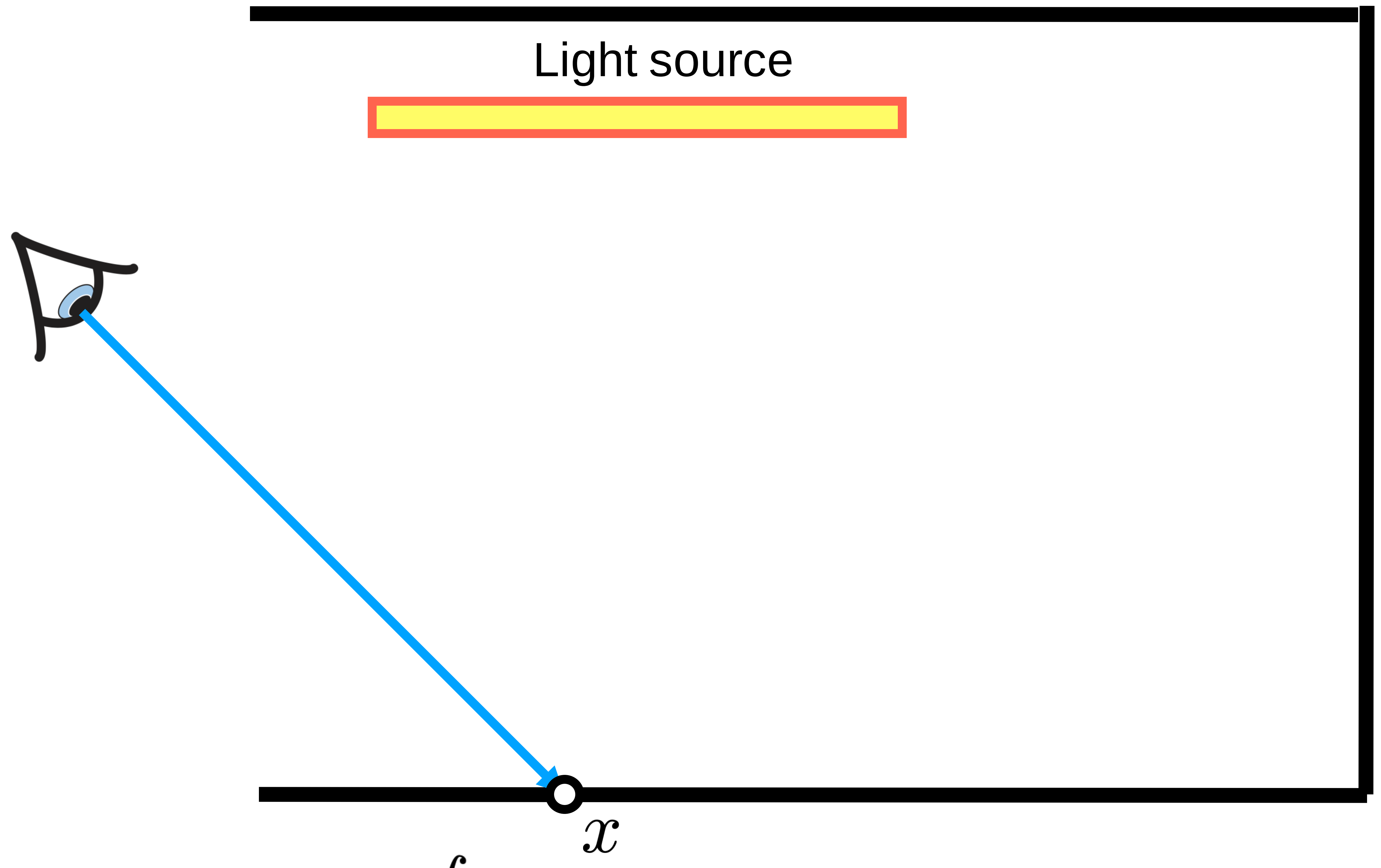
- Becomes Fredholm equation of the second kind (recursive)

Rendering Equation

$$L(x,\vec{\omega}) = L_e(x,\vec{\omega}) + \int_{\mathcal{H}^2} f(x,\vec{\omega}',\vec{\omega}) L(r(x,\vec{\omega}'), -\vec{\omega}') |\cos\theta'| d\vec{\omega}'$$

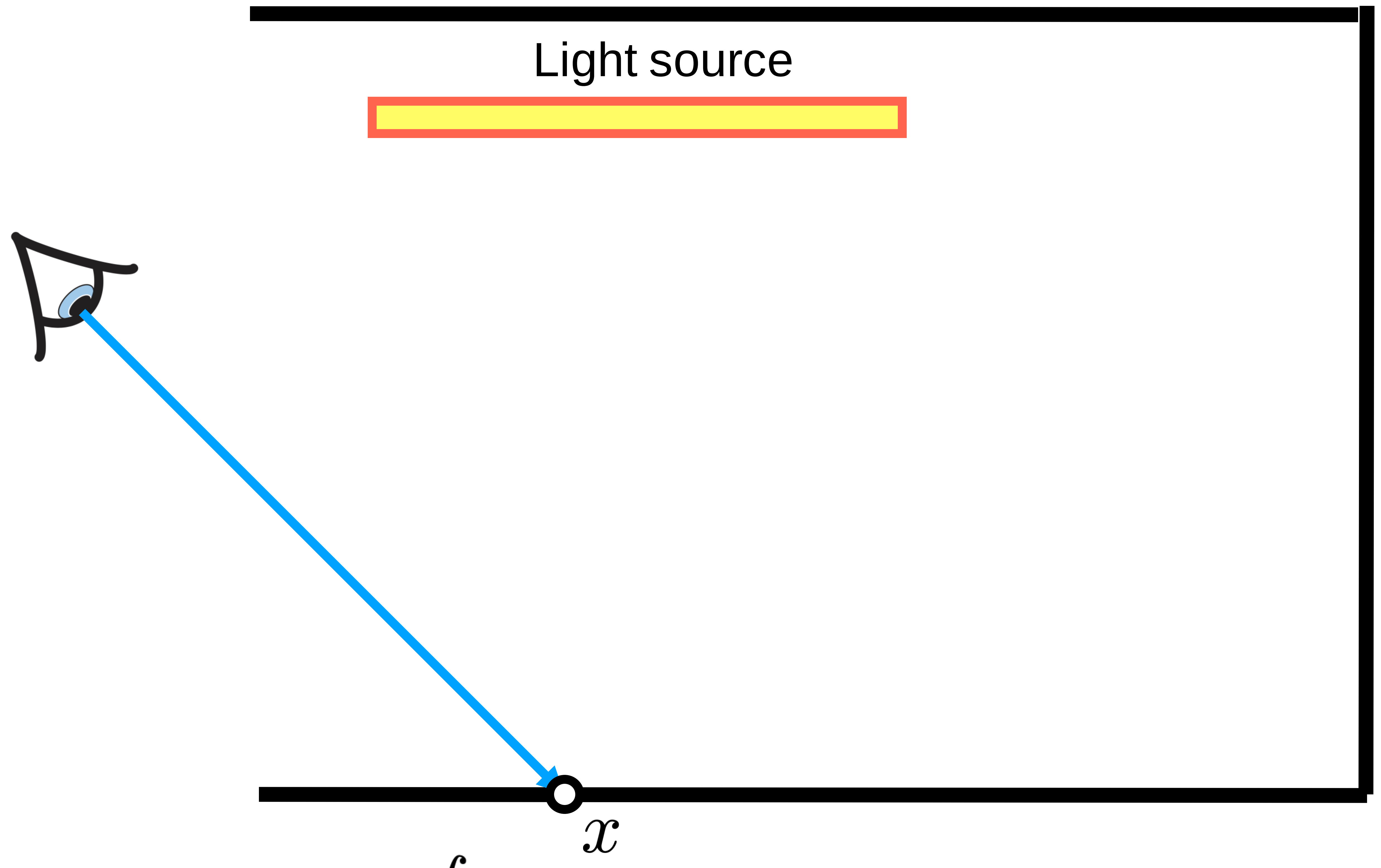
Path Tracing

Path Tracing



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'), -\omega') \cos\theta' d\omega'$$

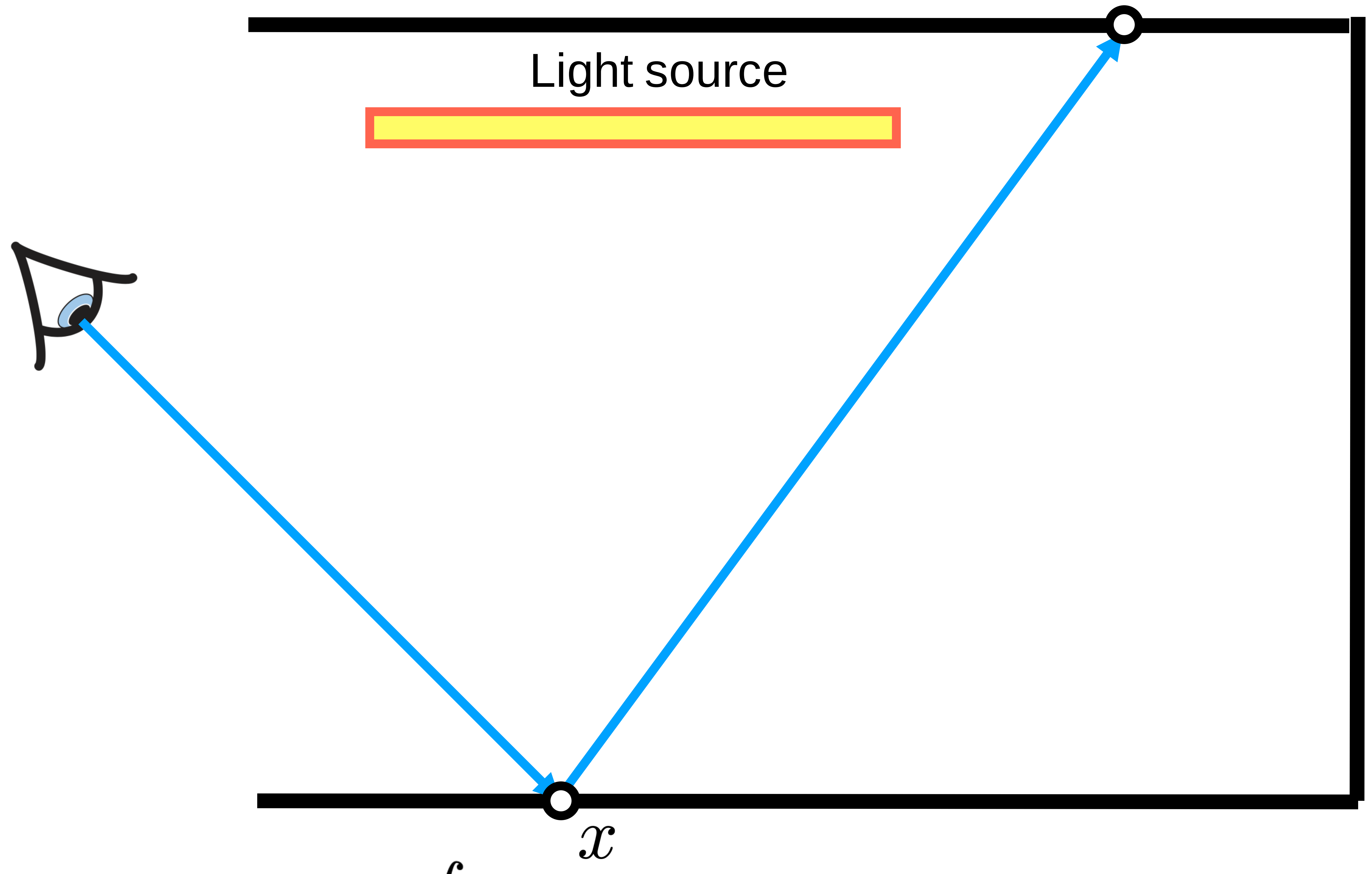
Path Tracing



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

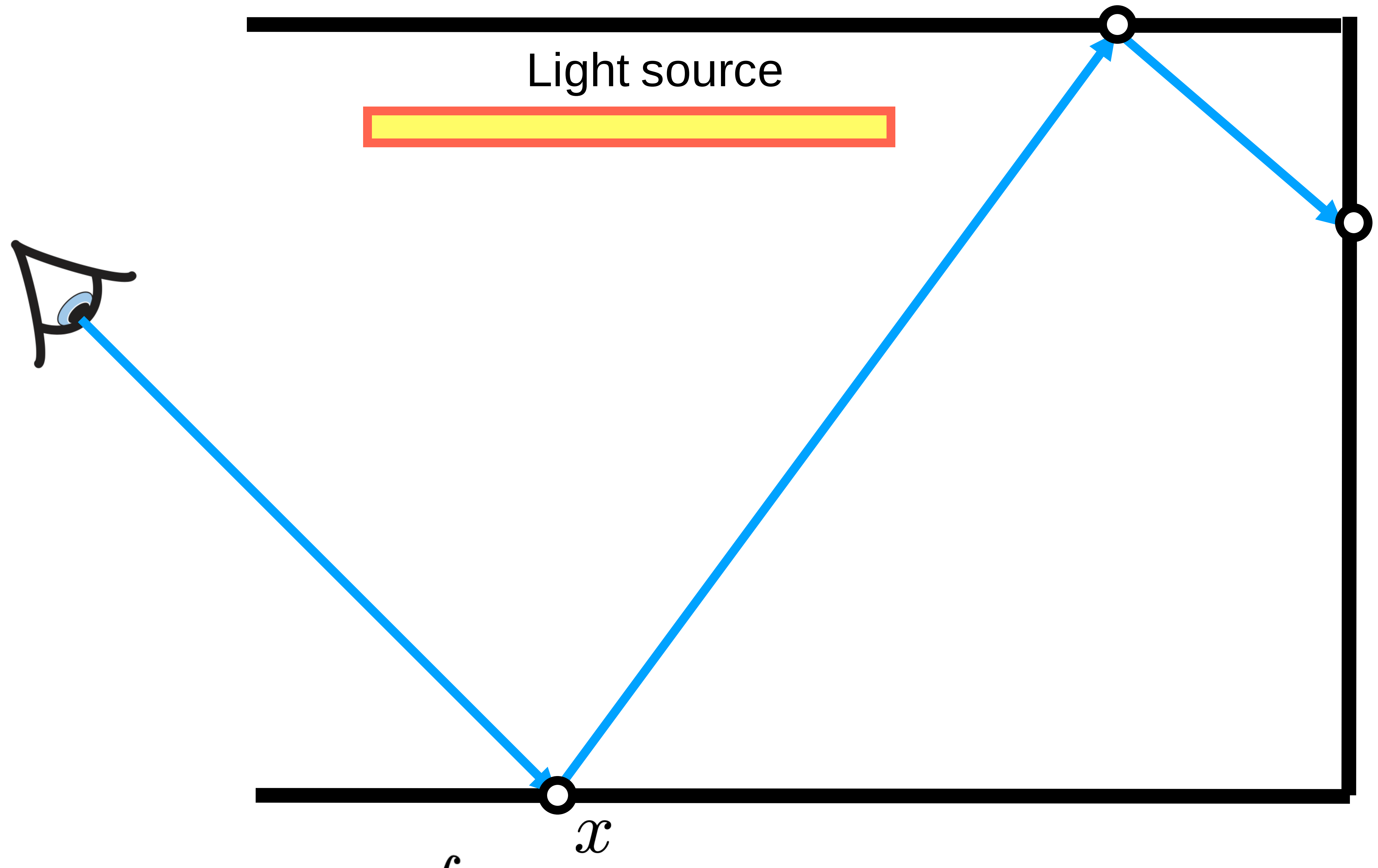
Path Tracing



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'), -\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'), -\omega') \cos\theta'}{p(\omega')}$$

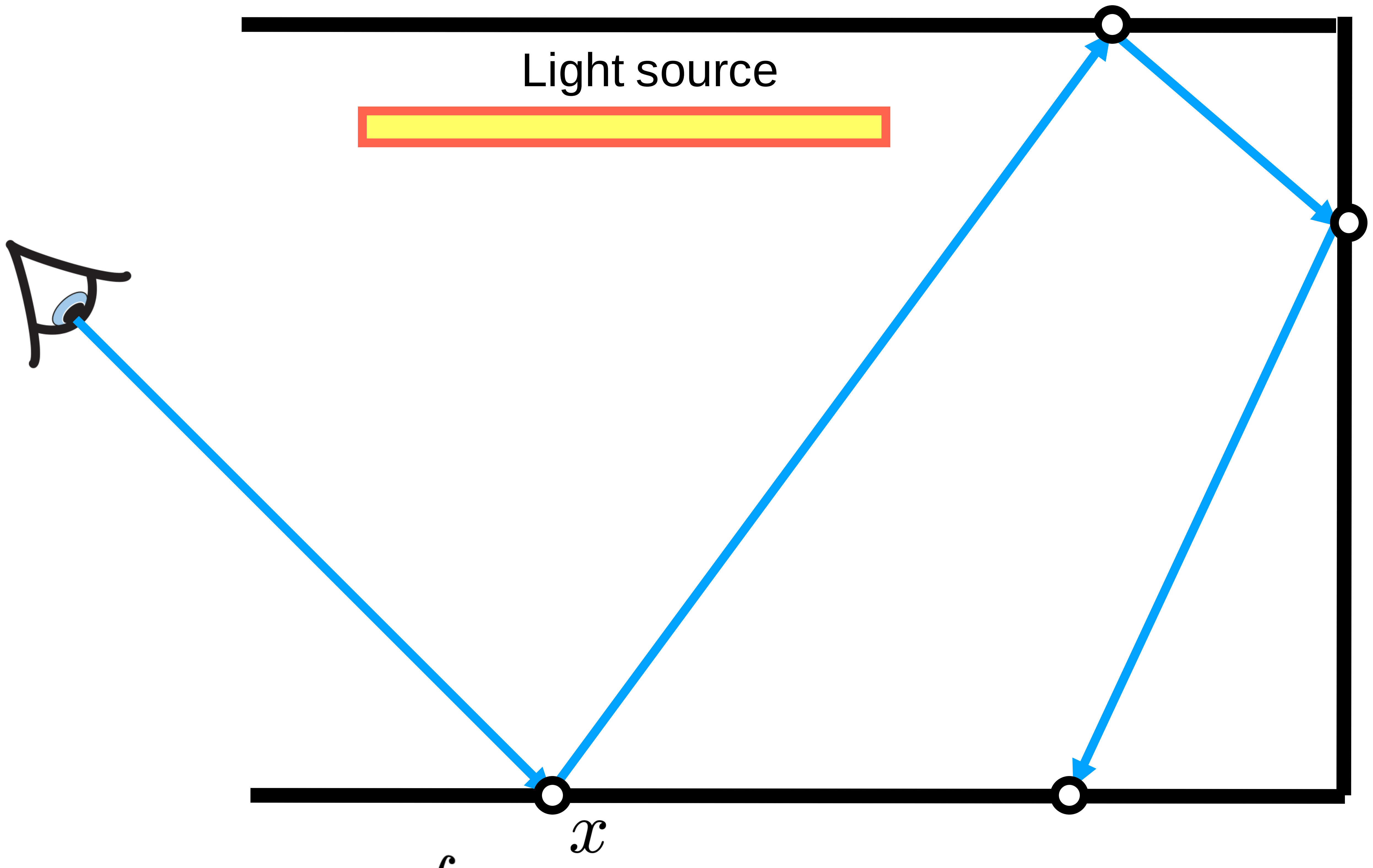
Path Tracing



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

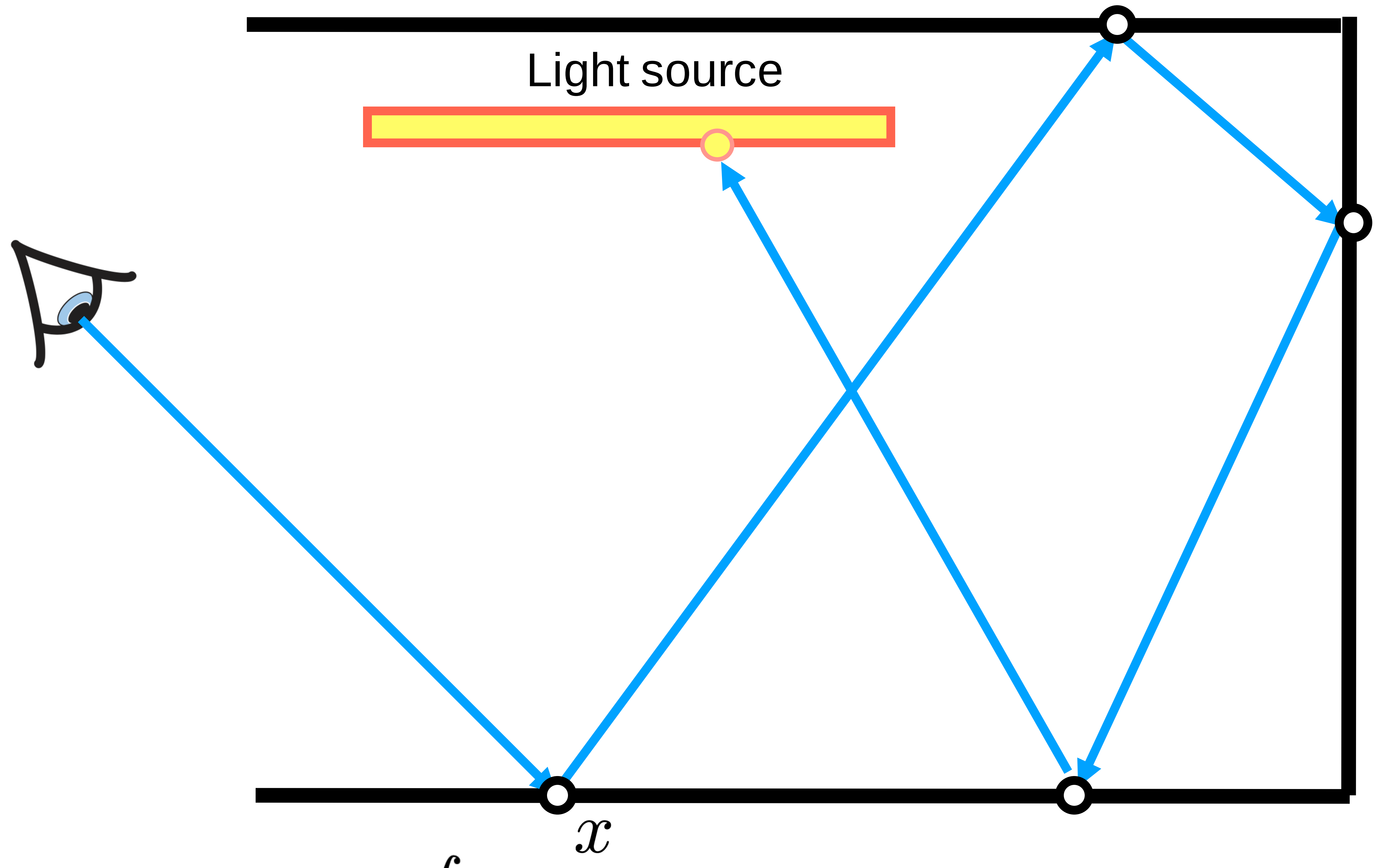
Path Tracing



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega)L(r(x,\omega'),-\omega')\cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega)L(r(x,\omega'),-\omega')\cos\theta'}{p(\omega')}$$

Path Tracing



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

Path Tracing Algorithm

$$L_o(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o)$$

```
Color color(Point x, Direction ω, int moreBounces):

    if not moreBounces:
        return Le(x,-ω)

    // sample recursive integral
    ω' = sample from BRDF
    return Le(x,-ω) + BRDF * color(trace(x, ω'), moreBounces-1) * dot(n, ω') / pdf(ω')
```

Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces

n,

Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces

Let's estimate direct illumination separately from indirect illumination, then add the two

Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces

Let's estimate direct illumination separately from indirect illumination, then add the two

- i.e., shoot shadow rays (direct) and gather rays (indirect)

- be careful not to double count!

Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces

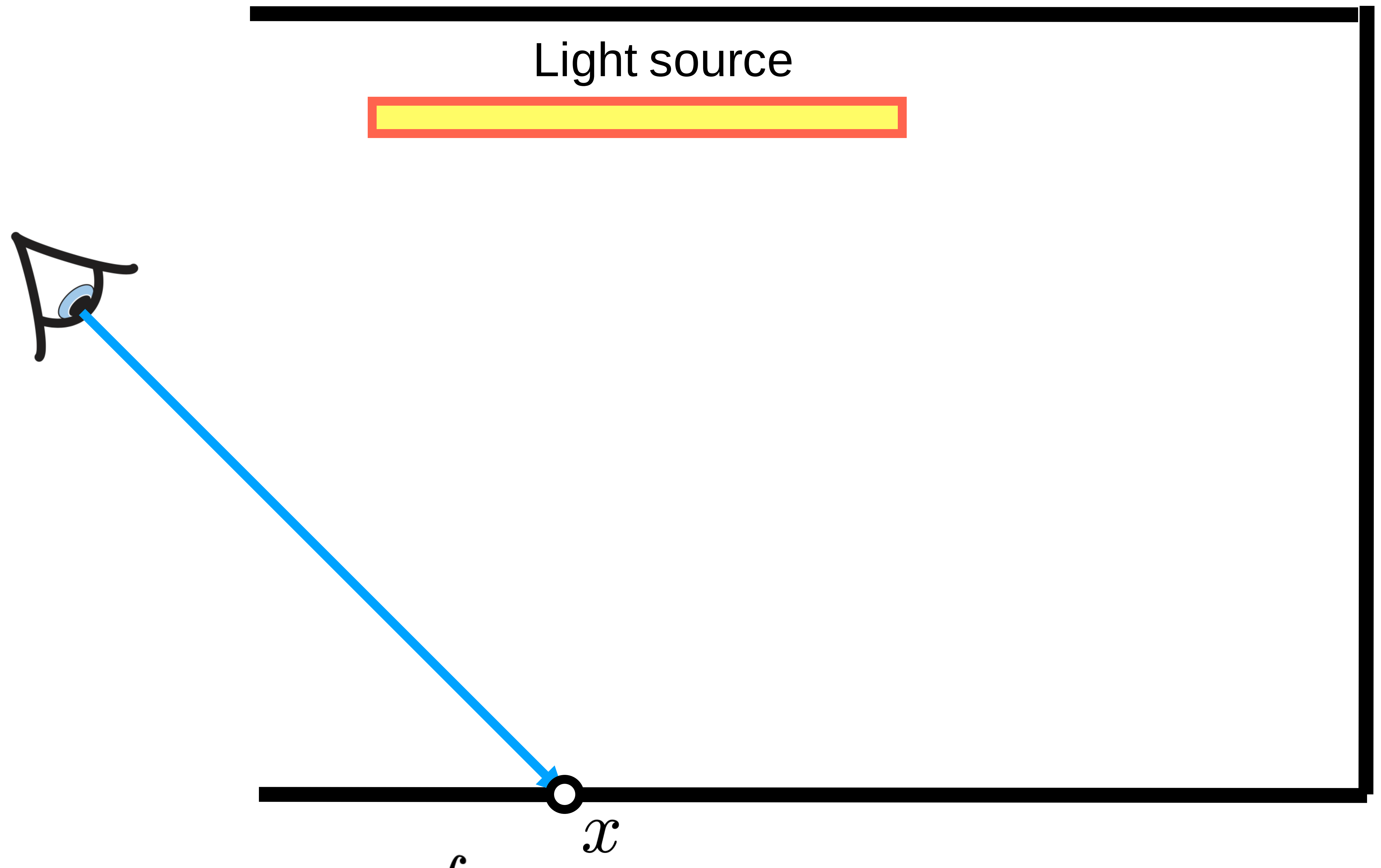
Let's estimate direct illumination separately from indirect illumination, then add the two

- i.e., shoot shadow rays (direct) and gather rays (indirect)

- be careful not to double count!

Also known as Next Event Estimation (NEE)

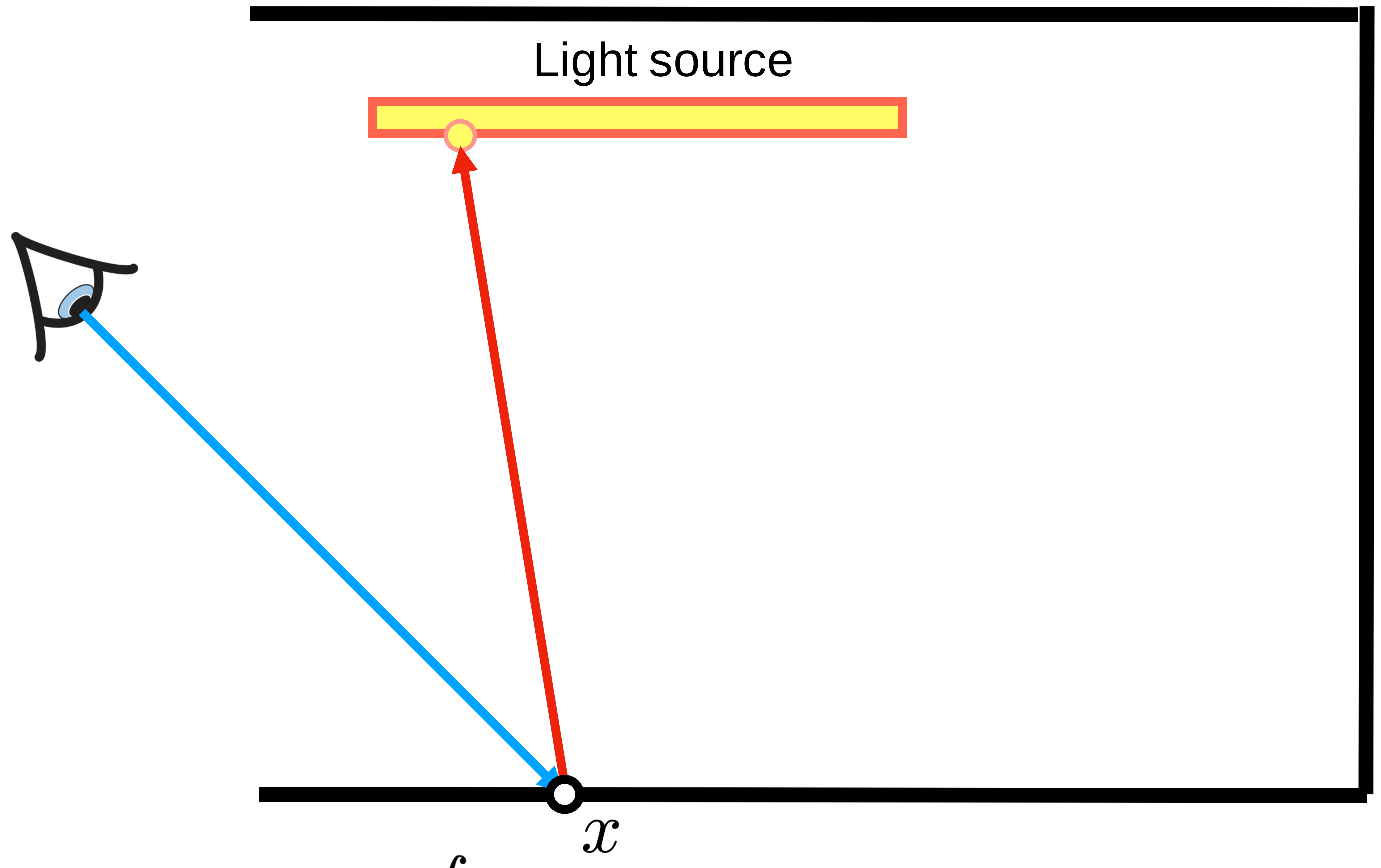
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

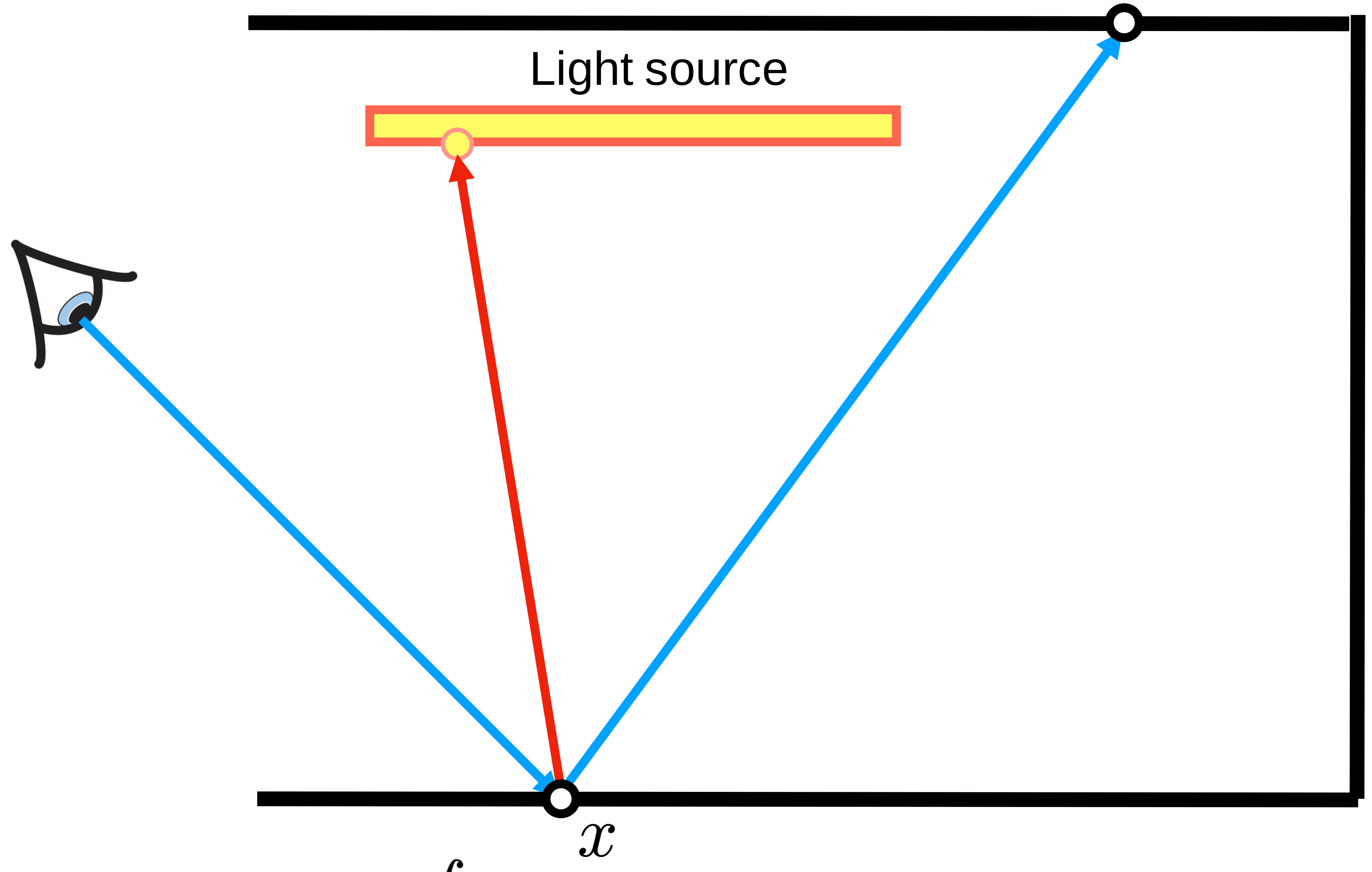
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega)L(r(x,\omega'),-\omega')\cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega)L(r(x,\omega'),-\omega')\cos\theta'}{p(\omega')}$$

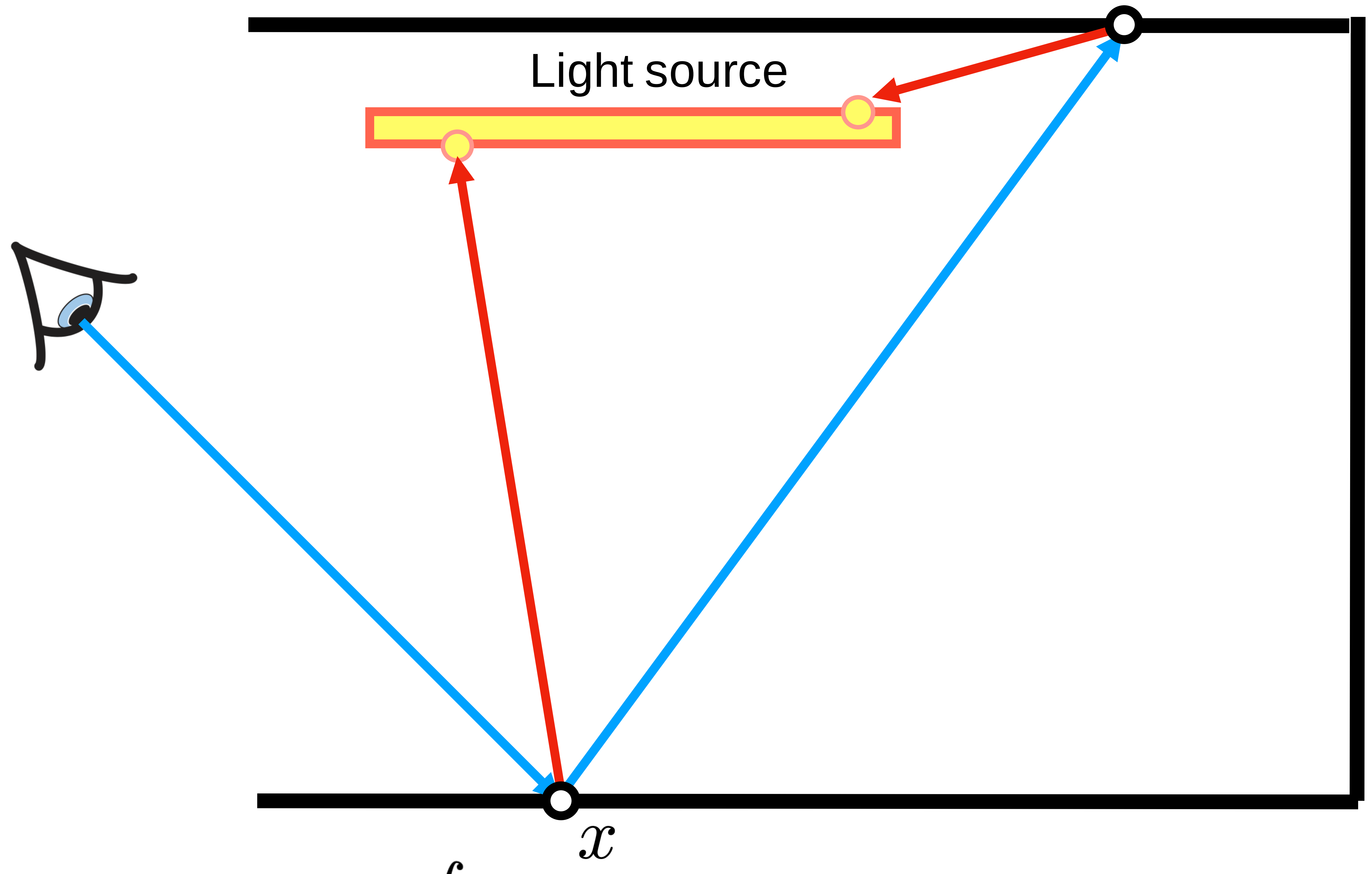
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

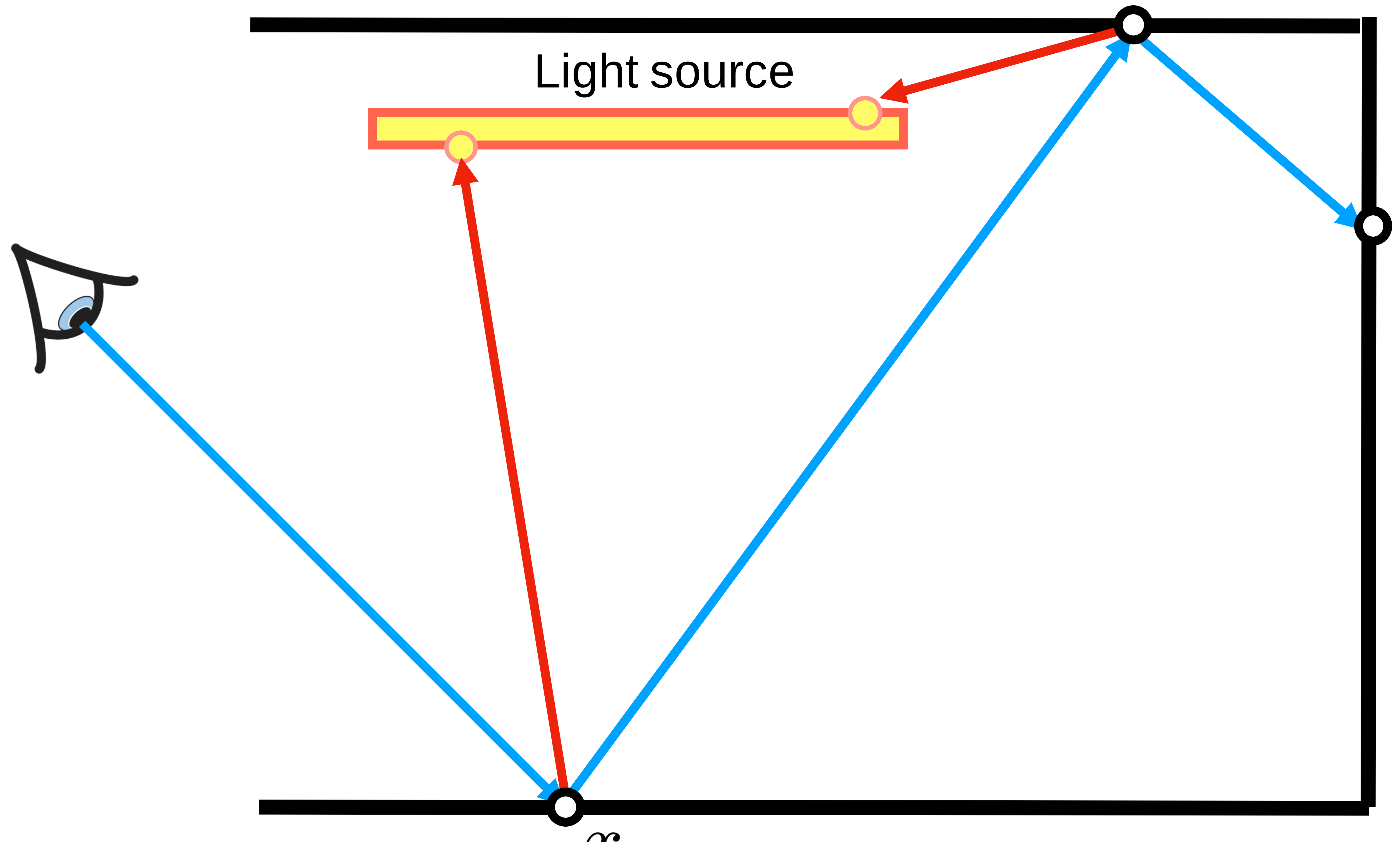
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

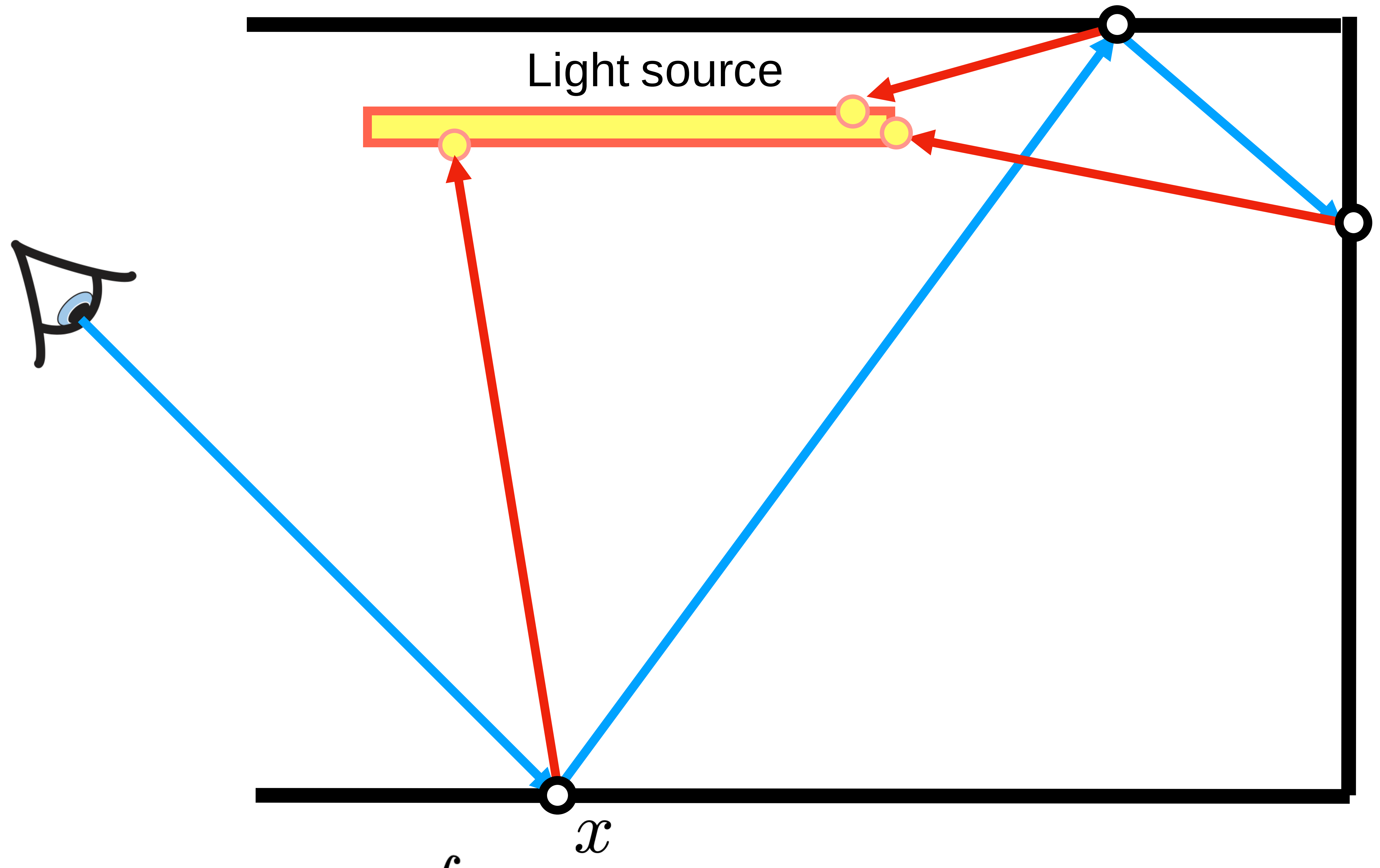
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

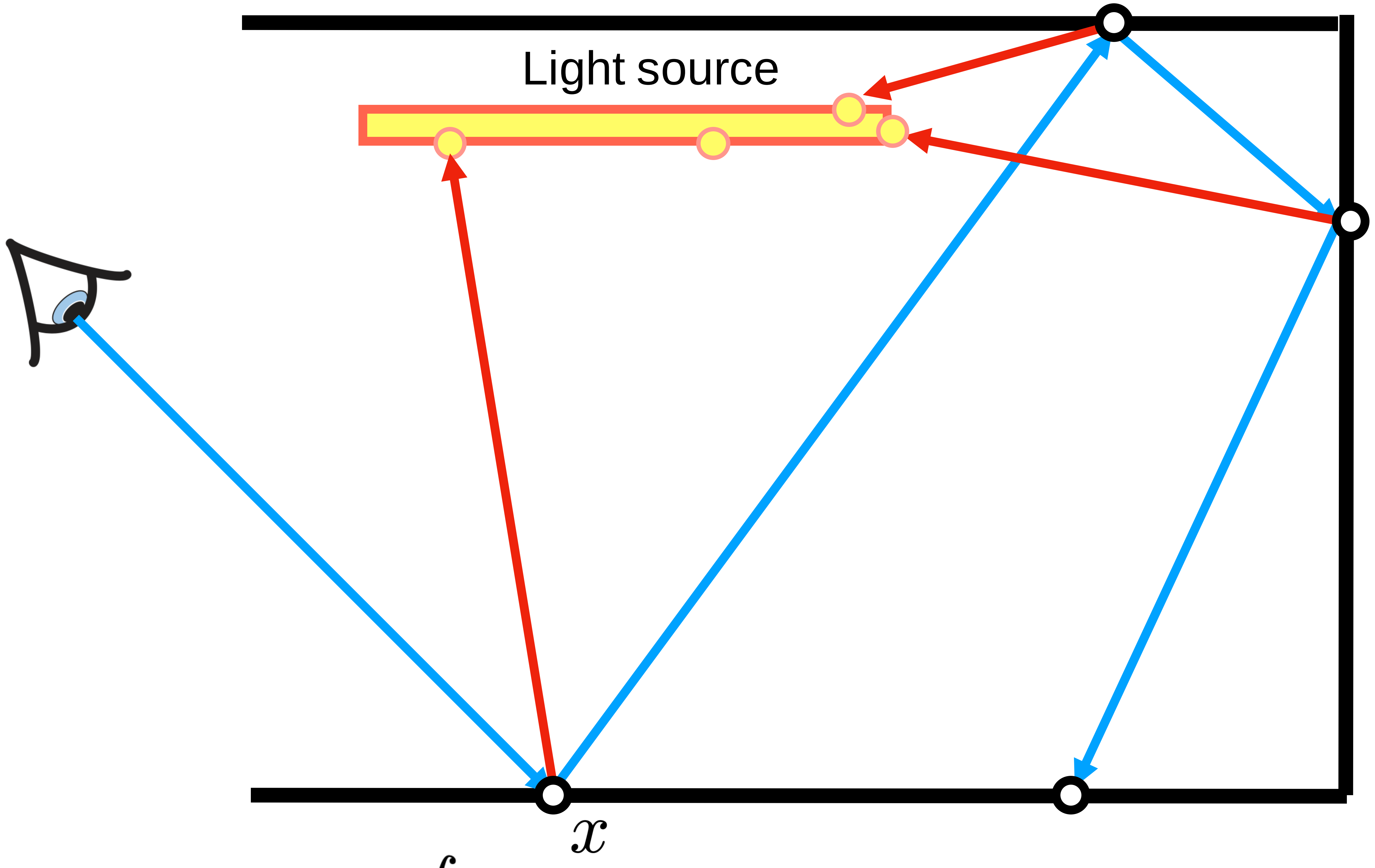
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

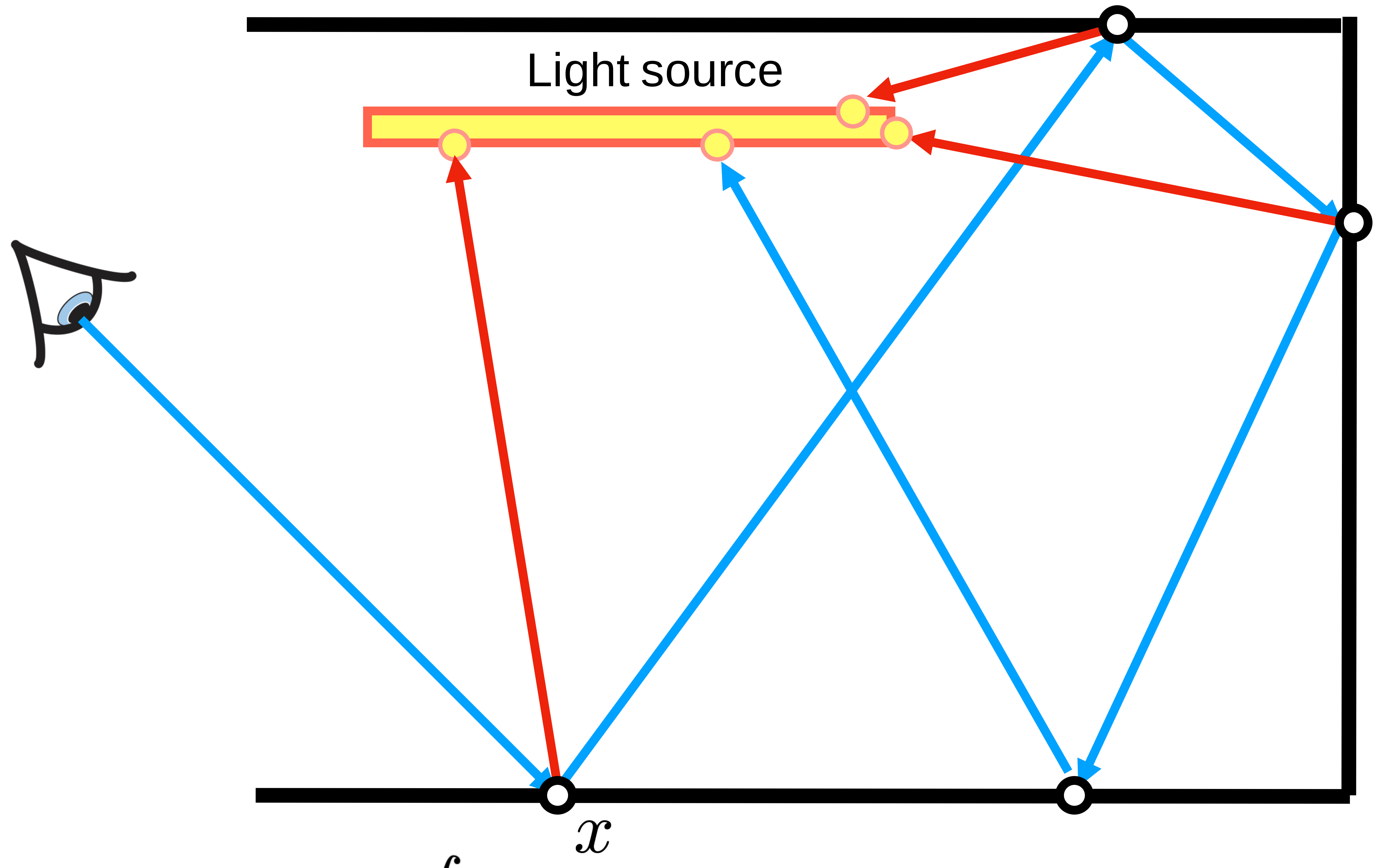
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

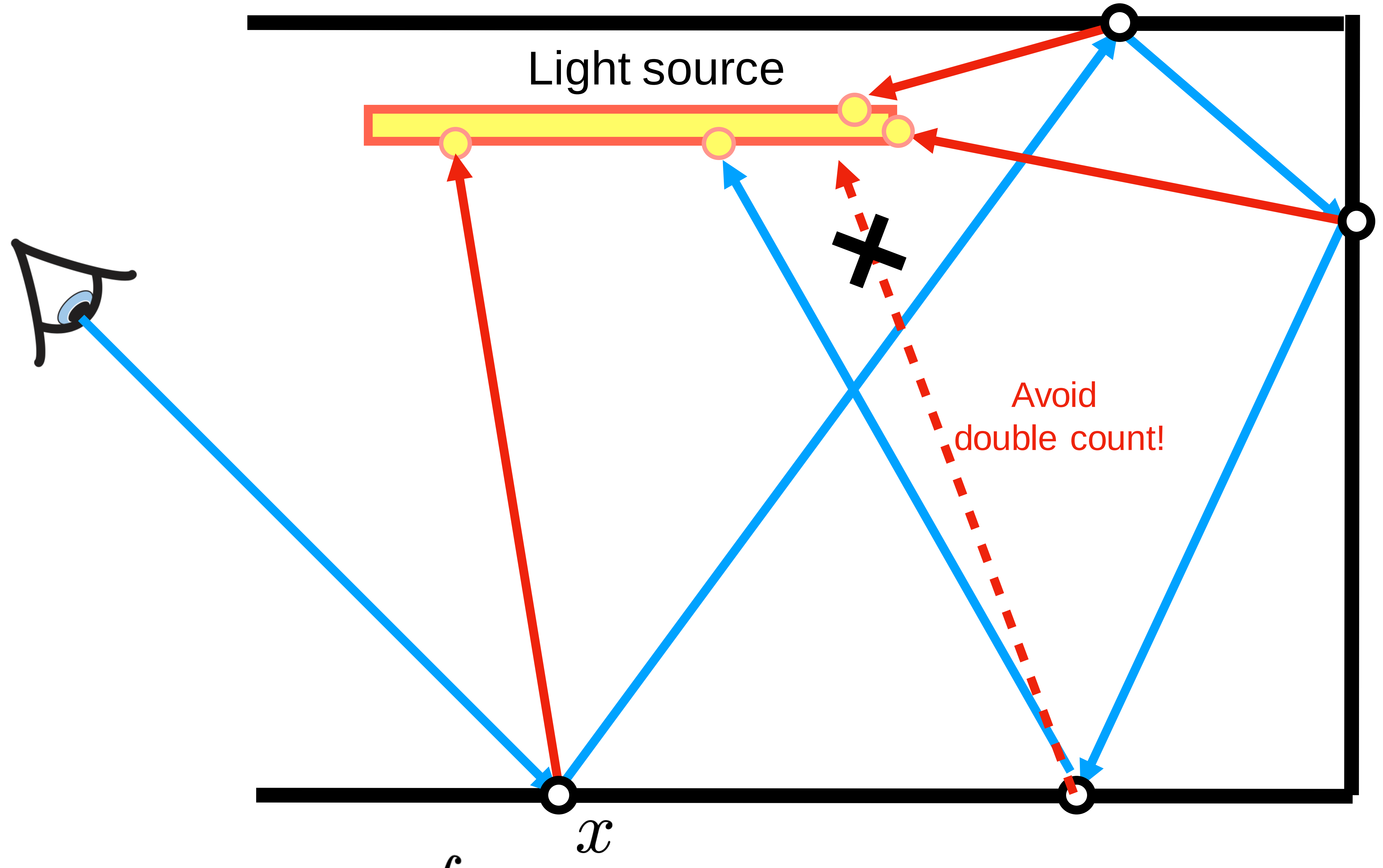
Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

Path Tracing Algorithm with NEE



$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{H}^2} f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta' d\omega'$$

$$\approx L_e(x,\omega) + \frac{f(x,\omega',\omega) L(r(x,\omega'),-\omega') \cos\theta'}{p(\omega')}$$

Path Tracing Algorithm with NEE

$$L(x,\omega) = L_e(x,\omega) + L_{dir}(x,\omega) + L_{ind}(x,\omega)$$

```
Color color(Point x, Direction ω, int moreBounces):

    if not moreBounces:
        return Le;

    // next-event estimation: compute Ldir by sampling the light
    ω1 = sample from light
    Ldir = BRDF * color(trace(x, ω1), 0) * dot(n, ω1) / pdf(ω1)

    // compute Lind by sampling the BSDF
    ω2 = sample from BSDF;
    Lind = BSDF * color(trace(x, ω2), moreBounces-1) * dot(n, ω2) / pdf(ω2)

    return Le + Ldir + Lind
```

Path Tracing Algorithm with NEE

$$L(x, \omega) = L_e(x, \omega) + L_{dir}(x, \omega) + L_{ind}(x, \omega)$$

```
Color color(Point x, Direction ω, int moreBounces):

    if not moreBounces:
        return Le;

    // next-event estimation: compute Ldir by sampling the light
    ω1 = sample from light
    Ldir = BRDF * color(trace(x, ω1), 0) * dot(n, ω1) / pdf(ω1)

    // compute Lind by sampling the BSDF
    ω2 = sample from BSDF;
    Lind = BSDF * color(trace(x, ω2), moreBounces-1) * dot(n, ω2) / pdf(ω2)

    return Le + Ldir + Lind
```

double counting!

Path Tracing Algorithm with NEE

$$L(x,\omega) = L_e(x,\omega) + L_{dir}(x,\omega) + L_{ind}(x,\omega)$$

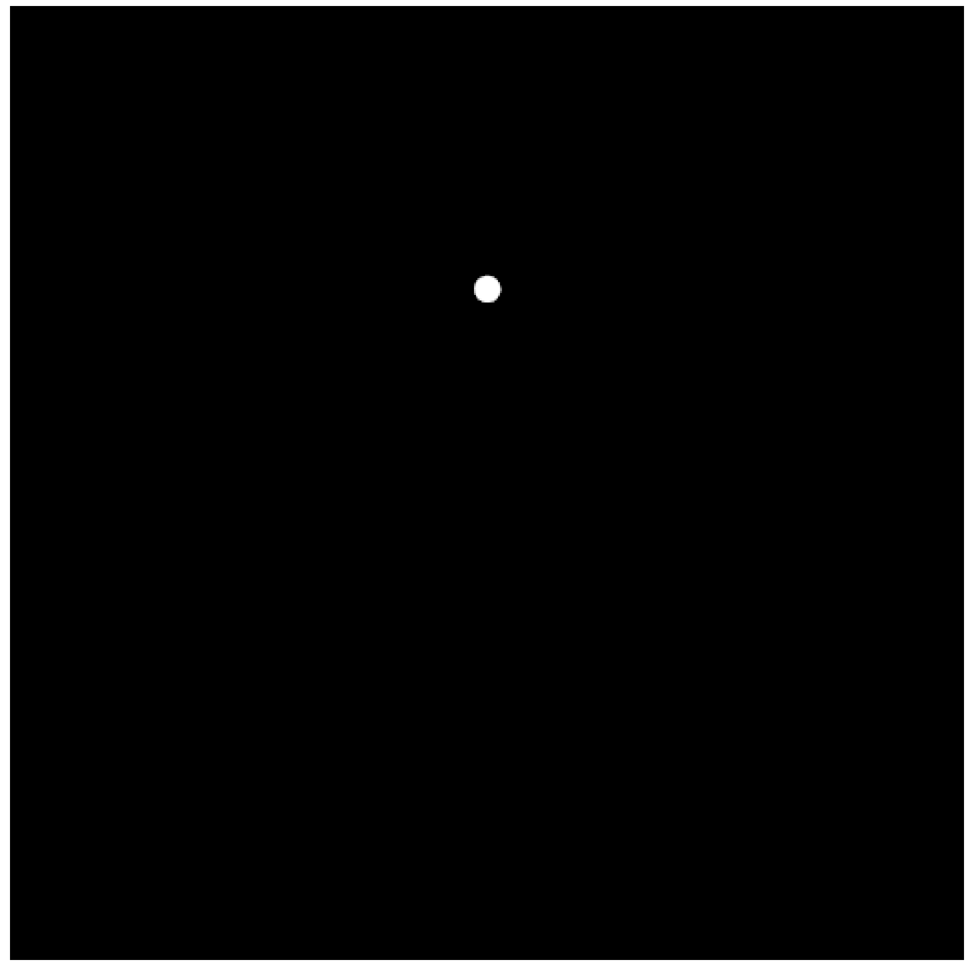
```
Color color(Point x, Direction ω, int moreBounces):

    if not moreBounces:
        return Le;

    // next-event estimation: compute Ldir by sampling the light
    ω1 = sample from light
    Ldir = BRDF * color(trace(x, ω1), 0) * dot(n, ω1) / pdf(ω1)

    // compute Lind by sampling the BSDF
    ω2 = sample from BSDF;
    Lind = BSDF * color(trace(x, ω2), moreBounces-1) * dot(n, ω2) / pdf(ω2)

    return Le + Ldir + Lind
```

Path Tracing Algorithm with NEE

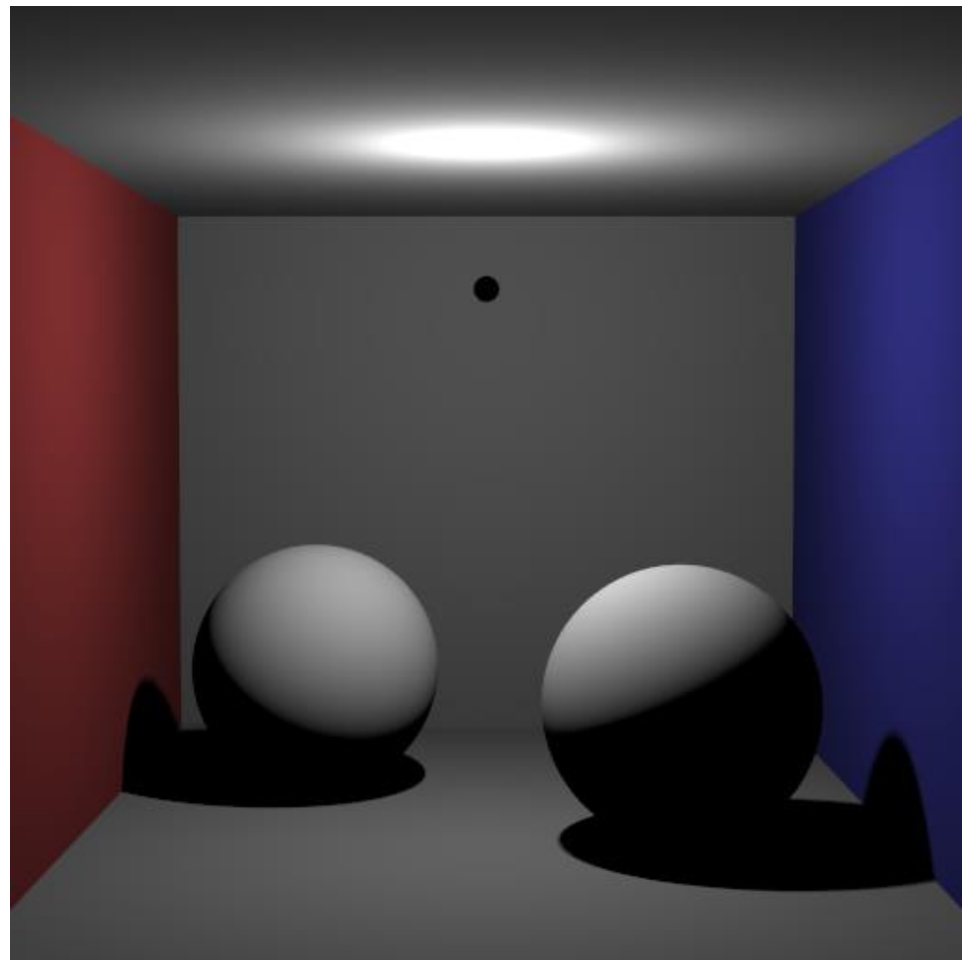
$$L(x,\omega) = L_e(x,\omega) + L_{dir}(x,\omega) + L_{ind}(x,\omega)$$

```
Color color(Point x, Direction ω, int moreBounces, bool includeL_e):

    L_e = includeL_e ? L_e(x,-ω) : black

    if not moreBounces:
        return L_e

    // next-event estimation: compute L_dir by sampling the light
    ω_1 = sample from light
    L_dir = BRDF * color(trace(x, ω_1), 0, true) * dot(n, ω_1) / pdf(ω_1)

    // compute L_ind by sampling the BSDF
    ω_2 = sample from BSDF
    L_ind = BSDF * color(trace(x, ω_2), moreBounces-1, false) * dot(n, ω_2) / pdf(ω_2)

    return L_e + L_dir + L_ind
```

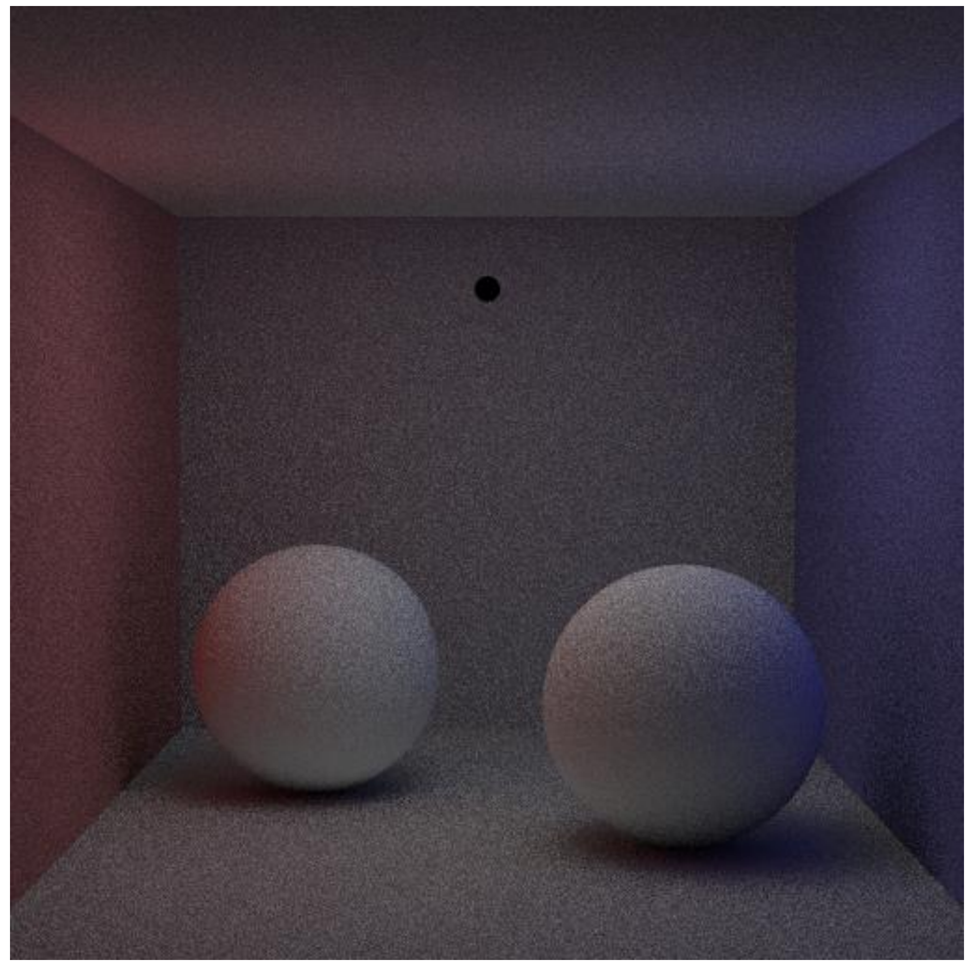
Path-wise Visualization



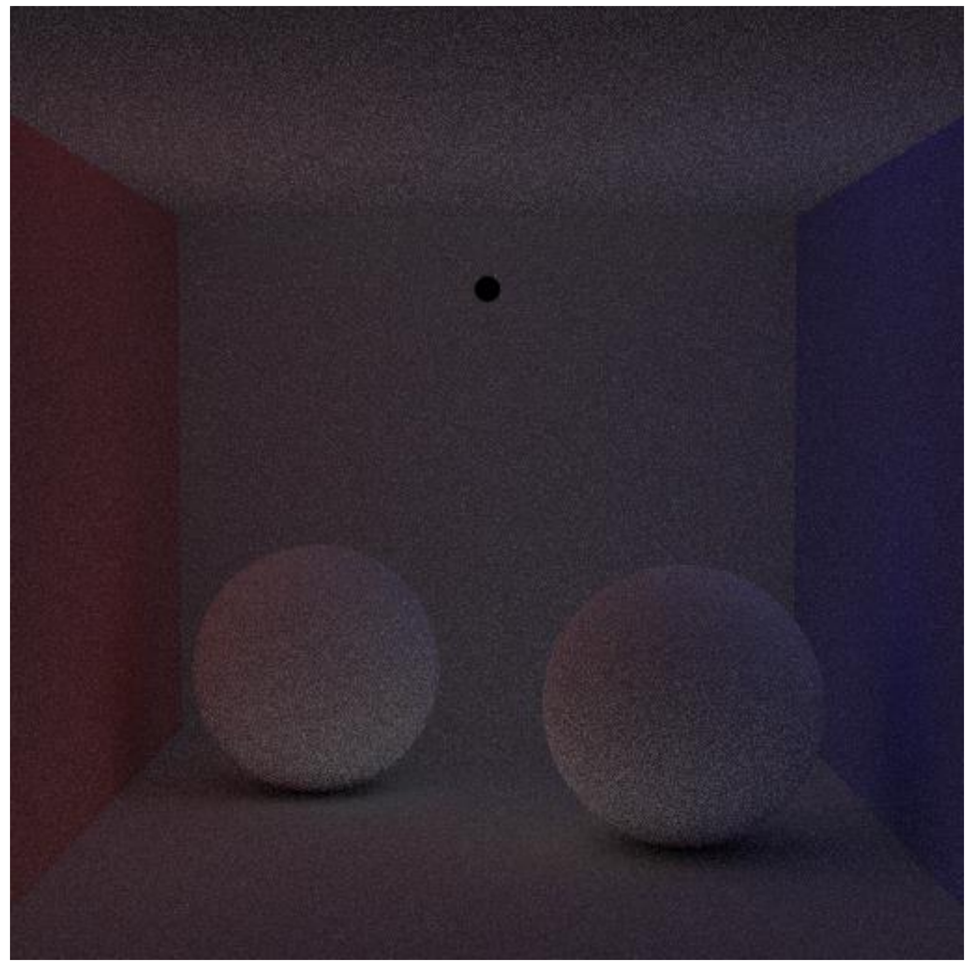
Path: 0



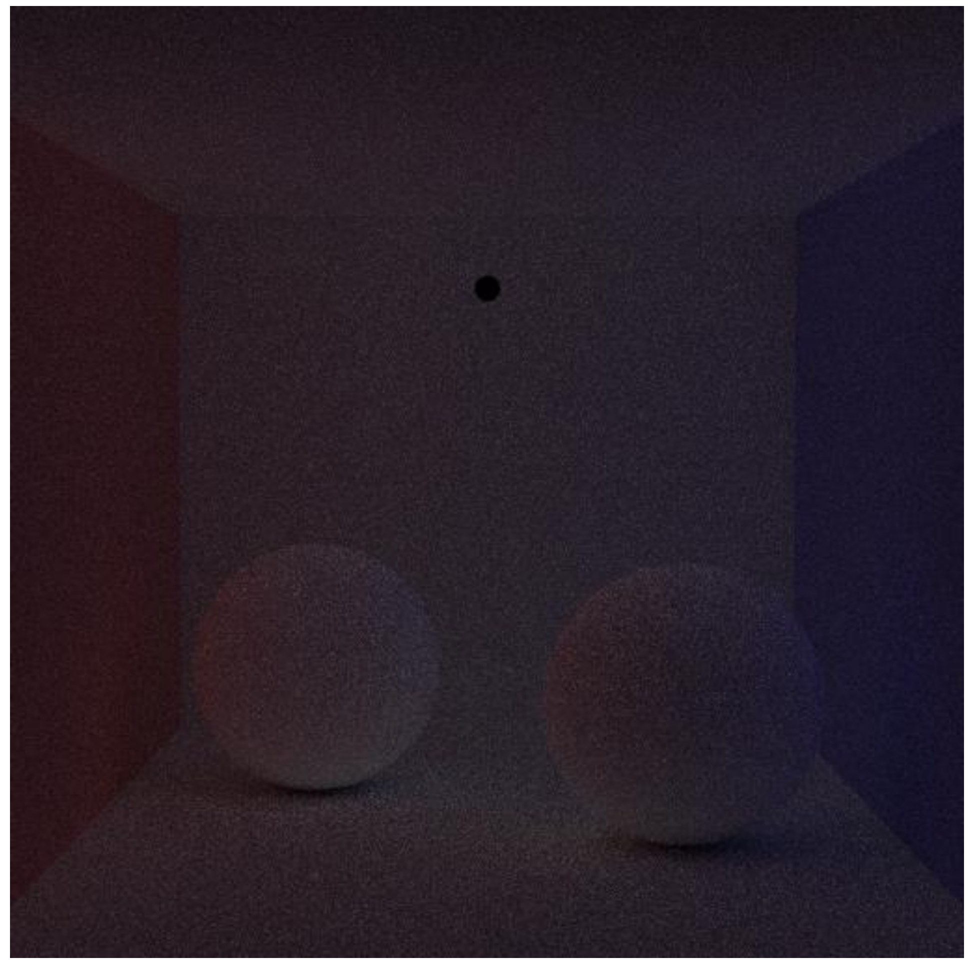
Path: 1



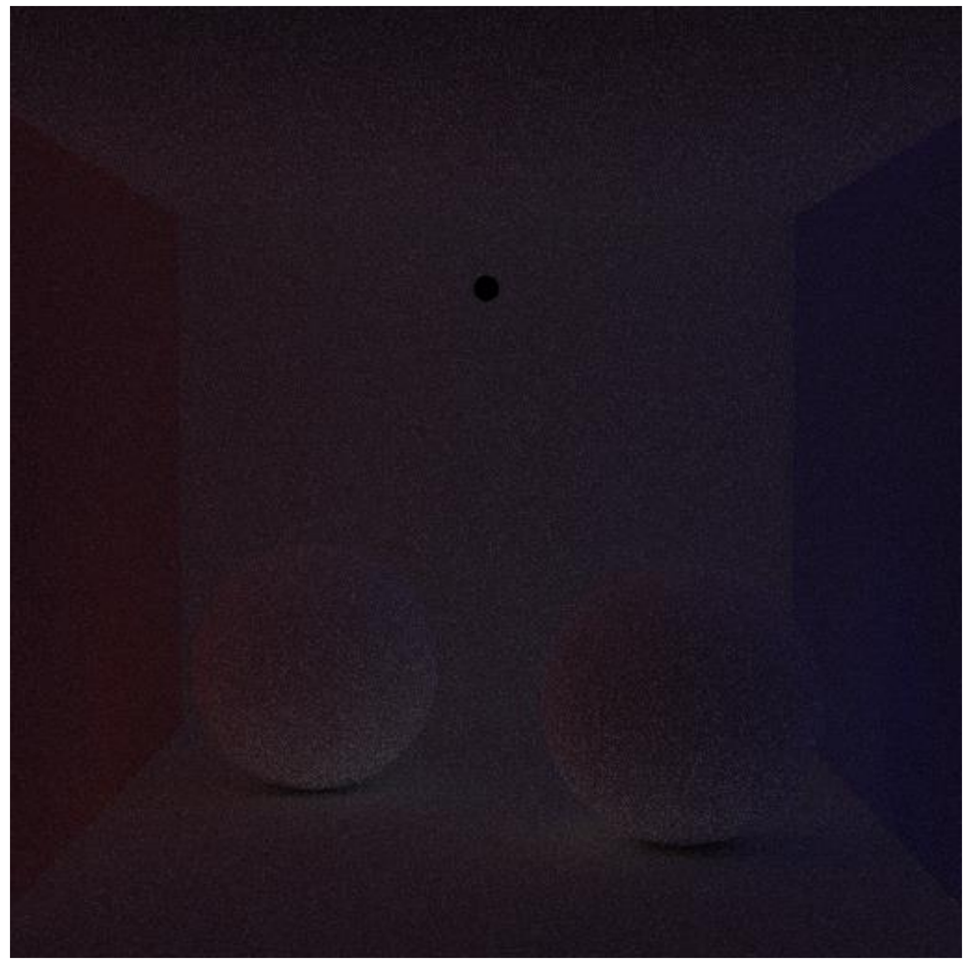
Path: 2



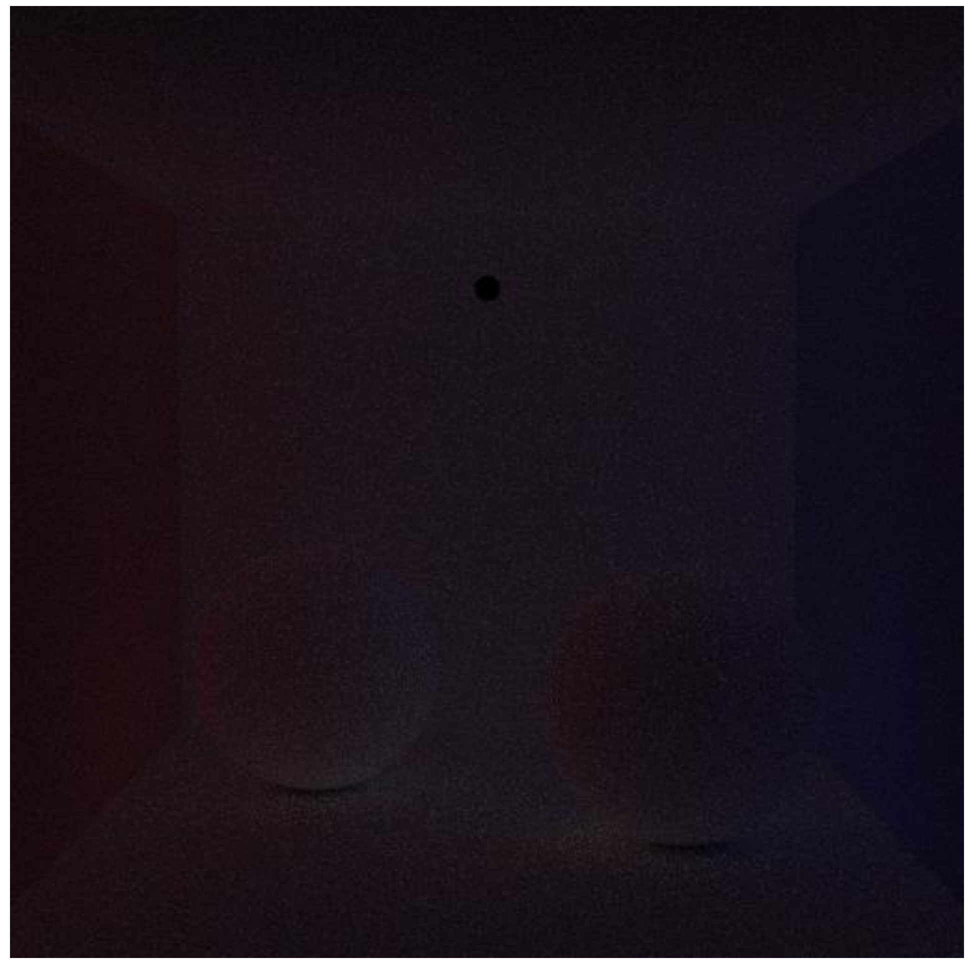
Path: 3



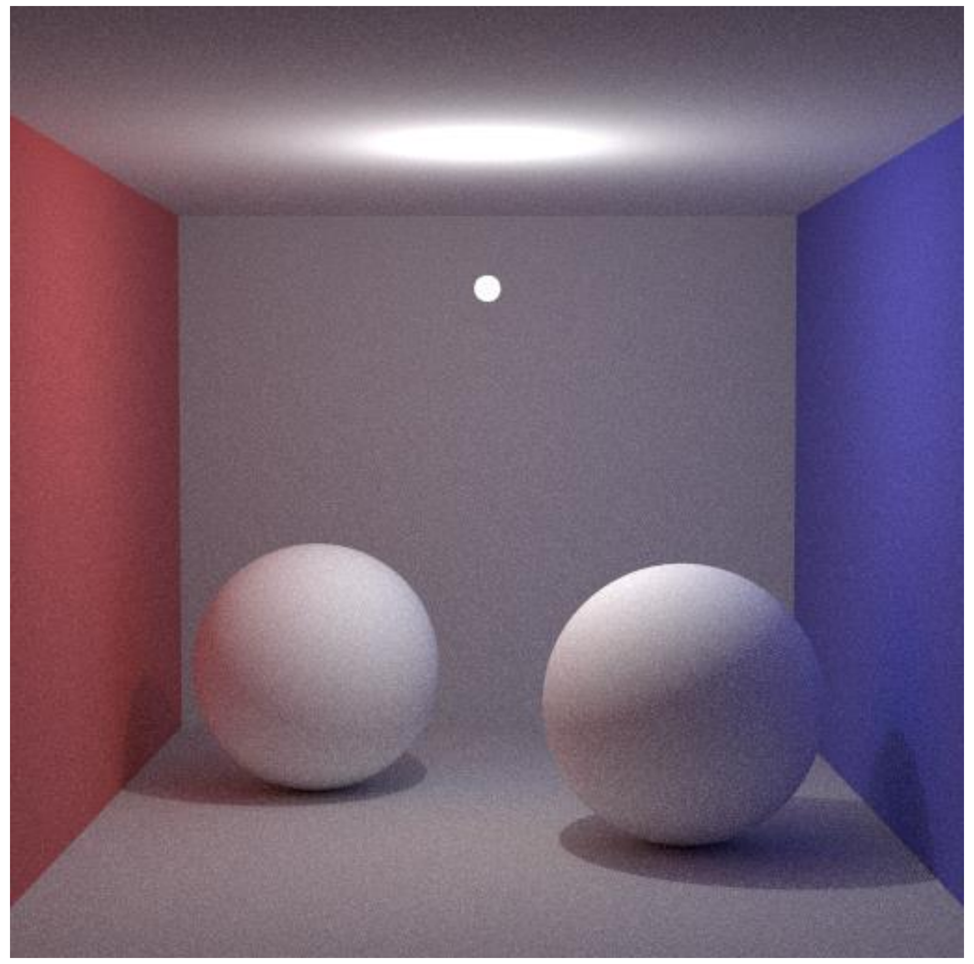
Path: 4



Path: 5



Path: 6



All Paths added

When we do stop recursion?

Truncating at some fixed depth introducing **bias**

Solution: Russian roulette

Russian Roulette

Probabilisticaly terminate the recursion

X

New estimator: evaluate original estimator X with probability P (but reweighted), otherwise return zero:

$$X_{rr} = \begin{cases} \frac{X}{P} & \xi < P \\ 0 & \text{otherwise} \end{cases}$$

Russian Roulette

This will increase variance!

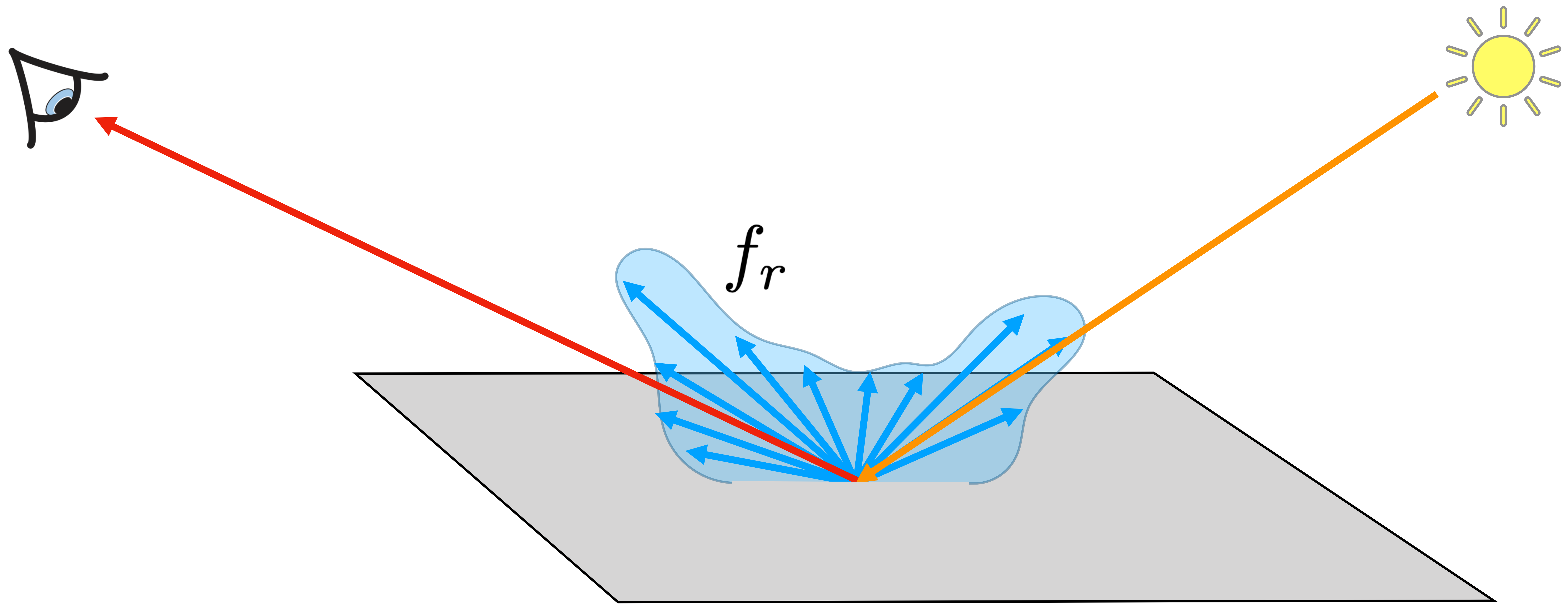
- but it will improve efficiency if P is chosen so that the samples that are expensive, but are likely to make small contribution, are skipped

Source: Midjourney

Microfacet BSDFs

BRDF

Bidirectional Reflectance Distribution Function



BRDF Properties

Real/Physically plausible BRDFs obey:

- Energy conservation: $$\int_{\mathcal{H}^2} f_r(\mathrm{x}, \vec{\omega}_i, \vec{\omega}_r) \cos\theta_i d\vec{\omega}_i \leq 1, \quad \forall\, \vec{\omega}_r$$

BRDF Properties

Real/Physically plausible BRDFs obey:

- Energy conservation:

$$\int_{\mathcal{H}^{\in}} f_r(\mathrm{x}, \vec{\omega}_i, \vec{\omega}_r) \cos\theta_i d\vec{\omega}_i \leq 1, \quad \forall\, \vec{\omega}_r$$

- Helmholtz reciprocity:

$$f_r(\mathrm{x}, \vec{\omega}_i, \vec{\omega}_r) = f_r(\mathrm{x}, \vec{\omega}_r, \vec{\omega}_i)$$

$$f_r(\mathrm{x}, \vec{\omega}_i \leftrightarrow \vec{\omega}_r)$$

Conductors vs. Dielectrics

Dielectrics: Materials that does not conduct electricity,
— e.g., glass, mineral oil, water and air
— index of refraction

Conductors: Materials that conduct electricity, e.g. metal
— Complex index of refraction.

Conductors vs. Dielectrics

Copper

Iron

Gold

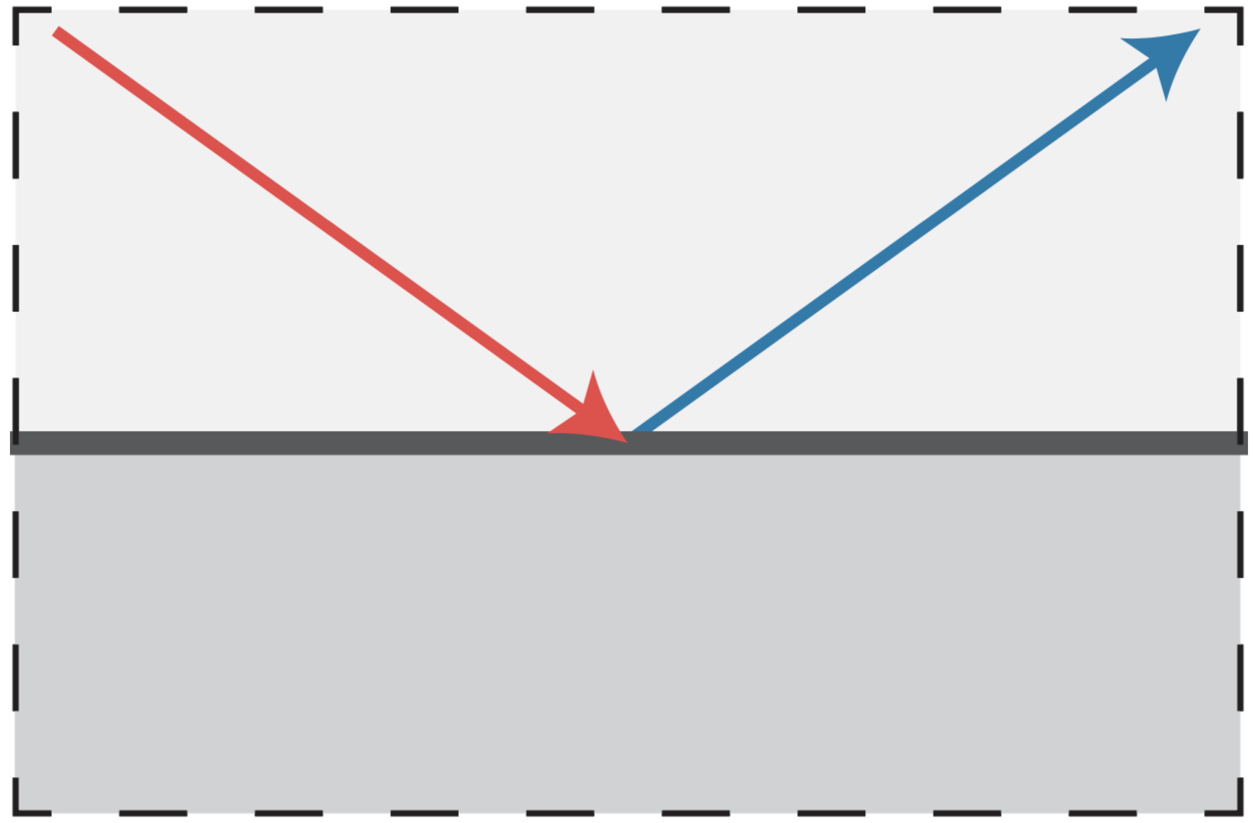
Glass

Crystal rocks

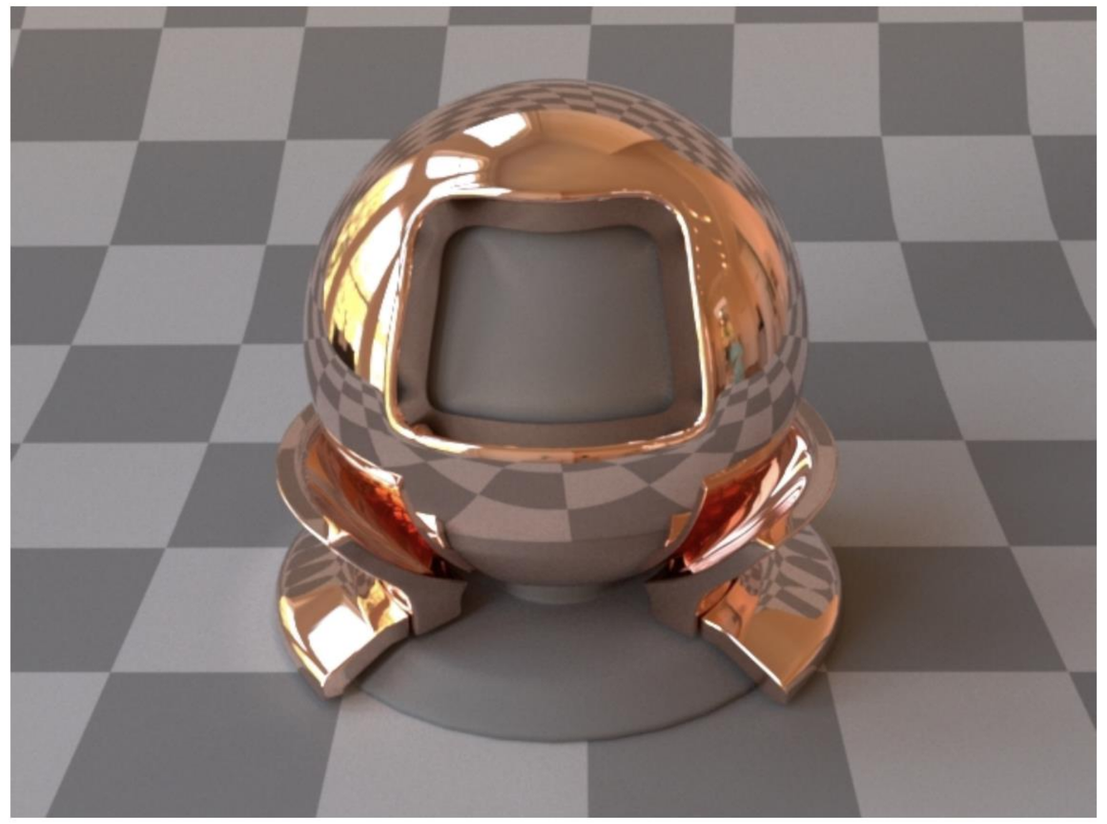
Mercury

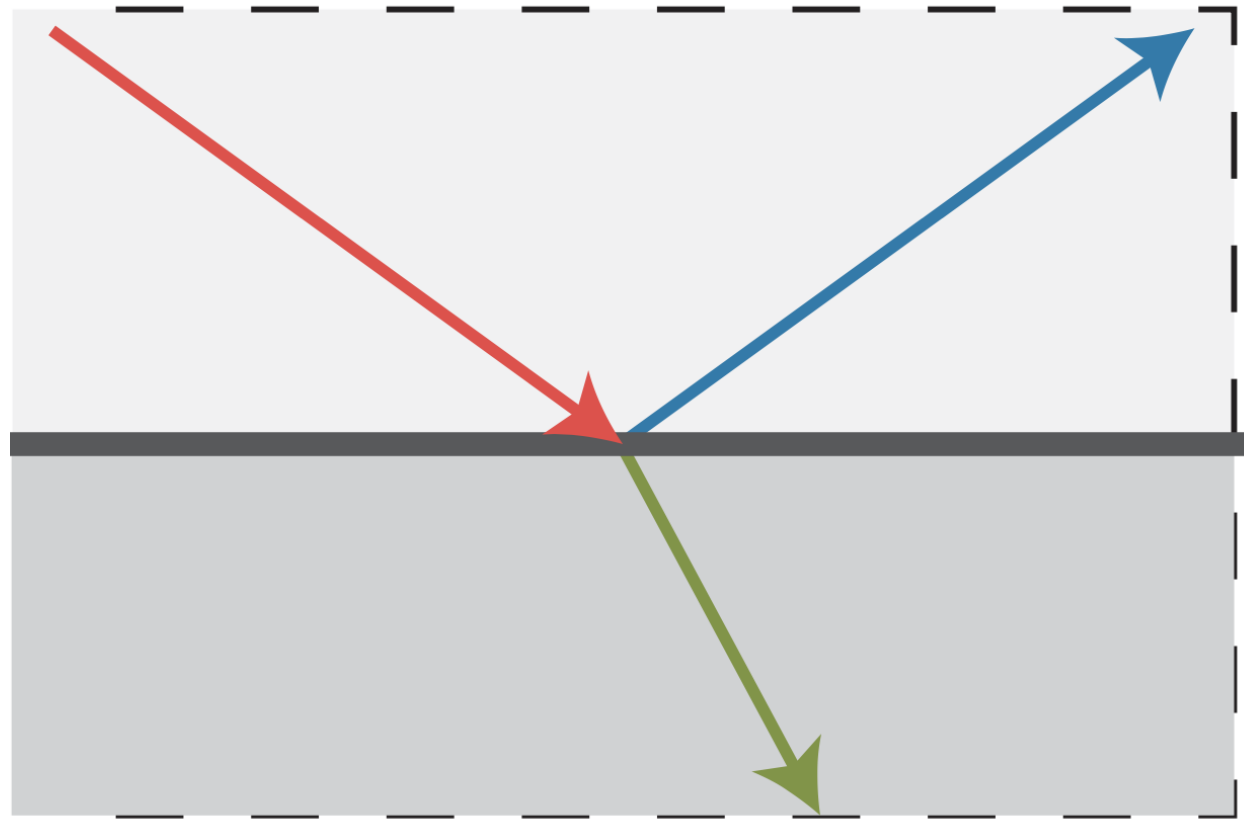
Clouds

Conductors vs. Dielectrics

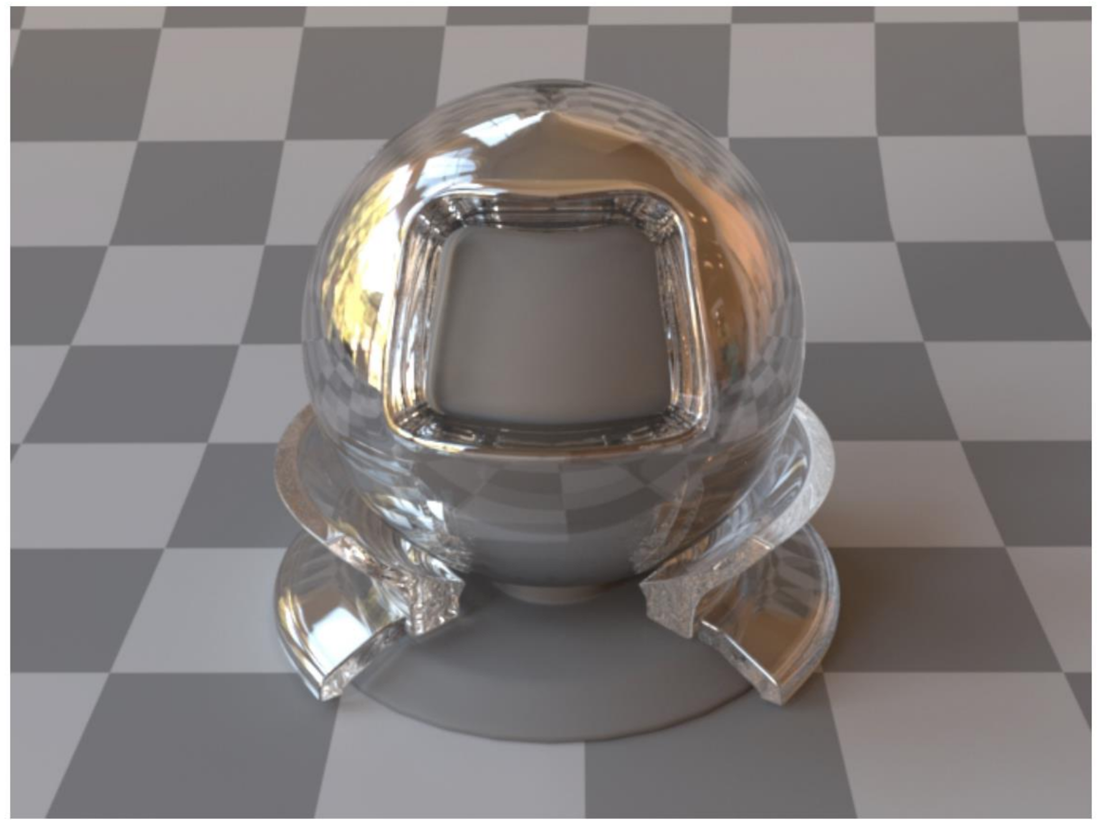


Smooth conducting material

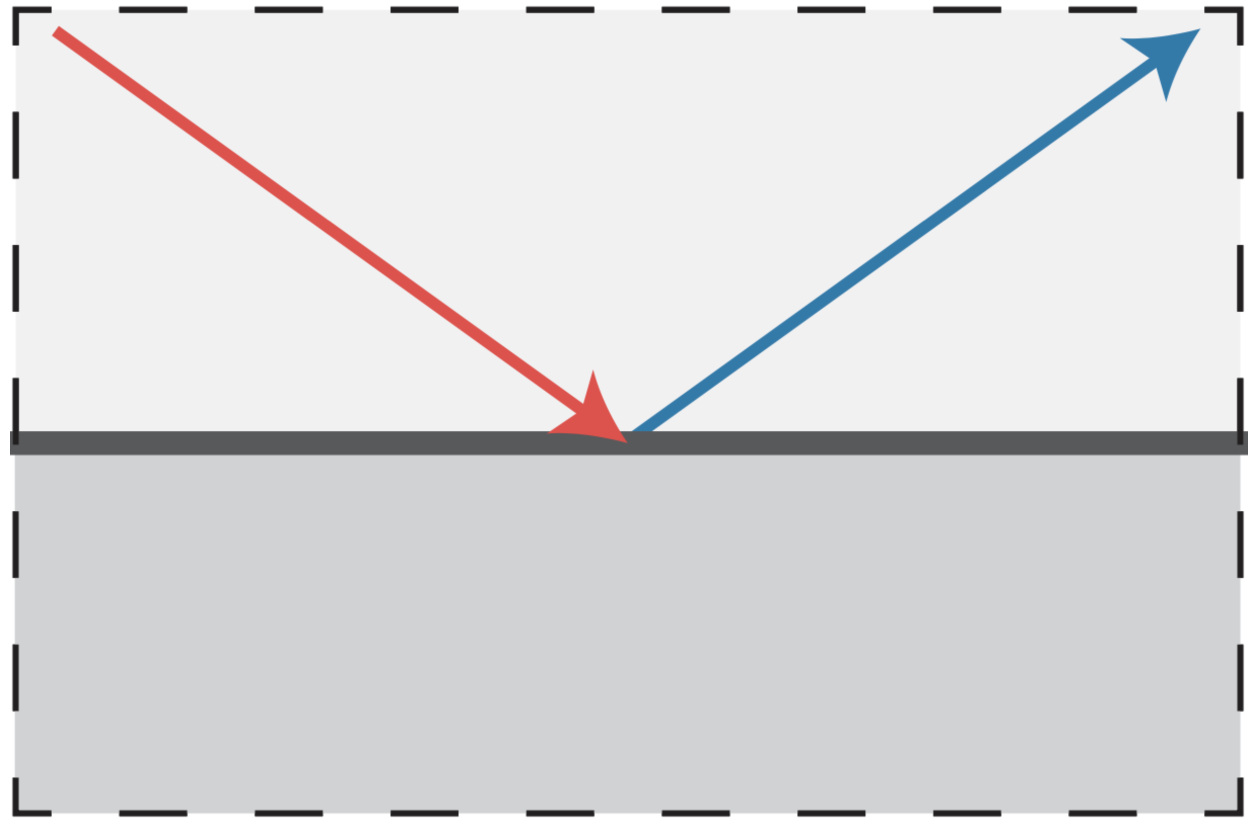
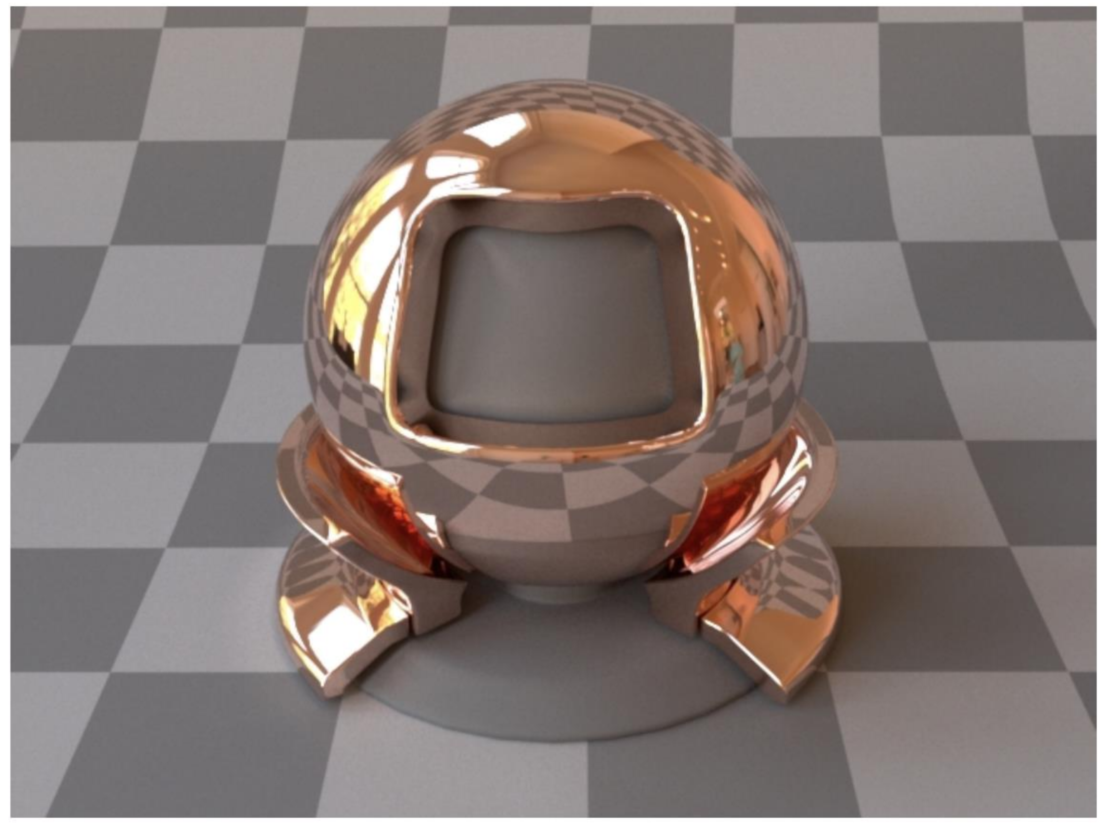




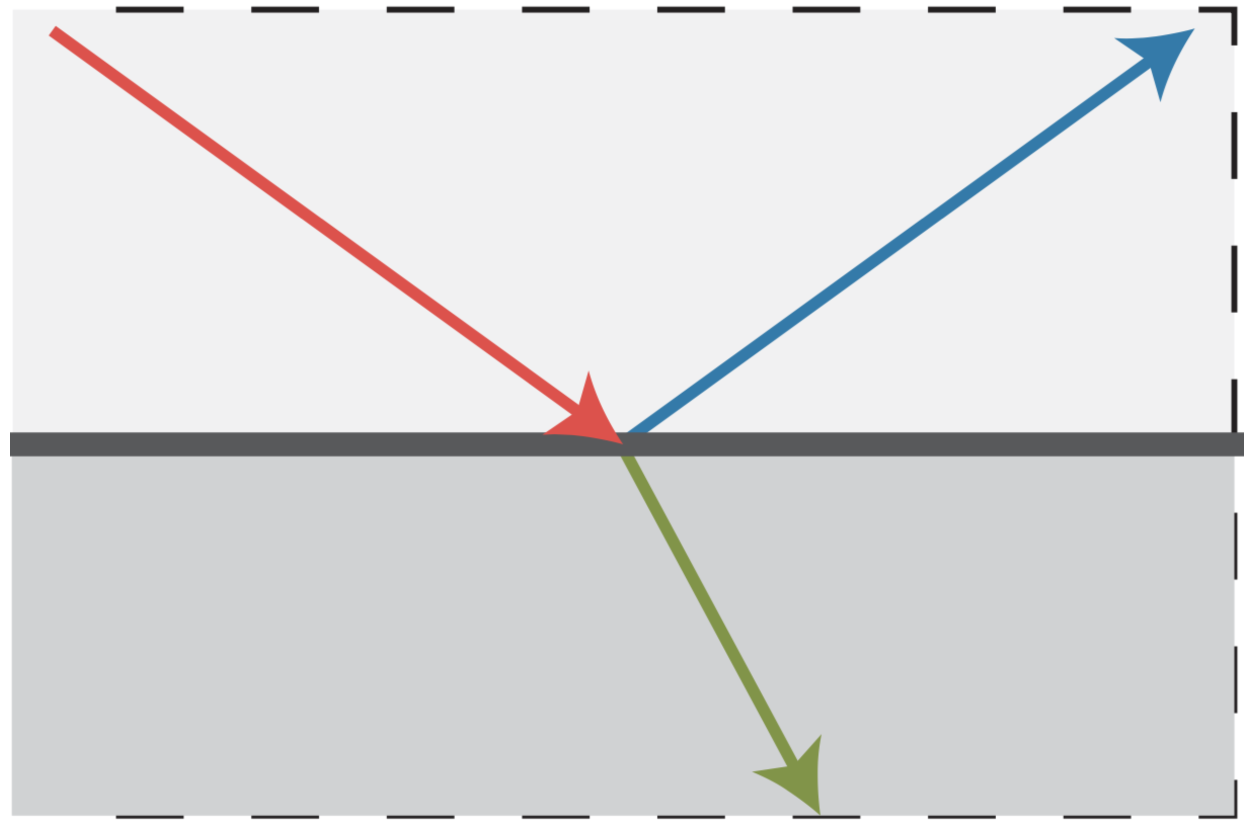
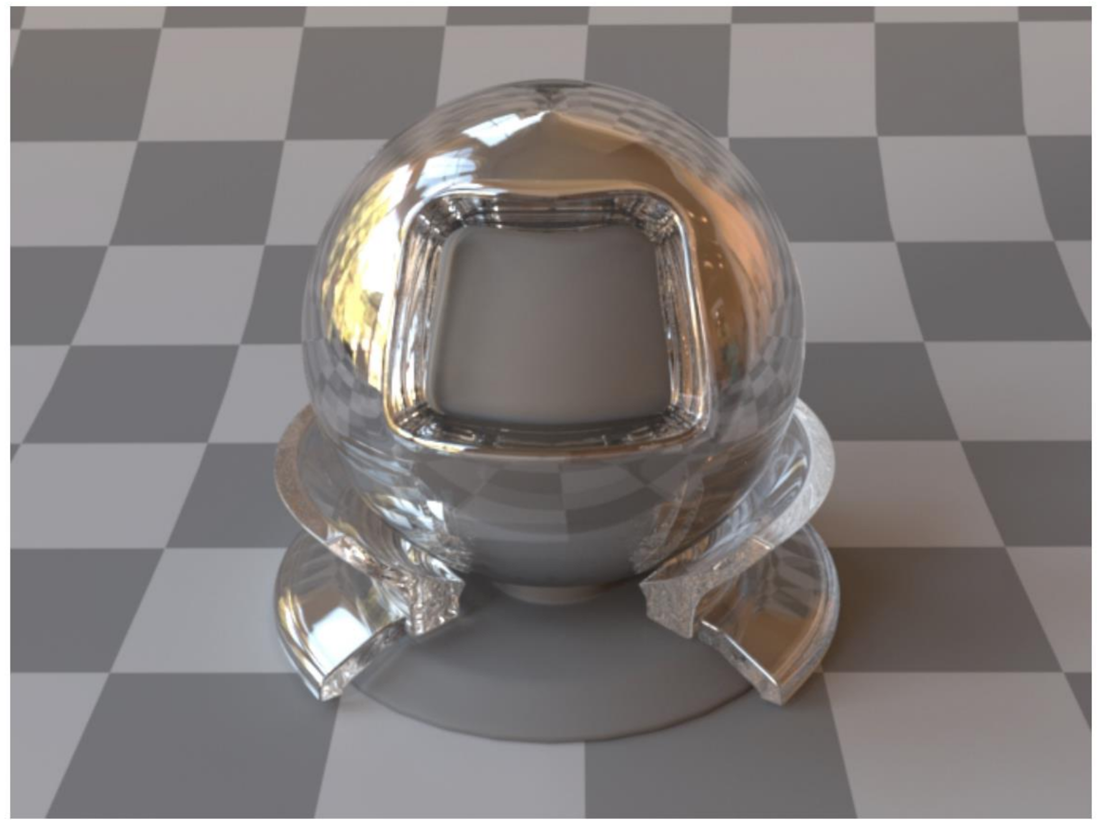
Smooth dielectric material



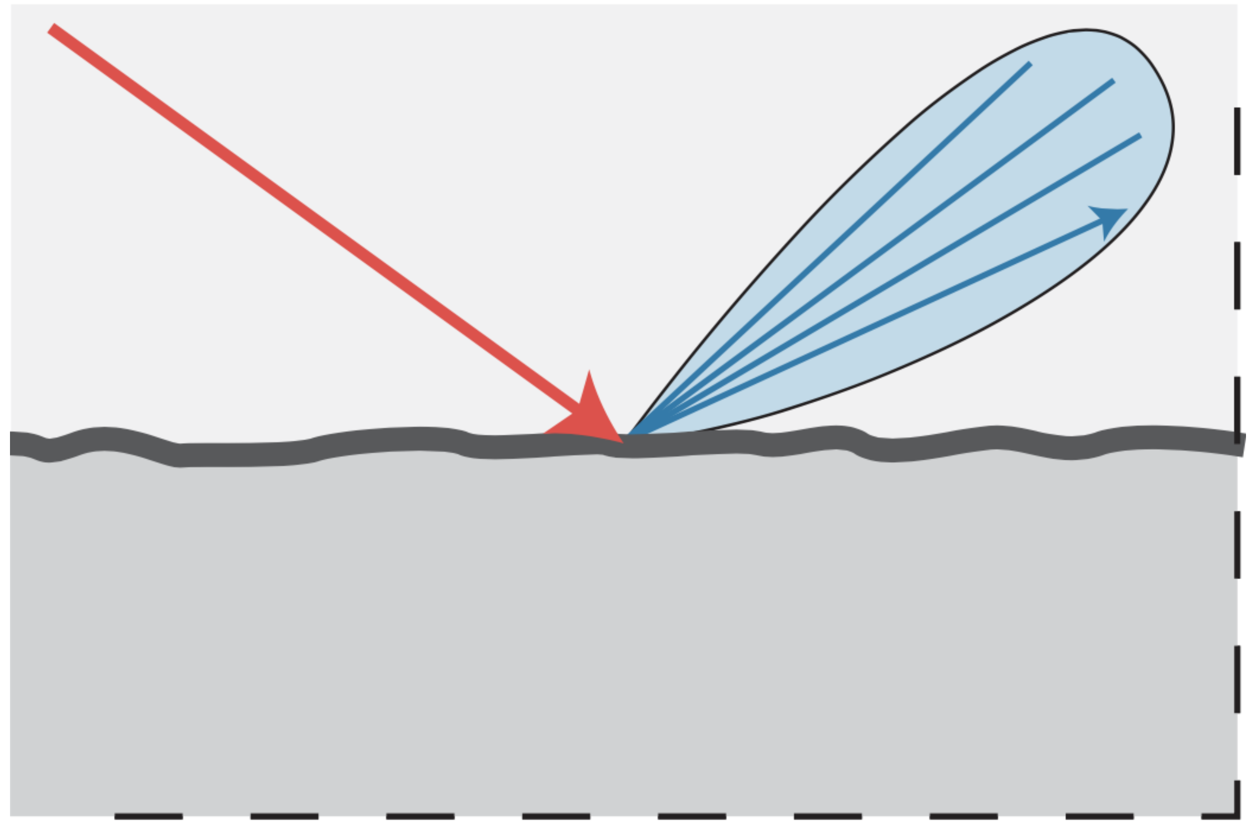
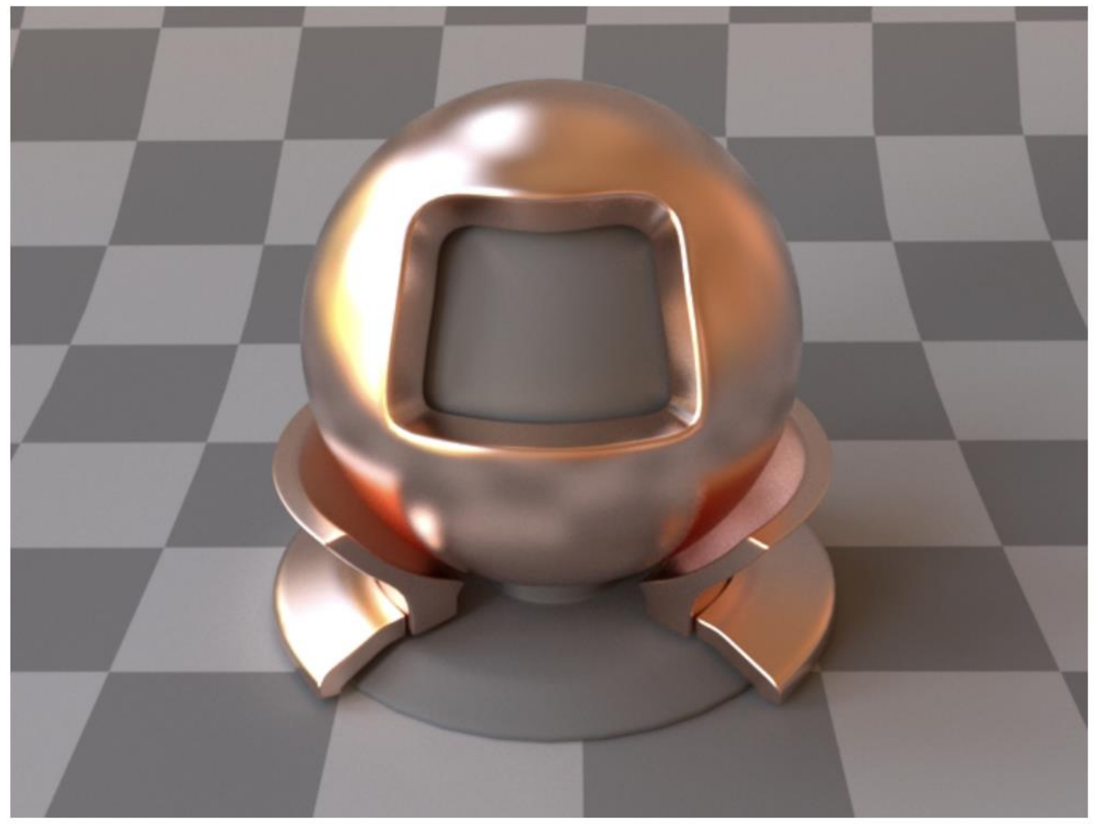
Conductors vs. Dielectrics

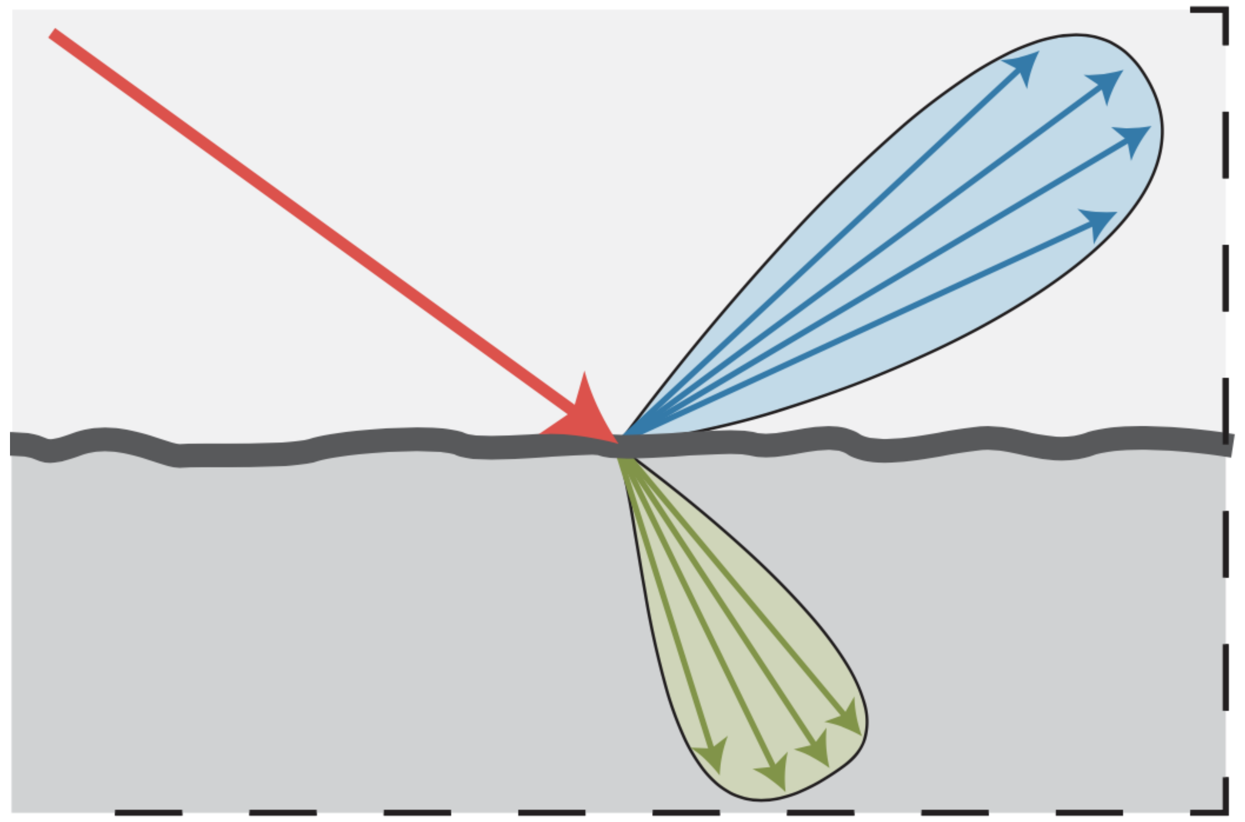
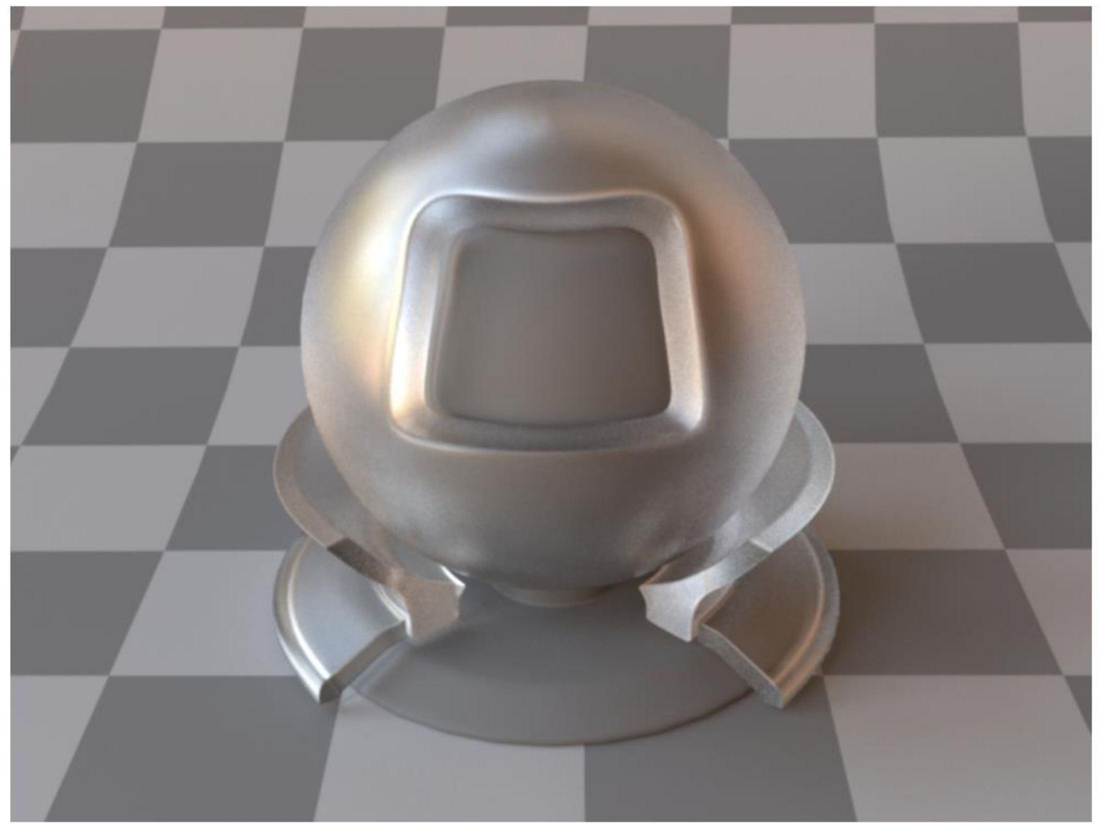
Smooth conducting material

Smooth dielectric material

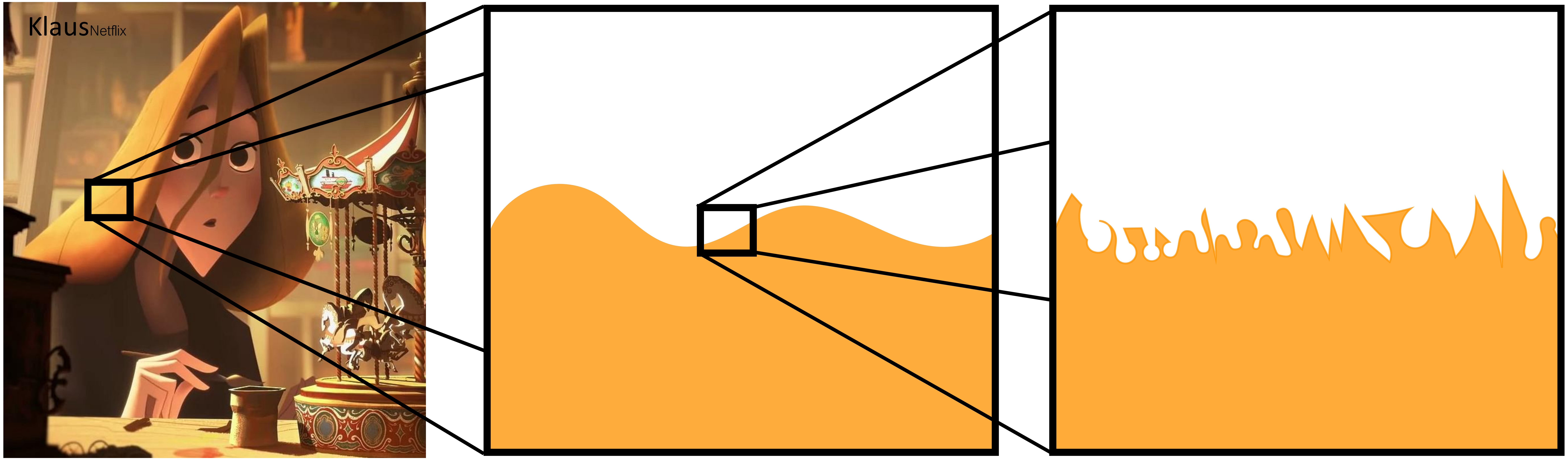
Rough conducting material

Rough dielectric material

Three Levels of Detail

Key Idea: transition from individual interactions to statistical averages



Macro Scale

Scene geometry

Meso Scale

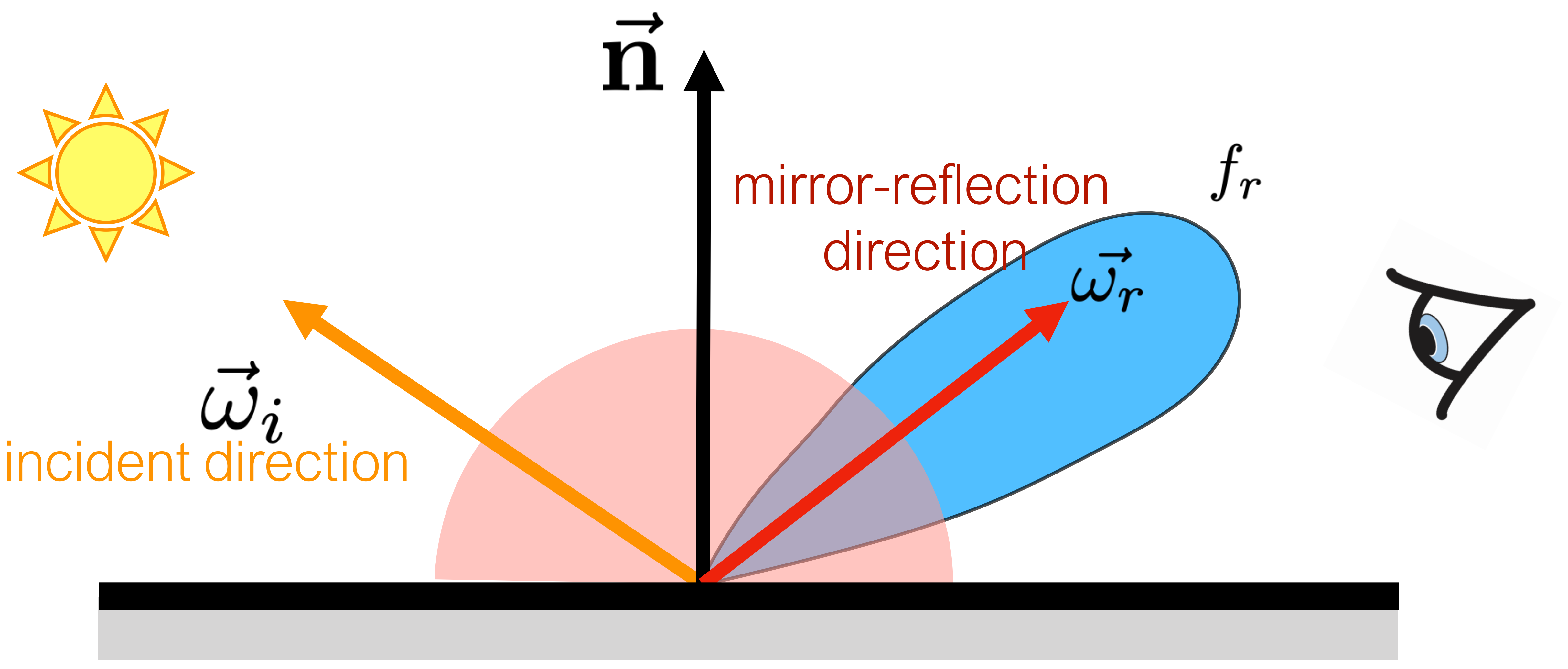
Detail at intermediate scale

Micro Scale

Roughness

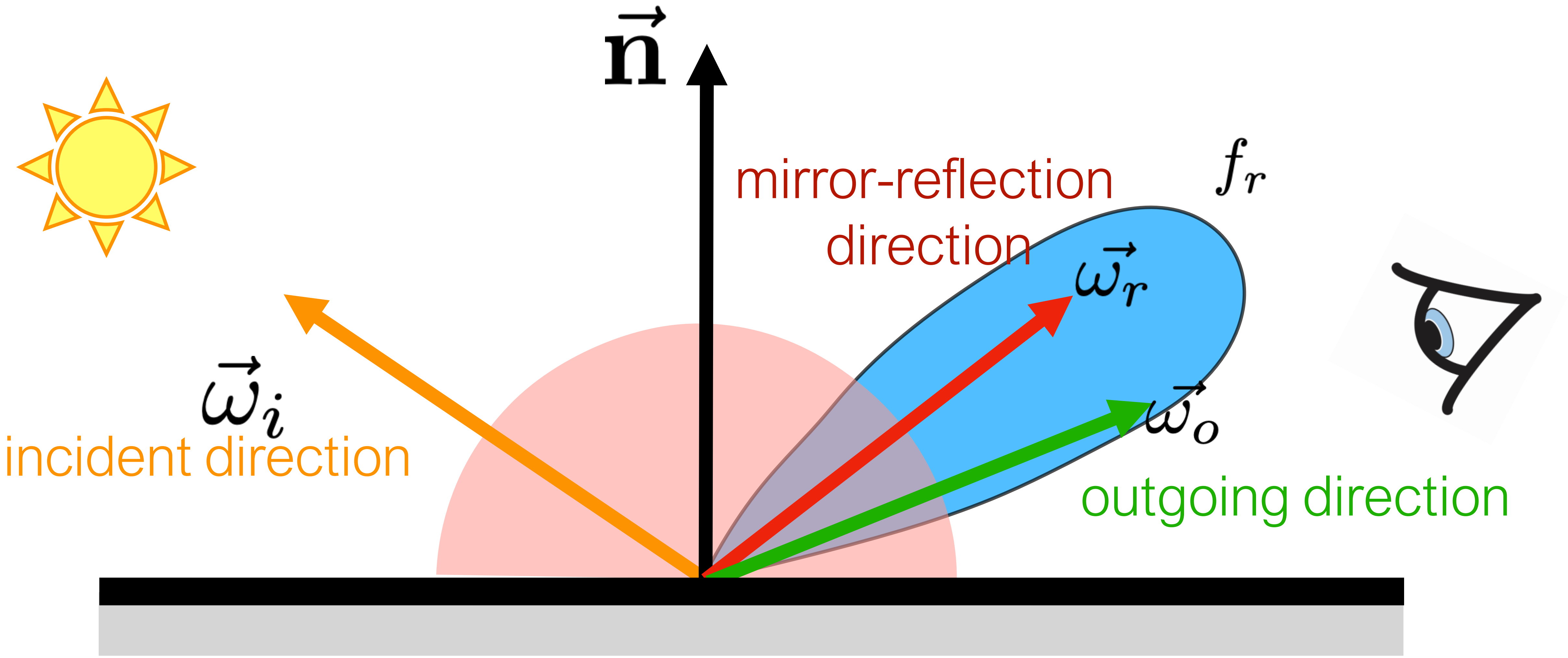
Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:



Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:

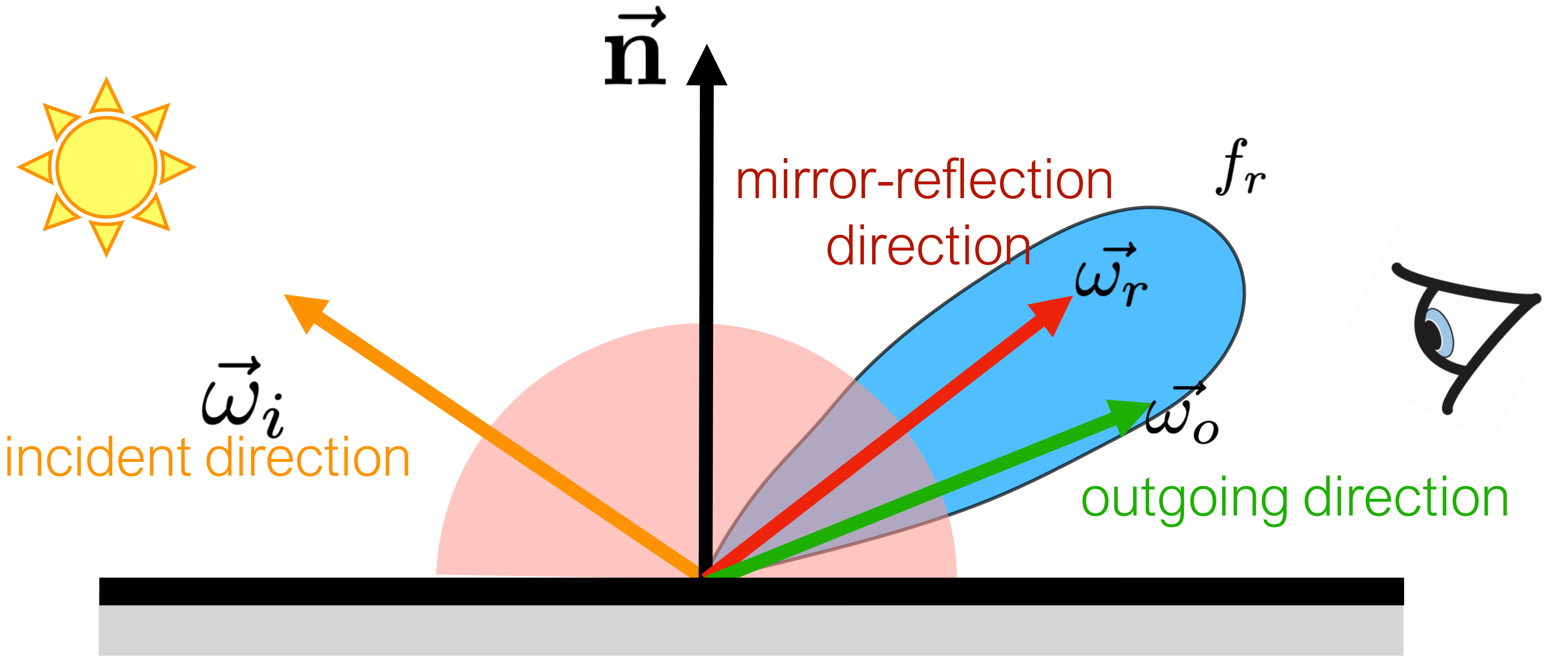


Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:

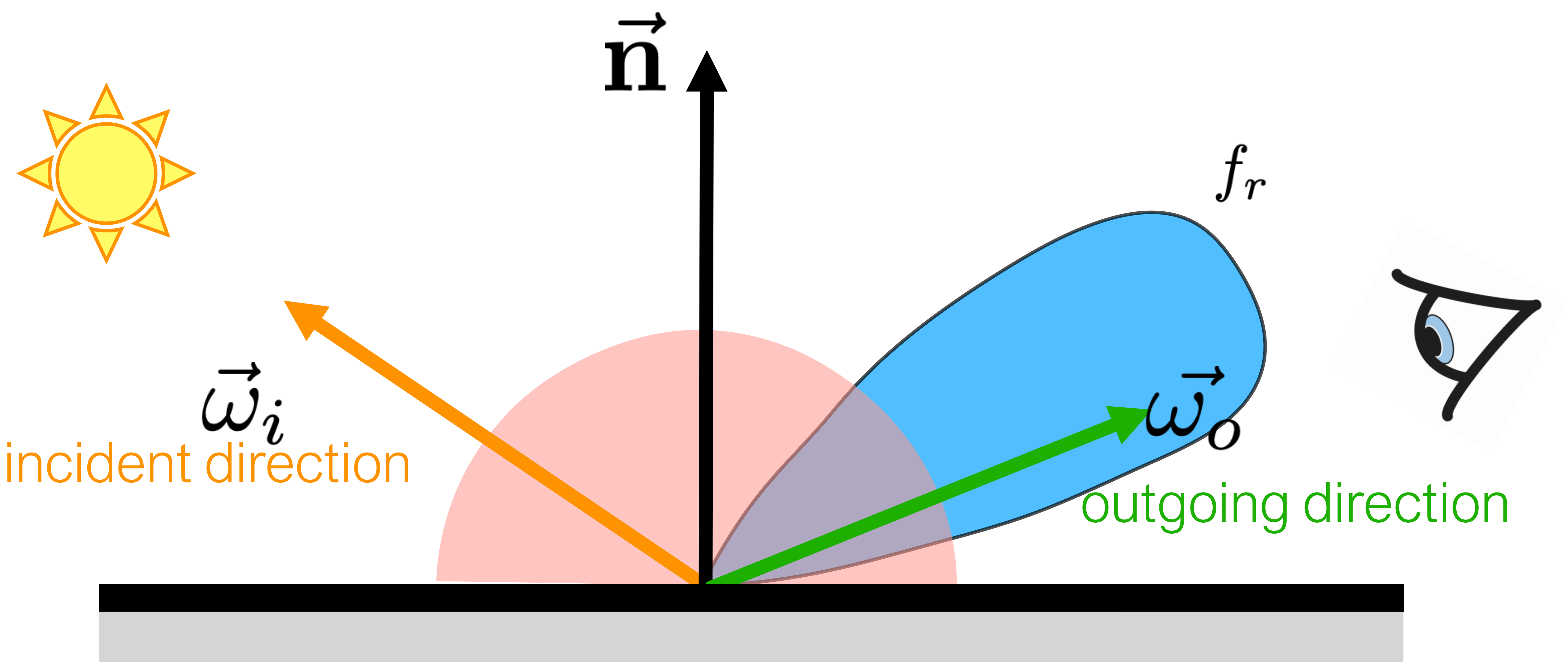
$$f_r(\vec{\omega}_o, \vec{\omega}_i) = \frac{e+2}{2\pi}(\vec{\omega_r} \cdot \vec{\omega_o})^e$$

$$\vec{\omega_r} = (2\vec{\mathbf{n}}(\vec{\mathbf{n}} \cdot \vec{\omega}_i) - \vec{\omega}_i)$$



Blinn-Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:



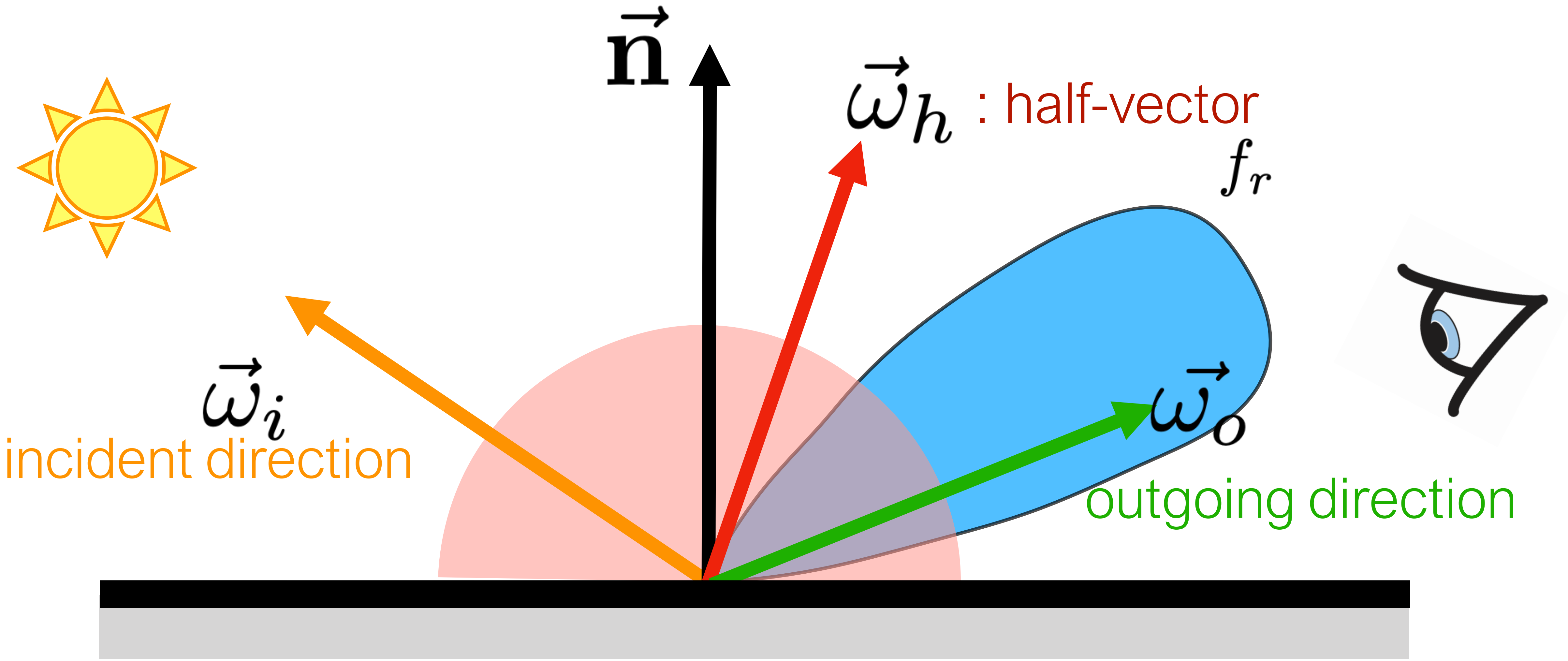
Blinn-Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:

$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{||\vec{\omega}_i + \vec{\omega}_o||}$$

$$f_r(\vec{\omega}_o, \vec{\omega}_i) = \frac{e+2}{2\pi}(\vec{\omega}_h \cdot \vec{\mathbf{n}})^e$$

$$\vec{\omega_r} = (2\vec{\mathbf{n}}(\vec{\mathbf{n}} \cdot \vec{\omega}_i) - \vec{\omega}_i)$$



Rough Surfaces

Empirical glossy models have limitations:

- not physically-based
- (often) not reciprocal

Rough Surfaces

Empirical glossy models have limitations:

- not physically-based
- (often) not reciprocal
- not energy-preserving (can be normalized): many conflicting normalizations in the literature
- (often) no Fresnel effects
- cannot accurately model appearance of many glossy surfaces

Rough Surfaces

Empirical glossy models have limitations:

- not physically-based
- (often) not reciprocal
- not energy-preserving (can be normalized): many conflicting normalizations in the literature
- (often) no Fresnel effects
- cannot accurately model appearance of many glossy surfaces

Blinn-Phong was first step in the right direction

Can do Better

Microfacet Theory

Microfacet Theory

In geometric-optics-based approaches, rough surfaces can be modeled as a collection of small microfacets.

Surfaces comprised of microfacets are often modeled as heightfields, where the distribution of facet orientations is described statistically

Microfacet Theory

Assume surface consists of tiny facets

Assume that the differential area being viewed/illuminated is relatively large compared to the size of microfacets

A facet can be perfectly specular or diffuse

Torrance-Sparrow Model

Developed by Torrance & Sparrow in 1967

Originally used in the physics community

Torrance-Sparrow Model

Developed by Torrance & Sparrow in 1967

Originally used in the physics community

Adapted by Cook & Torrance and Blinn for graphics

- added ambient and diffuse terms

Torrance-Sparrow Model

Developed by Torrance & Sparrow in 1967

Originally used in the physics community

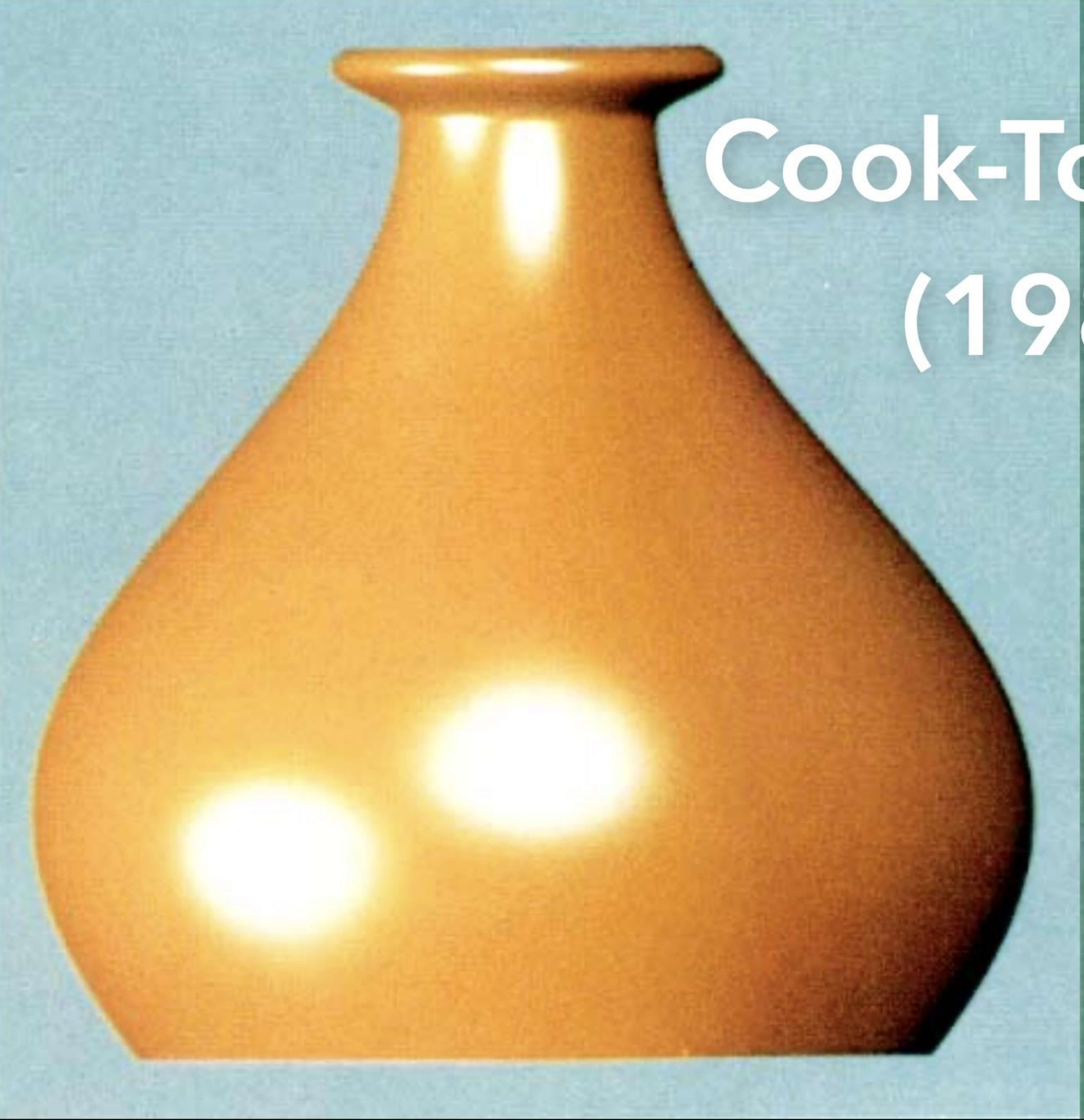
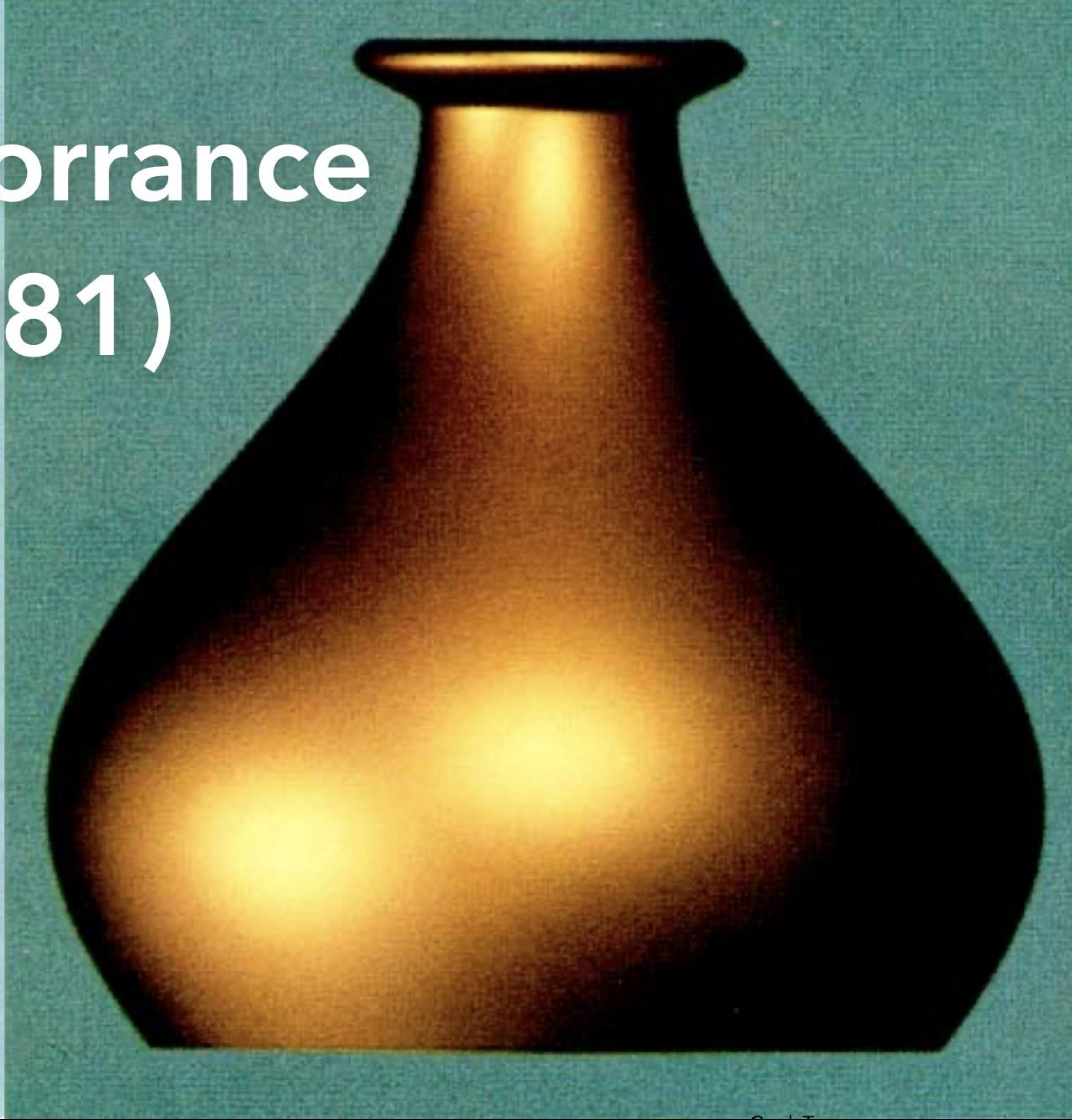
Adapted by Cook & Torrance and Blinn for graphics

- added ambient and diffuse terms

Explain off-specular peaks

Assumes surface is composed of many micro-grooves, each of which is a perfect mirror

Cook-Torrance (1981)

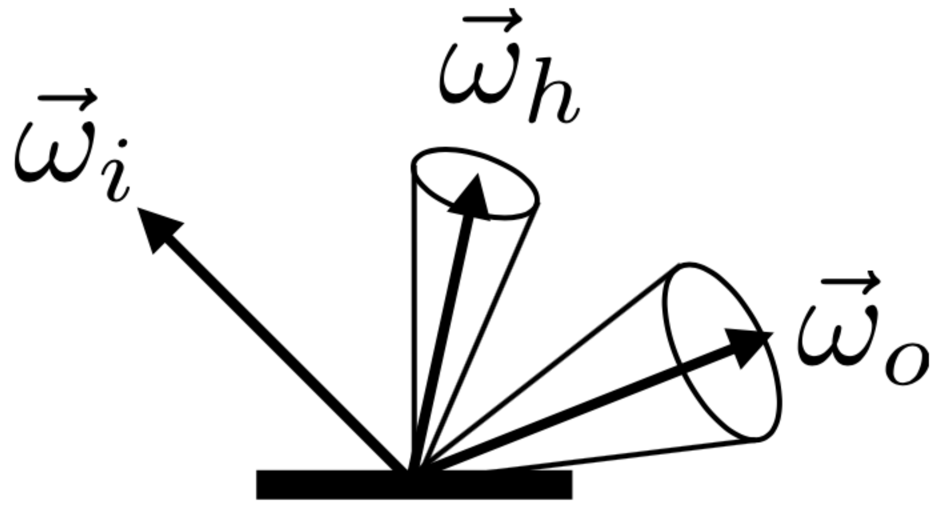



Copper-colored plastic

Copper

General Microfacet Model

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|(\vec{\omega}_i \cdot \vec{\mathbf{n}})(\vec{\omega}_o \cdot \vec{\mathbf{n}})|}$$

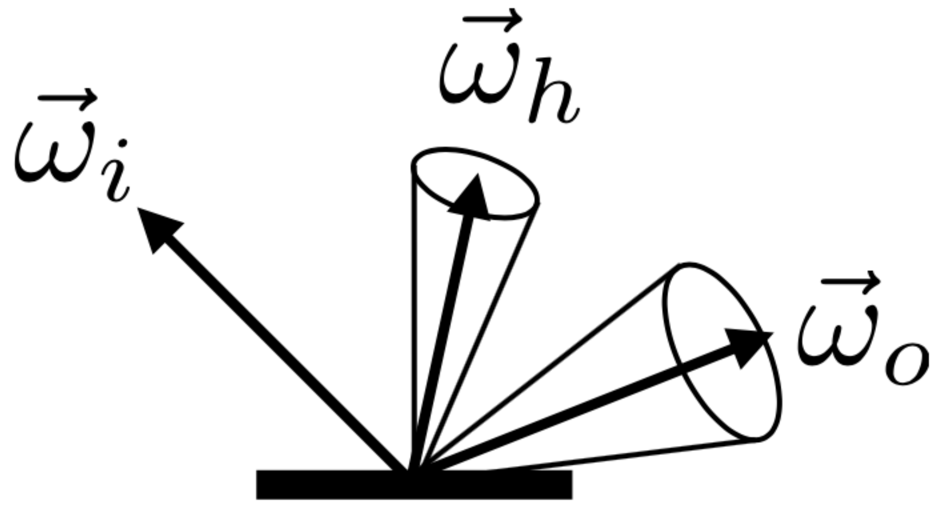


$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{\|\vec{\omega}_i + \vec{\omega}_o\|}$$

General Microfacet Model

Fresnel coefficient

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|(\vec{\omega}_i \cdot \vec{\mathbf{n}})(\vec{\omega}_o \cdot \vec{\mathbf{n}})|}$$



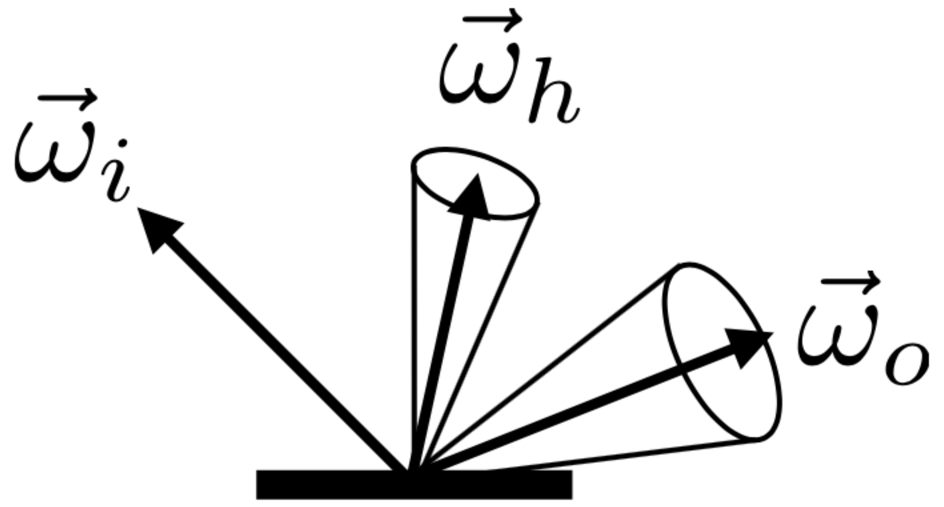
$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{\|\vec{\omega}_i + \vec{\omega}_o\|}$$

General Microfacet Model

Fresnel coefficient

Microfacet distribution

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|(\vec{\omega}_i \cdot \vec{\mathbf{n}})(\vec{\omega}_o \cdot \vec{\mathbf{n}})|}$$



$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{\|\vec{\omega}_i + \vec{\omega}_o\|}$$

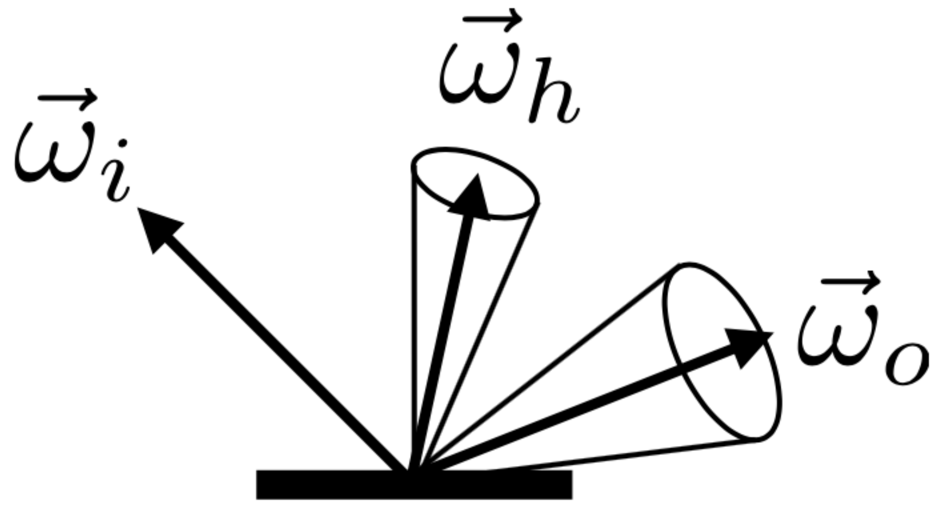
General Microfacet Model

Fresnel coefficient → $F(\vec{\omega}_h, \vec{\omega}_o)$

Microfacet distribution → $D(\vec{\omega}_h)$

shadowing/masking → $G(\vec{\omega}_i, \vec{\omega}_o)$

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4\left|(\vec{\omega}_i \cdot \mathbf{\vec{n}})(\vec{\omega}_o \cdot \mathbf{\vec{n}})\right|}$$

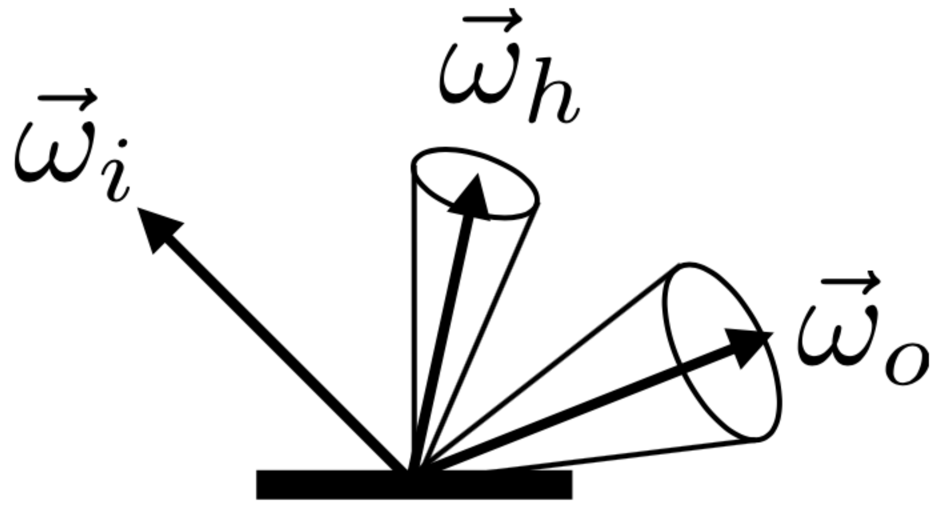


$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{\|\vec{\omega}_i + \vec{\omega}_o\|}$$

General Microfacet Model

Fresnel coefficient

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|(\vec{\omega}_i \cdot \vec{\mathbf{n}})(\vec{\omega}_o \cdot \vec{\mathbf{n}})|}$$



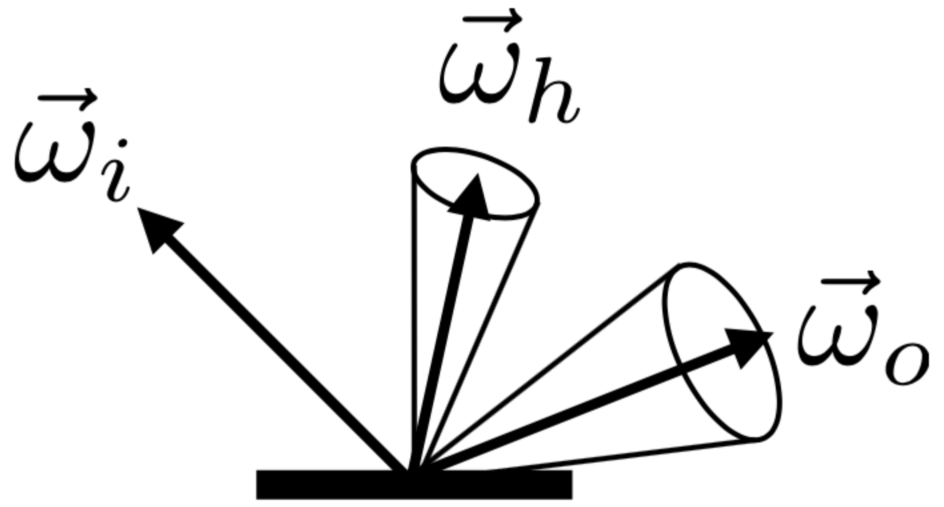
$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{\|\vec{\omega}_i + \vec{\omega}_o\|}$$

Fresnel Term

General Microfacet Model

Microfacet distribution

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|(\vec{\omega}_i \cdot \vec{\mathbf{n}})(\vec{\omega}_o \cdot \vec{\mathbf{n}})|}$$



$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{\|\vec{\omega}_i + \vec{\omega}_o\|}$$

Microfacet Distribution

Microfacet Distribution

Microfacet Distribution

Microfacet Distribution

Microfacet Distribution

How much of the surface reflects?

Microfacet Distribution

What fraction of the surface participates in the reflection?

1) difficult to say (need an actual micro surface to compute this, tedious..)

2) Solve using principles of statistical physics

- Is there anything general we can say about the surface when there are many bumps?

Microfacet Distribution

Fraction of facets facing each direction

Probability density function over projected solid angle (must be normalized):

$$\int_{\mathcal{H}^2} D(\vec{\omega}_h) \cos\theta_h d\vec{\omega}_h = 1$$

Beckmann-Spizzichino Model

The slopes follow a Gaussian distribution

Let's express slope distribution w.r.t. directions

$$D(\vec{\omega}_h) = \qquad \exp^{-\frac{tan^2\theta_h}{\alpha^2}}$$

Beckmann-Spizzichino Model

The slopes follow a Gaussian distribution

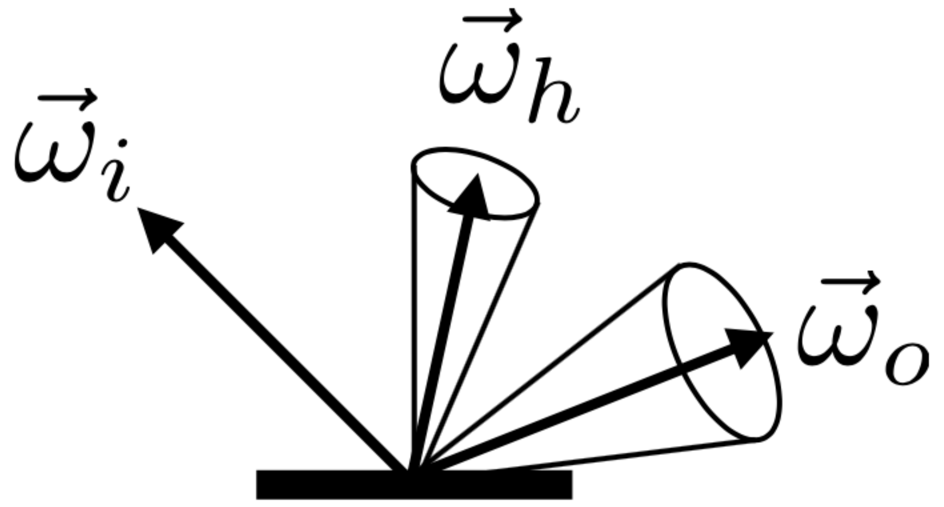
Let's express slope distribution w.r.t. directions

$$D(\vec{\omega}_h) = \frac{1}{\pi\alpha^2\cos^4\theta_h}\exp^{-\frac{tan^2\theta_h}{\alpha^2}}$$

General Microfacet Model

shadowing/masking

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|(\vec{\omega}_i \cdot \mathbf{\vec{n}})(\vec{\omega}_o \cdot \mathbf{\vec{n}})|}$$



$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{||\vec{\omega}_i + \vec{\omega}_o||}$$

Microfacet Distribution: Masking effect

Microfacet Distribution: Masking effect

The microfacet of interest not visible to the viewer due to occlusions

Masking effect:

Microfacet Distribution: Shadowing effect

Microfacet Distribution: Shadowing/Masking

Light bounces among the facets before reaching the viewer

Microfacet Distribution: Shadowing/Masking

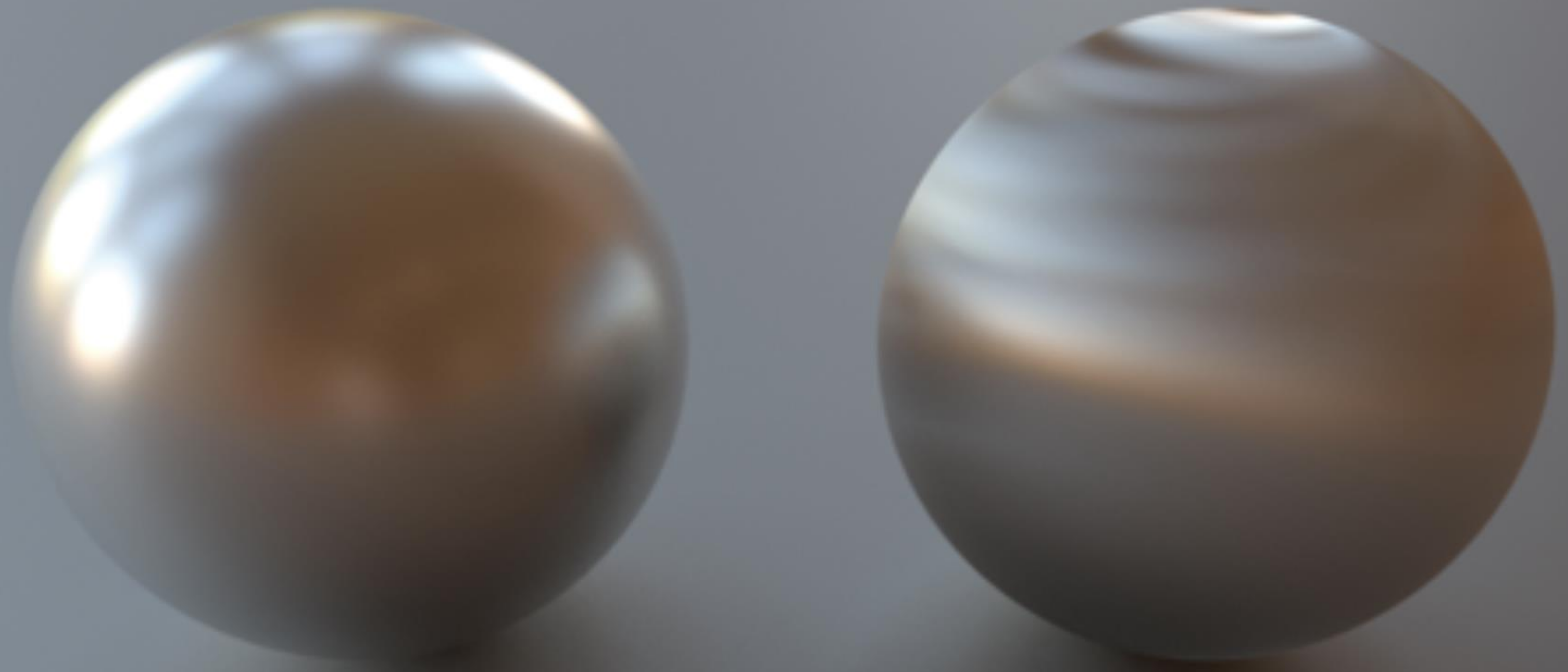
Light bounces among the facets before reaching the viewer

Microfacet Distribution: Interreflection

Light bounces among the facets before reaching the viewer

Reading

- PBRT Section 8.4
- GGX Distribution, Walter et al. (EGSR 2007)
- Isotropic and anisotropic microfacet distributions
- Oren–Nayar model, a "directed-diffuse" microfacet model, with perfectly diffuse (rather than specular) microfacets.
- Ashikhmin-Shirley model, allowing for anisotropic reflectance, along with a diffuse substrate under a specular surface



Isotropic microfacet distribution

Anisotropic microfacet distribution

PBRT v3 [2016]

Source: Midjourney

Acknowledgements

Slides material borrowed from multiple resources.

Special thanks to Wojciech Jarosz for making his rendering lectures available online