# Path Tracing in Production

- part 1: modern path tracing
- part 2: making movies

# Introduction (Johannes Hanika)

# Motivation

**continued effort to:**

- document and discuss the switch of the movie industry to path tracing
- educate researchers, artists, programmers, .. about the particular requirements of making movies
- this year: fundamentals/basics/advanced techniques
  - glimpse at theoretical background
  - entry points for research
  - state of the art
  - toolset for improving rendering systems
- bring together industry experts

# Speakers (part 1)

| | |
|---|---|
| **Johannes Hanika** | **Weta Digital, me** |
| Luca Fascione | Weta Digital, Head of Technology and Research |
| Marc Droske | Weta Digital, Head of Rendering Research |
| Jorge Schwarzhaupt | Weta Digital, Researcher + Manuka wizard |
| Christopher Kulla | SPI, principal software engineer + Arnold veteran |
| Daniel Heckenberg | Animal Logic, R&D Supervisor, ASWF chair |





# Course notes

- What do you need to know to write better path tracers?
- please see our course notes for a lot more detail!

https://jo.dreggn.org/path-tracing-in-production/2019/

# Path tracing in Production

- part 1 and part 2
- part 2 follows in the afternoon
  - here, 403AB, 2pm-5:15pm
  - more about material acquisition, modelling, no Lions, and GPUs

# Part 1: Modern path tracing

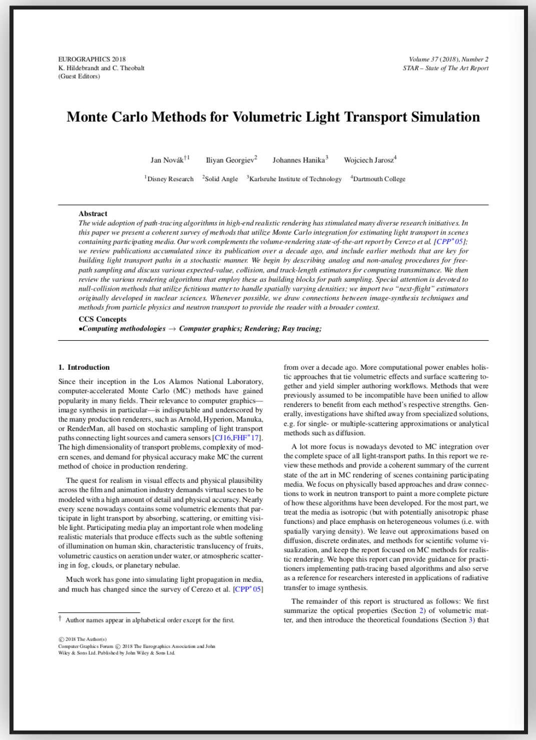
## how to increase visual quality? (in order of direct relevance for movies)

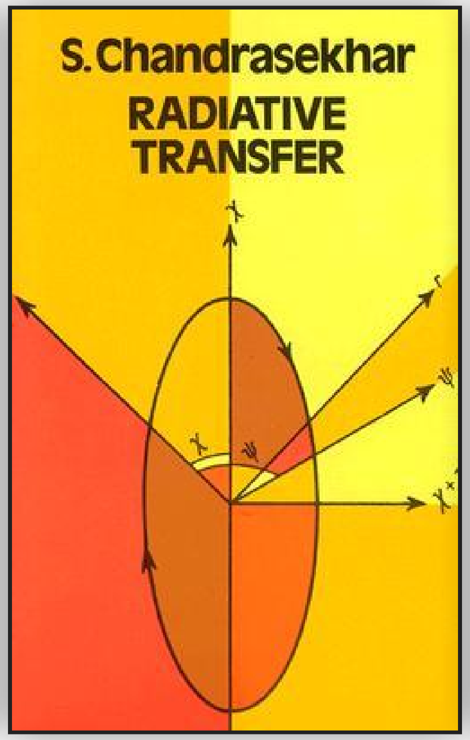
- Physics! (follows now)
- Mathematics! (Luca)
- Designing an architecture for Monte Carlo! (Marc)
- Special sauce sampling! (Jorge)
- Smoke and explosions! (Chris)
- Mangling crazy complex input! (Daniel)

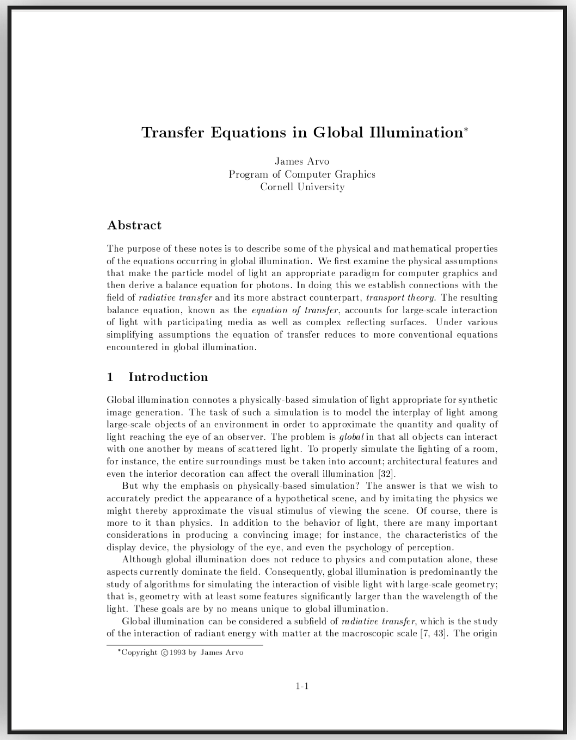
# Physics!

# Recommended literature

- Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz,
  *Monte Carlo methods for volumetric light transport simulation*
  Computer Graphics Forum (Eurographics State of the Art Reports), 2018.
- Subrahmanyan Chandrasekhar,
  *Radiative transfer*,
  Dover Publications, 1960
- James Arvo,
  *Transfer Equations in Global Illumination*
  SIGGRAPH Course Notes, 1993


Monte Carlo Methods for Volumetric Light Transport Simulation





Transfer Equations in Global Illumination

James Arvo
Program of Computer Graphics
Cornell University

Abstract

1 Introduction

# Photons

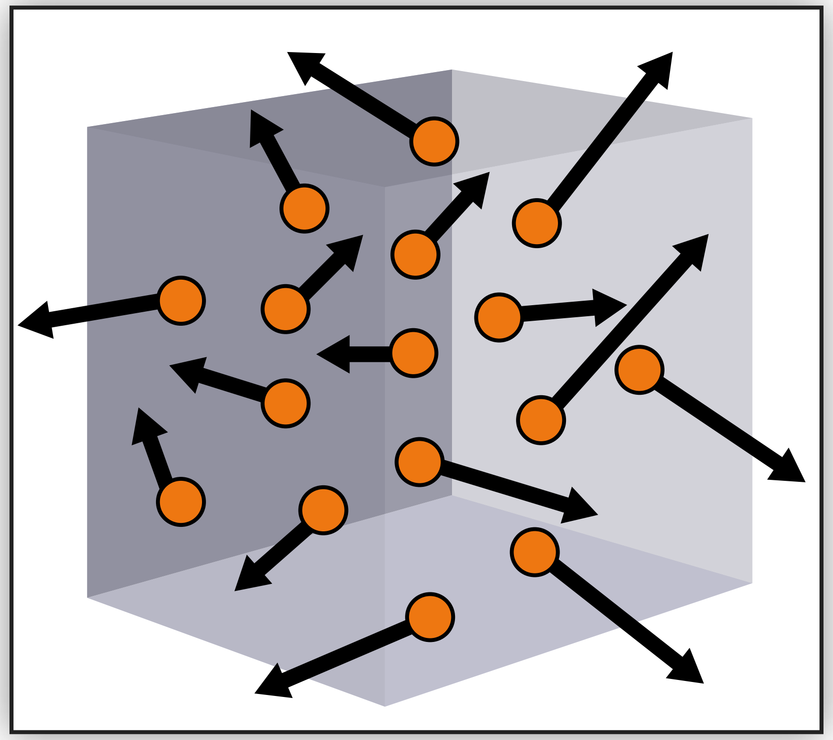
- particle/wave dualism
  - we'll go with particles (for the most part), with a position and direction
- a **photon** corresponds to an atomic portion of **energy** $E$ (measured in Joule $[J]$)

$$E = \frac{h \cdot c_m}{\lambda} \; [J]$$

  - where $h \approx 6.62607004 \times 10^{-34} \; [m^2 kg/s = Js]$ is Planck's constant,
  - $c_m = c/\eta_m$ is the **speed of light** in the material with **index of refraction** $\eta_m$, and $c = 299,792,458 \; [m/s]$ is the speed of light in vacuum,
  - and $\lambda \; [m]$ is the **wavelength** (often given in $[nm]$ instead to distinguish from world space lengths).
- we'll need a few ways to measure the energy of light

# Phase space

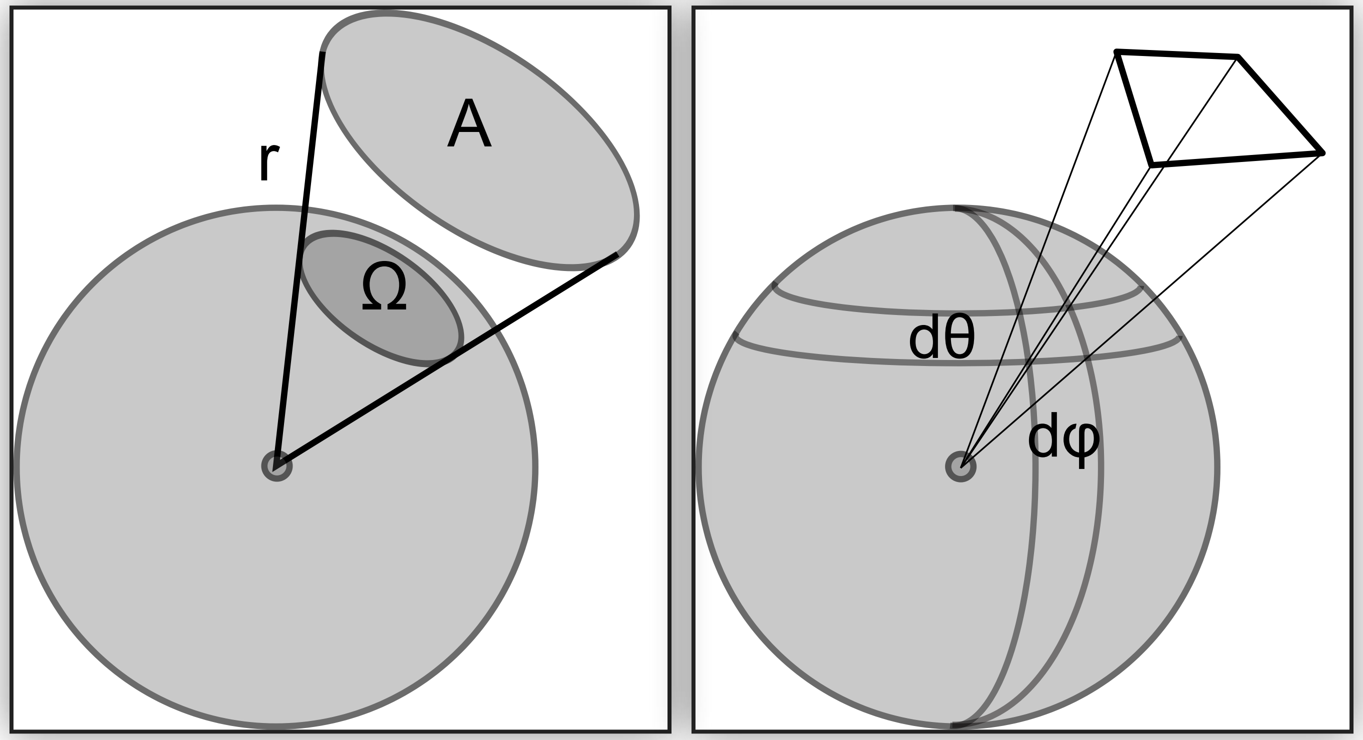
- a photon lives in 5D **phase space**: 3D position $\mathbf{x}$ and 2D direction $\omega$



- positions are in **meters** $[m]$
- directions in **steradian** $[sr]$
  - dimensionless, derived SI unit of solid angle

# Solid angle

- a direction is defined on the unit sphere, areas on this sphere are called **solid angle**



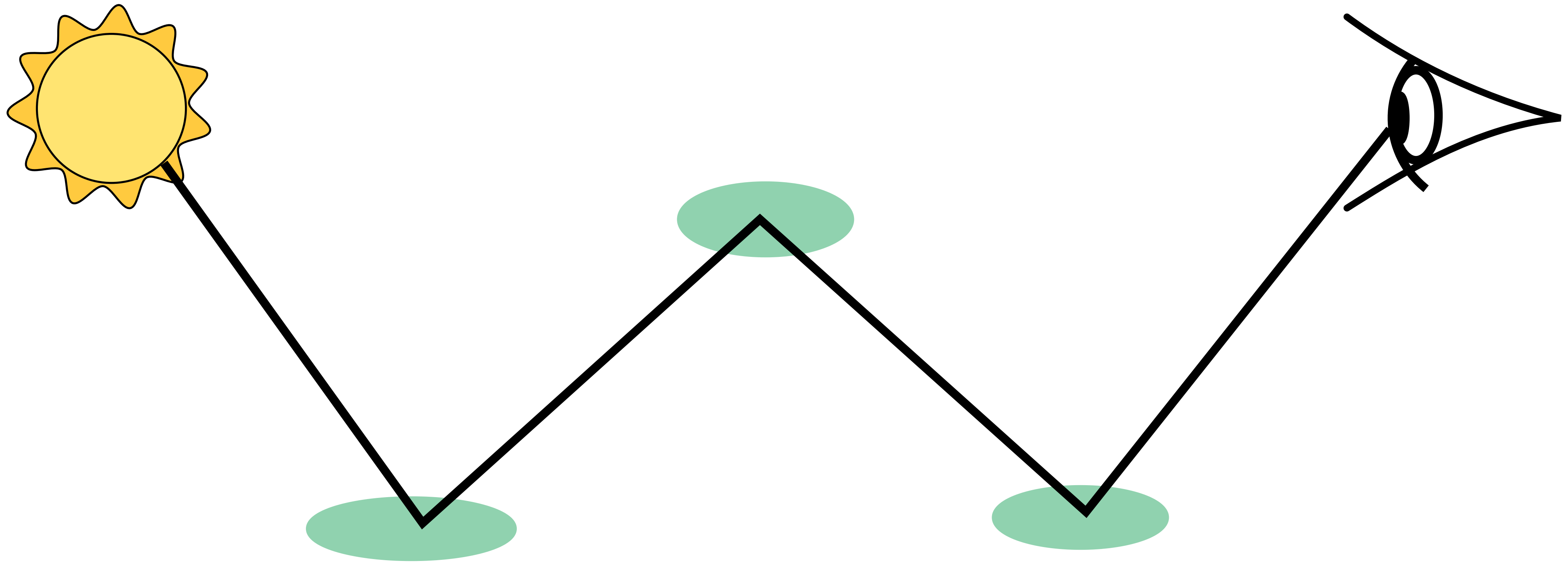

- $\Omega_A = A/r^2\ [sr]$
- as differential: $\mathrm{d}\omega$ or $\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$ in polar form, since $\int_\Omega \mathrm{d}\omega = \int_0^{2\pi}\int_0^{\pi} \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi = 4\pi$
- we will refer to directions as $\omega$ and $\Omega = 4\pi$

# Radiance

- goal: want to measure light along a ray suitable for ray tracing
  - because we can only really evaluate visibility on a straight line between two points
  - spoiler:

$$L(\mathbf{x}, \omega) \ \left[\frac{W}{m^2\, sr}\right]$$

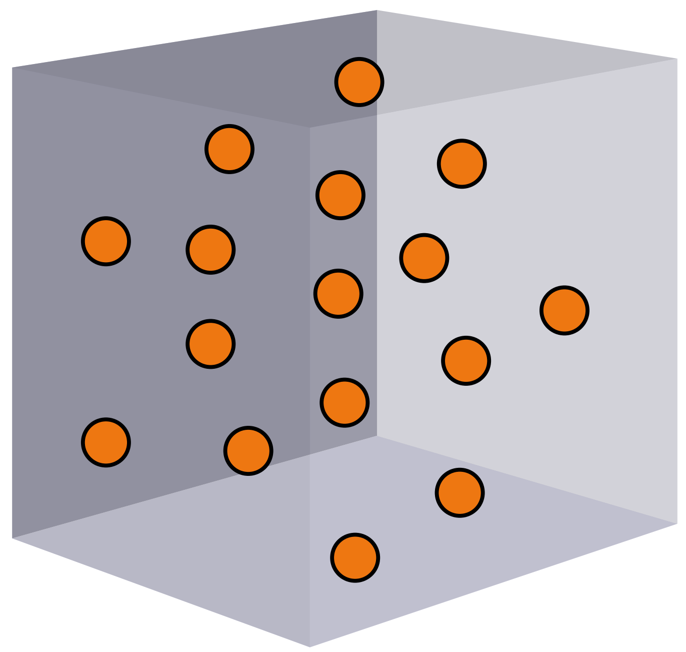
- but first: let's count some photons!



# Radiant energy

- count number $\#P$ of photons inside a certain volume, multiply by their energy $E$

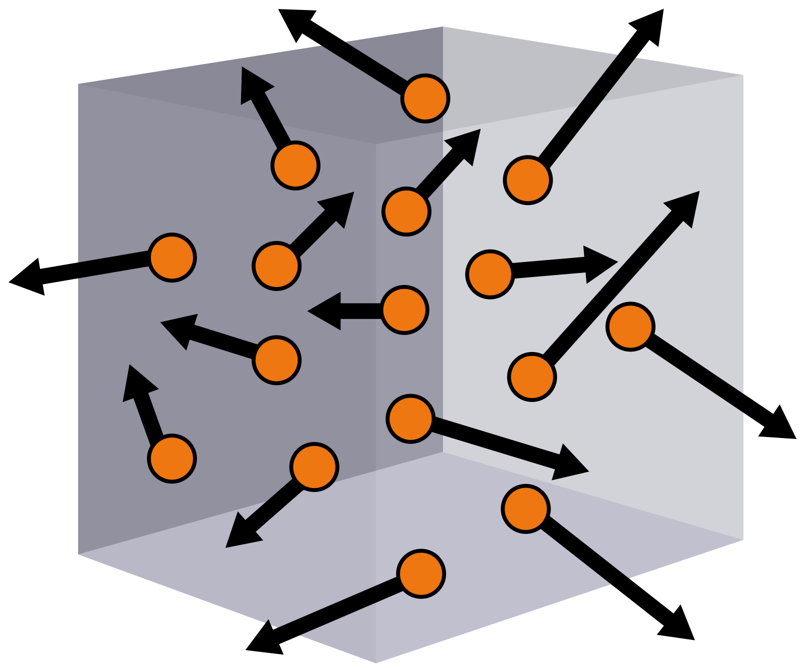
$$Q = \#P \cdot \frac{h \cdot c_m}{\lambda} \left[ J = \frac{kg \cdot m^2}{s^2} \right]$$



# Radiant power or flux

- count photons inside certain volume per time (measured in watts):

$$\Phi = \frac{\mathrm{d}Q}{\mathrm{d}t} \left[ \frac{J}{s} = W \right]$$

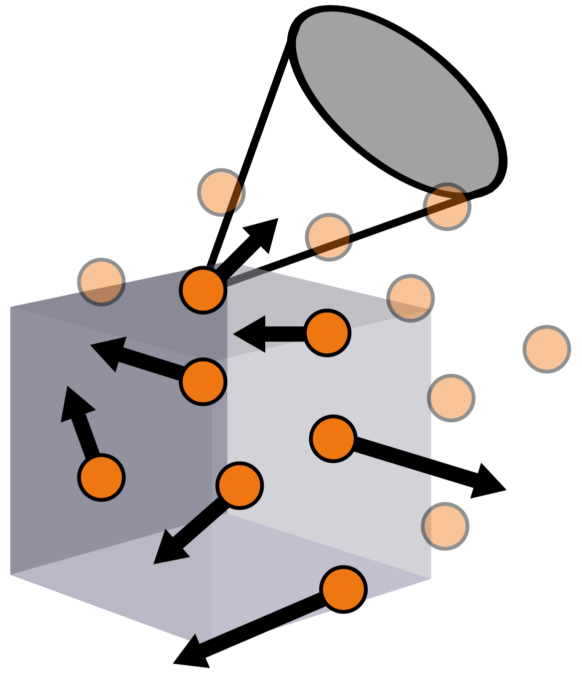


# Ideally: measure light

- count photons per time in a volume $V$ in the 5D phase space over 3D positions and 2D directions $(\mathbf{x}, \omega)$

$$\Phi = \int_\Omega \int_V P(\mathbf{x}, \omega)\, \mathrm{d}\mathbf{x}\mathrm{d}\omega \; [W]$$

- $P(\mathbf{x}, \omega)$ is a symbol "counting" photons per time going through a certain point and direction in phase space
- note that $\mathbf{x}$ is a 3D volume point here measured in cubic meters

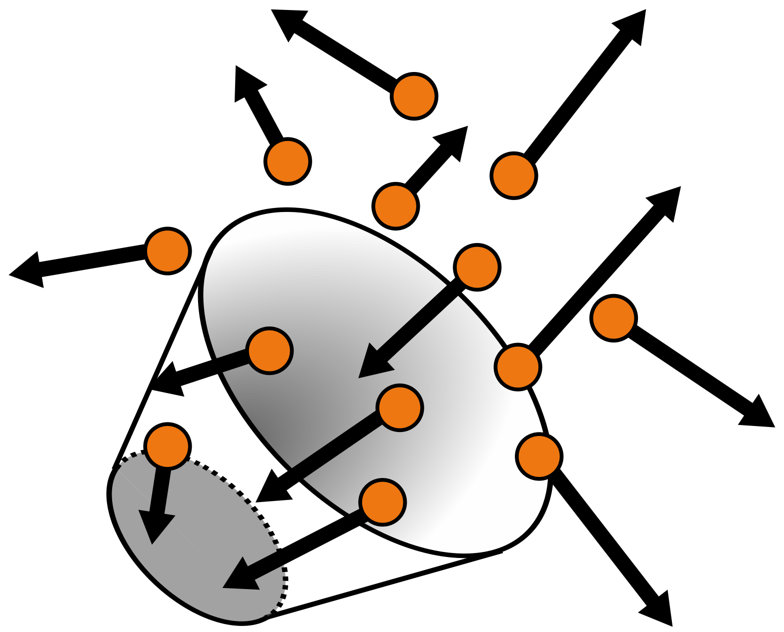


# Measure radiance

- power per area per solid angle (measured in watts per square meter per steradian)

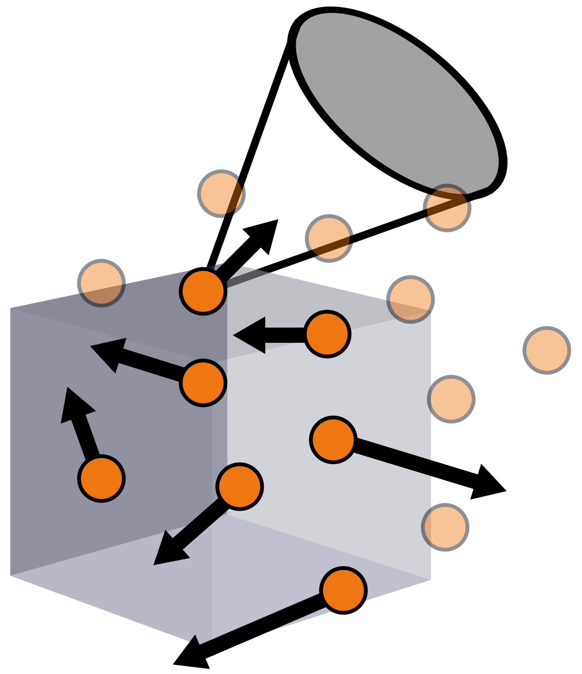
$$L(\mathbf{x}, \omega) = \frac{\mathrm{d}\Phi}{\mathrm{d}\mathbf{x}\mathrm{d}\omega} \left[ \frac{W}{m^2 sr} \right]$$

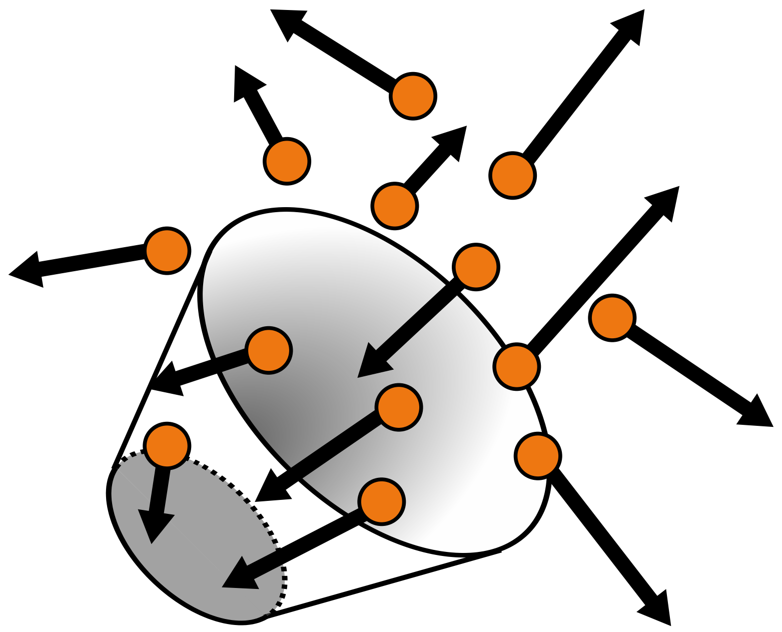
- *the* central unit for us in rendering
- this is what a ray of light transports
- differentially small measurement apparatus: area with funnel



# Measure radiance?

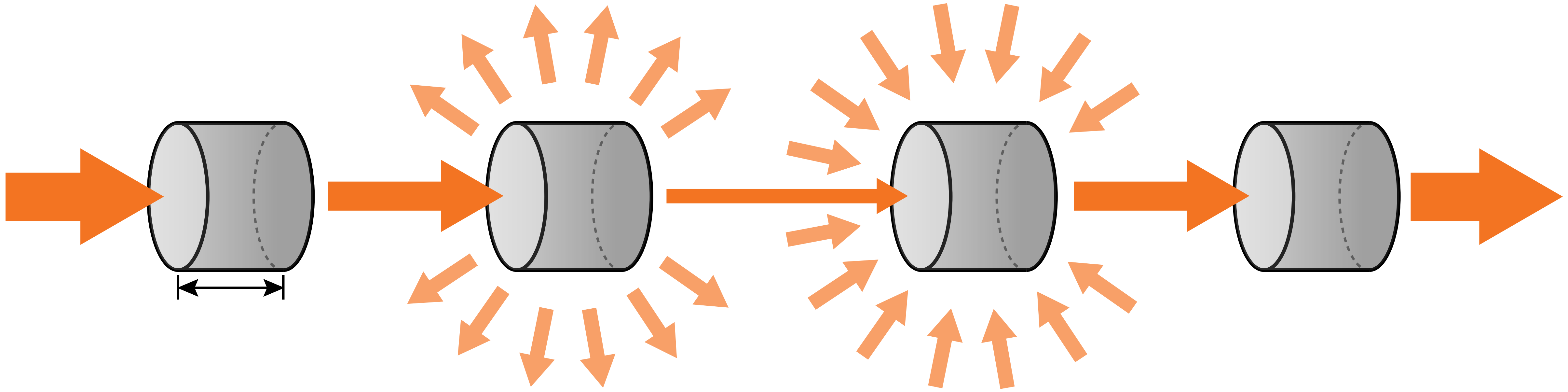
- dynamic system, light is in flow
- describe changes of radiance in phase space and solve for steady-state equilibrium!





# Losses and gains

- three effects change the observed amount of light:
  - **collision**: light interacts with matter
  - **emission**: light is emitted from matter
  - **streaming**: light enters or leaves a volume
- collision may incur **scattering**: change of direction



# The radiative transfer equation (RTE)

## summing all terms

- all terms need to sum to zero (energy conservation)

$$0 = \int_\Omega \int_V - \overbrace{\frac{\partial}{\partial \omega} L(\mathbf{x}, \omega)}^{\text{streaming}} + \overbrace{\mu_e(\mathbf{x}) L_e(\mathbf{x}, \omega)}^{\text{emission}} \overbrace{- \mu_t(\mathbf{x}) L(\mathbf{x}, \omega)}^{\text{extinction}}$$
$$+ \underbrace{\mu_s(\mathbf{x}) \int_\Omega f_s(\omega_i \cdot \omega) L(\mathbf{x}, \omega_i) \mathrm{d}\omega_i}_{\text{in-scattering}} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\omega$$

- if the equations hold for integration over any part of phase space, they have to hold for every individual point, too
- leave away integration over phase space $\Omega \times V$

# The radiative transfer equation (RTE)

## summing all terms

- all terms need to sum to zero (energy conservation), lose phase space integral

$$\overbrace{\frac{\partial}{\partial \omega} L(\mathbf{x}, \omega)}^{\text{streaming}} = \overbrace{\mu_e(\mathbf{x}) L_e(\mathbf{x}, \omega)}^{\text{emission}} \overbrace{- \mu_t(\mathbf{x}) L(\mathbf{x}, \omega)}^{\text{extinction}}$$
$$+ \underbrace{\mu_s(\mathbf{x}) \int_\Omega f_s(\omega_i \cdot \omega) L(\mathbf{x}, \omega_i) \mathrm{d}\omega_i}_{\text{in-scattering}}$$

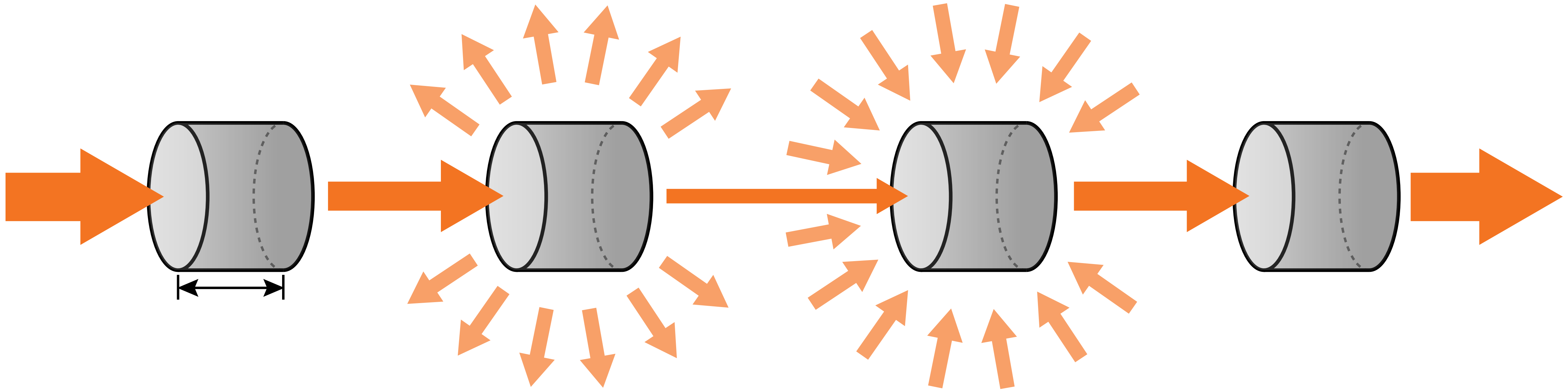
- we are interested in changes in radiance $L(\mathbf{x}, \omega)$
- the directional derivative $\frac{\partial}{\partial \omega} L(\mathbf{x}, \omega)$ (LHS) is change of radiance in direction $\omega$

# The radiative transfer equation (RTE)

## summing all terms

$$\frac{\partial}{\partial \omega} L(\mathbf{x}, \omega) = \mu_e(\mathbf{x}) L_e(\mathbf{x}, \omega) - \mu_t(\mathbf{x}) L(\mathbf{x}, \omega) + \mu_s(\mathbf{x}) \int_\Omega f_s(\omega_i \cdot \omega) L(\mathbf{x}, \omega_i) \mathrm{d}\omega_i$$

- integro differential equation along one ray $(\mathbf{x}, \omega)$
- governs scattering in participating media (volumes)
- we need to bring it to pure integral form for graphics applications!



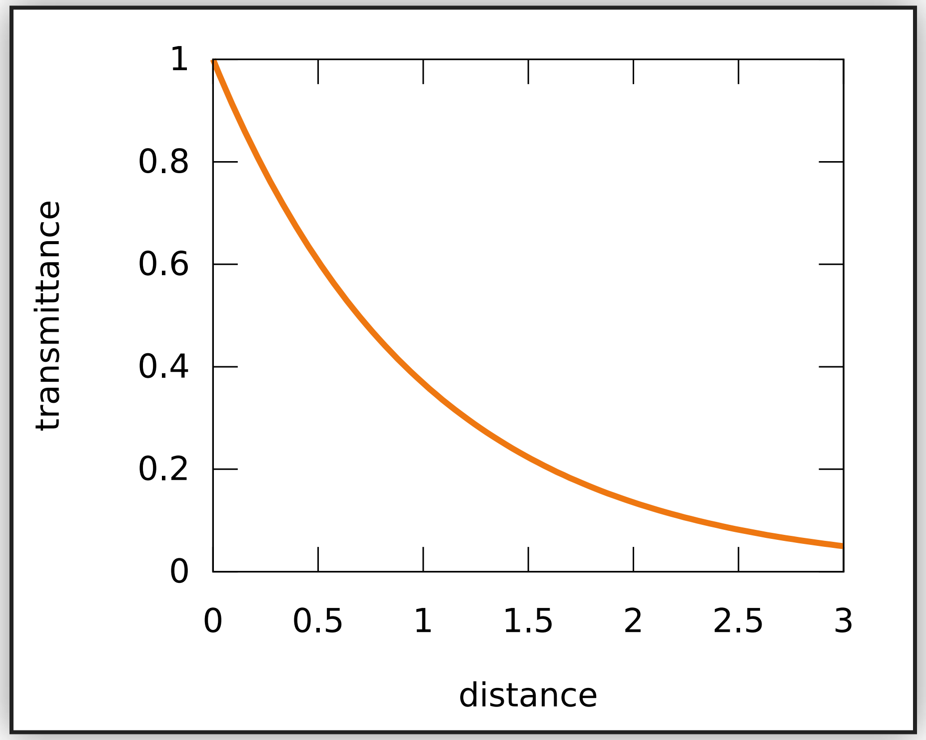
# Integrating the RTE

- example: solution of simple one-dimensional differential equation

$$\frac{\mathrm{d}}{\mathrm{d}x} L(x) = -\mu_a L(x) \text{ with } x > 0,\ L(0) = 1$$

  - closed-form solution using the exponential function:

$$L(x) = \exp(-\mu_a \cdot x) \cdot L(0)$$



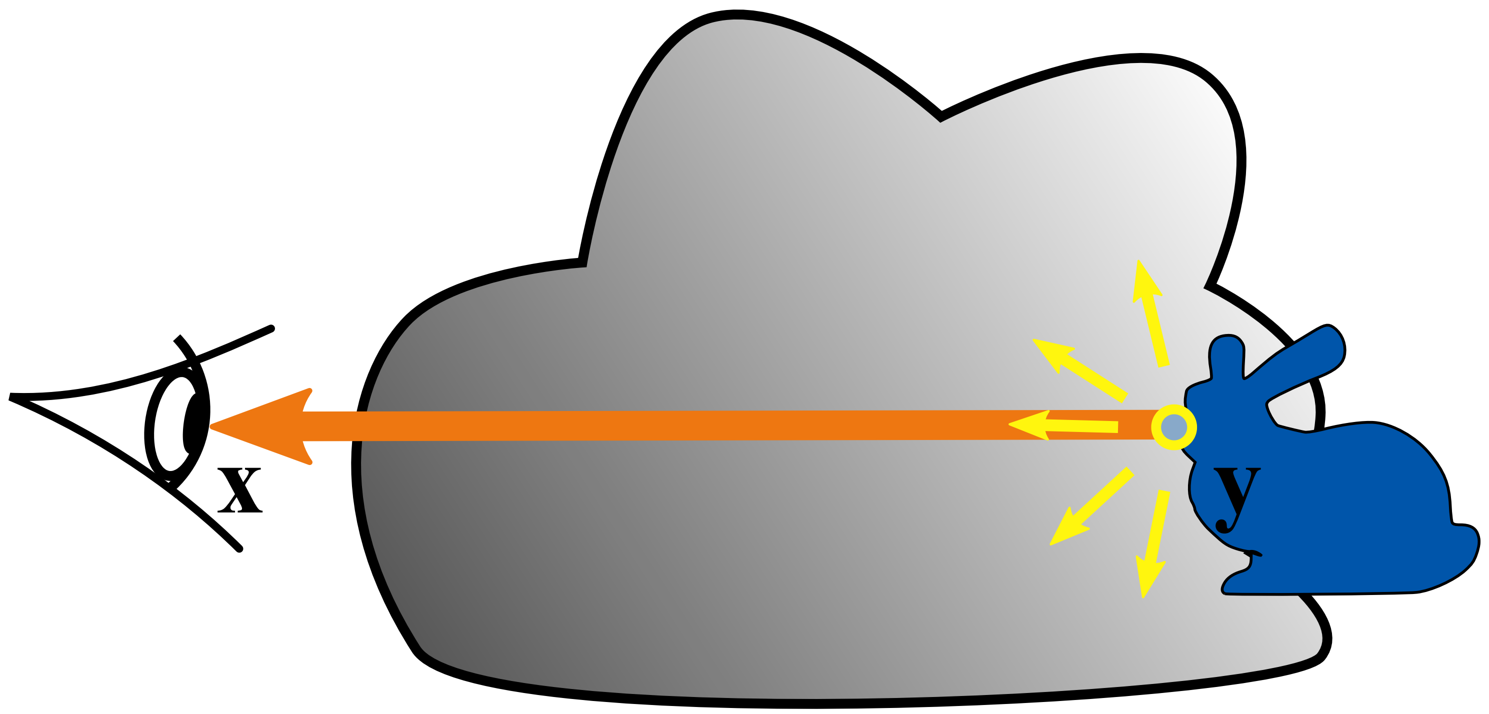

- gives rise to the *transmittance* term $T(\mathbf{x}, \mathbf{y})$

# The RTE in integral form

## Integrating the RTE: intuition

- contribution from surface emission (exactly one point $\mathbf{y}$)

$$L(\mathbf{x}, \omega) \mathrel{+}= T(\mathbf{x}, \mathbf{y}) \cdot L_e(\mathbf{y}, \omega)$$

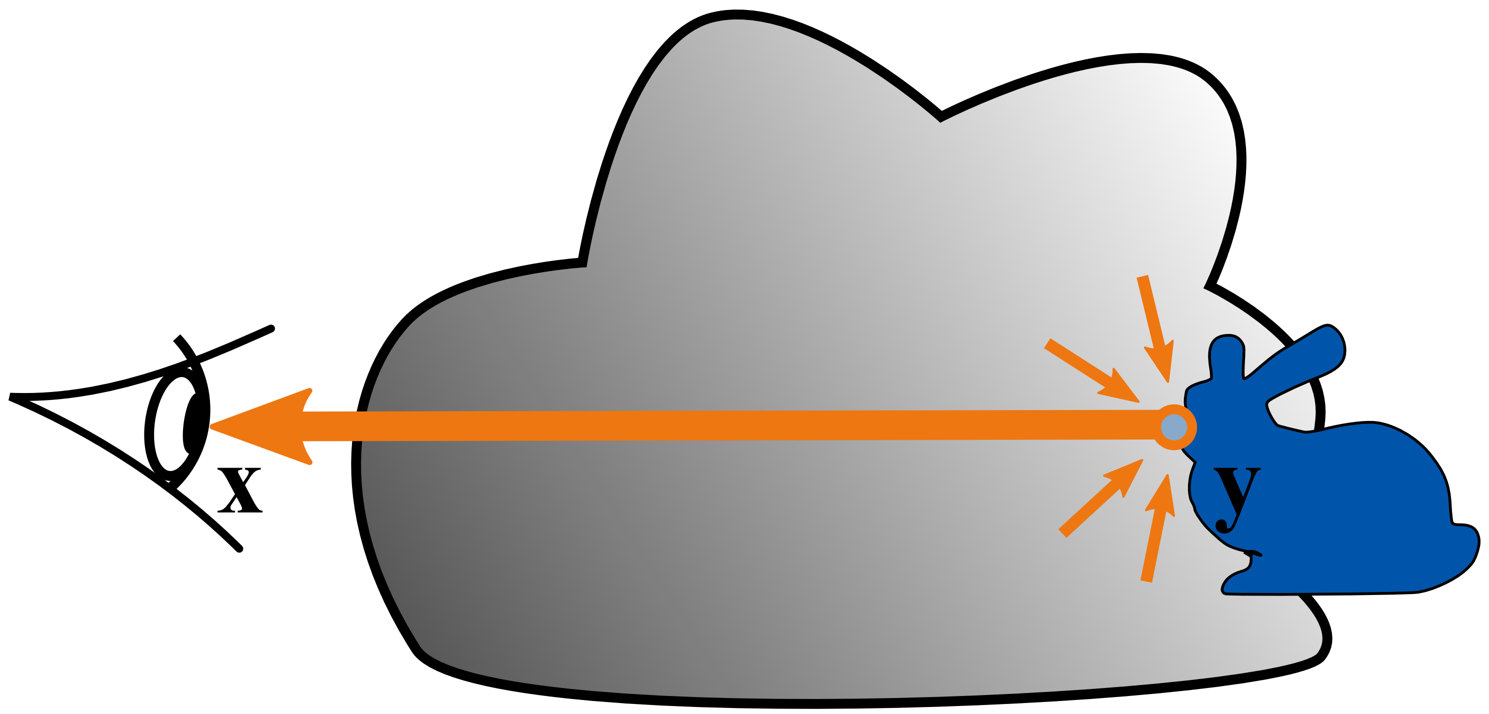


# The RTE in integral form

## Integrating the RTE: intuition

- contribution from surface scattering (exactly one point $\mathbf{y}$)

$$L(\mathbf{x},\omega) \mathrel{+}= T(\mathbf{x},\mathbf{y}) \cdot \int_{\Omega} f_r(\omega_i,\mathbf{y},\omega)L(\mathbf{y},\omega_i)\mathrm{d}\omega_i^{\perp}$$

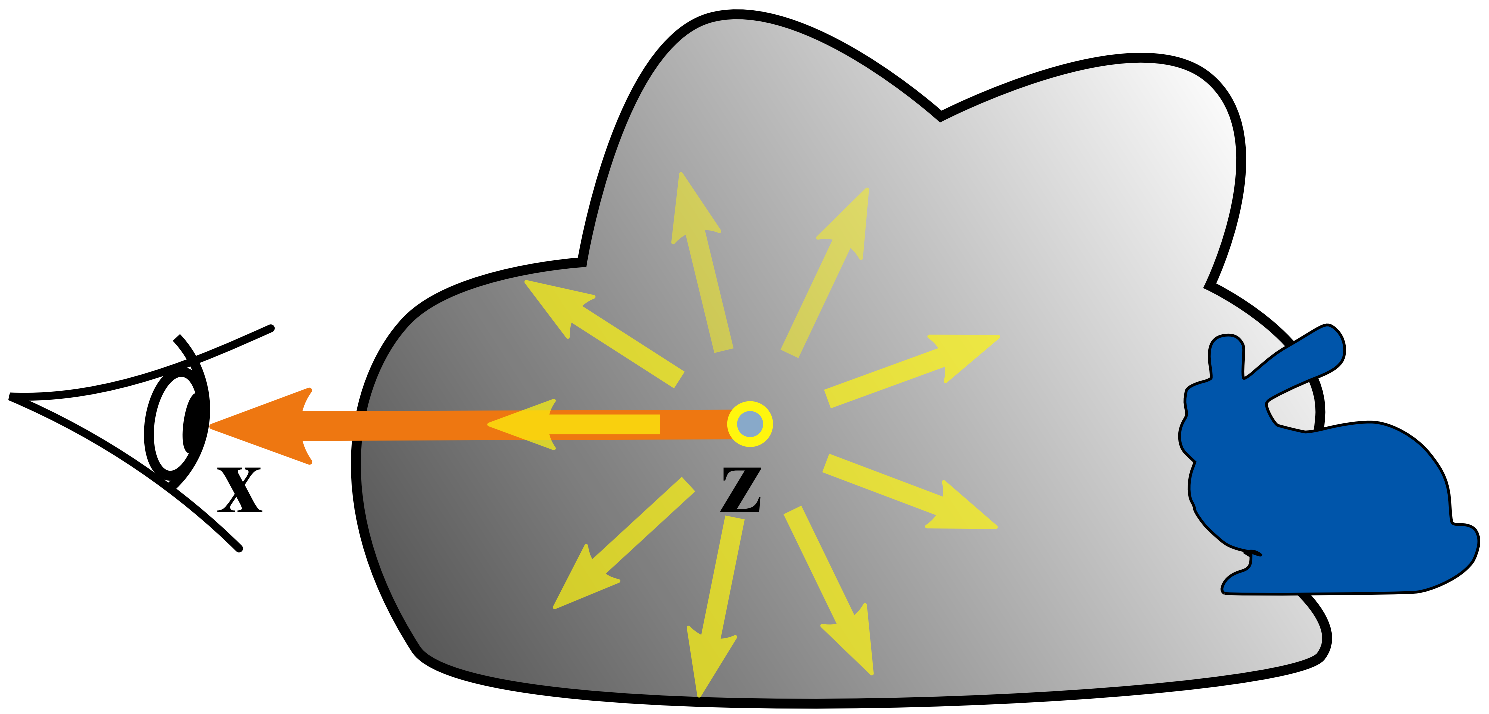


# The RTE in integral form

## Integrating the RTE: intuition

- contribution from volume emission (need to collect all $\mathbf{z}$)

$$L(\mathbf{x},\omega) \mathrel{+}= T(\mathbf{x},\mathbf{z}) \cdot \mu_e(\mathbf{z}) L_e(\mathbf{z},\omega)$$

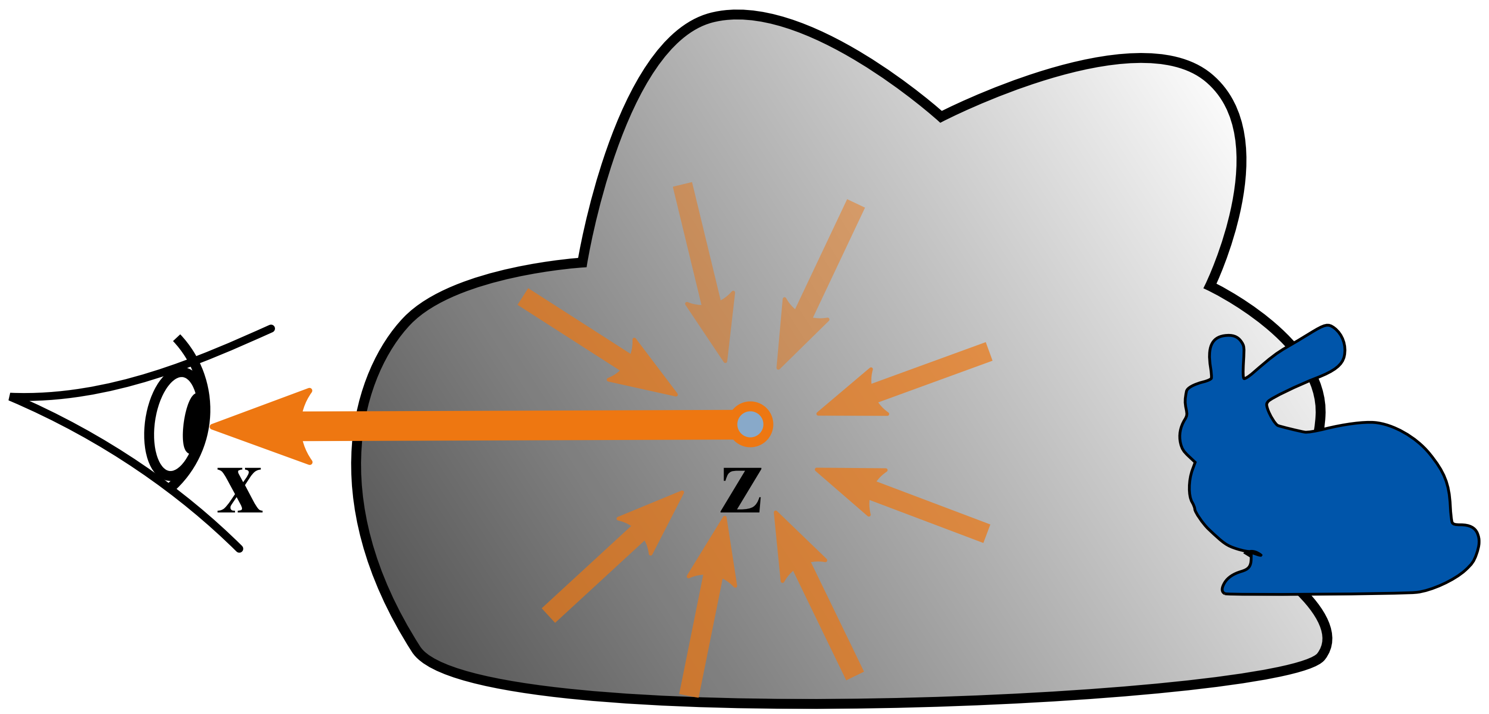


# The RTE in integral form

## Integrating the RTE: intuition

- contribution from volume scattering (need to collect all $\mathbf{z}$)

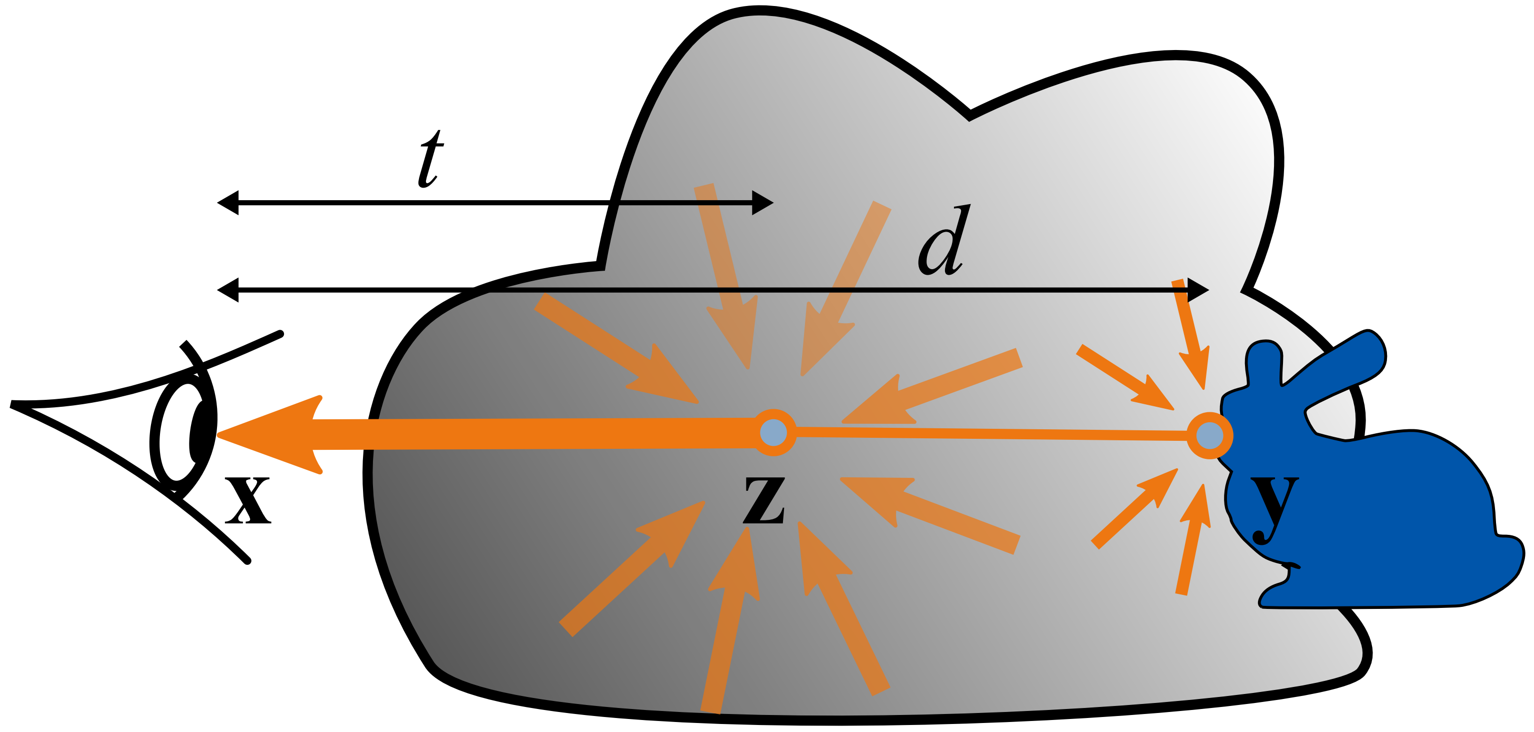
$$L(\mathbf{x},\omega) \mathrel{+}= T(\mathbf{x},\mathbf{z}) \cdot \mu_s(\mathbf{z}) \int_\Omega f_s(\omega \cdot \omega_i) L(\mathbf{z},\omega_i) \mathrm{d}\omega_i$$



# The RTE in integral form

$$L(\mathbf{x},\omega) = T(\mathbf{x},\mathbf{y})\overbrace{\left(L_e(\mathbf{y},\omega) + \int_\Omega f_r(\omega_i,\mathbf{y},\omega)L(\mathbf{y},\omega_i)\mathrm{d}\omega_i^\perp\right)}^{\text{contribution from the point }\mathbf{y}\text{ on surface}}$$

$$+\int_0^d T(\mathbf{x},\mathbf{z})\underbrace{\left(\mu_e(\mathbf{z})L_e(\mathbf{z},\omega) + \mu_s(\mathbf{z})\int_\Omega f_s(\omega\cdot\omega_i)L(\mathbf{z},\omega_i)\mathrm{d}\omega_i\right)}_{\text{contribution from any point }\mathbf{z}\text{ at distance }t\text{ in volume}}\mathrm{d}t$$



# Recursive rendering equation

- light is emission + transported light (either from surface or volume)

$$L = L_e + \mathbf{T}L$$

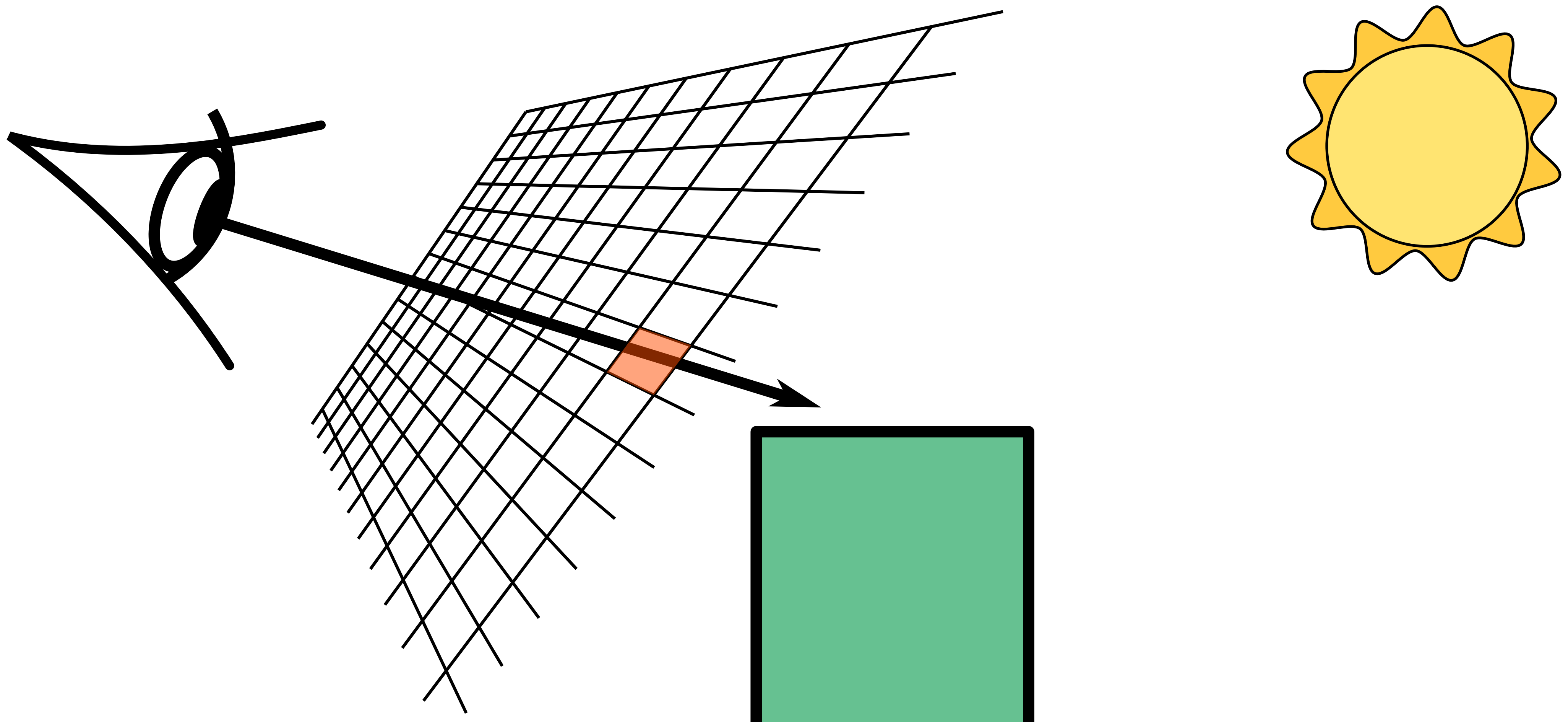
- Neumann series ($\mathbf{T}$ is a linear operator):

$$L = (1 - \mathbf{T})^{-1} L_e = \sum_{i=0}^{\infty} \mathbf{T}^i L_e$$

- turns recursion into sum over all path lengths!

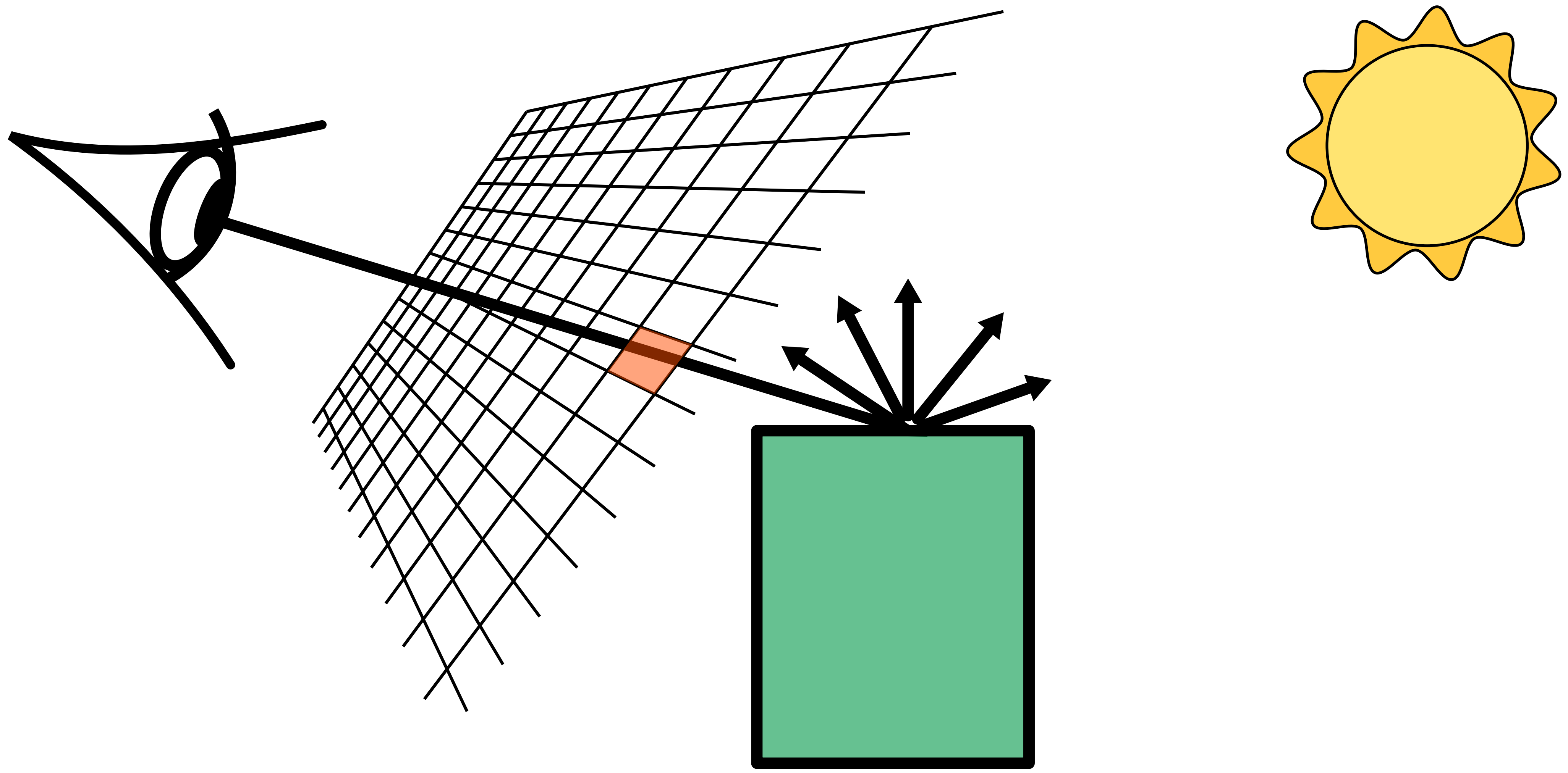
# Tracing paths the recursive way

- means tracing rays, usually through every pixel



# Tracing paths the recursive way

- means tracing rays, usually through every pixel



- recursive rendering equation: light is emission + transported light

$$L = L_e + \mathbf{T}L$$

# The rendering equation in path integral form

- expand using the Neumann series to arrive at path space

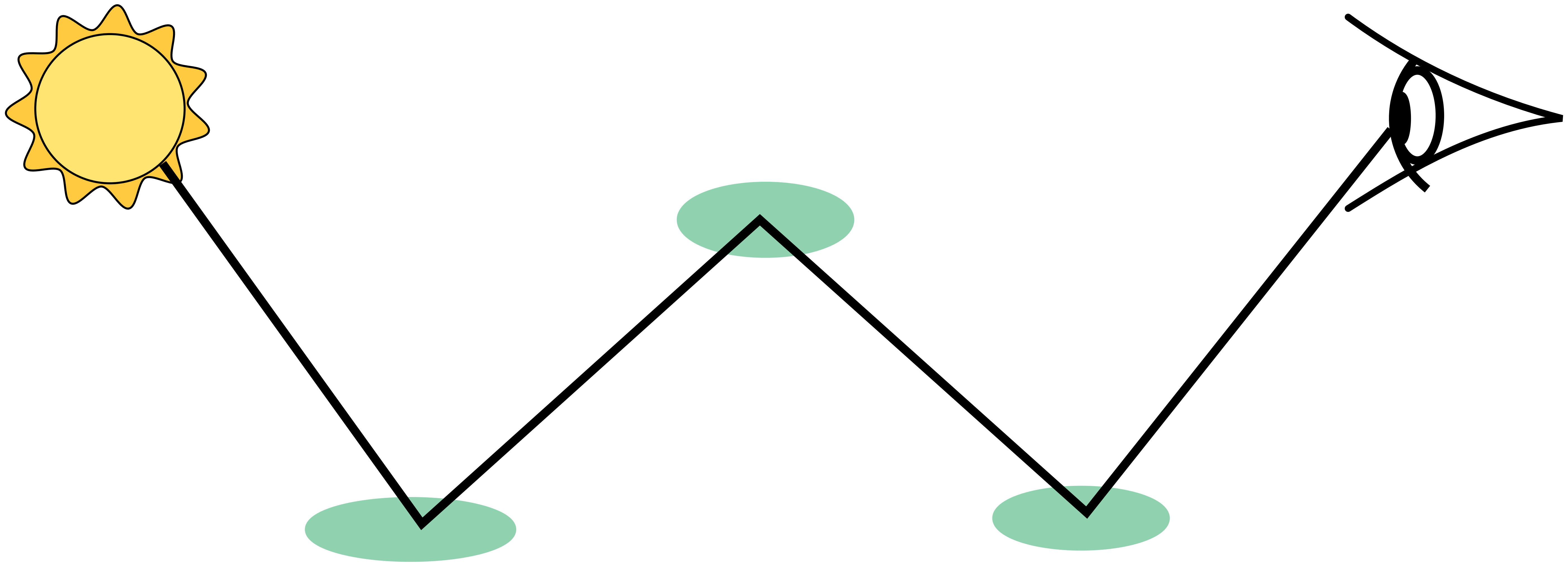
$$I_p = \int_{\mathcal{P}} h_p(\mathbf{X}) \cdot f(\mathbf{X}) \mathbf{d}\mathbf{X}$$

- $h_p(\mathbf{X})$ selects paths per pixel $p$ via pixel filter support
- path $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_k) \in \mathcal{P}$ is list of vertices $\mathbf{x}$
- $f(\mathbf{X})$ is the **measurement contribution function** in product vertex area measure $\mathbf{d}\mathbf{X}$

# The measurement contribution function

- measure differential power of path $\mathbf{X}$ with length $k$ ($= 5$ vertices here)

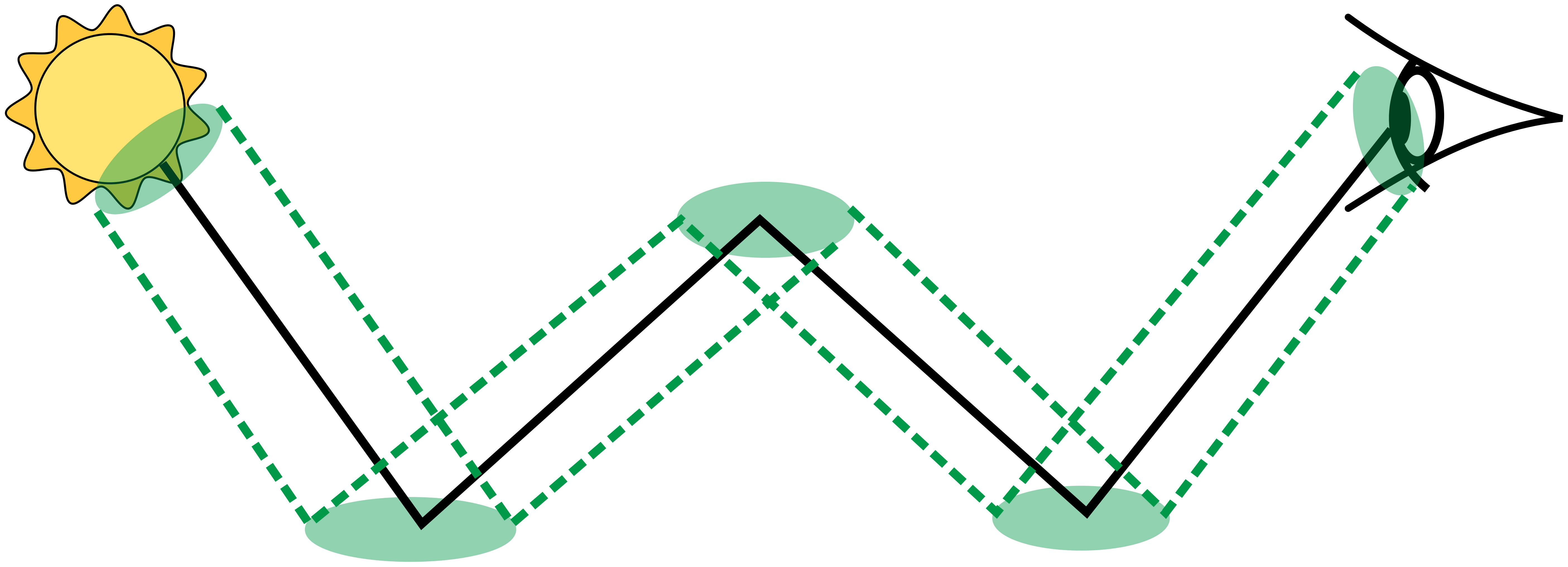
$$f(\mathbf{X}) = L_e G_{k-1} \left( \prod_{i=1}^{k-2} f_{r,i} G_i \right) W$$



# The measurement contribution function

- measure differential power of path $\mathbf{X}$ with length $k$ ($= 5$ vertices here)

$$f(\mathbf{X}) = \frac{\mathrm{d}^k \Phi}{\mathrm{d}^k \mathbf{x}} \qquad \left[\frac{W}{m^{2\cdot k}}\right]$$



# Monte Carlo integration

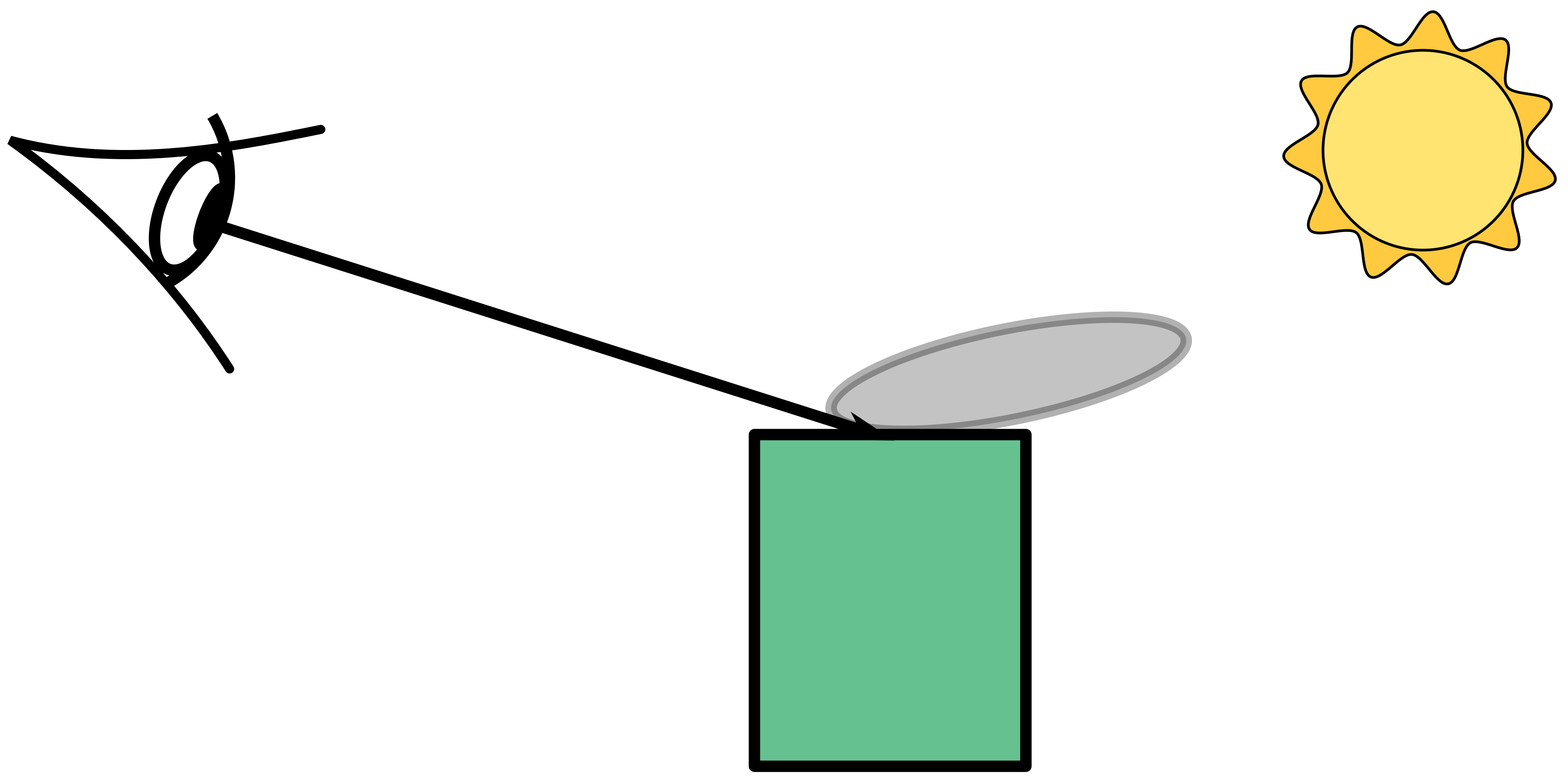
- approximate the integral by a Monte Carlo **estimator**

$$I_p \approx \frac{1}{N} \sum_{i=1}^{N} \frac{h_p(\mathbf{X}_i) \cdot f(\mathbf{X}_i)}{p(\mathbf{X}_i)}$$

- the expected value of the estimator is precisely the integral (the estimator is **unbiased**)
- error manifests itself as noise (**variance** of the estimator)
- how much noise for which path construction strategy determined by their PDF $p(\mathbf{X})$

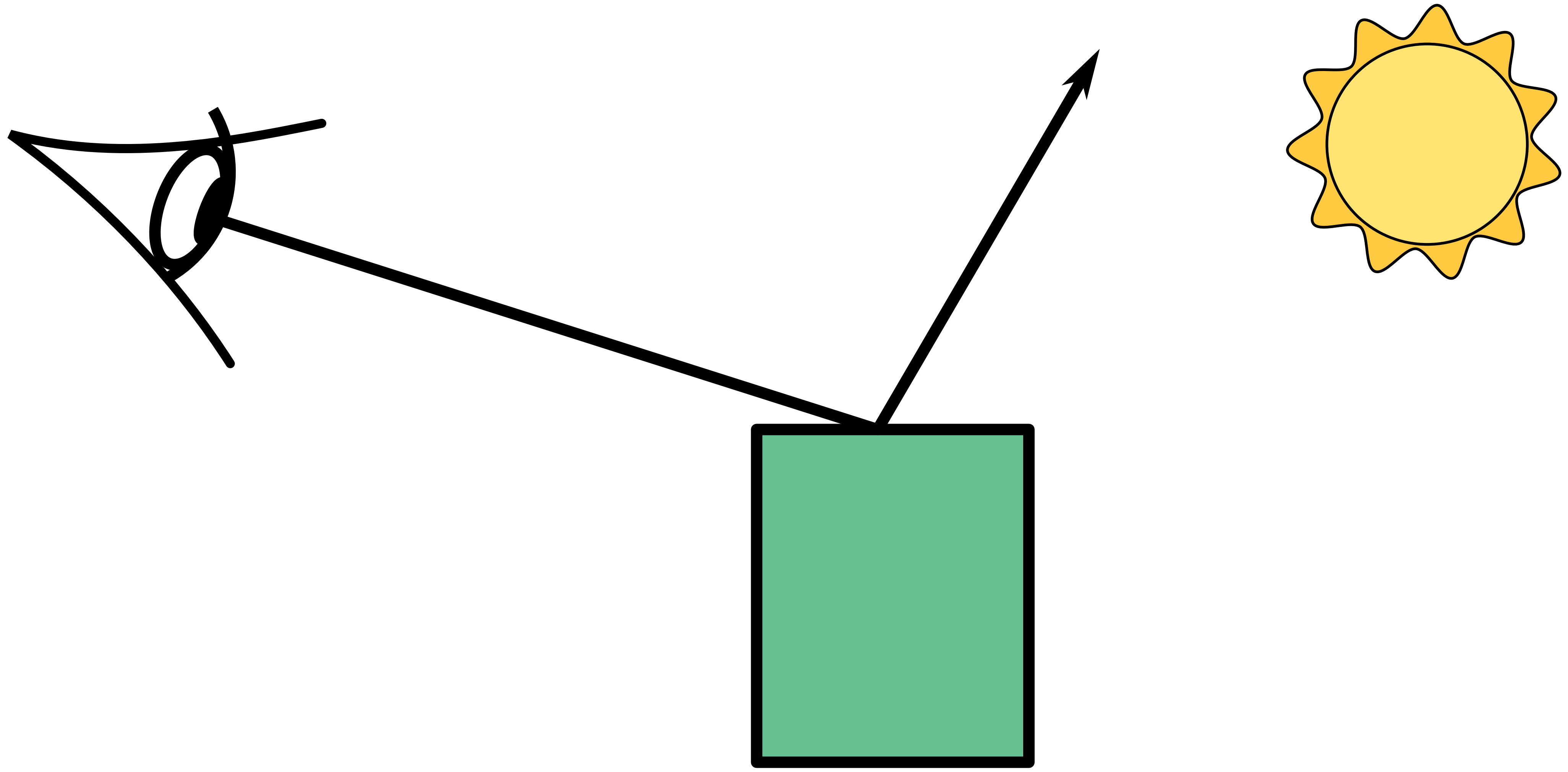
# Path tracing

- start constructing a path at the sensor
- sample outgoing direction locally by Bsdf



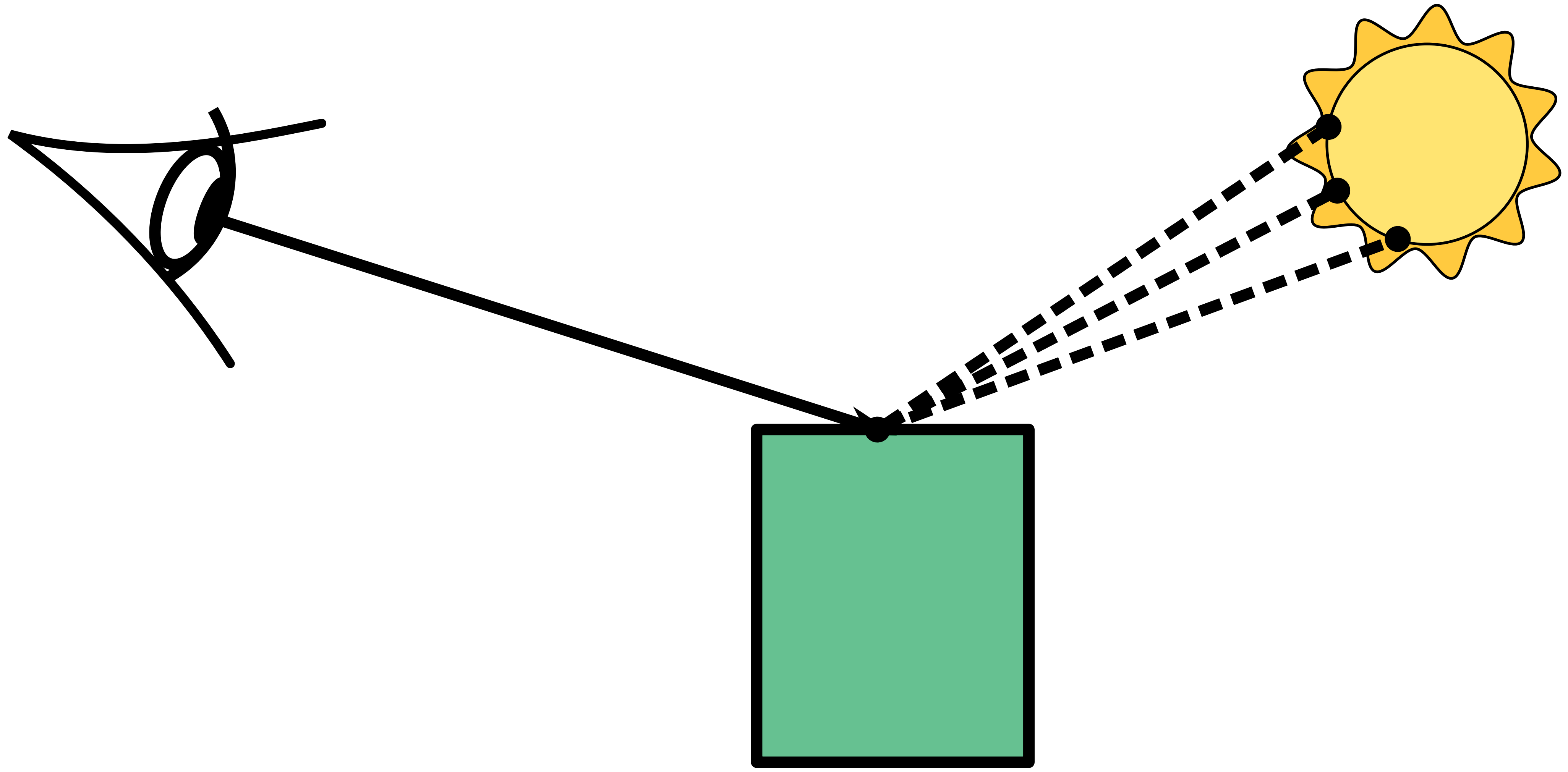
# Path tracing

- start constructing a path at the sensor
- problem intersecting light source by chance



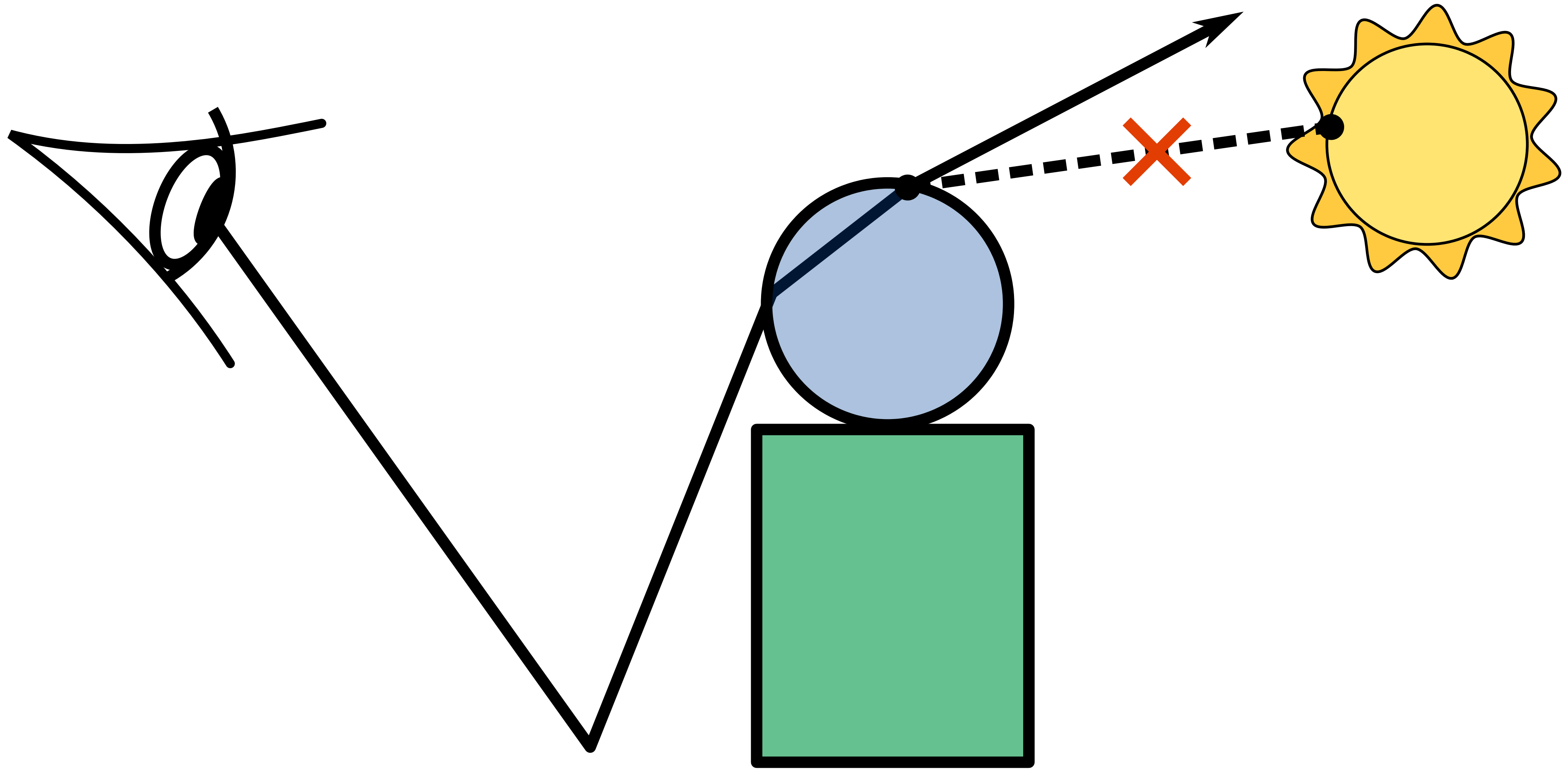
# Path tracing/next event estimation

- start constructing a path at the sensor
- direct connection(s) to light source in area measure



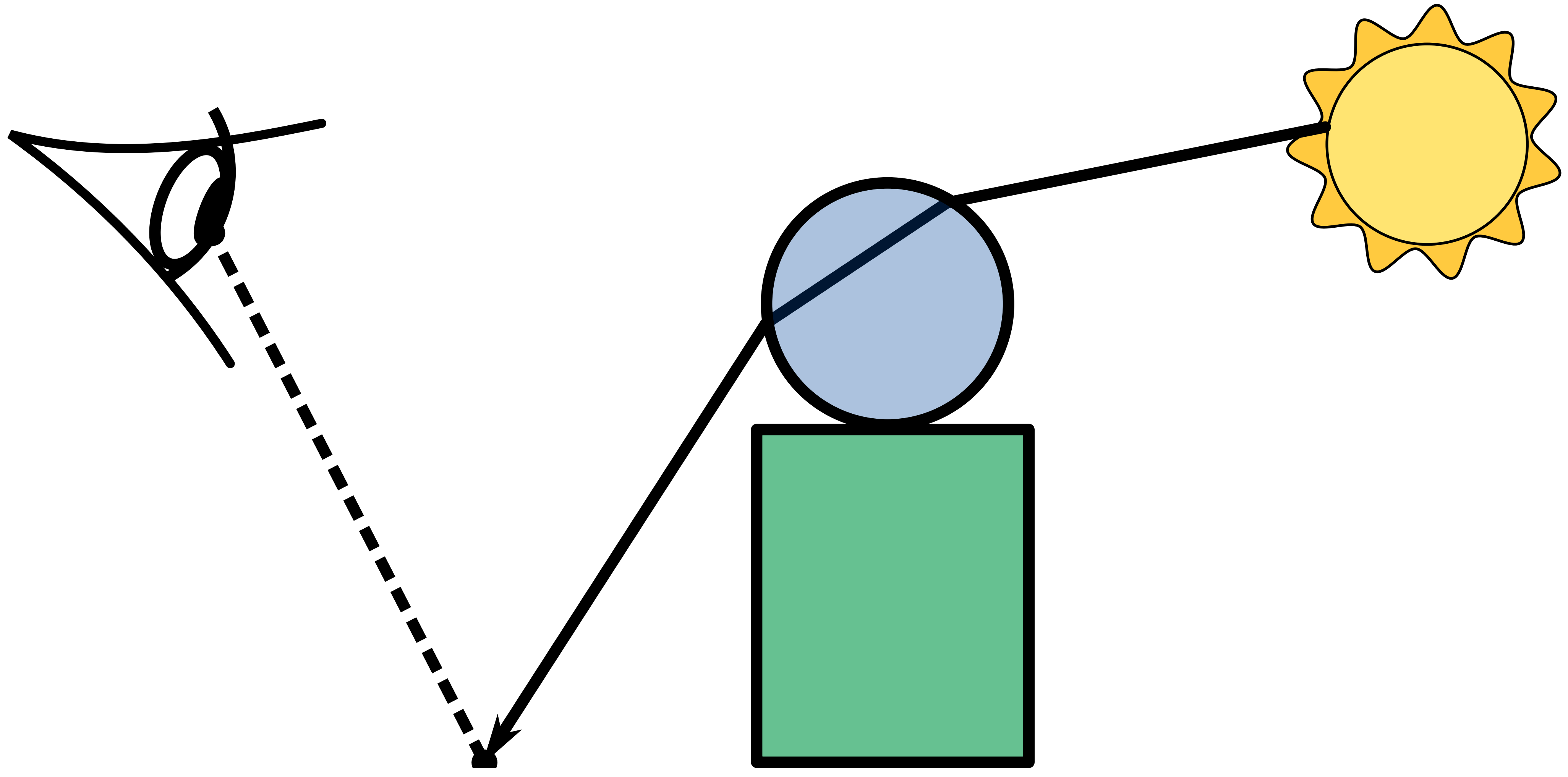
# Path tracing/next event estimation

- problem connecting glossy/specular materials
- Bsdf evaluates to zero (or close to for low roughness > 0)



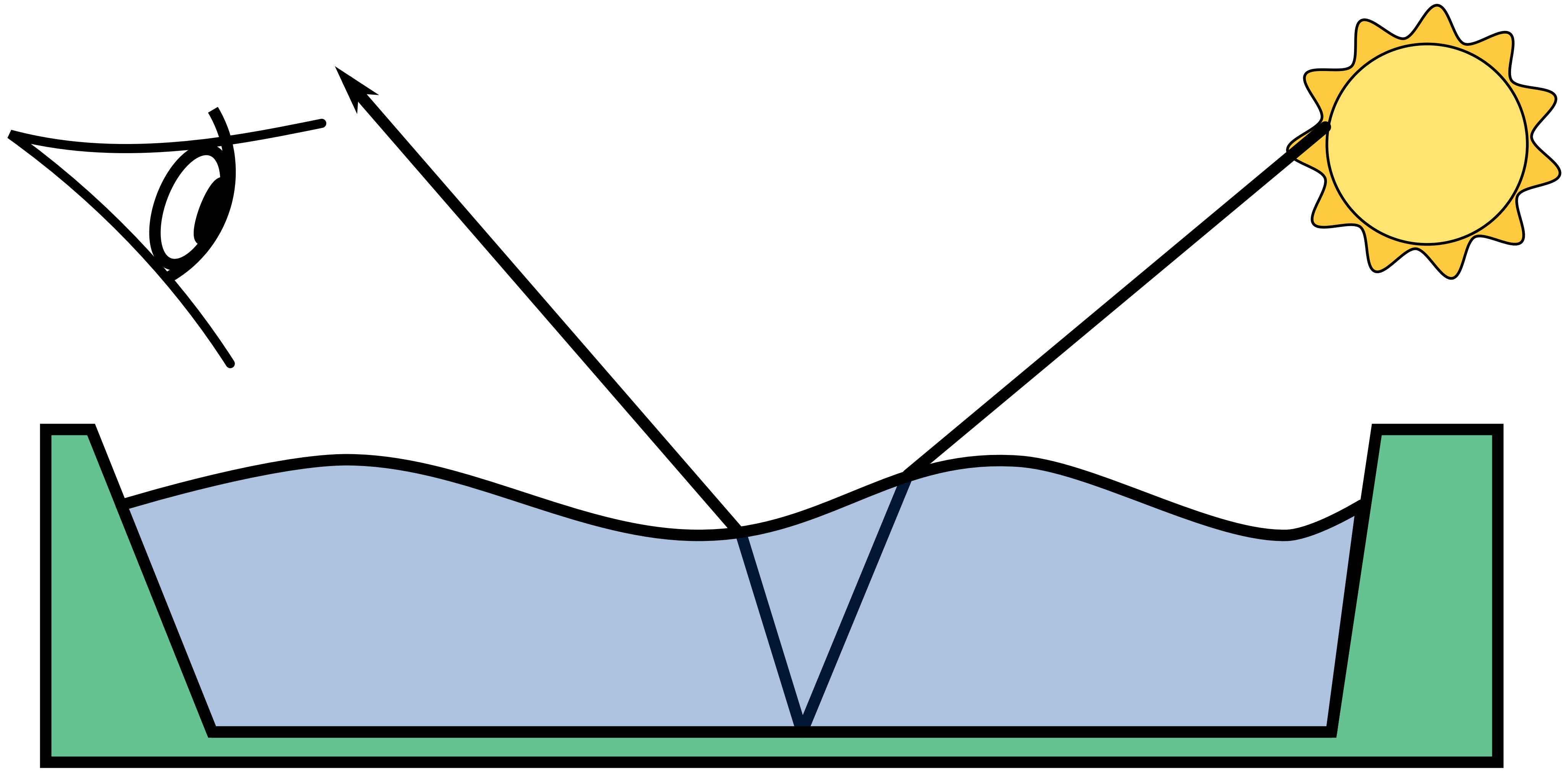
# Light tracing

- reverse the tracing direction, start at the light sources
- good for caustics



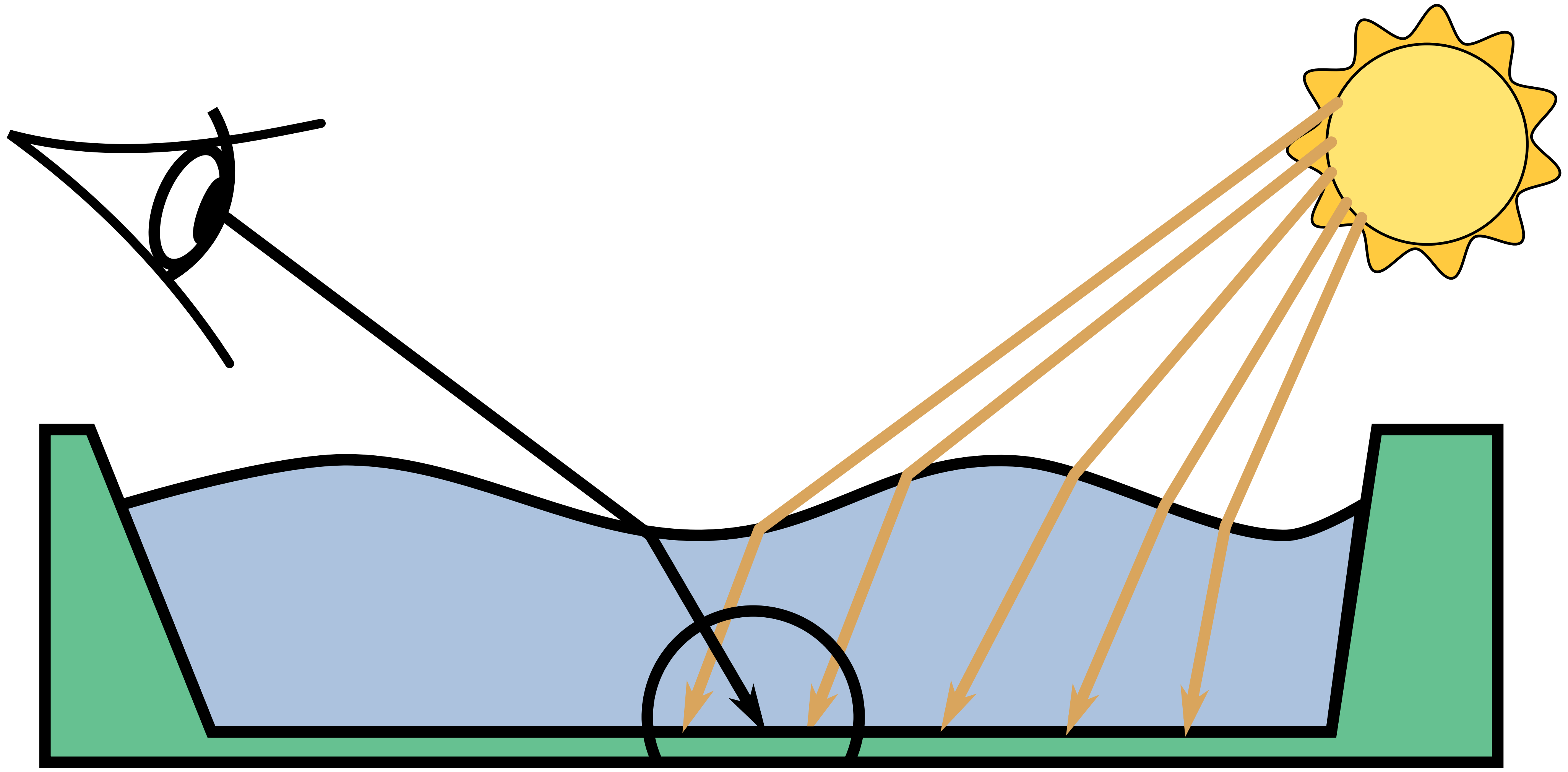
# Light tracing

- ..but doesn't work for specular, either
- SDS doesn't work even for combination of all the above techniques (called BDPT)



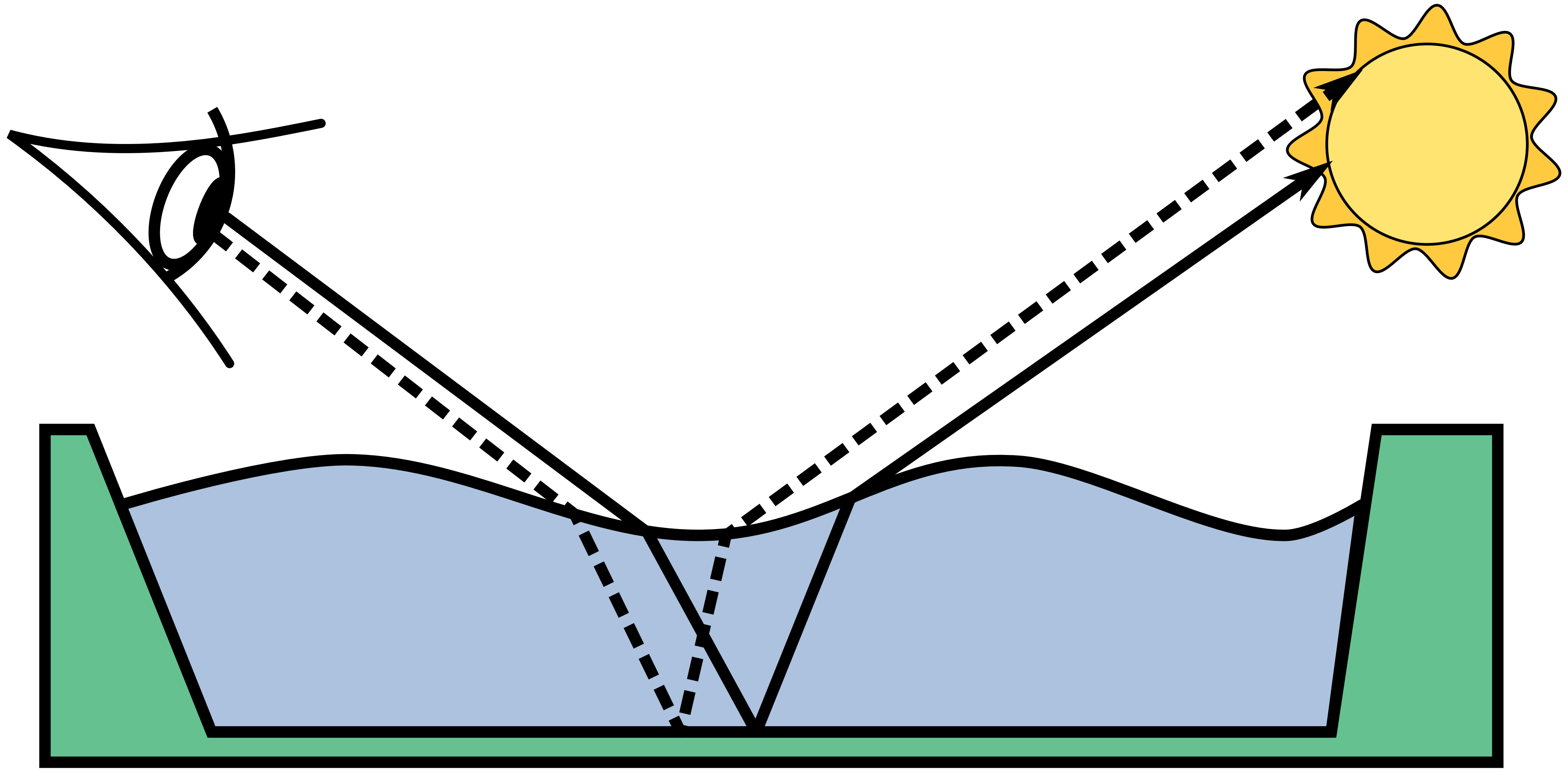
# Vertex connection and merging (VCM/UPS)

- uses photon maps to cover SDS paths
- expensive: big storage, long kd-tree build times, many combinations to evaluate



# Metropolis light transport (MLT)

- mutates initial sample, uses current path as Markov chain state
- often leads to temporal inconsistency/blotches



# Production images omitted for publication

- :(

# Program part 1:

- 09:00 — **Opening statement and introduction to path tracing (almost over now!)**
  (30 min, Johannes Hanika)
- 09:30 — **A short History of Monte Carlo**
  (30 min, Luca Fascione)
- 10:00 — **Implementing path sampling techniques**
  (30 min, Marc Droske)
- 10:30 — **Break** (15 min)

# Program part 1, cntd:

- 10:45 — **Finding good paths**
  (30 min, Jorge Schwarzhaupt)
- 11:15 — **Volumes**
  (30 min, Christopher Kulla)
- 11:45 — **The Ins of Production Rendering at Animal Logic**
  (30 min, Daniel Heckenberg)

# Path tracing in production, part 2

## will continue in the afternoon session

- here, 403AB, 2pm-5:15pm

# Let's get started!

# Thank you for listening

## questions?

- please find our course notes here: https://jo.dreggn.org/path-tracing-in-production/2019/