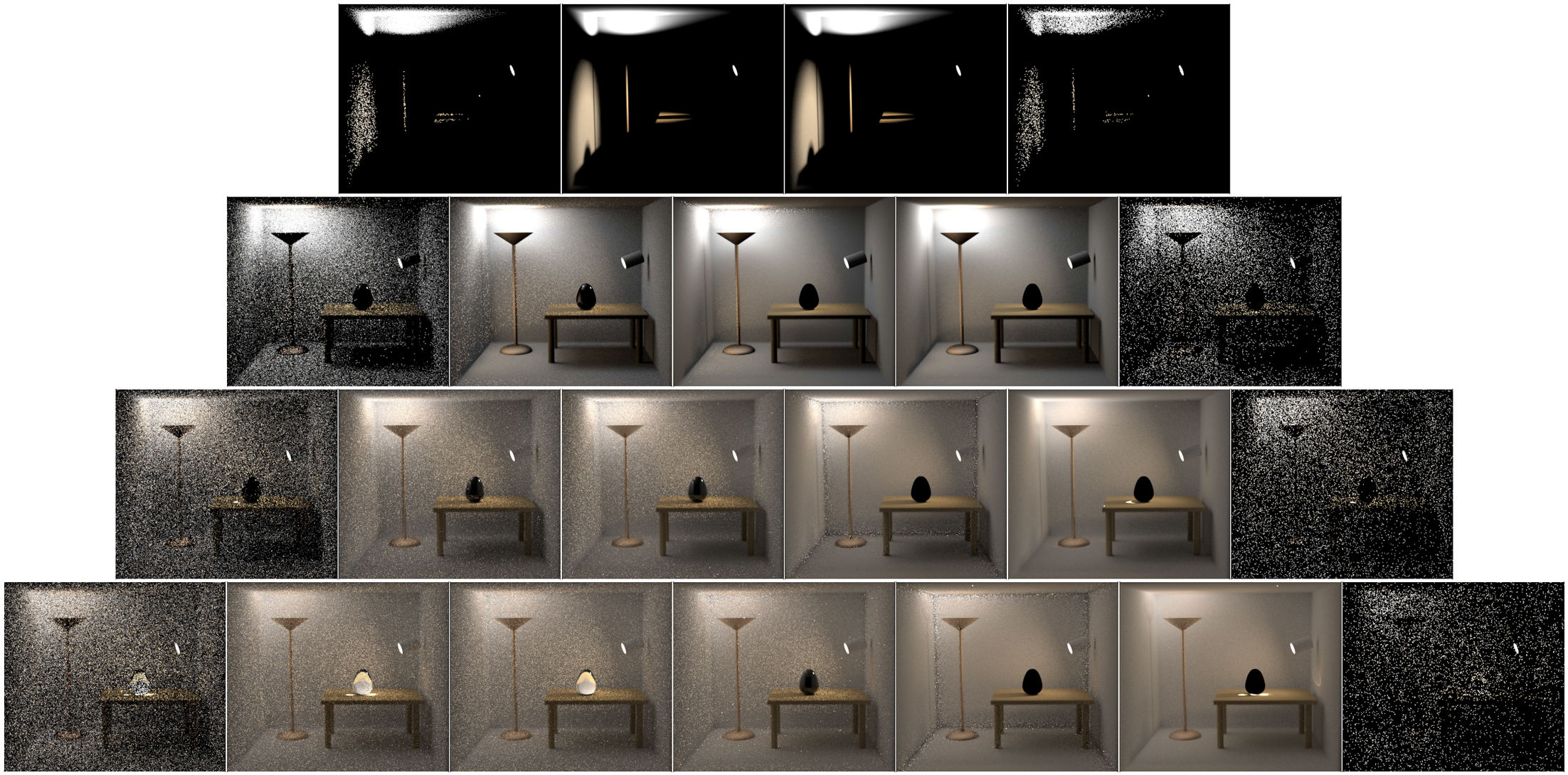
CS 87/187, Spring 2016

Rendering Algorithms

Global Illumination II: Bidirectional Path Tracing



Prof. Wojciech Jarosz

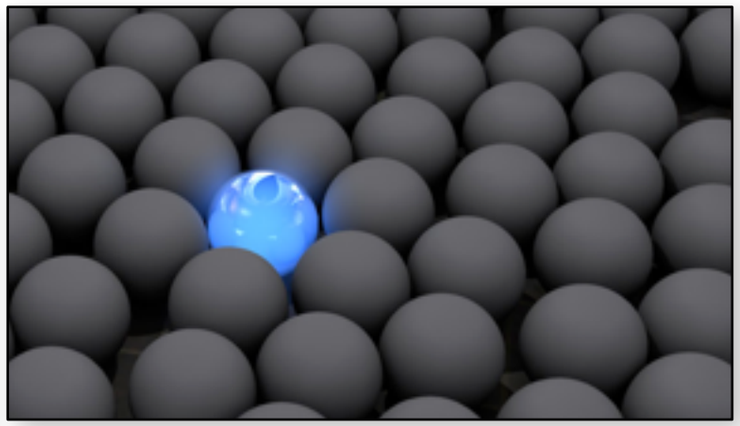
wojciech.k.jarosz@dartmouth.edu

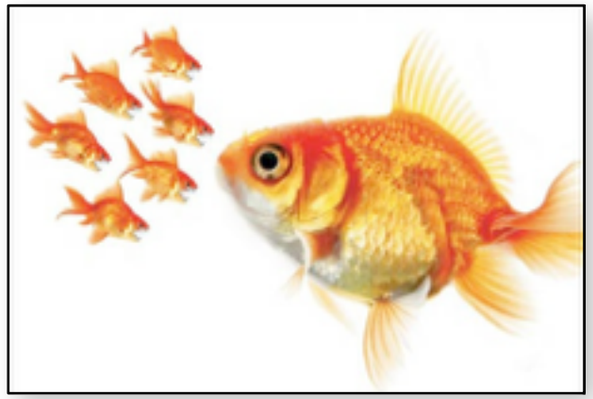
(with some slides by Jan Novák and Wenzel Jakob)

Dartmouth

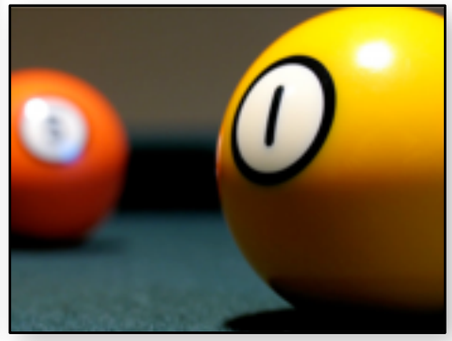
Rendering Competition

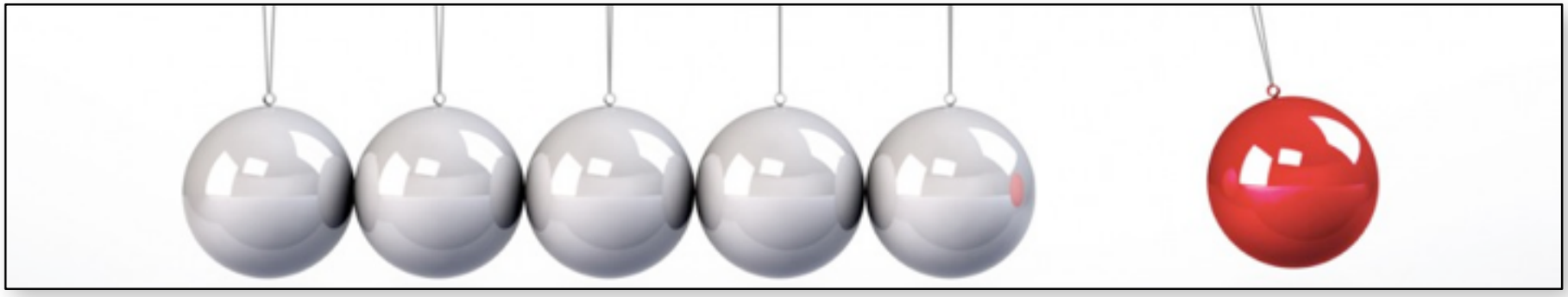
- Render a (nice) realistic image
  - evaluated on a combination of:
    aesthetic appeal, technical difficulty, realism
- Somehow incorporate the theme: "Contrast"






indigo
renderer





Competition Judges

- Lorie Loeb (CS Prof. and DALI)
- Neel Joshi (Microsoft Research)
- Derek Nowrouzezahrai (University of Montreal)

Grand Prize

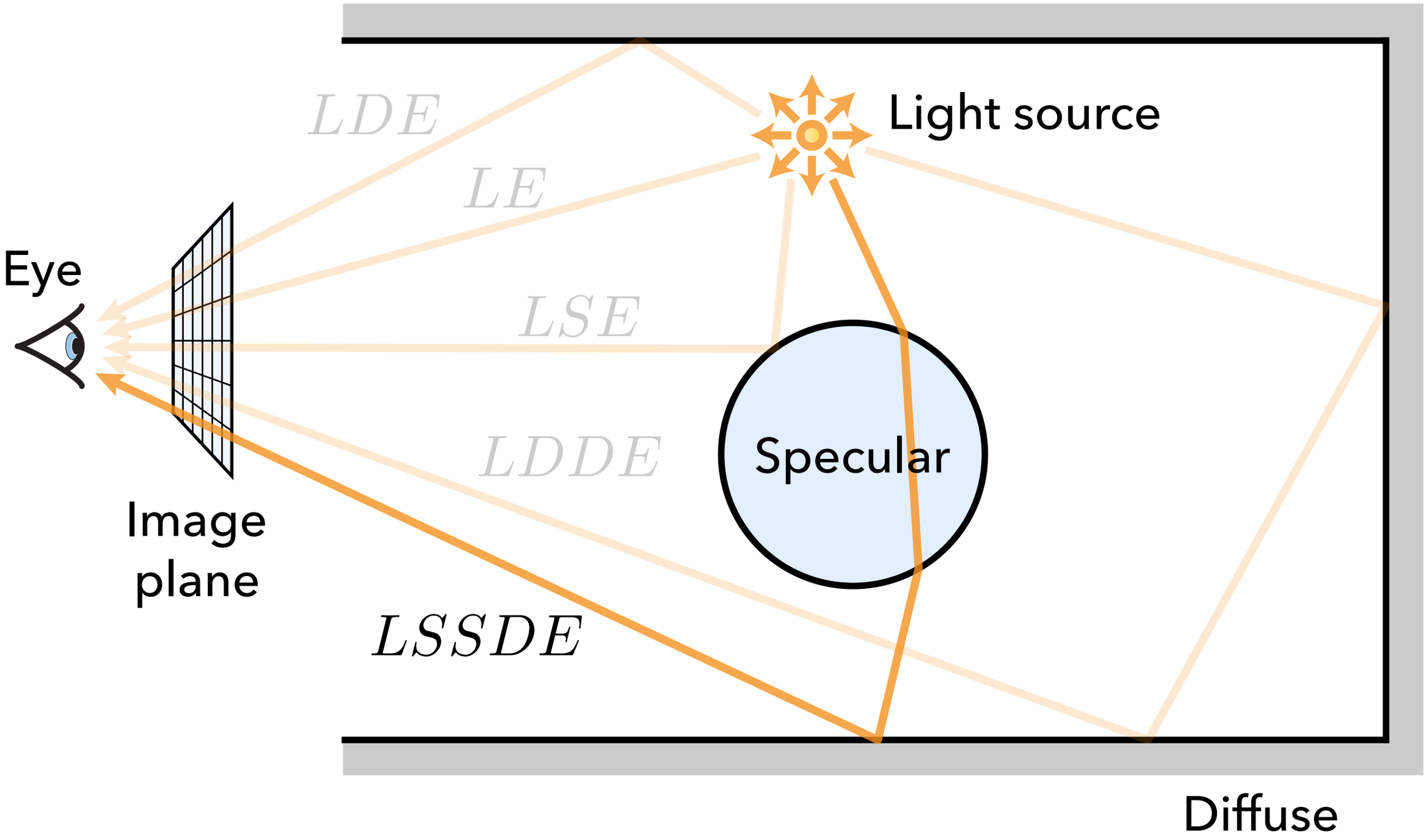
Generously donated by Microsoft!

Final project proposal

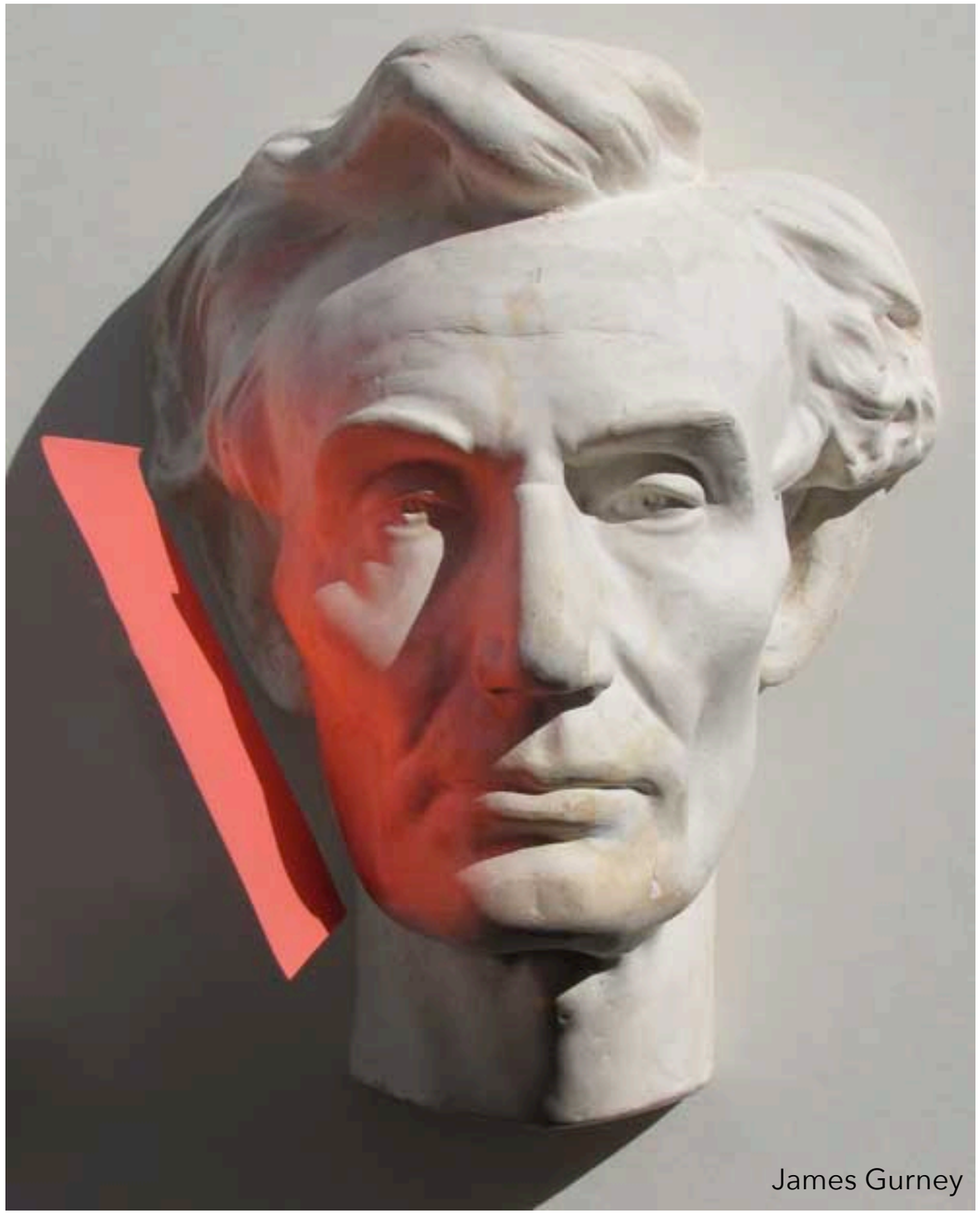
Last time

- Light paths & Heckbert classification
- Rendering Equation
- Solving the Rendering Equation
  - Recursive Monte Carlo ray tracing
  - Path tracing

Heckbert's Classification



Color Bleeding



Caustics

Rendering Equation

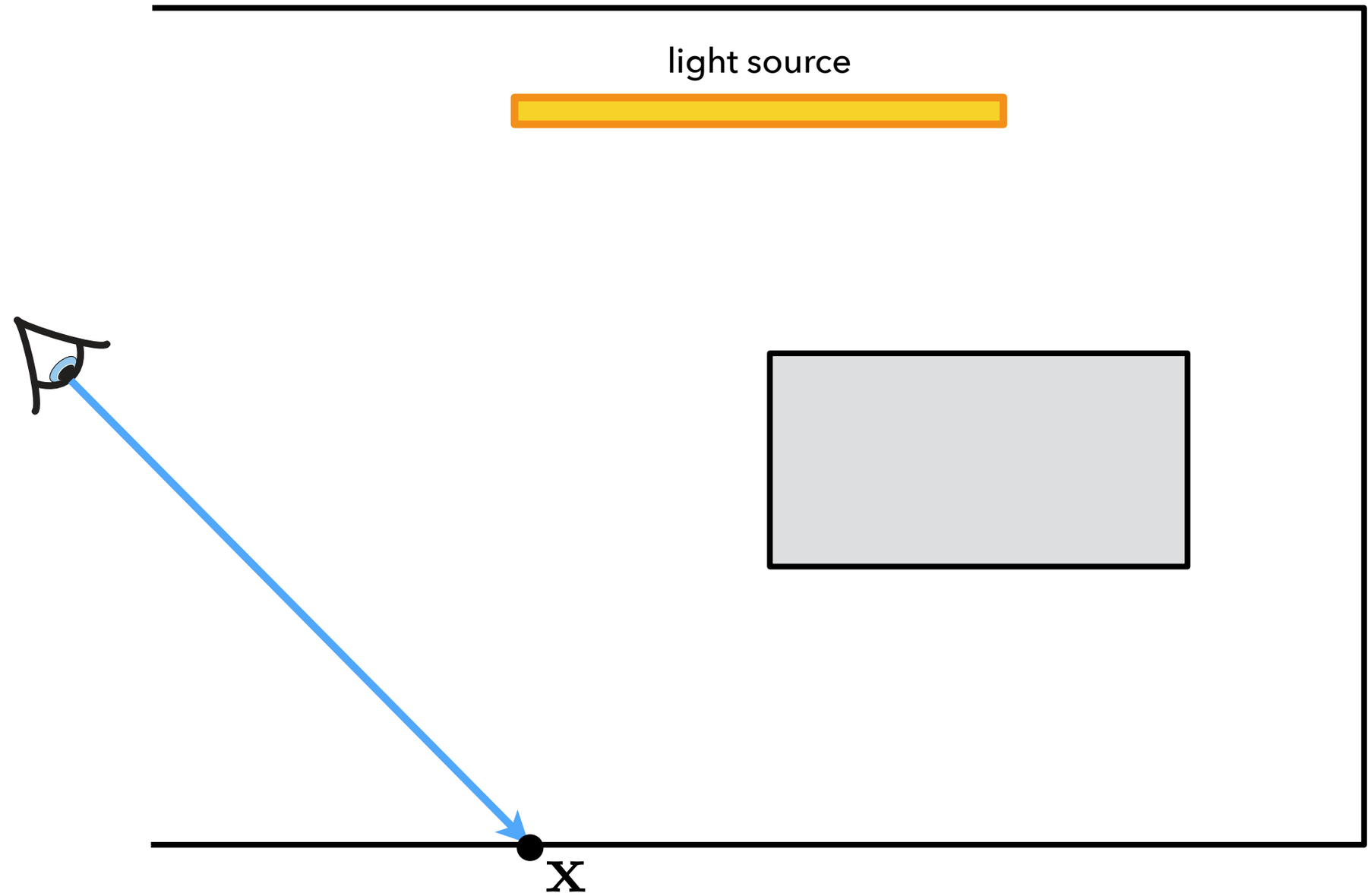
ray tracing function

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

- Only outgoing radiance on both sides

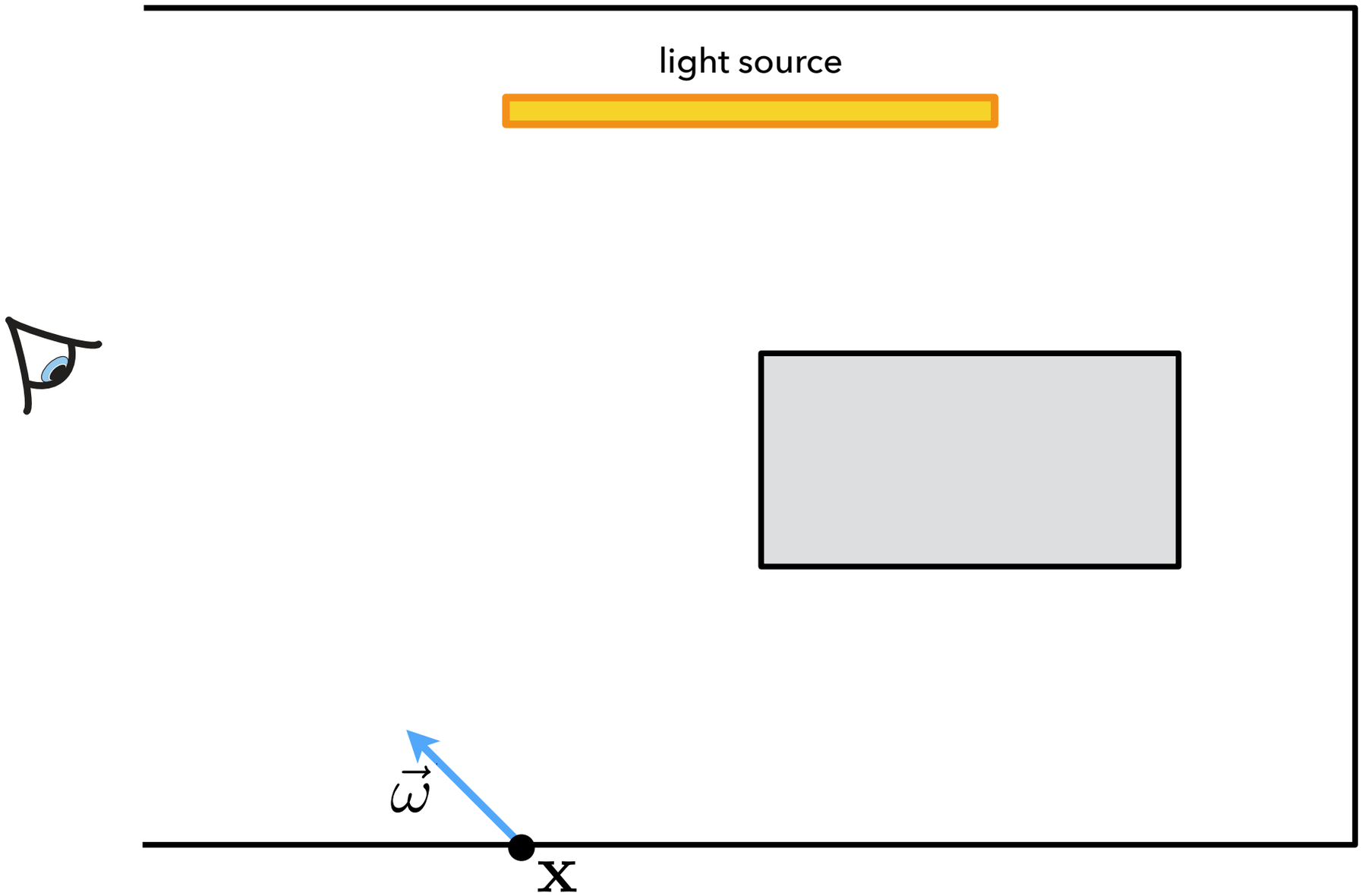
Rendering Equation

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



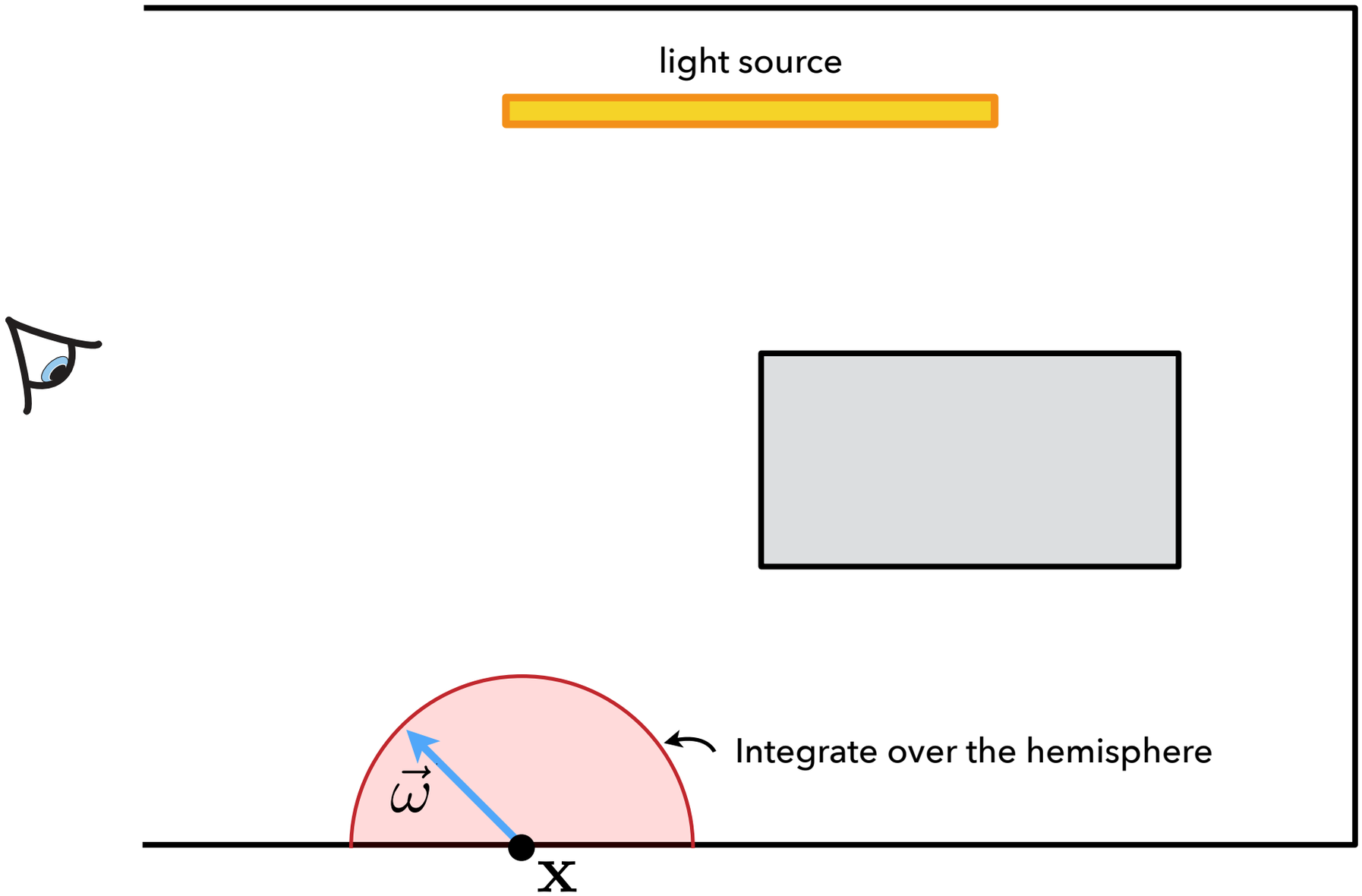
Rendering Equation

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



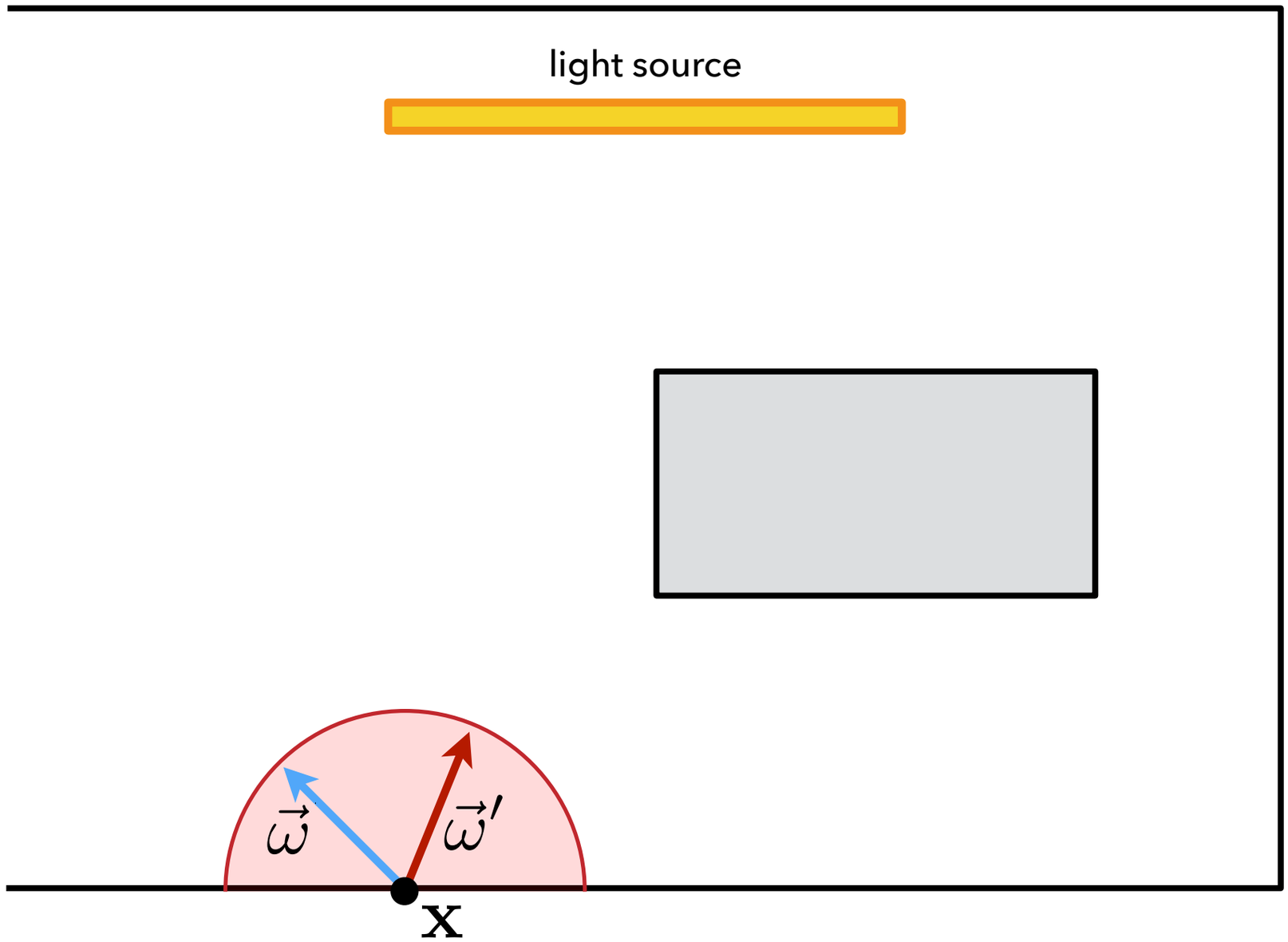
Rendering Equation

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



Rendering Equation

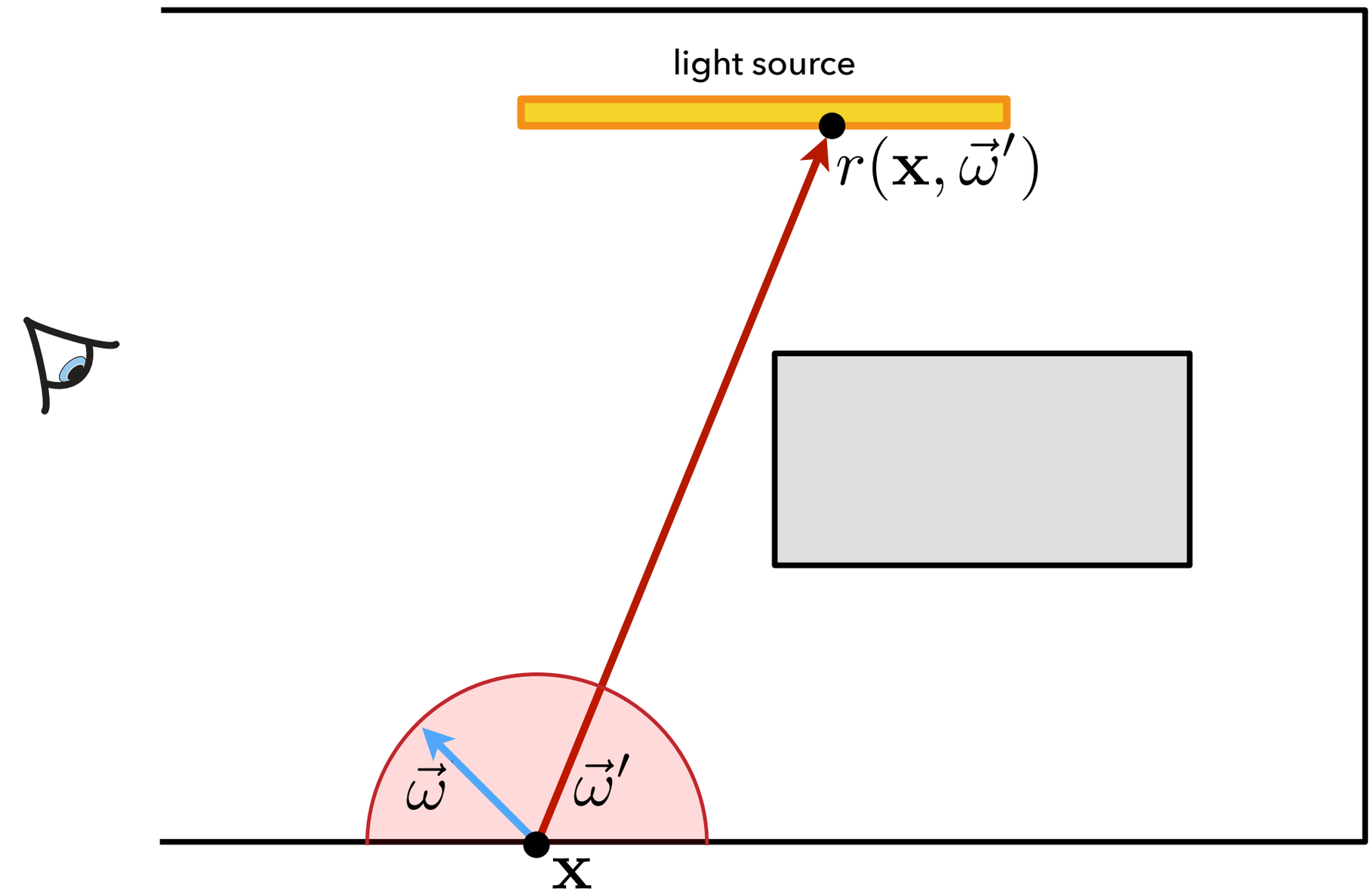
$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{H^2} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(r(\mathbf{x}, \vec{\omega}'), -\vec{\omega}') \cos\theta' \, d\vec{\omega}'$$



Rendering Equation

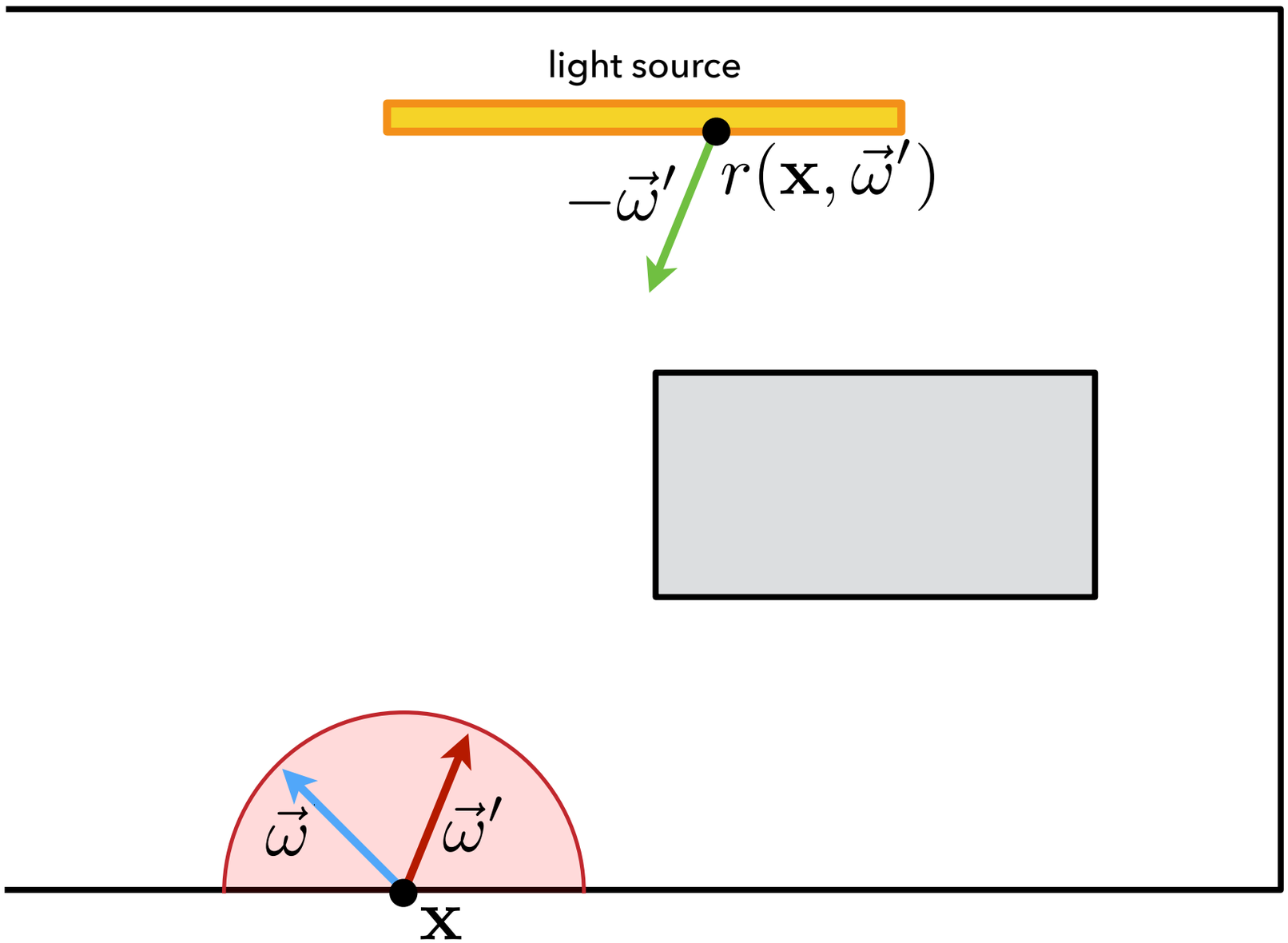
ray tracing function

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



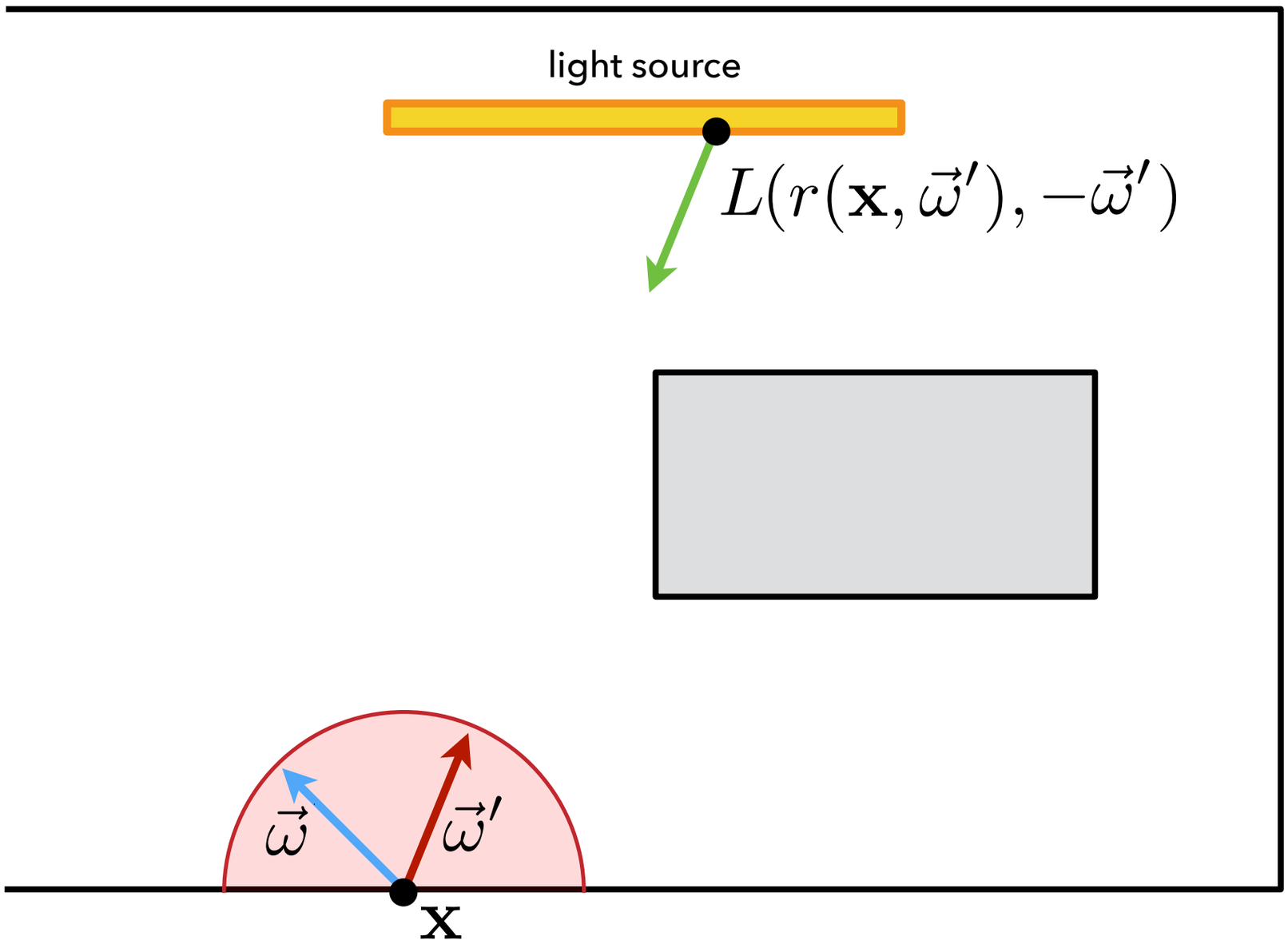
Rendering Equation

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



Rendering Equation

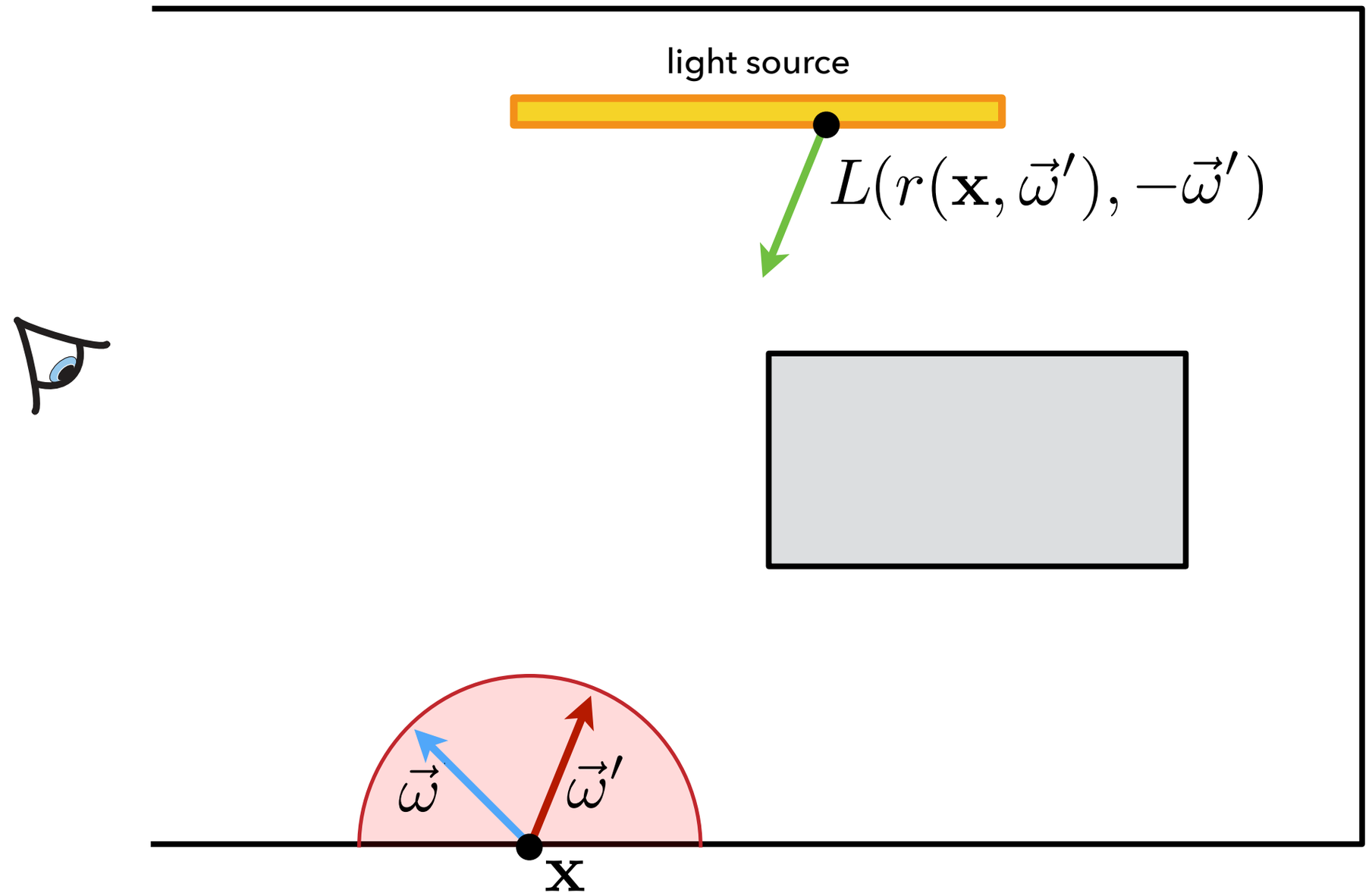
$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{H^2} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(r(\mathbf{x}, \vec{\omega}'), -\vec{\omega}') \cos\theta' \, d\vec{\omega}'$$



Rendering Equation

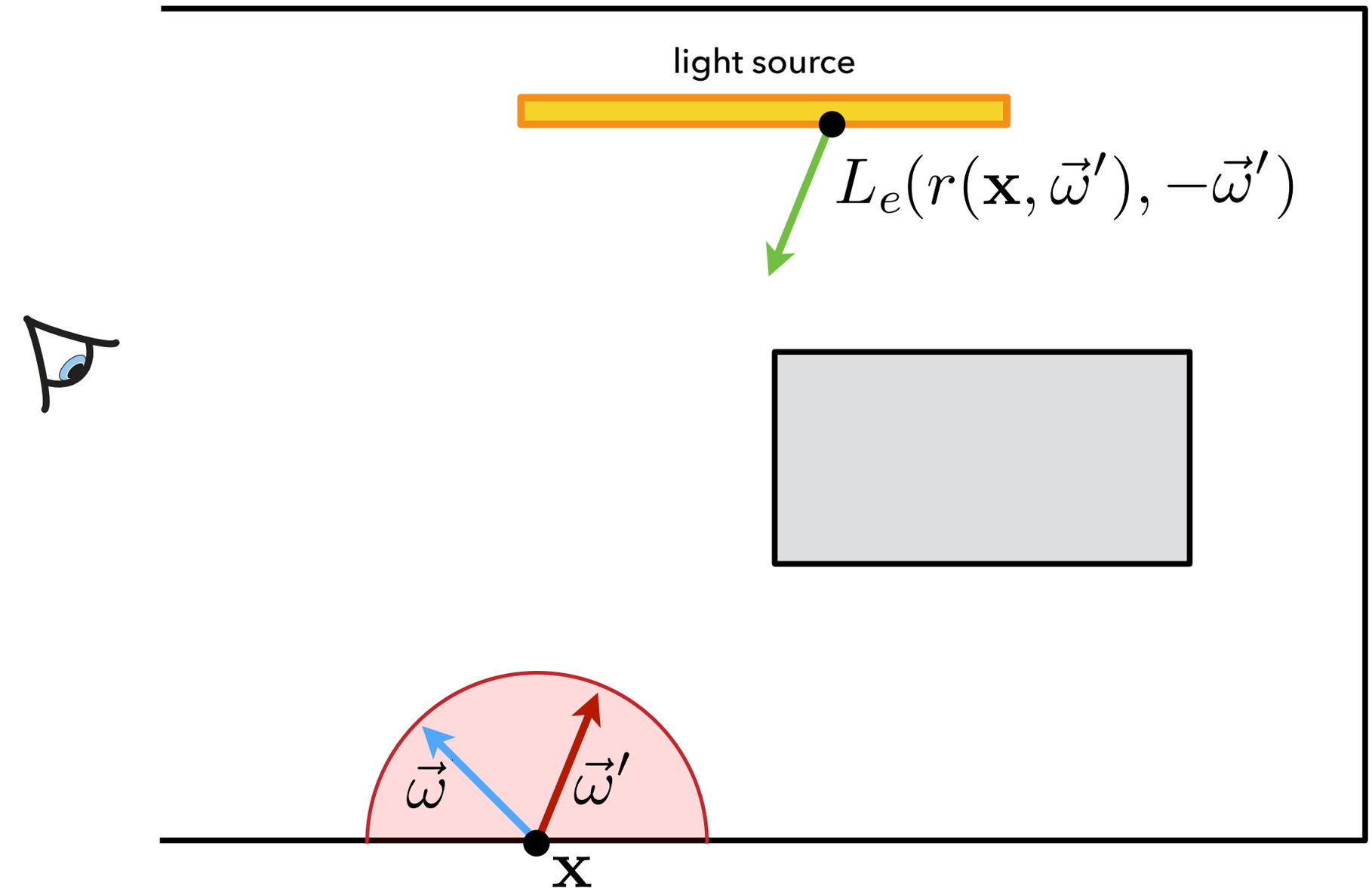
$$L(\mathbf{x},\vec{\omega}) = L_e(\mathbf{x},\vec{\omega}) + \int_{H^2} f_r(\mathbf{x},\vec{\omega}',\vec{\omega}) L(r(\mathbf{x},\vec{\omega}'),-\vec{\omega}') \cos\theta' d\vec{\omega}'$$

recursion



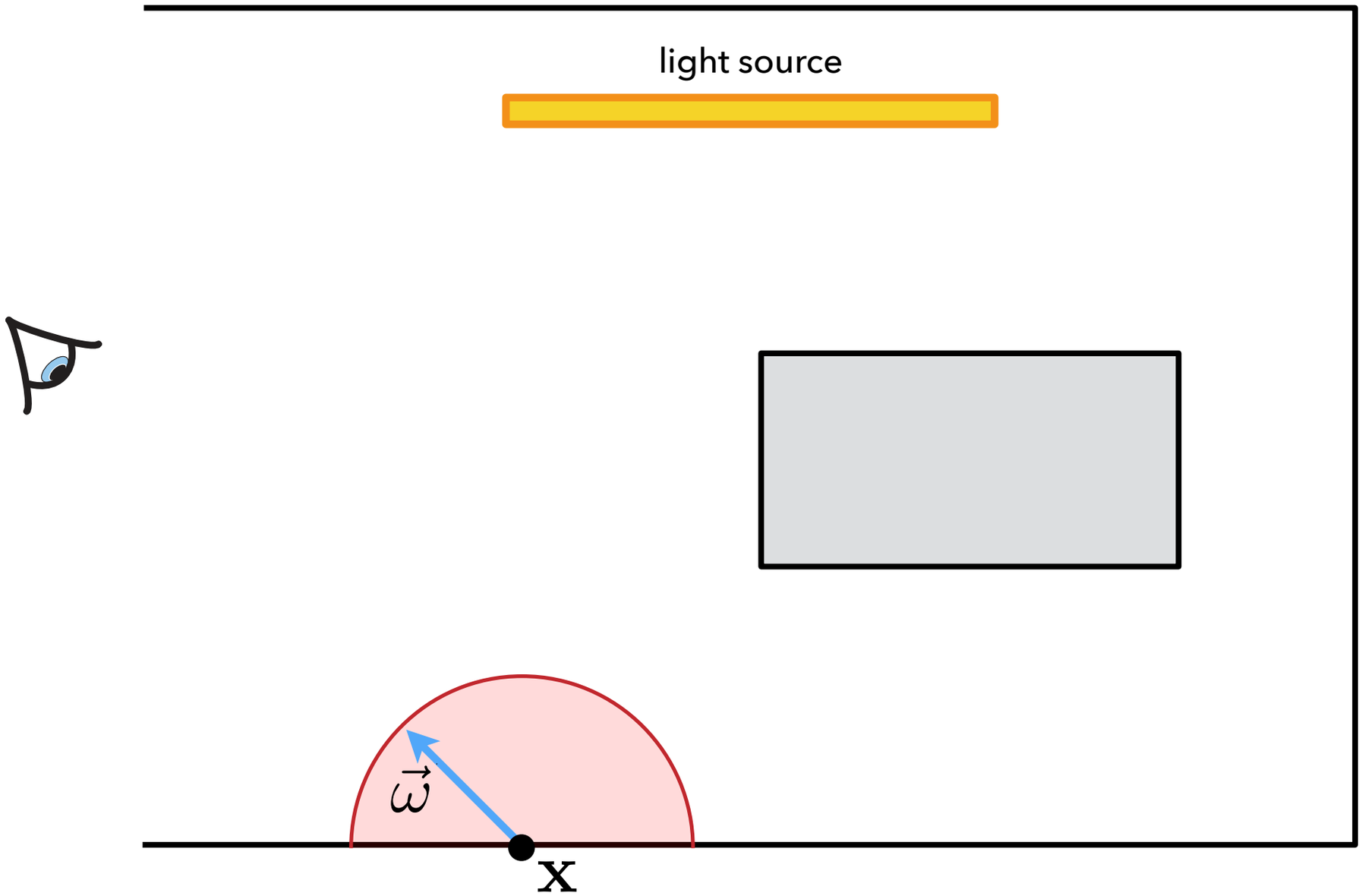
Rendering Equation

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



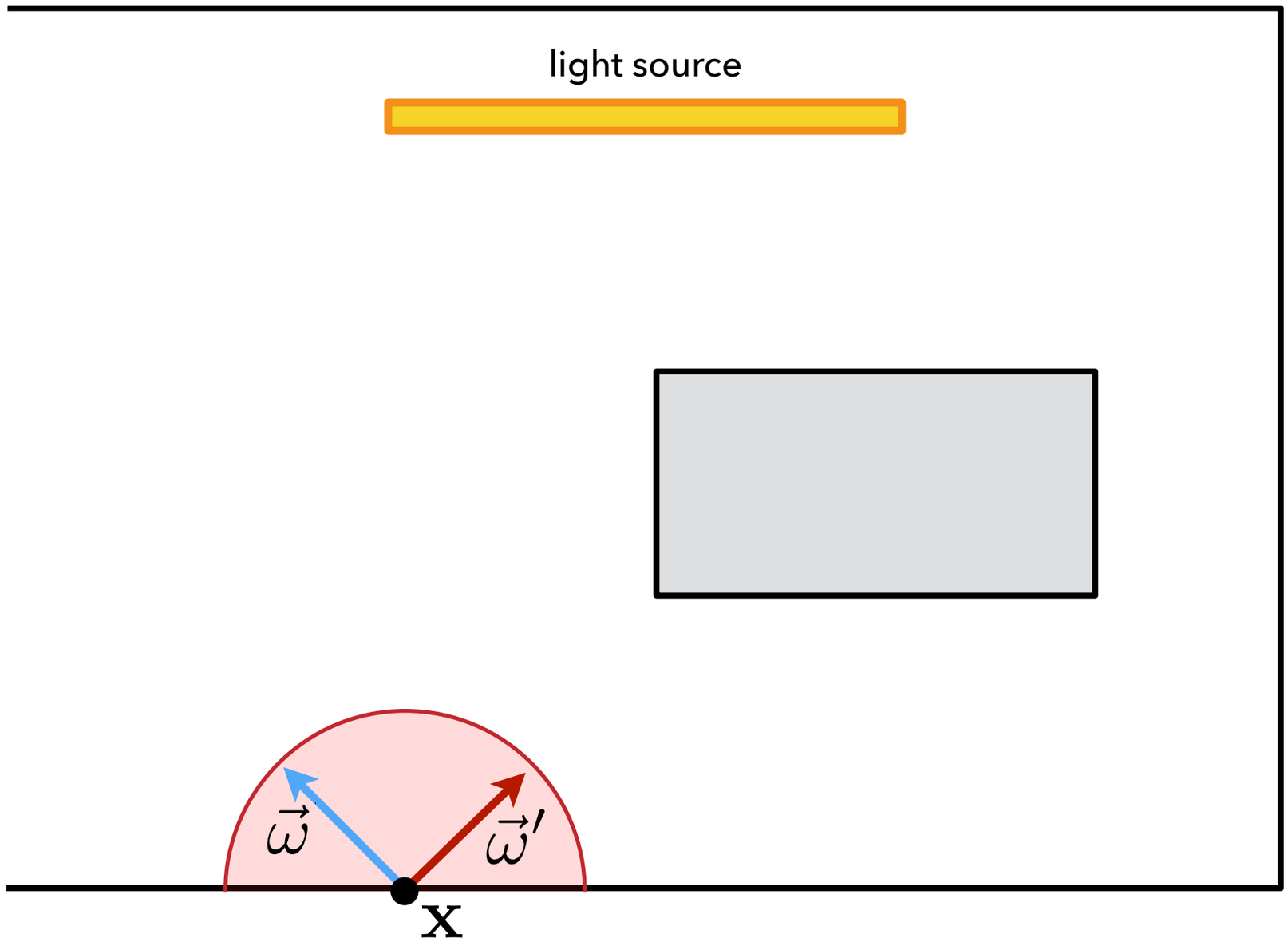
Rendering Equation

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



Rendering Equation

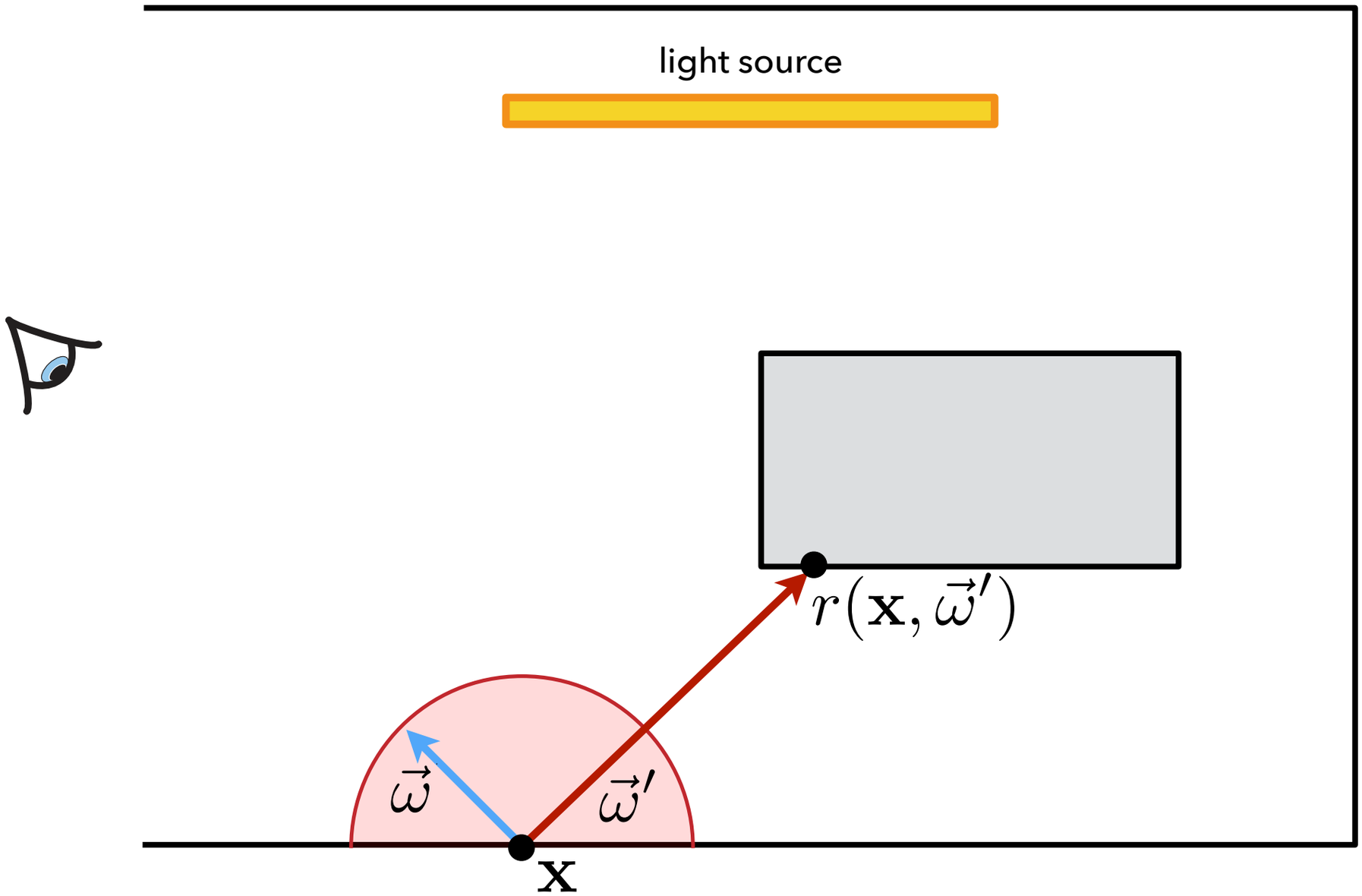
$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{H^2} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(r(\mathbf{x}, \vec{\omega}'), -\vec{\omega}') \cos\theta' \, d\vec{\omega}'$$



Rendering Equation

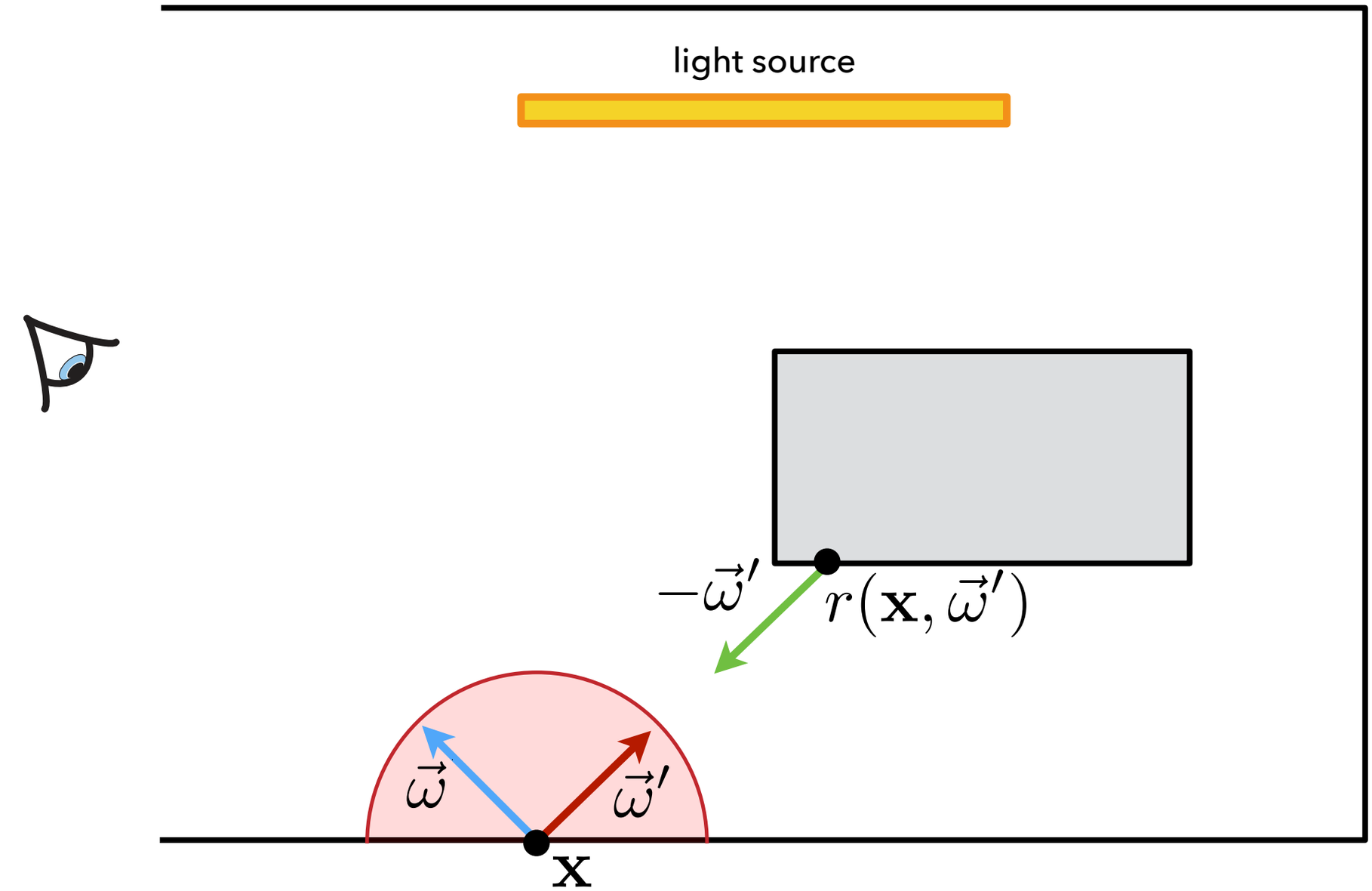
ray tracing function

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



Rendering Equation

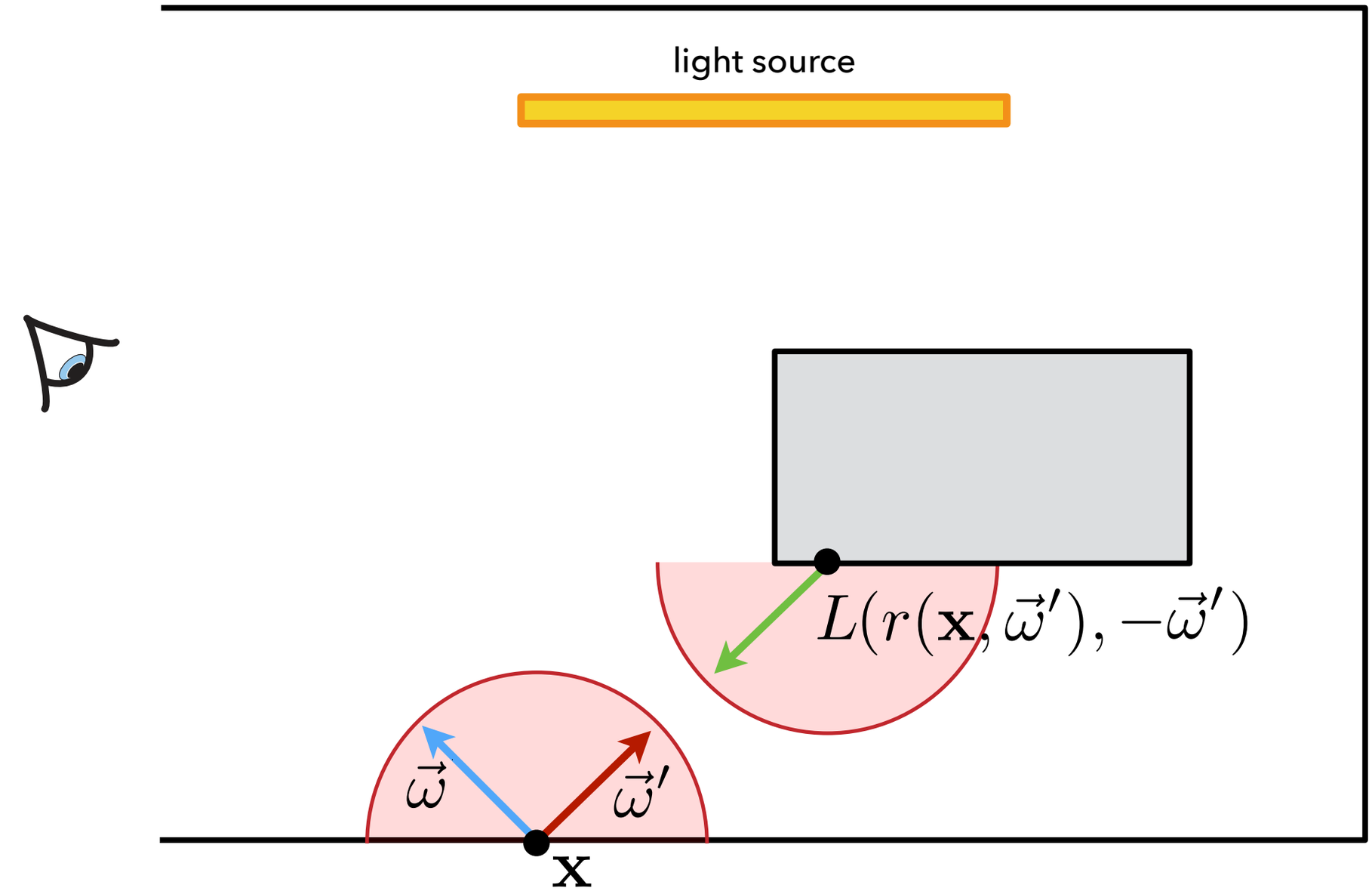
$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{H^2} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(r(\mathbf{x}, \vec{\omega}'), -\vec{\omega}') \cos\theta' d\vec{\omega}'$$



Rendering Equation

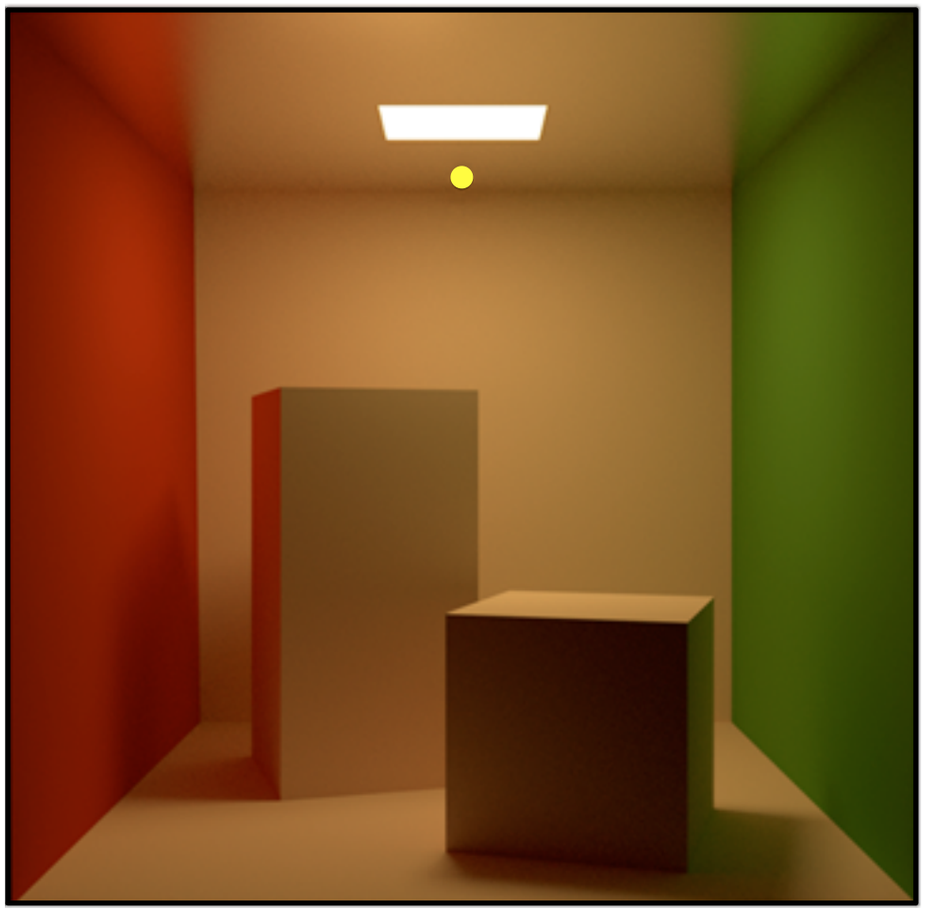
$$L(\mathbf{x},\vec{\omega}) = L_e(\mathbf{x},\vec{\omega}) + \int_{H^2} f_r(\mathbf{x},\vec{\omega}',\vec{\omega}) L(r(\mathbf{x},\vec{\omega}'),-\vec{\omega}') \cos\theta' d\vec{\omega}'$$

recursion



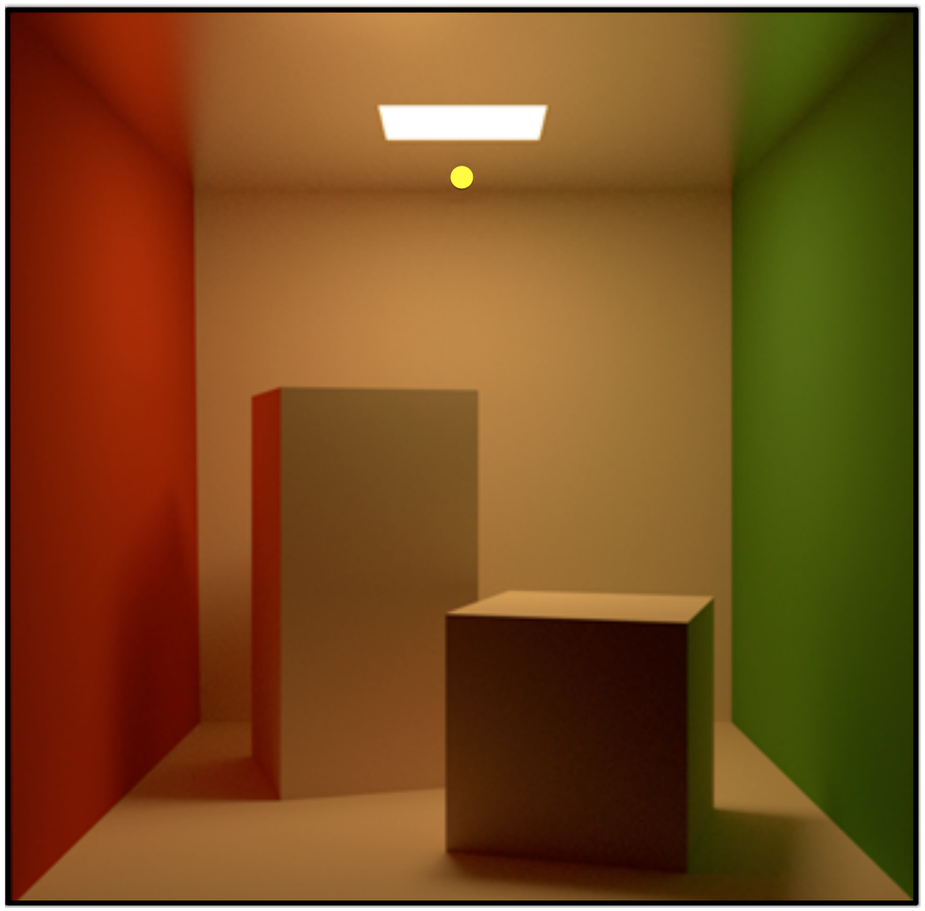
The Rendering Equation

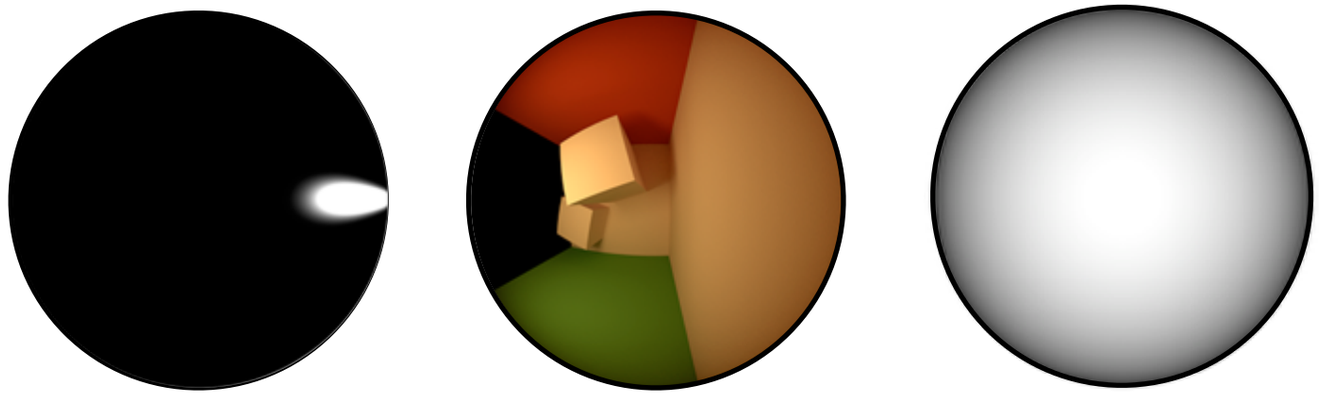
$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{H^2} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(r(\mathbf{x}, \vec{\omega}'), -\vec{\omega}') \cos\theta' d\vec{\omega}'$$



The Rendering Equation

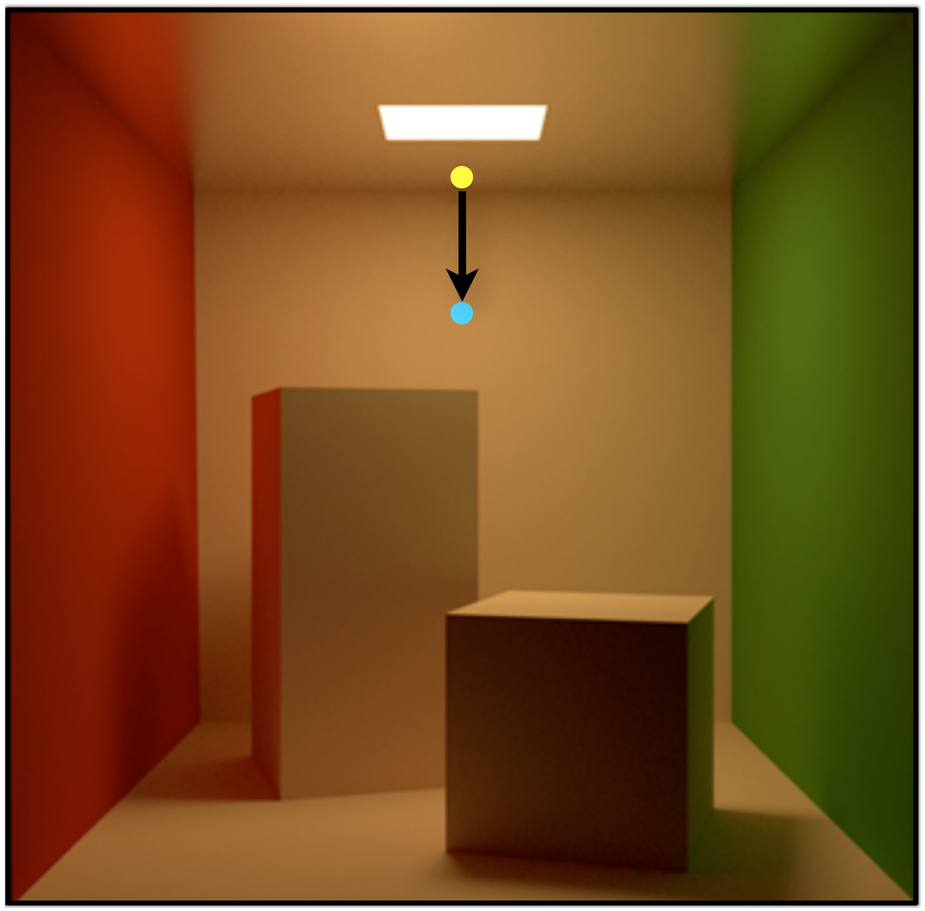
$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{H^2} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(r(\mathbf{x}, \vec{\omega}'), -\vec{\omega}') \cos\theta' \, d\vec{\omega}'$$

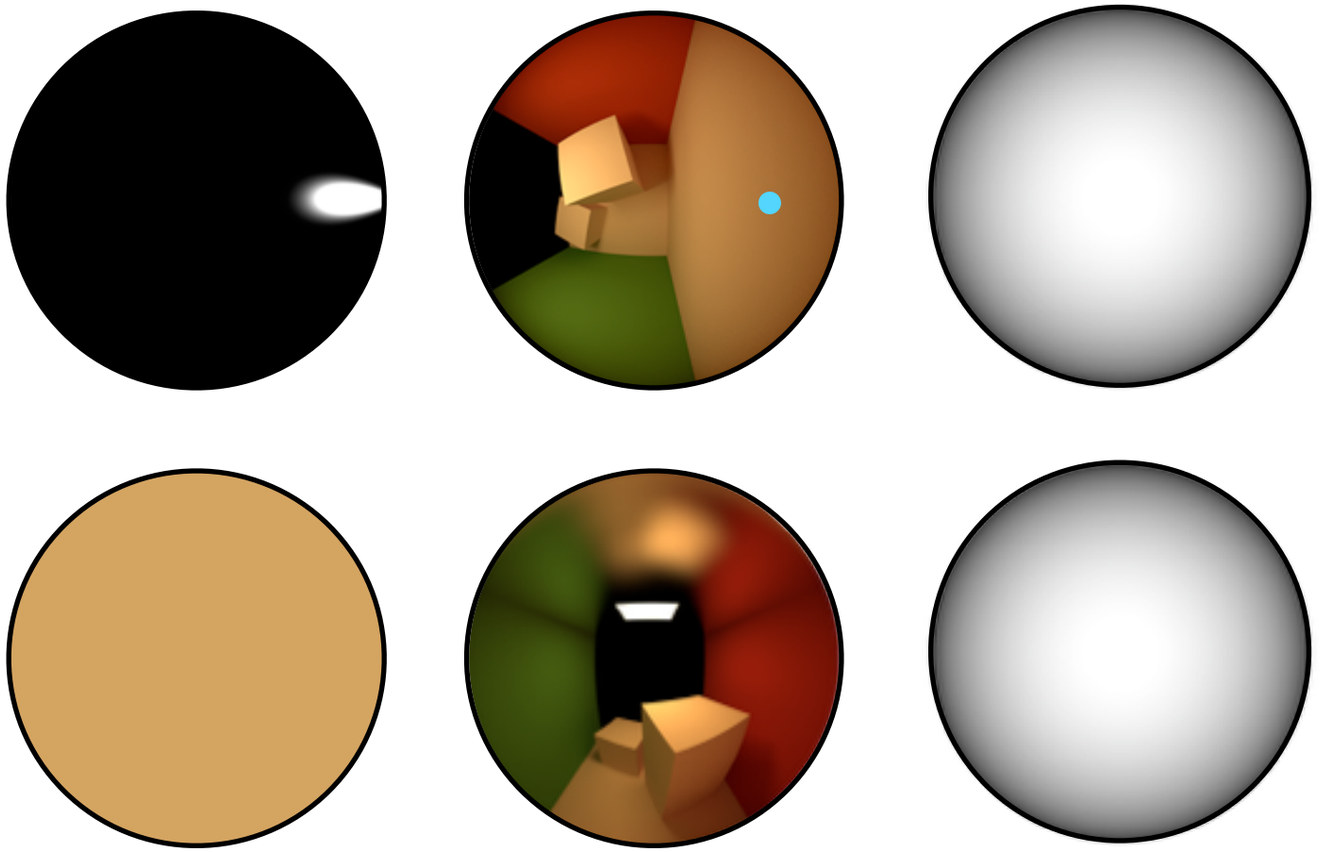




The Rendering Equation

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

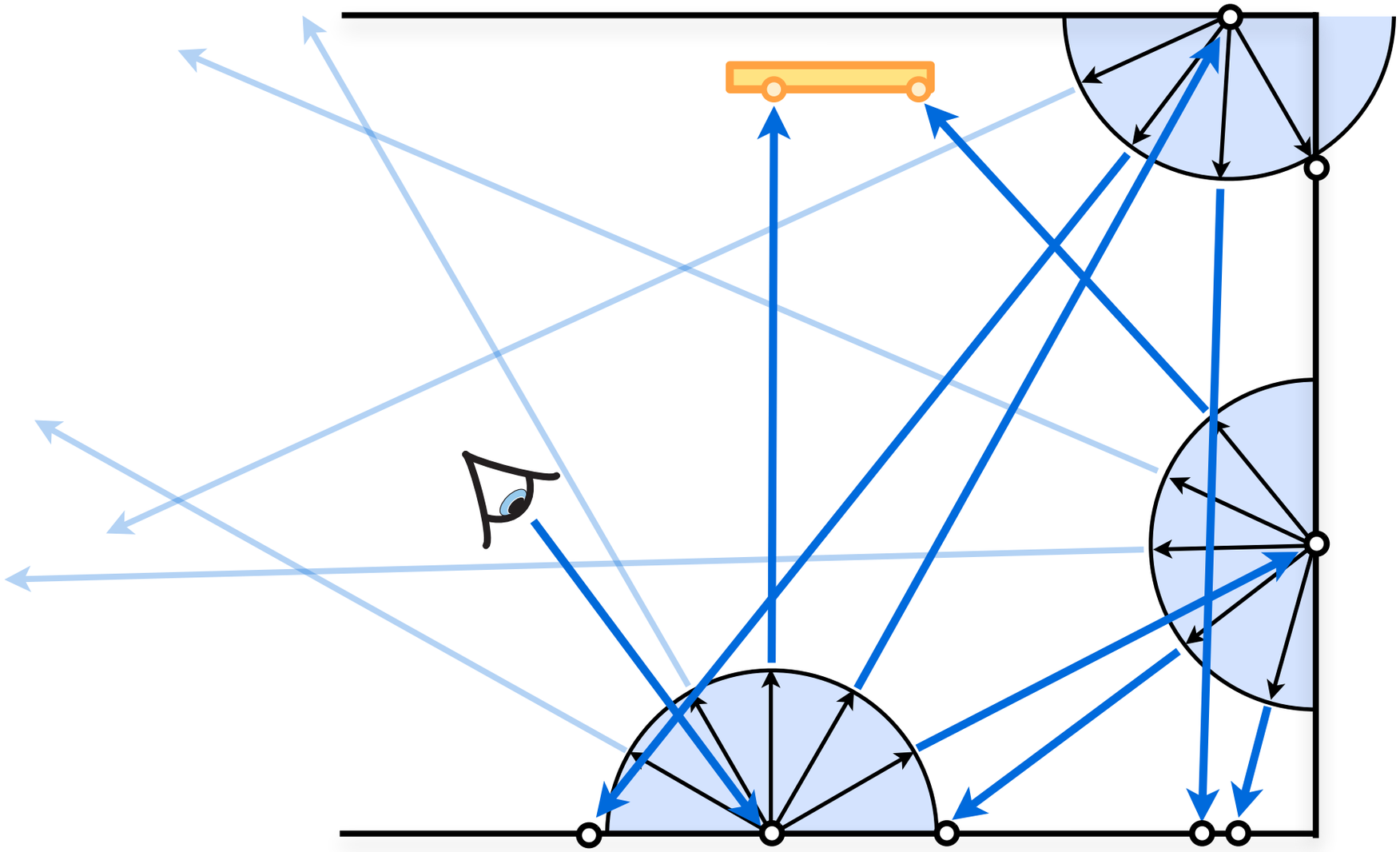




Solving the Rendering Equation

- Last time:
  - Recursive Monte Carlo ray tracing
  - Path tracing

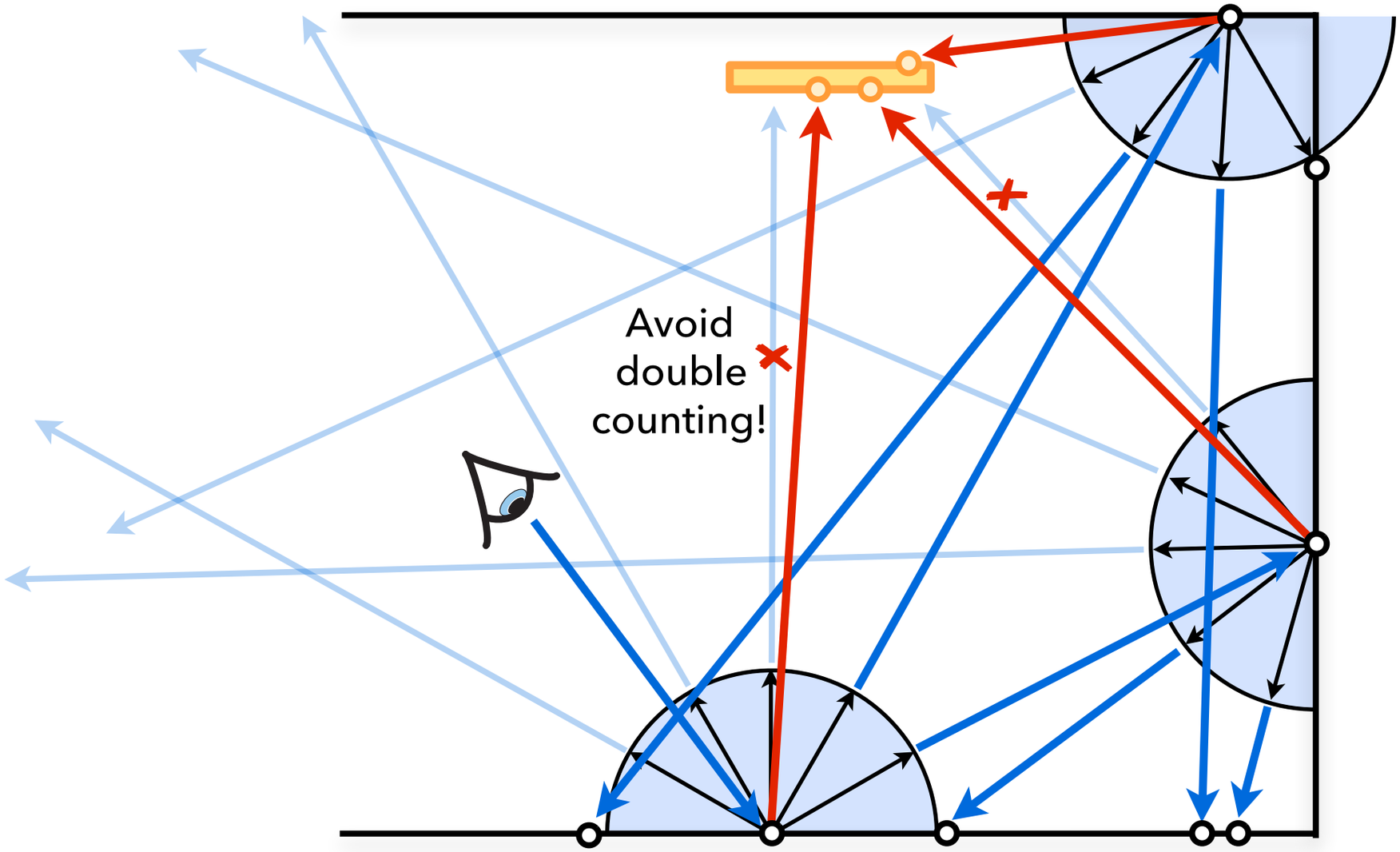
Recursive Monte Carlo Ray Tracing



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

$$L(\mathbf{x}, \vec{\omega}) \approx L_e(\mathbf{x}, \vec{\omega}) + \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(\vec{\mathbf{x}}, \vec{\omega}', \vec{\omega}) L(r(\mathbf{x}, \vec{\omega}'), -\vec{\omega}') \cos\theta'}{p(\vec{\omega}')}$$

Recursive Monte Carlo Ray Tracing



with explicit shadow rays

$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_d(\mathbf{x}, \vec{\omega}) + L_i(\mathbf{x}, \vec{\omega})$$

$$L(\mathbf{x}, \vec{\omega}) = L_e + \int_A \ldots L_e \ldots dA(\mathbf{x}') + \int_{H^2} \ldots L \ldots d\vec{\omega}'$$

Monte Carlo Ray Tracing Algorithm

$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_d(\mathbf{x}, \vec{\omega}) + L_i(\mathbf{x}, \vec{\omega})$$

```
color shade (point x, normal n)
{
    for all lights  // direct illumination
        Ld += contribution from light;

    for all N indirect sample rays  // indirect illumination
        ω' = random direction in hemisphere above n;
        Li += brdf * shade(trace(x, ω')) * dot(n, ω') / (p(ω') * N);

    if last bounce not specular  // prevent double-counting

        return Ld + Li;

    return Le + Ld + Li;
}
```

Monte Carlo Ray Tracing Algorithm

$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_d(\mathbf{x}, \vec{\omega}) + L_i(\mathbf{x}, \vec{\omega})$$

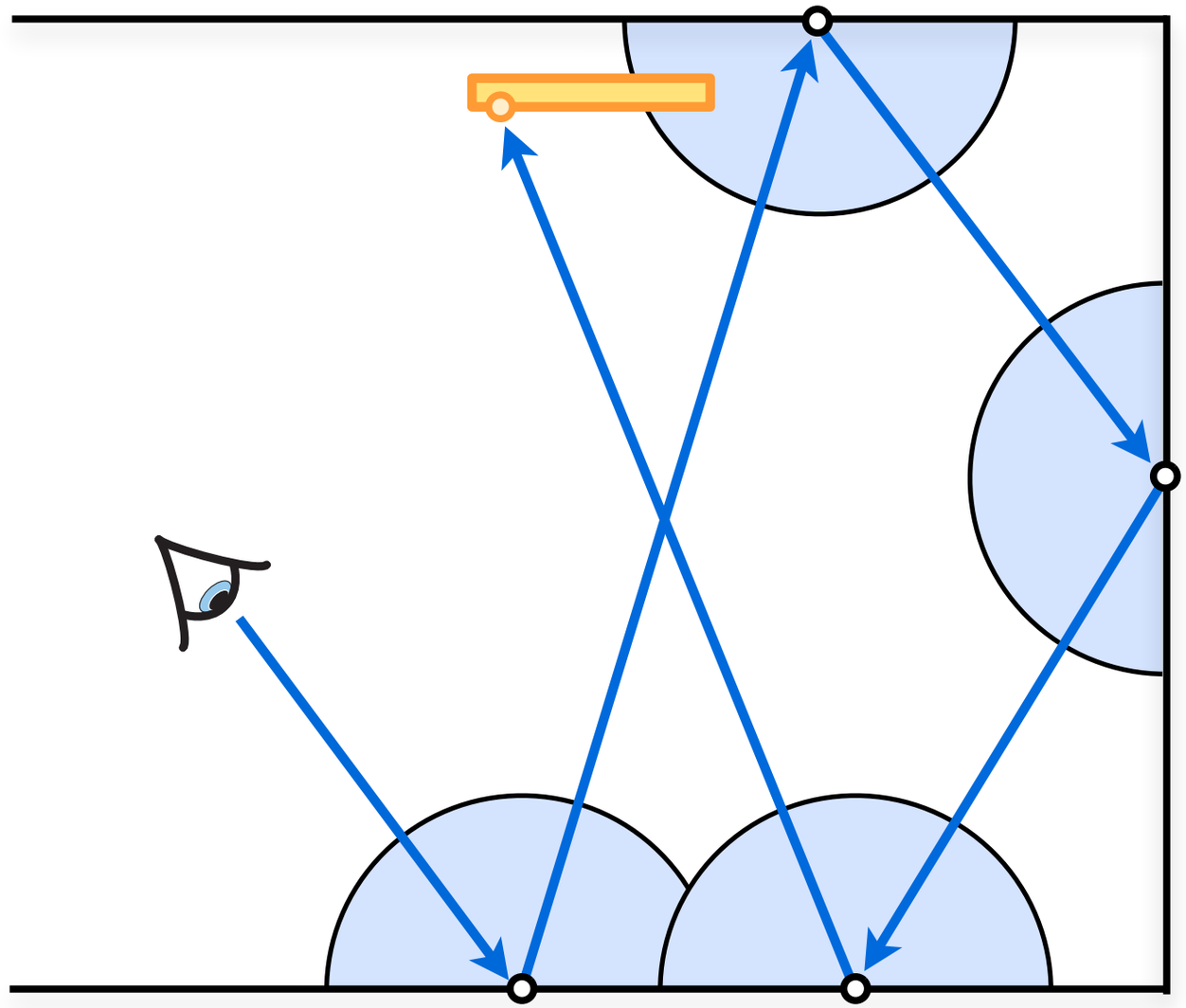
```
color shade (point x, normal n)
{
    for all lights  // direct illumination
        Ld += contribution from light;

    for all N indirect sample rays  // indirect illumination
        ω' = random direction in hemisphere above n;
        Li += brdf * shade(trace(x, ω')) * dot(n, ω') / (p(ω') * N);

    if last bounce not specular  // prevent double-counting

        return Ld + Li;

    return Le + Ld + Li;
}
```

Path Tracing Algorithm

$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_d(\mathbf{x}, \vec{\omega}) + L_i(\mathbf{x}, \vec{\omega})$$

```
color shade (point x, normal n)
{
    for all lights  // direct illumination
        Ld += contribution from light;

    // indirect illumination
    ω' = random direction in hemisphere above n;
    Li += brdf * shade(trace(x, ω')) * dot(n, ω') / (p(ω'));

    if last bounce not specular  // prevent double-counting

        return Ld + Li;

    return Le + Ld + Li;
}
```

Path Tracing

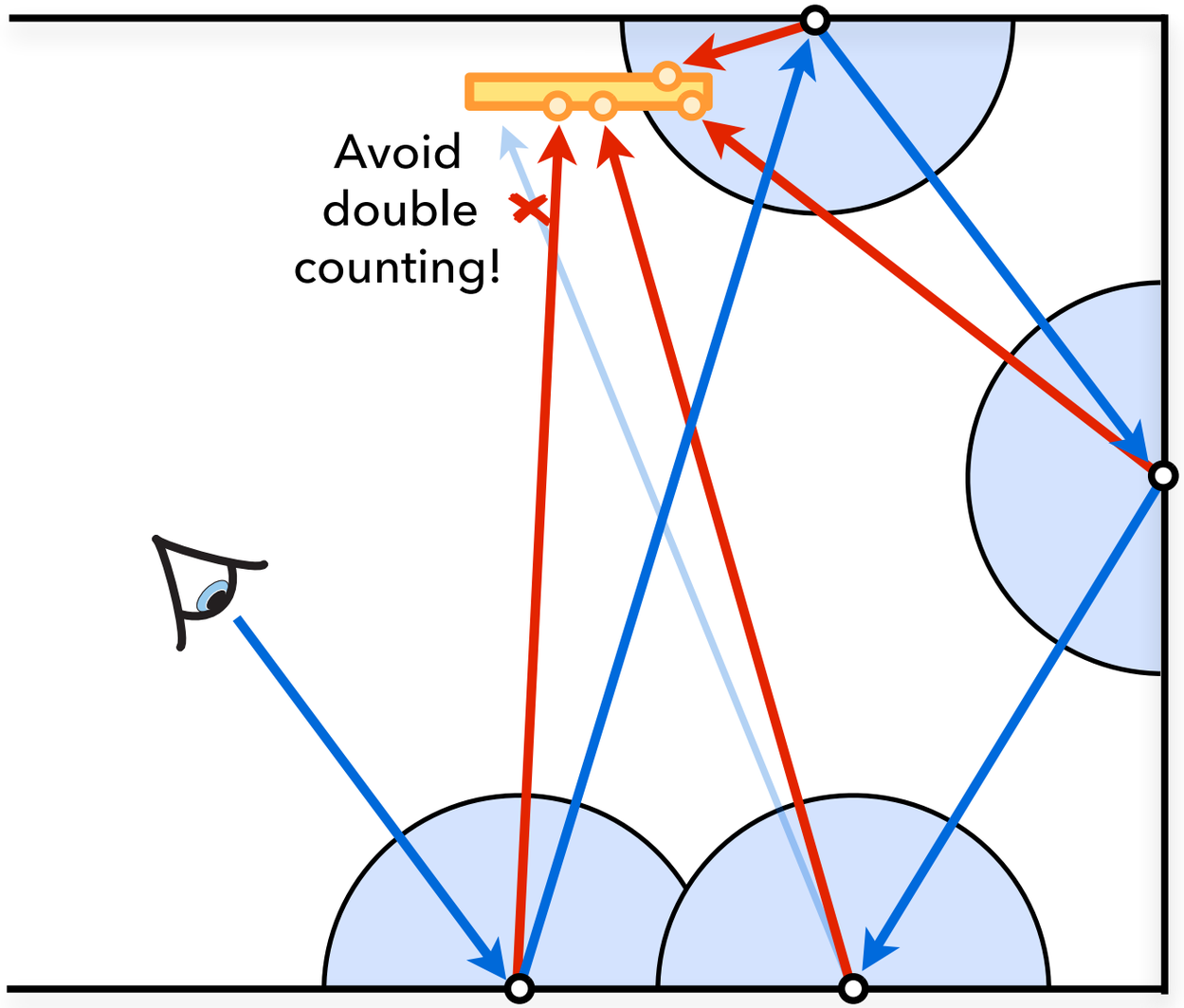


without explicit shadow rays

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

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

Path Tracing with Shadow Rays



$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_A \ldots L_e \ldots dA(\mathbf{x}') + \int_{H^2} \ldots L \ldots d\vec{\omega}'$$

Path Tracing Algorithm

$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_d(\mathbf{x}, \vec{\omega}) + L_i(\mathbf{x}, \vec{\omega})$$

```
color shade (point x, normal n)
{
    for all lights  // direct illumination
        Ld += contribution from light;

    // indirect illumination
    ω' = random direction in hemisphere above n;
    Li += brdf * shade(trace(x, ω')) * dot(n, ω') / (p(ω'));

    if last bounce not specular  // prevent double-counting

        return Ld + Li;

    return Le + Ld + Li;
}
```

Path Tracing Algorithm

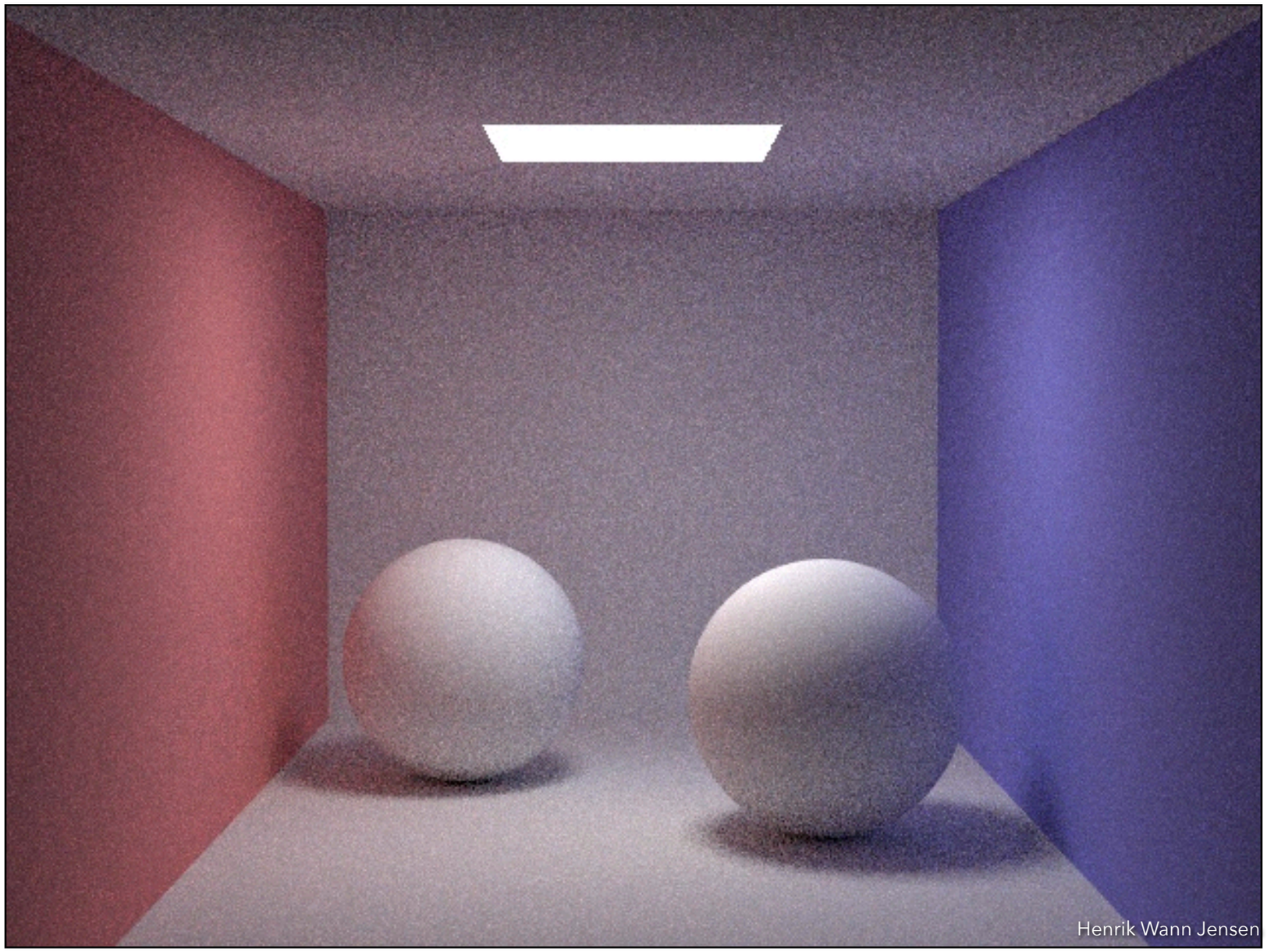
$$L(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_d(\mathbf{x}, \vec{\omega}) + L_i(\mathbf{x}, \vec{\omega})$$

```
color shade (point x, normal n)
{
    for all lights  // direct illumination
        Ld += contribution from light;

    if rand() > q  // indirect illumination
        ω' = random direction in hemisphere above n;
        Li += brdf * shade(trace(x, ω')) * dot(n, ω') / (p(ω'));

    if last bounce not specular  // prevent double-counting

        return Ld + Li / (1-q);

    return Le + Ld + Li / (1-q);
}
```

Improvements

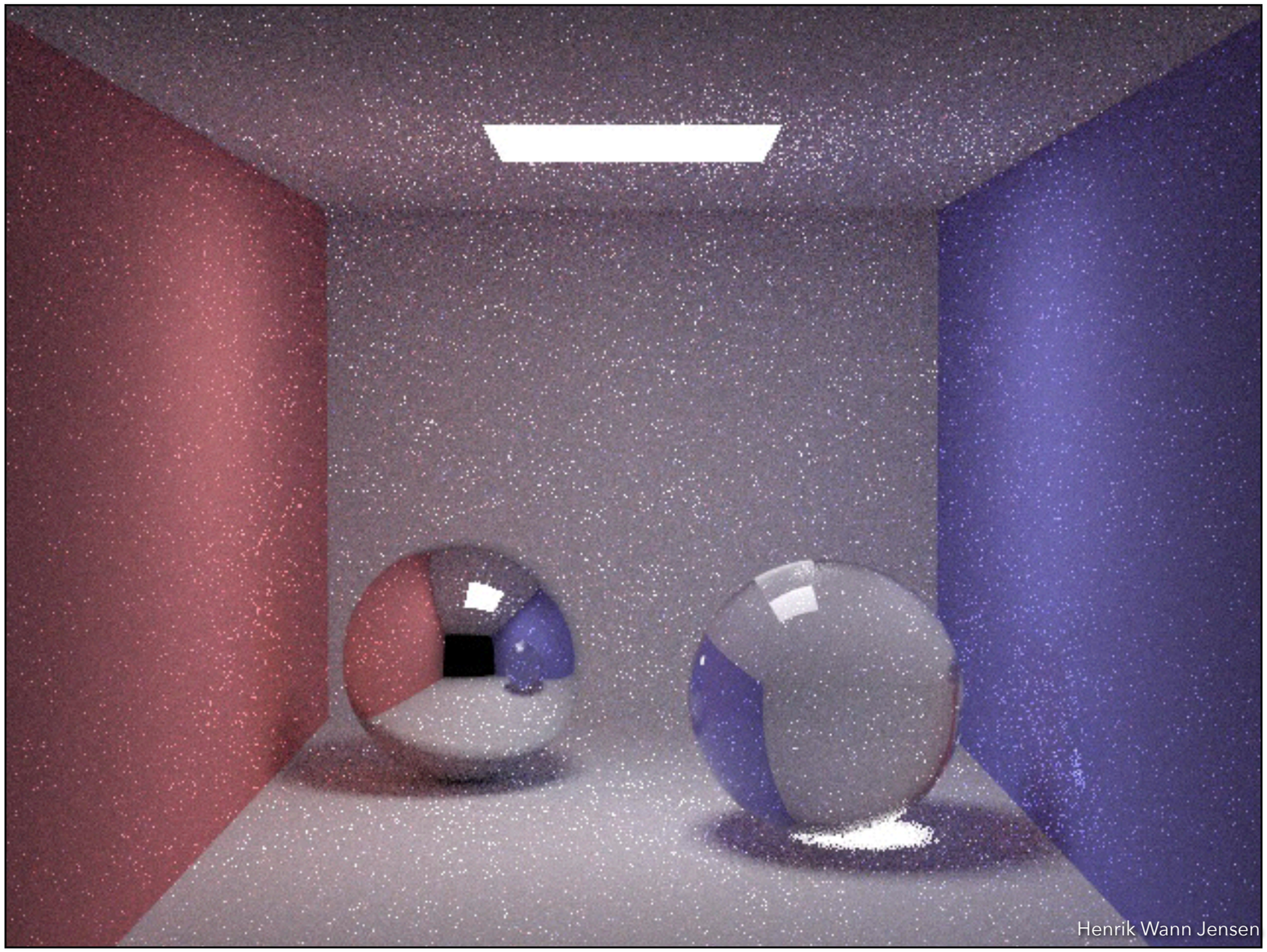
- Use MIS at least for direct illumination!
- Rewrite recursion as while loop

A Simple Scene



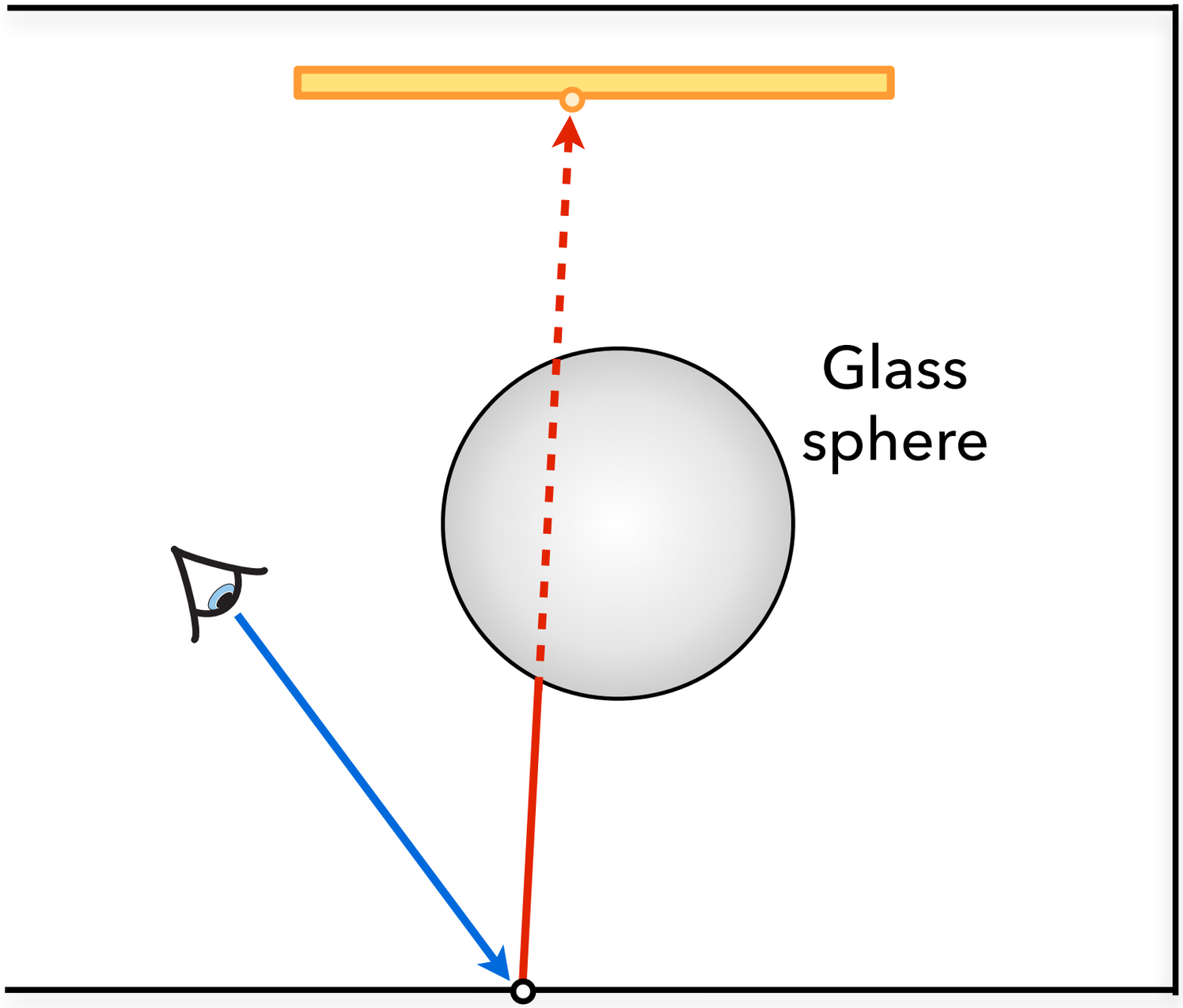
10 paths/pixel

+ Glass/Mirror Material

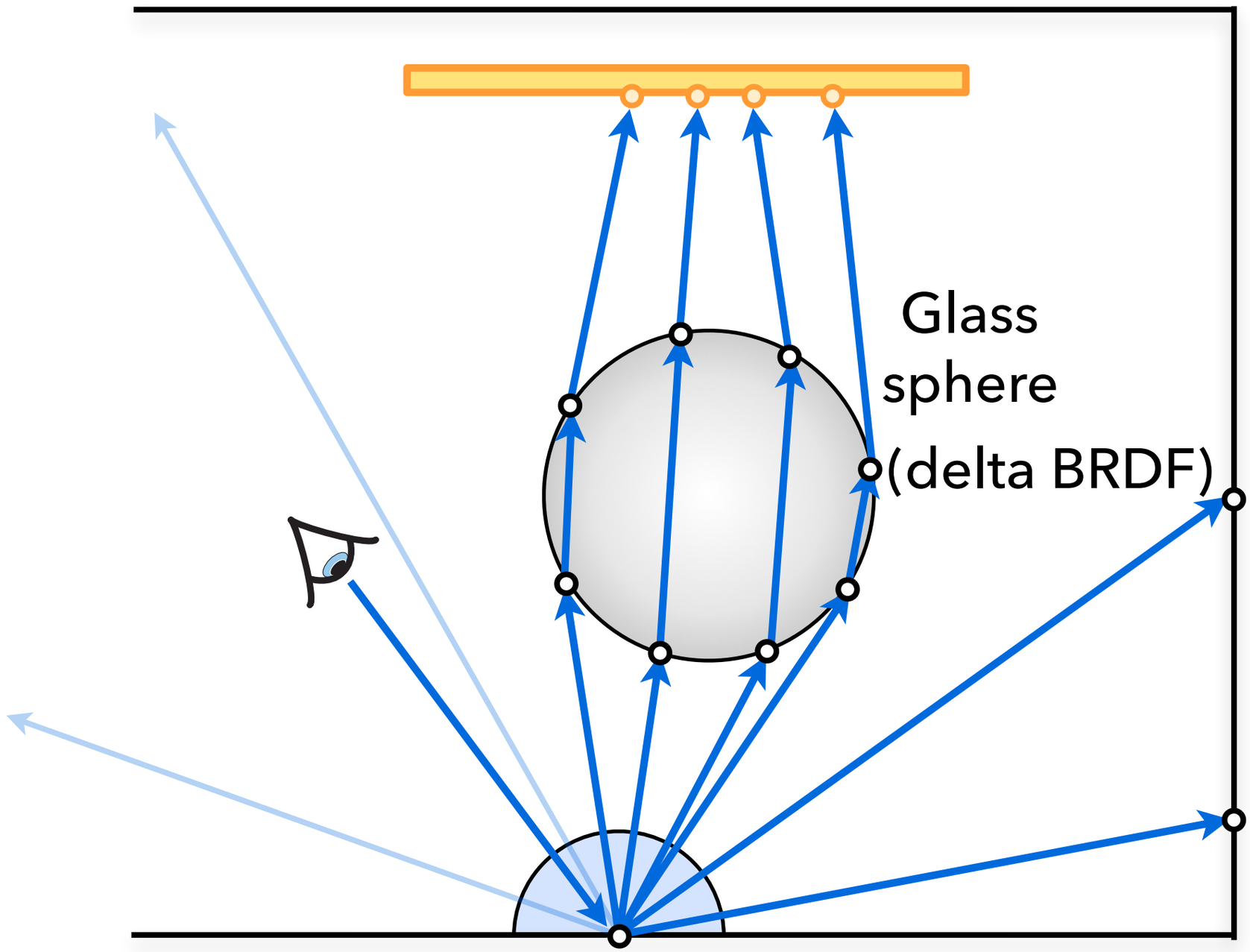


10 paths/pixel

Path Tracing Caustics

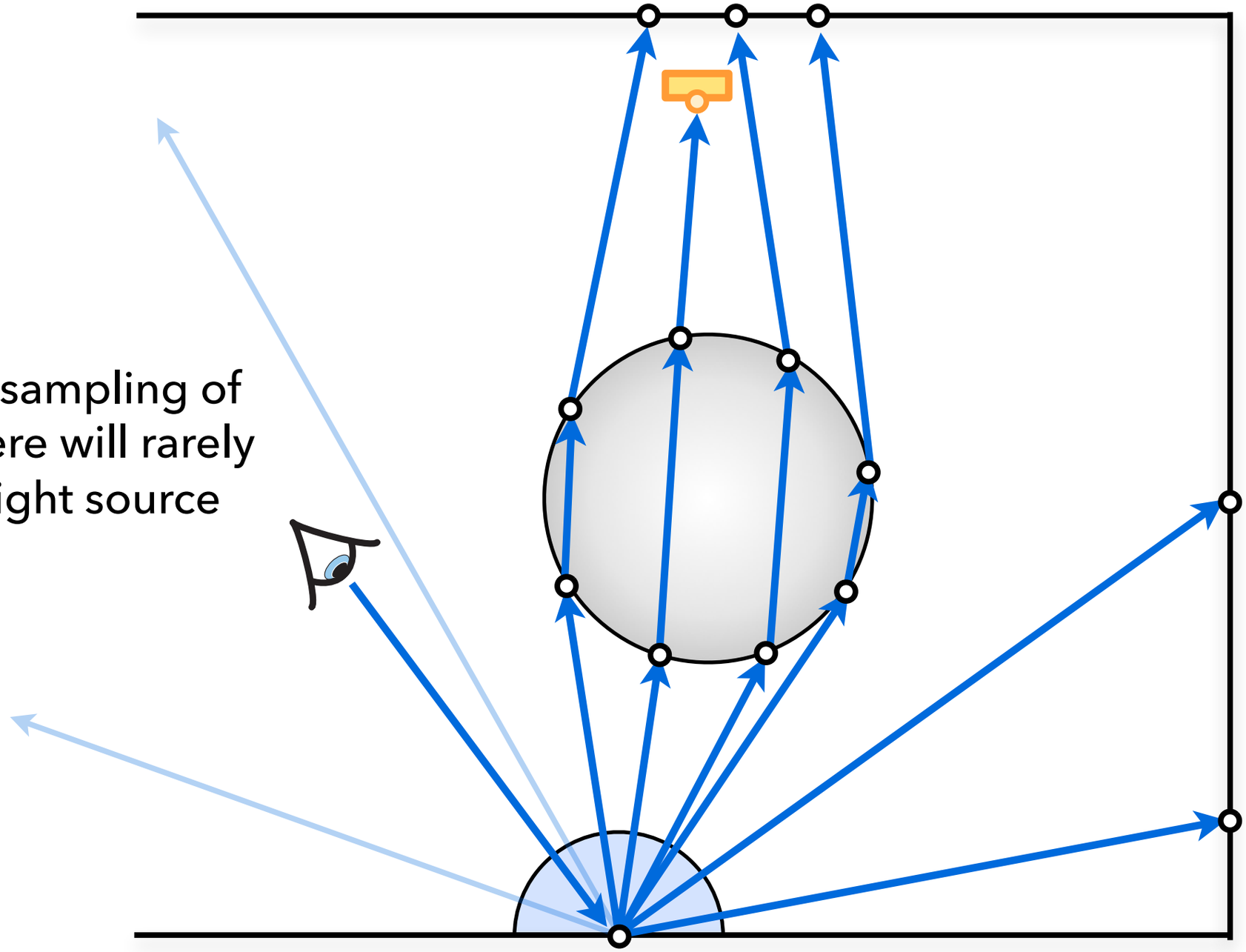


Path Tracing Caustics



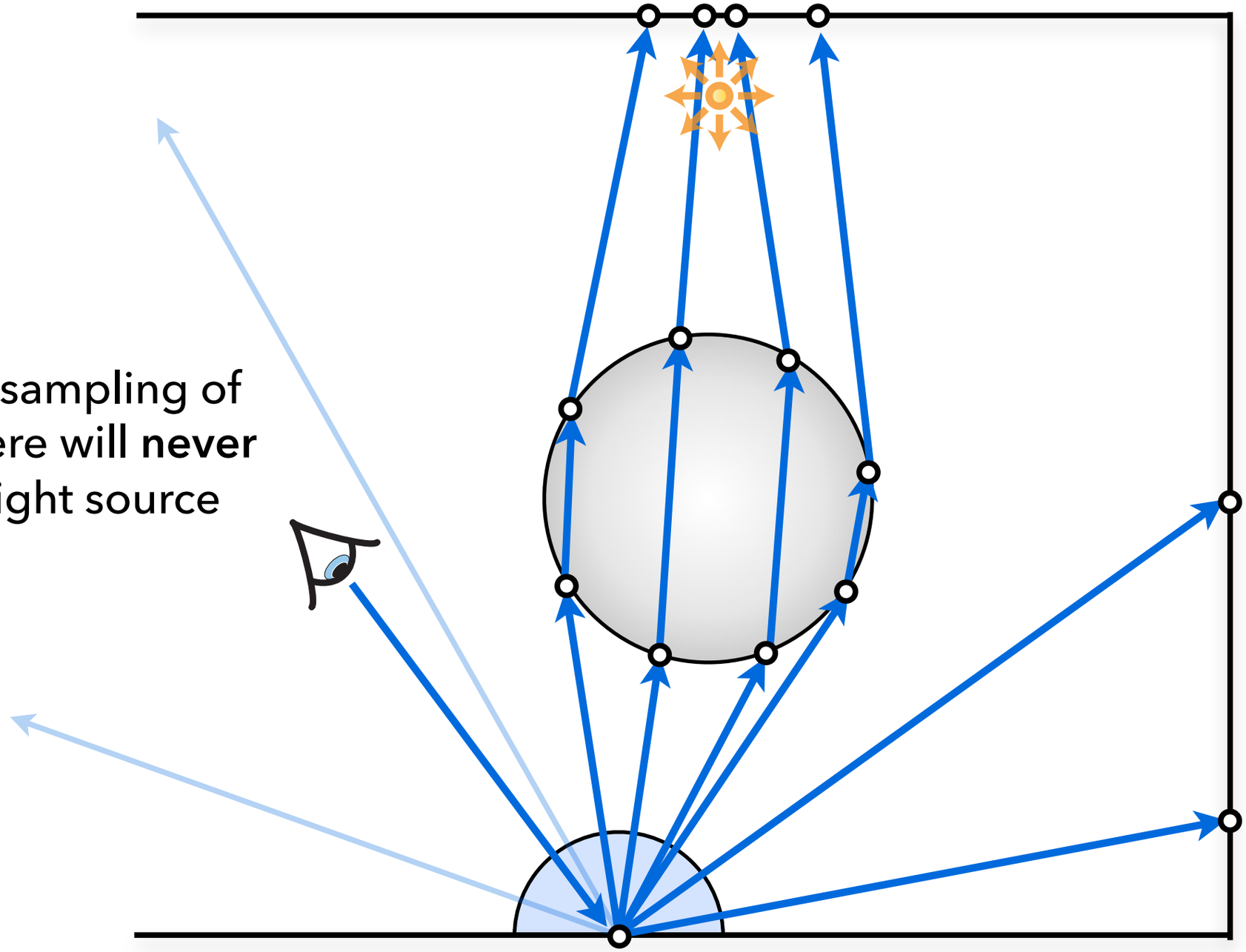
Path Tracing Caustics

Random sampling of hemisphere will rarely hit the light source



Path Tracing Caustics

Random sampling of hemisphere will **never** hit the light source



Let’s just give it more time…

- Nature ~ 2×10^{33} / second
- Fastest GPU ray tracer ~ 2×10^{8} / second

Tim Webber, Gravity VFX supervisor

Path Tracing - Summary

✓ Full solution to the rendering equation

✓ Simple to implement

✘ Slow convergence

- requires 4x more samples to half the error

✘ Robustness issues

- does not handle some light paths well (or not at all), e.g. caustics ($LS+DE$)

✘ No reuse or caching of computation

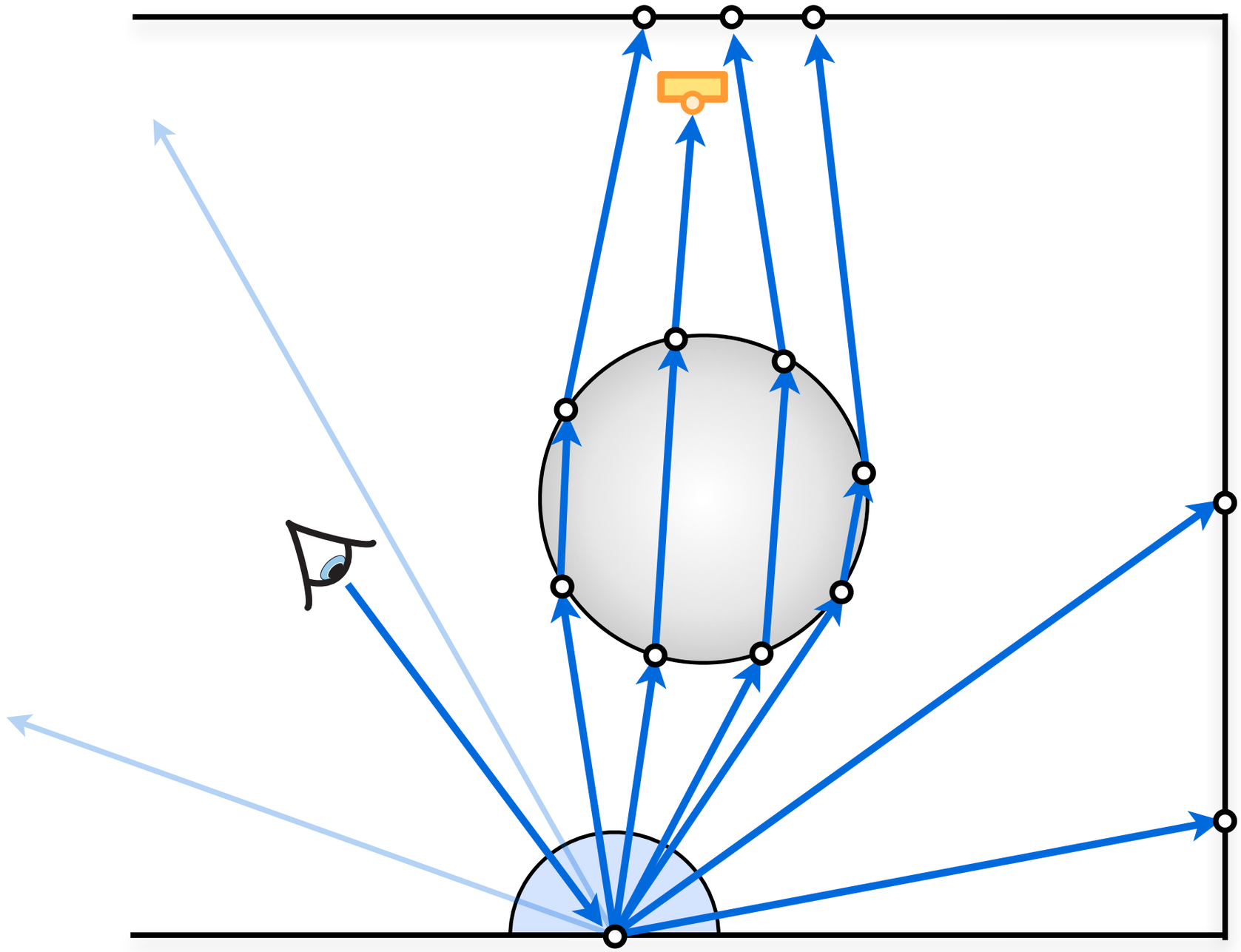
✘ General sampling issue

- makes only locally good decisions

Today's Menu

- Measurement Equation
- Path Integral Framework
- Solving the Rendering Equation
  - Light tracing
  - Bidirectional path tracing
- Operator notation

Can we simulate this better?



Visual Break

Visual Break

Visual Break

Duality of Radiance and Importance

Measurement Equation

- Rendering equation describe radiative equilibrium at point $\mathbf{x}$:

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

- We are interested in the total radiance contributing to pixel j:

$$I_j = \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}$$

response of the sensor at film location $\mathbf{x}$ to radiance arriving from direction $\vec{\omega}$ (often referred to as *emitted importance*)

Radiance vs. Importance

- Radiance
  - emitted from light sources
  - describes *amount of light* traveling within a differential beam

- Importance
  - "emitted" from sensors
  - describes the *response of the sensor* to radiance traveling within a differential beam

Duality of Radiance & Importance

$$I_j = \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}$$

Duality of Radiance & Importance

$$I_j = \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}$$

$$= \int_{A_{\text{film}}} \int_{A} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) L_o(\mathbf{y}, \mathbf{x}) \, d\mathbf{y} d\mathbf{x}$$

outgoing quantities

Let's expand L_o and consider direct illumination only

Duality of Radiance & Importance

$$I_j = \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}$$

$$= \int_{A_{\text{film}}} \int_{A} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) L_o(\mathbf{y}, \mathbf{x}) \, d\mathbf{y} d\mathbf{x}$$

$$= \int_{A_{\text{film}}} \int_{A} \int_{A_{\text{light}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, \mathbf{z}, \mathbf{x}) G(\mathbf{y}, \mathbf{z}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{z} d\mathbf{y} d\mathbf{x}$$

emitted quantities with identical measure

Let's swap the inner and outer integral

Duality of Radiance & Importance

$$I_j = \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}$$

$$= \int_{A_{\text{film}}} \int_{A} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) L_o(\mathbf{y}, \mathbf{x}) \, d\mathbf{y} d\mathbf{x}$$

$$= \int_{A_{\text{film}}} \int_{A} \int_{A_{\text{light}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, \mathbf{z}, \mathbf{x}) G(\mathbf{y}, \mathbf{z}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{z} d\mathbf{y} d\mathbf{x}$$

$$= \int_{A_{\text{light}}} \int_{A} \int_{A_{\text{film}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, \mathbf{z}, \mathbf{x}) G(\mathbf{y}, \mathbf{z}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{x} d\mathbf{y} d\mathbf{z}$$

symmetric functions

Duality of Radiance & Importance

$$I_j = \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}$$

$$= \int_{A_{\text{film}}} \int_{A} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) L_o(\mathbf{y}, \mathbf{x}) \, d\mathbf{y} d\mathbf{x}$$

$$= \int_{A_{\text{film}}} \int_{A} \int_{A_{\text{light}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, \mathbf{z}, \mathbf{x}) G(\mathbf{y}, \mathbf{z}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{z} d\mathbf{y} d\mathbf{x}$$

$$= \int_{A_{\text{light}}} \int_{A} \int_{A_{\text{film}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{y}, \mathbf{x}) f(\mathbf{y}, \mathbf{x}, \mathbf{z}) G(\mathbf{z}, \mathbf{y}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{x} d\mathbf{y} d\mathbf{z}$$

symmetric functions

Duality of Radiance & Importance

$$
\begin{aligned}
I_j &= \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x} \\
&= \int_{A_{\text{film}}} \int_{A} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) L_o(\mathbf{y}, \mathbf{x}) \, d\mathbf{y} d\mathbf{x} \\
&= \int_{A_{\text{film}}} \int_{A} \int_{A_{\text{light}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, \mathbf{z}, \mathbf{x}) G(\mathbf{y}, \mathbf{z}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{z} d\mathbf{y} d\mathbf{x} \\
&= \int_{A_{\text{light}}} \int_{A} \int_{A_{\text{film}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{y}, \mathbf{x}) f(\mathbf{y}, \mathbf{x}, \mathbf{z}) G(\mathbf{z}, \mathbf{y}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{x} d\mathbf{y} d\mathbf{z} \\
&= \int_{A_{\text{light}}} \int_{A} W_o(\mathbf{y}, \mathbf{z}) G(\mathbf{z}, \mathbf{y}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{y} d\mathbf{z}
\end{aligned}
$$

Duality of Radiance & Importance

$$
\begin{aligned}
I_j &= \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x} \\
&= \int_{A_{\text{film}}} \int_{A} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) L_o(\mathbf{y}, \mathbf{x}) \, d\mathbf{y} d\mathbf{x} \\
&= \int_{A_{\text{film}}} \int_{A} \int_{A_{\text{light}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}, \mathbf{z}, \mathbf{x}) G(\mathbf{y}, \mathbf{z}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{z} d\mathbf{y} d\mathbf{x} \\
&= \int_{A_{\text{light}}} \int_{A} \int_{A_{\text{film}}} W_e(\mathbf{x}, \mathbf{y}) G(\mathbf{y}, \mathbf{x}) f(\mathbf{y}, \mathbf{x}, \mathbf{z}) G(\mathbf{z}, \mathbf{y}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{x} d\mathbf{y} d\mathbf{z} \\
&= \int_{A_{\text{light}}} \int_{A} W_o(\mathbf{y}, \mathbf{z}) G(\mathbf{z}, \mathbf{y}) L_e(\mathbf{z}, \mathbf{y}) \, d\mathbf{y} d\mathbf{z} \\
&= \int_{A_{\text{light}}} \int_{H^2} W_i(\mathbf{z}, \vec{\omega}) L_e(\mathbf{z}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{z}
\end{aligned}
$$

Duality of Radiance & Importance

emitted *importance*

incident *radiance*

$$I_j = \int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}$$

$$= \int_{A_{\text{light}}} \int_{H^2} W_i(\mathbf{z}, \vec{\omega}) L_e(\mathbf{z}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{z}$$

emitted *radiance*

incident *importance*

Duality of Radiance & Importance

Path tracing

start from ***film***, search for ***radiance*** at light

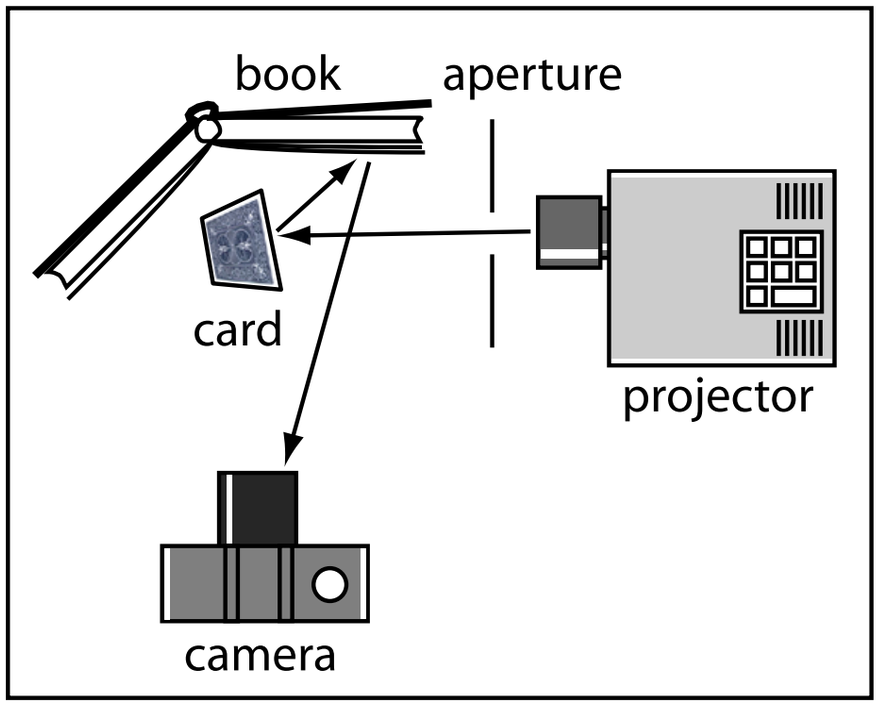
$$I_j = \overbrace{\int_{A_{\text{film}}} \int_{H^2} W_e(\mathbf{x}, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{x}}$$

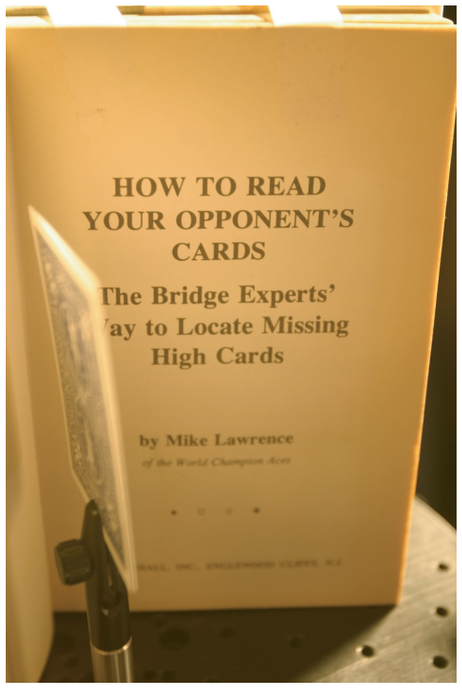
$$= \underbrace{\int_{A_{\text{light}}} \int_{H^2} W_i(\mathbf{z}, \vec{\omega}) L_e(\mathbf{z}, \vec{\omega}) \cos\theta \, d\vec{\omega} d\mathbf{z}}$$

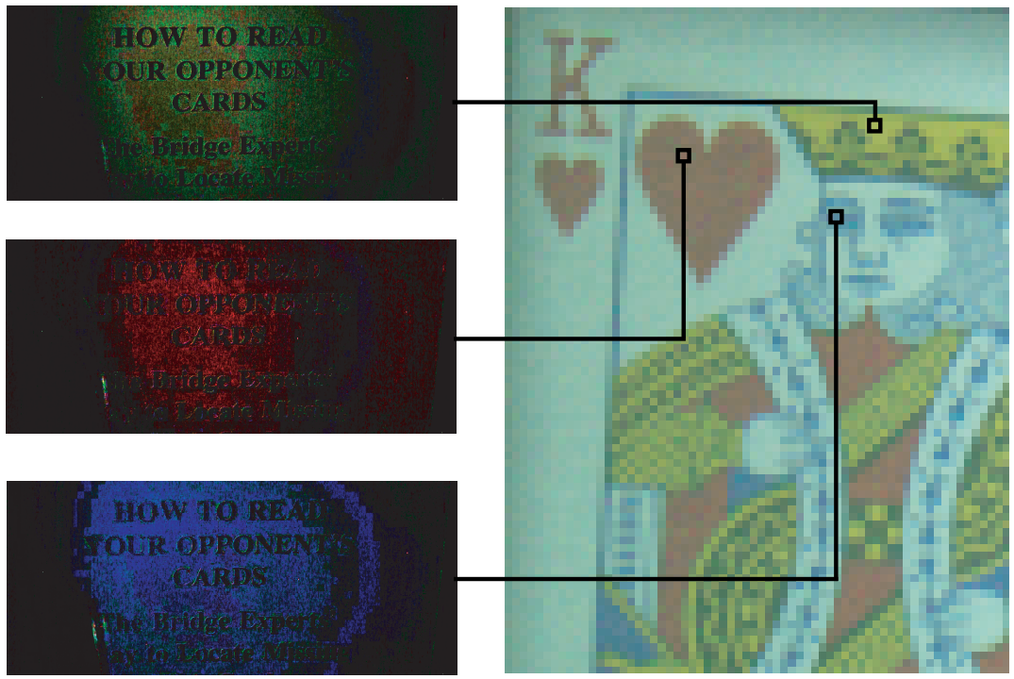
Light tracing

start from ***light***, search for ***importance*** at sensor

Light transport is symmetric







Dual Photography [Sen et al. 2005]

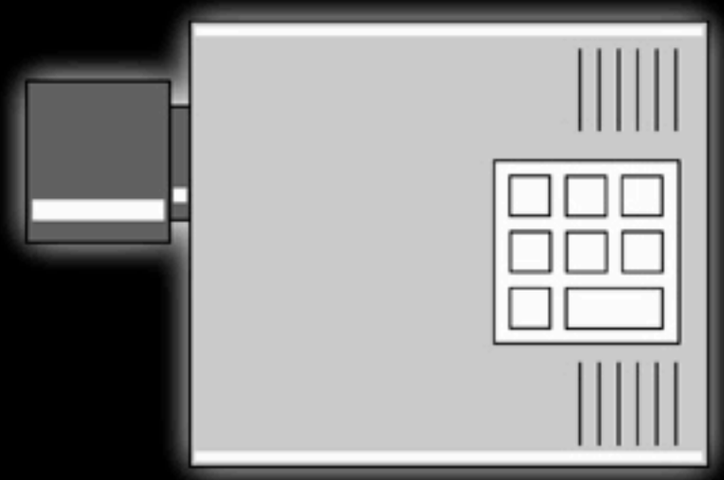
Dual Photography

Pradeep Sen* Billy Chen* Gaurav Garg* Stephen R. Marschner†
Mark Horowitz* Marc Levoy* Hendrik P.A. Lensch*

*Stanford University †Cornell University

SIGGRAPH2005

card



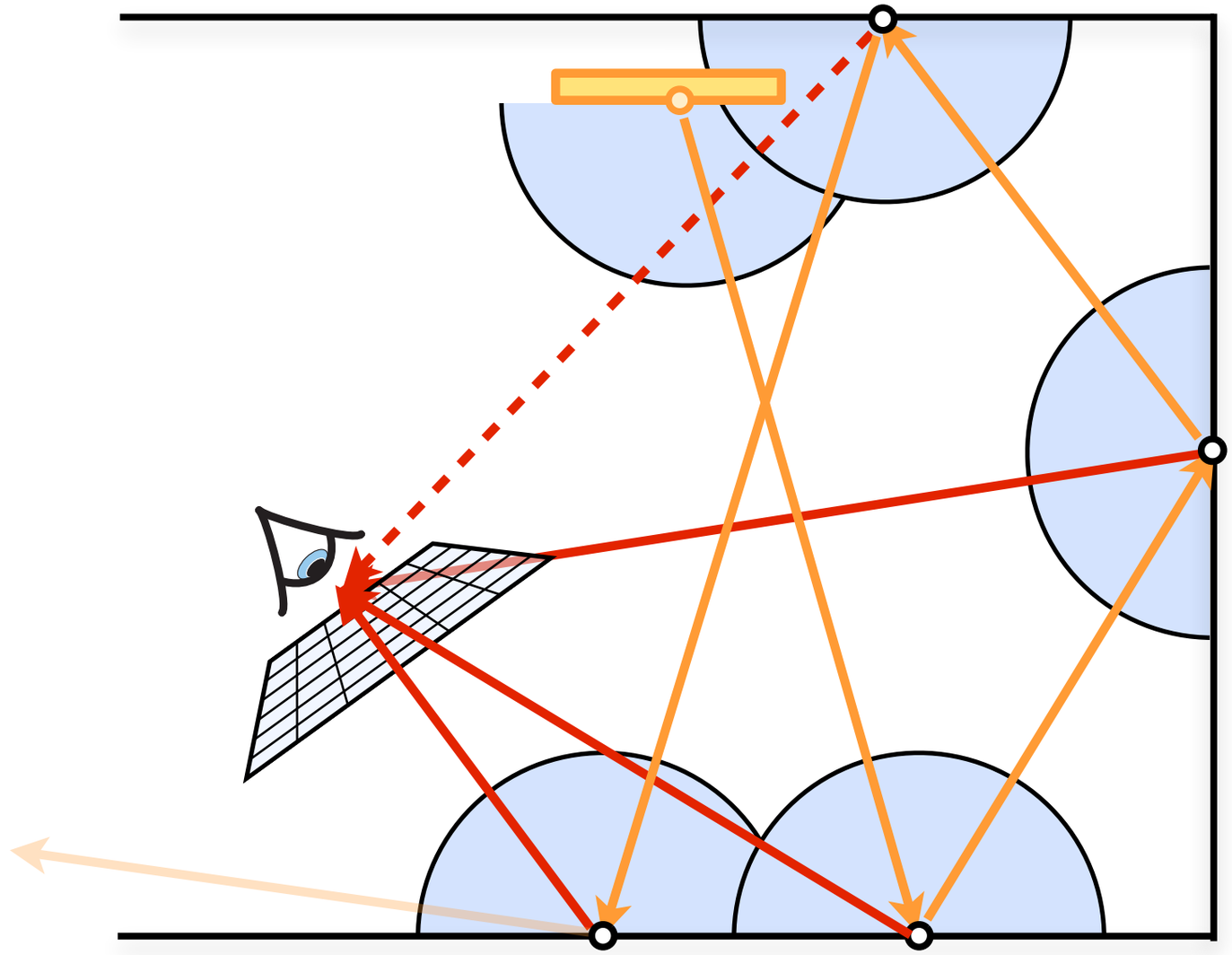
projector

Light Tracing

Light Tracing

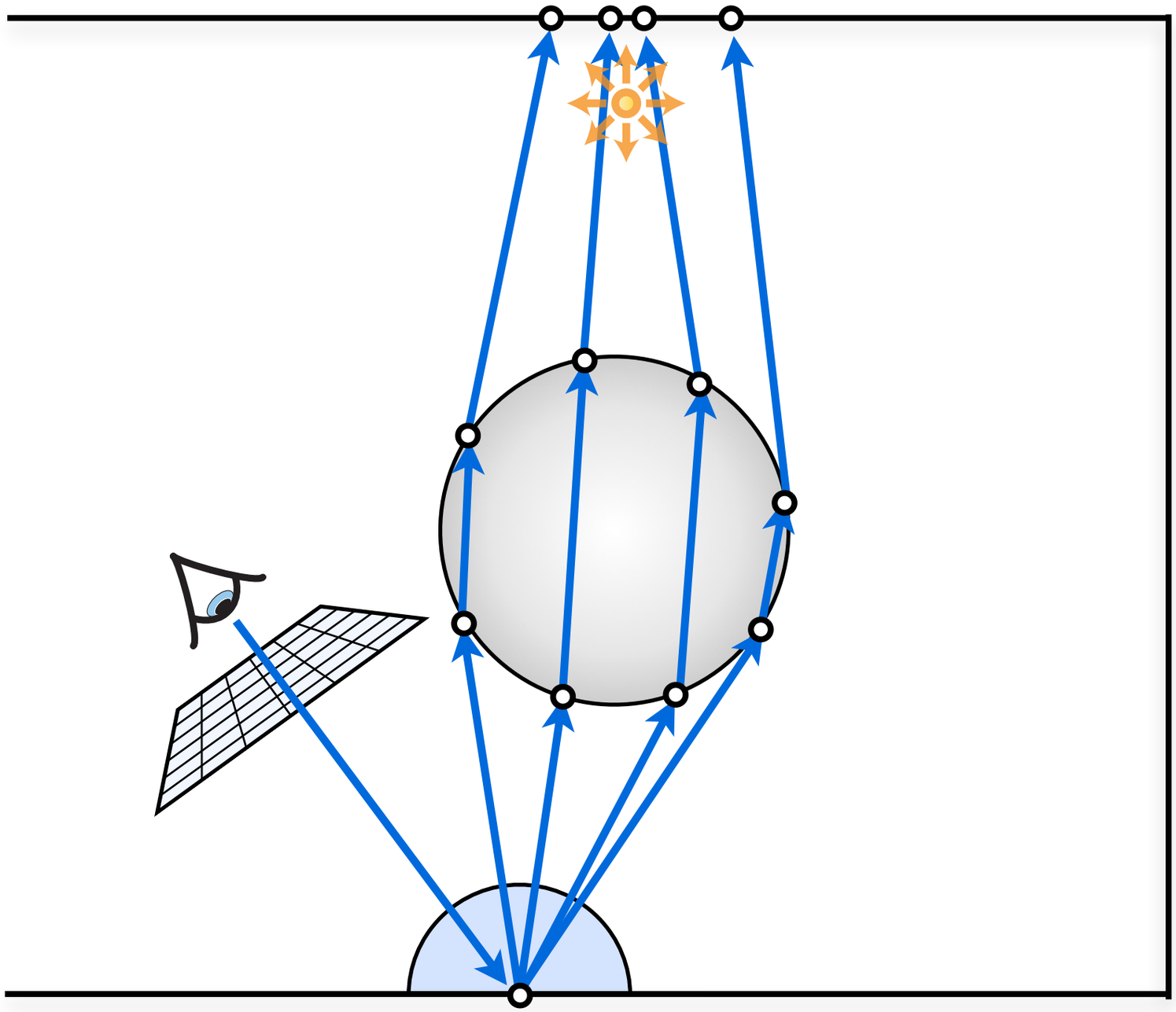
- Shoot multiple paths from light sources
- Connect to the image using next-event estimation (a.k.a. shadow rays in PT)

Light Tracing with NEE

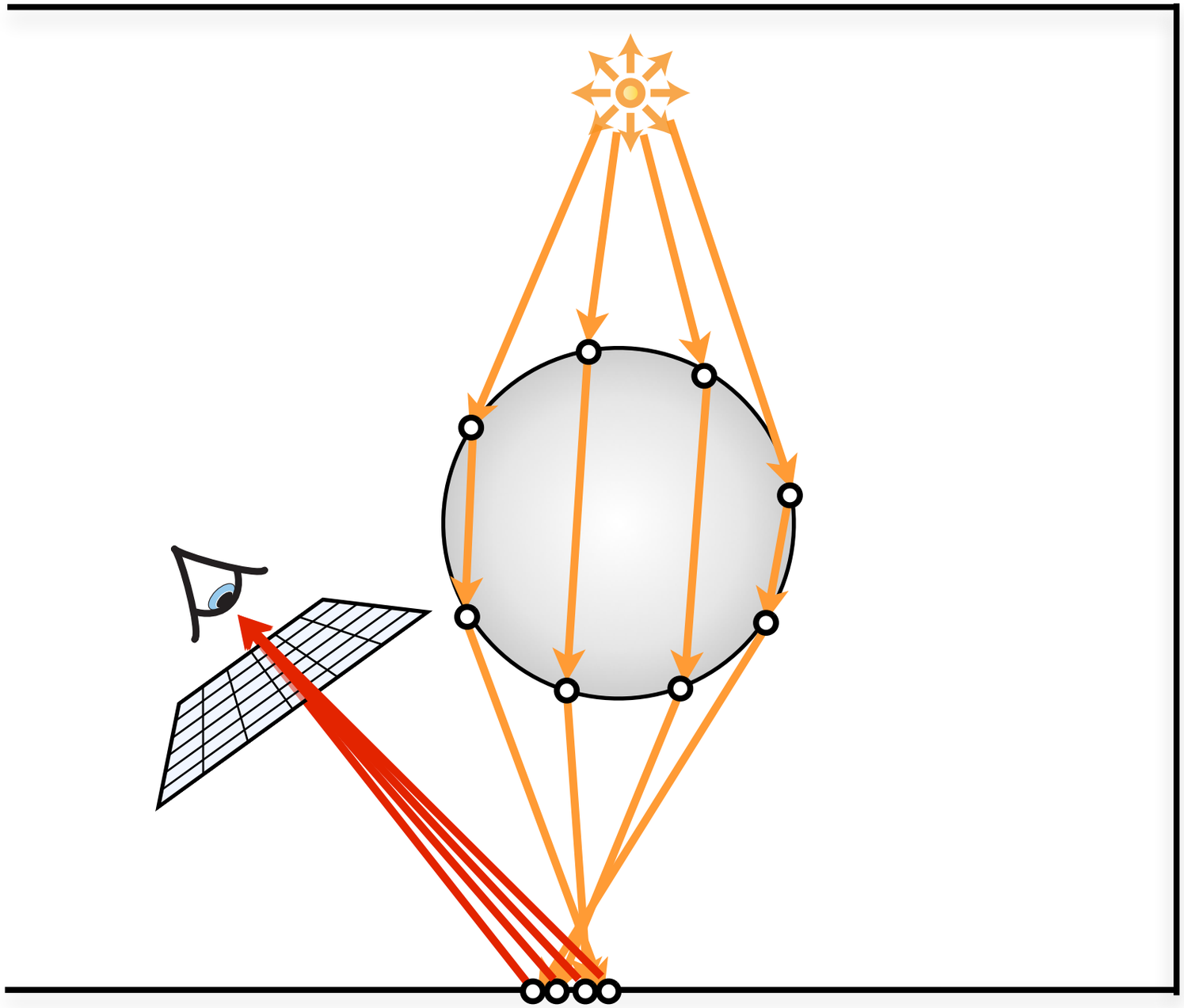


Splat to the image at each vertex

Path Tracing Caustics

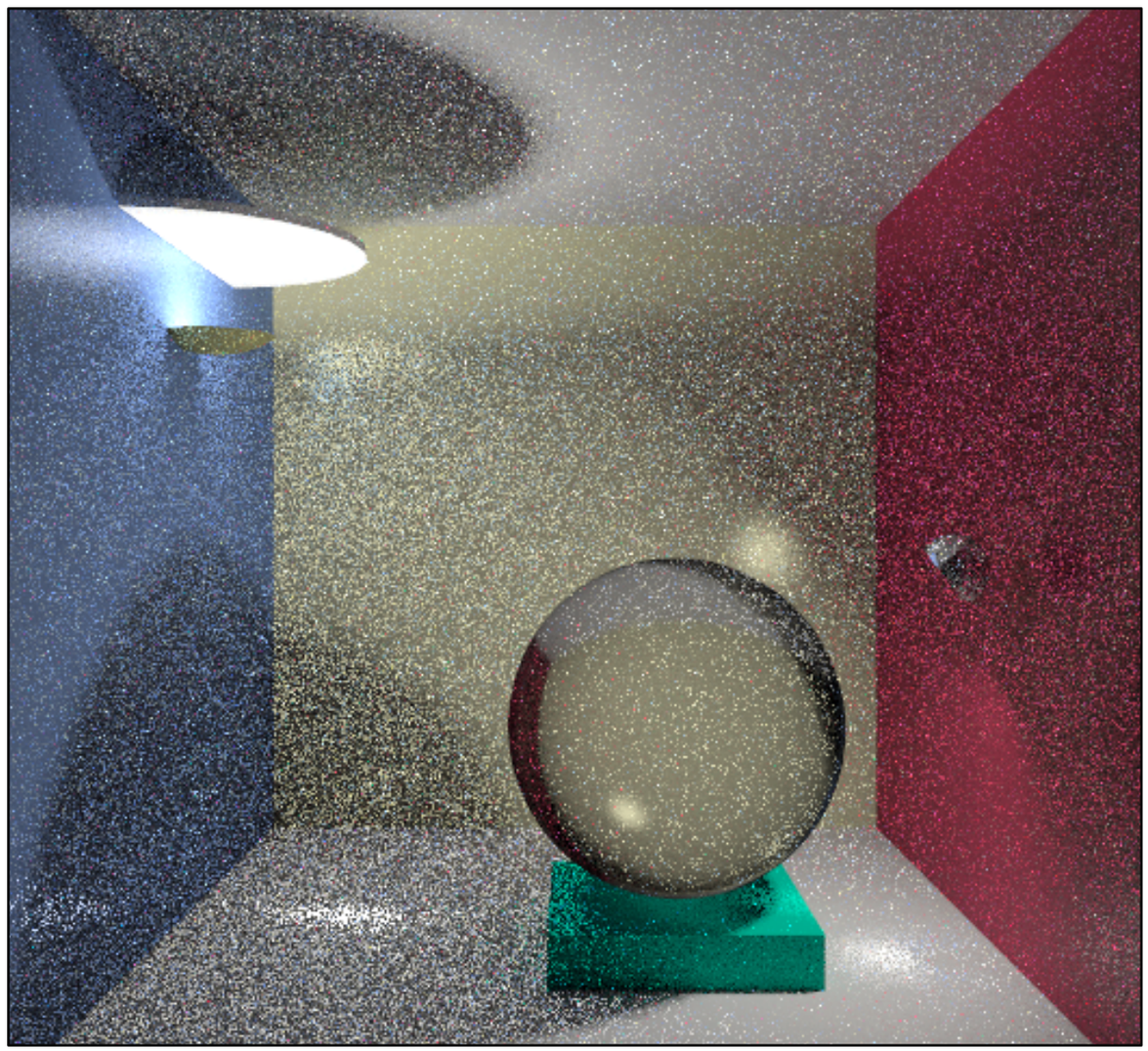


Light Tracing Caustics

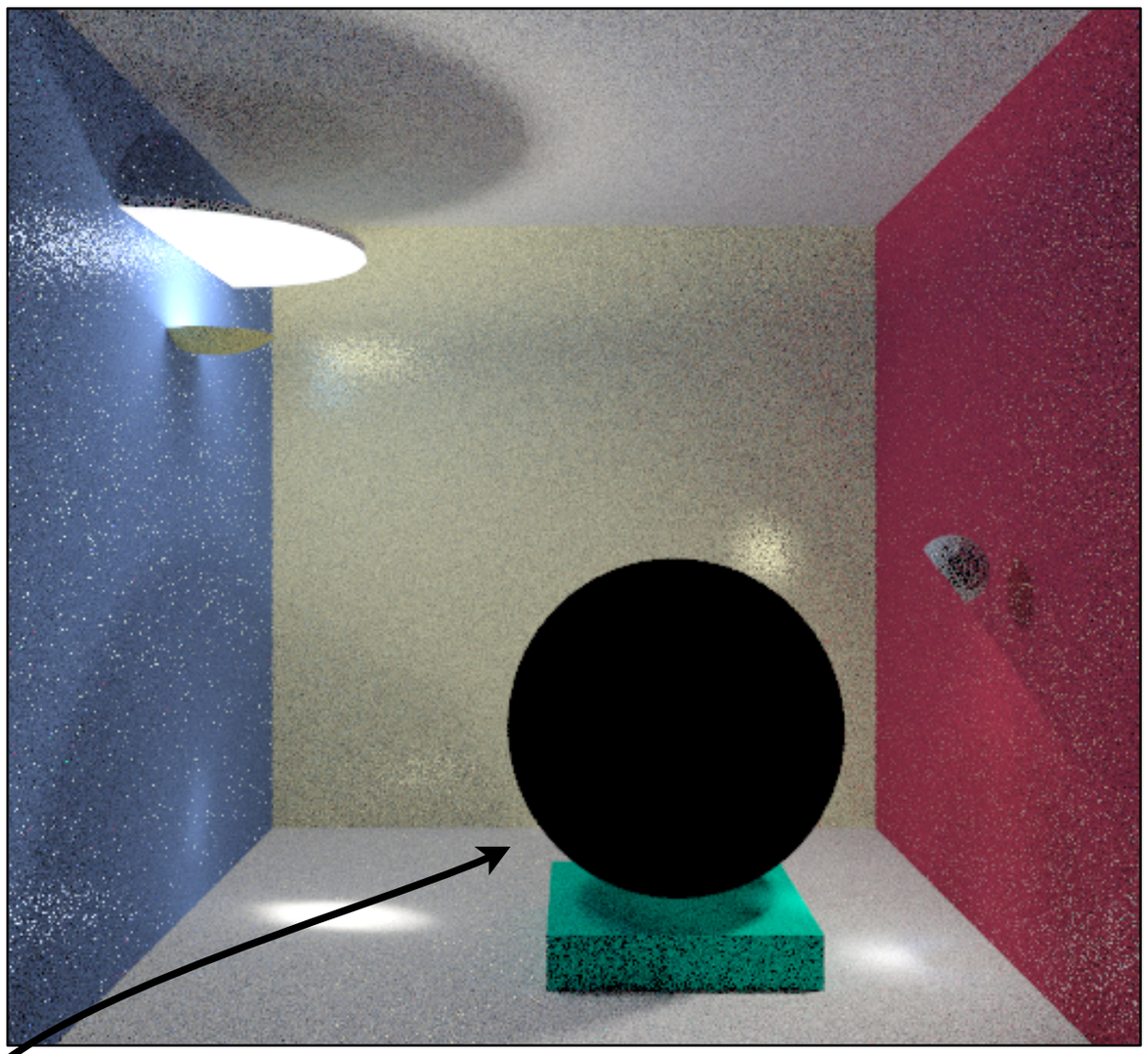


Path vs. Light Tracing

Path tracing



Light tracing

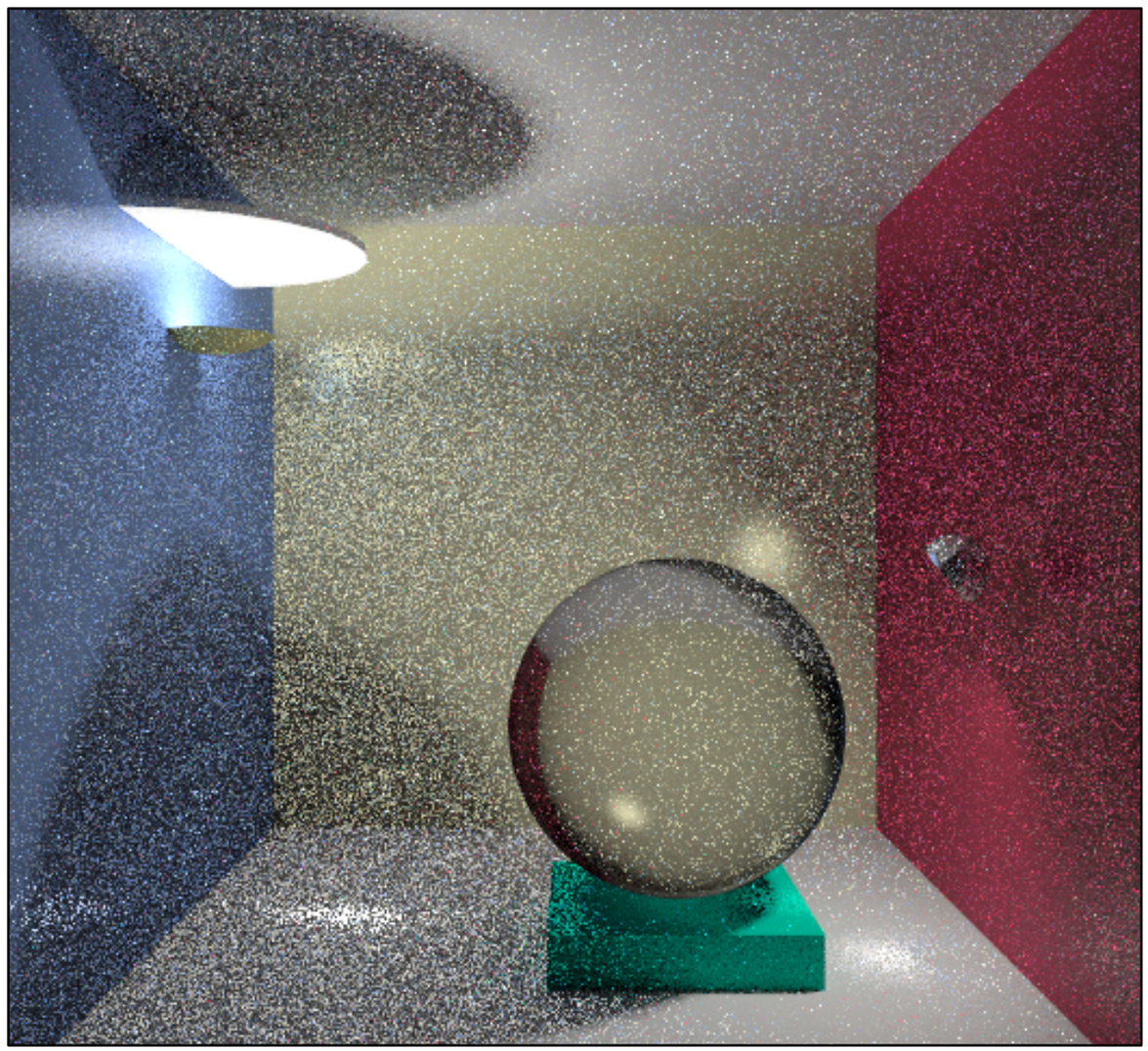


Images courtesy of F. Suykens

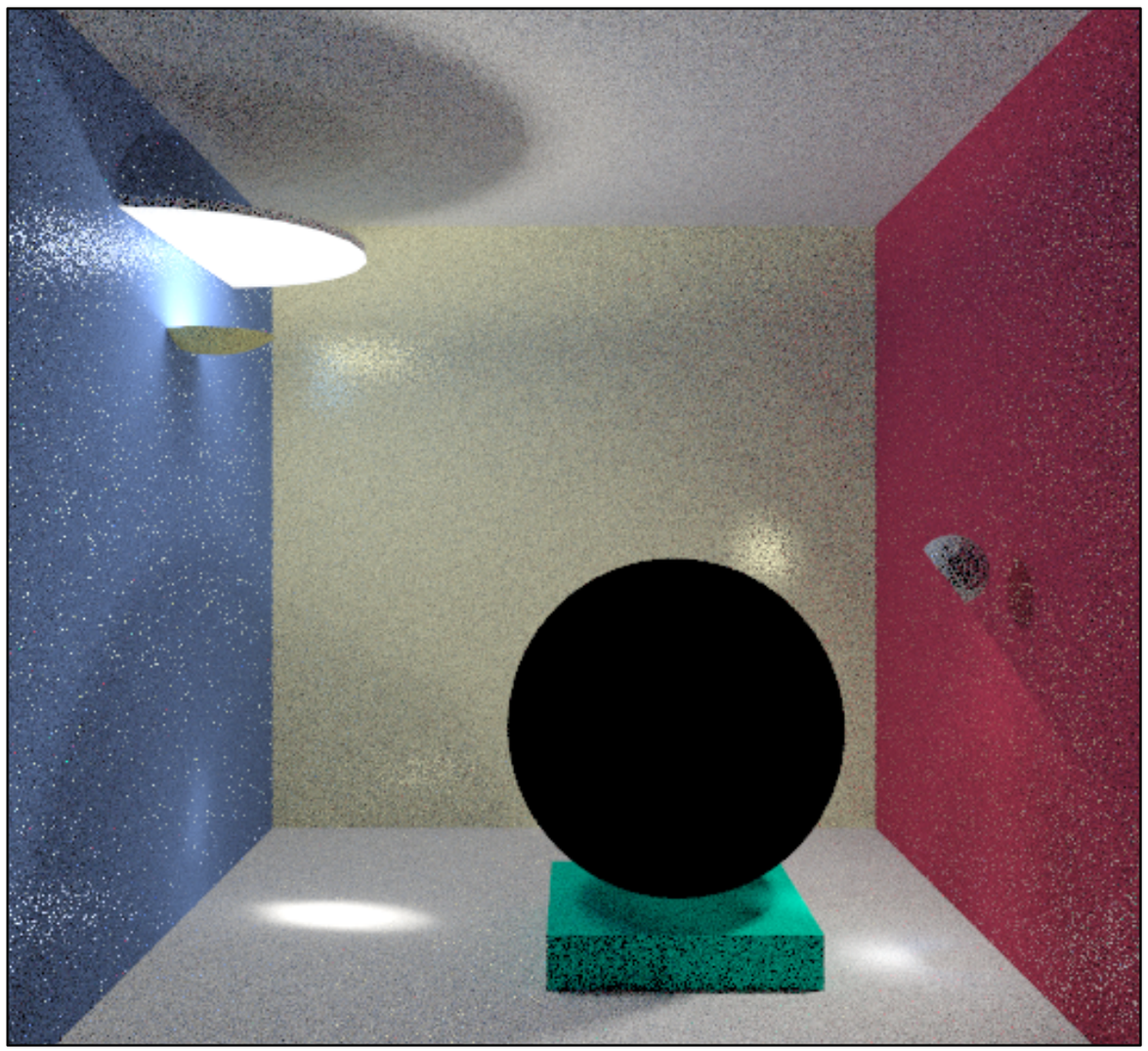
Why is this glass sphere black?

Path vs. Light Tracing

Path tracing



Light tracing



Images courtesy of F. Suykens

Can we <u>combine</u> them?

Visual Break

Visual Break

Visual Break

Path Integral Framework

Measurement Equation

$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0)\, d\mathbf{x}_1 d\mathbf{x}_0$$

$$= \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) + \int_A f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) L_o(\mathbf{x}_2, \mathbf{x}_1) d\mathbf{x}_2 d\mathbf{x}_1 d\mathbf{x}_0$$

$$= \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) + \int_A f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) L_e(\mathbf{x}_2, \mathbf{x}_1) + \int_A f(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_1) G(\mathbf{x}_2, \mathbf{x}_3)\, L_e(\mathbf{x}_3, \mathbf{x}_2) + \int_A \cdots\, d\mathbf{x}_4 d\mathbf{x}_3 d\mathbf{x}_2 d\mathbf{x}_1 d\mathbf{x}_0$$

Hard to concisely express arbitrary light transport with all the nested integrals

Only some can do it....

Let’s find a better way

Path Integral Form of Measurement Eq.

$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0)\, d\mathbf{x}_1 d\mathbf{x}_0$$

Emission

$$= \iint_A W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) G(\mathbf{x}_0, \mathbf{x}_1) d\mathbf{x}_1 d\mathbf{x}_0$$

Direct illumination (3 vertices)

$$+ \iiint_A W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_2, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) d\mathbf{x}_2 d\mathbf{x}_1 d\mathbf{x}_0 + \cdots$$

(k-2)-bounce illumination (k vertices)

$$+ \int \cdots \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1})\, d\mathbf{x}_k \cdots d\mathbf{x}_0 + \cdots$$

introduce: $\mathcal{P}_k = \{\bar{\mathbf{x}} = \mathbf{x}_0 \cdots \mathbf{x}_k;\ \ \mathbf{x}_0 \cdots \mathbf{x}_k \in A\}$

space of all paths with k segments

Path Integral Form of Measurement Eq.

$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0)\, d\mathbf{x}_1 d\mathbf{x}_0$$

$$= \int_{\mathcal{P}_1} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) G(\mathbf{x}_0, \mathbf{x}_1) d\bar{\mathbf{x}}_1$$

Emission

$$+ \int_{\mathcal{P}_2} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_2, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_0) G(\mathbf{x}_1, \mathbf{x}_2) d\bar{\mathbf{x}}_2 + \cdots$$

Direct illumination (3 vertices)

$$+ \int_{\mathcal{P}_k} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1})\, d\bar{\mathbf{x}}_k + \cdots$$

(k-2)-bounce illumination (k vertices)

introduce: $T(\bar{\mathbf{x}}_k) = G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1})$

throughput of path $\bar{\mathbf{x}}_k$

Path Integral Form of Measurement Eq.

$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0)\, d\mathbf{x}_1 d\mathbf{x}_0$$

Emission

$$= \int_{\mathcal{P}_1} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_1, \mathbf{x}_0) T(\bar{\mathbf{x}}_1) d\bar{\mathbf{x}}_1$$

Direct illumination (3 vertices)

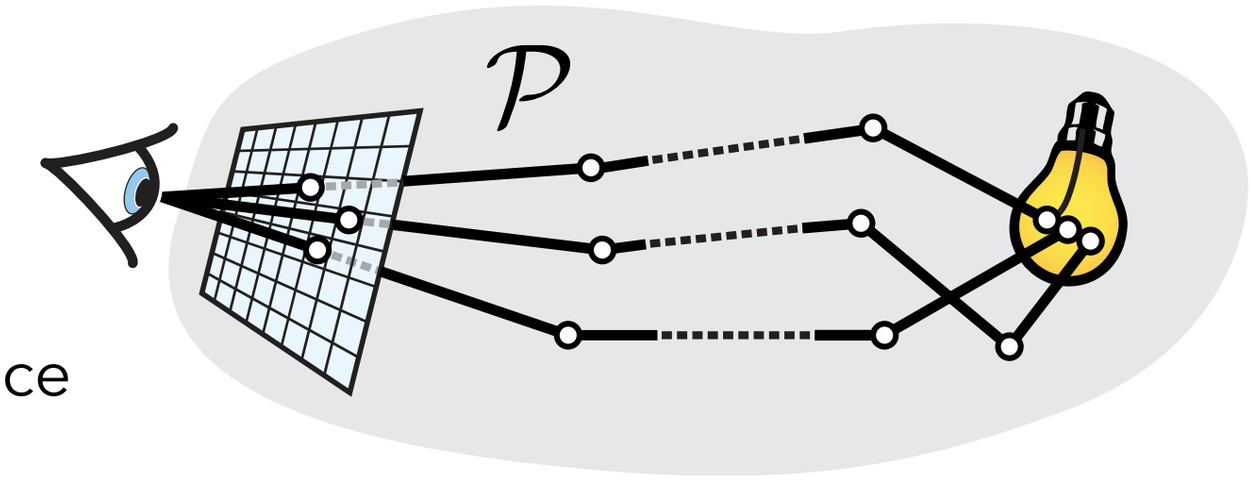
$$+ \int_{\mathcal{P}_2} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_2, \mathbf{x}_1) T(\bar{\mathbf{x}}_2)\, d\bar{\mathbf{x}}_2 + \cdots$$

(k-2)-bounce illumination (k vertices)

$$+ \int_{\mathcal{P}_k} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}_k)\, d\bar{\mathbf{x}}_k + \cdots$$

introduce: $\mathcal{P} = \bigcup_{k=1}^{\infty} \mathcal{P}_k$

the *path space*, i.e. the space of all paths of all lengths



Path Integral Form of Measurement Eq.

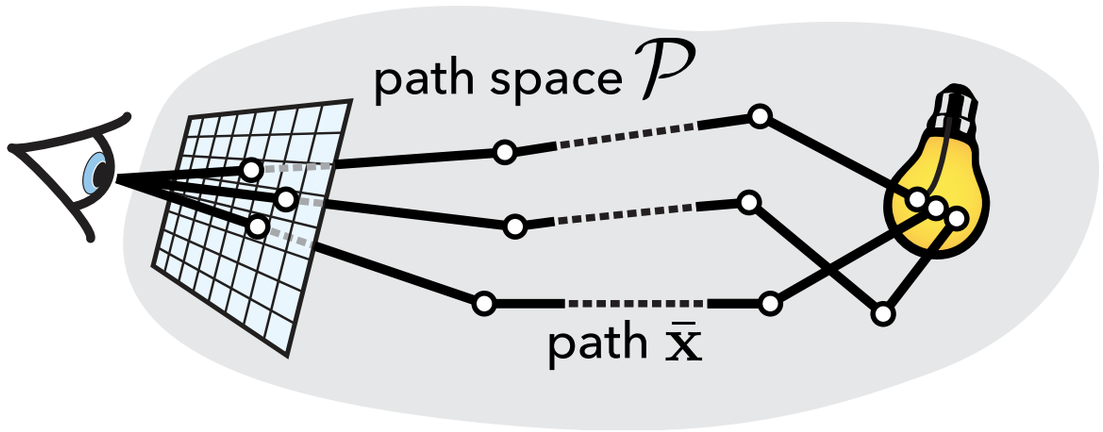
$$I_j = \int_A \int_A W_e(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) L_o(\mathbf{x}_1, \mathbf{x}_0)\, d\mathbf{x}_1 d\mathbf{x}_0$$

global illumination (all paths of all lenghts)

$$= \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}})\, d\bar{\mathbf{x}}$$

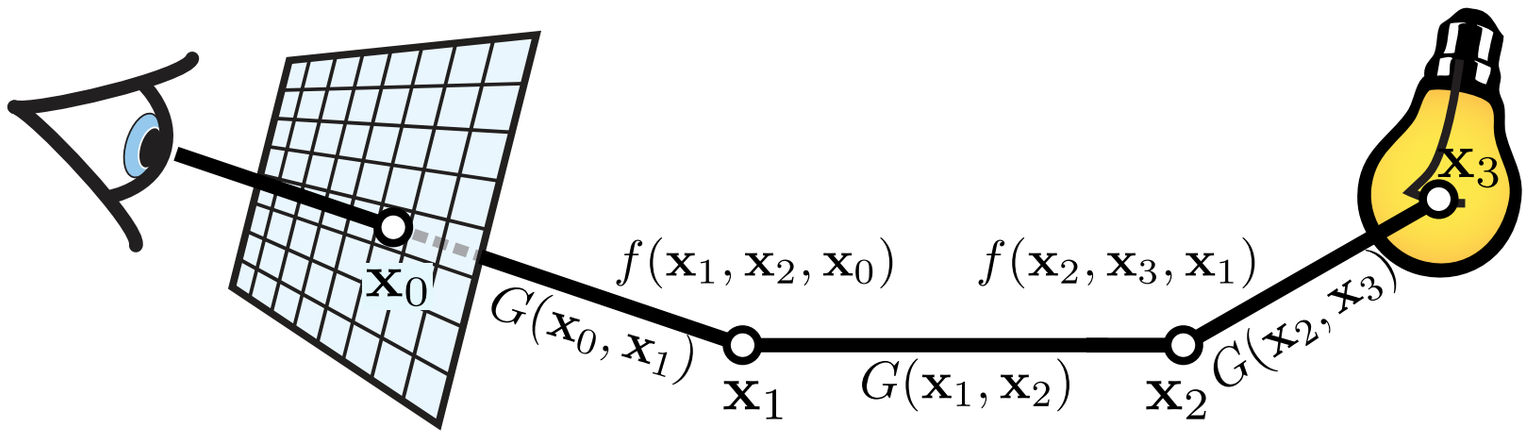
Path Integral Form of Measurement Eq.

$$I_j = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}})\, d\bar{\mathbf{x}}$$



path throughput

$$T(\bar{\mathbf{x}}) = G(\mathbf{x}_0, \mathbf{x}_1) \prod_{j=1}^{k-1} f(\mathbf{x}_j, \mathbf{x}_{j+1}, \mathbf{x}_{j-1}) G(\mathbf{x}_j, \mathbf{x}_{j+1})$$



Path Integral Form of Measurement Eq.

$$I_j = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}})\, d\bar{\mathbf{x}}$$

- Advantages:
  - no recursion, no “nasty” nested integrals
  - emphasizes symmetry of light transport
  - easy to relate different rendering algorithms
  - focuses on path geometry, independent of strategy for constructing paths
  - MC estimator on path space looks much simpler

Path Integral Form of Measurement Eq.

$$I_j = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}})\, d\bar{\mathbf{x}}$$

- Monte Carlo estimator:

$$I_j \approx \frac{1}{N} \sum_{i=1}^{N} \frac{W_e(\mathbf{x}_{i,0}, \mathbf{x}_{i,1}) L_e(\mathbf{x}_{i,k}, \mathbf{x}_{i,k-1}) T(\bar{\mathbf{x}}_i)}{p(\bar{\mathbf{x}}_i)}$$

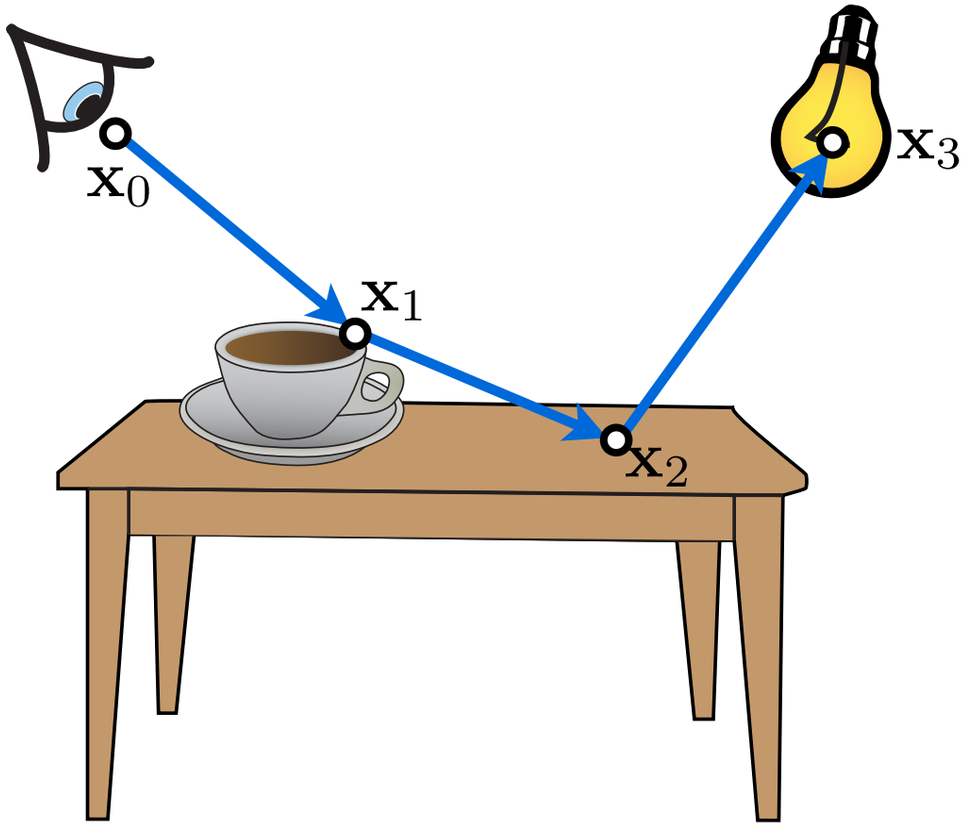
$$p(\bar{\mathbf{x}}) = \underbrace{p(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_k)}_{\text{joint PDF of path vertices}}$$

path PDF

Path Construction

$$p(\bar{\mathbf{x}}) = p(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_k)$$

Path tracing w/o NEE

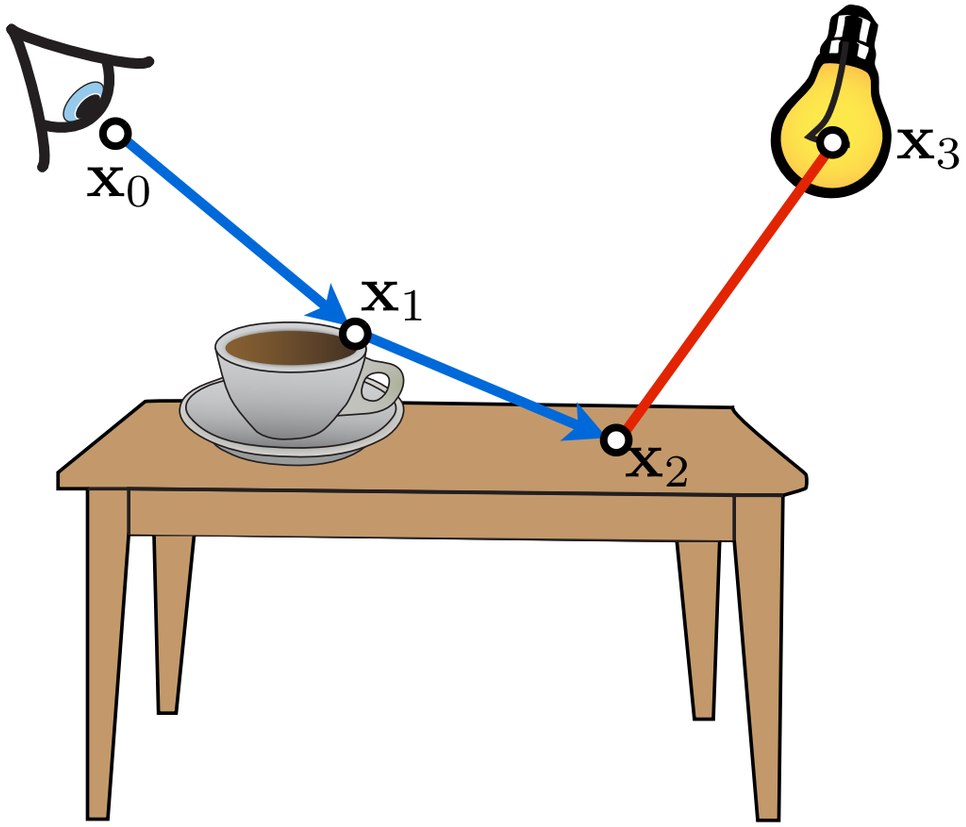


$$\begin{aligned} p(\bar{\mathbf{x}}) = & p(\mathbf{x}_0) \\ & \times p(\mathbf{x}_1 | \mathbf{x}_0) \\ & \times p(\mathbf{x}_2 | \mathbf{x}_0 \mathbf{x}_1) \\ & \times p(\mathbf{x}_3 | \mathbf{x}_0 \mathbf{x}_1 \mathbf{x}_2) \end{aligned}$$

Path Construction

$$p(\bar{\mathbf{x}}) = p(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_k)$$

Path tracing with NEE



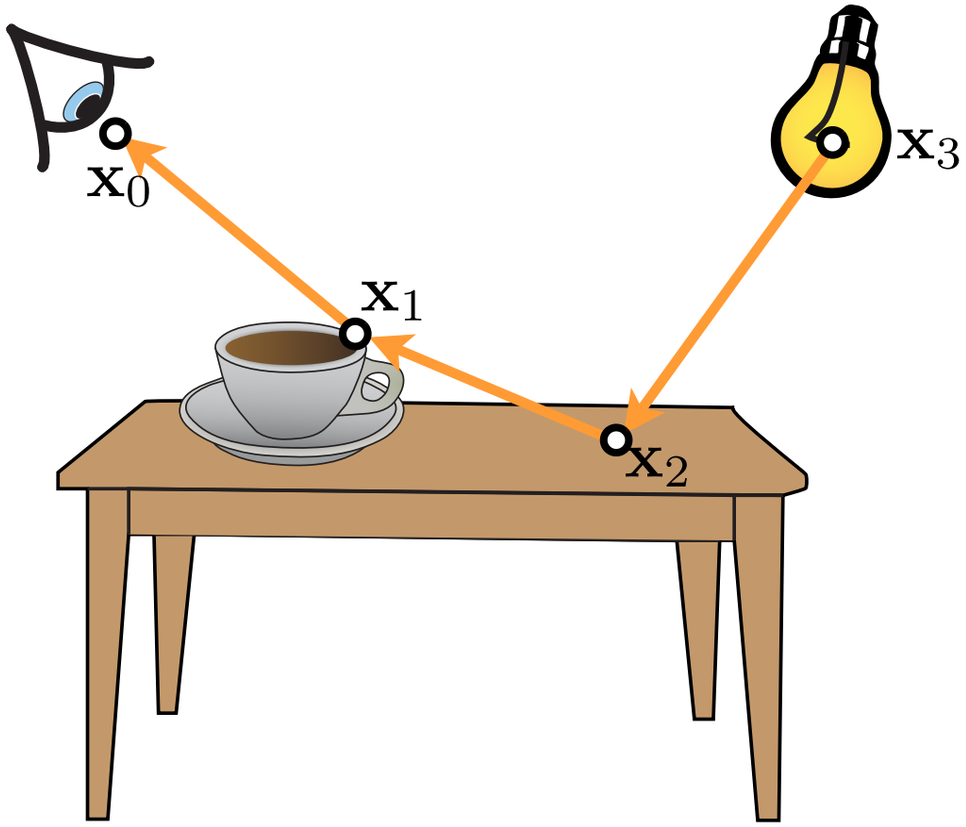
$$\begin{aligned} p(\bar{\mathbf{x}}) = & p(\mathbf{x}_0) \\ & \times p(\mathbf{x}_1 | \mathbf{x}_0) \\ & \times p(\mathbf{x}_2 | \mathbf{x}_0 \mathbf{x}_1) \\ & \times p(\mathbf{x}_3) \end{aligned}$$

assuming uniform area sampling

Path Construction

$$p(\bar{\mathbf{x}}) = p(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_k)$$

Light tracing

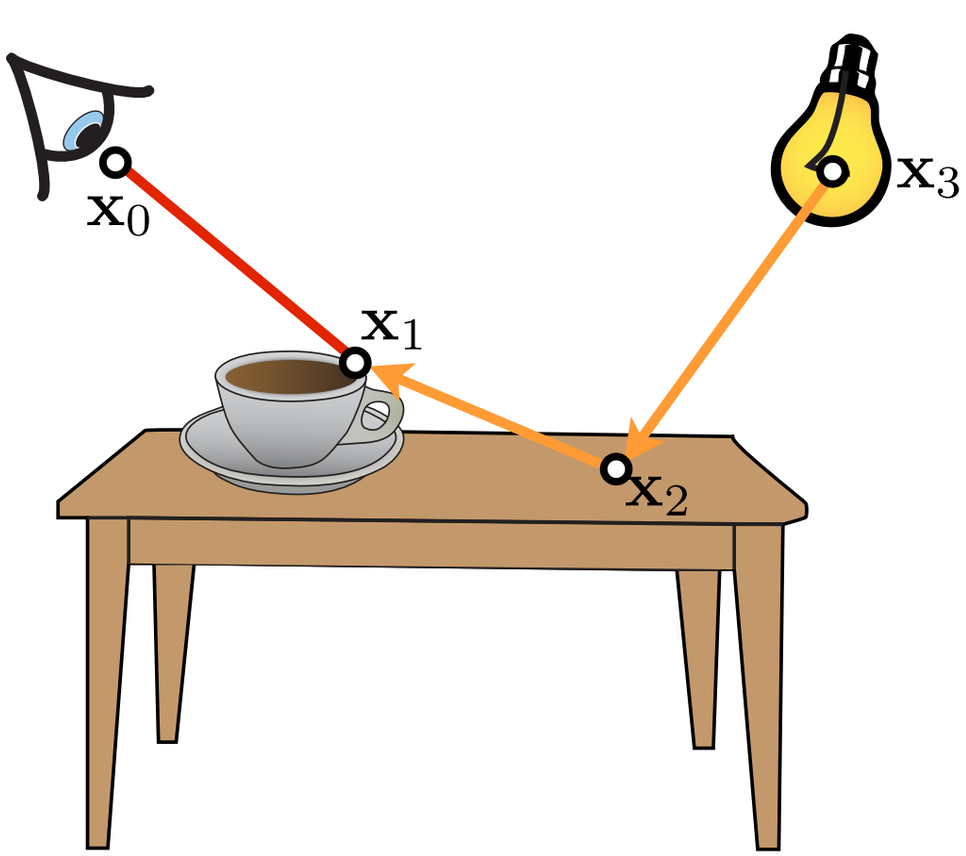


$$\begin{aligned} p(\bar{\mathbf{x}}) = & p(\mathbf{x}_0|\mathbf{x}_3\mathbf{x}_2\mathbf{x}_1) \\ & \times p(\mathbf{x}_1|\mathbf{x}_3\mathbf{x}_2) \\ & \times p(\mathbf{x}_2|\mathbf{x}_3) \\ & \times p(\mathbf{x}_3) \end{aligned}$$

Path Construction

$$p(\bar{\mathbf{x}}) = p(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_k)$$

Light tracing with NEE



assuming uniform aperture sampling

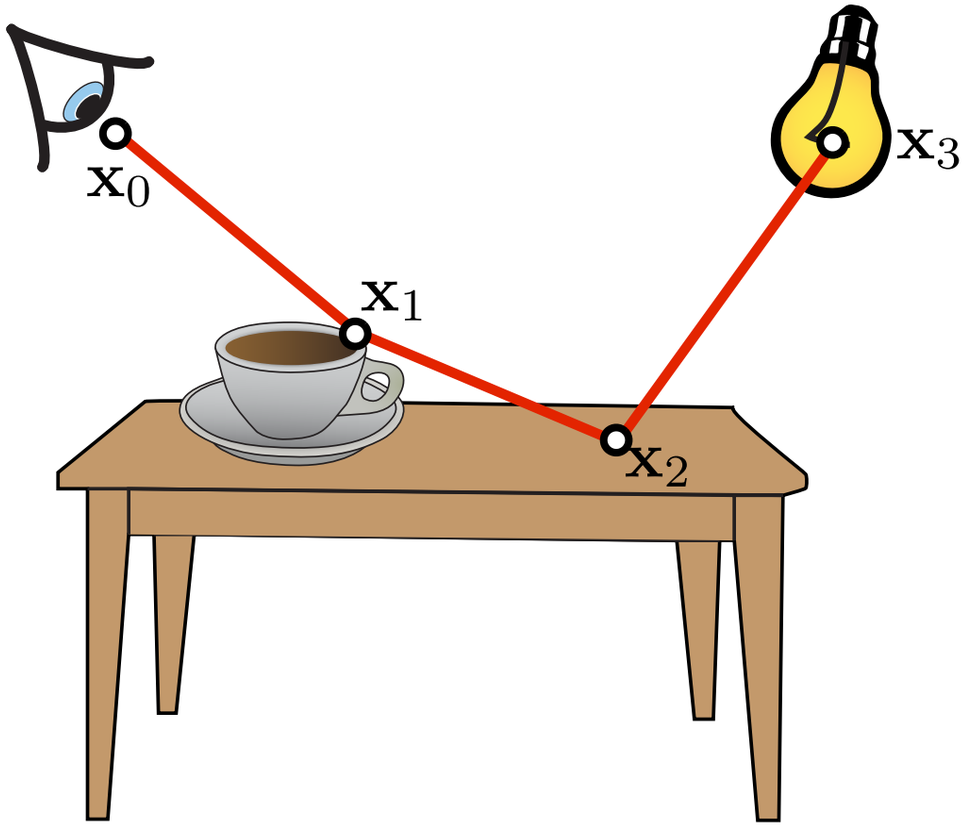
$$\begin{aligned} p(\bar{\mathbf{x}}) &= p(\mathbf{x}_0) \\ &\times p(\mathbf{x}_1 | \mathbf{x}_3 \mathbf{x}_2) \\ &\times p(\mathbf{x}_2 | \mathbf{x}_3) \\ &\times p(\mathbf{x}_3) \end{aligned}$$

Path Construction

$$p(\bar{\mathbf{x}}) = p(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_k)$$

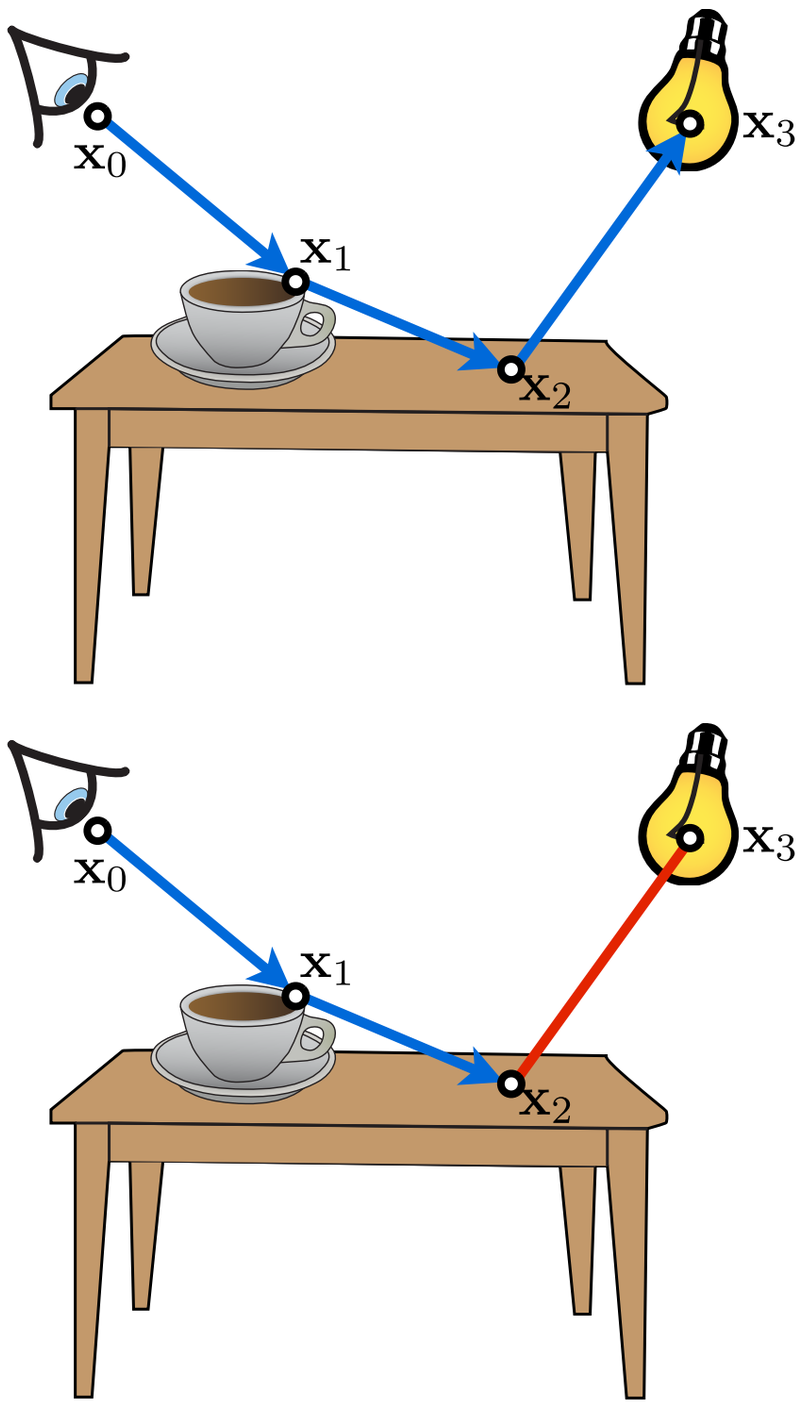
Independent sampling of path vertices

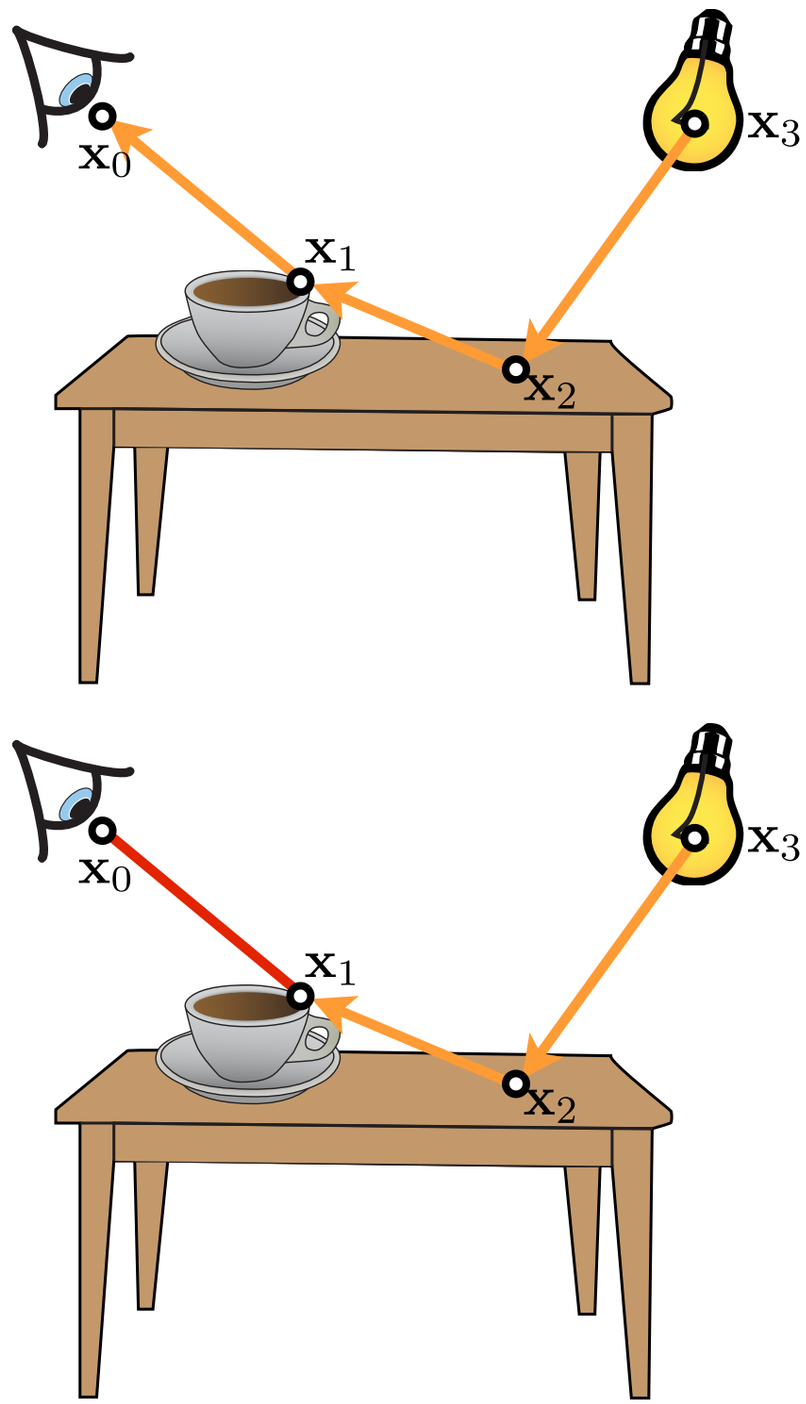
(not very practical though)



$$\begin{aligned} p(\bar{\mathbf{x}}) = & p(\mathbf{x}_0) \\ & \times p(\mathbf{x}_1) \\ & \times p(\mathbf{x}_2) \\ & \times p(\mathbf{x}_3) \end{aligned}$$

Can we combine them?

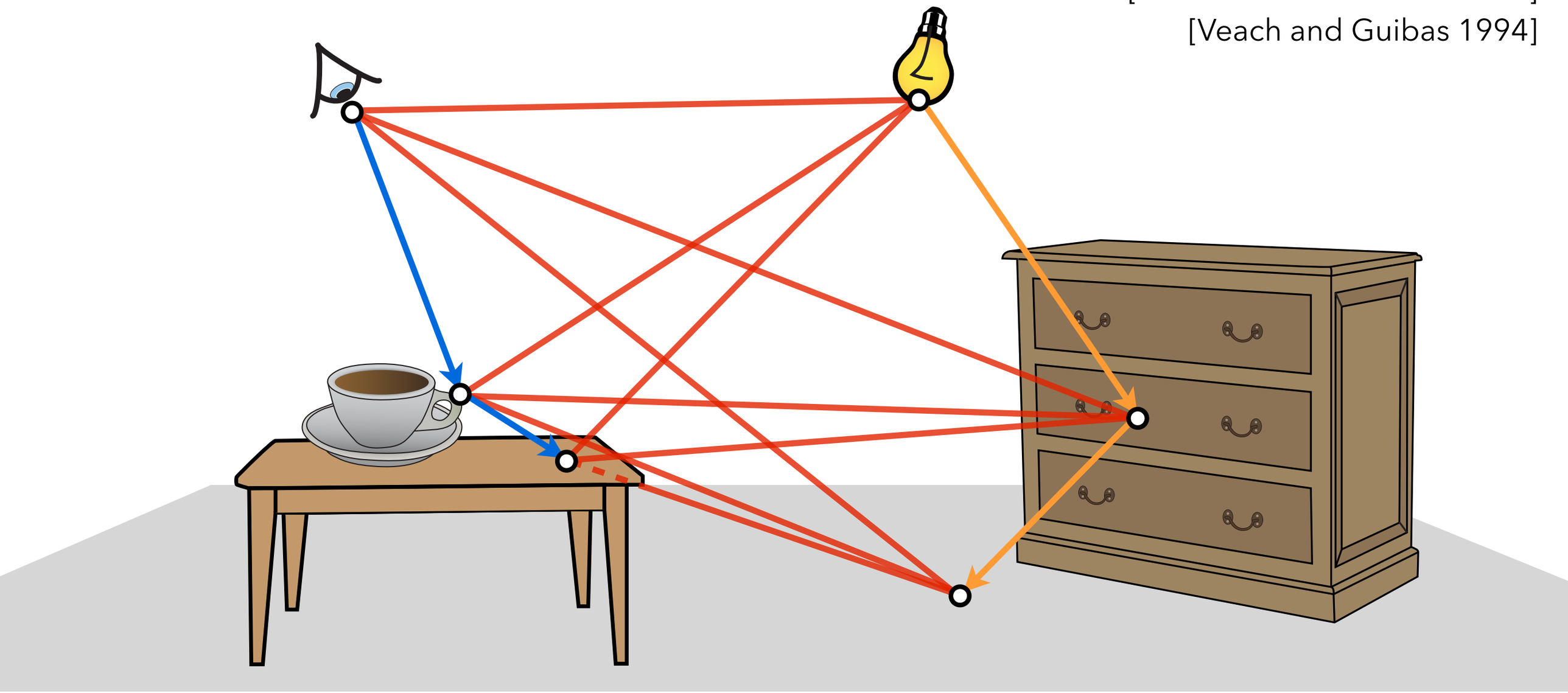




Bidirectional Path Tracing

Bidirectional Path Tracing

[Lafortune and Willems 1993]
[Veach and Guibas 1994]



t - # vertices on camera subpath

s - # vertices on light subpath

ts - # connections

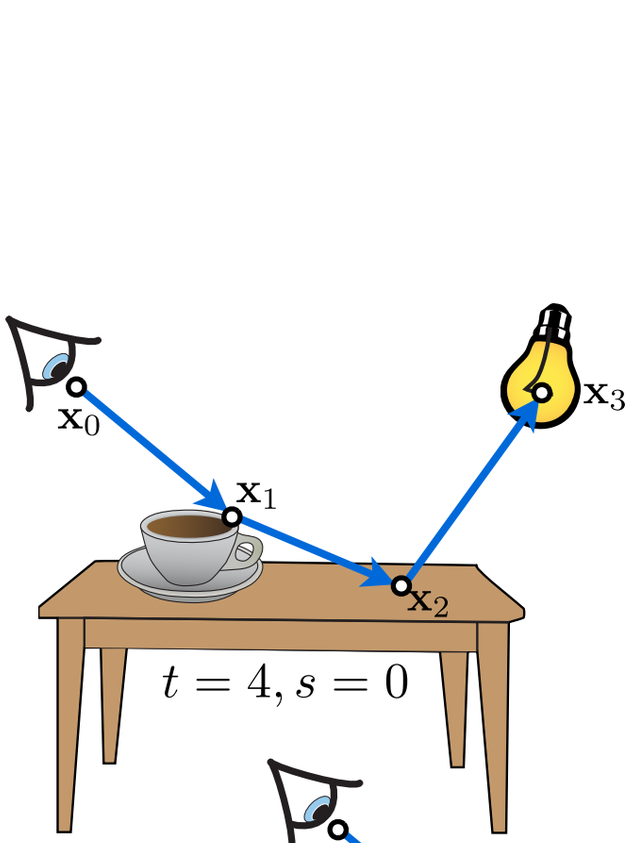
Bidirectional Path Tracing

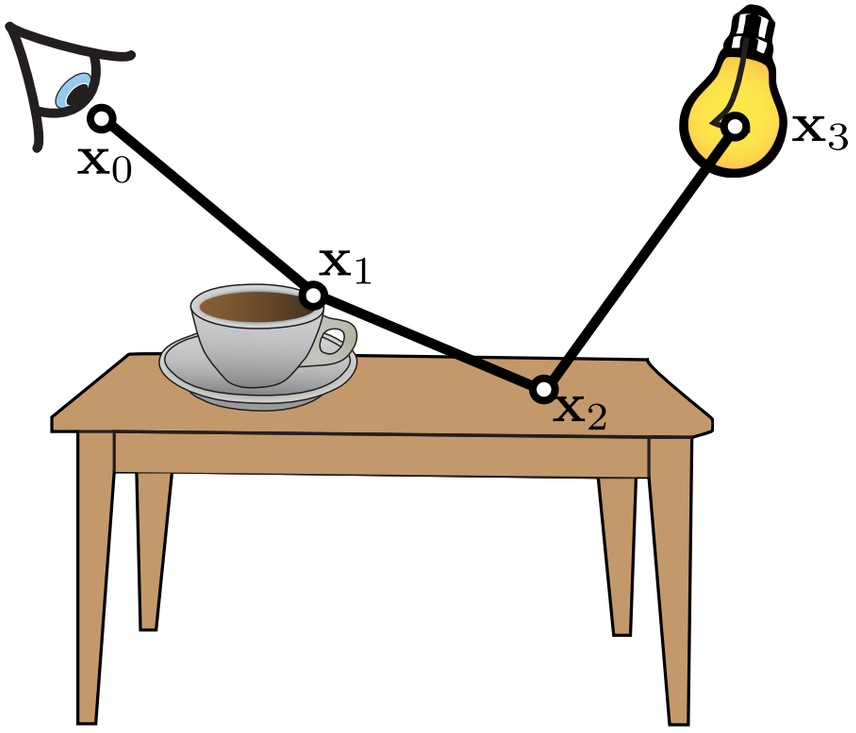
```
color estimate (point x)
{
    lp = sample light subpath
    cp = sample camera subpath for image point x
    for each vertex s in lp
        for each vertex t in cp
            fullPath = join(cp[0..s], lp[0..t])
            splat(fullPath.screenPos, fullPath.contrib)
}
```

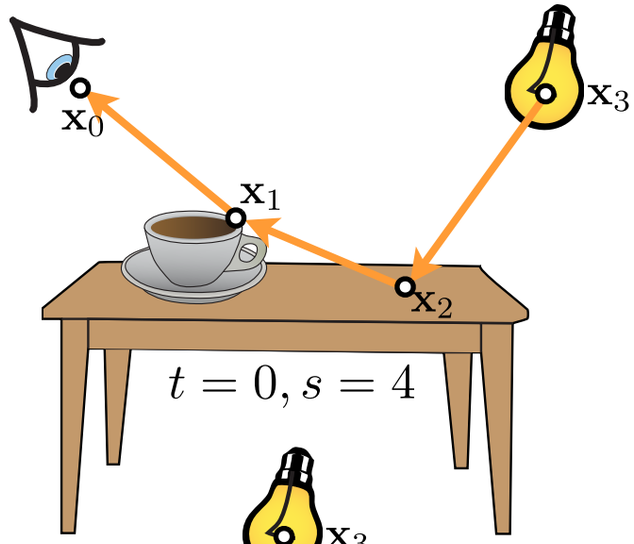
Bidirectional Path Tracing

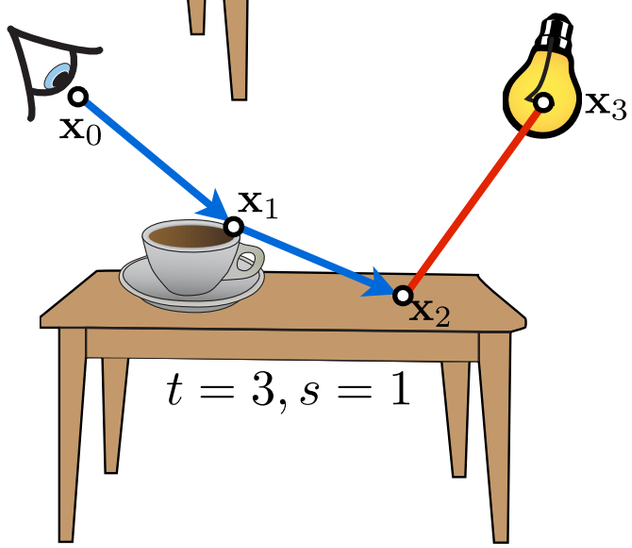
- Key observations:
  - Every path (formed by connecting camera sub-path to light sub-path) with k vertices can be constructed using $k+1$ *strategies*
  - For a particular path length, all strategies estimate the *same integral*
  - Each strategy has a *different PDF*, i.e. each strategy has different strengths and weaknesses
  - Let's combine them using MIS!

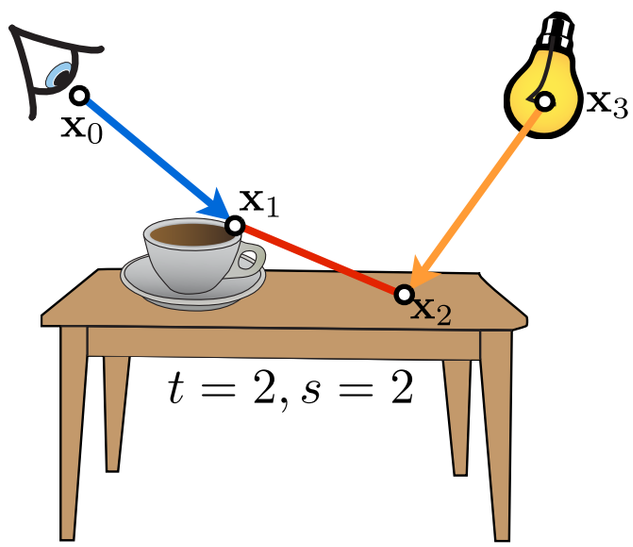
Bidirectional Path Tracing

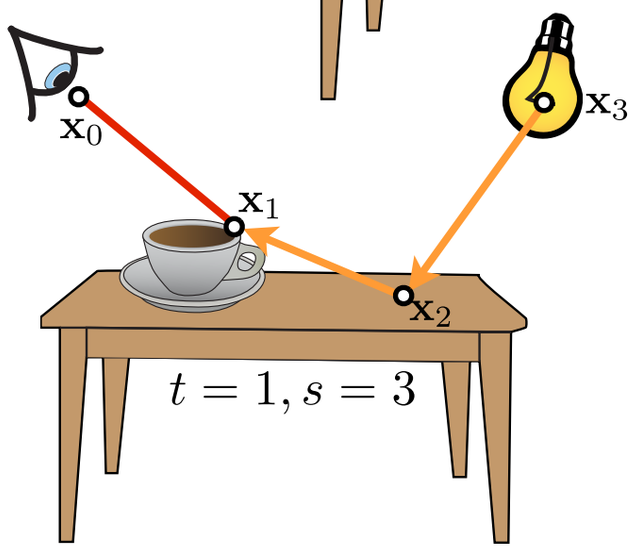




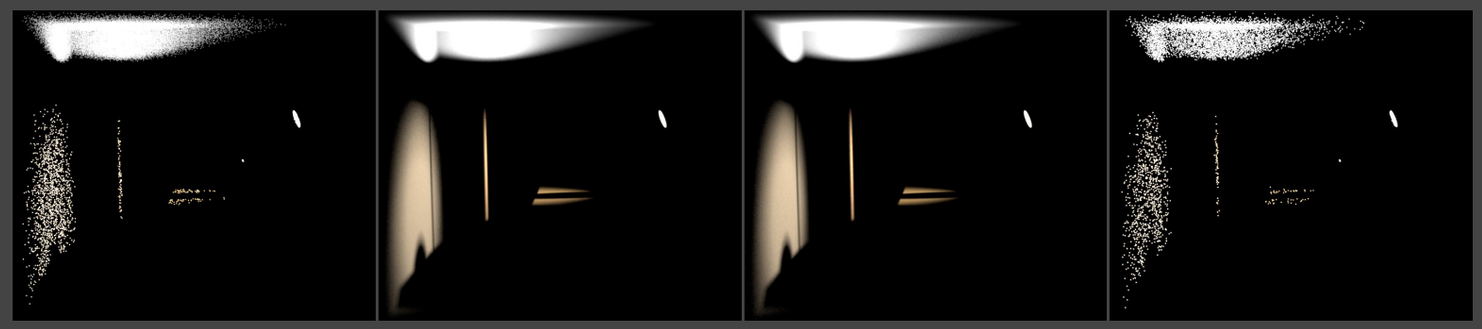




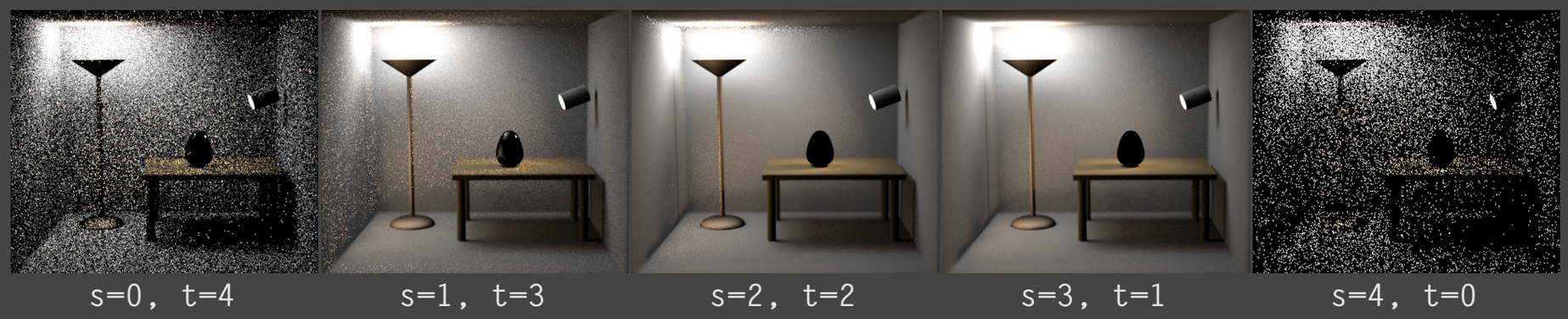




Bidirectional Path Tracing



s=0, t=3 s=1, t=2 s=2, t=1 s=3, t=0

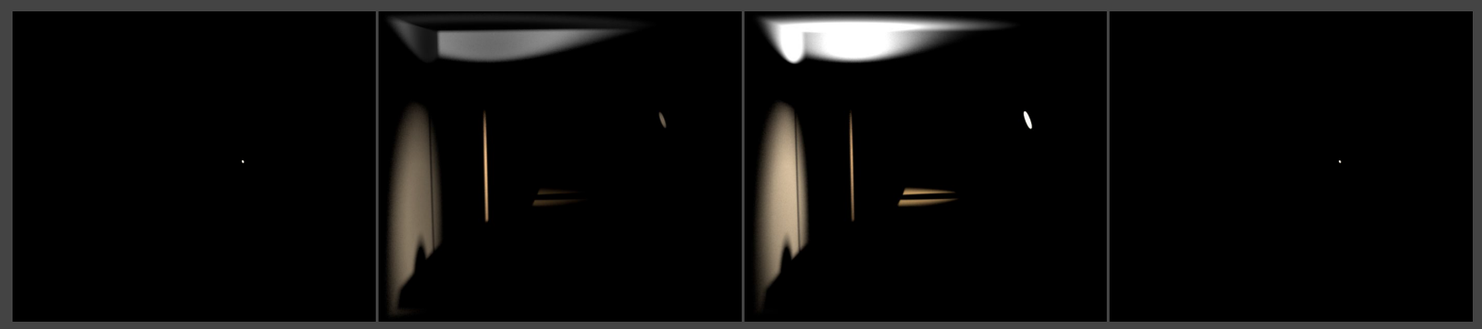


s=0, t=4 s=1, t=3 s=2, t=2 s=3, t=1 s=4, t=0

s=0, t=5 s=1, t=4 s=2, t=3 s=3, t=2 s=4, t=1 s=5, t=0

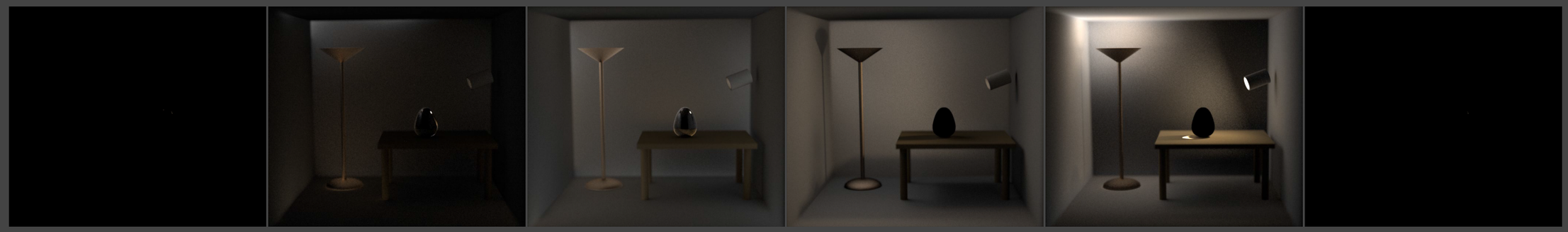
s=0, t=6 s=1, t=5 s=2, t=4 s=3, t=3 s=4, t=2 s=5, t=1 s=6, t=0

Bidirectional Path Tracing (MIS)

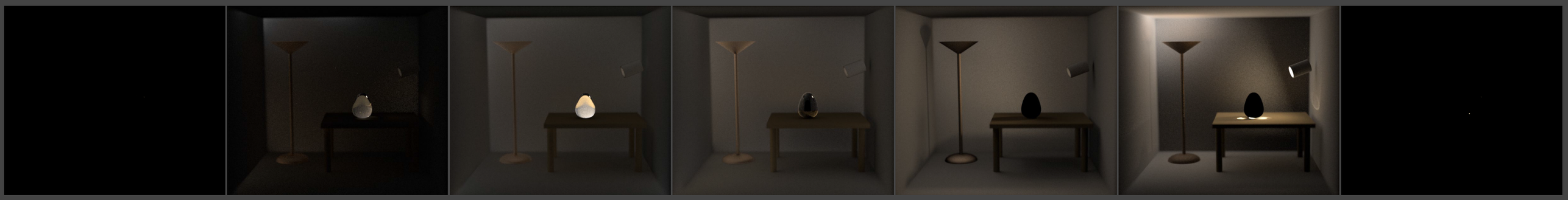


s=0, t=3 s=1, t=2 s=2, t=1 s=3, t=0

s=0, t=4 s=1, t=3 s=2, t=2 s=3, t=1 s=4, t=0



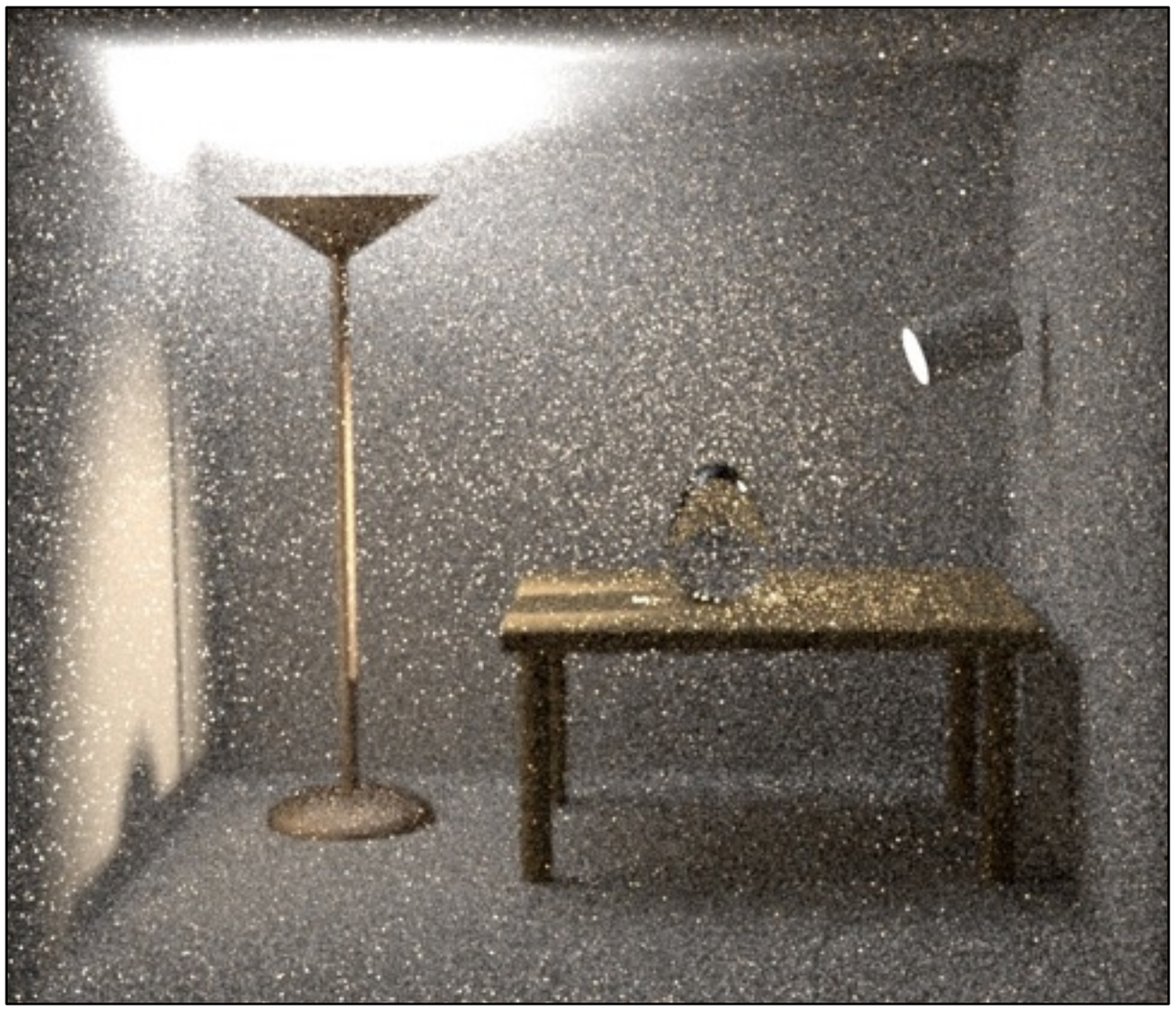
s=0, t=5 s=1, t=4 s=2, t=3 s=3, t=2 s=4, t=1 s=5, t=0



s=0, t=6 s=1, t=5 s=2, t=4 s=3, t=3 s=4, t=2 s=5, t=1 s=6, t=0

Bidirectional Path Tracing

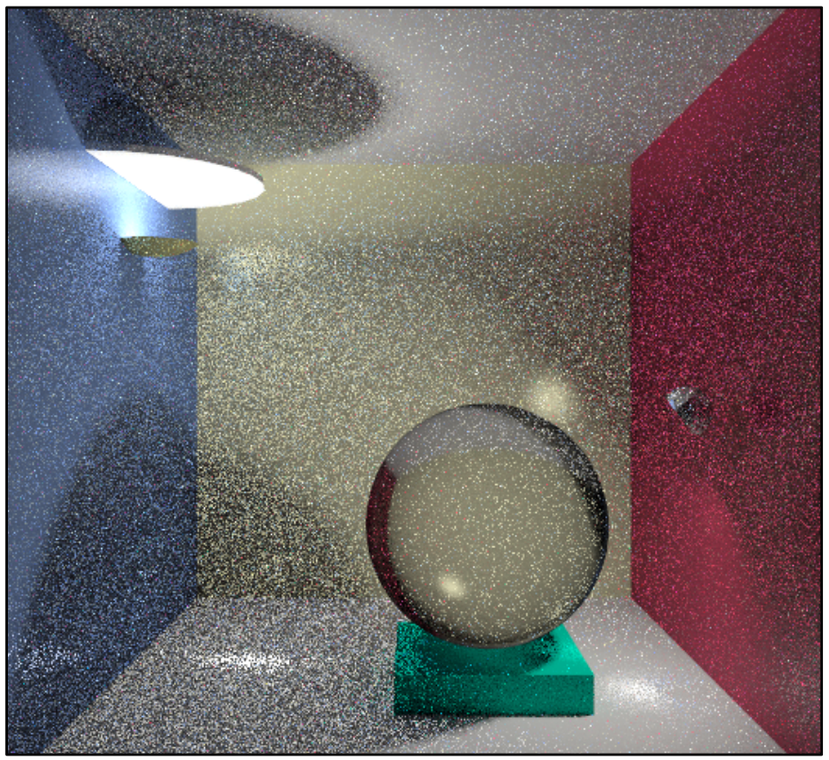
(Unidirectional) path tracing



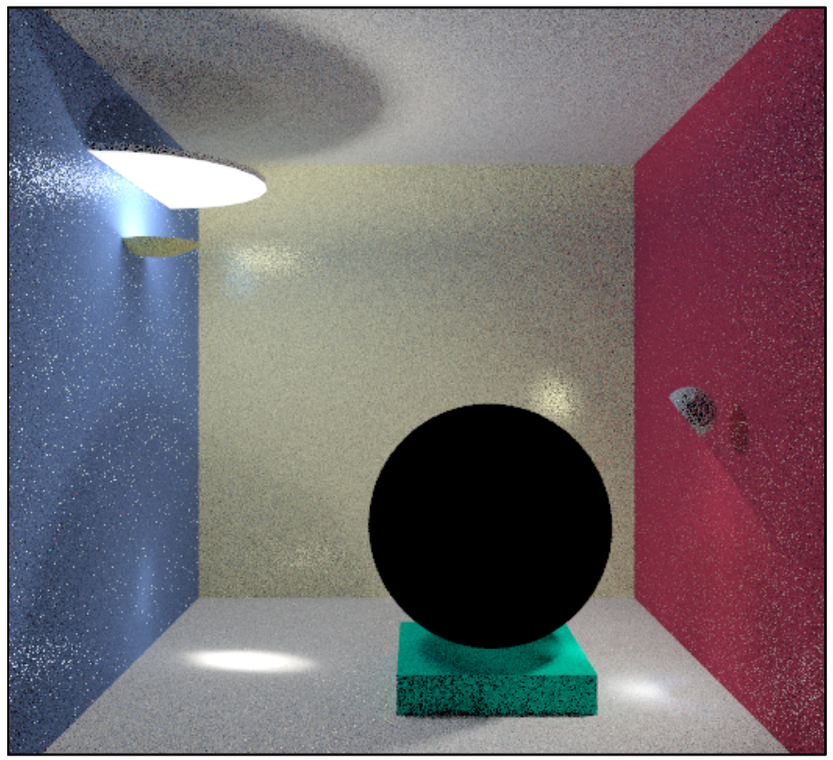
Bidirectional path tracing

Bidirectional Path Tracing

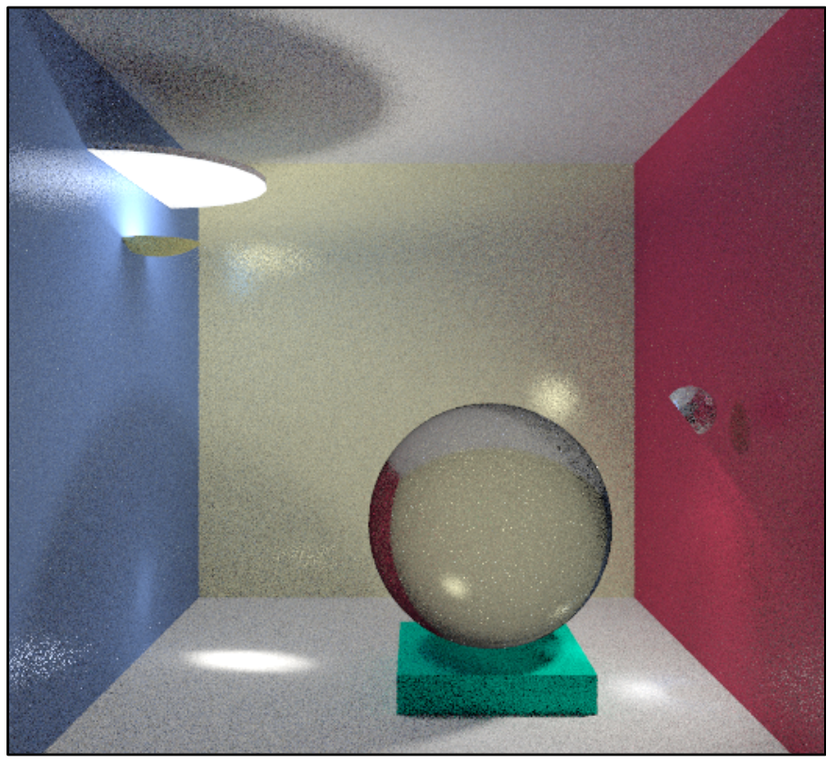
Path tracing



Light tracing



Bidirectional path tracing

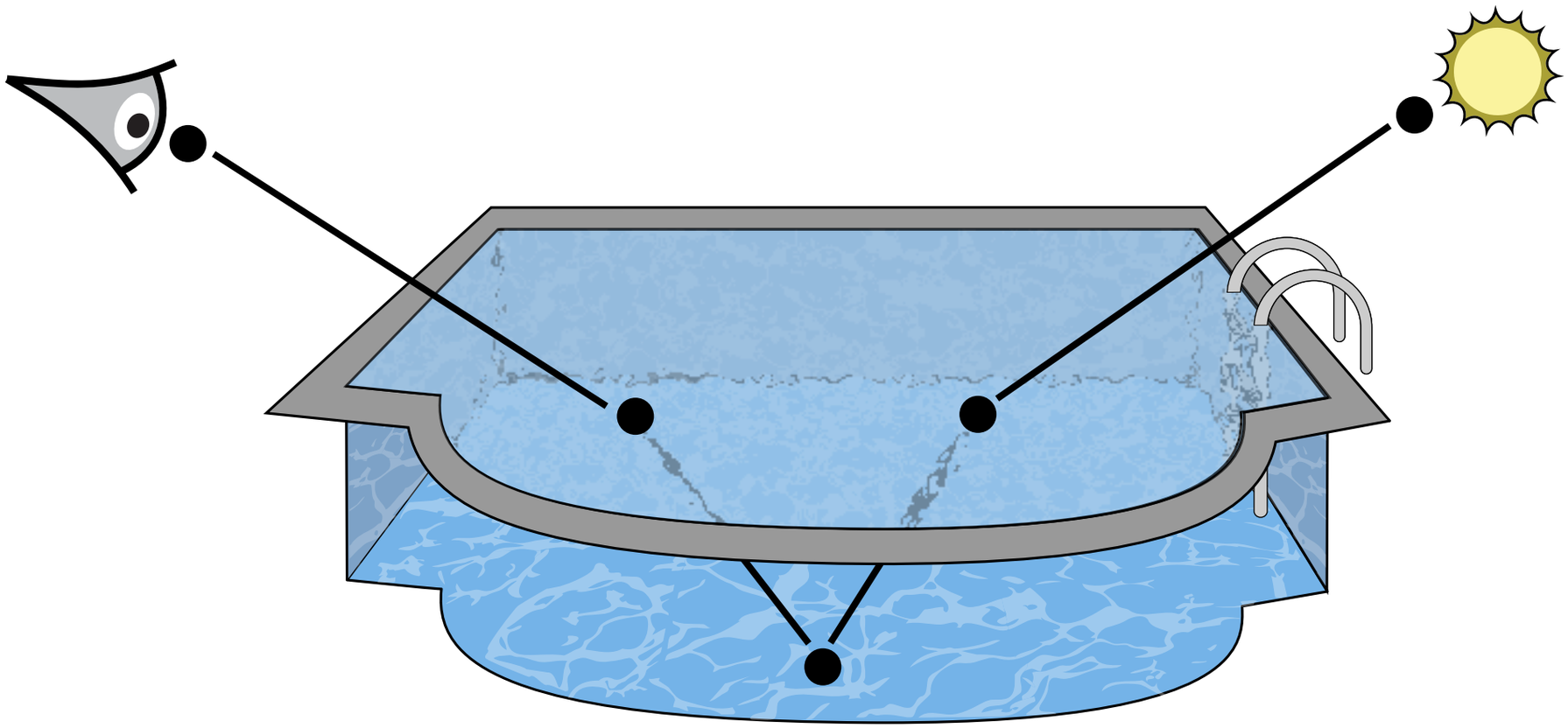


Still not robust enough...

Reference

Bidirectional PT

Still not robust enough...



$LSDSE$ paths are difficult for any unbiased method

Still not robust enough...

- Extensions
  - Combination with photon mapping
    - Unified Path Sampling [Hachisuka et al. 2012]
    - Vertex Connection Merging [Georgiev et al. 2012]
  - Metropolis sampling (global PDF)

Visual Break

Operator Notation

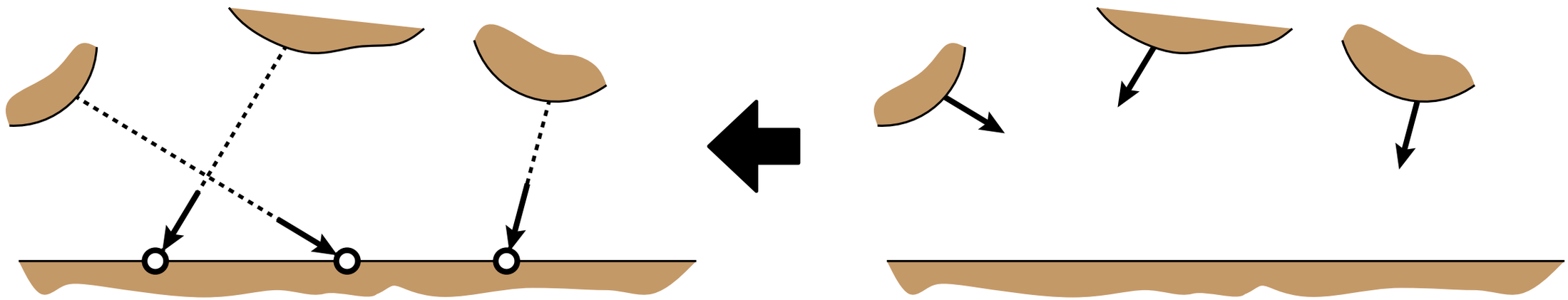
Operator Notation

- Goal: concisely express an integral equation using a linear operator that maps between functions
- Introduced to graphics by James Arvo in 1994

Propagation Operator $\mathbf{G}$

- Applies the ray tracing function

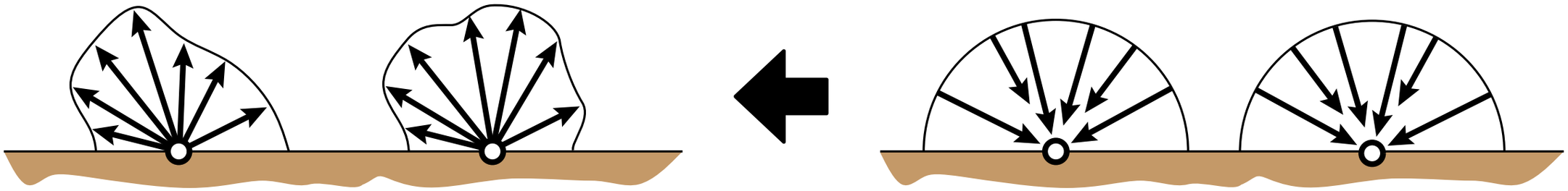
$$L_i(\mathbf{x}, \vec{\omega}) = (\mathbf{G}L_o)(\mathbf{x}, \vec{\omega}) = L_o(r(\mathbf{x}, \vec{\omega}), -\vec{\omega})$$



Local Scattering Operator $\mathbf{K}$

- Amounts to the reflection equation

$$L_o(\mathbf{x}, \vec{\omega}_o) = (\mathbf{K}L_i)(\mathbf{x}, \vec{\omega}_o) = \int_{S^2} f_s(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) \left|\cos\theta\right| \, d\vec{\omega}_i$$



Neumann Series Expansion

$$L_o = L_e + \mathbf{KG} L_o$$

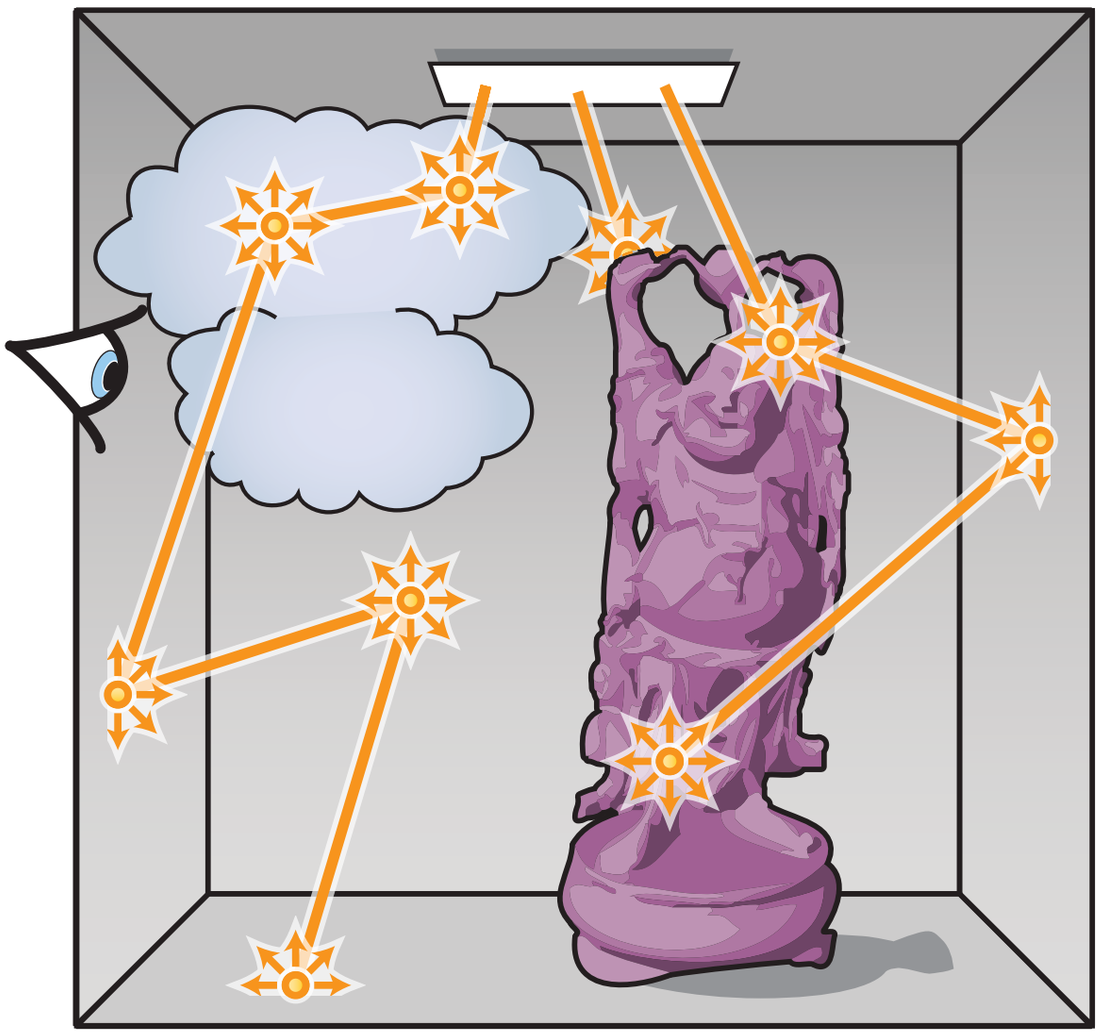
$$L_o = L_e + \mathbf{KG} L_e + (\mathbf{KG})^2 L_e + \cdots$$

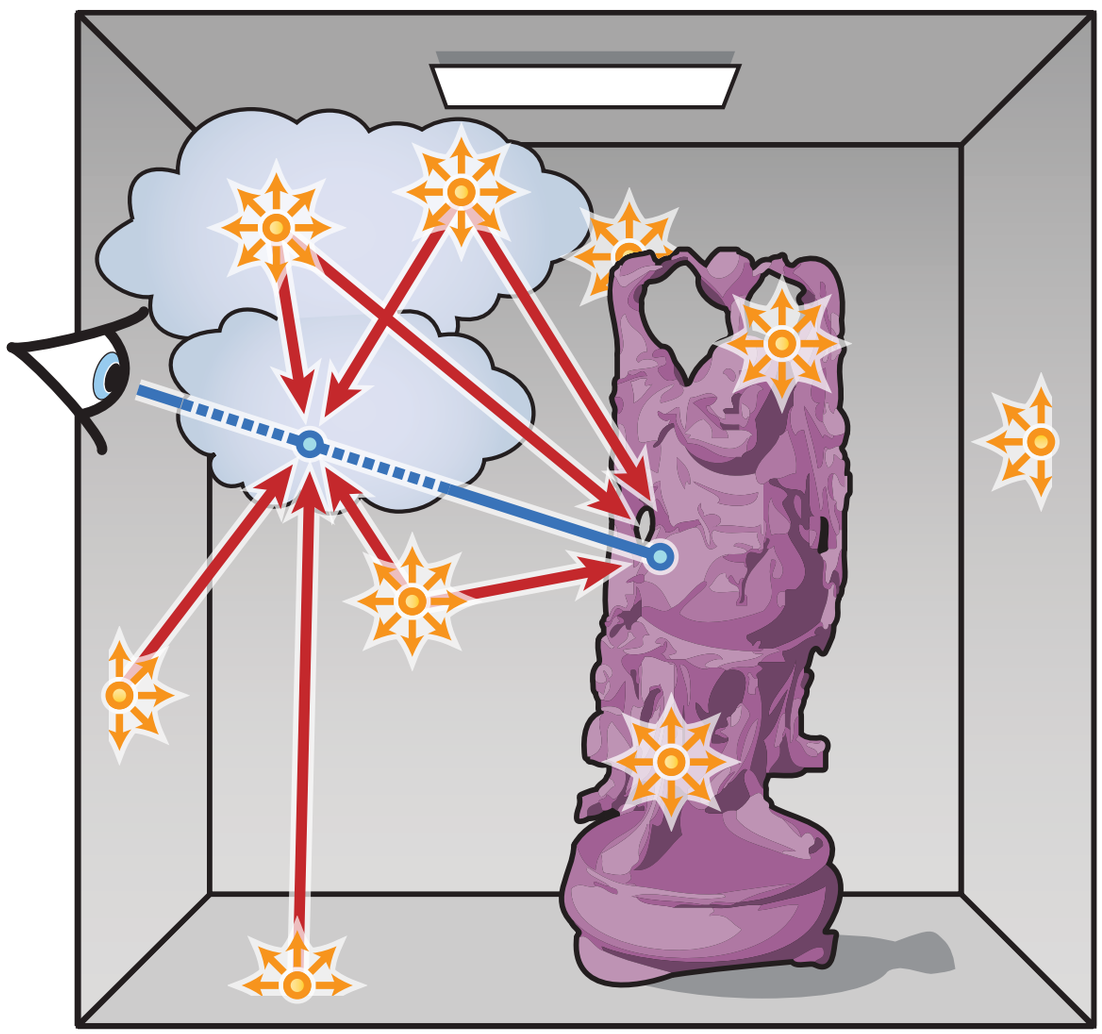
emission — scattered once — scattered twice

$$L_o = \sum_{i=0}^{\infty} (\mathbf{KG})^i L_e$$

Next Lecture

- Many-Light Rendering
  - "specific flavor of bidirectional path tracing"





References

- Veach E., *Robust Monte Carlo methods for light transport simulation*, PhD thesis, Stanford University, 1997. http://www.graphics.stanford.edu/papers/veach_thesis/
- Feynman, R. P. and Hibbs, A. R., *Quantum Mechanics and Path Integrals*, New York: McGraw-Hill, 1965.