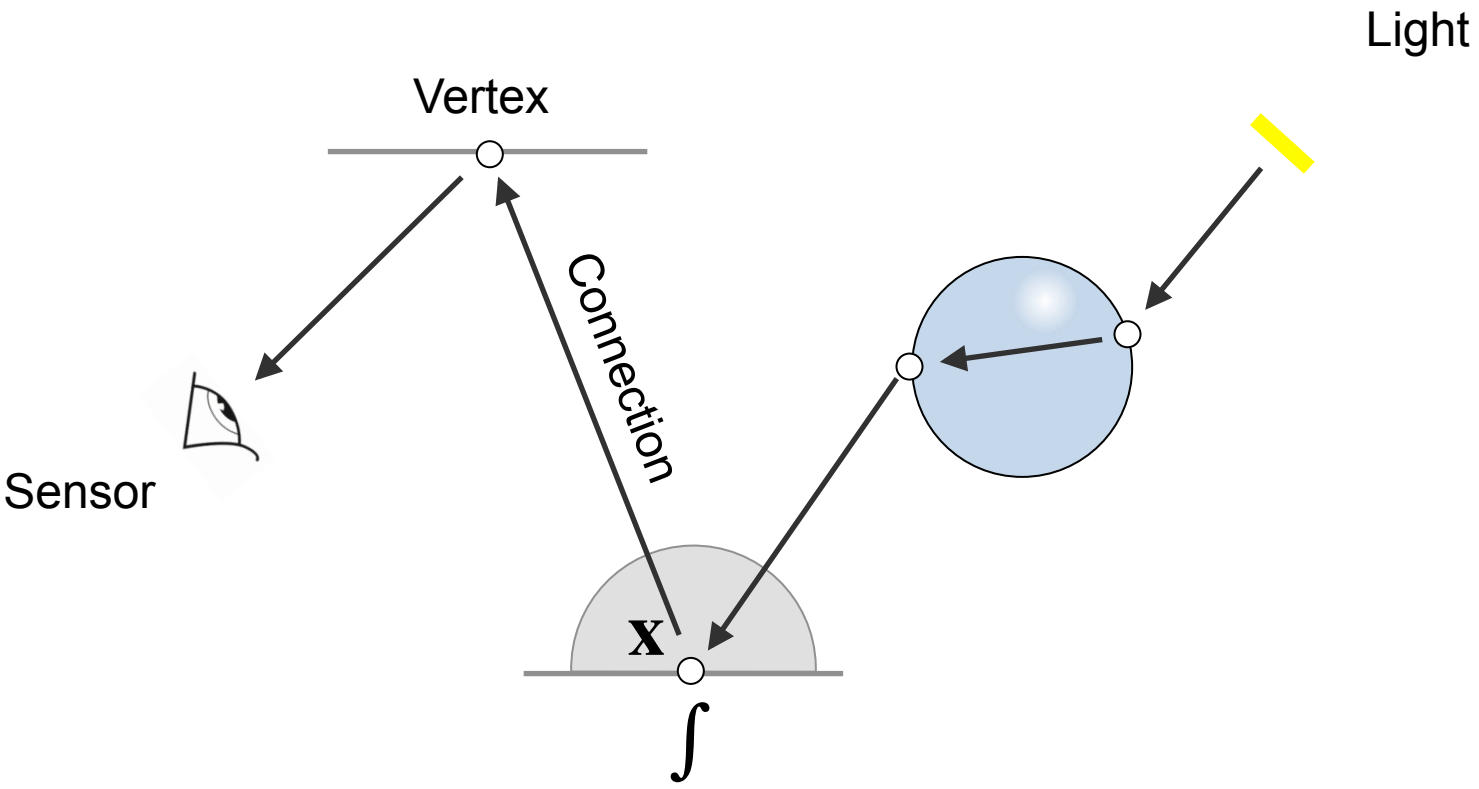
Photon Mapping

Today

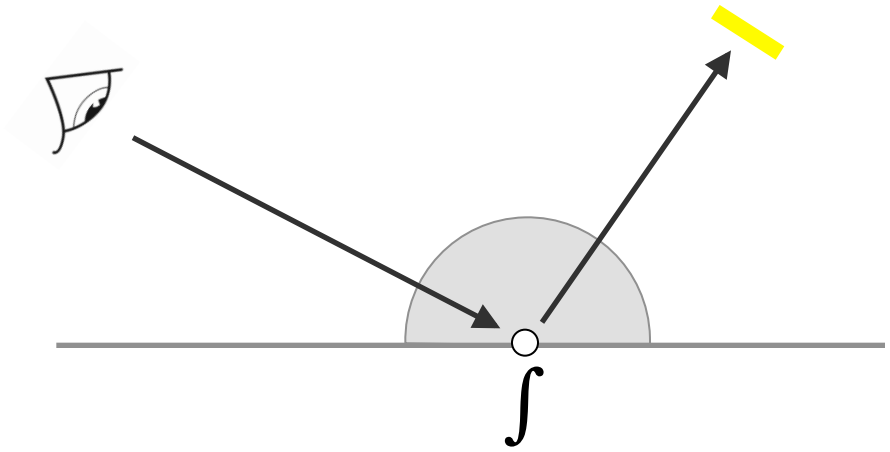
- Motivation
- Idea: Starting light paths at the light
- Methods
  - Light tracing
  - Bi-directional path tracing
  - Photon Mapping
  - Instant Radiosity

Path space jargon



Example 1: Small lights

- Small light sources imply a small hit chance
- Solution: Next-event estimation
- Pre-condition: We know what is a light



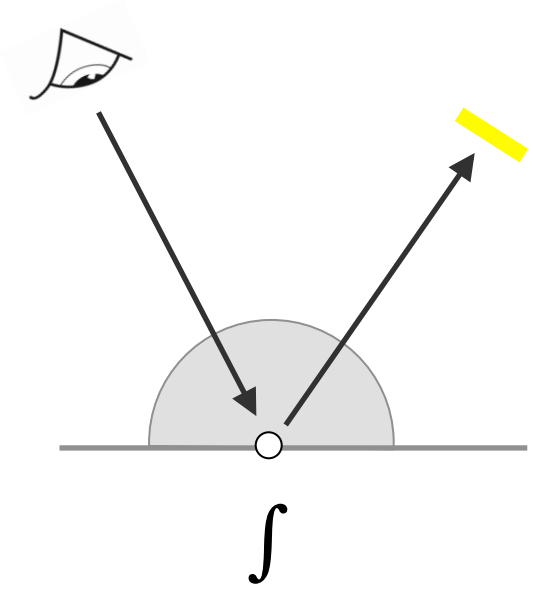
What is hard for Path tracing

- Two examples:
  - Occlusion (key hole-like)
  - Specular (caustics)

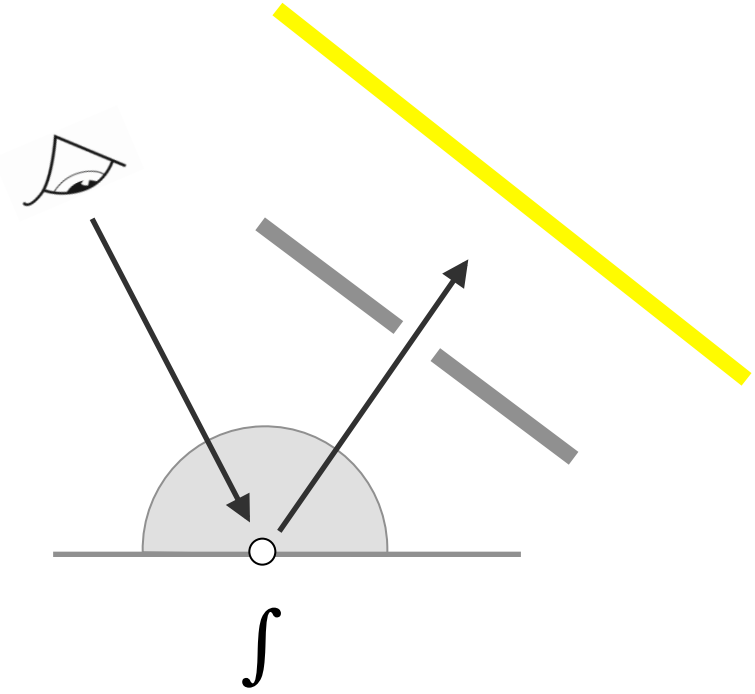
Small lights

Limits of next-event estimation

Small light: Fine!

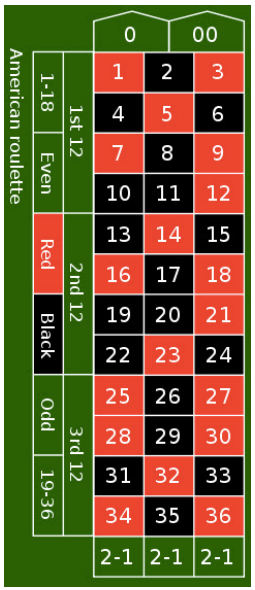


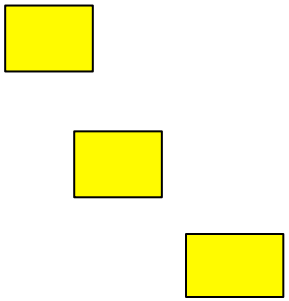
Large light, small holes, now what?

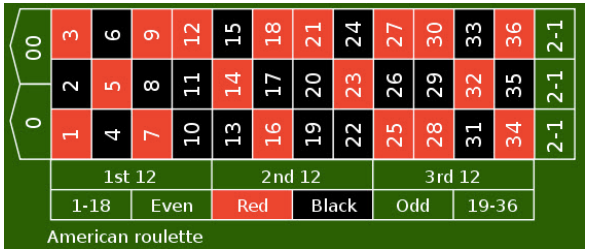


Roulette in spherical domain

θ

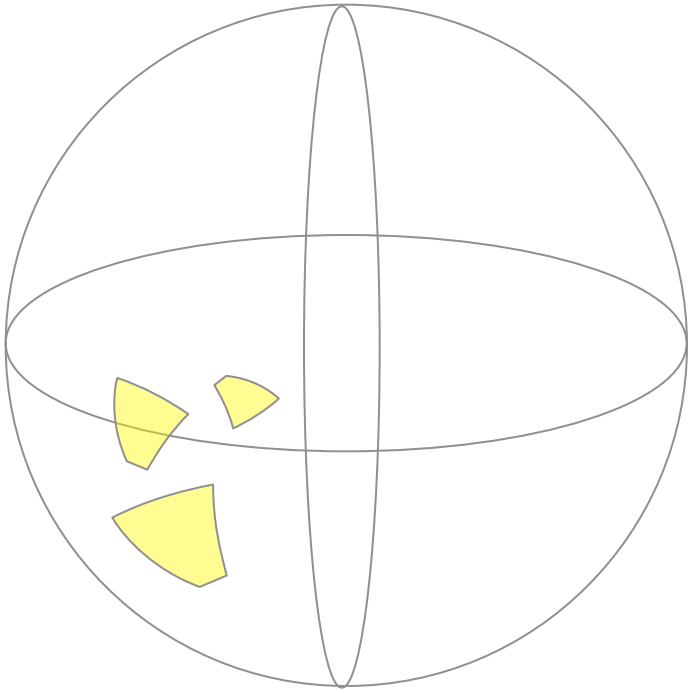




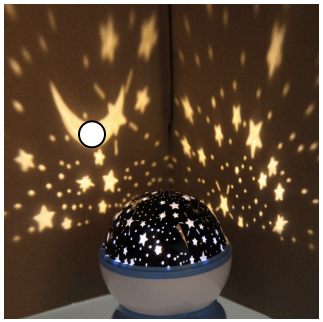


φ

θ



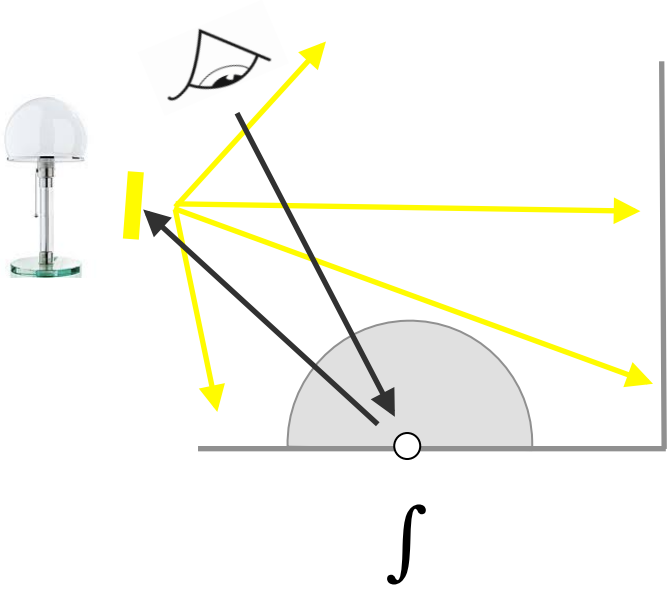
φ



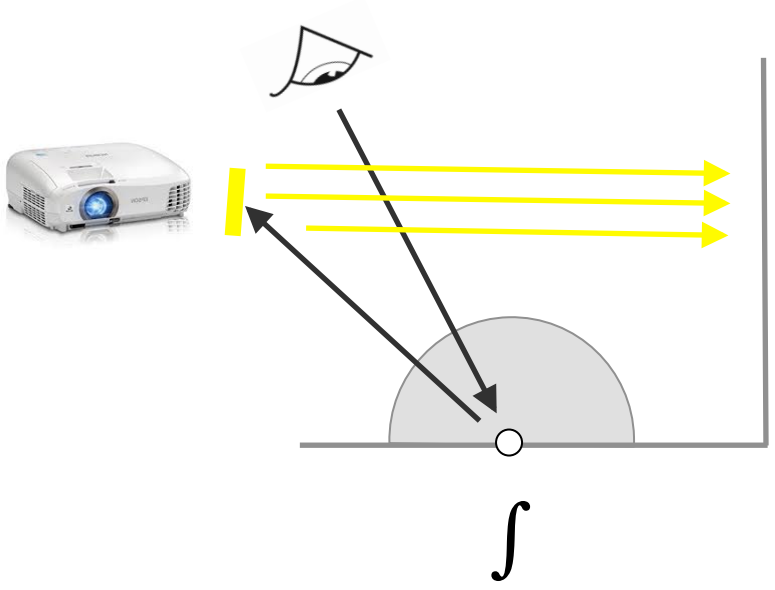

Projector

Next-event cant help

- The projector send in very few directions only
- Connecting to it is useless most if the time



$\int$

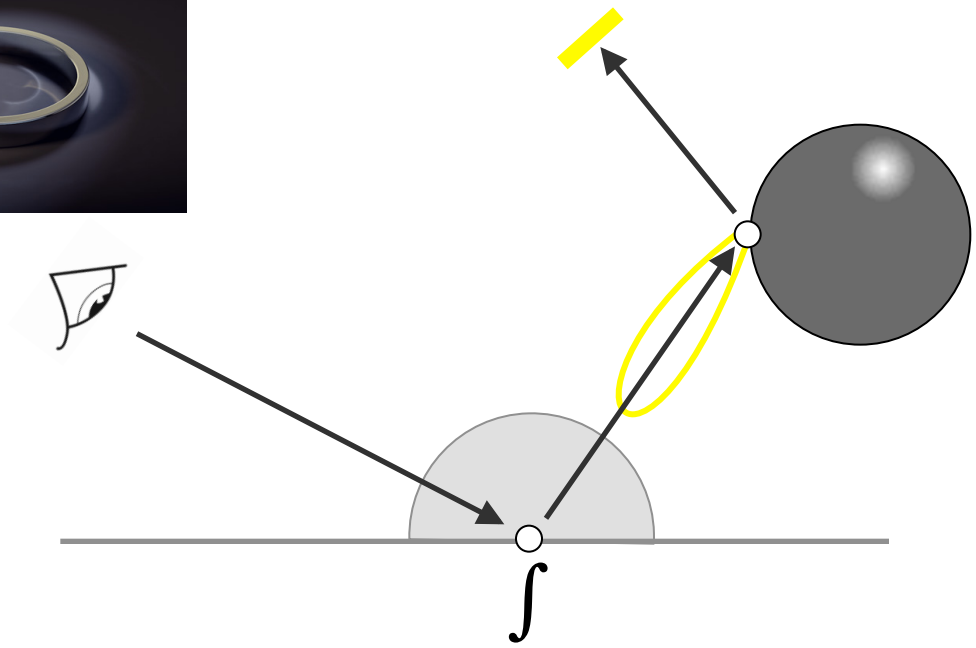


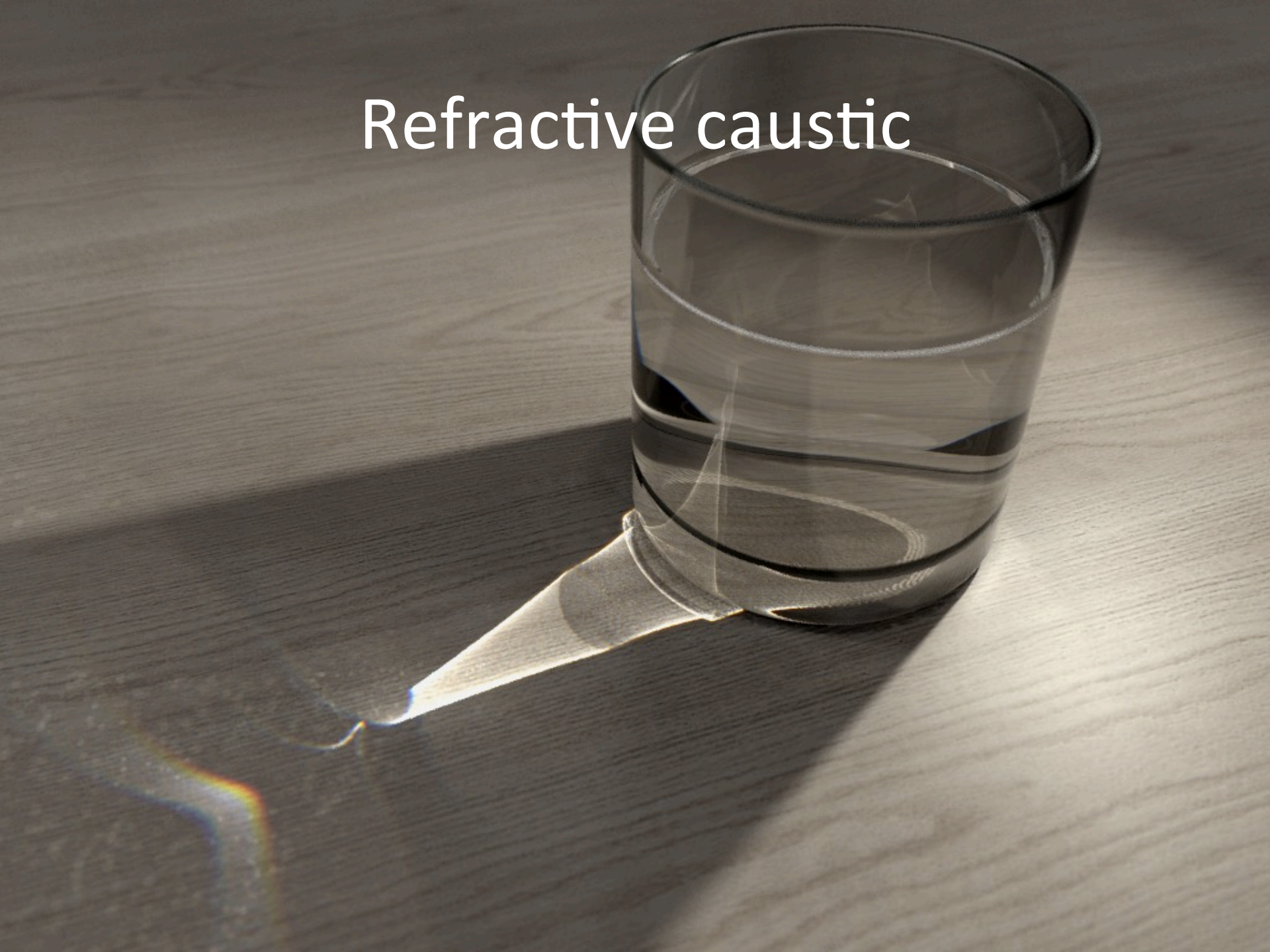
$\int$

Reflective caustic

Example 2: Caustics

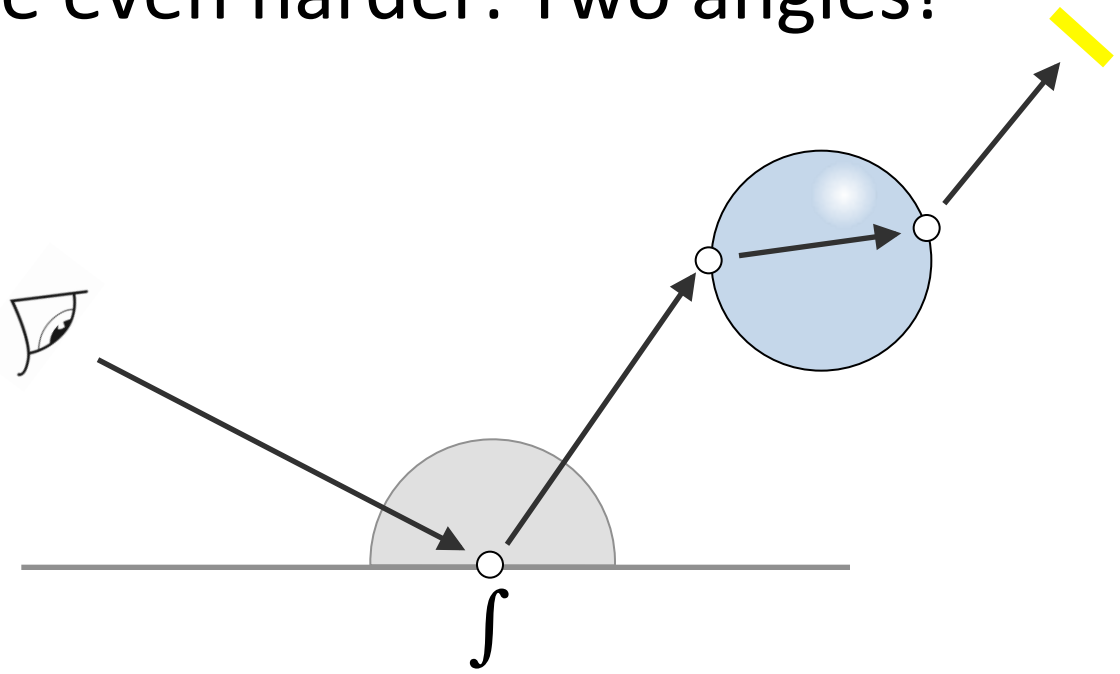
- A **caustic** is similar to a small light sources
- No obvious way to find it




Refractive caustic

Example 2: Caustics

- A **caustic** is similar to a small light sources
- No obvious solution
- Refractive even harder: Two angles!

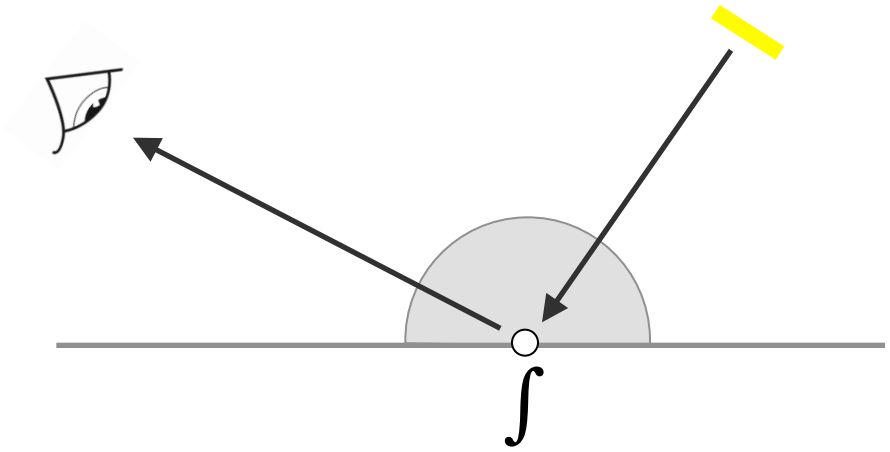


What is hard for path tracing?

- Paths of the form `LSDE`, `LSSDE`, etc
- Light that undergoes one or more specular reflections and then a diffuse bounce
- Reflective or refractive caustics

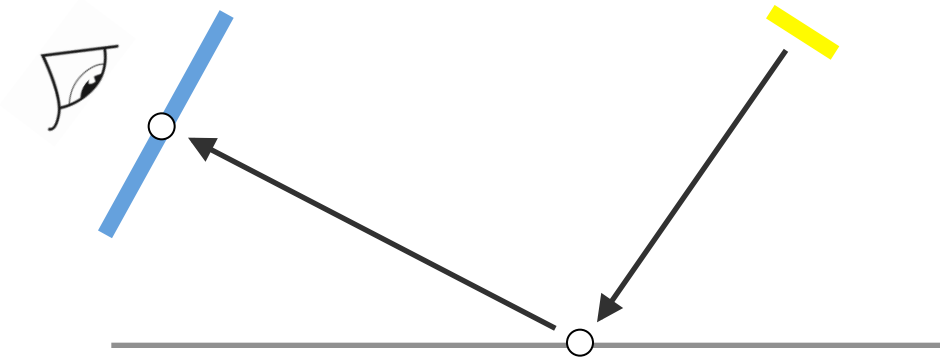
Solution: Light tracing method

- Start path at the light
- Trace rays through the scene
- In the end **project** onto sensor



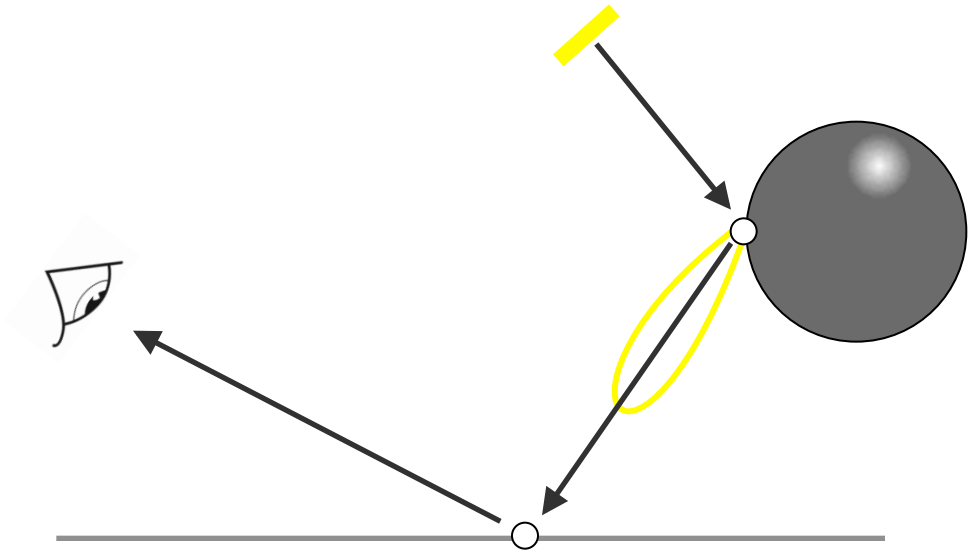
Light tracing

- Need next-event estimation at final vertex
- Sensor is even smaller than light
- Finding it by chance even less likely



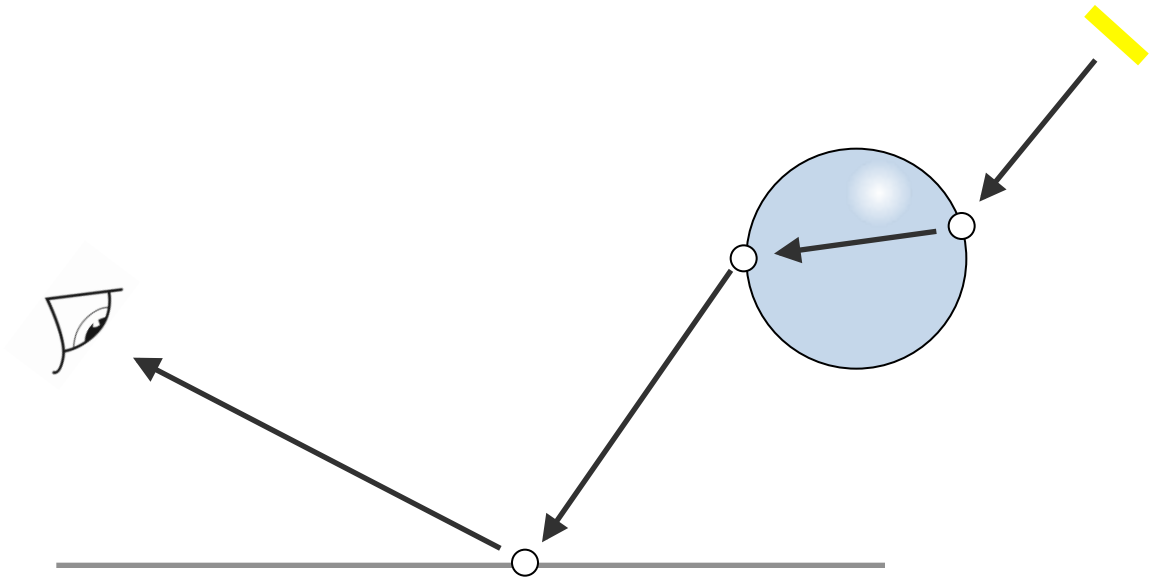
Light tracing for a reflective caustic

- Works quite well

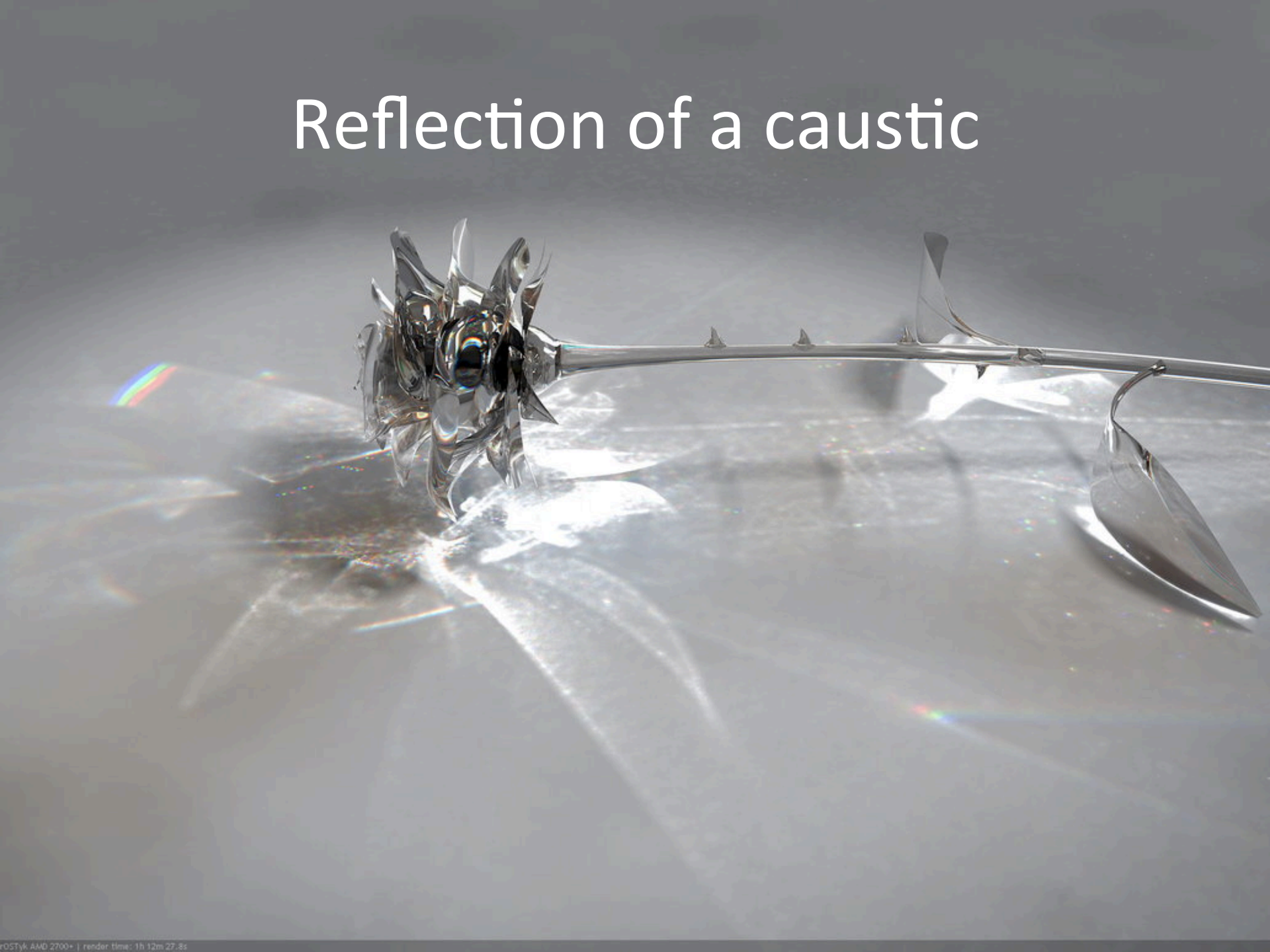


Light tracing for a refractive caustic

- Works quite well

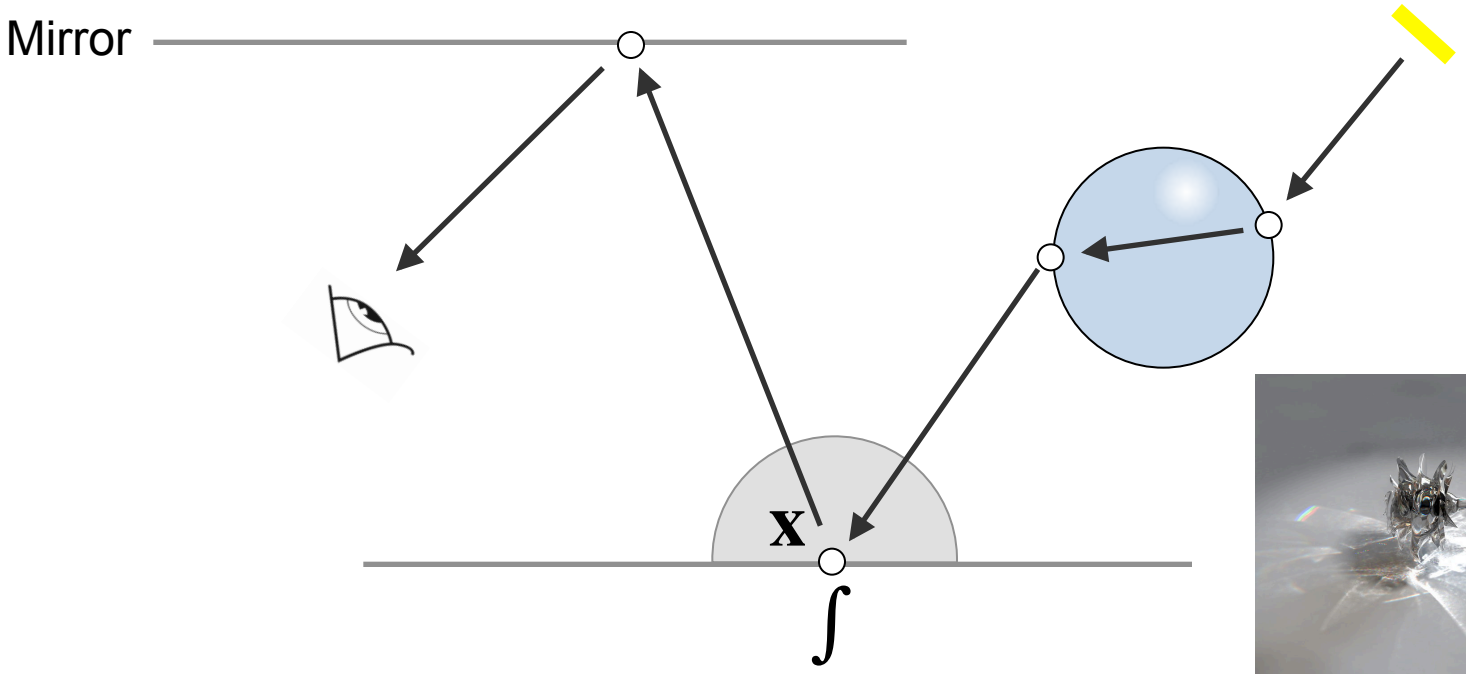


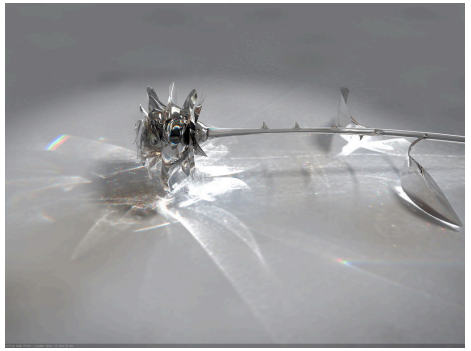
Reflection of a caustic



Caustics in a mirror

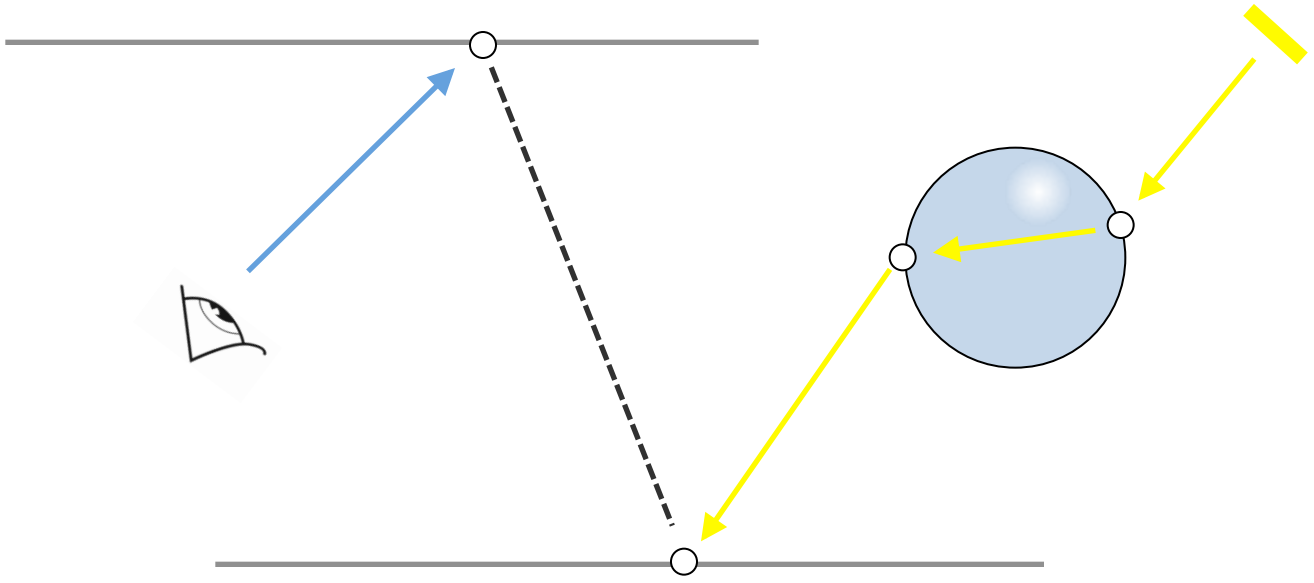
- Very hard!
- How could we know at **x** how to go on?





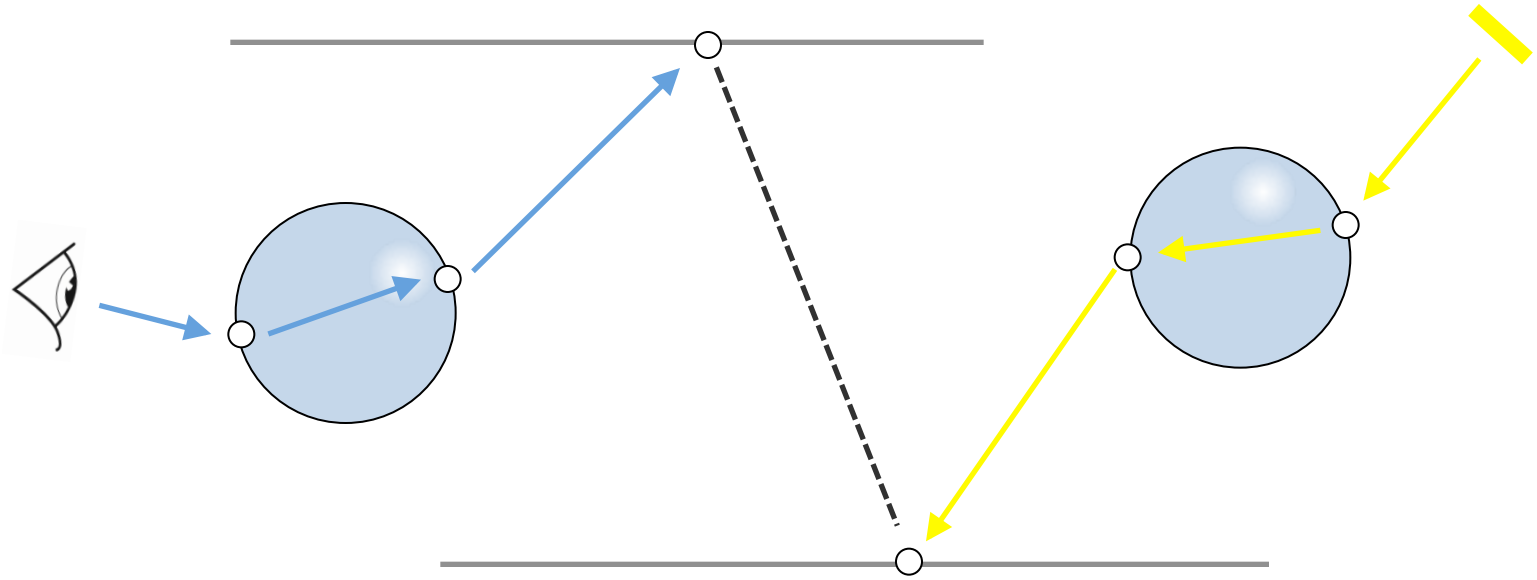
Bi-directional path tracing

- Start a path both at the eye and the light
- Connect the ends



Bi-directional path tracing

- Start a path both at the eye and the light
- Connect the ends

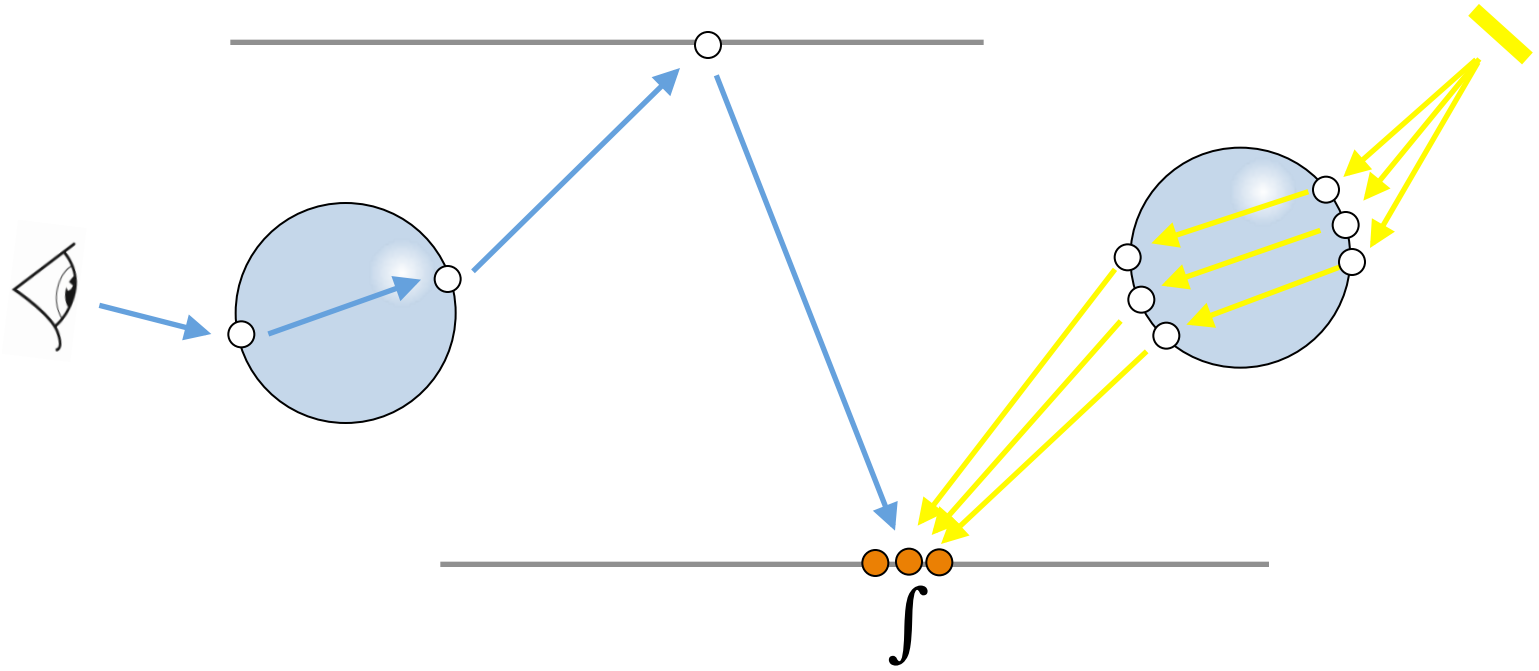


Photon mapping

- When eye paths connect to light paths we don't care about their path, only about the vertex
- Idea:
  - Store end-vertices from the light
  - Re-use from the eye

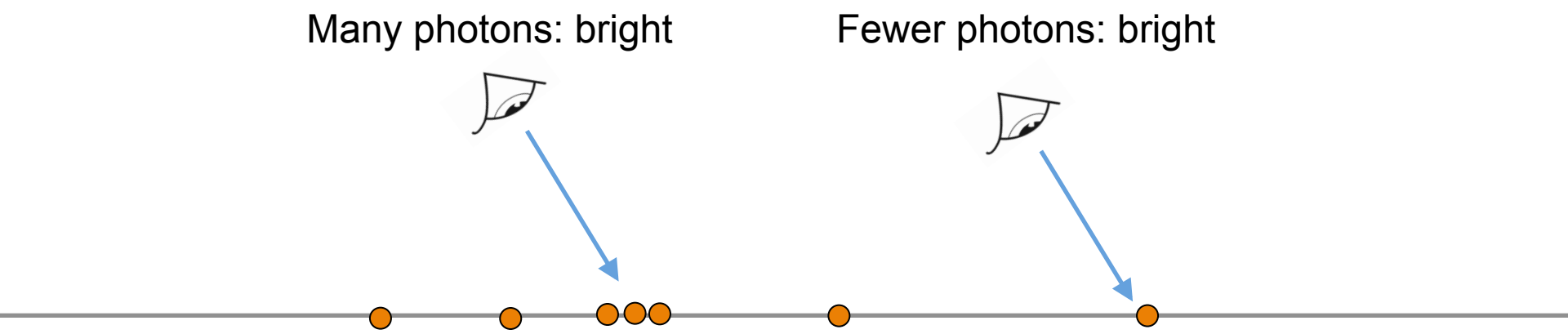
Photon mapping

- Start many rays at the light, store last vertex
- Re-use form the eye



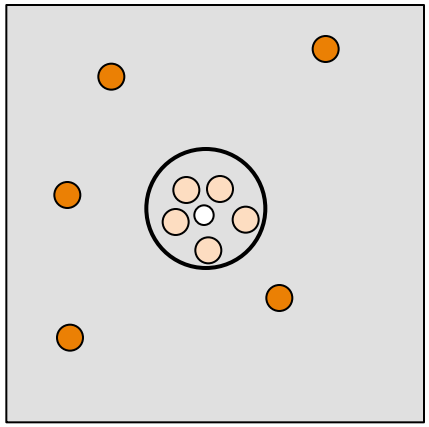
Density estimation

- Photon are just a list of 3D points
- How to convert into $L_i(\mathbf{x}, \omega)$?
- Find how many are nearby!



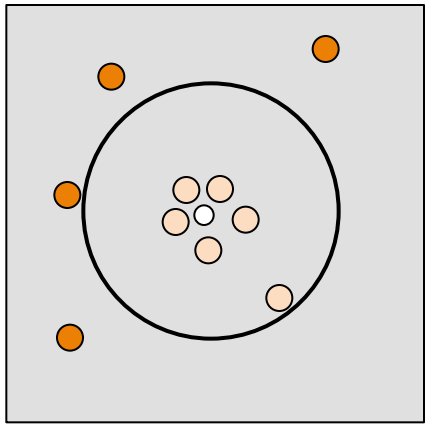
Two options: Option 1

- Find the nearest k
- See how large their radius r is
- Large a is small density and low L



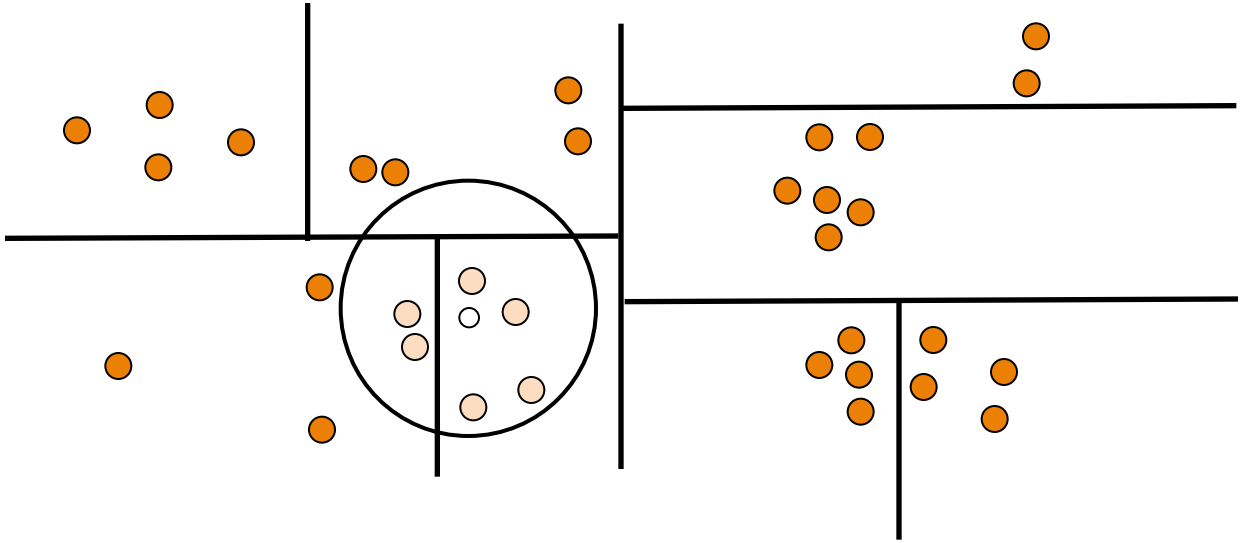
Two options: Option 2

- Fix a radius r
- Count how many k are in this radius
- Large k is high density and high L



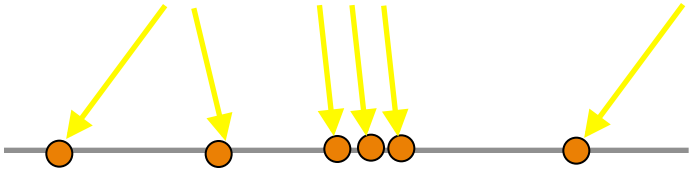
How to find k nearest?

- Spatial bounding structures
- k d tree
- Can find k NN in $\log(n)+k$ time for n points



BRDF in PM

- The light also depends on the direction
- Photons also store from where they came
- Like this, can take direction into account



Instant radiosity

- Two passes
  - Instead of doing density estimation, just consider each photon a light sources
  - Proceed with your usual rendering pipeline with shadow maps to light from these
- Can even be OpenGL and shadow mapping
- Can work in real-time
- Closest to real-time GI we got for now

Real-time 5 ys ago

Recap

- Can also start from the light
- Sometimes better
- Bias / Variance / Consistence
- Three ways to to this
  - Light tracing
  - Path Tracing
  - Instant radiosity
    - Bi-Directional Ray Tracing
      - Photon Mapping