Global Illumination and Path Tracing

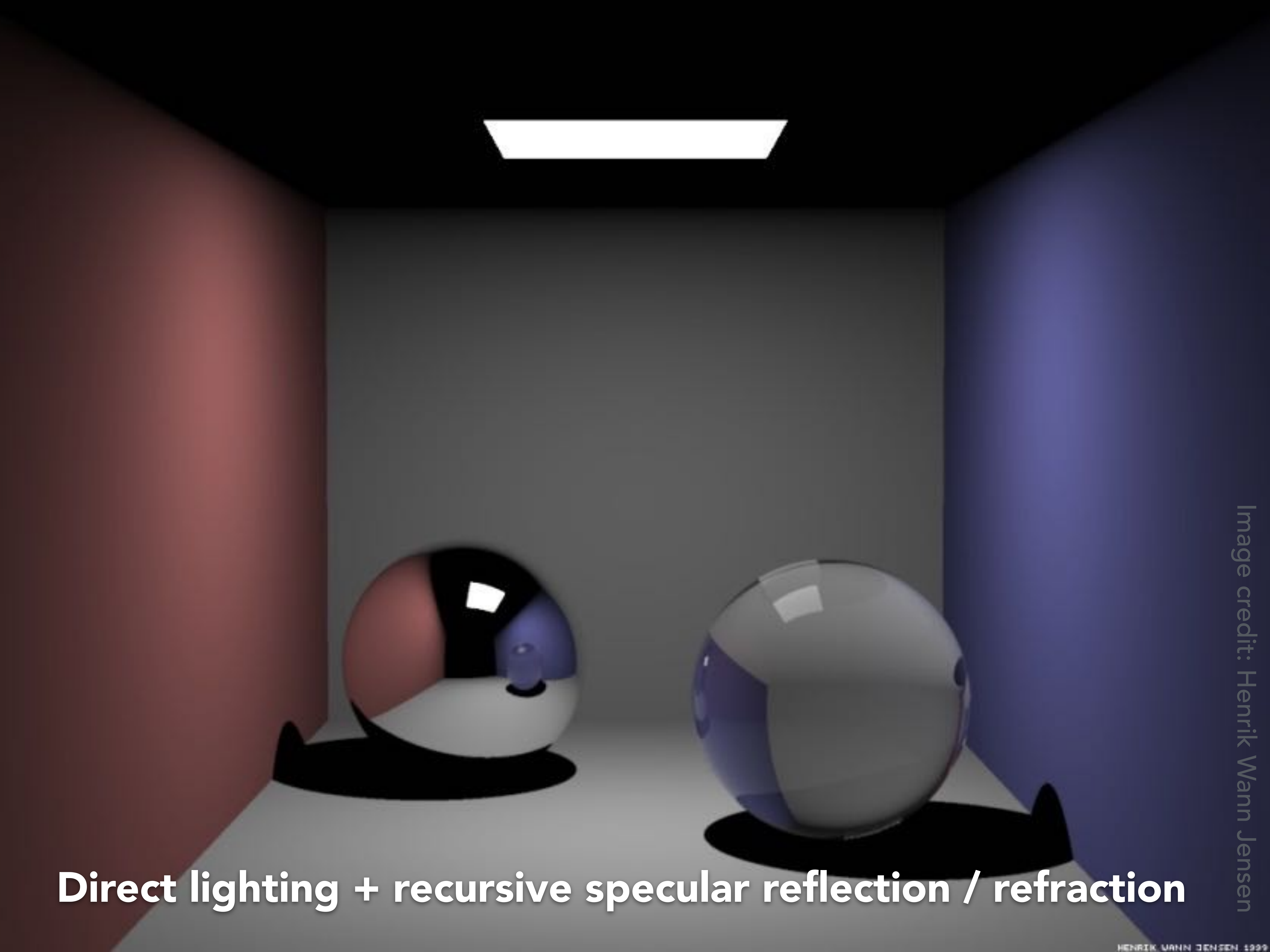
Computer Graphics and Imaging

UC Berkeley CS184

Summer 2020

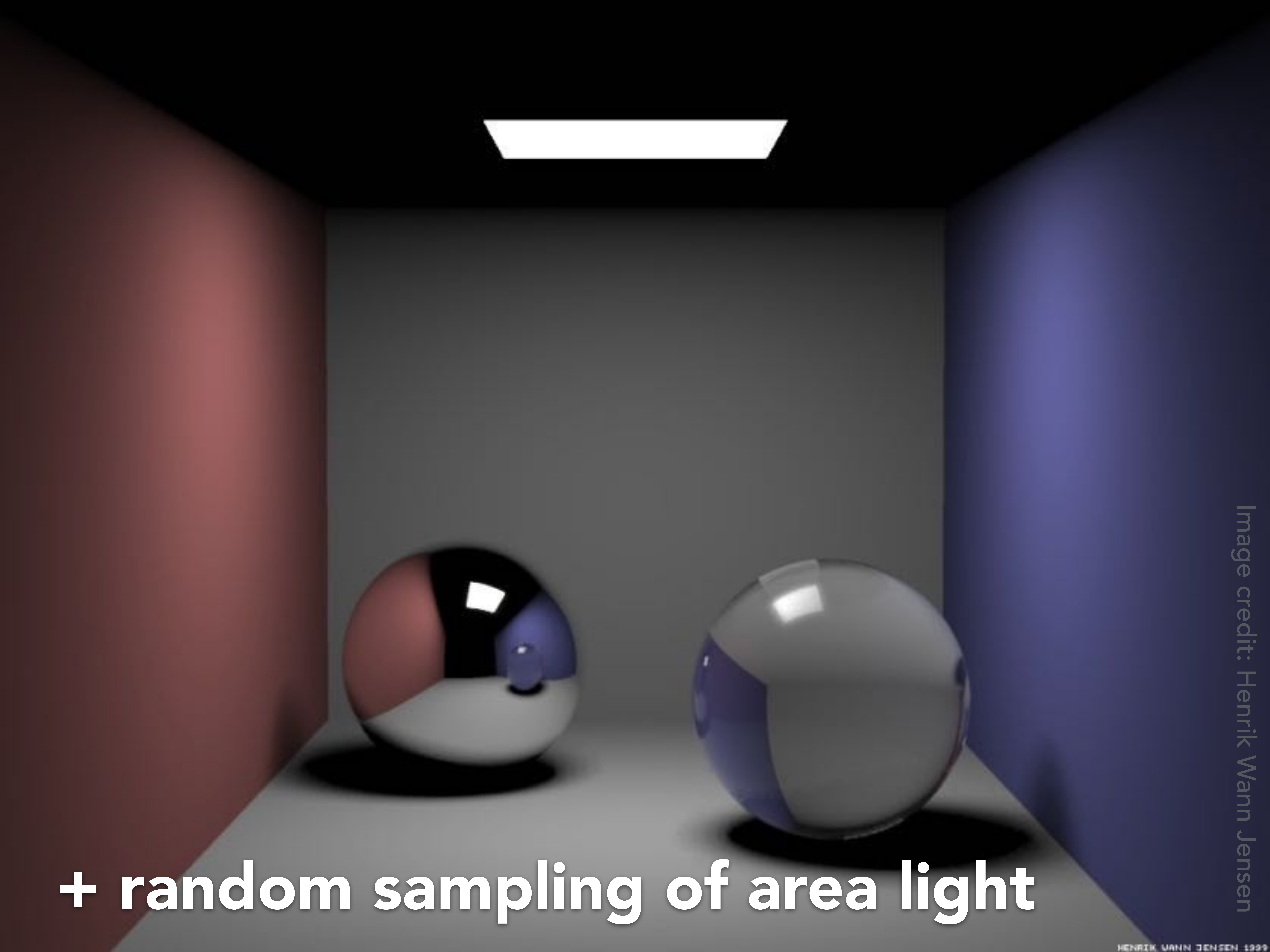
Announcements

- Assignment 3-1 is out!
  - Please, please, please, please, please, *please* start as early as possible!
- Exam is in two weeks
  - (More details at the end of this week)
  - Basic logistics: available for 24 hours, ~2-3 hours of work, open resources but *no* collaboration
- Assignment 1 grades are out
  - Regrade requests open until the end of this week
  - Grade on Gradescope doesn't reflect any deductions from late days



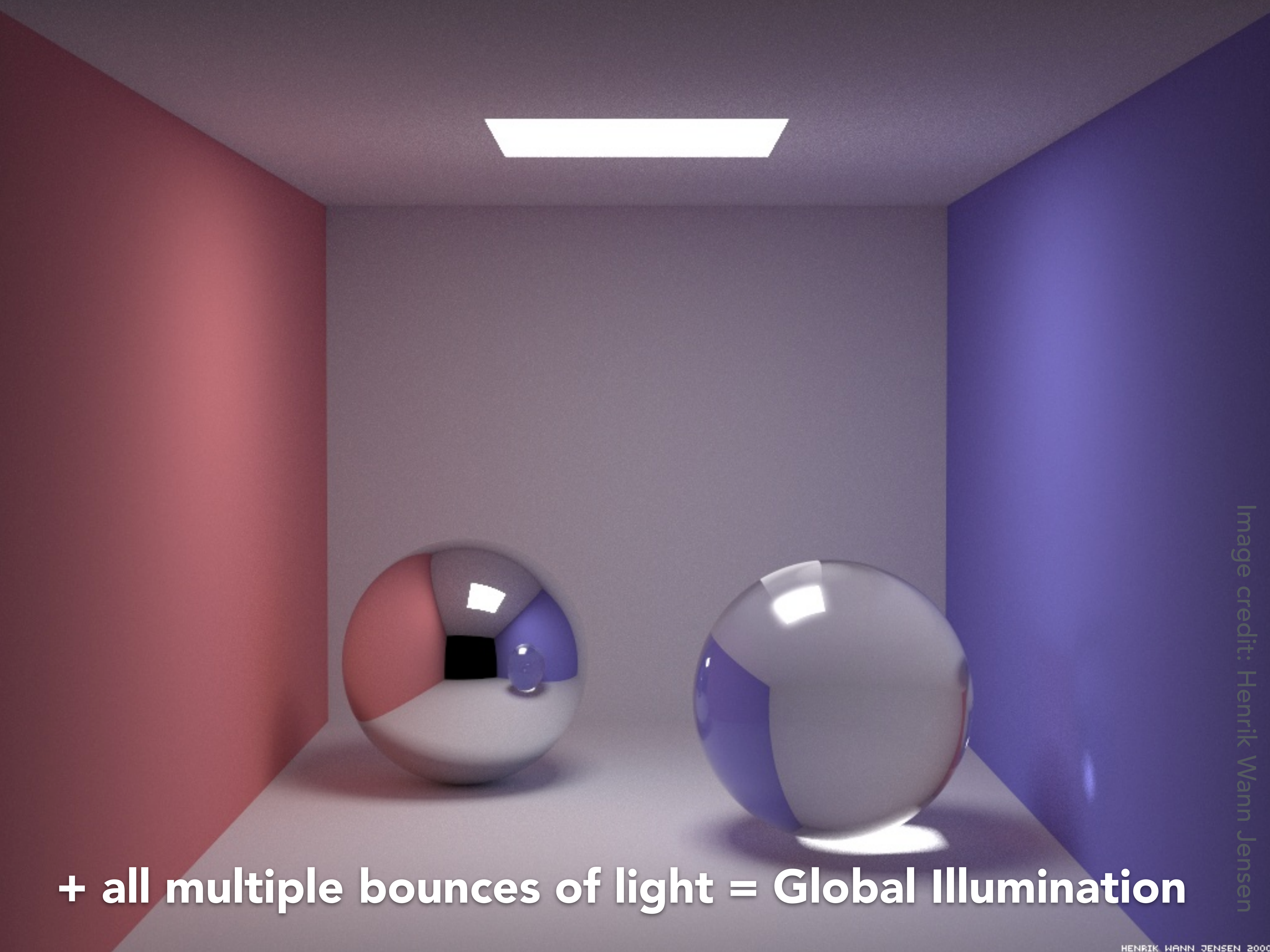
Direct lighting + recursive specular reflection / refraction

Image credit: Henrik Wann Jensen



+ random sampling of area light

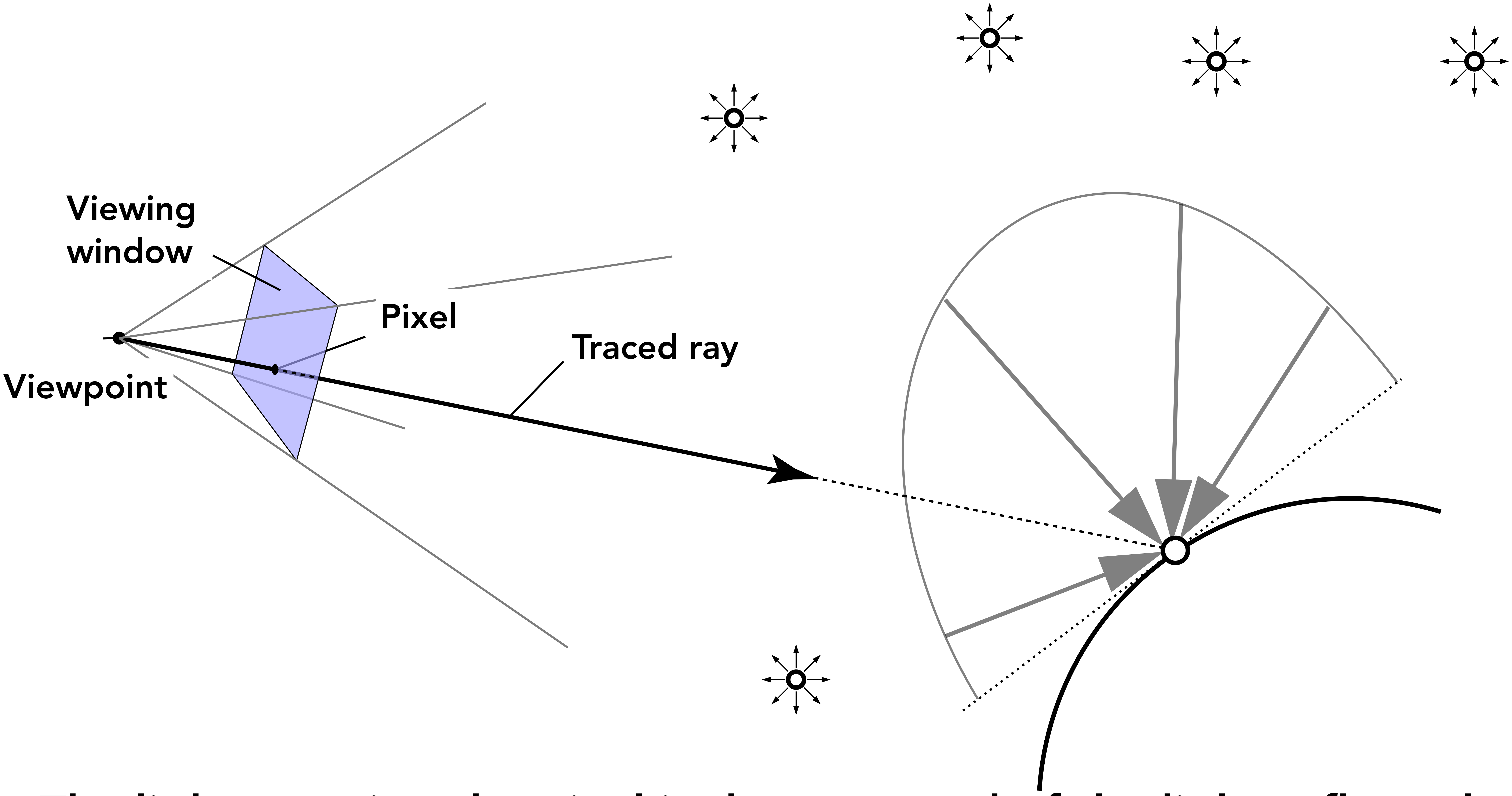
Image credit: Henrik Wann Jensen



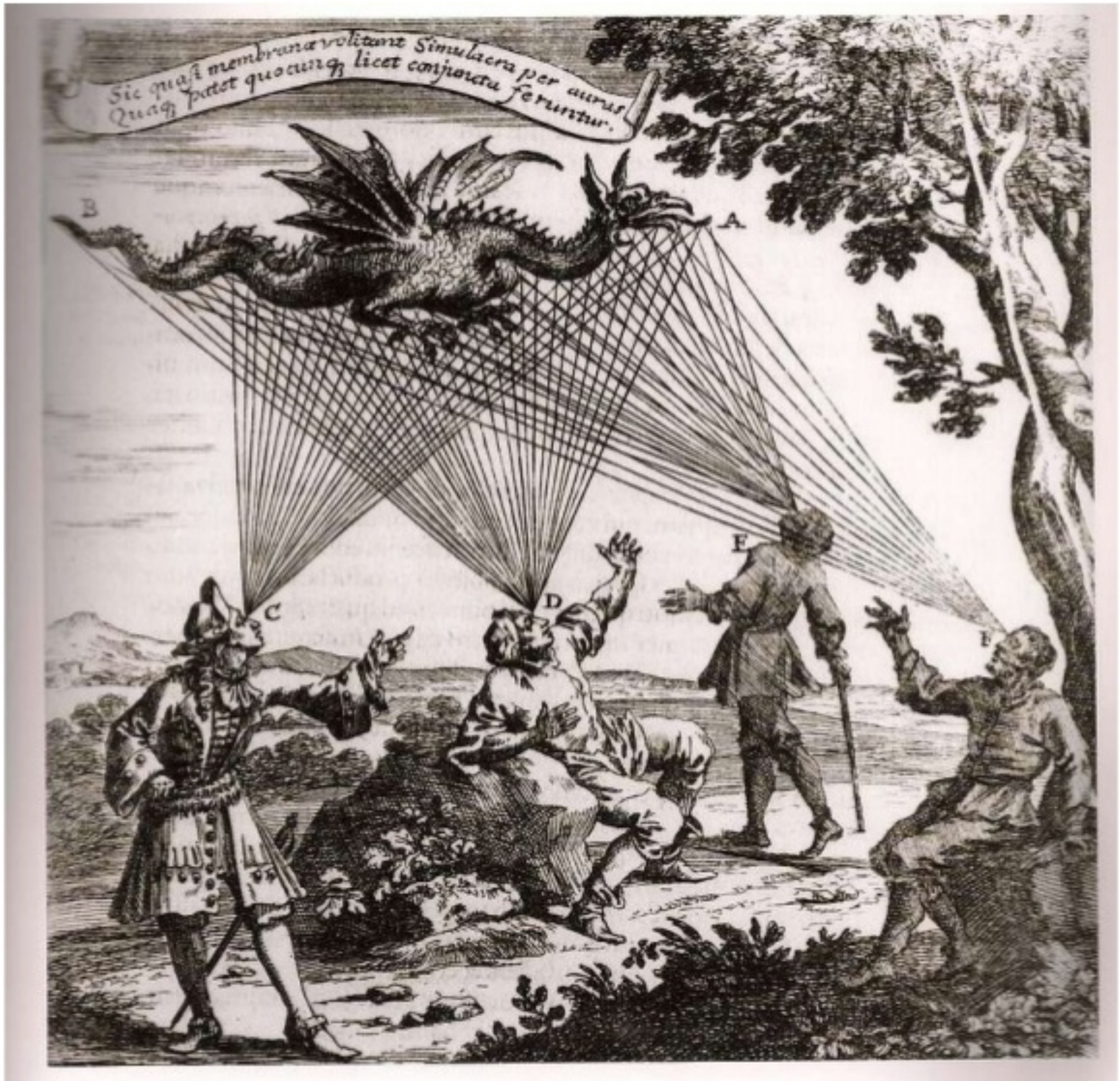
+ all multiple bounces of light = Global Illumination

Image credit: Henrik Wann Jensen

Ray Tracer Samples Radiance Along A Ray



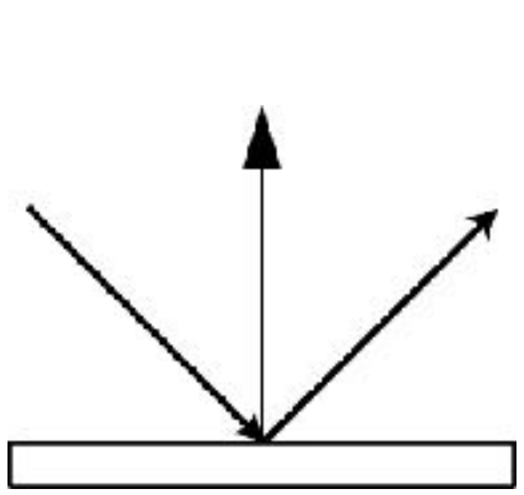
The light entering the pixel is the sum total of the light reflected off the surface into the ray’s (reverse) direction


Sic quaſi membranæ volitant Simulacra per auras
Quaq; patet quocunq; licet conjuncta feruntur.
B
A
C
D
E
F

Categories of Reflection Functions

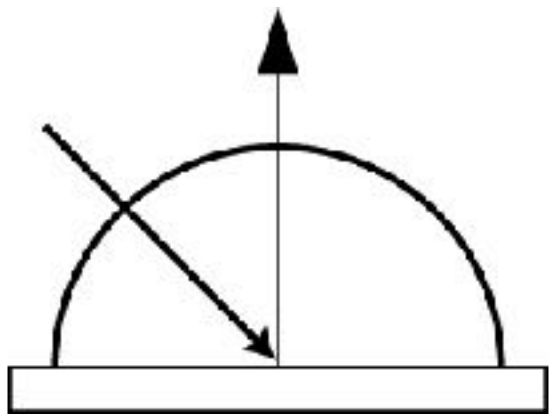
Ideal specular

- Perfect mirror reflection



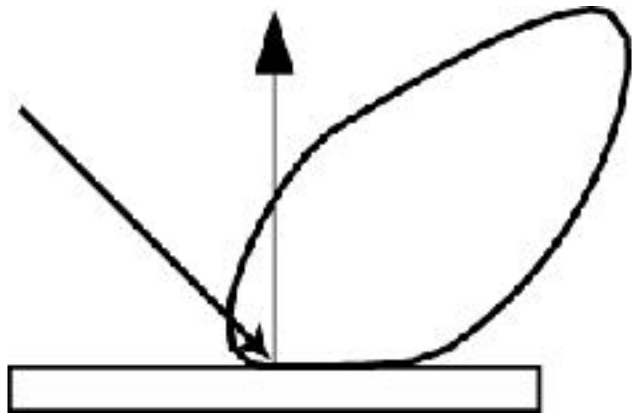
Ideal diffuse

- Equal reflection in all directions



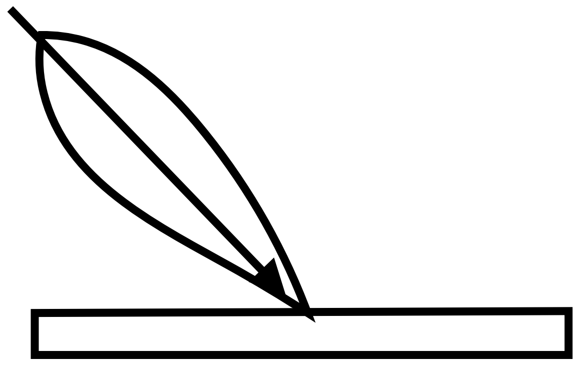
Glossy specular

- Majority of light reflected near mirror direction



Retro-reflective

- Light reflected back towards light source



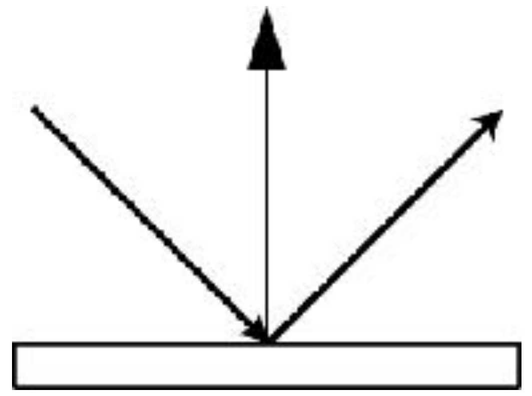
Diagrams illustrate how light from incoming direction is reflected in various outgoing directions.

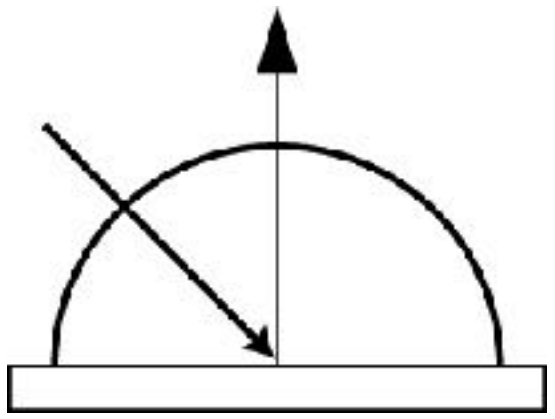
Reflection Functions in Pathtracing

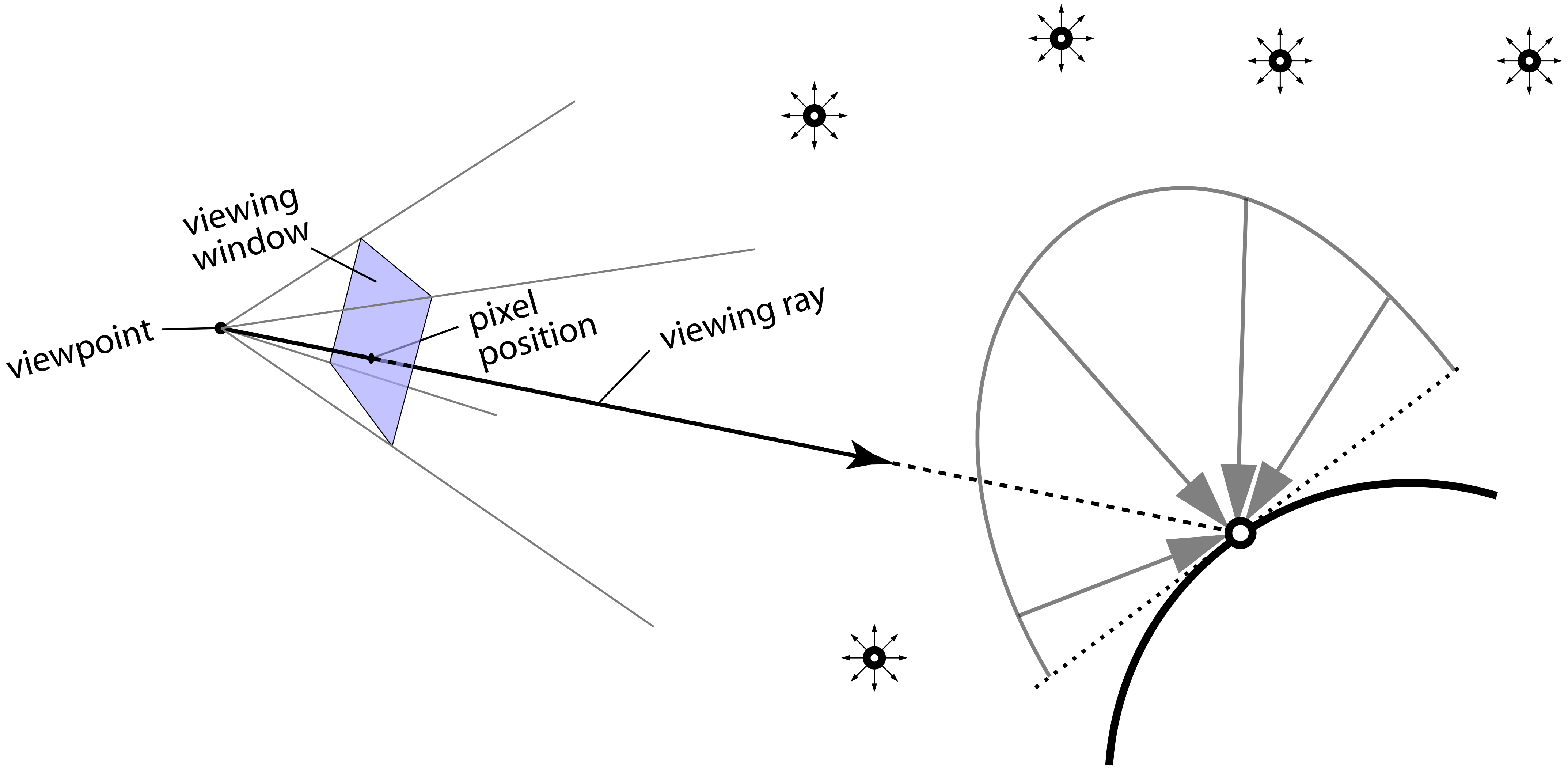
Light comes in from a particular direction, where is it likely to be reflected?

=

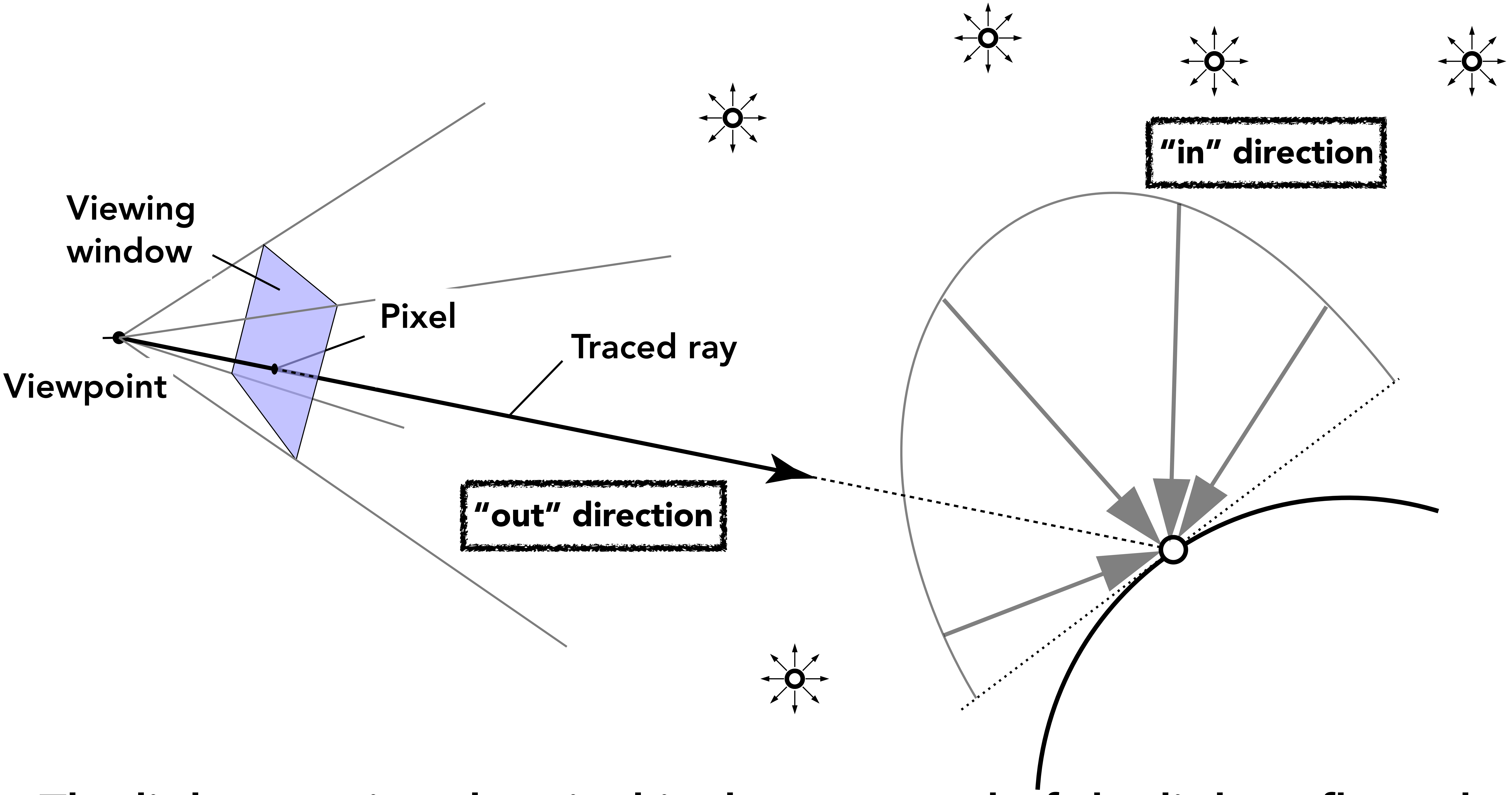
Light exits from a particular direction, where was it likely to come from?





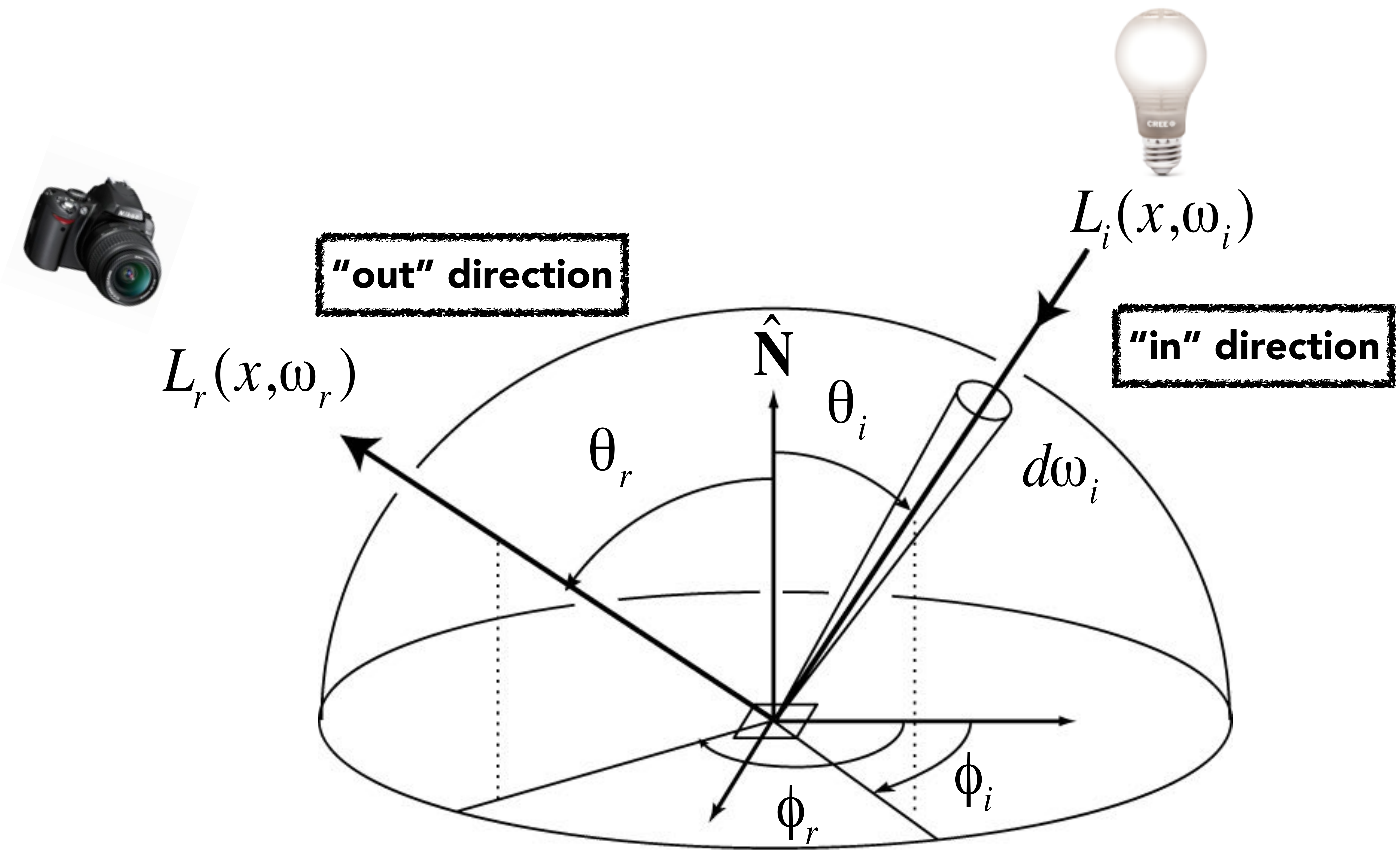


Ray Tracer Samples Radiance Along A Ray



The light entering the pixel is the sum total of the light reflected off the surface into the ray's (reverse) direction

The Reflection Equation



"out" direction

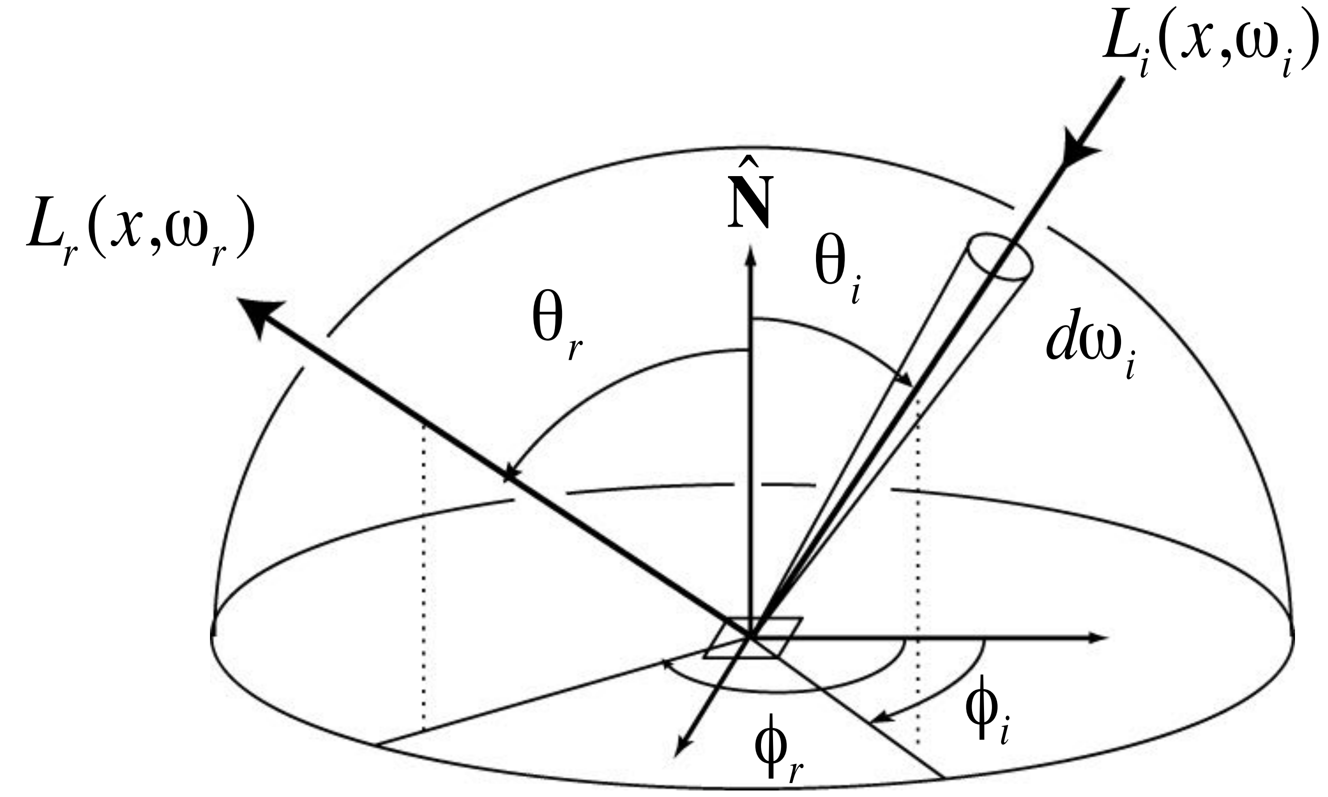
"in" direction

$$L_r(\mathrm{p},\omega_r) = \int_{H^2} f_r(\mathrm{p},\omega_i \rightarrow \omega_r)\, L_i(\mathrm{p},\omega_i)\, \cos\theta_i \,\mathrm{d}\omega_i$$

Challenge: This is Actually A Recursive Equation

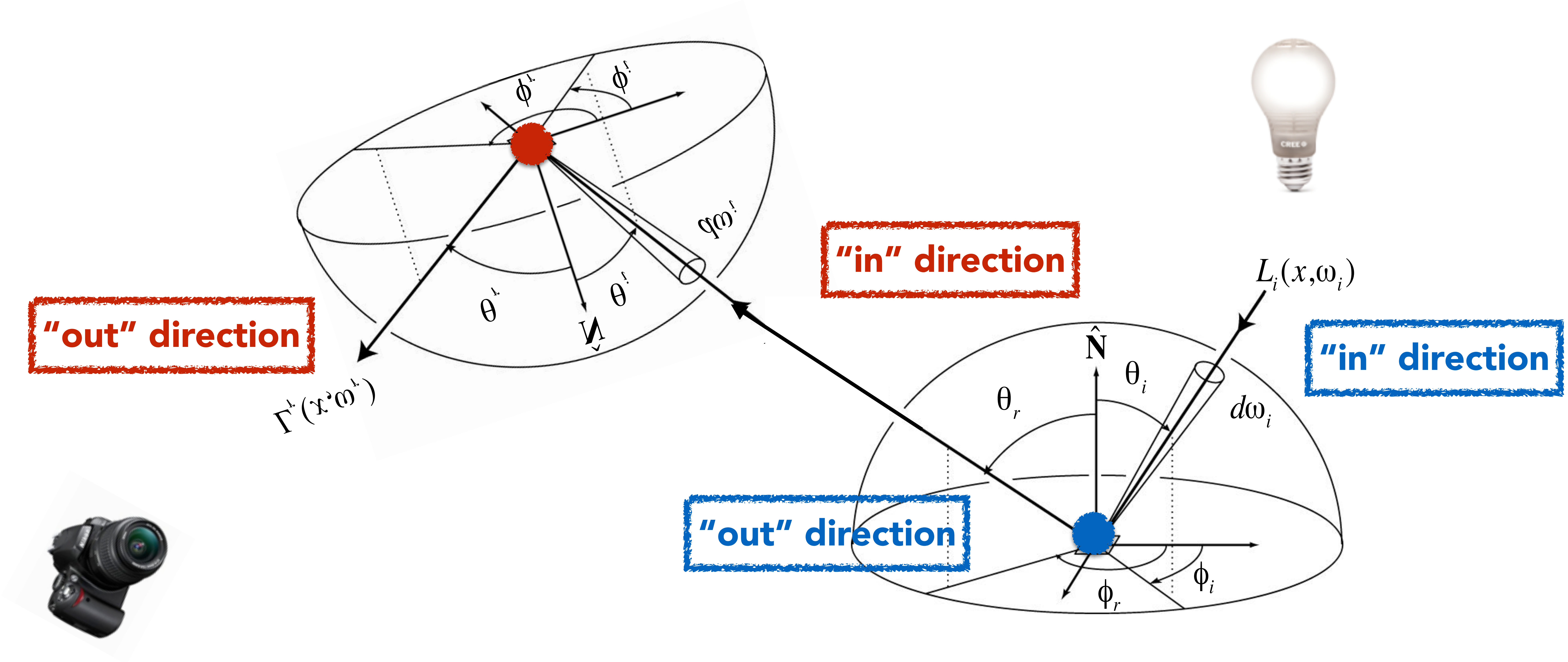
Reflected radiance depends on incoming radiance

$$\boxed{L_r(\mathrm{p}, \omega_r)} = \int_{H^2} f_r(\mathrm{p}, \omega_i \to \omega_r) \boxed{L_i(\mathrm{p}, \omega_i)} \cos\theta_i \, \mathrm{d}\omega_i$$



But incoming radiance depends on reflected radiance (at another point in the scene)

Recursive Light Bounces

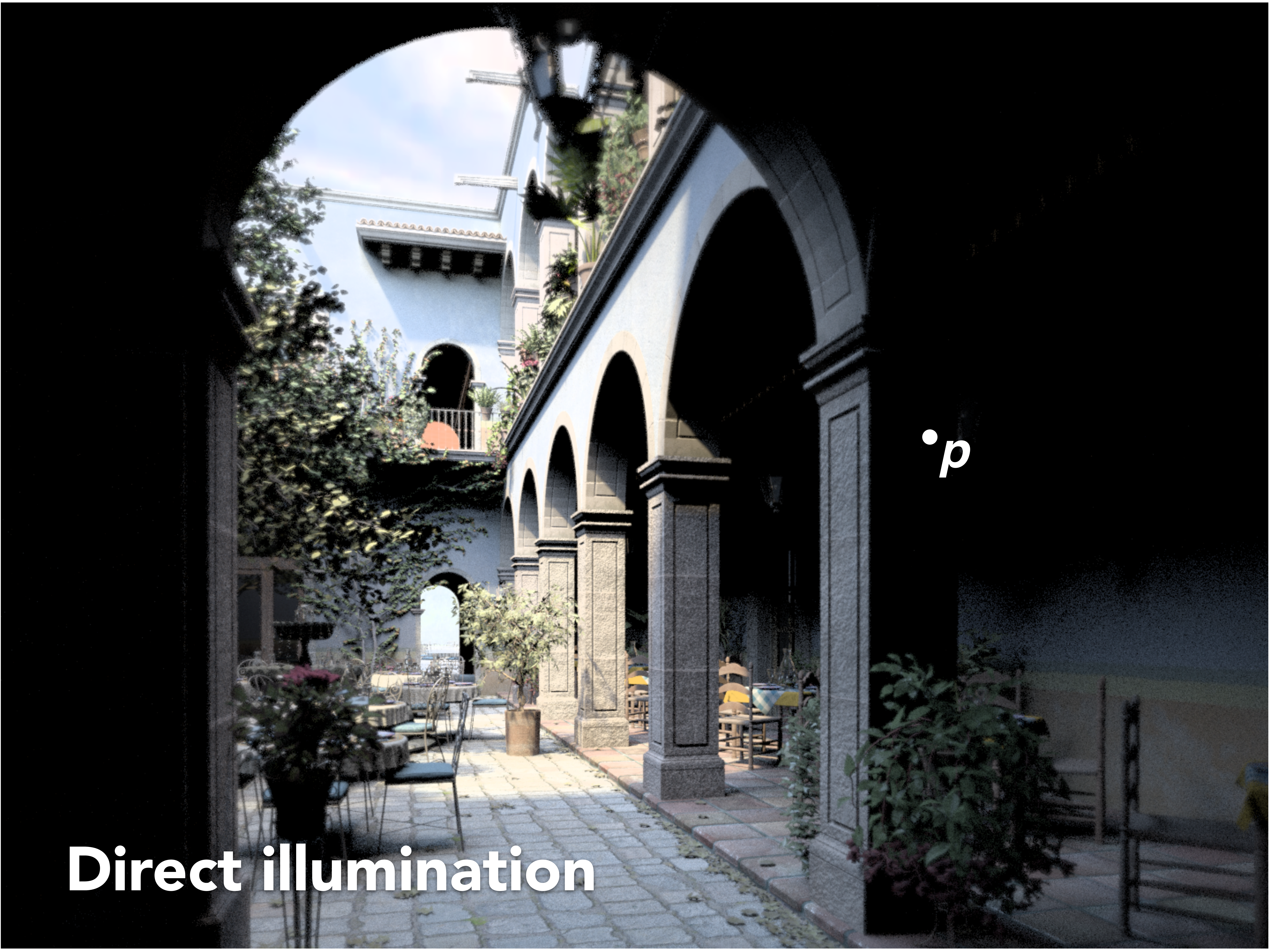


"out" direction

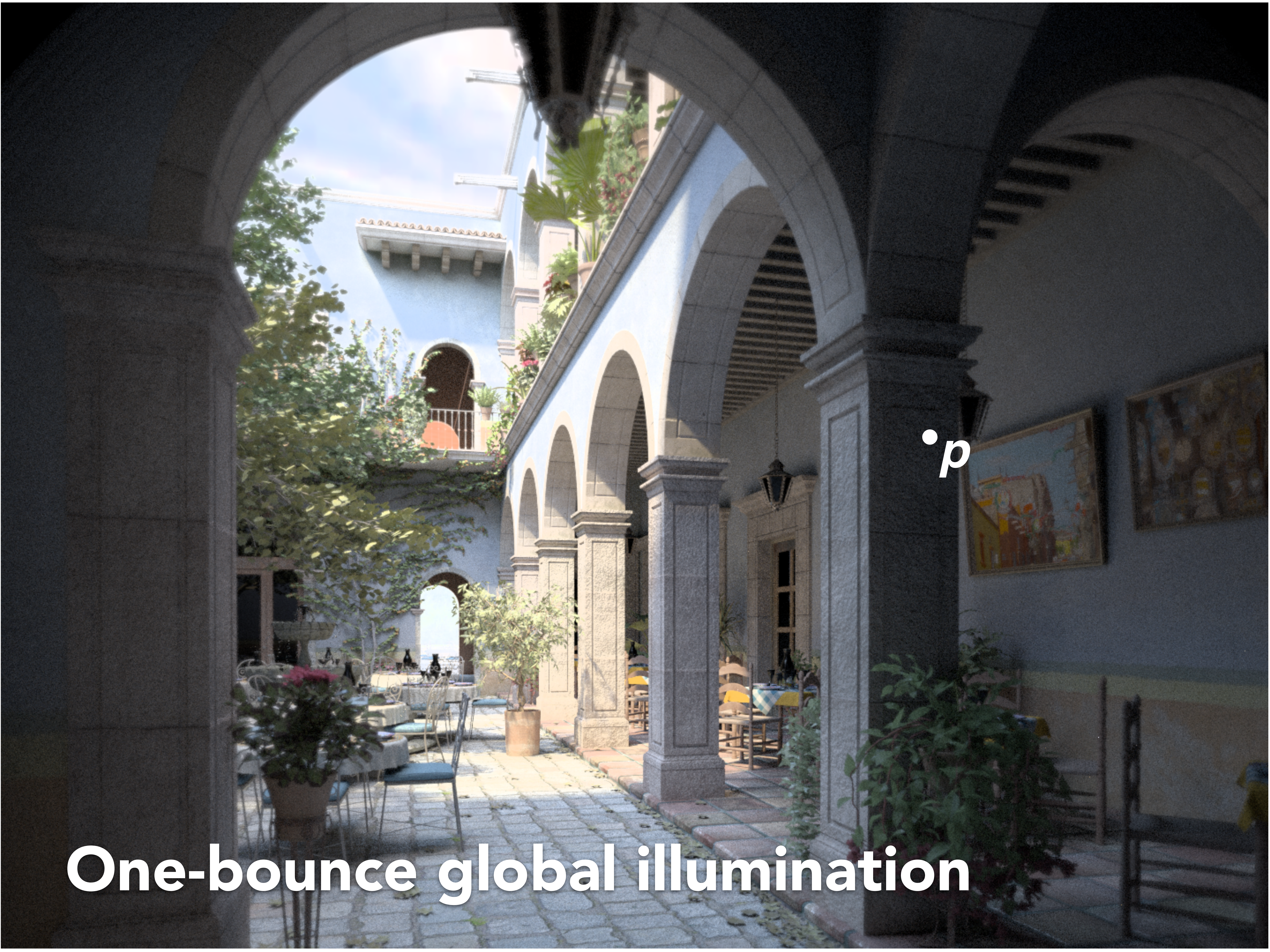
"in" direction

"out" direction

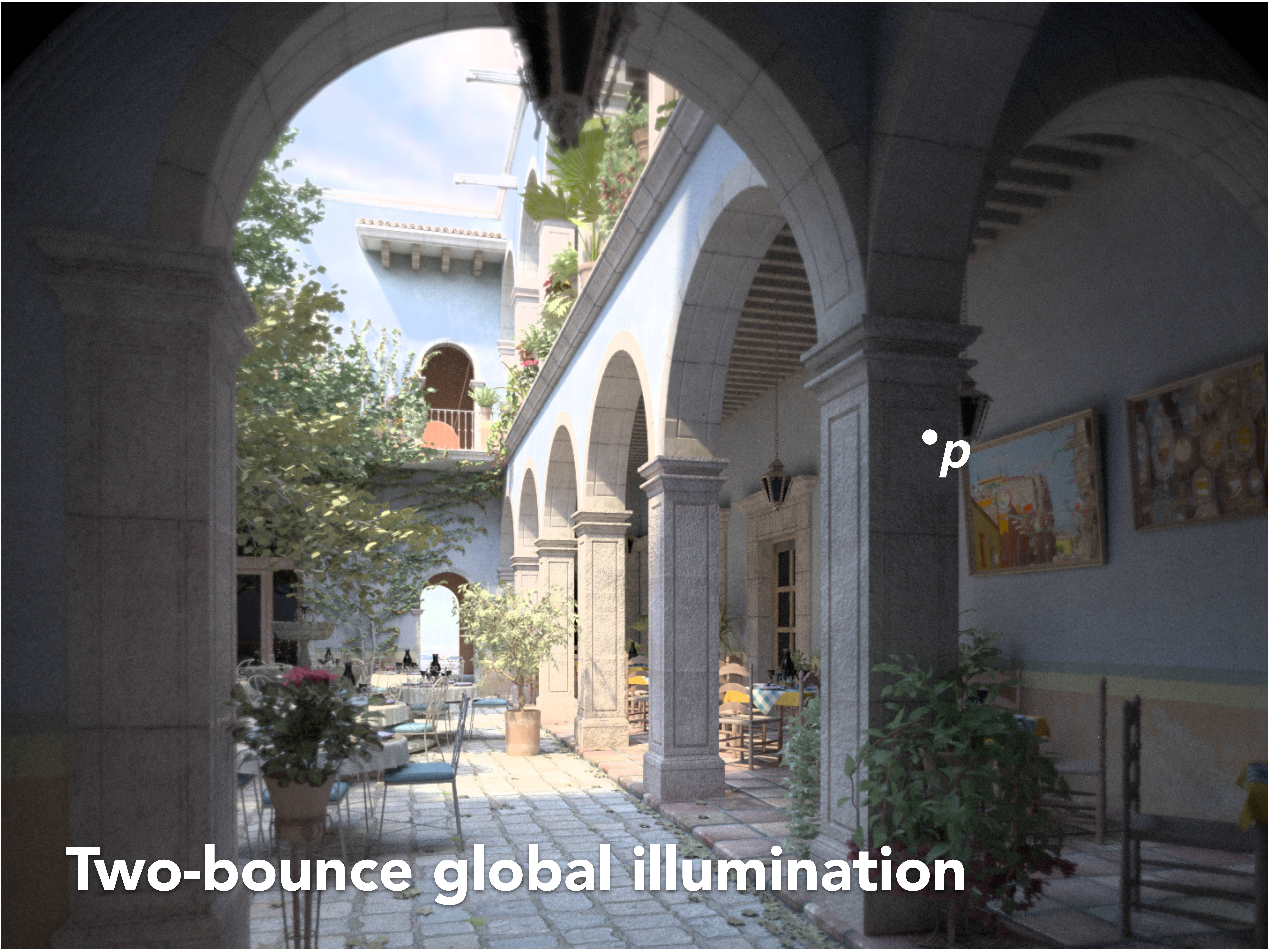
"in" direction



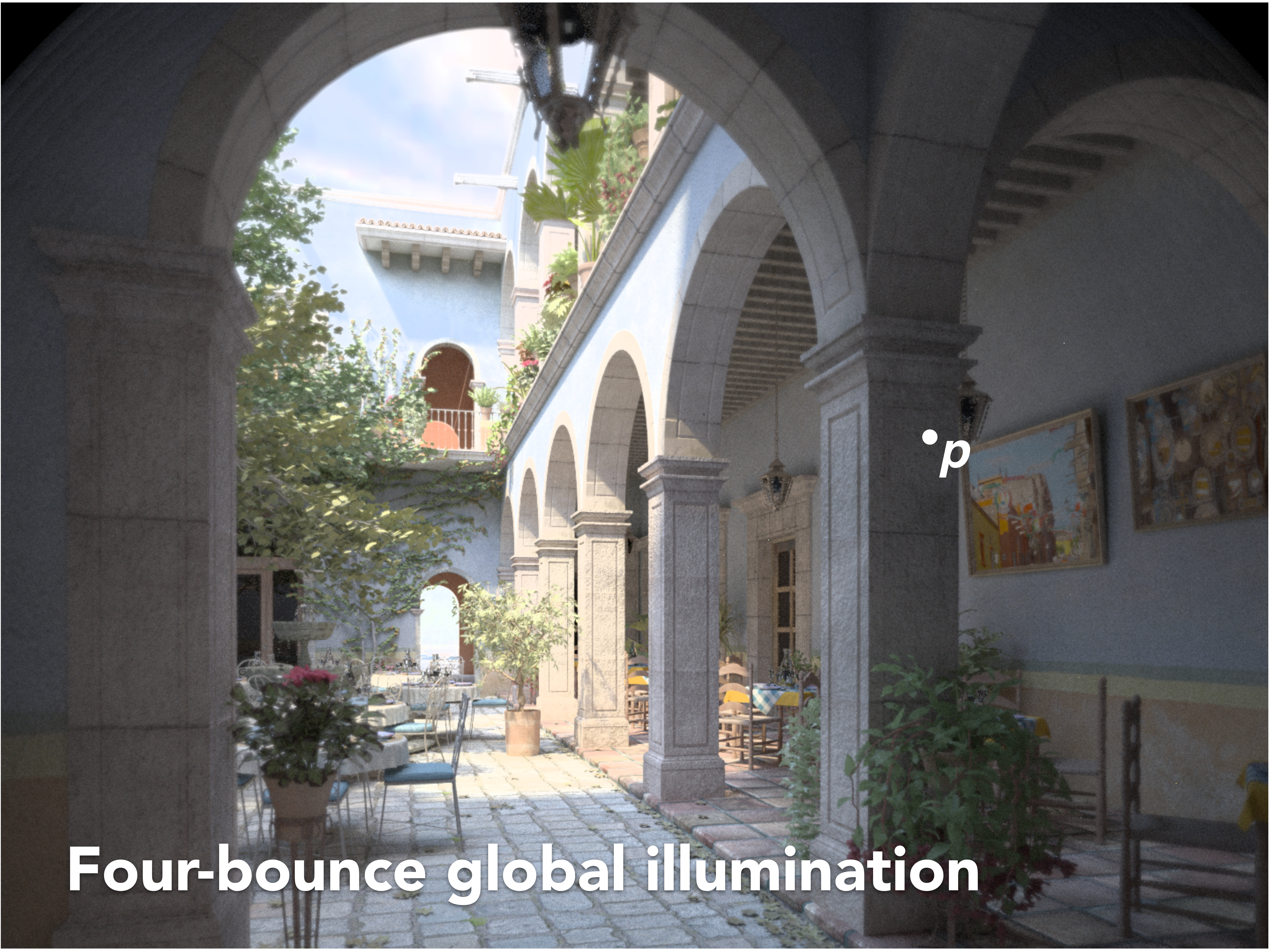
Direct illumination



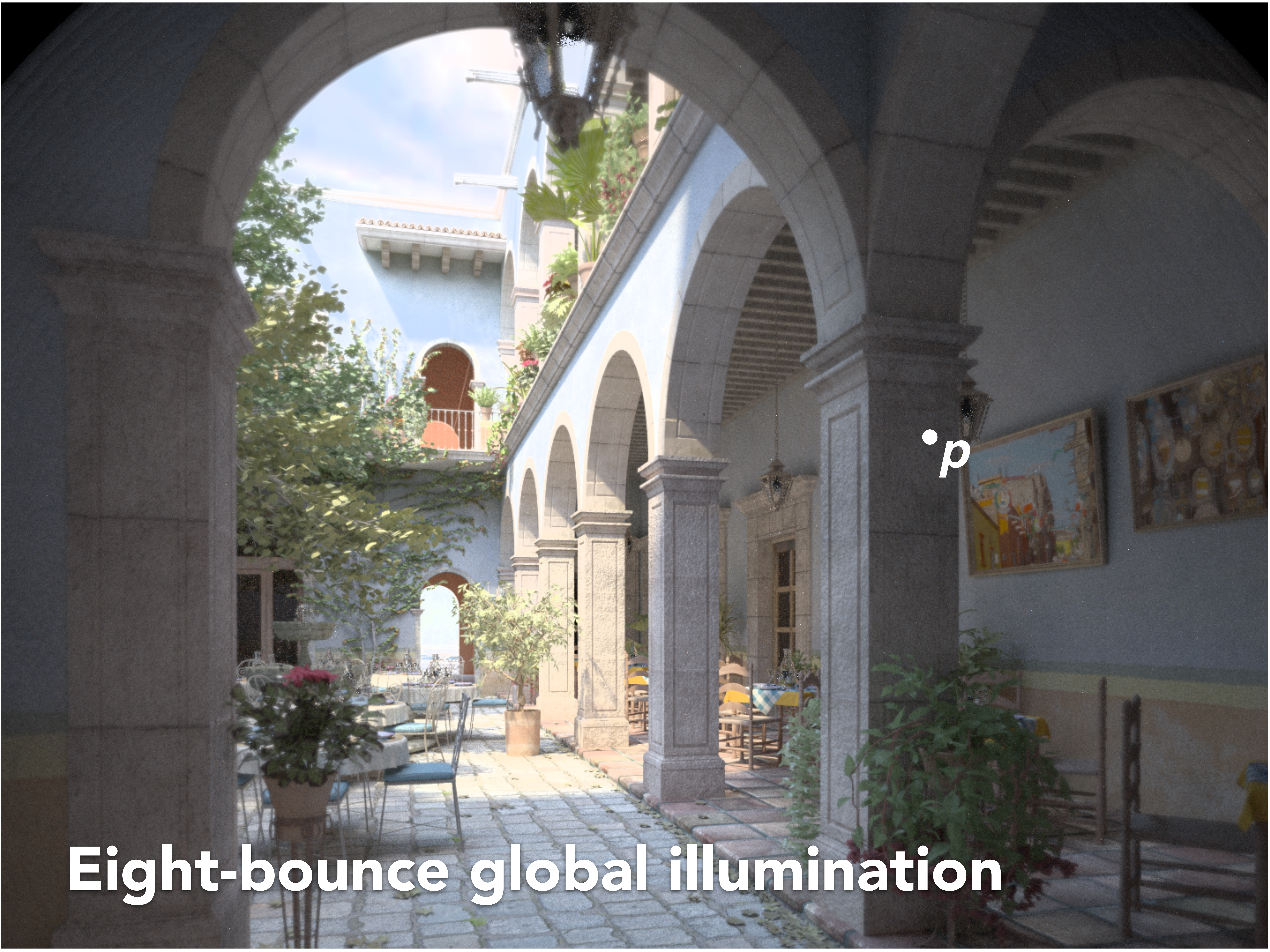
One-bounce global illumination



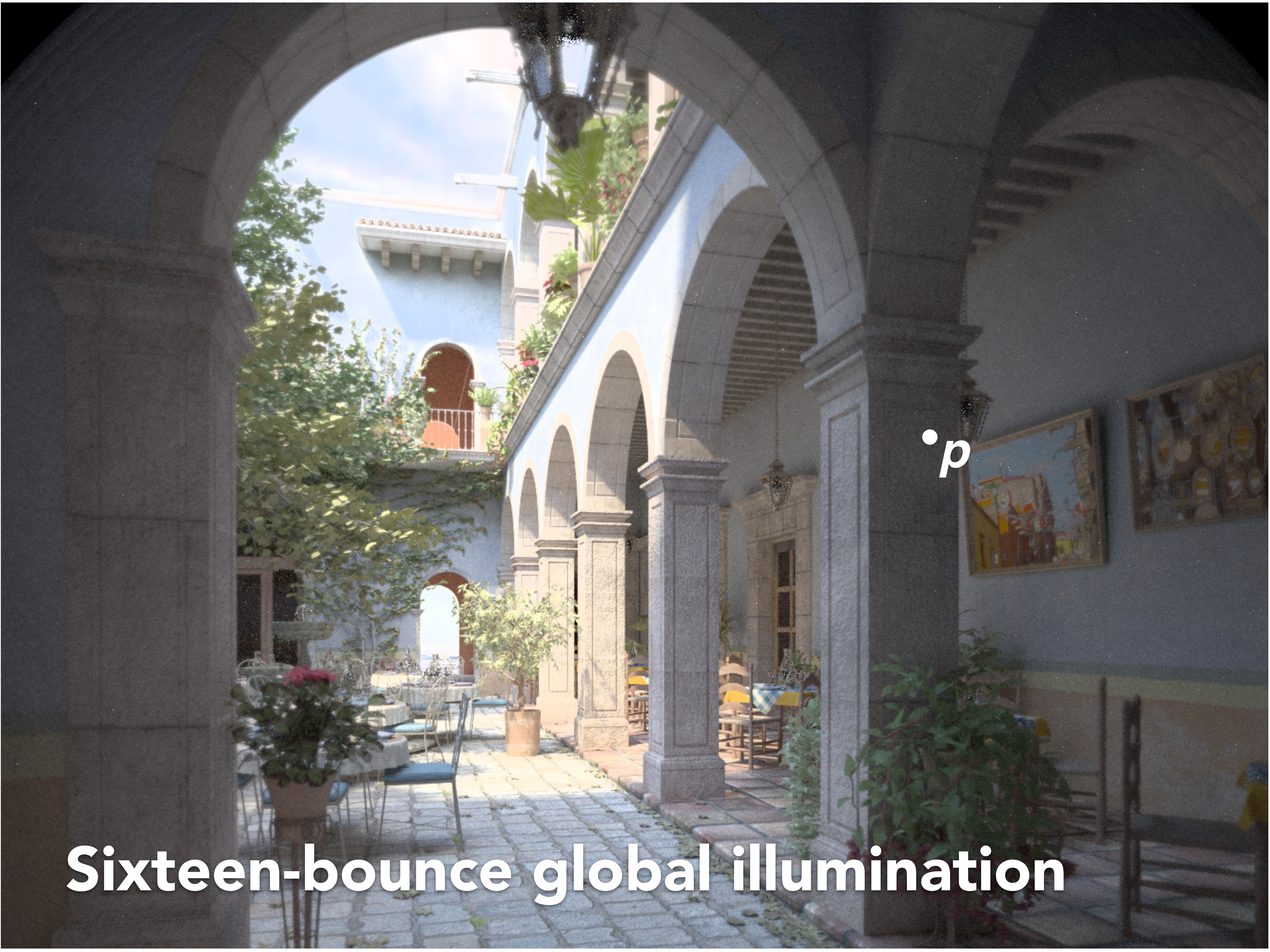
Two-bounce global illumination



Four-bounce global illumination

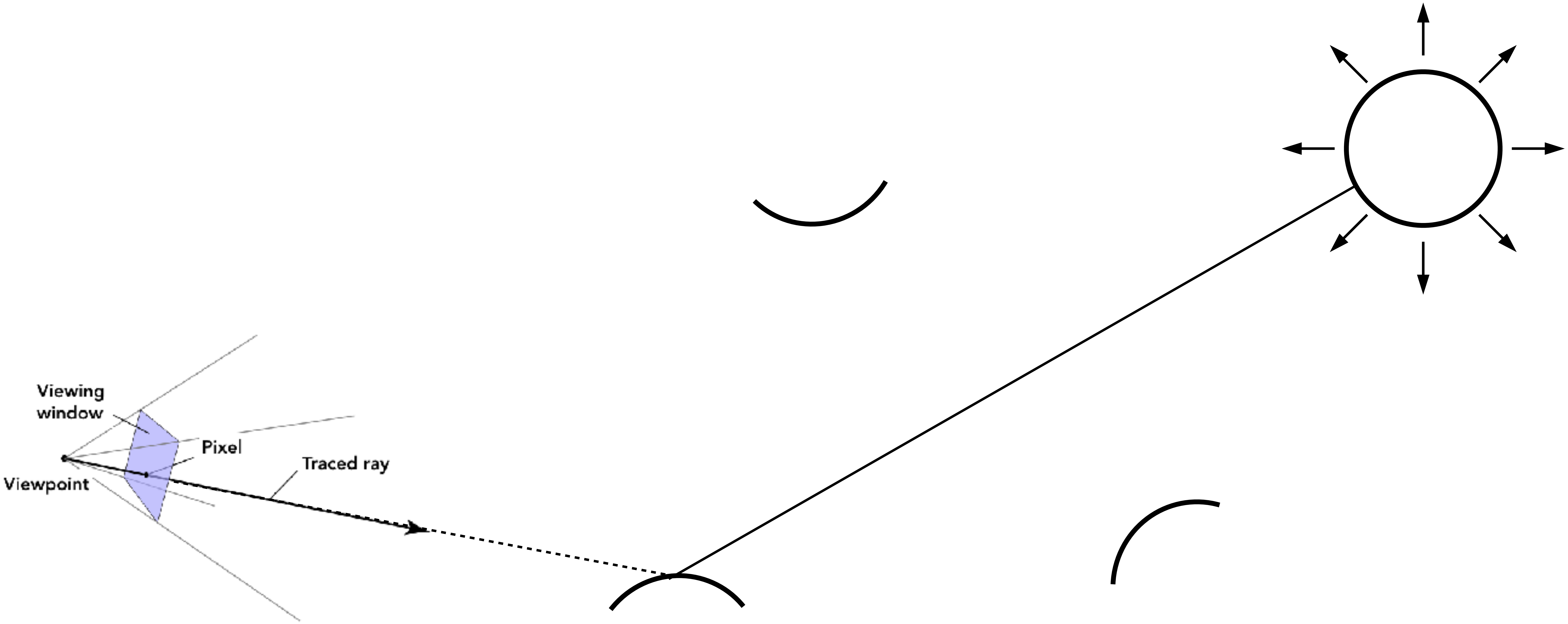


Eight-bounce global illumination



Sixteen-bounce global illumination

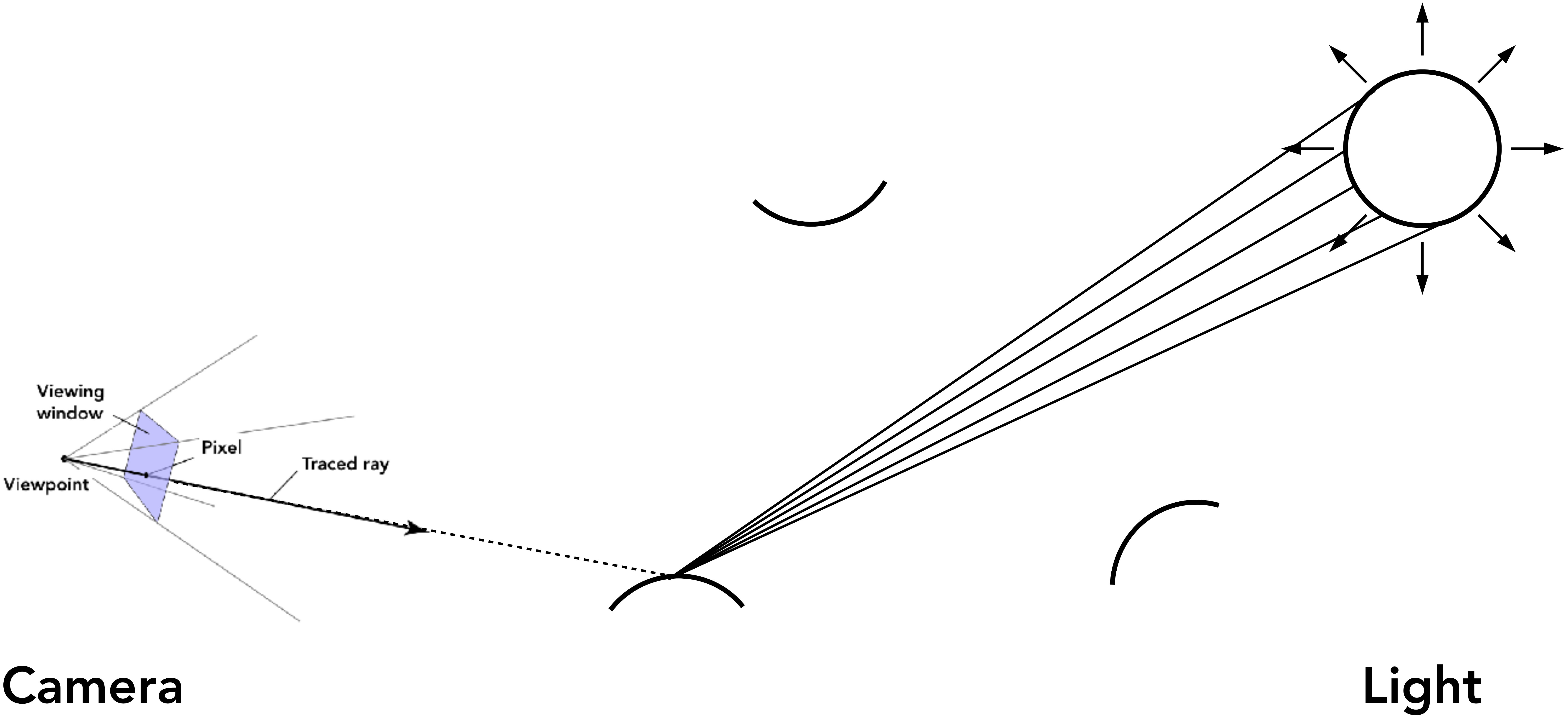
1-Bounce Path Connecting Ray to Light



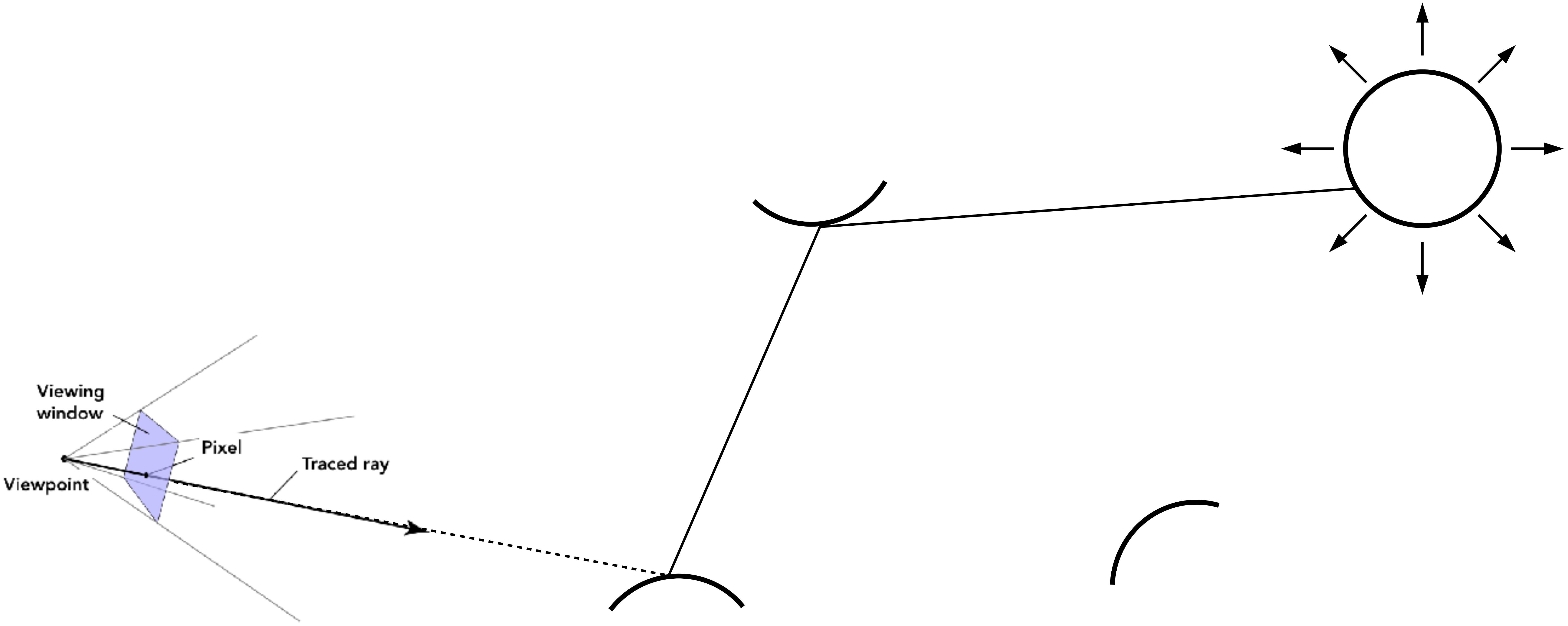
Camera

Light

1-Bounce Paths Connecting Ray to Light



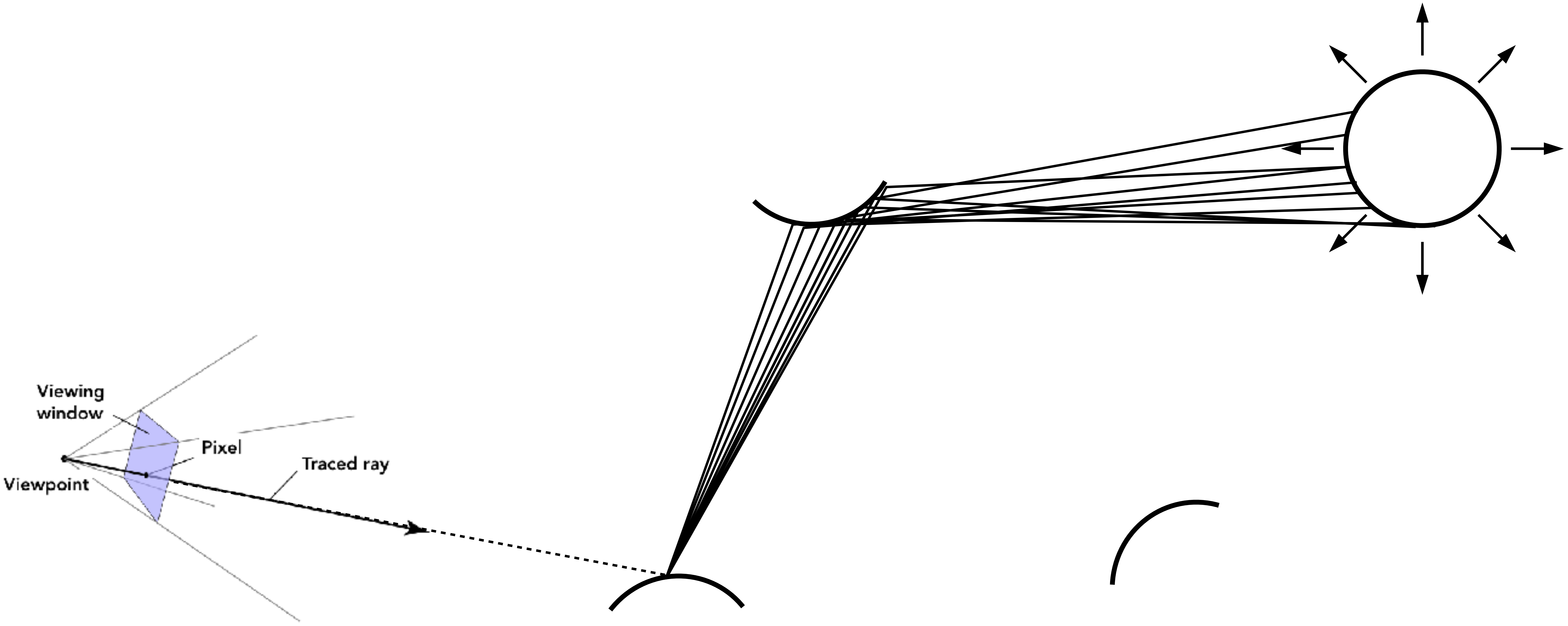
2-Bounce Path Connecting Ray to Light



Camera

Light

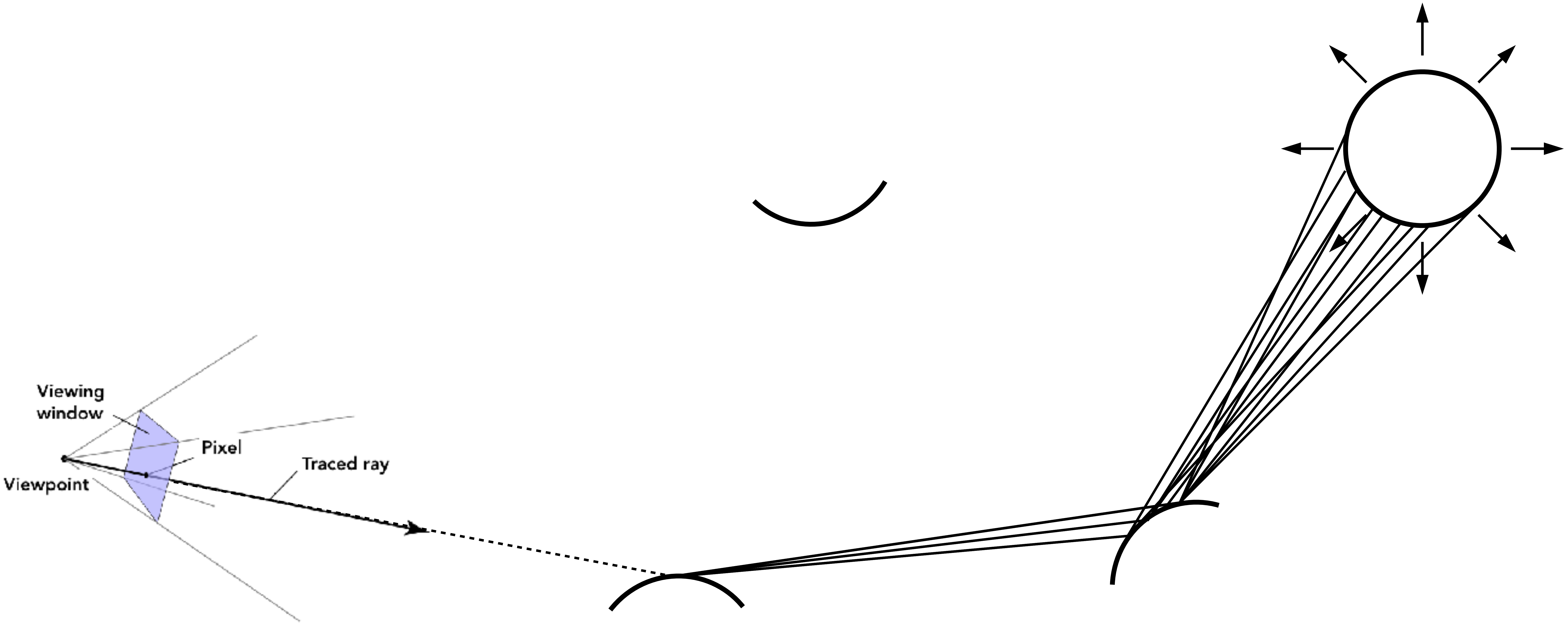
2-Bounce Paths Connecting Ray to Light



Camera

Light

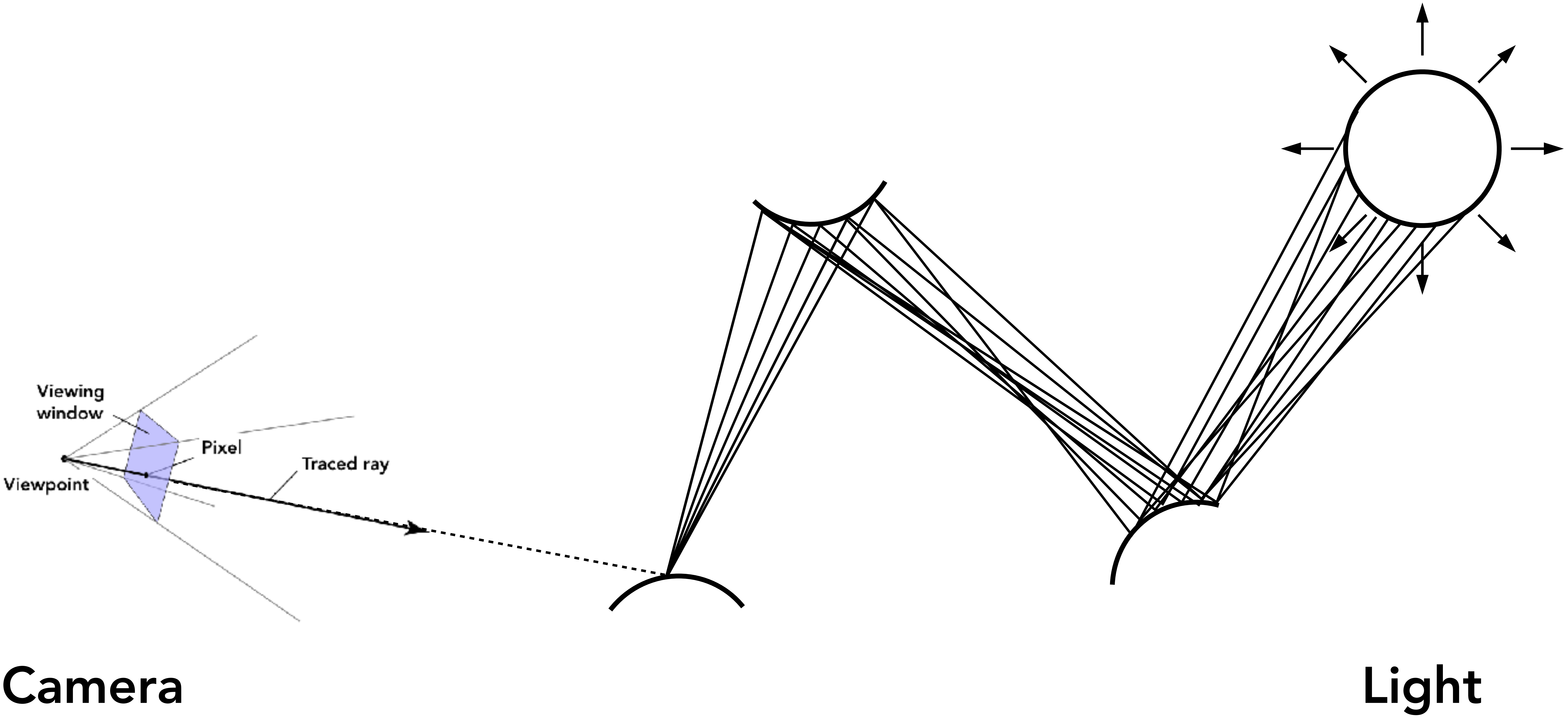
2-Bounce Paths Connecting Ray to Light



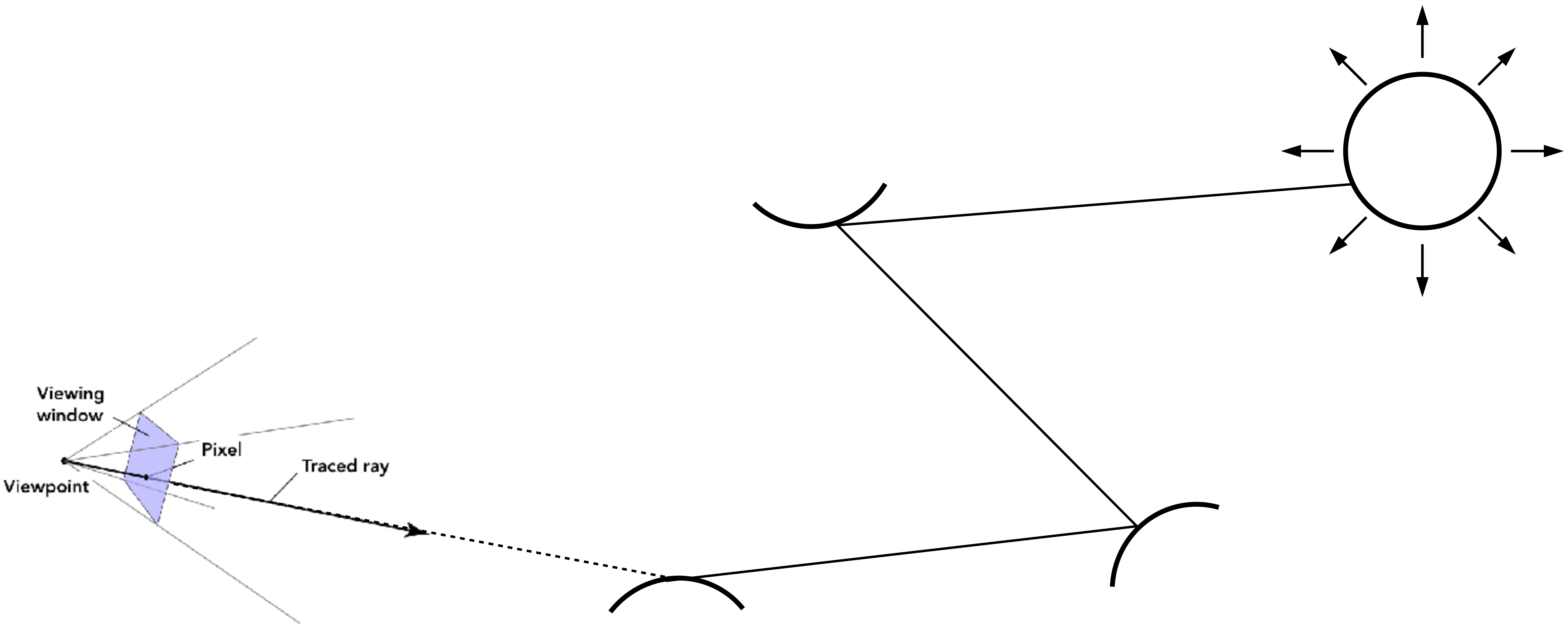
Camera

Light

3-Bounce Paths Connecting Ray to Light



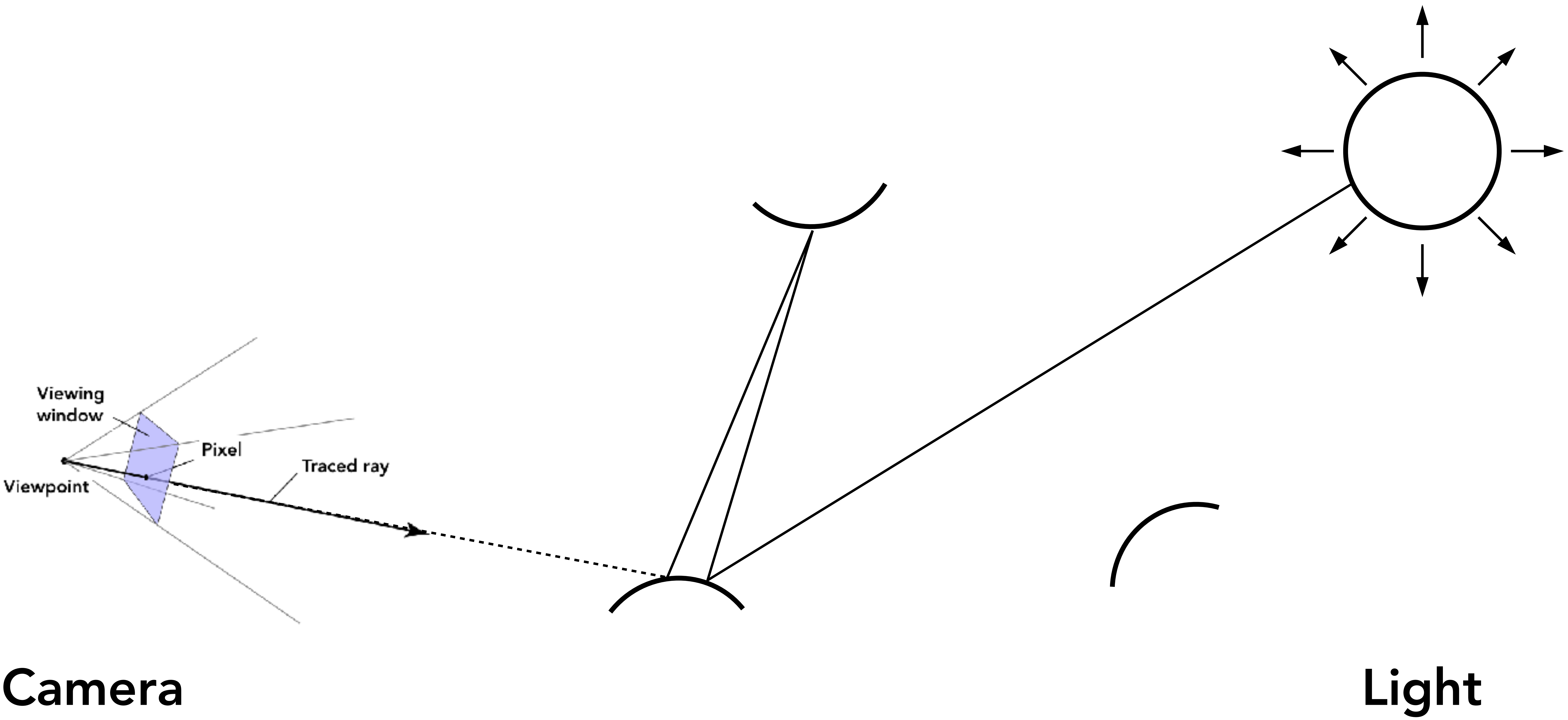
3-Bounce Path Connecting Ray to Light



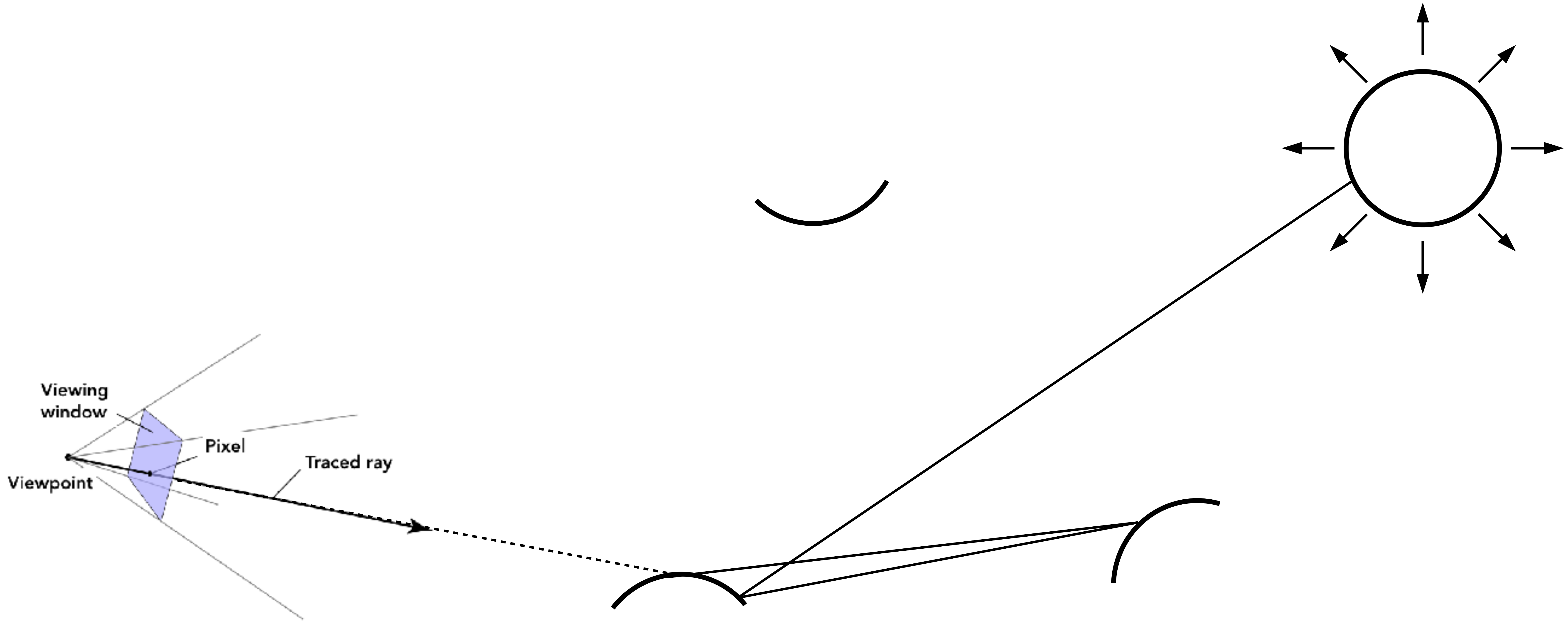
Camera

Light

3-Bounce Path Connecting Ray to Light



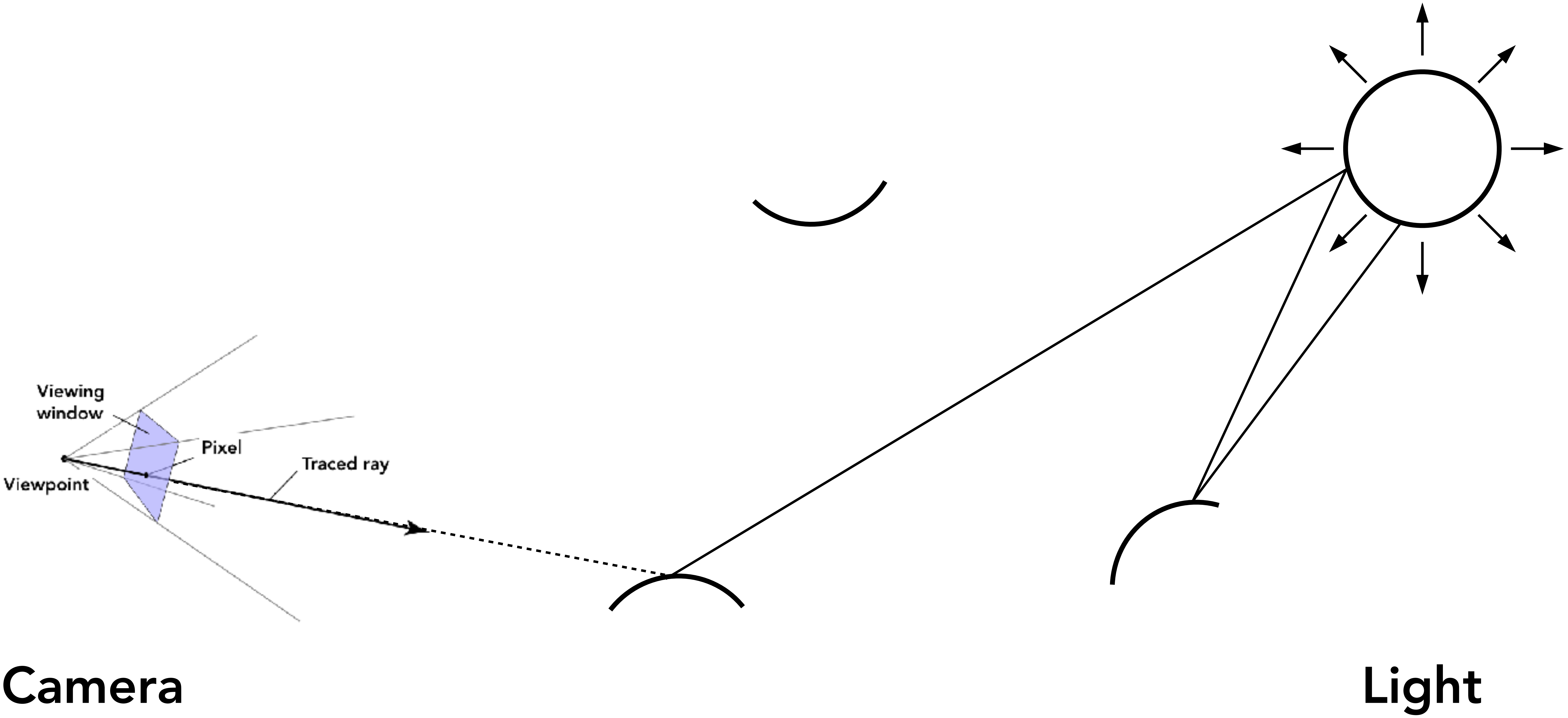
3-Bounce Path Connecting Ray to Light



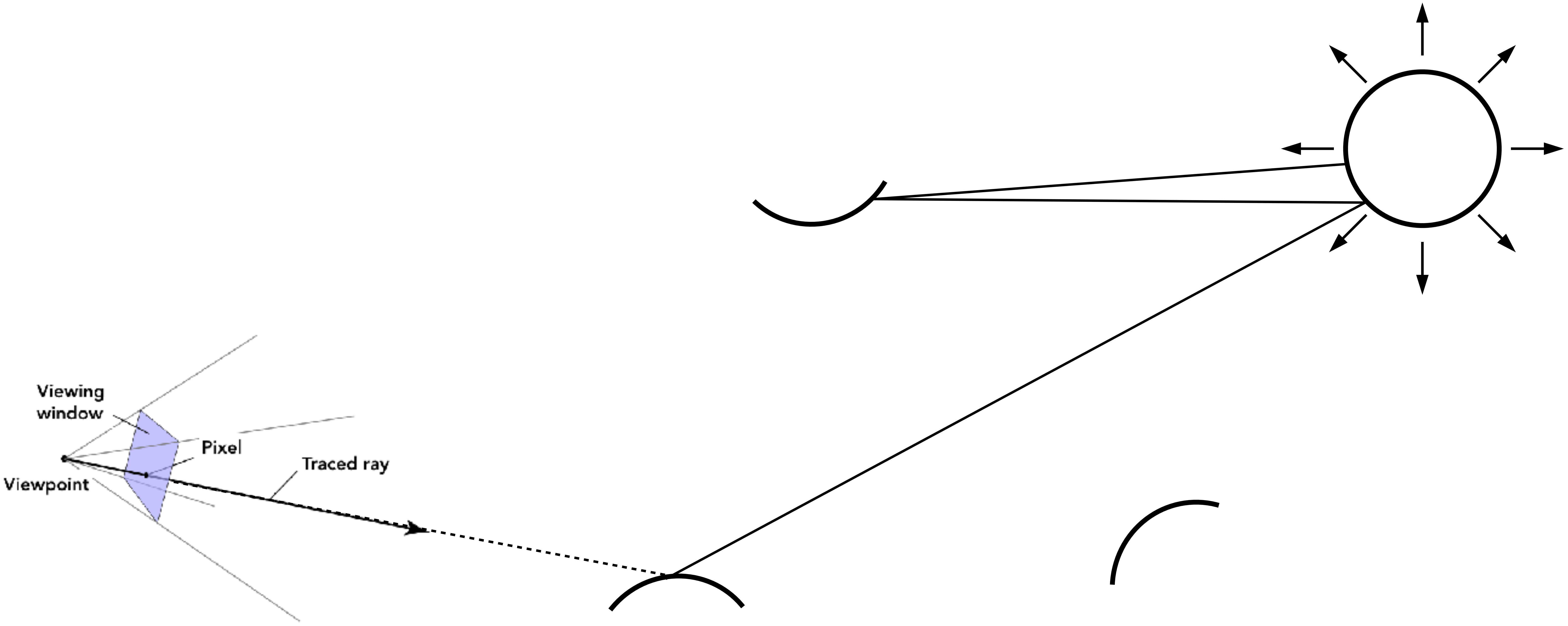
Camera

Light

3-Bounce Path Connecting Ray to Light



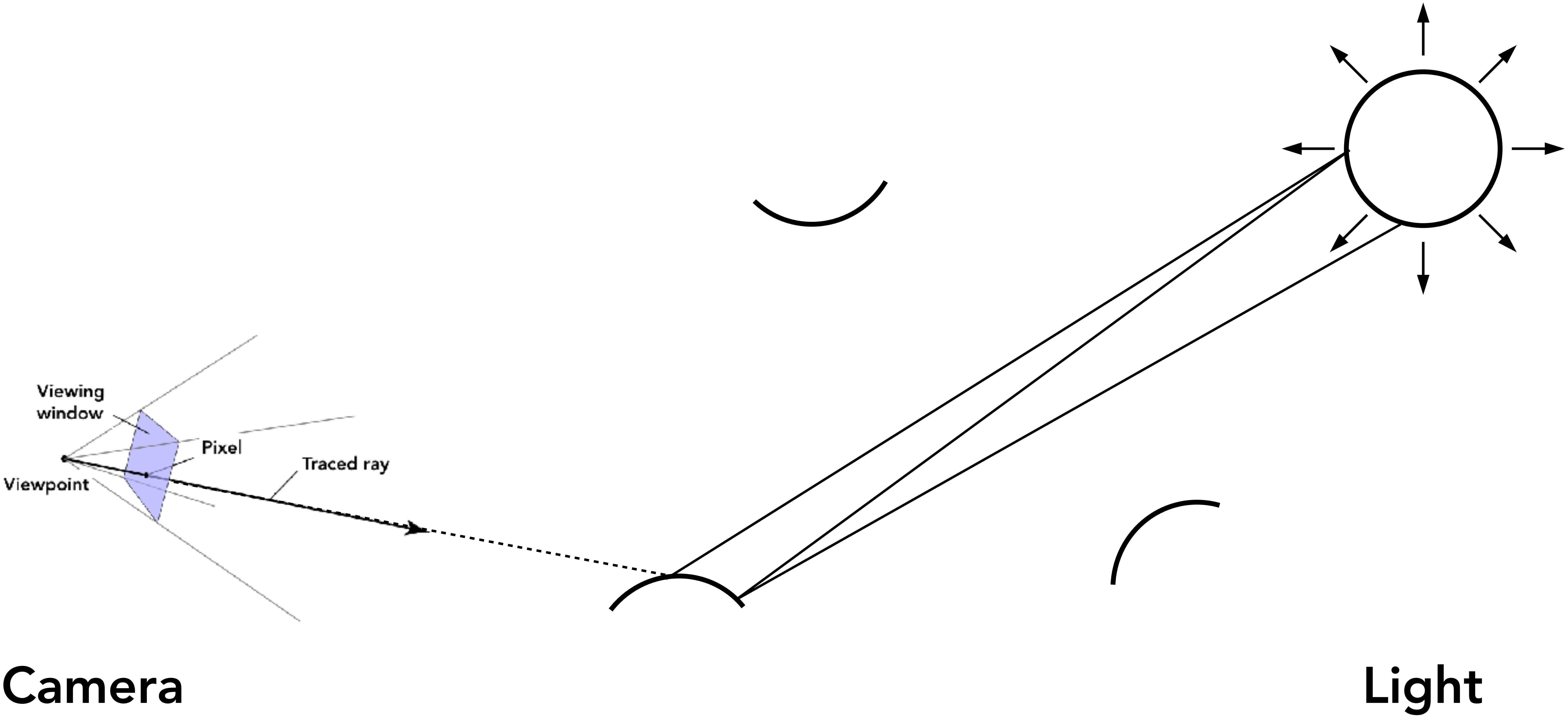
3-Bounce Path Connecting Ray to Light



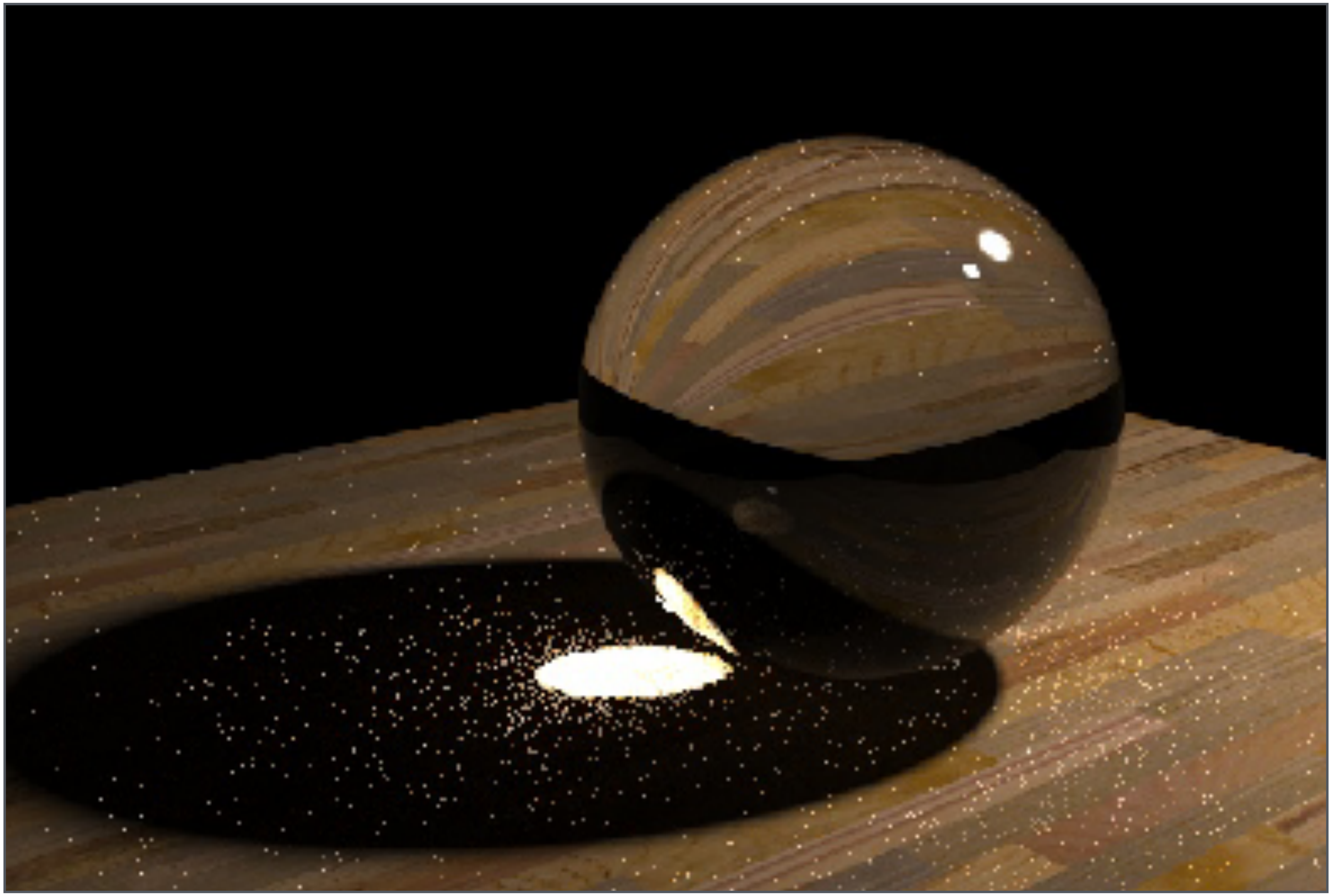
Camera

Light

3-Bounce Path Connecting Ray to Light



A Challenging Scene for Path Tracing – Why?



Henrik Wann Jensen

1000 paths / pixel

Summary of Intuition on G.I. & P.T.

- Operator notation leads to insight that solution is adding successive bounces of light
- Trace N paths through a pixel, sample radiance
- Build paths by recursively tracing to next surface point and choosing a random reflection direction. At each surface, sum emitted light and reflected light
- How to terminate paths? We use Russian Roulette to kill probabilistically.
- How to reduce noise? Use importance sampling in choosing random direction. Two ways: importance sample the lights, and importance sample the BRDF.