

Physics-based rendering
and its applications in
computational photography and imaging

Adithya Pediredla

Dartmouth College

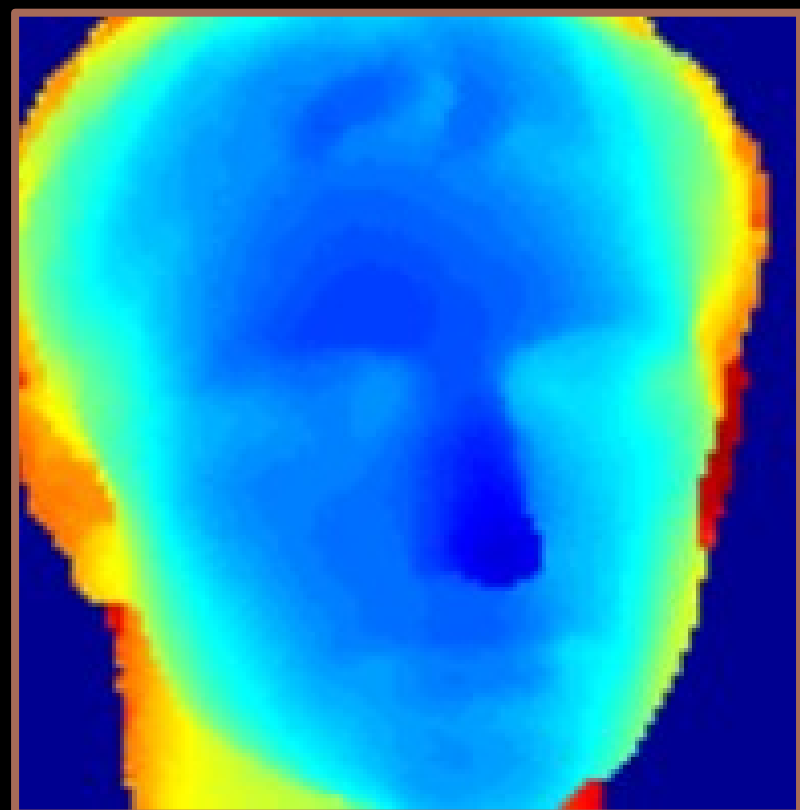
adithya.k.pediredla@dartmouth.com

Ioannis Gkioulekas

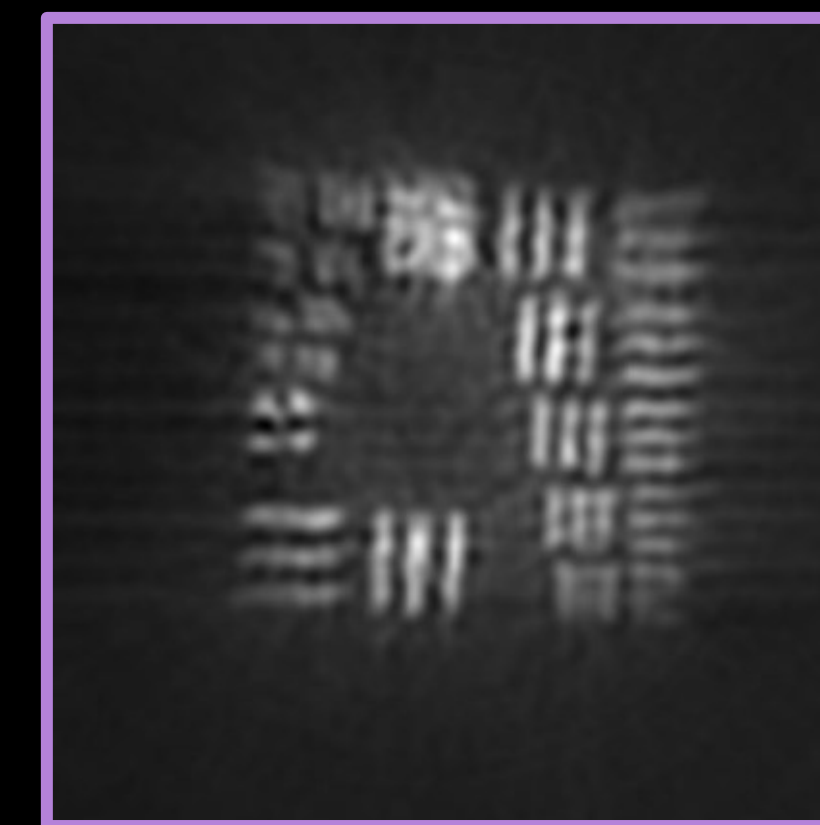
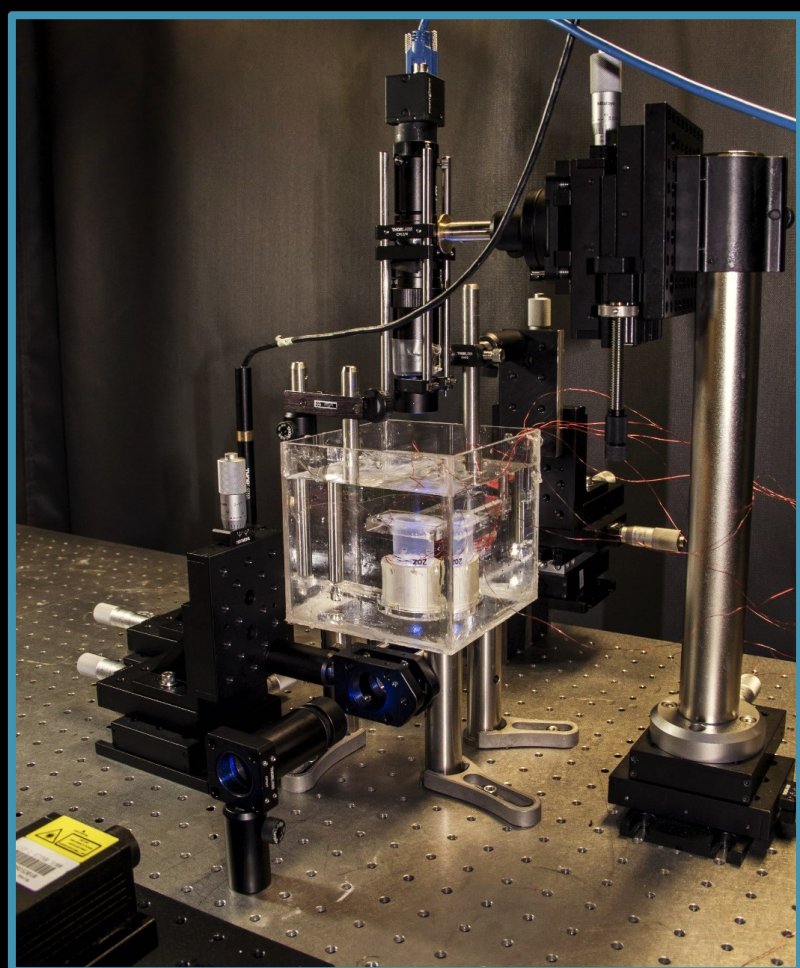
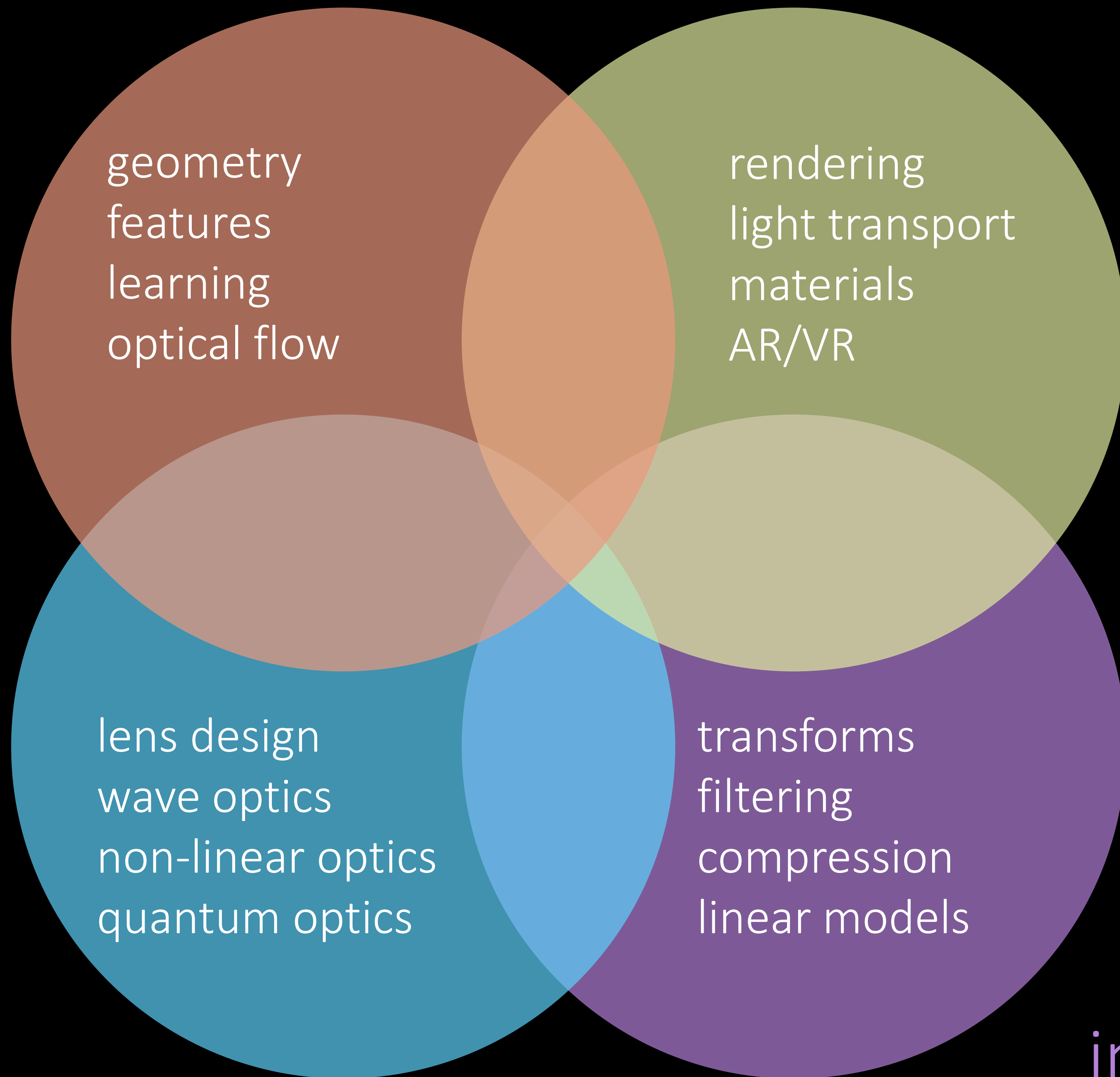
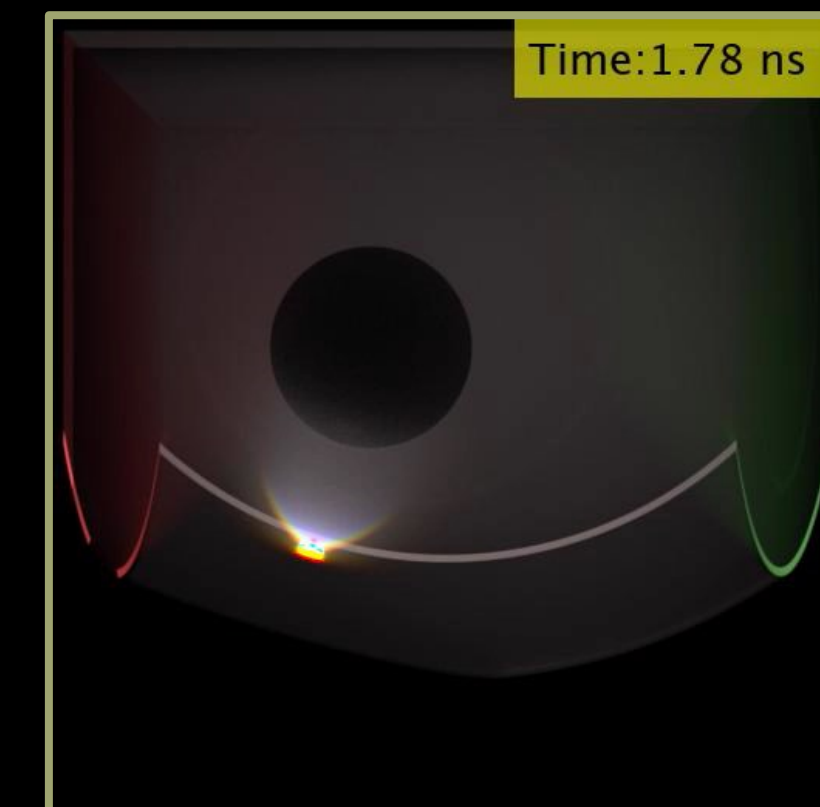
Carnegie Mellon University

igkioule@andrew.cmu.edu

computer
vision



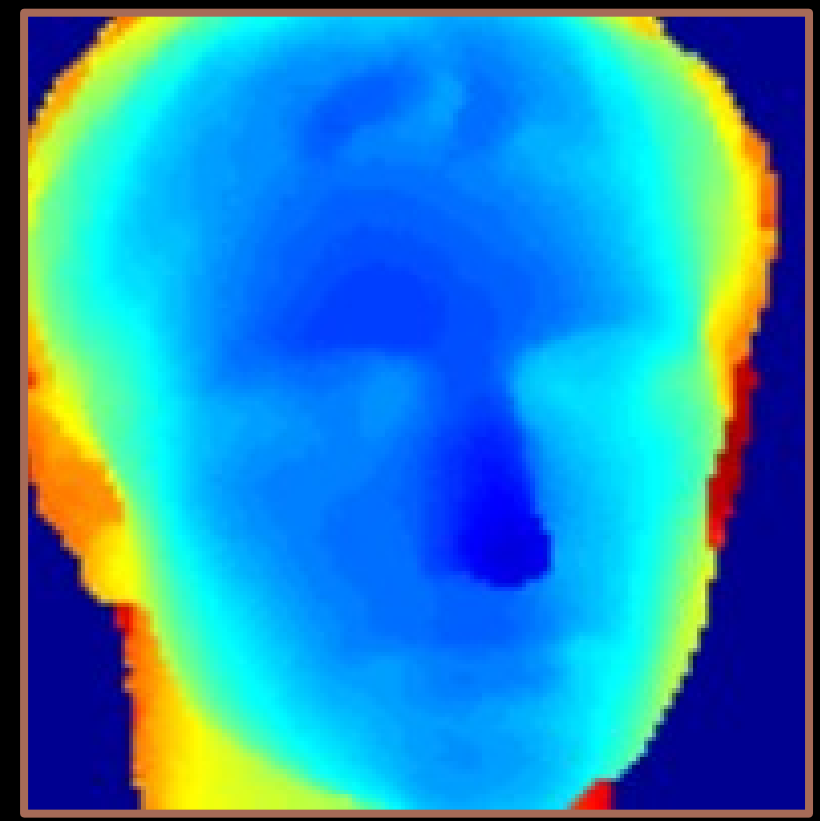
computer
graphics



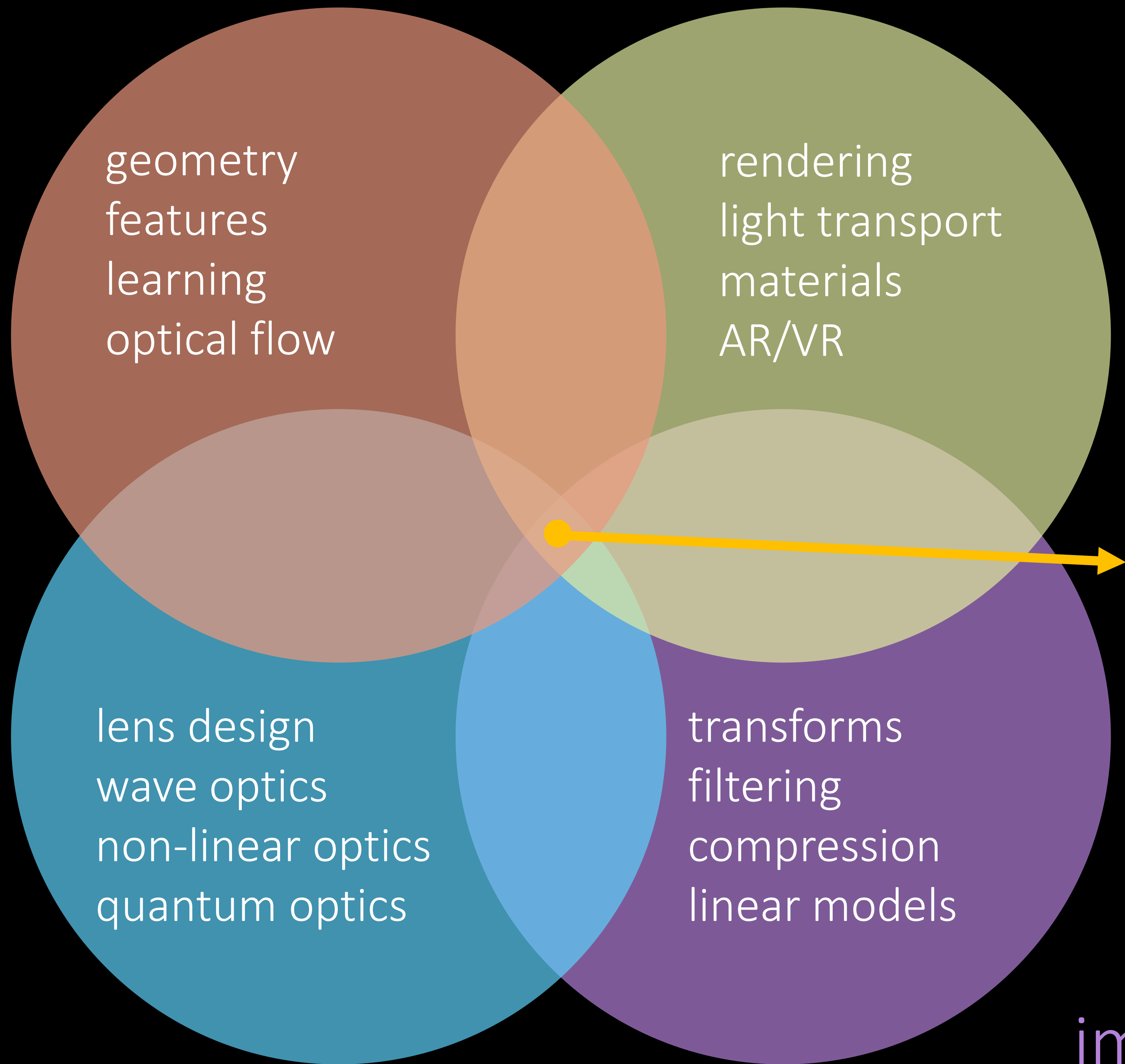
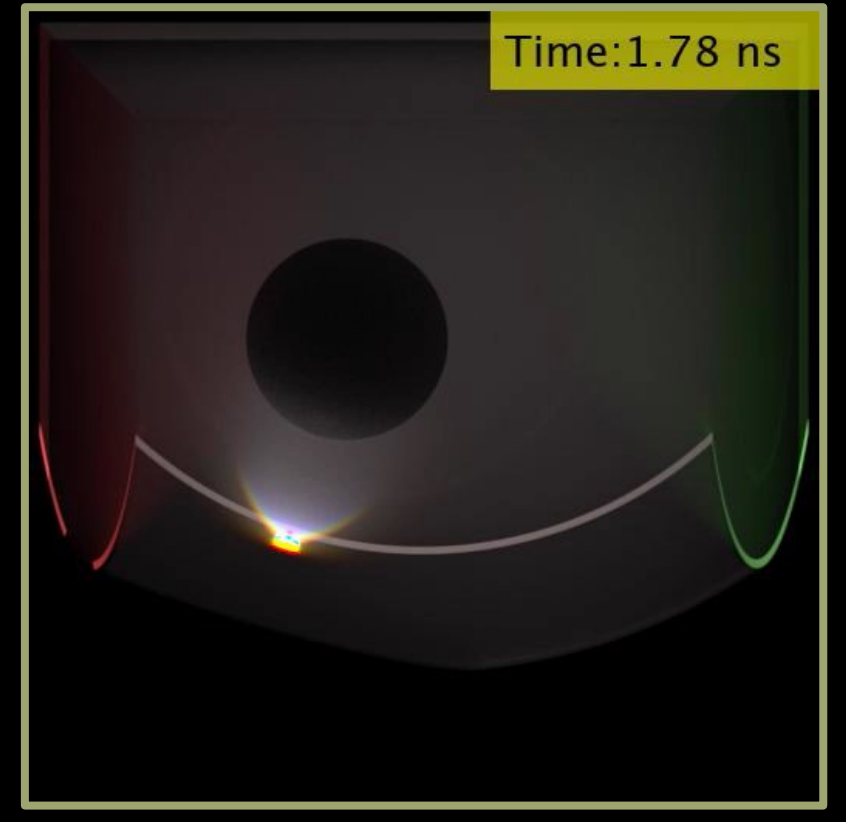
optics

image processing

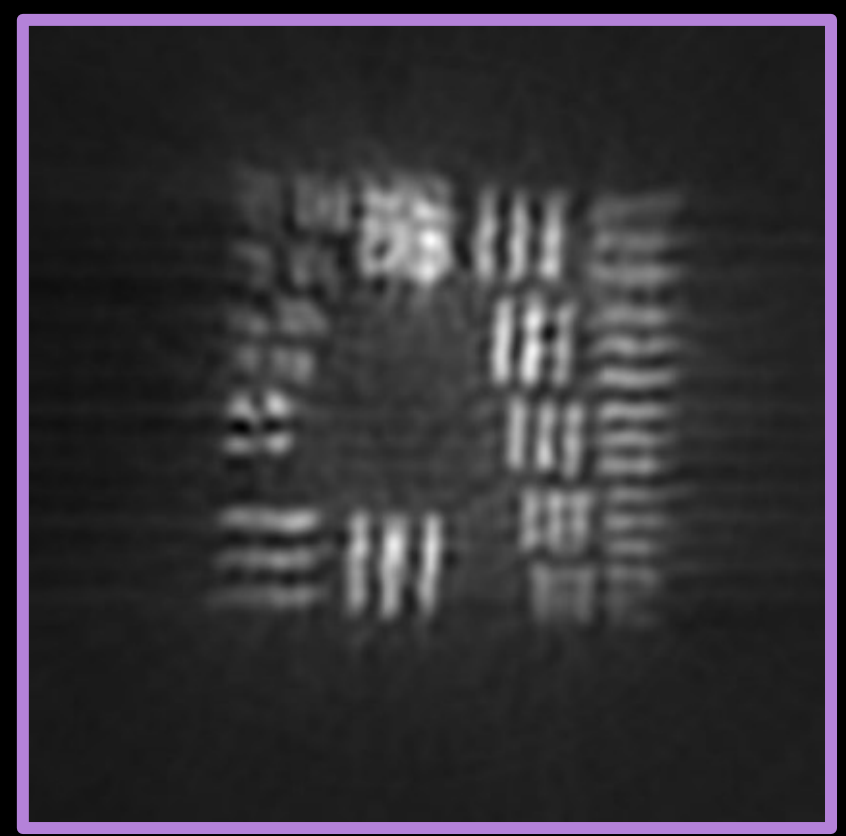
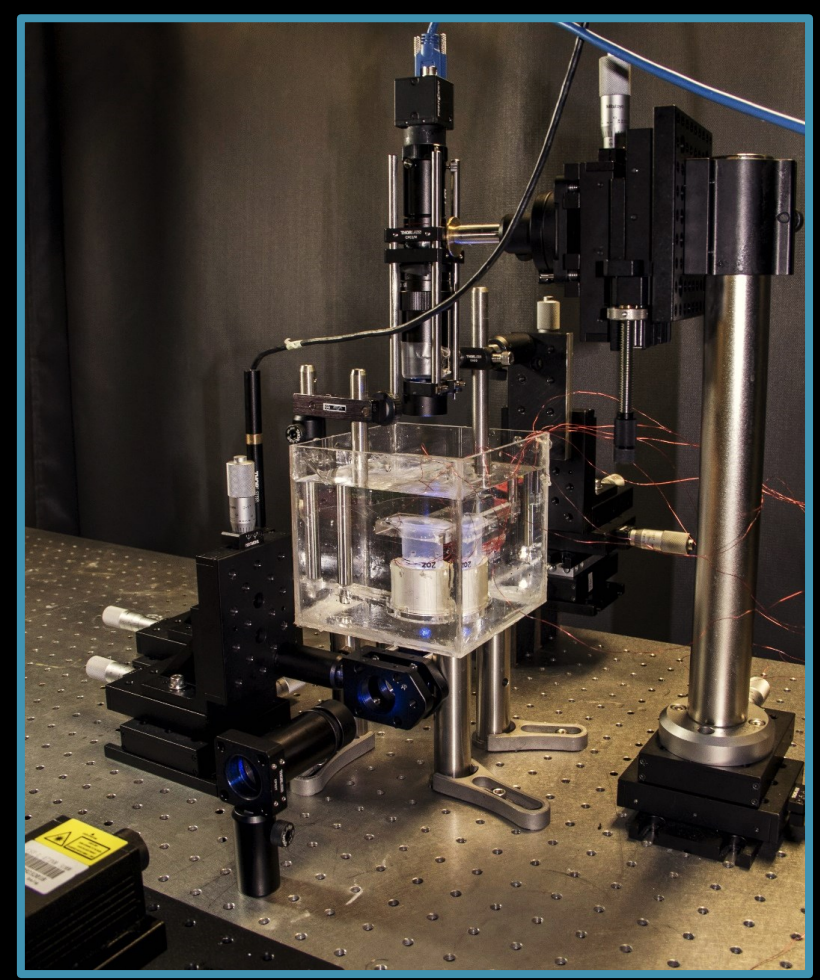
computer vision



computer graphics



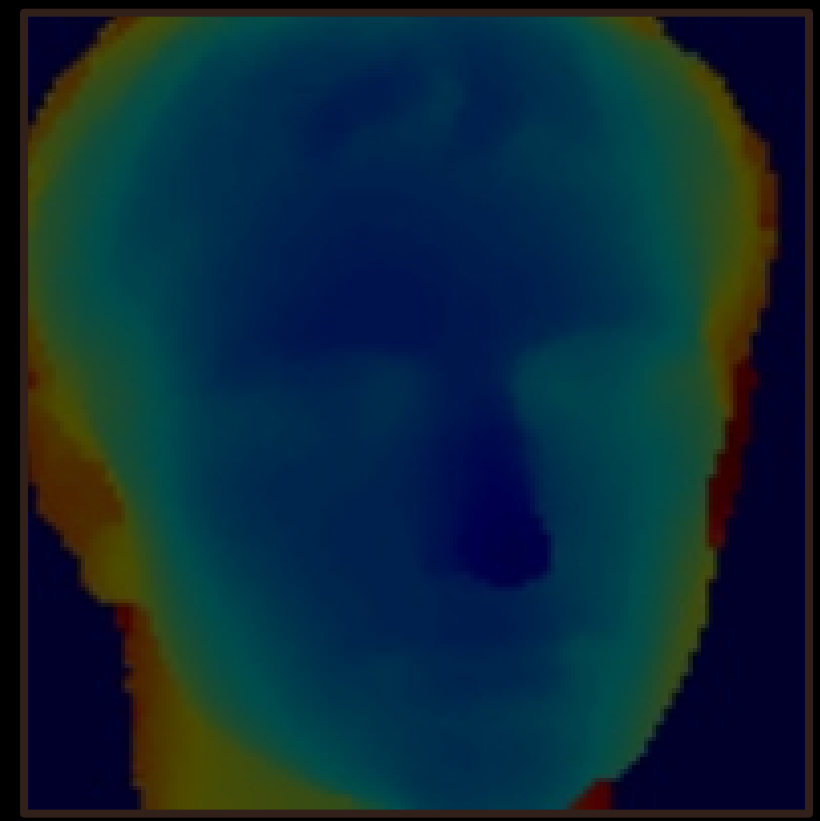
computational imaging



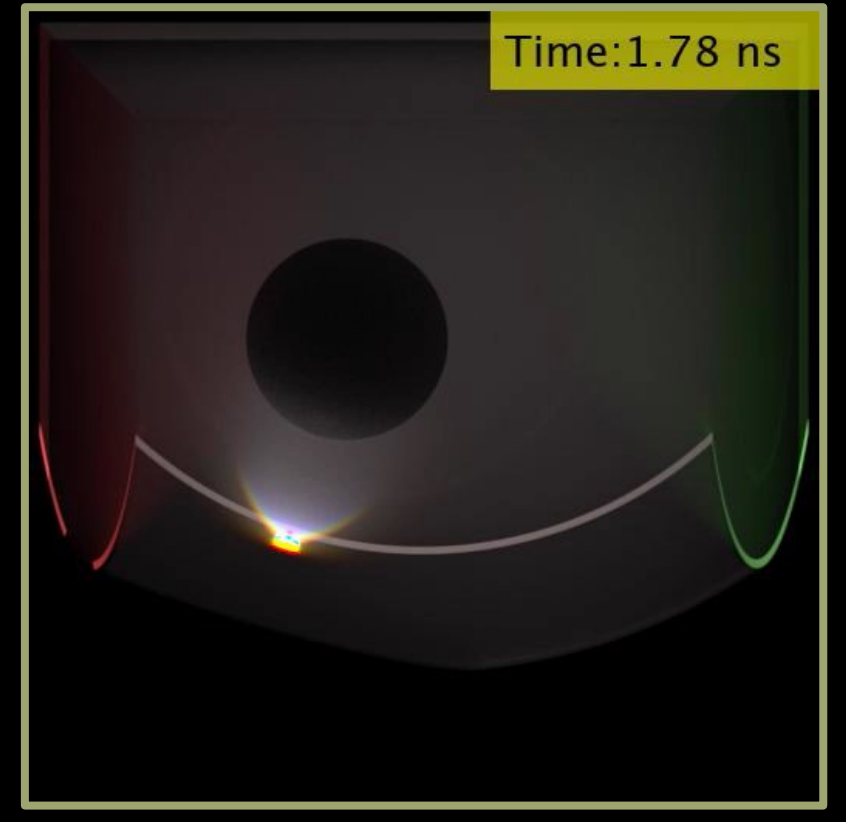
optics

image processing

computer vision



computer graphics



geometry
features
learning
optical flow

rendering
light transport
materials
AR/VR

lens design
wave optics
non-linear optics
quantum optics

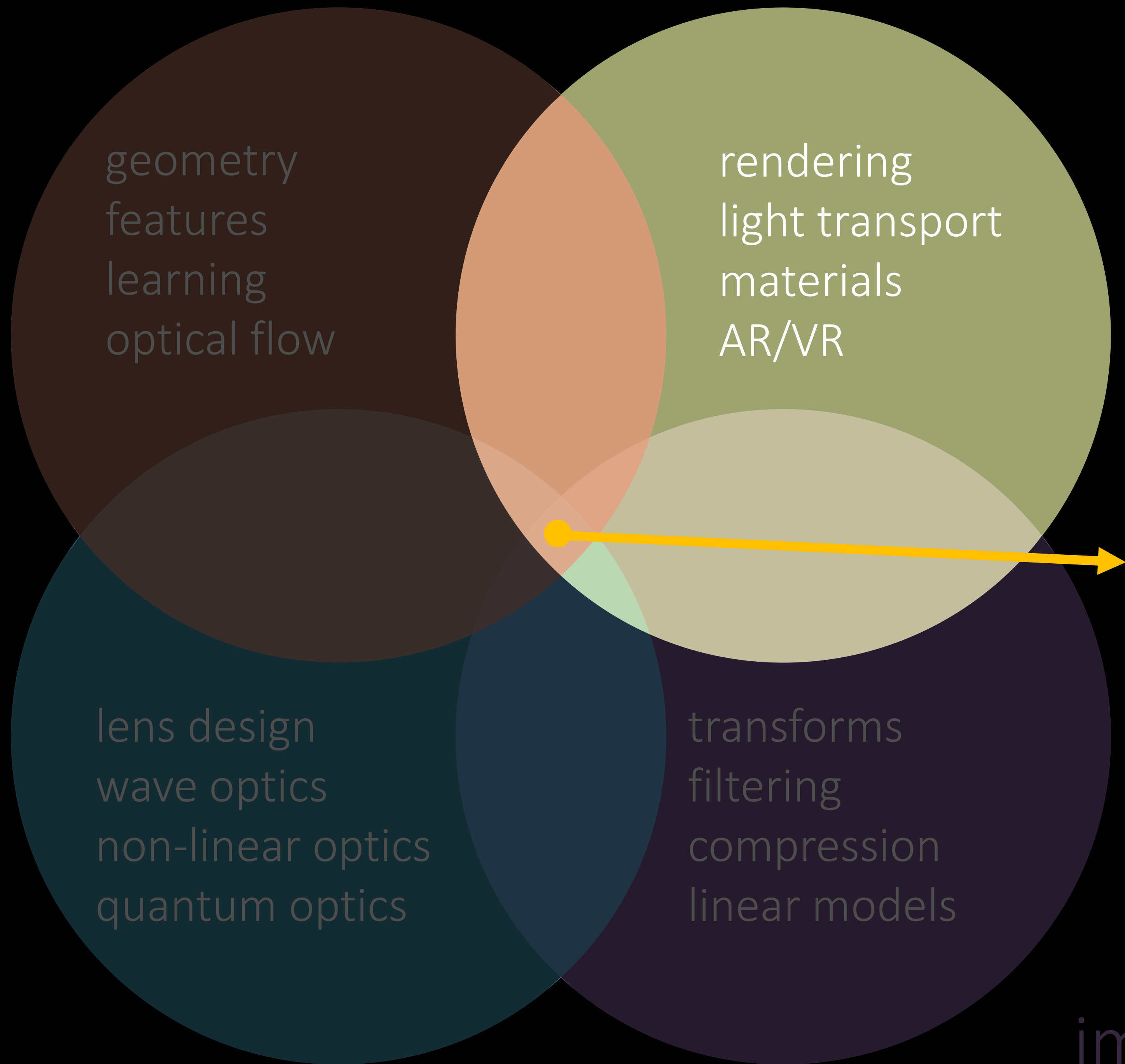
transforms
filtering
compression
linear models

computational imaging

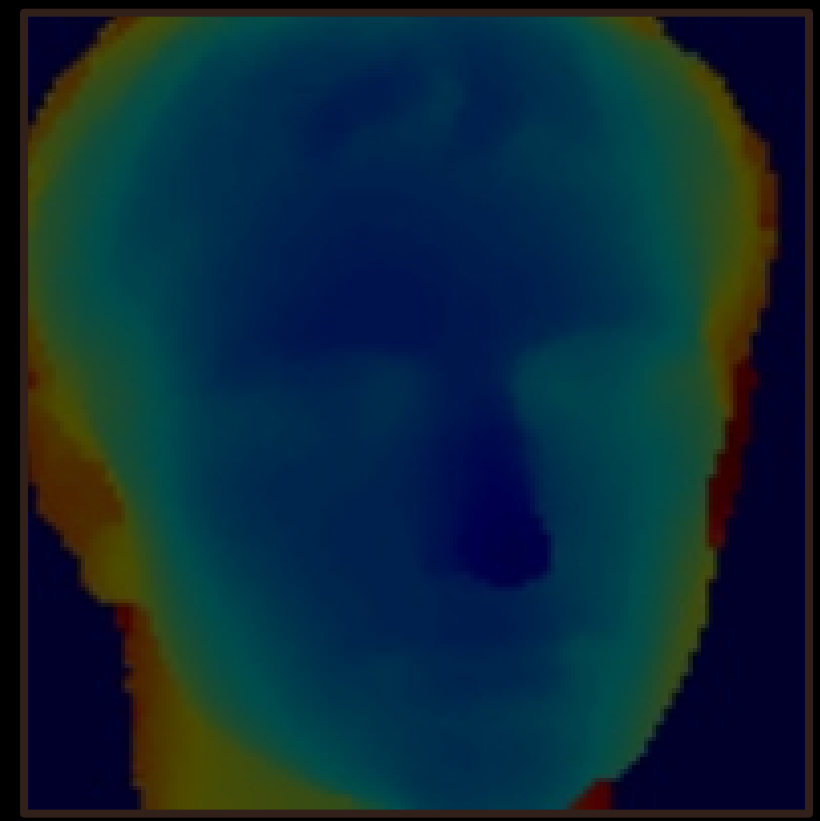


optics

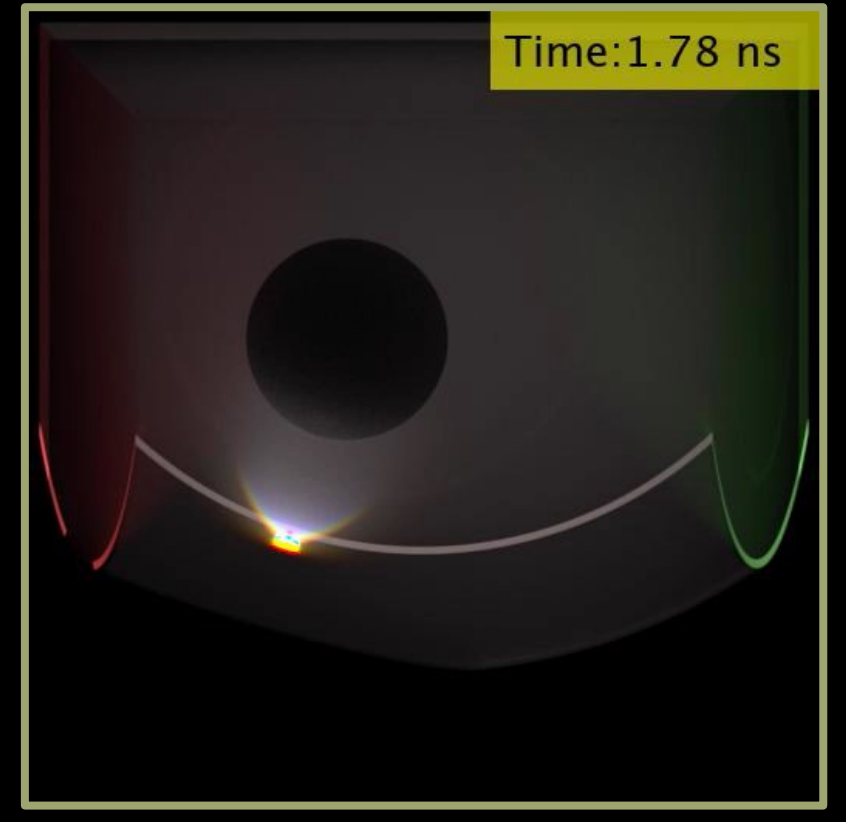
image processing



computer vision



computer graphics



geometry
features
learning
optical flow

rendering
light transport
materials
AR/VR

lens design
wave optics
non-linear optics
quantum optics

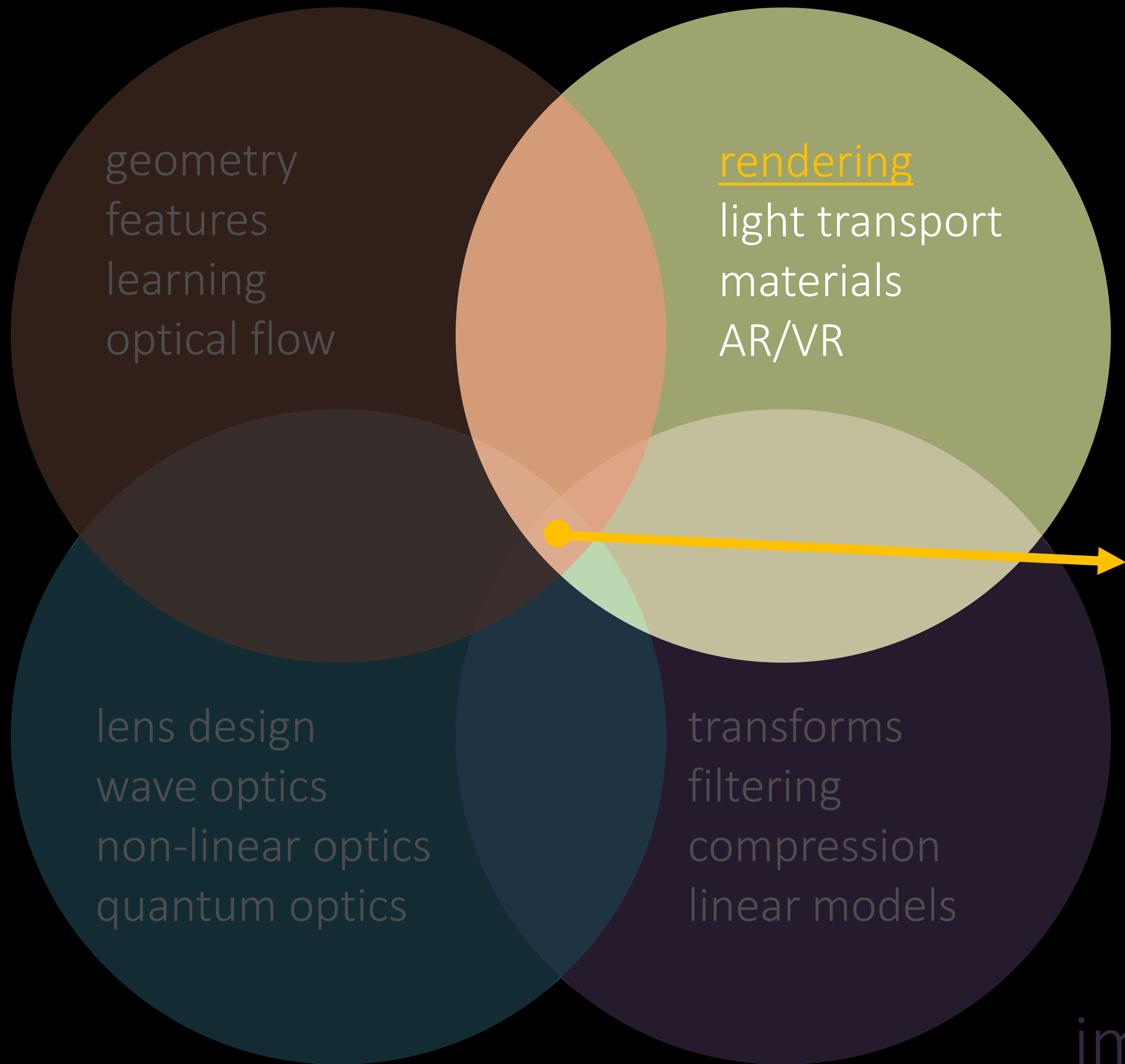
transforms
filtering
compression
linear models

computational imaging



optics

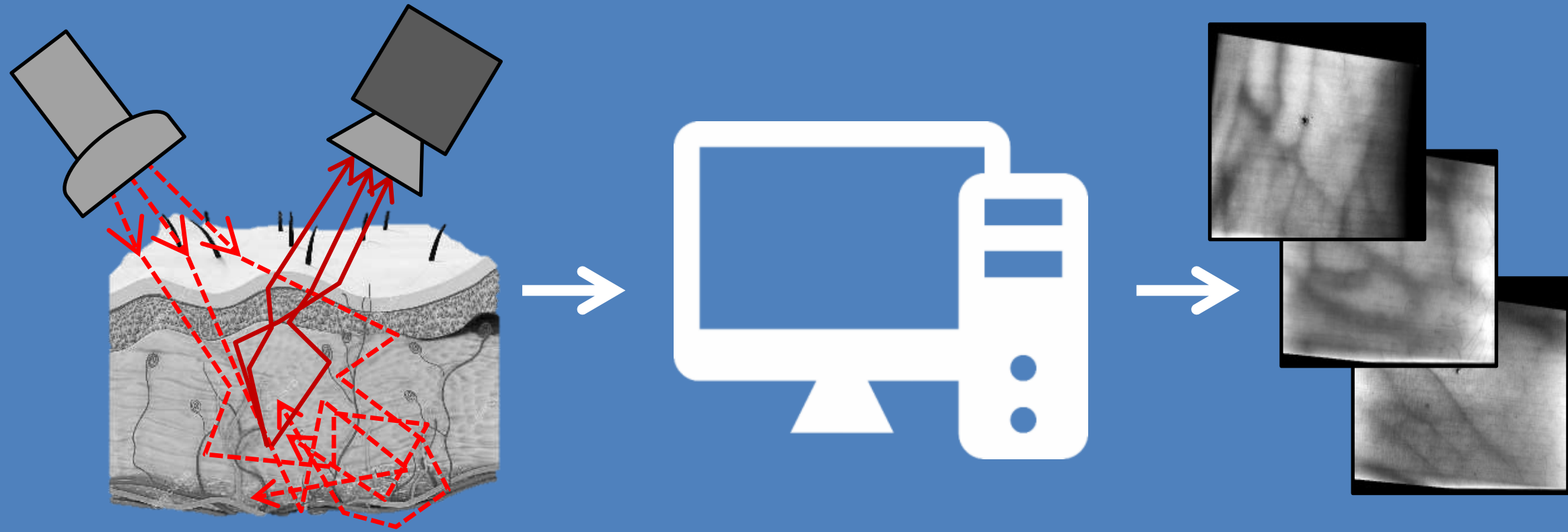
image processing



Physics-based rendering and its applications to computational imaging

Physics-based rendering and its applications to computational imaging

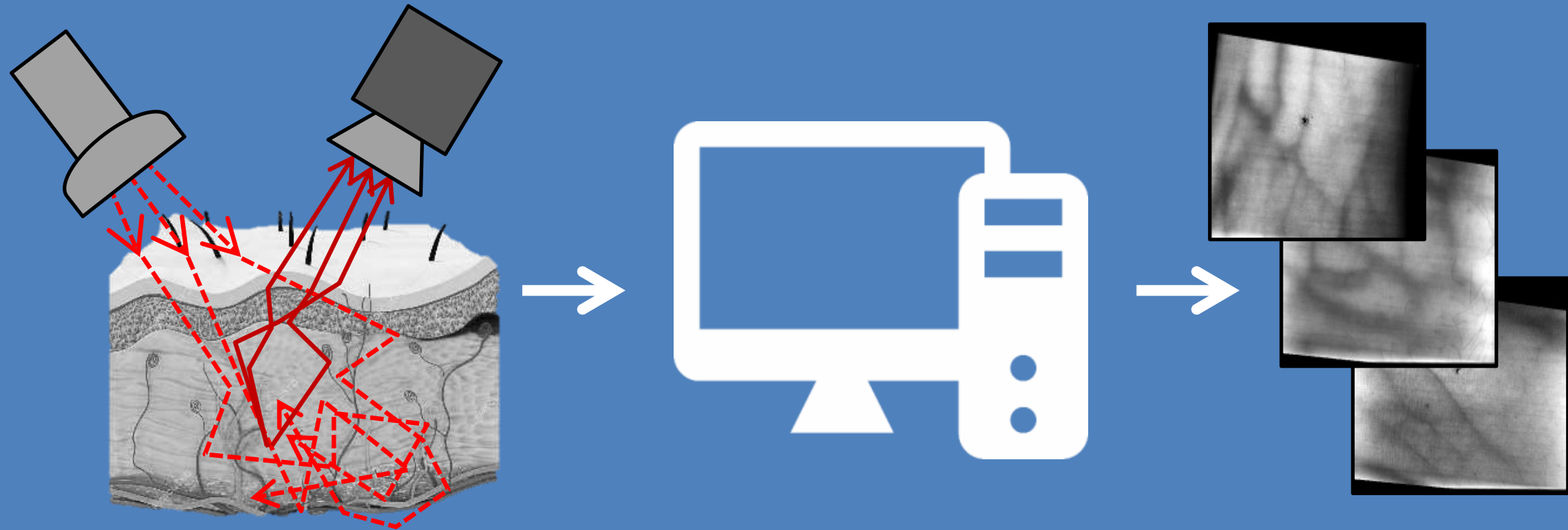
forward rendering



- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

Physics-based rendering and its applications to computational imaging

forward rendering



- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

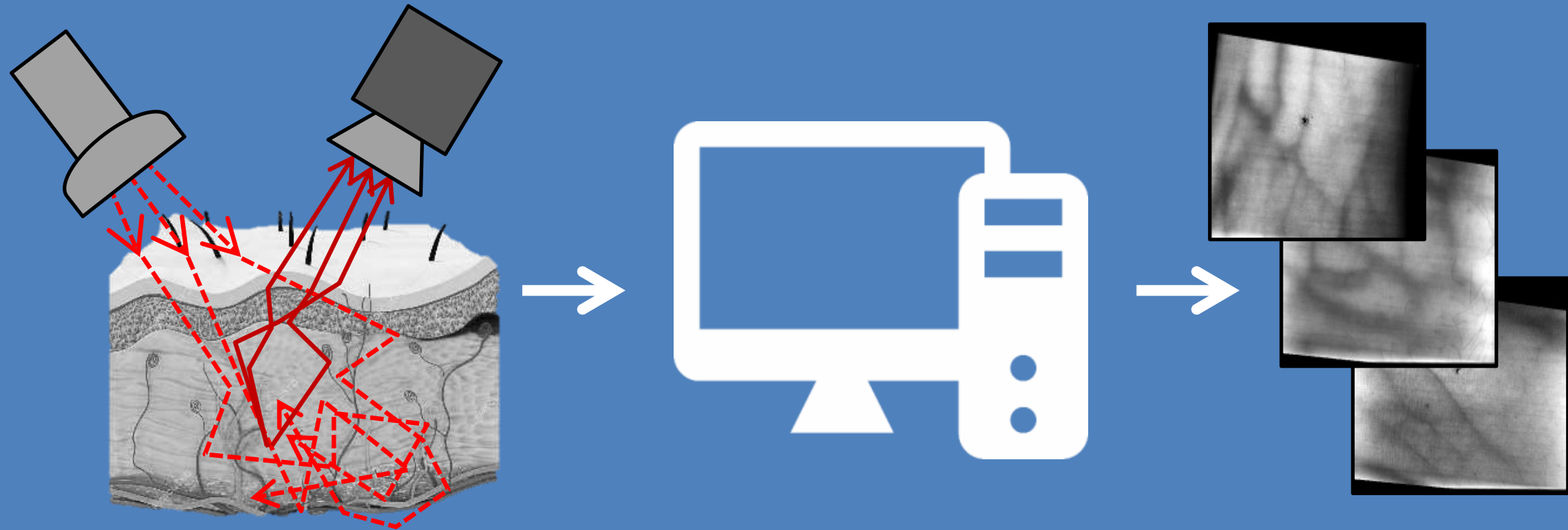
inverse rendering



- accurate and efficient differentiable simulation
- tractably solve general inverse problems

Physics-based rendering and its applications to computational imaging

forward rendering



- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

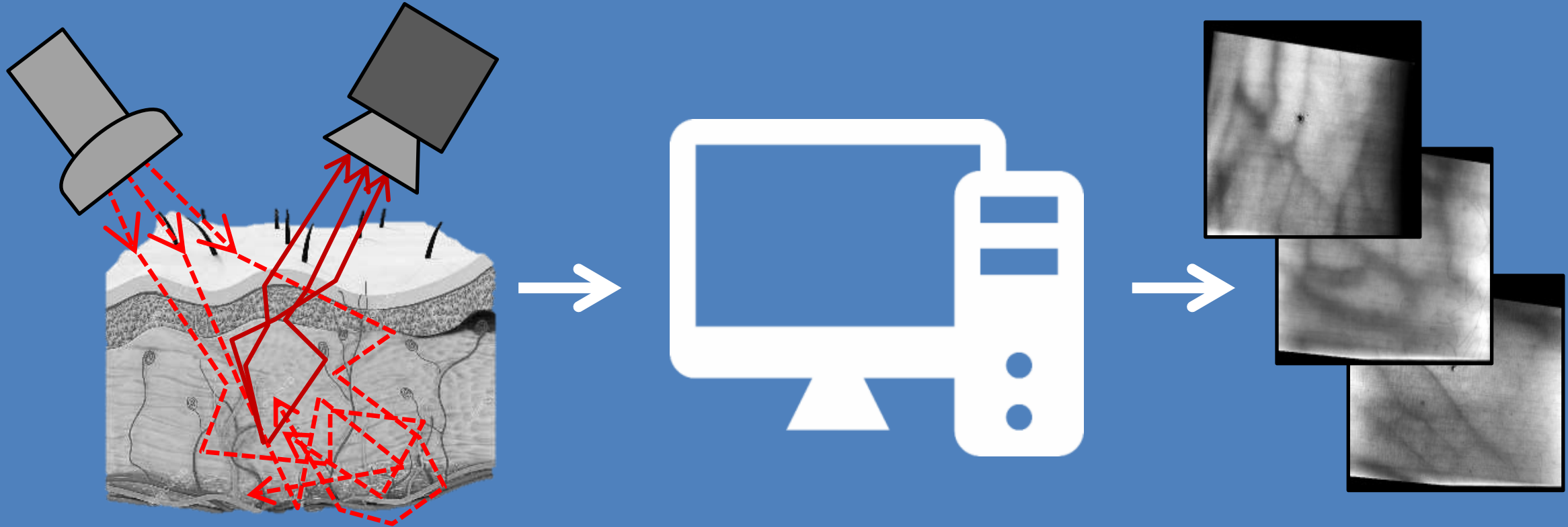
inverse rendering



- accurate and efficient differentiable simulation
- tractably solve general inverse problems

Physics-based rendering and its applications to computational imaging

forward rendering

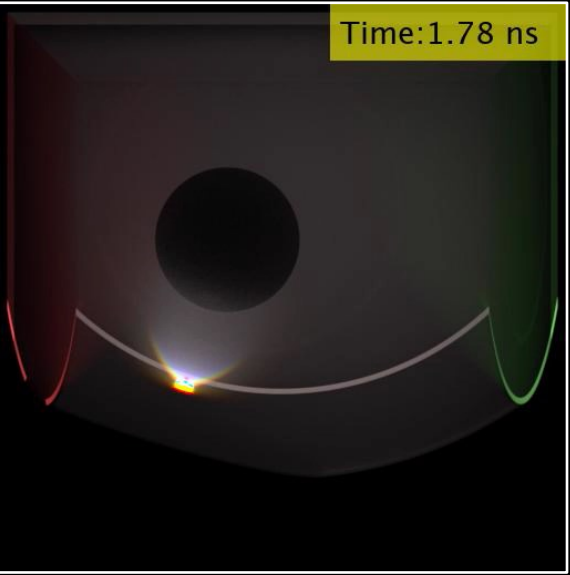


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

inverse rendering



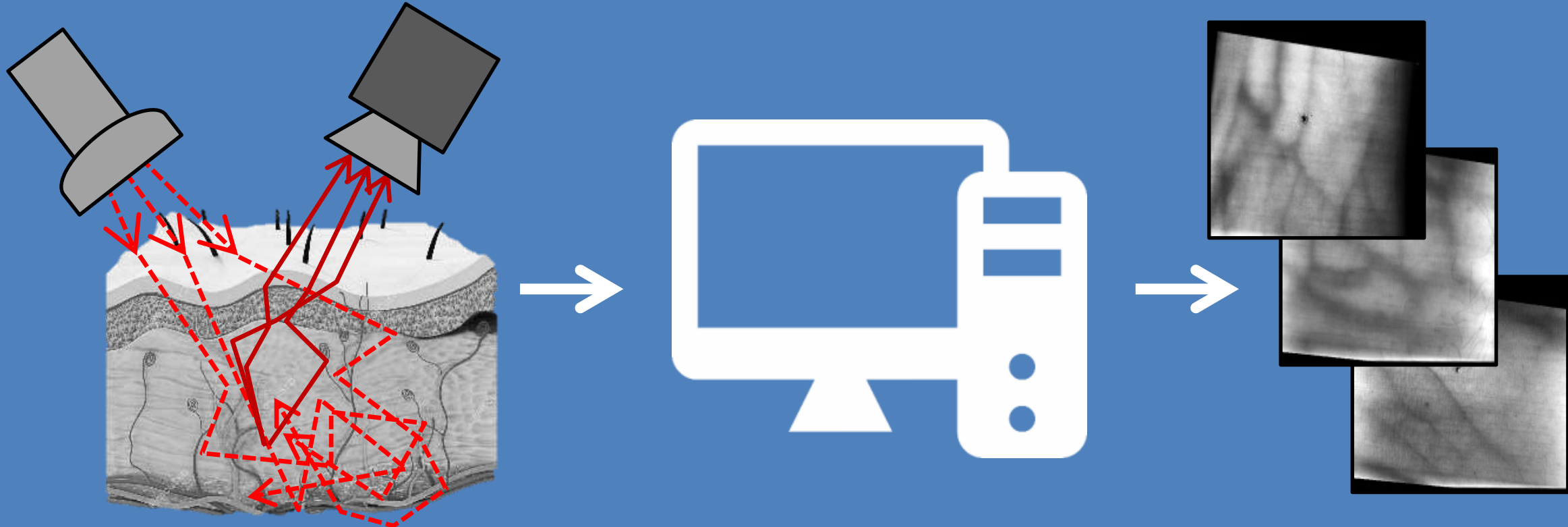
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



time-of-flight
imaging

Physics-based rendering and its applications to computational imaging

forward rendering

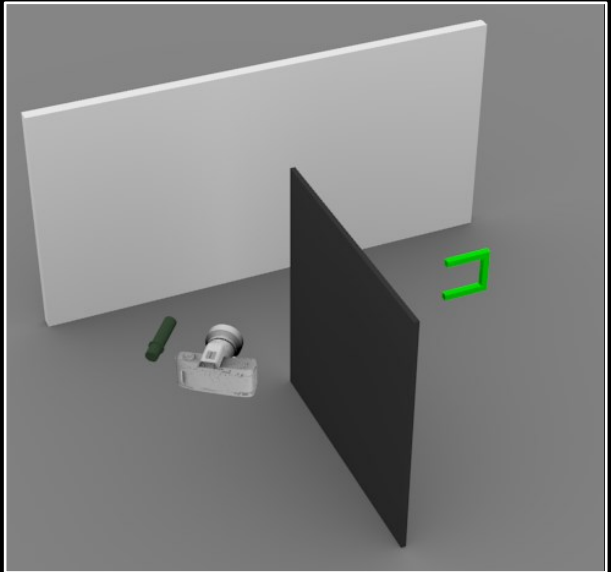
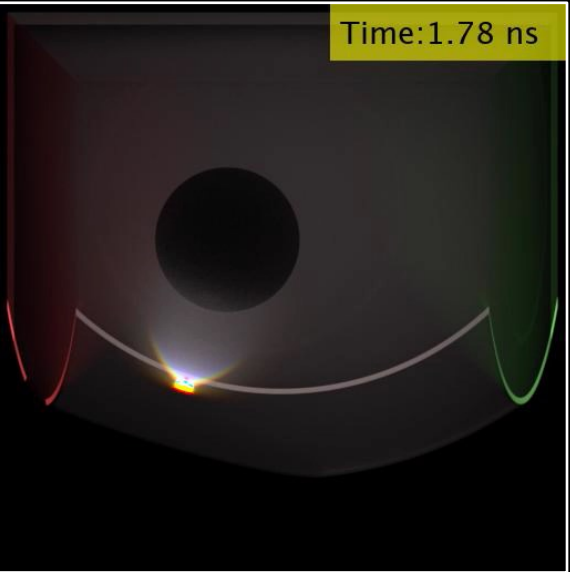


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

inverse rendering



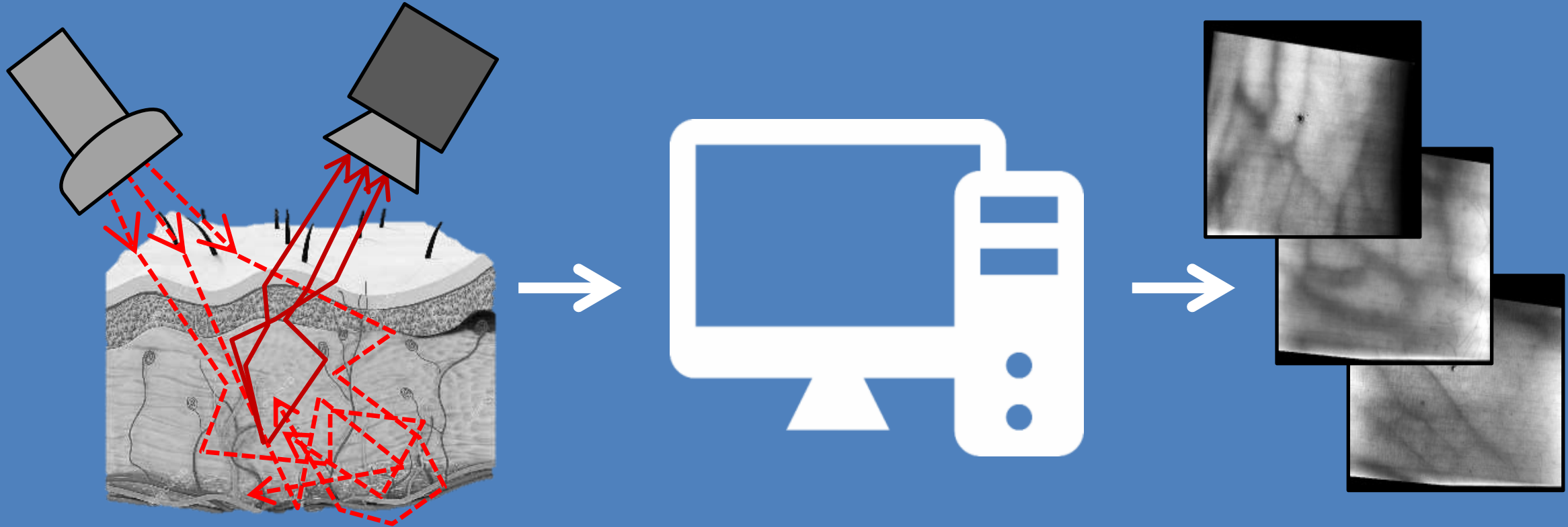
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



time-of-flight imaging non-line-of-sight imaging

Physics-based rendering and its applications to computational imaging

forward rendering

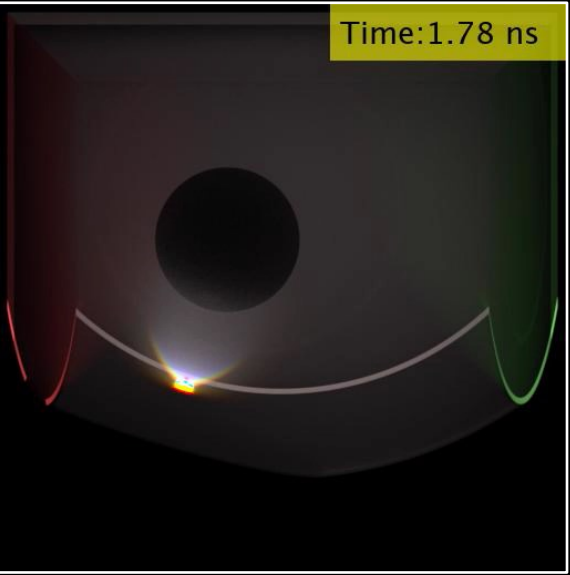


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

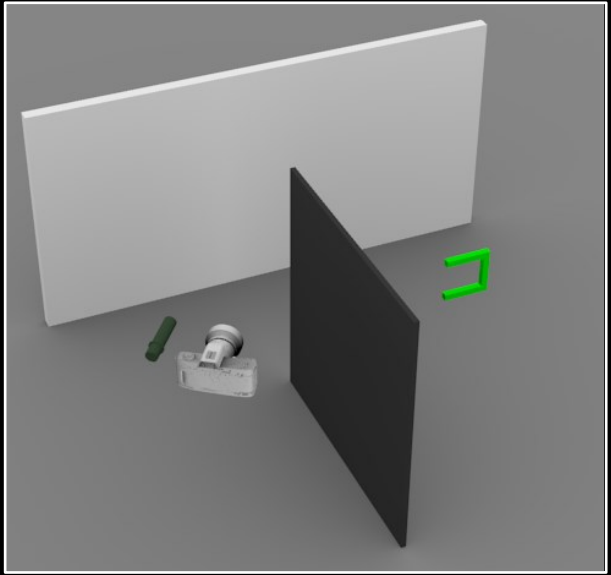
inverse rendering



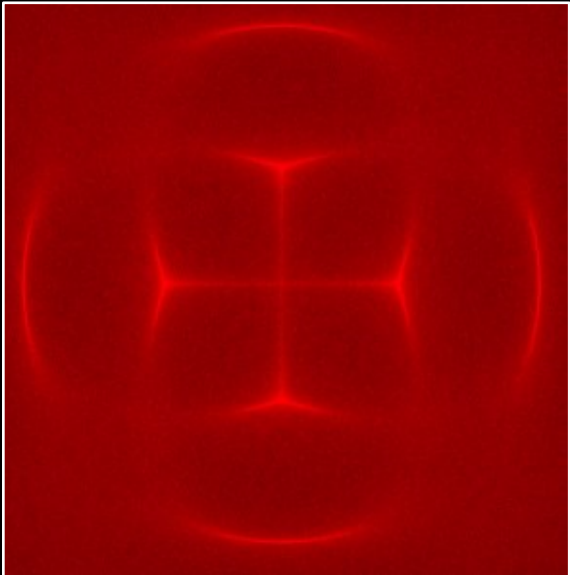
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



time-of-flight
imaging



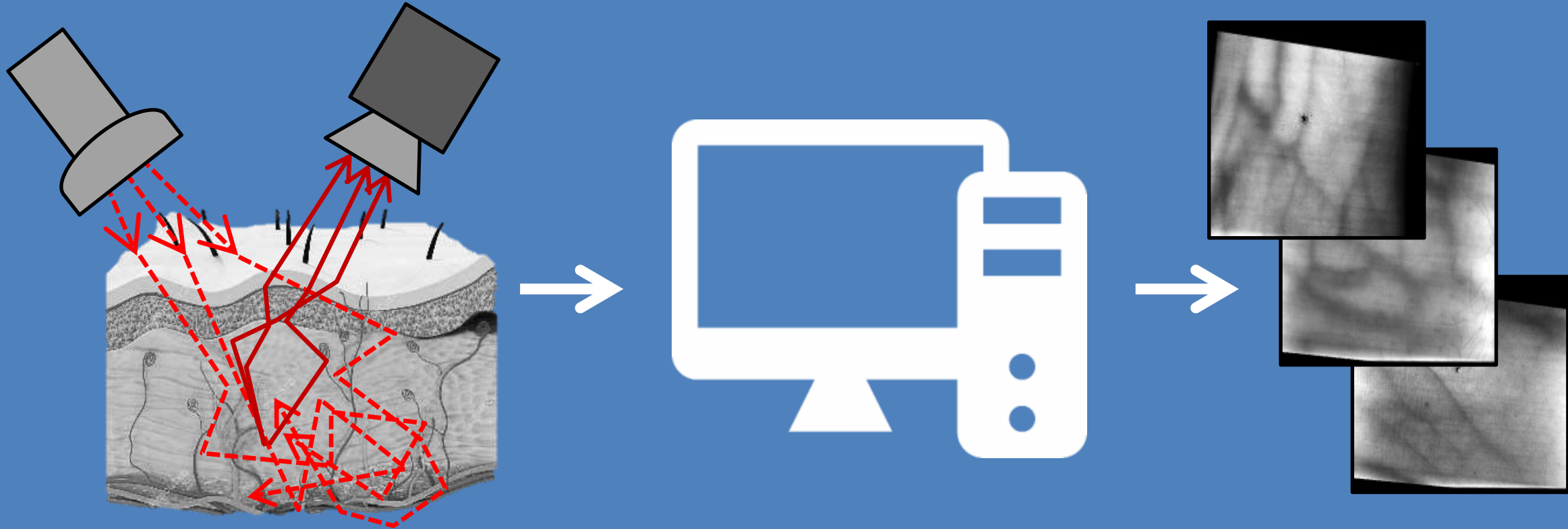
non-line-of-sight
imaging



acousto-optic
lensing

Physics-based rendering and its applications to computational imaging

forward rendering

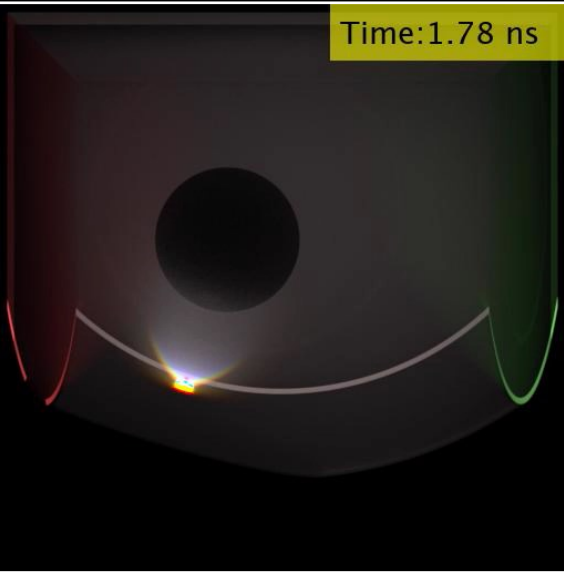


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

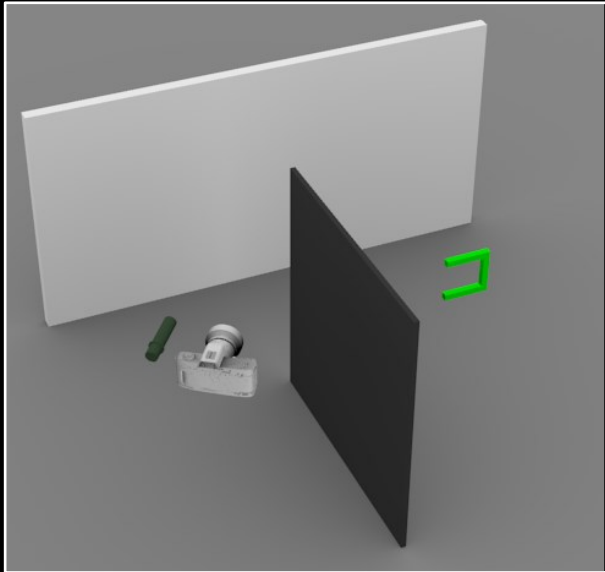
inverse rendering



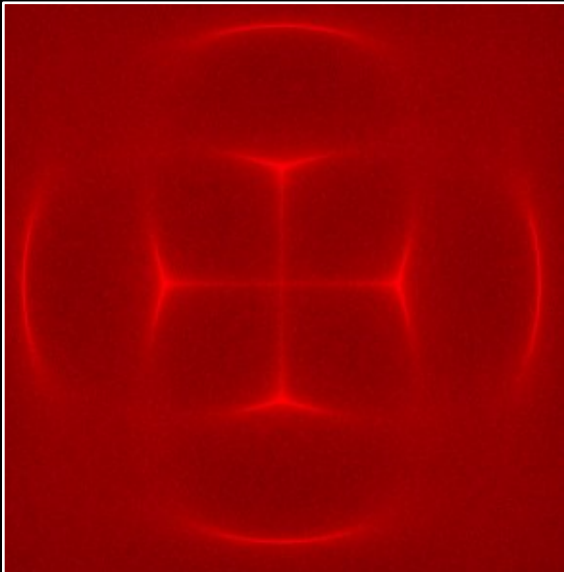
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



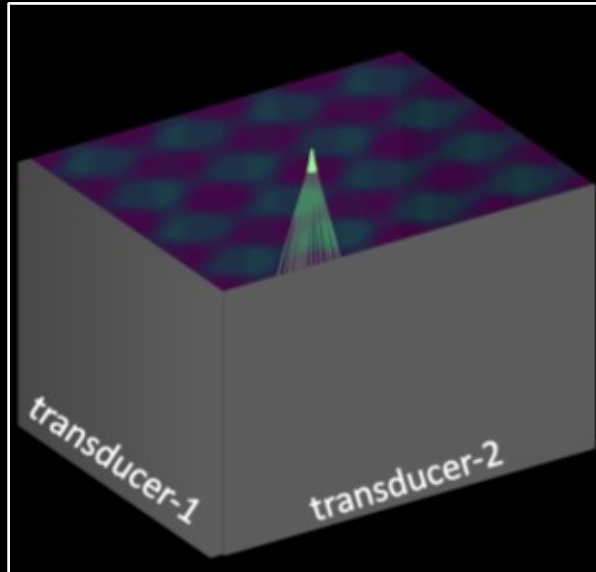
time-of-flight imaging



non-line-of-sight imaging



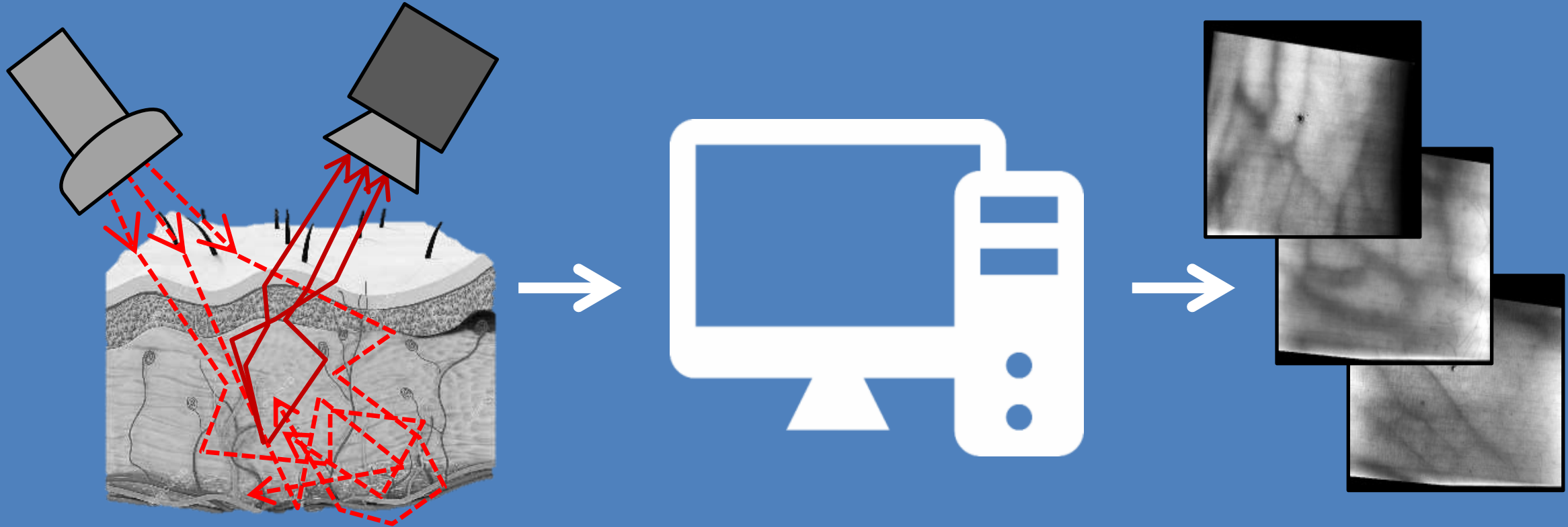
acousto-optic lensing



ultrafast light scanning

Physics-based rendering and its applications to computational imaging

forward rendering

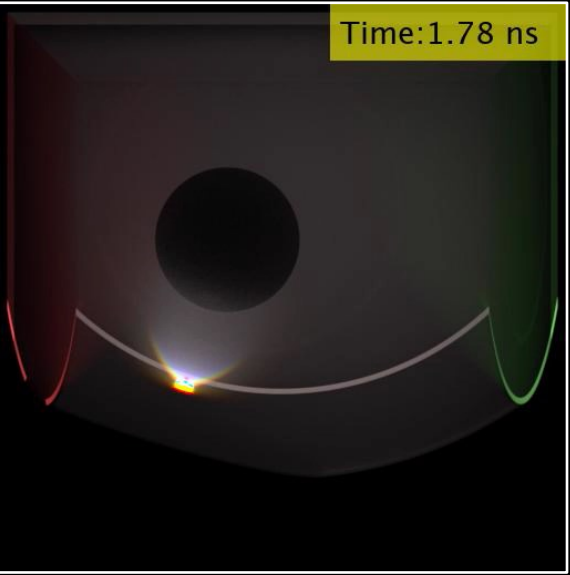


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

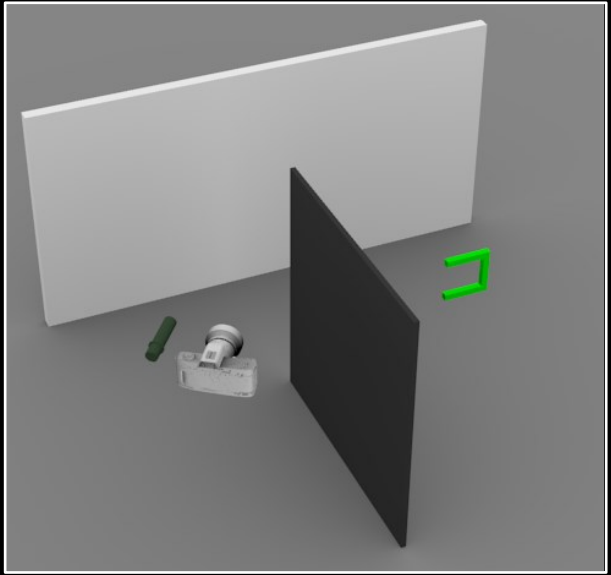
inverse rendering



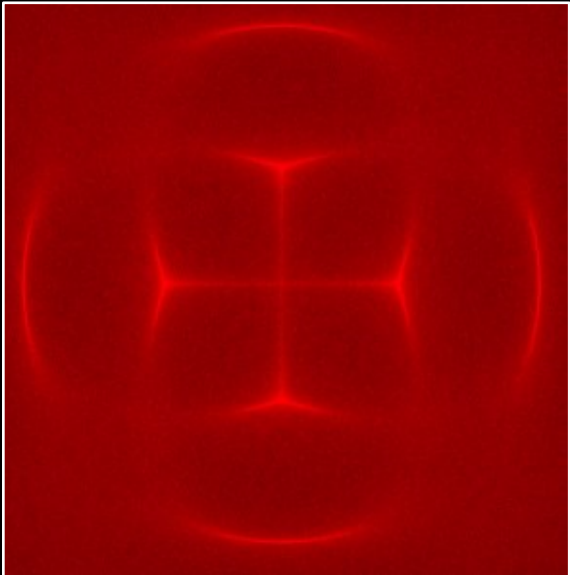
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



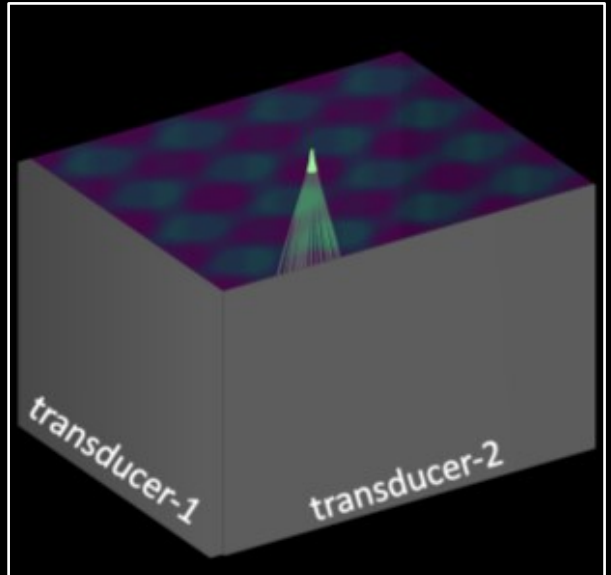
time-of-flight imaging



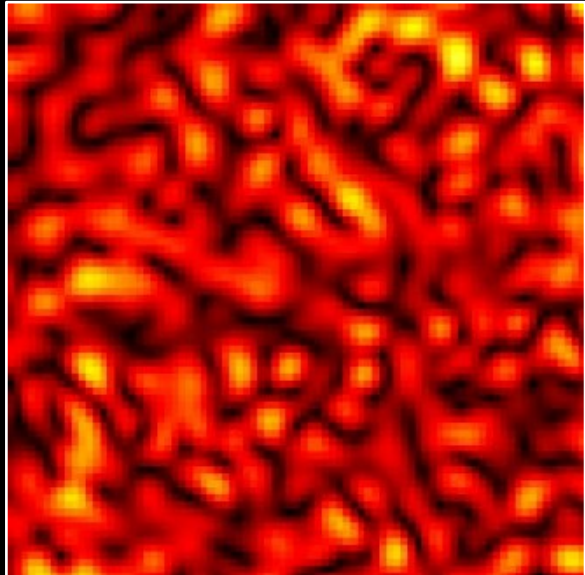
non-line-of-sight imaging



acousto-optic lensing



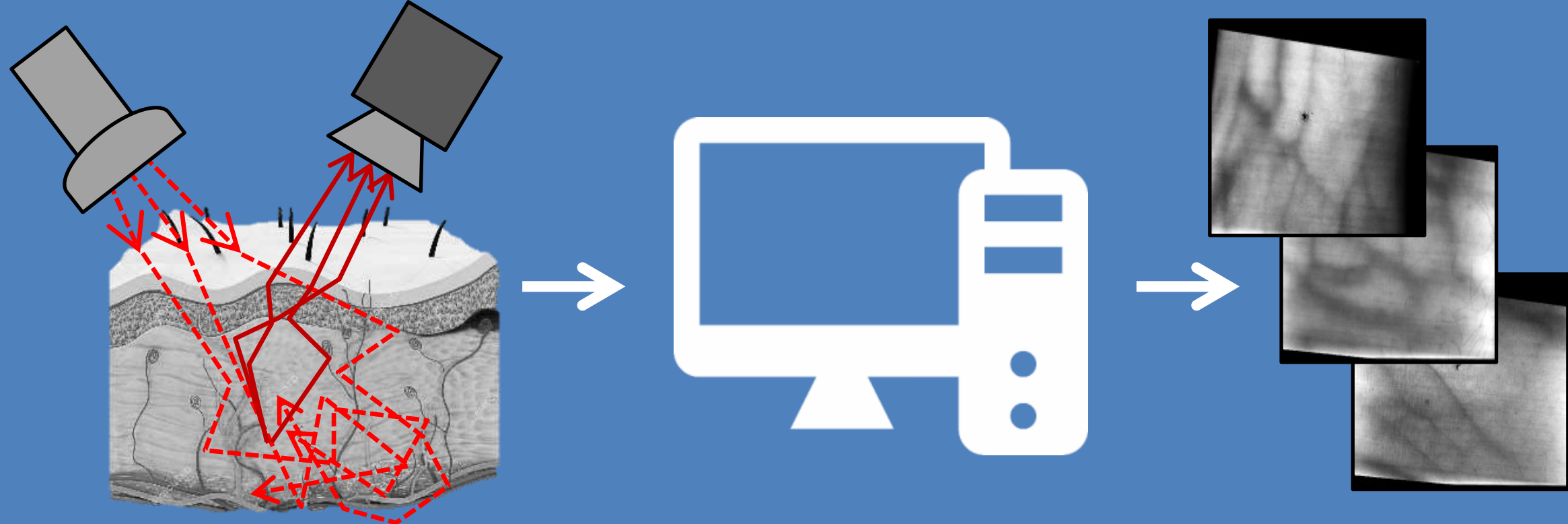
ultrafast light scanning



speckle imaging

Physics-based rendering and its applications to computational imaging

forward rendering

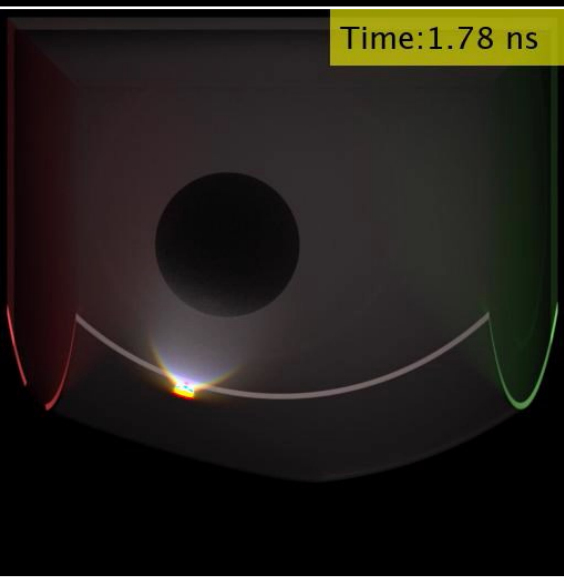


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

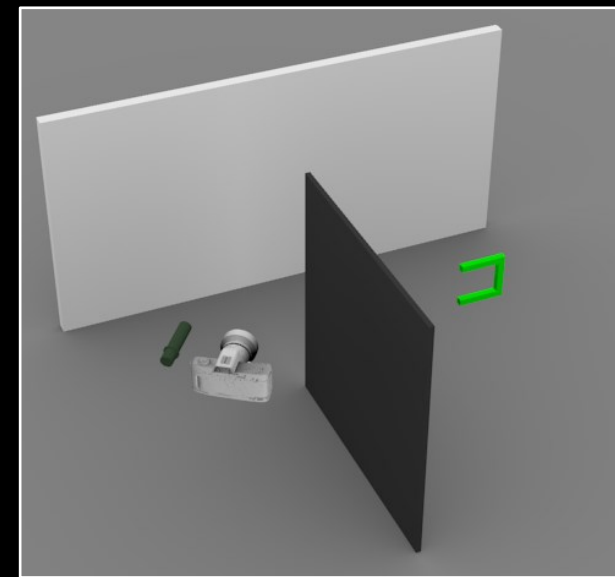
inverse rendering



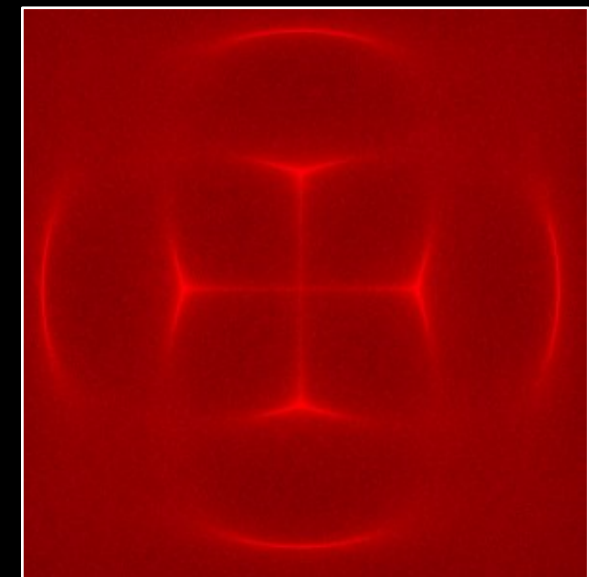
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



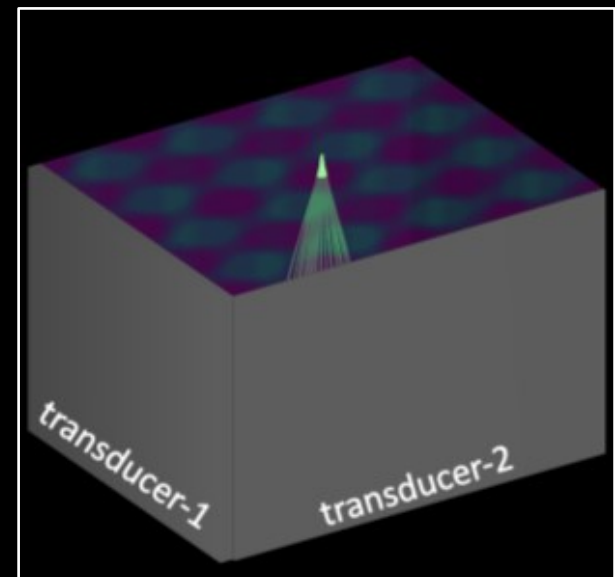
time-of-flight imaging



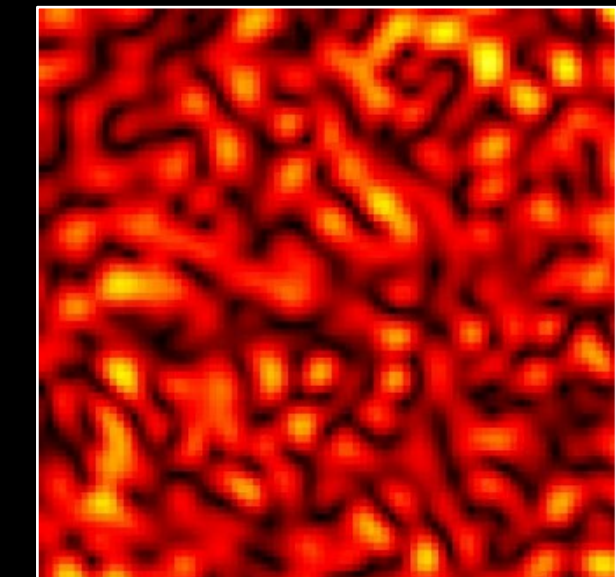
non-line-of-sight imaging



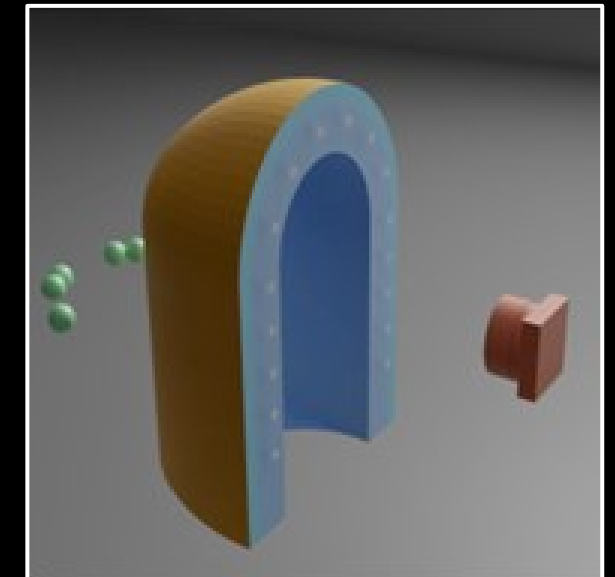
acousto-optic lensing



ultrafast light scanning



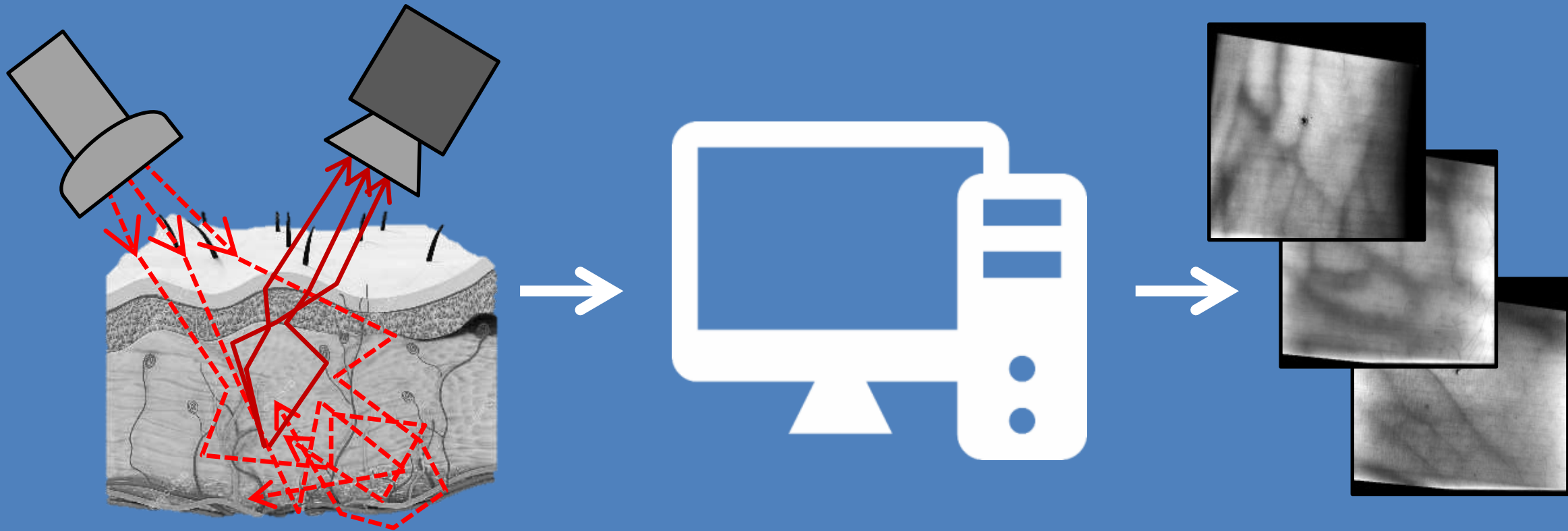
speckle imaging



tactile sensor design

Physics-based rendering and its applications to computational imaging

forward rendering

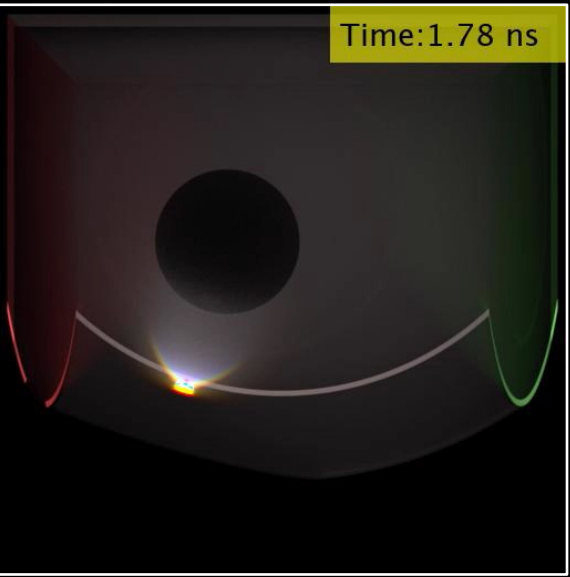


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

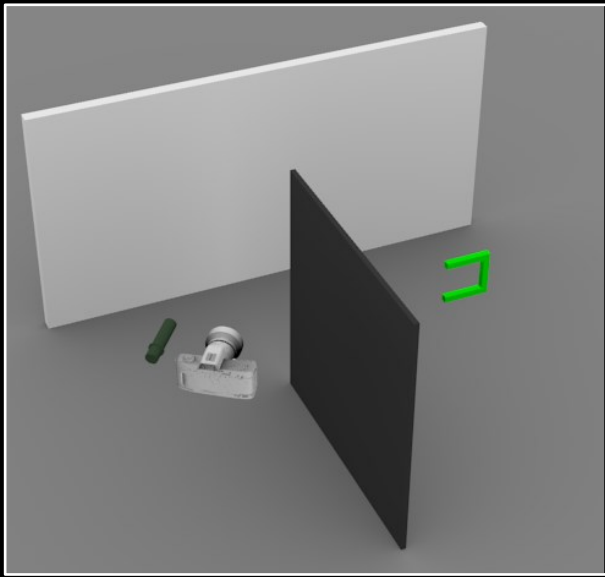
inverse rendering



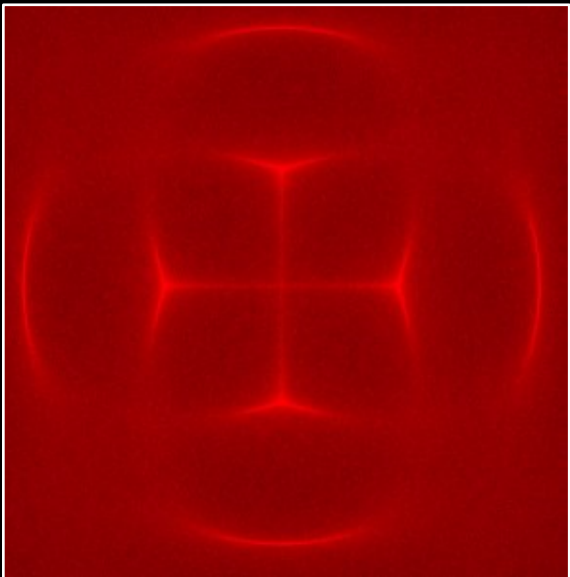
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



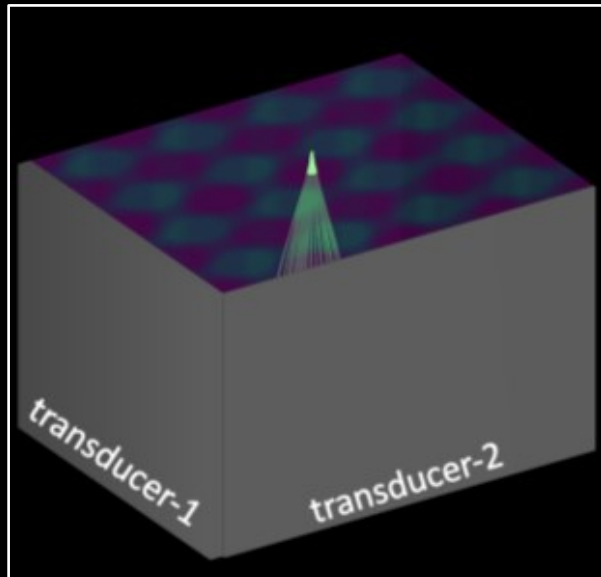
time-of-flight imaging



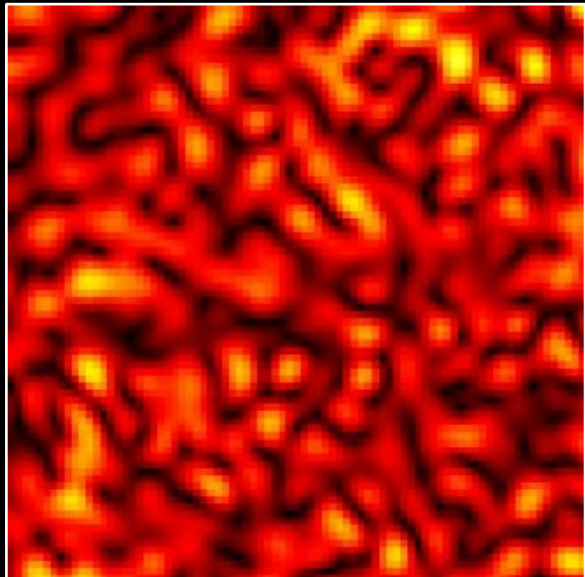
non-line-of-sight imaging



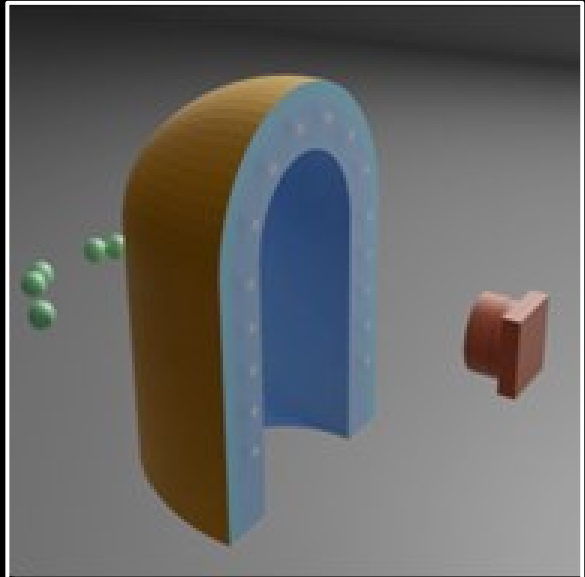
acousto-optic lensing



ultrafast light scanning



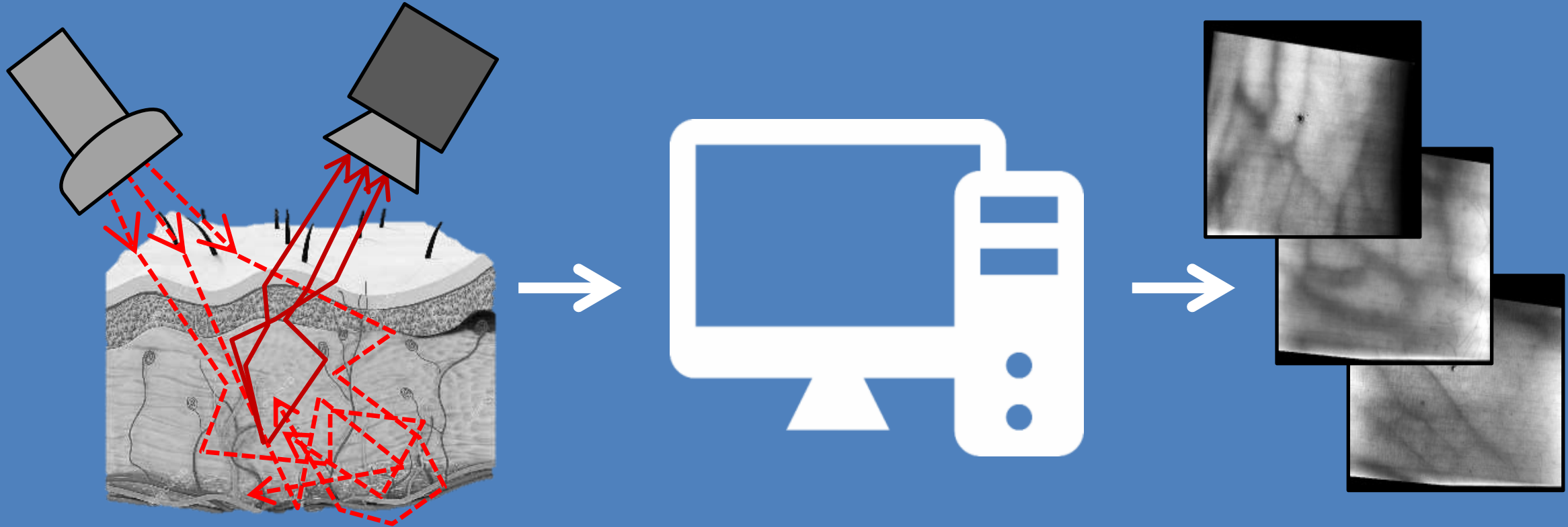
speckle imaging



tactile sensor design

Physics-based rendering and its applications to computational imaging

forward rendering

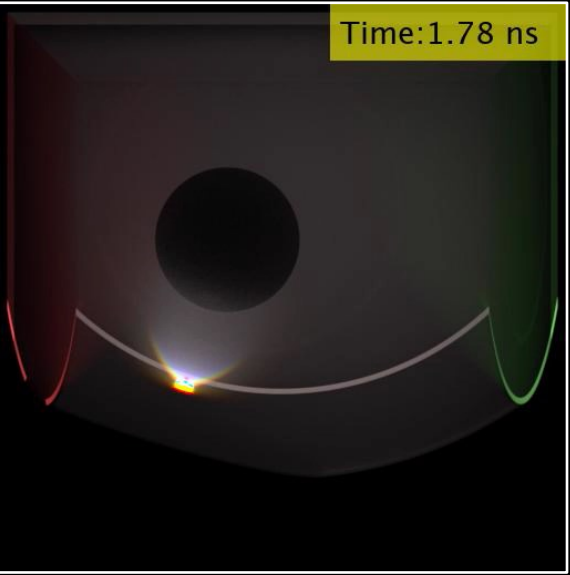


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

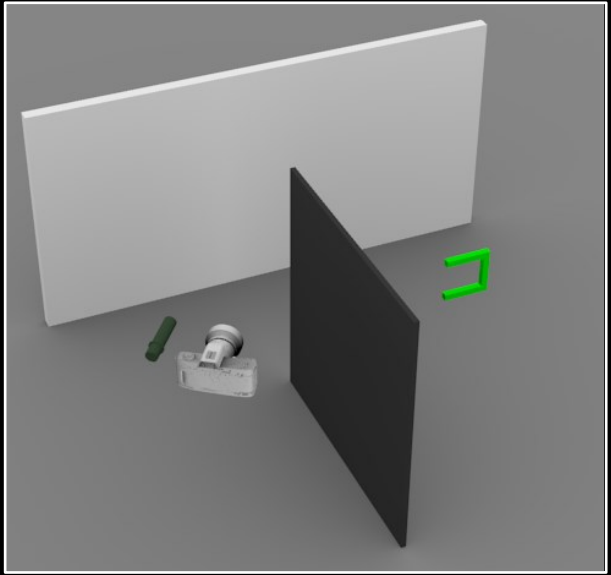
inverse rendering



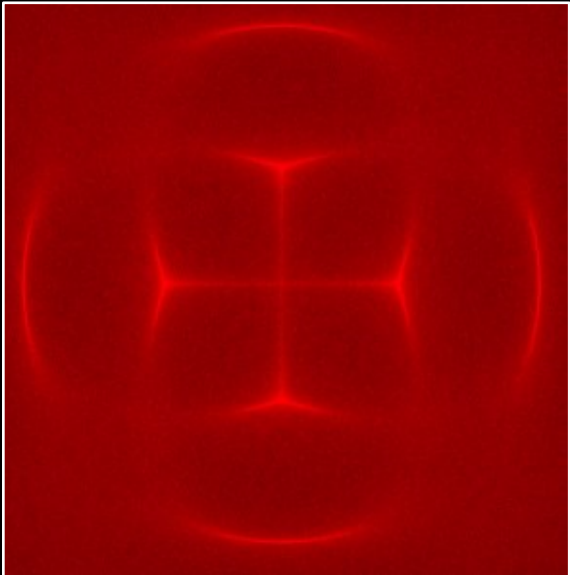
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



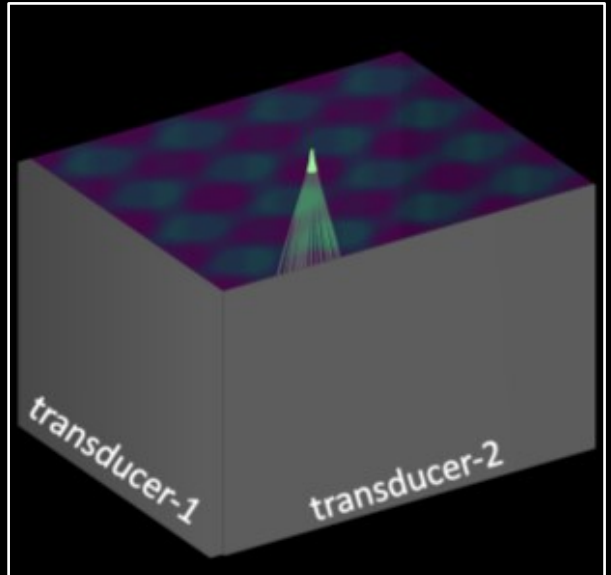
time-of-flight imaging



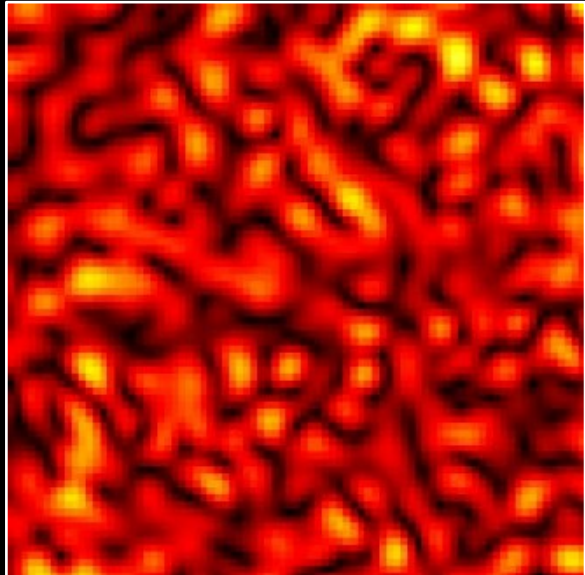
non-line-of-sight imaging



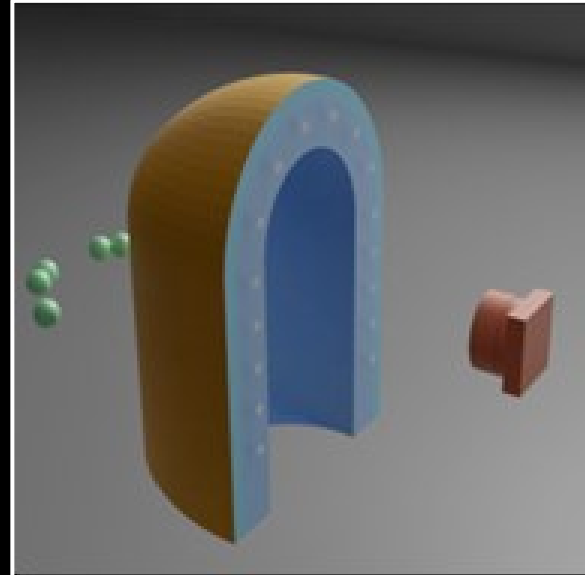
acousto-optic lensing



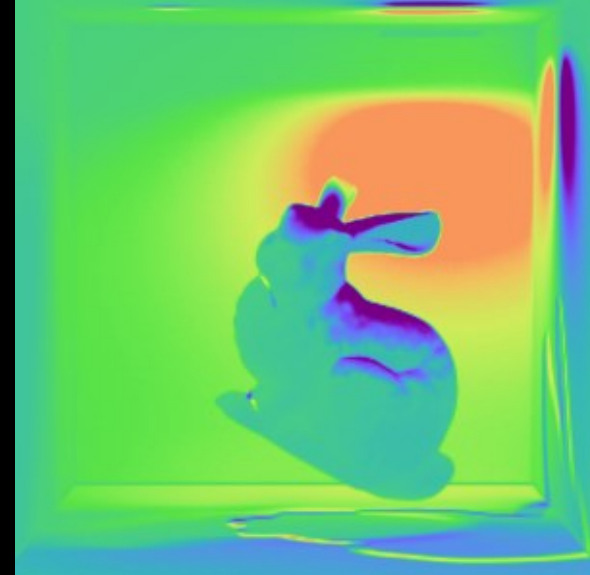
ultrafast light scanning



speckle imaging



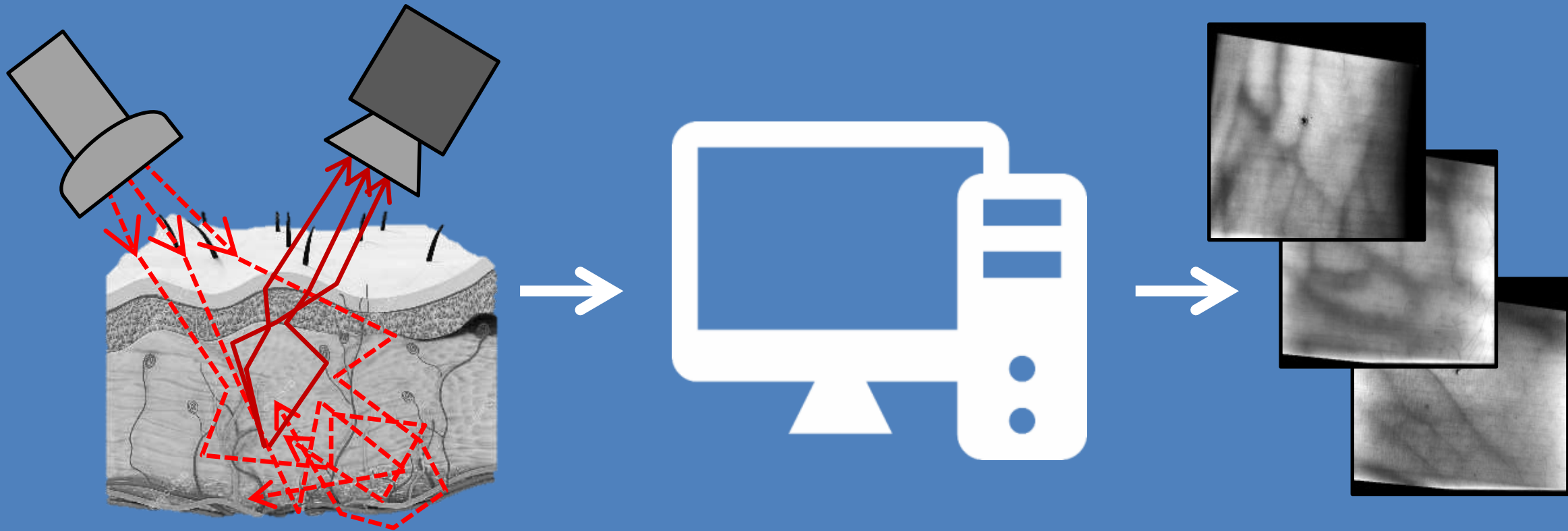
tactile sensor design



differentiable rendering

Physics-based rendering and its applications to computational imaging

forward rendering

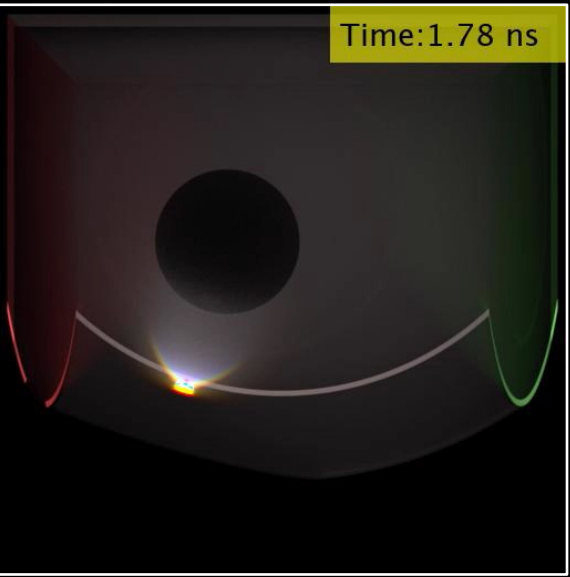


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

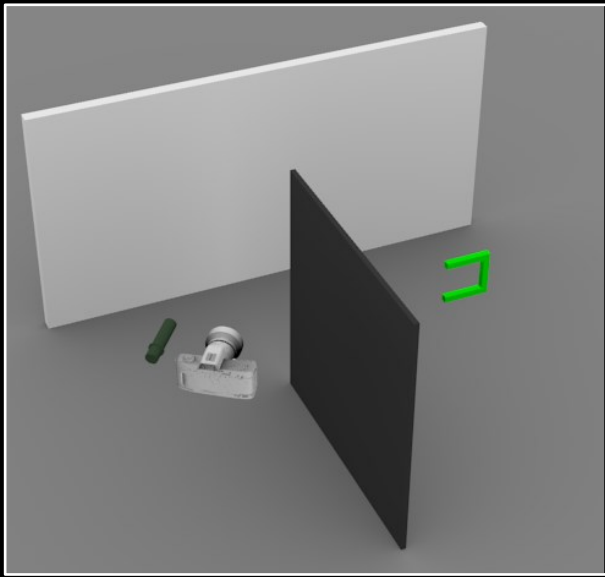
inverse rendering



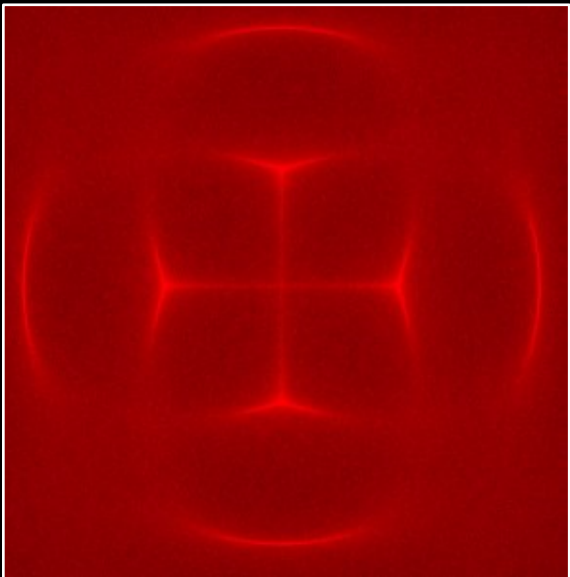
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



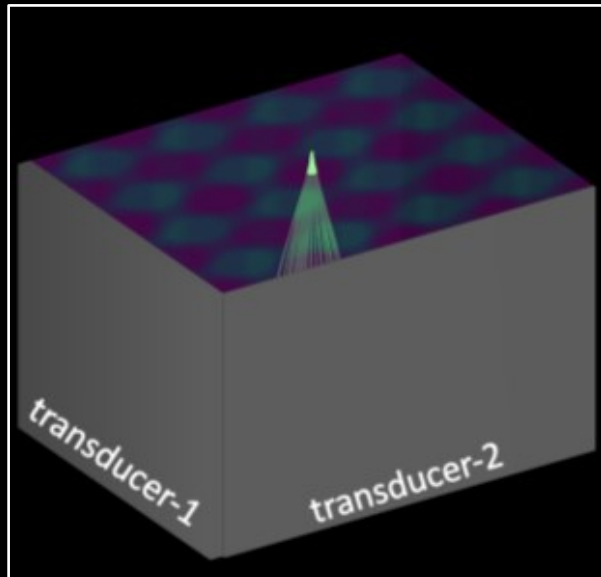
time-of-flight imaging



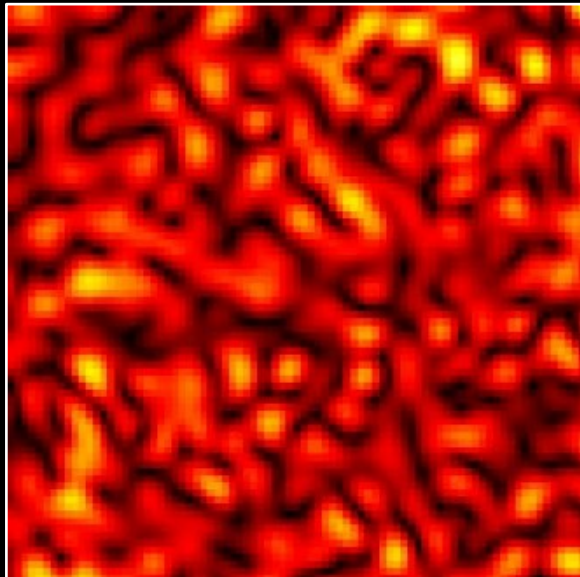
non-line-of-sight imaging



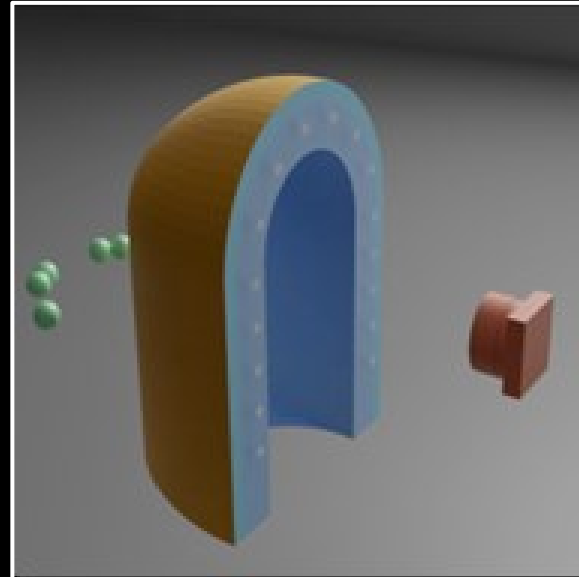
acousto-optic lensing



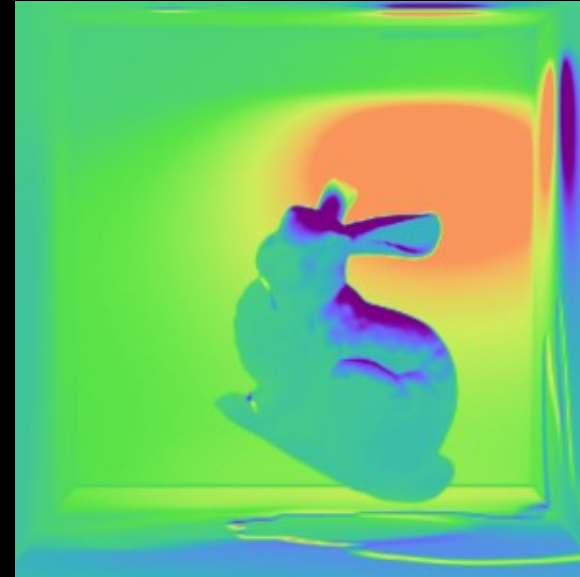
ultrafast light scanning



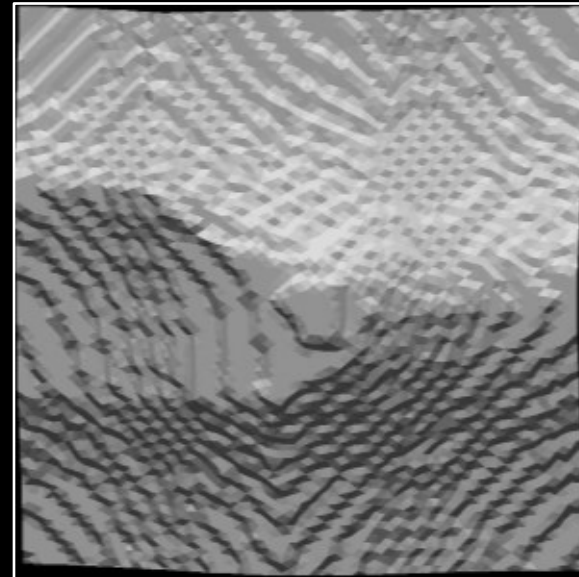
speckle imaging



tactile sensor design



differentiable rendering



inverse problems

Complex light transport



After [Ritschel et al 2011]

Complex light transport



Complex light transport



Complex light transport



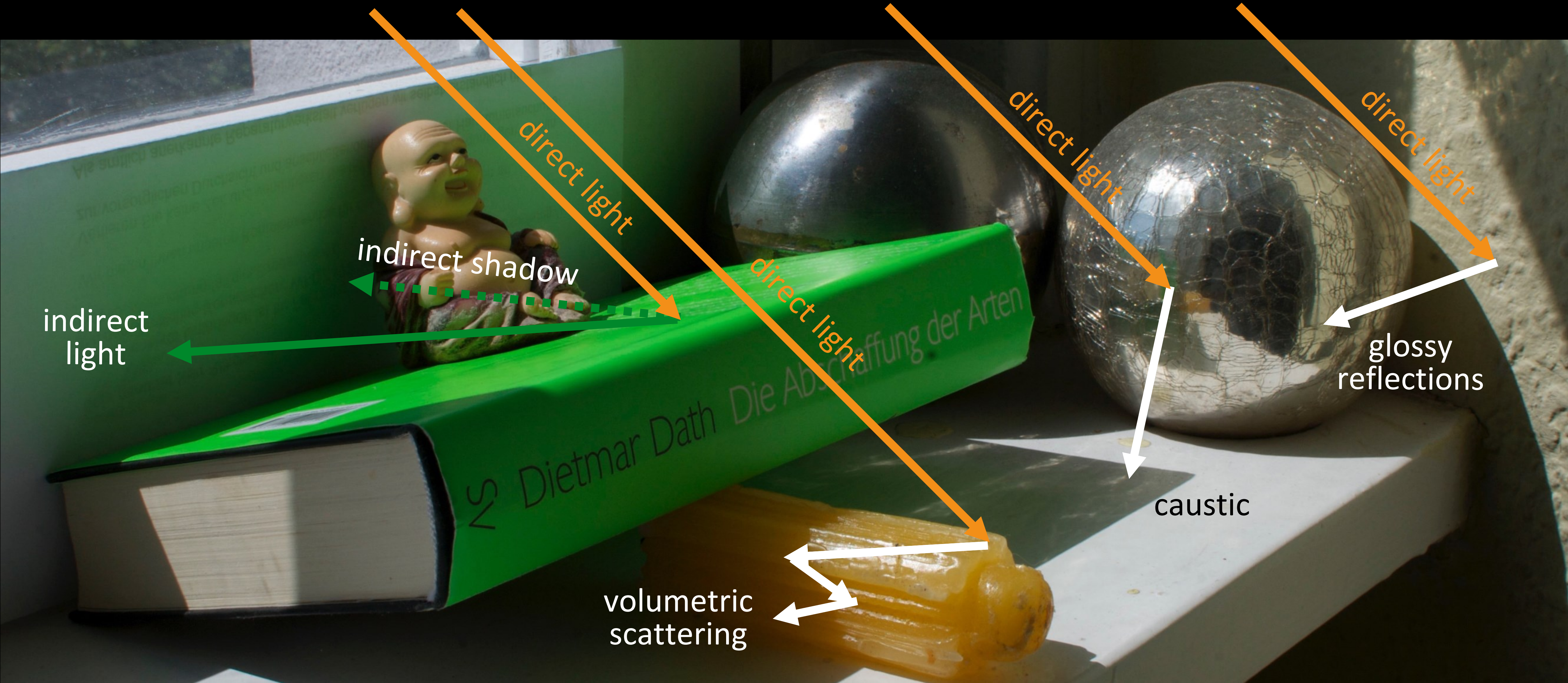
Complex light transport



Complex light transport

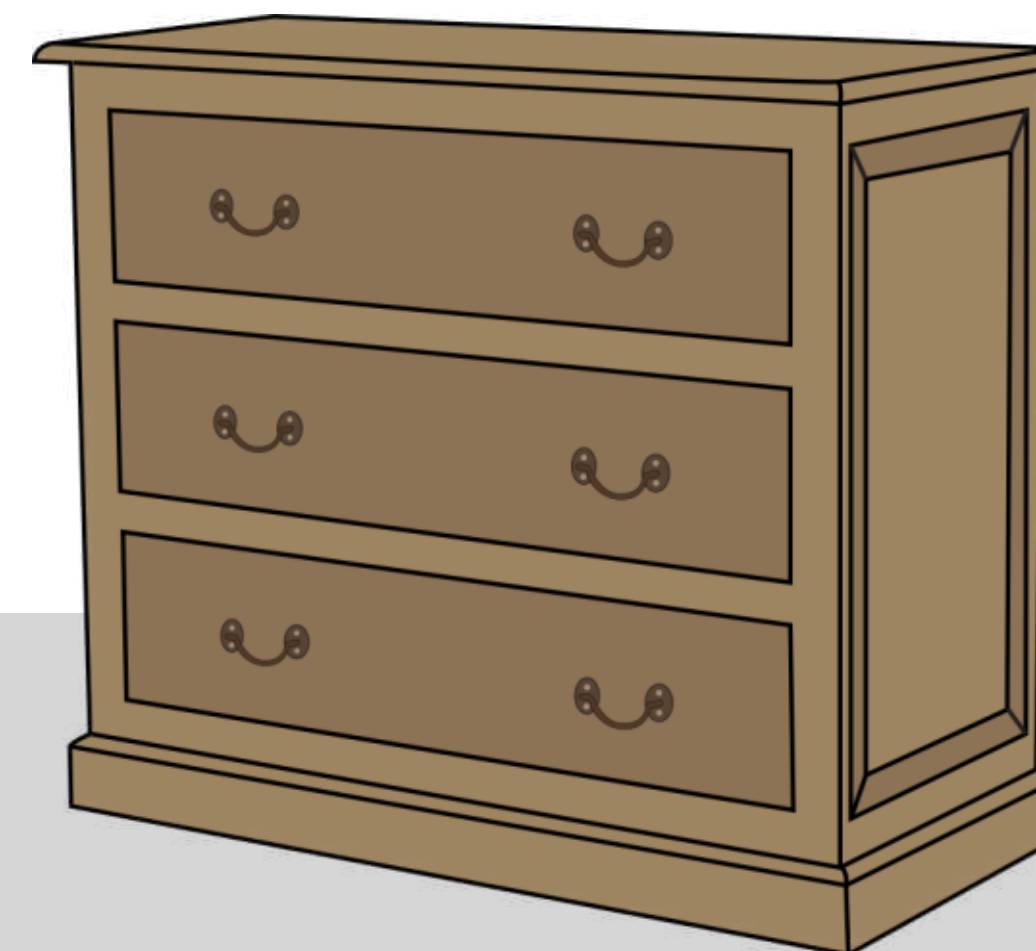


Complex light transport



Path integral form of light transport

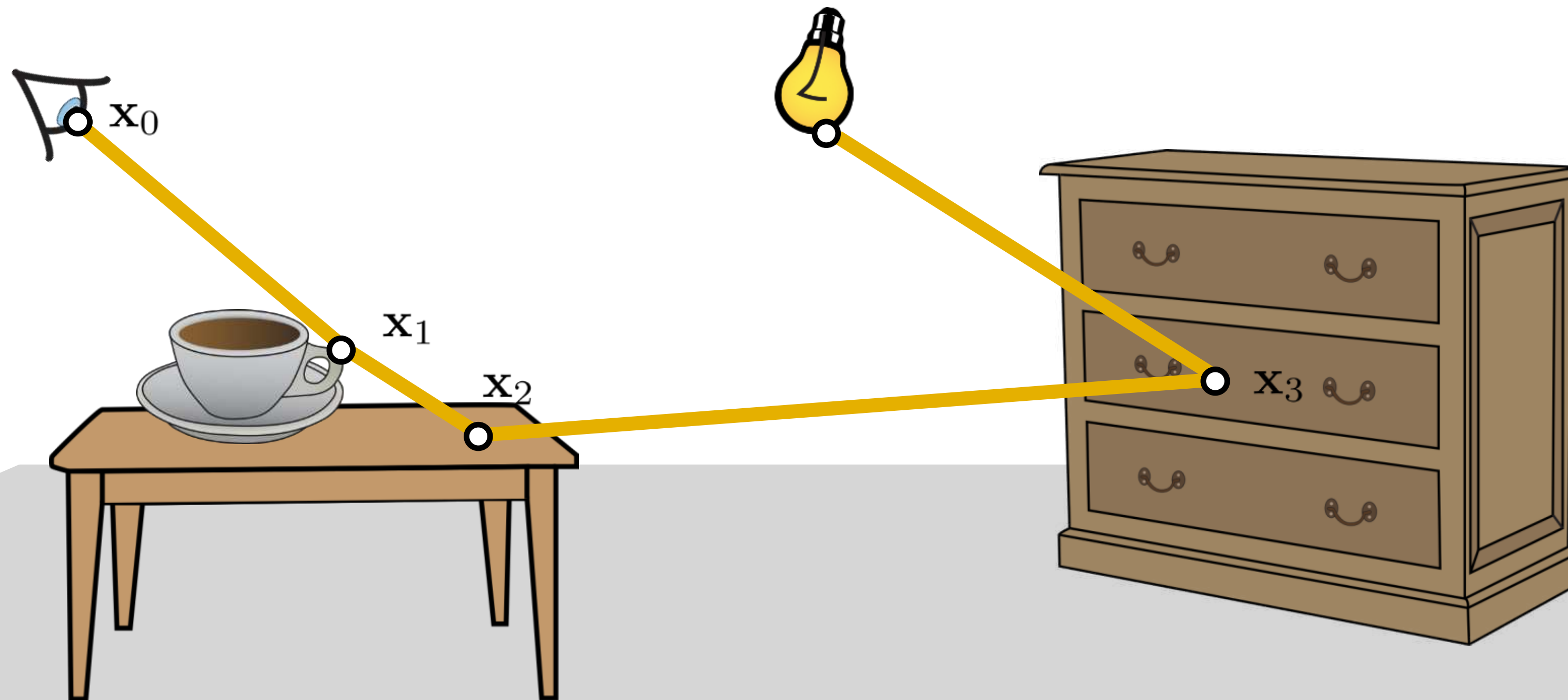
image $I = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}) d\bar{\mathbf{x}}$



Path integral form of light transport

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

space of all light paths



Path integral form of light transport

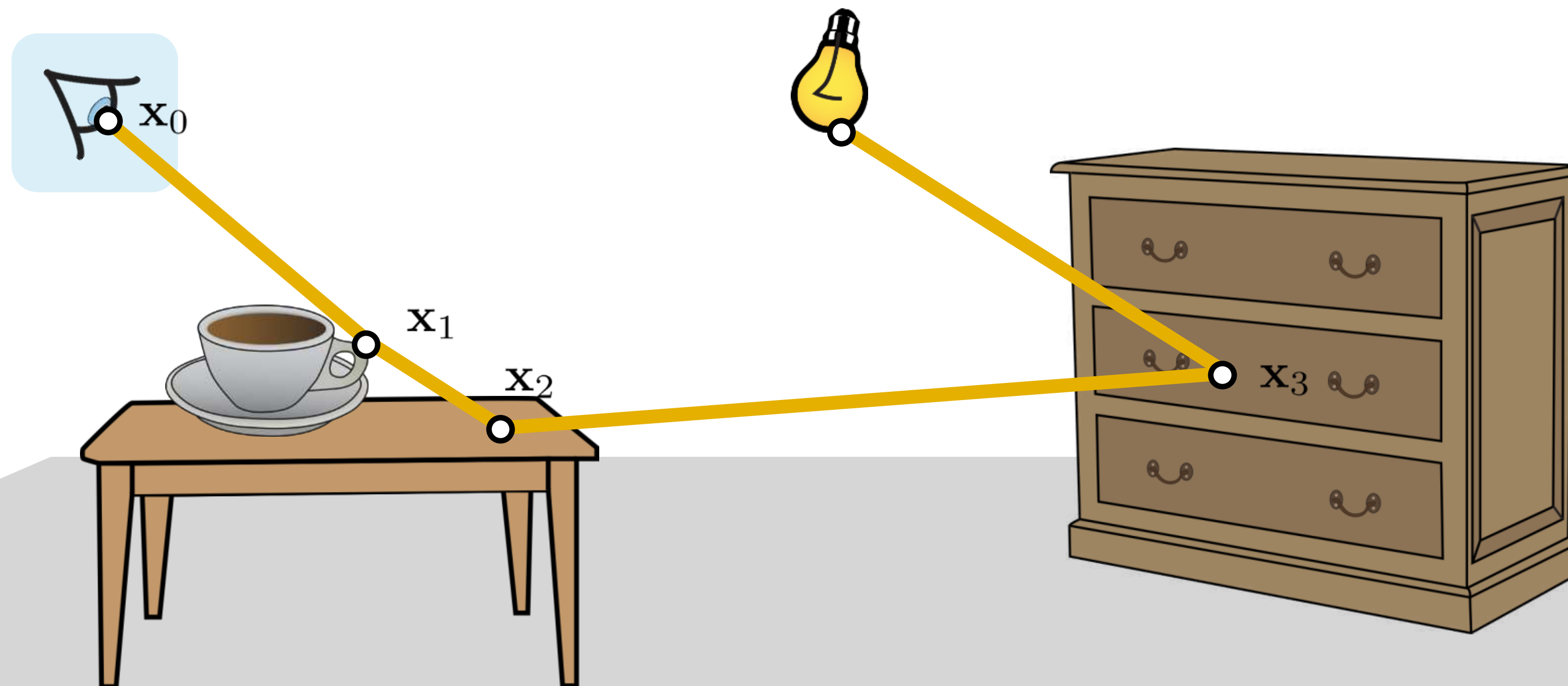
image

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

space of all light paths

sensor weight

light path



Path integral form of light transport

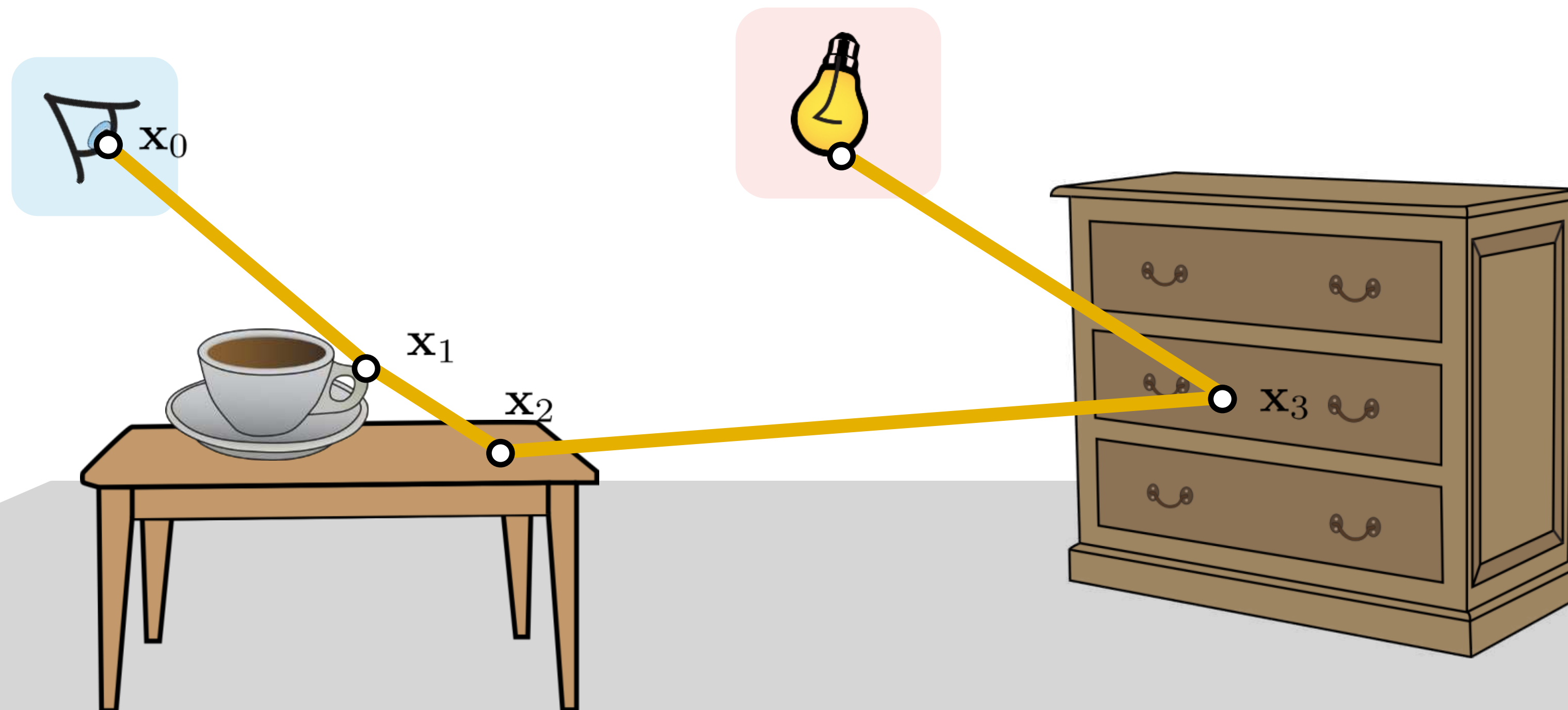
image

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

space of all light paths

sensor weight source weight

light path



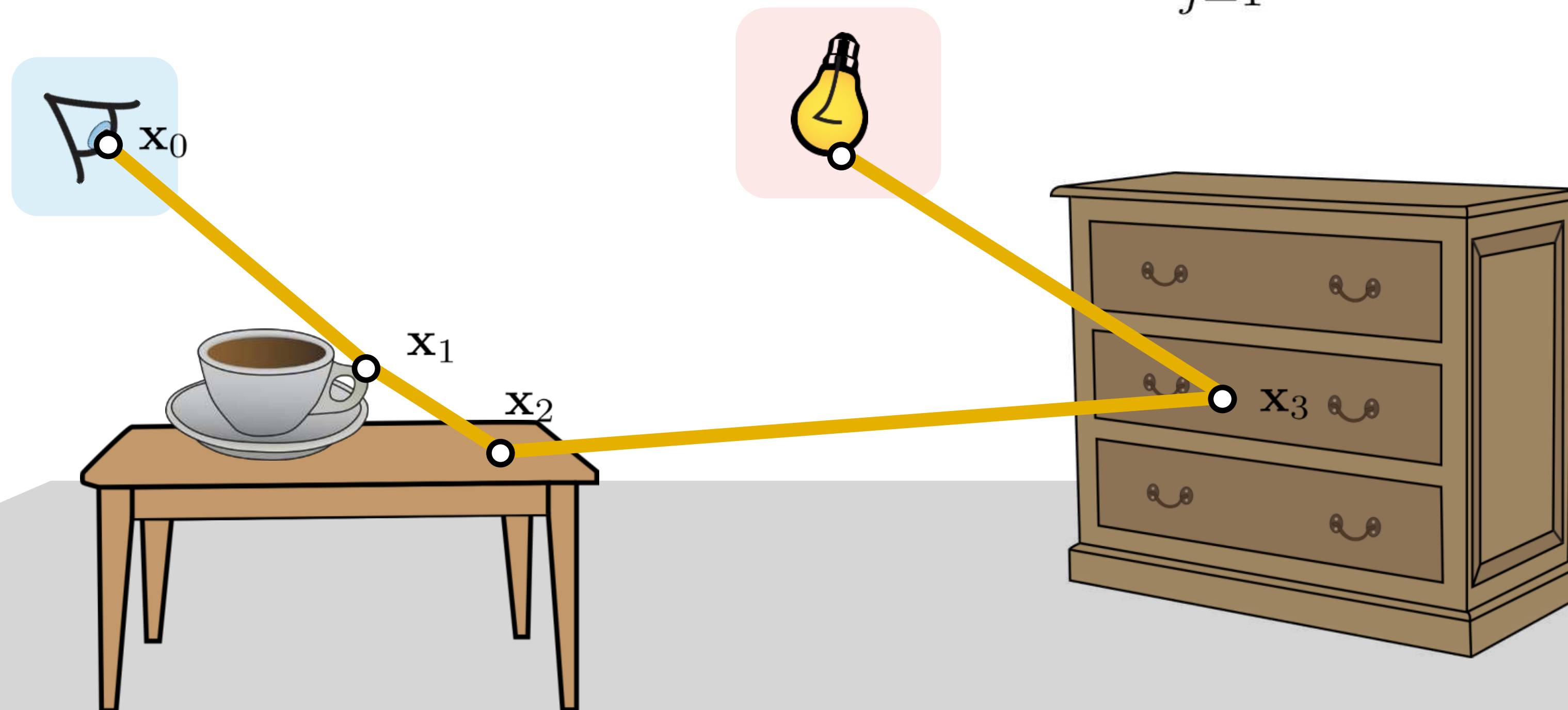
Path integral form of light transport

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

sensor weight source weight
path throughput

space of all light paths

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



Path integral form of light transport

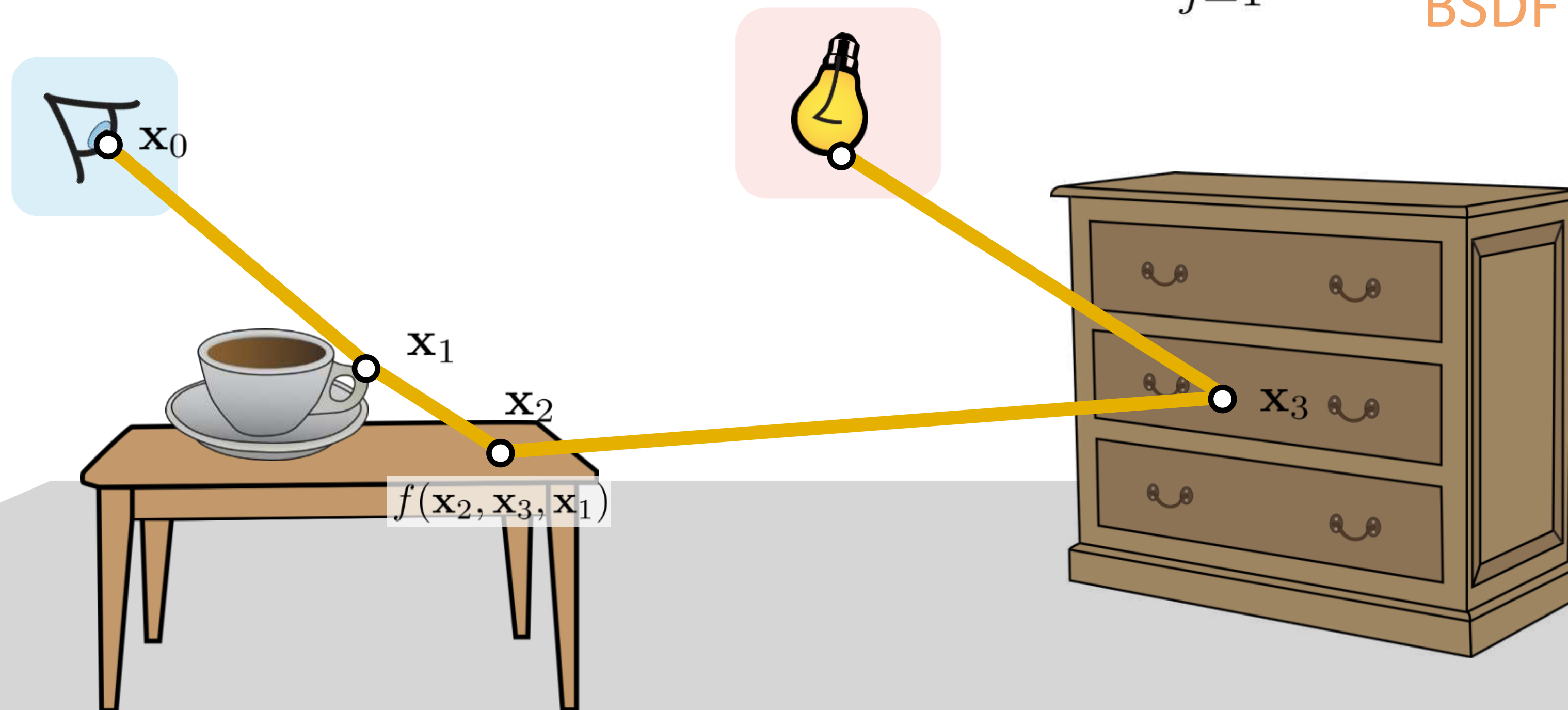
image $I = \int_{\mathcal{P}} W_e(\mathbf{x}_0, \mathbf{x}_1) L_e(\mathbf{x}_k, \mathbf{x}_{k-1}) T(\bar{\mathbf{x}}) d\bar{\mathbf{x}}$

sensor weight
source weight
path throughput
light path

space of all light paths

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

BSDF



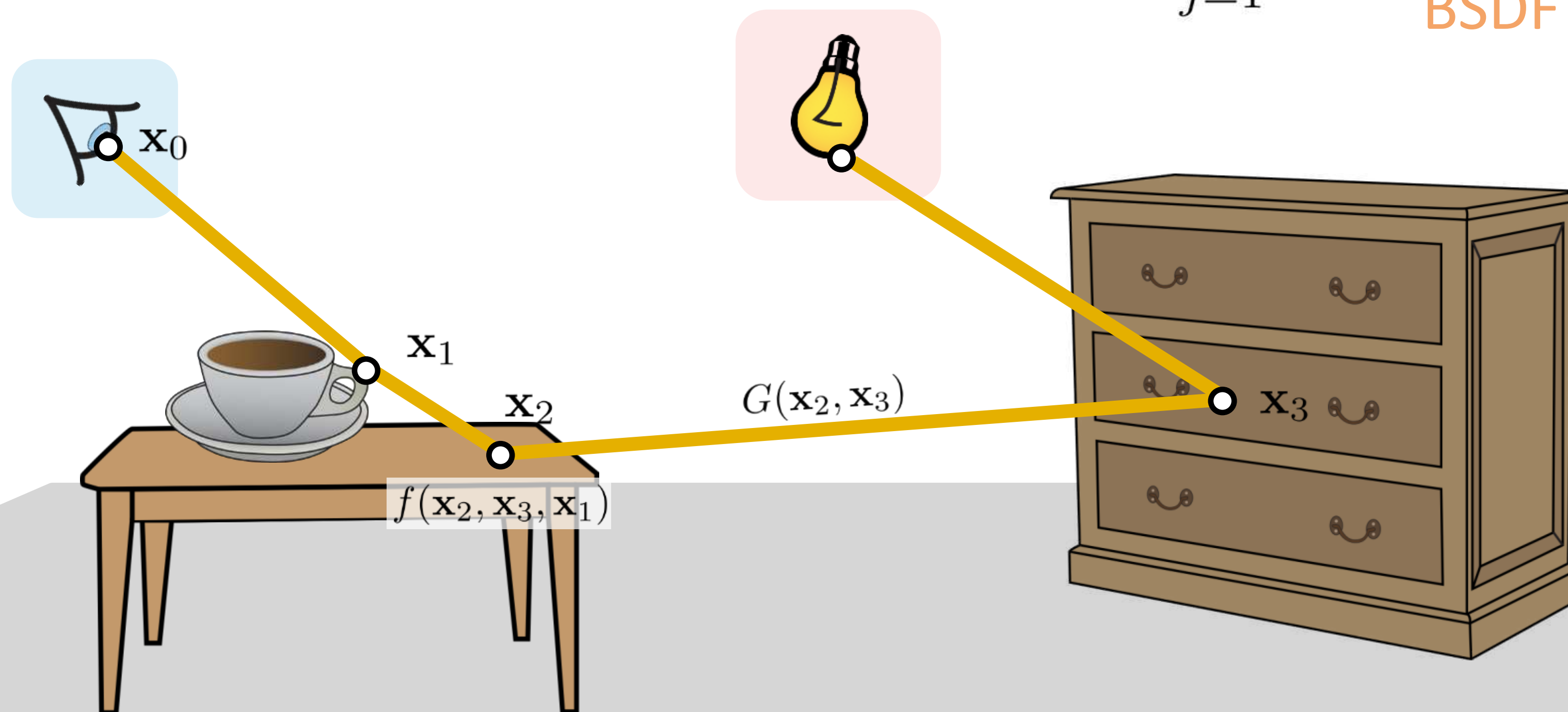
Path integral form of light transport

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

sensor weight
source weight
path throughput
light path
space of all light paths

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

BSDF
geometry



Monte Carlo rendering

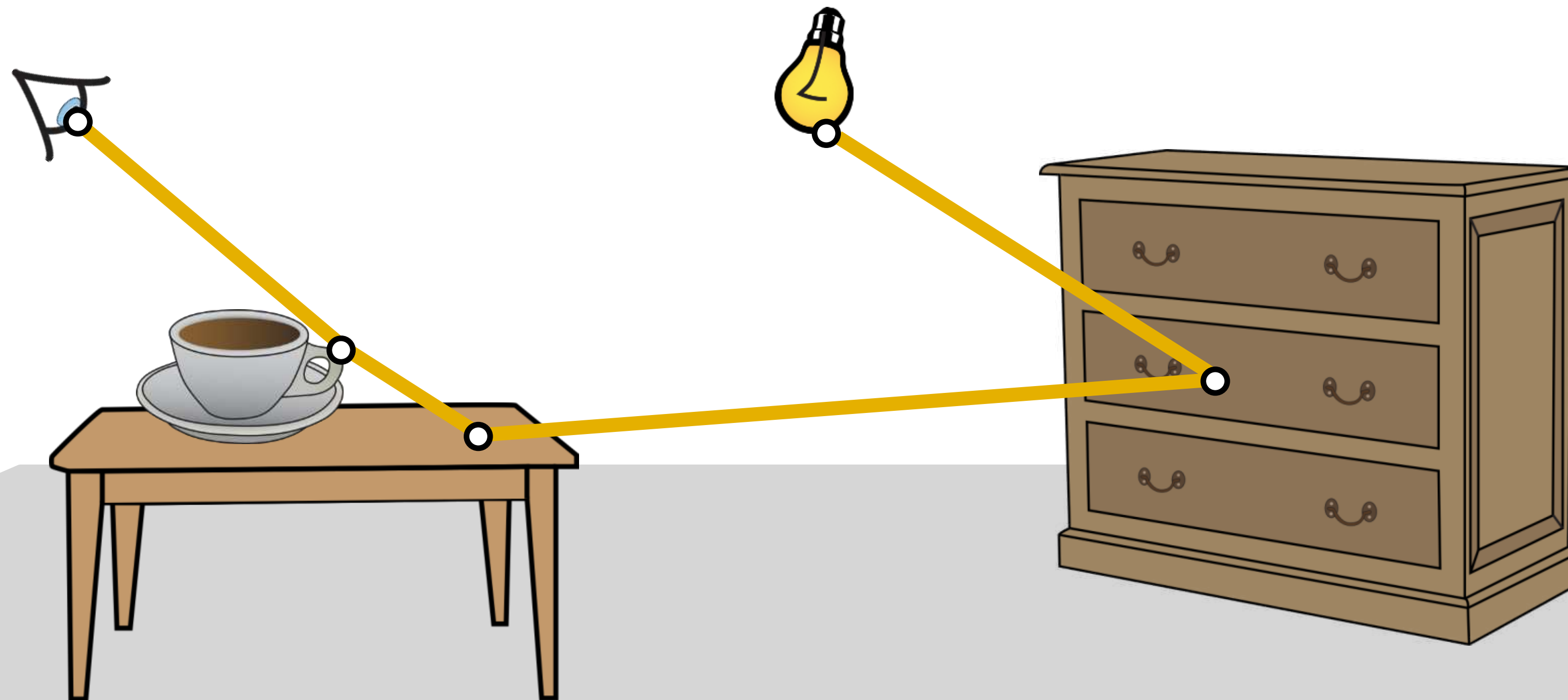
approximate image as
$$I \approx \frac{1}{N} \sum_{i=1}^N \frac{W_e(\mathbf{x}_{i,0}, \mathbf{x}_{i,1}) L_e(\mathbf{x}_{i,k}, \mathbf{x}_{i,k-1}) T(\bar{\mathbf{x}}_i)}{p(\bar{\mathbf{x}}_i)}$$



Monte Carlo rendering

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

sum over *randomly* sampled paths

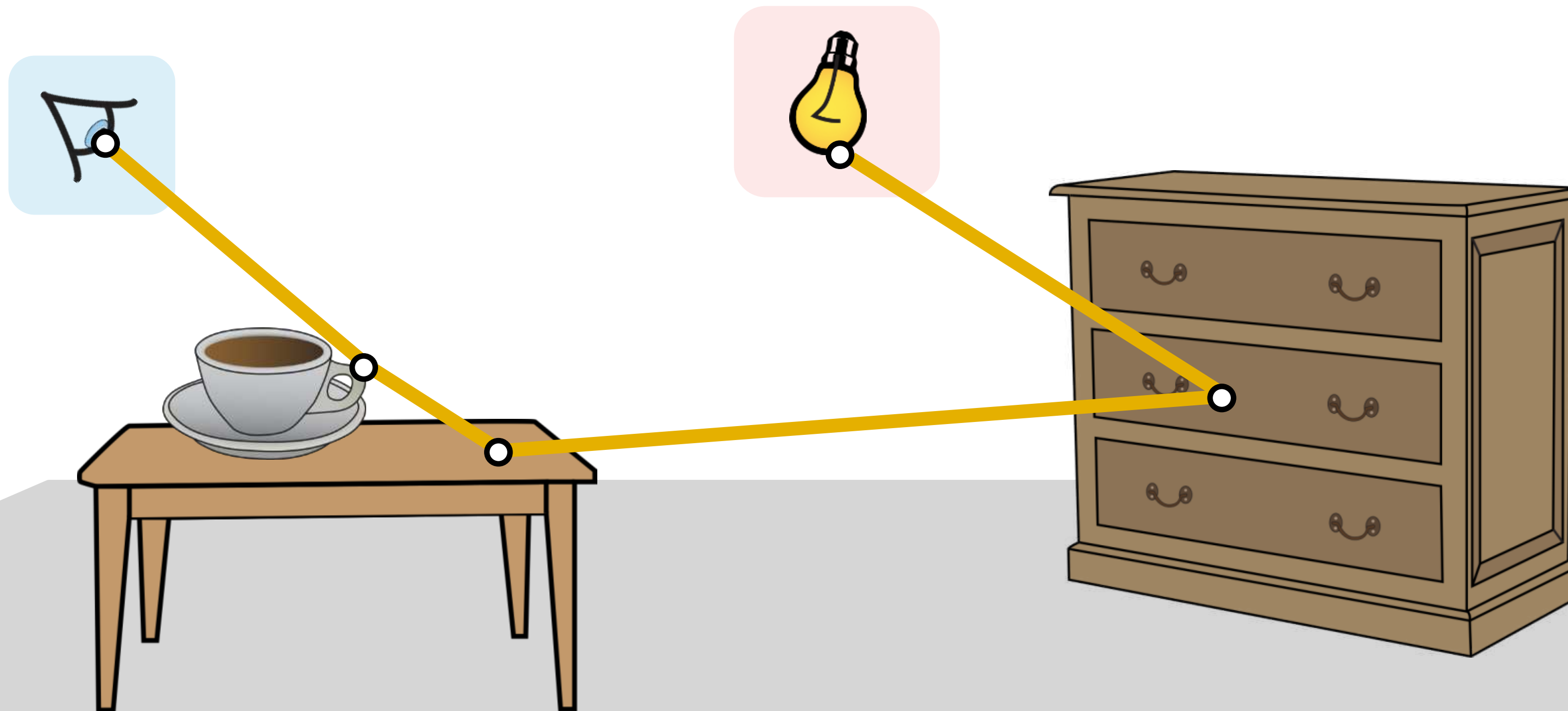


Monte Carlo rendering

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

sensor weight source weight path throughput

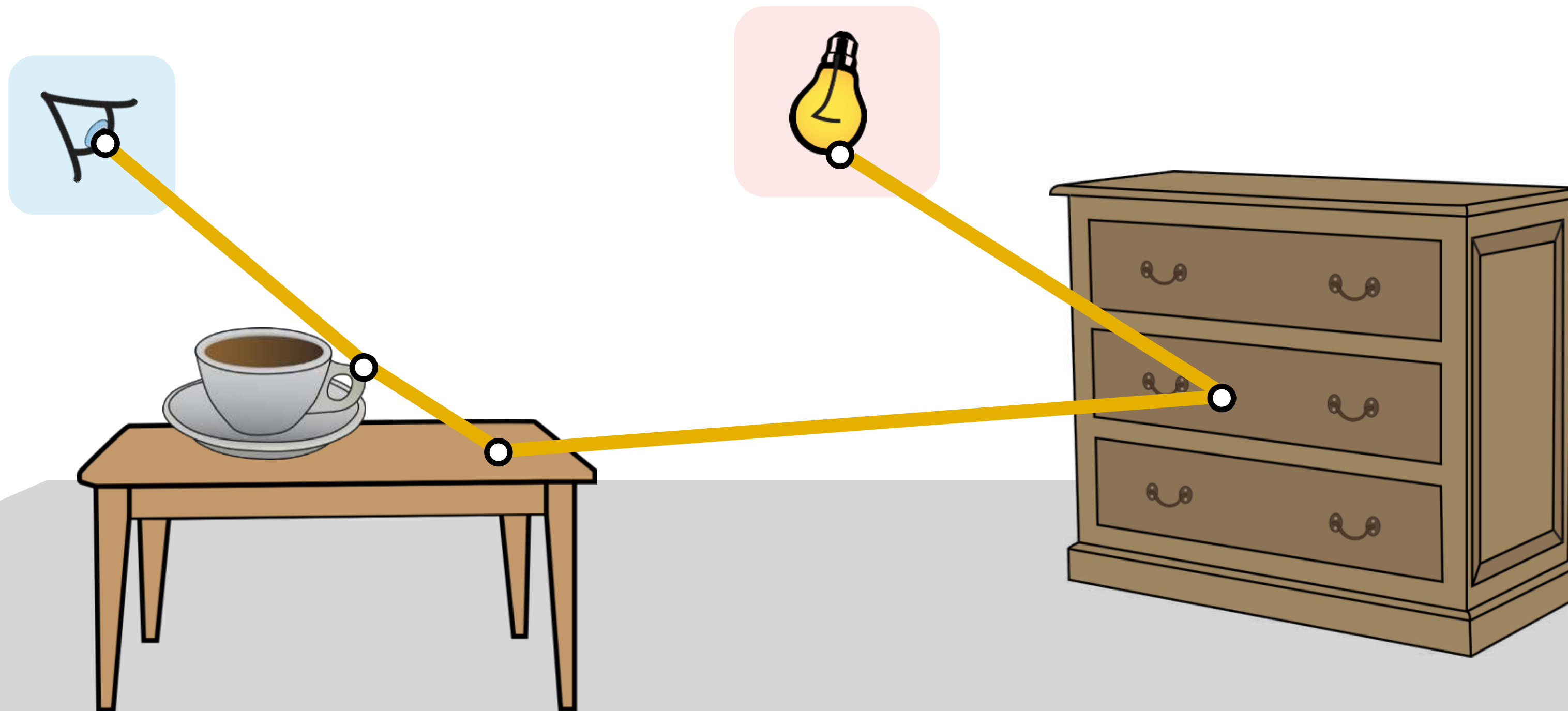
sum over *randomly* sampled paths



Monte Carlo rendering

approximate image as $I \approx \frac{1}{N} \sum_{i=1}^N \frac{\overset{\text{sensor weight}}{W_e(\mathbf{x}_{i,0}, \mathbf{x}_{i,1})} \overset{\text{source weight}}{L_e(\mathbf{x}_{i,k}, \mathbf{x}_{i,k-1})} \overset{\text{path throughput}}{T(\bar{\mathbf{x}}_i)}}{\underset{\text{PDF of random path}}{p(\bar{\mathbf{x}}_i)}}$

sum over *randomly* sampled paths



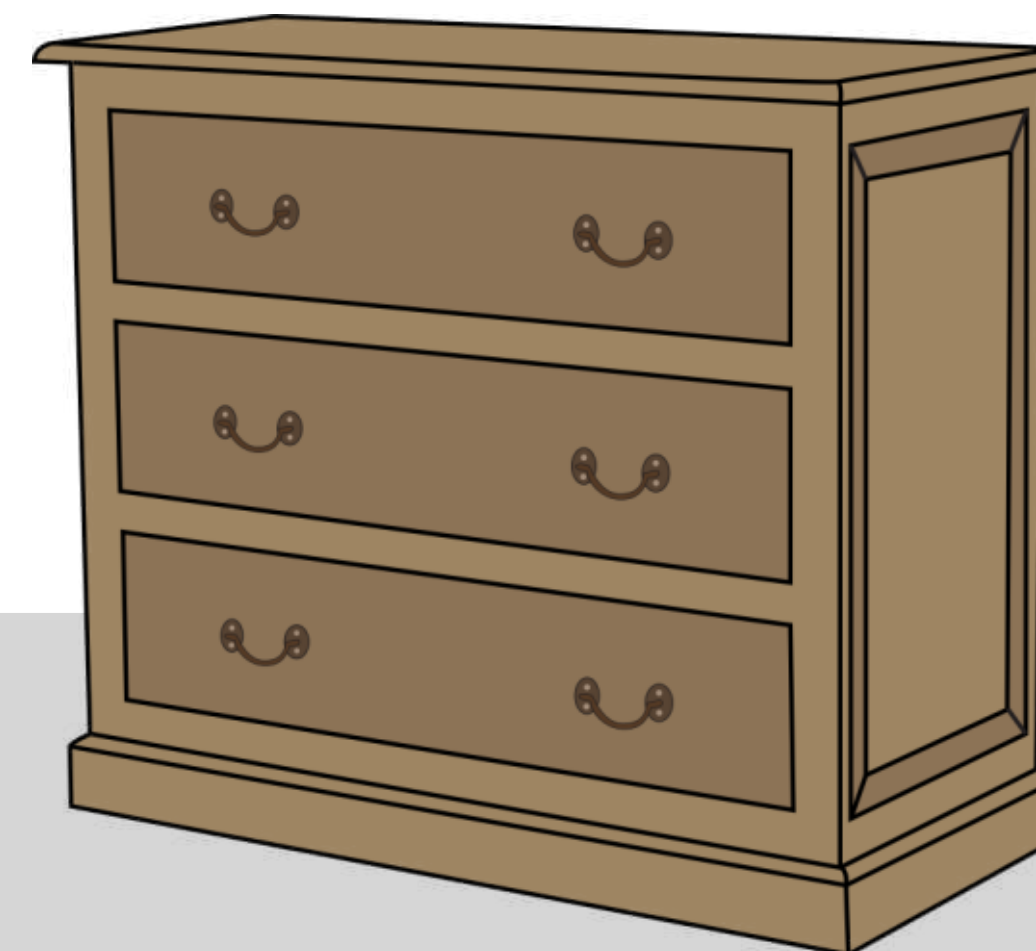
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Path tracing: sample path starting from sensor



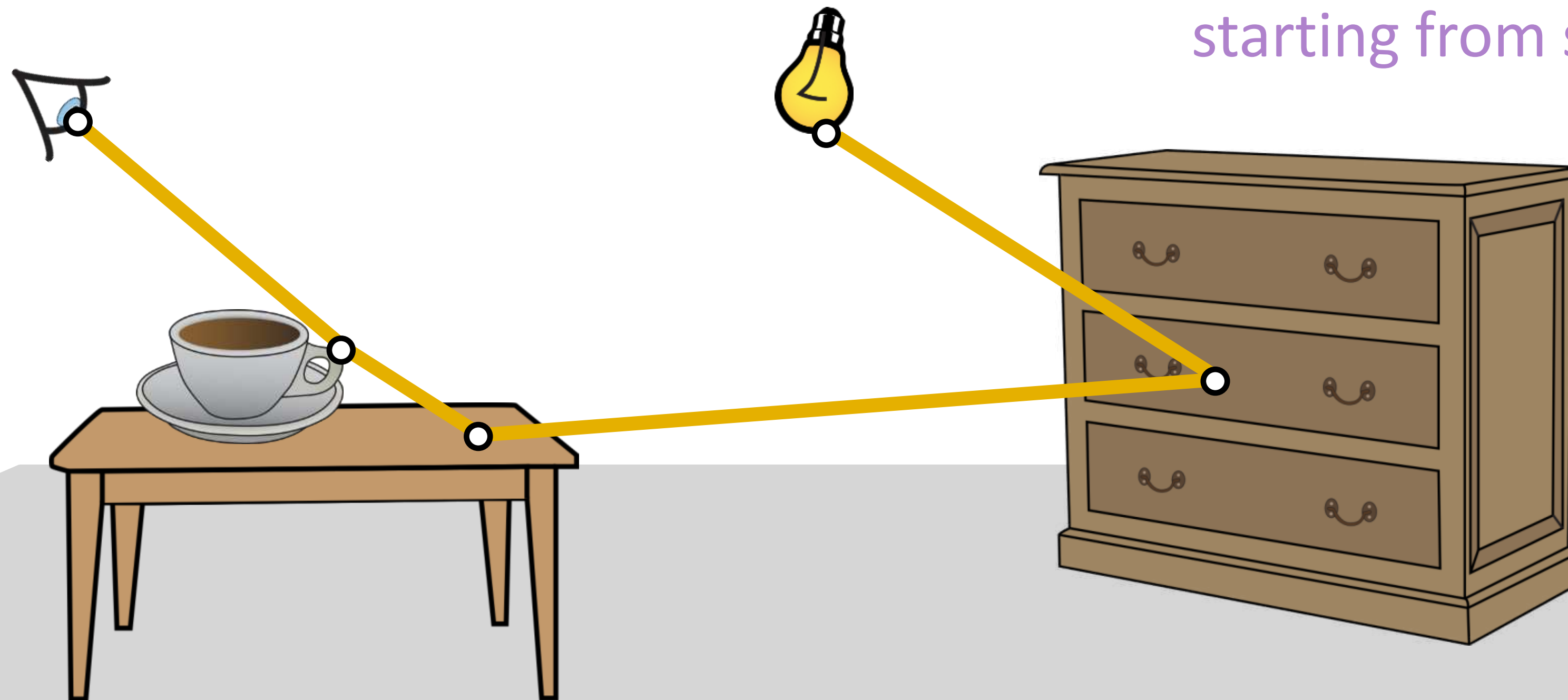
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Path tracing: sample path starting from sensor



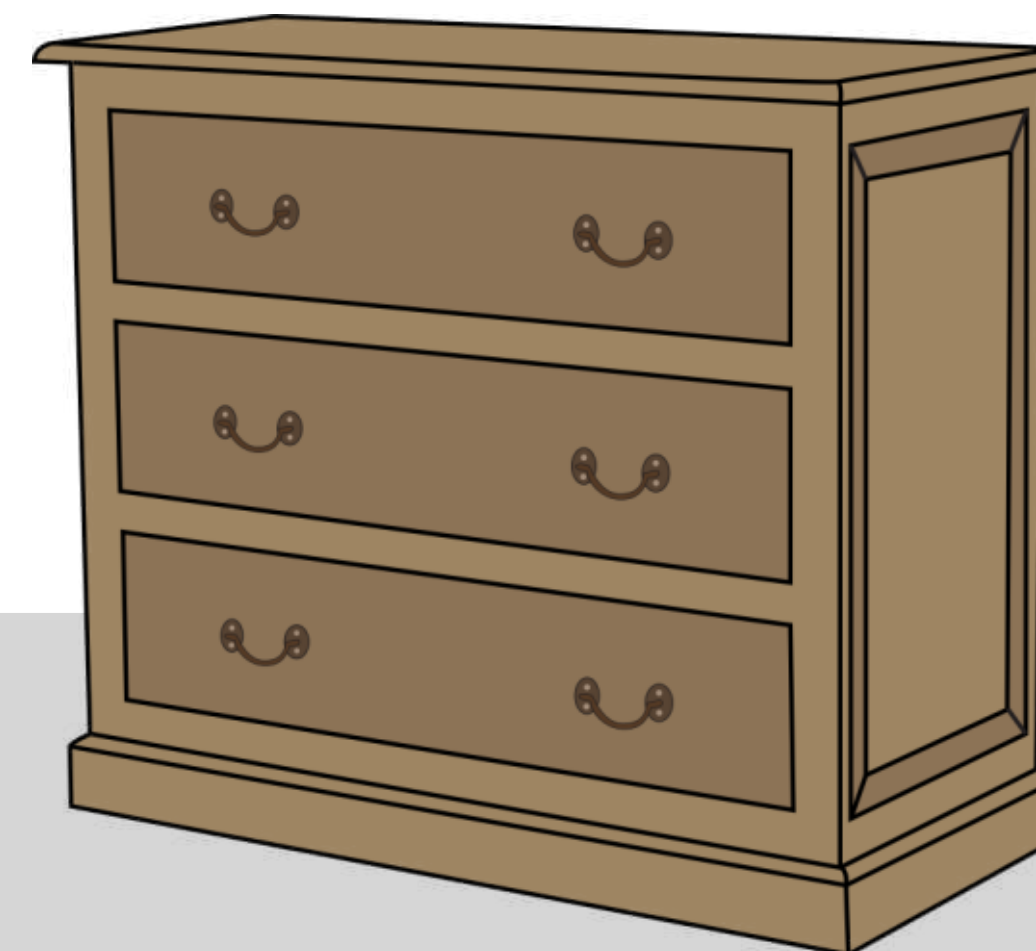
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Light tracing: sample path starting from source



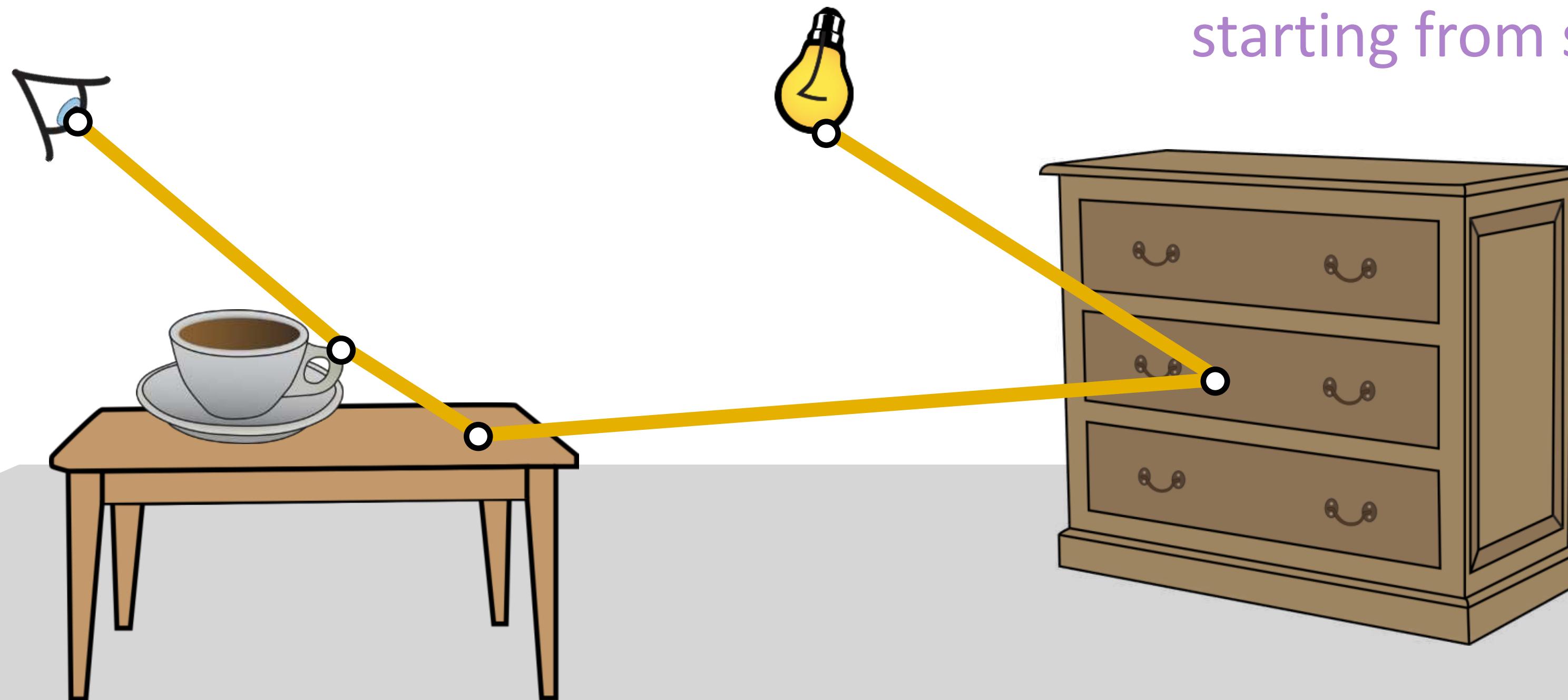
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Light tracing: sample path starting from source



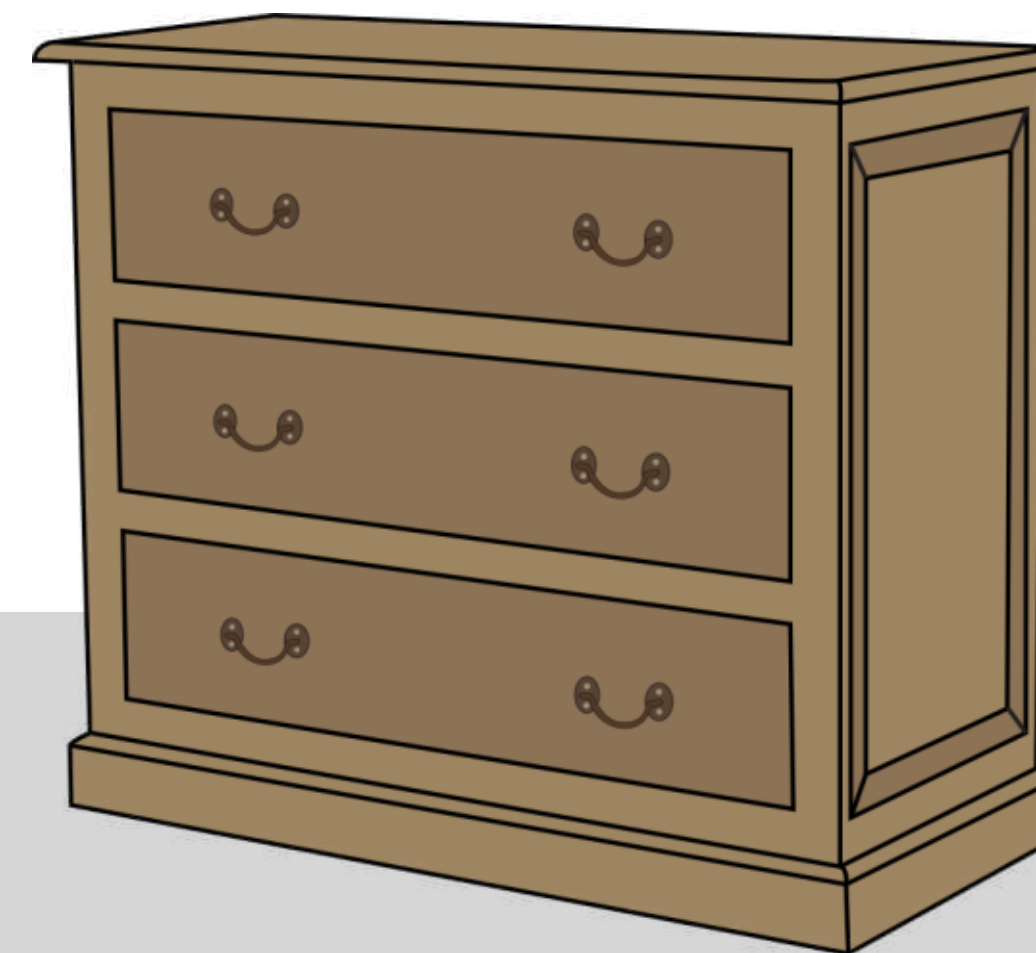
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Bidirectional path tracing: sample path starting from both source and sensor



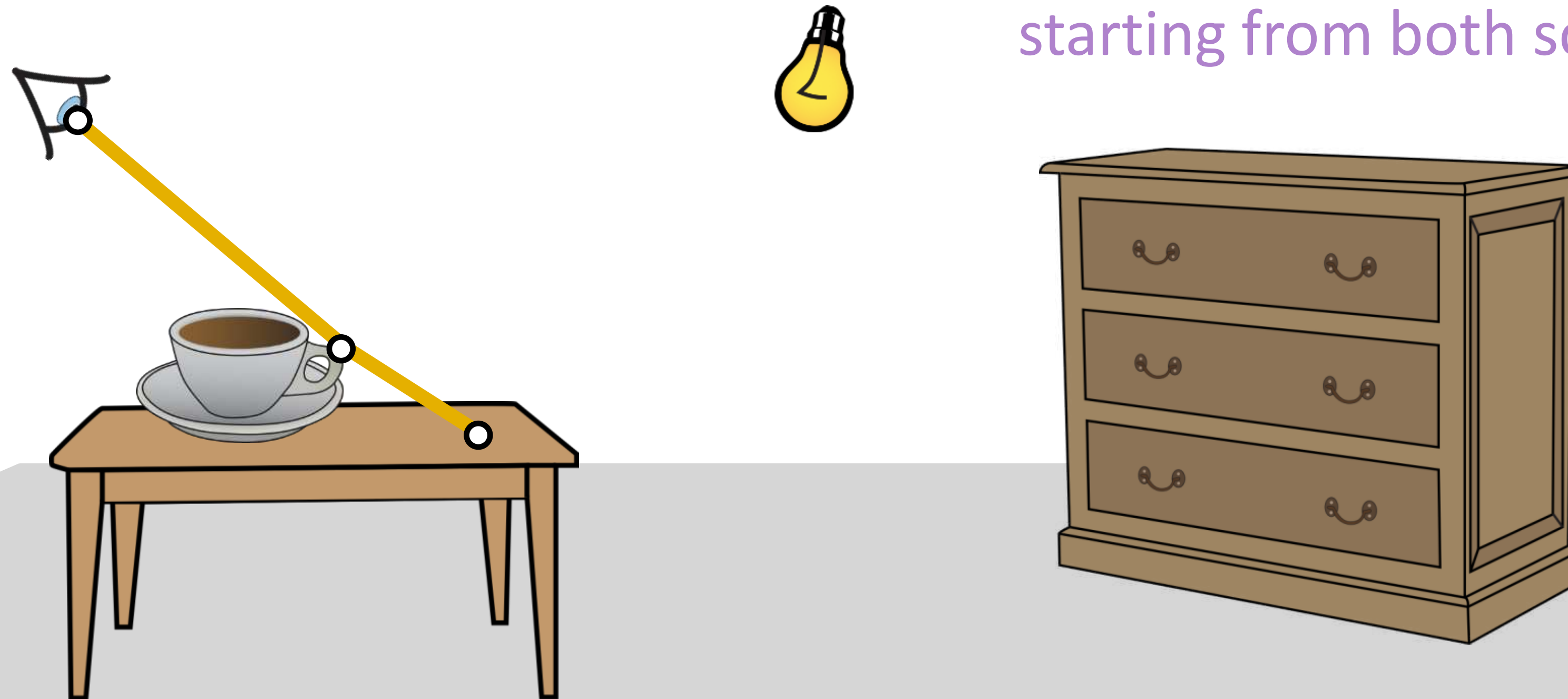
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Bidirectional path tracing: sample path starting from both source and sensor



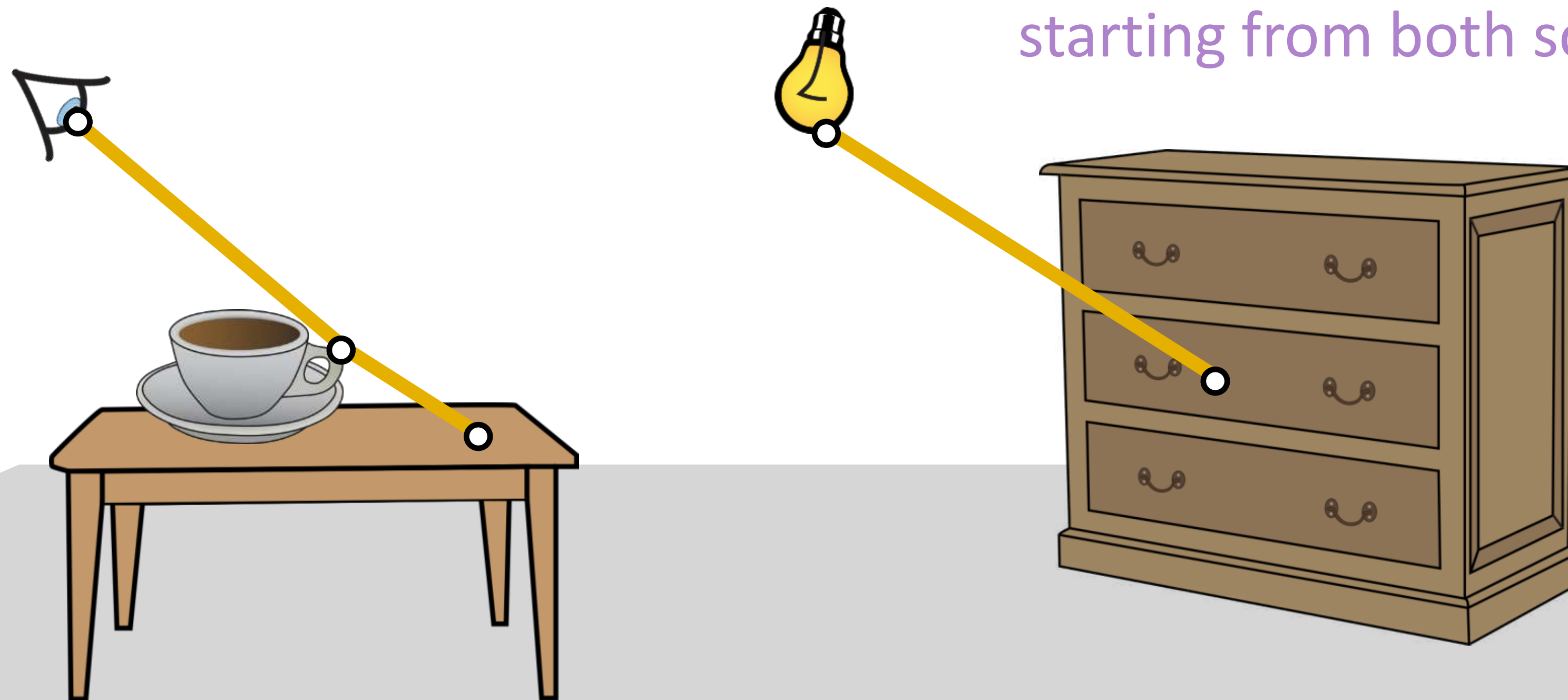
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Bidirectional path tracing: sample path starting from both source and sensor



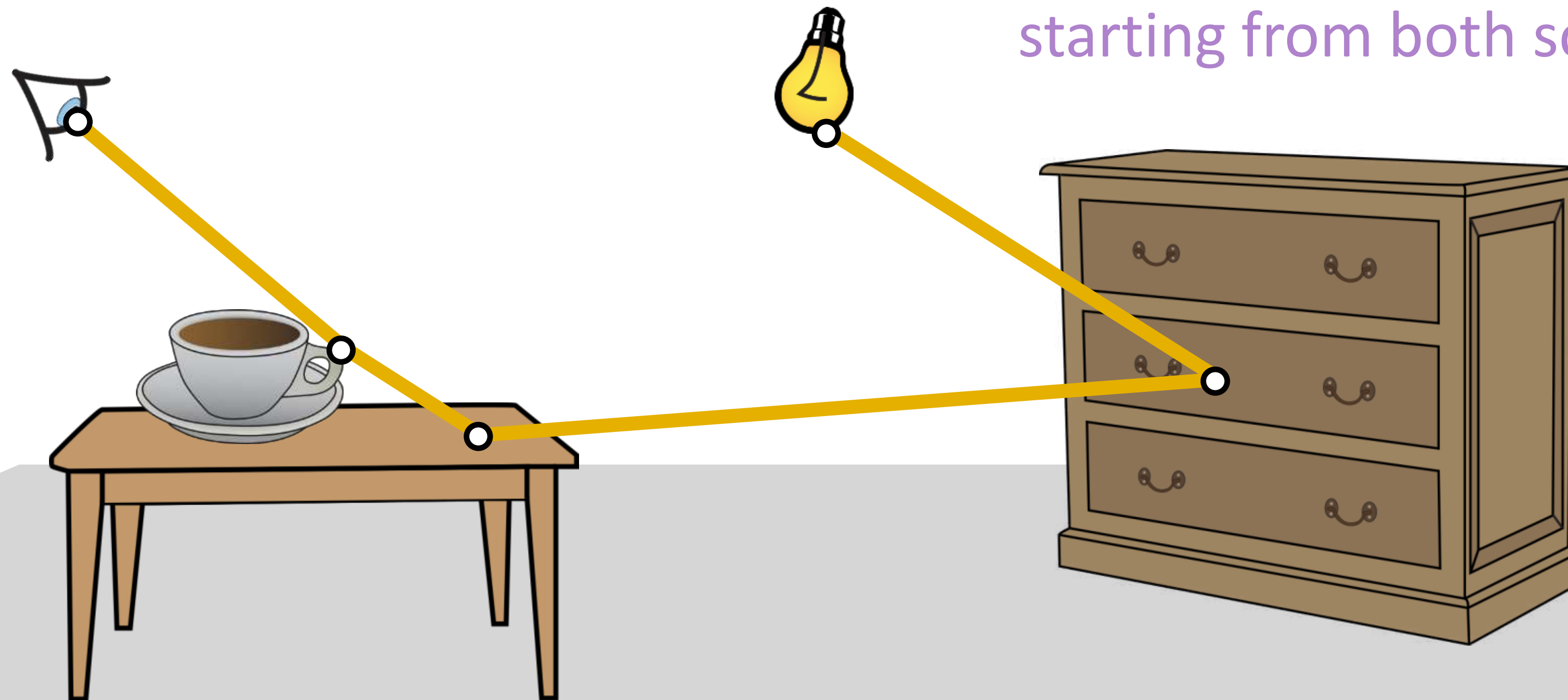
Monte Carlo rendering

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

sum over *randomly* sampled paths

PDF of random path

Bidirectional path tracing: sample path starting from both source and sensor



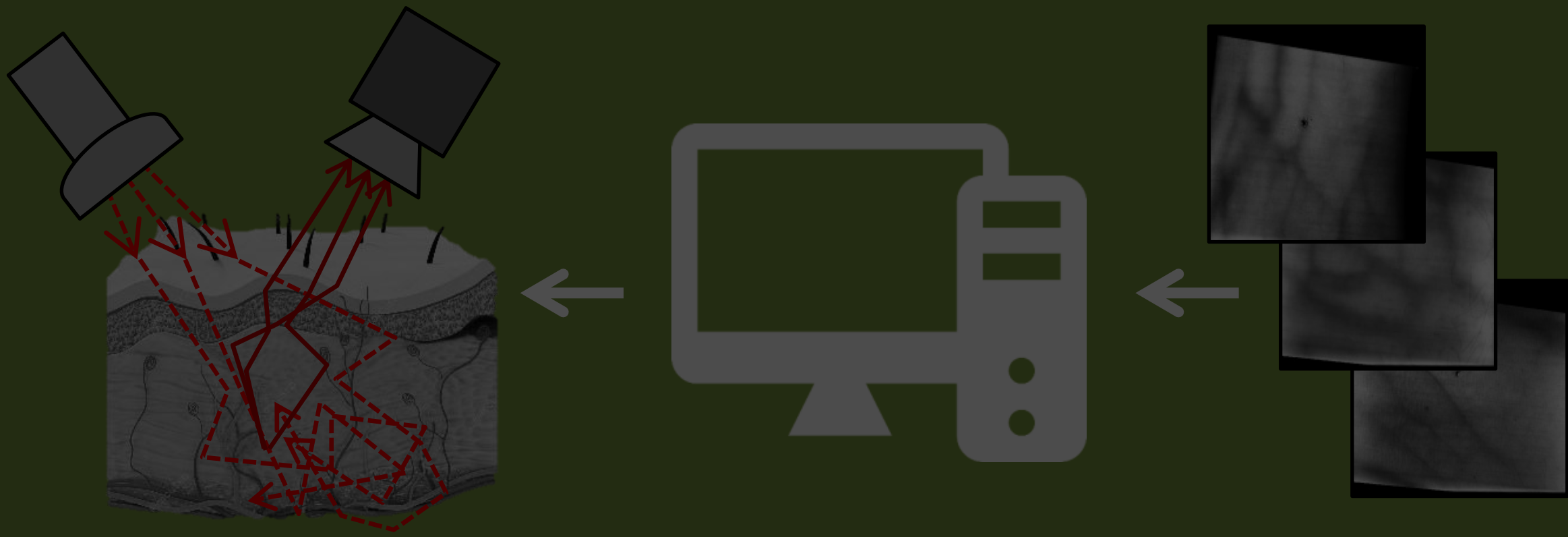
Physics-based rendering and its applications to computational imaging

forward rendering

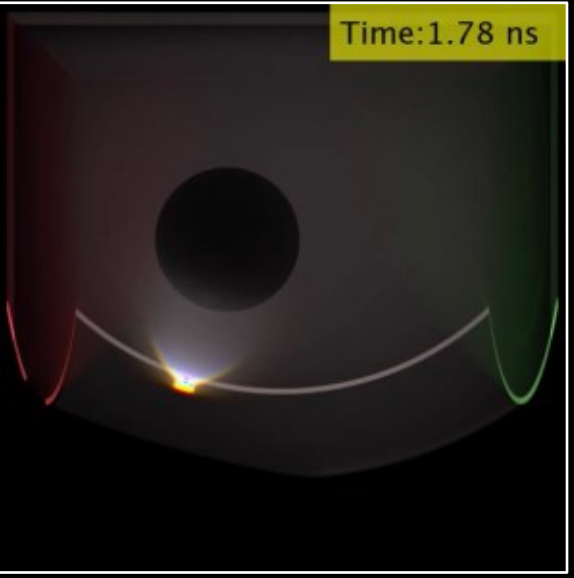


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

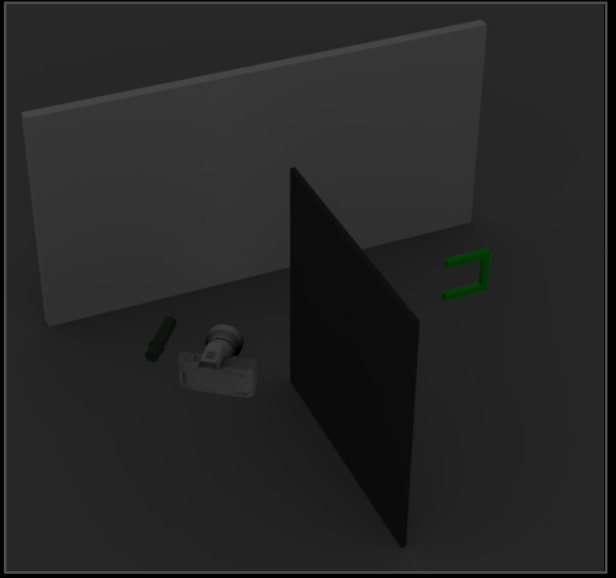
inverse rendering



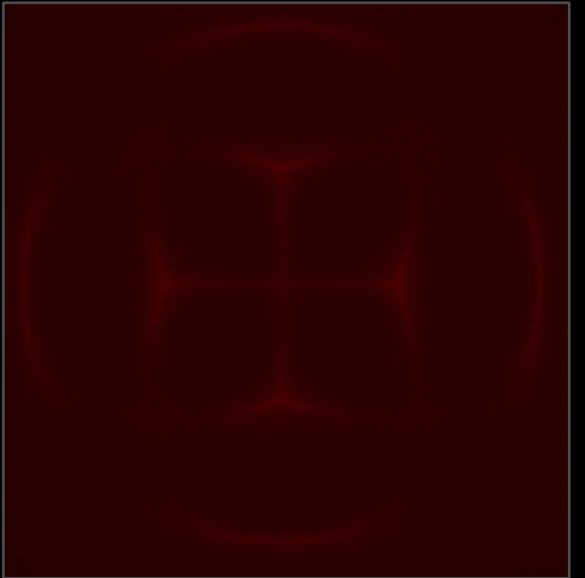
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



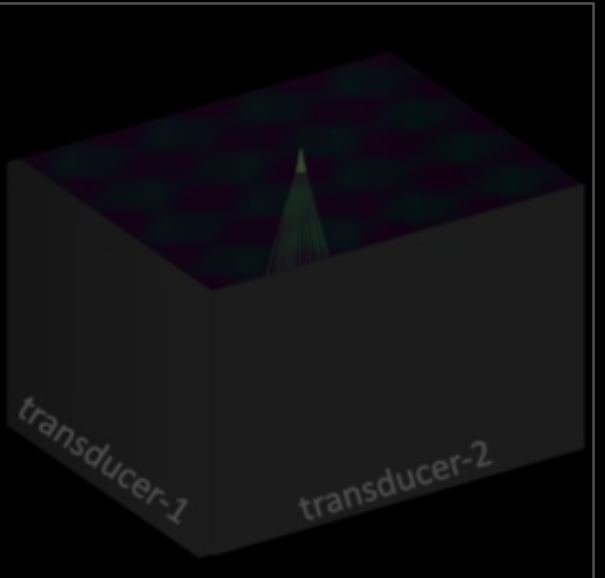
time-of-flight imaging



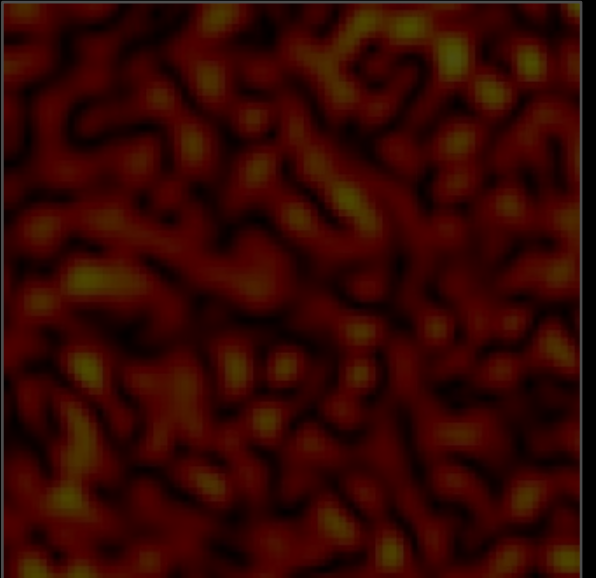
non-line-of-sight imaging



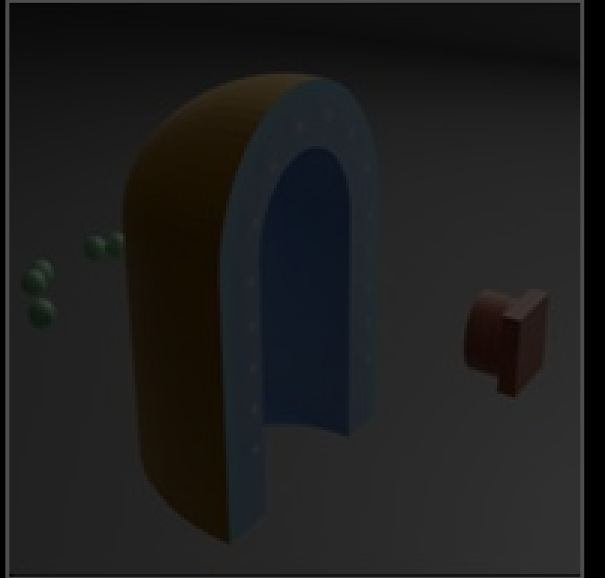
acousto-optic lensing



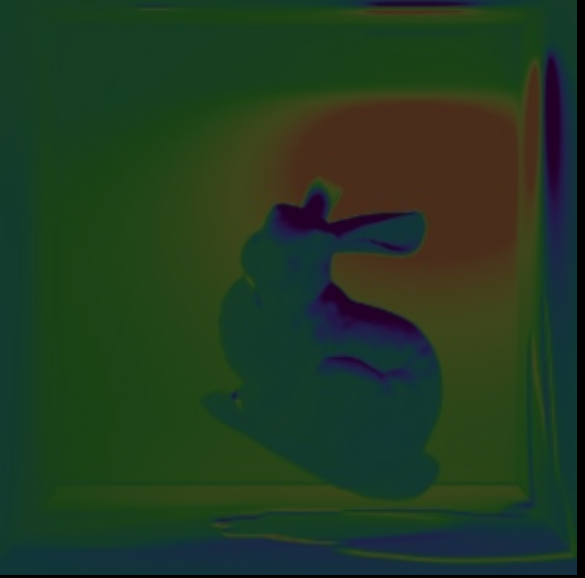
ultrafast light scanning



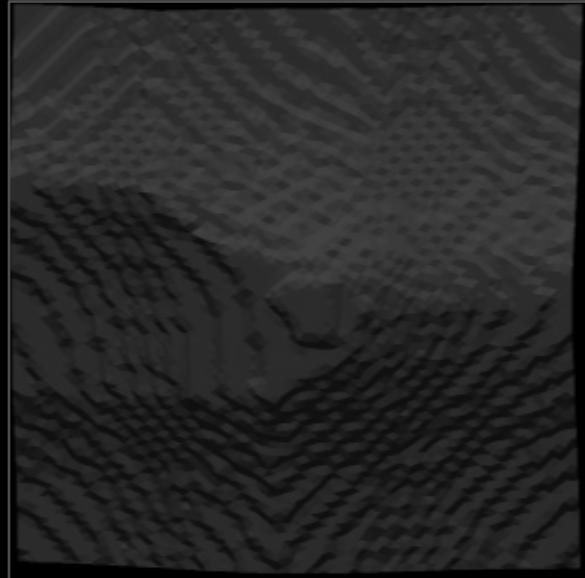
speckle imaging



tactile sensor design



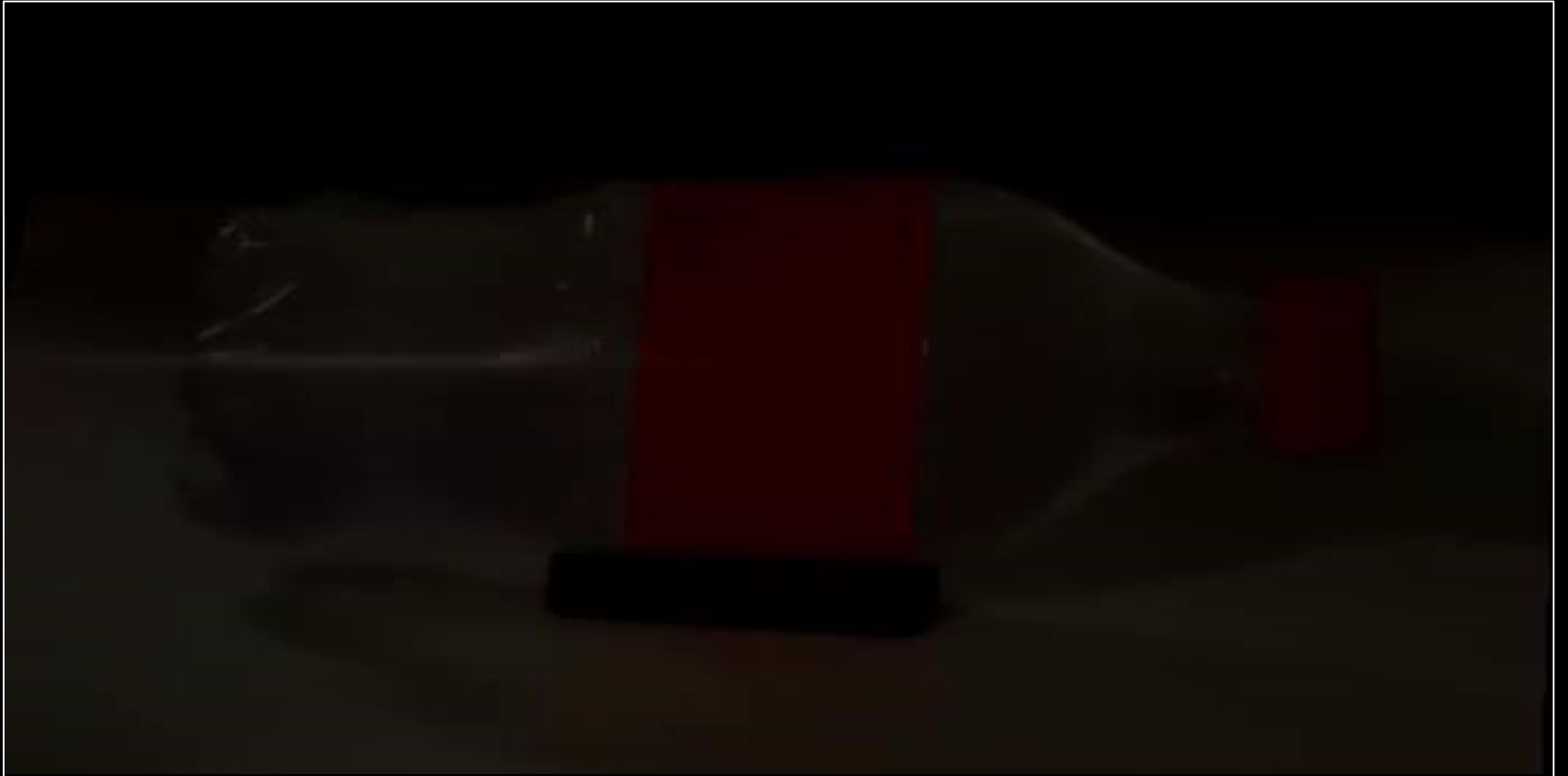
differentiable renderer



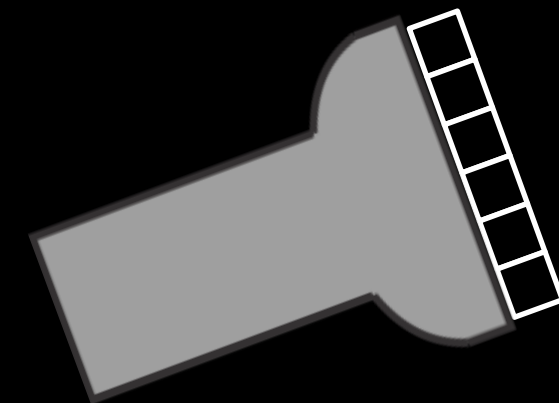
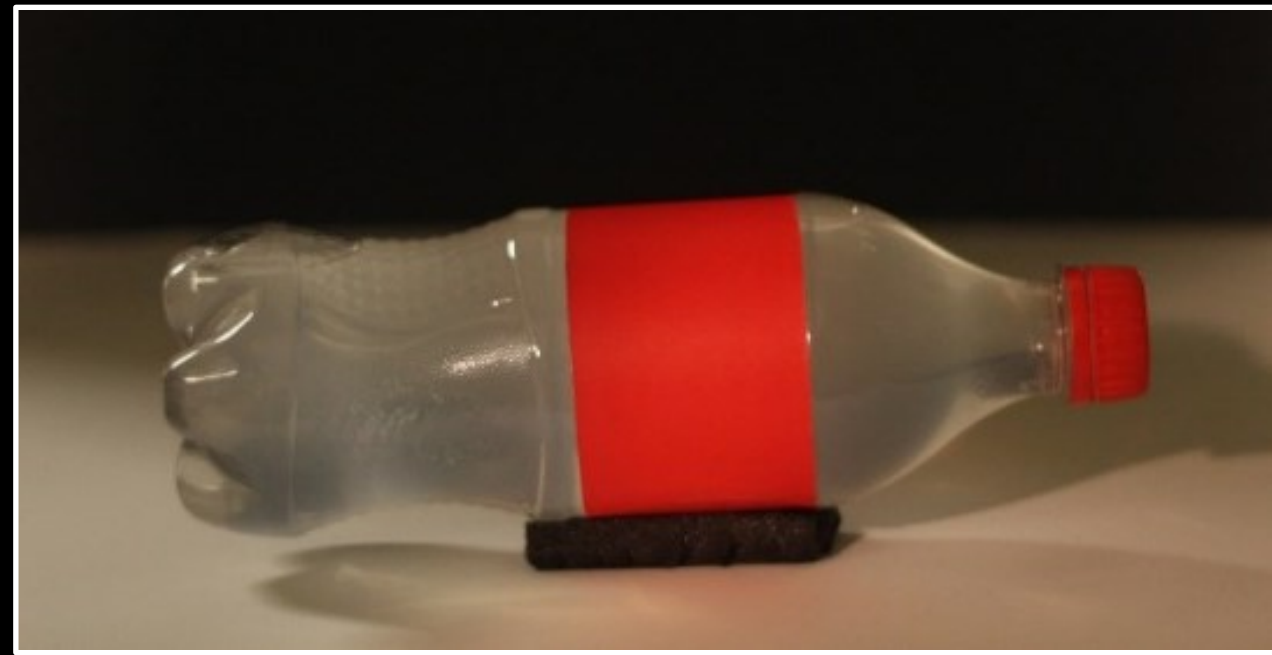
inverse problems

Time-of-flight cameras

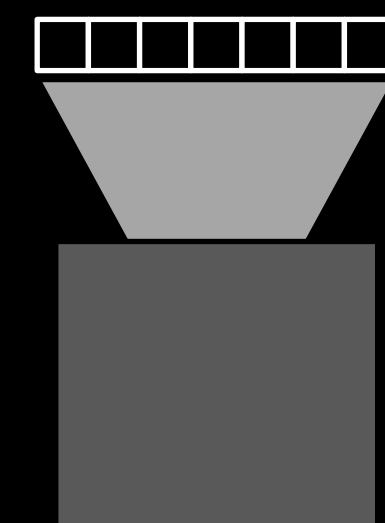
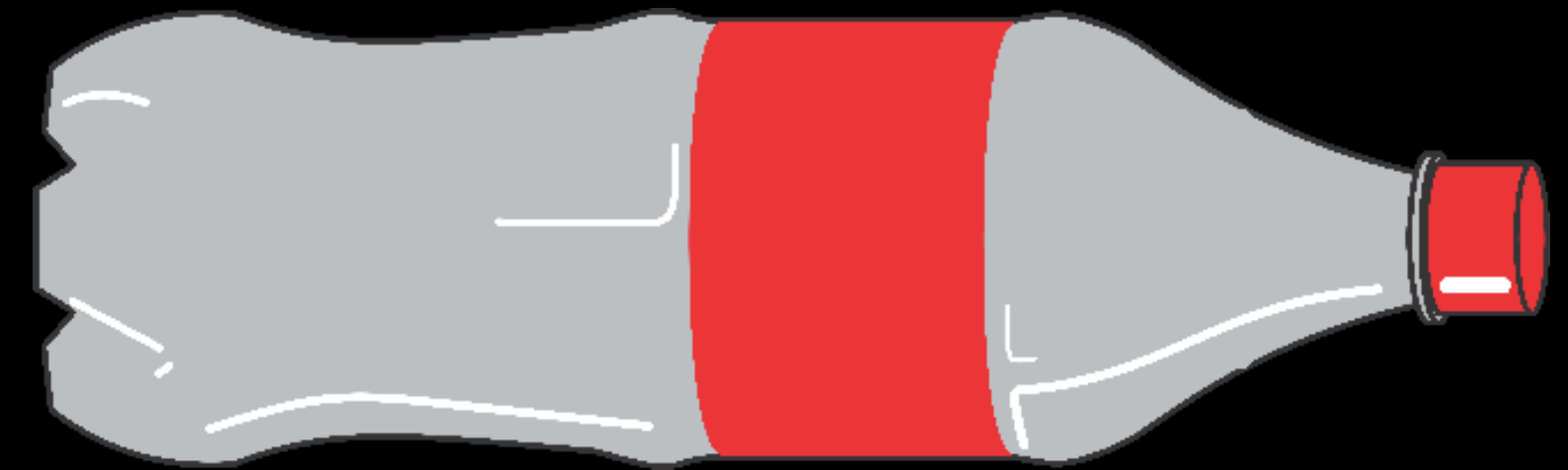
Time-of-flight cameras



Time-of-flight cameras

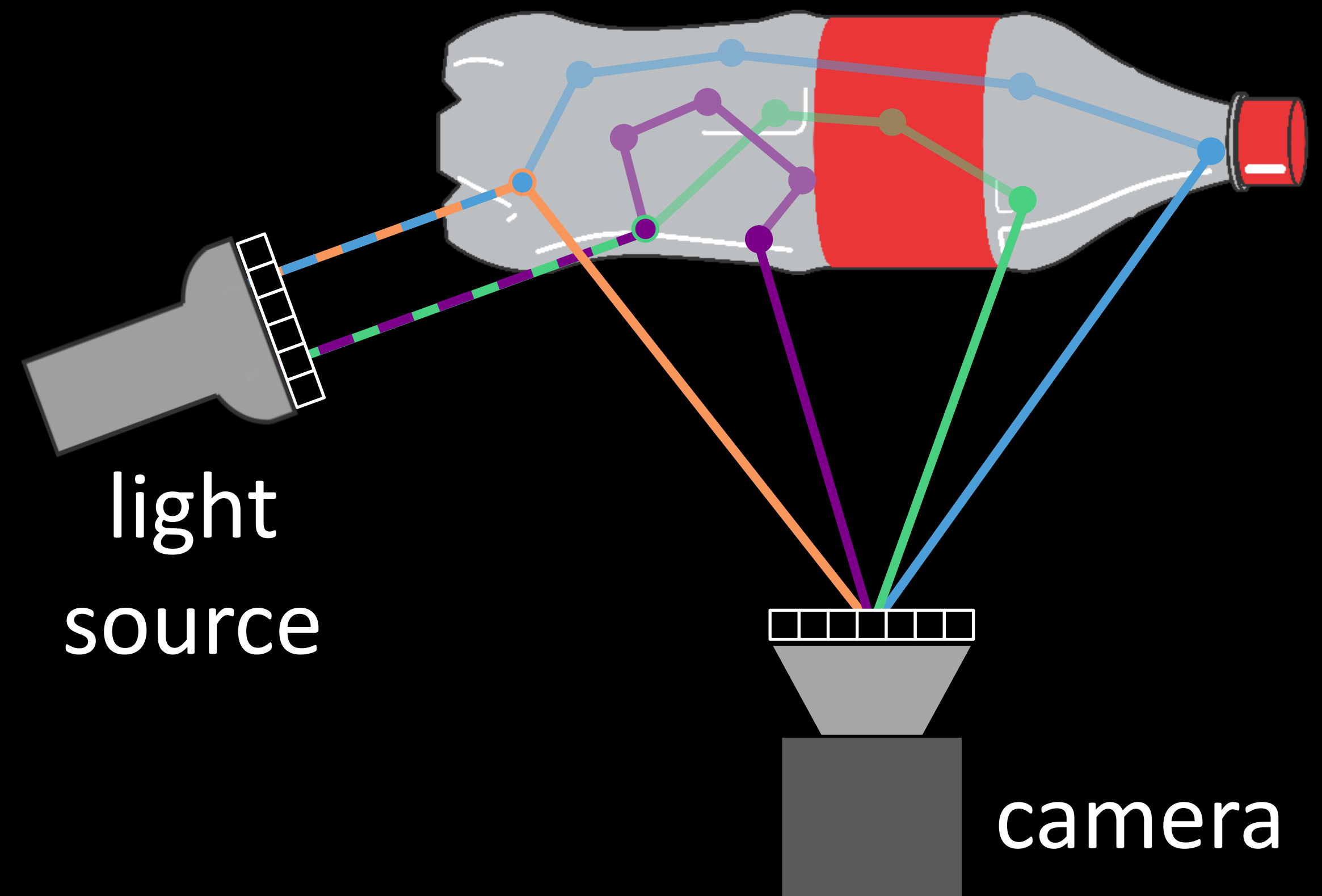
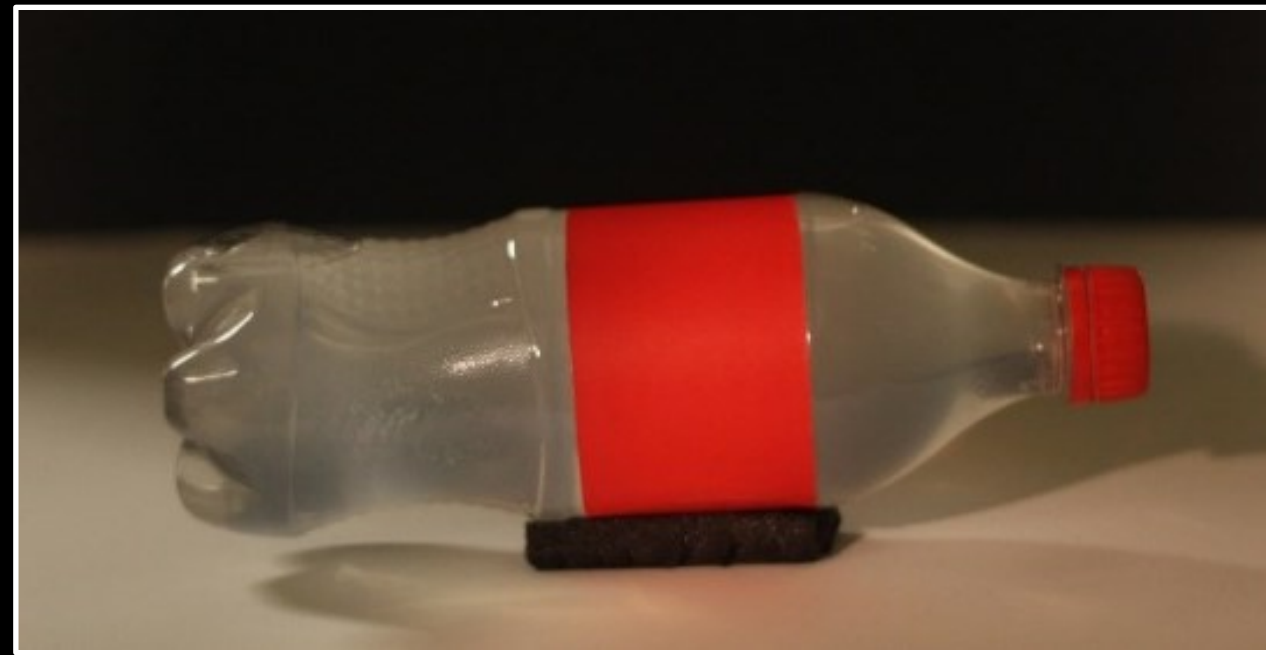


light
source

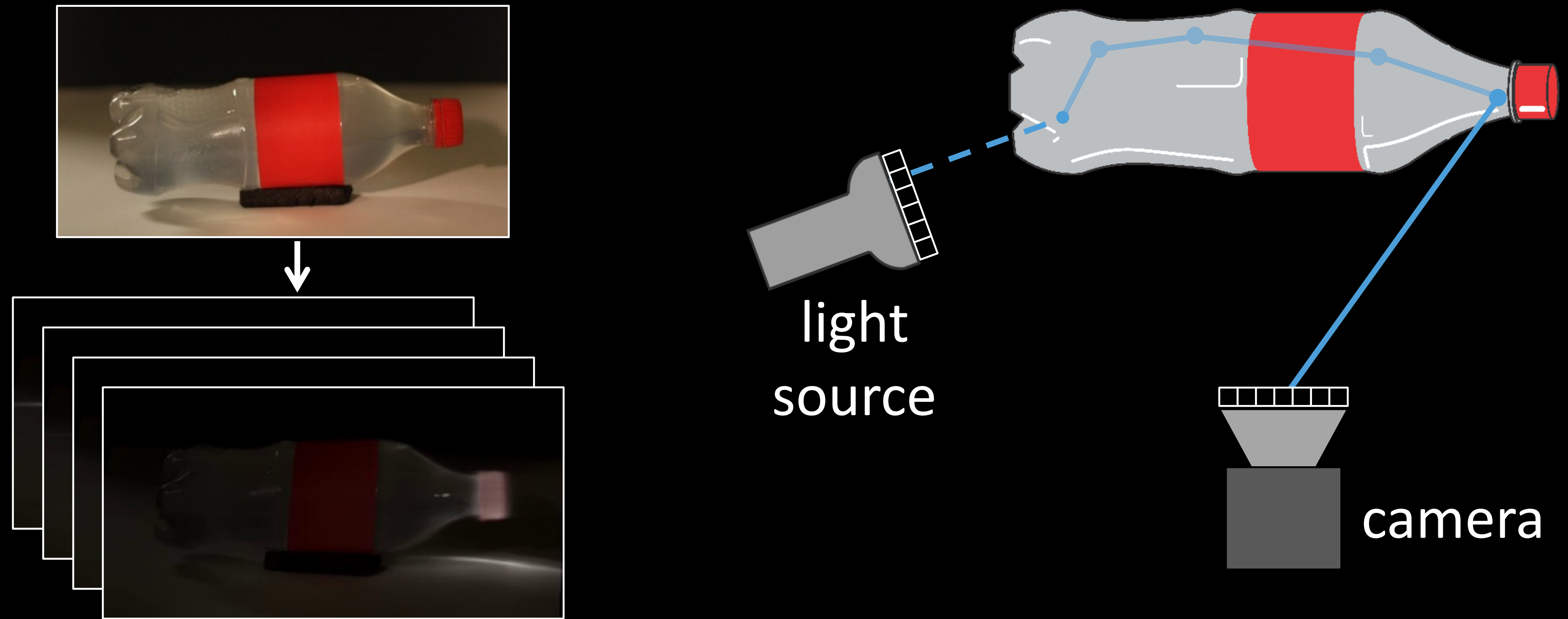


camera

Time-of-flight cameras



Time-of-flight cameras



Time-of-flight cameras



SwissRanger SR4000



Hamamatsu streak



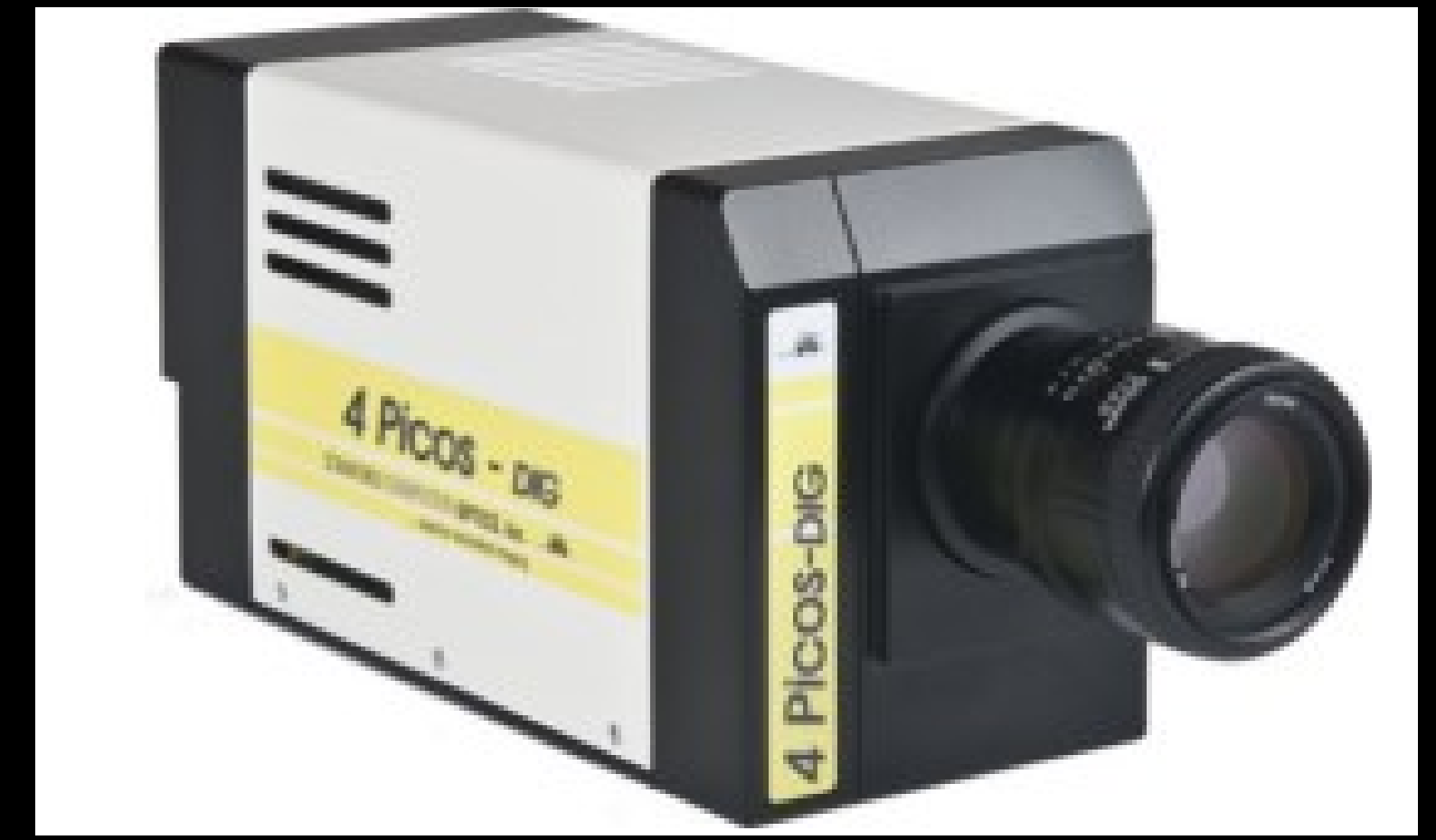
Brightway VISDOM



Microsoft Kinect



MPD SPAD



Stanford ICCD

Time-of-flight cameras



SwissRanger SR4000



Hamamatsu streak



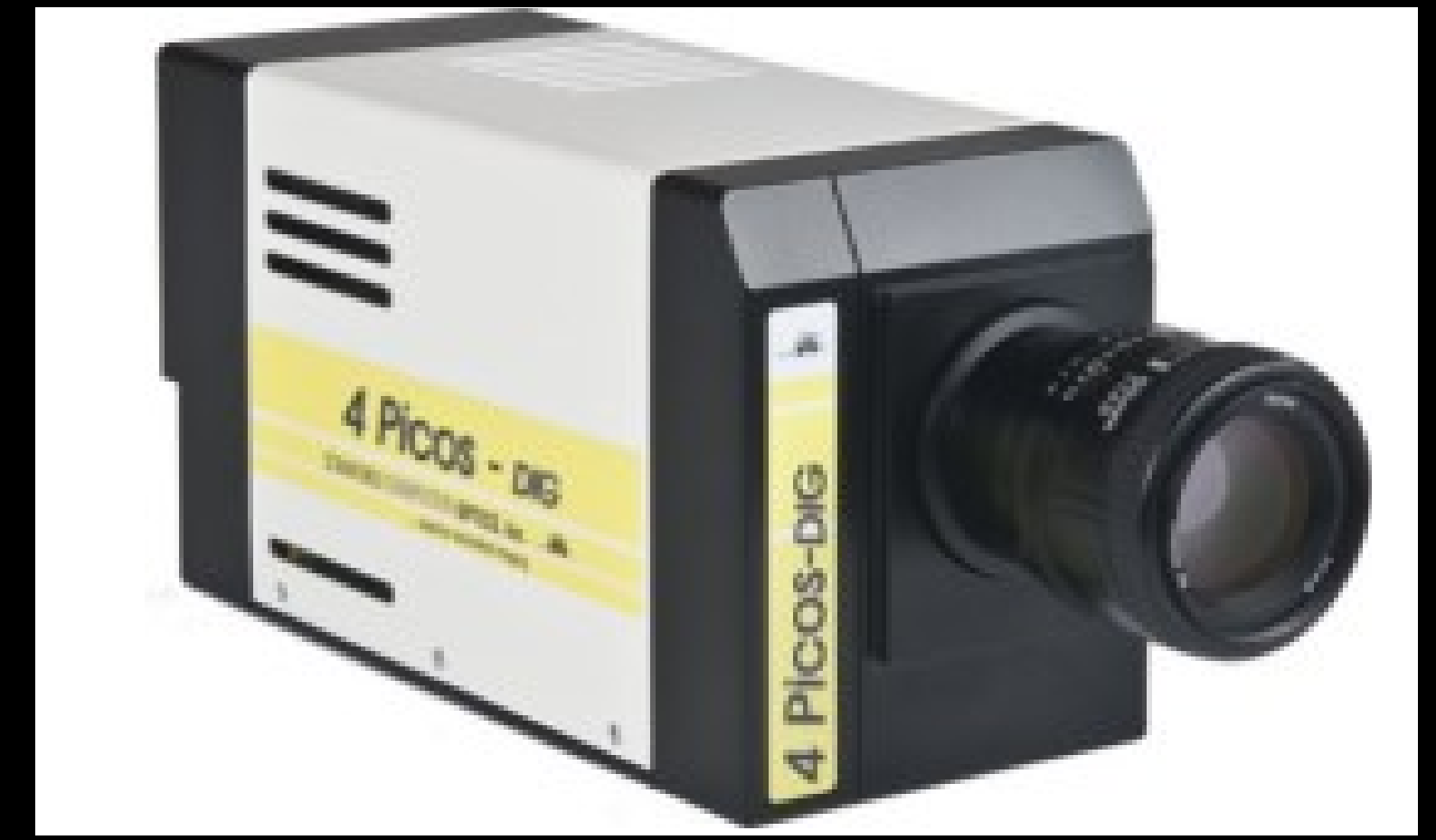
Brightway VISDOM



Microsoft Kinect
continuous-wave



MPD SPAD
transient



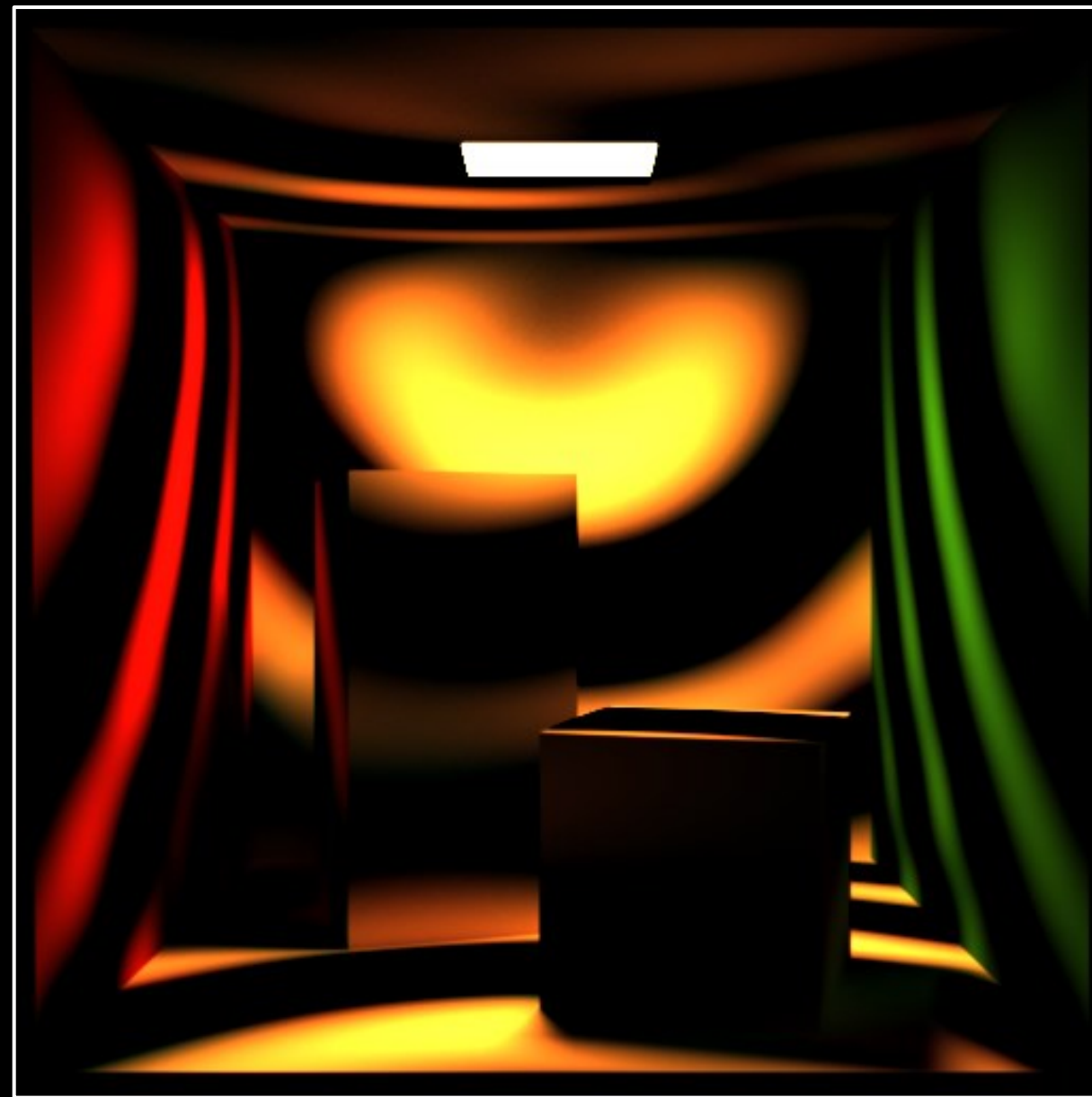
Stanford ICCD
time-gated

Time-of-flight cameras

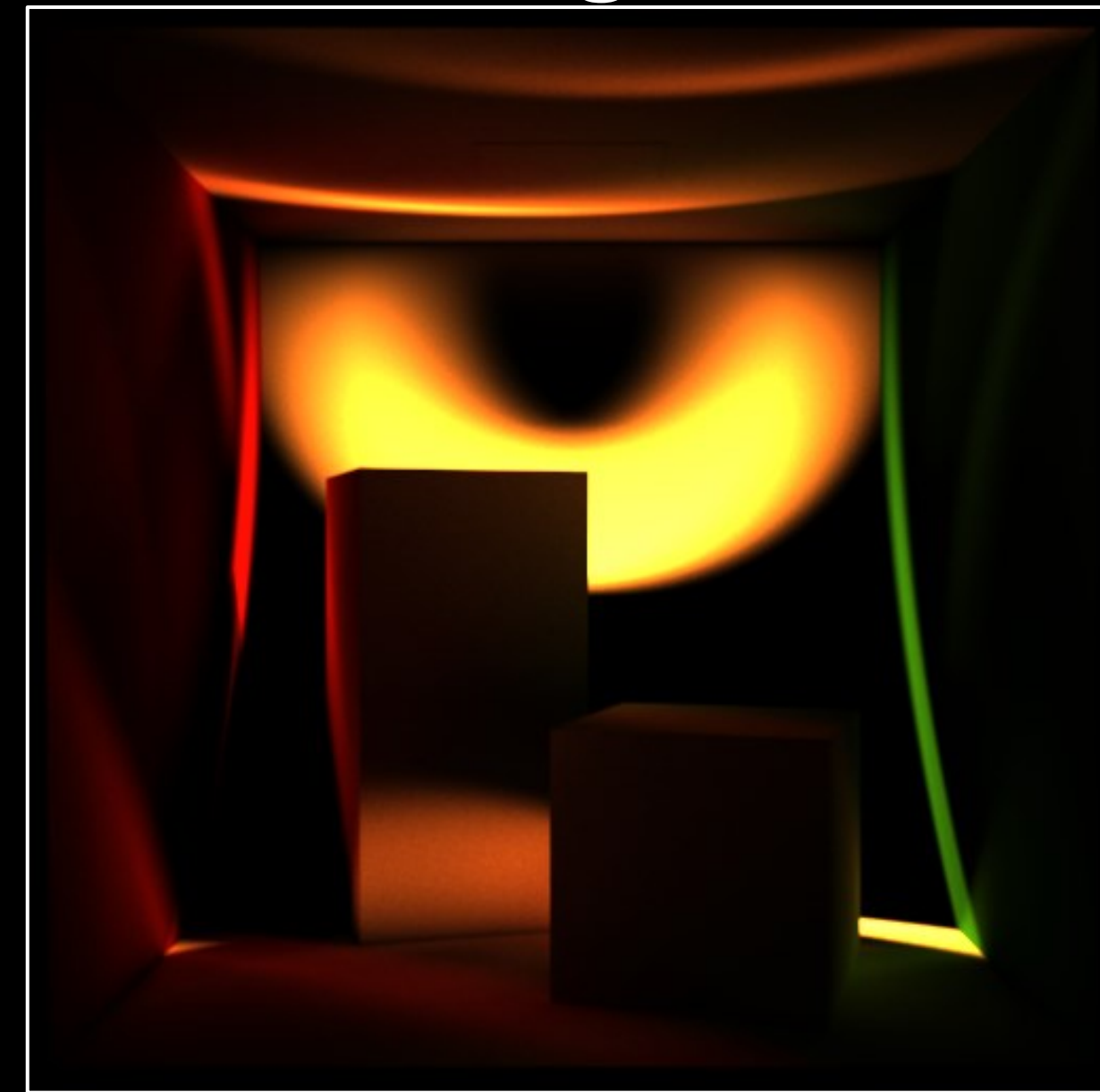
intensity



continuous wave

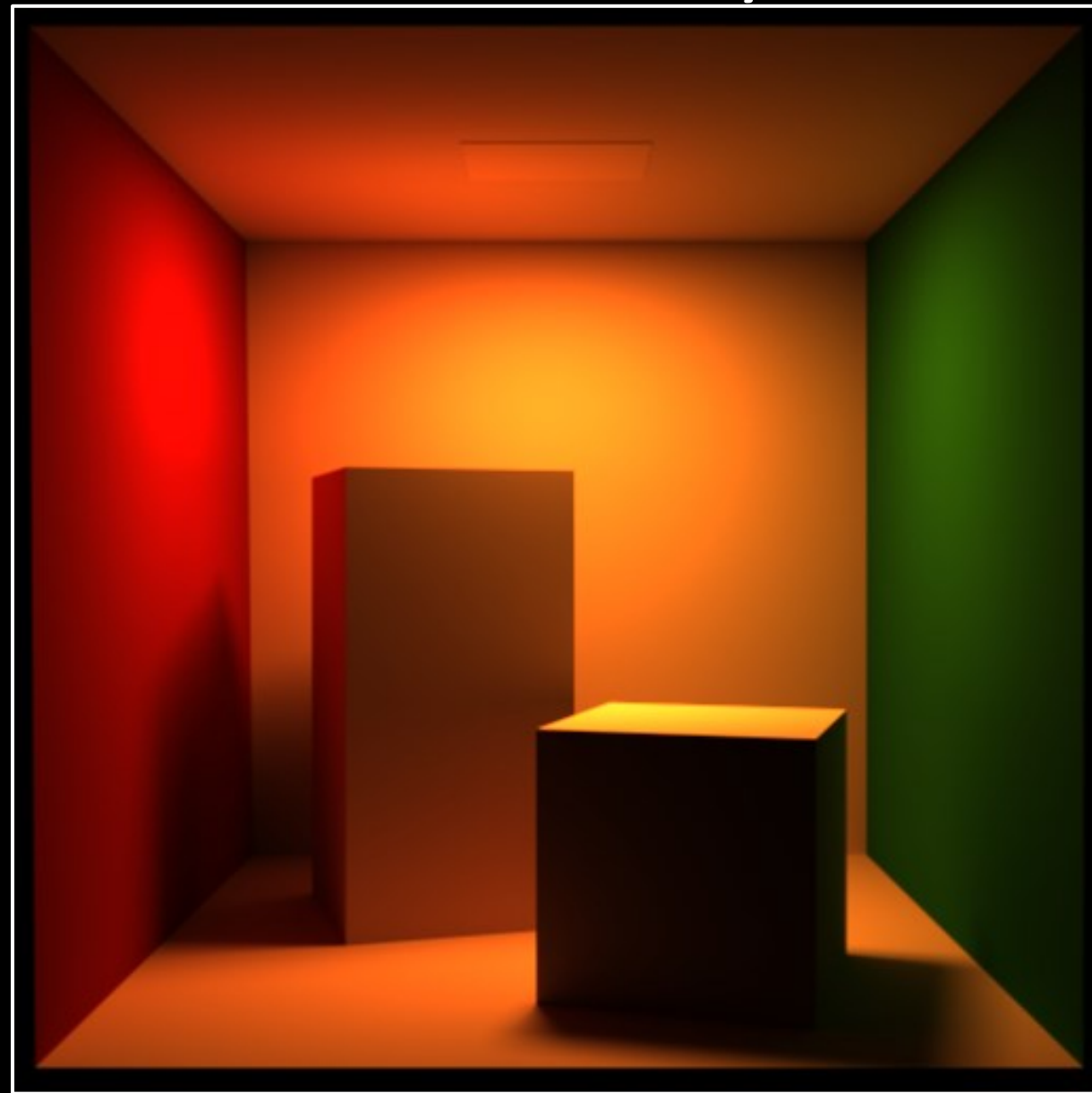


transient

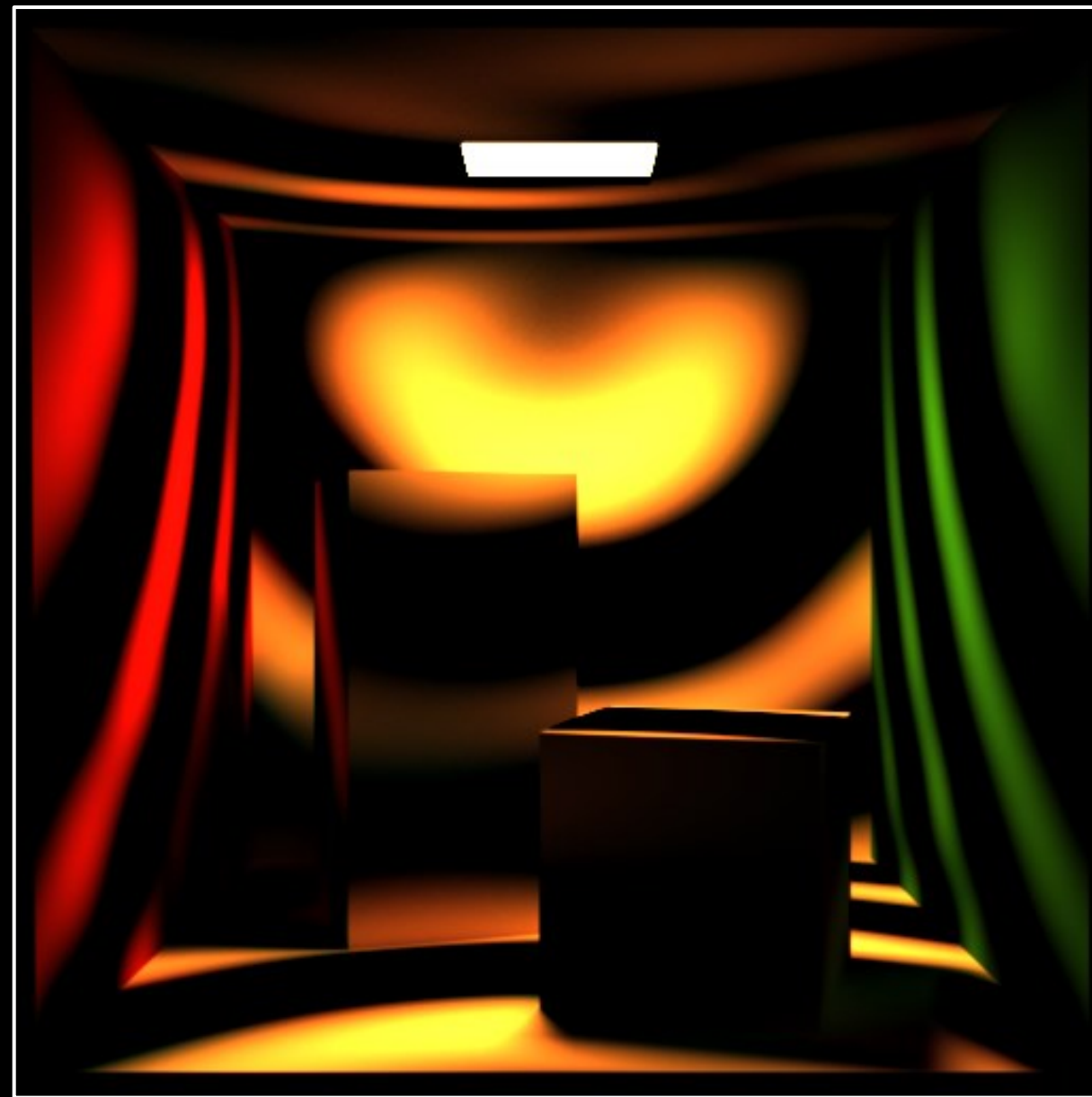


Time-of-flight cameras

intensity



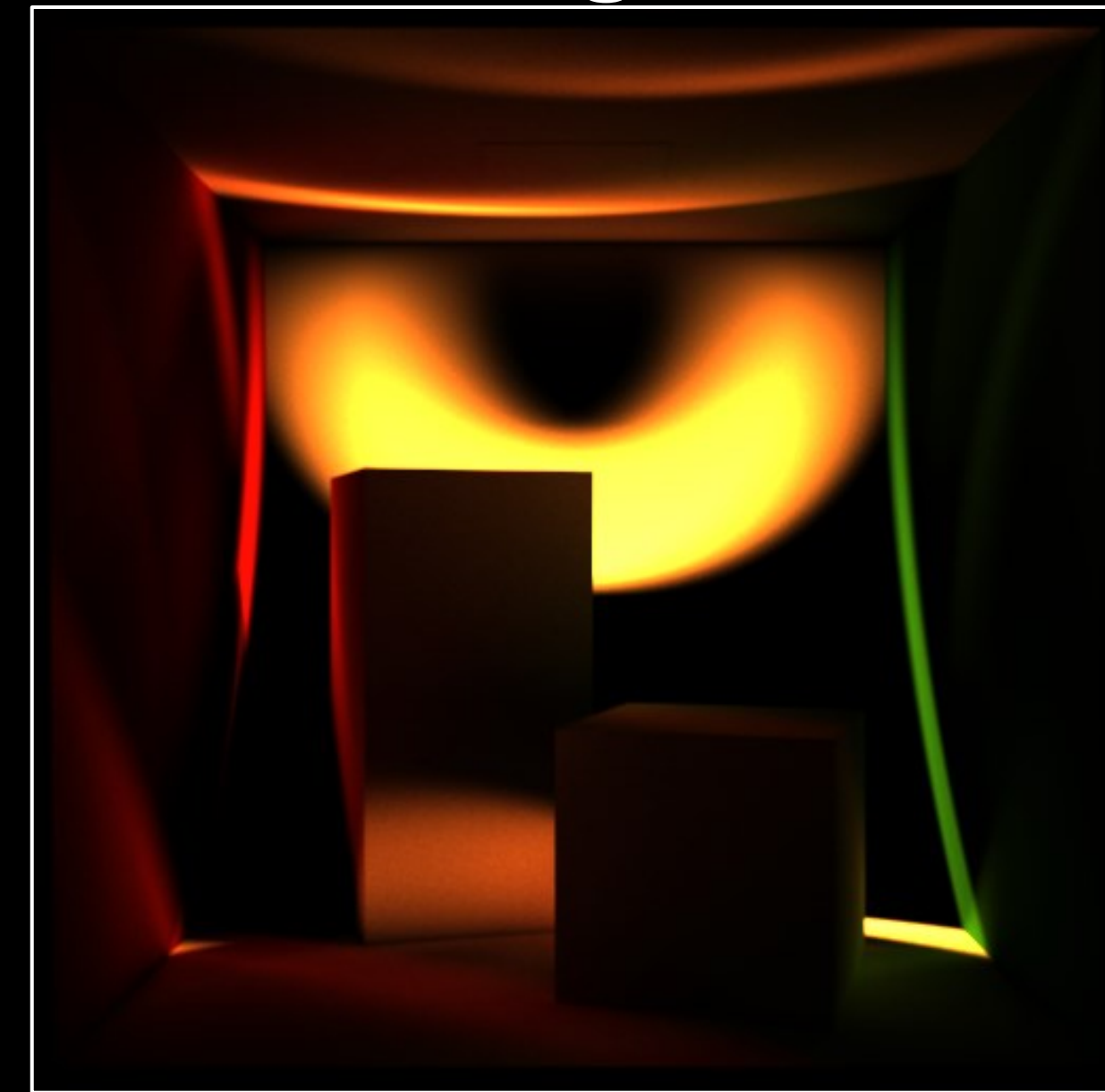
continuous wave



transient



time-gated

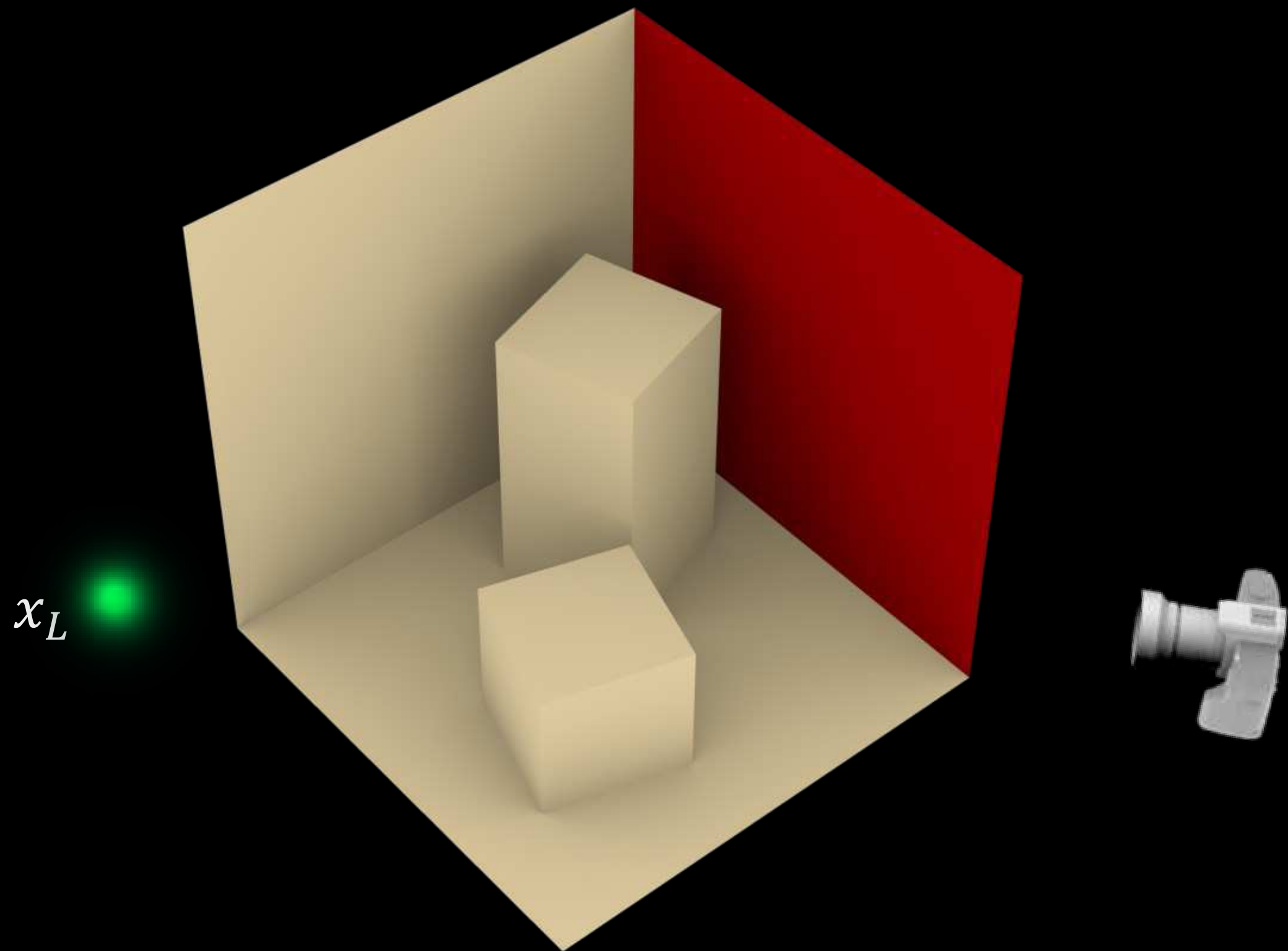


Rendering time-of-flight cameras: Path space integral

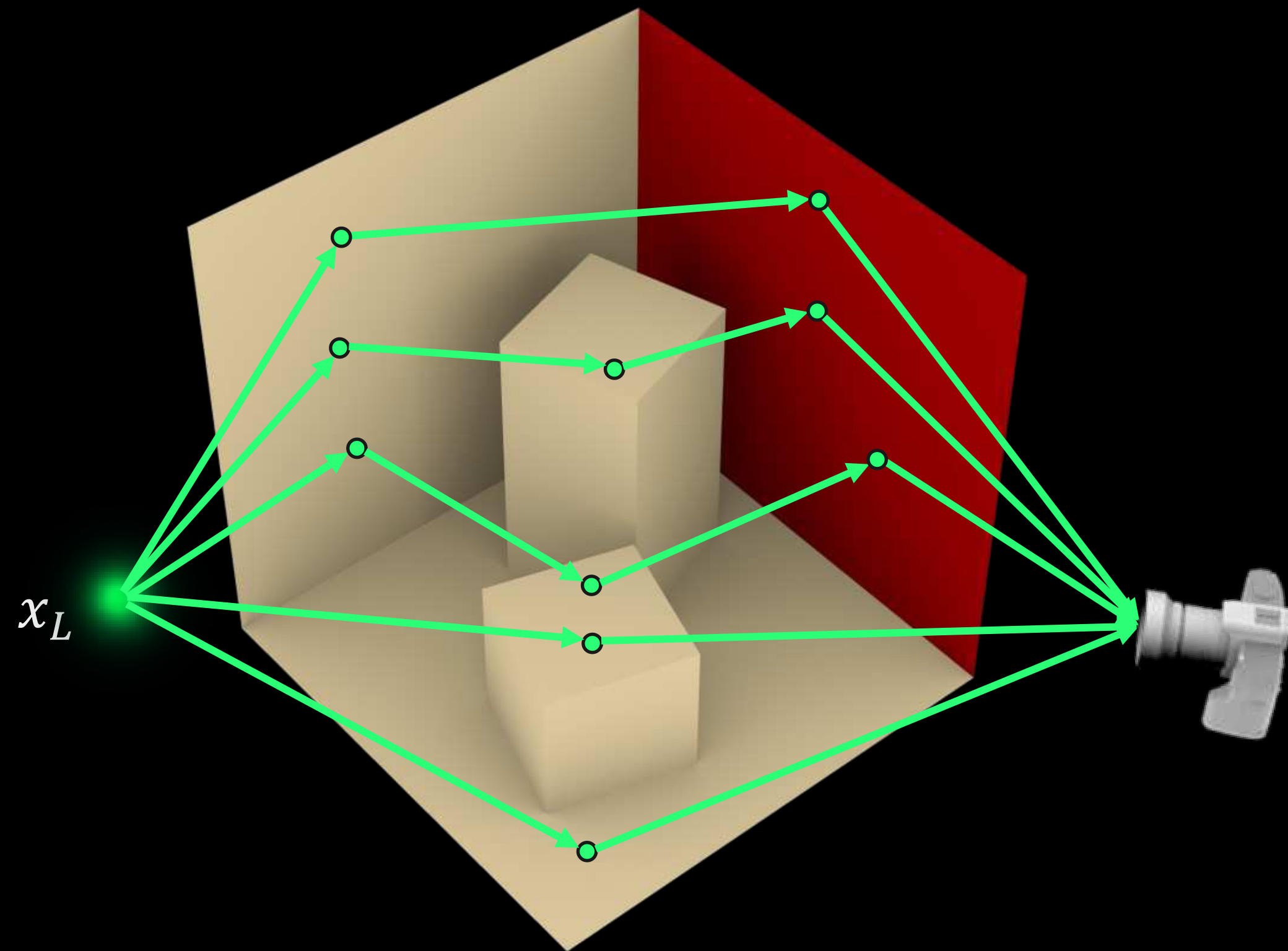
$$\text{image} = \int_{\text{light paths}} f(\text{path})$$

light paths

path contribution, depends on scene properties, light source, and sensor



Rendering time-of-flight cameras: Path space integral



$$\text{image} = \int_{\text{light paths}} f(\text{path})$$

light paths

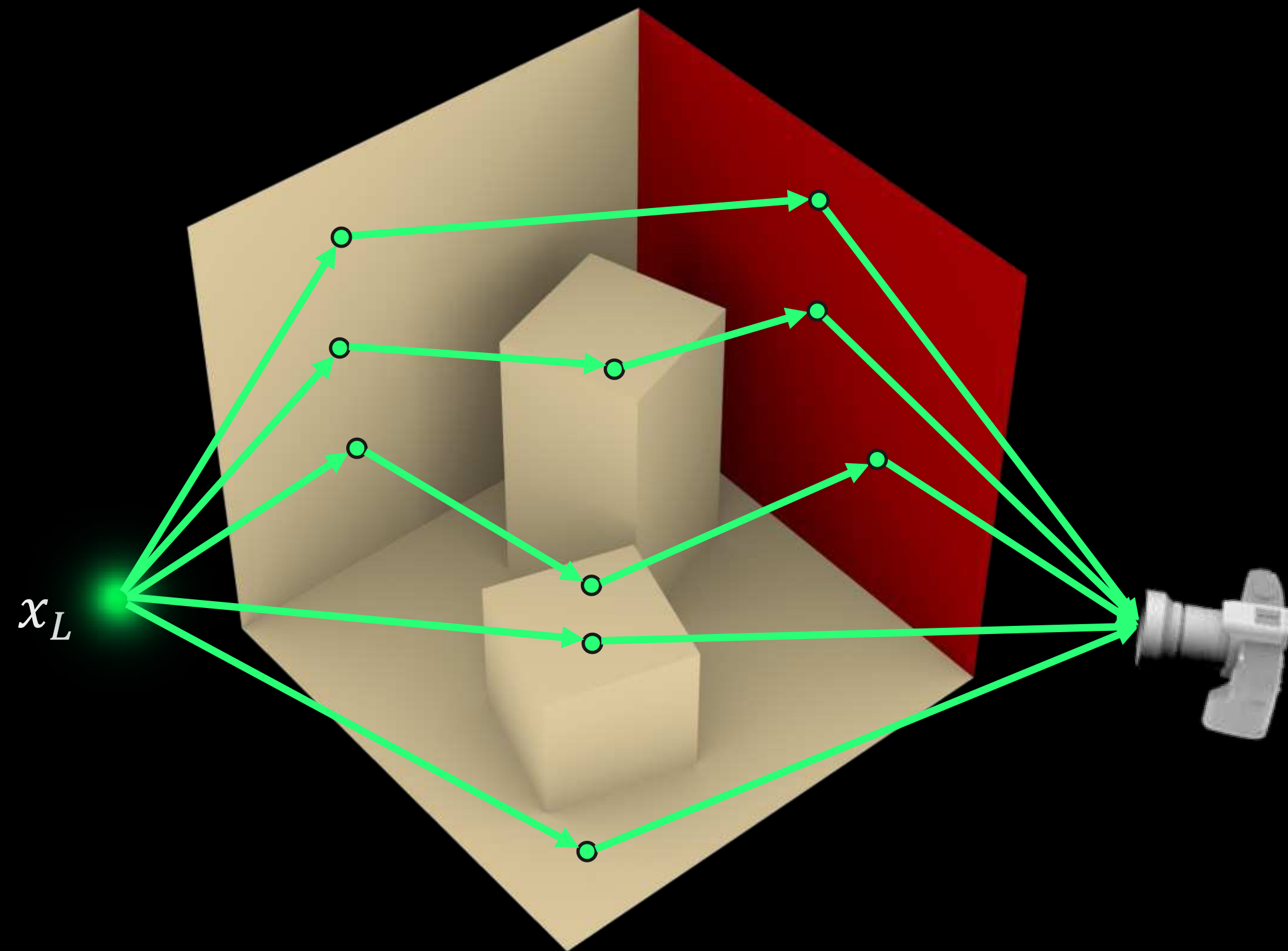
path contribution, depends on scene properties, light source, and sensor

Monte Carlo rendering:

- randomly sample paths: $\text{path}_1, \text{path}_2, \dots, \text{path}_N$
- approximate image as:

$$\text{image} \approx \sum_n \frac{f(\text{path}_n)}{\text{prob}(\text{path}_n)}$$

Rendering time-of-flight cameras: Path space integral



$$\text{image} = \int_{\text{light paths}} f(\text{path}) W(|\text{path}|)$$

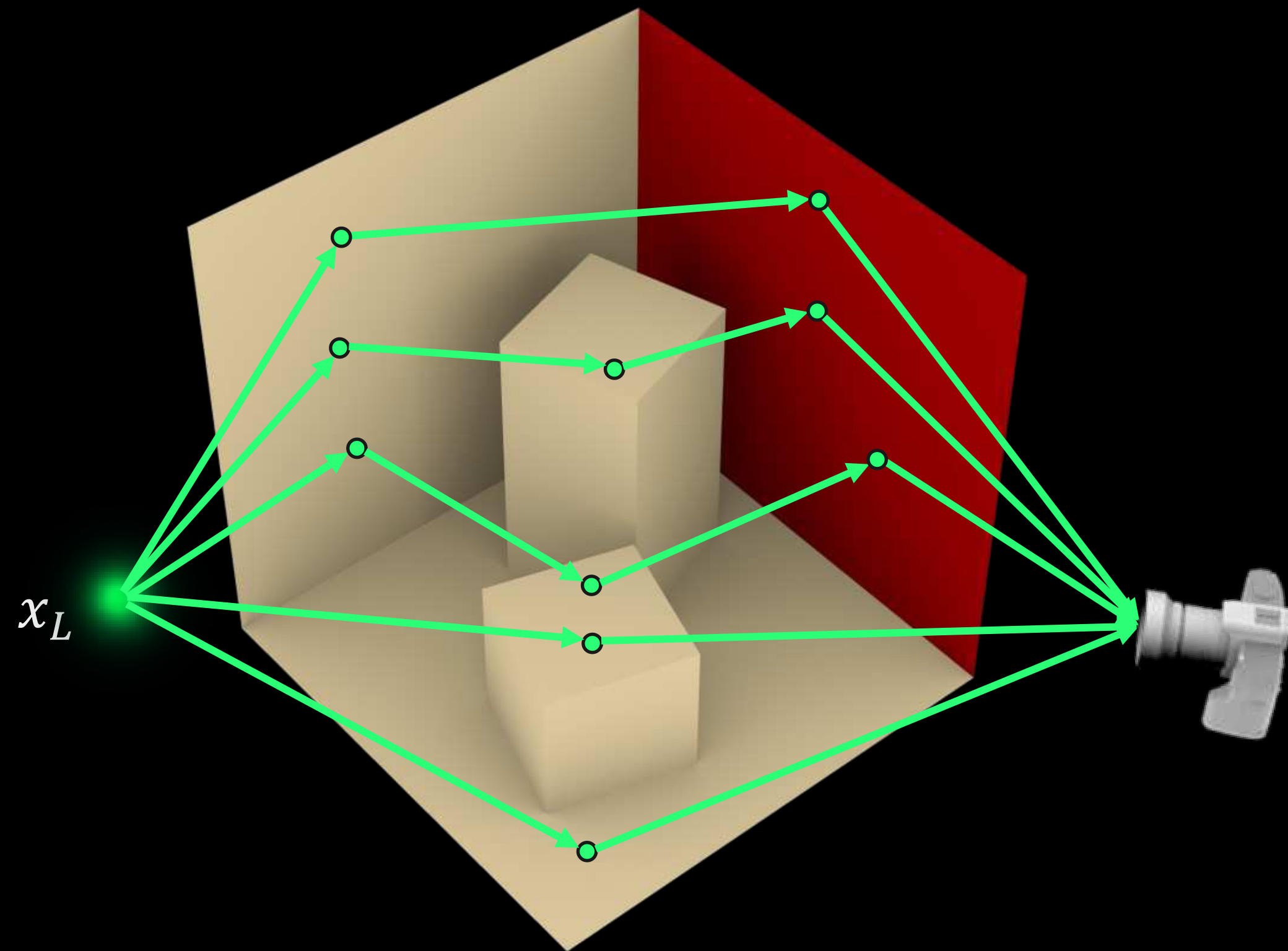
path length weight depends only on sensor

Monte Carlo rendering:

- randomly sample paths: $\text{path}_1, \text{path}_2, \dots, \text{path}_N$
- approximate image as:

$$\text{image} \approx \sum_n \frac{f(\text{path}_n)}{\text{prob}(\text{path}_n)}$$

Rendering time-of-flight cameras: Path space integral



$$\text{image} = \int_{\text{light paths}} f(\text{path}) W(|\text{path}|)$$

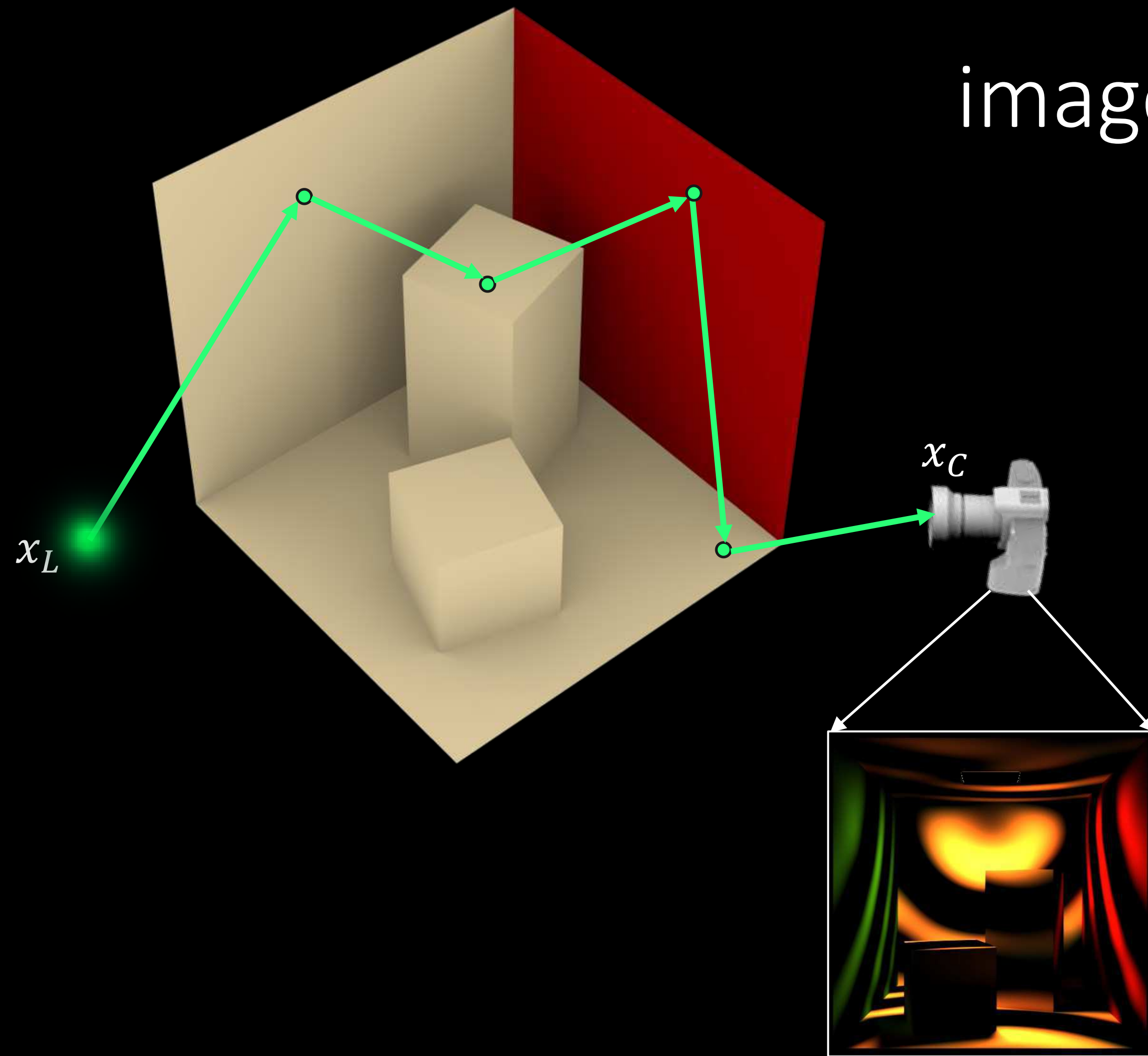
path length weight depends only on sensor

Monte Carlo rendering:

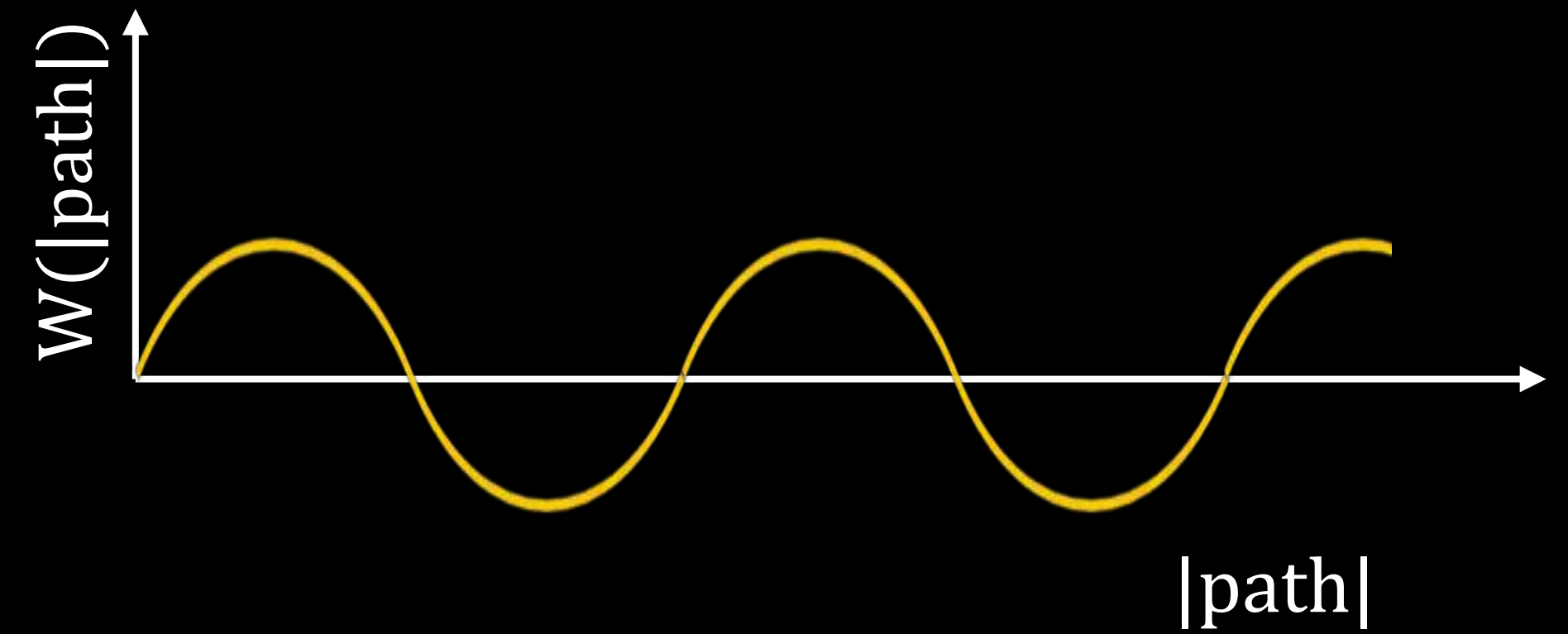
- randomly sample paths: $\text{path}_1, \text{path}_2, \dots, \text{path}_N$
- approximate image as:

$$\text{time-of-flight image} \approx \sum_n \frac{f(\text{path}_n)}{\text{prob}(\text{path}_n)} W(|\text{path}_n|)$$

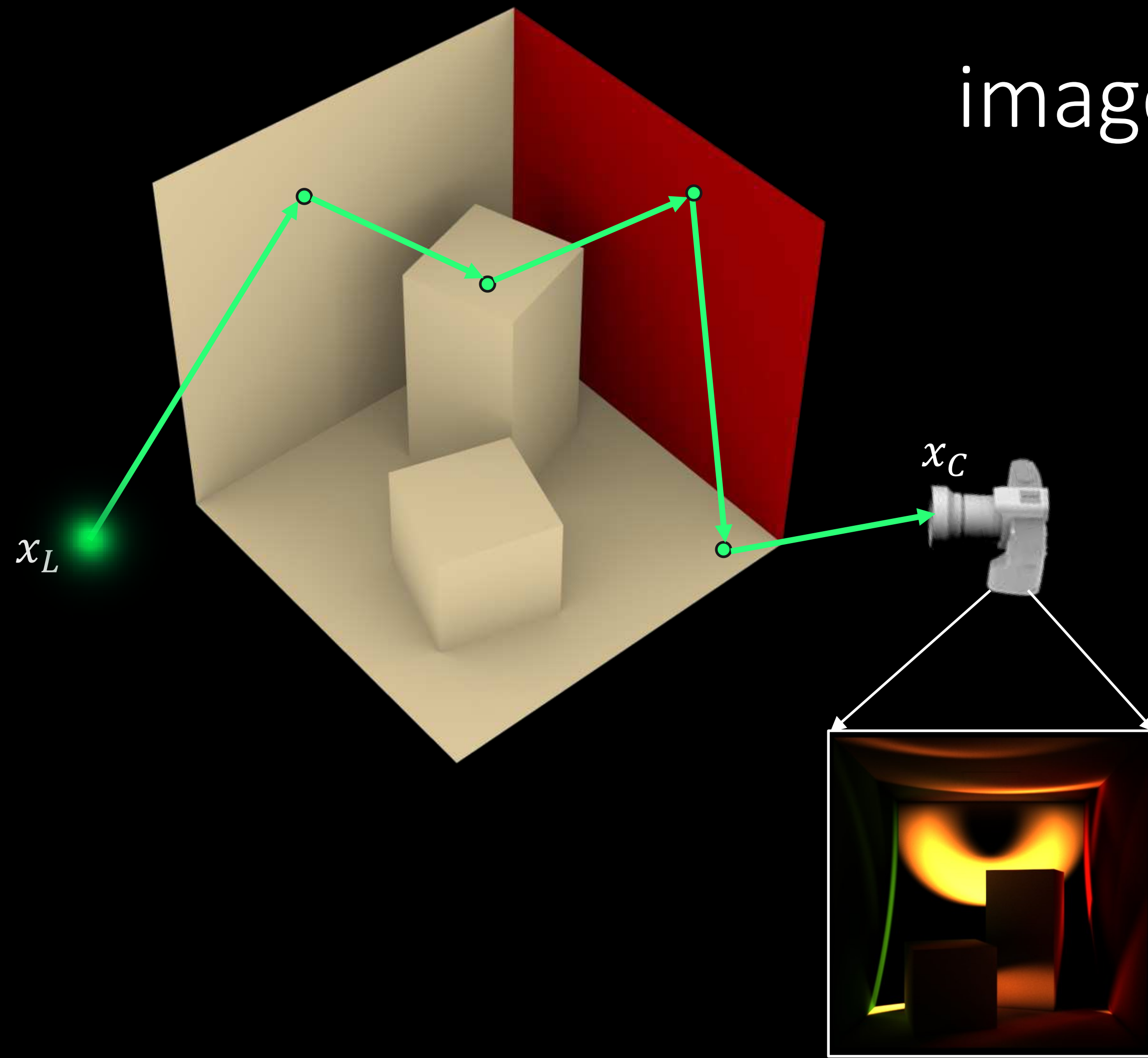
Rendering continuous wave time-of-flight camera



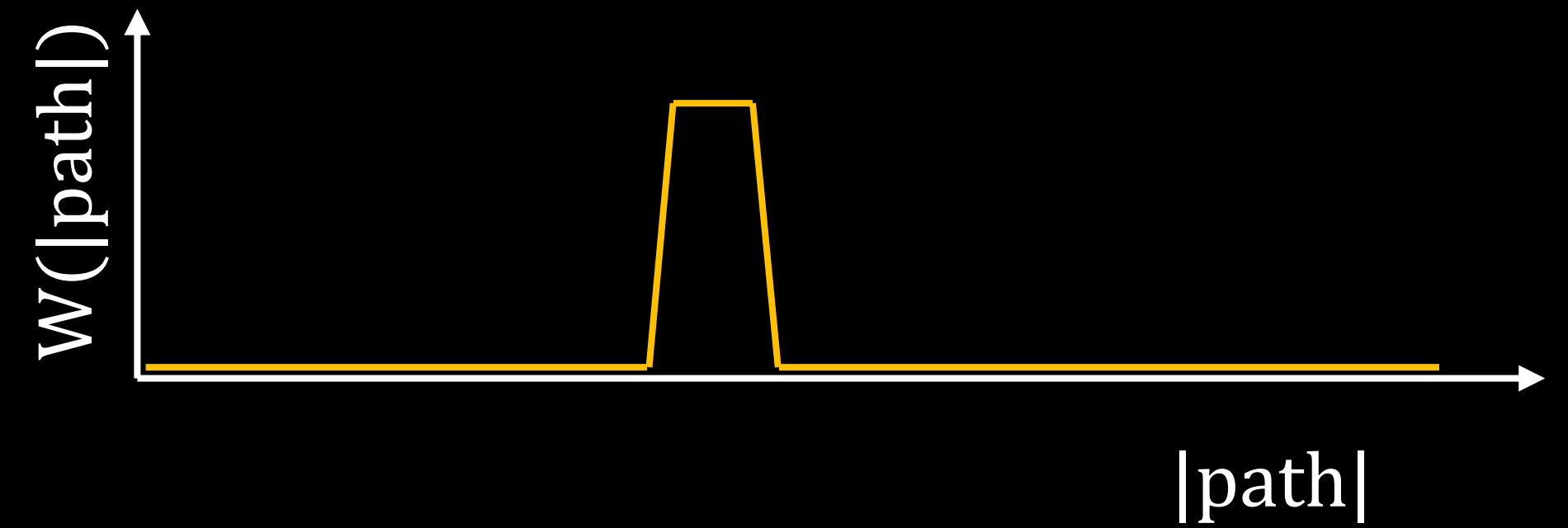
$$\text{image} = \int_{\text{light paths}} f(\text{path}) W(|\text{path}|)$$



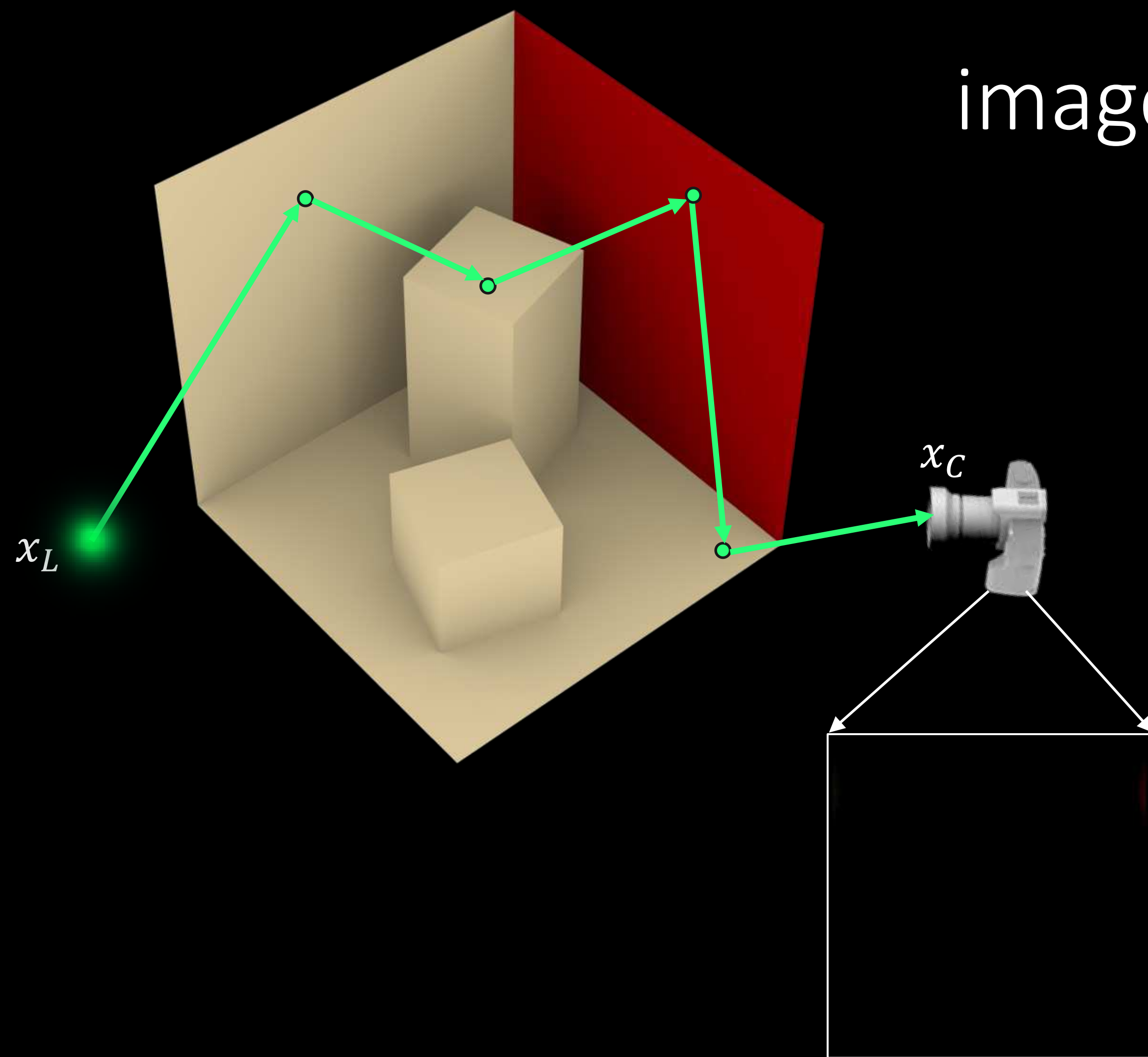
Classification: time-gated camera



$$\text{image} = \int_{\text{light paths}} f(\text{path}) W(|\text{path}|)$$



Rendering transient camera



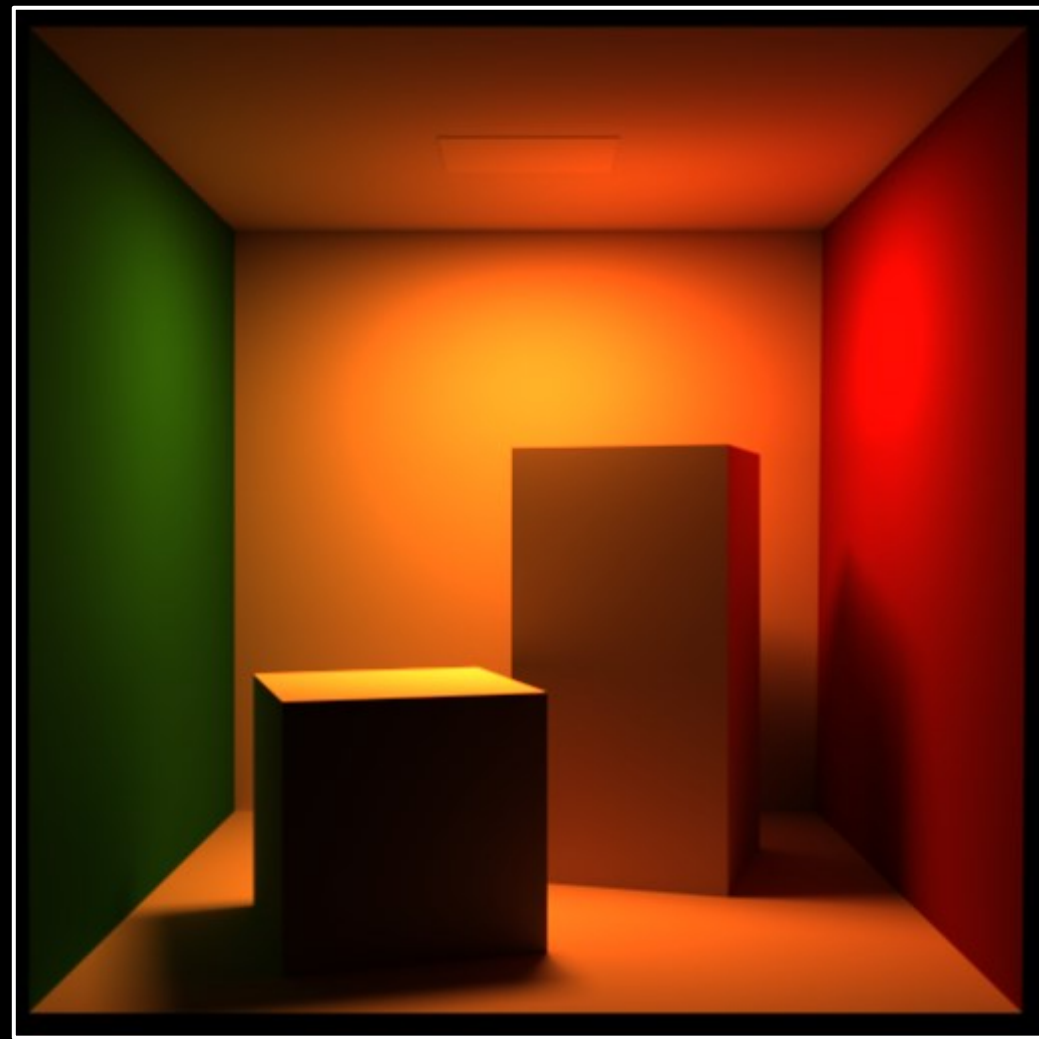
$$\text{image} = \int_{\text{light paths}} f(\text{path}) W(|\text{path}|)$$



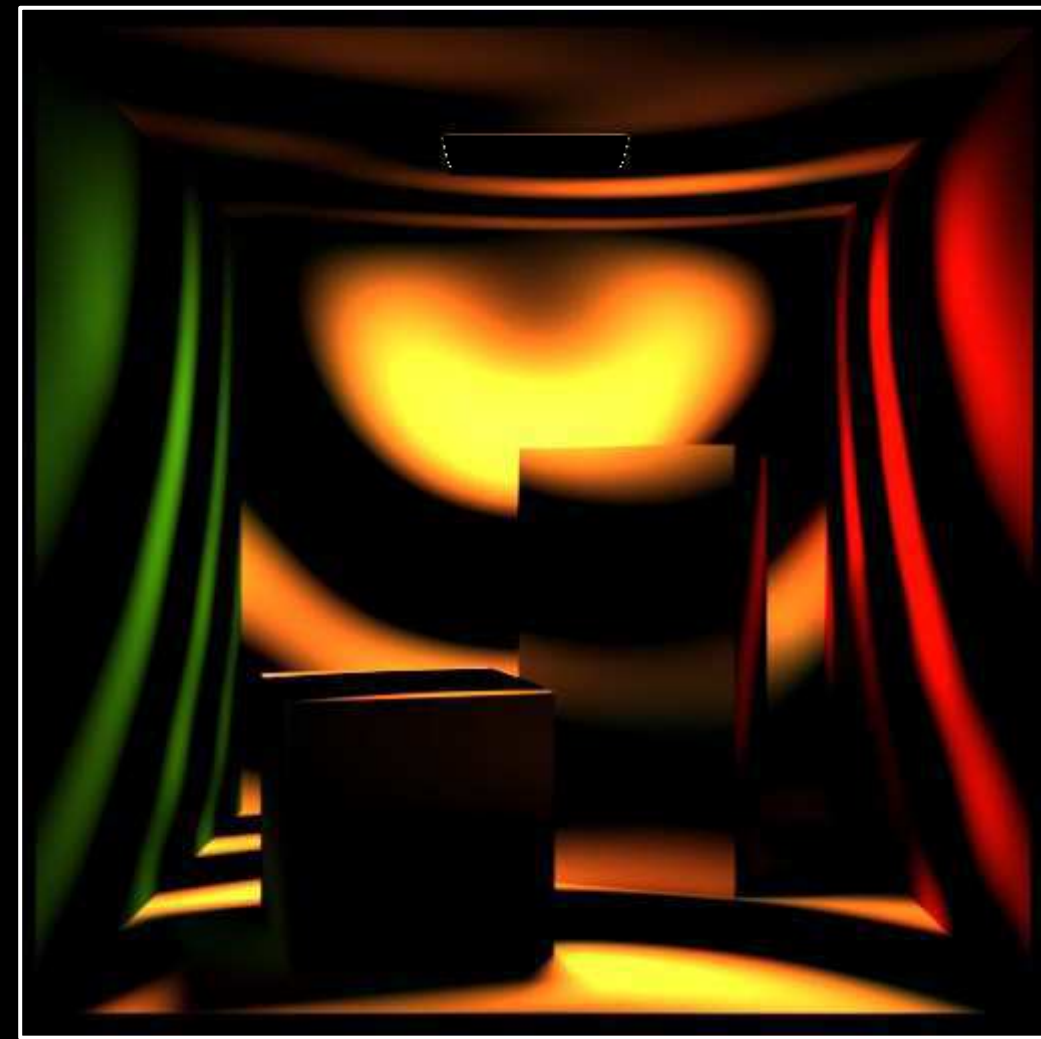
Path space integral for time-of-flight cameras

$$\text{image} = \int_{\text{light paths}} f(\text{path})W(|\text{path}|)$$

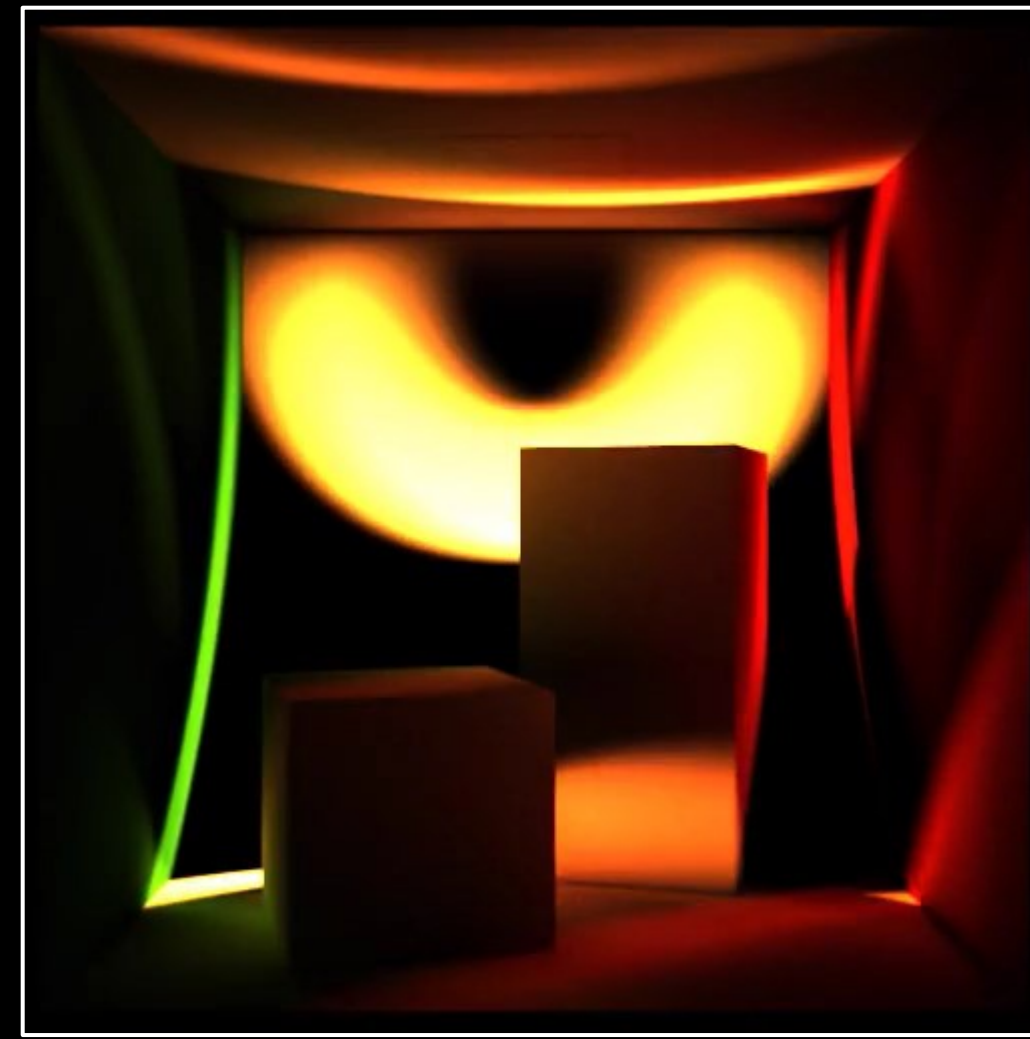
intensity



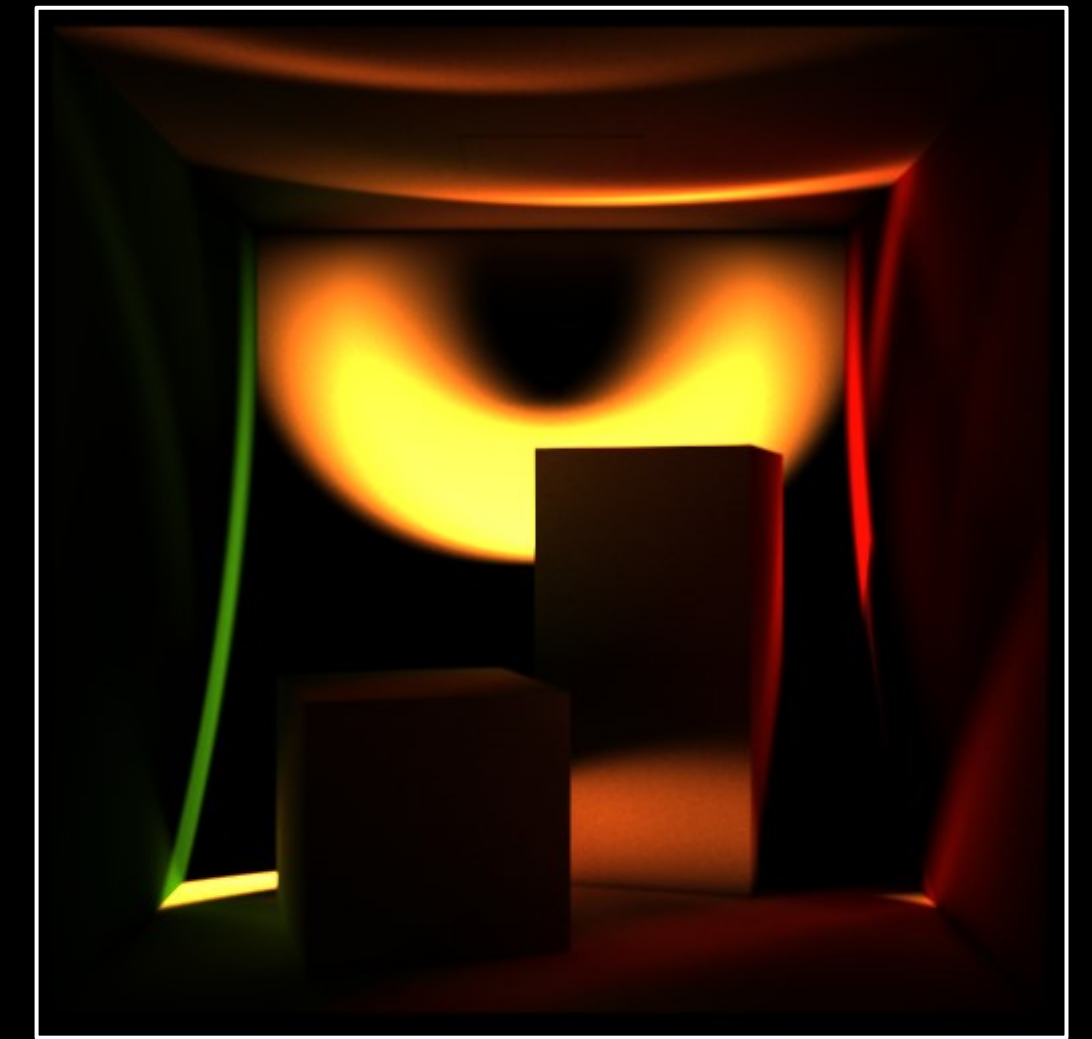
continuous-wave



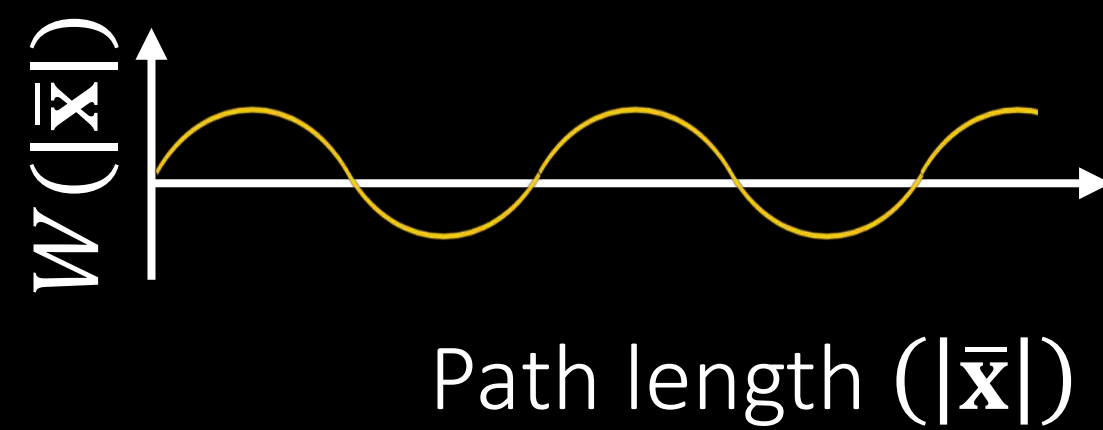
transient



time-gated



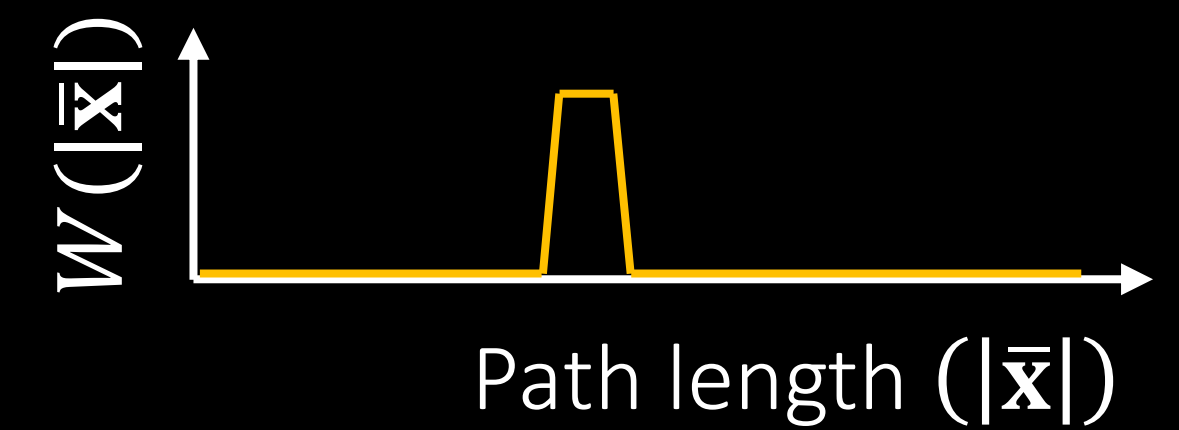
BDPT, PT, PM, KDE, etc



BDPT, PT, PM, KDE, etc

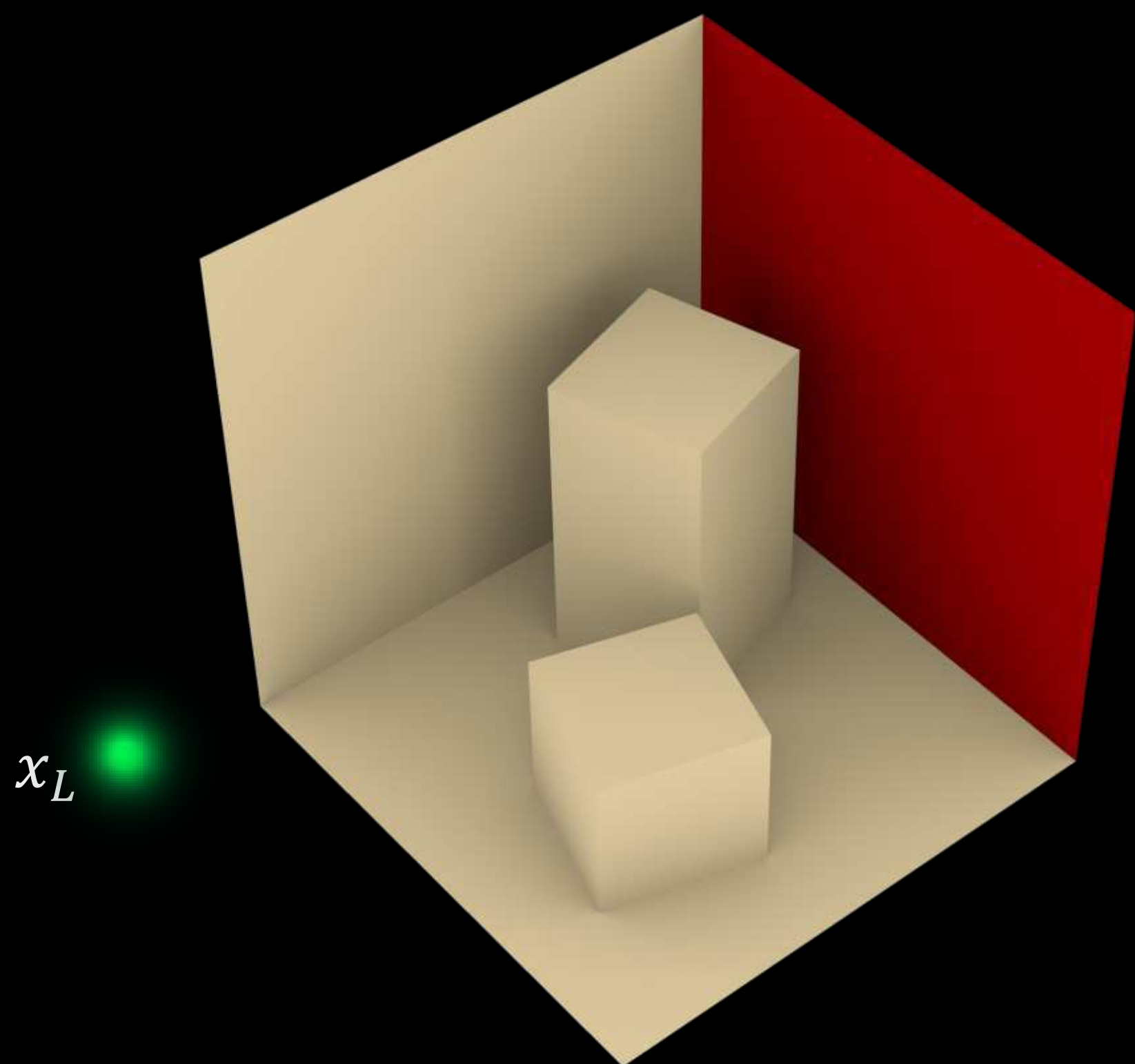


BDPT, PT, PM, KDE, etc
[Jarabo et al., 2014, 2017]
[Marco et al. 2017, 2018]

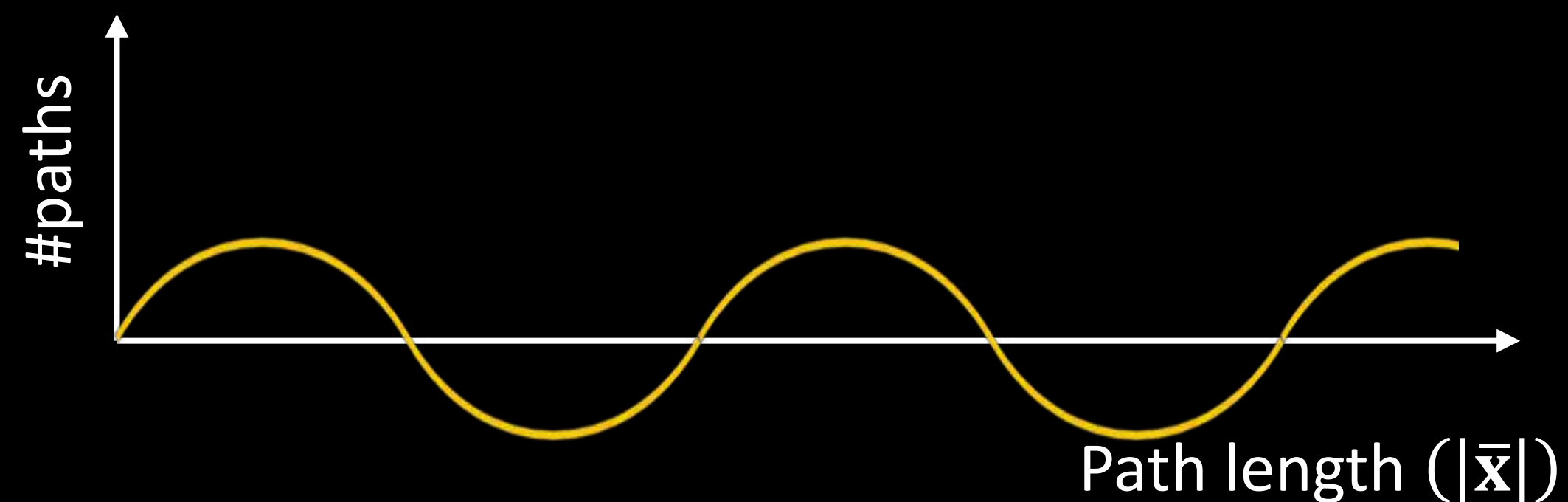


no efficient renderer

Path sampling for time-gated rendering is challenging



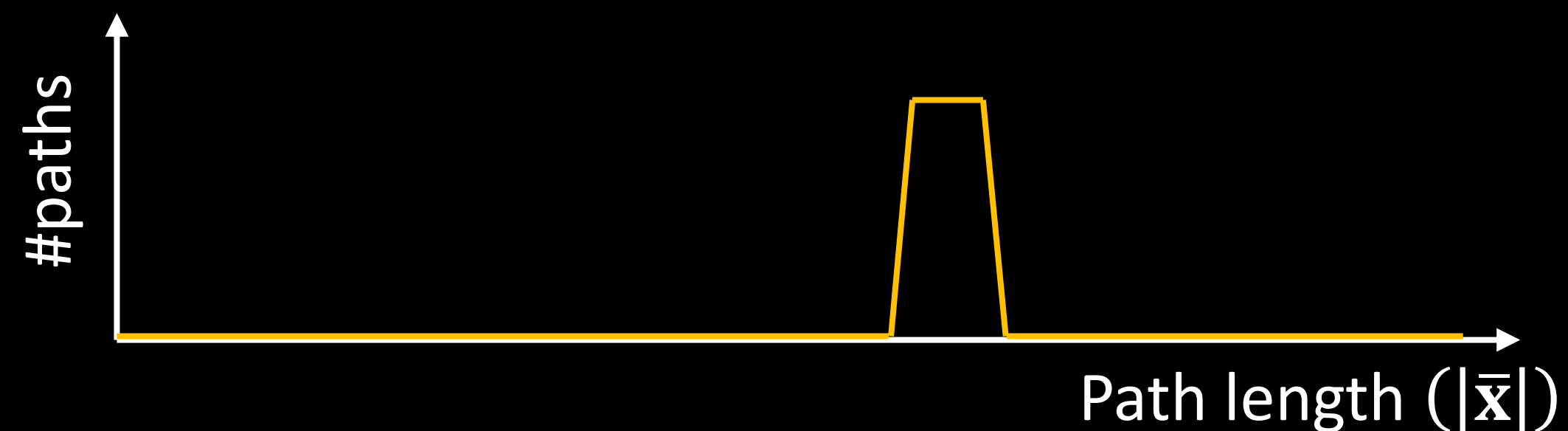
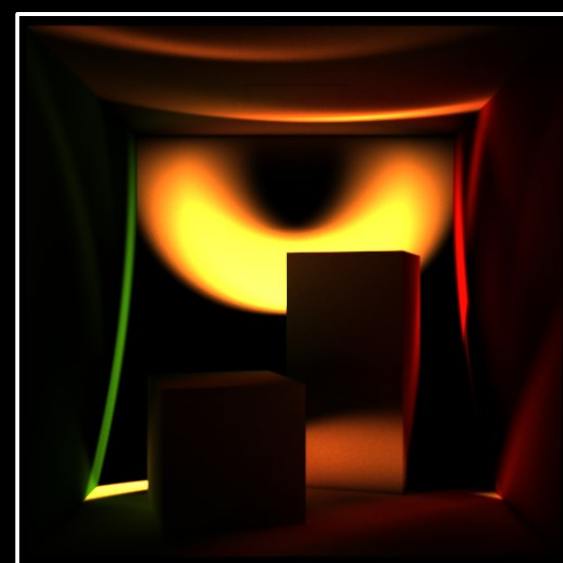
continuous-wave



transient

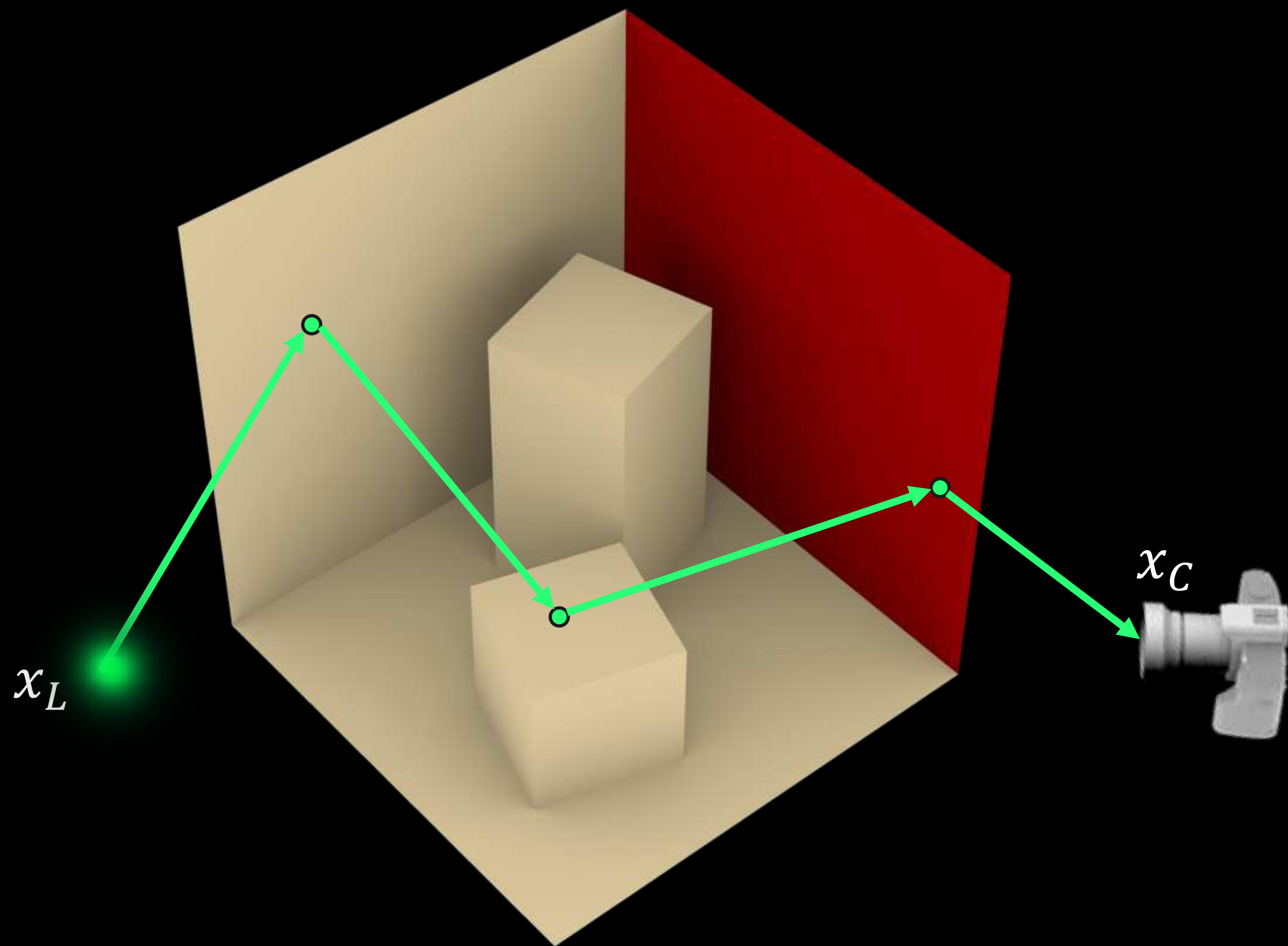


time-gated

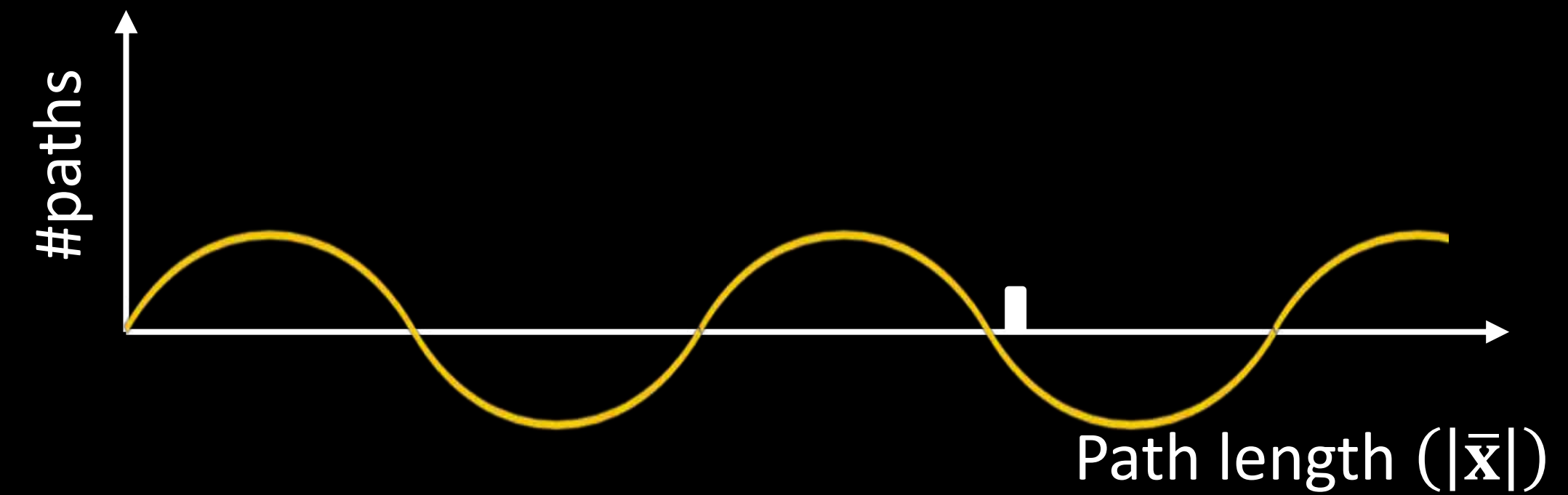


No control over path length

Path sampling for time-gated rendering is challenging



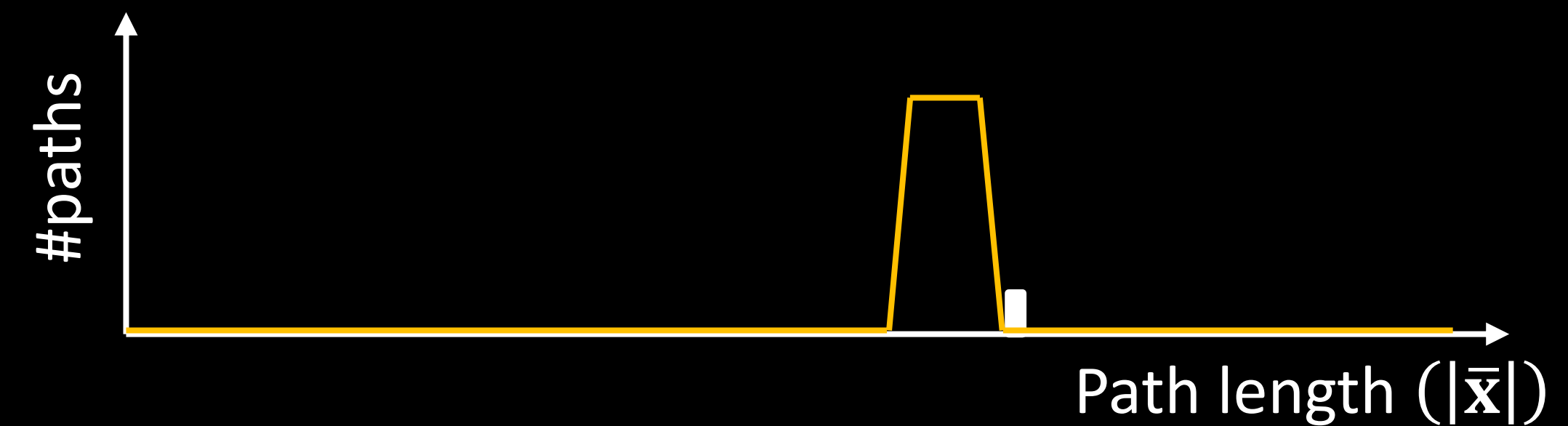
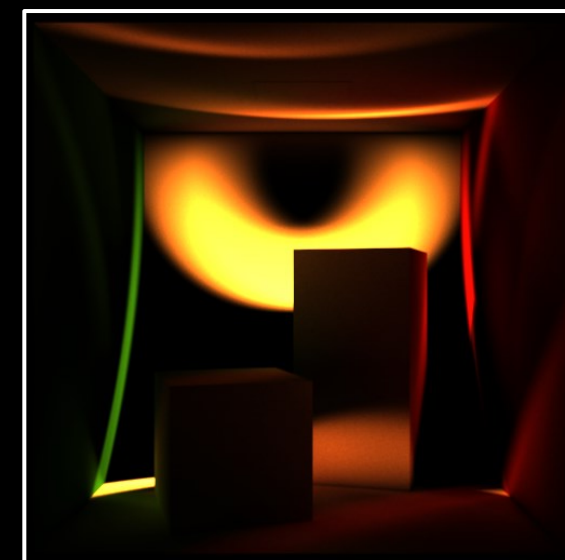
continuous-wave



transient

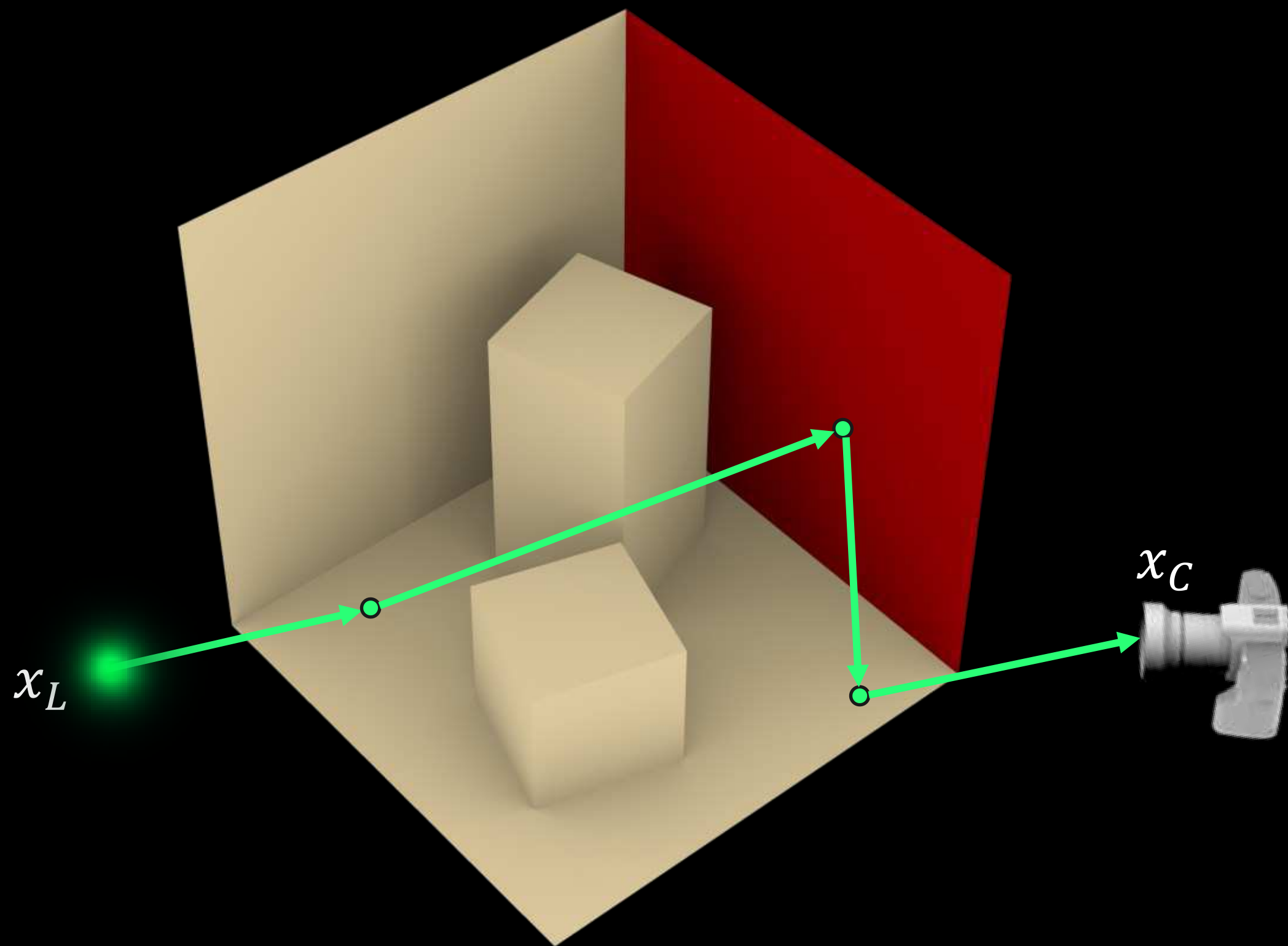


time-gated

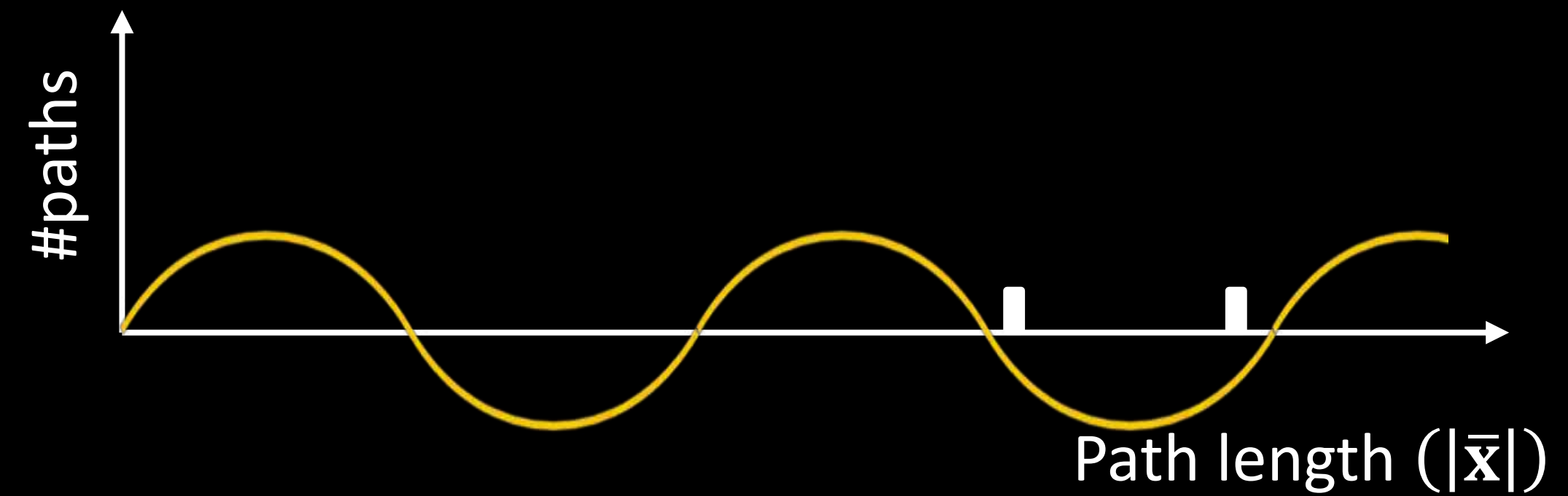
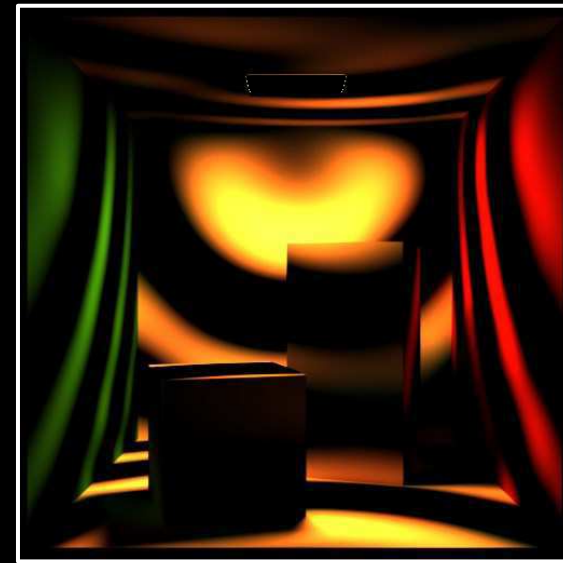


No control over path length

Path sampling for time-gated rendering is challenging



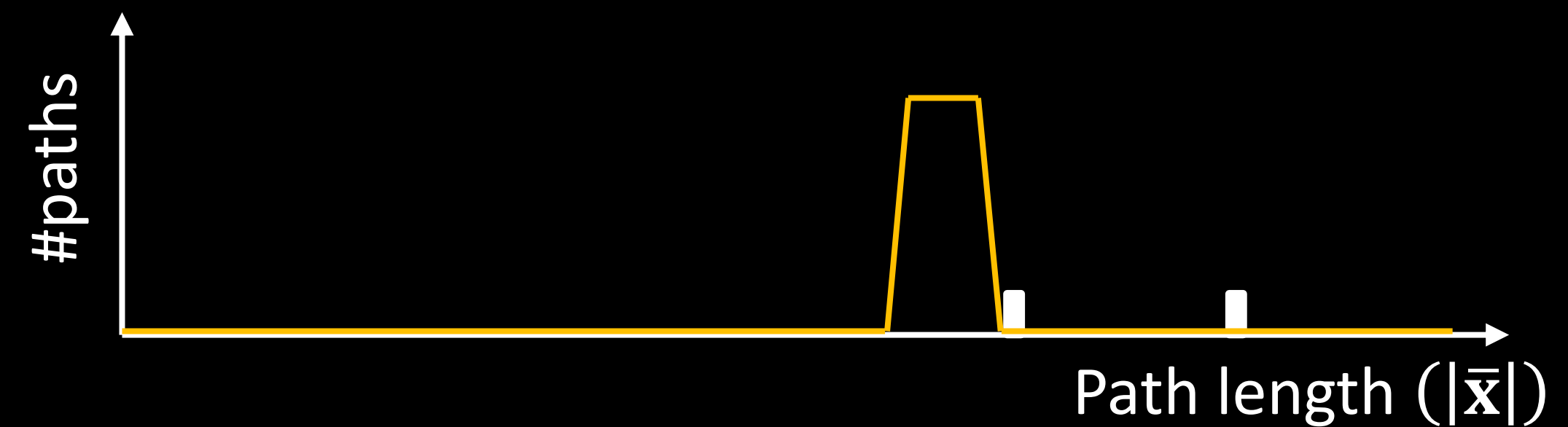
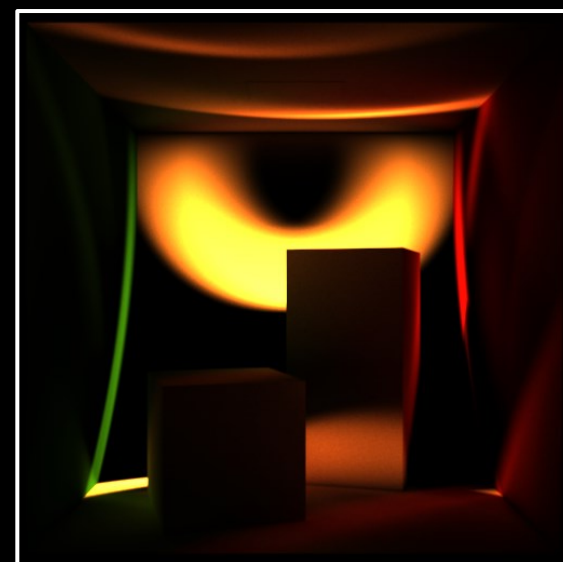
continuous-wave



transient

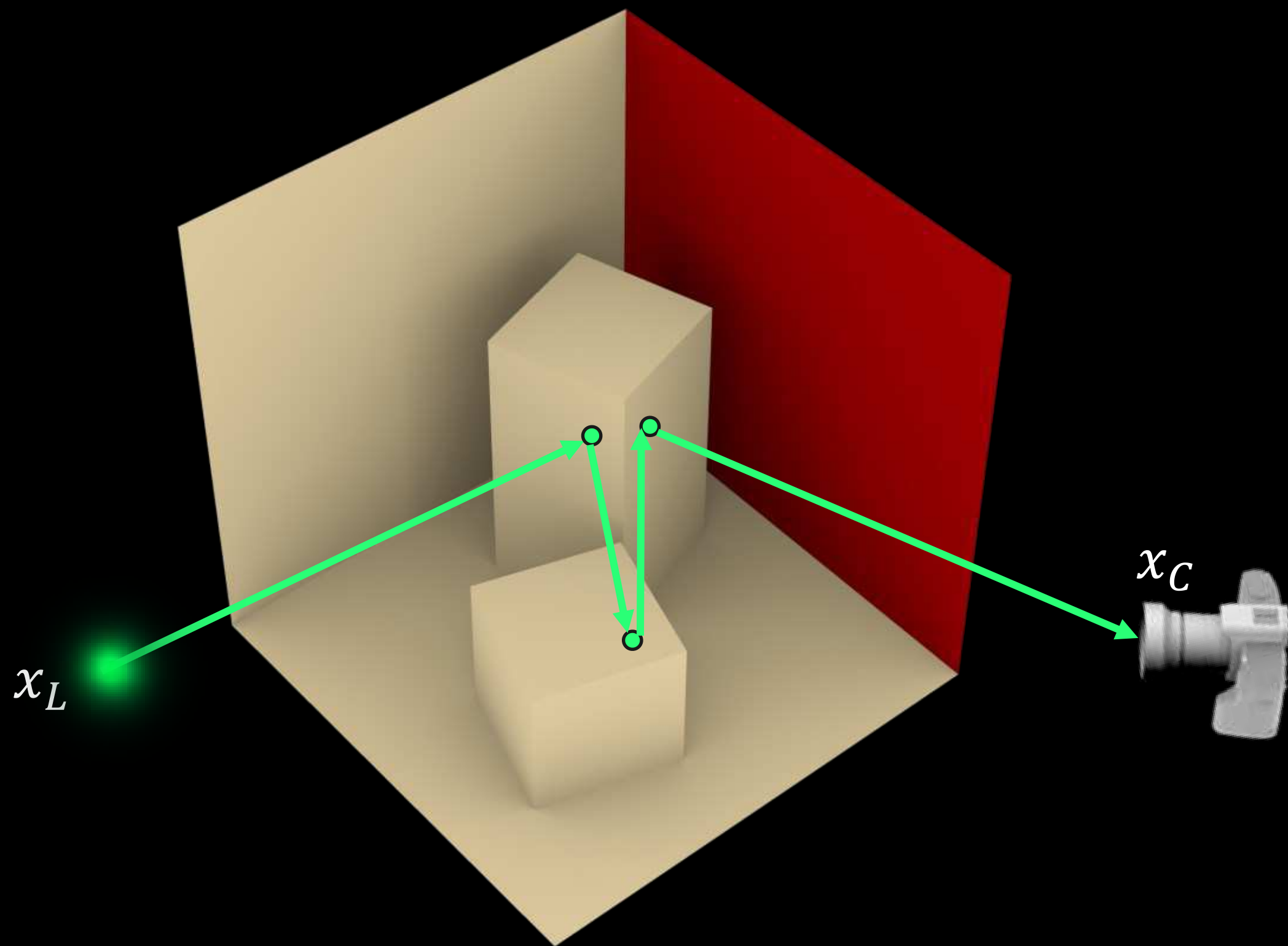


time-gated

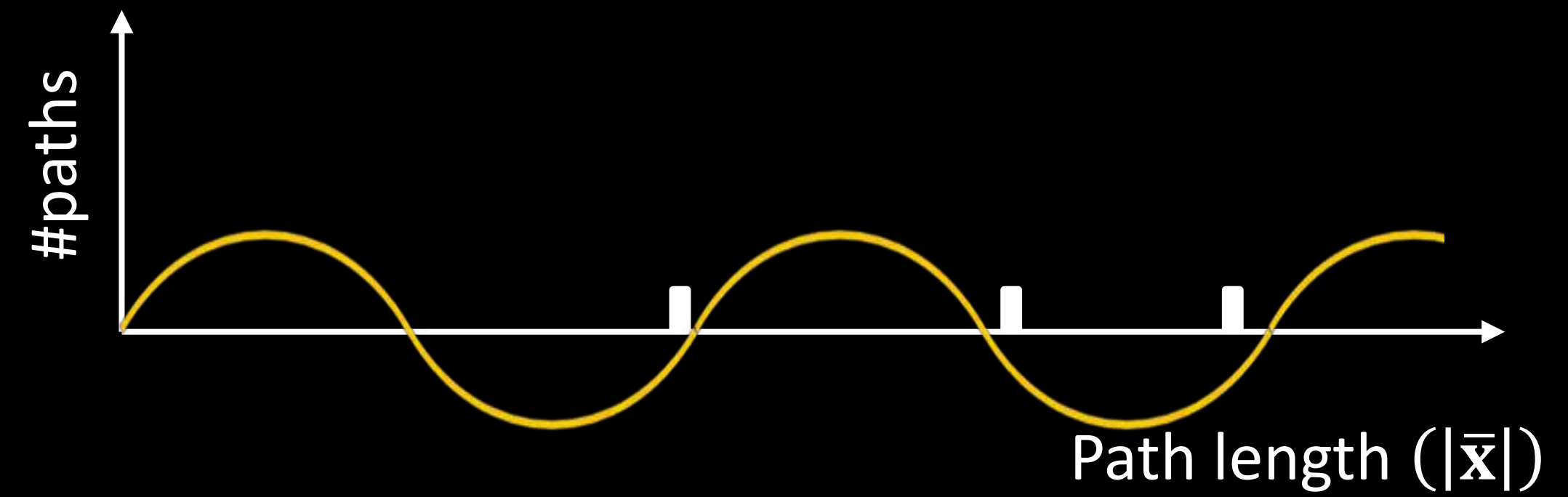
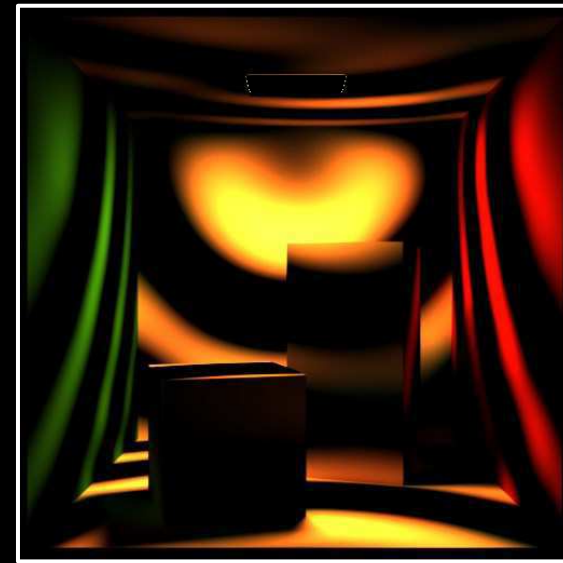


No control over path length

Path sampling for time-gated rendering is challenging



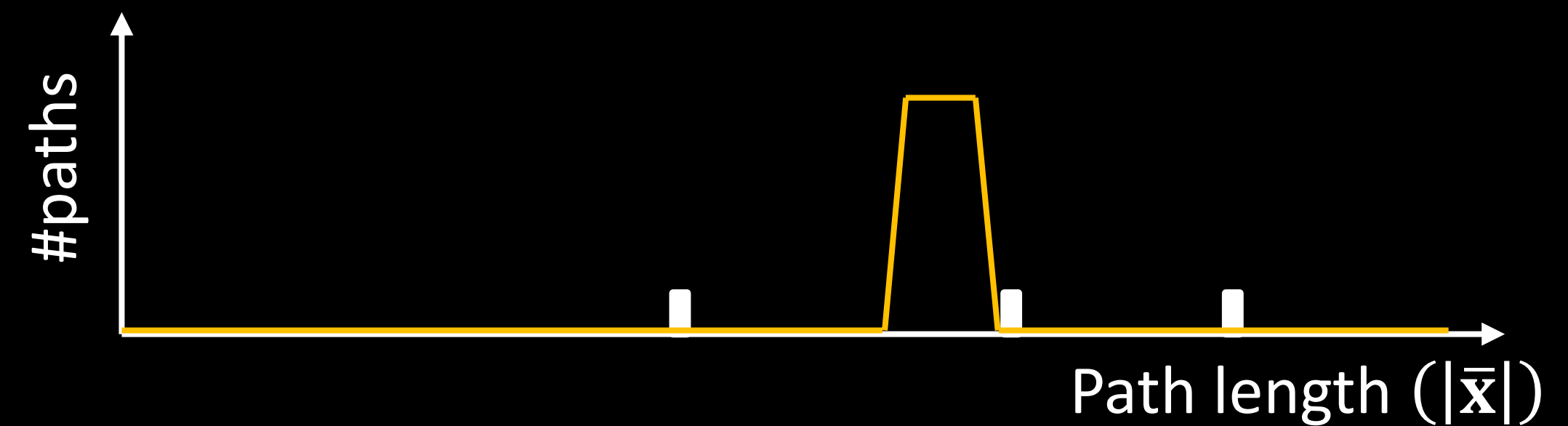
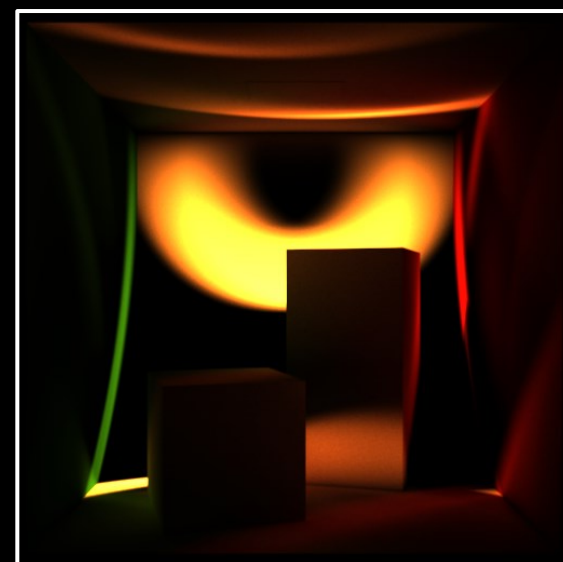
continuous-wave



transient

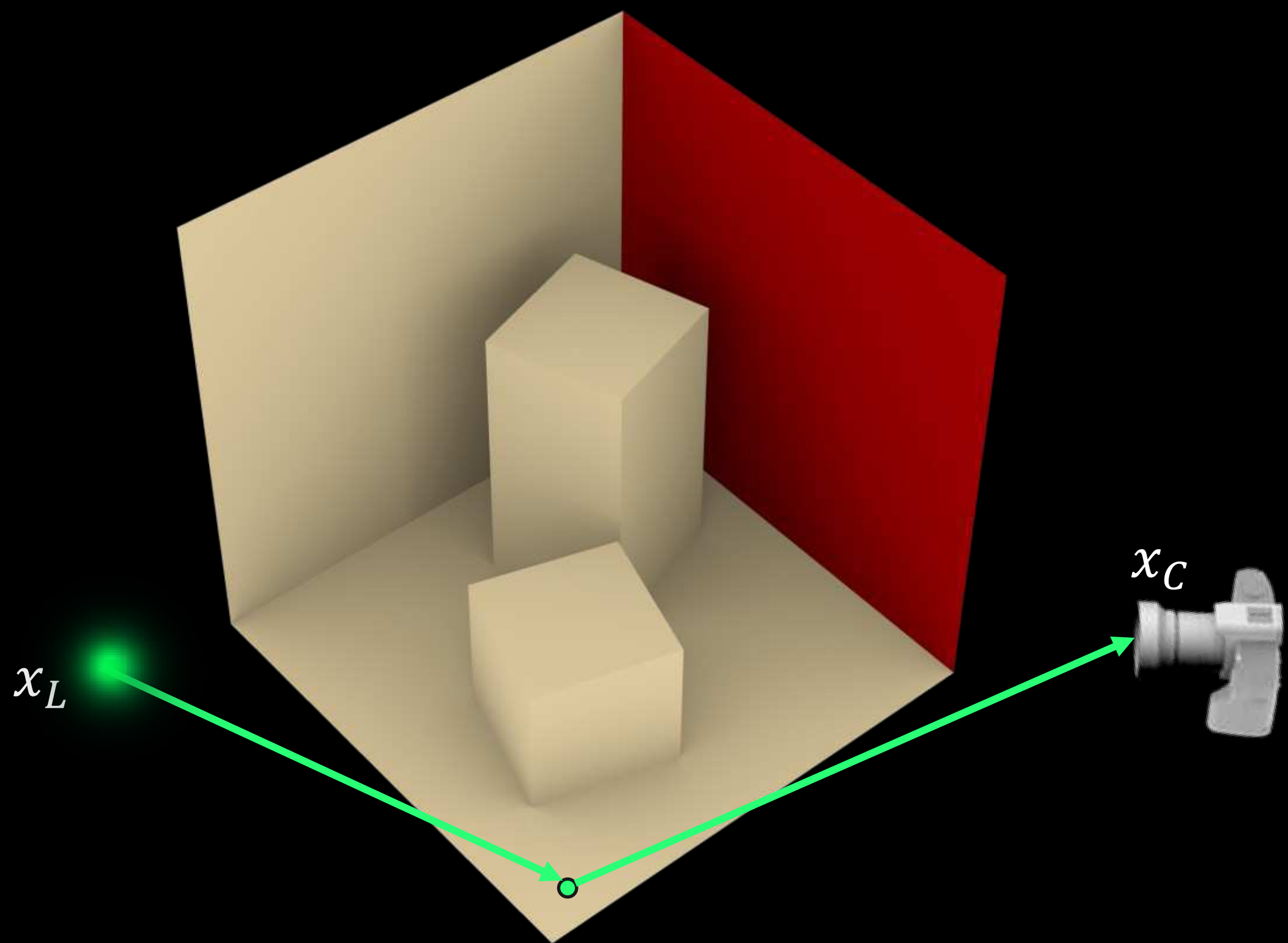


time-gated

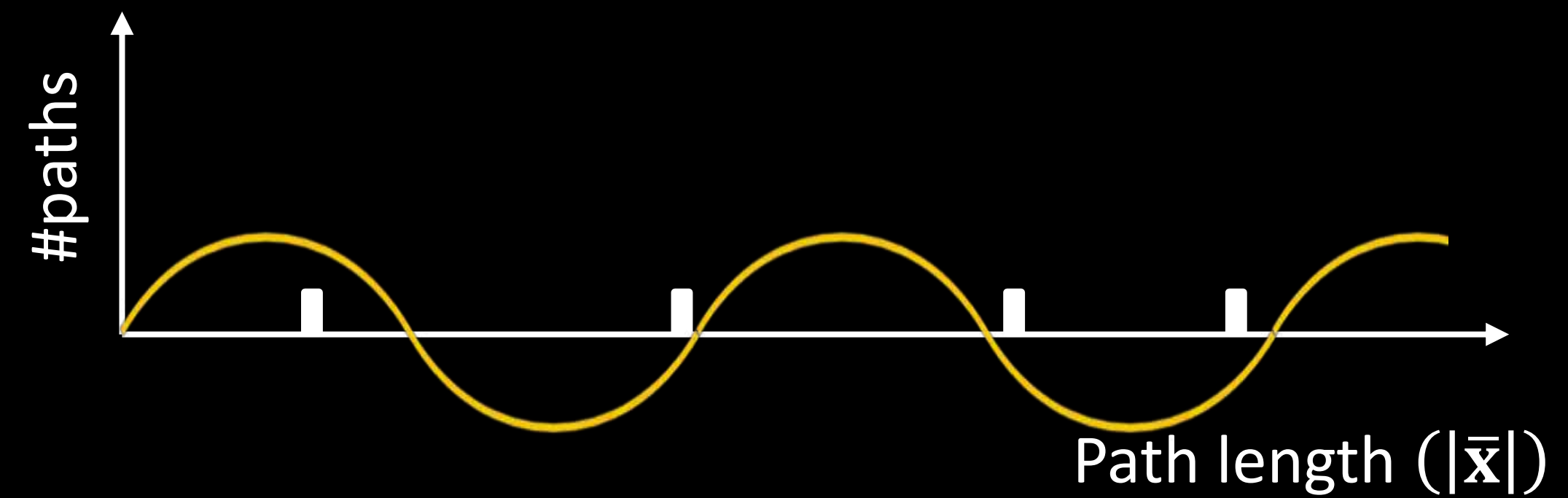
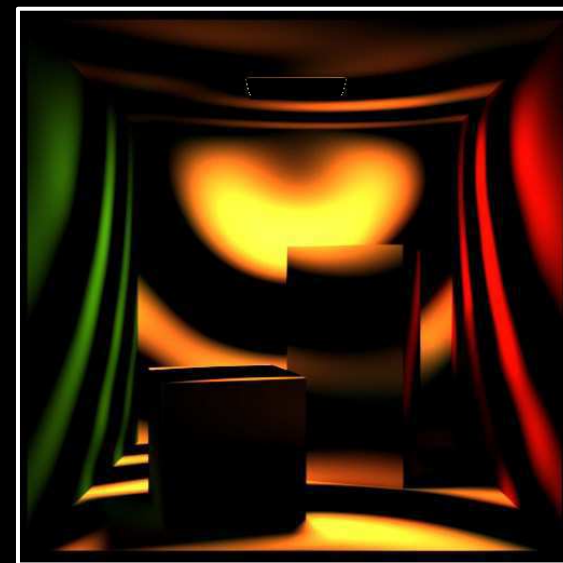


No control over path length

Path sampling for time-gated rendering is challenging



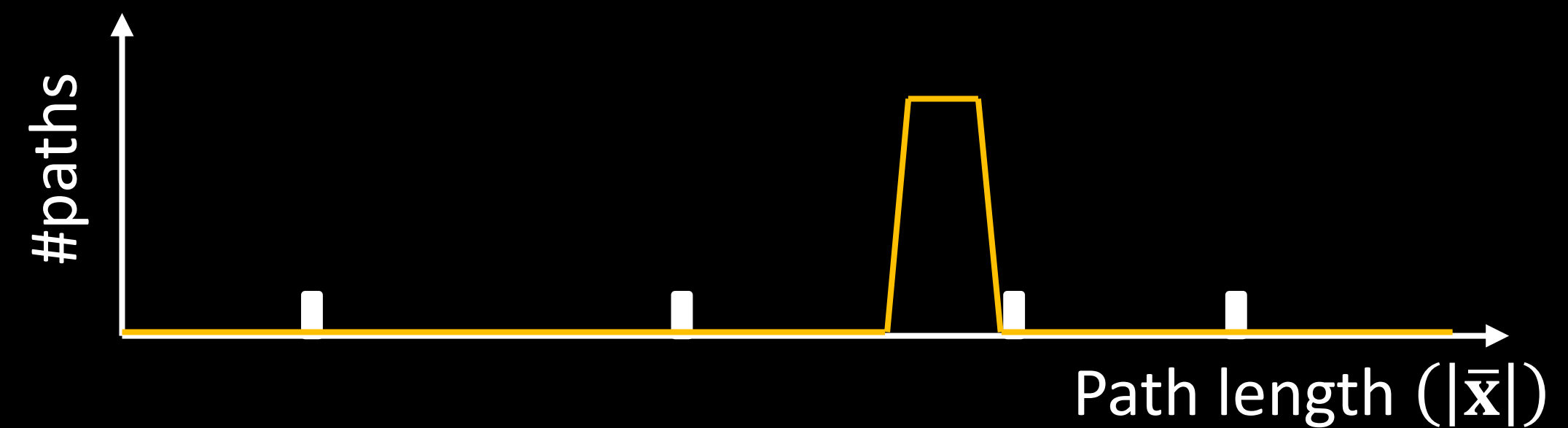
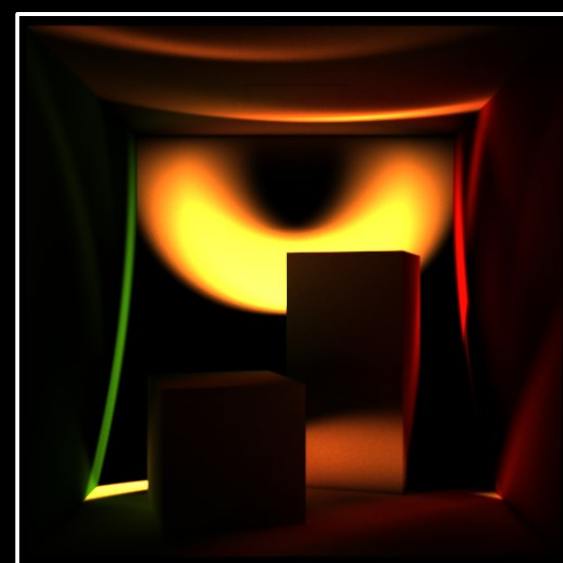
continuous-wave



transient

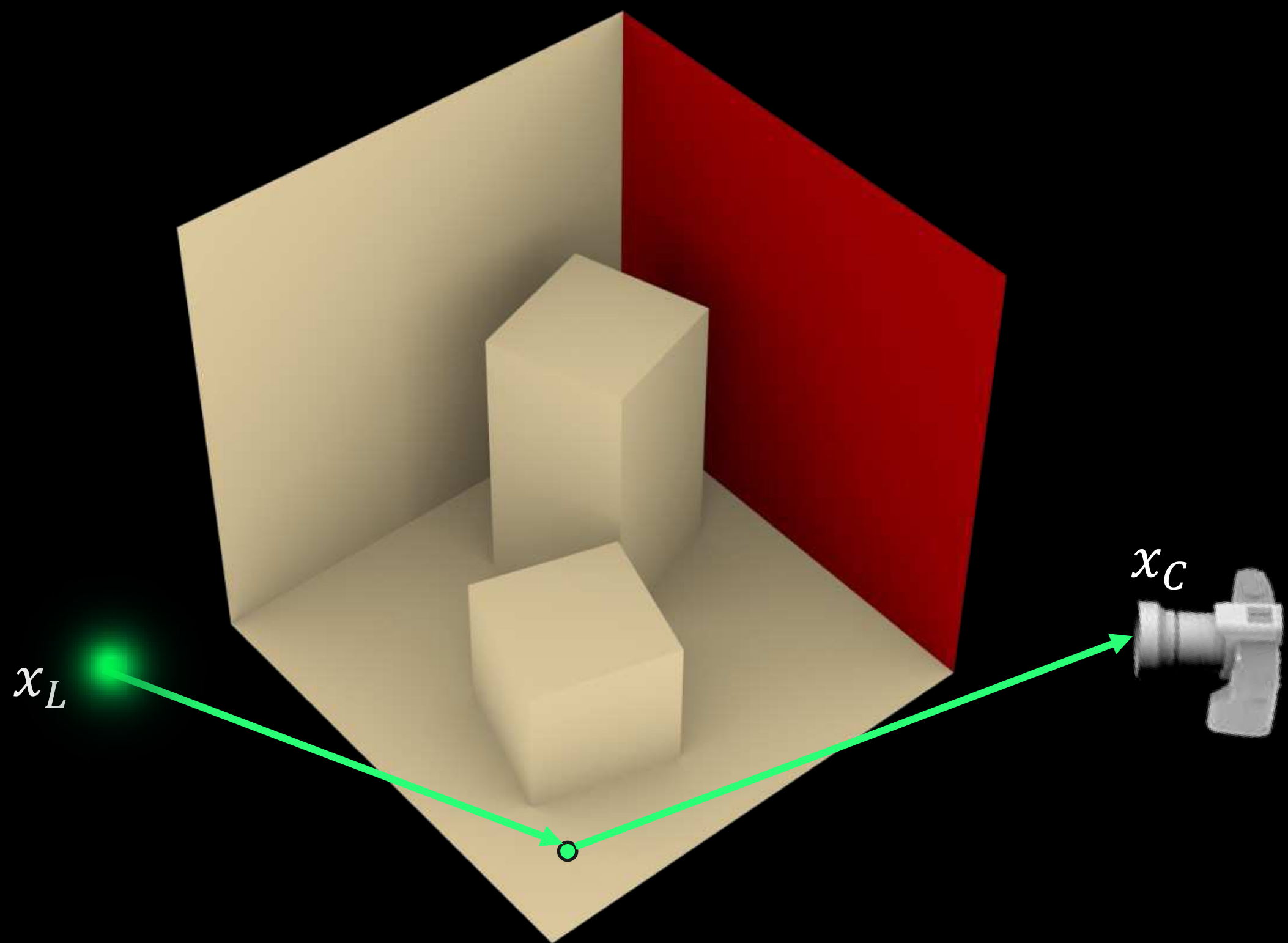


time-gated

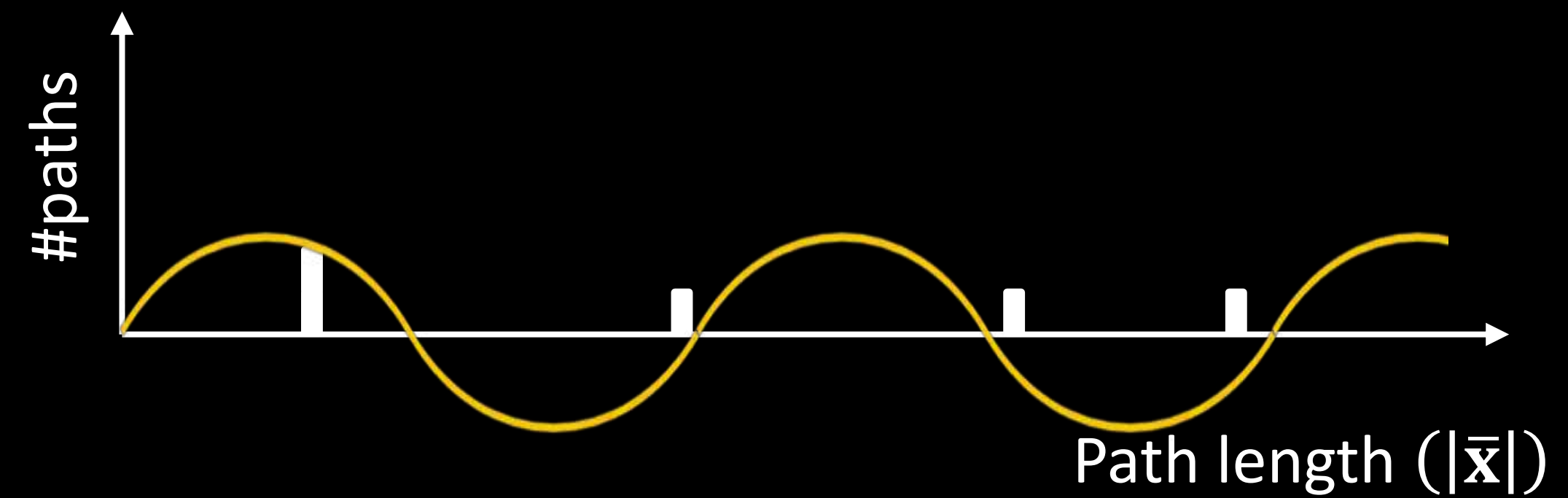
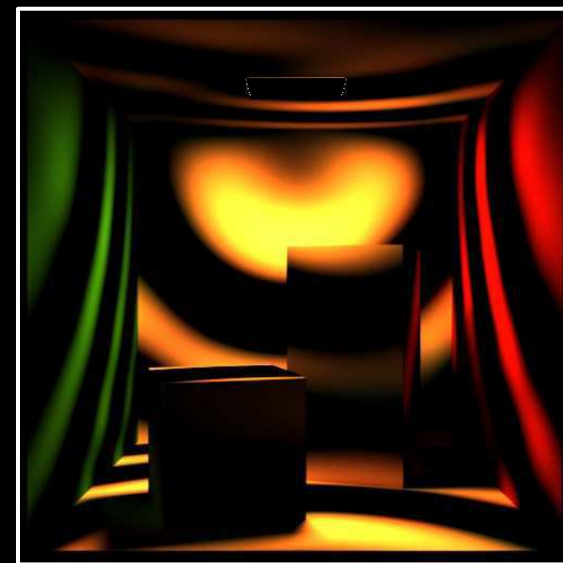


No control over path length

Path sampling for time-gated rendering is challenging



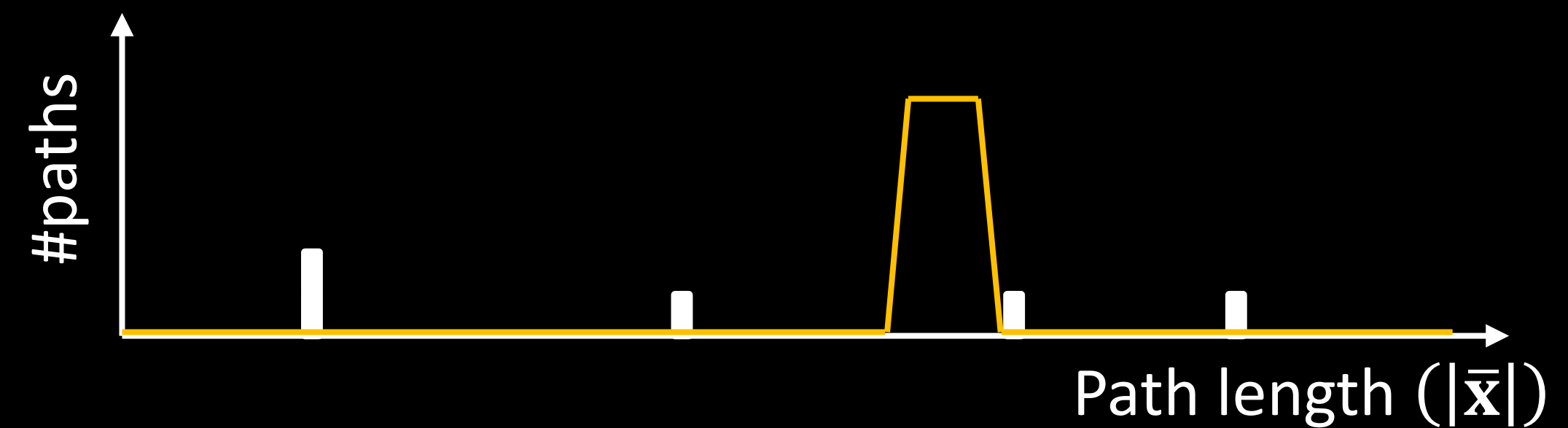
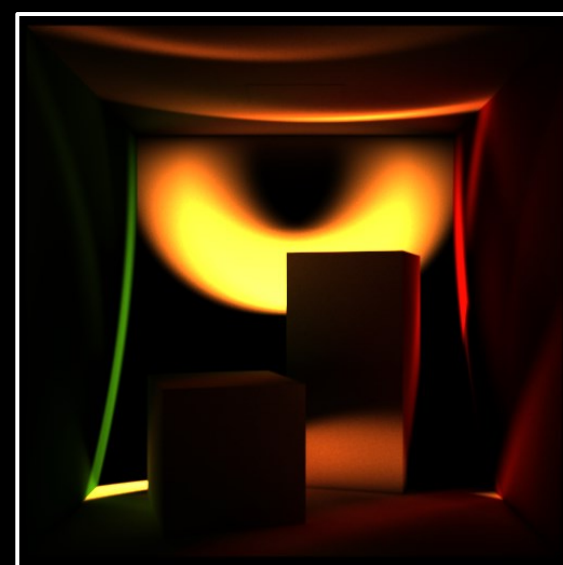
continuous-wave



transient

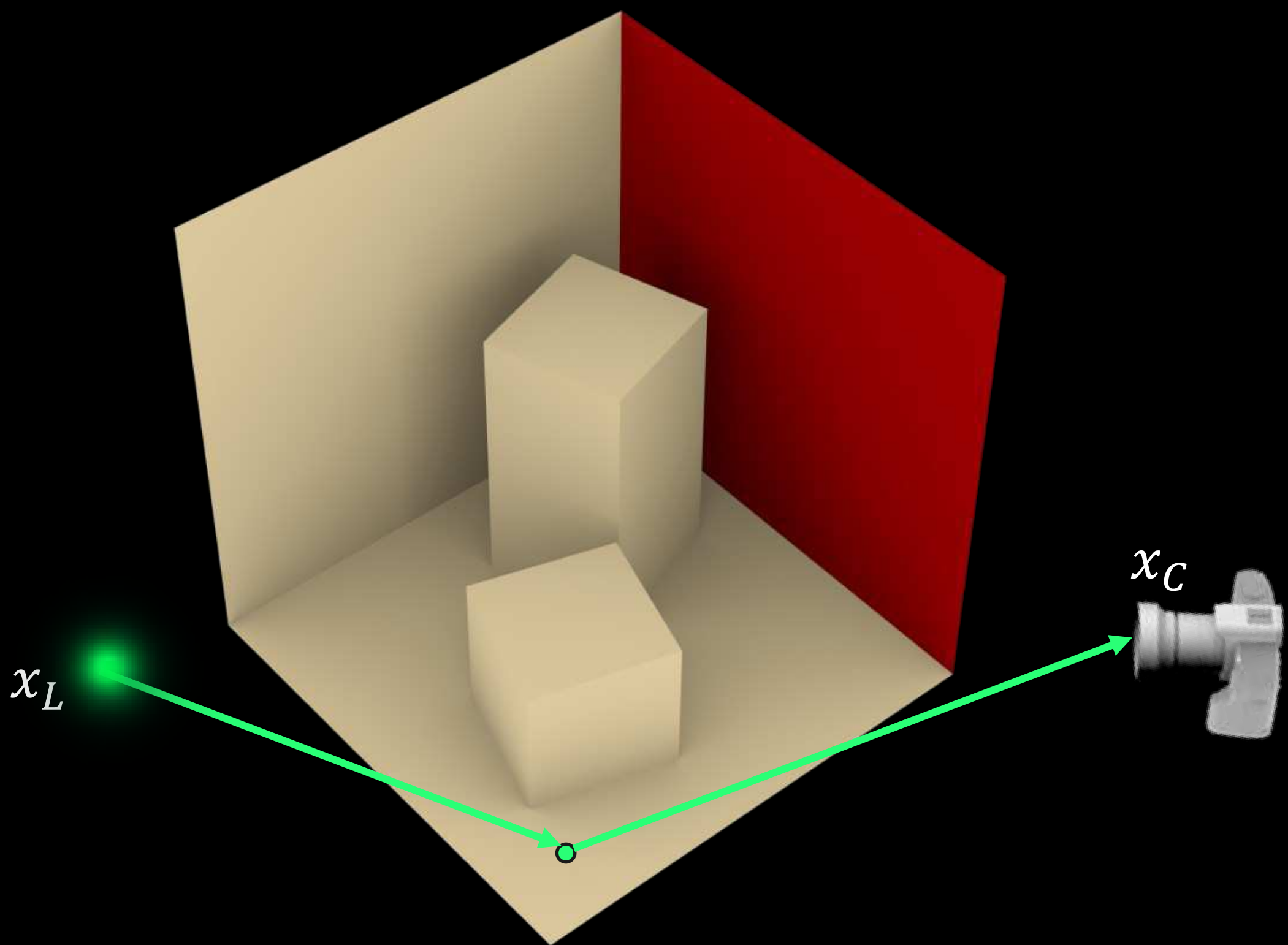


time-gated

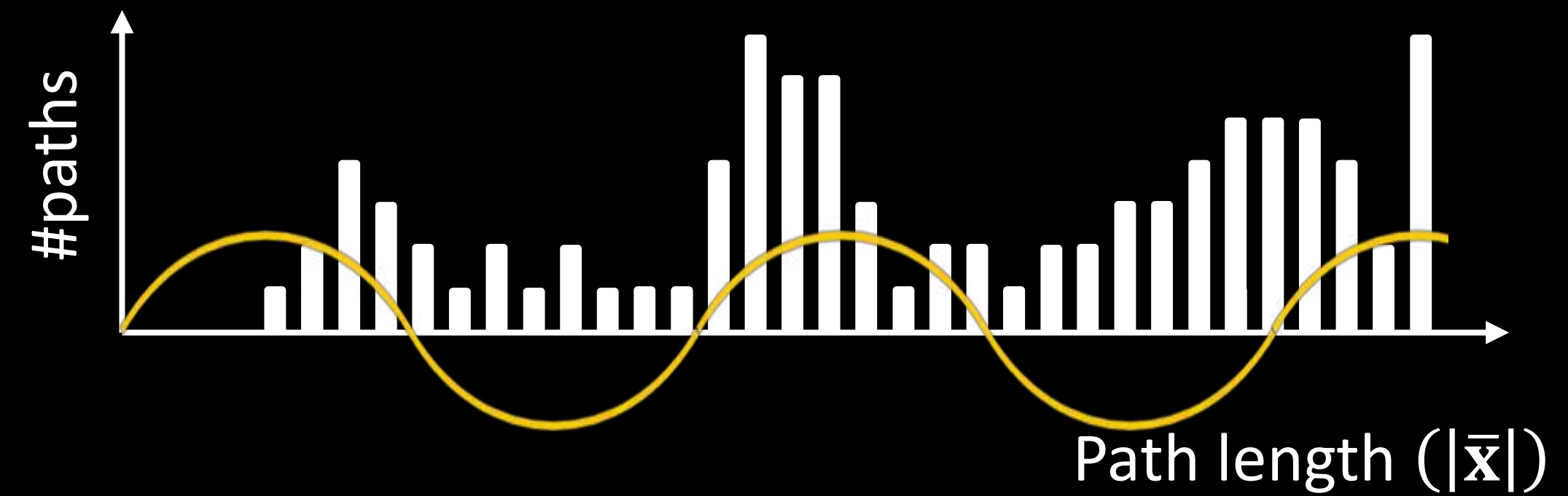
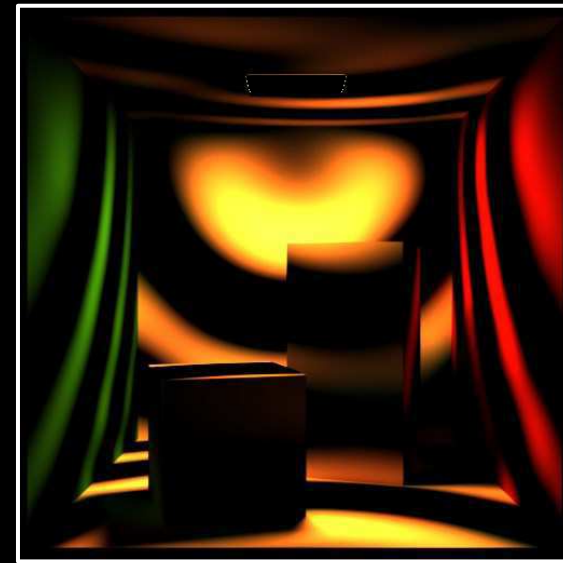


No control over path length

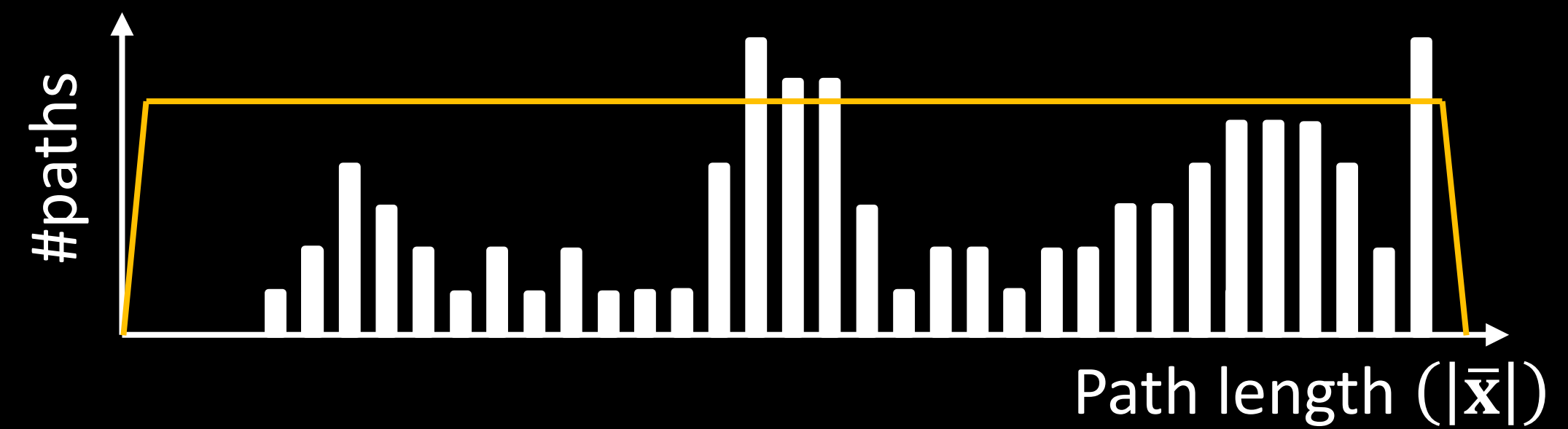
Path sampling for time-gated rendering is challenging



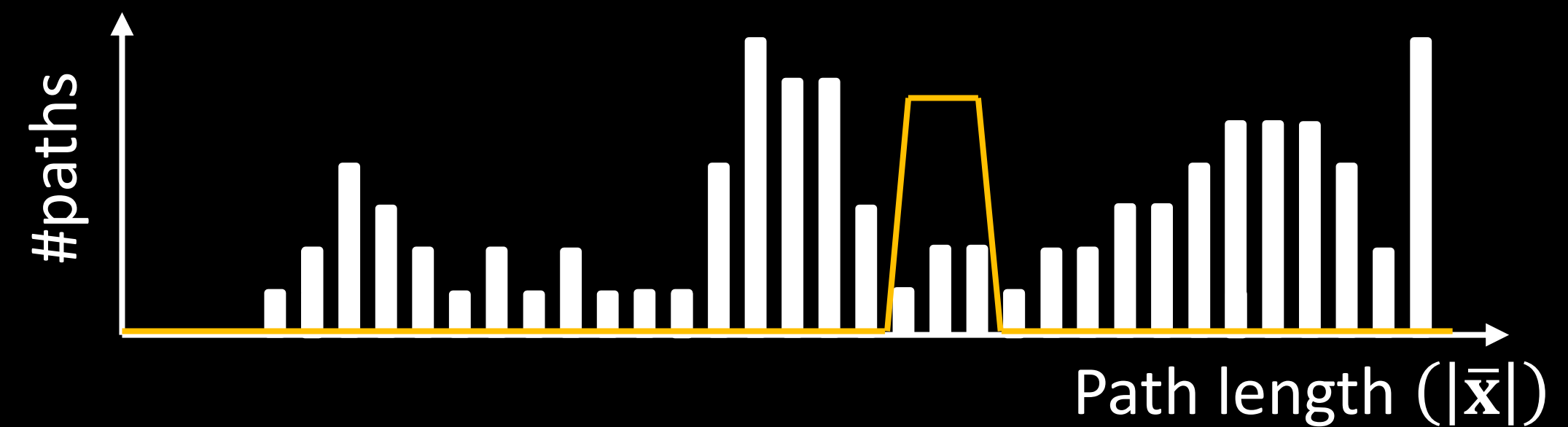
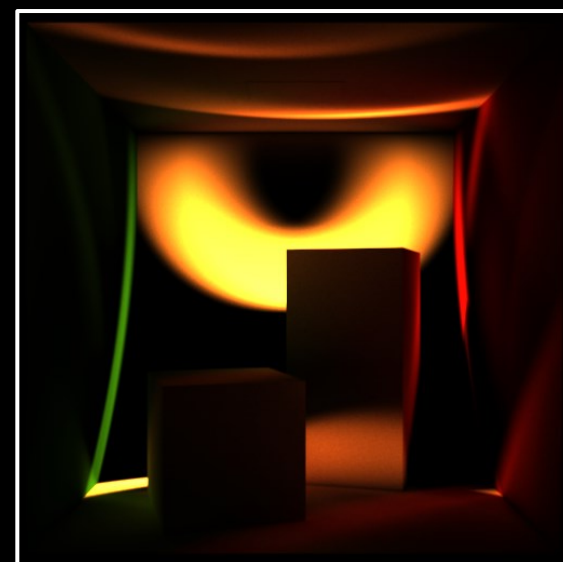
continuous-wave



transient

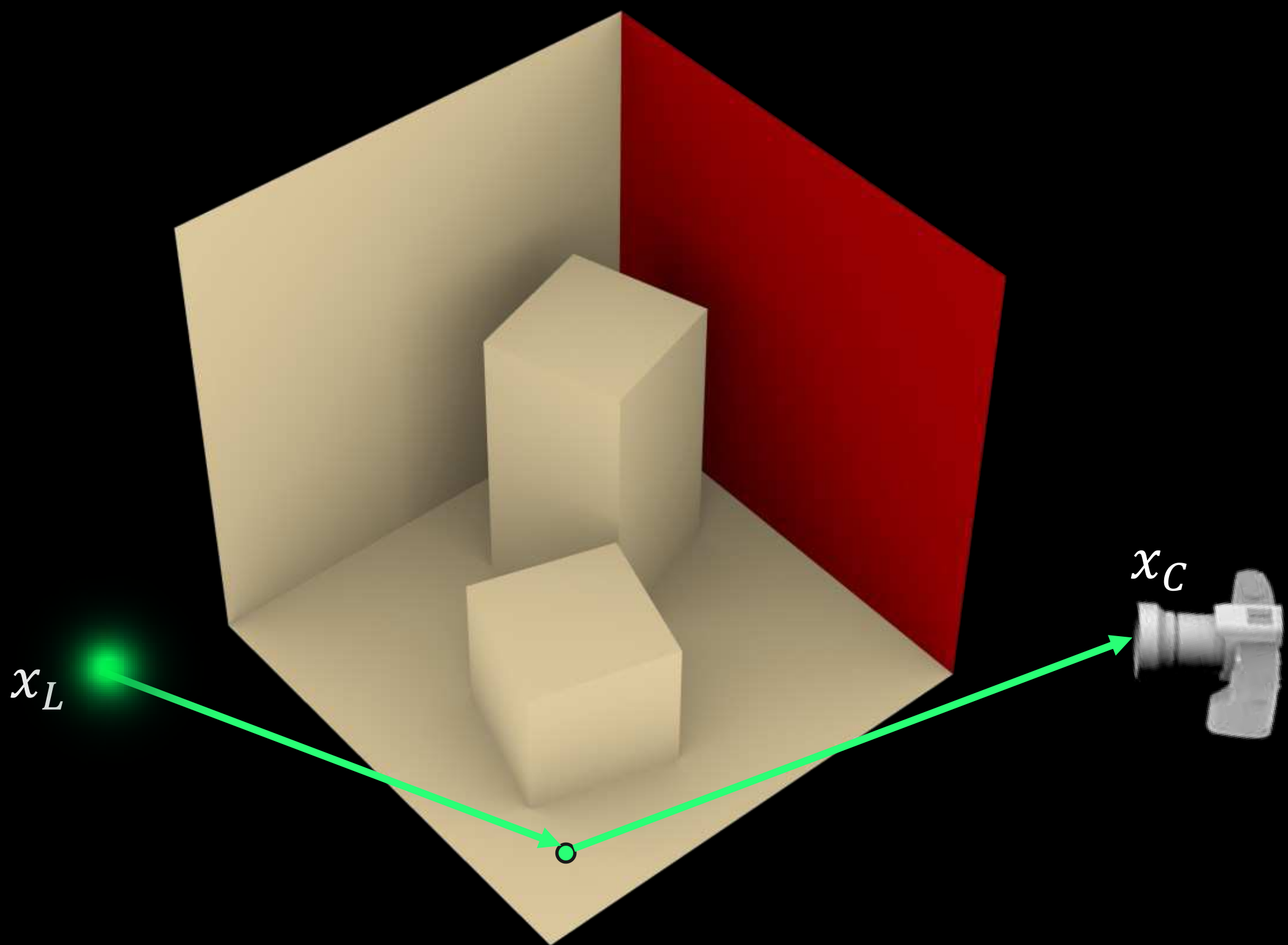


time-gated

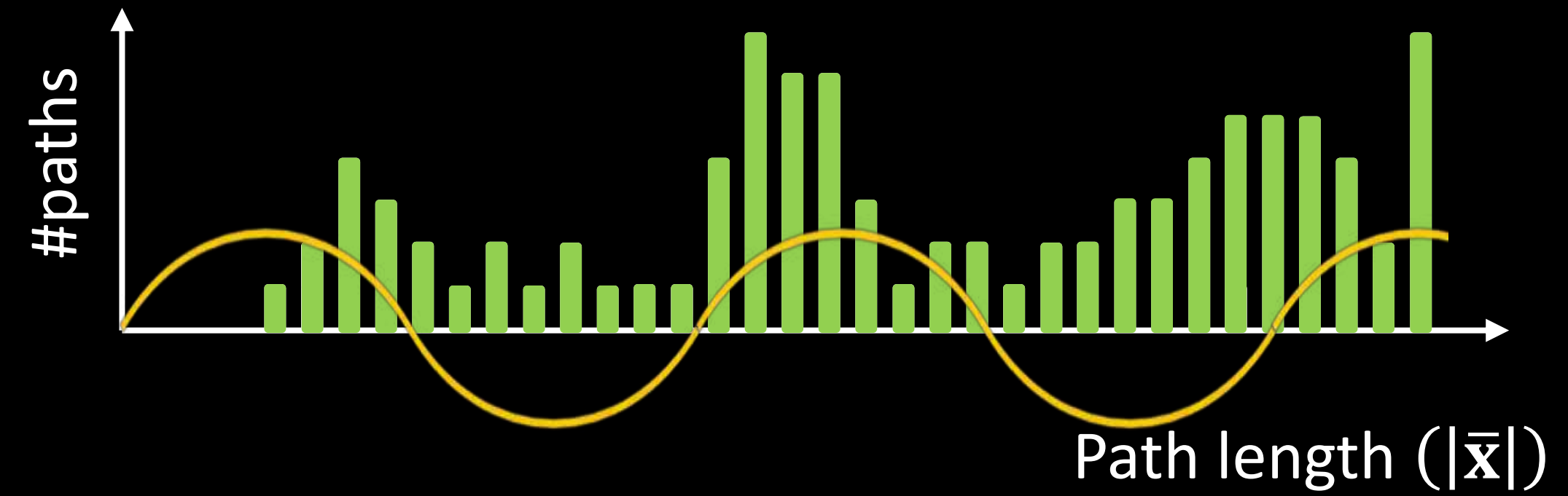
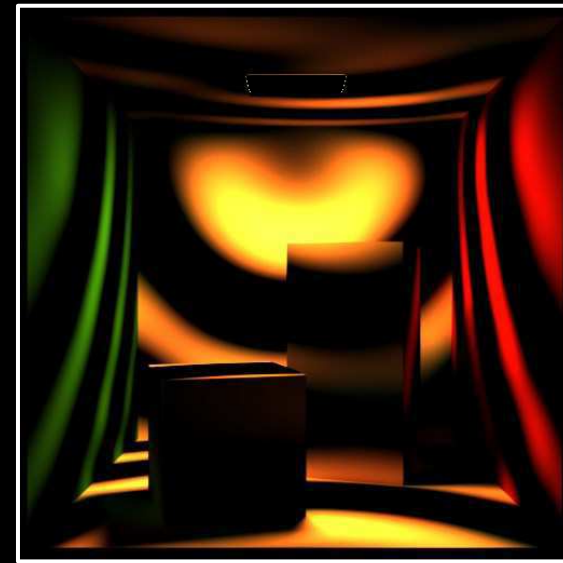


No control over path length

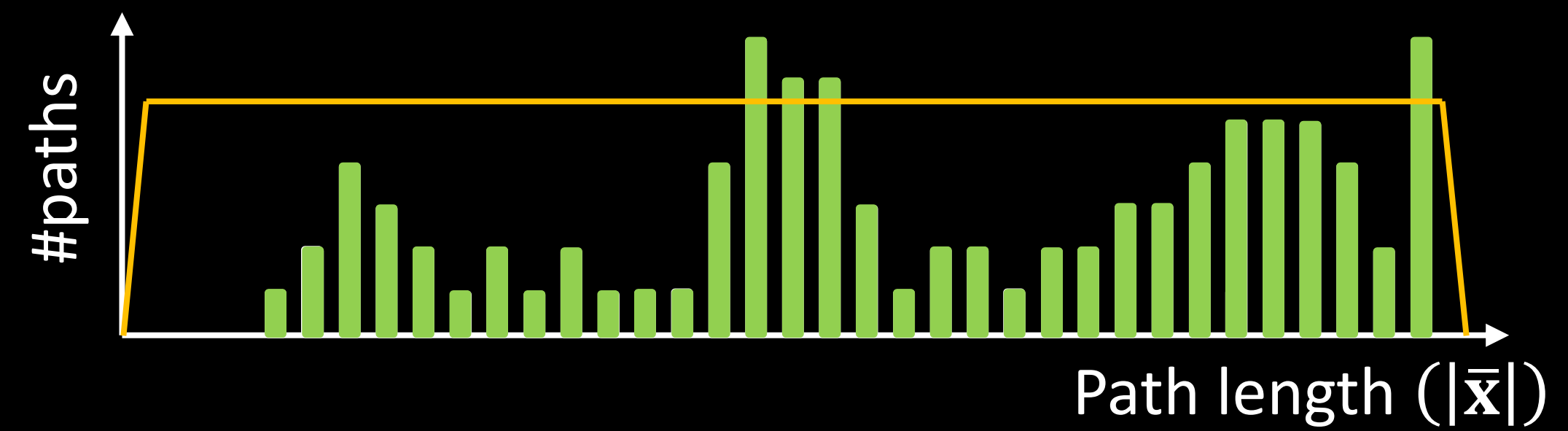
Path sampling for time-gated rendering is challenging



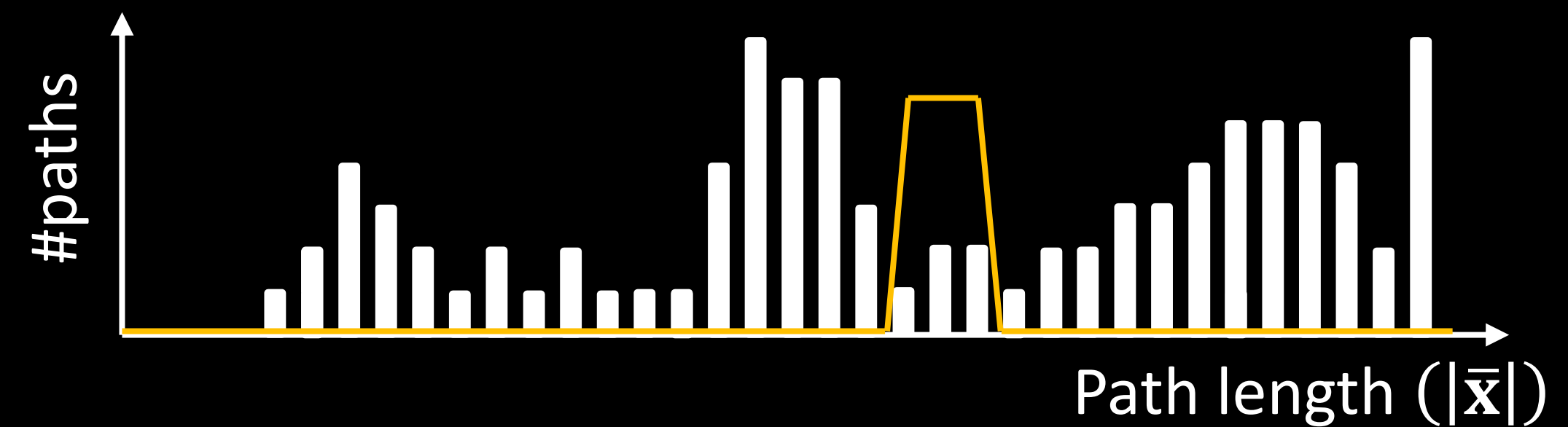
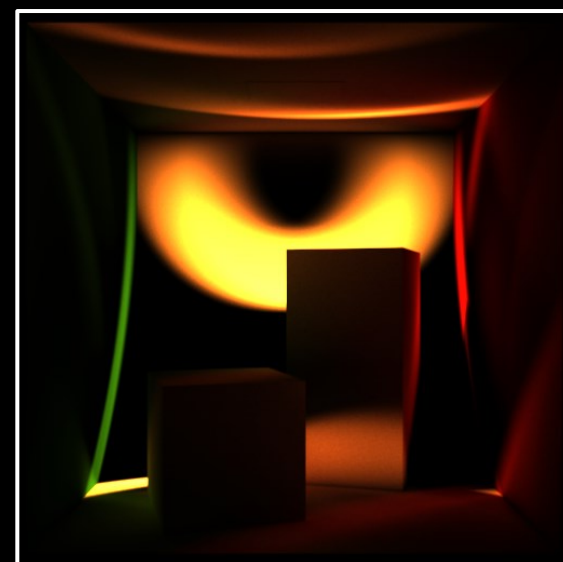
continuous-wave



transient

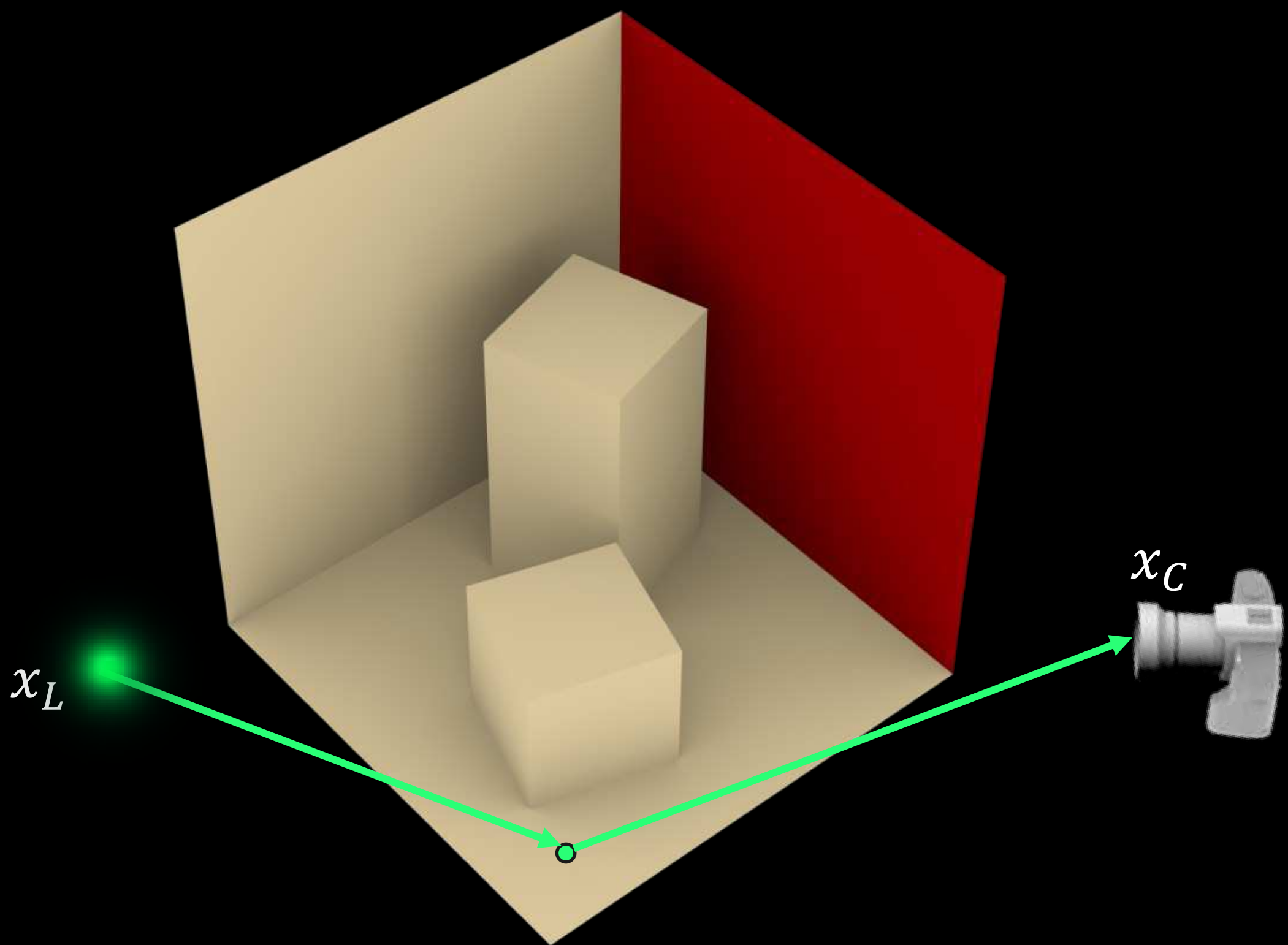


time-gated

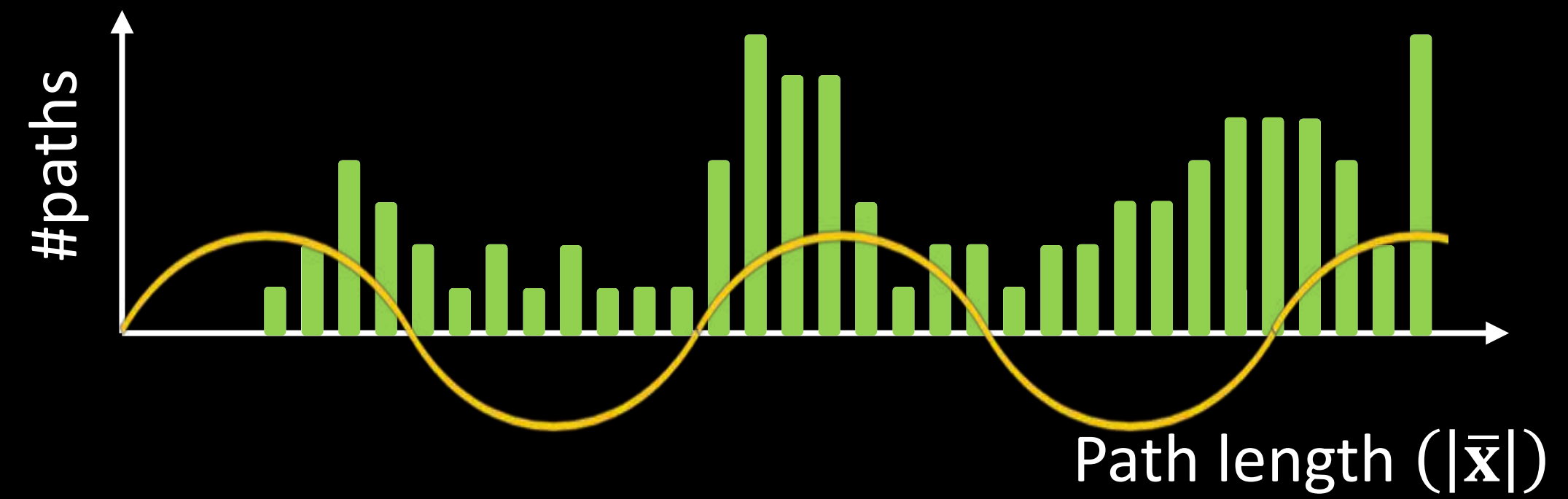
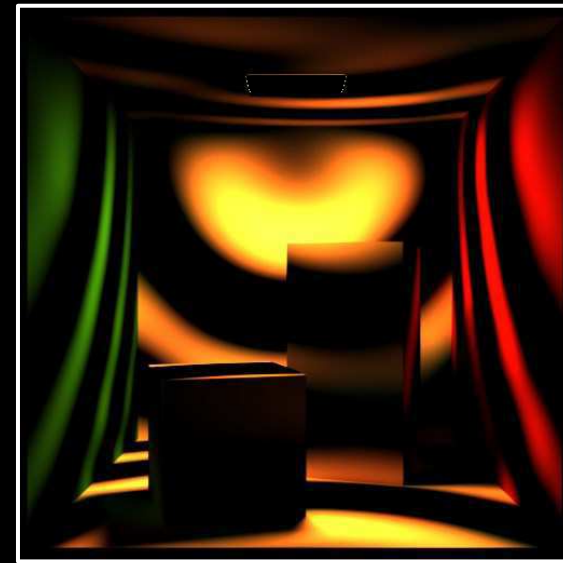


No control over path length

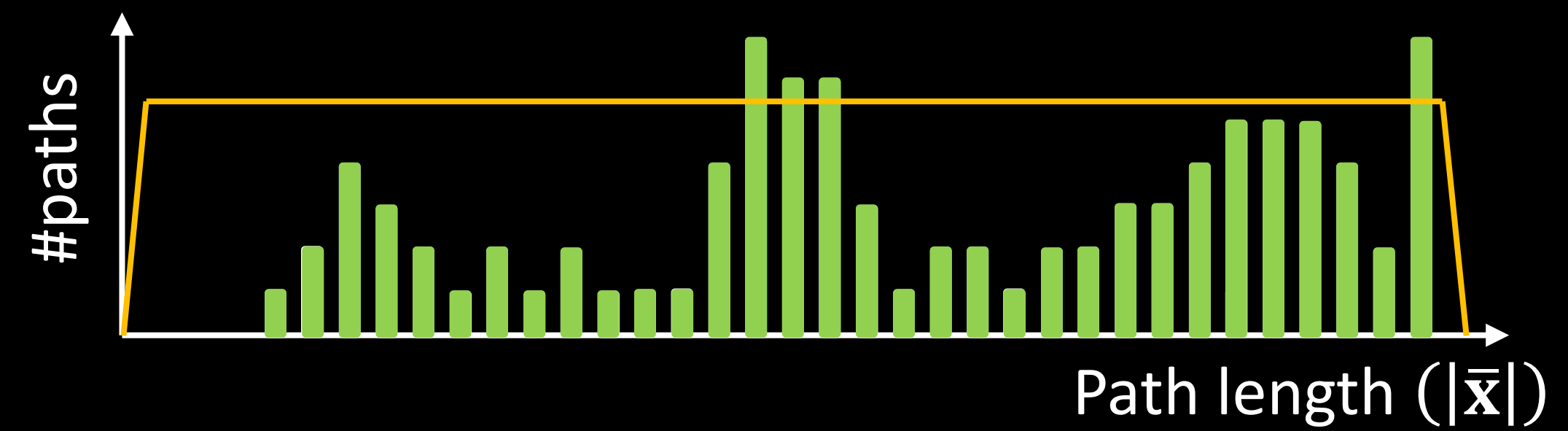
Path sampling for time-gated rendering is challenging



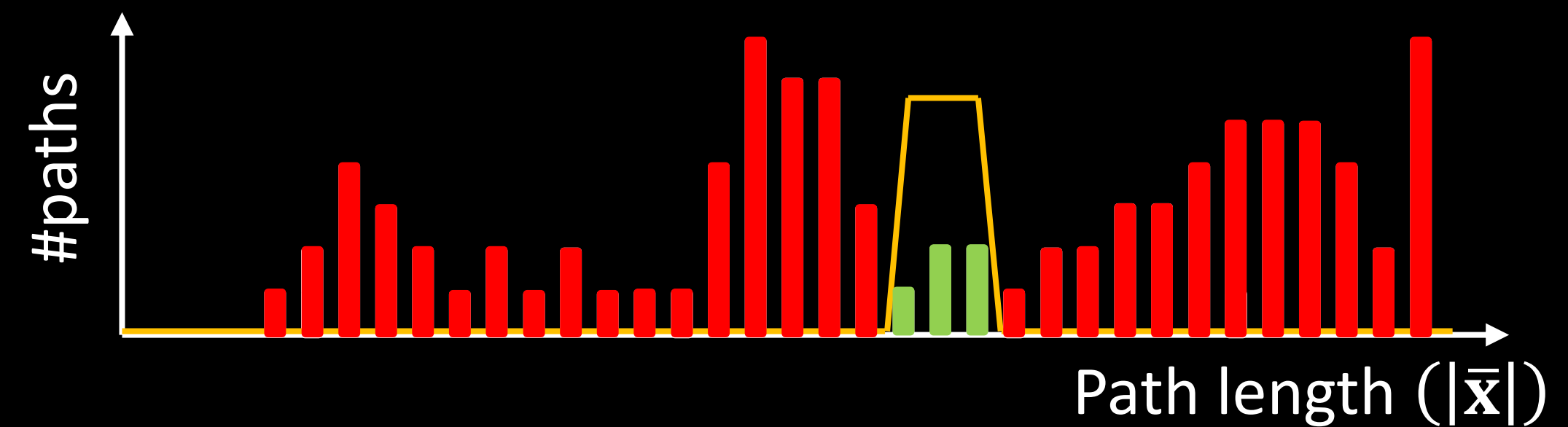
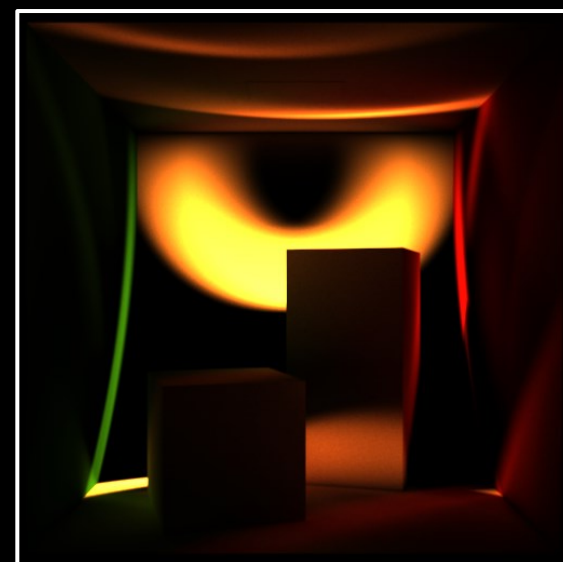
continuous-wave



transient

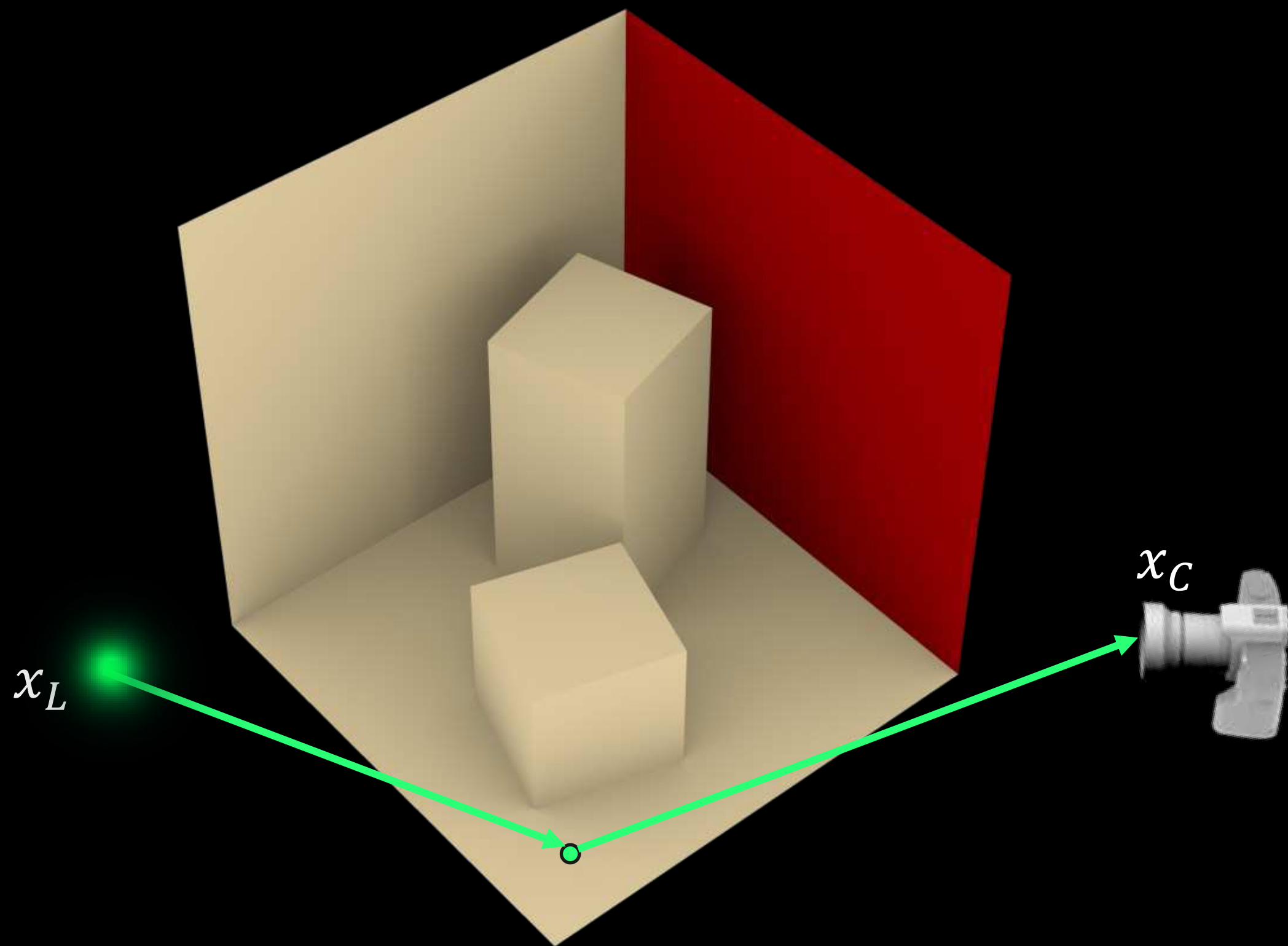


time-gated

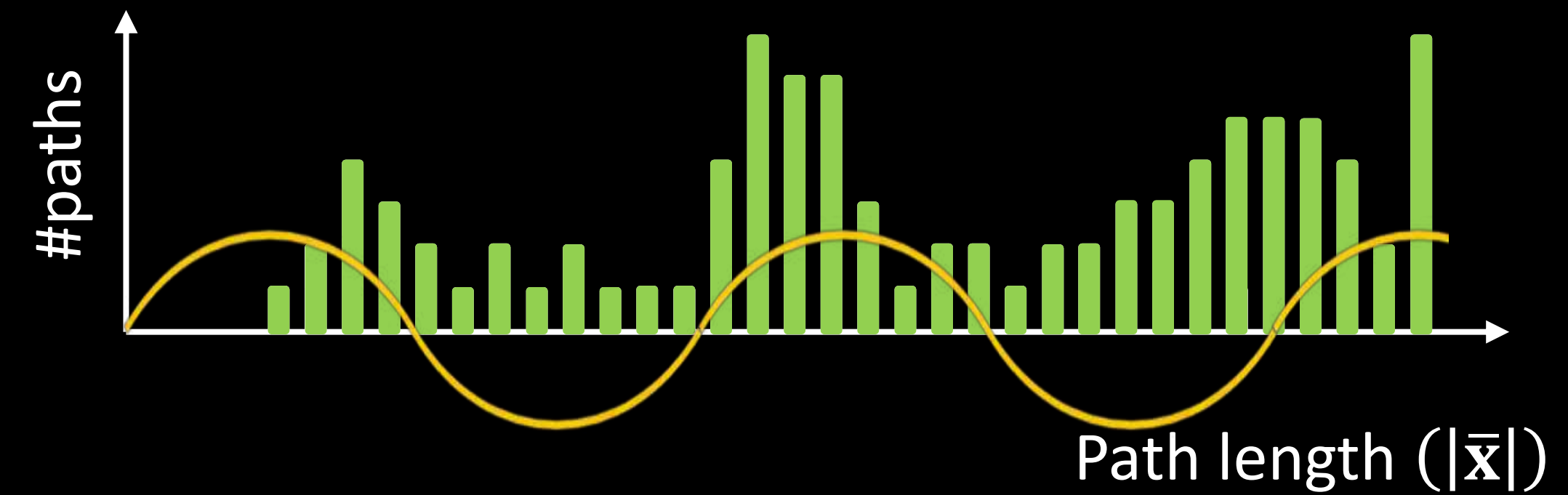


No control over path length

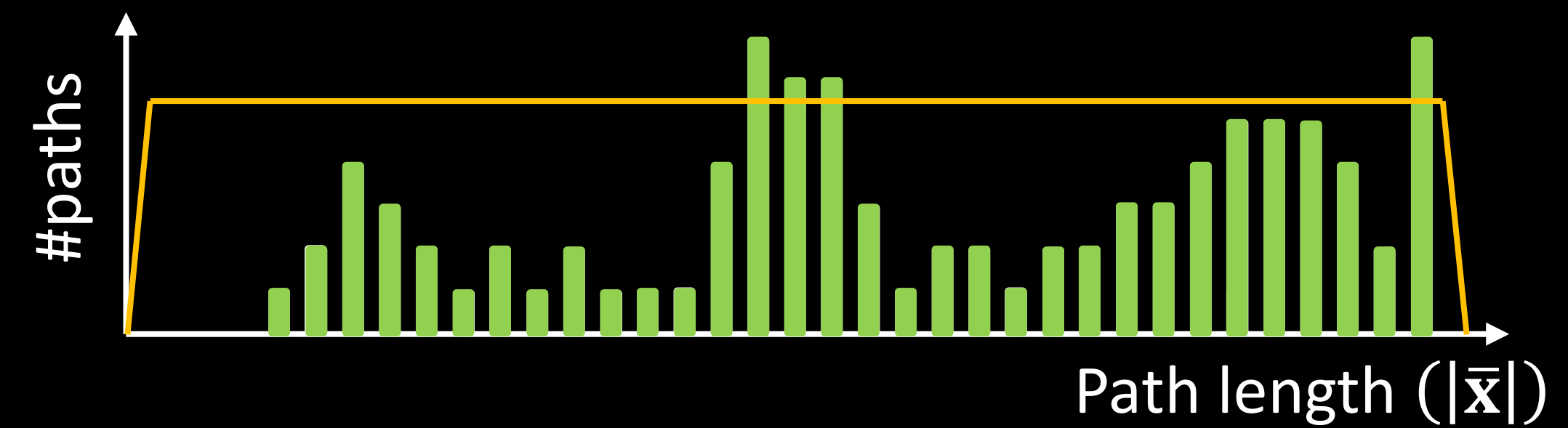
Path sampling for time-gated rendering is challenging



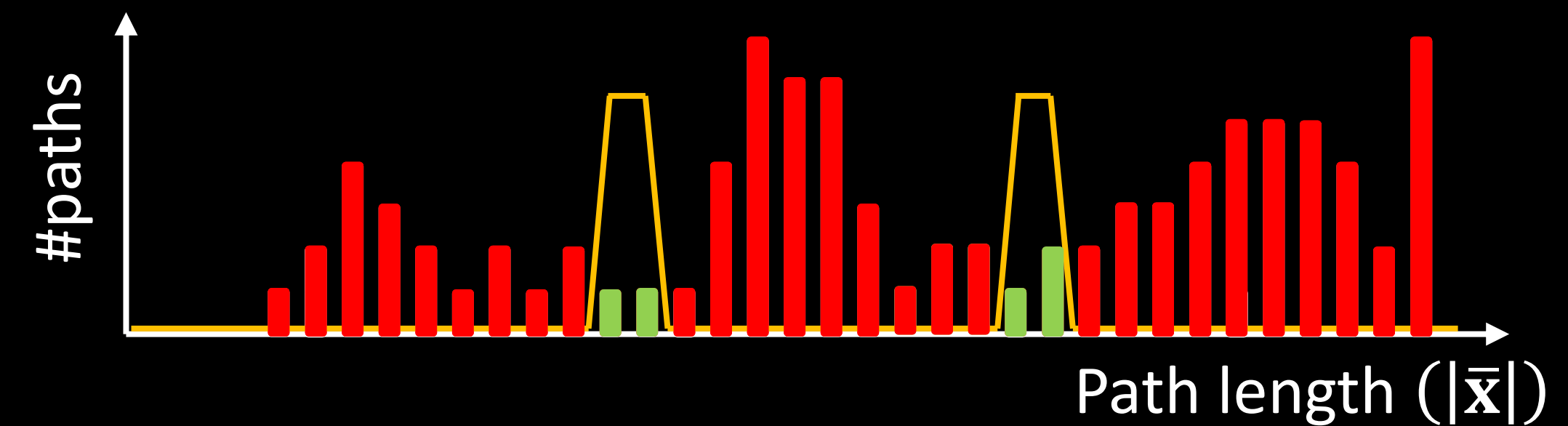
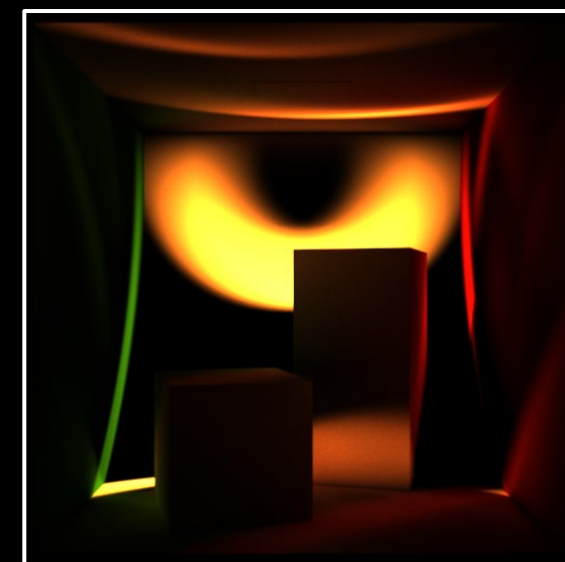
continuous-wave



transient

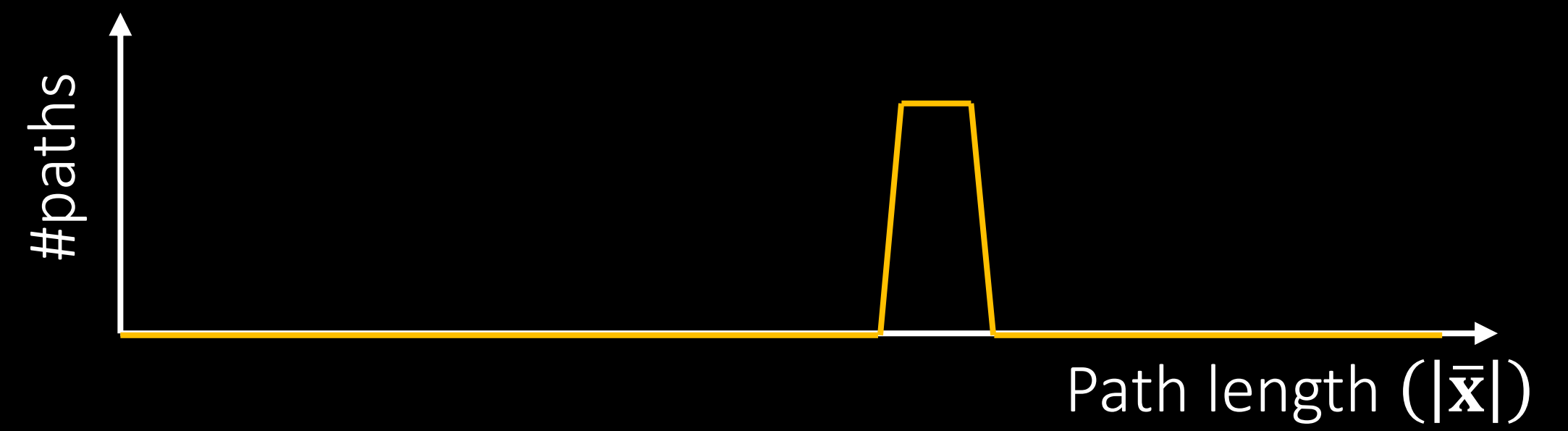
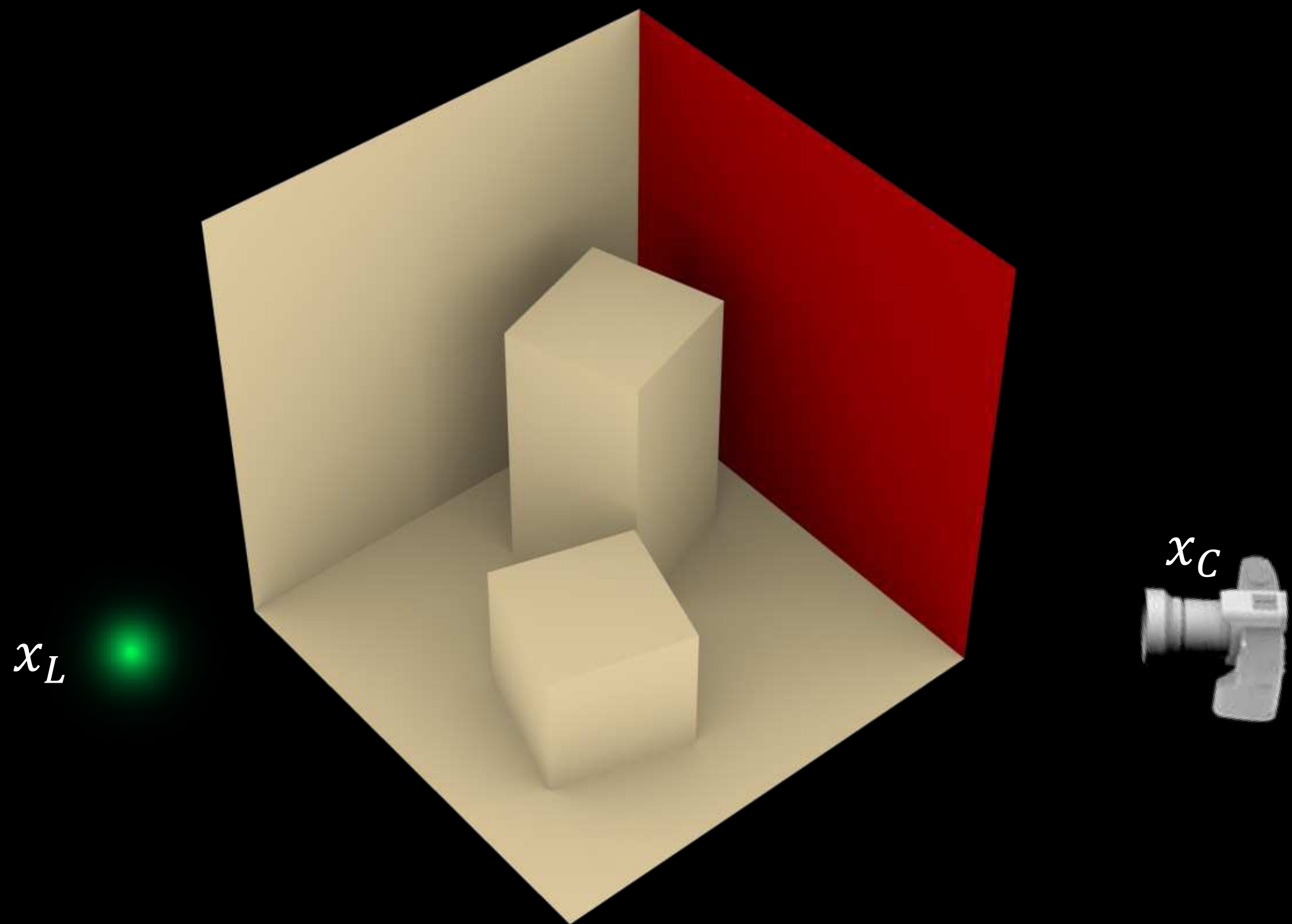


time-gated

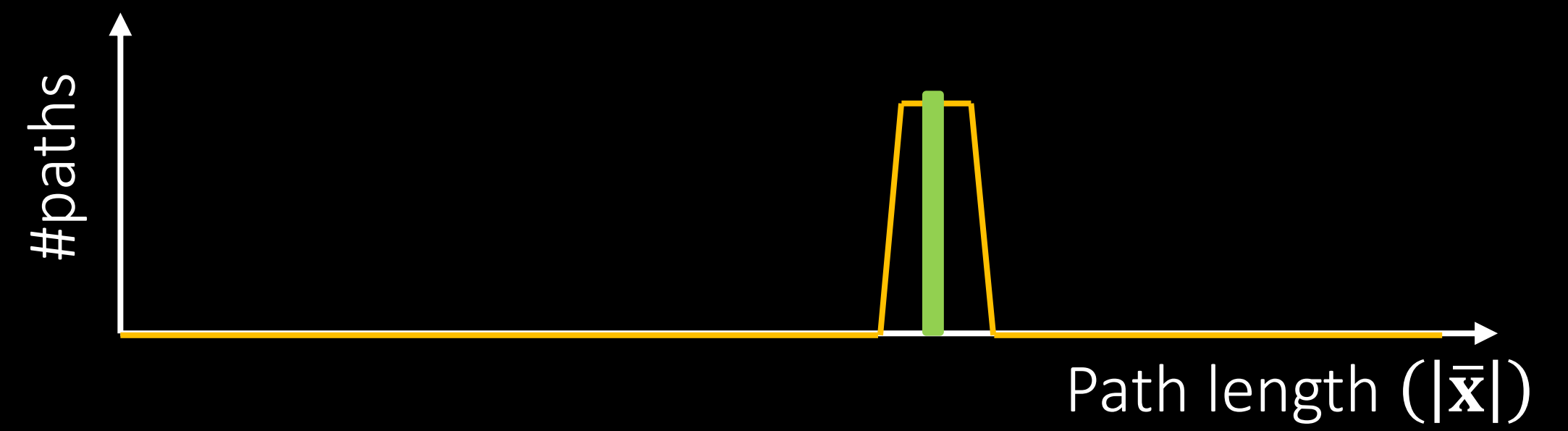
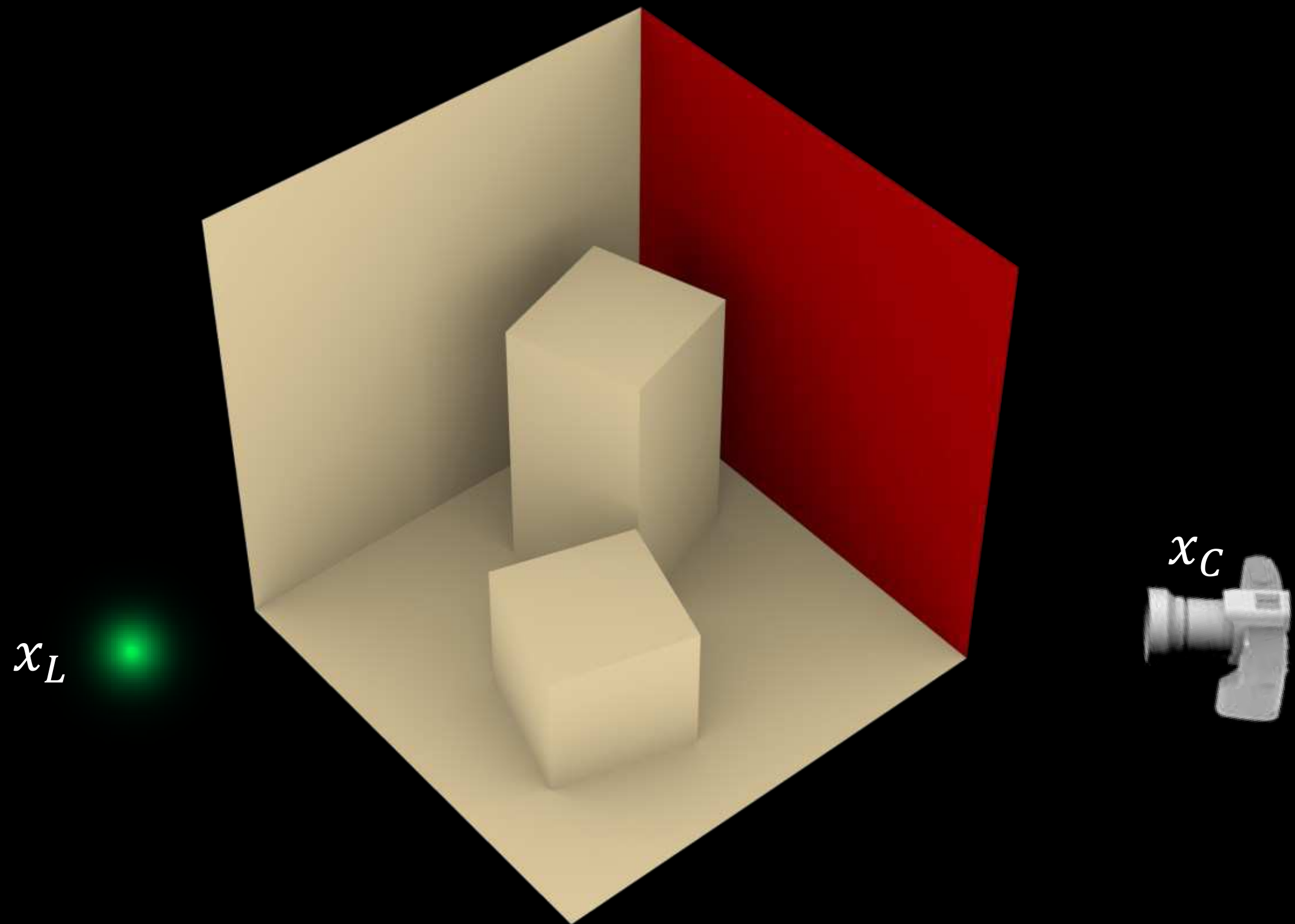


No control over path length

Path sampling for time-gated case

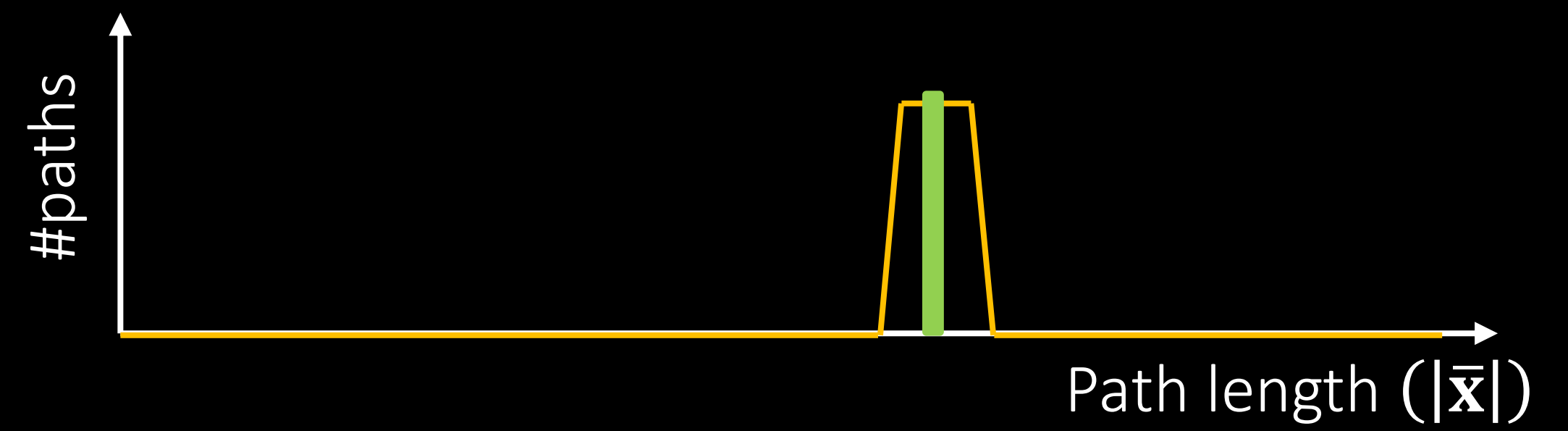
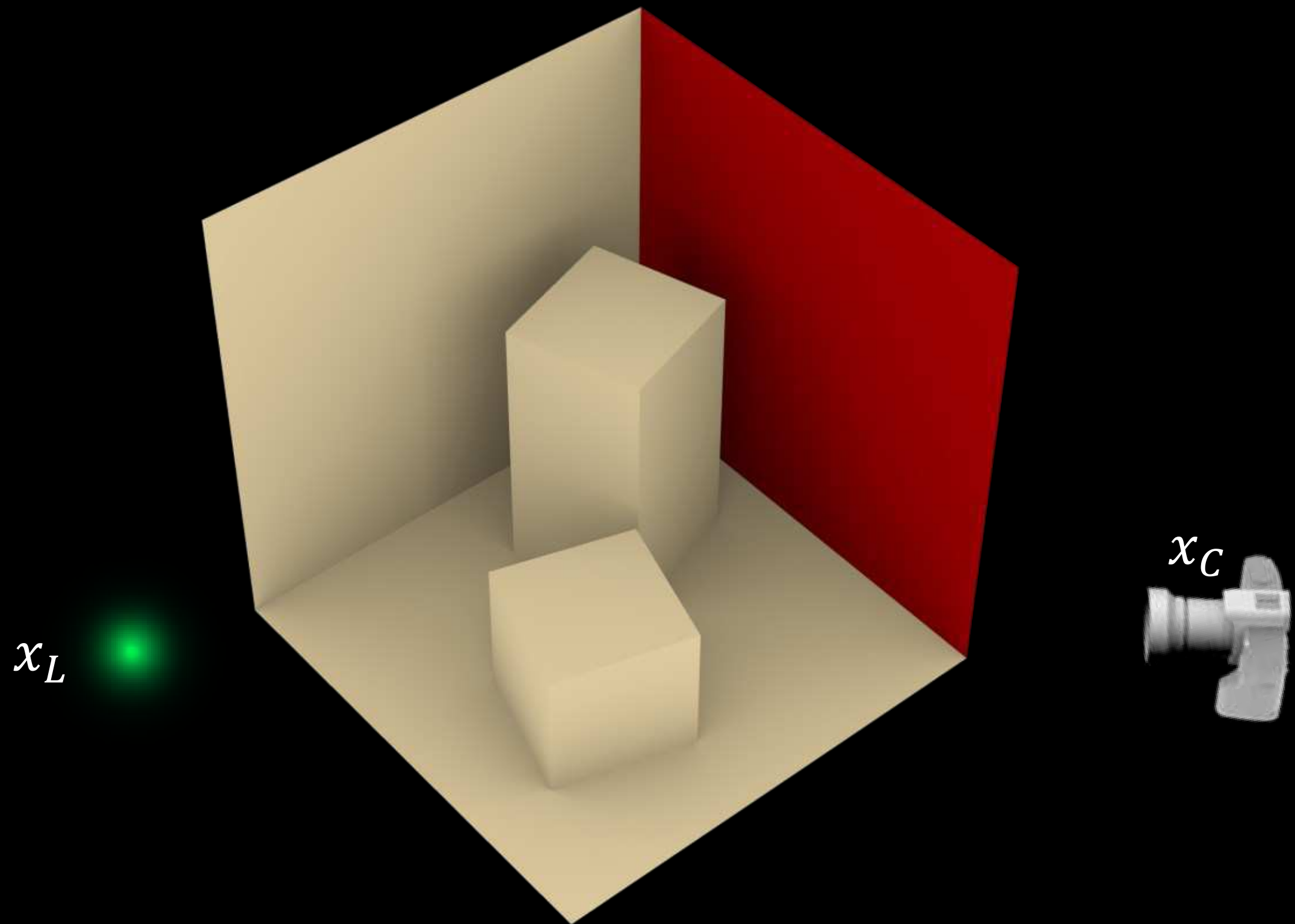


Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

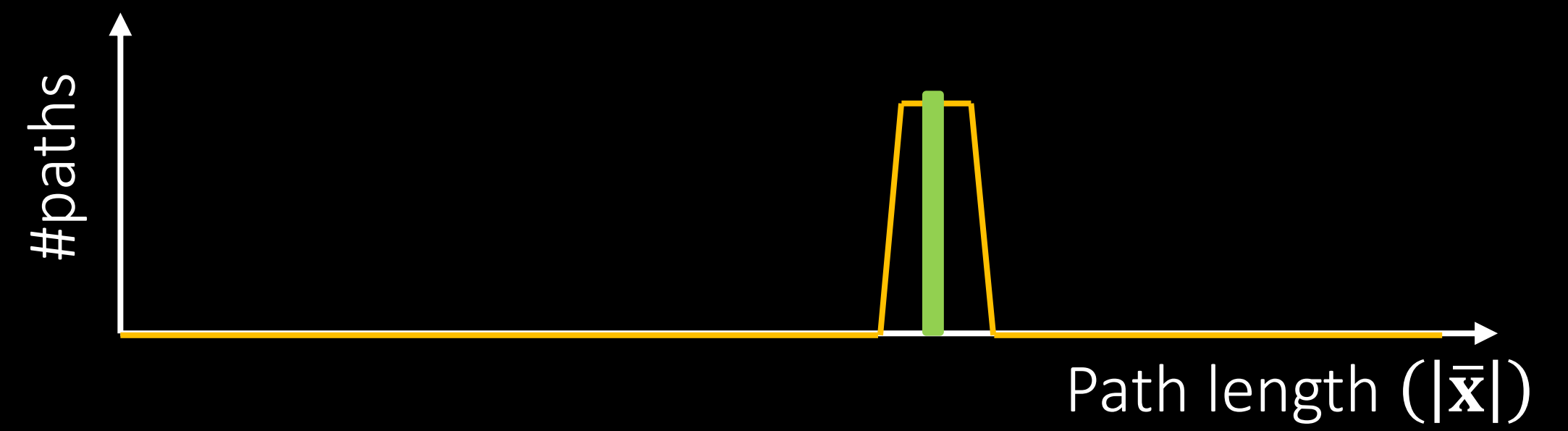
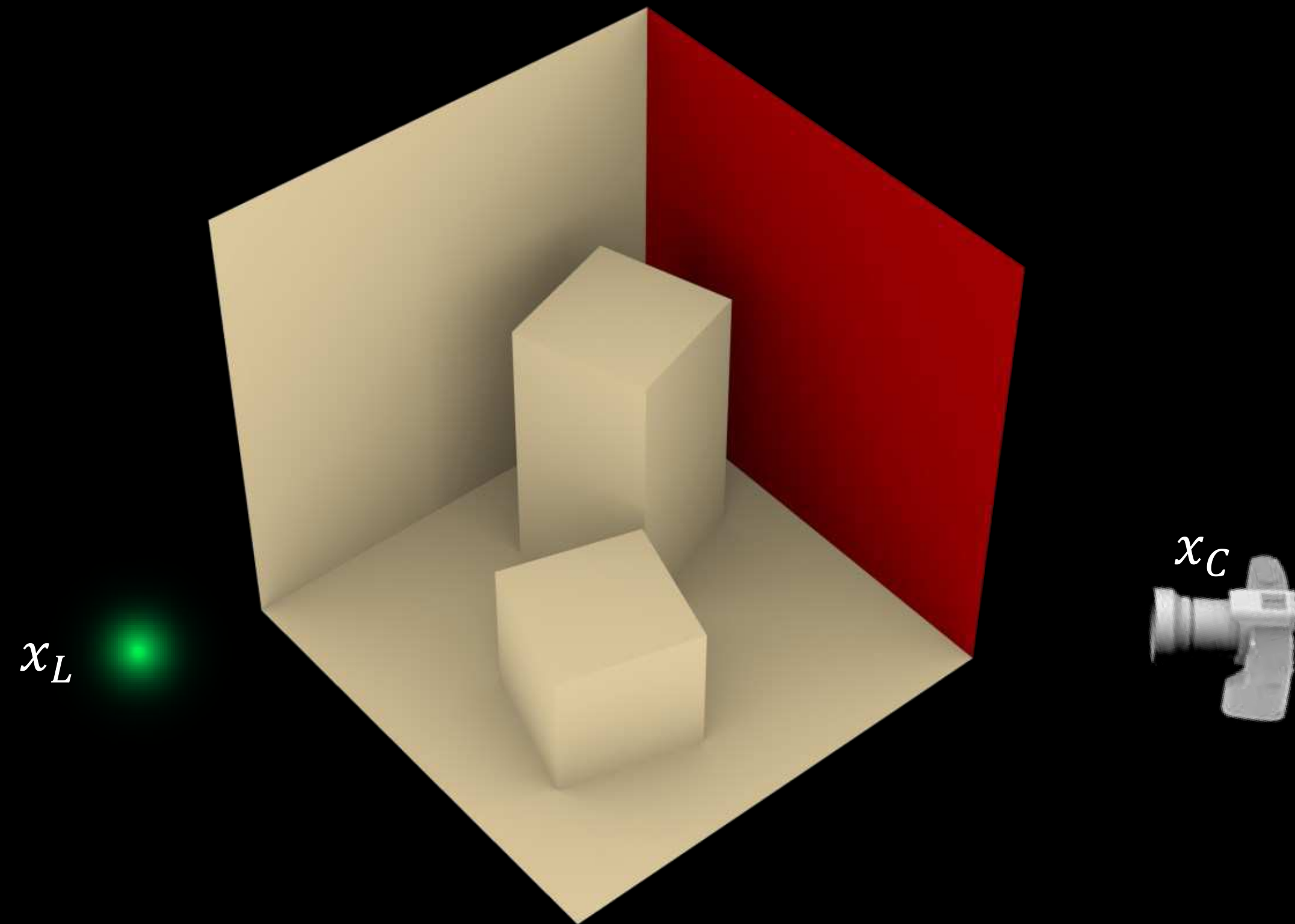
Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

Path sampling for time-gated case

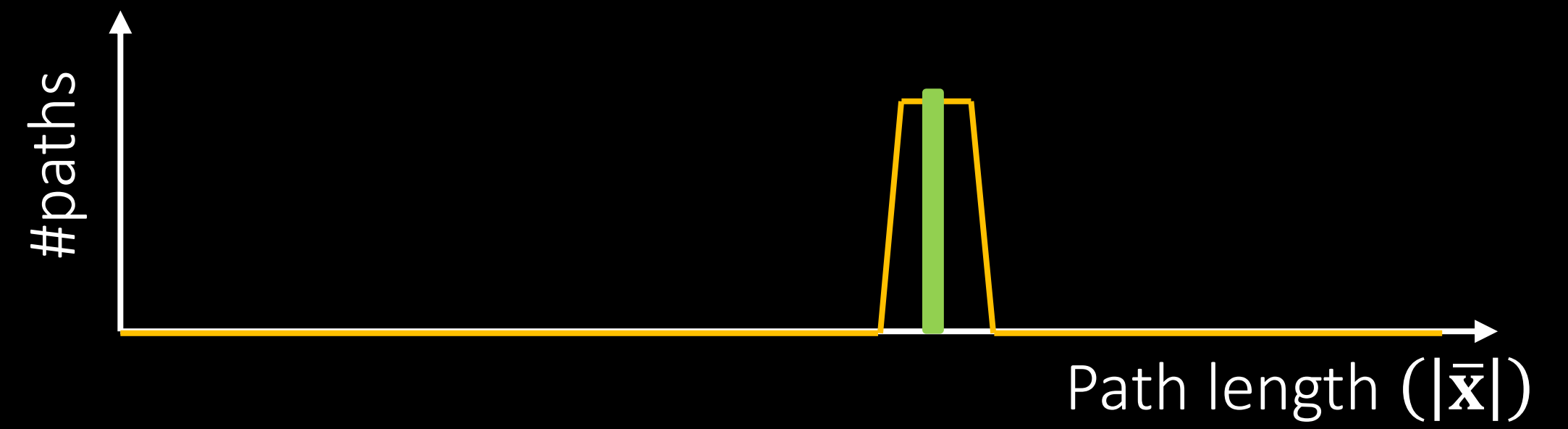
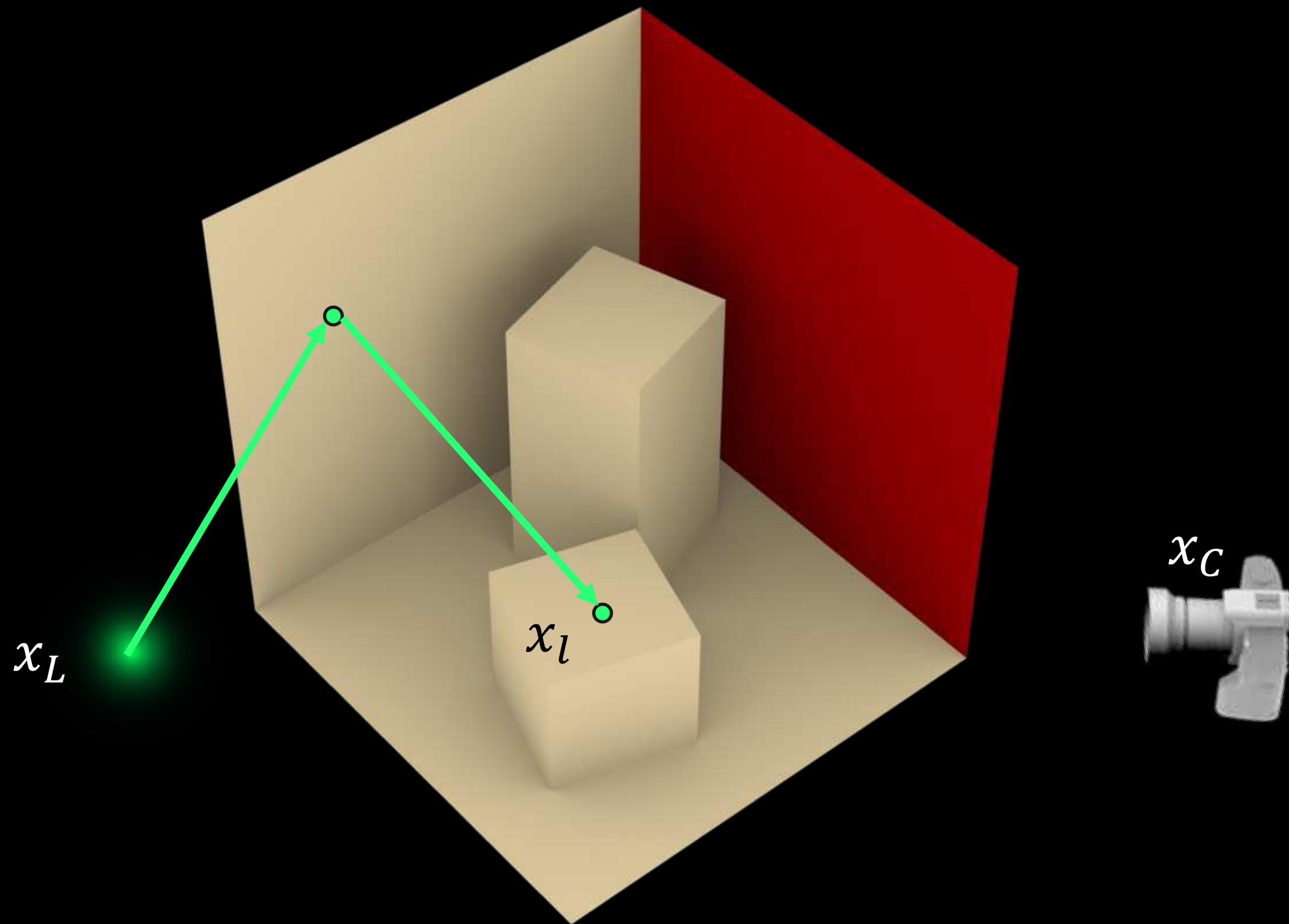


step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

Path sampling for time-gated case



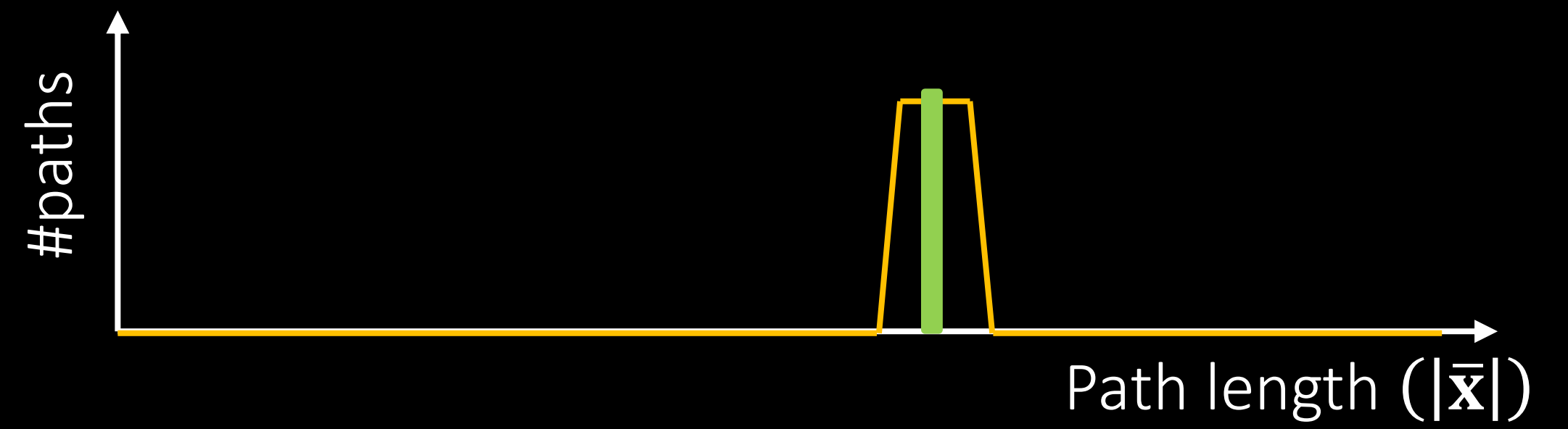
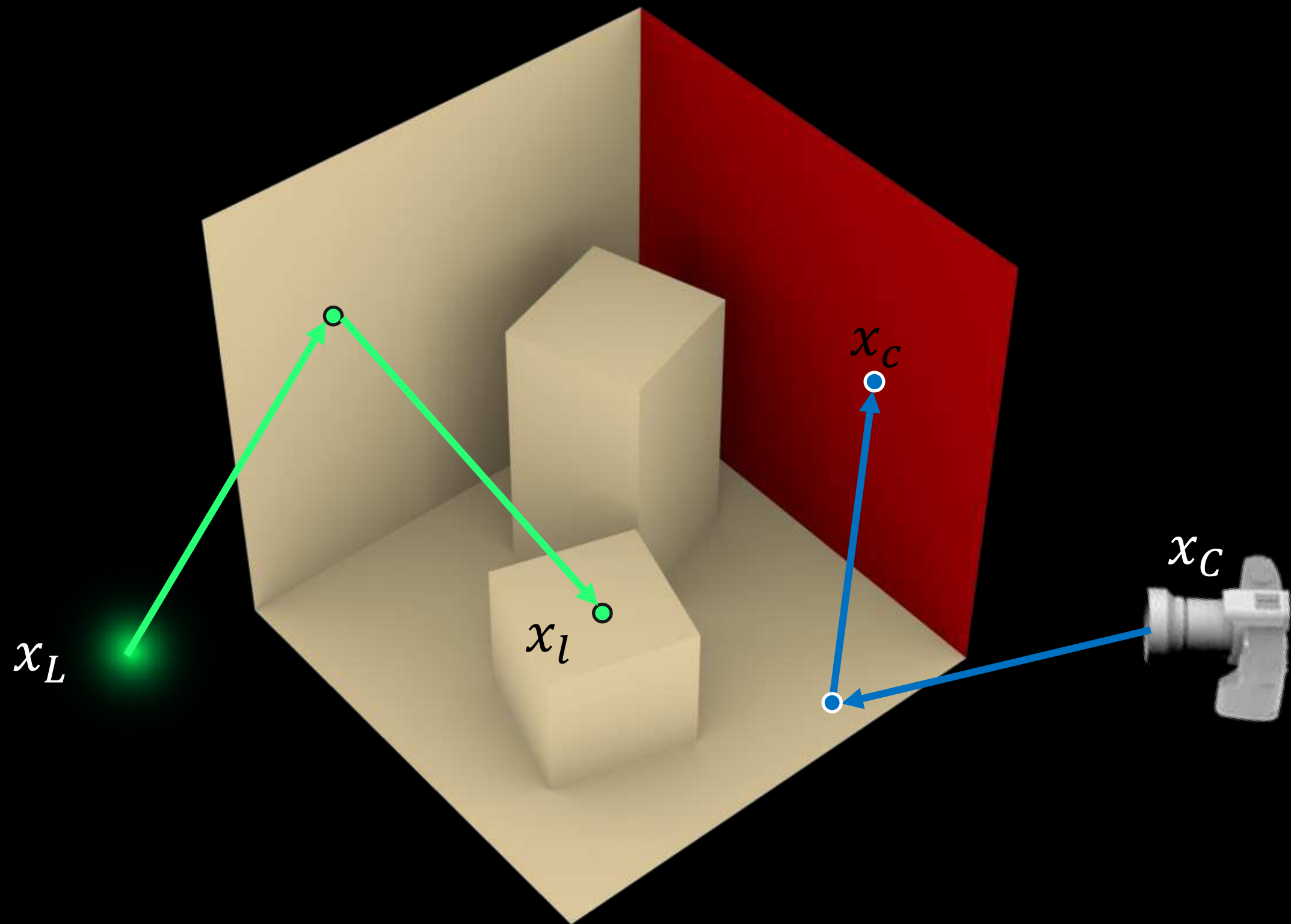
step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

Path sampling for time-gated case



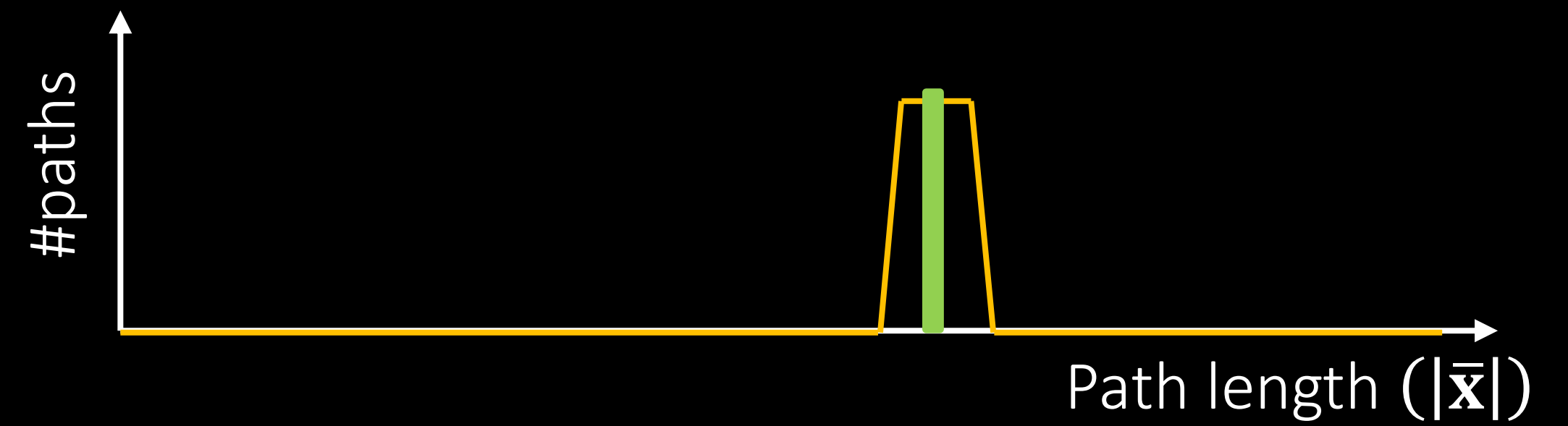
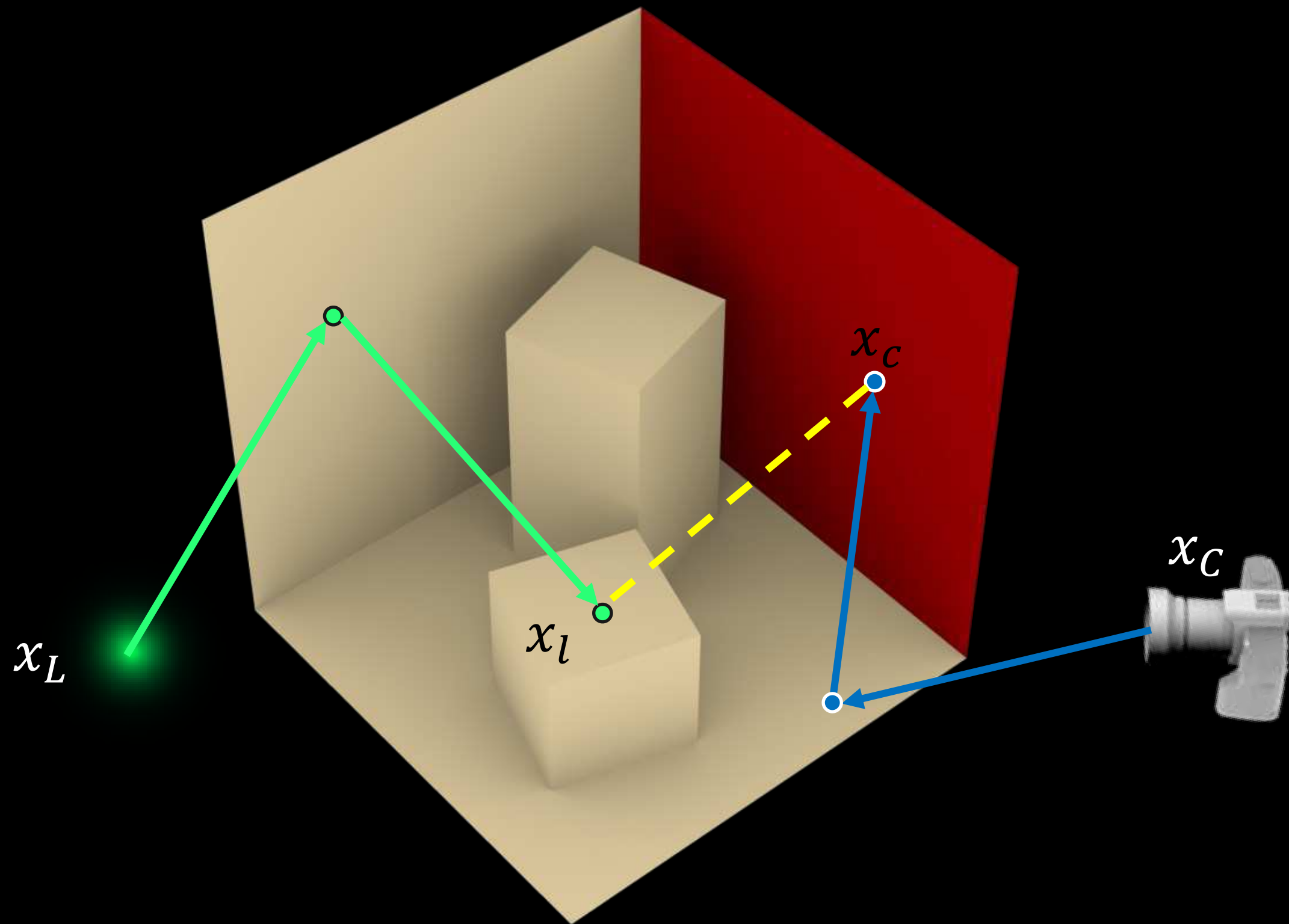
step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

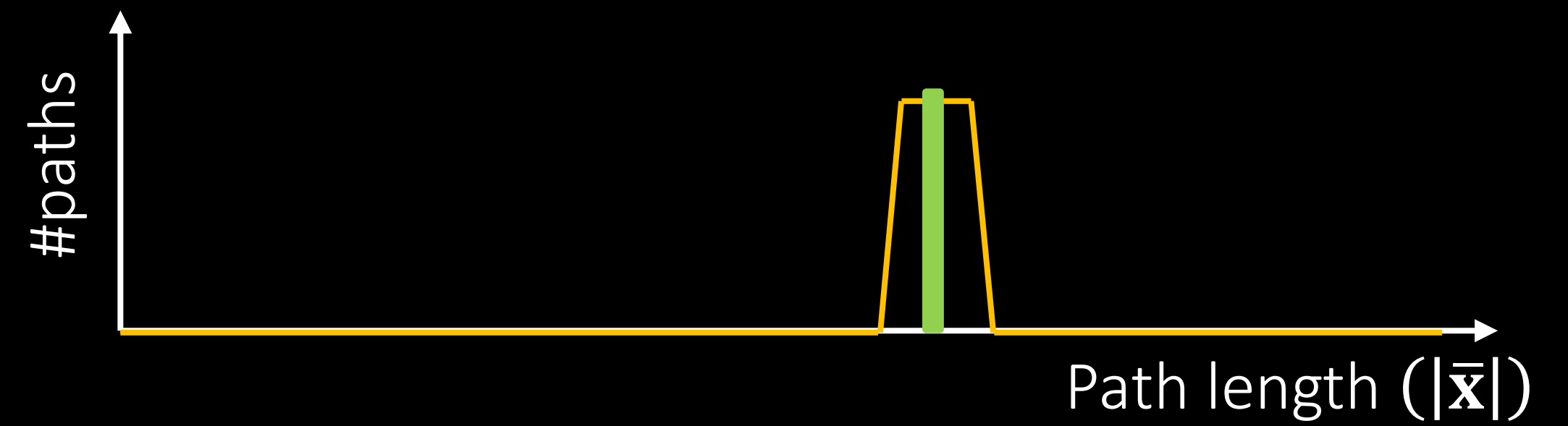
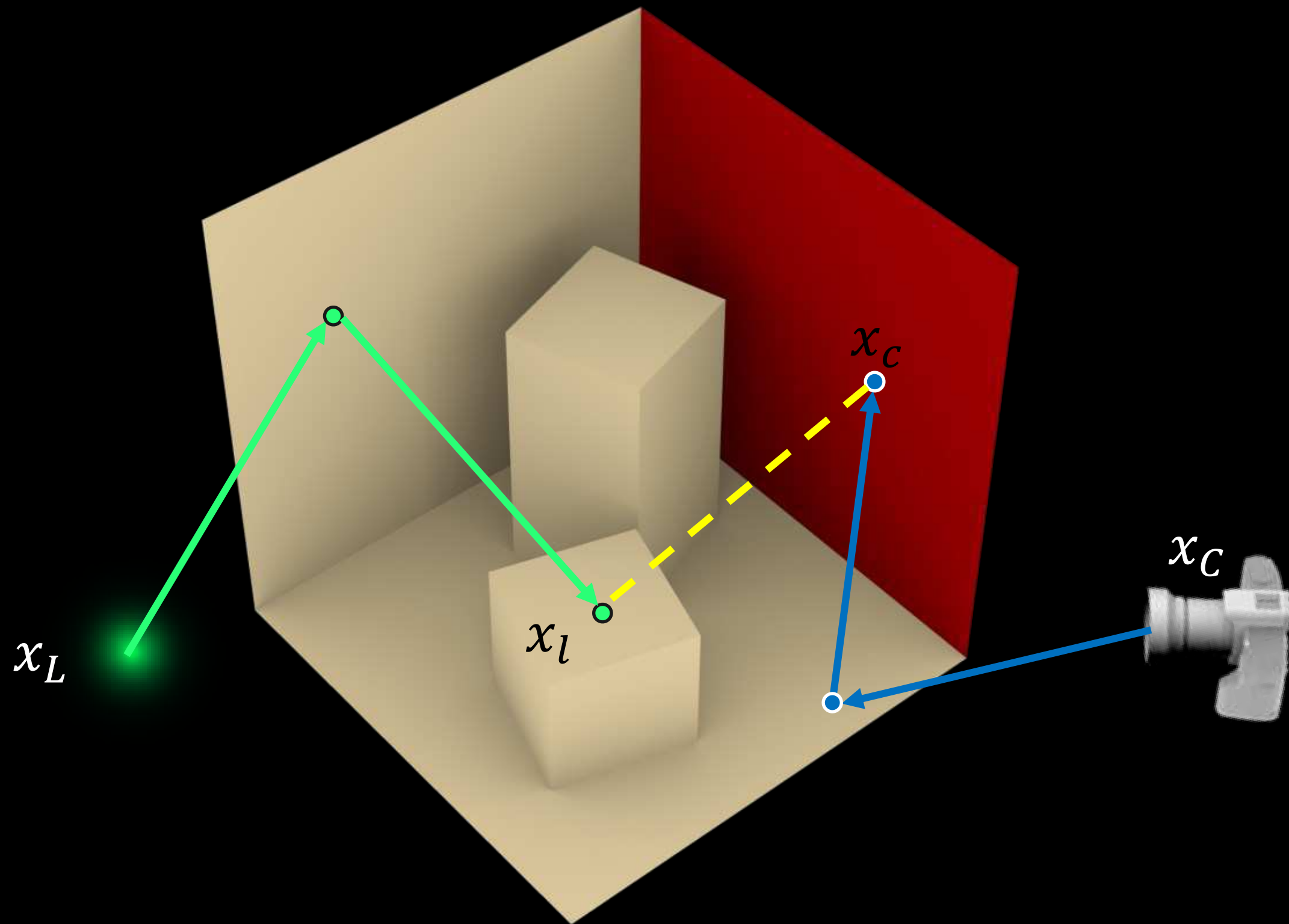
step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join source sub-path end x_s and camera sub-path end x_c

Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

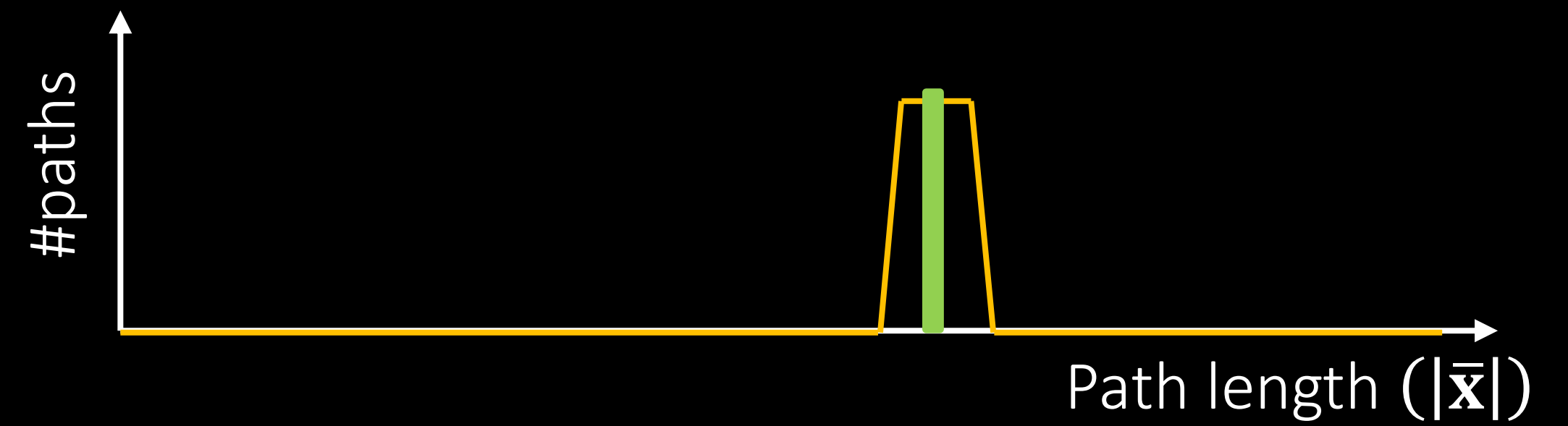
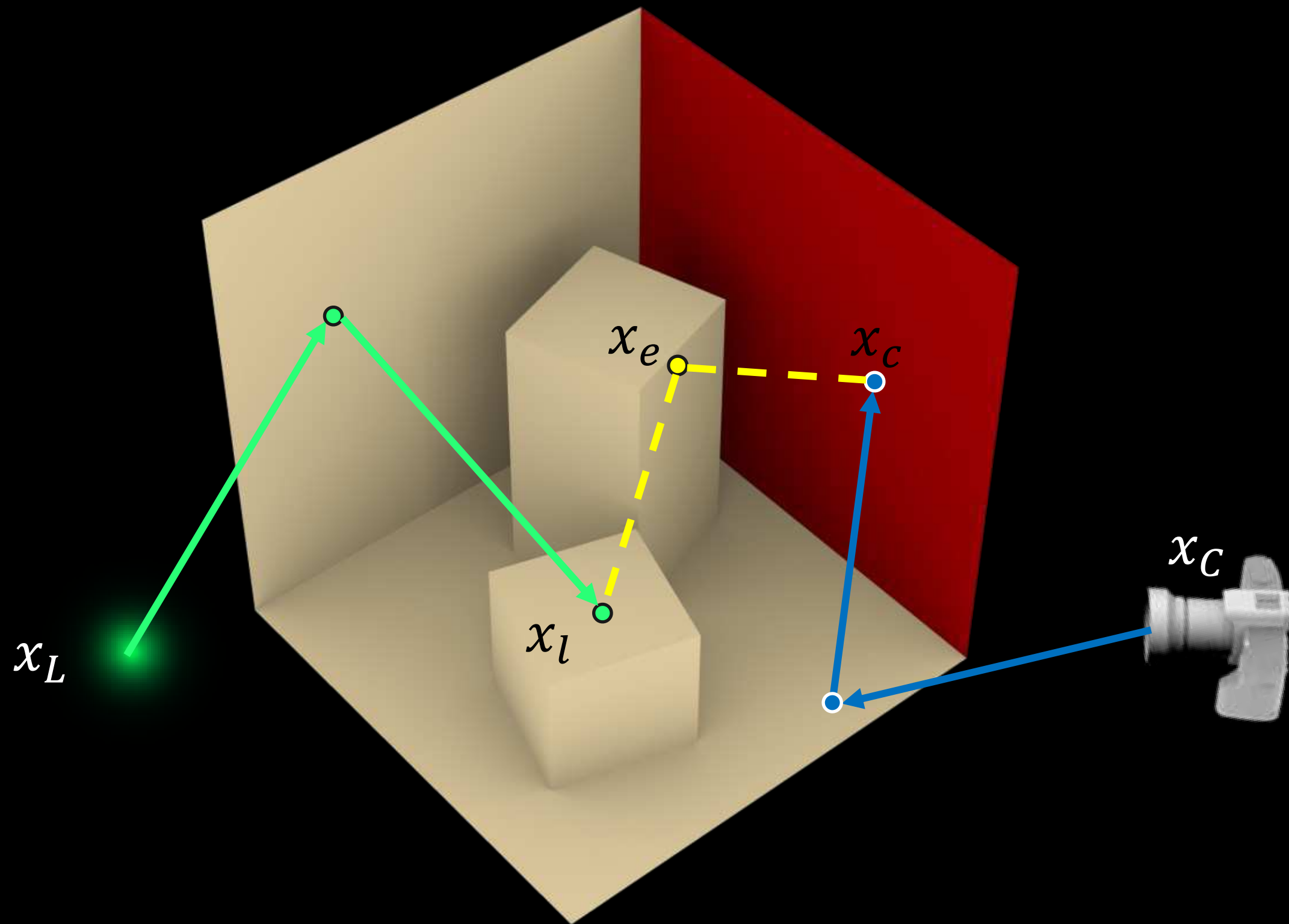
step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

~~join source sub path end x_s and camera sub path end x_c~~

Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

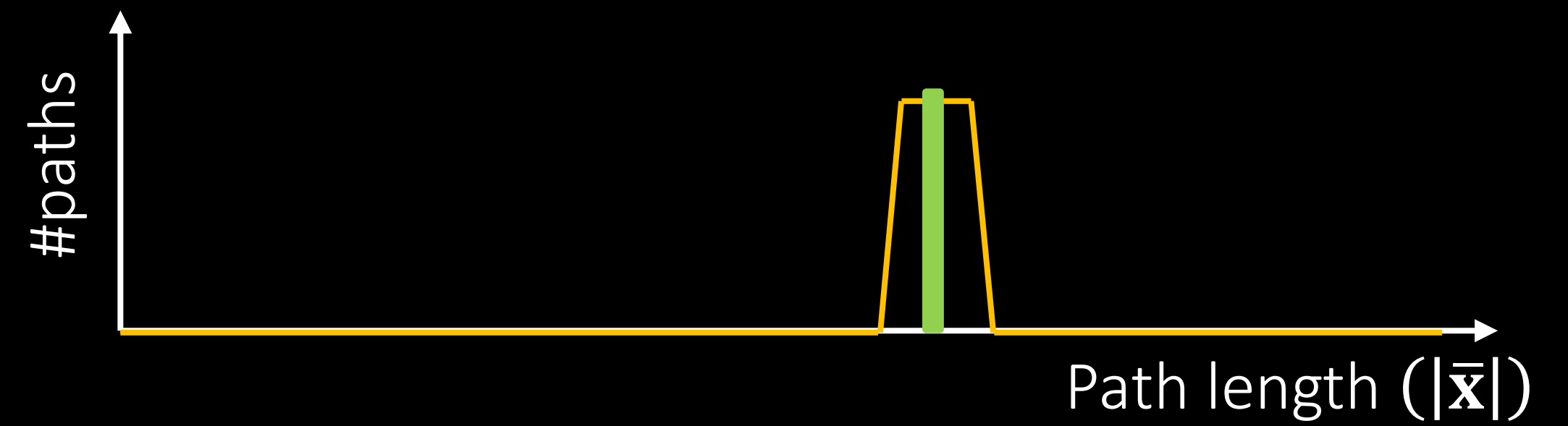
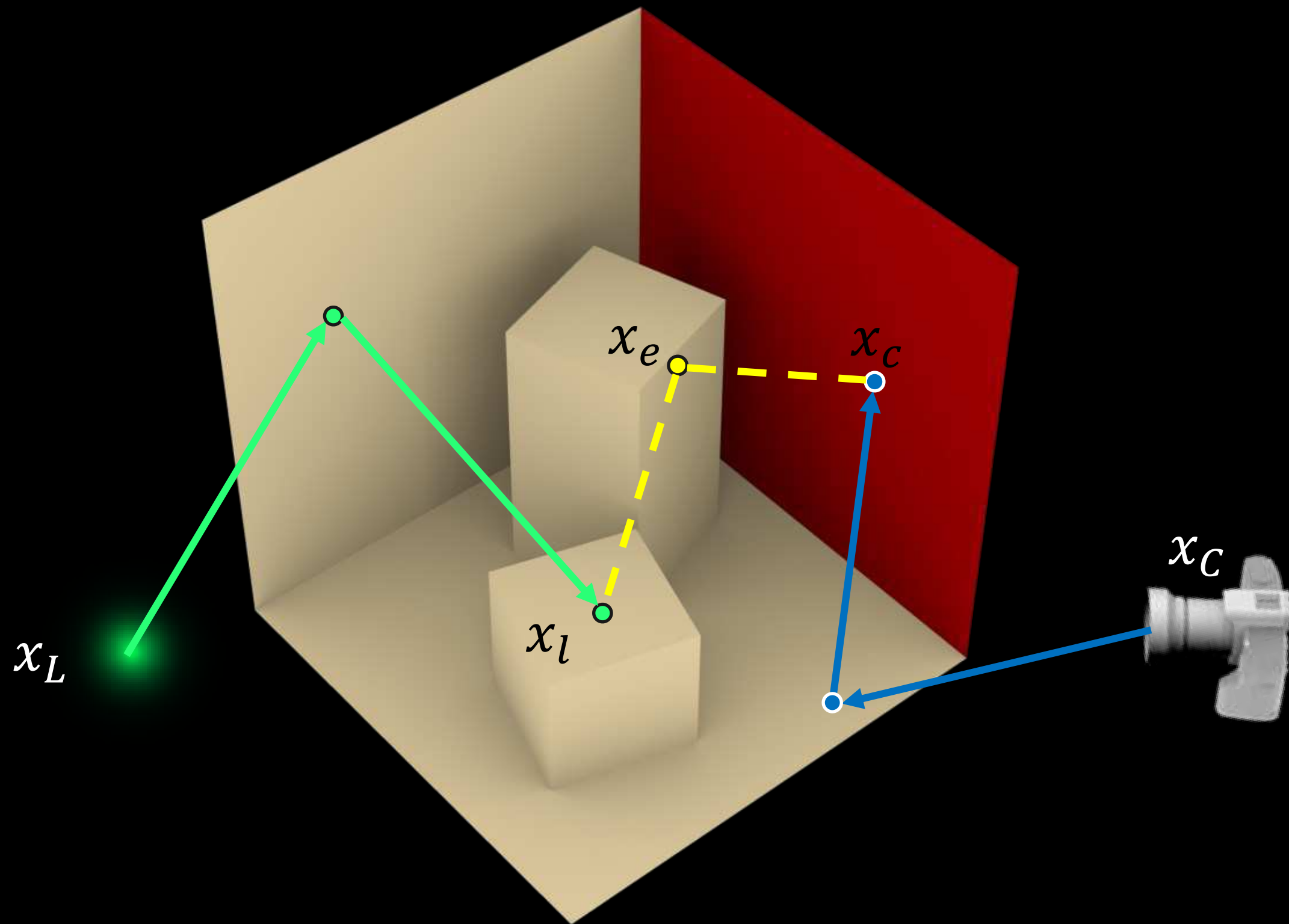
step 2: generate path with target length $|\bar{x}|$

bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join x_l and x_c via connecting vertex (x_e)

Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

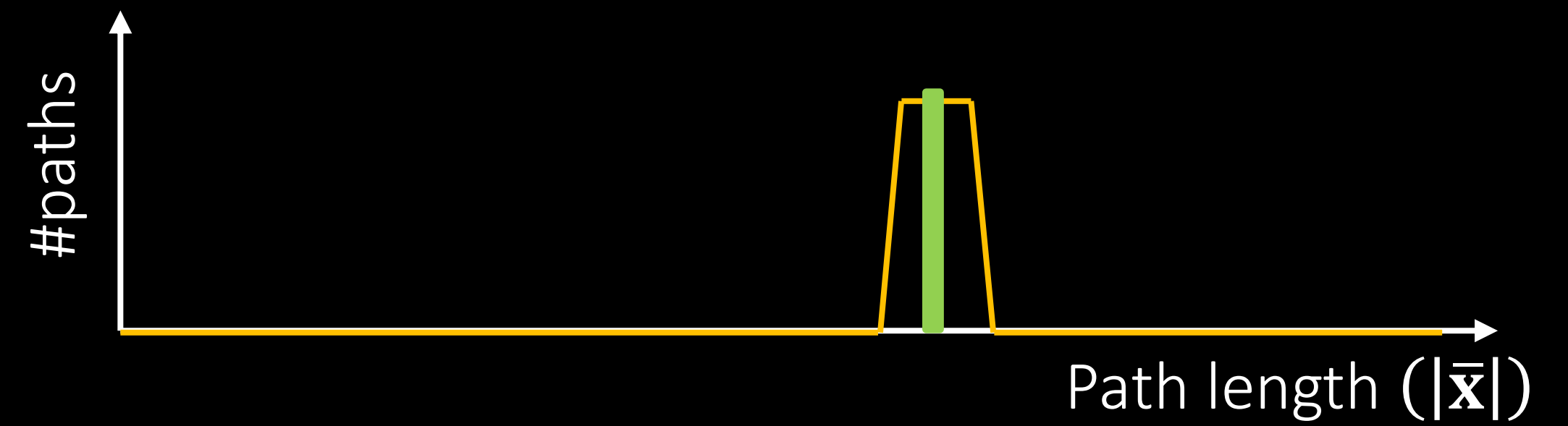
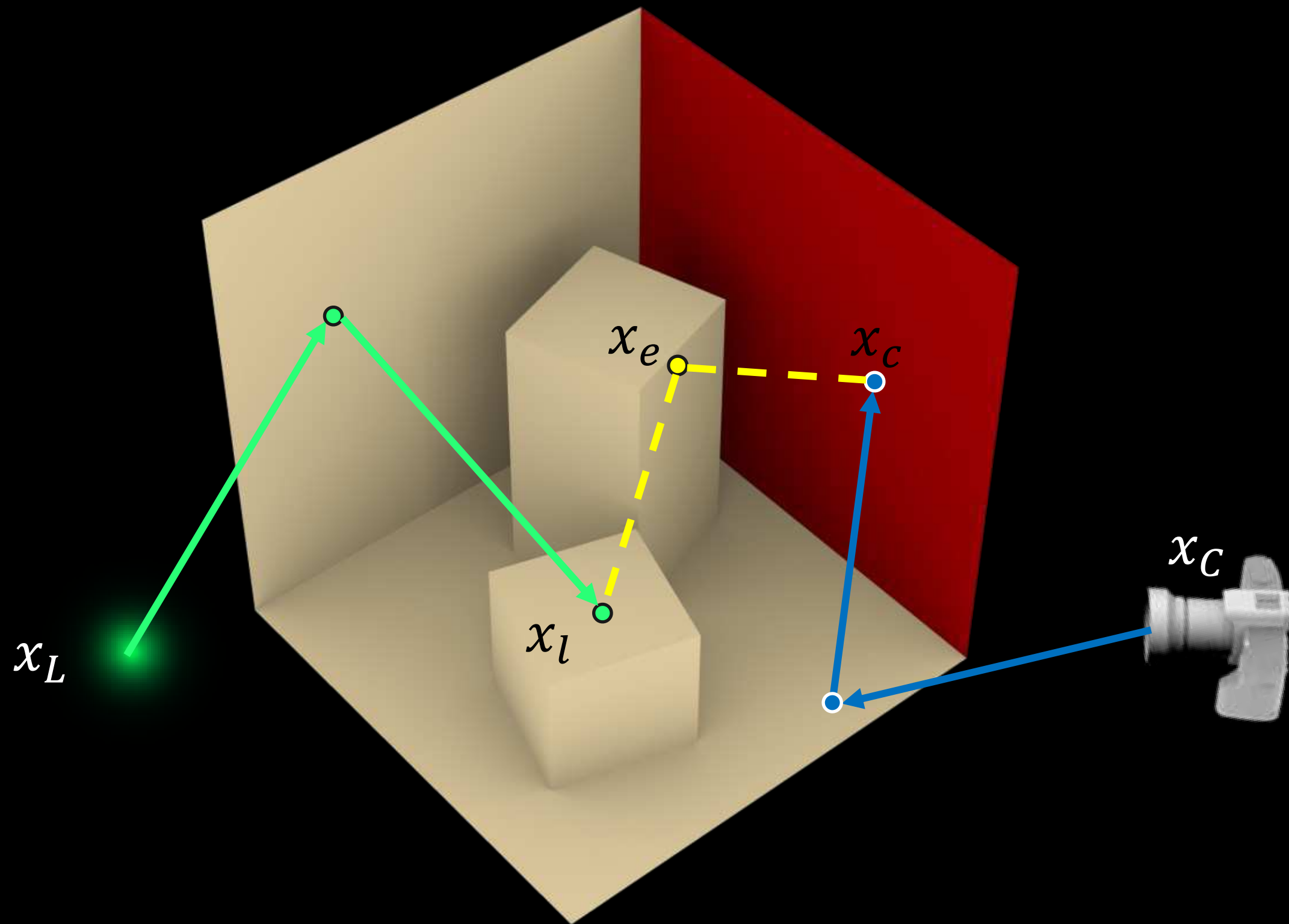
bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join x_l and x_c via connecting vertex (x_e)

$$\left. \begin{array}{l} \text{source sub-path length} + \\ \text{sensor sub-path length} + \\ |x_e \rightarrow x_l| + |x_e \rightarrow x_c| \end{array} \right\} = |\bar{x}|$$

Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

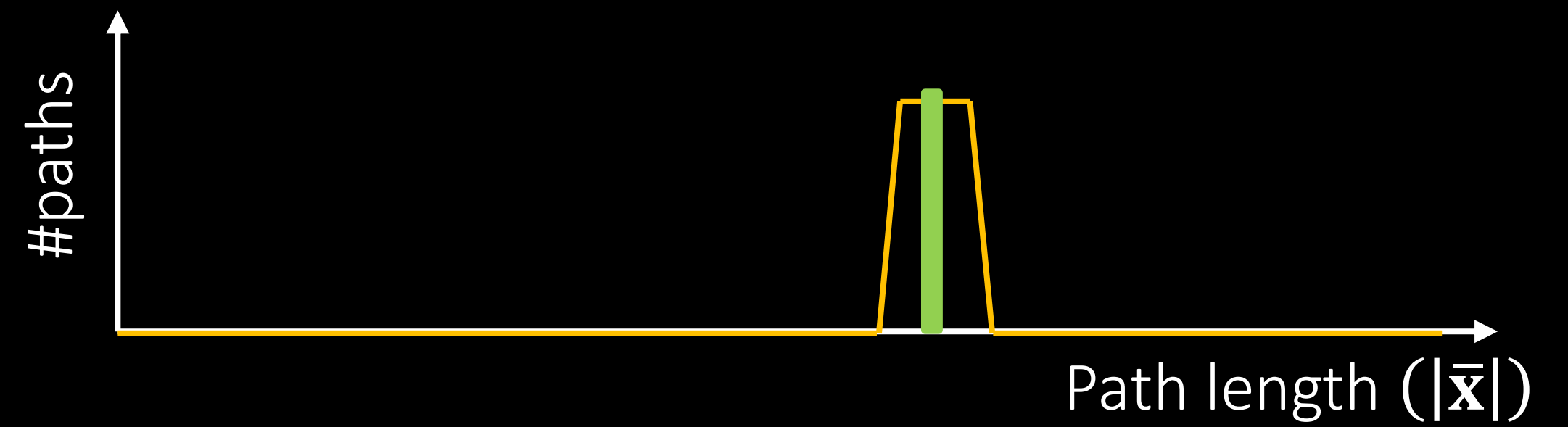
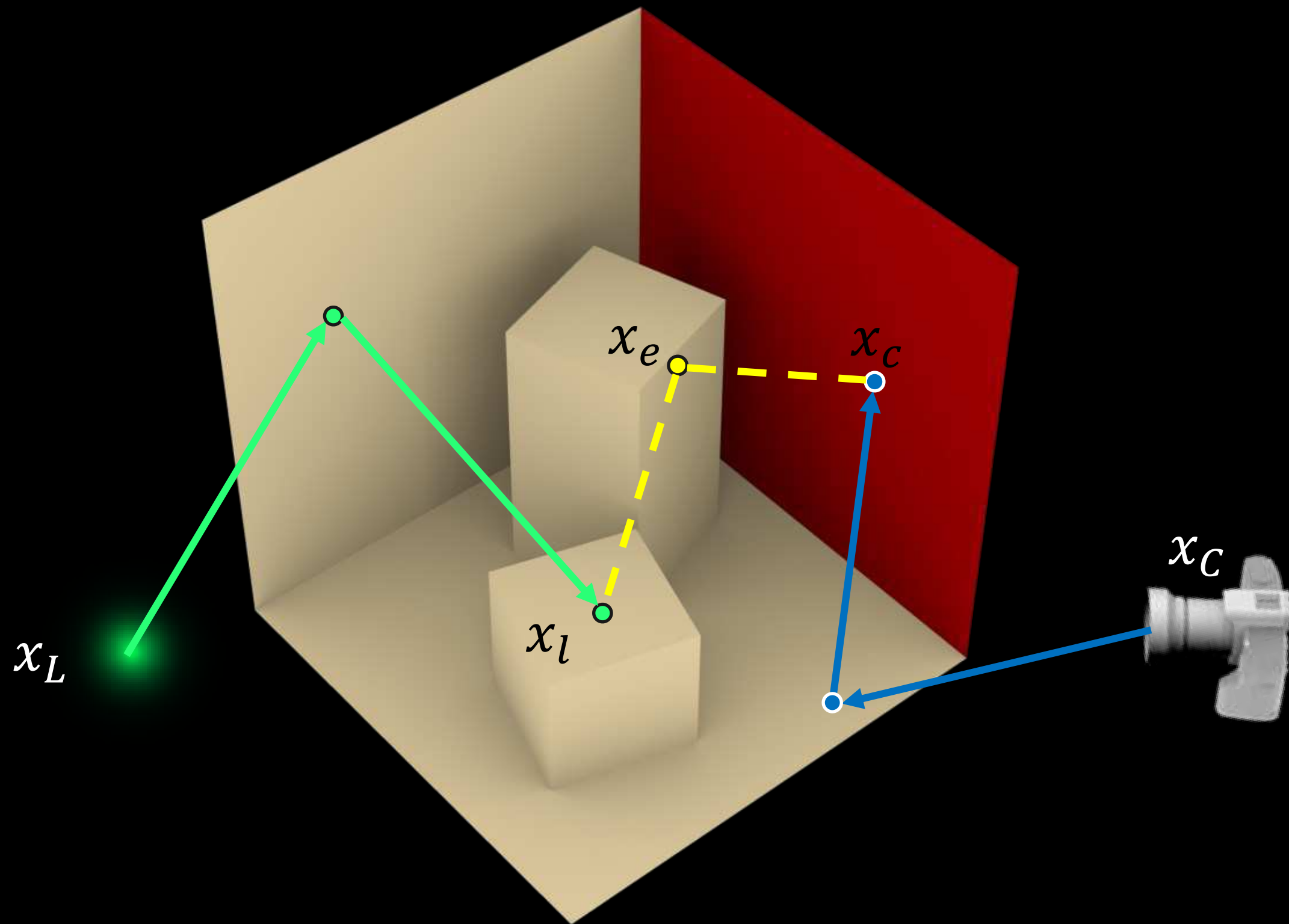
bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join x_l and x_c via connecting vertex (x_e)

$$|x_e \rightarrow x_l| + |x_e \rightarrow x_c| = |\bar{x}| - \begin{matrix} \text{source + sensor} \\ \text{sub-path lengths} \end{matrix}$$

Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

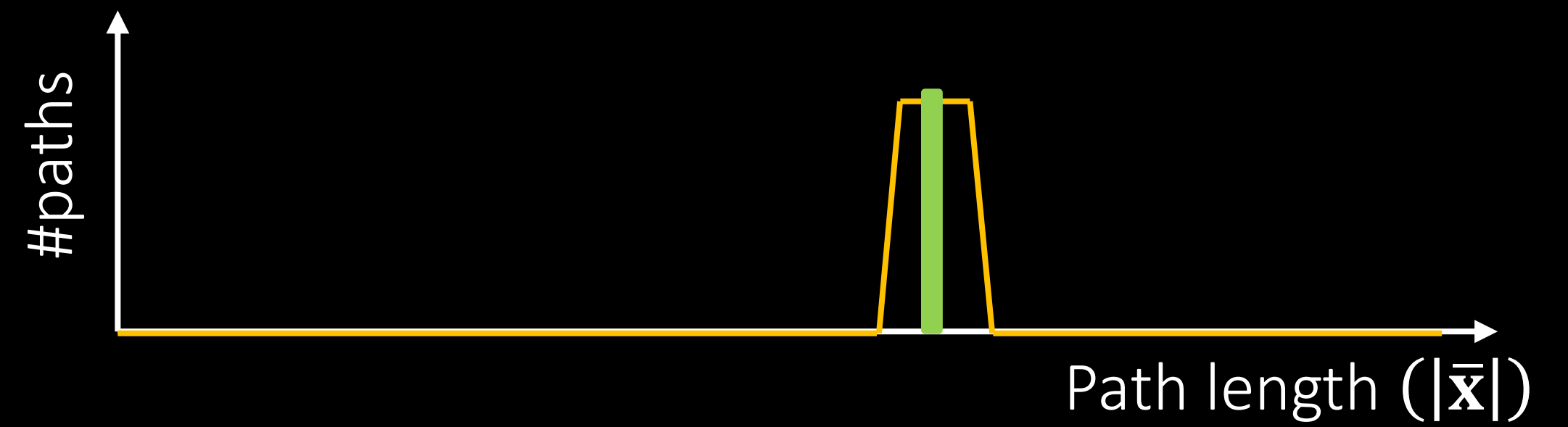
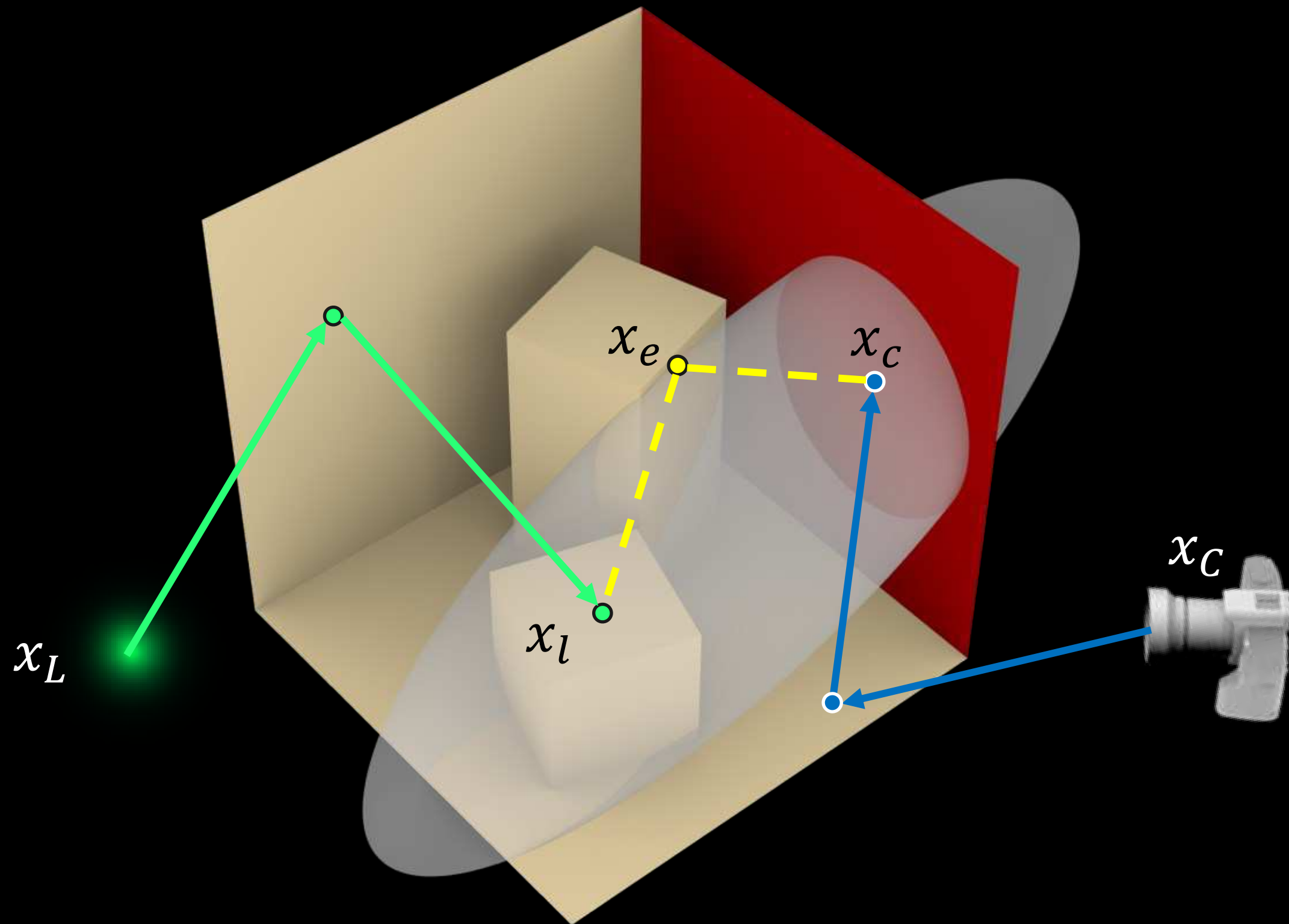
bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

join x_l and x_C via connecting vertex (x_e)

Definition of an ellipsoid: $|x_e \rightarrow x_l| + |x_e \rightarrow x_C| = |\bar{x}| - \text{source + sensor sub-path lengths}$

Path sampling for time-gated case



step 1: sample path length $|\bar{x}|$

step 2: generate path with target length $|\bar{x}|$

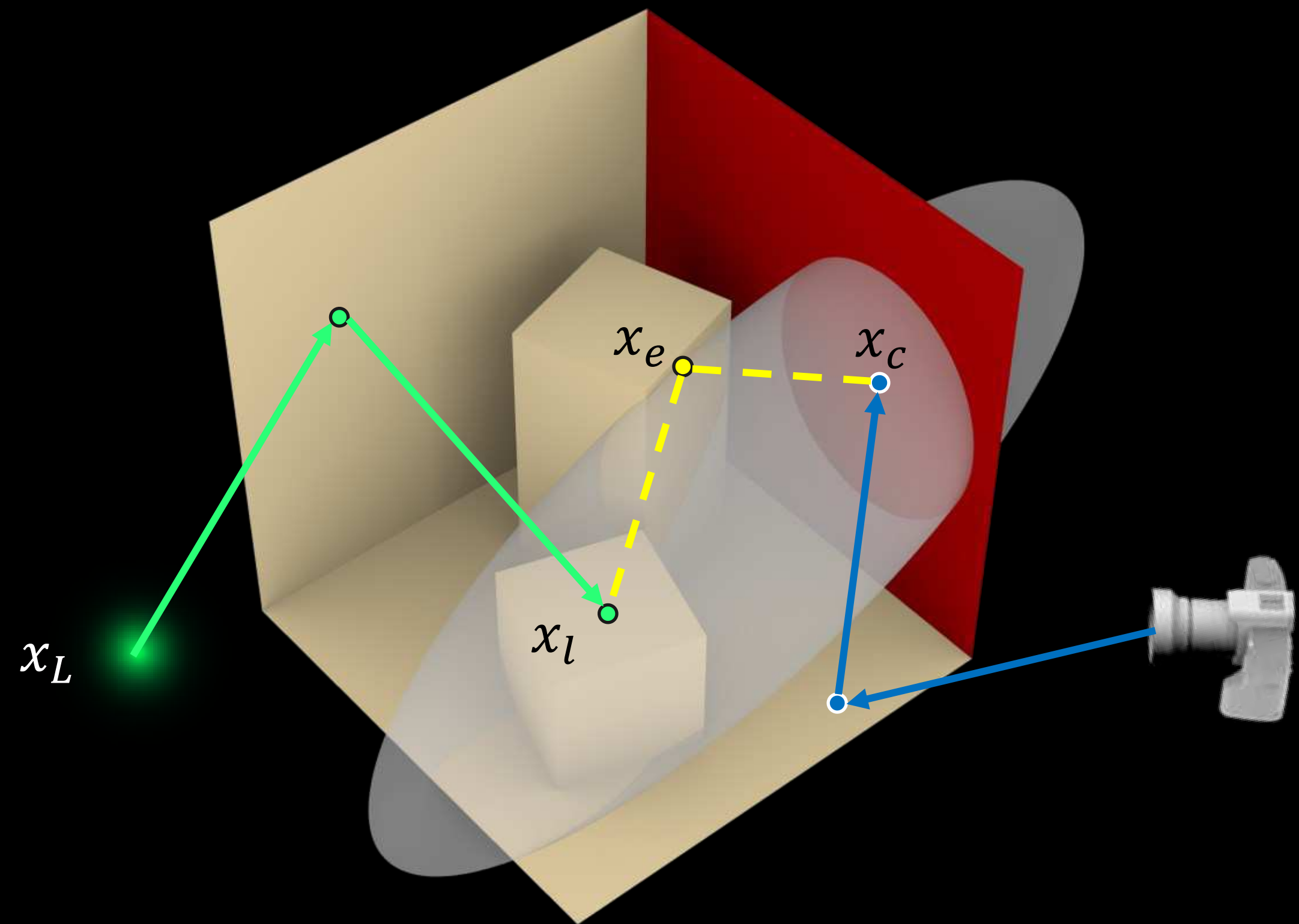
bidirectional path tracing (BDPT)

generate light sub-path, camera sub-path

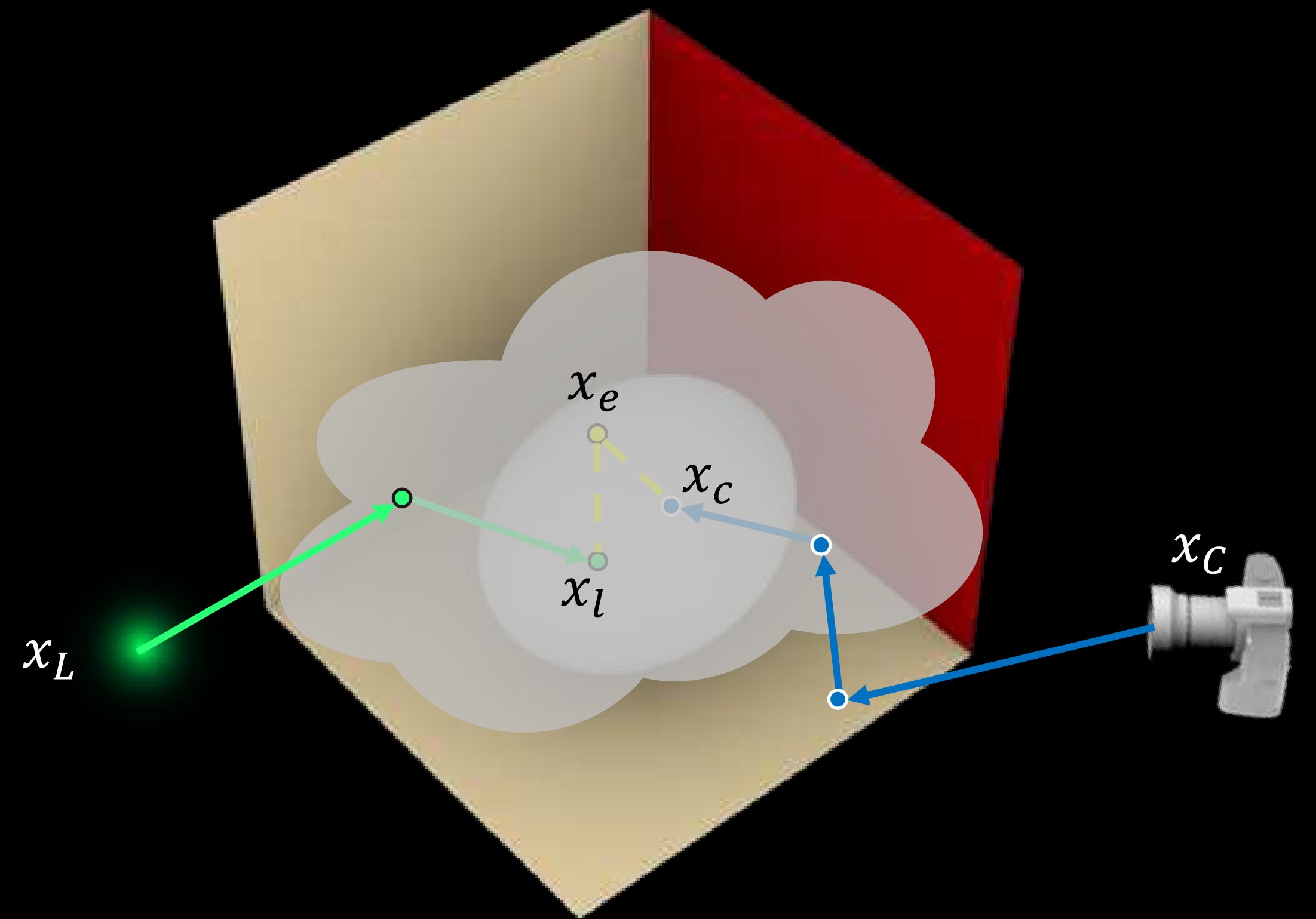
join x_l and x_c via connecting vertex (x_e)

Definition of an ellipsoid: $|x_e \rightarrow x_l| + |x_e \rightarrow x_c| = |\bar{x}| - \text{source + sensor sub-path lengths}$

Path sampling for time-gated case

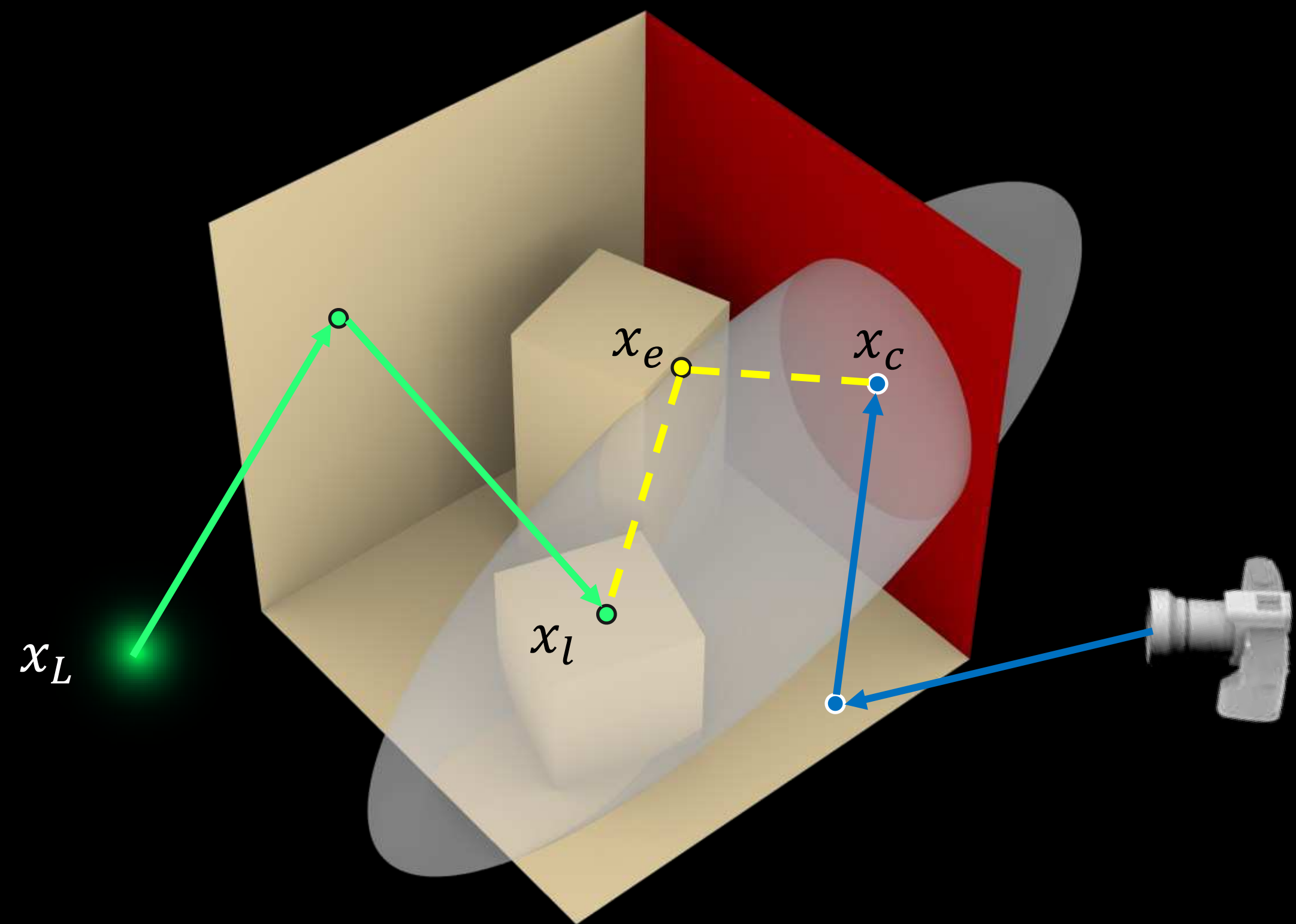


surface case

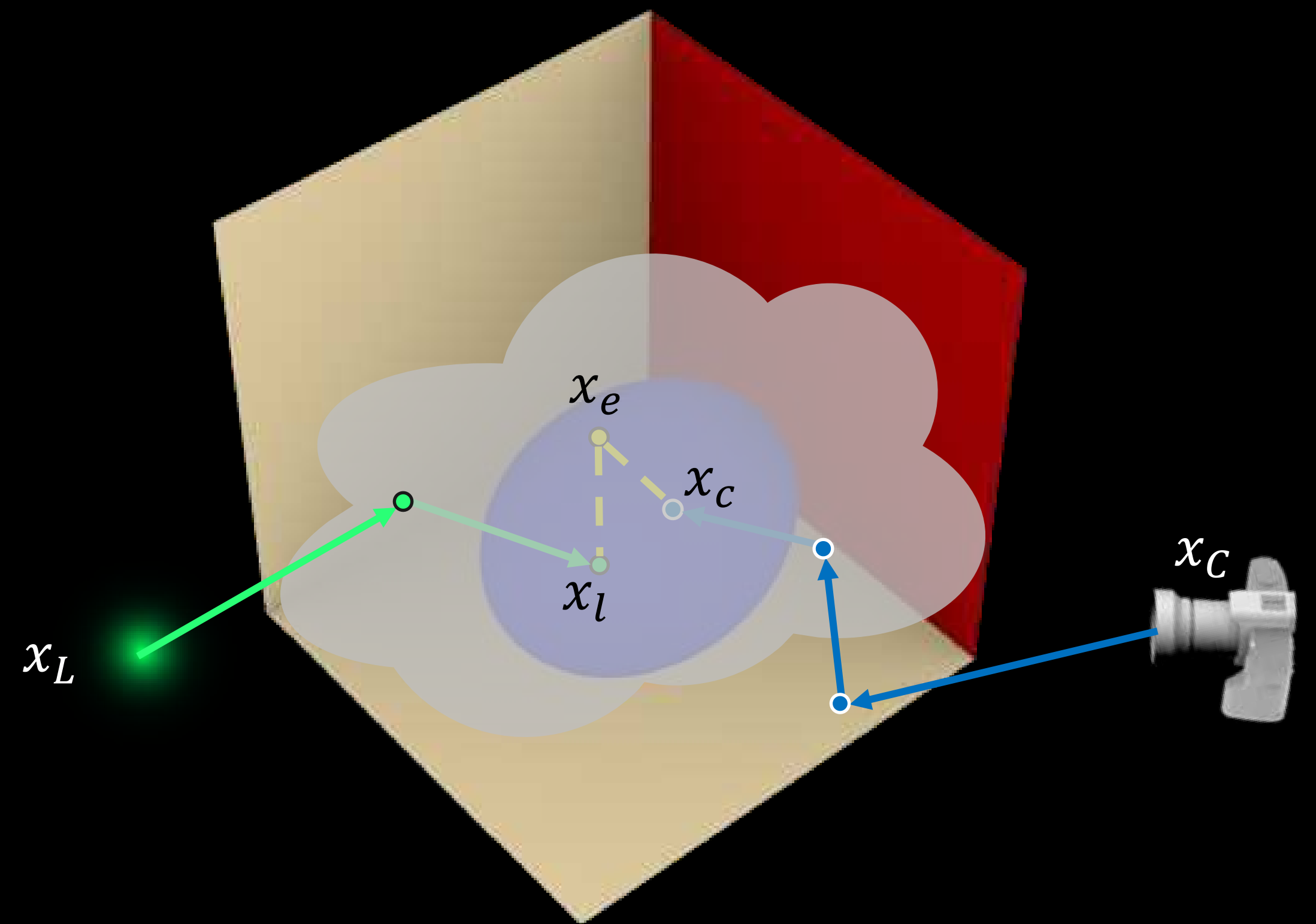


volumetric case

Path sampling for time-gated case

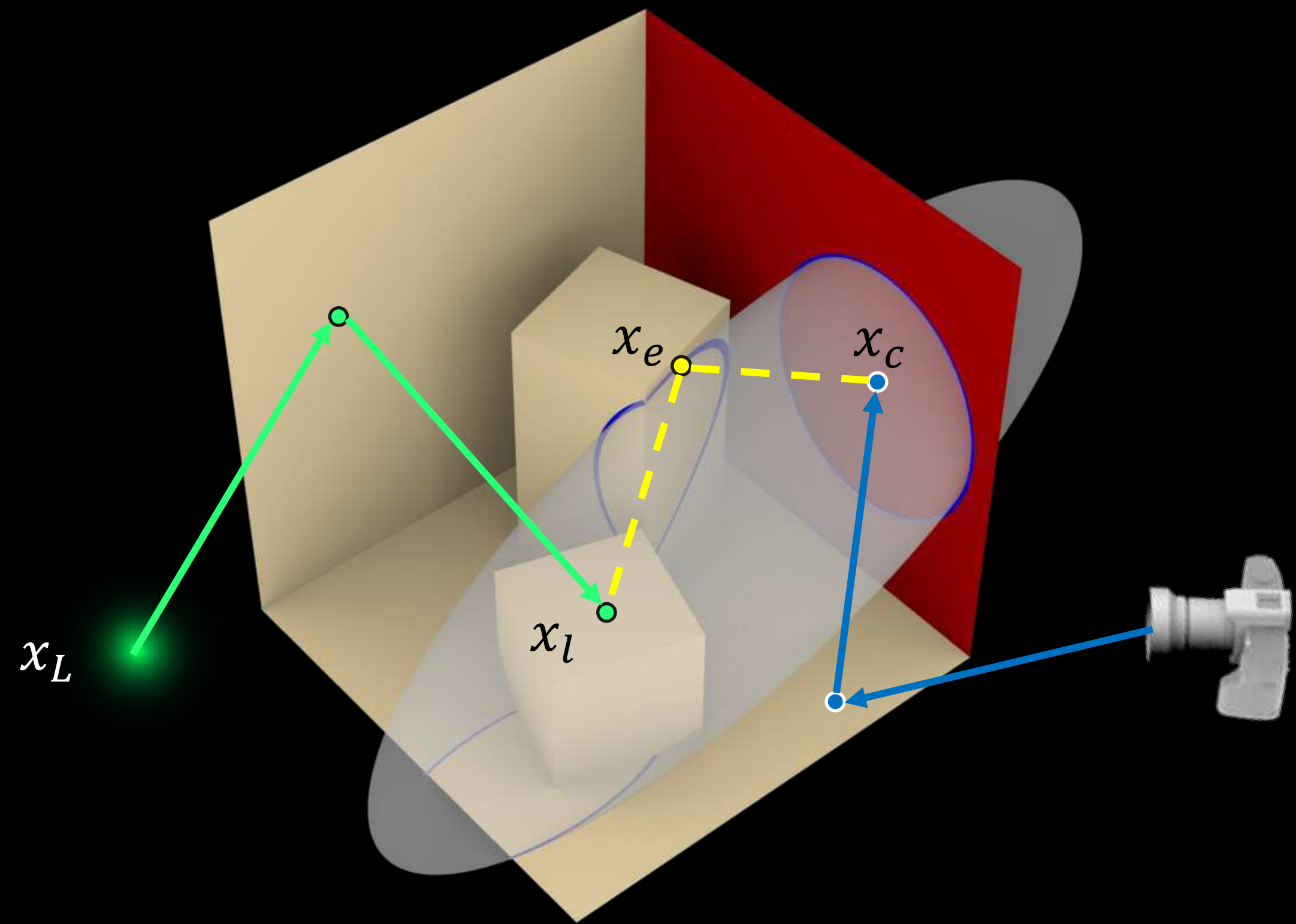


surface case

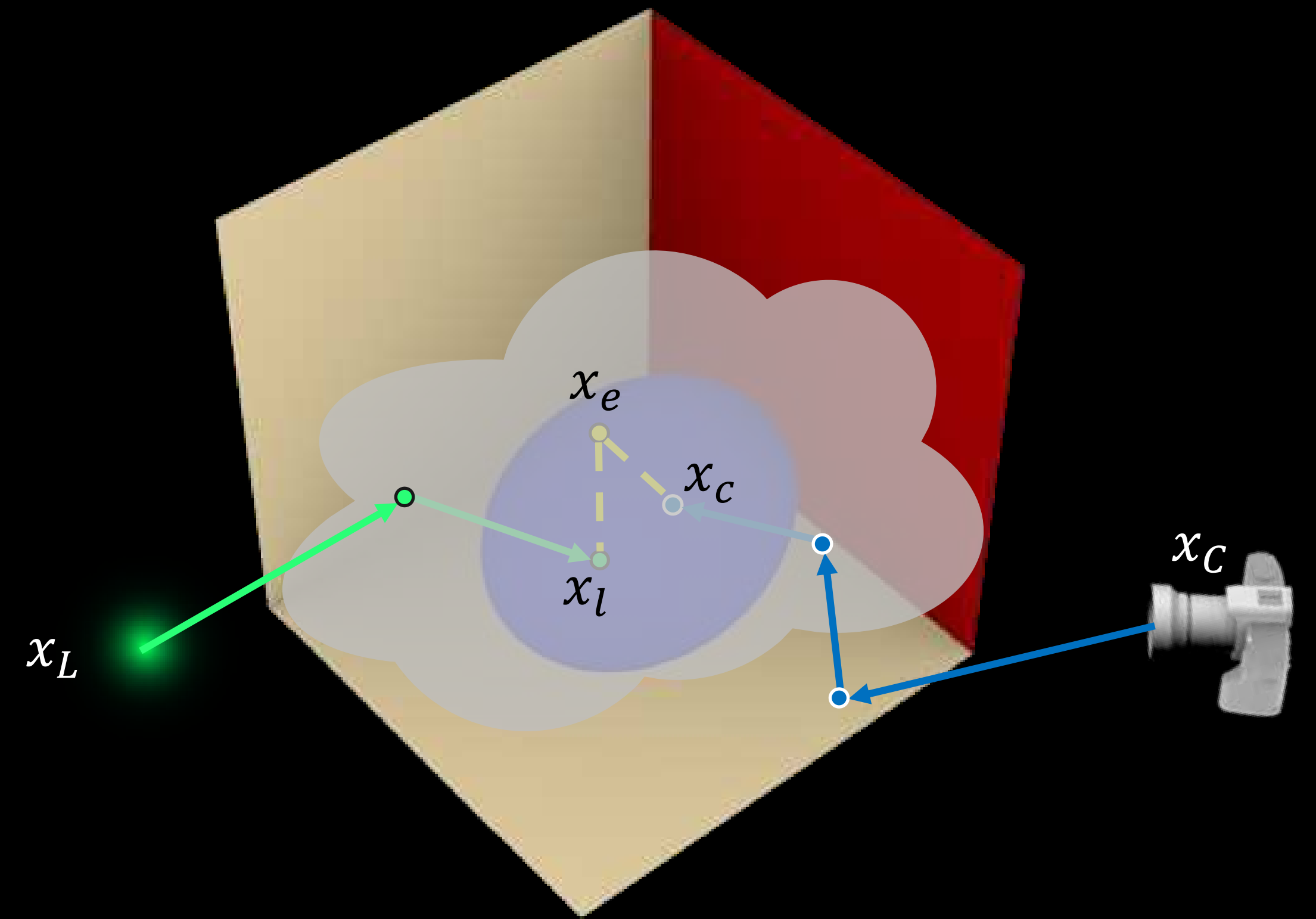


volumetric case

Path sampling for time-gated case



surface case



volumetric case

Application: proximity detector for cars



proximity detected

virtual light curtain

Application: proximity detector for cars

road scene

standard BDPT

BDPT w/ ellipsoidal
connections

Gate width: 200 ps (1.14% scene)
Rendering time: 10s per frame

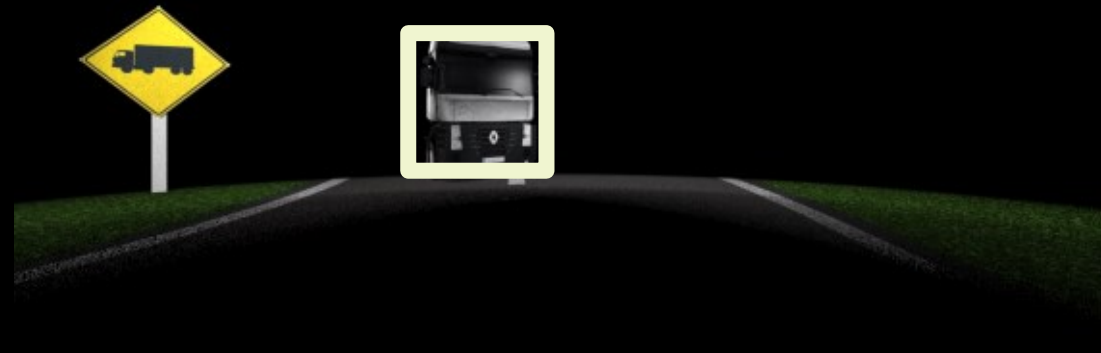
Application: proximity detector for cars

road scene

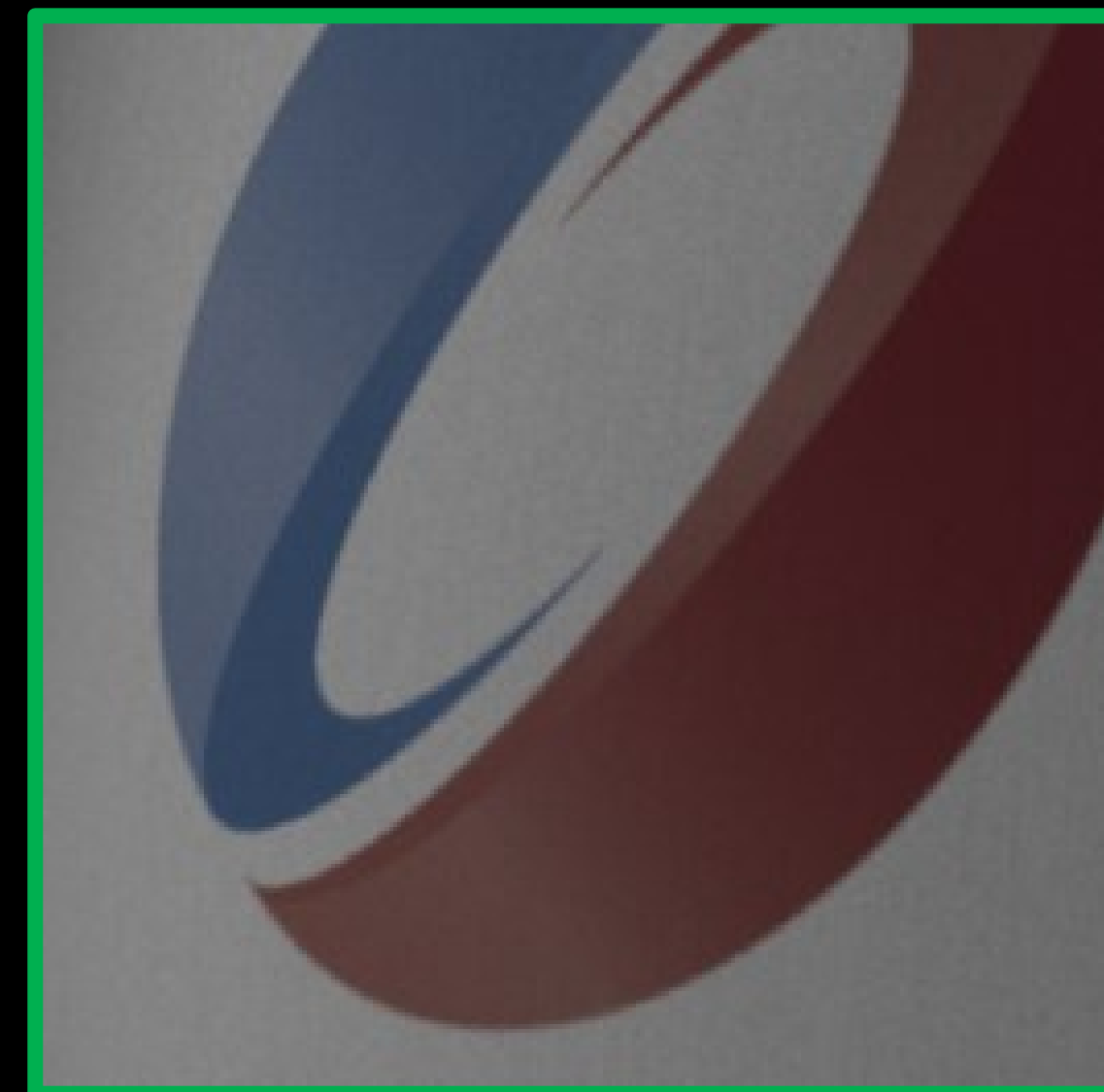
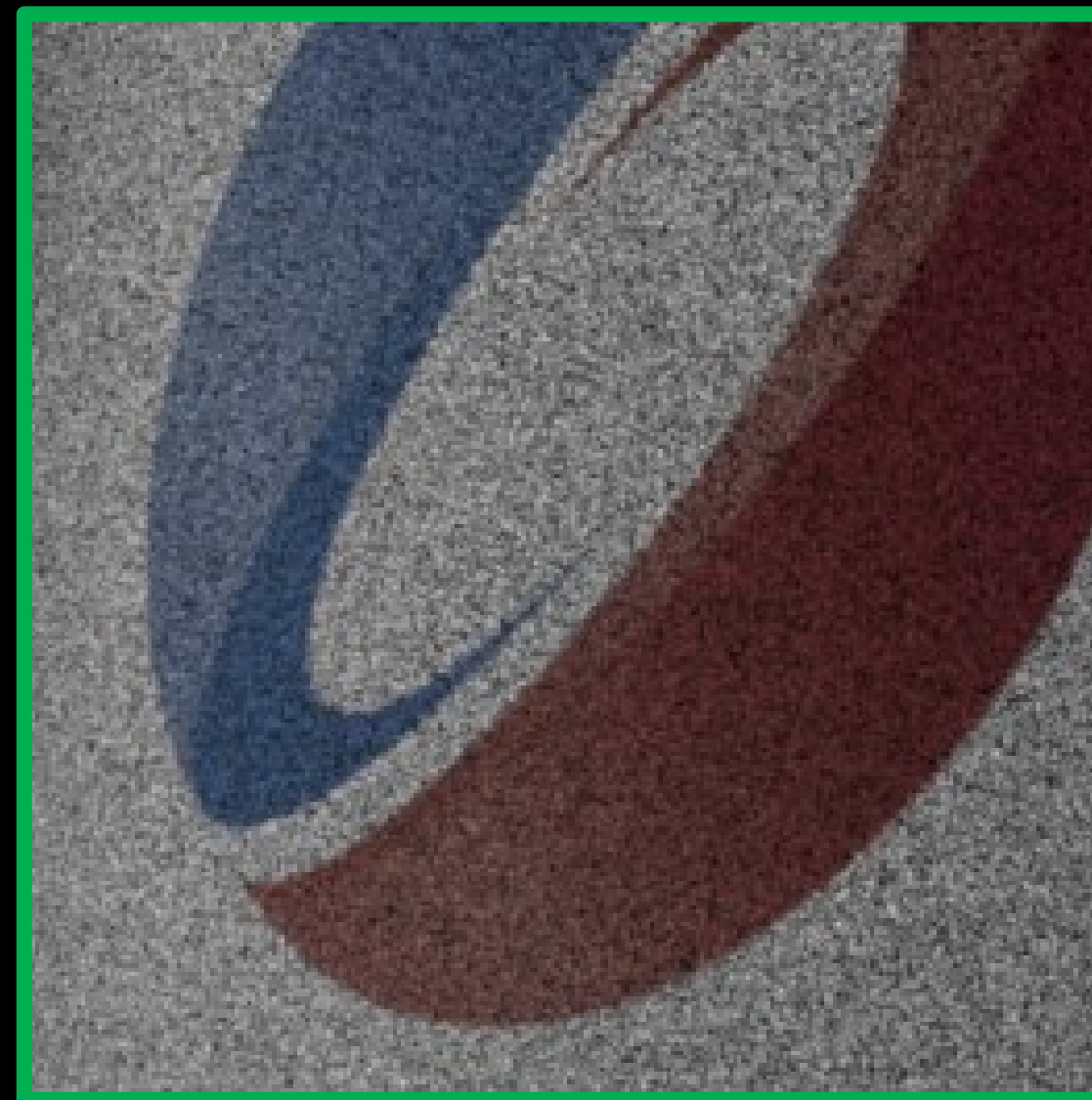
standard BDPT

BDPT w/ ellipsoidal connections

time = 1.34s

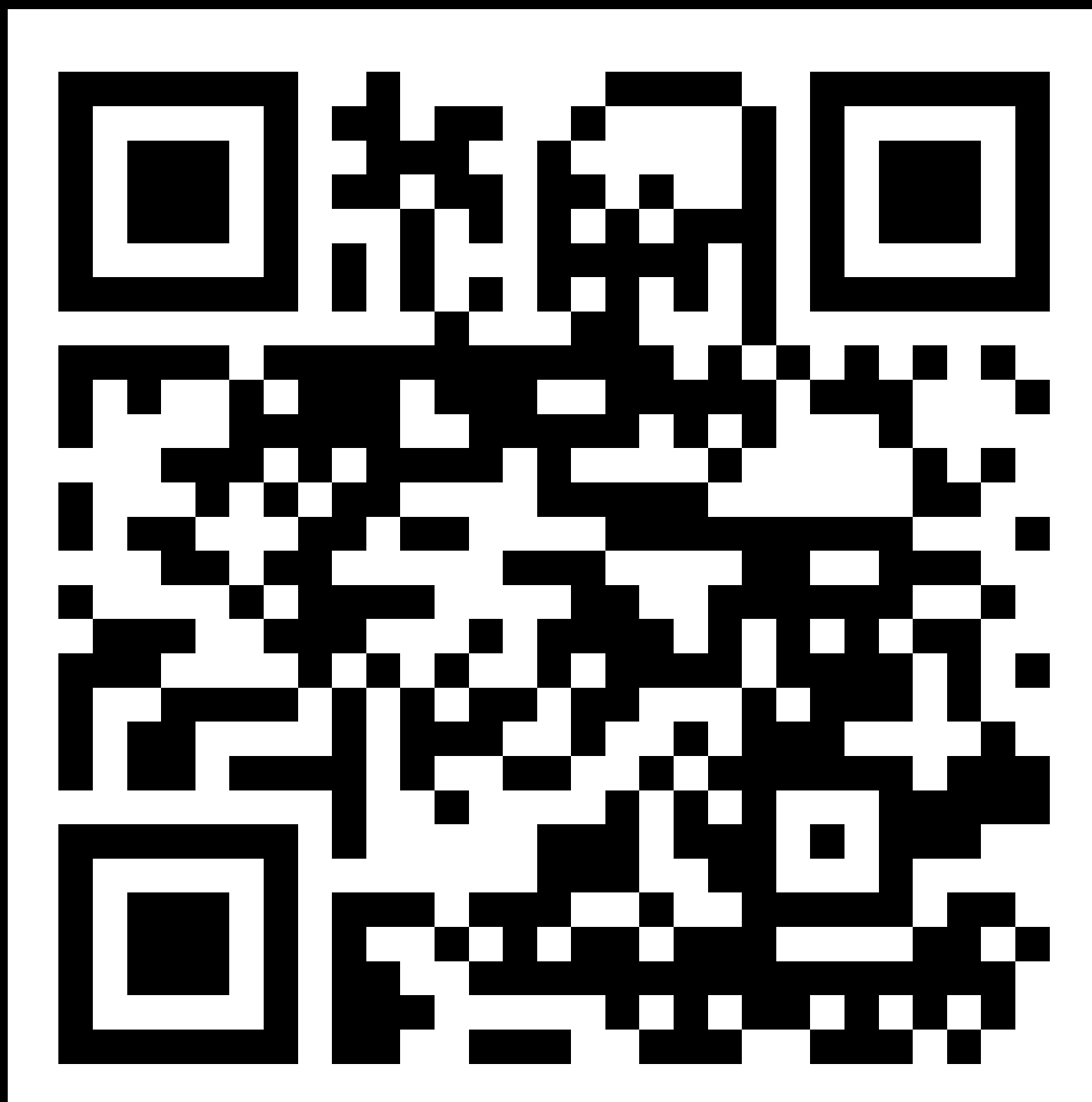


time = 1.74s



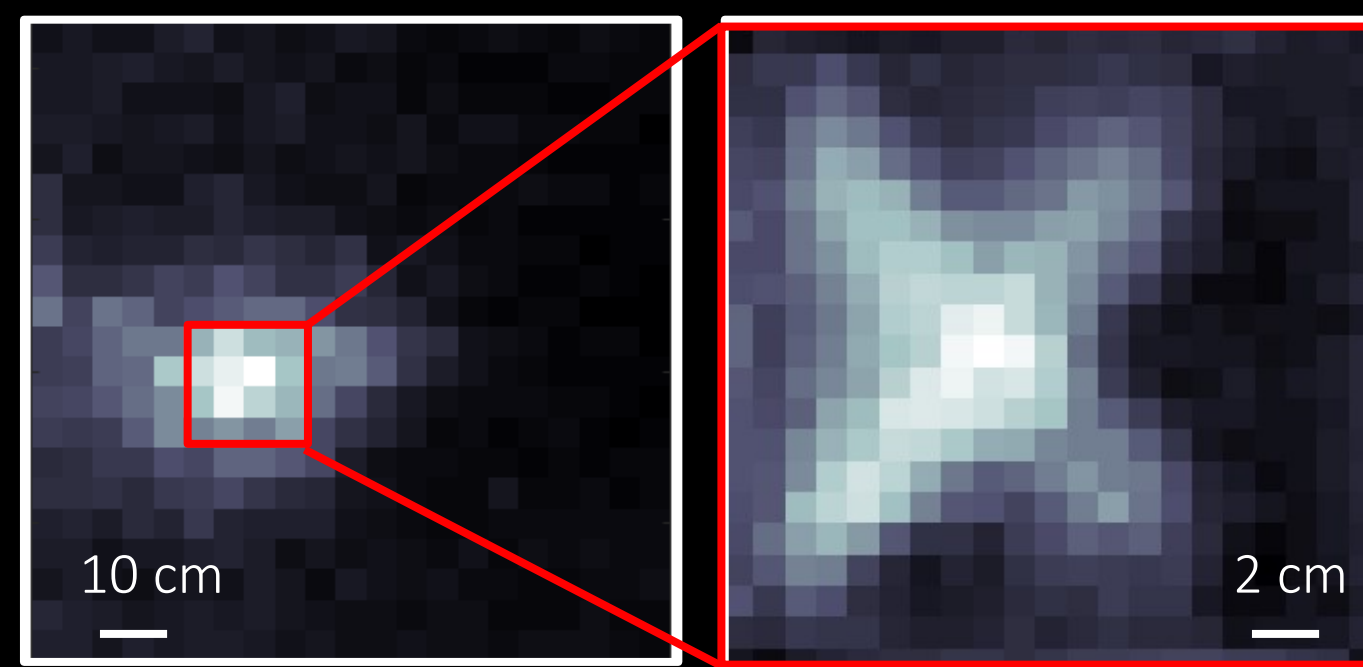
Imaging projects using this renderer

Mitsuba based
open source implementation

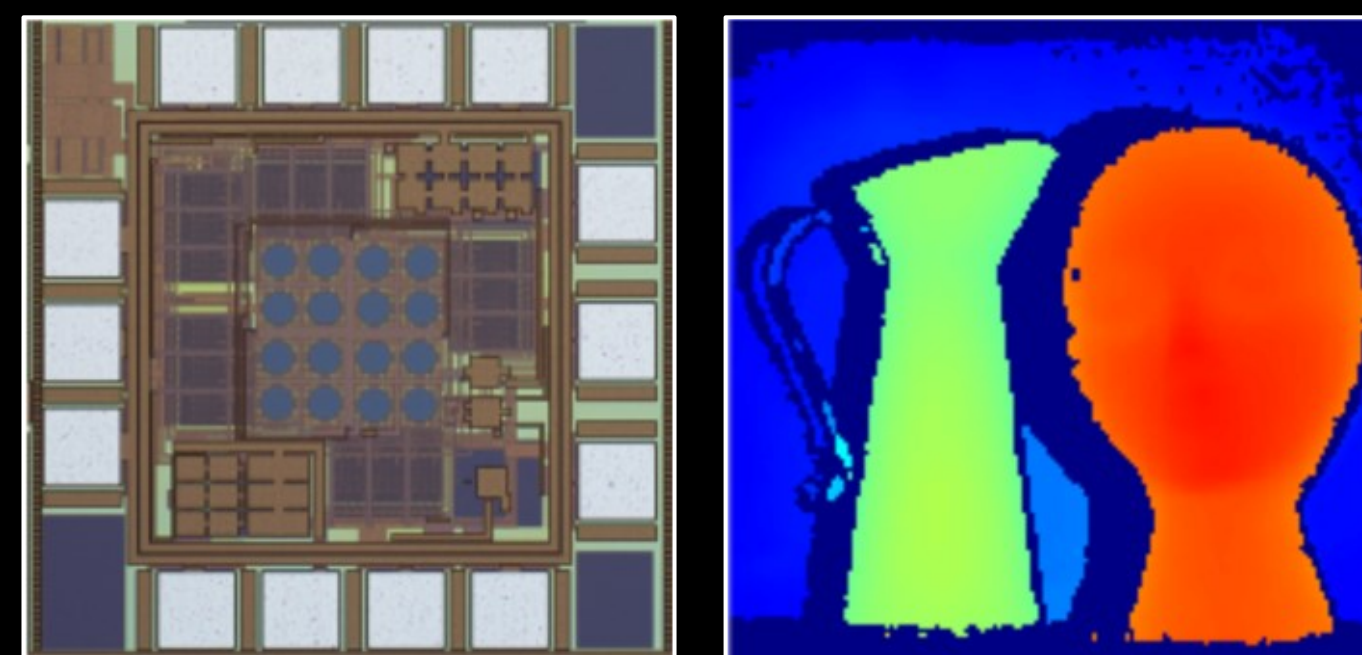


<http://bit.ly/32Uzm1m>

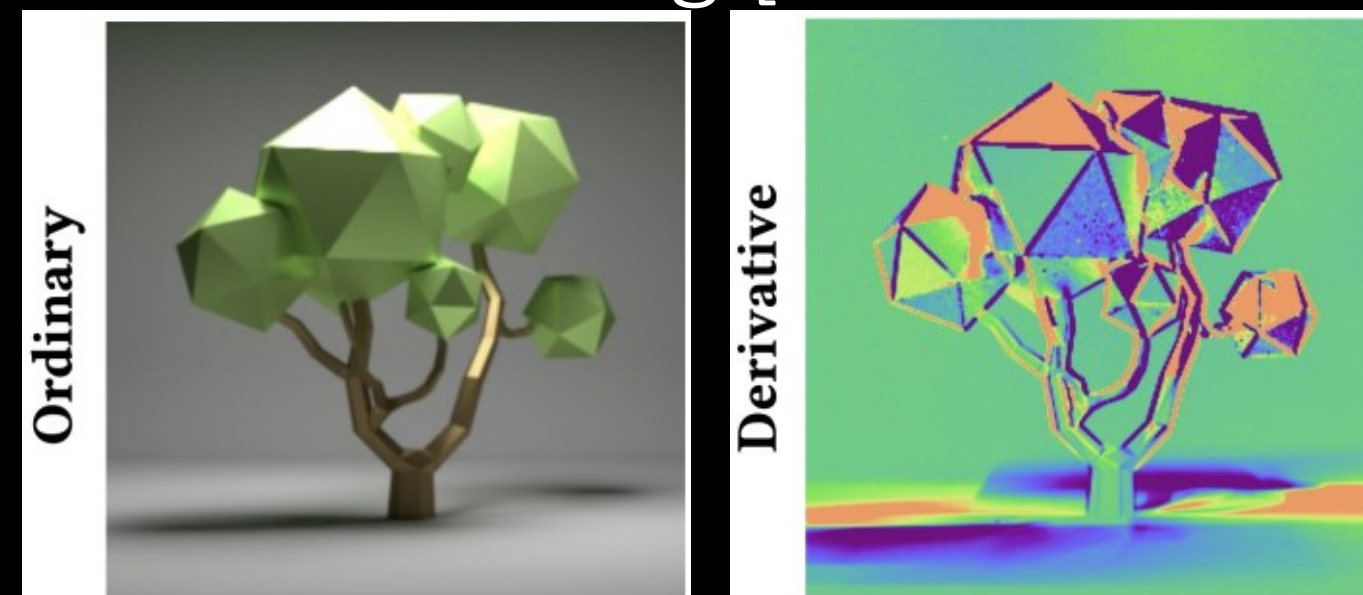
non-line-of-sight



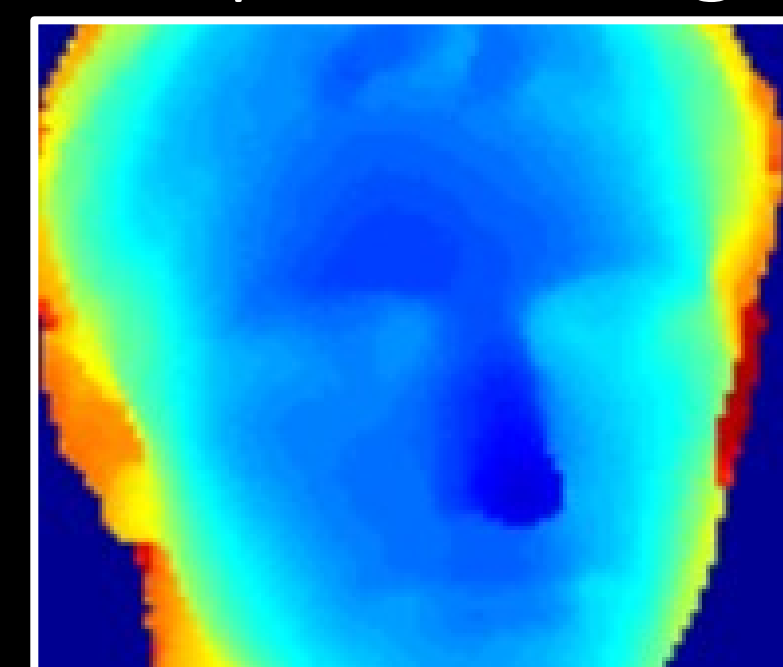
sensor design [White et al. 2022]



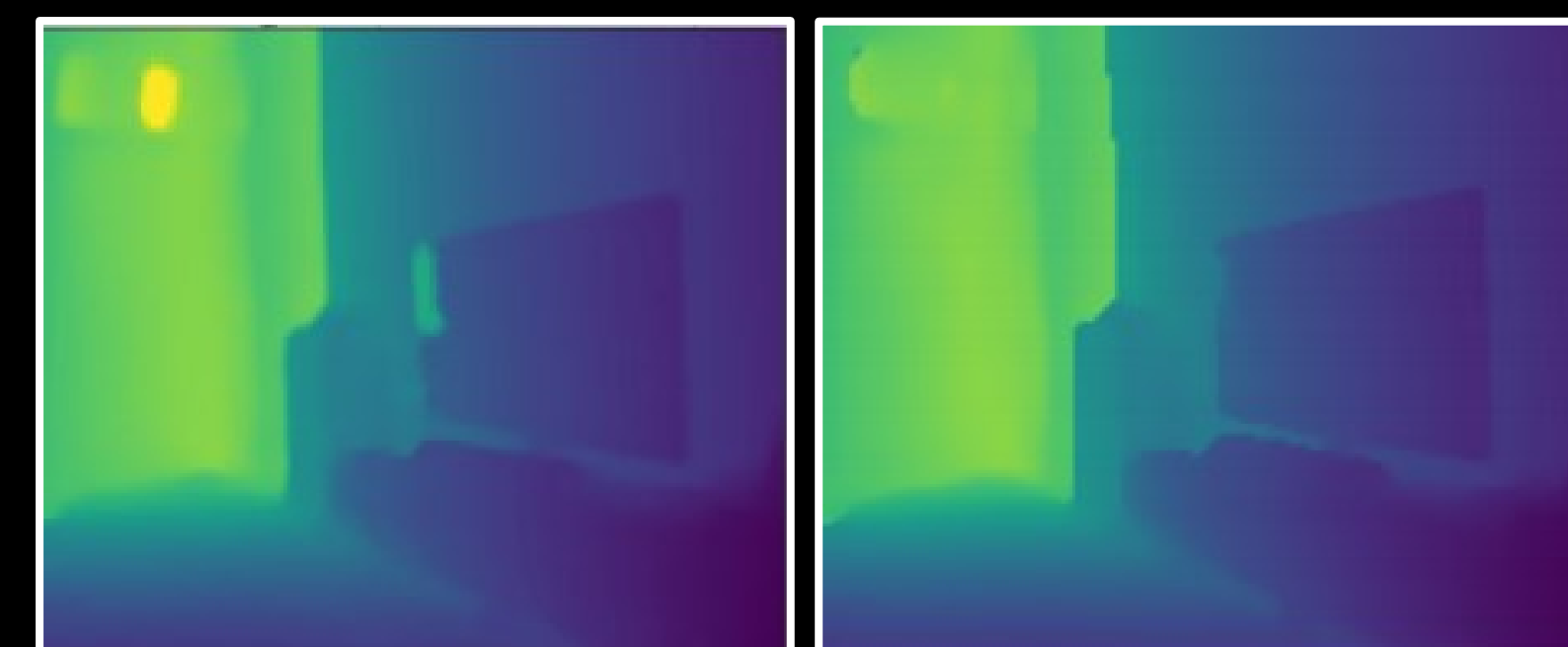
inverse rendering [Wu et al. 2021]



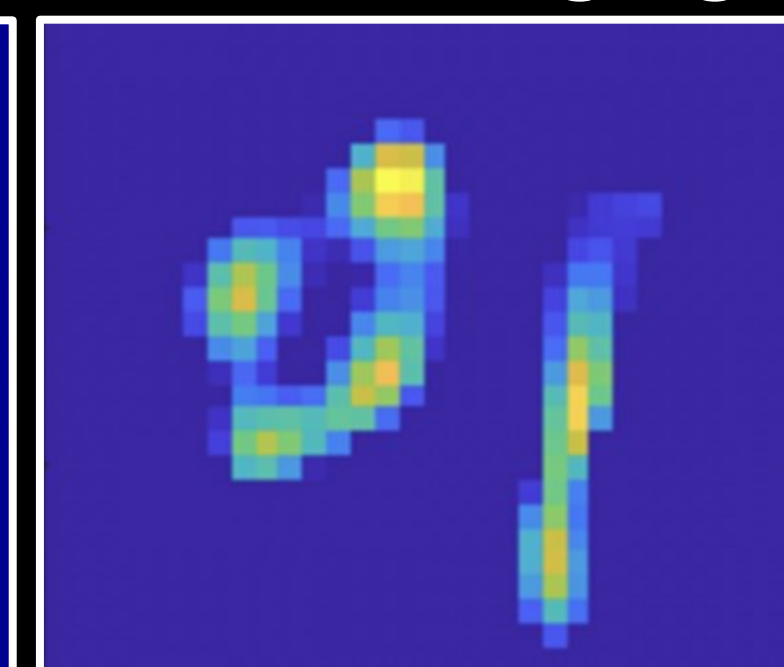
depth sensing



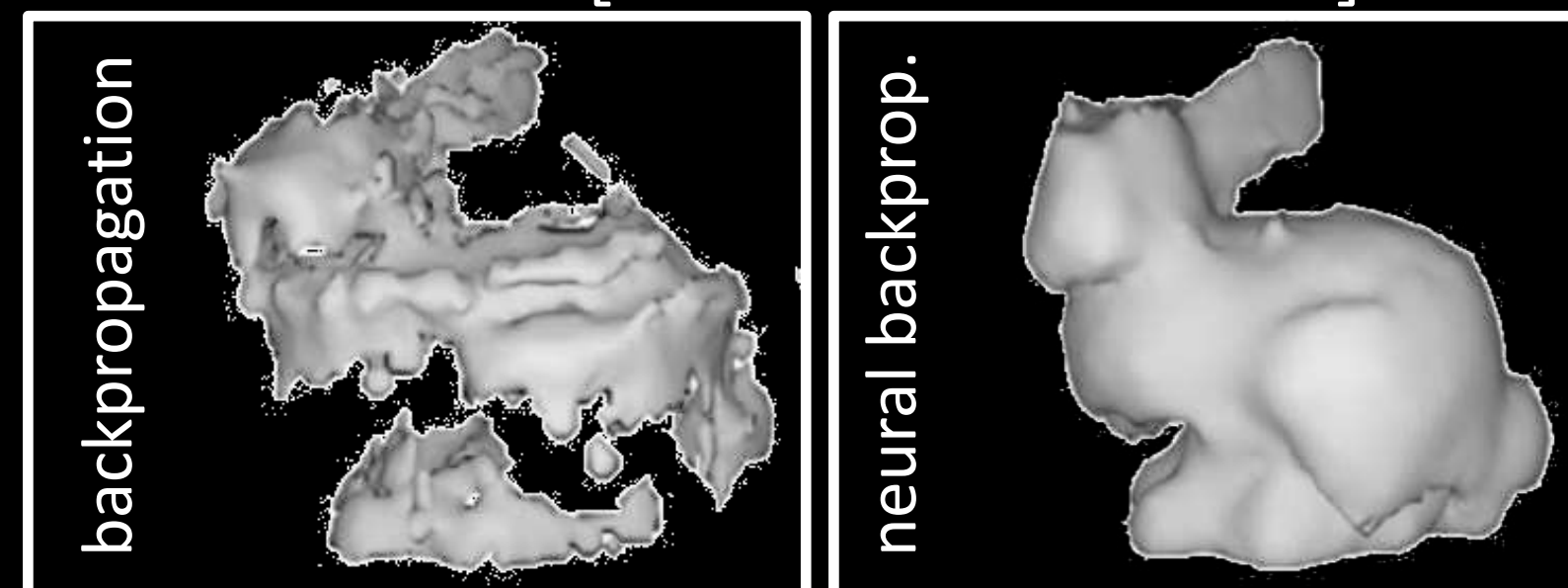
deep learning [Barragan et al. 2021]



tissue imaging

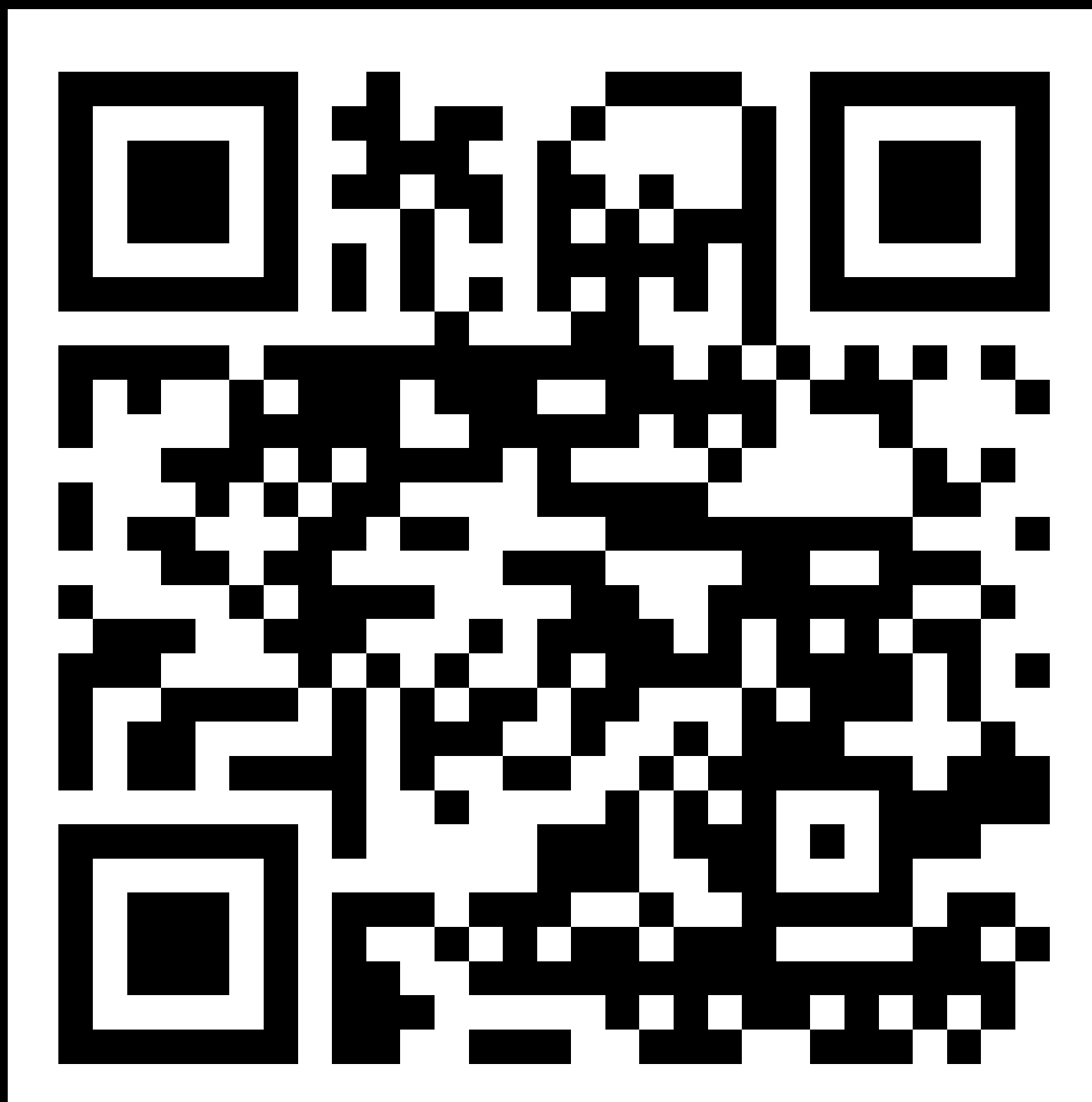


SONAR [Reed et al. 2023]

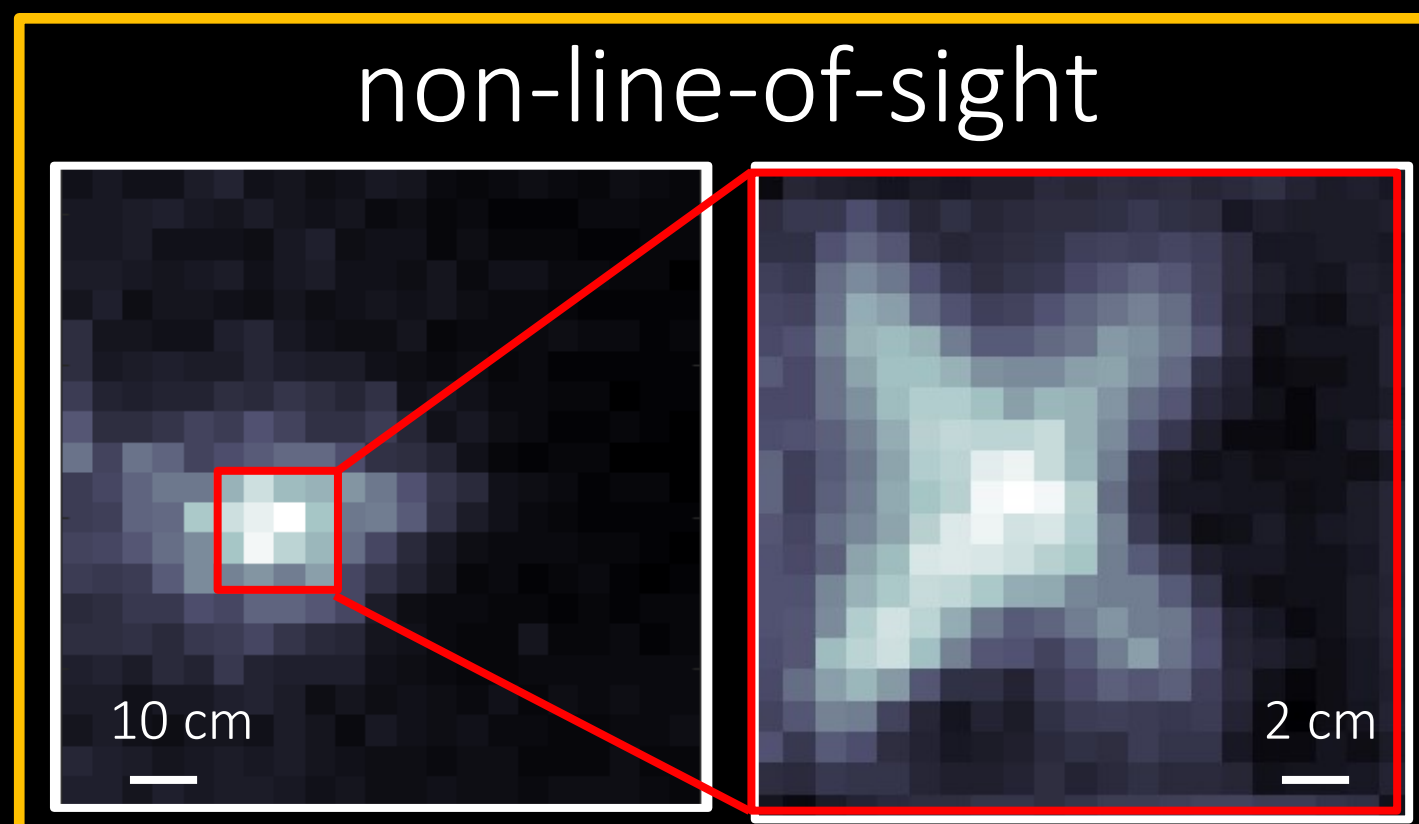


Imaging projects using this renderer

Mitsuba based
open source implementation

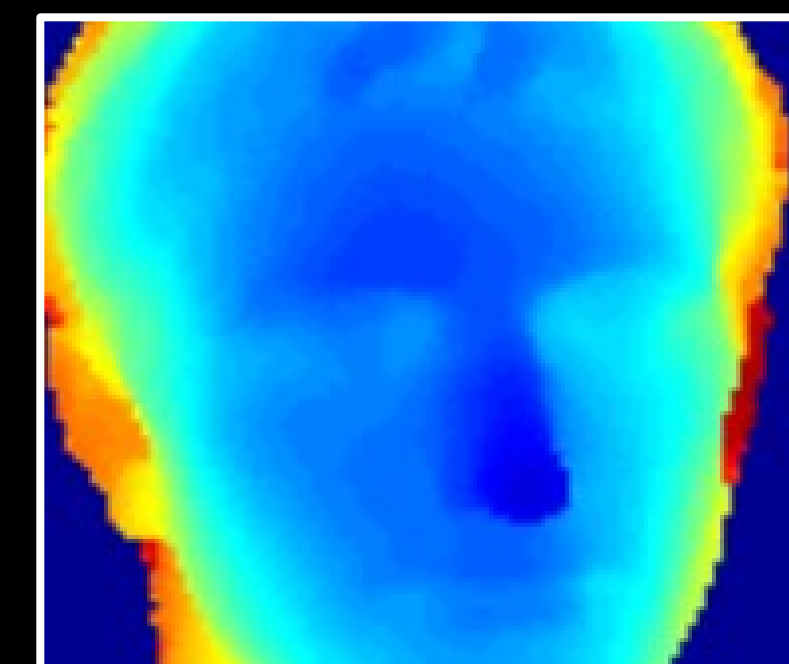


<http://bit.ly/32Uzm1m>

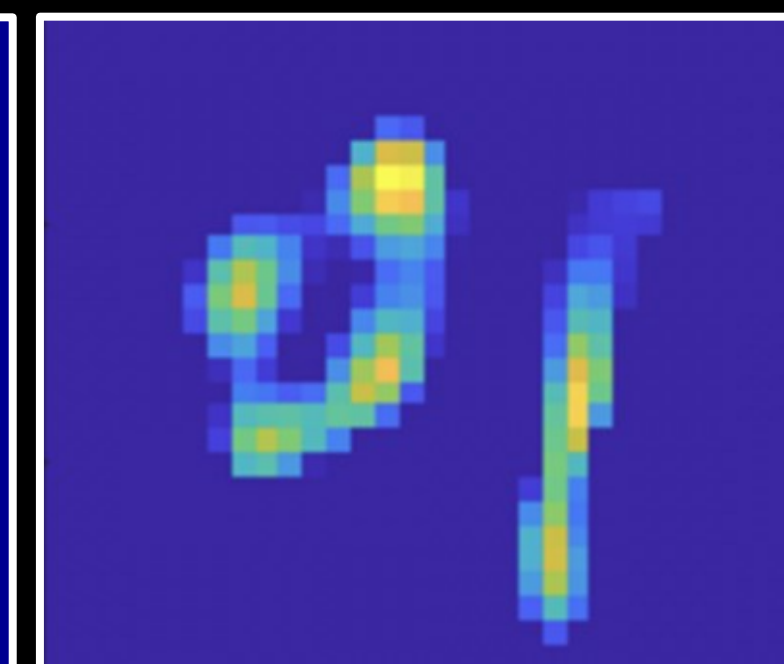


sensor design [White et al. 2022]

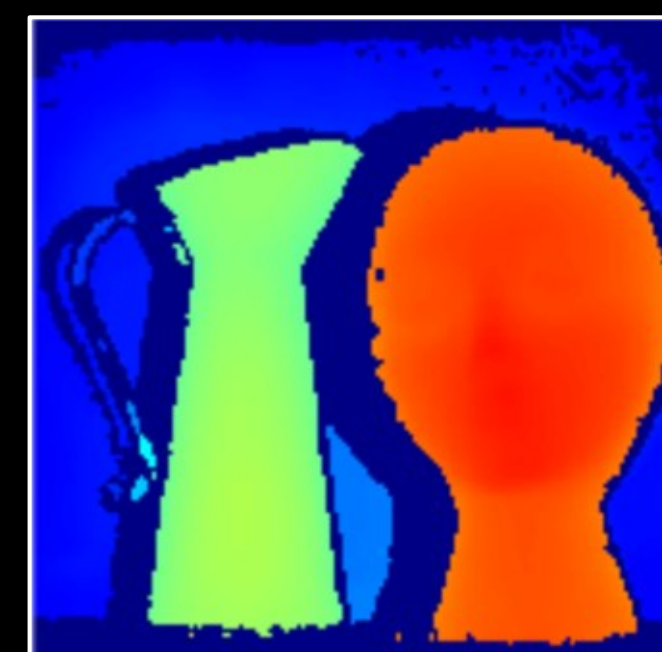
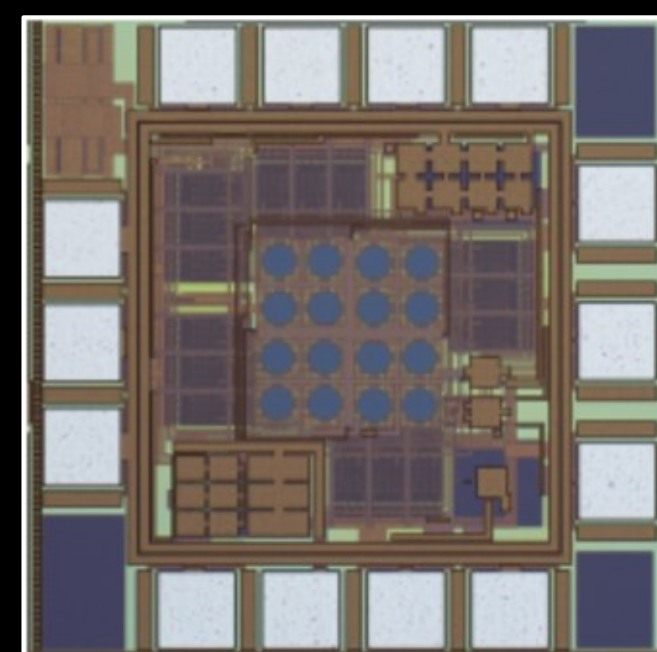
depth sensing



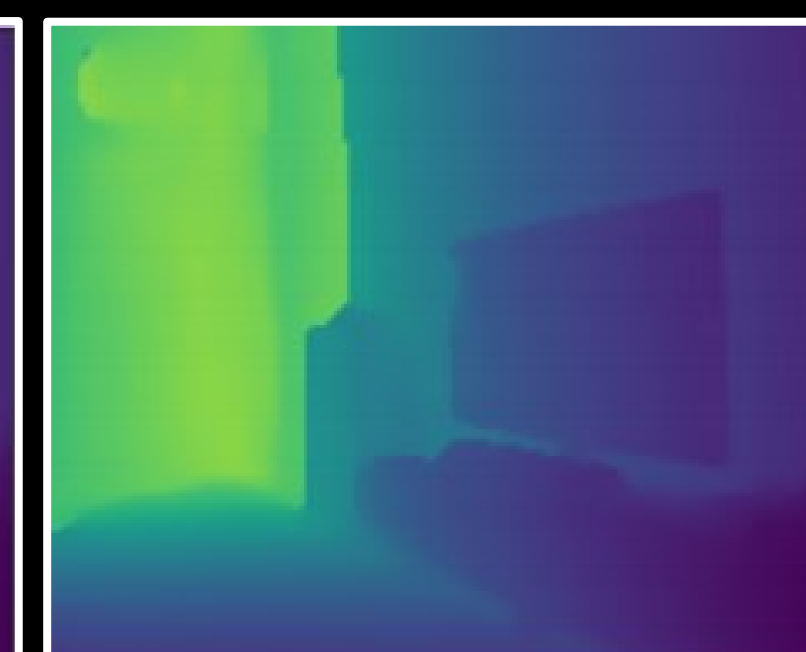
tissue imaging



deep learning [Barragan et al. 2021]



inverse rendering [Wu et al. 2021]



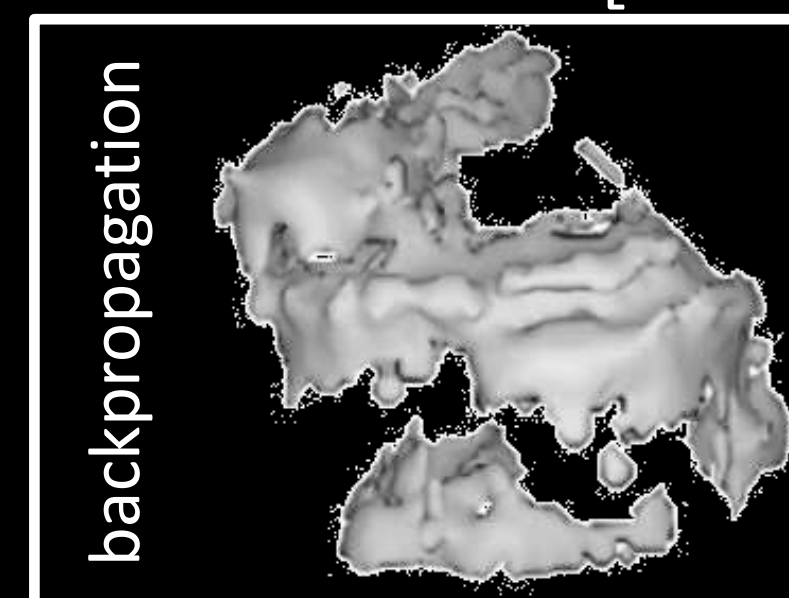
SONAR [Reed et al. 2023]



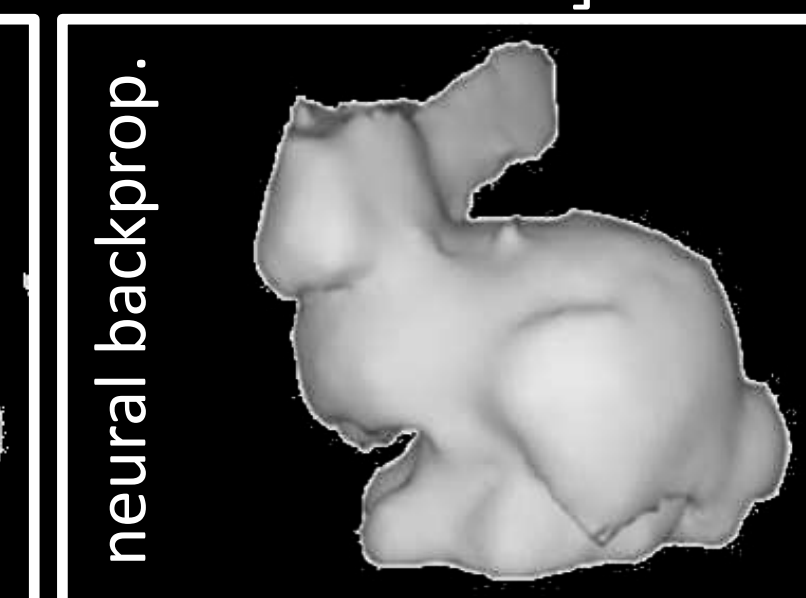
Ordinary



Derivative



backpropagation



neural backprop.

Physics-based rendering and its applications to computational imaging

forward rendering

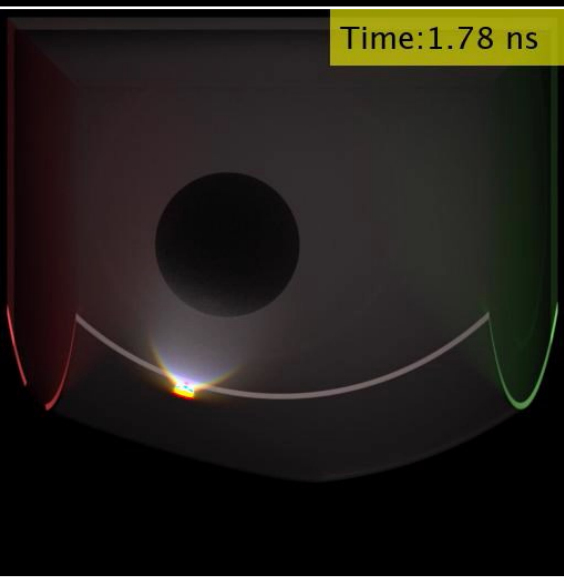


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

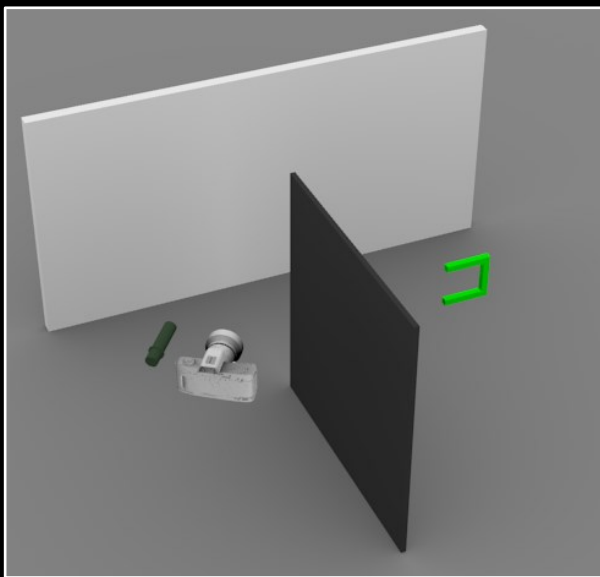
inverse rendering



- accurate and efficient differentiable simulation
- tractably solve general inverse problems



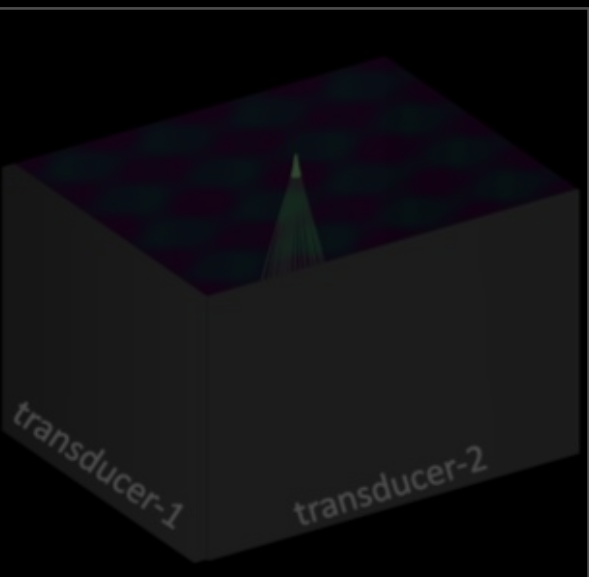
time-of-flight imaging



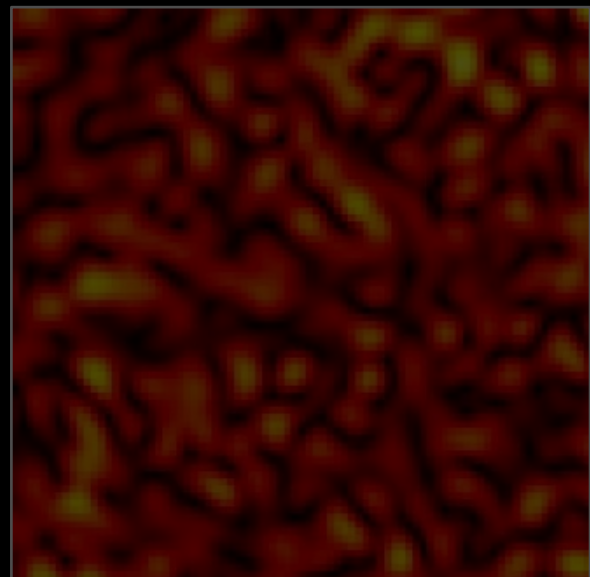
non-line-of-sight imaging



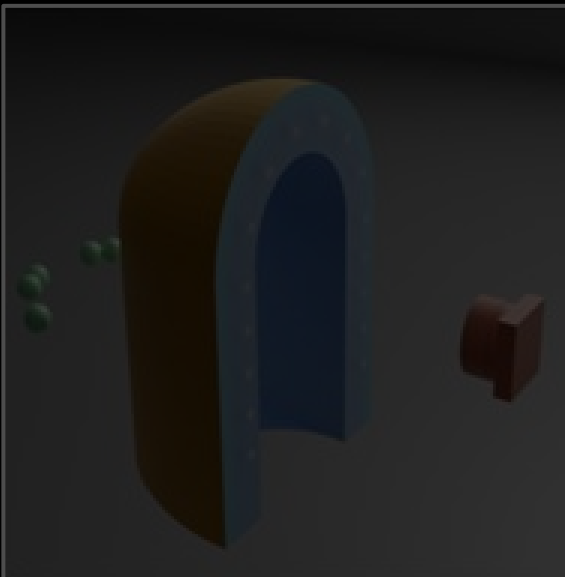
acousto-optic lensing



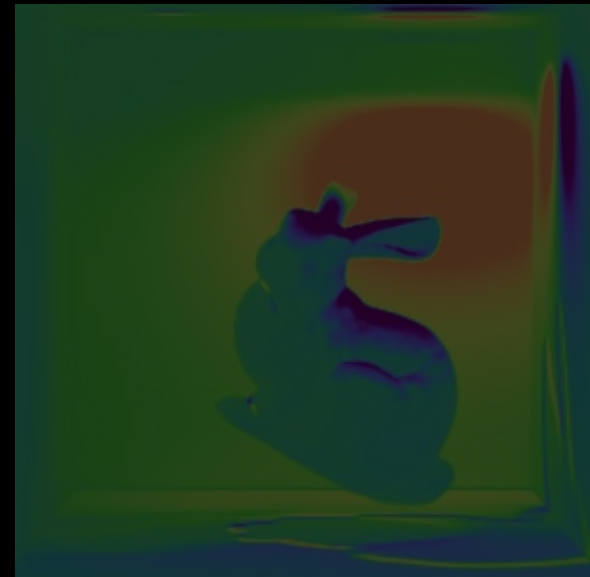
ultrafast light scanning



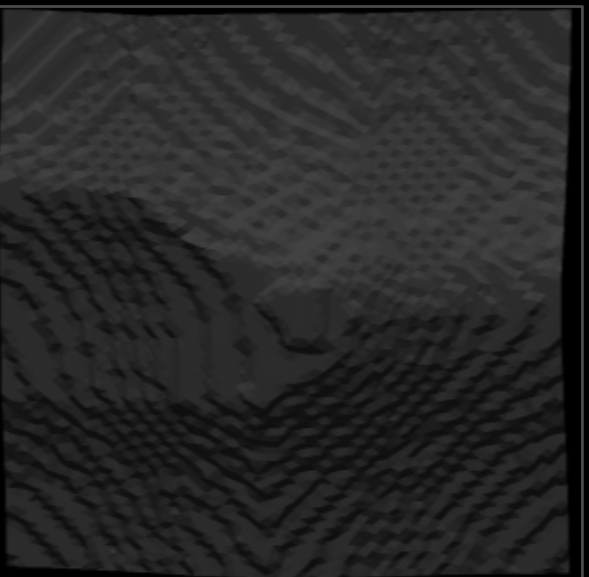
speckle imaging



tactile sensor design

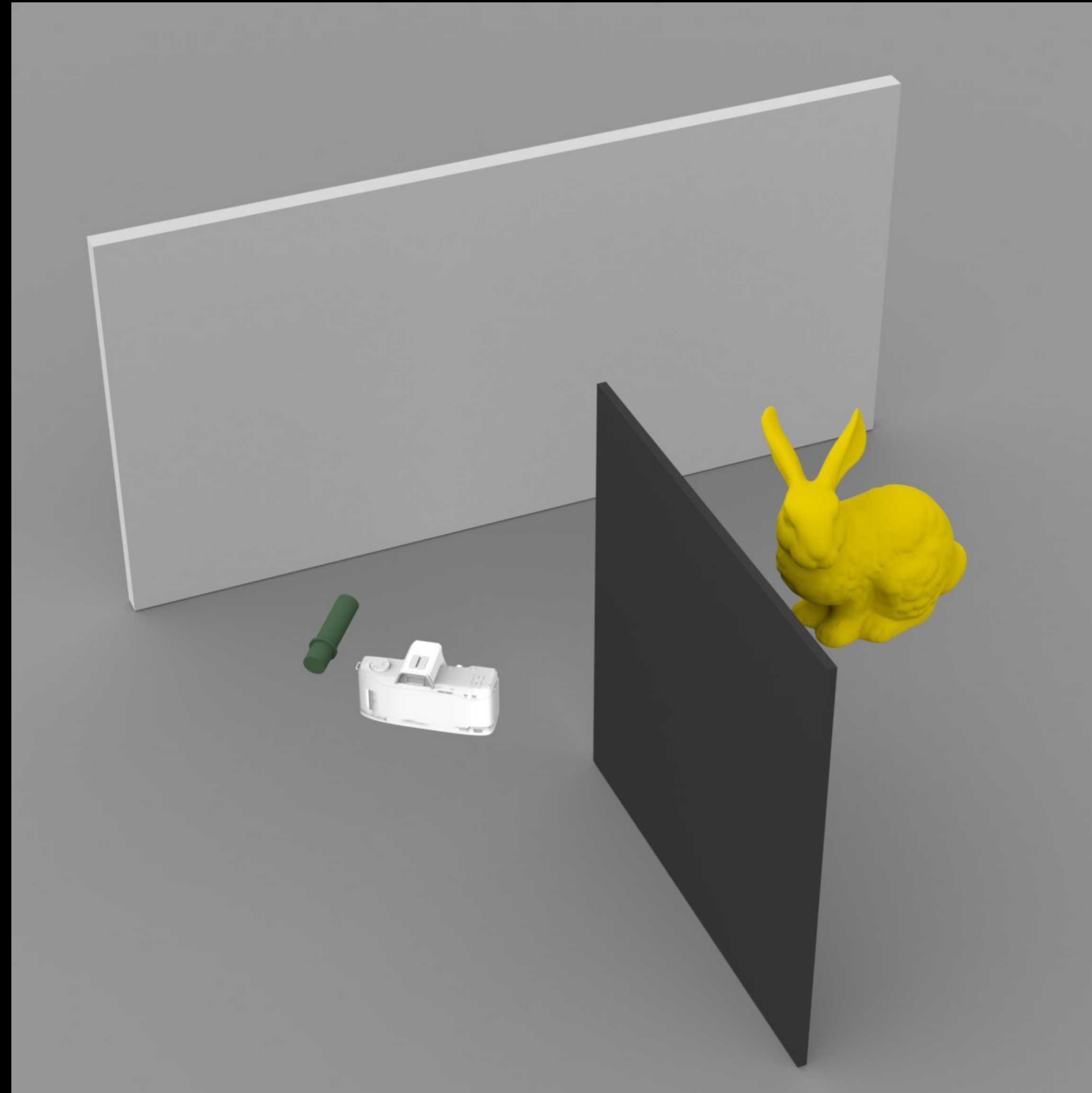


differentiable renderer

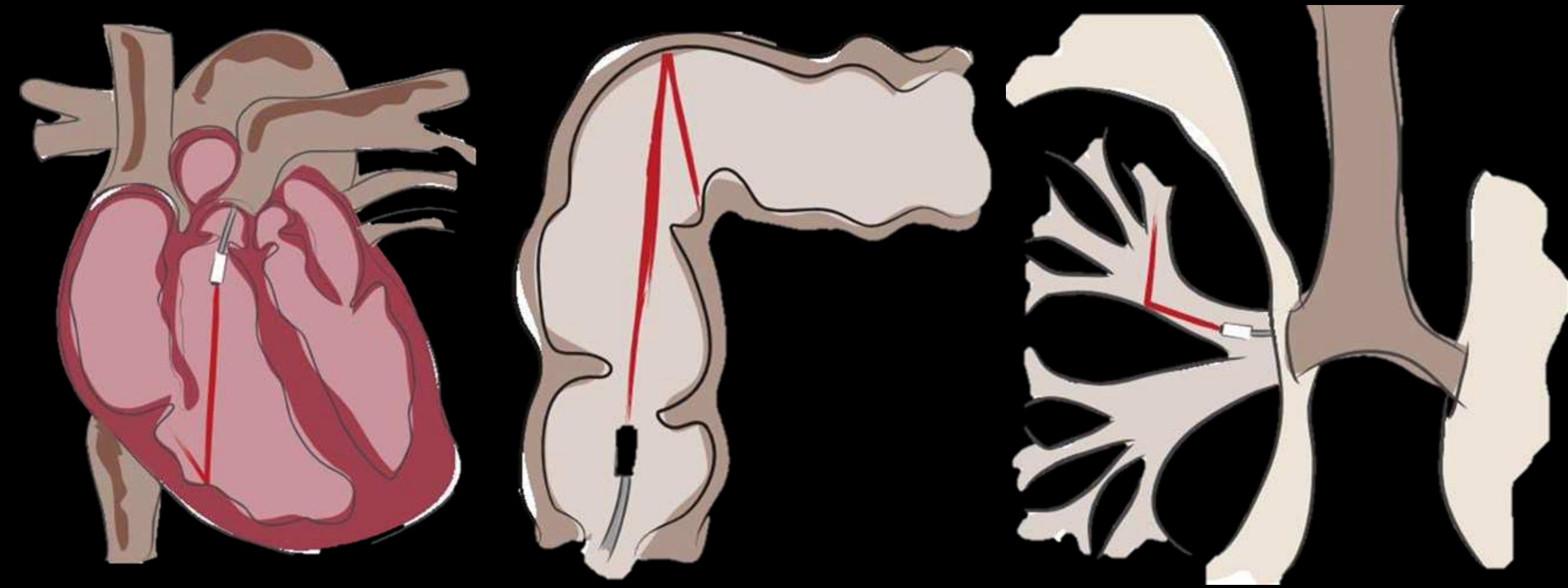


inverse problems

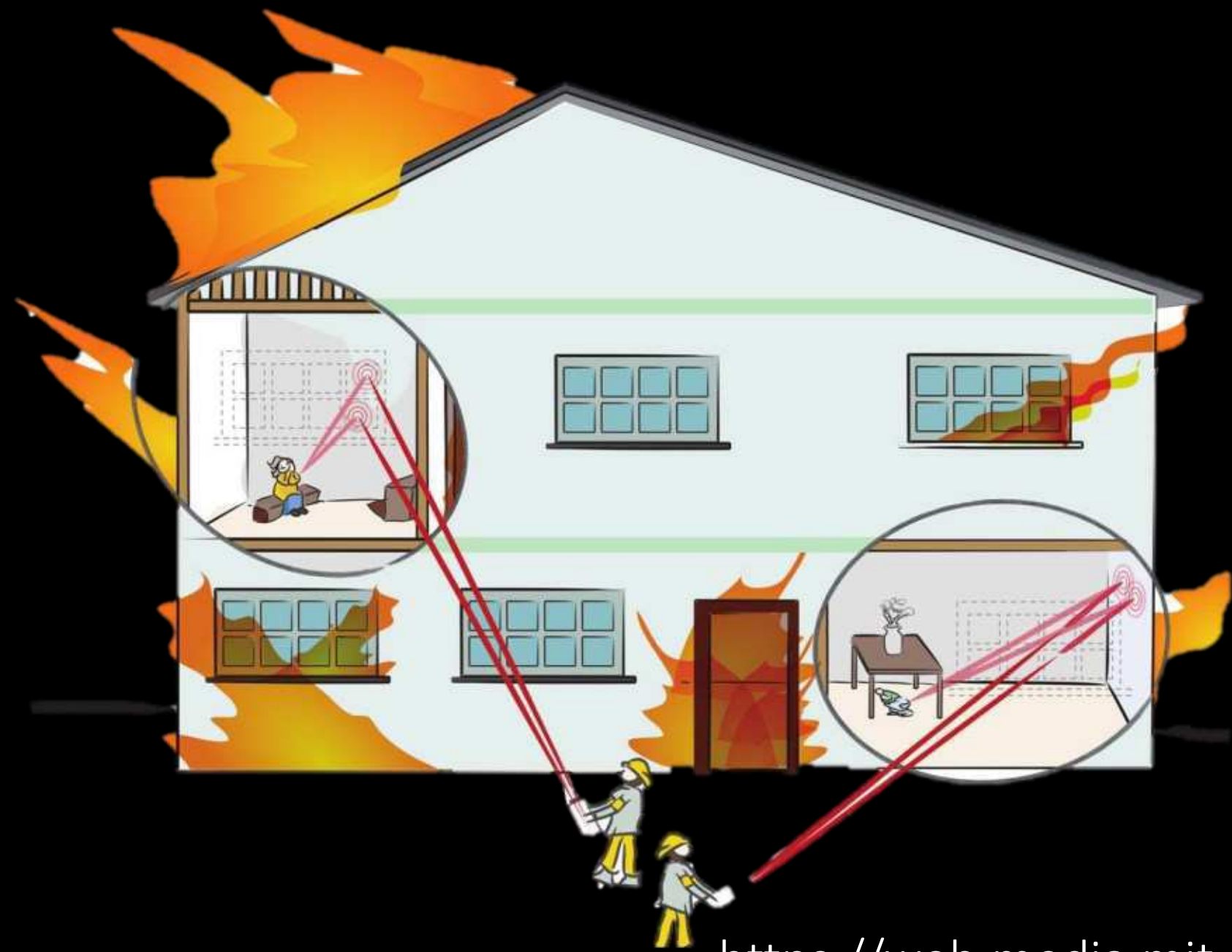
Non-line-of-sight imaging



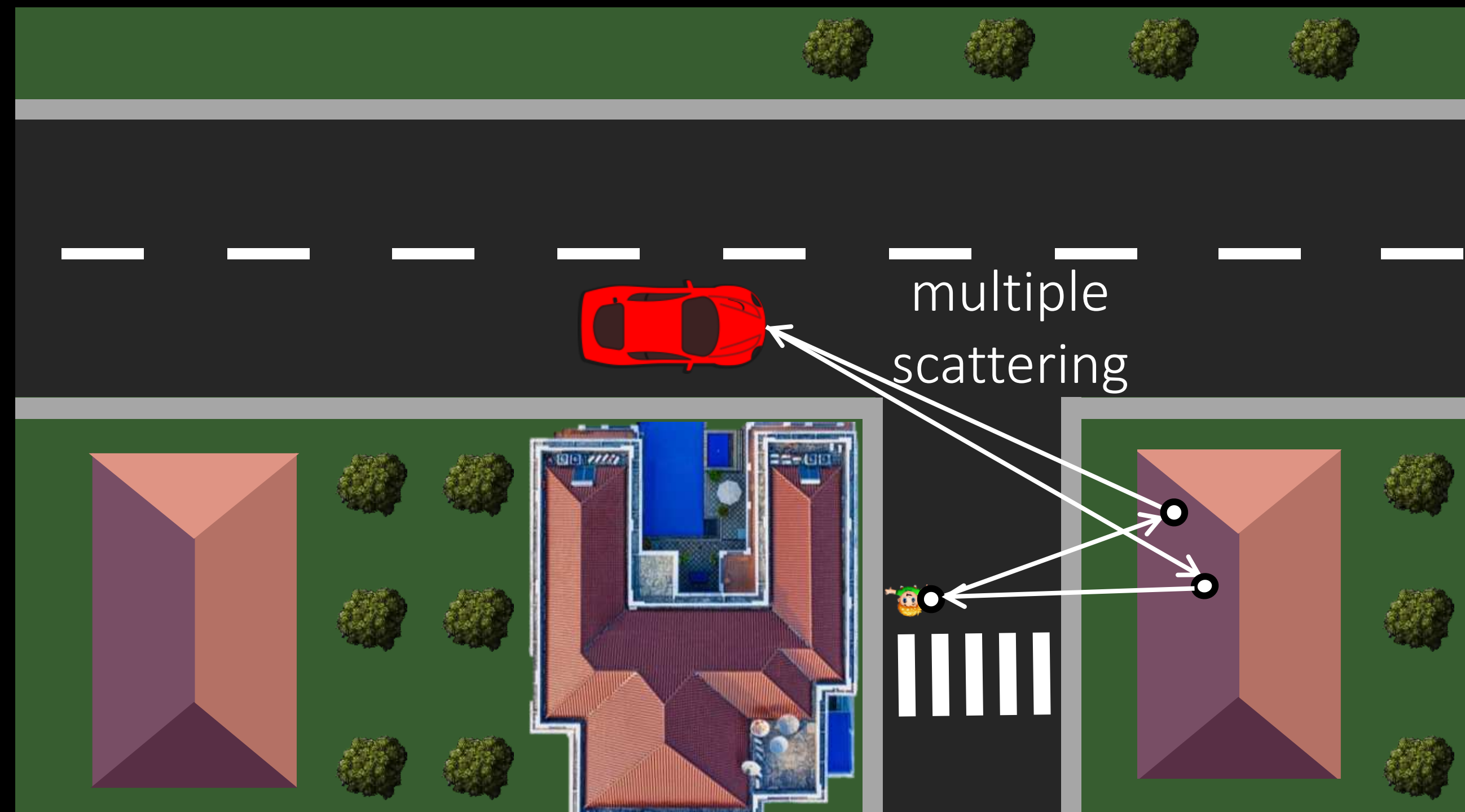
Applications



<https://web.media.mit.edu/~raskar/cornar/>

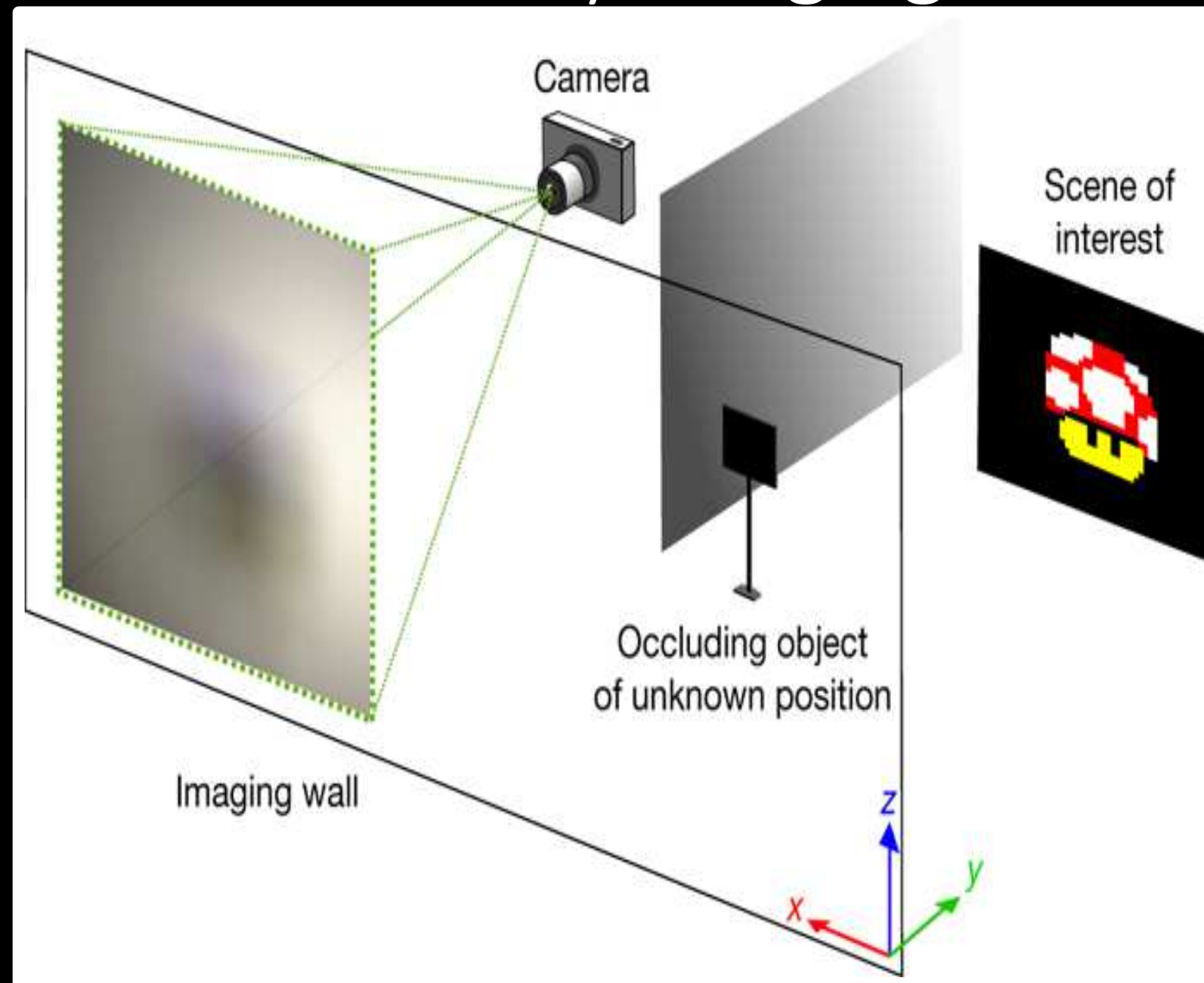


<https://web.media.mit.edu/~raskar/cornar/>



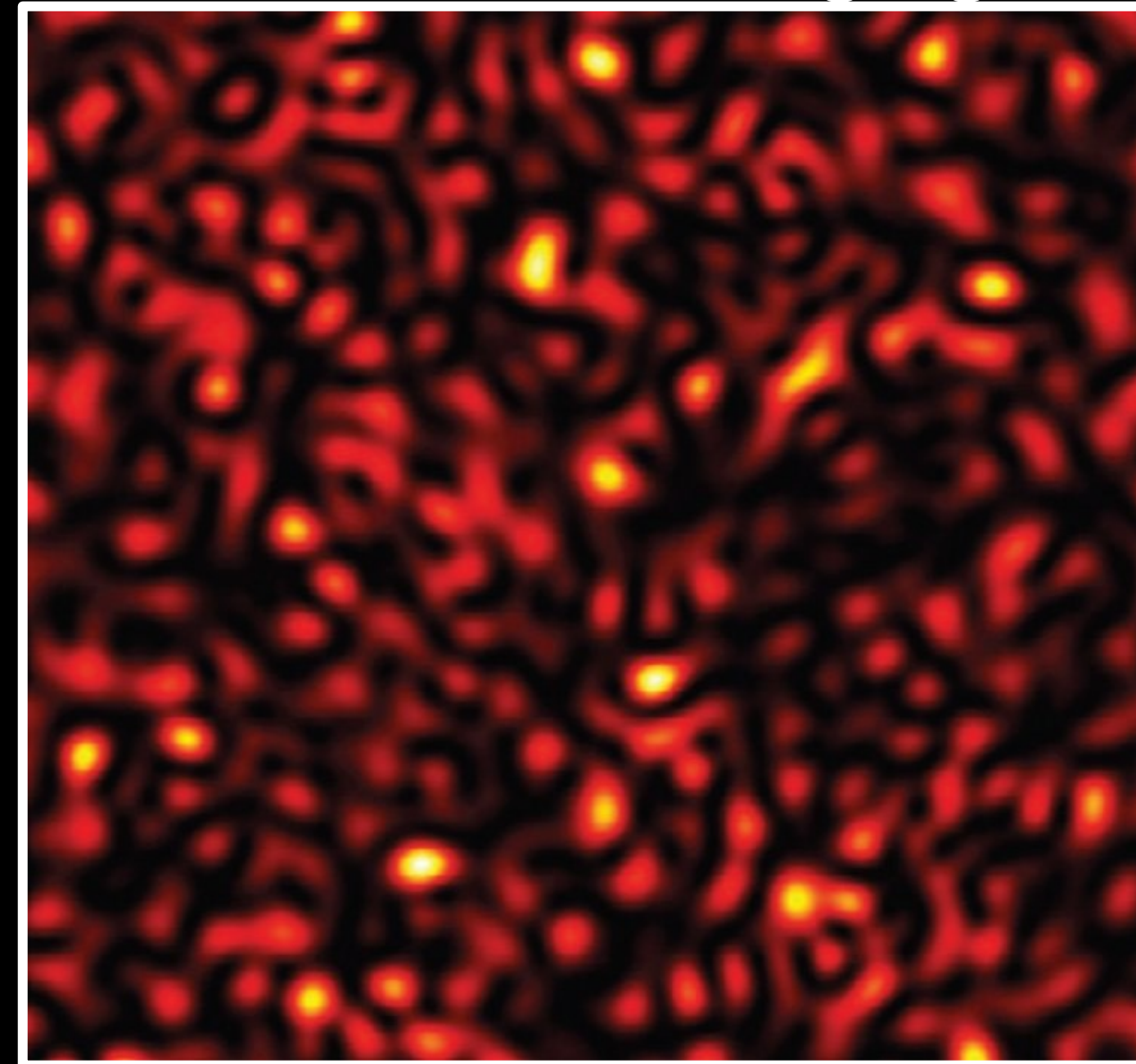
A lot of research on non-line-of-sight imaging

intensity imaging



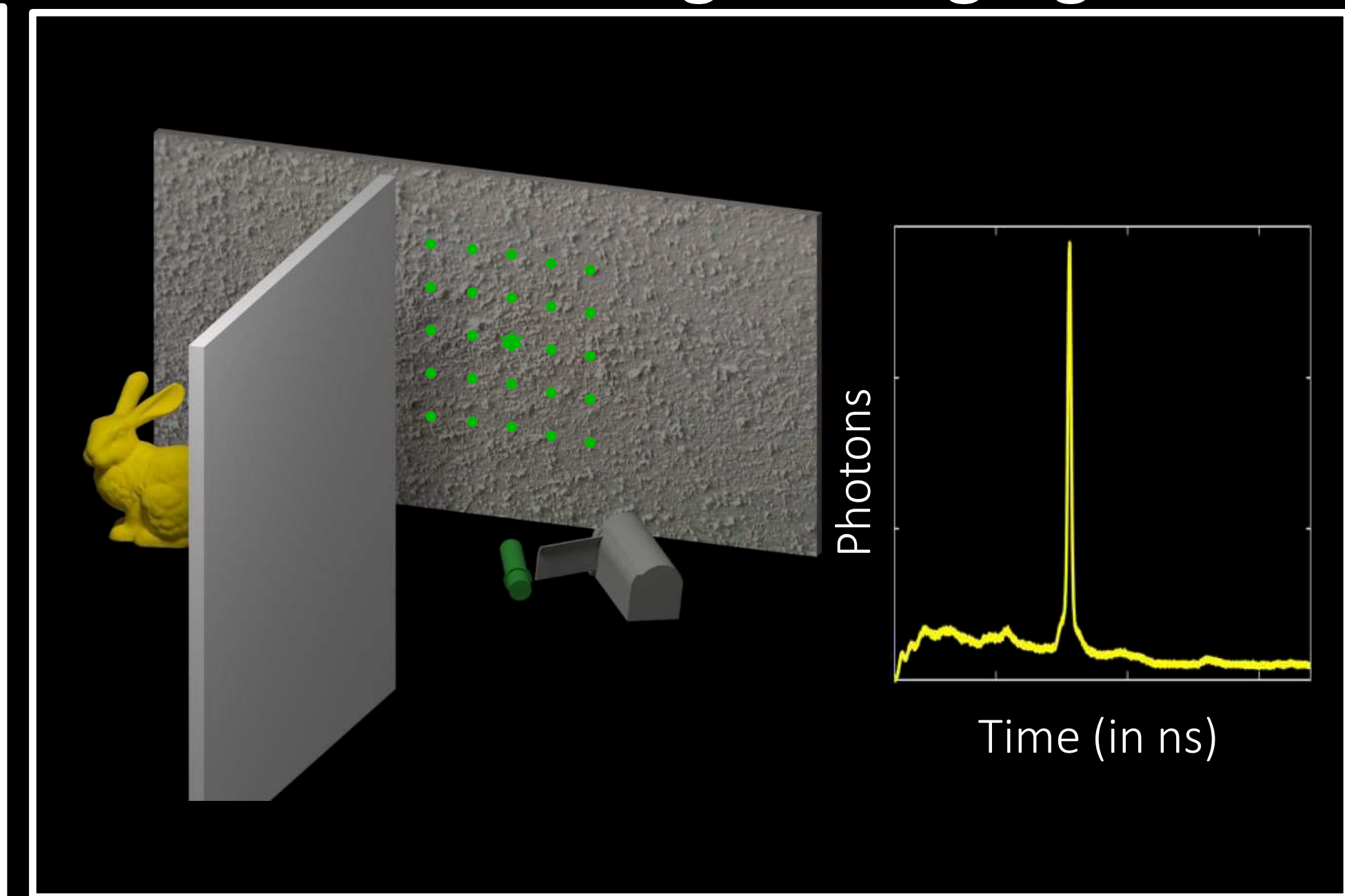
Bouman et al., ICCV 2017
Saunders et al., Nature 2019
Saunders et al., COSI 2019
Maeda et al., ICCP 2019
Lin et al., COSI 2020
Sharma, ICCV 2021

coherence imaging



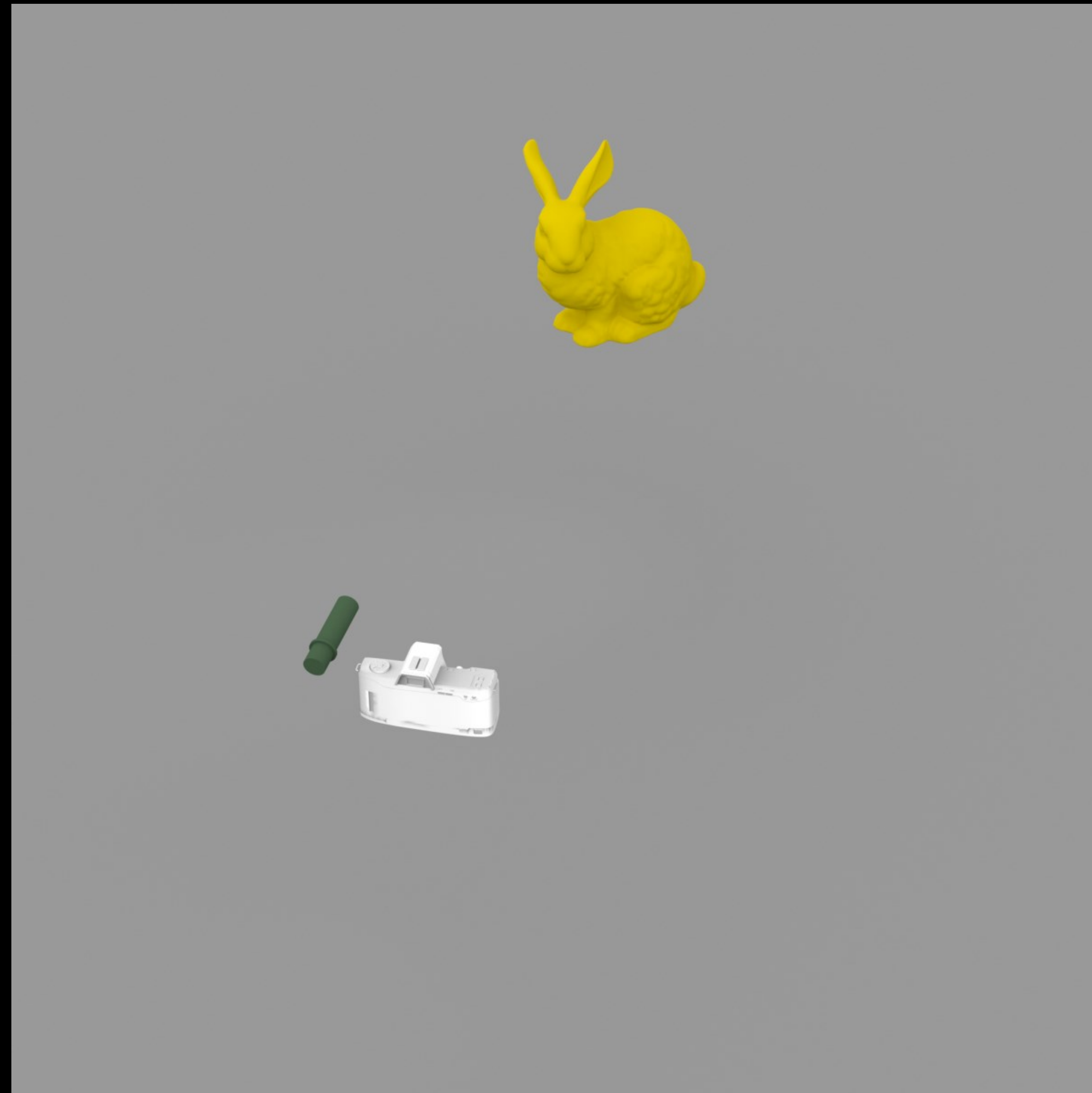
Katz et al., Nat. Photonics 2014
Lei et al., CVPR, 2019
Boger-Lombard, Nat. Com. 2019
Metzler et al., Optica 2020
Willomitzer et al., Nat. Com. 2021
Chen et al. SPIE 2022

time-of-flight imaging

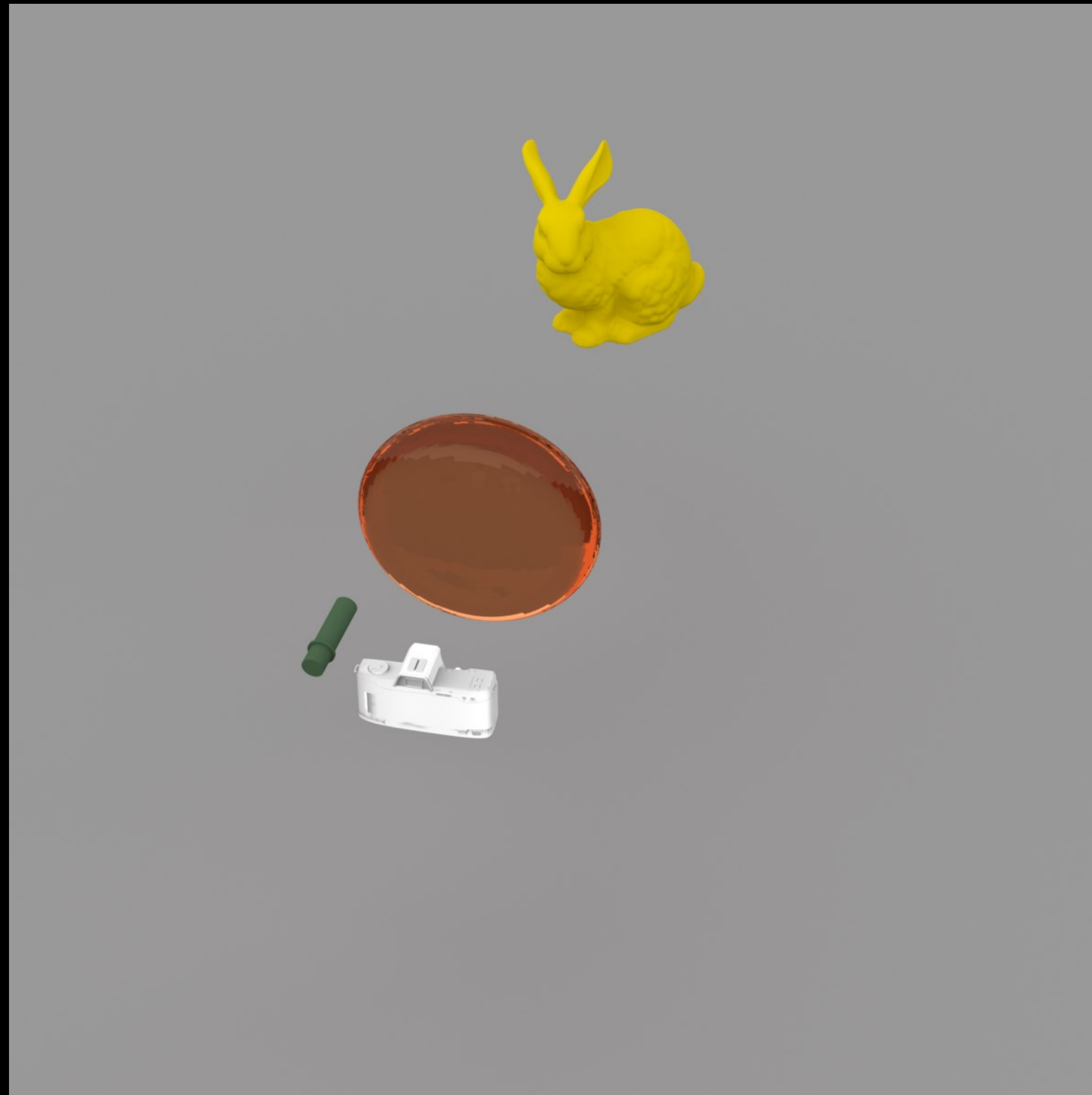


Velten et al., Nat. comm. 2012
Toole et al., Nature 2018
Liu et al., Nature 2019, Nat. comm. 2020
Rapp et al. Nat. comm. 2020
Xin et al., CVPR 2019
Nam et al., Nat. comm. 2021

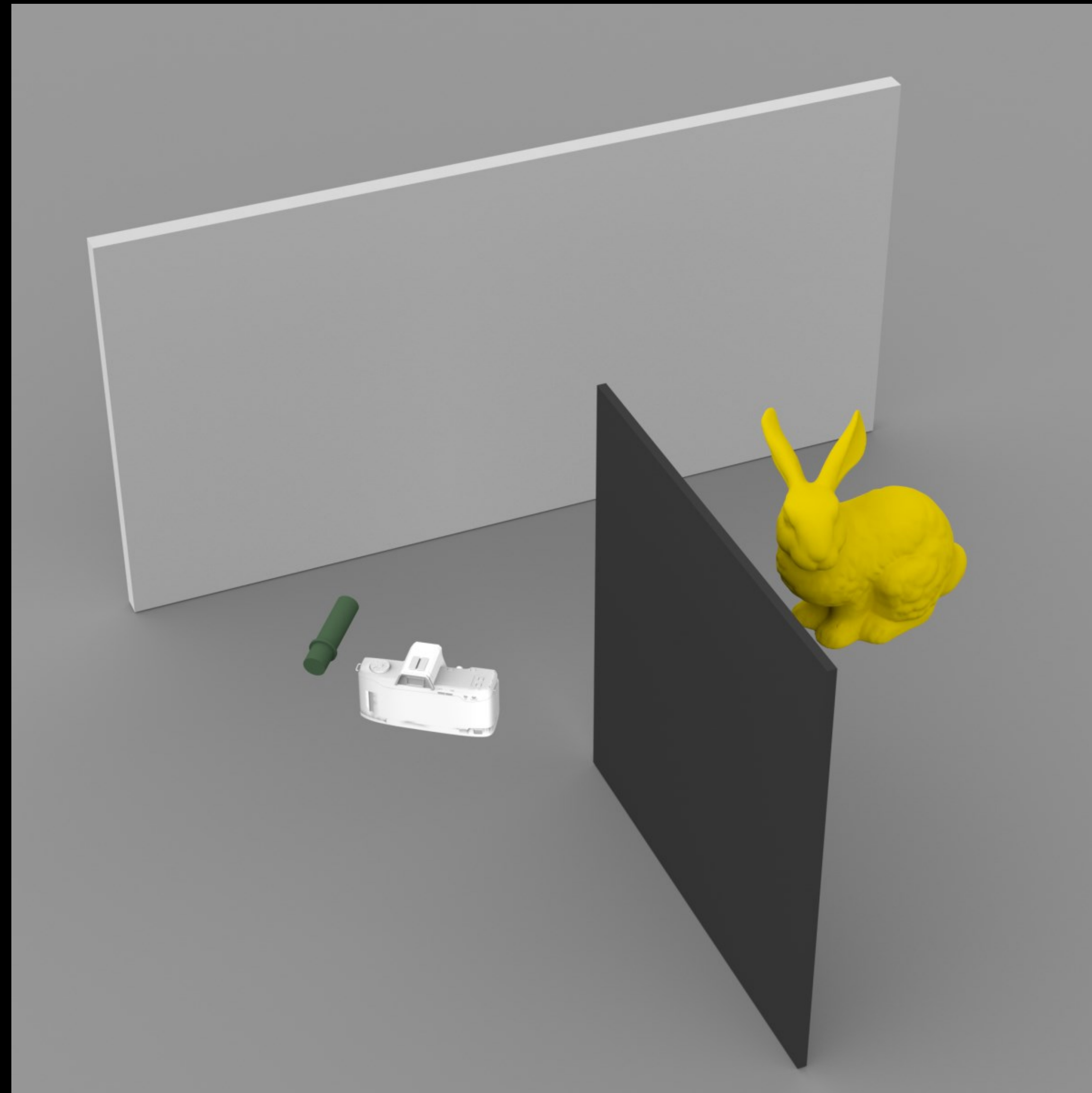
How do we image in line-of-sight?



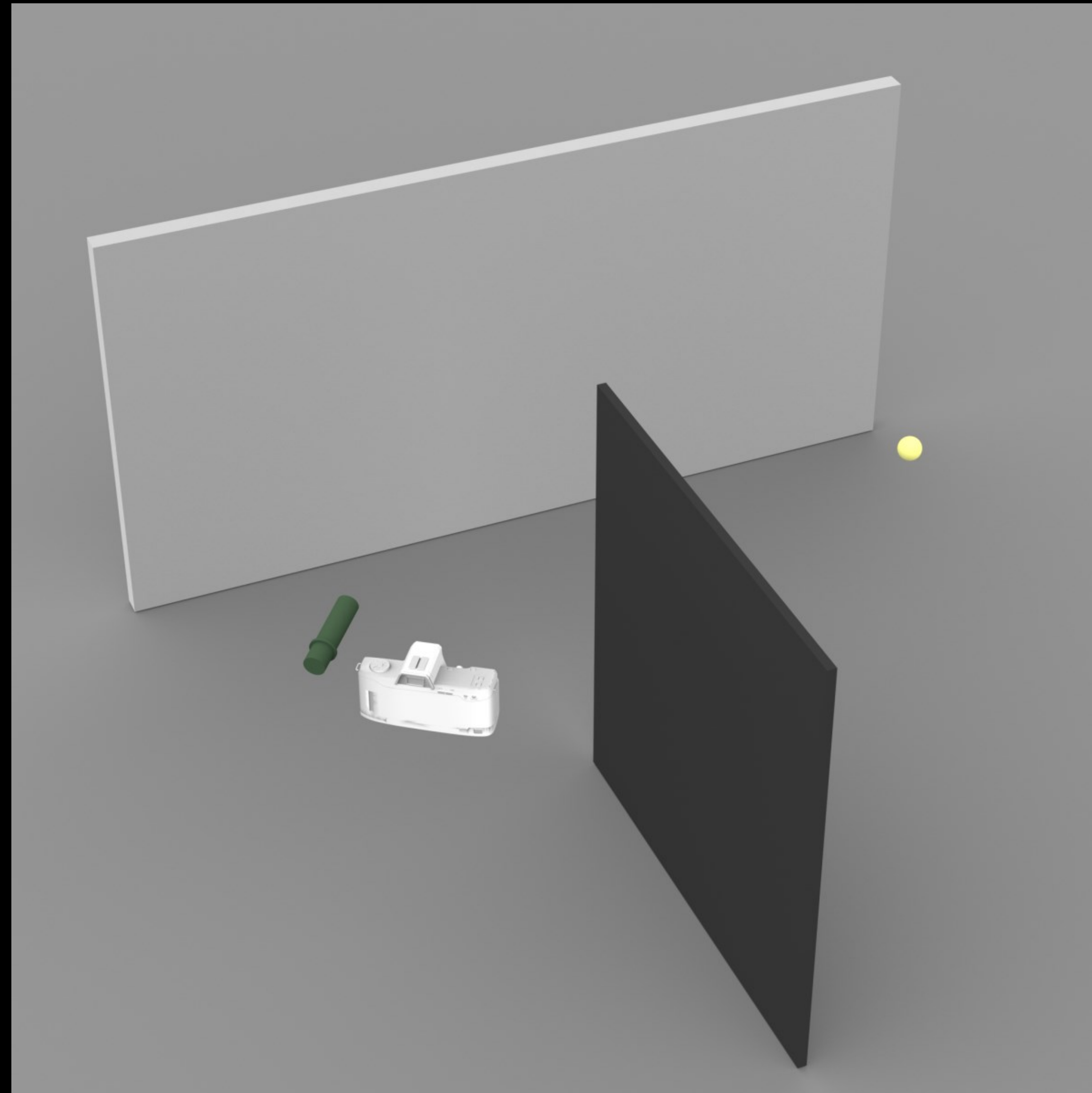
How do we image in line-of-sight?



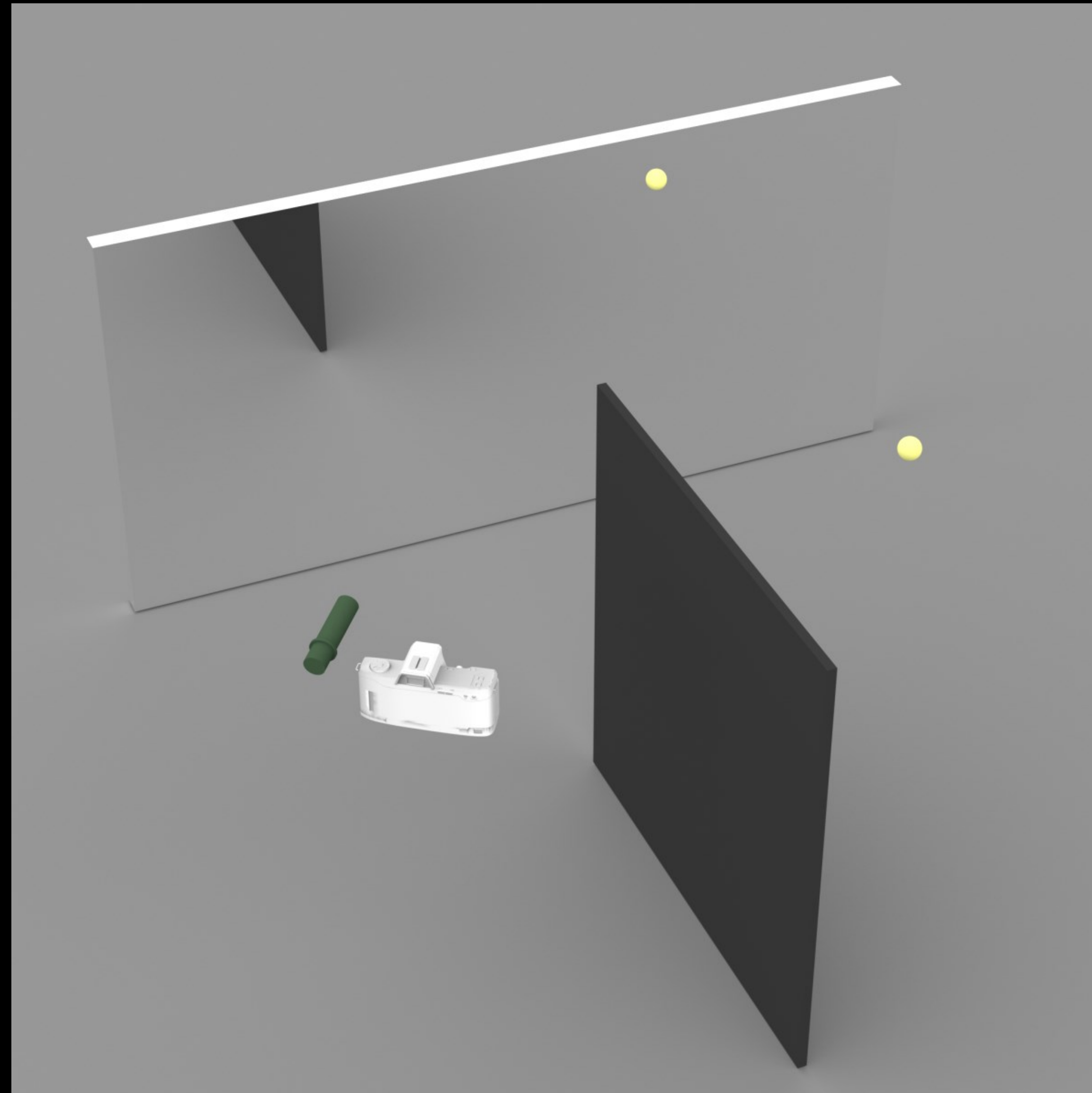
How do we image in non-line-of-sight?



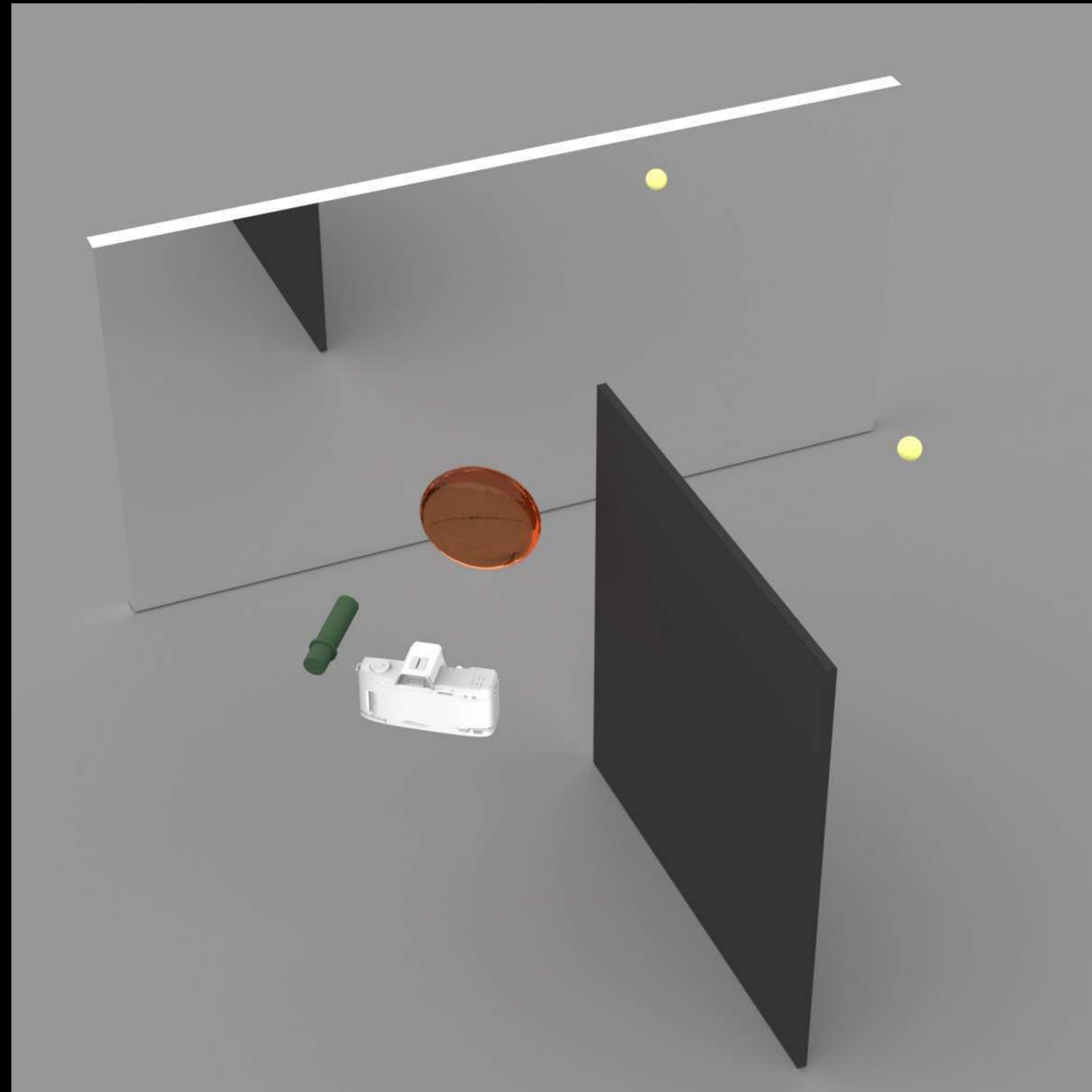
How do we focus on a voxel?



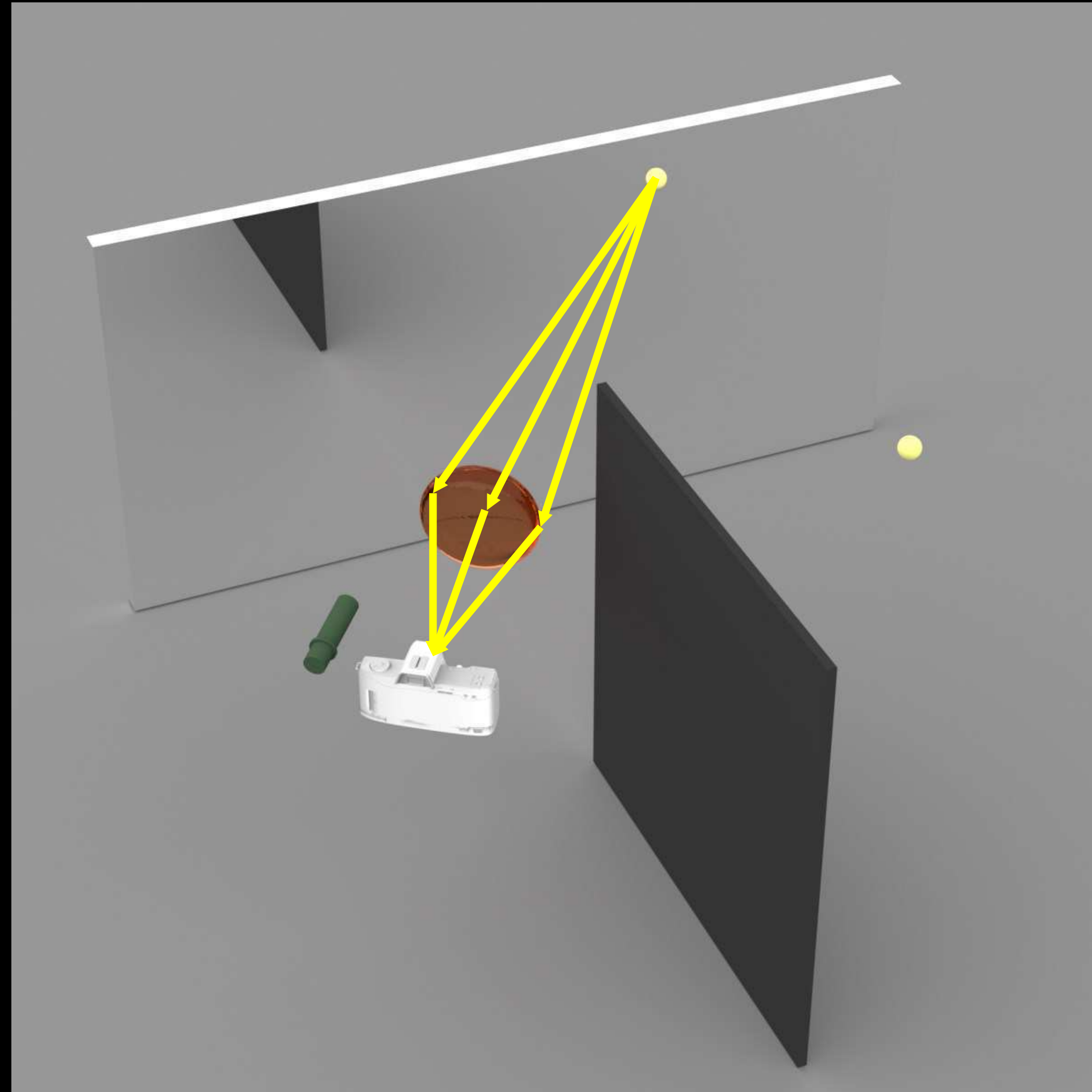
How do we focus on a voxel?



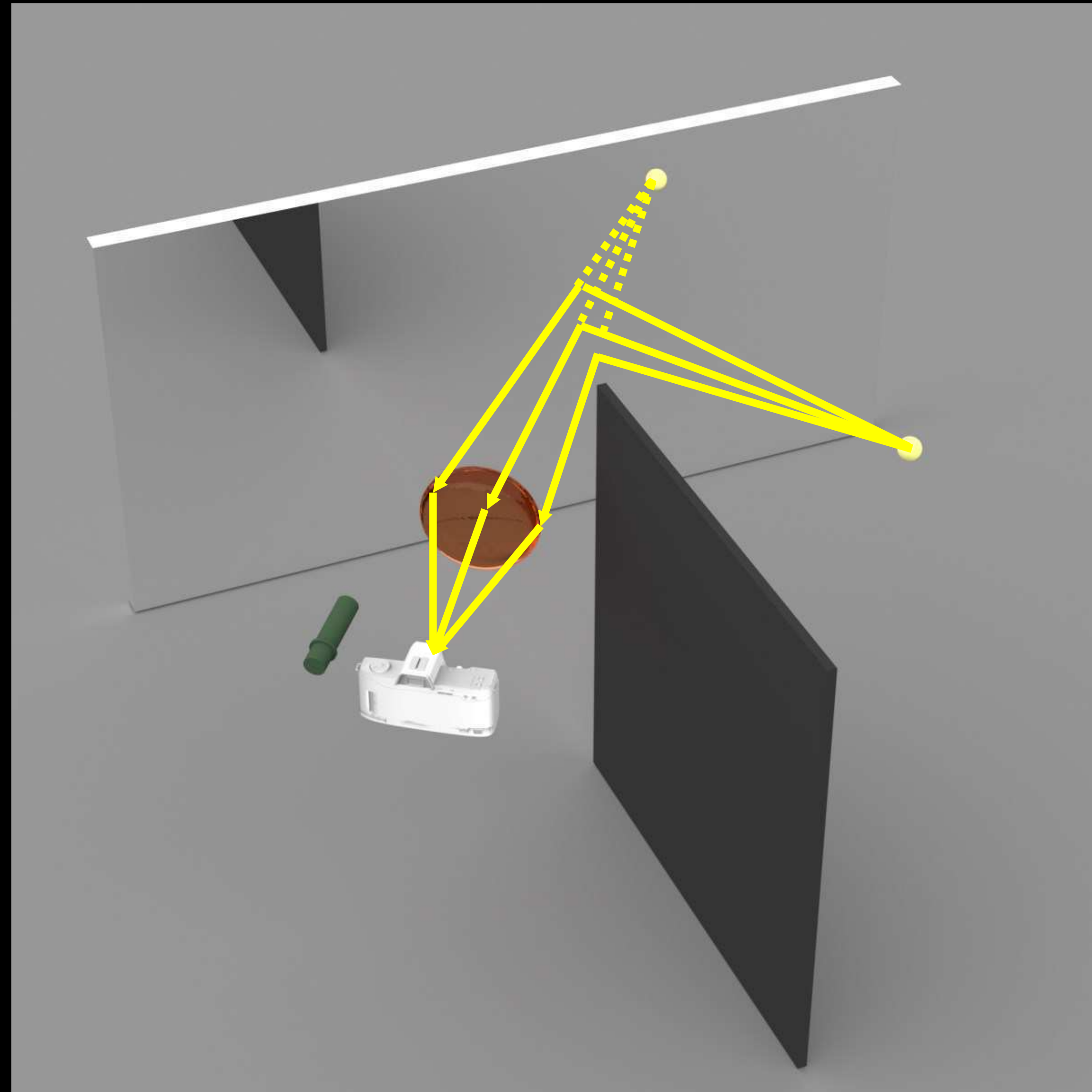
How do we focus on a voxel?



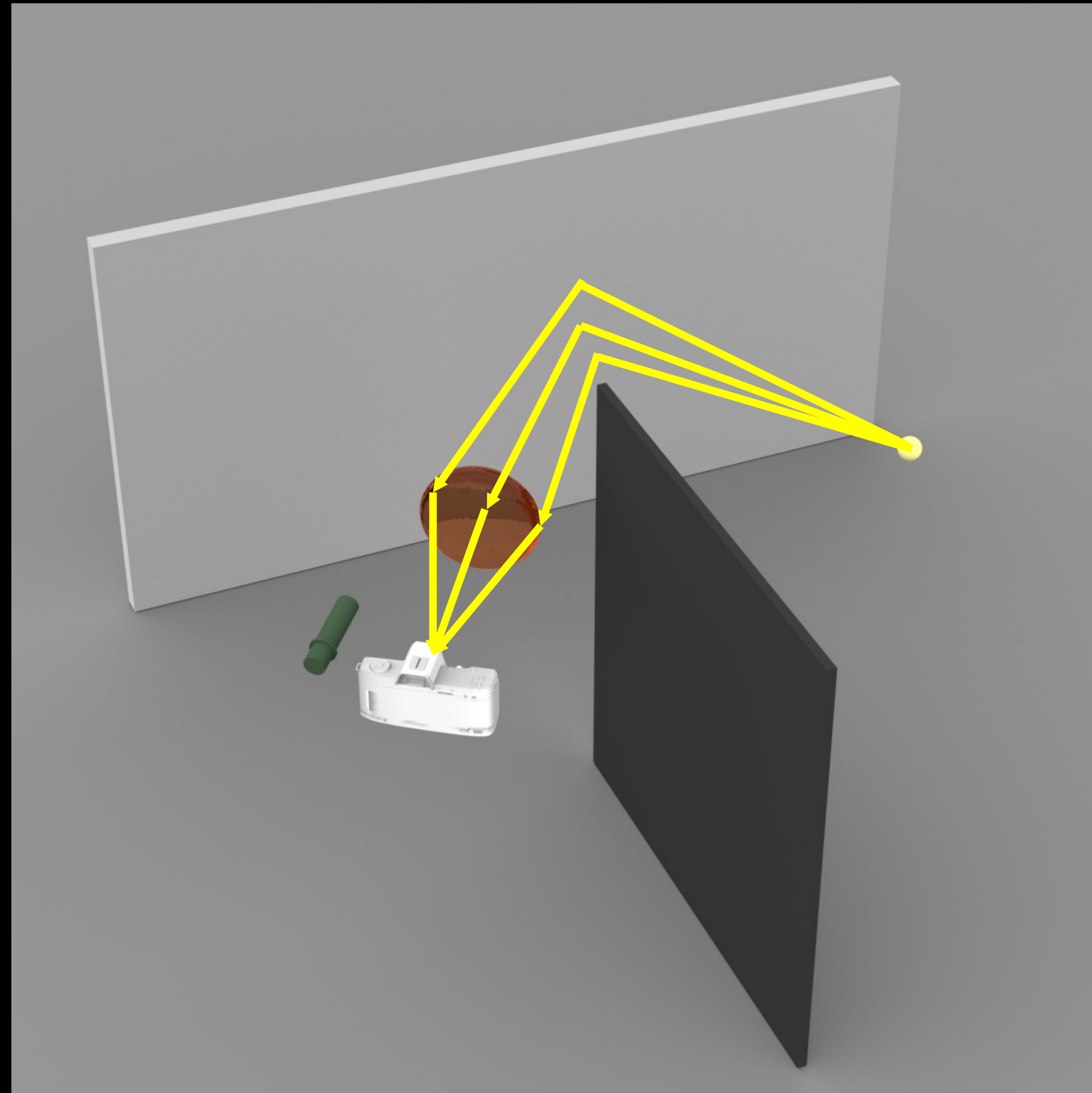
How do we focus on a voxel?



How do we focus on a voxel?

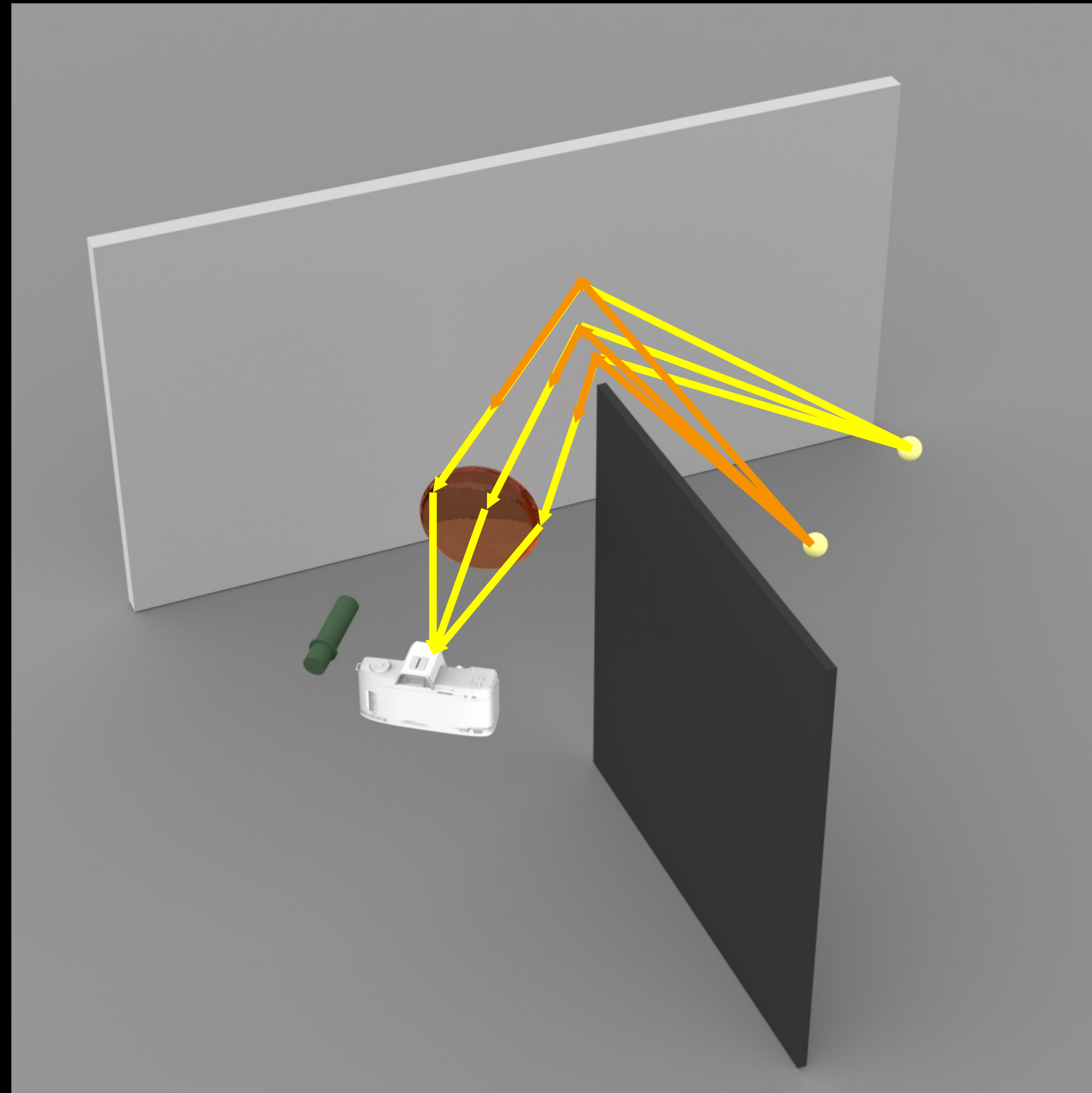


How do we focus on a voxel?



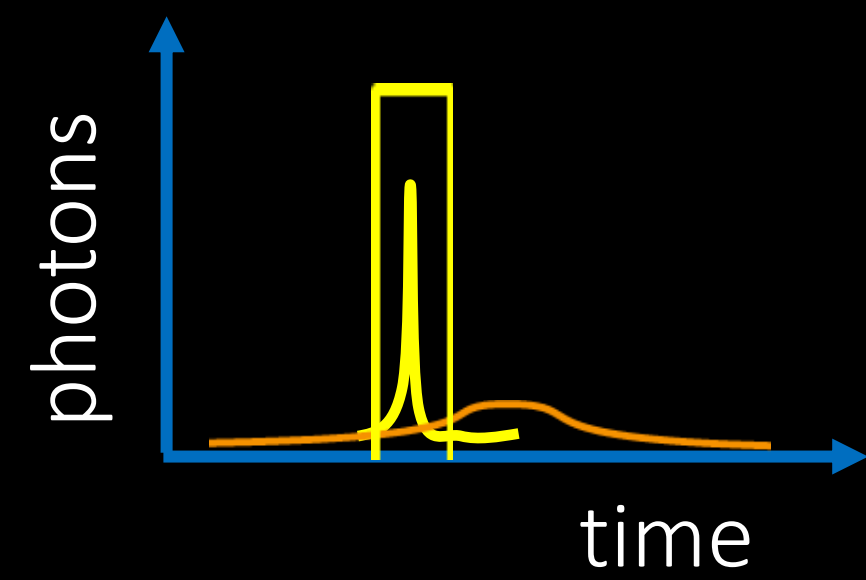
How do we focus on a voxel?

challenge:
out of focus voxels

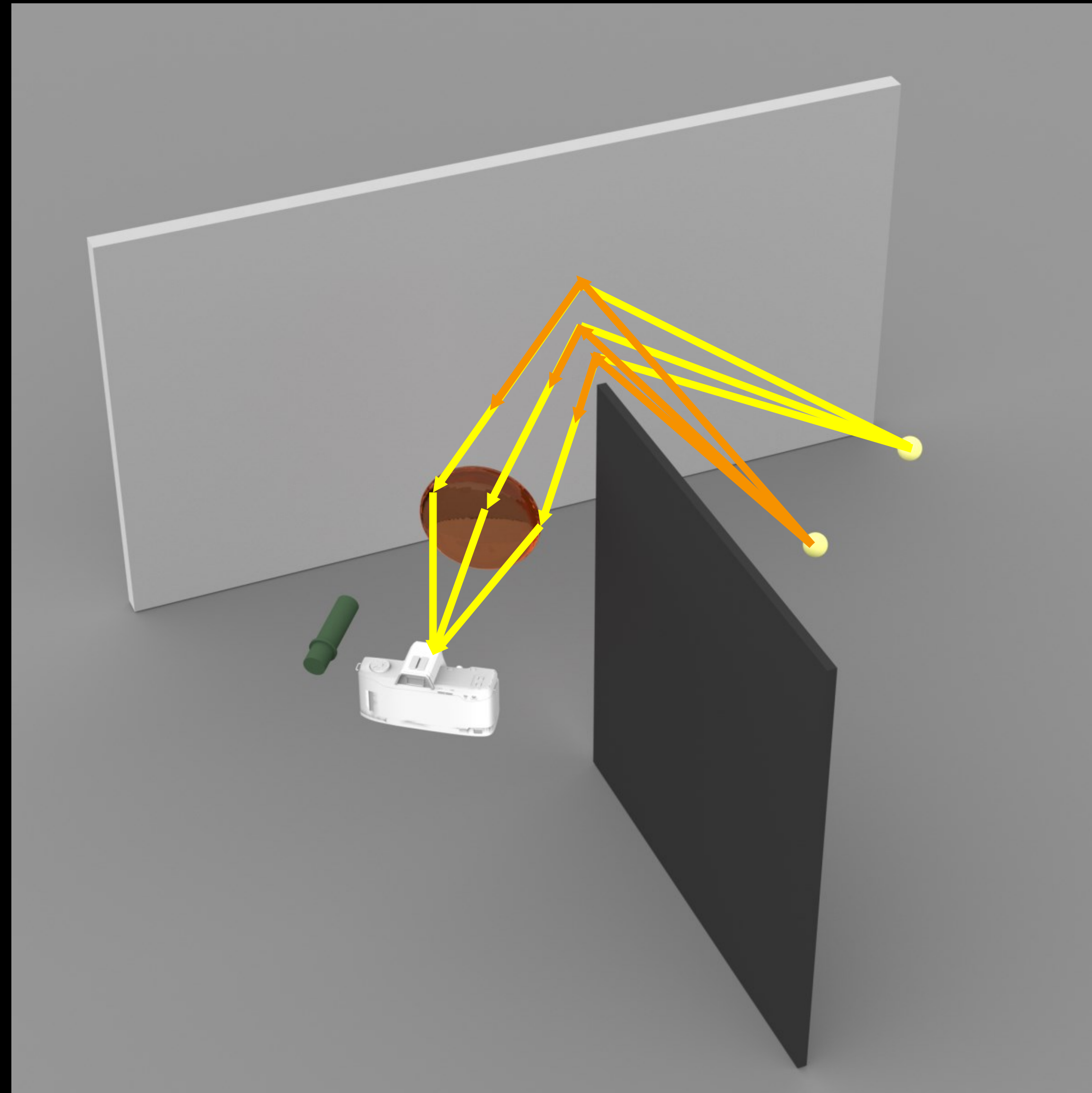


How do we focus on a voxel?

challenge:
out of focus voxels

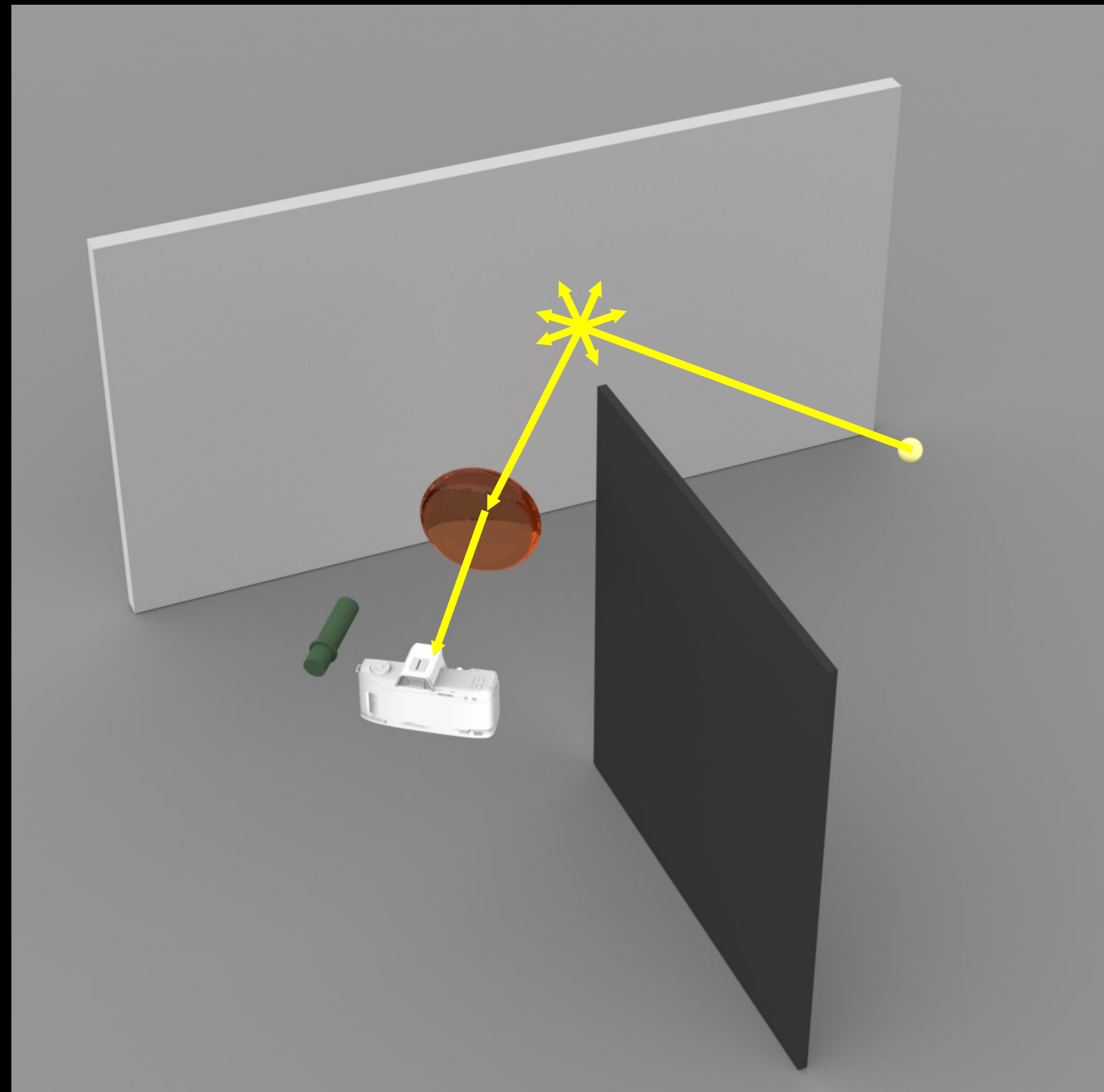


solution:
time-gate photons



How do we focus on a voxel?

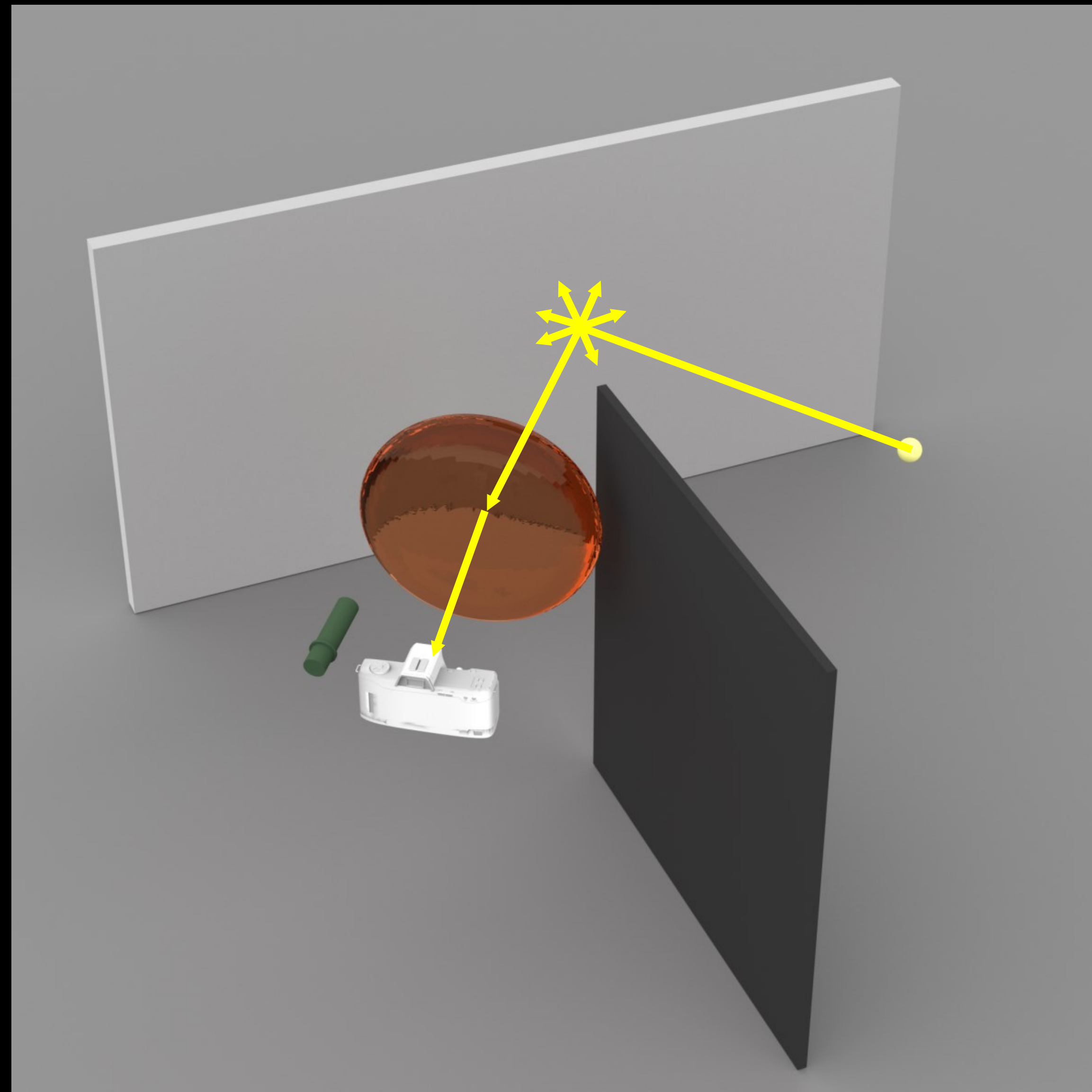
challenge:
non-specular photons



How do we focus on a voxel?

challenge:
non-specular photons

solution:
use a large lens?
expensive !!



How do we focus on a voxel?

challenge:

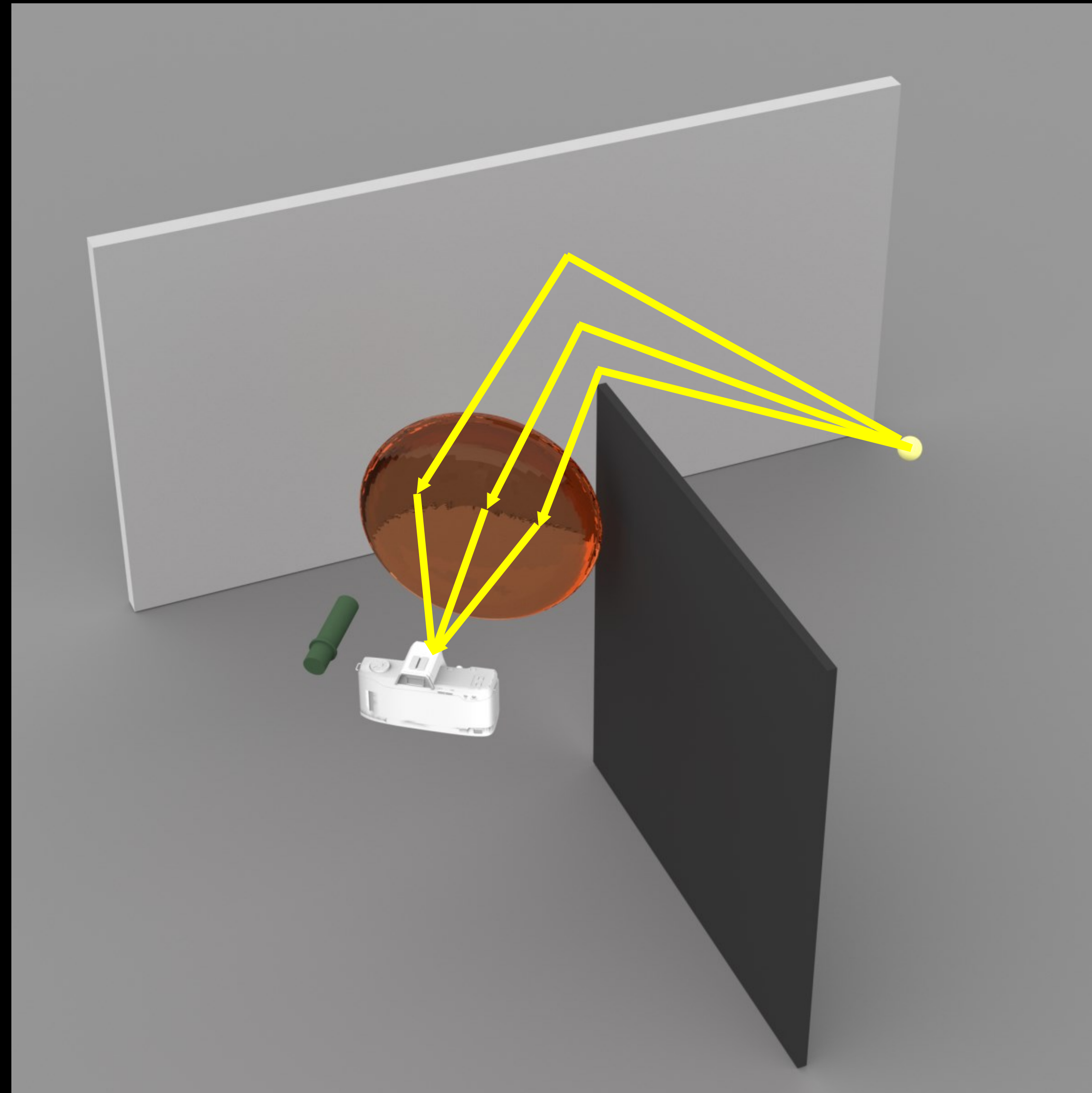
non-specular photons

solution:

use a large lens?

expensive !!

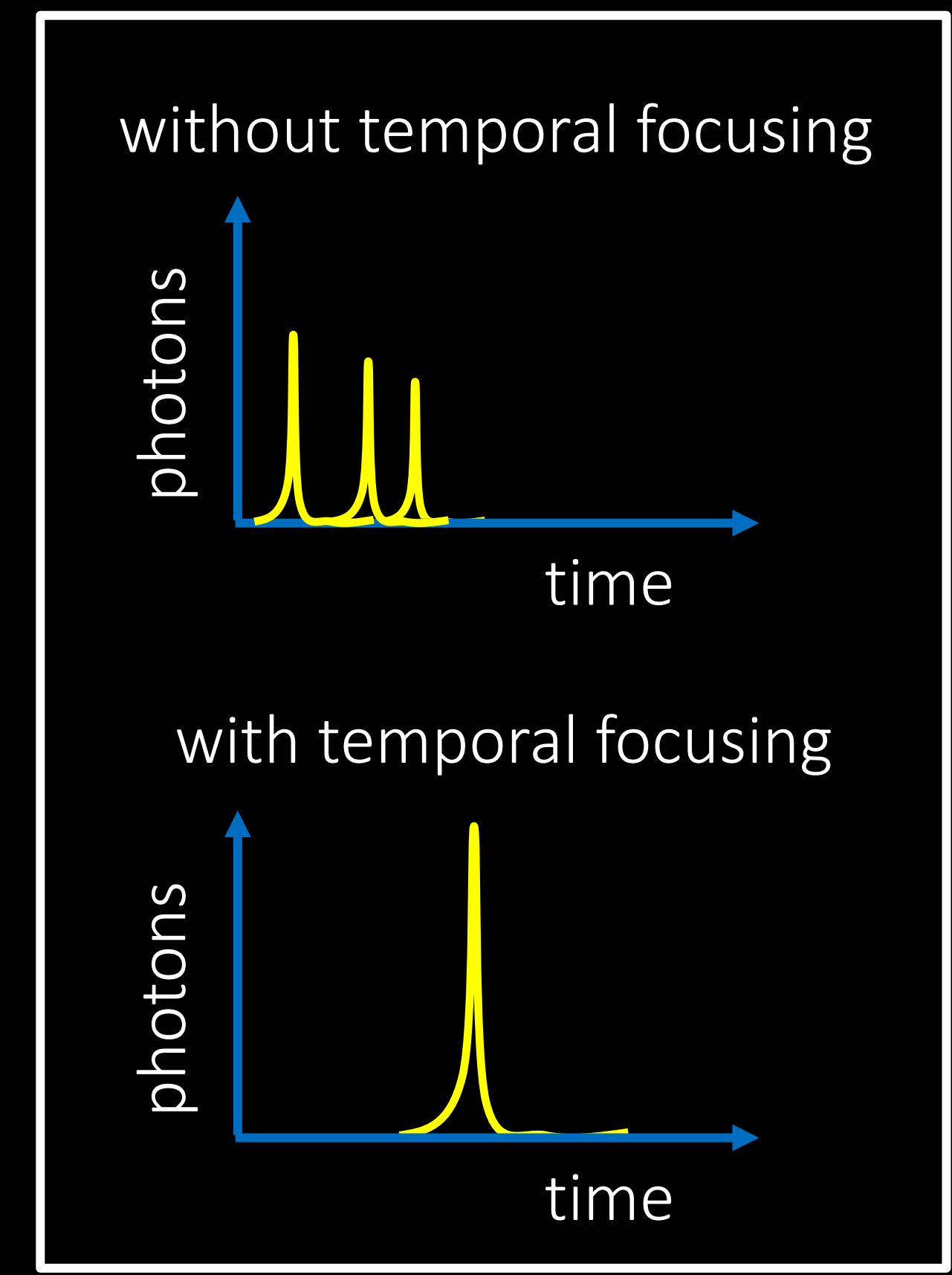
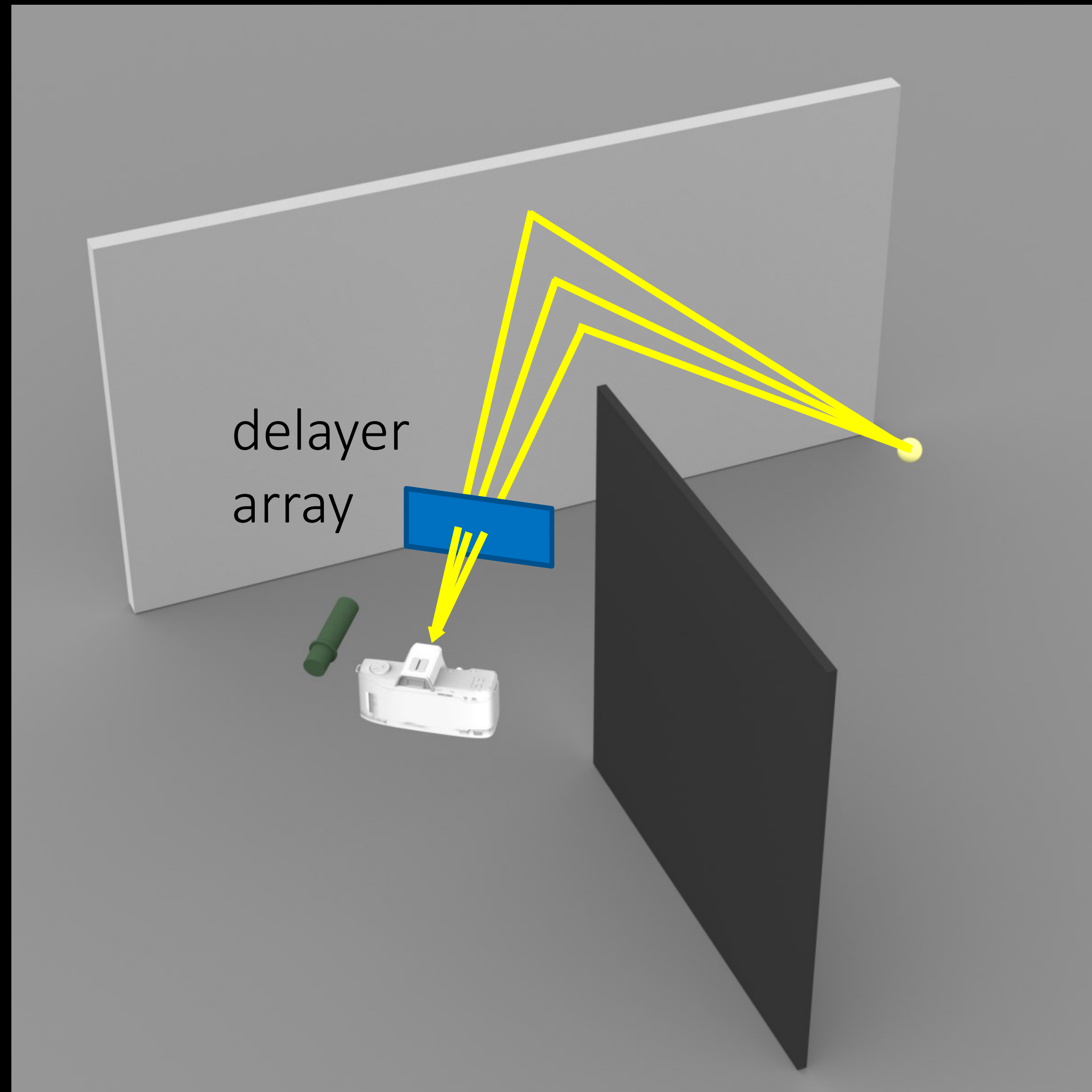
what does a lens do?
delays rays such that
they reach detector at
same time instant



Temporal focusing: imitate the lens

challenge:
non-specular photons

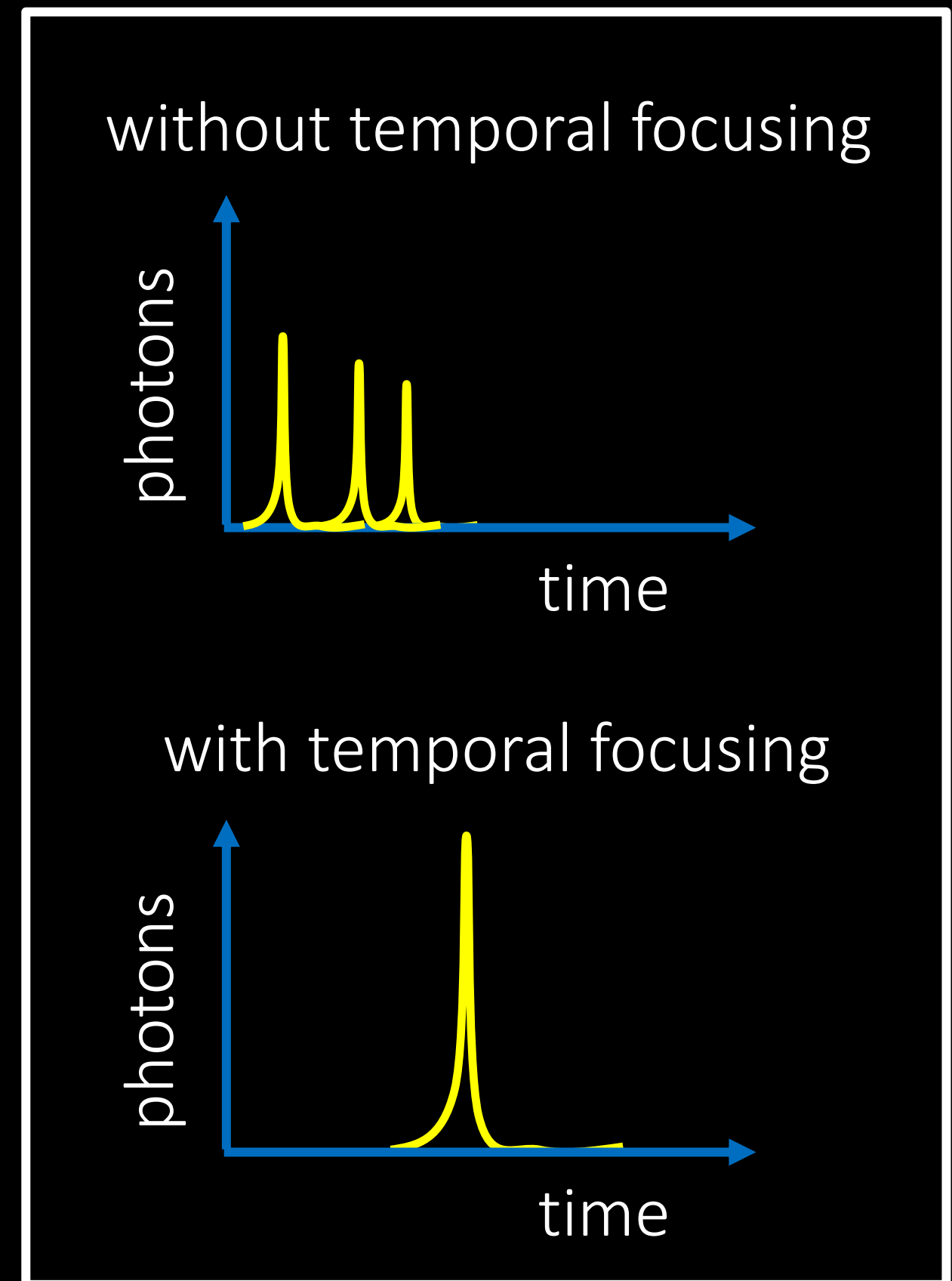
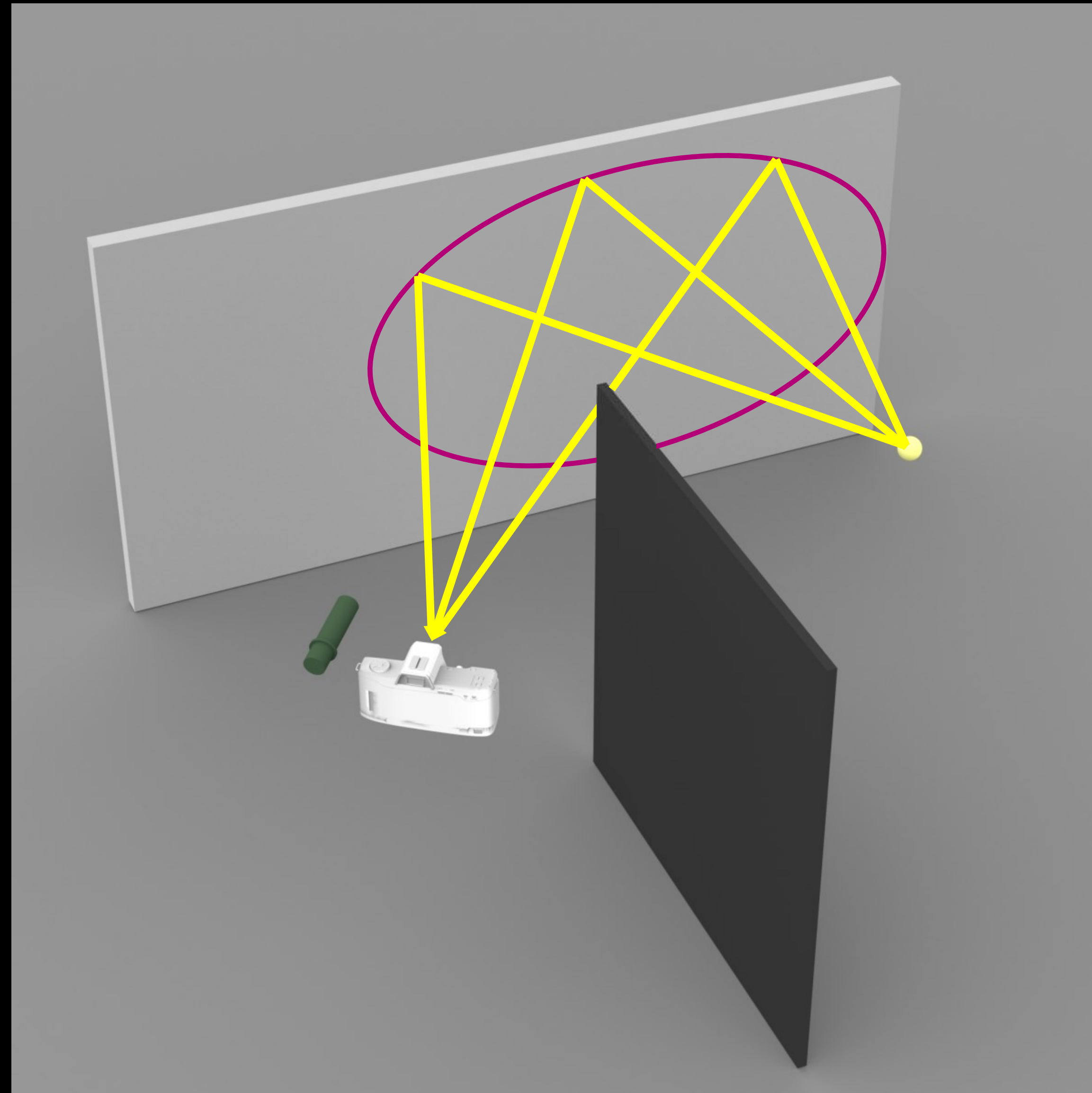
solution:
use a large lens?
expensive !!
temporal focusing:
imitate large lens



Temporal focusing: imitate the lens

challenge:
non-specular photons

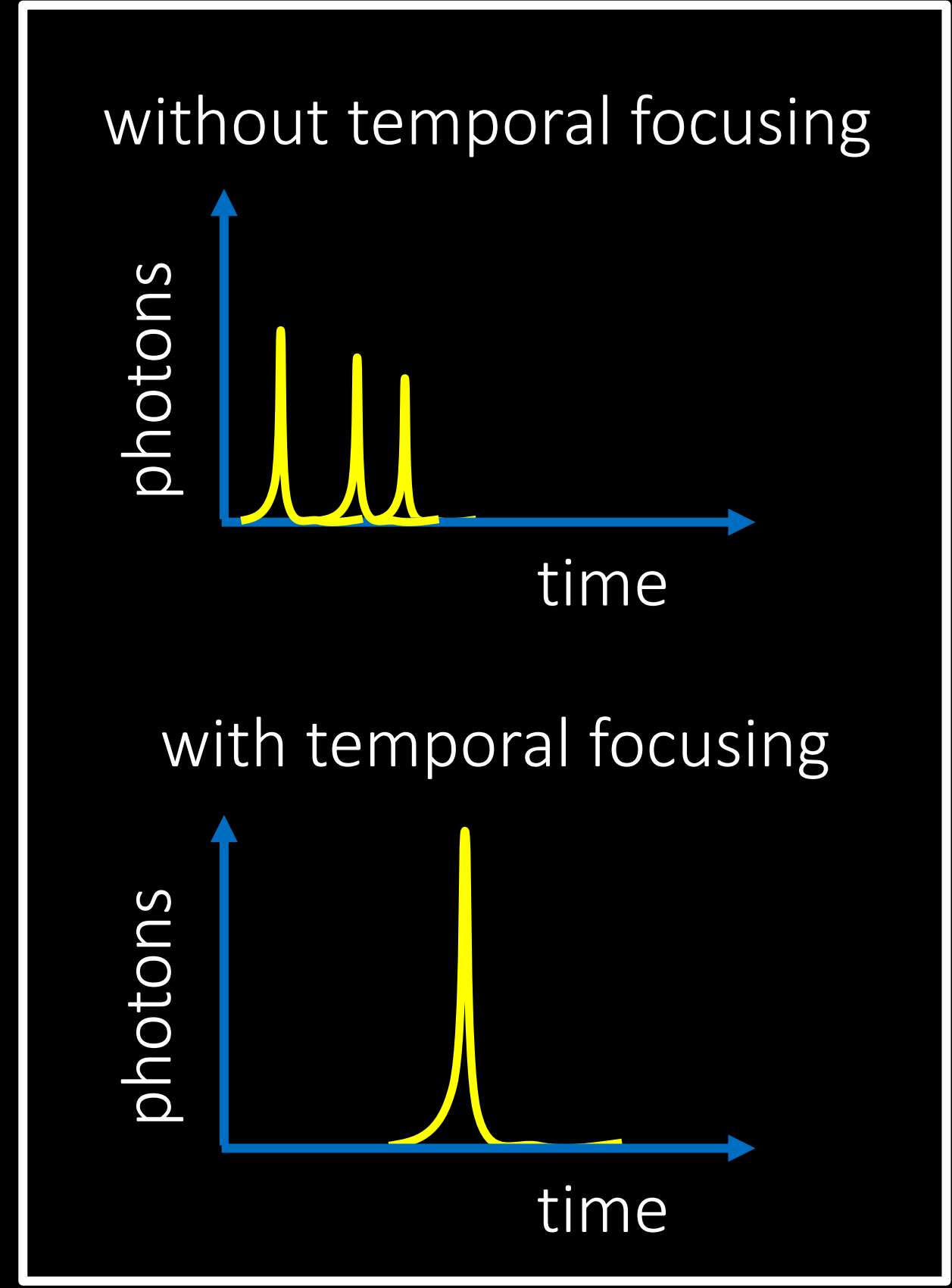
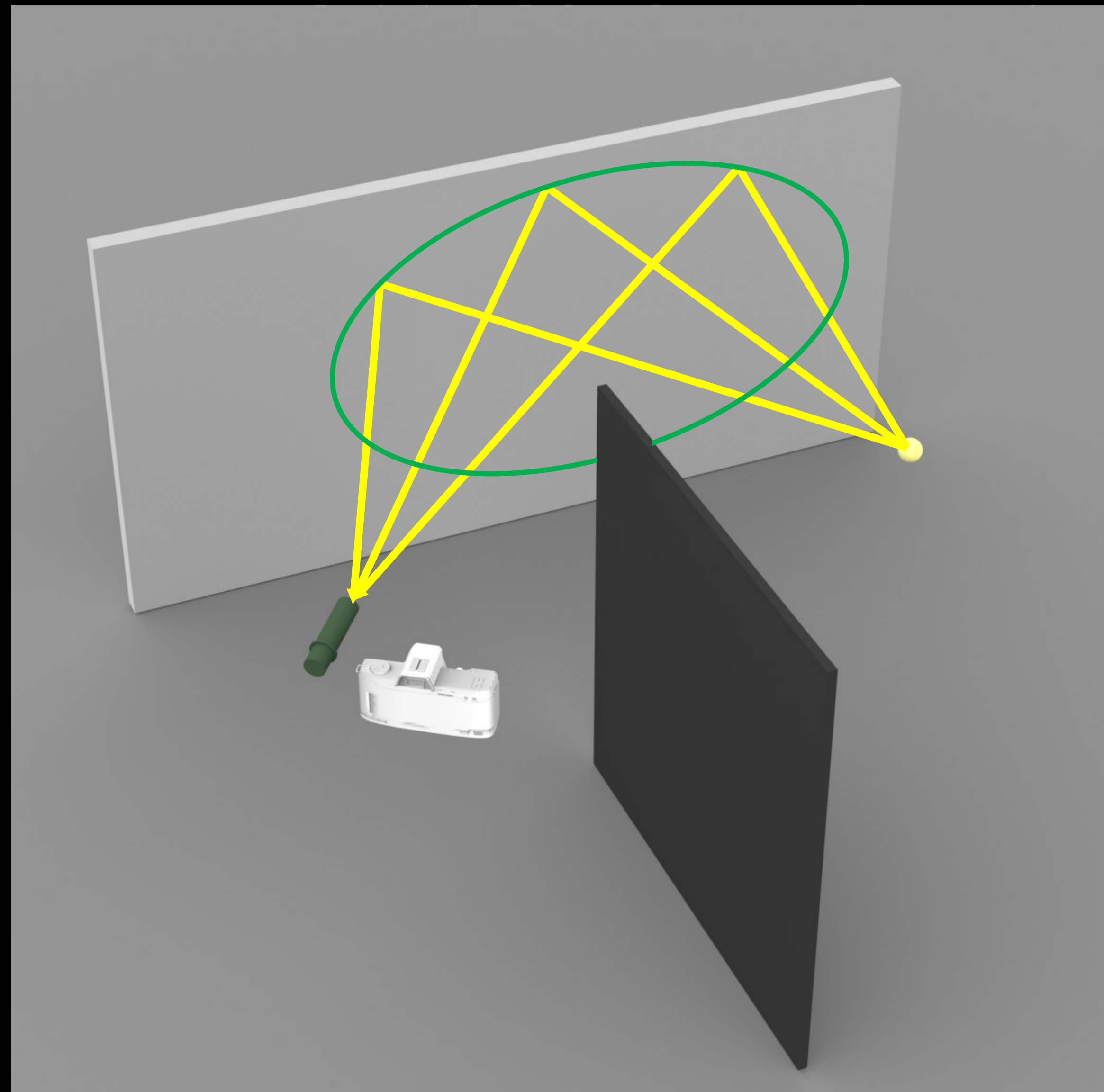
solution:
use a large lens?
expensive !!
temporal focusing:
imitate large lens



Temporal focusing: Illumination should also be an ellipse

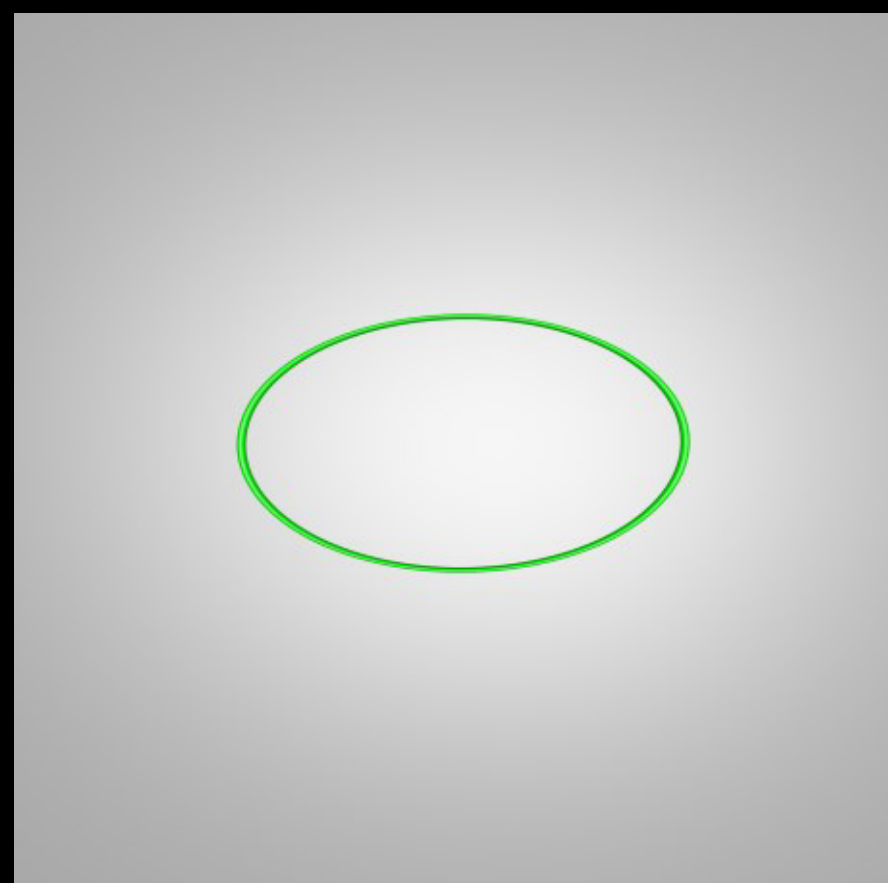
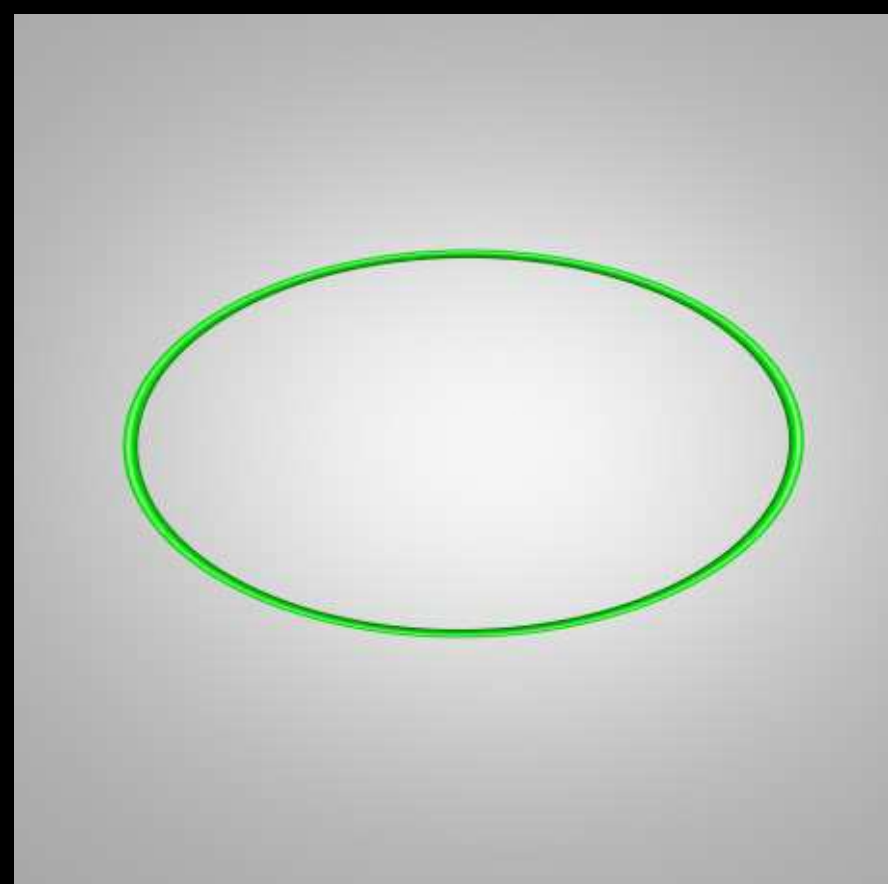
challenge:
non-specular photons

solution:
use a large lens?
expensive !!
temporal focusing:
imitate large lens

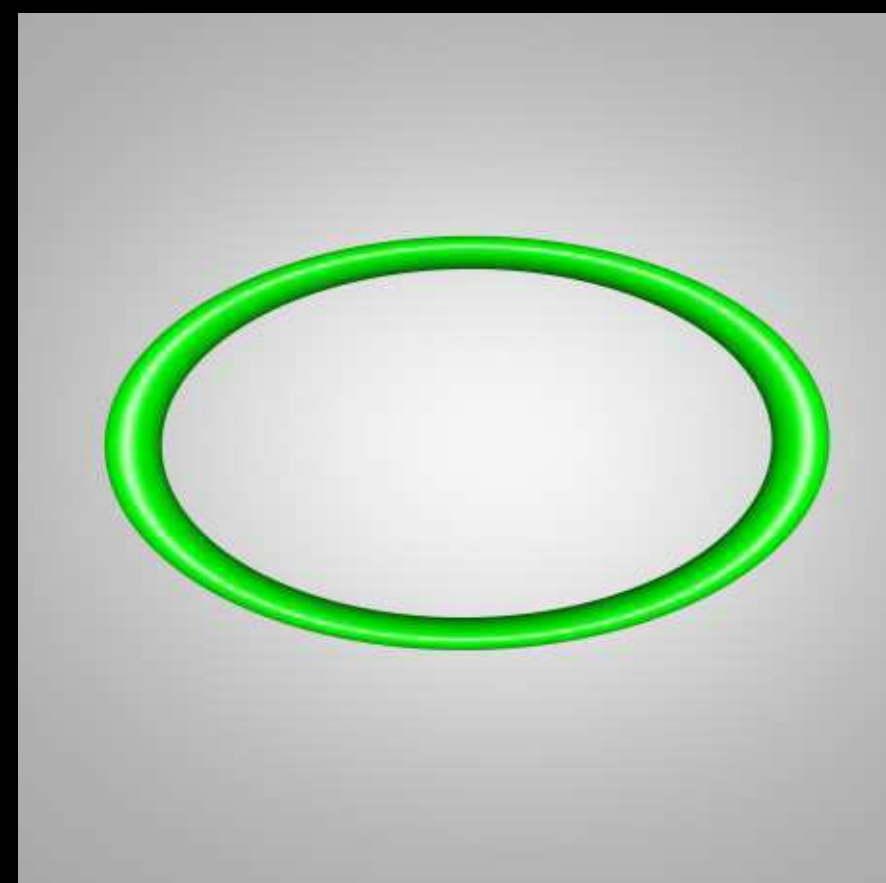
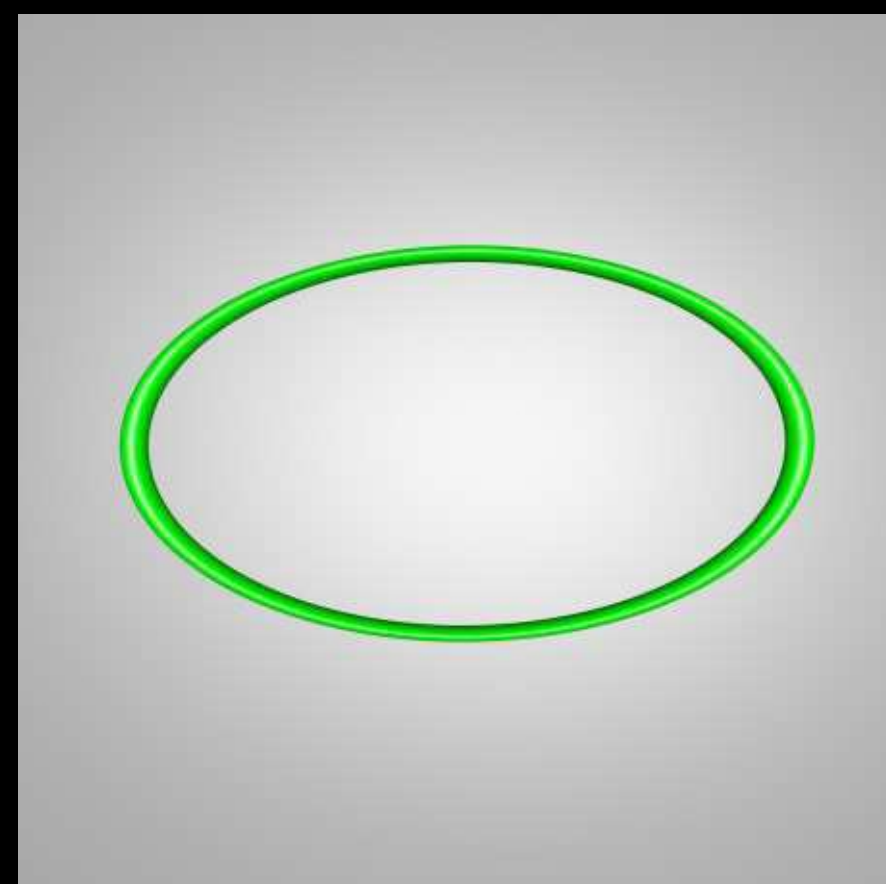


Design choices for temporal focusing

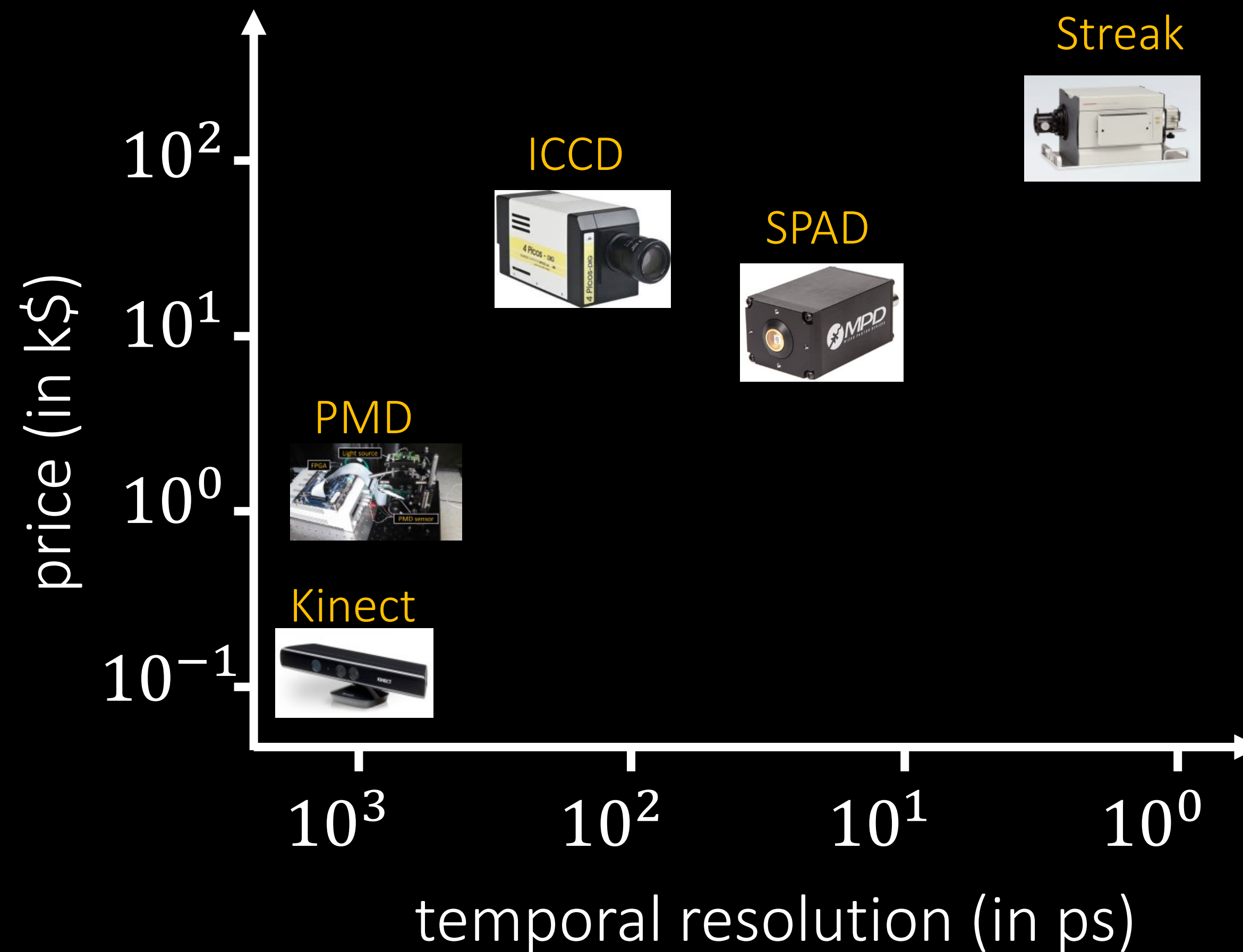
ellipse size



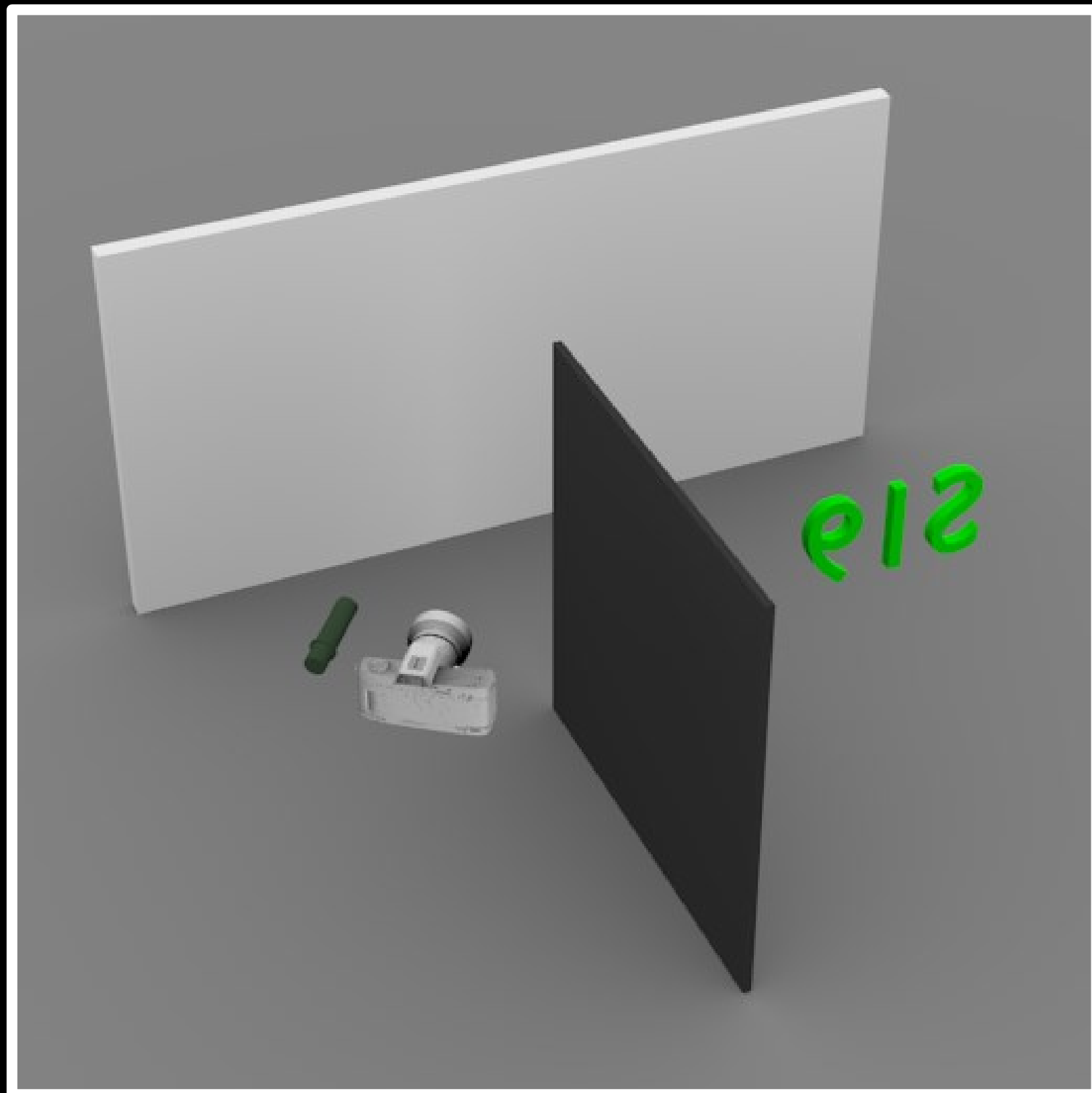
ellipse thickness



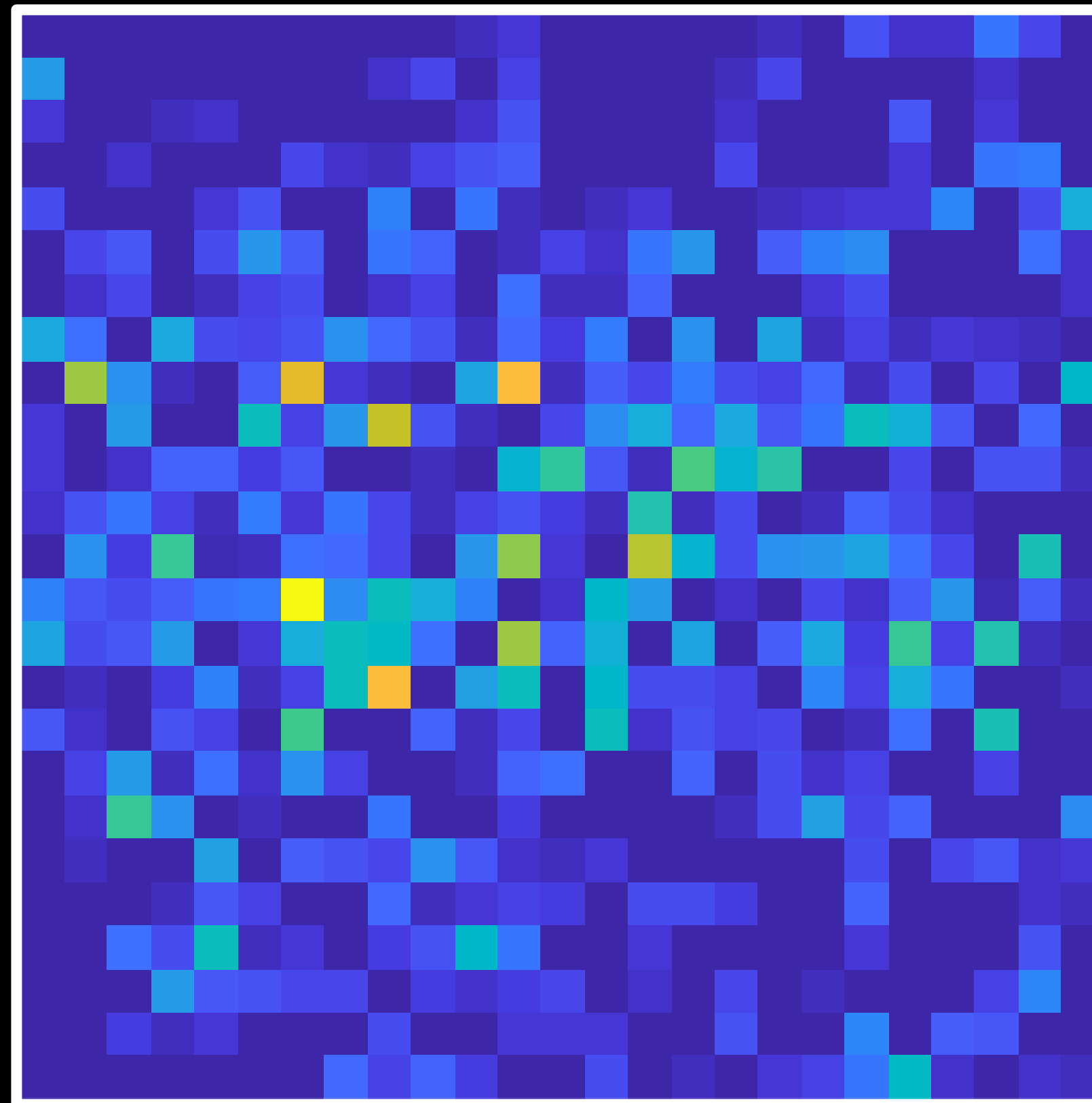
time-of-flight cameras



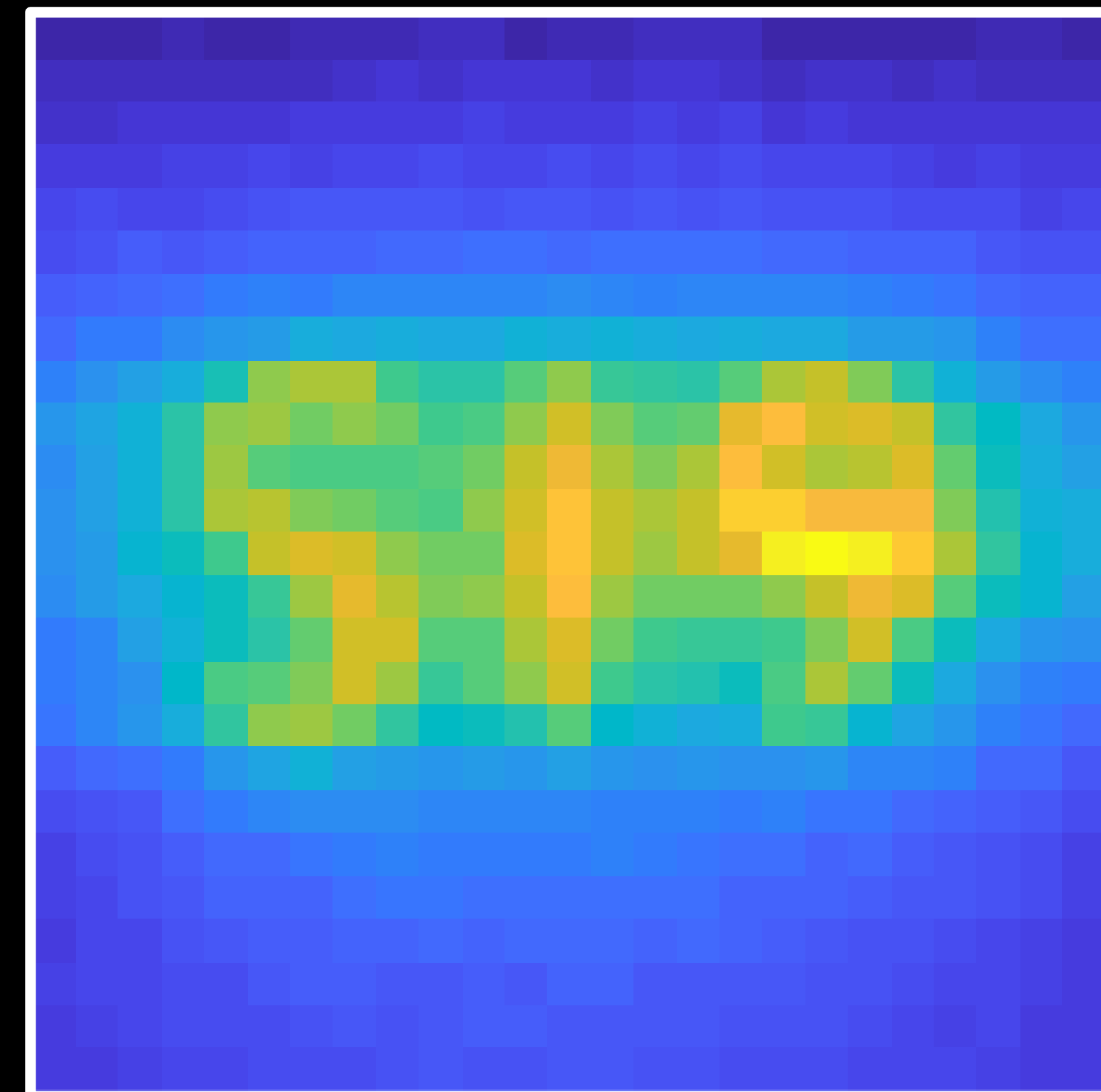
Rendering non-line-of-sight imaging by temporal focusing



scene



standard BDPT



BDPT w/ ellipsoidal connections

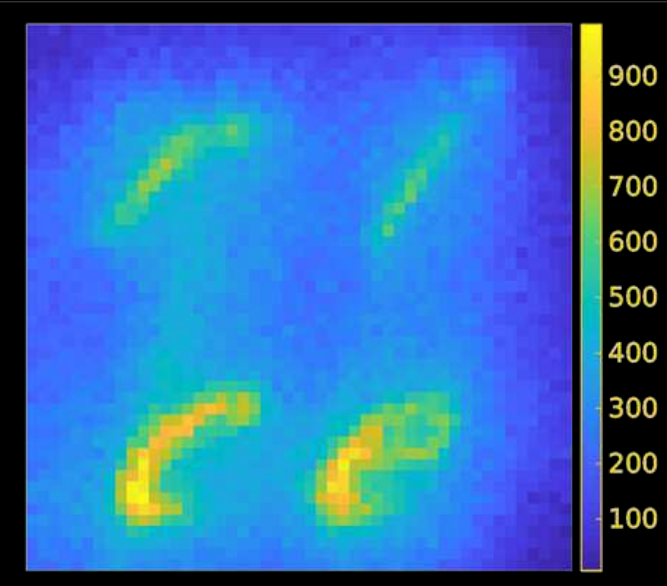
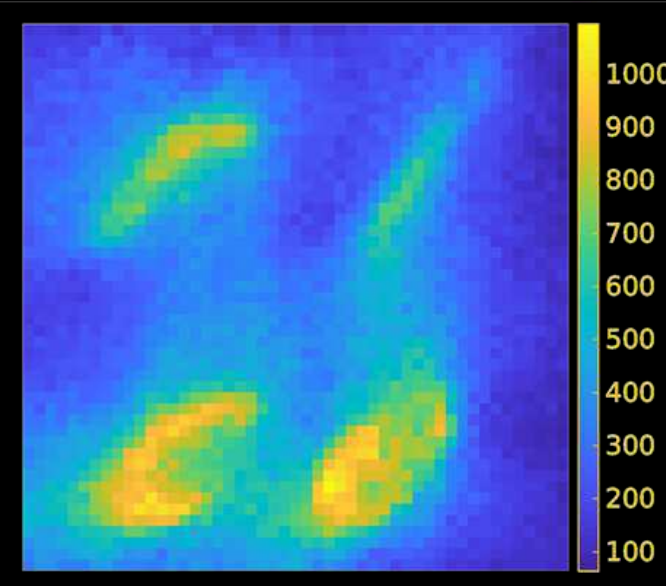
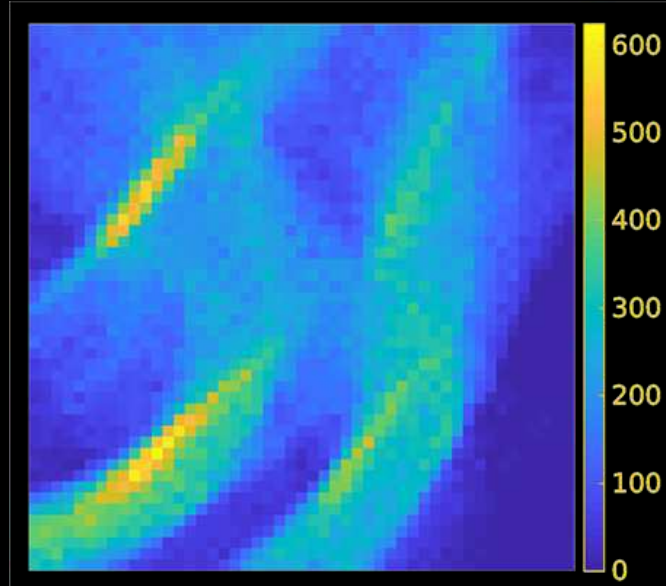
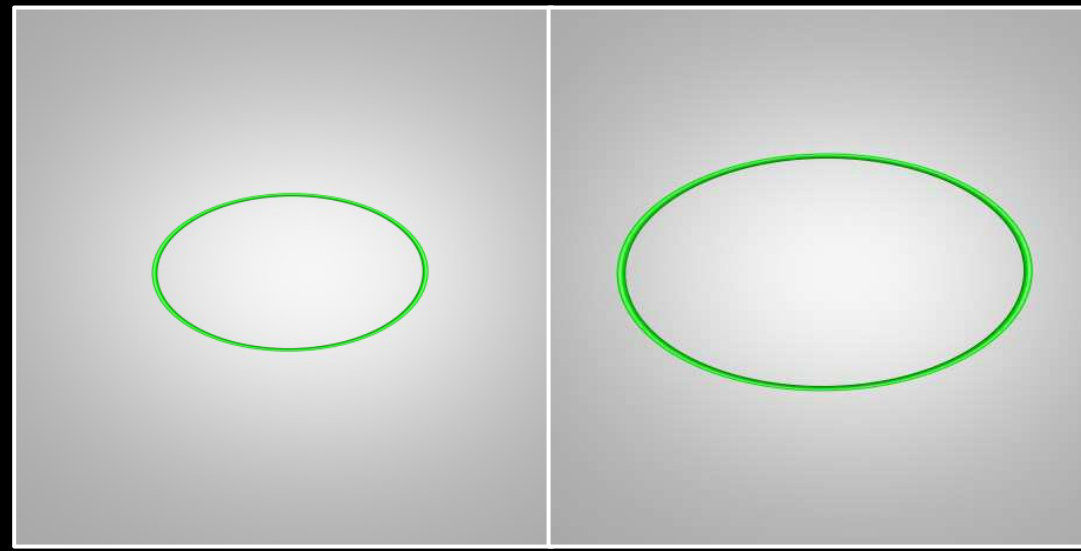
Gate width: 4 ps (0.4% scene)
Rendering time: 3 hr

* simulation results

Design choices

ellipse

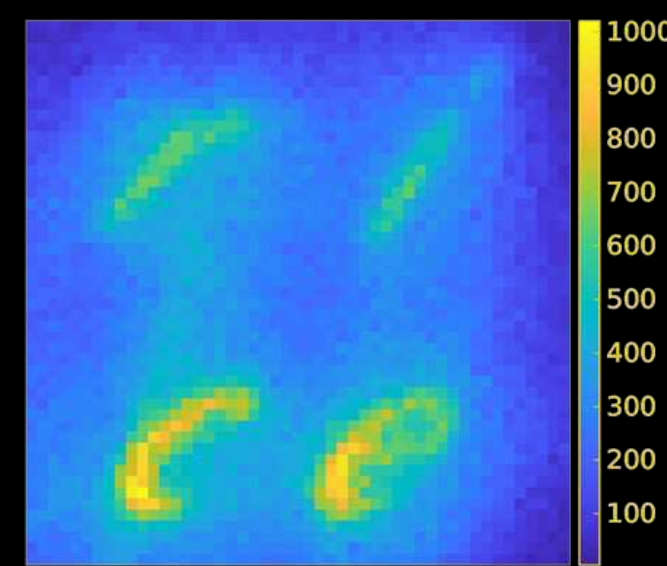
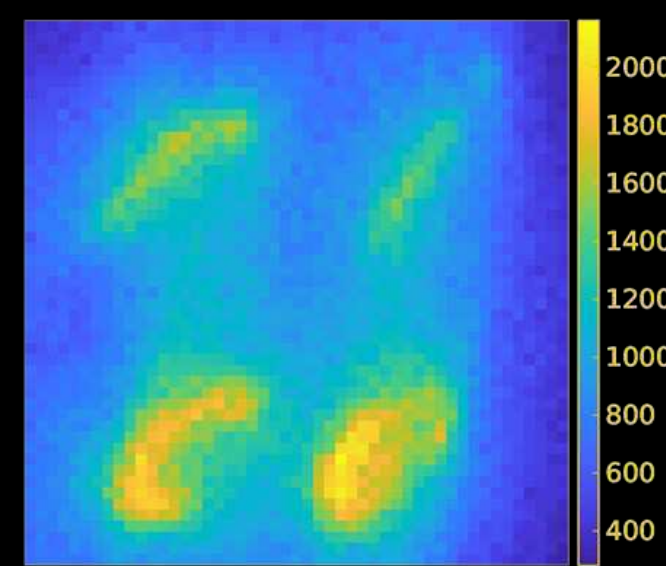
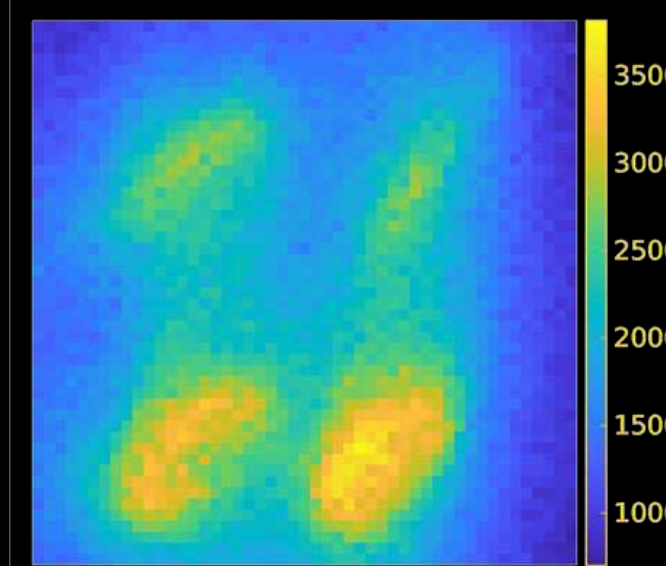
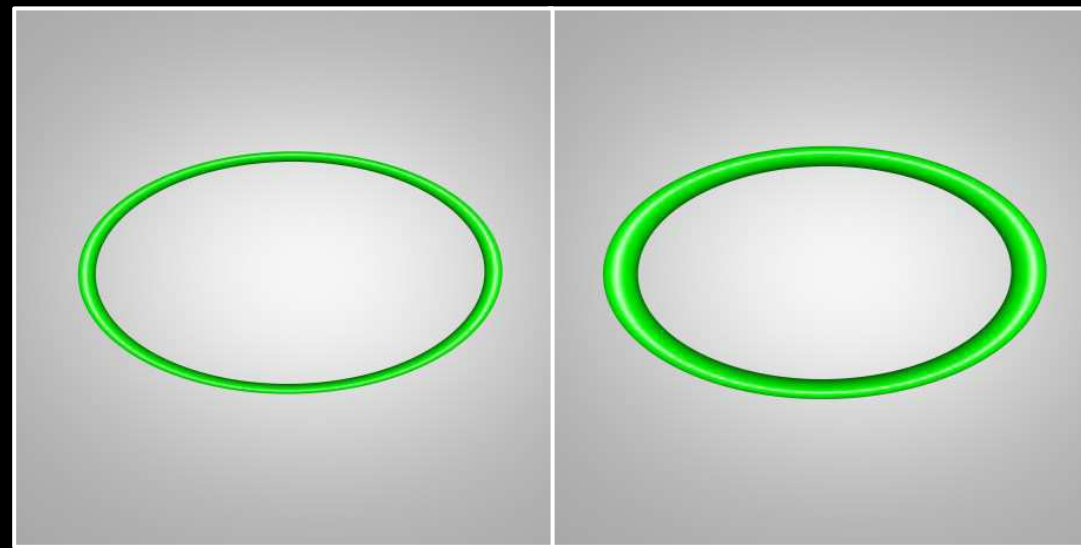
size



large ellipses result in better resolution

ellipse

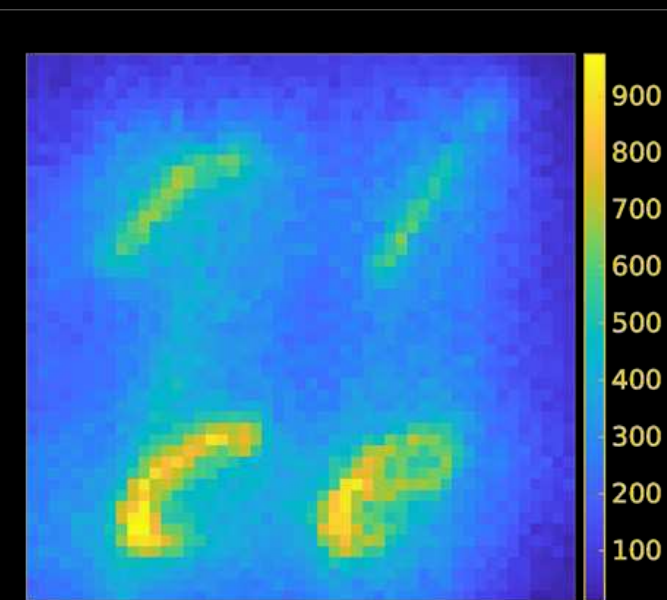
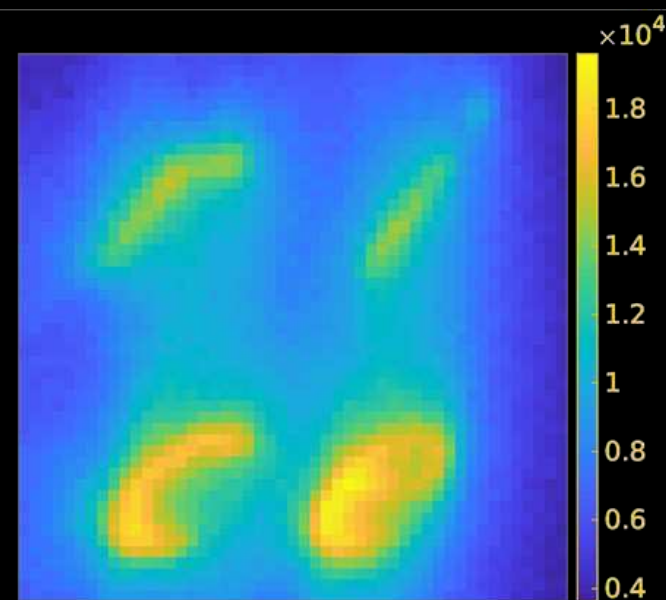
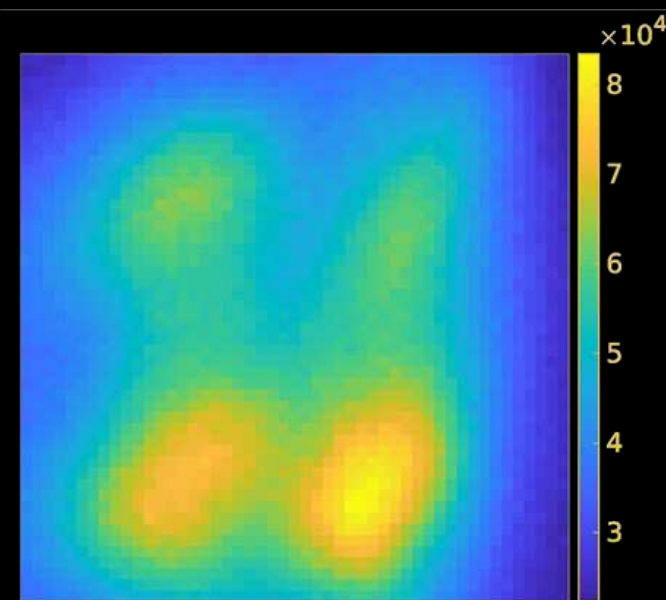
thickness



thinner ellipses result in better resolution, but loses light

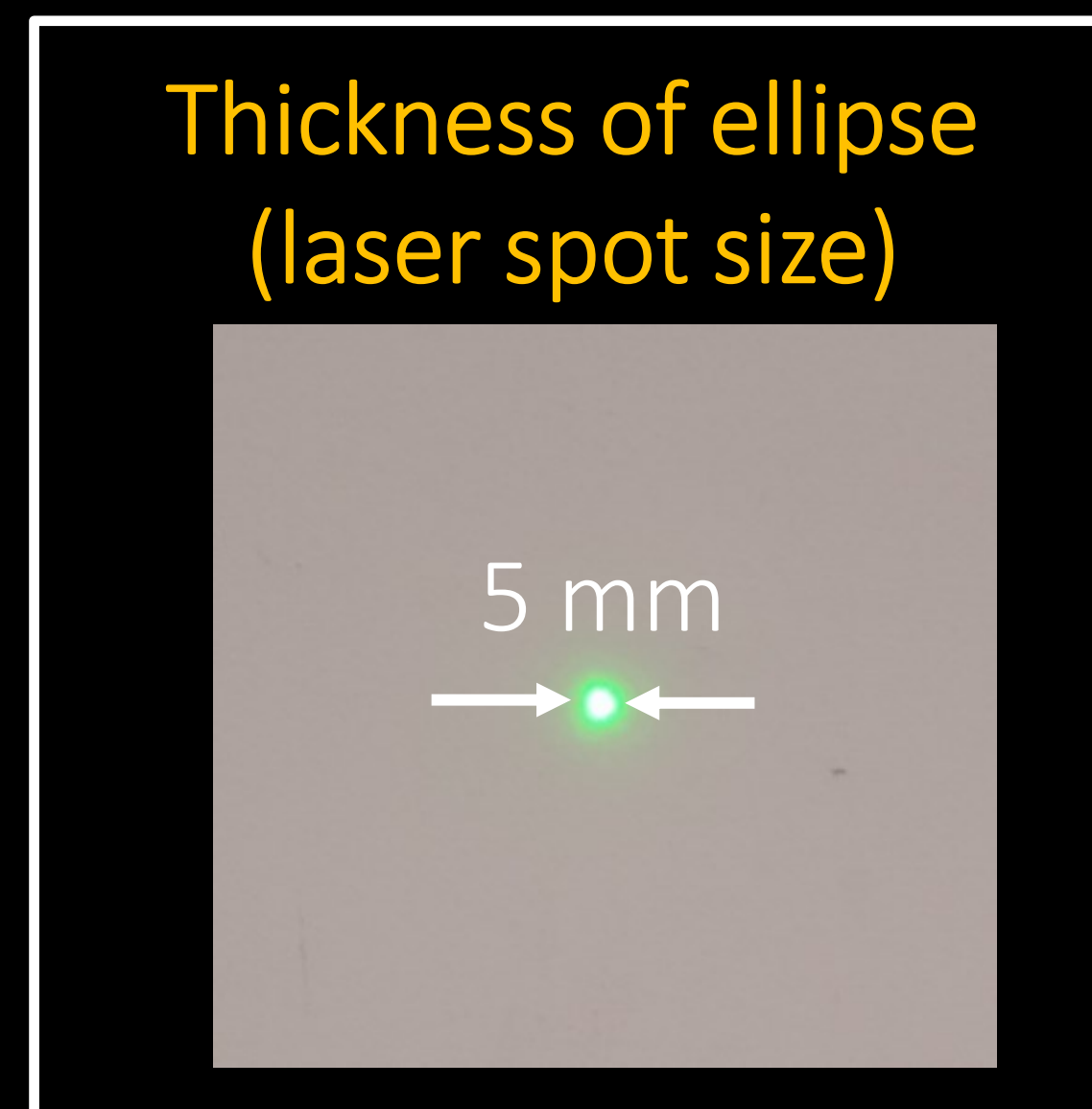
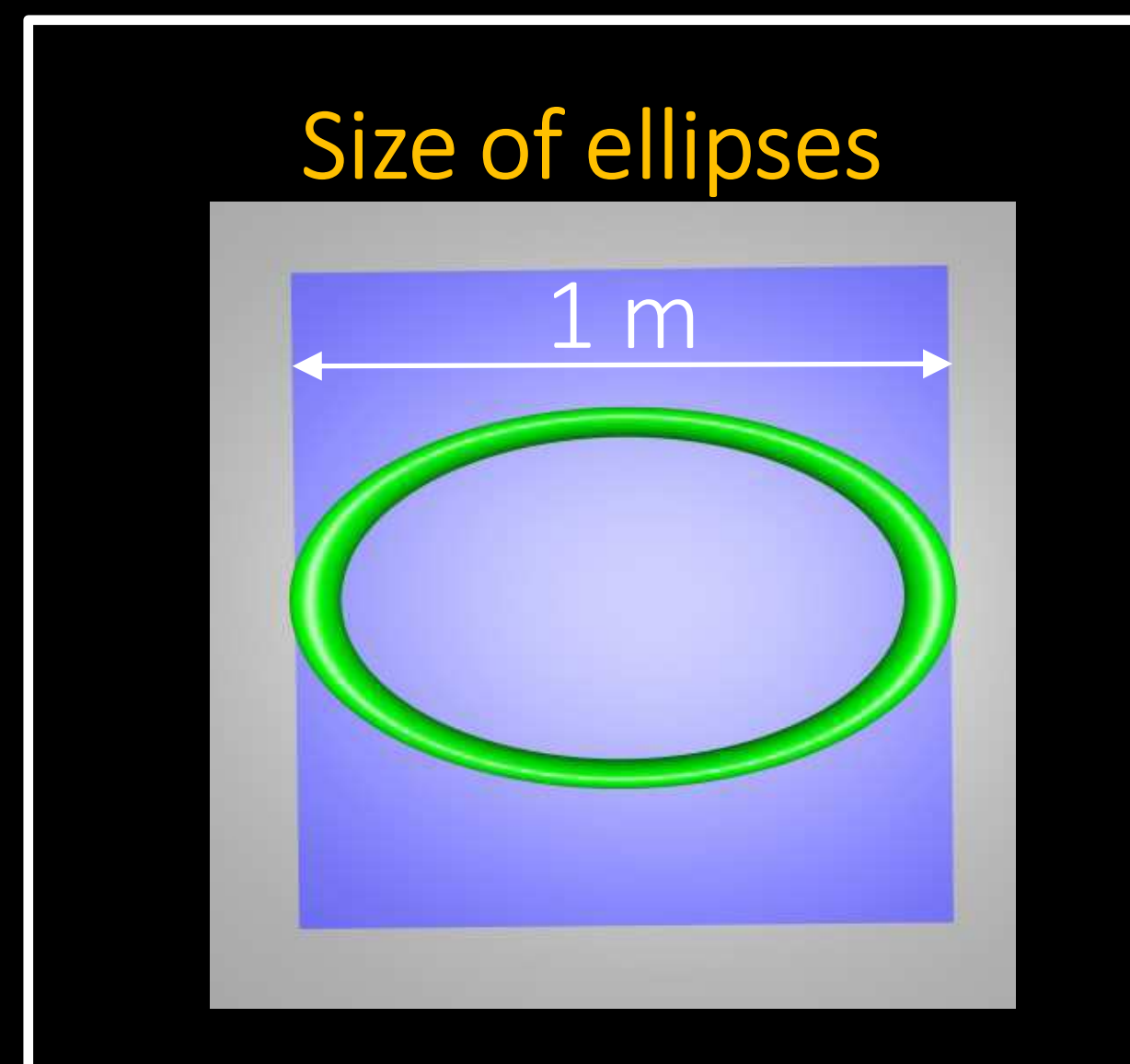
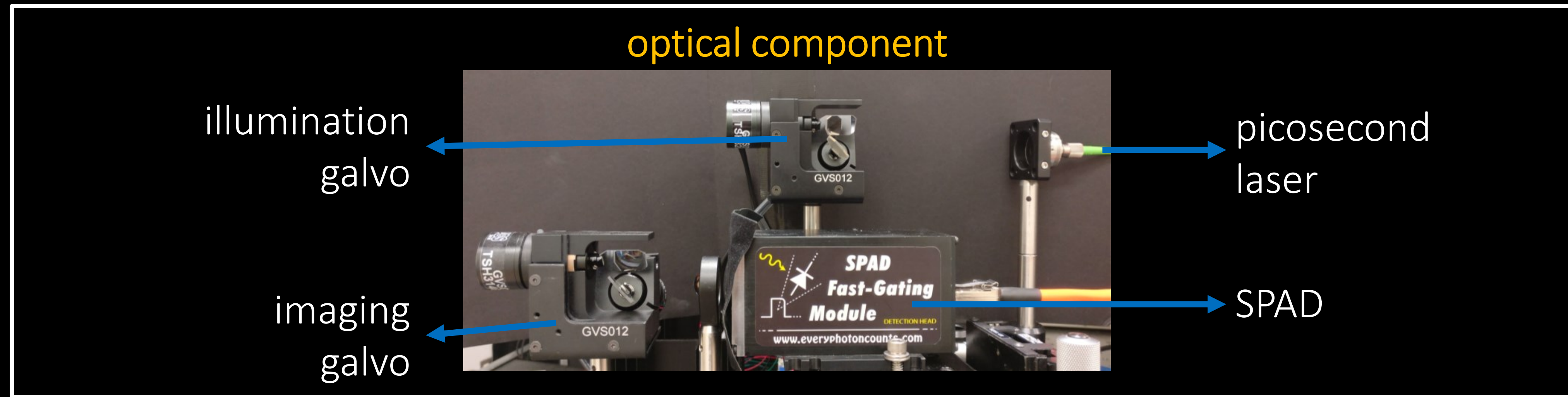
temporal

resolution

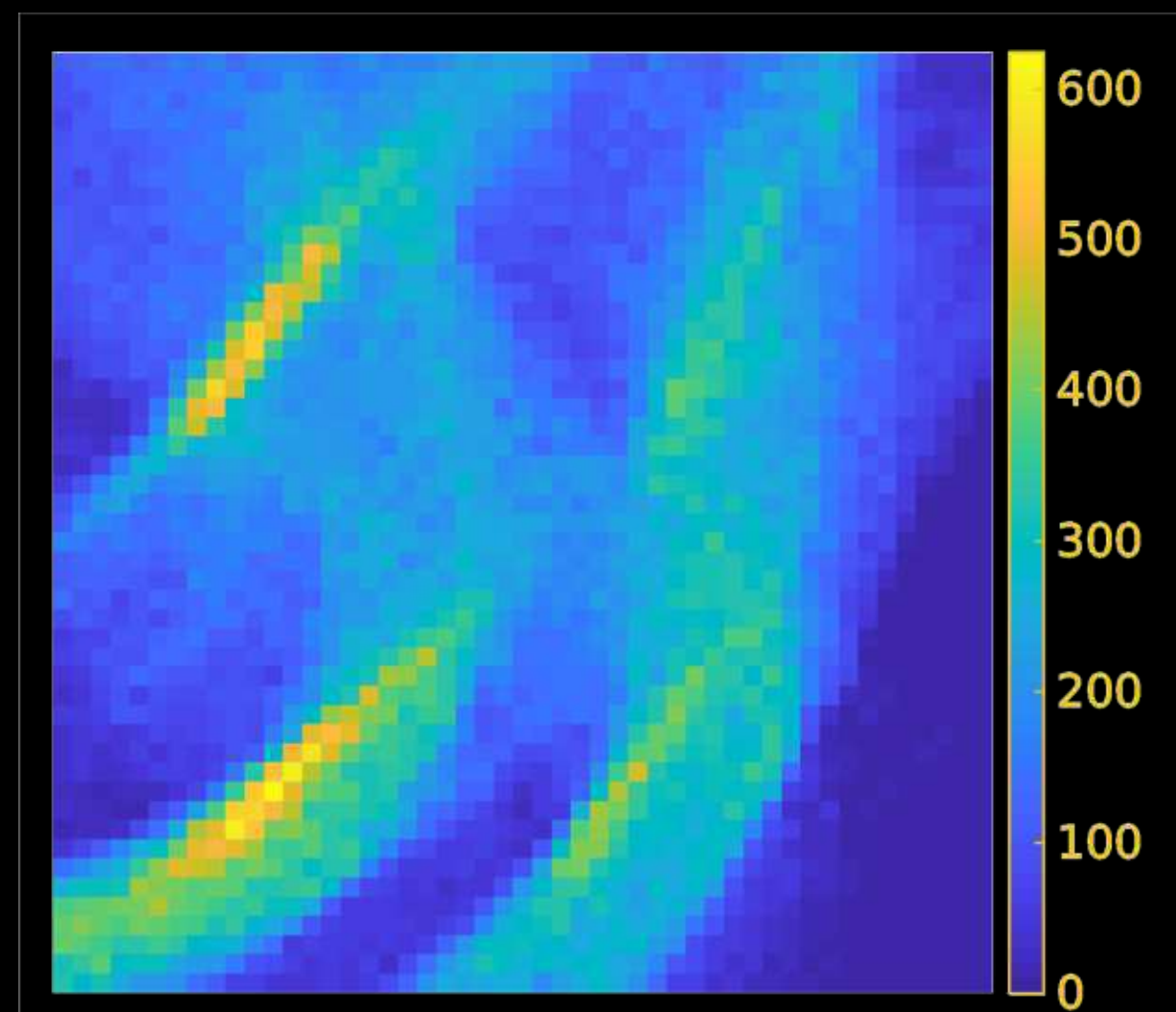
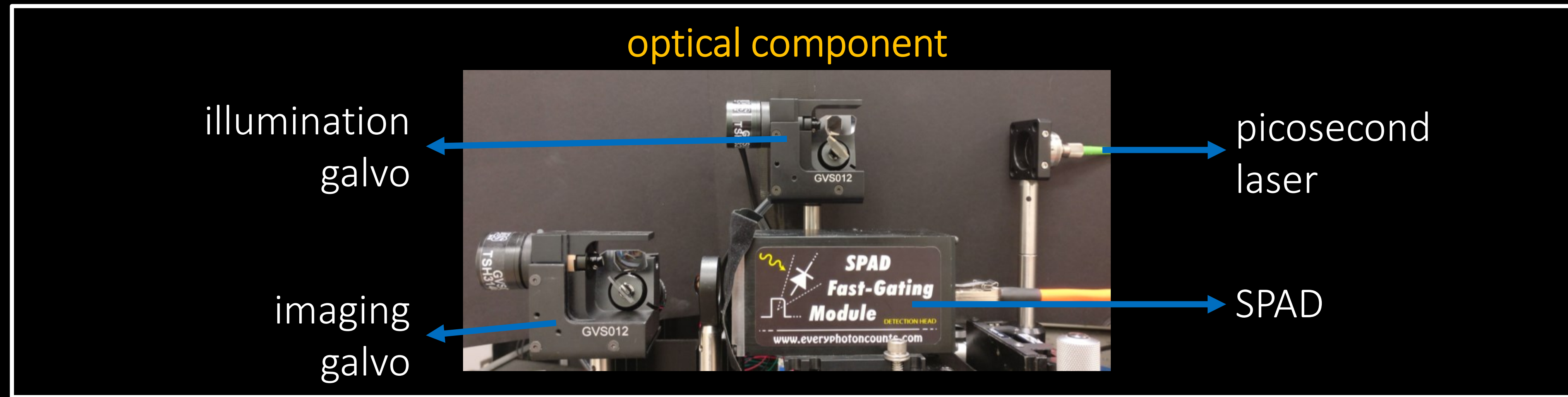


high resolution time-gate results in better resolution, but loses light

Hardware prototype

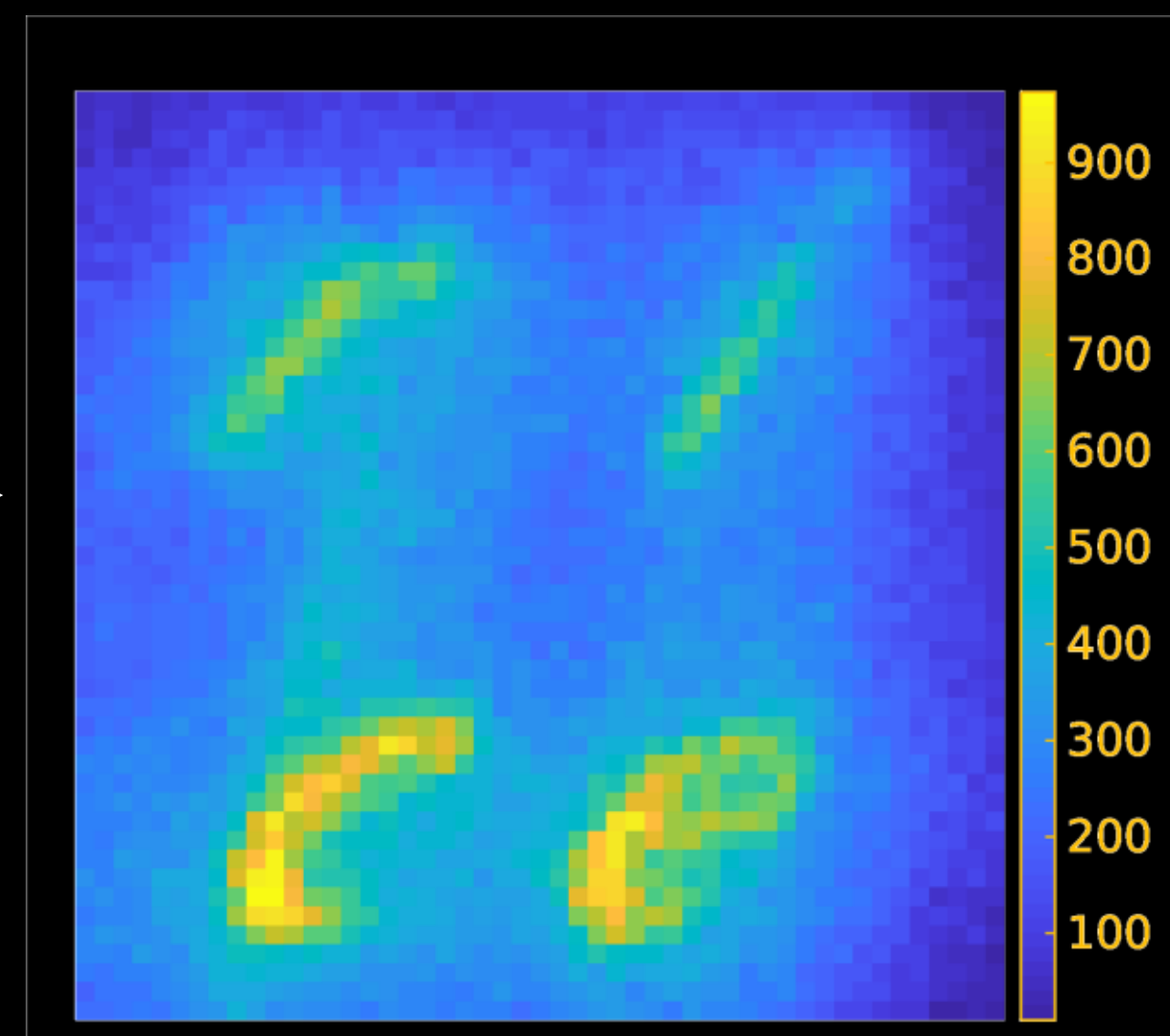


Hardware prototype



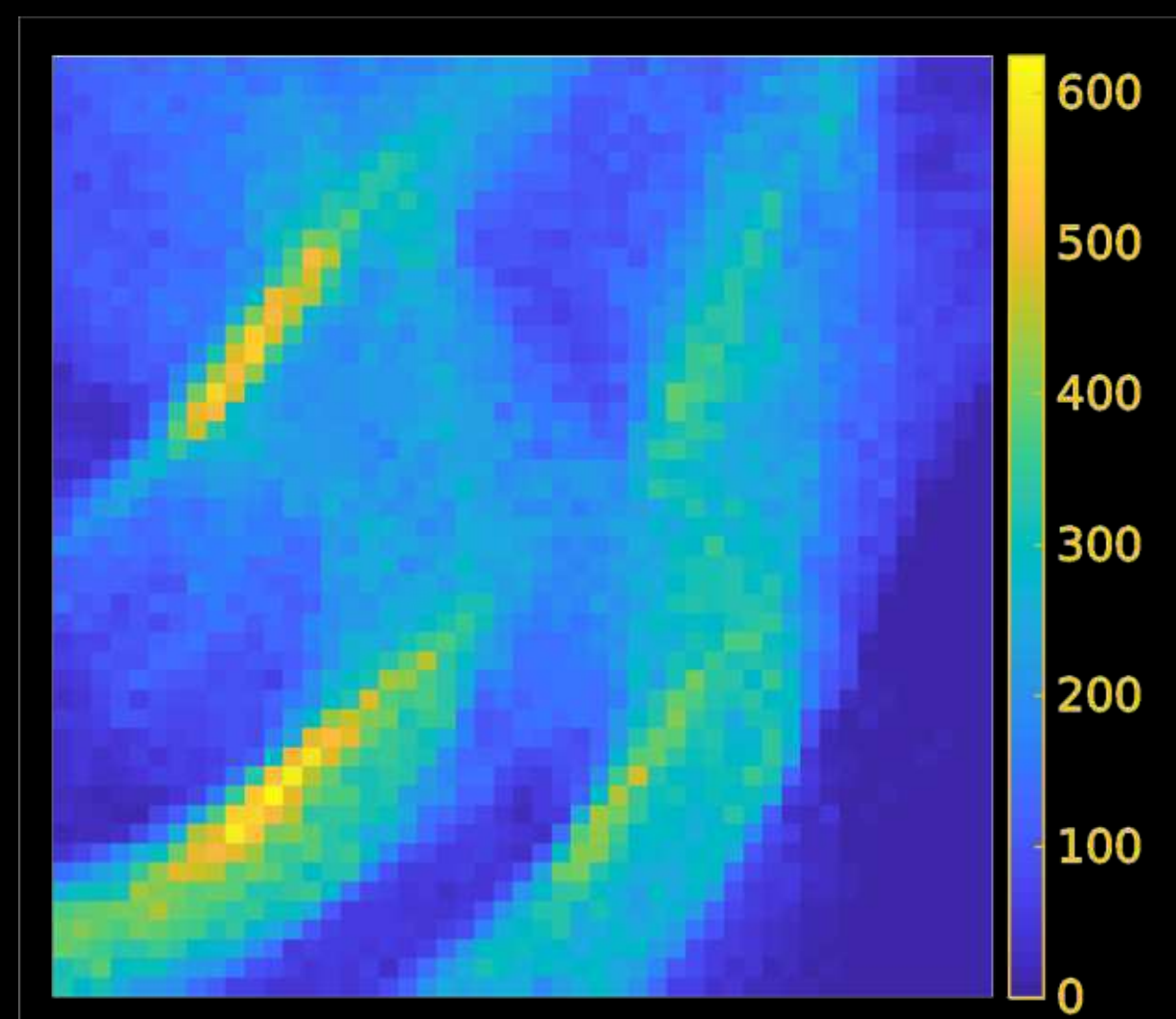
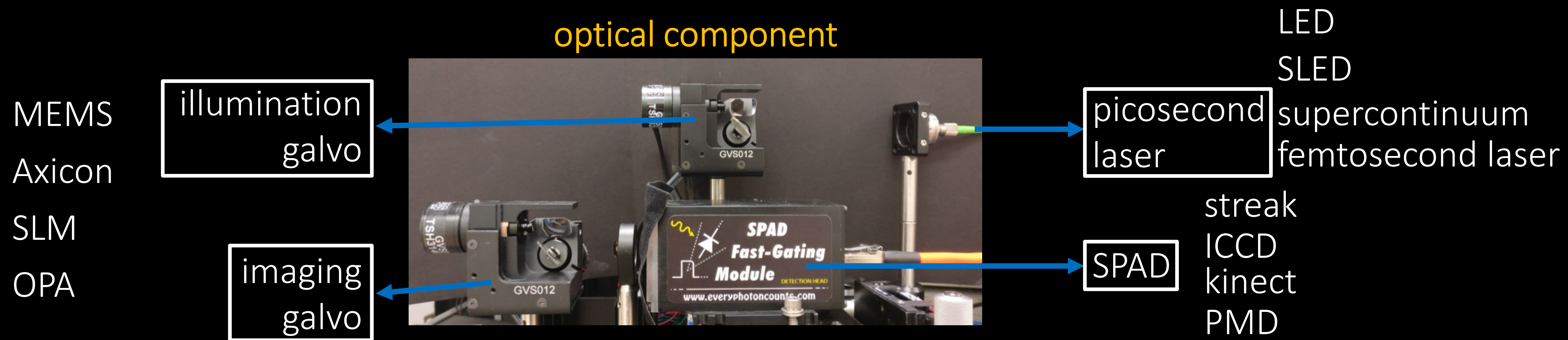
unoptimized

renderer driven
optimization



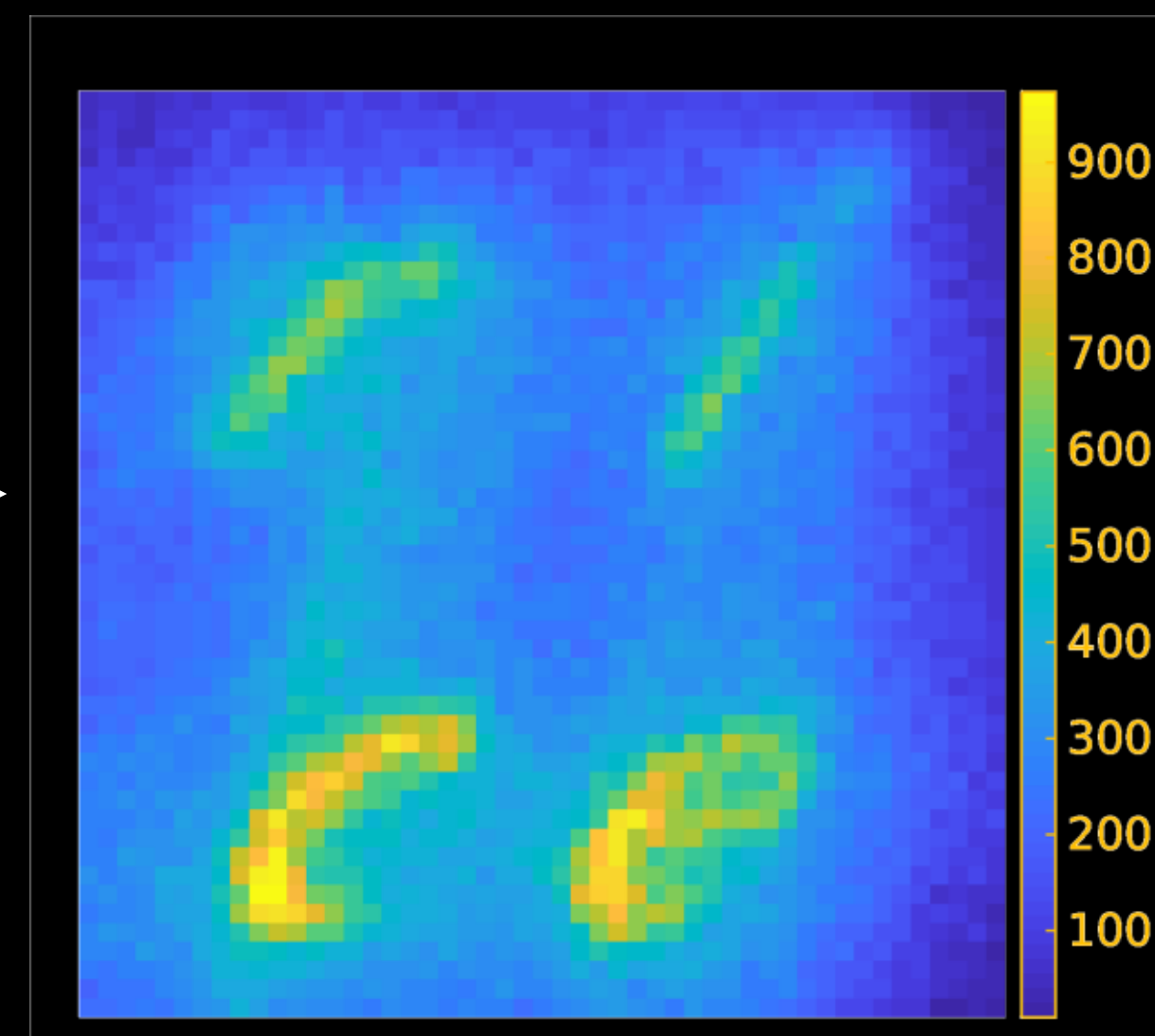
optimized

Hardware prototype



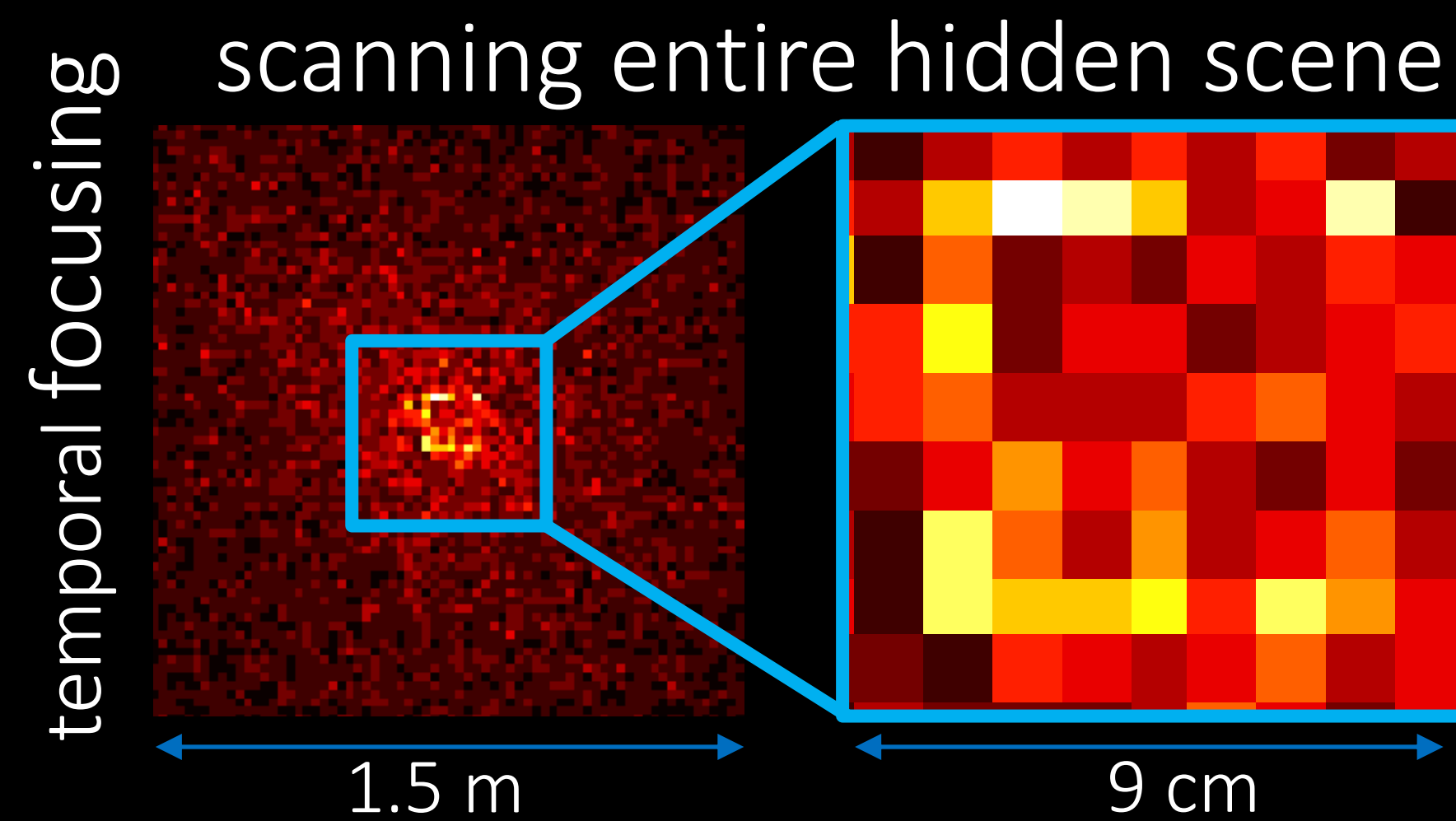
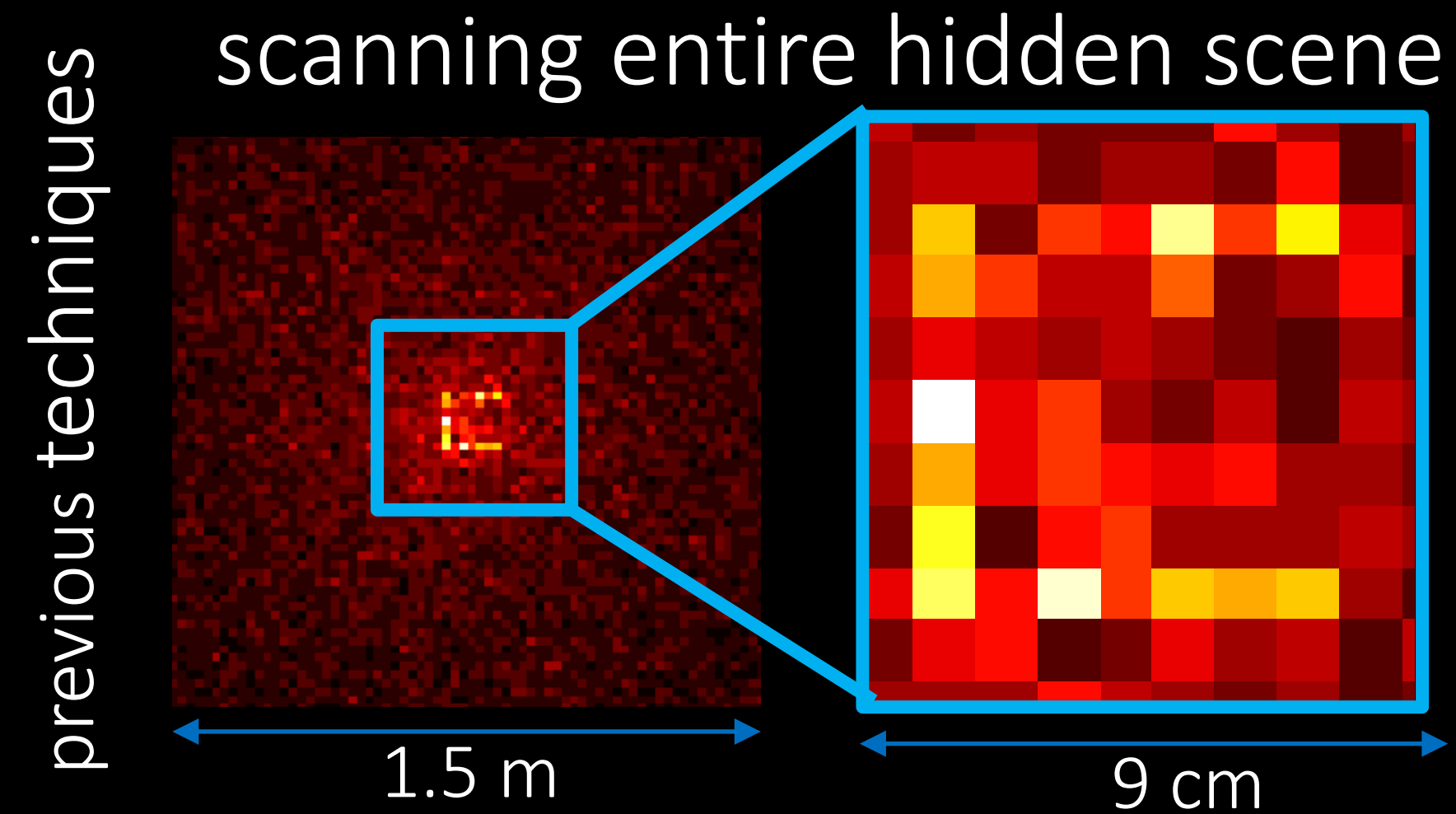
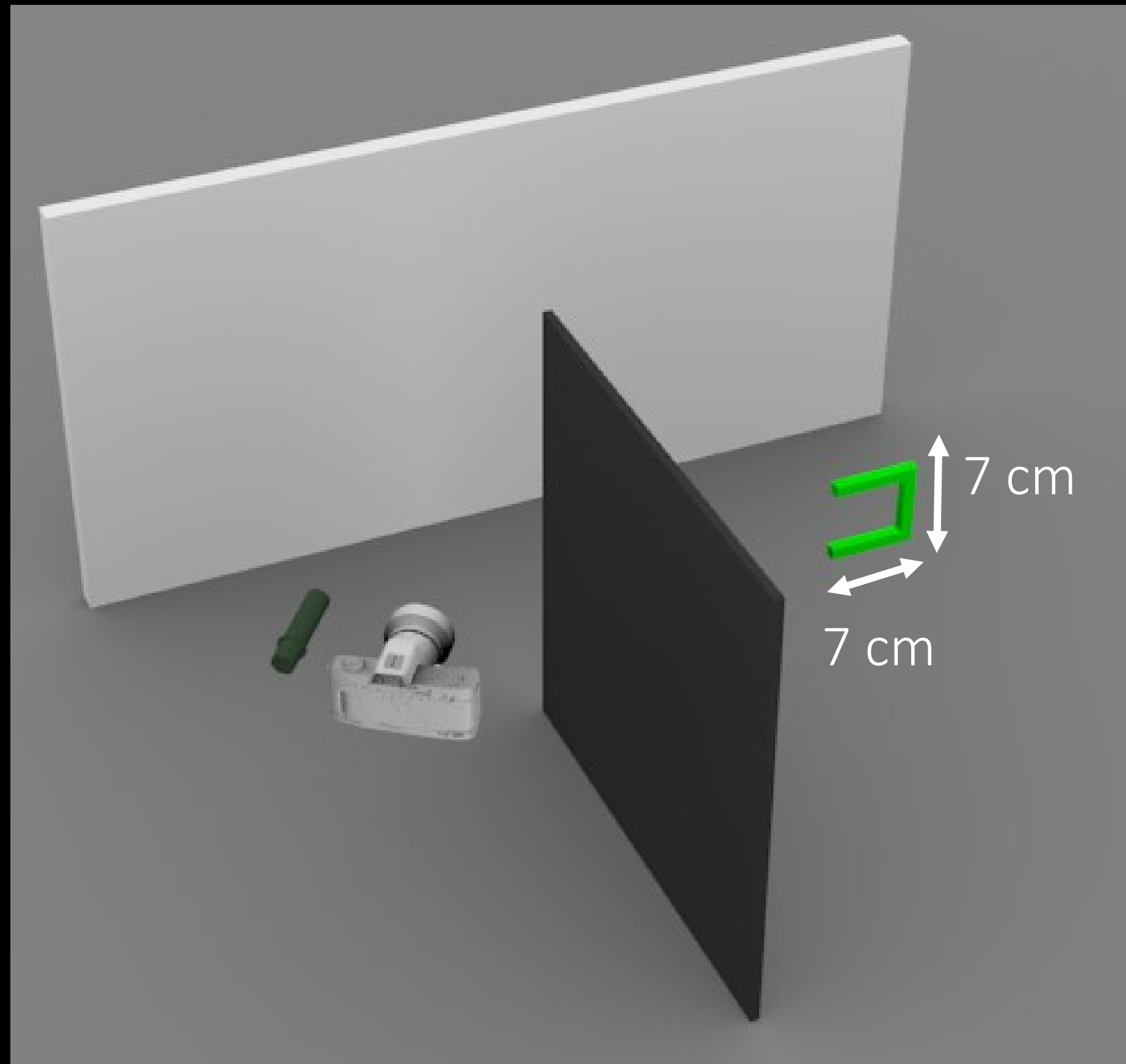
unoptimized

renderer driven
optimization

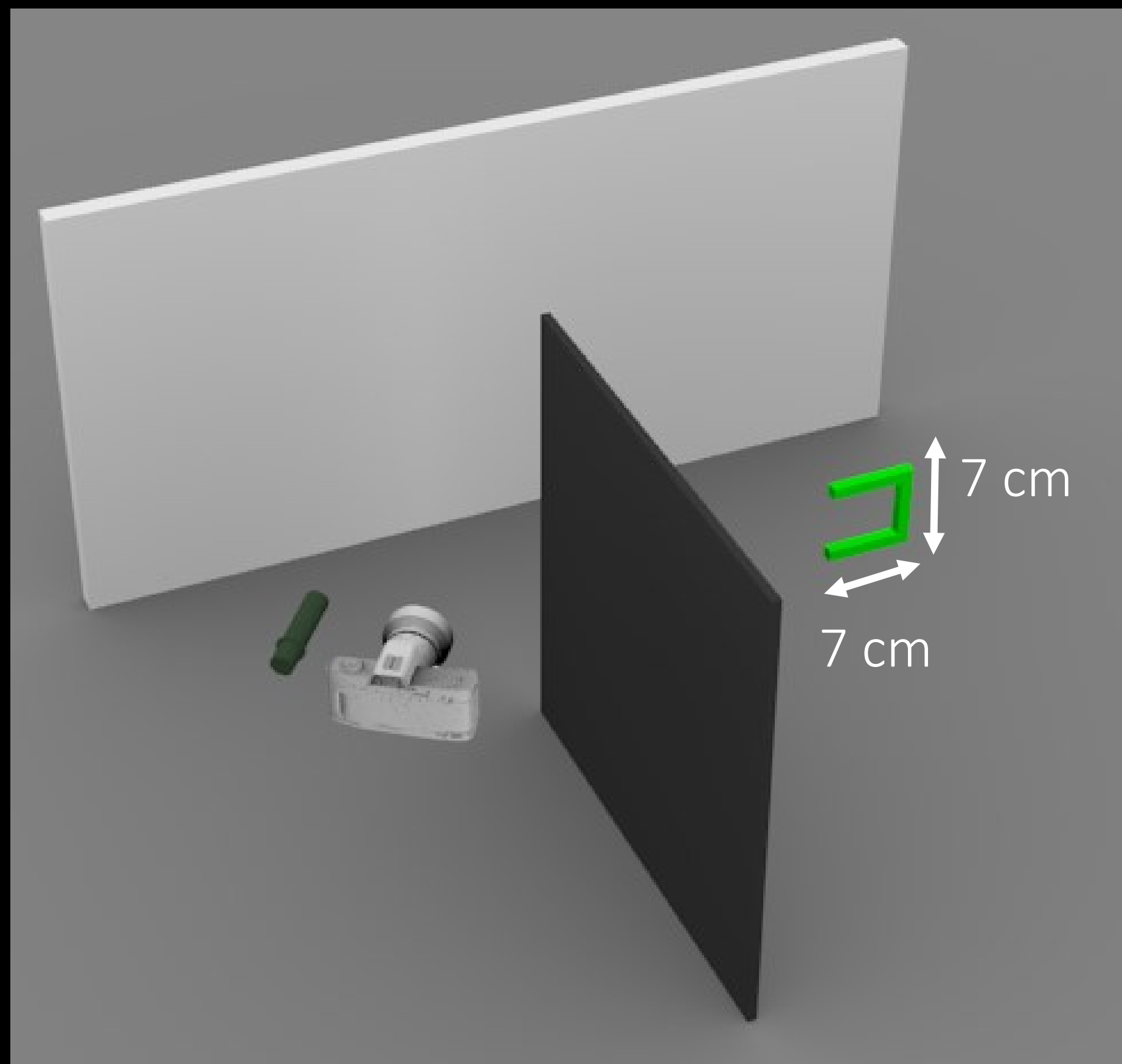


optimized

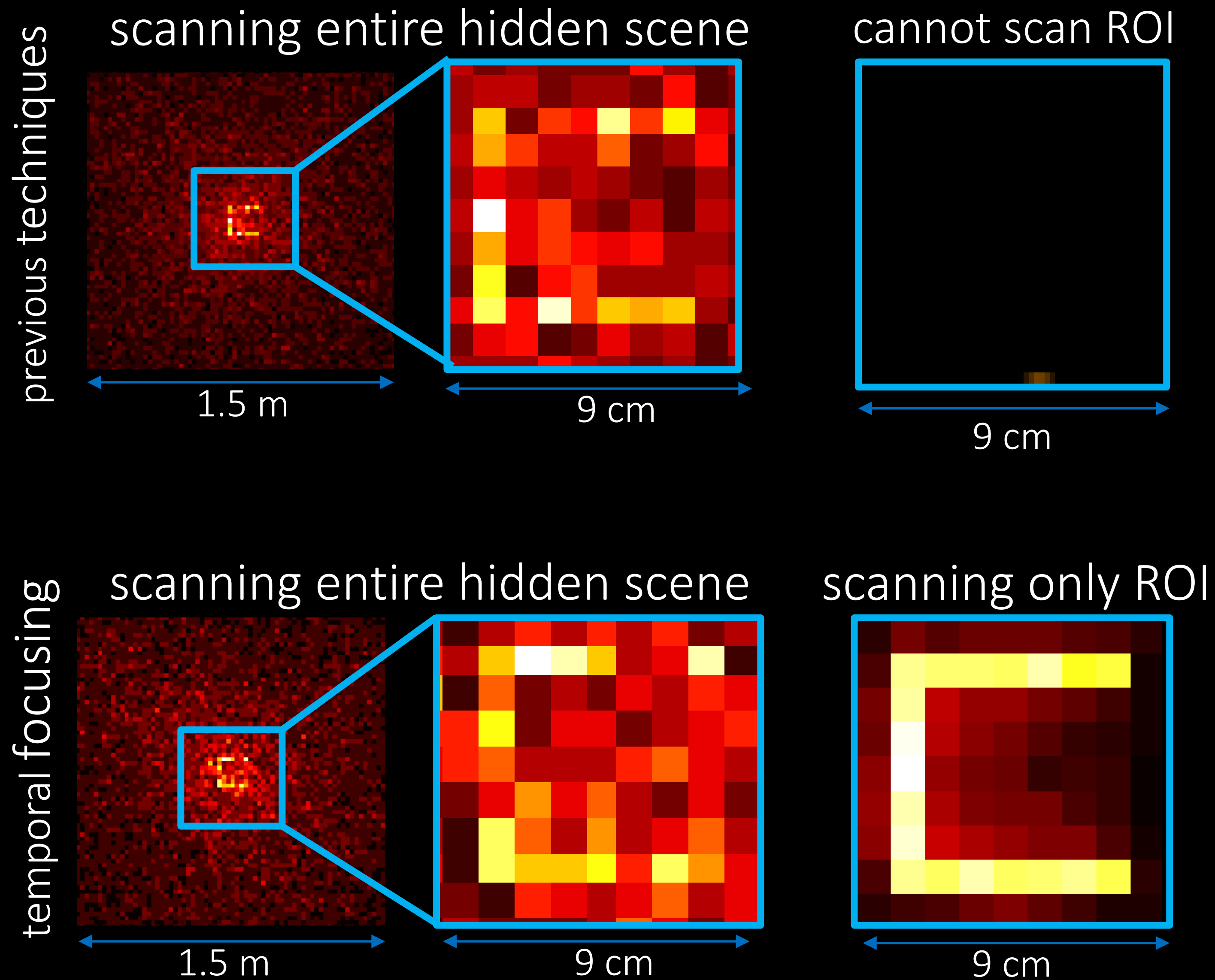
Results: scanning limited ROI



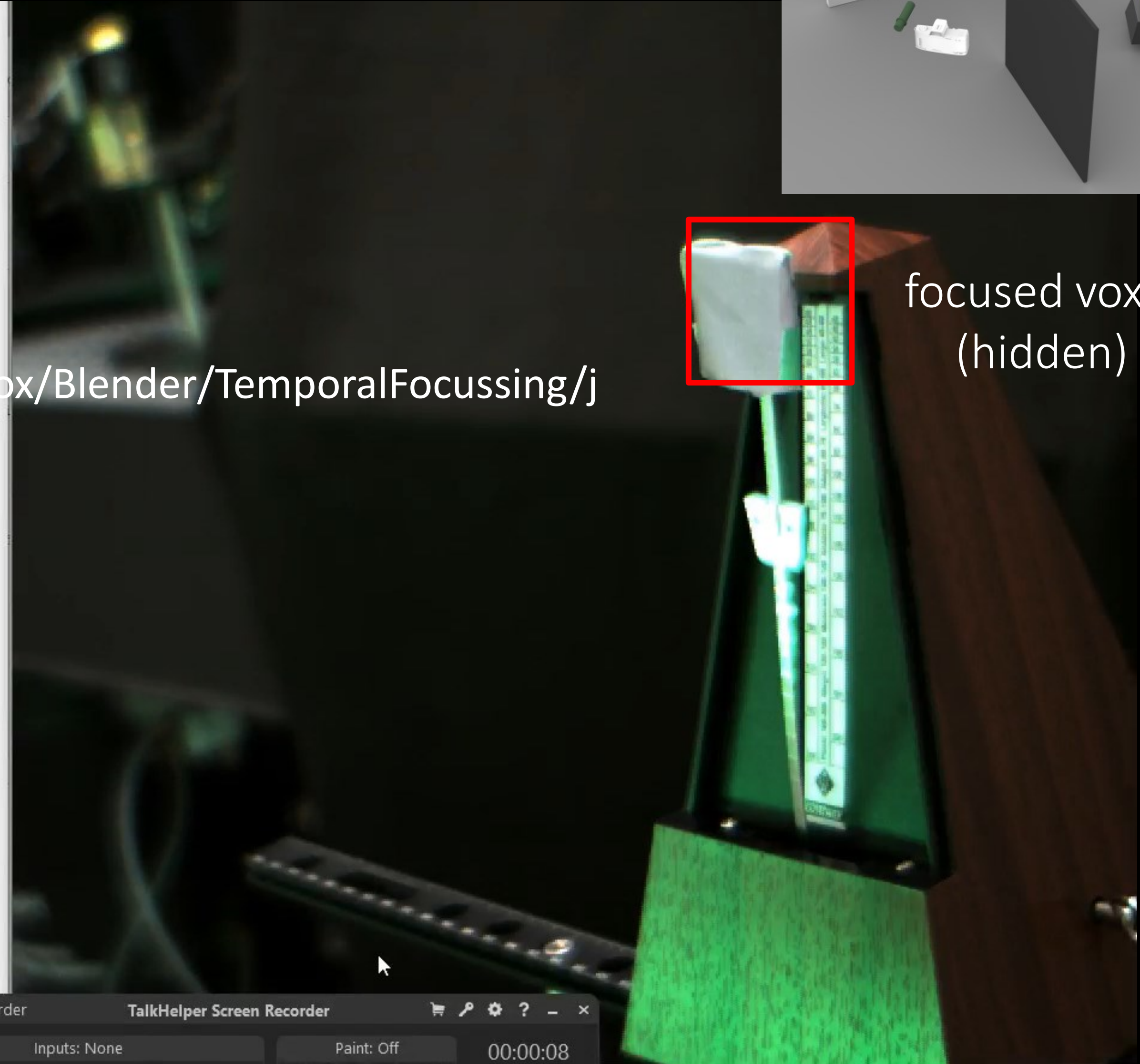
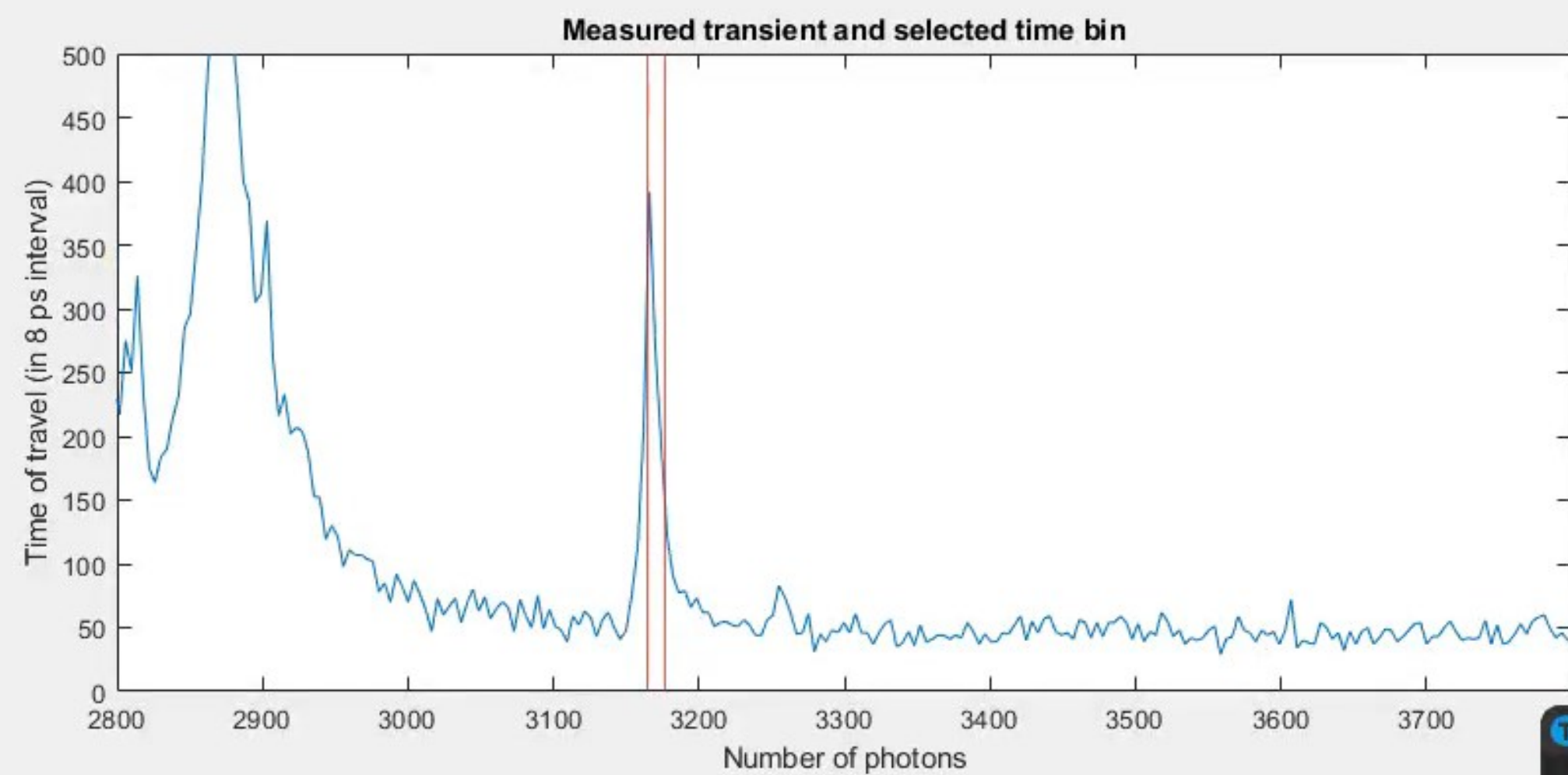
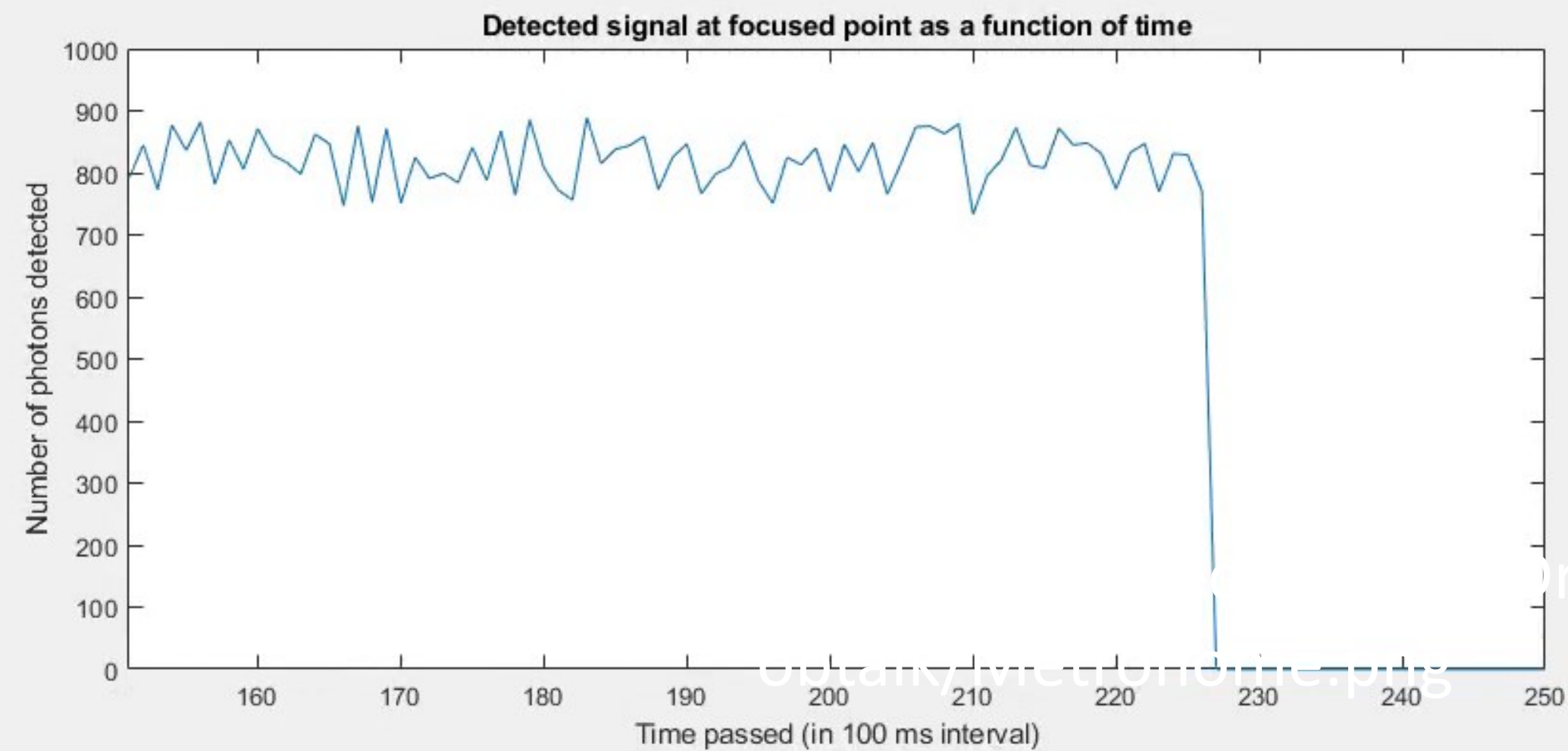
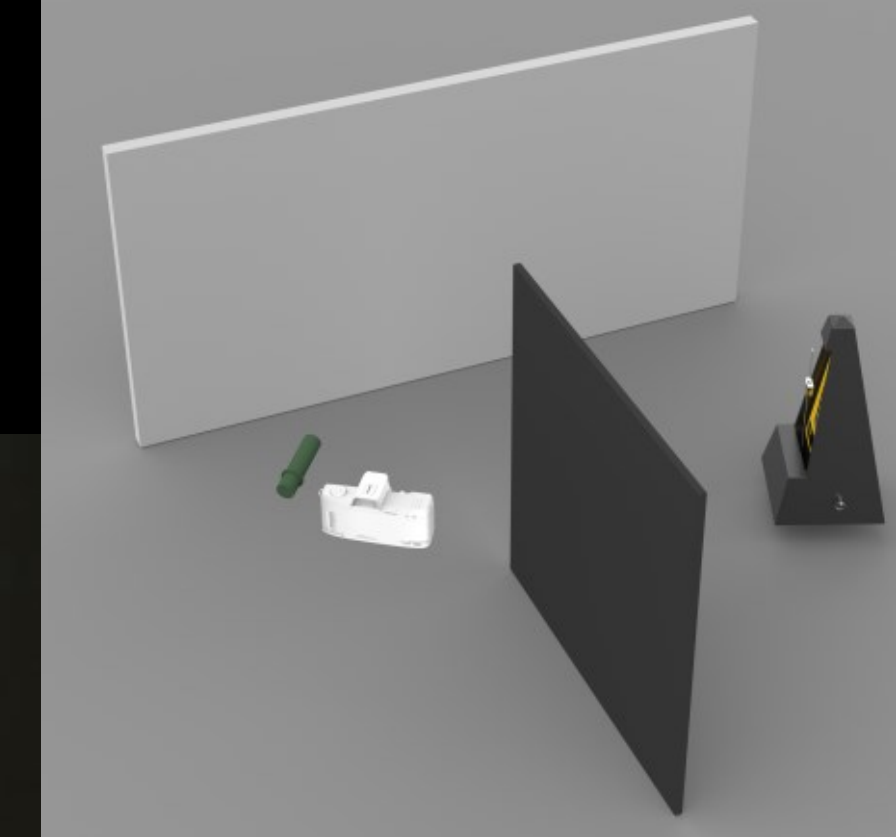
Results: scanning limited ROI



SNR of temporal focusing is $> 10\times$ higher for small ROI



Results: real-time occupancy detection



Video Recorder TalkHelper Screen Recorder
Inputs: None Paint: Off 00:00:08

Physics-based rendering and its applications to computational imaging

forward rendering

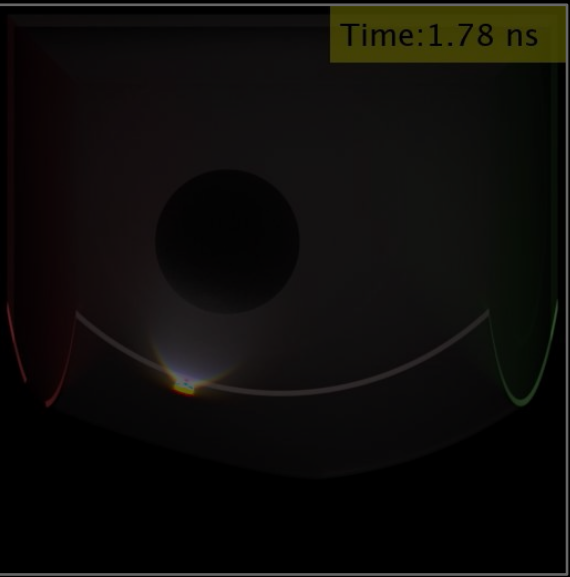


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

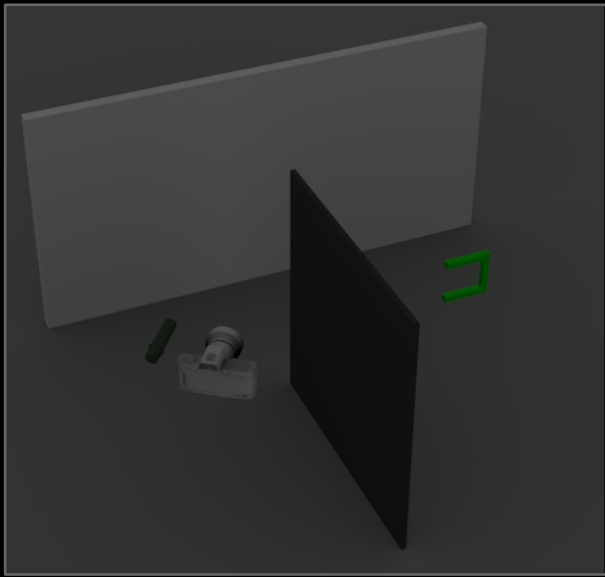
inverse rendering



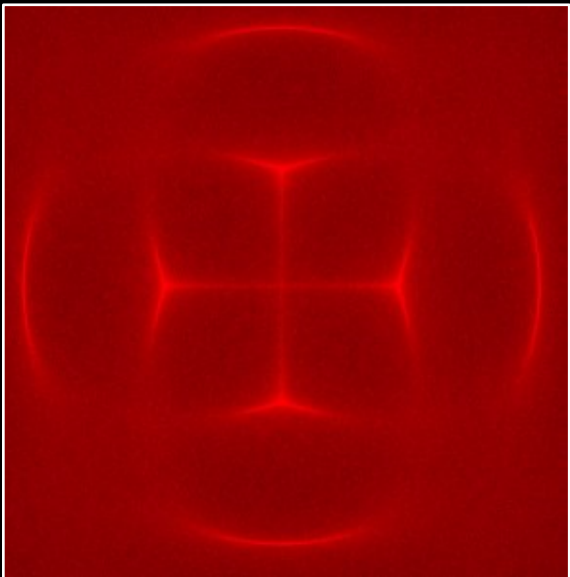
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



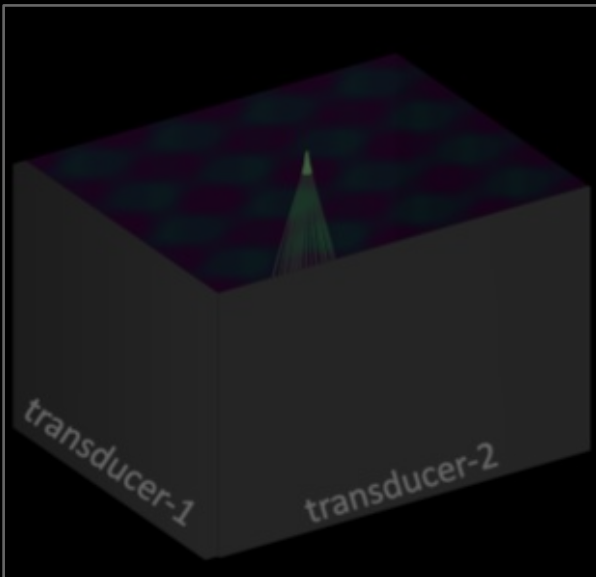
time-of-flight imaging



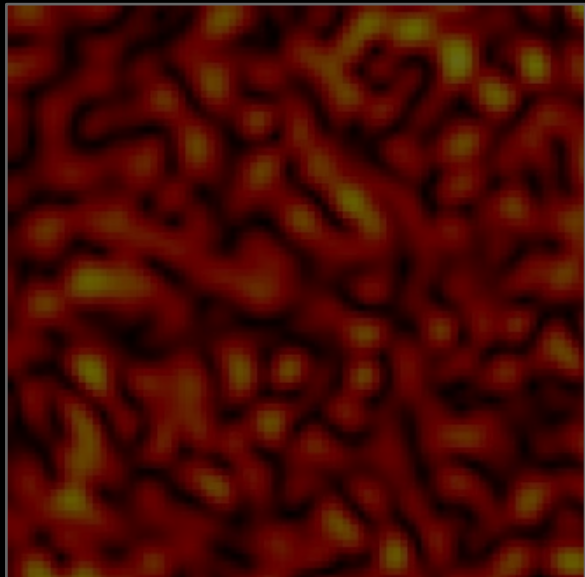
non-line-of-sight imaging



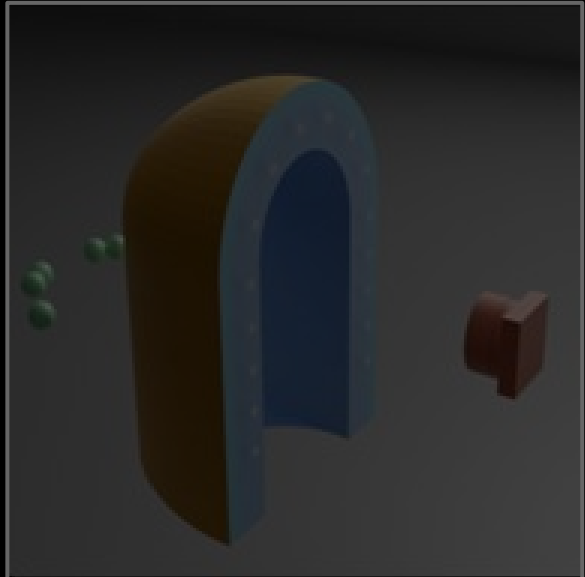
acousto-optic lensing



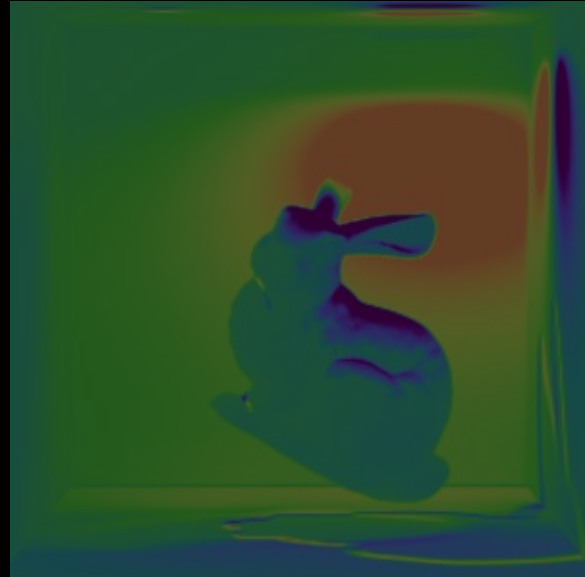
ultrafast light scanning



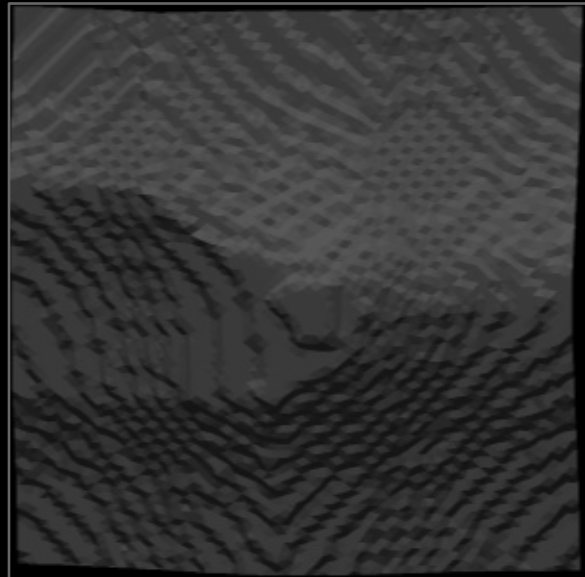
speckle imaging



tactile sensor design

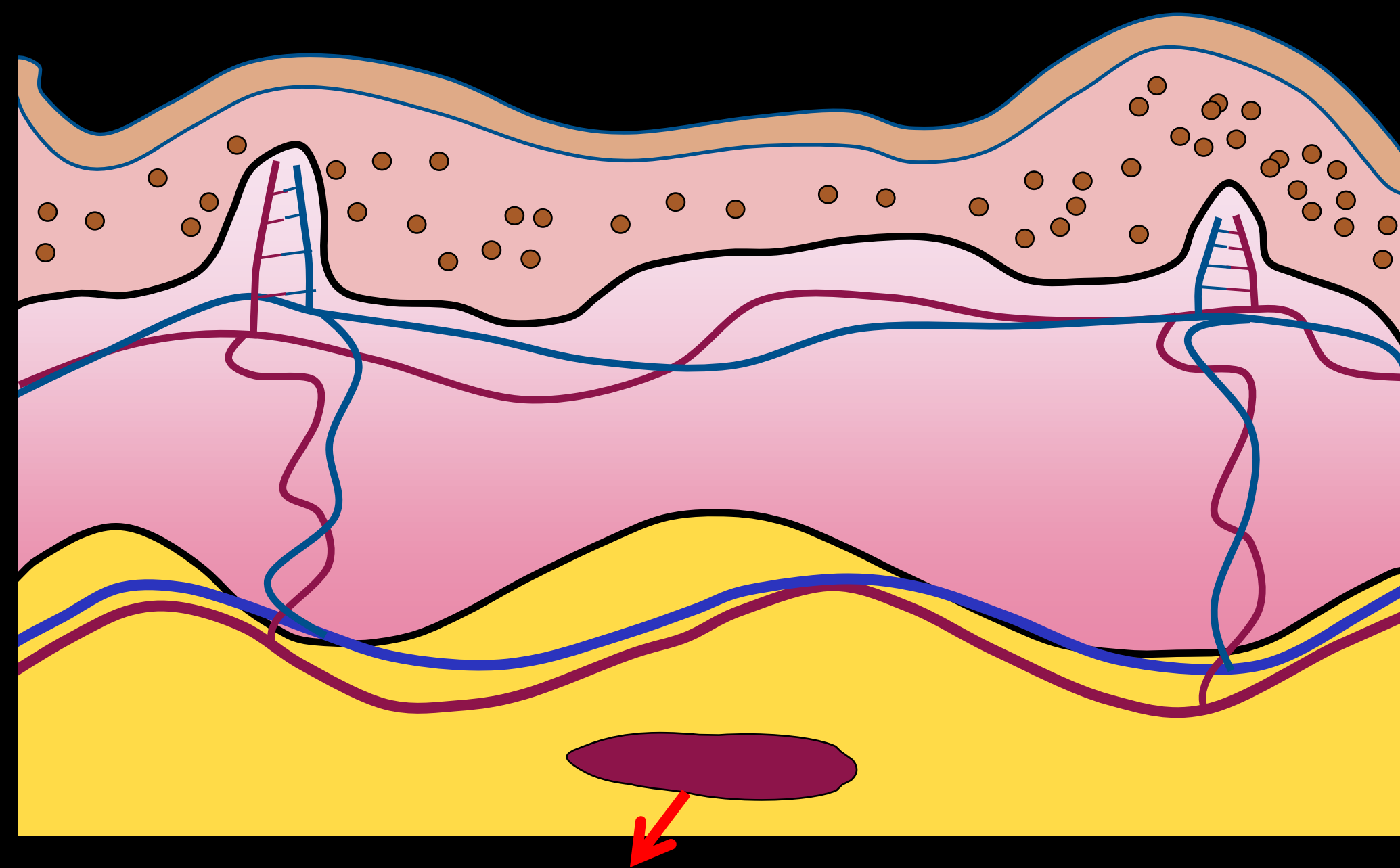


differentiable renderer



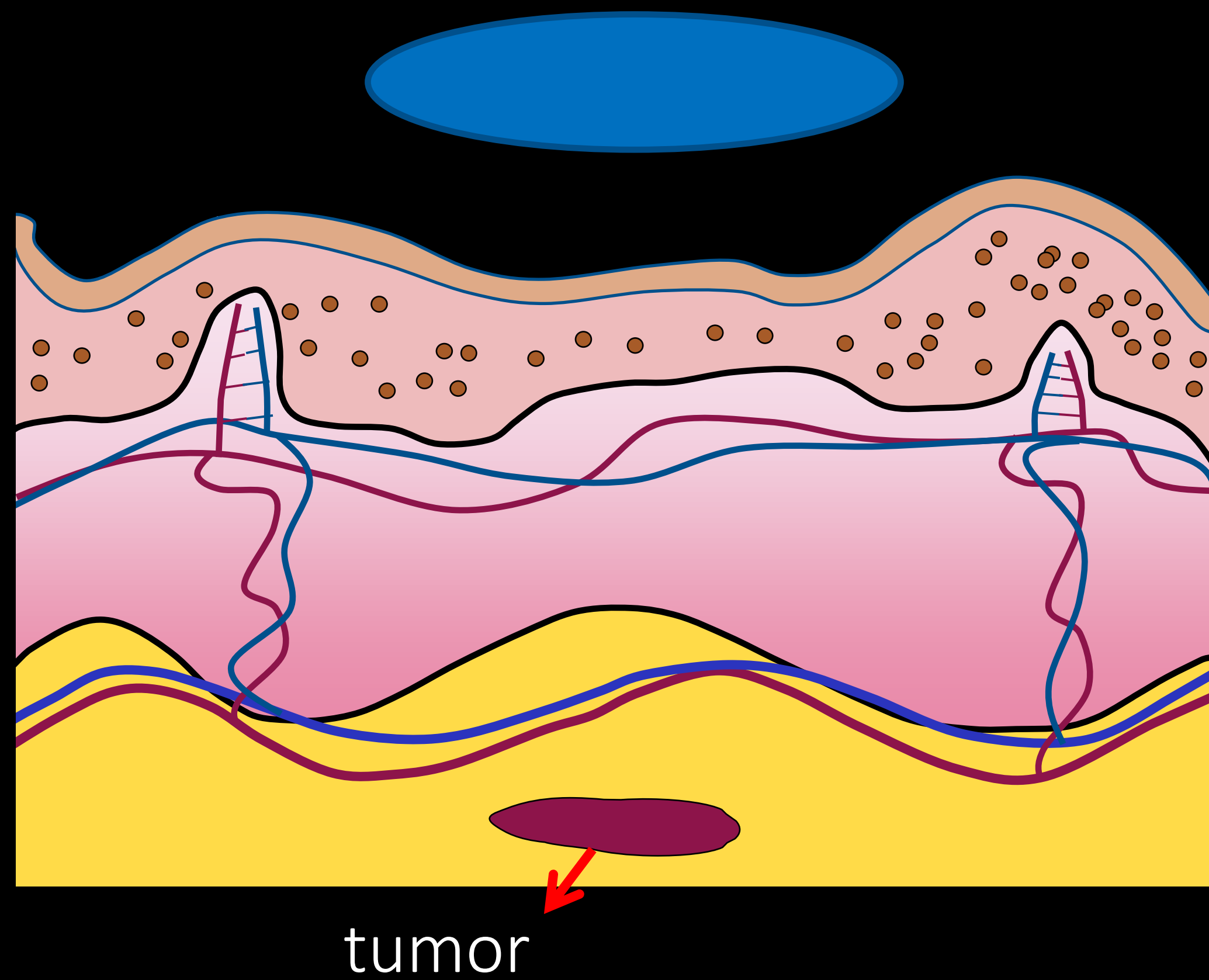
inverse problems

Focusing light inside tissue

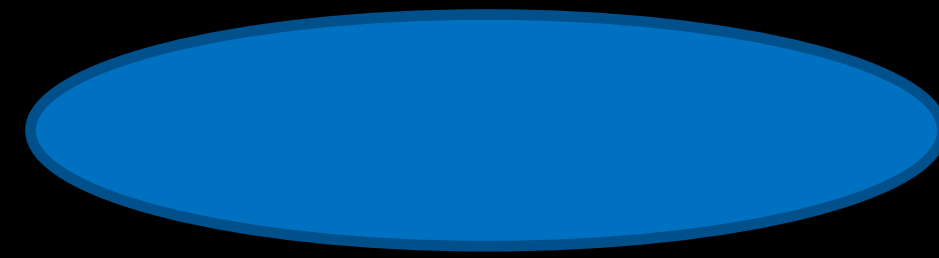


tumor

Focusing light inside tissue

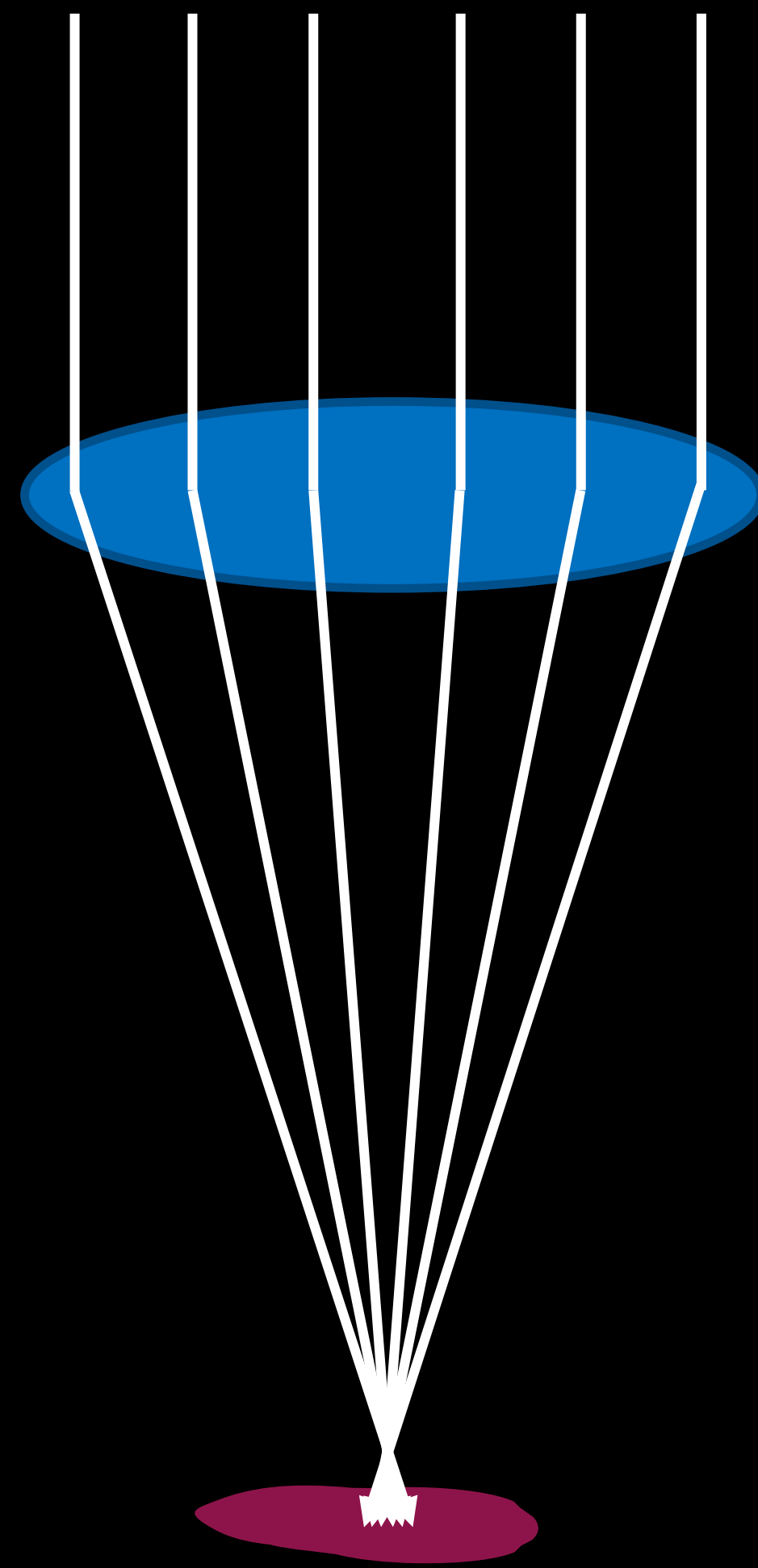


Focusing light inside tissue



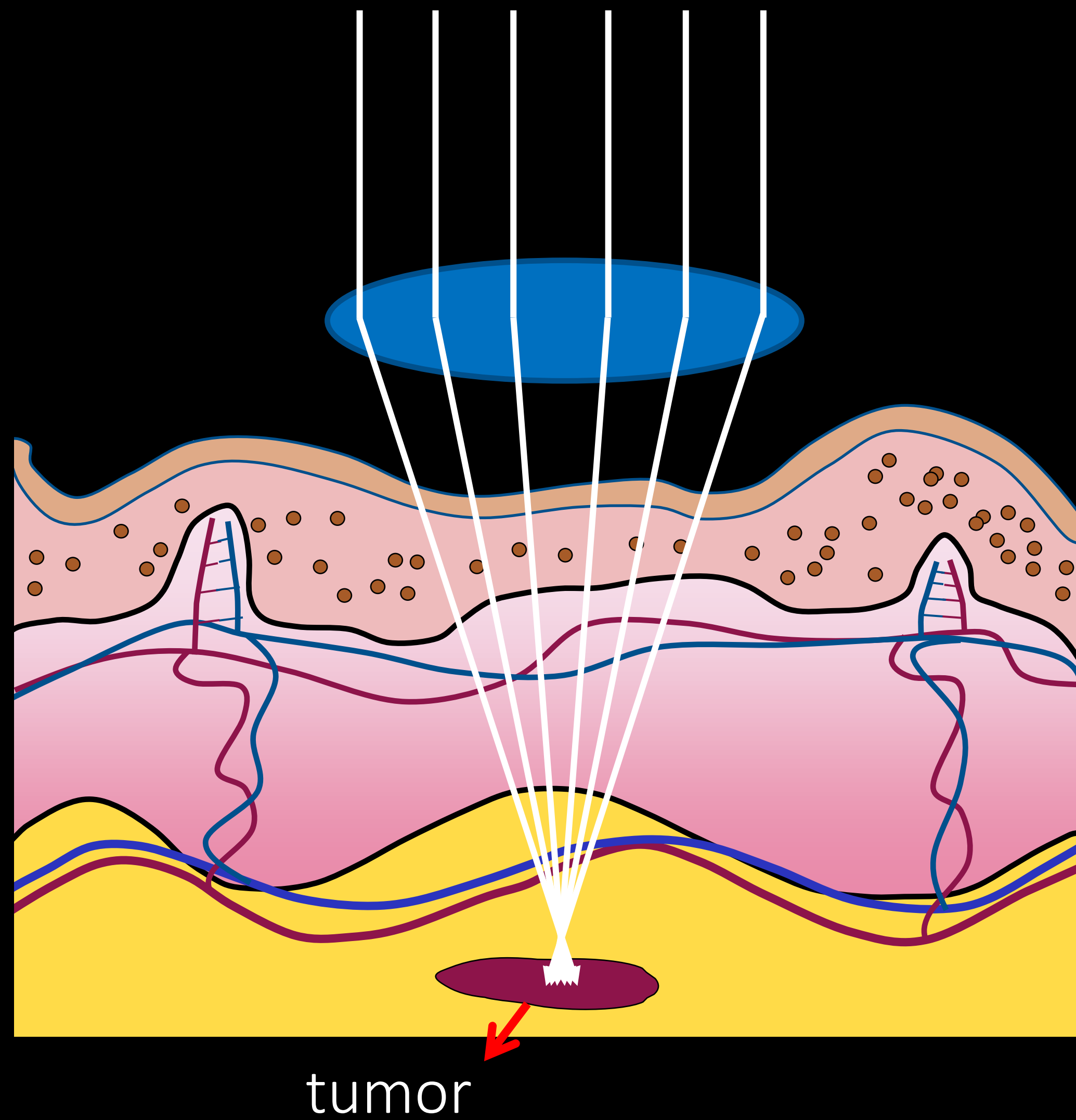
tumor

Focusing light inside tissue

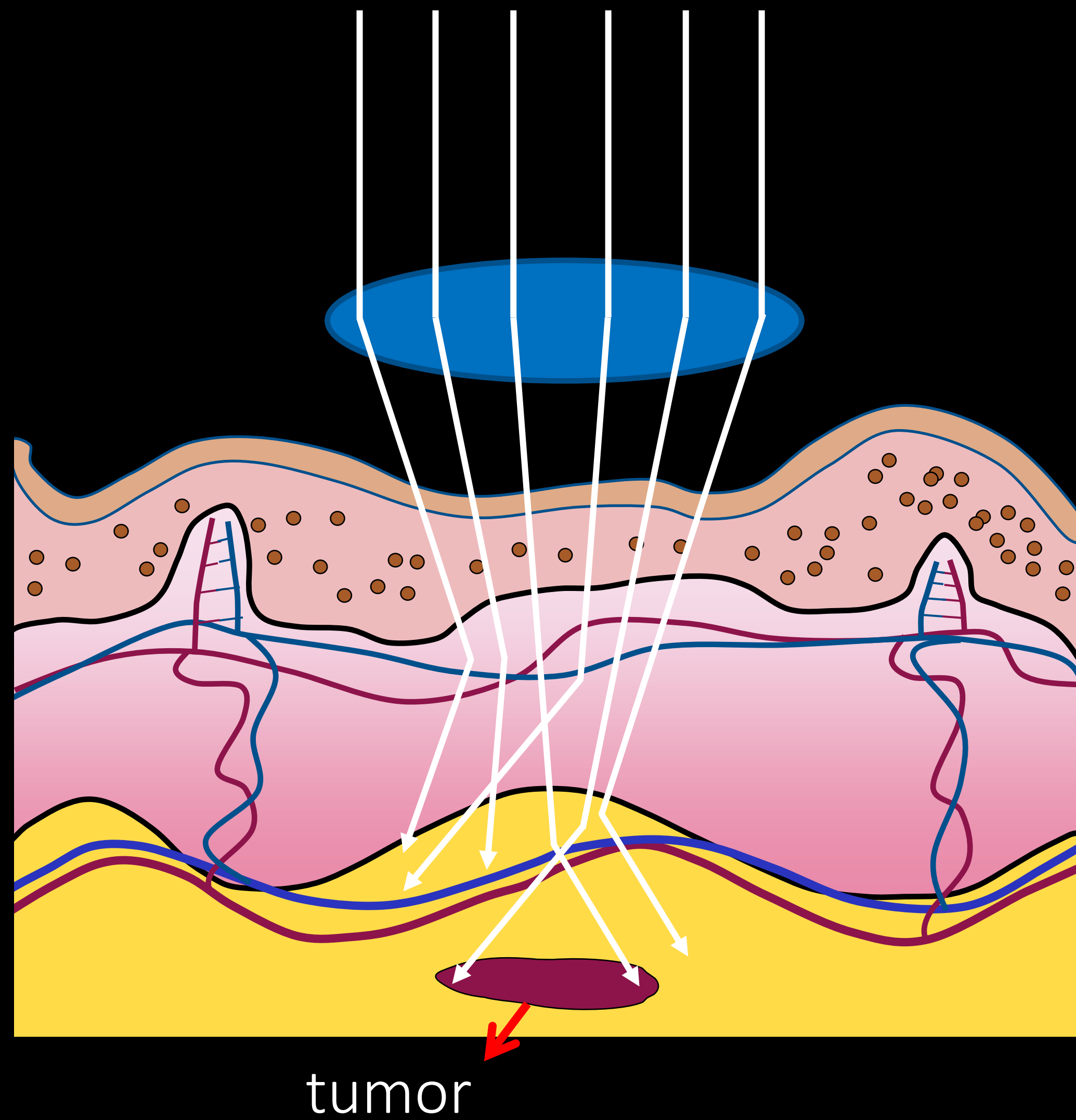


tumor

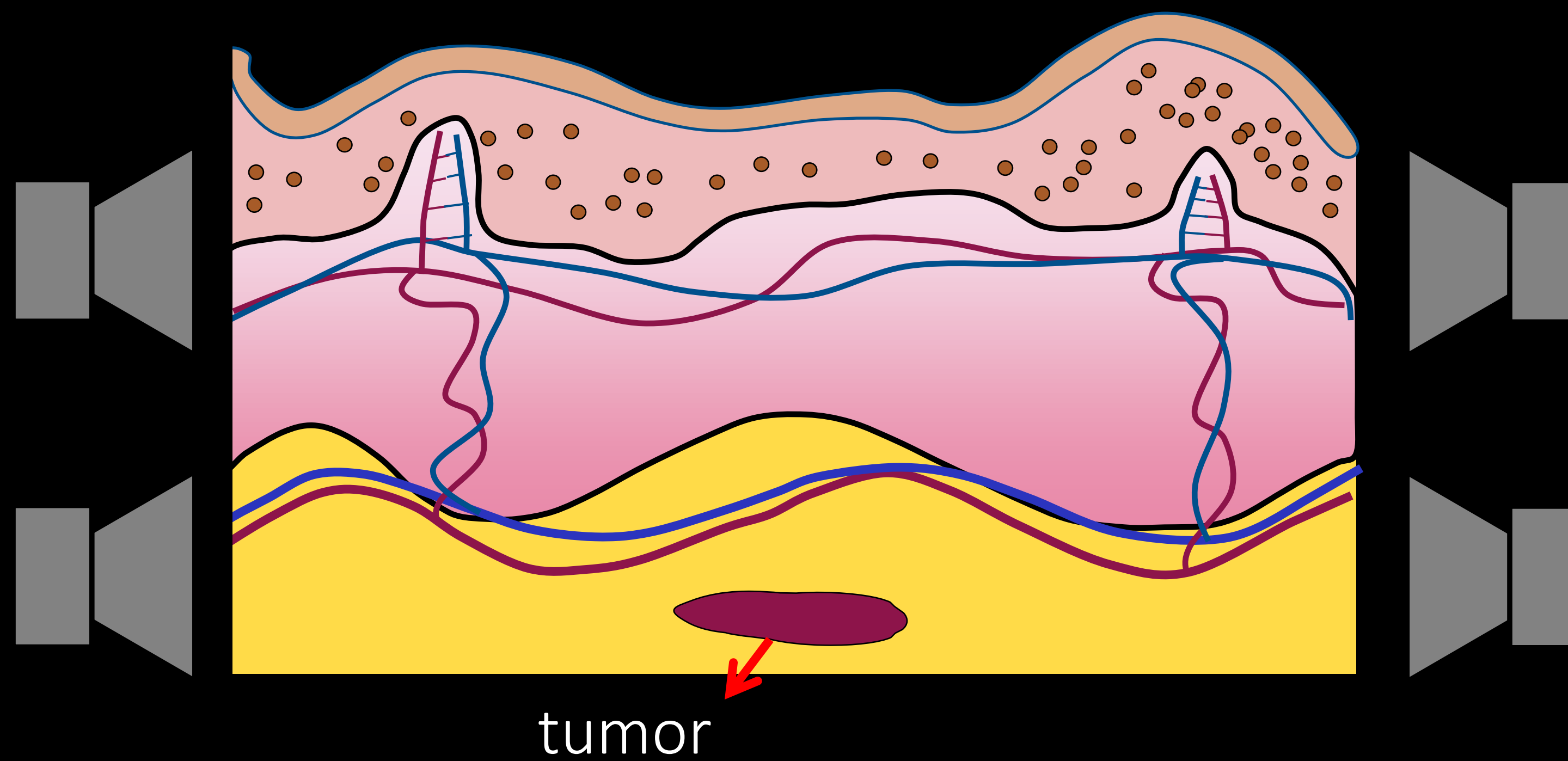
Focusing light inside tissue



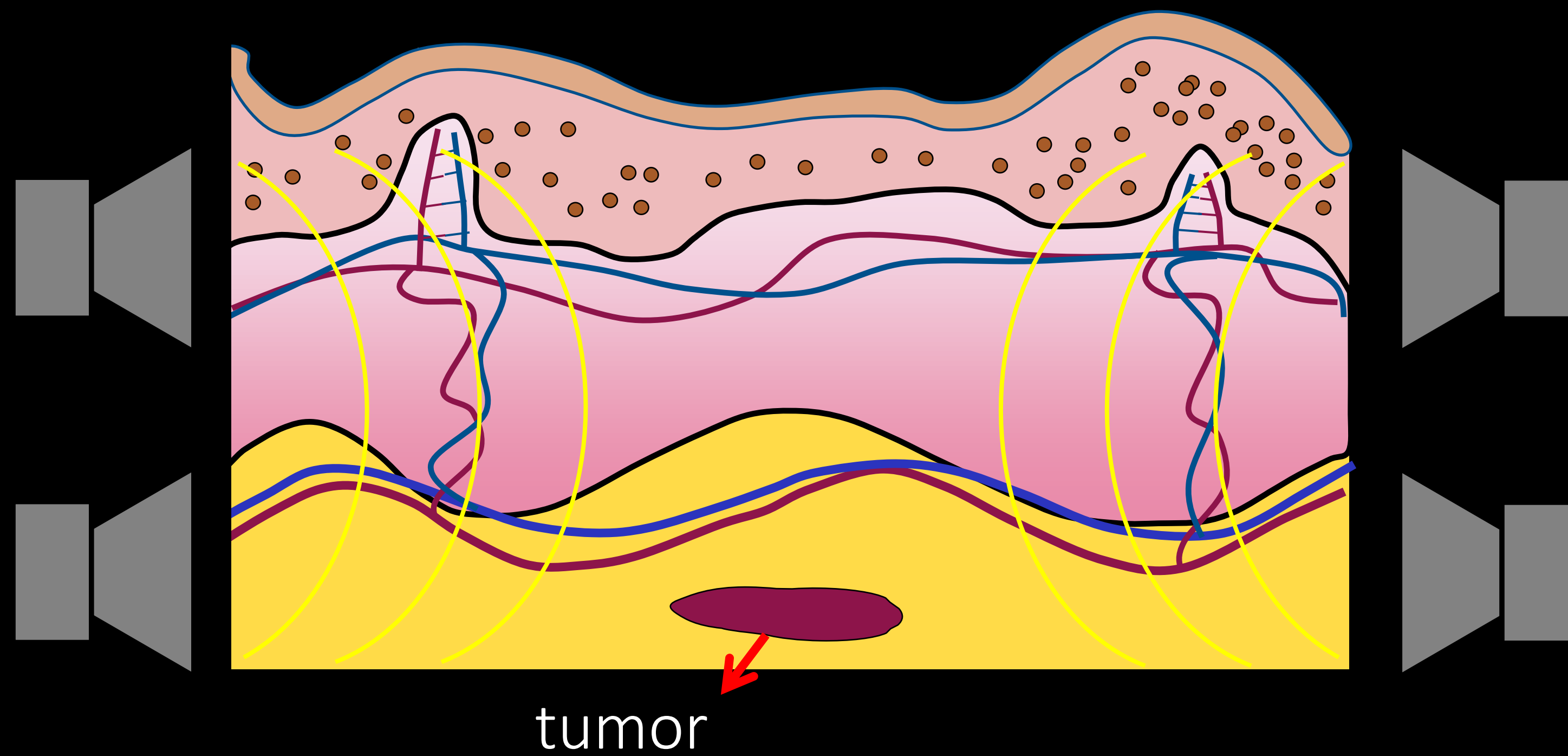
Focusing light inside tissue



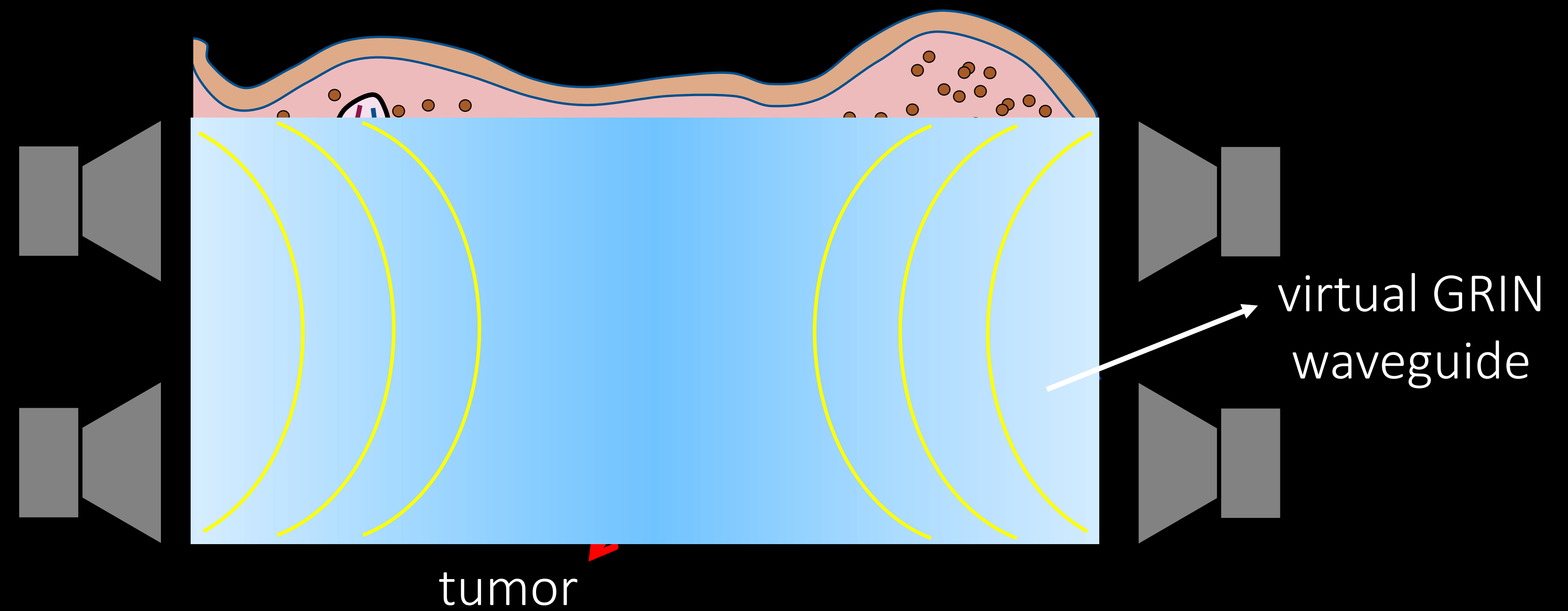
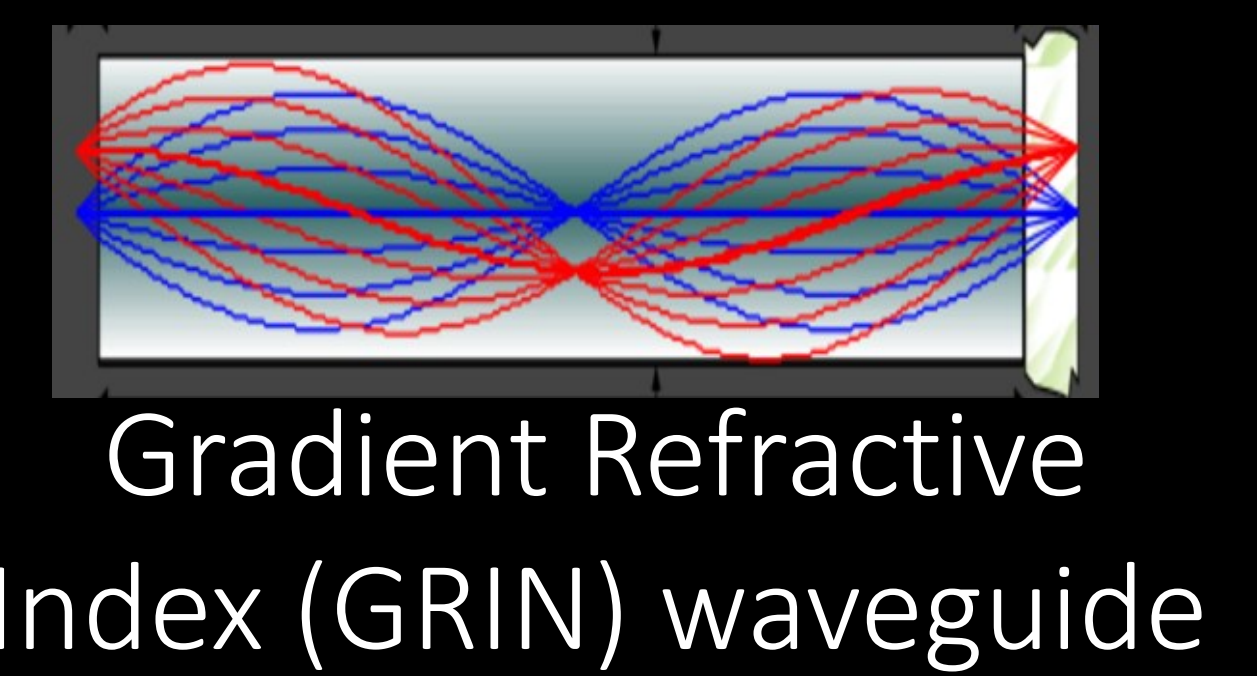
Focusing light inside tissue



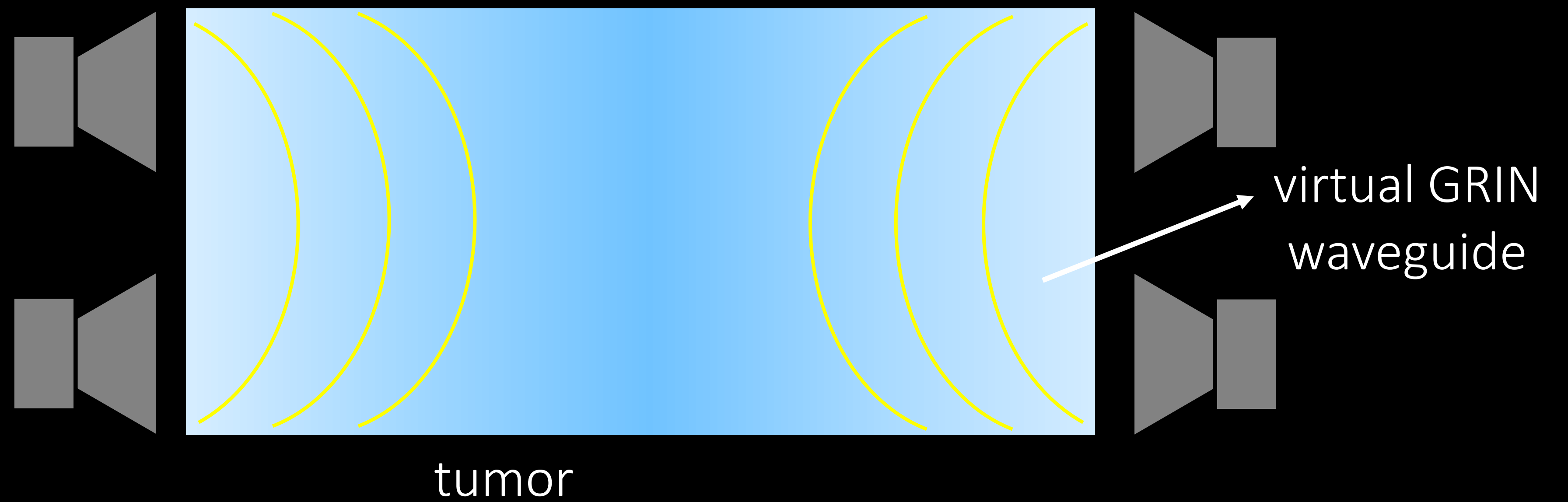
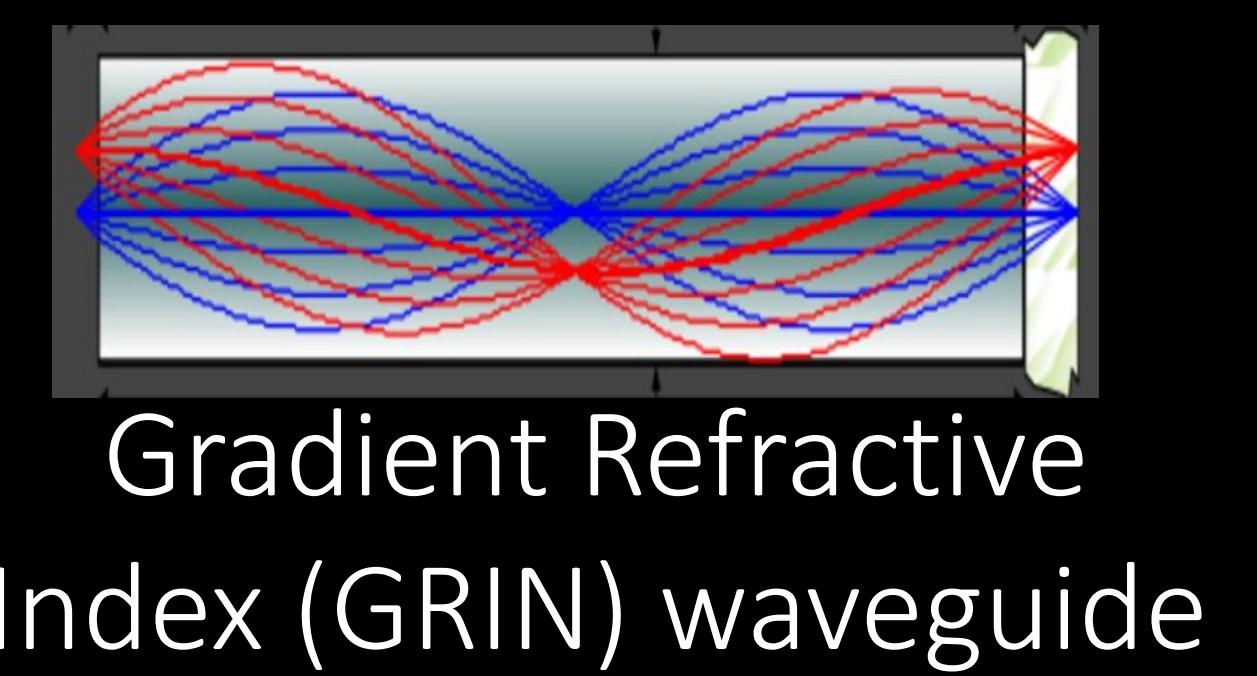
Focusing light inside tissue



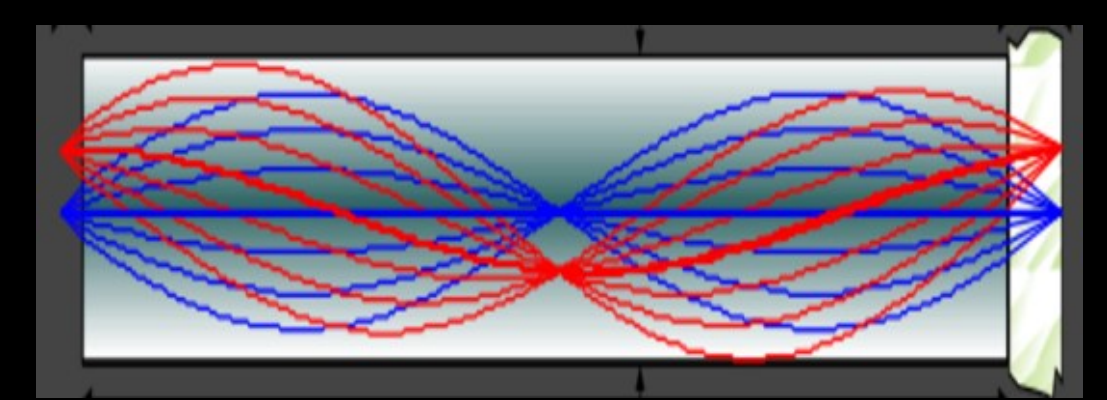
Focusing light inside tissue



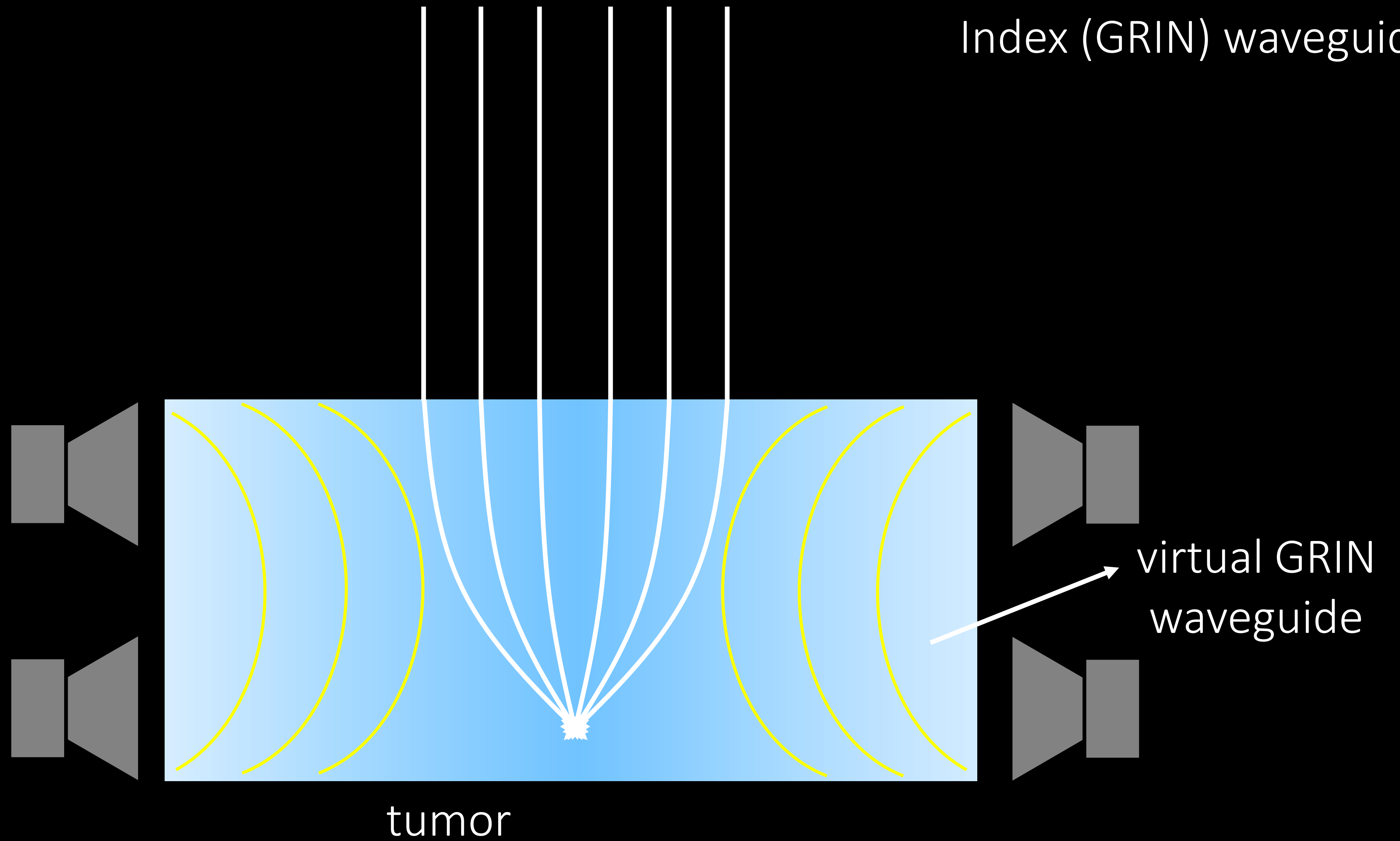
Focusing light inside tissue



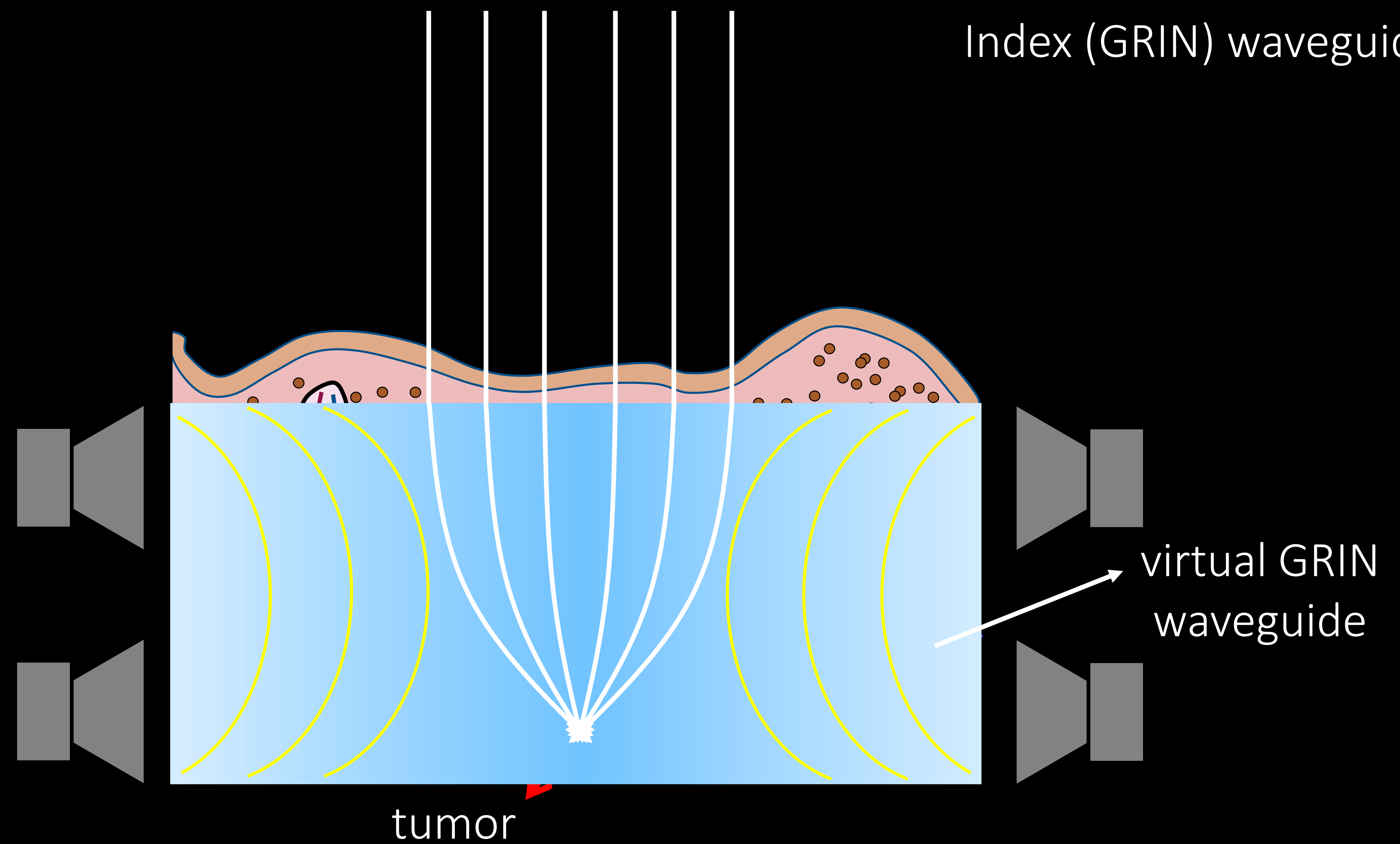
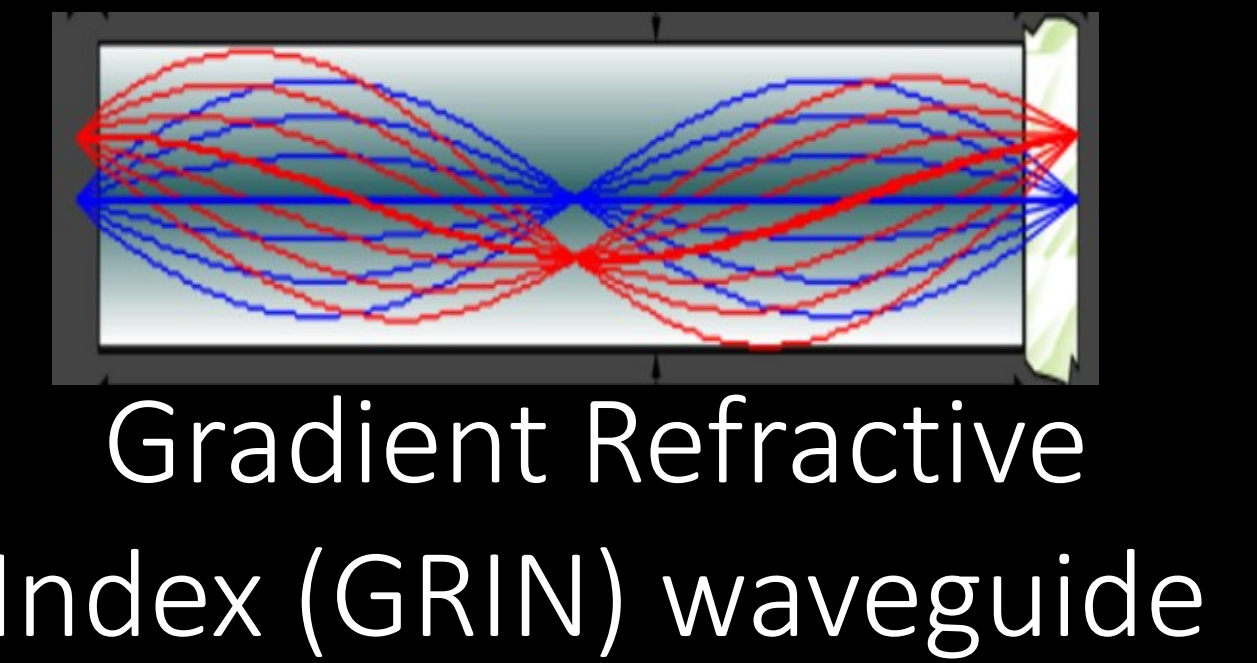
Focusing light inside tissue



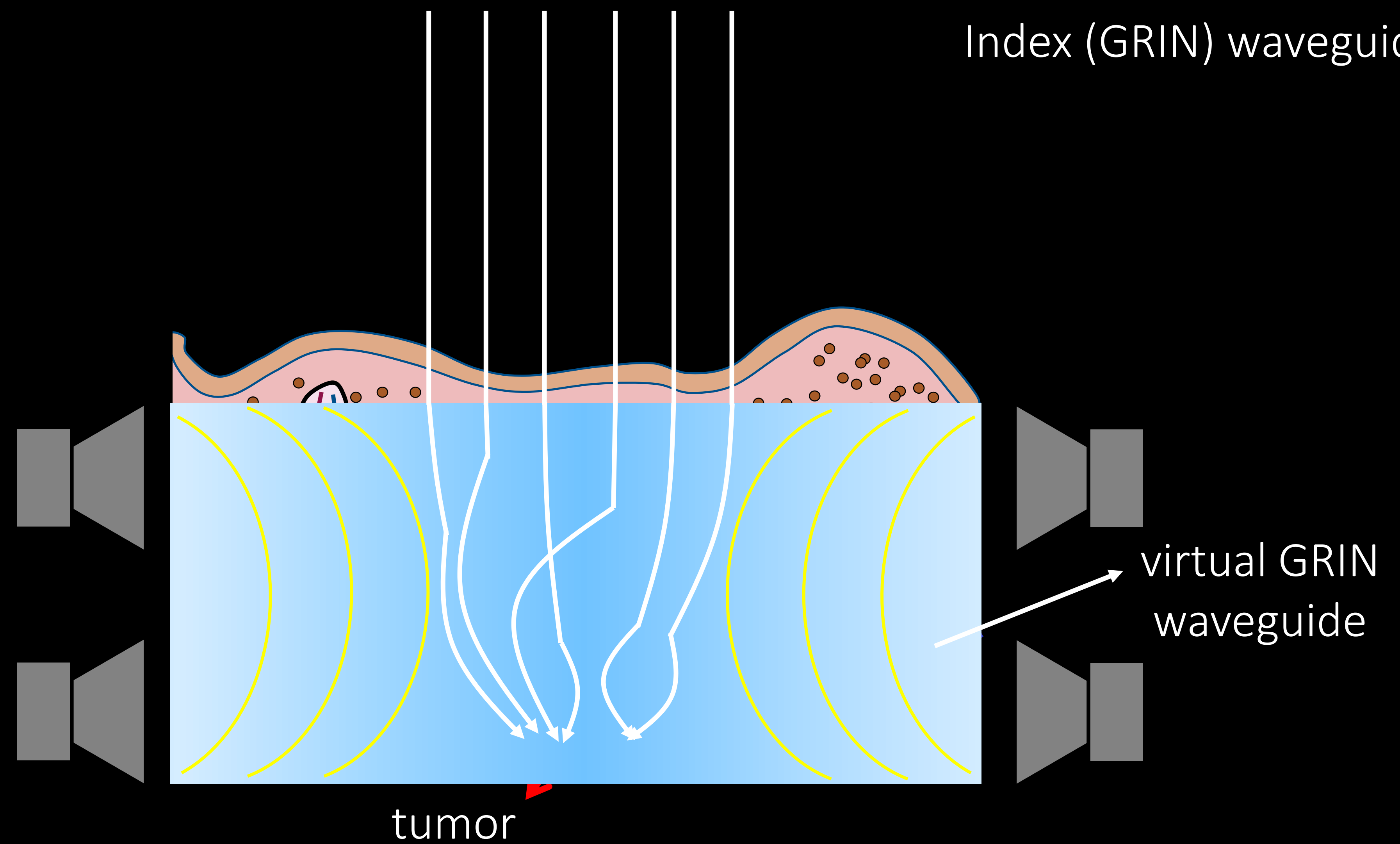
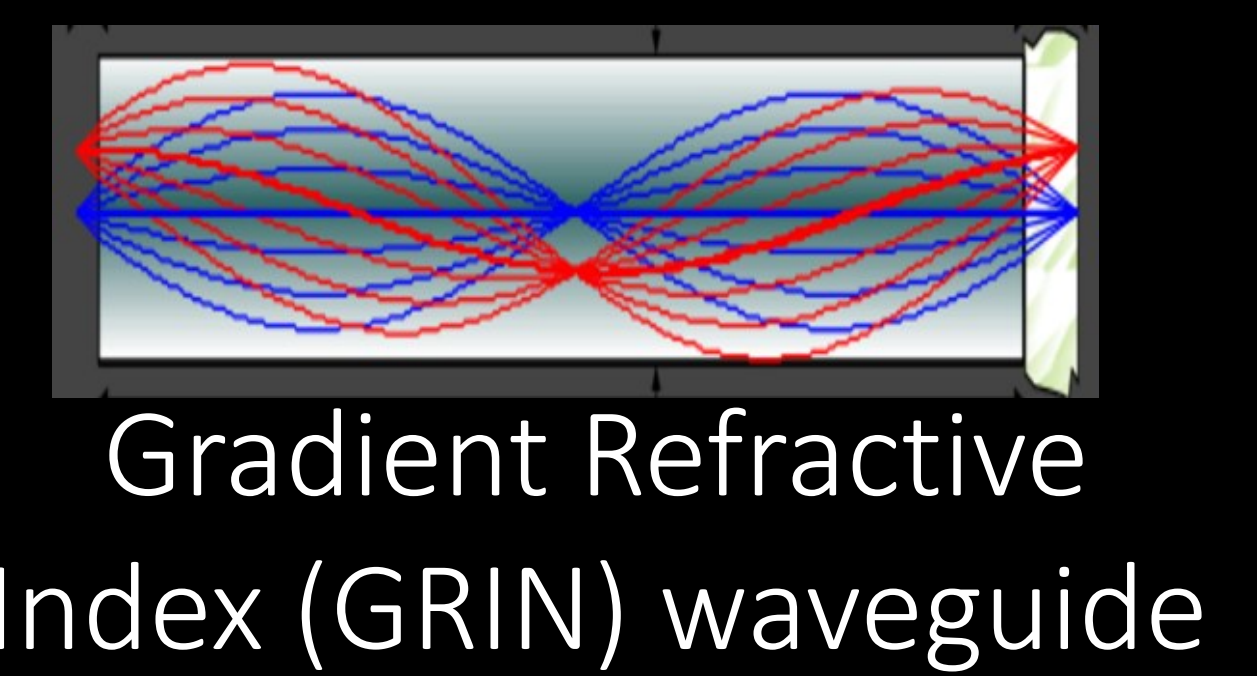
Gradient Refractive Index (GRIN) waveguide



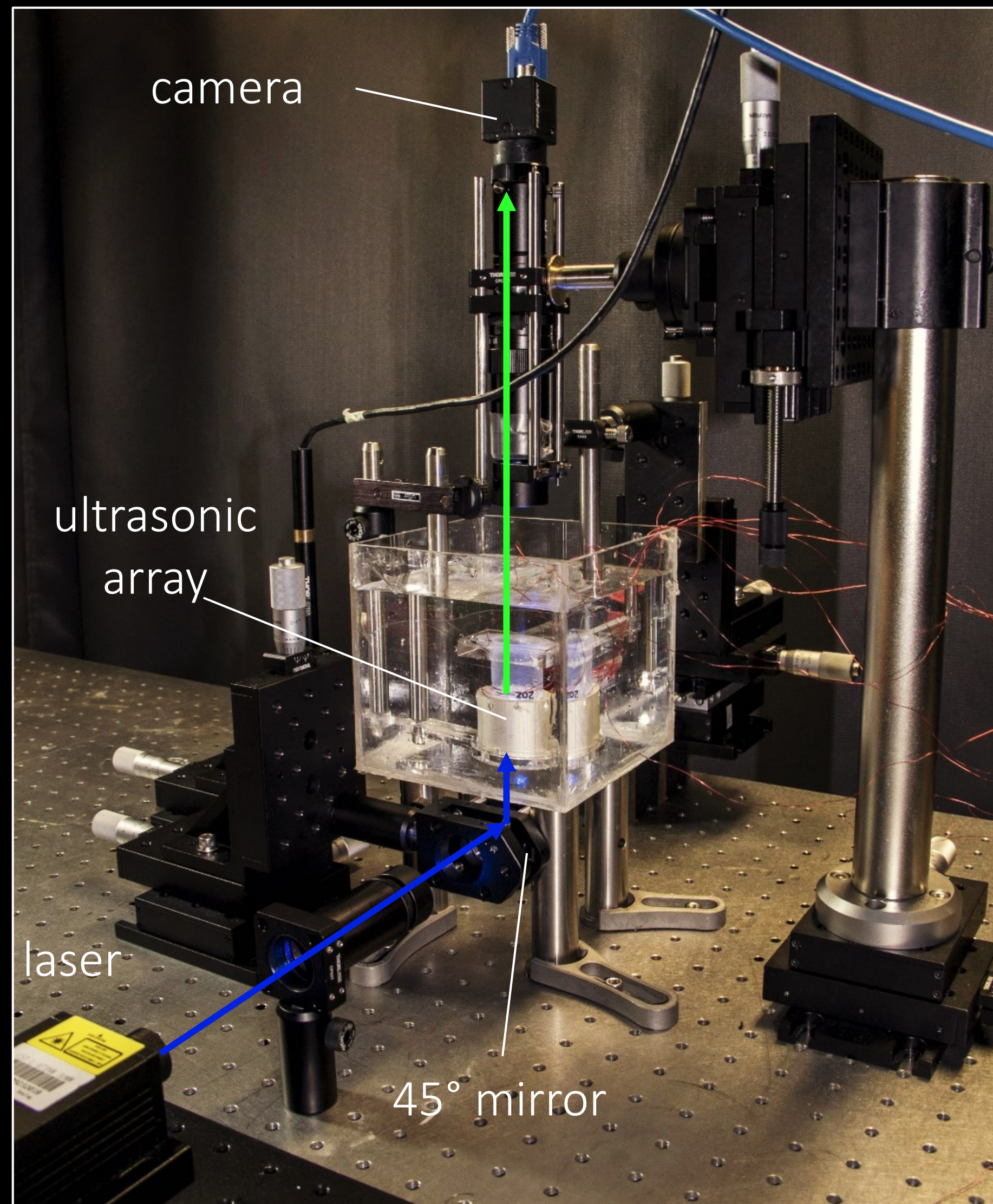
Focusing light inside tissue



Focusing light inside tissue

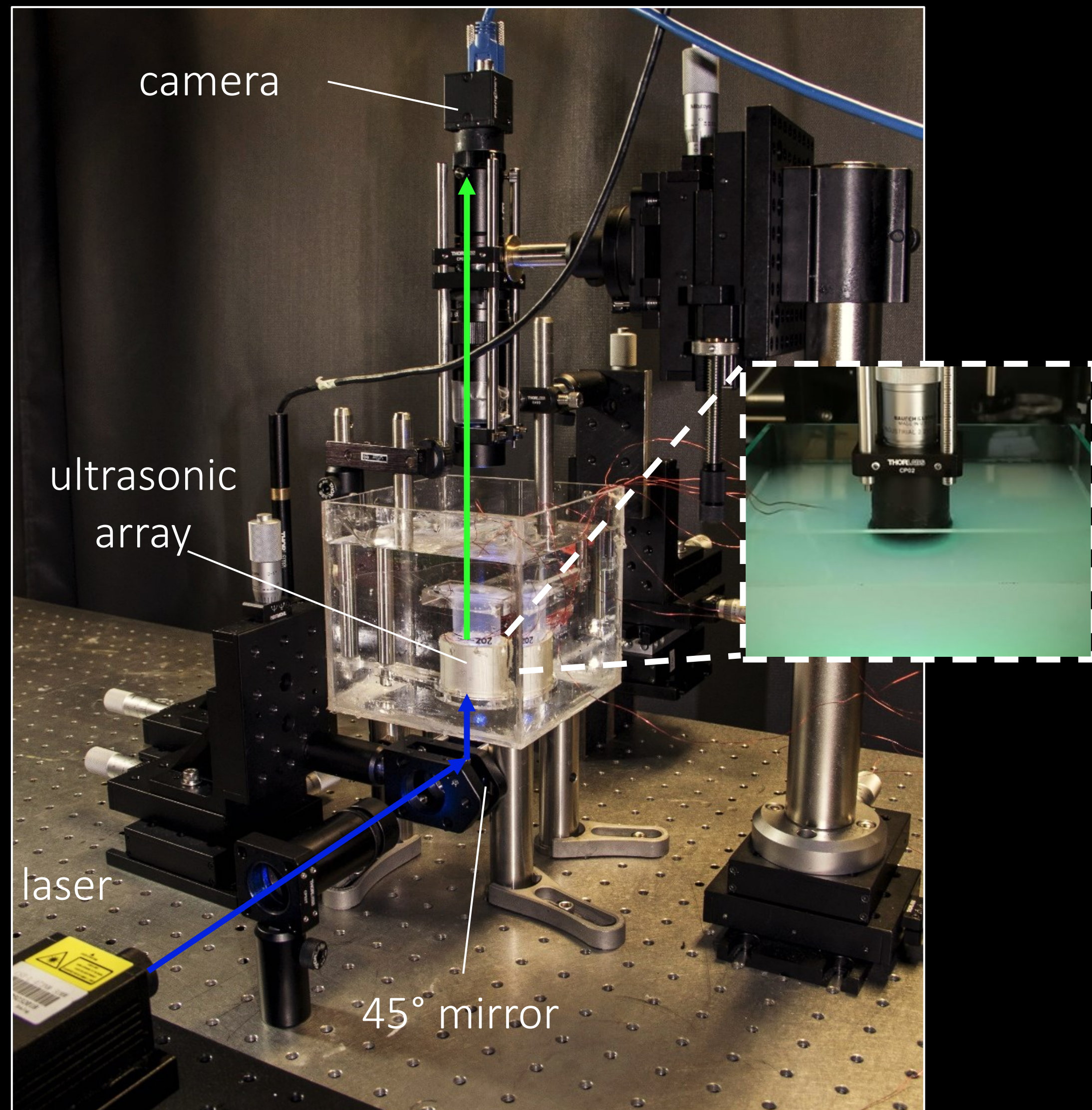


Ultrasonic light guiding inside tissue



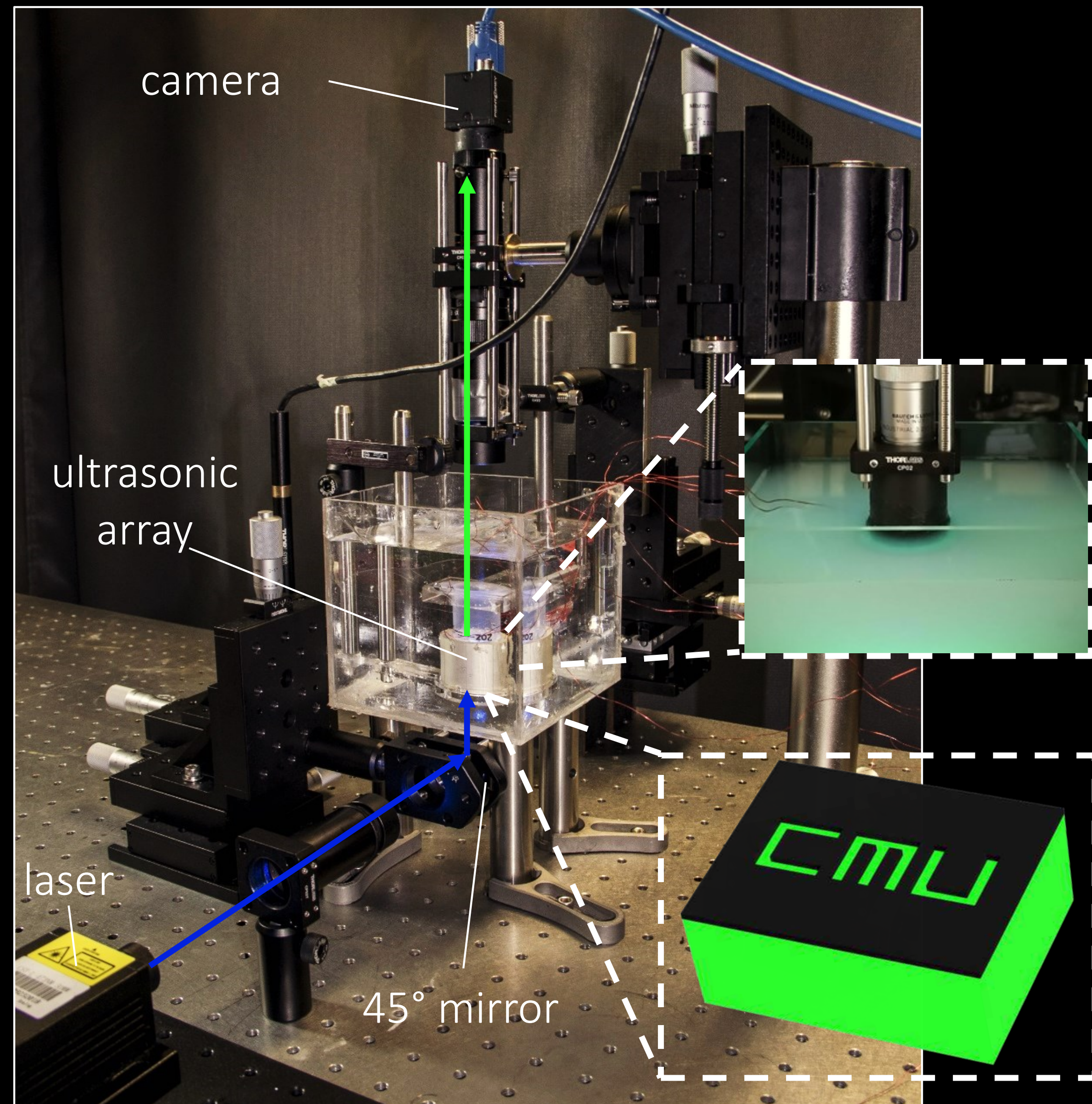
[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]

Ultrasonic light guiding inside tissue



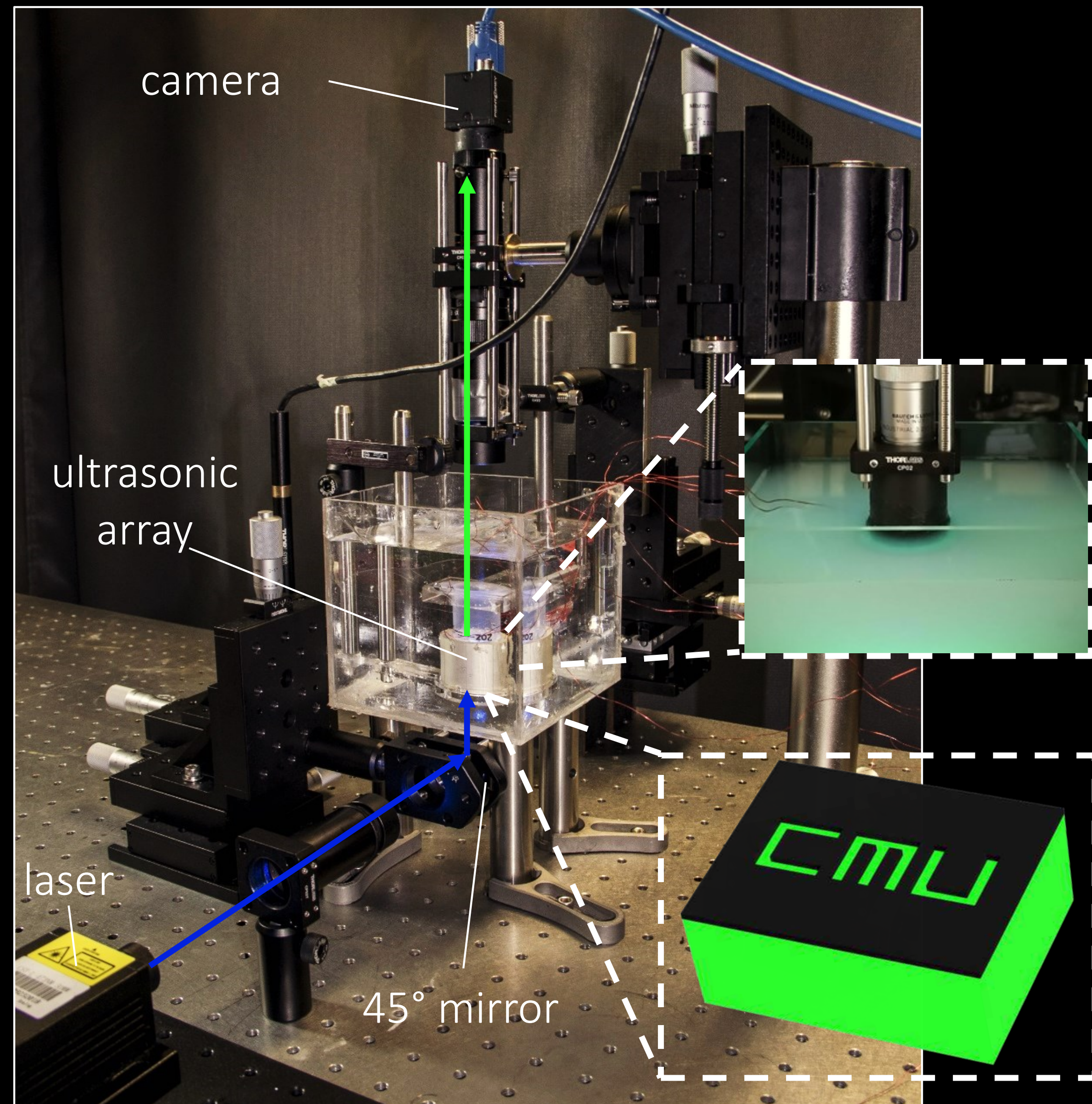
[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]

Ultrasonic light guiding inside tissue



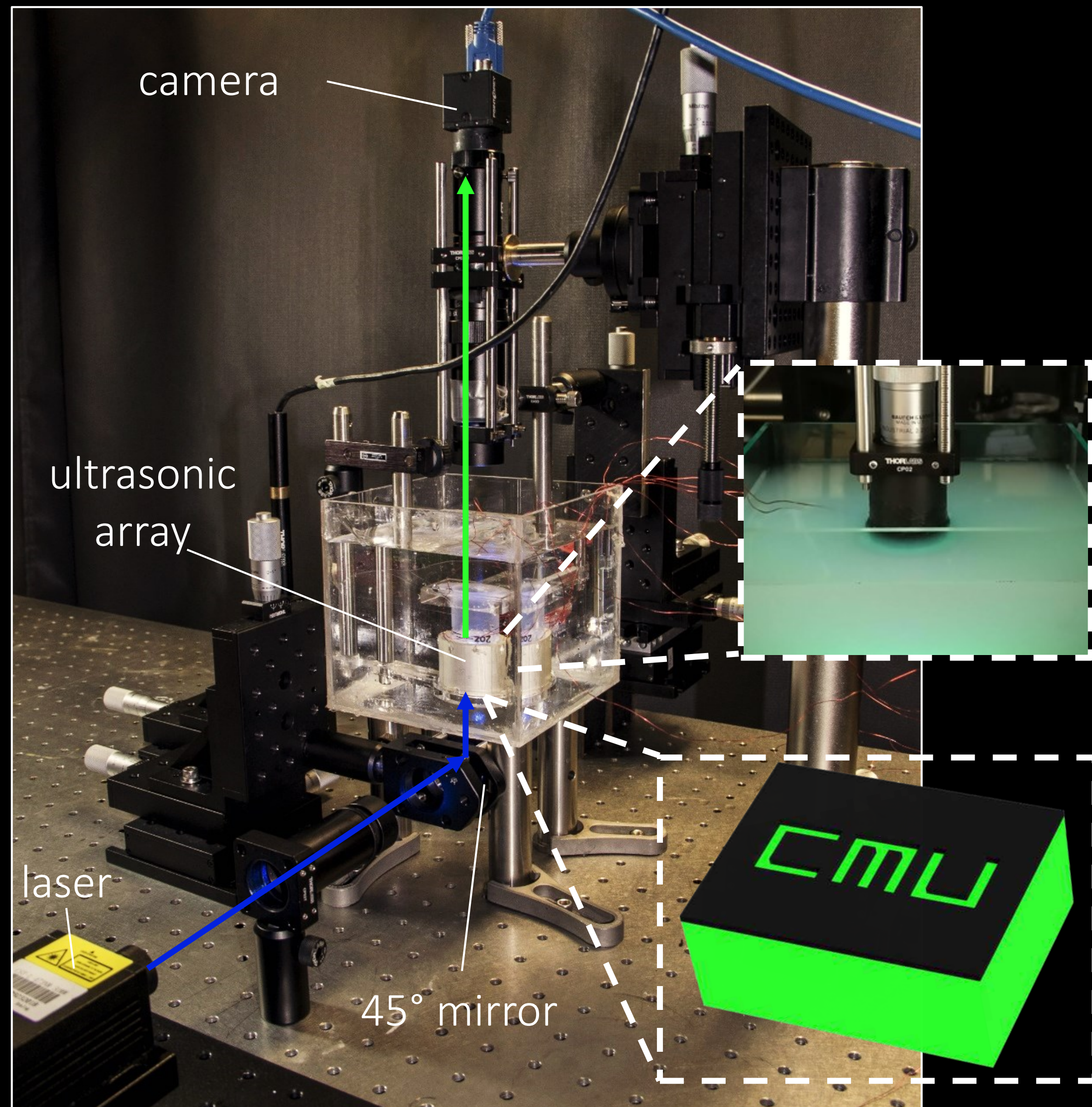
[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]

Ultrasonic light guiding inside tissue



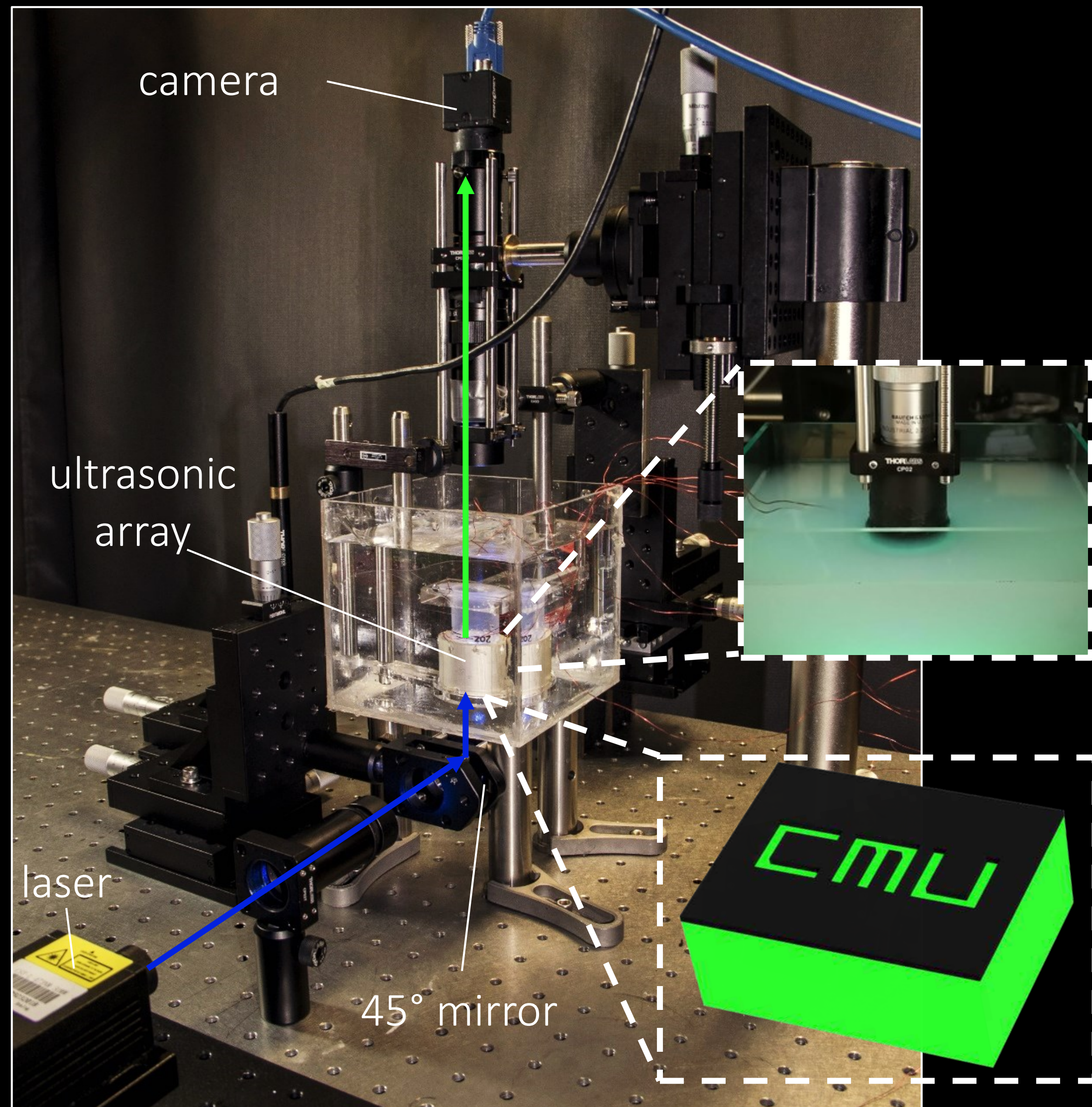
[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]

Ultrasonic light guiding inside tissue



[Chamanzar et al., Nat. Comm. 2019]
[Karimi et al., Optics Express, 2019]
[Scopelliti et al., LSA, 2019]

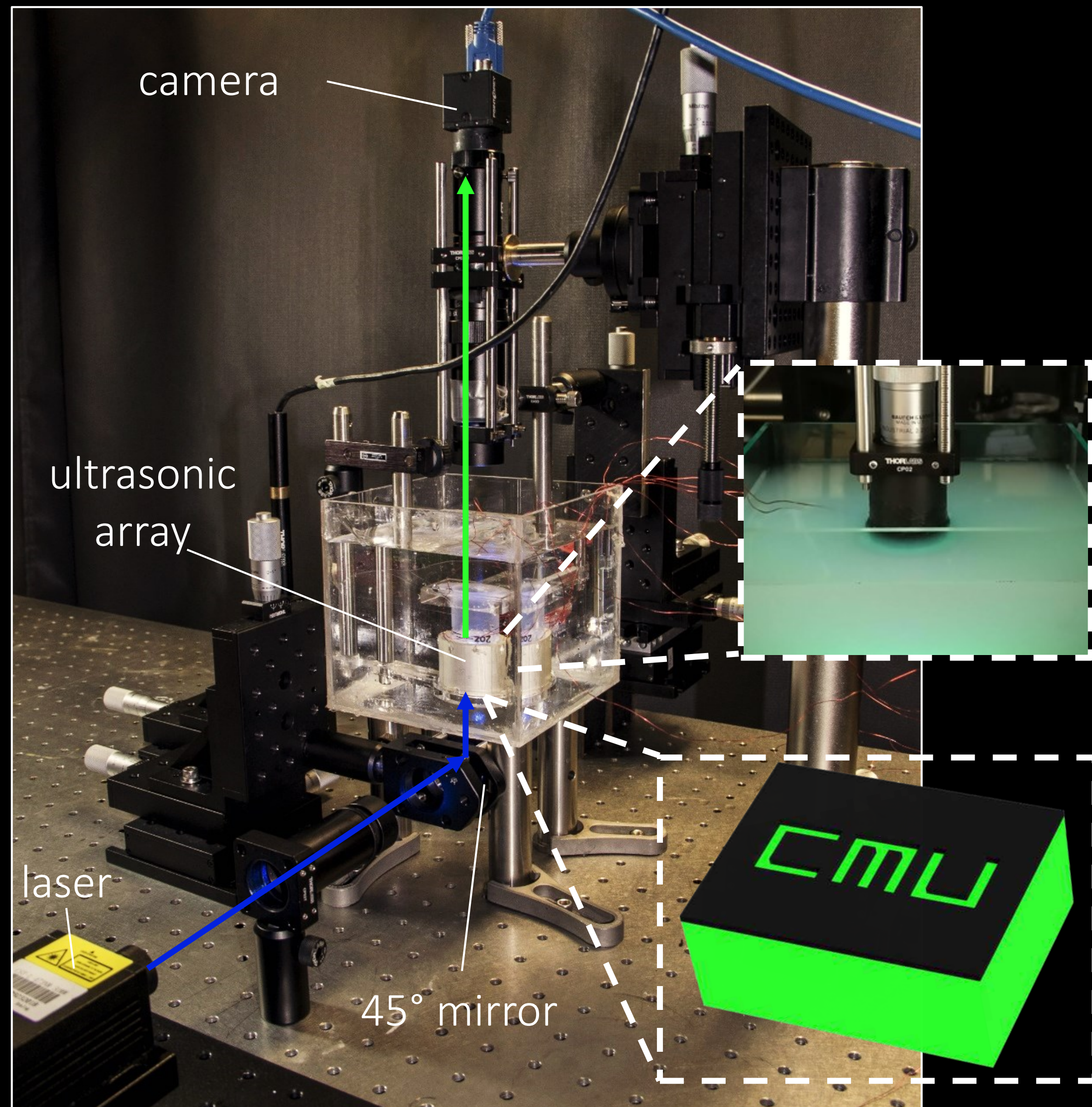
Ultrasonic light guiding inside tissue



High-dimensional, highly-non-linear design problem:

- ultrasound frequency
- ultrasound voltage
- placement of transducers
- shape of waveguides
- waveform shape
- and more...

Ultrasonic light guiding inside tissue

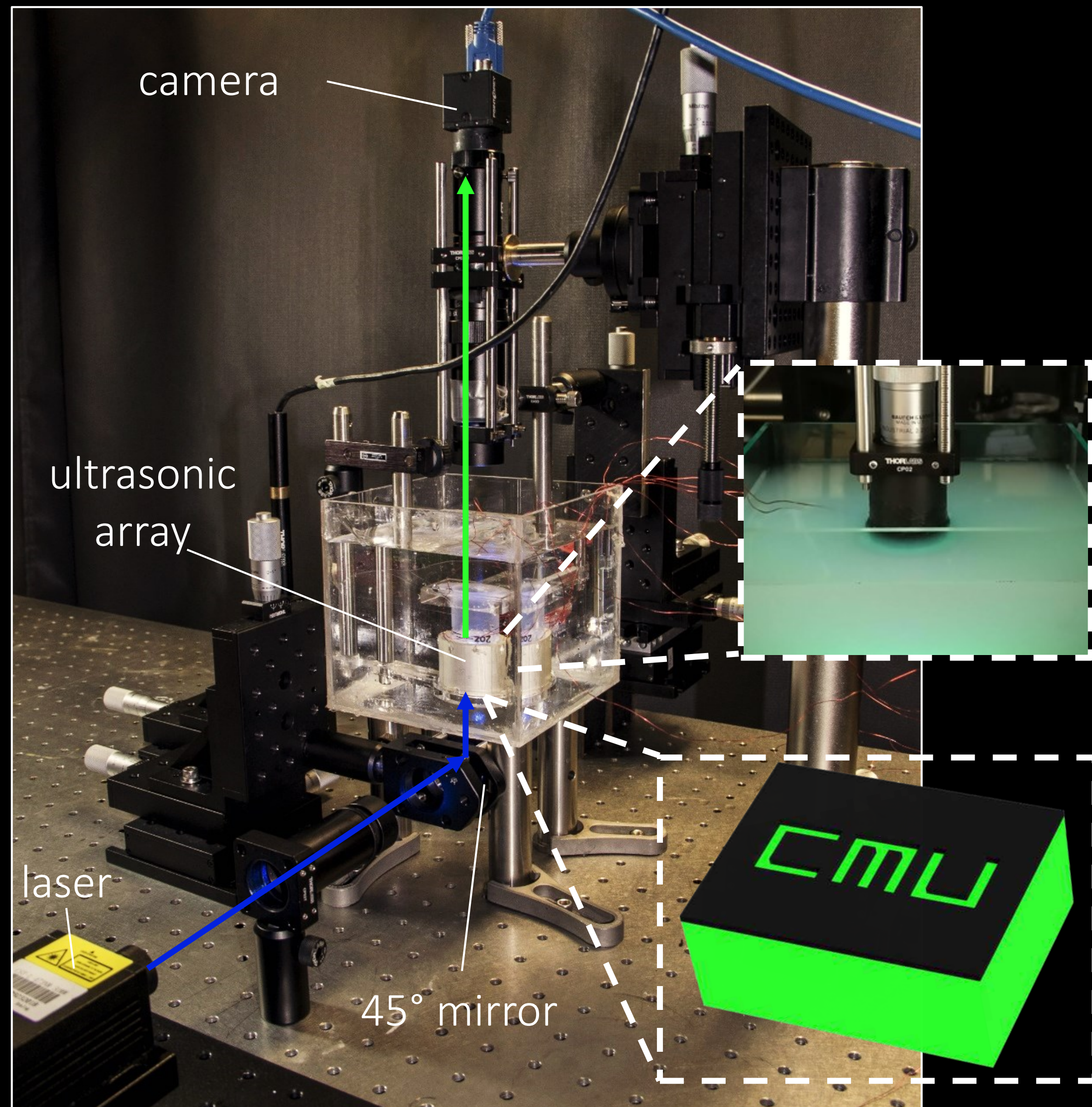


High-dimensional, highly-non-linear design problem:

- ultrasound frequency
- ultrasound voltage
- placement of transducers
- shape of waveguides
- waveform shape
- and more...

Efficiently explore using rendering

Ultrasonic light guiding inside tissue



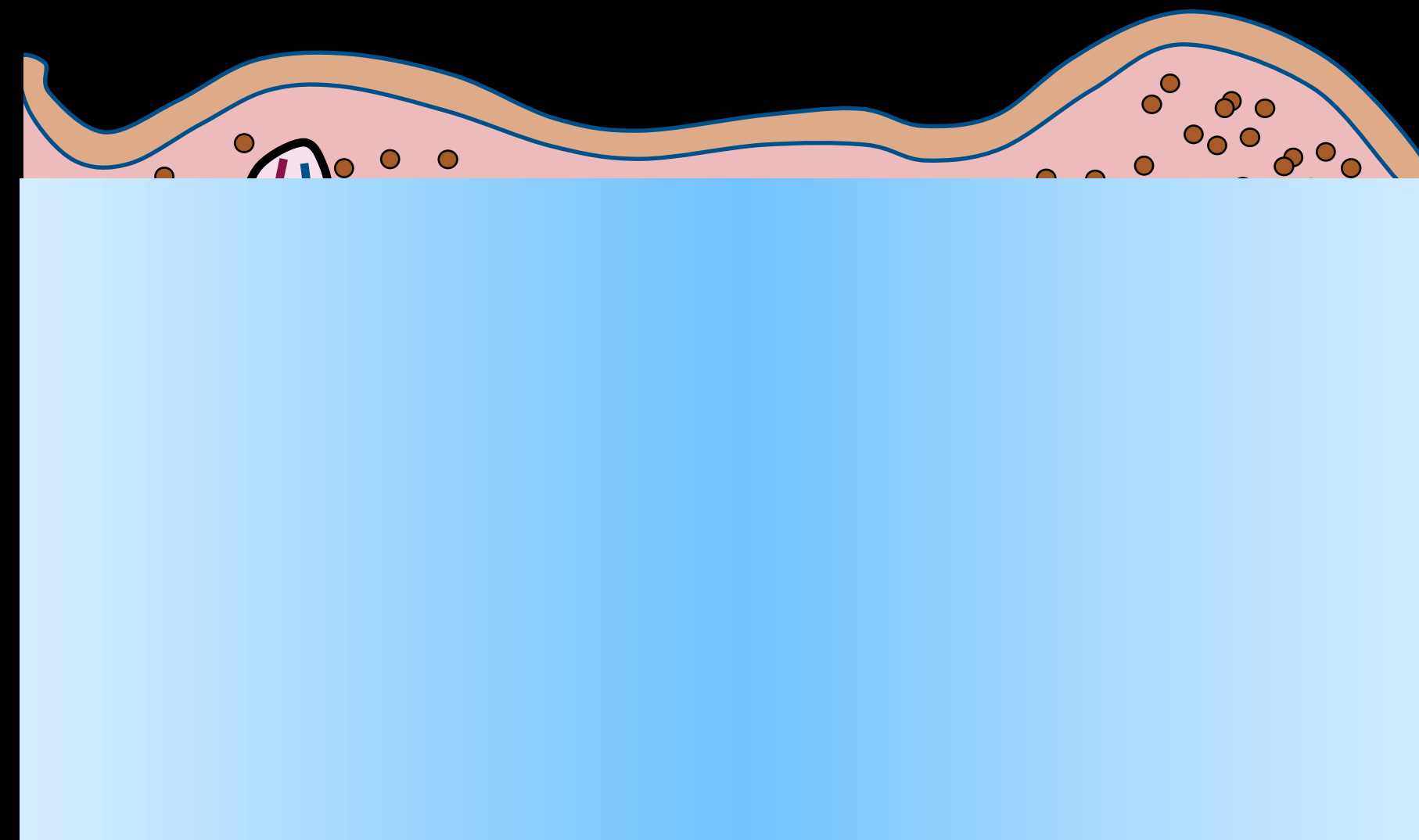
High-dimensional, highly-non-linear design problem:

- ultrasound frequency
- ultrasound voltage
- placement of transducers
- shape of waveguides
- waveform shape
- and more...

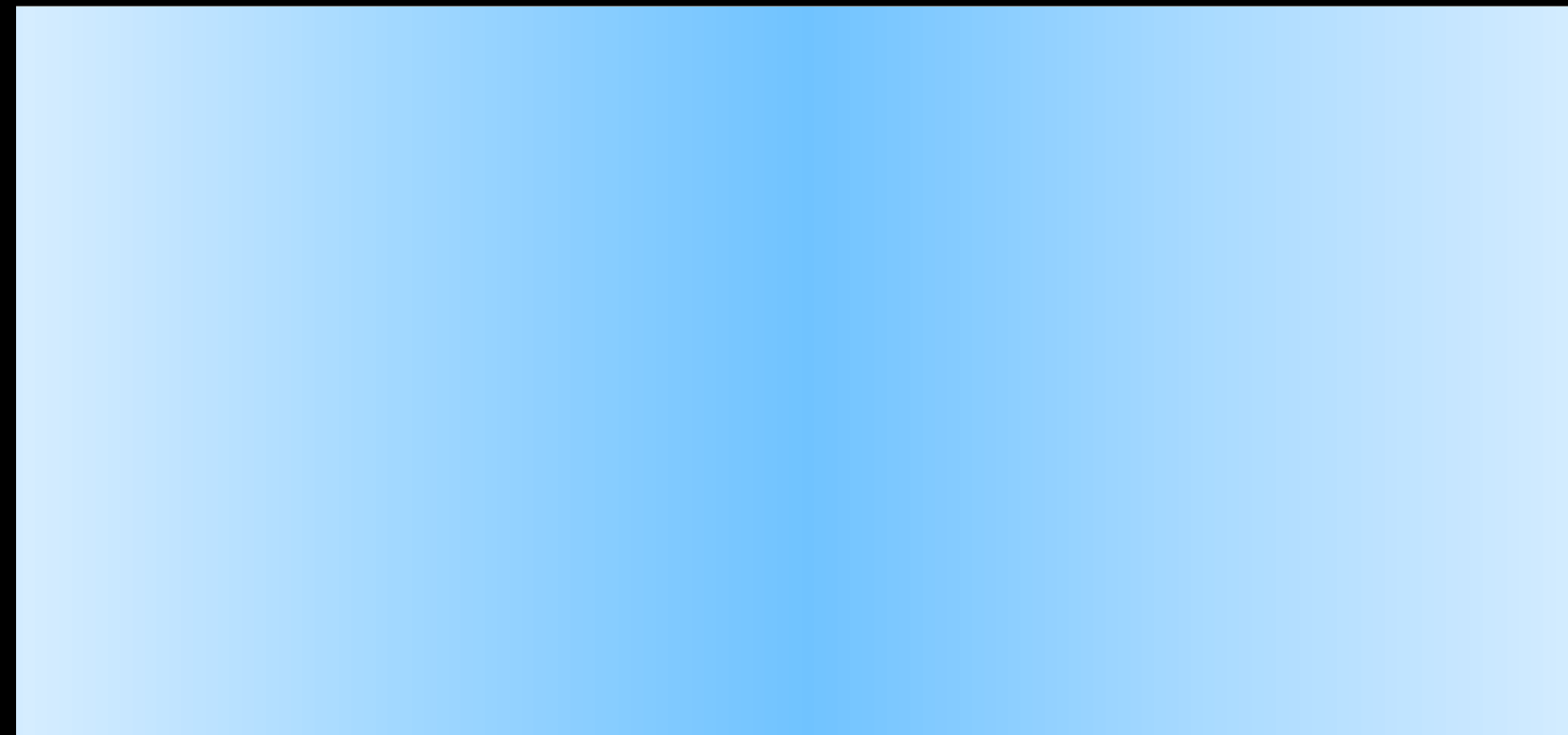
Efficiently explore using rendering

Build first rendering algorithm

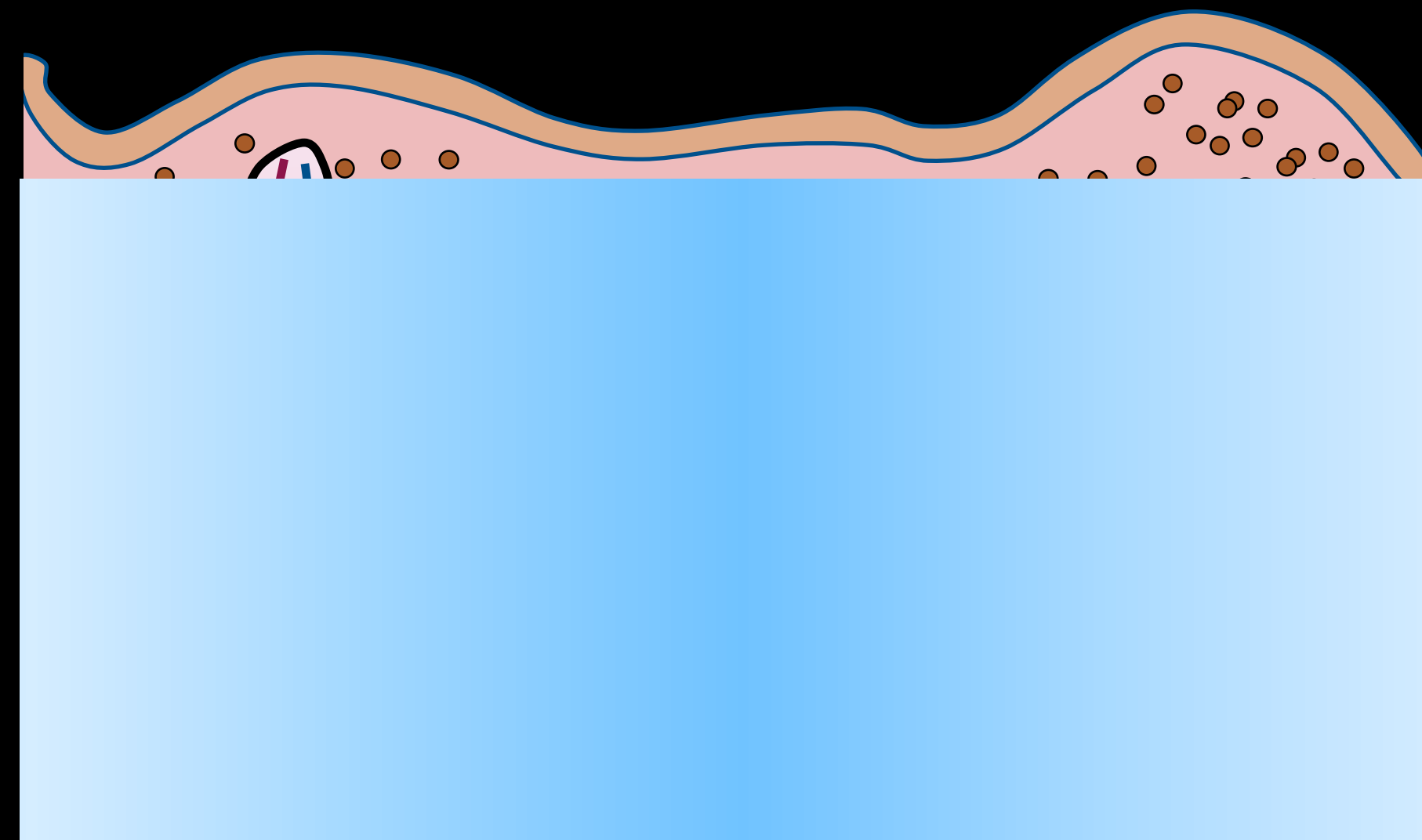
Rendering continuous refraction and scattering



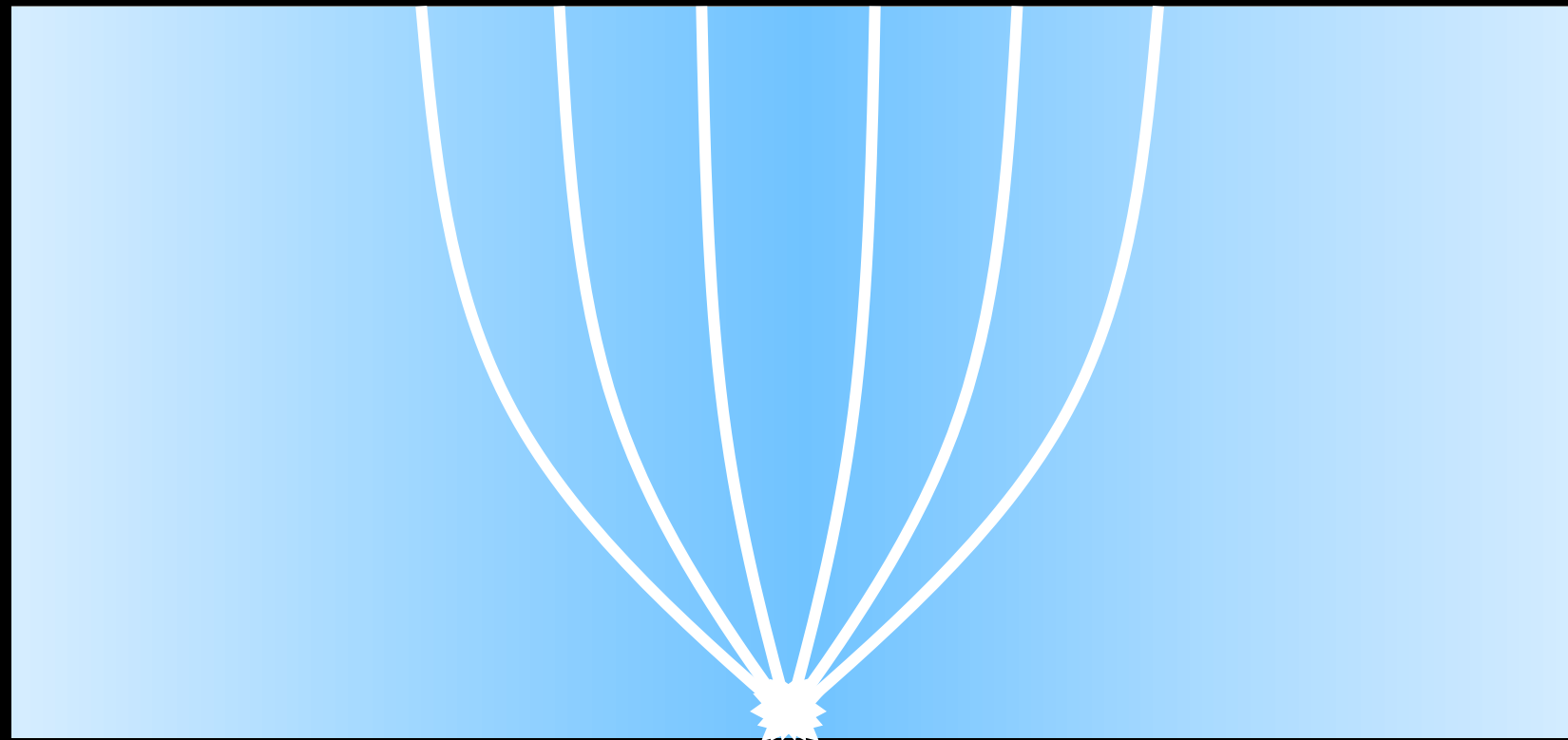
Rendering continuous refraction and scattering



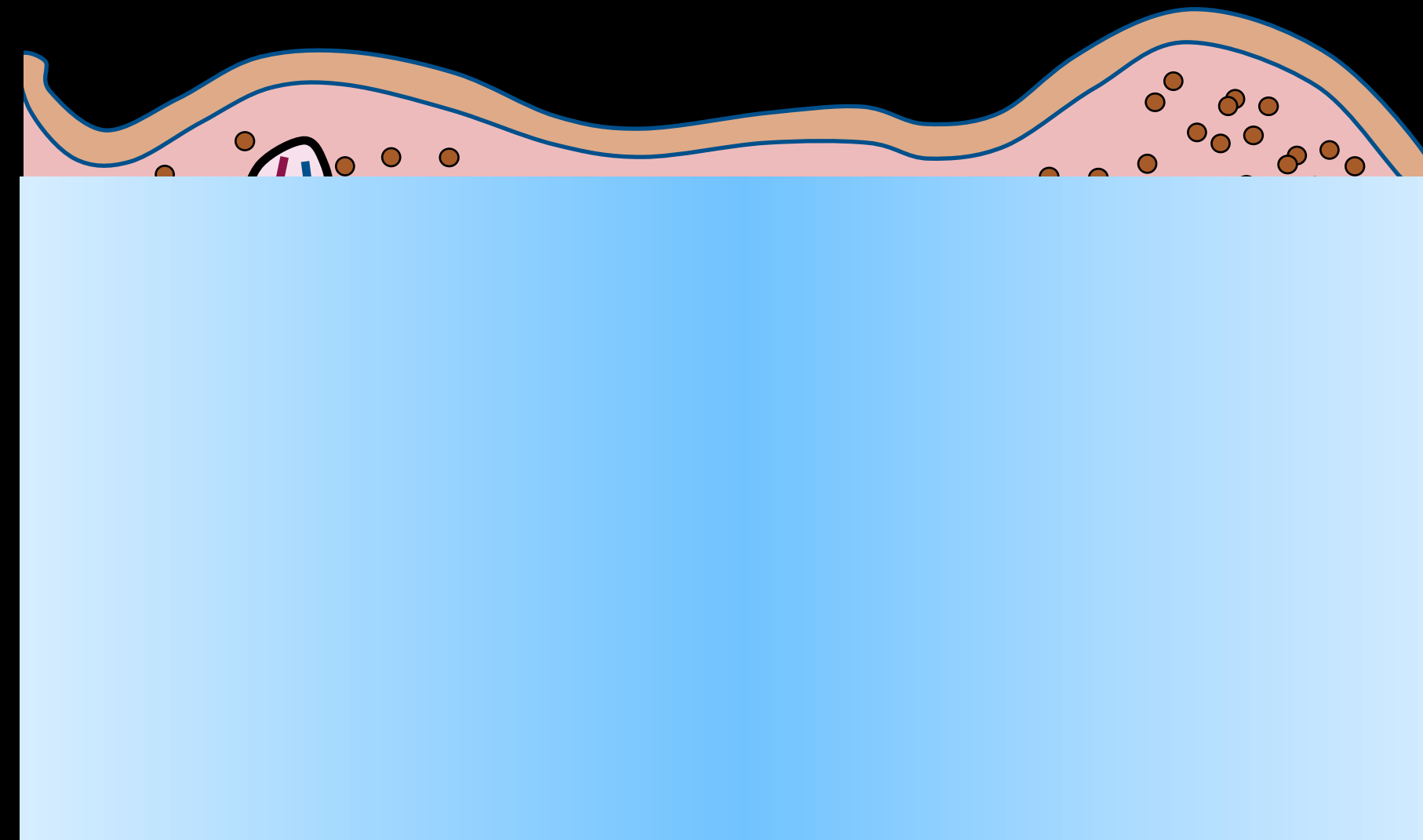
continuous refraction



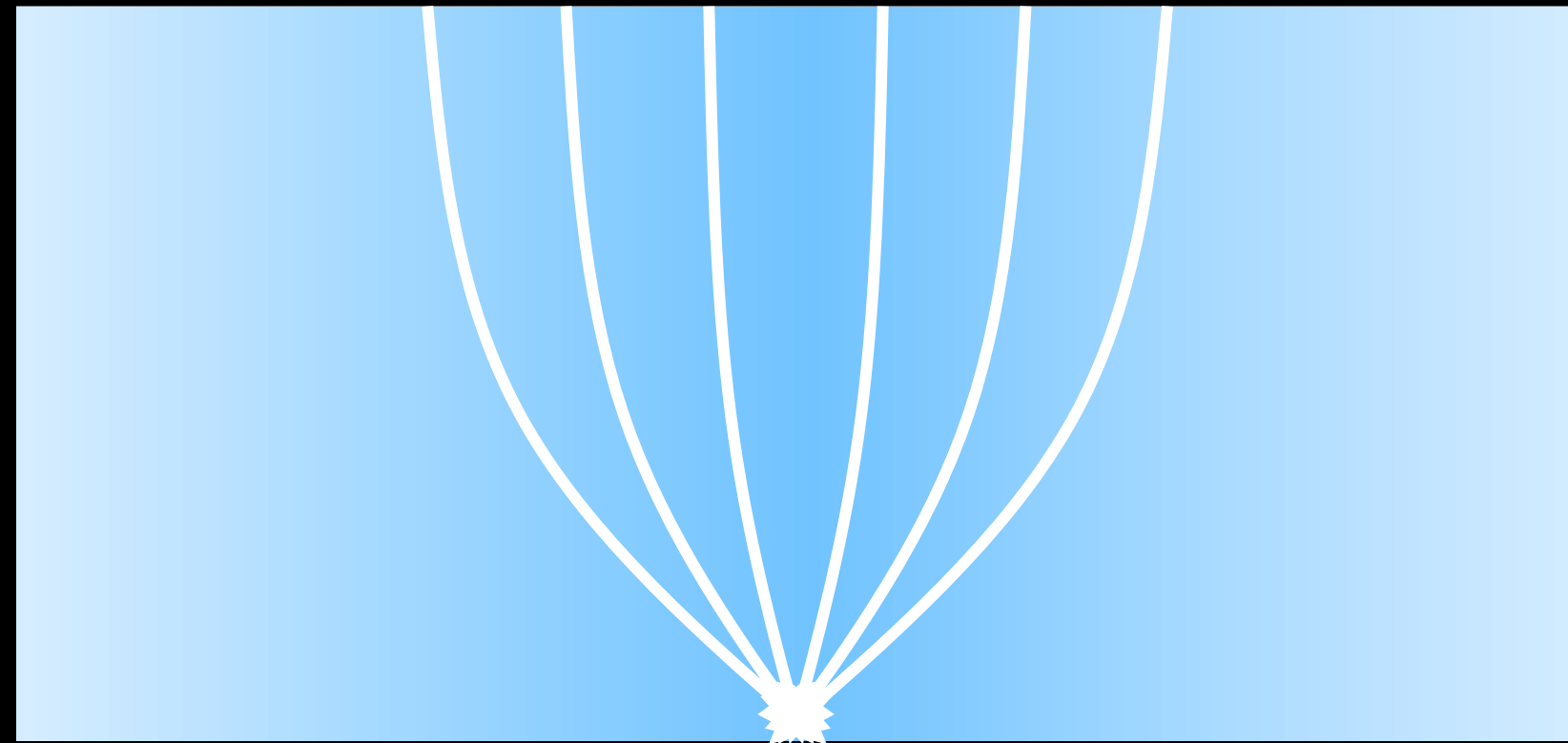
Rendering continuous refraction and scattering



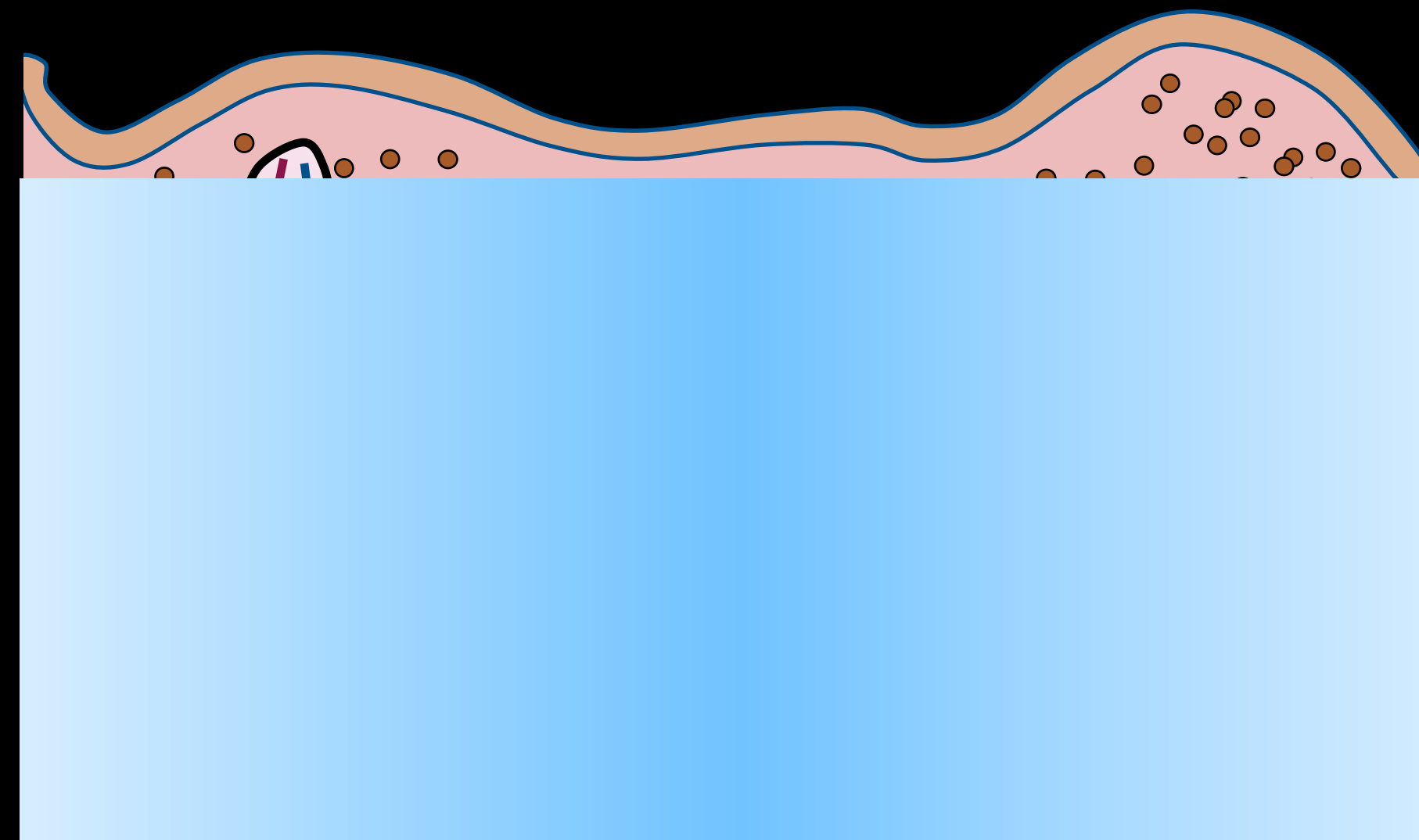
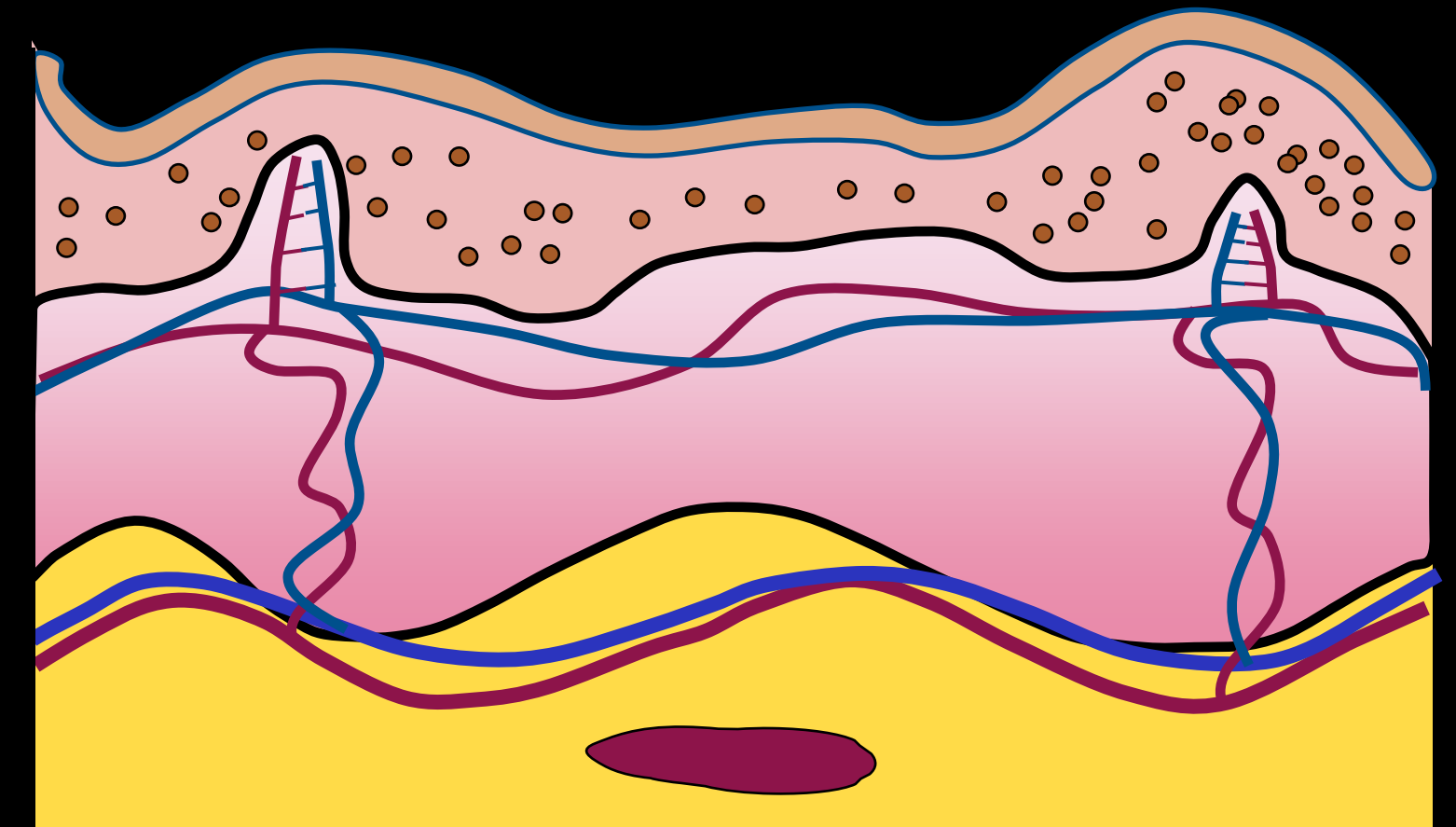
continuous refraction



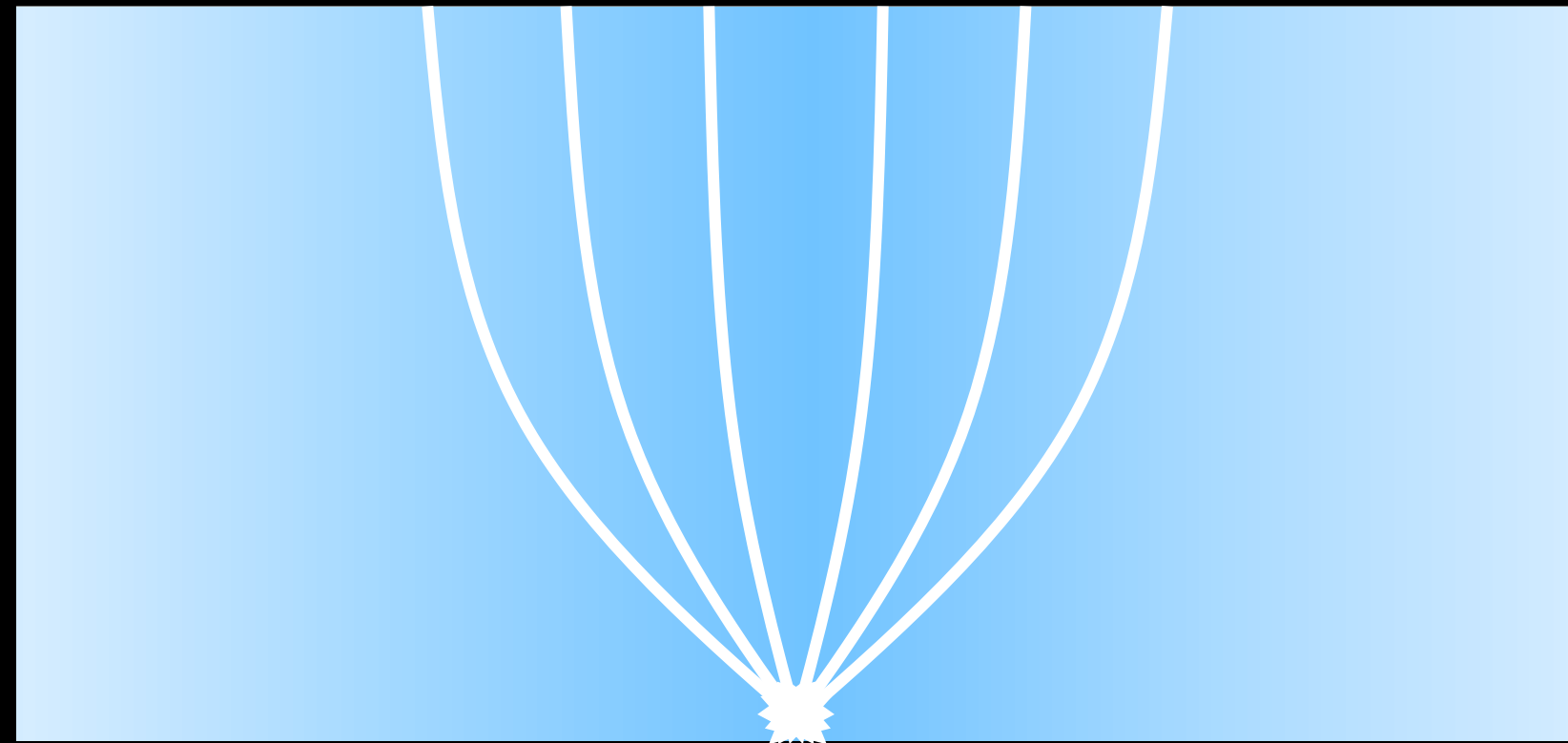
Rendering continuous refraction and scattering



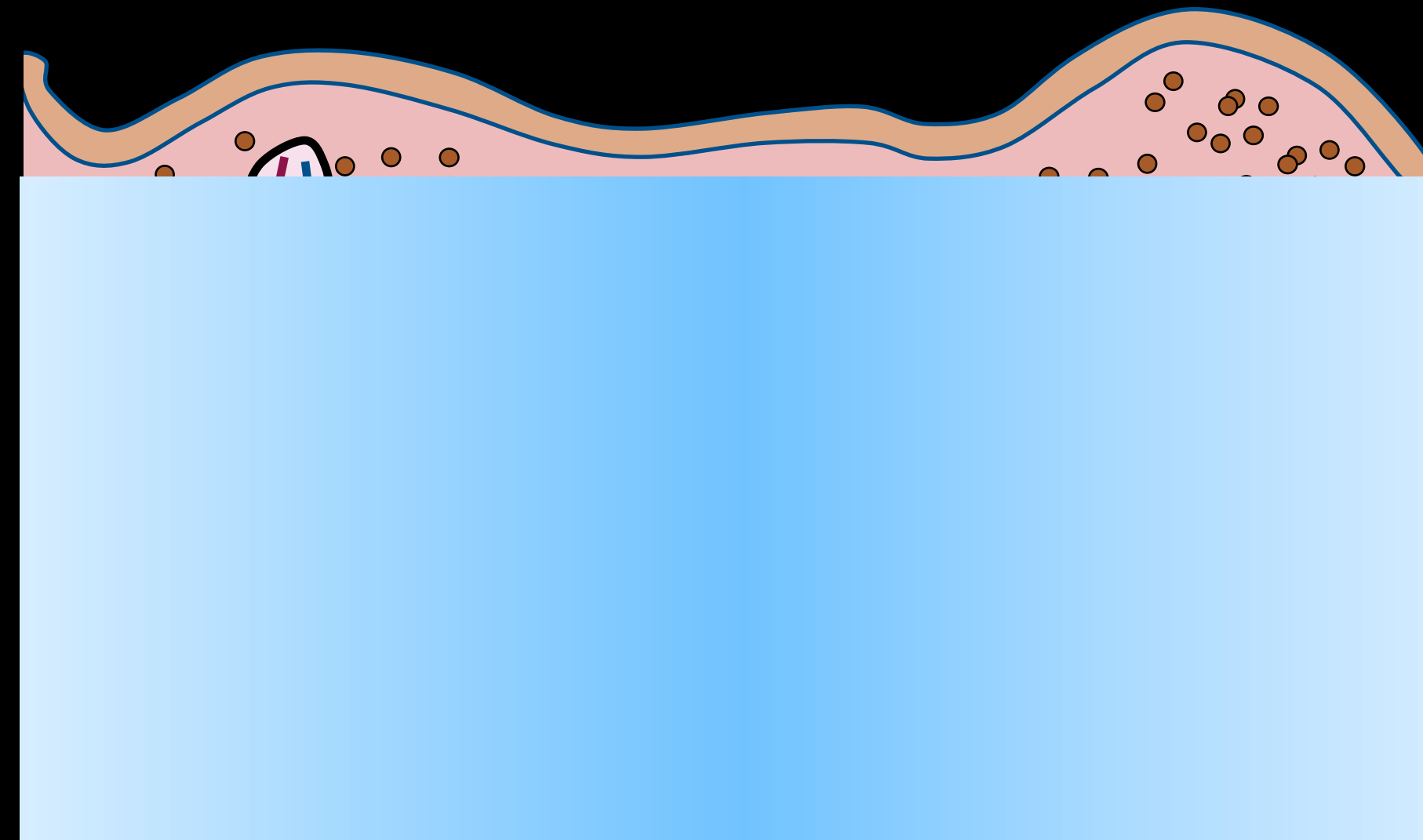
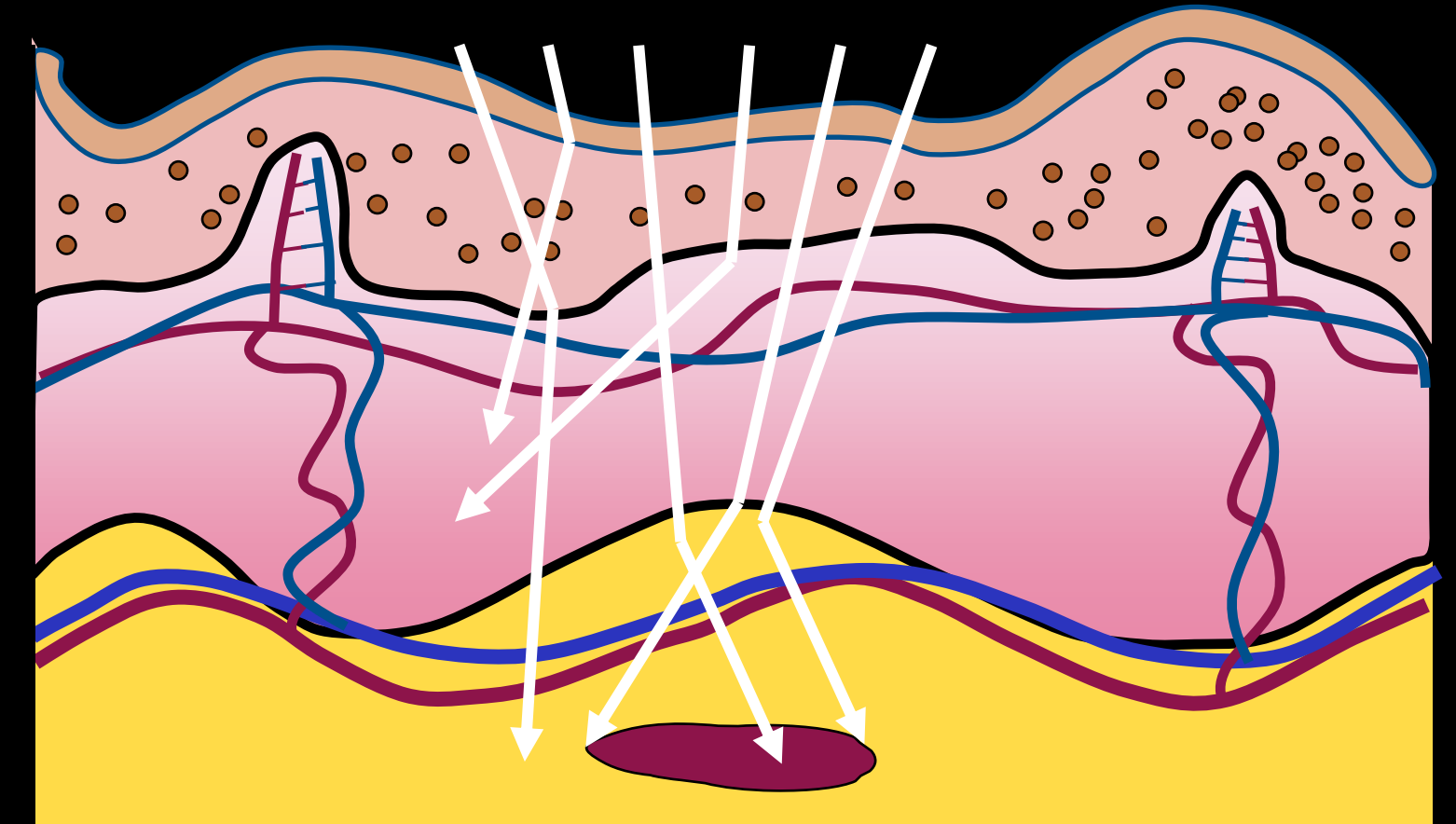
continuous refraction



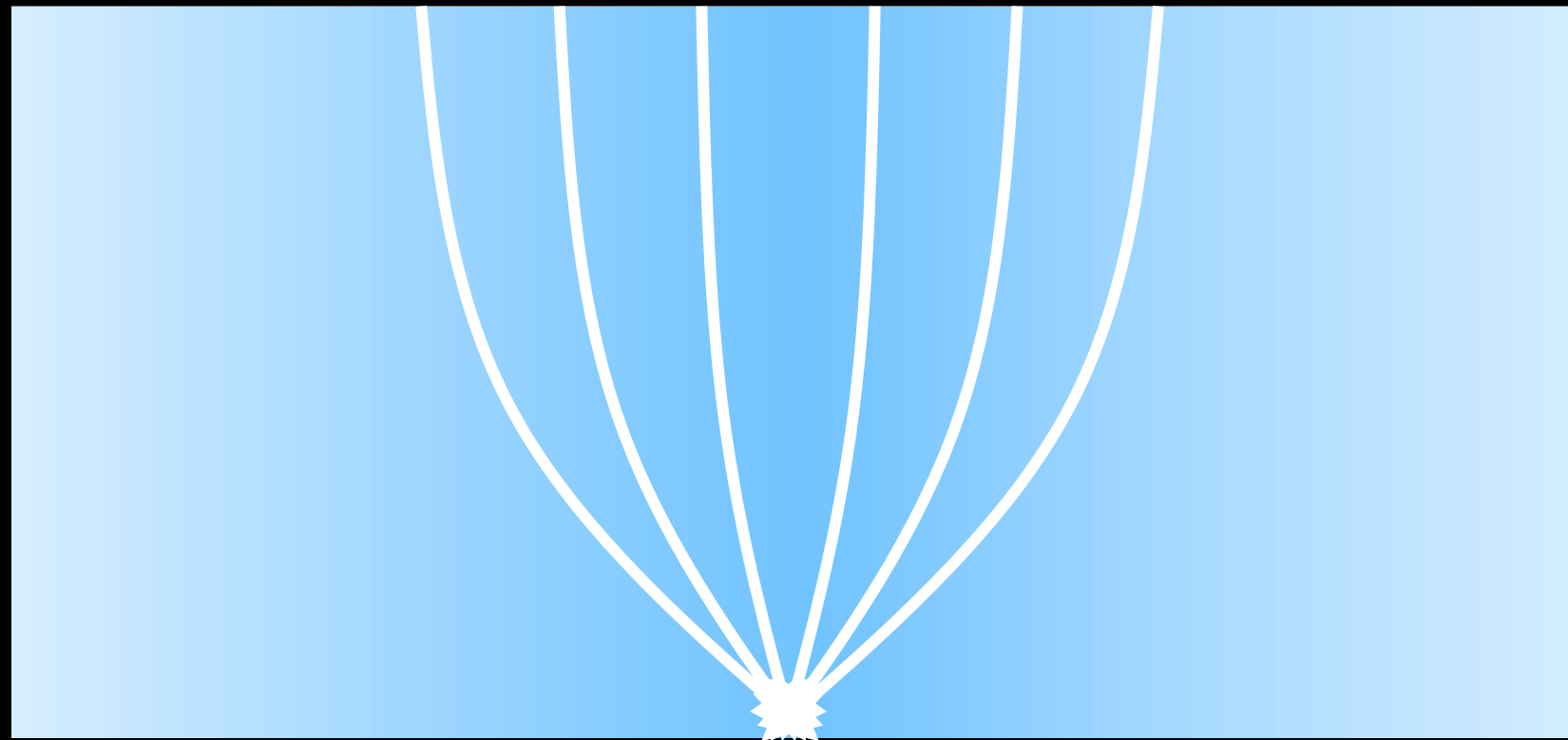
Rendering continuous refraction and scattering



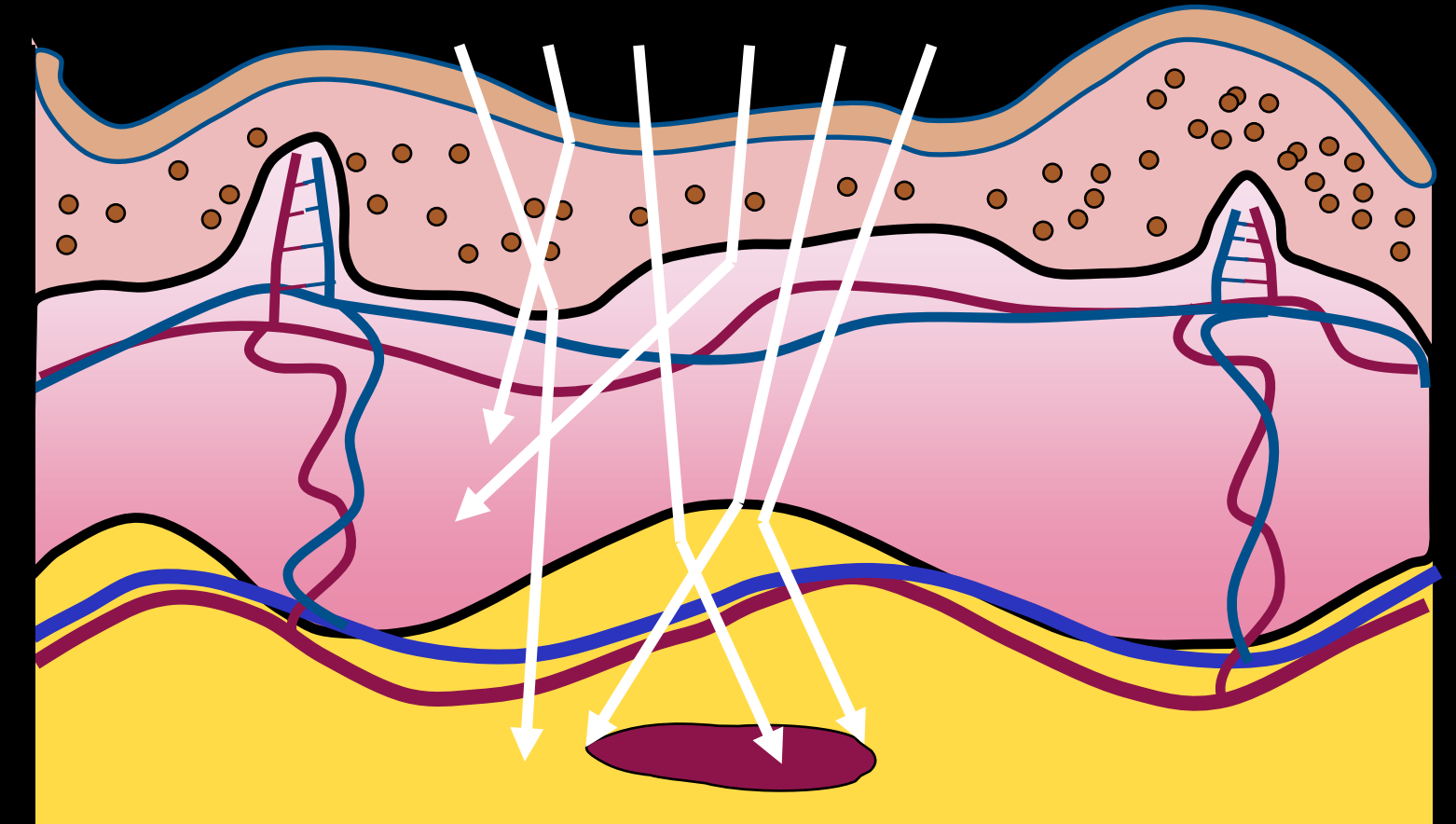
continuous refraction
+
scattering



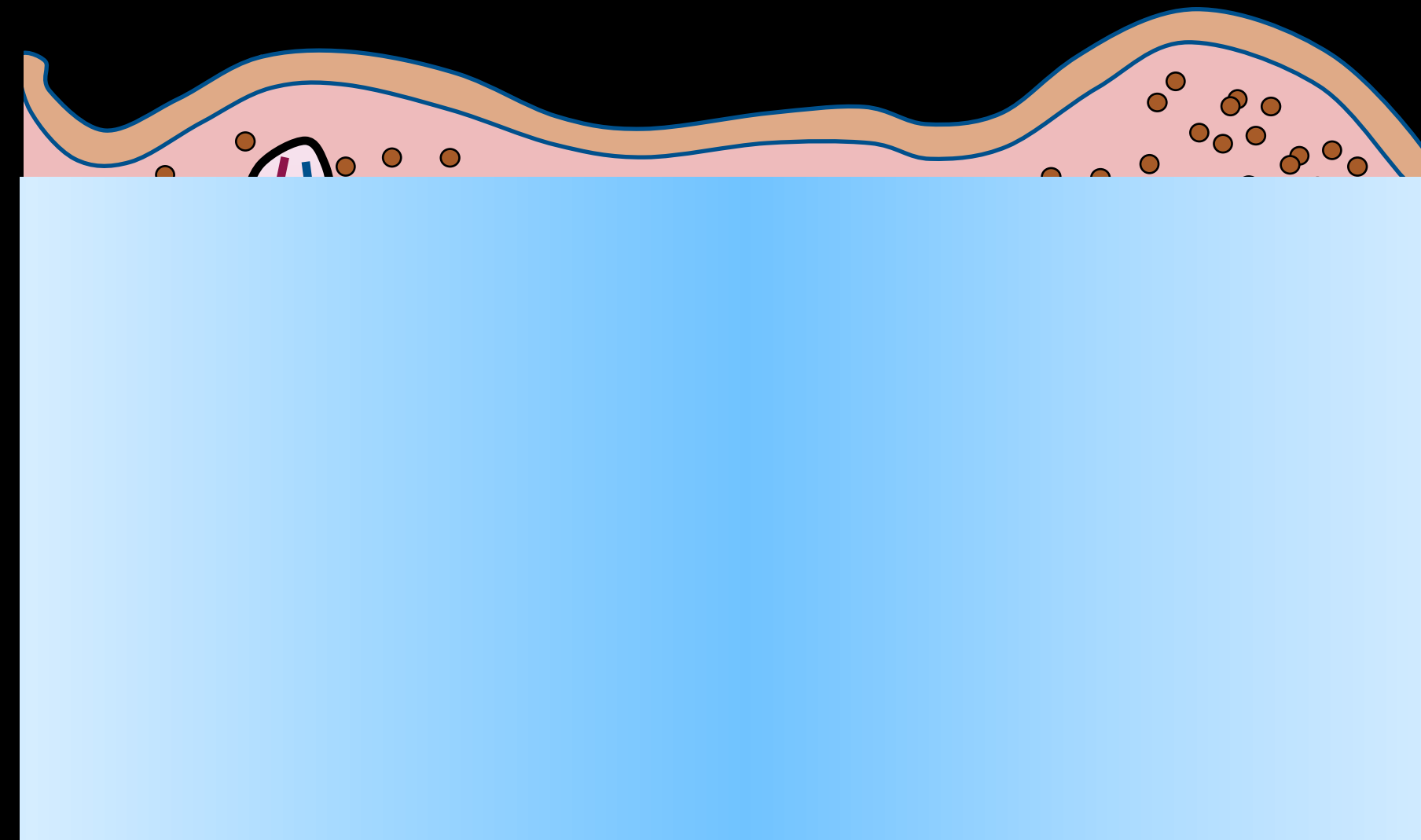
Rendering continuous refraction and scattering



continuous refraction
+
scattering



[Kravtsov and Orlov, Book, 1990]
[Gröller, Visual Comp., 1995]
[Stam and Languénou, Rend. Techn., 1996]
[Weiskopf et al., Com. Graph. forum, 2004]
[Gutierrez et al., In Rend. Techn., 2005]
[Ihrke et al., ToG, 2007]
[Atcheson et al., ToG, 2008]
[Ji et al., CVPR, 2013]
[Pedrotti et al., Book, 2017]
[Scopelliti et al., Nature LSA, 2019]



[Chandrasekhar, book, 1960]
[Lenoble, book, 1985]
[Lafortune and Willems, 1996]
[Cammarrano and Jensen, 2002]
[Gutierrez et al., Com. and Graph. 2006]
[Jarosz et al., Comp. Graph. forum, 2008]
[Jakob et al., ToG, 2010]
[Jarosz et al., ToG, 2011]
[Pediredla et al., JBO, 2016]
[Novak et al., Comp. Graph. forum, 2018]
[Bitterli et al., ToG, 2018]

Rendering continuous refraction ~~and scattering~~

[Kravtsov and Orlov, Book, 1990]

[Gröller, Visual Comp., 1995]

[Stam and Languénou, Rend. Techn., 1996]

[Weiskopf et al., Com. Graph. forum, 2004]

[Gutierrez et al., In Rend. Techn., 2005]

[Ihrke et al., ToG, 2007]

[Atcheson et al., ToG, 2008]

[Ji et al., CVPR, 2013]

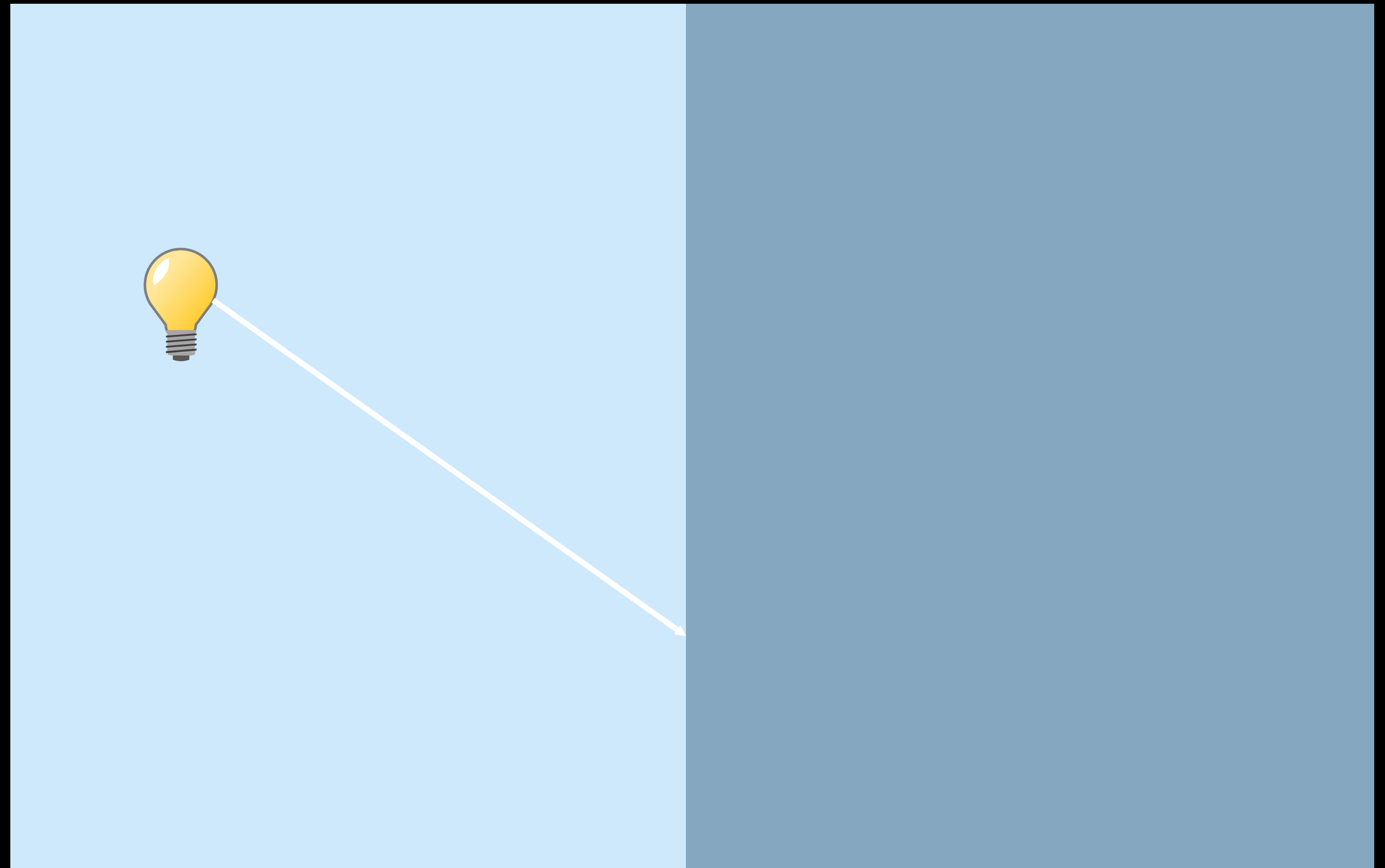
[Pedrotti et al., Book, 2017]

[Scopelliti et al., Nature LSA, 2019]



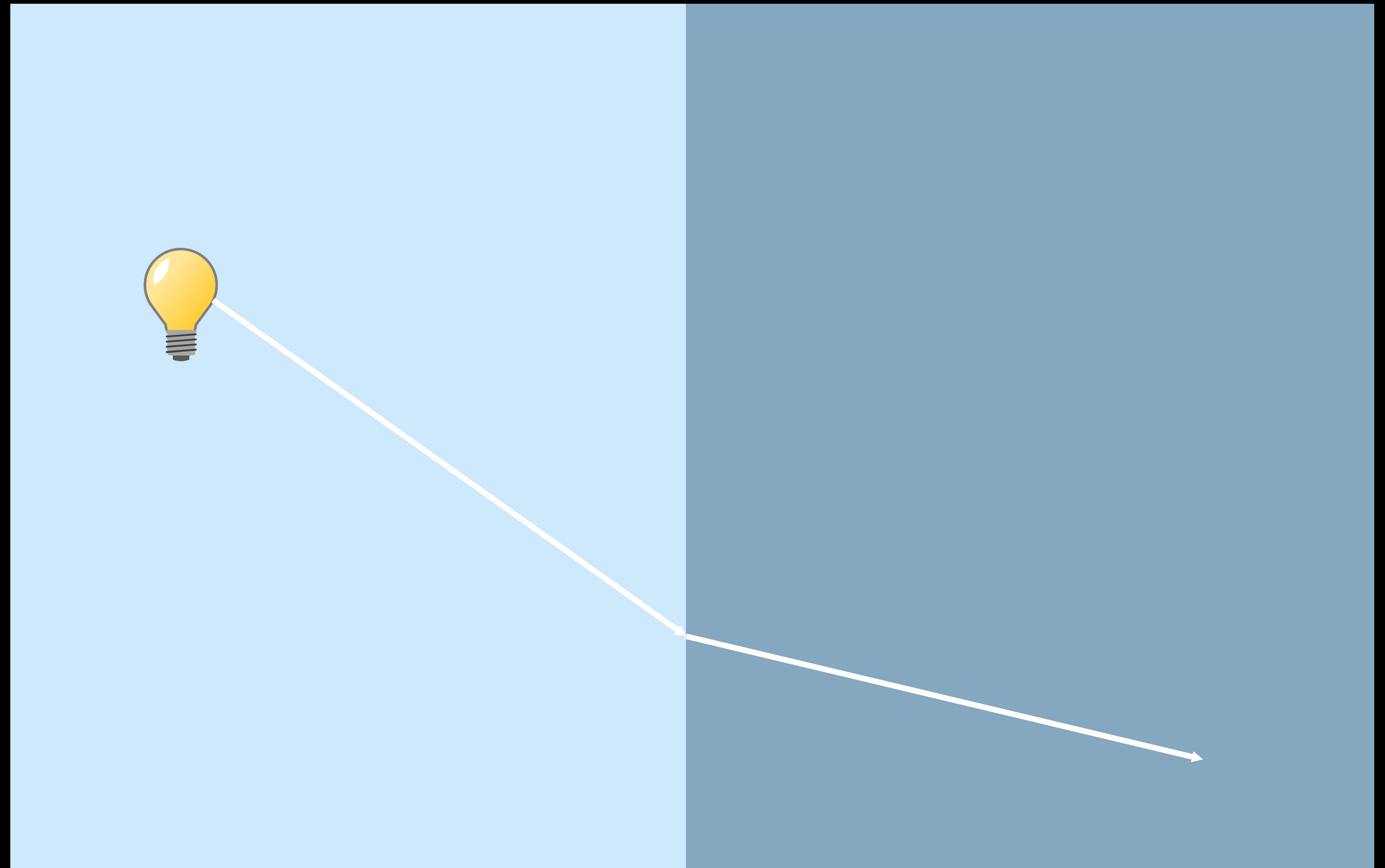
Rendering continuous refraction ~~and scattering~~

- [Kravtsov and Orlov, Book, 1990]
- [Gröllner, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction ~~and scattering~~

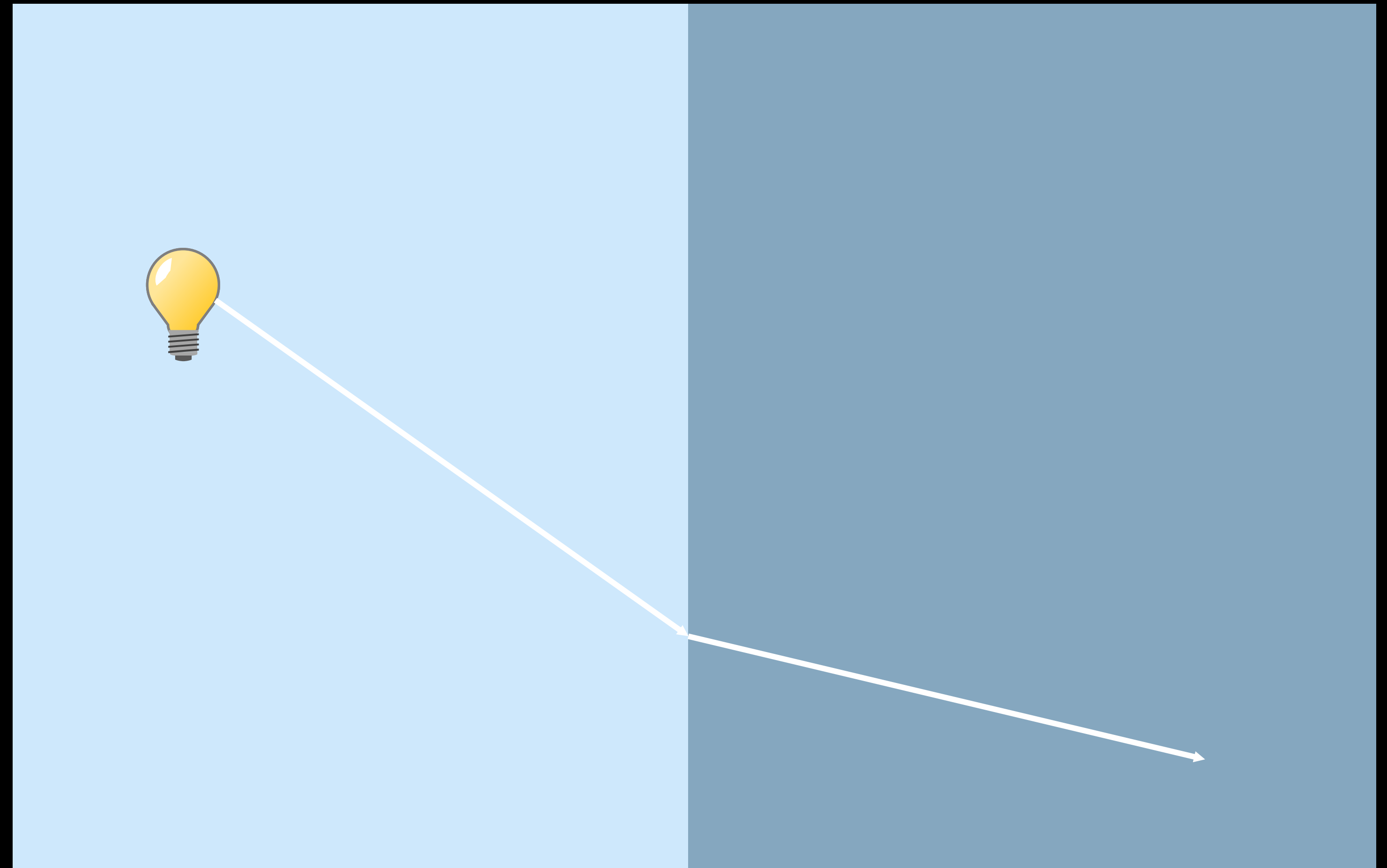
- [Kravtsov and Orlov, Book, 1990]
- [Gröllner, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction ~~and scattering~~

Snell's law

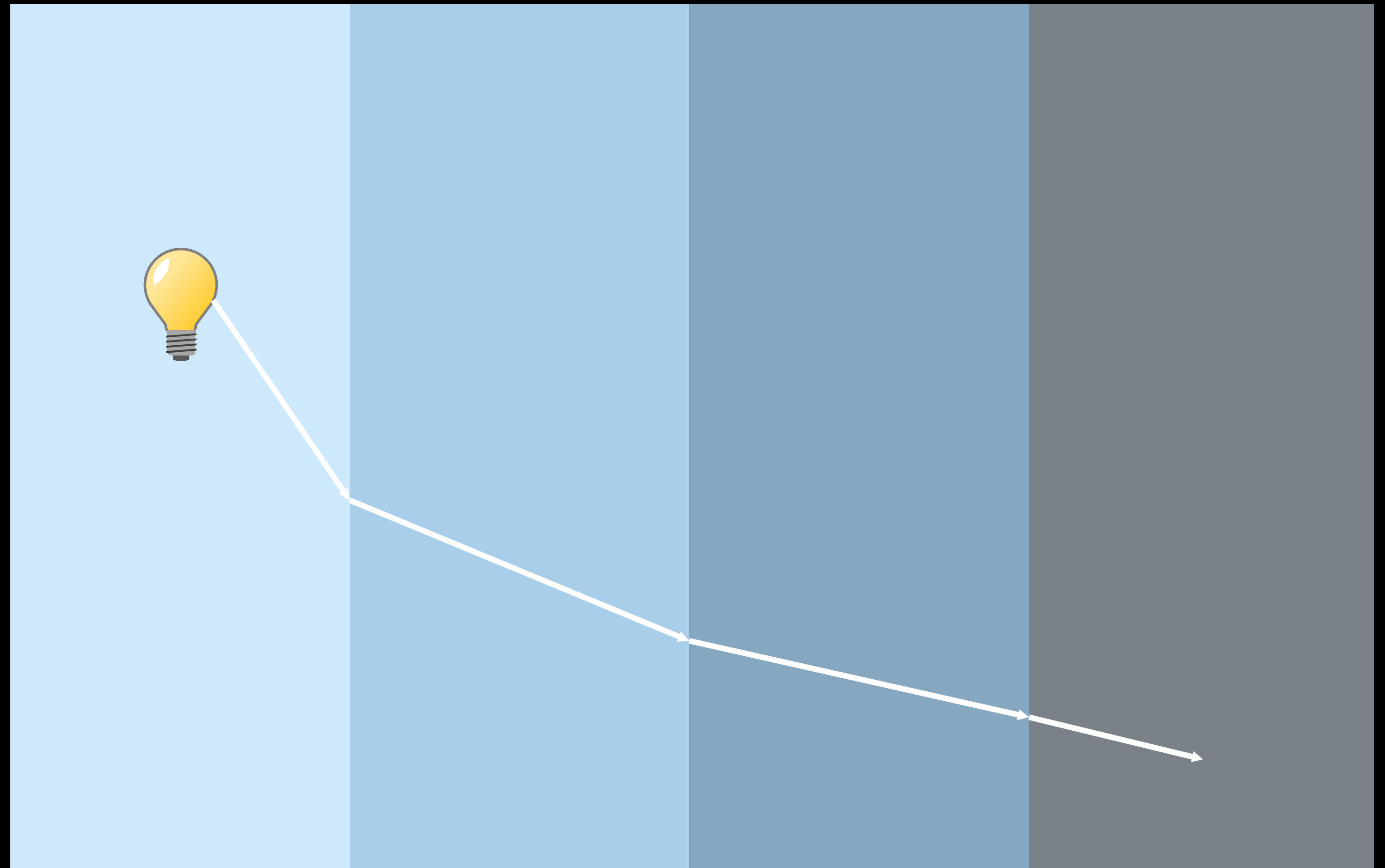
- [Kravtsov and Orlov, Book, 1990]
- [Gröllner, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction ~~and scattering~~

Snell's law

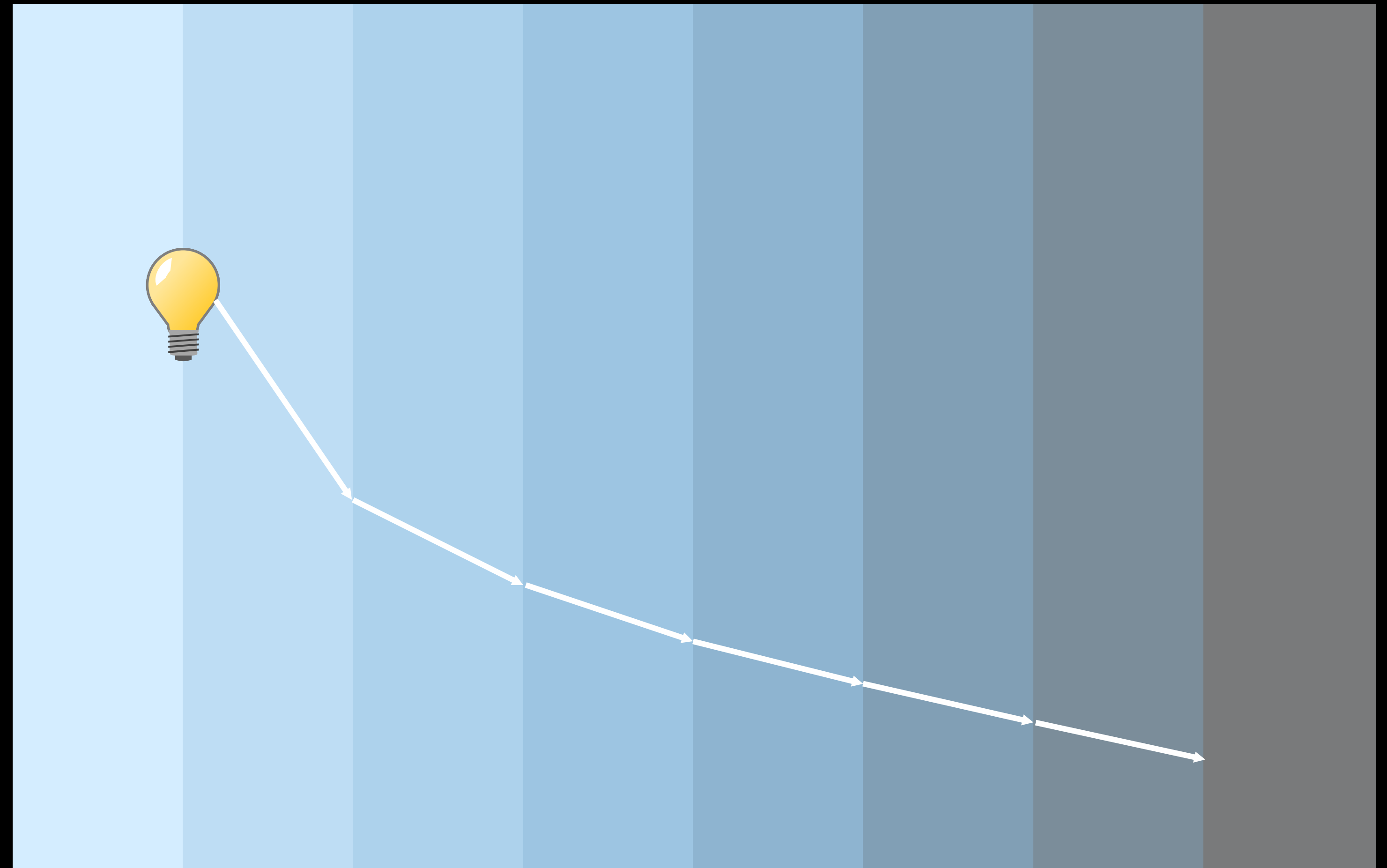
- [Kravtsov and Orlov, Book, 1990]
- [Gröllner, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction ~~and scattering~~

Snell's law

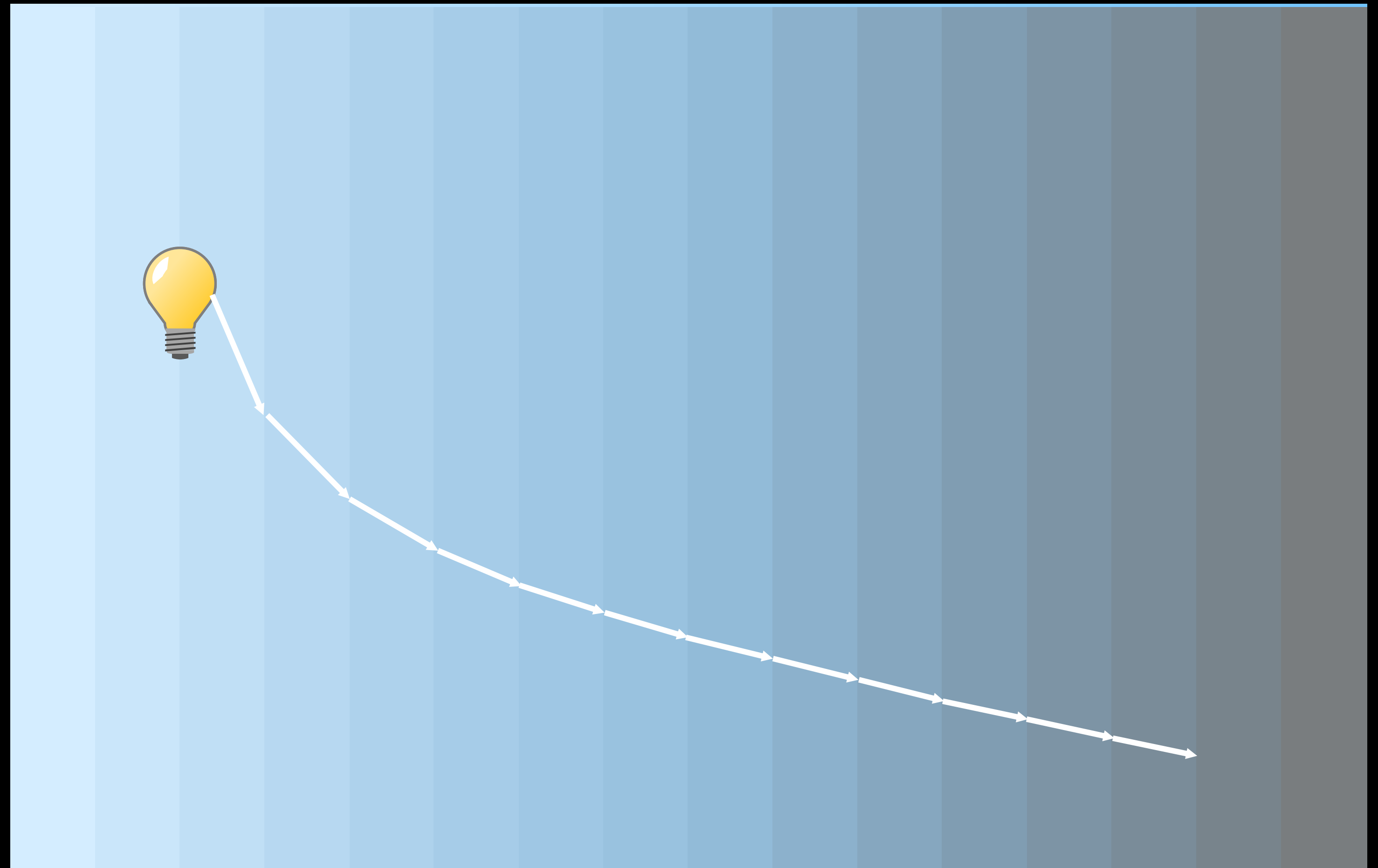
- [Kravtsov and Orlov, Book, 1990]
- [Gröllner, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction ~~and scattering~~

Snell's law

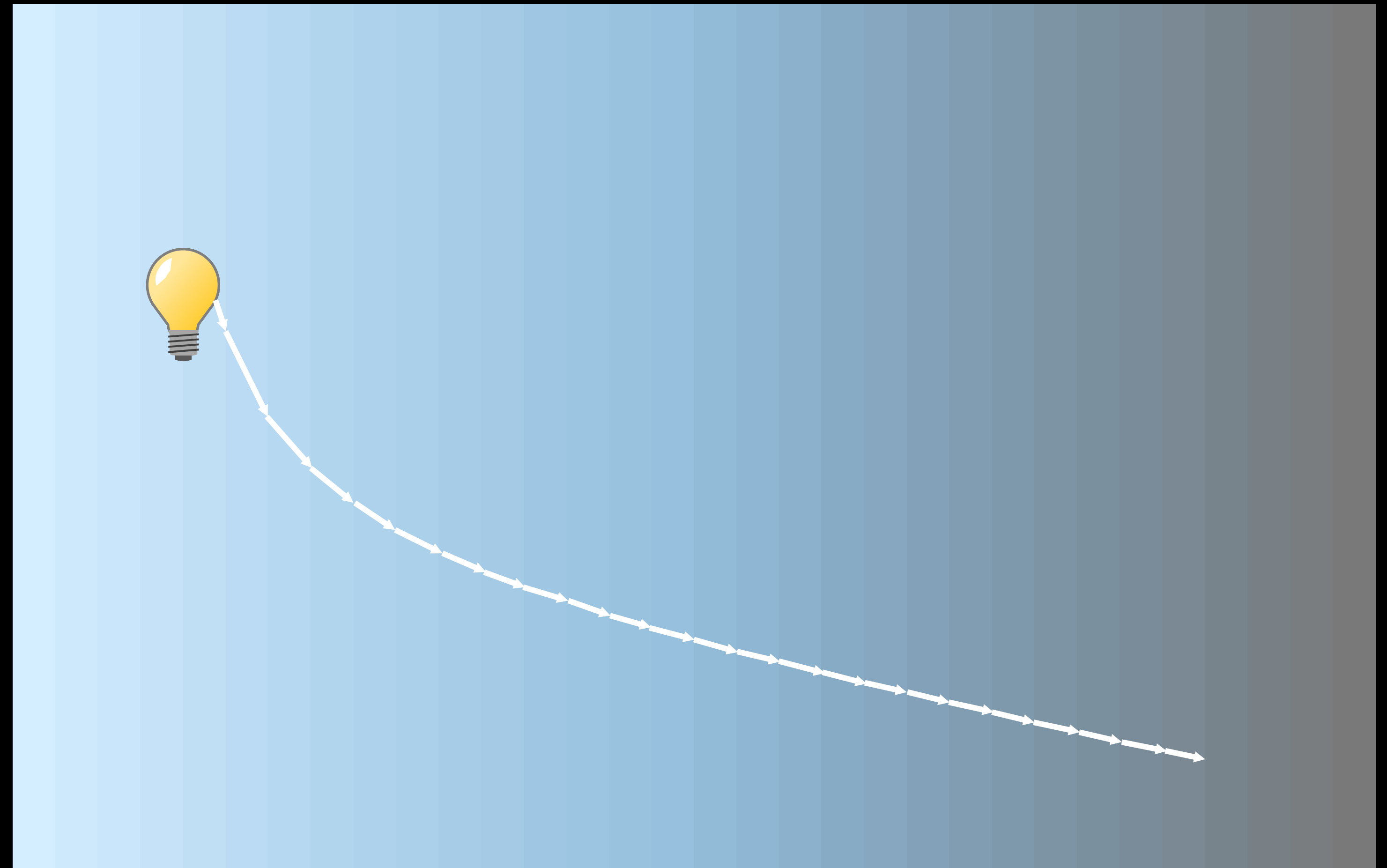
- [Kravtsov and Orlov, Book, 1990]
- [Gröllner, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction ~~and scattering~~

Snell's law

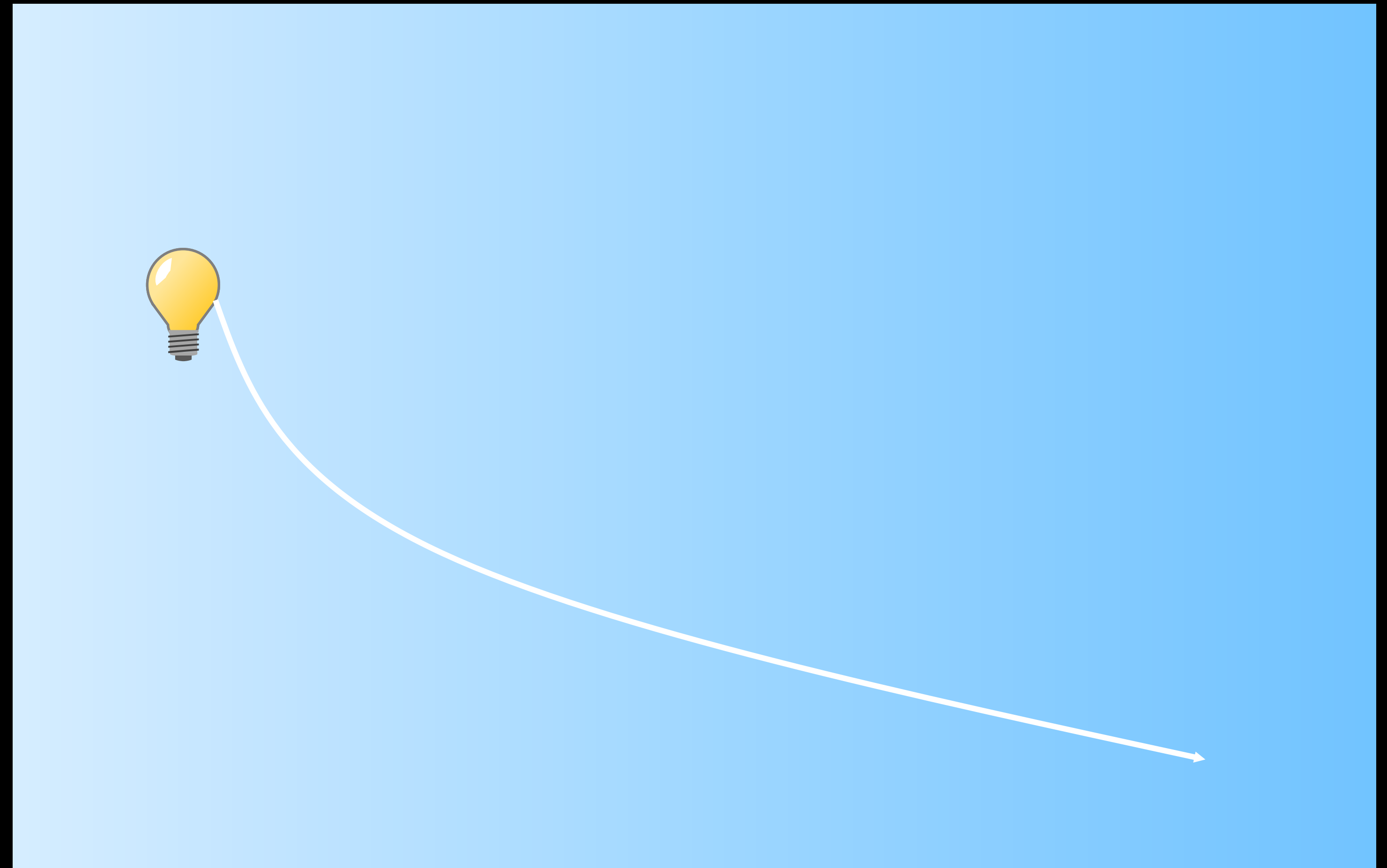
- [Kravtsov and Orlov, Book, 1990]
- [Gröllner, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction ~~and scattering~~

refractive ray tracing

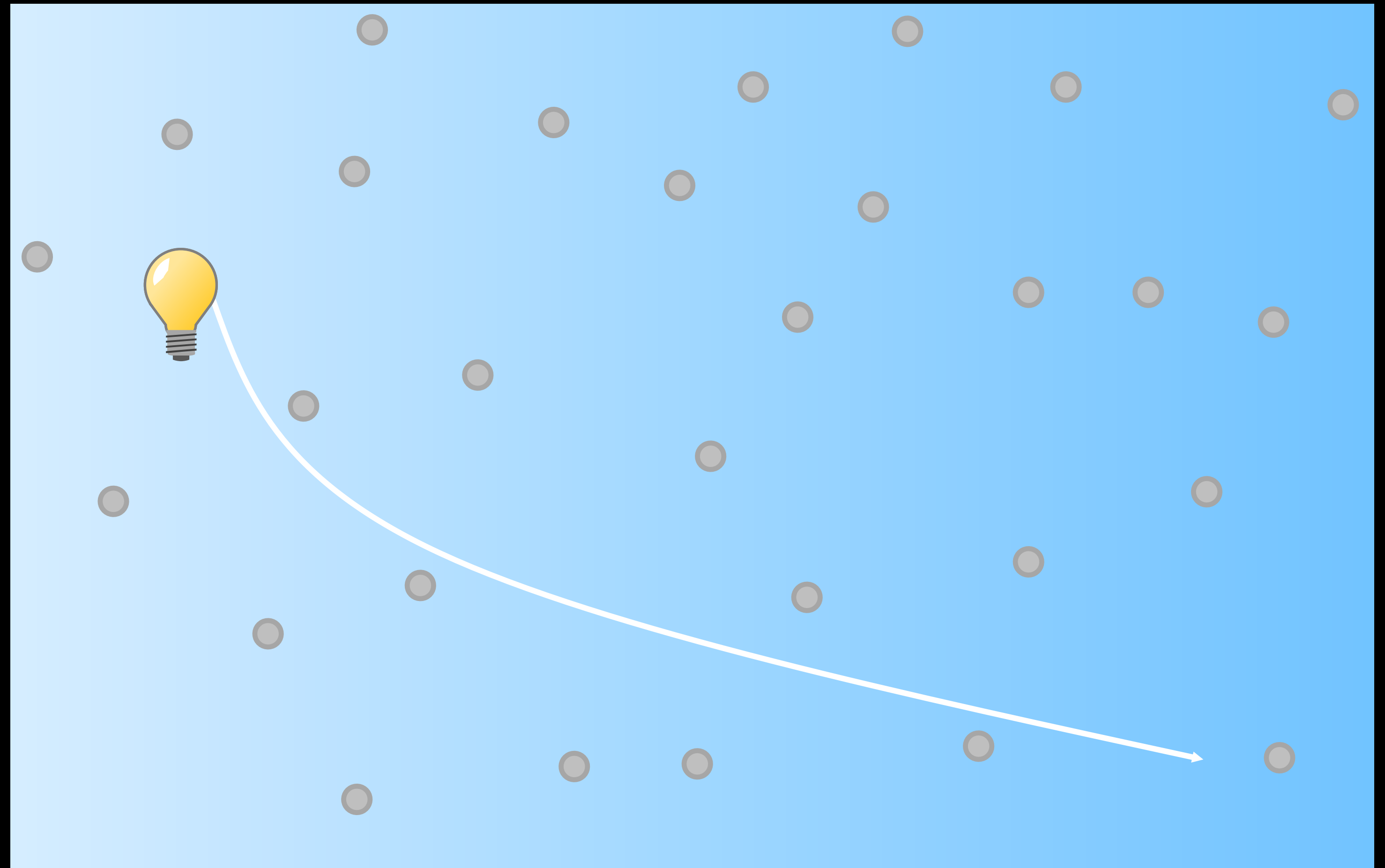
- [Kravtsov and Orlov, Book, 1990]
- [Gröller, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction and scattering

refractive ray tracing

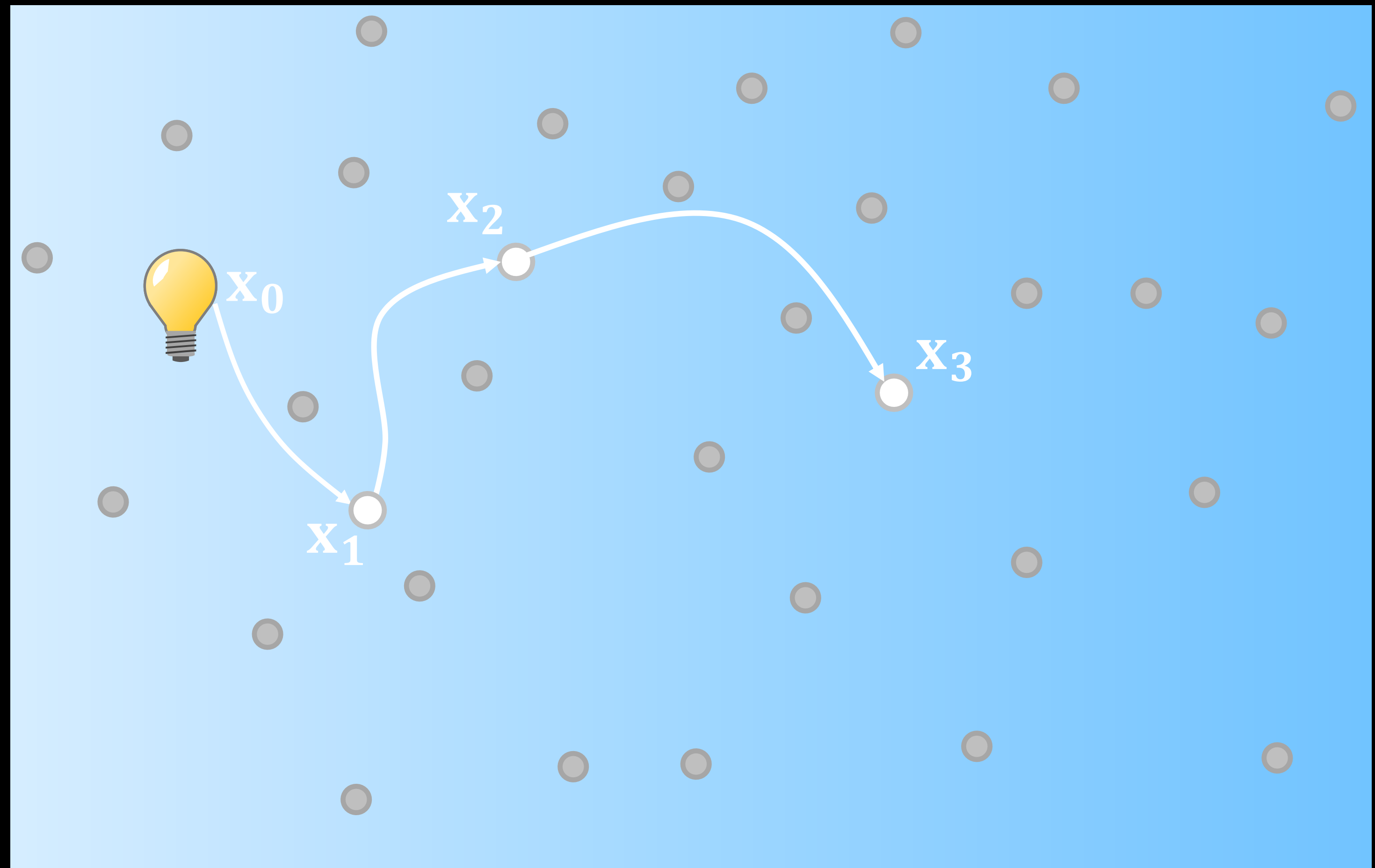
- [Kravtsov and Orlov, Book, 1990]
- [Gröller, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction and scattering

refractive ray tracing

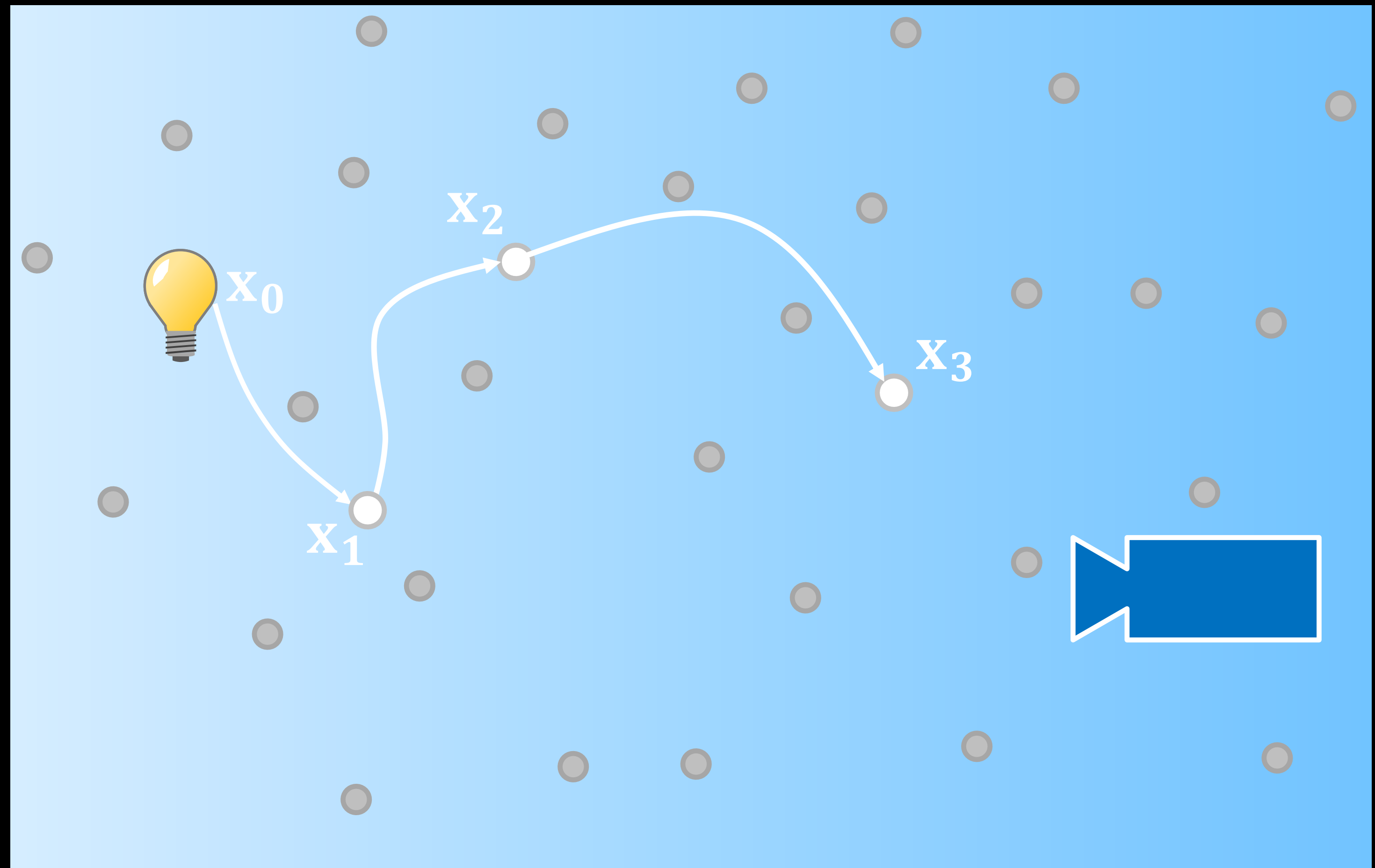
- [Kravtsov and Orlov, Book, 1990]
- [Gröller, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction and scattering

refractive ray tracing

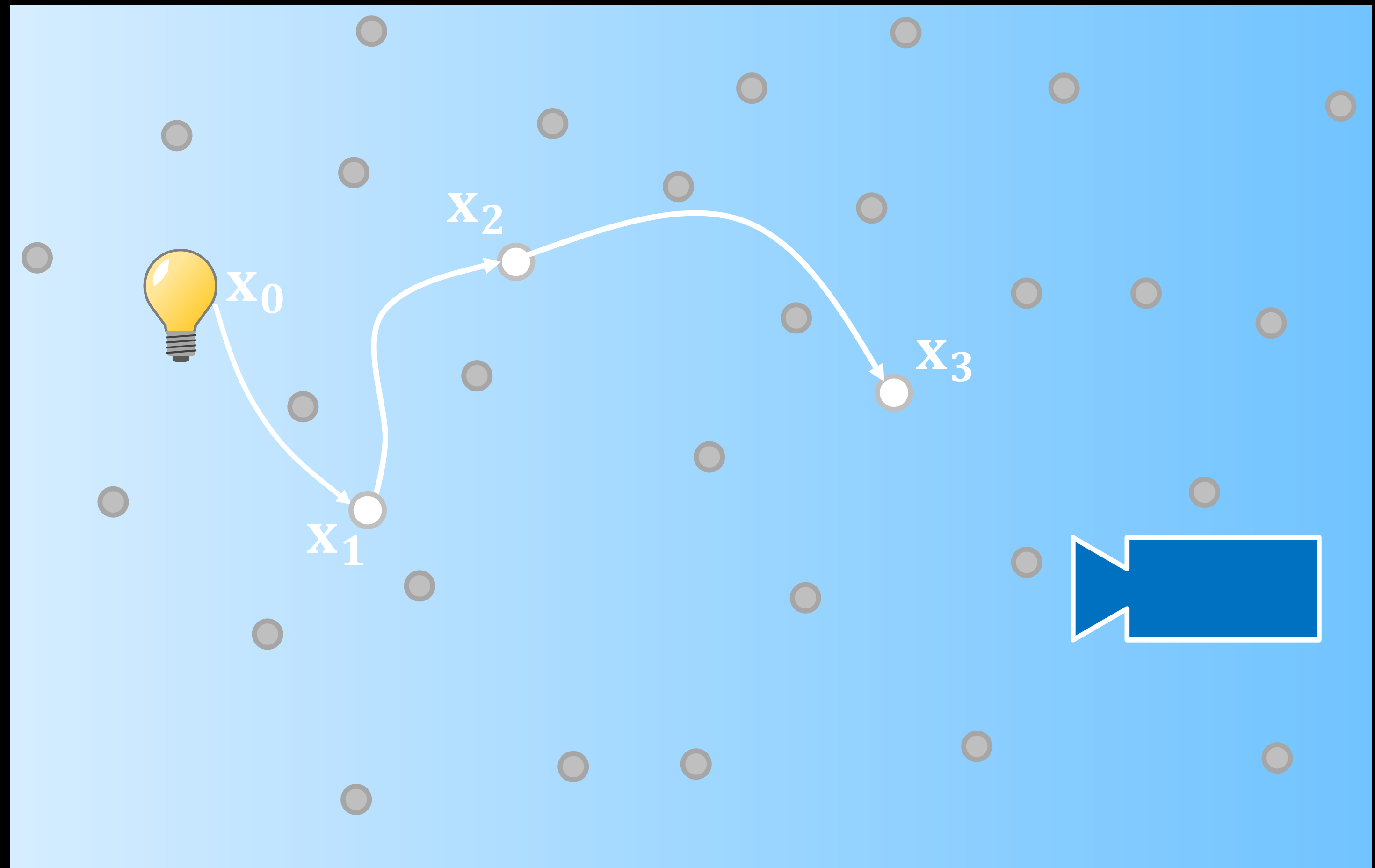
- [Kravtsov and Orlov, Book, 1990]
- [Gröller, Visual Comp., 1995]
- [Stam and Languénou, Rend. Techn., 1996]
- [Weiskopf et al., Com. Graph. forum, 2004]
- [Gutierrez et al., In Rend. Techn., 2005]
- [Ihrke et al., ToG, 2007]
- [Atcheson et al., ToG, 2008]
- [Ji et al., CVPR, 2013]
- [Pedrotti et al., Book, 2017]
- [Scopelliti et al., Nature LSA, 2019]



Rendering continuous refraction and scattering

radiative transfer equation

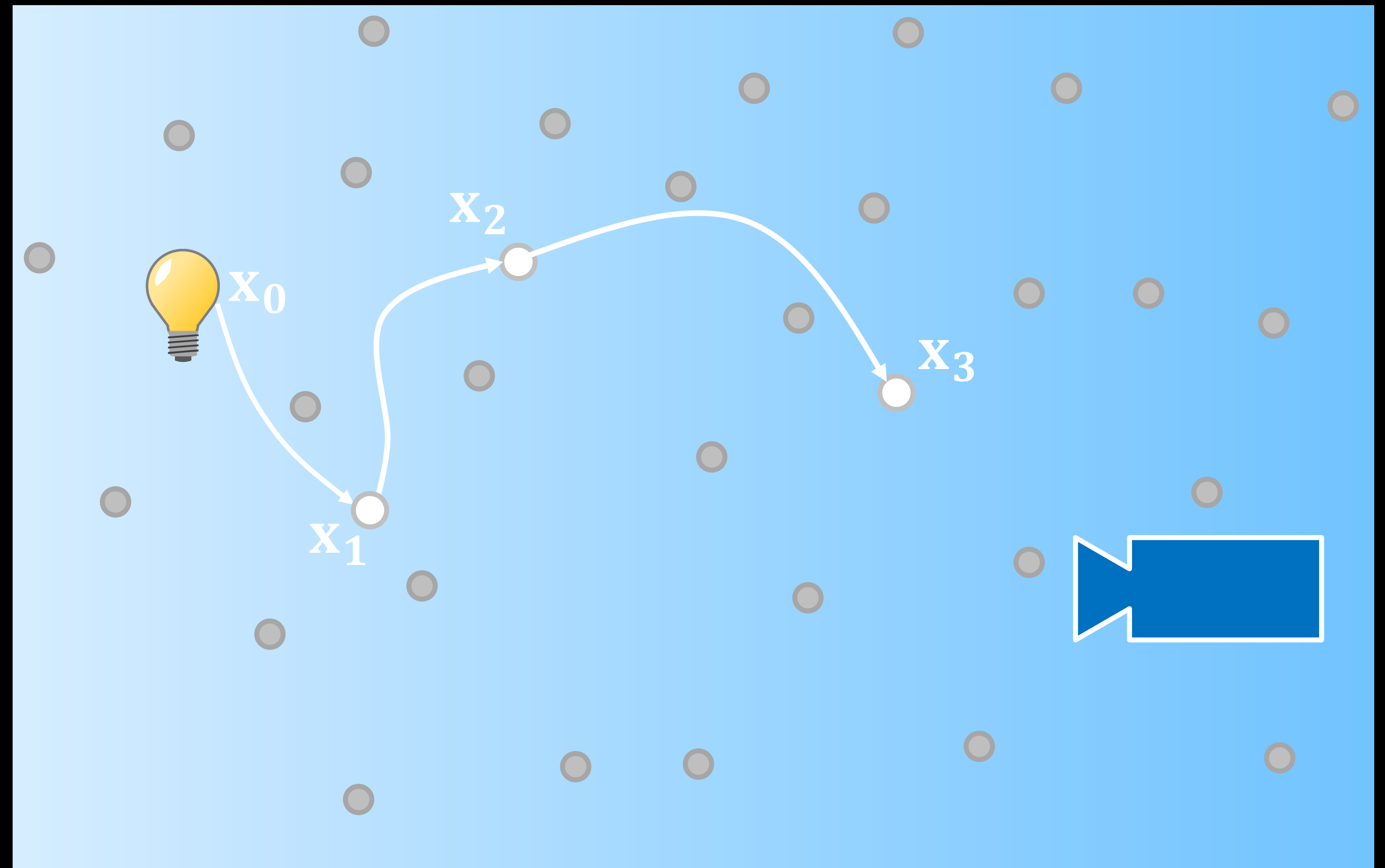
- [Chandrasekhar, book, 1960]
- [Lenoble, book, 1985]
- [Lafortune and Willems, 1996]
- [Cammarano and Jensen, 2002]
- [Gutierrez et al., Com. and Graph. 2006]
- [Jarosz et al., Comp. Graph. forum, 2008]
- [Jakob et al., ToG, 2010]
- [Jarosz et al., ToG, 2011]
- [Pediredla et al., JBO, 2016]
- [Novak et al., Comp. Graph. forum, 2018]
- [Bitterli et al., ToG, 2018]



Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

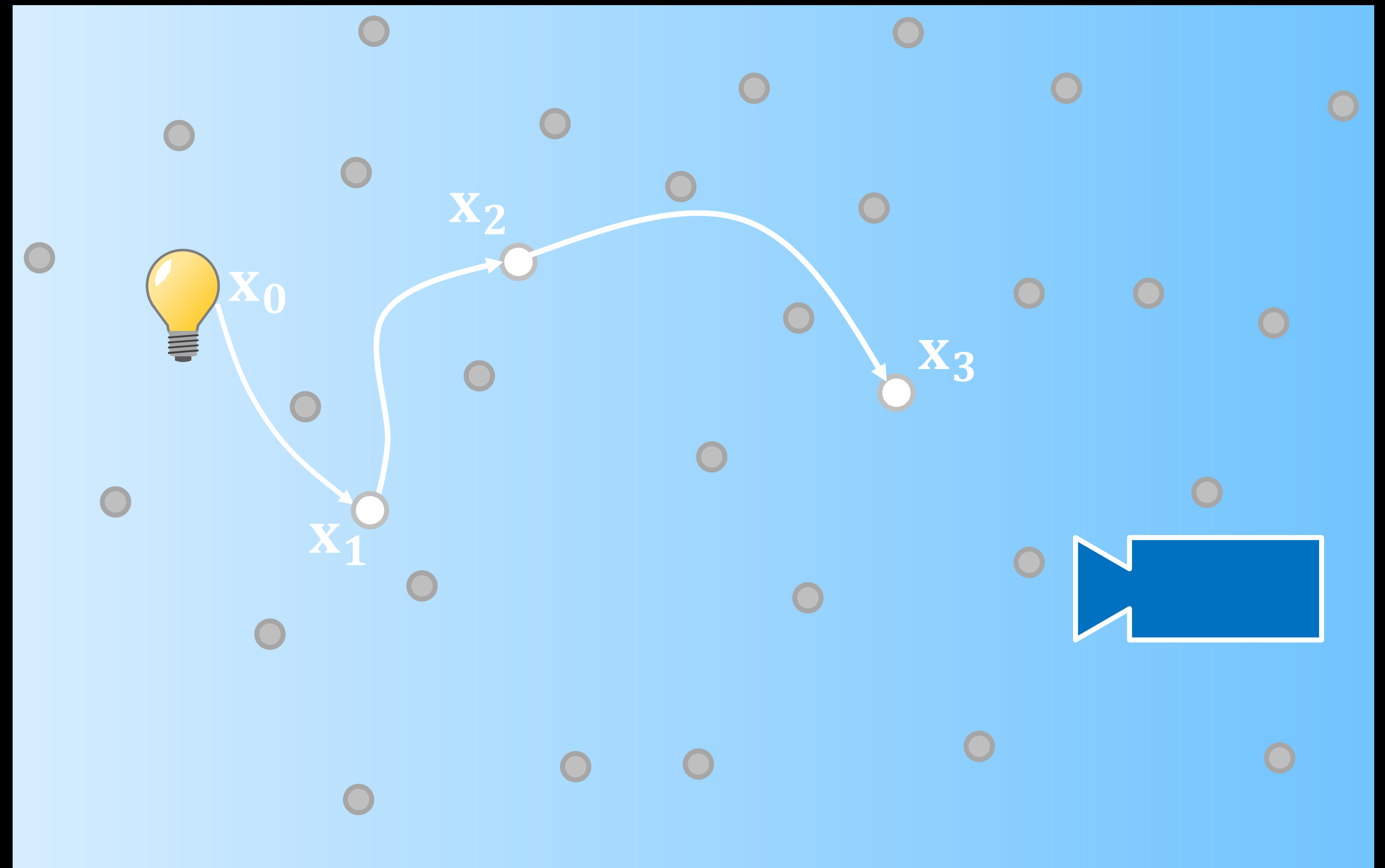


Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath

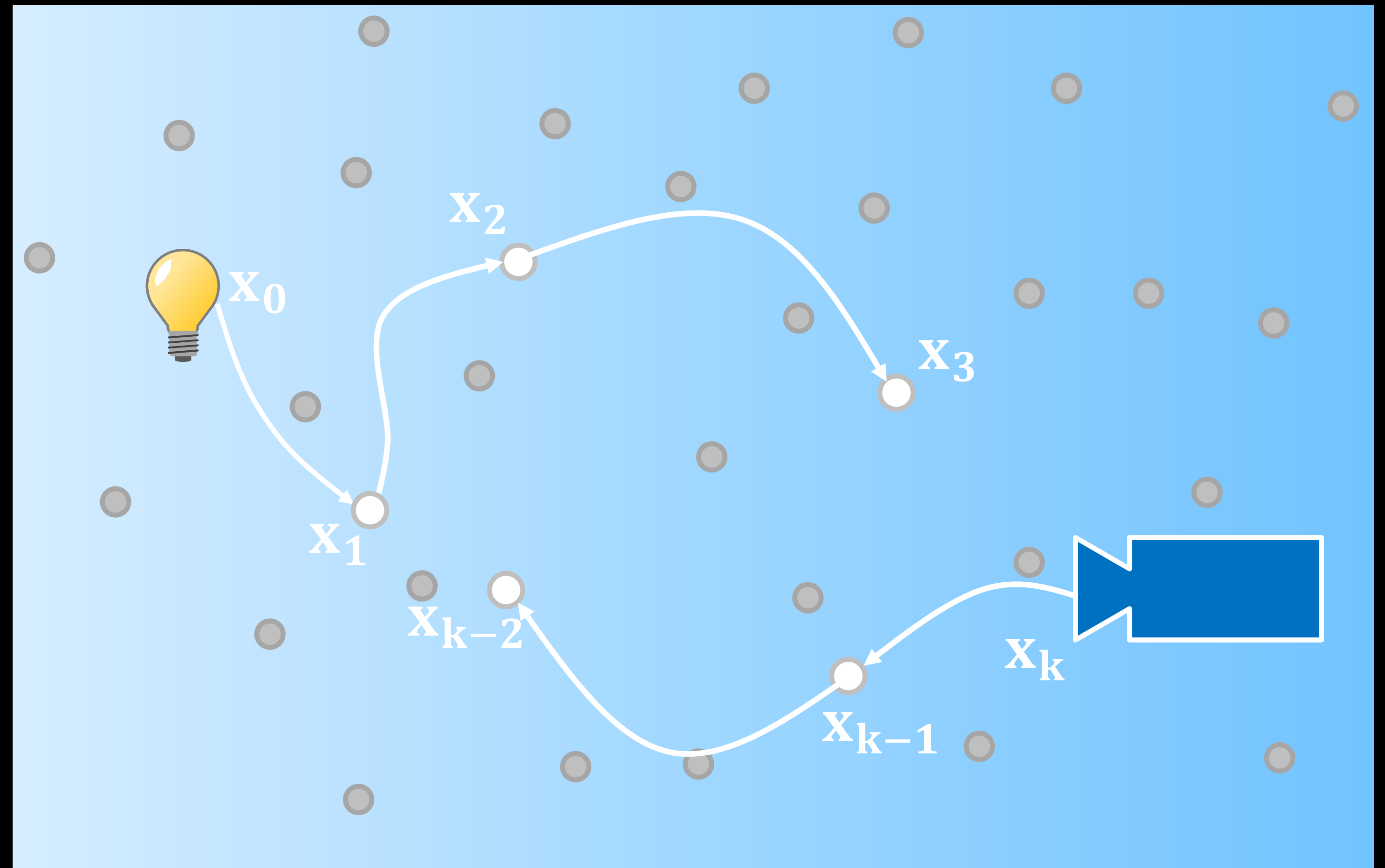


Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath

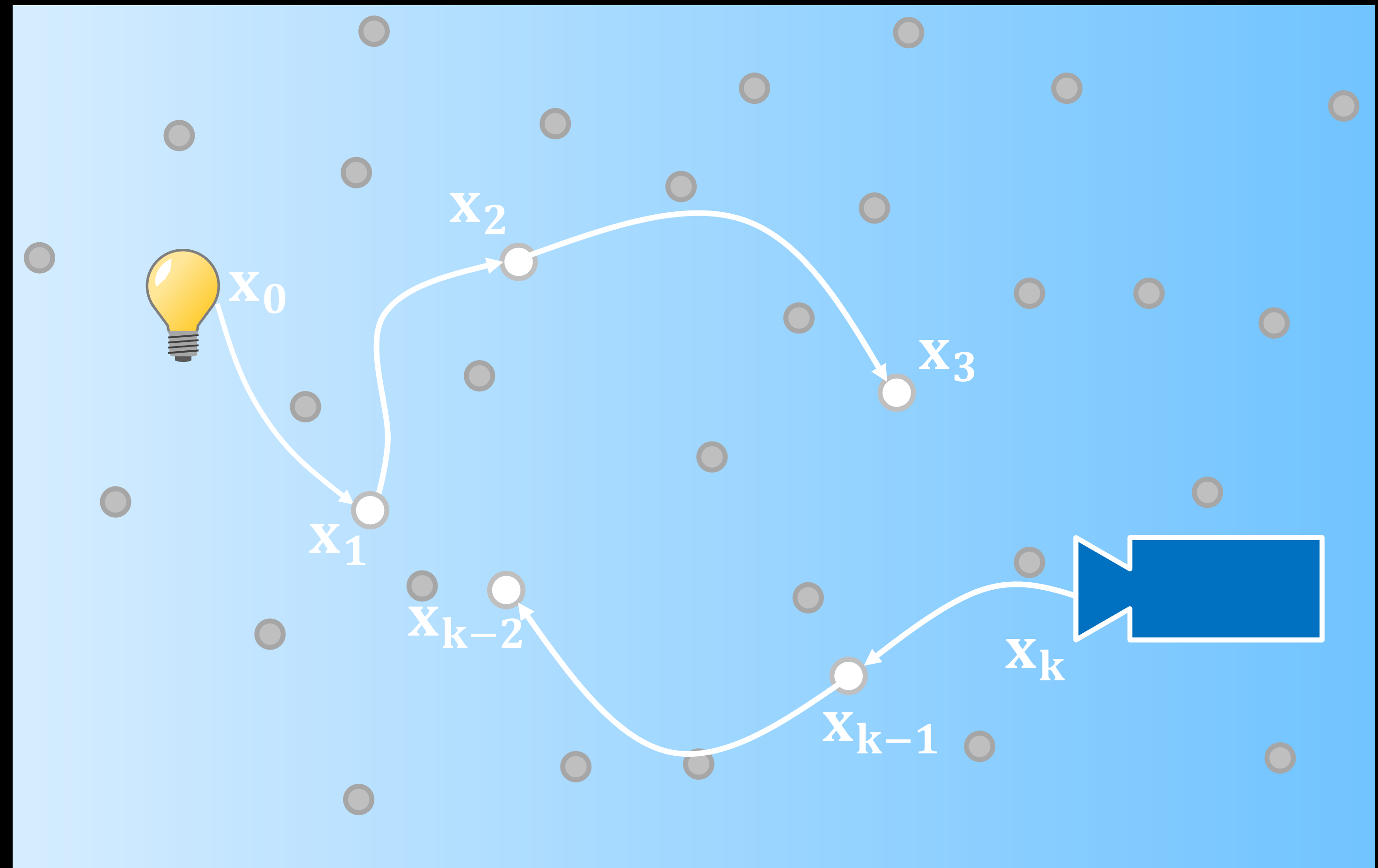


Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a straight line

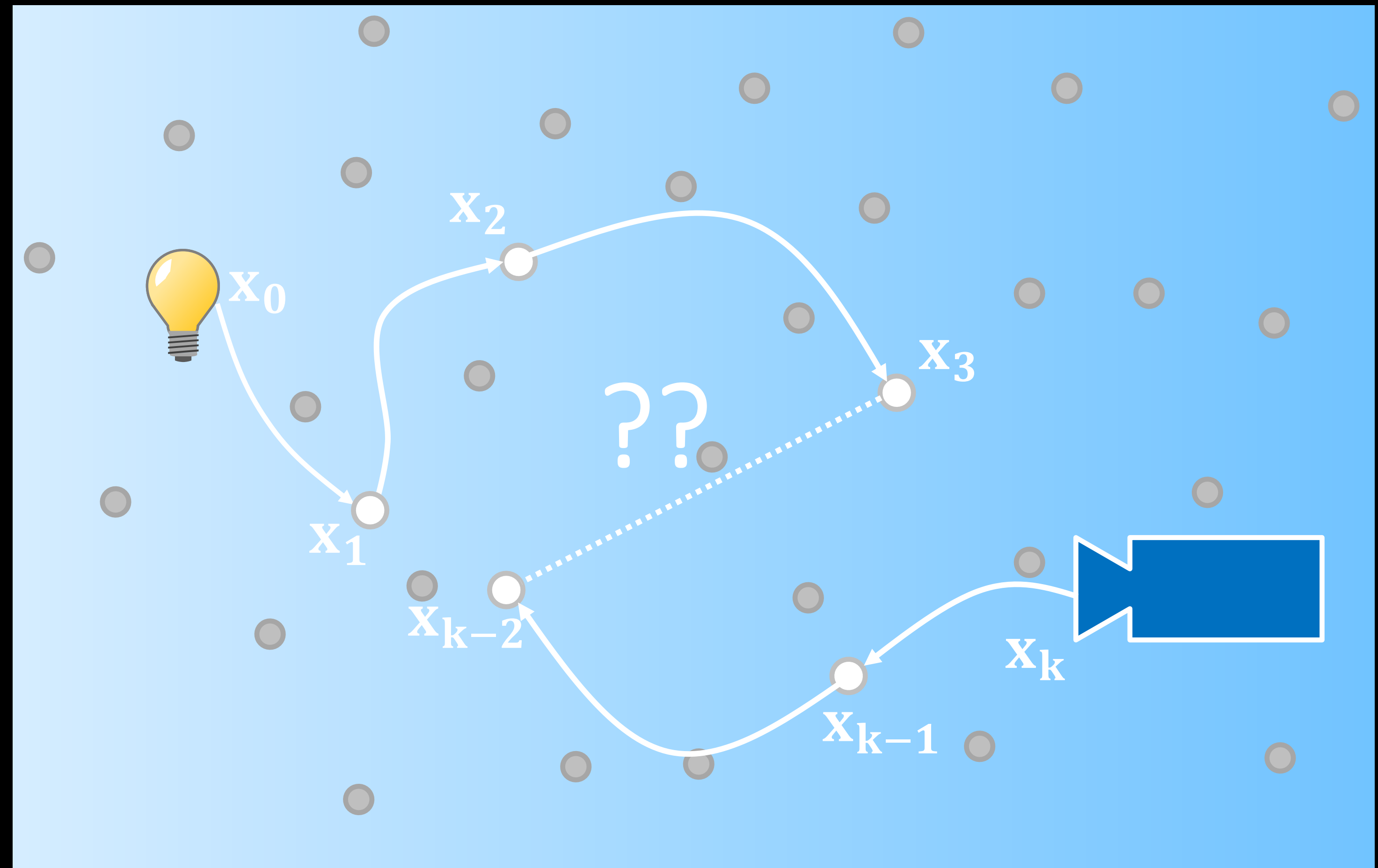


Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a ~~straight line~~ curve

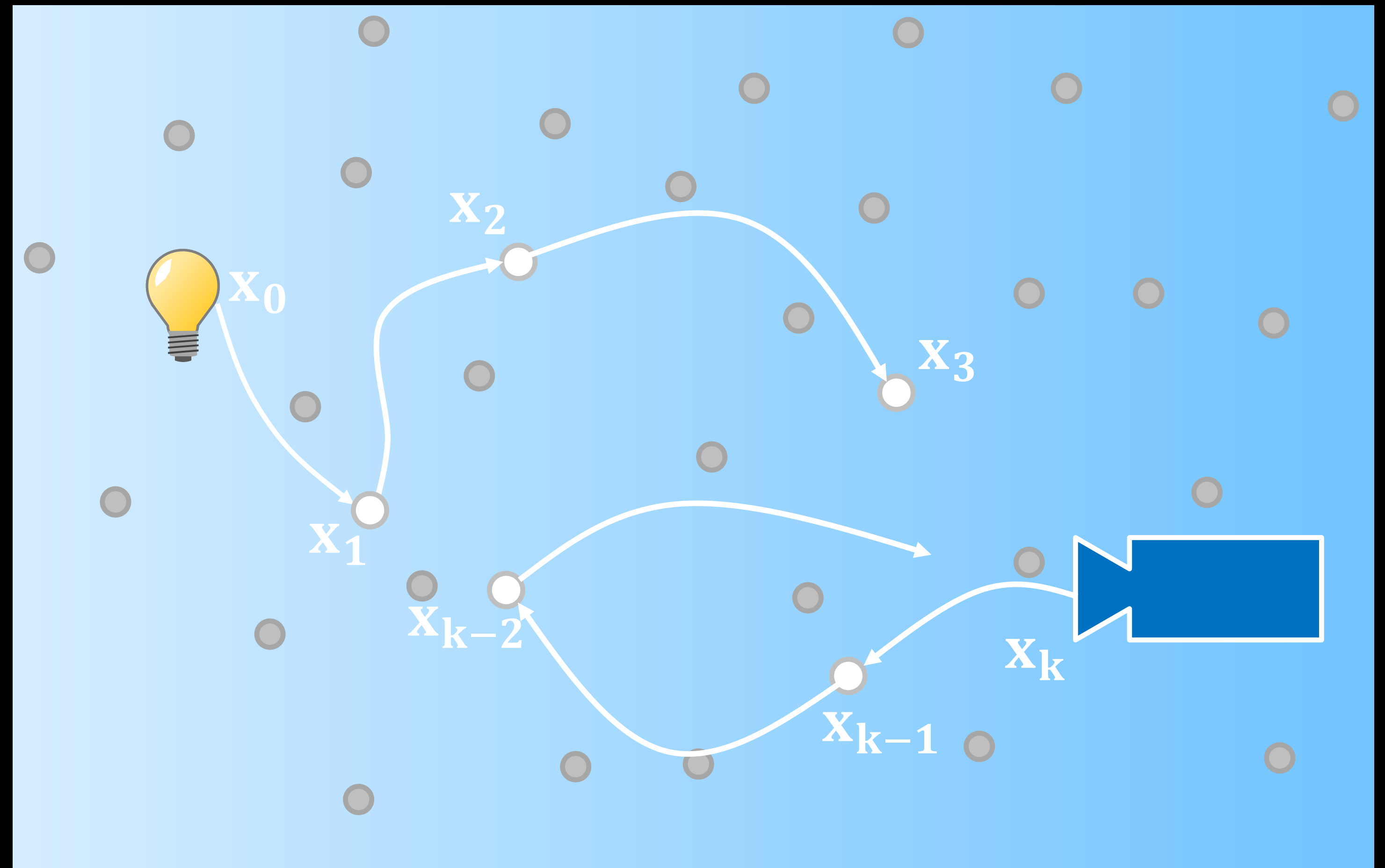


Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a ~~straight line~~ curve

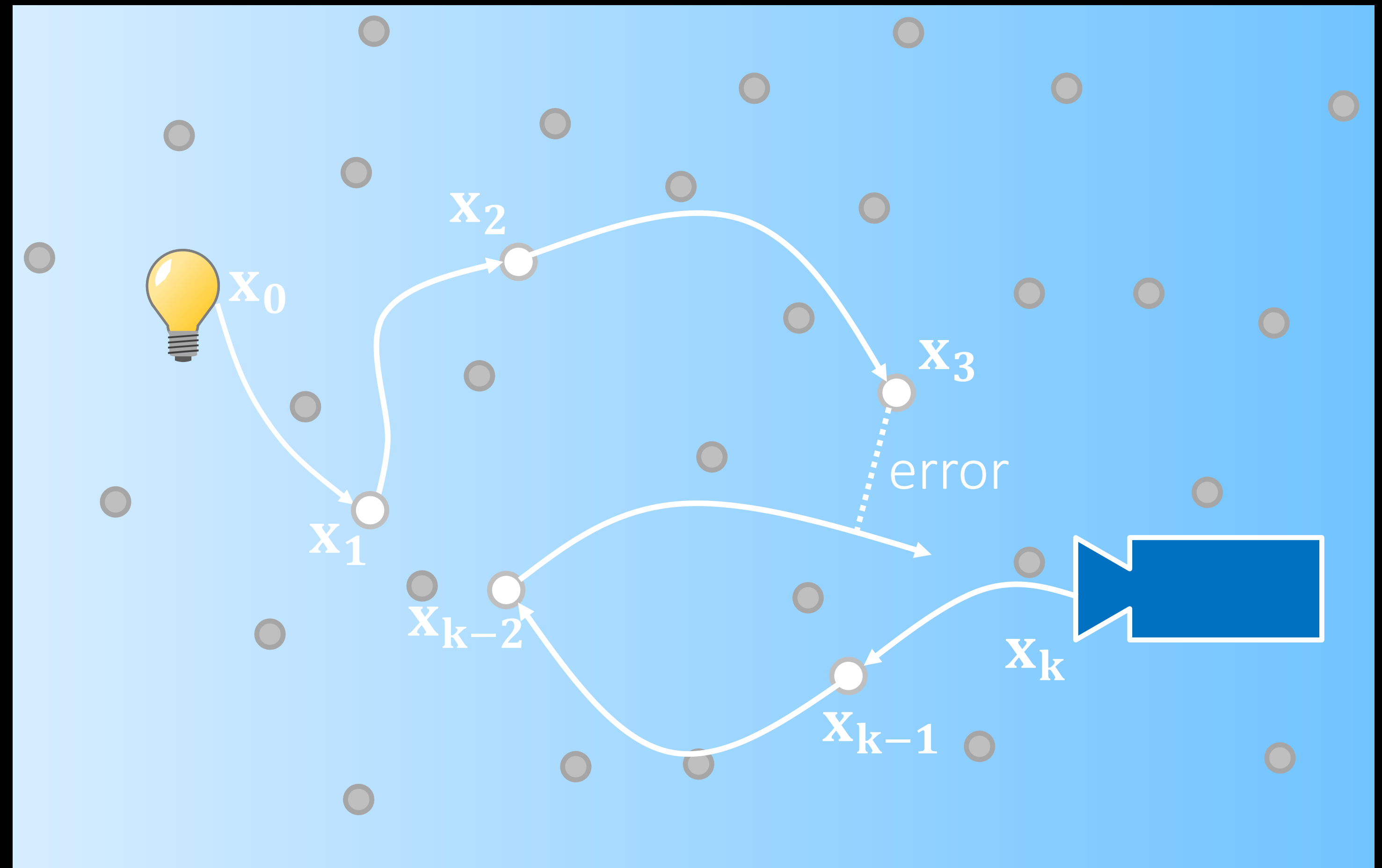


Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a ~~straight line~~ curve

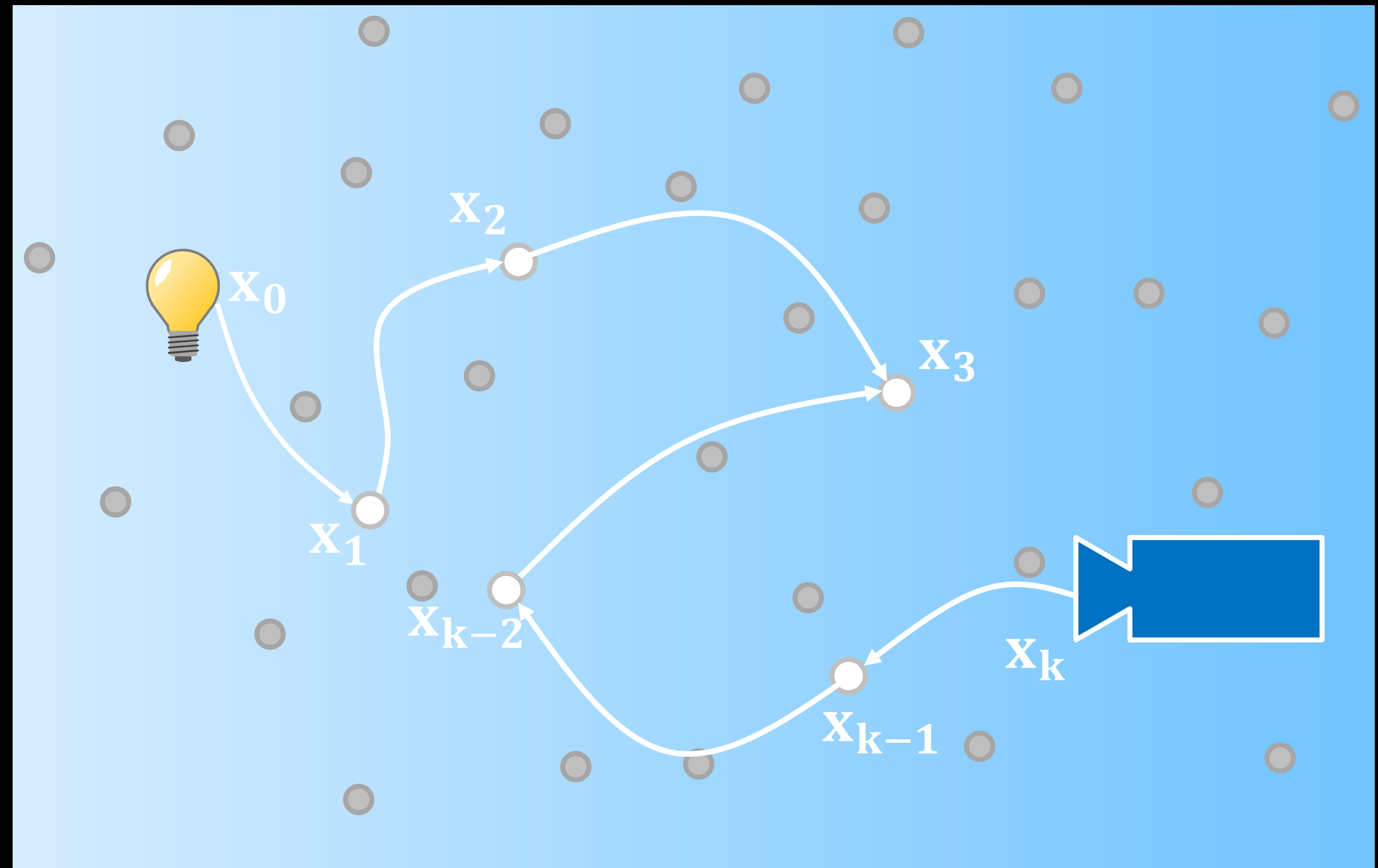


Rendering continuous refraction and scattering

radiative transfer equation

bidirectional path tracing (BDPT):

1. trace a random emitter subpath
2. trace a random sensor subpath
3. join vertices with a ~~straight line~~ curve



Rendering continuous refraction and scattering

$$\frac{d\mathcal{L}}{d\mathbf{v}_0} = (\mathbf{x}_{s^*} - \mathbf{y})^T \frac{d\mathbf{x}_{s^*}}{d\mathbf{v}_0},$$

$$\frac{d\mathbf{x}_{s^*}}{d\mathbf{v}_0} = \frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0} + \frac{\partial \mathbf{x}_{s^*}}{\partial s^*} \frac{ds^*}{d\mathbf{v}_0}.$$

$$g(\mathbf{x}_{s^*}) = 0 \Rightarrow \frac{dg(\mathbf{x}_{s^*})}{d\mathbf{x}_{s^*}} \frac{d\mathbf{x}_{s^*}}{d\mathbf{v}_0} = 0$$

$$\Rightarrow \frac{ds^*}{d\mathbf{v}_0} = - \frac{\frac{dg(\mathbf{x}_{s^*})}{d\mathbf{x}_{s^*}} \frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0}}{\frac{dg(\mathbf{x}_{s^*})}{d\mathbf{x}_{s^*}} \frac{\partial \mathbf{x}_{s^*}}{\partial s^*}},$$

$$\frac{d\mathbf{x}_{s^*}}{d\mathbf{v}_0} = \left(I_{3 \times 3} - \frac{\frac{\partial \mathbf{x}_{s^*}}{\partial s^*} \frac{dg(\mathbf{x}_{s^*})}{d\mathbf{x}_{s^*}}}{\frac{dg(\mathbf{x}_{s^*})}{d\mathbf{x}_{s^*}} \frac{\partial \mathbf{x}_{s^*}}{\partial s^*}} \right) \frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0}.$$

Algorithm 2: Symplectic integration for derivative tracing

Input: $n(\mathbf{x}), \nabla n(\mathbf{x}), H_n(\mathbf{x}), \text{ray}(\mathbf{x}, \mathbf{v}), \frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0} = O_3, \frac{\partial \mathbf{v}_{s^*}}{\partial \mathbf{v}_0} = I_3,$

$nSteps, s = \text{step size}$

Output: $\frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0}, \frac{\partial \mathbf{v}_{s^*}}{\partial \mathbf{v}_0}$

for $i = 1 : nSteps$ **do**

$\text{ray.v}+ = 0.5s \nabla n(\text{ray.x});$

$\frac{\partial \mathbf{v}_{s^*}}{\partial \mathbf{v}_0} + = 0.5s H_n(\text{ray.x}) \frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0};$

$\text{ray.x}+ = s \frac{\text{ray.v}}{n(\text{ray.x})};$

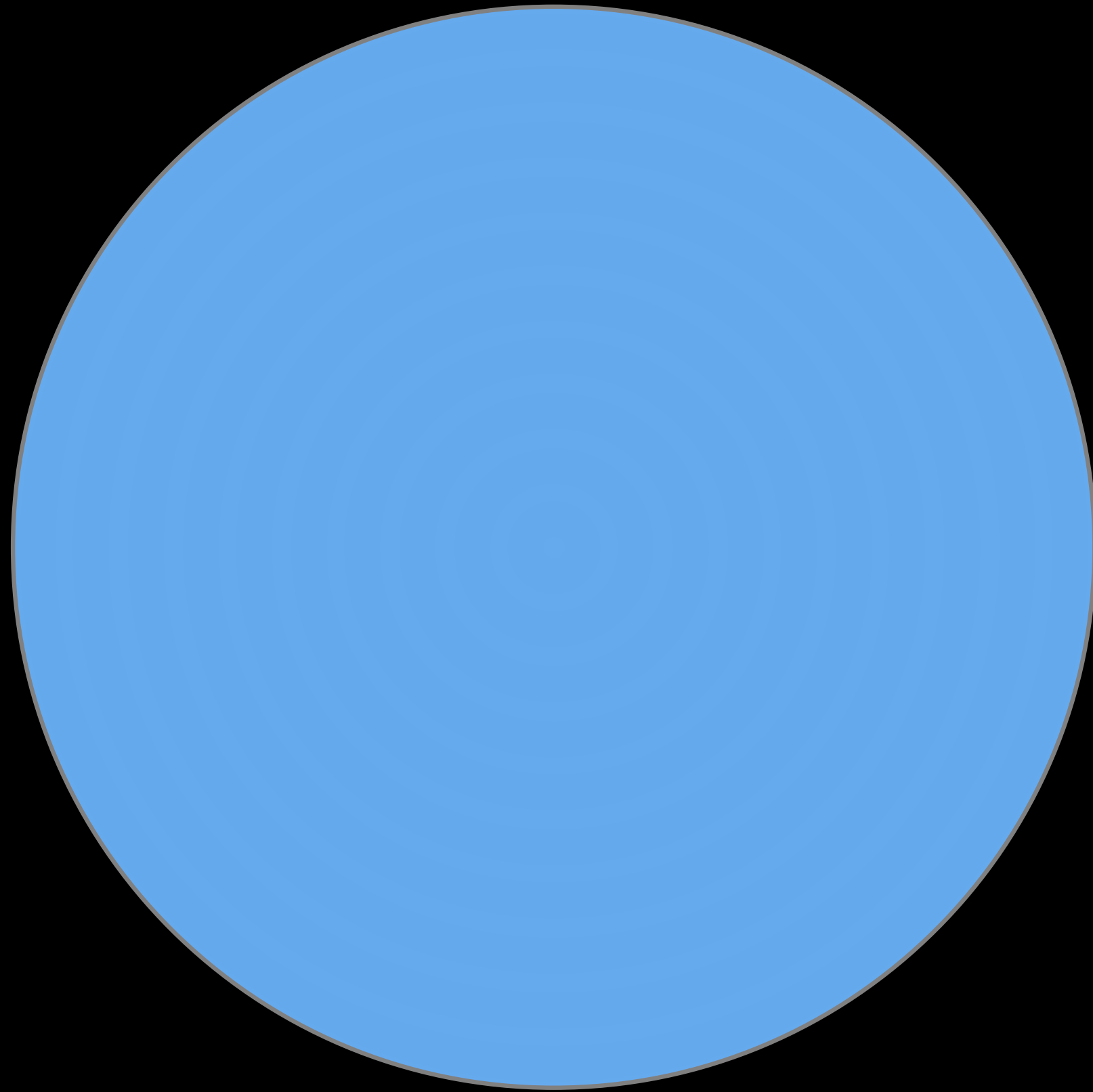
$\frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0} + = s \left(- \frac{\mathbf{v}_s \nabla n(\text{ray.x})}{n(\text{ray.x})^2} \frac{\partial \mathbf{x}_s}{\partial \mathbf{v}_0} + \frac{1}{n(\text{ray.x})} \frac{\partial \mathbf{v}_s}{\partial \mathbf{v}_0} \right);$

$\text{ray.v}+ = 0.5s \nabla n(\text{ray.x});$

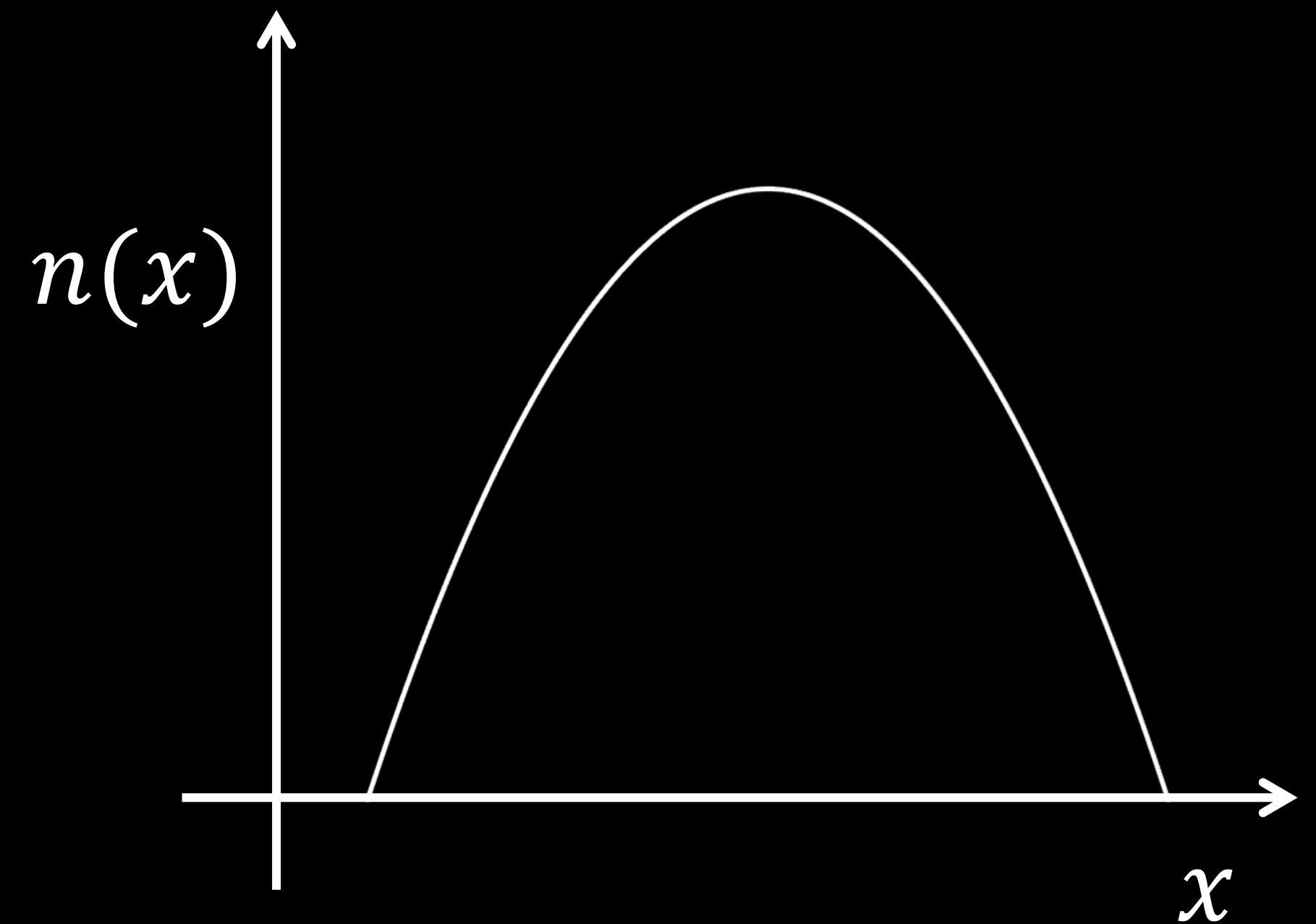
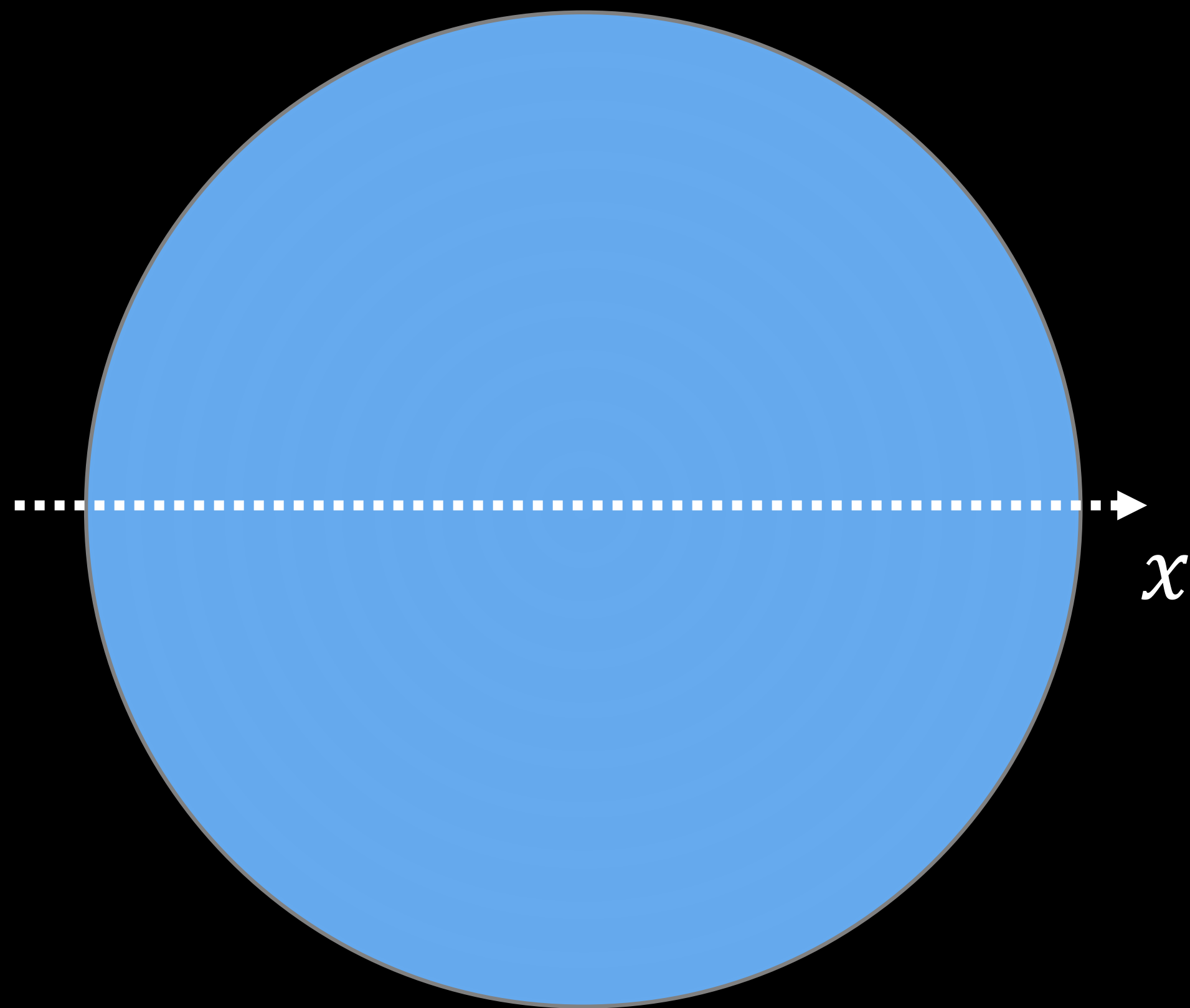
$\frac{\partial \mathbf{v}_{s^*}}{\partial \mathbf{v}_0} + = 0.5s H_n(\text{ray.x}) \frac{\partial \mathbf{x}_{s^*}}{\partial \mathbf{v}_0};$

end

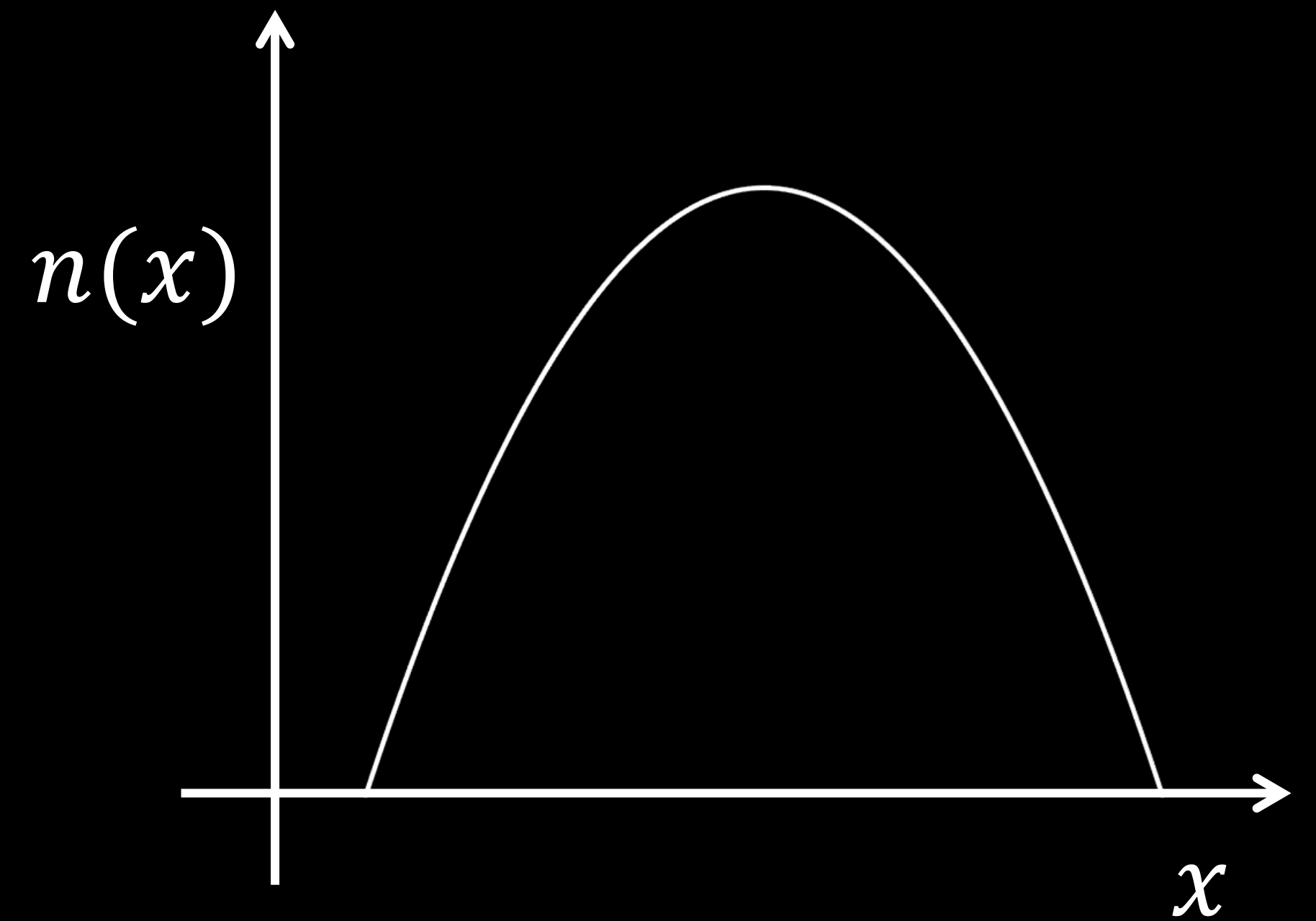
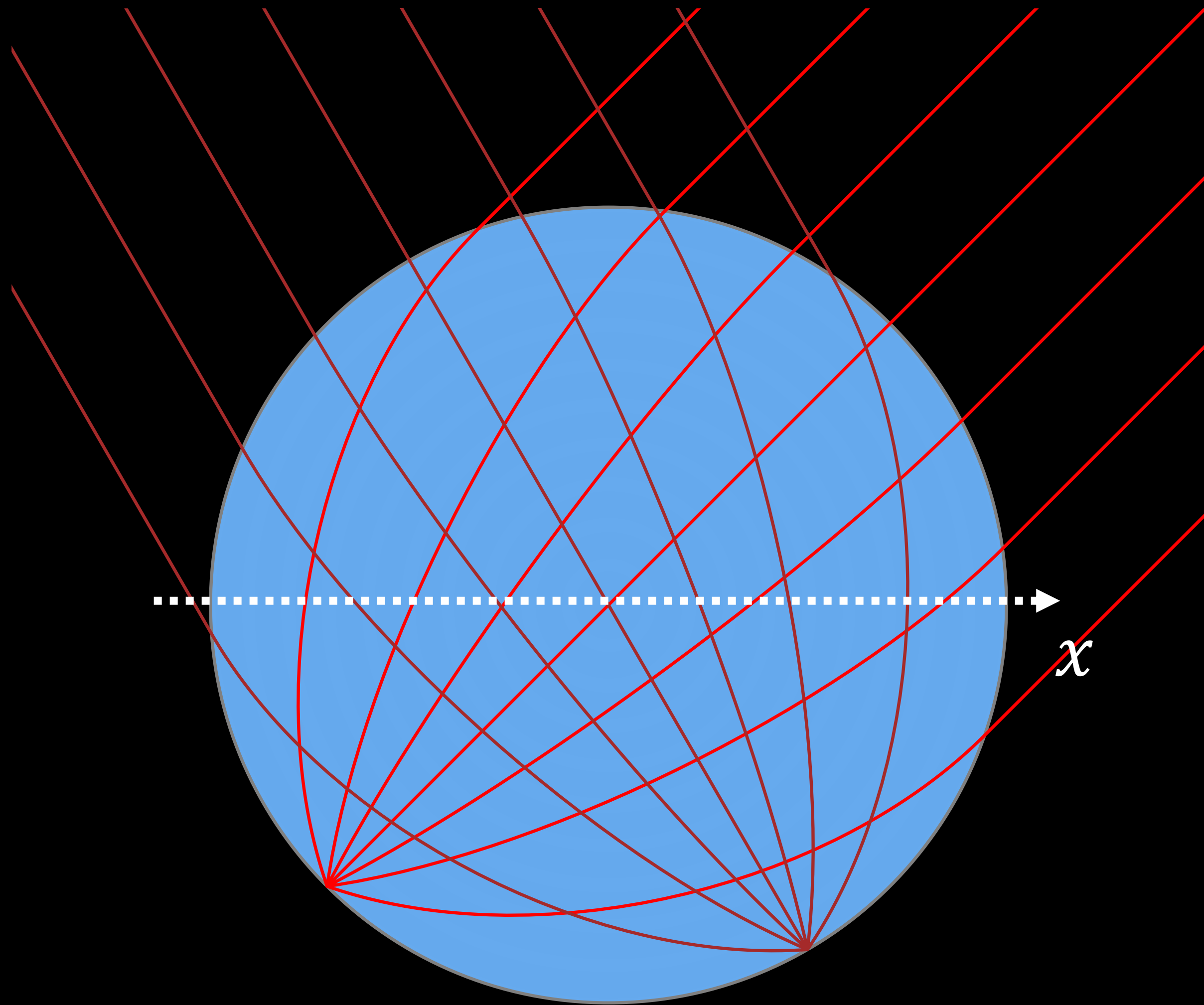
Application: simulate Luneburg lenses



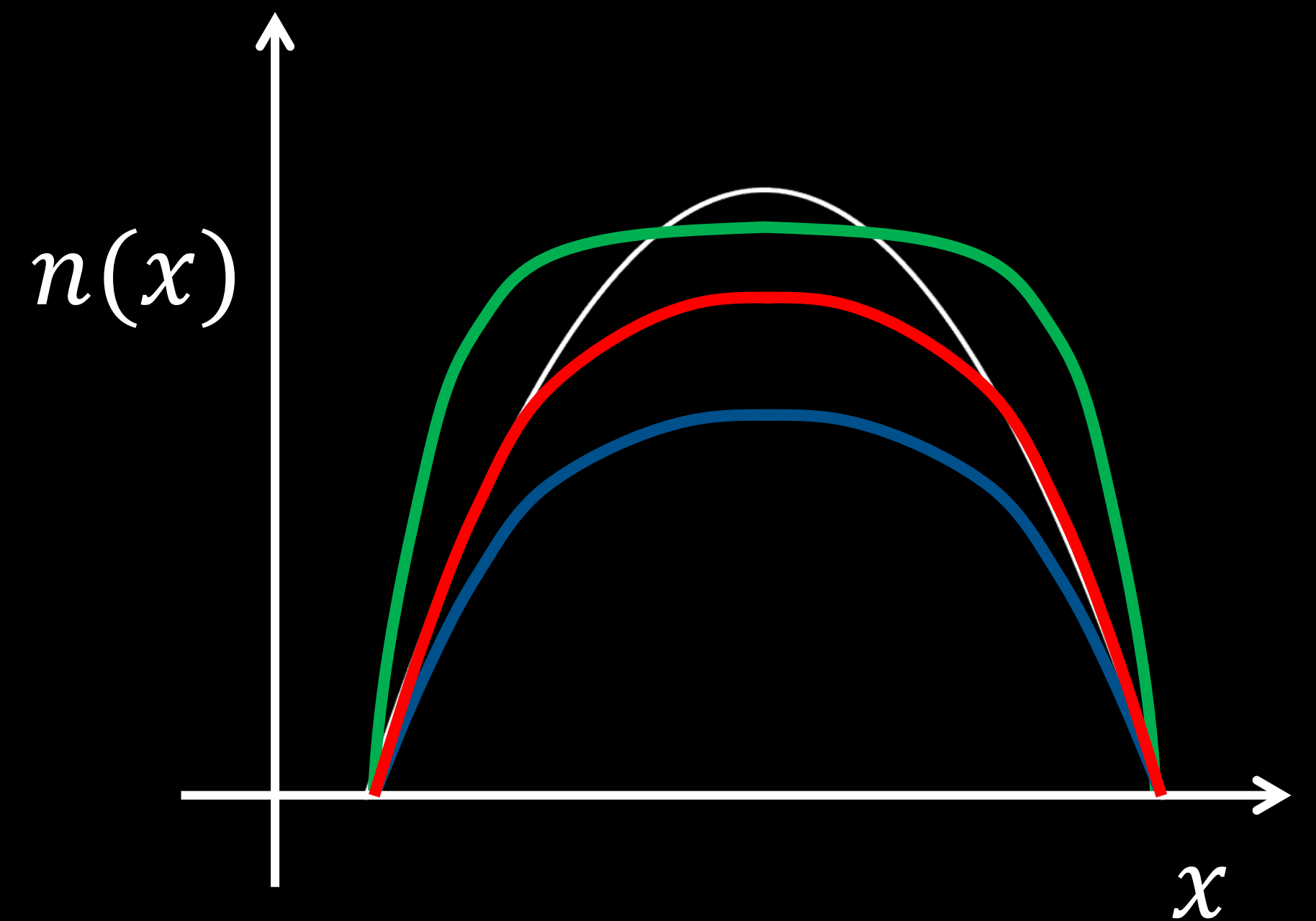
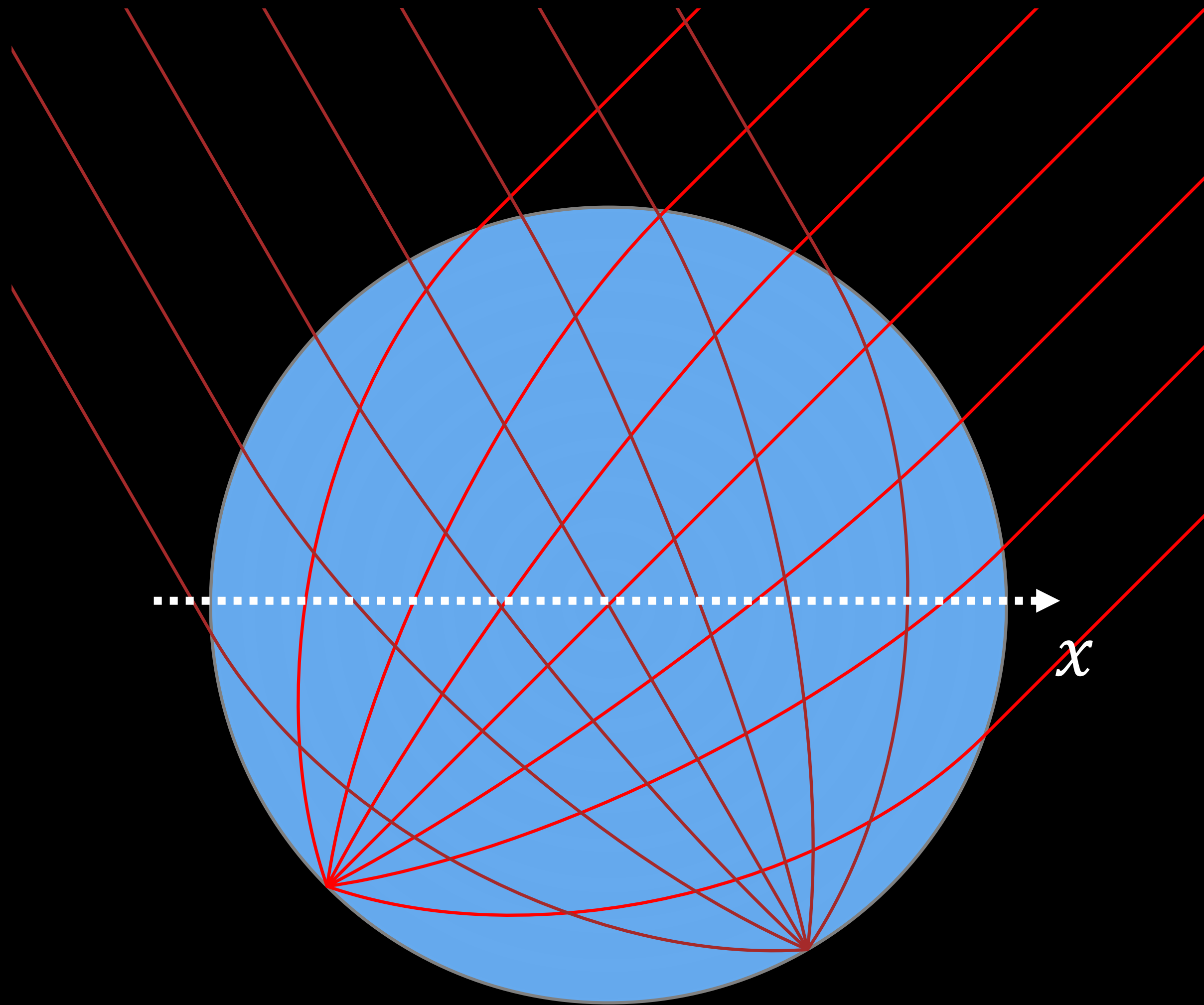
Application: simulate Luneburg lenses



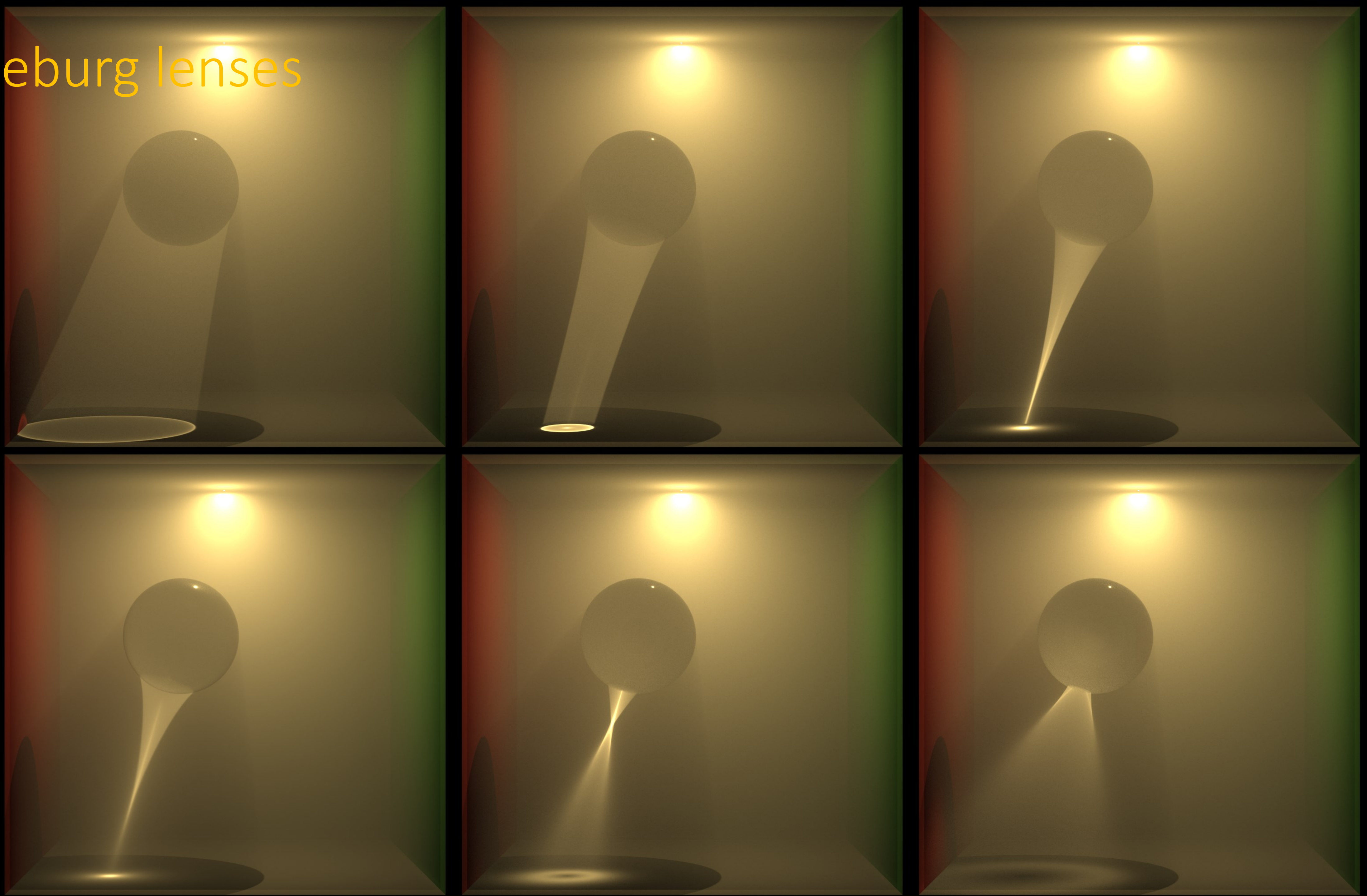
Application: simulate Luneburg lenses



Application: simulate Luneburg lenses



Luneburg lenses



Application: transient rendering

constant refractive index

continuous refractive index

Application: transient rendering

constant refractive index

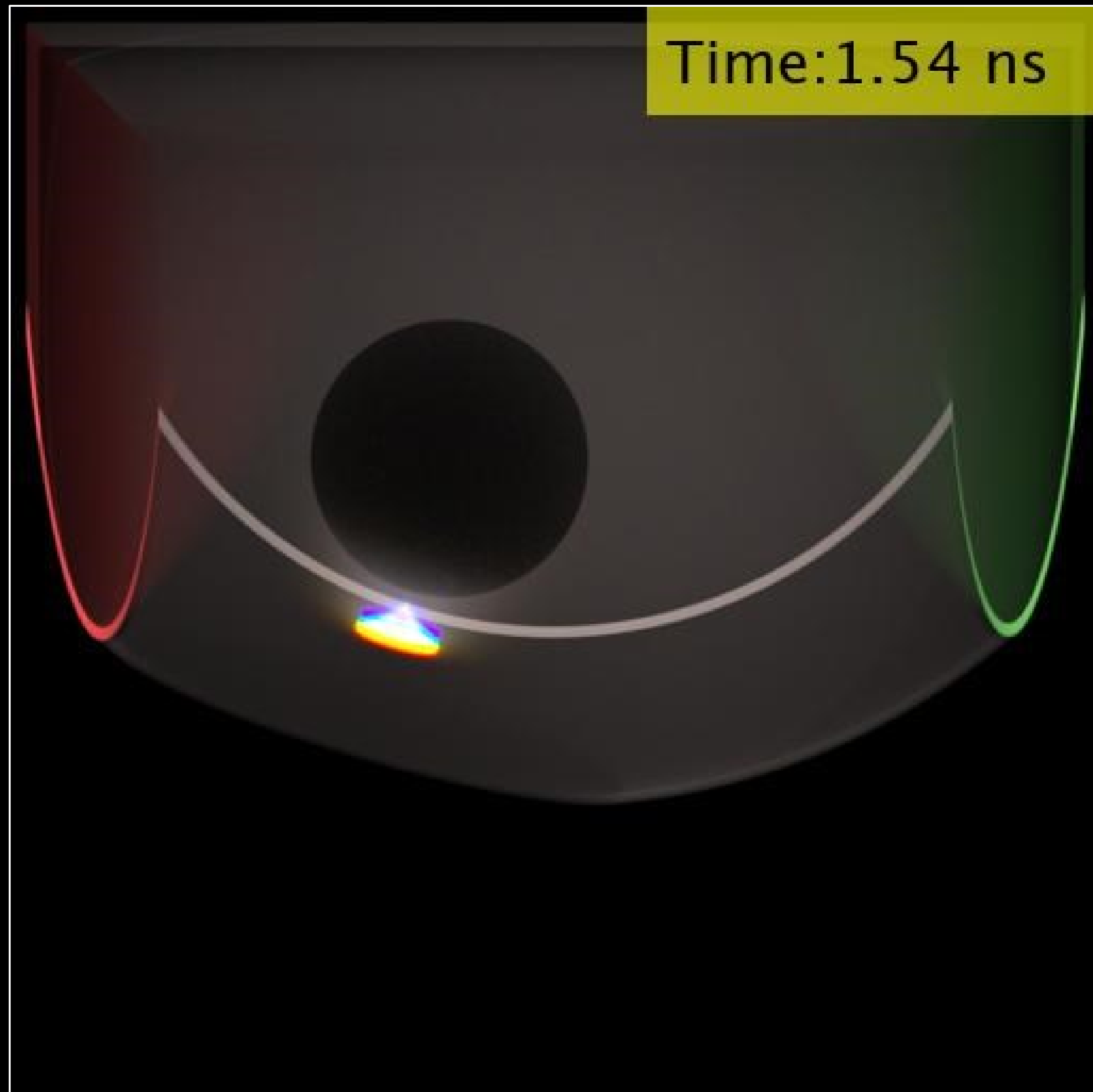
Time:0.02 ns

continuous refractive index

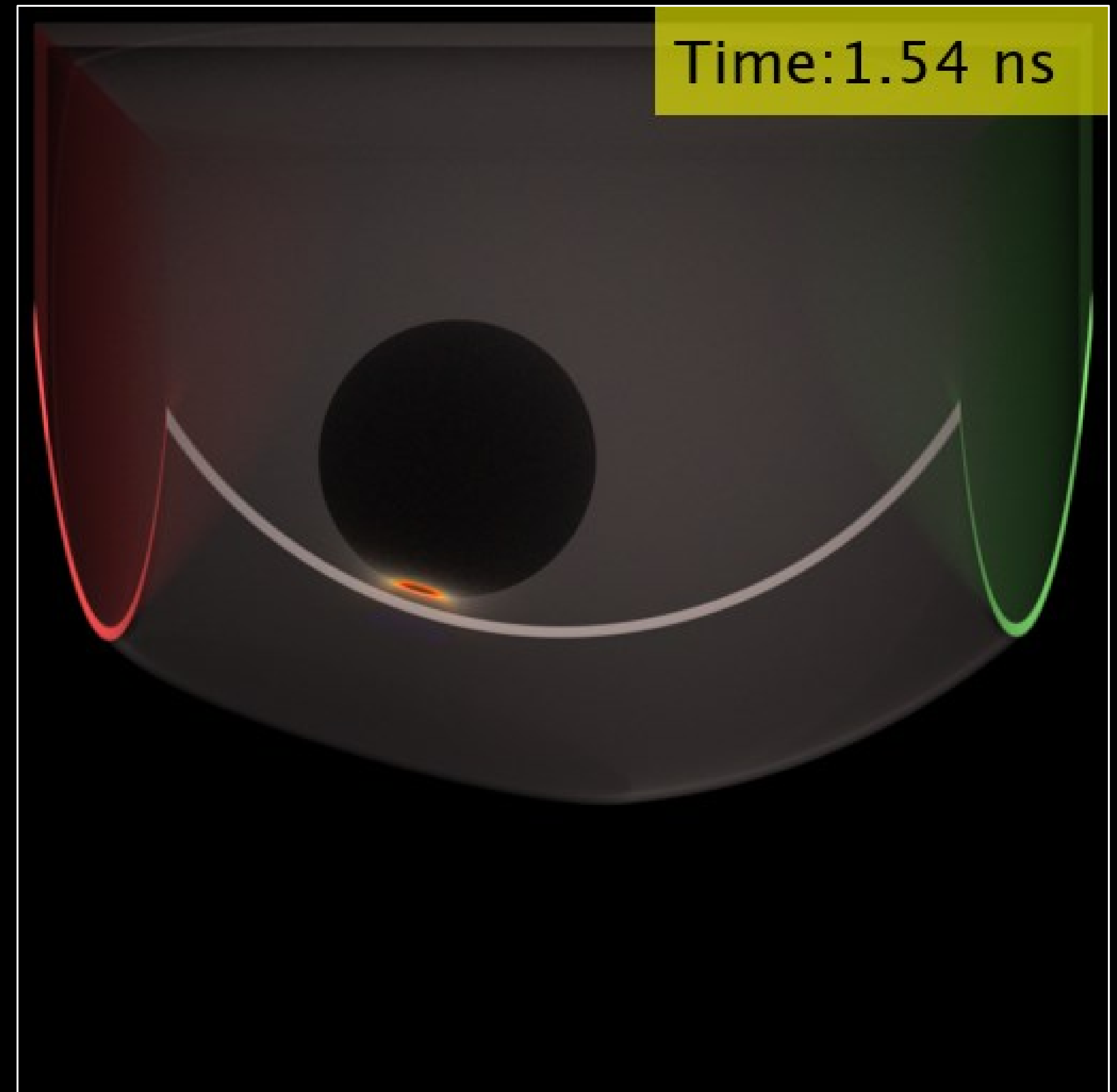
Time:0.02 ns

Application: transient rendering

constant refractive index



continuous refractive index

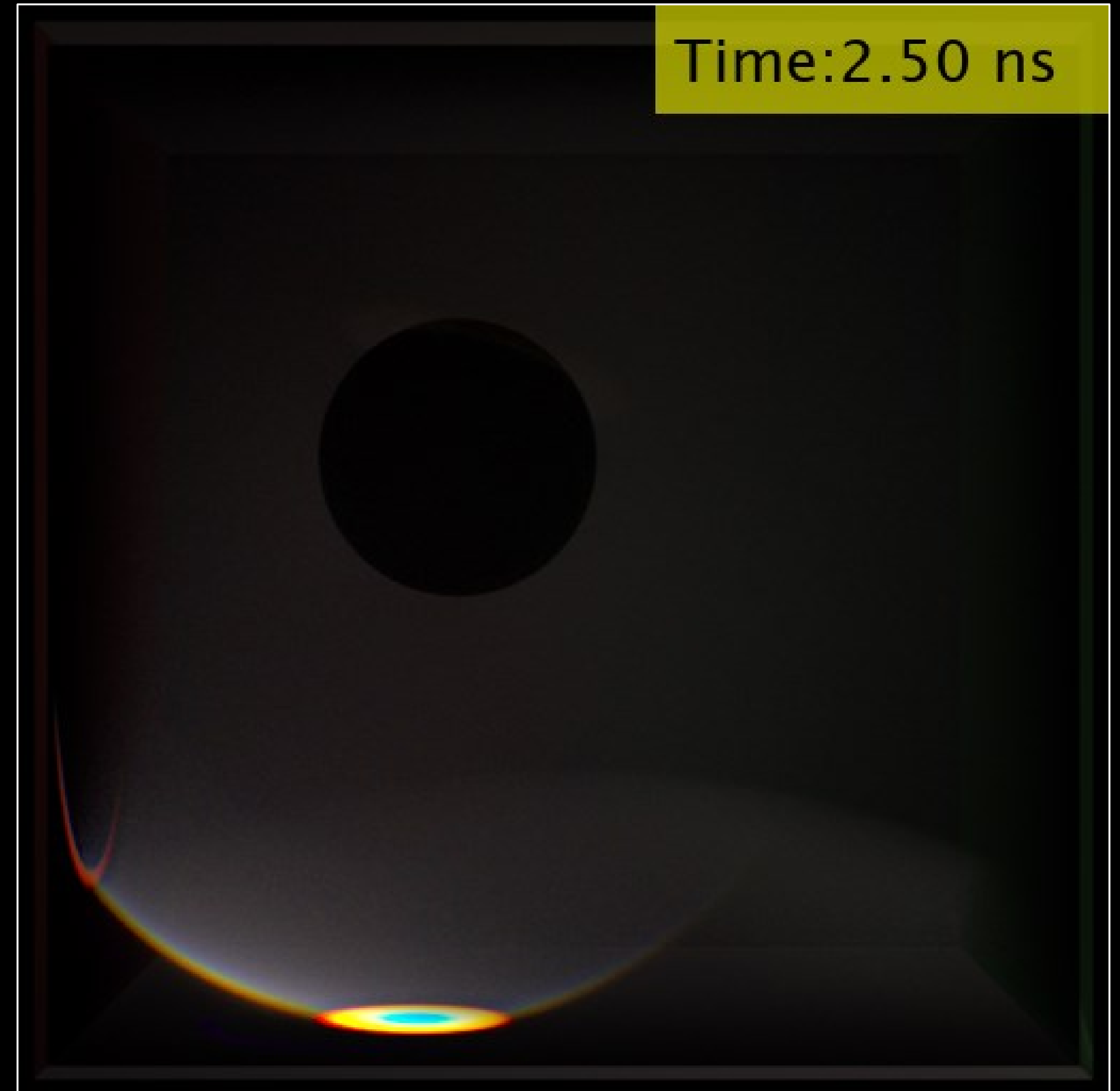


Application: transient rendering

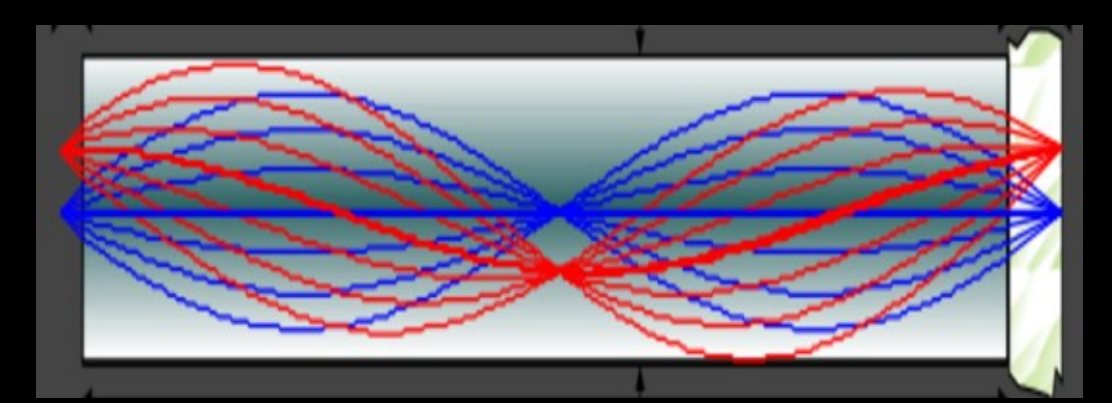
constant refractive index



continuous refractive index



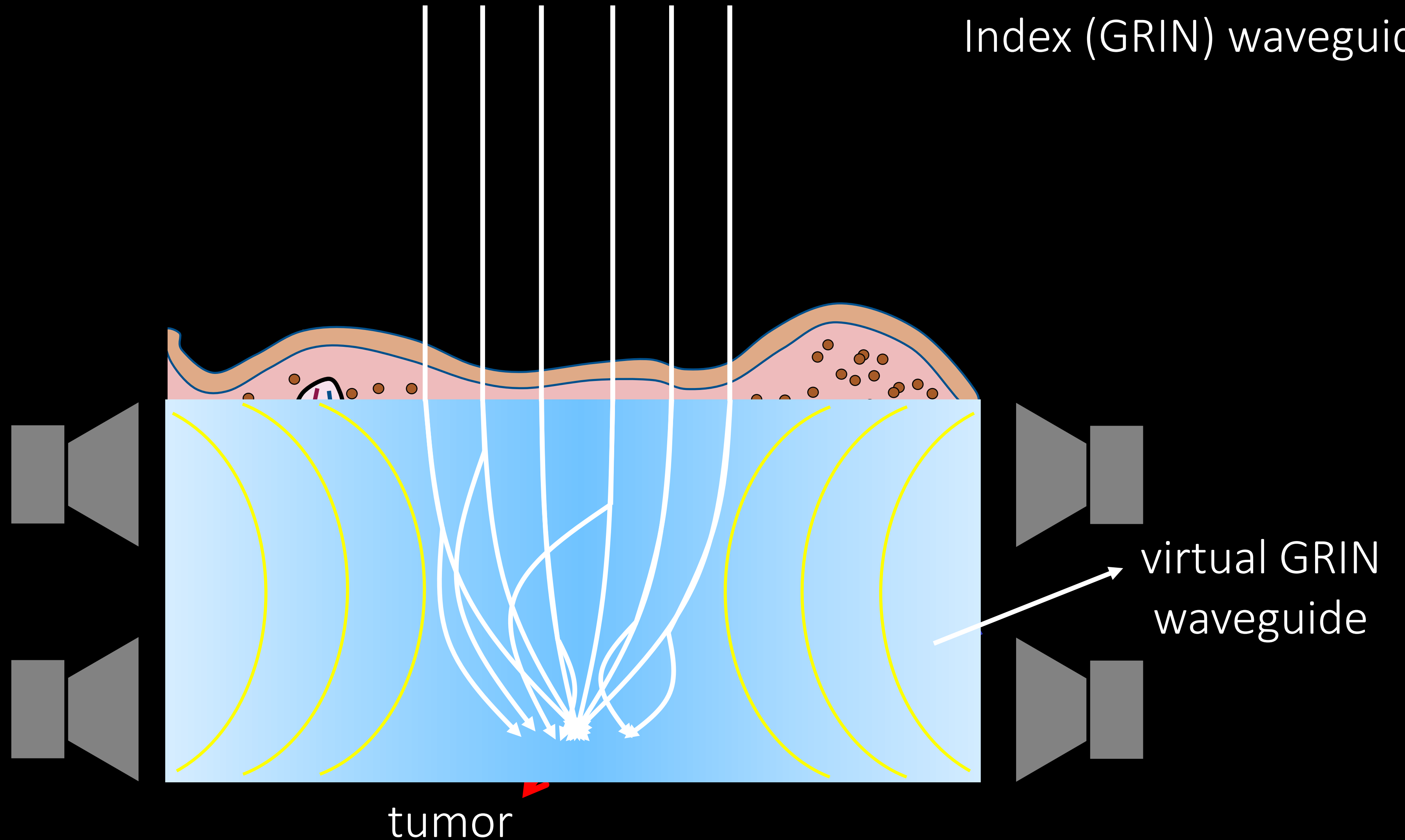
Application: focusing light inside tissue



Gradient Refractive Index (GRIN) waveguide

High-dimensional, highly-non-linear design problem:

- ultrasound frequency
- ultrasound voltage
- placement of transducers
- waveform shape
- and more...



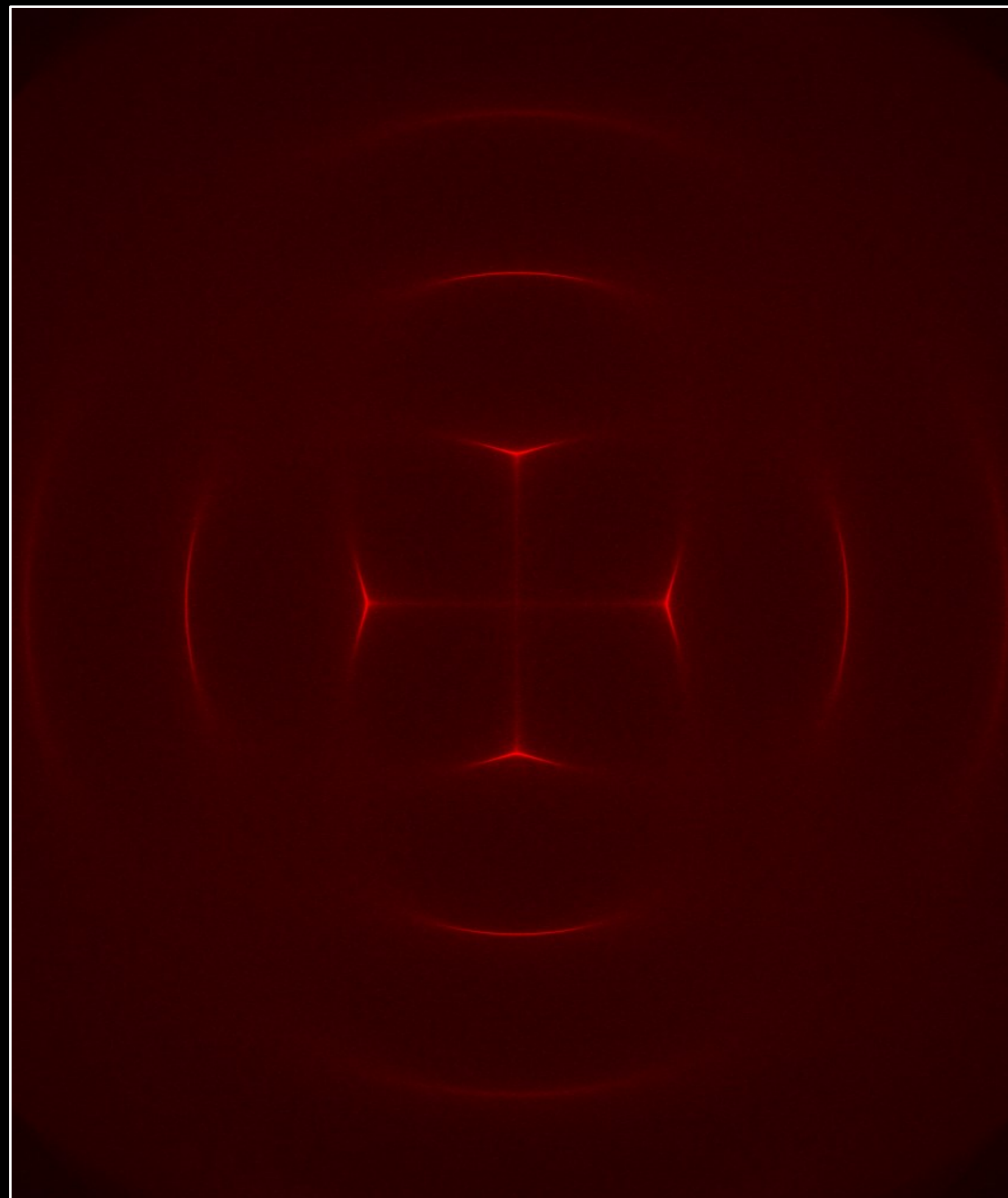
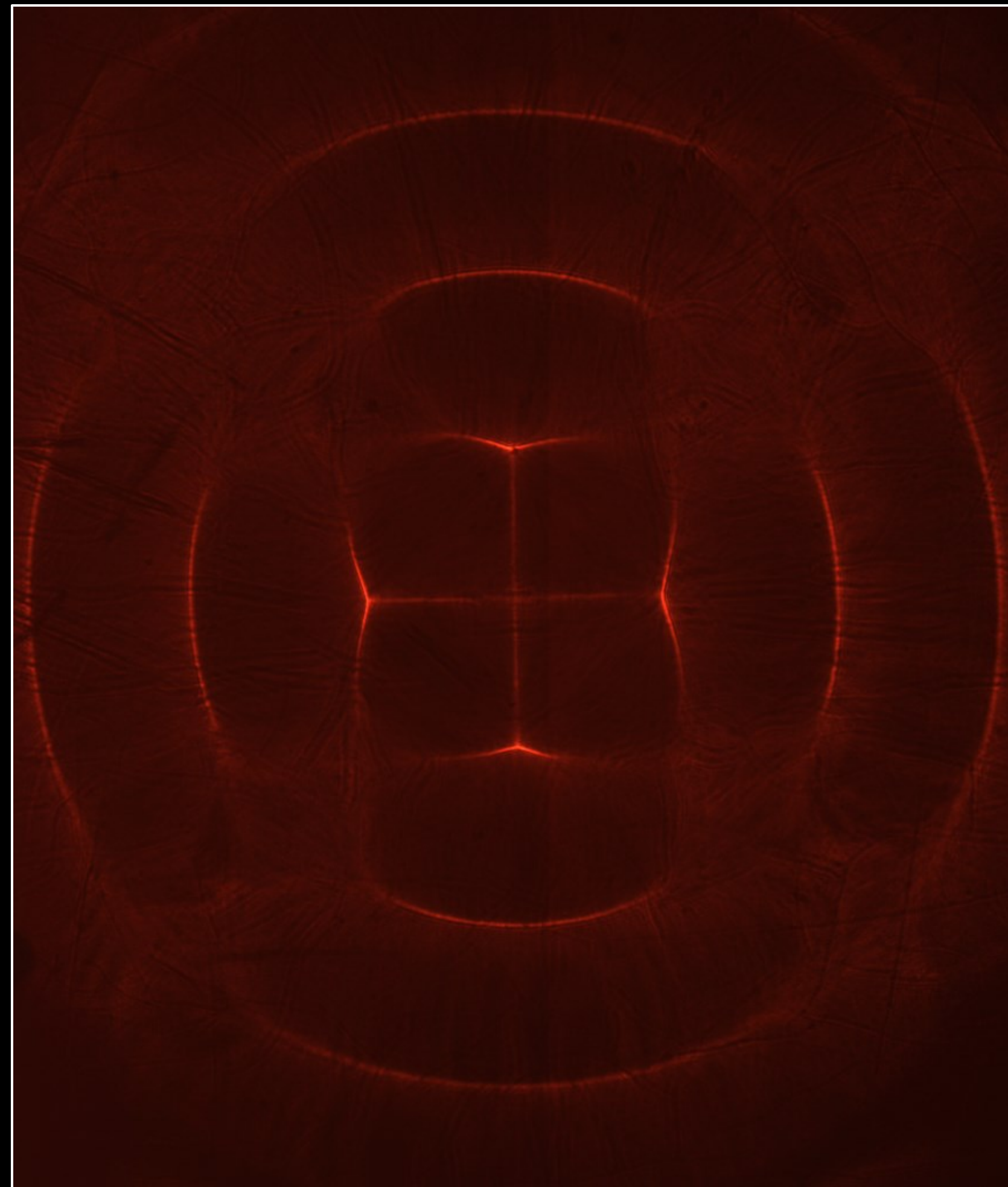
Efficiently explore using rendering

Rendering virtual ultrasonic waveguides

real measurement

BDPT
(our technique)

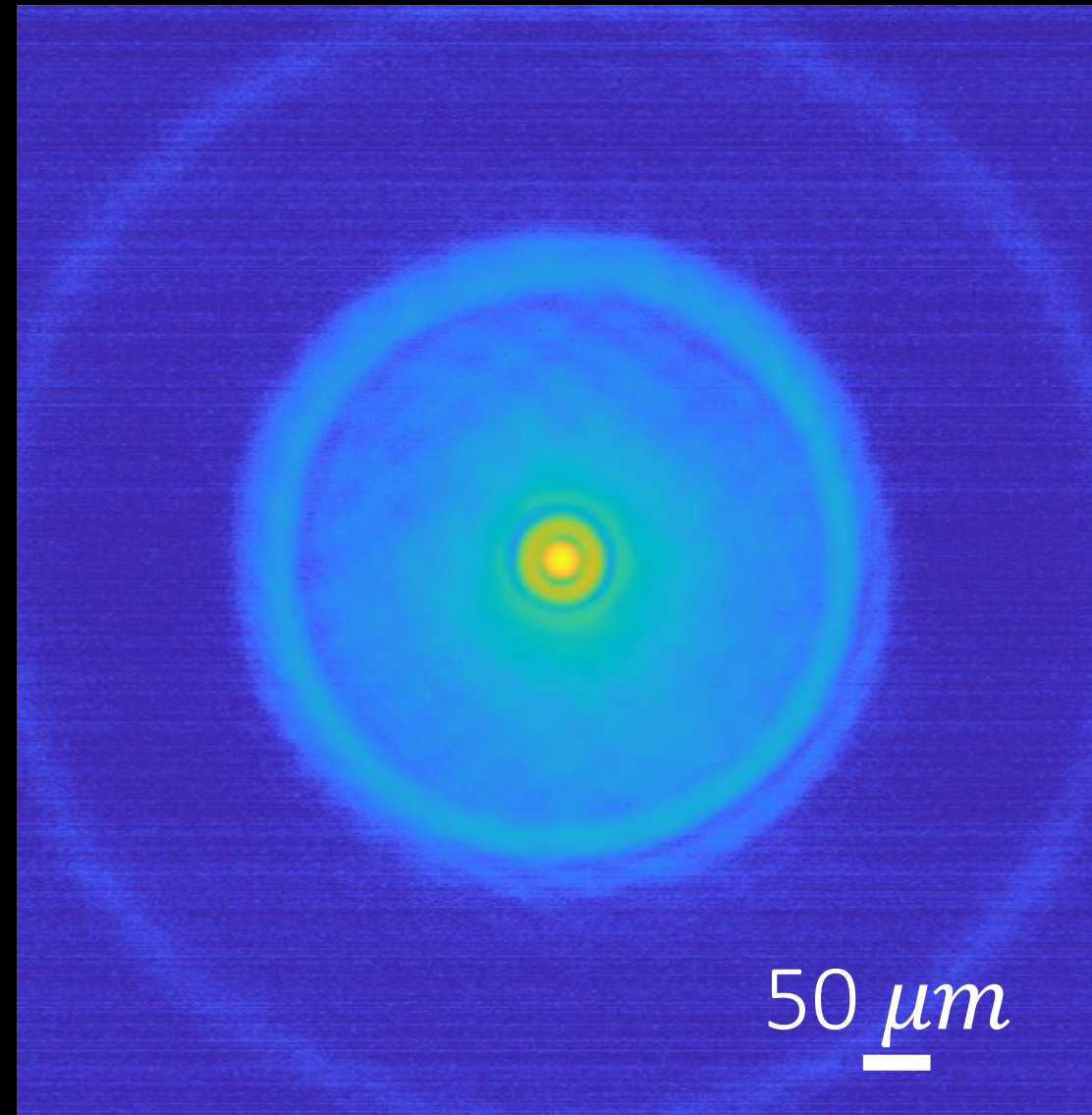
photon mapping
(previous technique)



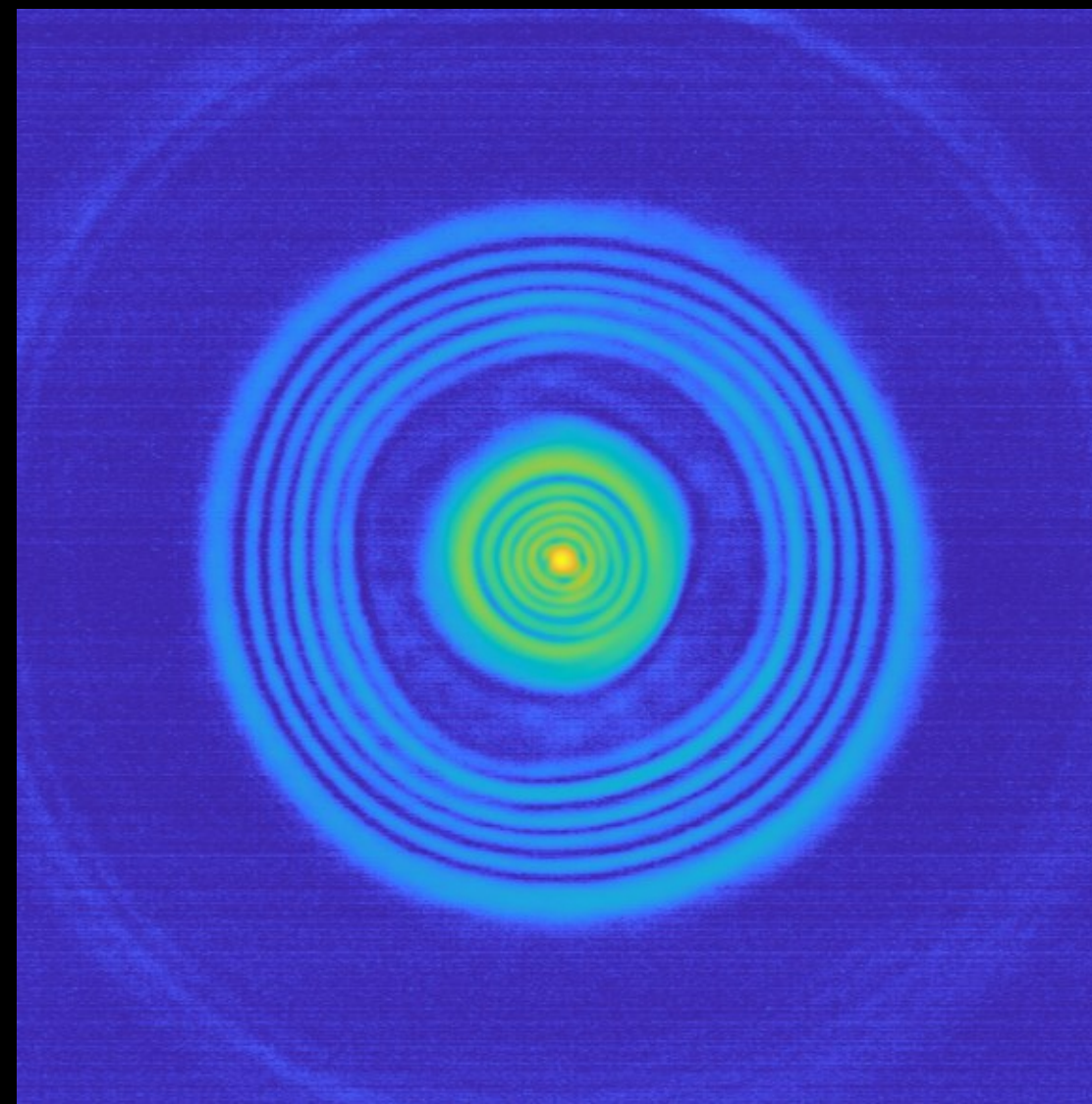
Validation of simulated data

experimental data

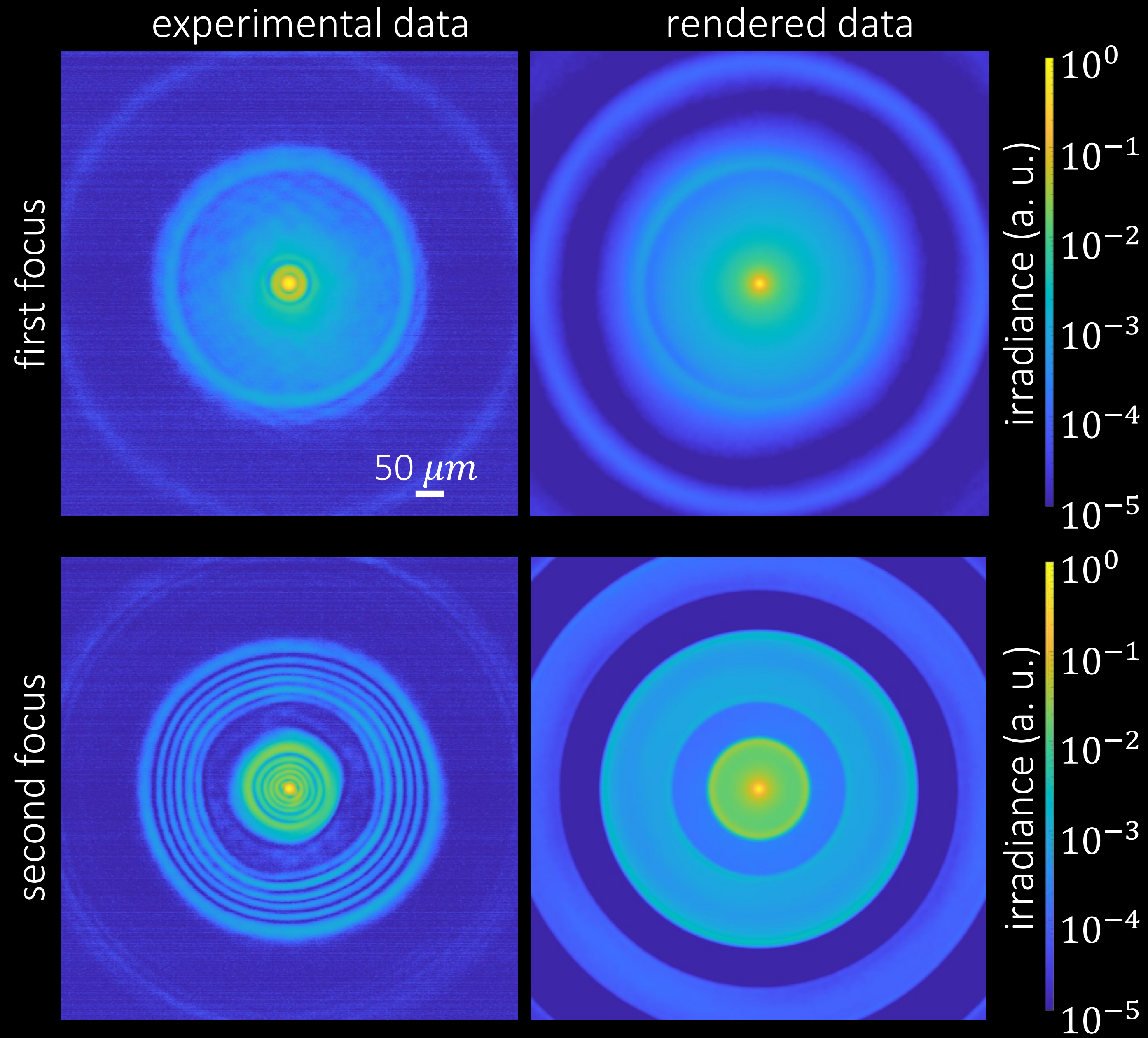
first focus



second focus



Validation of simulated data

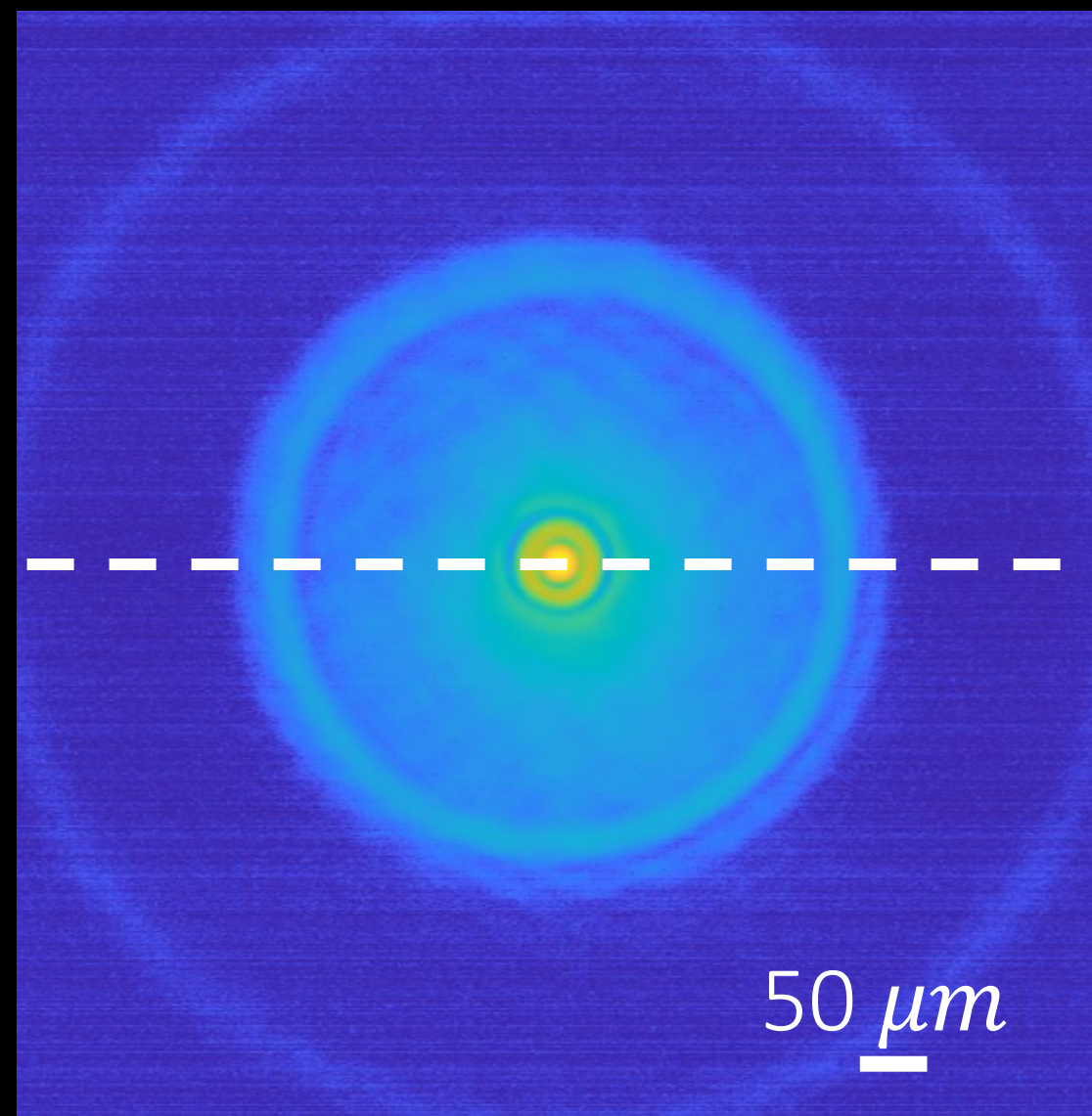


Validation of simulated data

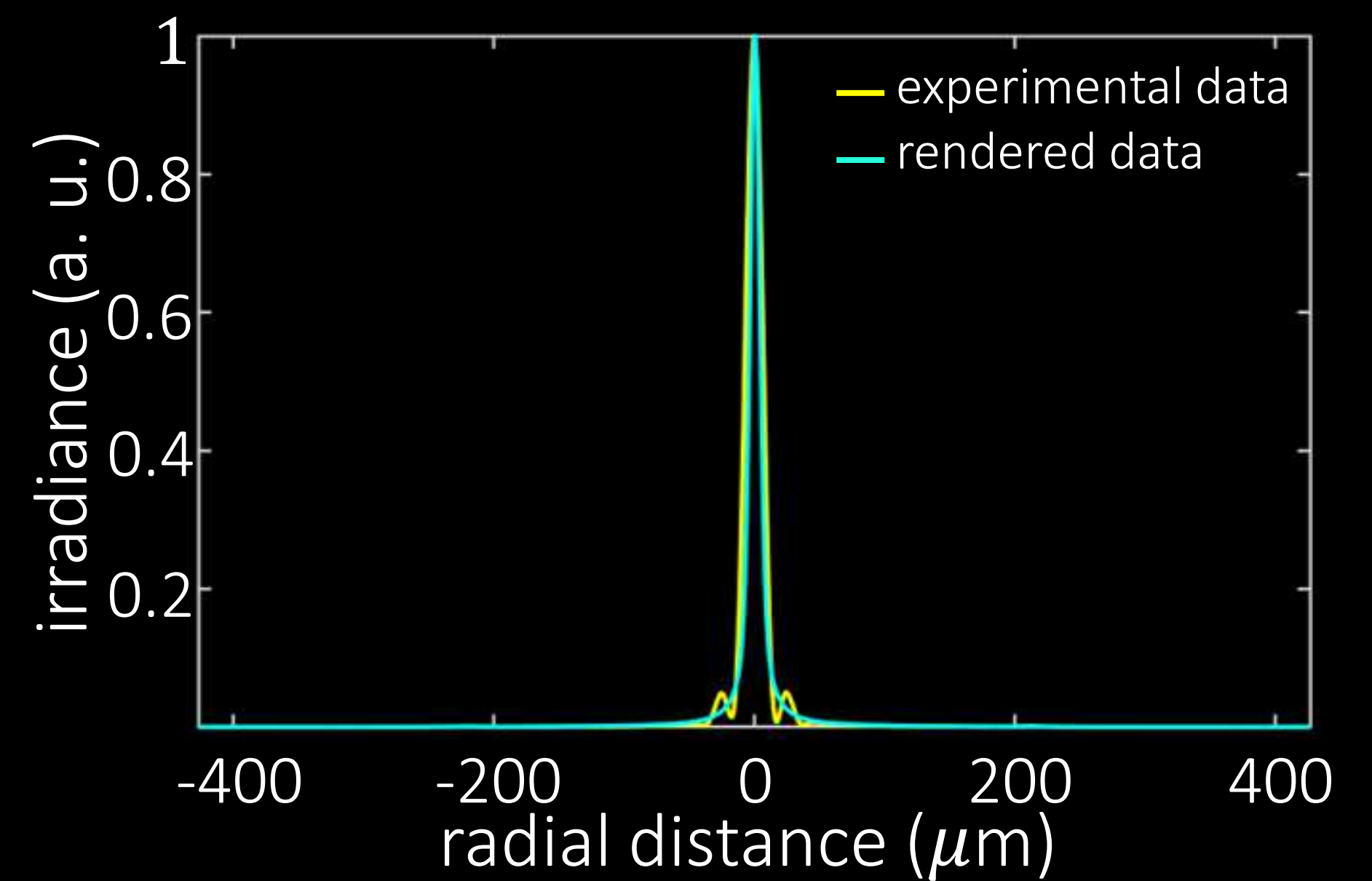
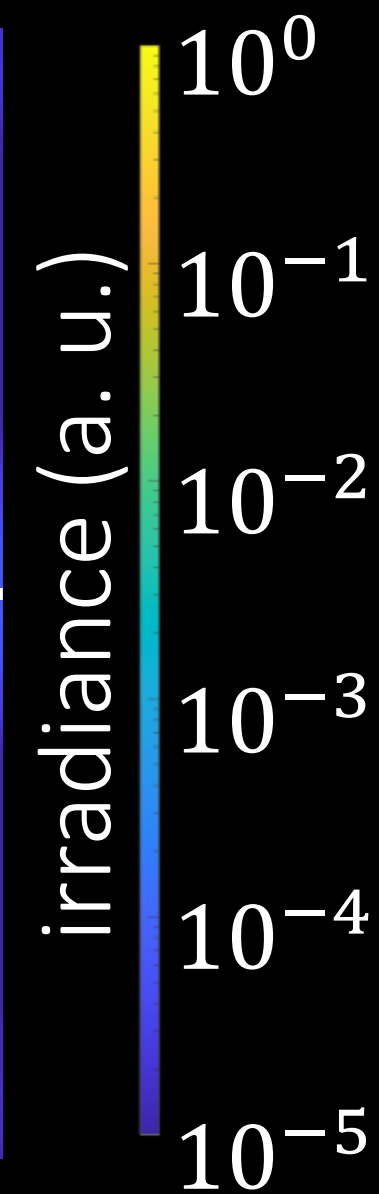
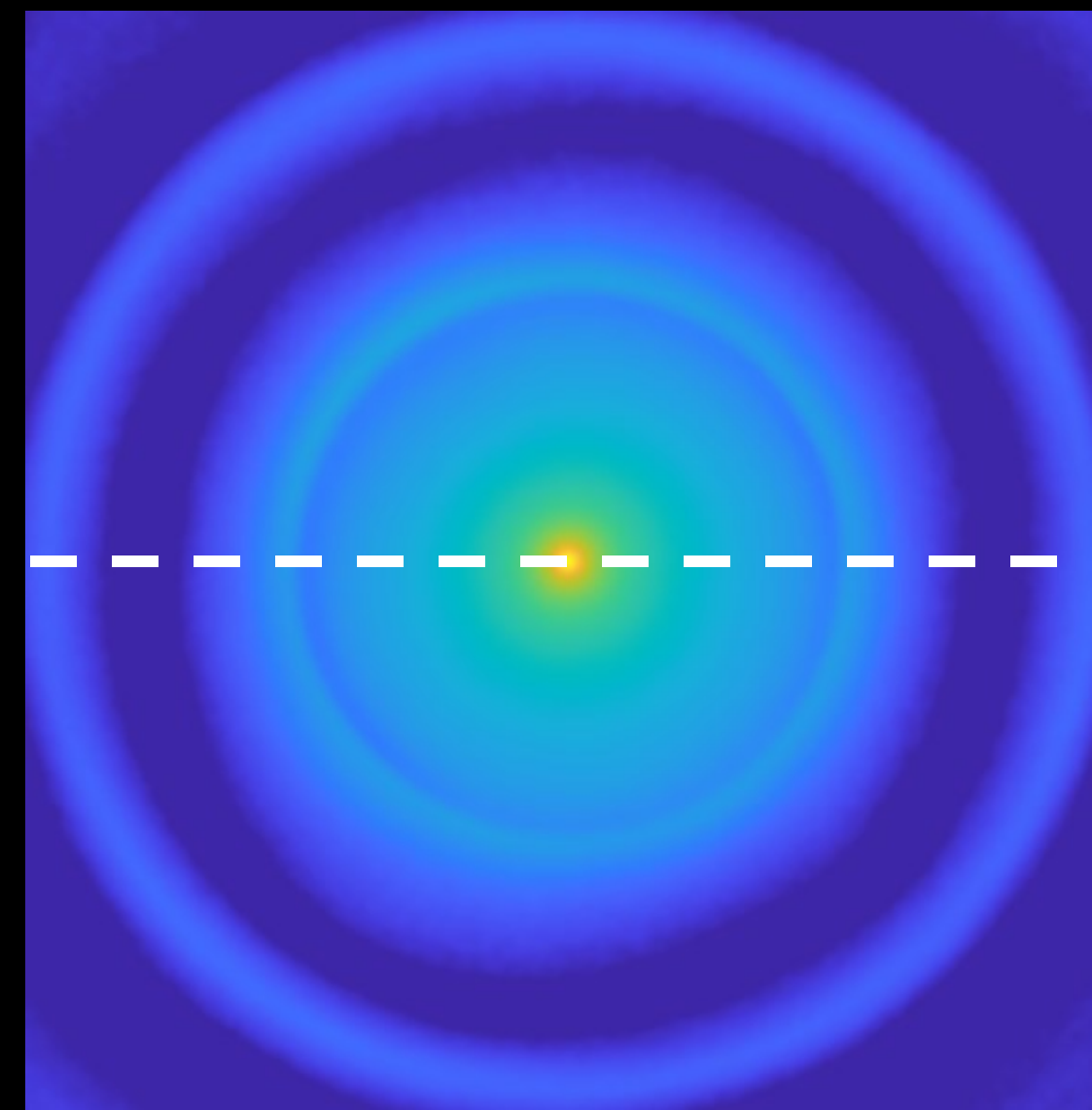
experimental data

rendered data

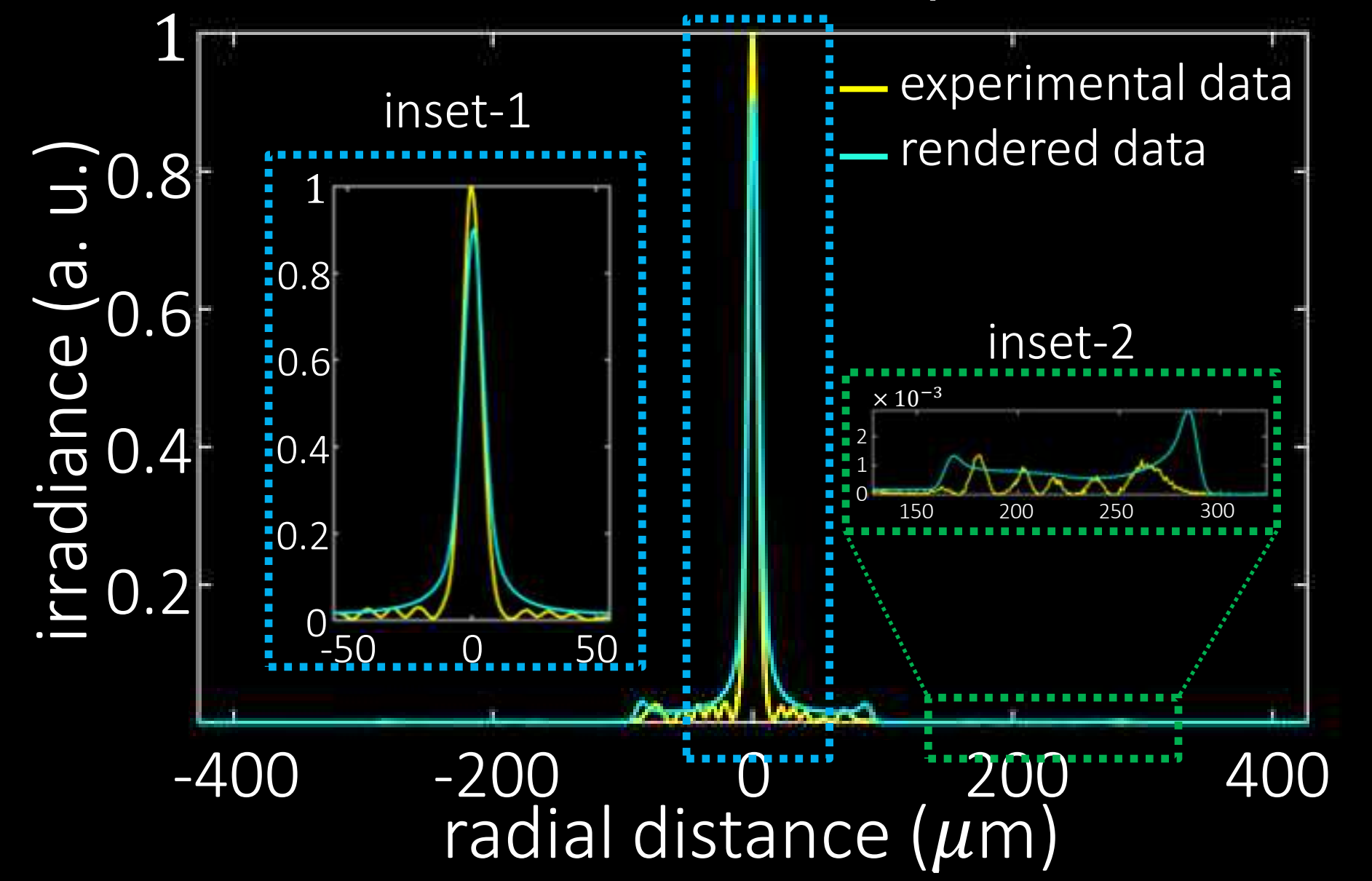
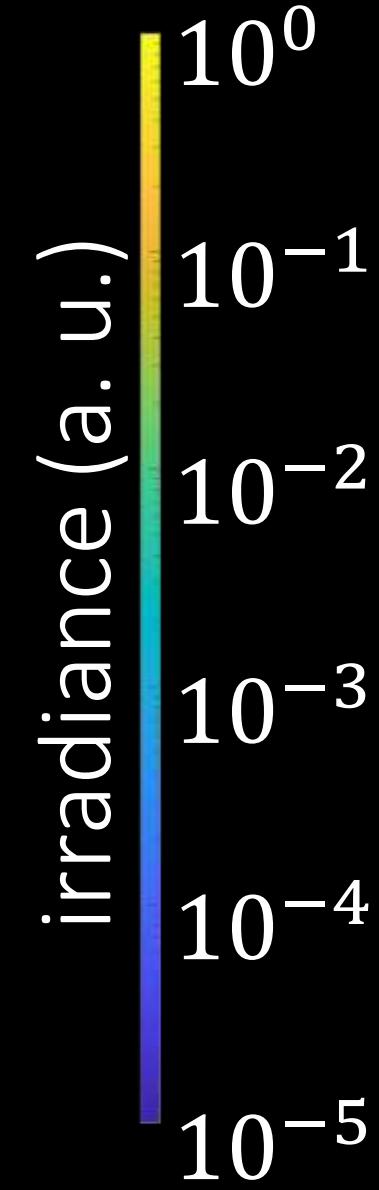
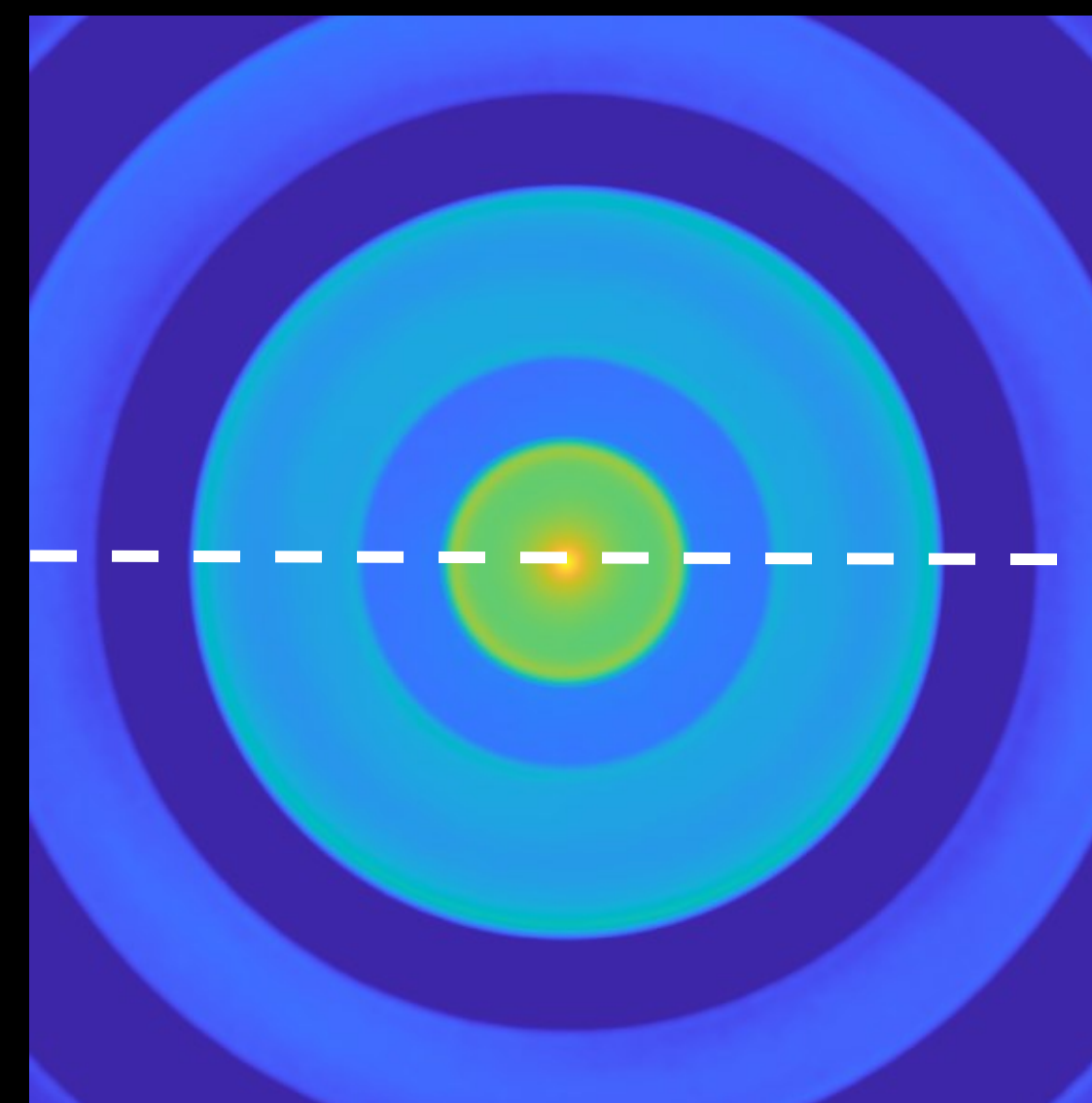
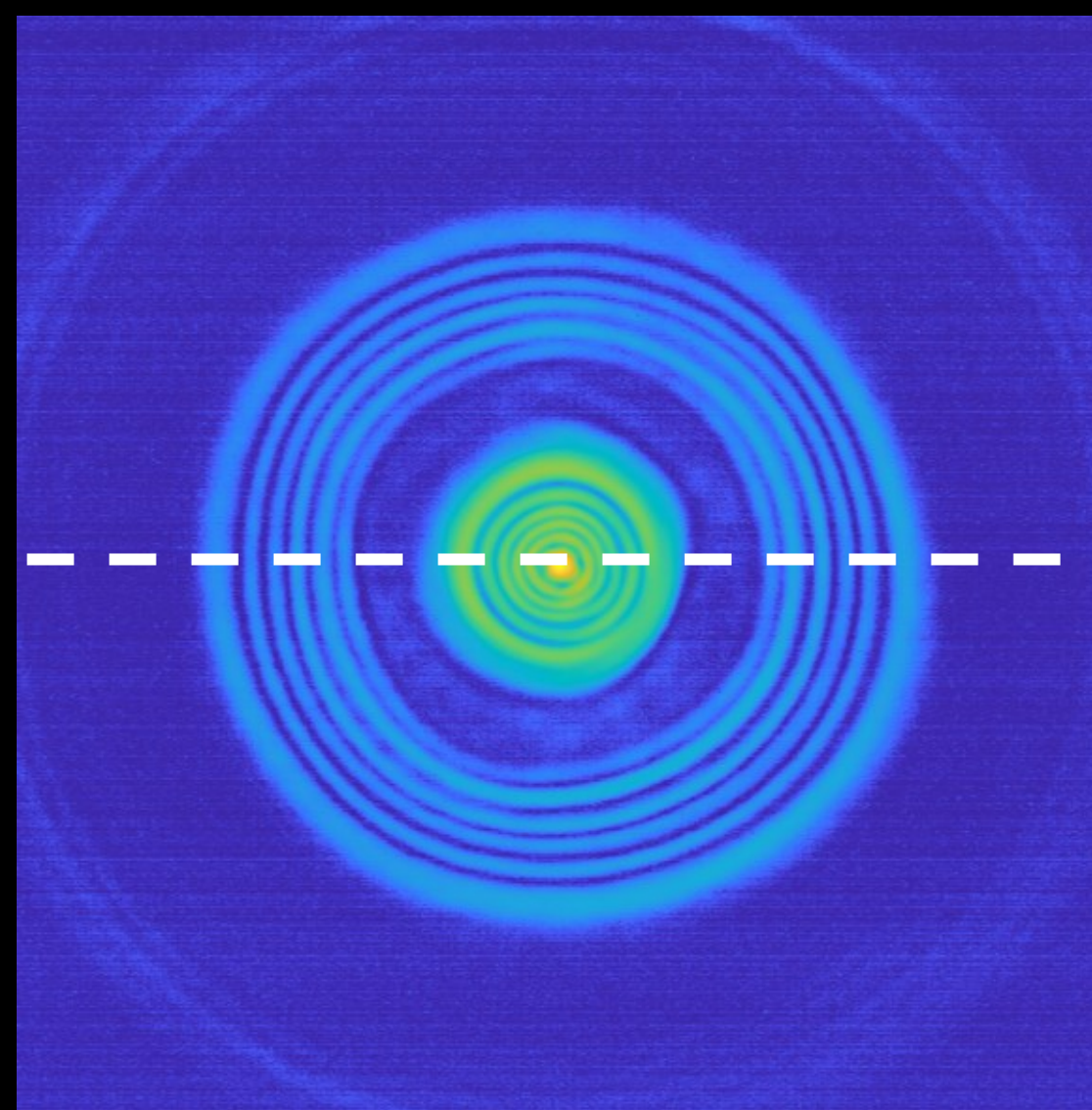
first focus



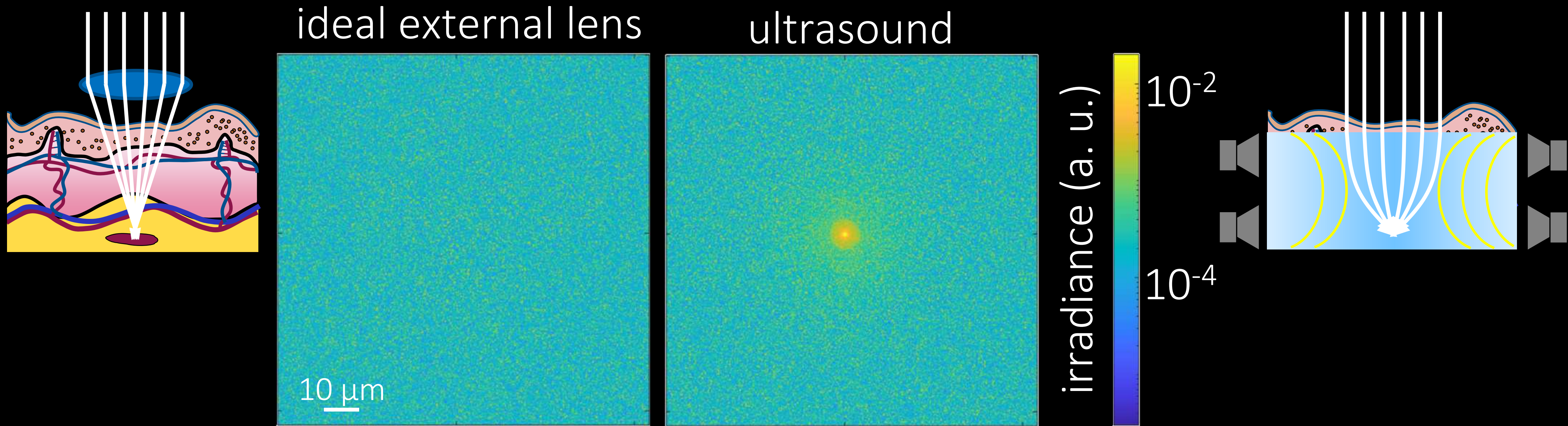
50 μm



second focus



Optimized configurations are better than ideal external lens



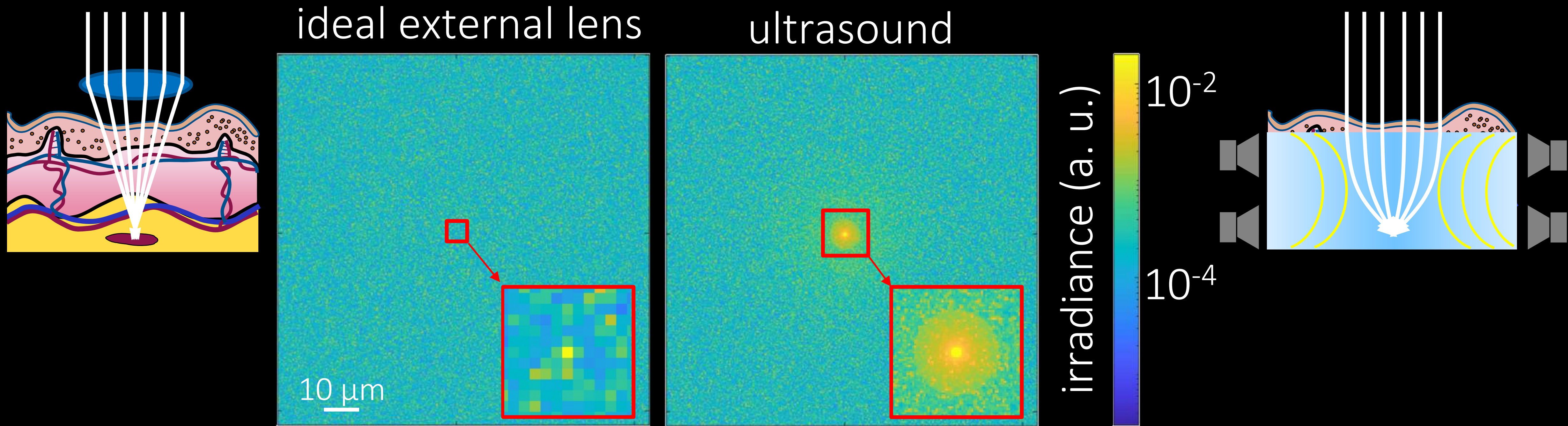
On human bladder (10 scattering lengths, 2.67 mm thick)

- 50% higher focusing performance than external lens.
- 300% higher focusing performance than previous designs.

On brain tissue (50 scattering lengths, 7.5 mm thick)

- 15% higher focusing performance than external lens.
- Experimentally validated on tissue phantoms.

Optimized configurations are better than ideal external lens



On human bladder (10 scattering lengths, 2.67 mm thick)

- 50% higher focusing performance than external lens.
- 300% higher focusing performance than previous designs.

On brain tissue (50 scattering lengths, 7.5 mm thick)

- 15% higher focusing performance than external lens.
- Experimentally validated on tissue phantoms.

Physics-based rendering and its applications to computational imaging

forward rendering

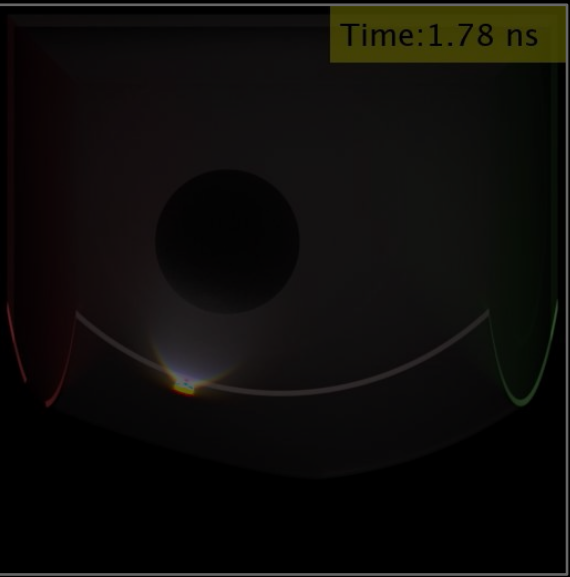


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

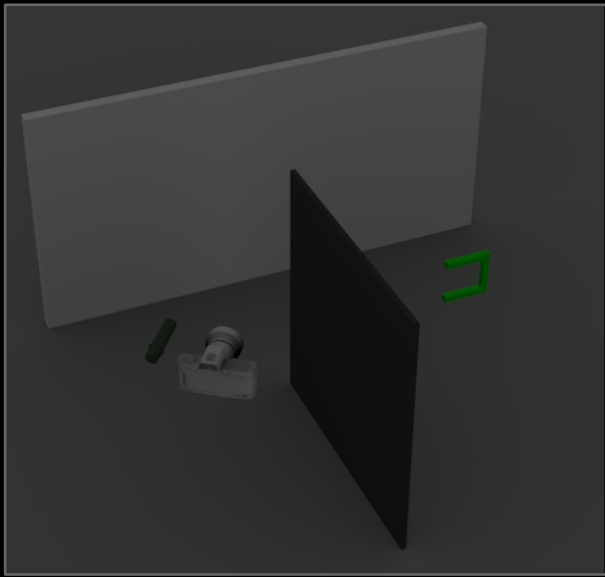
inverse rendering



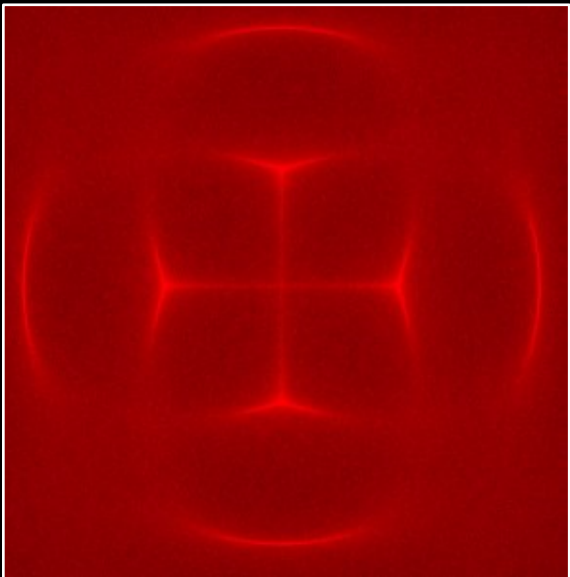
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



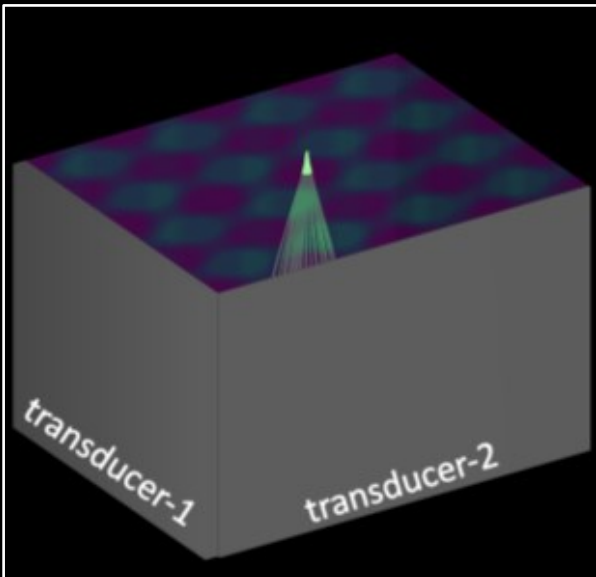
time-of-flight imaging



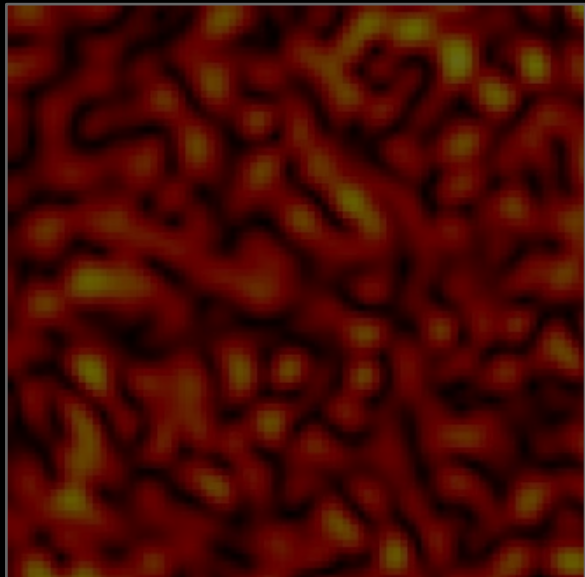
non-line-of-sight imaging



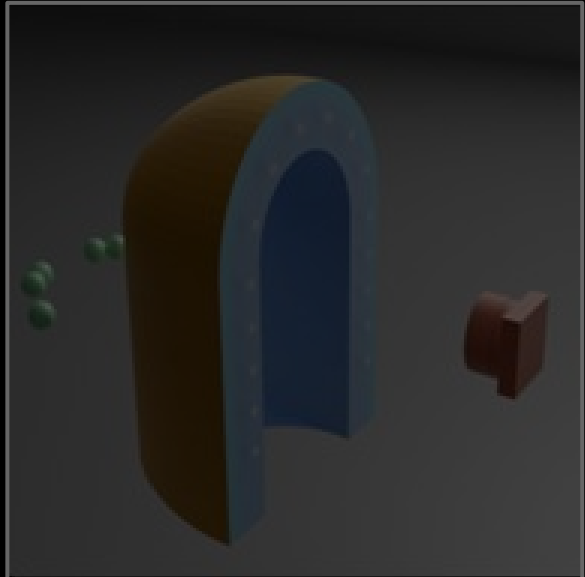
acousto-optic lensing



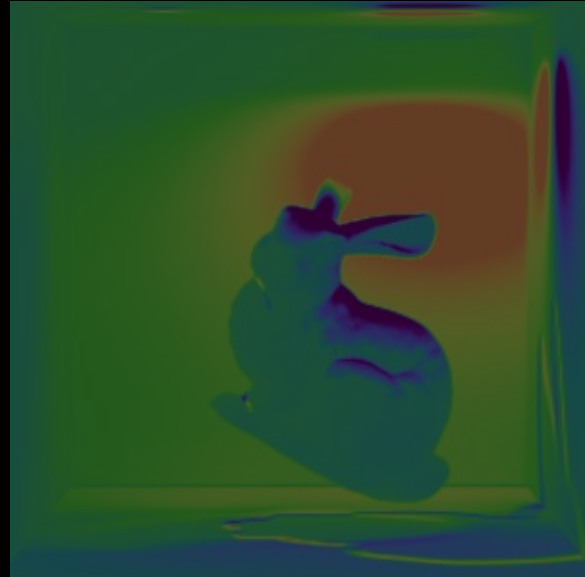
ultrafast light scanning



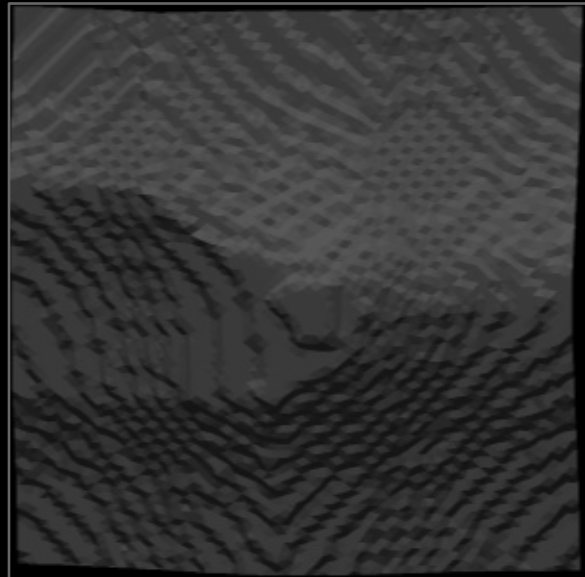
speckle imaging



tactile sensor design

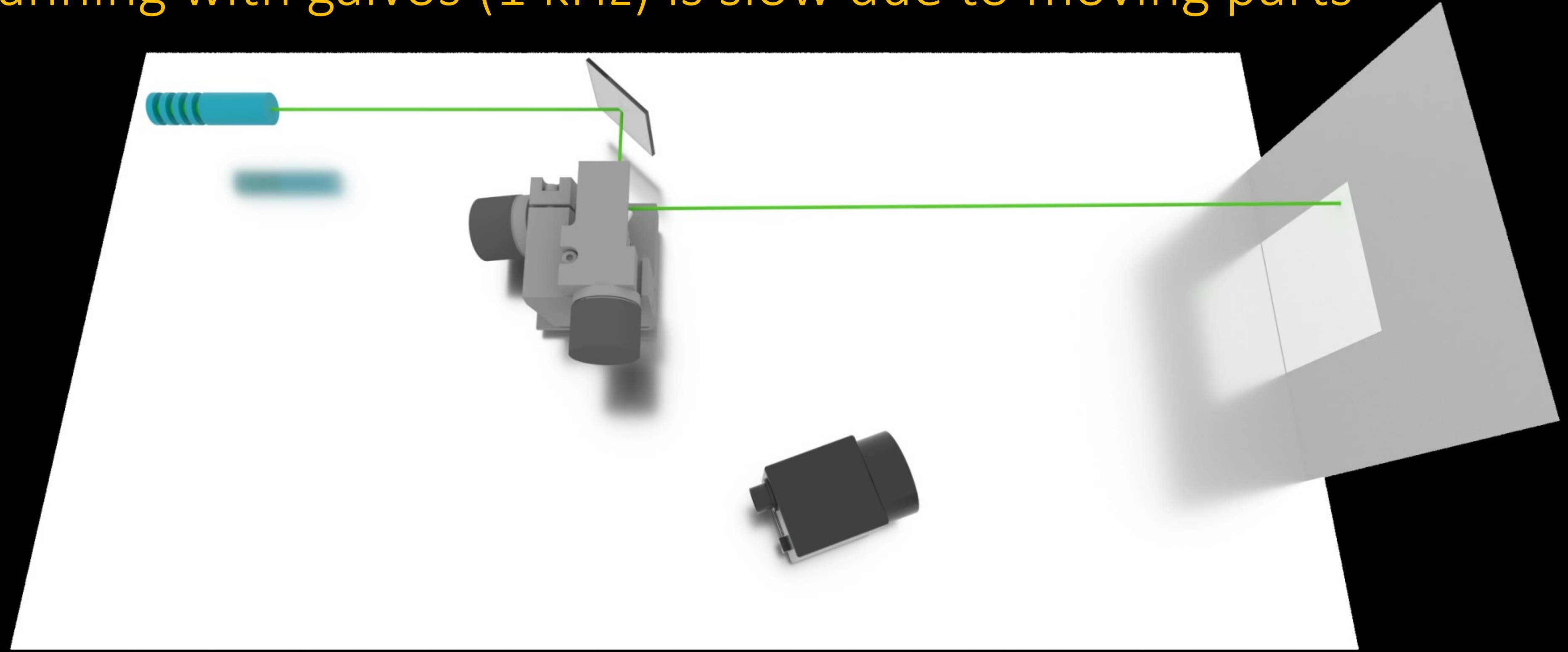


differentiable renderer



inverse problems

scanning with galvos (1 kHz) is slow due to moving parts



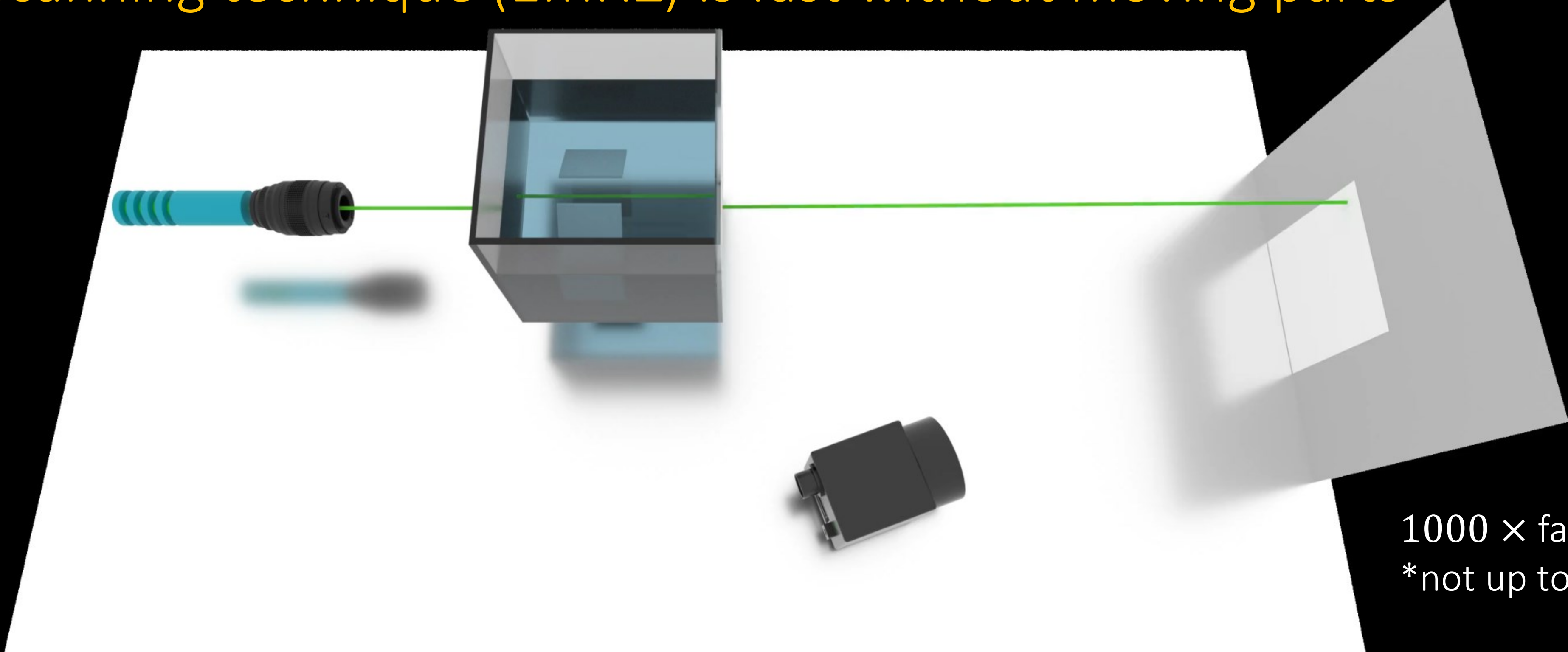
projector

microscopy

lidar



our scanning technique (1MHz) is fast without moving parts



1000 × faster
*not up to scale

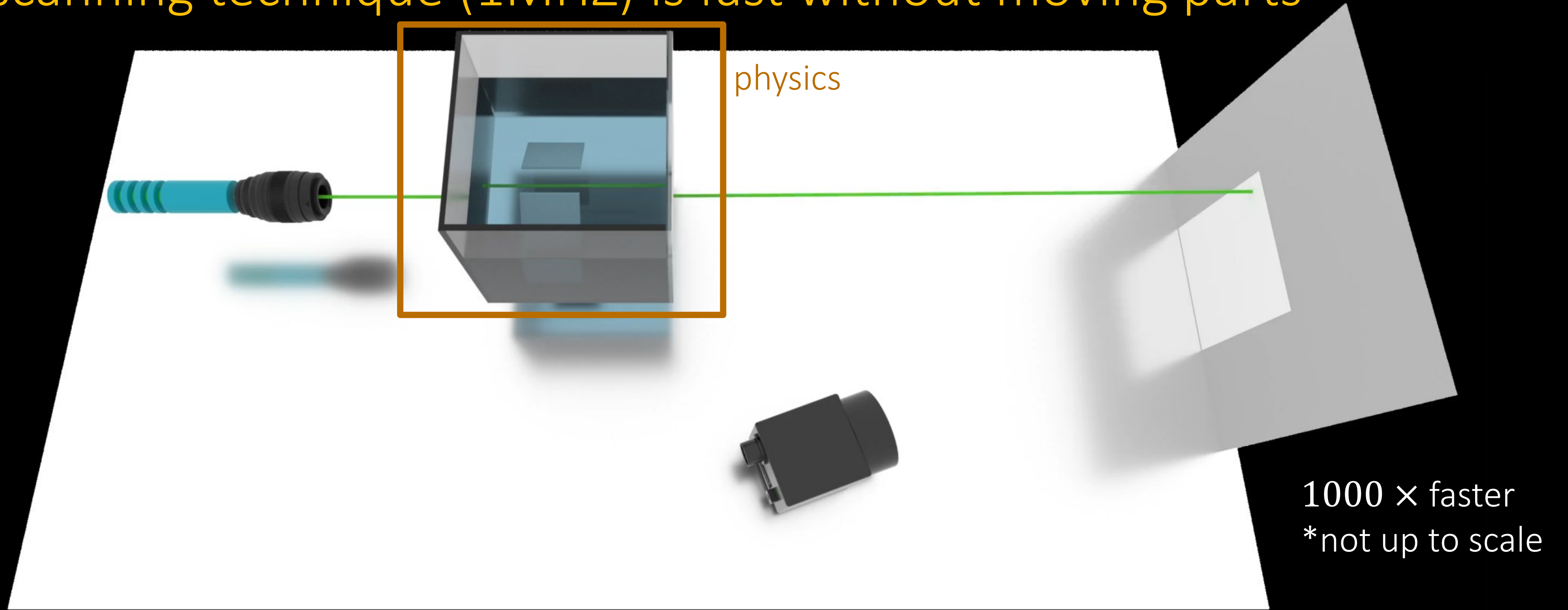
projector

microscopy

lidar



our scanning technique (1MHz) is fast without moving parts



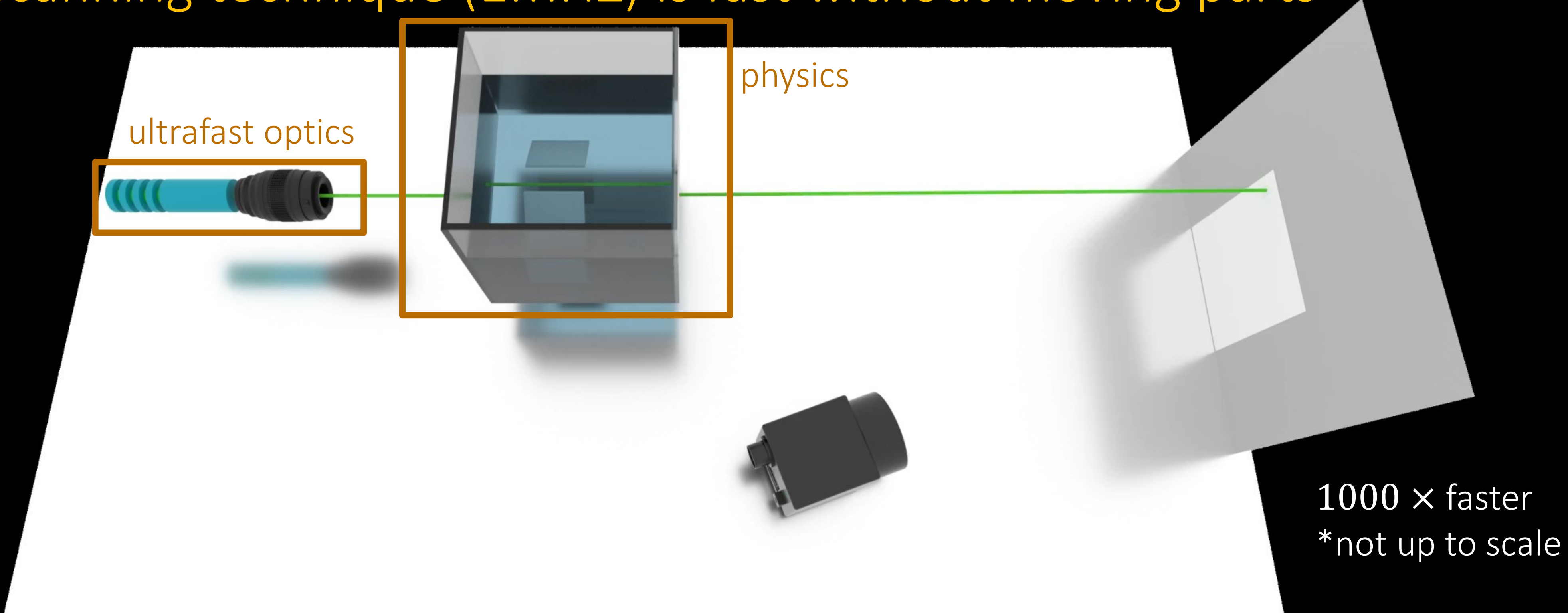
projector

microscopy

lidar



our scanning technique (1MHz) is fast without moving parts



projector



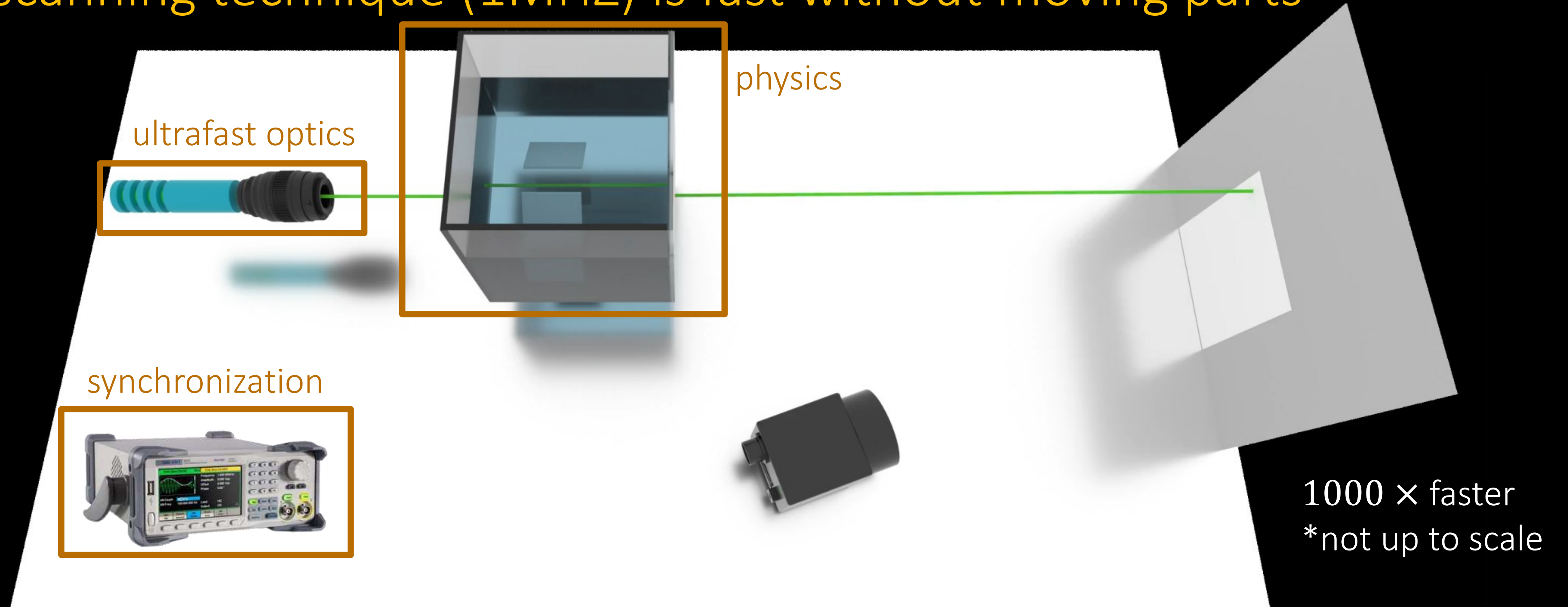
microscopy



lidar



our scanning technique (1MHZ) is fast without moving parts



projector



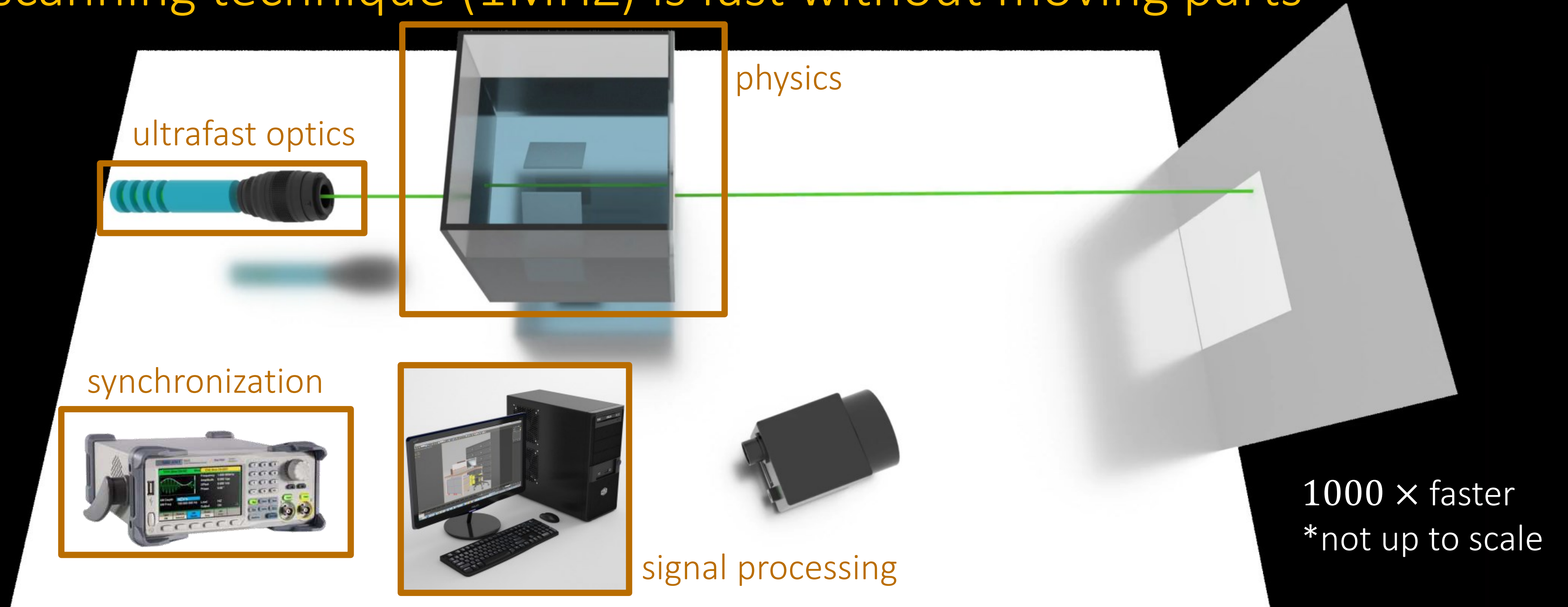
microscopy



lidar



our scanning technique (1MHz) is fast without moving parts



projector



microscopy



lidar



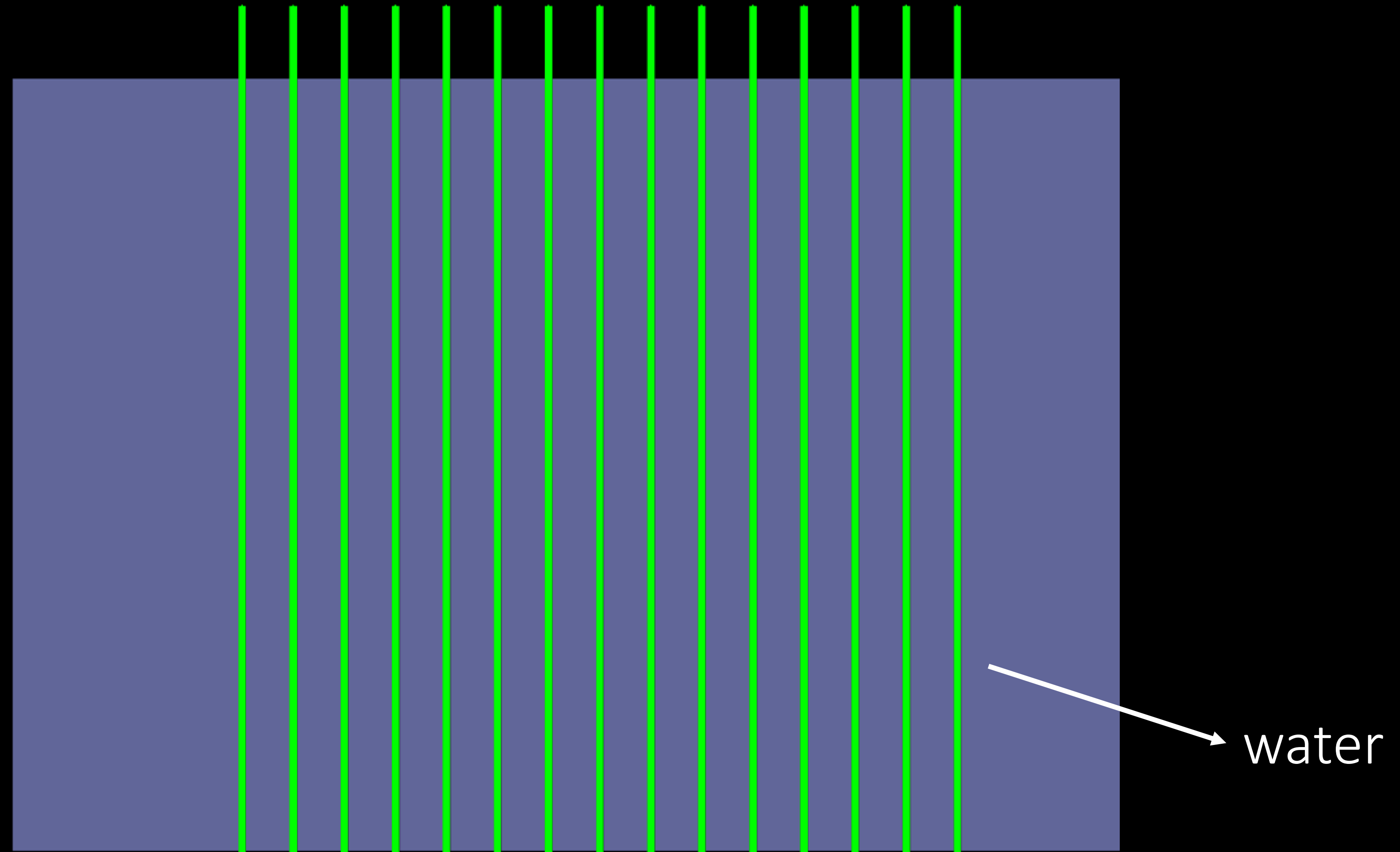
Physics sound, light, and matter interaction



water

2D visualization

Physics sound, light, and matter interaction



2D visualization

Physics sound, light, and matter interaction



2D visualization

Physics sound, light, and matter interaction

nm scale
(exaggerated)



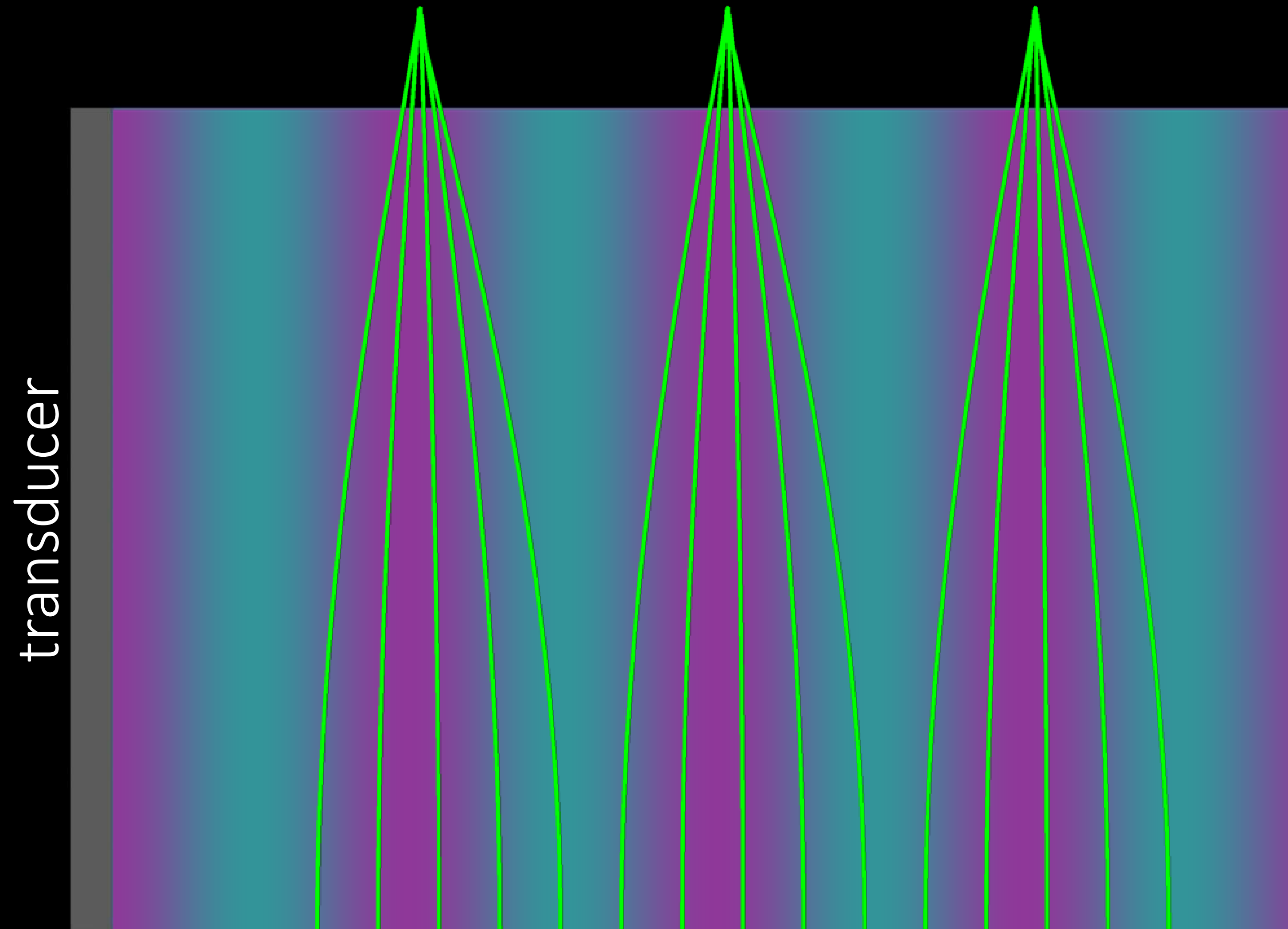
2D visualization

Physics sound, light, and matter interaction



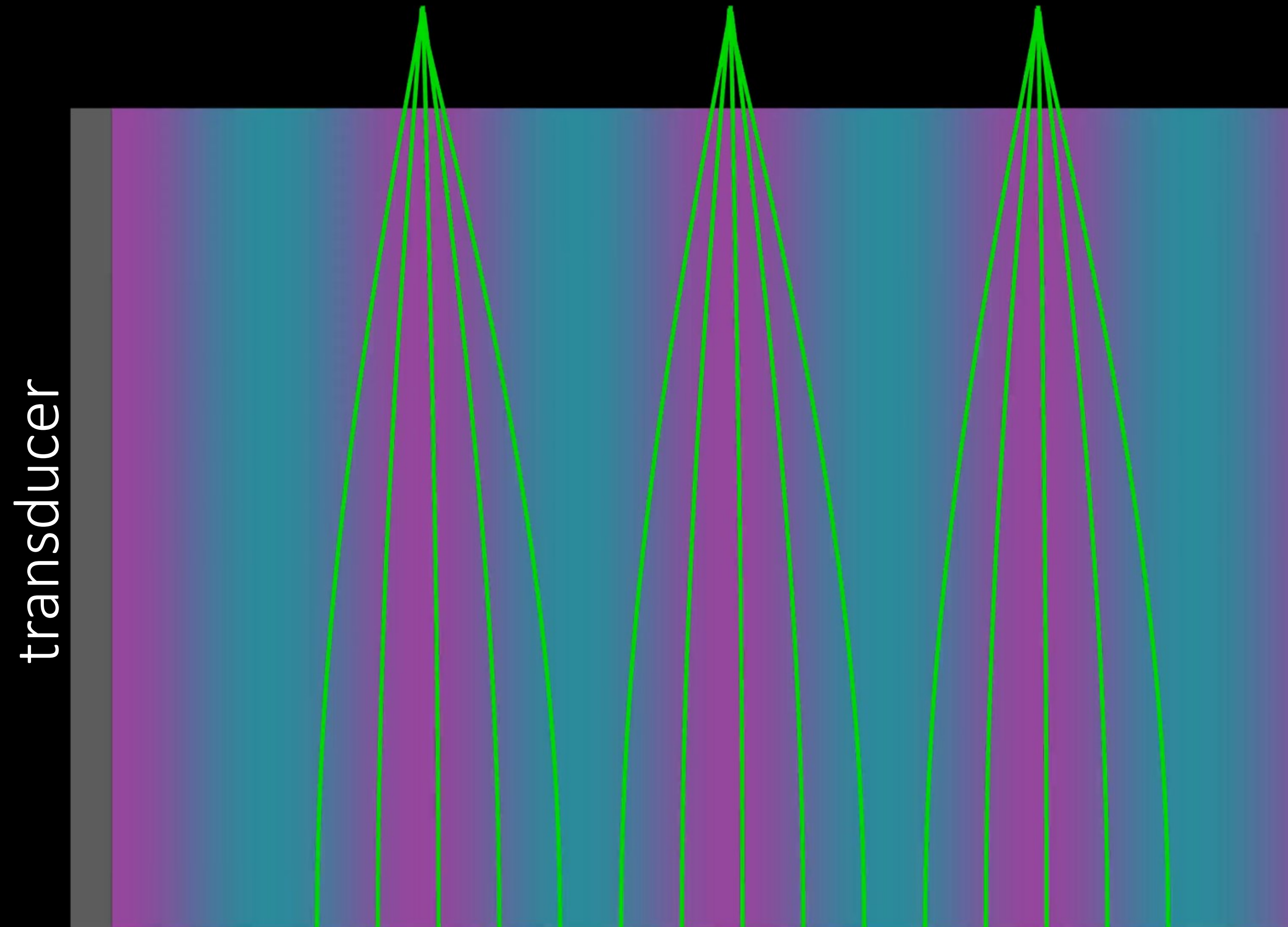
2D visualization

Physics sound, light, and matter interaction



2D visualization

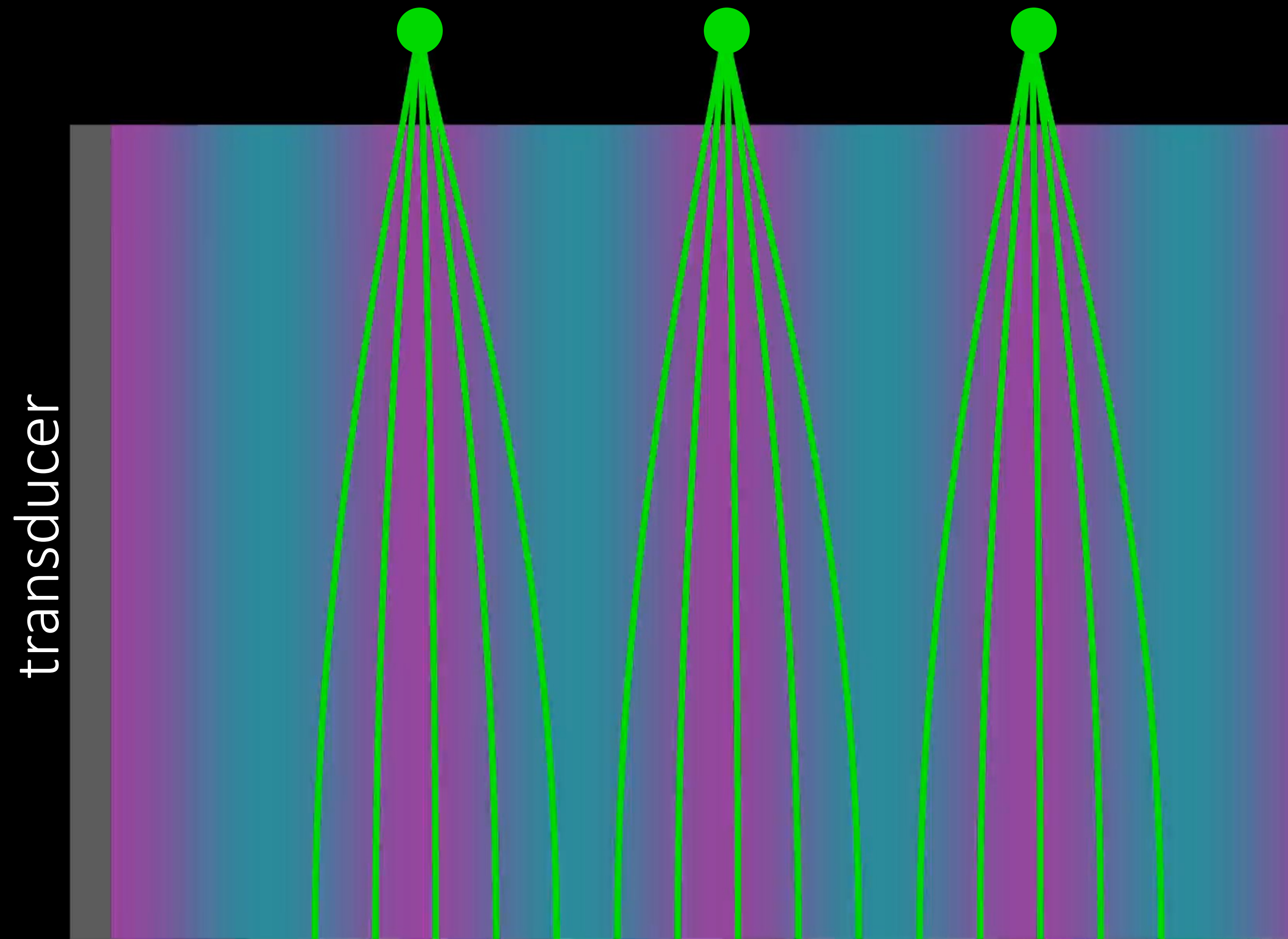
Physics sound, light, and matter interaction



focus travels at the speed of 1.5 km/s

2D visualization

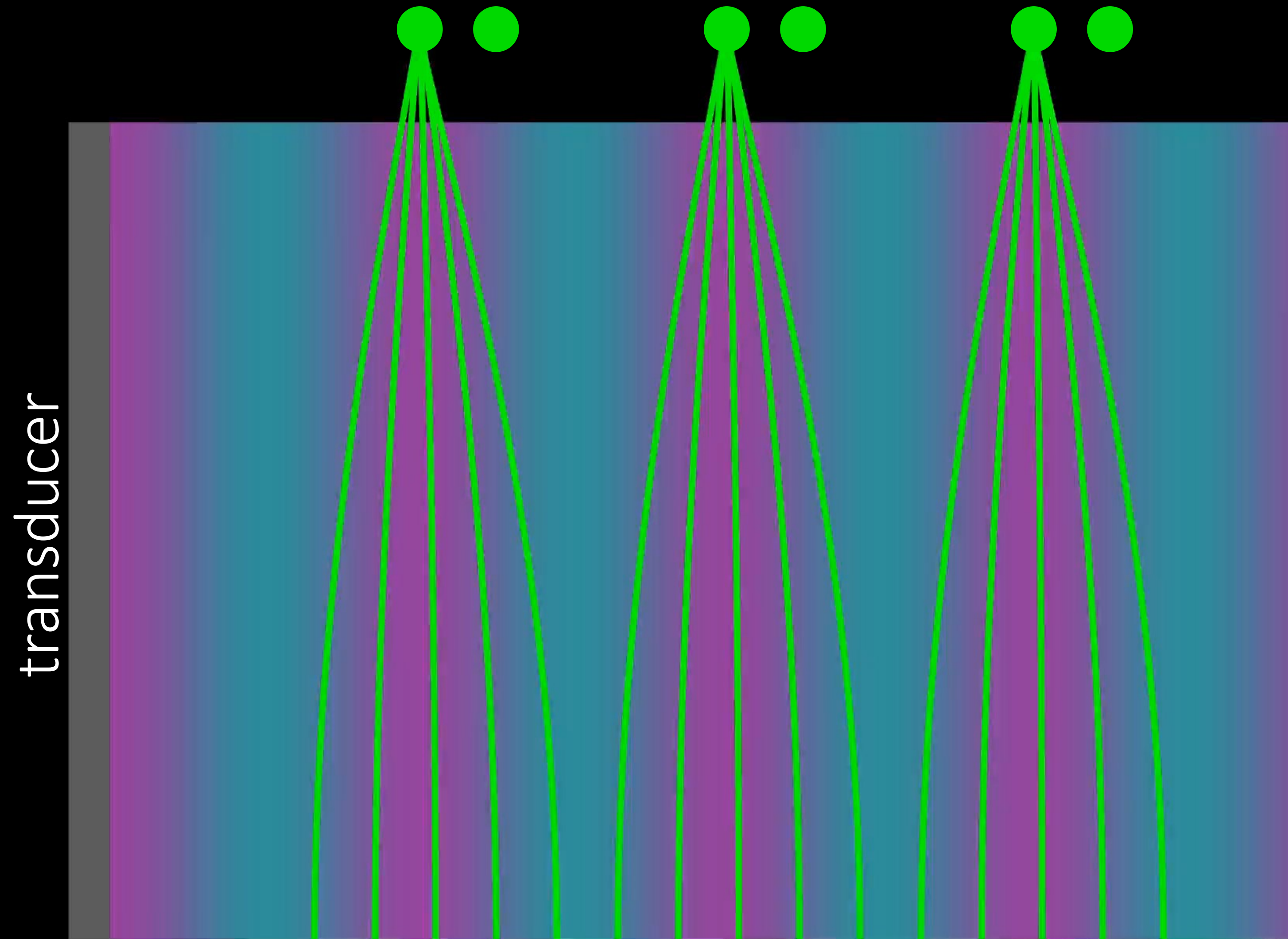
Ultrafast optics and synchronization



ultrafast synchronized optics that illuminates same spot

2D visualization

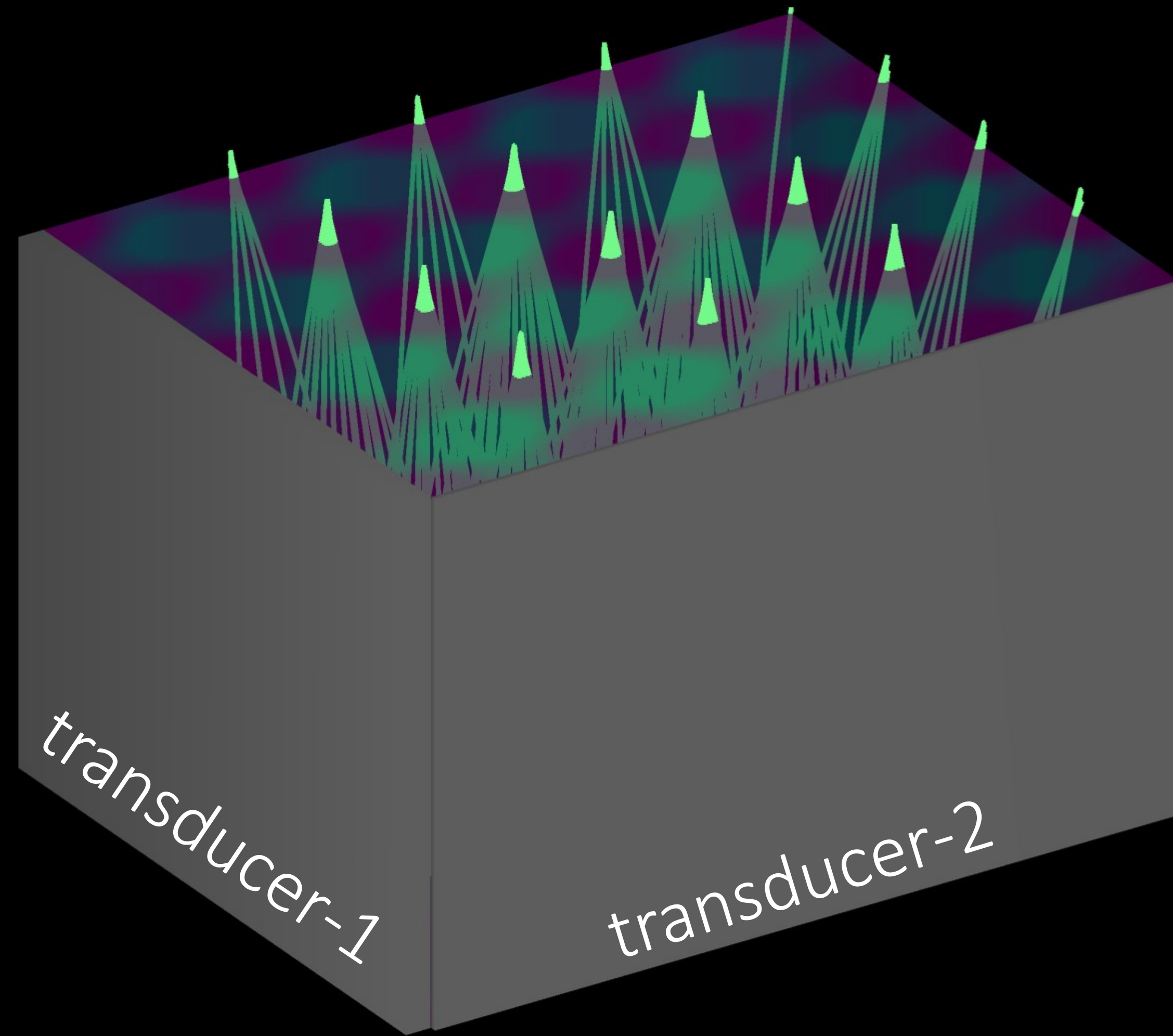
Ultrafast optics and synchronization



illuminating two spots per cycle

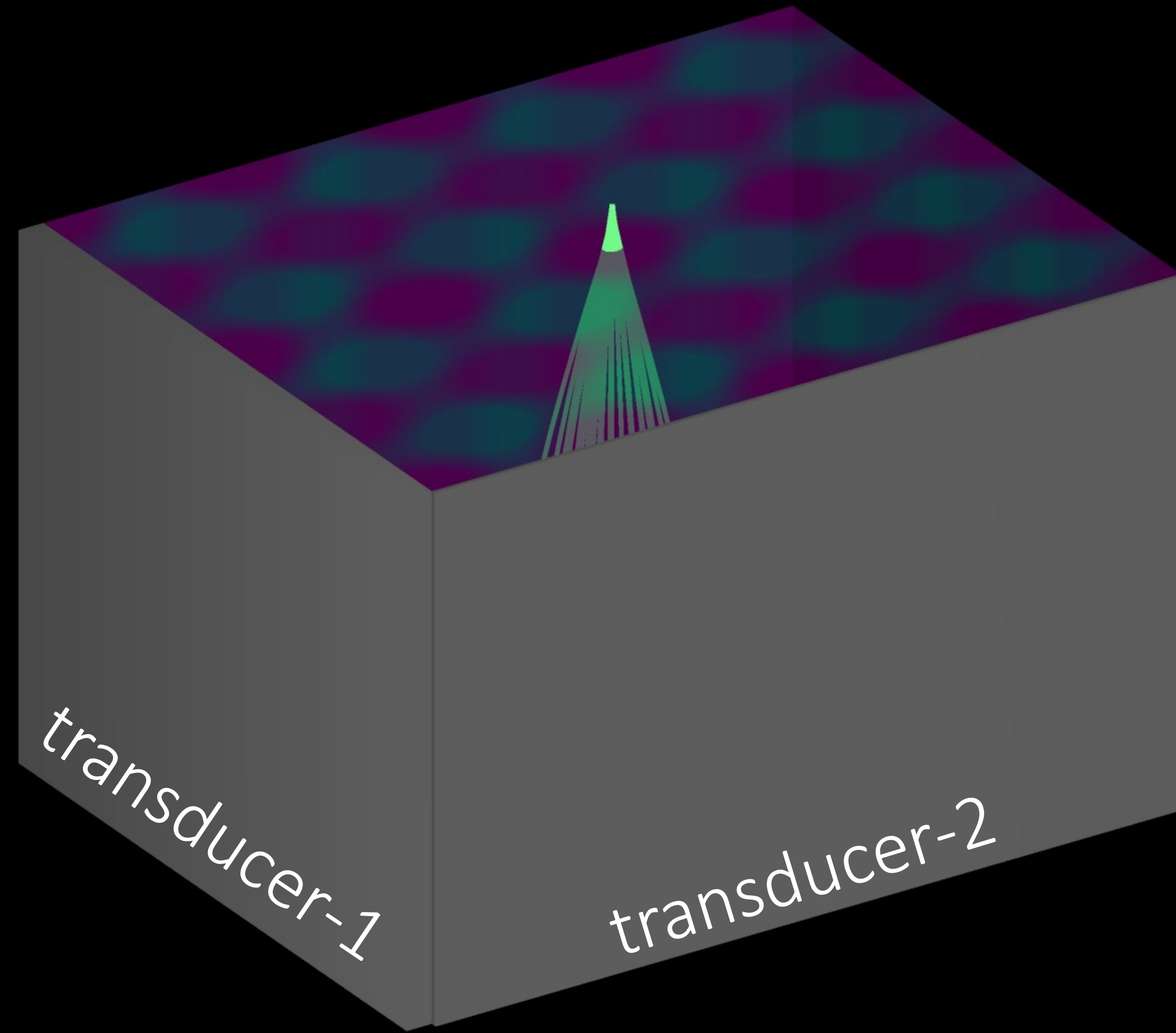
2D visualization

Dot projector



2D visualization

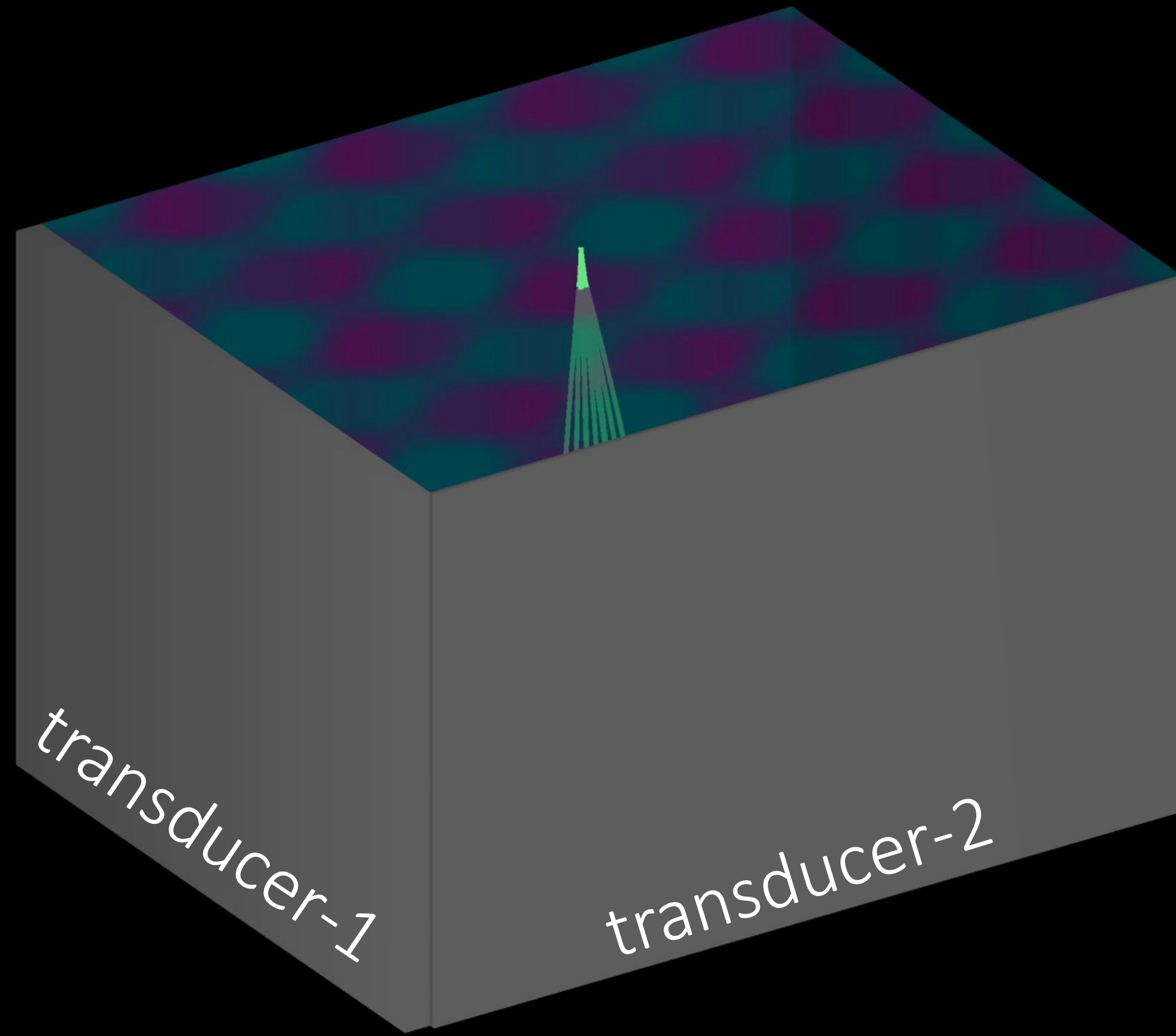
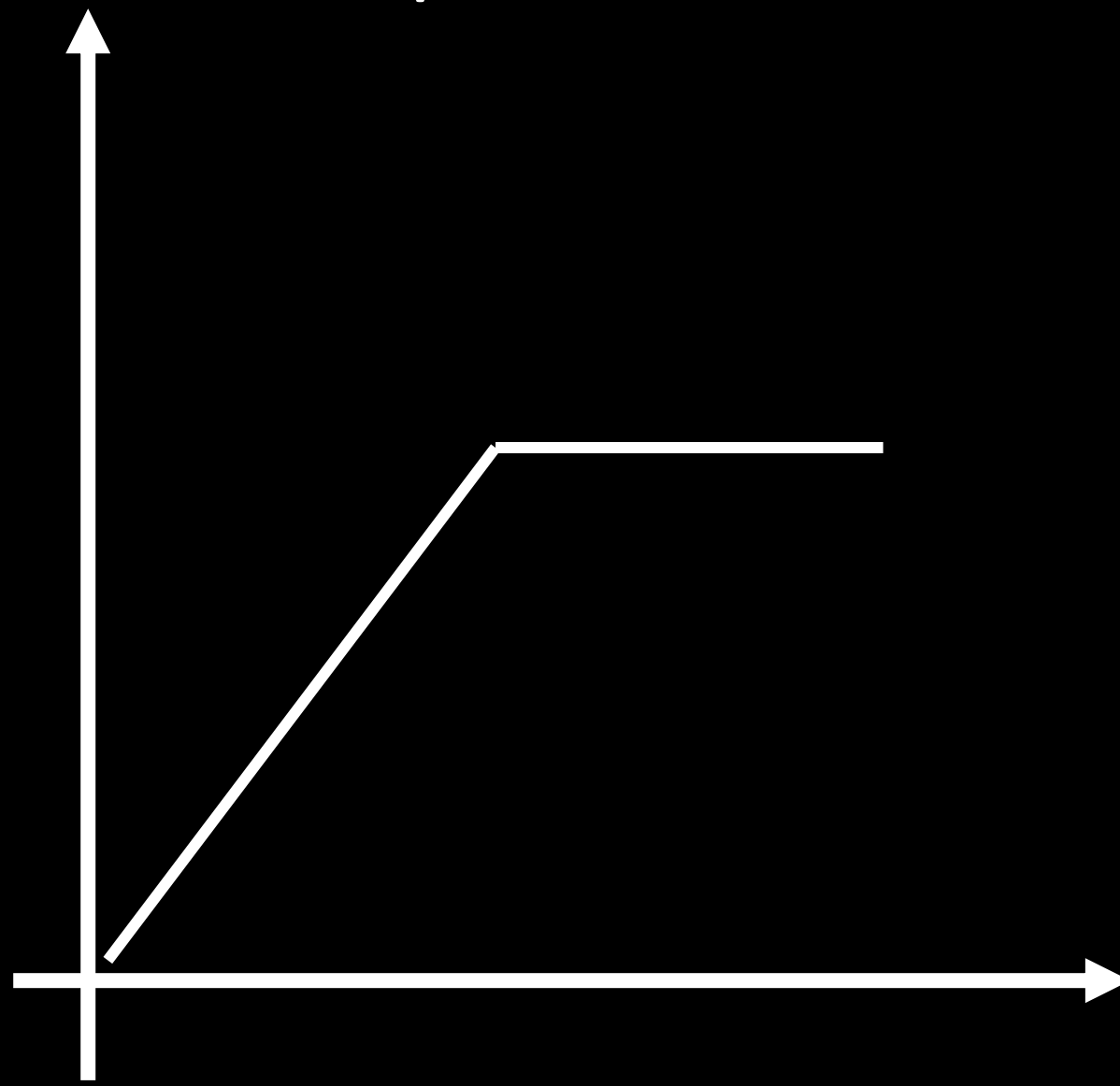
Dot projector



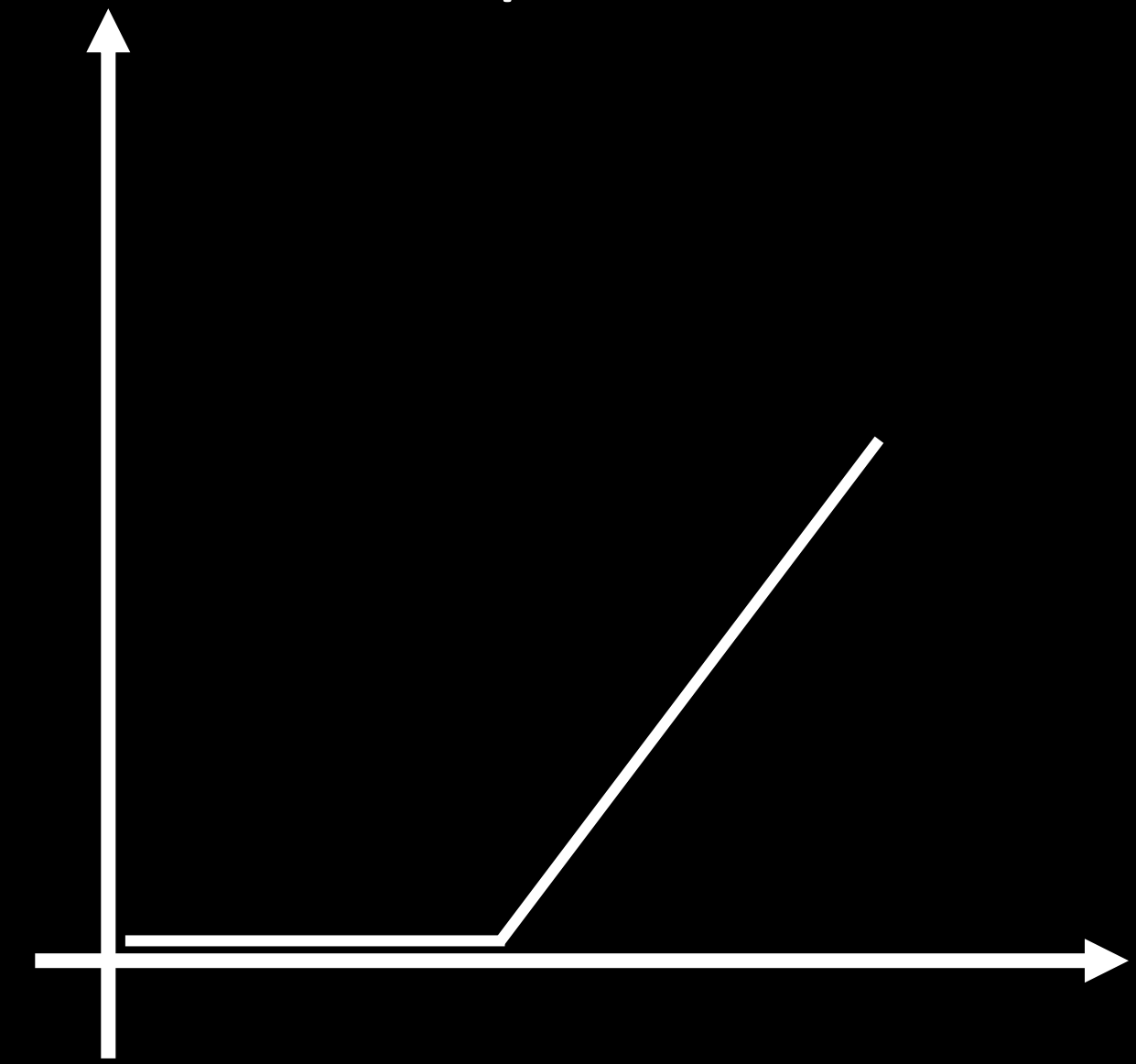
2D visualization

Dot projector

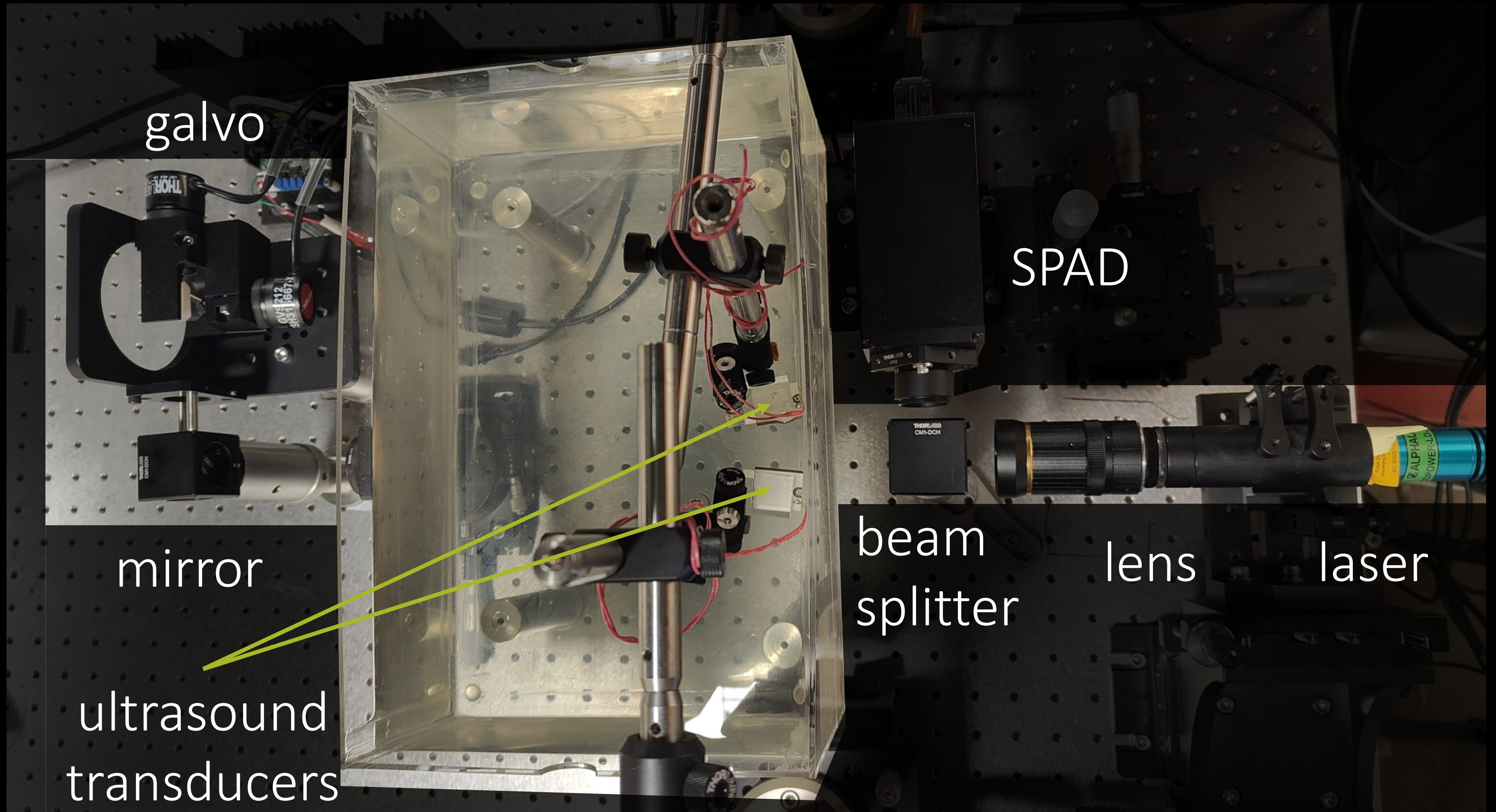
transducer-1
phase



transducer-2
phase



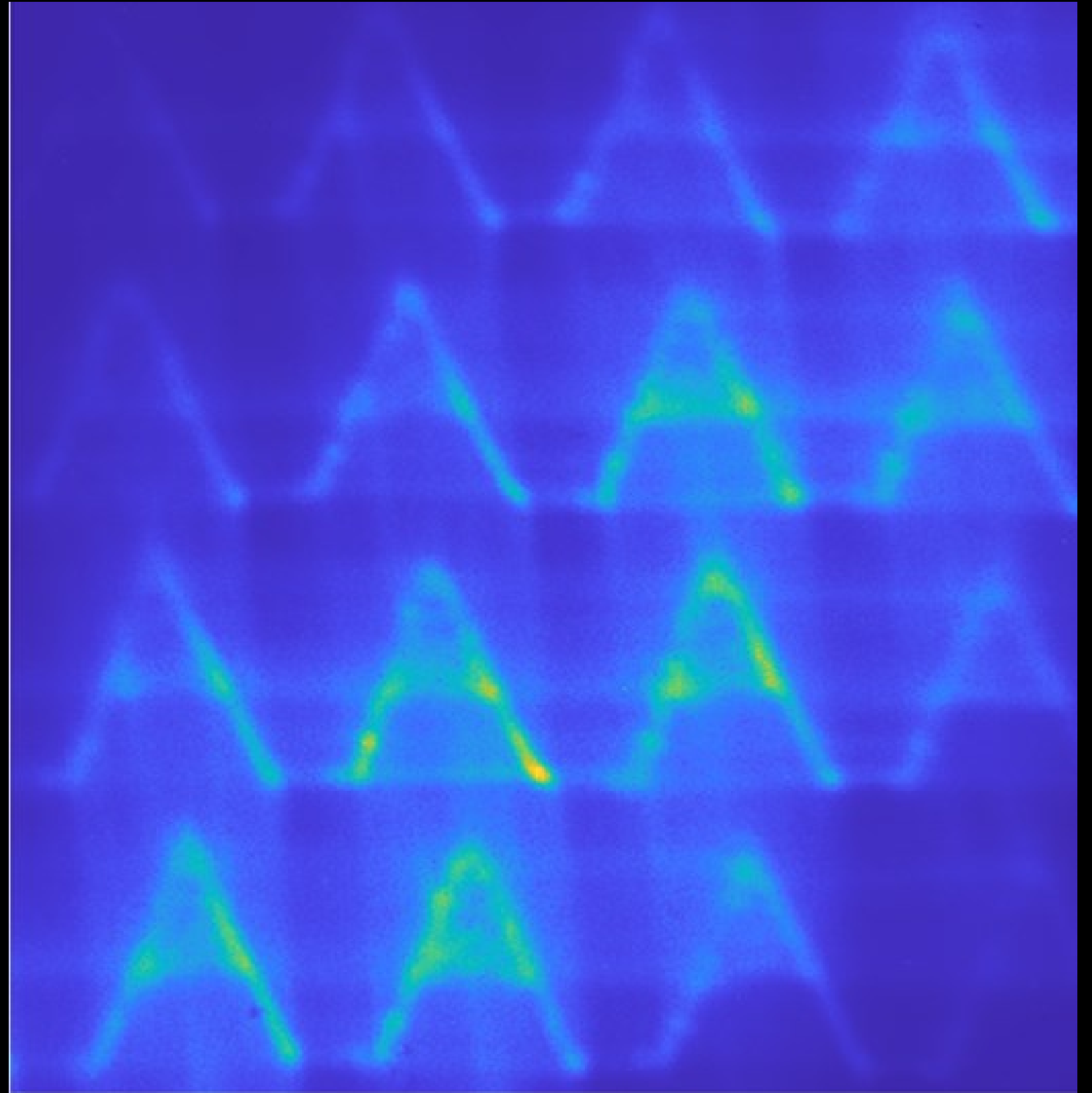
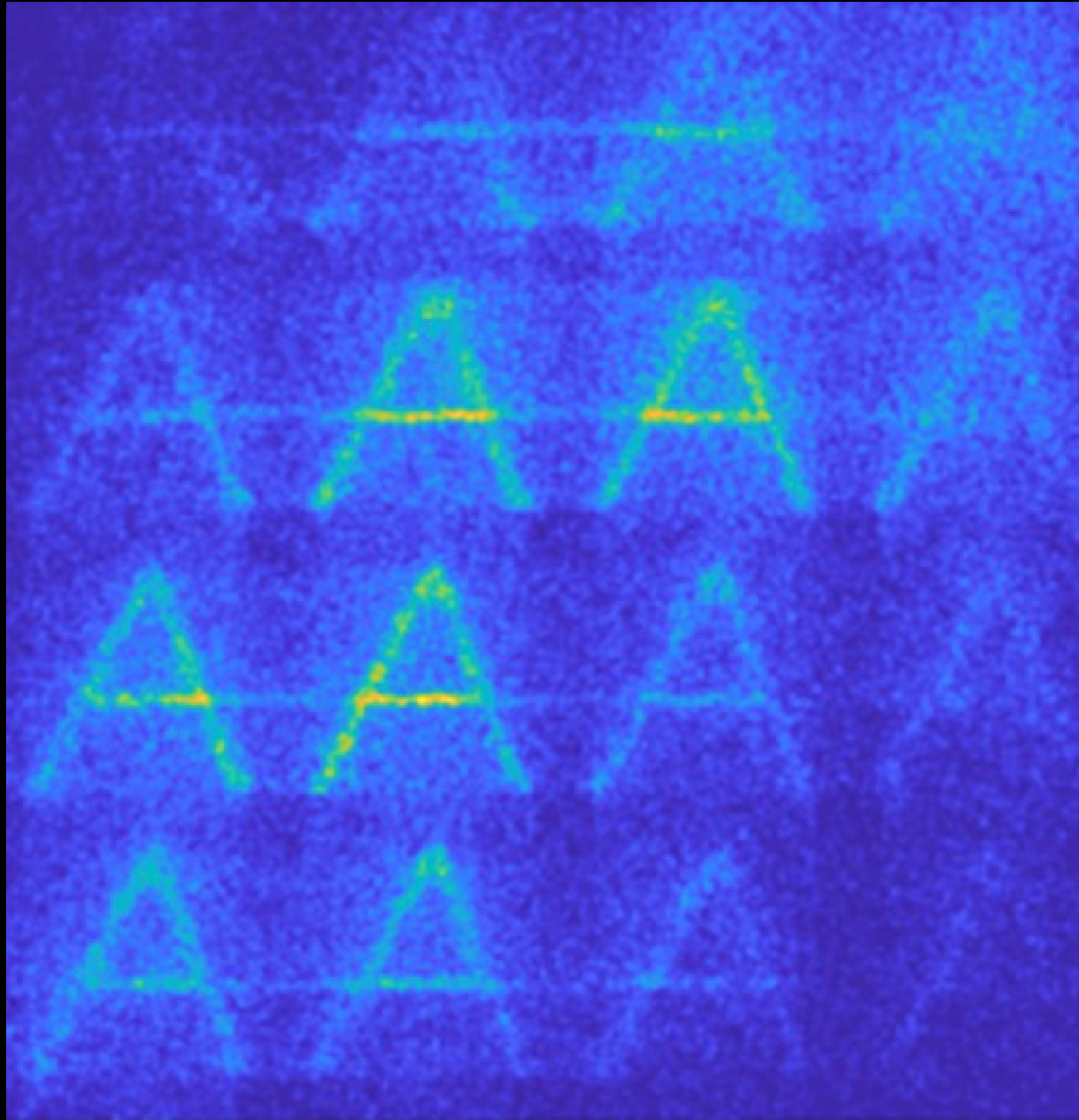
Hardware experiments



Renderer

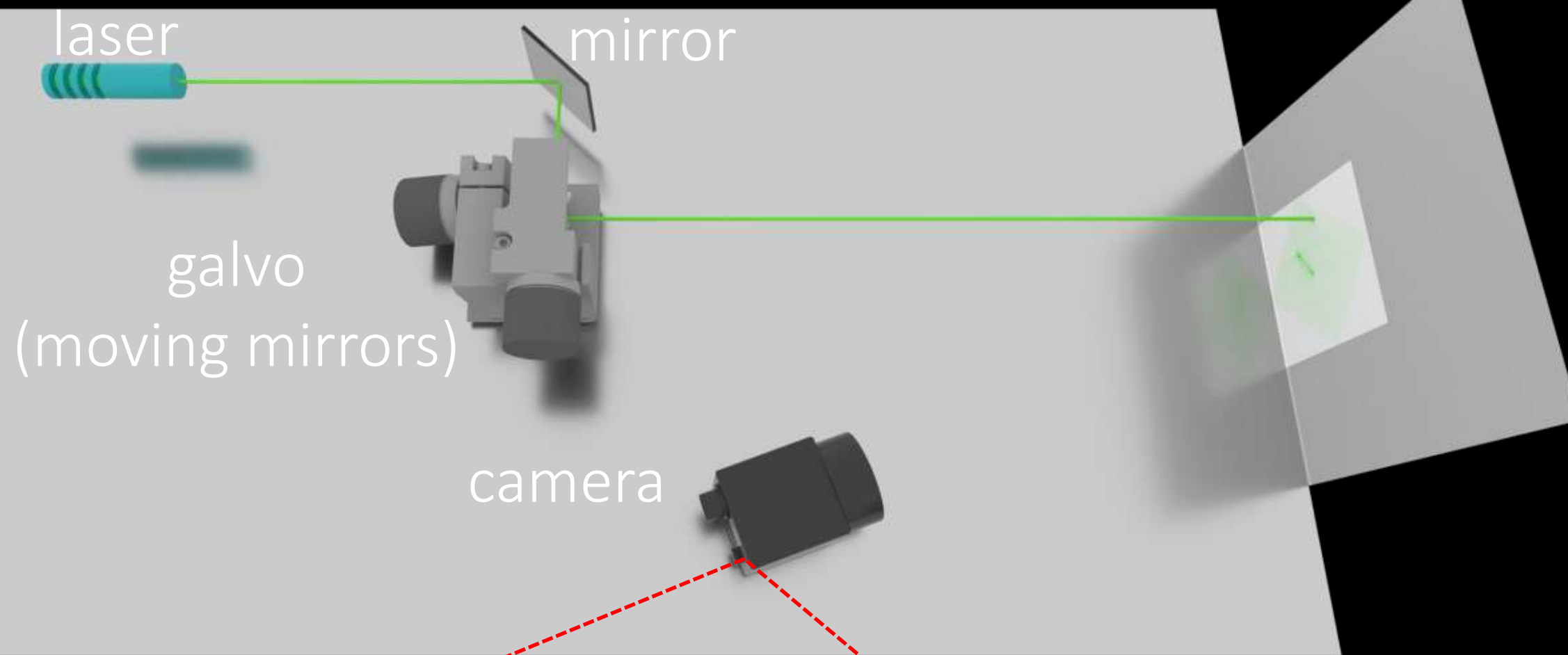
vs.

Hardware

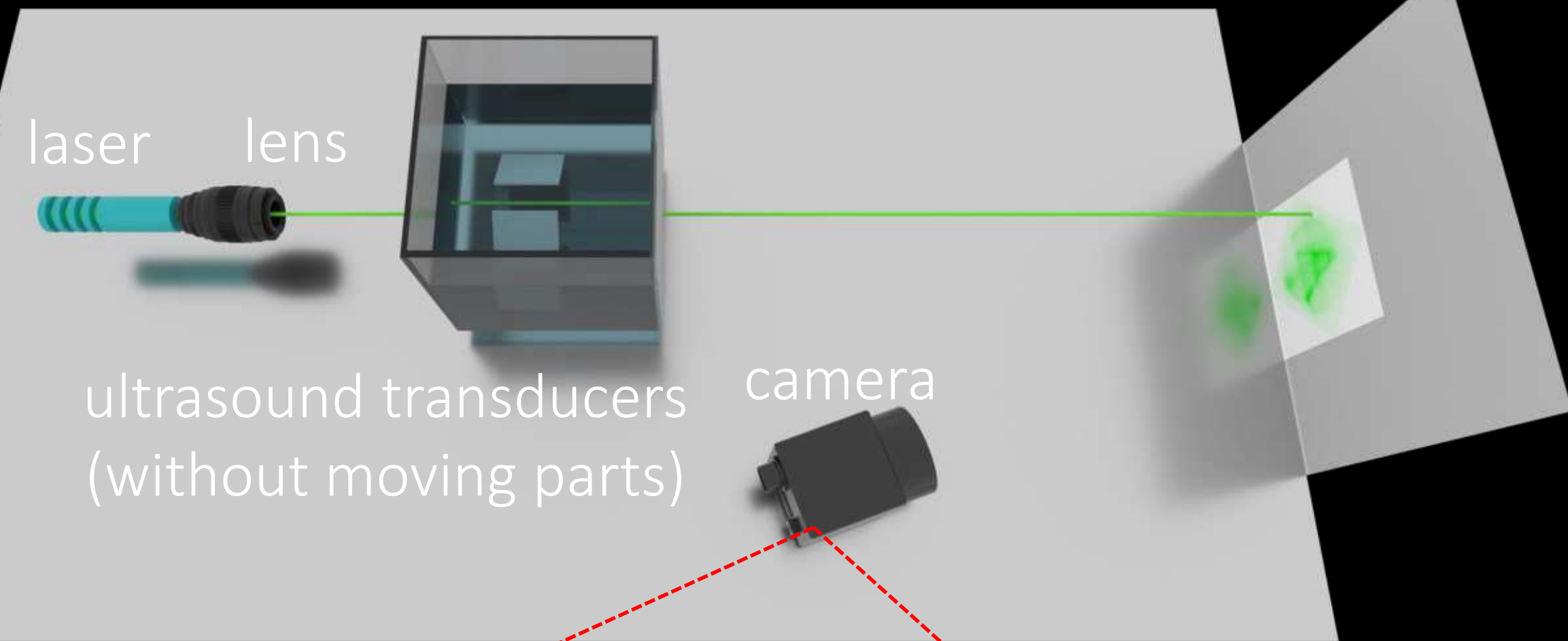


Hardware results: dot projector

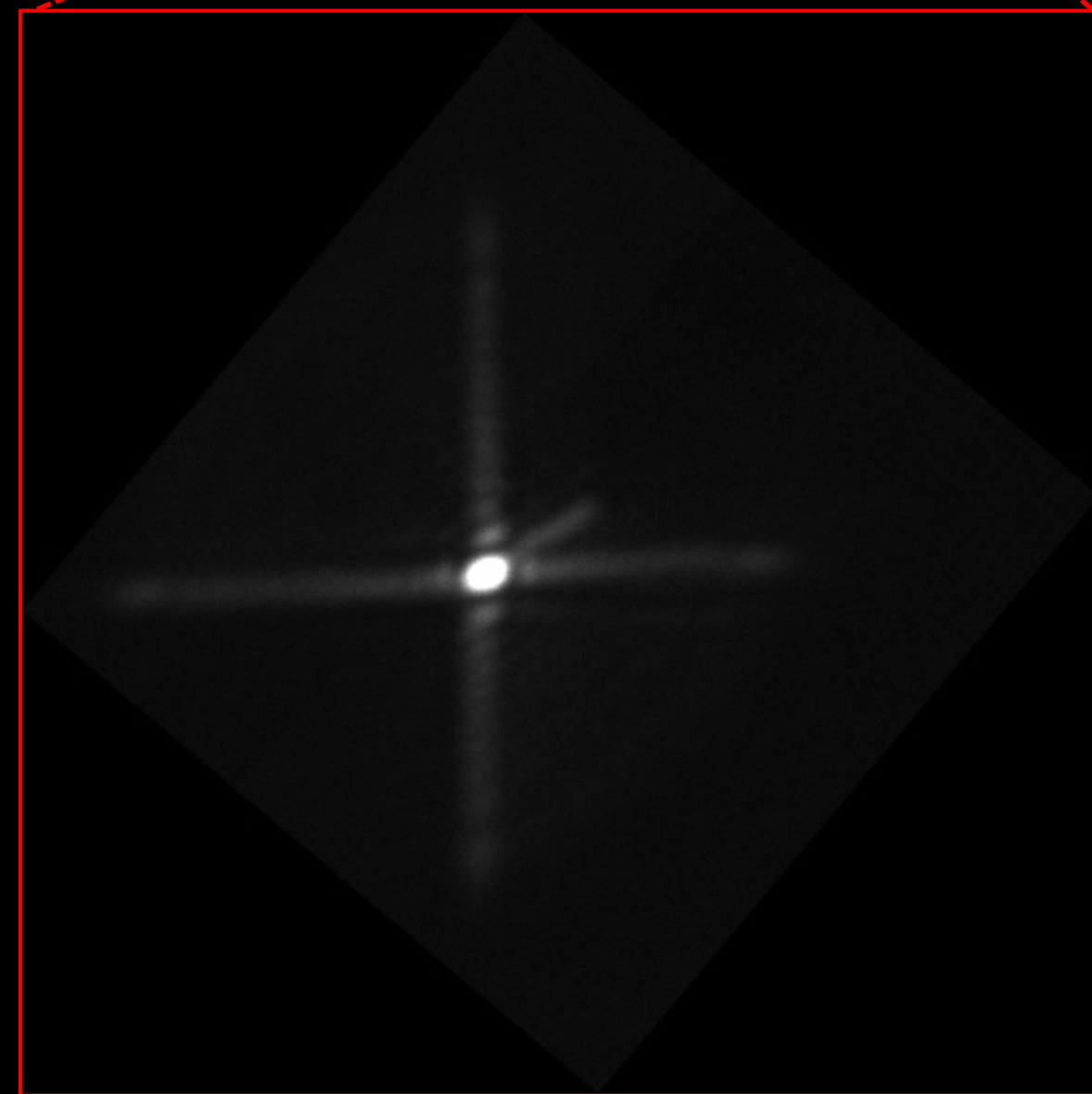
galvo steering (1 kHz)



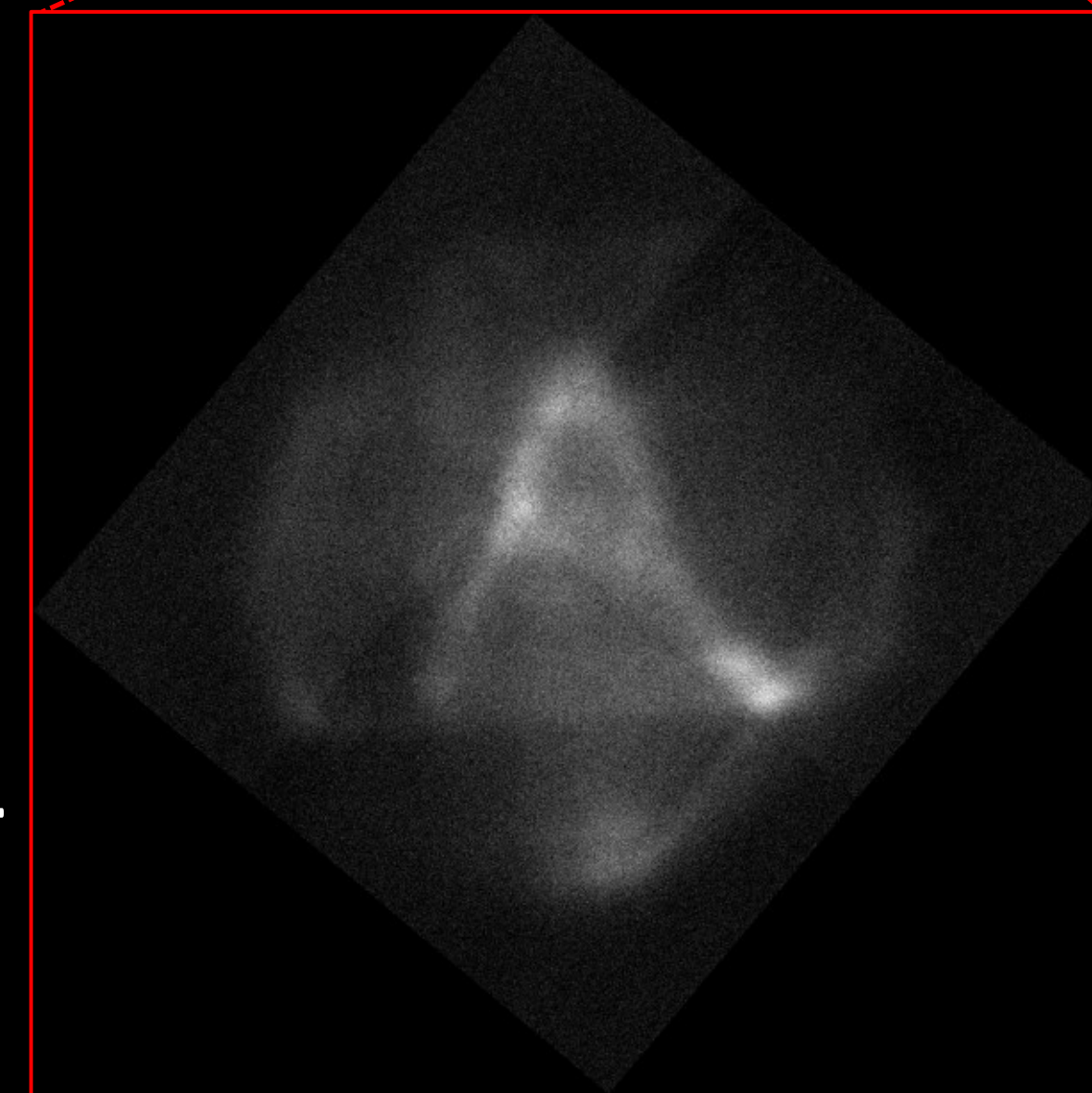
proposed steering (1 MHz)



exposure= 1 ms

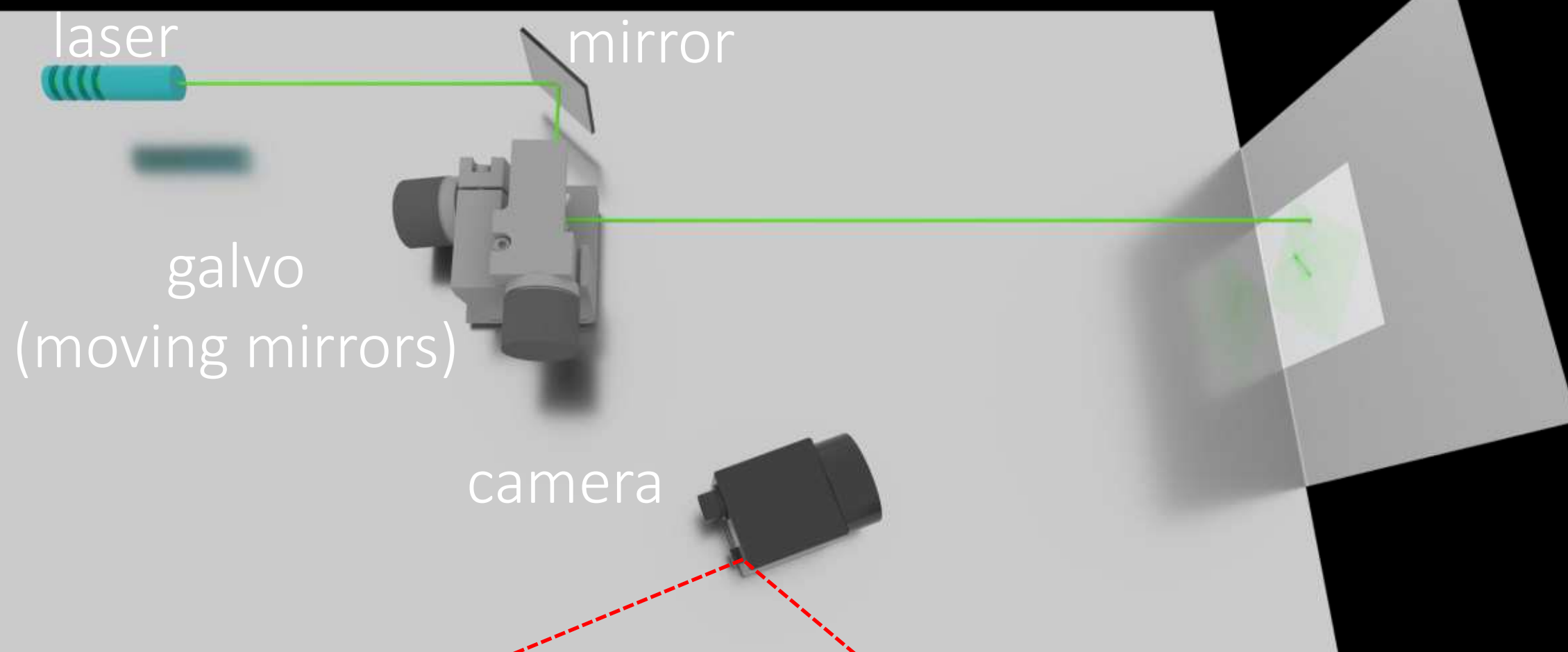


exposure= 1 ms

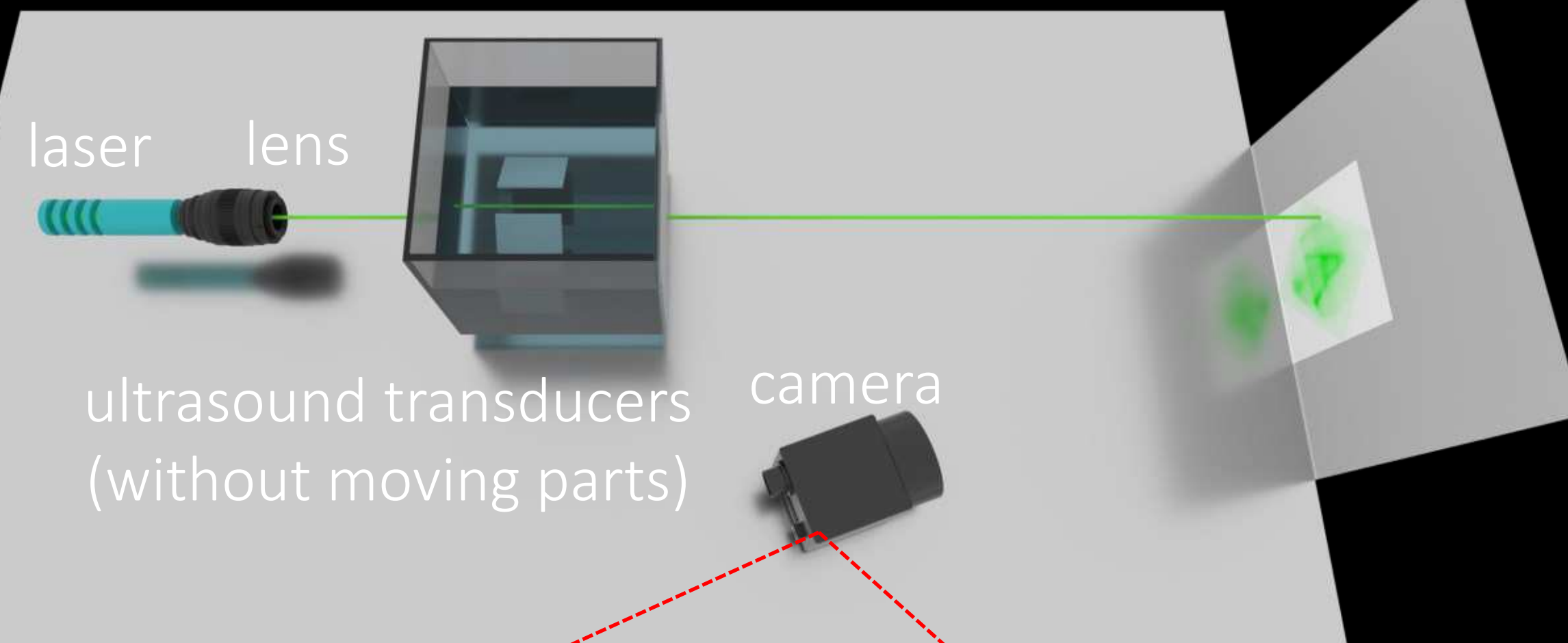


Hardware results: dot projector

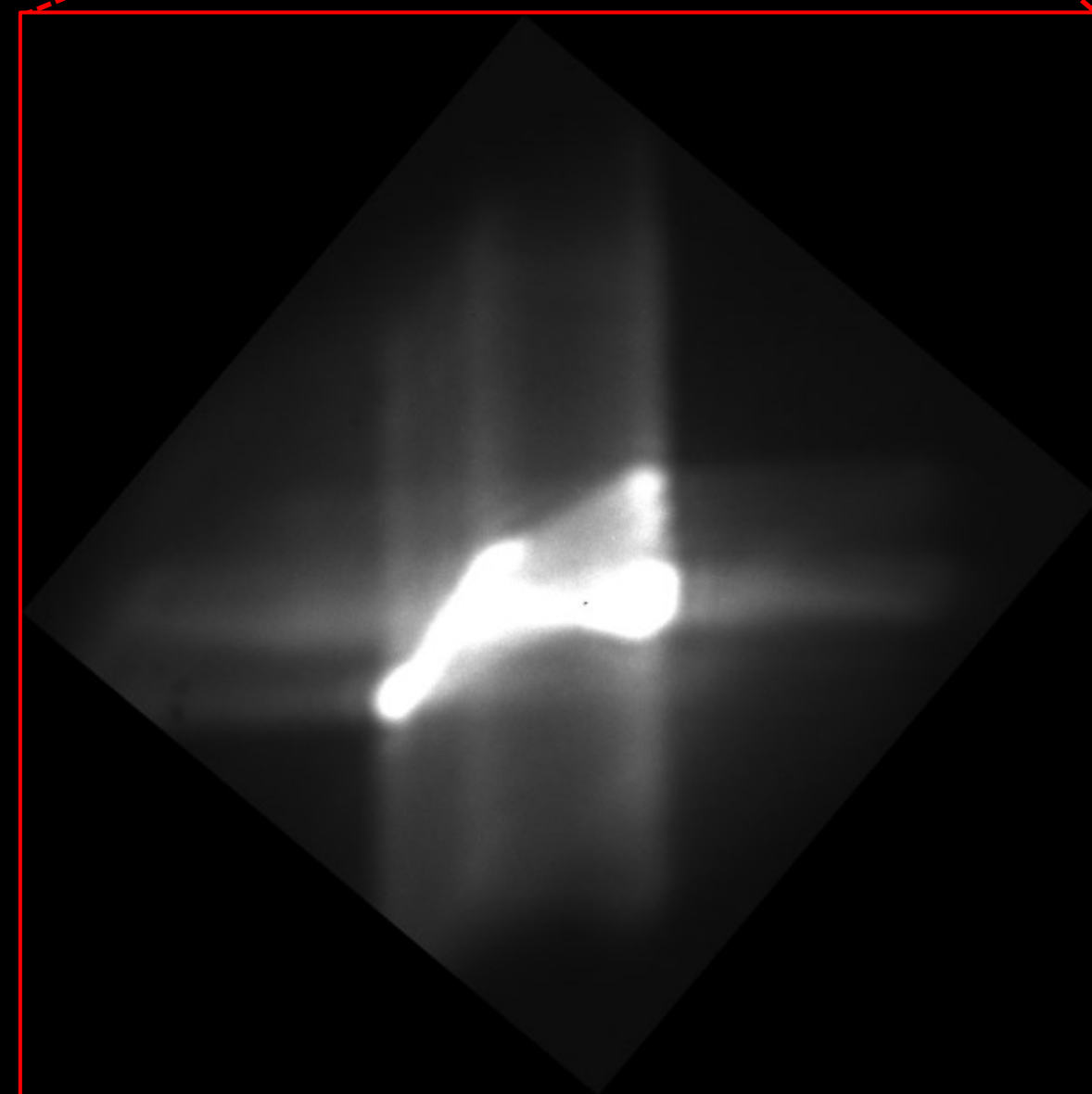
galvo steering (1 kHz)



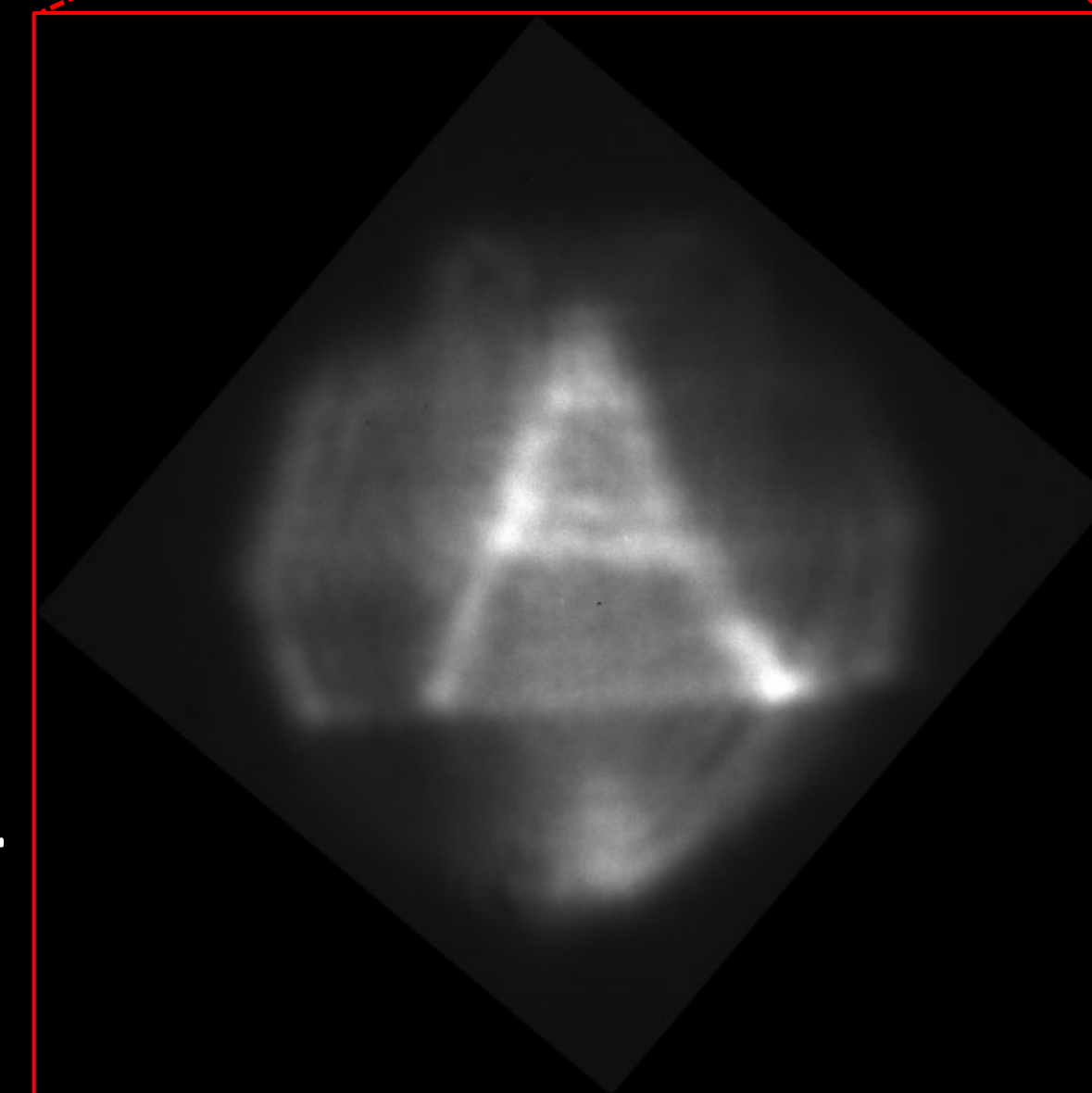
proposed steering (1 MHz)



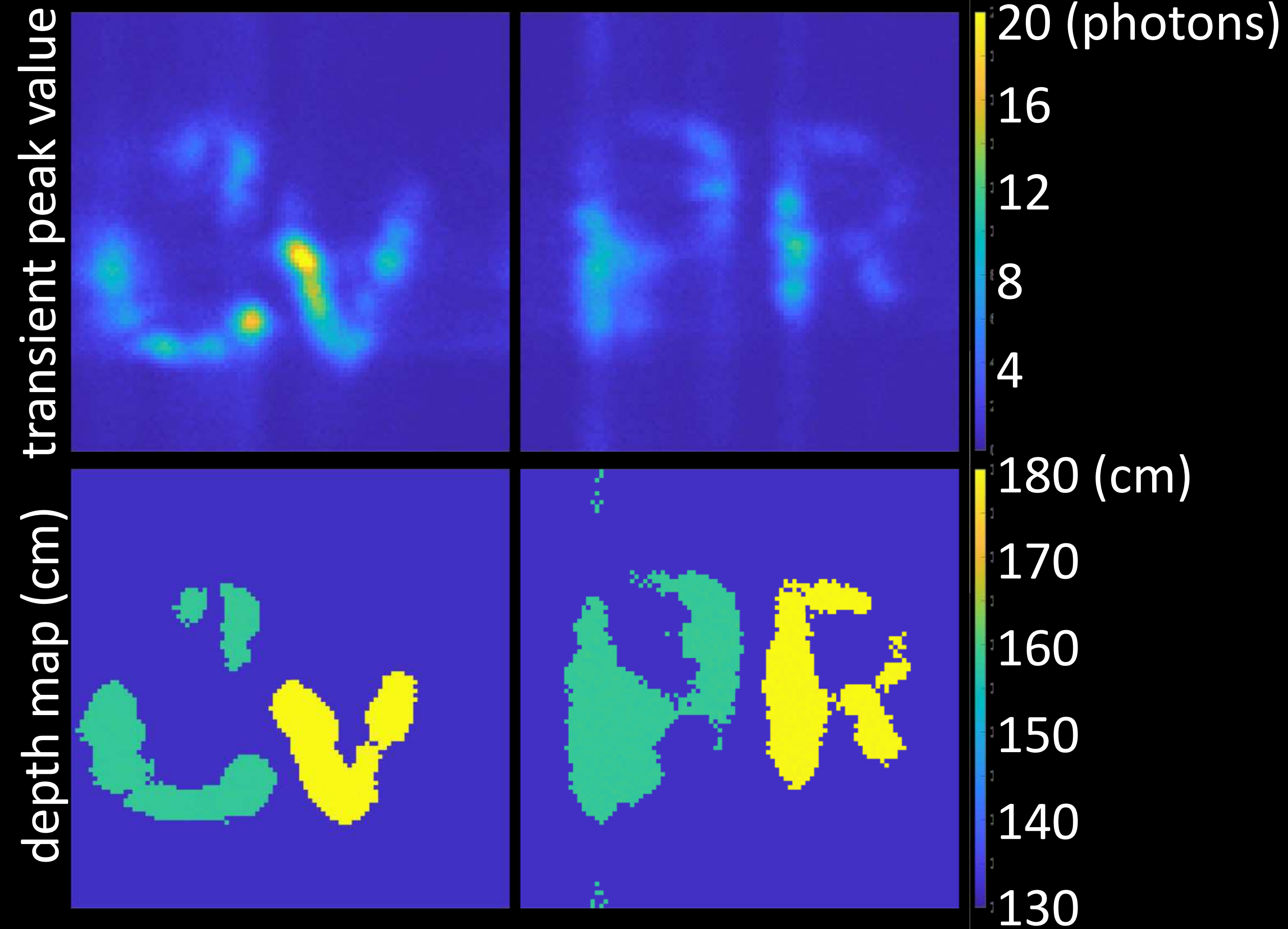
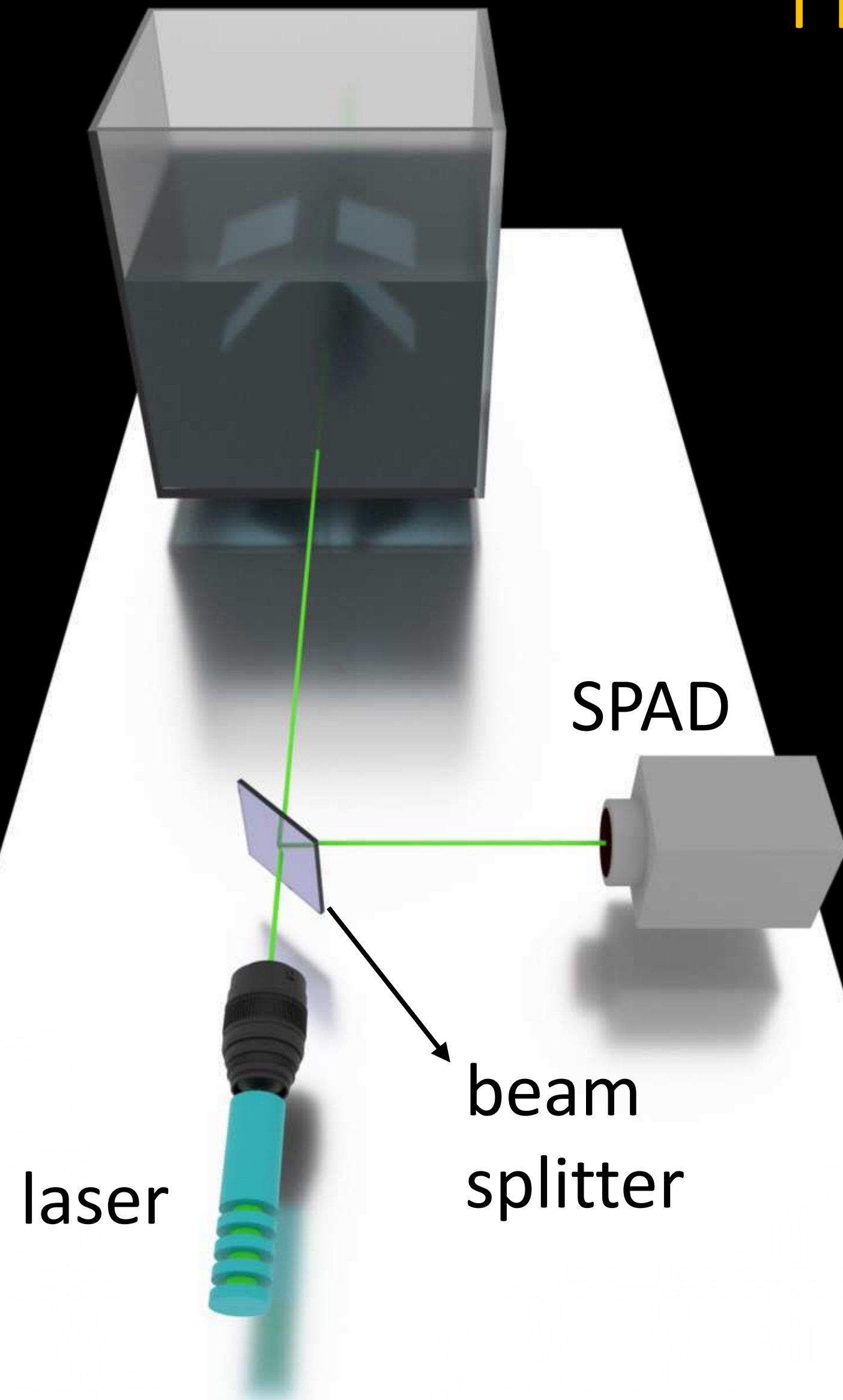
exposure= 50 ms



exposure= 50 ms

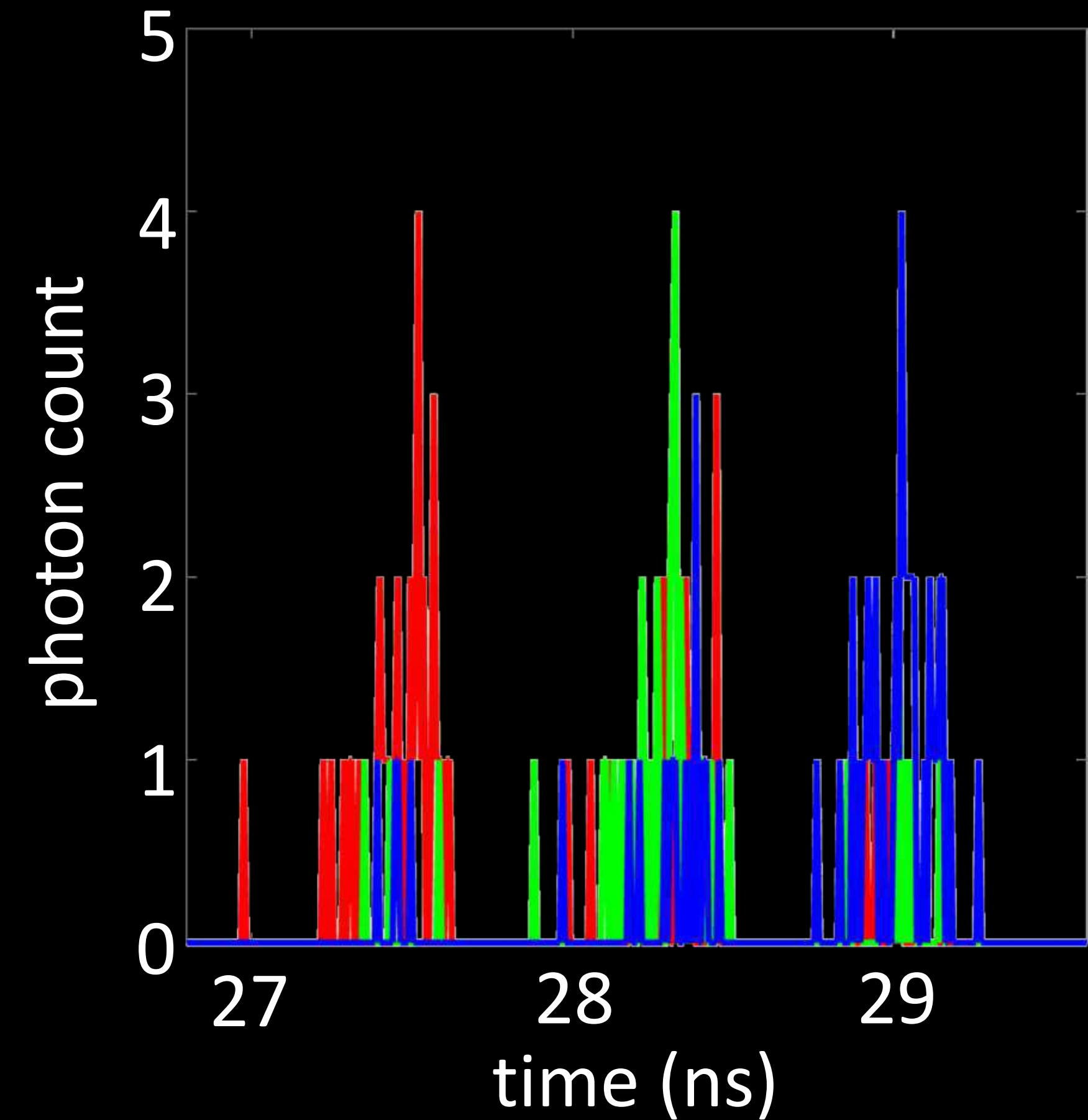
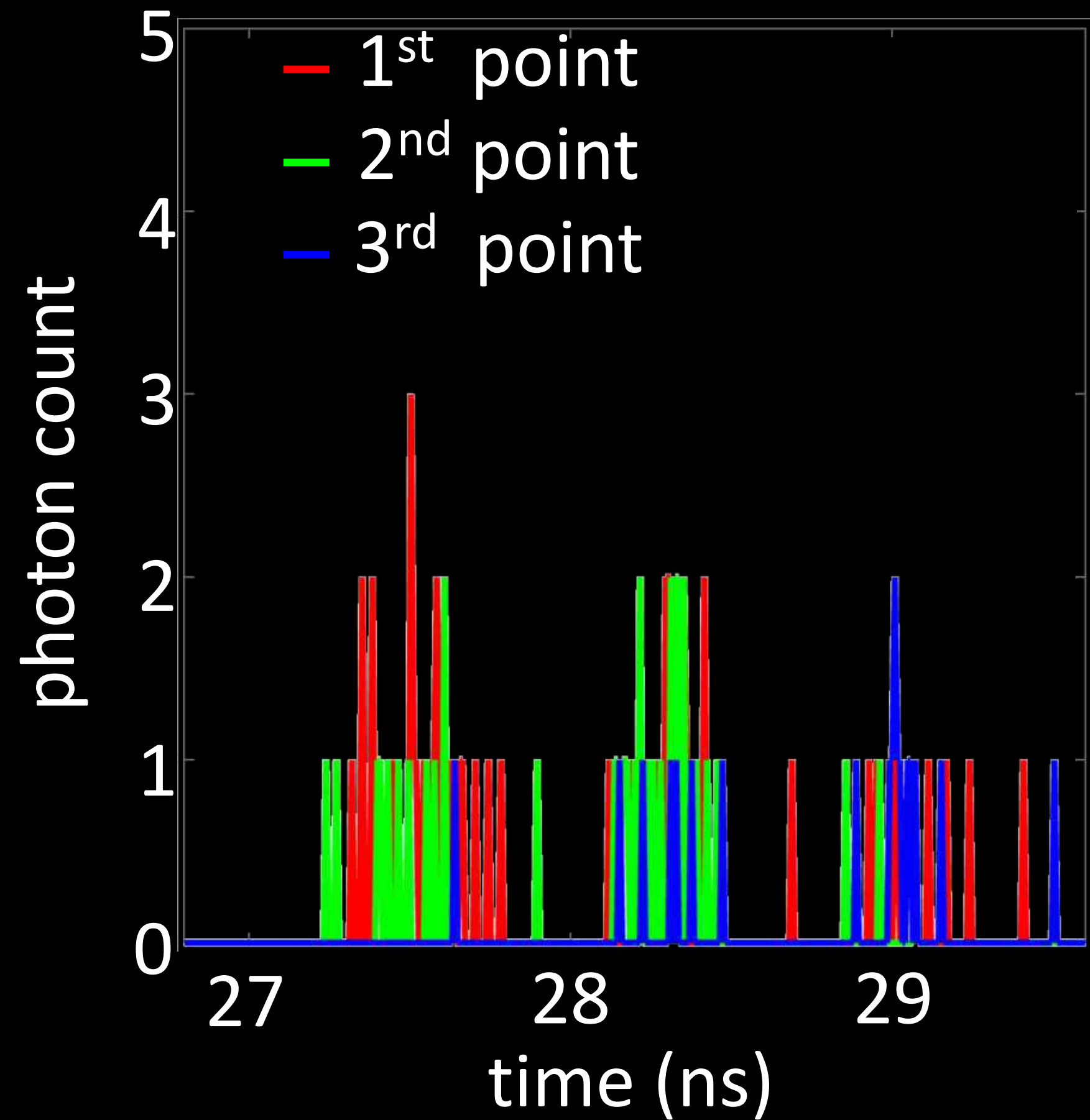


Hardware results: Lidar



100 × faster than standard lidar

Hardware results: adaptive depth measurement

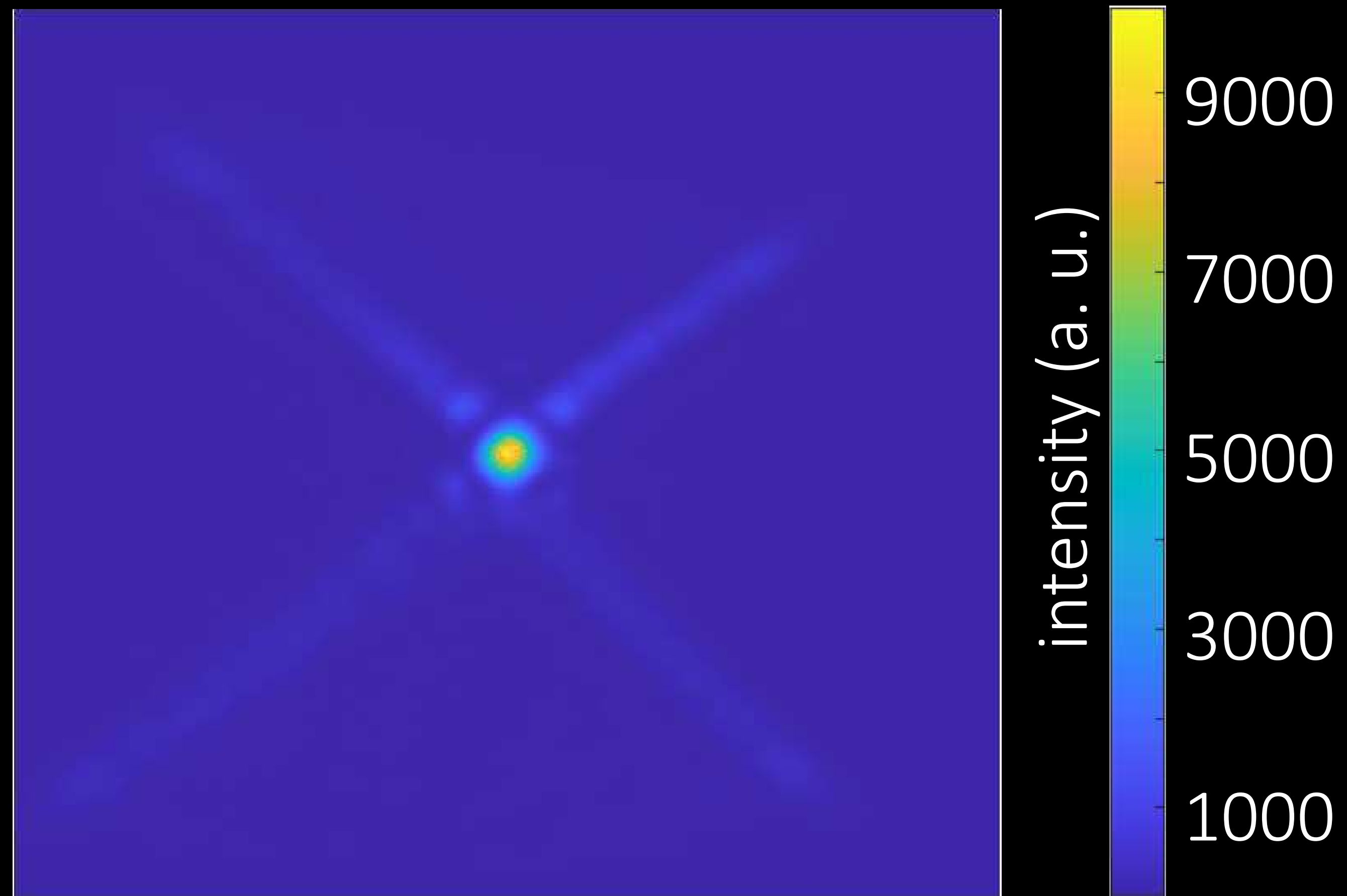
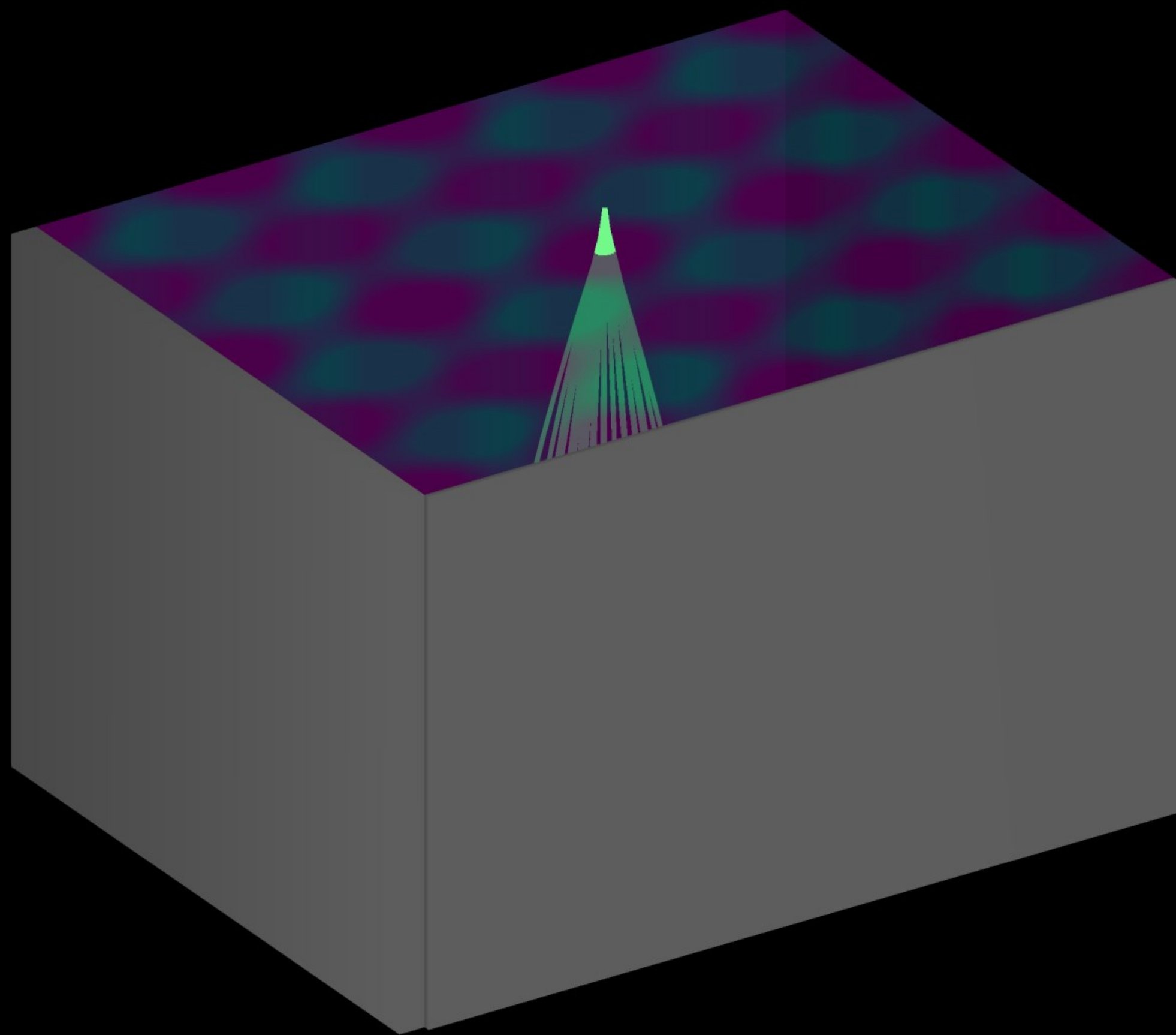


standard galvo (depth error = 51.3 cm)

our technique (depth error = 3.28 cm)

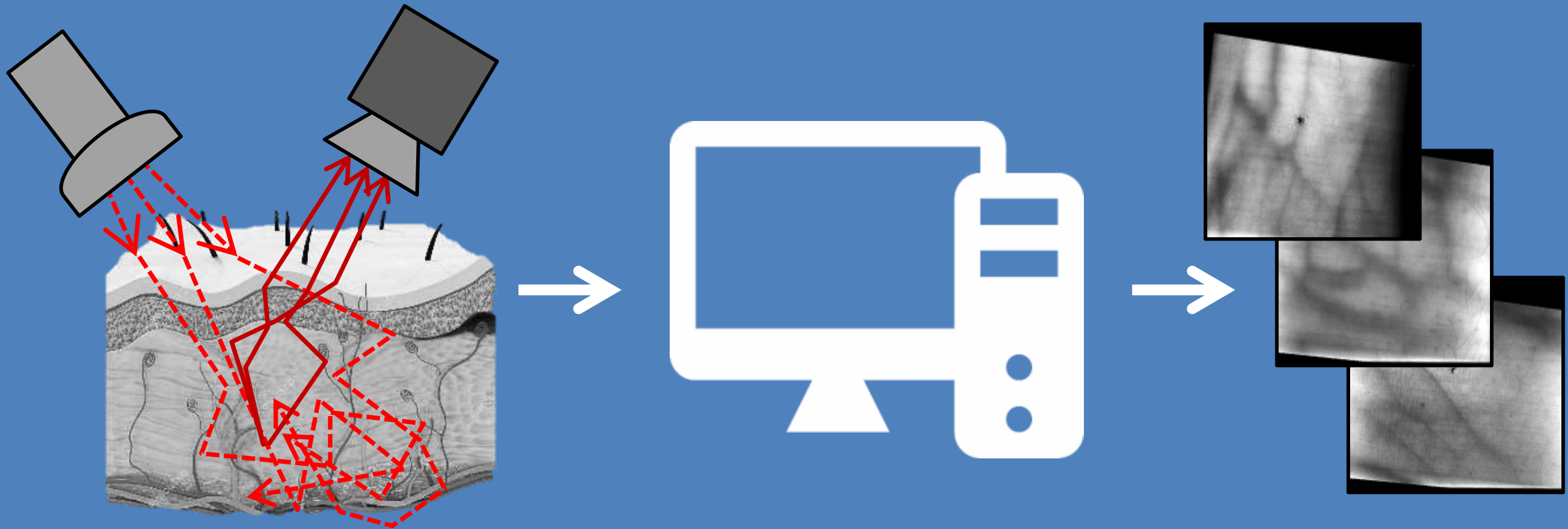
15 × depth accuracy

Limitation



Physics-based rendering and its applications to computational imaging

forward rendering

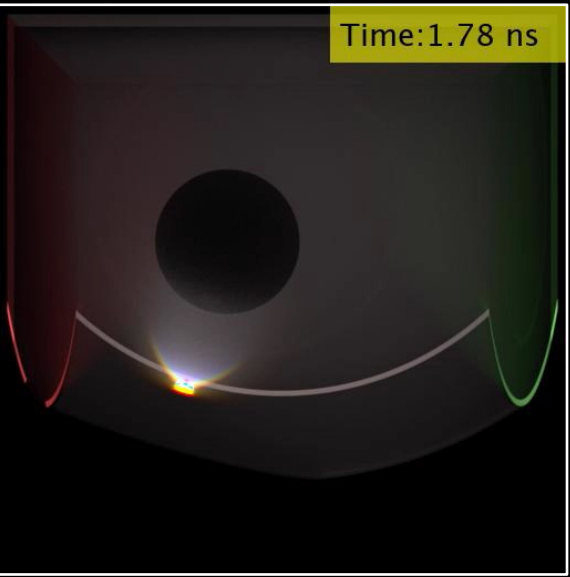


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

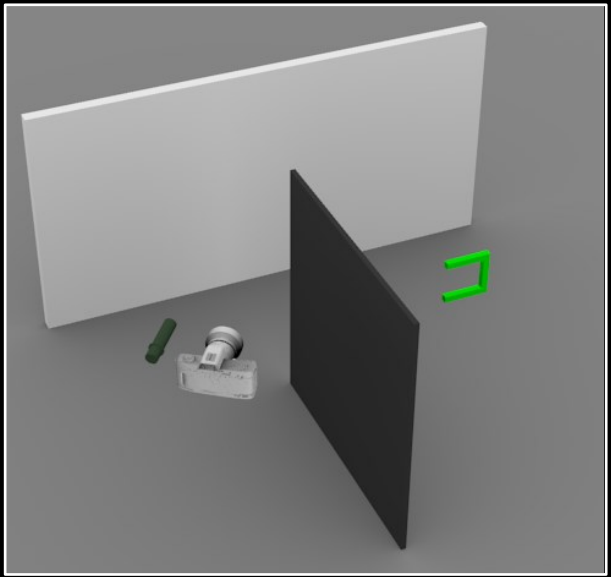
inverse rendering



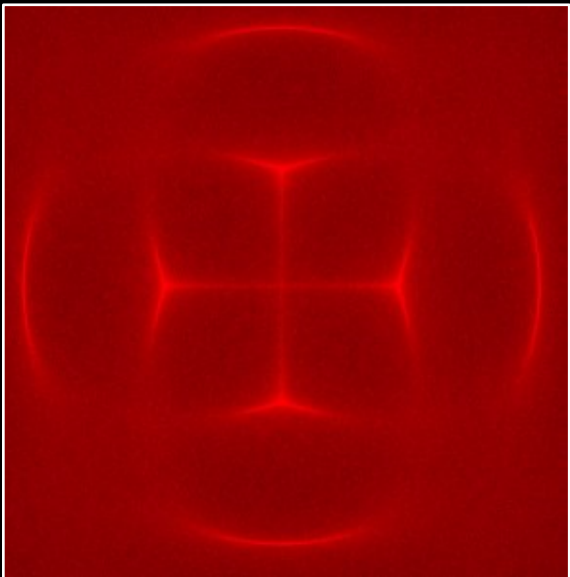
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



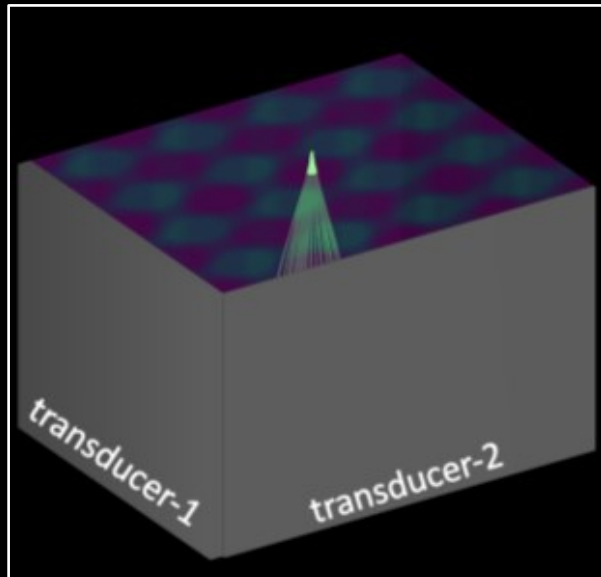
time-of-flight imaging



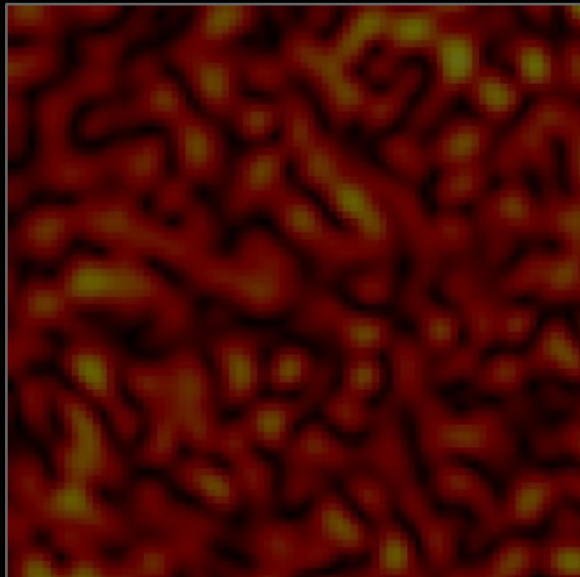
non-line-of-sight imaging



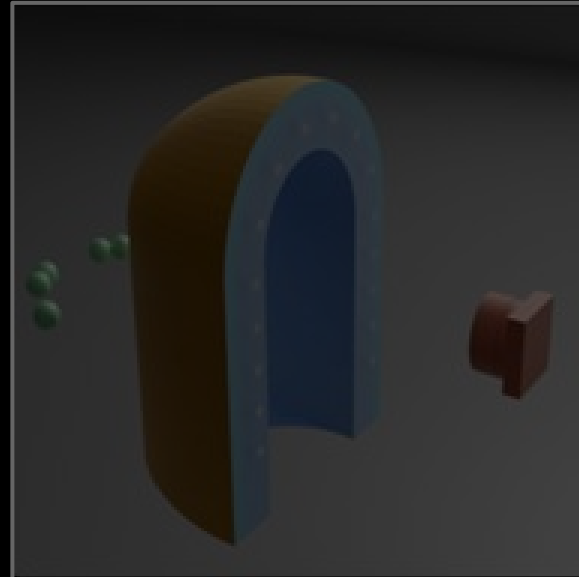
acousto-optic lensing



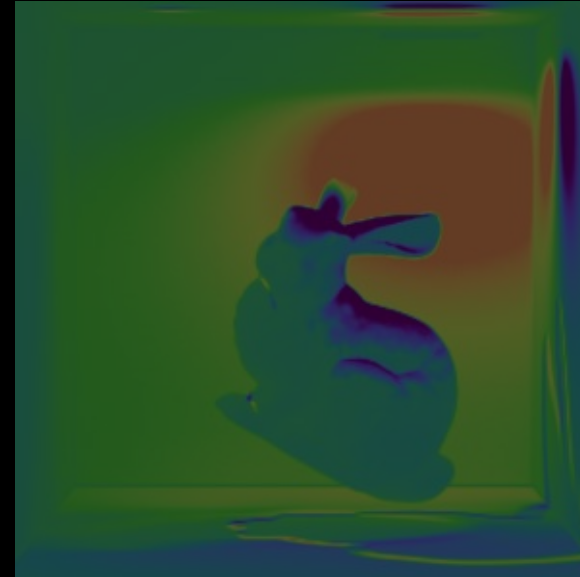
ultrafast light scanning



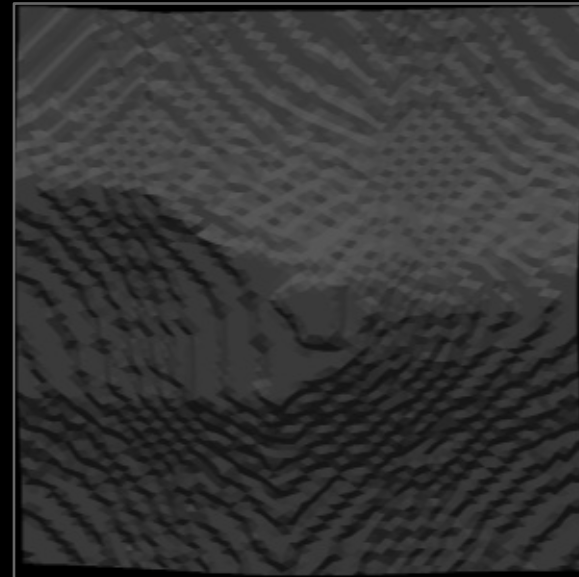
speckle imaging



tactile sensor design



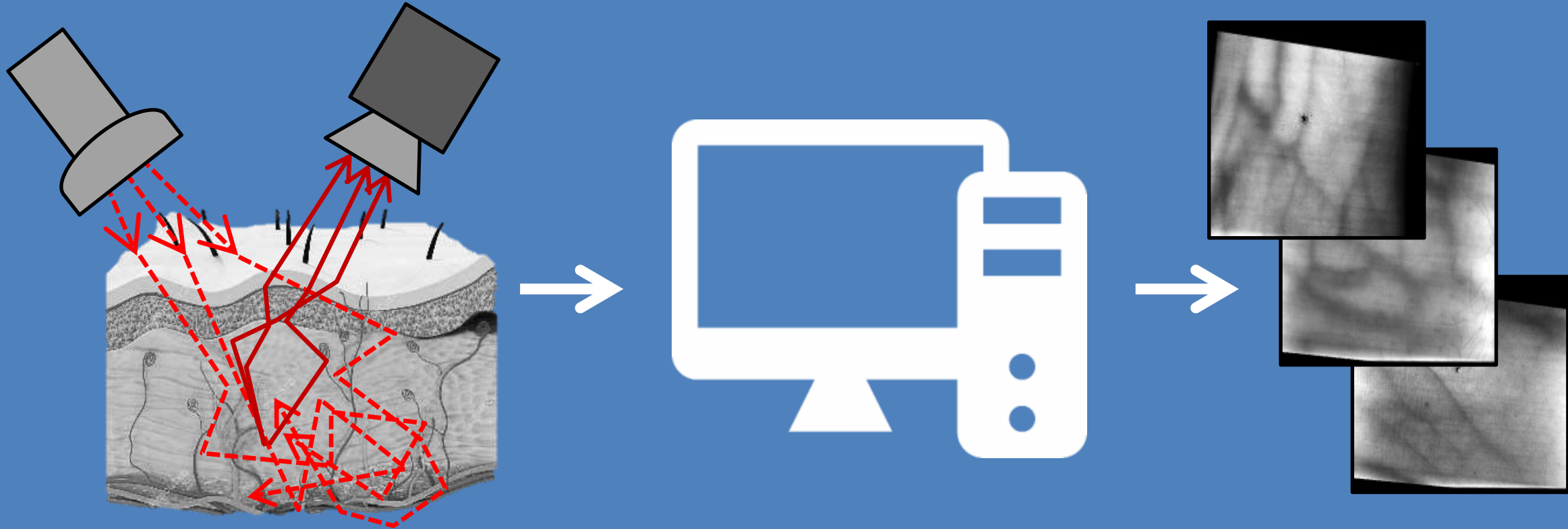
differentiable renderer



inverse problems

Physics-based rendering and its applications to computational imaging

forward rendering

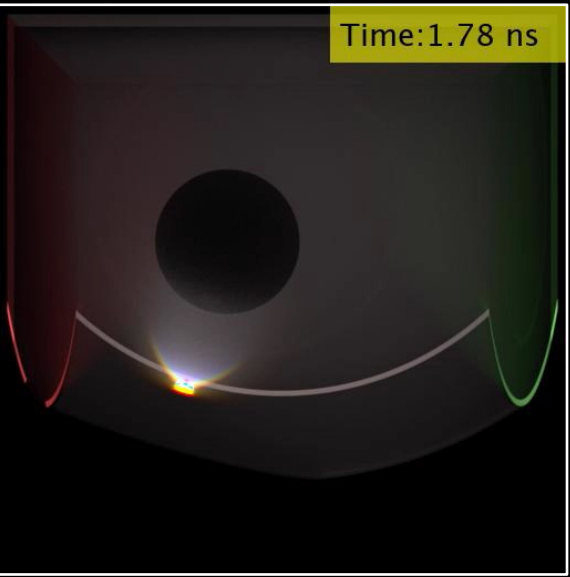


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

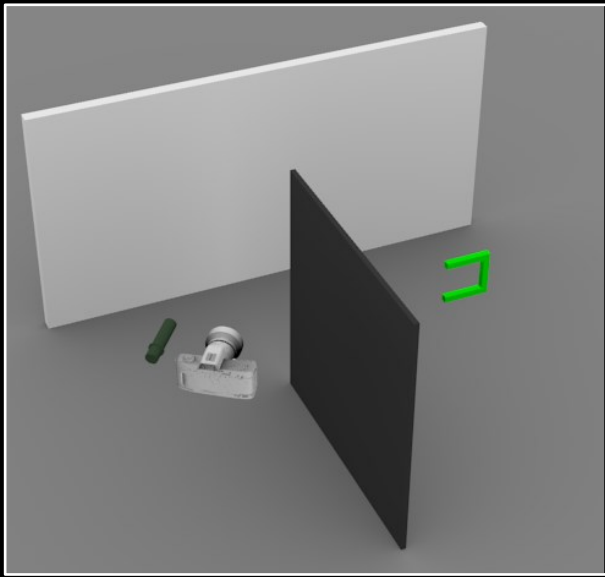
inverse rendering



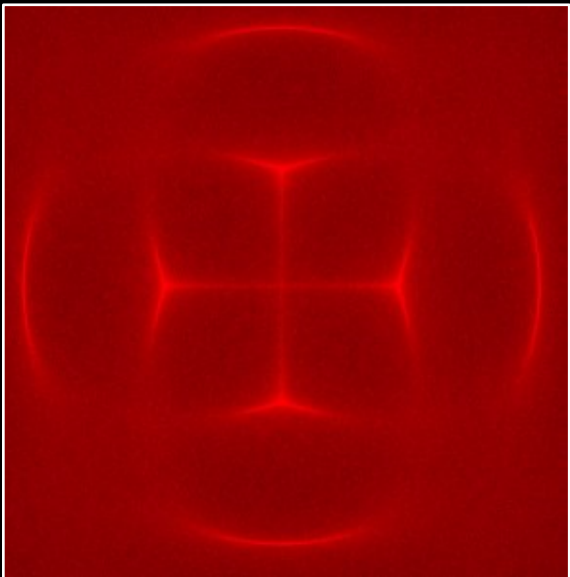
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



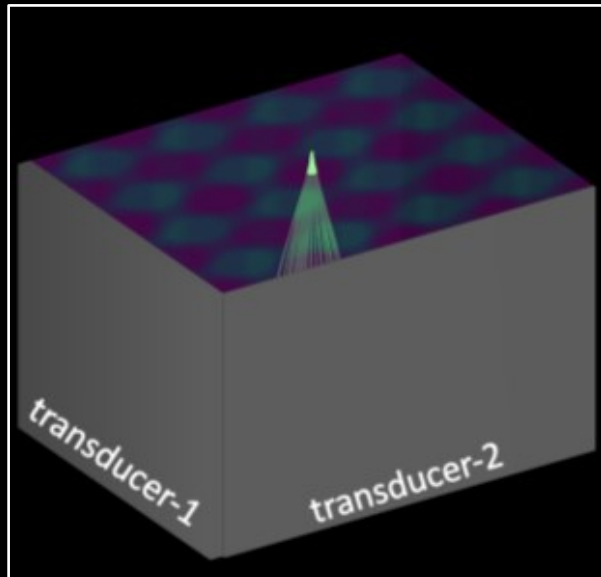
time-of-flight
imaging



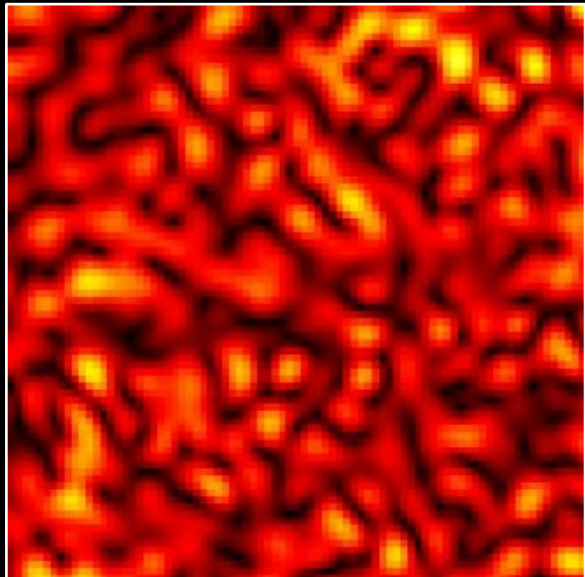
non-line-of-sight
imaging



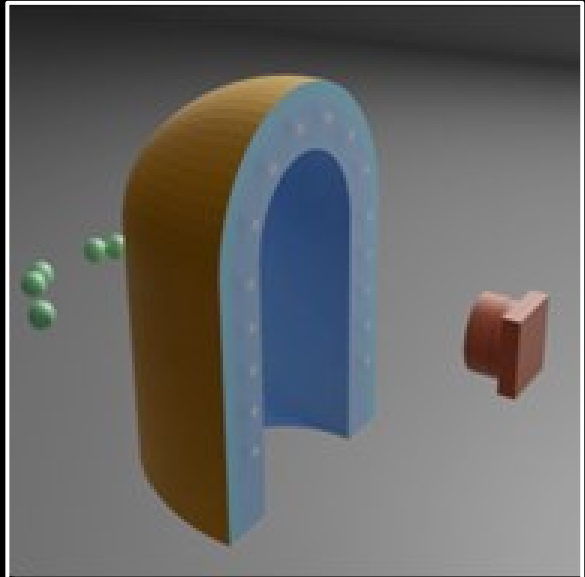
acousto-optic
lensing



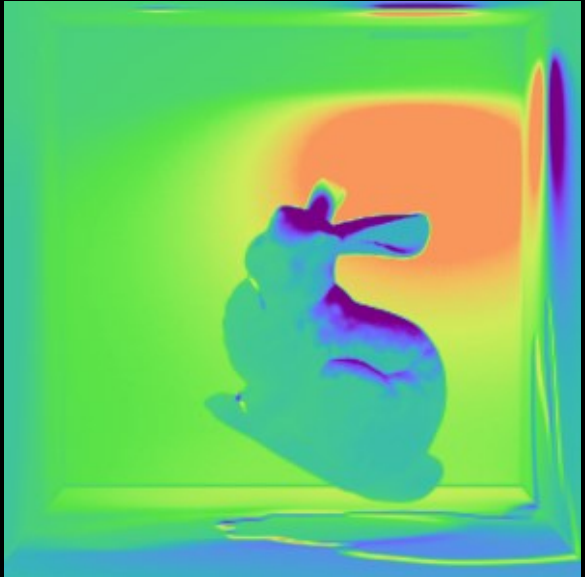
ultrafast light
scanning



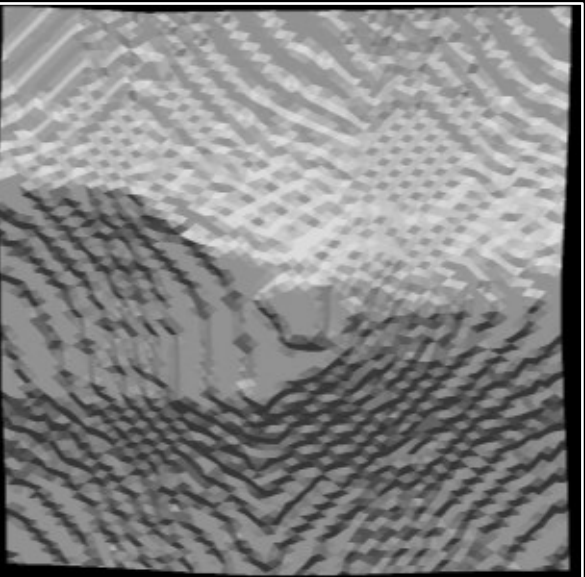
speckle
imaging



tactile sensor
design



differentiable
renderer



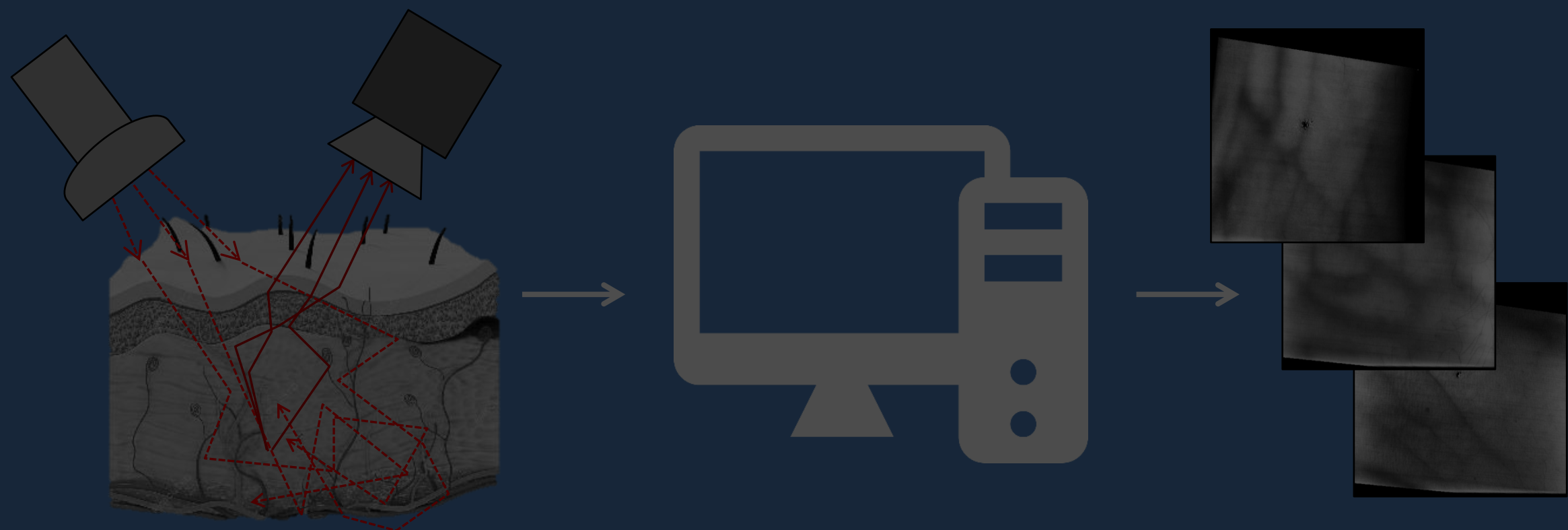
inverse
problems

Coffee break
See you at 3.30 PM



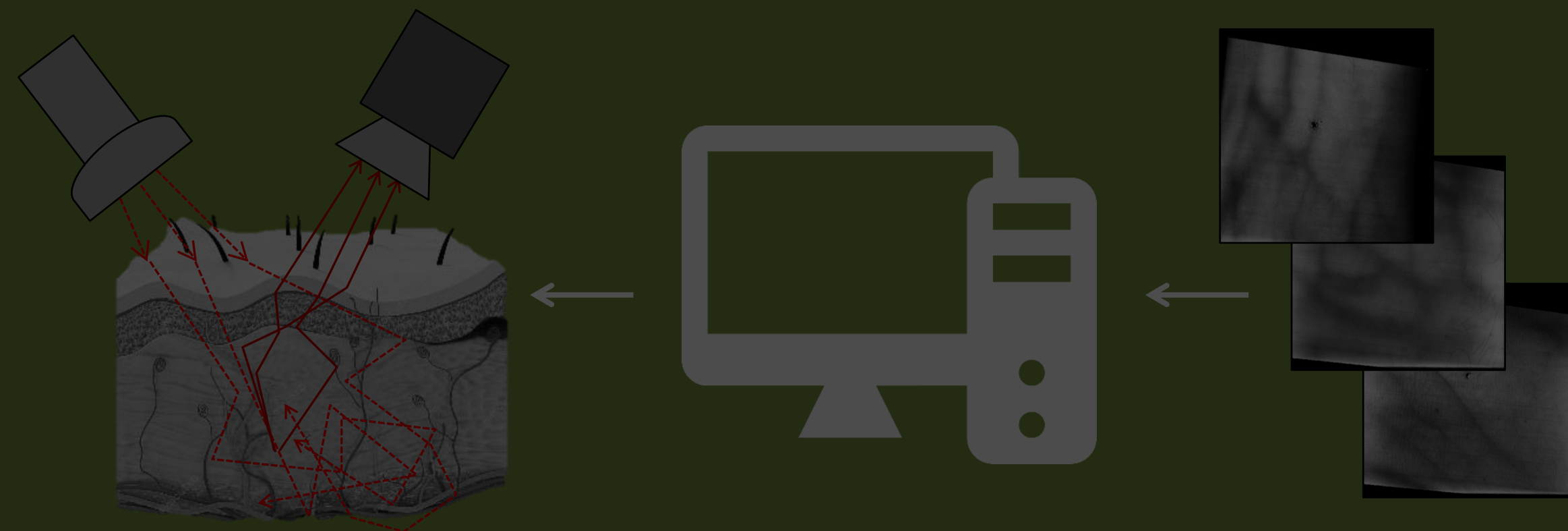
Physics-based rendering and its applications to computational imaging

forward rendering

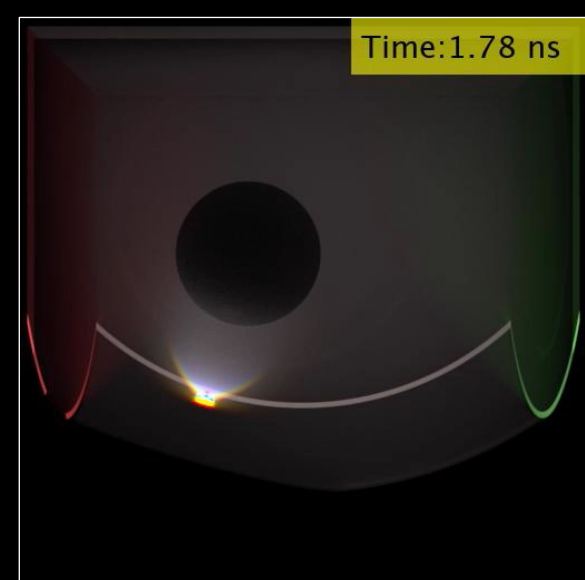


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

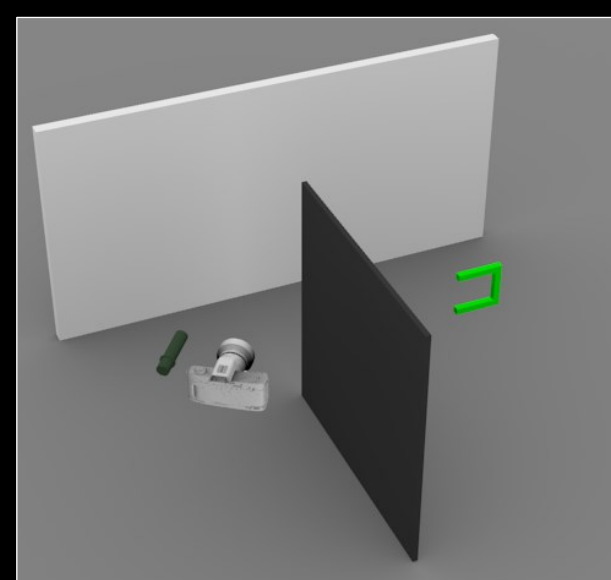
inverse rendering



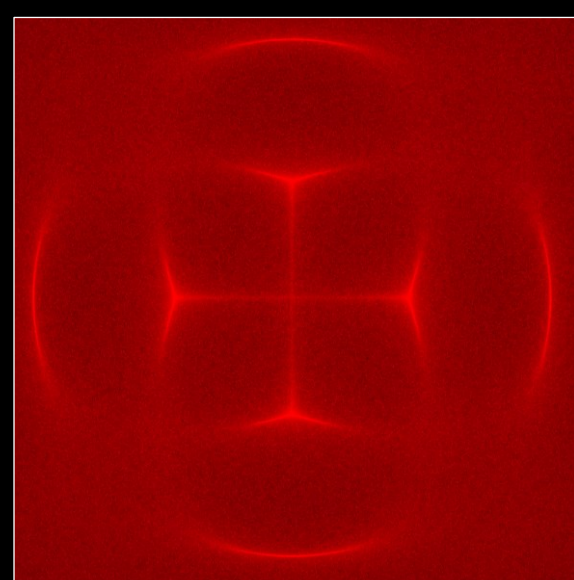
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



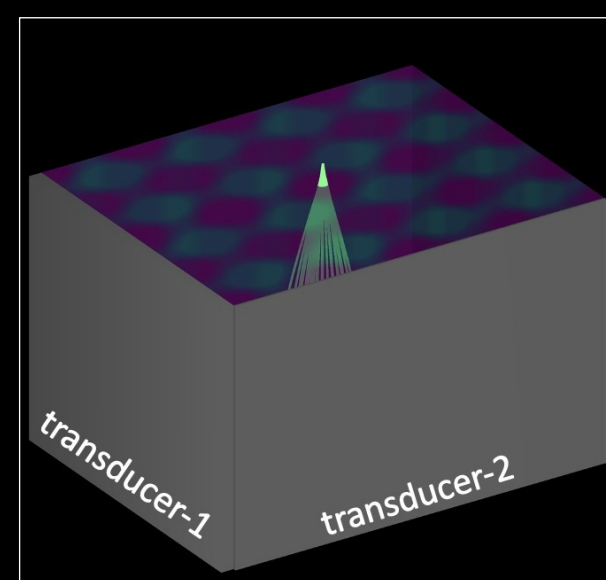
time-of-flight
imaging



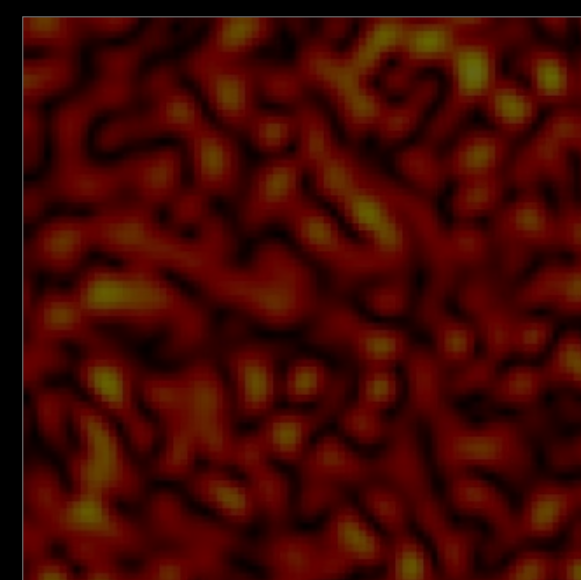
non-line-of-sight
imaging



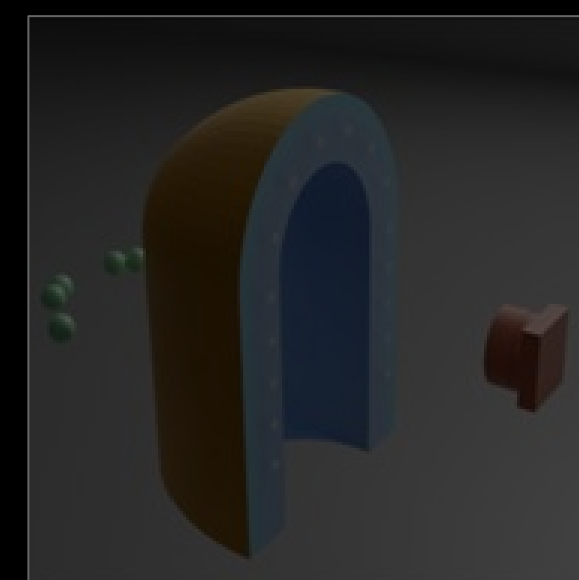
acousto-optic
lensing



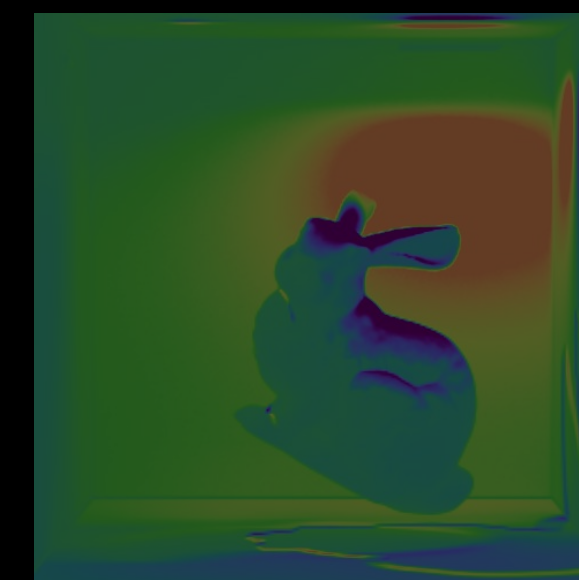
ultrafast light
scanning



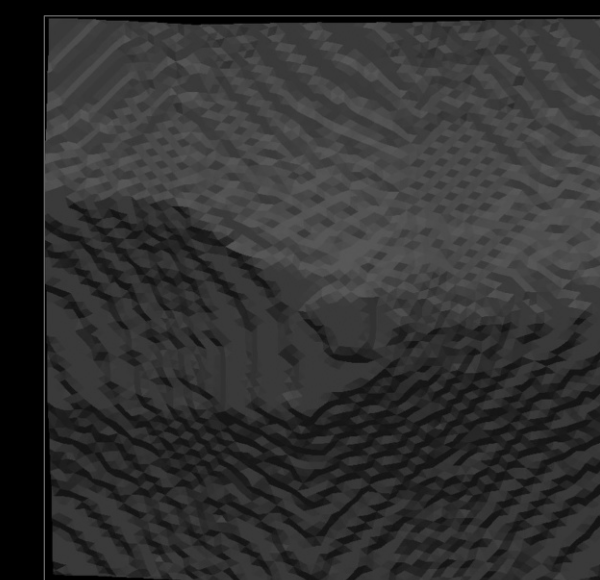
speckle
imaging



tactile sensor
design



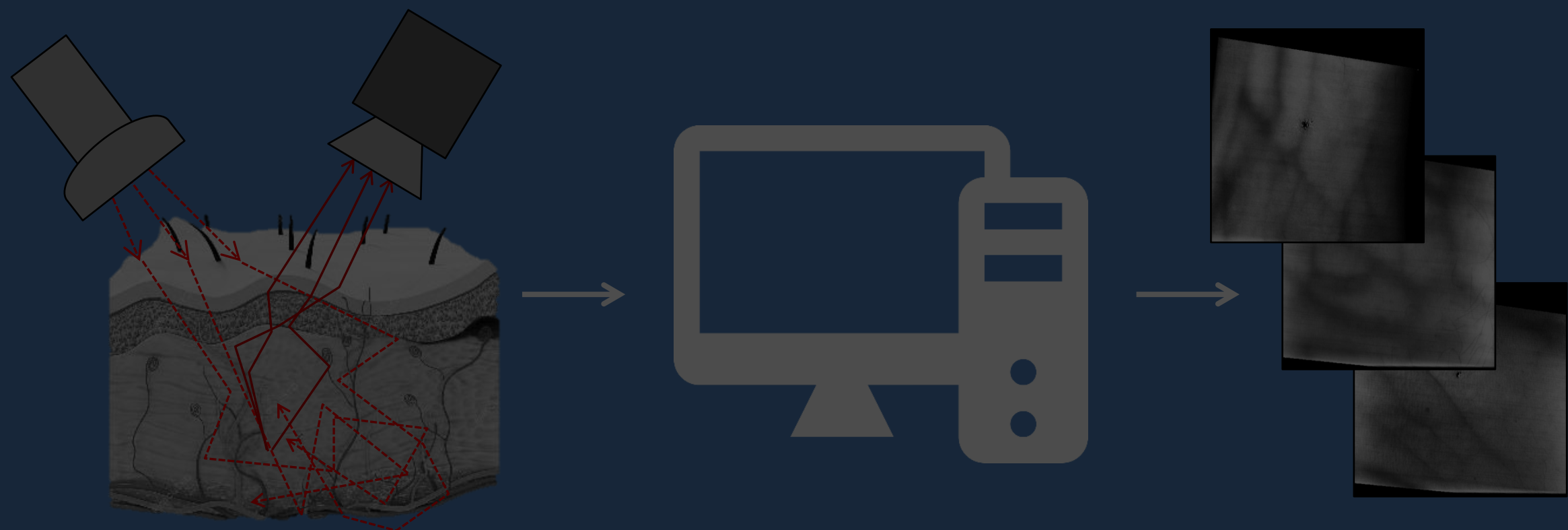
differentiable
renderer



inverse
problems

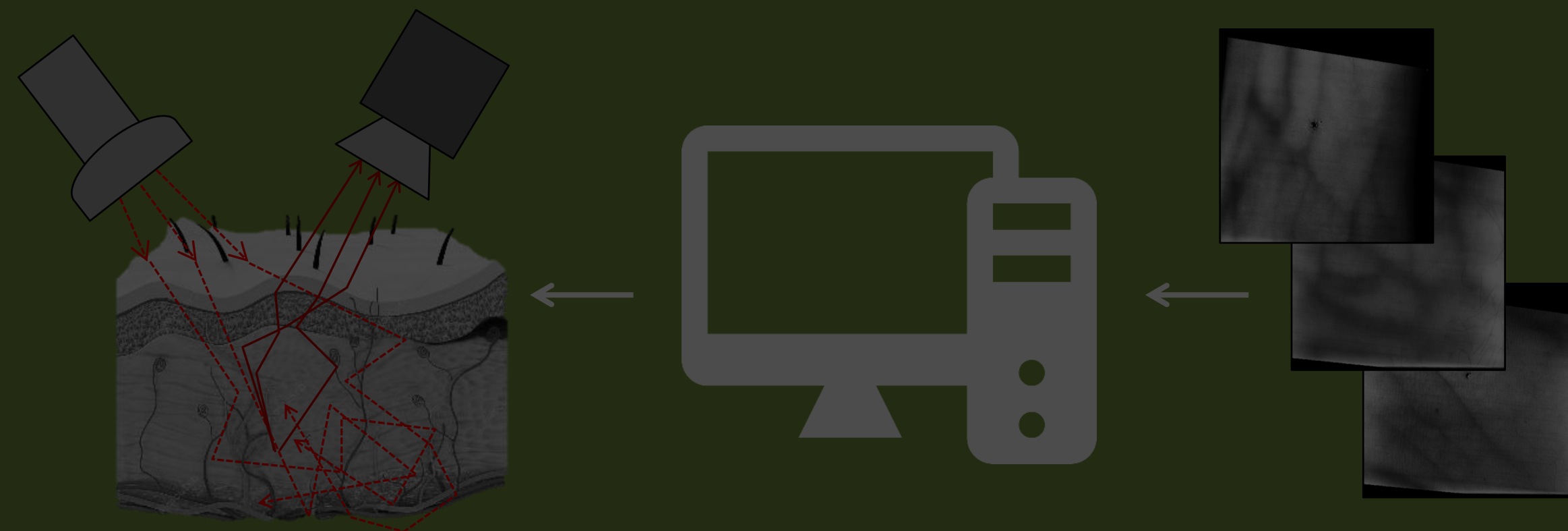
Physics-based rendering and its applications to computational imaging

forward rendering

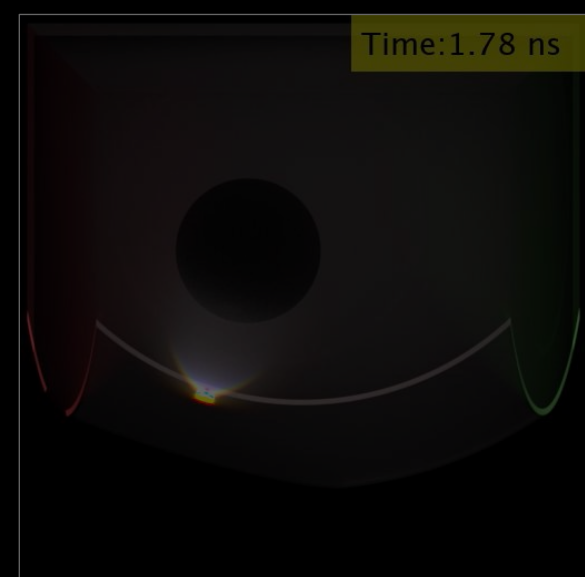


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

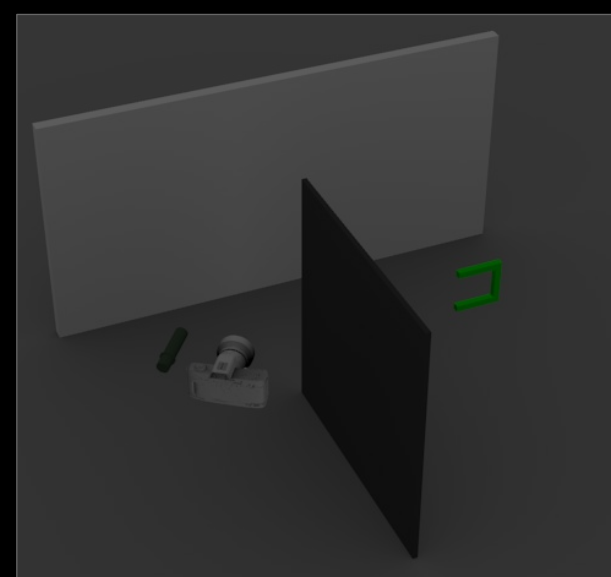
inverse rendering



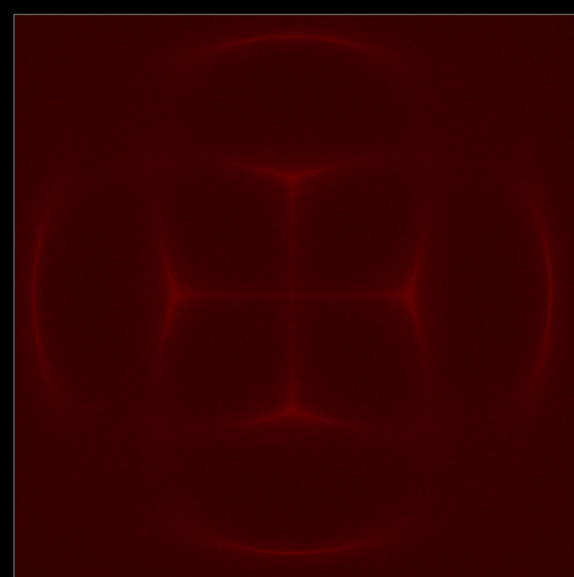
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



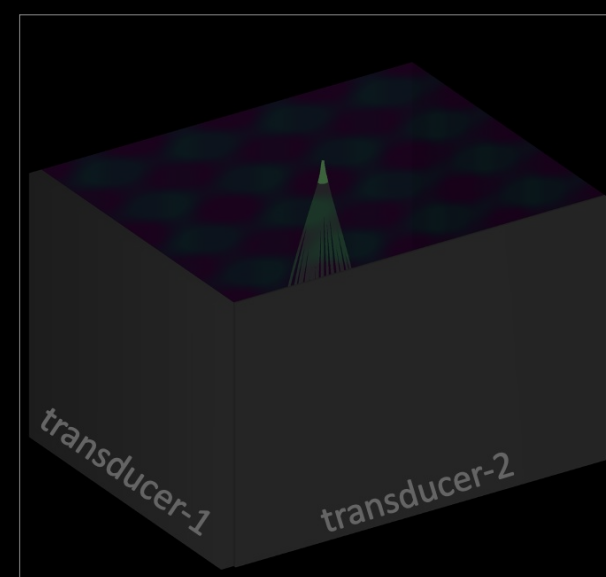
time-of-flight
imaging



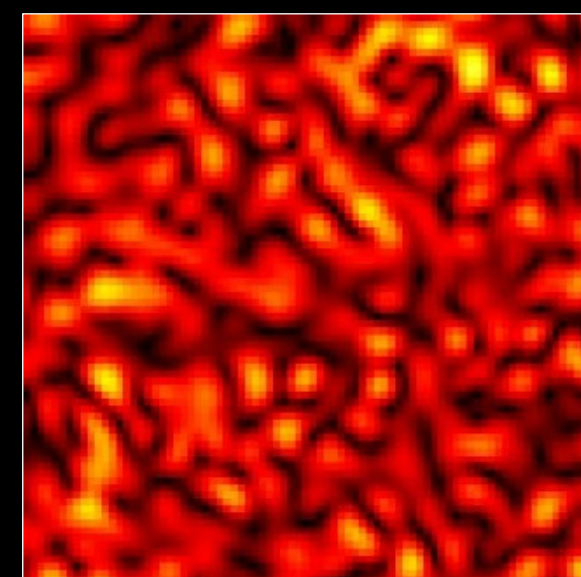
non-line-of-sight
imaging



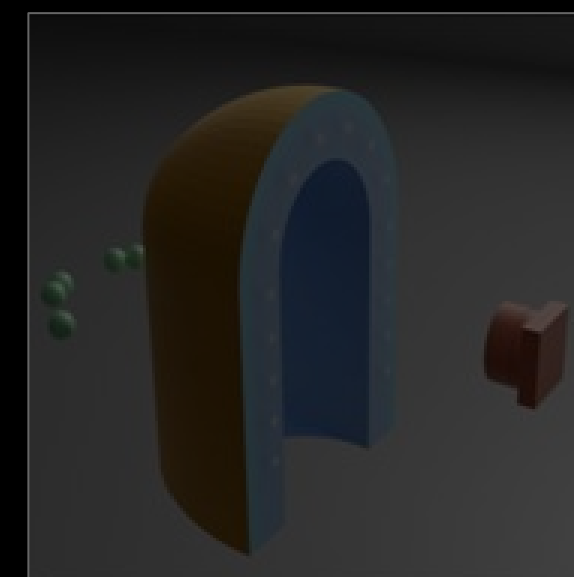
acousto-optic
lensing



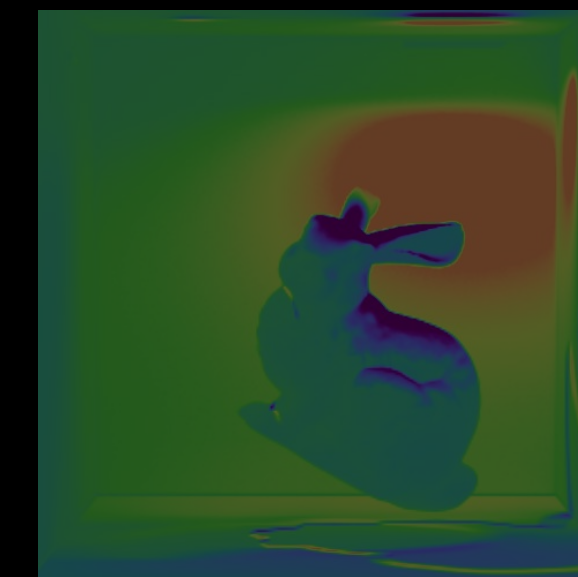
ultrafast light
scanning



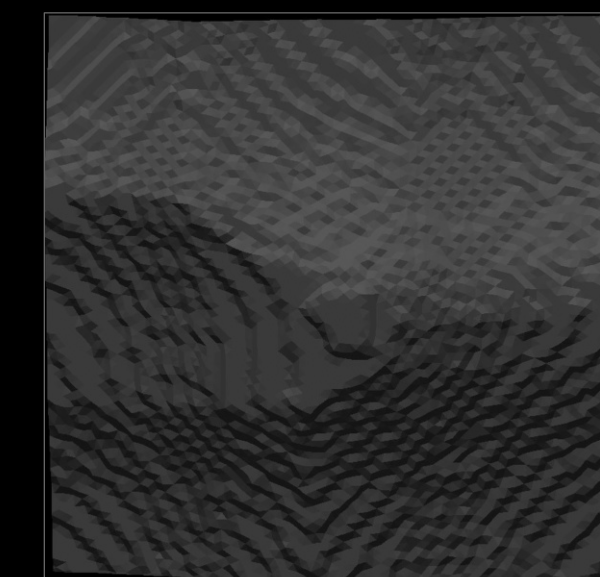
speckle
imaging



tactile sensor
design

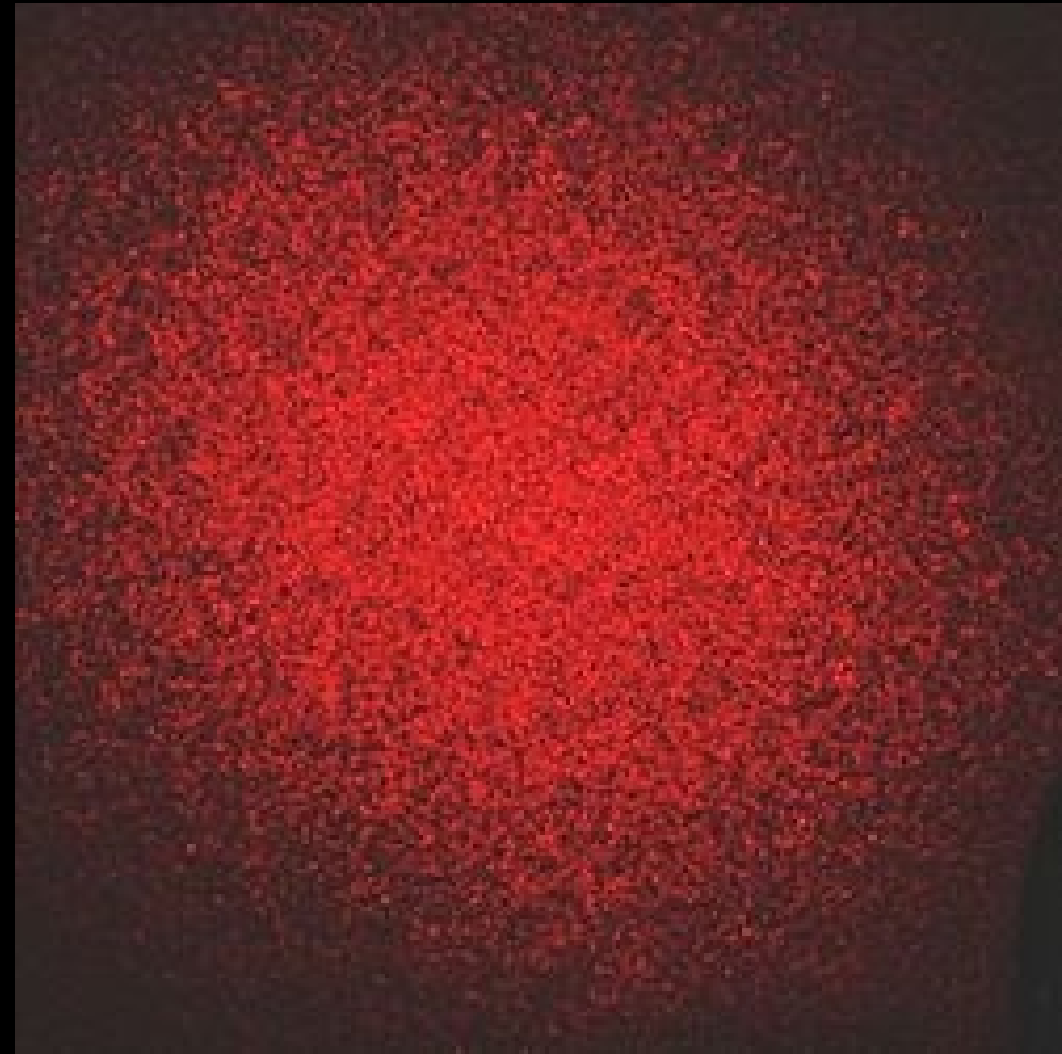


differentiable
renderer



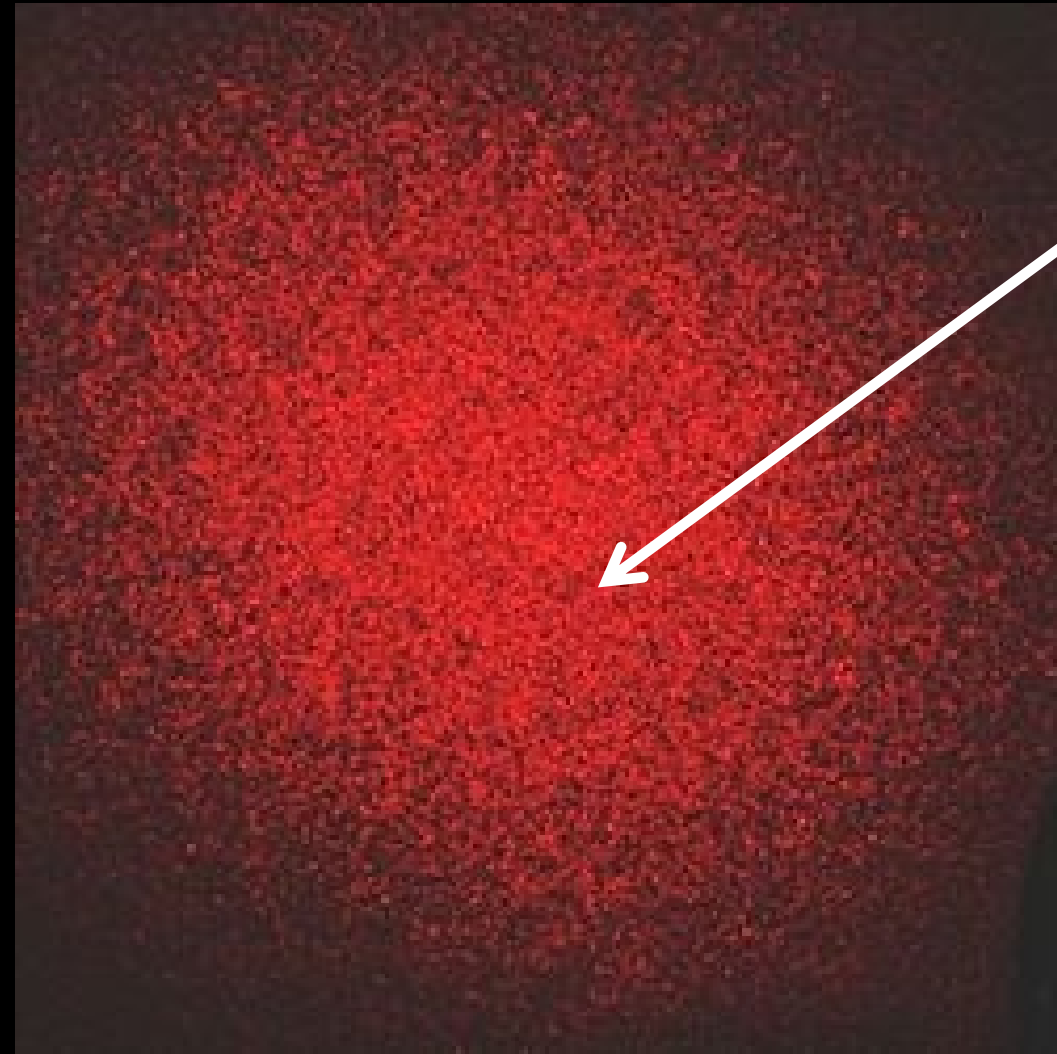
inverse
problems

Rendering wave-optics effects



what real laser
images look like

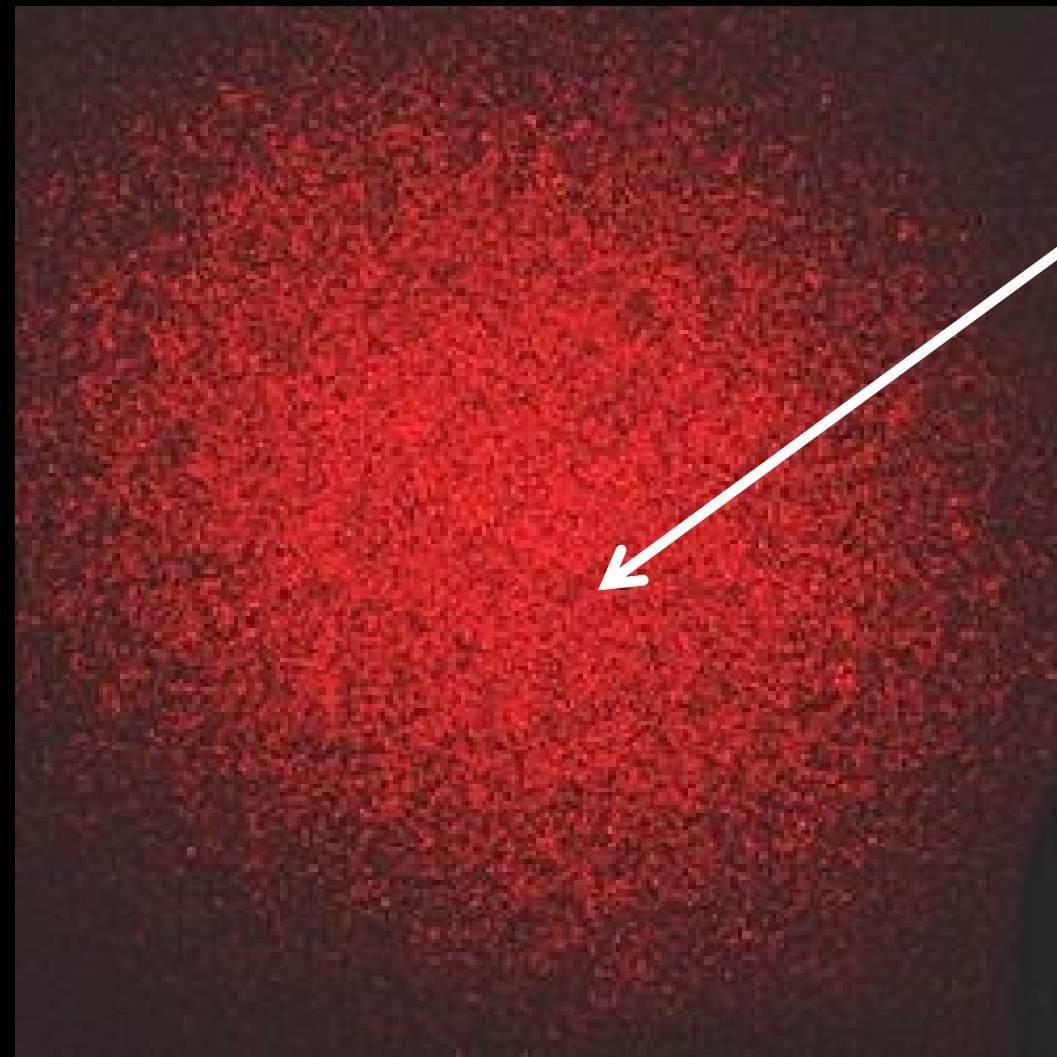
Rendering wave-optics effects



speckle: noise-
like pattern

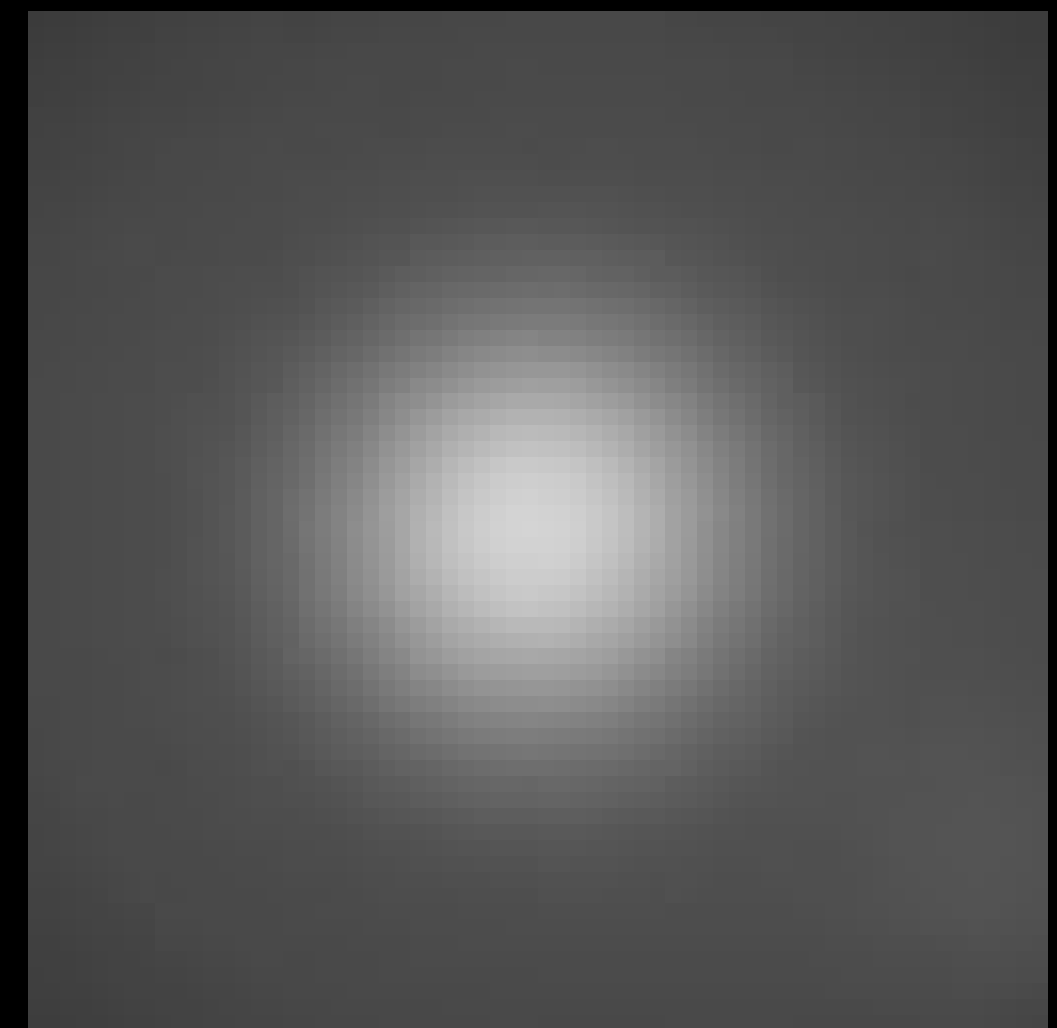
what real laser
images look like

Rendering wave-optics effects



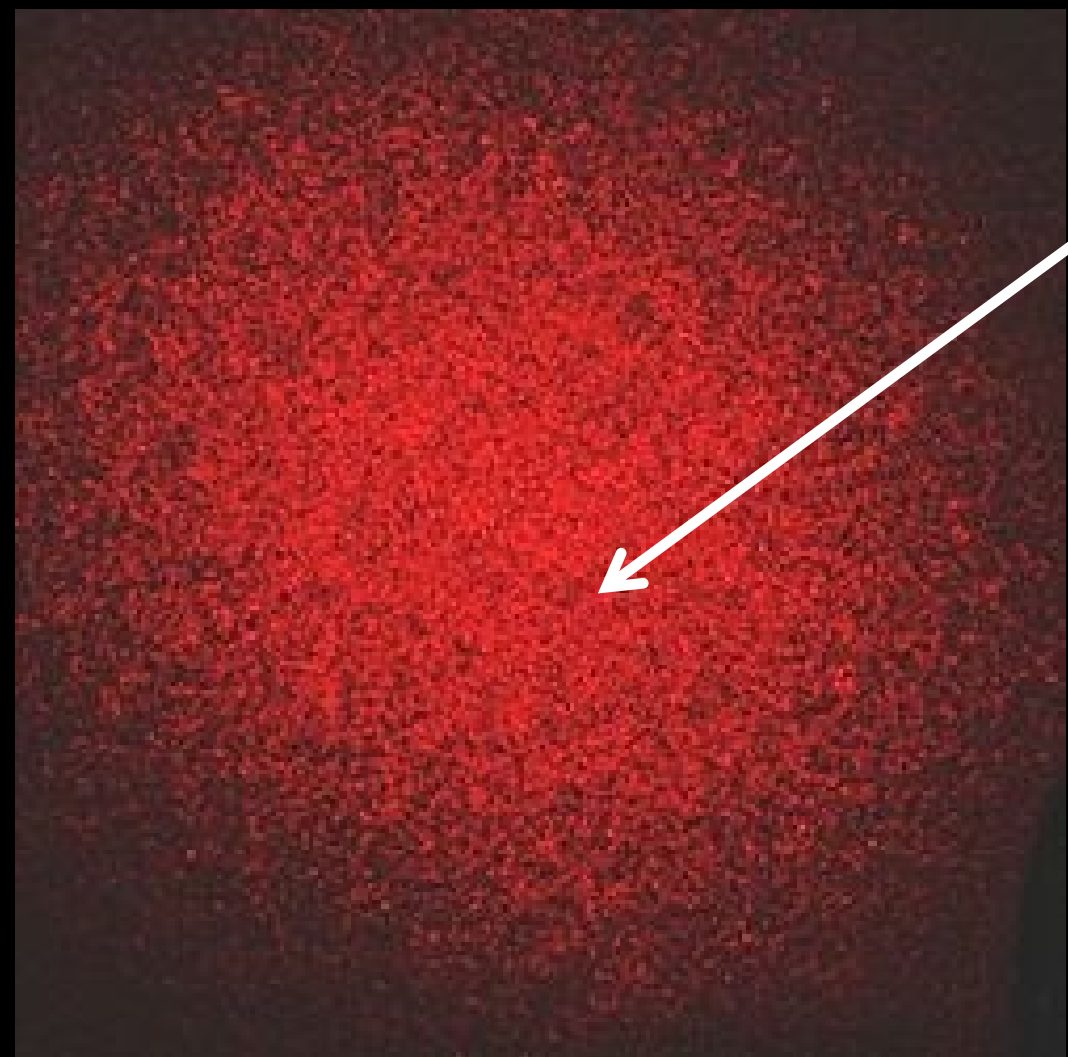
speckle: noise-
like pattern

what real laser
images look like

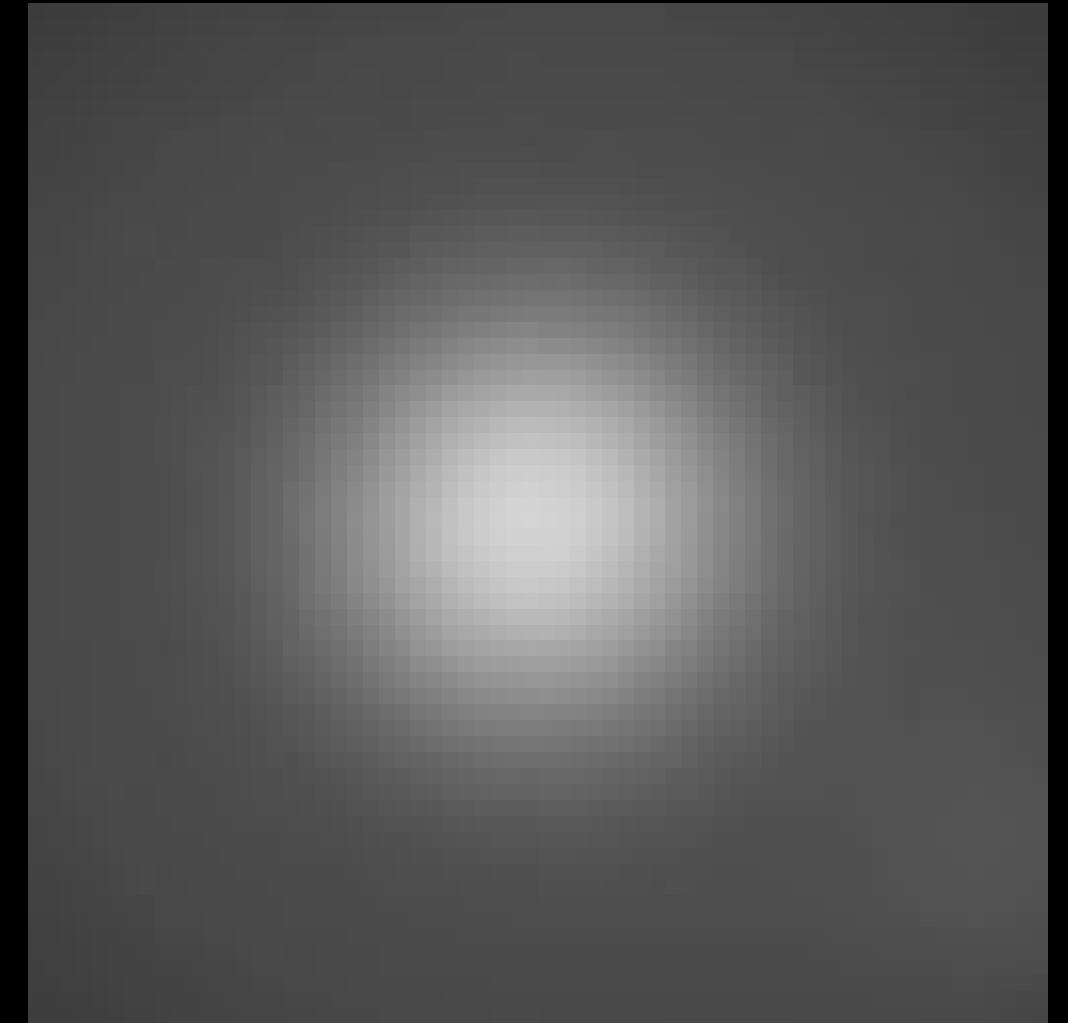


what standard
Monte Carlo
renderings look like

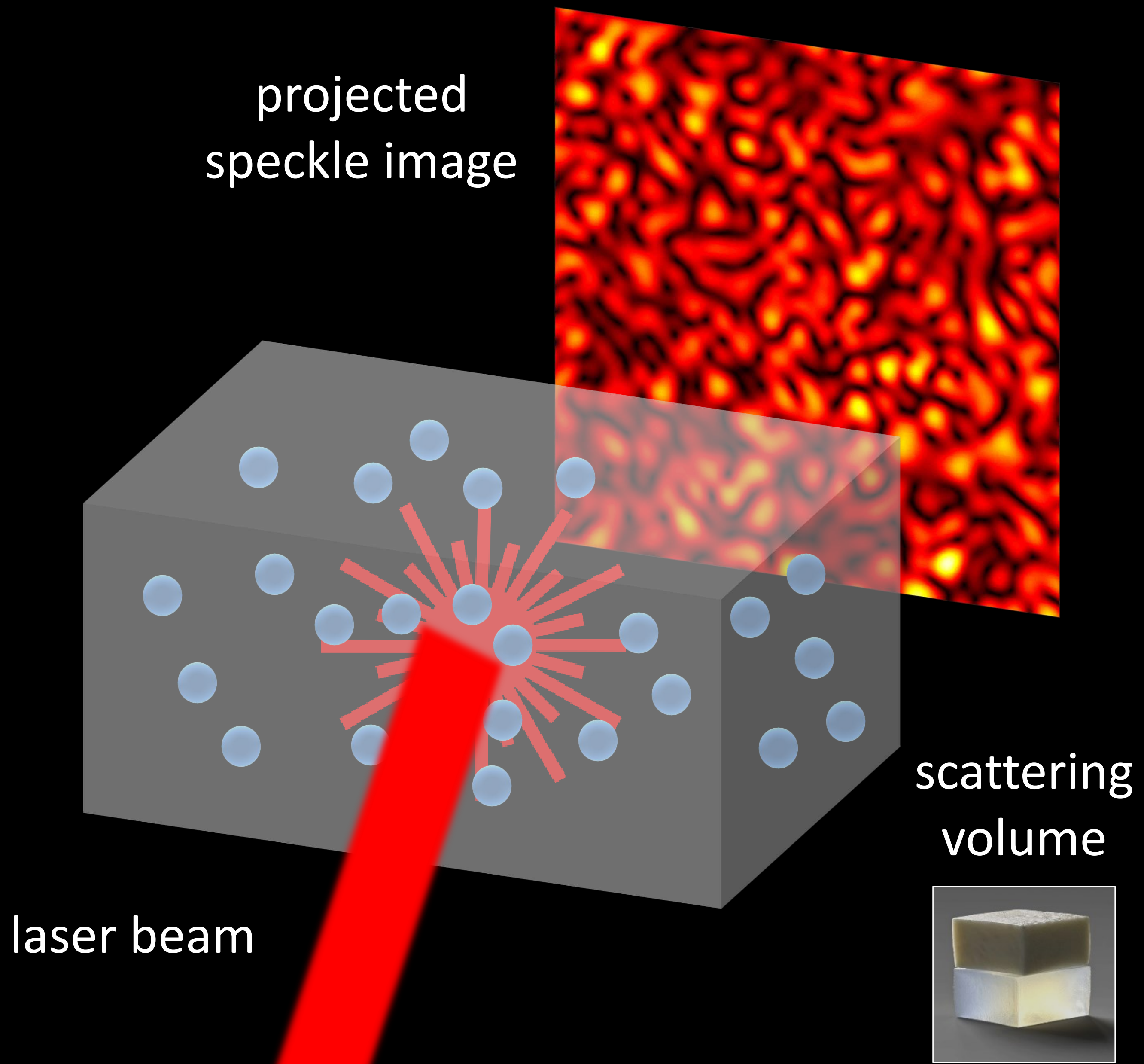
Rendering wave-optics effects



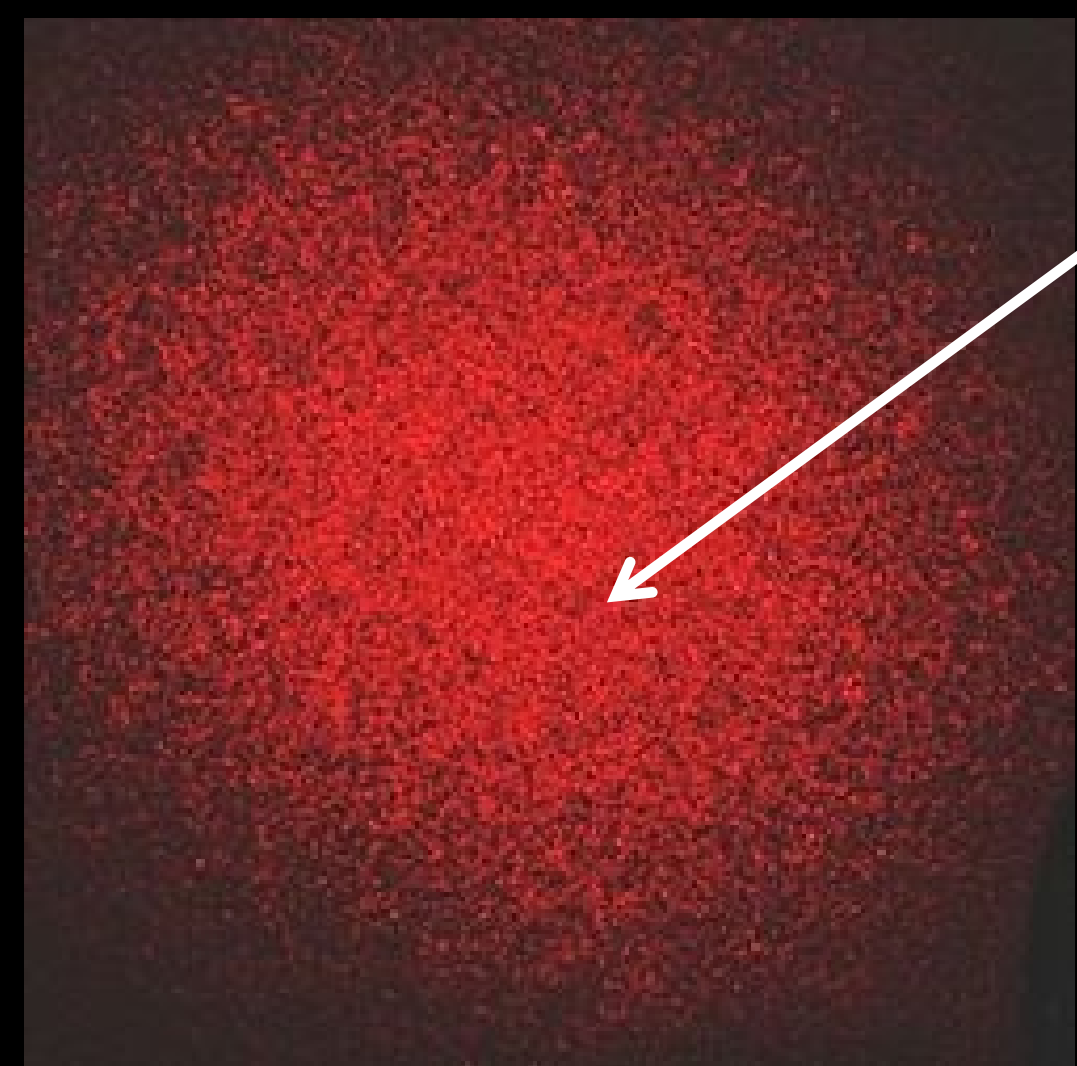
speckle: noise-like pattern
what real laser images look like



what standard Monte Carlo renderings look like

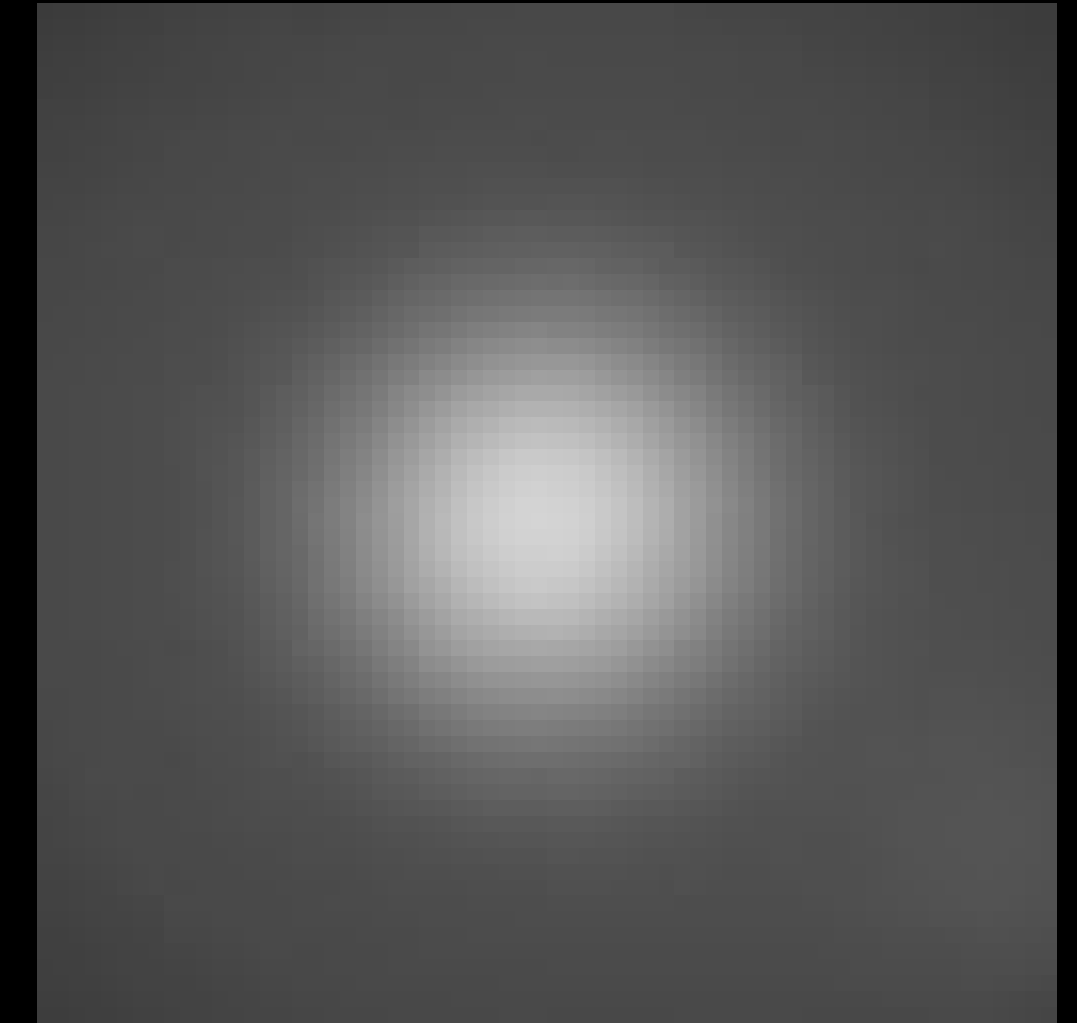
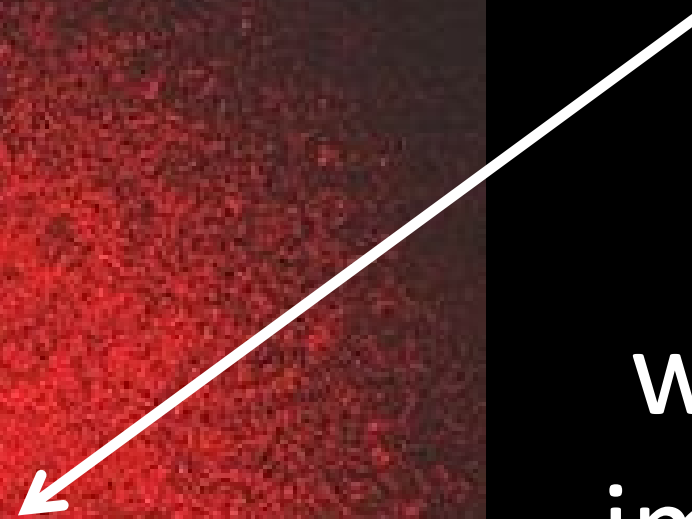


Rendering wave-optics effects



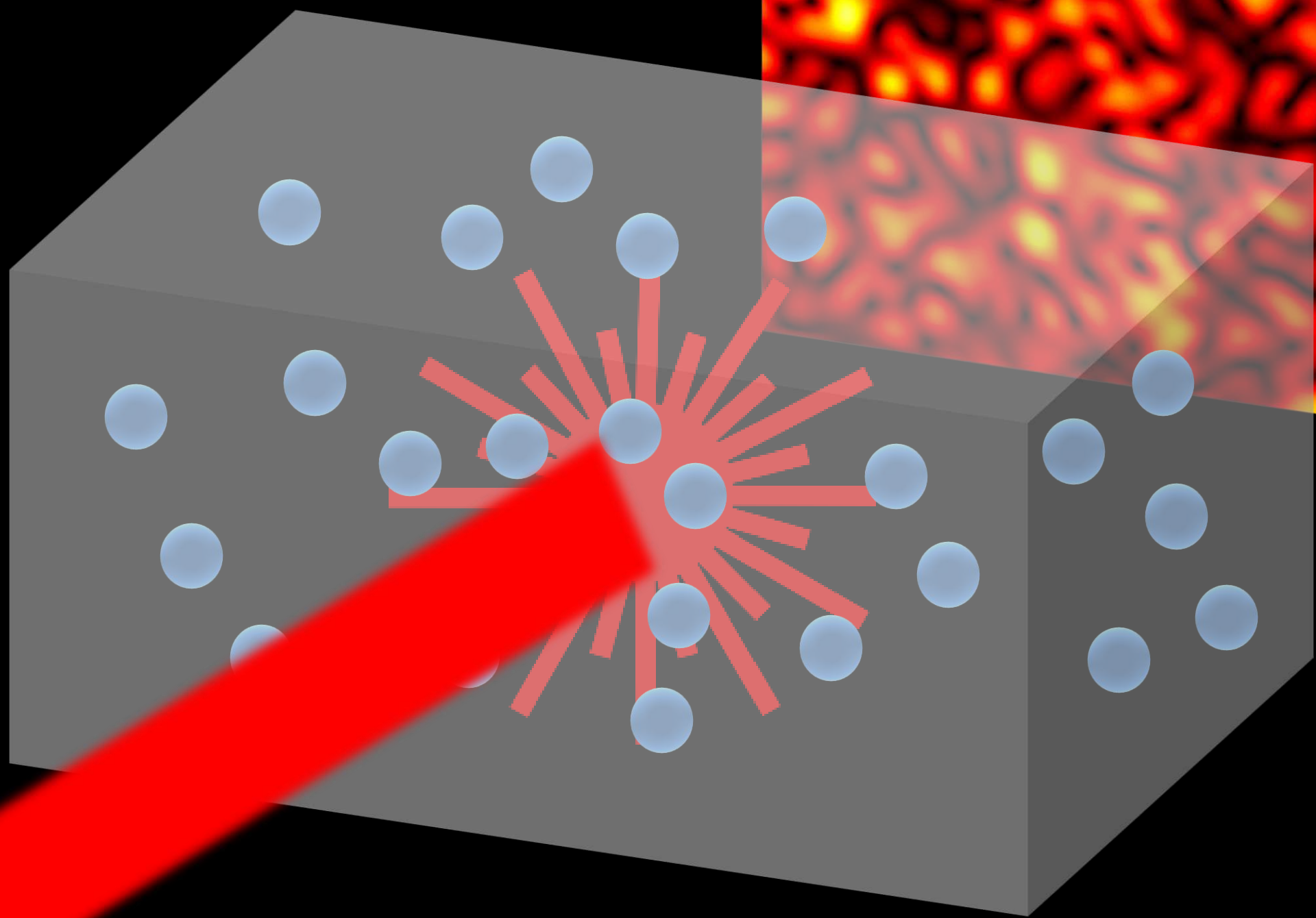
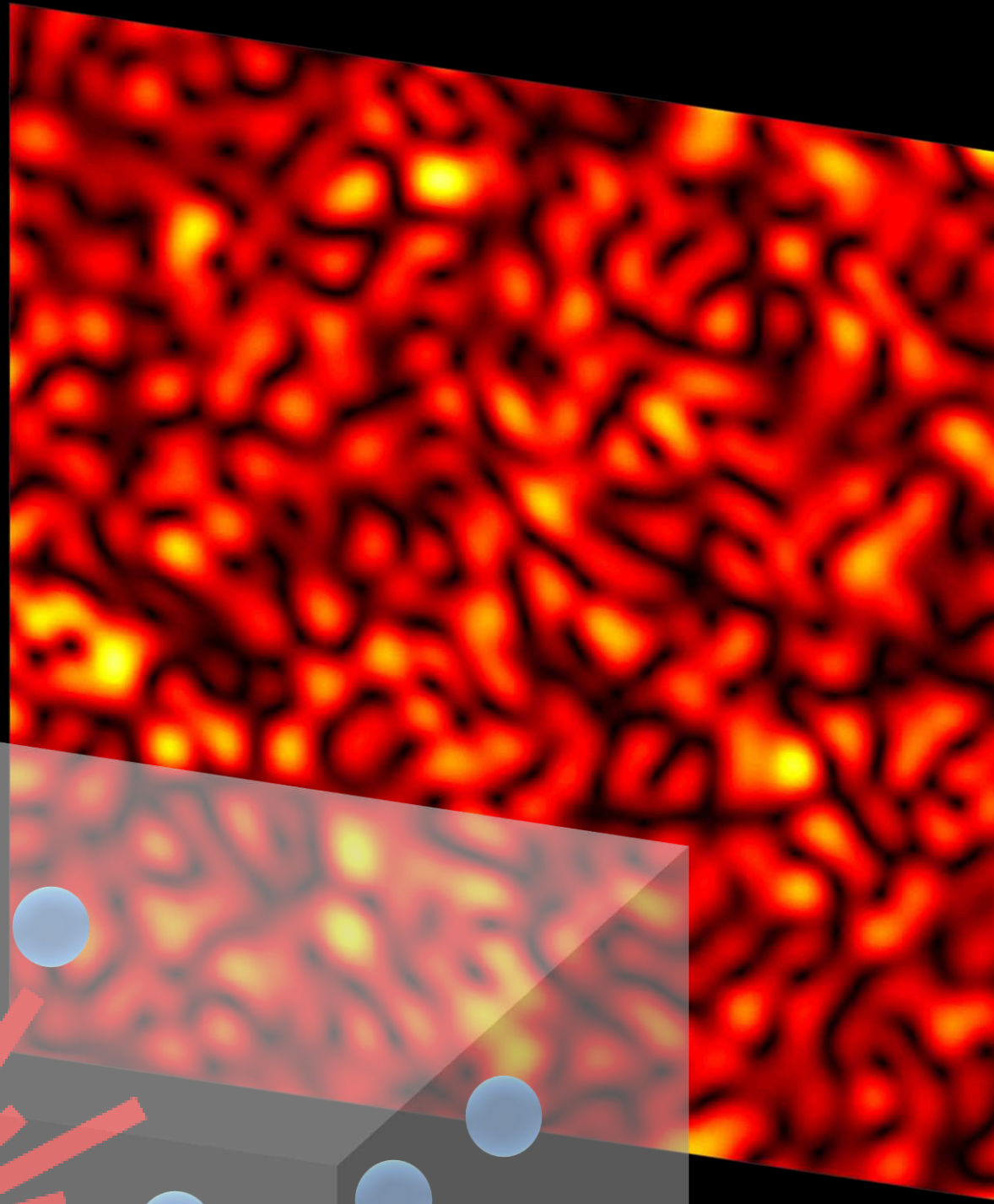
speckle: noise-like pattern

what real laser images look like



what standard Monte Carlo renderings look like

projected speckle image



laser beam

scattering volume



Applications of memory effect

SCIENTIFIC REPORTS

OPEN Memory-effect based deconvolution microscopy for super-resolution imaging through scattering media

Eitan Edrei & Giuliano Scarcelli
 High-resolution imaging through turbid media is a fundamental challenge of optical sciences that has attracted a lot of attention in recent years for its wide range of potential applications. Here, we demonstrate that the resolution of imaging systems looking behind a highly scattering medium can be improved below the diffraction-limit. To achieve this, we demonstrate a novel microscopy technique enabled by the optical memory effect that uses a deconvolution image processing and thus it does not require iterative focusing, scanning or phase retrieval procedures. We show that this newly established ability of direct imaging through turbid media provides fundamental and practical advantages such as three-dimensional refocusing and unambiguous object reconstruction.

Imaging performances of traditional optical systems quickly degrade with increasing scattering^{1,2}, so that in the diffusive regime only low resolution optical images can be obtained^{3,4}. However, recently, novel strategies such as phase conjugation, scattering-matrix inversion, ultrasonic encoding or the so-called "memory effect" have revolutionized this field showing high resolution imaging and focusing behind turbid media up to several scattering lengths^{5–19}. In many of these studies, the ability to image through turbid media is facilitated by additional information gathered on the scattering medium. For example, placing a "guide-star" in the scattered field has been used to correct for the aberrations induced by the medium^{6,7}, or the scattered field has been recorded and reversed to focus or scan the light back onto the object plane^{8–10}. Here, we use the fundamental principles of the memory effect to enable direct wide-field imaging through turbid media using deconvolution algorithms and a single-shot of the scattered intensity pattern. This enabled us to achieve, through turbid media, such as improved resolution observed when light propagates through as the memory-effect angular spectrum²⁰, microscopy²¹ as well as for which the imperfect image function with the point spread function or estimating the PSF of the

OPEN

Single-shot

Xiaohan Li, Andrew Steyer
 Imaging through opaque scattering media and astronomical imaging to media produces scattered light is contained in the signal and studies have shown that statistical effects, allow for diffraction-limited access to the source or scatterer, be static during the measurement, imaging with coded-aperture-based the first time single-shot video of important implications for a wide

Conventional optical imaging techniques assume that one can measure material intervenes, most of the light is lost, rendering isomorphic imaging impossible through opaque materials by focusing on the scatterer^{1–4}, conducting correlations in the scattered light^{5–11}. Of these, allows for imaging through high to the scatterer, object, or illumination. Several recent approaches have attempted stationary. For example, imposing a temporal point spread function (PSF)¹² enables fast show that proper filtering in correlation space going simple (e.g. linear translational) motion image a static object through a dynamic scatterer as well as significant signal at objects. Thus, none of the previously-developed dynamic, unknown scatterer.

Here, we demonstrate a method for imaging arbitrary dynamics that may even exceed temporally-code the speckle image on the im-

LOOKING THROUGH WALLS AND AROUND CORNERS

Isaac FREUND
 Department of Physics, Bar-Ilan University, Ramat-Gan, Israel

It is shown theoretically that under appropriate conditions a visually scattering optical barrier can be made to serve as a thin lens which produces a real, paraxial image of objects lying behind the barrier. Preliminary experiments described which verify the validity of the underlying assumptions. The method to serve as various other types of optical instruments, such as optical Fourier analyzers, theodolites, etc. Thus it is now clear that multi-should no longer be considered barriers to optical propagation, but are regarded as potential high-precision optical instruments.

1. Introduction

With the advent of radar half a century ago, detection of visually opaque barriers, such as dense cloud cover, became randomness in size and position of water droplets which may lead to substantial scattering of the coherent electromagnetic wave. Thus, the study of coherent wave propagation through random media became an important British Admiralty. Some of the earliest theoretical studies were carried out by Cyril Domb while seconded to the Admiralty. He was carried out by Cyril Domb while seconded to the Admiralty. He was carried out by Cyril Domb while seconded to the Admiralty.

As always, the problems attacked early on by Domb were to form later on the basis of his very first published fields of study down to this day, and, indeed, over the last have been an enormous upsurge of interest in the propagation waves in highly random media [2–5]. Here, we consider a rich reservoir of new knowledge may be applied to the imaging through highly random, multiply scattering media as a "wall", with the understanding that scattering rather

ARTICLES

PUBLISHED ONLINE: 29 JUNE 2015 | DOI: 10.1038/NPHYS3373

Translation correlations in anisotropically scattering media

Benjamin Judkewitz^{1,2*}, Roarke Horstmeyer^{2†}, Ivo M. Vellekoop³, Ioannis N. Papadimitrakis⁴ and Changhui Yang²

Controlling light propagation across scattering media by wavefront shaping holds great promise for communications and imaging applications. But, finding the right shape for the wavefront is a challenge because the input and output scattered wavefronts (that is, the transmission matrix) is not known. Correlations in the scattered wavefronts, however, have been exploited to address this limitation. The memory effect applies to thin scattering layers at a distance from the target, which preclude scattering media, such as fog and biological tissue. Here, we theoretically predict and experimentally demonstrate matrix correlations within thick anisotropically scattering media, with important implications for adaptive optics.

Focusing light through strongly scattering media is an important goal in optical imaging and communication. Long considered impossible, recent advances in the field of wavefront shaping^{1–7} have changed this view by demonstrating that diffuse light can be focused through inhomogeneous media—as long as the correct input wavefront is used. With direct optical access to the target plane, the correct wavefront can be obtained by iterative optimization^{8,9}, phase conjugation¹⁰, or by measuring the transmission matrix¹¹. In many imaging scenarios, however, there is no direct access to the target plane. In those cases, nonlinear¹², fluorescent¹³, kinematic¹⁴, acousto-optic^{15,16} and photo-acoustic^{17,18} guide stars can be used as reference beacons. However, these techniques provide wavefront information for only one target location at a time. Although transmission matrices can be sampled quickly with a photo-acoustic approach¹⁹, this method requires absorbing samples. As a result, many samples' transmission matrices can be sampled only sparsely. Correlations within a transmission matrix can compensate for sparse sampling and could enable high-speed imaging. One of the most widely known transmission matrix correlations is the so-called "memory effect"^{20,21}, which describes the following phenomenon: when an input wavefront reaching a diffusing sample is tilted within a certain angular range, the output wavefront is equally tilted, resulting in the translation of the far-field speckle pattern at a distance behind the sample (see Fig. 1).

The translation distance within which this effect holds (that is, the range of angles for which the correlation holds) depends on the scattering medium and the geometry of the setup. When assuming highly randomizing multiple measurement of input and output wavefronts, we often modelled as a random medium over the

ARTICLES

PUBLISHED ONLINE: 31 AUGUST 2014 | DOI: 10.1038/NPHOTON.2014.189

Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations

Ori Katz^{1,2*}, Pierre Heidmann¹, Mathias Fink¹ and Sylvain Gigan^{1,2}

Optical imaging through and inside complex samples is a difficult challenge with important applications in many fields. The fundamental problem is that inhomogeneous samples such as biological tissue randomly scatter and diffuse light, preventing the formation of diffraction-limited images. Despite many recent advances, no current method can perform non-invasive imaging in real-time using diffused light. Here, we show that, owing to the 'memory-effect' for speckle correlations, a single high-resolution image of the scattered light, captured with a standard camera, encodes sufficient information to image through visually opaque layers and around corners using spatially incoherent light and various samples, from white paint to dynamic biological samples. Our single-shot lensless technique is simple, does not require wavefront-shaping nor time-gated or interferometric detection, and is realized here using a camera-phone. It has the potential to enable imaging in currently inaccessible scenarios.

Diffraction-limited optical imaging is an indispensable tool in many fields of research. Unfortunately, however, the inherent inhomogeneity of complex samples such as biological tissues induces light scattering, which diffuses any optical beam into a complex speckle pattern¹, limiting the resolution and penetration depth of optical imaging techniques². Many approaches to overcome this fundamental, yet practical, problem have been put forward over the years, with pioneering experiments in holography dating back to just a few years after the invention of the laser^{3,4}. However, to date, no approach allows real-time non-invasive imaging using diffused light. Modern techniques that are based on using only unscattered, 'ballistic' light, such as optical coherence tomography and two-photon microscopy, have proven very useful, but are inherently limited to shallow depths where a measurable amount of unscattered photons is present. Adaptive optics techniques⁵ can nearly perfectly correct low-order aberrations using deformable mirrors, but require the presence of a bright point-source 'guide star' or a high initial image contrast⁶. Recent exciting advances in controlled wavefront shaping⁷ have allowed focusing and imaging through highly scattering samples^{8–26}. However, these techniques either require initial access to both sides of the scattering medium^{8–15}, the presence of a guide-star or a known object^{16–19}, or a long acquisition sequence that involves the projection of a large number of optical patterns^{20–26}. A recent breakthrough approach reported by Bertolotti *et al.* has removed the requirement for a guide-star or a known object by exploiting the inherent angular correlation of the scattered light that diffuses through a scattering medium. Specifically, derived from concepts used in 'stellar speckle interferometry'^{29–31} and the angular 'memory-effect' for speckle correlations^{32–34} exploited in Bertolotti's technique²⁷, we show that the autocorrelation of the scattered light pattern (Fig. 1c) is essentially identical to the autocorrelation of the object's image itself (Fig. 1g). We then reconstruct the object's image from its autocorrelation using an iterative Fienup-type algorithm³⁵. As proofs of concept, we experimentally demonstrate our single-shot non-invasive technique through a variety of highly scattering samples, from optical diffusers to dynamically varying biological tissues. In addition, we demonstrate imaging 'around corners' by recording the diffuse light back-scattered off white-painted 'walls'.

Principle

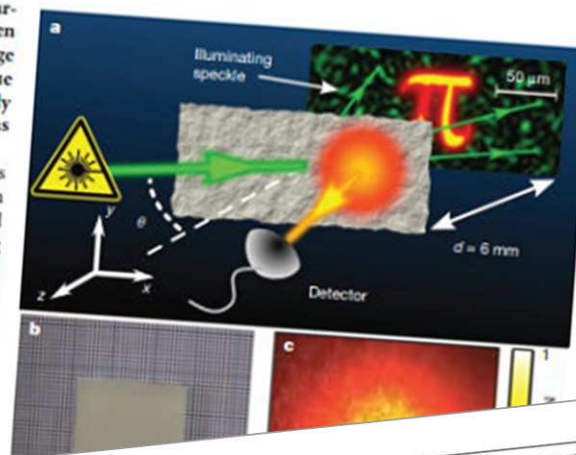
A schematic of the experiment for imaging through a scattering medium, as well as a numerical example, are presented in Fig. 1a–d. An object is hidden at a distance u behind a highly scattering medium of thickness L . The object is illuminated by a spatially incoherent, narrowband source, and a high-resolution camera that is placed at a distance v on the other side of the medium records the pattern of the scattered light that has diffused through the scattering medium. Although the raw recorded camera image is a low-contrast, random and seemingly information-less image (Fig. 1b), its autocorrelation (Fig. 1c) is essentially identical to the object's autocorrelation (Fig. 1g) if it had been imaged by an aberration-free diffraction-limited camera. The scattered light that diffuses through a scattering medium allows light to pass in all directions, completely scrambling all the spatial information. However, it has proved very successful in imaging to separate the small amount of light owing to random scattering (ballistic light) using a gated technique such as time-gated imaging³⁶. In this way it is possible to obtain detailed information about the sample enclosed between two opaque media, but for stronger scattering media, this does not allow one to

LETTER

Non-invasive imaging through opaque scattering media

Ori Katz^{1,2*}, Albert G. van Putten^{1,2}, Christian Blum¹, Ad Lagendijk^{1,4}, Willem L. Vos¹ & Allard P. Mosk¹

Optical imaging techniques, such as optical coherence tomography^{1,2}, are essential diagnostic tools in many disciplines from medicine to nanotechnology. However, present techniques are limited to imaging through transparent media, as they require either a detector³ or a nonlinear lens⁴ behind the scattering layer. Here we report a method for non-invasive imaging of a fluorescent object through an opaque scattering layer. We scan the angle of incidence of the total fluorescence of the object from the detector, we obtain the image of the object from a single-shot of the scattered light. As a proof of concept, we image a fluorescent object, comparable to a typical human cell, hidden behind an opaque optical diffuser, and an image sample enclosed between two opaque media. Our non-invasive imaging through strongly scattering media does not require wavefront-shaping nor time-gated or interferometric detection, and is realized here using a camera-phone.



day, scattering of light severely impairs imaging. However, it has proved very successful in imaging to separate the small amount of light owing to random scattering (ballistic light) using a gated technique such as time-gated imaging³⁶. In this way it is possible to obtain detailed information about the sample enclosed between two opaque media, but for stronger scattering media, this does not allow one to

14; revised 20 March 2015; accepted 21 March 2015 (Doc. ID 226377); published 27 April 2015

Correlation resolution enhancement of fluorescence imaging

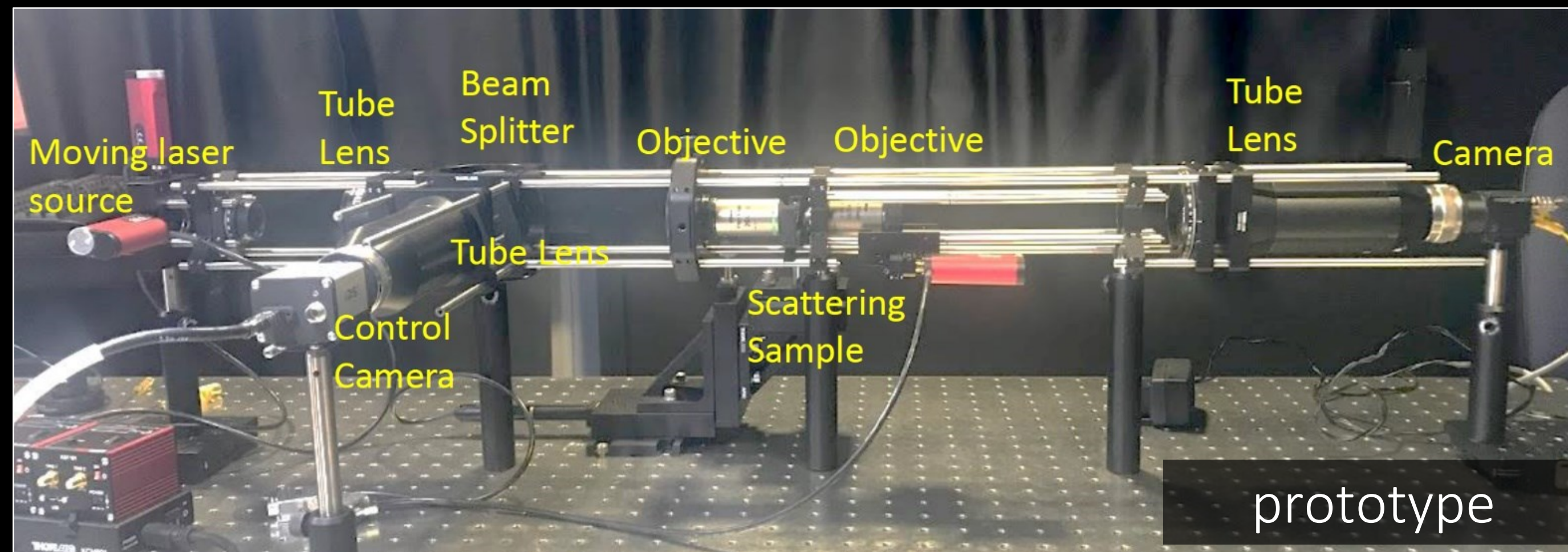
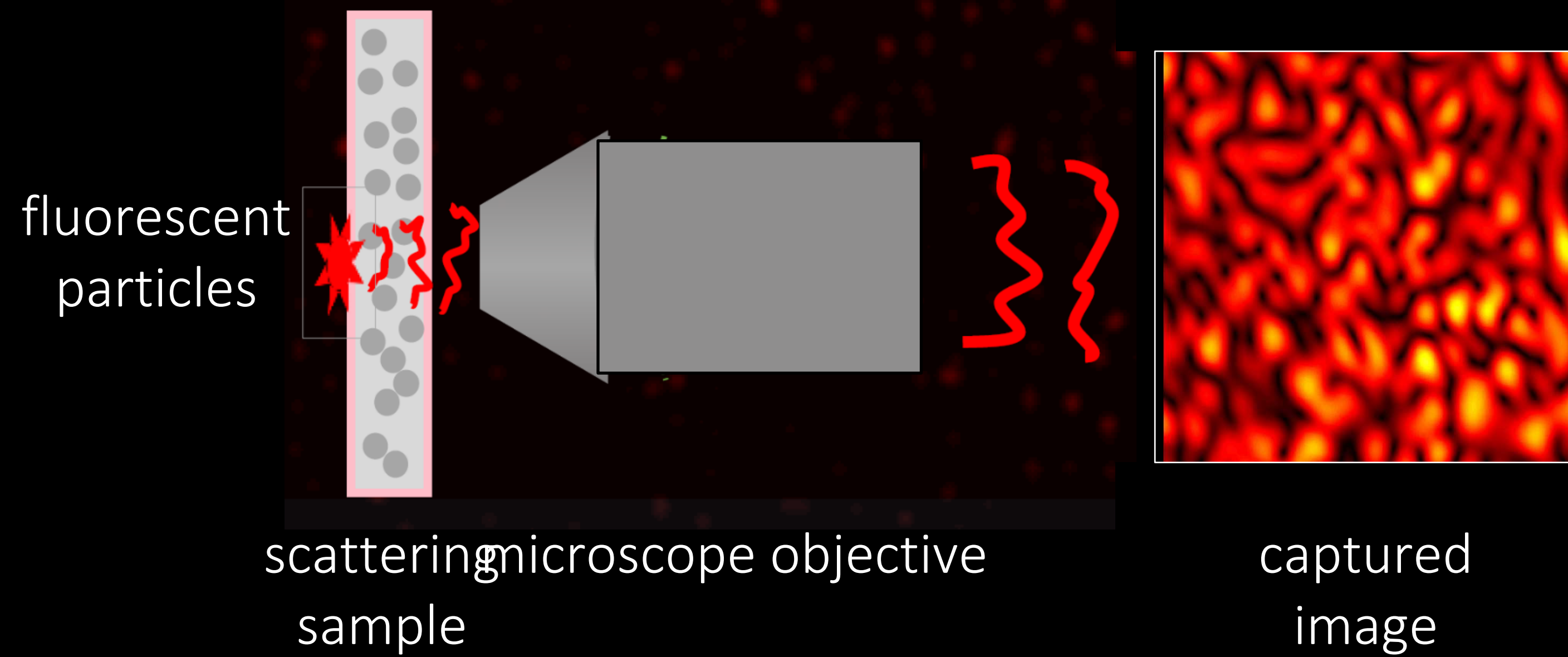
ELBERT G. VAN PUTTEN^{1,2}, JACOPO BERLOTTI^{1,3}, AD LAGENDIJK¹, AND ALLARD P. MOSK¹
¹MESA+ Institute for Nanotechnology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
²Research Laboratories, 5656 AE Eindhoven, The Netherlands
³Physics and Astronomy Department, University of Exeter, Stocker Road, Exeter EX4 4QL, UK
 *ylmaz@utwente.nl

Fluorescence imaging is essential in nanoscience and biological sciences. Due to the diffraction limit, imaging systems can only resolve structures larger than 200 nm. Here, we introduce a new fluorescence imaging method that enhances the resolution by using a high-index scattering medium as an imaging lens. We achieve a wide field of view. We develop a new image reconstruction algorithm that converges on such a scattering medium generates a speckle pattern on the interface of the substrate [26]. Coherent light illuminating a scattering medium generates a speckle pattern on the interface of the substrate [26]. Coherent light illuminating a scattering medium generates a speckle pattern on the interface of the substrate [26].

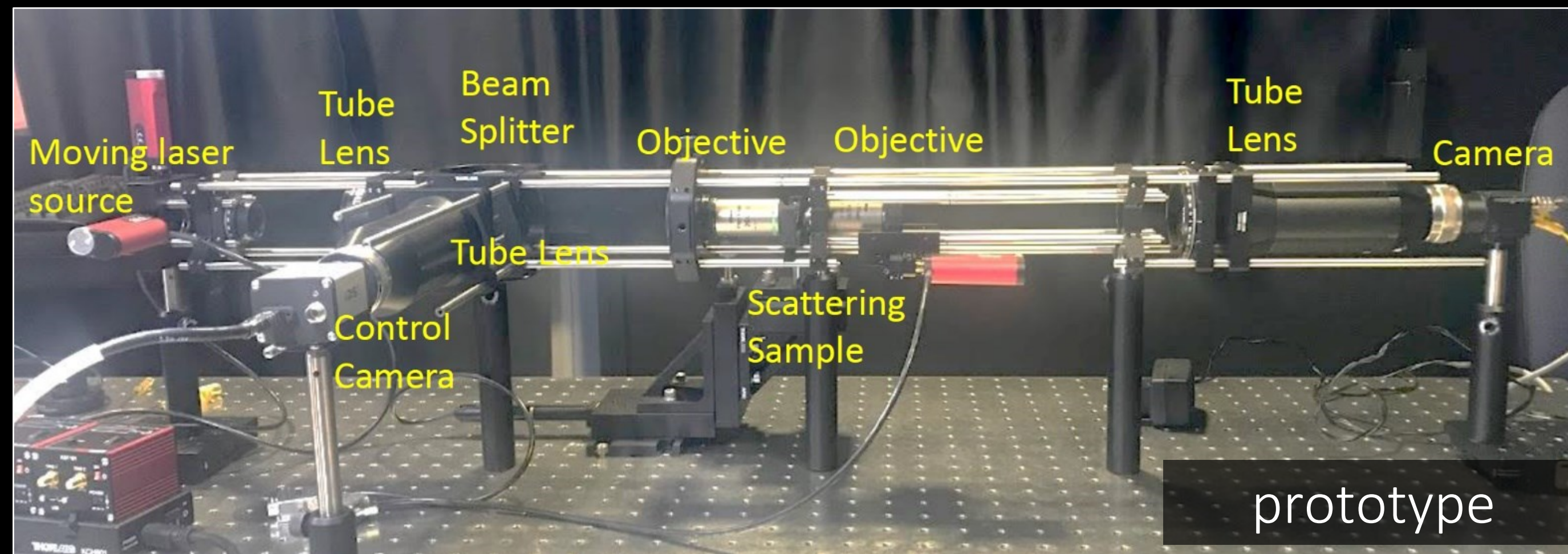
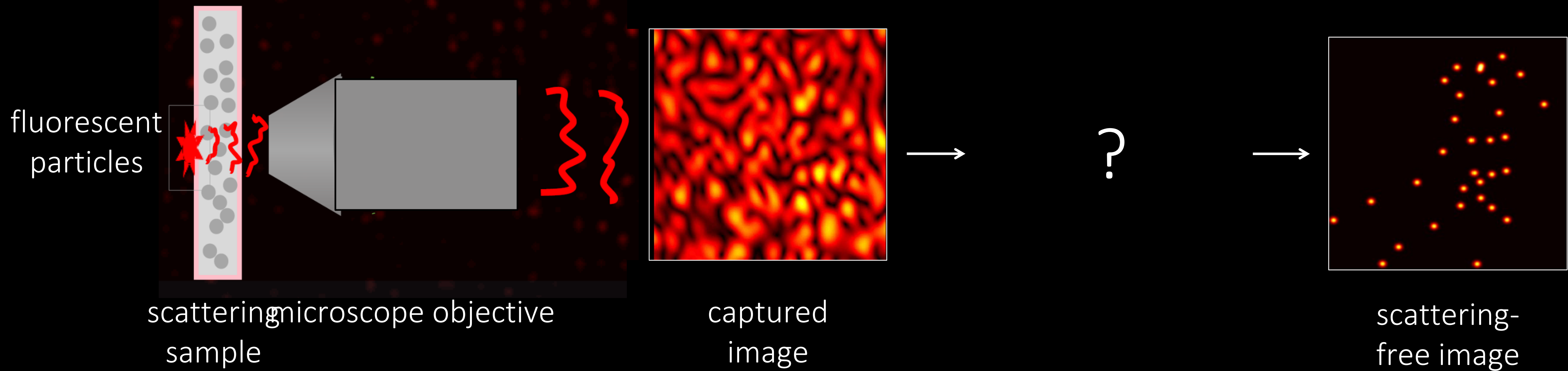
0.180) Microscopy; (290.0290) Scattering; (180.2520) Fluorescence microscopy; (110.6150) Speckle imaging; (100.5070) Phase retrieval.
 10.1364/OPTICA.2.000424

microscope produces images with a resolution that is limited by the numerical aperture (NA) of the imaging lens. However, it has proved very successful in imaging to separate the small amount of light owing to random scattering (ballistic light) using a gated technique such as time-gated imaging³⁶. In this way it is possible to obtain detailed information about the sample enclosed between two opaque media, but for stronger scattering media, this does not allow one to

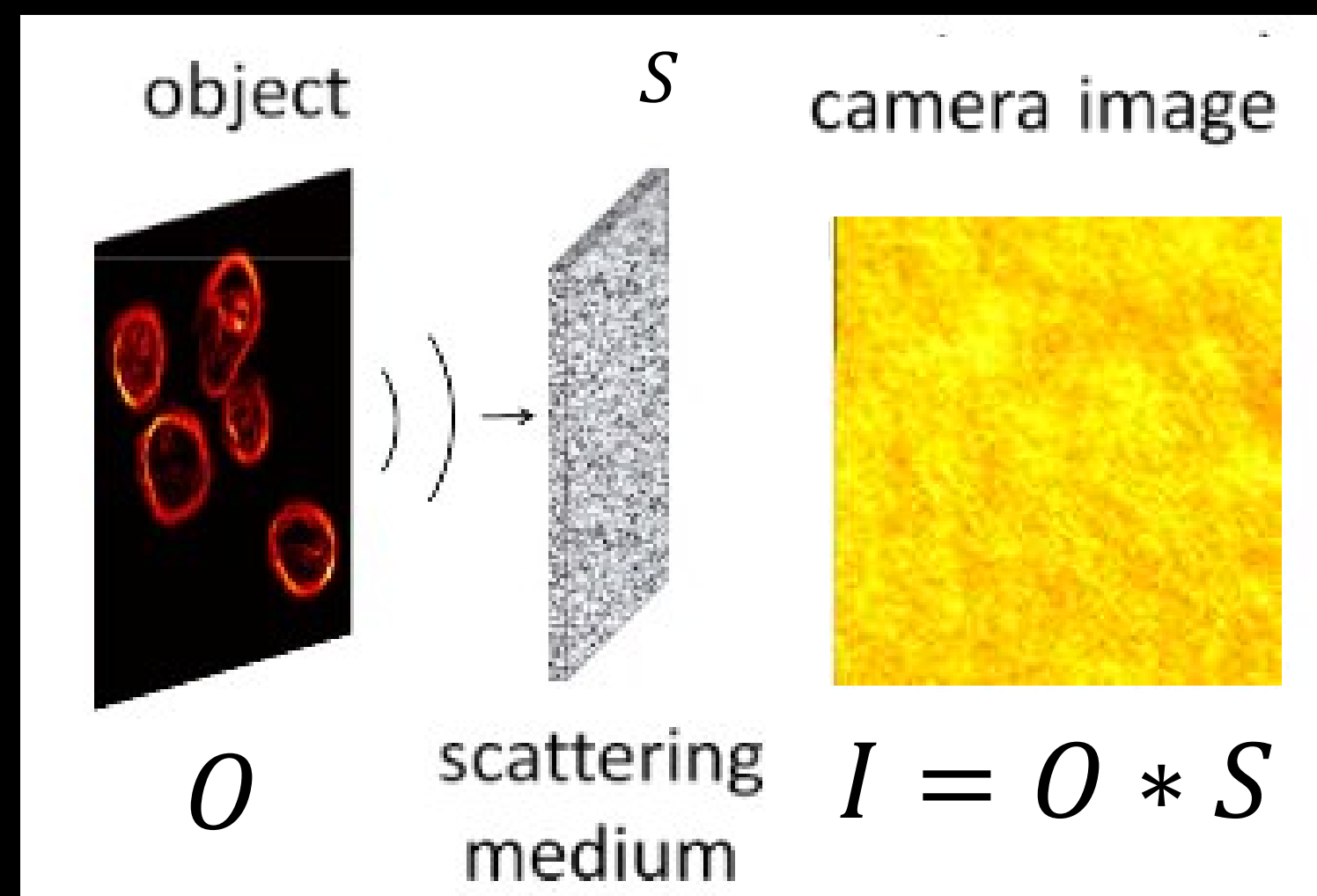
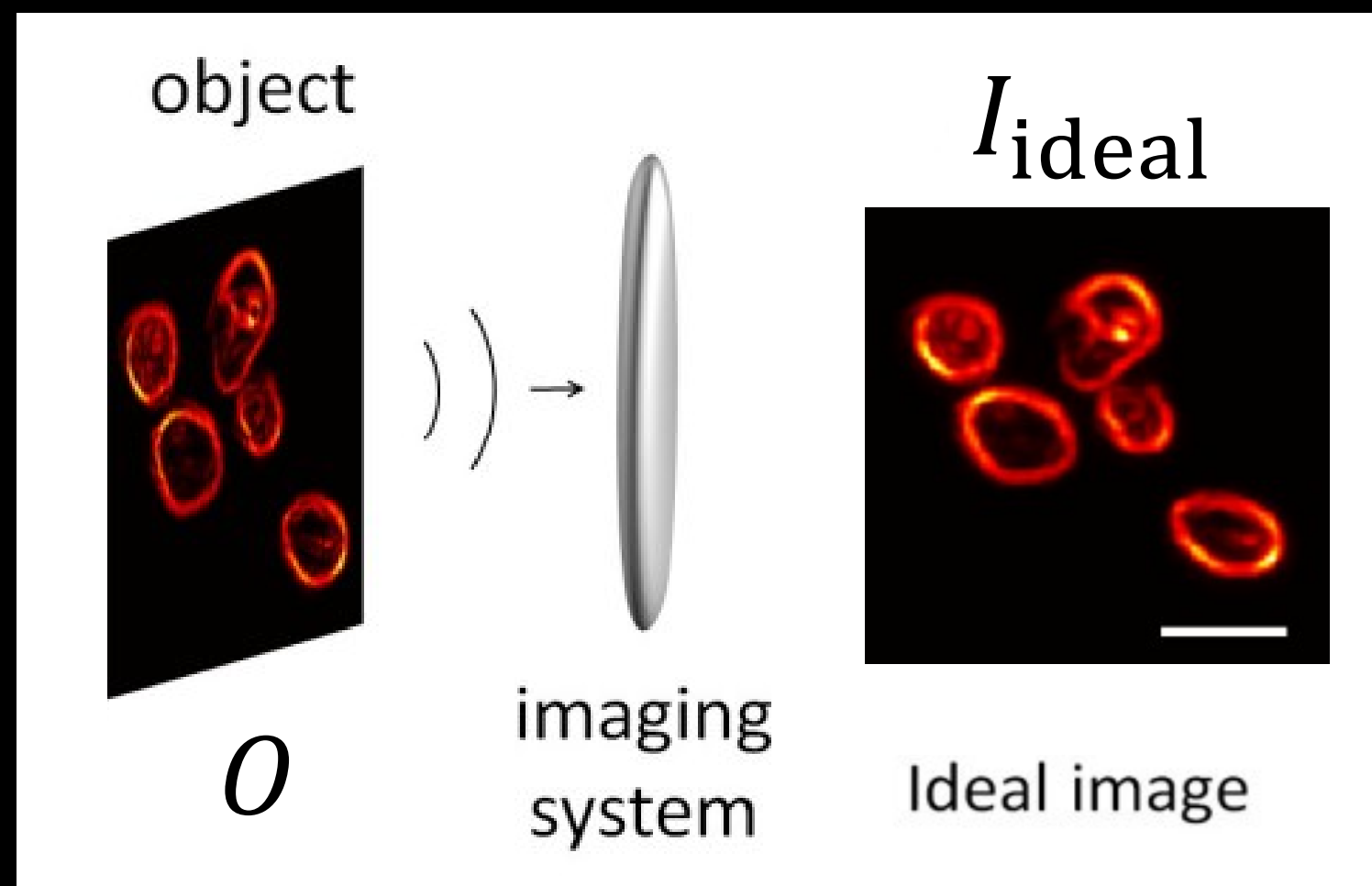
Speckle-based fluorescence microscopy



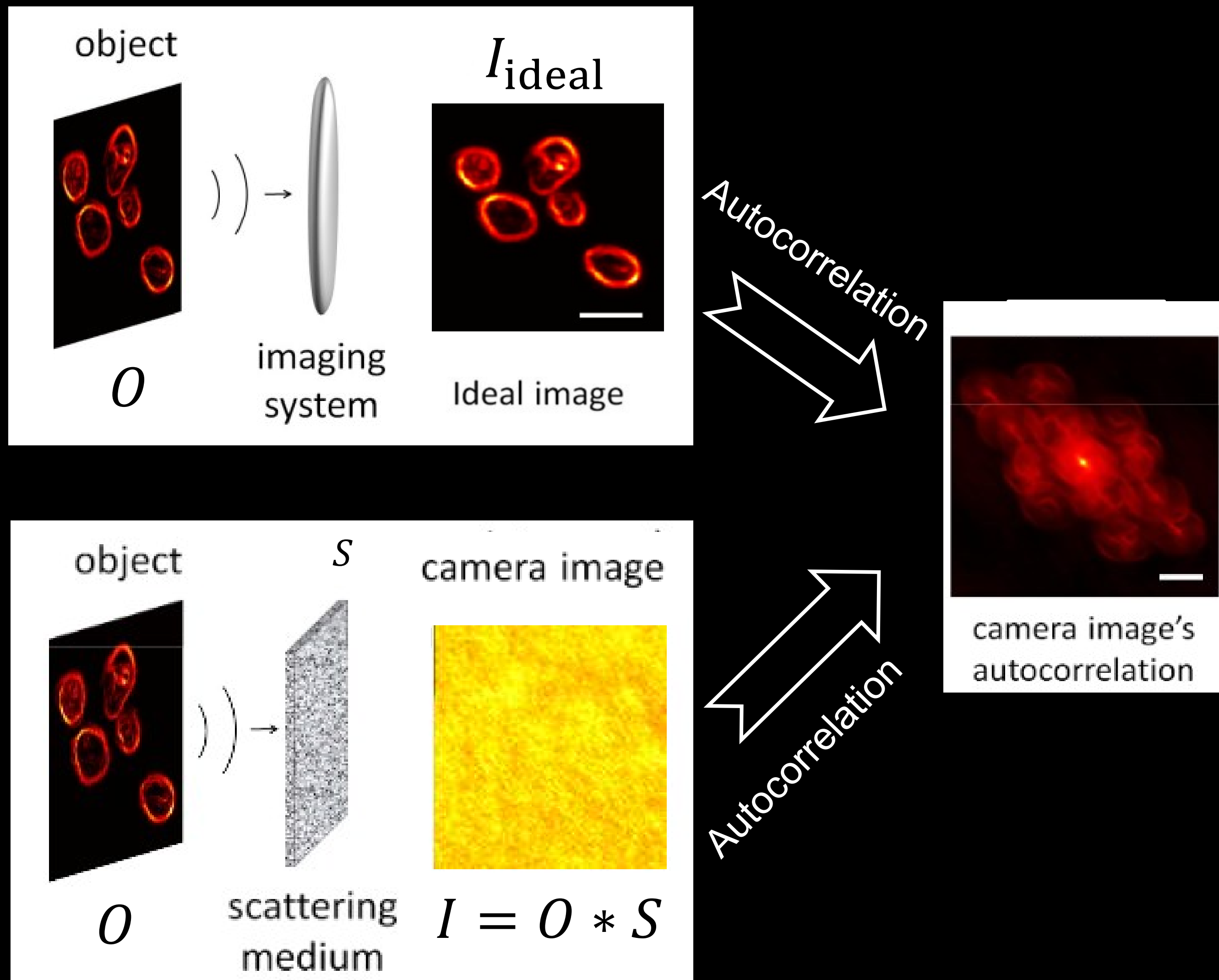
Speckle-based fluorescence microscopy



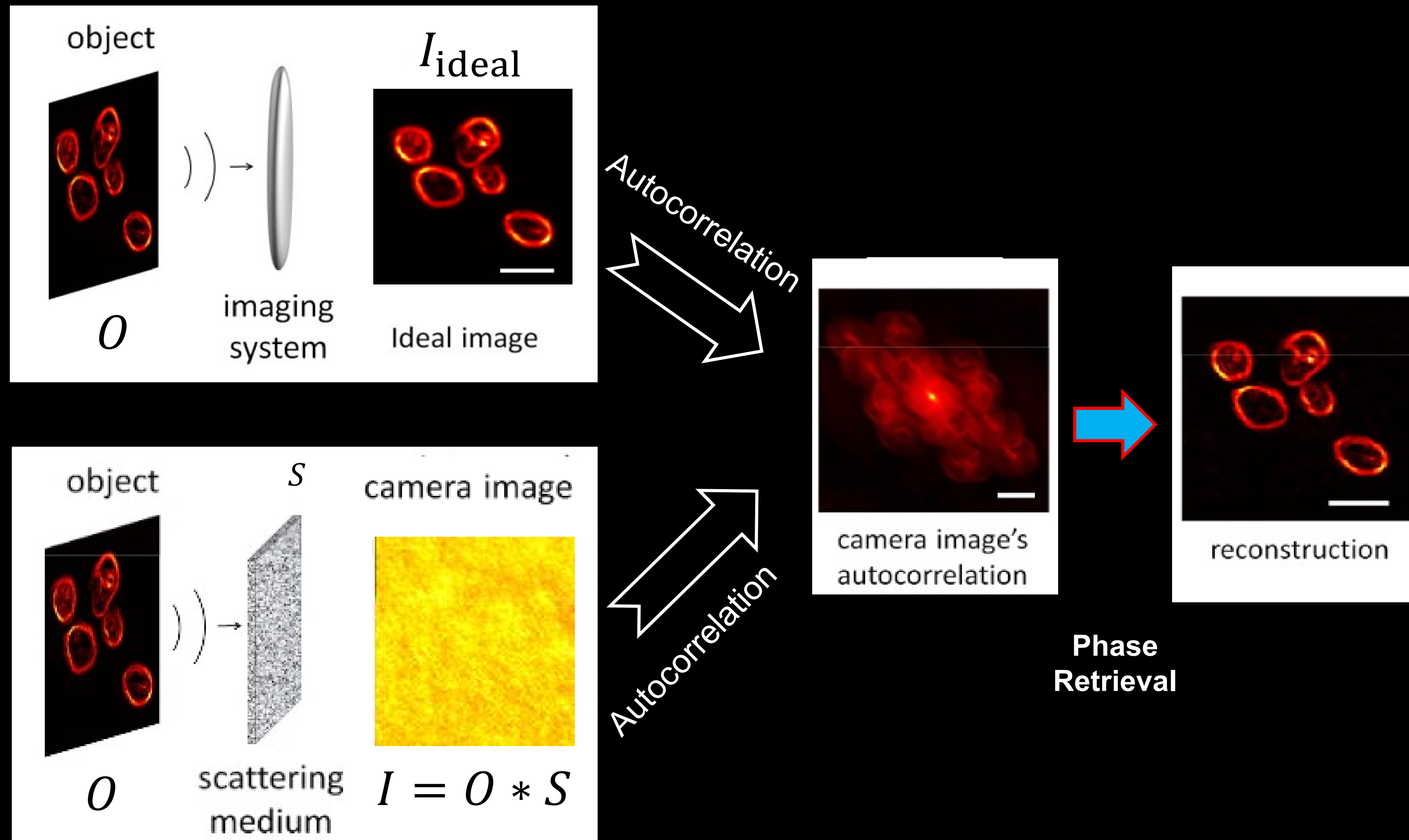
Use the memory effect to image through scattering



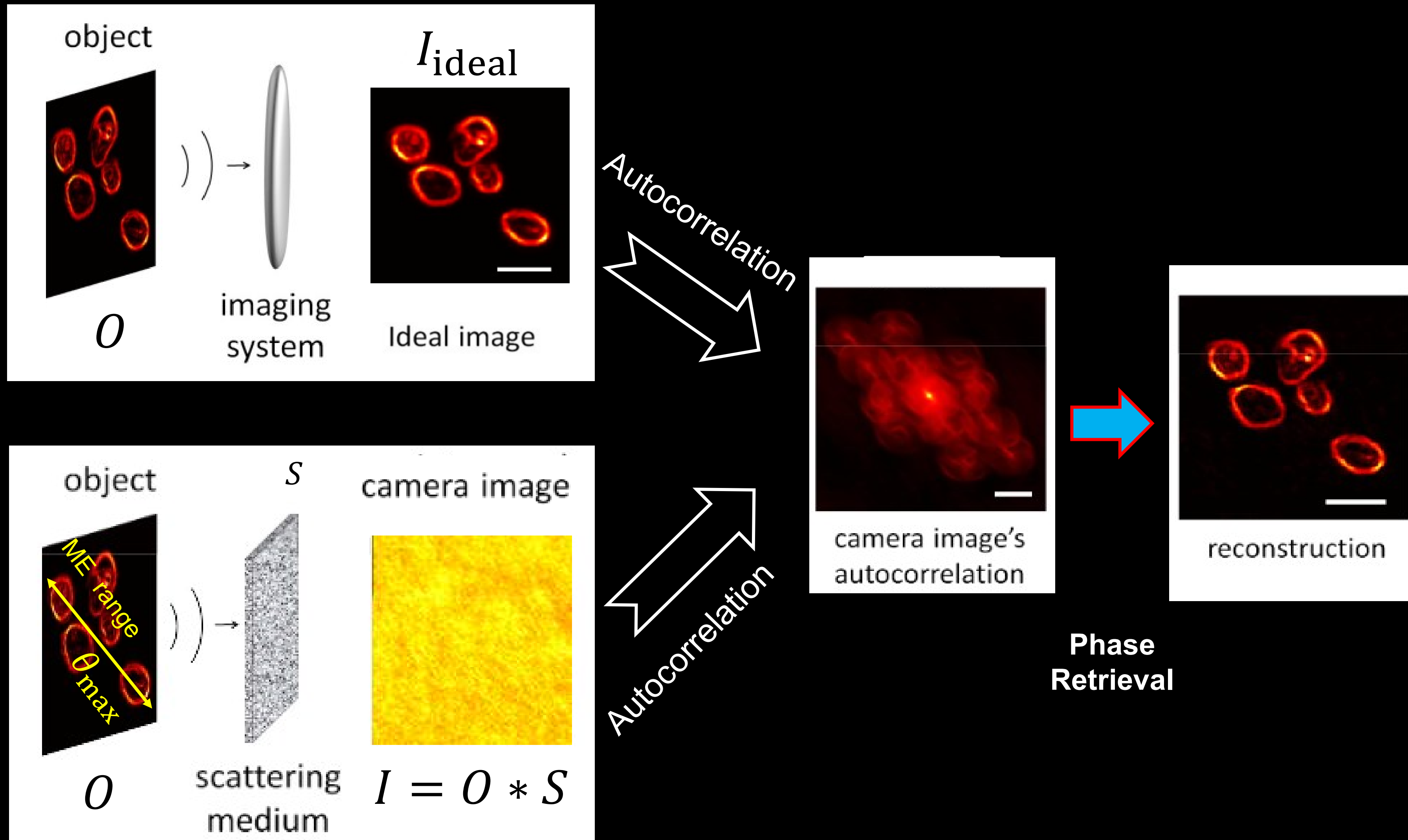
Use the memory effect to image through scattering



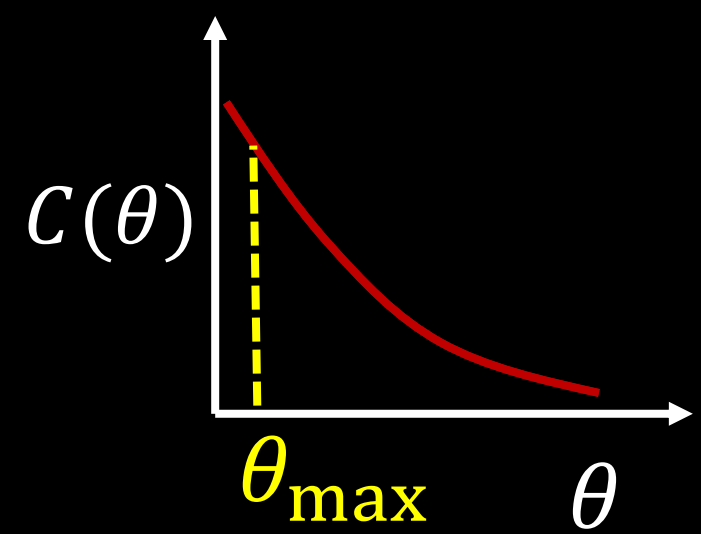
Use the memory effect to image through scattering



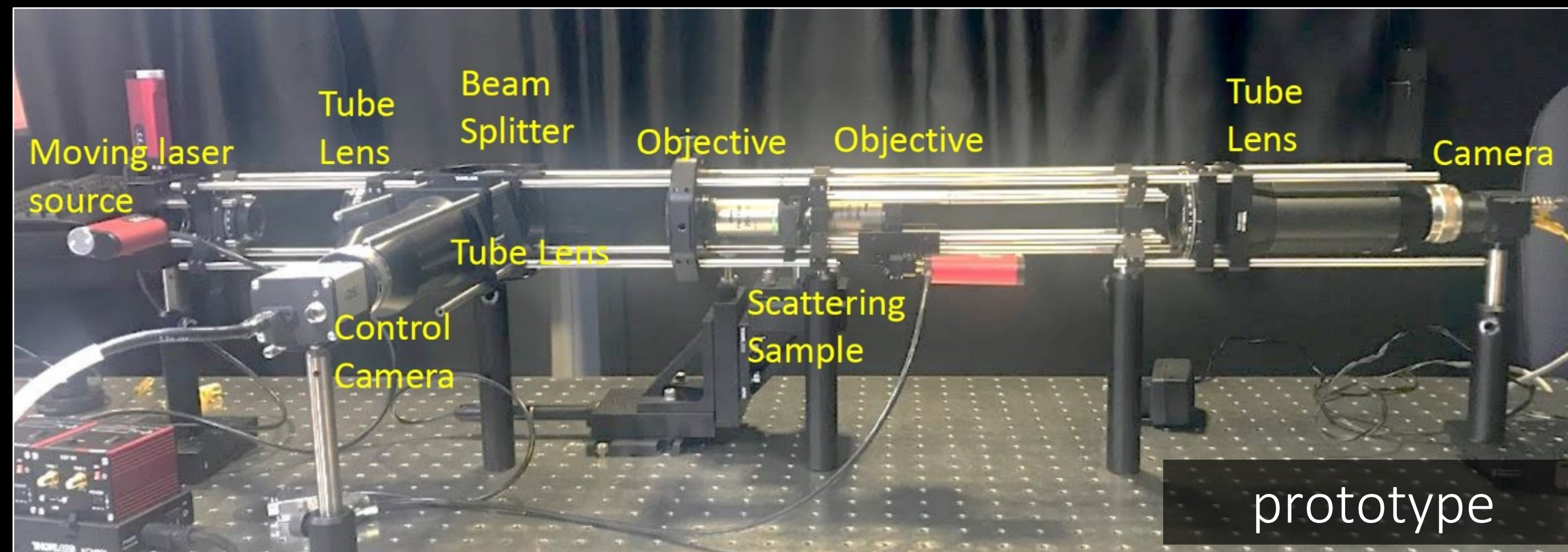
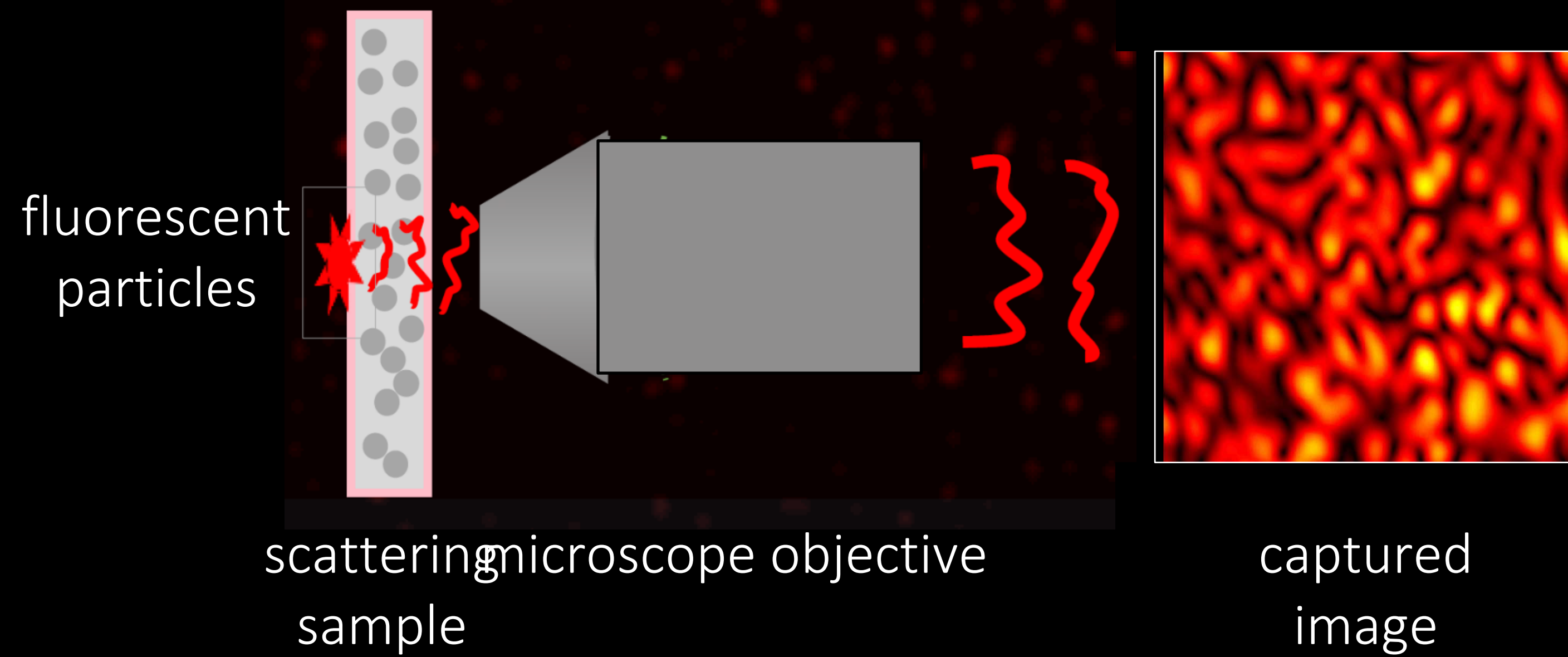
Use the memory effect to image through scattering



memory-effect range

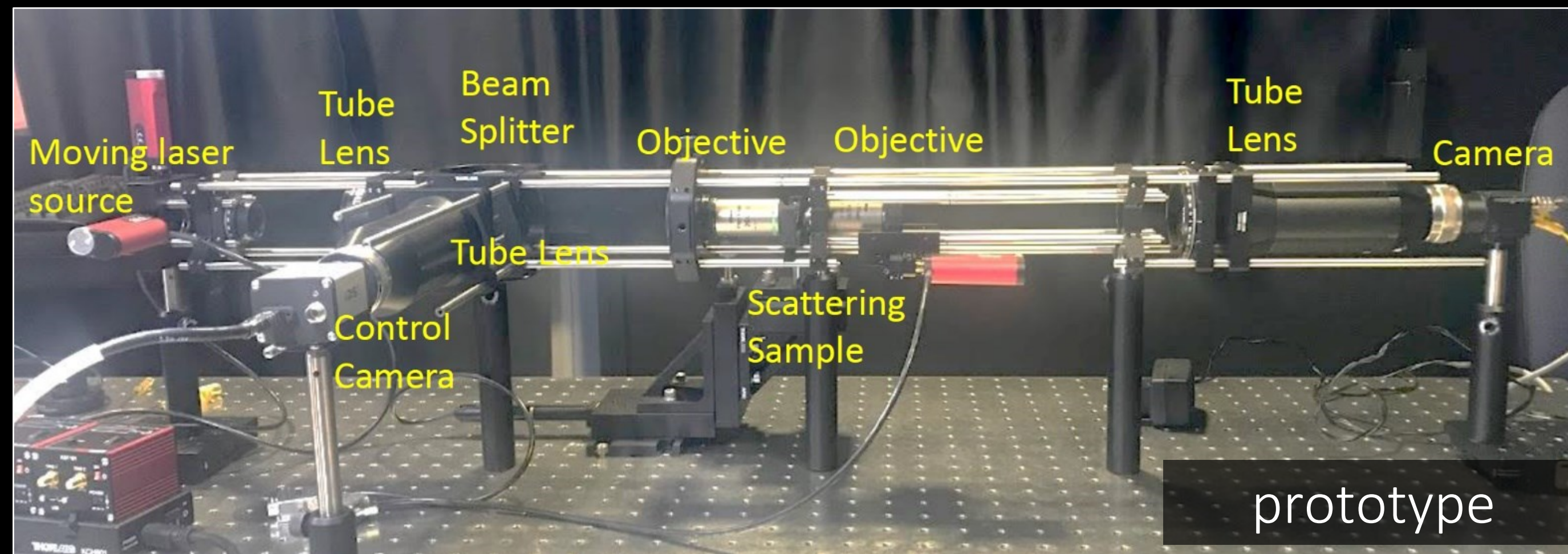
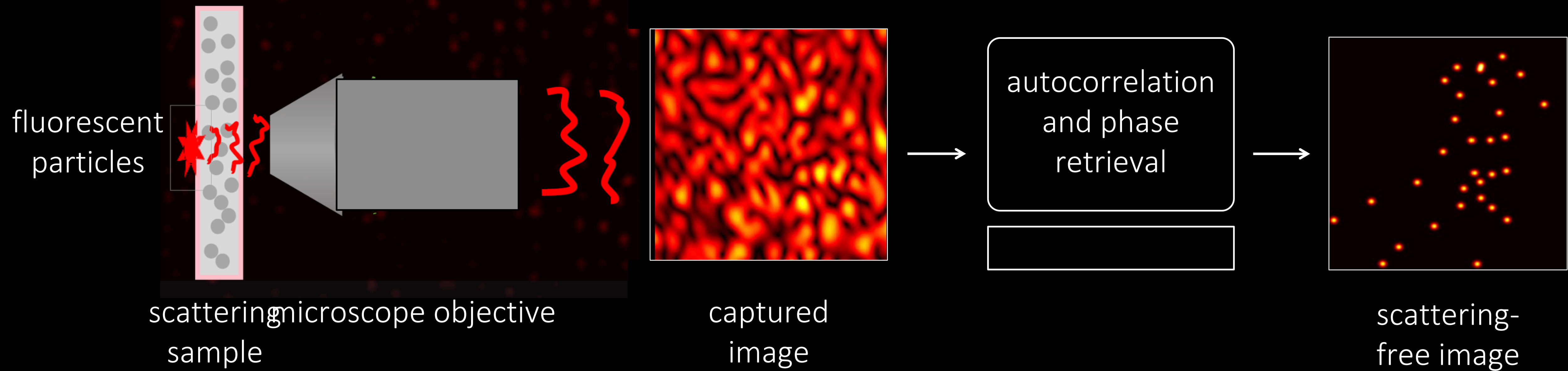


Speckle-based fluorescence microscopy



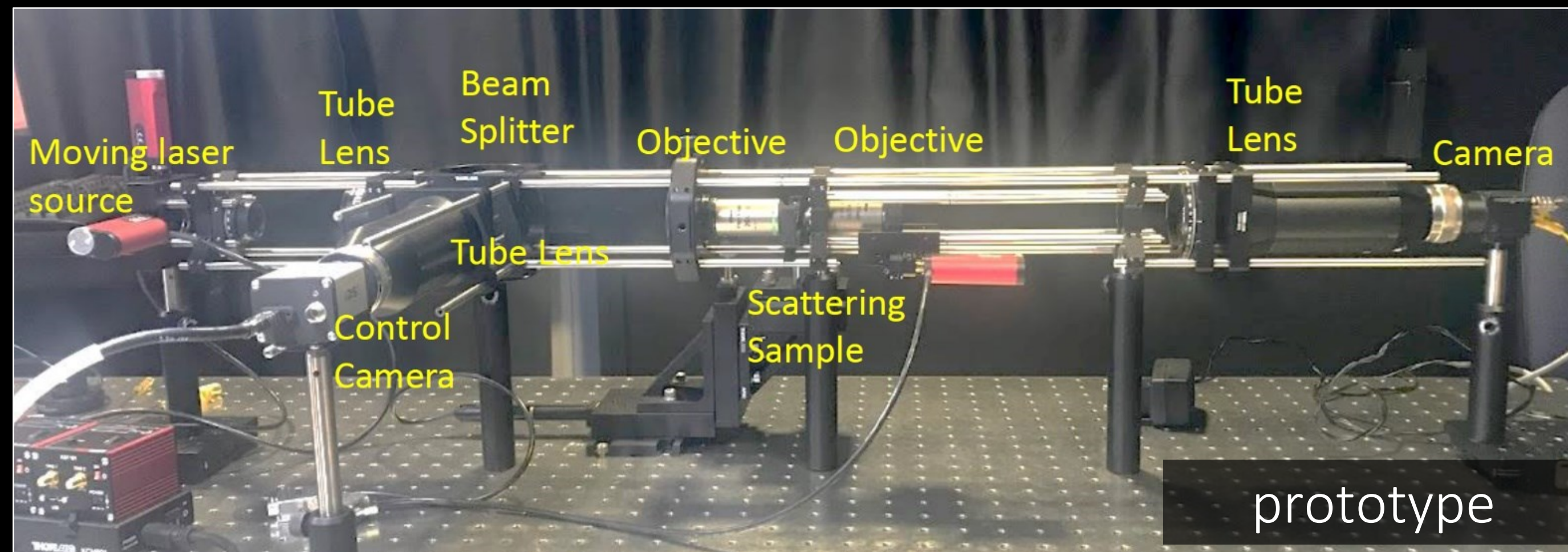
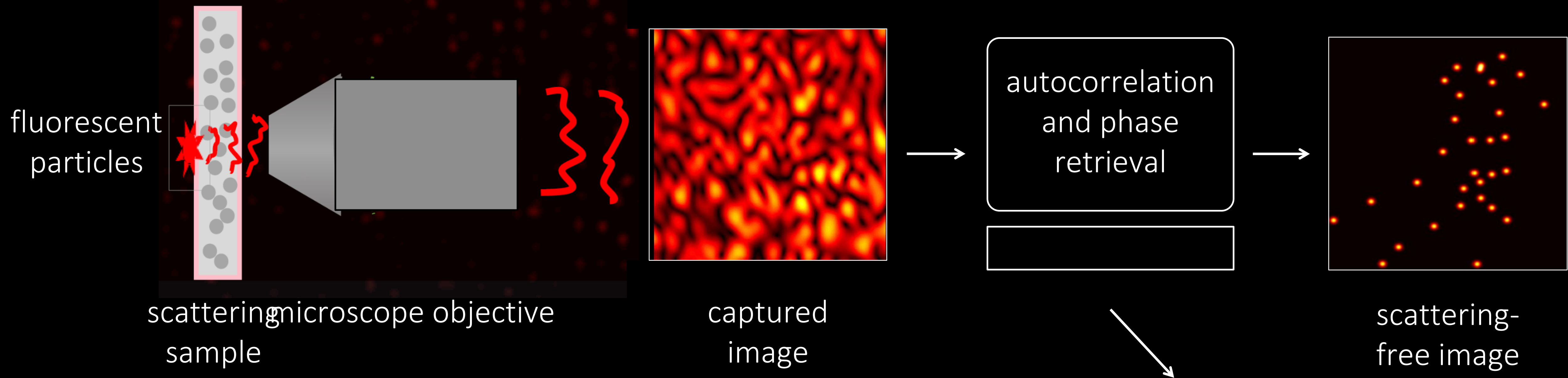
[PIs: Gkioulekas, Levin]

Speckle-based fluorescence microscopy



[PIs: Gkioulekas, Levin]

Speckle-based fluorescence microscopy

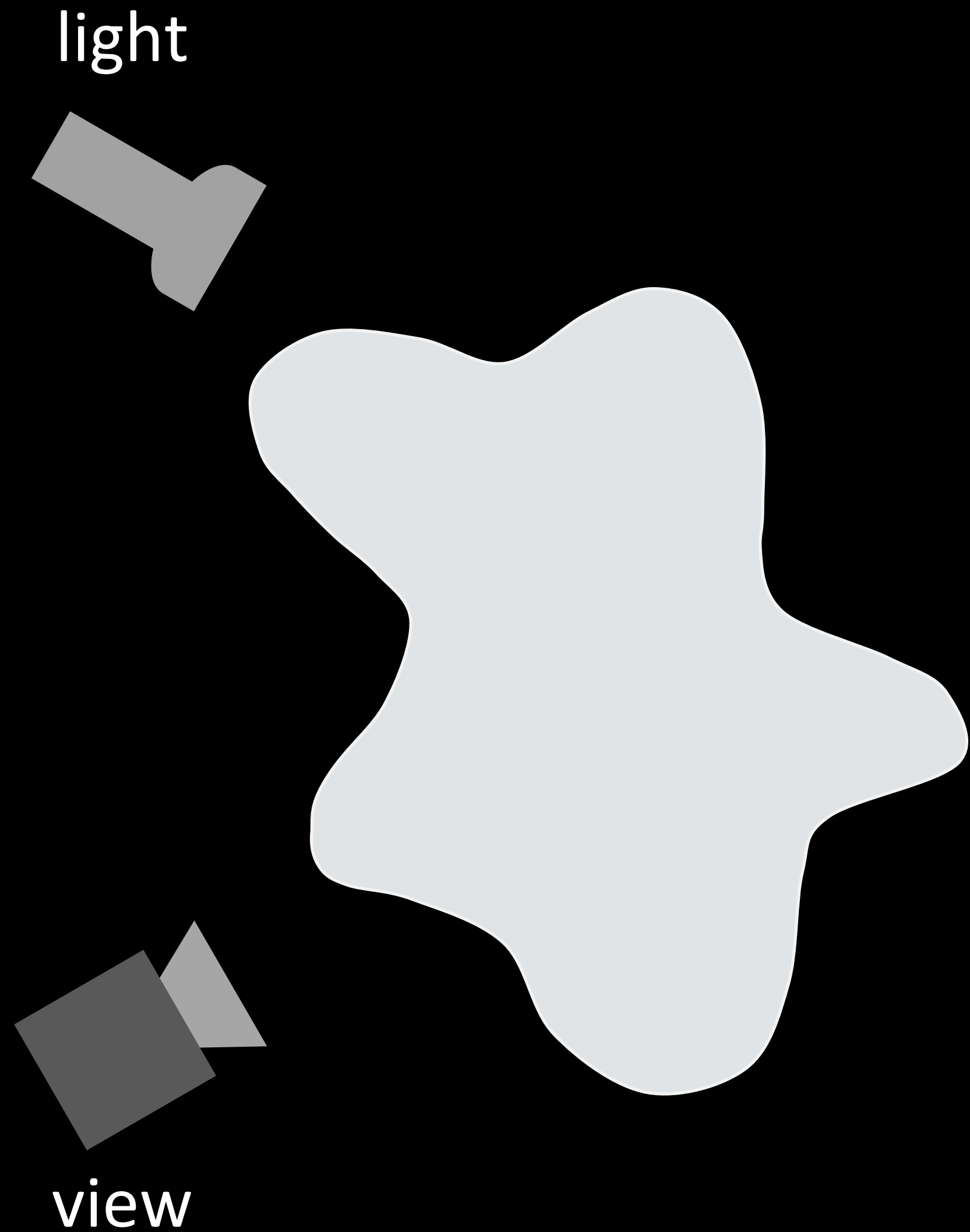


[PIs: Gkioulekas, Levin]

Performance strongly depends on:

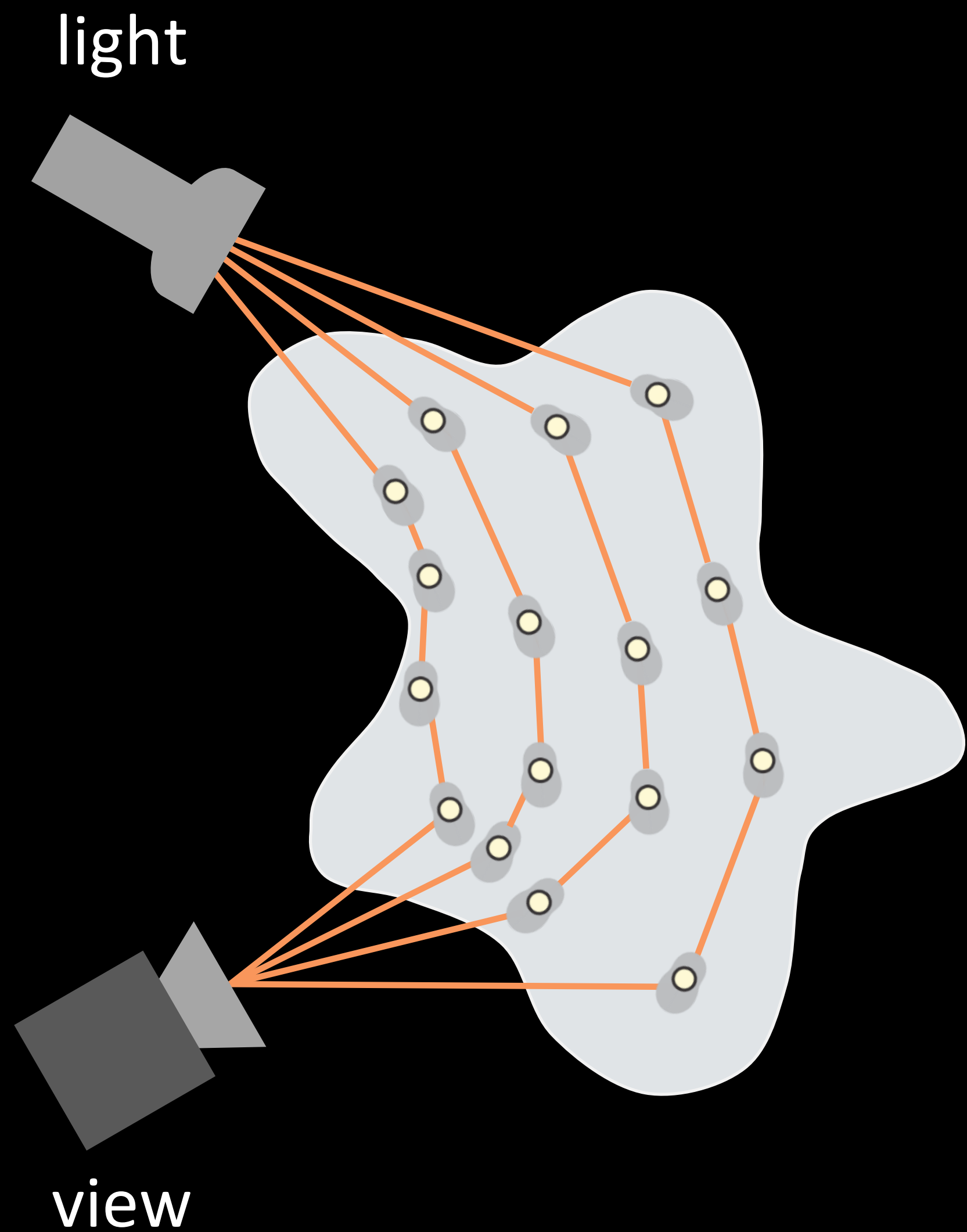
- speckle statistics
- image priors
- tissue parameters

Recap: Monte Carlo (volumetric) rendering



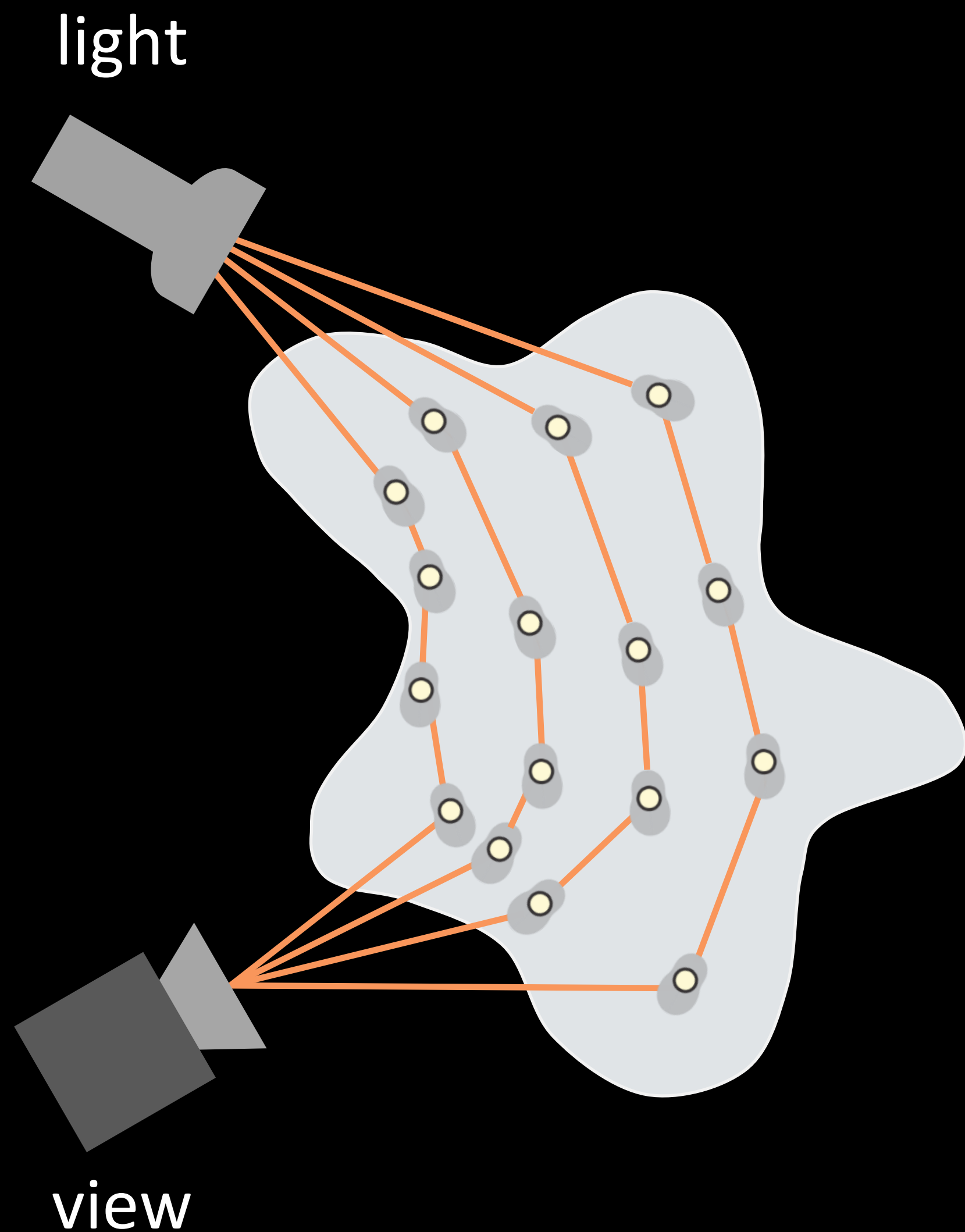
$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

Recap: Monte Carlo (volumetric) rendering



$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

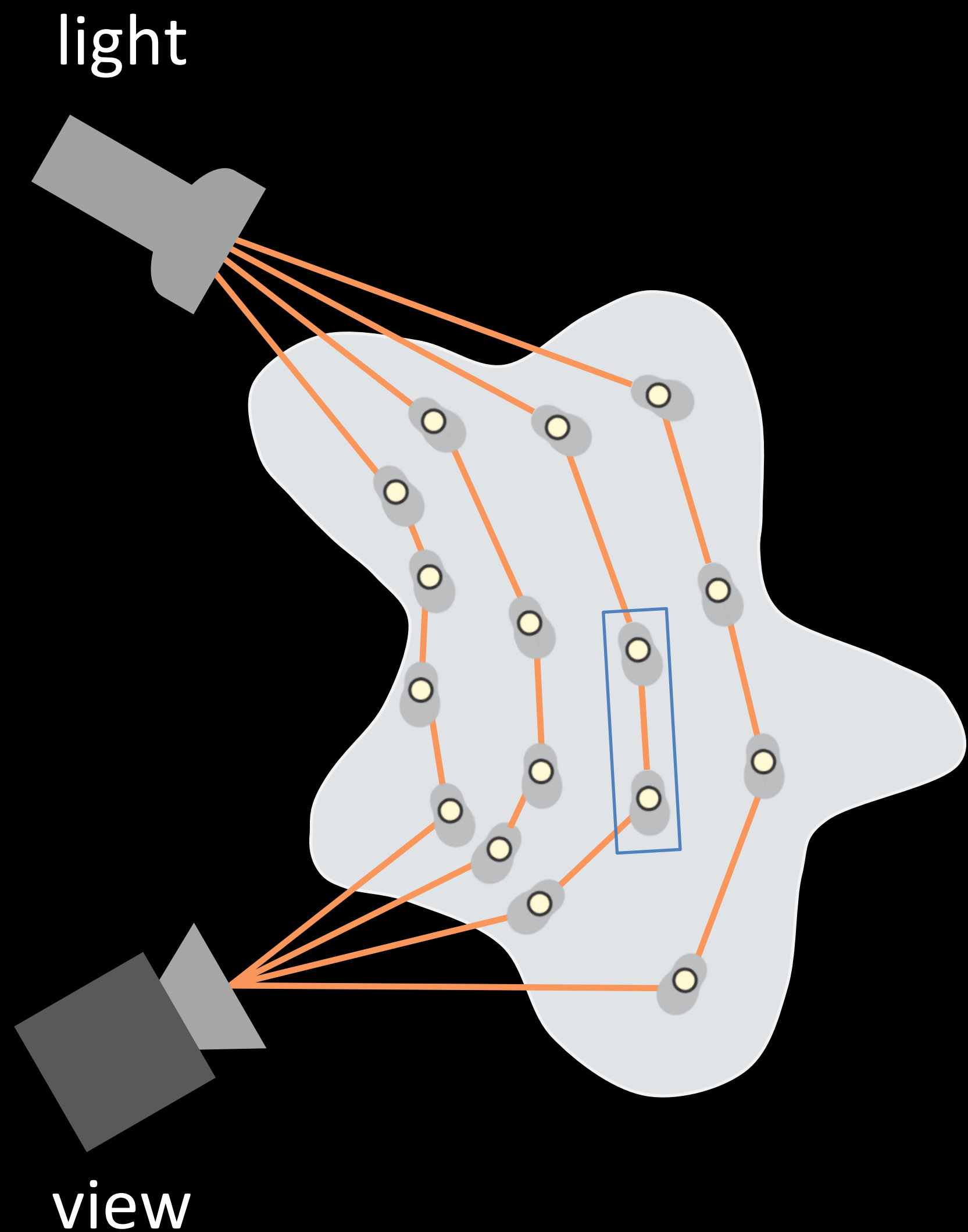
Recap: Monte Carlo (volumetric) rendering



$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

Path contribution, depends on the scattering material

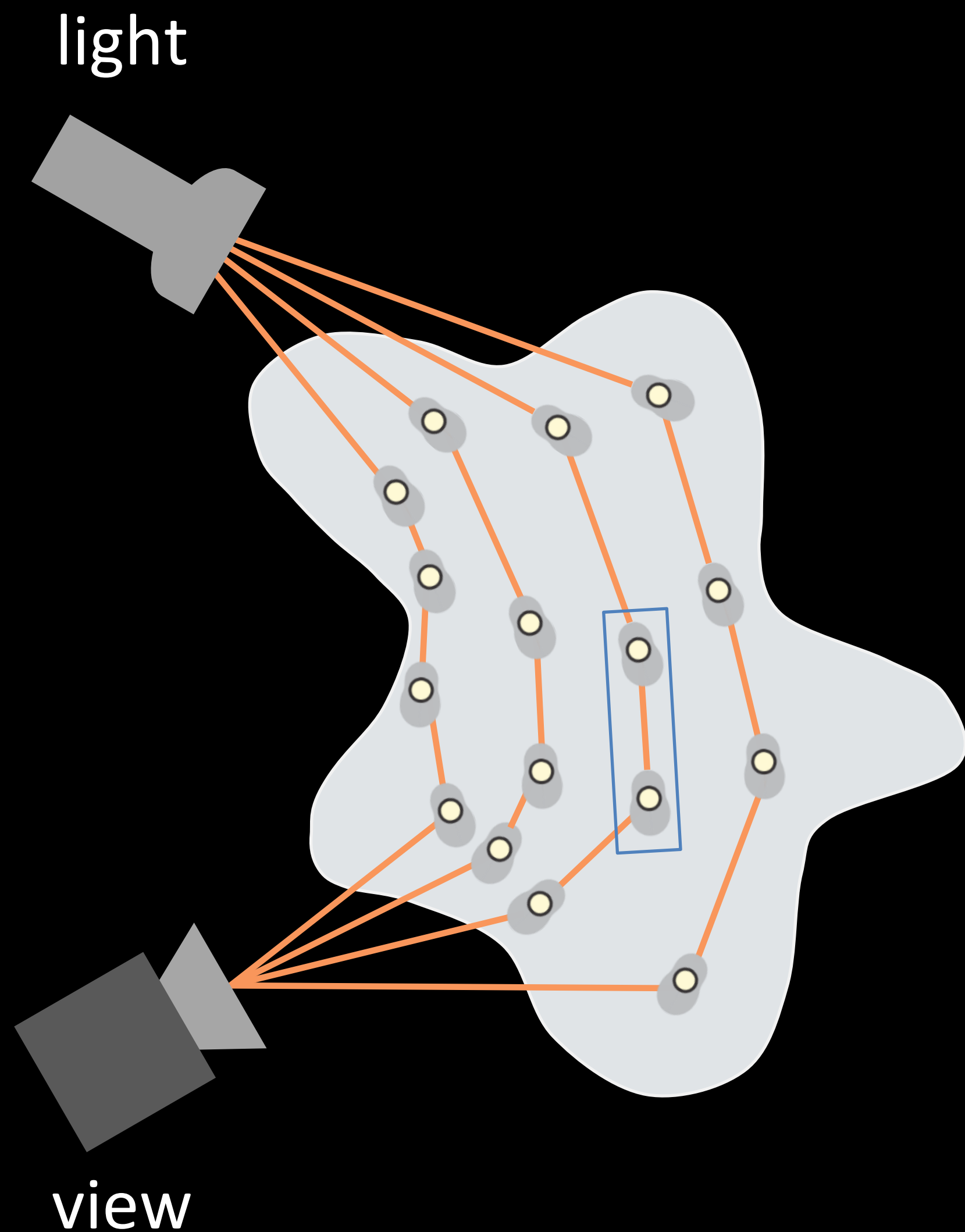
Recap: Monte Carlo (volumetric) rendering



$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

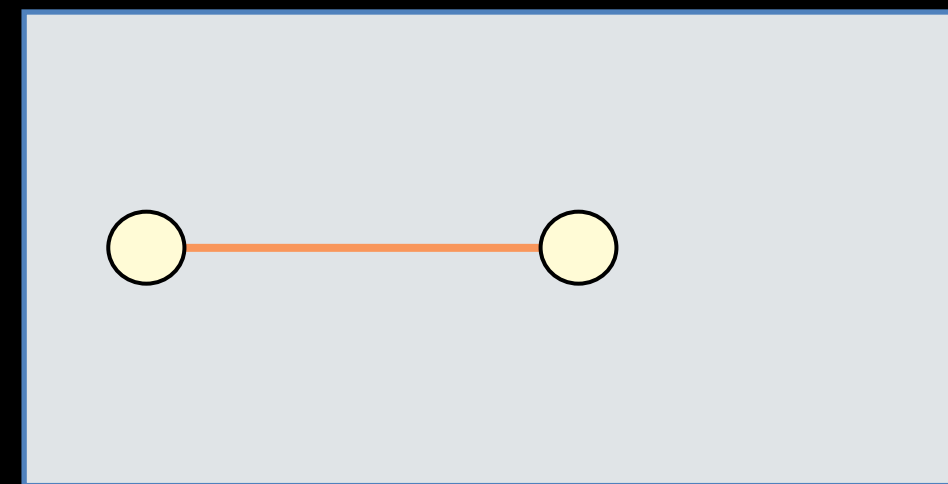
Path contribution, depends on the scattering material

Recap: Monte Carlo (volumetric) rendering



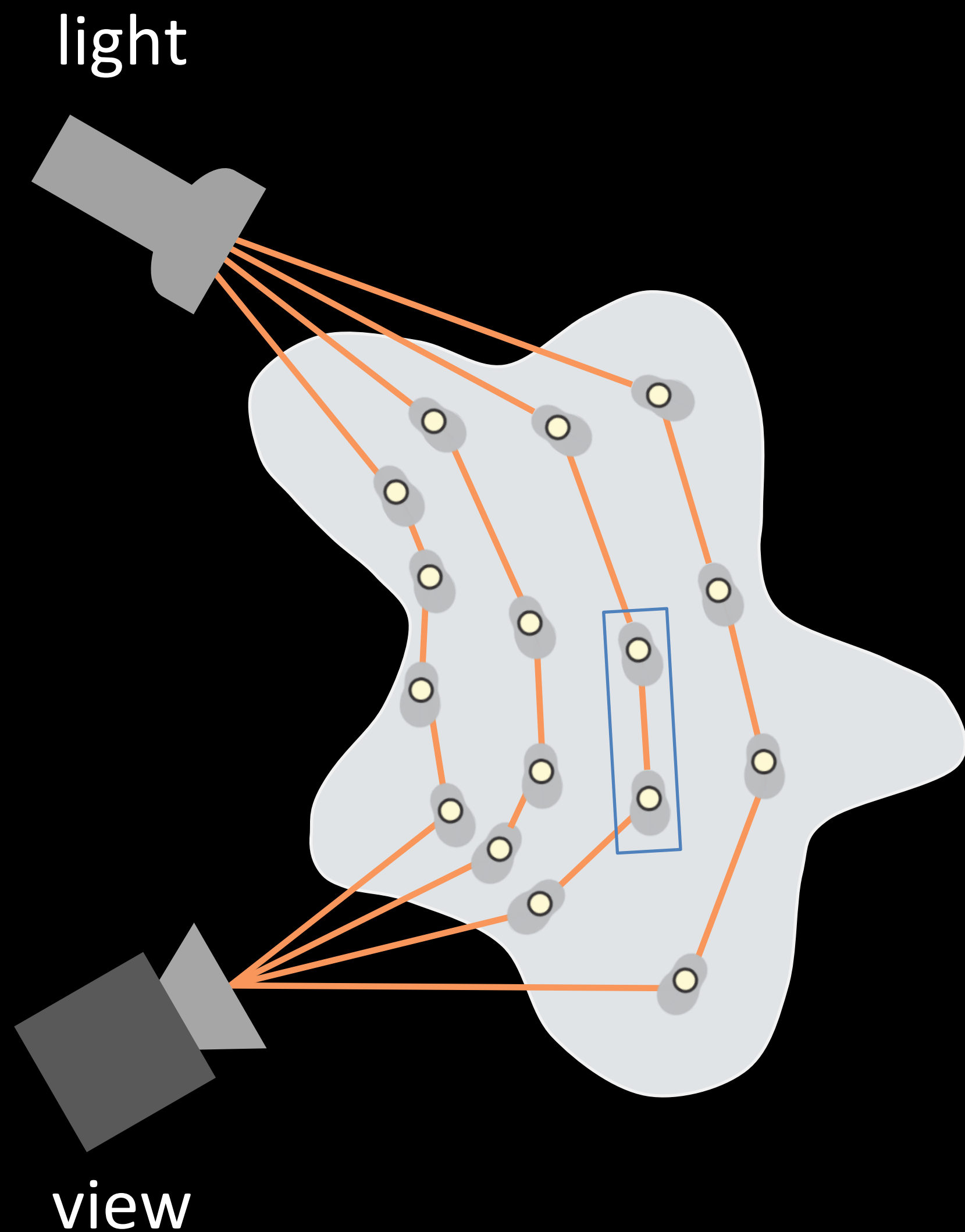
$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

Path contribution, depends on the scattering material



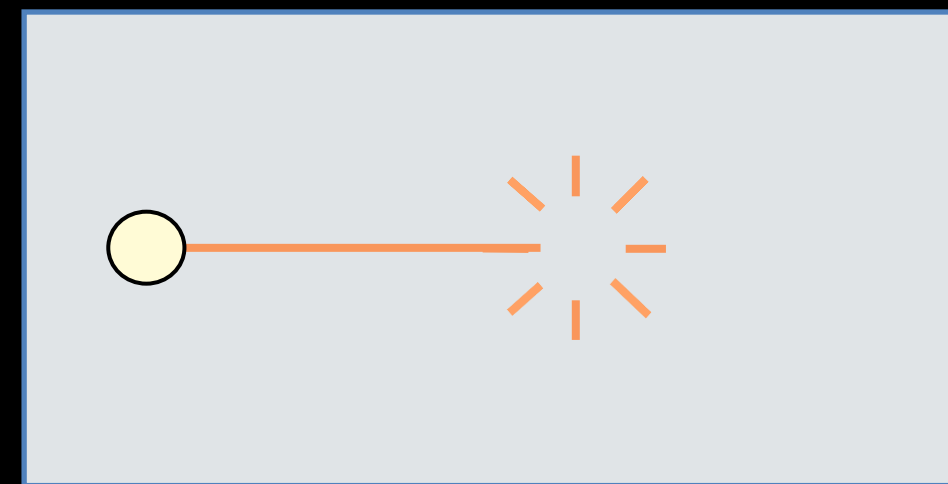
volumetric density
(extinction coefficient) σ

Recap: Monte Carlo (volumetric) rendering



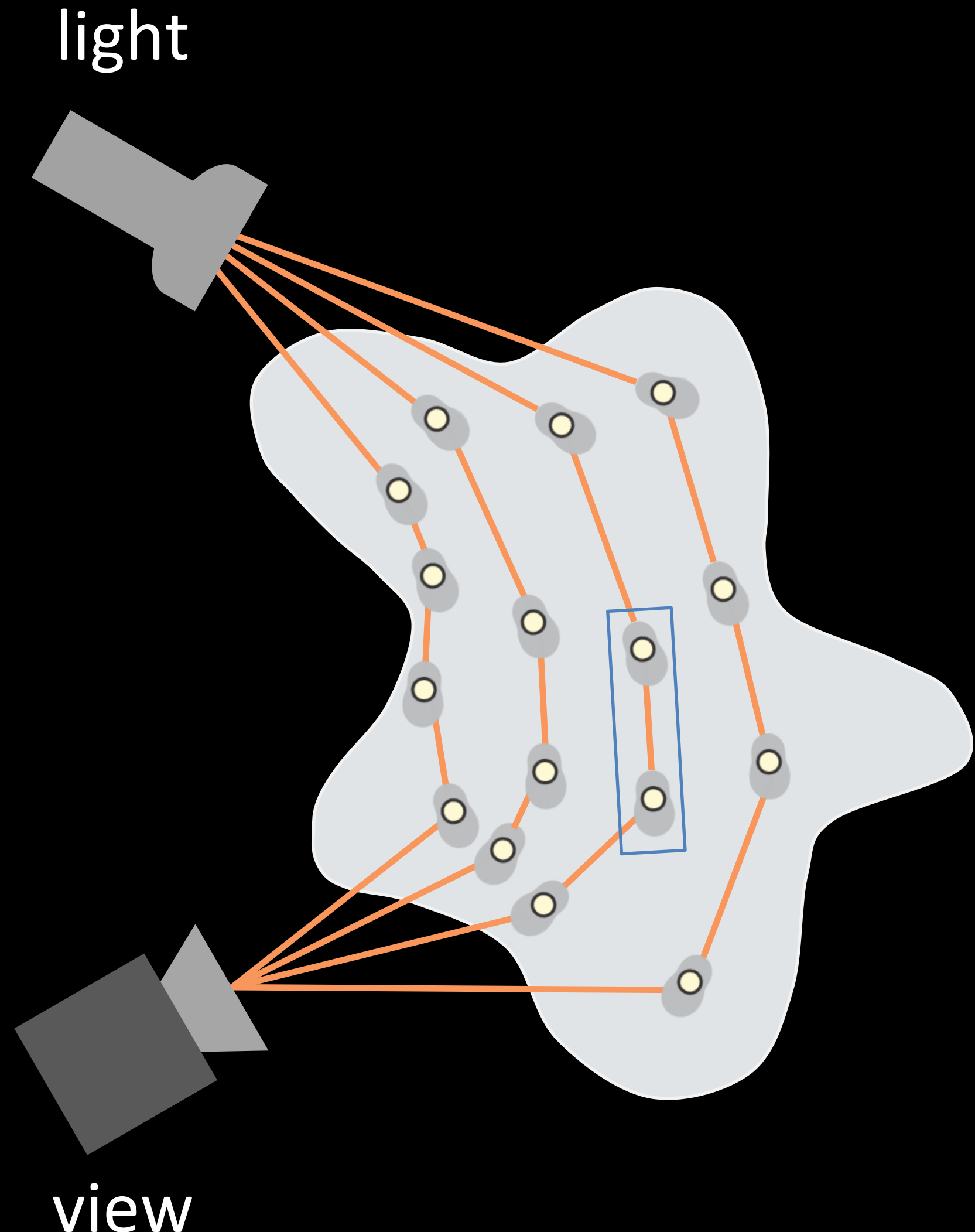
$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

Path contribution, depends on the scattering material



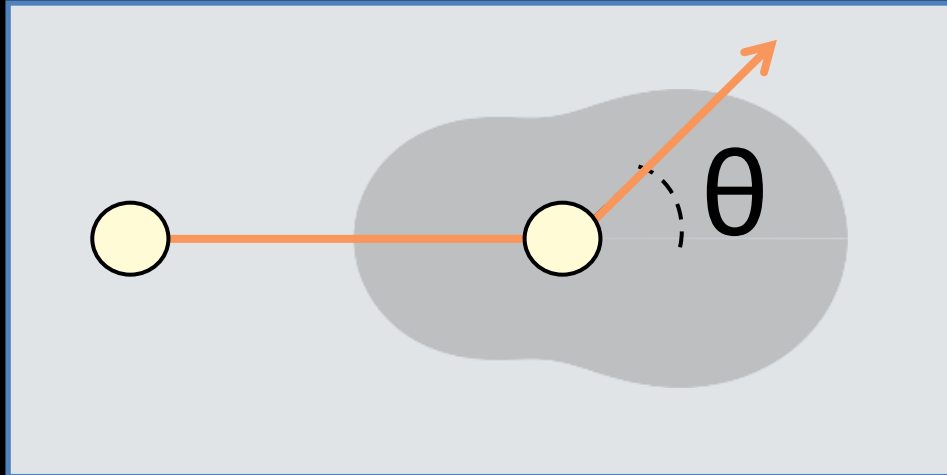
volumetric density
(extinction coefficient) σ
scattering albedo a

Recap: Monte Carlo (volumetric) rendering



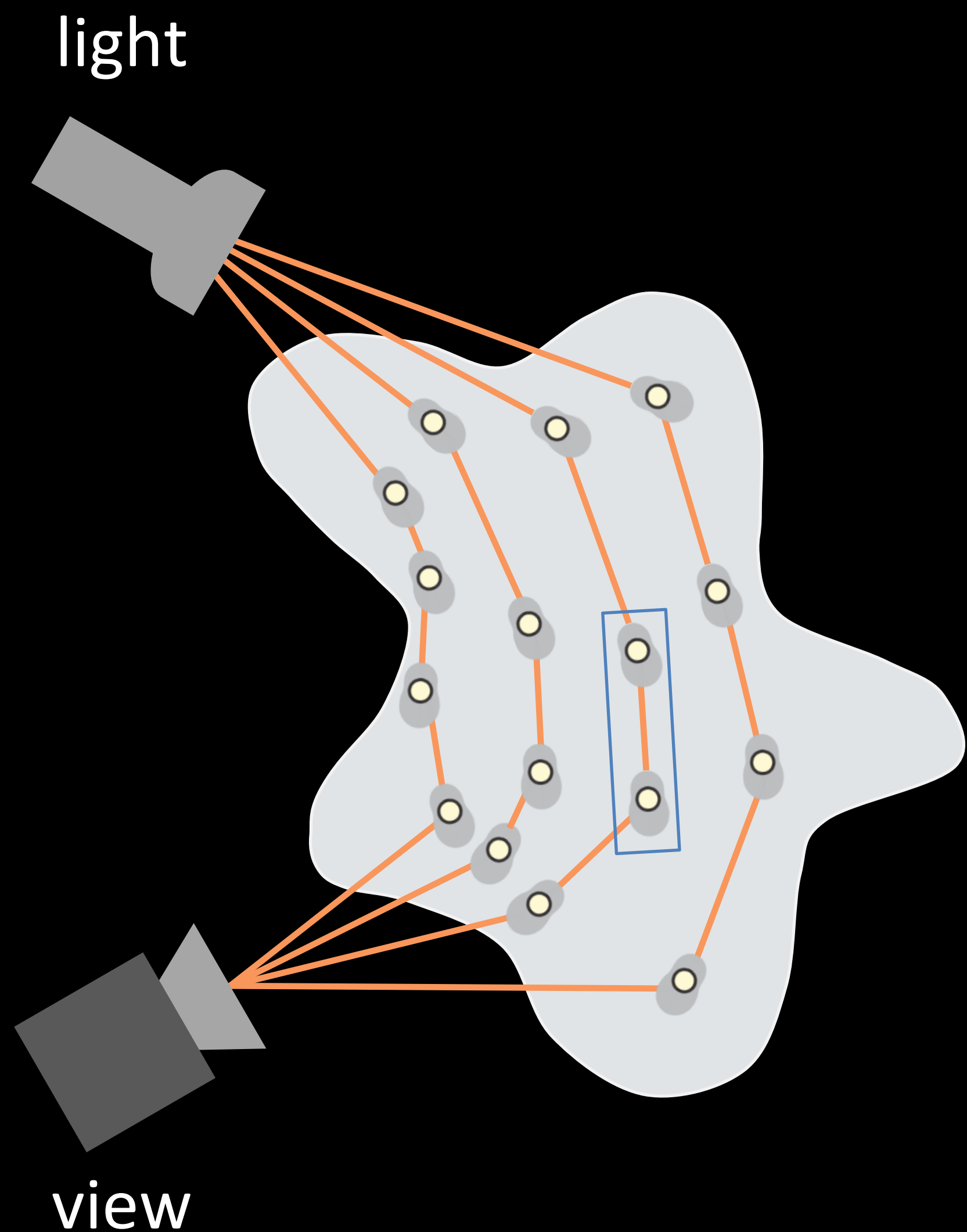
$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

Path contribution, depends on the scattering material



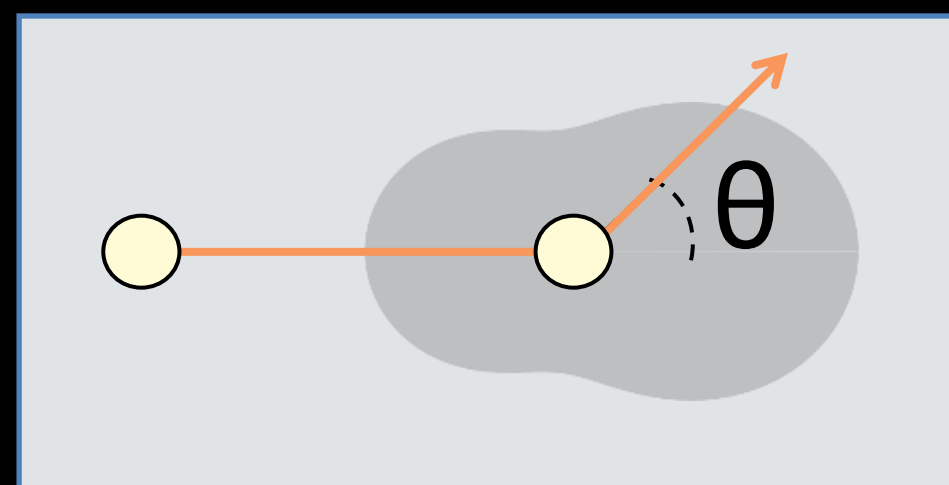
- volumetric density (extinction coefficient) σ
- scattering albedo a
- phase function p_{θ}

Recap: Monte Carlo (volumetric) rendering



$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

Path contribution, depends on the scattering material

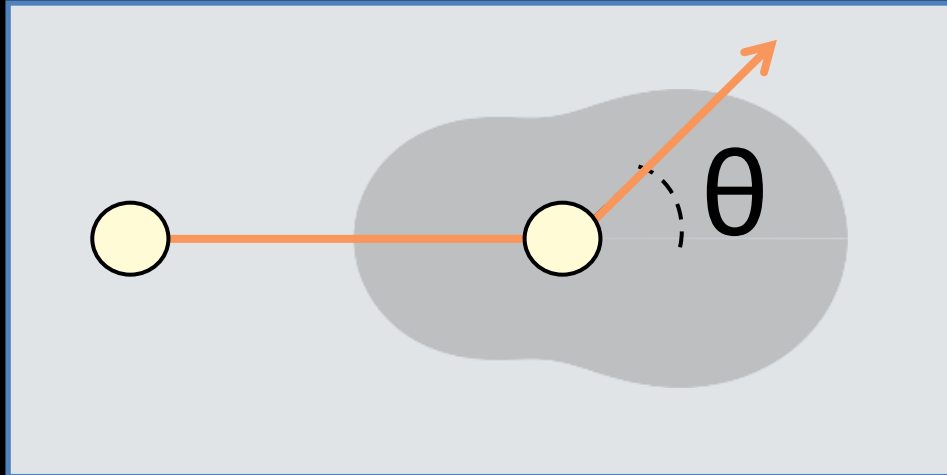
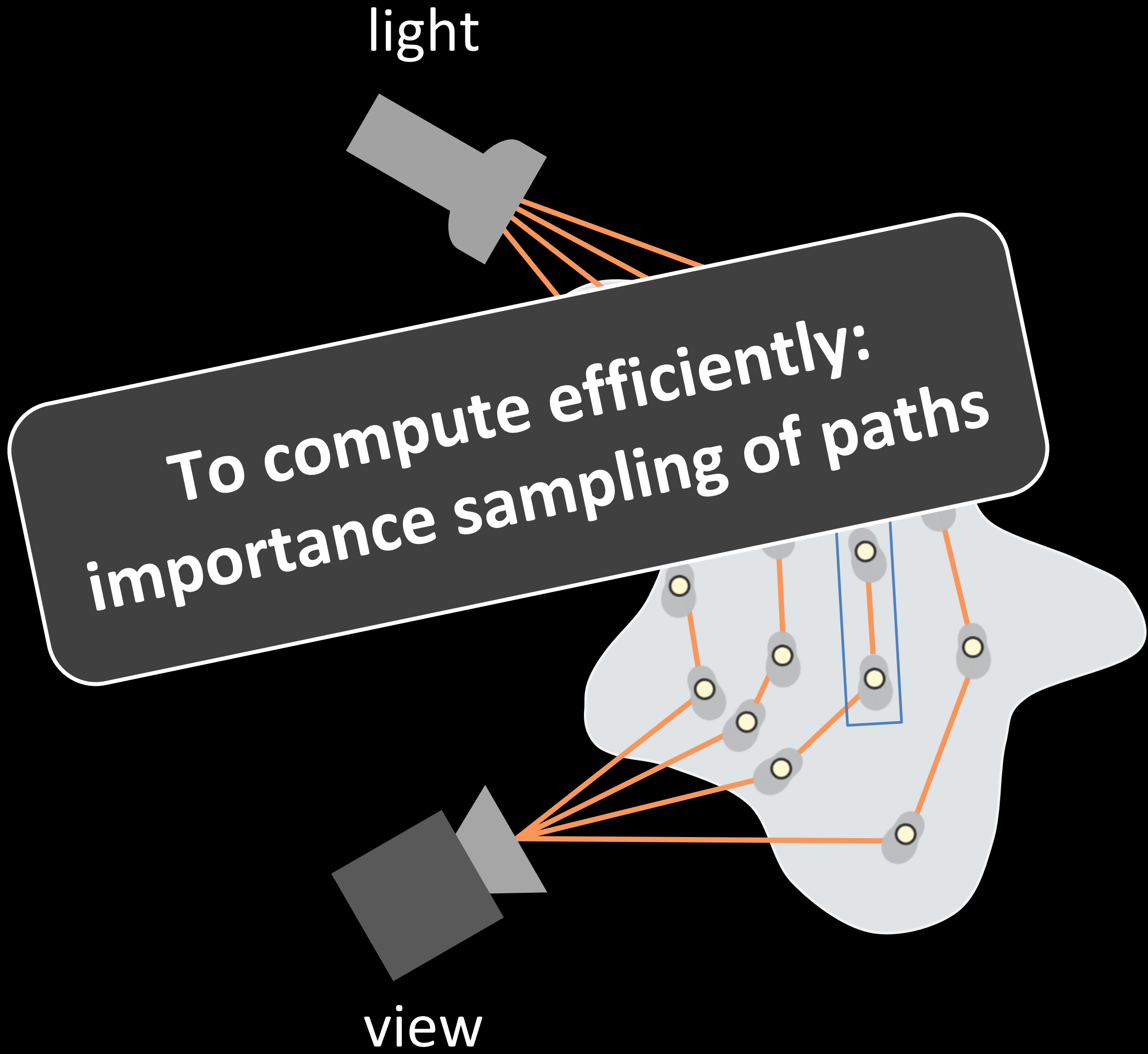


$$\text{material} = \begin{bmatrix} \sigma \\ a \\ p_{\theta} \end{bmatrix}$$

Recap: Monte Carlo (volumetric) rendering

$$\text{Image} = \int_{\text{paths}} f(\text{path})$$

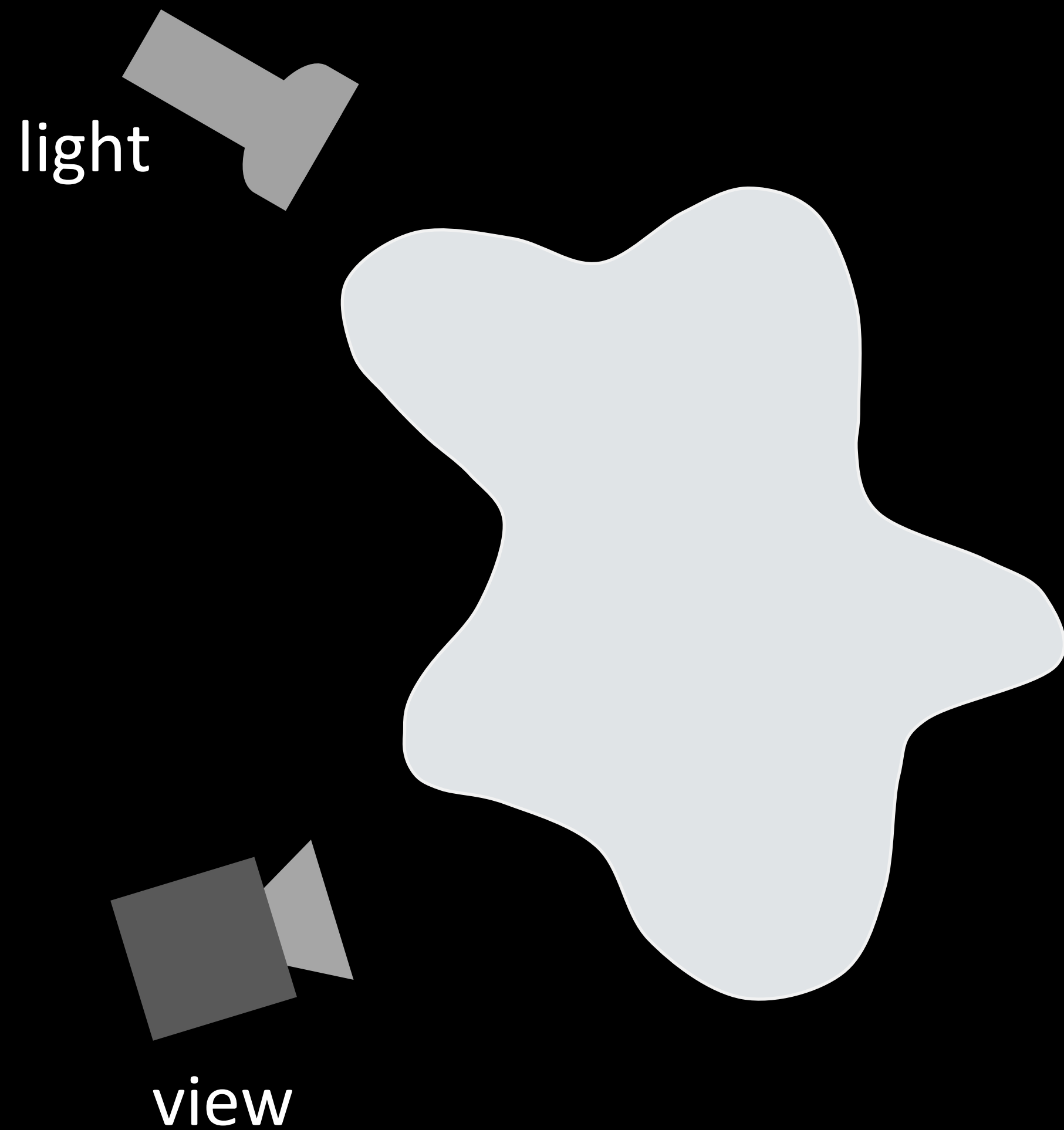
Path contribution, depends on the scattering material



$$\text{material} = \begin{bmatrix} \sigma \\ a \\ p_{\theta} \end{bmatrix}$$

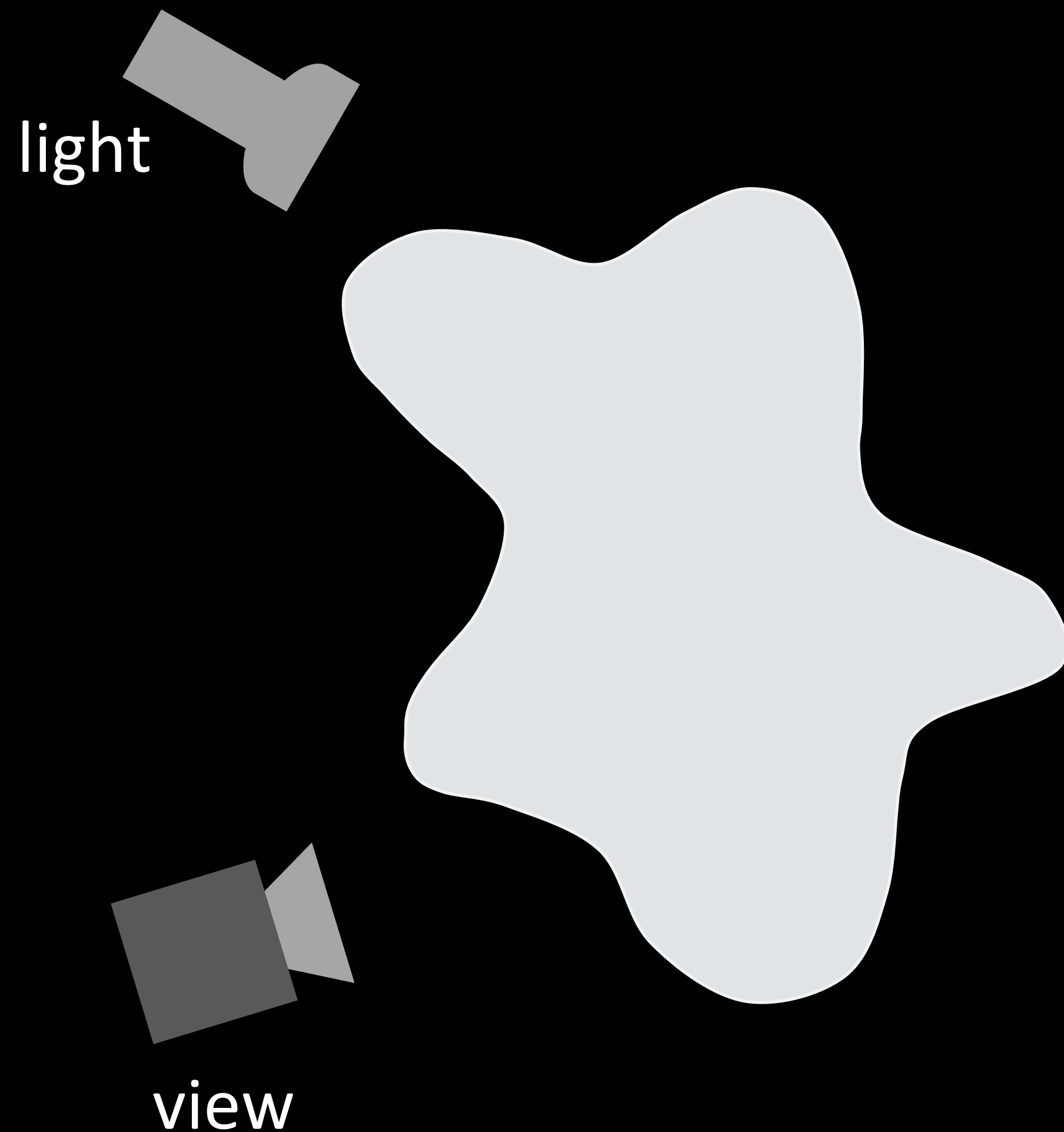
Memory effect: simulate *covariance* of speckle images

$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



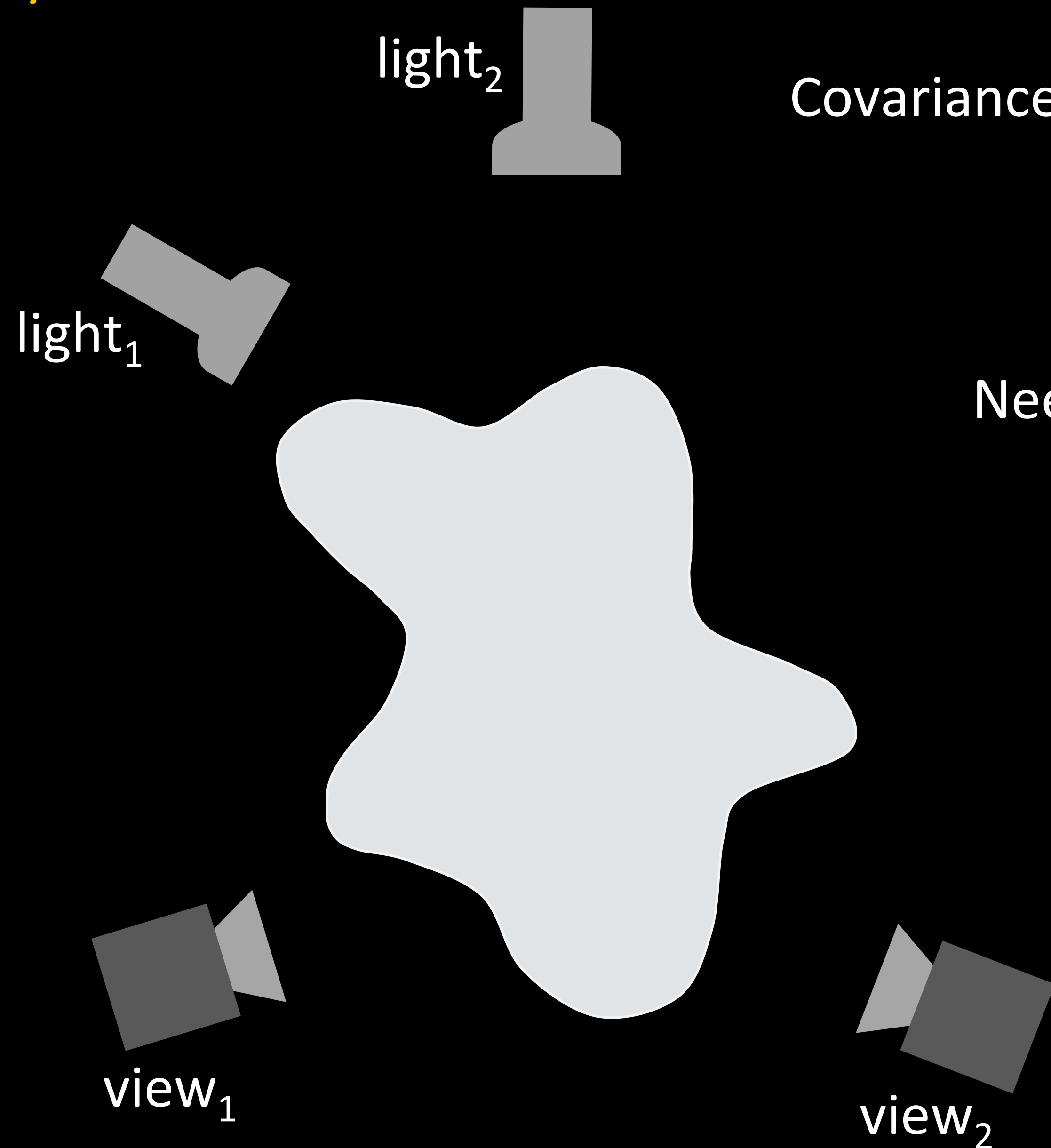
Memory effect: simulate *covariance* of speckle images

$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



Need to consider products of *pairs* of paths

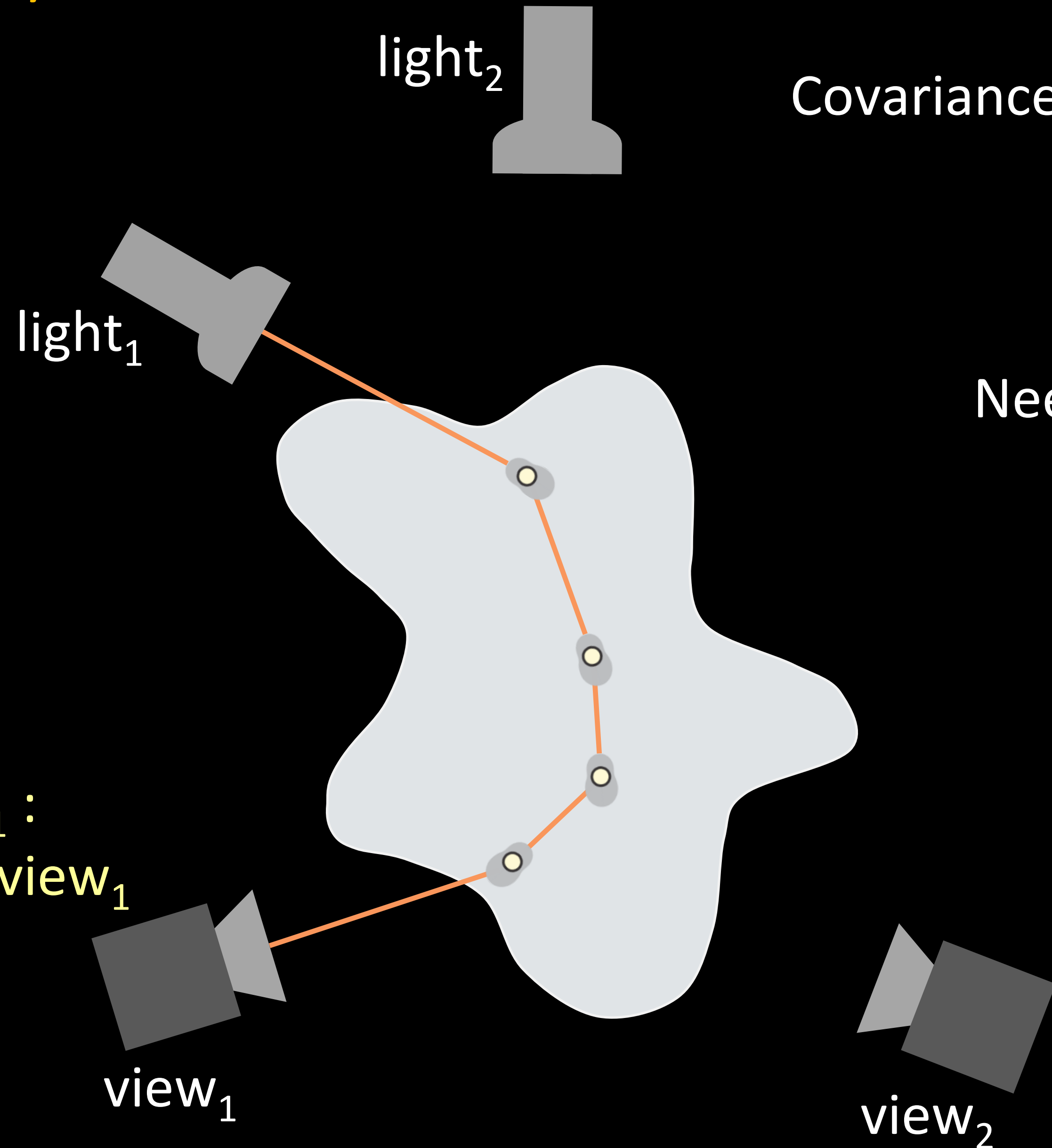
Memory effect: simulate *covariance* of speckle images



$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Need to consider products of *pairs* of paths

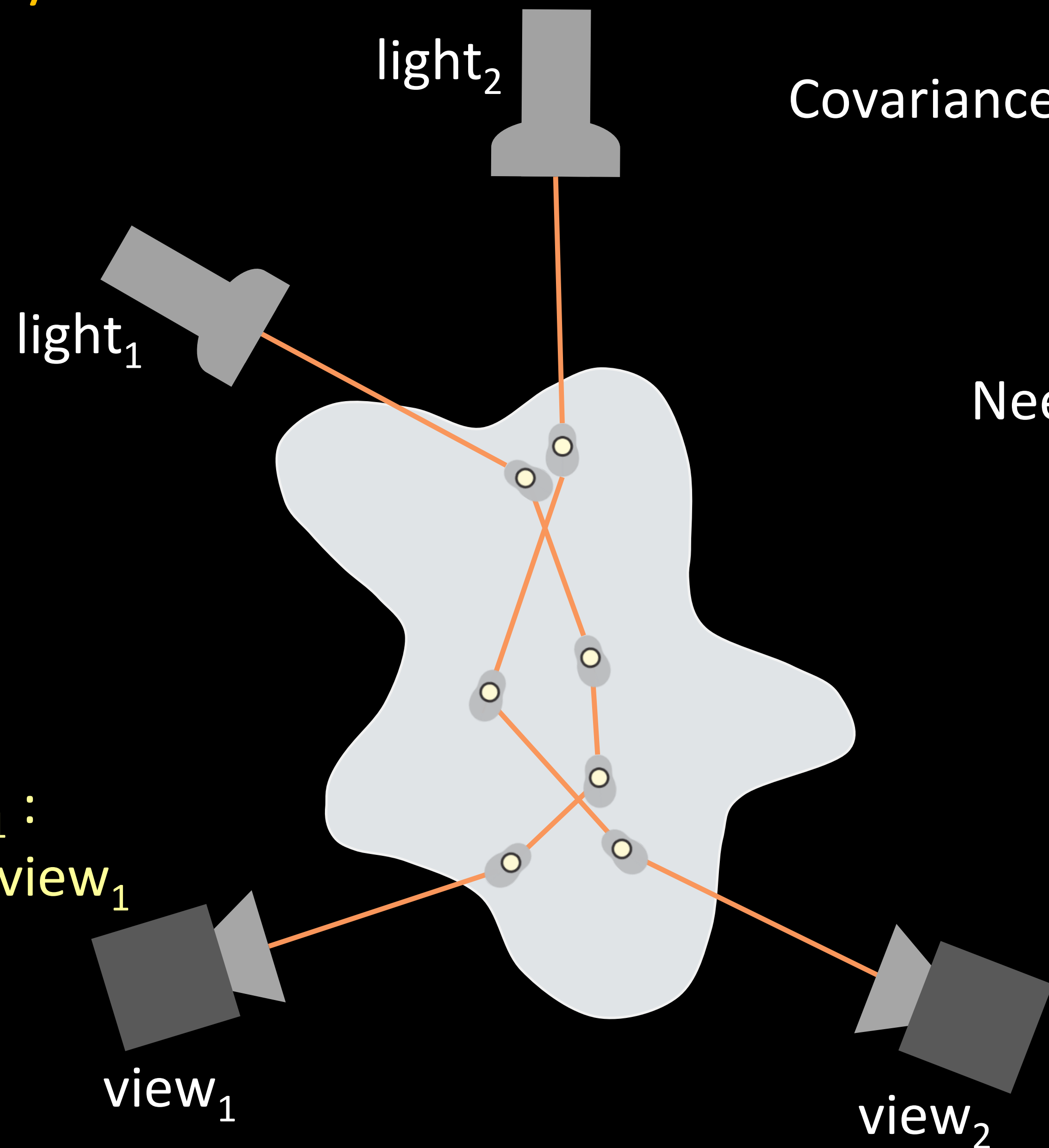
Memory effect: simulate *covariance* of speckle images



$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Need to consider products of ***pairs*** of paths

Memory effect: simulate *covariance* of speckle images



$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Need to consider products of *pairs* of paths

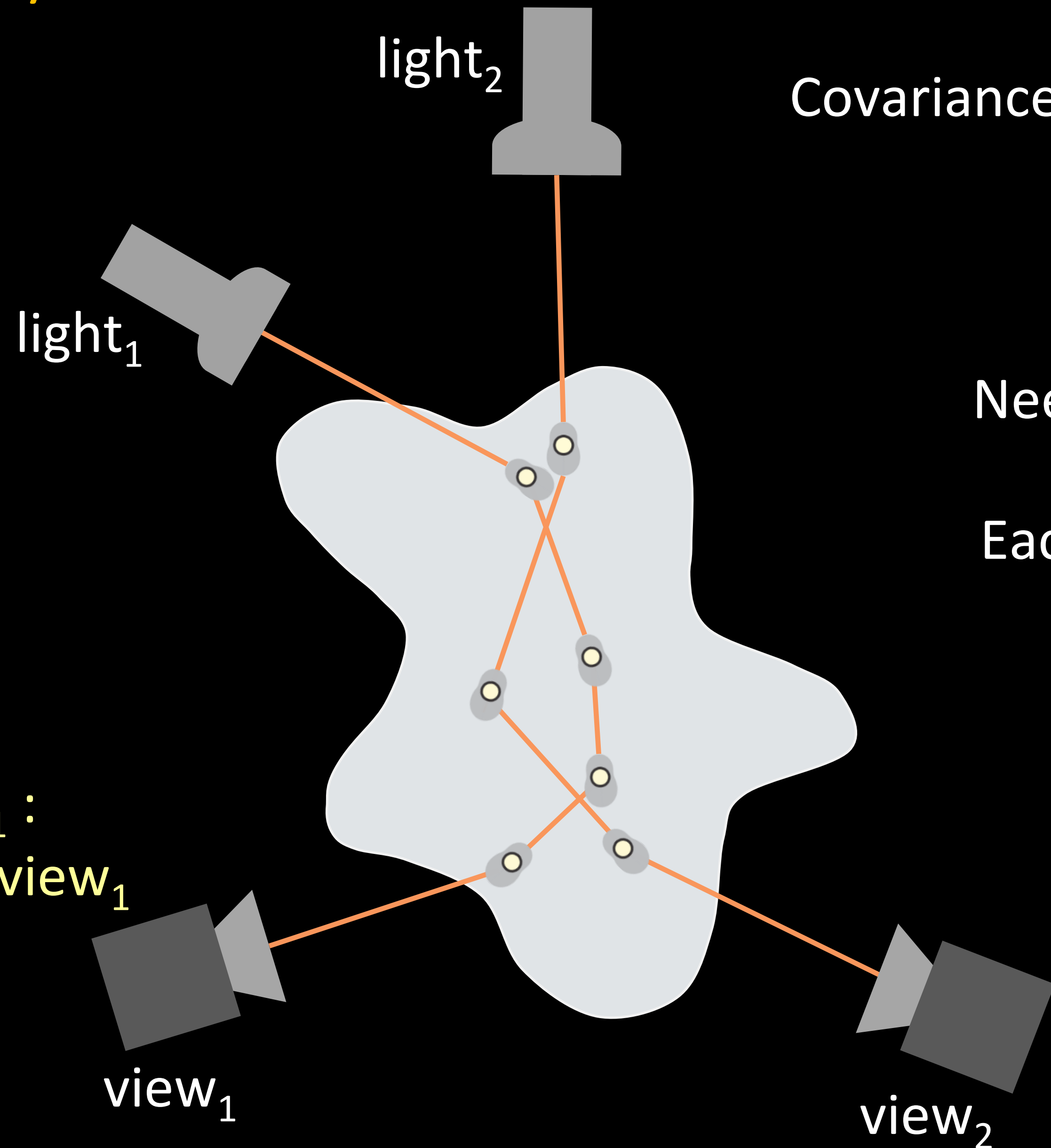
$path_1$:
 $light_1 \rightarrow view_1$

$view_1$

$view_2$

$path_2$:
 $light_2 \rightarrow view_2$

Memory effect: simulate *covariance* of speckle images



$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

$$u = |u| e^{i \cdot \text{phase}}$$

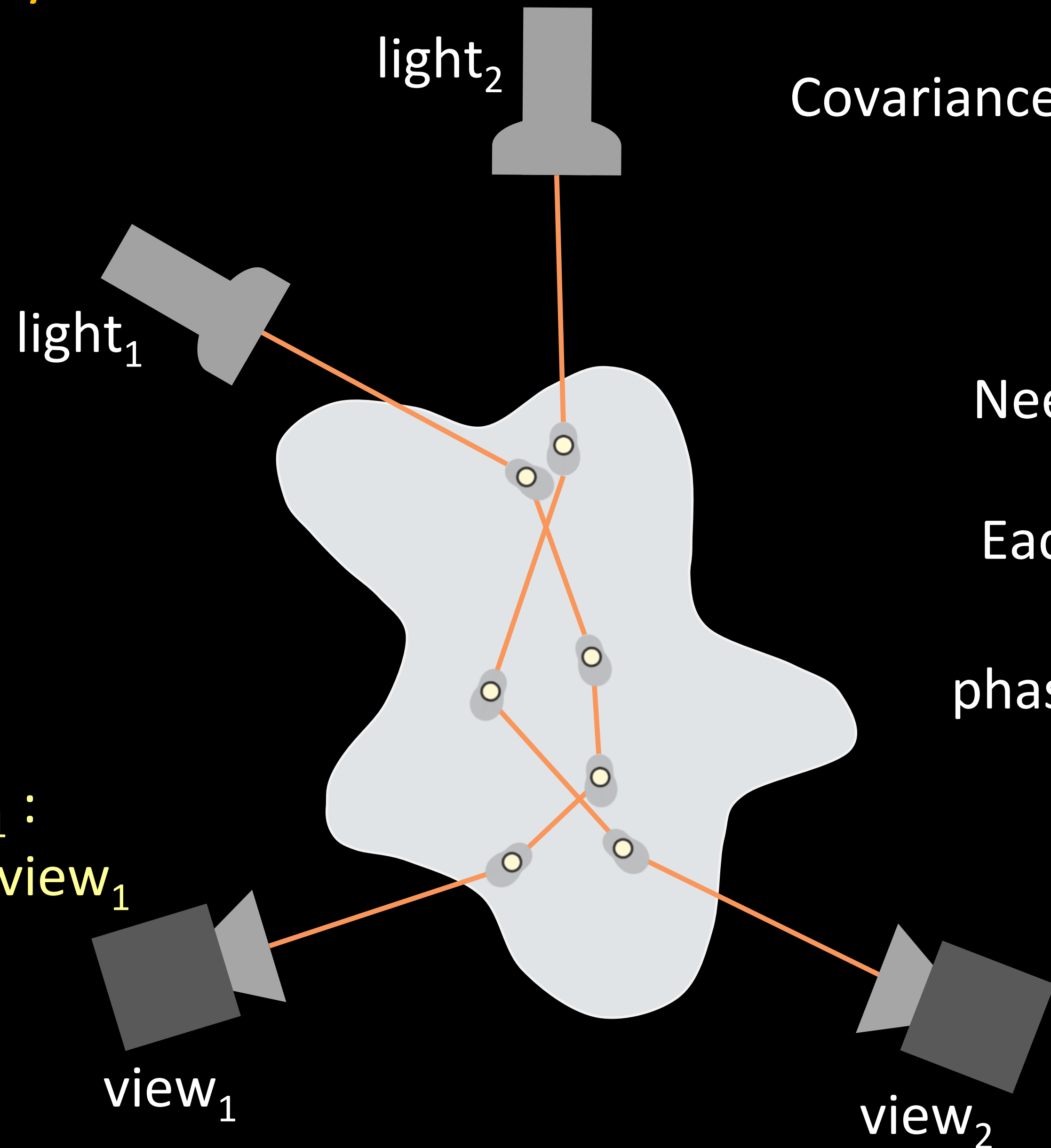
Need to consider products of **pairs** of paths

Each path contributes a complex number u

path₁ :
light₁ → view₁

path₂ :
light₂ → view₂

Memory effect: simulate *covariance* of speckle images



$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

$$u = |u| e^{i \cdot \text{phase}}$$

Need to consider products of ***pairs*** of paths

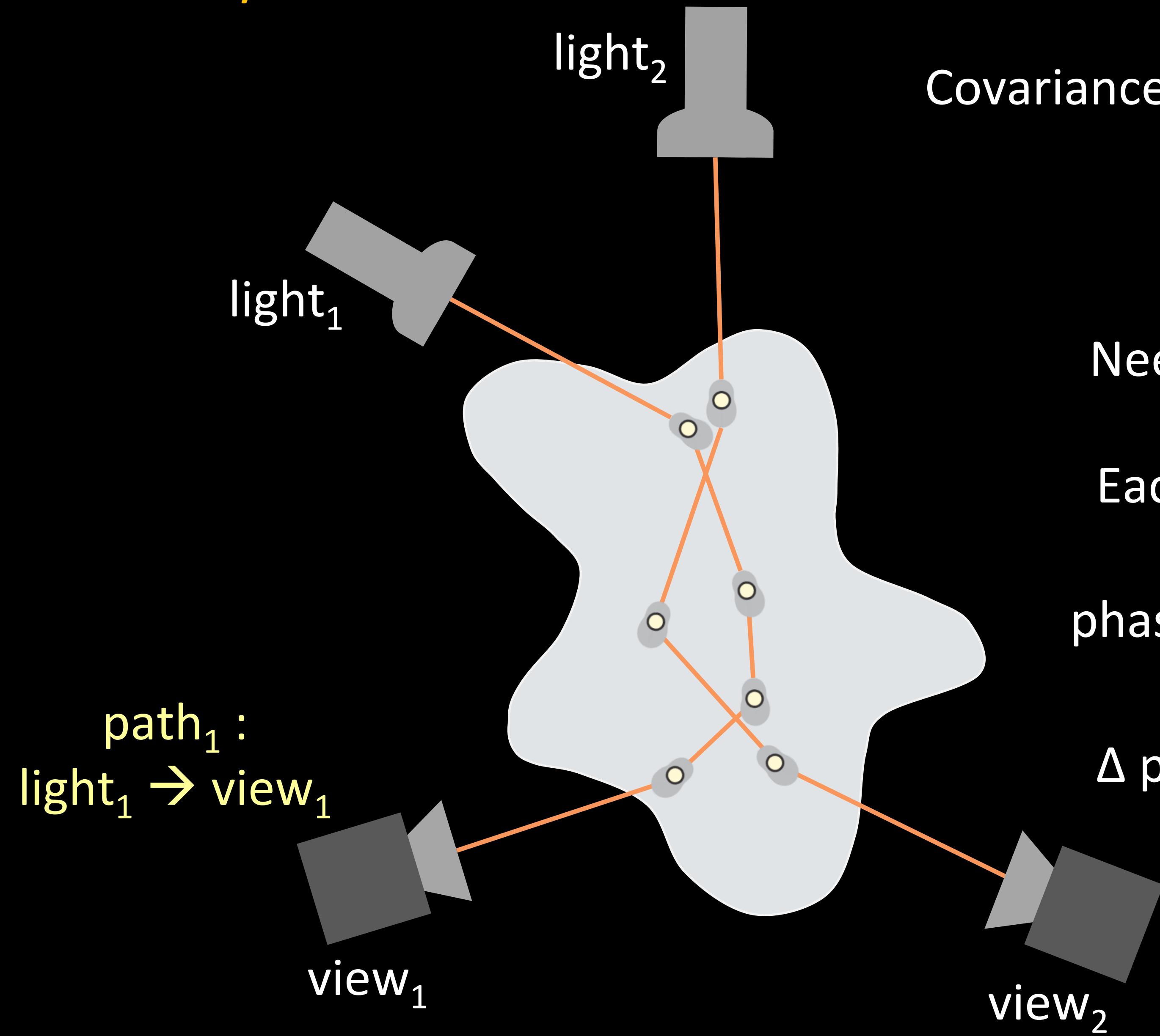
Each path contributes a complex number u

phase \propto Length (path)

$path_1$:
 $light_1 \rightarrow view_1$

$path_2$:
 $light_2 \rightarrow view_2$

Memory effect: simulate *covariance* of speckle images



$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

$$u = |u| e^{i \cdot \text{phase}}$$

Need to consider products of **pairs** of paths

Each path contributes a complex number u

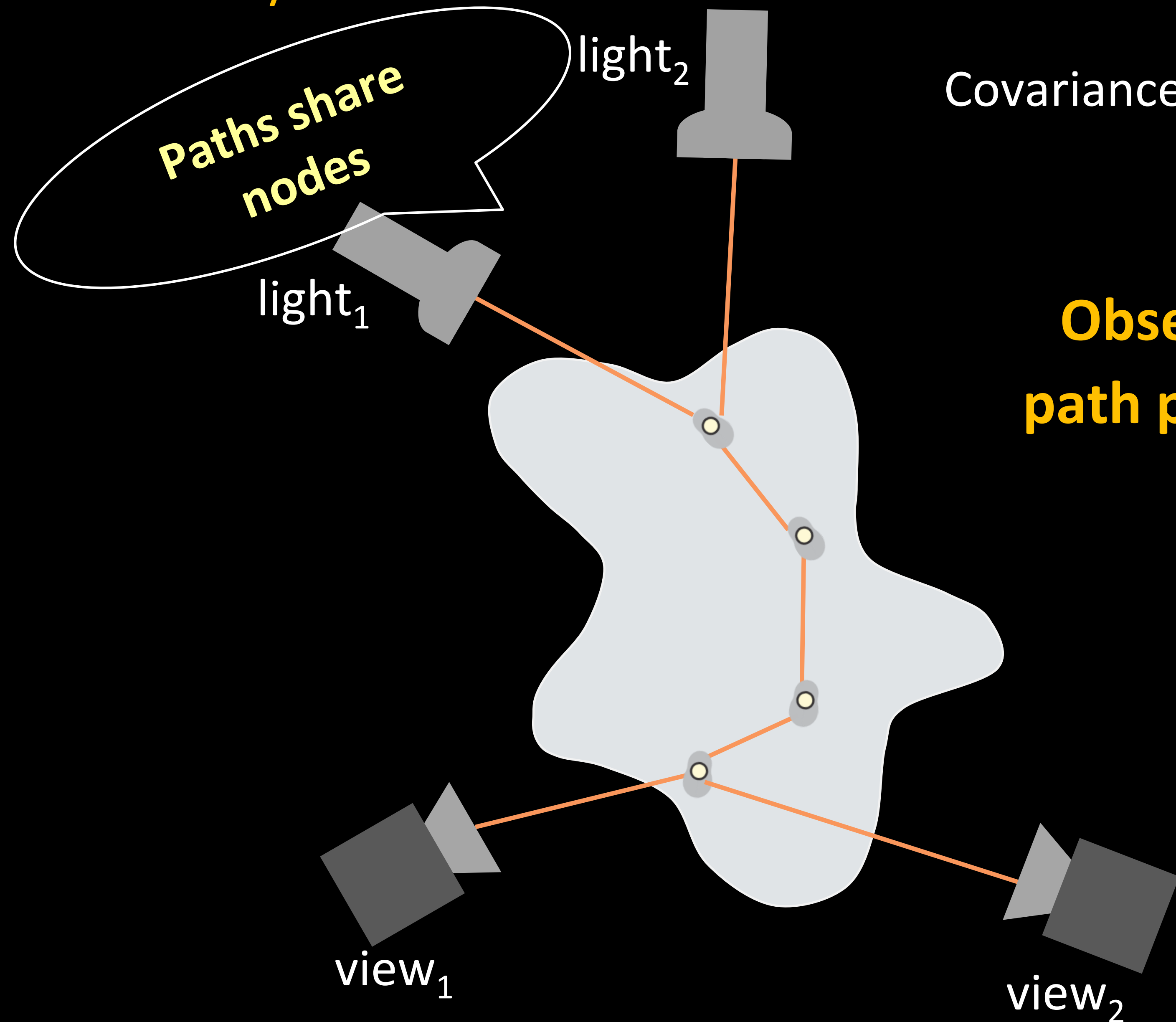
phase \propto Length (path)

Δ phase \propto Length (path₁) - Length (path₂)

path₁ :
light₁ → view₁

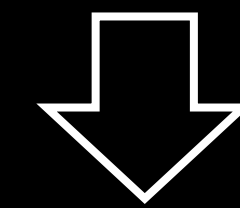
path₂ :
light₂ → view₂

Memory effect: simulate *covariance* of speckle images



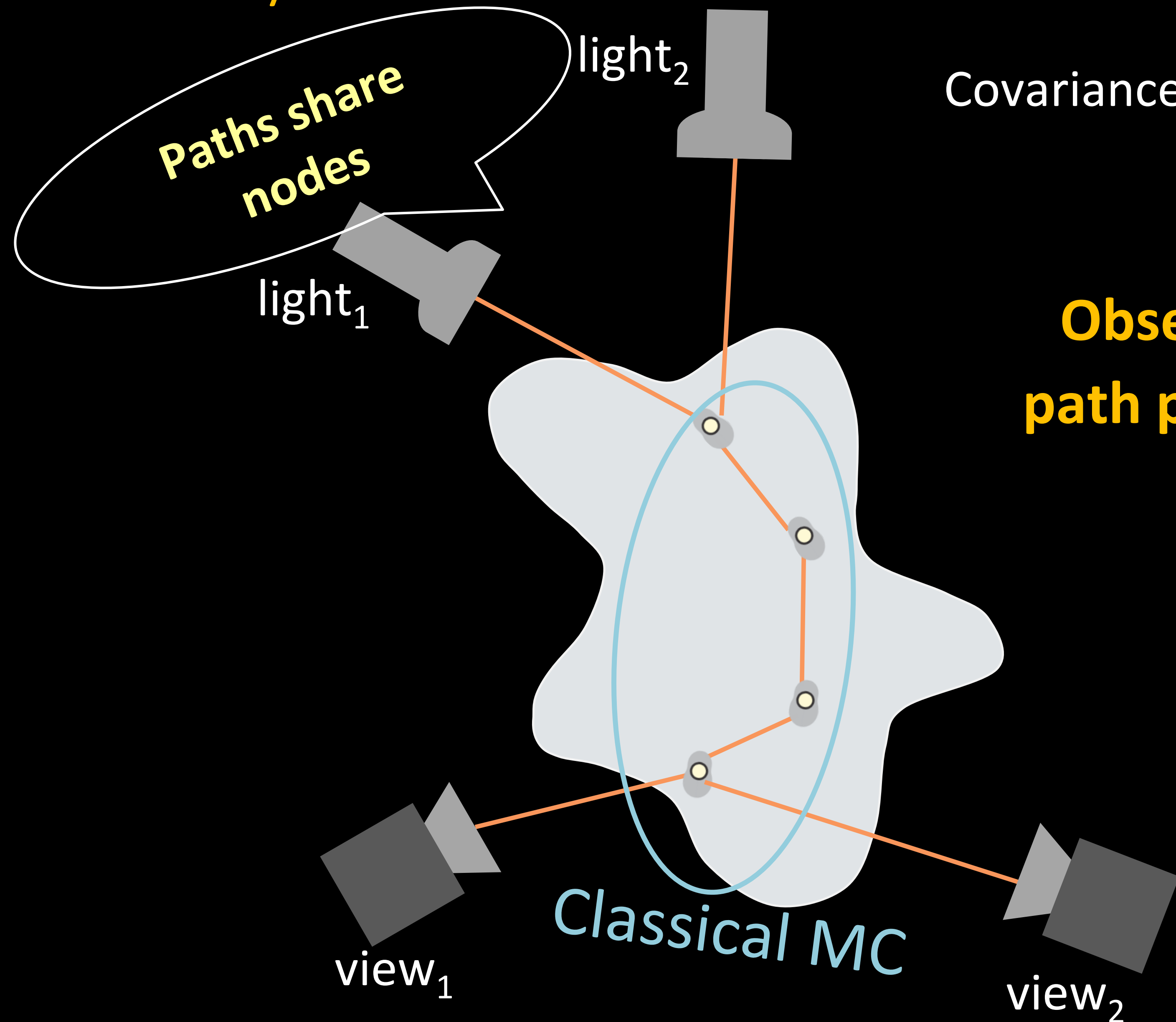
$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Observation: need to consider only path pairs that share the *same* nodes (except start and end)



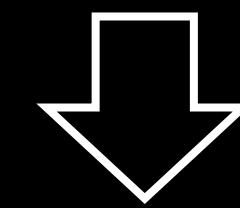
All other path pairs are averaged out in the integration

Memory effect: simulate *covariance* of speckle images



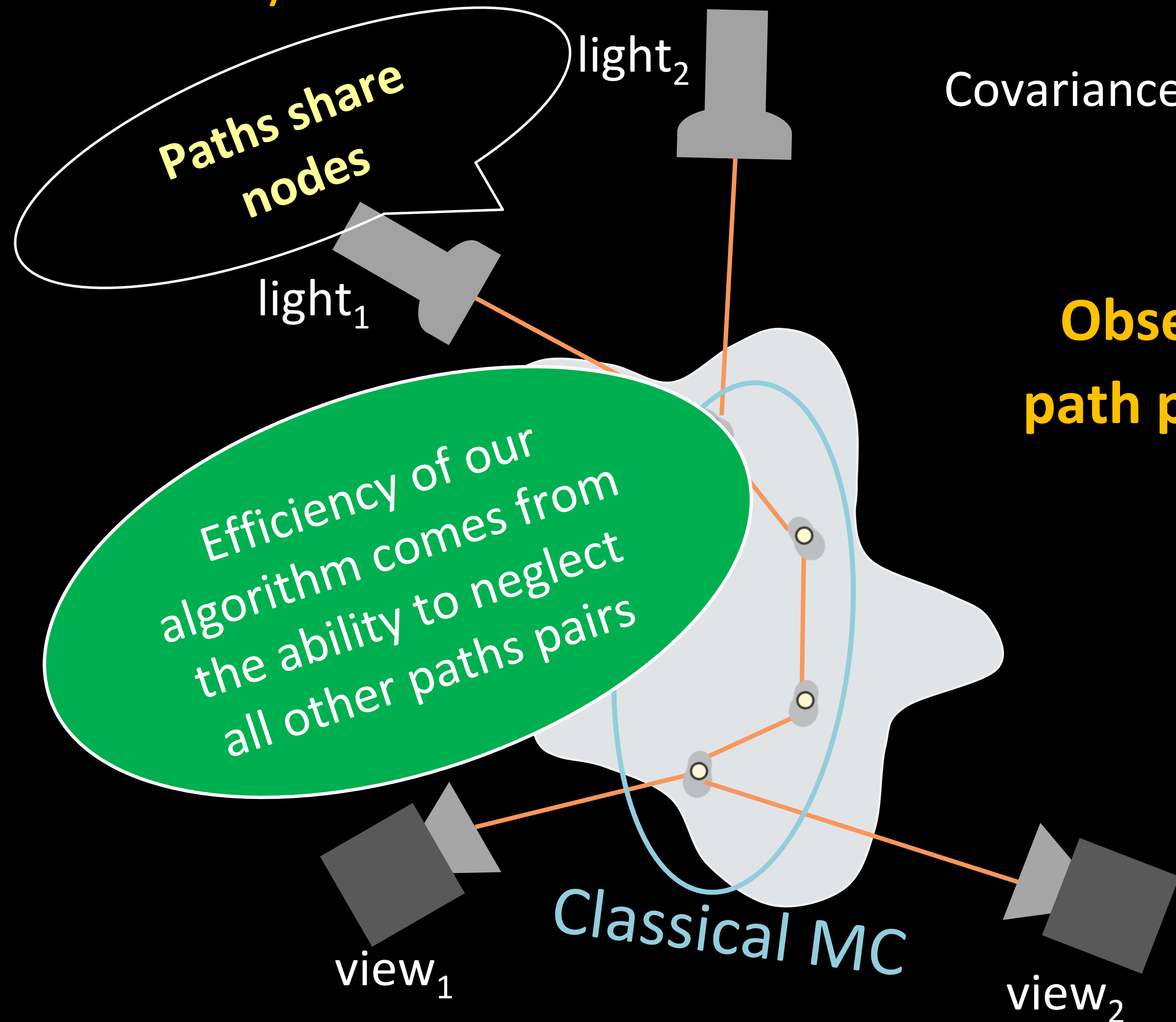
$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Observation: need to consider only path pairs that share the *same* nodes (except start and end)



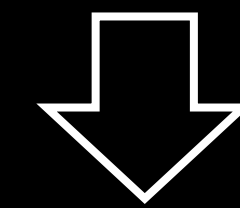
All other path pairs are averaged out in the integration

Memory effect: simulate *covariance* of speckle images



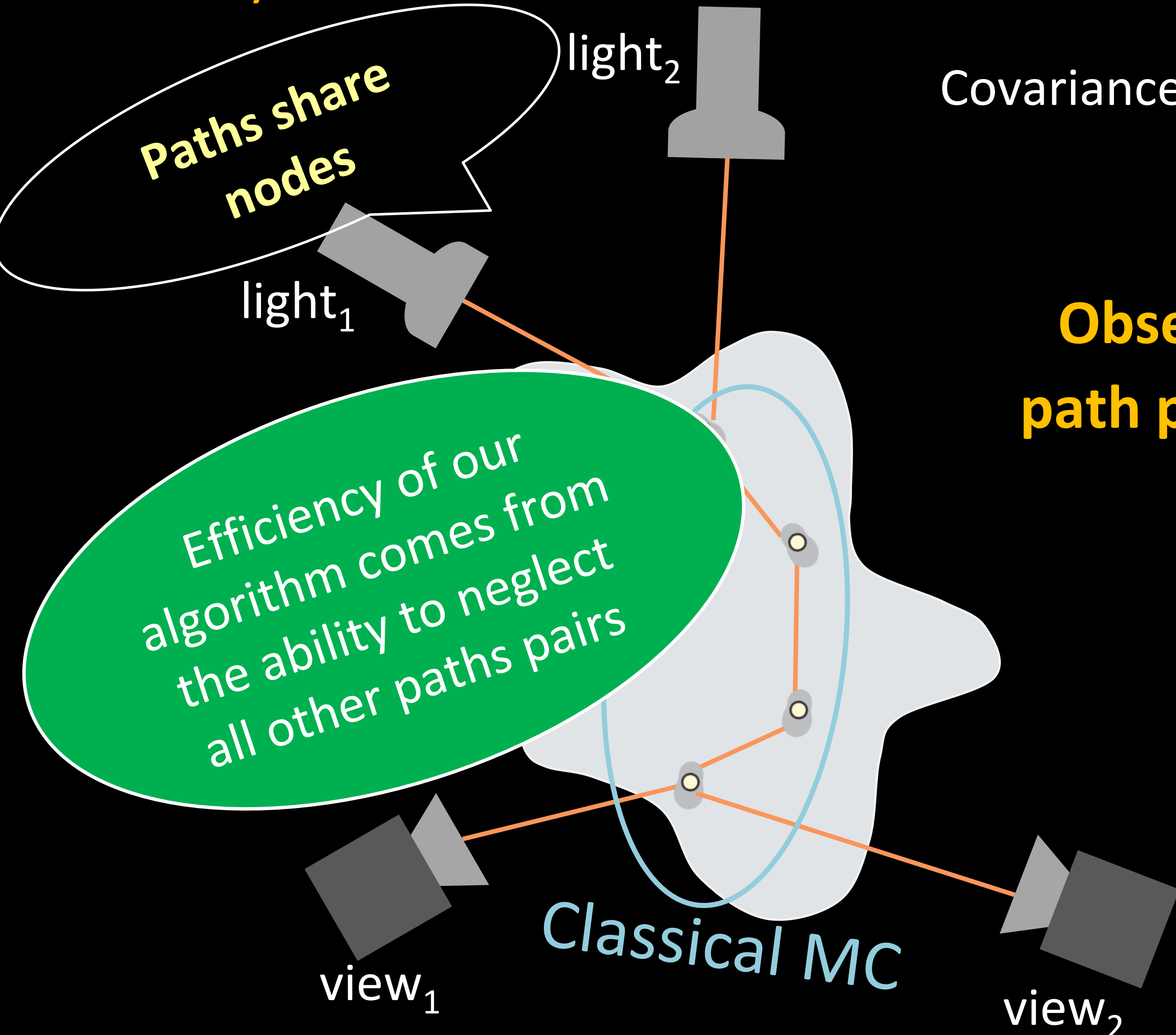
$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Observation: need to consider only path pairs that share the *same* nodes (except start and end)



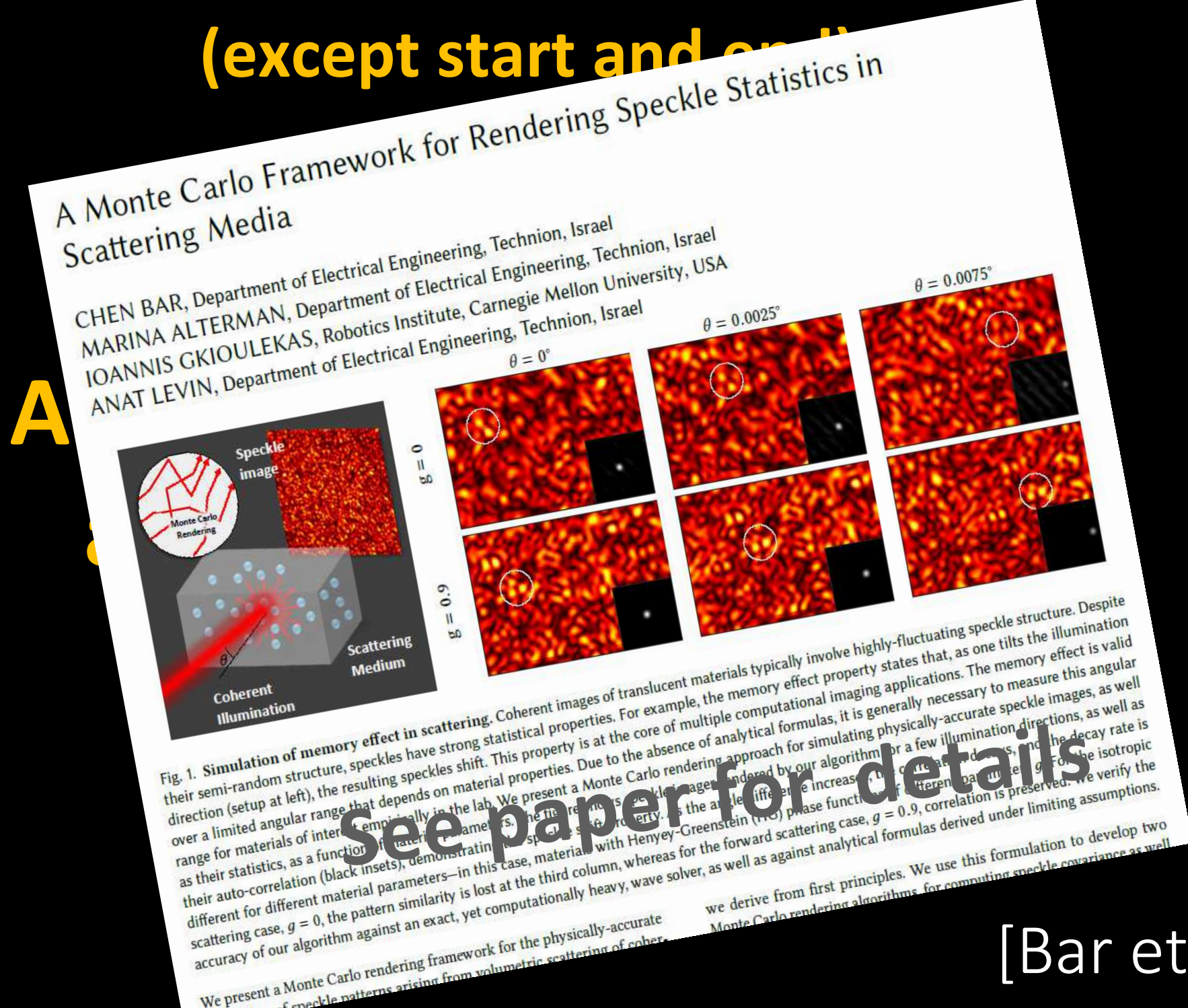
All other path pairs are averaged out in the integration

Memory effect: simulate *covariance* of speckle images



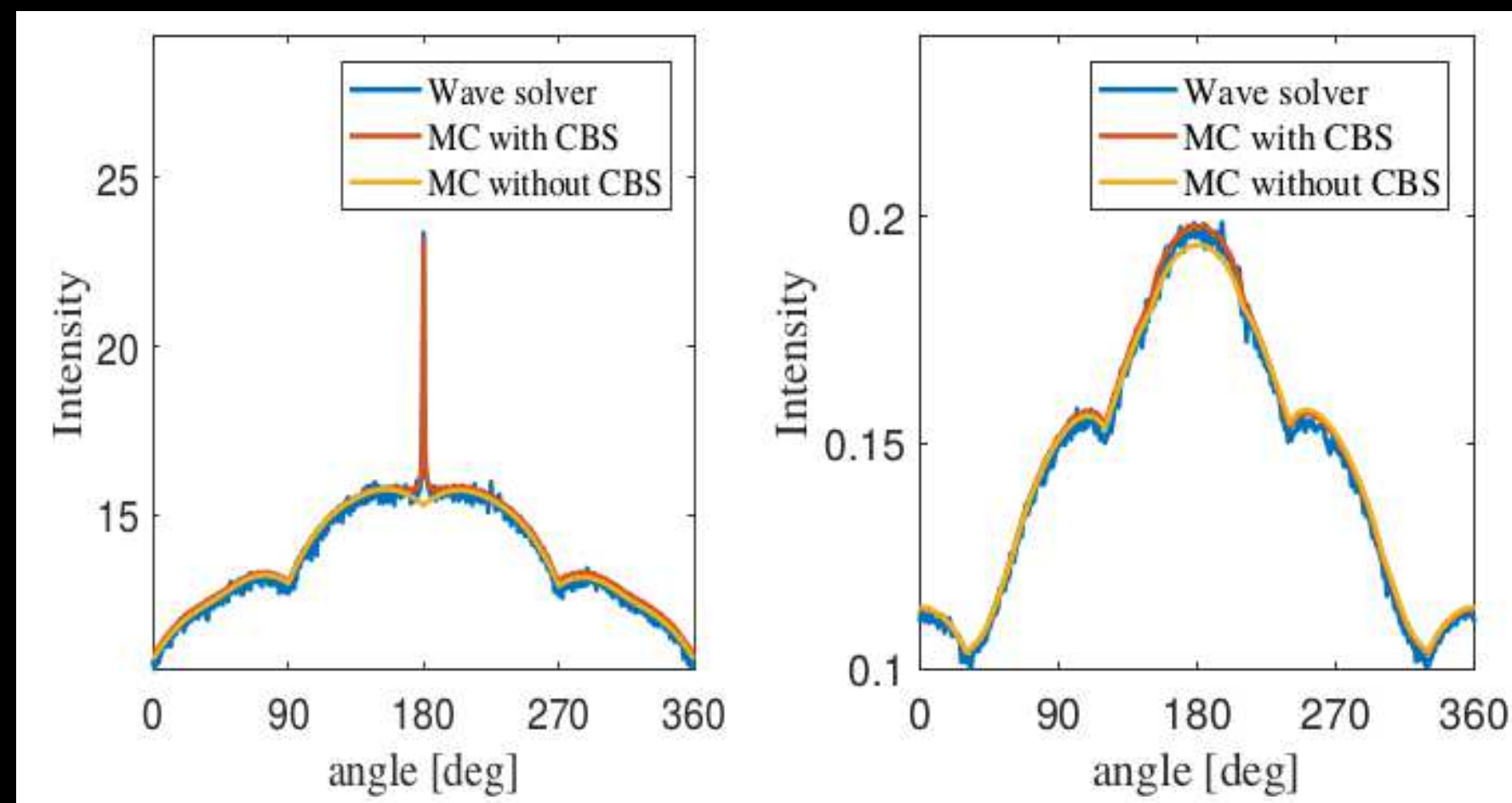
$$\text{Covariance} = \int_{\text{path}_1, \text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

Observation: need to consider only path pairs that share the *same* nodes (except start and end)

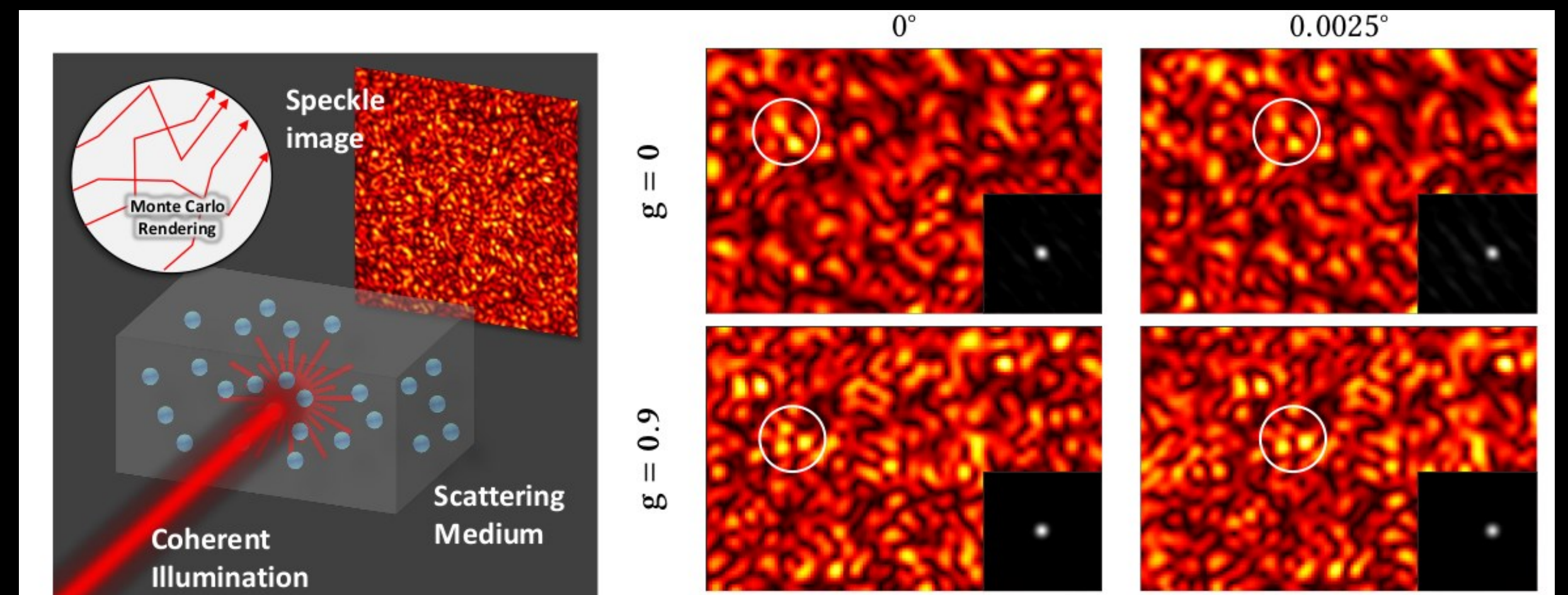


See paper for details

Comparison with wave-equation solver and real measurements

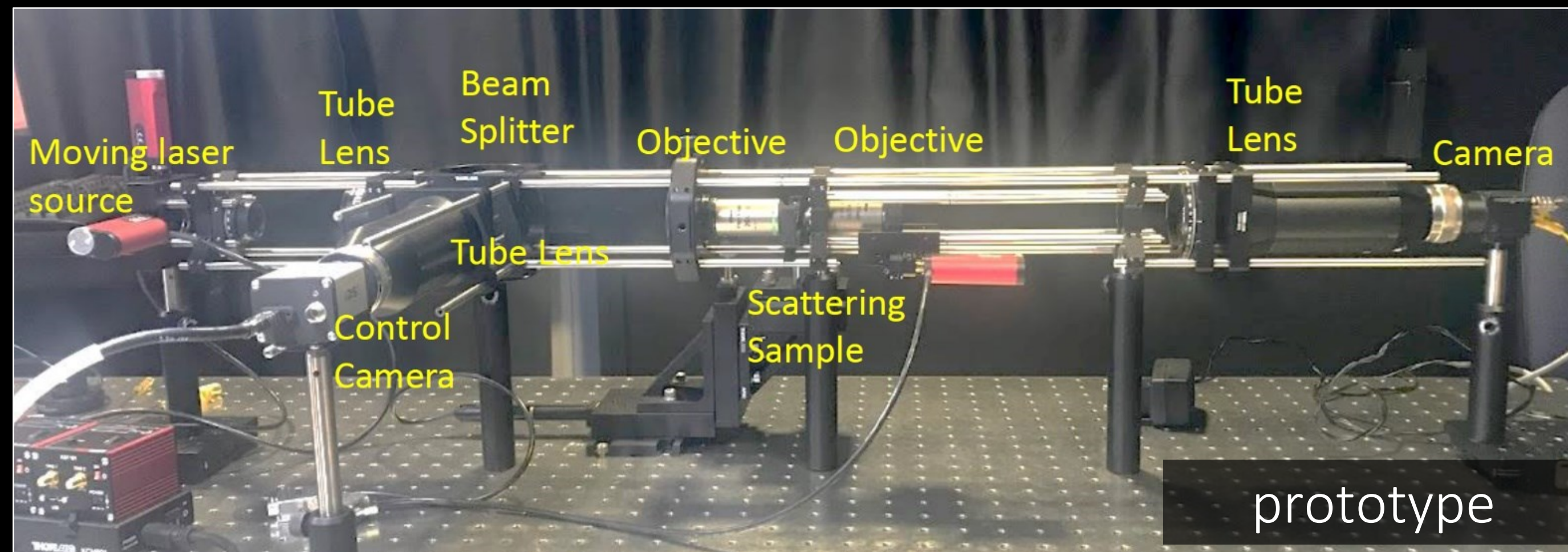
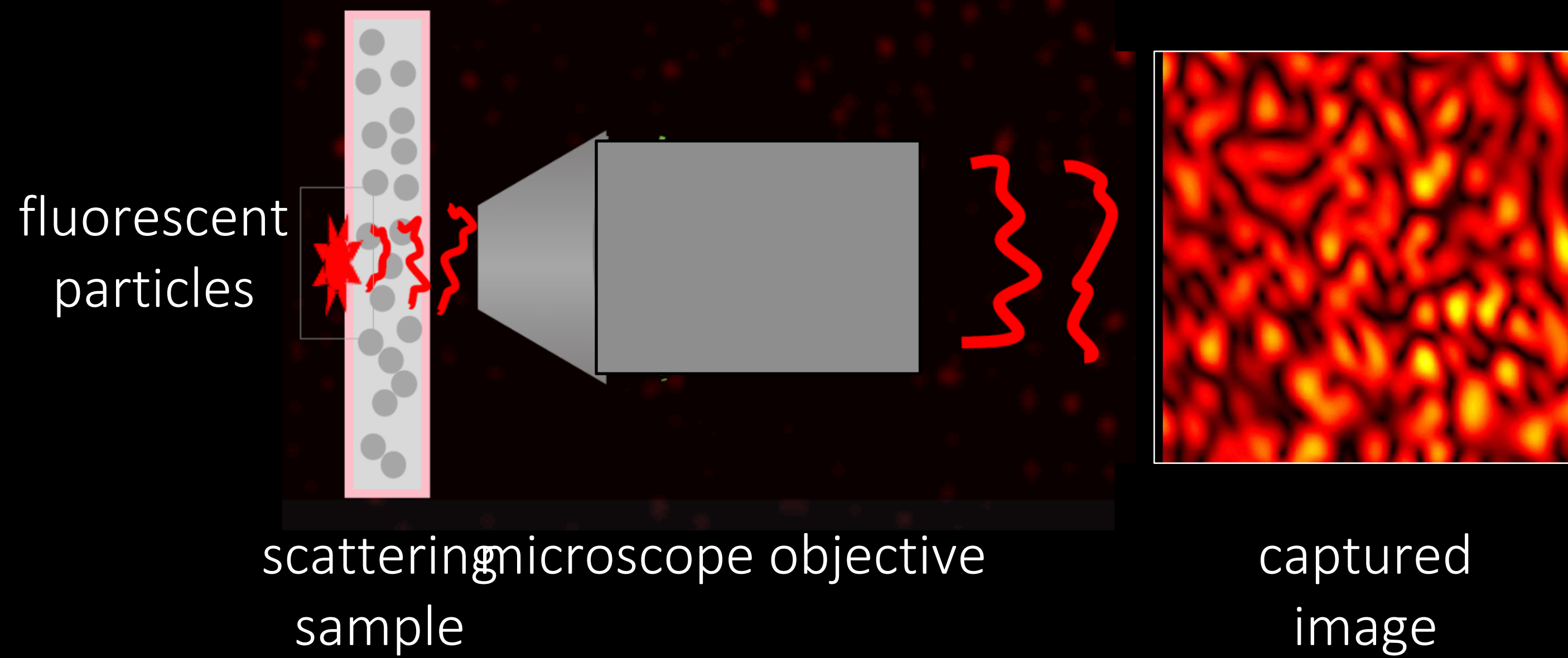


match wave equation solvers, 10^5 x faster

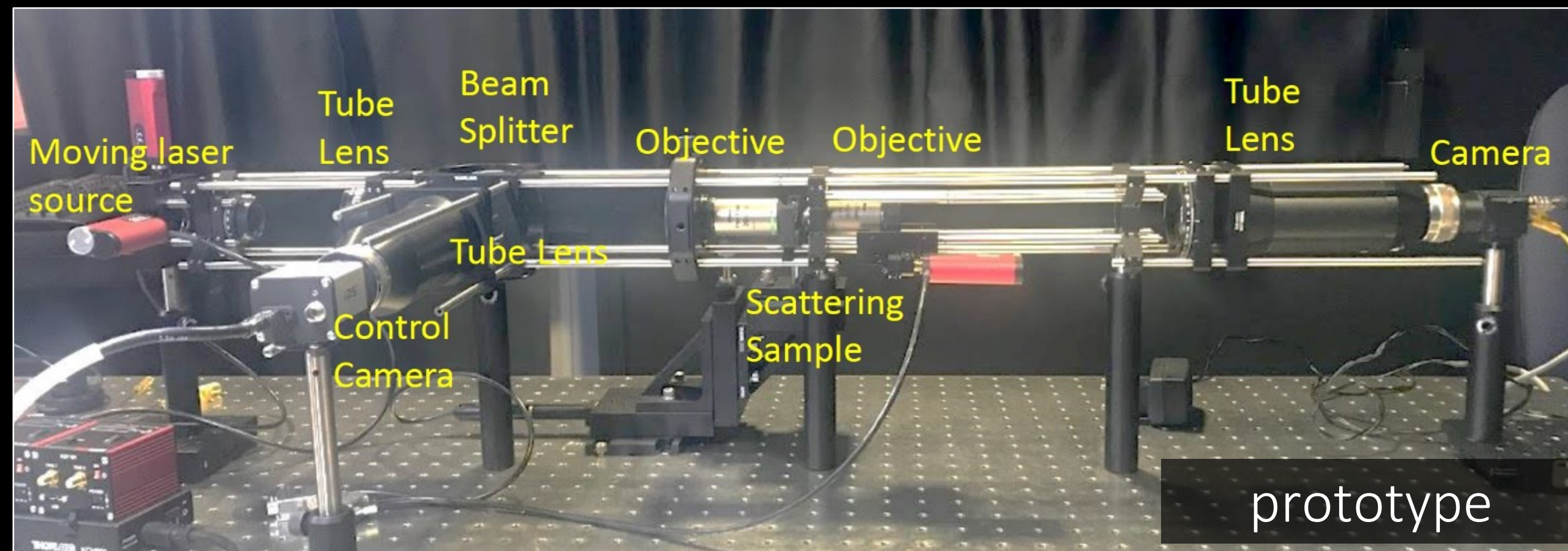
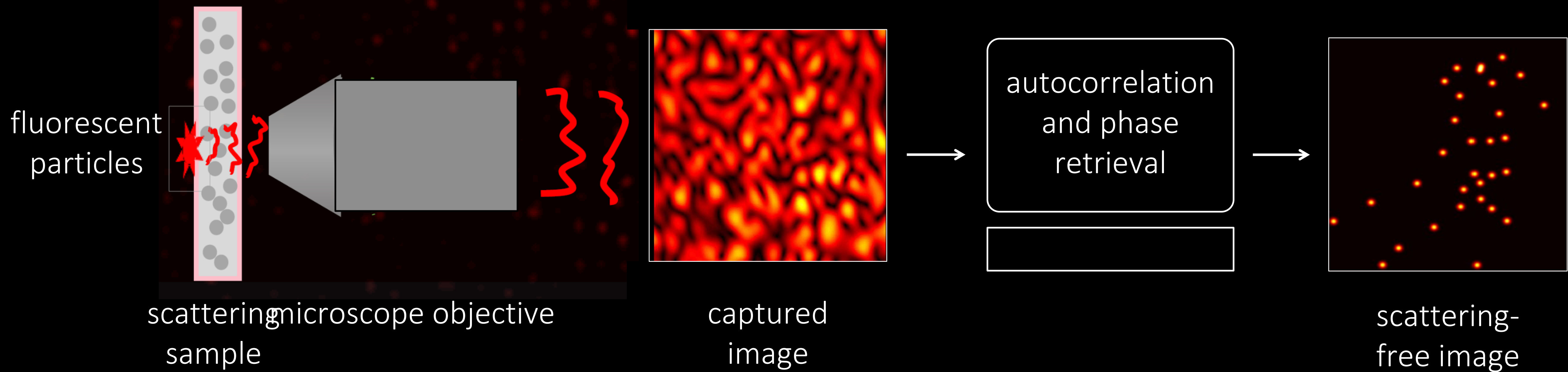


match real measurements of memory effect

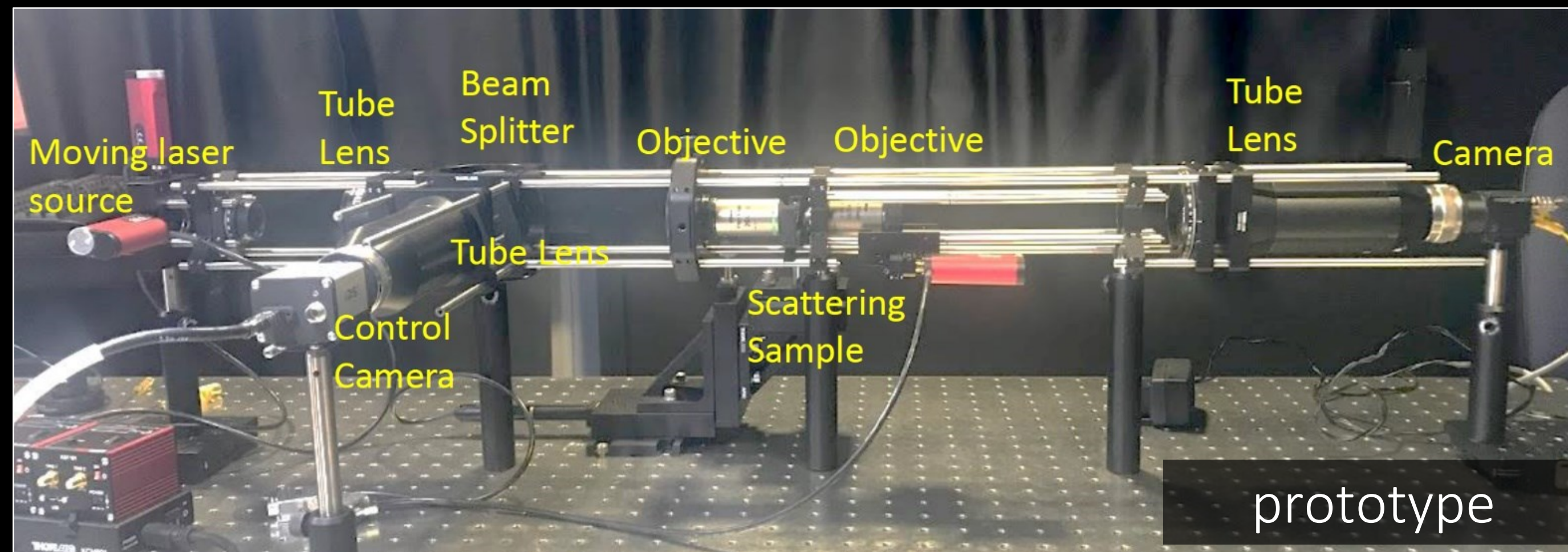
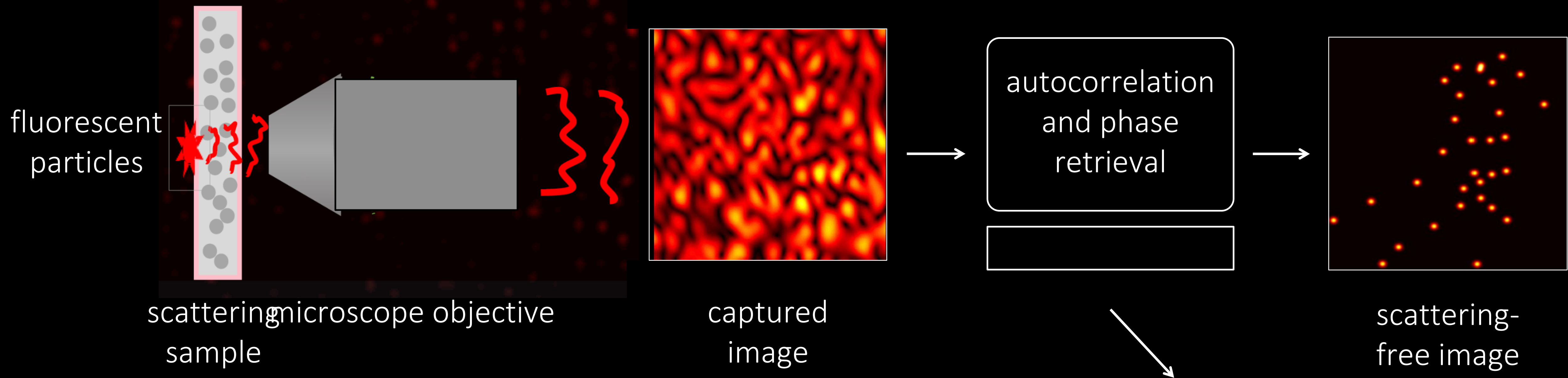
Speckle-based fluorescence microscopy



Speckle-based fluorescence microscopy



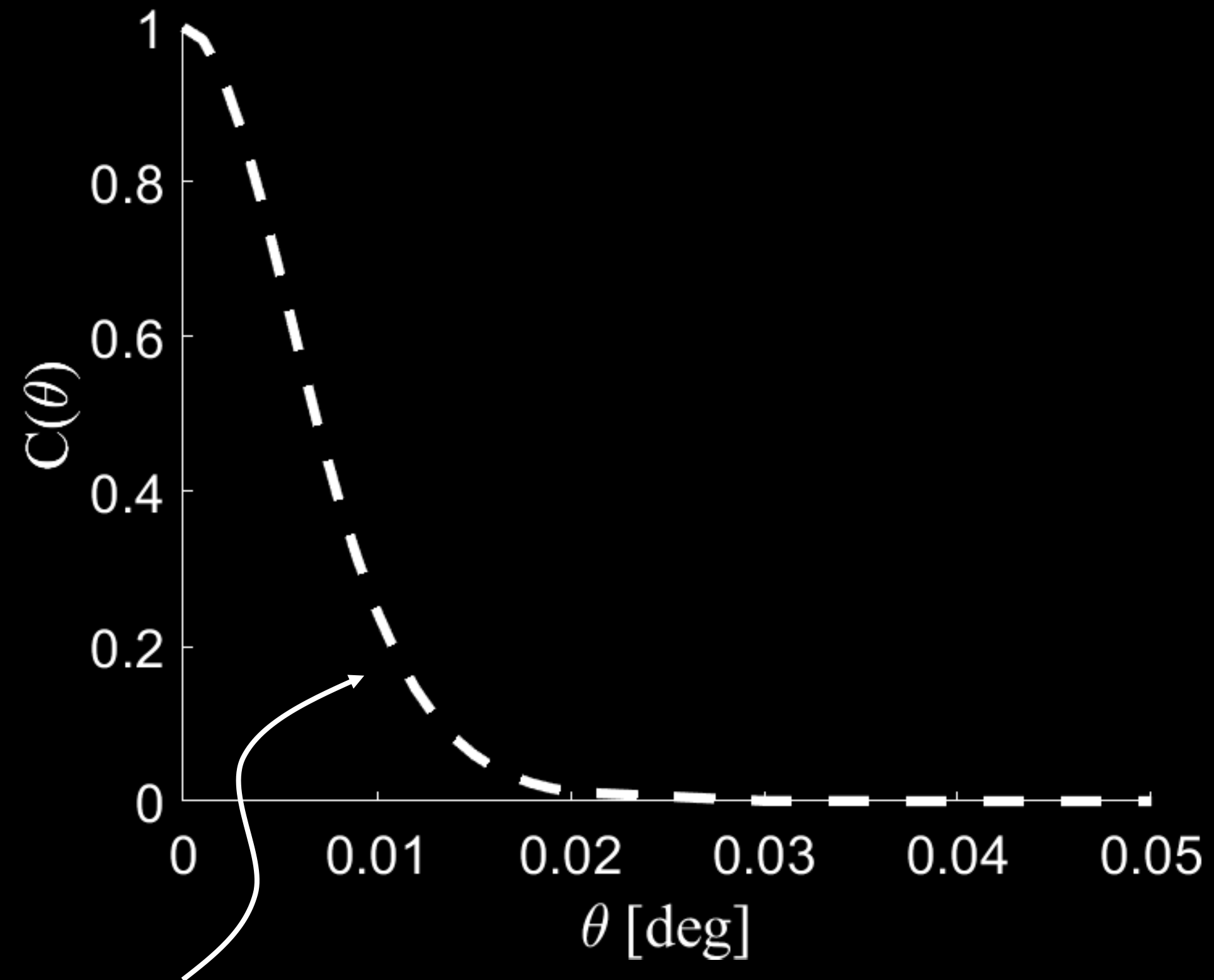
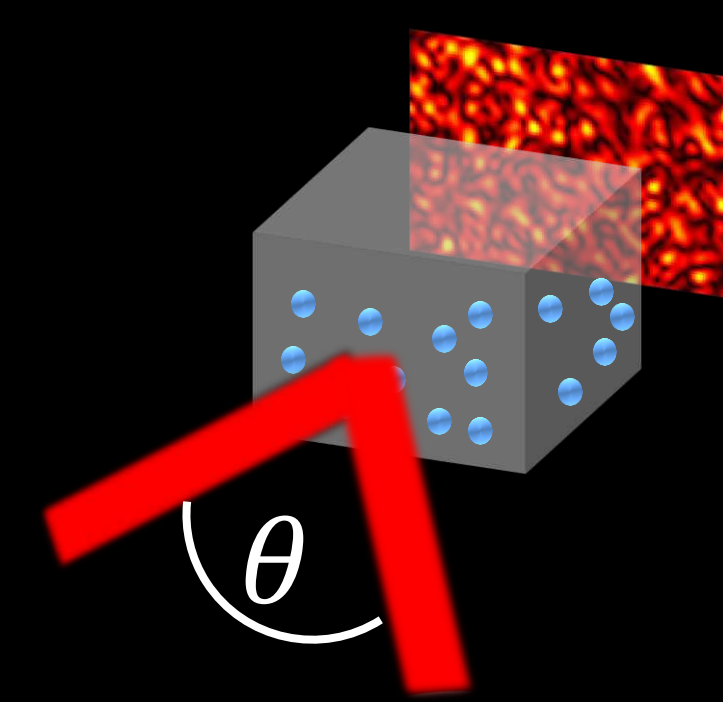
Speckle-based fluorescence microscopy



Performance strongly depends on:

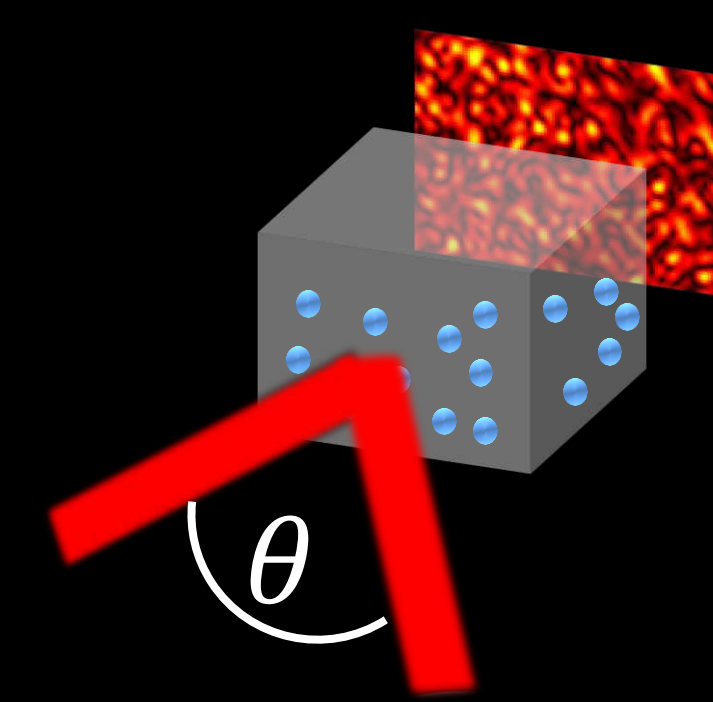
- speckle statistics
- image priors
- tissue parameters

Evaluate the memory effect

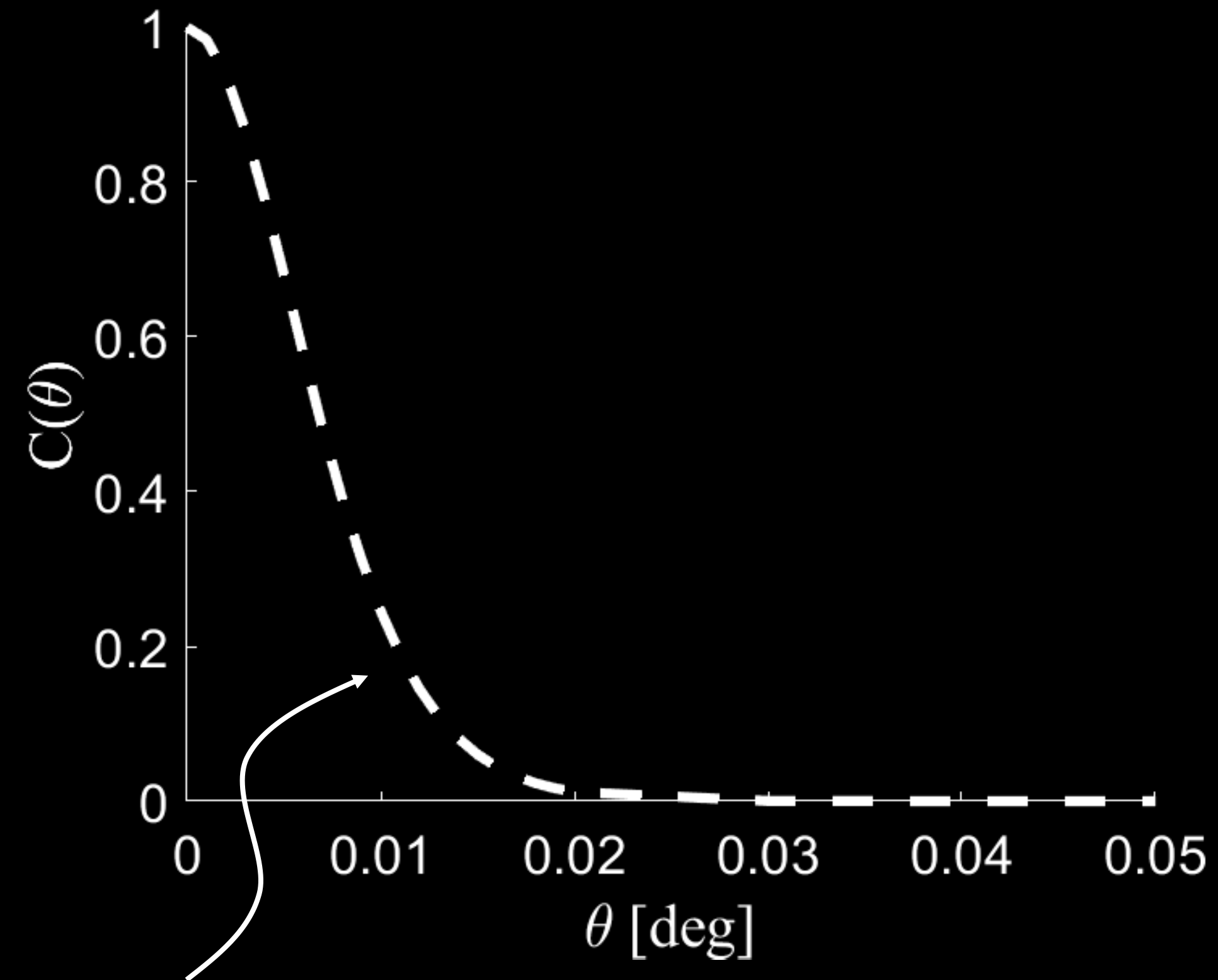


Analytical solution
based on diffusion
approximation

Evaluate the memory effect

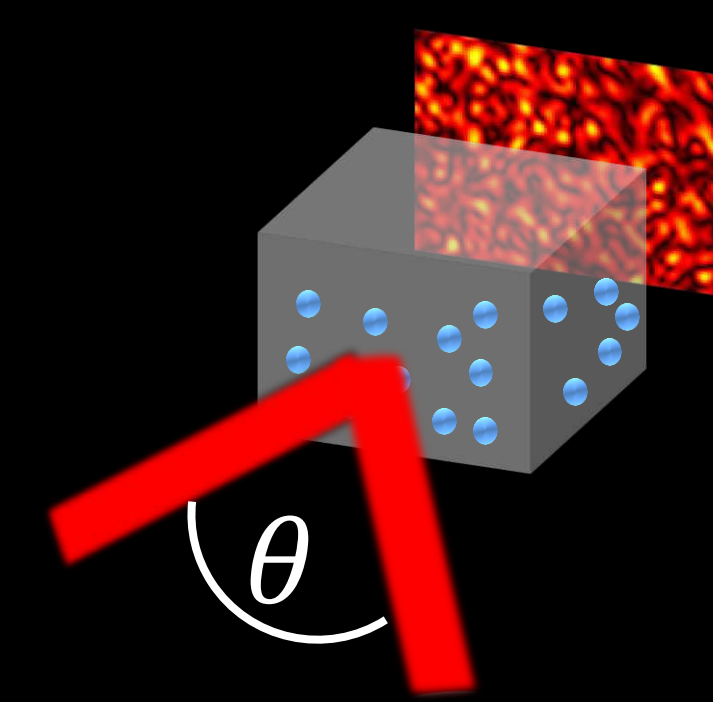


This approximation is correct for some materials and configurations



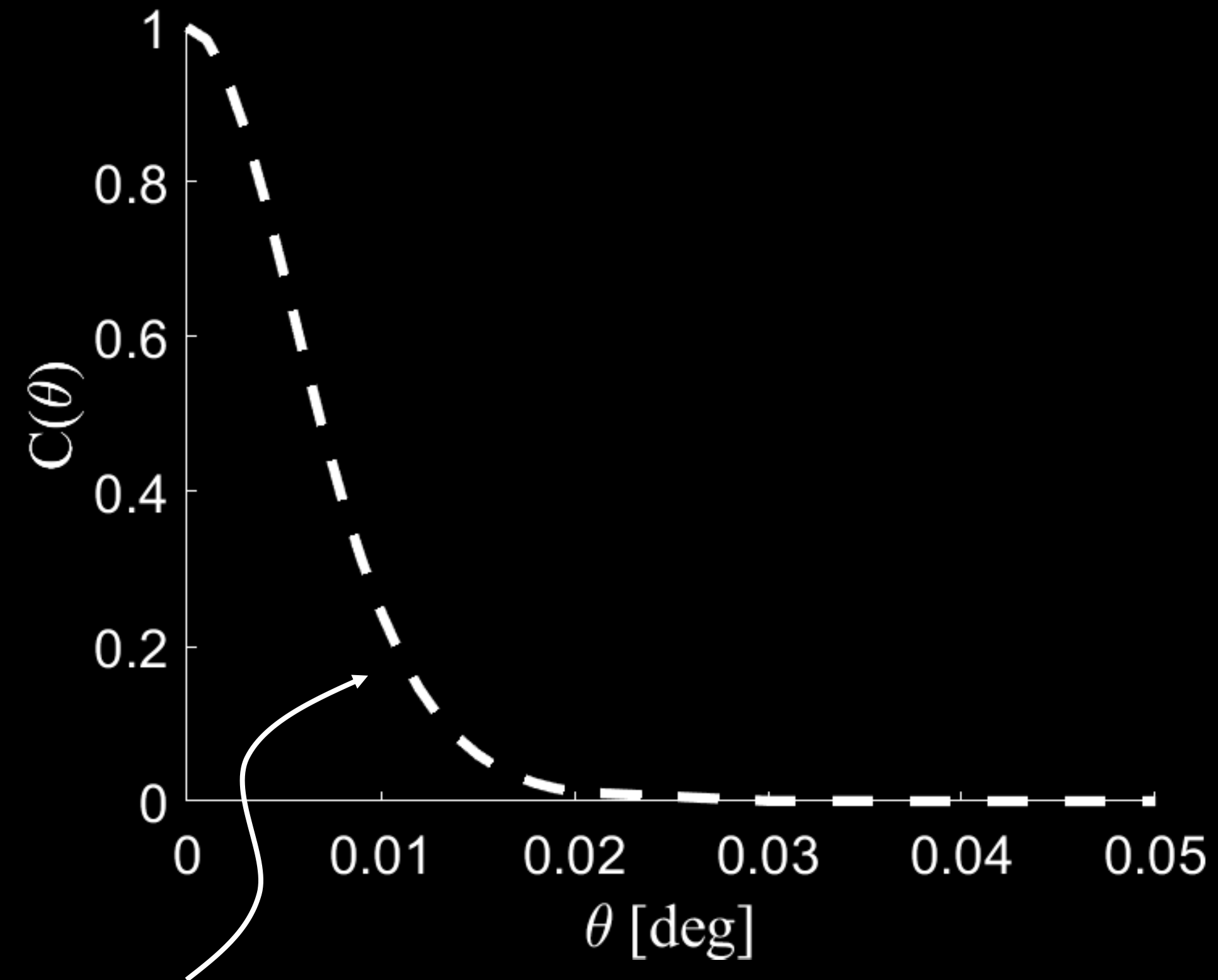
Analytical solution
based on diffusion
approximation

Evaluate the memory effect



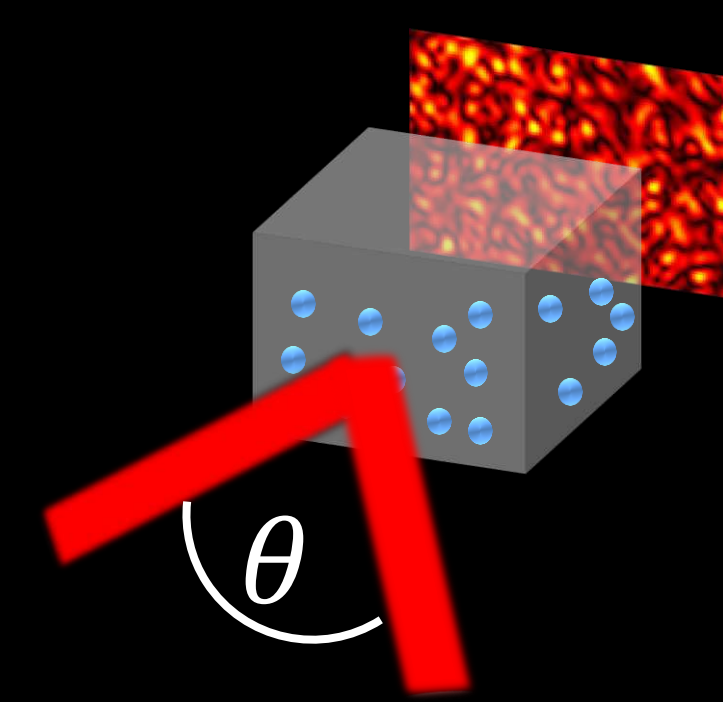
This approximation is correct for some materials and configurations

In practice ME extent is often wider



Analytical solution
based on diffusion
approximation

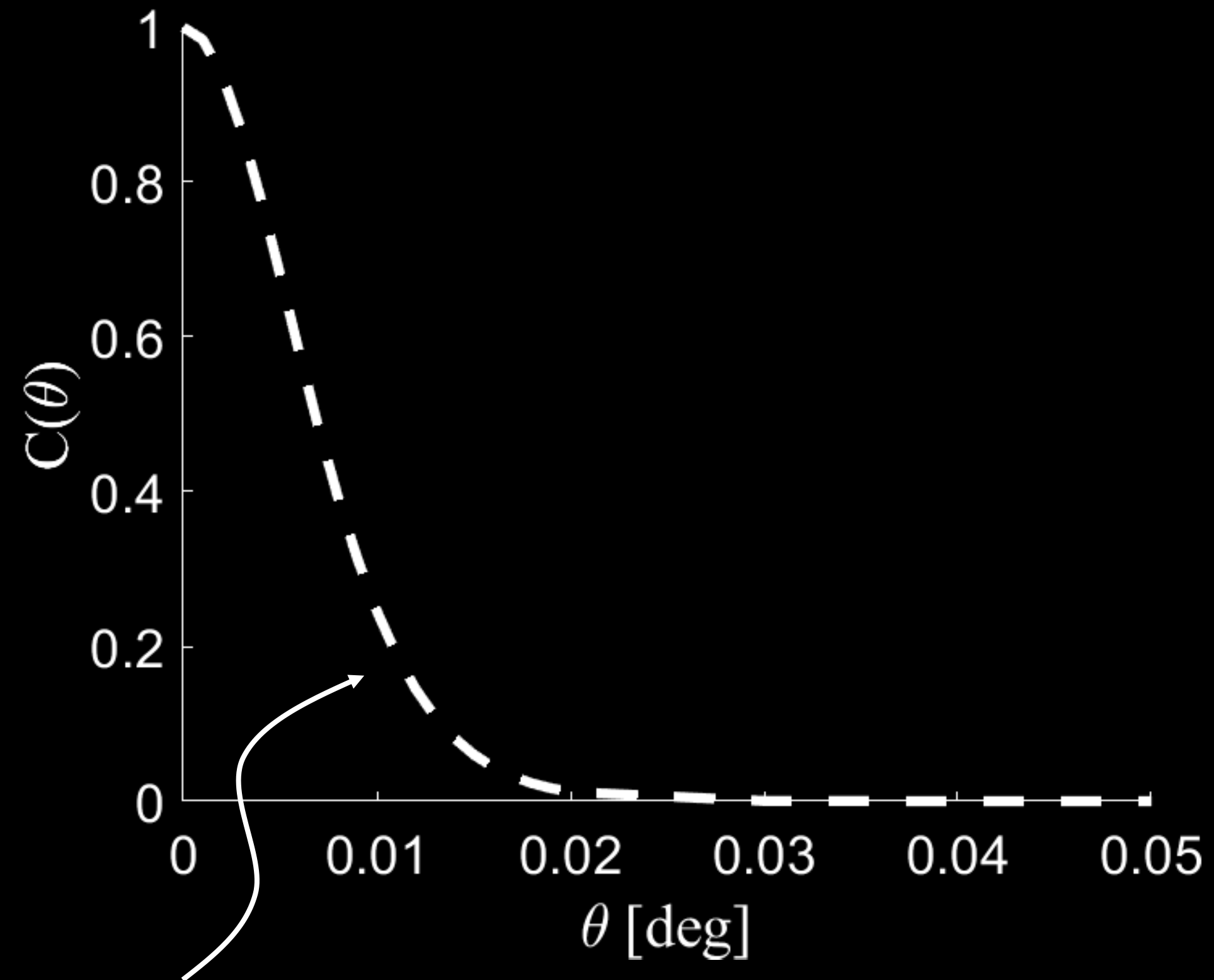
Evaluate the memory effect



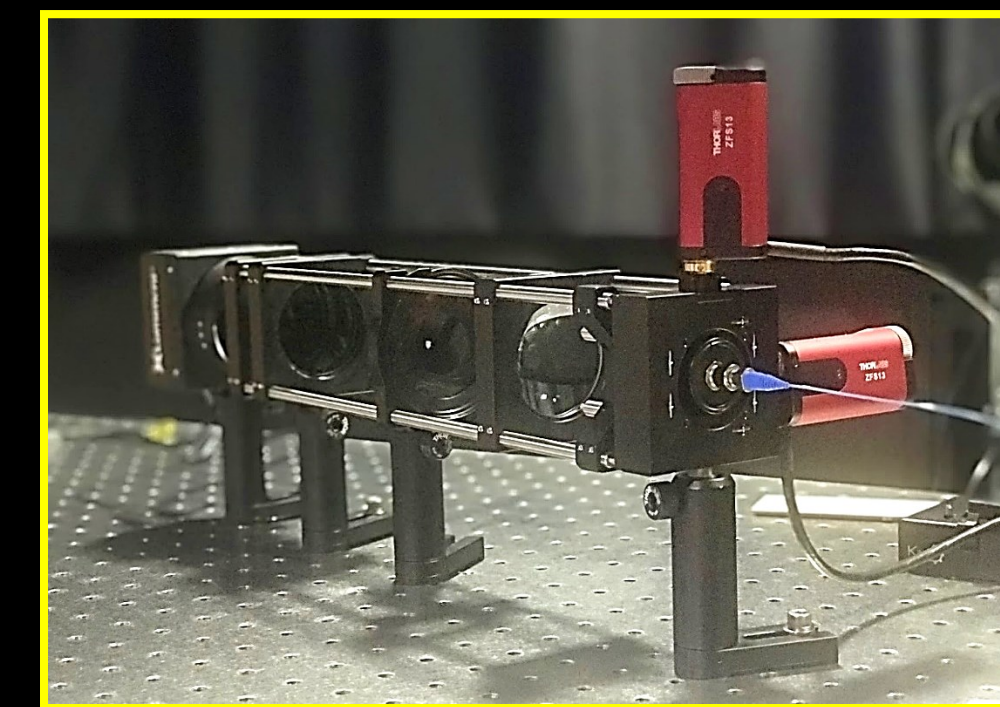
This approximation is correct for some materials and configurations

In practice ME extent is often wider

Currently: measured *empirically* in the lab

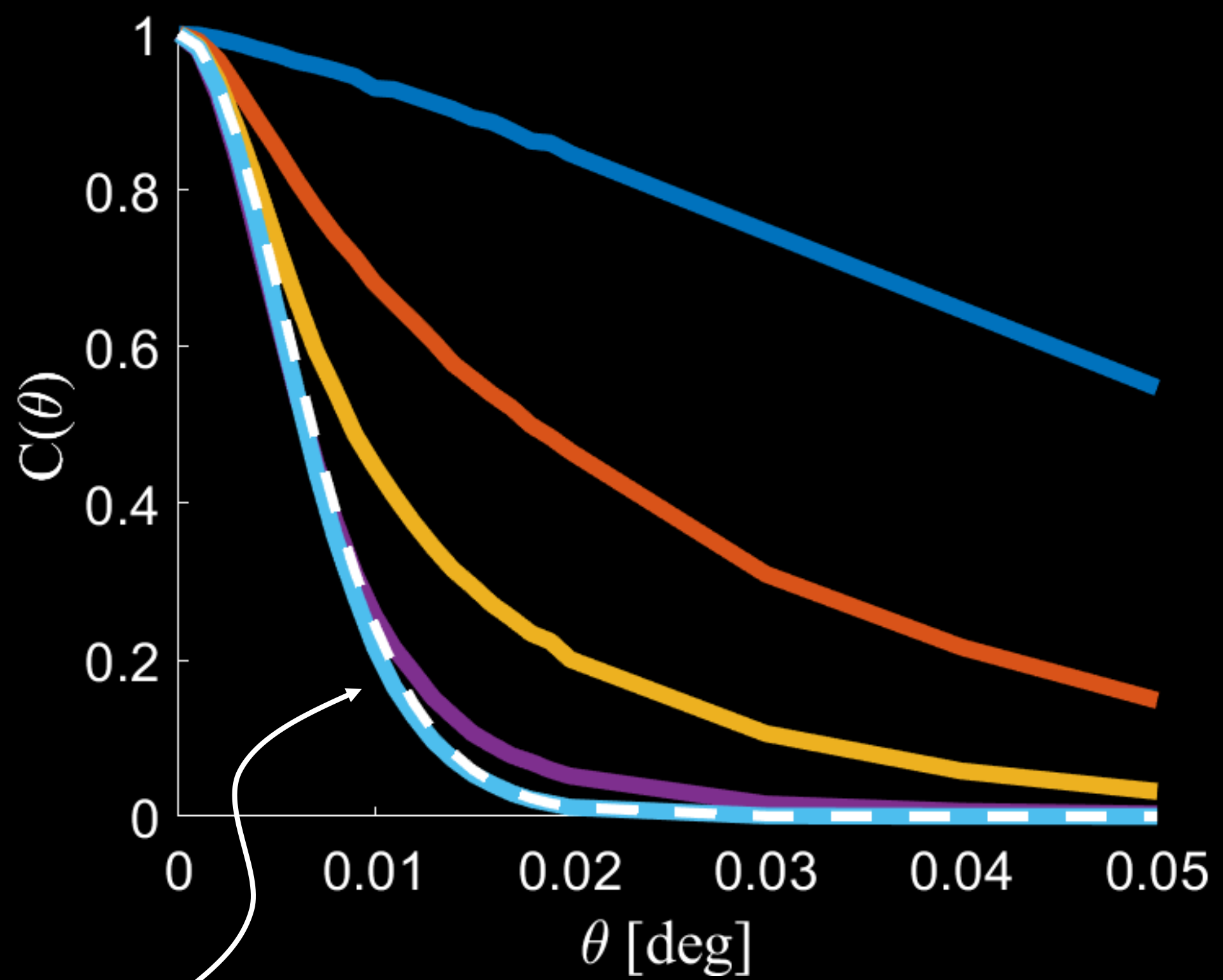


Analytical solution
based on diffusion
approximation

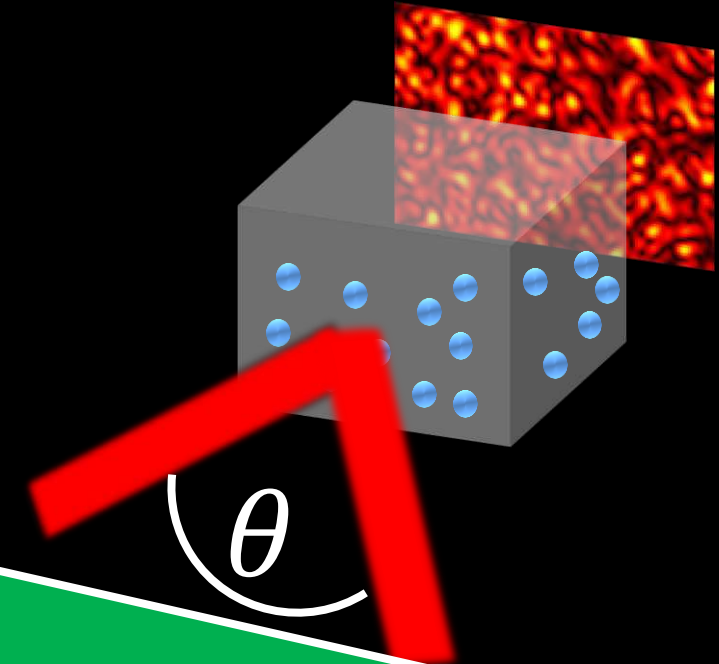
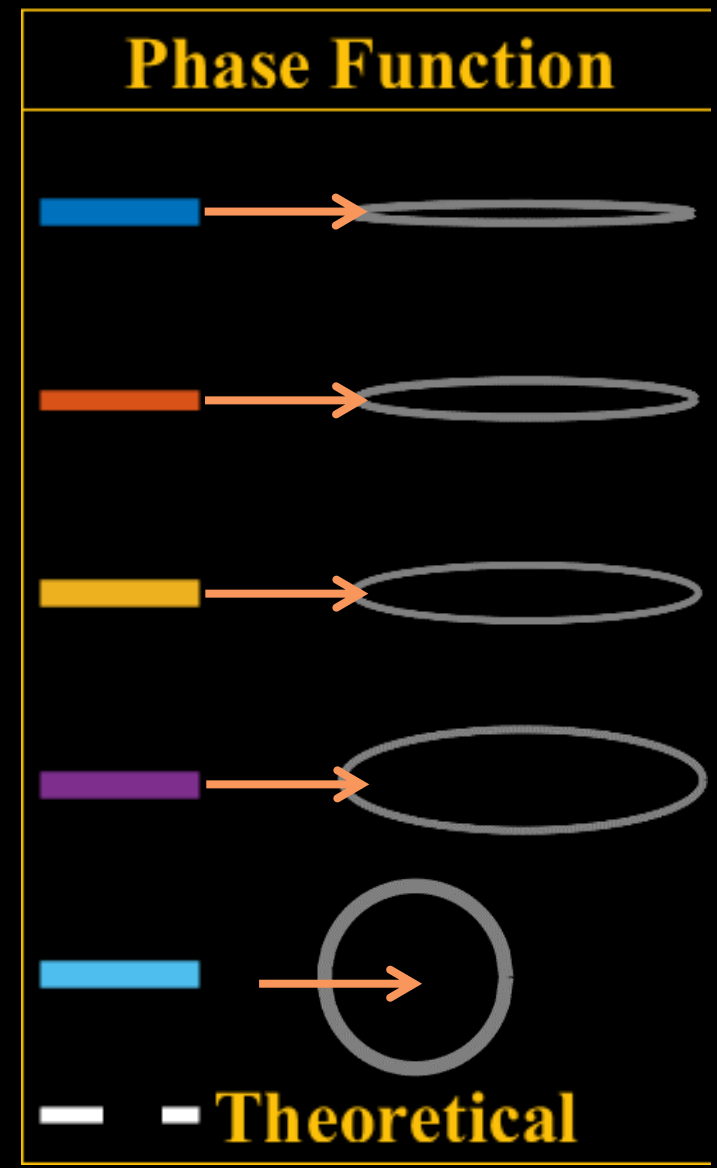


[Alterman et al., 2021]

Evaluate the memory effect



Analytical solution based on diffusion approximation

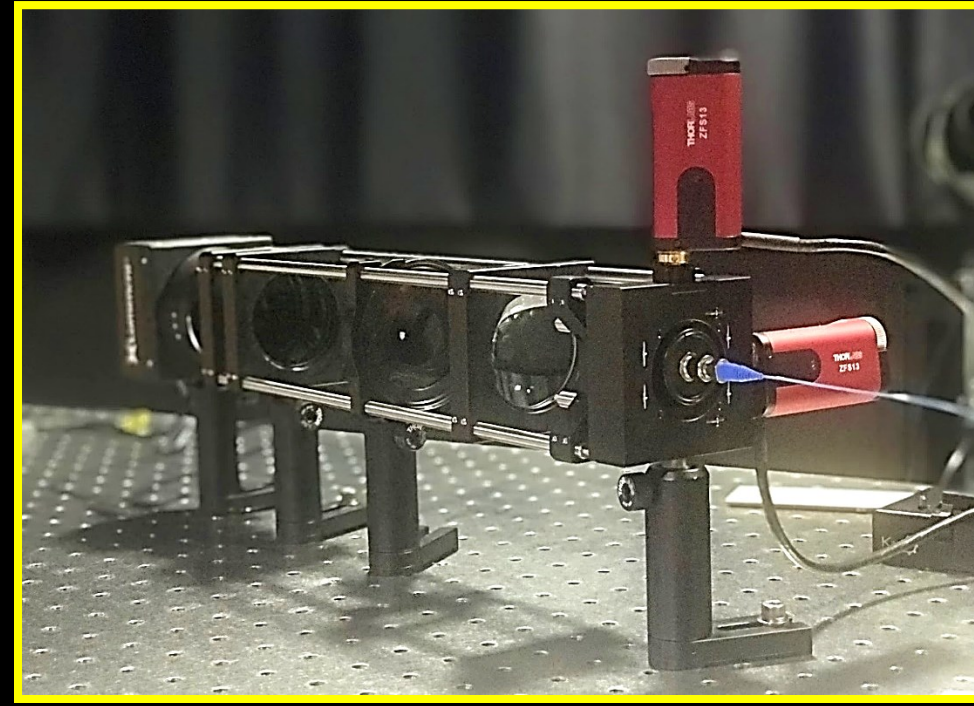


Now we can efficiently compute ME curves for all scattering parameters

This approach

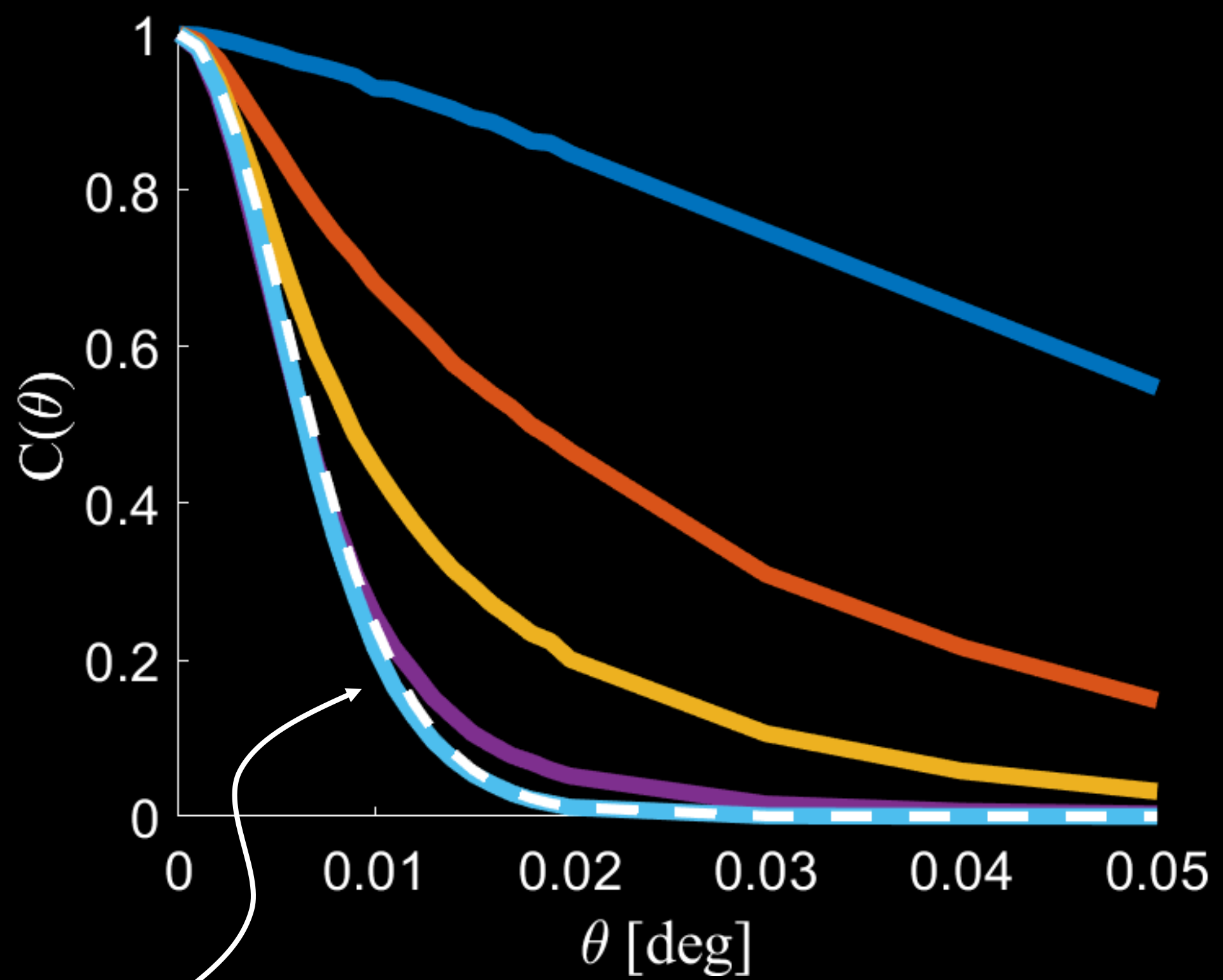
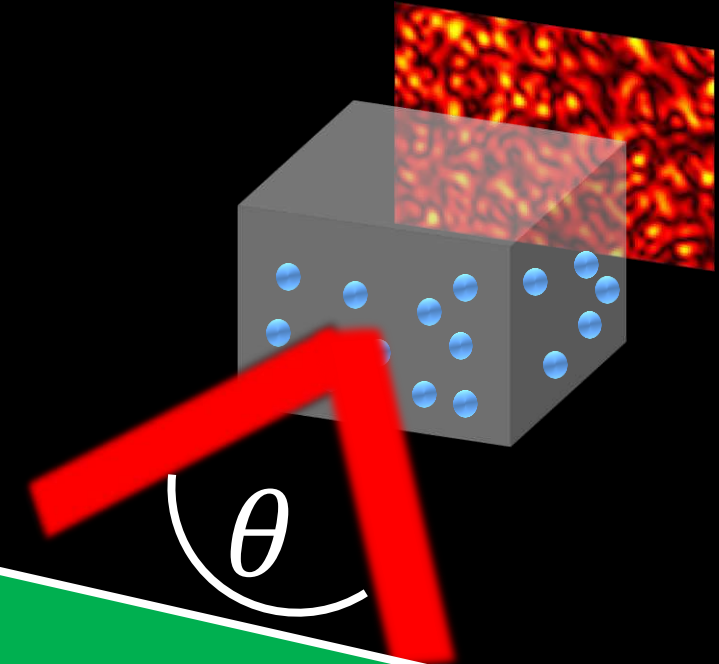
In practice ME experiments

Currently: measured *empirically* in the lab

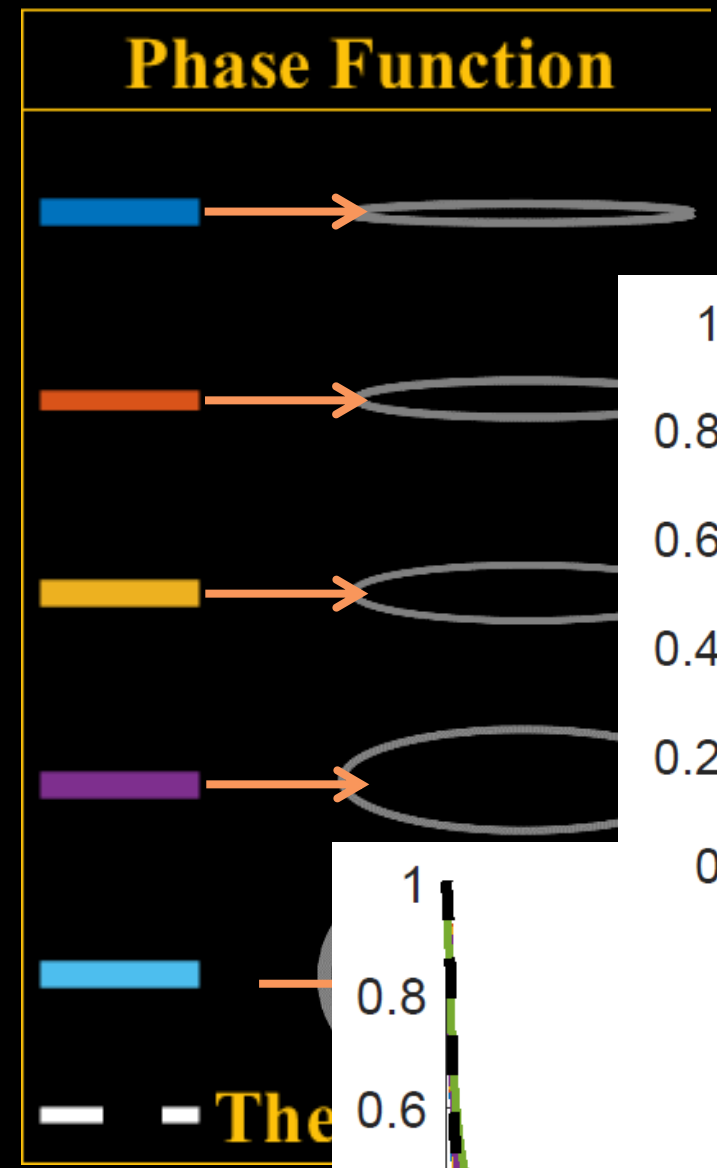


[Alterman et al., 2021]

Evaluate the memory effect

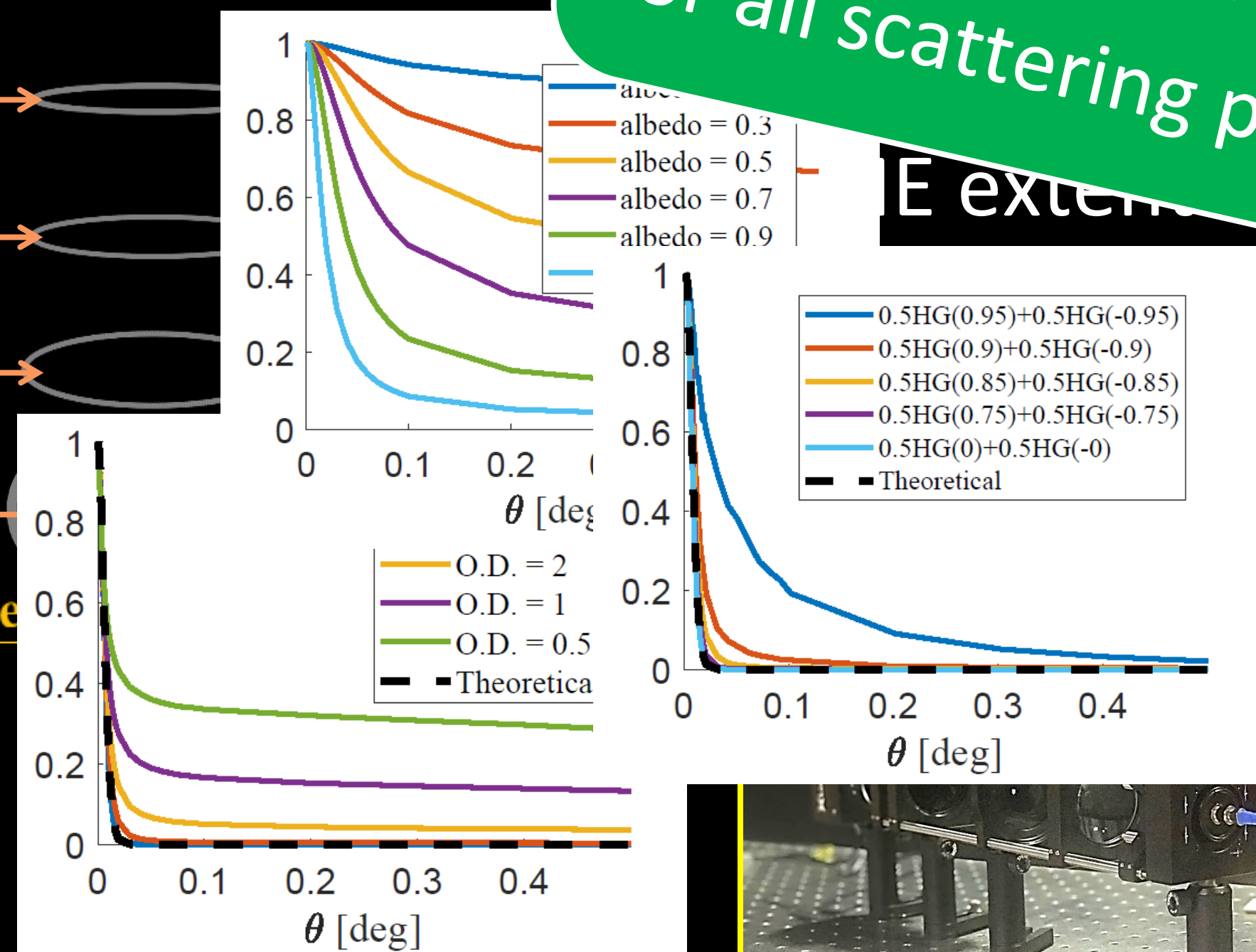


Analytical solution based on diffusion approximation

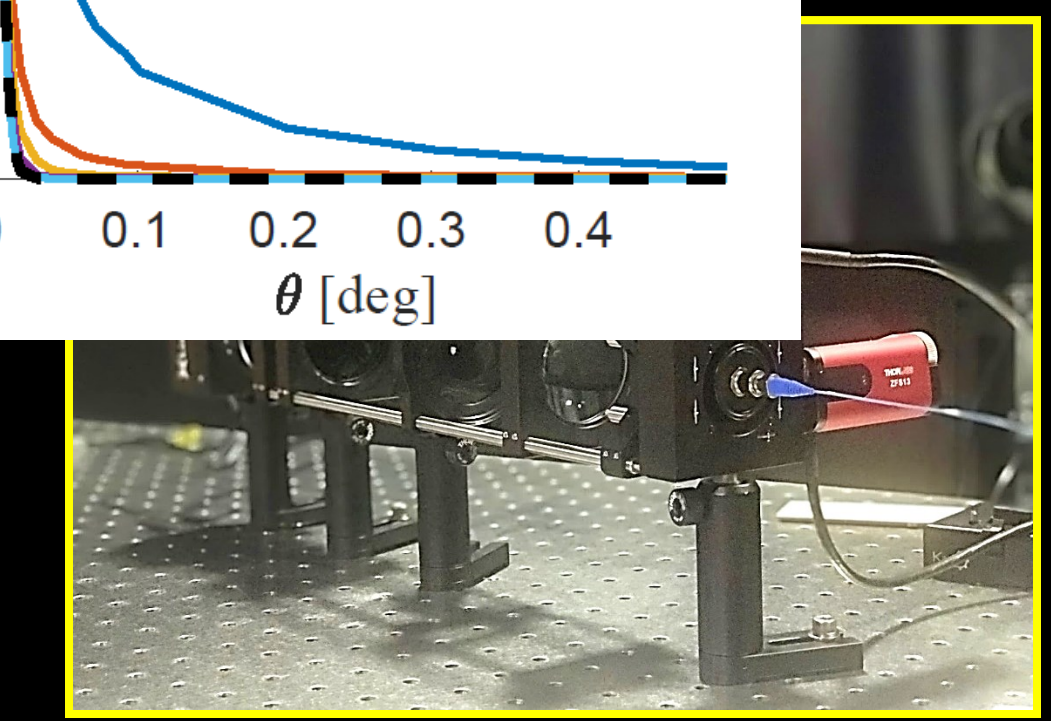


This approach

Now we can efficiently compute ME curves for all scattering parameters



empirically in the



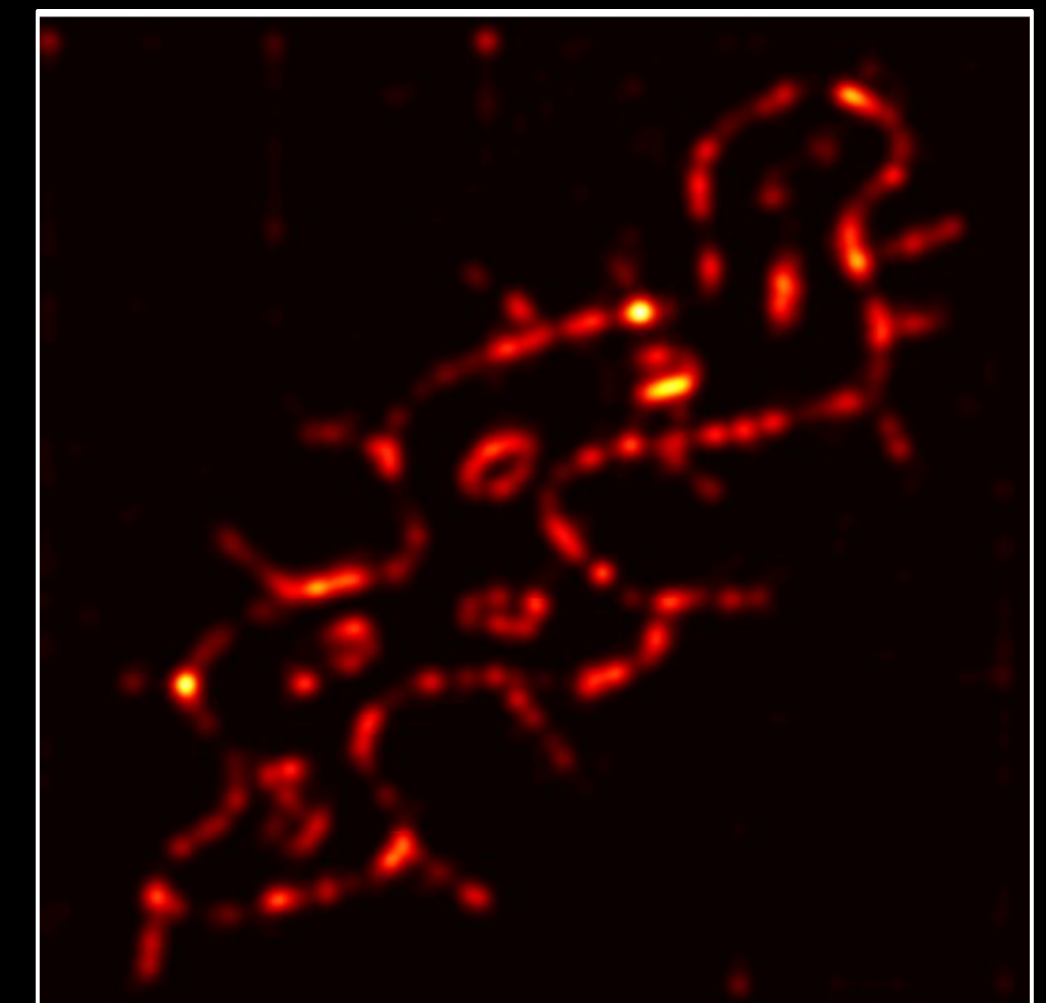
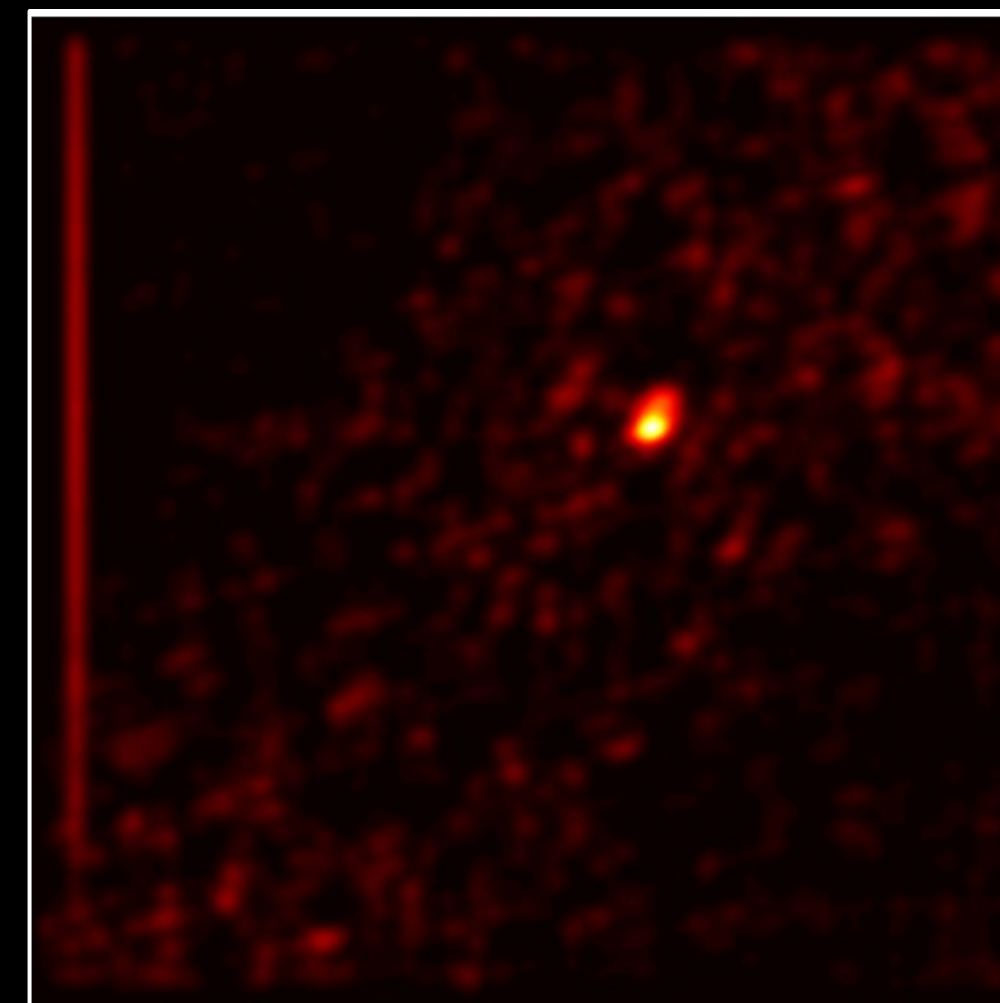
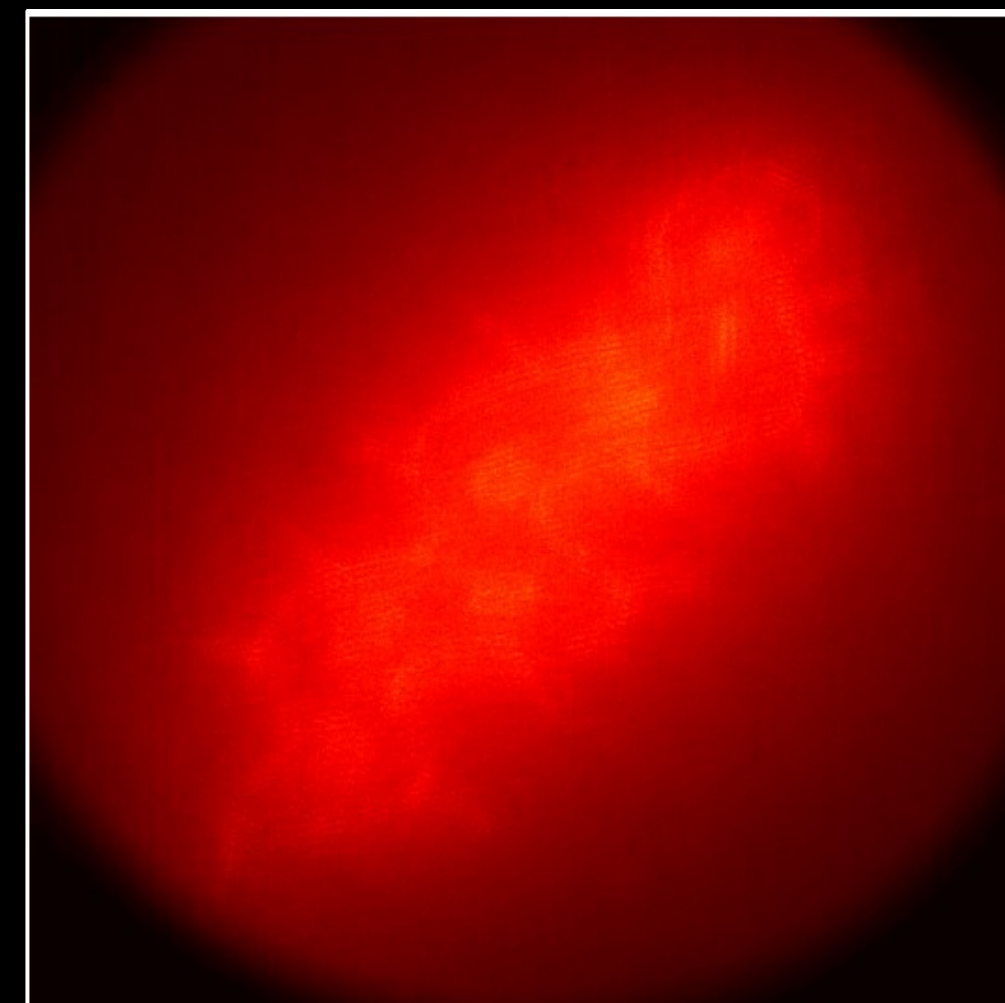
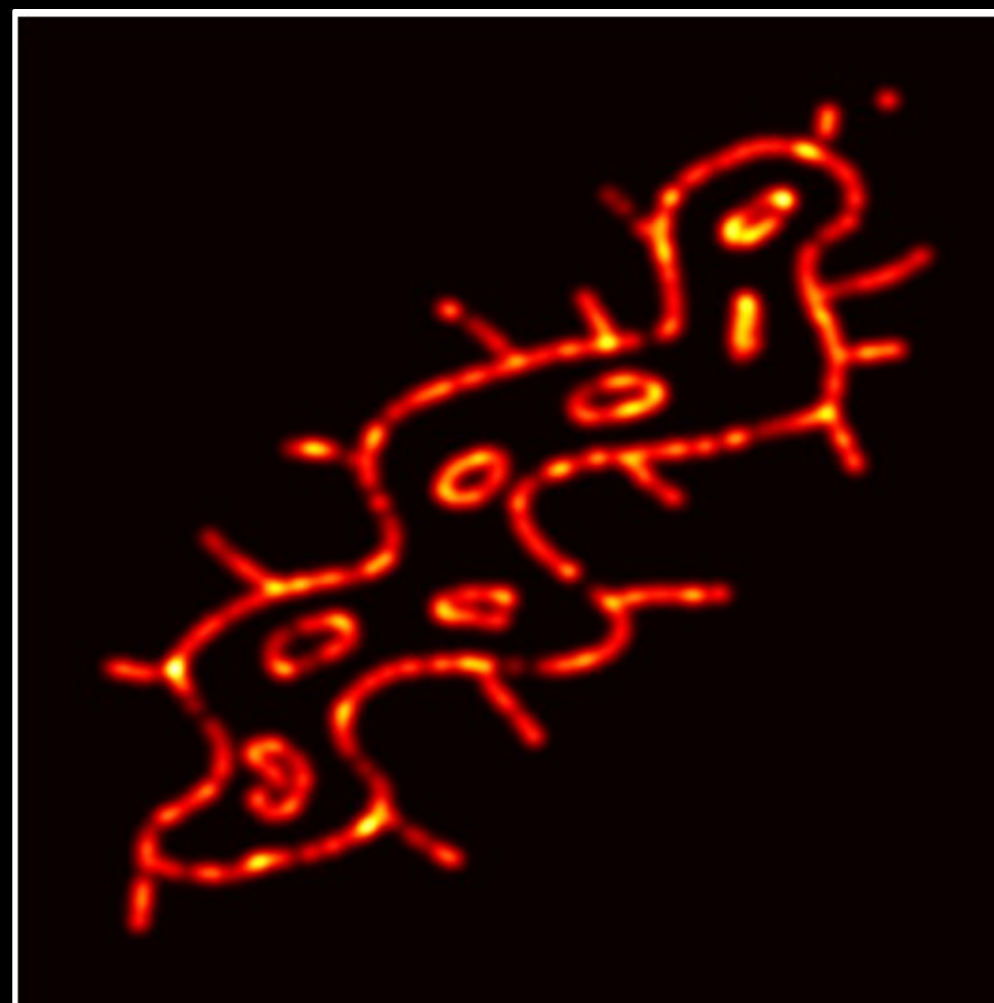
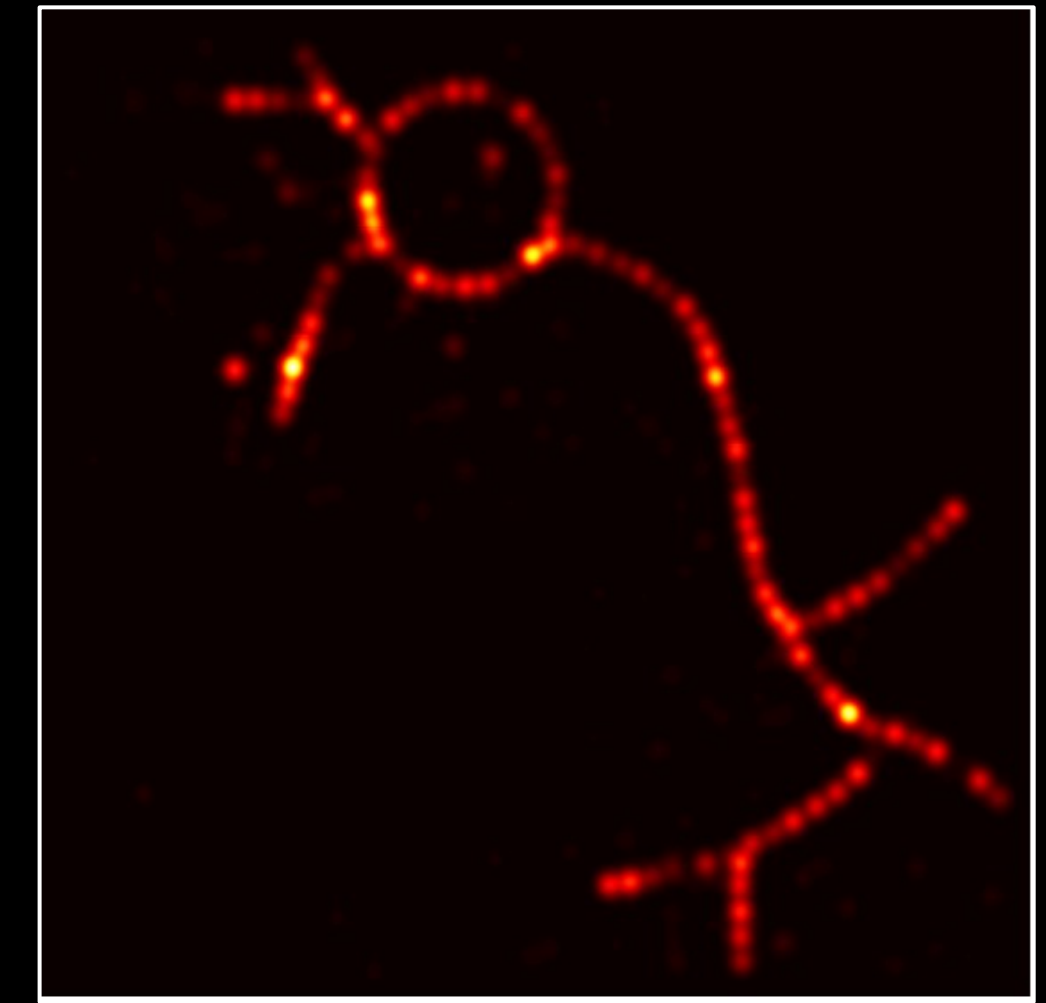
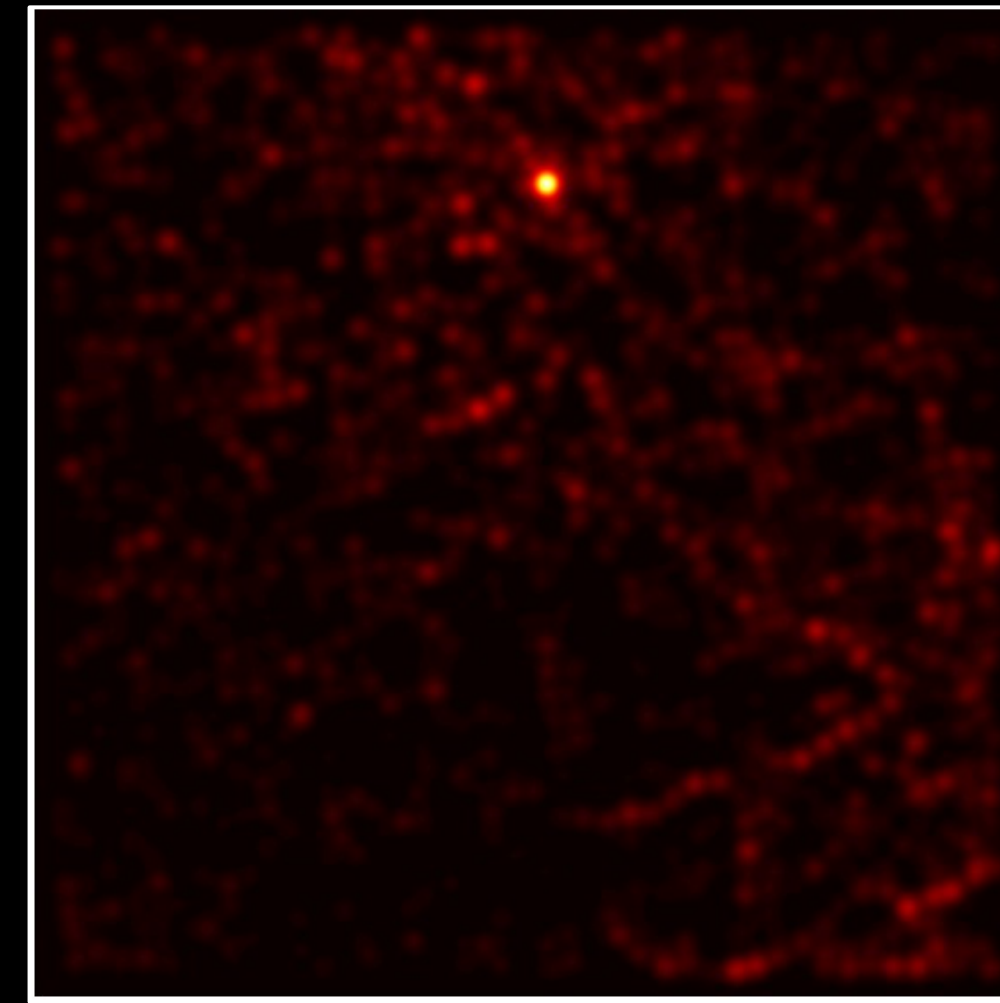
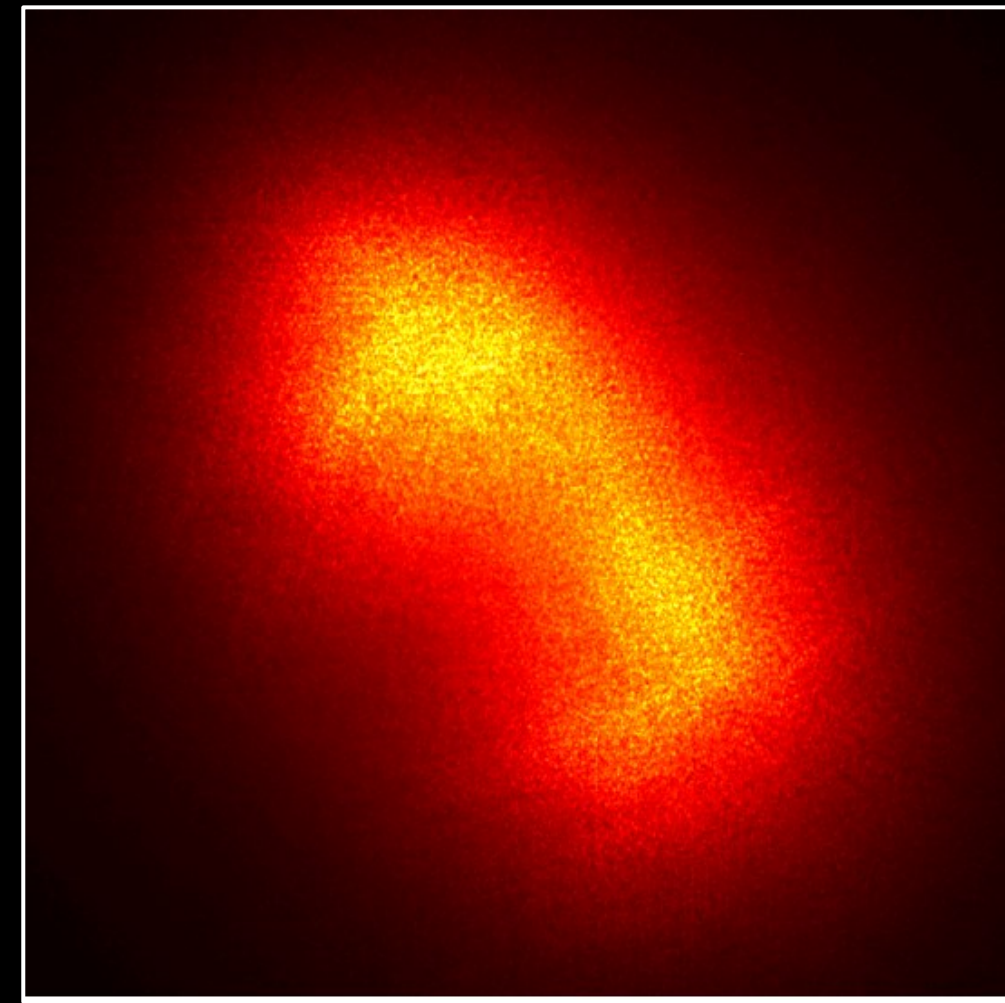
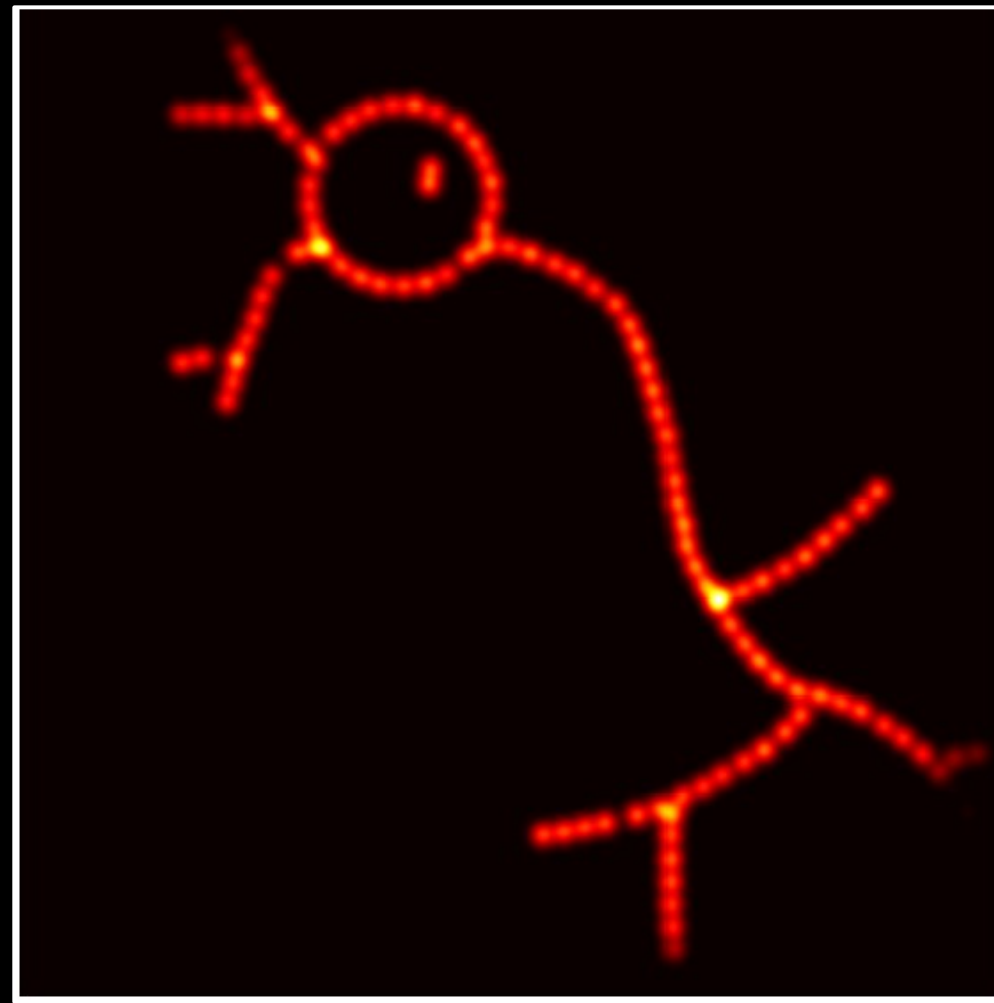
Better algorithms for fluorescence microscopy

groundtruth

input image

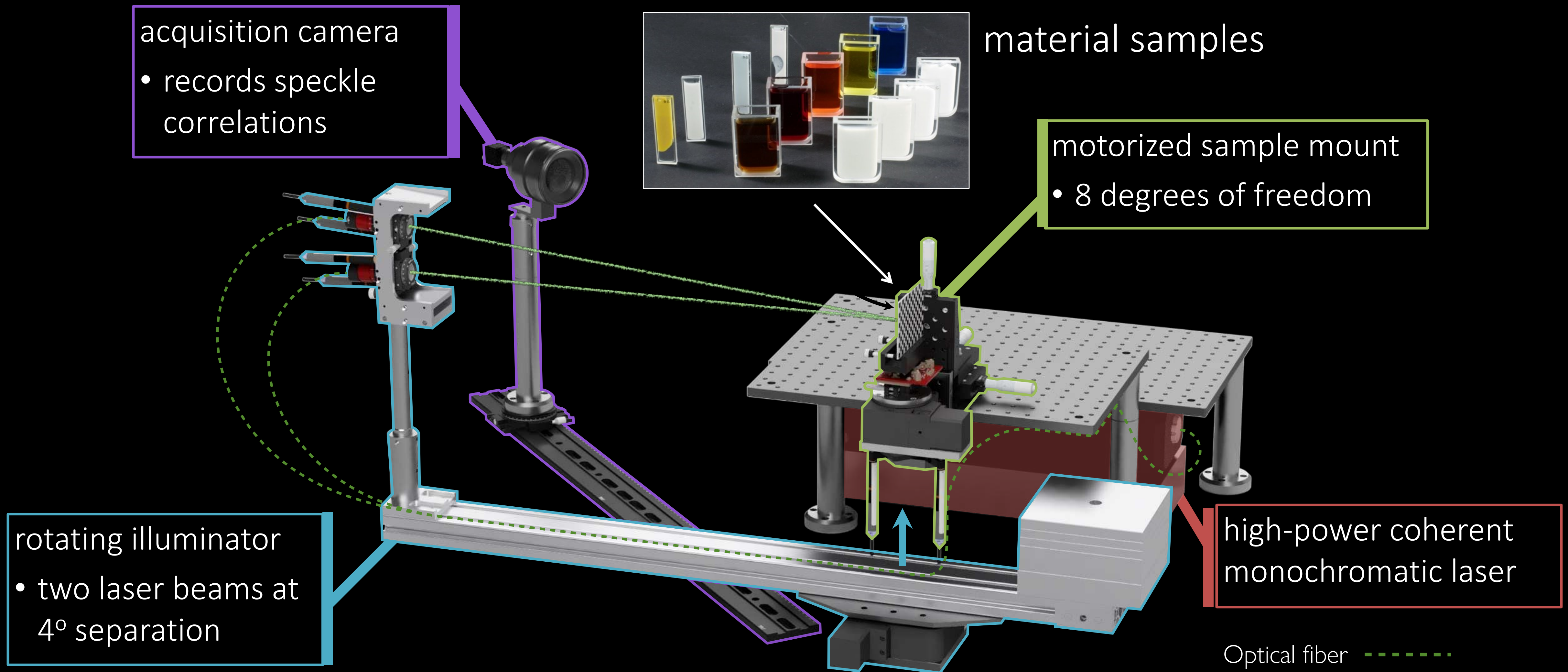
prior algorithm

our algorithm



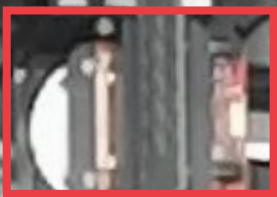
Acquisition of scattering materials

Use differentiable speckle rendering to recover material parameters from speckle images

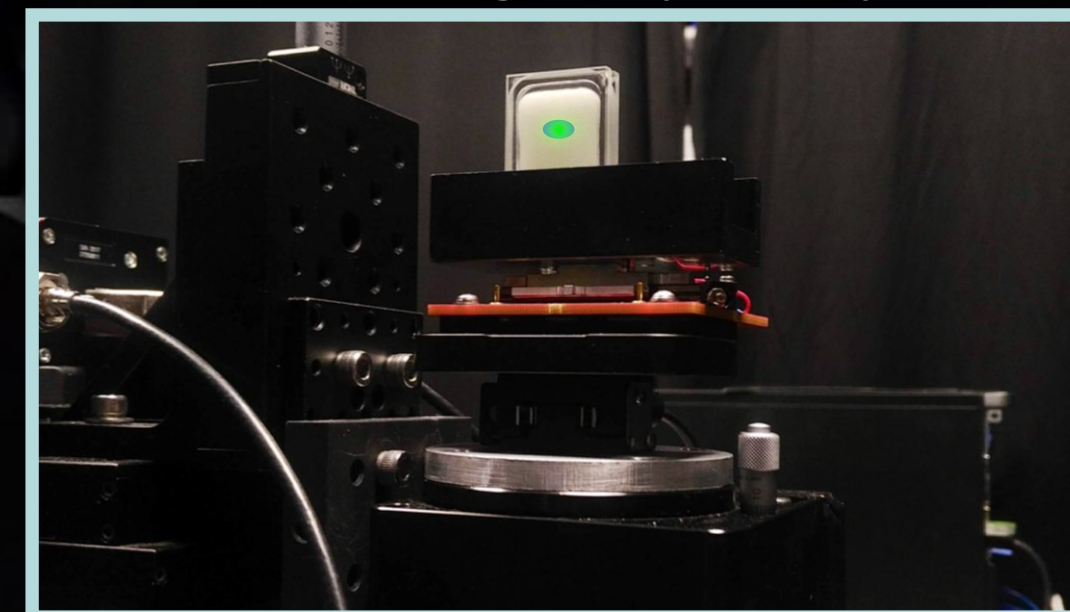


Acquisition

Speckle Video
(Camera Feed)



Scattering Soap Sample



Rendering wave optics is a very active area

A Generic Framework for Physical Light Transport

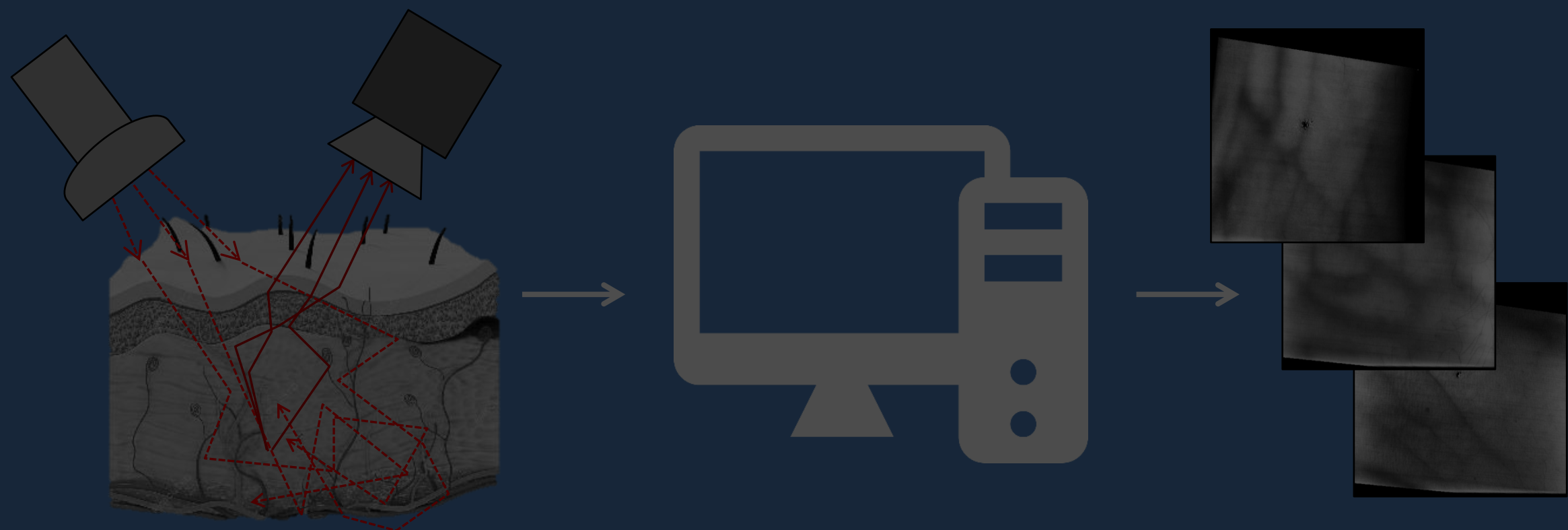
SHLOMI STEINBERG, University of California, Santa Barbara, USA

LING-QI YAN, University of California, Santa Barbara, USA



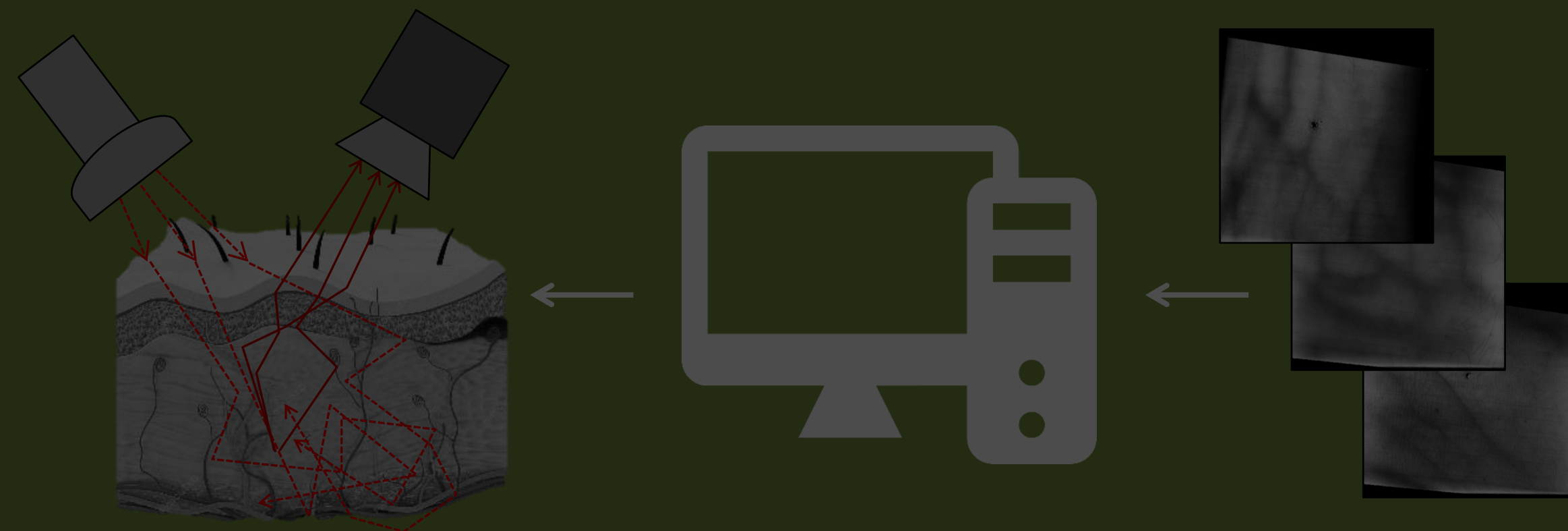
Physics-based rendering and its applications to computational imaging

forward rendering

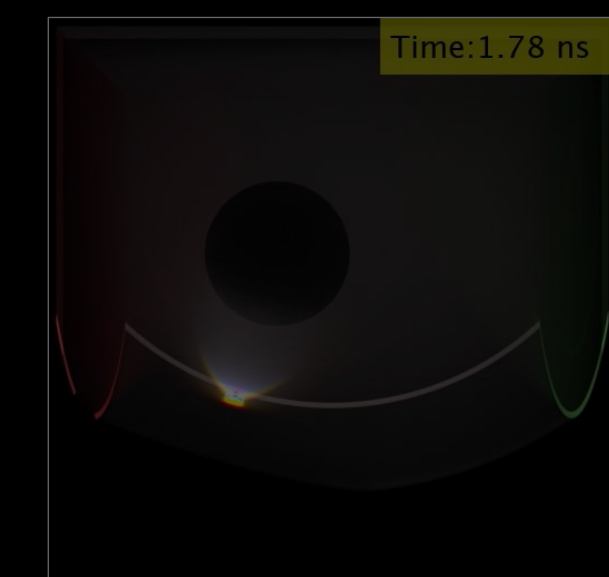


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

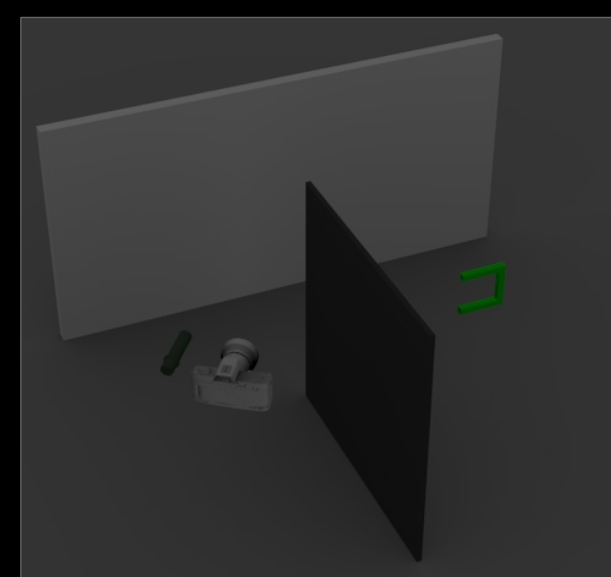
inverse rendering



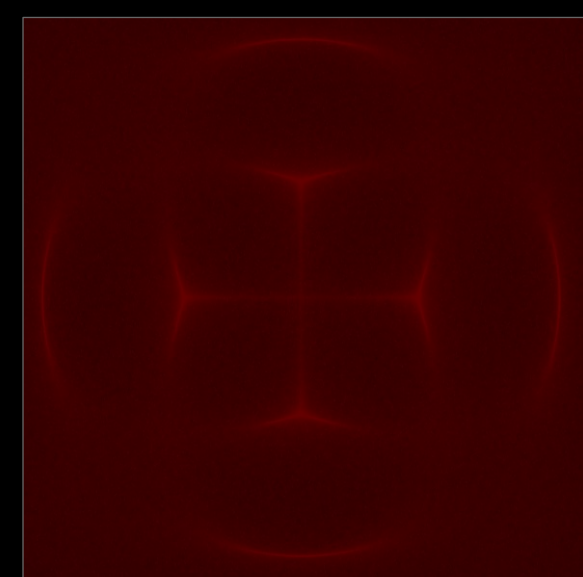
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



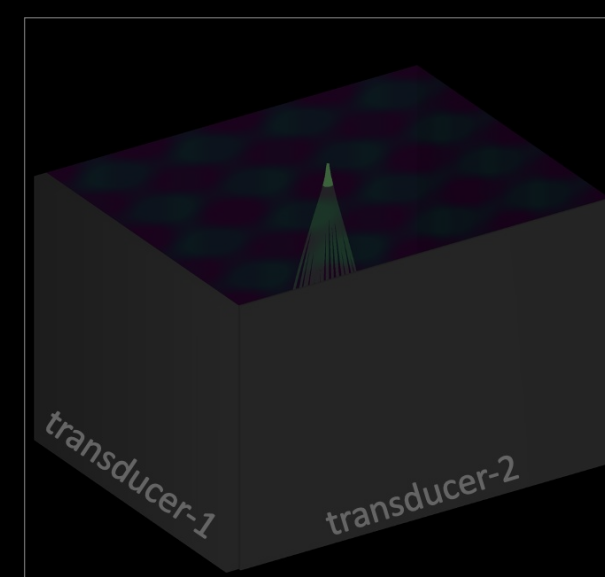
time-of-flight imaging



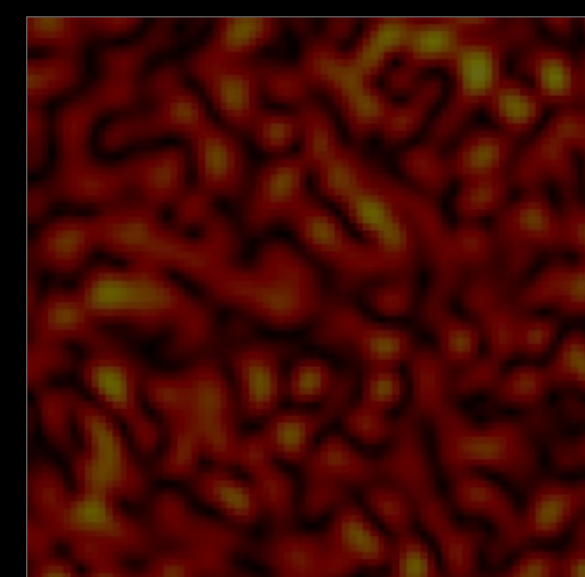
non-line-of-sight imaging



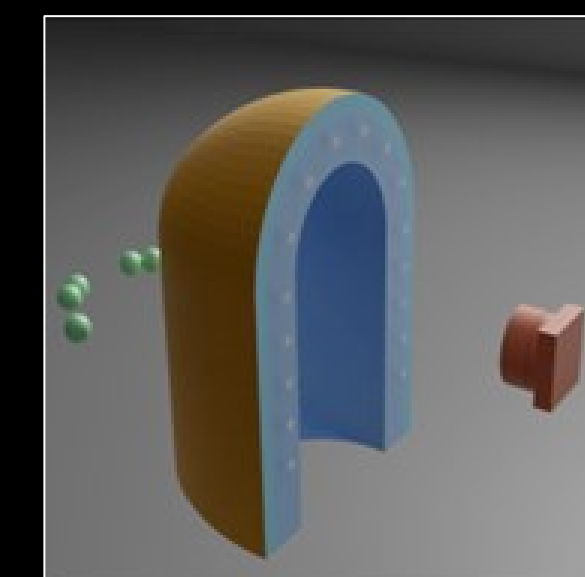
acousto-optic lensing



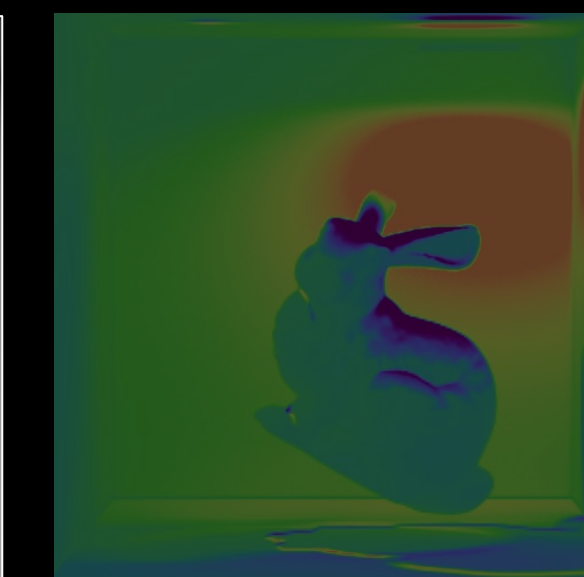
ultrafast light scanning



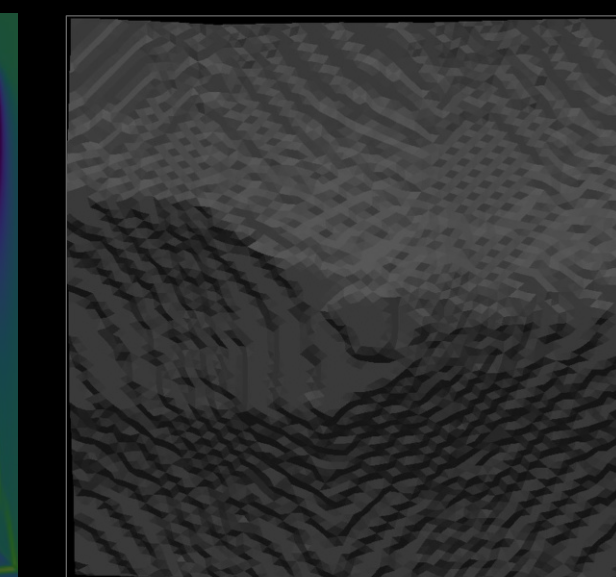
speckle imaging



tactile sensor design

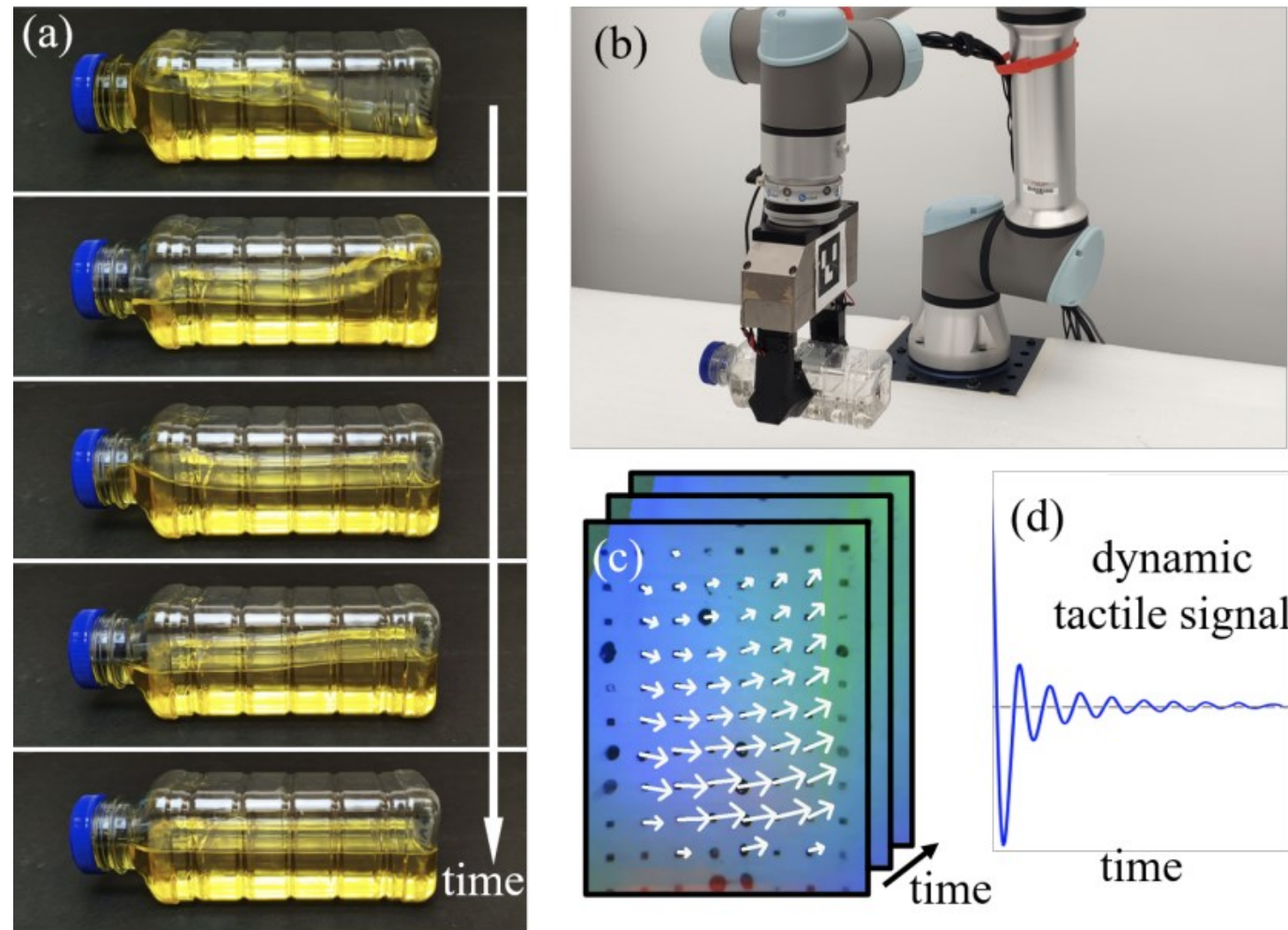


differentiable renderer

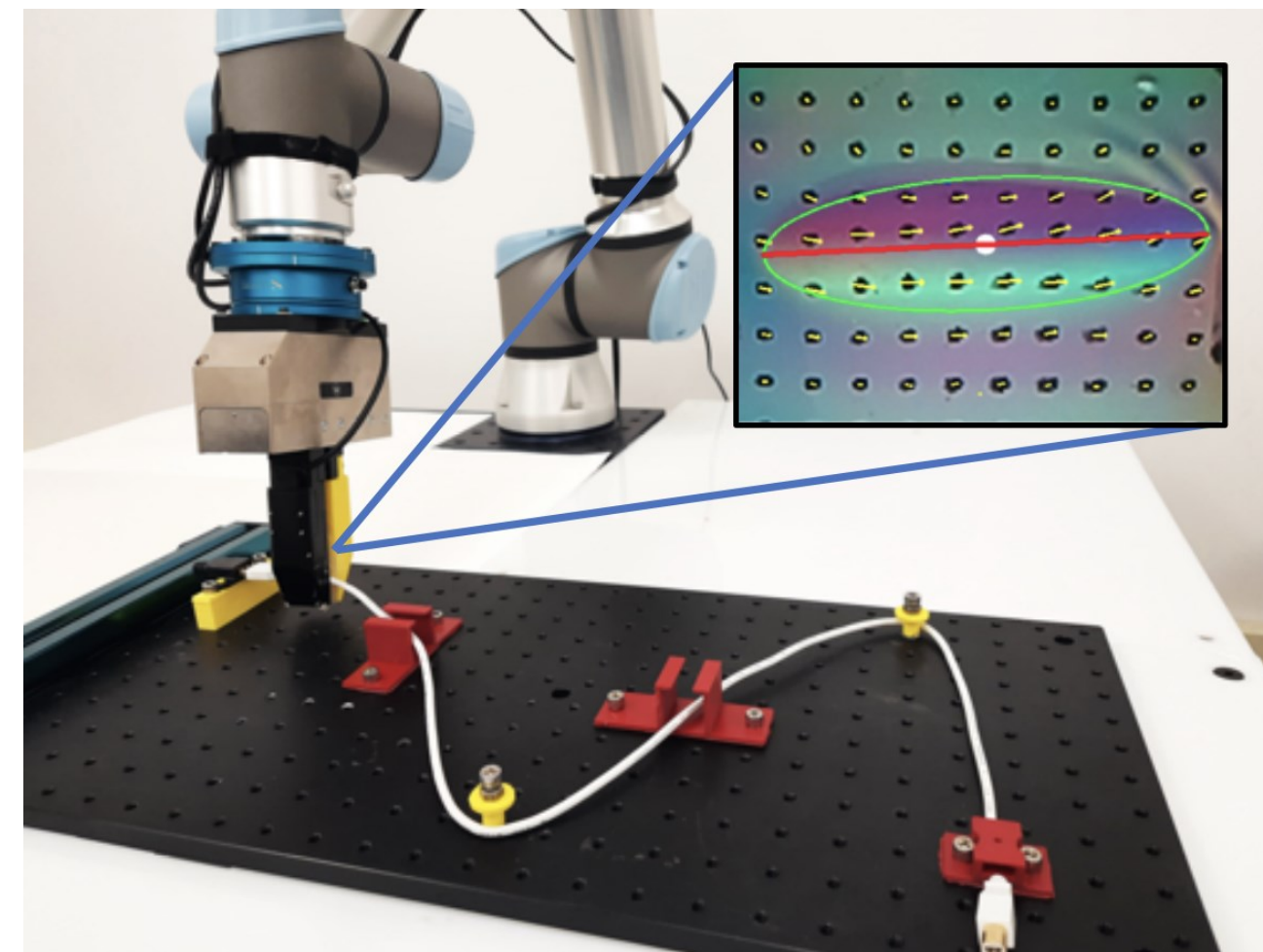
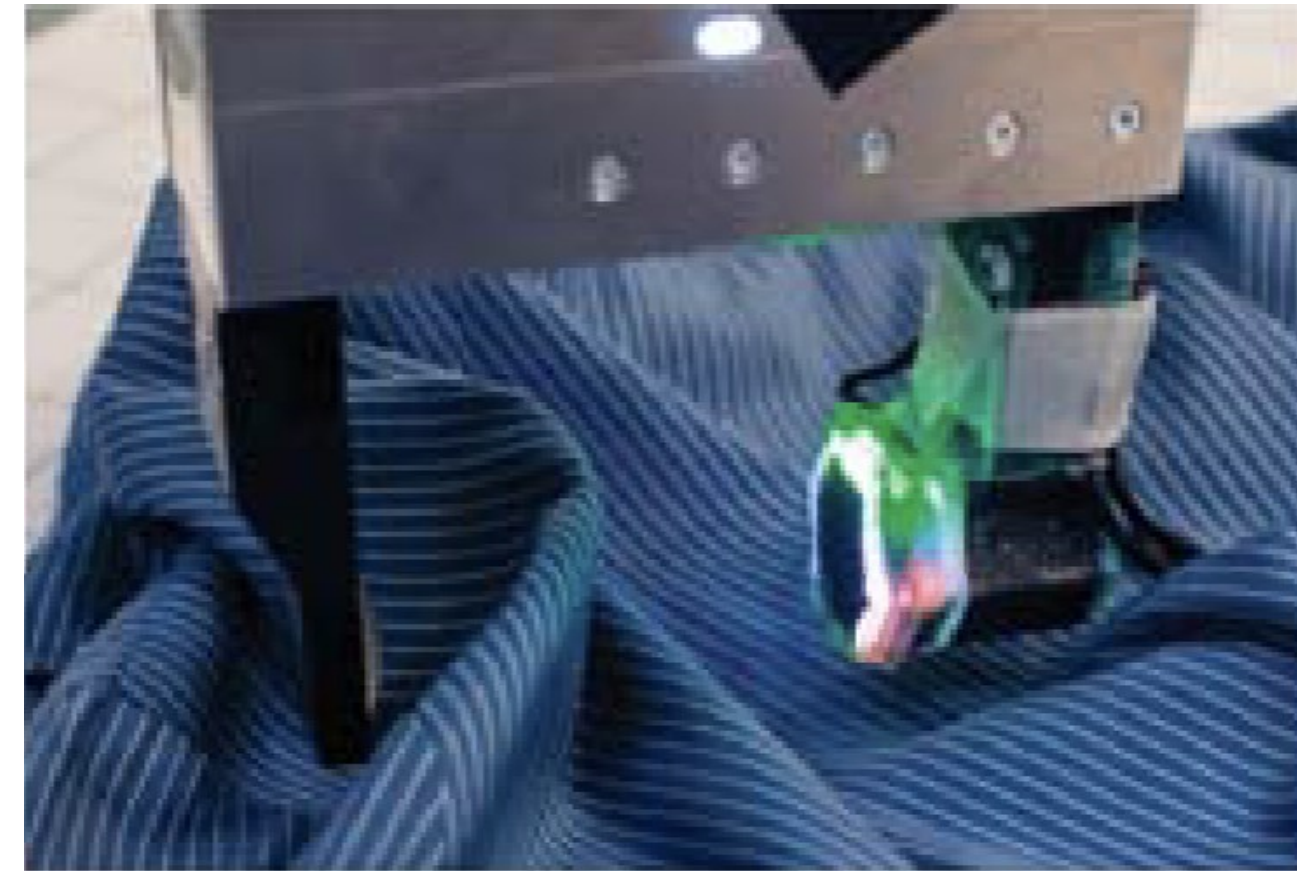


inverse problems

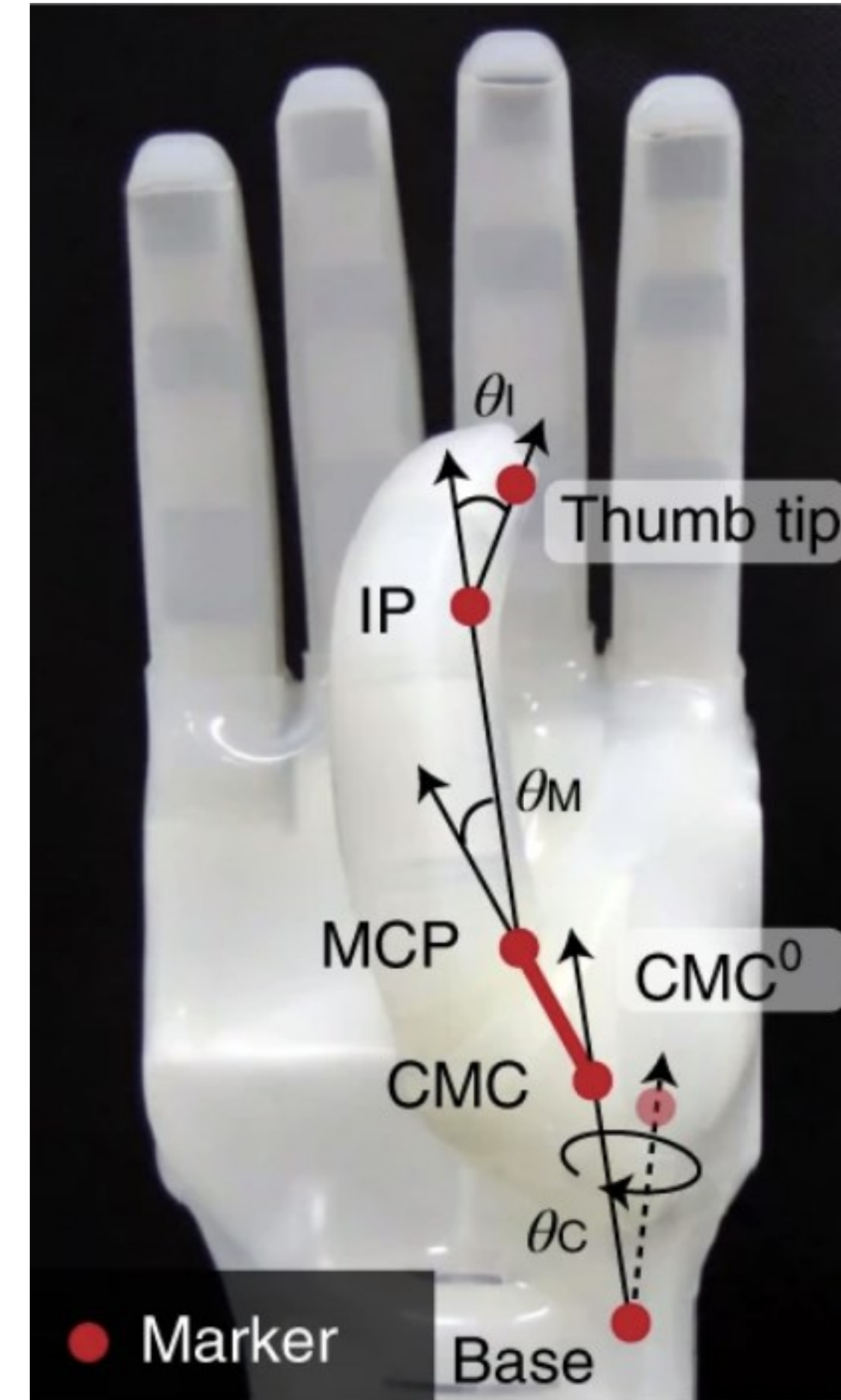
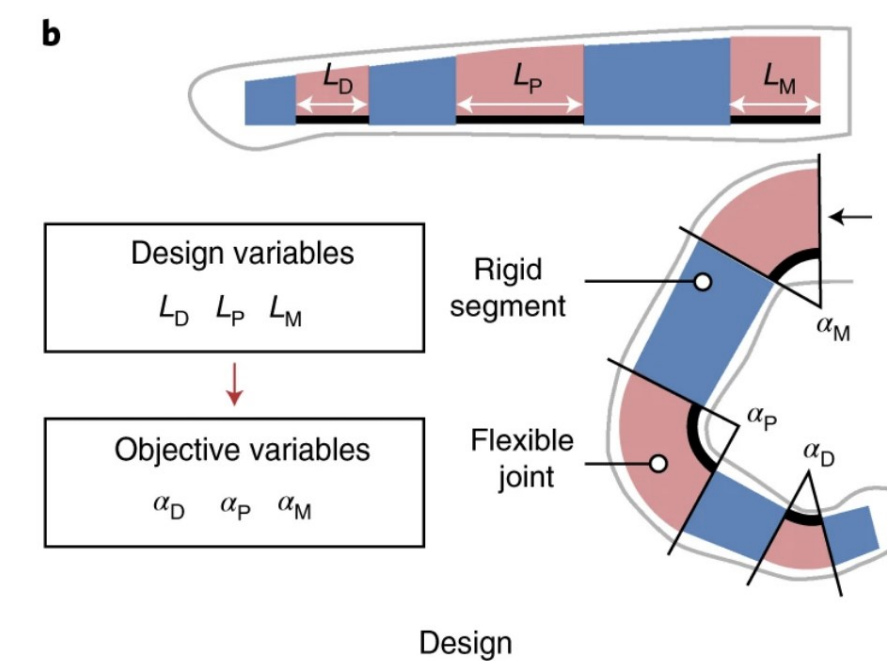
Why tactile sensing?



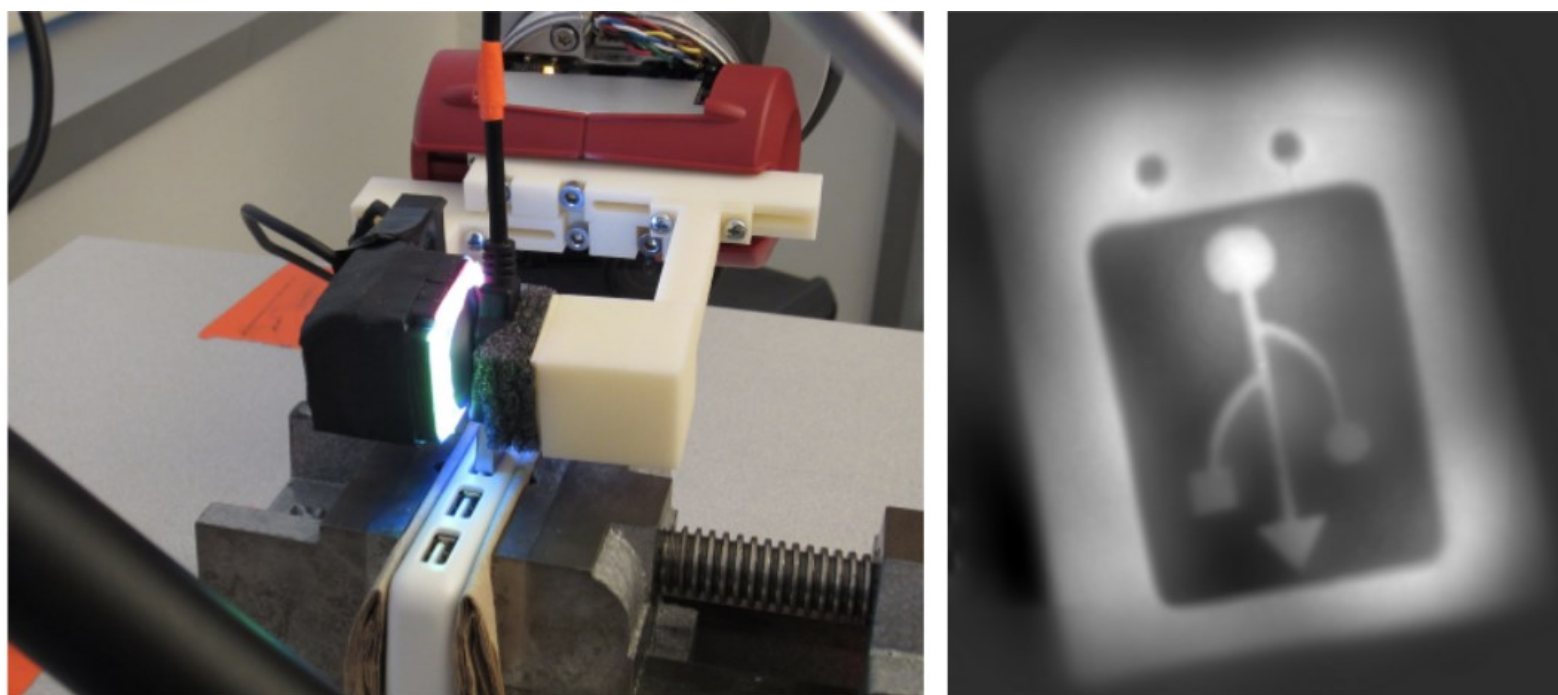
Object perception
[Huang et.al. 2022]



Robotic manipulation
[Yuan et.al. 2018]
[Wilson et.al. 2023]



Neuroprosthetics
[Gu et.al. 2023]

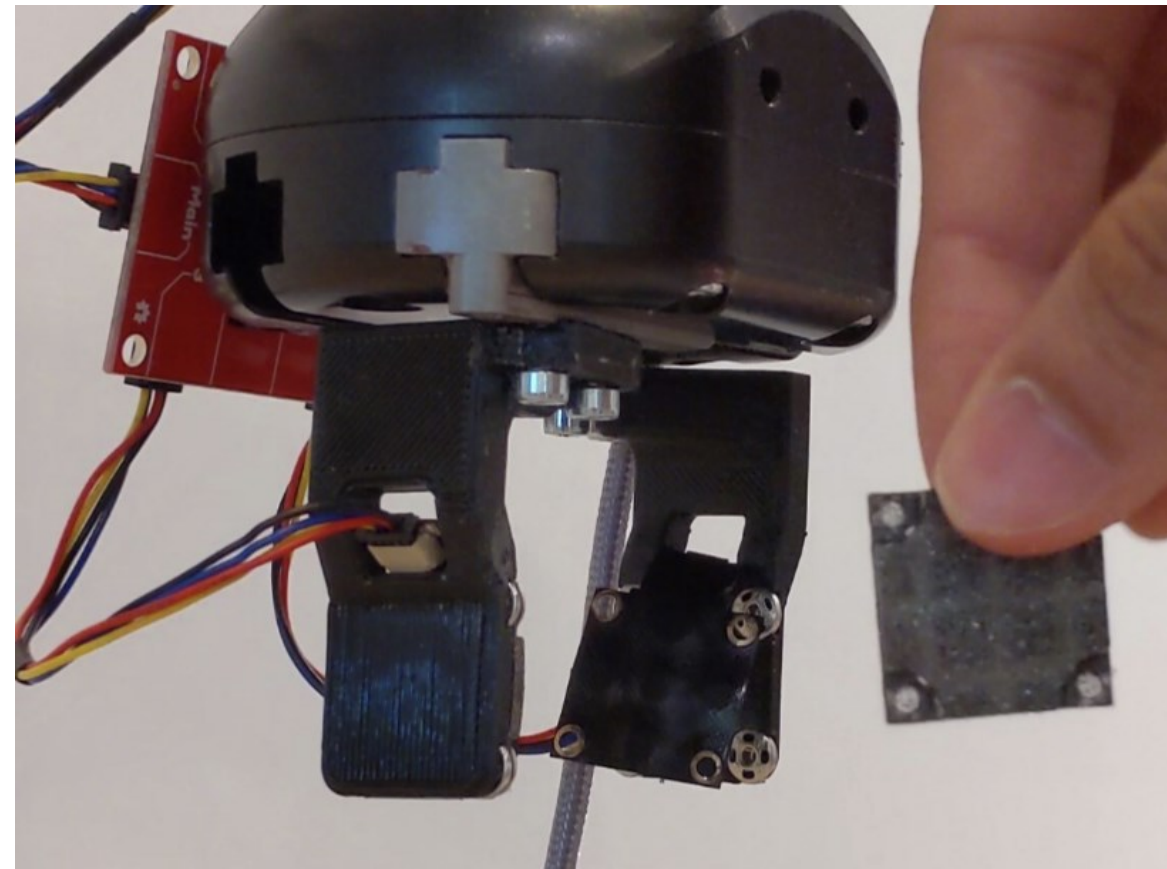


Advanced manufacturing
[Li et.al. 2014]

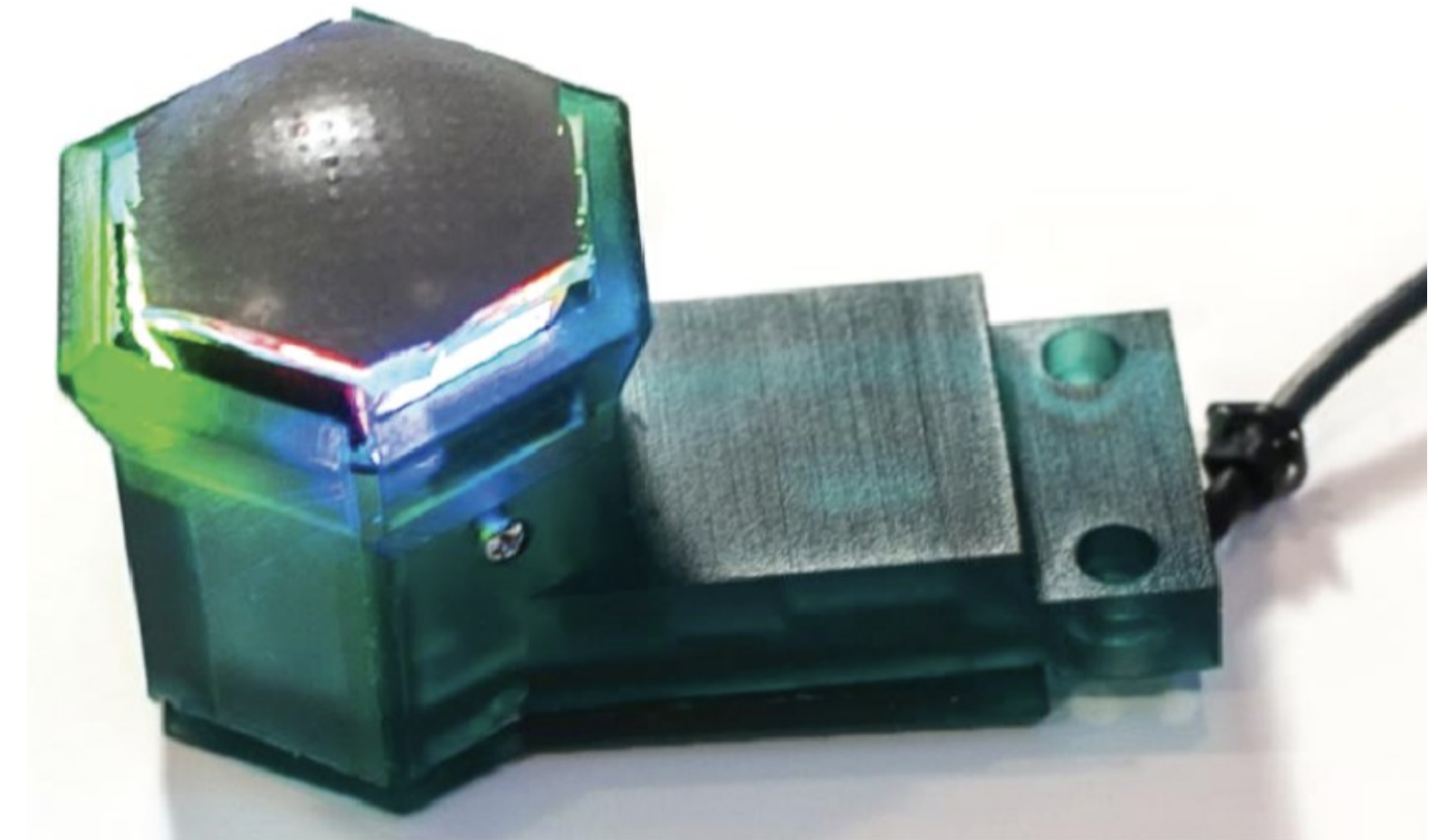
Tactile sensors



BioTac
ionically-conductive fluid based



ReSkin
Magnetic field based

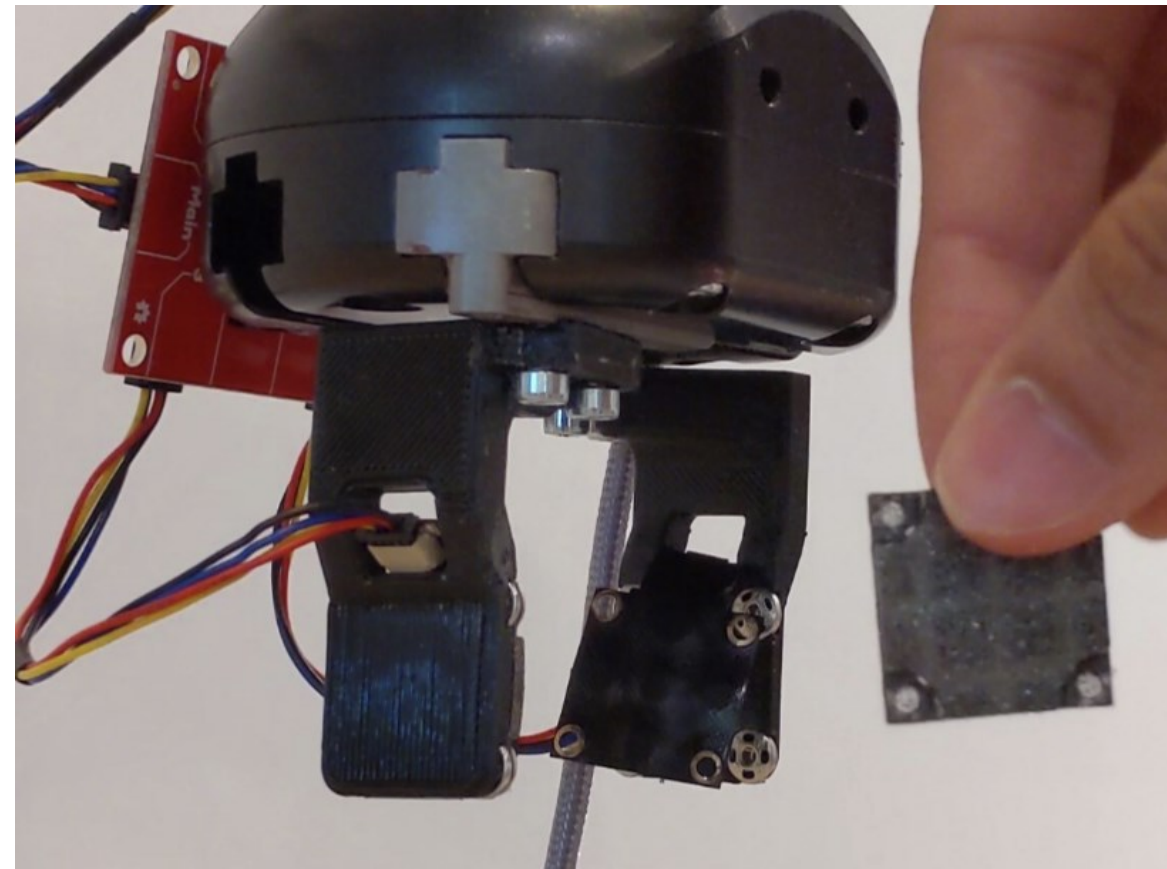


GelSight
Vision-based

Tactile sensors



BioTac
ionically-conductive fluid based

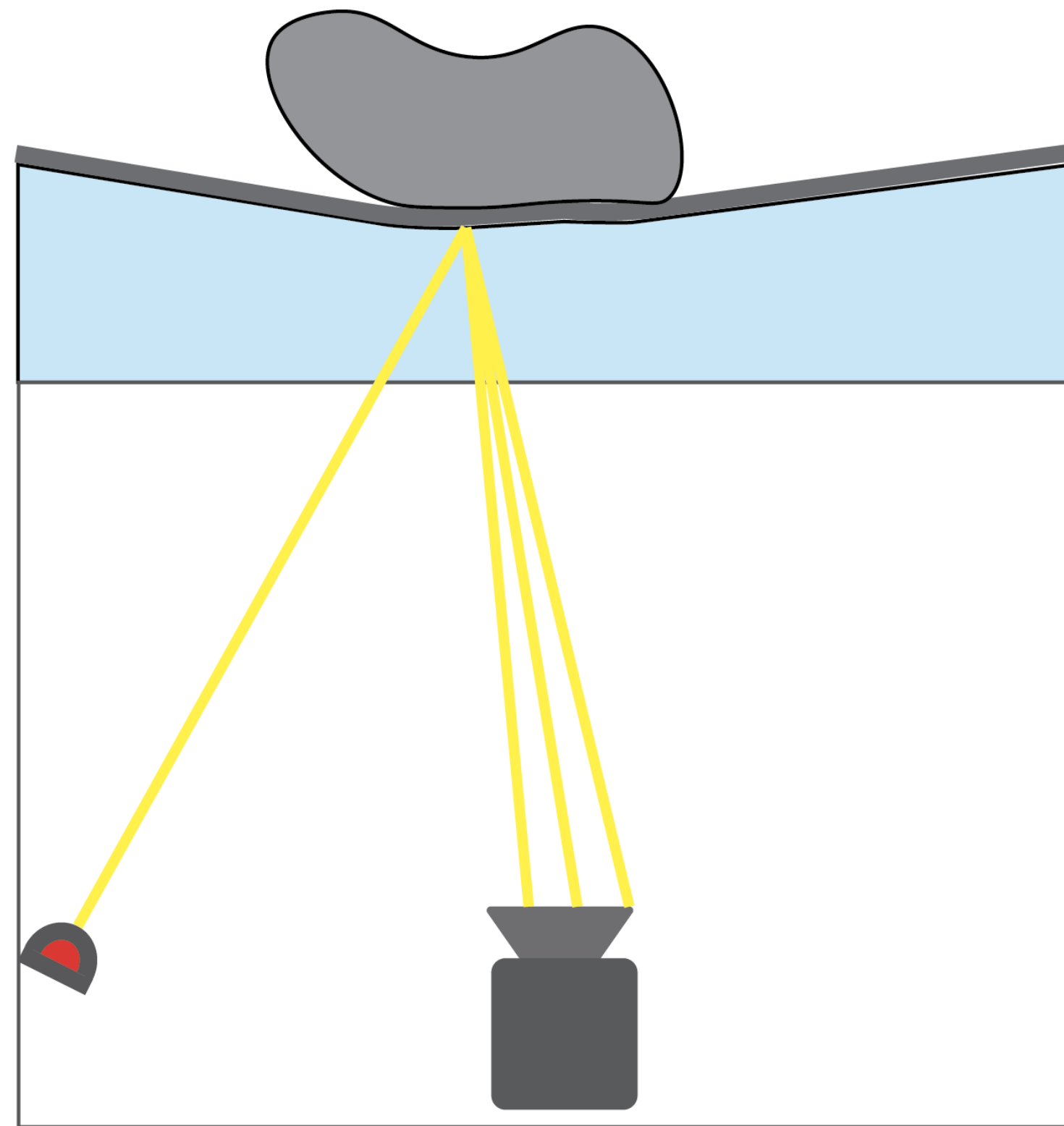


ReSkin
Magnetic field based

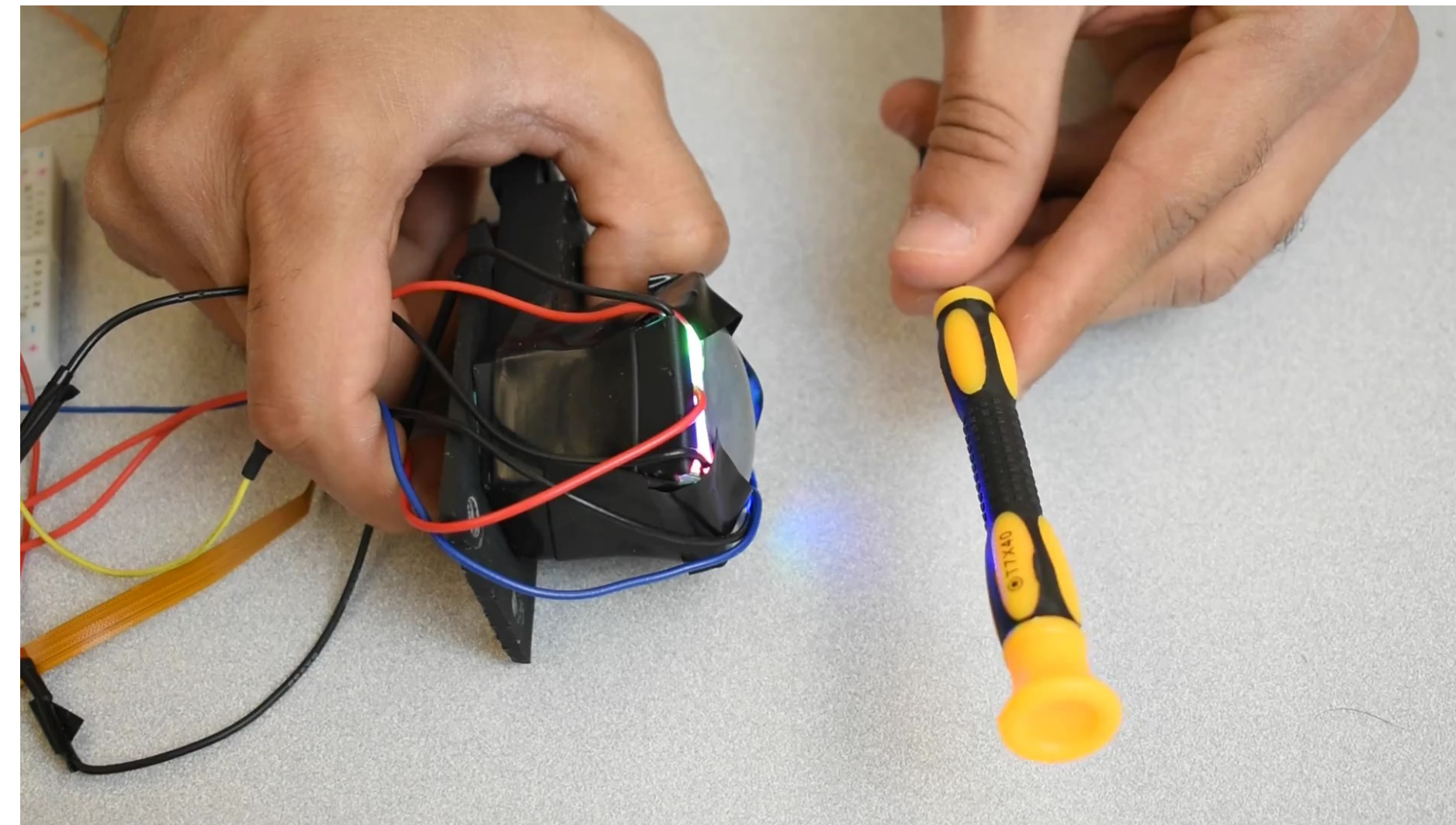


GelSight
Vision-based

Vision-based tactile sensors: working principle



Sensor schematic

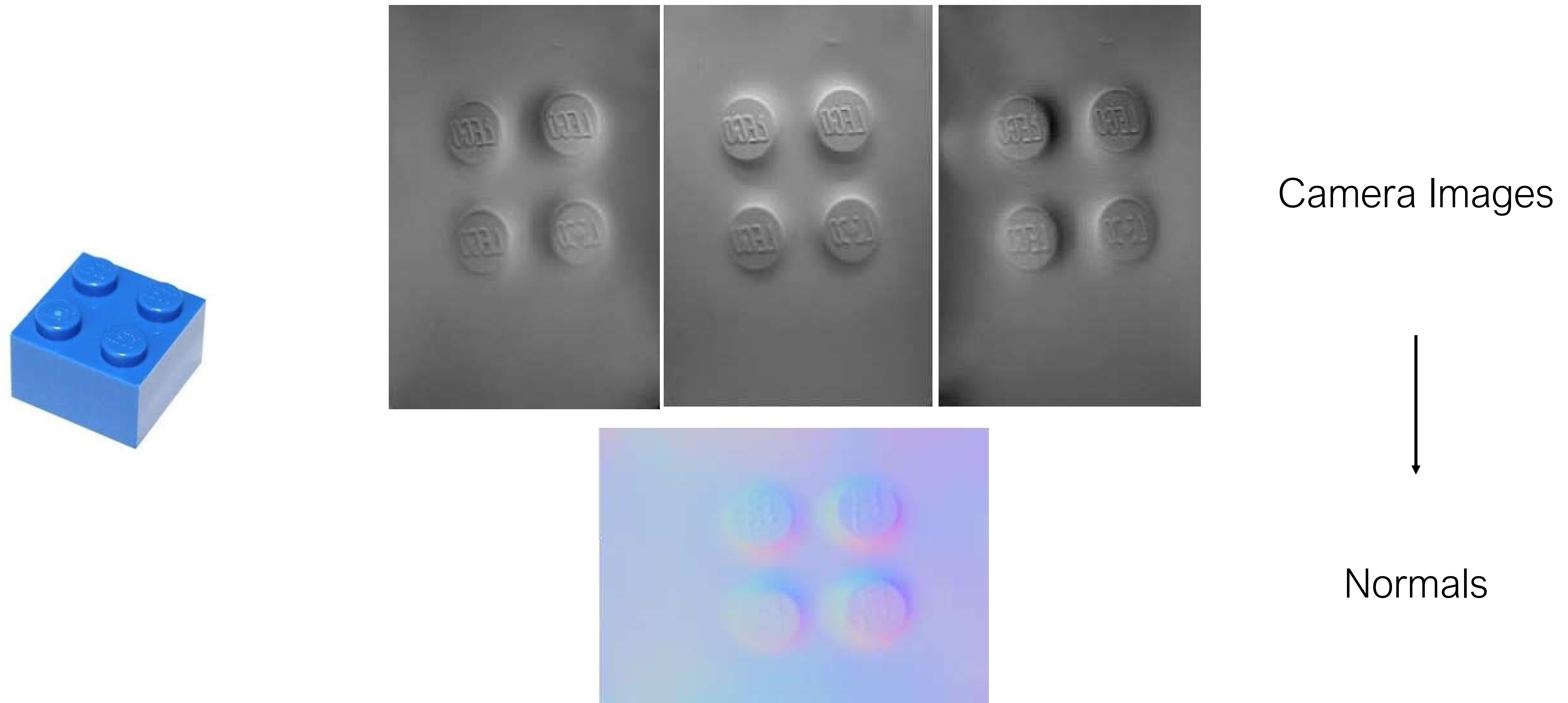


Live view



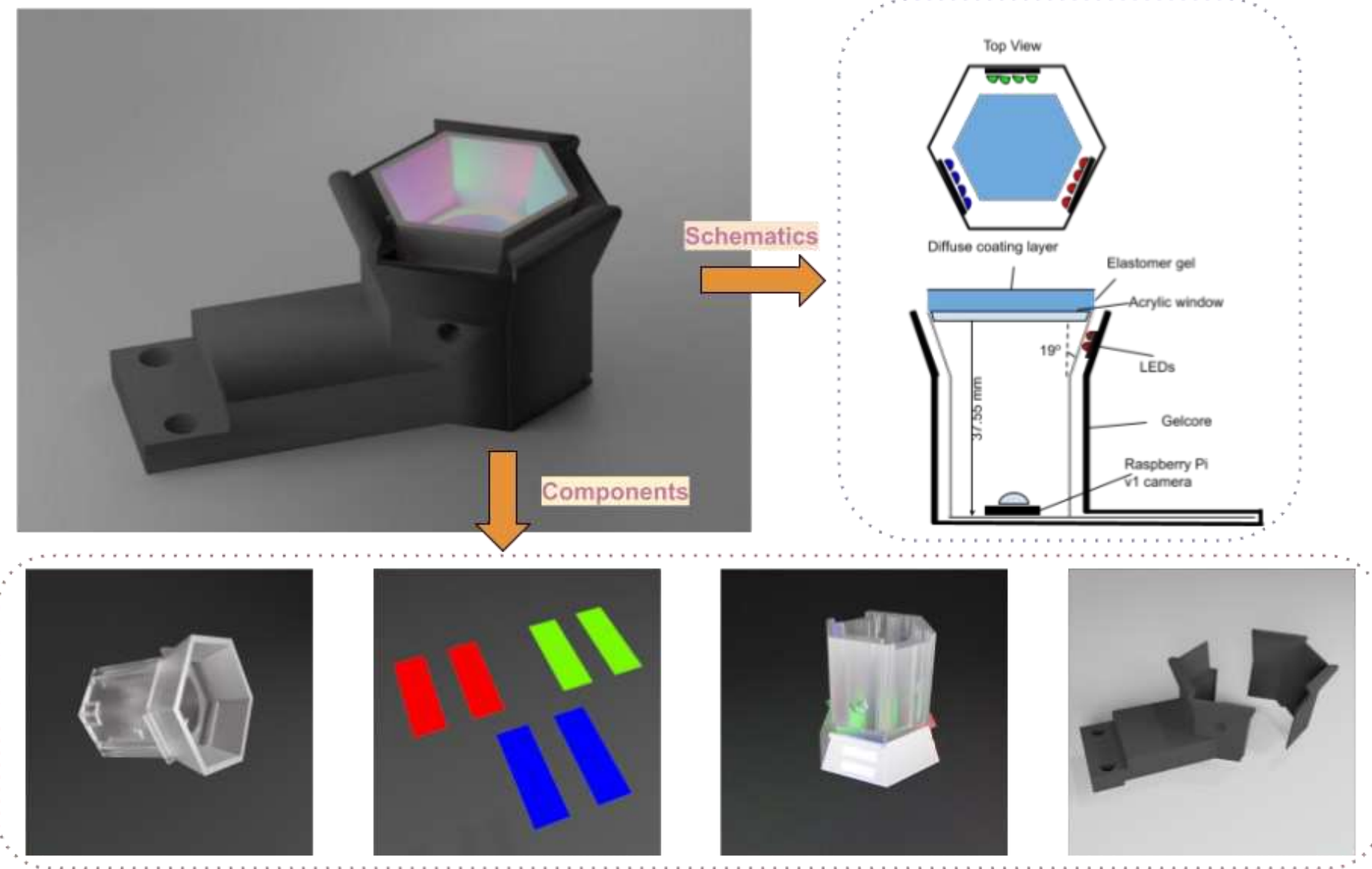
Sensor view

Vision-based tactile sensors: photometric stereo

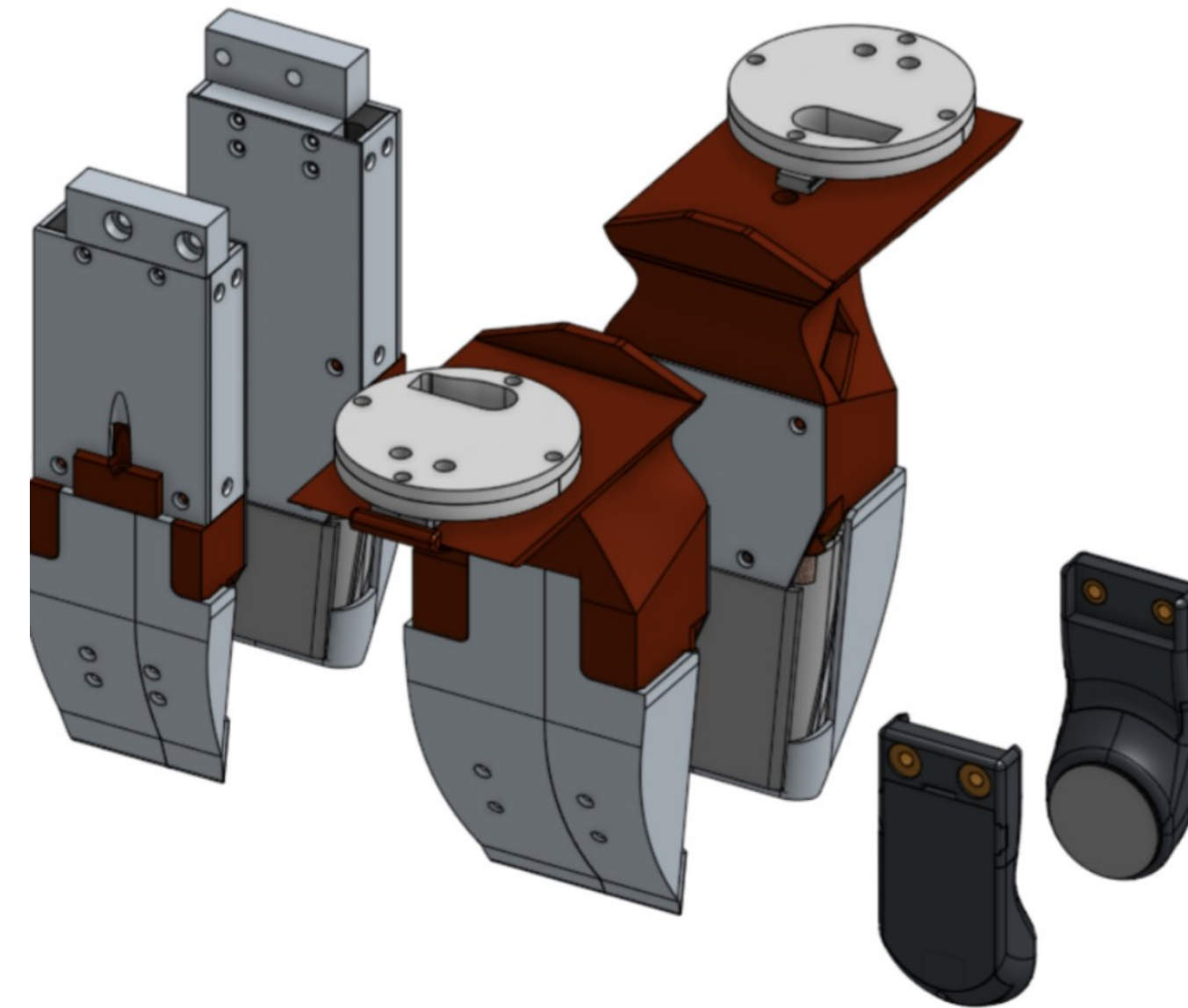


With photometric stereo, GelSight can encode surface normals as an RGB image.

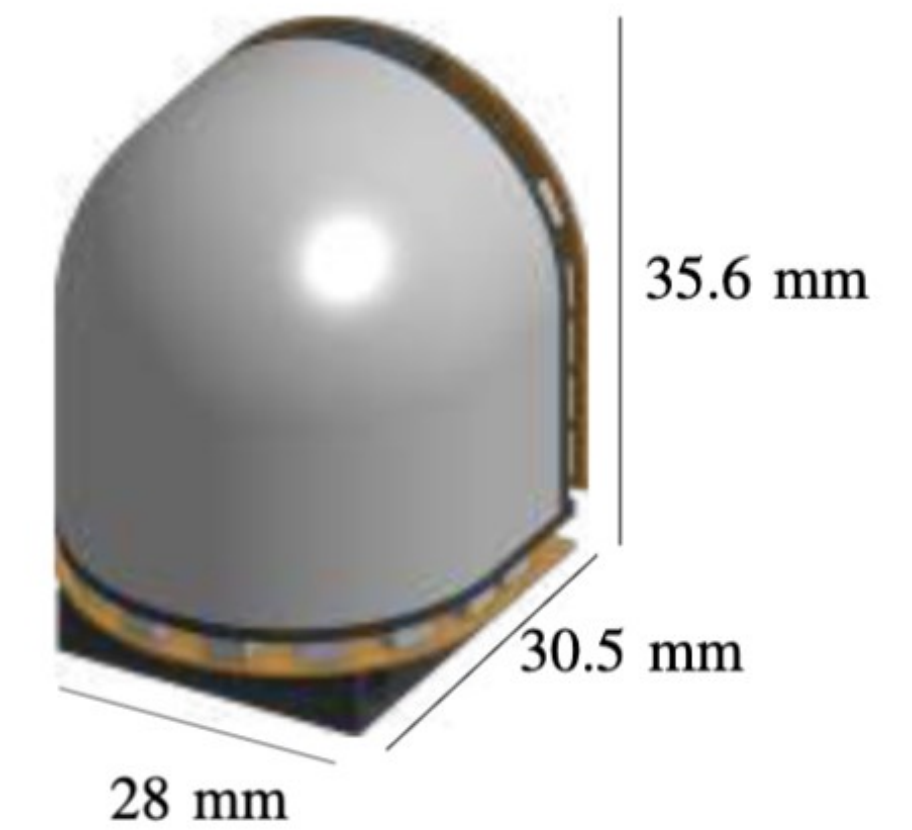
Vision-based tactile sensors: design variants



FlatGel GelSight
Dong et. al. 2017
Agarwal et. al. 2021



GelSlim Family
Donlon et. al. 2018
Ma et. al. 2019
Hogan et. al. 2020
Taylor et. al. 2021

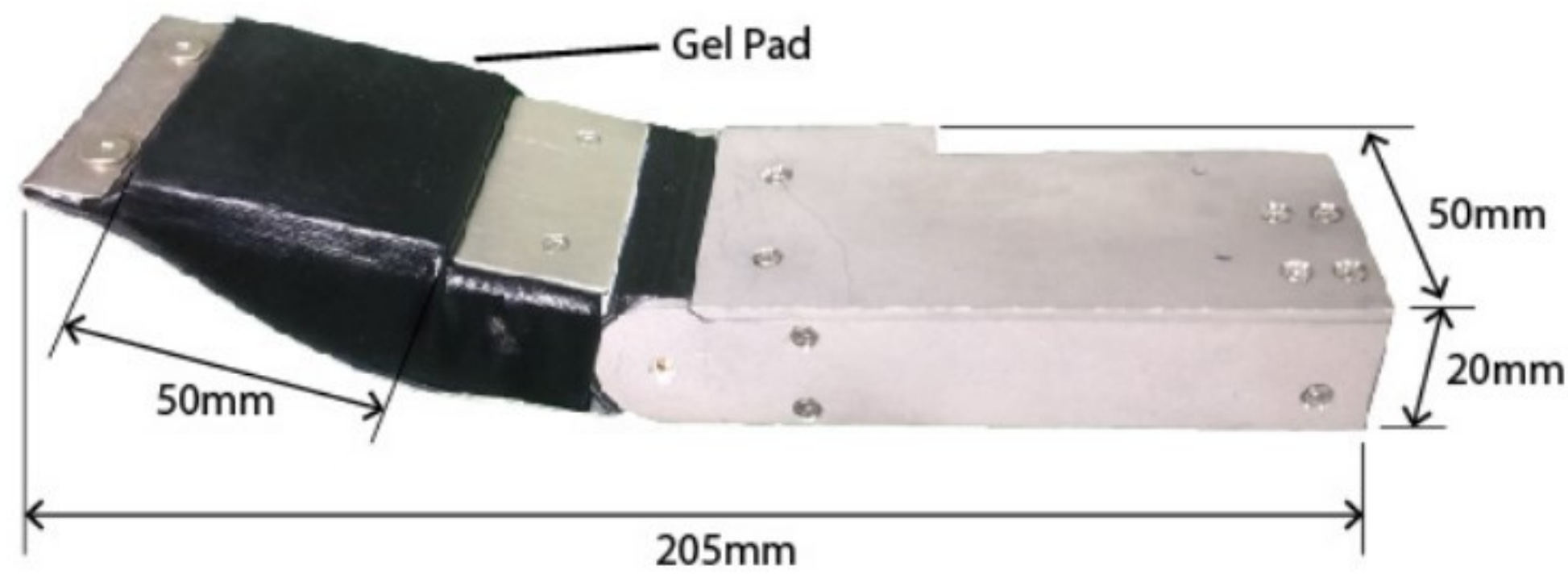


RoundTip GelSight
Romero et. al. 2020

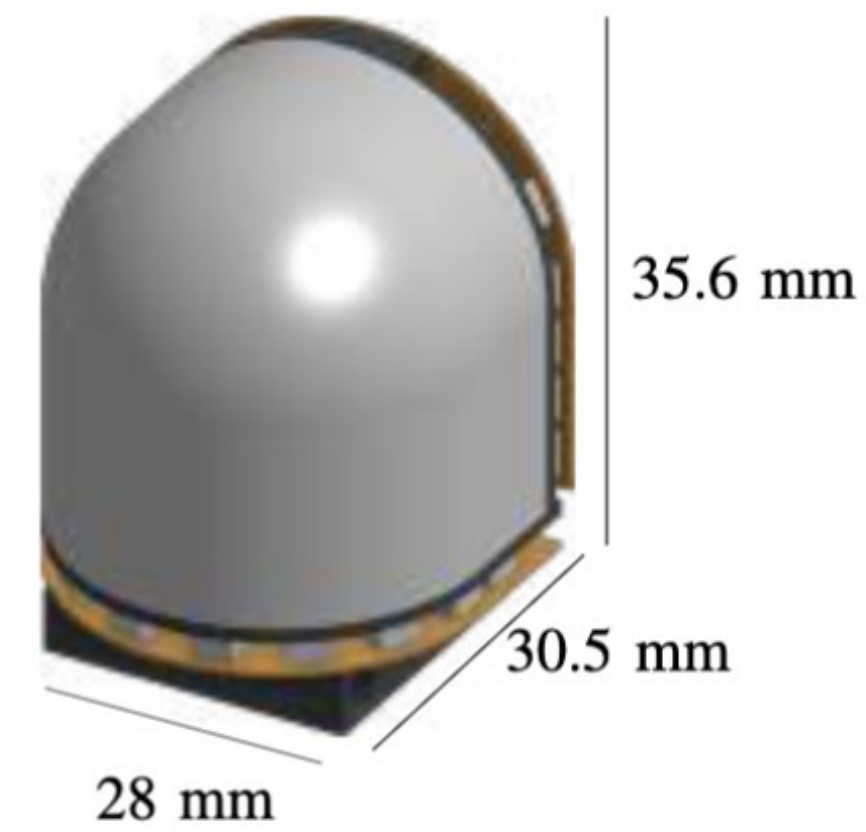
Designing VBTS is hard

Designing VBTS is hard

- Diversity of sensor shape and required form-factor



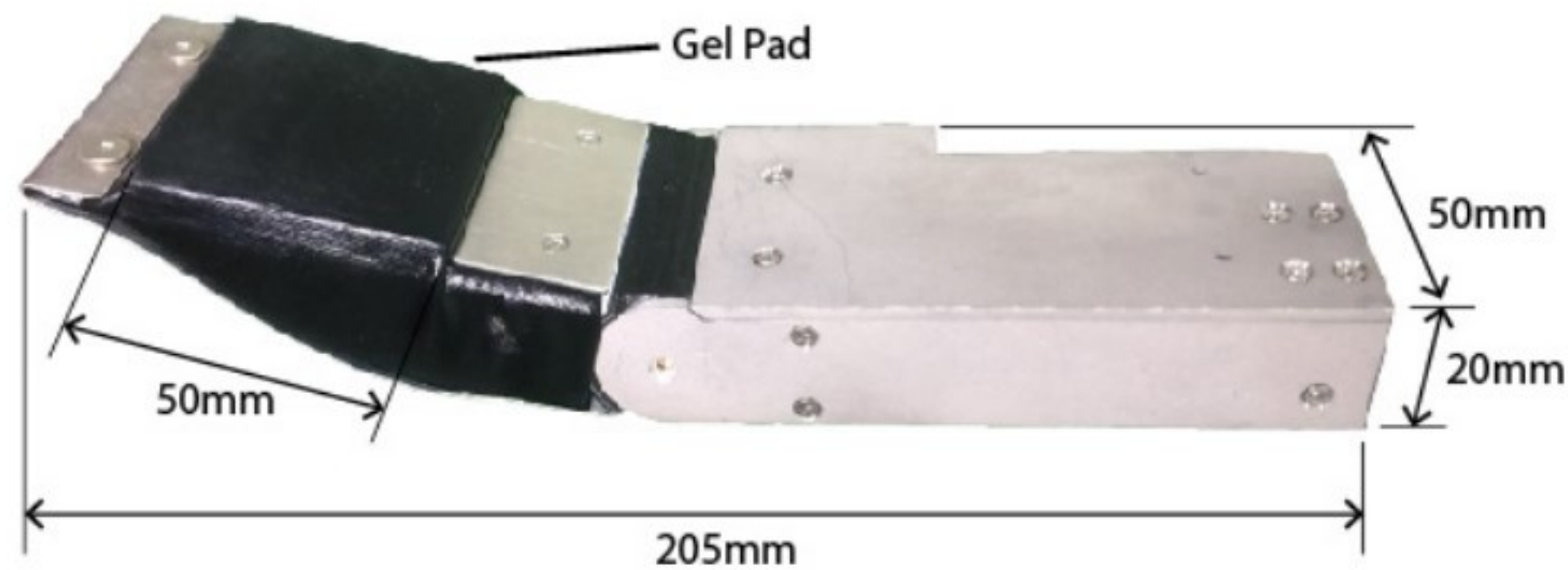
Flat sensing surface



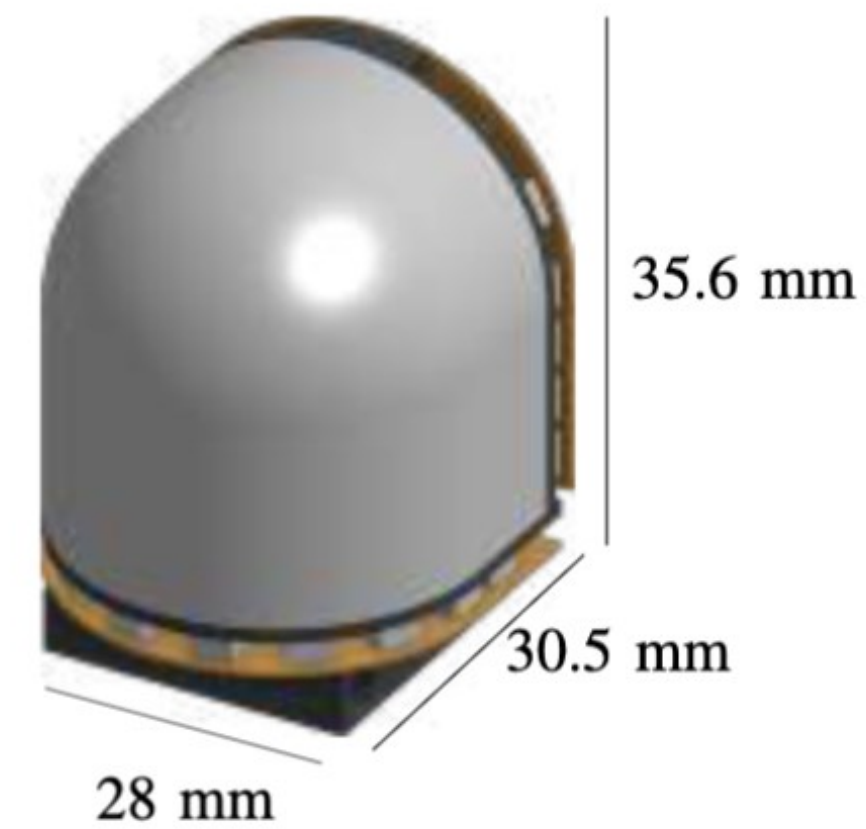
Curved sensing surface

Designing VBTS is hard

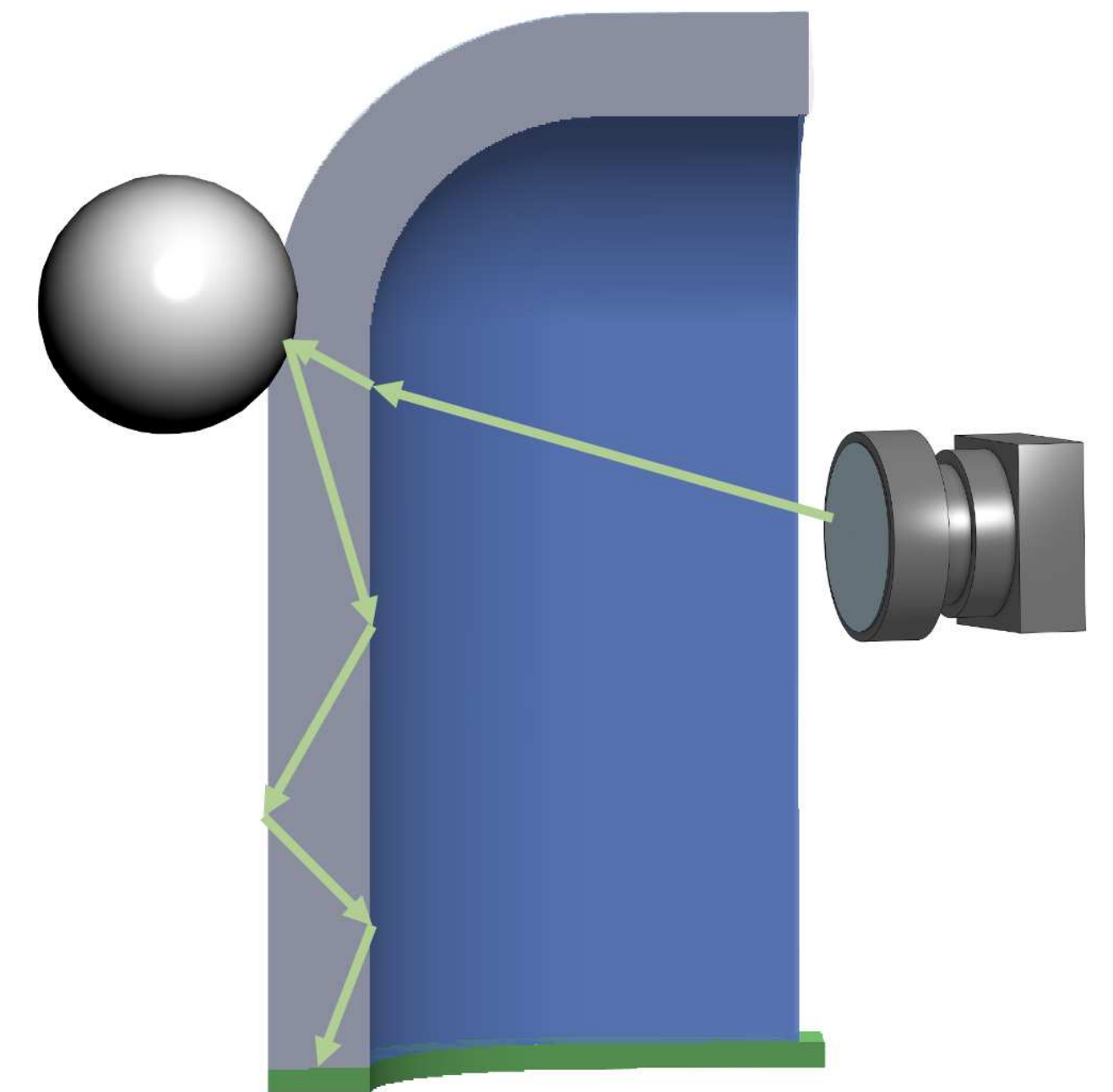
- Diversity of sensor shape and required form-factor
- Complex light interaction



Flat sensing surface

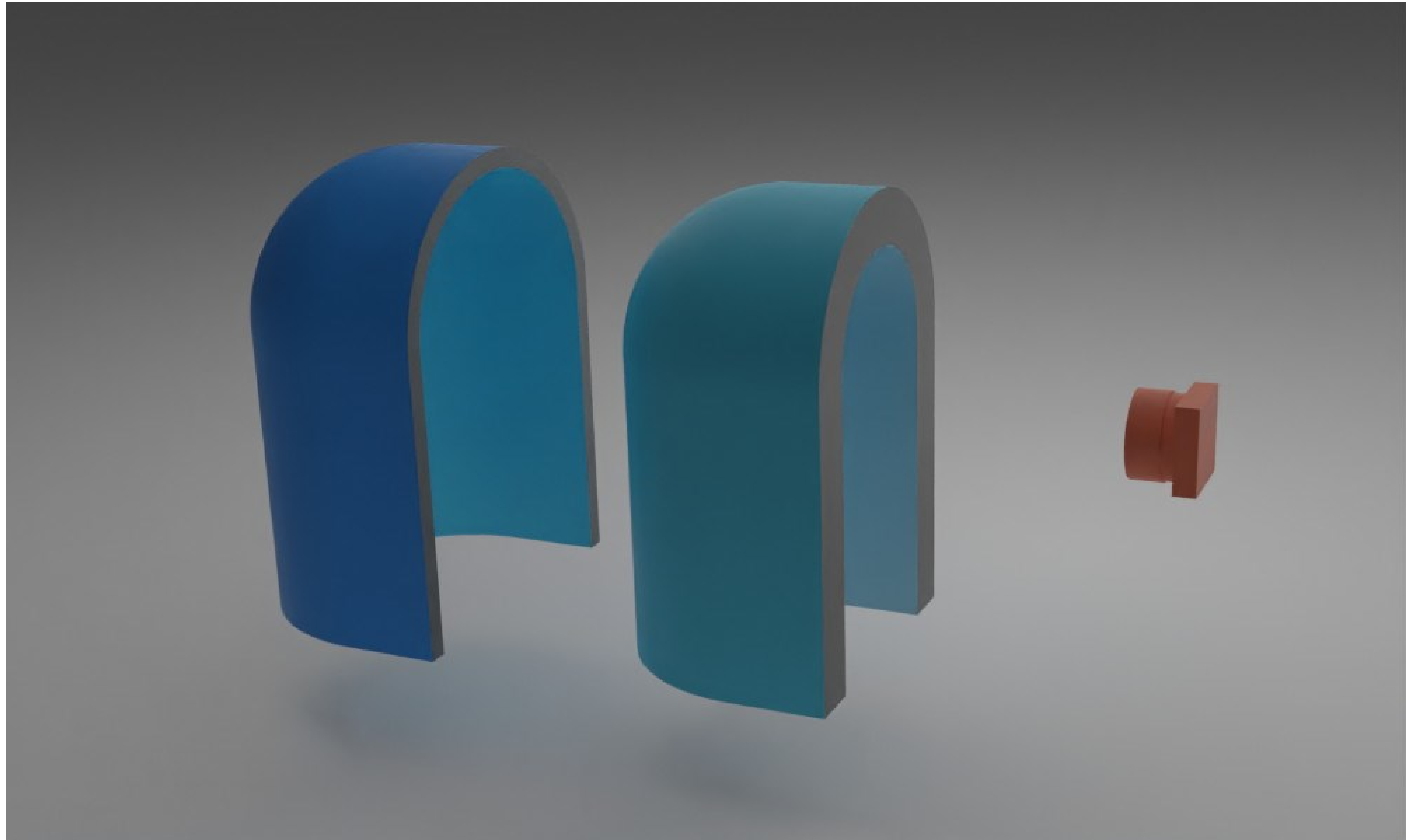


Curved sensing surface

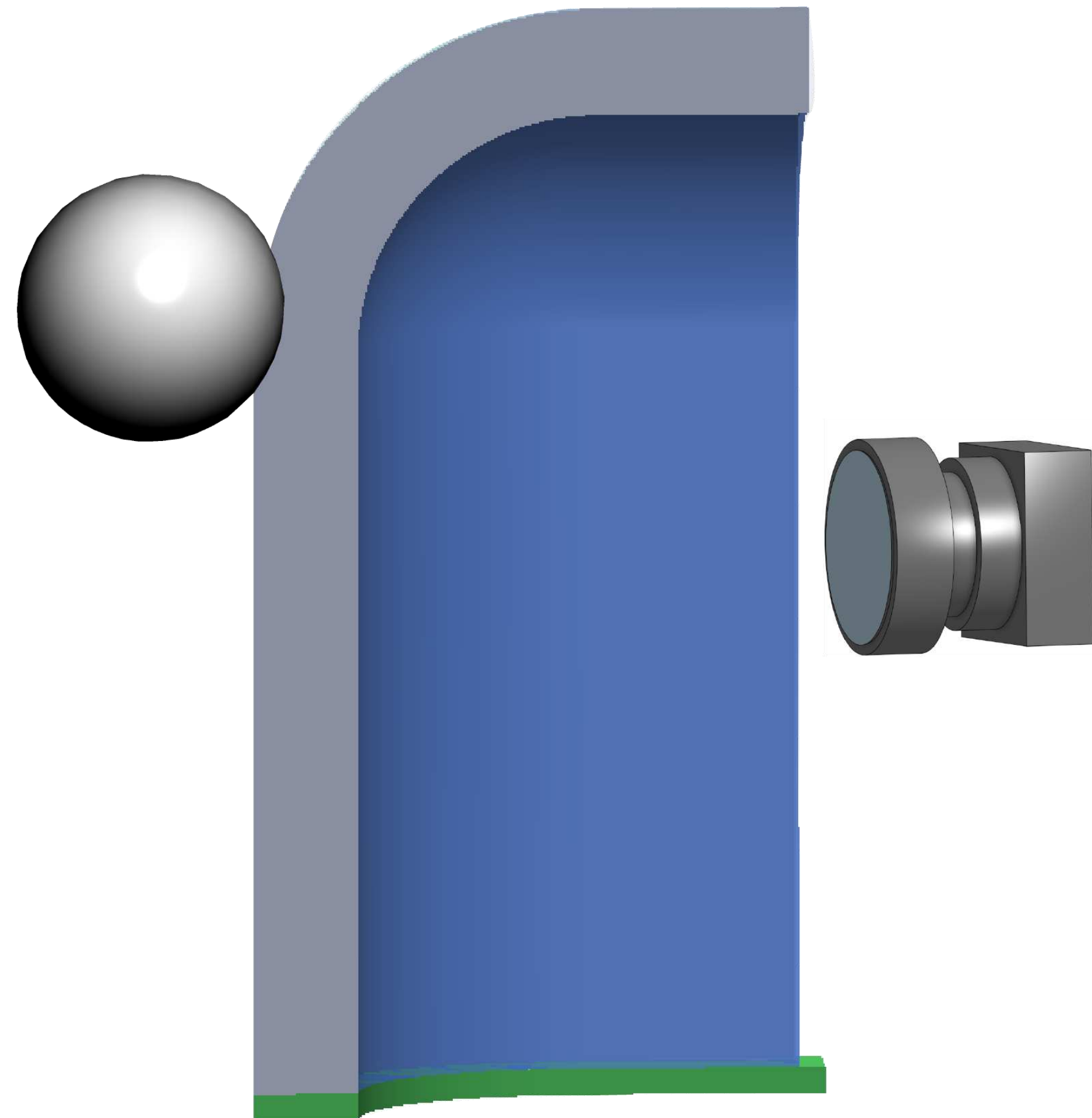


Curved tactile sensor

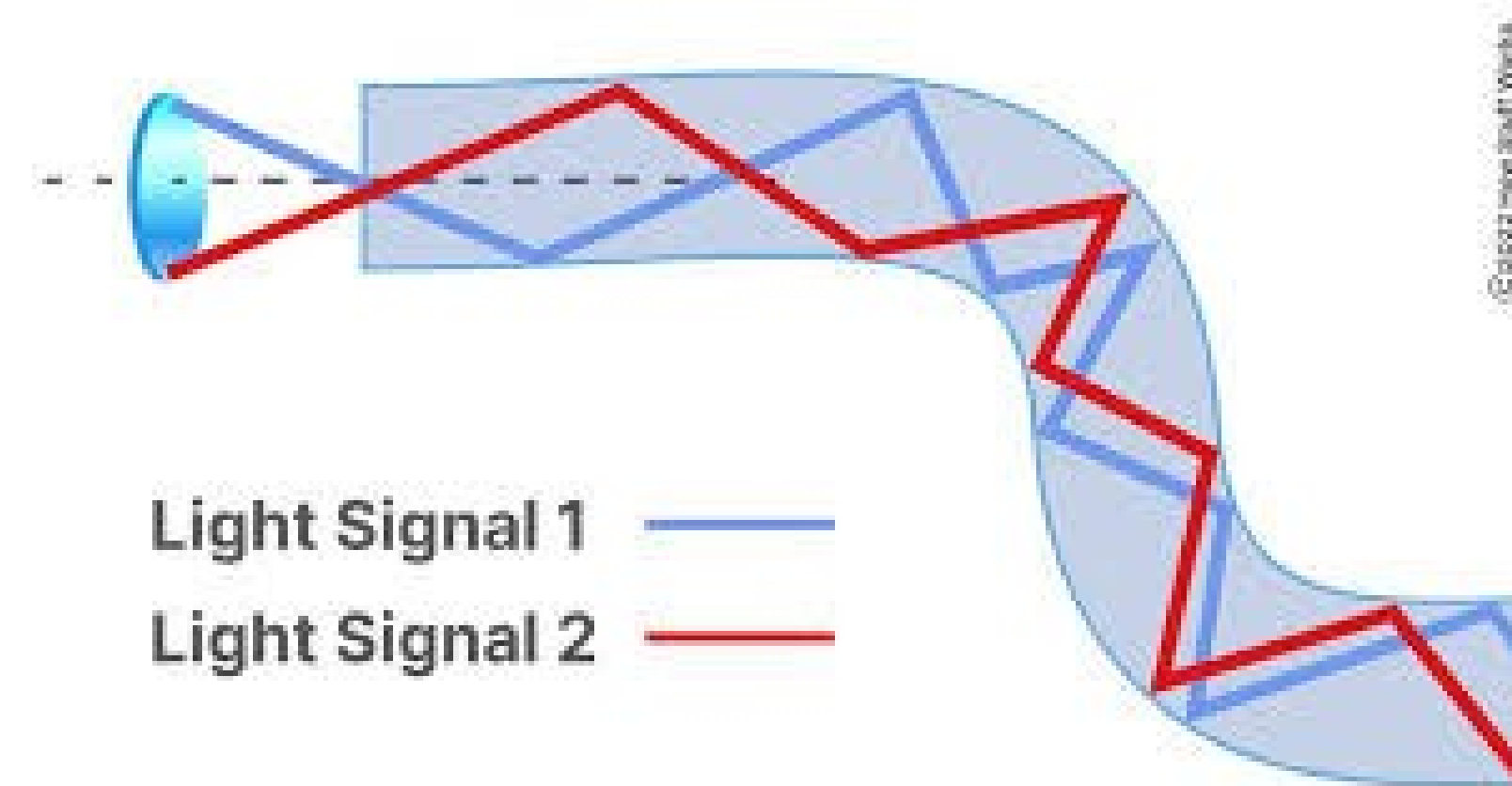
Curved tactile sensor



Curved tactile sensor: working principle

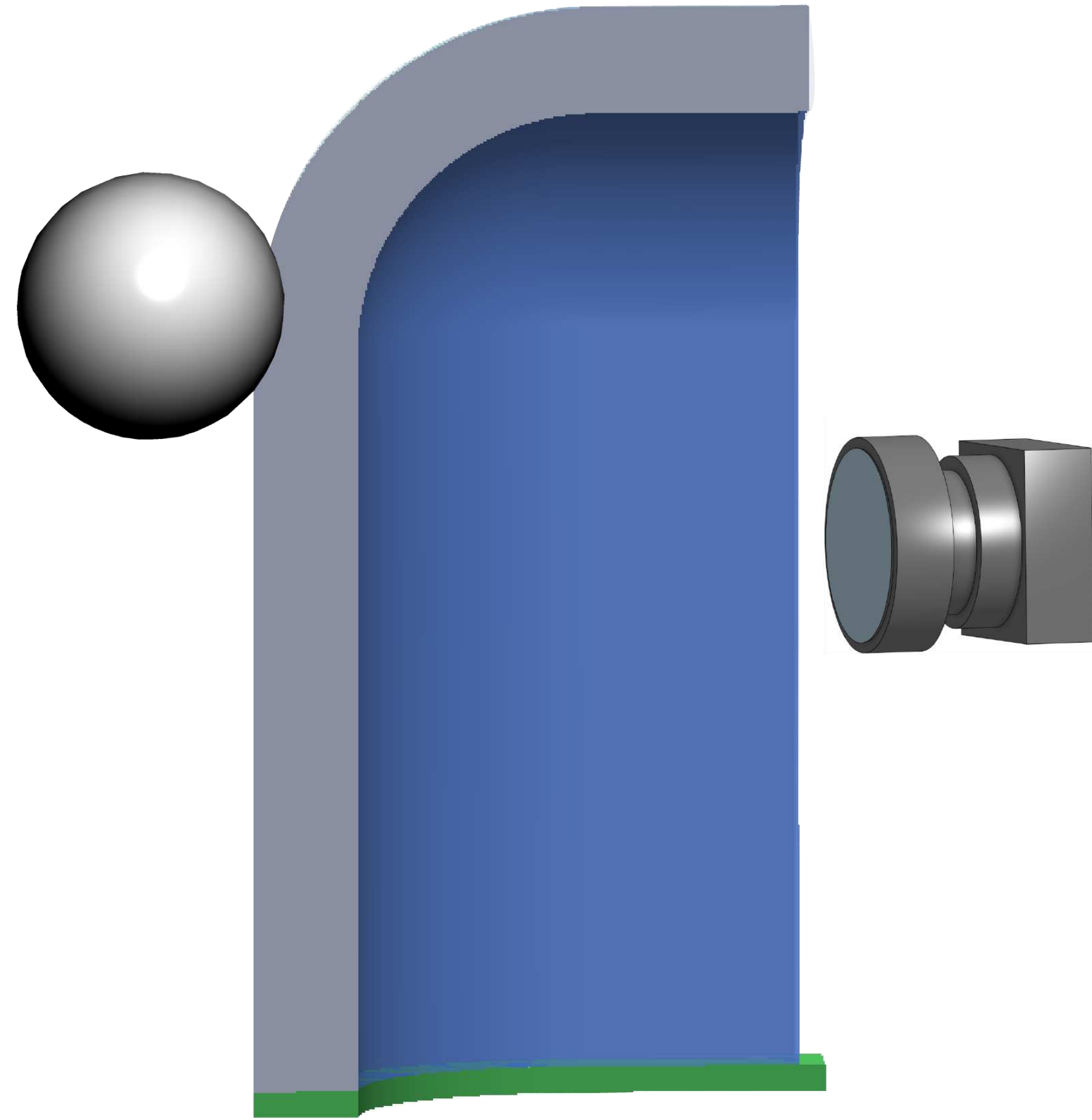


Section View



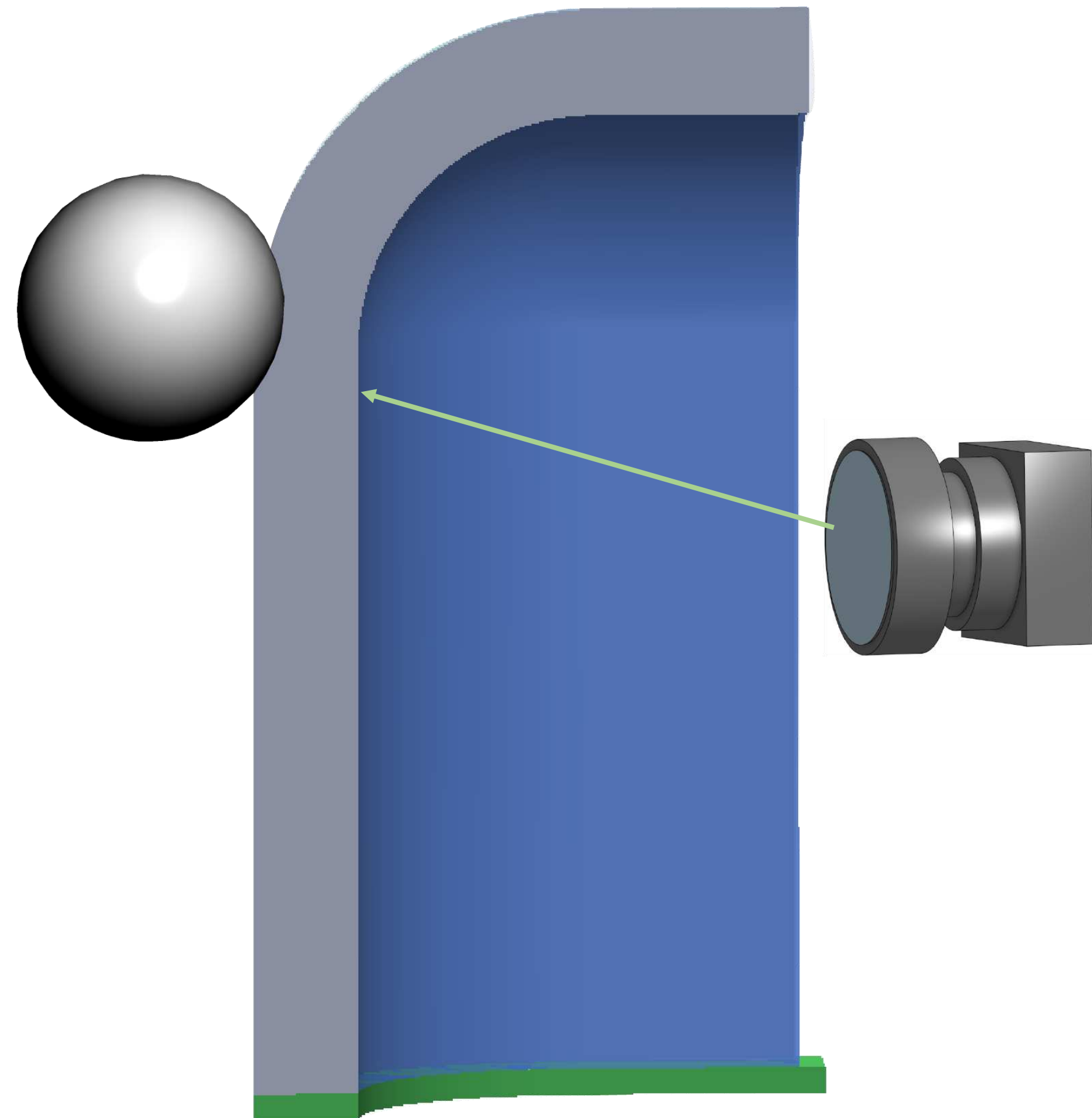
Light Piping

Curved tactile sensor: working principle

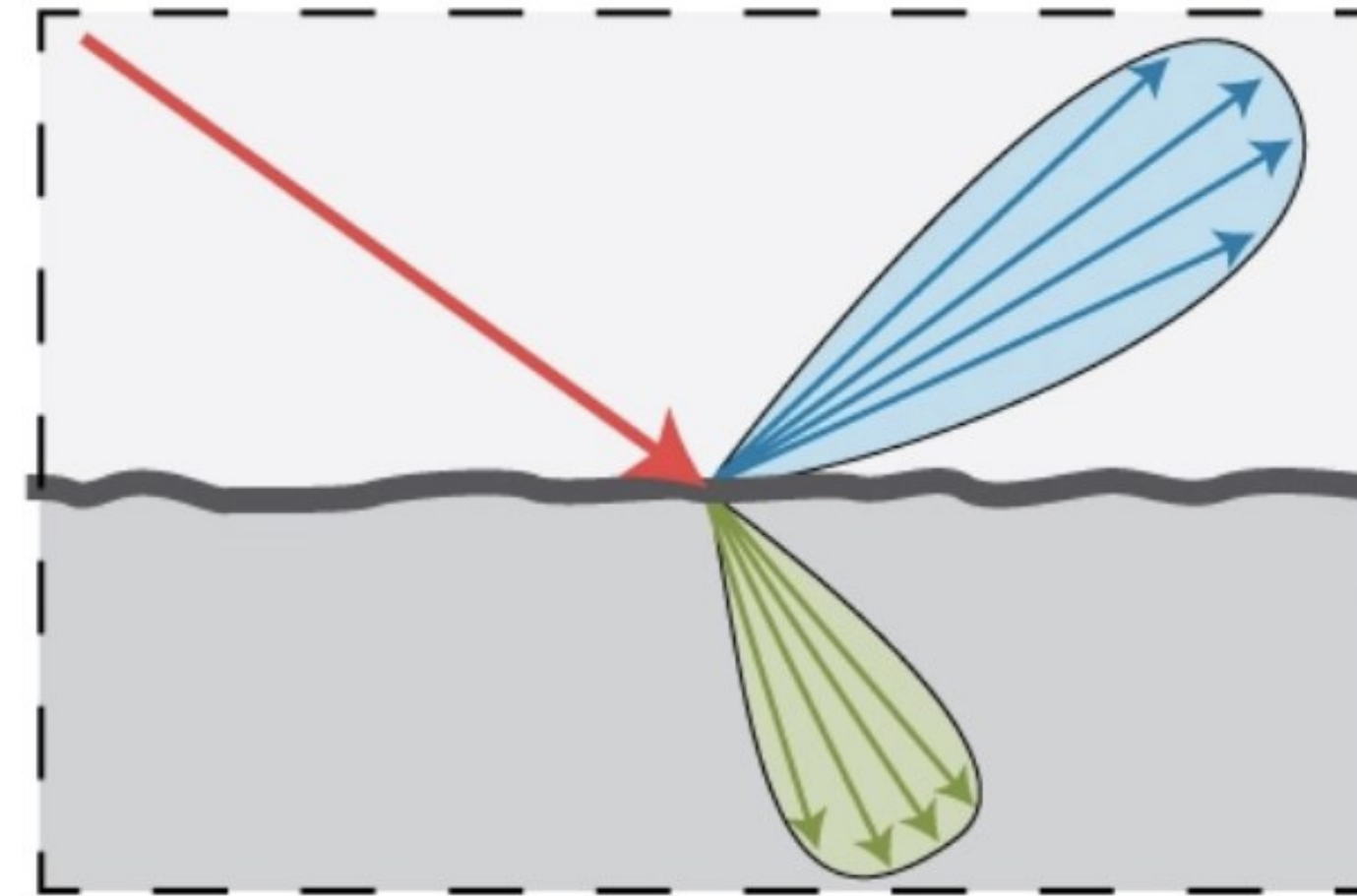


Section View

Curved tactile sensor: working principle

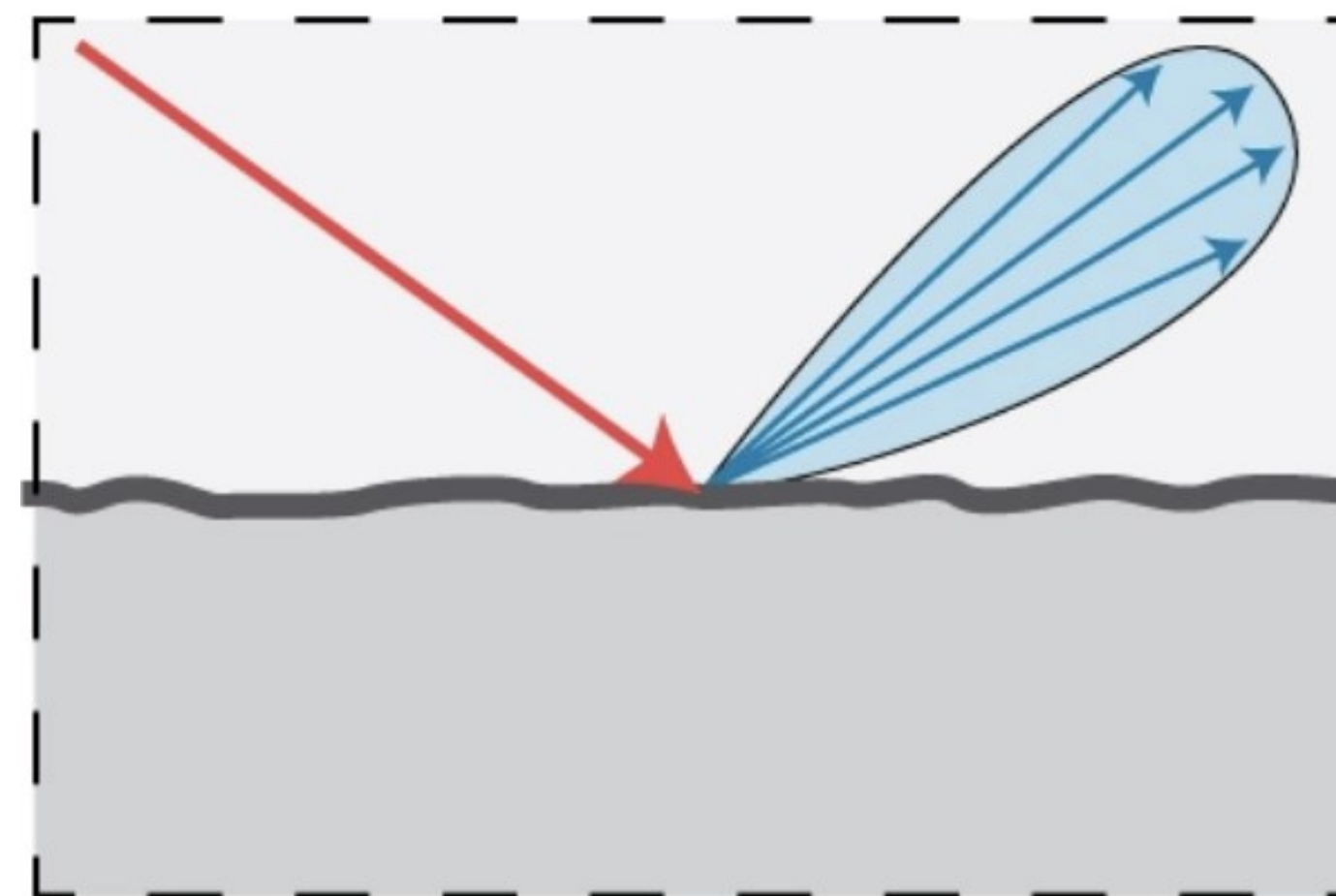
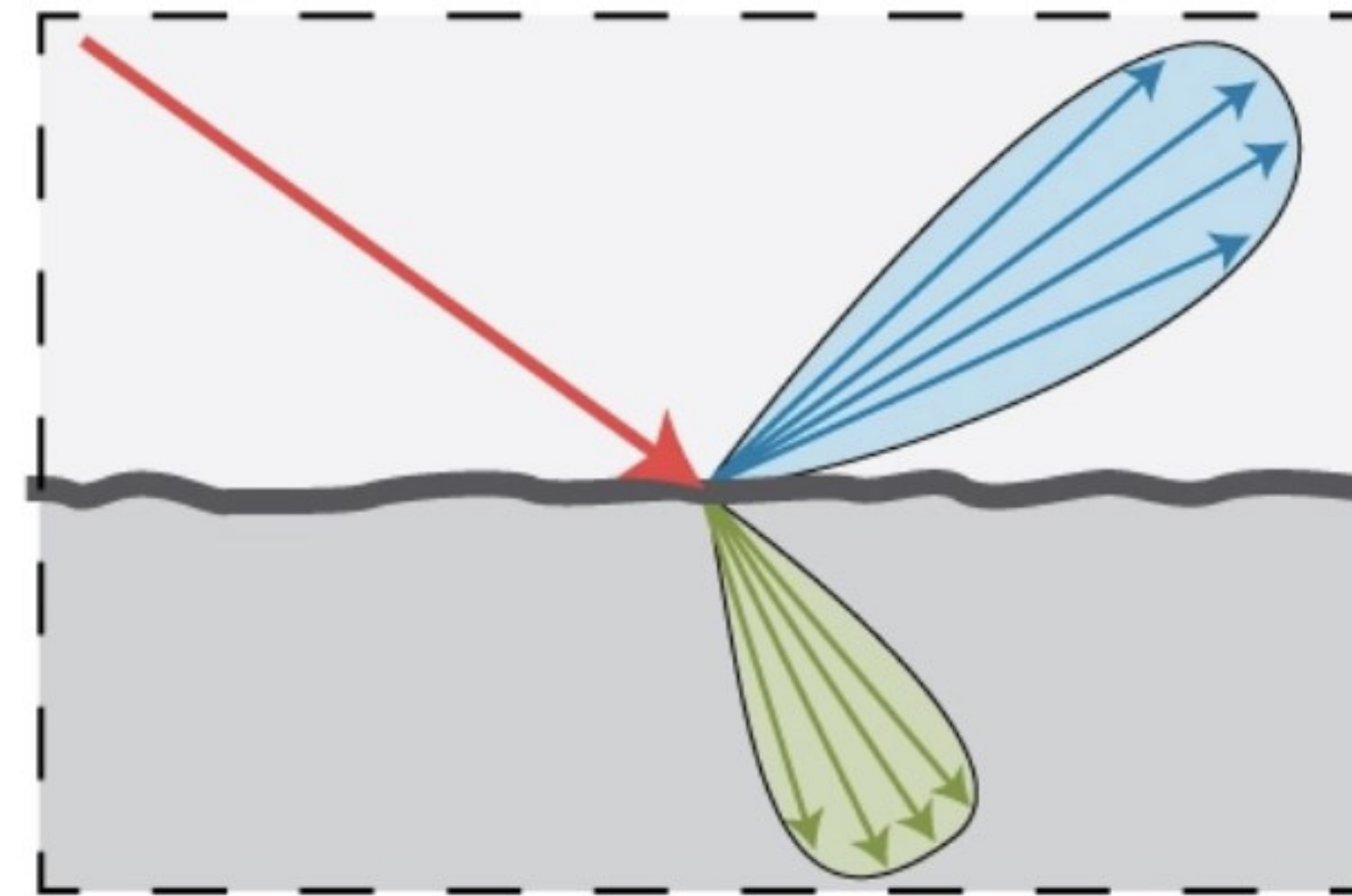
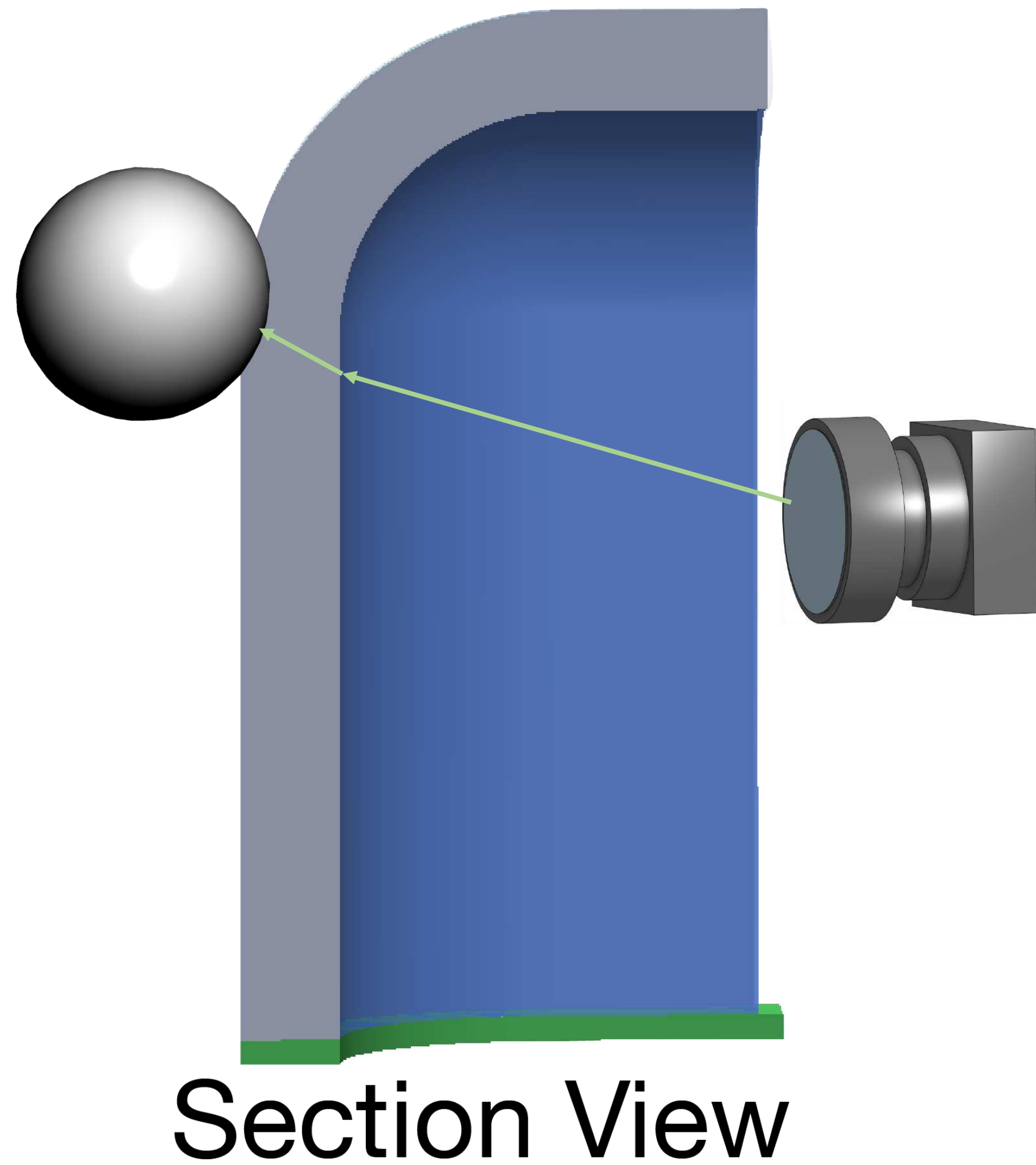


Section View

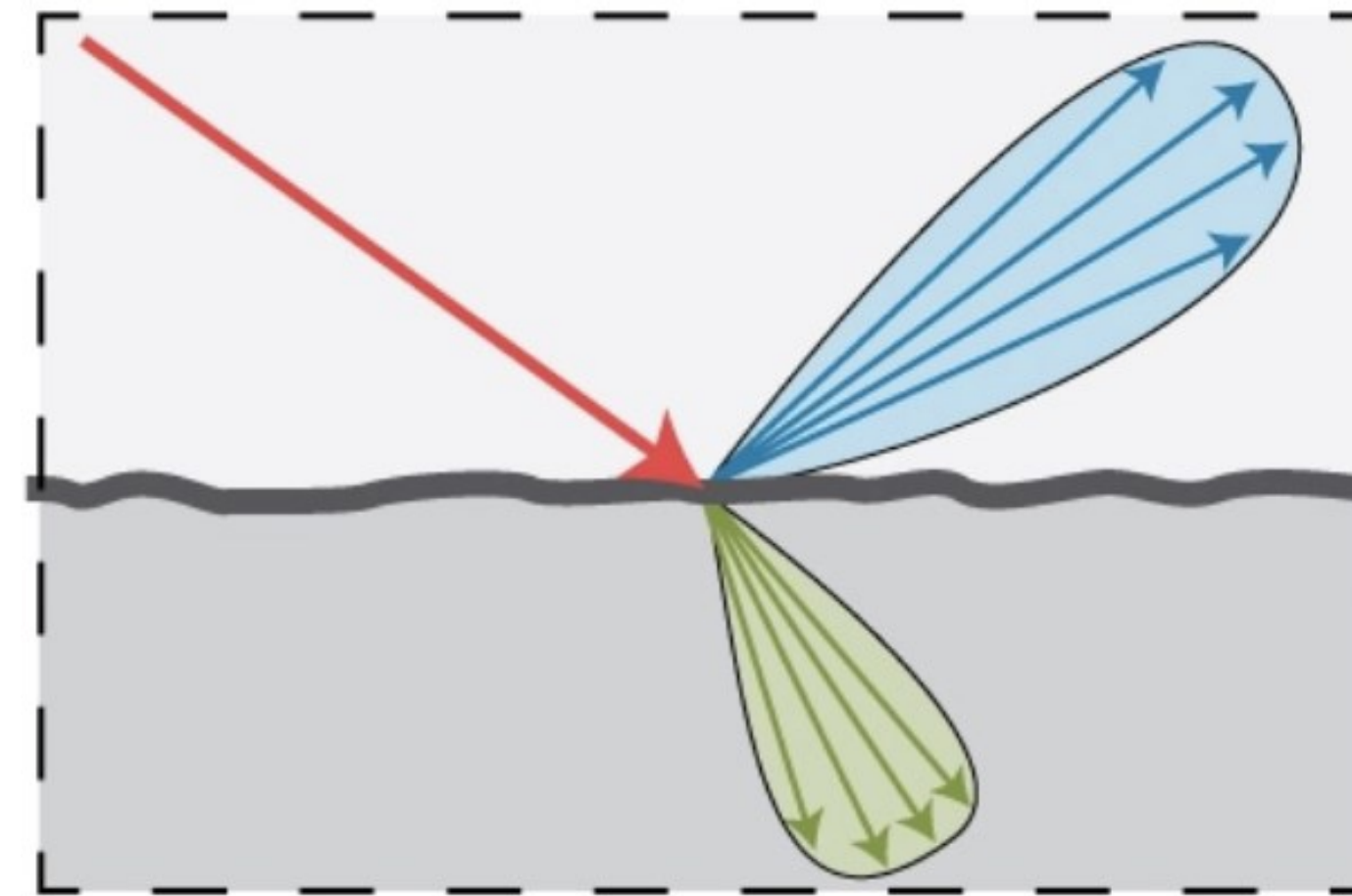
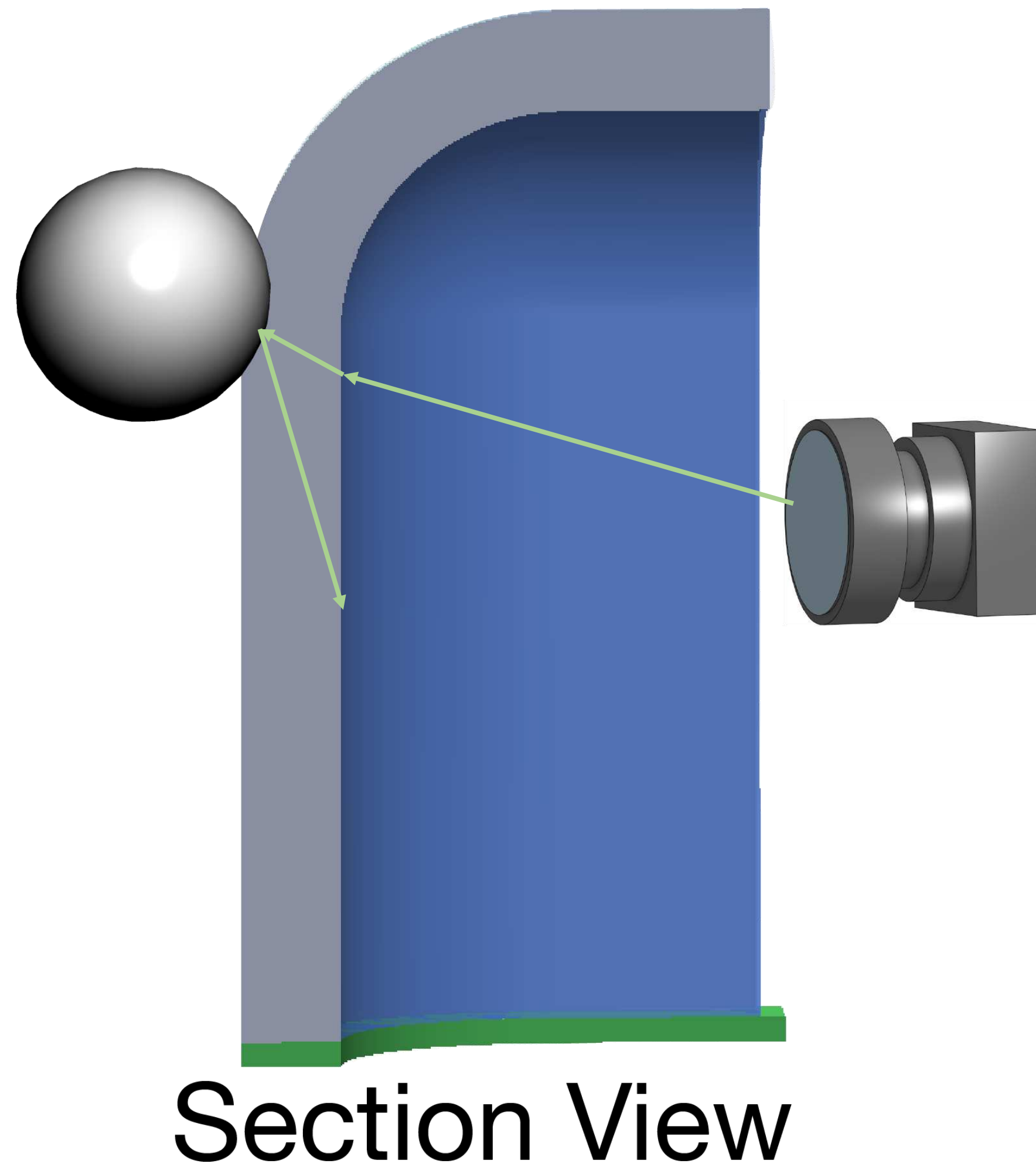


Refraction at rough interface

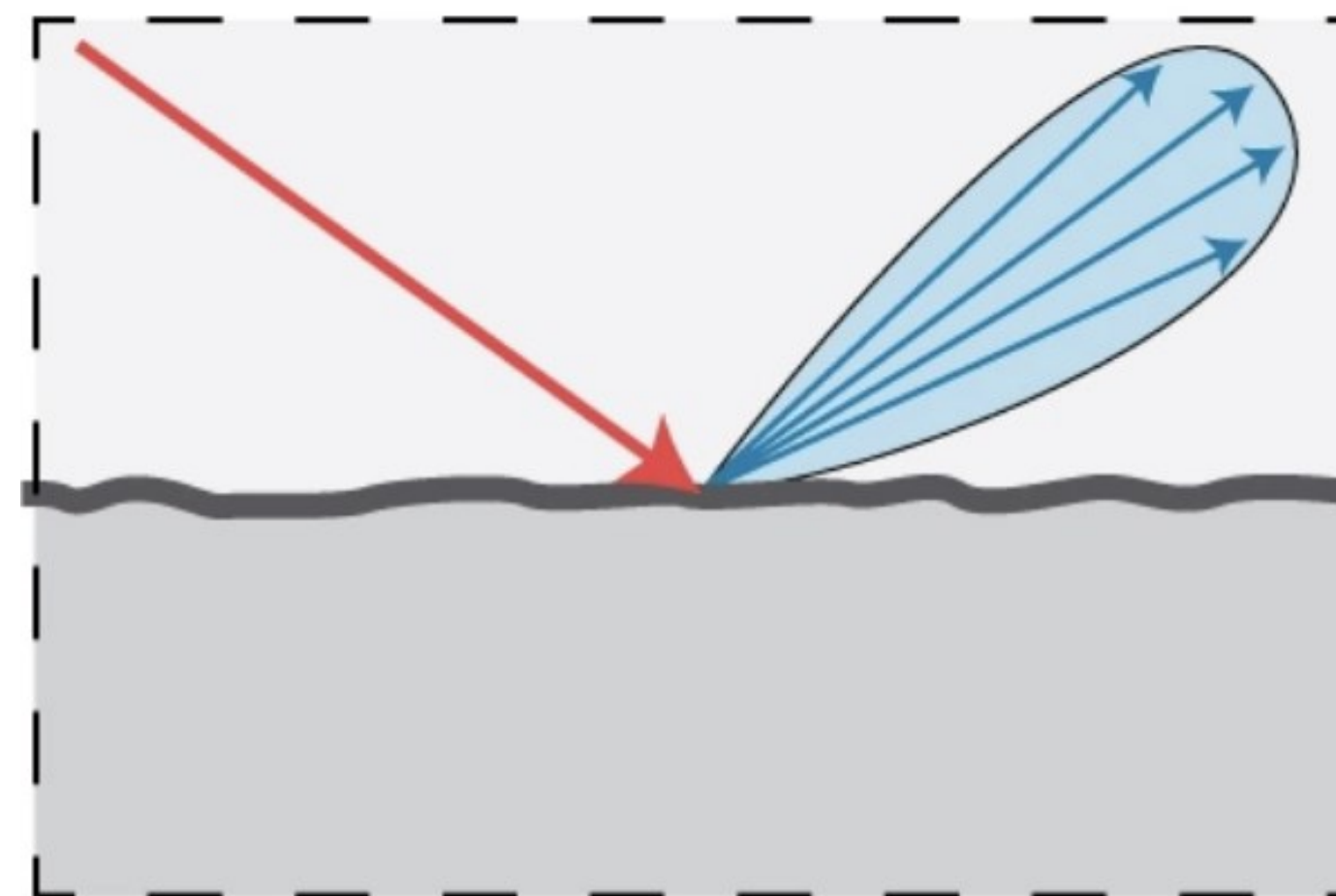
Curved tactile sensor: working principle



Curved tactile sensor: working principle

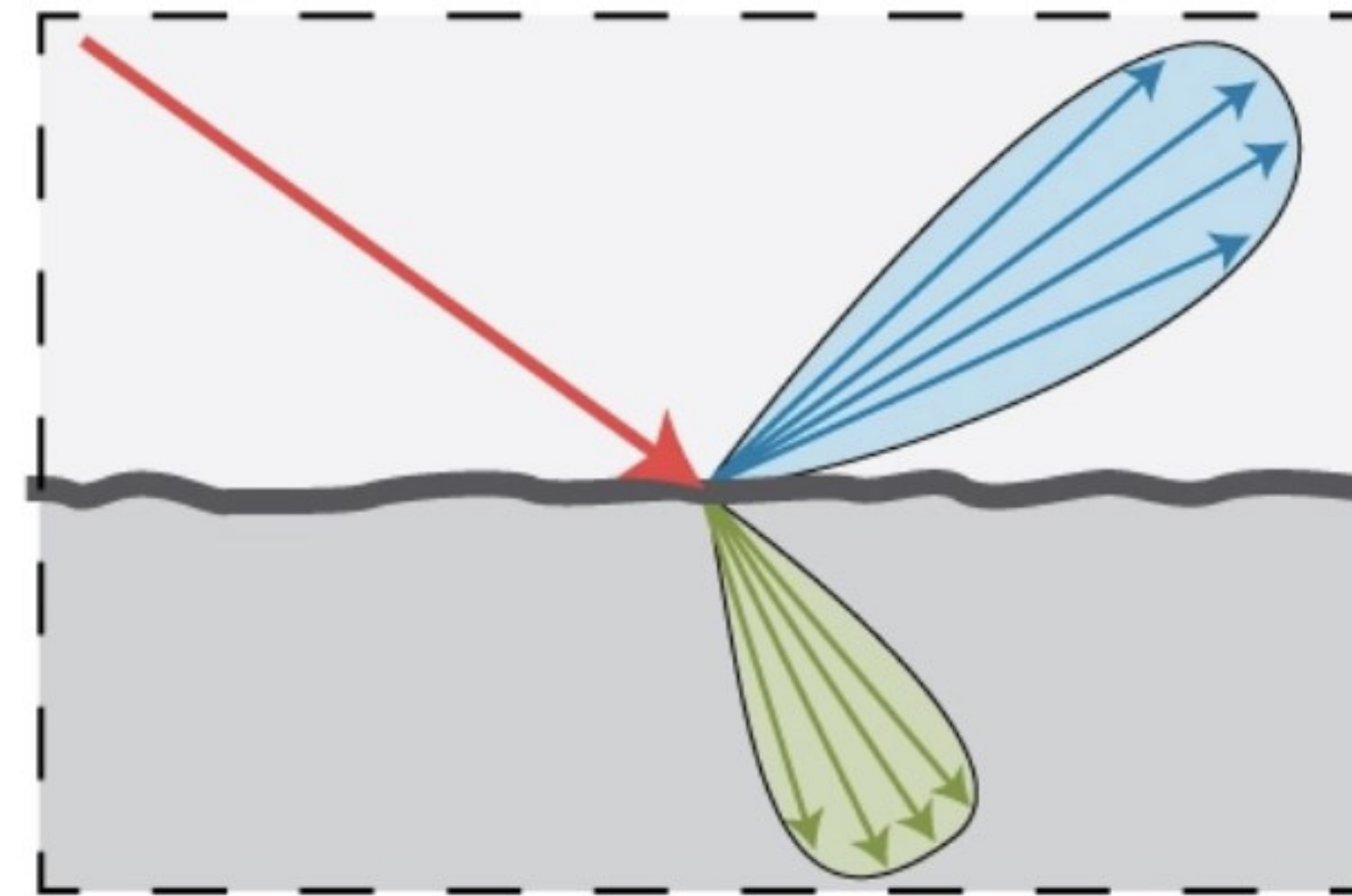
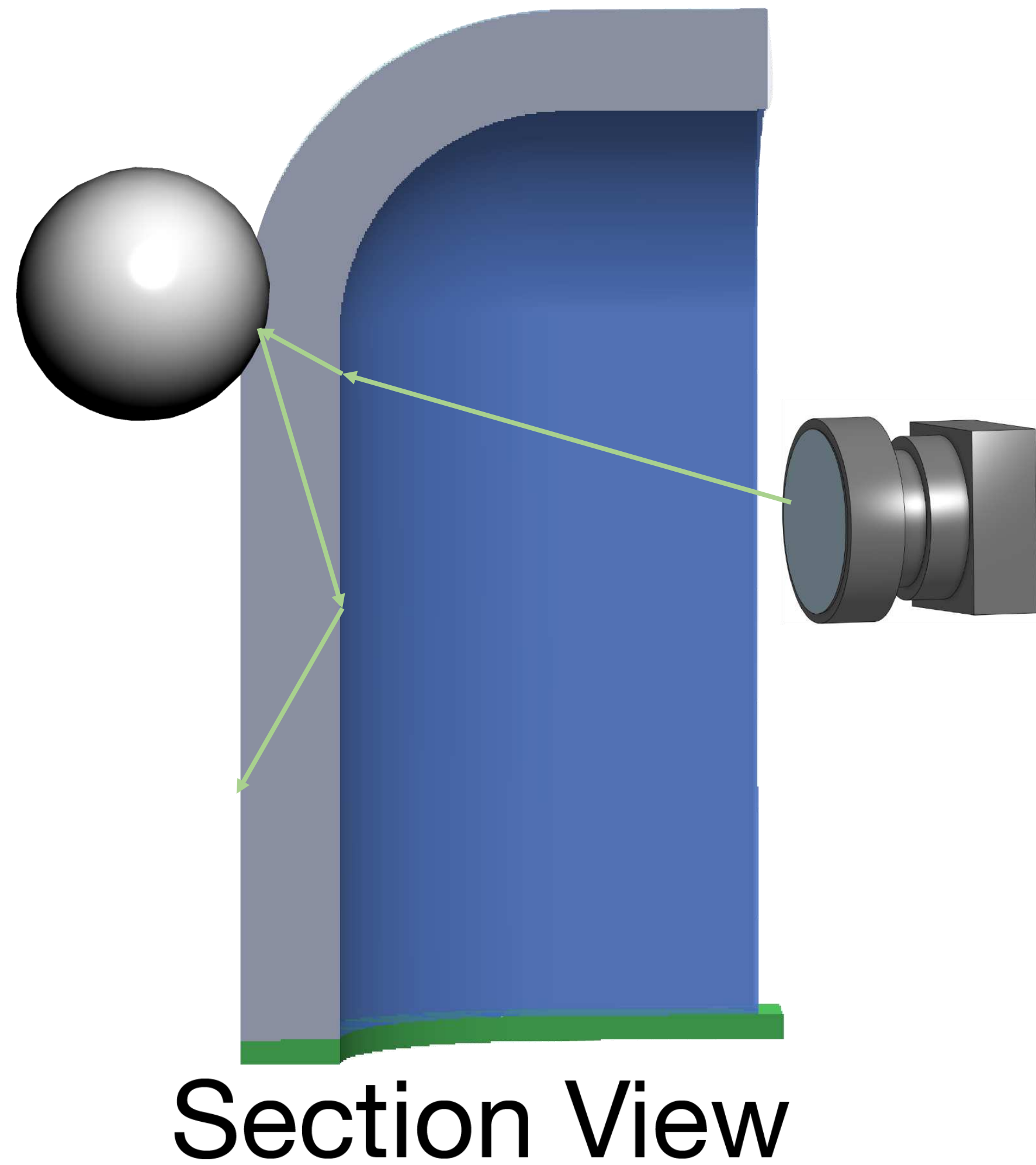


Refraction at rough interface

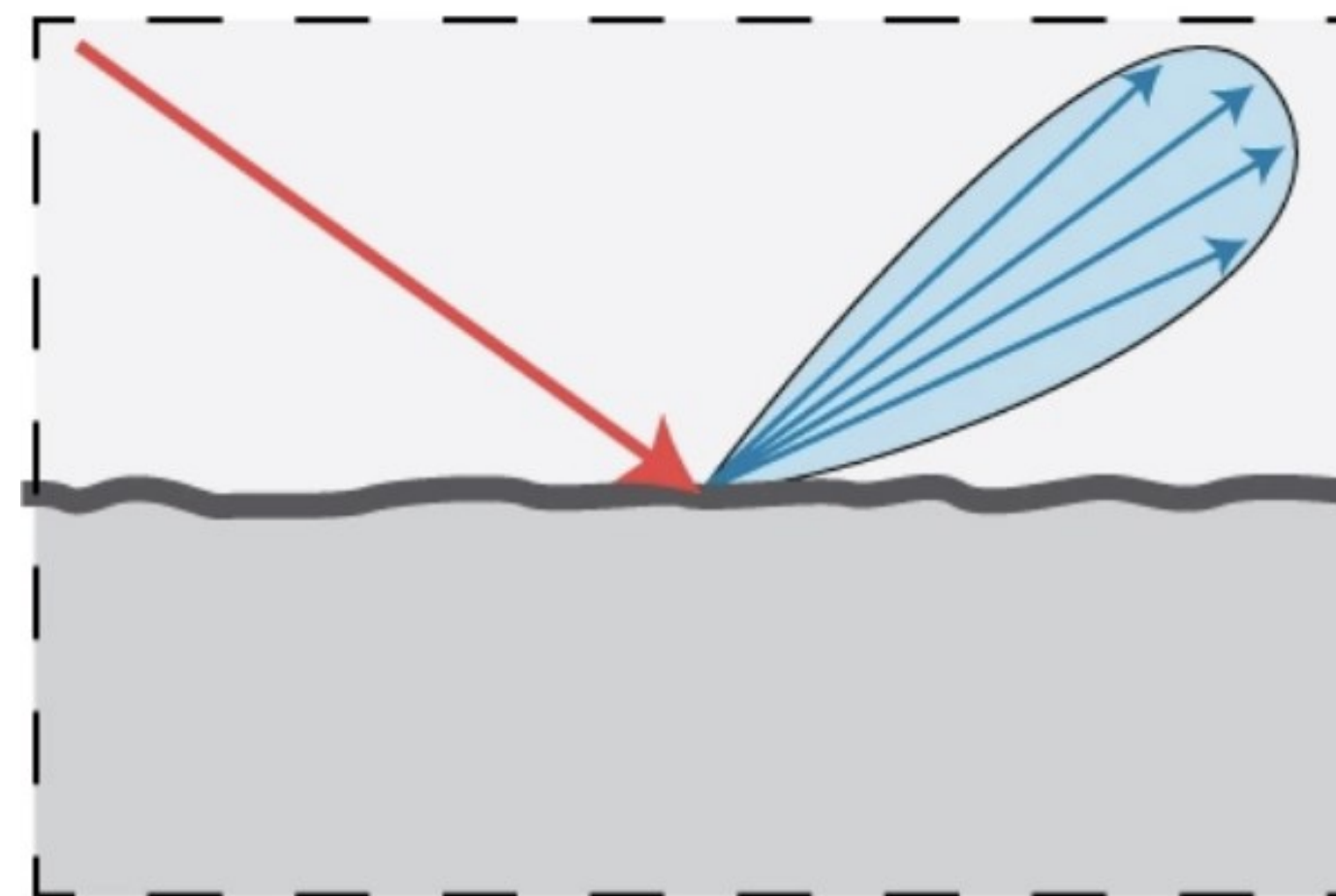


Reflection at glossy surfaces

Curved tactile sensor: working principle

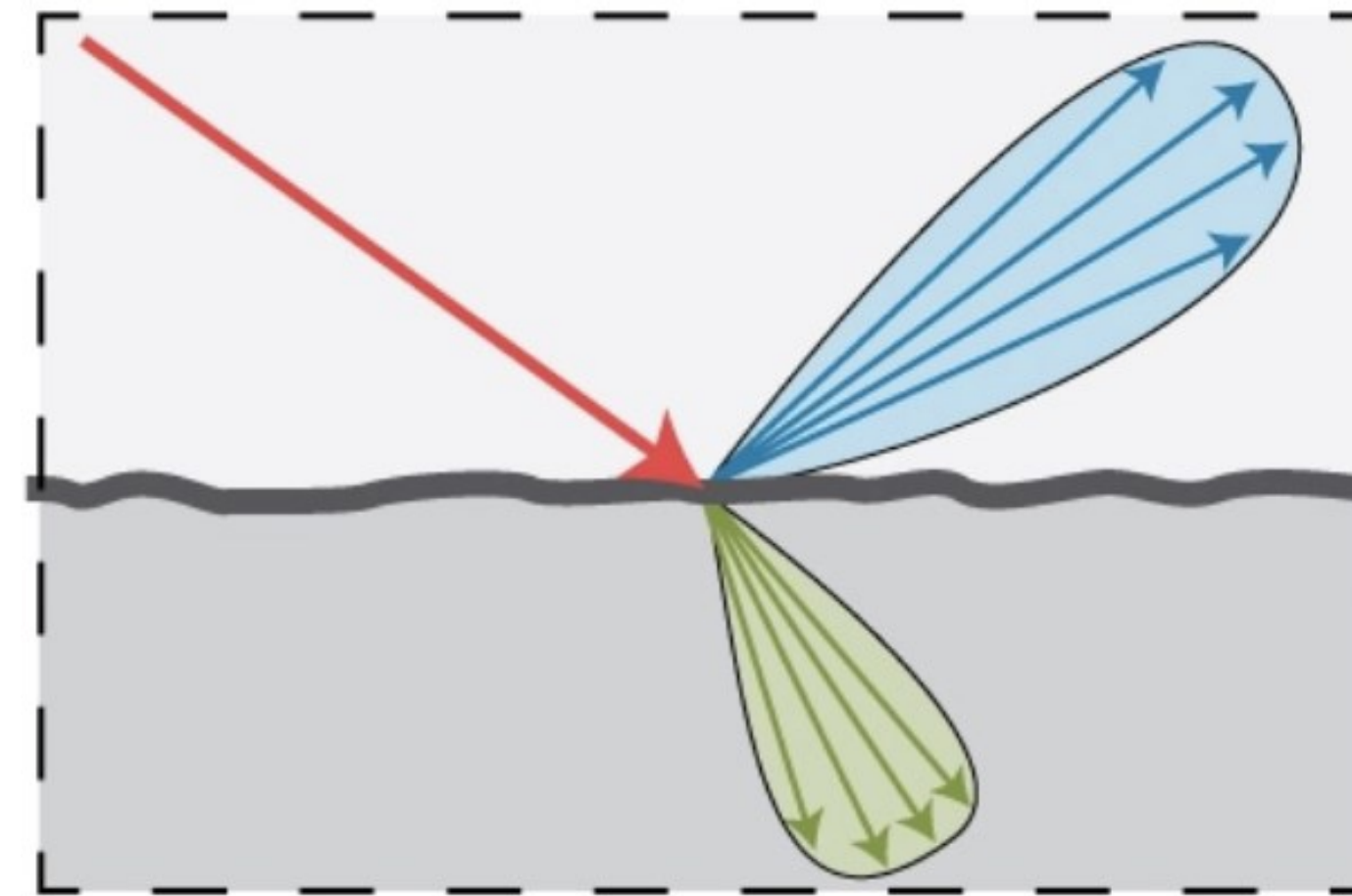
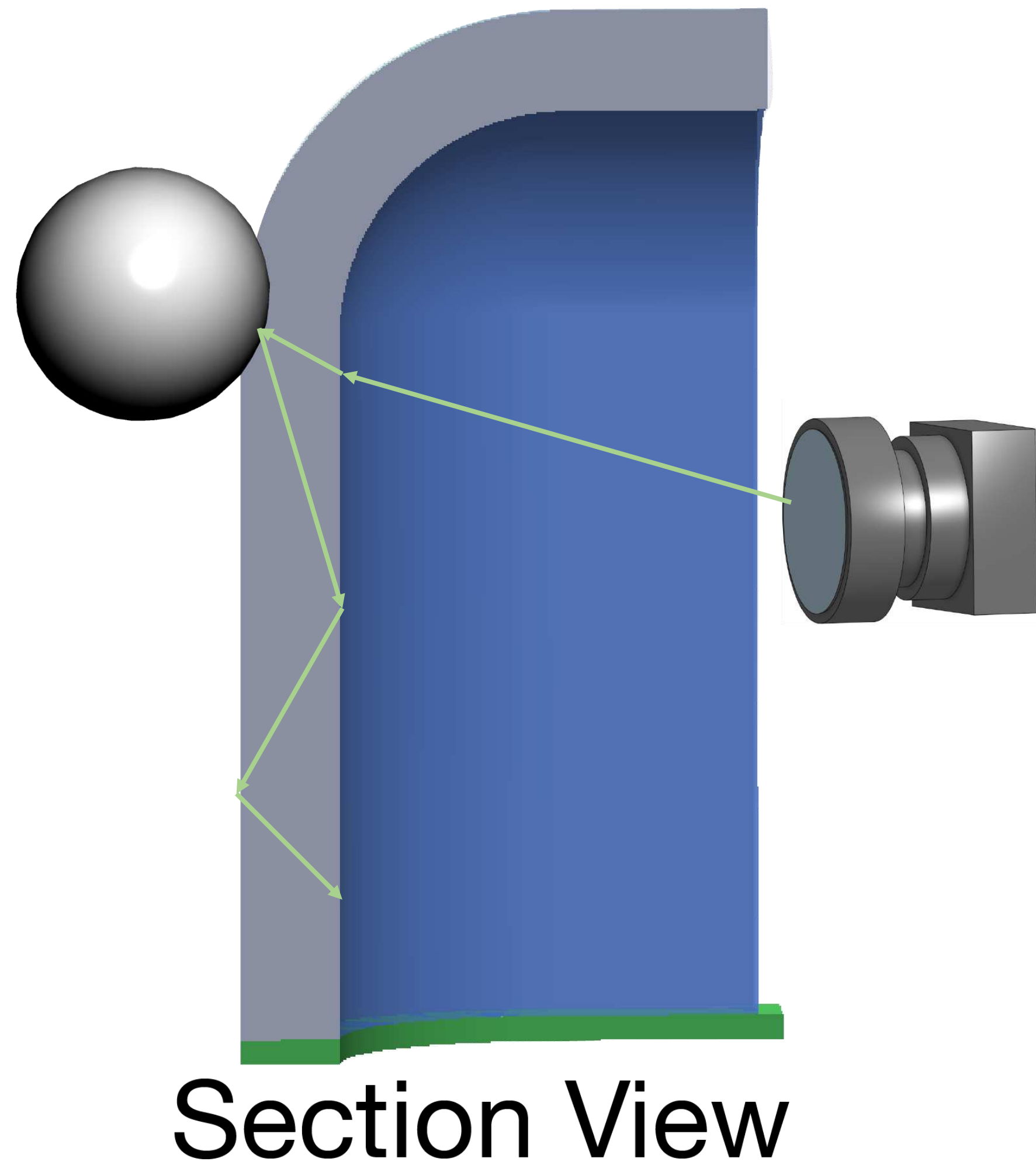


Refraction at rough interface

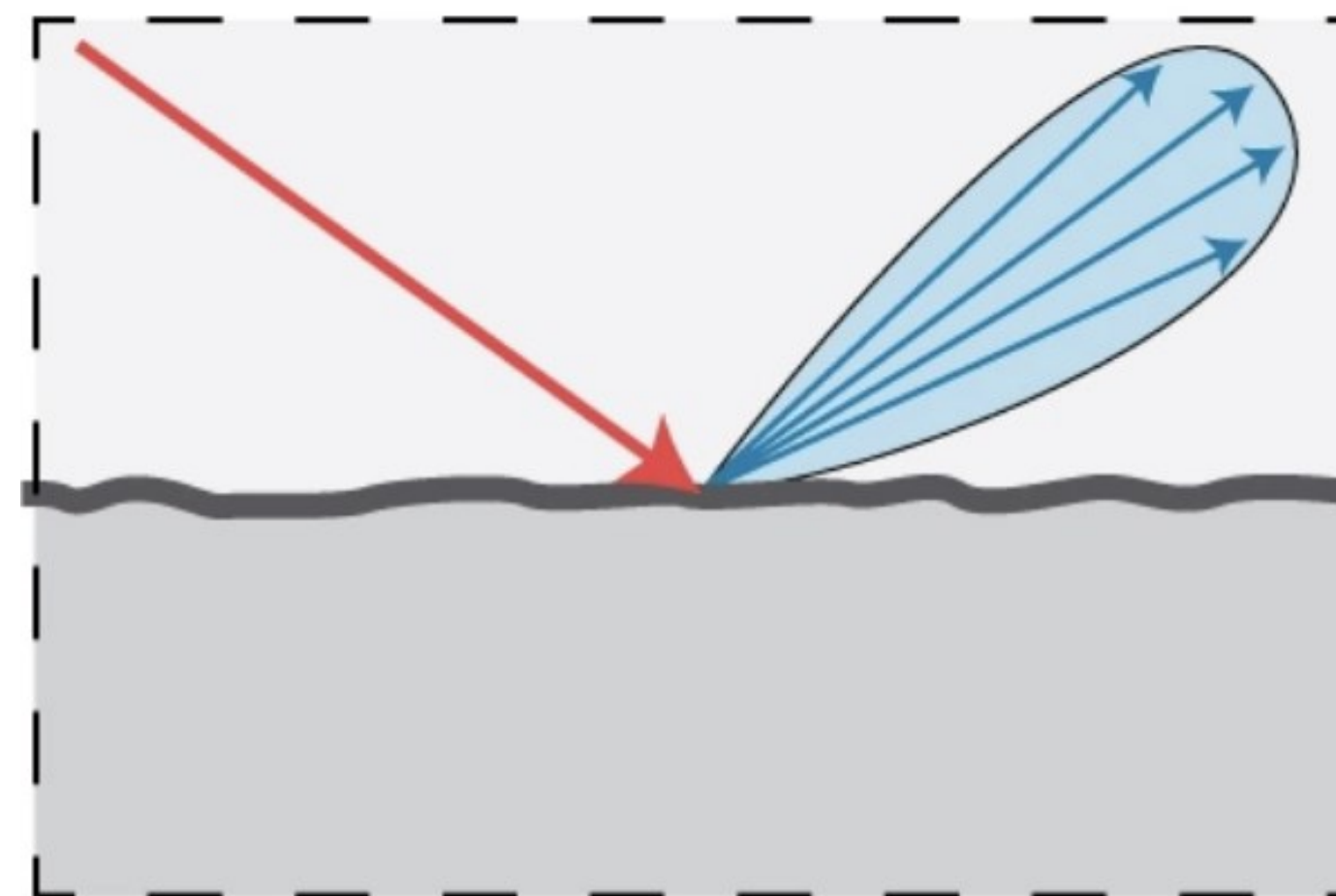


Reflection at glossy surfaces

Curved tactile sensor: working principle

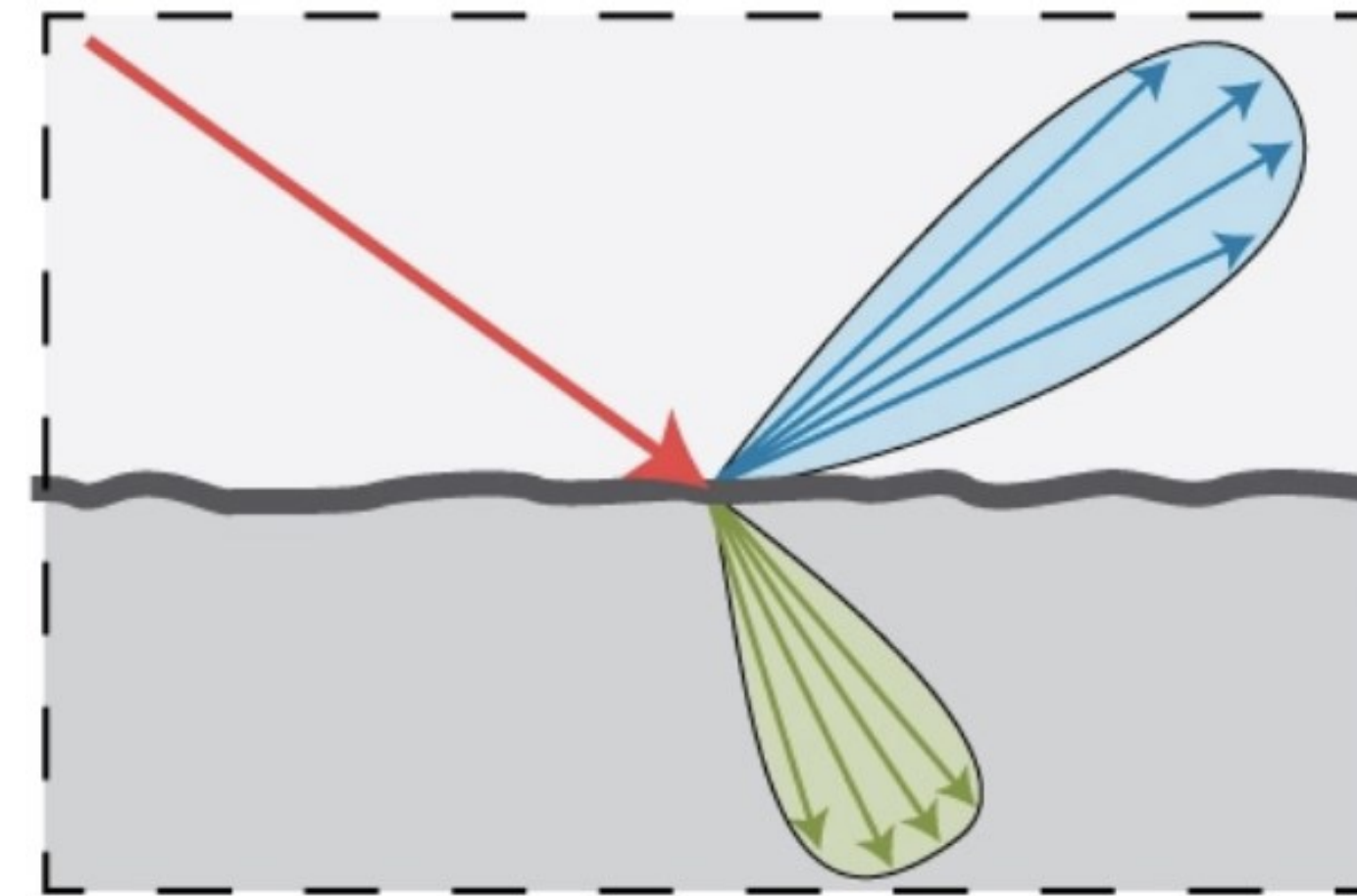
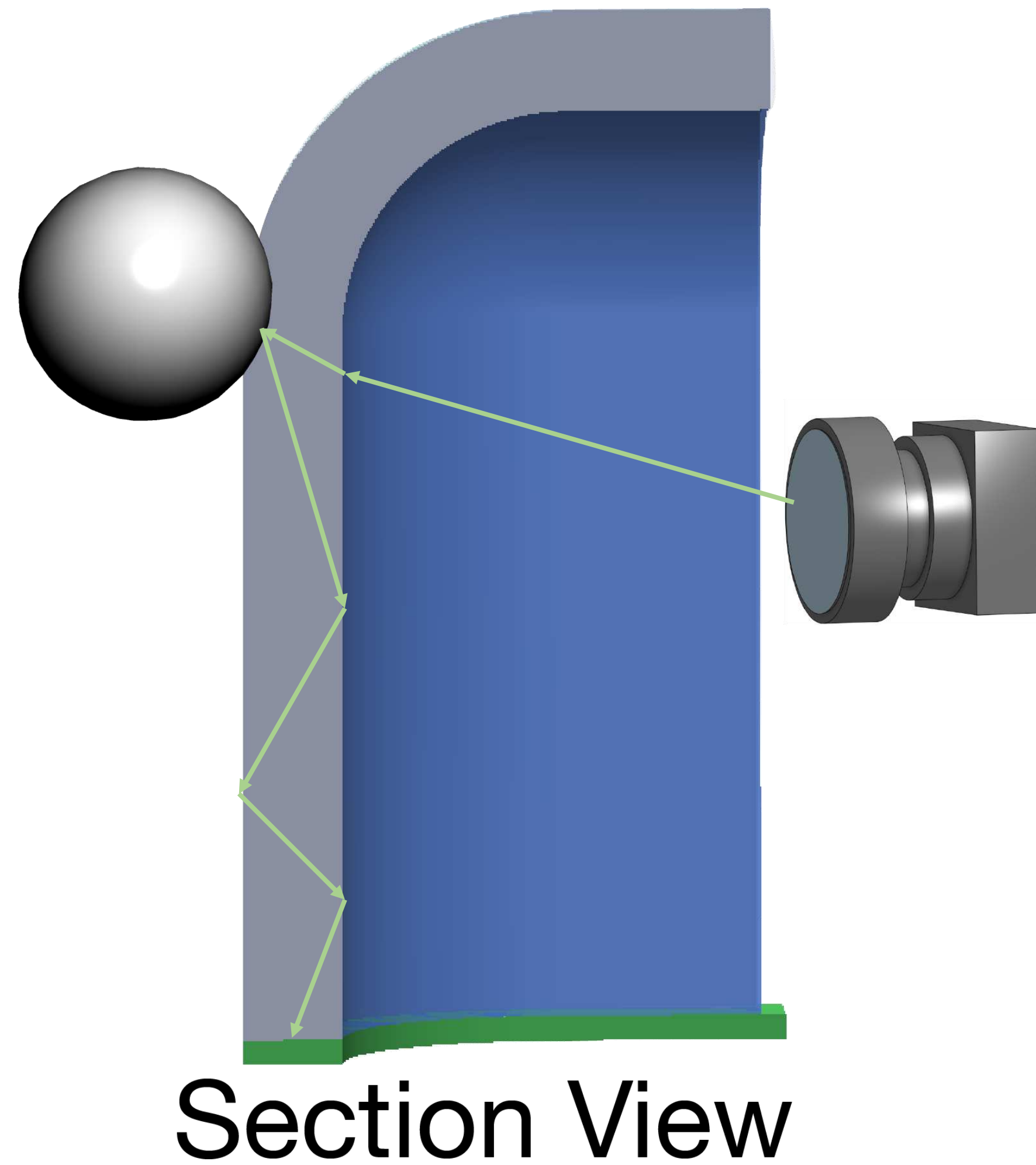


Refraction at rough interface

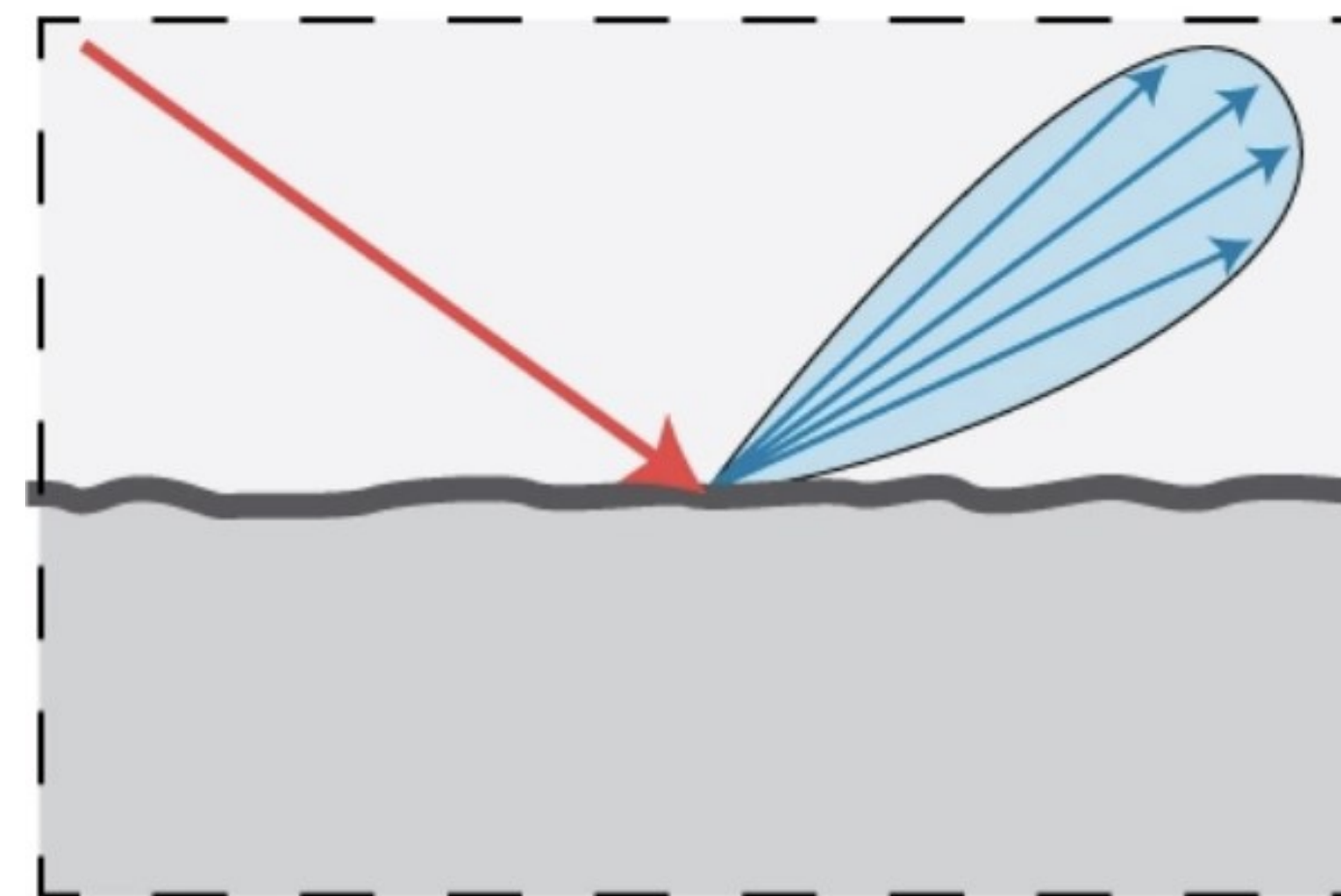


Reflection at glossy surfaces

Curved tactile sensor: working principle



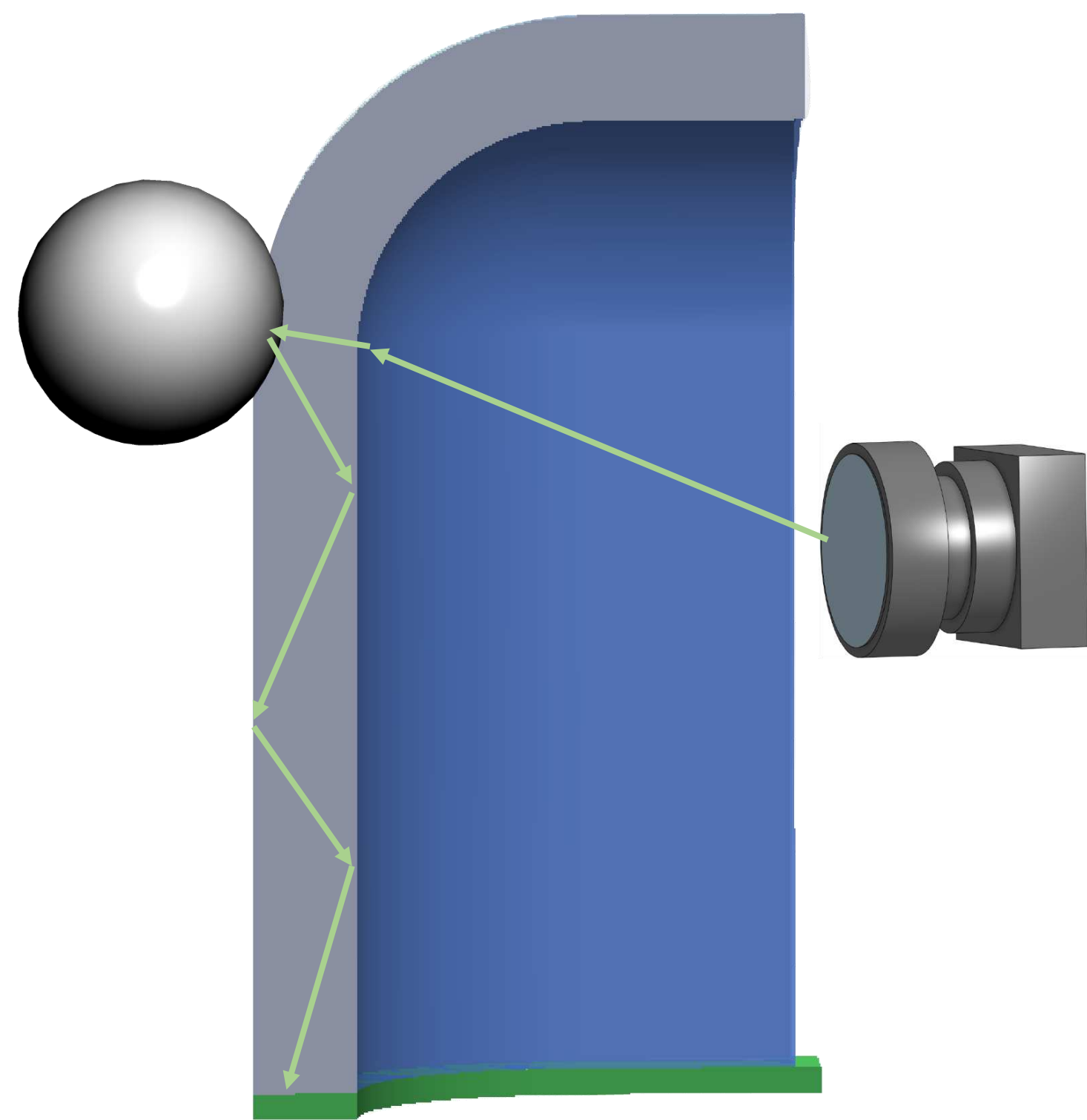
Refraction at rough interface



Reflection at glossy surfaces

Sensor design framework: optical simulation

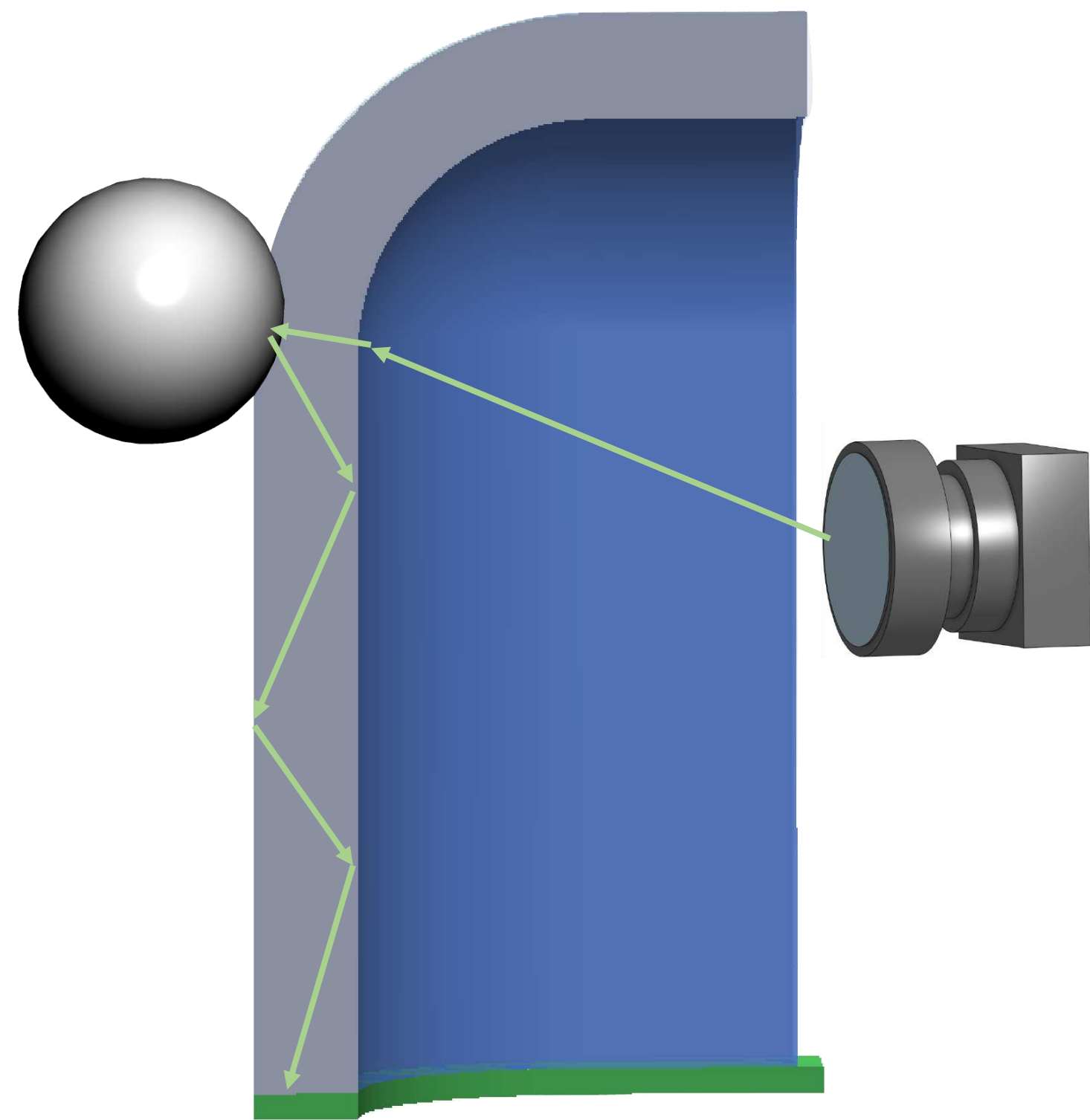
- Sensor geometry
- Material specifications
- Placement of camera and lights



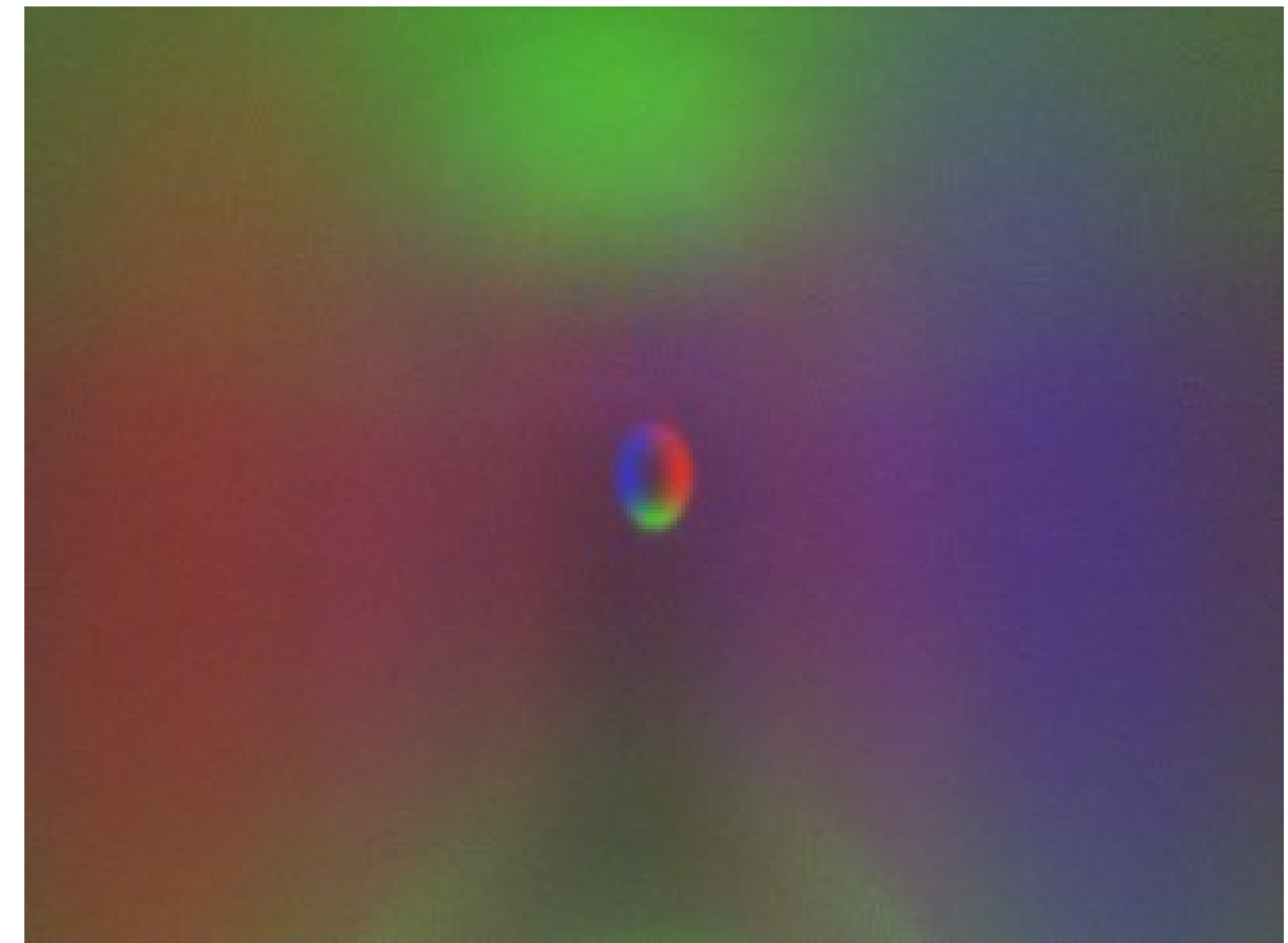
Section view

Sensor design framework: optical simulation

- Sensor geometry
- Material specifications
- Placement of camera and lights



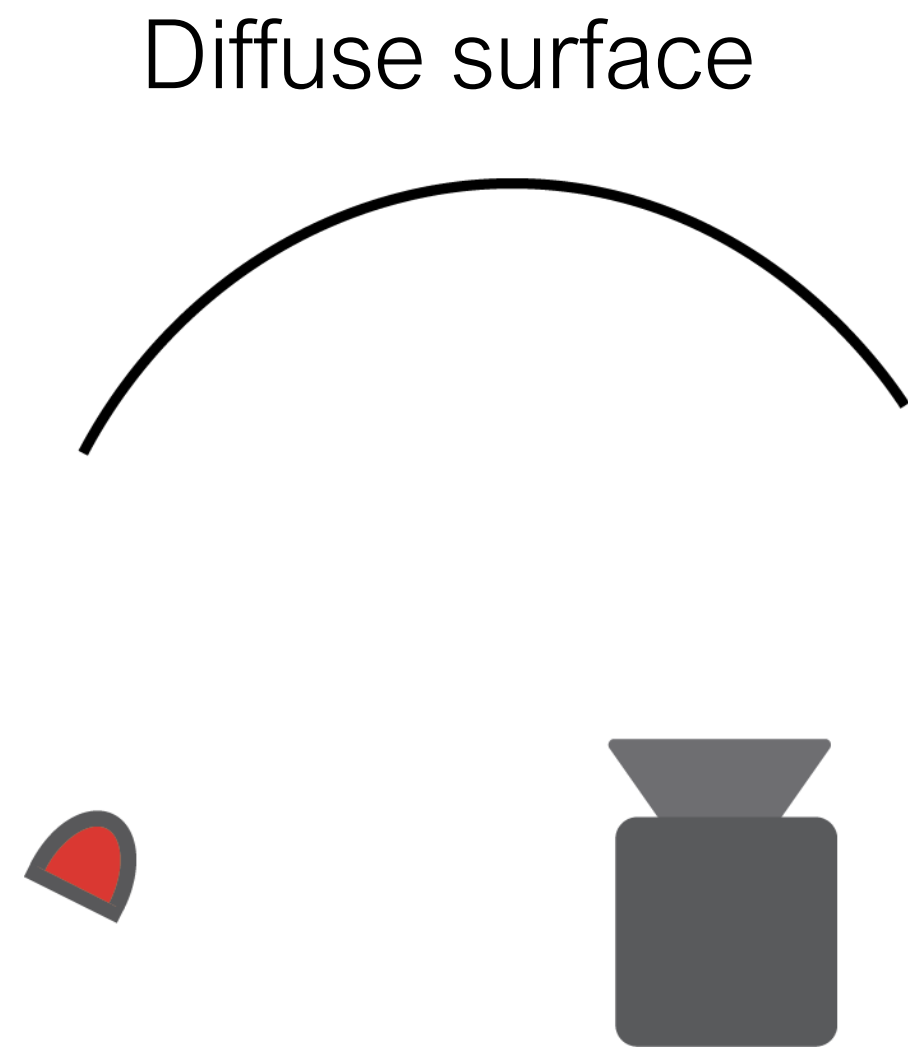
Section view



Tactile image [Agarwal et al., 2023]

Sensor design framework: optical simulation

- What makes simulation challenging

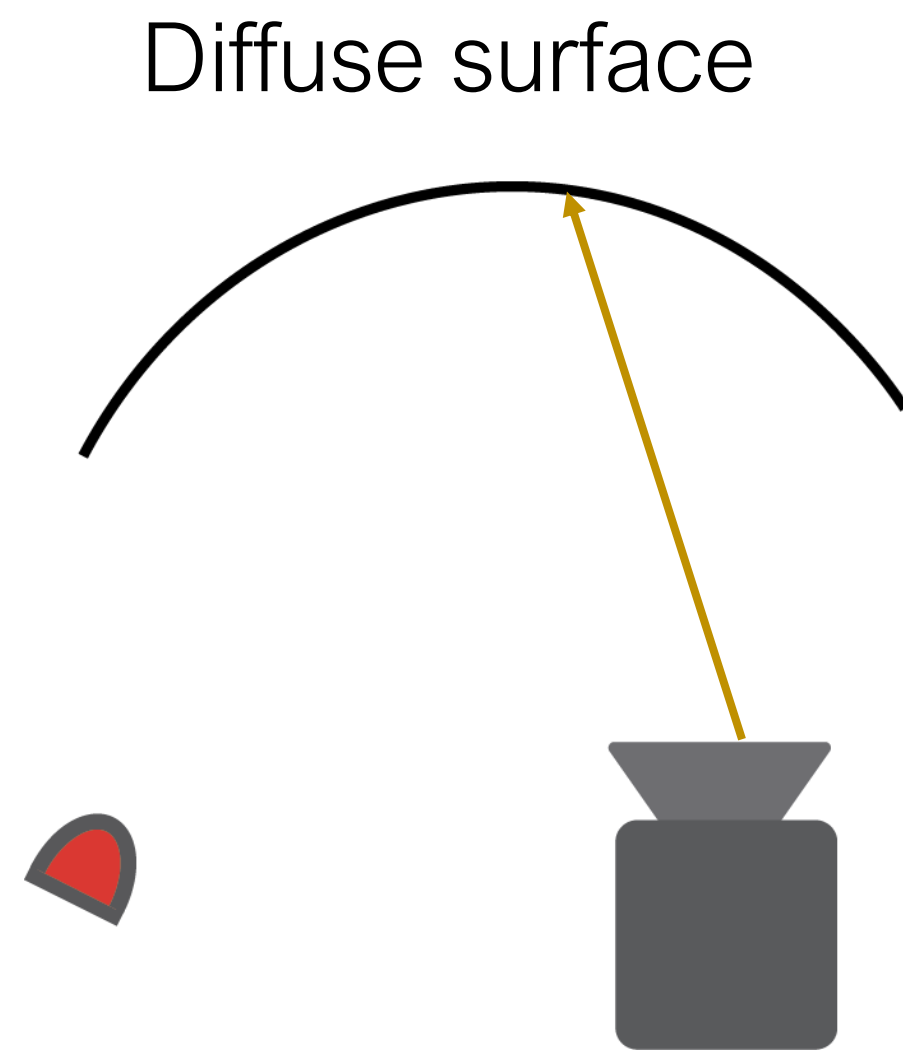


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

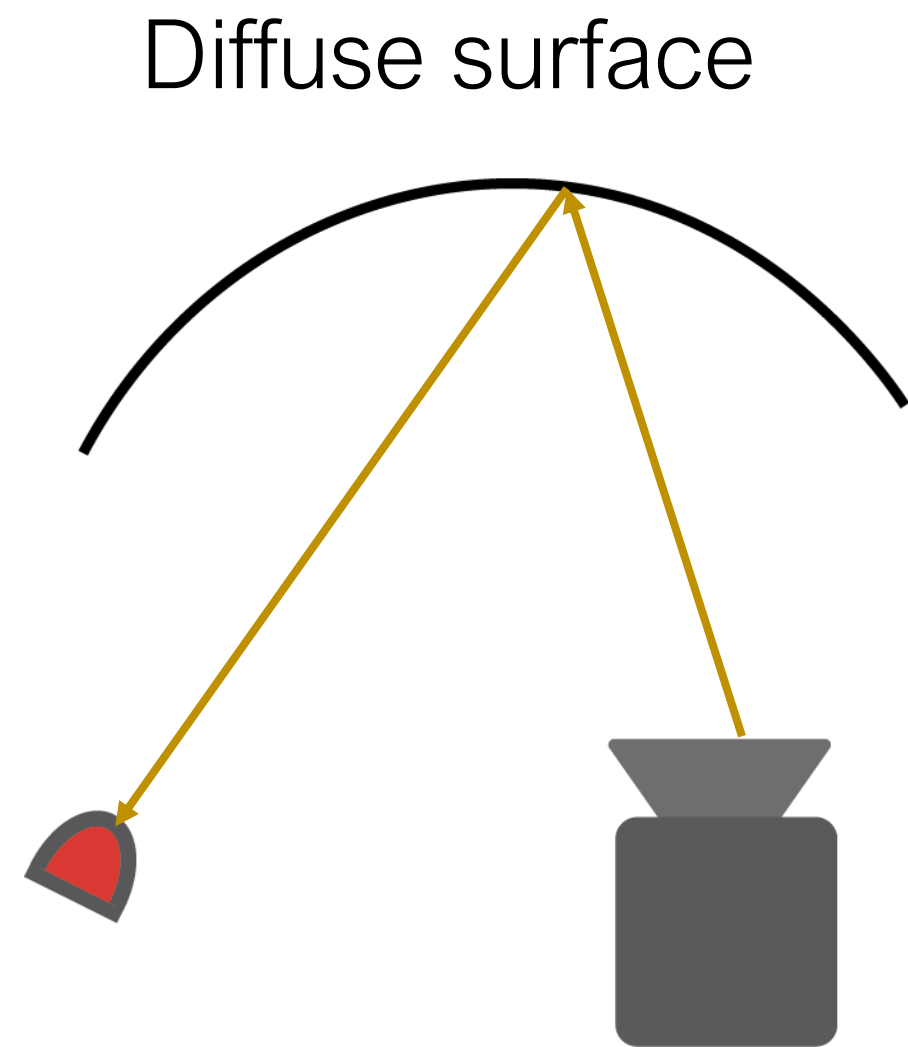


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

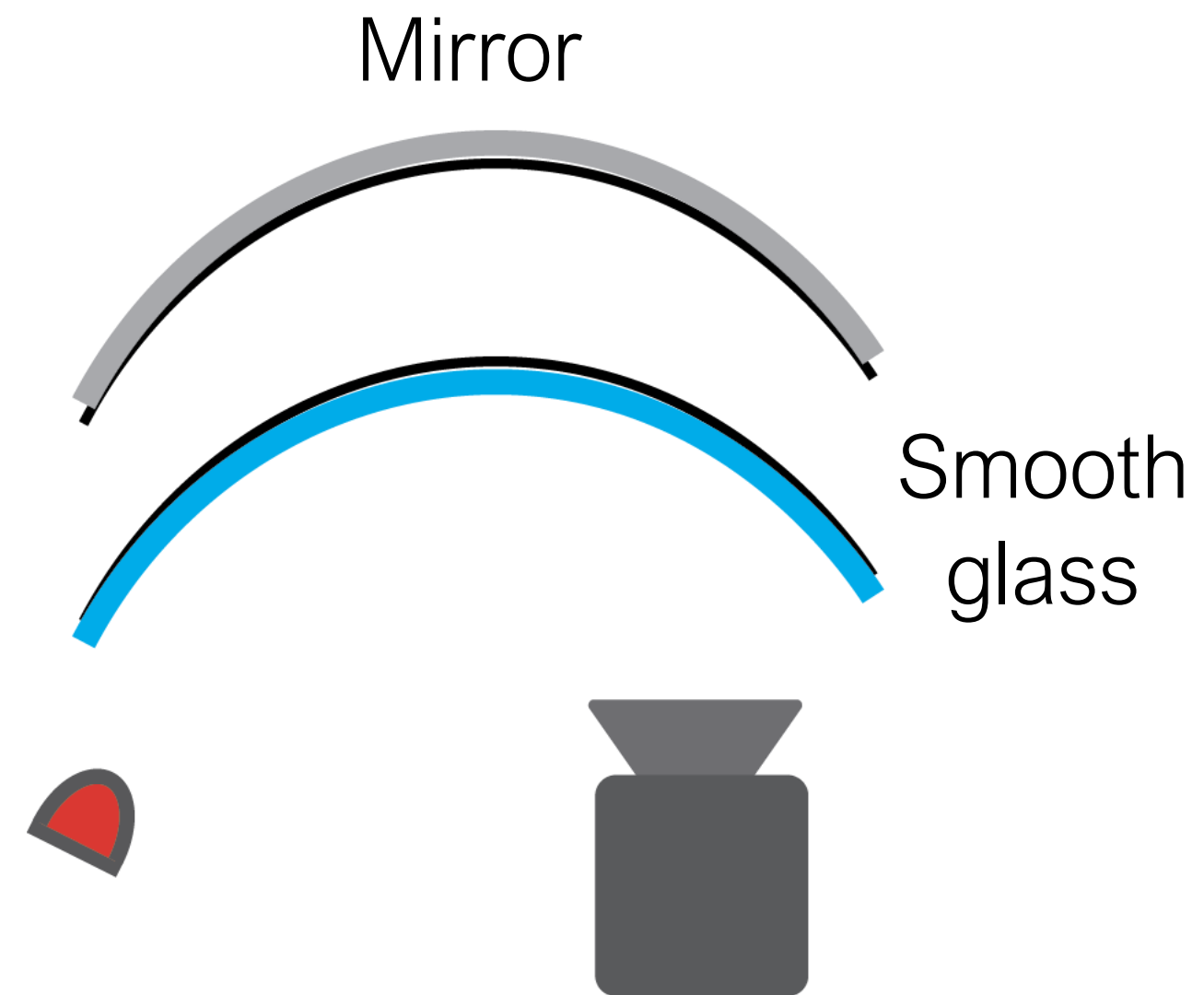
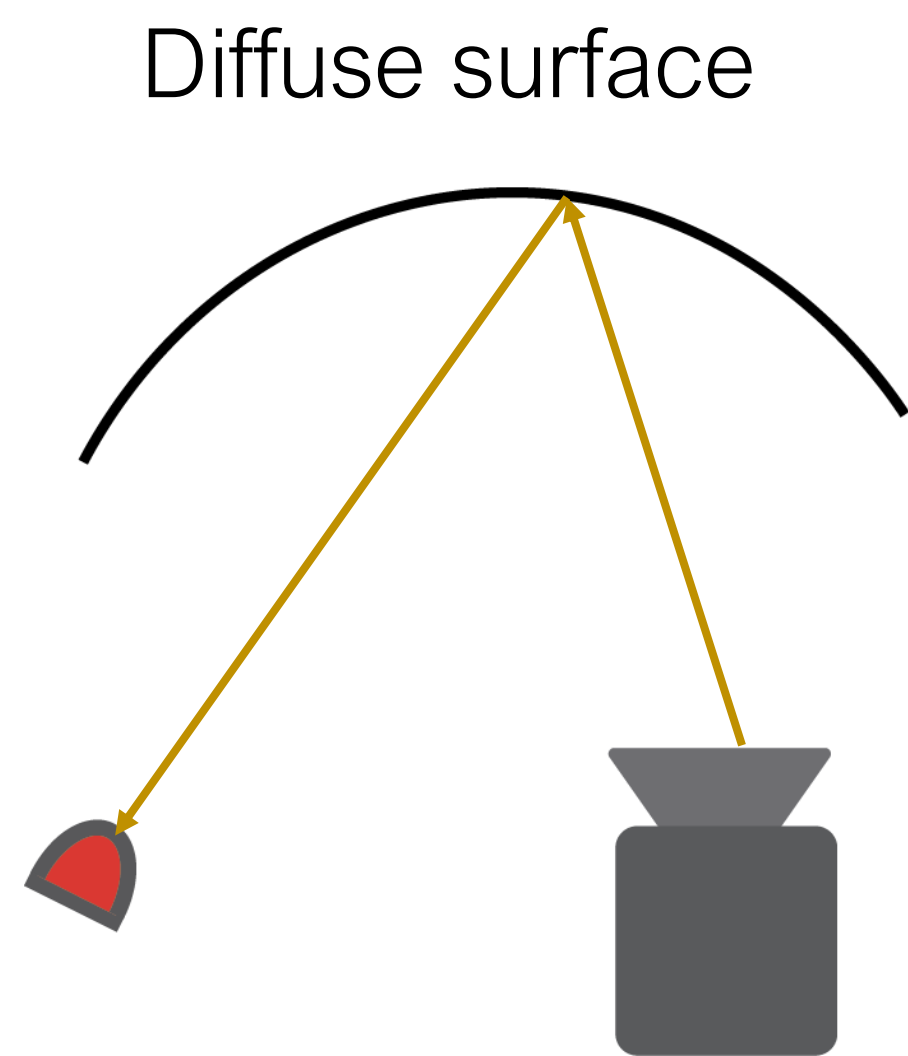


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

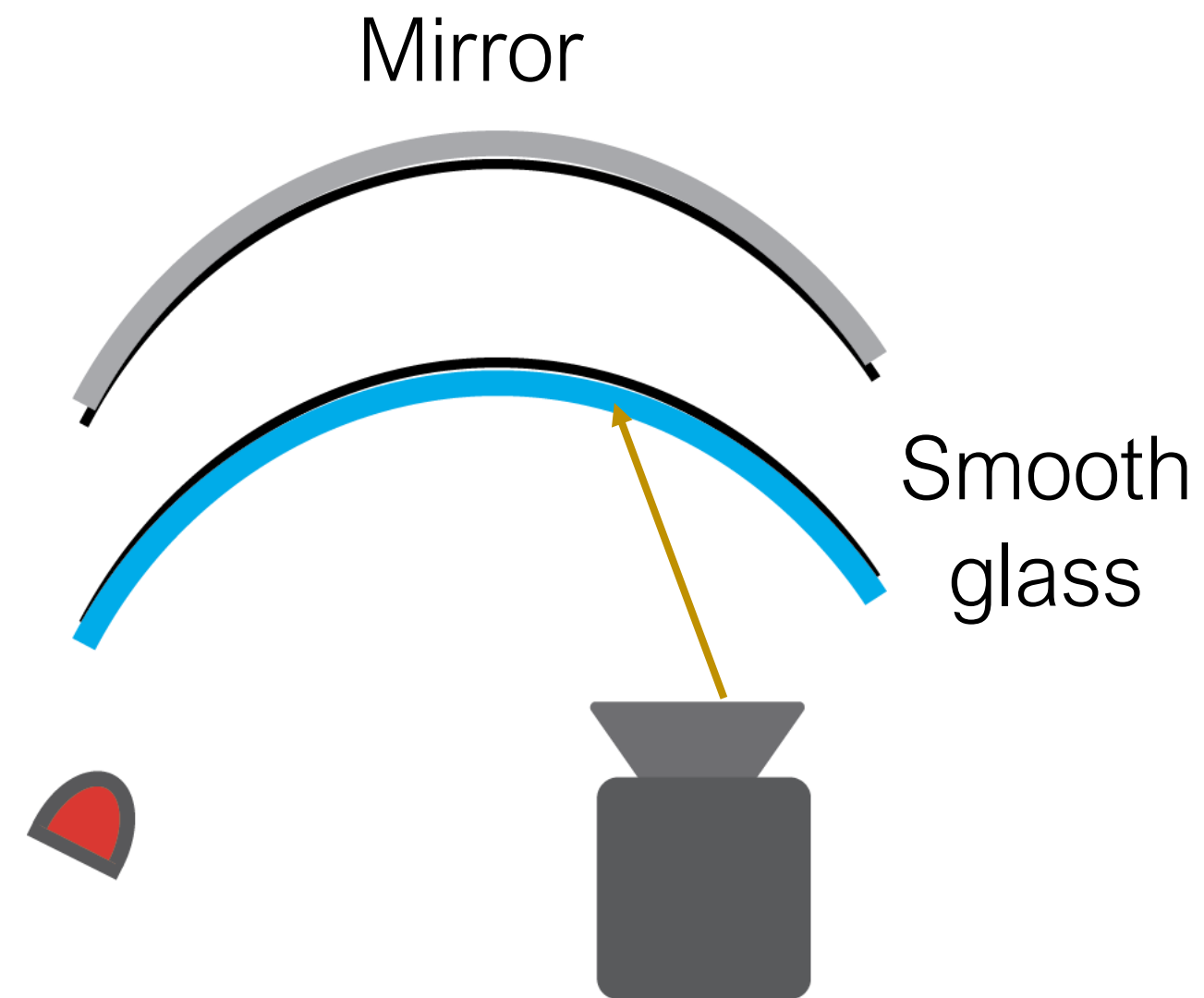
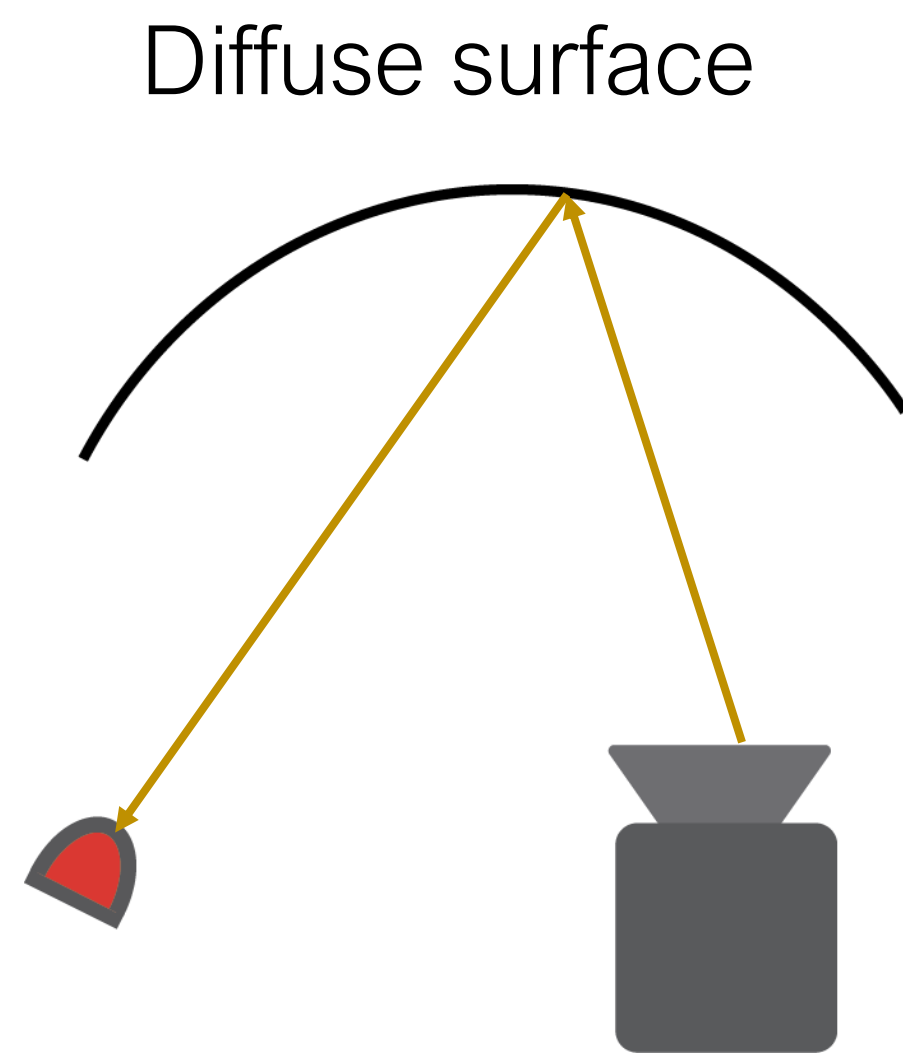


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

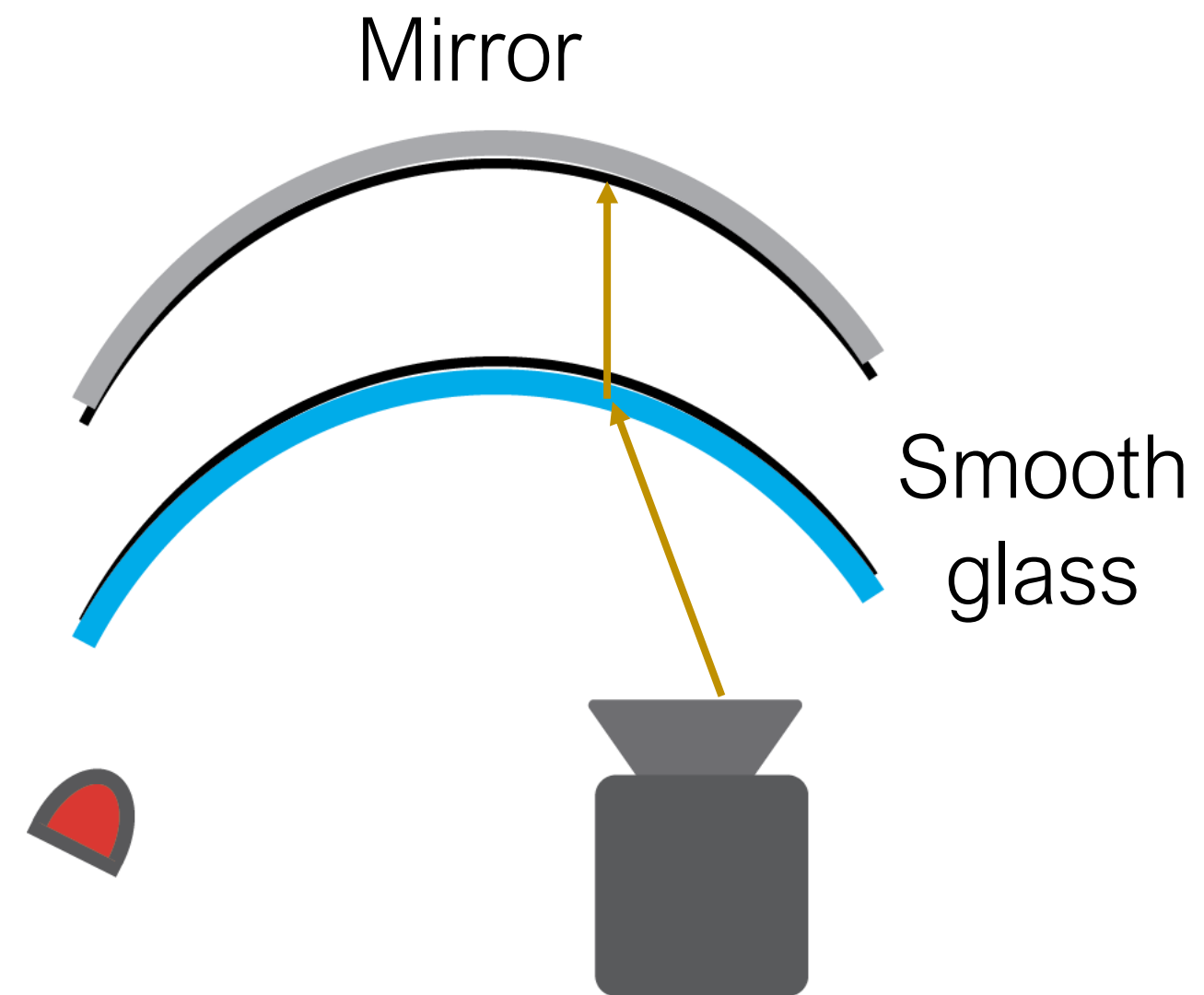
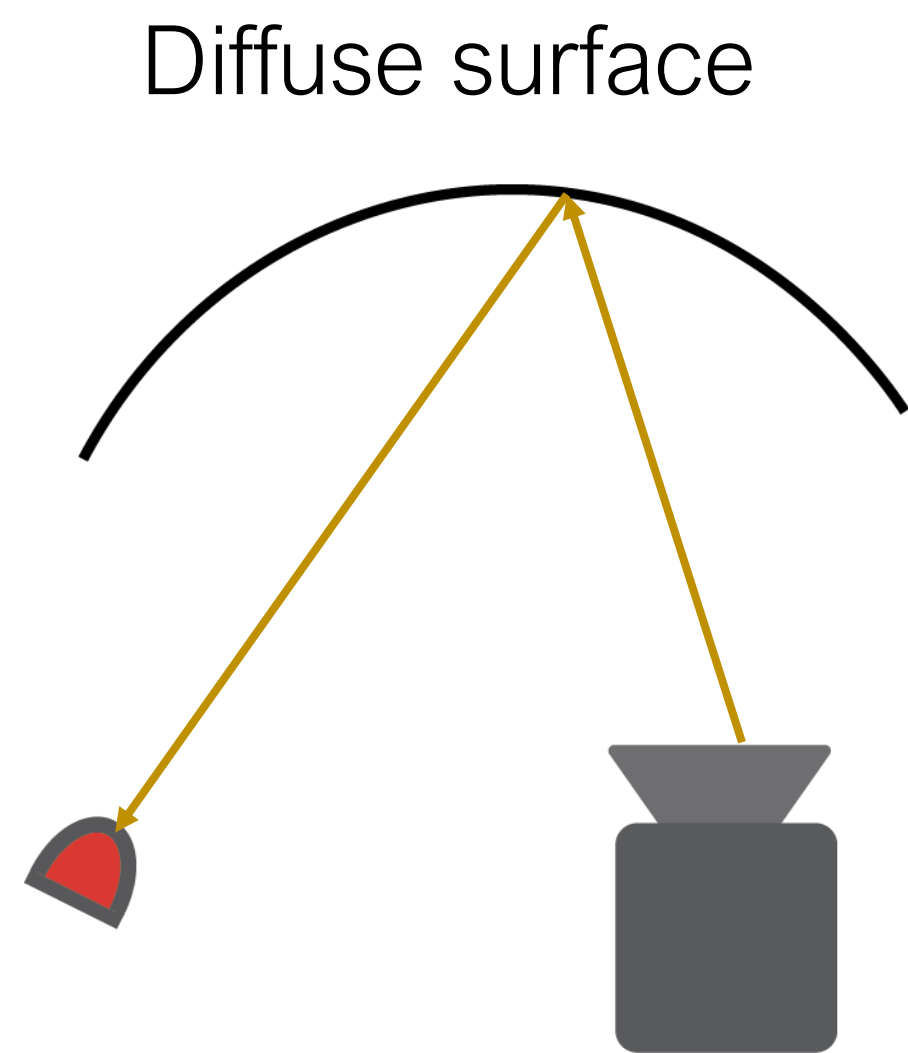


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

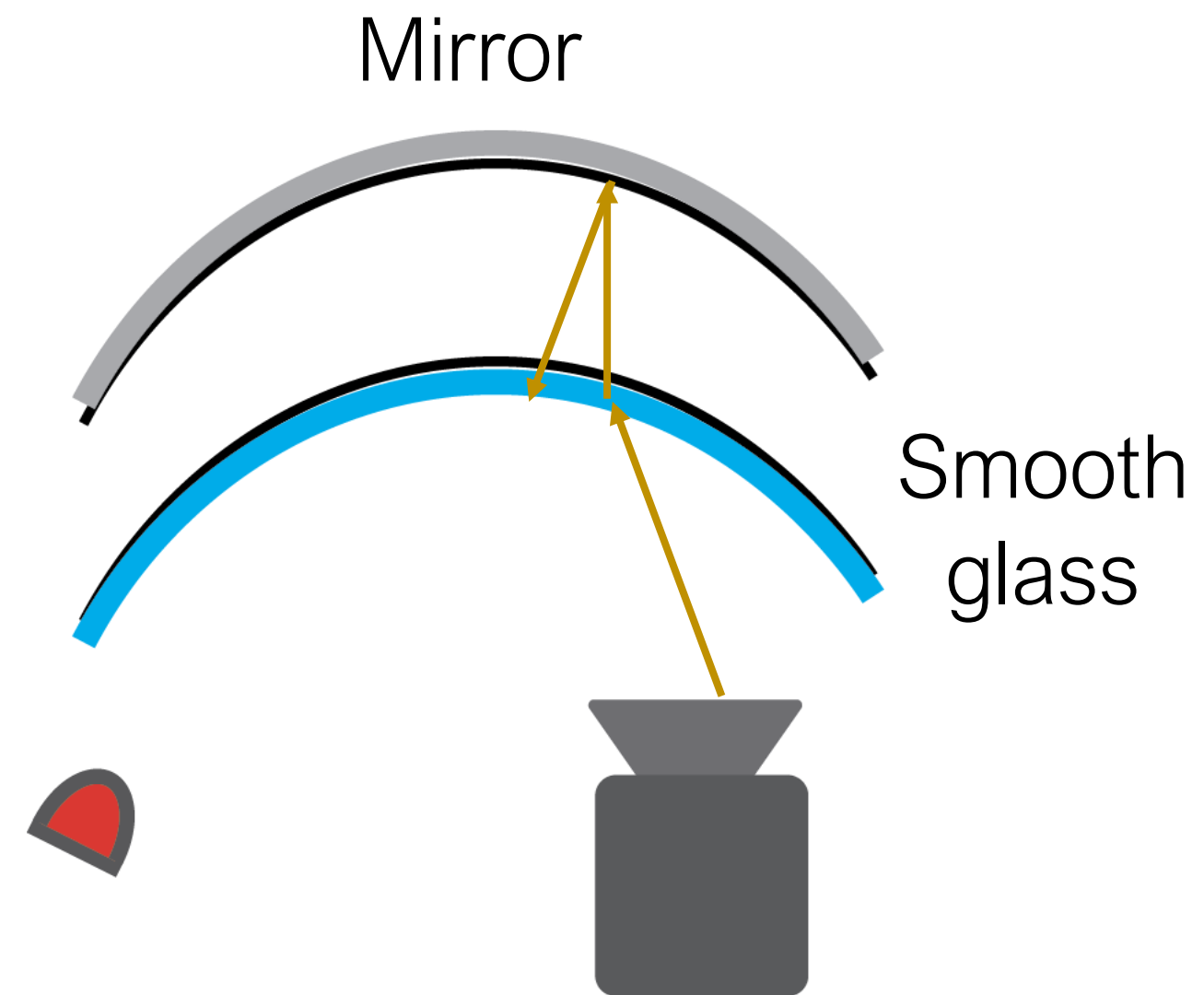
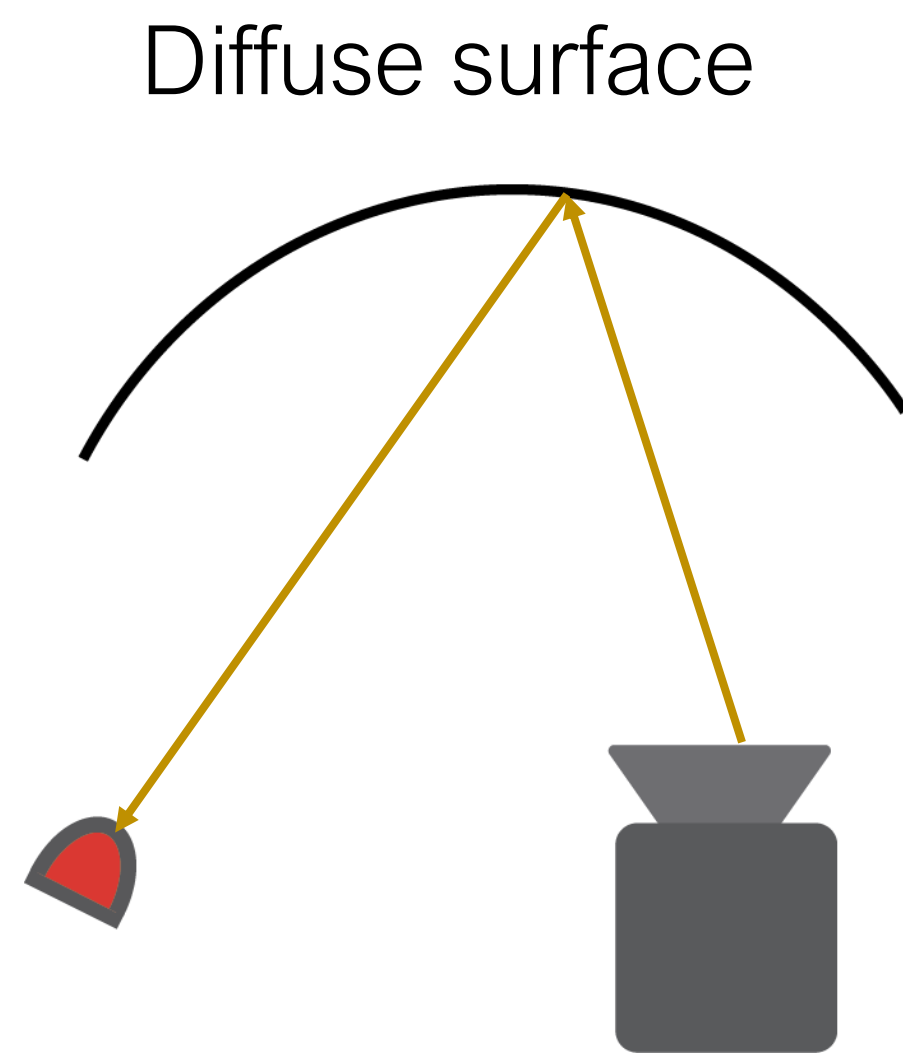


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

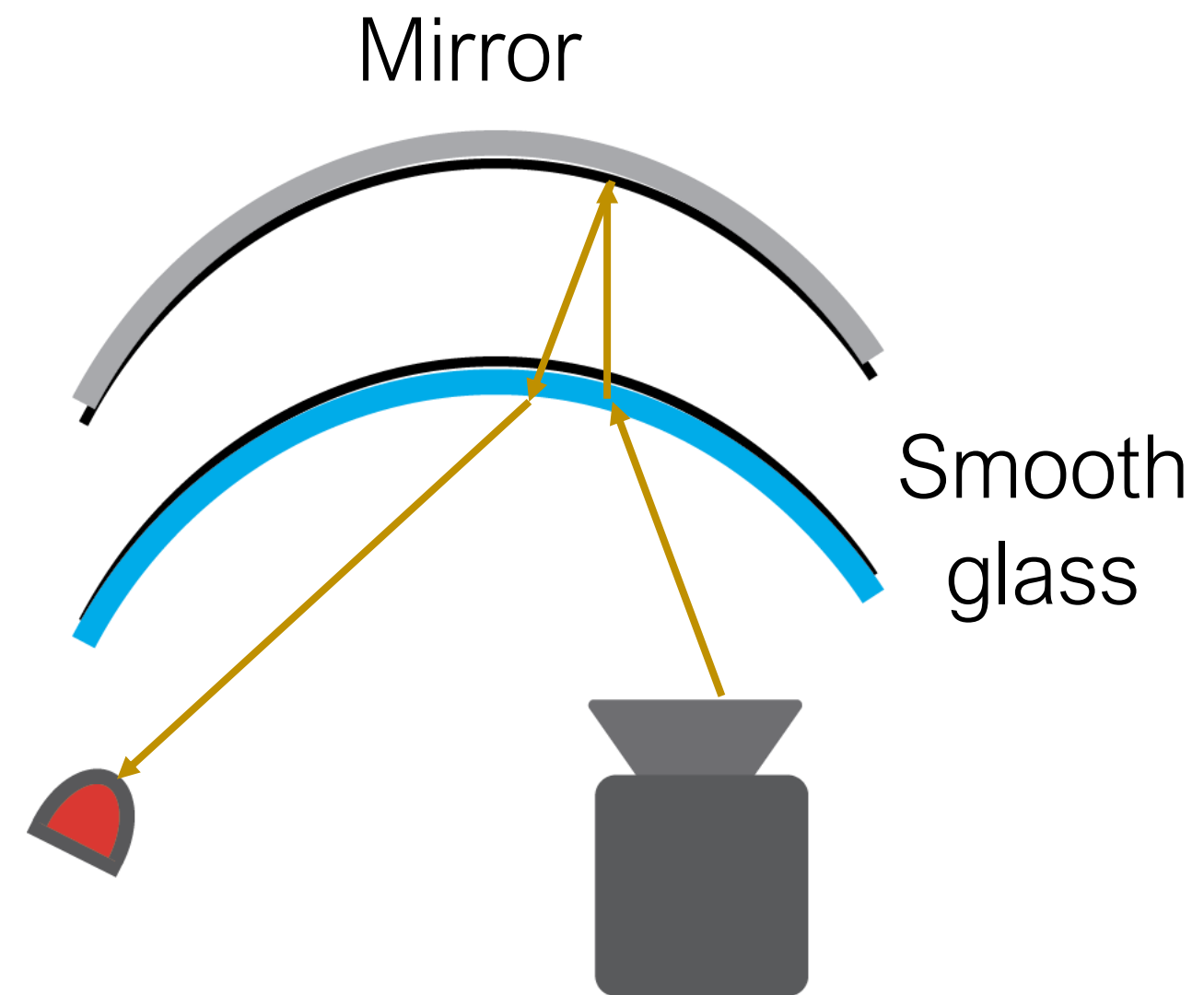
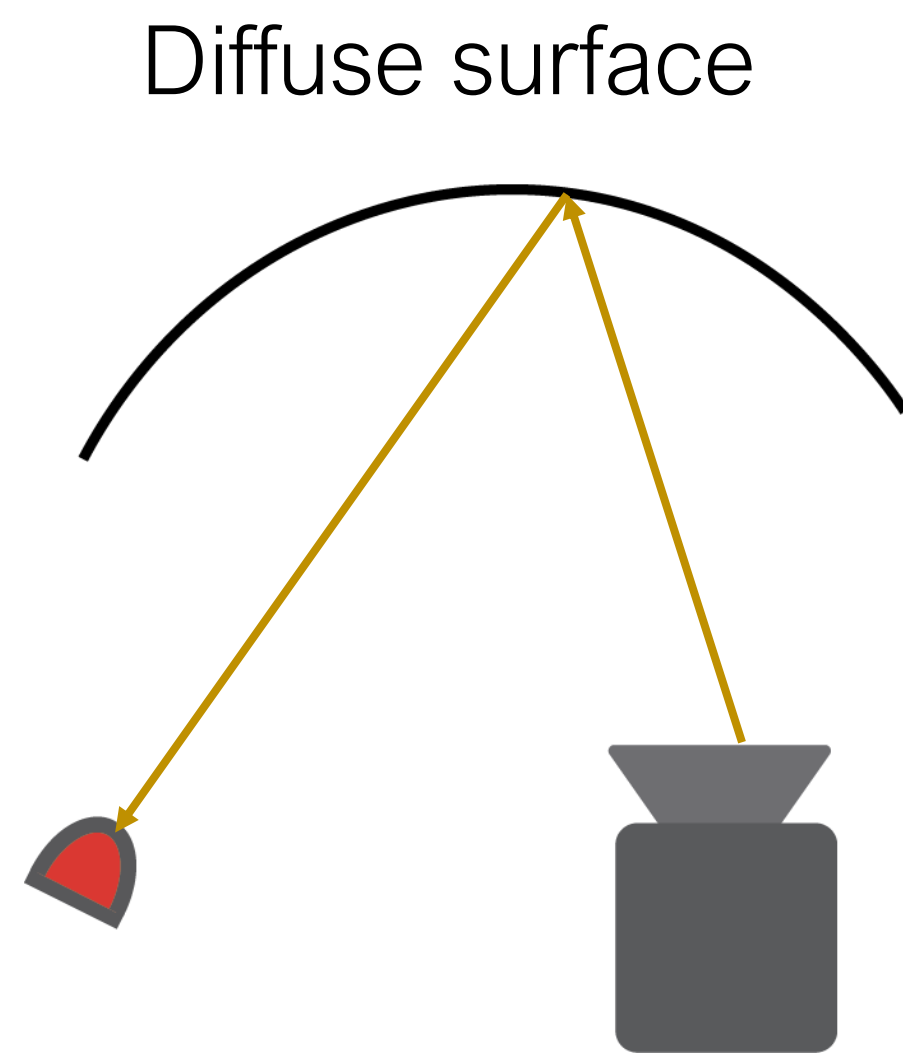


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

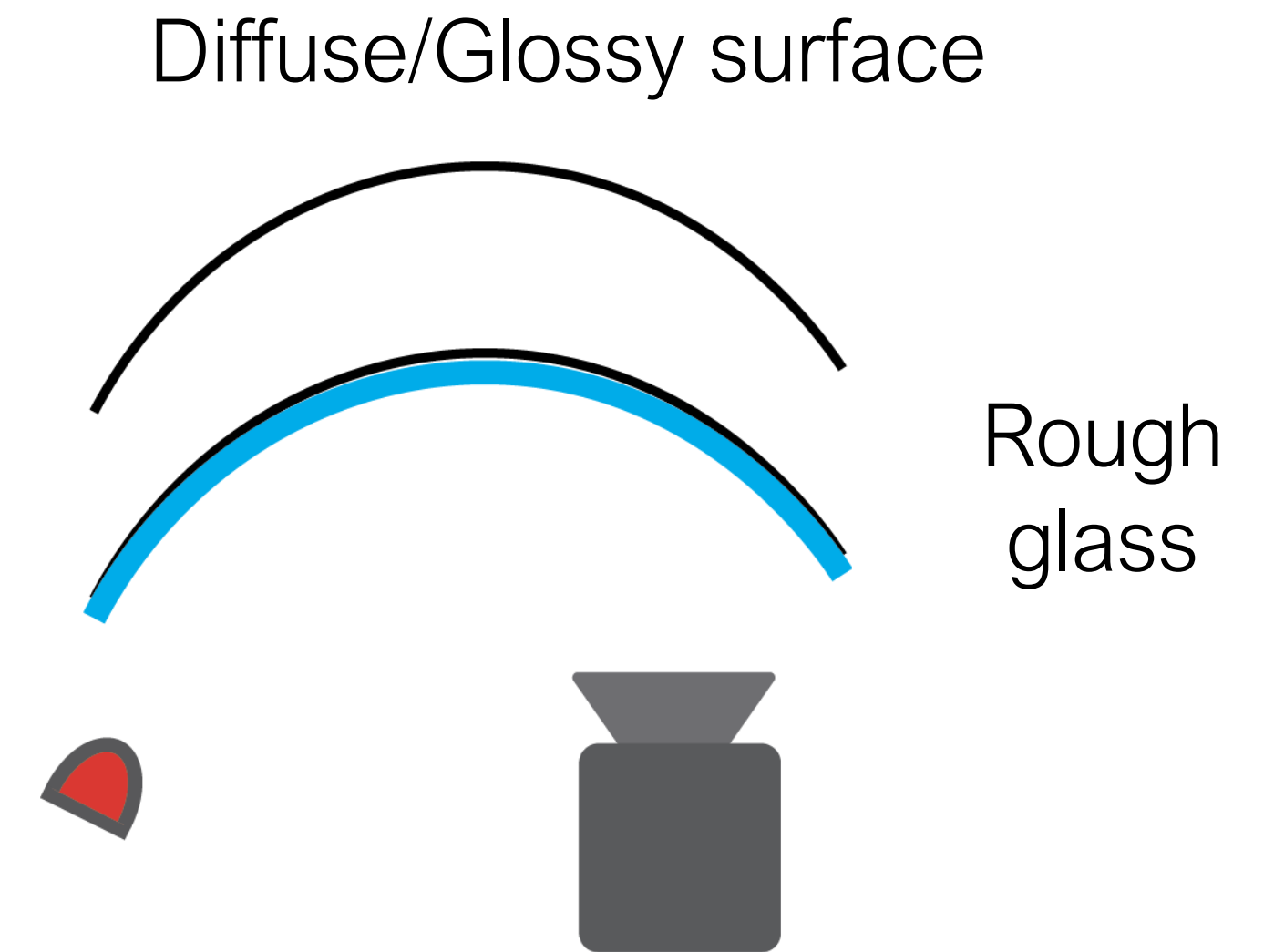
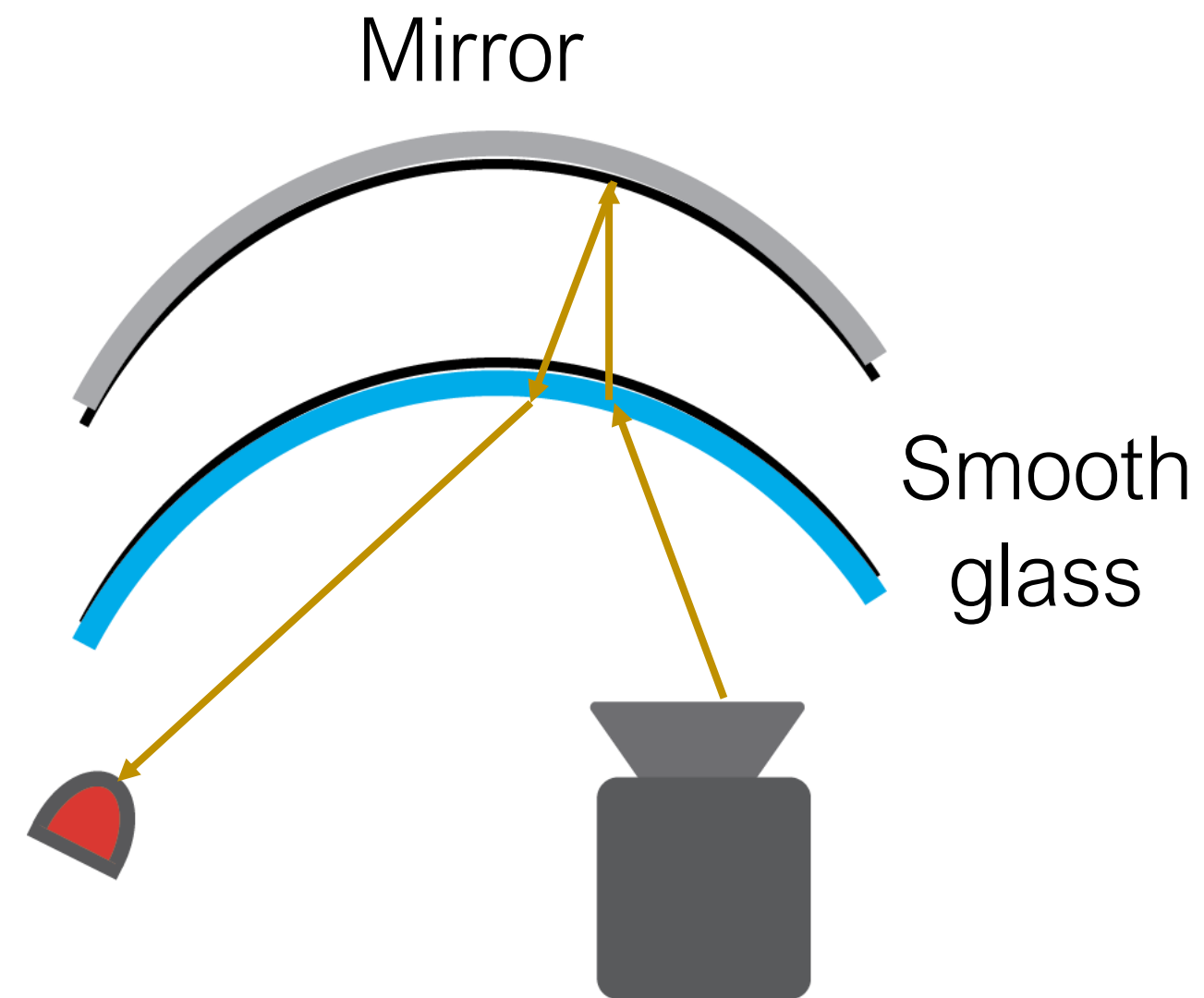
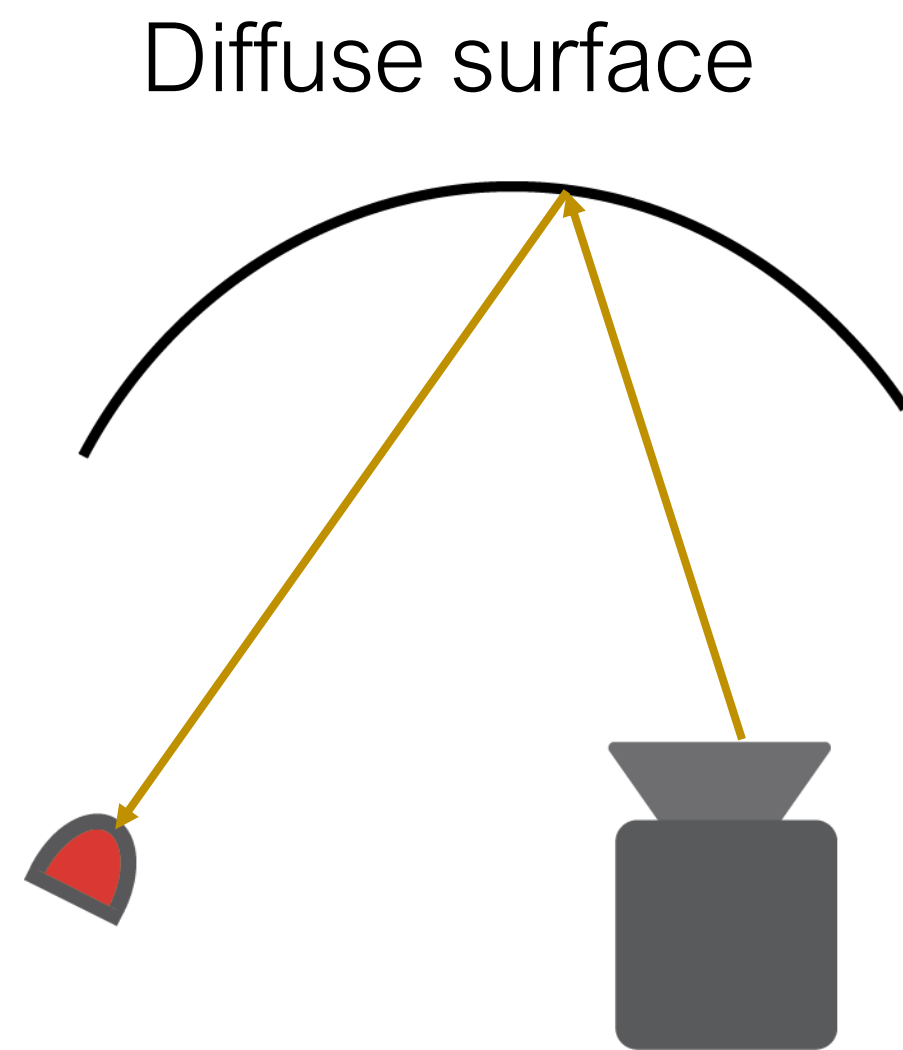


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

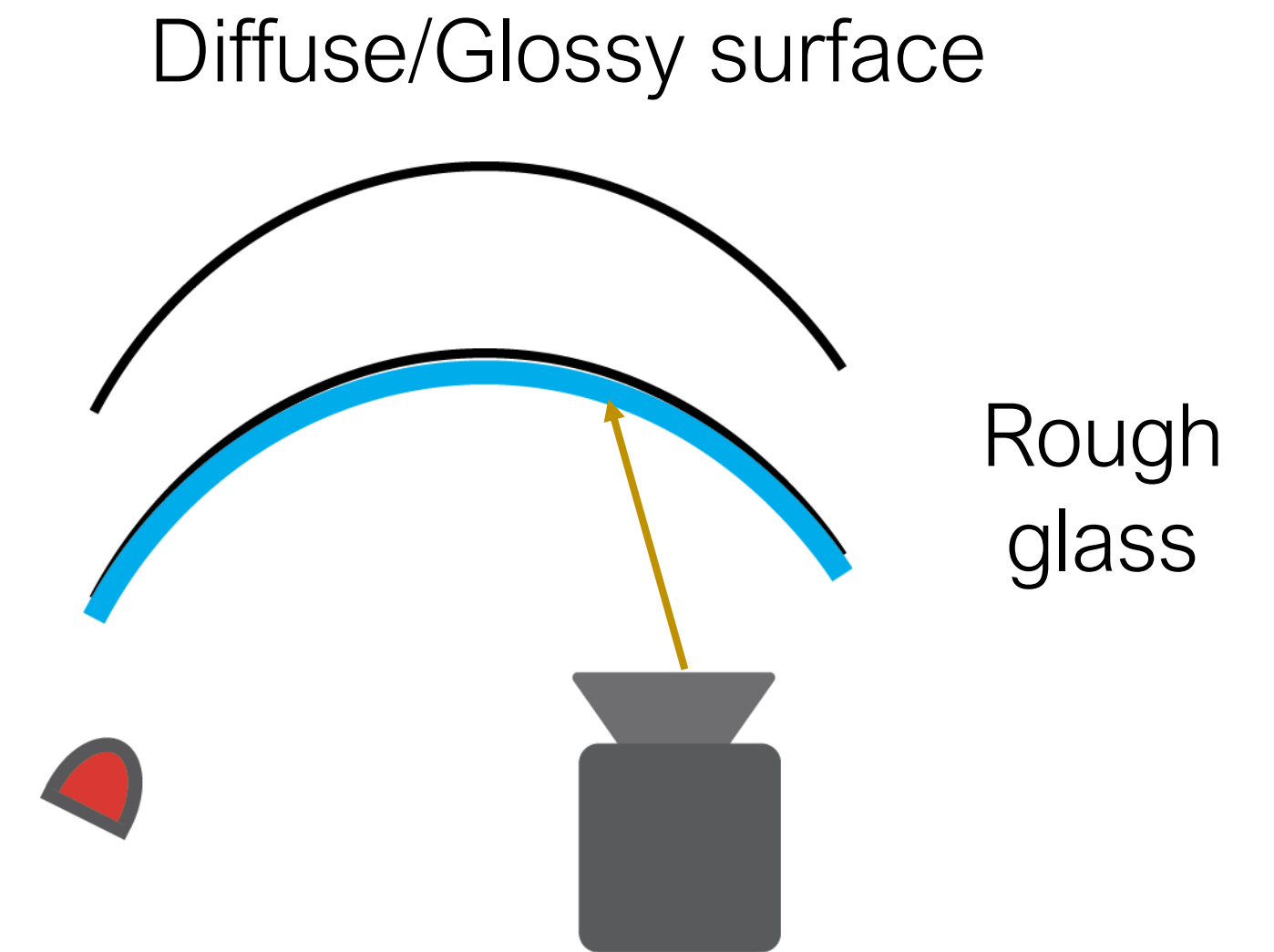
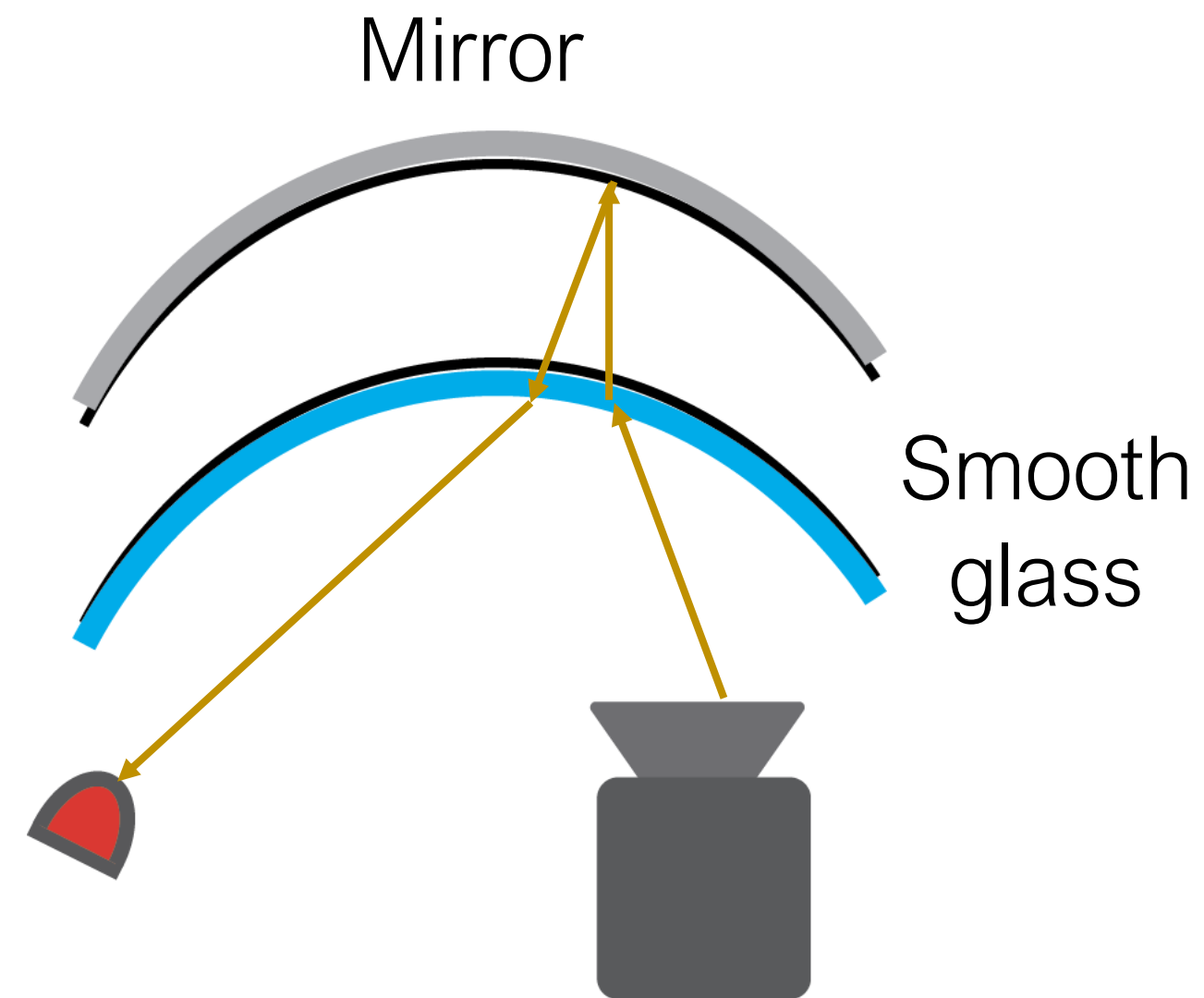
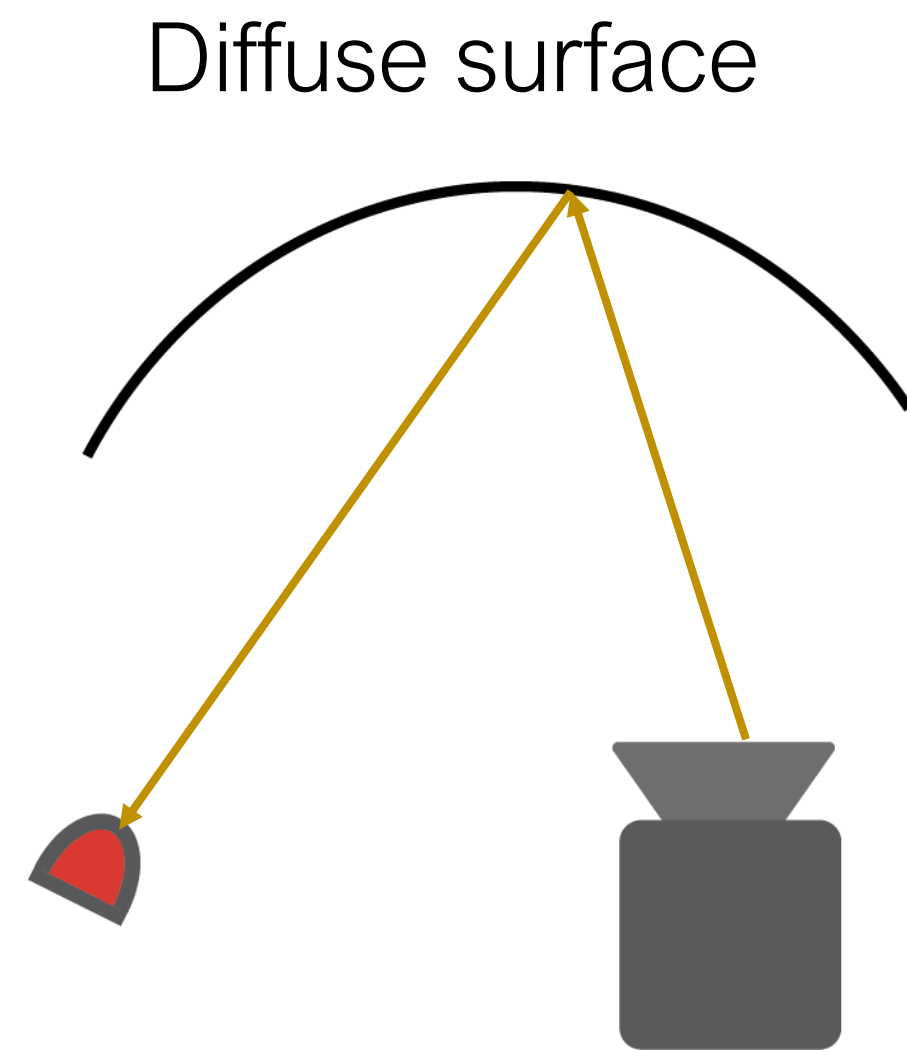


(Relatively) easy cases

Hard case

Sensor design framework: optical simulation

- What makes simulation challenging

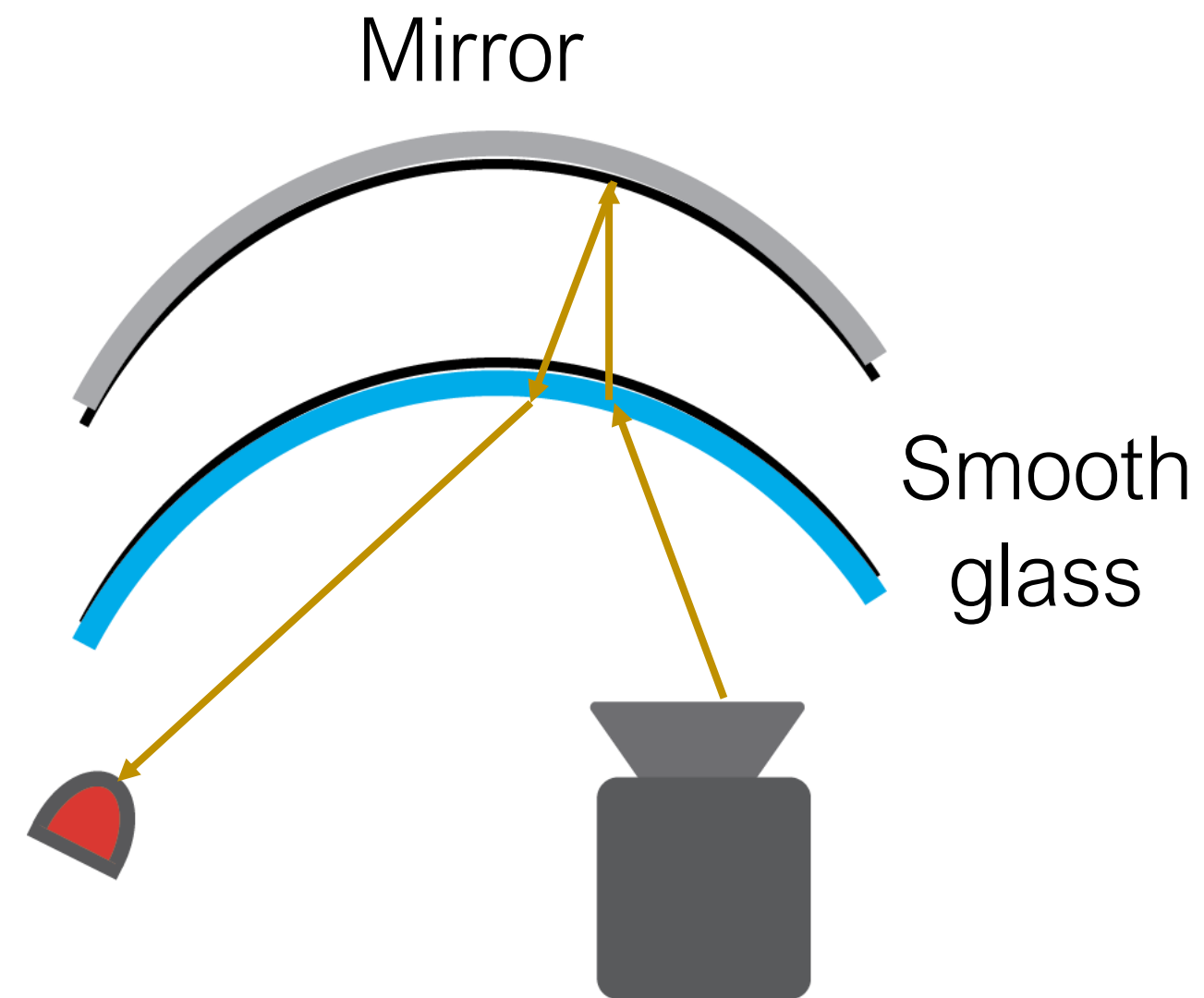
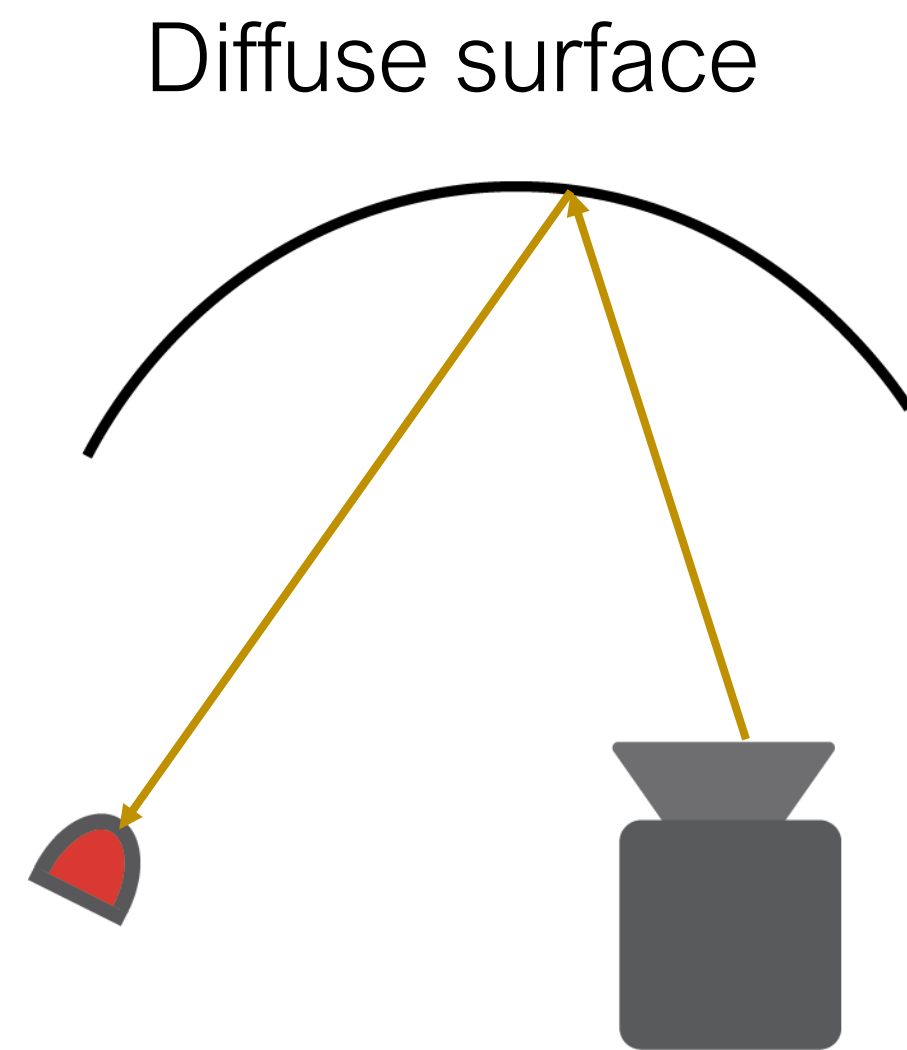


(Relatively) easy cases

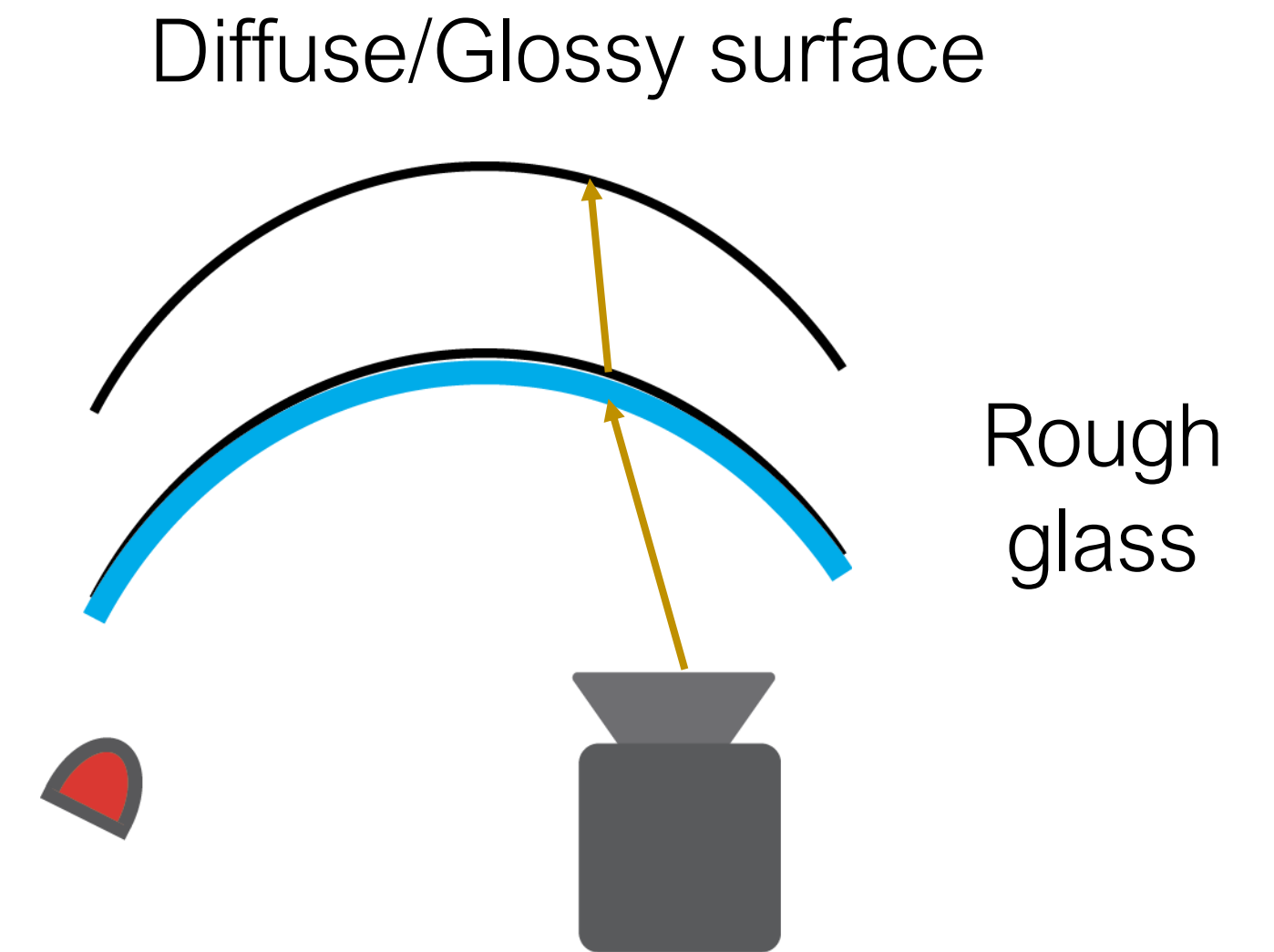
Hard case

Sensor design framework: optical simulation

- What makes simulation challenging



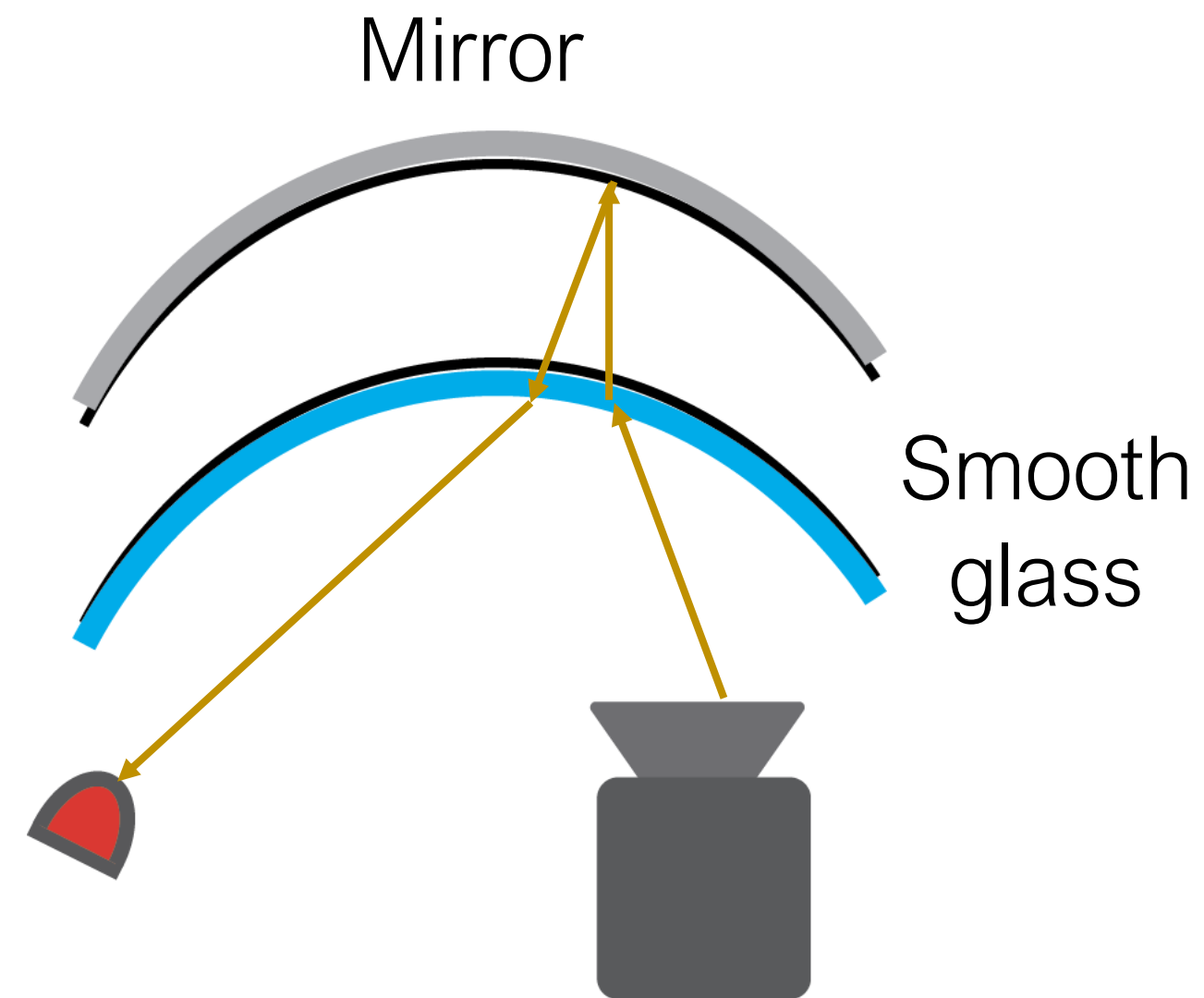
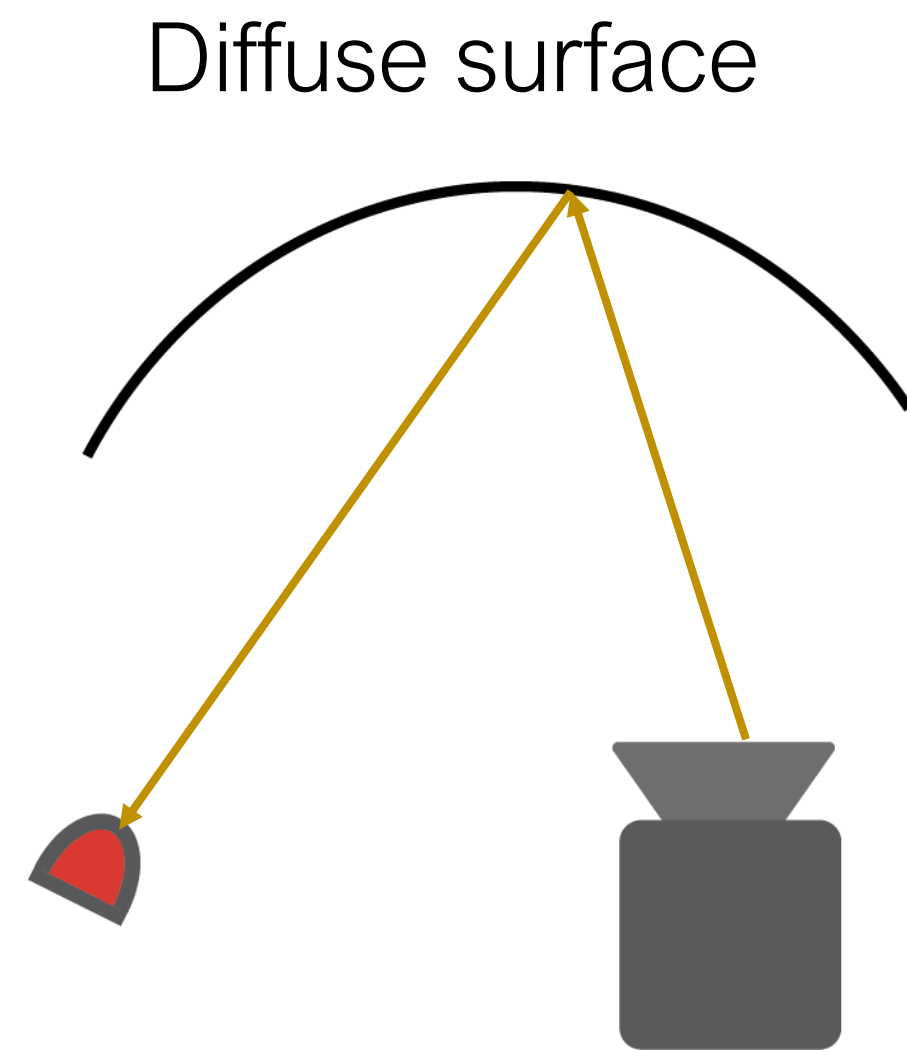
(Relatively) easy cases



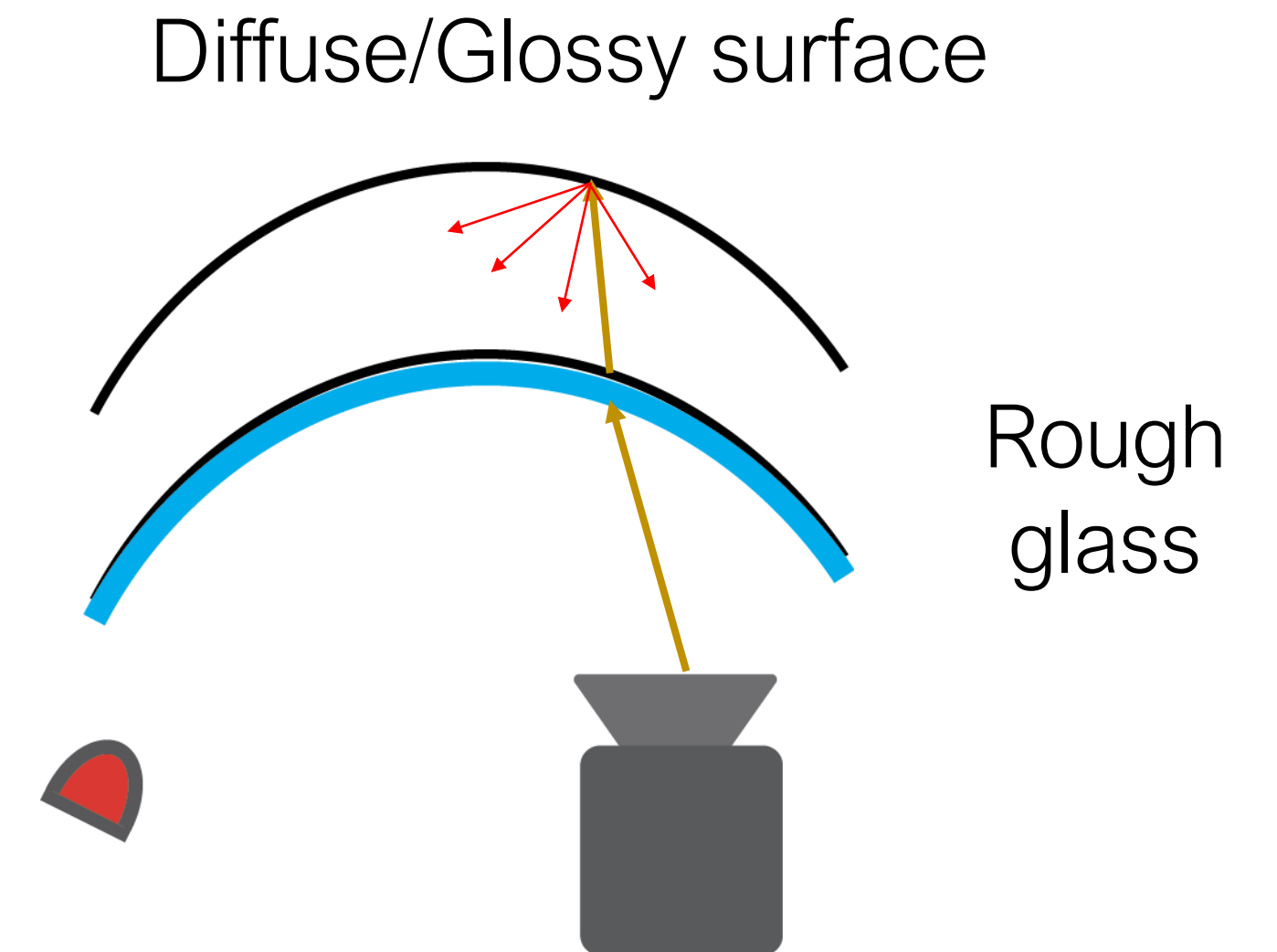
Hard case

Sensor design framework: optical simulation

- What makes simulation challenging



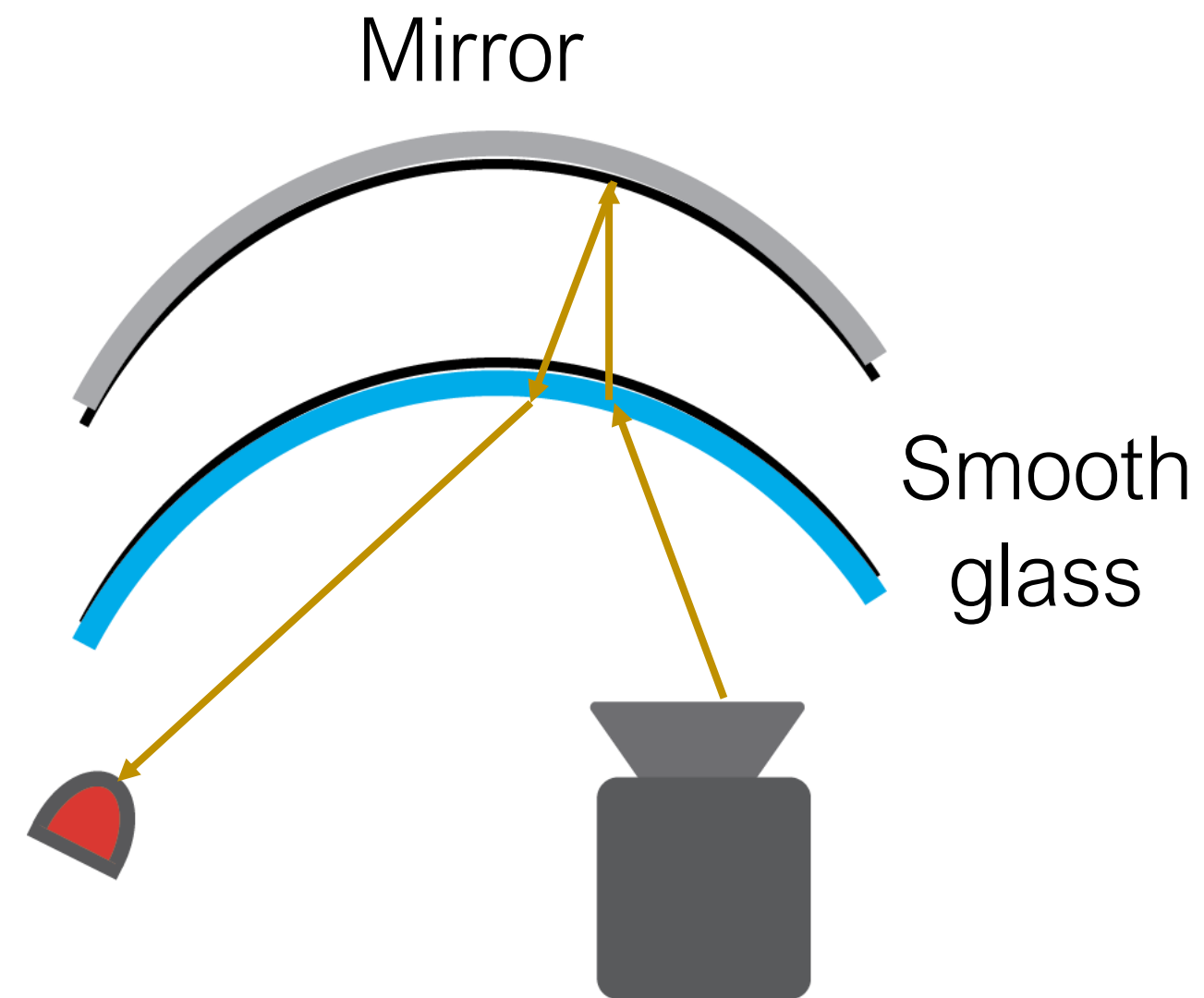
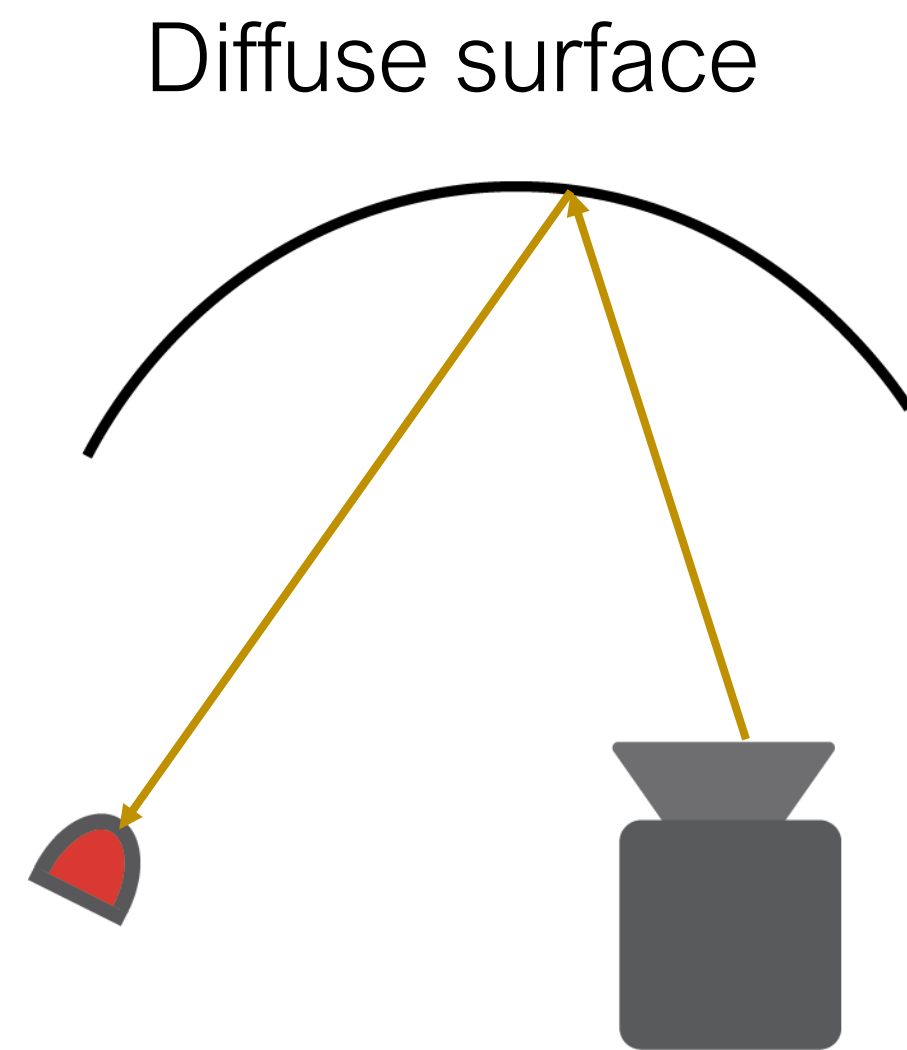
(Relatively) easy cases



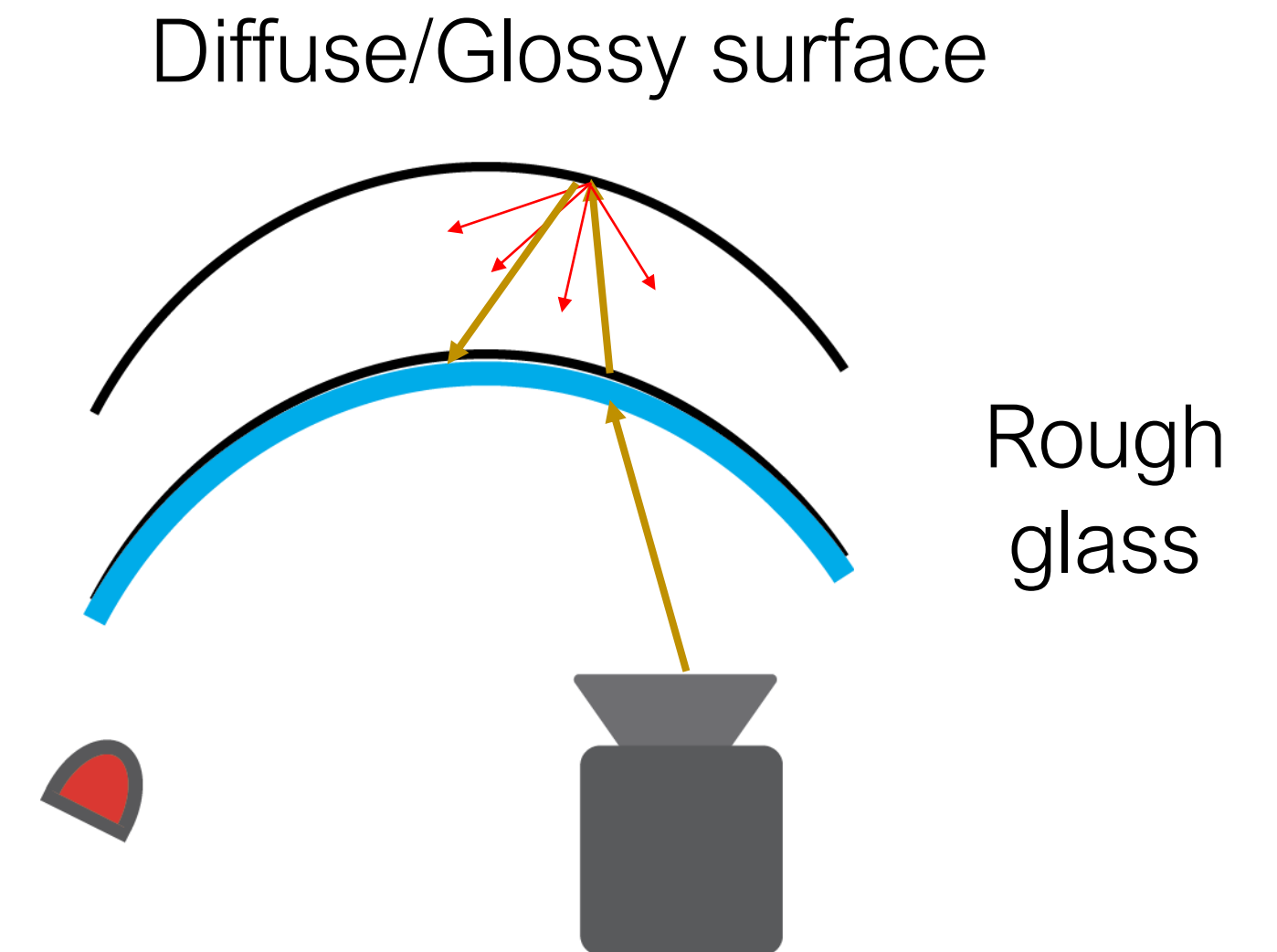
Hard case

Sensor design framework: optical simulation

- What makes simulation challenging



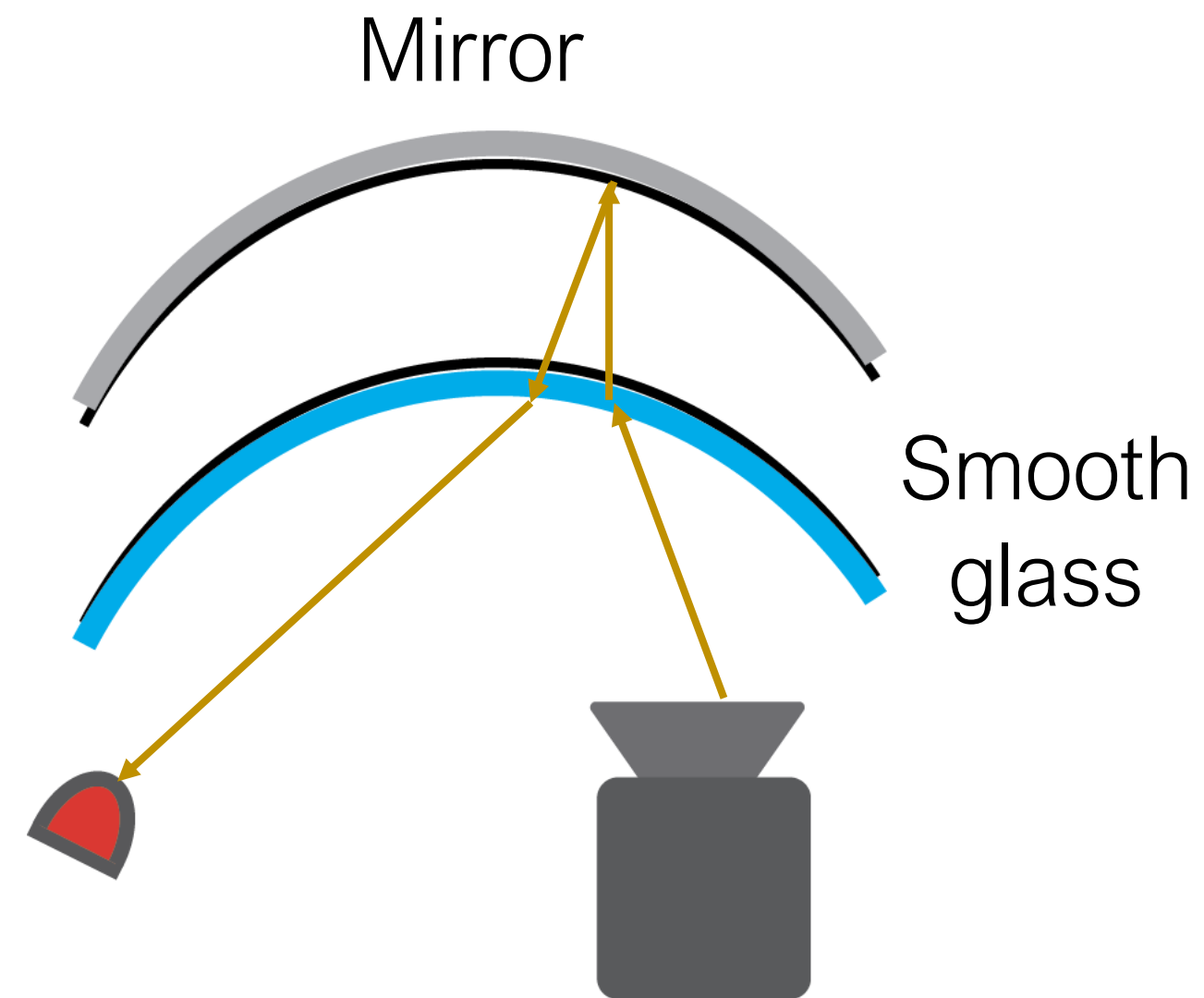
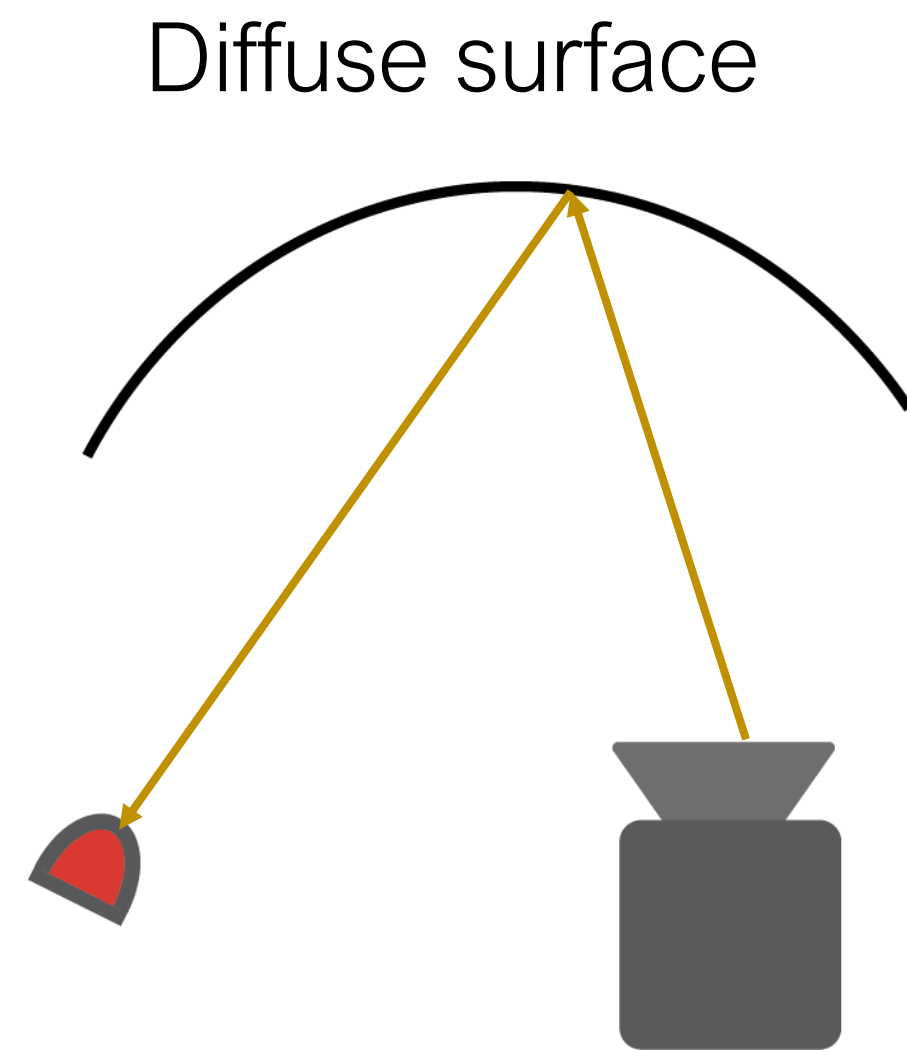
(Relatively) easy cases



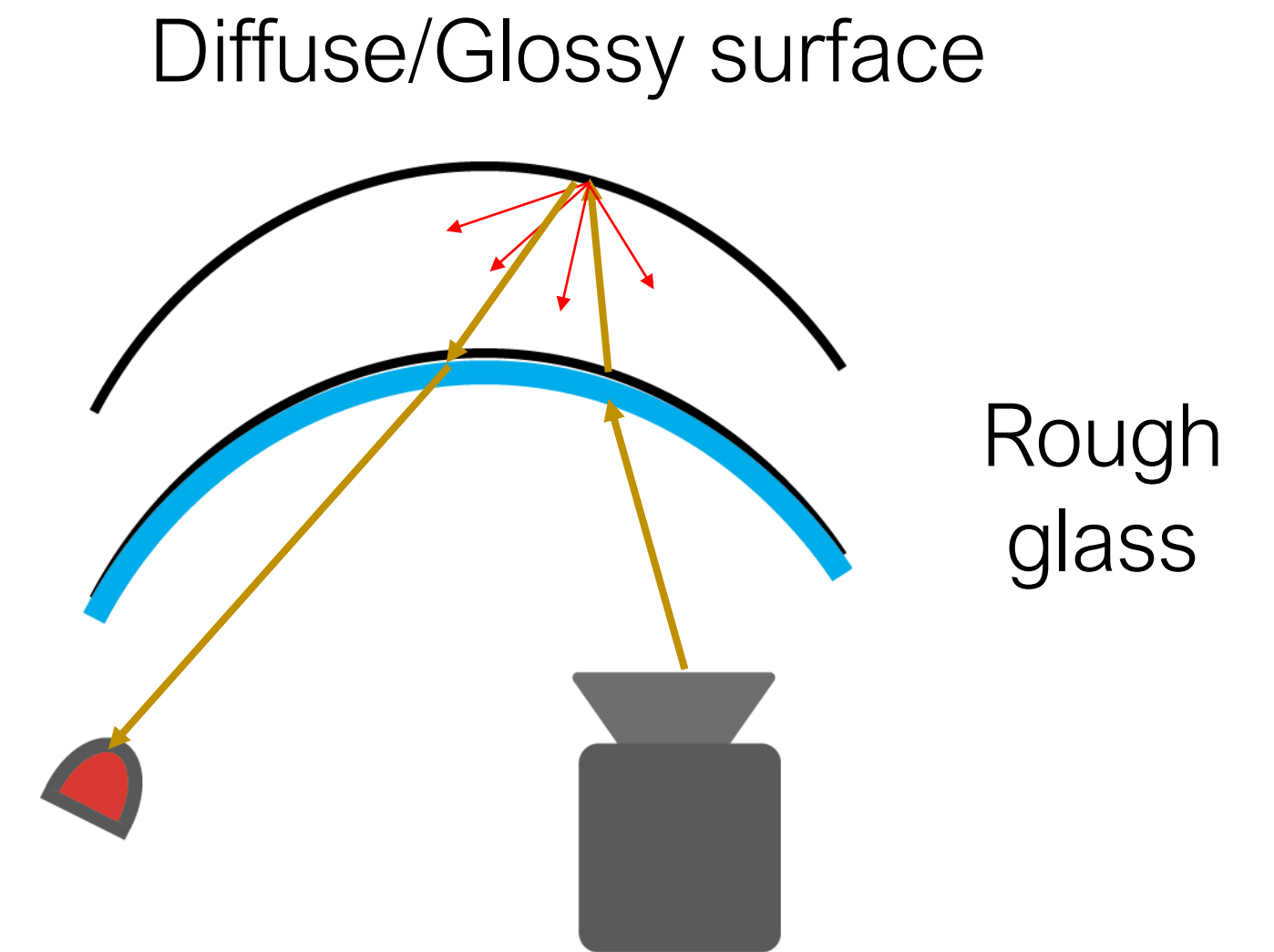
Hard case

Sensor design framework: optical simulation

- What makes simulation challenging



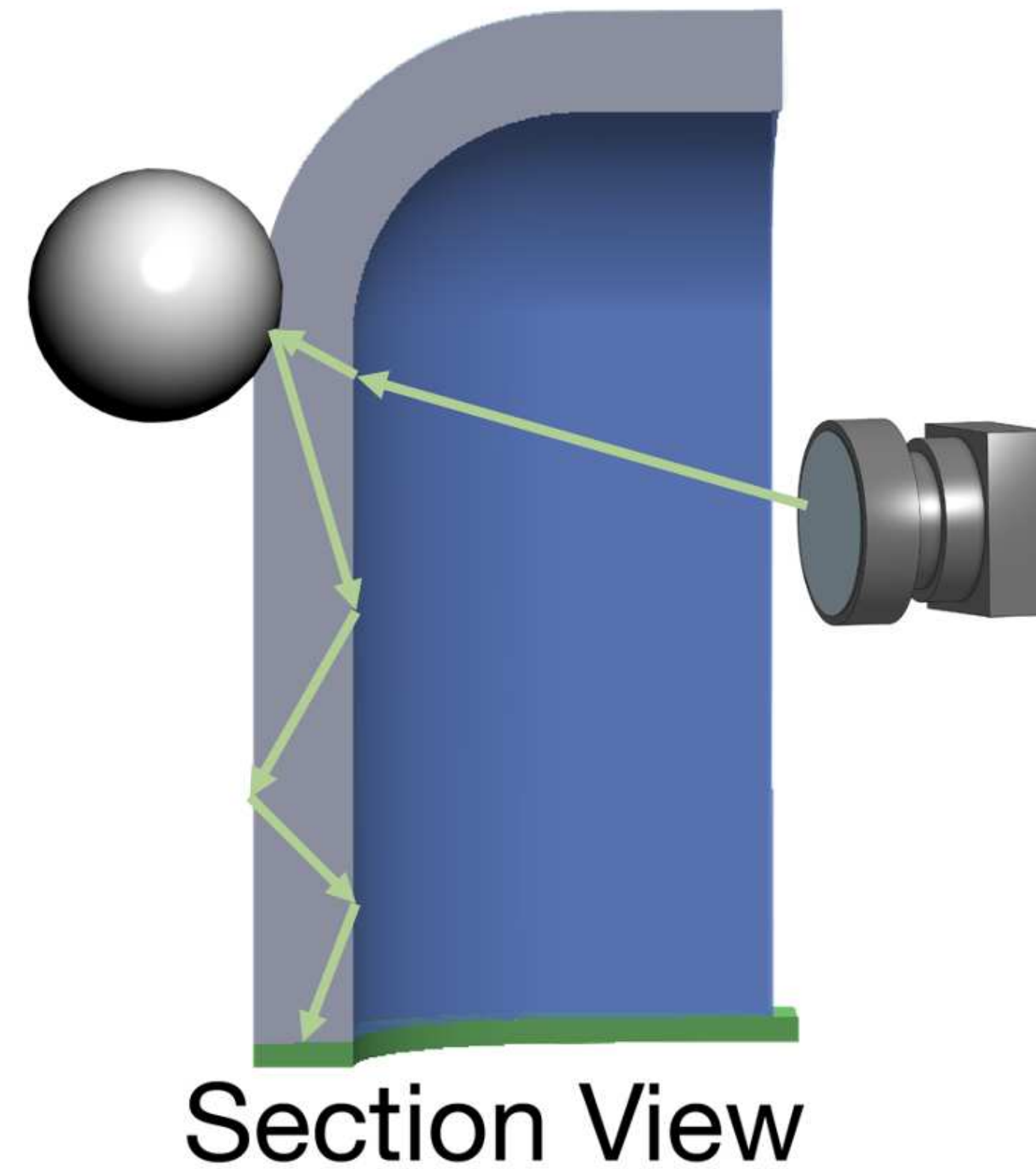
(Relatively) easy cases




Hard case

Sensor design framework: optical simulation


- What makes simulation challenging
 - SDS light paths
 - Indirect illumination

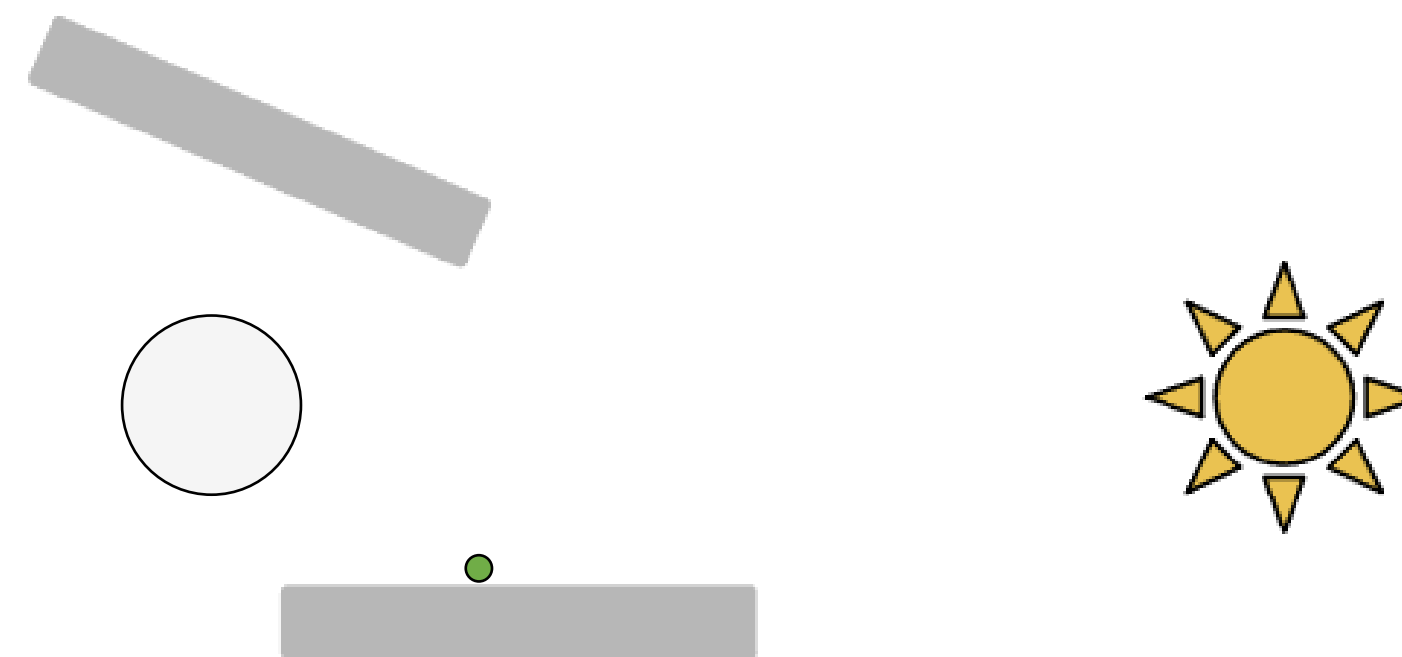
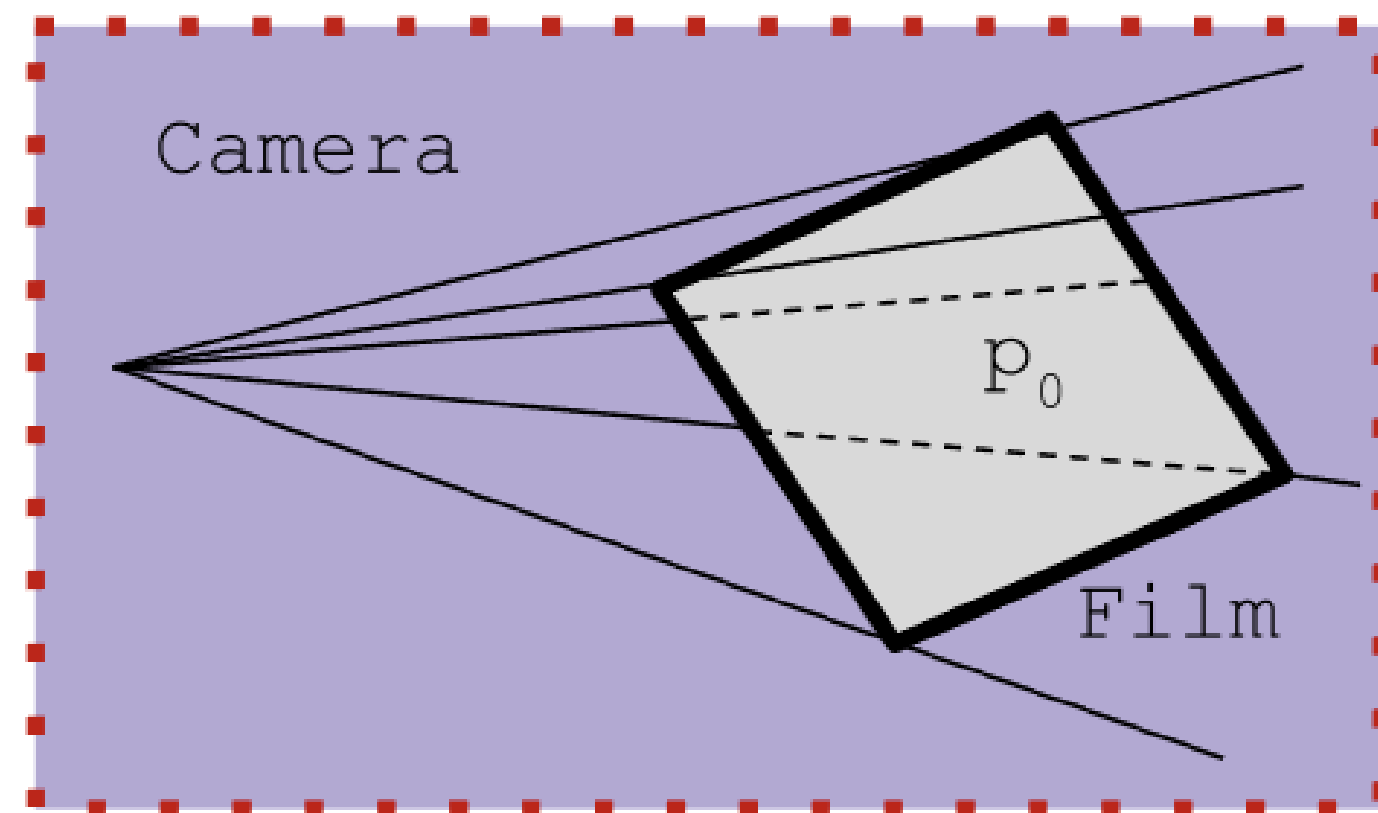


Sensor design framework: optical simulation


- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths

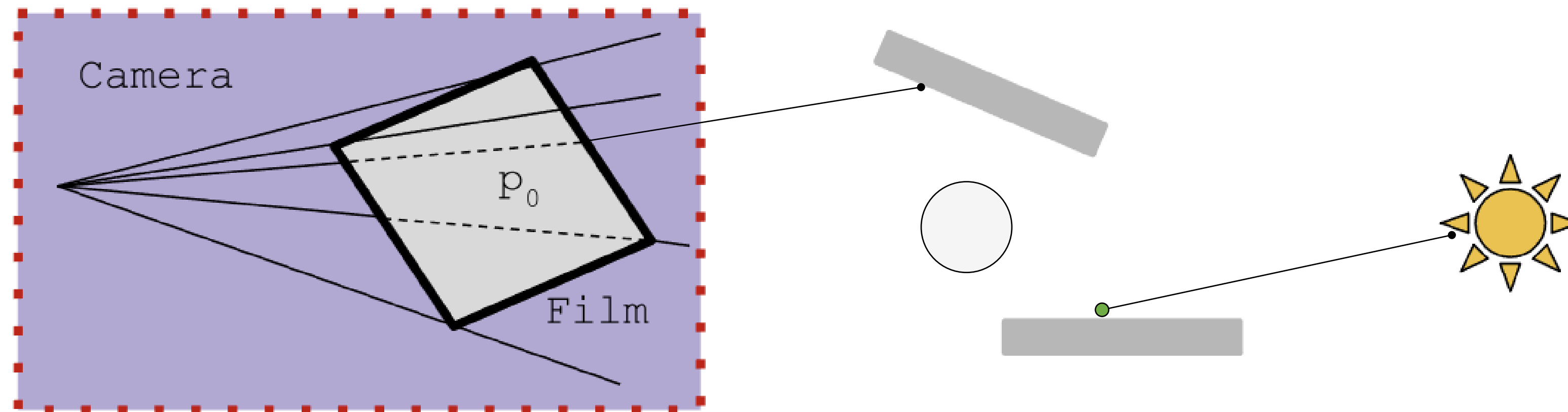
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




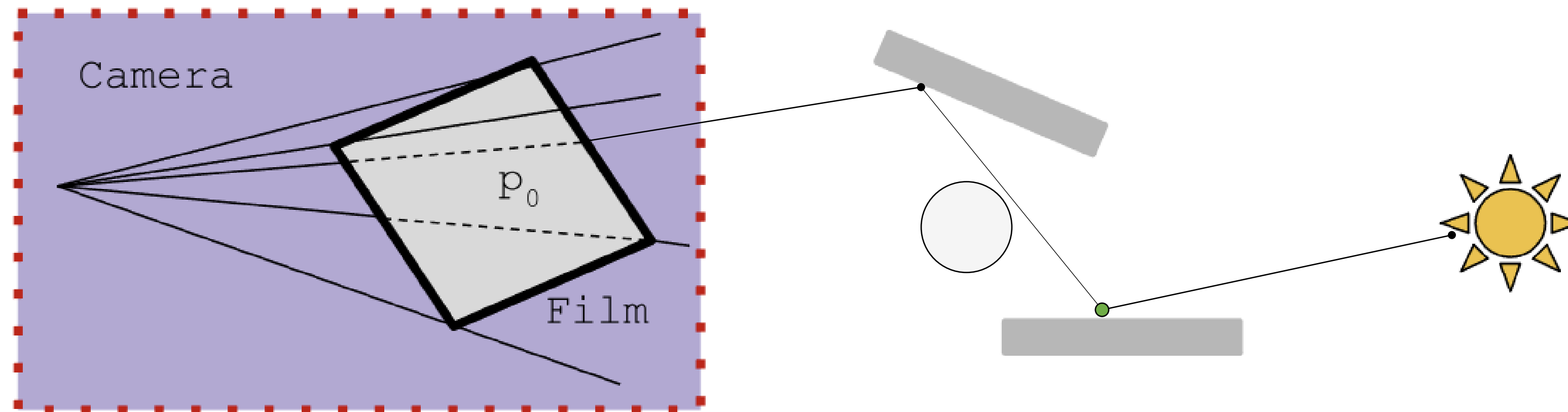
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




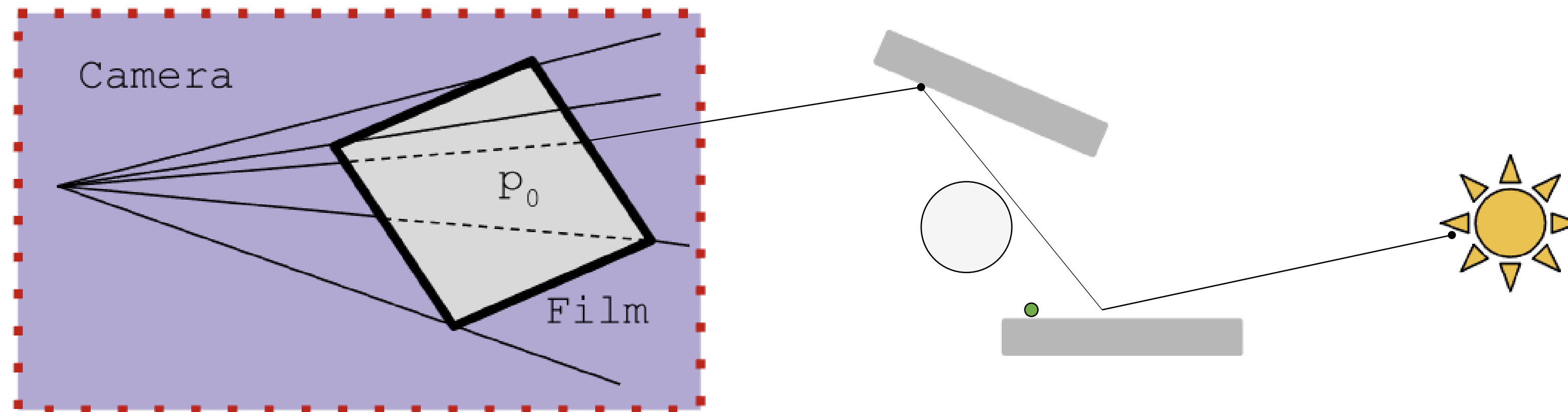
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




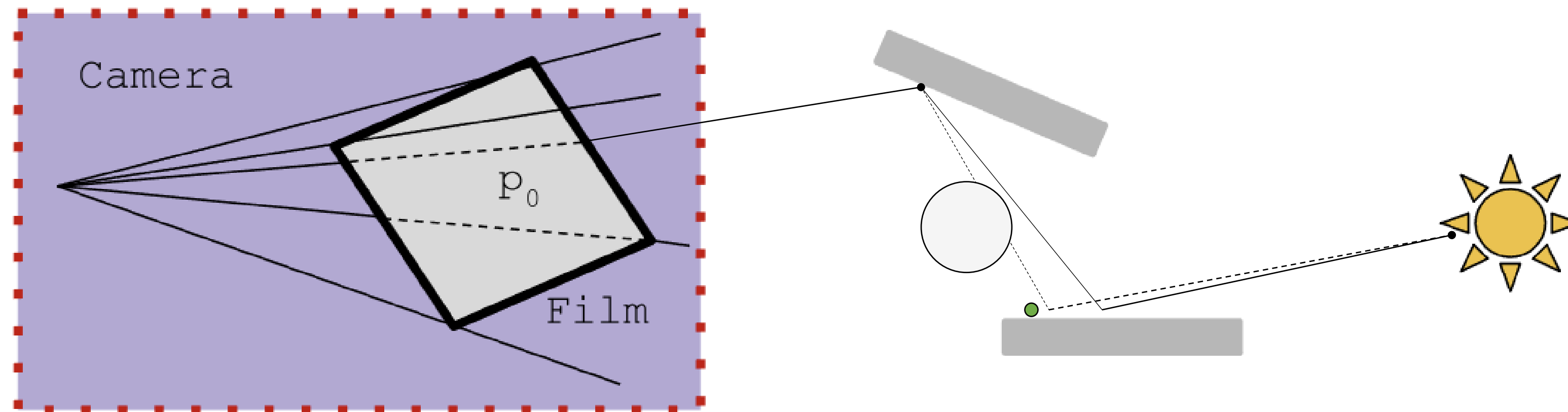
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




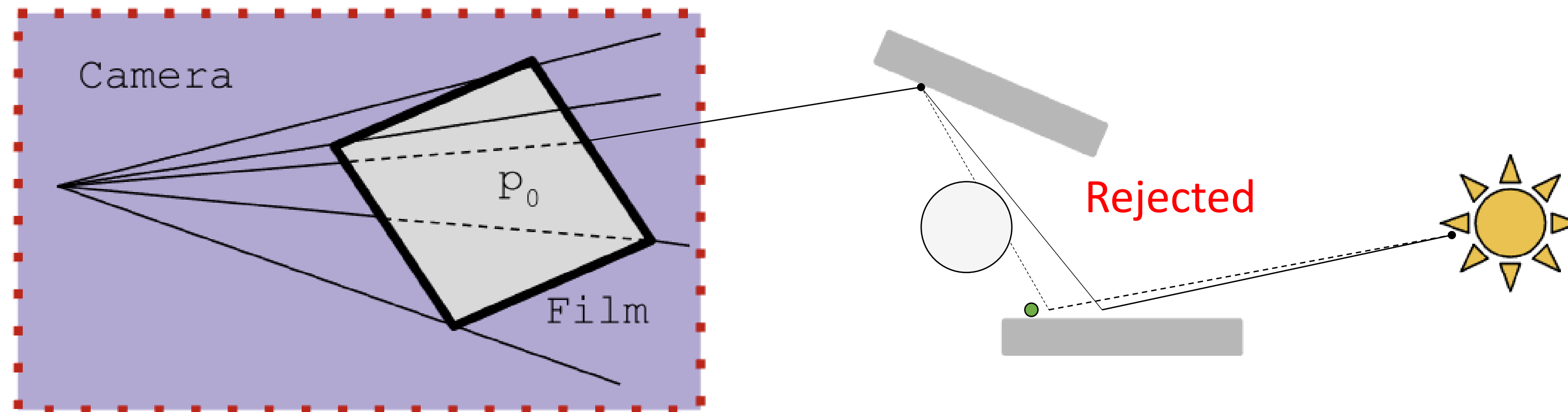
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




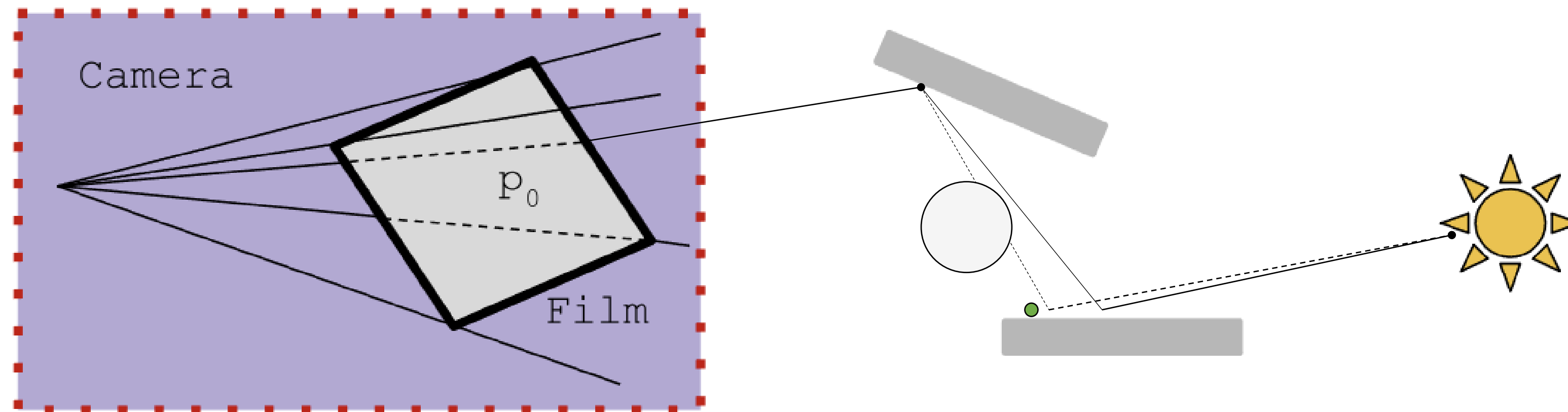
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




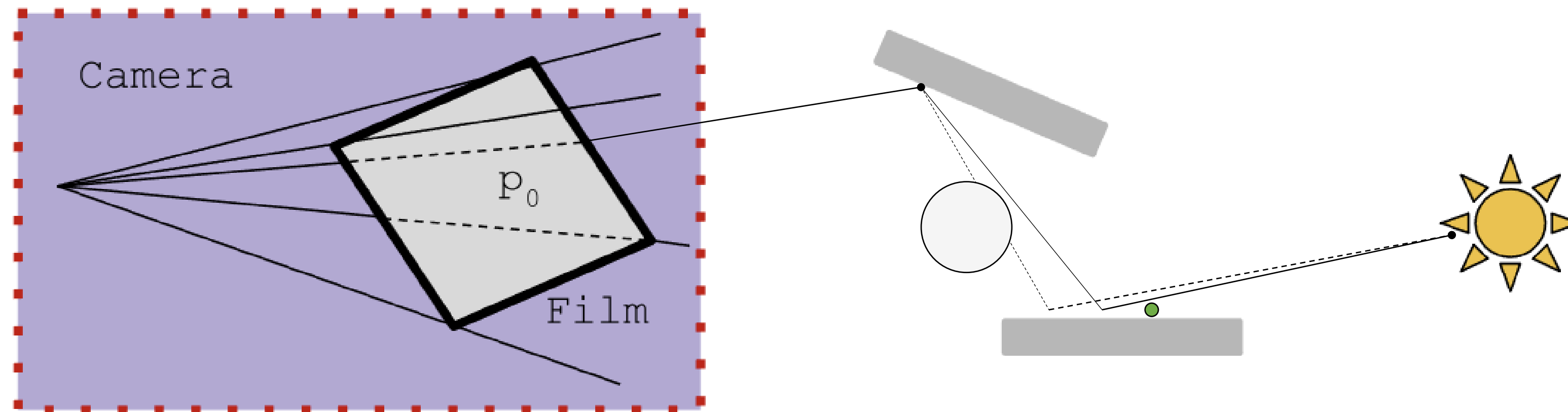
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




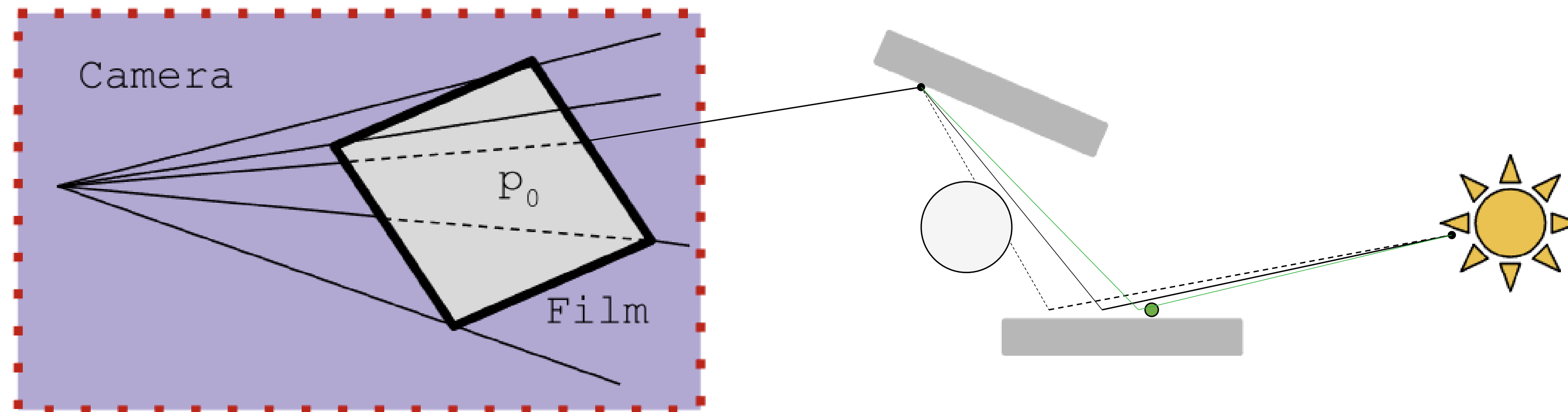
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths




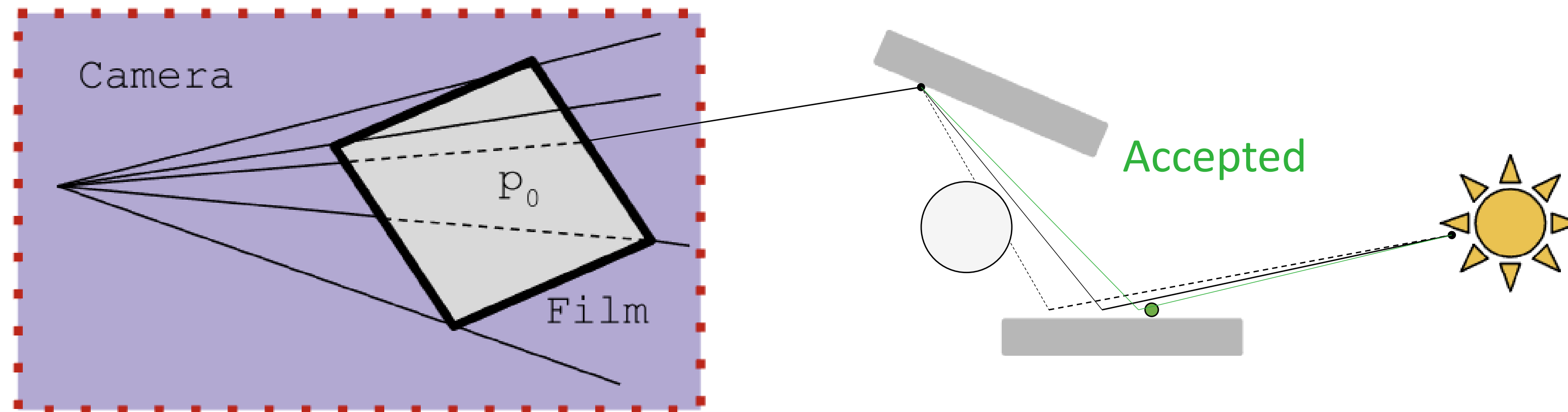
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths



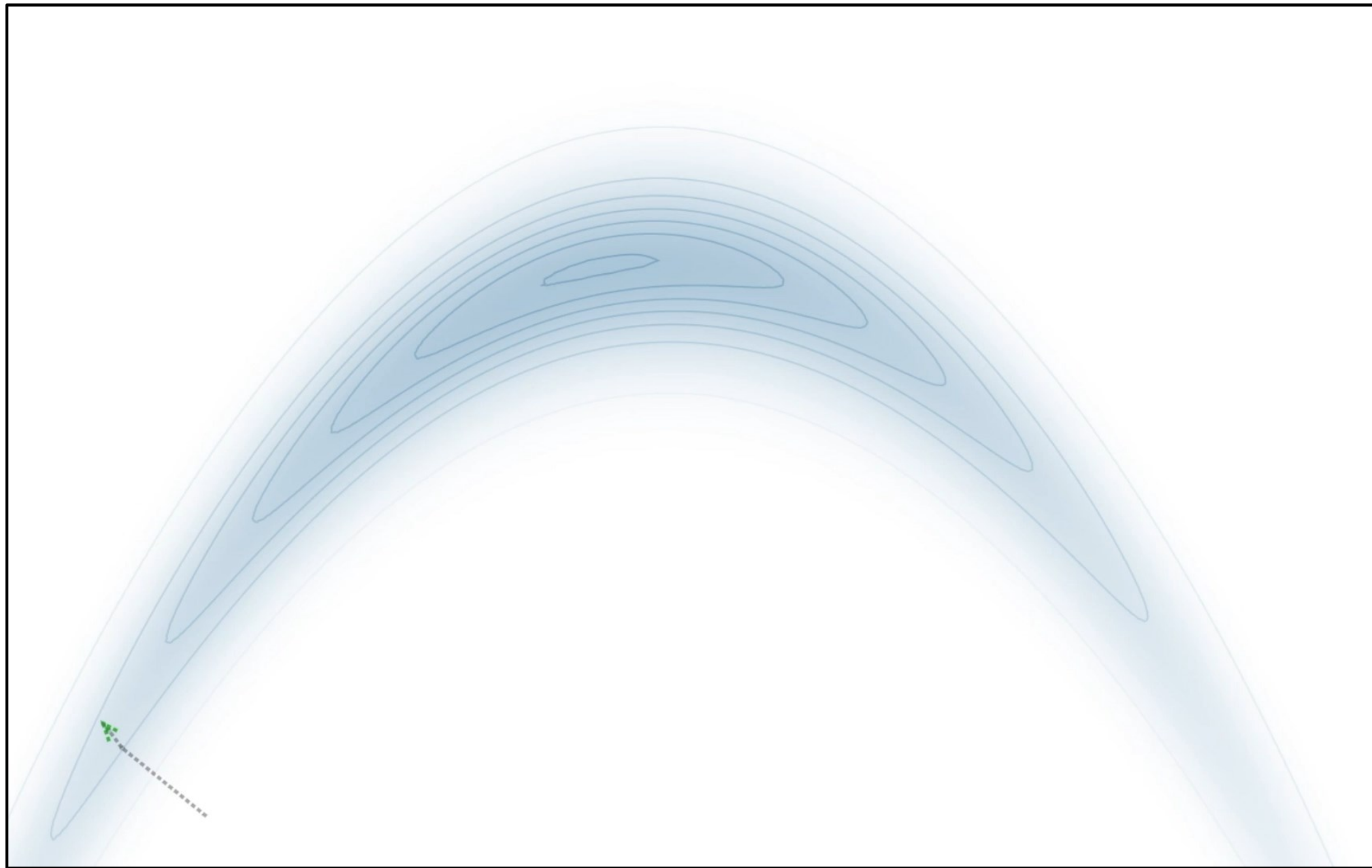
Sensor design framework: optical simulation

- What makes simulation challenging
 - SDS light paths
 - Indirect illumination
- Require Markov Chain Monte Carlo rendering techniques
 - Key idea  Slowly mutate paths to generate useful paths



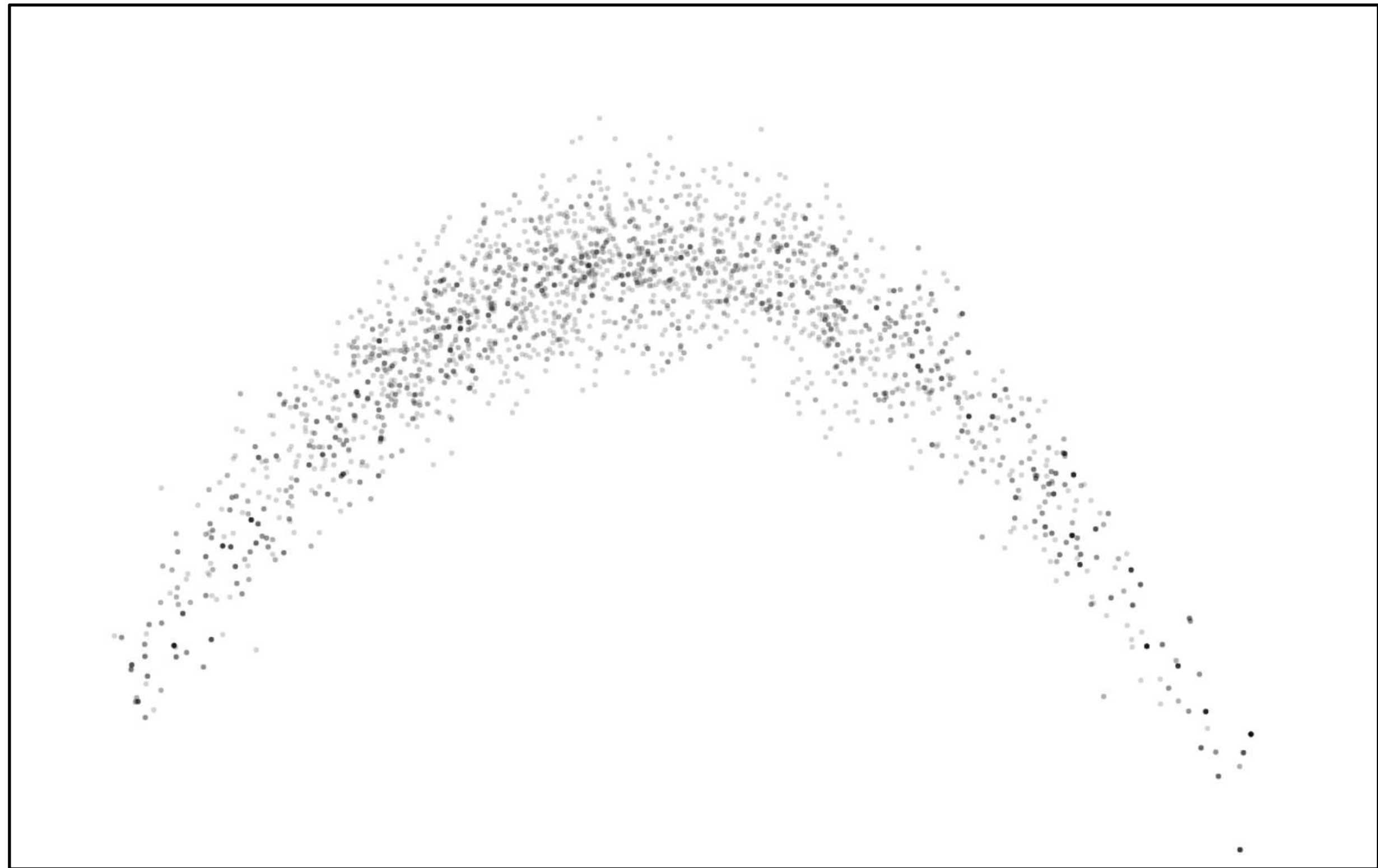
Optimization:

- Stochastic Gradient Descent (SGD)

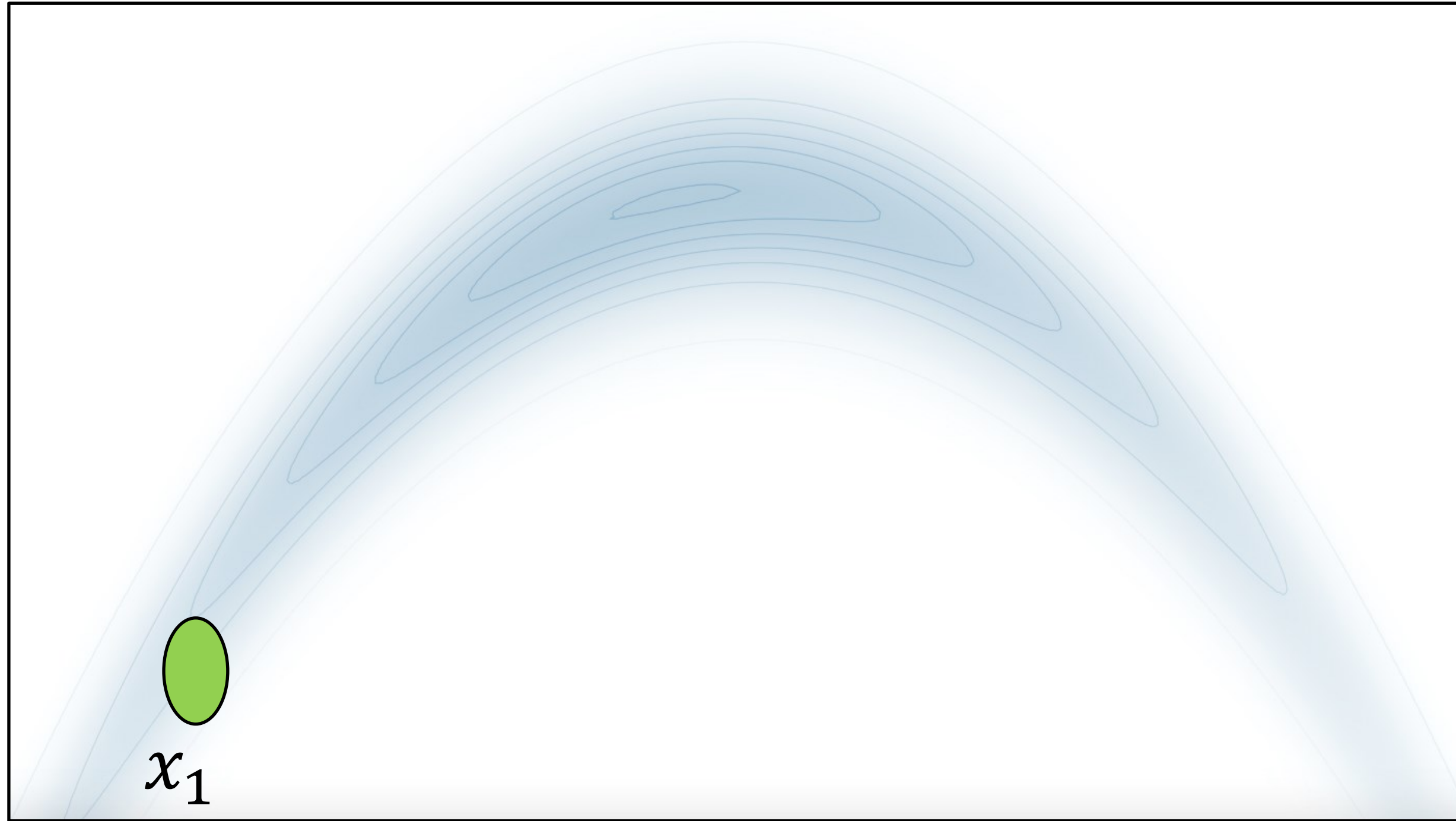


MCMC sampling:

- Langevin Monte Carlo (LMC)



GD OVERVIEW



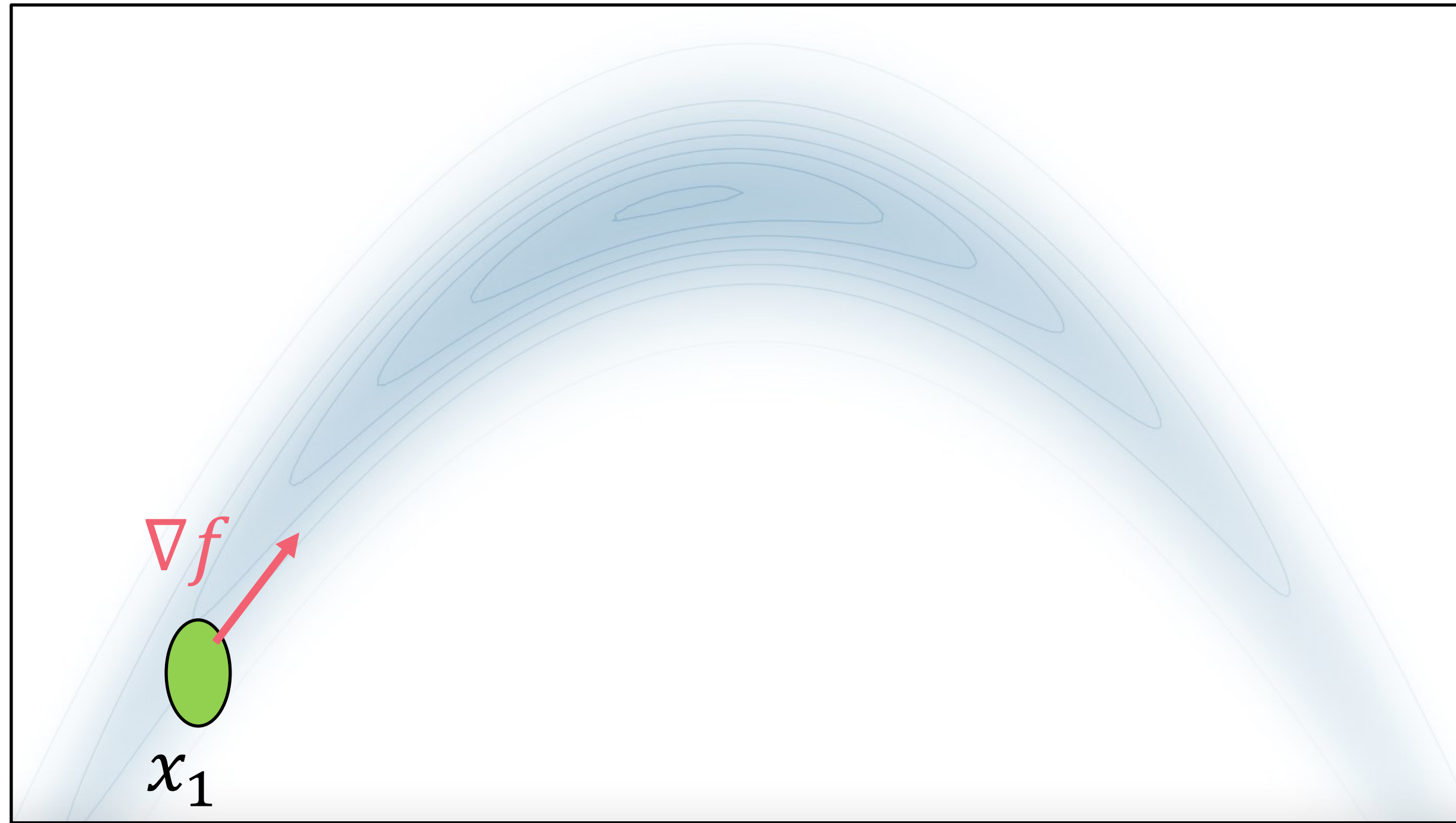
Optimization problem:

$$\max_x f(\mathbf{x})$$

Gradient descent/ascent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

GD OVERVIEW



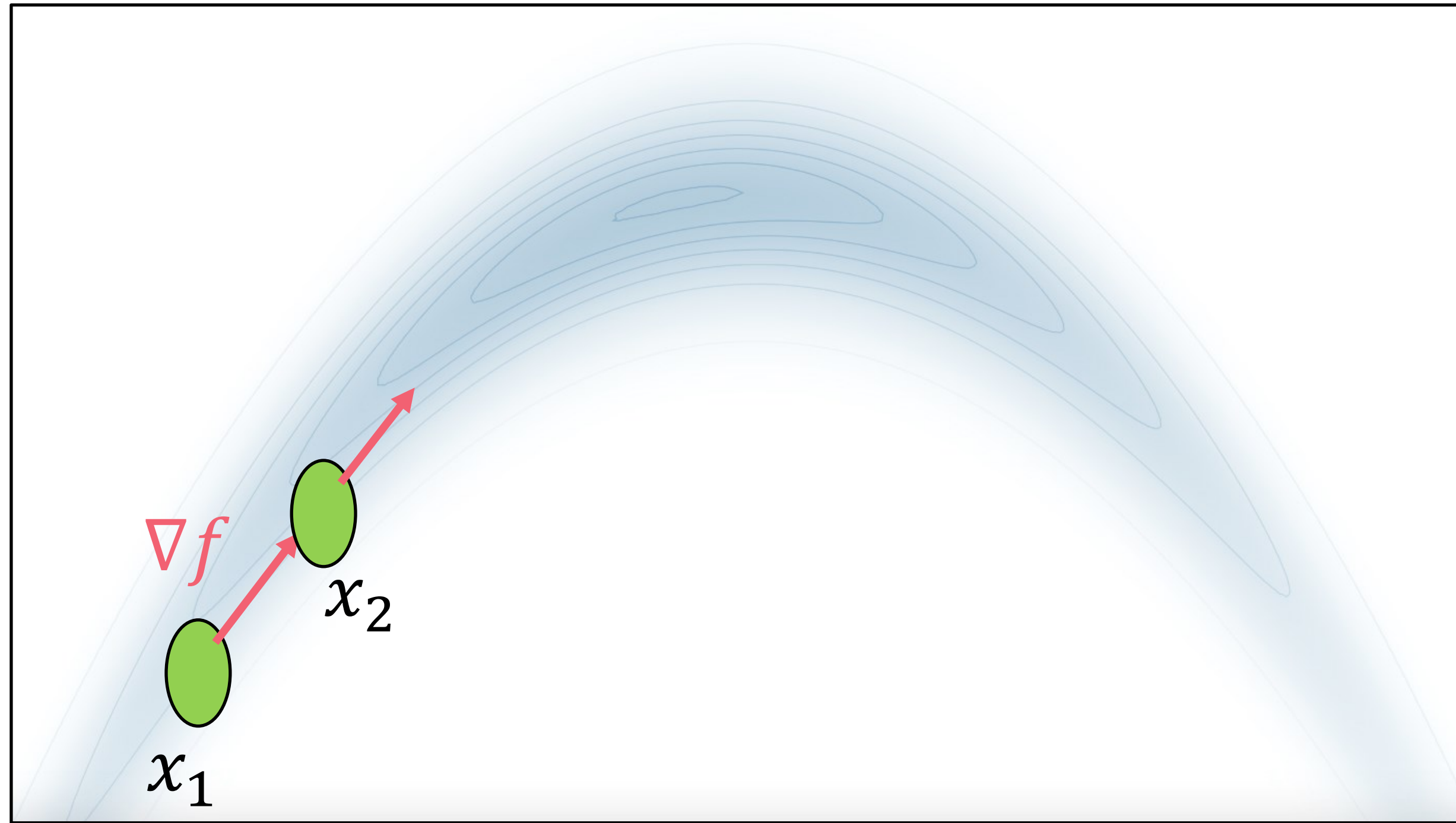
Optimization problem:

$$\max_x f(x)$$

Gradient descent/ascent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

GD OVERVIEW



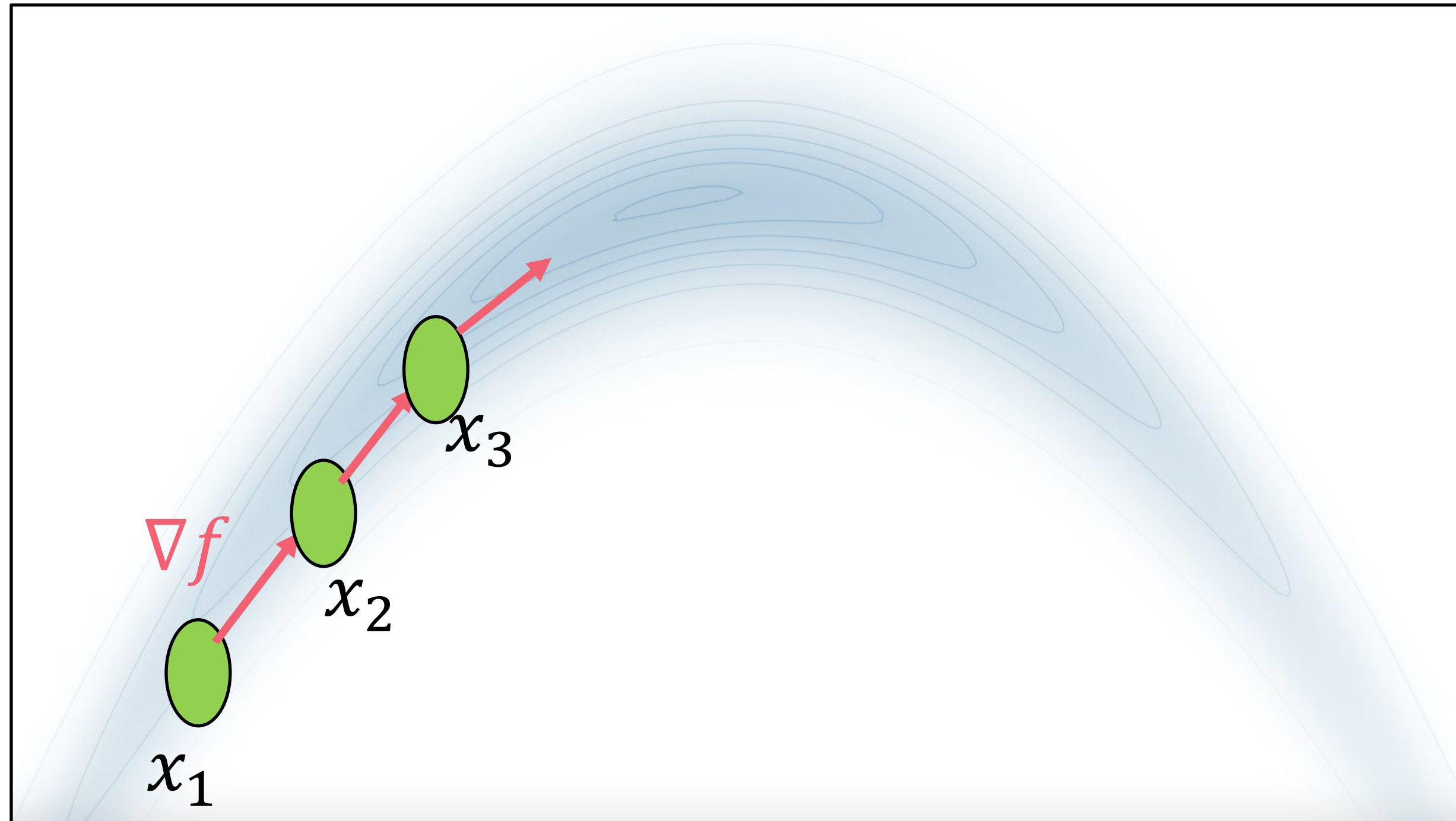
Optimization problem:

$$\max_x f(x)$$

Gradient descent/ascent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

GD OVERVIEW



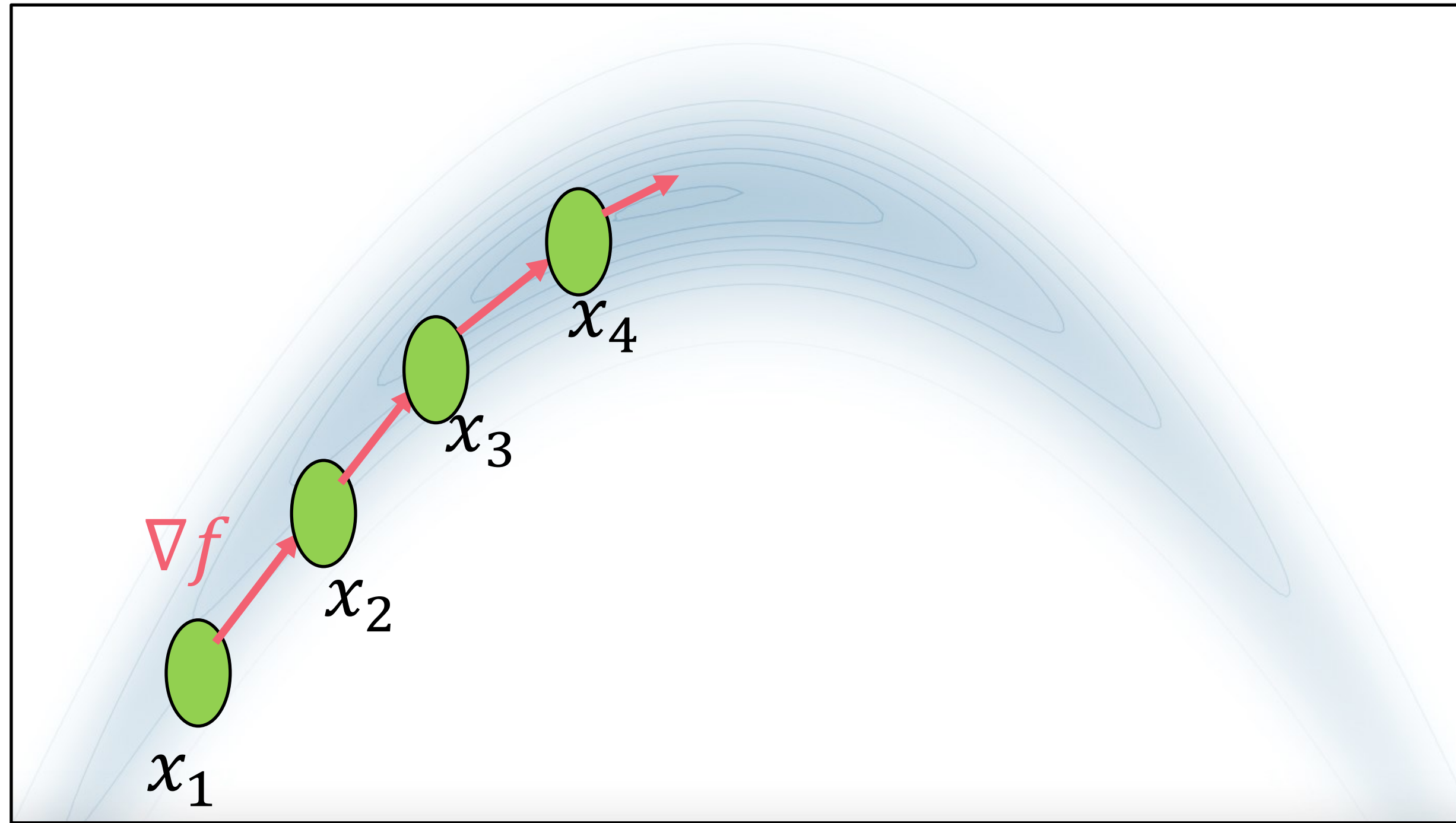
Optimization problem:

$$\max_x f(\mathbf{x})$$

Gradient descent/ascent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

GD OVERVIEW



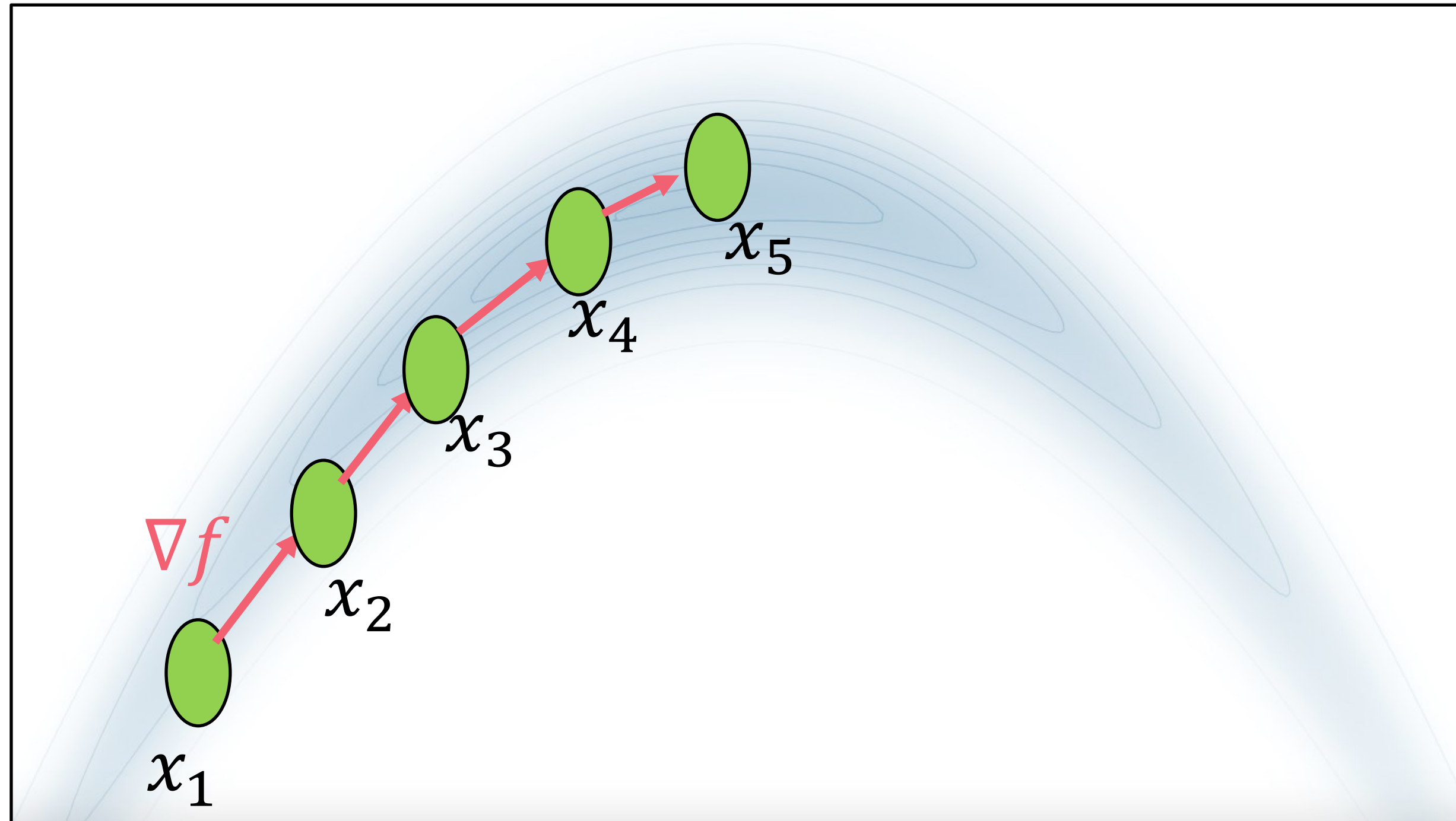
Optimization problem:

$$\max_x f(\mathbf{x})$$

Gradient descent/ascent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

GD OVERVIEW



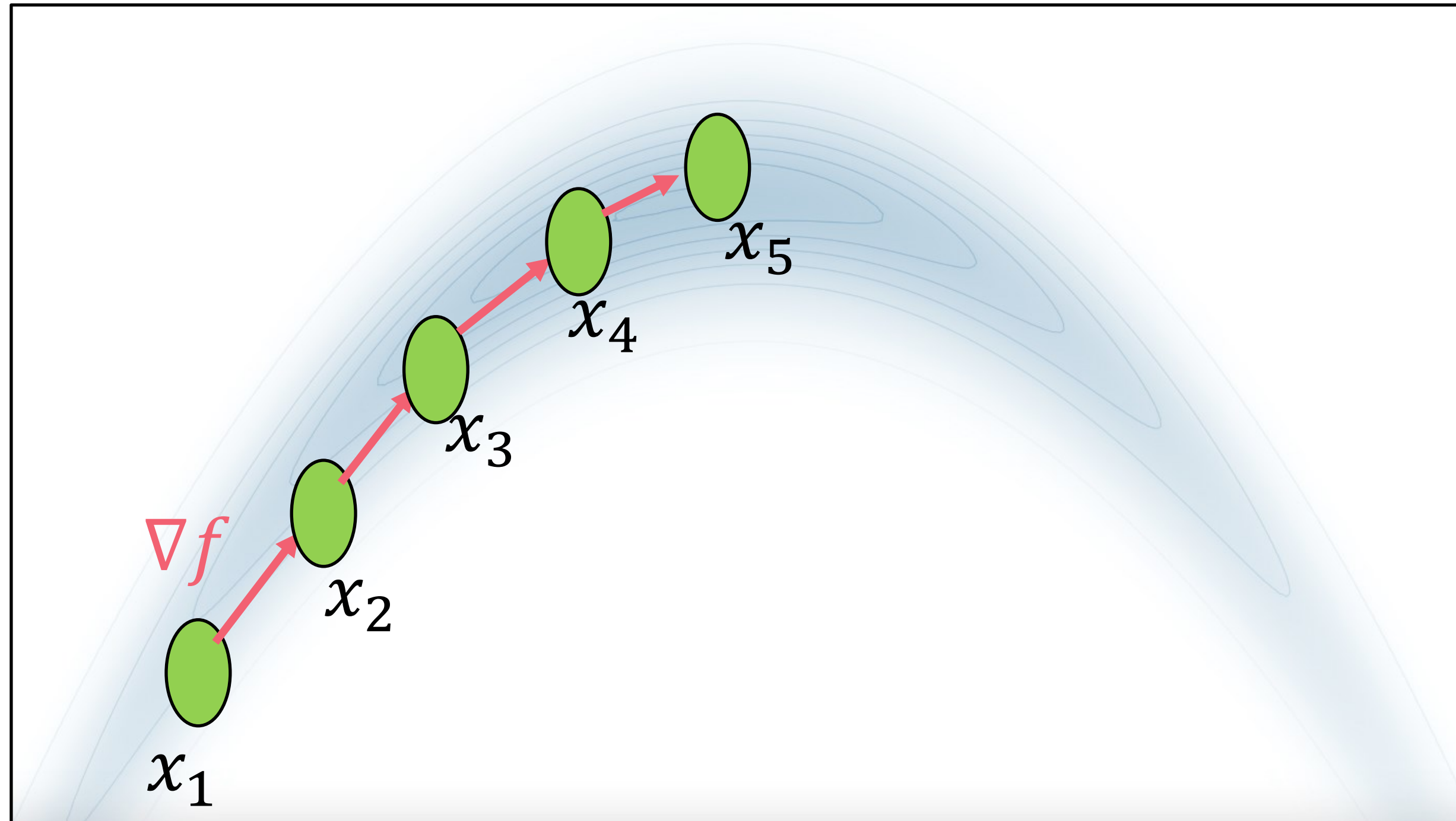
Optimization problem:

$$\max_x f(\mathbf{x})$$

Gradient descent/ascent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

GD OVERVIEW



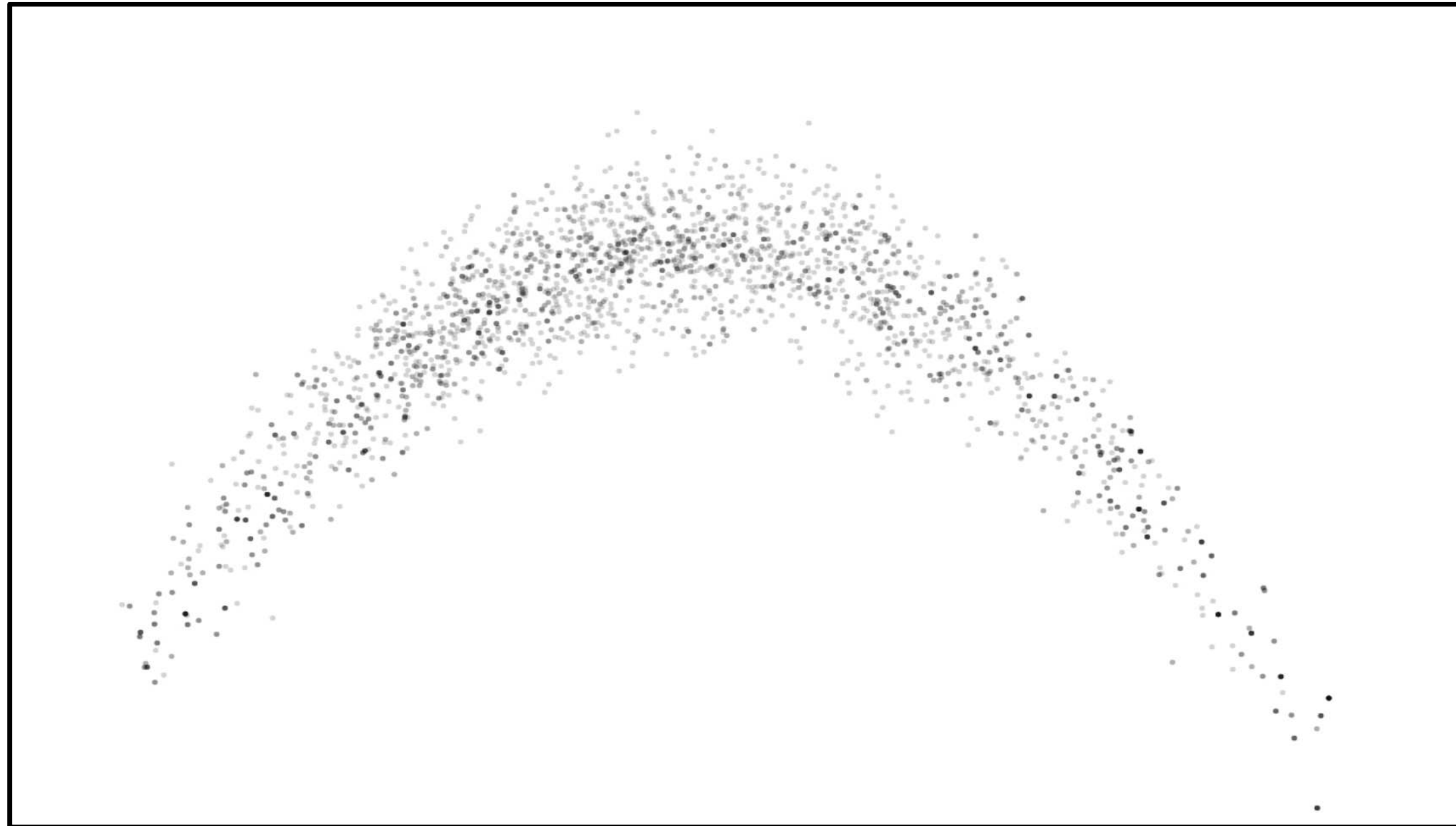
Optimization problem:
$$\max_x f(\mathbf{x})$$

Gradient descent/ascent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

↑
scalar step size

LMC OVERVIEW



Sampling problem:

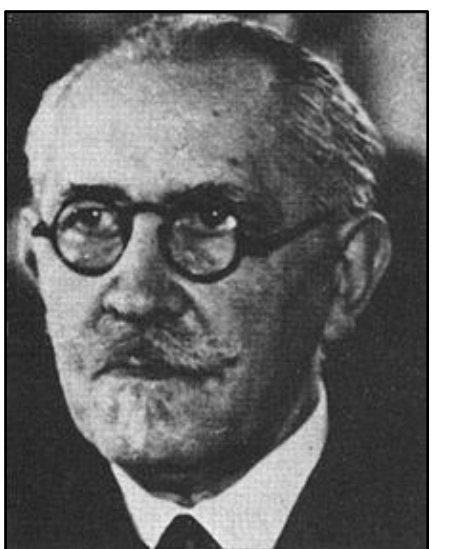
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

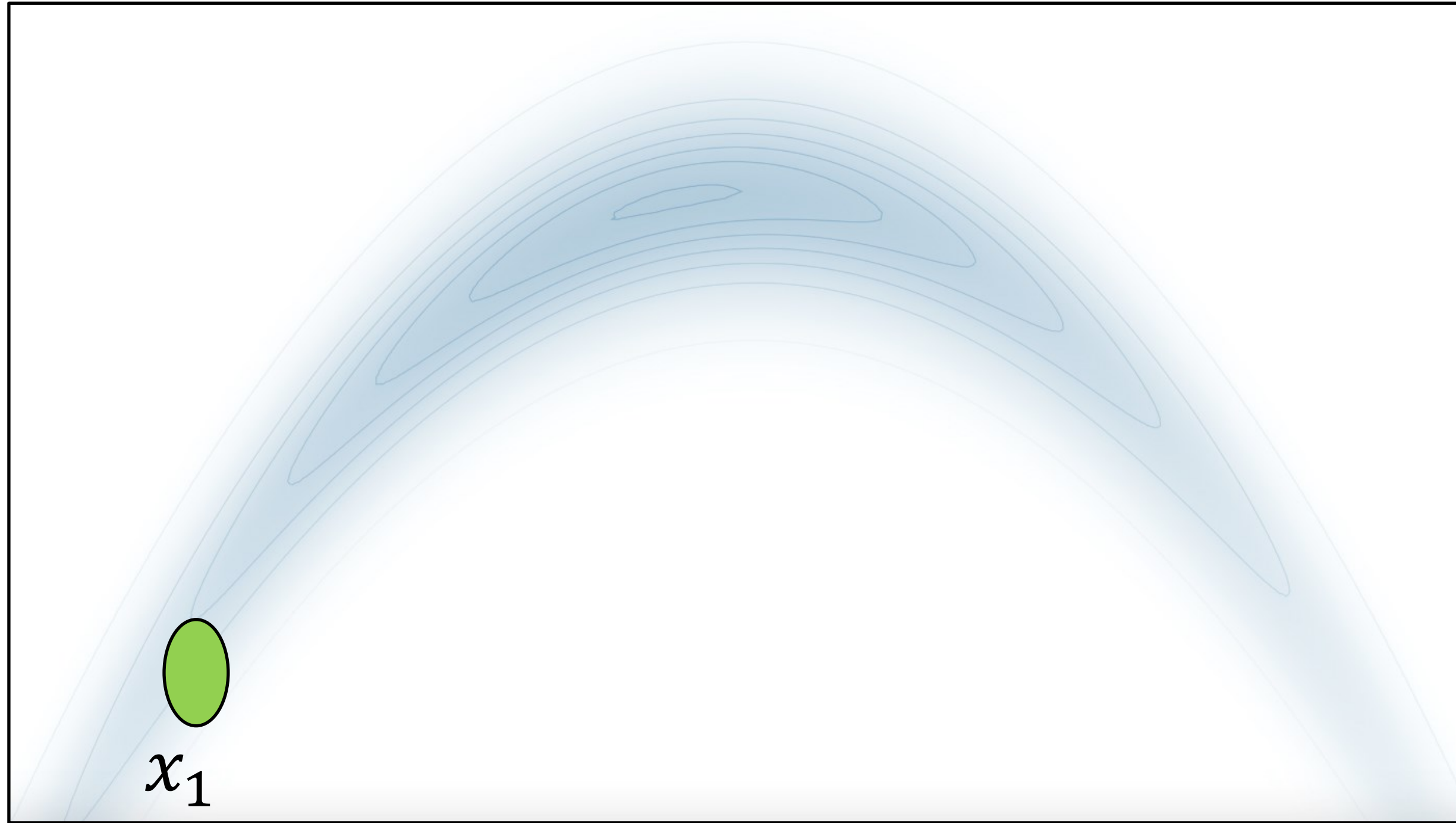
Apply Metropolis Hastings to accept/reject

Paul Langevin



[Luan et al., 2020]

LMC OVERVIEW



Sampling problem:

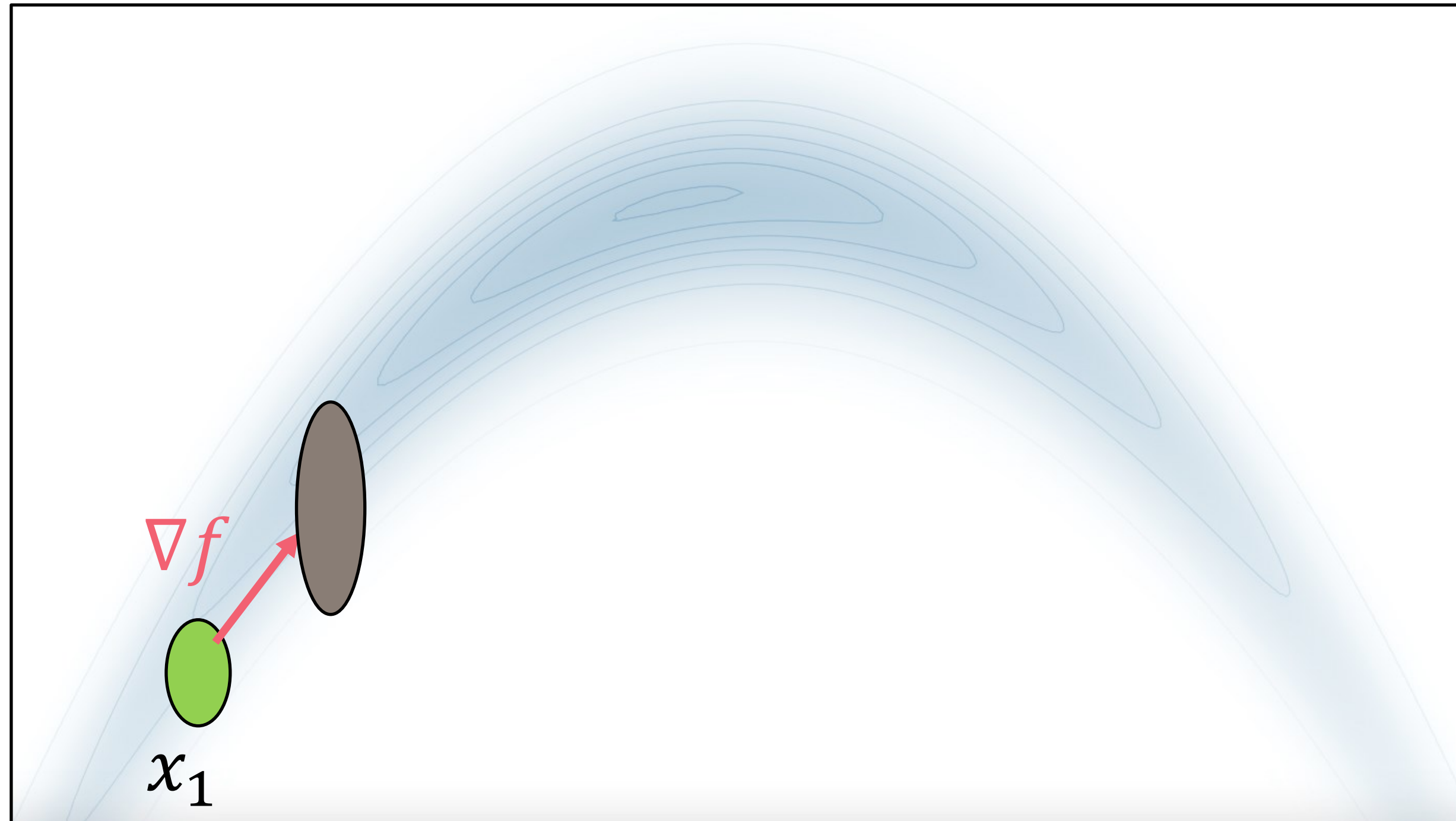
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

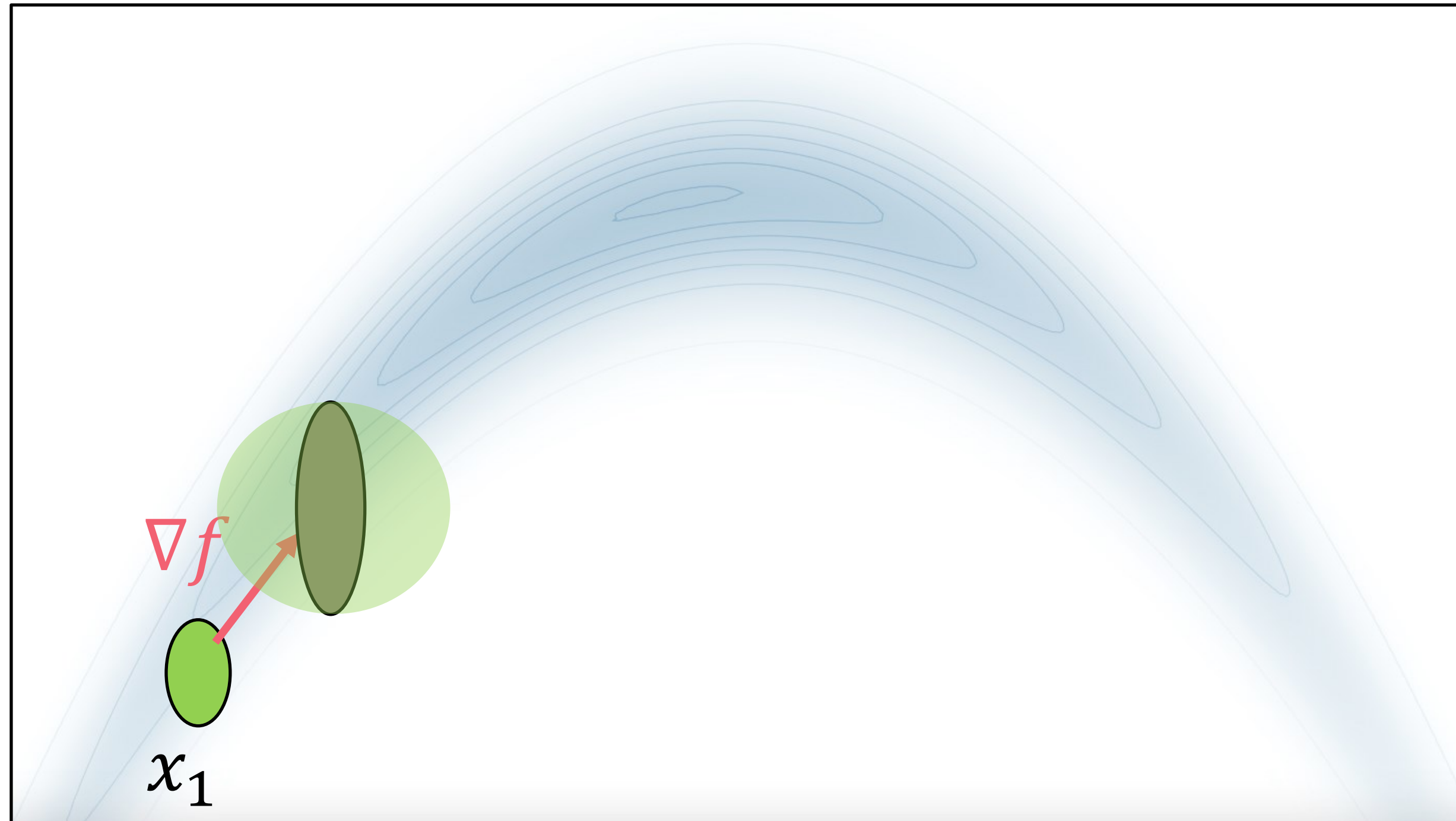
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

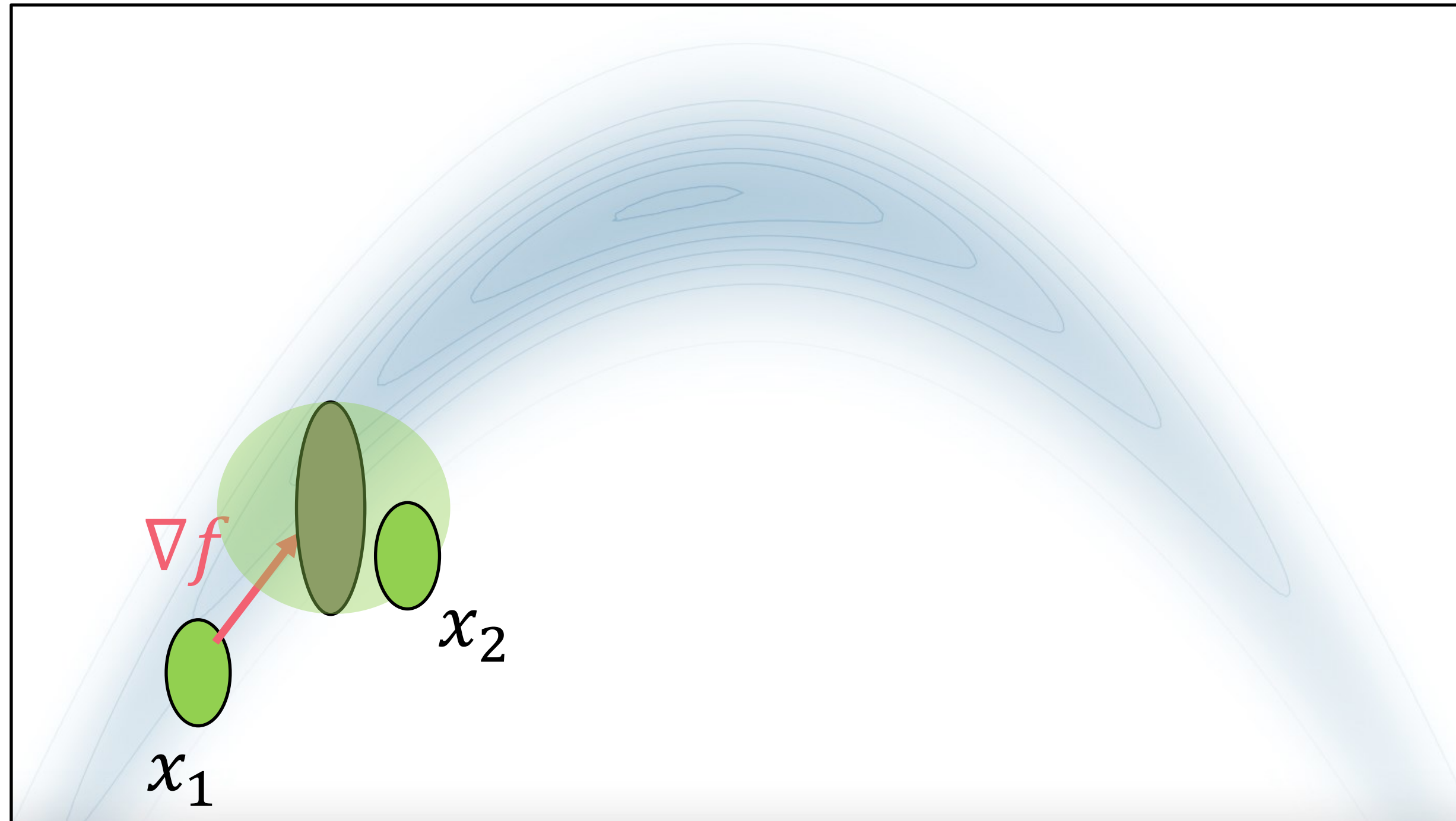
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

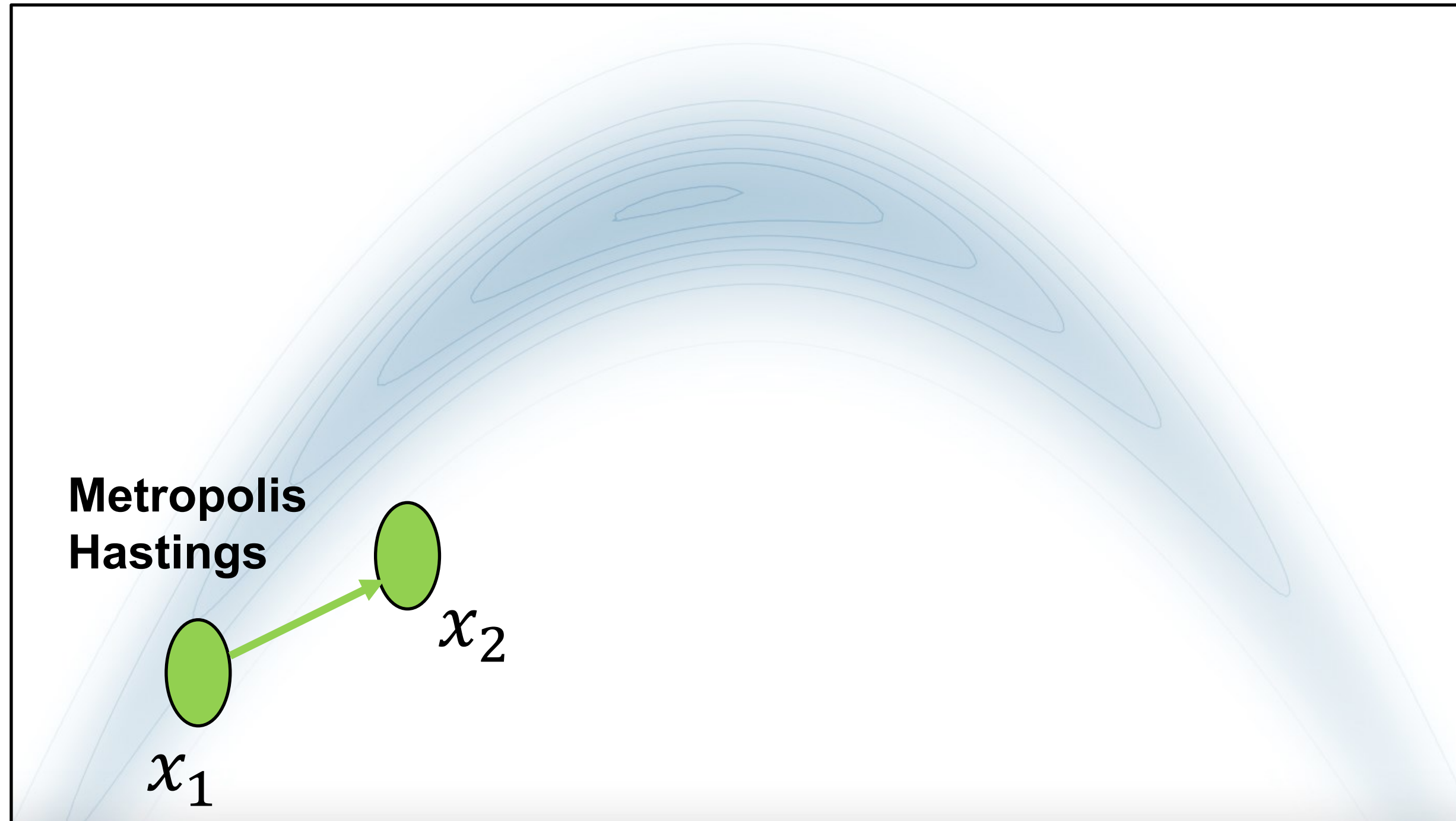
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

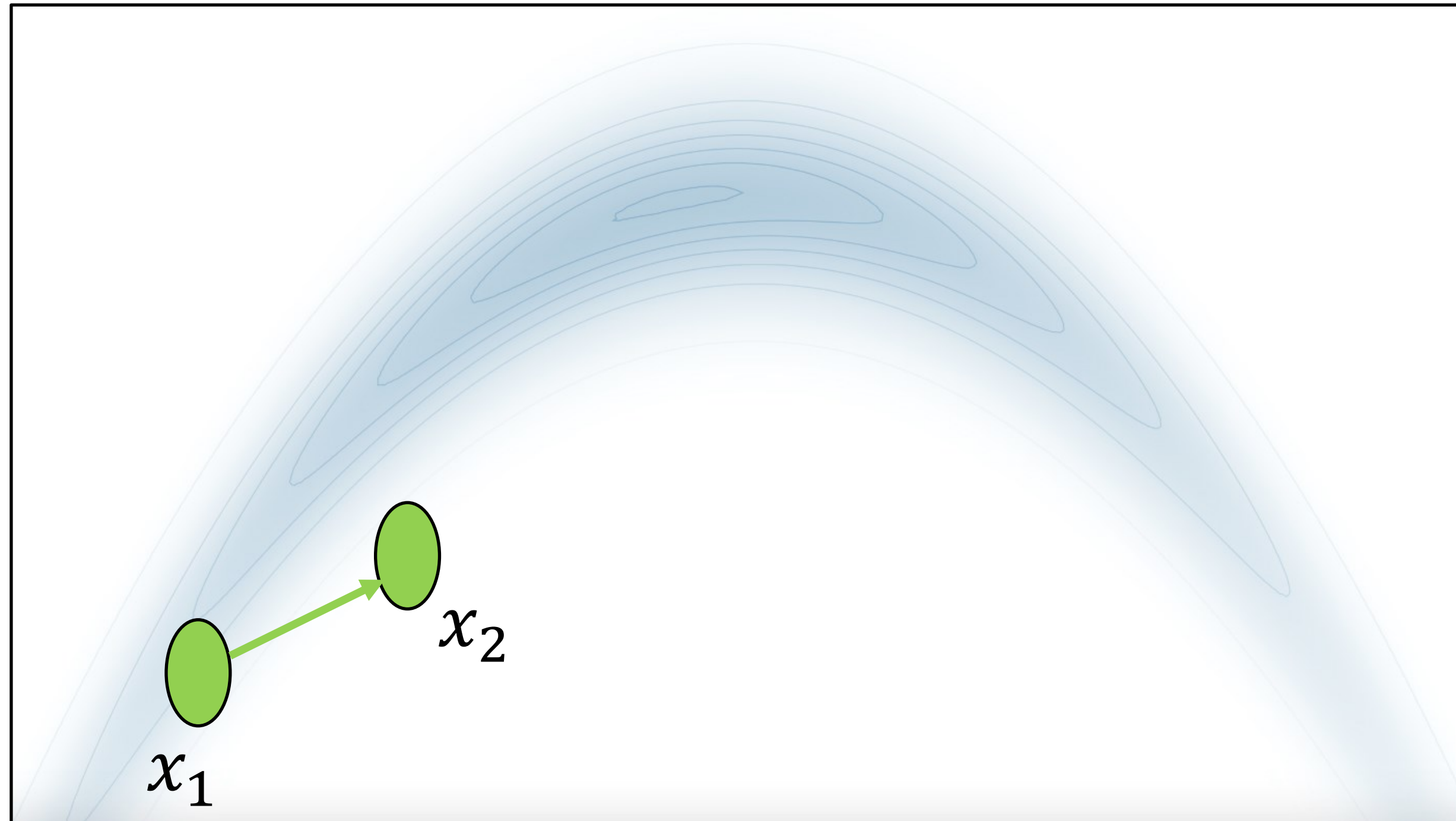
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

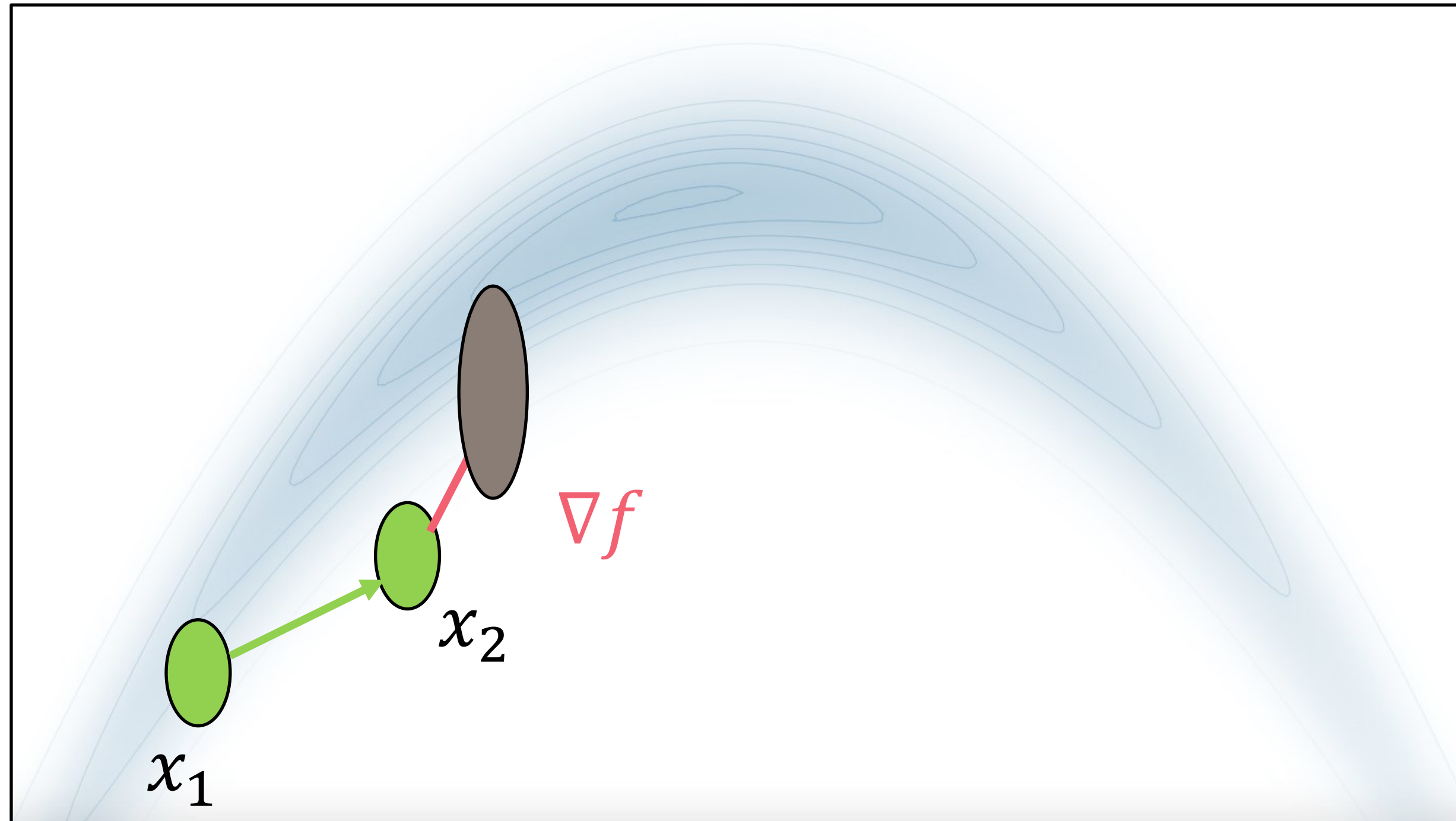
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

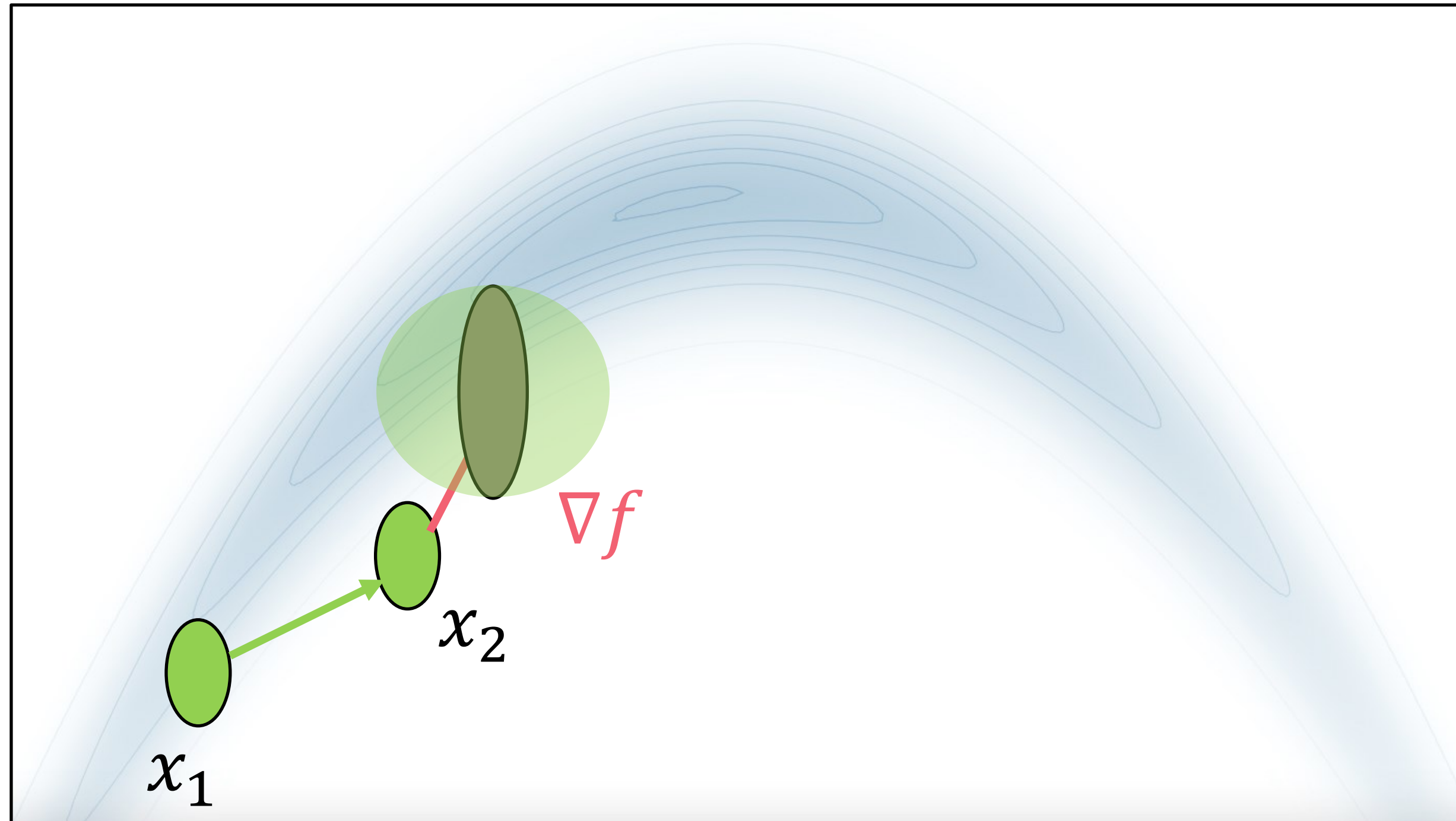
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

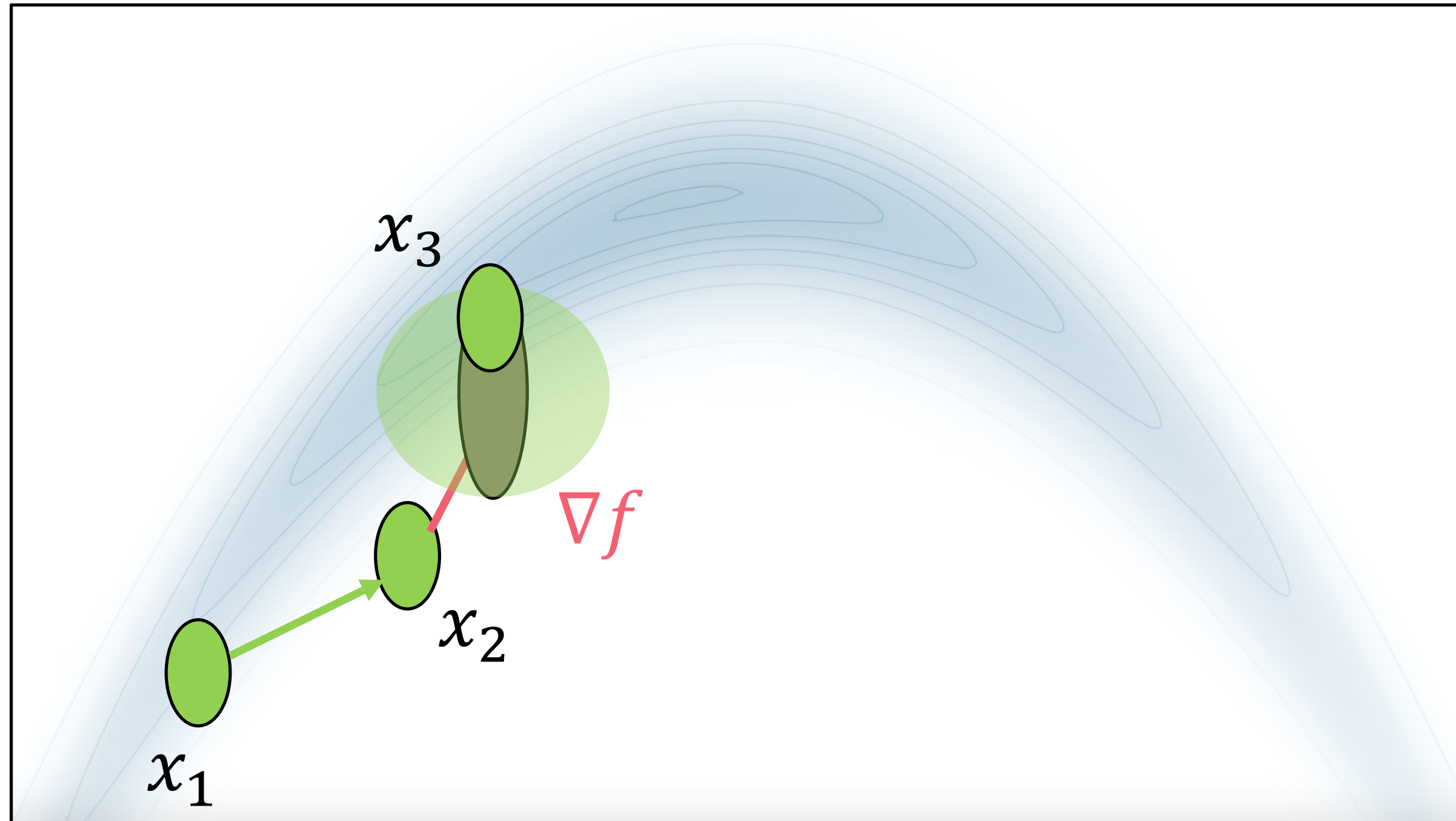
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

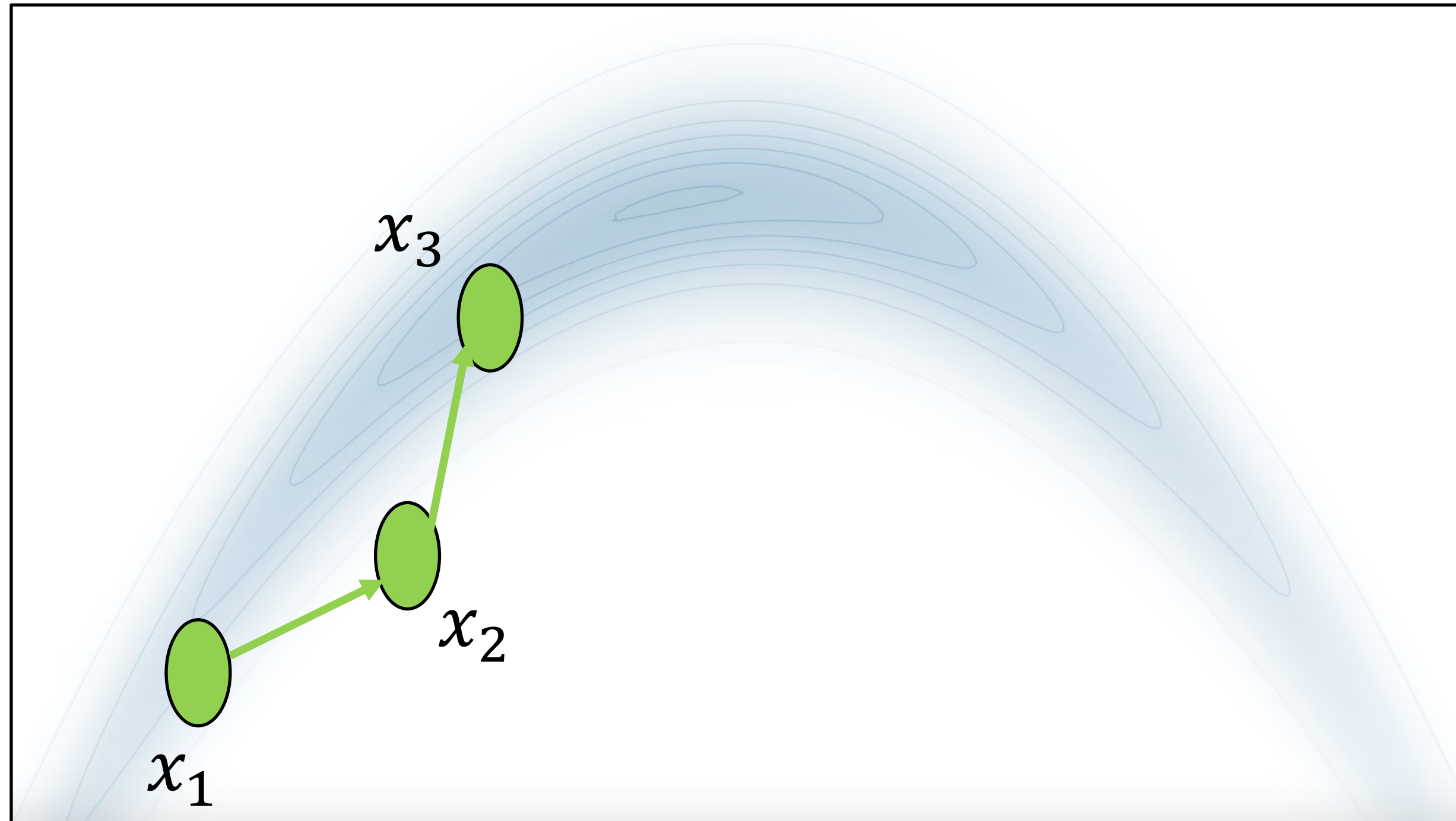
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

LMC OVERVIEW



Sampling problem:

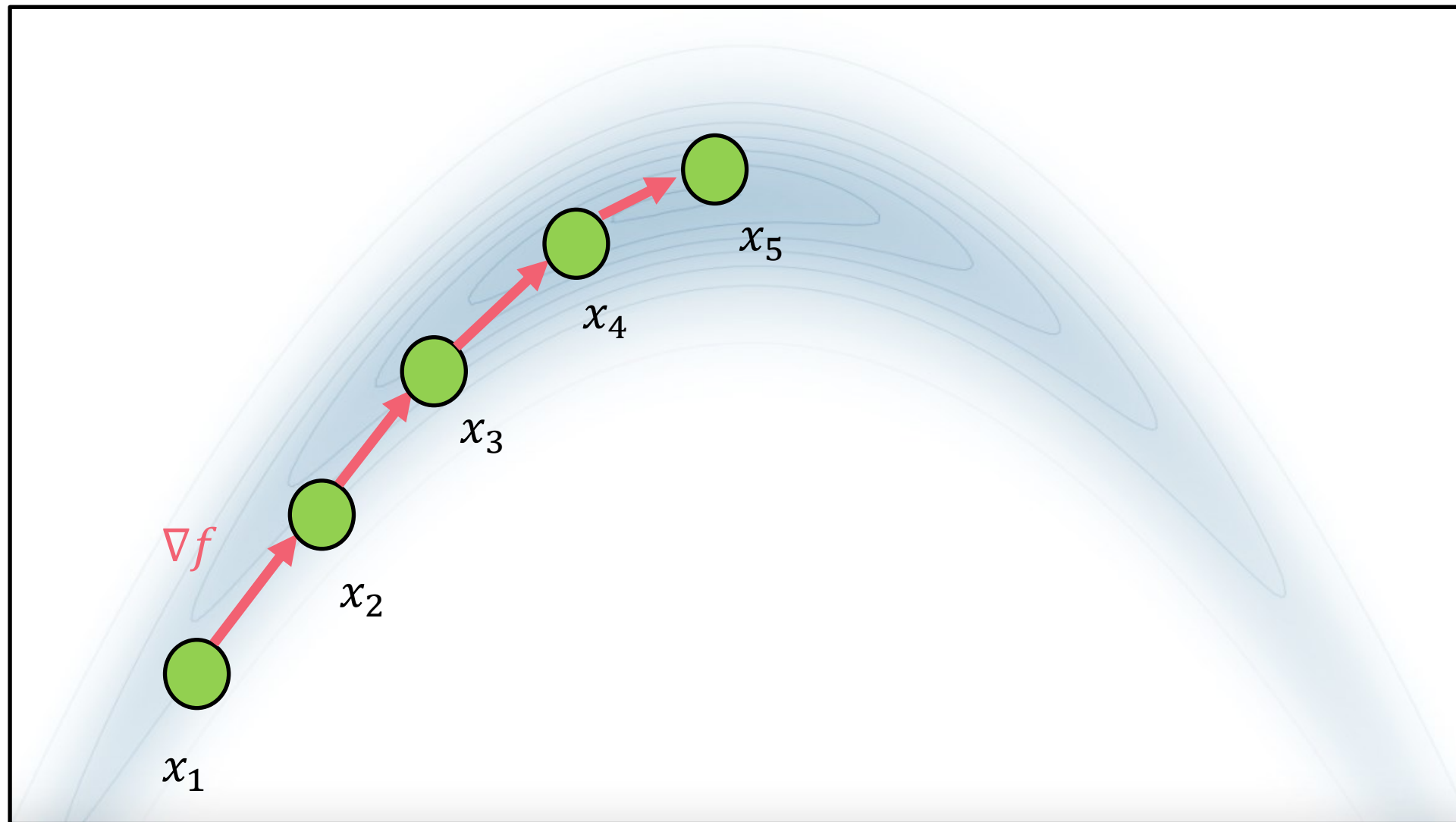
$$\mathbf{x}_t \sim f$$

Langevin MC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

Apply Metropolis Hastings to accept/reject

GD VS. LMC

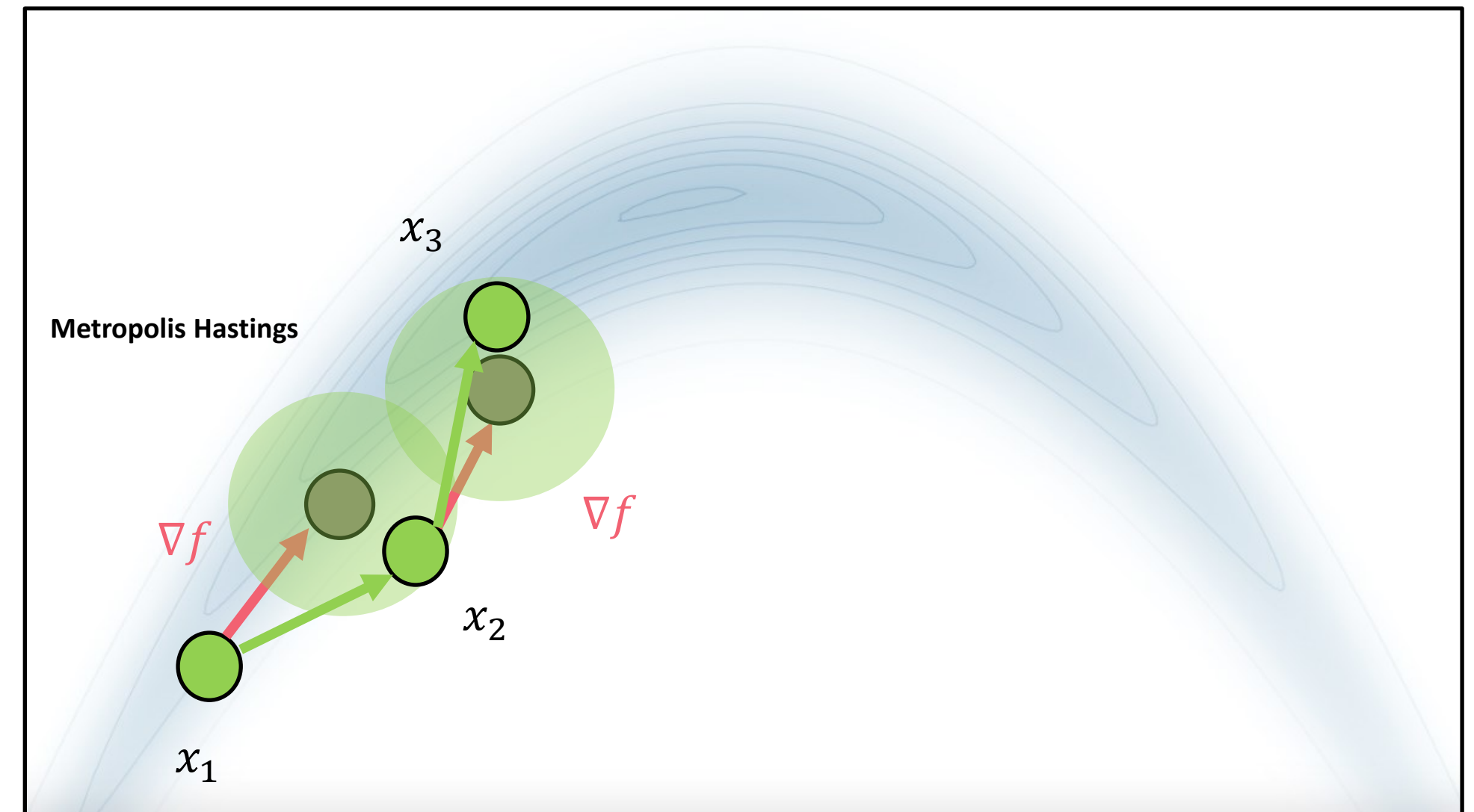


Optimization:

$$\max_{\mathbf{x}} f(\mathbf{x})$$

GD:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$



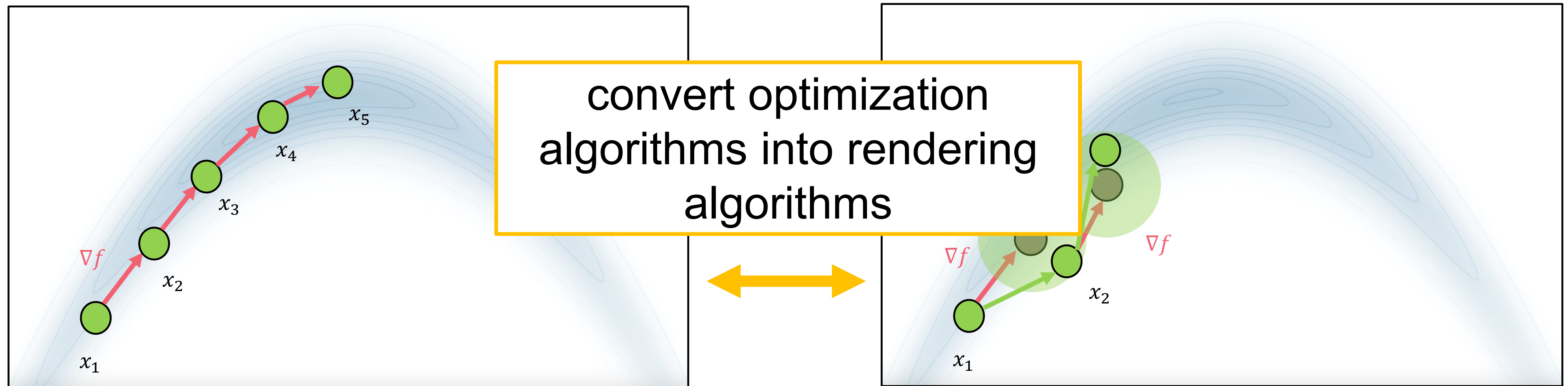
Sampling:

$$\mathbf{x}_t \sim f$$

LMC:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$

GD VS. LMC



Optimization:

$$\max_{\mathbf{x}} f(\mathbf{x})$$

GD:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1})$$

Sampling:

$$\mathbf{x}_t \sim f$$

LMC:

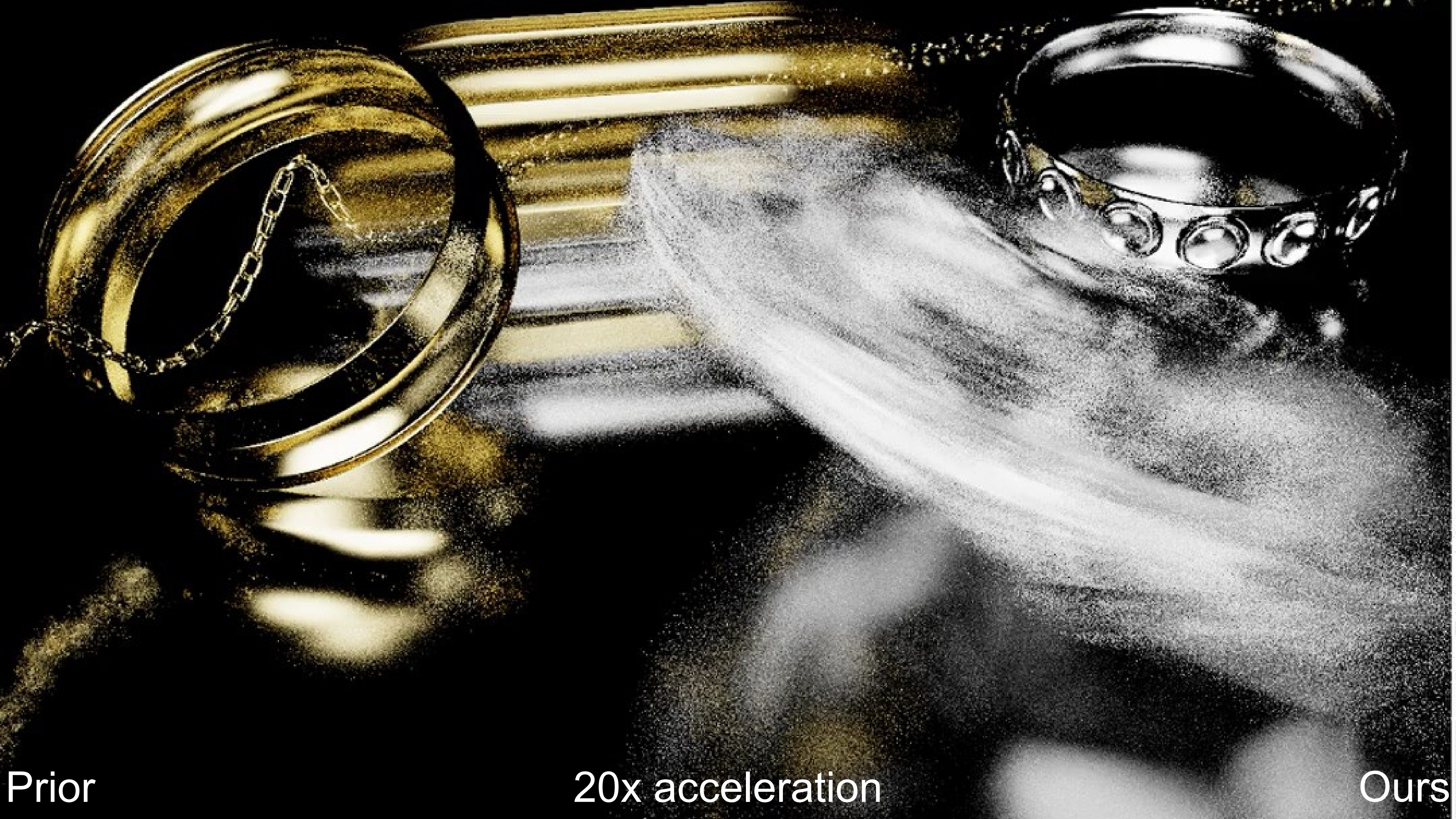
$$\mathbf{x}_t = \mathbf{x}_{t-1} + s_{t-1} \nabla f(\mathbf{x}_{t-1}) + \frac{1}{s_{t-1}} N(0, \sigma^2 \mathbf{I})$$



Prior

10x acceleration

Ours

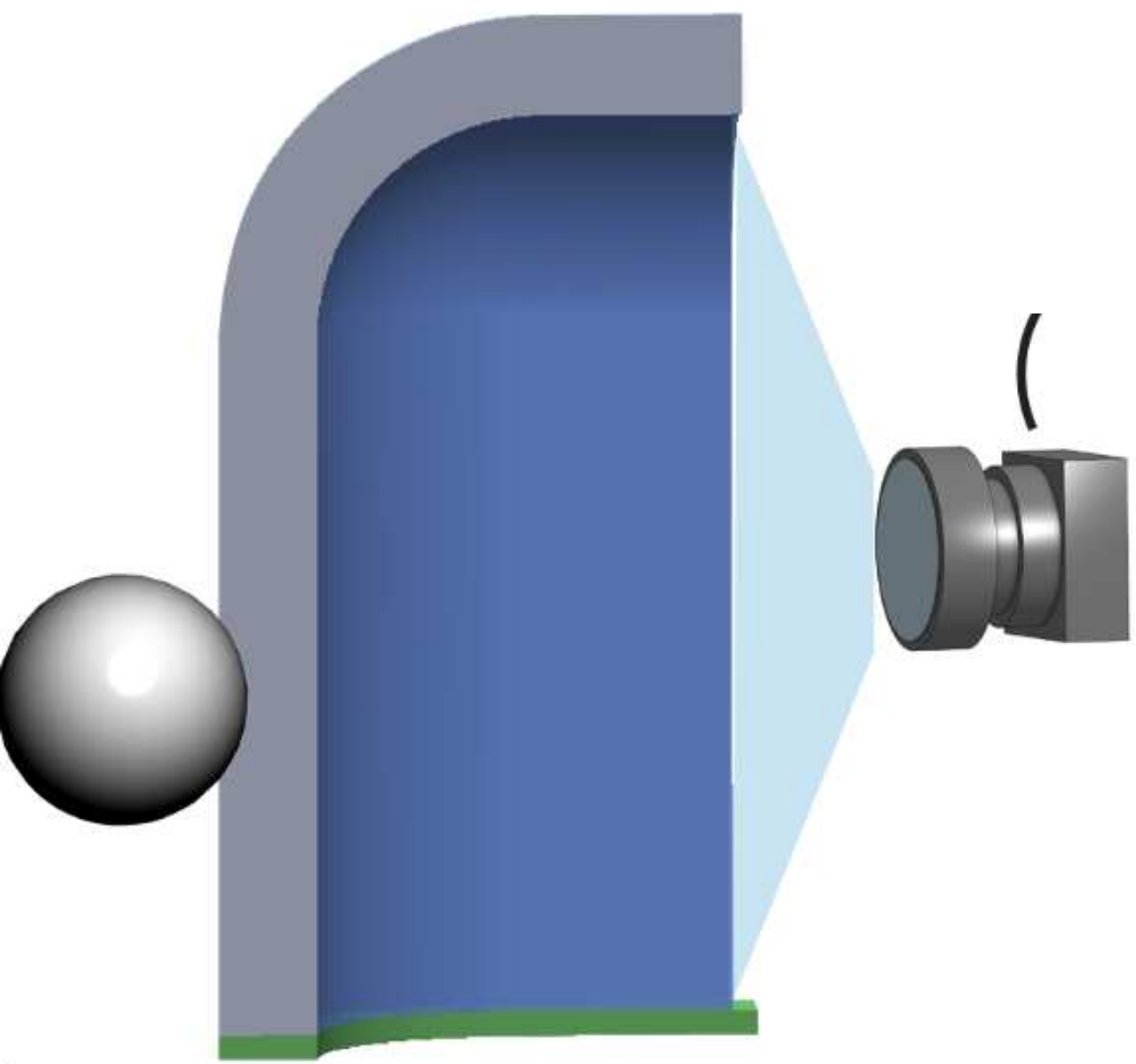


Prior

20x acceleration

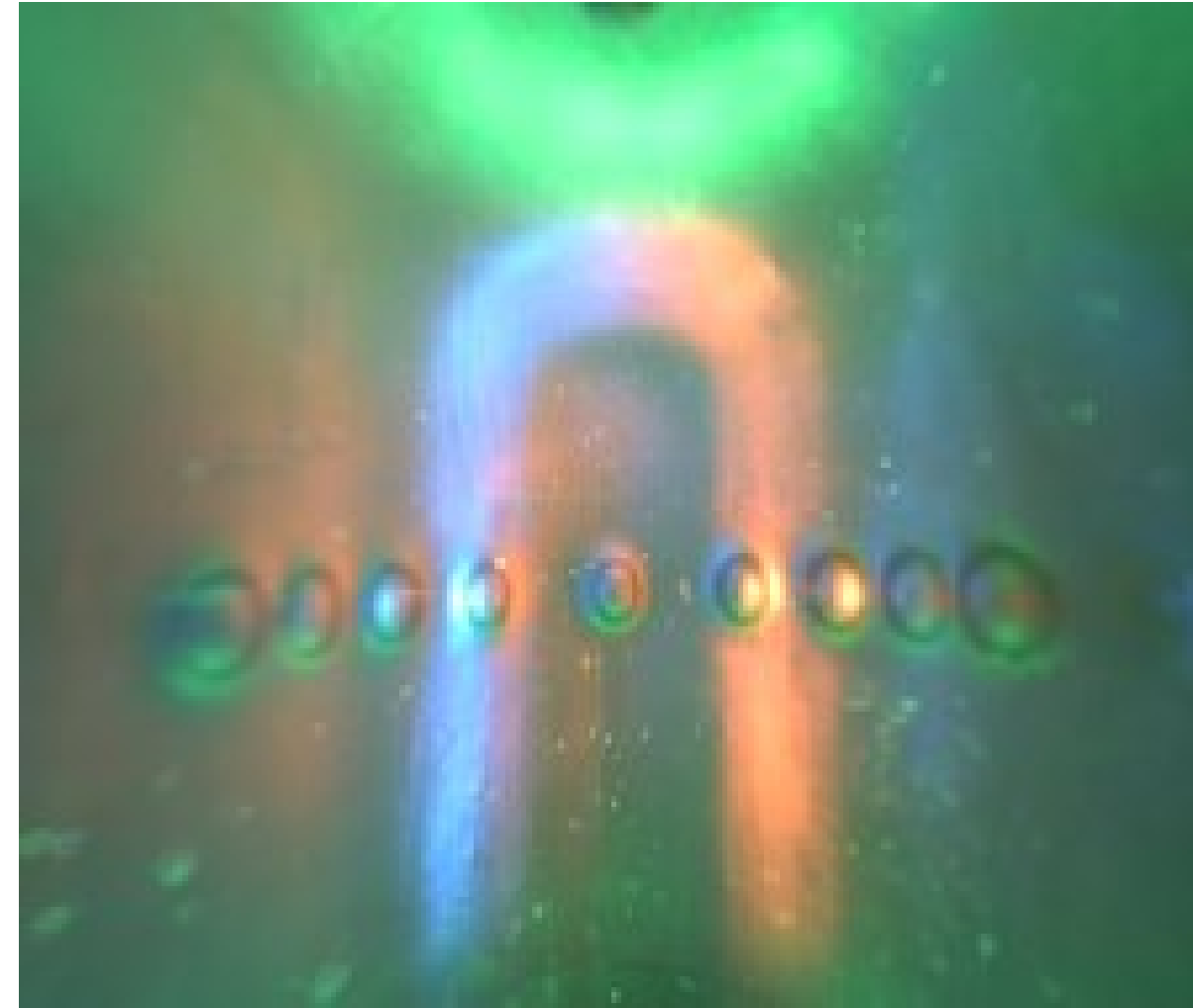
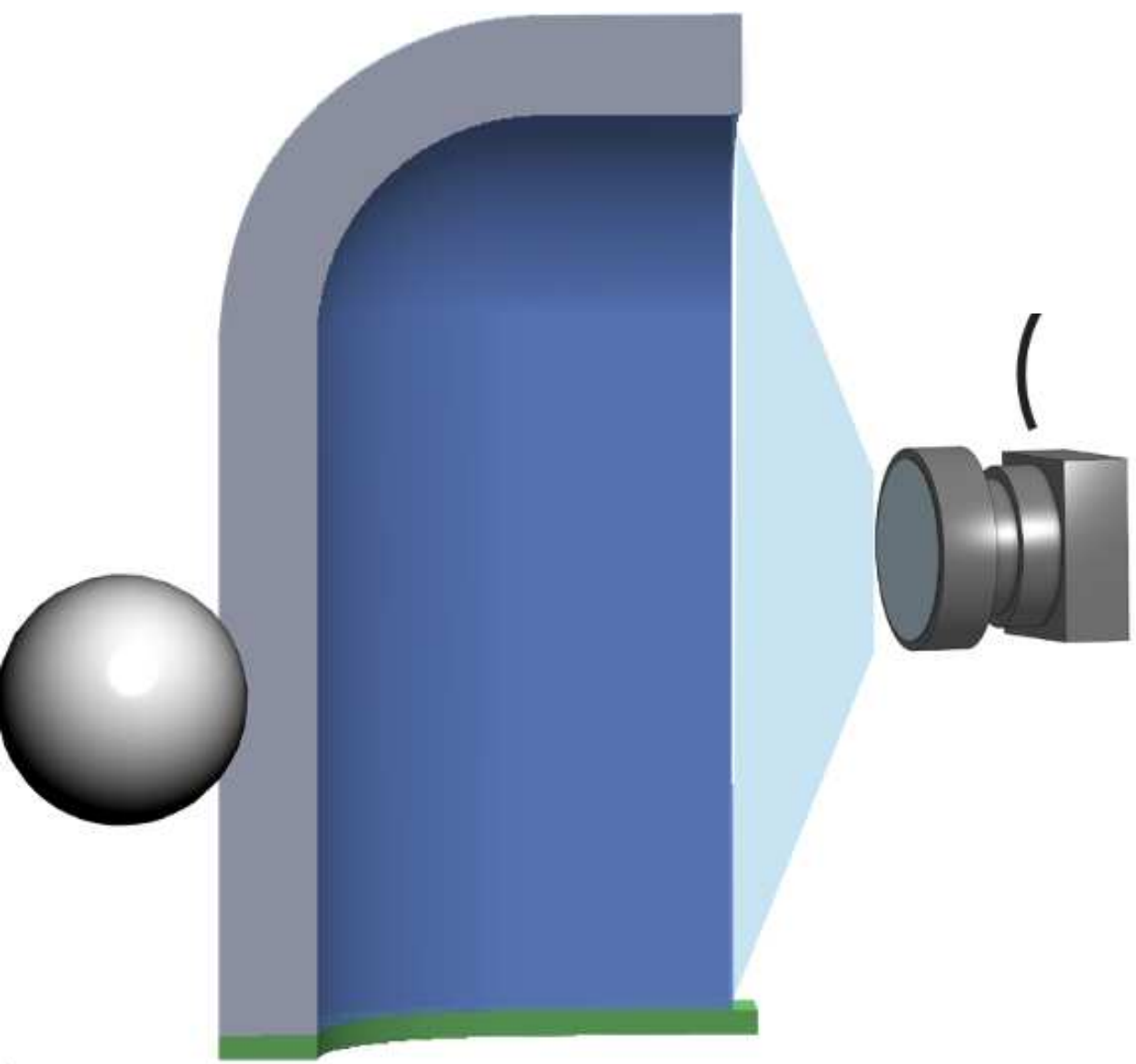
Ours

Evaluating optical simulation framework



Real-world prototype

Evaluating optical simulation framework

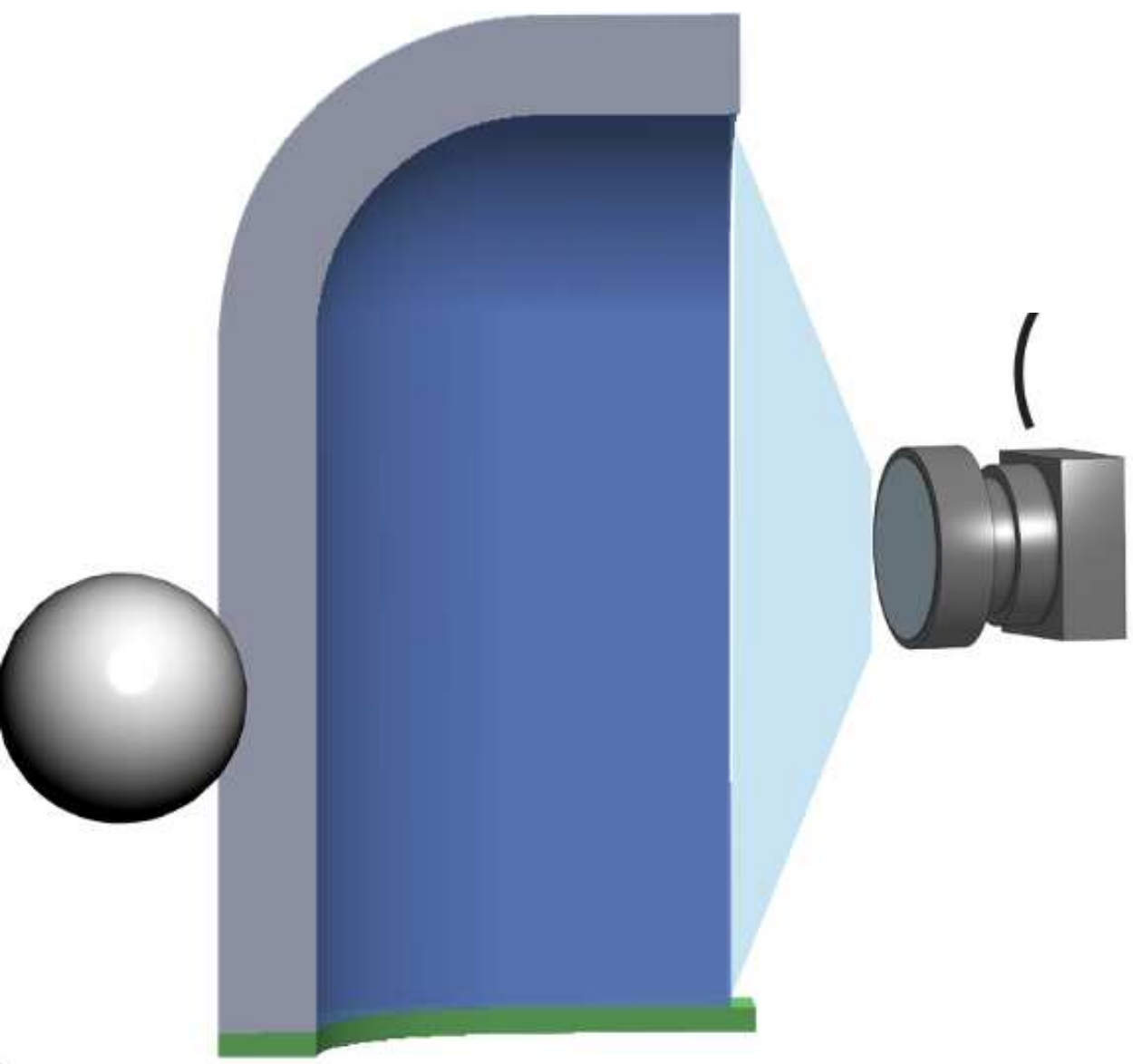


Real-world prototype



Rasterization

Evaluating optical simulation framework



Real-world prototype



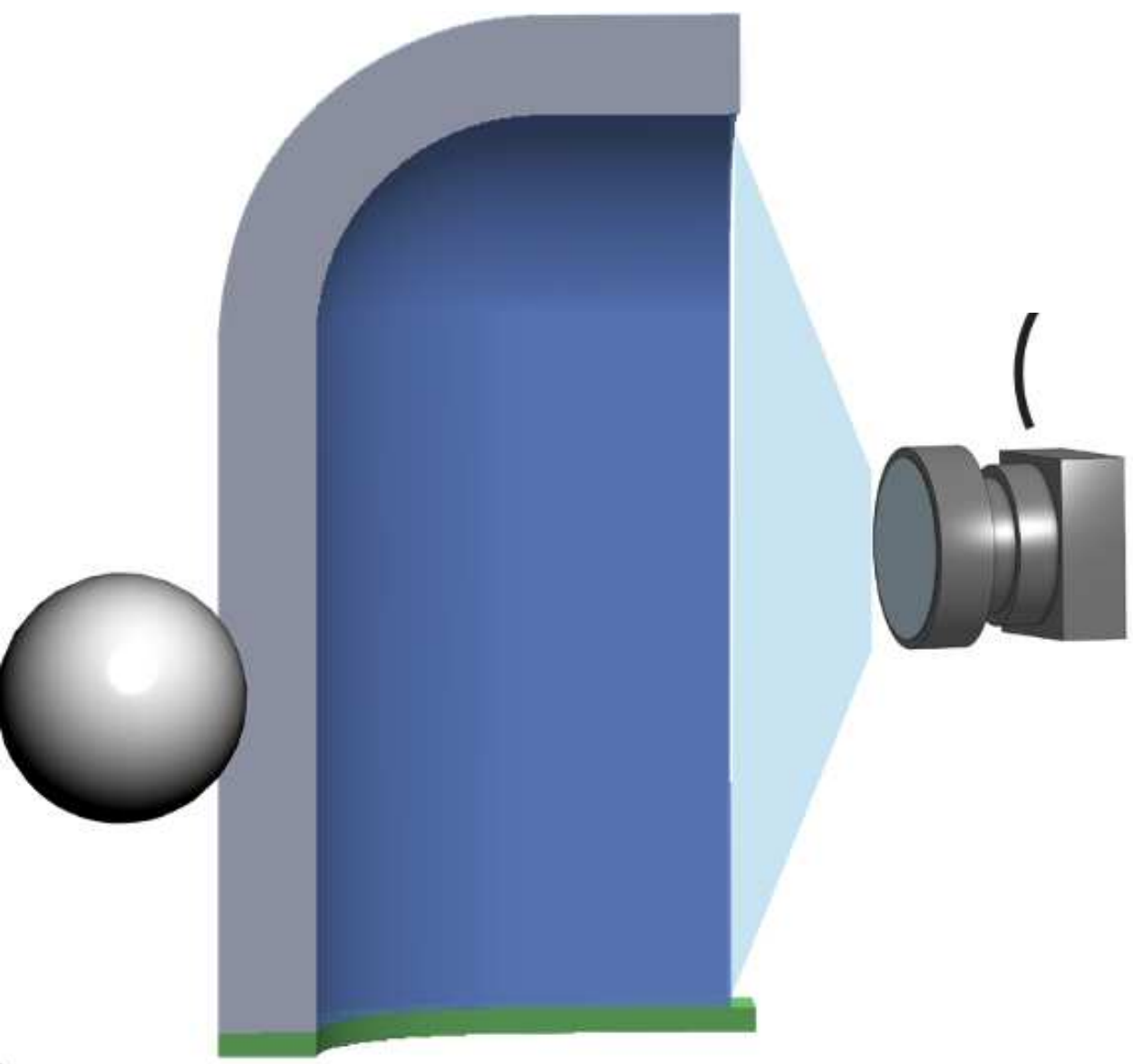
Rasterization



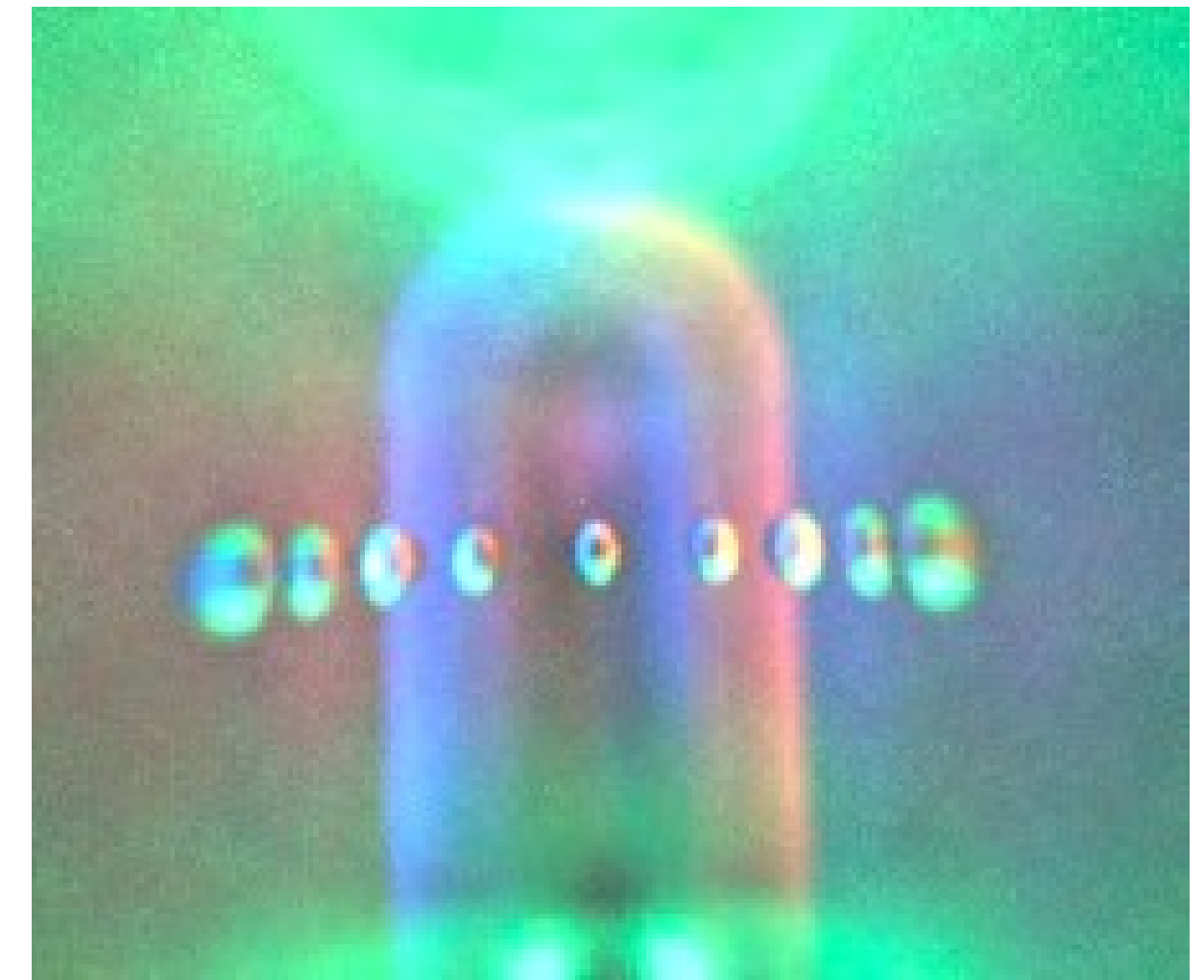
Path tracing

[Agarwal et al., 2023]

Evaluating optical simulation framework



Real-world prototype



Our simulation result

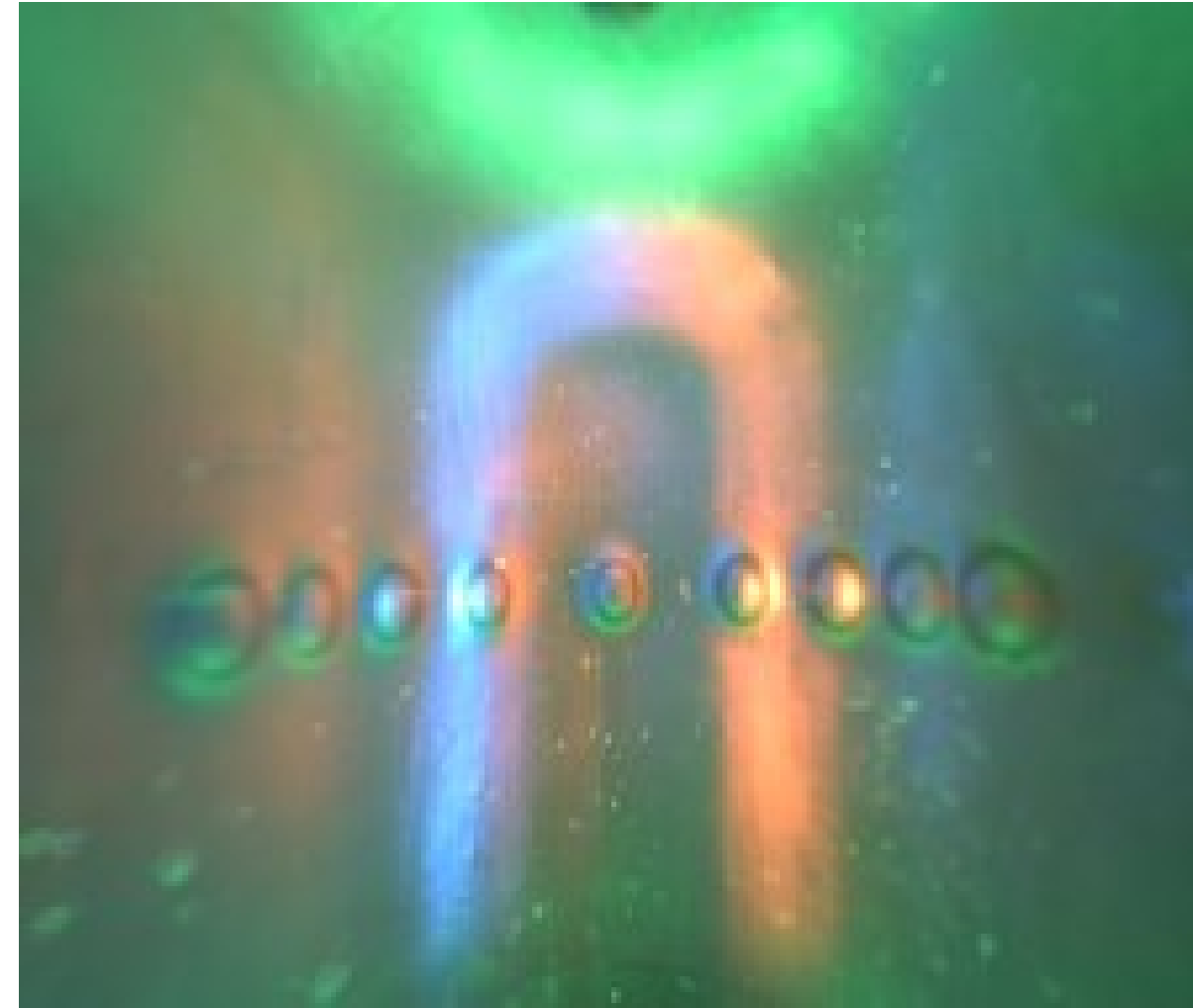
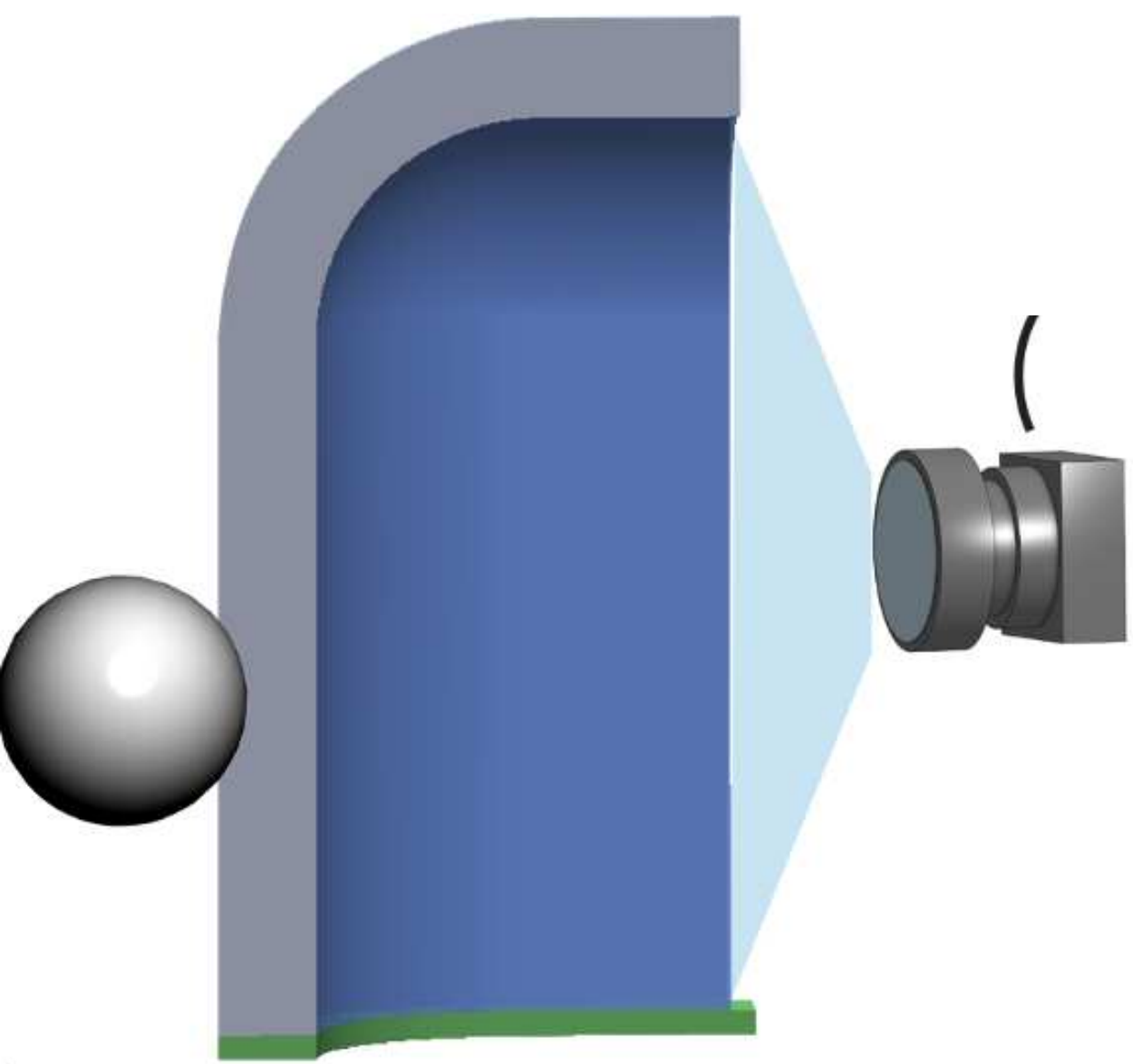


Rasterization

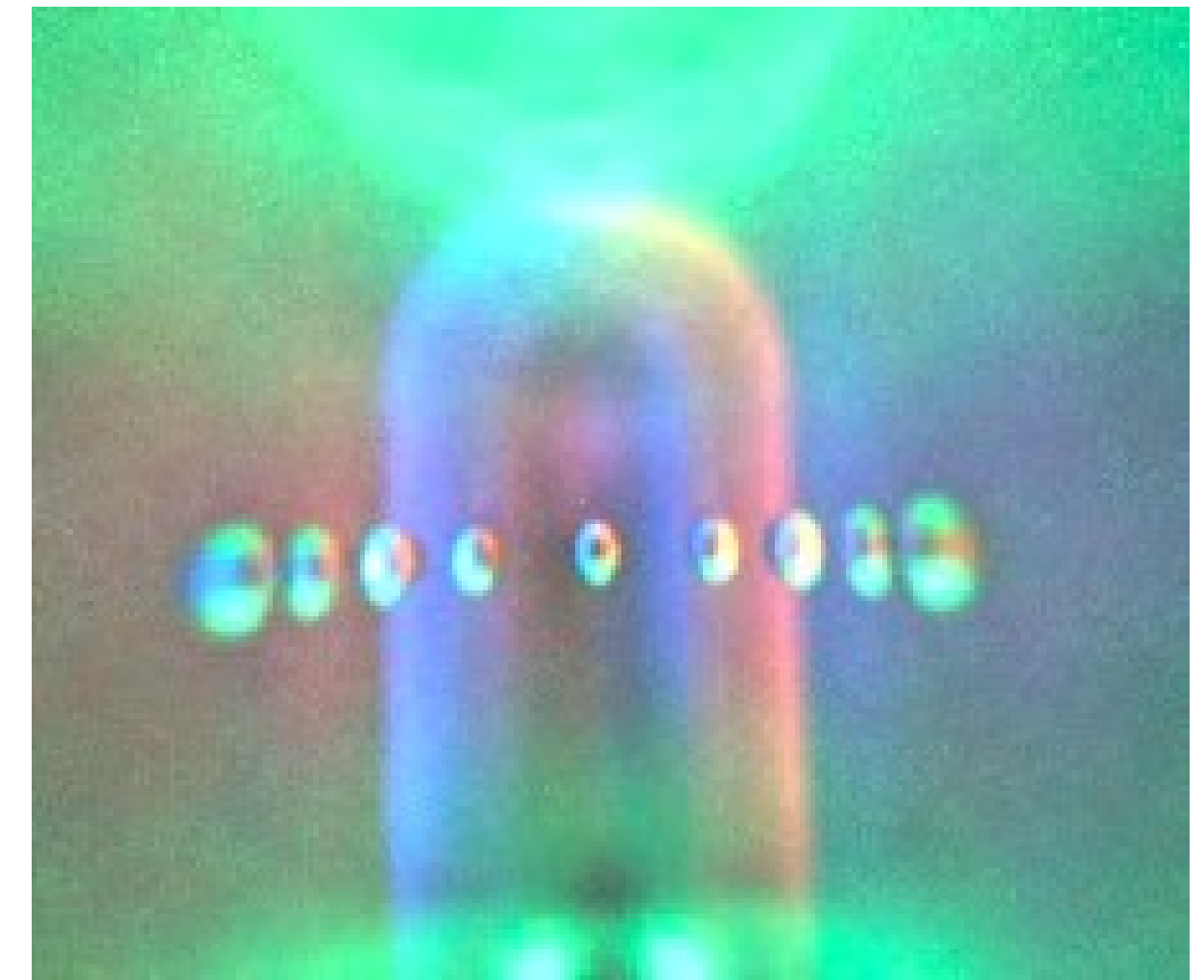


Path tracing

Evaluating optical simulation framework



Real-world prototype



Our simulation result



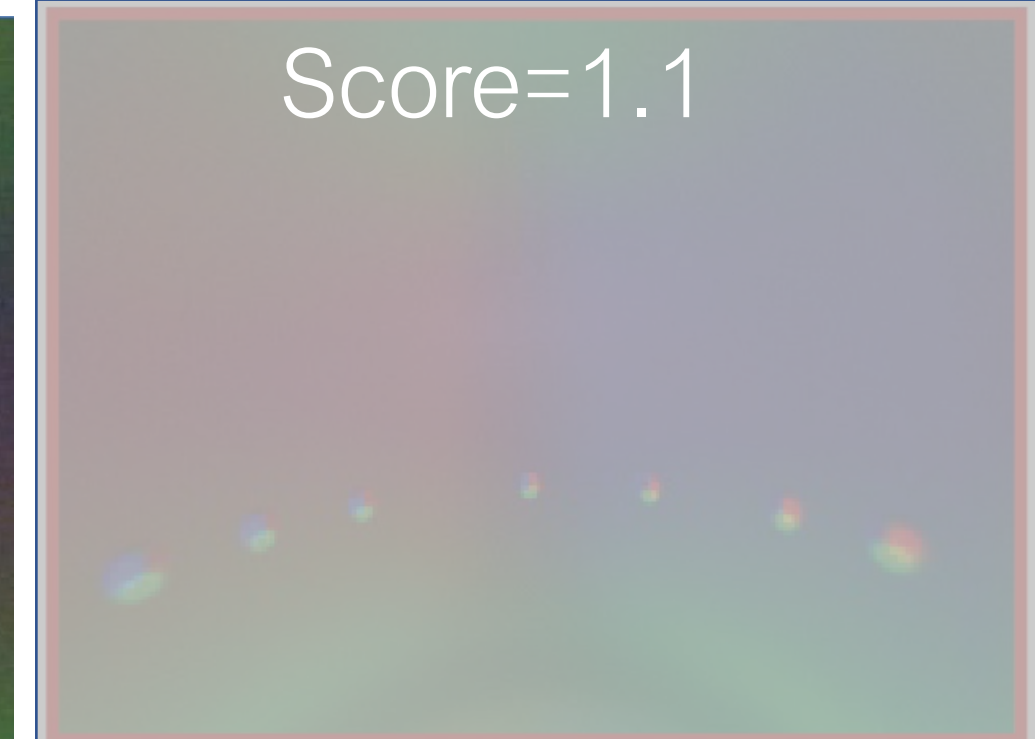
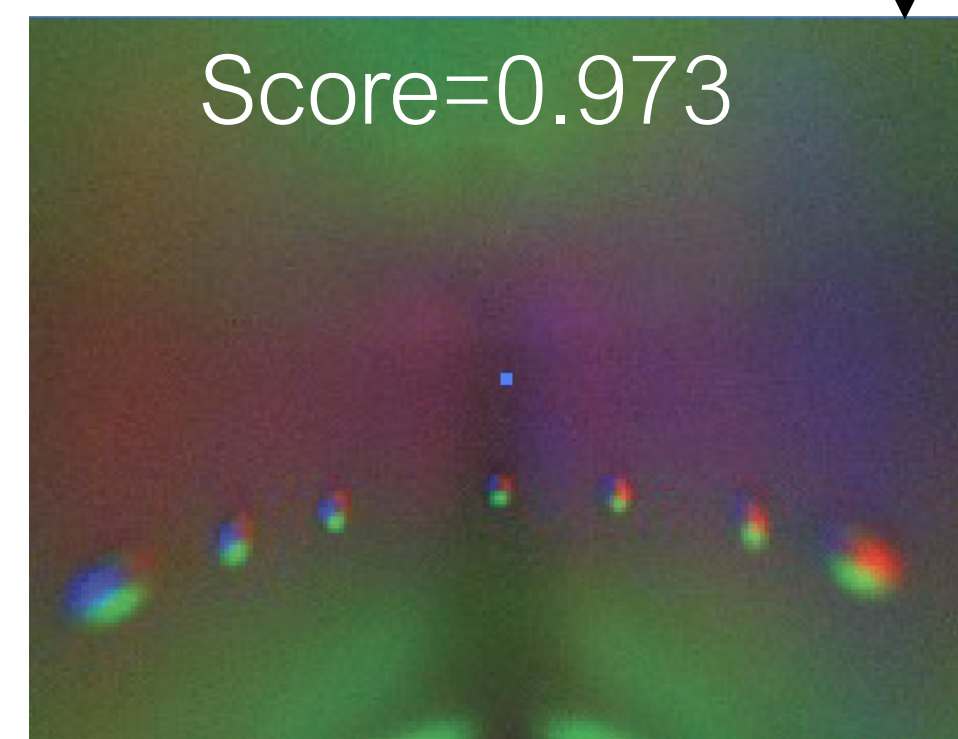
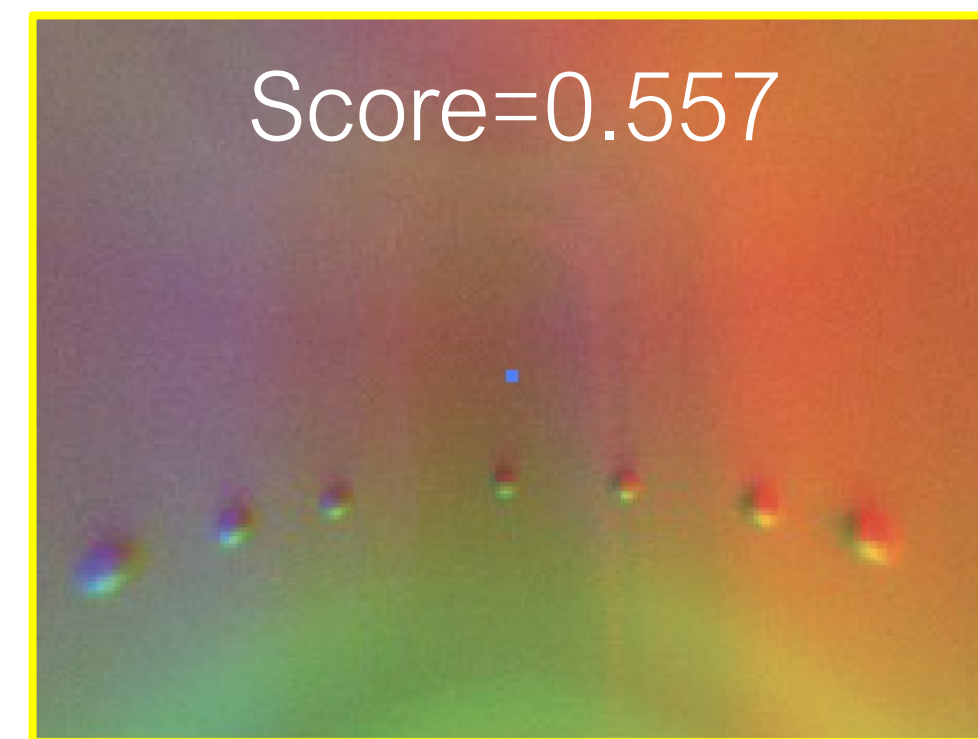
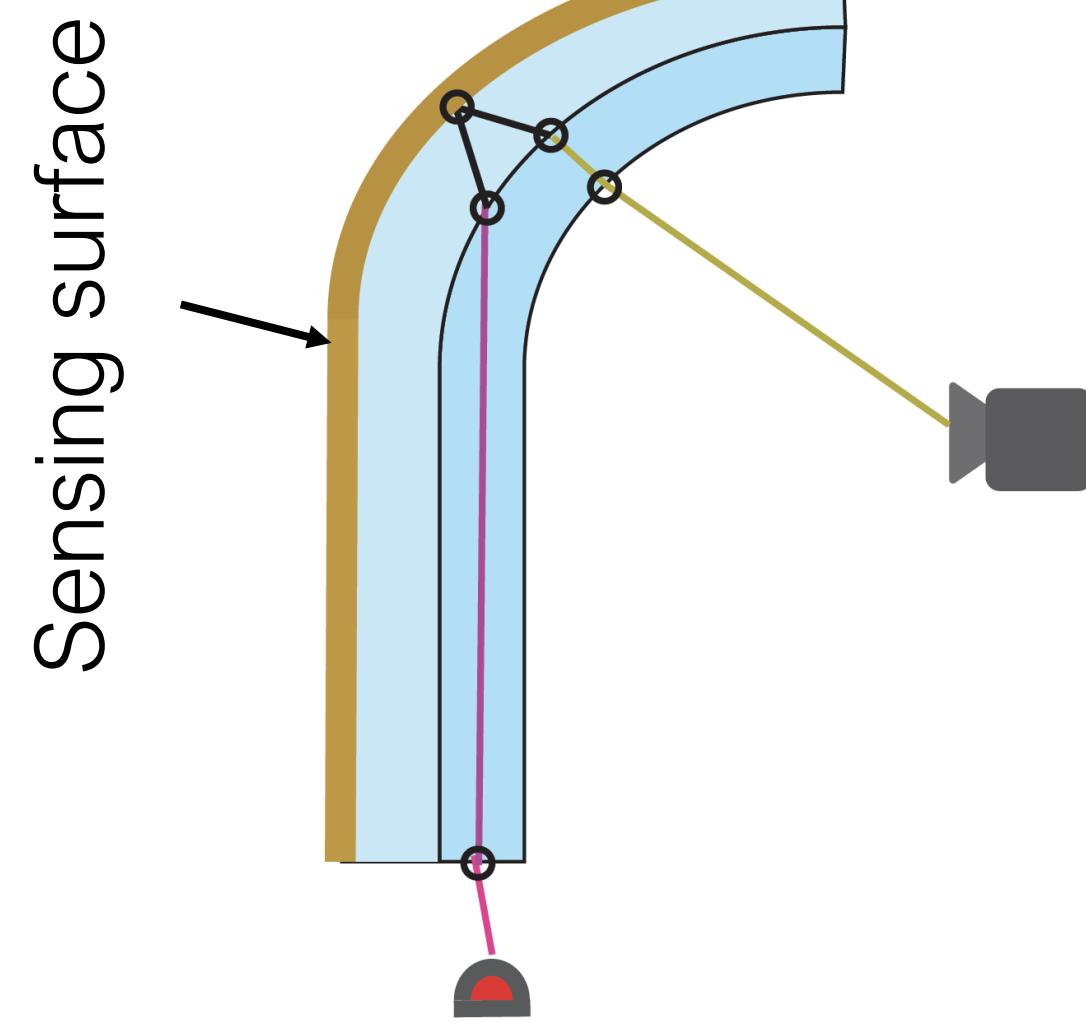
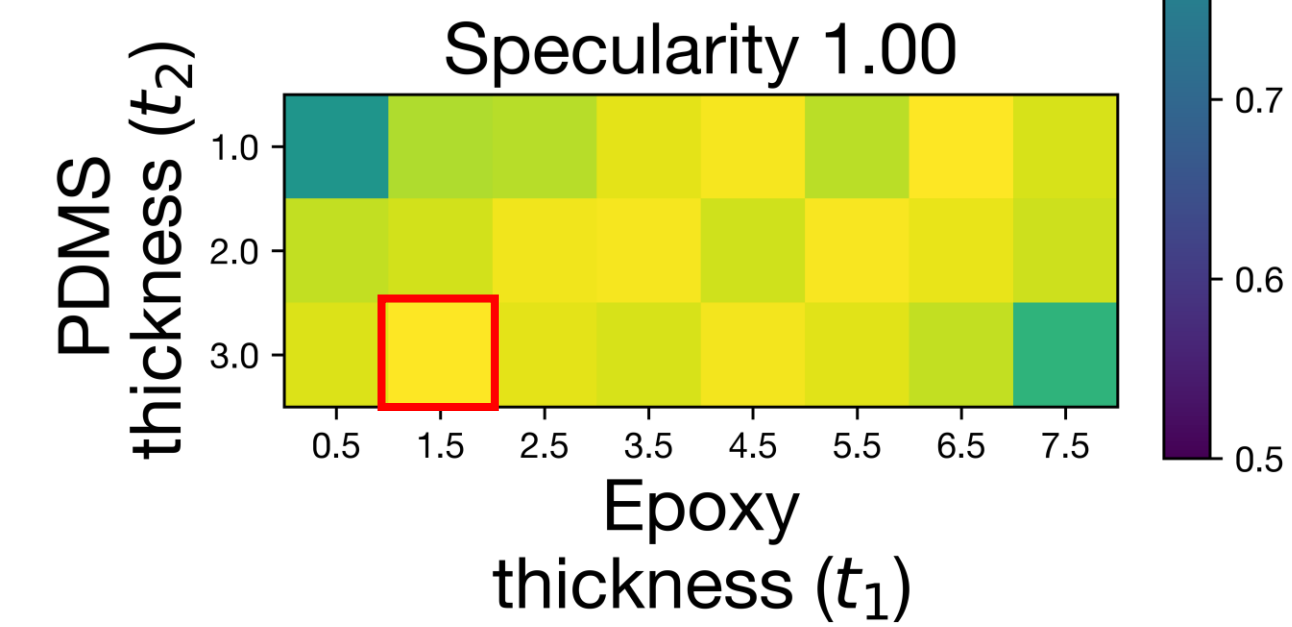
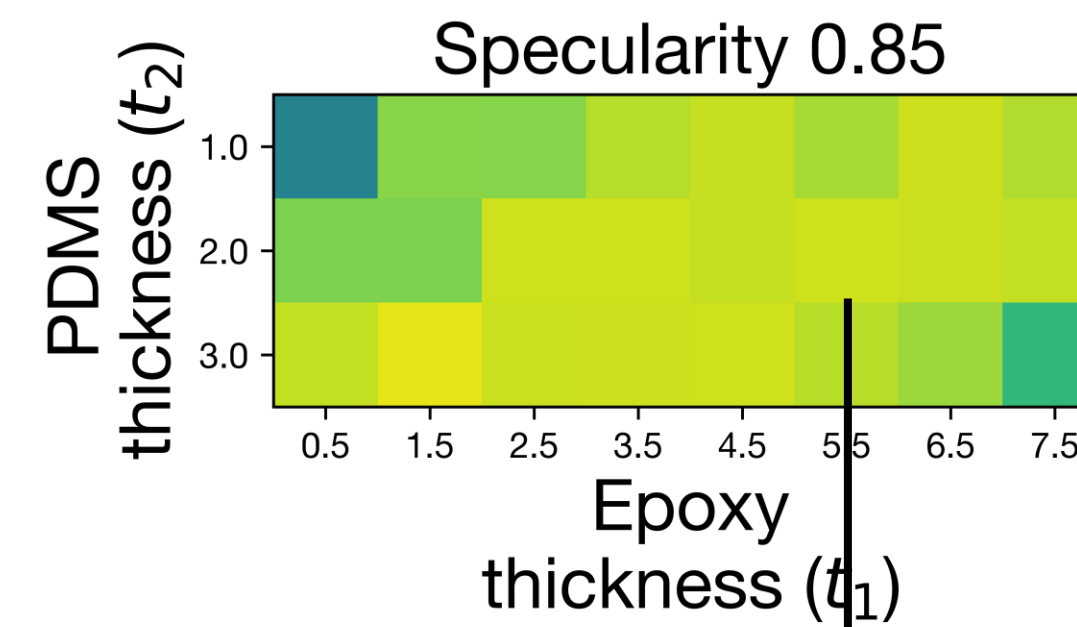
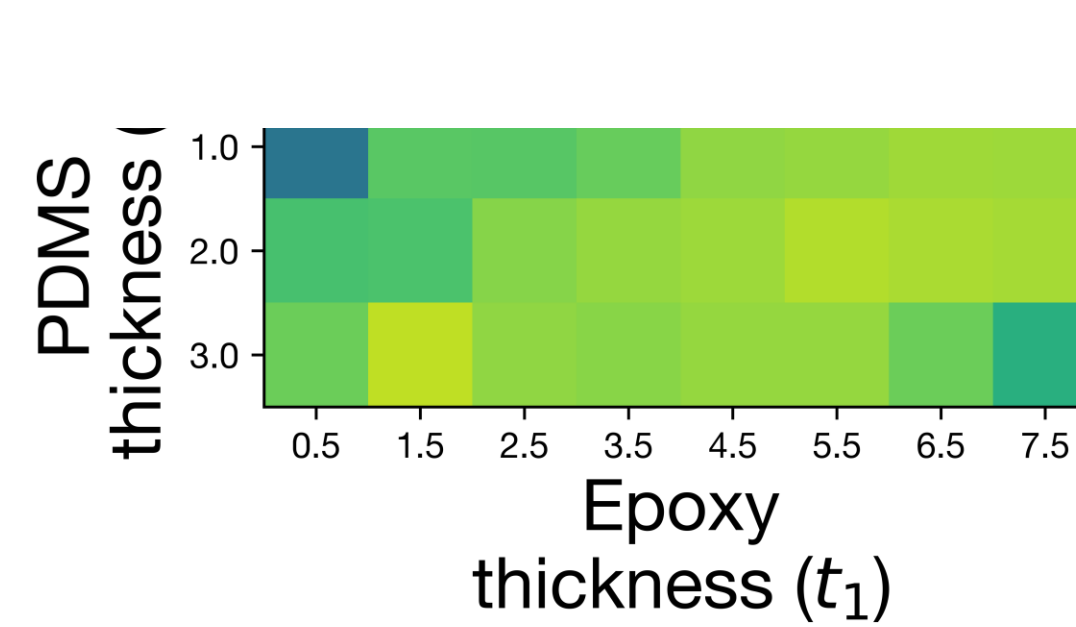
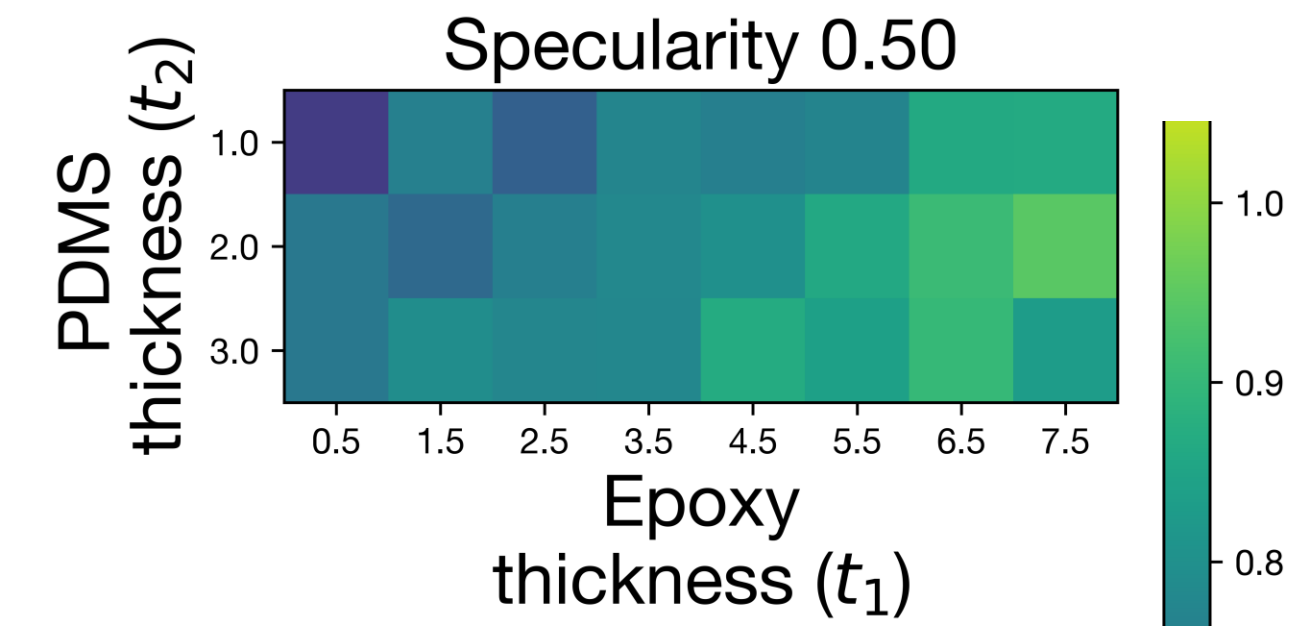
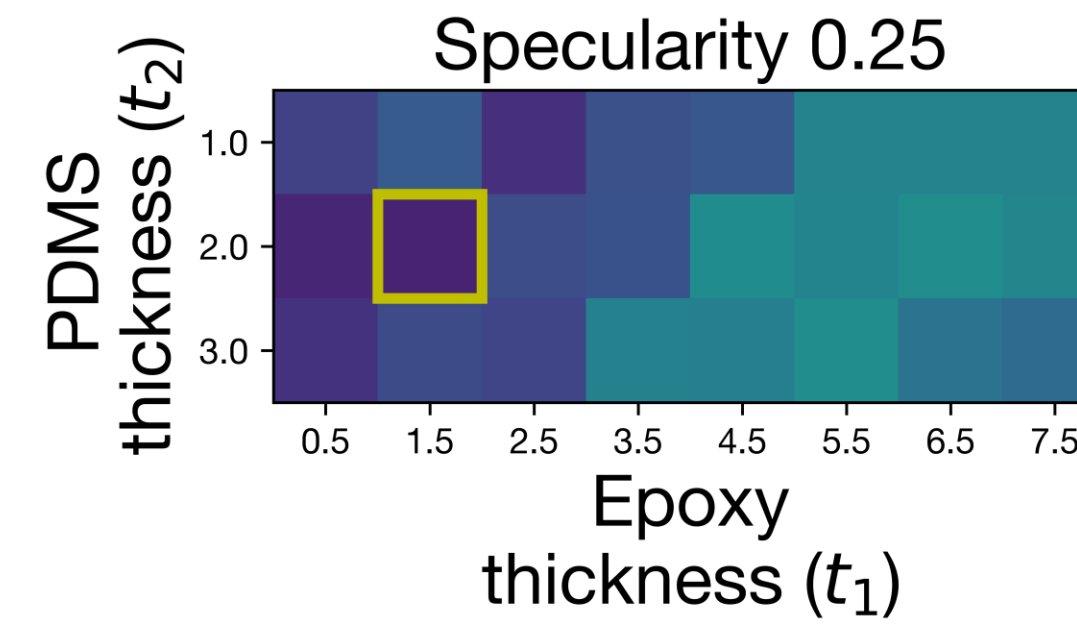
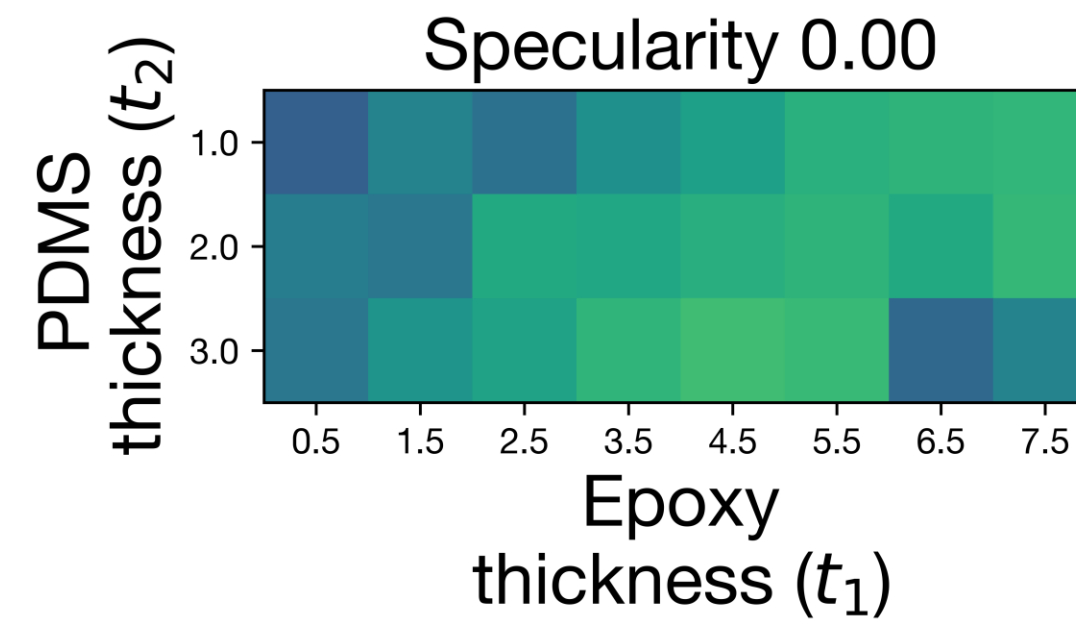
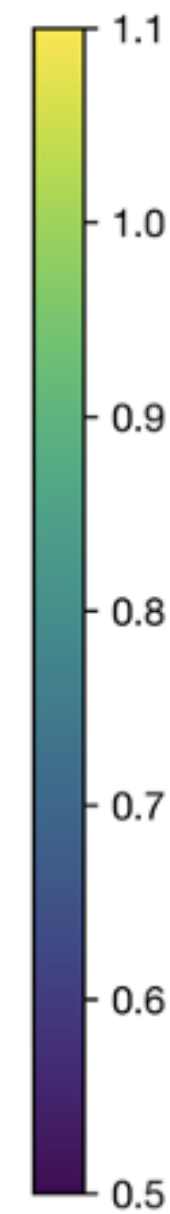
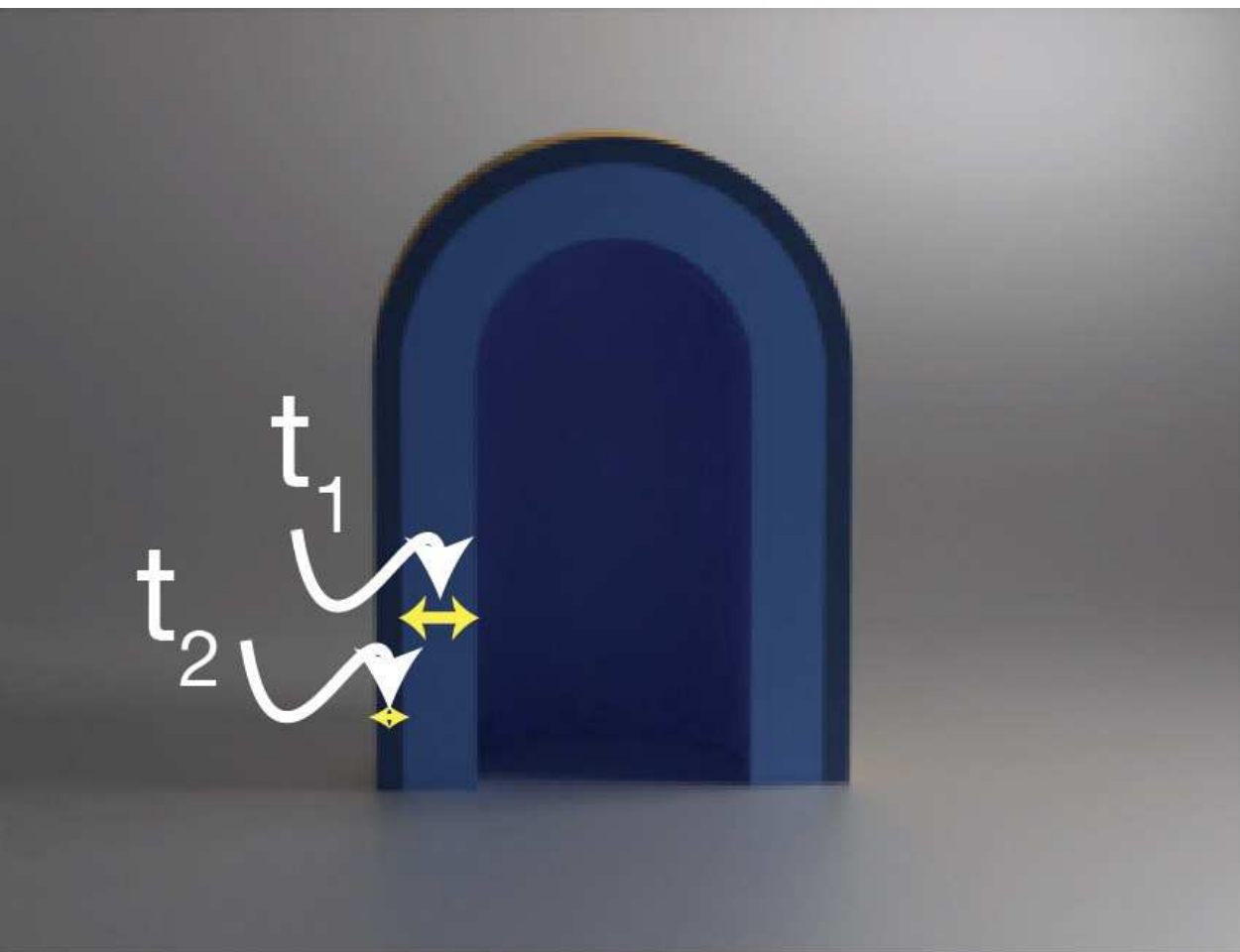
Rasterization



Path tracing

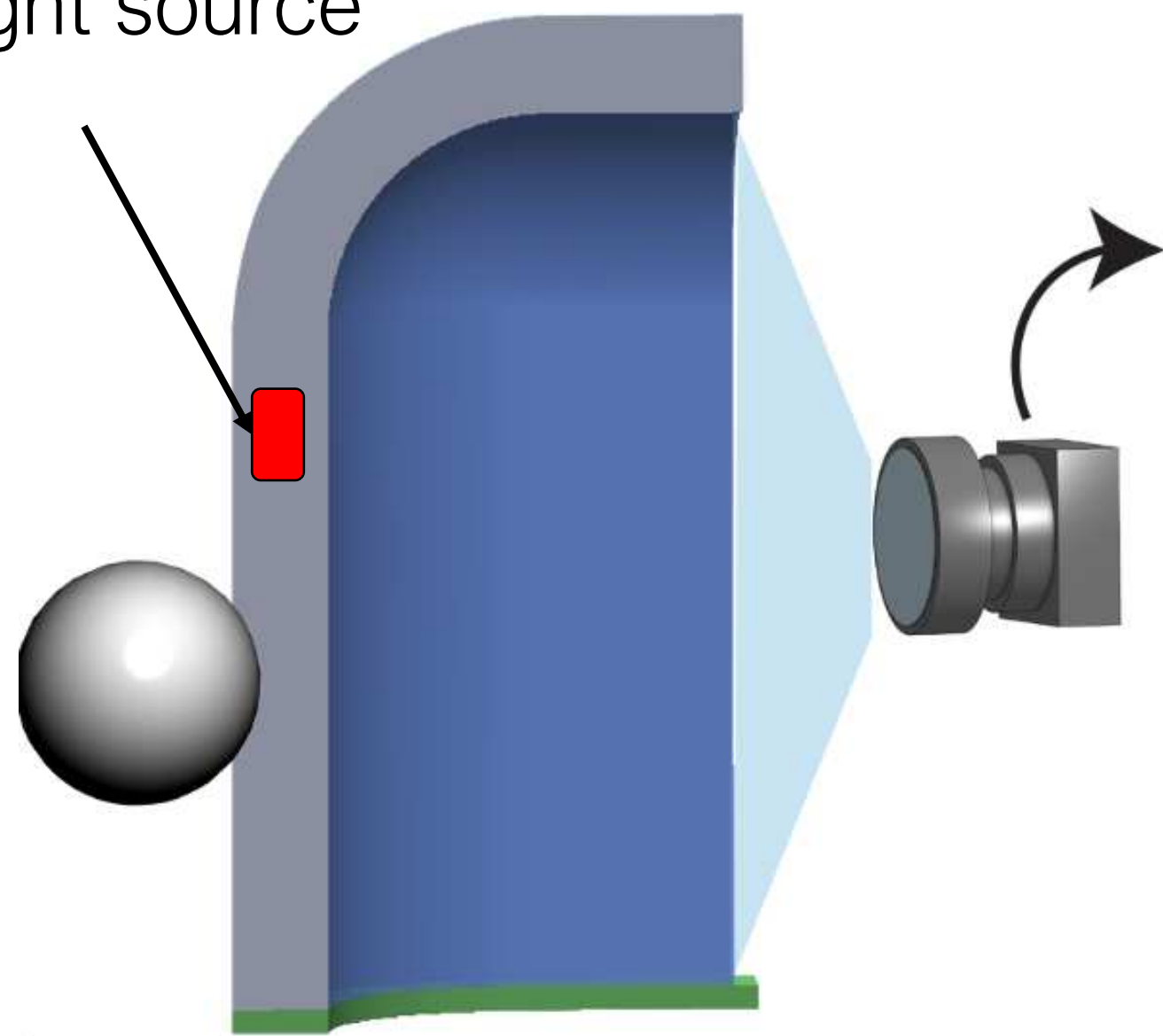
[Agarwal et al., 2023]

Curved sensor material design



Curved sensor illumination design

Light source

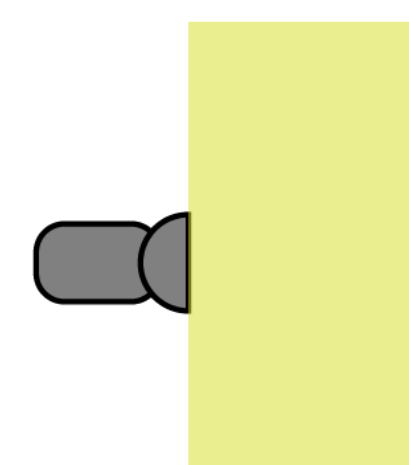
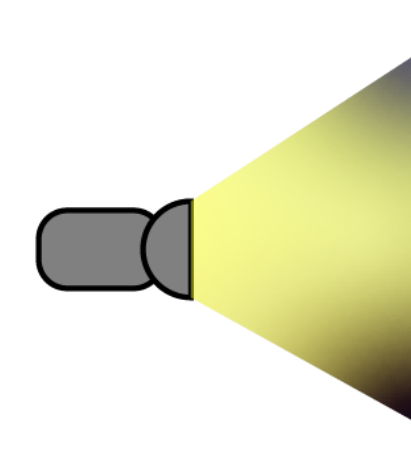
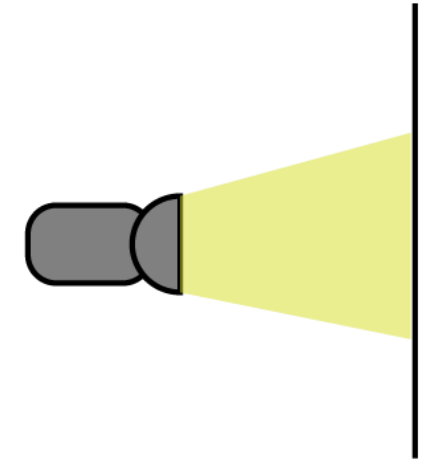


Spot Light

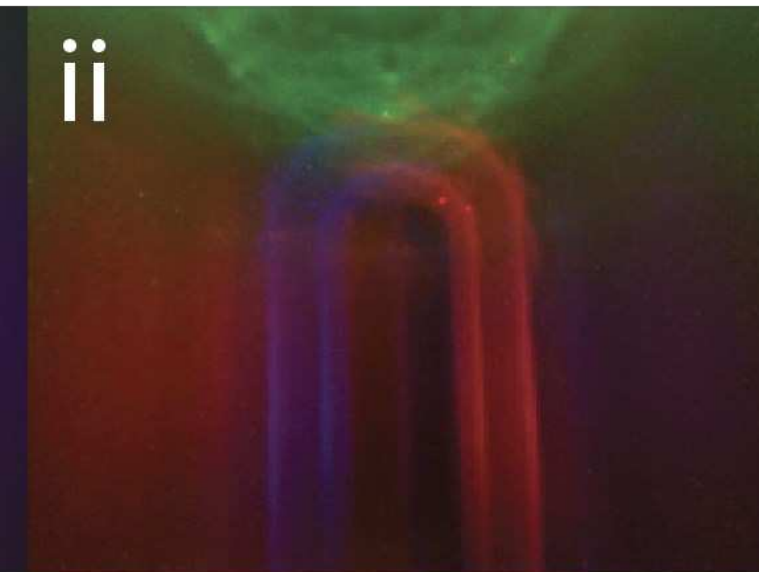
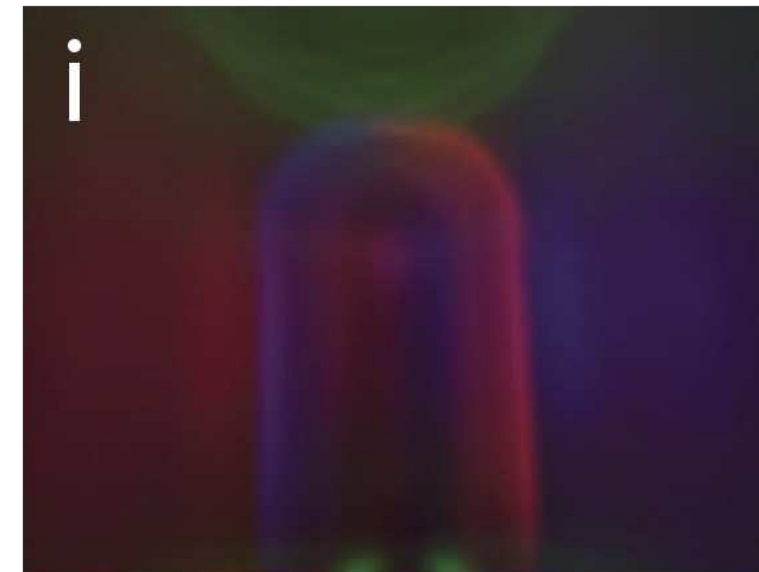
IES Light

Area Light

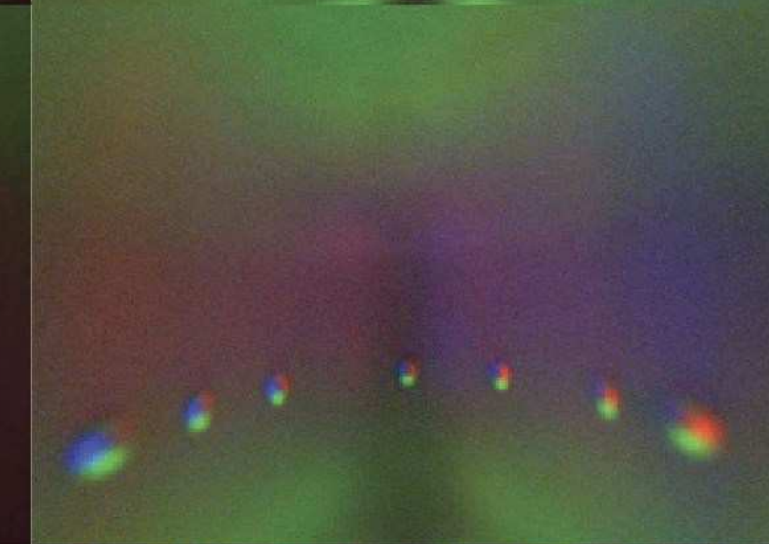
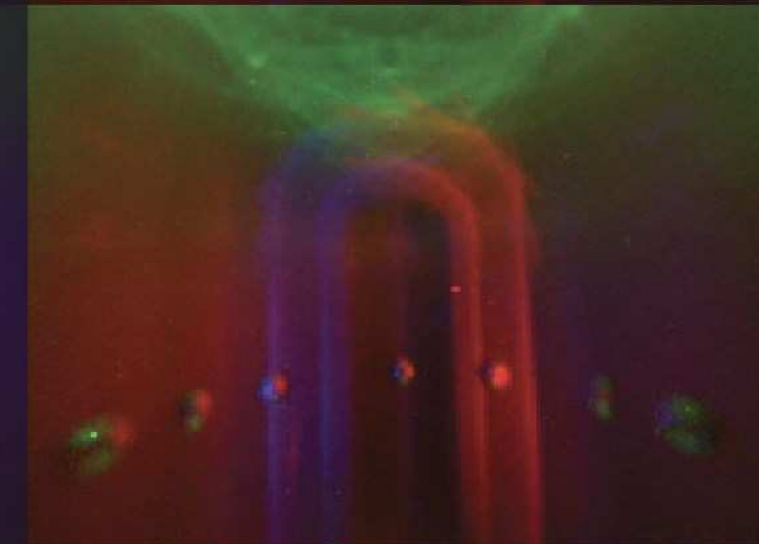
Light Type



Background Image



Indenter Image

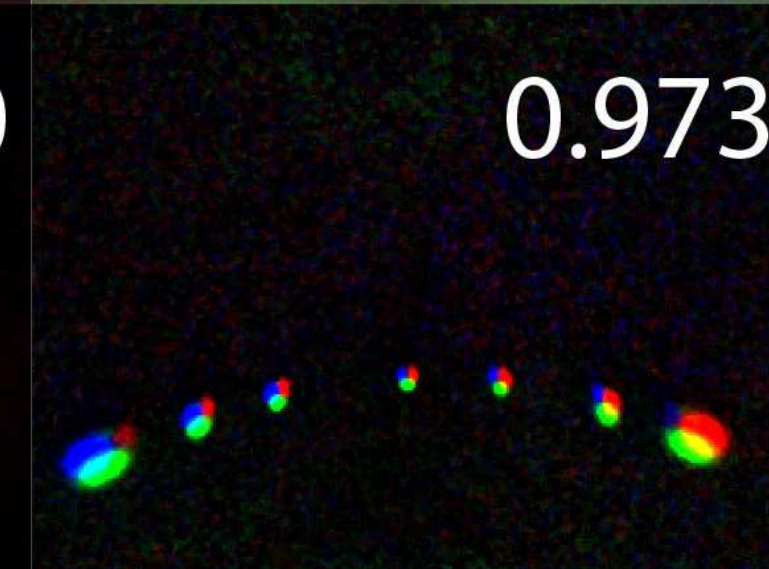
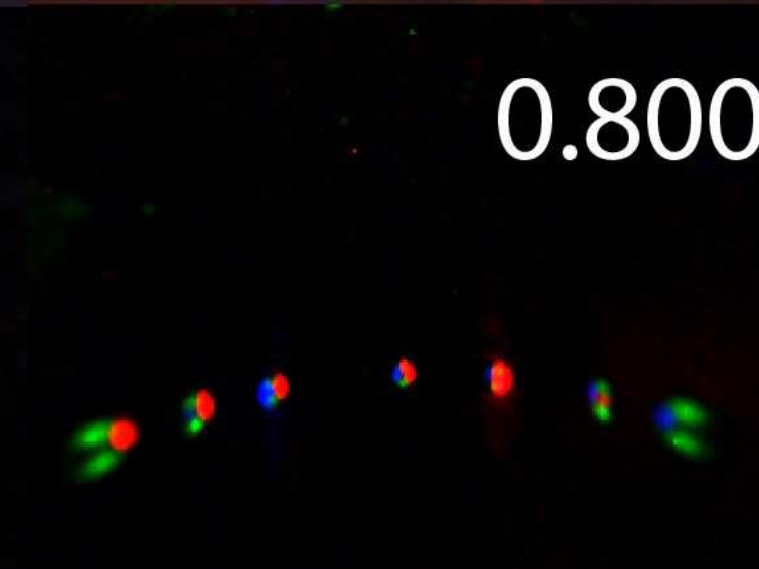
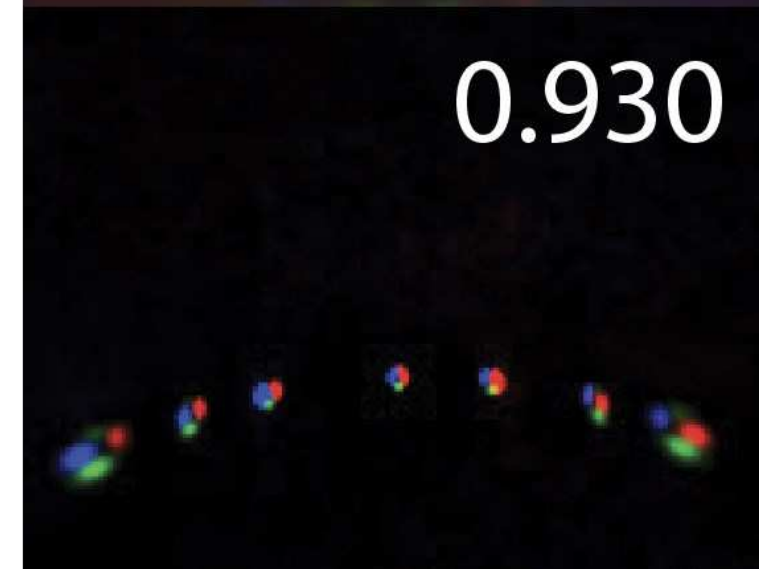


Color Signal

0.930

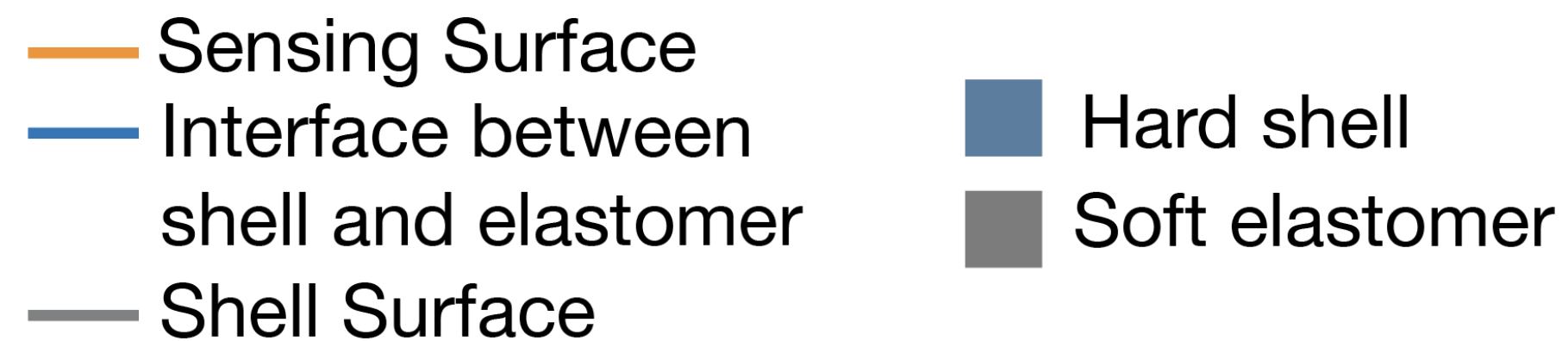
0.800

0.973



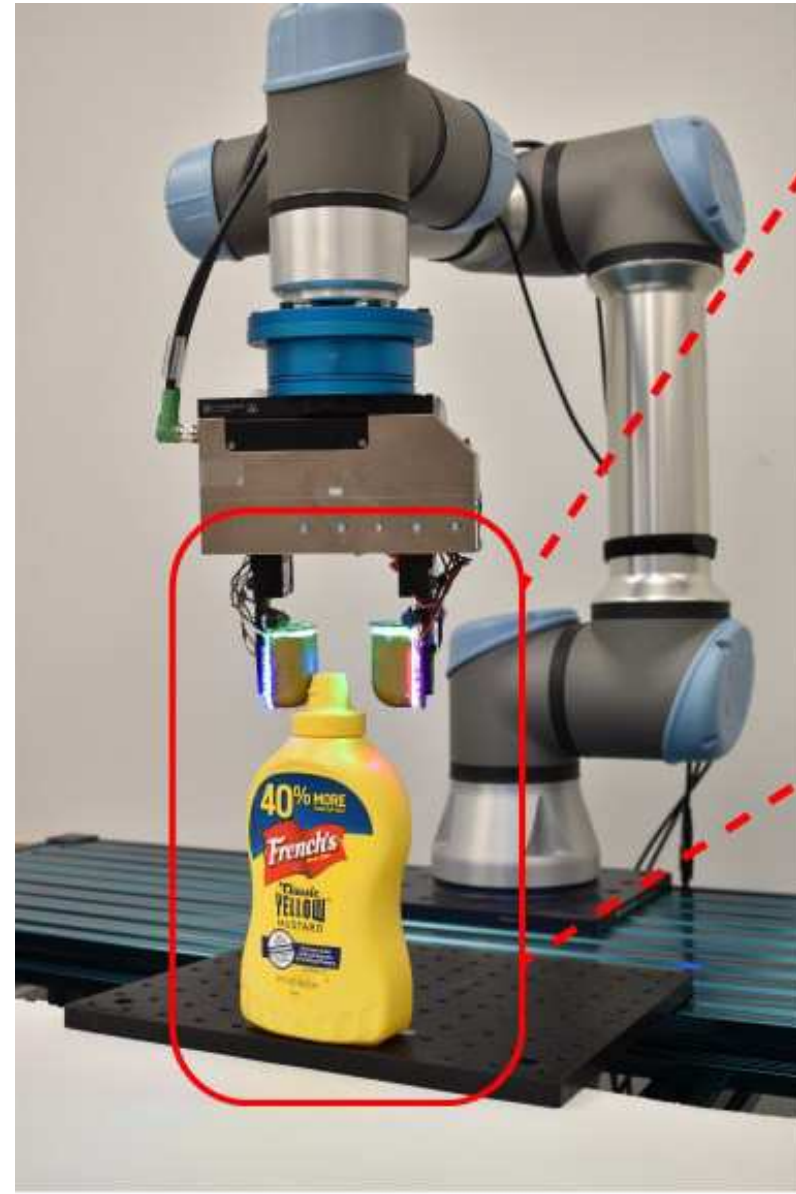
Curved sensor shape design

- Used gradient-free optimization, CMA-ES, for optimizing the sensor shape
- Optimized sensor design is 35% better than initial design
- Optimized sensor design outperforms human-expert design in 3D shape reconstruction



| | 2D Curve | CAD Model | Simulated Image |
|---------------------|----------|-----------|-----------------|
| Initial Design | | | |
| Human Expert Design | | | |
| Optimized Design | | | |

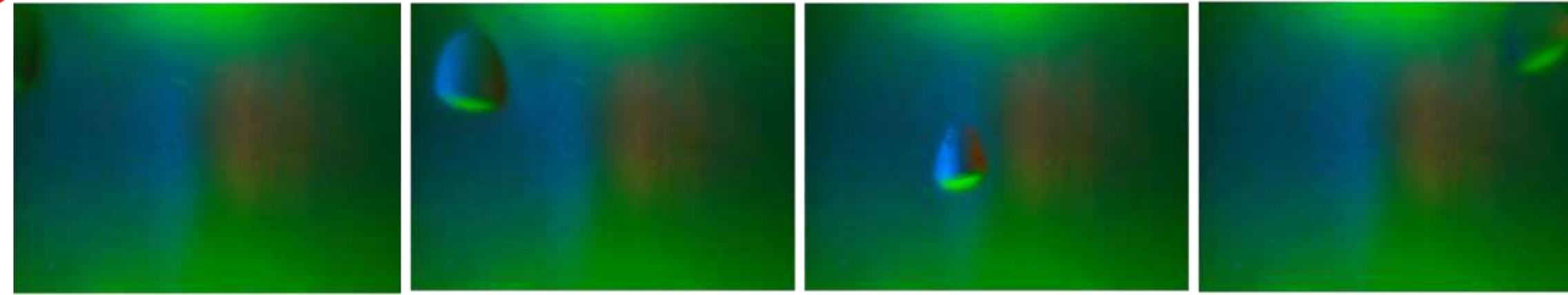
Results: Robotic grasping



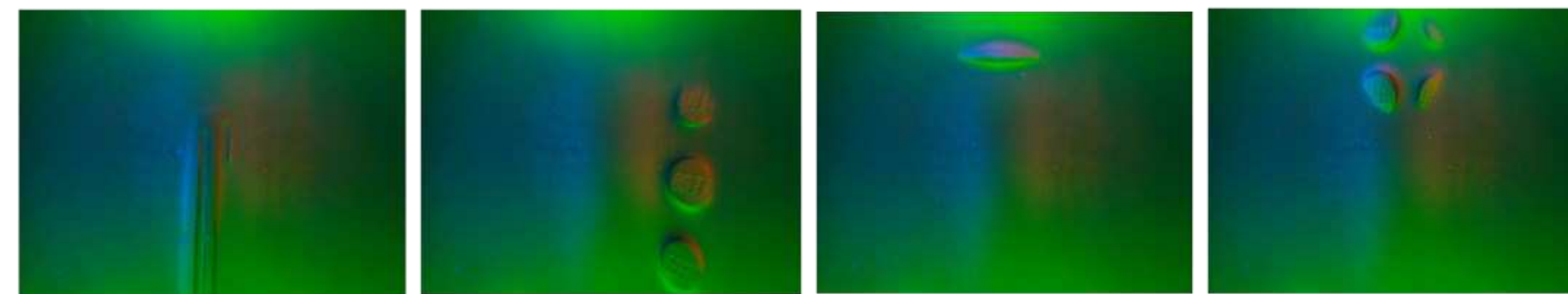
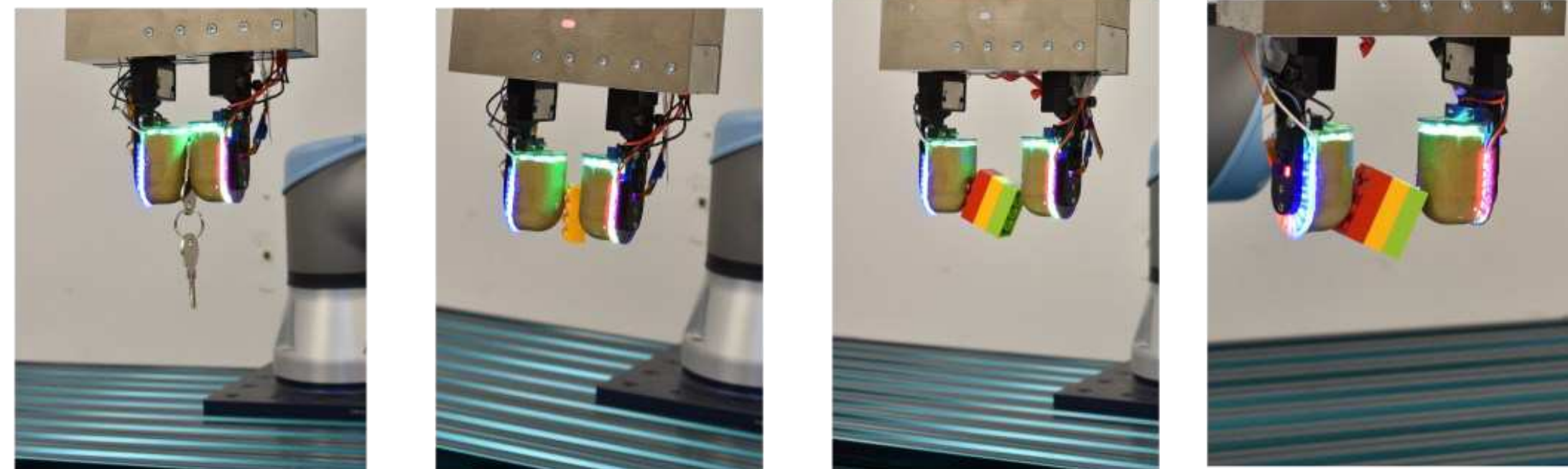
Robotic arm with optimized tactile sensor



Common Objects



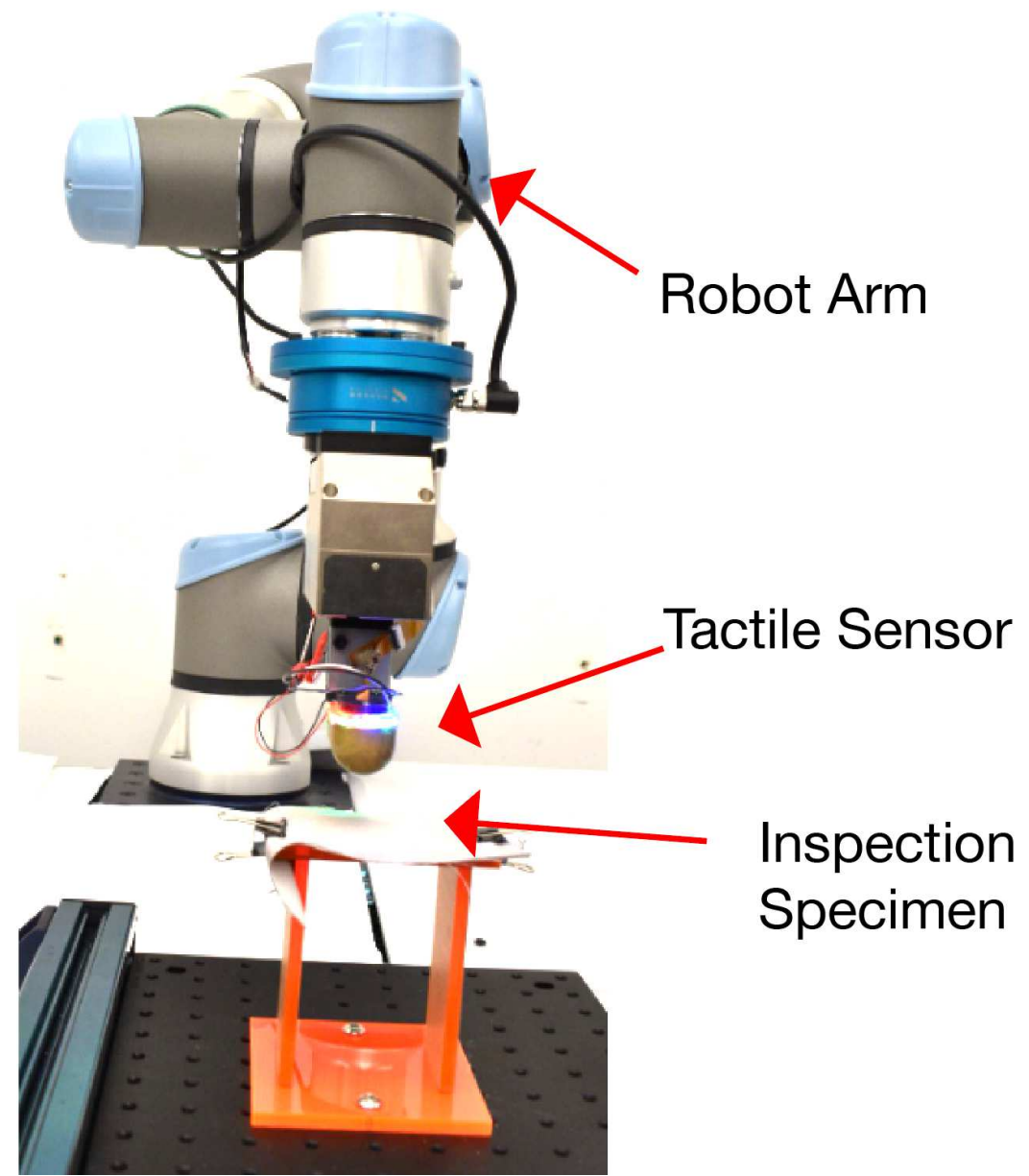
Tactile Images



Tactile Images

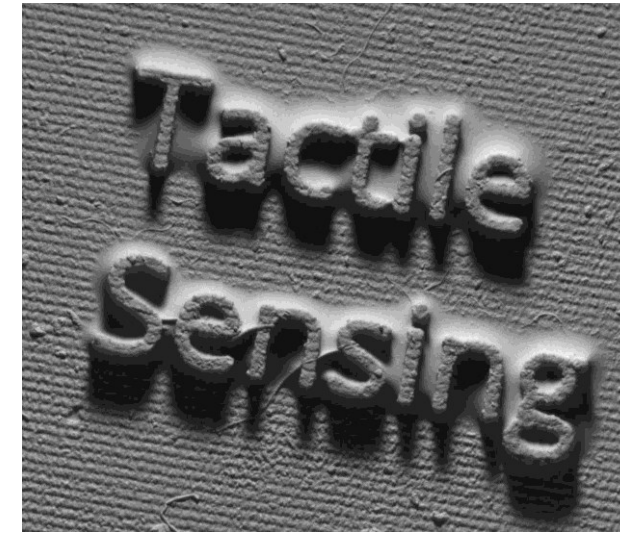
Results: surface inspection

Experiment 1: Detection accuracy vs text size



Experiment setup

Text description:
Tactile
Sensing

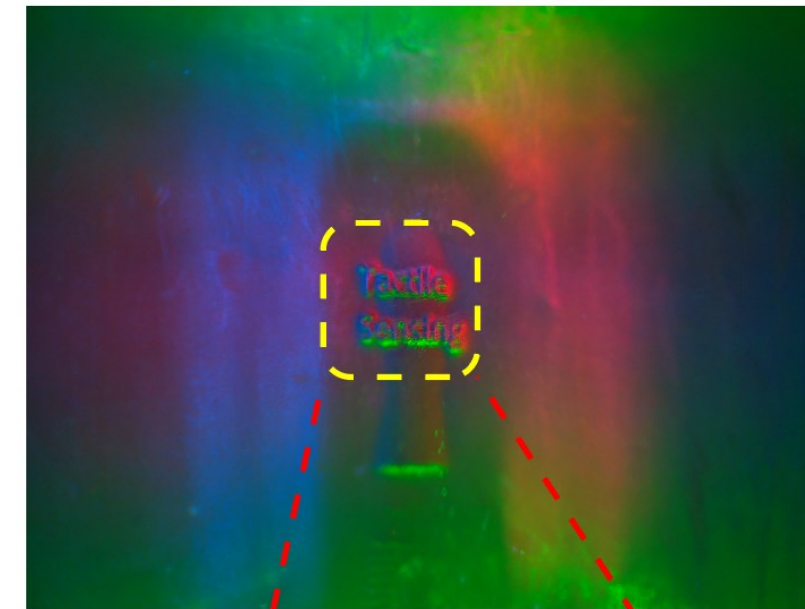


Text size = 1.5 mm

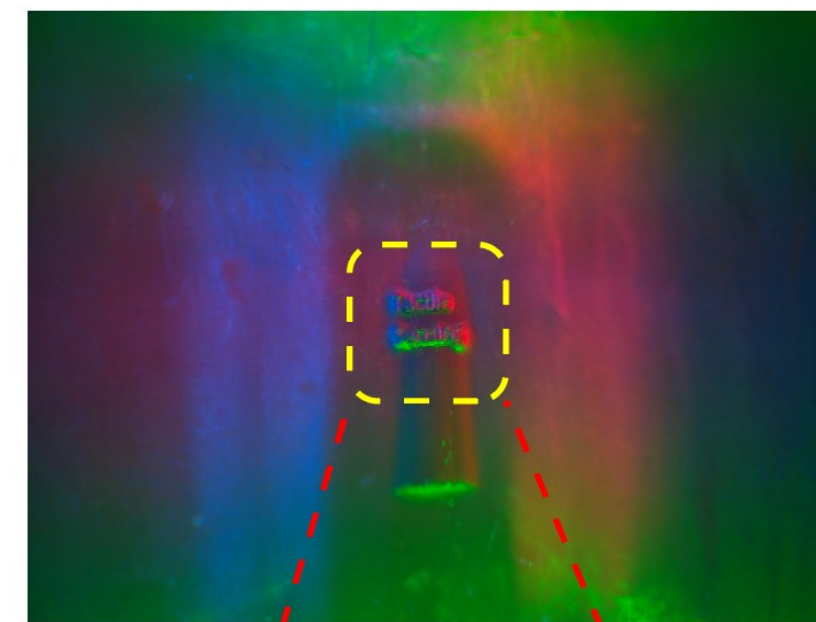


Text size = 1.0 mm

Human expert design

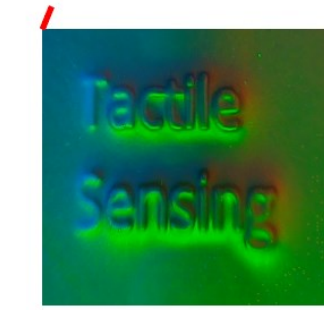
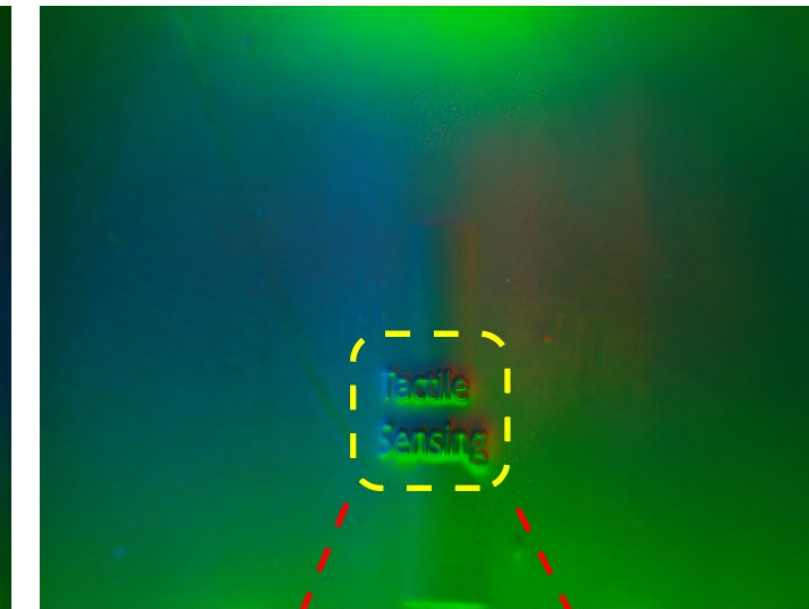


“Tatle Sending”

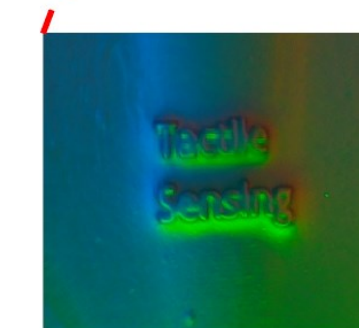
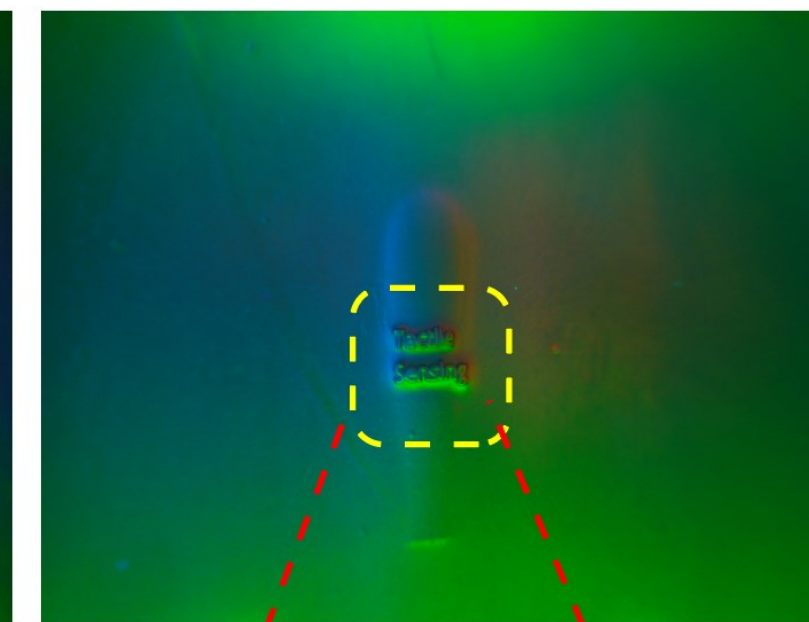


“_”

Optimized design



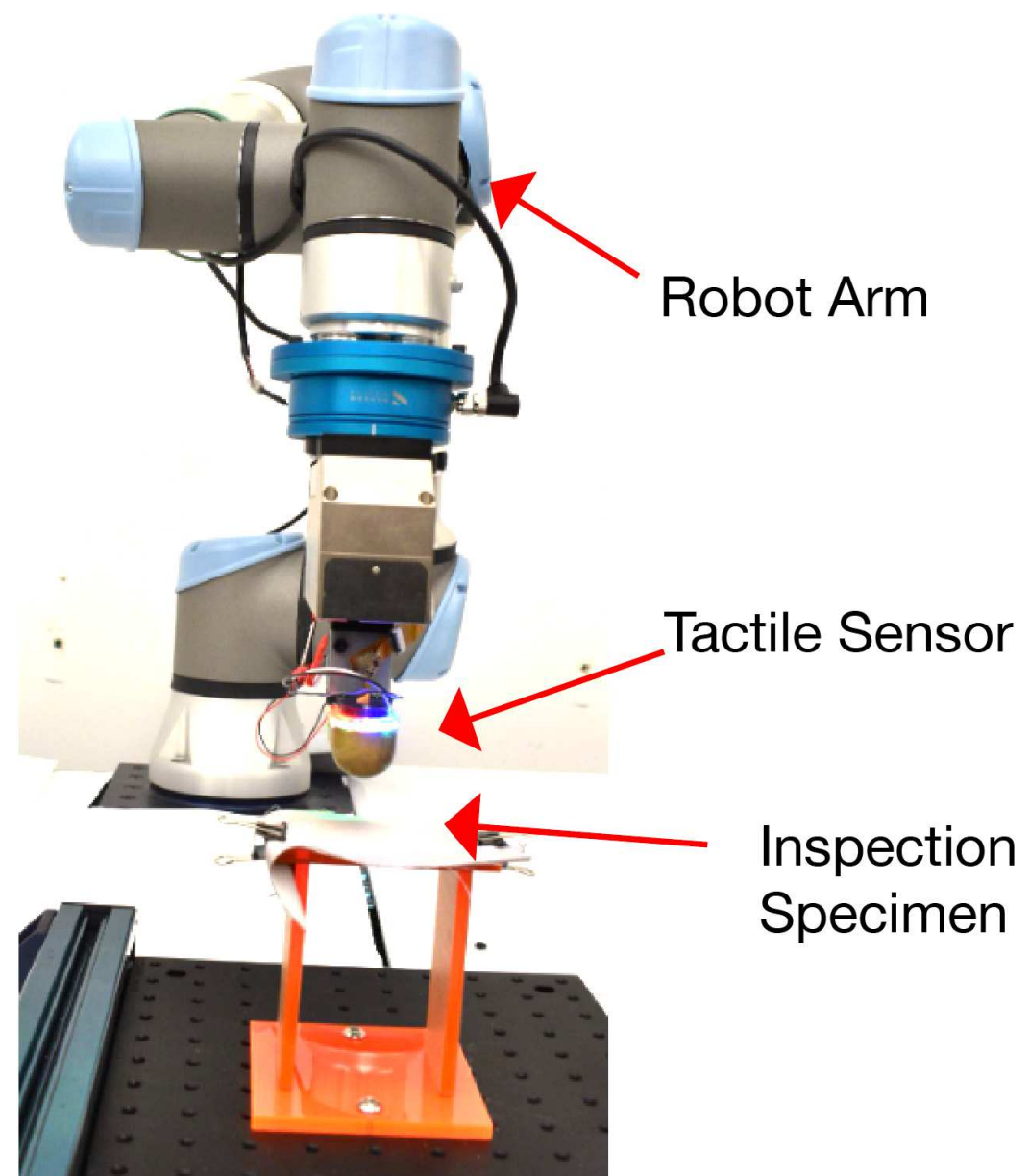
“Tacolle Sending”



“Tacolle Sensing”

[Agarwal et al., 2023]

Results: surface inspection



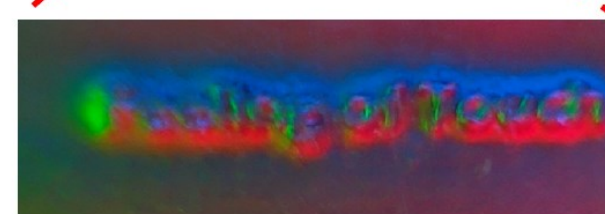
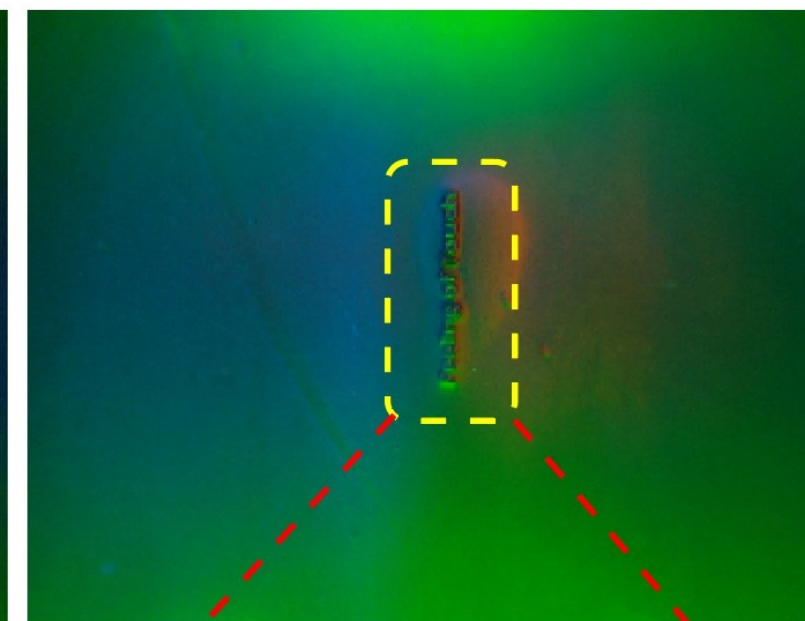
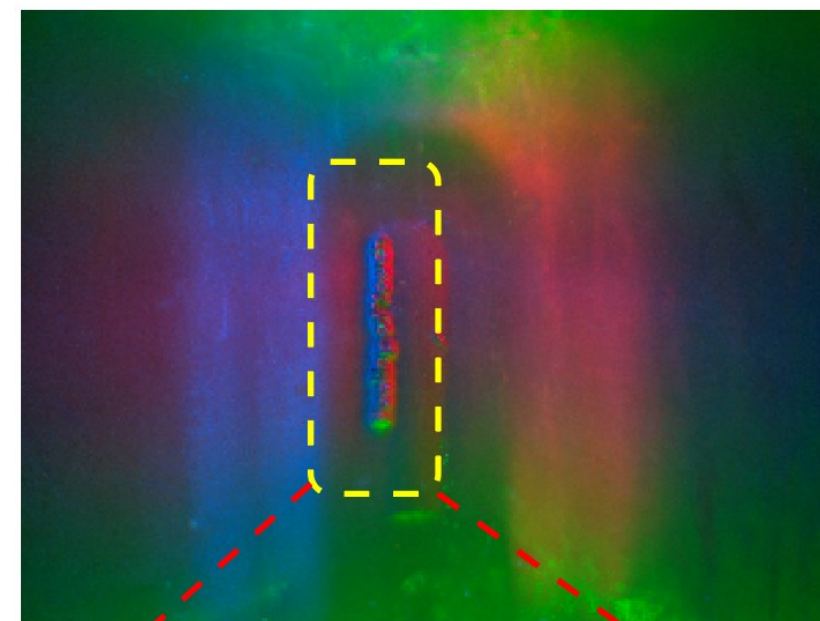
Experiment setup

Text description: Feeling of Touch

Experiment 2: Detection accuracy vs contact location

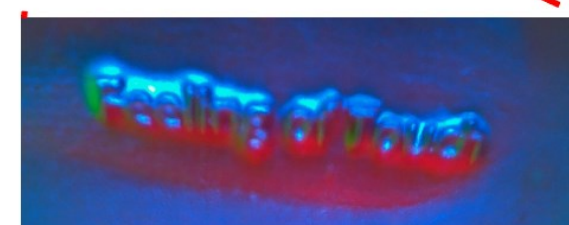
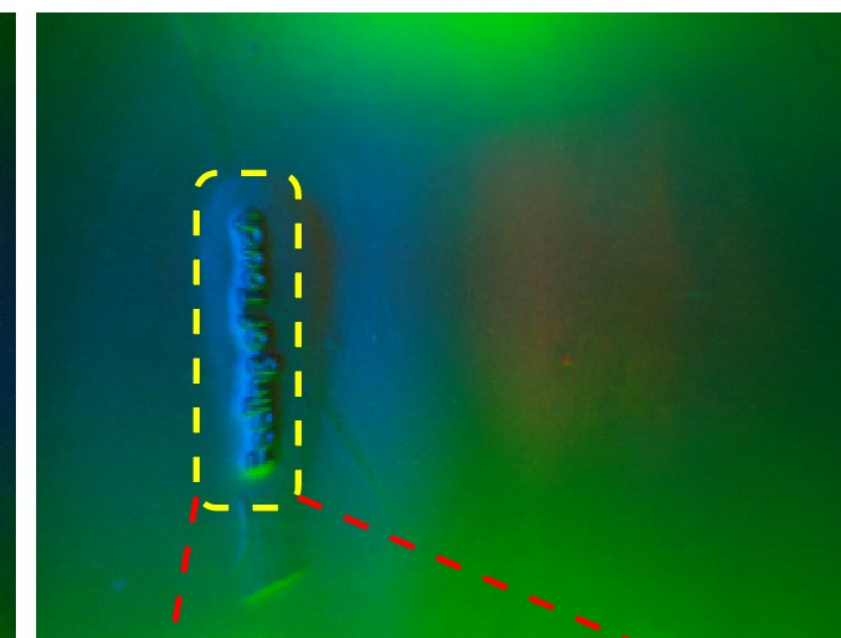
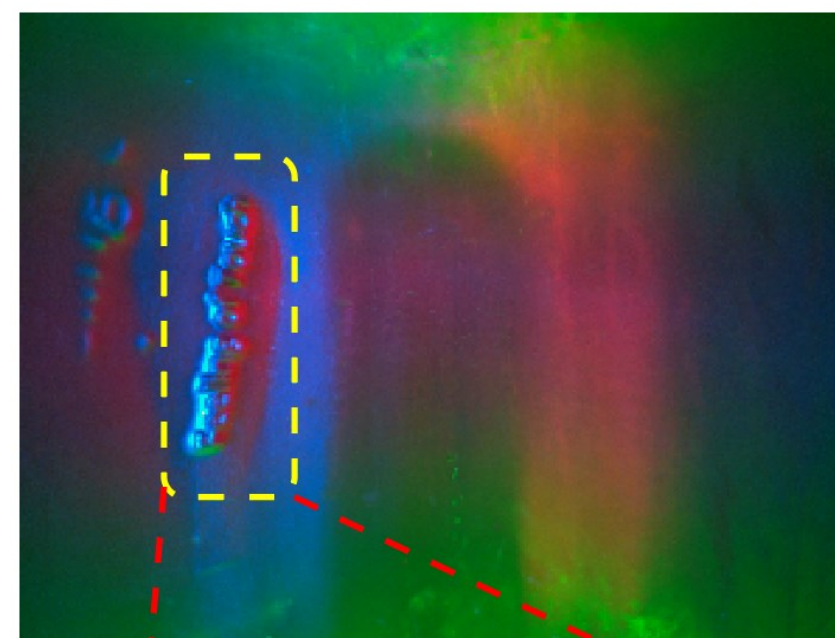
Human expert design

Optimized design



“_”

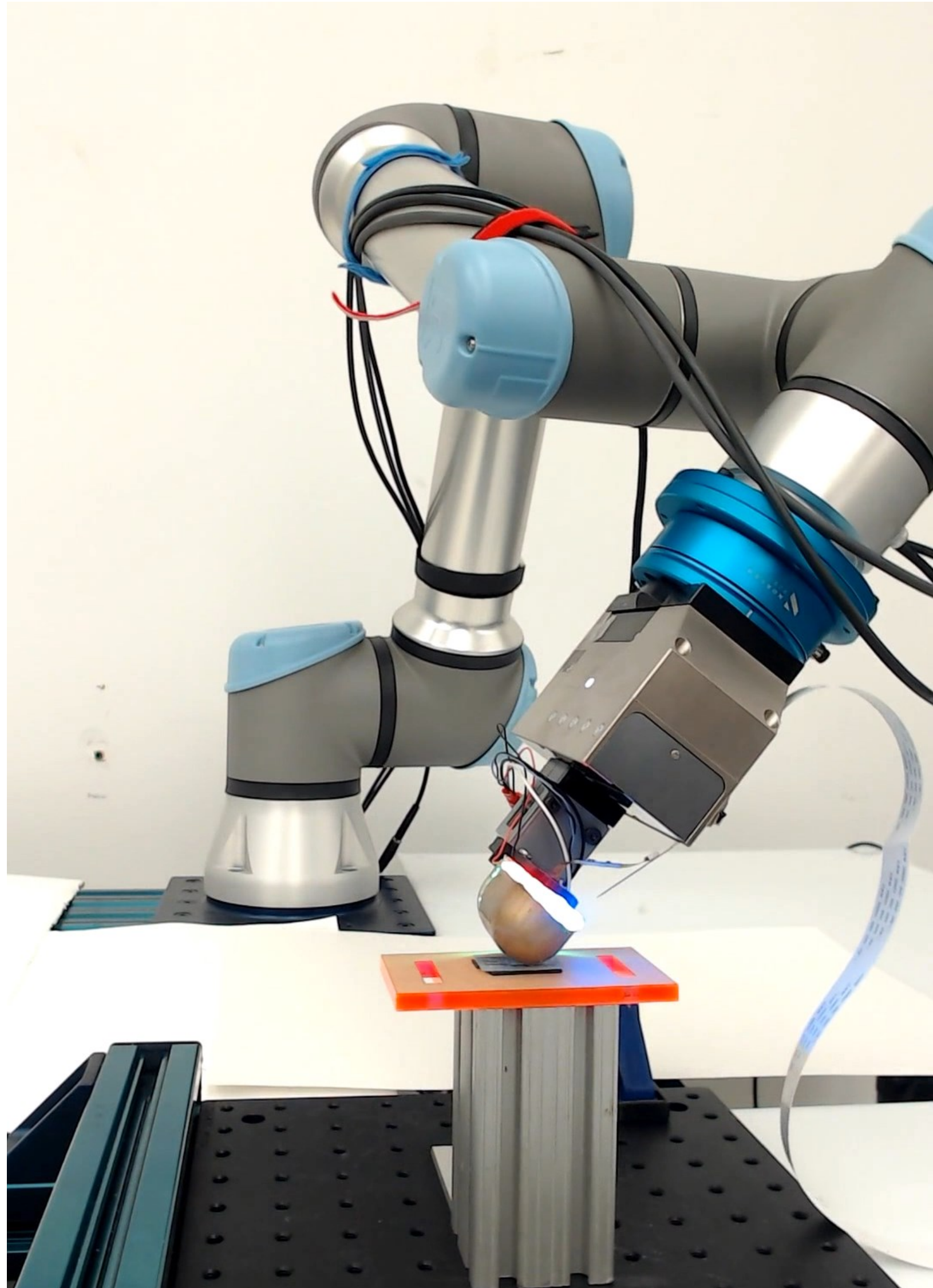
“Feeling of Touch”



“_”

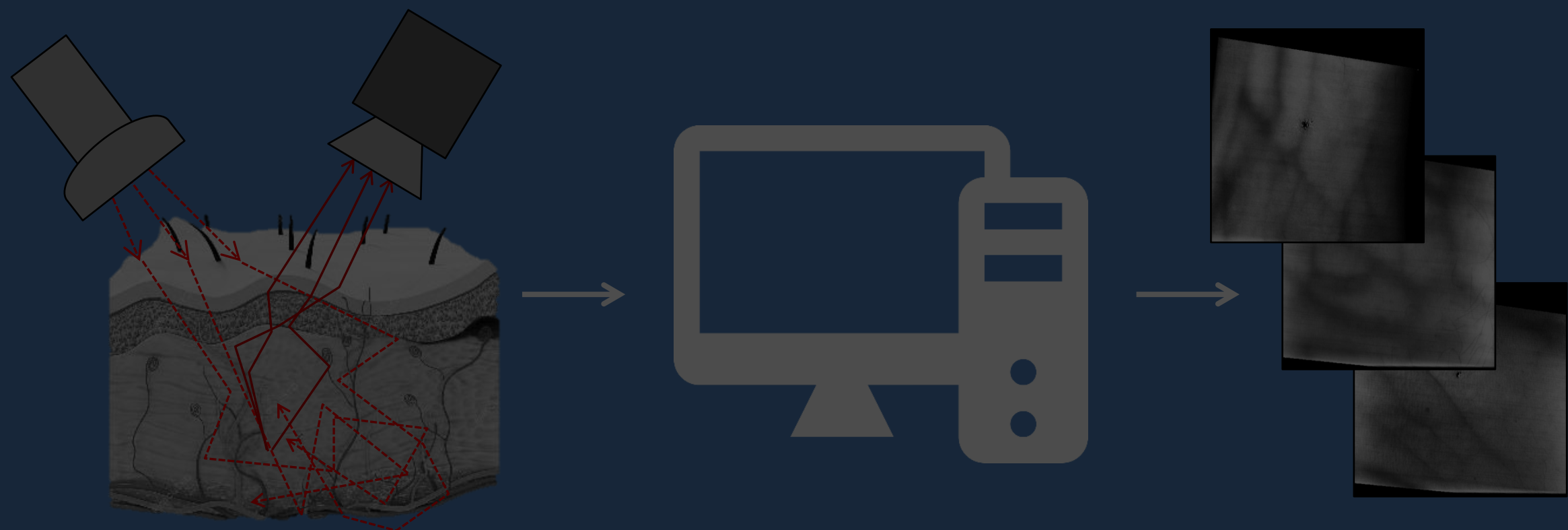
“Peeling of Touch”

Results: surface inspection



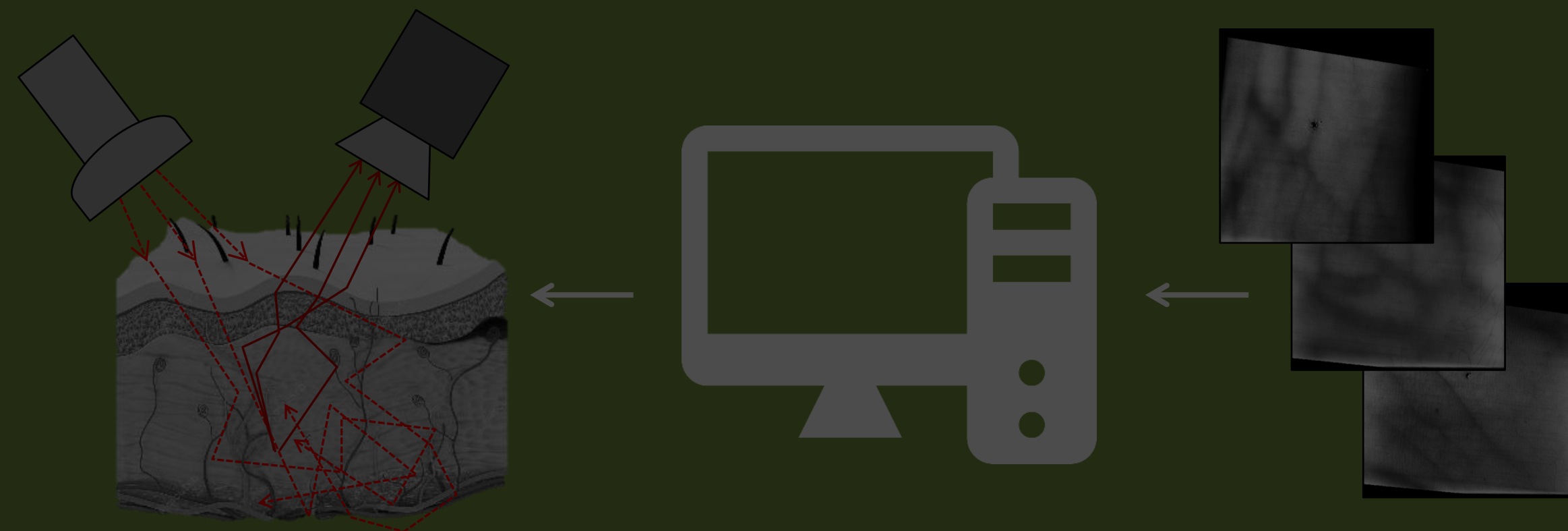
Physics-based rendering and its applications to computational imaging

forward rendering

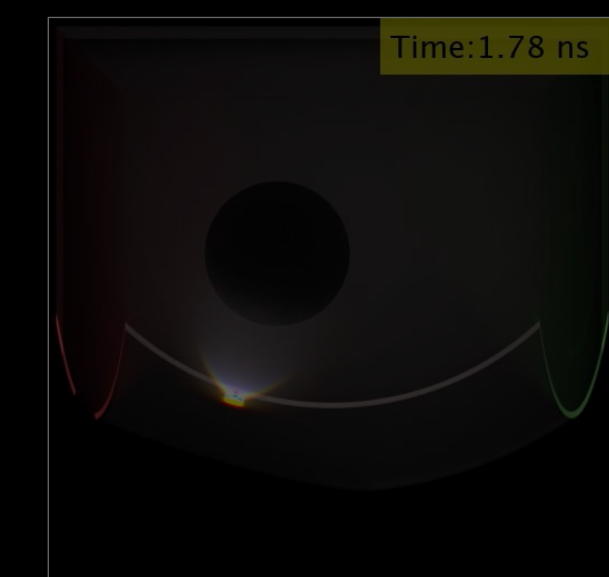


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

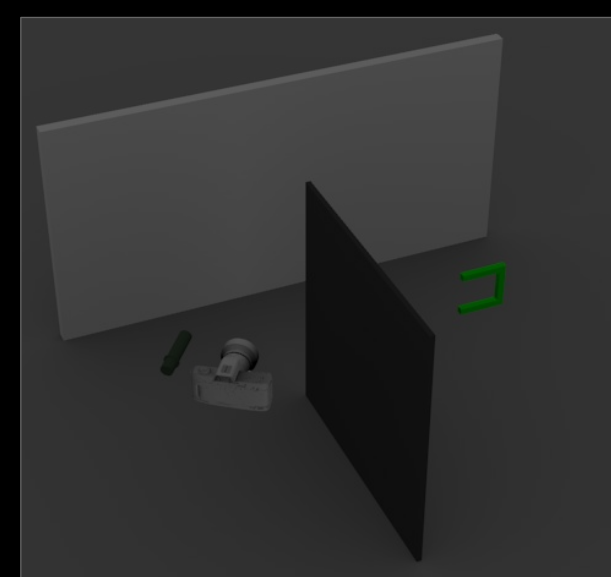
inverse rendering



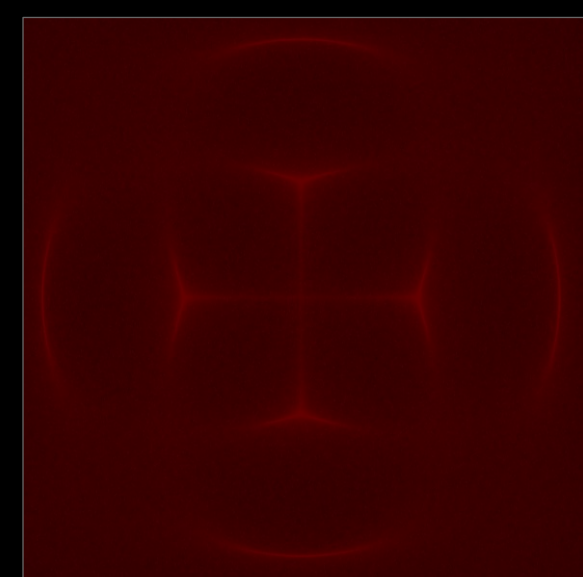
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



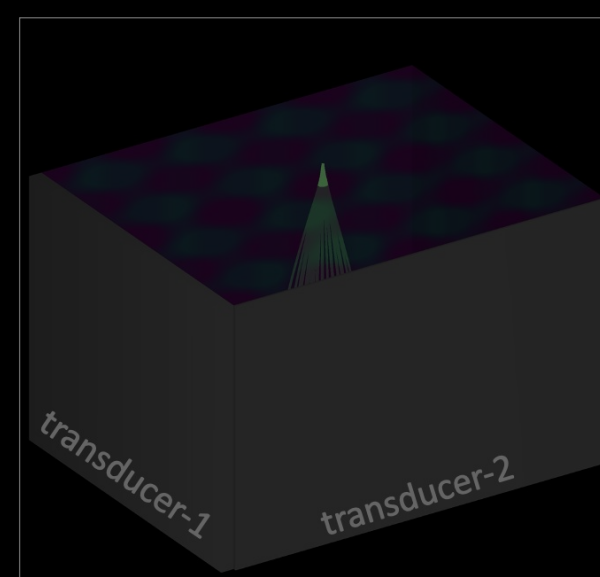
time-of-flight imaging



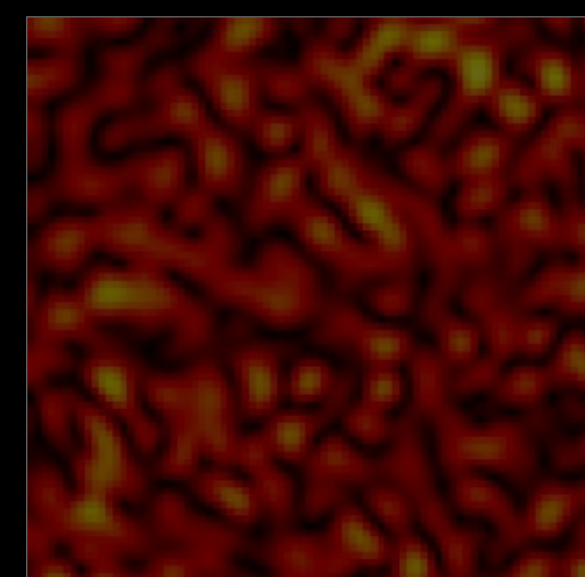
non-line-of-sight imaging



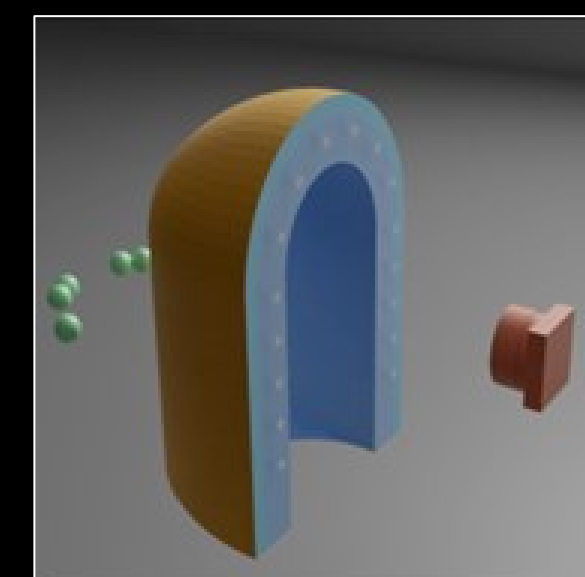
acousto-optic lensing



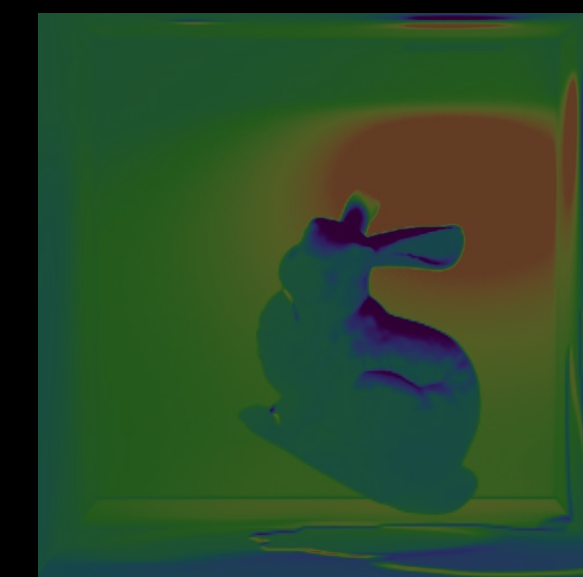
ultrafast light scanning



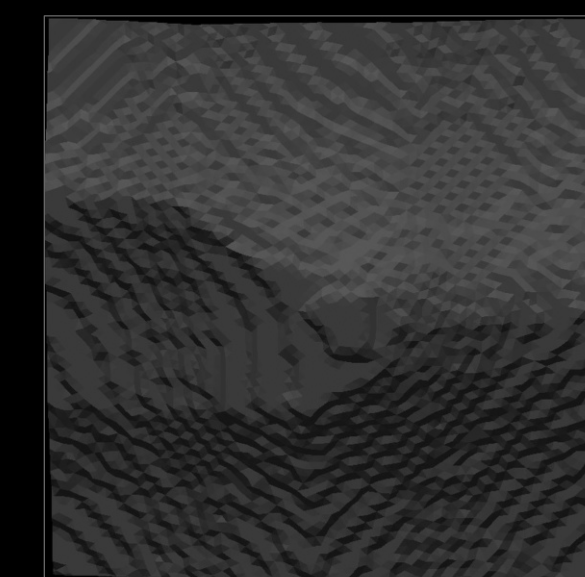
speckle imaging



tactile sensor design



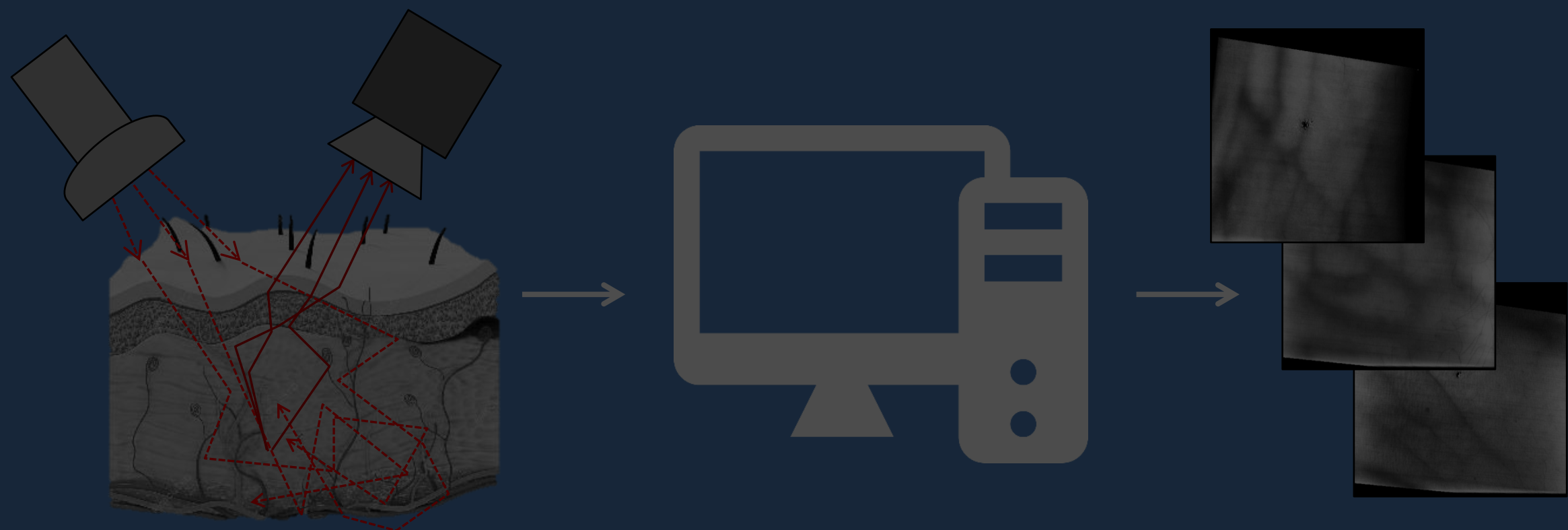
differentiable renderer



inverse problems

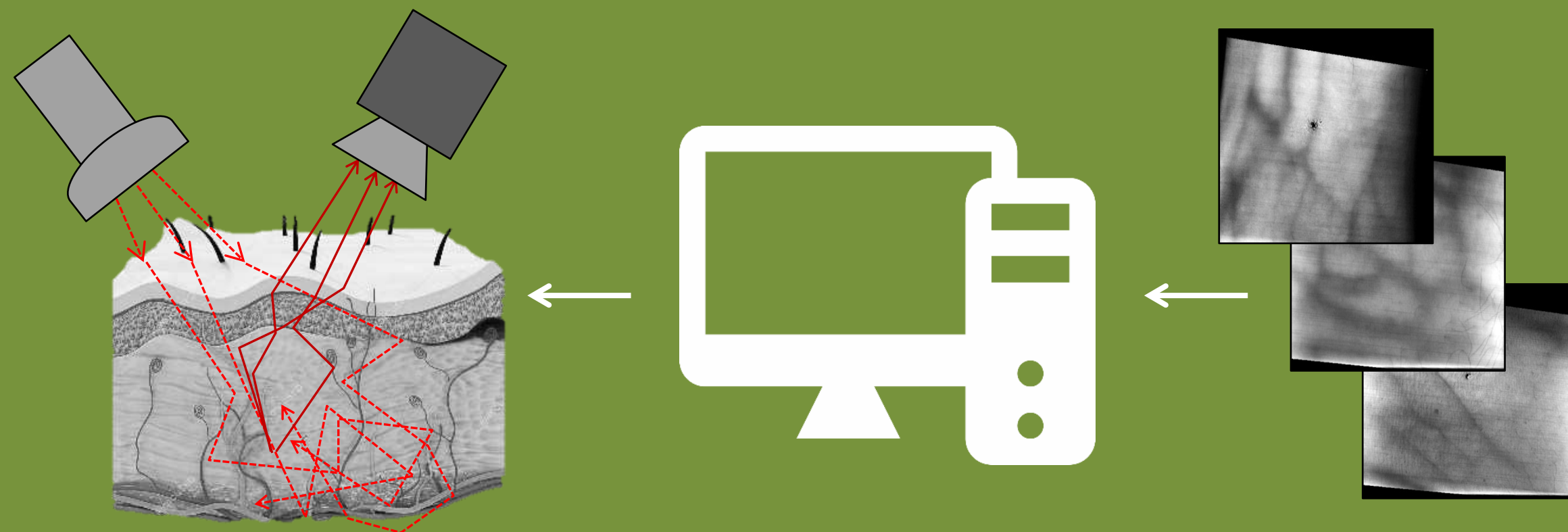
Physics-based rendering and its applications to computational imaging

forward rendering

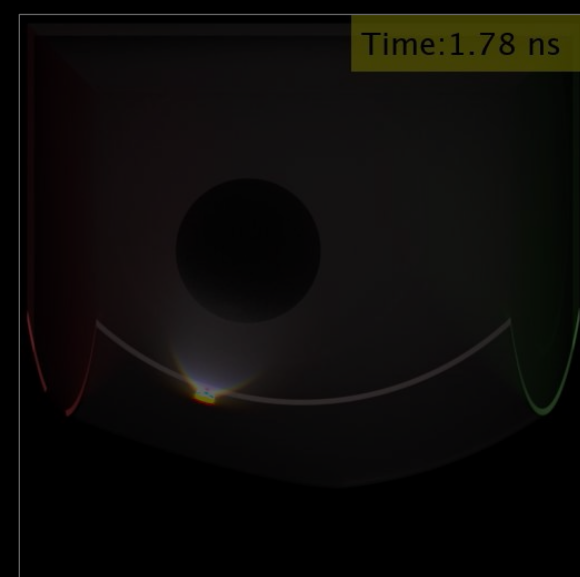


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

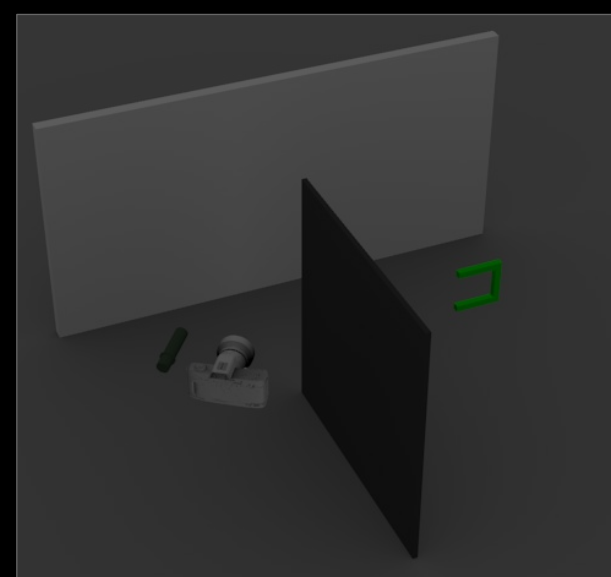
inverse rendering



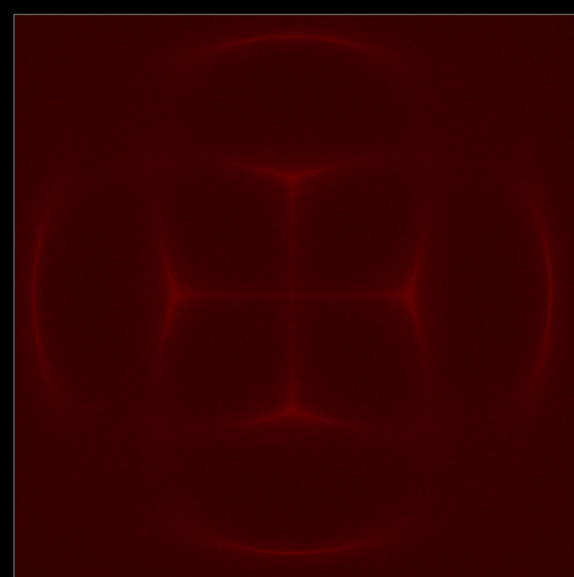
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



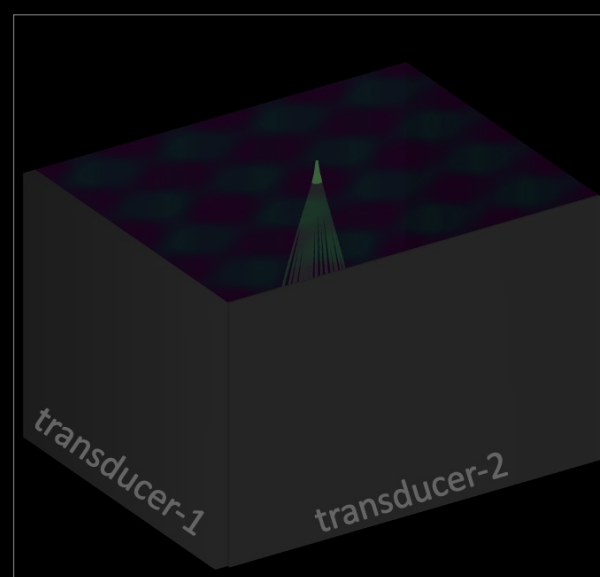
time-of-flight
imaging



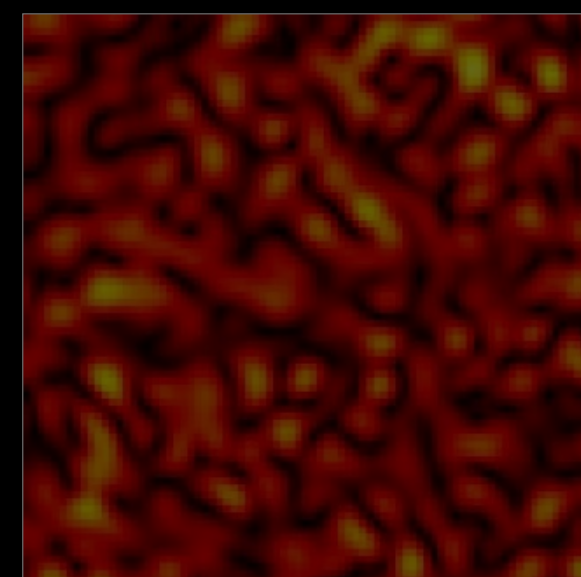
non-line-of-sight
imaging



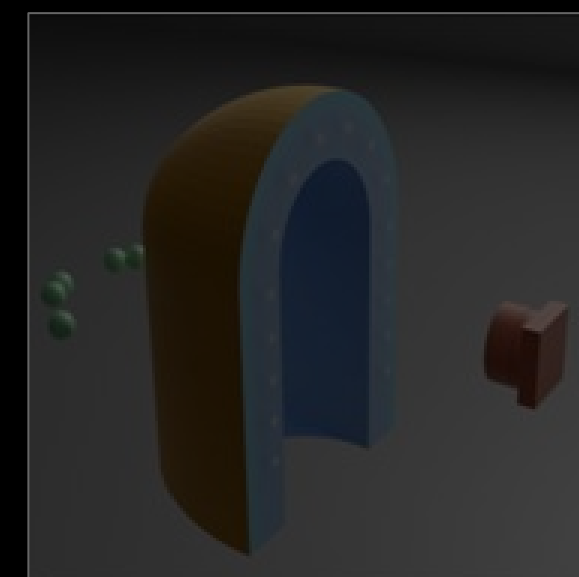
acousto-optic
lensing



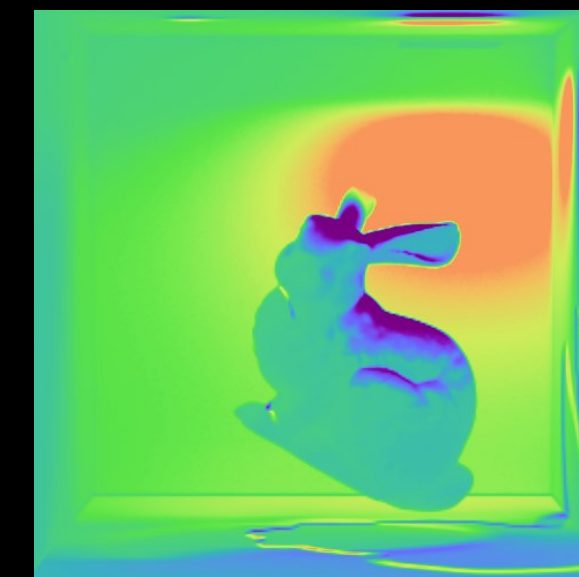
ultrafast light
scanning



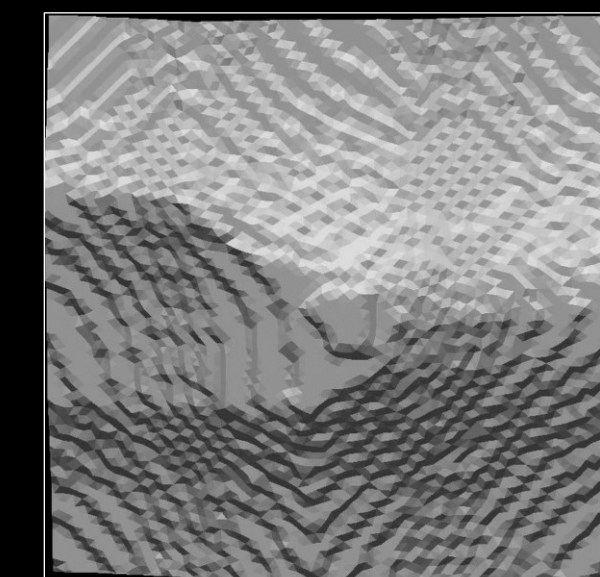
speckle
imaging



tactile sensor
design

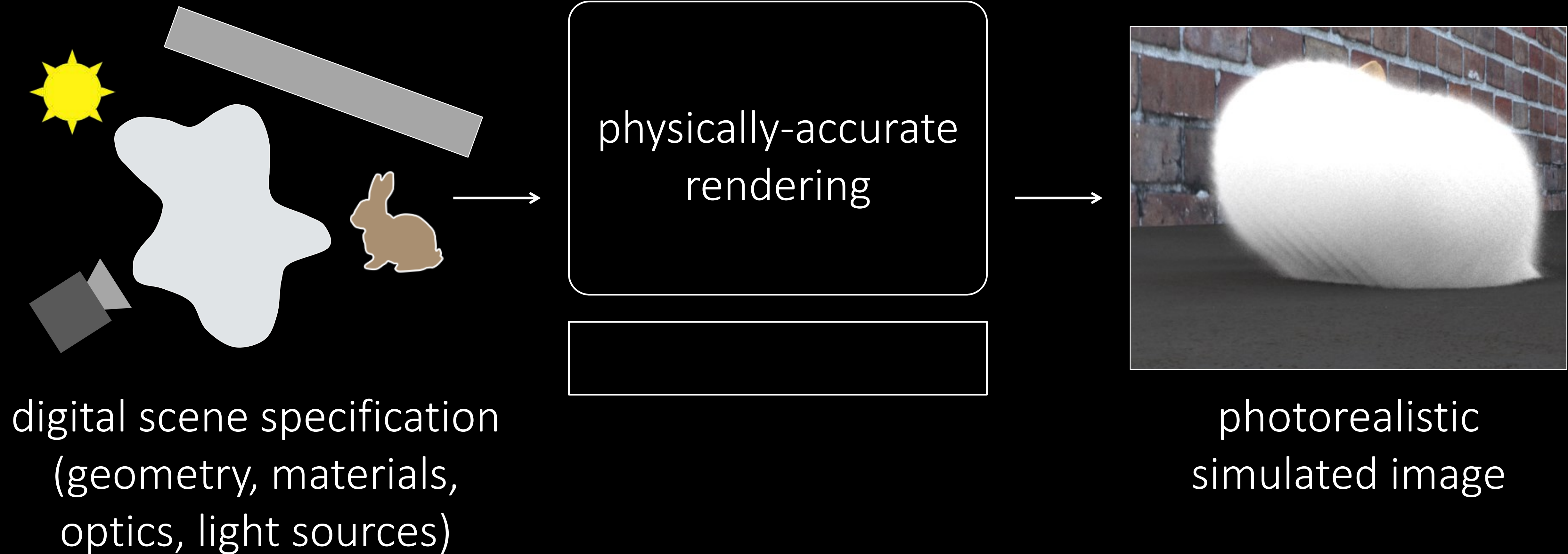


differentiable
renderer

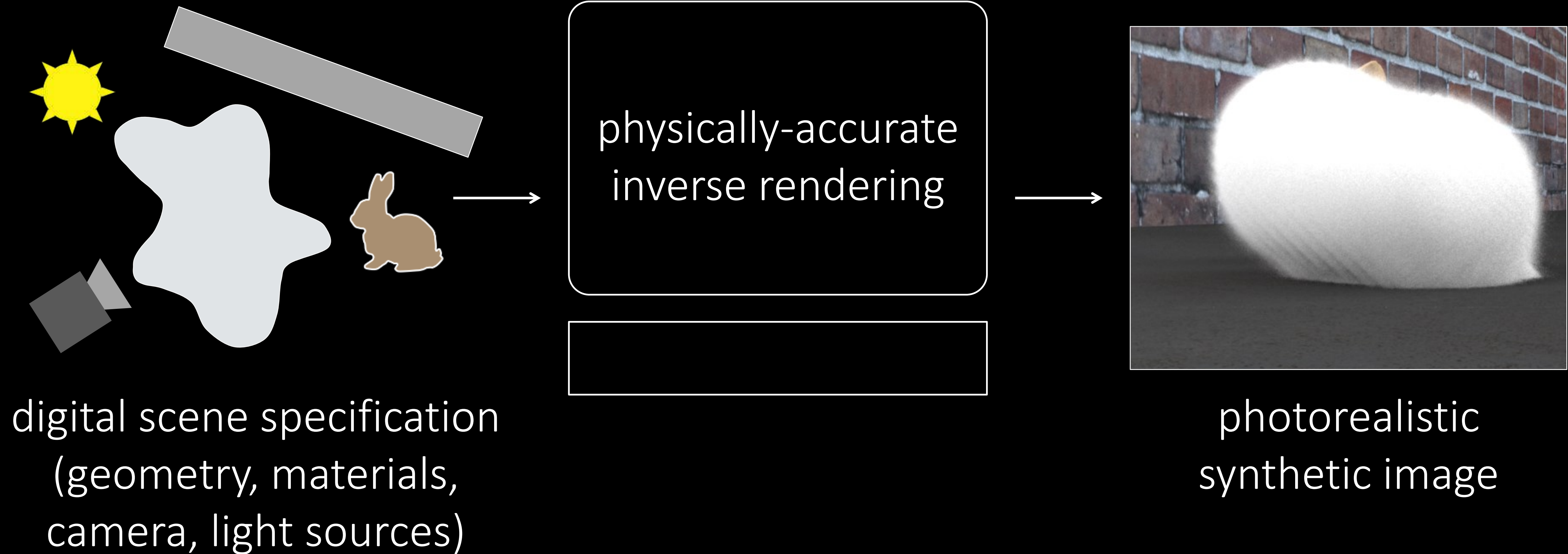


inverse
problems

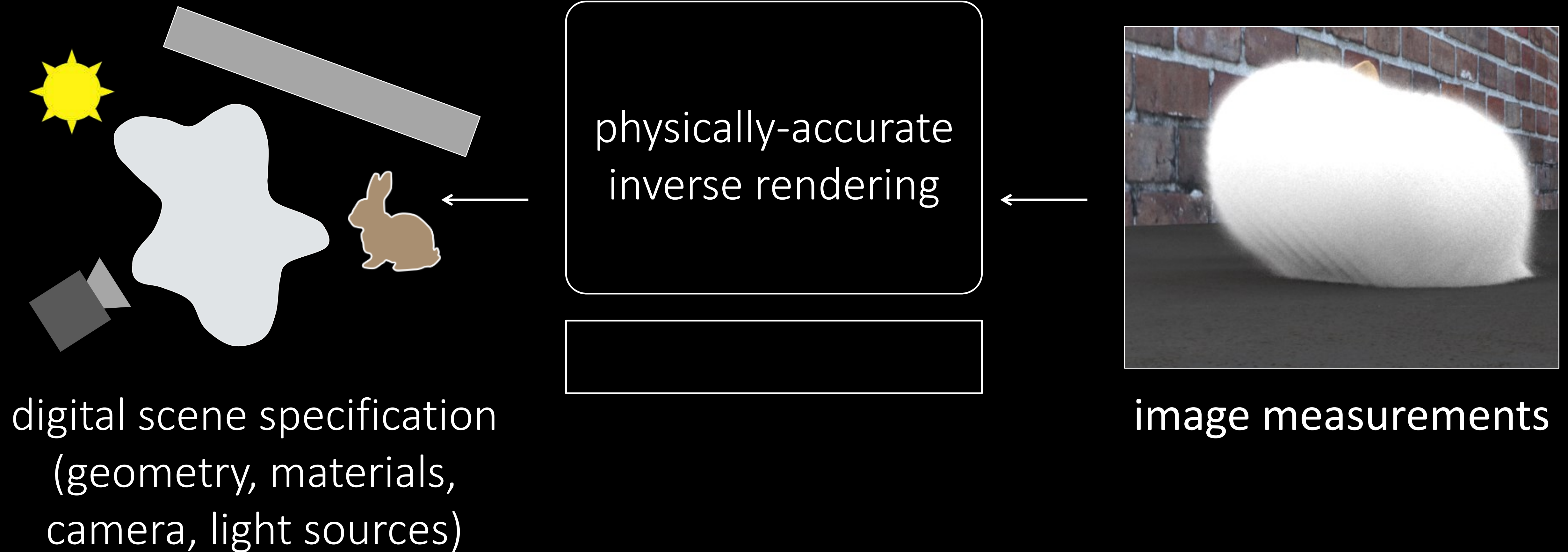
Forward rendering



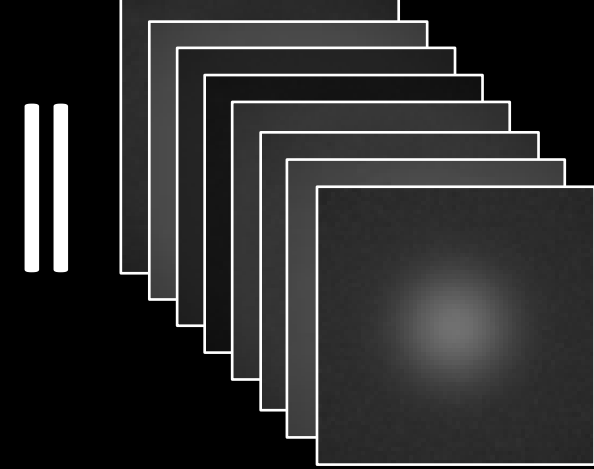
Inverse rendering



Inverse rendering

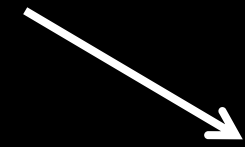


Analysis by synthesis (a.k.a. inverse rendering)

$$\min_{\substack{\text{unknowns } m \\ \text{(tissue properties)}}} \left\| \begin{array}{c} \text{stack of images} \\ \text{image}(m) \end{array} \right\|^2$$


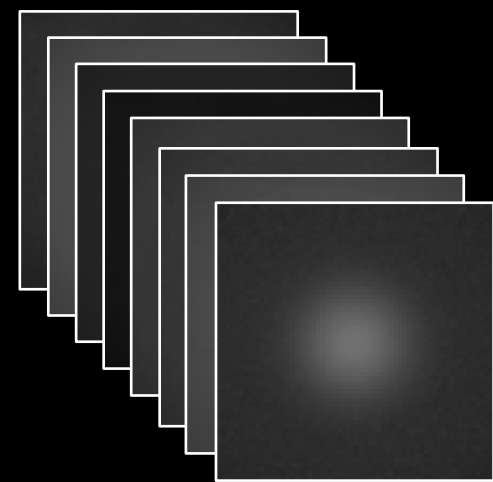
Analysis by synthesis (a.k.a. inverse rendering)

captured
measurements

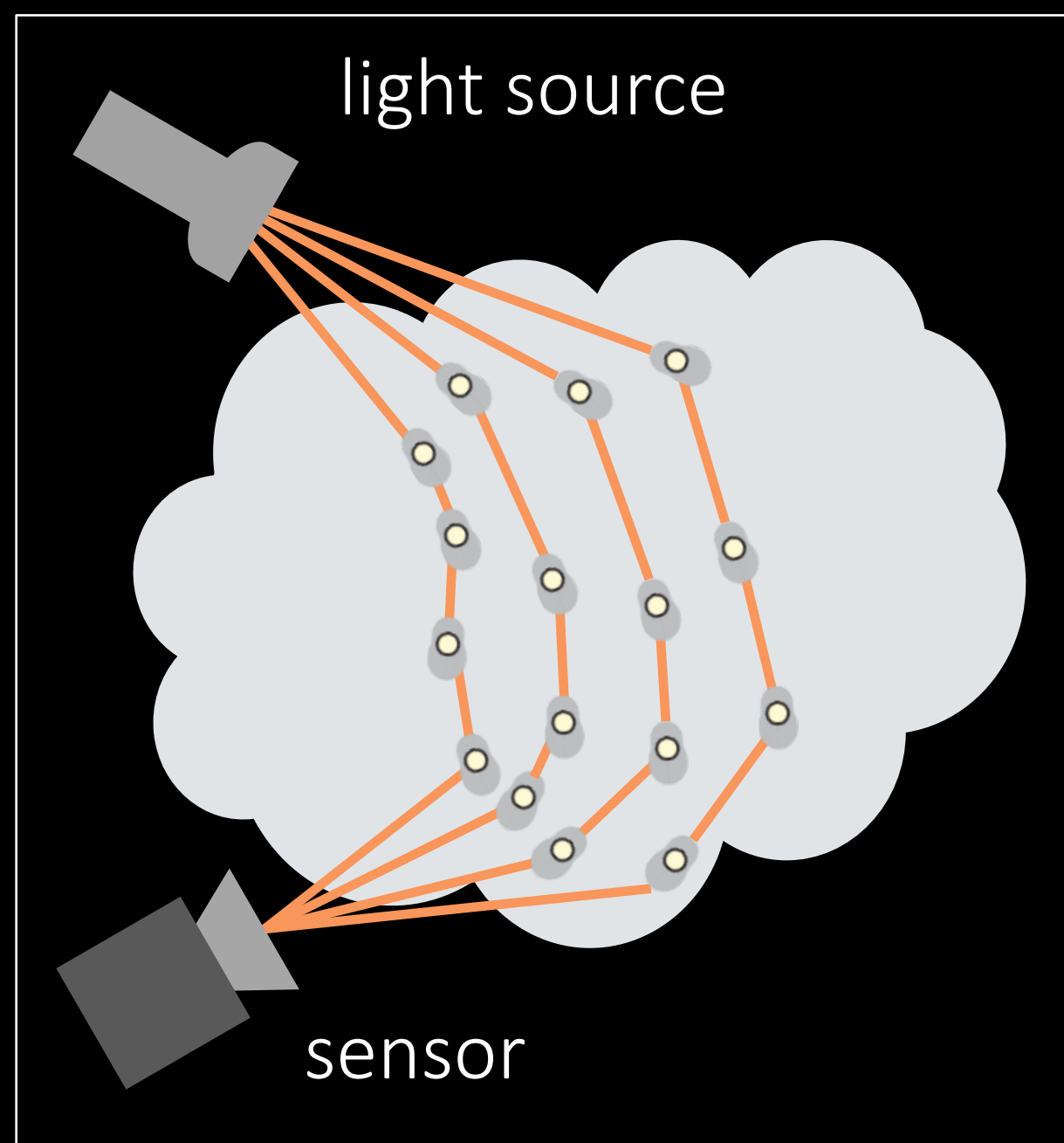
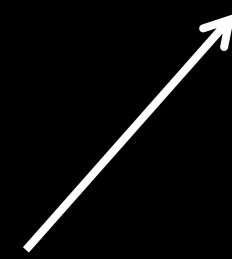


min
unknowns m
(tissue properties)

$\|$



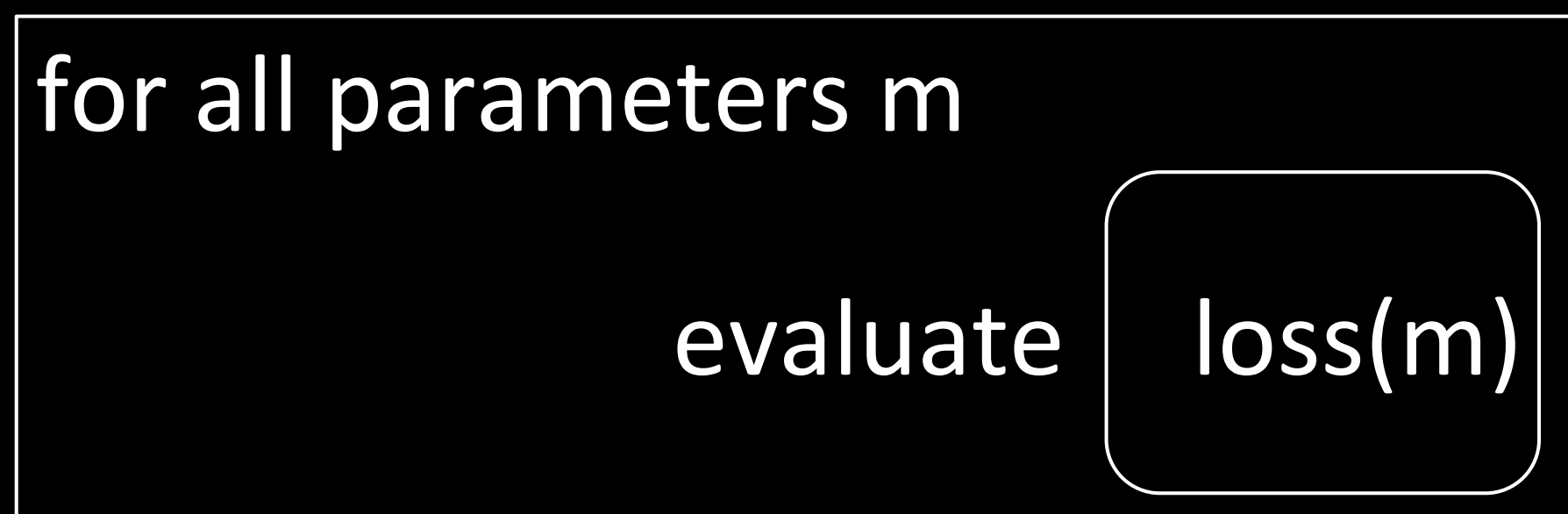
$- \text{image}(m) \|^2$



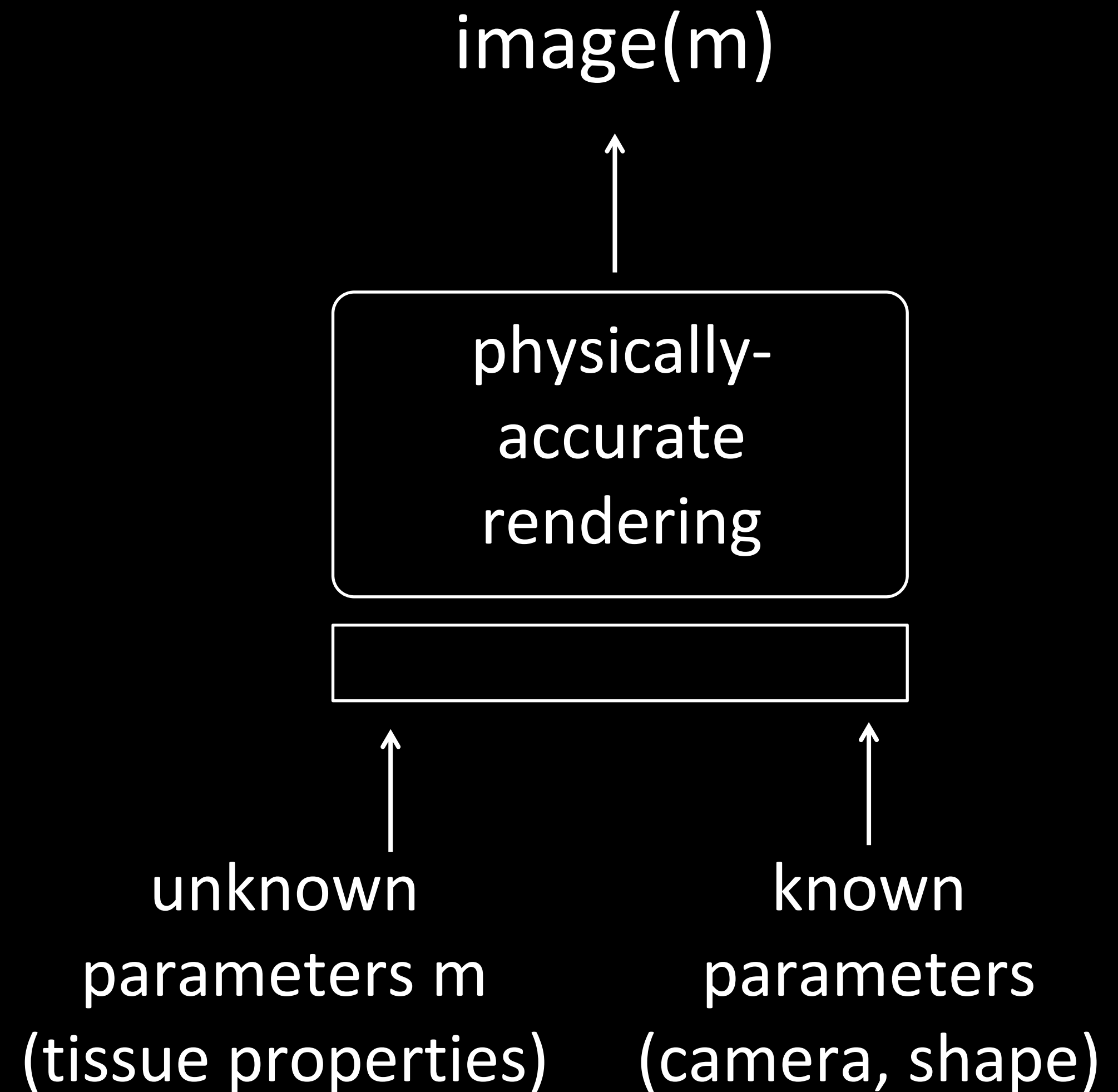
Analysis by synthesis (a.k.a. inverse rendering)

$$\min_{\text{unknowns } m \text{ (tissue properties)}} \left\| \begin{array}{c} \text{stack of images} \\ \text{image}(m) \end{array} \right\|^2$$

solve with exhaustive search



← computed with rendering



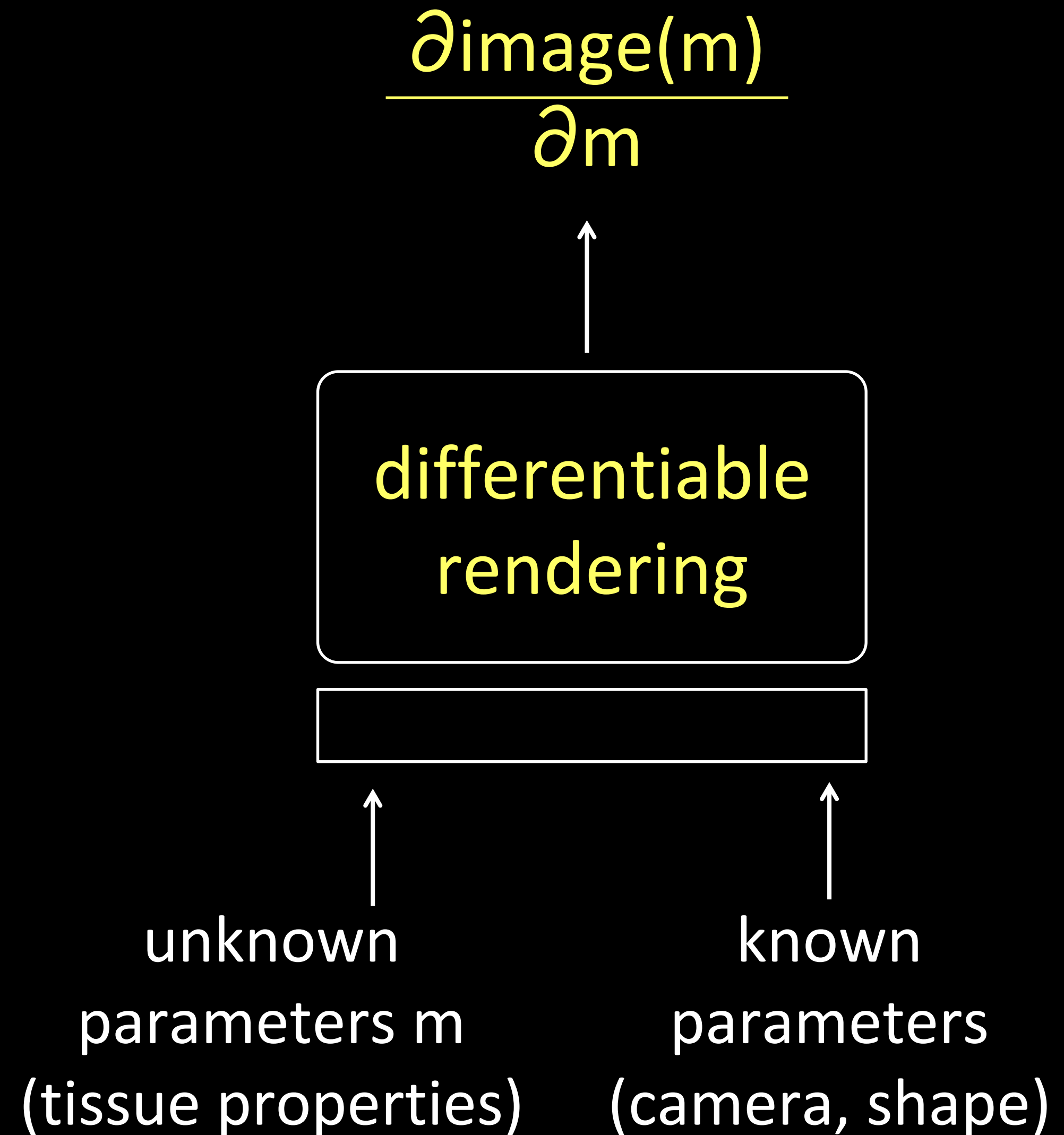
Analysis by synthesis (a.k.a. inverse rendering)

$$\min_{\text{unknowns } m \text{ (tissue properties)}} \| \text{stack of images} - \text{image}(m) \|^2$$

solve with gradient descent

while (not converged)
update m with $\frac{\partial \text{loss}(m)}{\partial m}$

← computed with differentiable rendering

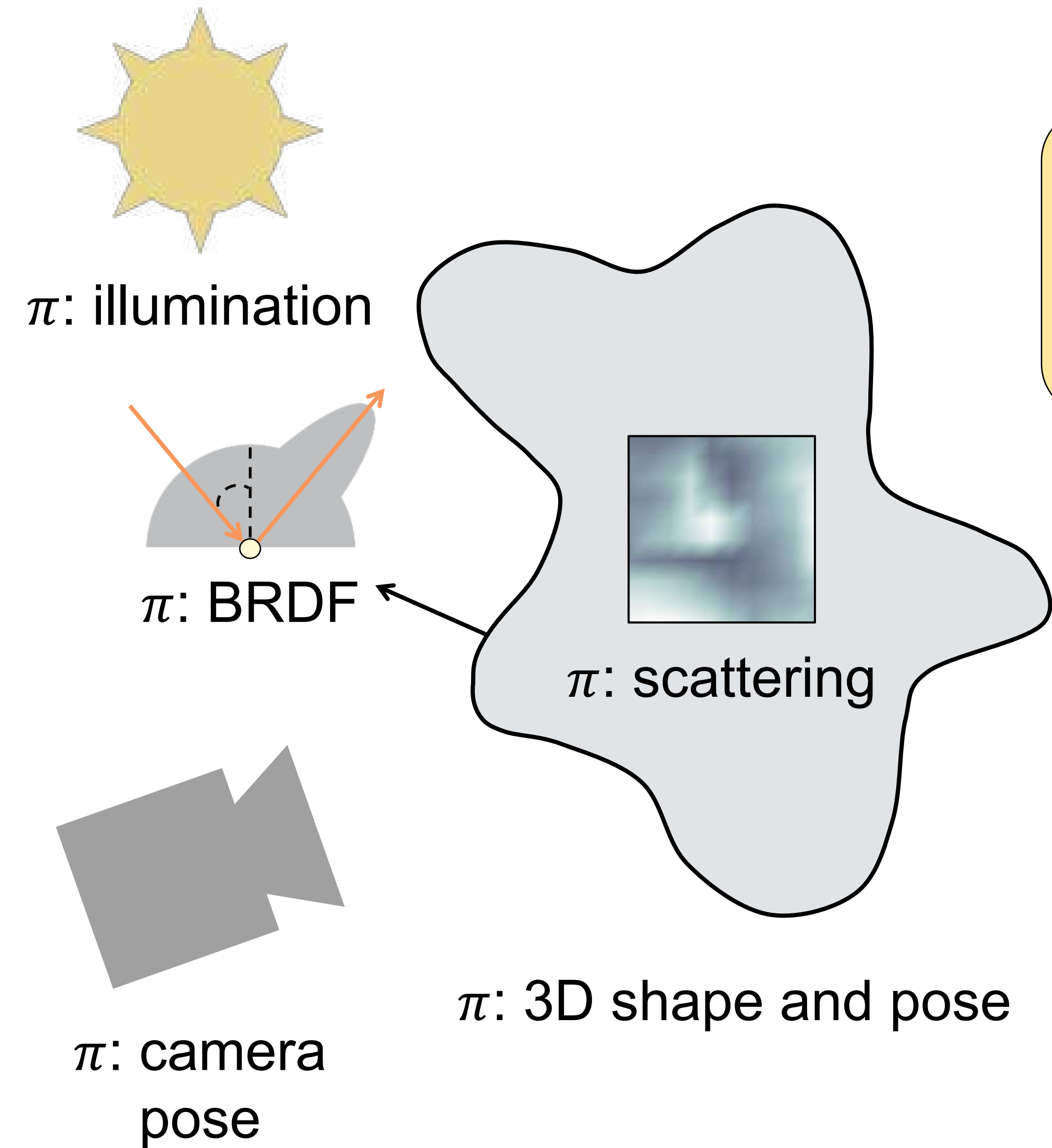


Analysis by synthesis (a.k.a. inverse rendering)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

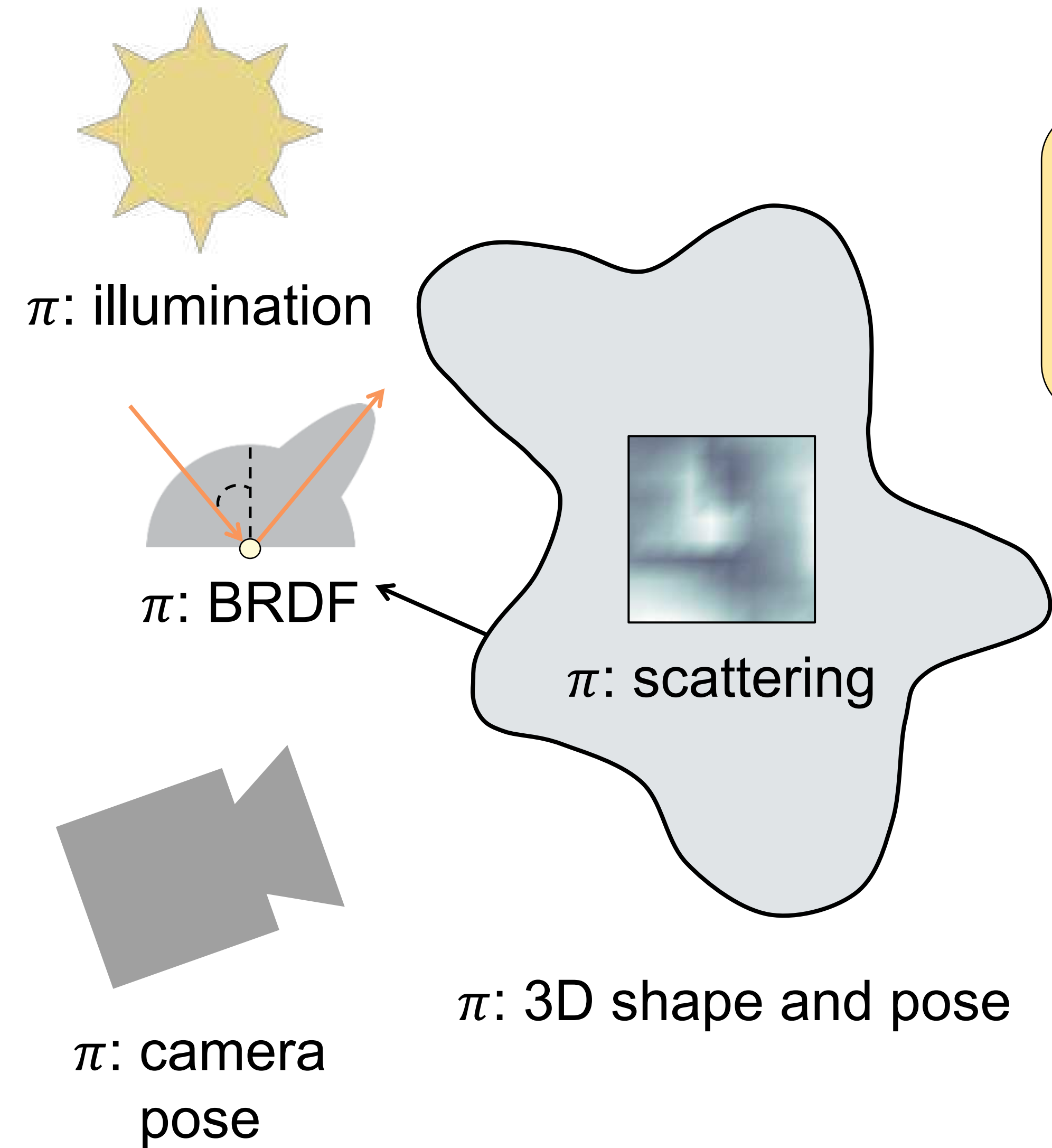

Analysis by synthesis (a.k.a. inverse rendering)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{reference image}, \text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

Analysis by synthesis (a.k.a. inverse rendering)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{image of pumpkin}, \text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

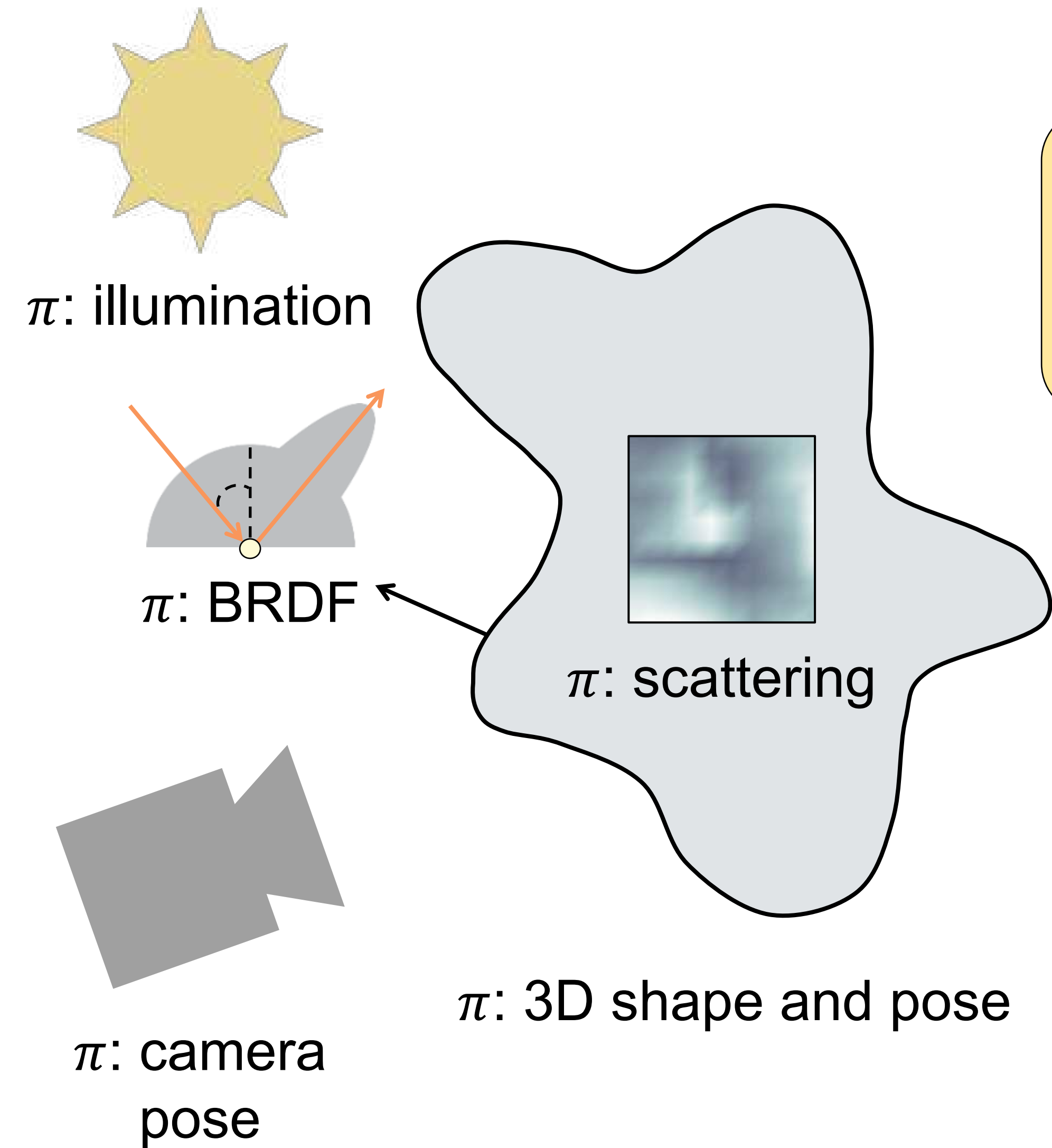
Stochastic gradient descent (e.g., Adam):

initialize $\pi \leftarrow \pi_0$

while (not converged)

update $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Analysis by synthesis (a.k.a. inverse rendering)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$


Stochastic gradient descent (e.g., Adam):

initialize $\pi \leftarrow \pi_0$

while (not converged)

update $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Differentiable rendering

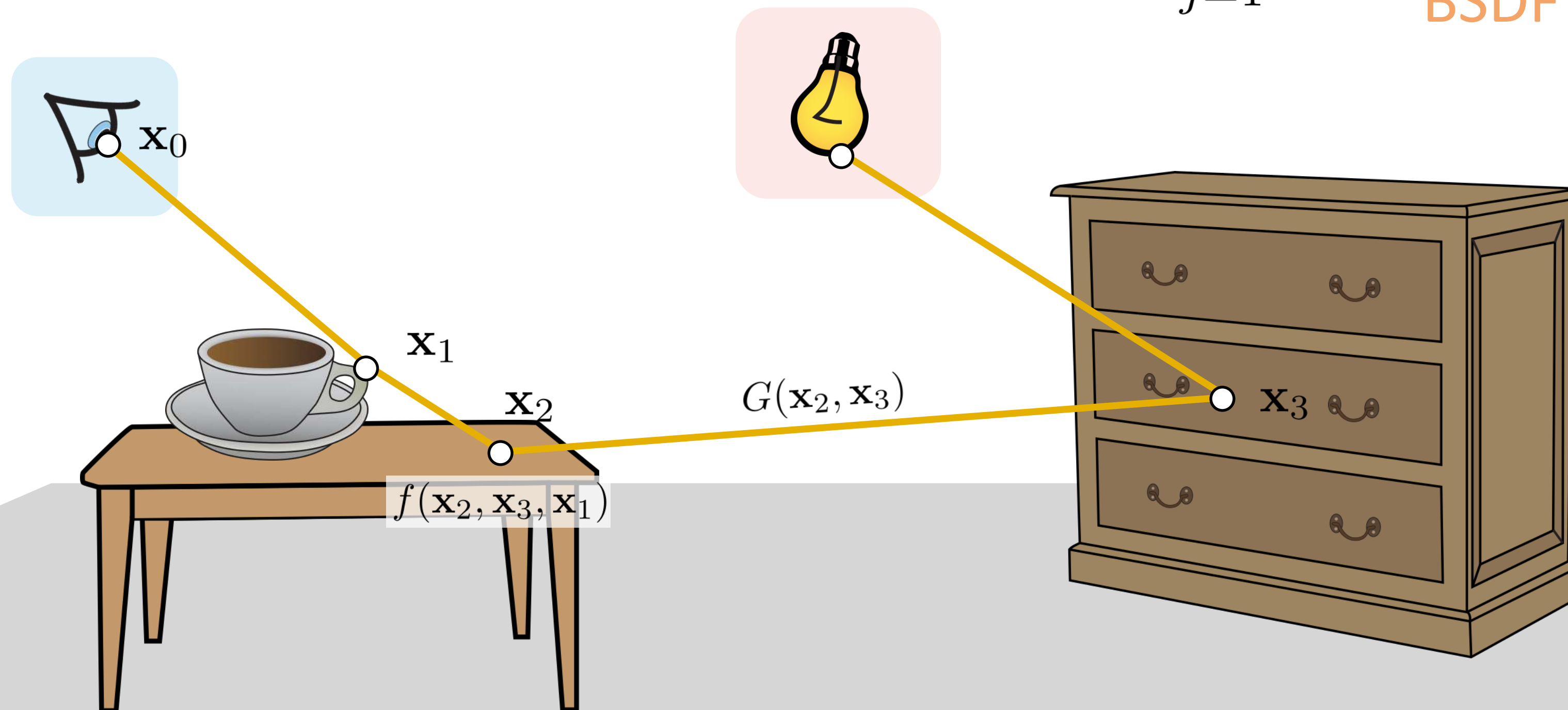
How do we differentiate light transport?

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

sensor weight
source weight
path throughput
light path
space of all light paths

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

BSDF
geometry



REMINDER (?) FROM CALCULUS

Reminder from calculus

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx \quad \stackrel{?}{=} \quad$$

Reminder from calculus

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx \quad \stackrel{?}{=} \quad$$

Reminder from calculus

Differentiation under the integral sign

Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{\infty} f(x, \pi) dx = ?$$

Reminder from calculus

Differentiation under the integral sign

Also known as the Leibniz integral rule

$$\begin{aligned} \frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx &= \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx \\ &+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi} \\ &+ \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi} \end{aligned}$$

Reminder from calculus

Differentiation under the integral sign

Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx$$

Move derivative
inside integral

$$+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi}$$

$$+ \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$$

Reminder from calculus

Differentiation under the integral sign

Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx$$

Move derivative
inside integral

Account for changes in
integration limits

$$+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi}$$

$$+ \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$$

Reminder from calculus

Differentiation under the integral sign

Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx$$

Move derivative
inside integral

Account for changes in
integration limits

$$+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi}$$

Account for discontinuities of
integrand that depend on π

$$+ \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$$

A simple example

$$f(x, \pi) = \begin{cases} 0 & \text{if } x < 2\pi \\ 1 & \text{if } x \geq 2\pi \end{cases}$$

$$\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) dx =$$

A simple example

$$f(x, \pi) = \begin{cases} 0 & \text{if } x < 2\pi \\ 1 & \text{if } x \geq 2\pi \end{cases}$$

$$\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) dx =$$

A simple example

$$f(x, \pi) = \begin{cases} 0 & \text{if } x < 2\pi \\ 1 & \text{if } x \geq 2\pi \end{cases}$$

$$\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) dx = \int_0^{2\pi} \frac{d}{d\pi} 0 dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 dx$$

Move derivative inside integral

A simple example

$$f(x, \pi) = \begin{cases} 0 & \text{if } x < 2\pi \\ 1 & \text{if } x \geq 2\pi \end{cases}$$

$$\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) dx = \int_0^{2\pi} \frac{d}{d\pi} 0 dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 dx \quad \text{Move derivative inside integral}$$

Account for changes in integration limits

$$+ 1 \frac{d(4\pi)}{d\pi} - 0 \frac{d0}{d\pi}$$

A simple example

$$f(x, \pi) = \begin{cases} 0 & \text{if } x < 2\pi \\ 1 & \text{if } x \geq 2\pi \end{cases}$$

$$\frac{d}{d\pi} \int_0^{4\pi} f(x, \pi) dx = \int_0^{2\pi} \frac{d}{d\pi} 0 dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 dx \quad \text{Move derivative inside integral}$$

Account for changes in integration limits

$$+ 1 \frac{d(4\pi)}{d\pi} - 0 \frac{d0}{d\pi}$$

Account for discontinuities of integrand that depend on π

$$+ (0 - 1) \frac{d(2\pi)}{d\pi}$$

Leibniz integral rule

Differentiation under the integral sign
Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx$$

Move derivative
inside integral

Account for changes in
integration limits

$$+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi}$$

Account for discontinuities of
integrand that depend on π

$$+ \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$$

Leibniz integral rule

Differentiation under the integral sign
Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx$$

Move derivative
inside integral

Account for changes in
integration limits

$$+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi}$$

Account for discontinuities of
integrand that depend on π

$$+ \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$$

Leibniz integral rule

Differentiation under the integral sign
Also known as the Leibniz integral rule

$$\frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) dx =$$

Interior integral

$$\int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) dx$$

Move derivative inside integral

Account for changes in integration limits

Boundary terms

$$+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(a(\pi), \pi) \frac{da(\pi)}{d\pi}$$

Account for discontinuities of integrand that depend on π

$$+ \sum_i (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$$

Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) \stackrel{?}{=} \quad$$

Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) \stackrel{?}{=} \quad$$

Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) \stackrel{?}{=} \quad ?$$

Reynolds transport theorem [1903]

Generalization of the Leibniz rule

Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x, \pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)$$

Reynolds transport theorem [1903]

Generalization of the Leibniz rule

Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x, \pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)$$

Boundary domain

||

discontinuity points \cup boundary of domain Ω
(if they depend on π)

Reynolds transport theorem [1903]
Generalization of the Leibniz rule

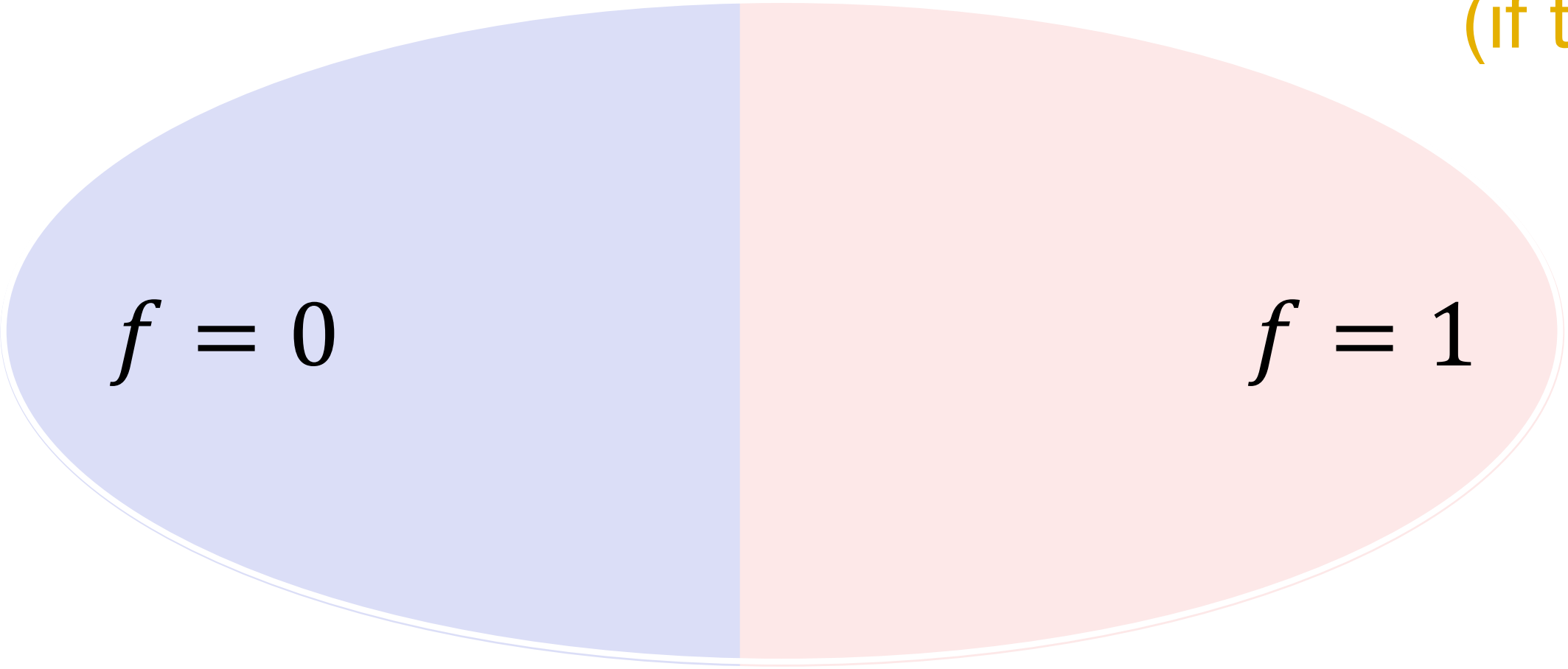
Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x, \pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)$$

Boundary domain

Reynolds transport theorem [1903]
 Generalization of the Leibniz rule

||
 discontinuity points \cup boundary of domain Ω
 (if they depend on π)



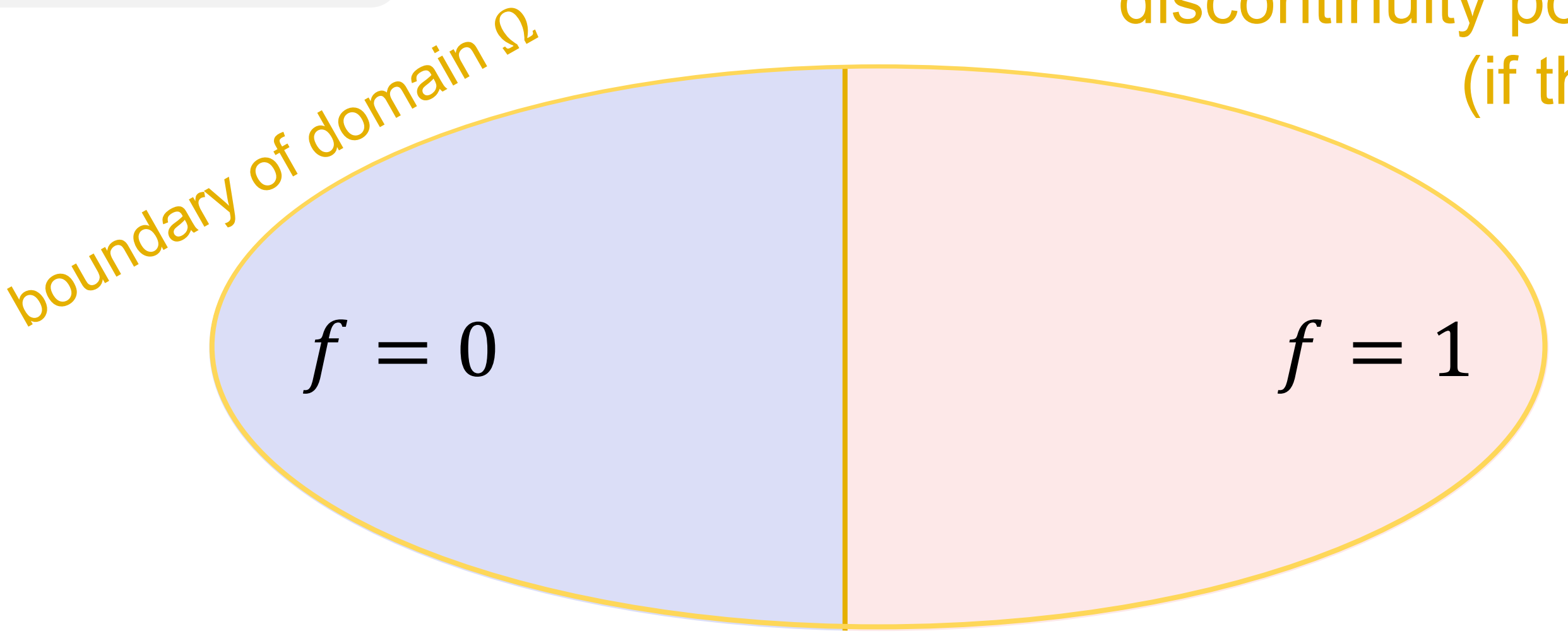
Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x, \pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)$$

Boundary domain

Reynolds transport theorem [1903]
 Generalization of the Leibniz rule

||
 discontinuity points \cup boundary of domain Ω
 (if they depend on π)



discontinuity points

π



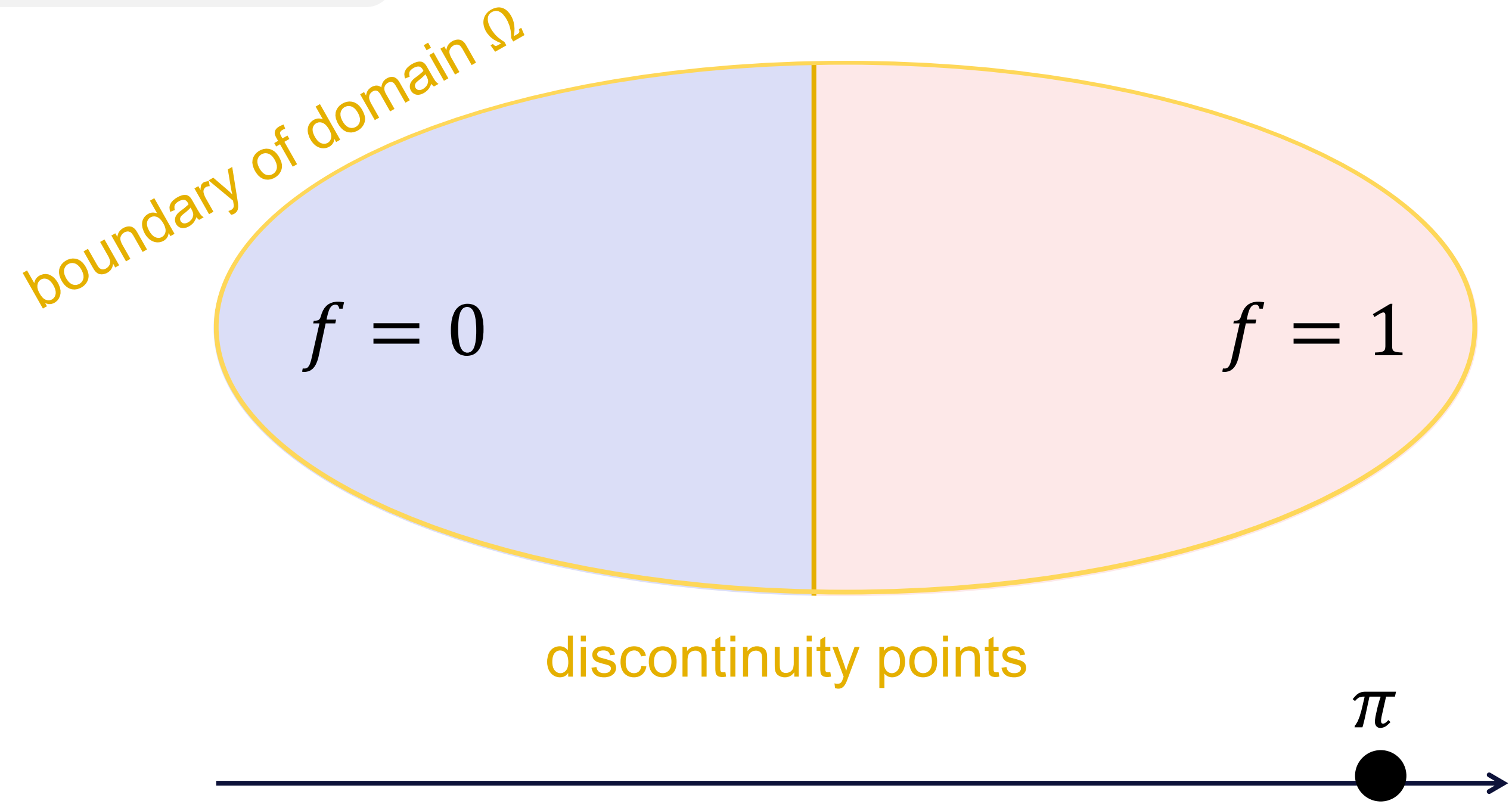
Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x, \pi) dA(x) = \int_{\Omega(\pi)} \frac{df(x, \pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x, \pi) dl(x)$$

Reynolds transport theorem [1903]
Generalization of the Leibniz rule

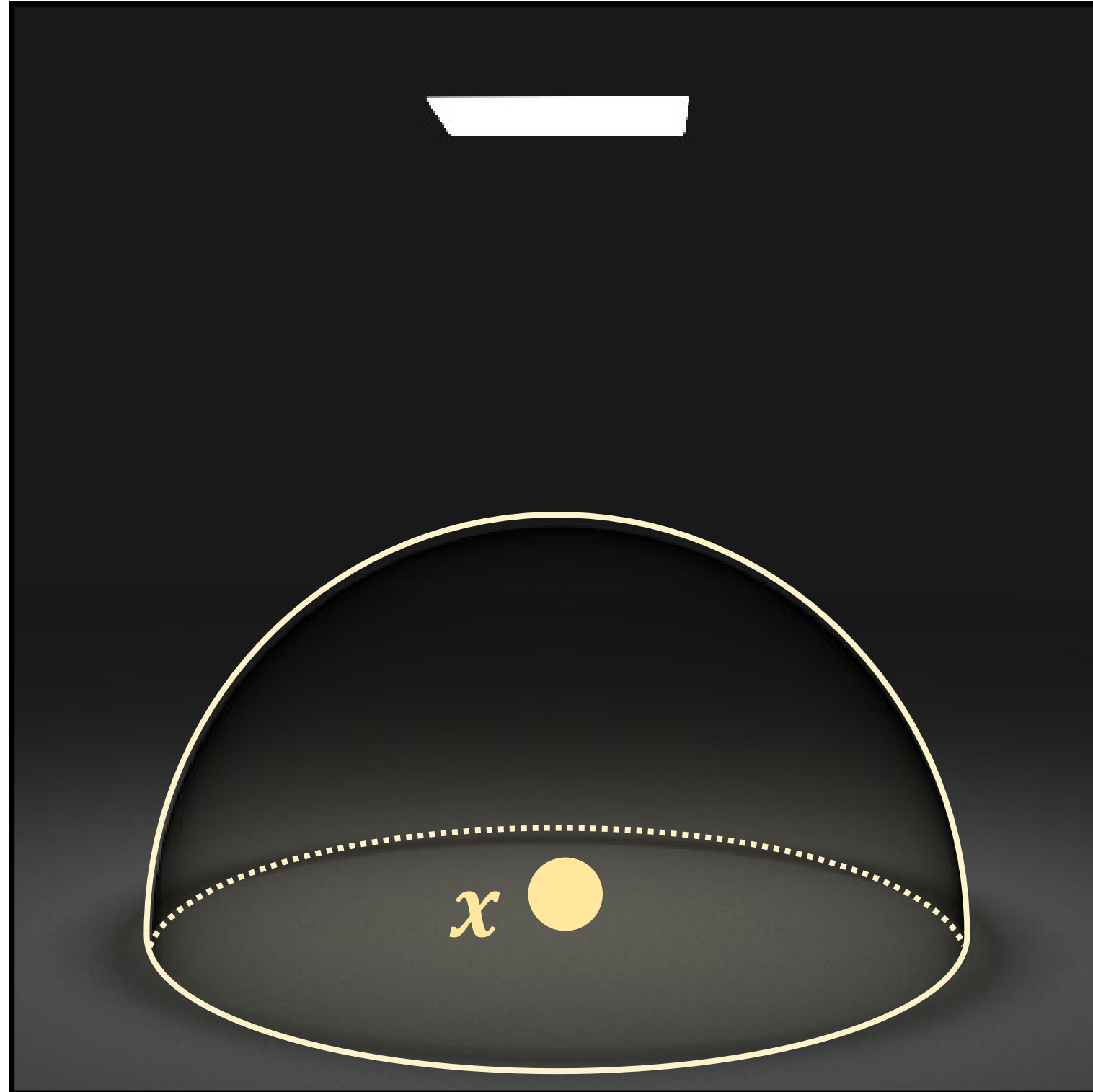
Interior integral

Boundary integral



DIFFERENTIATING DIRECT ILLUMINATION

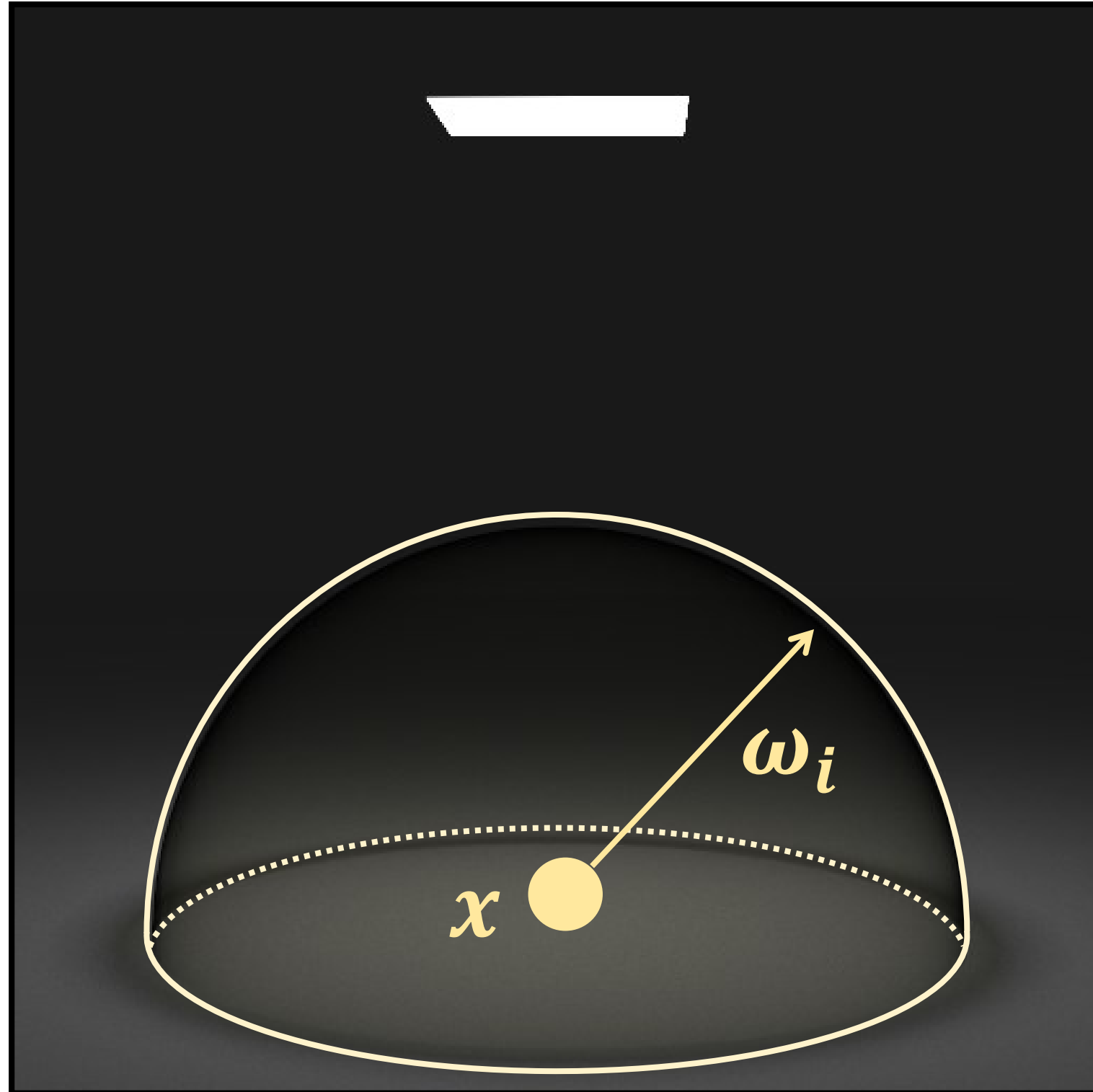
Direct illumination integral



Radiance from x :

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) d\sigma(\omega_i)$$

Direct illumination integral

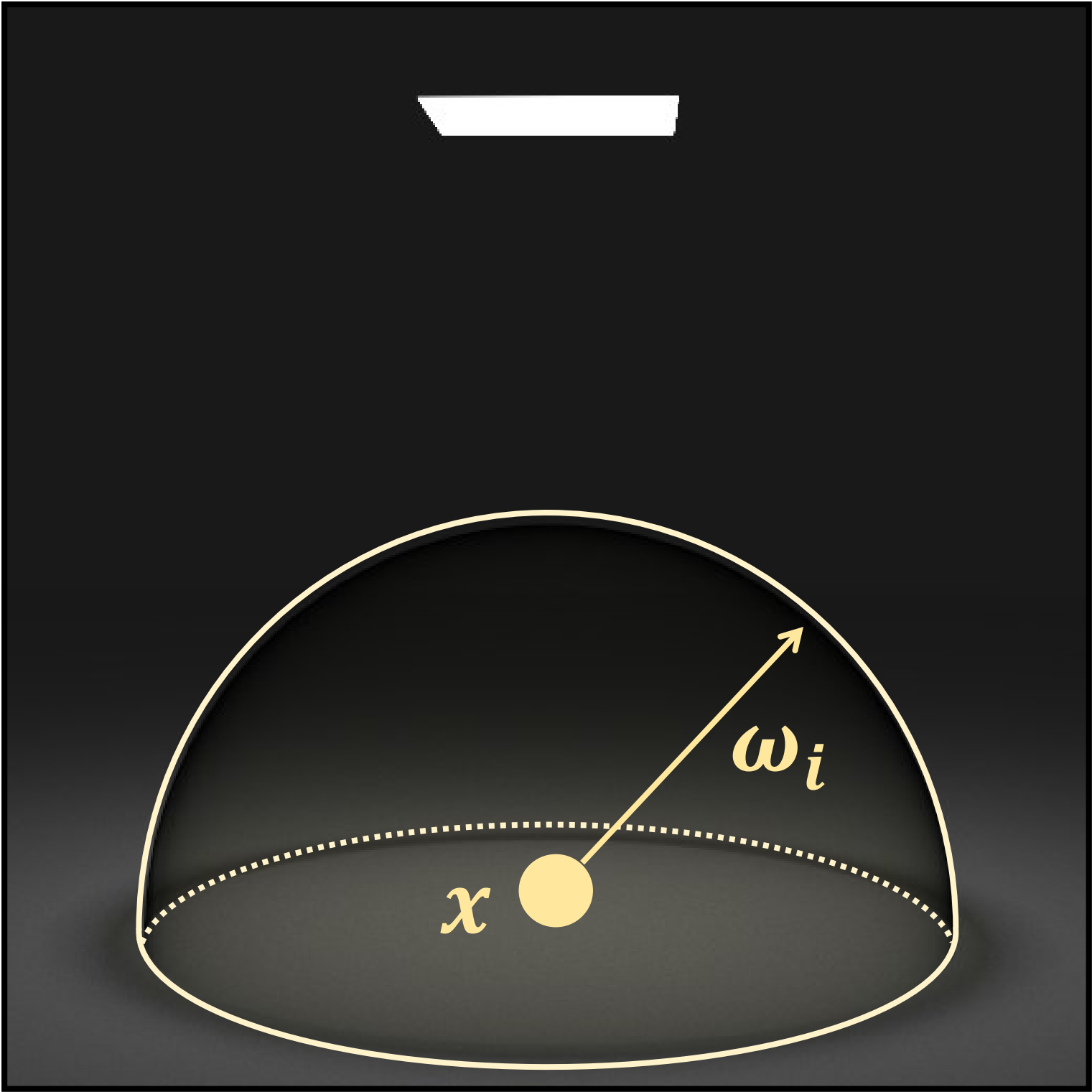


Radiance from x :

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) d\sigma(\omega_i)$$

Unit hemisphere

Direct illumination integral

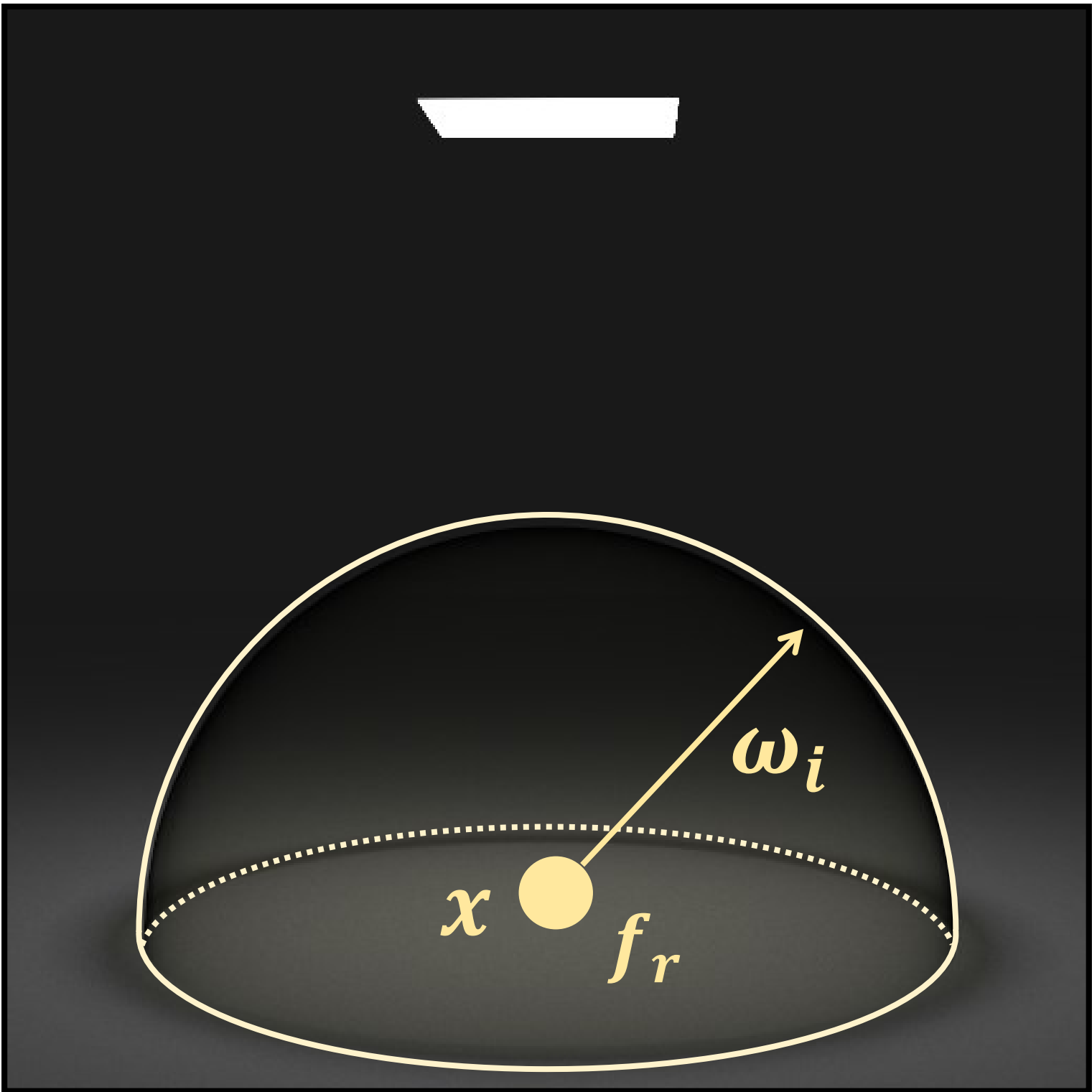


Radiance from x :

$$I = \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) \overset{\text{Incident radiance}}{L_i(\omega_i)} (n \cdot \omega_i) d\sigma(\omega_i)$$

Unit hemisphere

Direct illumination integral

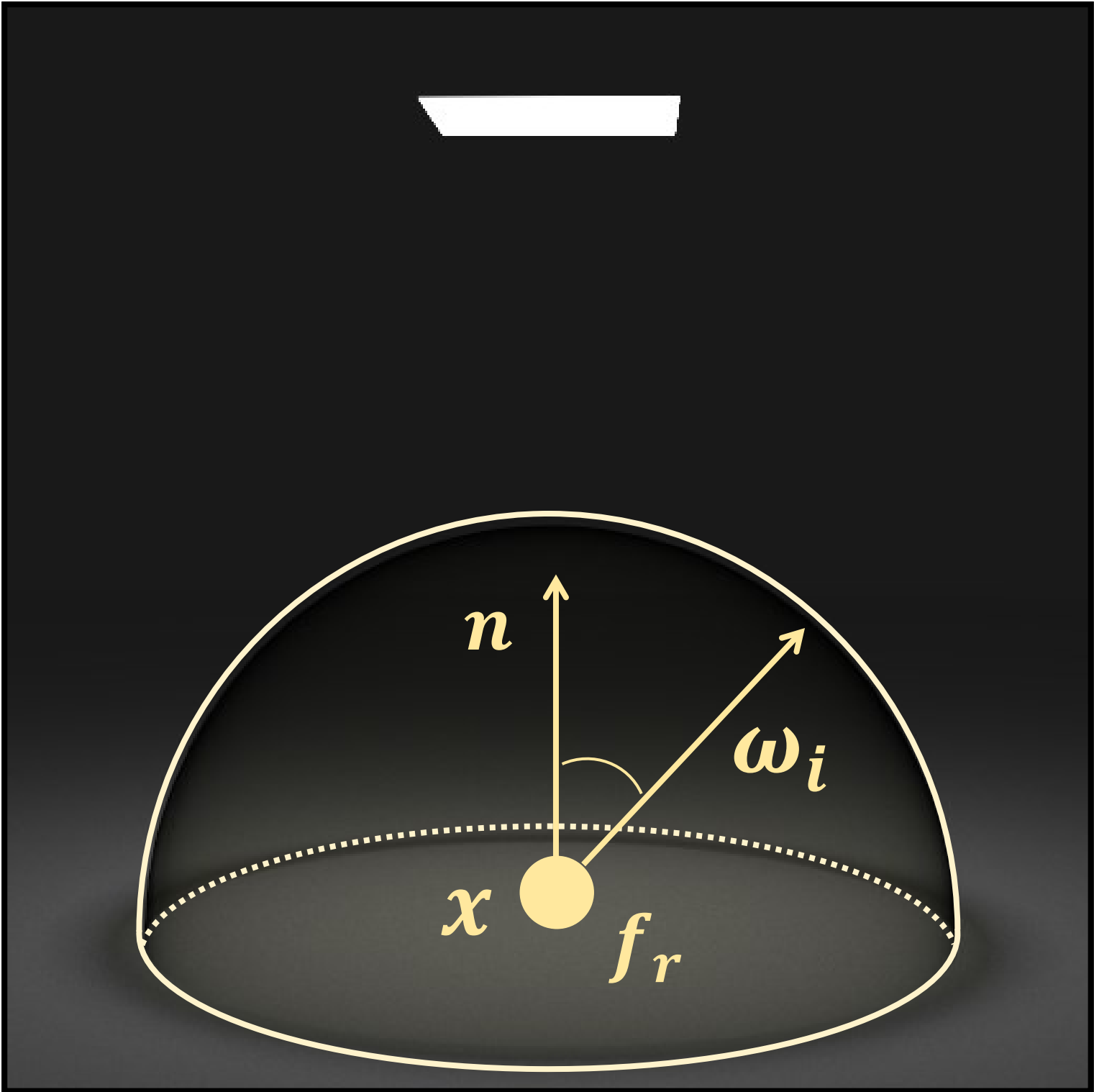


Radiance from x :

$$I = \int_{\mathbb{H}^2} \overset{\substack{\text{Reflectance} \\ \text{(BRDF)}}}{f_r(\omega_i, \omega_o)} \overset{\substack{\text{Incident} \\ \text{radiance}}}{L_i(\omega_i)} (n \cdot \omega_i) d\sigma(\omega_i)$$

Unit hemisphere

Direct illumination integral

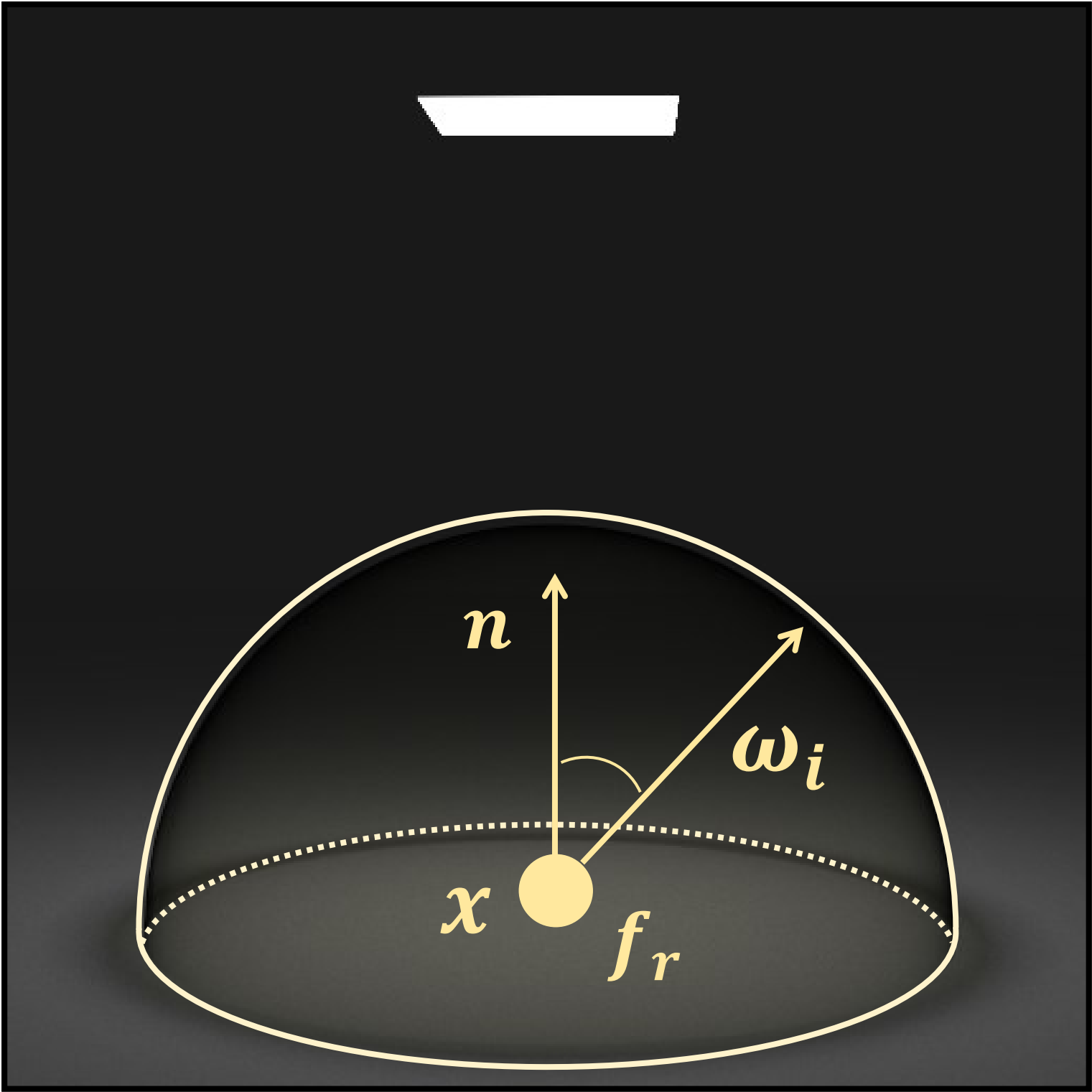


Radiance from x :

$$I = \int_{\mathbb{H}^2} \overset{\substack{\text{Reflectance} \\ \text{(BRDF)}}}{f_r(\omega_i, \omega_o)} \overset{\substack{\text{Incident} \\ \text{radiance}}}{L_i(\omega_i)} \overset{\substack{\text{Shading wrt} \\ \text{normal } n}}{(n \cdot \omega_i)} d\sigma(\omega_i)$$

Unit hemisphere

Direct illumination integral



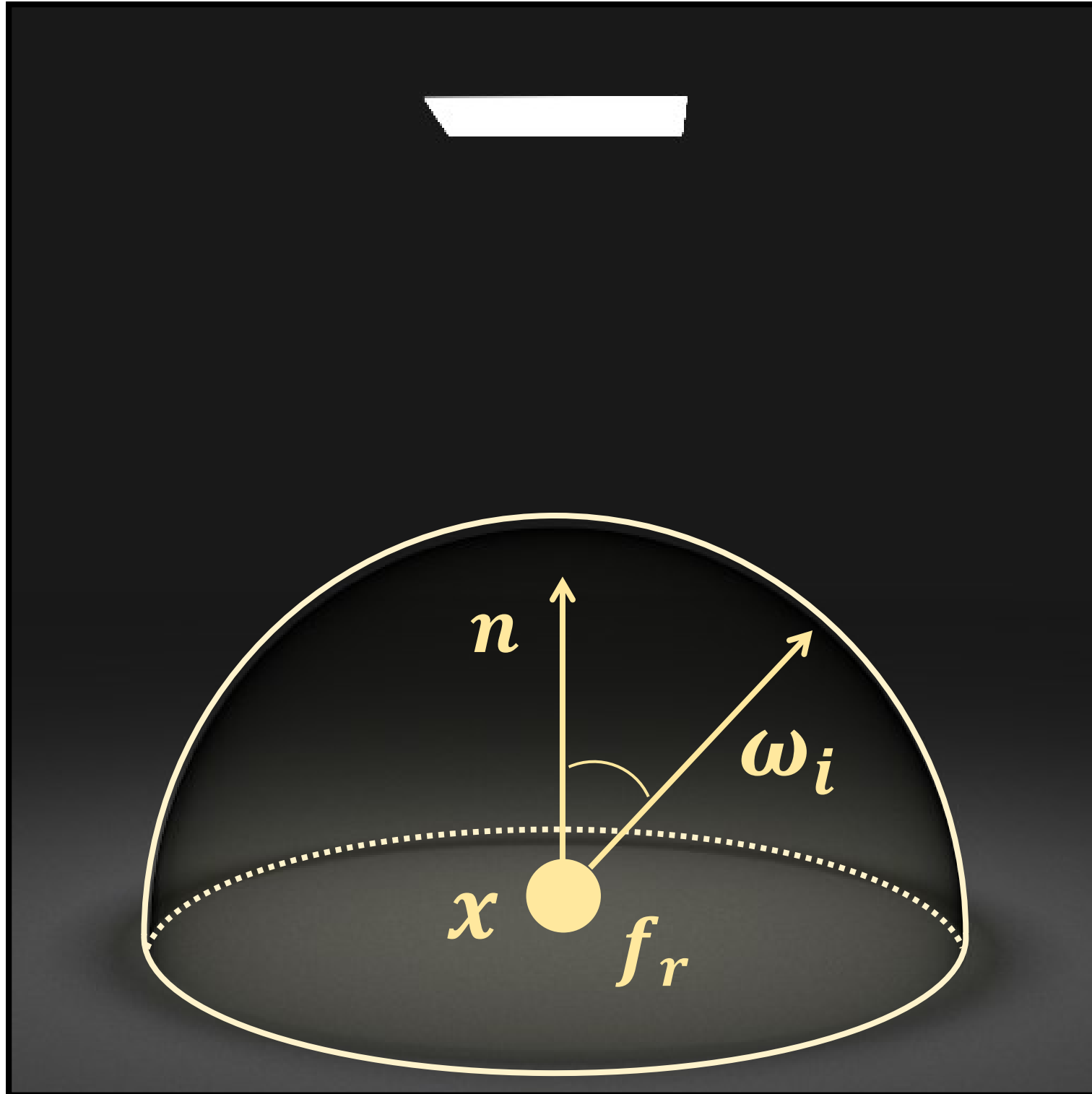
Radiance from x :

$$I = \int_{\mathbb{H}^2} \overset{\text{Reflectance (BRDF)}}{f_r(\omega_i, \omega_o)} \overset{\text{Incident radiance}}{L_i(\omega_i)} \overset{\text{Shading wrt normal } n}{(n \cdot \omega_i)} d\sigma(\omega_i)$$

Unit hemisphere

Monte Carlo rendering:

Direct illumination integral



Radiance from x :

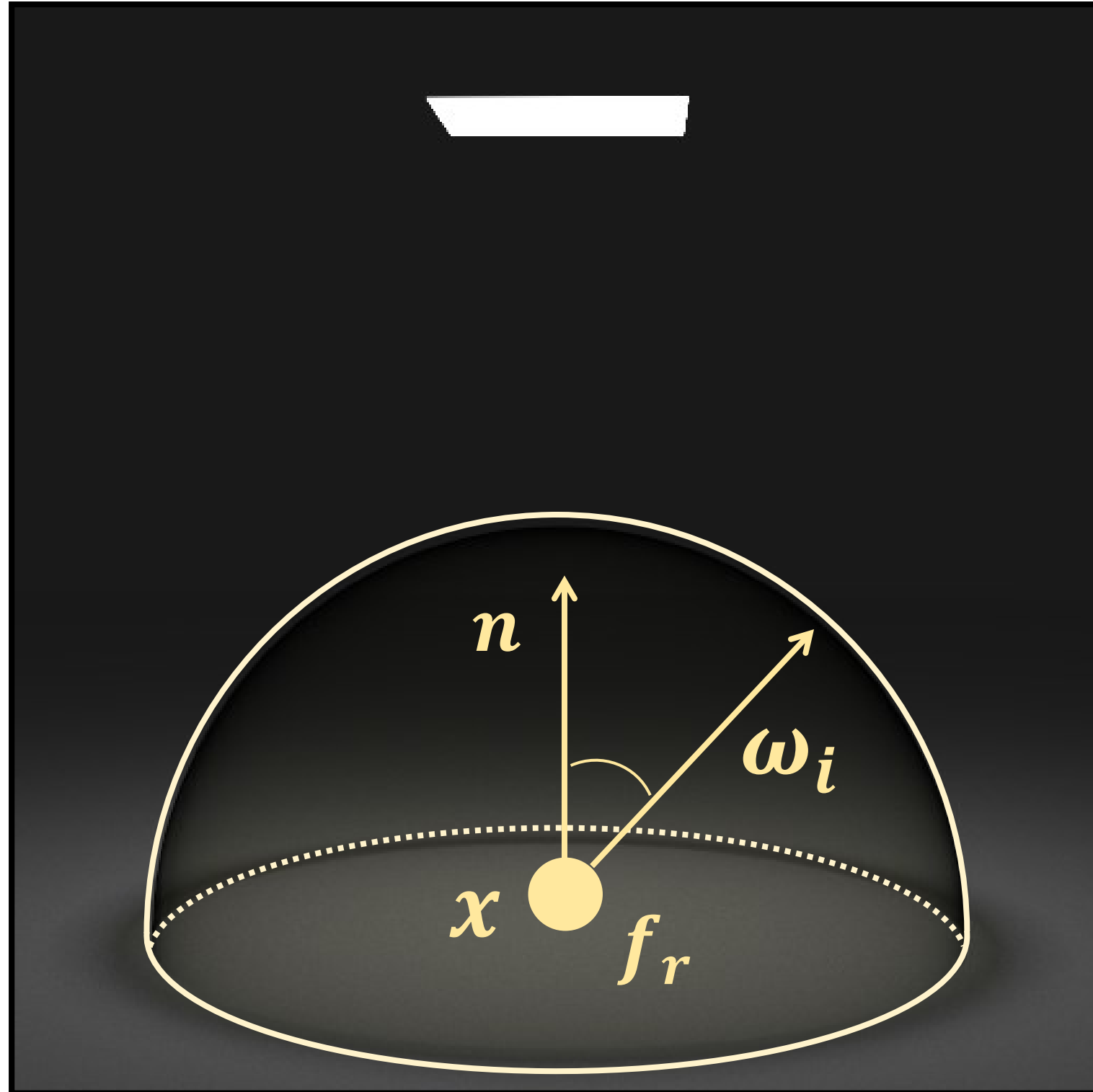
$$I = \int_{\mathbb{H}^2} \overset{\text{Reflectance (BRDF)}}{f_r(\omega_i, \omega_o)} \overset{\text{Incident radiance}}{L_i(\omega_i)} \overset{\text{Shading wrt normal } \mathbf{n}}{(\mathbf{n} \cdot \omega_i)} d\sigma(\omega_i)$$

Unit hemisphere

Monte Carlo rendering:

- Sample random directions ω_i^S from PDF $p(\omega_i)$

Direct illumination integral



Radiance from x :

$$I = \int_{\mathbb{H}^2} \overset{\text{Reflectance (BRDF)}}{f_r(\omega_i, \omega_o)} \overset{\text{Incident radiance}}{L_i(\omega_i)} \overset{\text{Shading wrt normal } n}{(n \cdot \omega_i)} d\sigma(\omega_i)$$

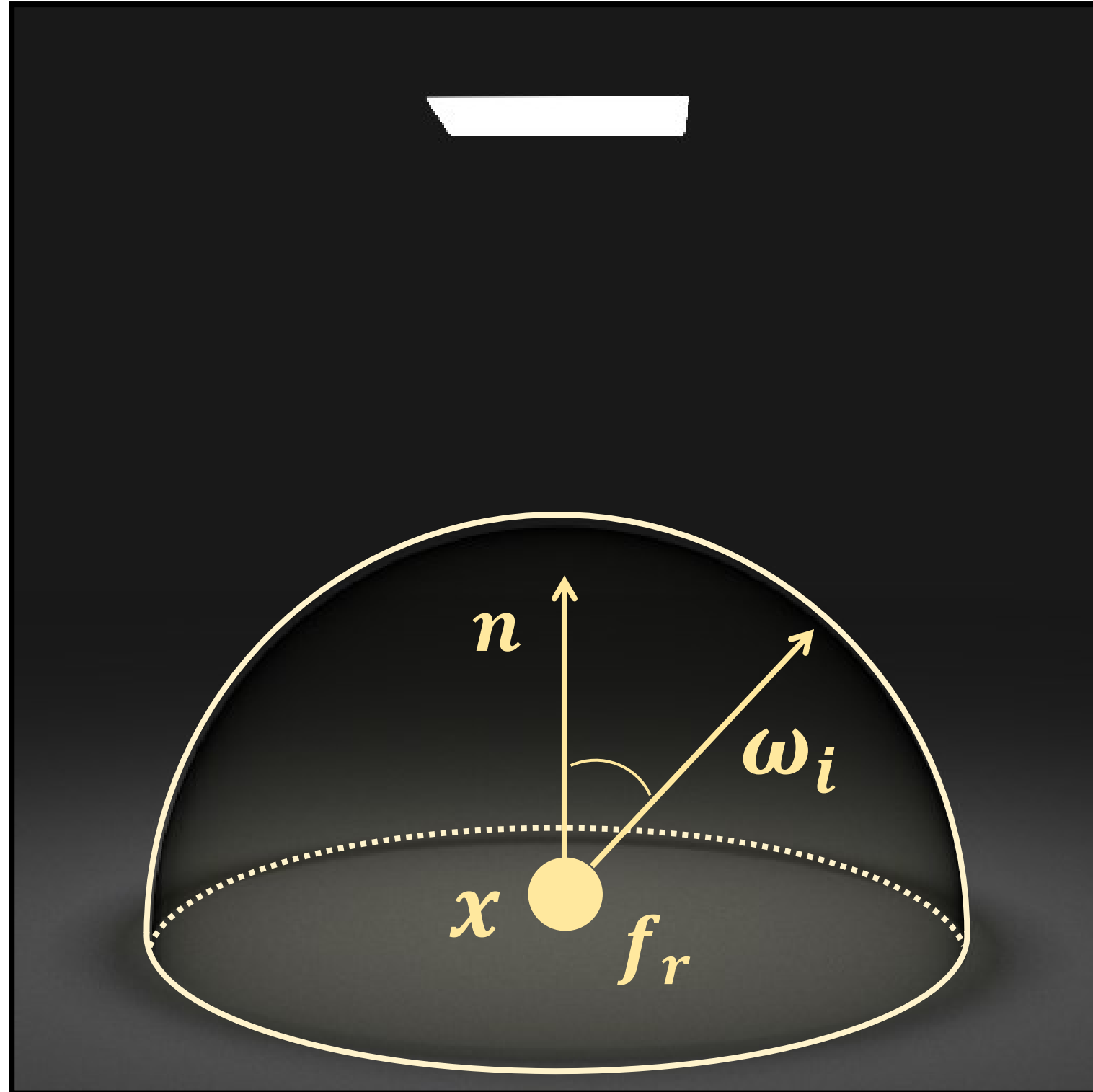
Unit hemisphere

Monte Carlo rendering:

- Sample random directions ω_i^s from PDF $p(\omega_i)$
- Form estimator

$$I \approx \sum_s \frac{f_r(\omega_i^s, \omega_o) L_i(\omega_i^s) (n \cdot \omega_i^s)}{p(\omega_i^s)}$$

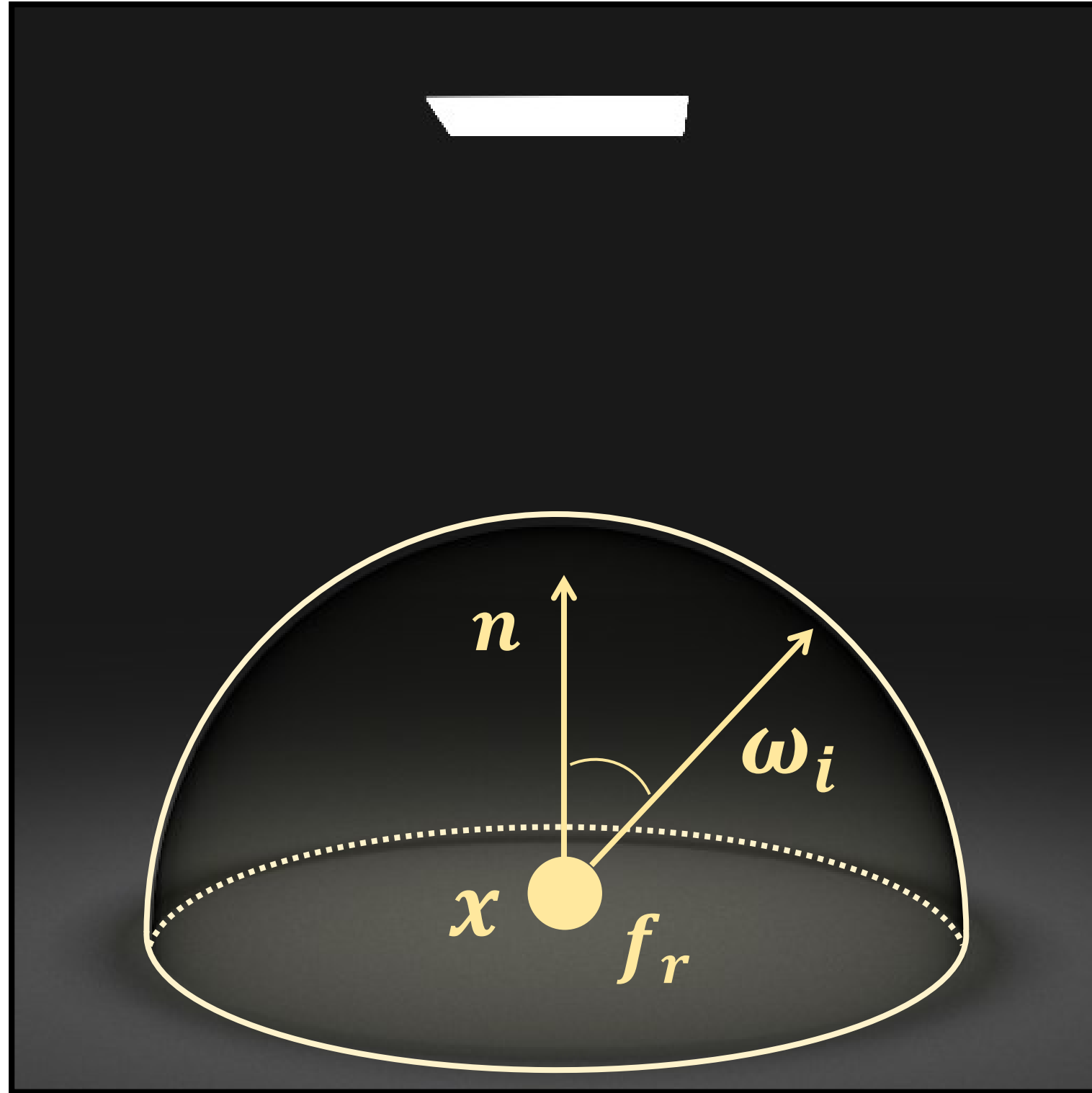
Differential direct illumination



Differential radiance from x :

$$\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) d\sigma(\omega_i)$$

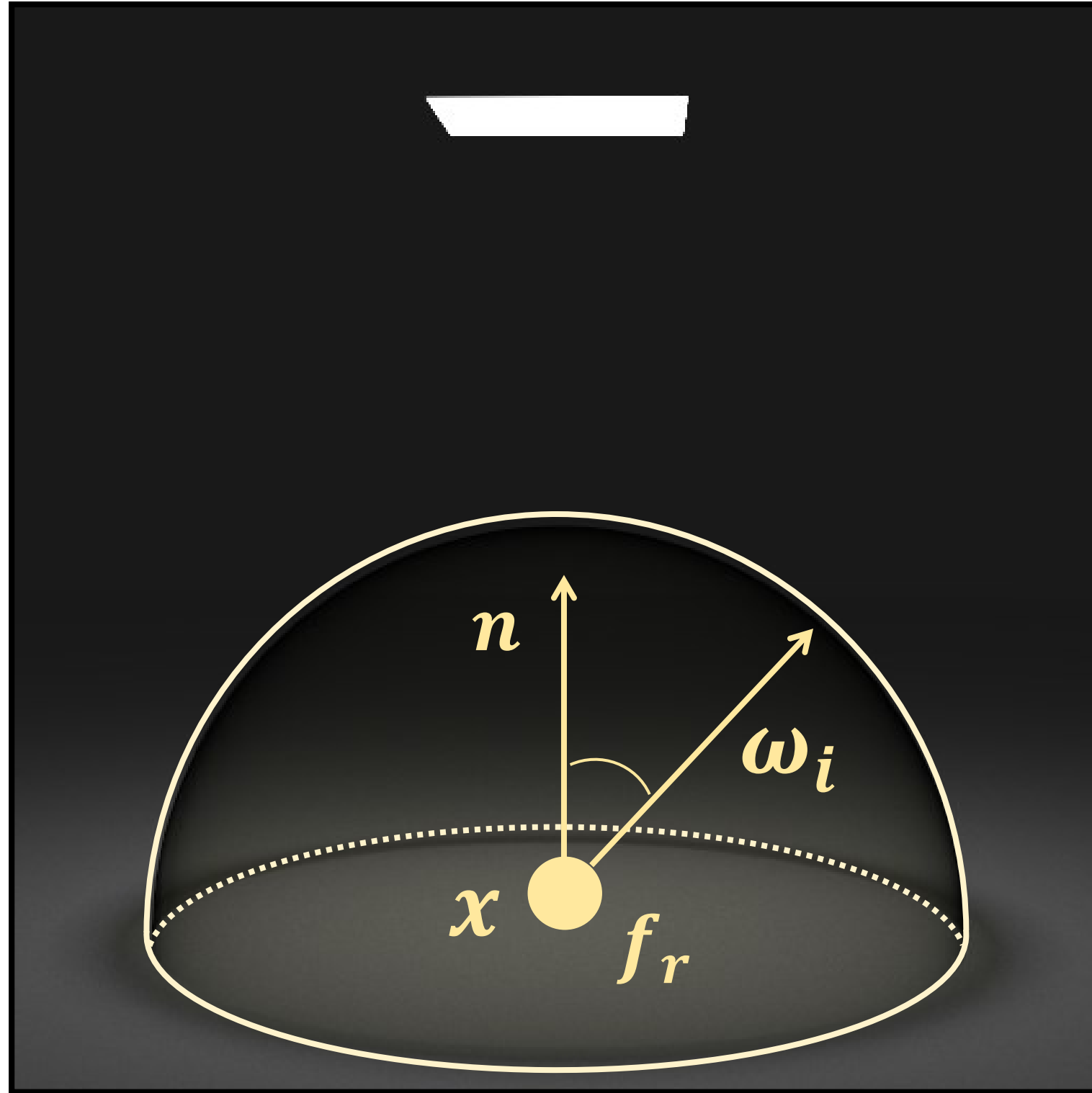
Differential direct illumination: local parameters



Differential radiance from x :

$$\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) d\sigma(\omega_i)$$

Differential direct illumination: local parameters



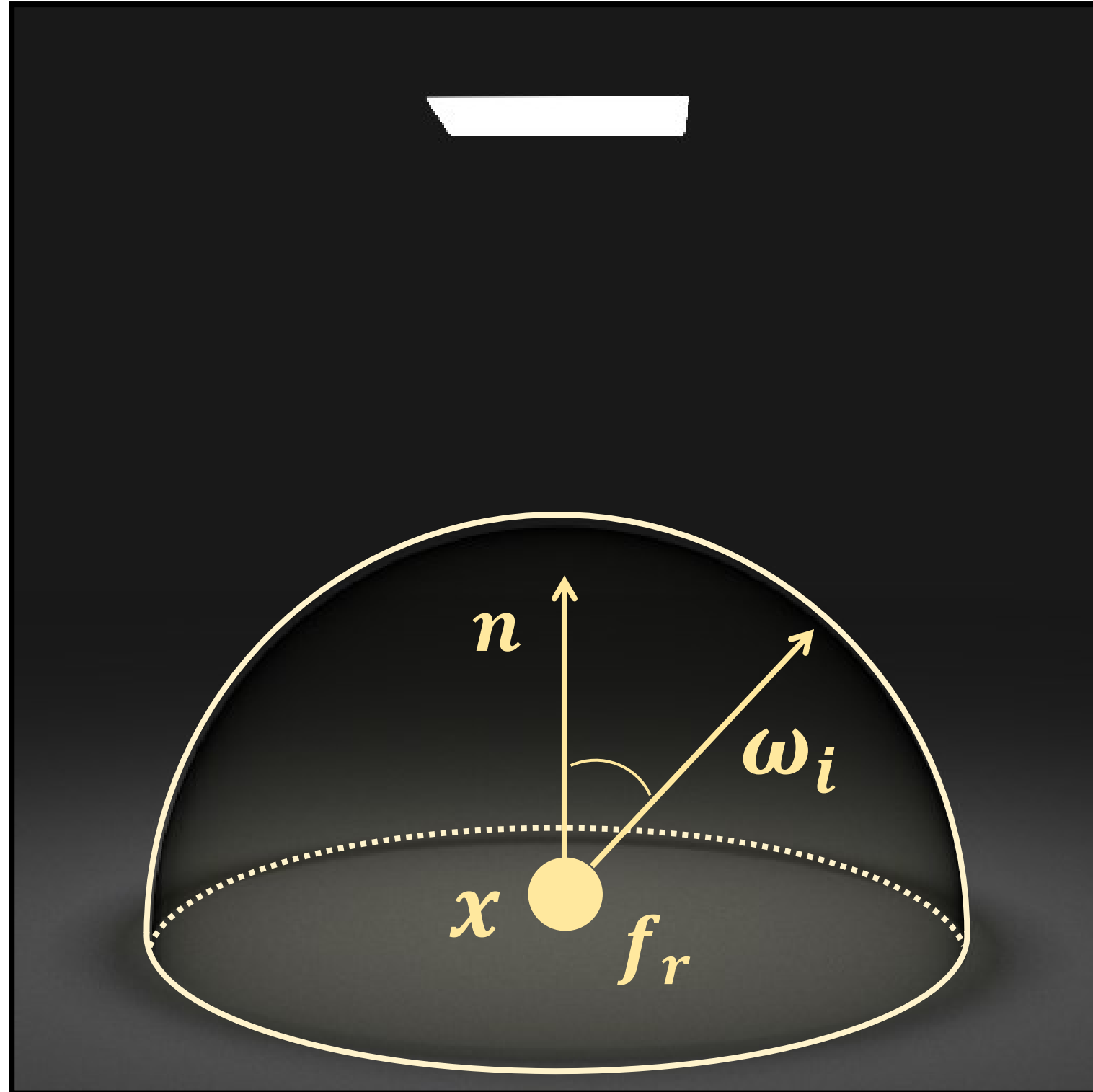
Differential radiance from x :

$$\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) d\sigma(\omega_i)$$

π : local parameters

- BRDF parameters
- *shading* normal
- illumination brightness

Differential direct illumination: local parameters



π : local parameters

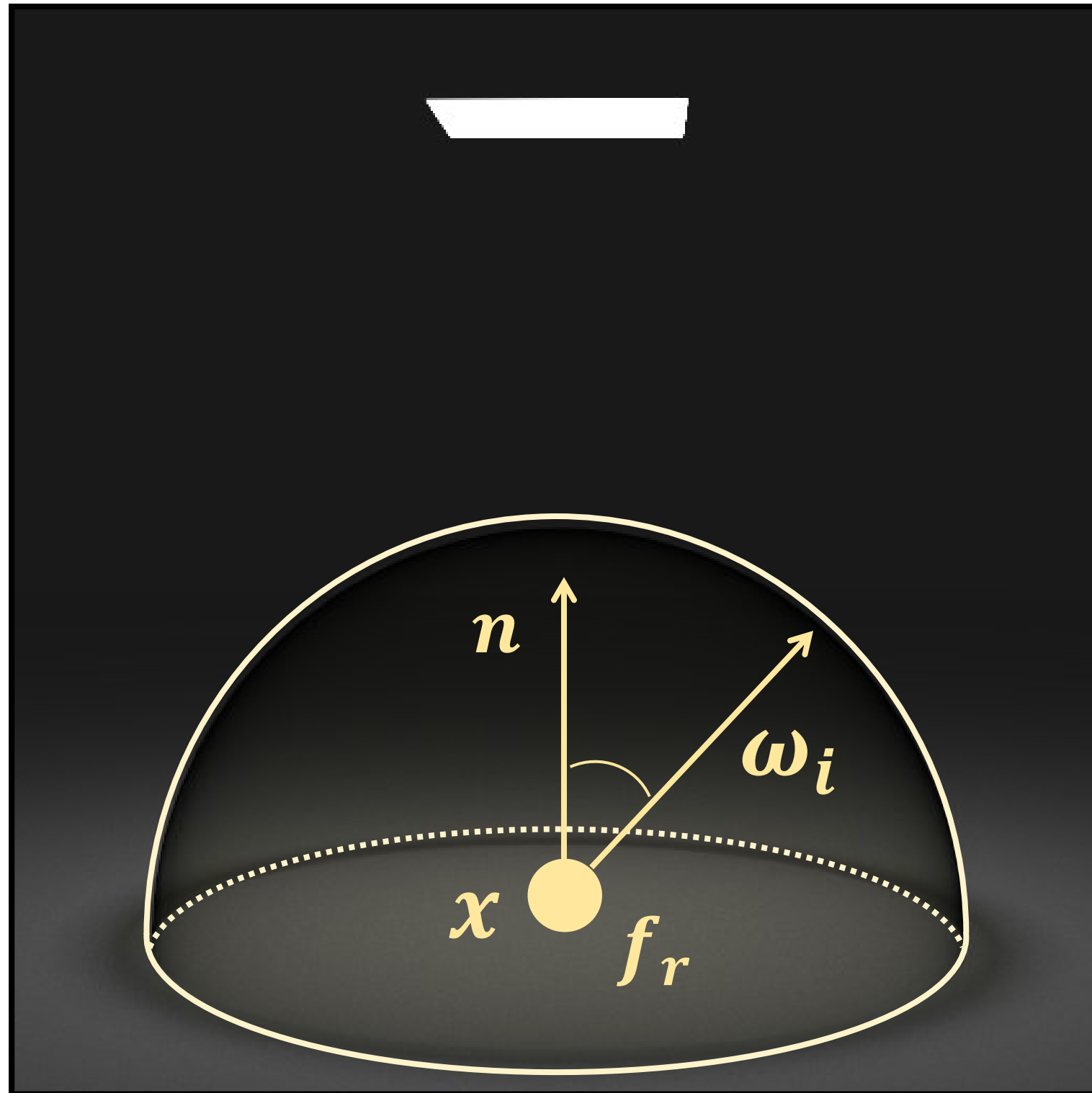
- BRDF parameters
- shading normal
- illumination brightness

Differential radiance from x :

$$\frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{d}{d\pi} \{f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i)\} d\sigma(\omega_i)$$

Just move derivative inside integral

Differential direct illumination: local parameters



π : local parameters

- BRDF parameters
- shading normal
- illumination brightness

Differential radiance from x :

$$\frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{d}{d\pi} \{f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i)\} d\sigma(\omega_i)$$

Just move derivative inside integral

Monte Carlo differentiable rendering:

- Sample random directions ω_i^s from PDF $p(\omega_i)$

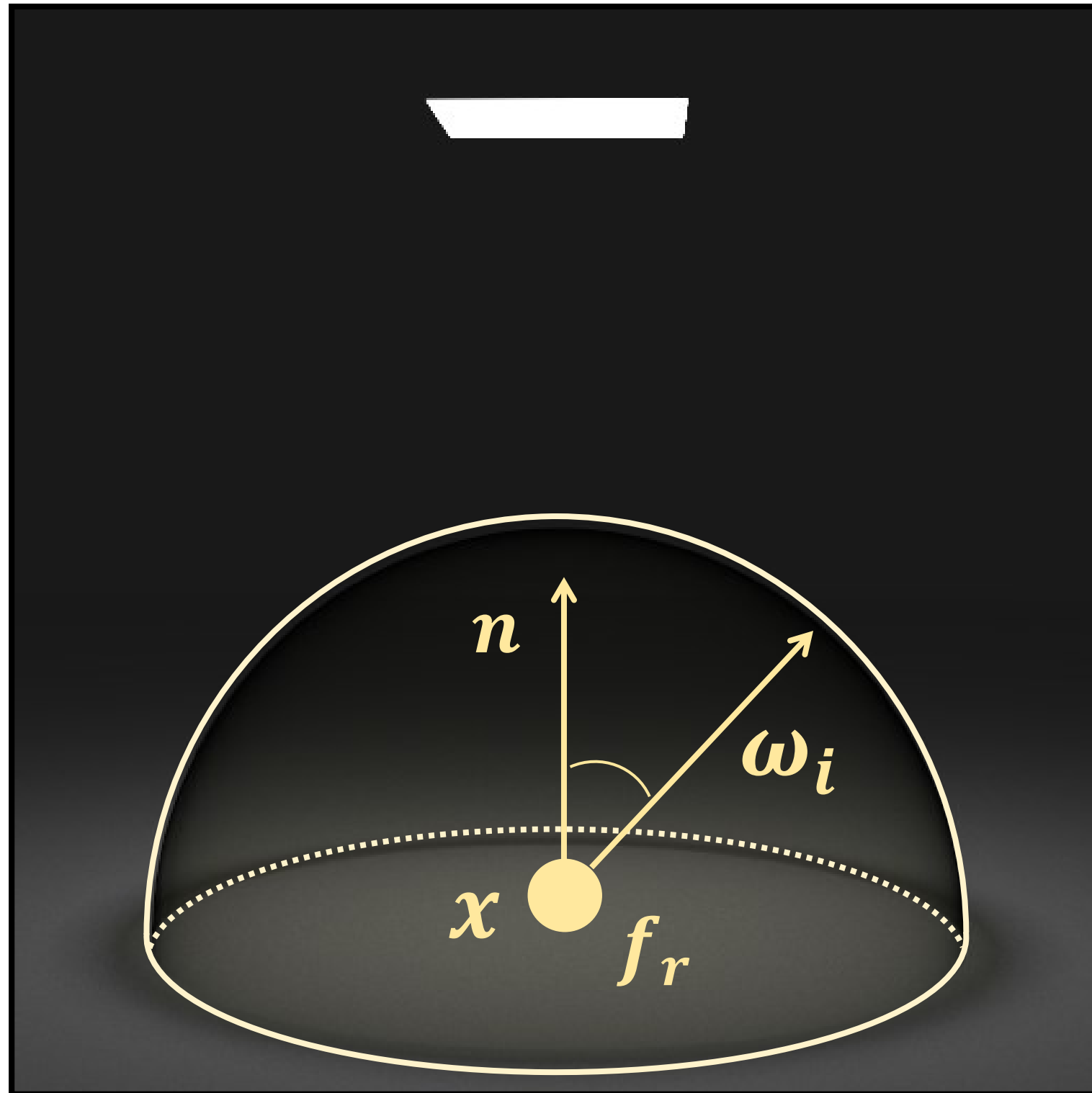
Just differentiate numerator

- Form estimator

[Khungurn et al. 2015, Gkioulekas et al. 2015]

$$\frac{dI}{d\pi} \approx \sum_s \frac{\frac{d}{d\pi} \{f_r(\omega_i^s, \omega_o) L_i(\omega_i^s) (n \cdot \omega_i^s)\}}{p(\omega_i^s)}$$

Alternative estimator



π : local parameters

- BRDF parameters

Differential radiance from x :

$$\frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{d}{d\pi} \{f_r(\omega_i, \omega_o, \pi) L_i(\omega_i) (n \cdot \omega_i)\} d\sigma(\omega_i)$$

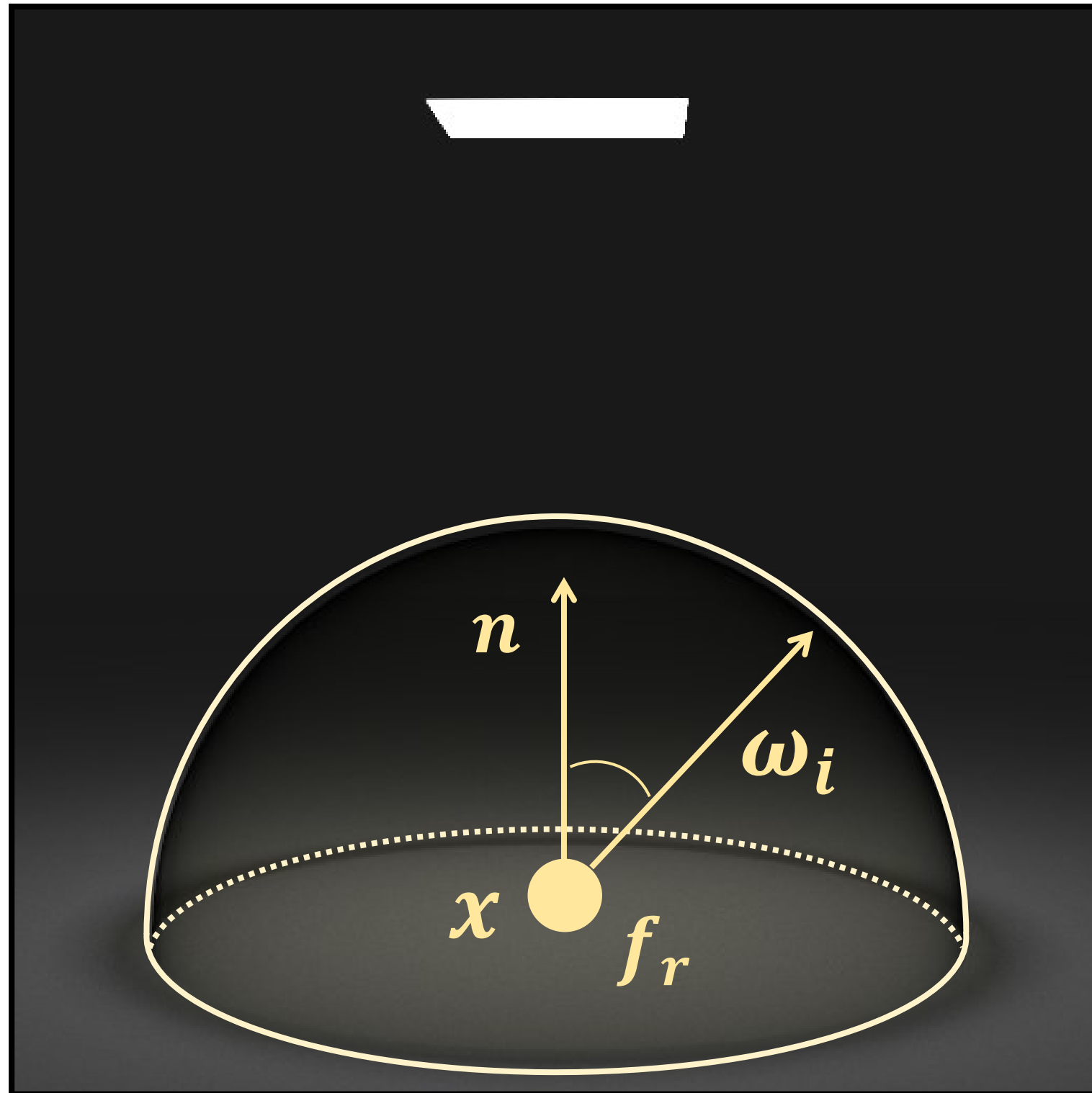
Just move derivative inside integral

Monte Carlo estimation:

- Sample random directions ω_i^s from PDF $p(\omega_i, \pi)$
- Form estimator

$$\frac{dI}{d\pi} \approx \sum_s \frac{\frac{d}{d\pi} \{f_r(\omega_i^s, \omega_o, \pi) L_i(\omega_i^s) (n \cdot \omega_i^s)\}}{p(\omega_i^s, \pi)}$$

Alternative estimator



π : local parameters

- BRDF parameters

Differential radiance from x :

$$\frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{d}{d\pi} \{f_r(\omega_i, \omega_o, \pi) L_i(\omega_i) (n \cdot \omega_i)\} d\sigma(\omega_i)$$

Just move derivative inside integral

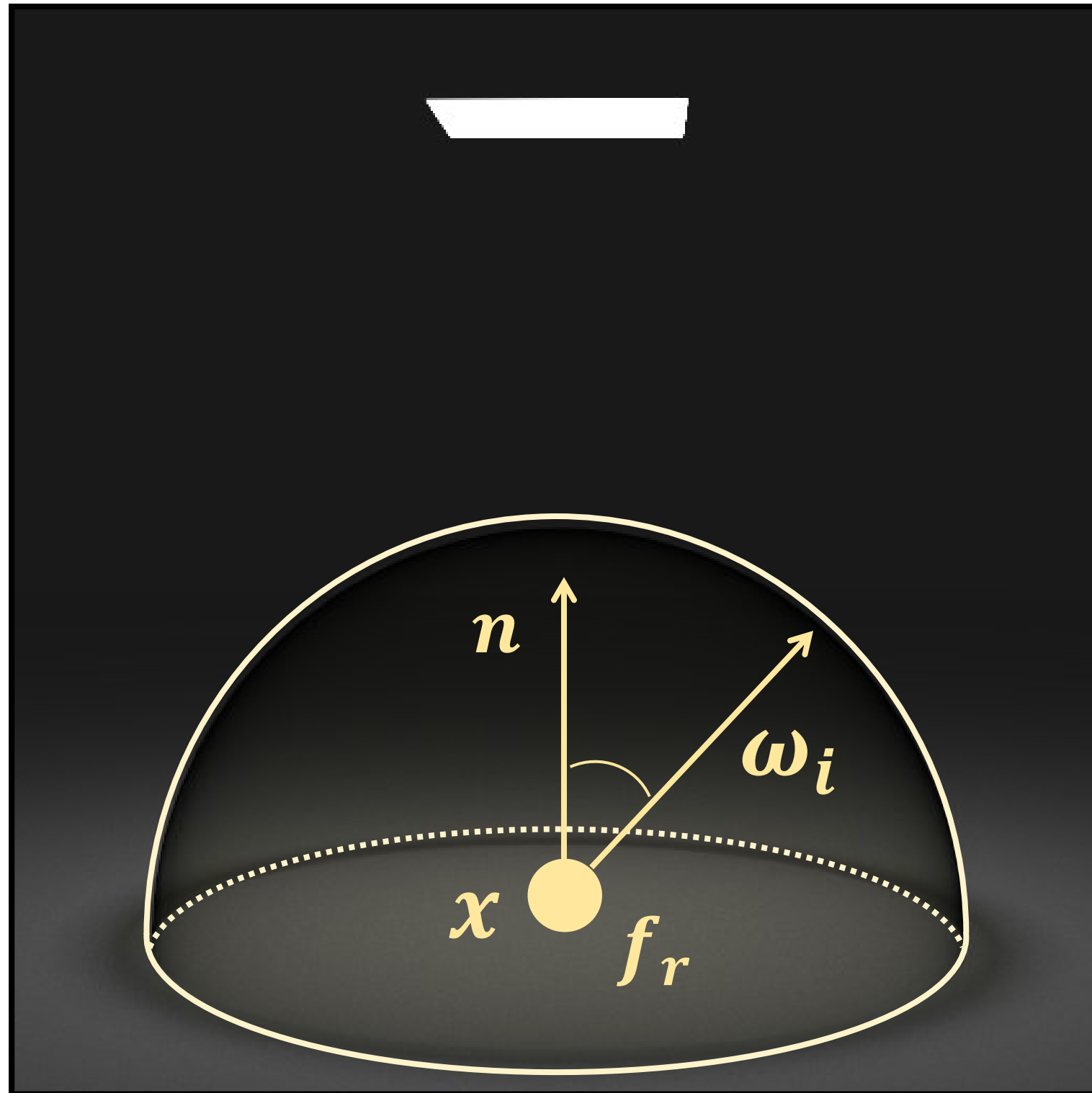
Monte Carlo estimation:

- Sample random directions ω_i^s from PDF $p(\omega_i, \pi)$
- Form estimator

Differentiate entire contribution
[Zeltner et al. 2021]

$$\frac{dI}{d\pi} \approx \sum_s \frac{d}{d\pi} \left\{ \frac{f_r(\omega_i^s, \omega_o, \pi) L_i(\omega_i^s) (n \cdot \omega_i^s)}{p(\omega_i^s, \pi)} \right\}$$

Differential direct illumination: global parameters



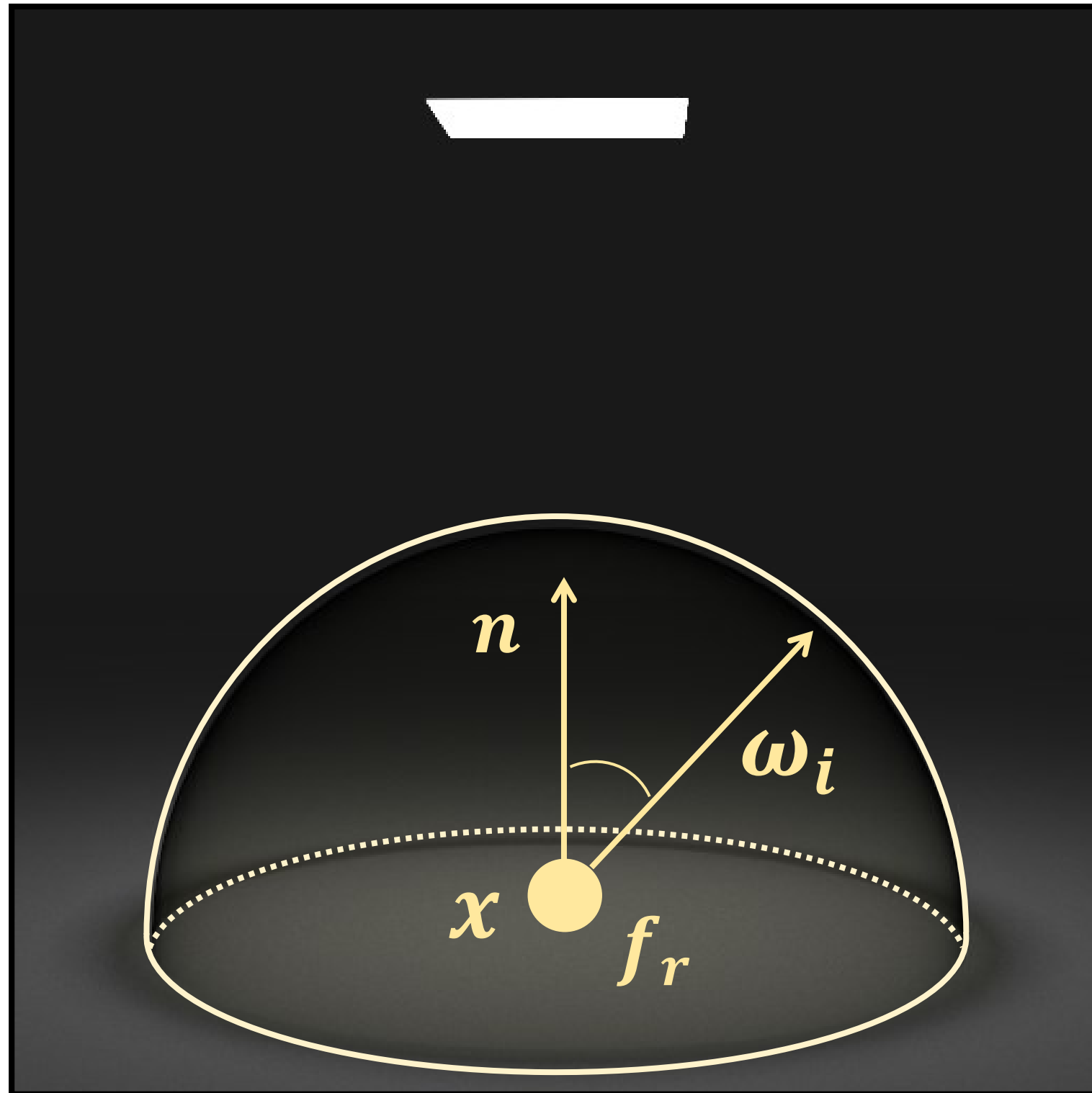
Differential radiance from x :

$$\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) d\sigma(\omega_i)$$

π : global parameters

- shape and pose of different scene elements (camera, sources, objects)

Differential direct illumination: global parameters



Differential radiance from x :

$$\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) d\sigma(\omega_i)$$

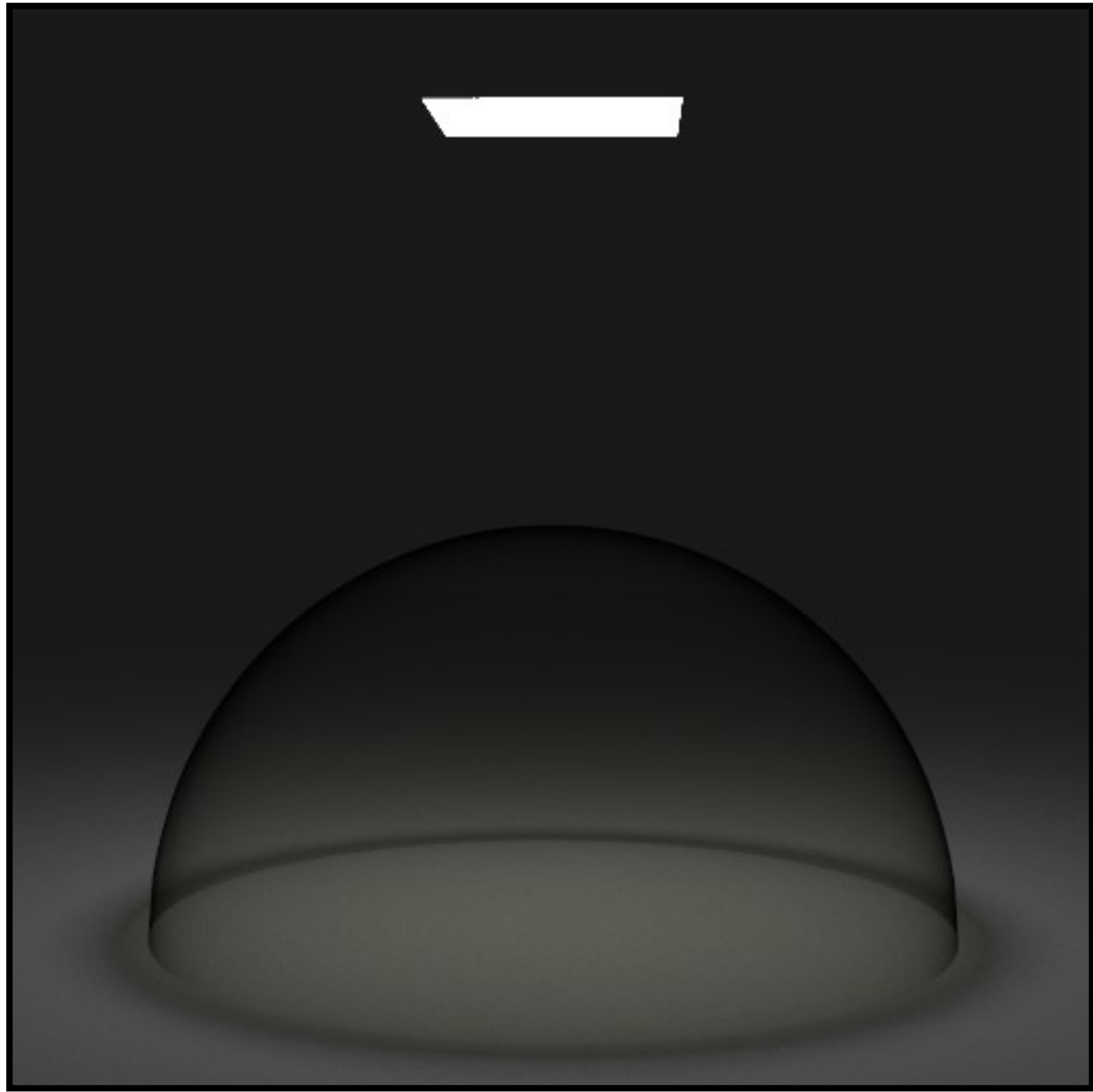
~~$$= \int_{\mathbb{H}^2} \frac{d}{d\pi} \{f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i)\} d\sigma(\omega_i)$$~~

Need to use full Reynolds transport theorem

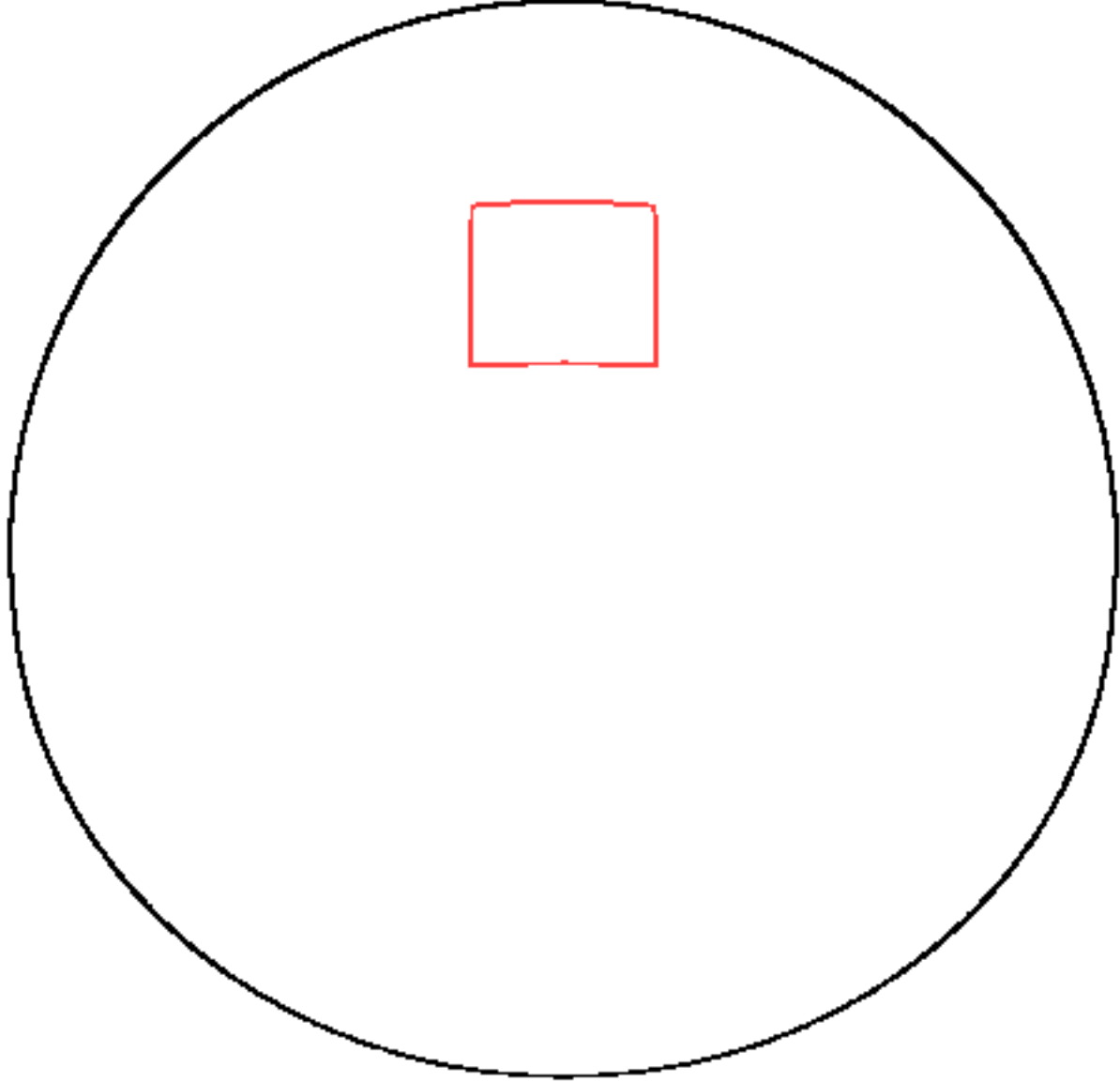
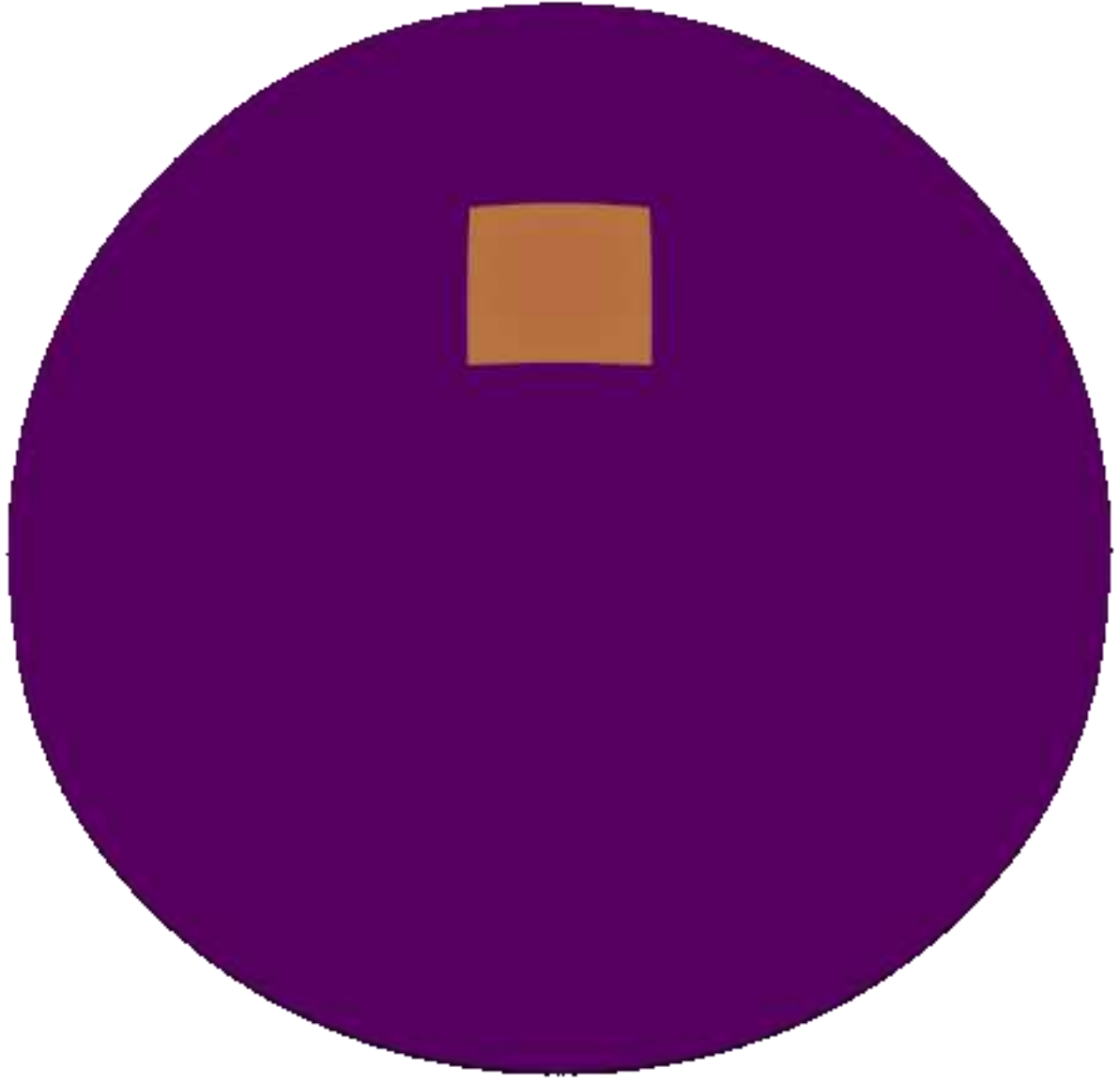
π : global parameters

- shape and pose of different scene elements (camera, sources, objects)

Discontinuities in the integrand



Low  High



π : size of the emitter

$$I = \int_{\mathbb{H}^2} \underbrace{f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i)}_{f(\omega_i)} d\sigma(\omega_i)$$

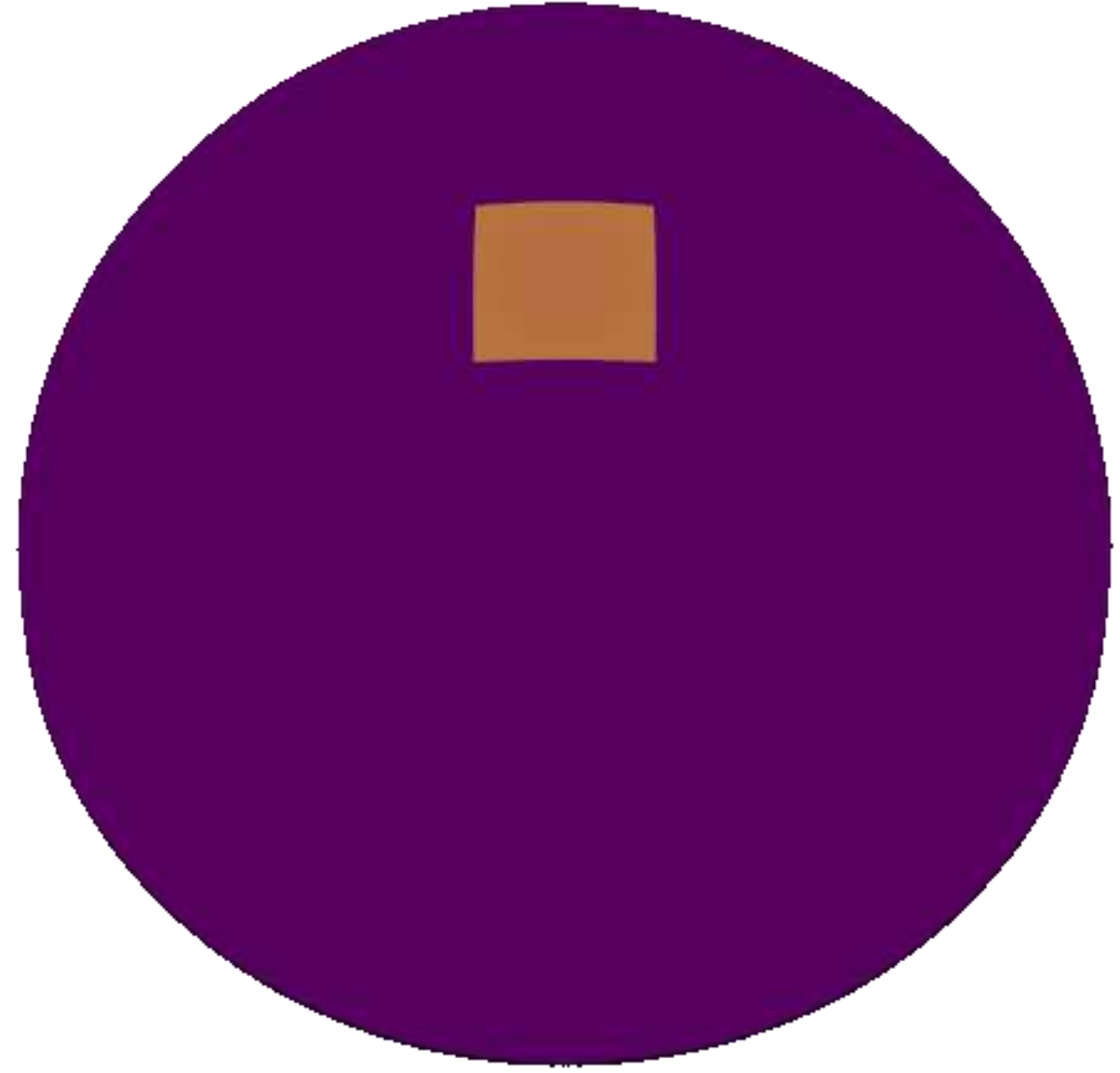
Integrand
 $f(\omega_i)$

Discontinuous points
(π -dependent)

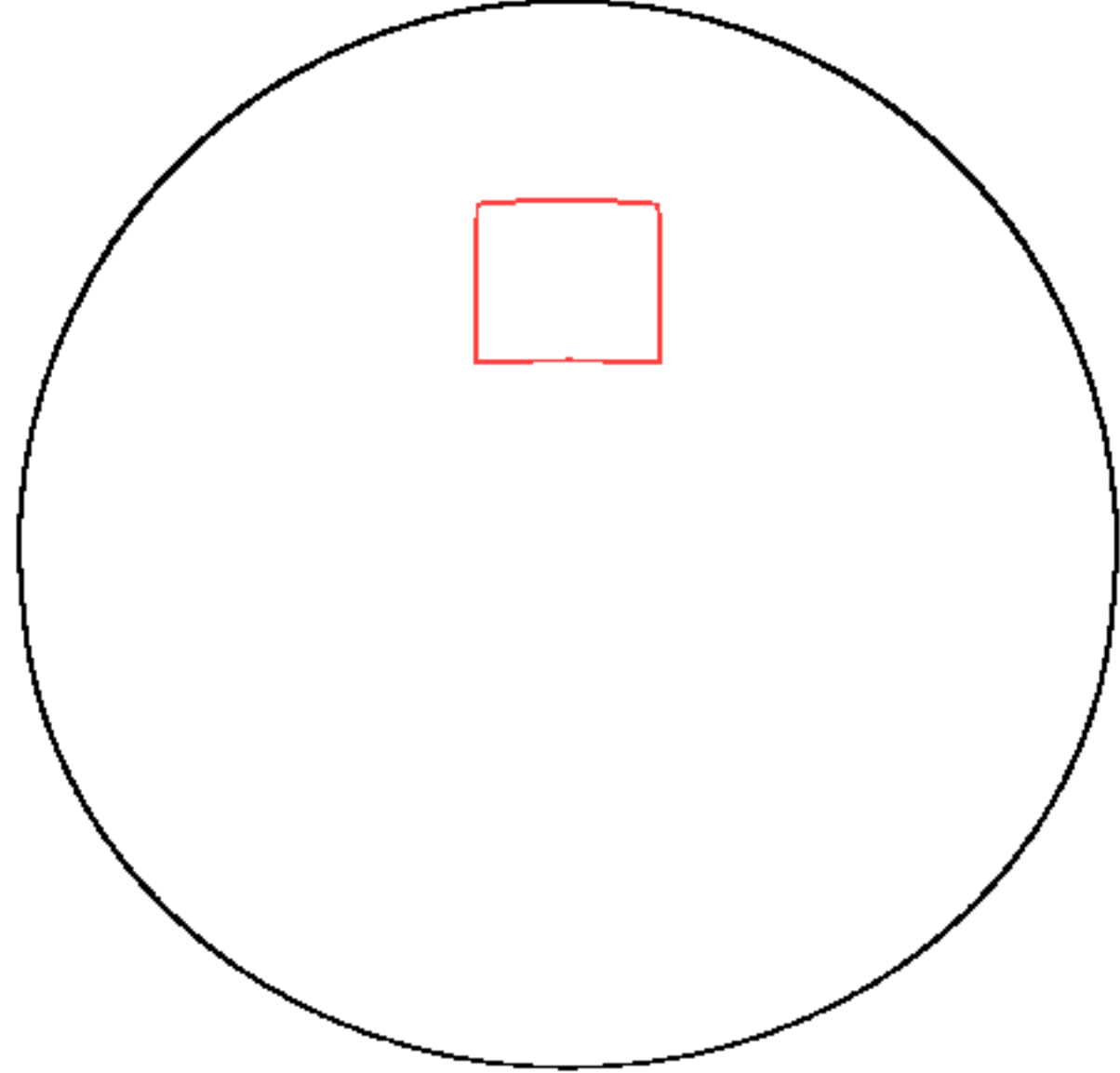
Applying the Reynolds transport theorem

$$I = \int_{\mathbb{H}^2} f(\omega_i, \omega_o) d\sigma(\omega_i)$$

Low  High



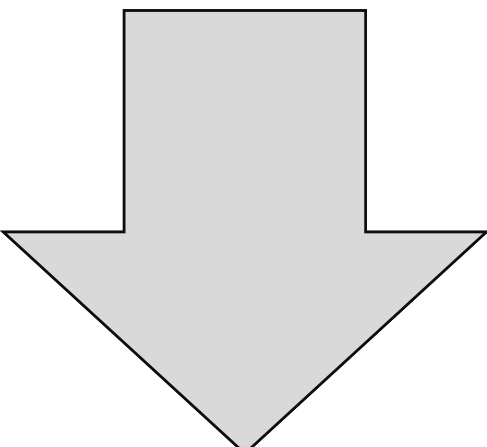
Integrand
 $f(\omega_i)$



Discontinuous points
(π -dependent)

Applying the Reynolds transport theorem

$$I = \int_{\mathbb{H}^2} f(\omega_i, \omega_o) d\sigma(\omega_i)$$

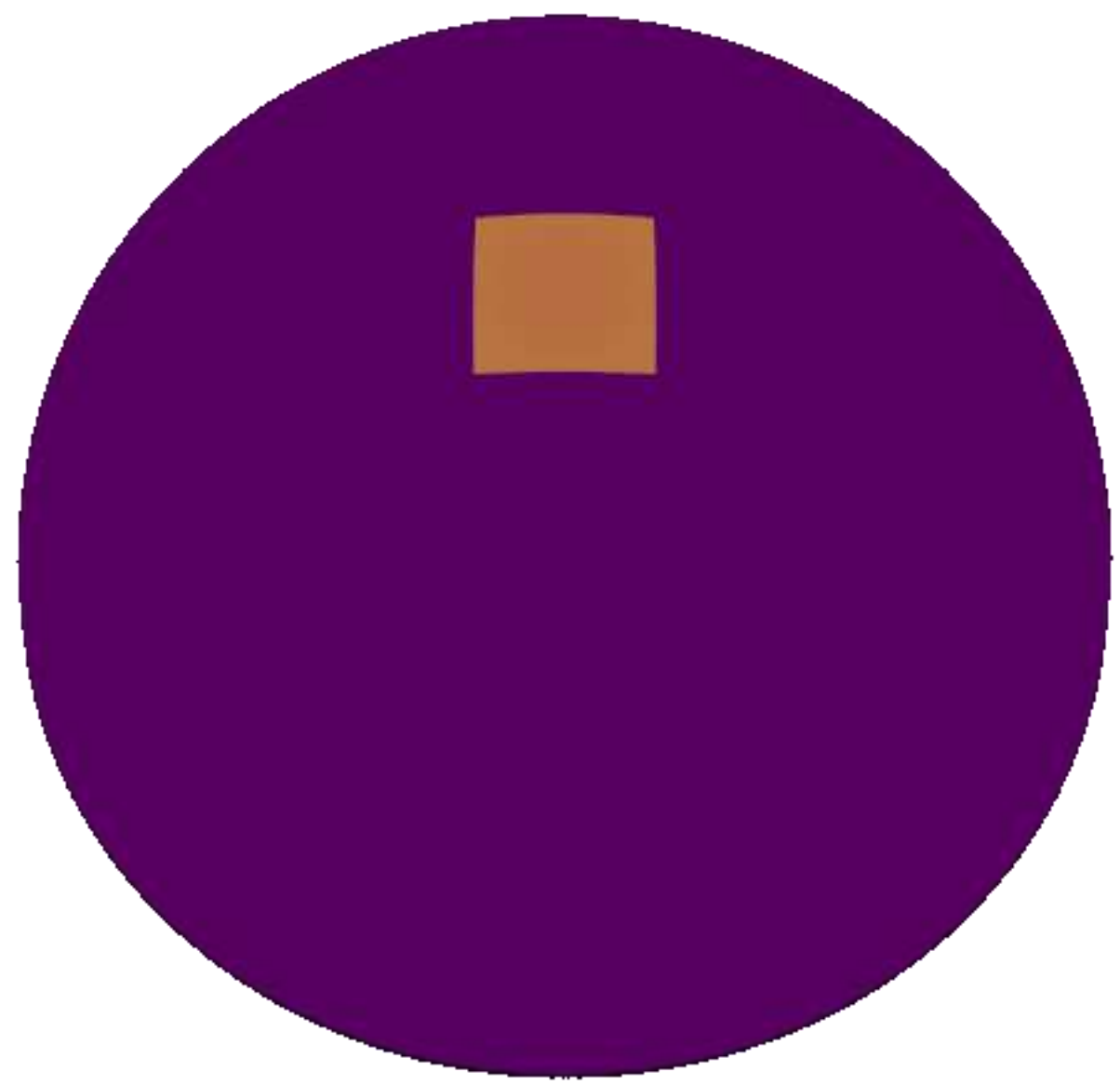


$$\frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{df}{d\pi} d\sigma + \int_{\partial\mathbb{H}^2} g dl$$

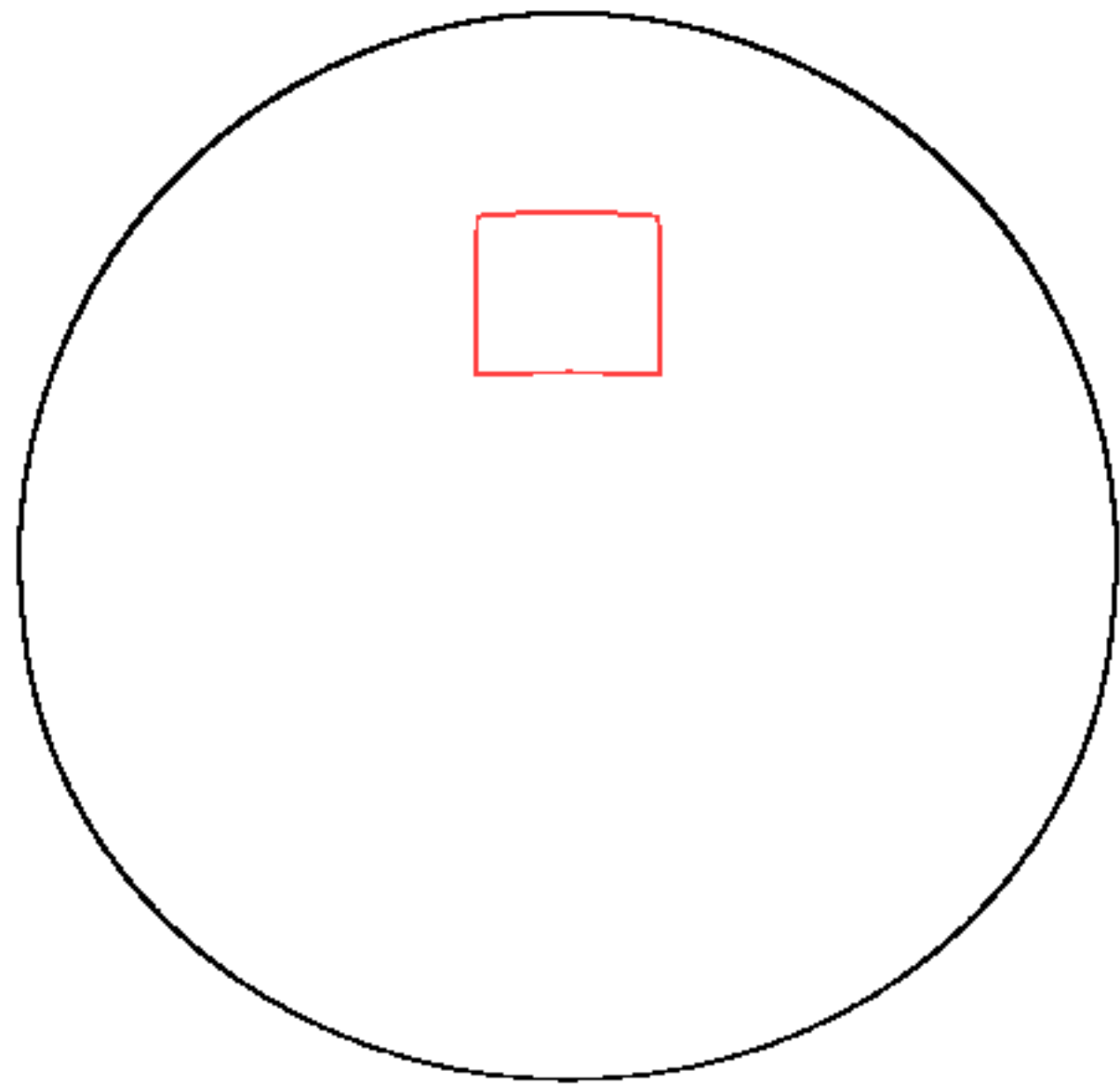
Interior integral
(same as for local parameters)

Boundary integral

Low  High



Integrand
 $f(\omega_i)$

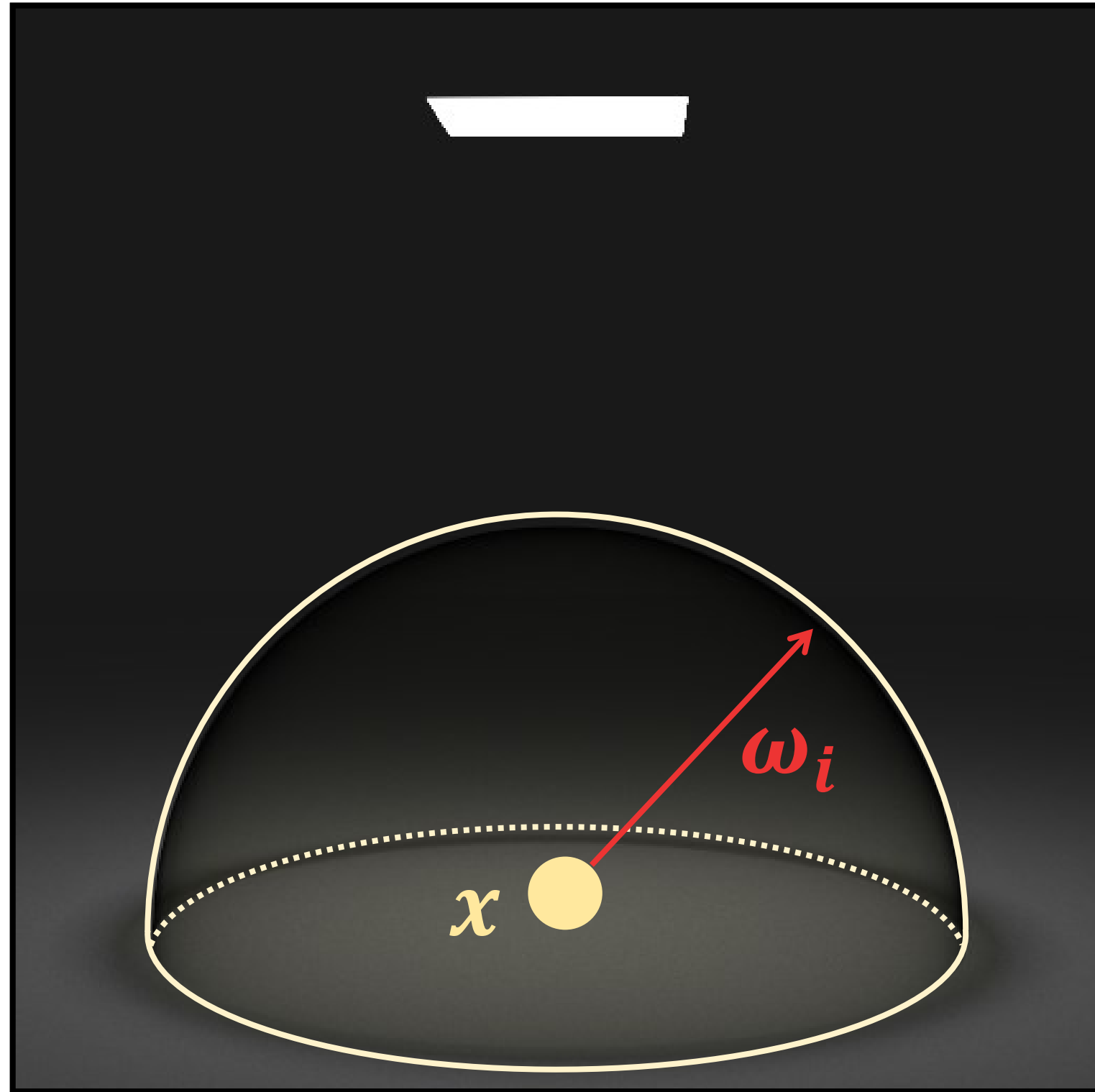


Discontinuous points
(π -dependent)

[Ramamoorthi et al. 2007, Li et al. 2019]

Reparameterizing the direct illumination integral

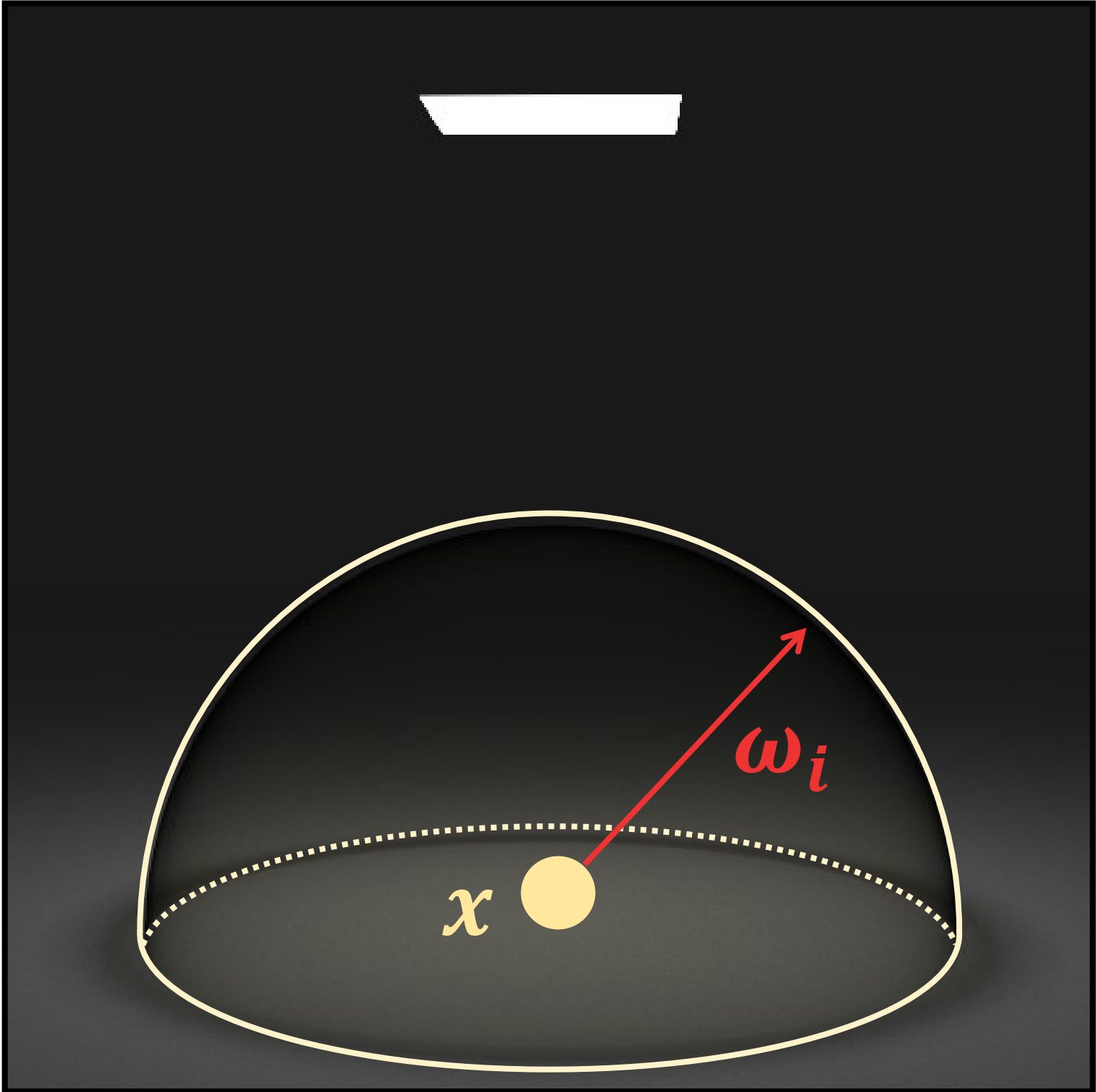
Hemispherical integral



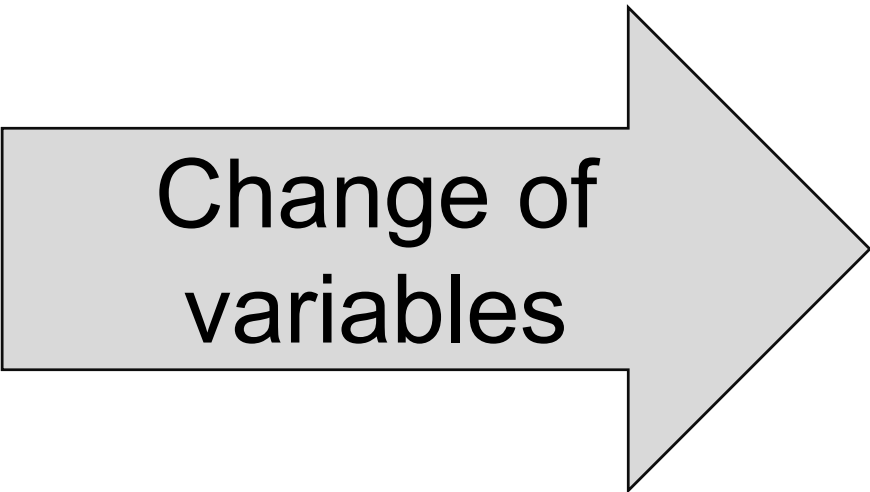
$$I = \int_{\mathbb{H}^2} f(\omega_i) d\sigma(\omega_i)$$

Reparameterizing the direct illumination integral

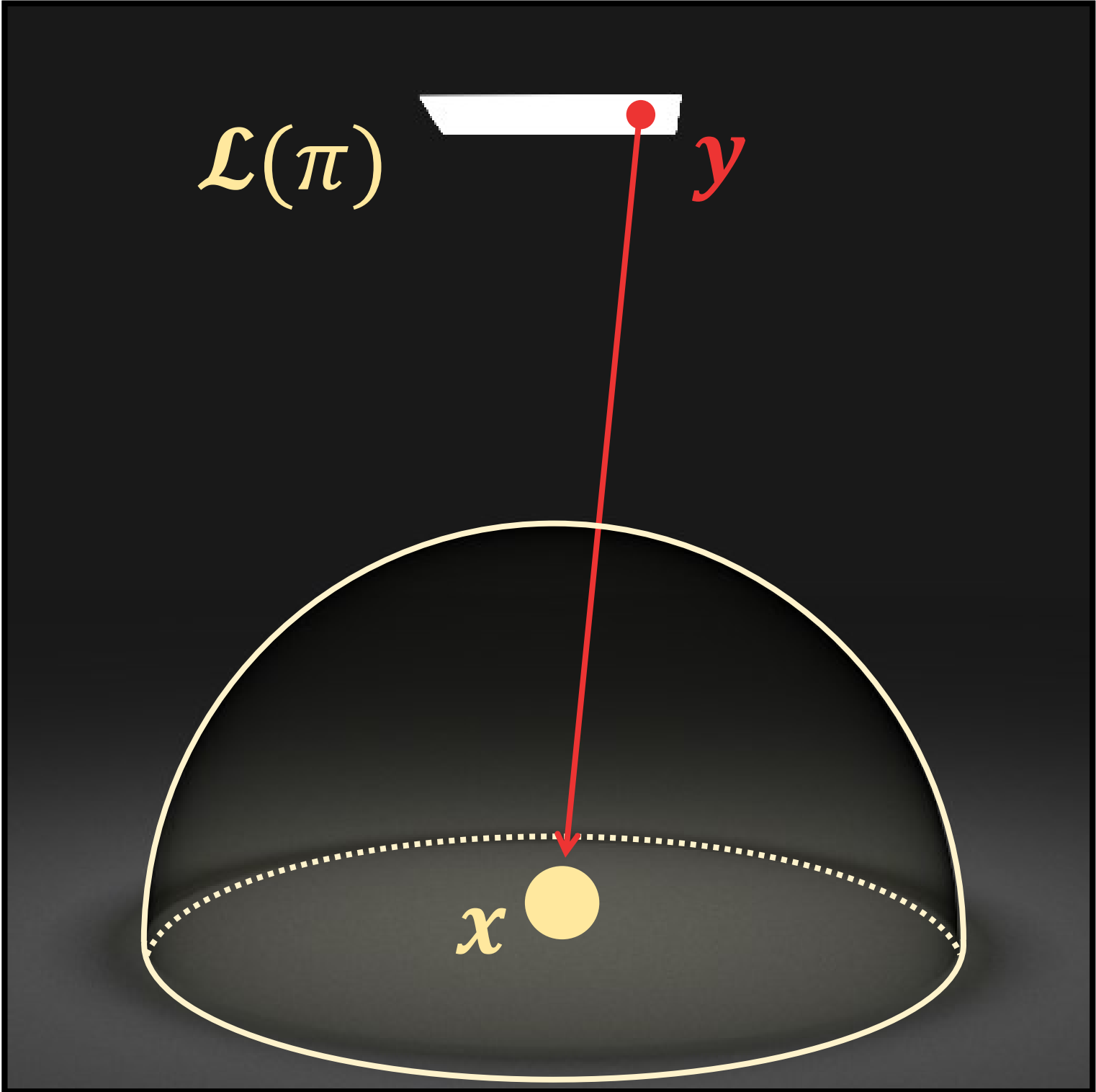
Hemispherical integral



$$I = \int_{\mathbb{H}^2} f(\boldsymbol{\omega}_i) d\sigma(\boldsymbol{\omega}_i)$$



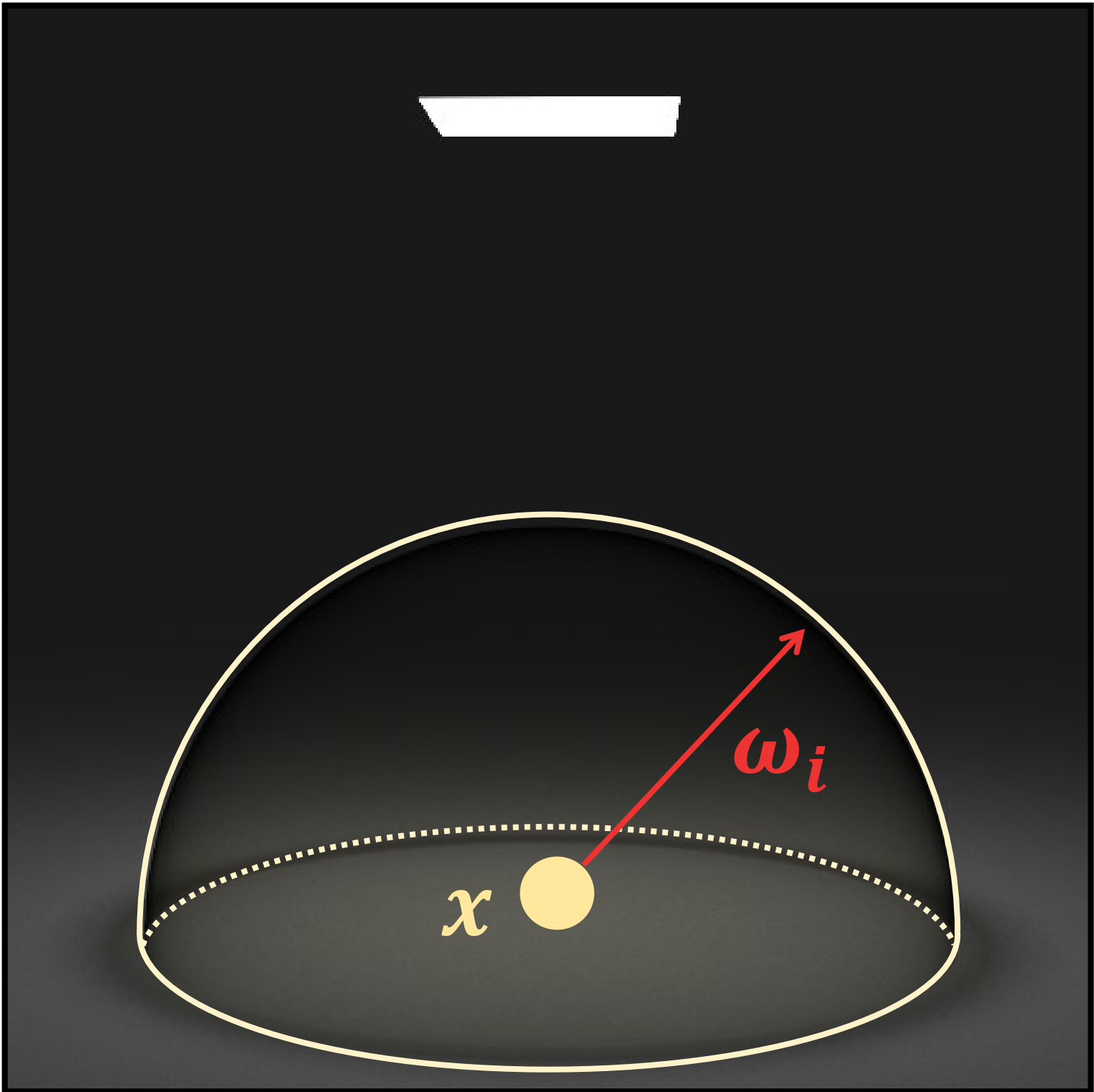
Surface integral



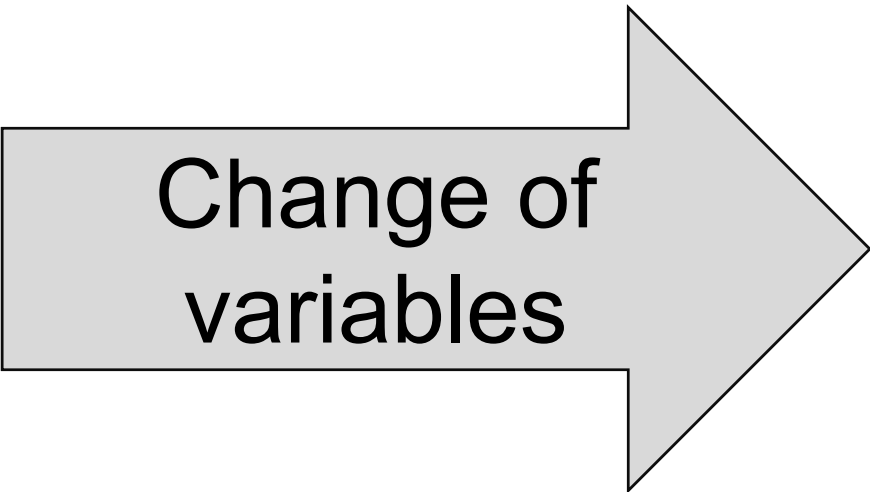
$$I = \int_{\mathcal{L}(\pi)} f(\mathbf{y} \rightarrow \mathbf{x}) G(\mathbf{x}, \mathbf{y}) dA(\mathbf{y})$$

Reparameterizing the direct illumination integral

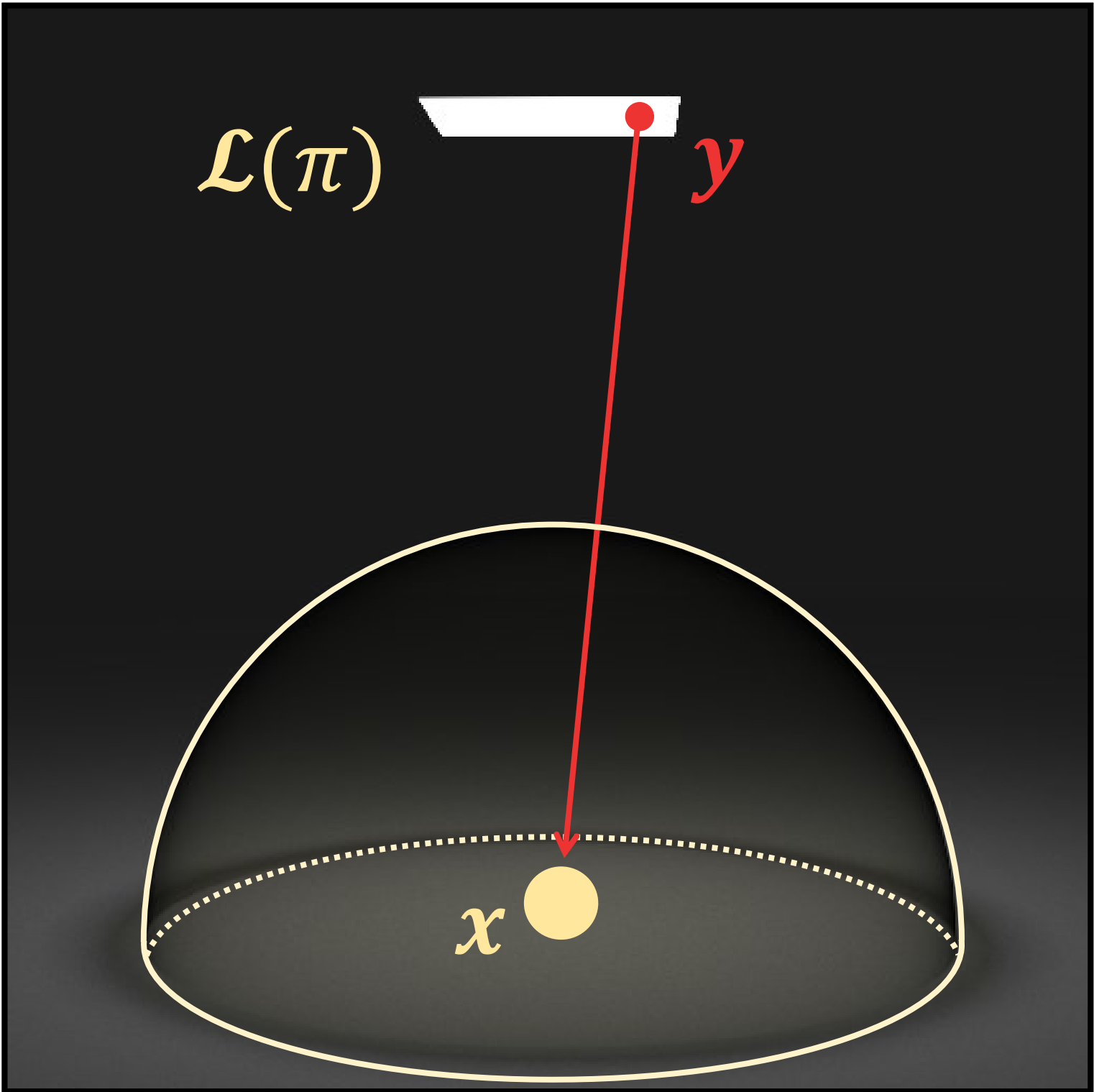
Hemispherical integral



$$I = \int_{\mathbb{H}^2} f(\boldsymbol{\omega}_i) d\sigma(\boldsymbol{\omega}_i)$$



Surface integral

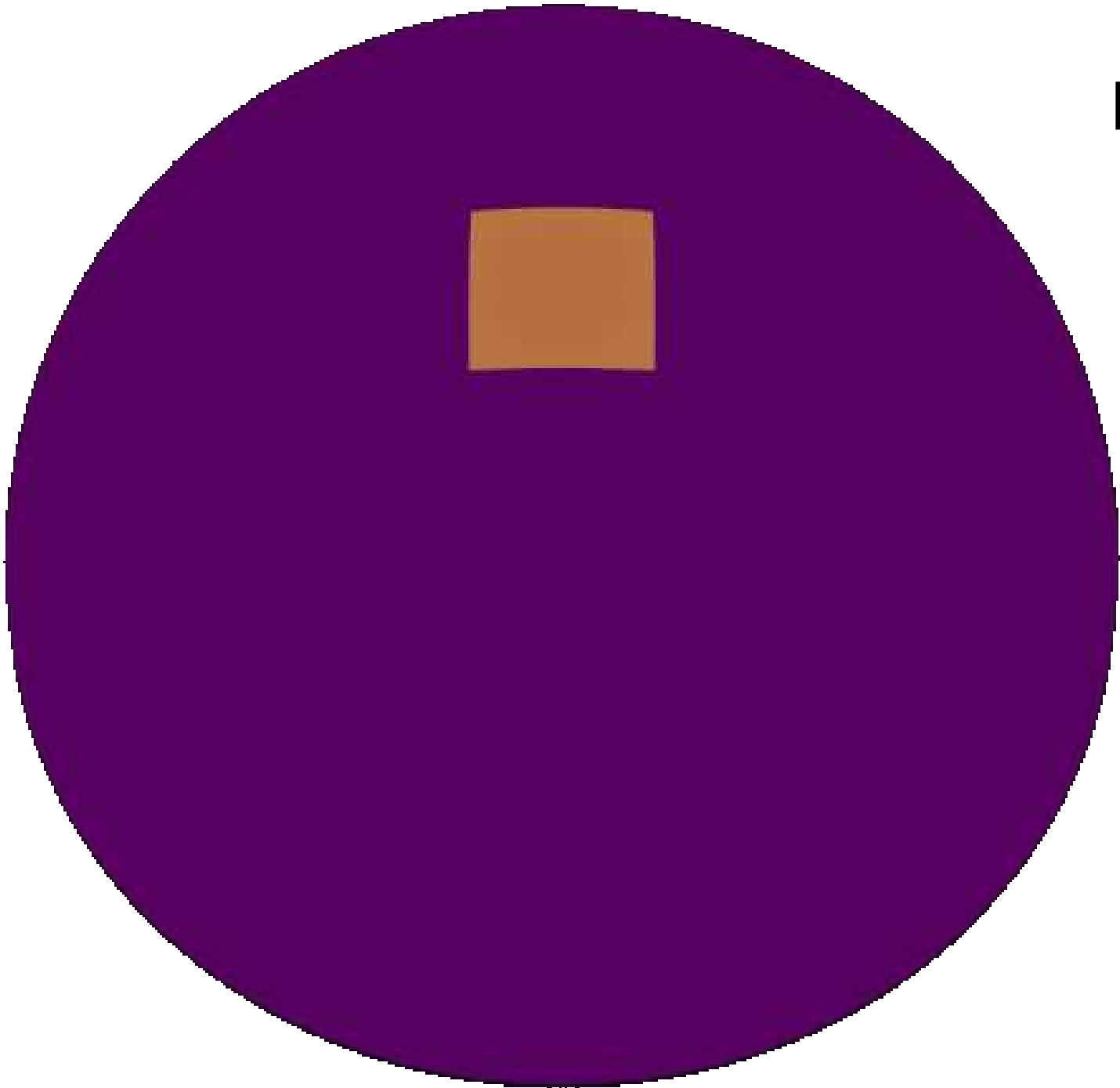


$$I = \int_{\mathcal{L}(\pi)} f(\mathbf{y} \rightarrow \mathbf{x}) G(\mathbf{x}, \mathbf{y}) dA(\mathbf{y})$$

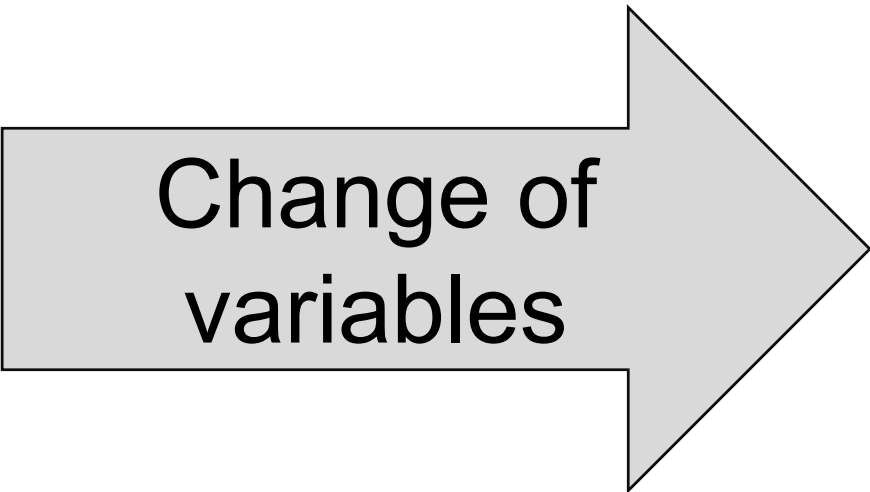
Includes visibility, fall-off, and foreshortening terms

Reparameterizing the direct illumination integral

Hemispherical integral



Low  High



Surface integral

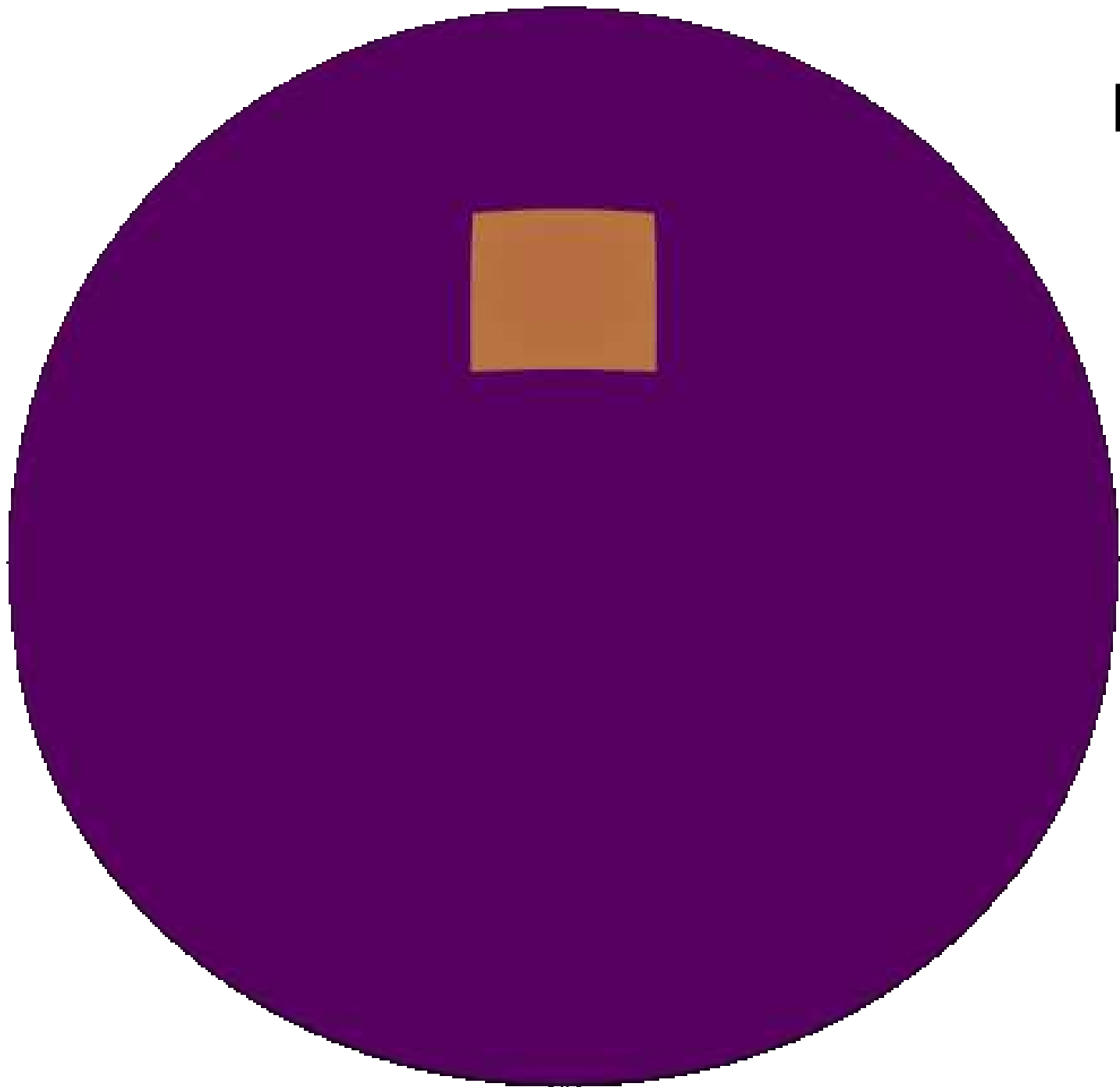


$$I = \int_{\mathbb{H}^2} f(\omega_i) d\sigma(\omega_i)$$

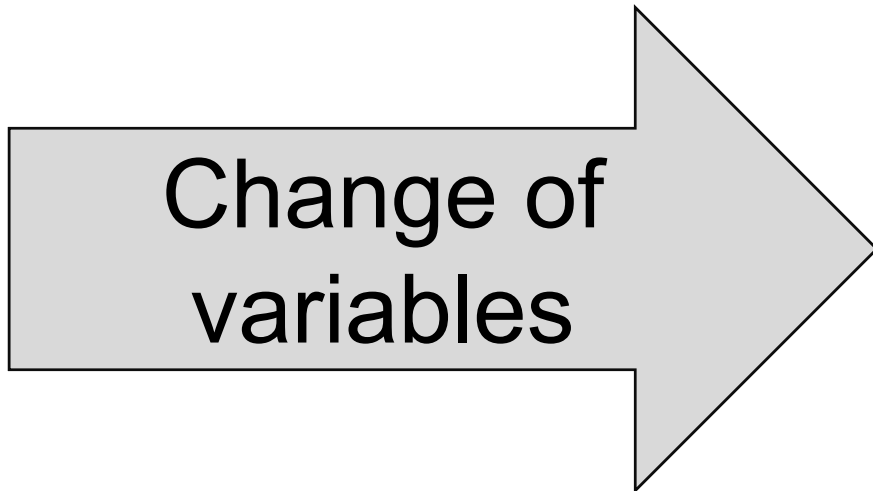
$$I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) G(x, y) dA(y)$$

Reparameterizing the direct illumination integral

Hemispherical integral



Low  High



Surface integral



discontinuous

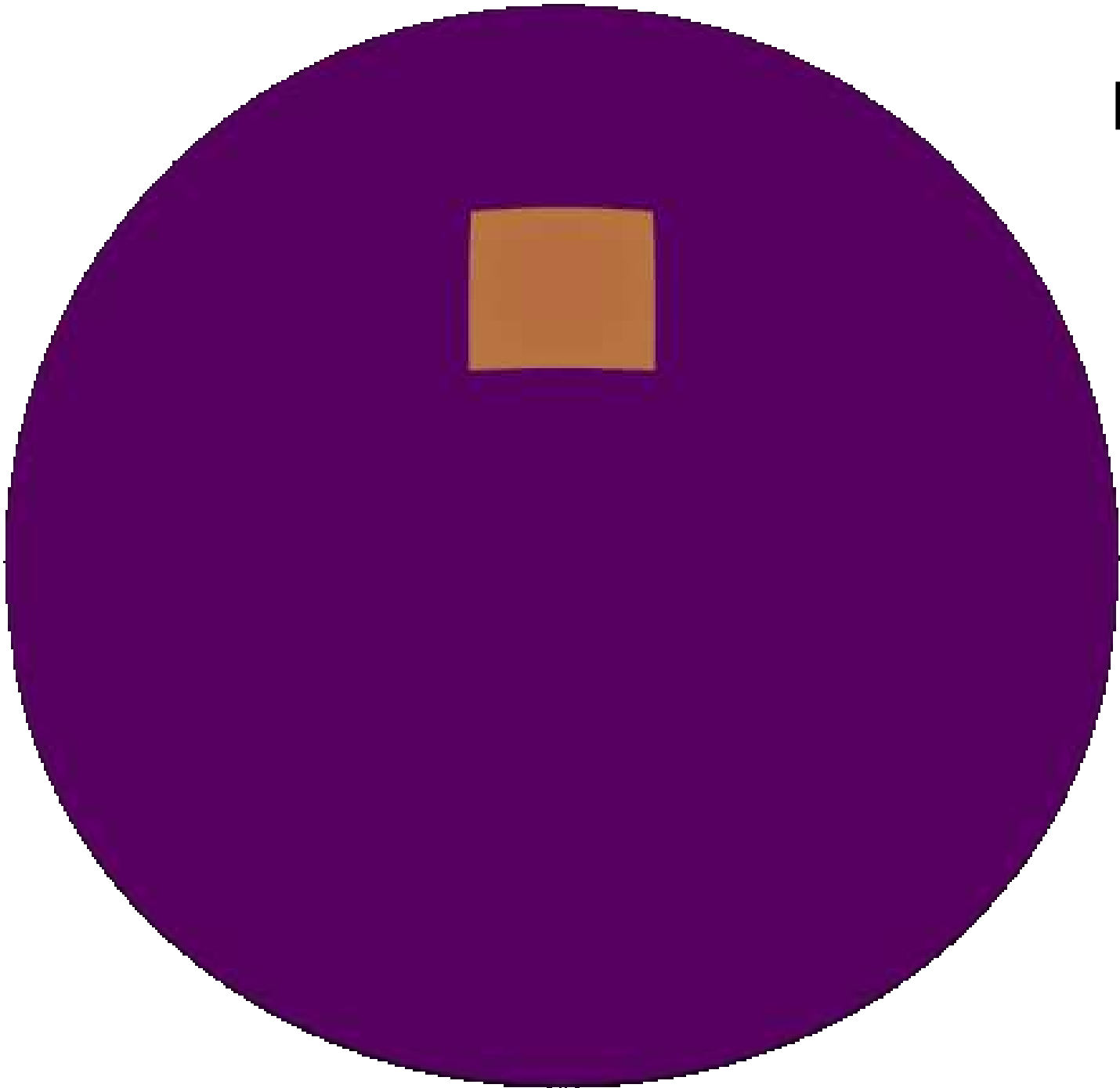
$$I = \int_{\mathbb{H}^2} f(\omega_i) d\sigma(\omega_i)$$

continuous

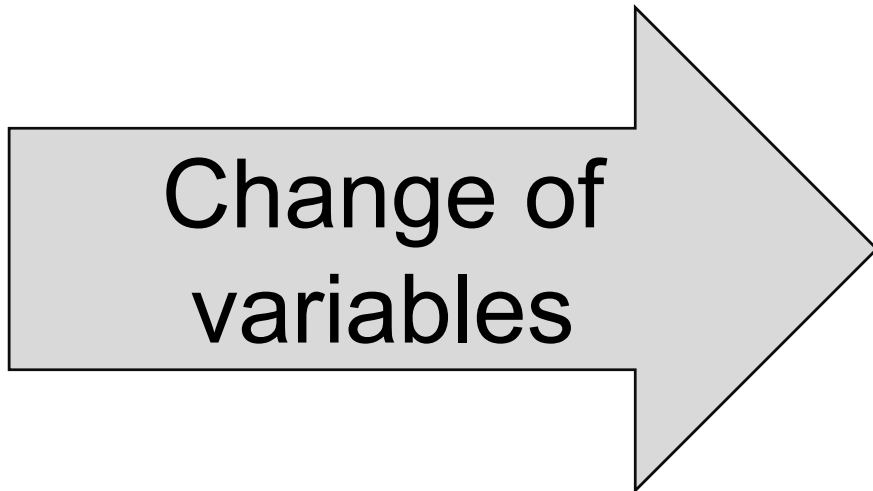
$$I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) G(x, y) dA(y)$$

Reparameterizing the direct illumination integral

Hemispherical integral



Low  High



Surface integral



discontinuous

$$I = \int_{\mathbb{H}^2} f(\omega_i) d\sigma(\omega_i)$$

constant domain

continuous

$$I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) G(x, y) dA(y)$$

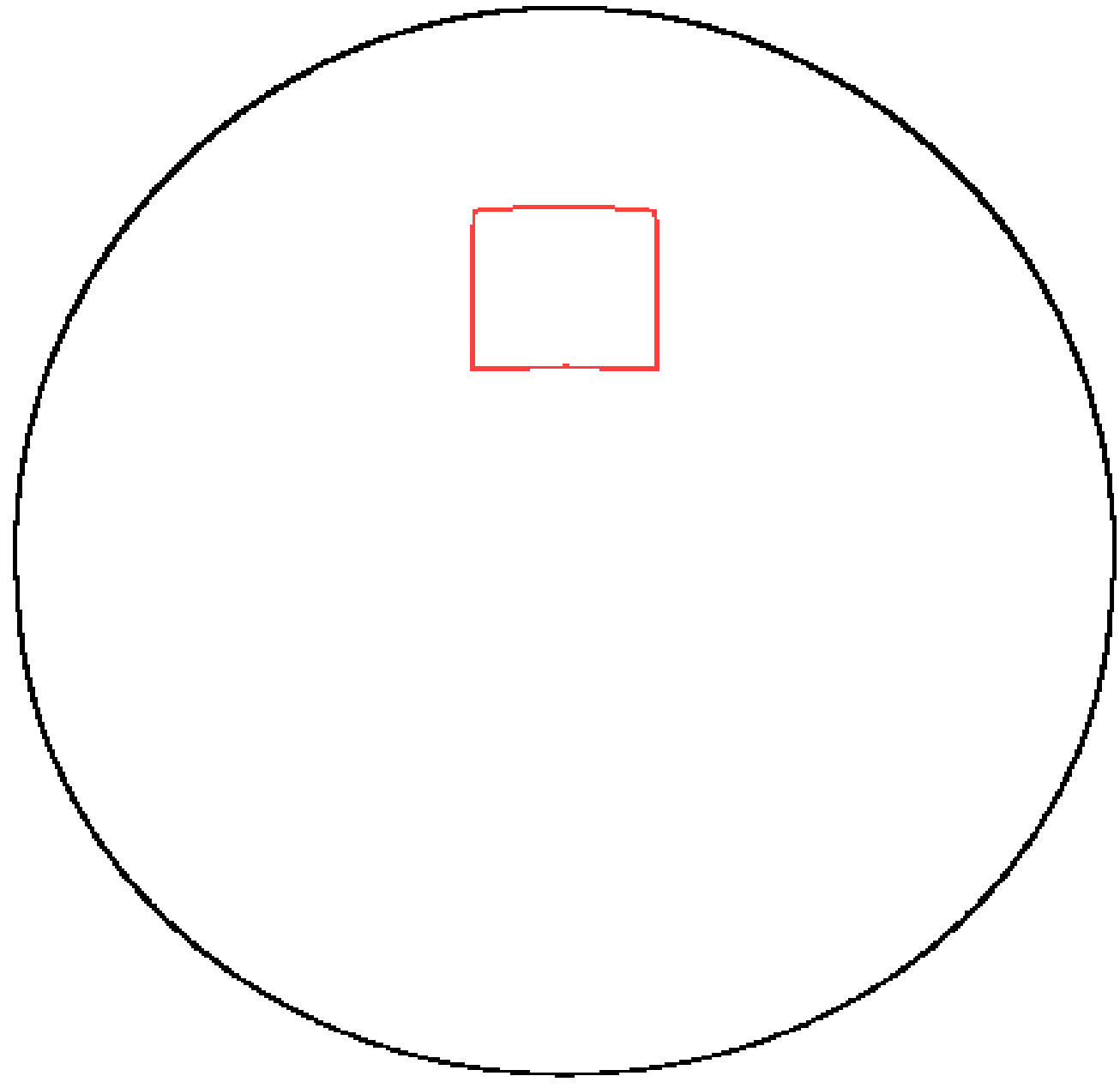
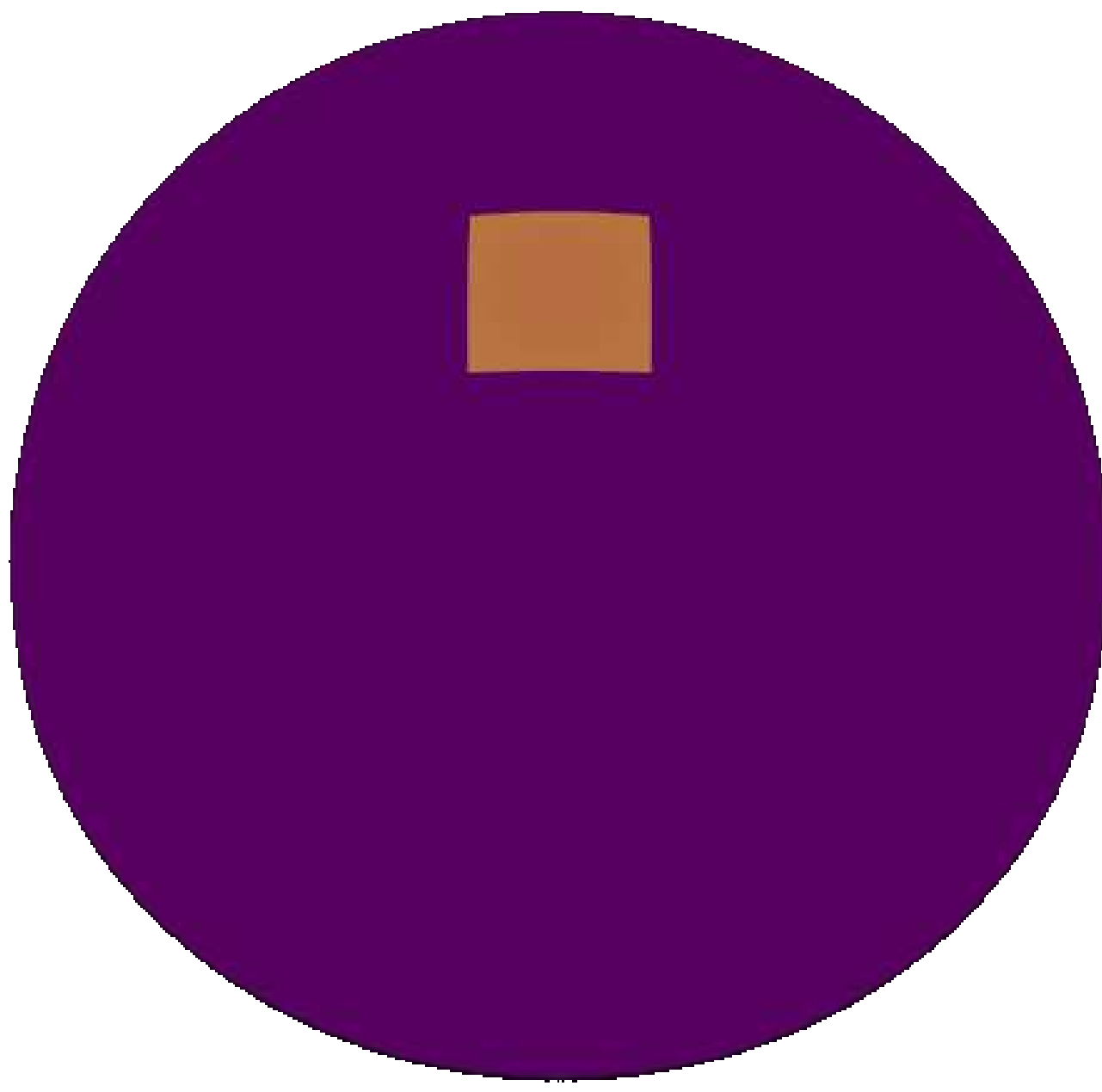
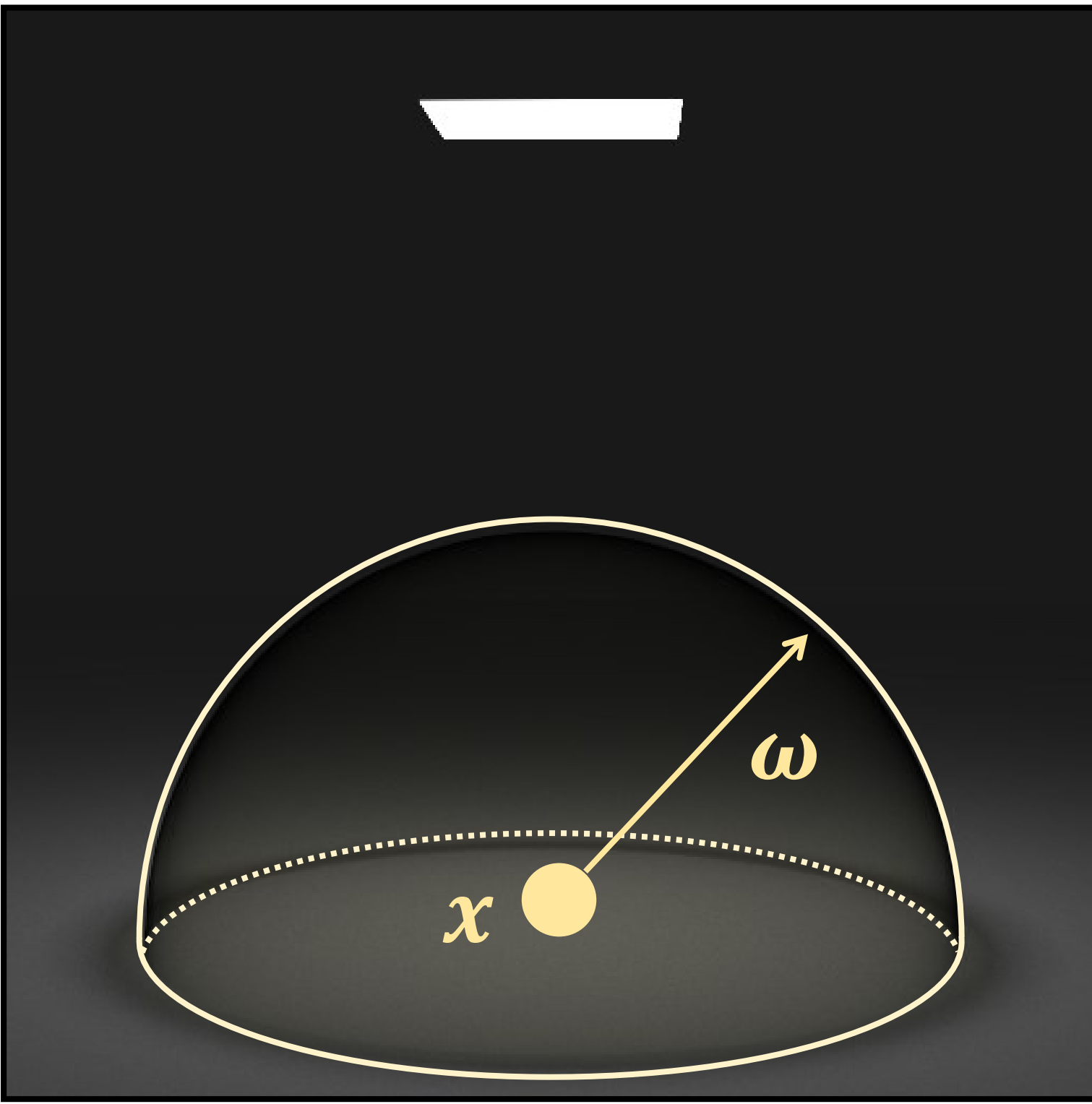
evolving domain

Differentiating the hemispherical integral

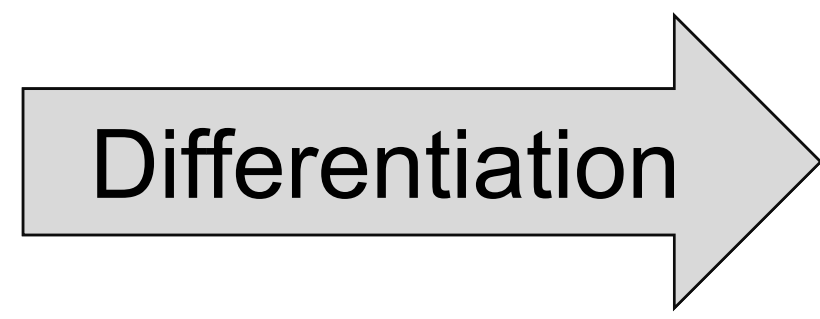
π : size of the emitter



Discontinuities of f



$$I = \int_{\mathbb{H}^2} f(\omega_i) d\sigma(\omega_i)$$



Reynolds transport theorem

$$\frac{dI}{d\pi} = \int_{\mathbb{H}^2} \frac{d(f)}{d\pi} d\sigma + \int_{\partial\mathbb{H}^2} g dl$$

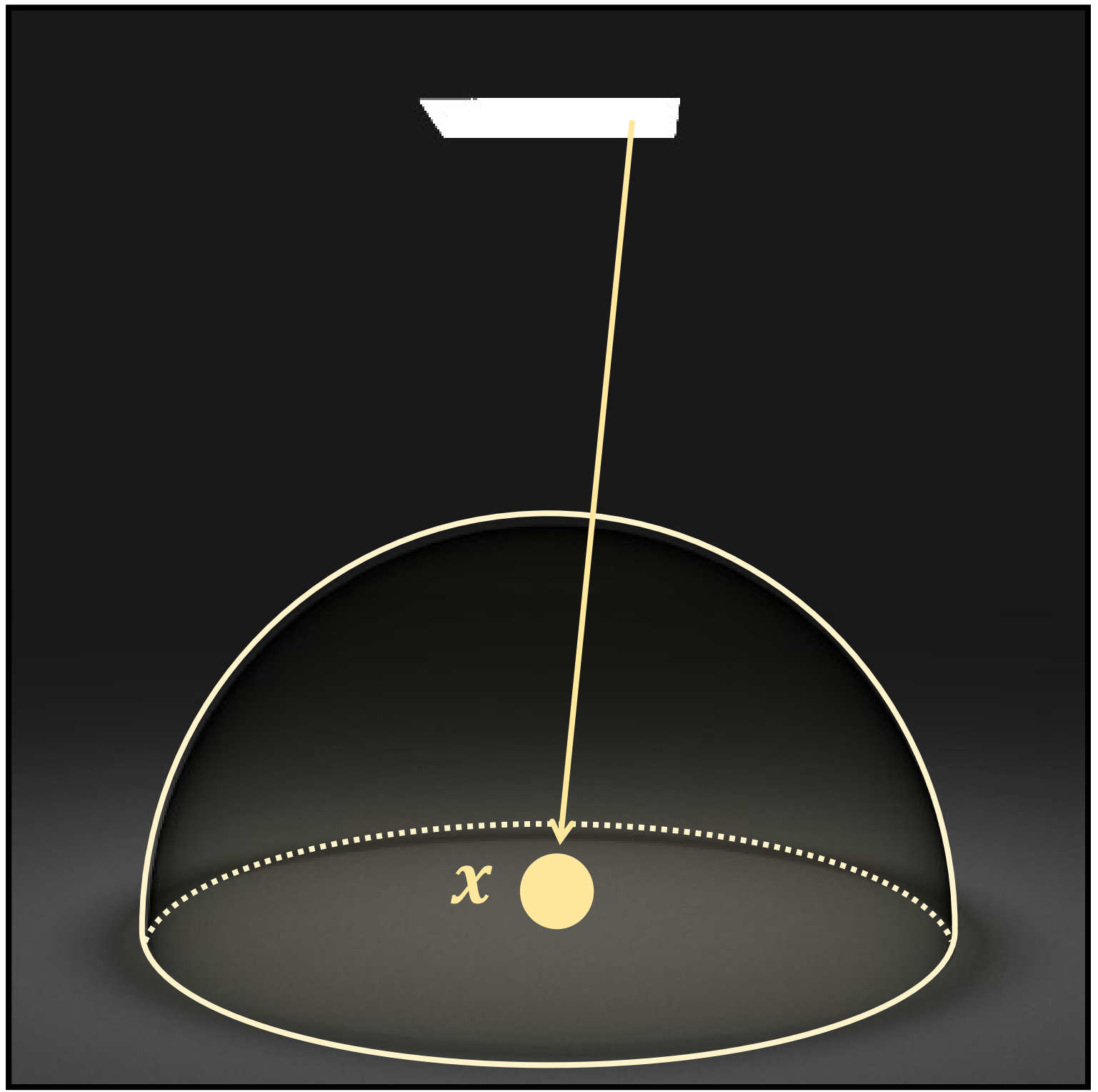
Interior Boundary

Differentiating the area integral

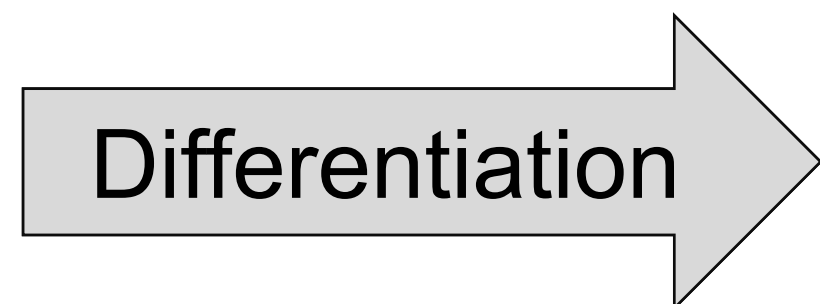
π : size of the emitter



Boundary of $\mathcal{L}(\pi)$



$$I = \int_{\mathcal{L}(\pi)} f(y \rightarrow x) G(x, y) dA(y)$$



Reynolds transport theorem

$$\frac{dI}{d\pi} = \int_{\mathcal{L}(\pi)} \frac{d(fG)}{d\pi} dA + \int_{\partial\mathcal{L}(\pi)} g dl$$

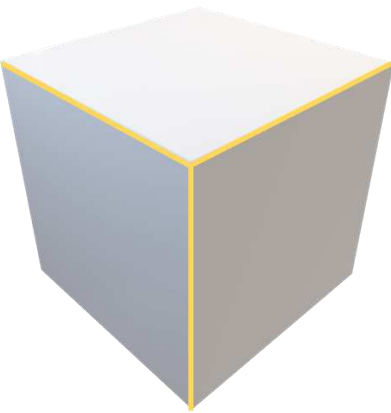
Interior
Boundary

Sources of discontinuities

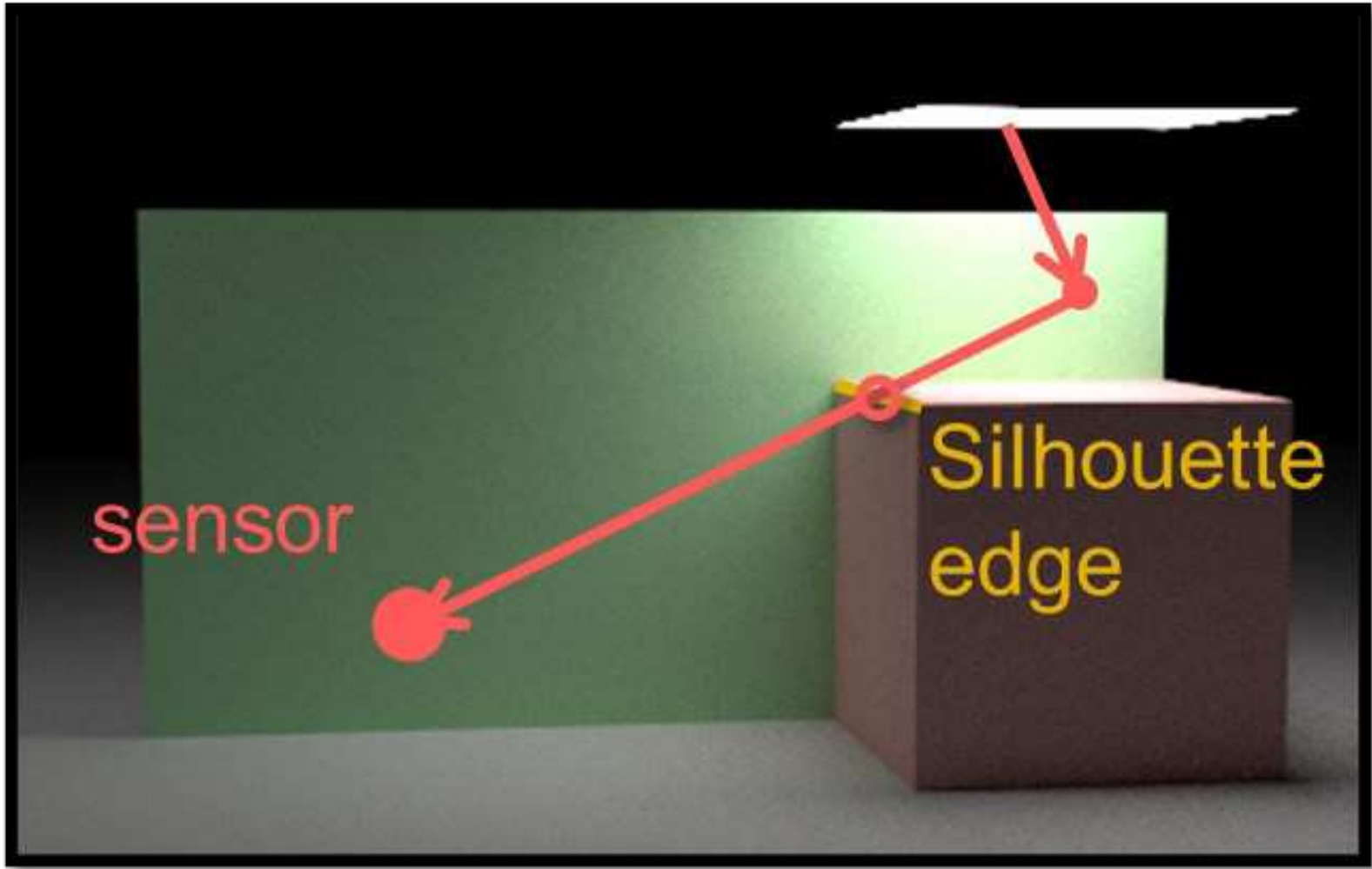
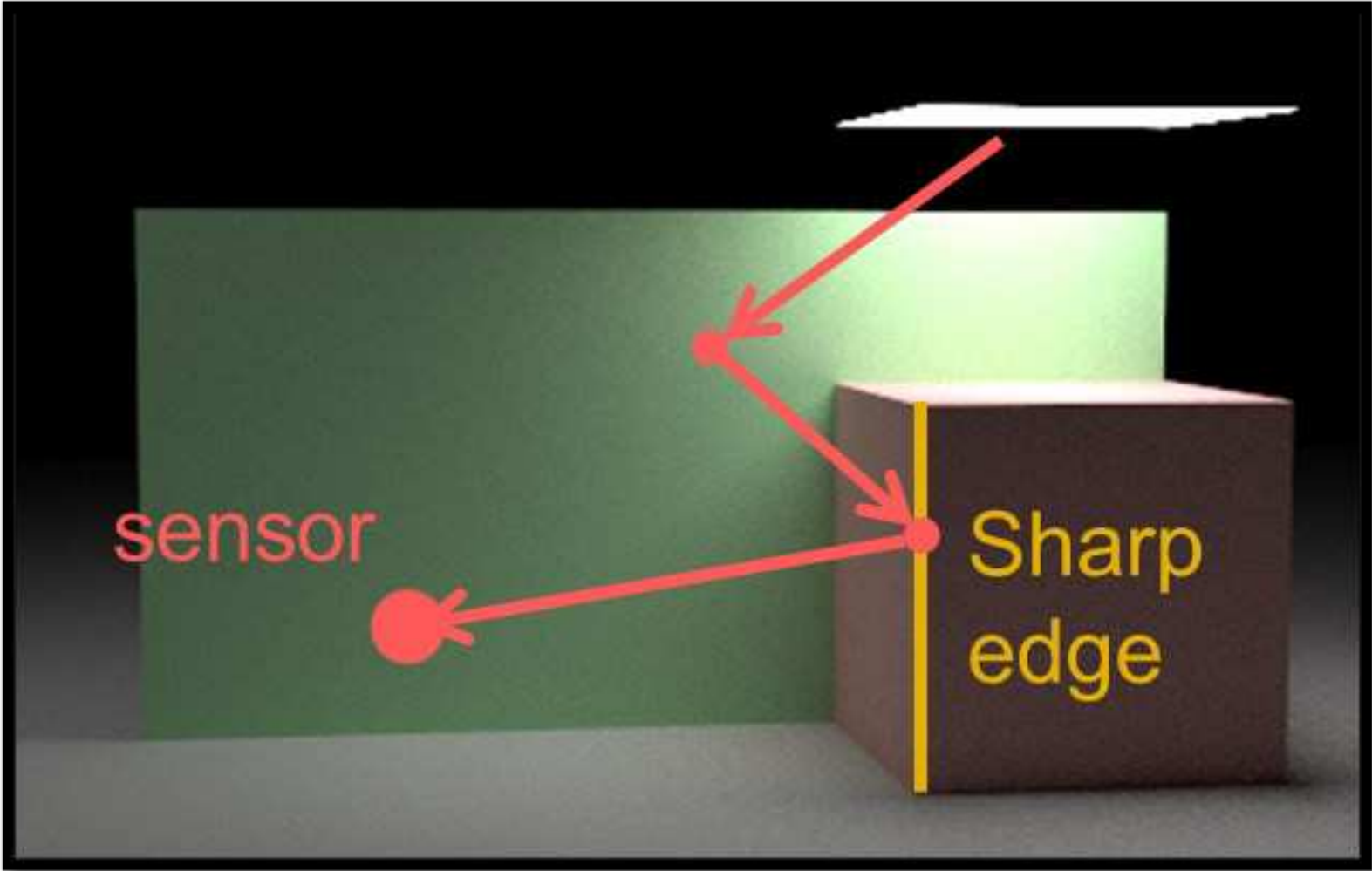
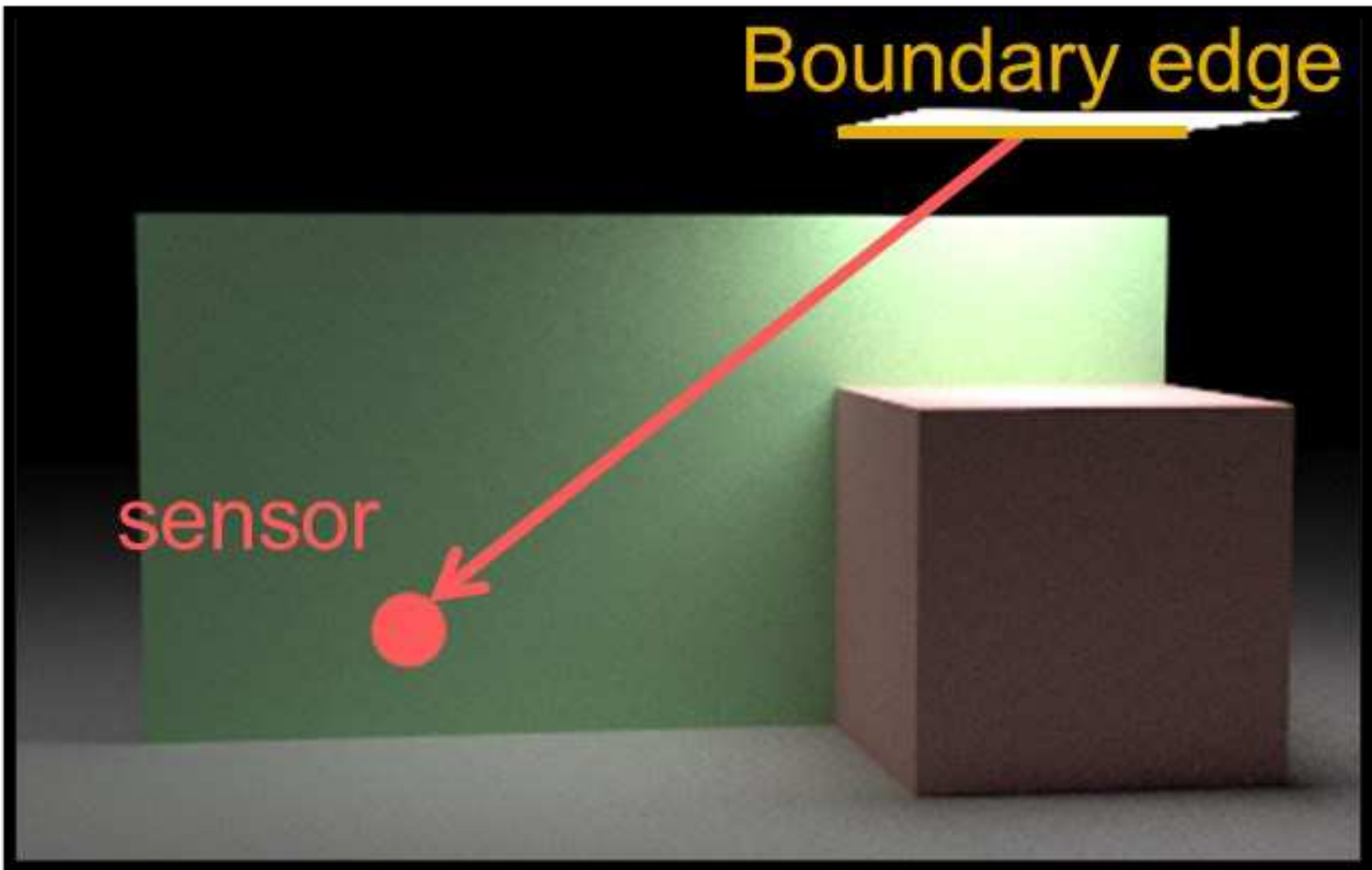
Boundary edge



Sharp edge



Silhouette edge

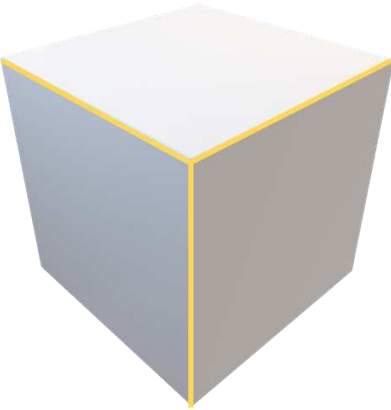


Sources of discontinuities

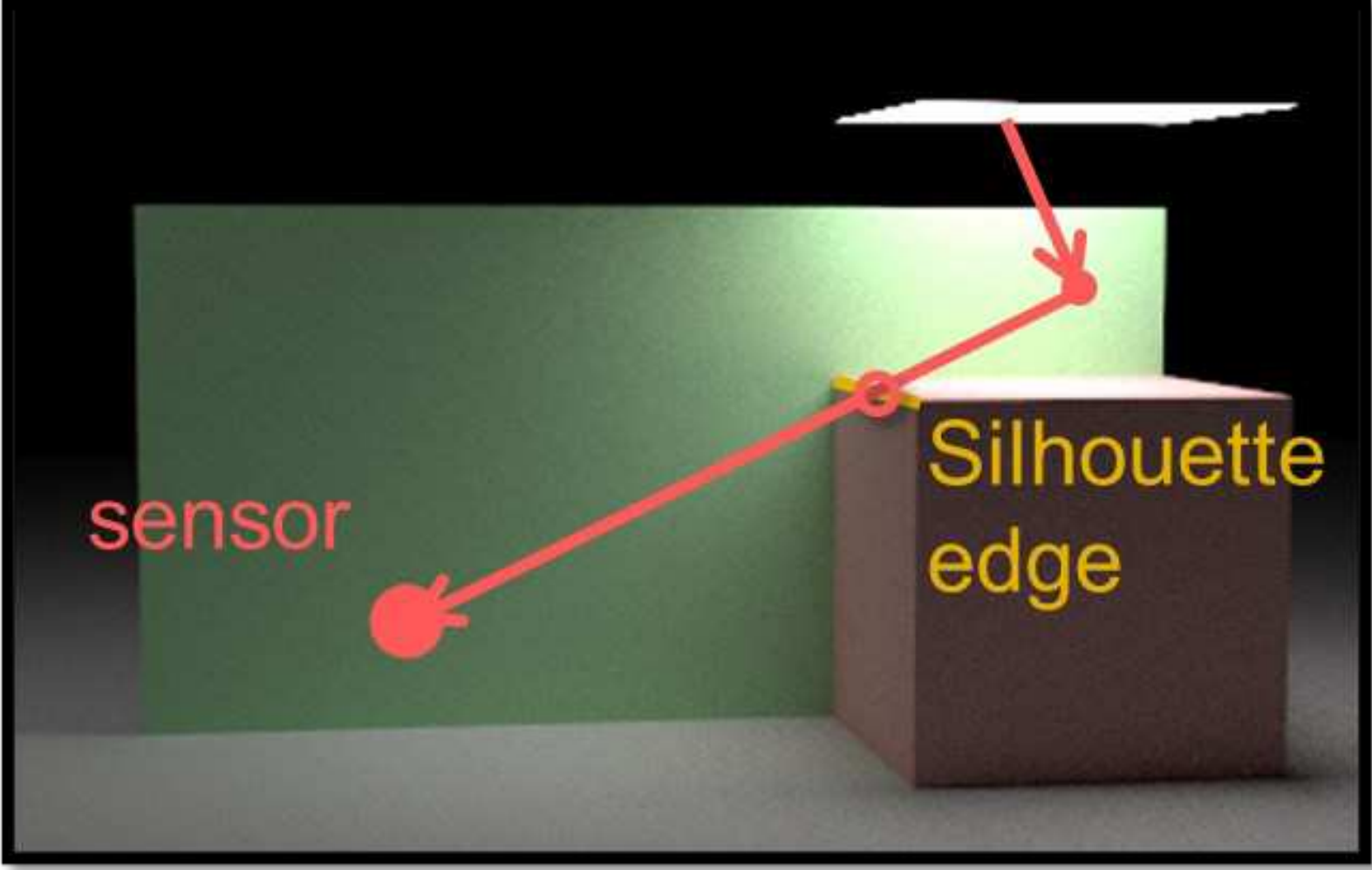
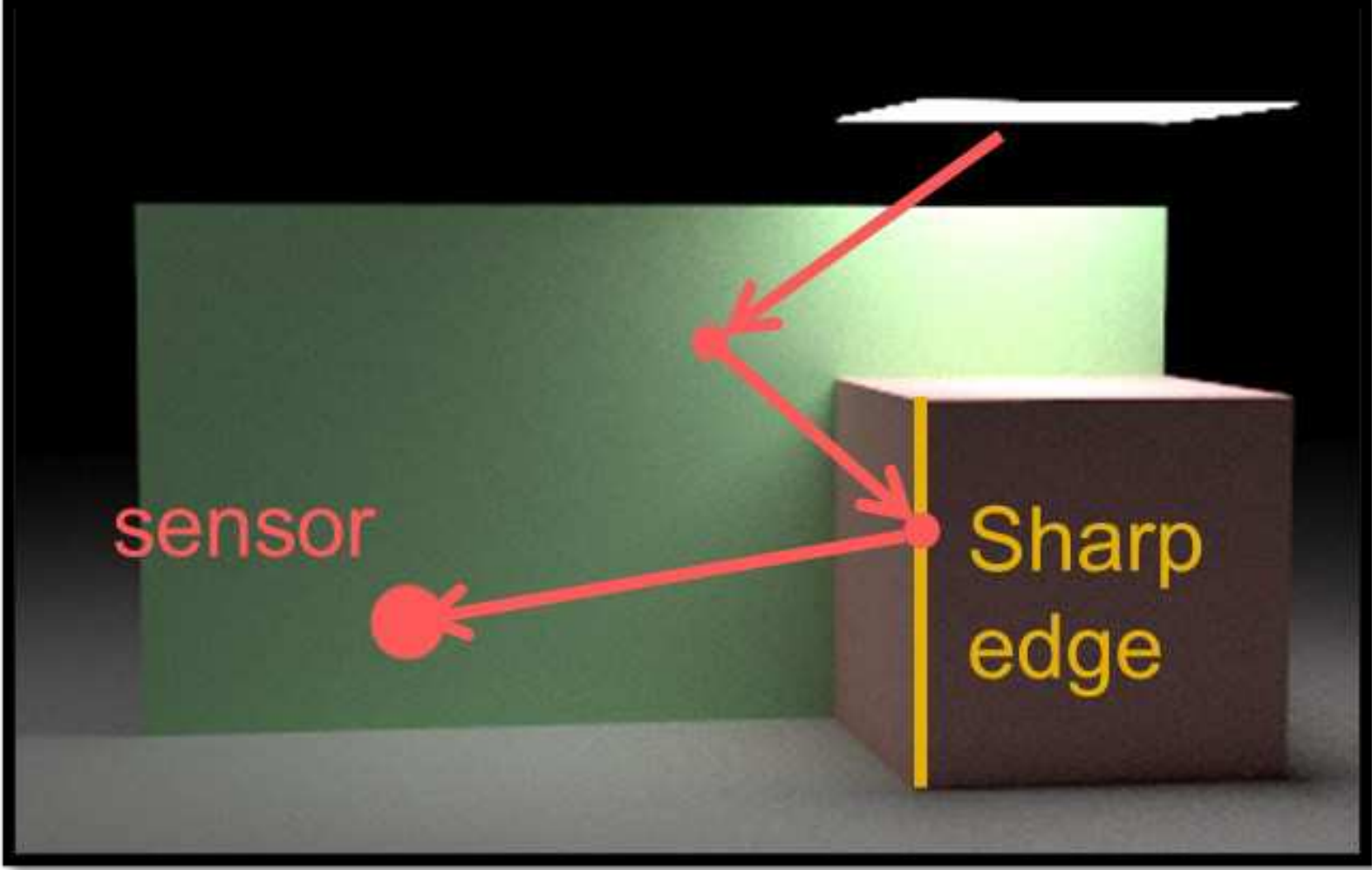
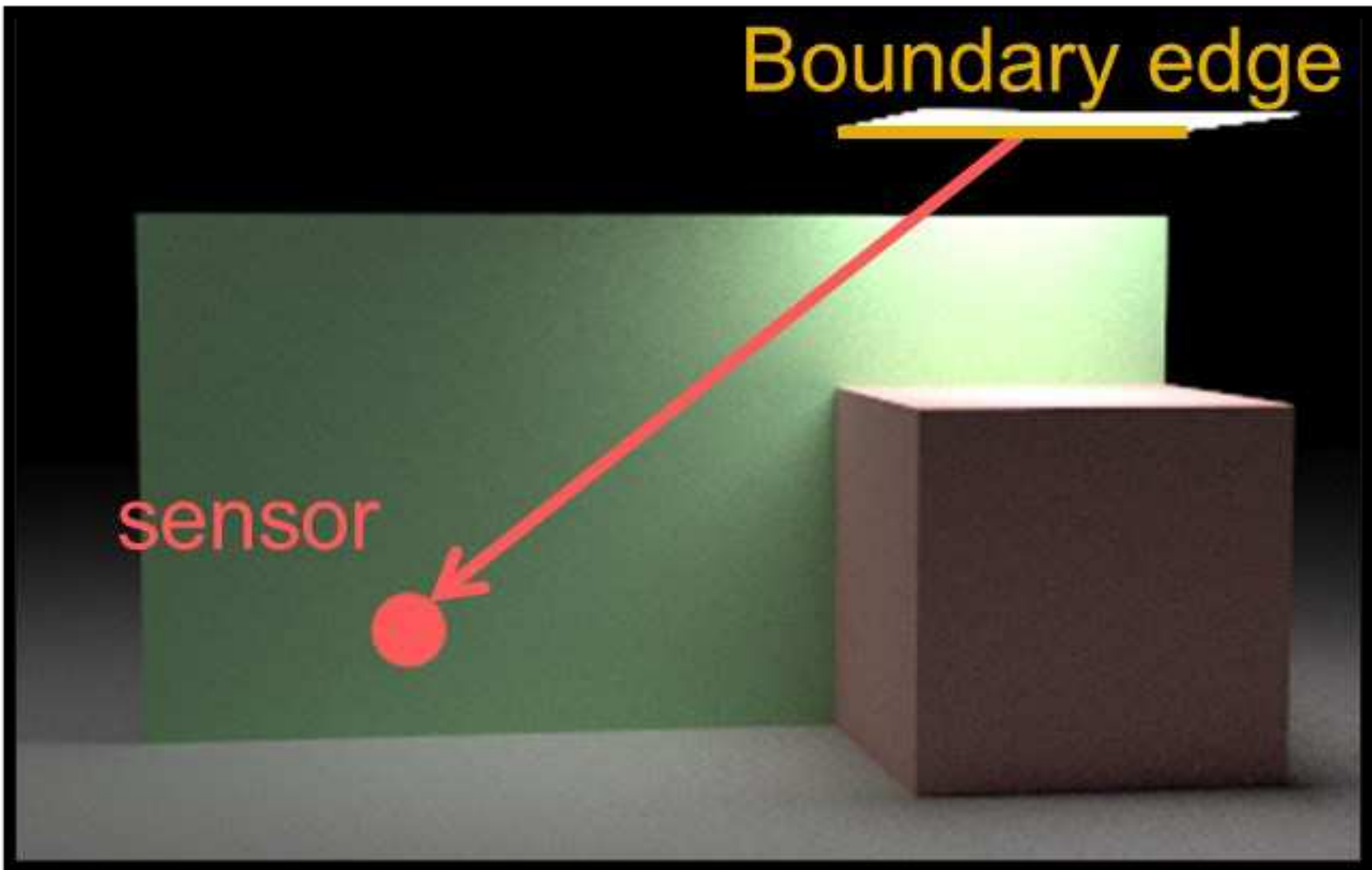
Boundary edge



Sharp edge



Silhouette edge

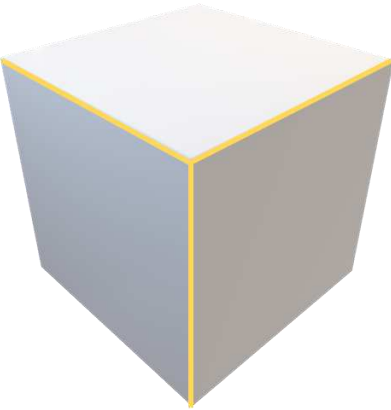


Sources of discontinuities

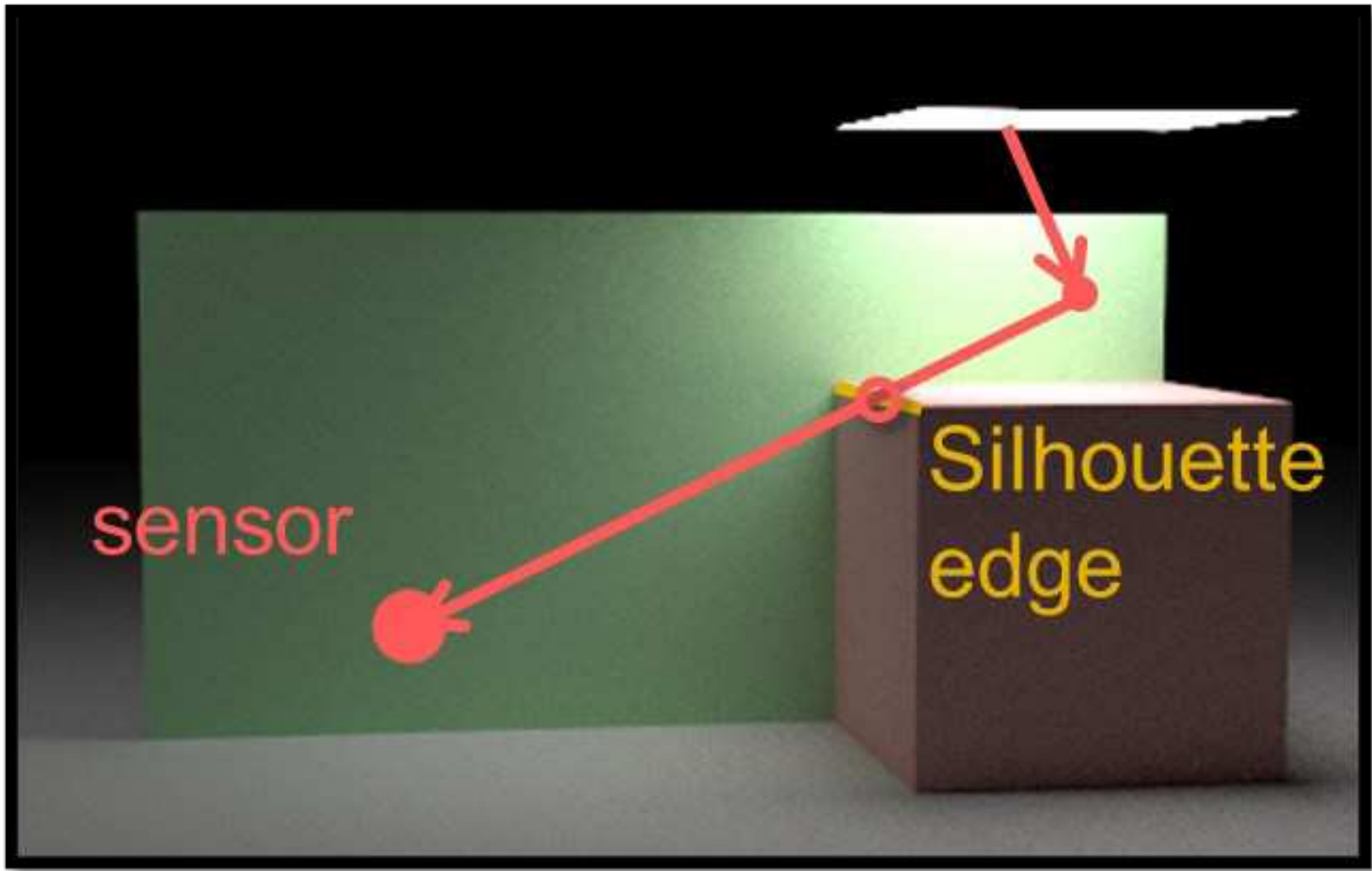
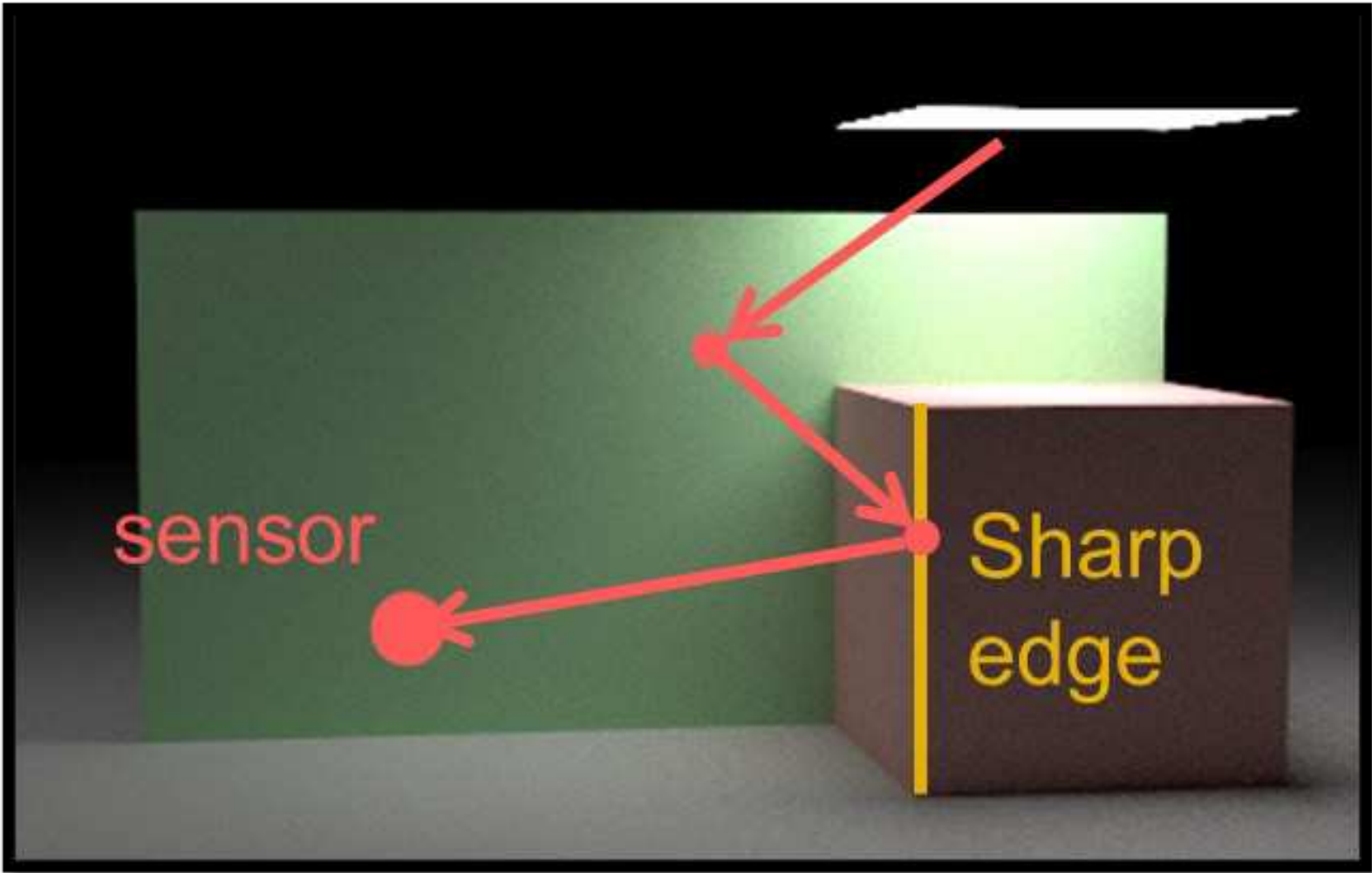
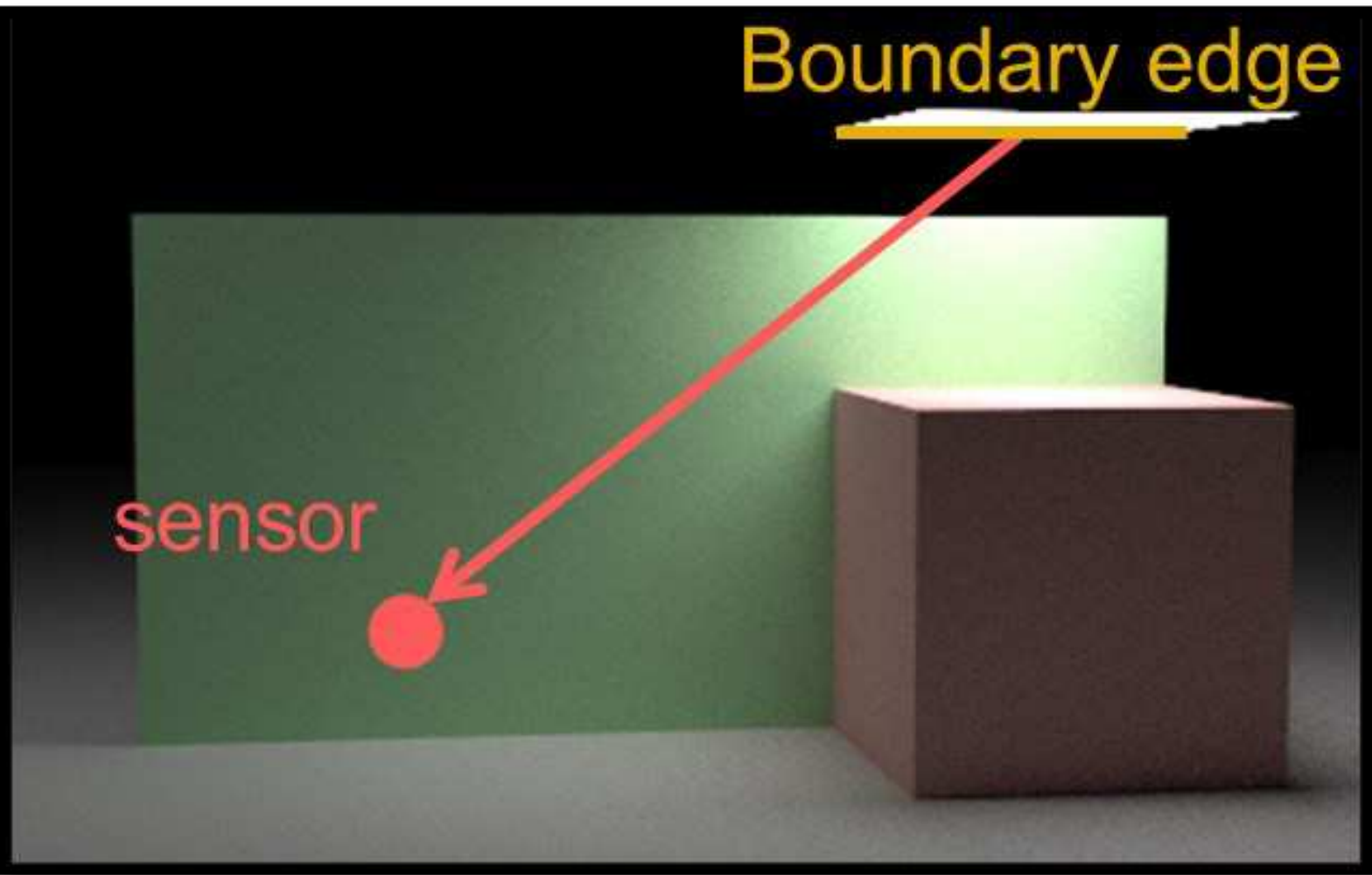
Boundary edge



Sharp edge



Silhouette edge

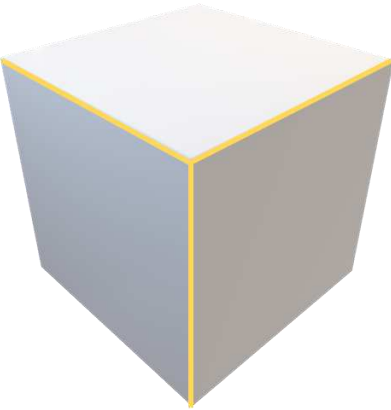


Sources of discontinuities

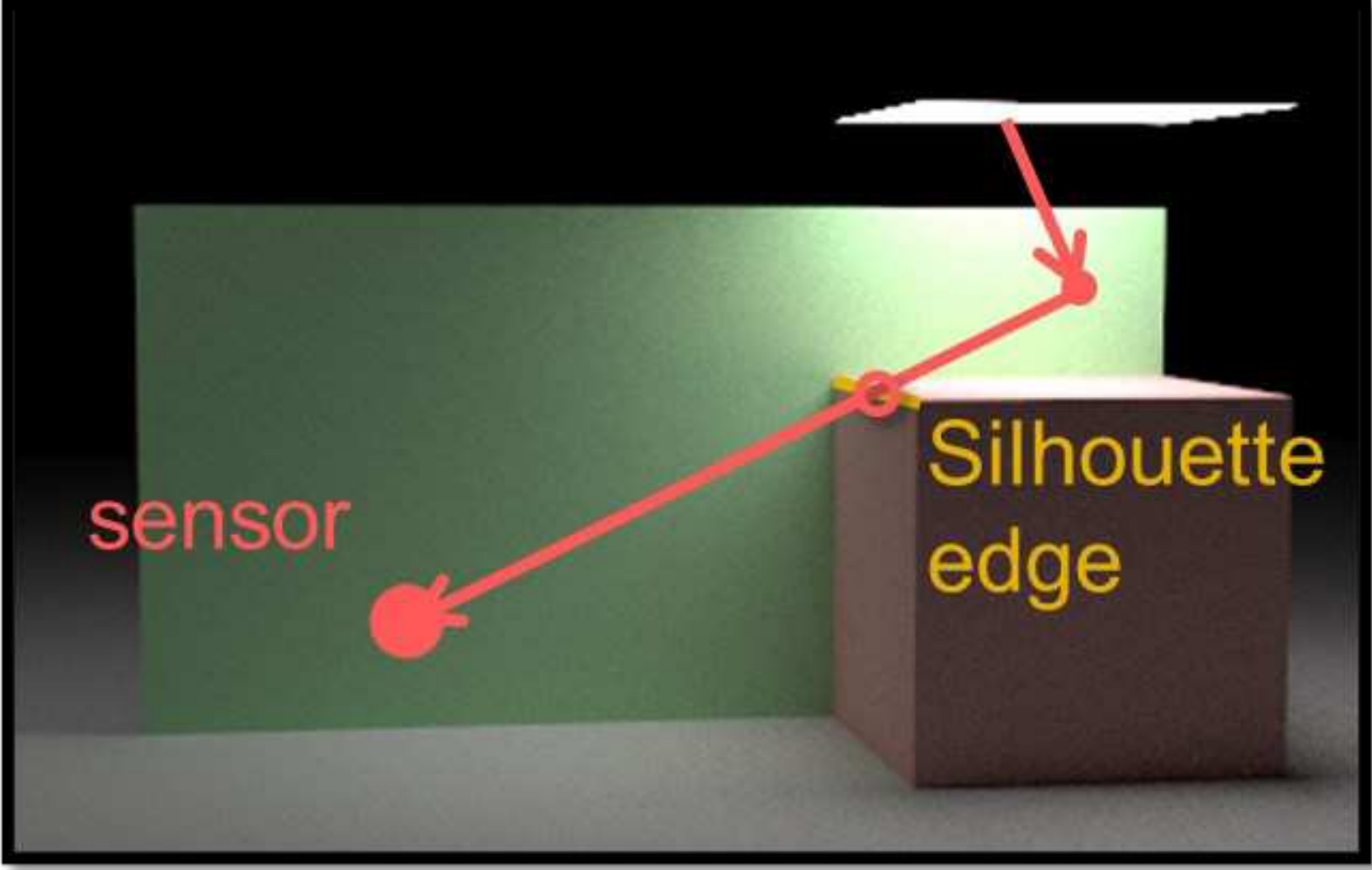
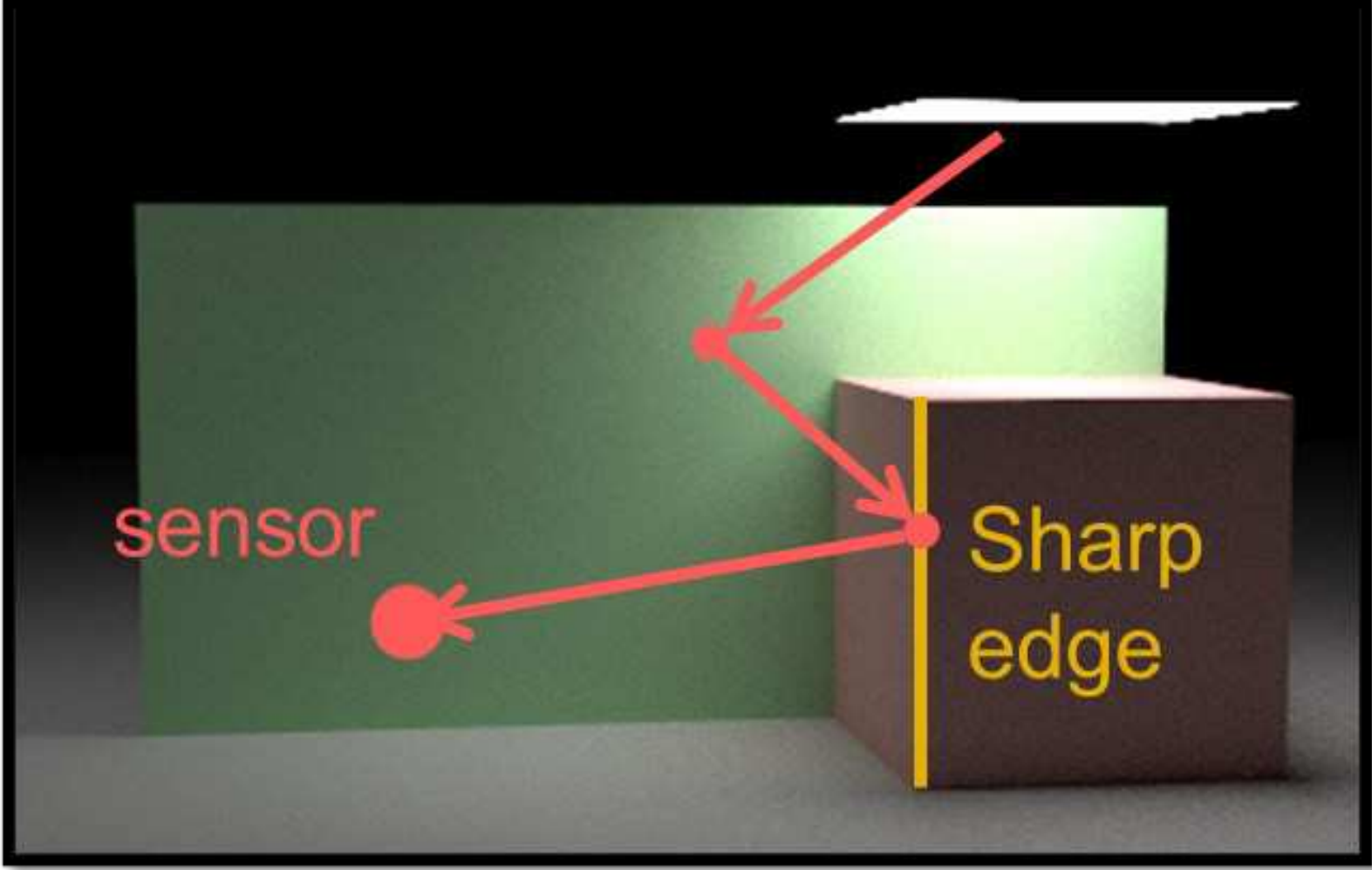
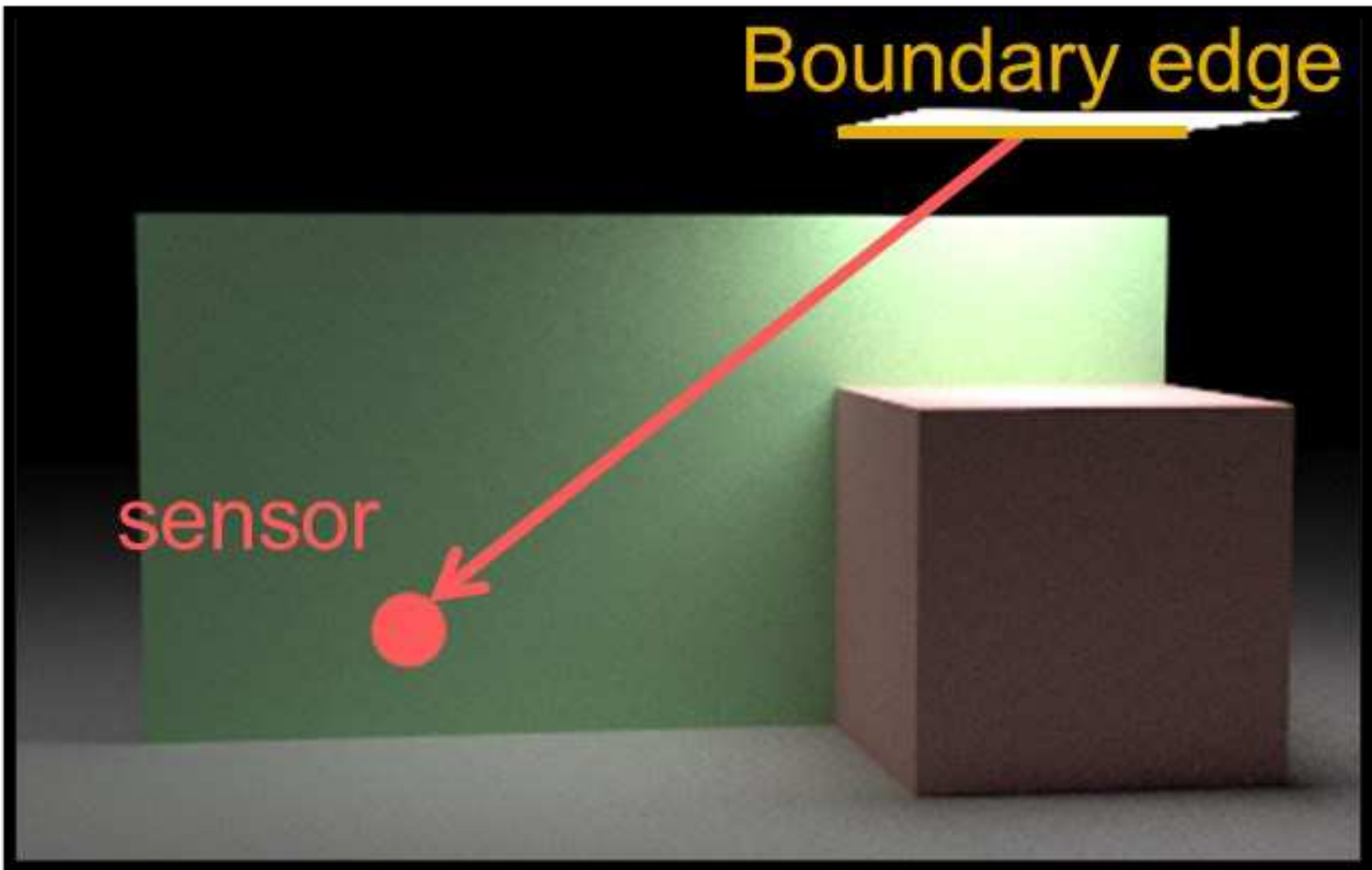
Boundary edge



Sharp edge



Silhouette edge

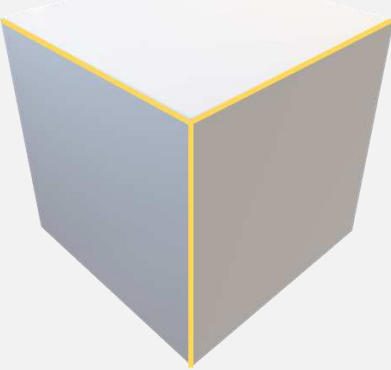


Sources of discontinuities

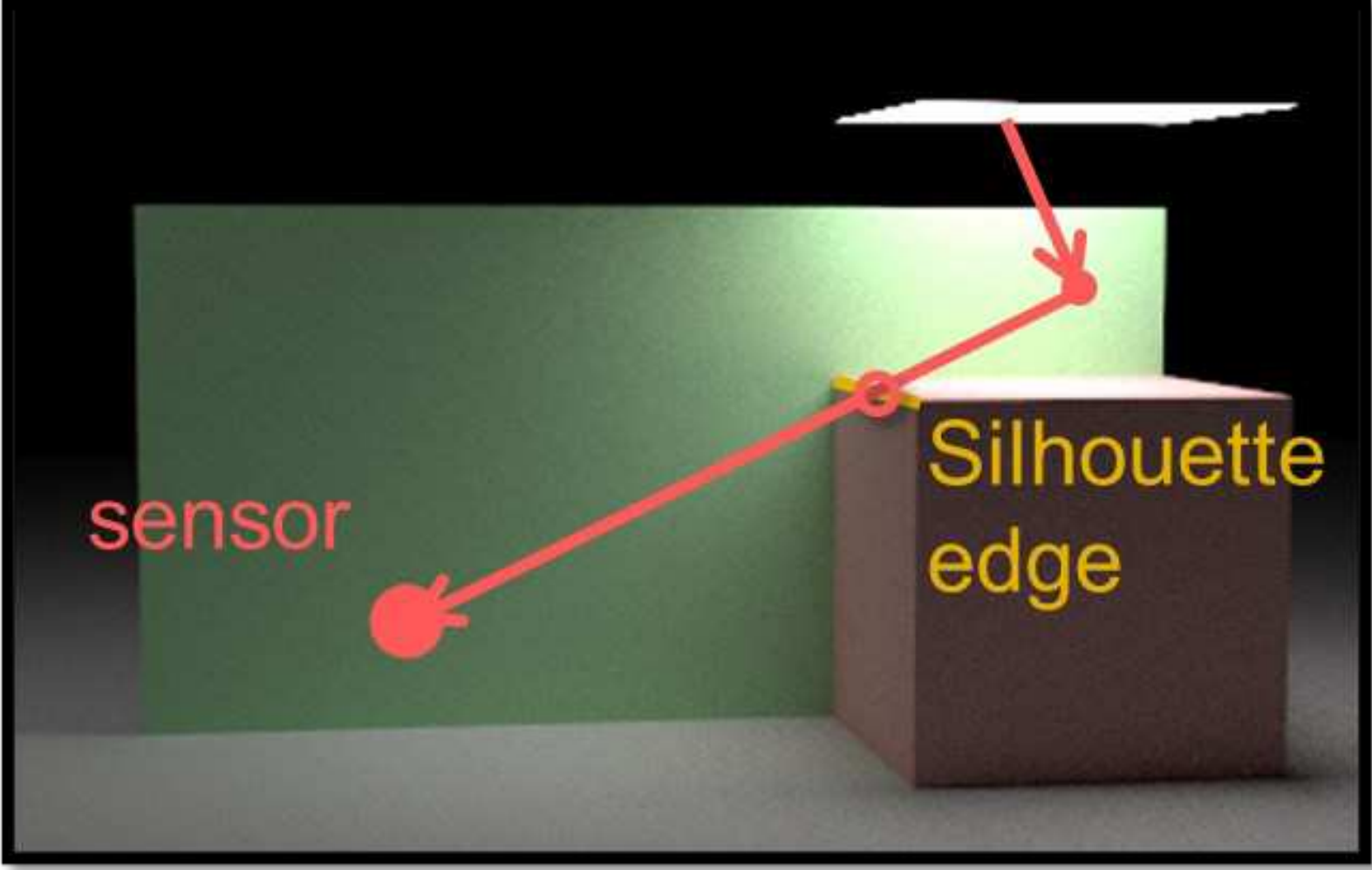
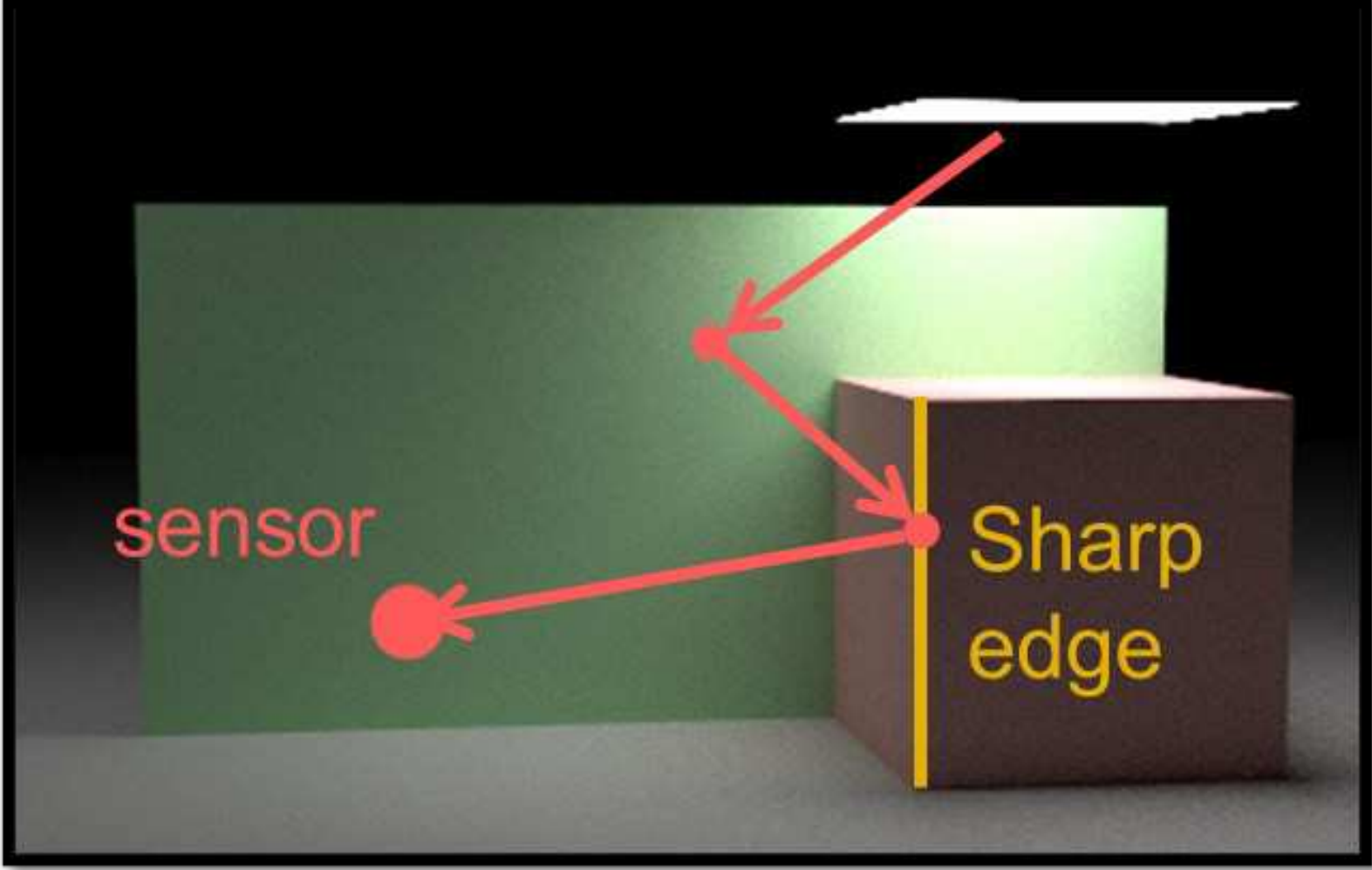
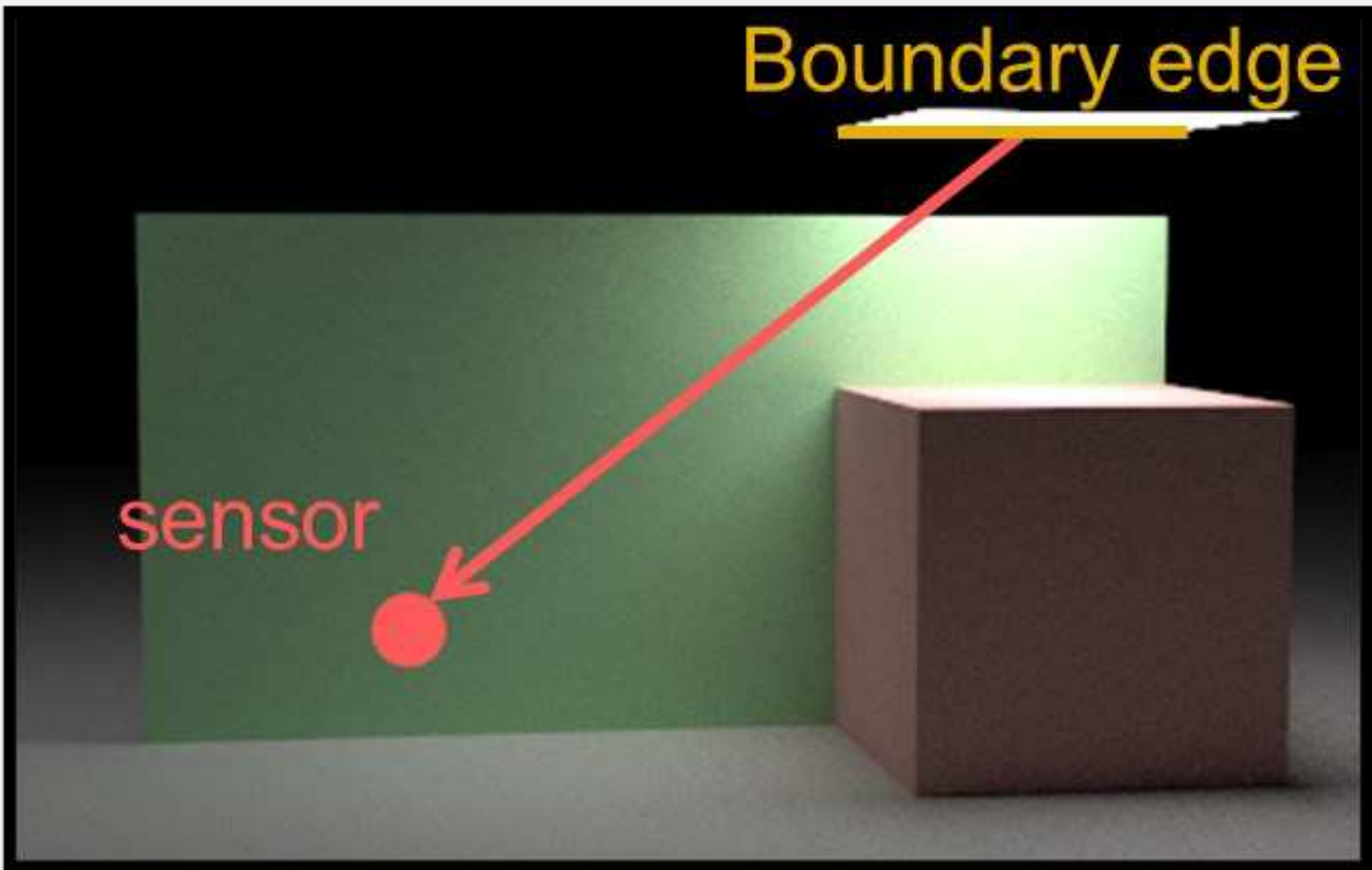
Boundary edge



Sharp edge



Silhouette edge



Topology-driven

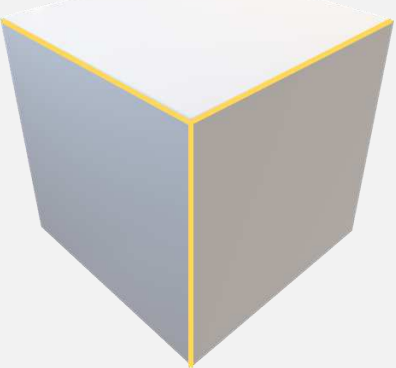
Visibility-driven

Sources of discontinuities

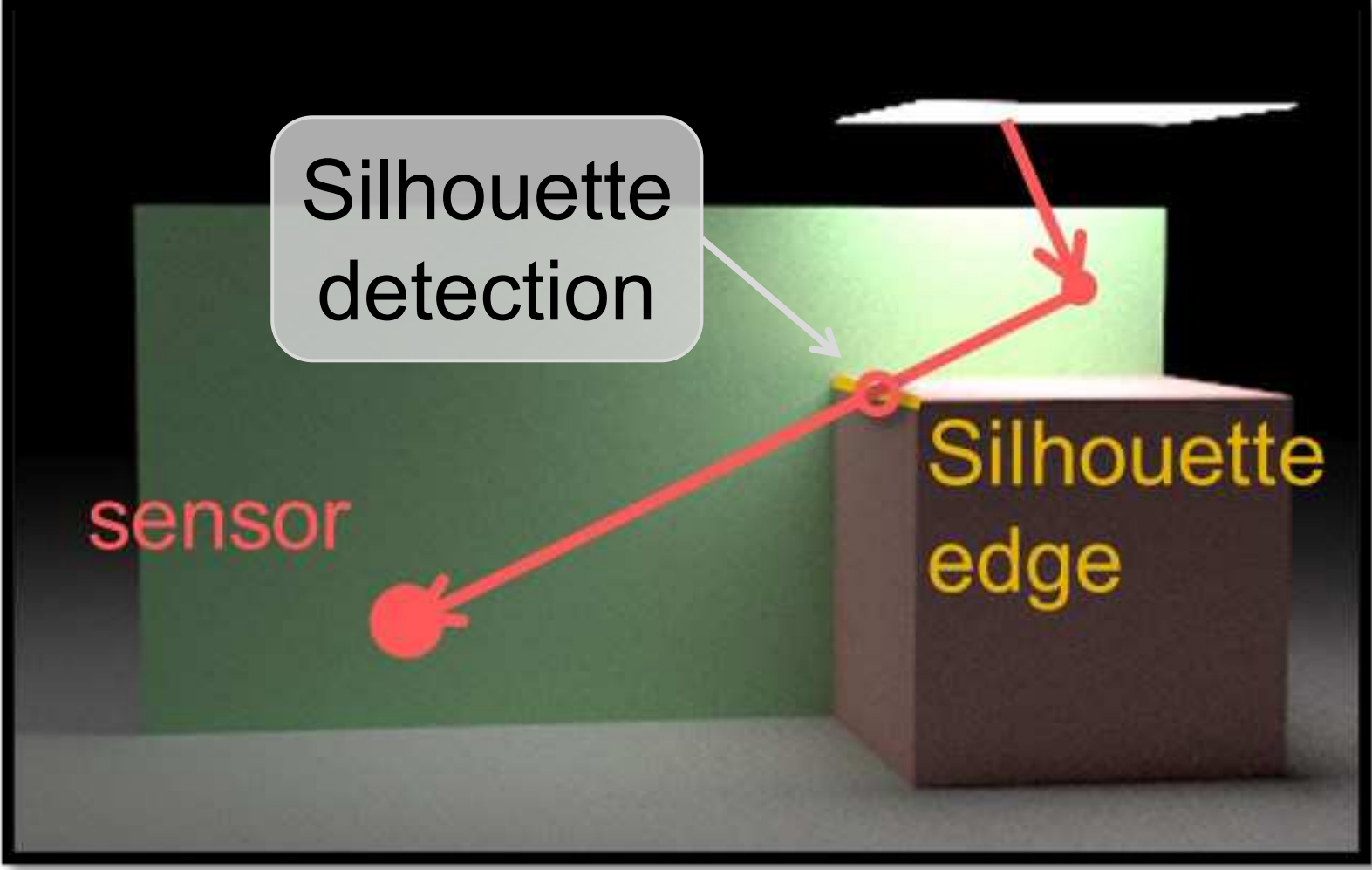
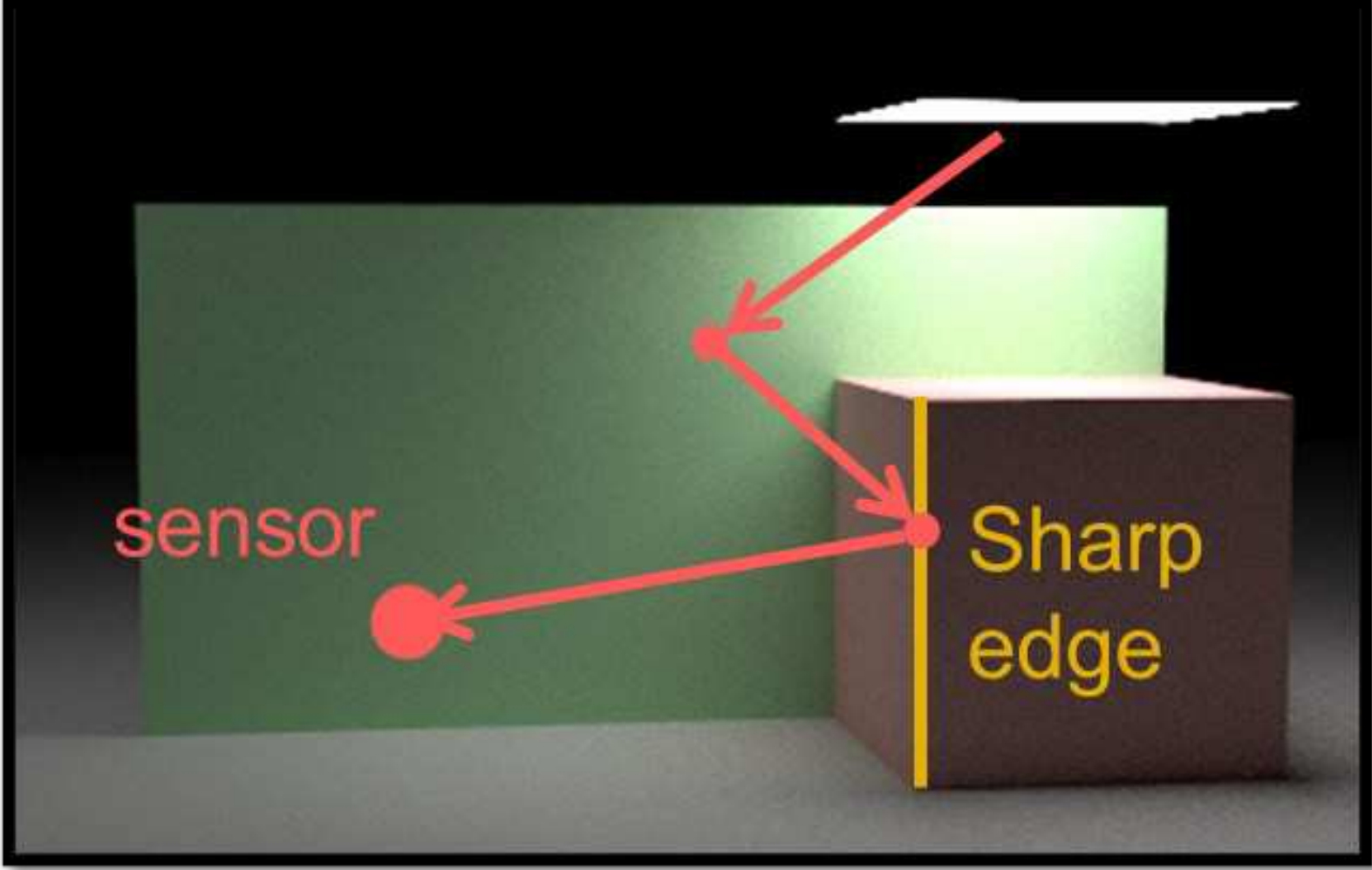
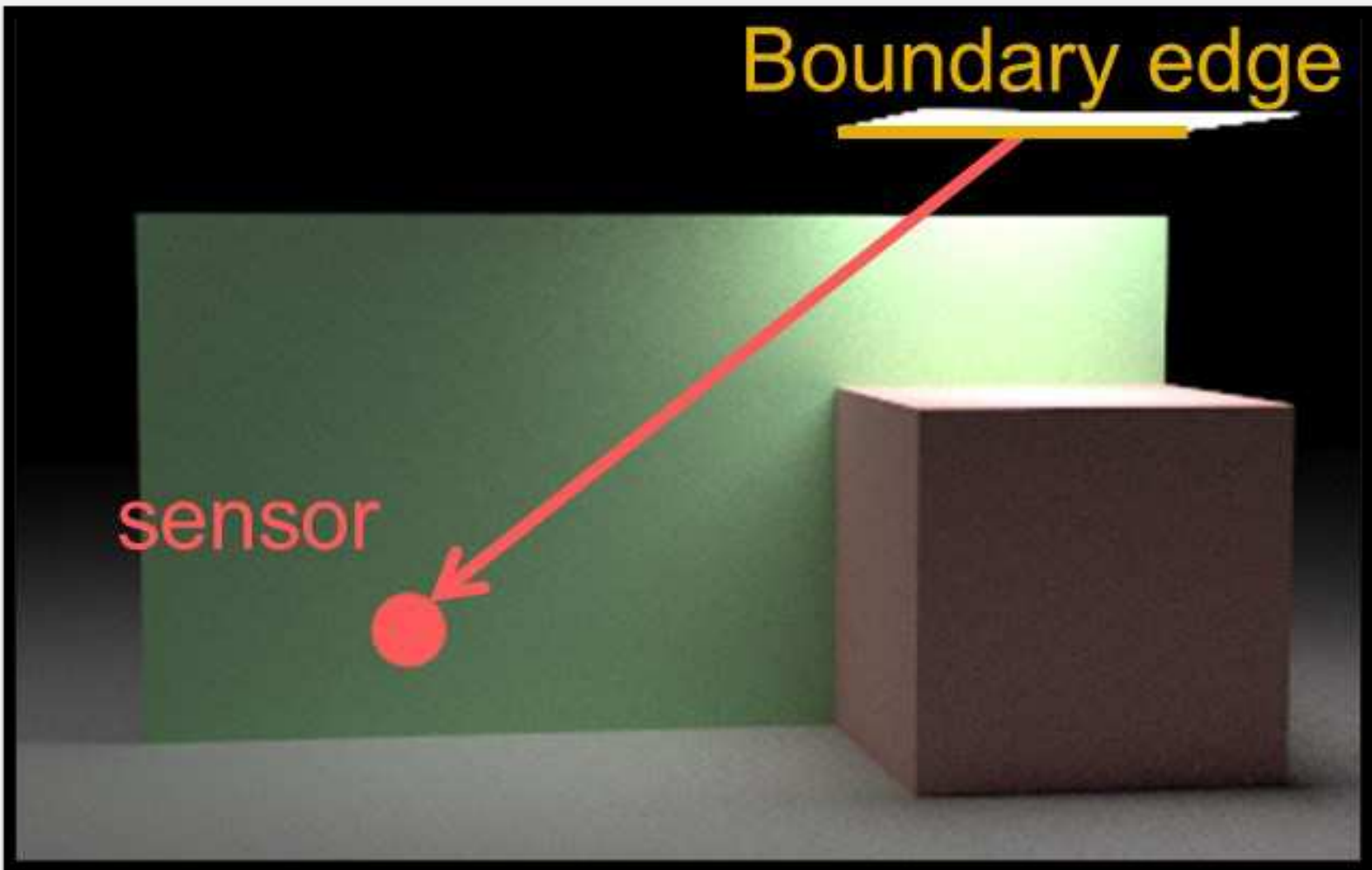
Boundary edge



Sharp edge



Silhouette edge



Topology-driven

Visibility-driven

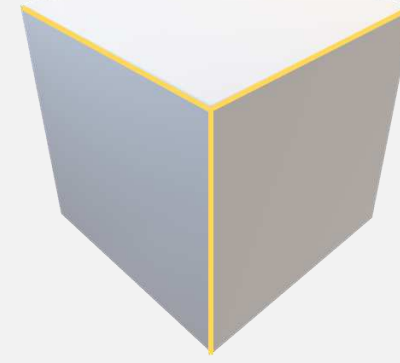
Sources of discontinuities

- We still need to account for discontinuities when using smooth closed surfaces (e.g., neural SDFs)

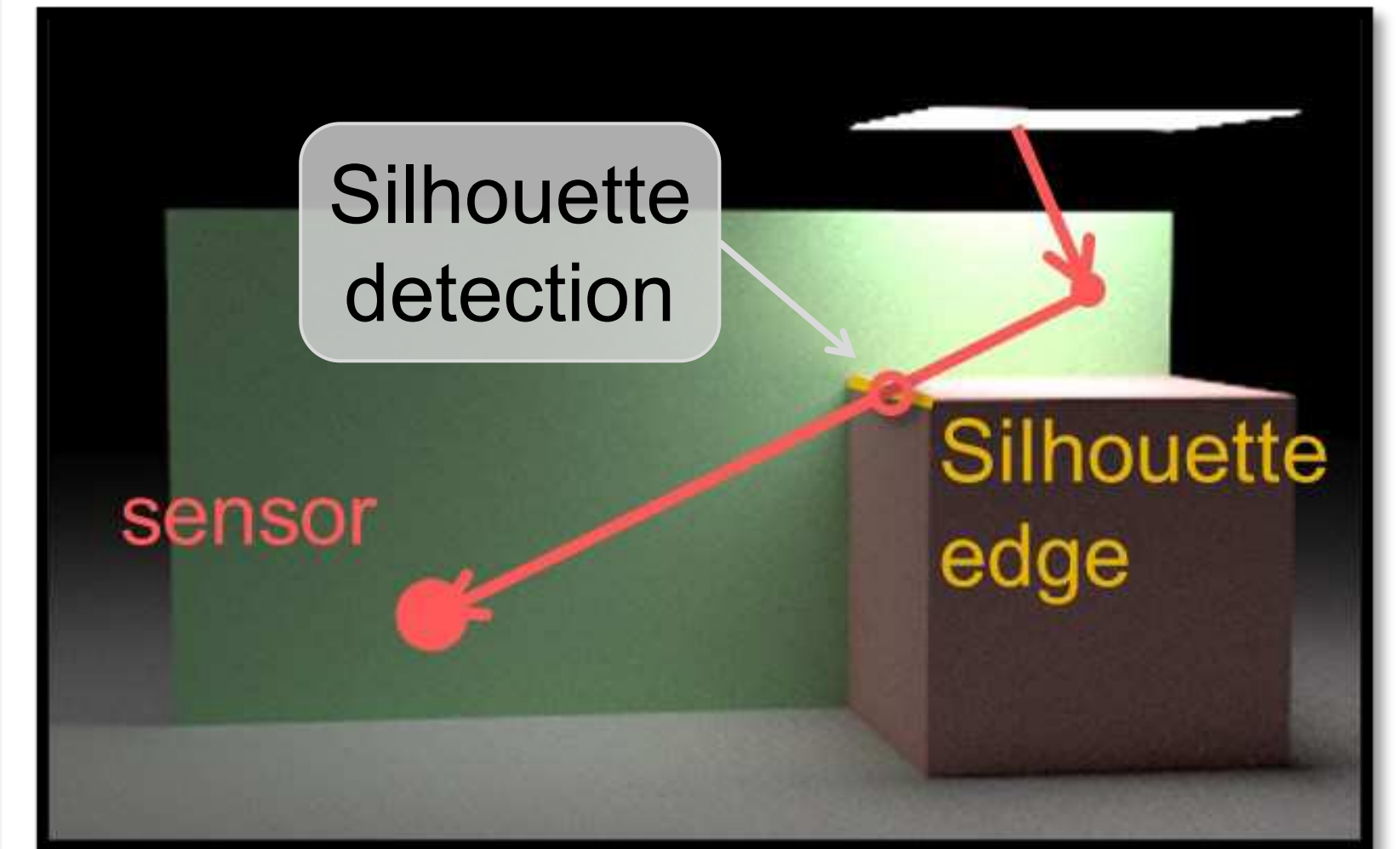
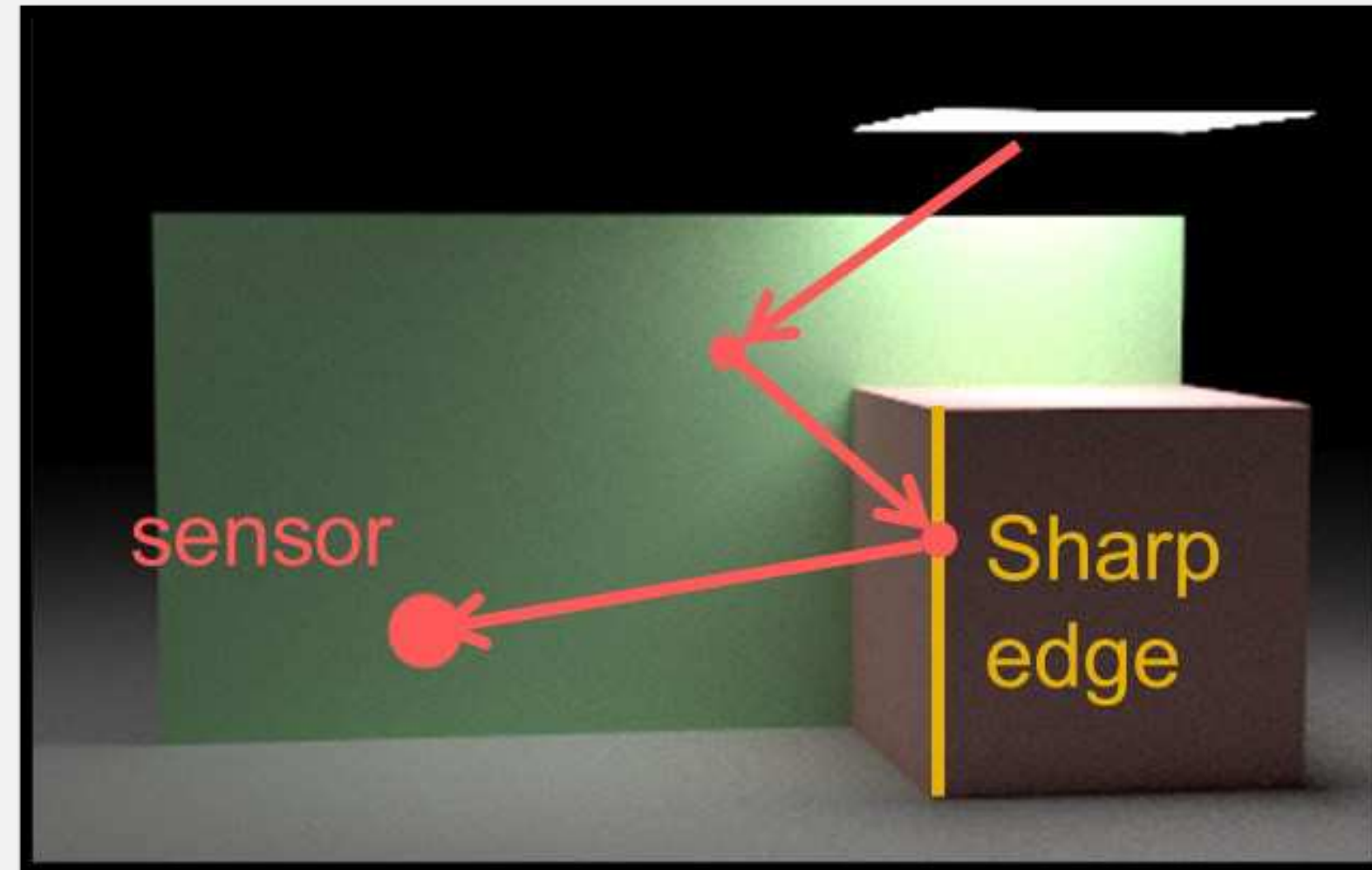
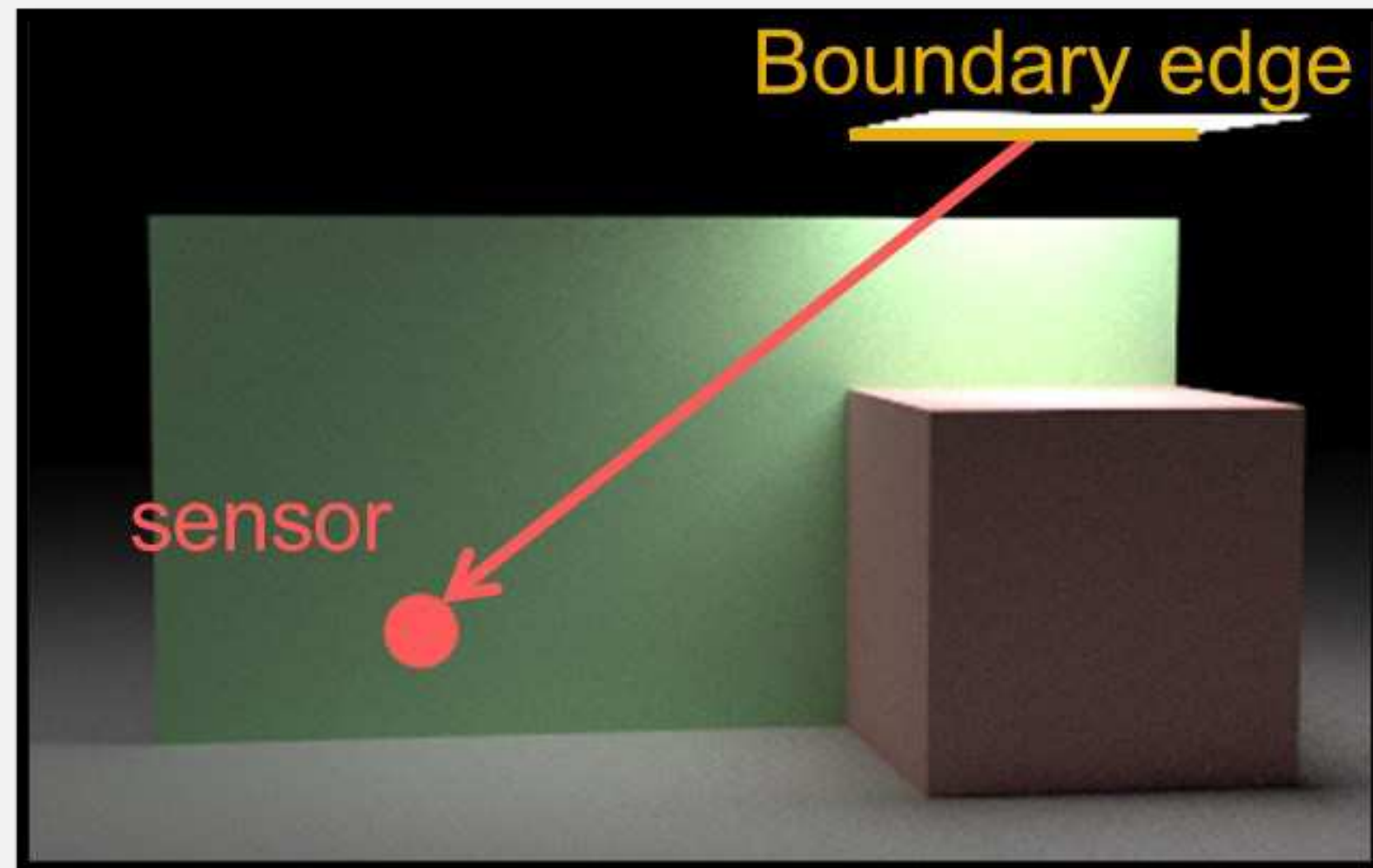
Boundary edge



Sharp edge



Silhouette edge



Topology-driven

Visibility-driven

Sources of discontinuities

- We still need to account for discontinuities when using smooth closed surfaces (e.g., neural SDFs)

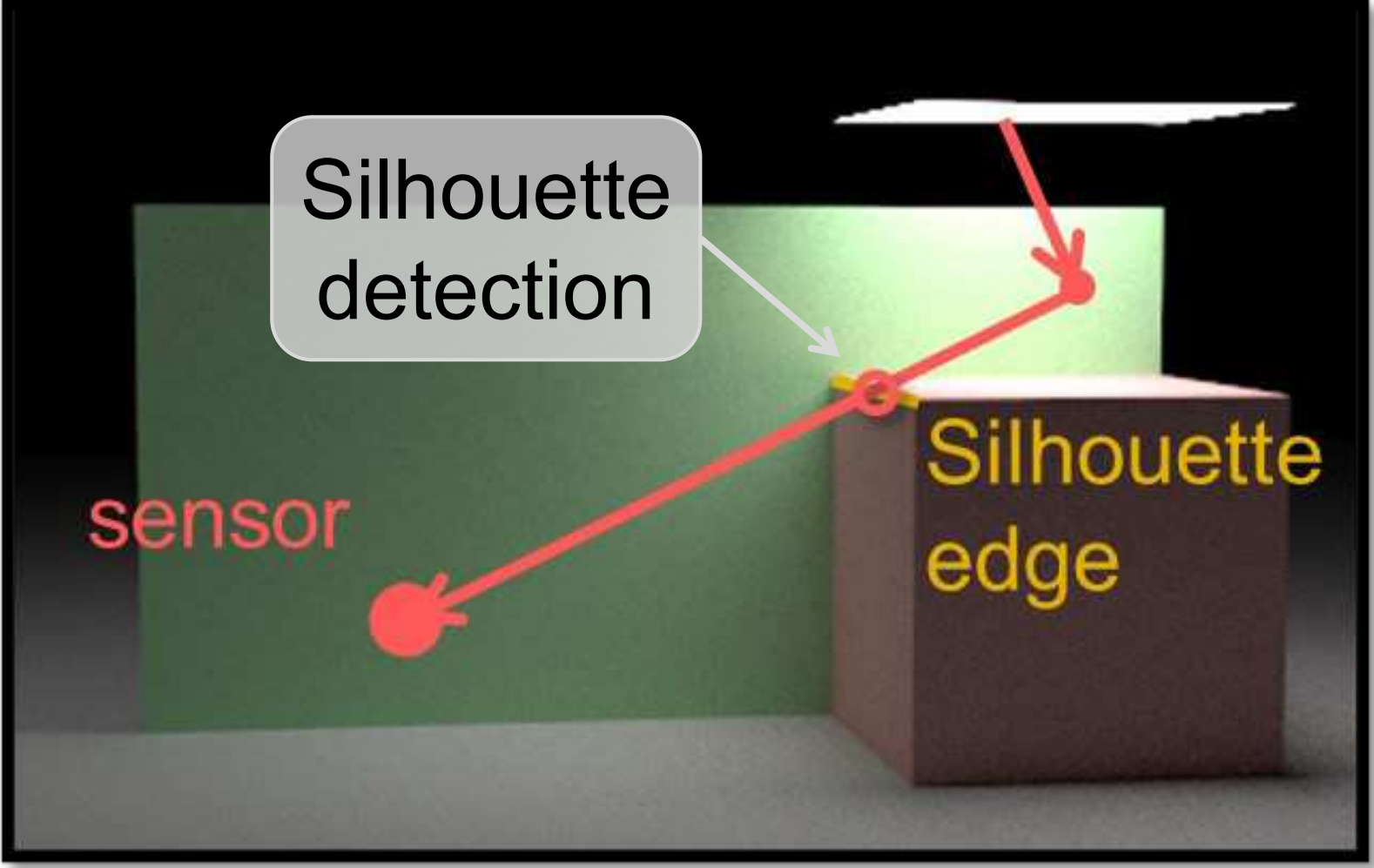
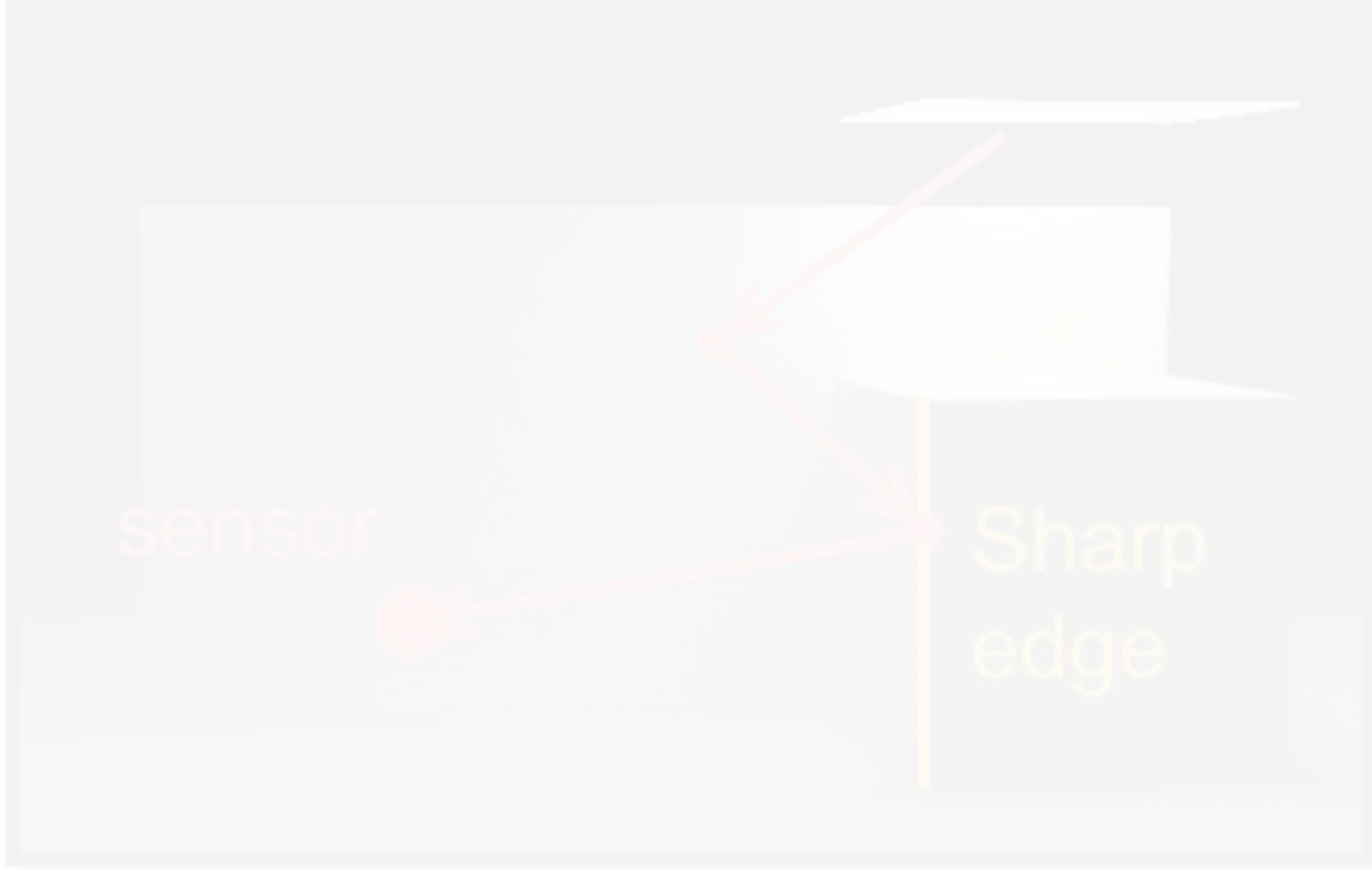
Boundary edge



Sharp edge



Silhouette edge

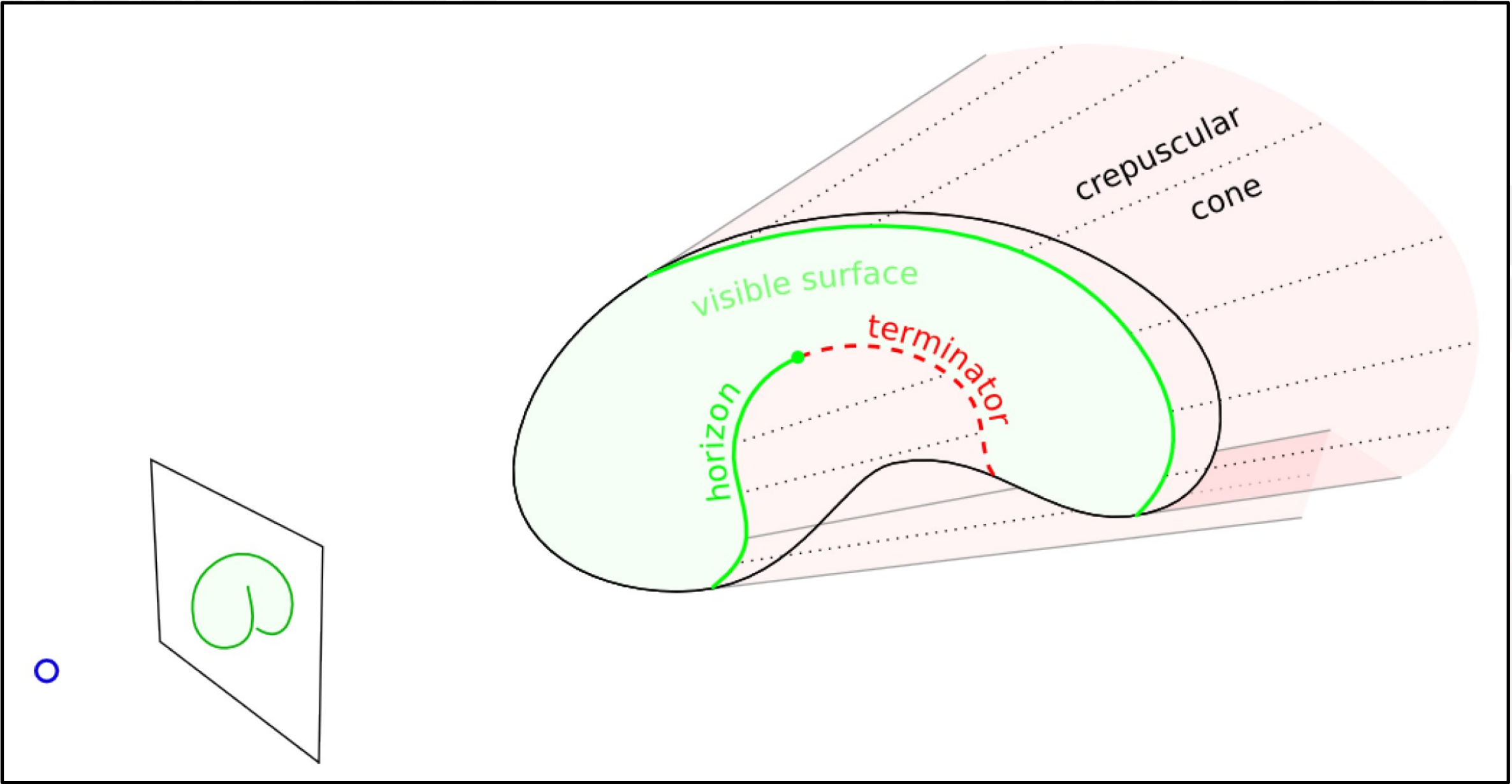


Topology-driven

Visibility-driven

Sources of discontinuities

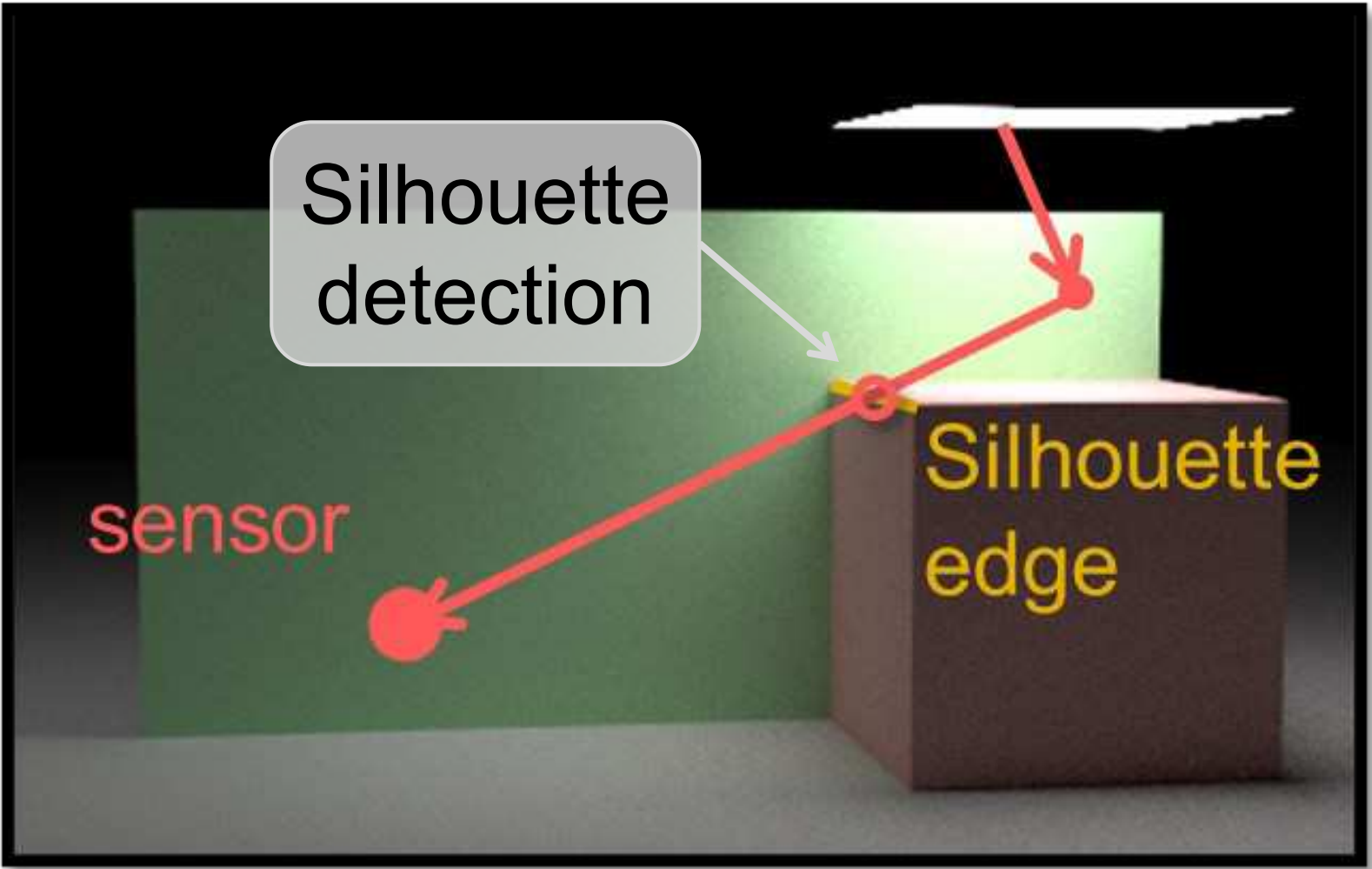
- We still need to account for discontinuities when using smooth closed surfaces (e.g., neural SDFs)



[Gargallo et al., ICCV 2007]

Topology-driven

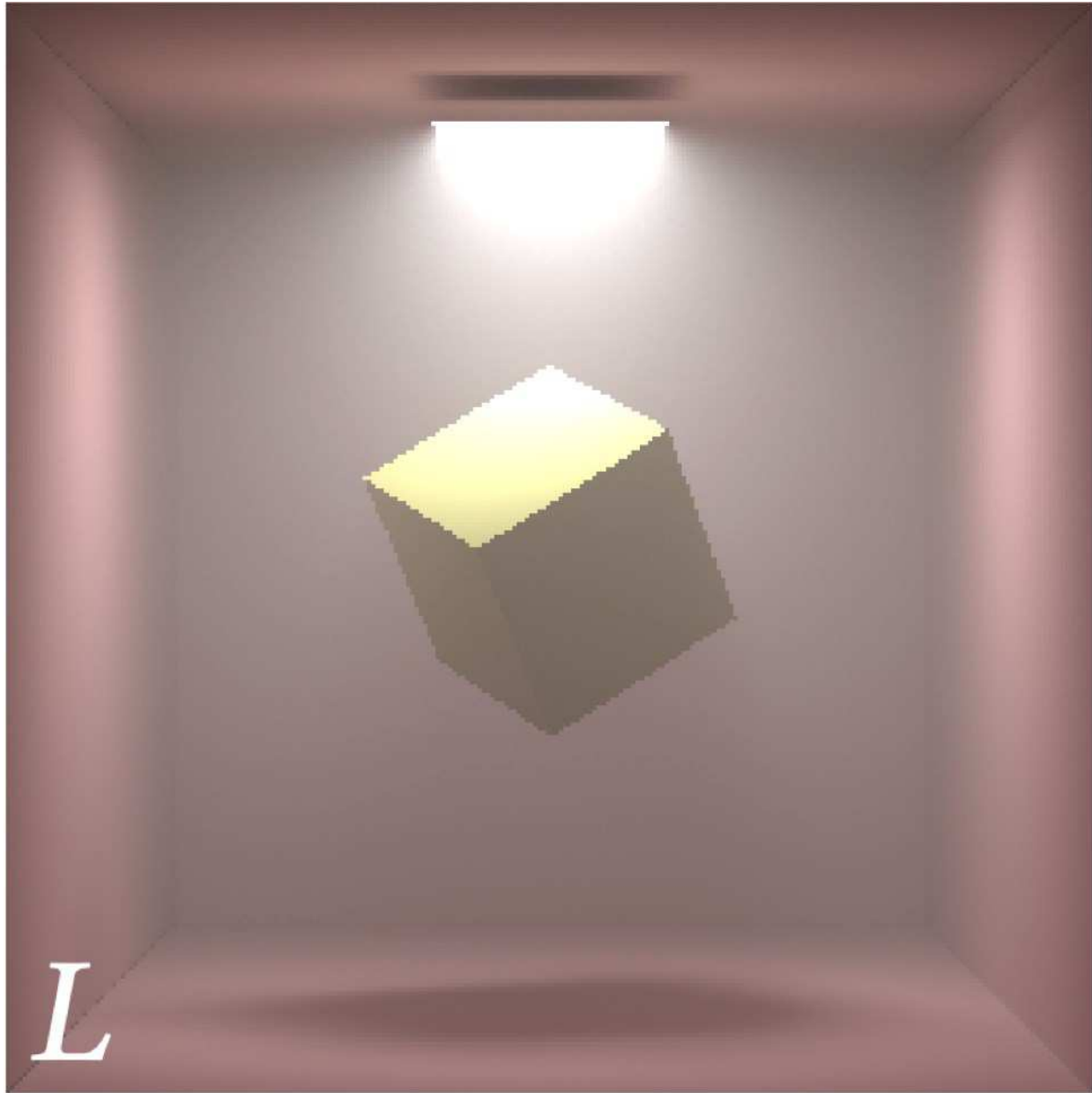
Silhouette edge



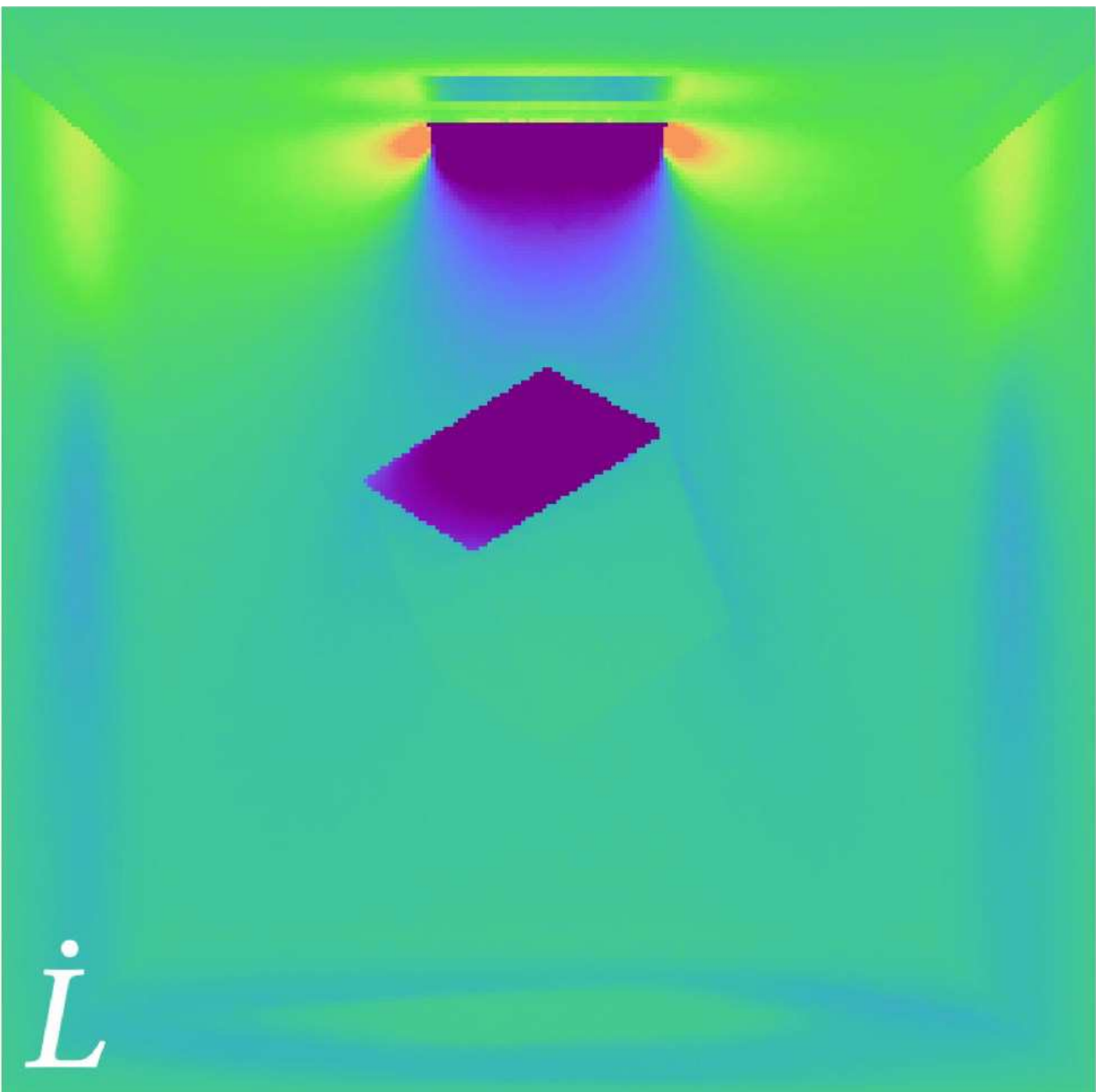
Visibility-driven

Significance of the boundary integral

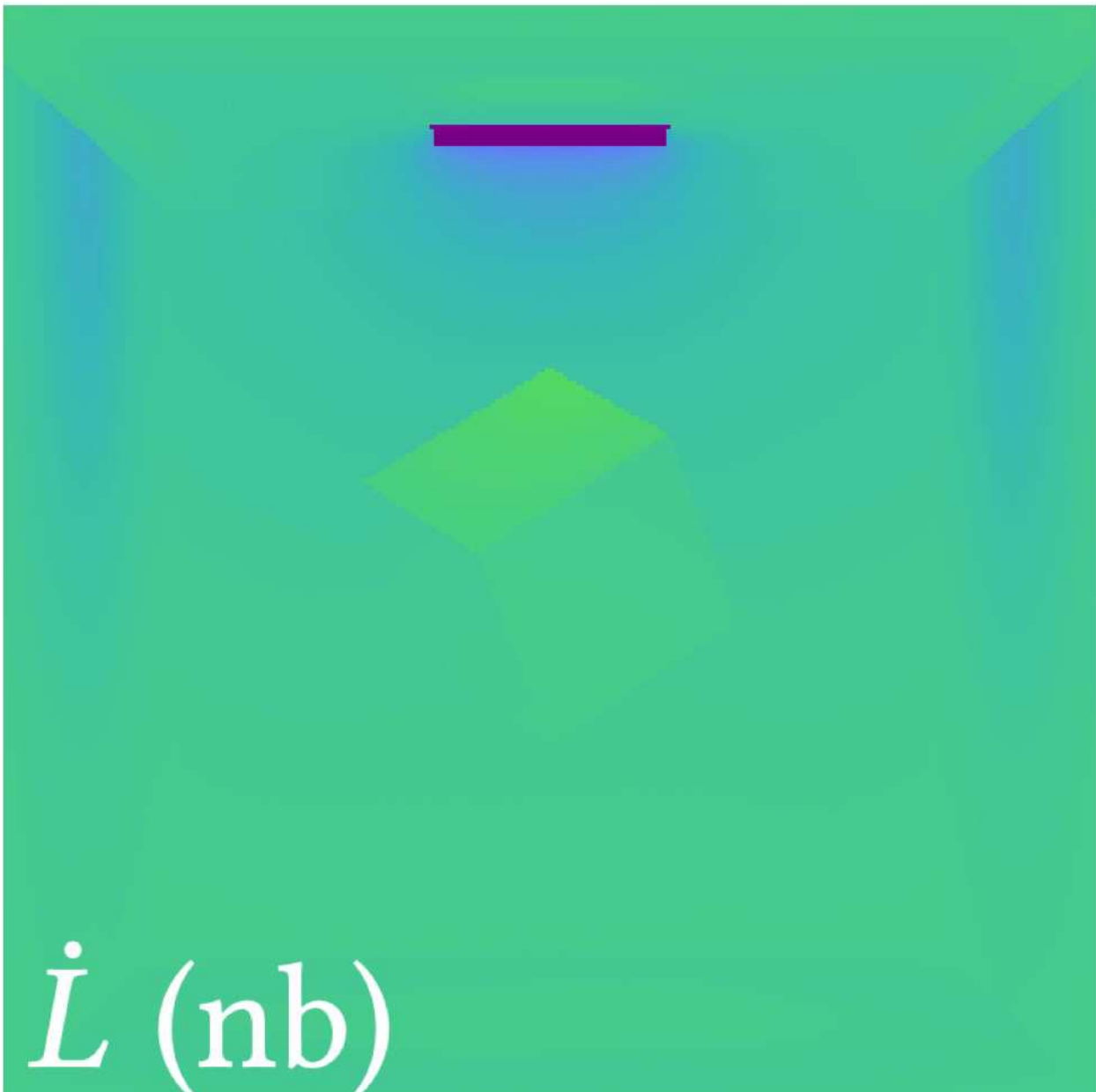
Negative  Zero  Positive



Original image



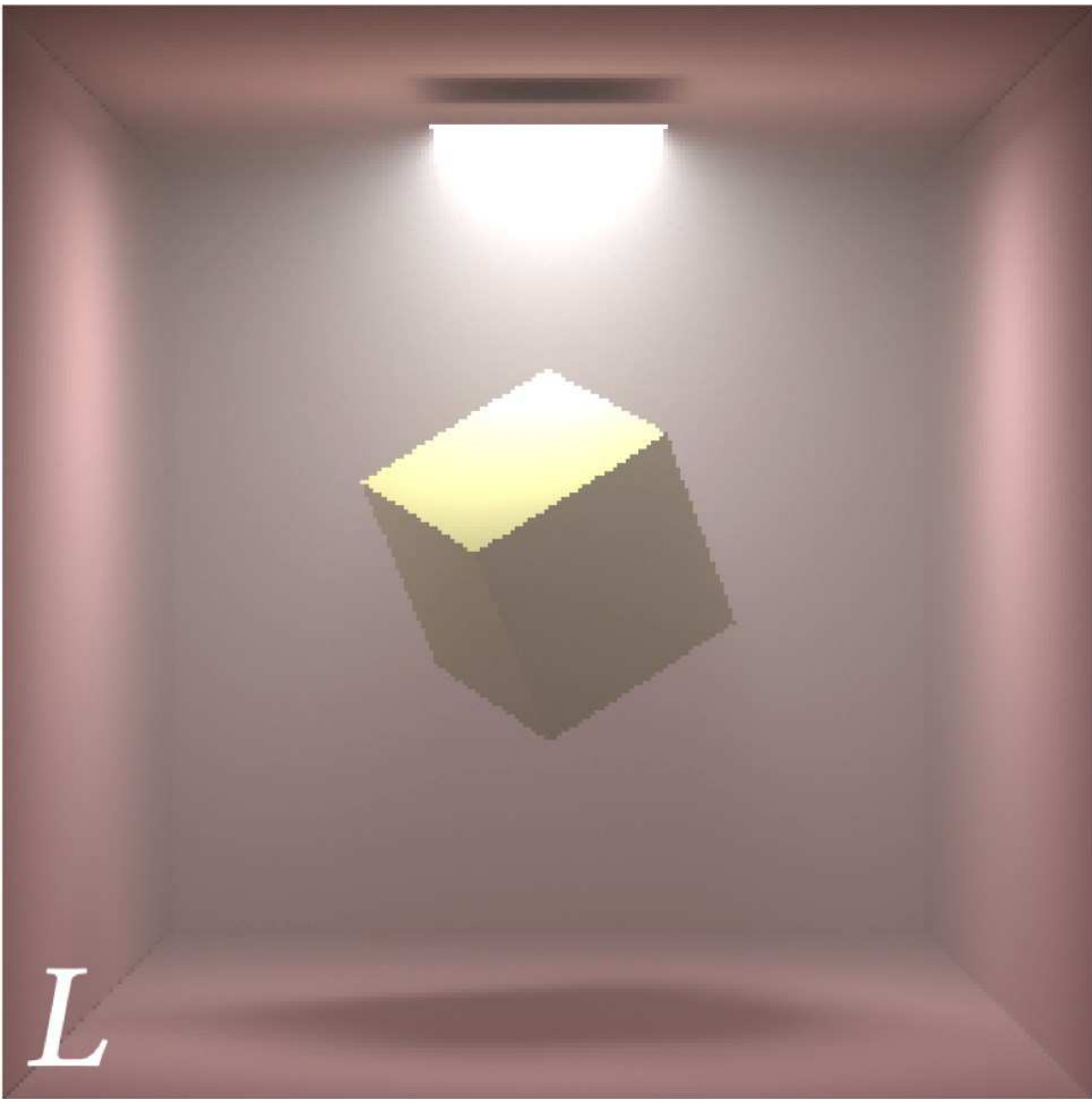
Derivative image
w.r.t. vertical offset of
the area light and the cube



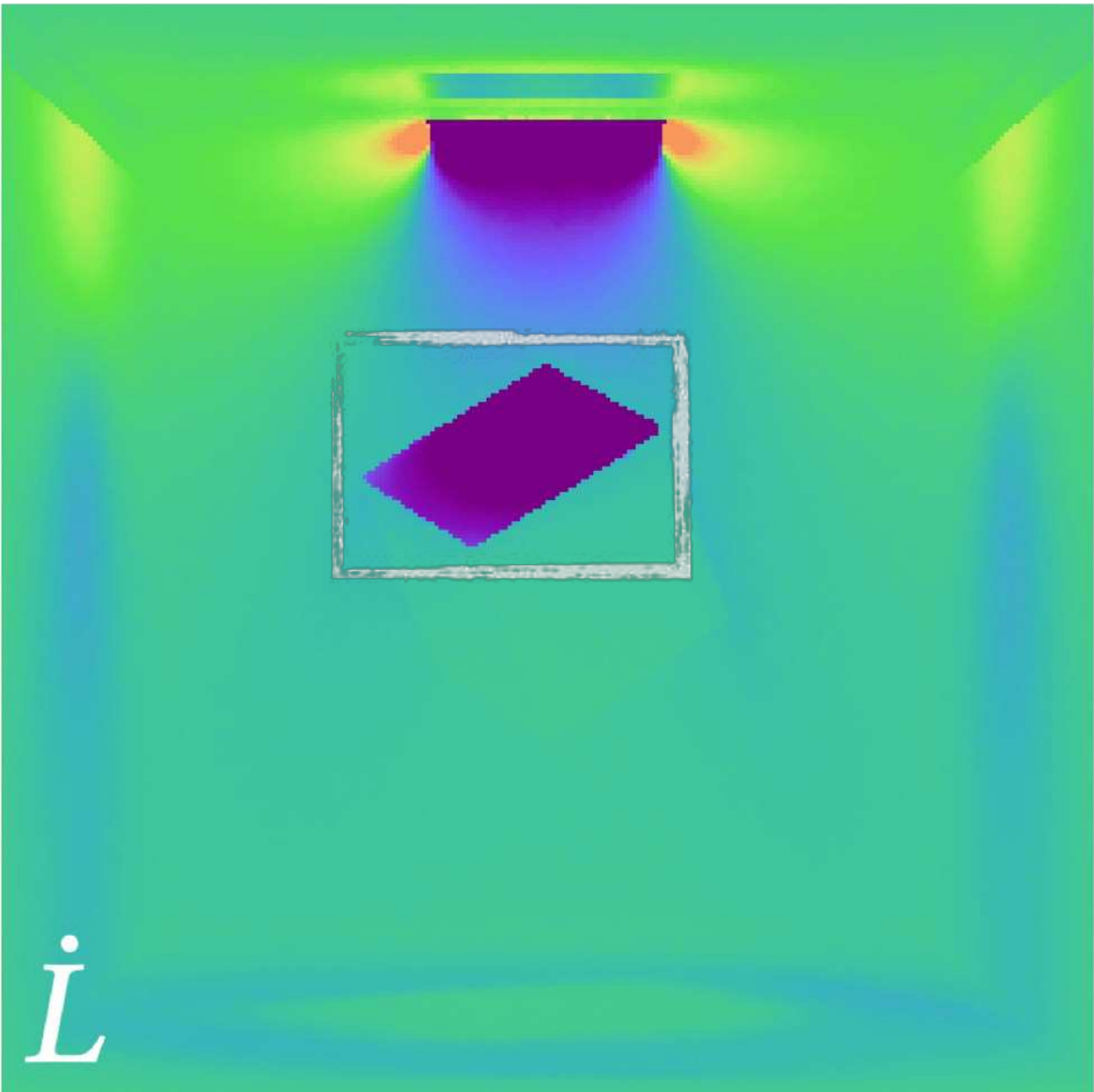
Derivative image
w/o boundary integral

Significance of the boundary integral

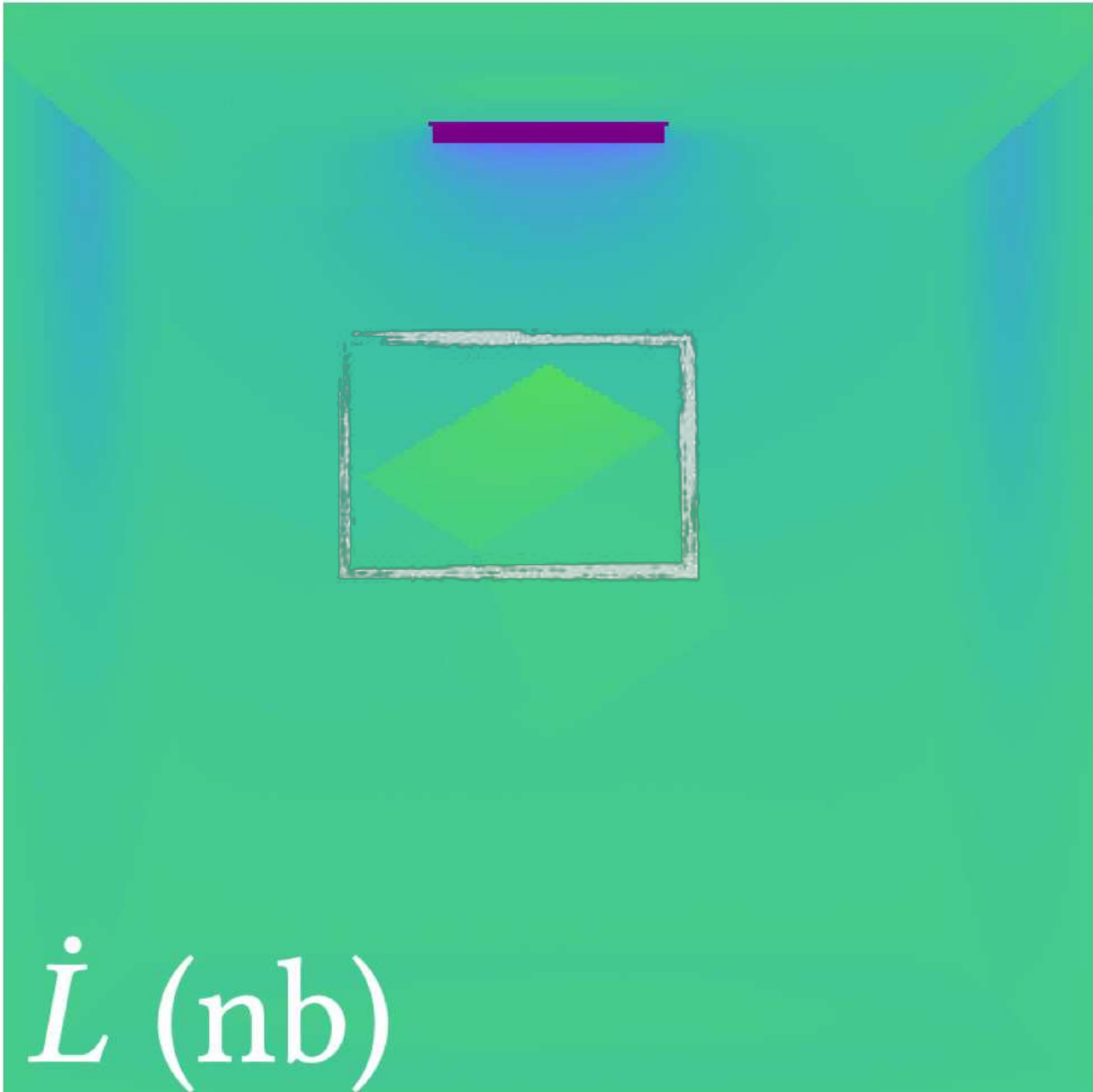
Negative  Zero  Positive



Original image



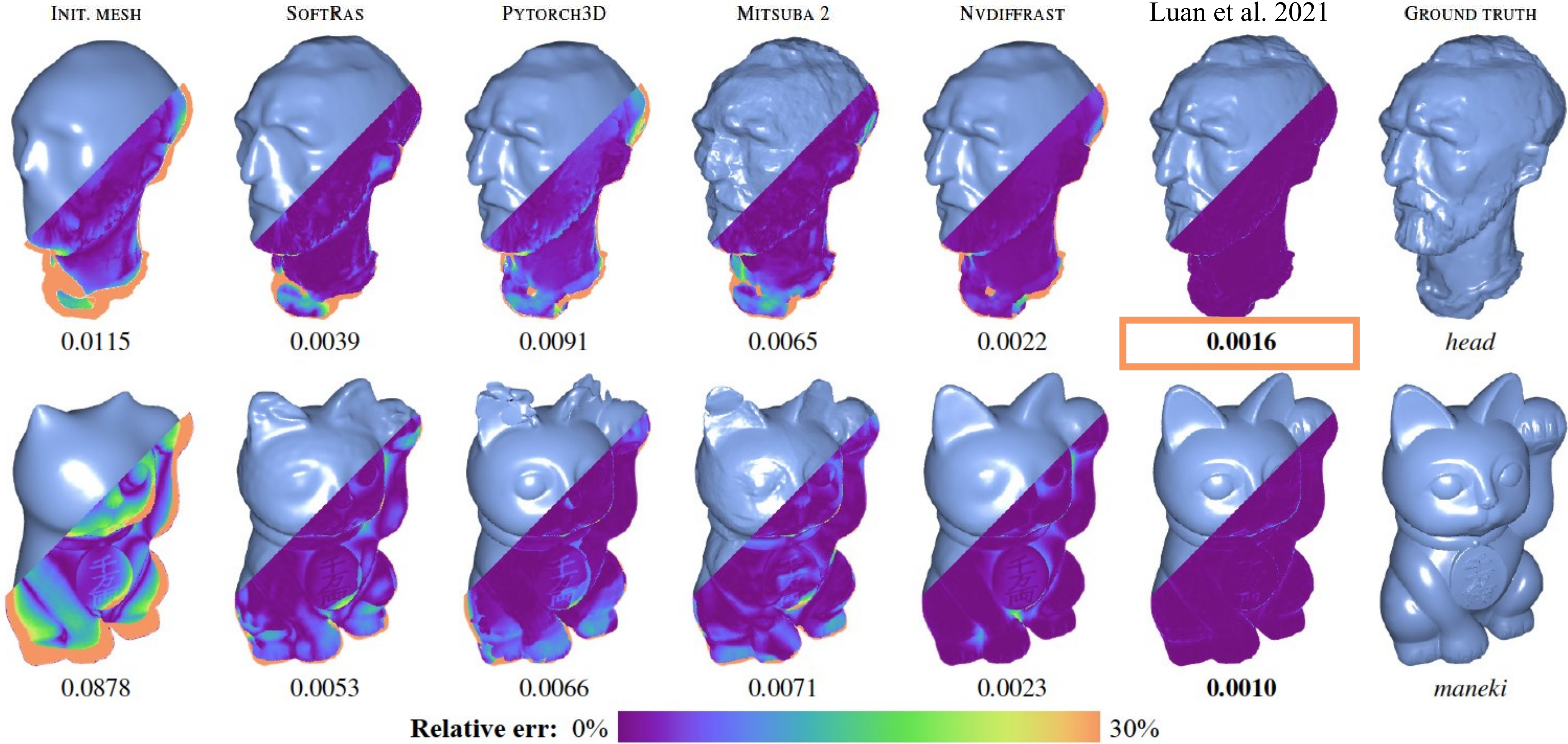
Derivative image
w.r.t. vertical offset of
the area light and the cube



Derivative image
w/o boundary integral

Gradient Accuracy Matters

Inverse-rendering results with *identical* optimization settings



Differential Global Illumination

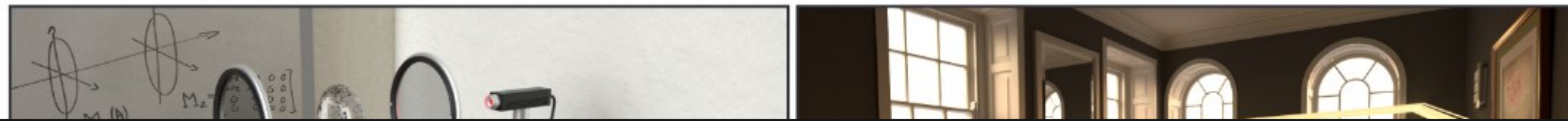
Very active area of research

Path-Space Differentiable Rendering

CHENG ZHANG, University of California, Irvine
BAILEY MILLER, Carnegie Mellon University
KALYAN, University of California, Irvine

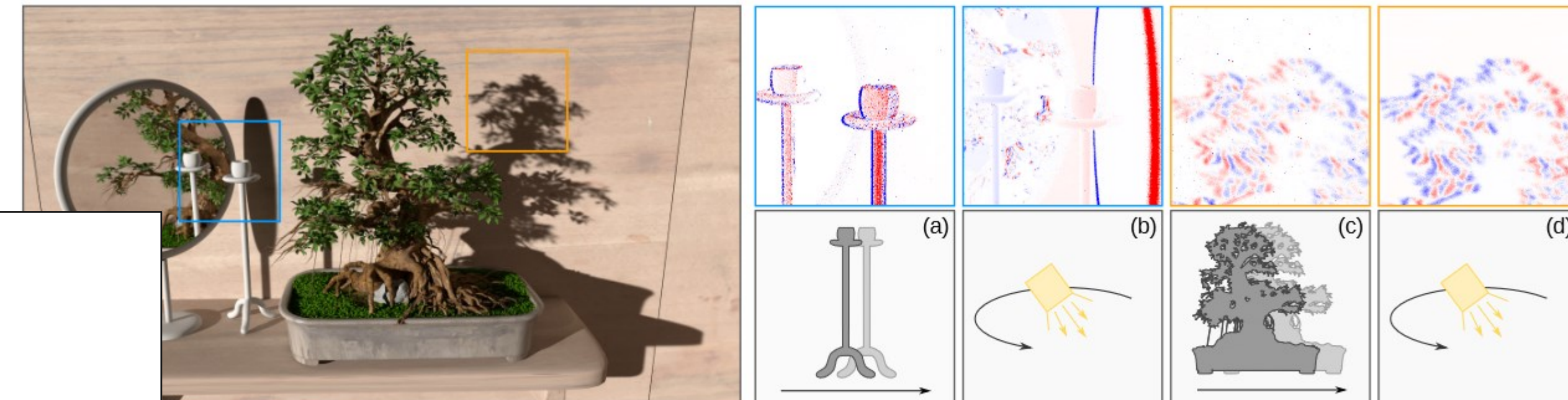
Mitsuba 2: A Retargetable Forward and Inverse Renderer

MERLIN NIMIER-DAVID*, École Polytechnique Fédérale de Lausanne
DELIO VICINI*, École Polytechnique Fédérale de Lausanne
TIZIAN ZELTNER, École Polytechnique Fédérale de Lausanne
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne



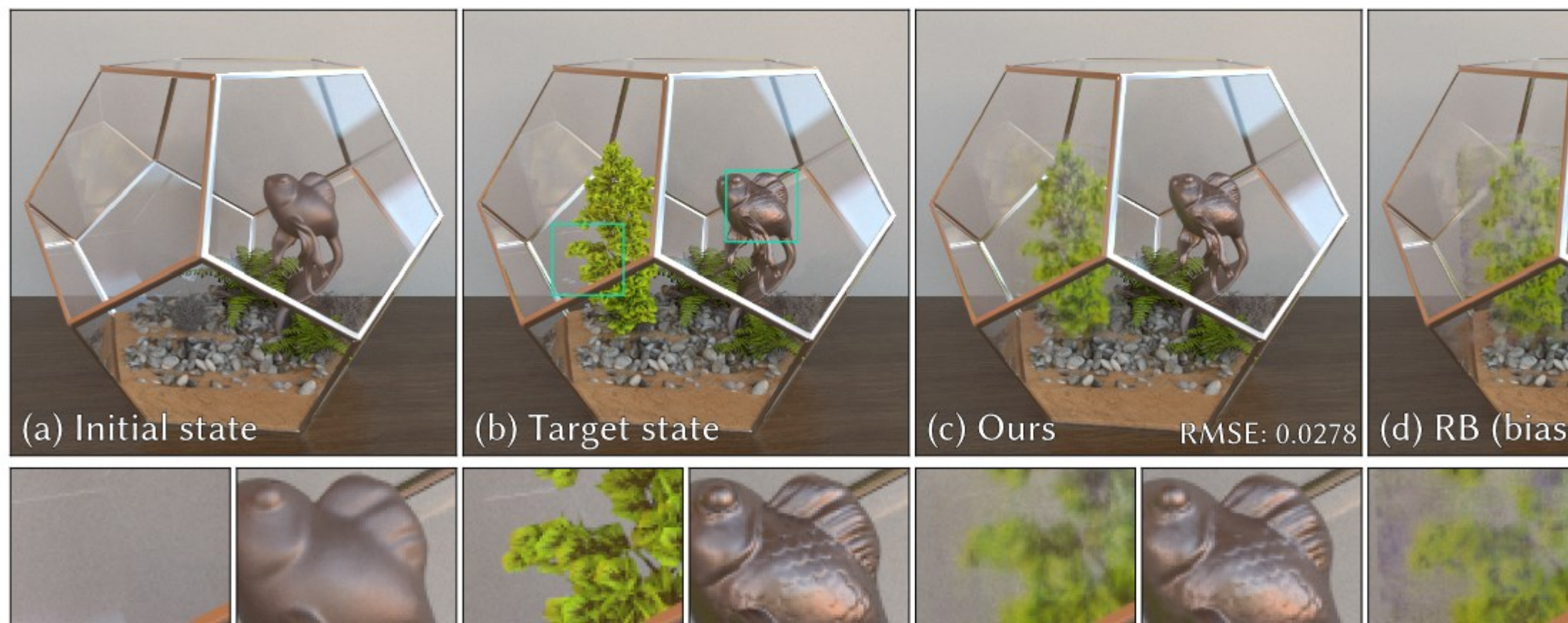
Reparameterizing Discontinuous Integrands for Differentiable Rendering

GUILLAUME LOUBET, École Polytechnique Fédérale de Lausanne (EPFL)
NICOLAS HOLZSCHUCH, Inria, Univ. Grenoble-Alpes, CNRS, LJK
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL)



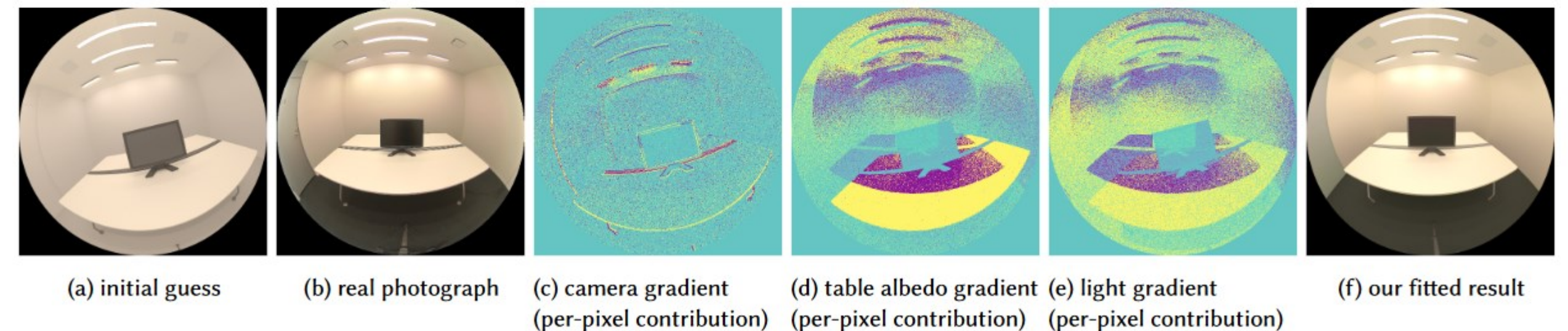
Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time

DELIO VICINI, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
SÉBASTIEN SPEIERER, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
WENZEL JAKOB, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland



Differentiable Monte Carlo Ray Tracing through Edge Sampling

TZU-MAO LI, MIT CSAIL
MIIKA AITTALA, MIT CSAIL
FRÉDO DURAND, MIT CSAIL
JAAKKO LEHTINEN, Aalto University & NVIDIA



Remember: Path Integral for Global Illumination

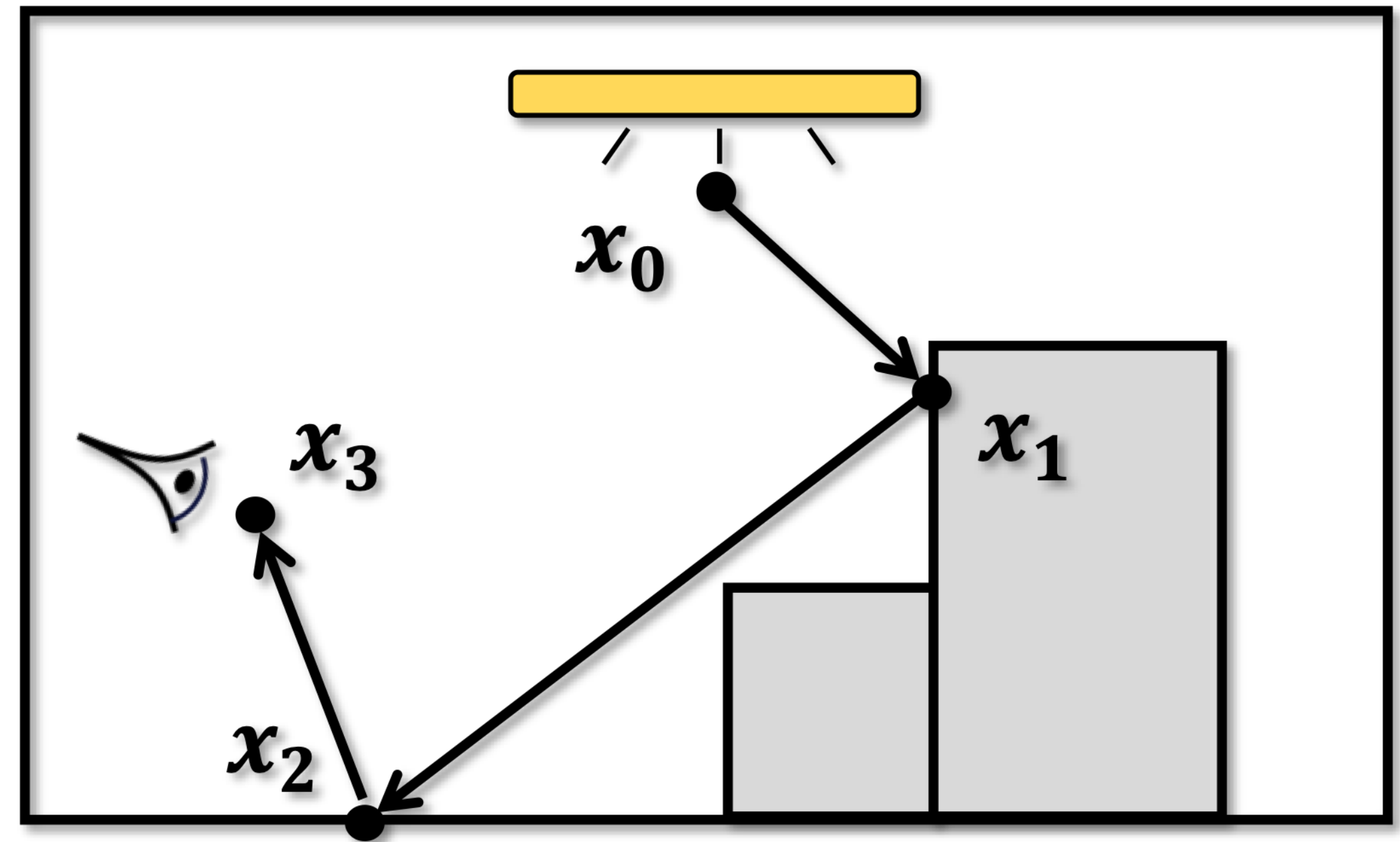
Pixel value

$$I = \int_{\Omega} f(\bar{x}) d\mu(\bar{x})$$

Measurement contribution

Path space

Area-product measure



Light path $\bar{x} = (x_0, x_1, x_2, x_3)$

Differential Path Integral

Path-space differentiable rendering

$$\frac{d}{d\theta} \left(\int_{\Omega} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) \right) = \int_{\Omega} \dot{f}(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) + \int_{\partial\Omega} g(\bar{\mathbf{x}}) d\mu'(\bar{\mathbf{x}})$$

Interior integral **Boundary** integral

Differential Path Integral

Path-space differentiable rendering

$$\frac{d}{d\theta} \left(\int_{\Omega} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) \right) = \int_{\Omega} \dot{f}(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) + \int_{\partial\Omega} g(\bar{\mathbf{x}}) d\mu'(\bar{\mathbf{x}})$$

Interior integral Boundary integral

We now derive $\partial I_N / \partial \pi$ in Eq. (25) using the recursive relations provided by Eqs. (21) and (24). Let

$$h_n^{(0)} := \left[\prod_{n'=n+1}^N g(\mathbf{x}_{n'}; \mathbf{x}_{n'-2}, \mathbf{x}_{n'-1}) \right] W_e(\mathbf{x}_N \rightarrow \mathbf{x}_{N-1}), \quad (52)$$

$$h_n^{(1)} := \sum_{n'=n+1}^N \kappa(\mathbf{x}_{n'}) V(\mathbf{x}_{n'}), \quad (53)$$

$$\Delta h_{n,n'}^{(0)} := h_n^{(0)} \Delta g(\mathbf{x}_{n'}; \mathbf{x}_{n'-2}, \mathbf{x}_{n'-1}) / g(\mathbf{x}_{n'}; \mathbf{x}_{n'-2}, \mathbf{x}_{n'-1}), \quad (54)$$

for $0 \leq n < n' \leq N$. We omit the dependencies of $h_n^{(0)}$, $h_n^{(1)}$, and $\Delta h_{n,n'}^{(0)}$ on $\mathbf{x}_{n+1}, \dots, \mathbf{x}_N$ for notational convenience.

We now show that, for all $0 \leq n < N$, it holds that

$$h_n(\mathbf{x}_n; \mathbf{x}_{n-1}) = \int_{\mathcal{M}^{N-n}} h_n^{(0)} \prod_{n'=n+1}^N dA(\mathbf{x}_{n'}), \quad (55)$$

and

$$\begin{aligned} \dot{h}_n(\mathbf{x}_n; \mathbf{x}_{n-1}) &= \int_{\mathcal{M}^{N-n}} \left[\left(h_n^{(0)} \right)^{\cdot} - h_n^{(0)} h_n^{(1)} \right] \prod_{n'=n+1}^N dA(\mathbf{x}_{n'}) \\ &+ \sum_{n'=n+1}^N \int \Delta h_{n,n'}^{(0)} V_{\partial \mathcal{M}_{n'}}(\mathbf{x}_{n'}) d\ell(\mathbf{x}_{n'}) \prod_{\substack{n < i \leq N \\ i \neq n'}} dA(\mathbf{x}_i), \end{aligned} \quad (56)$$

where the integral domain of the second term on the right-hand side, which is omitted for notational clarity, is $\mathcal{M}(\pi)$ for each \mathbf{x}_i with $i \neq n'$ and $\partial \mathcal{M}_{n'}(\pi)$, which depends on $\mathbf{x}_{n'-1}$, for $\mathbf{x}_{n'}$.

It is easy to verify that Eqs. (55) and (56) hold for $n = N - 1$. We now show that, if they hold for some $0 < n < N$, then it is also the case for $n - 1$. Let $g_{n-1} := g(\mathbf{x}_n; \mathbf{x}_{n-2}, \mathbf{x}_{n-1})$ for all $0 < n \leq N$. Then,

$$\begin{aligned} h_{n-1}(\mathbf{x}_{n-1}; \mathbf{x}_{n-2}) &= \int_{\mathcal{M}} g_{n-1} \int_{\mathcal{M}^{N-n}} h_n^{(0)} \prod_{n'=n+1}^N dA(\mathbf{x}_{n'}) dA(\mathbf{x}_n) \\ &= \int_{\mathcal{M}^{N-n+1}} h_{n-1}^{(0)} \prod_{n'=n}^N dA(\mathbf{x}_{n'}), \end{aligned} \quad (57)$$

and

$$\begin{aligned} \dot{h}_{n-1}(\mathbf{x}_{n-1}; \mathbf{x}_{n-2}) &= \int_{\mathcal{M}} \left[\dot{g}_{n-1} h_n + g_{n-1} (\dot{h}_n - h_n \kappa(\mathbf{x}_n) V(\mathbf{x}_n)) \right] dA(\mathbf{x}_n) \\ &+ \int_{\partial \mathcal{M}_n} \Delta g_{n-1} h_n V_{\partial \mathcal{M}_n} d\ell(\mathbf{x}_n) \\ &= \int_{\mathcal{M}^{N-n+1}} \left\{ \dot{g}_{n-1} h_n^{(0)} + g_{n-1} \left[\left(h_n^{(0)} \right)^{\cdot} - h_n^{(0)} h_n^{(1)} \right] \right\} \prod_{n'=k}^N dA(\mathbf{x}_{n'}) \\ &+ \sum_{n'=n+1}^N \int g_{n-1} \Delta h_{n,n'}^{(0)} V_{\partial \mathcal{M}_{n'}}(\mathbf{x}_{n'}) d\ell(\mathbf{x}_{n'}) \prod_{\substack{n \leq i \leq N \\ i \neq n'}} dA(\mathbf{x}_i) \\ &+ \int \Delta g_{n-1} h_n^{(0)} V_{\partial \mathcal{M}_n} d\ell(\mathbf{x}_n) \prod_{n'=n+1}^N dA(\mathbf{x}_{n'}) \\ &= \int_{\mathcal{M}^{N-n+1}} \left[\left(h_{n-1}^{(0)} \right)^{\cdot} - h_{n-1}^{(0)} h_{n-1}^{(1)} \right] \prod_{n'=n}^N dA(\mathbf{x}_{n'}) \\ &+ \sum_{n'=n}^N \int \Delta h_{n-1,n'}^{(0)} V_{\partial \mathcal{M}_{n'}}(\mathbf{x}_{n'}) d\ell(\mathbf{x}_{n'}) \prod_{\substack{n \leq i \leq N \\ i \neq n'}} dA(\mathbf{x}_i). \end{aligned} \quad (58)$$

Thus, using mathematical induction, we know that Eqs. (55) and (56) hold for all $0 \leq n < N$.

Notice that $h_0^{(0)} = f$ and $\Delta h_{0,n'}^{(0)} = \Delta f_{n'}$, where $\Delta f_{n'}$ follows the definition in Eq. (28). Letting $n = 0$ in Eq. (56) yields

$$\begin{aligned} \dot{h}_0(\mathbf{x}_0) &= \int_{\mathcal{M}^N} \left[\dot{f}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}}) \sum_{n'=1}^N \kappa(\mathbf{x}_{n'}) V(\mathbf{x}_{n'}) \right] \prod_{n'=1}^N dA(\mathbf{x}_{n'}) \\ &+ \sum_{n'=1}^N \int \Delta f_{n'}(\bar{\mathbf{x}}) V_{\partial \mathcal{M}_{n'}} d\ell(\mathbf{x}_{n'}) \prod_{\substack{0 < i \leq N \\ i \neq n'}} dA(\mathbf{x}_i). \end{aligned} \quad (59)$$

Lastly, based on the assumption that h_0 is continuous in \mathbf{x}_0 , Eq. (25) can be obtained by differentiating Eq. (23):

$$\begin{aligned} \frac{\partial I_N}{\partial \pi} &= \frac{\partial}{\partial \pi} \int_{\mathcal{M}} h_0(\mathbf{x}_0) dA(\mathbf{x}_0) \\ &= \int_{\mathcal{M}} \left[\dot{h}_0(\mathbf{x}_0) - h_0(\mathbf{x}_0) \kappa(\mathbf{x}_0) V(\mathbf{x}_0) \right] dA(\mathbf{x}_0) \\ &+ \int_{\partial \mathcal{M}_0} h_0(\mathbf{x}_0) V_{\partial \mathcal{M}_0}(\mathbf{x}_0) d\ell(\mathbf{x}_0) \\ &= \int_{\Omega_N} \left[\dot{f}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}}) \sum_{K=0}^N \kappa(\mathbf{x}_K) V(\mathbf{x}_K) \right] d\mu(\bar{\mathbf{x}}) \\ &+ \sum_{K=0}^N \int_{\Omega_{N,K}} \Delta f_K(\bar{\mathbf{x}}) V_{\partial \mathcal{M}_K} d\mu'_{N,K}(\bar{\mathbf{x}}). \end{aligned} \quad (60)$$

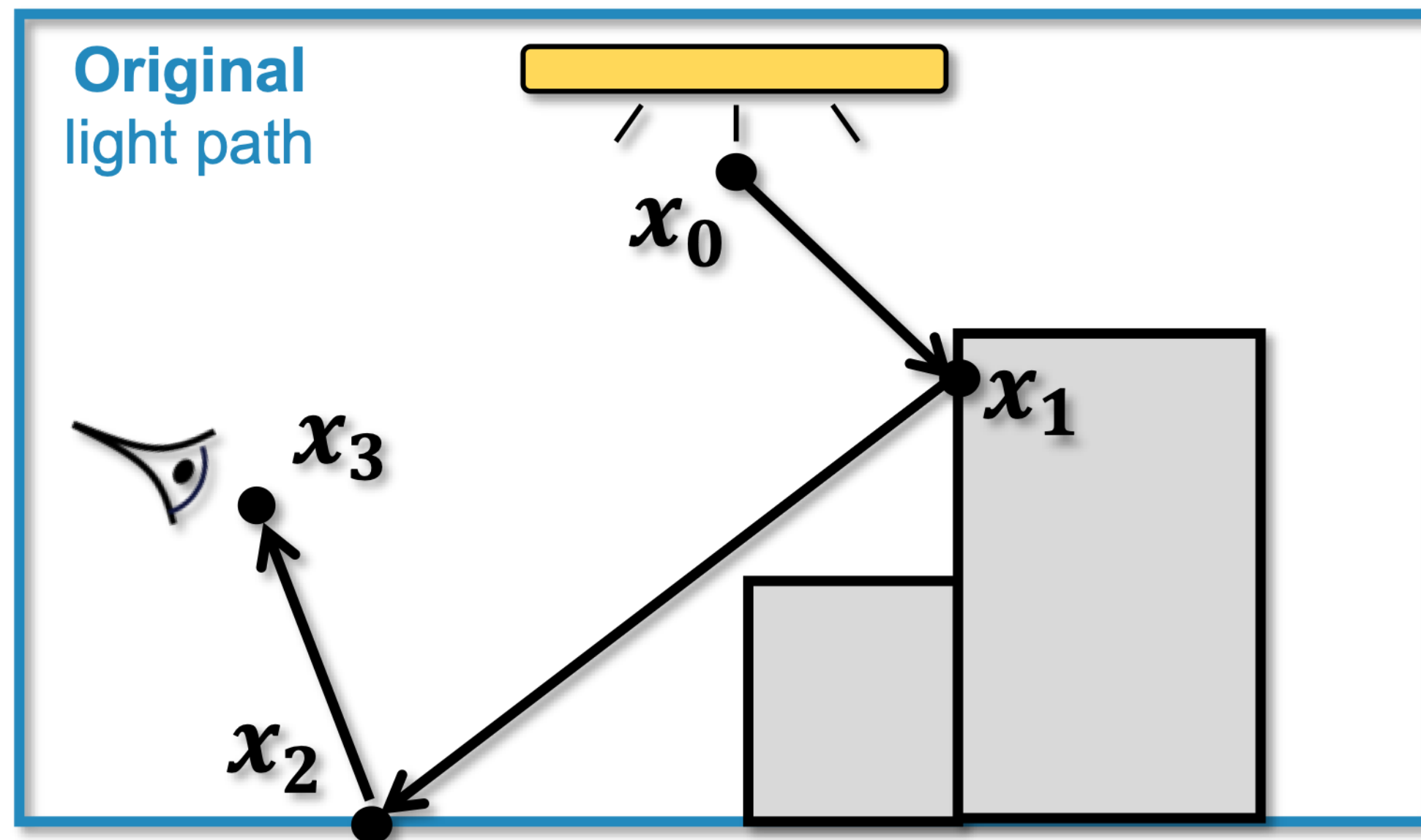
(The full derivation is quite involved...)

Differential Path Integral

Path-space differentiable rendering

$$\frac{d}{d\theta} \left(\int_{\Omega} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) \right) = \int_{\Omega} \dot{f}(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) + \int_{\partial\Omega} g(\bar{\mathbf{x}}) d\mu'(\bar{\mathbf{x}})$$

Interior integral



Interior integral

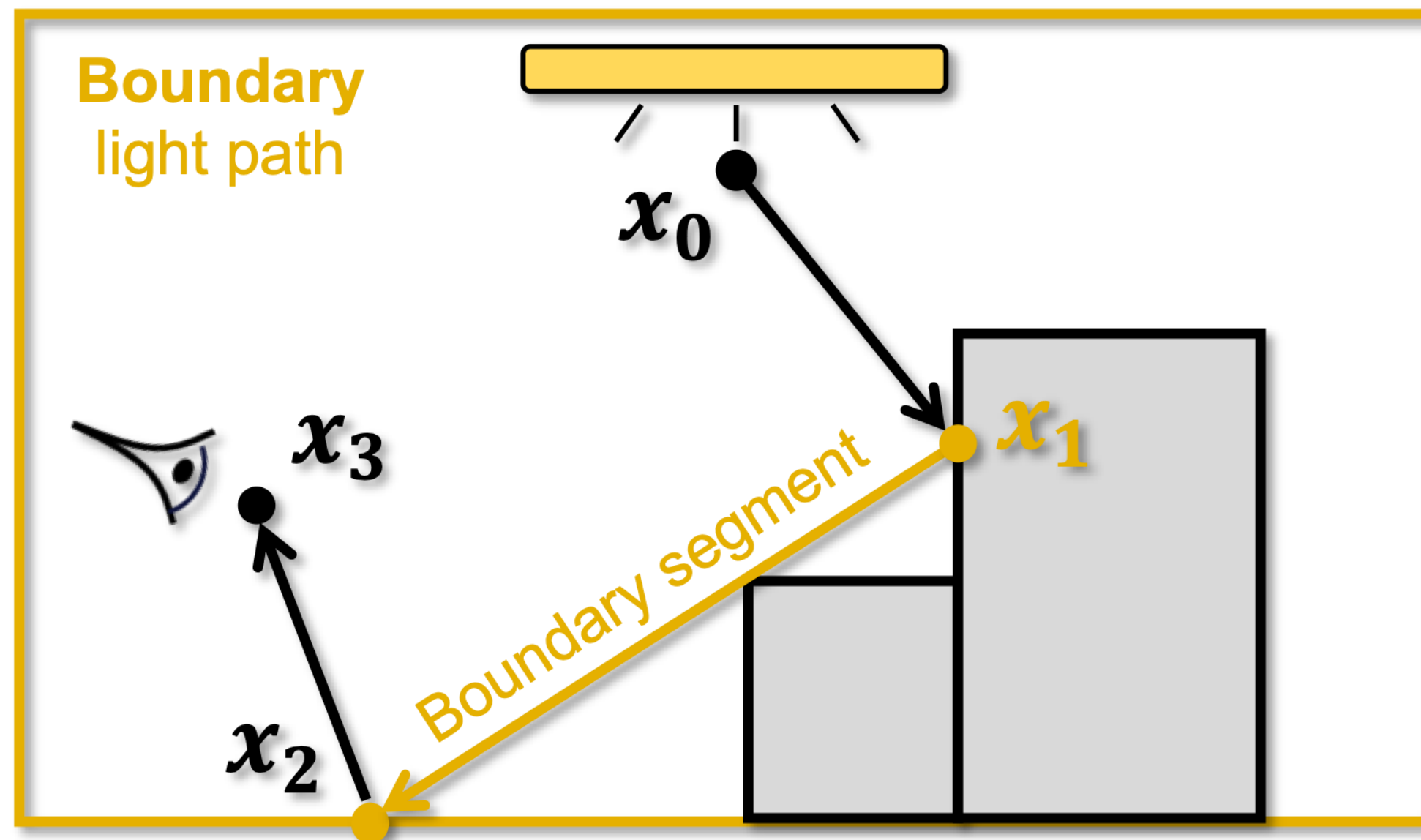
- Defined on the ordinary path space Ω
- The integrand f can be obtained by differentiating the ordinary *measurement contribution function* f

Differential Path Integral

Path-space differentiable rendering

$$\frac{d}{d\theta} \left(\int_{\Omega} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) \right) = \int_{\Omega} \dot{f}(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) + \int_{\partial\Omega} g(\bar{\mathbf{x}}) d\mu'(\bar{\mathbf{x}})$$

Boundary integral



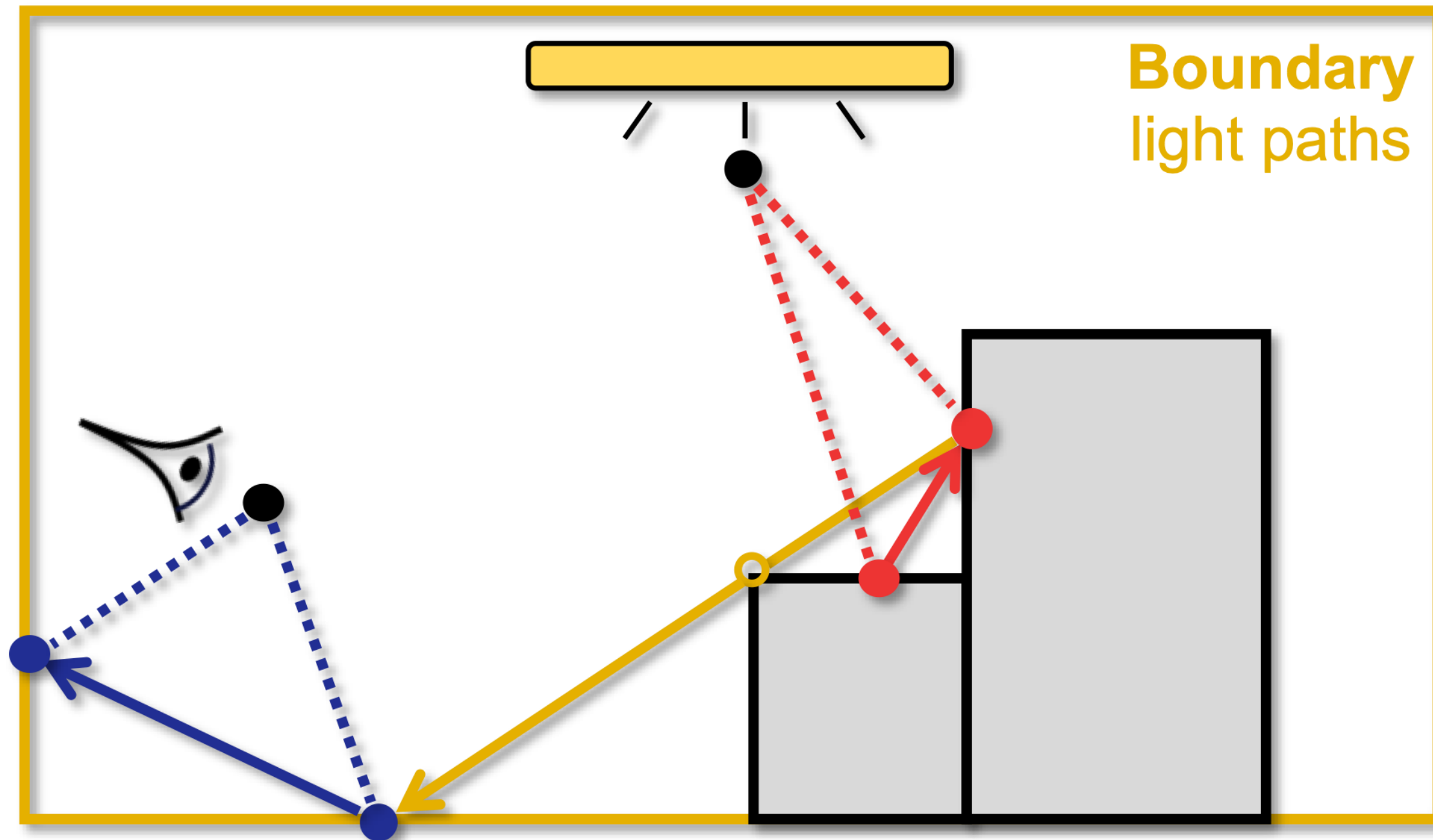
Boundary integral

- Defined on the boundary path space $\partial\Omega$
- A **boundary** light path is the same as an **original one** except having exactly one **boundary segment**

Path-Space Differentiable Path Tracing

Unidirectional estimator

- **Interior**: *unidirectional* path tracing
- **Boundary**: *unidirectional* sampling of subpaths

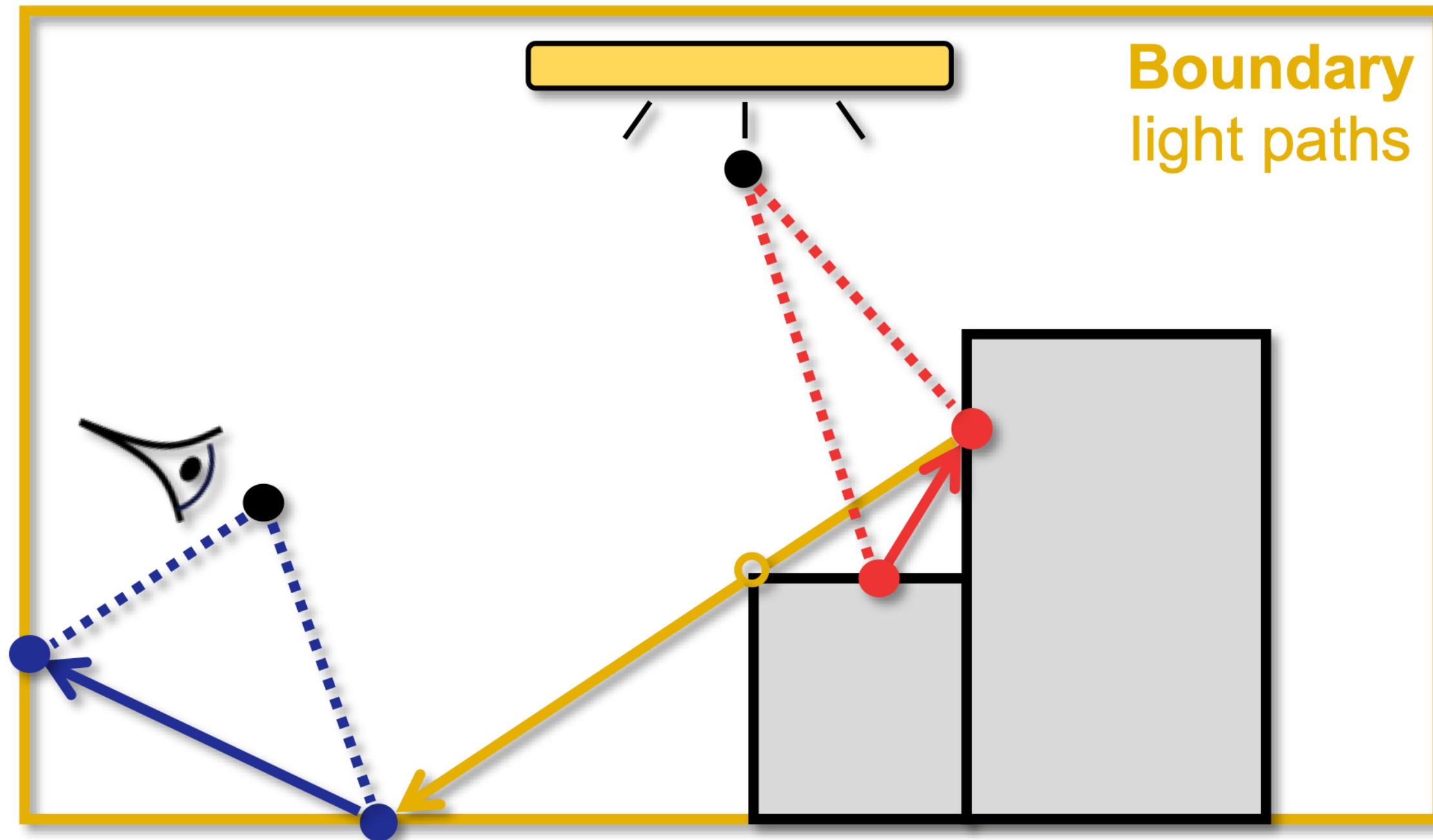


Unidirectional path tracing + NEE

Path-Space Differentiable Path Tracing

Unidirectional estimator

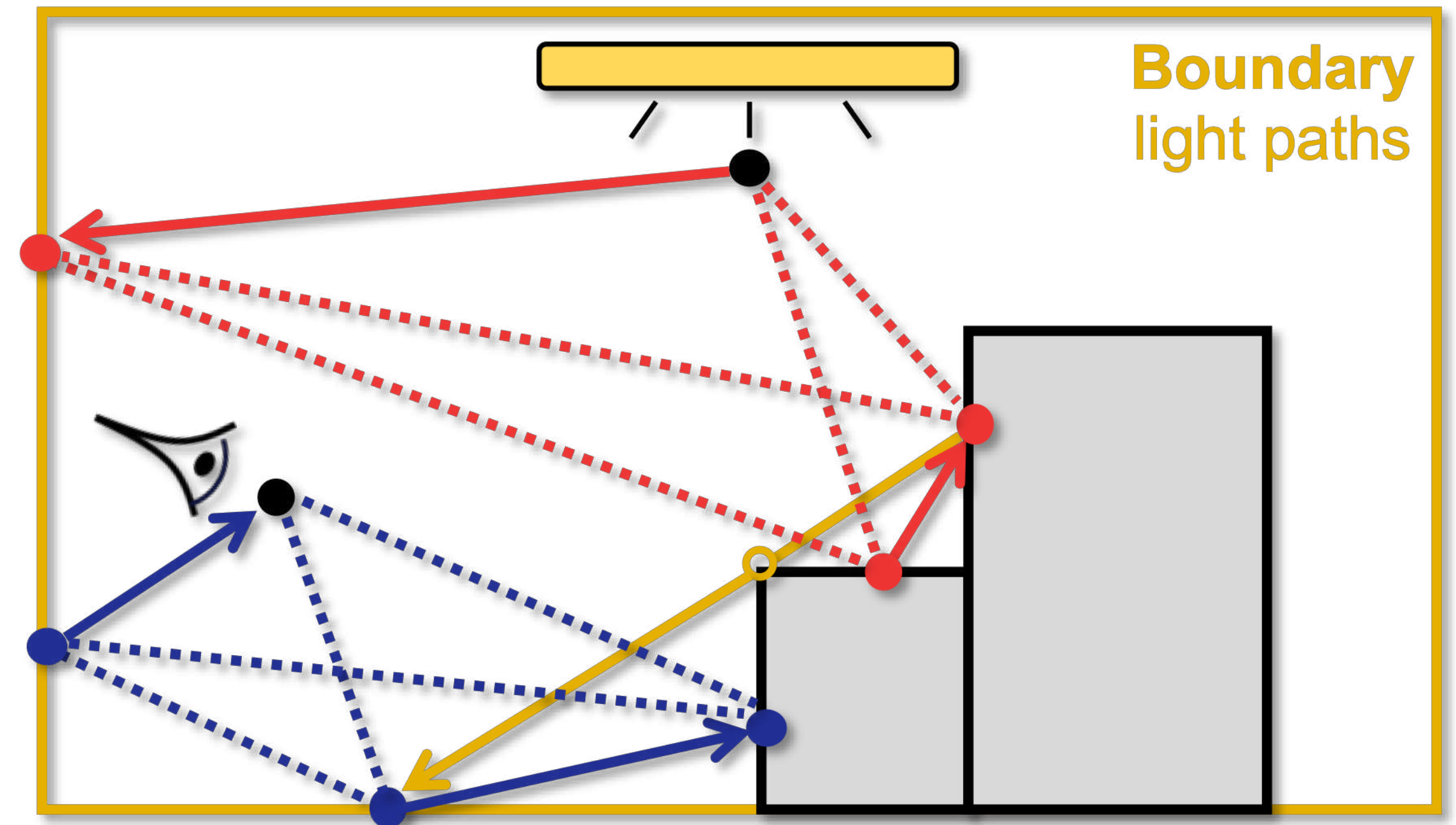
- **Interior**: *unidirectional* path tracing
- **Boundary**: *unidirectional* sampling of subpaths



Unidirectional path tracing + NEE

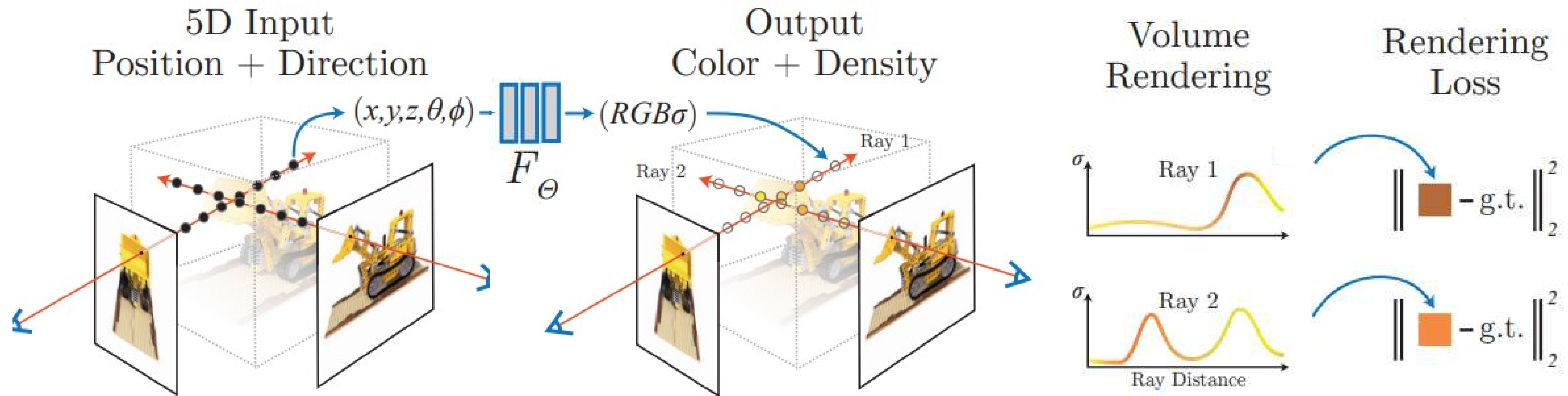
Bidirectional estimator

- **Interior**: *bidirectional* path tracing
- **Boundary**: *bidirectional* sampling of subpaths



Bidirectional path tracing

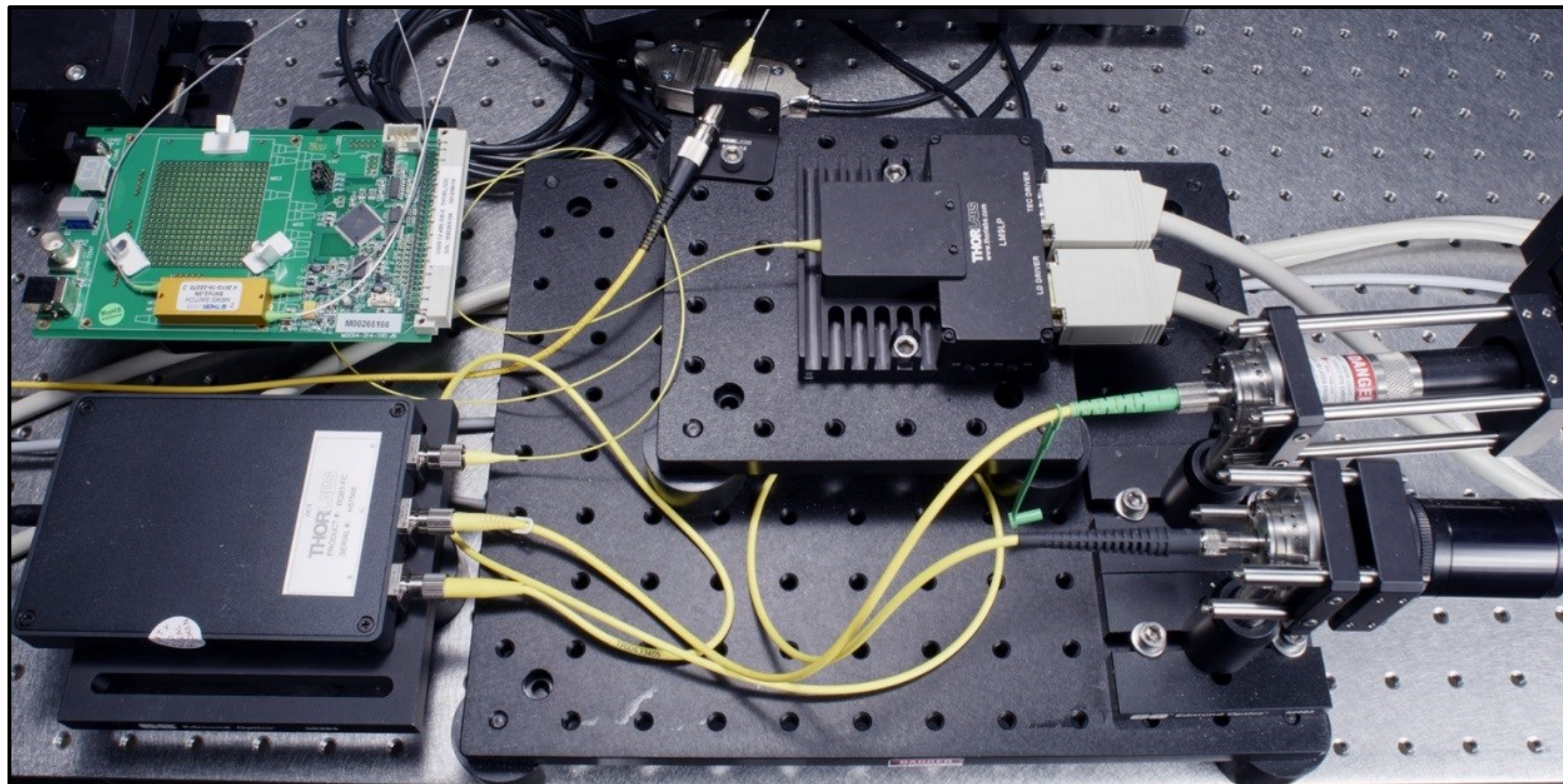
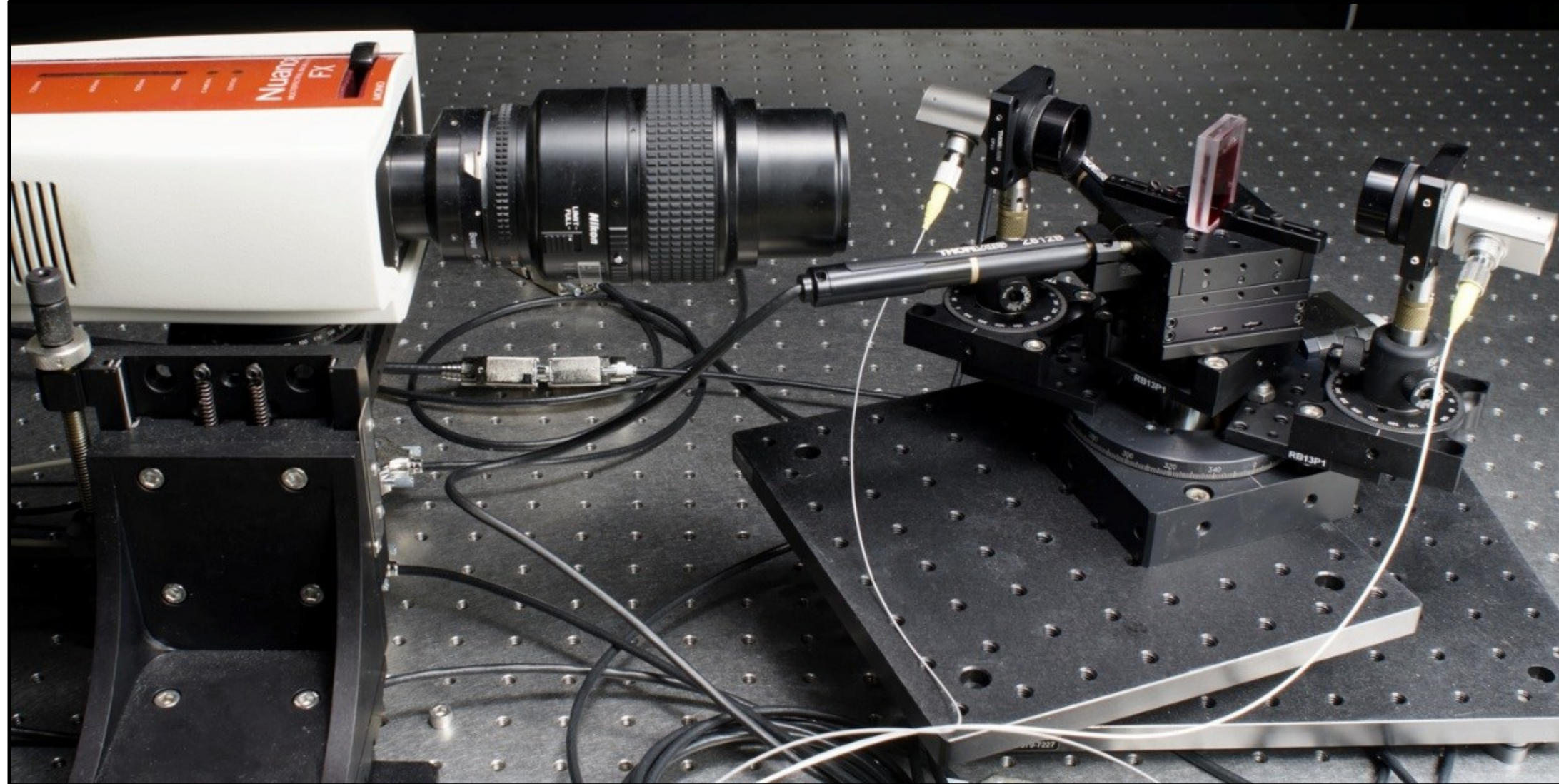
Application: neural rendering



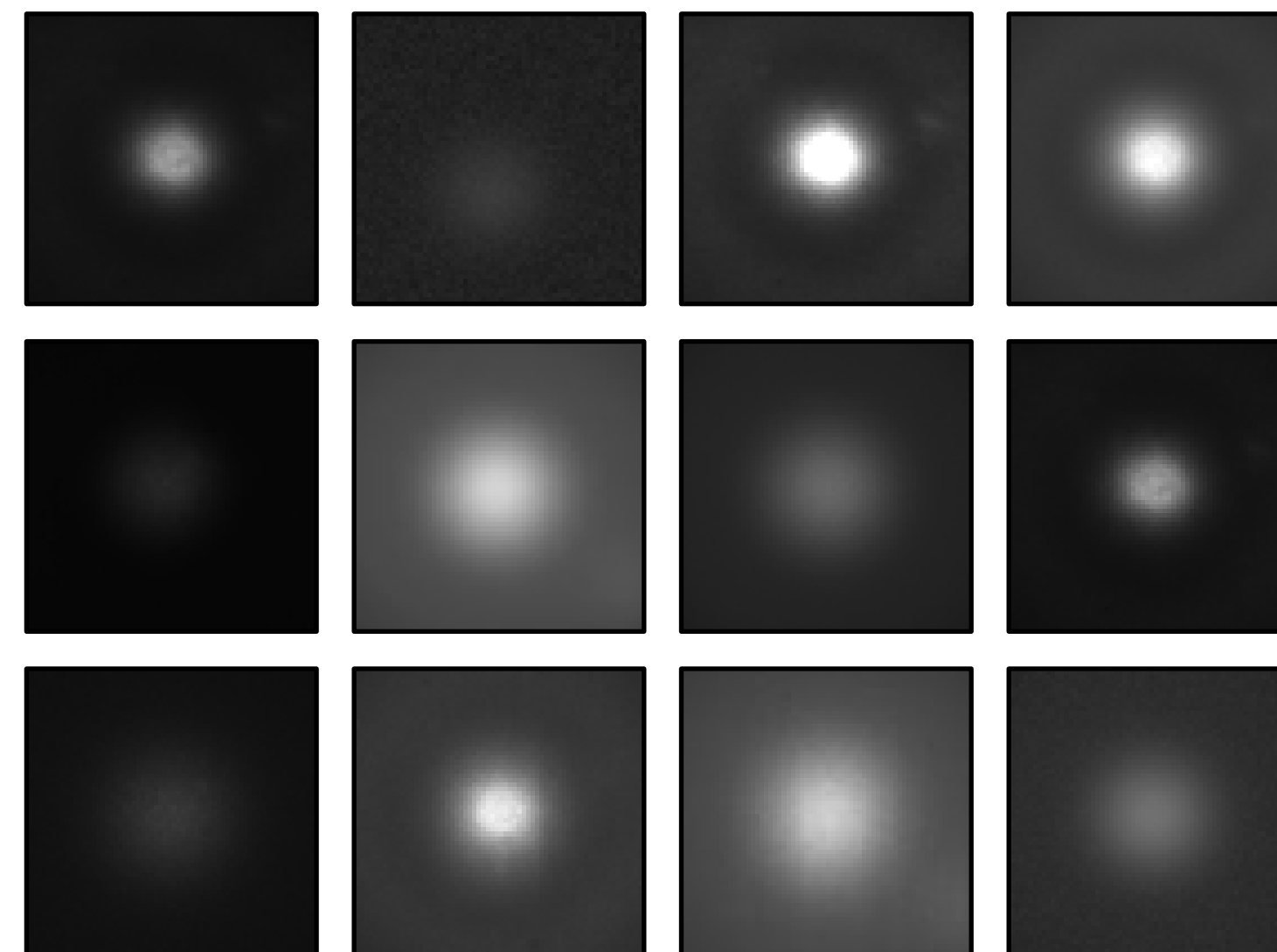
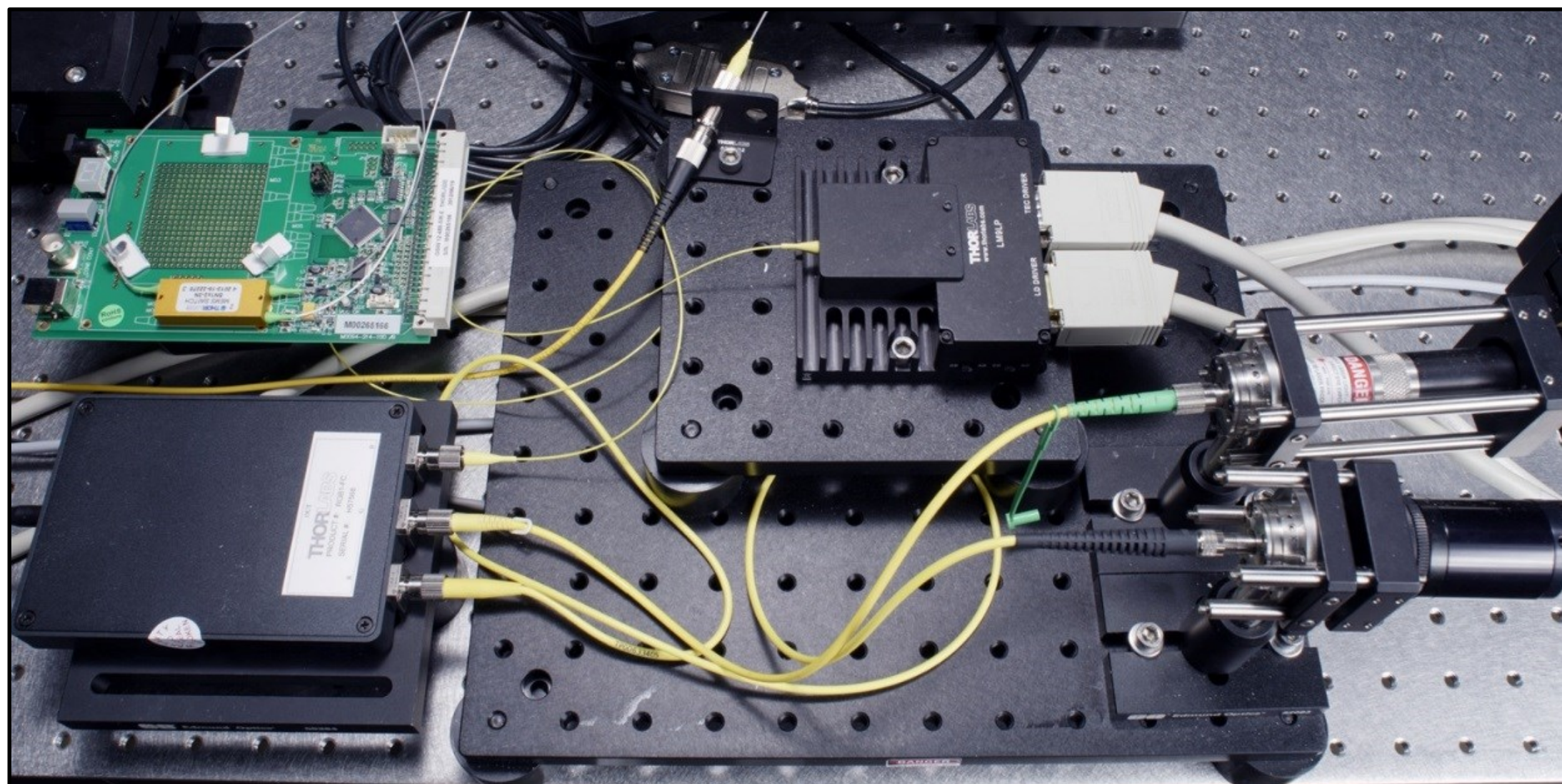
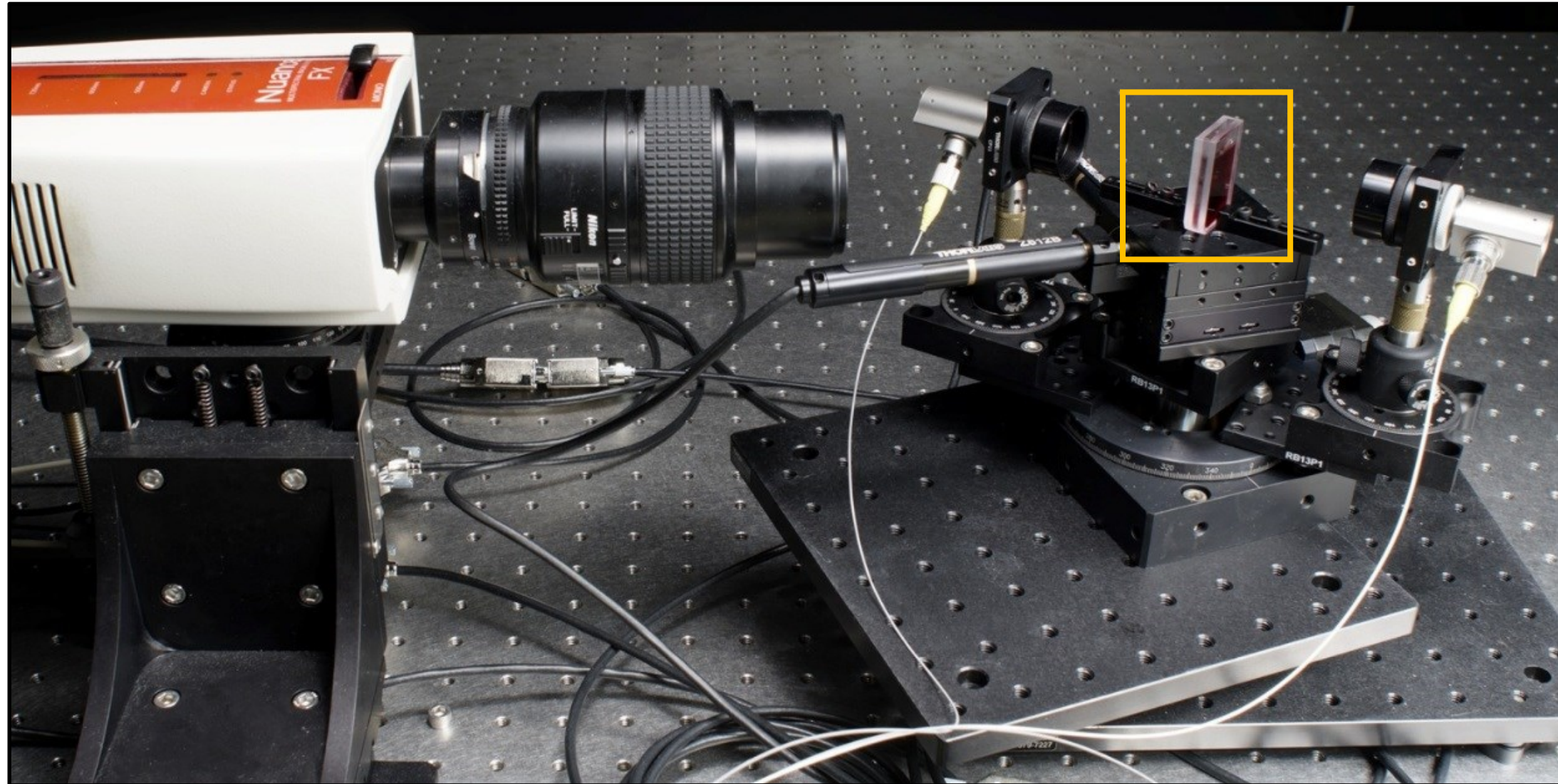
Acquisition of scattering materials



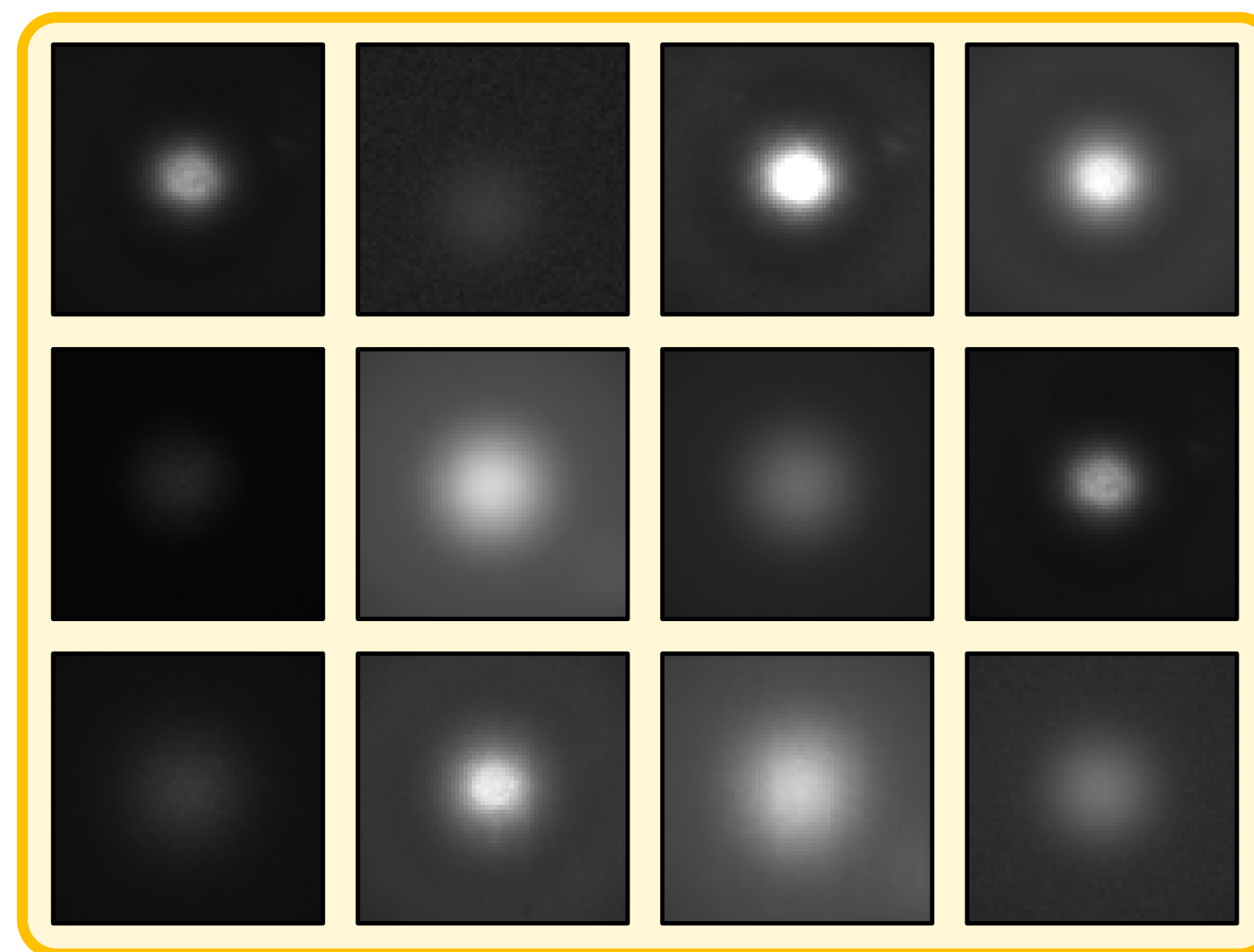
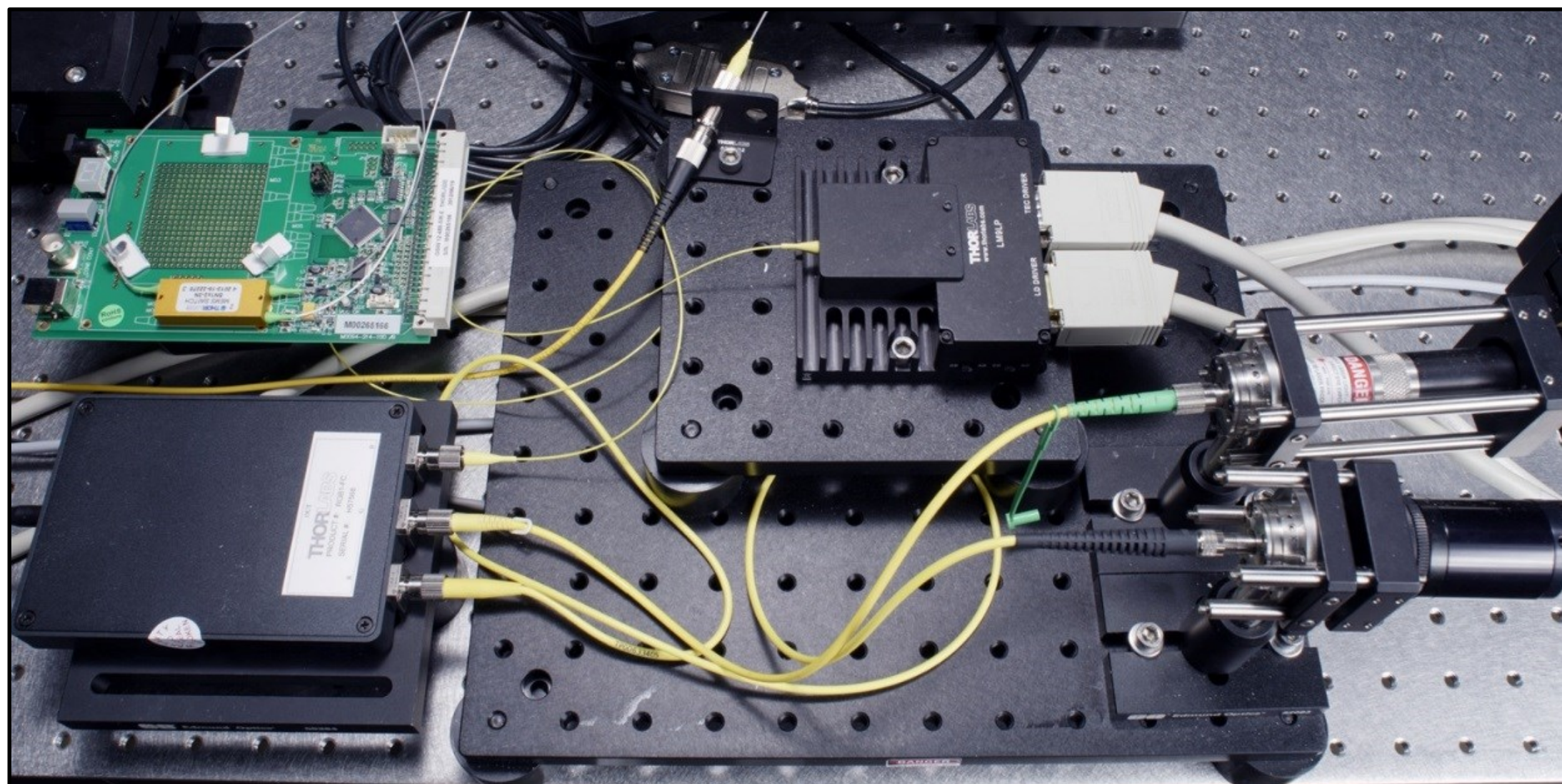
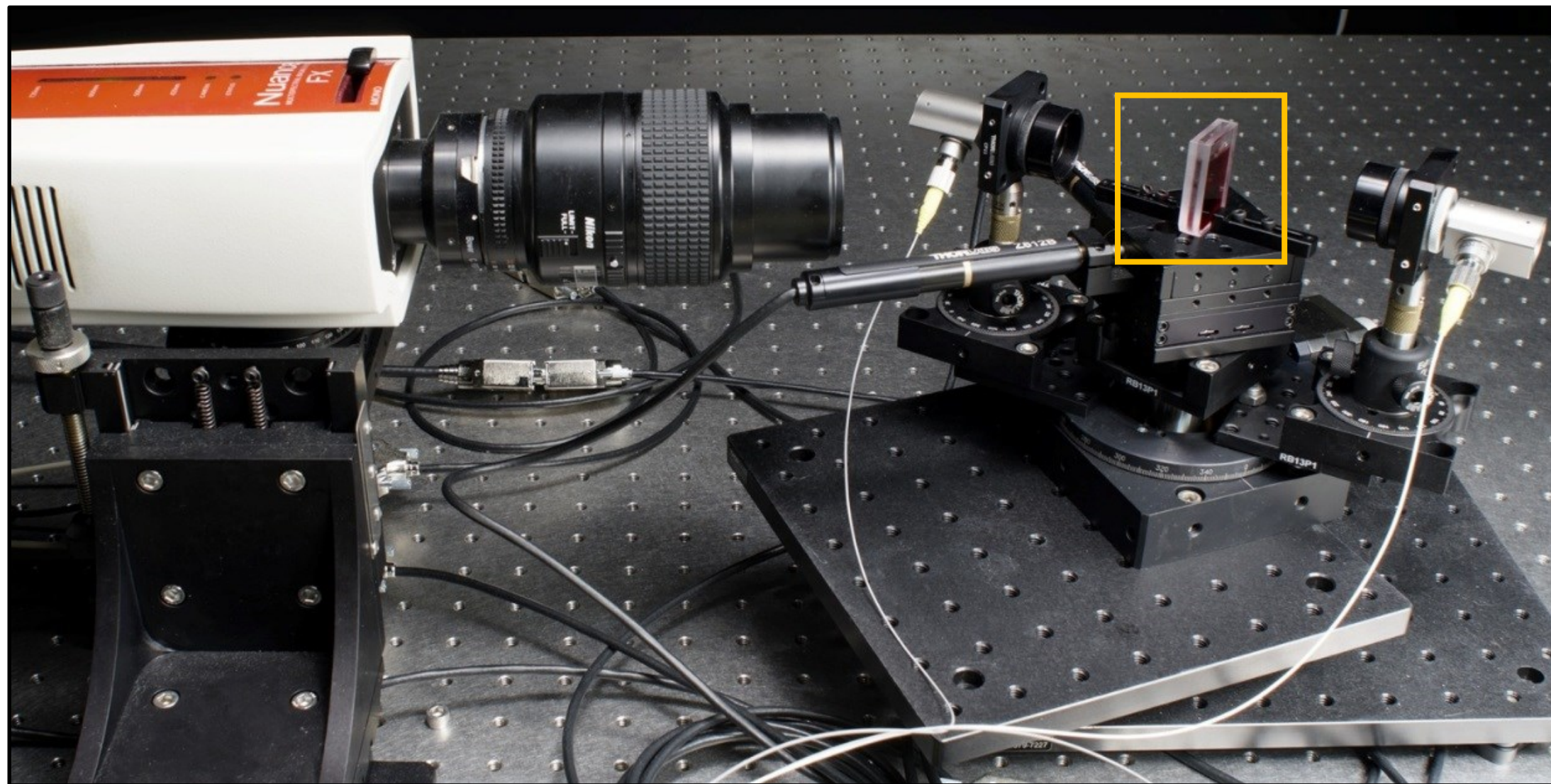
Acquisition setup



Acquisition setup



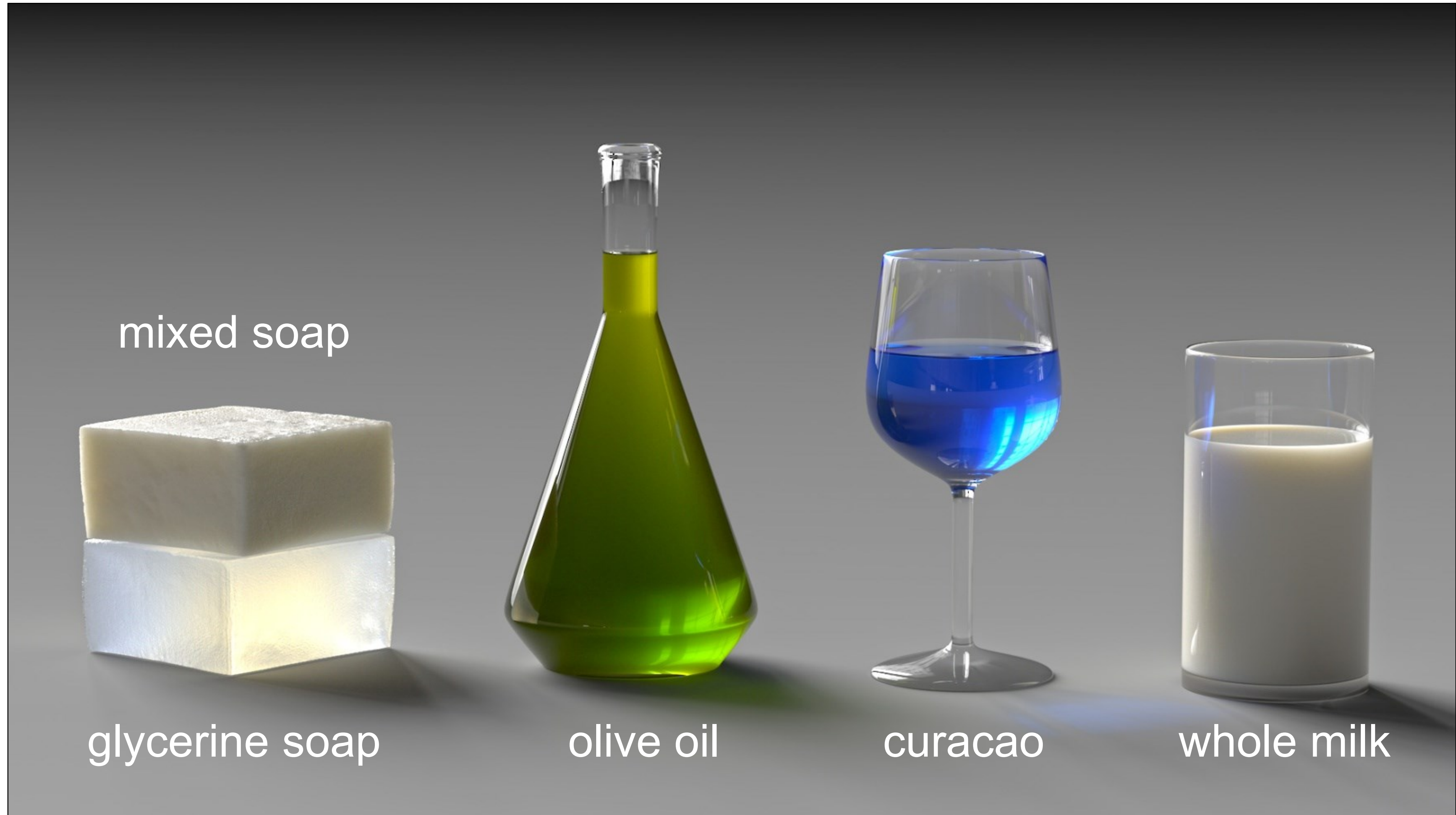
Acquisition setup



Invert using
differentiable
rendering

[Gkioulekas et al., 2013]

Synthetic renderings



mixed soap

glycerine soap



olive oil



curacao

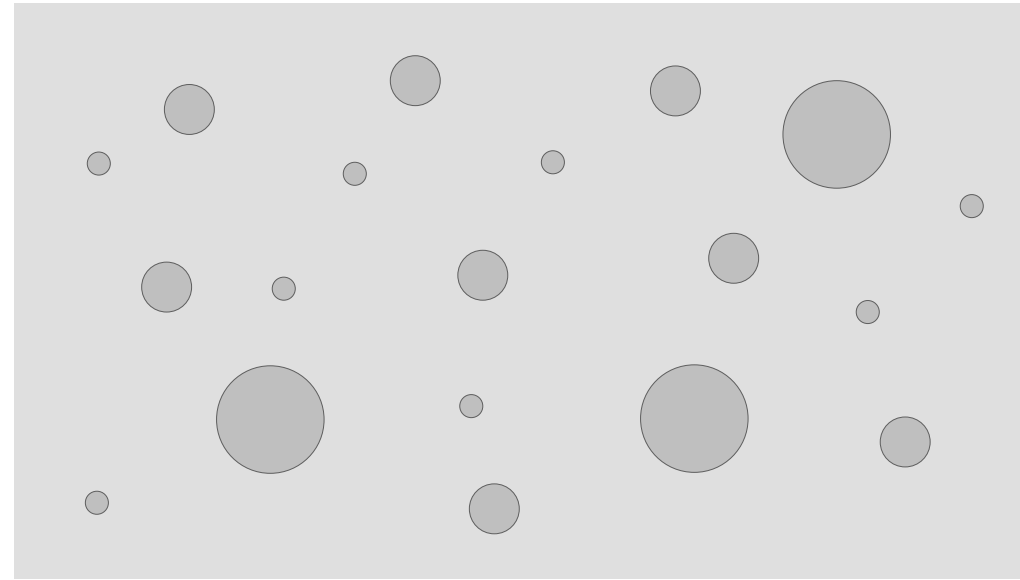


whole milk

Particle sizing of industrial nanodispersions

Particle sizing of industrial nanodispersions

unknown nanodispersion

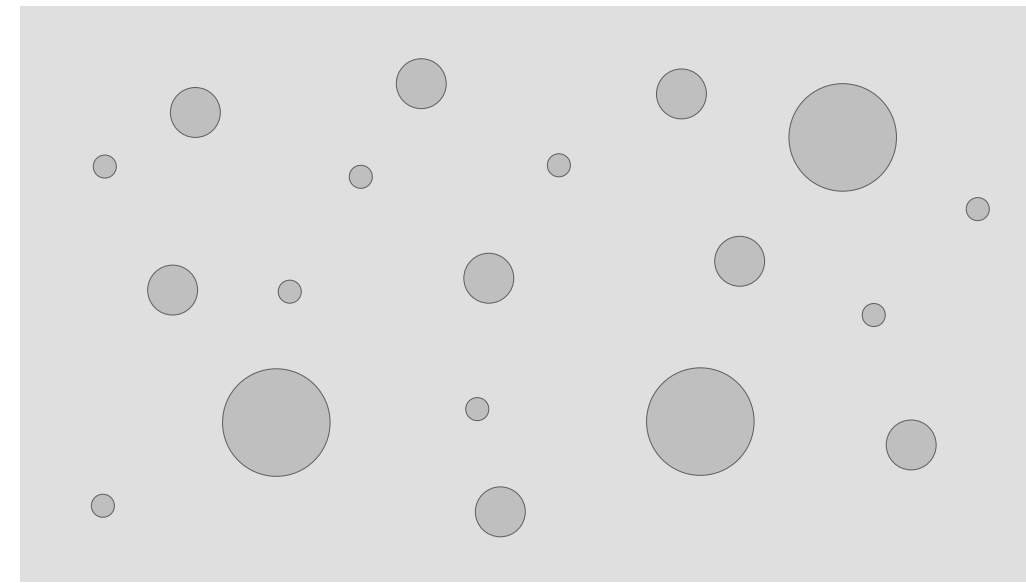


■ dispersing medium

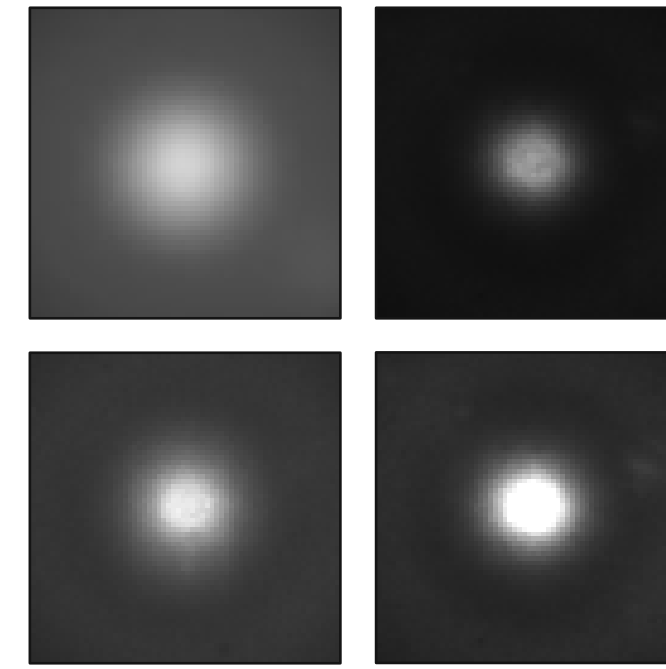
● particle material

Particle sizing of industrial nanodispersions

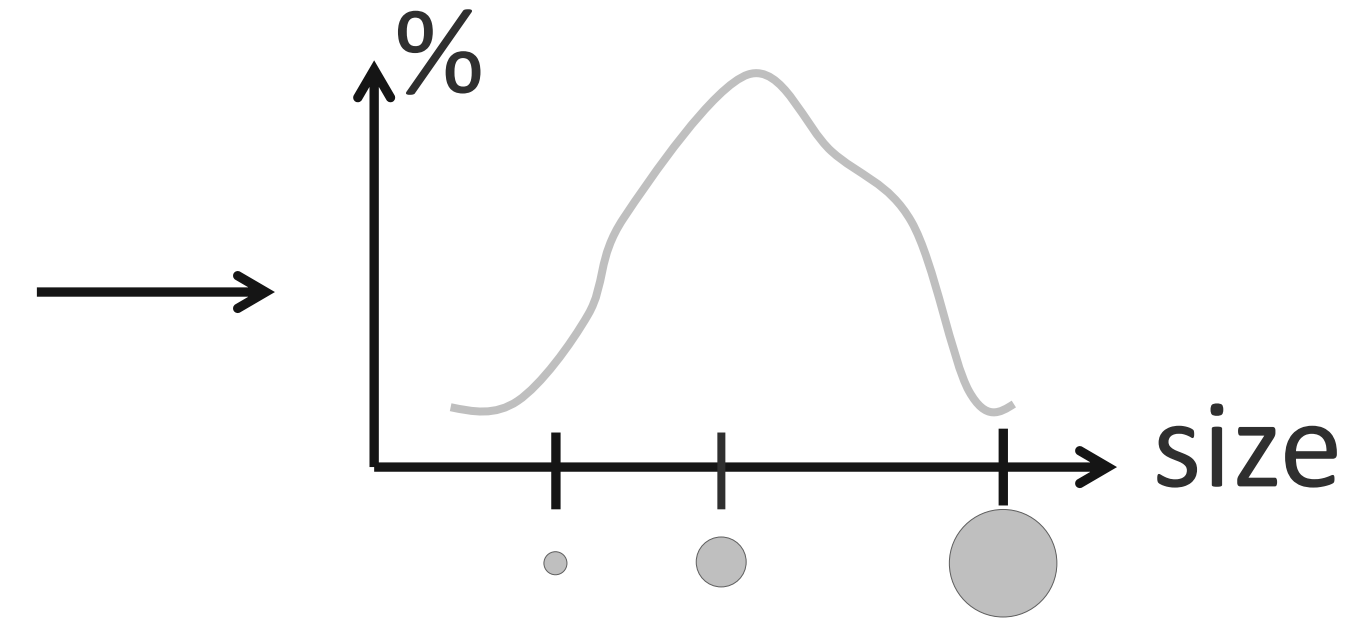
unknown nanodispersion



- dispersing medium
- particle material

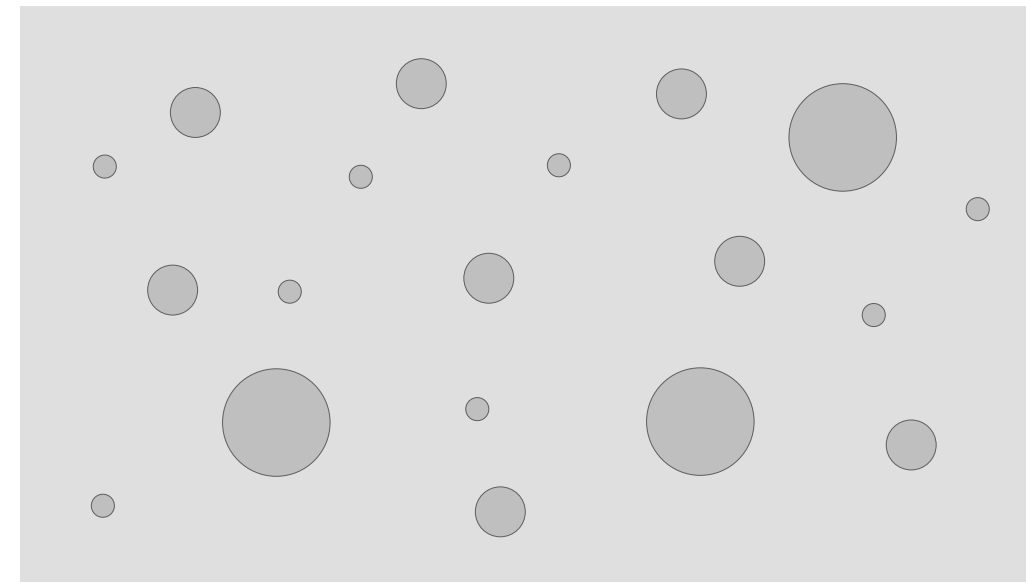


measurements

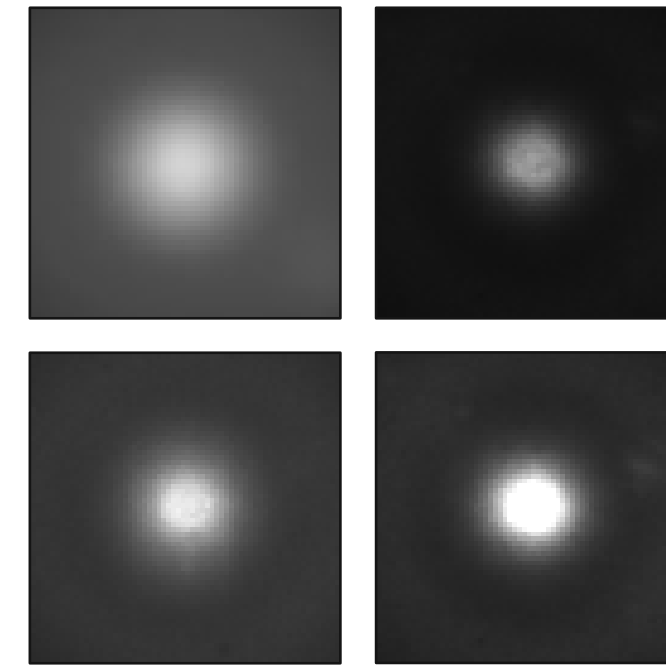


Particle sizing of industrial nanodispersions

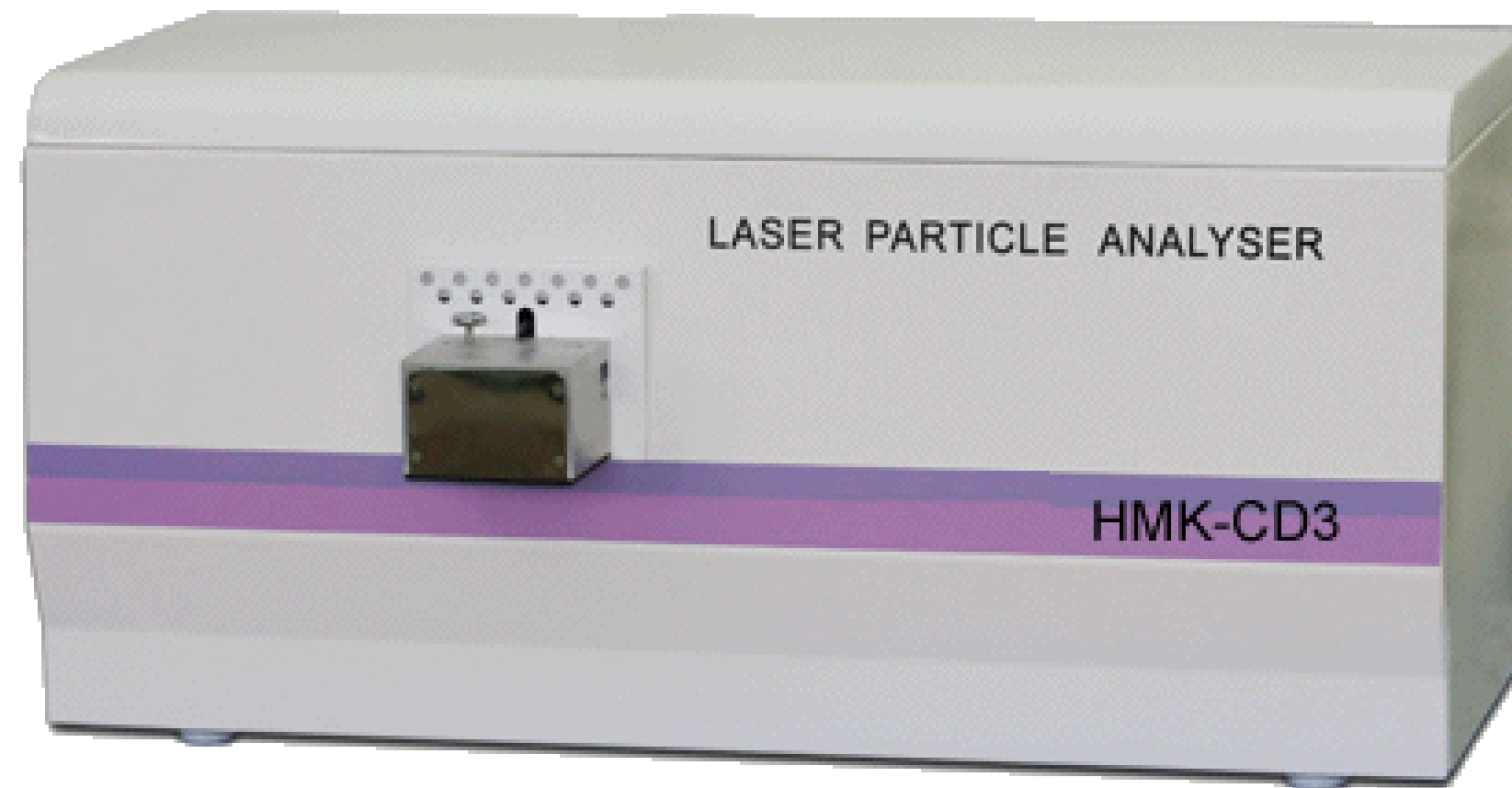
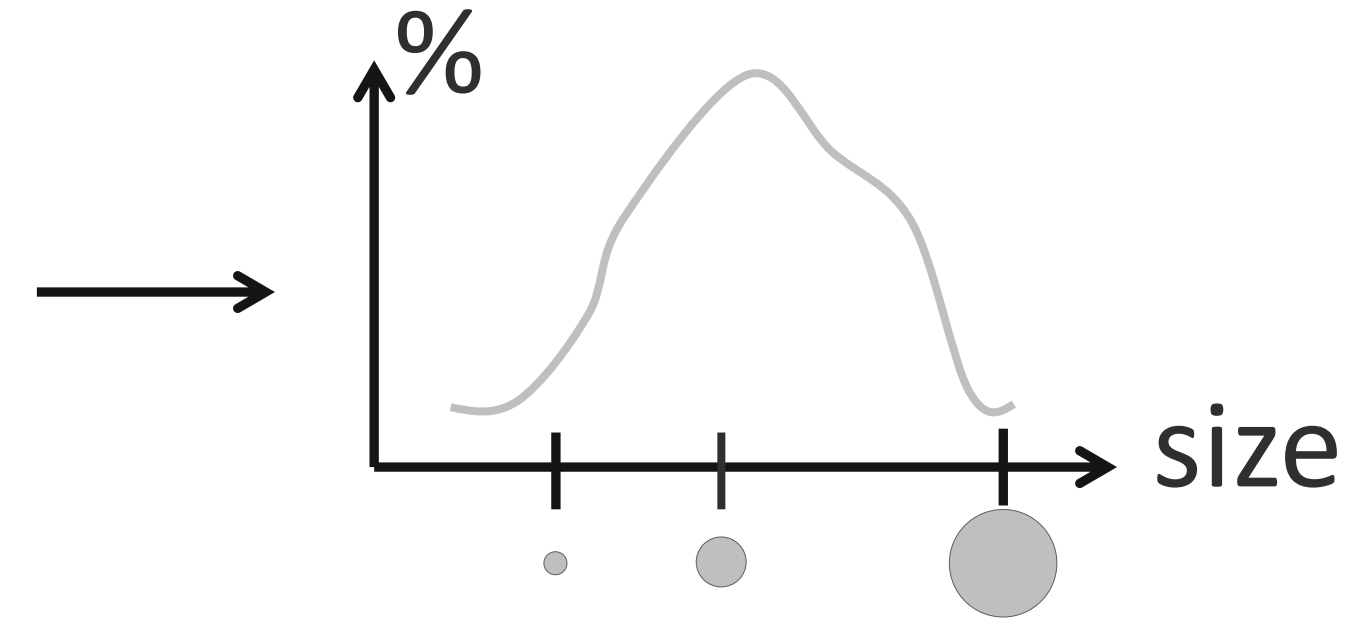
unknown nanodispersion



- dispersing medium
- particle material



measurements



Particle sizing of industrial nanodispersions



polystyrene



aluminum oxide

very precise dispersions (NIST Traceable Standards)

Particle sizing of industrial nanodispersions

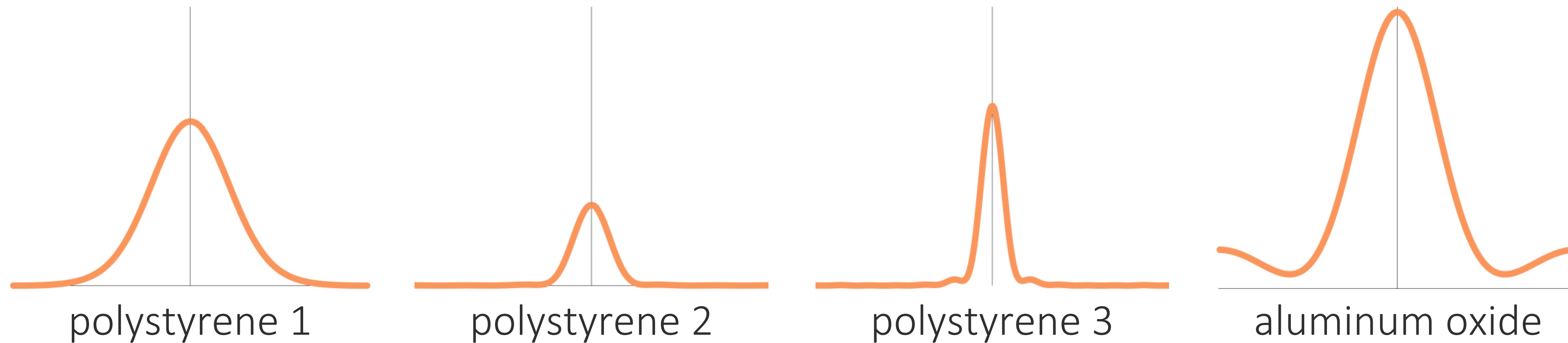


polystyrene



aluminum oxide

very precise dispersions (NIST Traceable Standards)



— ground-truth

Particle sizing of industrial nanodispersions

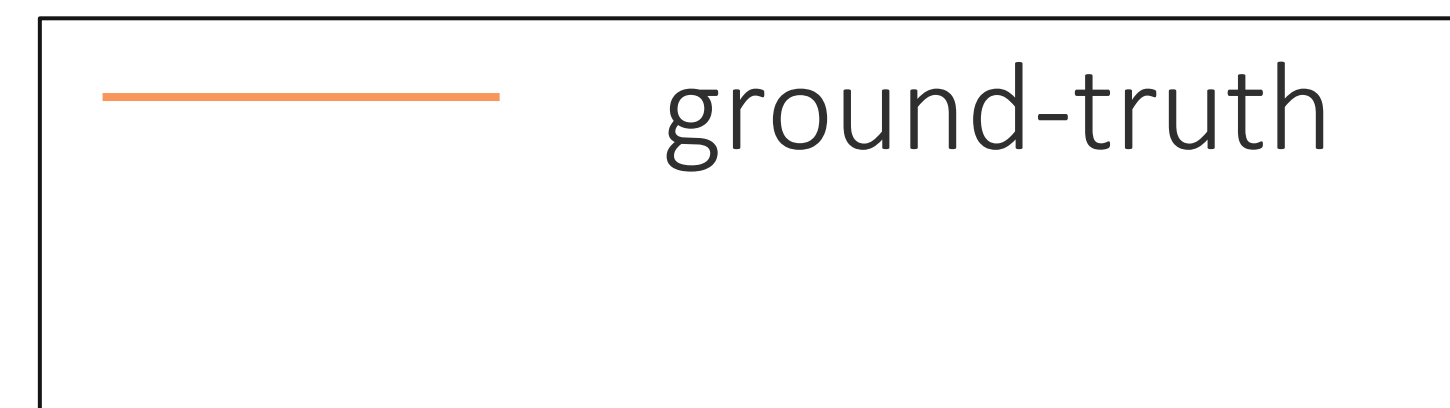
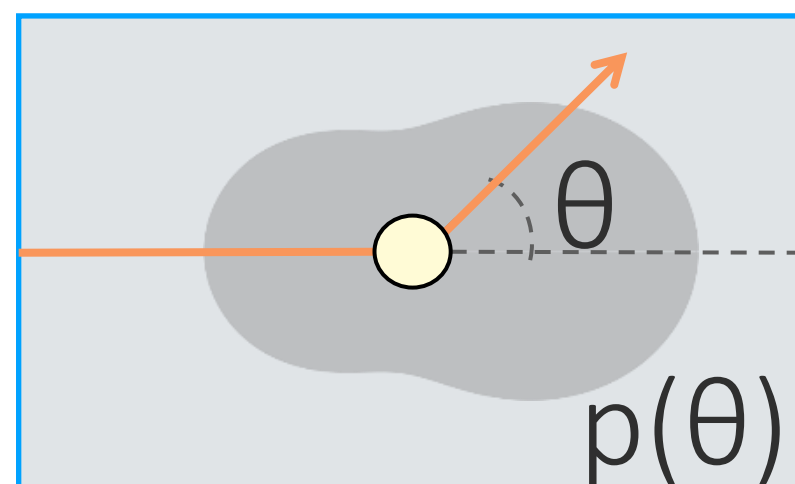
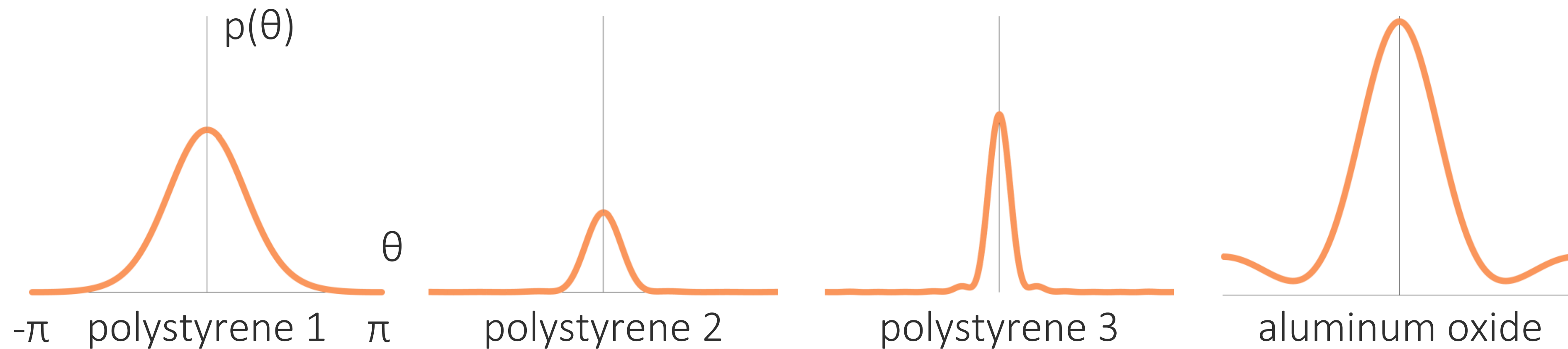


polystyrene



aluminum oxide

very precise dispersions (NIST Traceable Standards)



Particle sizing of industrial nanodispersions

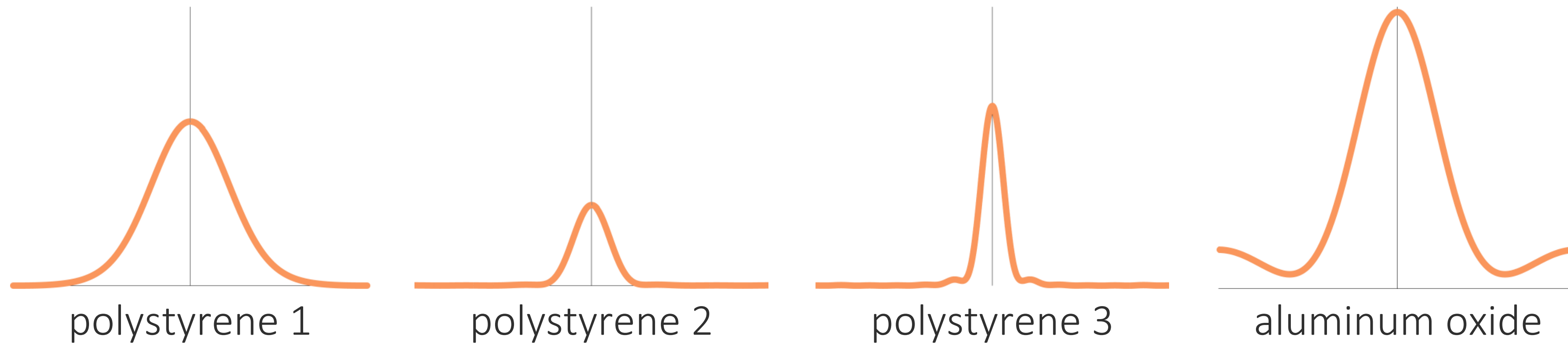


polystyrene



aluminum oxide

very precise dispersions (NIST Traceable Standards)



— ground-truth

Particle sizing of industrial nanodispersions

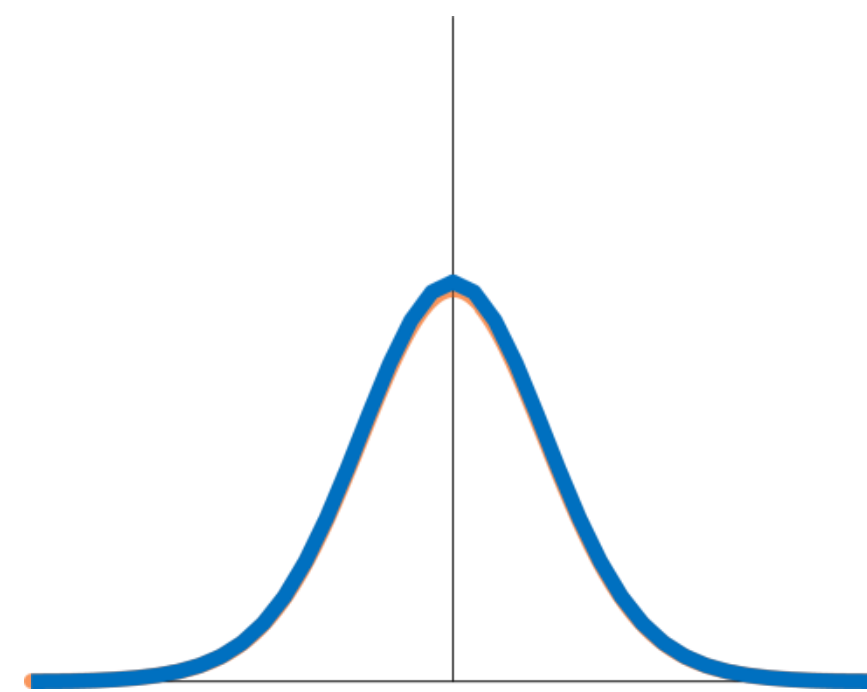


polystyrene

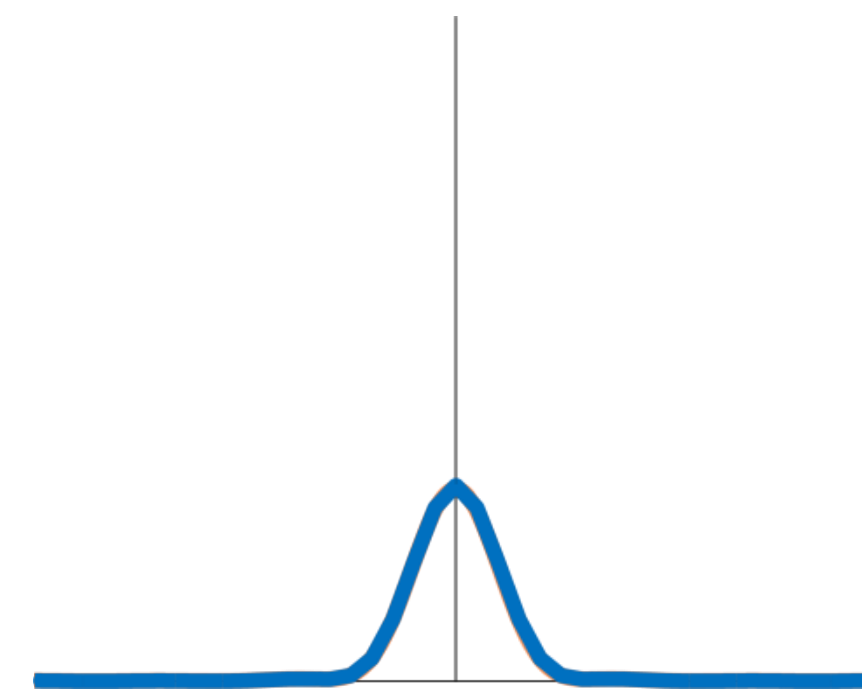


aluminum oxide

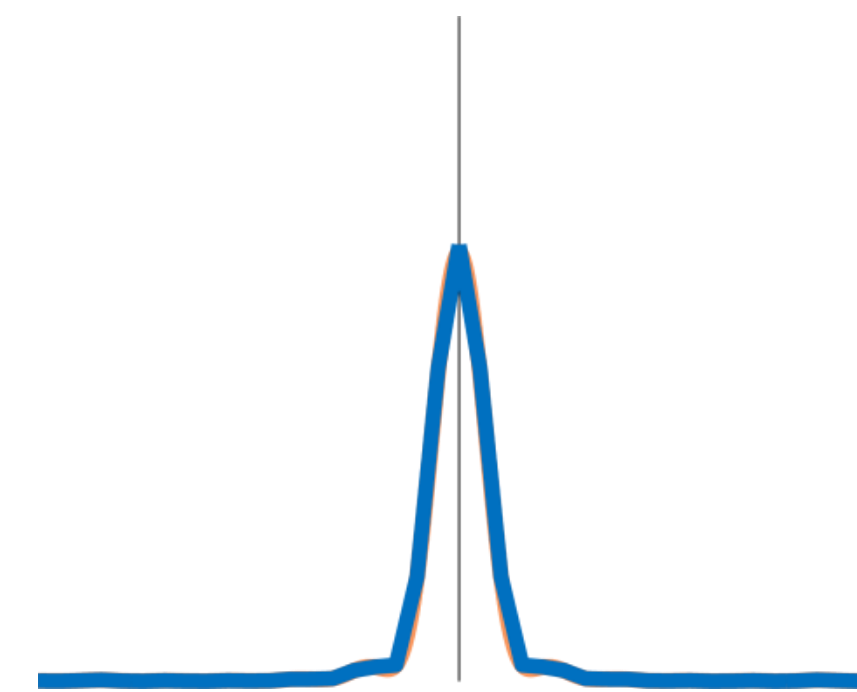
very precise dispersions (NIST Traceable Standards)



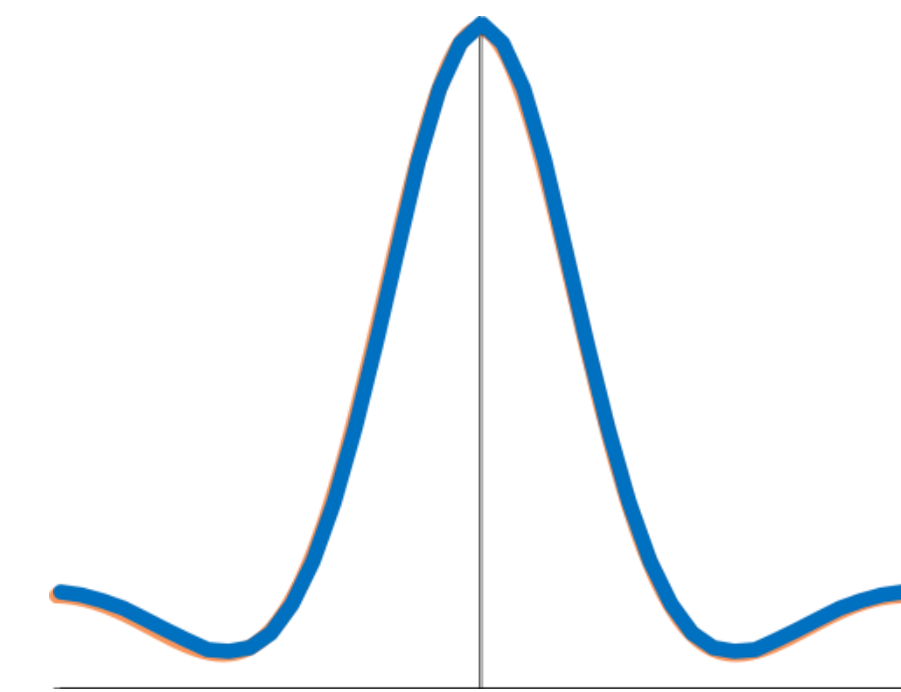
polystyrene 1



polystyrene 2



polystyrene 3



aluminum oxide



Particle sizing of industrial nanodispersions

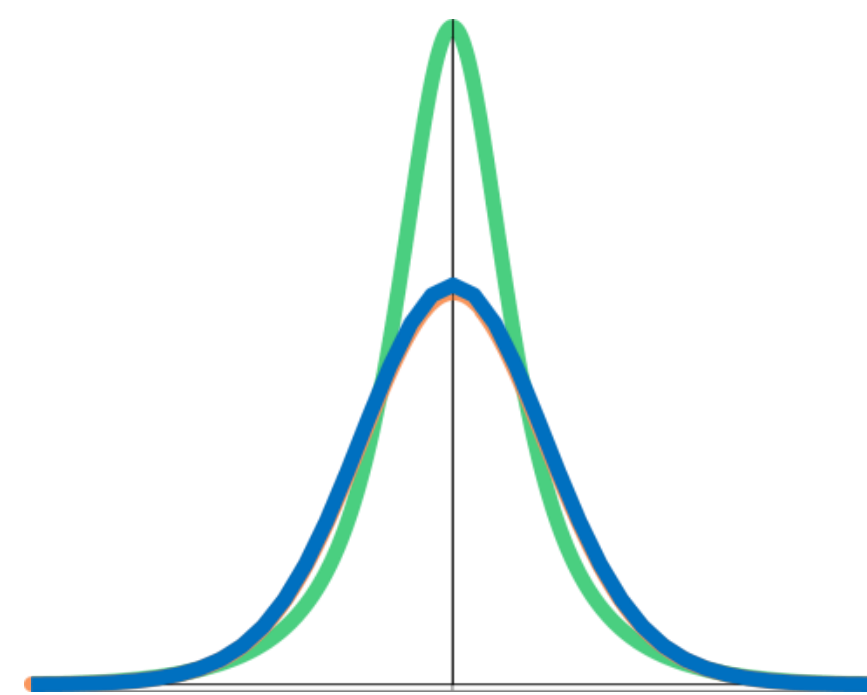


polystyrene

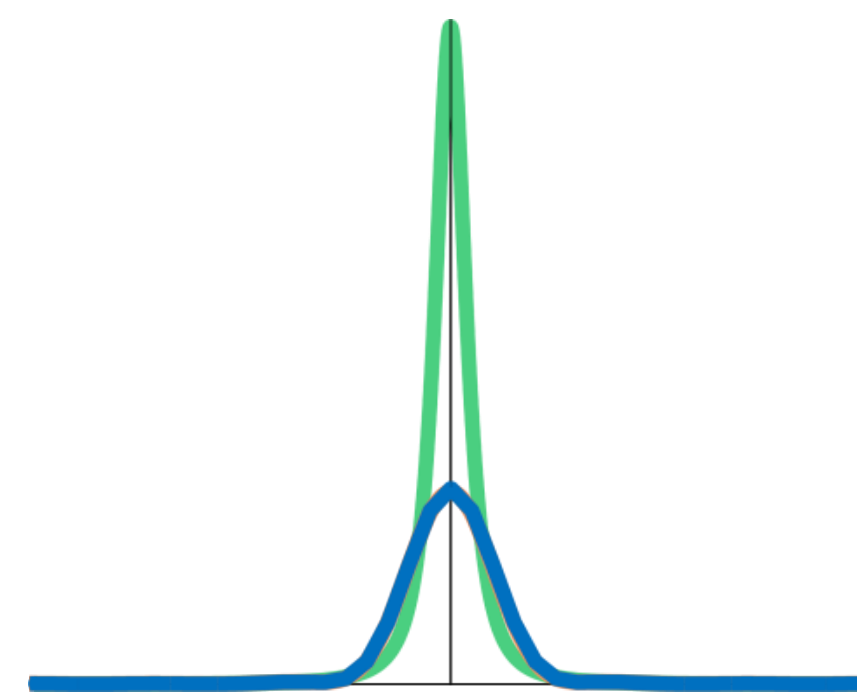


aluminum oxide

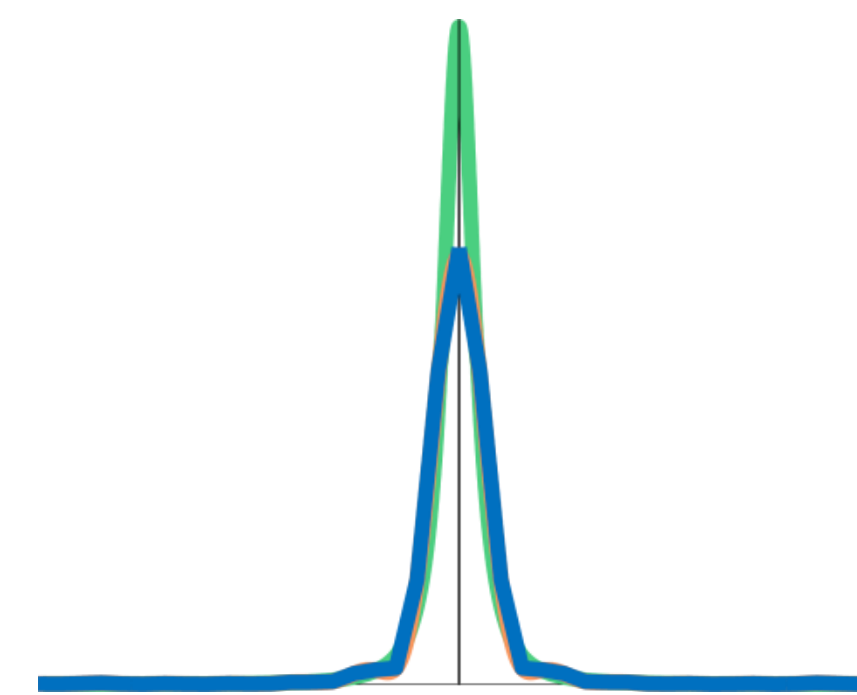
very precise dispersions (NIST Traceable Standards)



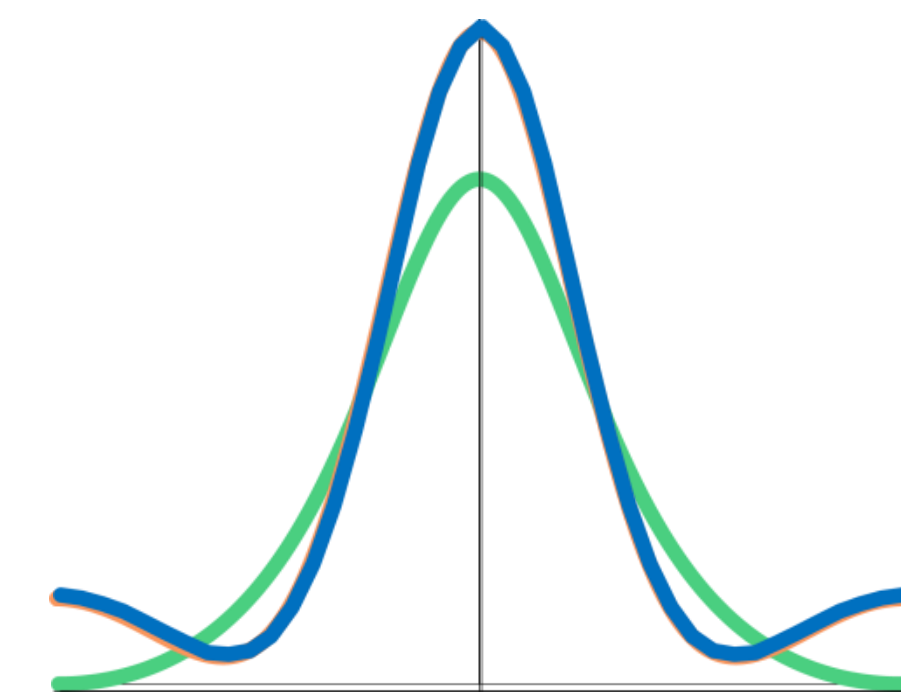
polystyrene 1



polystyrene 2



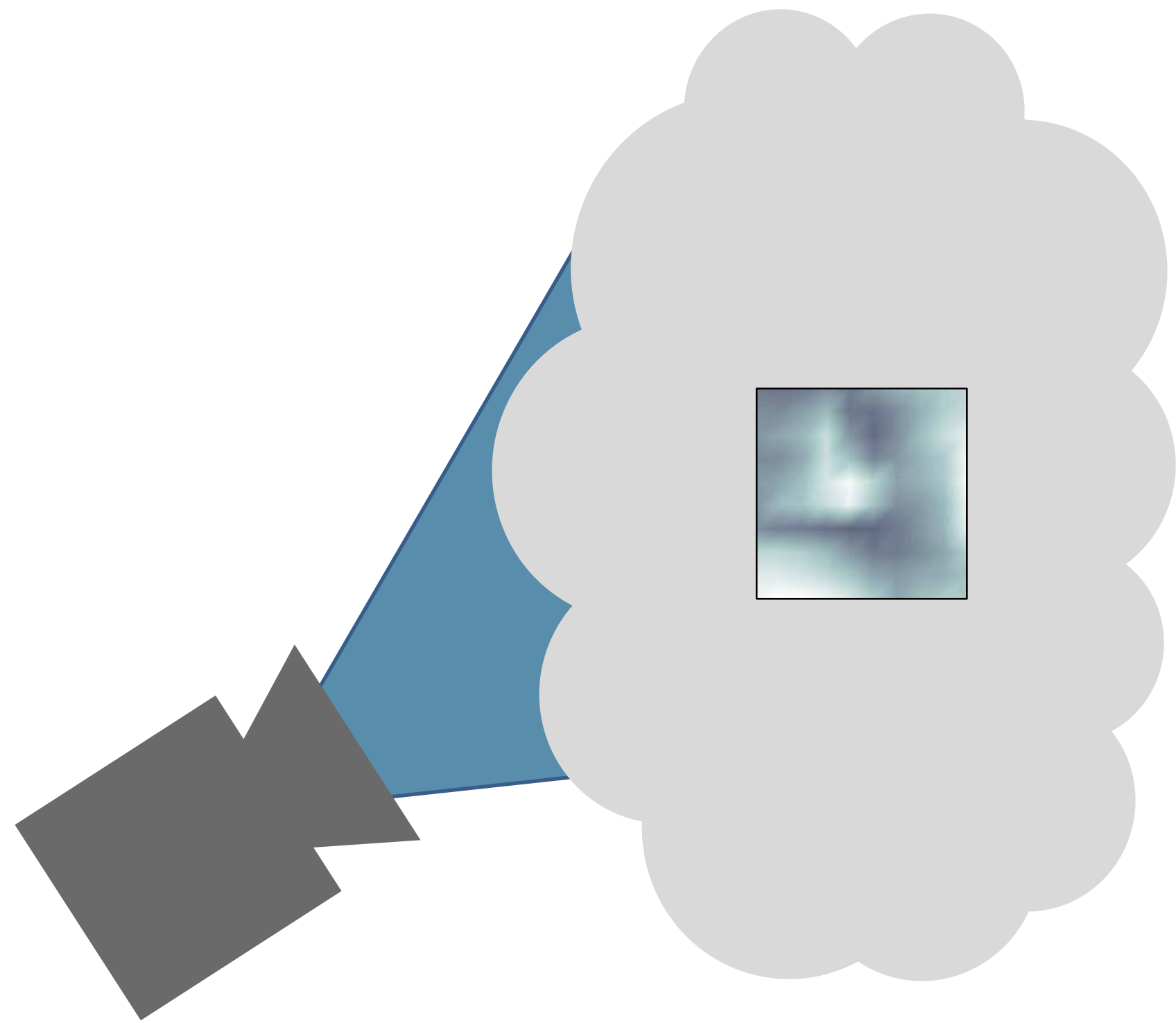
polystyrene 3



aluminum oxide



Optical tomography



camera

thick smoke cloud



simulated camera
measurements



reconstructed
cloud volume



slice through
the cloud

Computational Fabrication

Determining the material configuration for individual voxels in full-color inkjet 3D printing

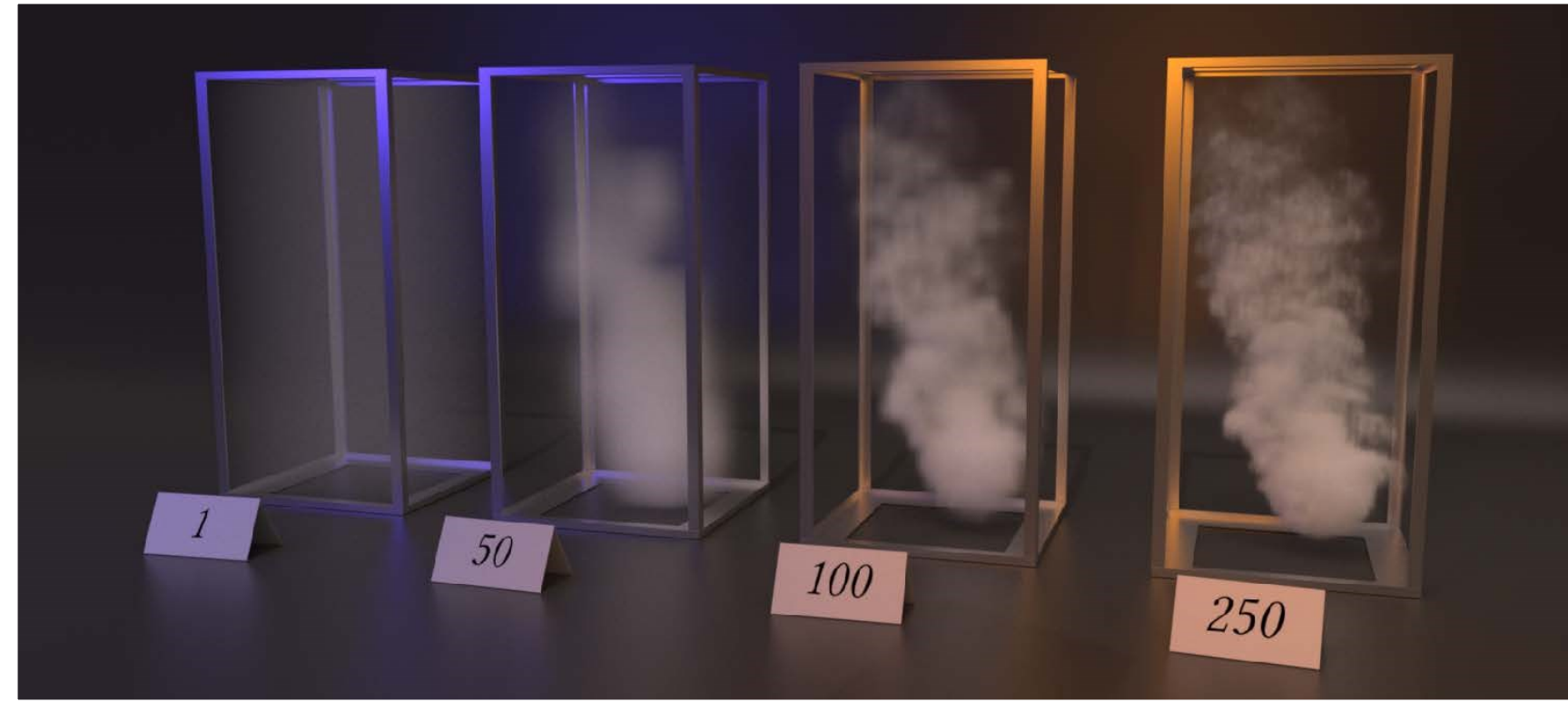


[Nindel et al. 2021]

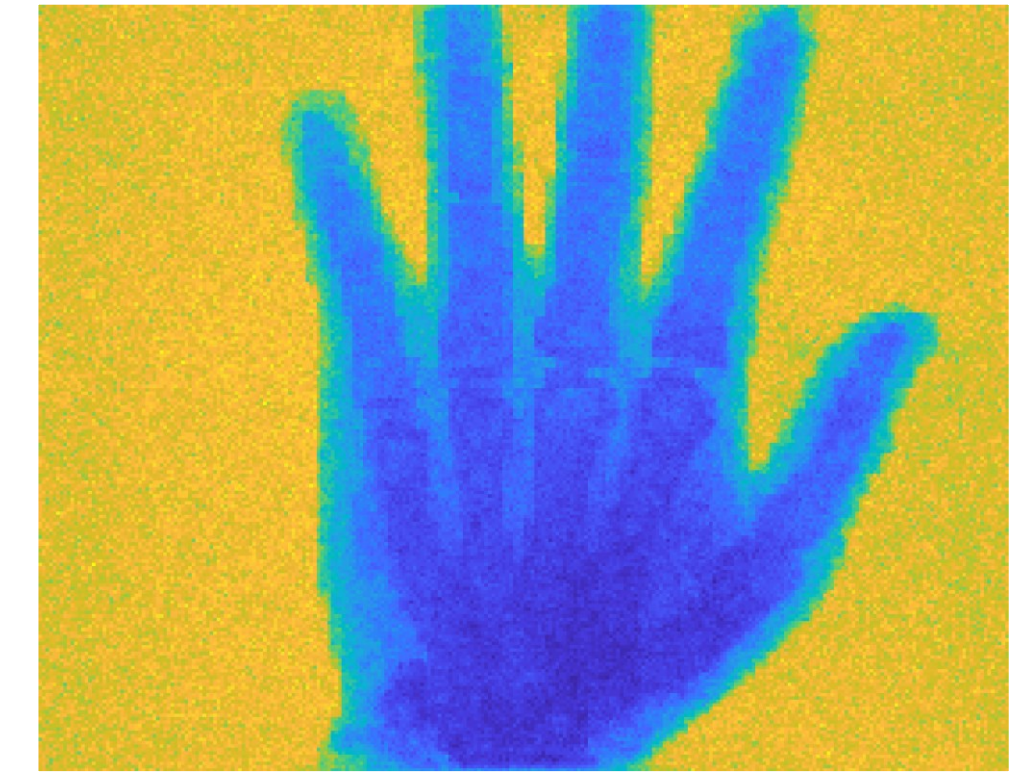
Active area of research



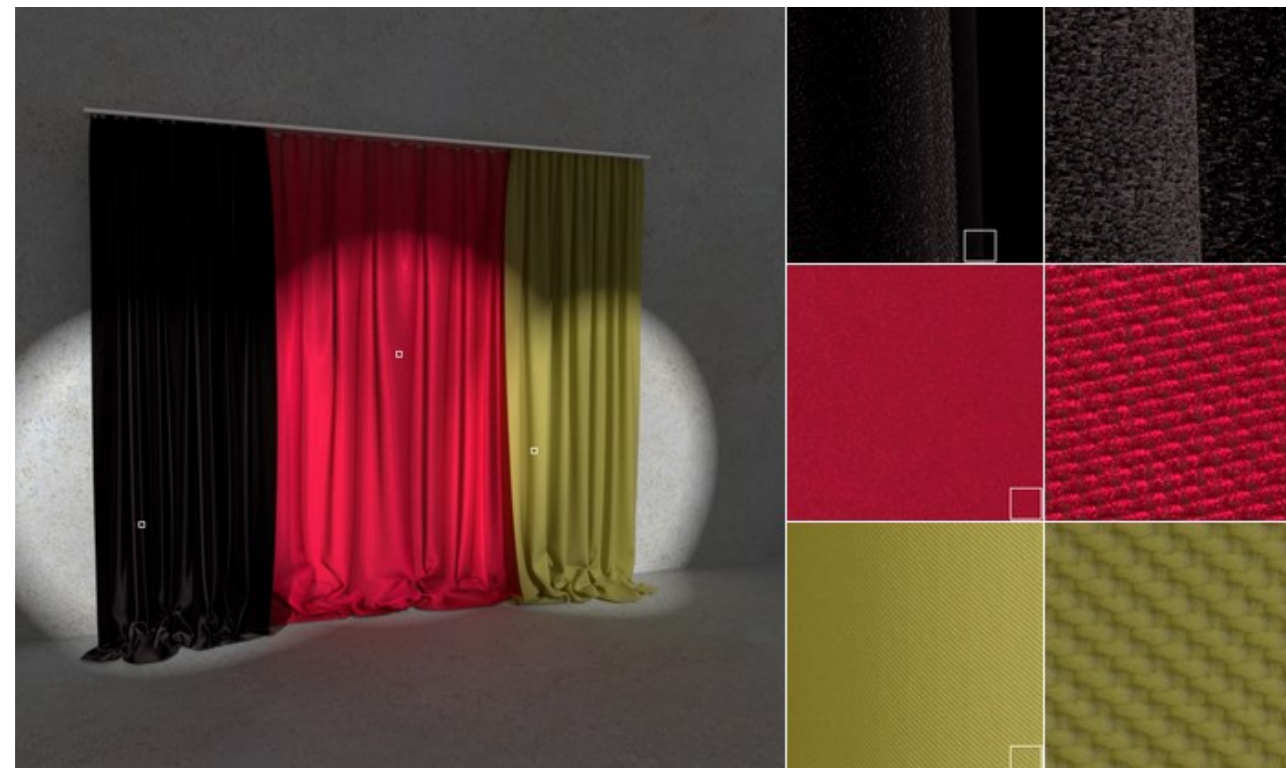
industrial dispersions
[Gkioulekas et al. 2013]



efficient algorithms
[Nimier-David et al. 2019, 2020]



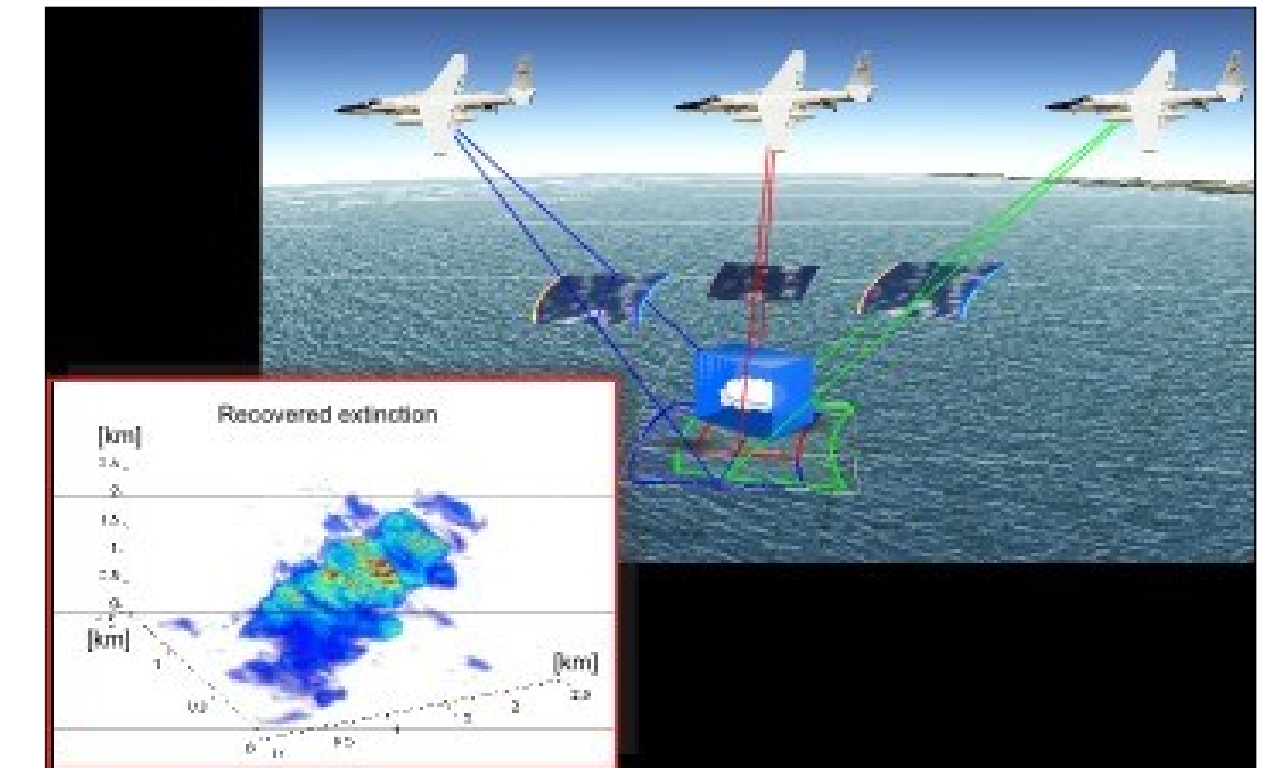
computed tomography
[Geva et al. 2018]



woven fabrics
[Khungurn et al. 2015,
Zhao et al. 2016]

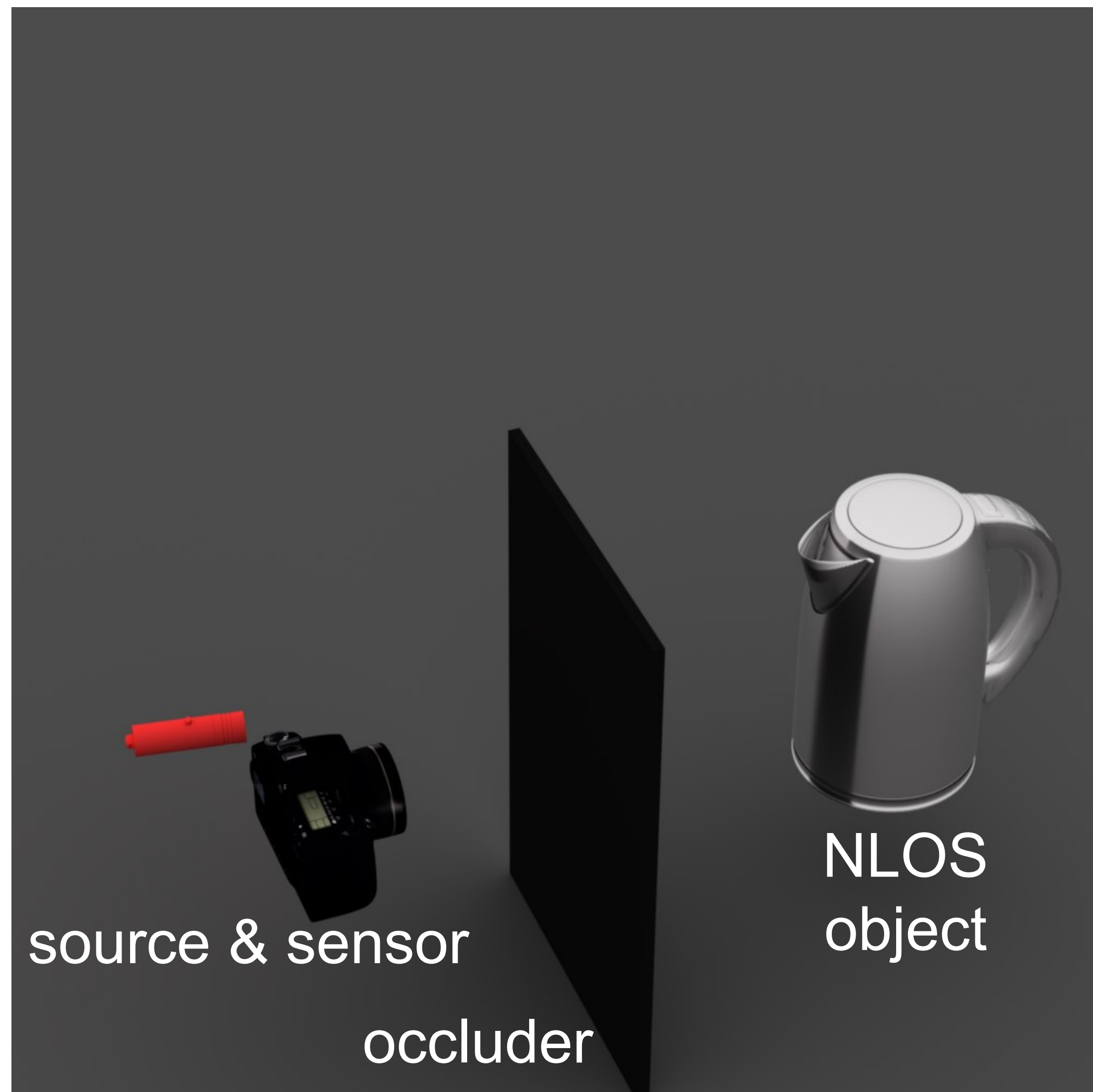


3D printing
[Elek et al. 2019,
Nindel et al. 2021]

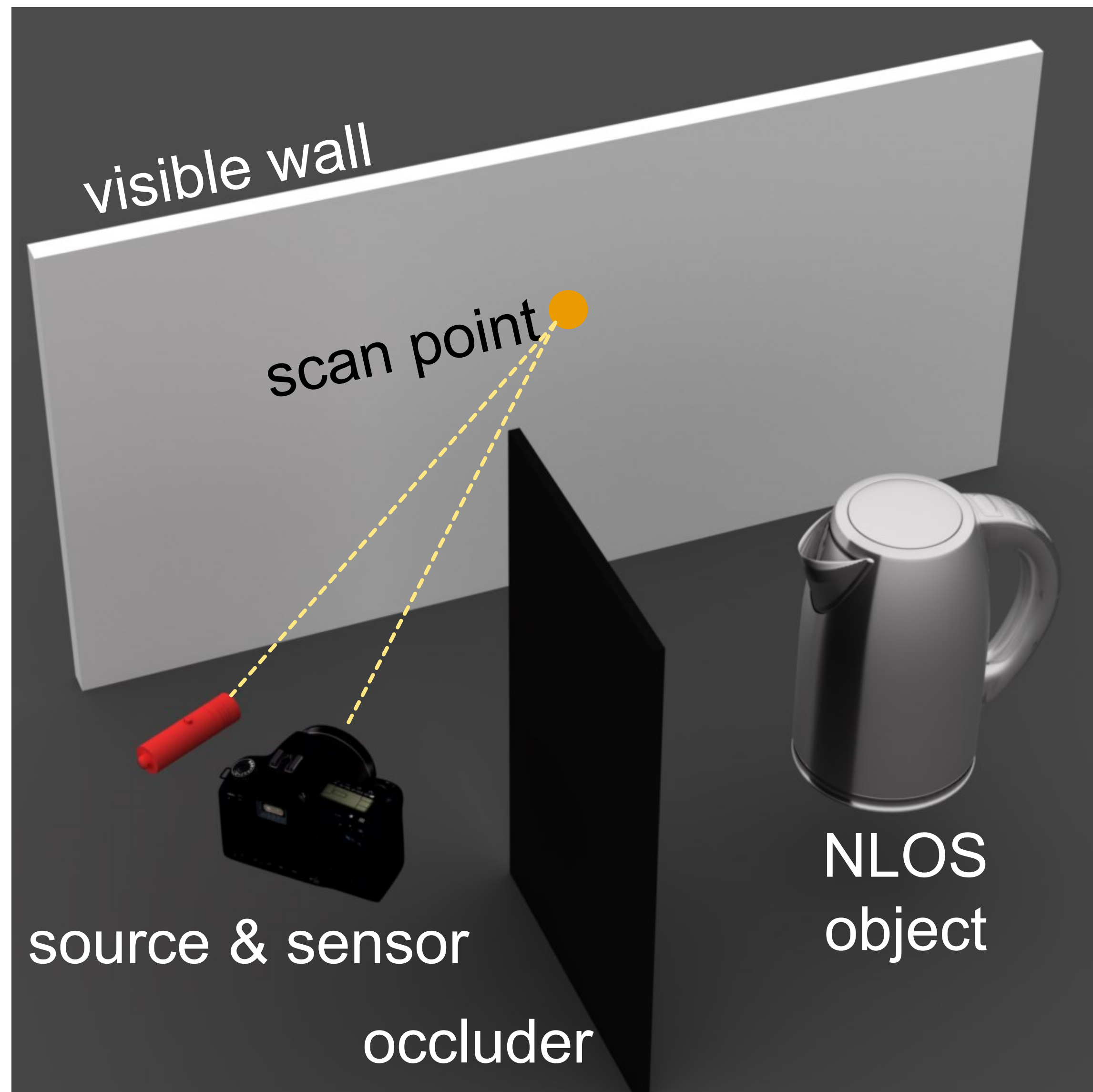


cloud tomography
[Levis et al. 2015,
2017, 2020]

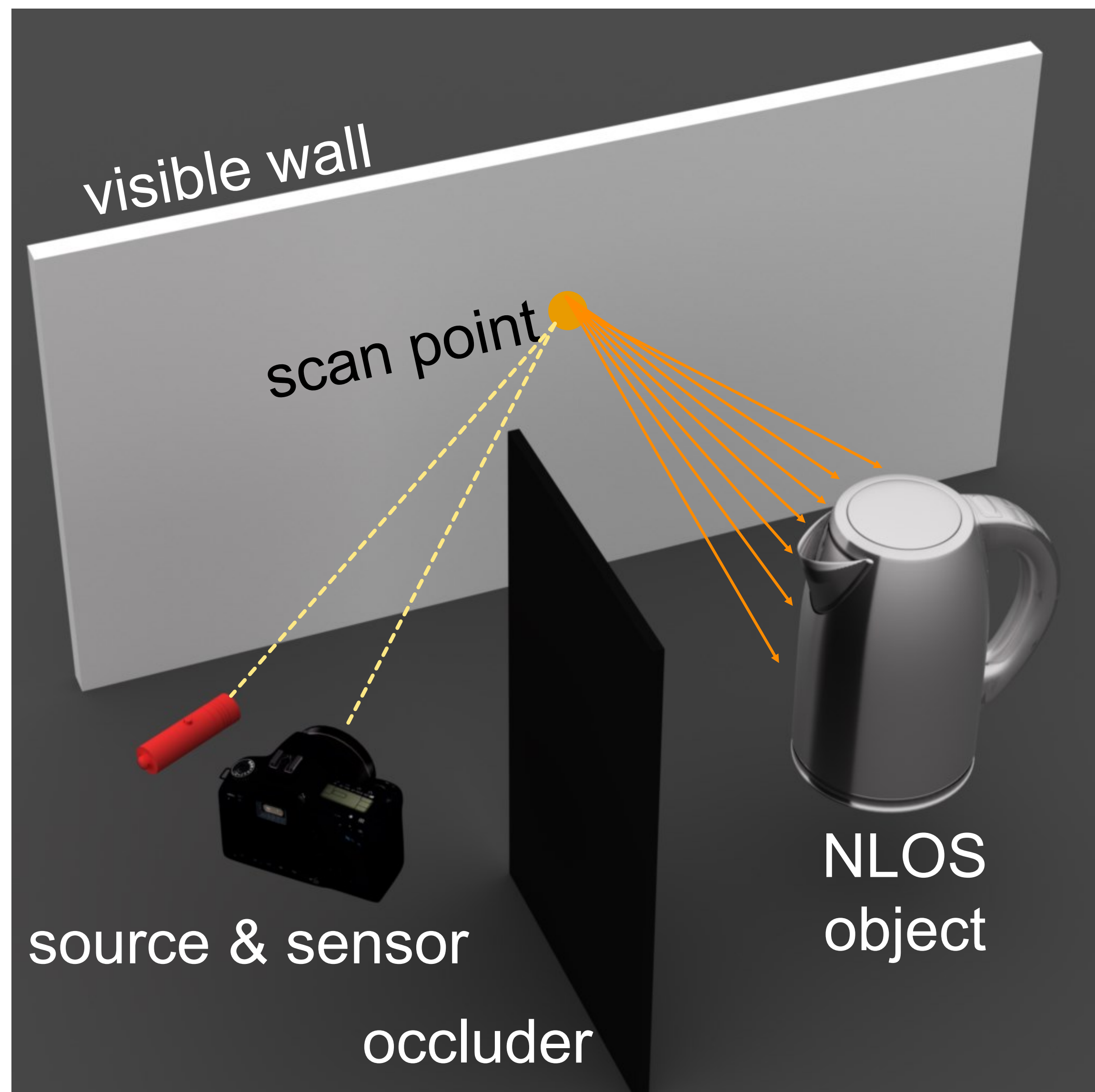
Non-line-of-sight (NLOS) imaging



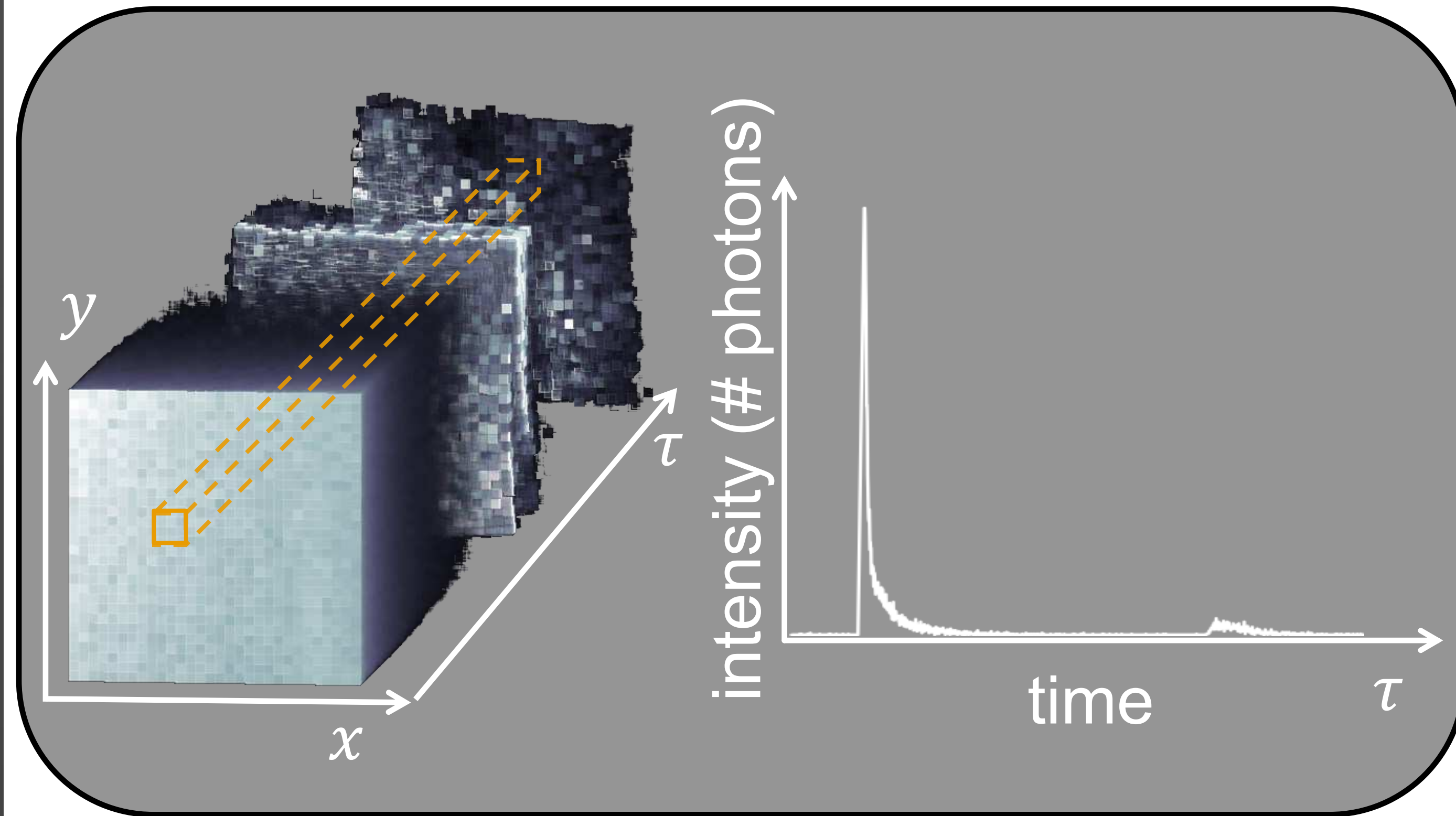
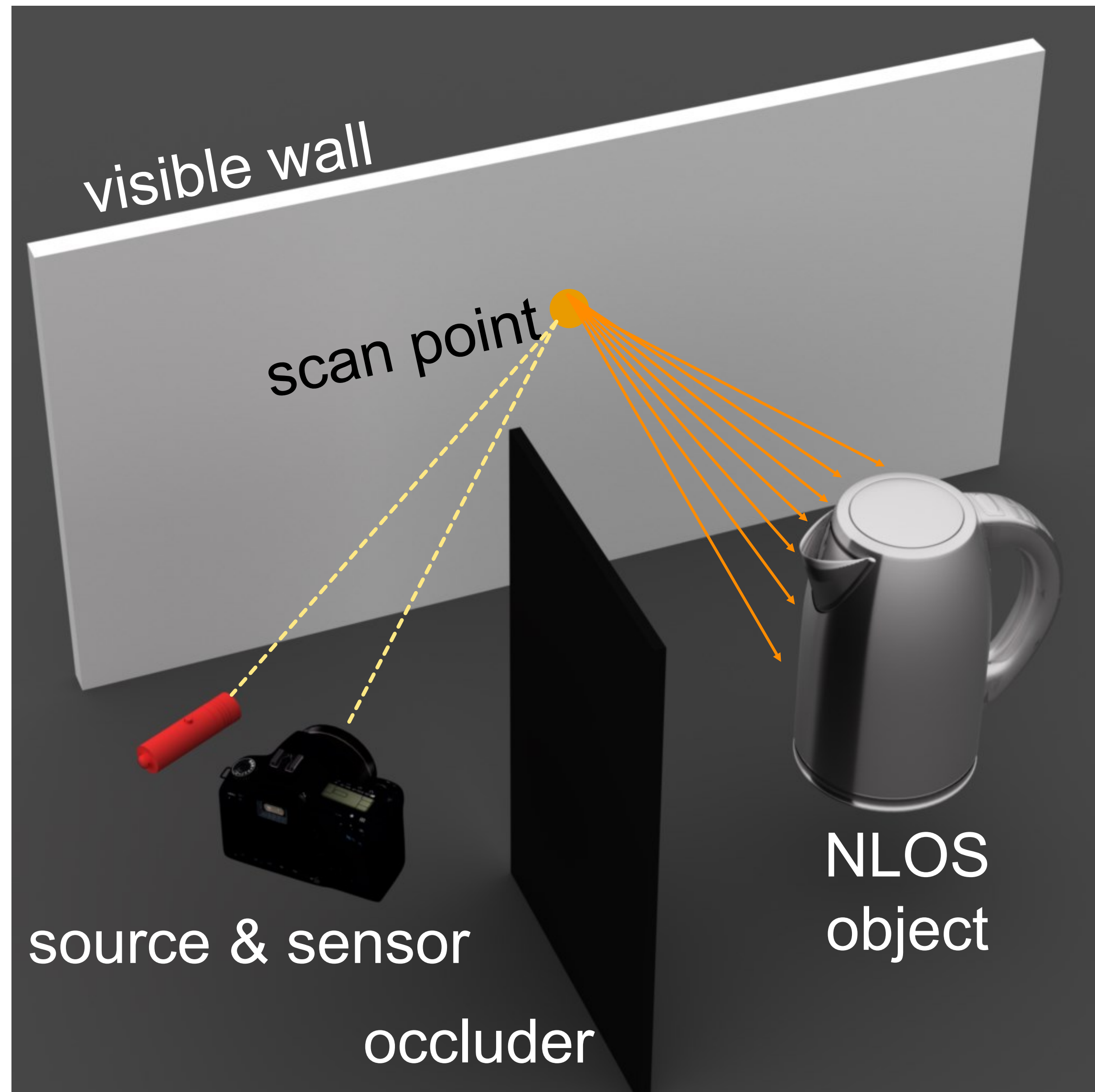
Non-line-of-sight (NLOS) imaging



Non-line-of-sight (NLOS) imaging

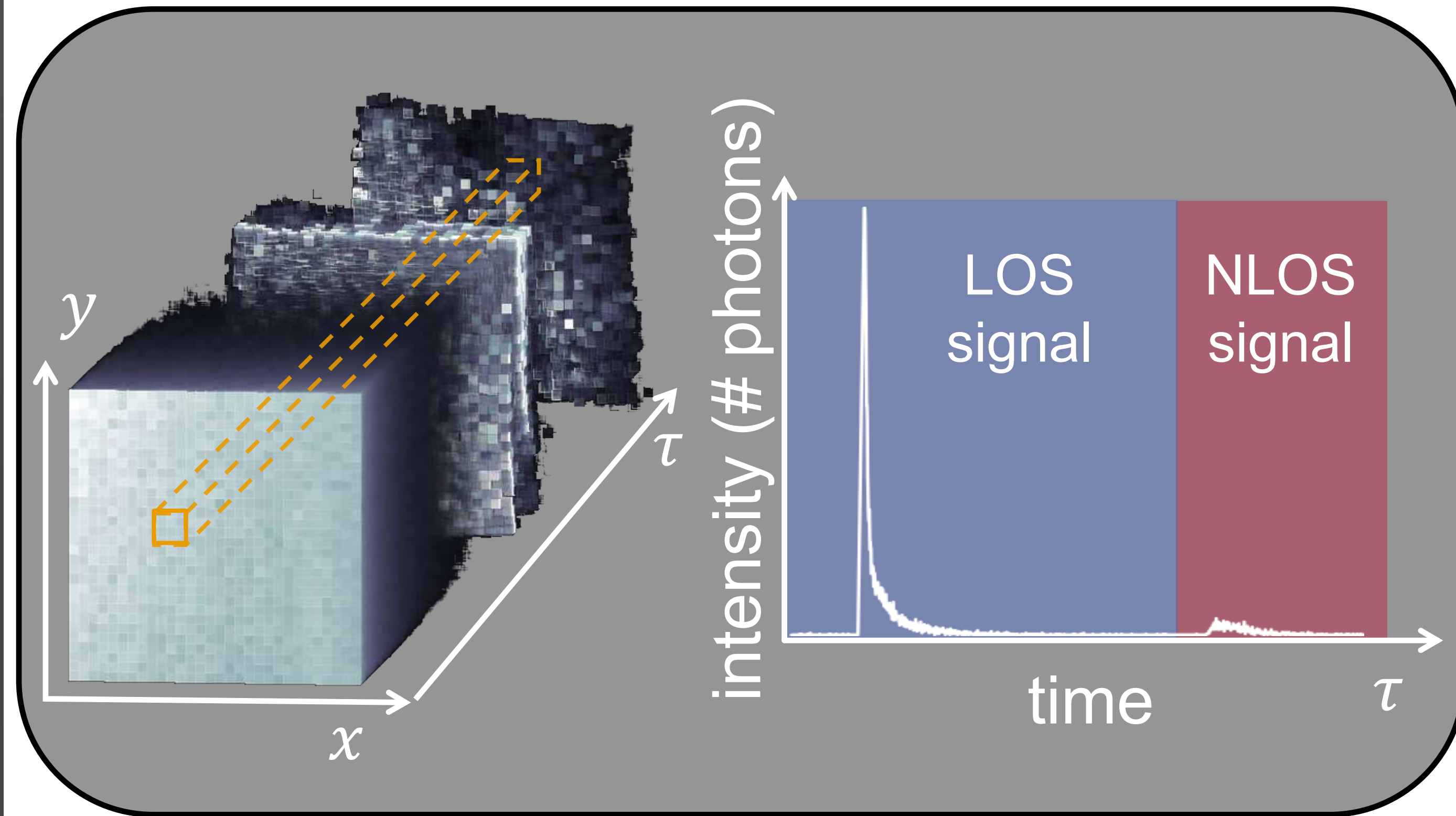
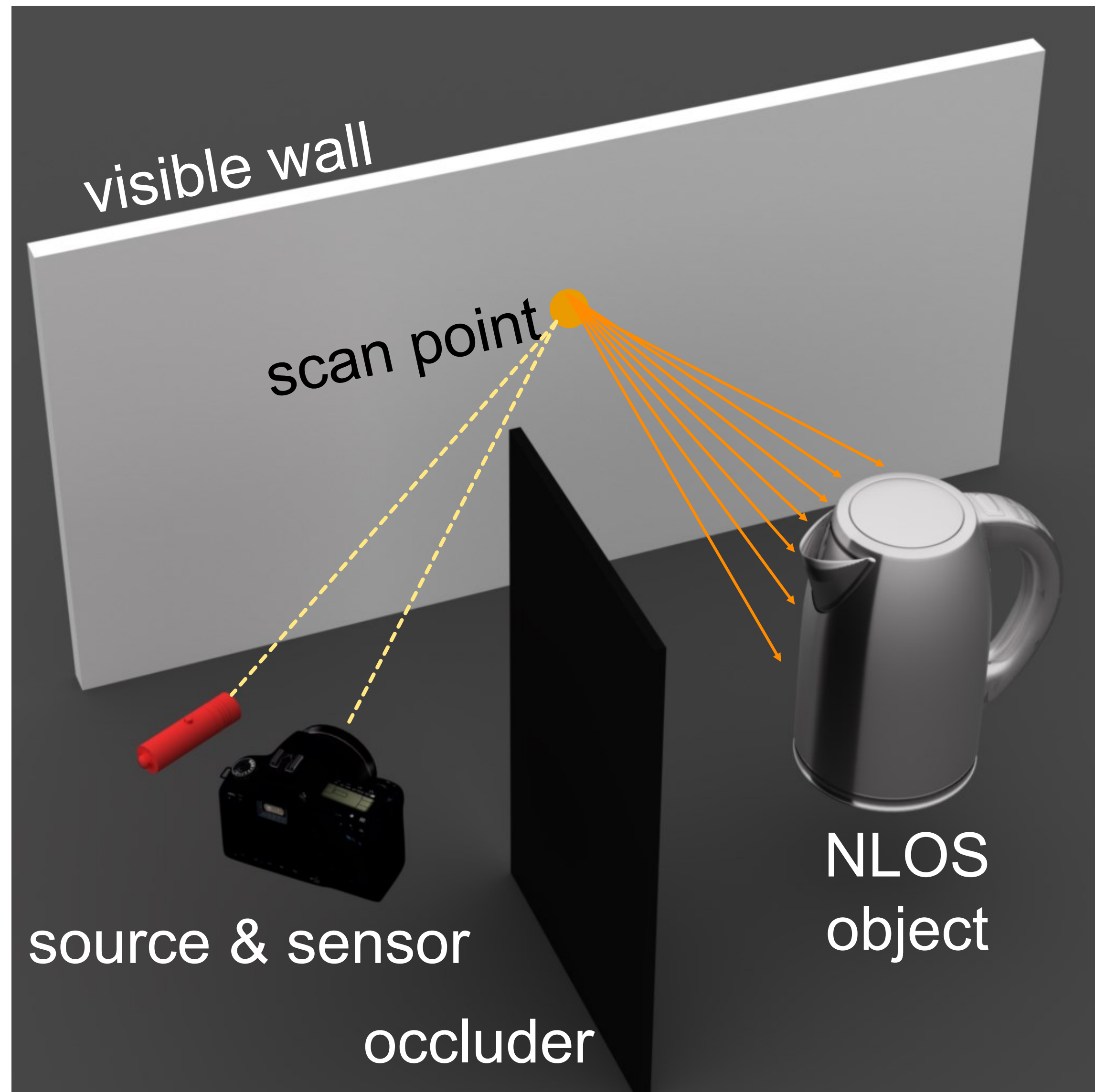


Non-line-of-sight (NLOS) imaging



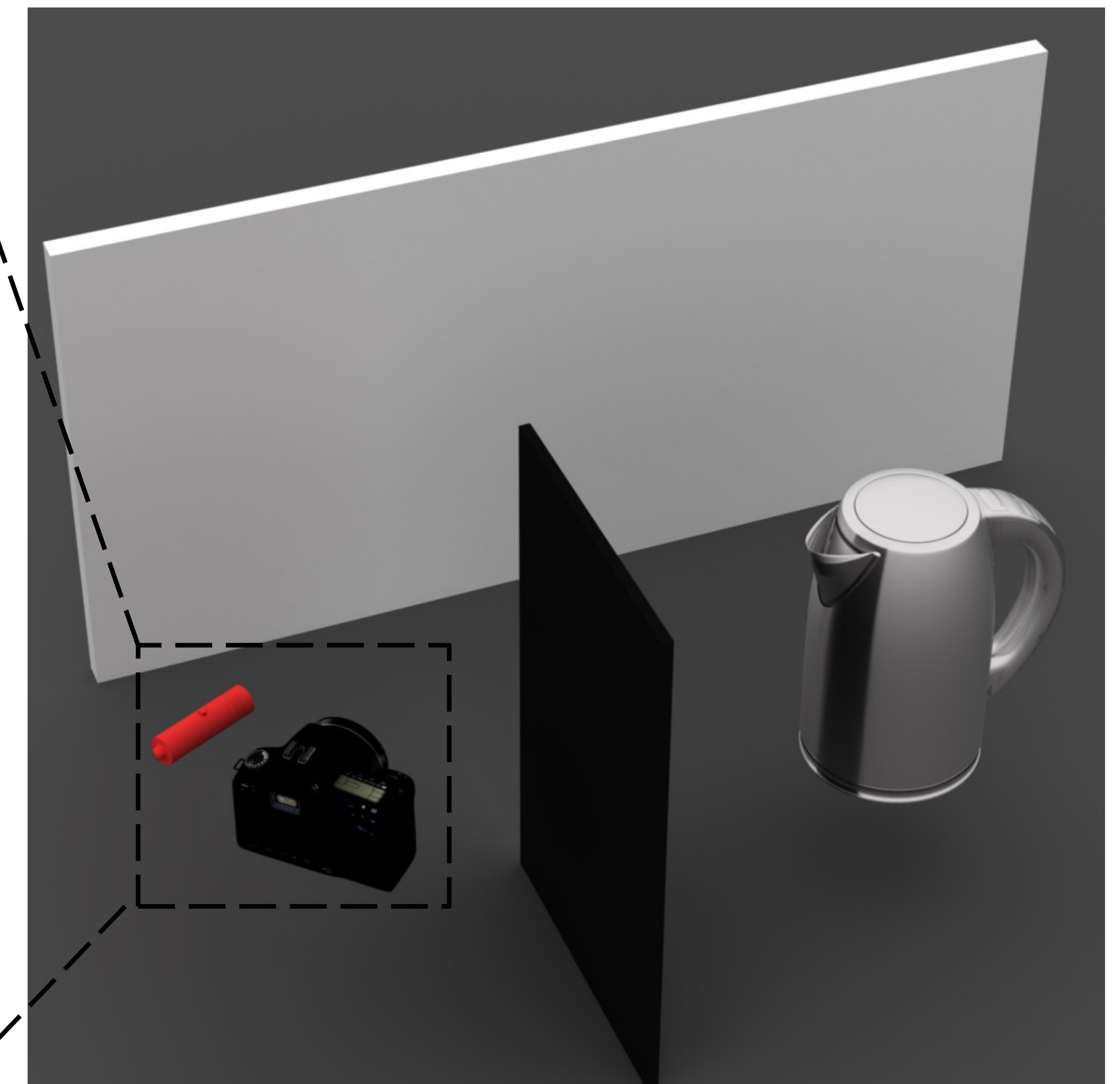
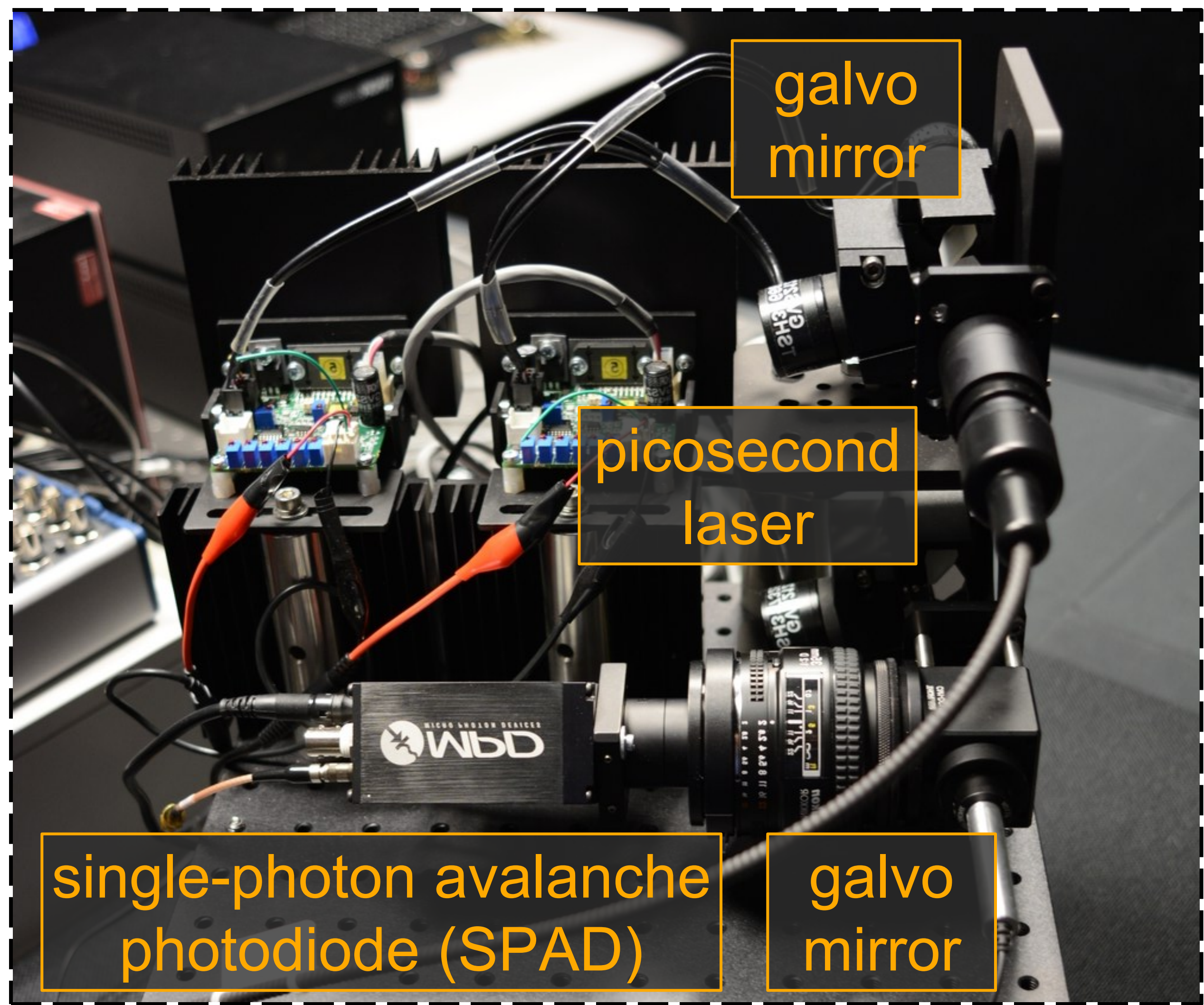
Time-of-flight measurements

Non-line-of-sight (NLOS) imaging

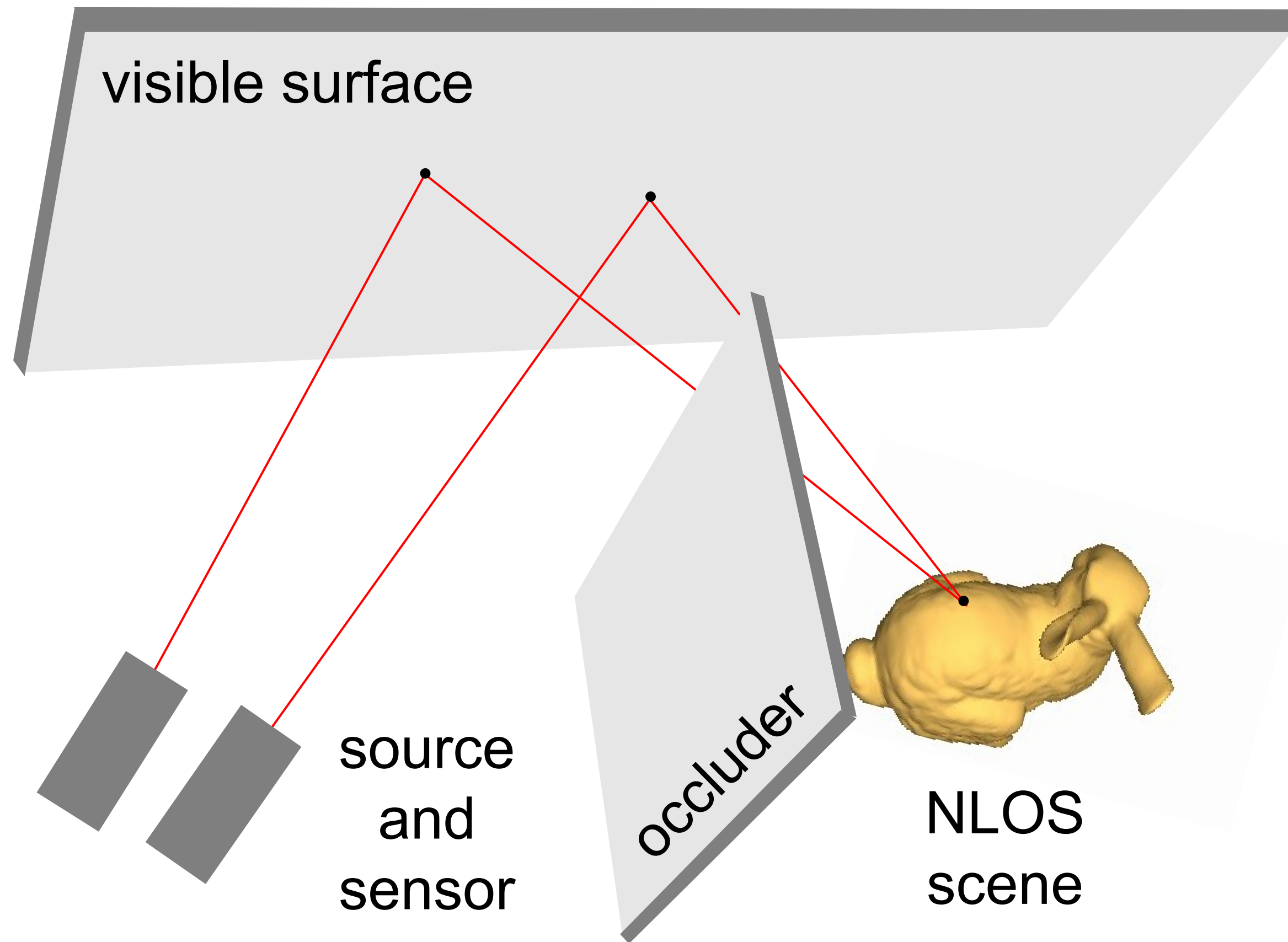


Time-of-flight measurements

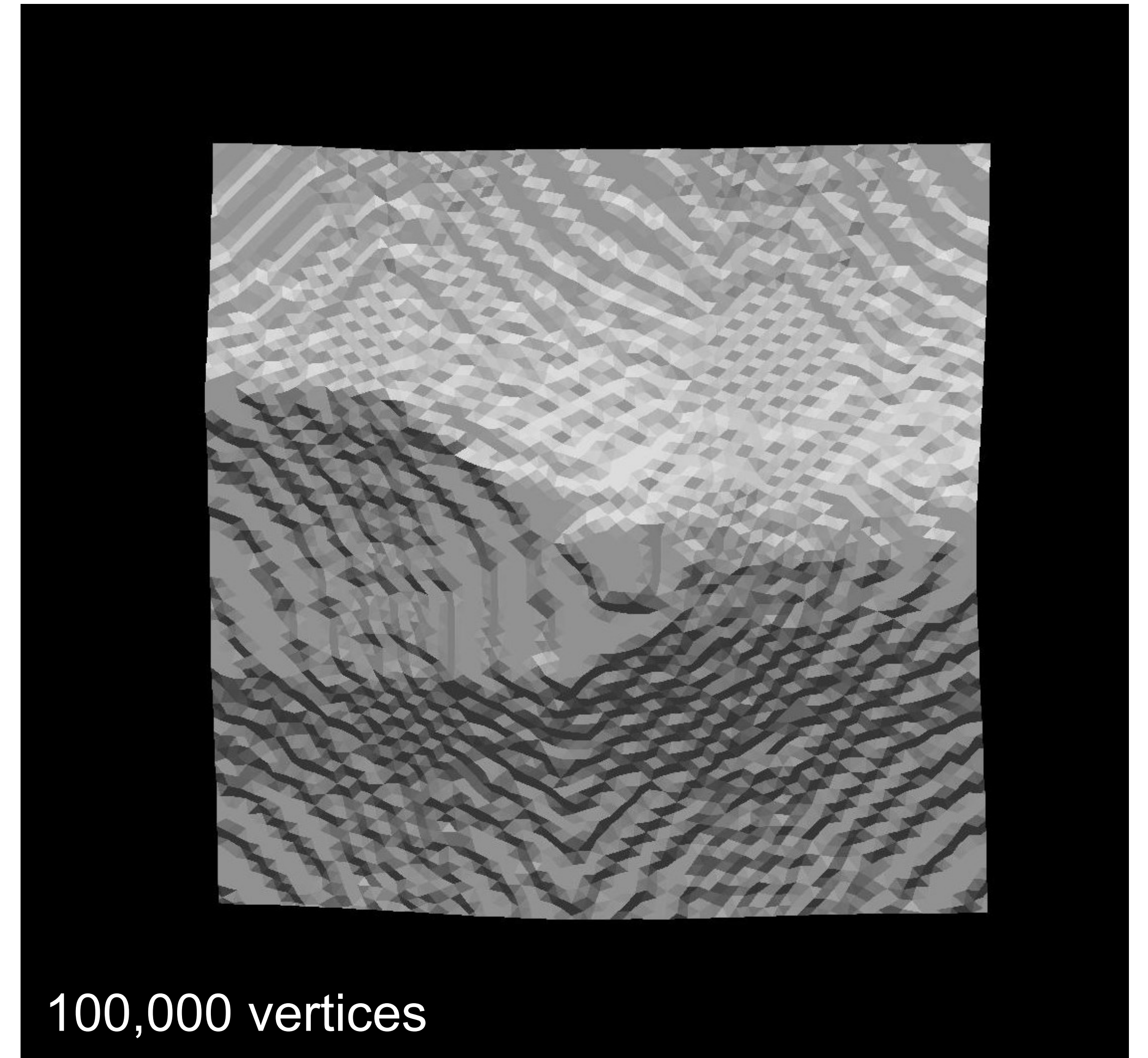
SPAD-based lidar



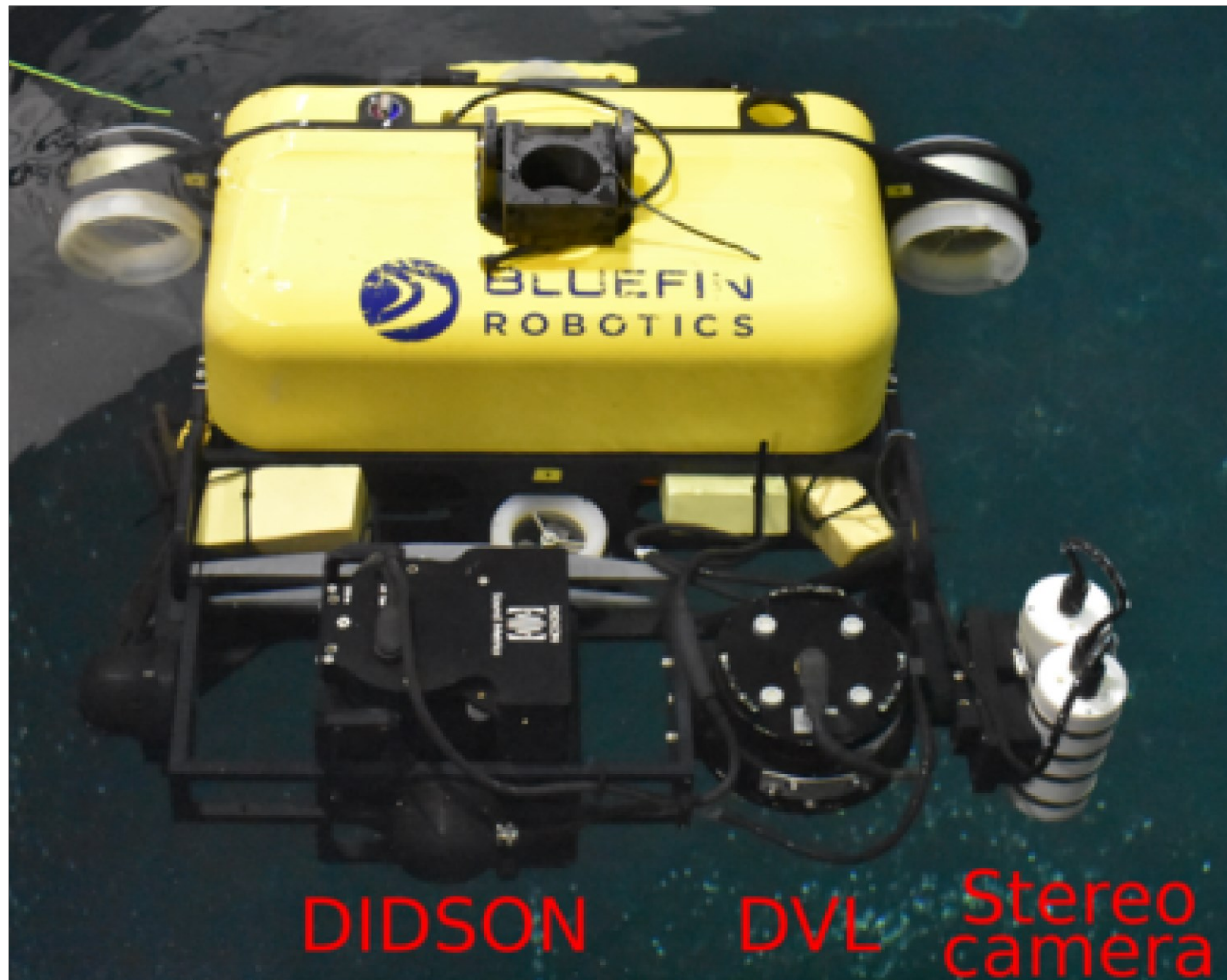
NLOS shape optimization



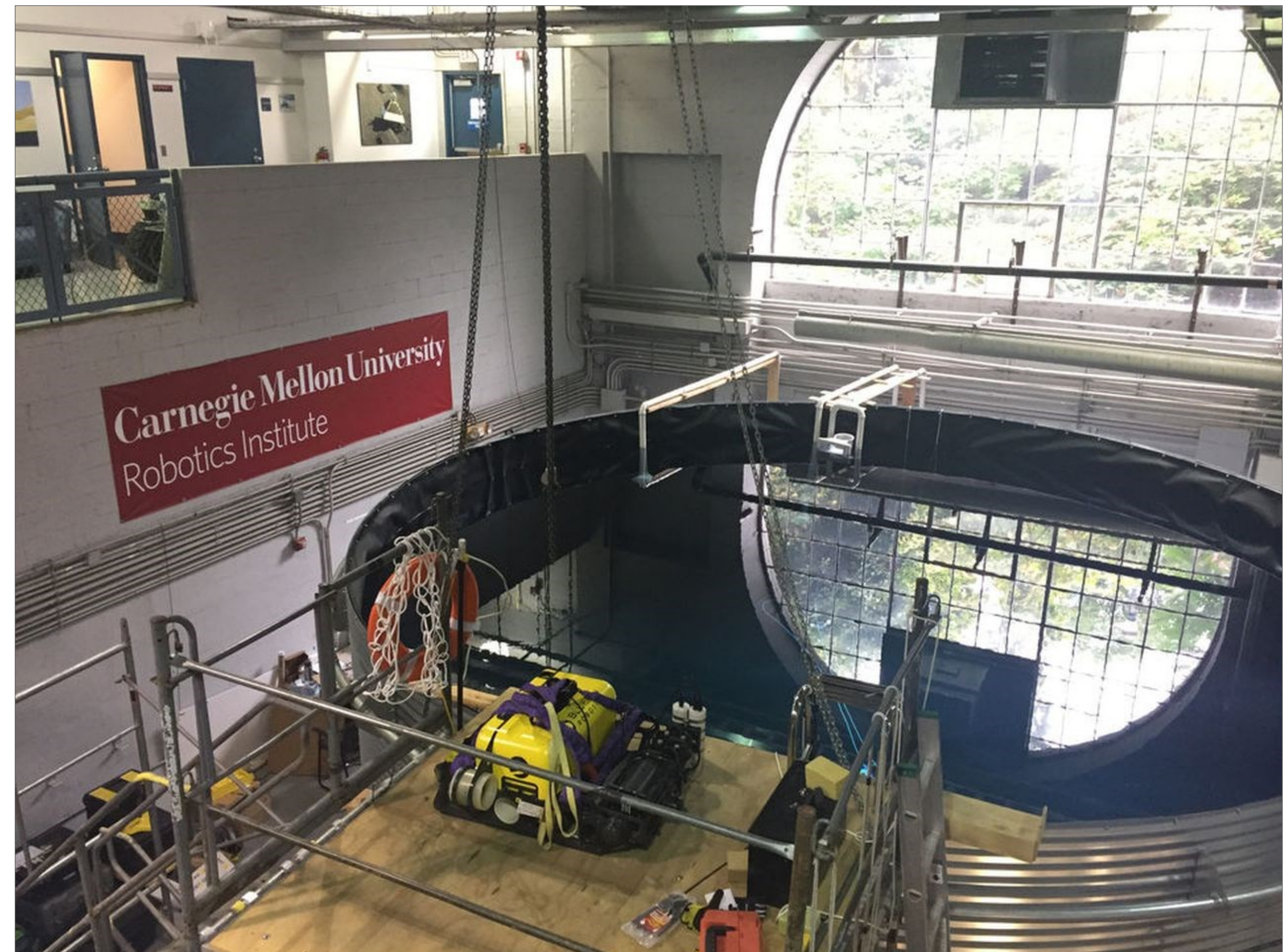
Simulated time-of-flight data



Underwater 3D using imaging SONAR

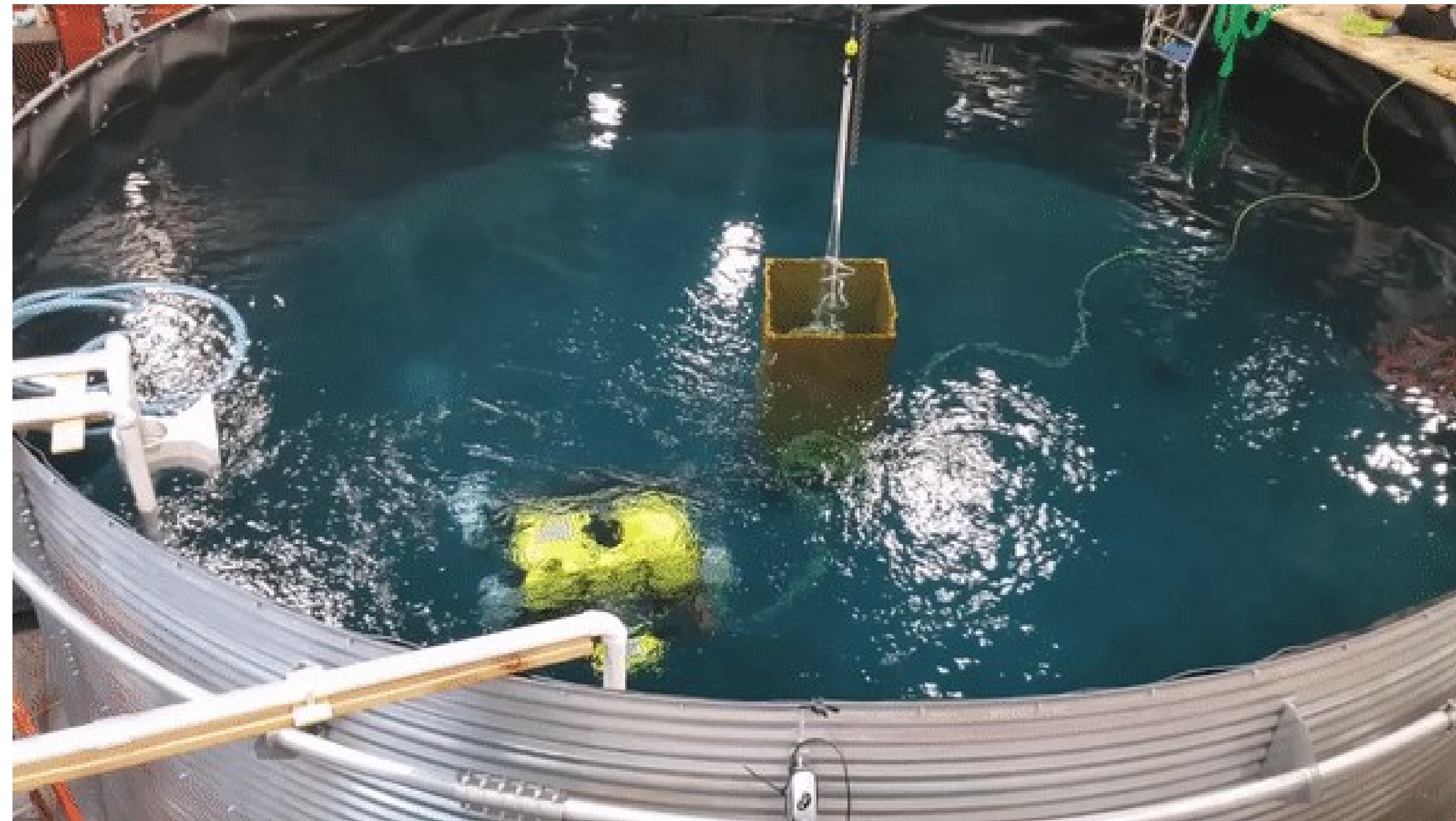


Underwater robot with sonar



Robotics Institute High Bay

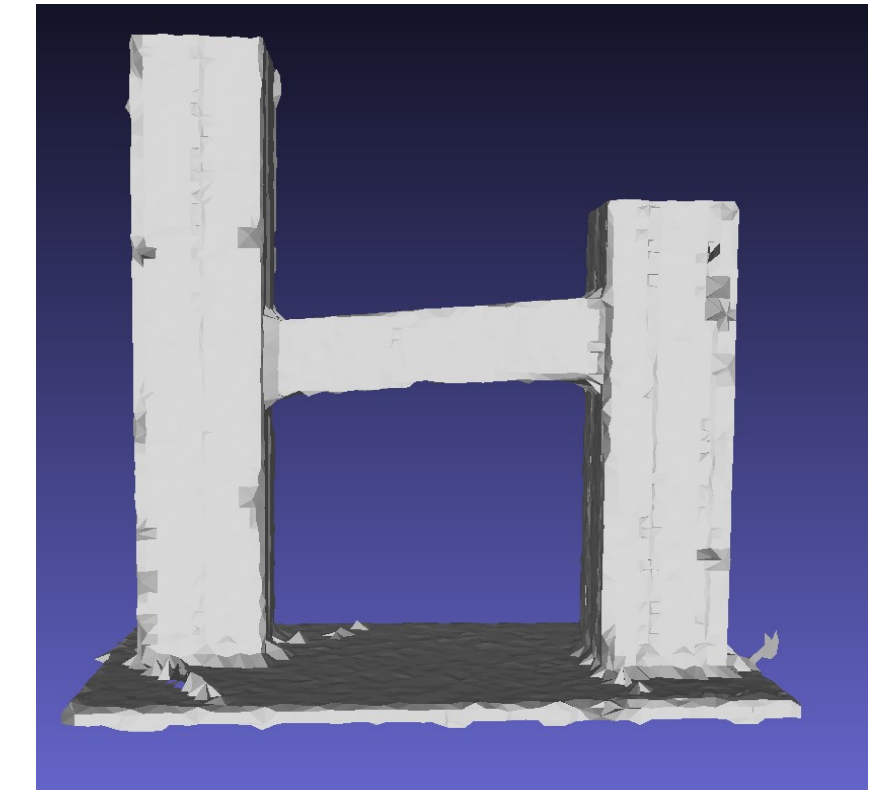
Underwater 3D using imaging SONAR



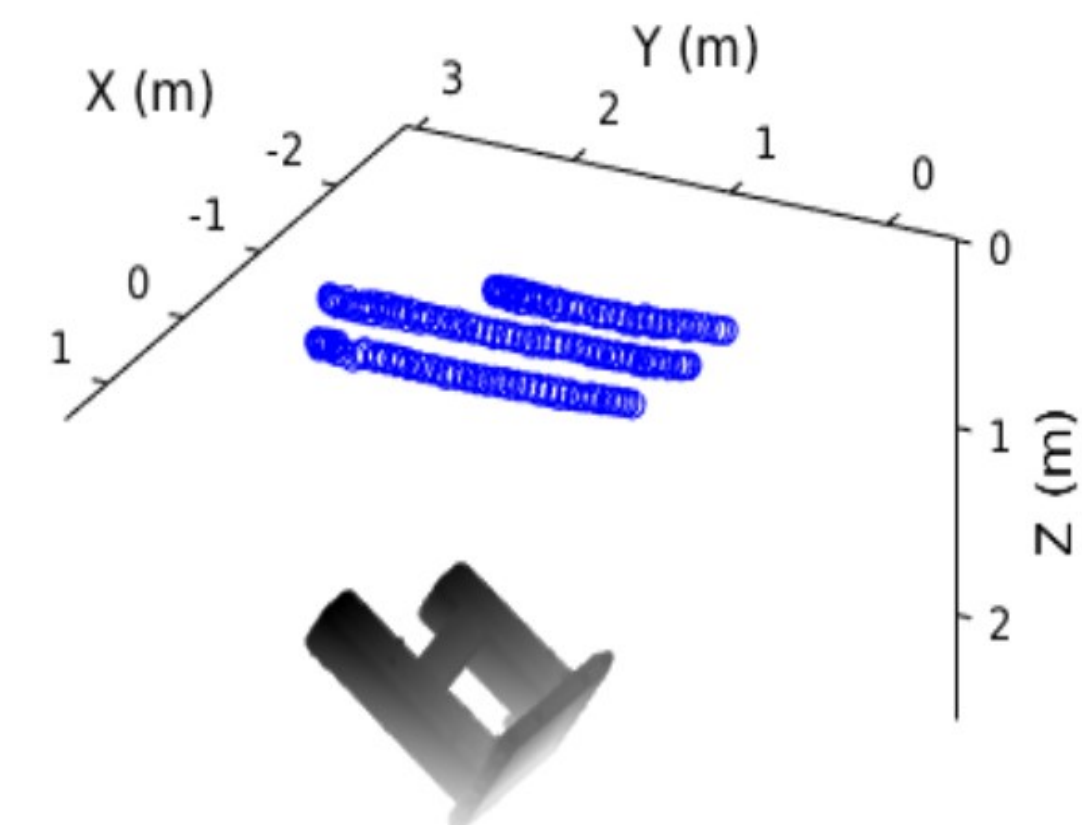
Test structure



Ground truth mesh
obtained using a laser scan



Sonar image collection points



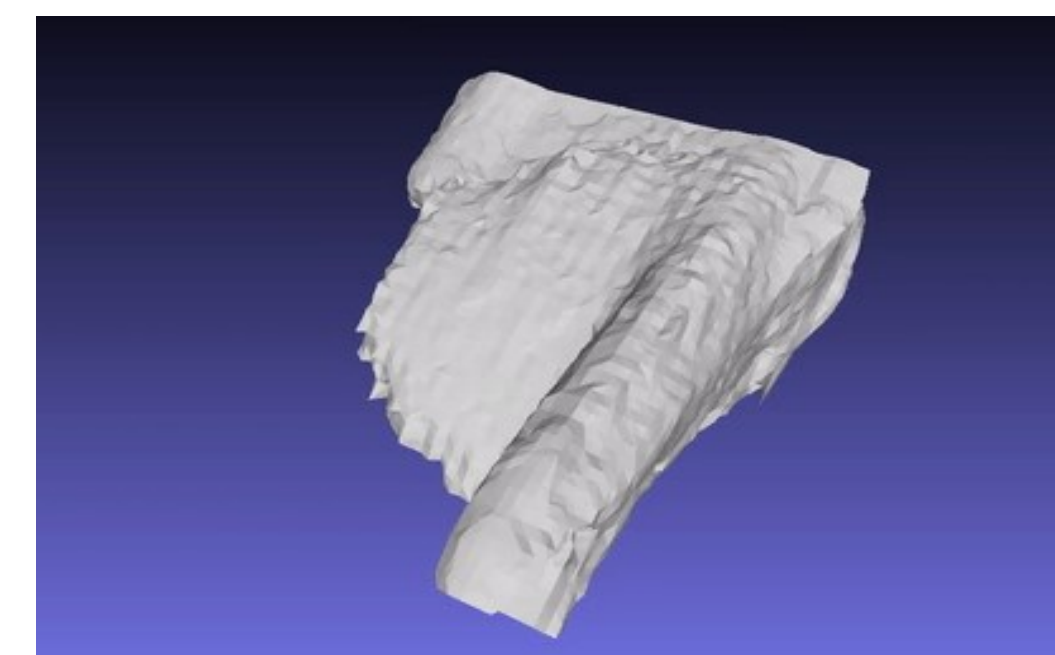
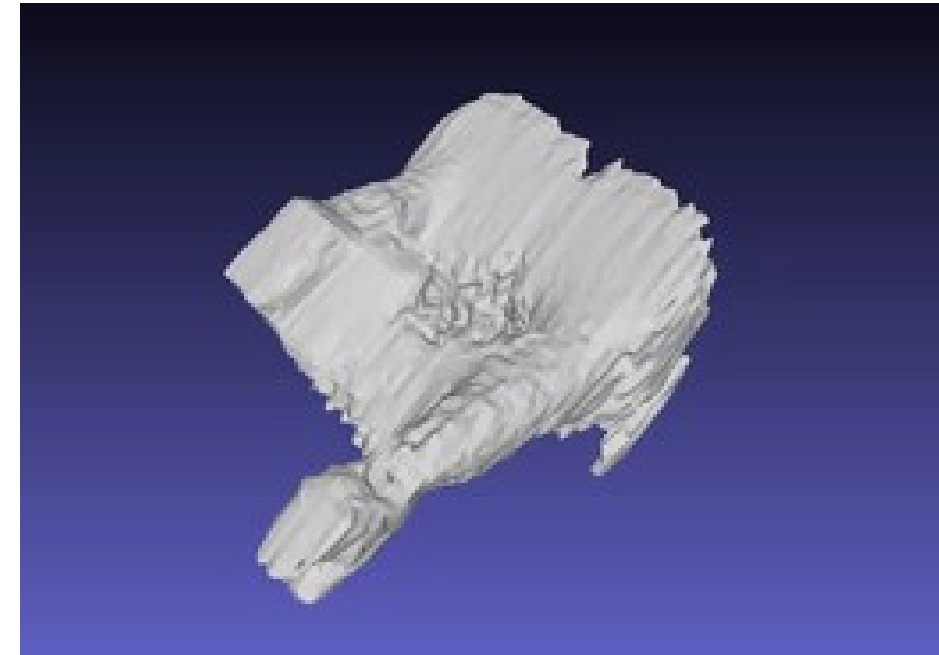
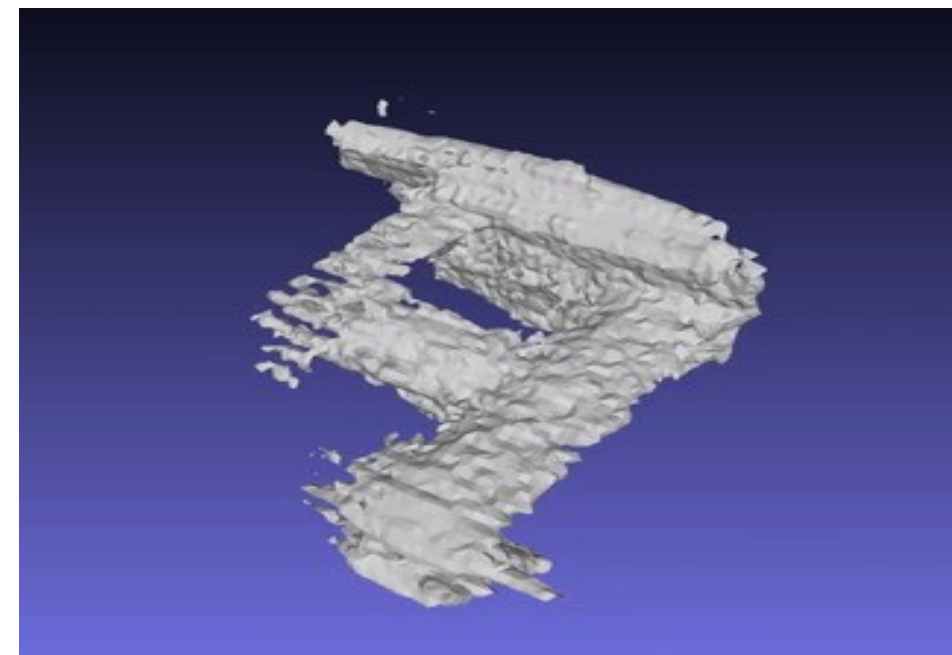
Underwater 3D using imaging SONAR

1 degree

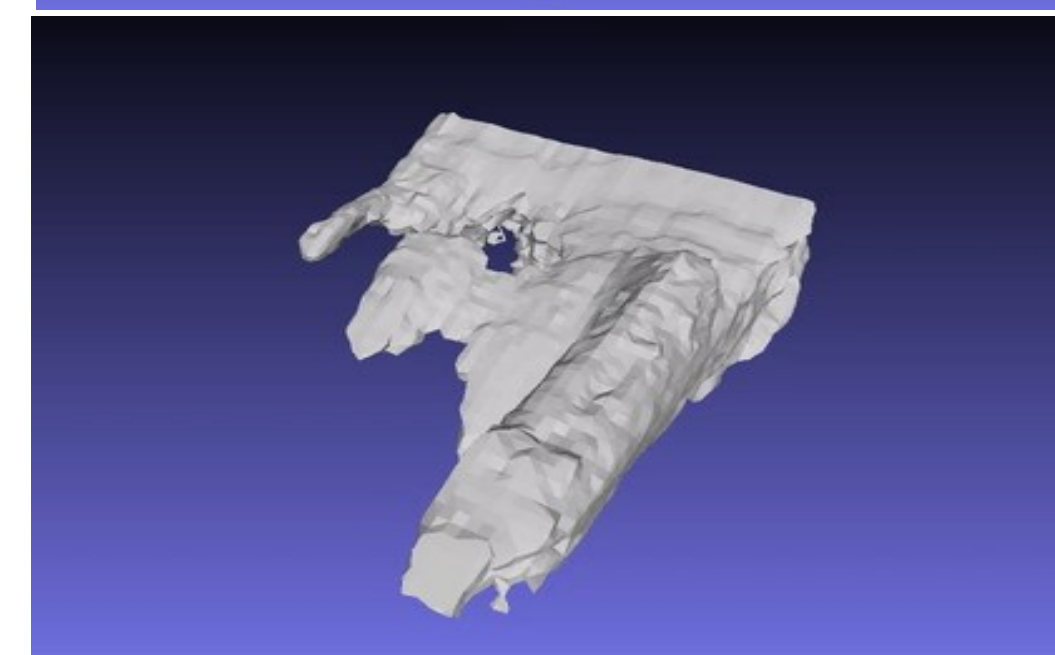
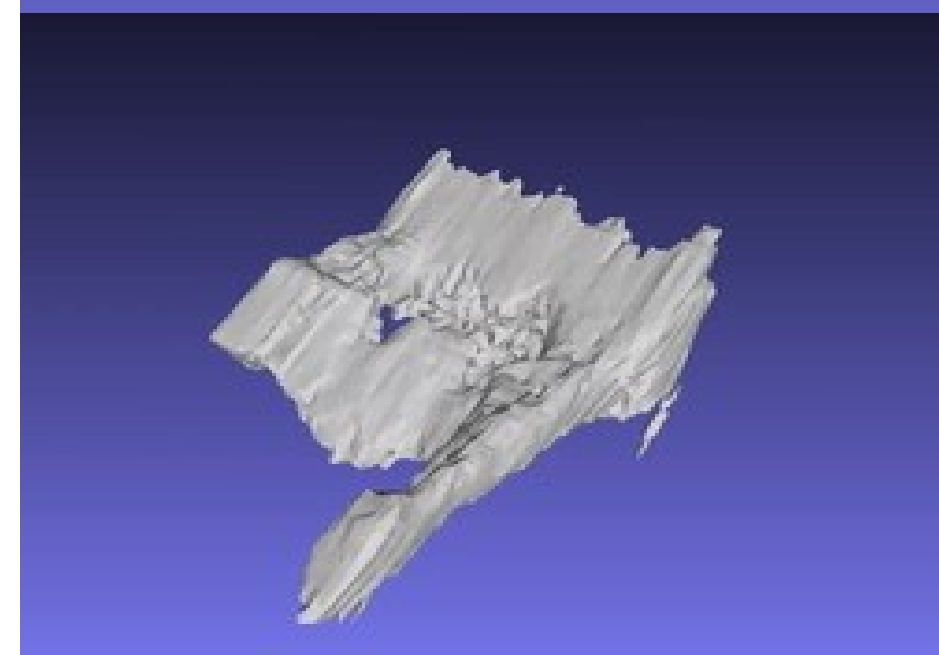
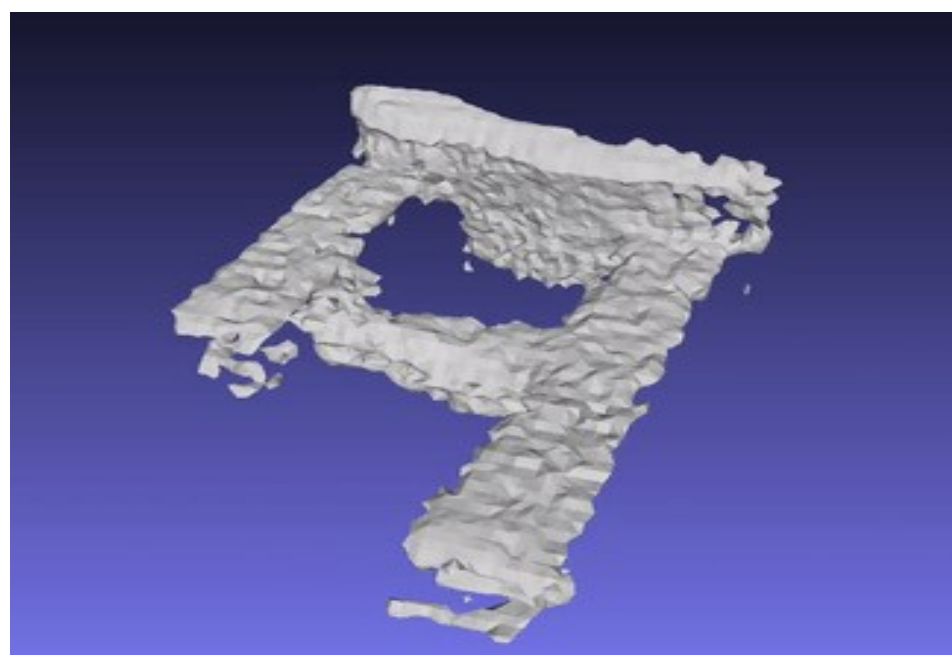
14 degrees

28 degrees

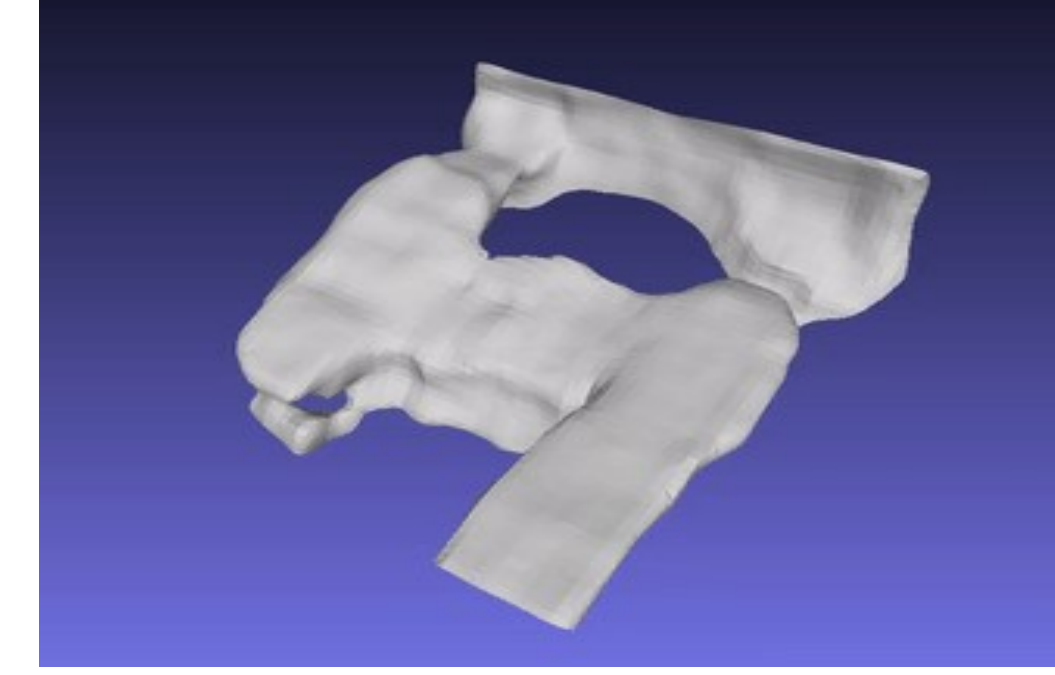
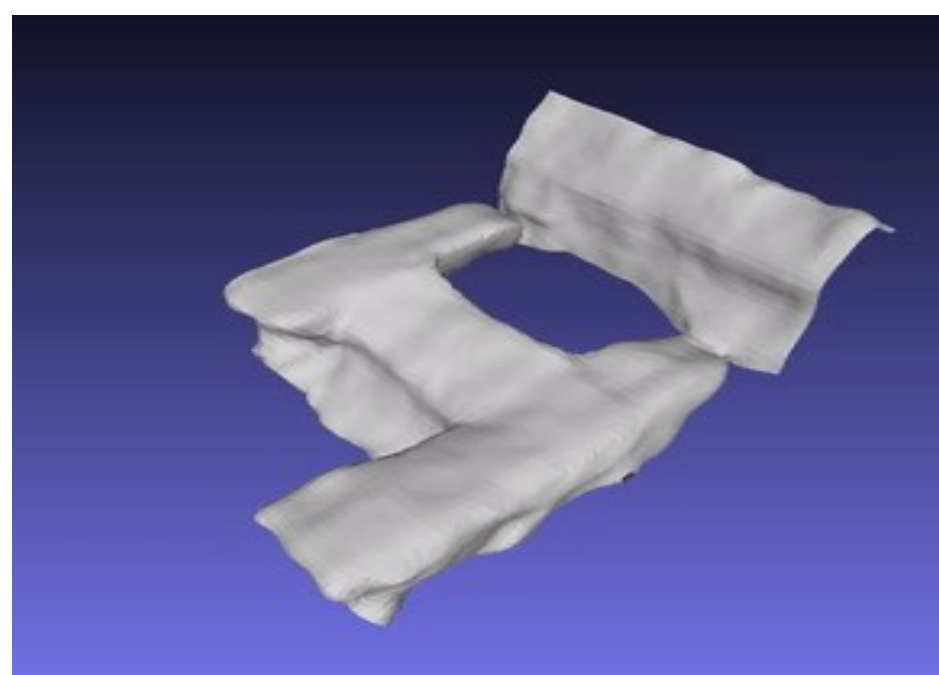
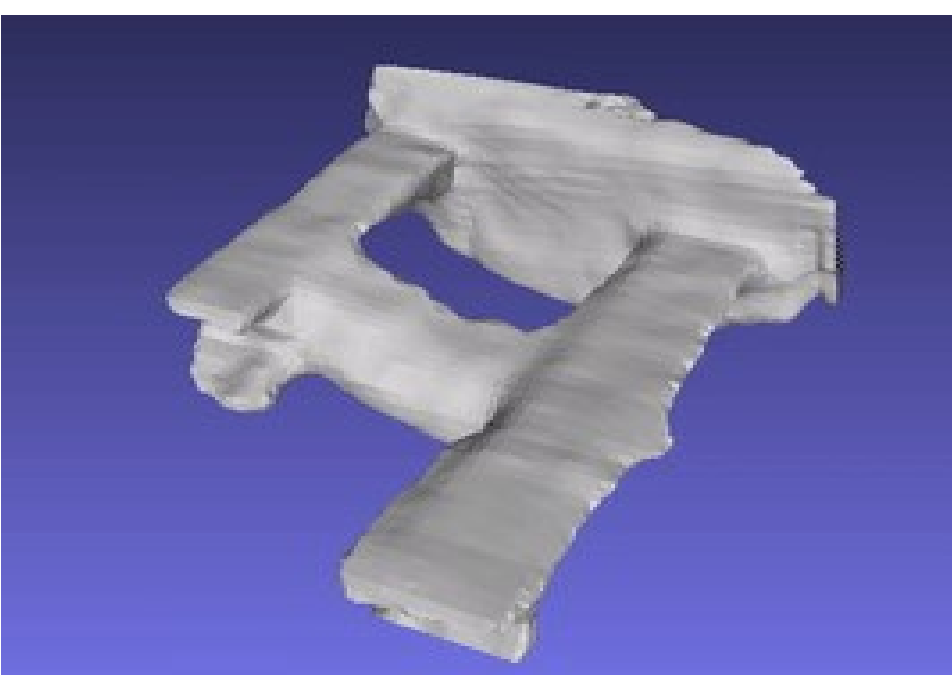
back-
projection



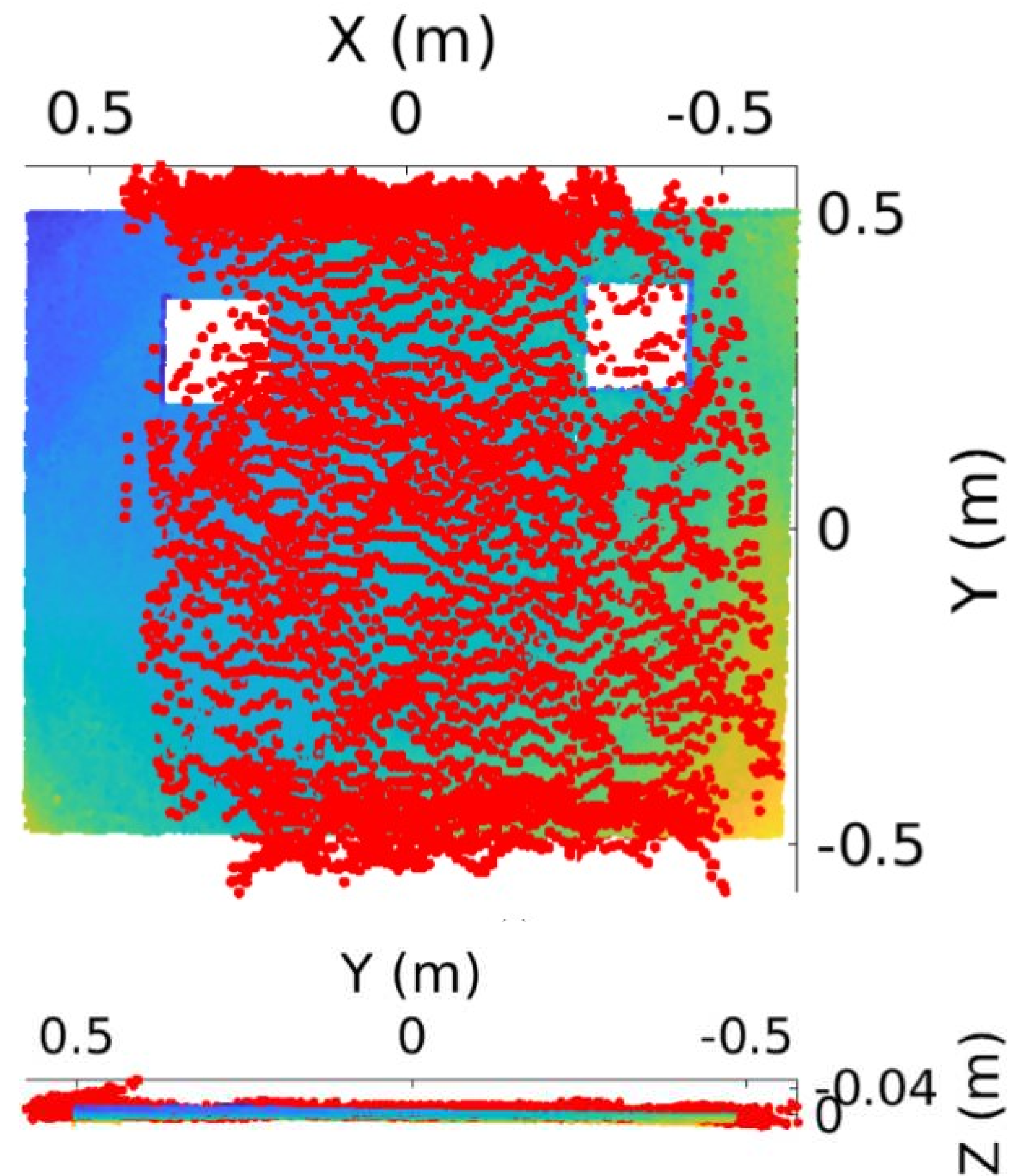
virtual
aperture



differentiable
rendering



Underwater 3D using imaging SONAR



Millimeter-accuracy underwater 3D reconstructions using data captured with an acoustic sonar mounted on robot.

Kaleidoscopic 3D scanning

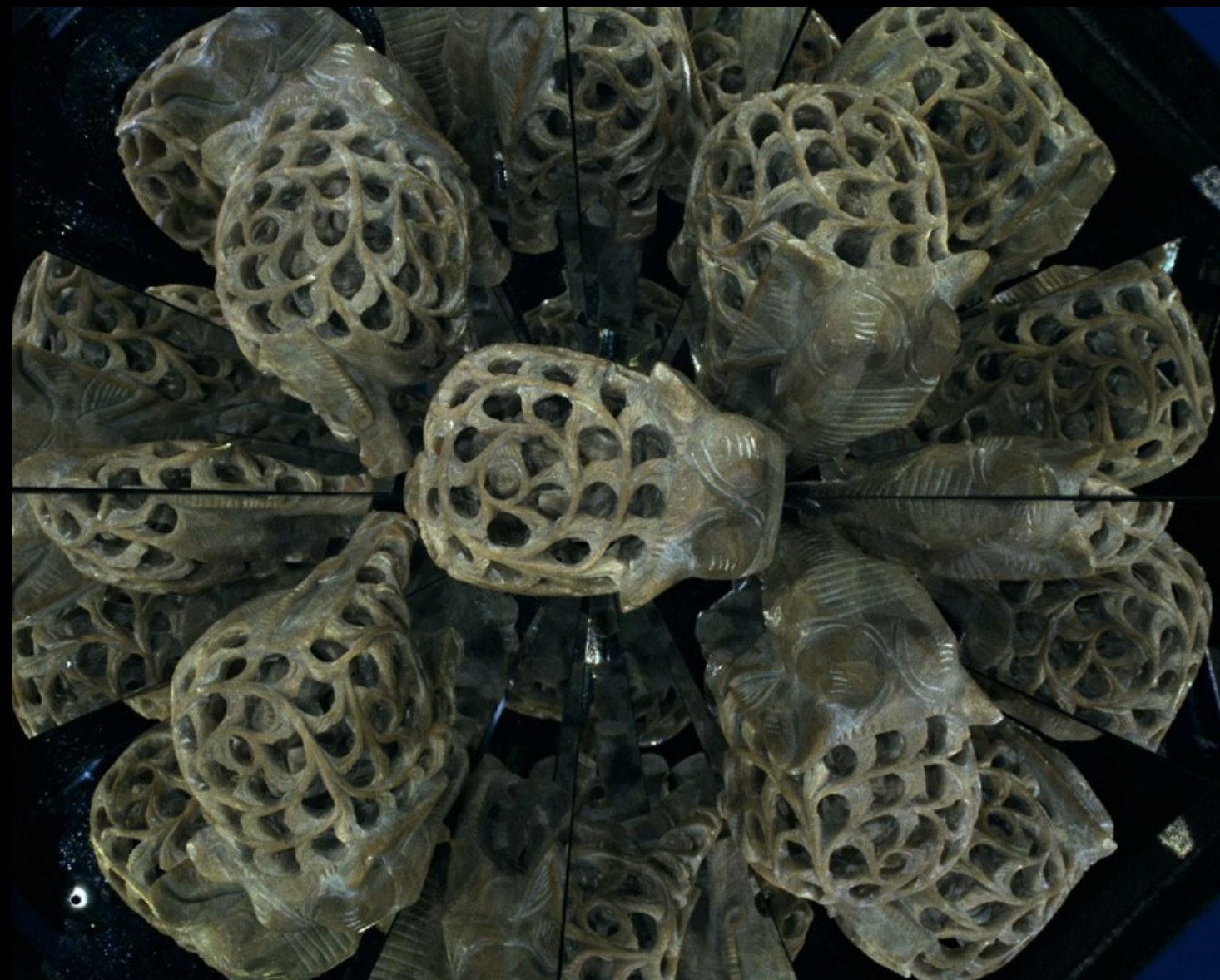


kaleidoscopic system

Kaleidoscopic 3D scanning

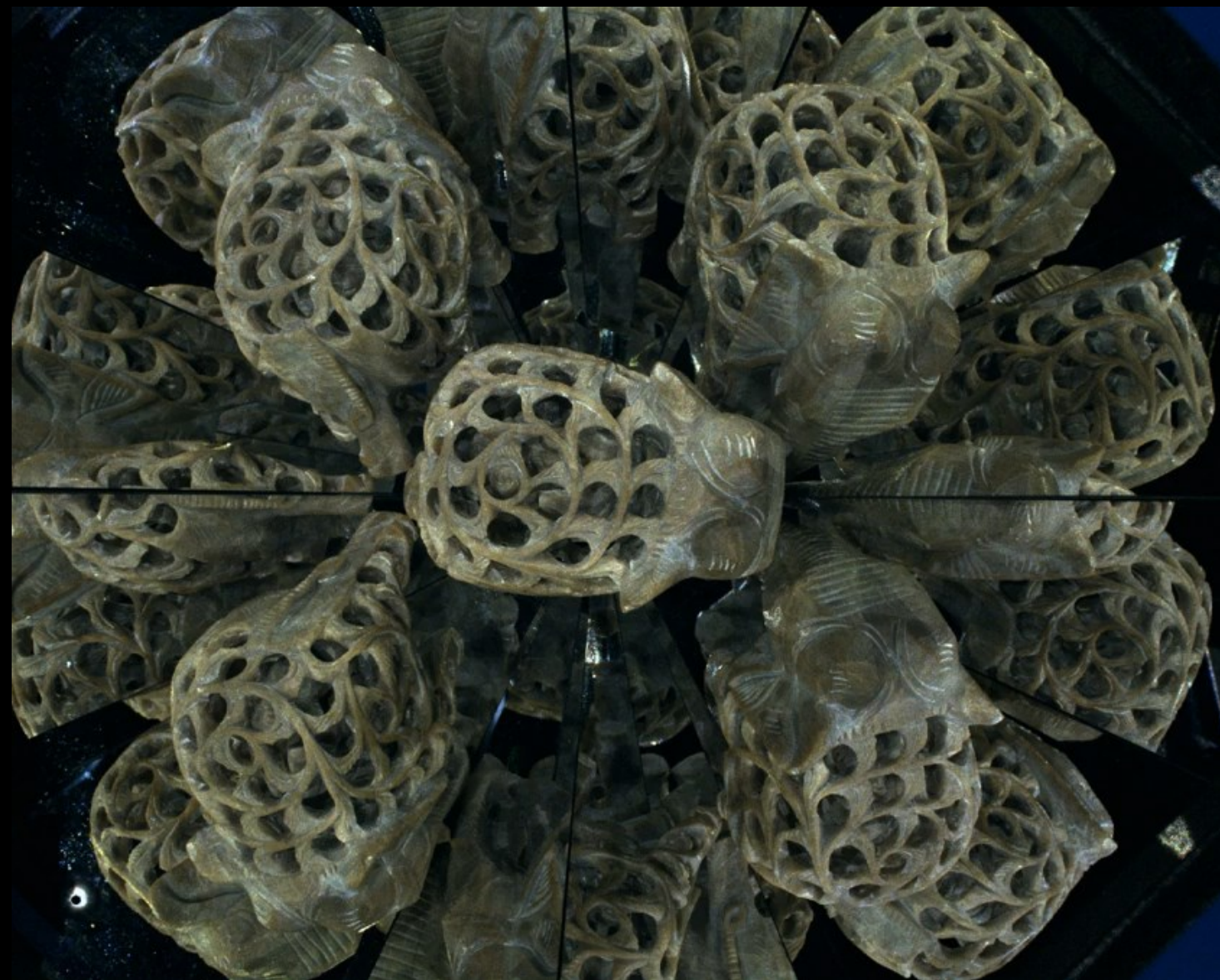


kaleidoscopic system



camera view

Kaleidoscopic 3D scanning



kaleidoscopic system

camera view

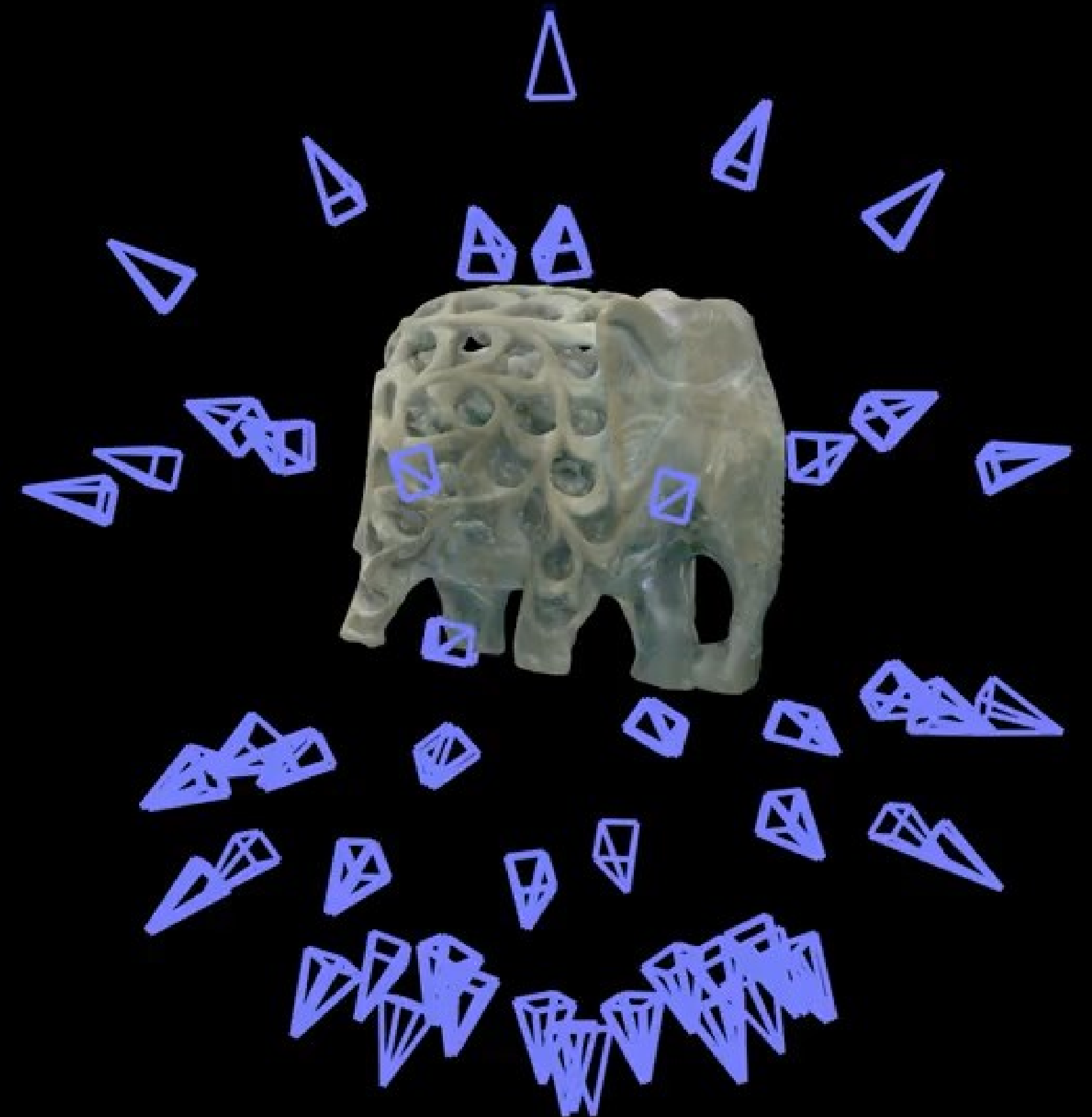
Kaleidoscopic 3D scanning



kaleidoscopic system



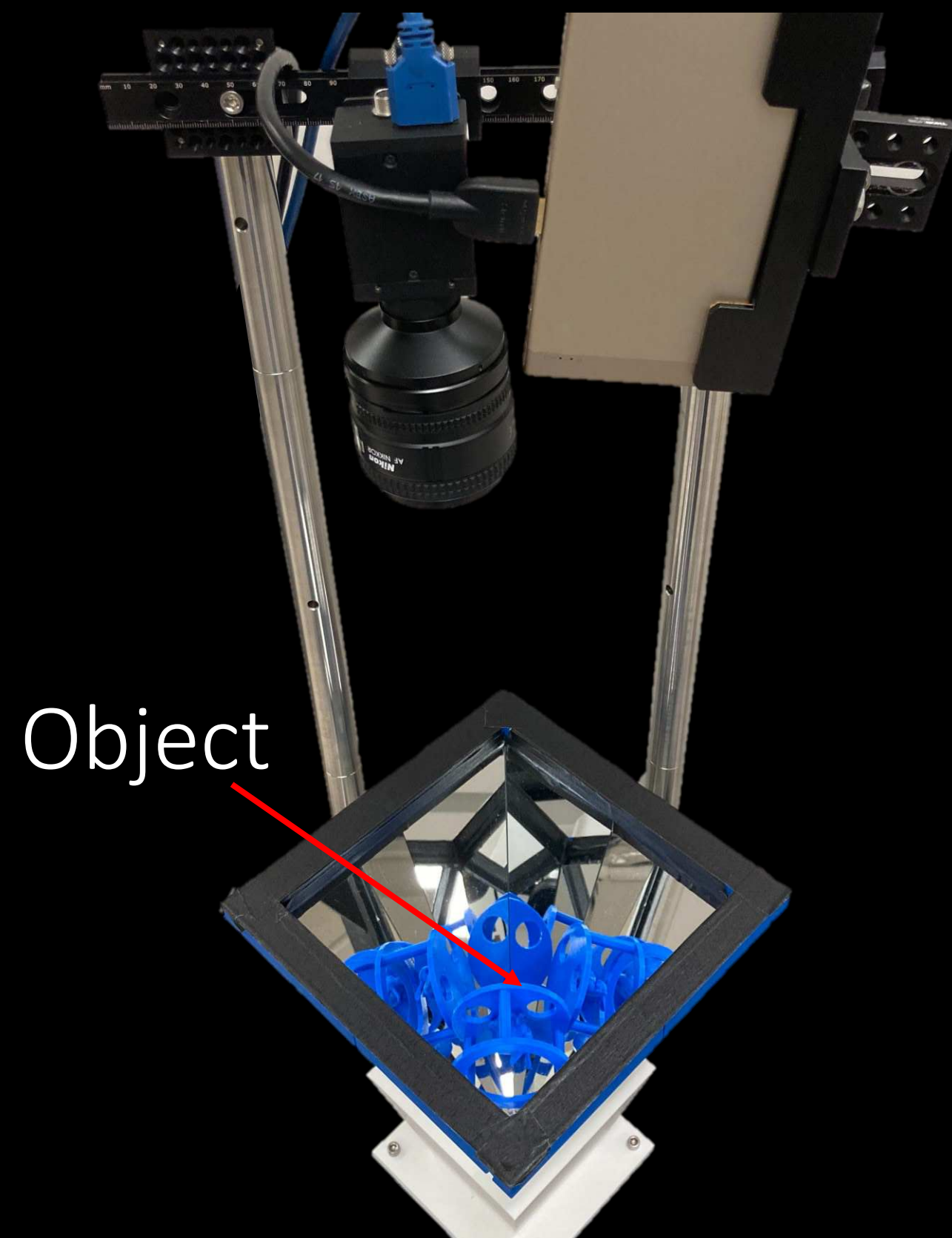
camera view



virtual cameras

Kaleidoscopic 3D scanning

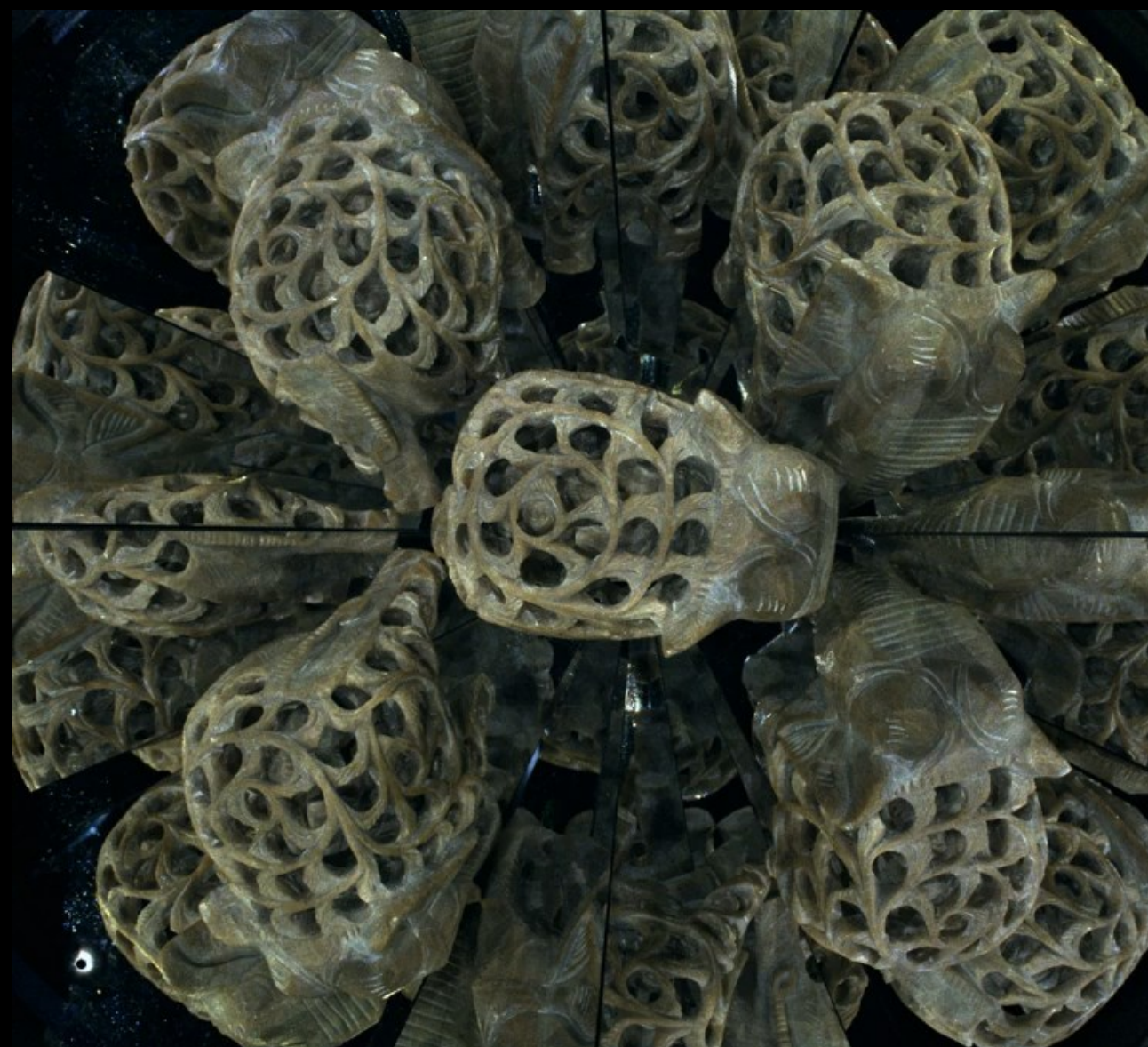
Camera Projector



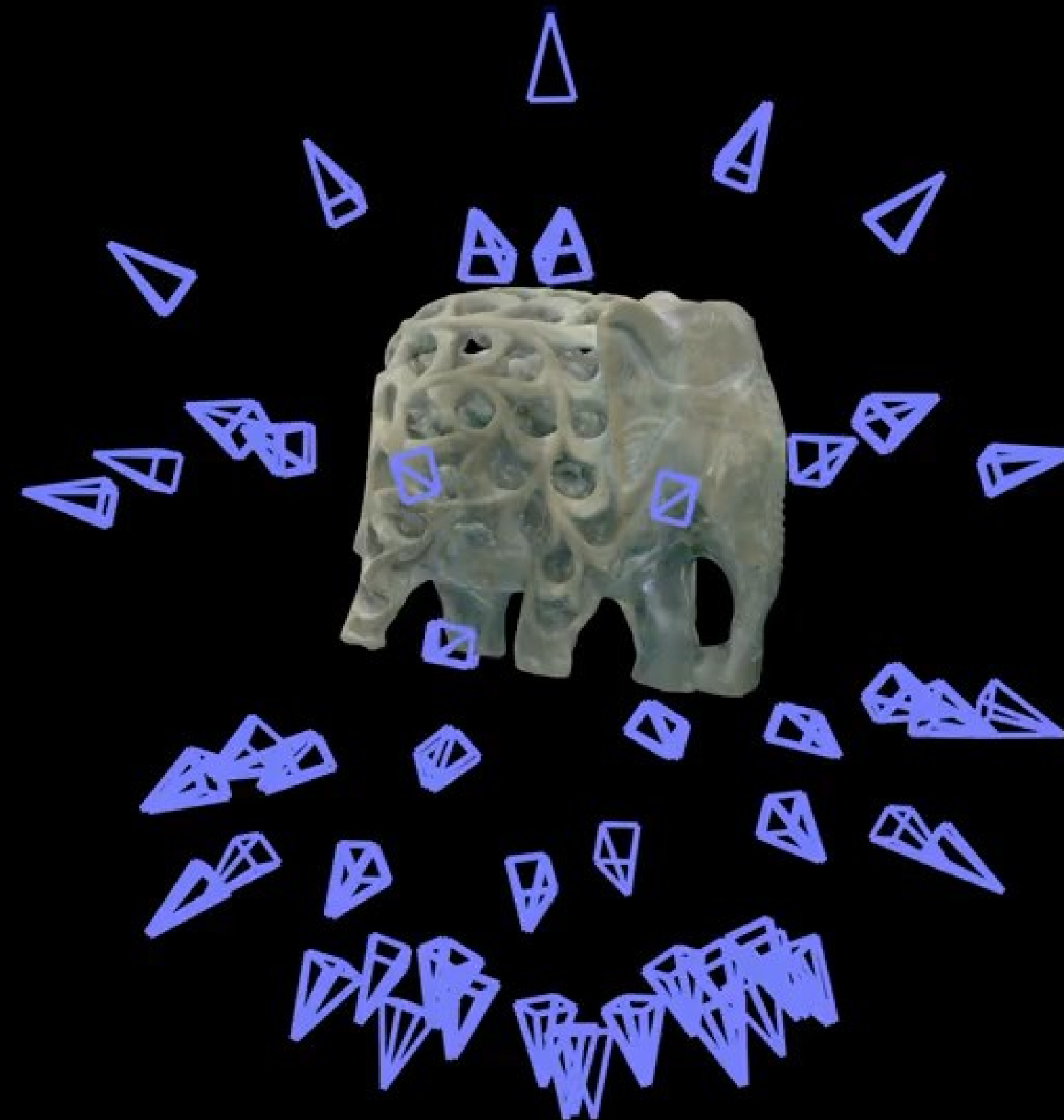
Object

Kaleidoscope

kaleidoscopic system



camera view

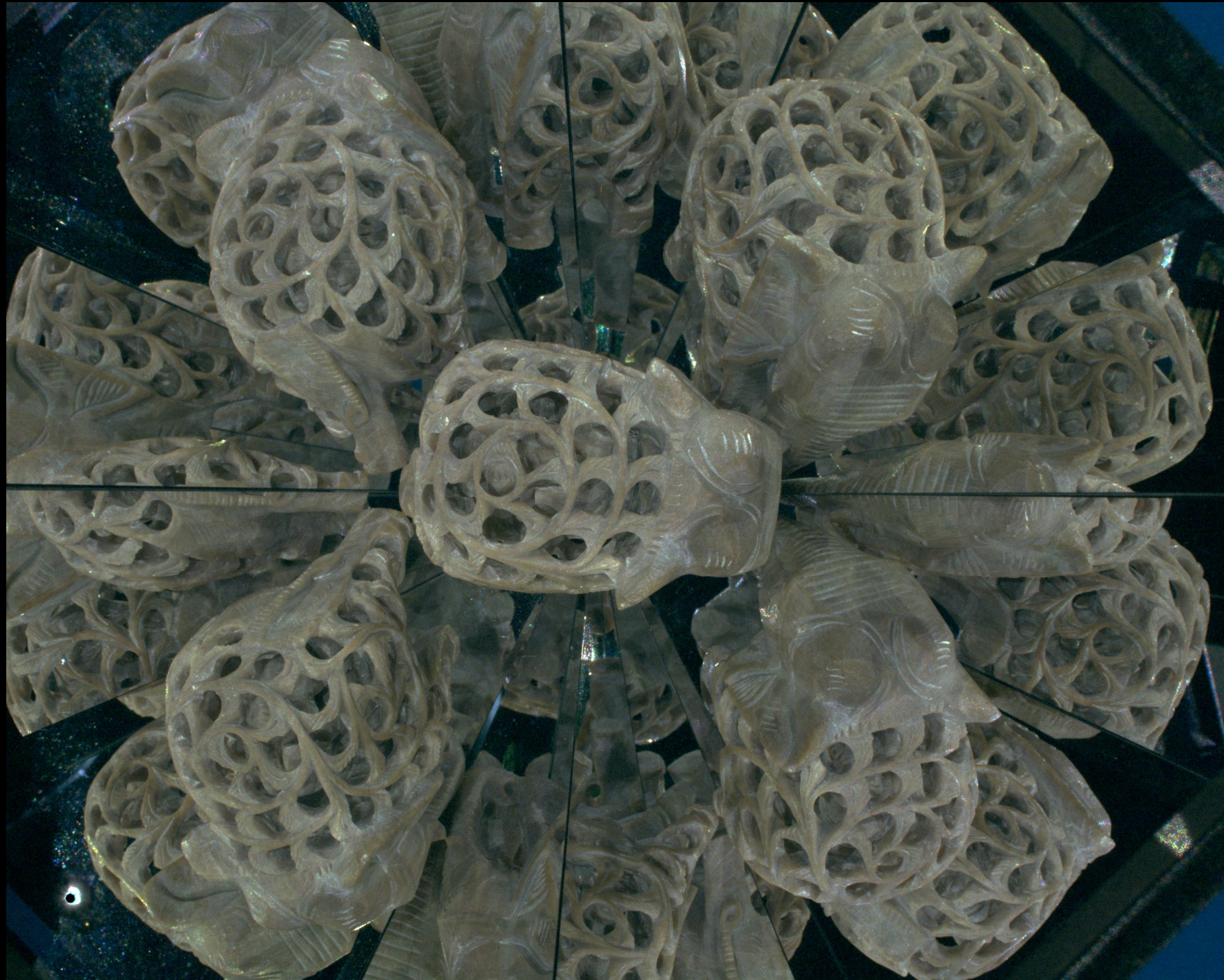


virtual cameras

[Ahn et al., CVPR 2023, Tuesday PM]

Example 3D scans

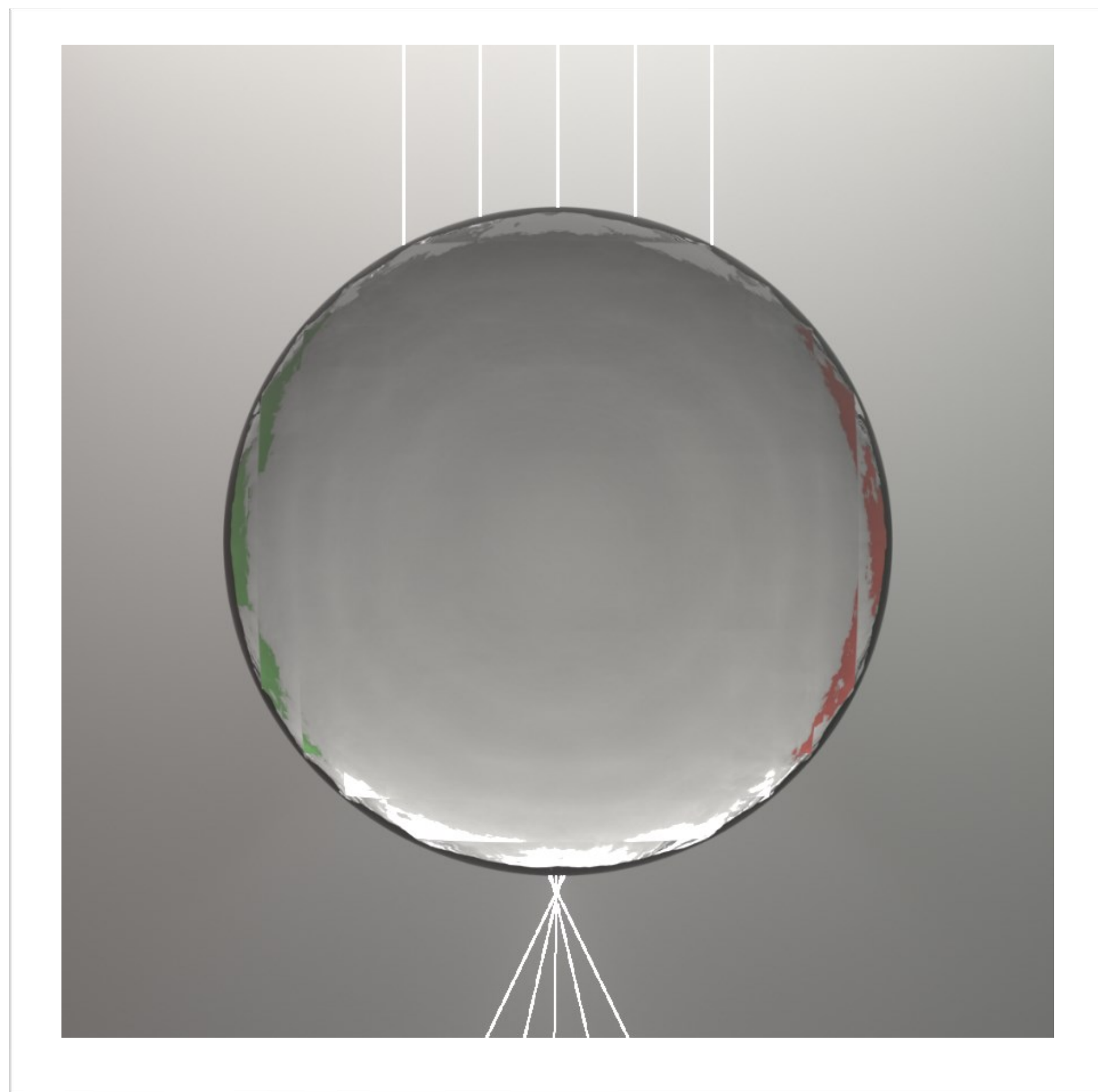
photograph



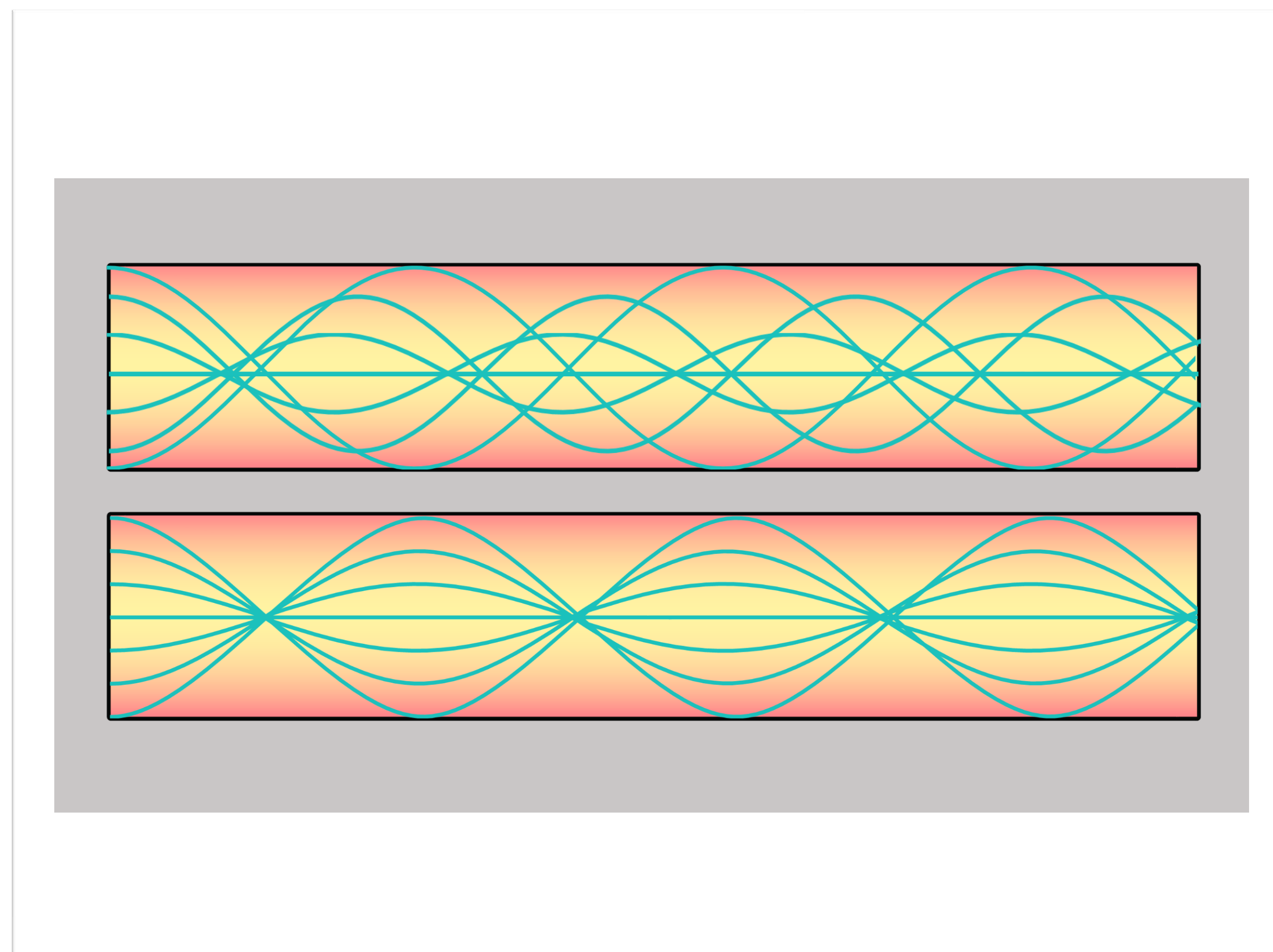
3D reconstruction



Optimizing Gradient-Index (GRIN) Optics



Luneburg Lens



GRIN Fiber

Nonlinear Ray Tracing



$\eta(\mathbf{x})$: refractive index of the volume at location, \mathbf{x}

Nonlinear Ray Tracing



Nonlinear Ray Tracing



Nonlinear Ray Tracing



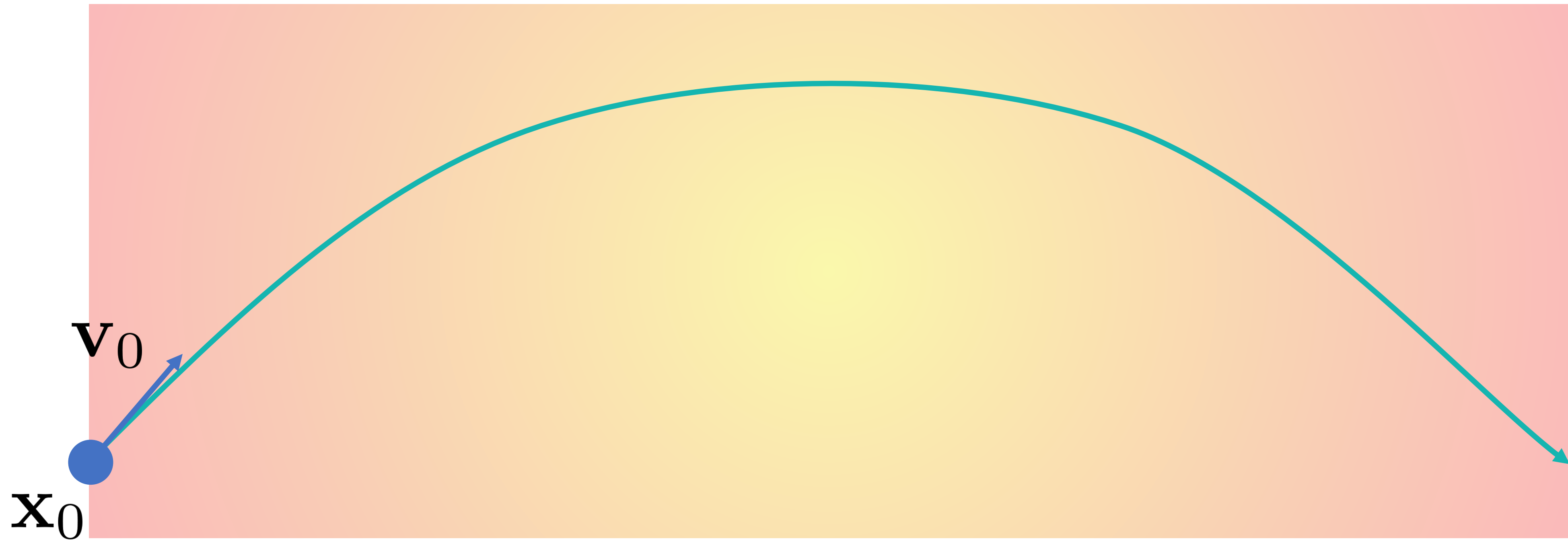
Nonlinear Ray Tracing



$$\frac{d\mathbf{x}}{dt} = \mathbf{v}$$

$$\frac{d\mathbf{v}}{dt} = \eta \nabla \eta$$

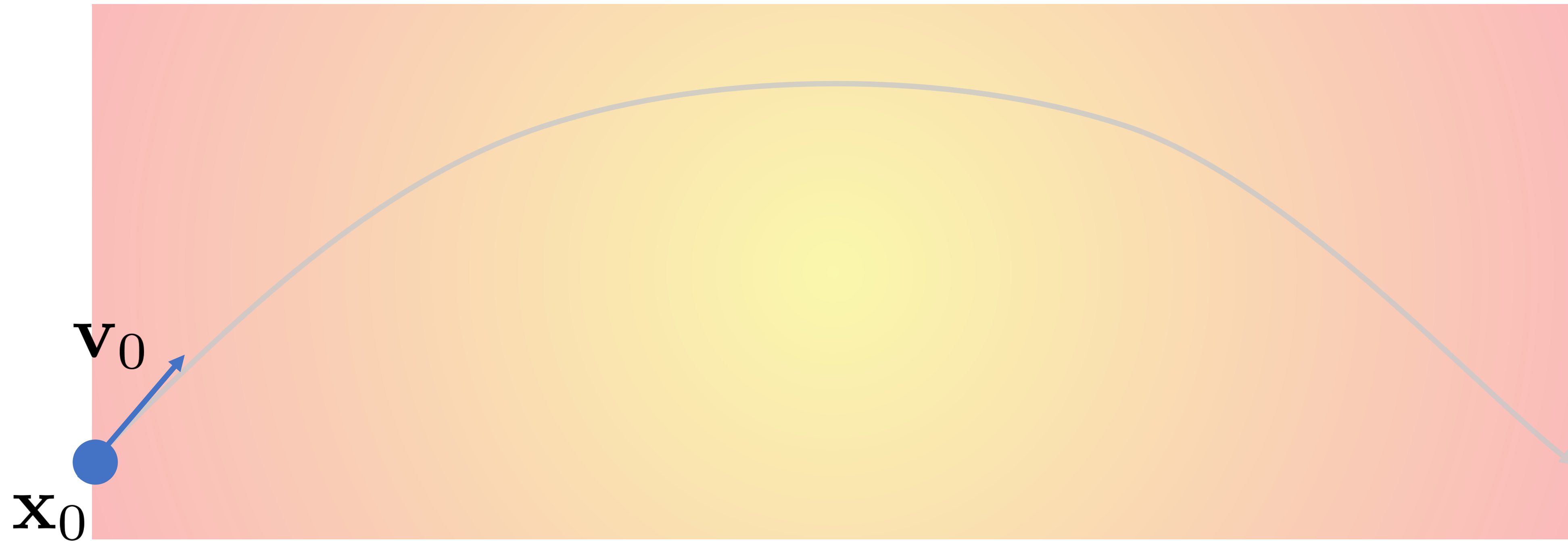
Nonlinear Ray Tracing



$$\frac{d\mathbf{x}}{dt} = \mathbf{v}$$

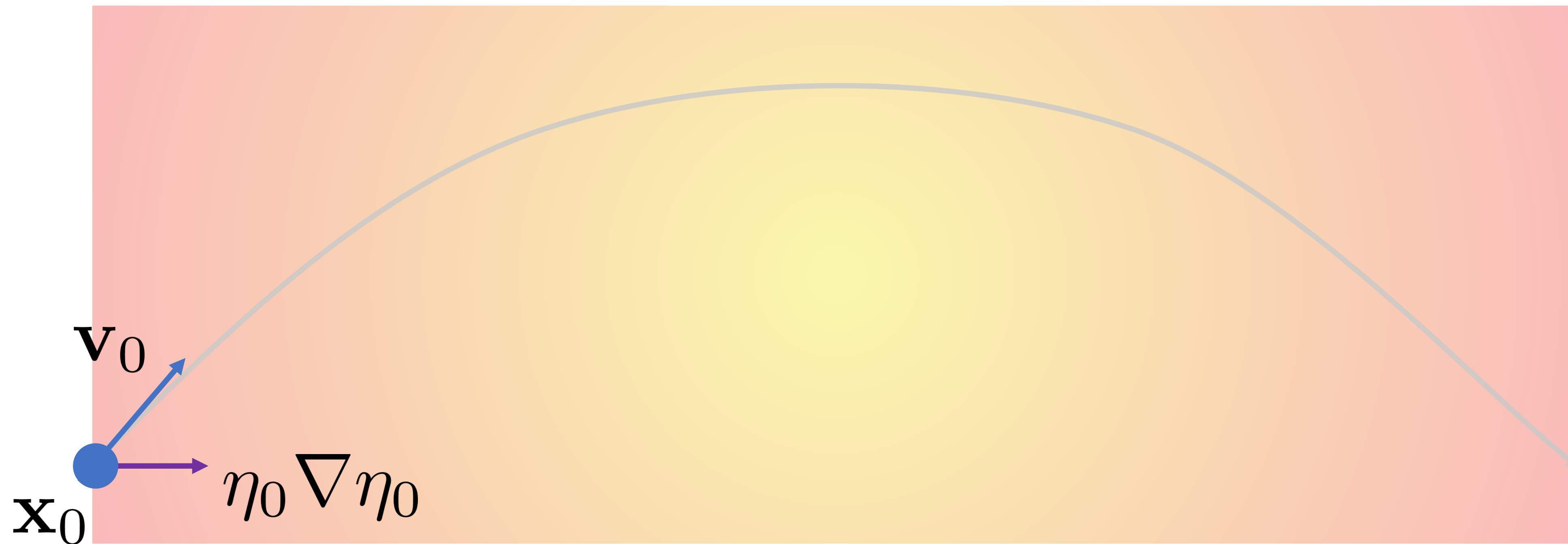
$$\frac{d\mathbf{v}}{dt} = \eta \nabla \eta$$

Nonlinear Ray Tracing



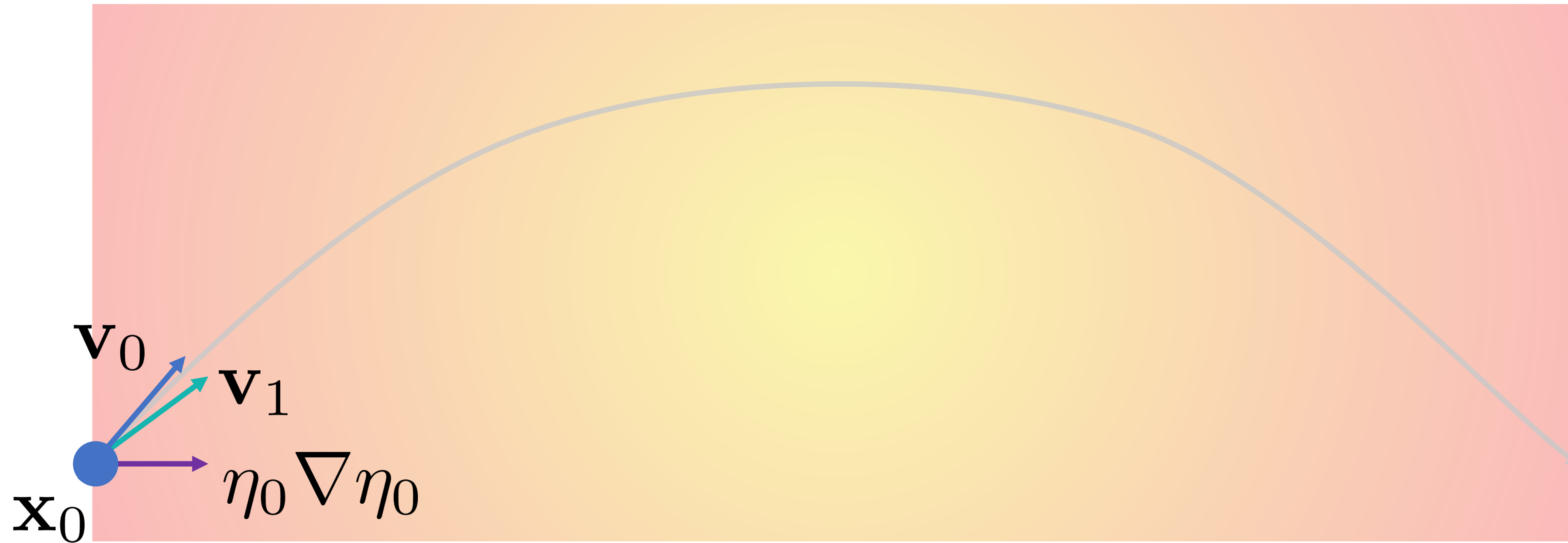
$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t \quad \mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

Nonlinear Ray Tracing



$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t \quad \mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

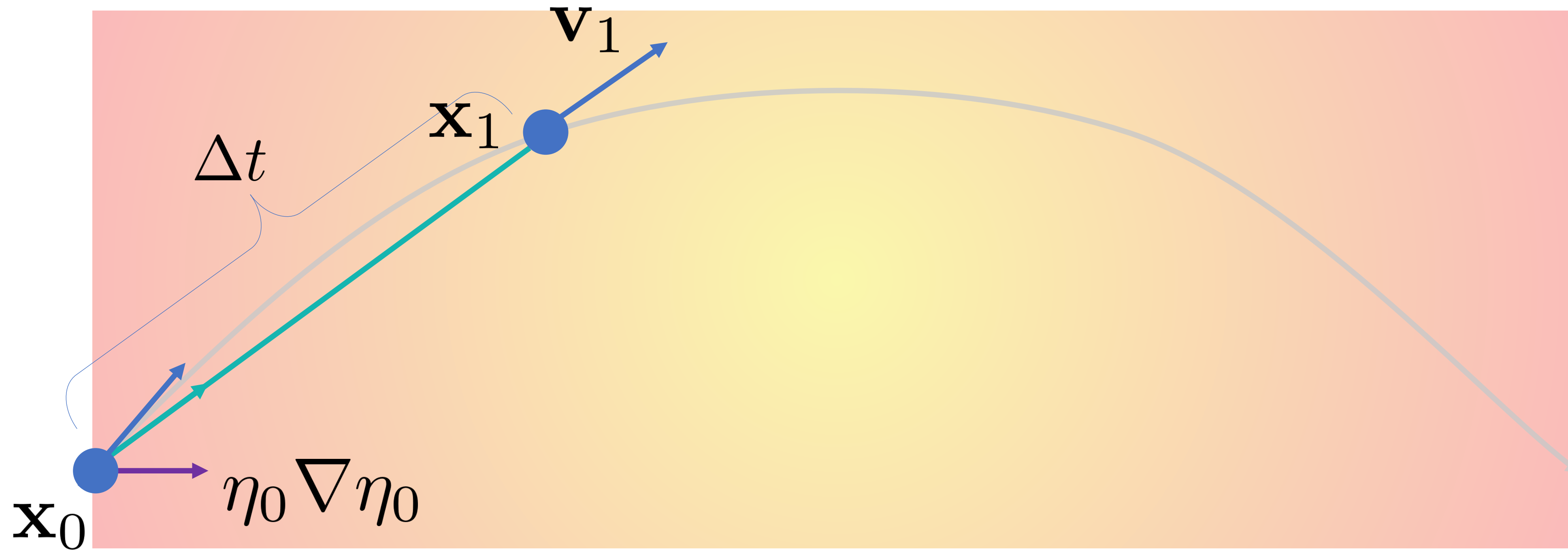
Nonlinear Ray Tracing



$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t$$

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

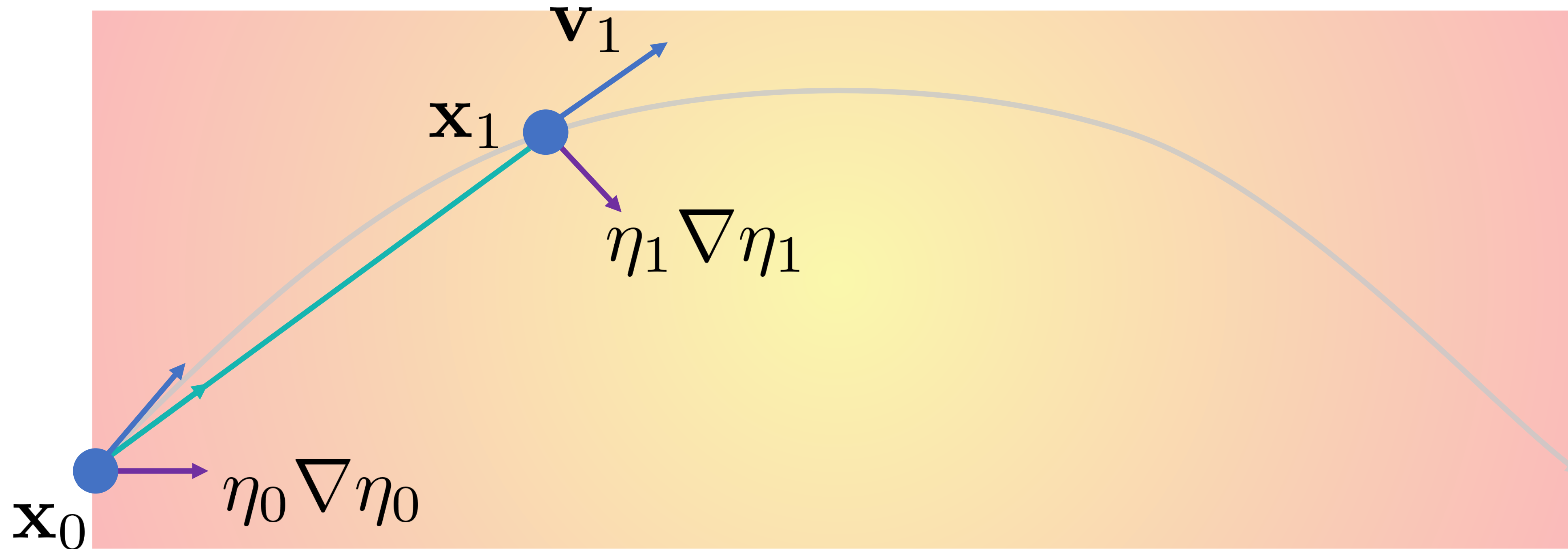
Nonlinear Ray Tracing



$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t$$

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

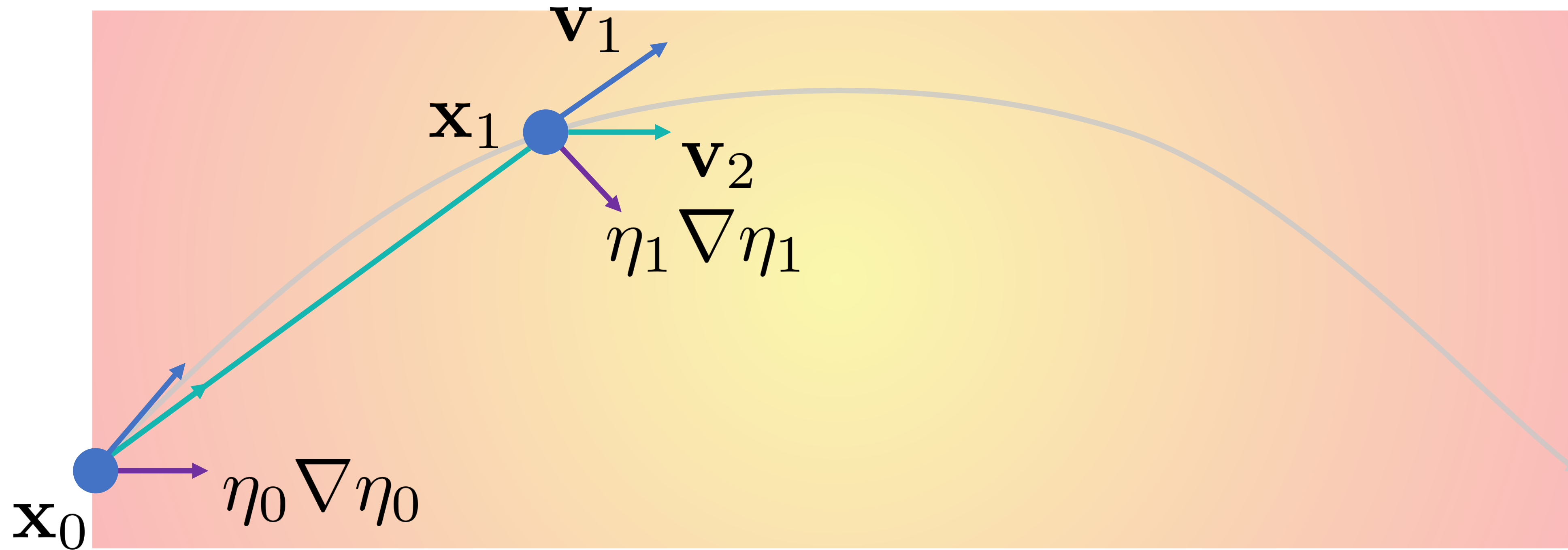
Nonlinear Ray Tracing



$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t$$

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

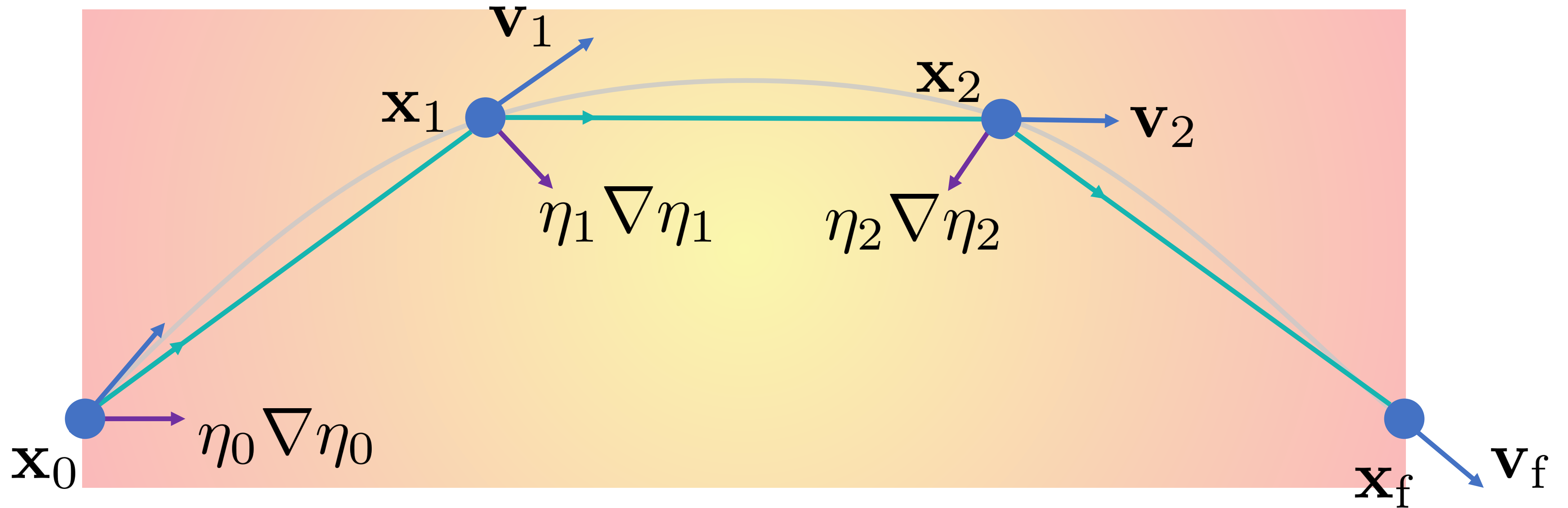
Nonlinear Ray Tracing



$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t$$

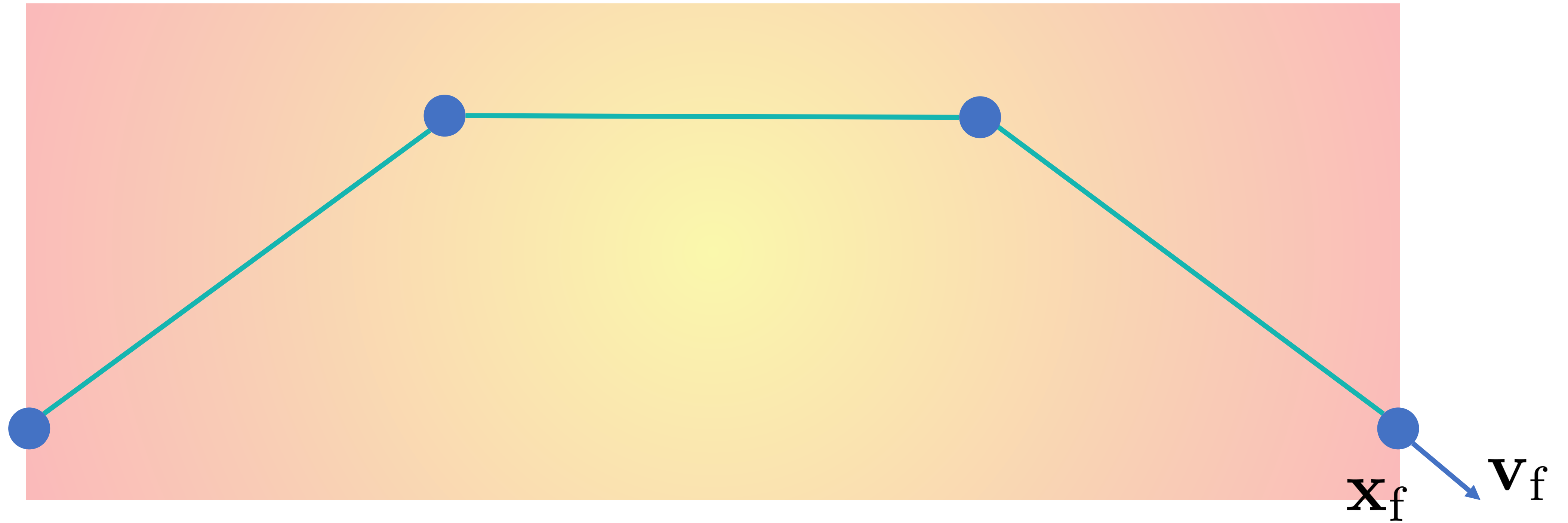
$$\mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

Nonlinear Ray Tracing

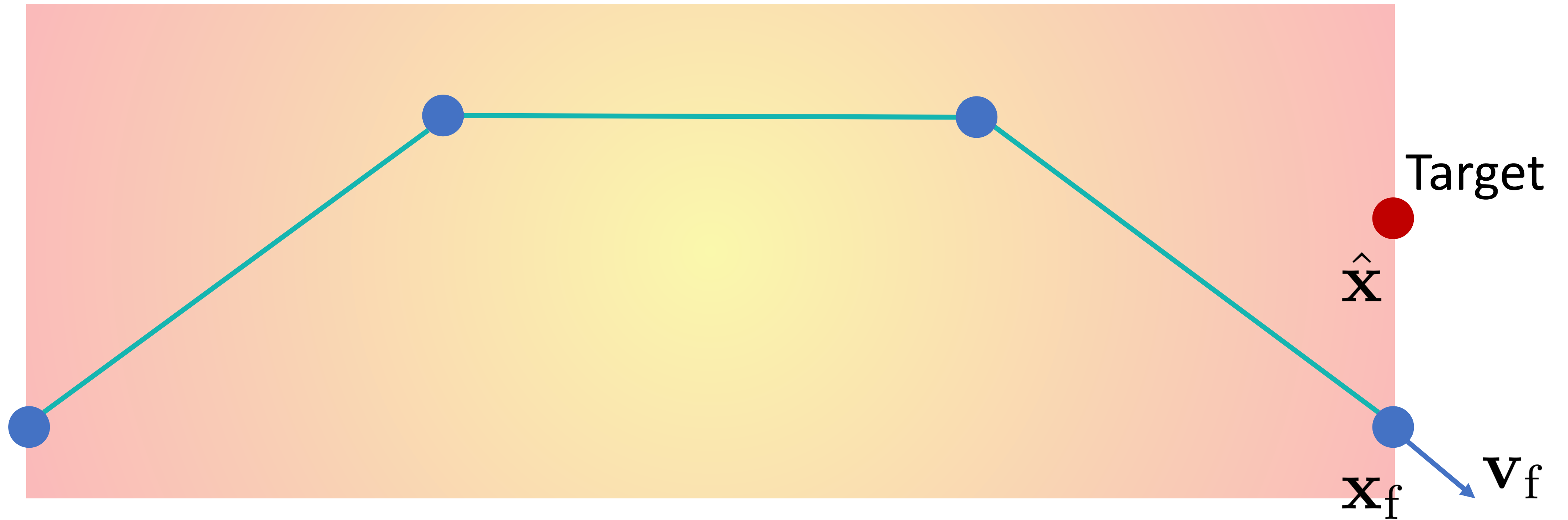


$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \Delta t \quad \mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

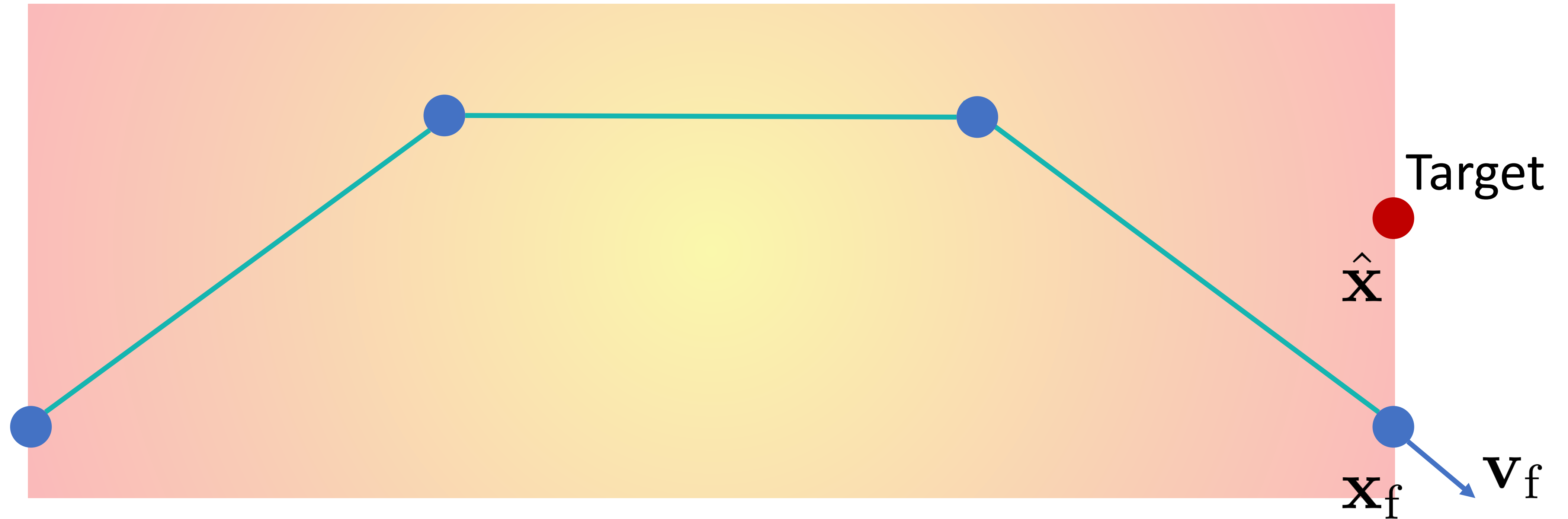
Nonlinear Ray Tracing



Nonlinear Ray Tracing

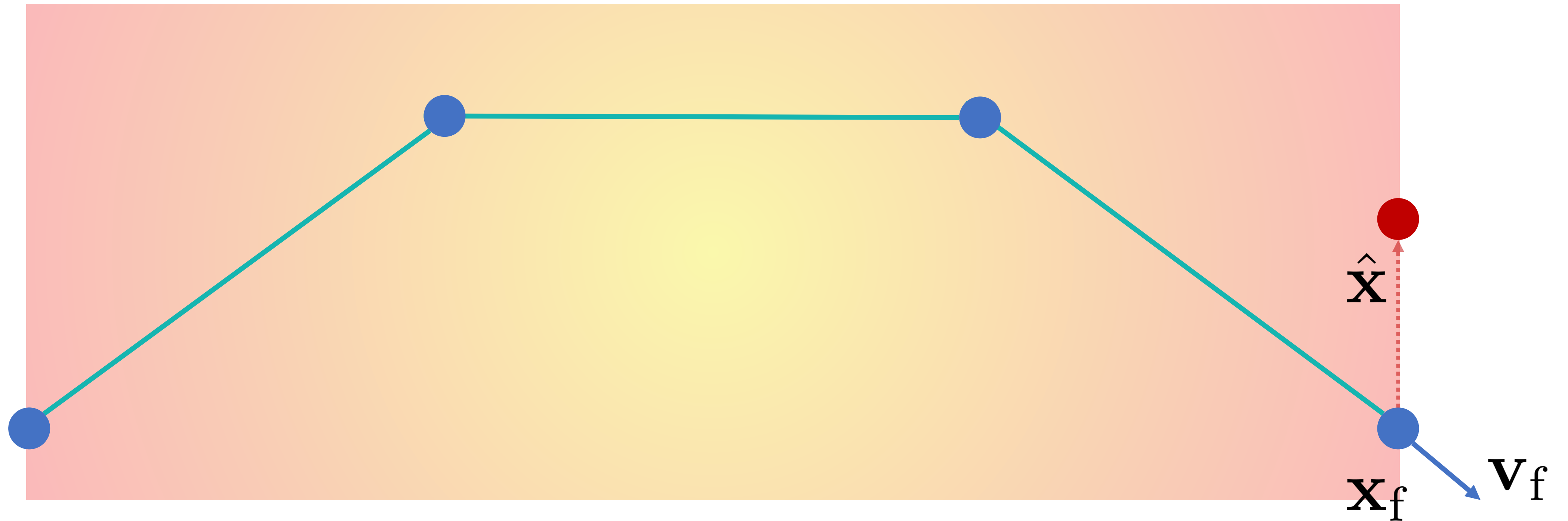


Nonlinear Ray Tracing



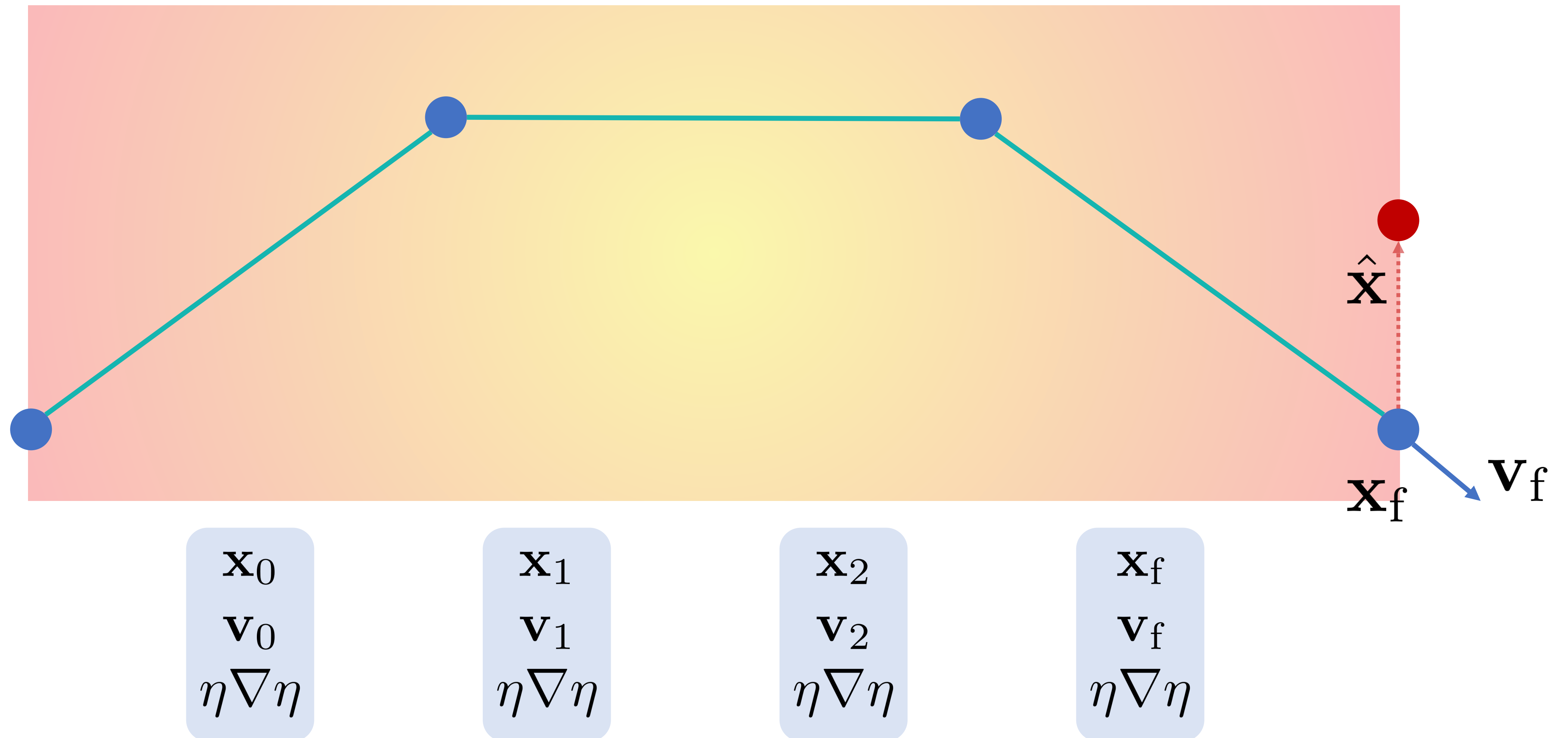
$$\min_{\eta} \|\hat{\mathbf{x}} - \mathbf{x}_f\|^2$$

Nonlinear Ray Tracing



$$\frac{d}{d\eta} \|\hat{\mathbf{x}} - \mathbf{x}_f\|^2$$

Nonlinear Ray Tracing in reverse

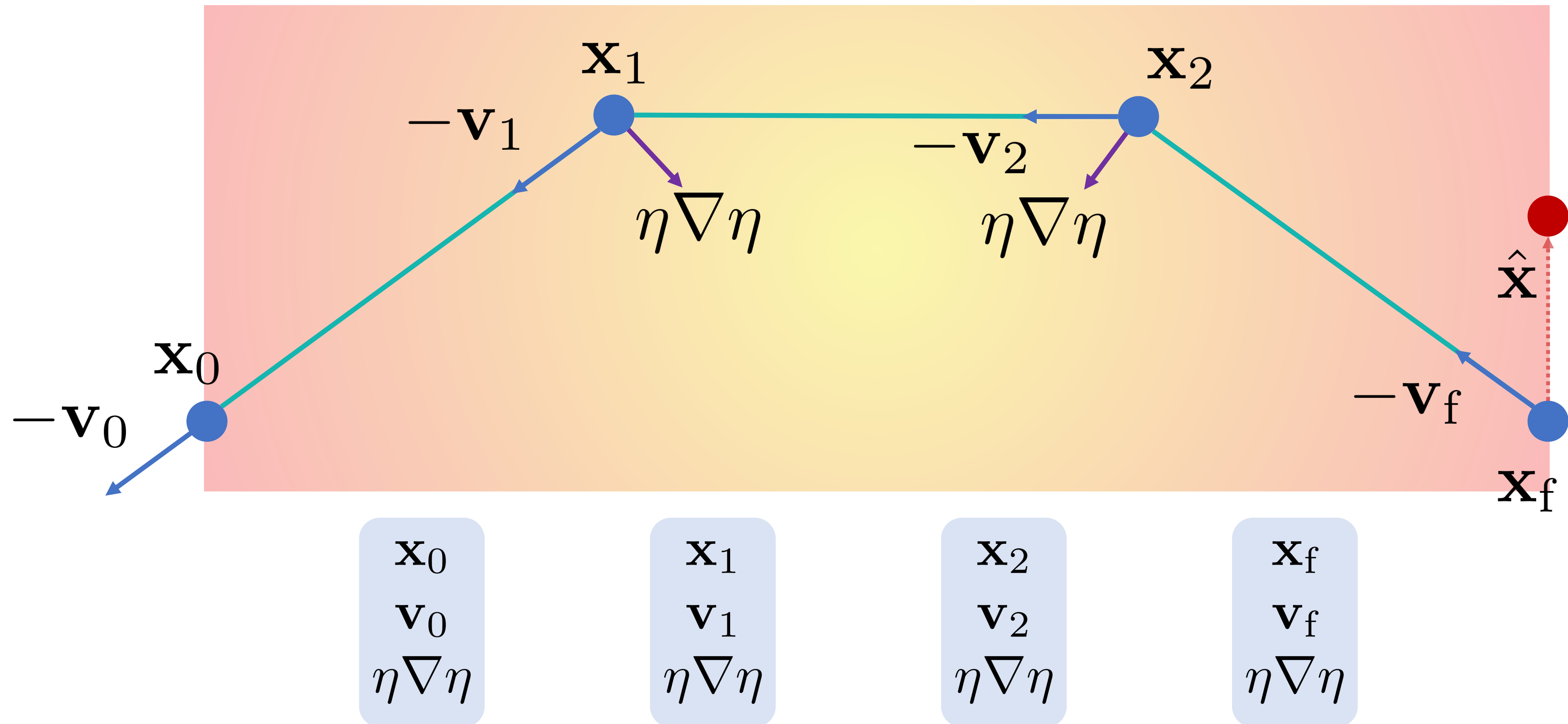


Nonlinear Ray Tracing in reverse



\mathbf{x}_f
 \mathbf{v}_f
 $\eta \nabla \eta$

Nonlinear Ray Tracing in reverse



Nonlinear Ray Tracing in reverse

$$\min_{\eta} \sum_{i=1}^N \mathcal{F}_i \left[\iint_{(\mathbf{x}_0, \mathbf{v}_0) \in \Omega} C_i \left(\mathbf{x} \left(\sigma_f; \eta, \mathbf{x}_0, \mathbf{v}_0 \right), \mathbf{v} \left(\sigma_f; \eta, \mathbf{x}_0, \mathbf{v}_0 \right) \right) d\mathbf{x}_0 d\mathbf{v}_0 \right]$$

$$\text{s.t. } \dot{\mathbf{x}} \left(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0 \right) = \mathbf{v}, \quad \forall \sigma \in [0, \sigma_f],$$

$$\dot{\mathbf{v}} \left(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0 \right) = \eta \nabla \eta, \quad \forall \sigma \in [0, \sigma_f],$$

$$\mathbf{x} \left(0; \eta, \mathbf{x}_0, \mathbf{v}_0 \right) = \mathbf{x}_0,$$

$$\mathbf{v} \left(0; \eta, \mathbf{x}_0, \mathbf{v}_0 \right) = \mathbf{v}_0,$$

(15)

\mathbf{x}_0

$\hat{\mathbf{x}}$

$$\dot{\lambda} = - \left(\nabla \eta \left(\nabla \eta \right)^\top + \eta \text{Hess} \left(\eta \right) \right) \mu, \quad \forall \sigma \in [0, \sigma_f] \quad (19)$$

$$\dot{\mu} = -\lambda, \quad \forall \sigma \in [0, \sigma_f] \quad (20)$$

$$\lambda \left(\sigma_f \right) = \frac{\partial \mathcal{C}}{\partial \mathbf{x}}, \quad (21)$$

$$\mu \left(\sigma_f \right) = \frac{\partial \mathcal{C}}{\partial \mathbf{v}}. \quad (22)$$

$\eta \nabla \eta$

$\eta \nabla \eta$

$$\mathbf{x}_{i-1} = \mathbf{x}_i - \mathbf{v}_i \Delta \sigma, \quad (28)$$

$$\mathbf{v}_{i-1} = \mathbf{v}_i - \eta \left(\mathbf{x}_{i-1} \right) \nabla \eta \left(\mathbf{x}_{i-1} \right) \Delta \sigma, \quad (29)$$

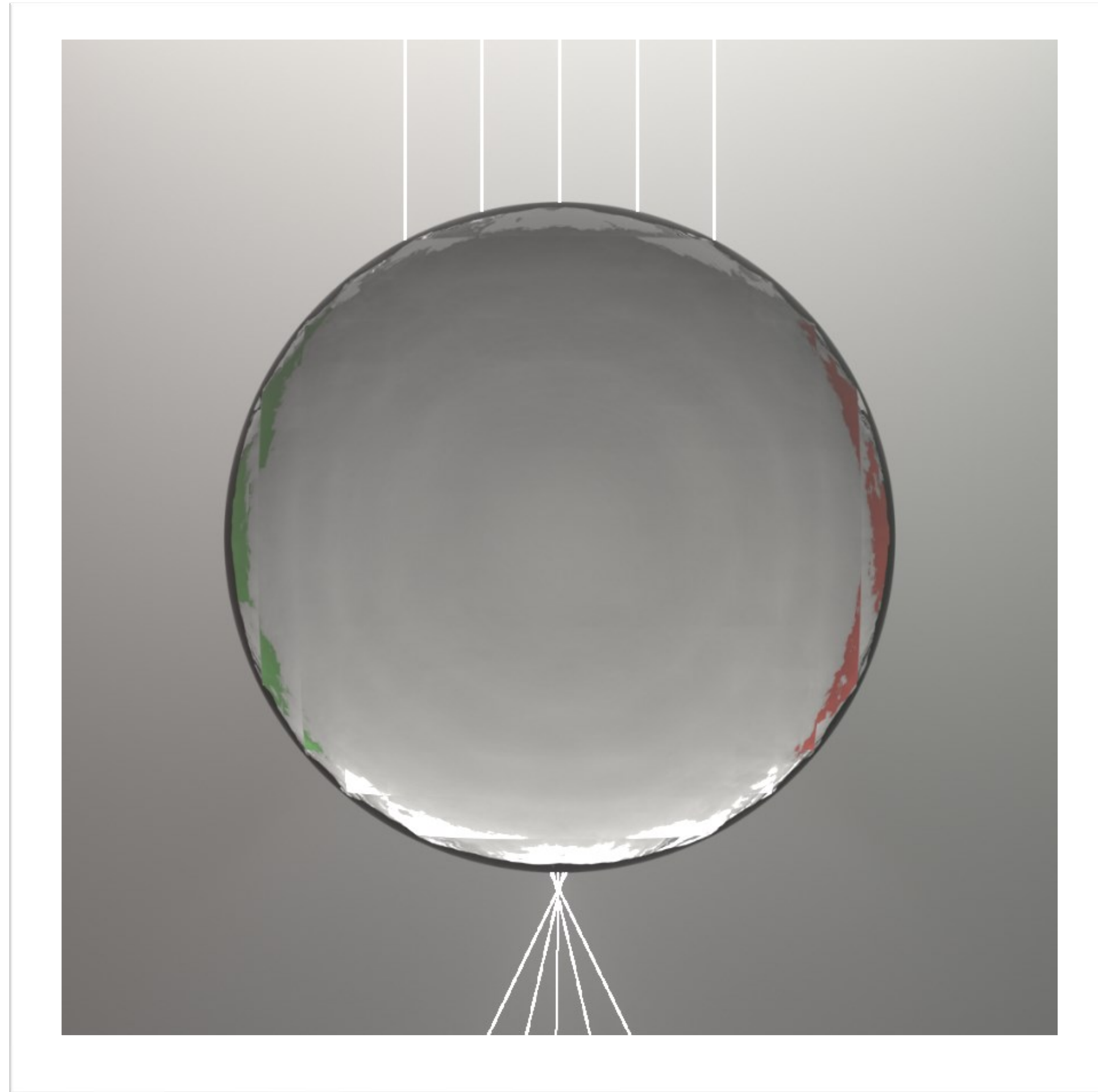
$$\lambda_{i-1} = \lambda_i + \left(\nabla \eta \left(\mathbf{x}_{i-1} \right) \left(\nabla \eta \left(\mathbf{x}_{i-1} \right) \right)^\top + \eta \left(\mathbf{x}_{i-1} \right) \text{Hess} \left(\eta \left(\mathbf{x}_{i-1} \right) \right) \right) \mu_i \Delta \sigma, \quad (30)$$

$$\mu_{i-1} = \mu_i + \lambda_{i-1} \Delta \sigma. \quad (31)$$

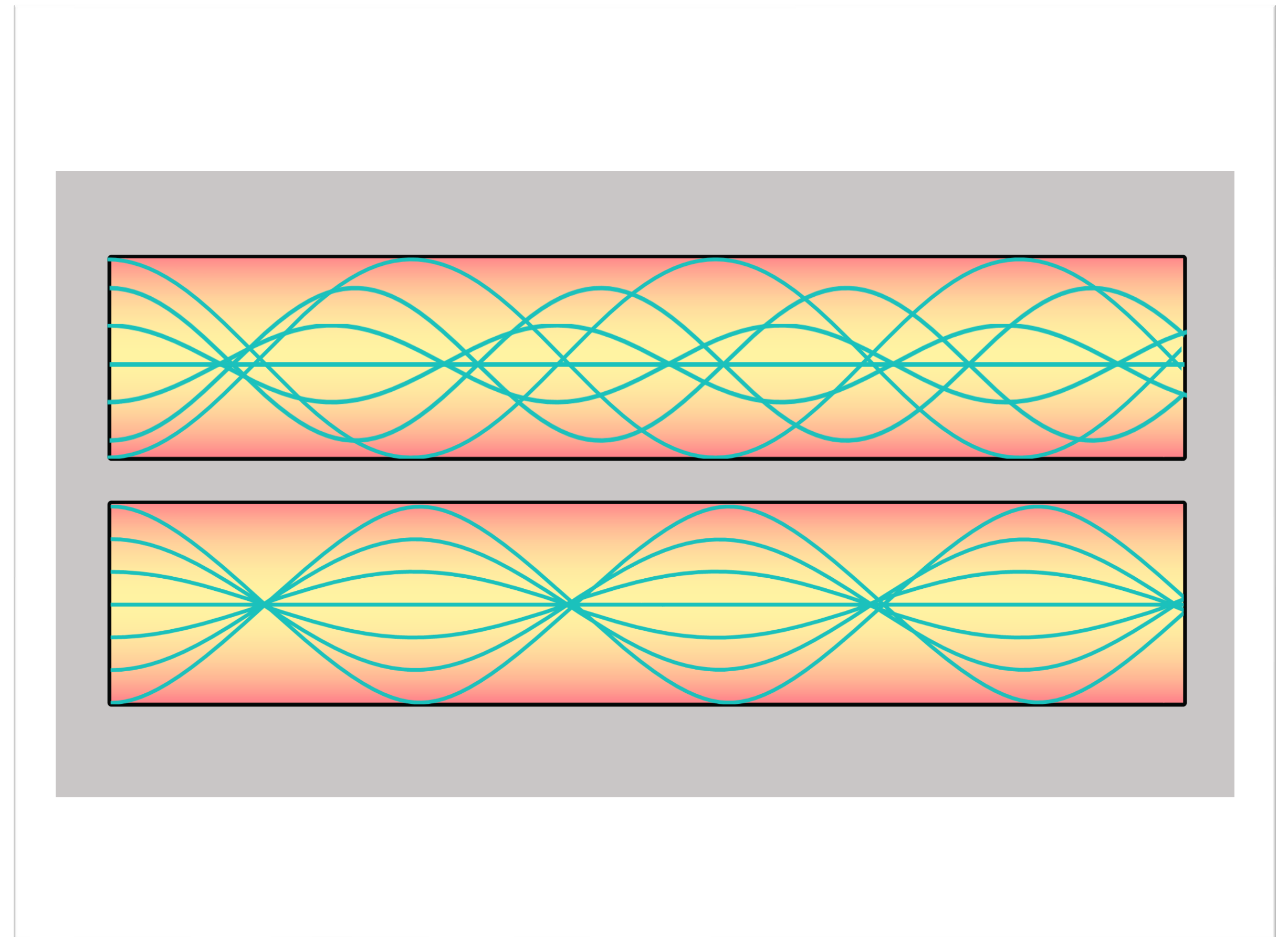
$\eta \nabla \eta$

$\eta \nabla \eta$

Optimizing Gradient-Index (GRIN) Optics

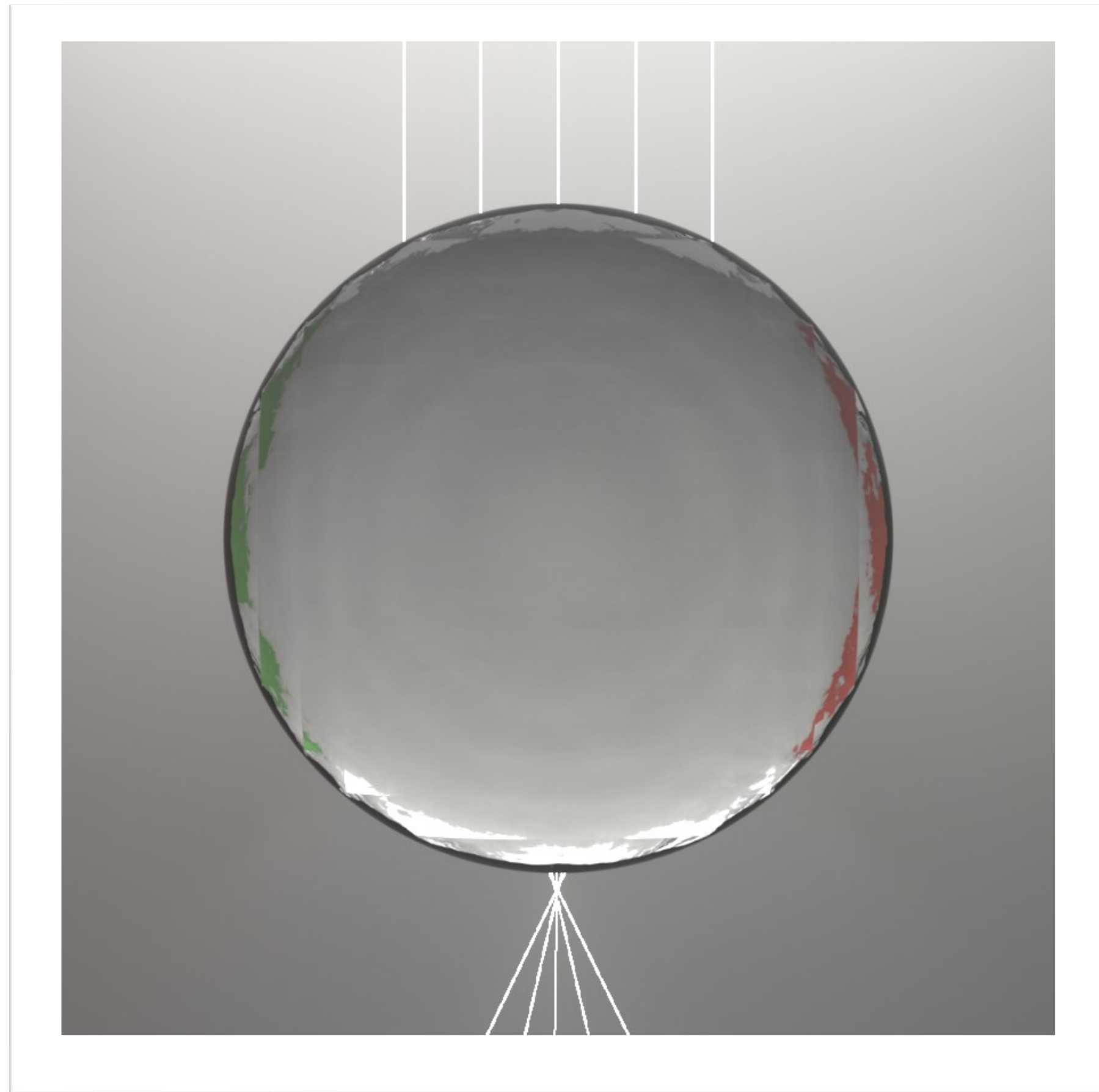


Luneburg Lens

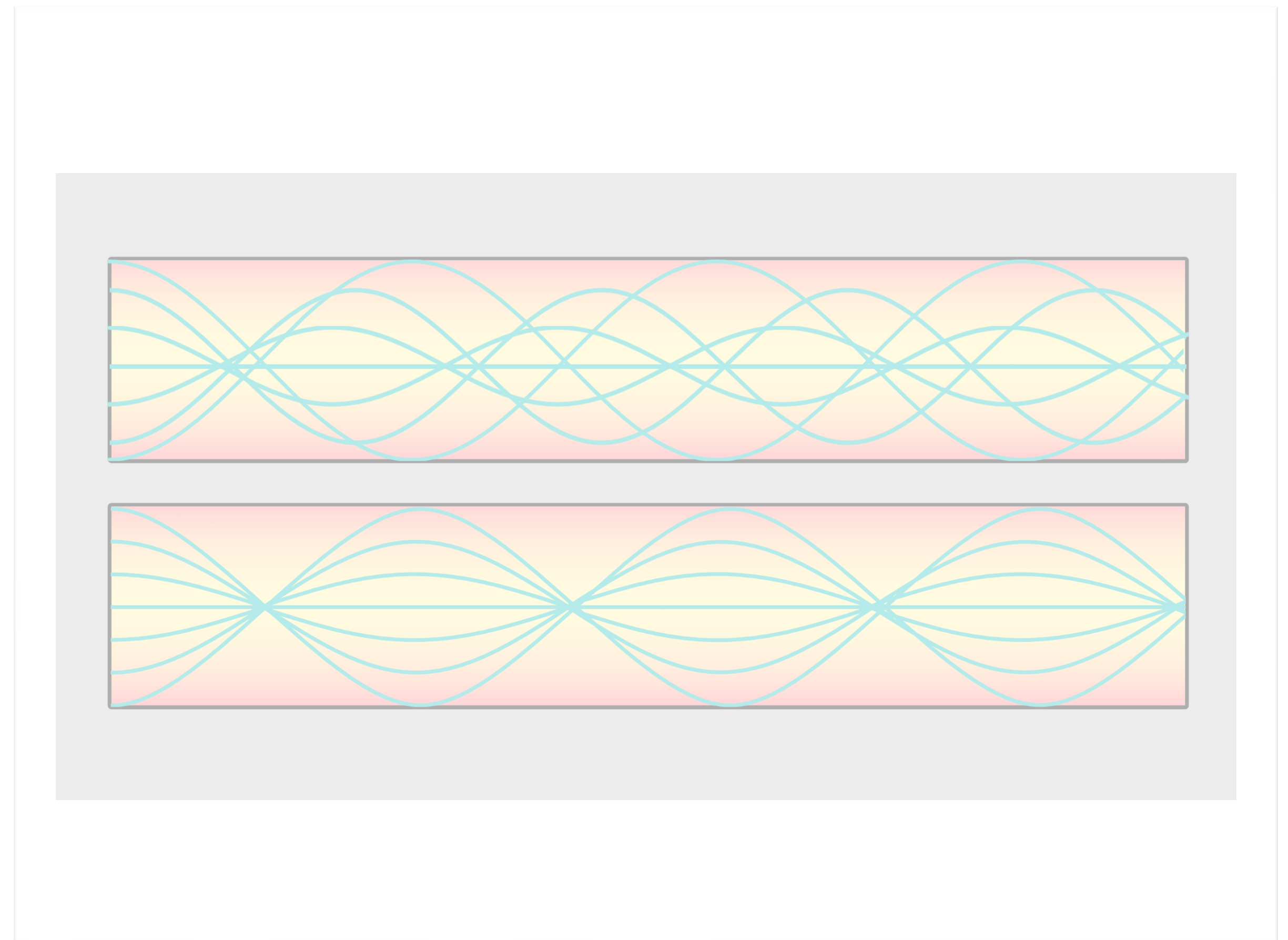


GRIN Fiber

Optimizing Gradient-Index (GRIN) Optics

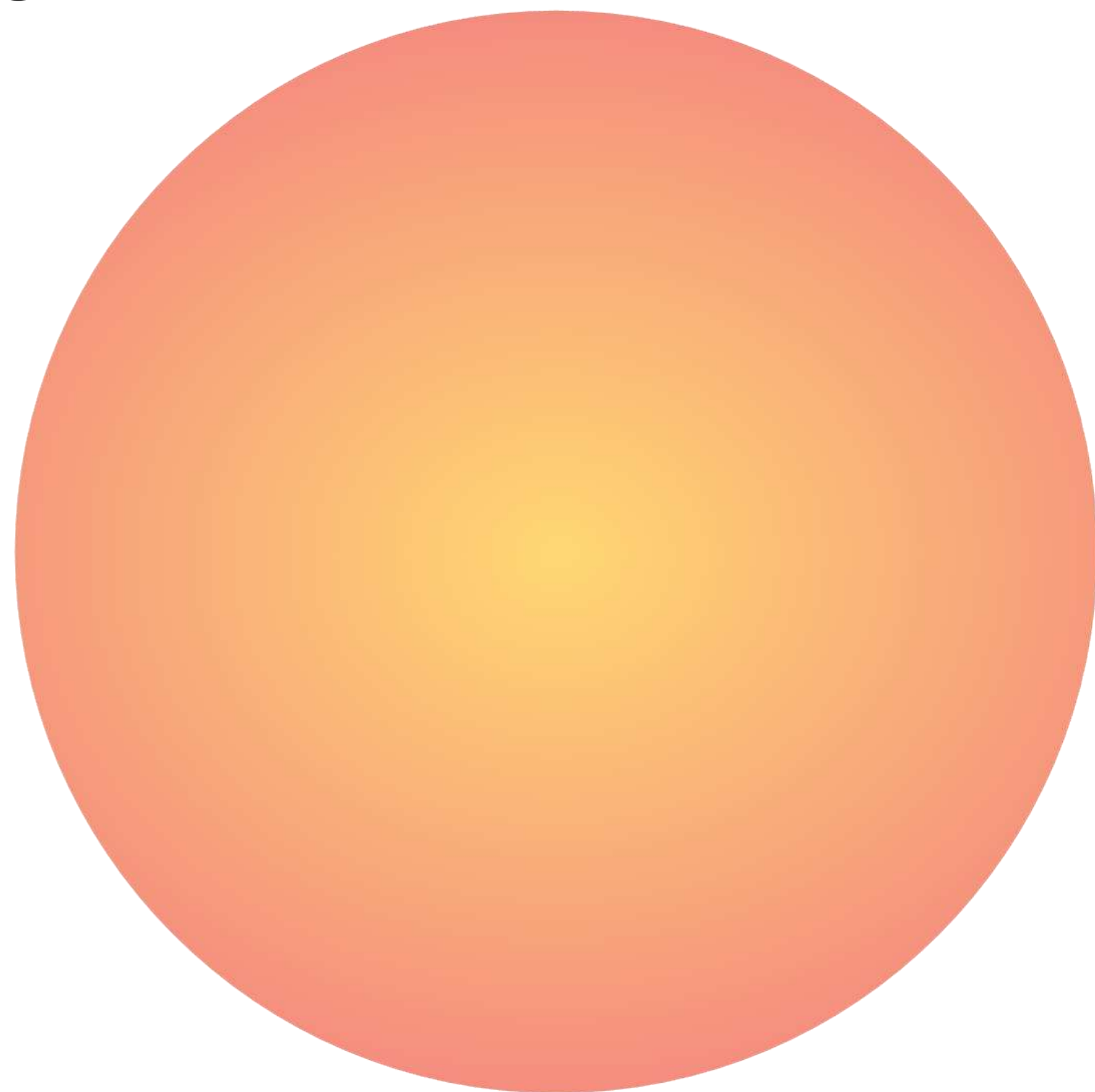


Luneburg Lens

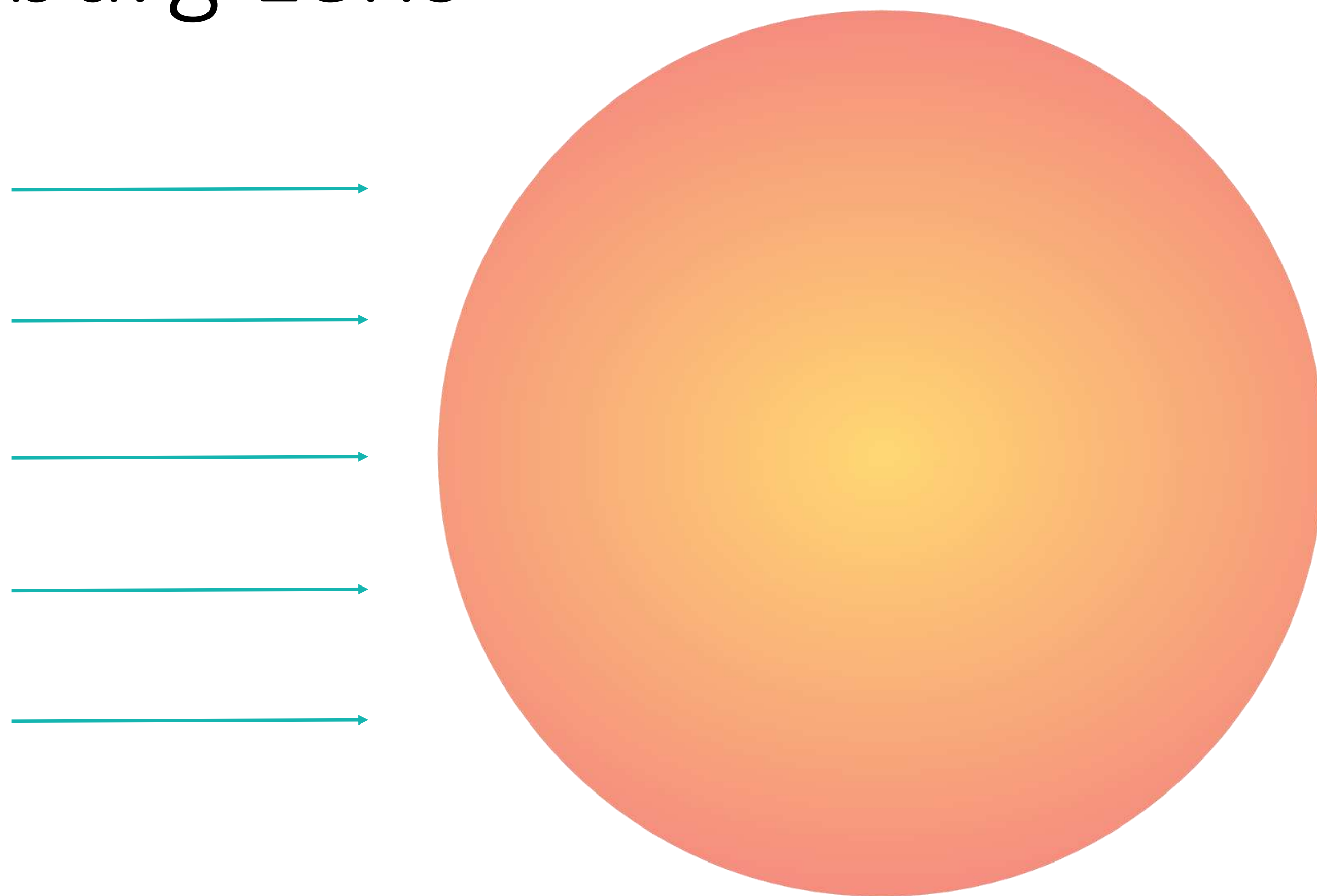


GRIN Fiber

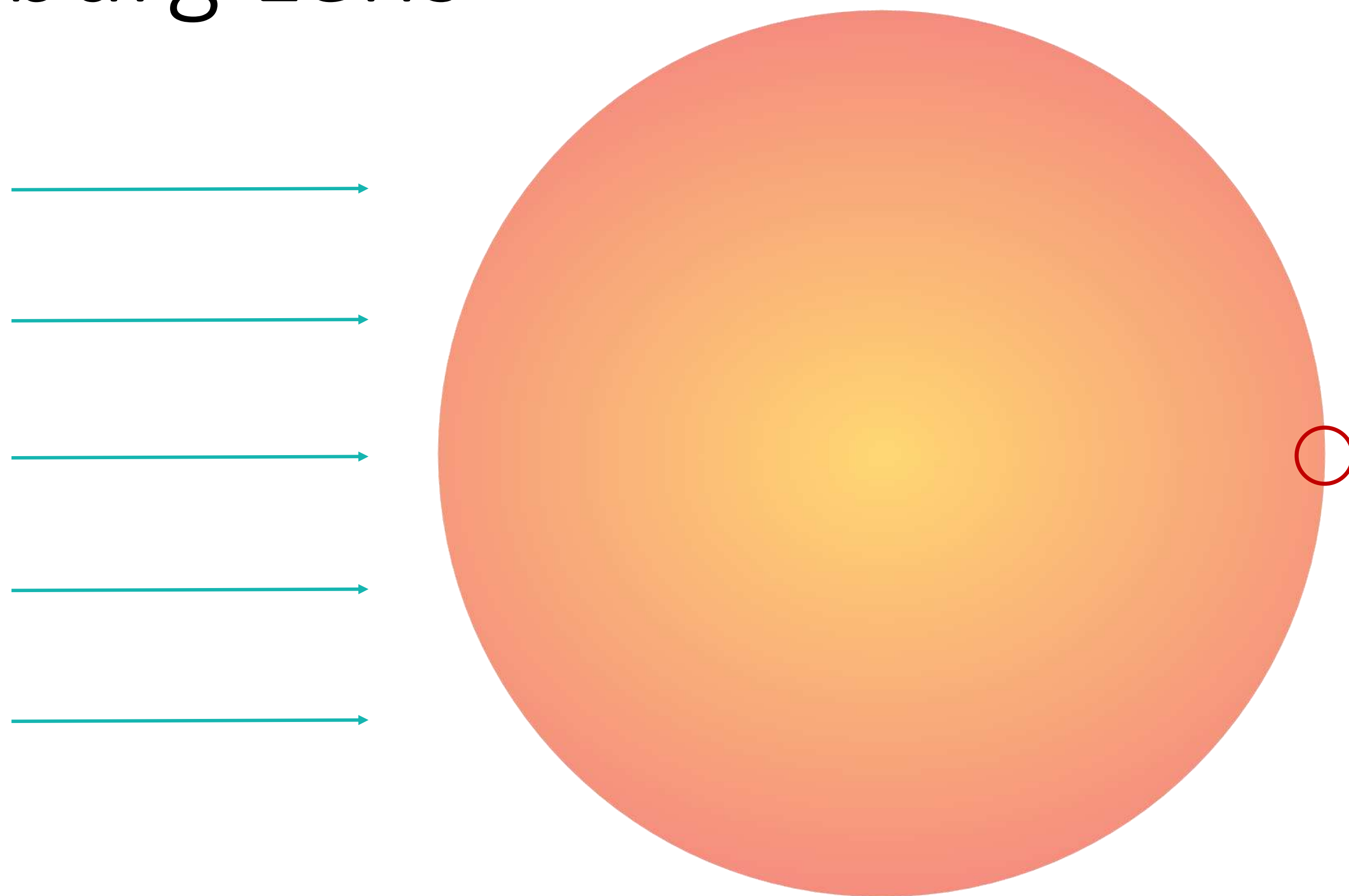
Luneburg Lens



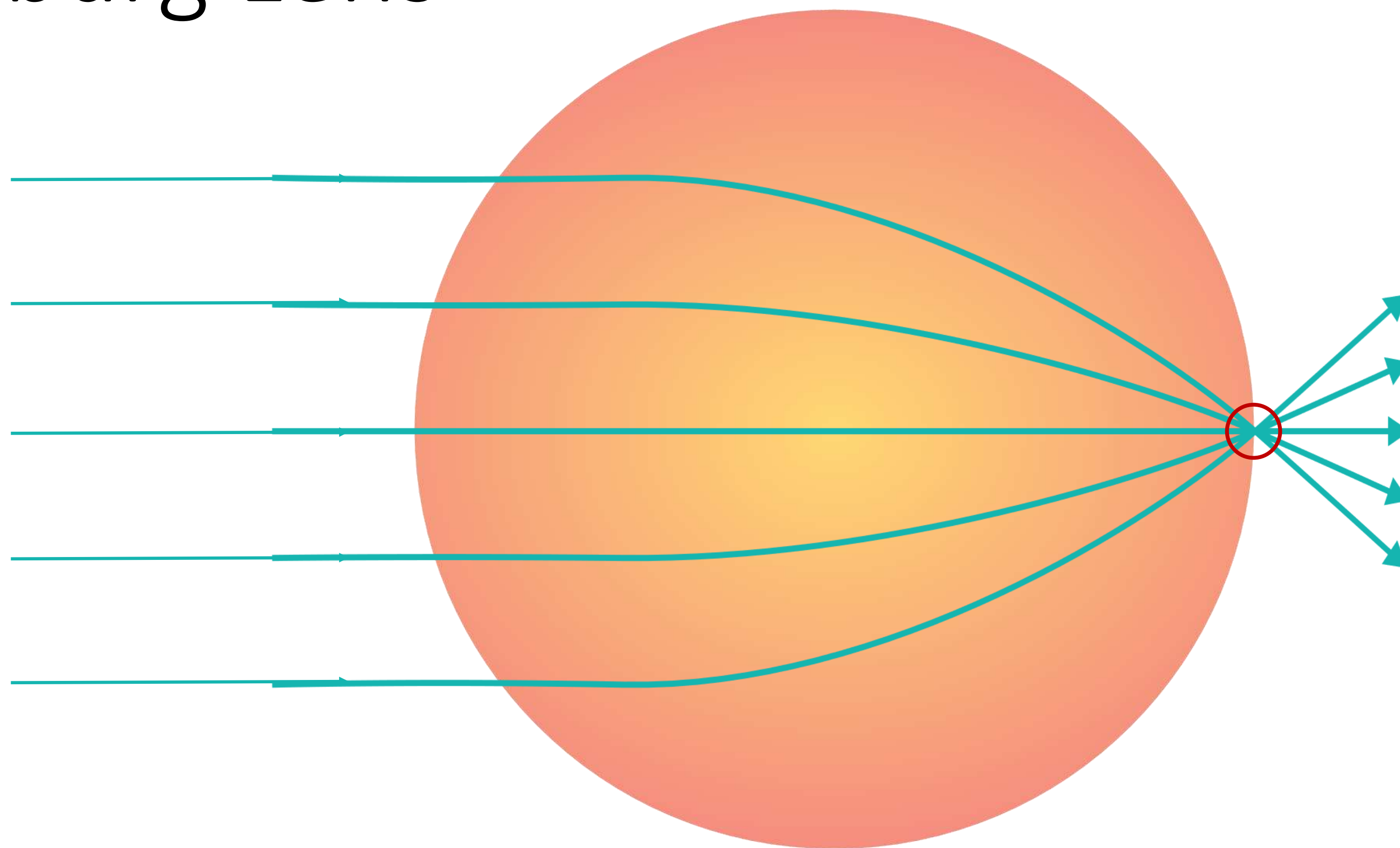
Luneburg Lens



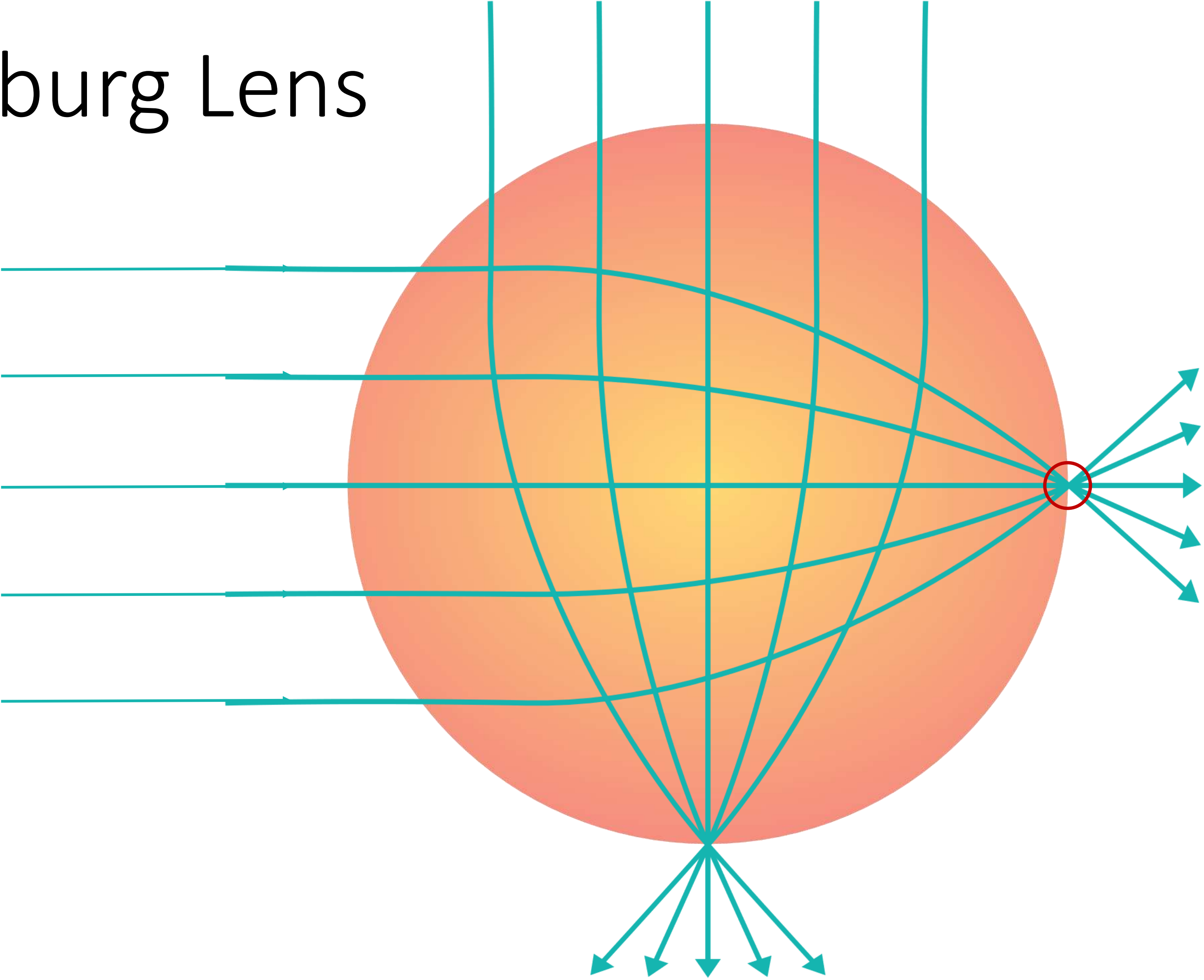
Luneburg Lens



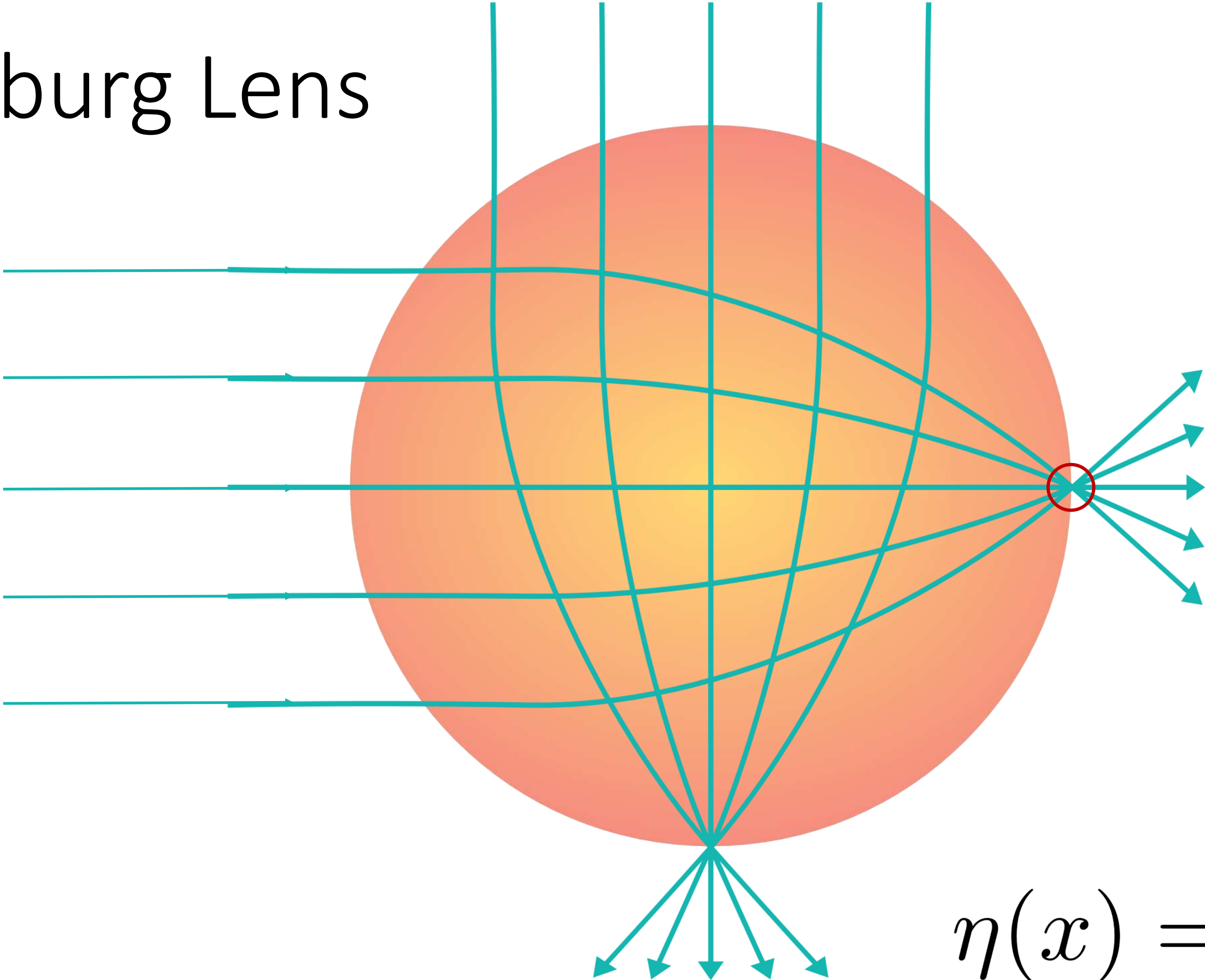
Luneburg Lens



Luneburg Lens



Luneburg Lens



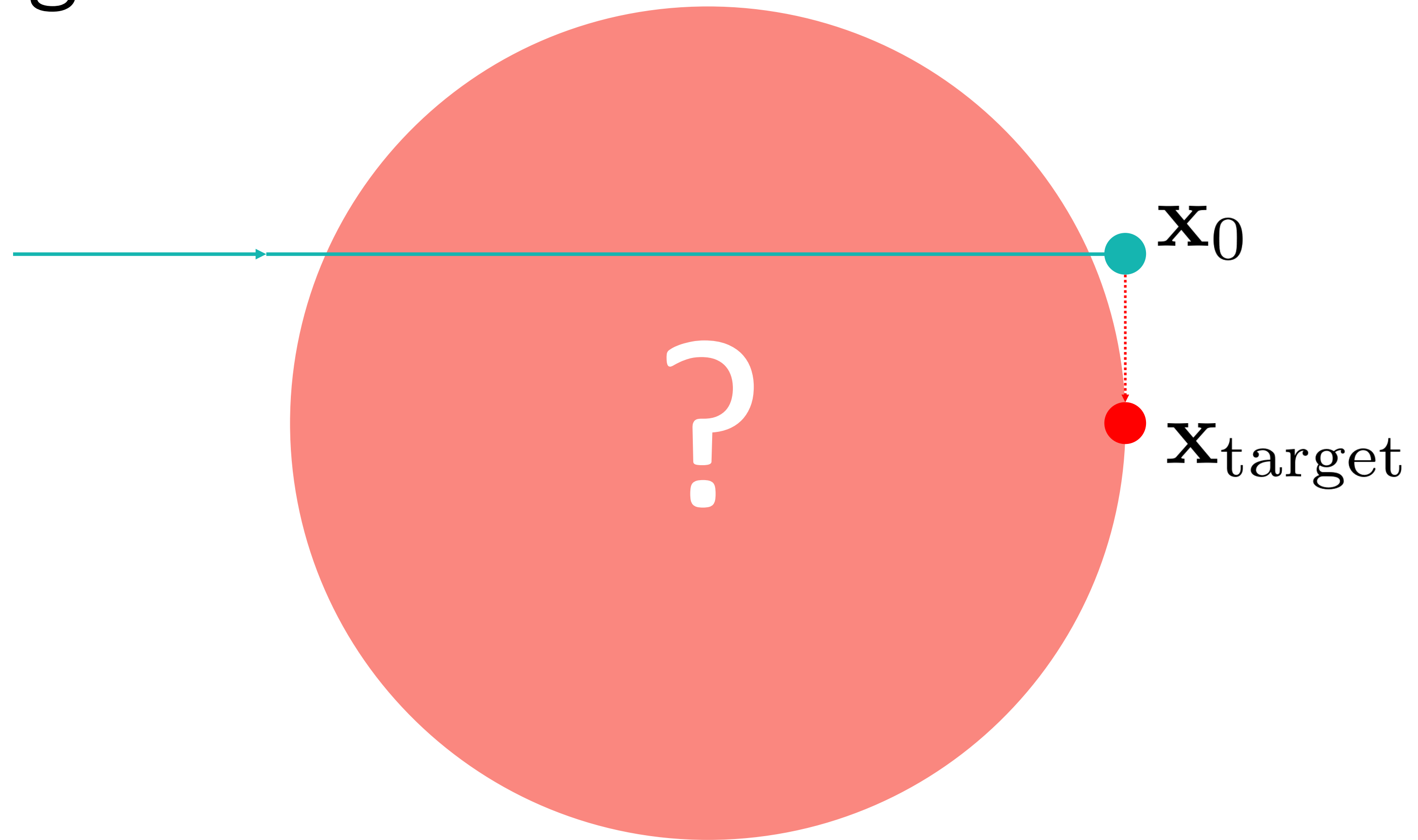
$$\eta(x) = \sqrt{2 - \|x\|^2}$$

[Luneburg, R. K. 1944]

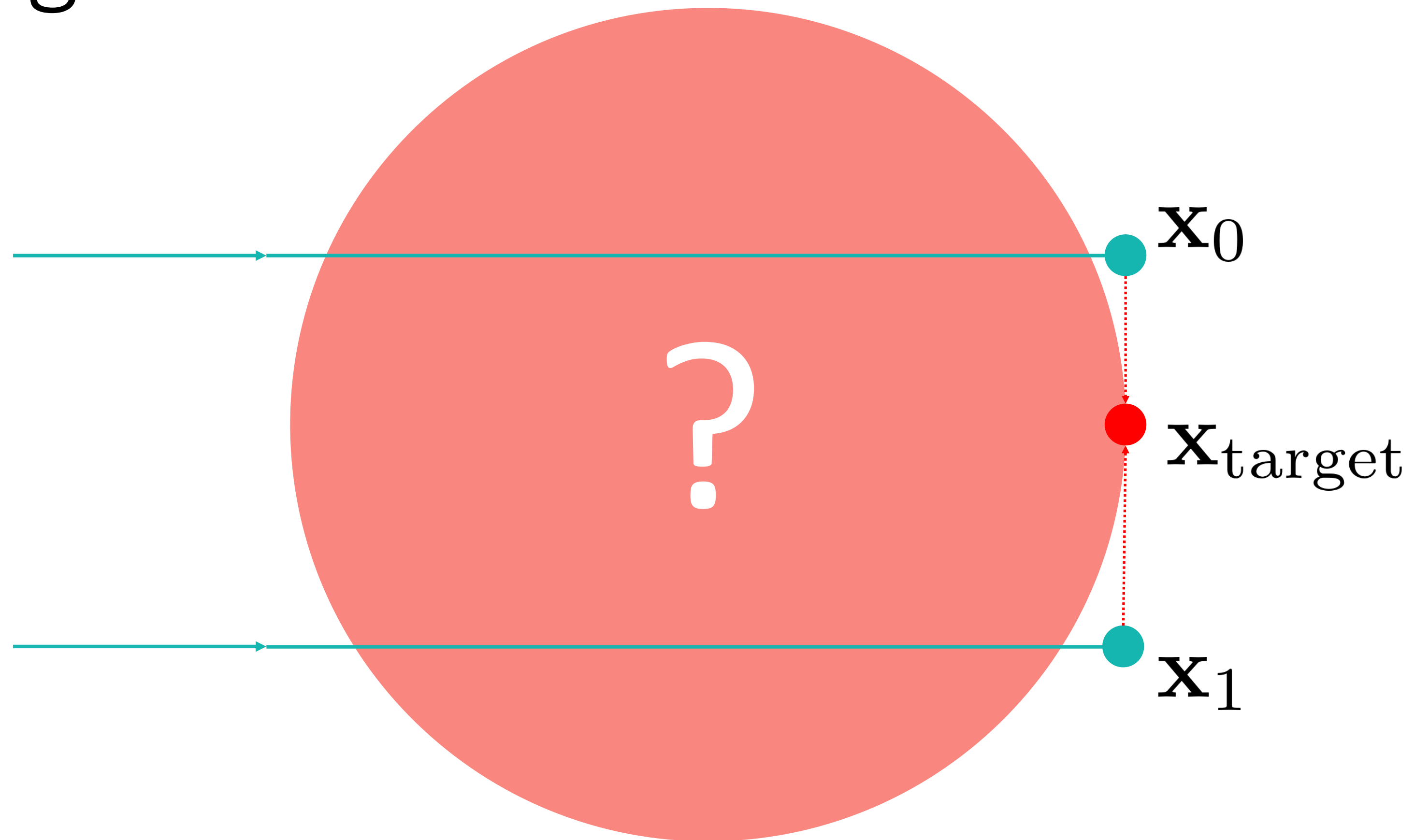
Luneburg Lens



Luneburg Lens

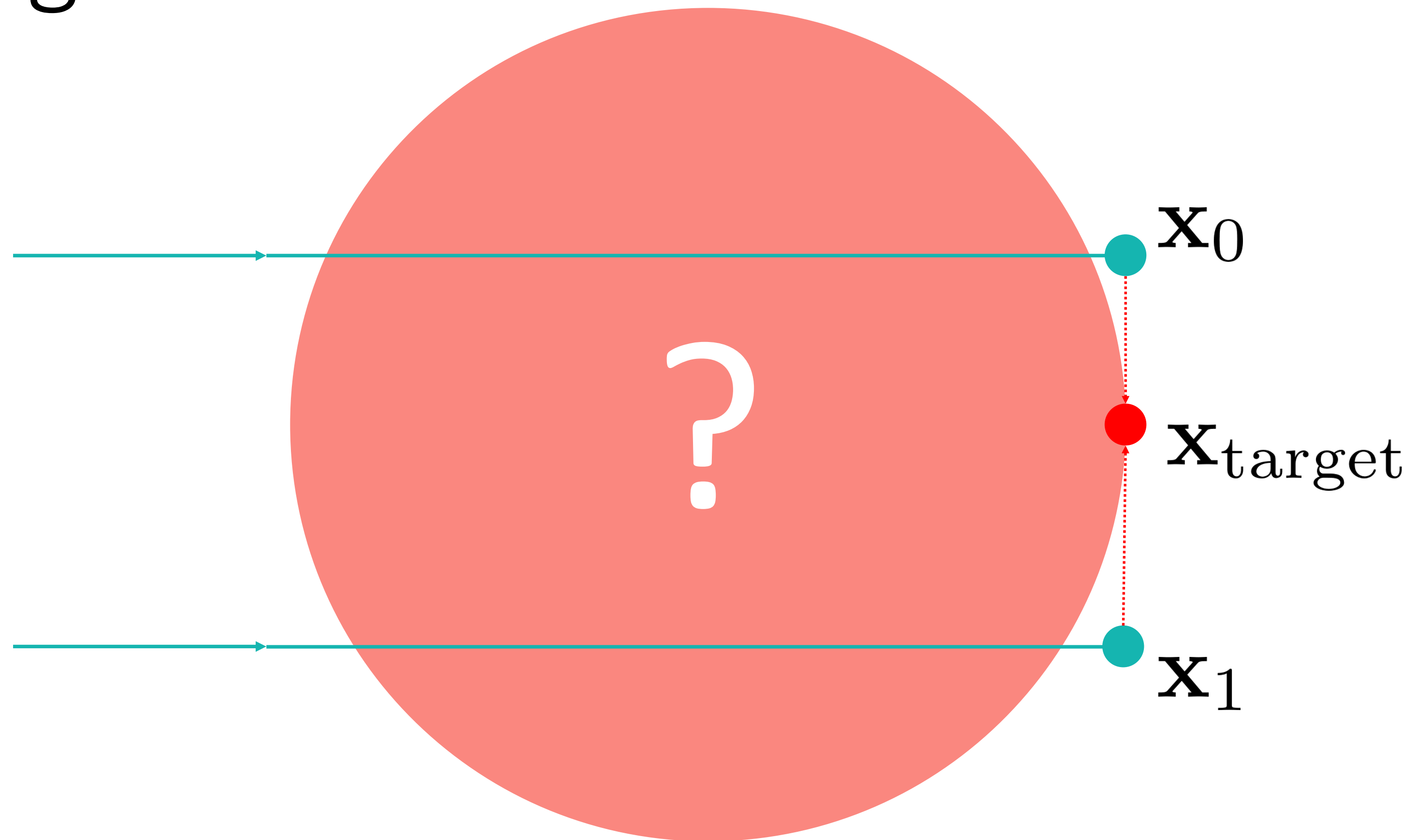


Luneburg Lens



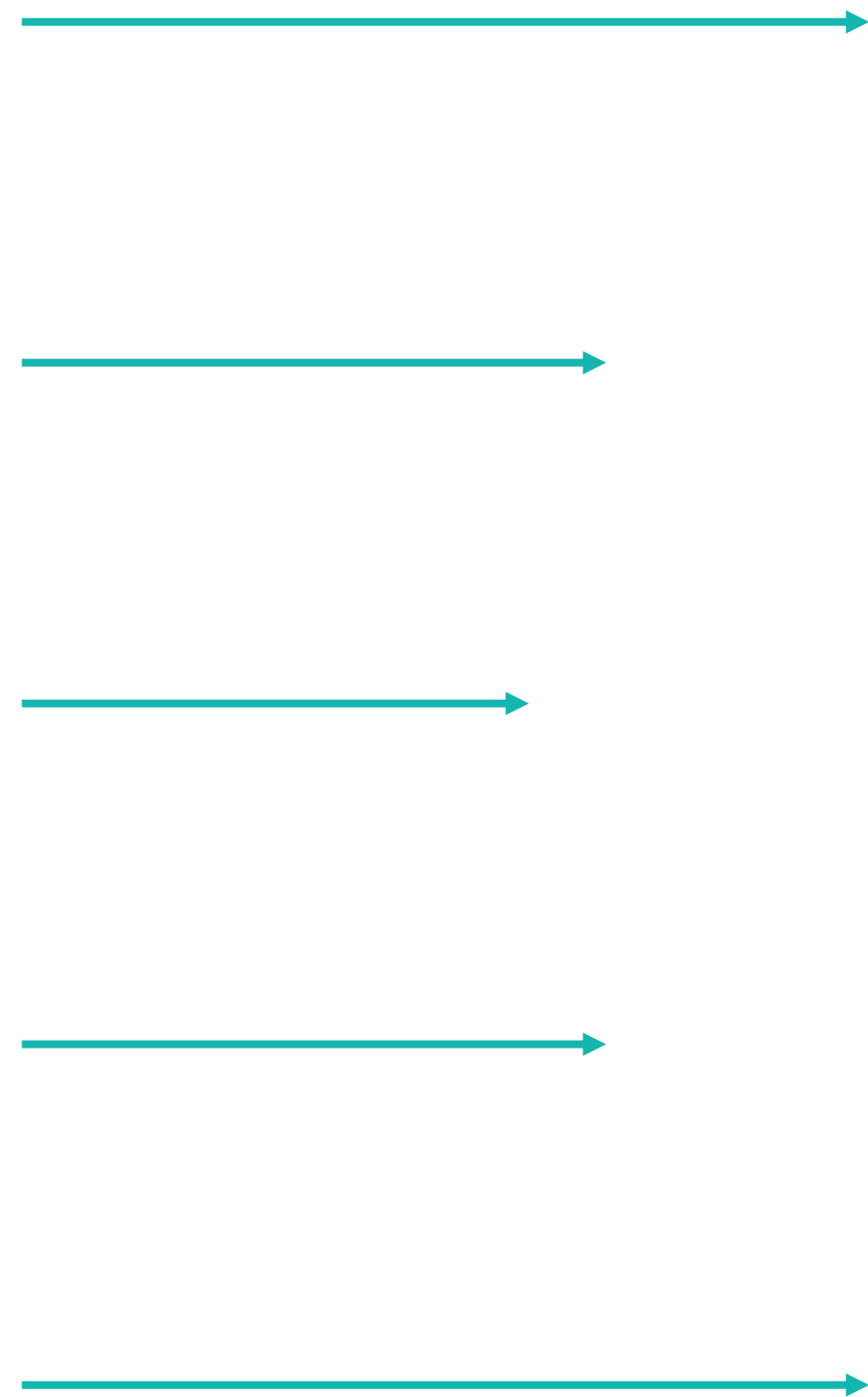
$$\min_{\eta} \sum_{i \in \text{rays}} \|\mathbf{x}_{\text{target}} - \mathbf{x}_i\|^2$$

Luneburg Lens

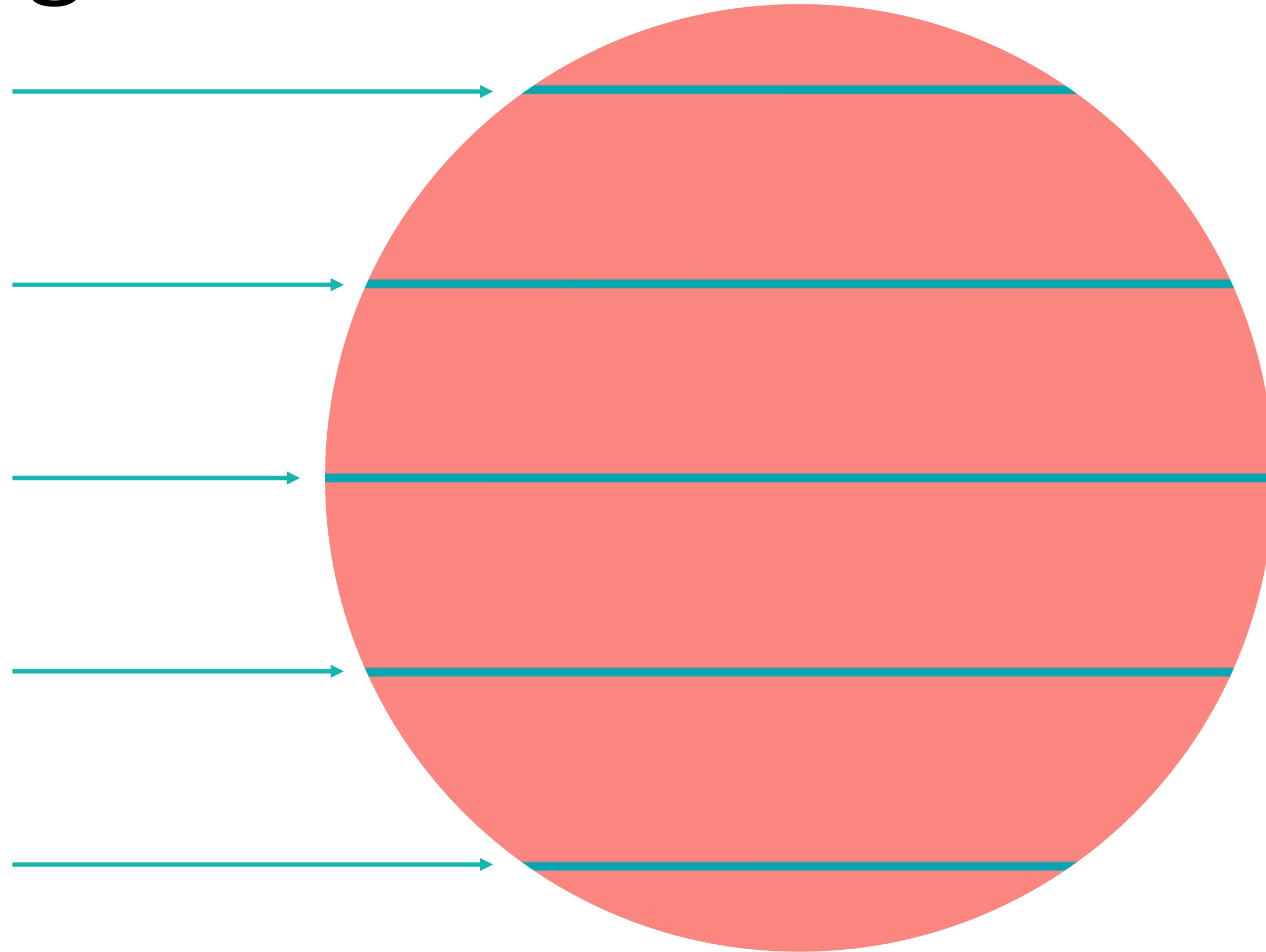


$$\min_{\eta} \sum_{i \in \text{rays}} \|\mathbf{x}_{\text{target}} - \mathbf{x}_i\|^2$$

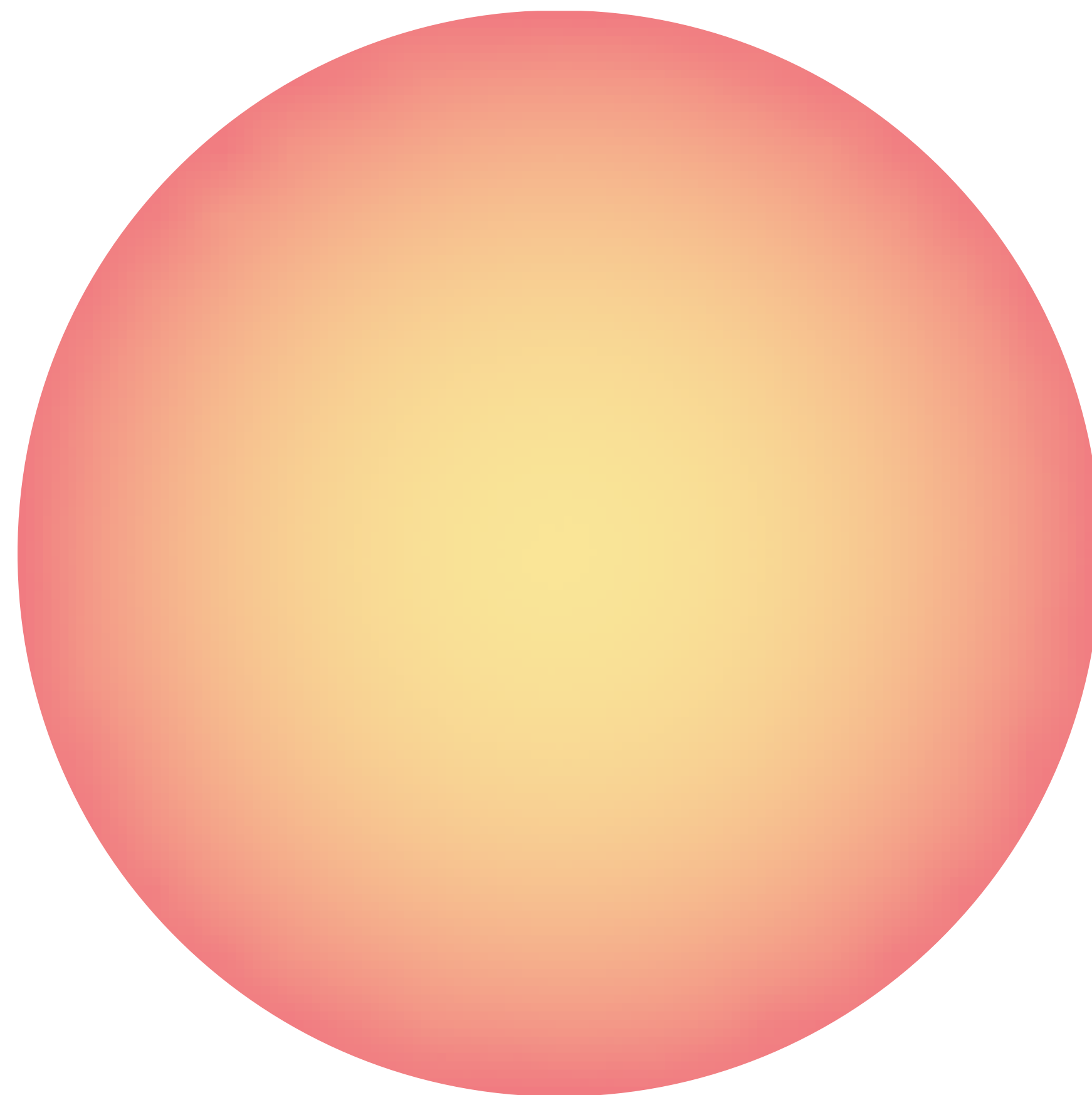
Luneburg Lens



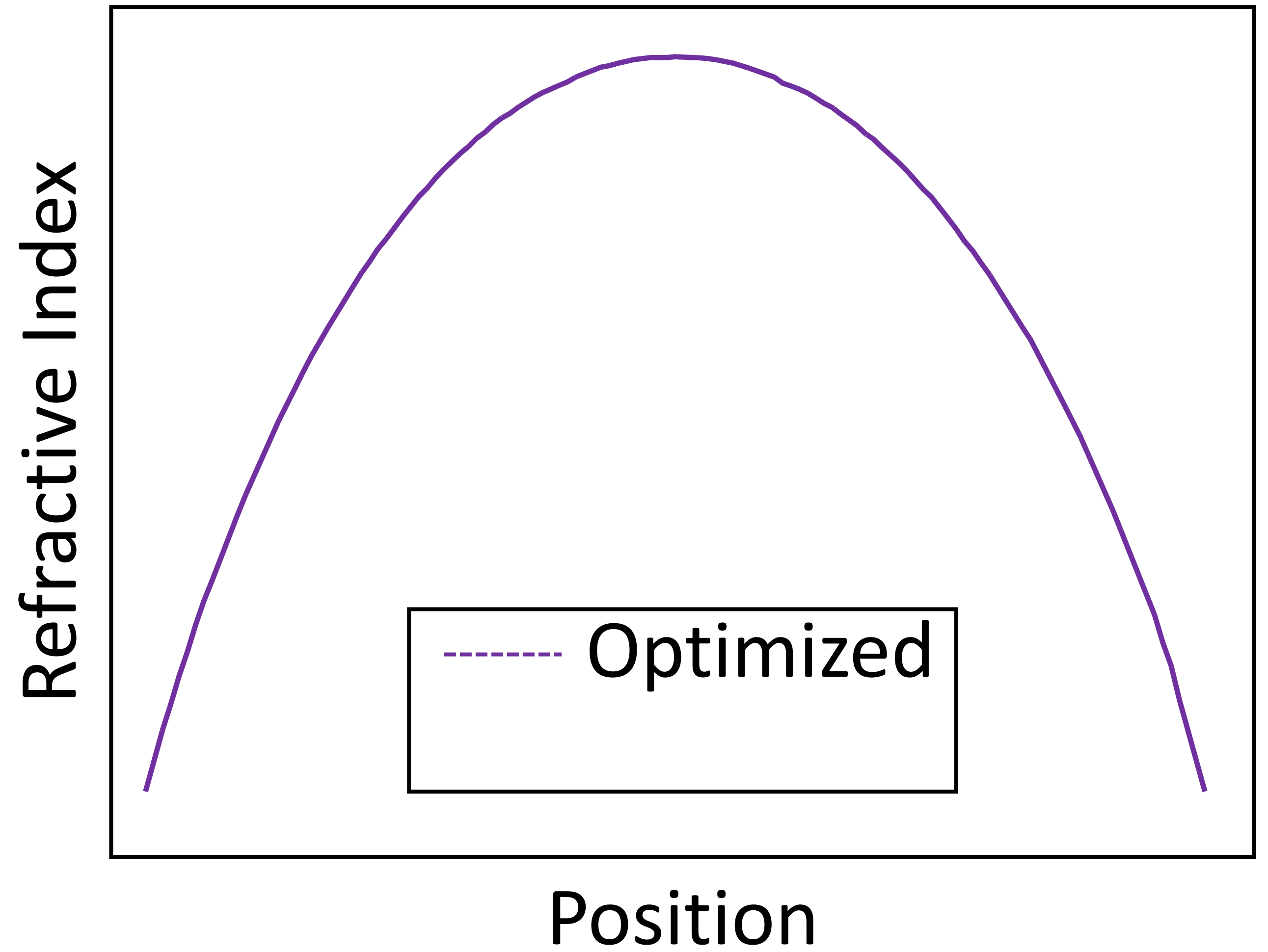
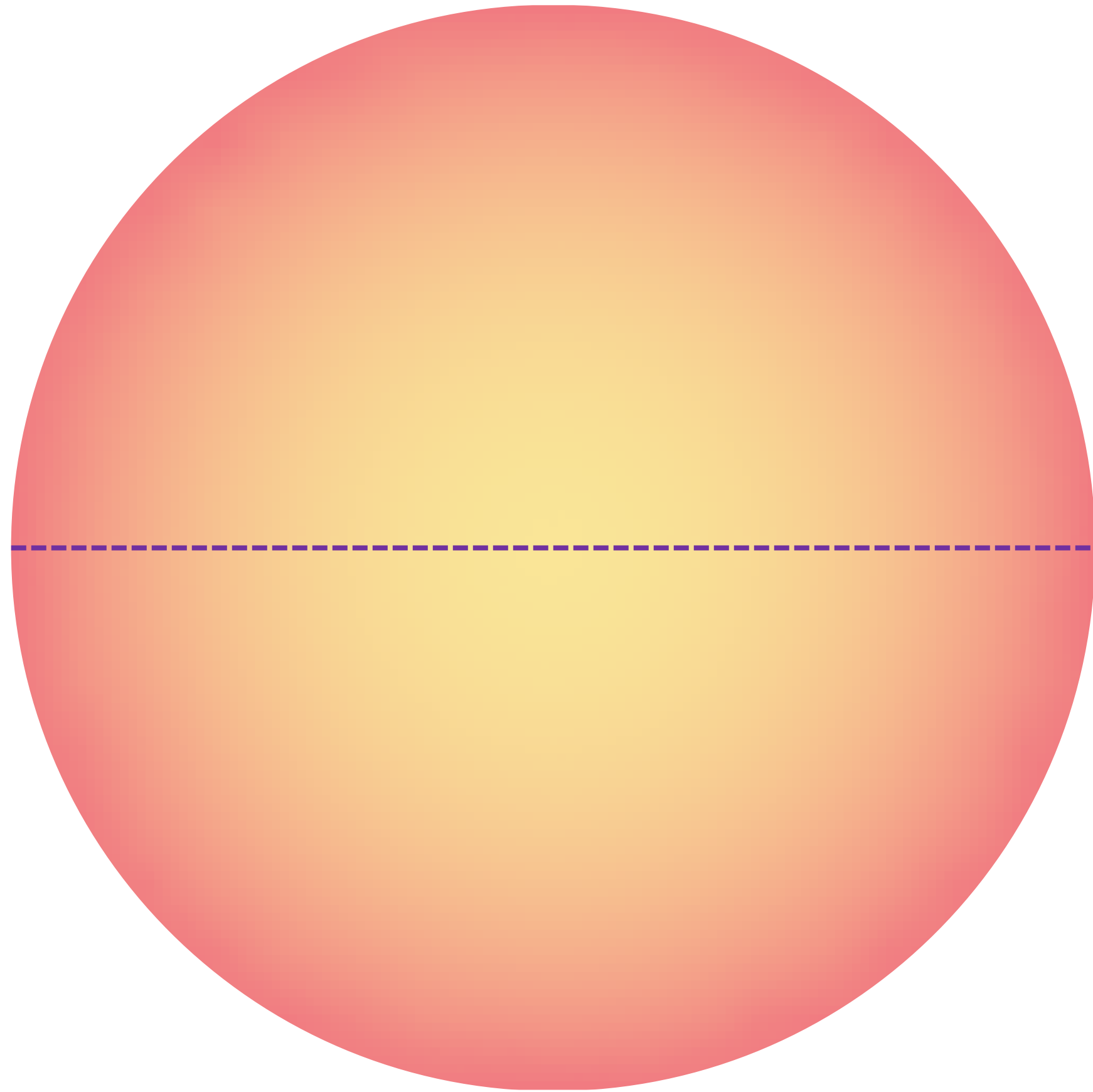
Luneburg Lens



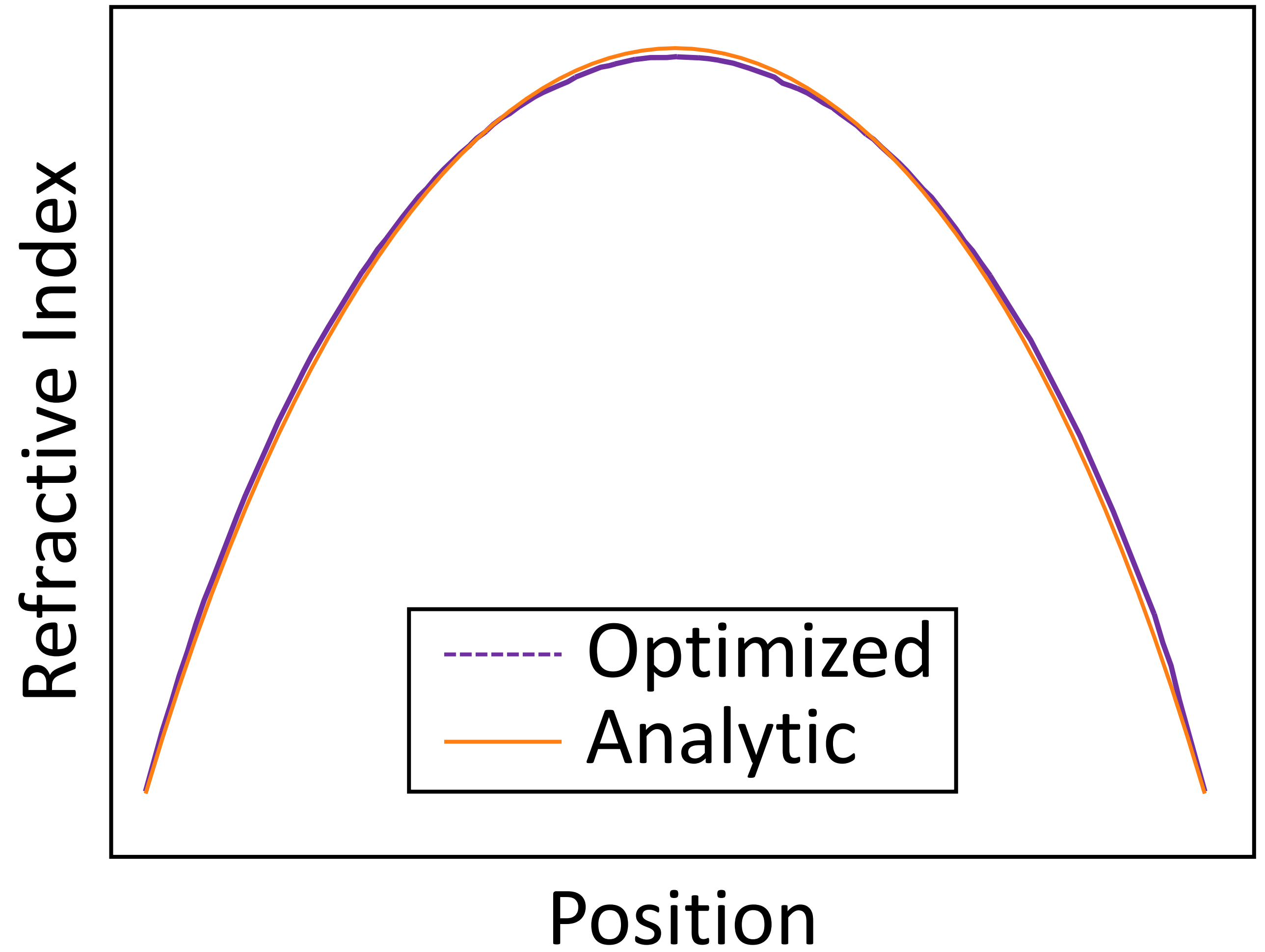
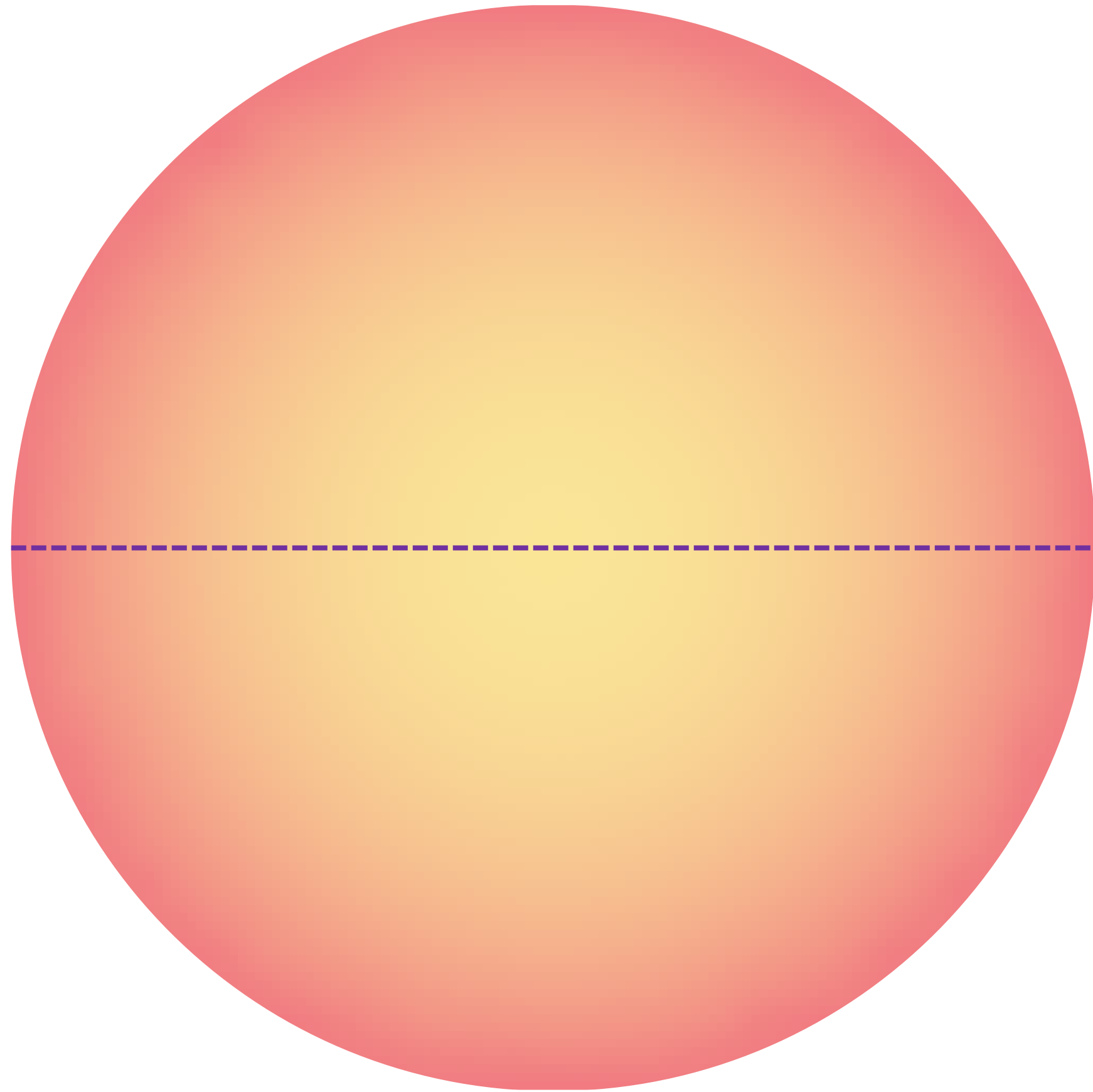
Luneburg Lens



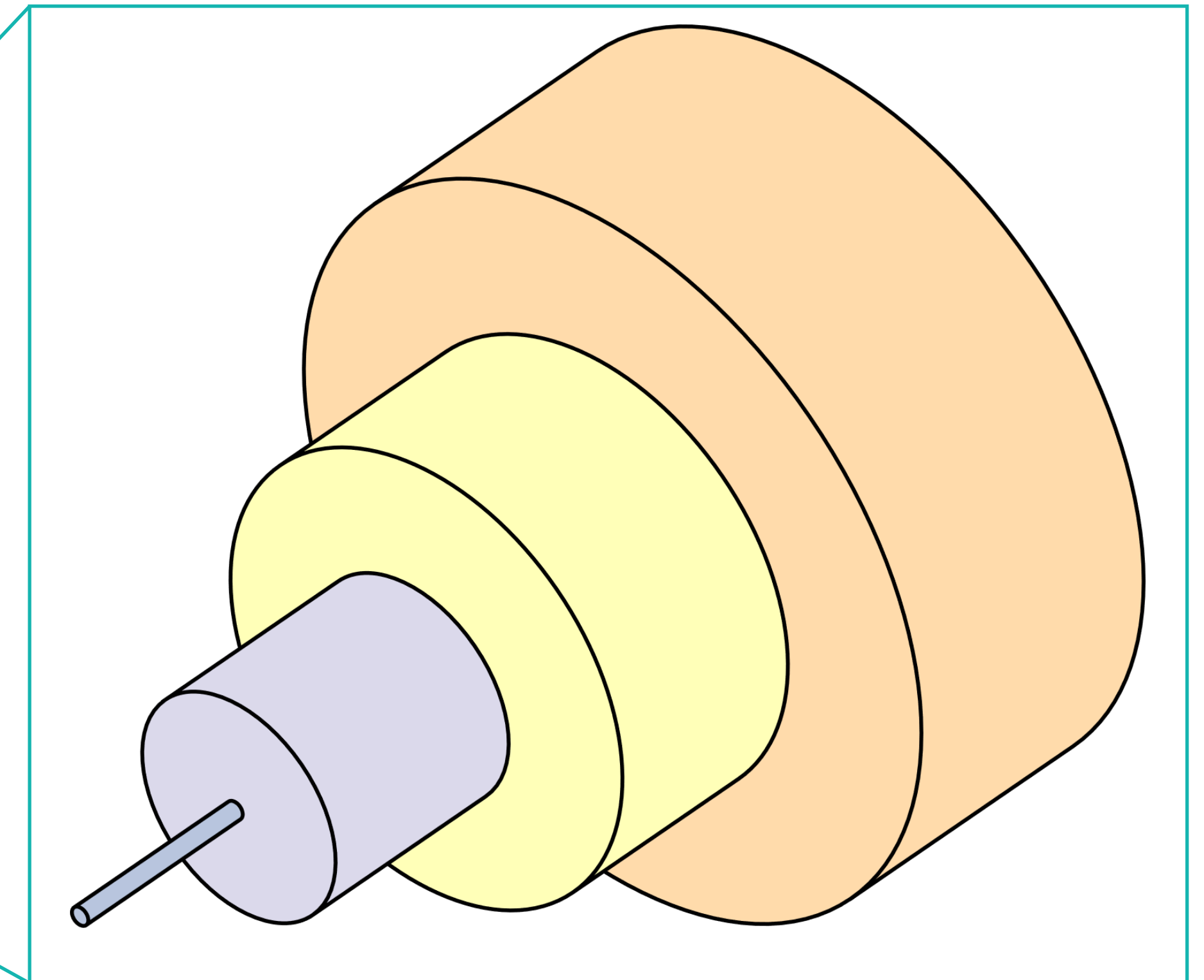
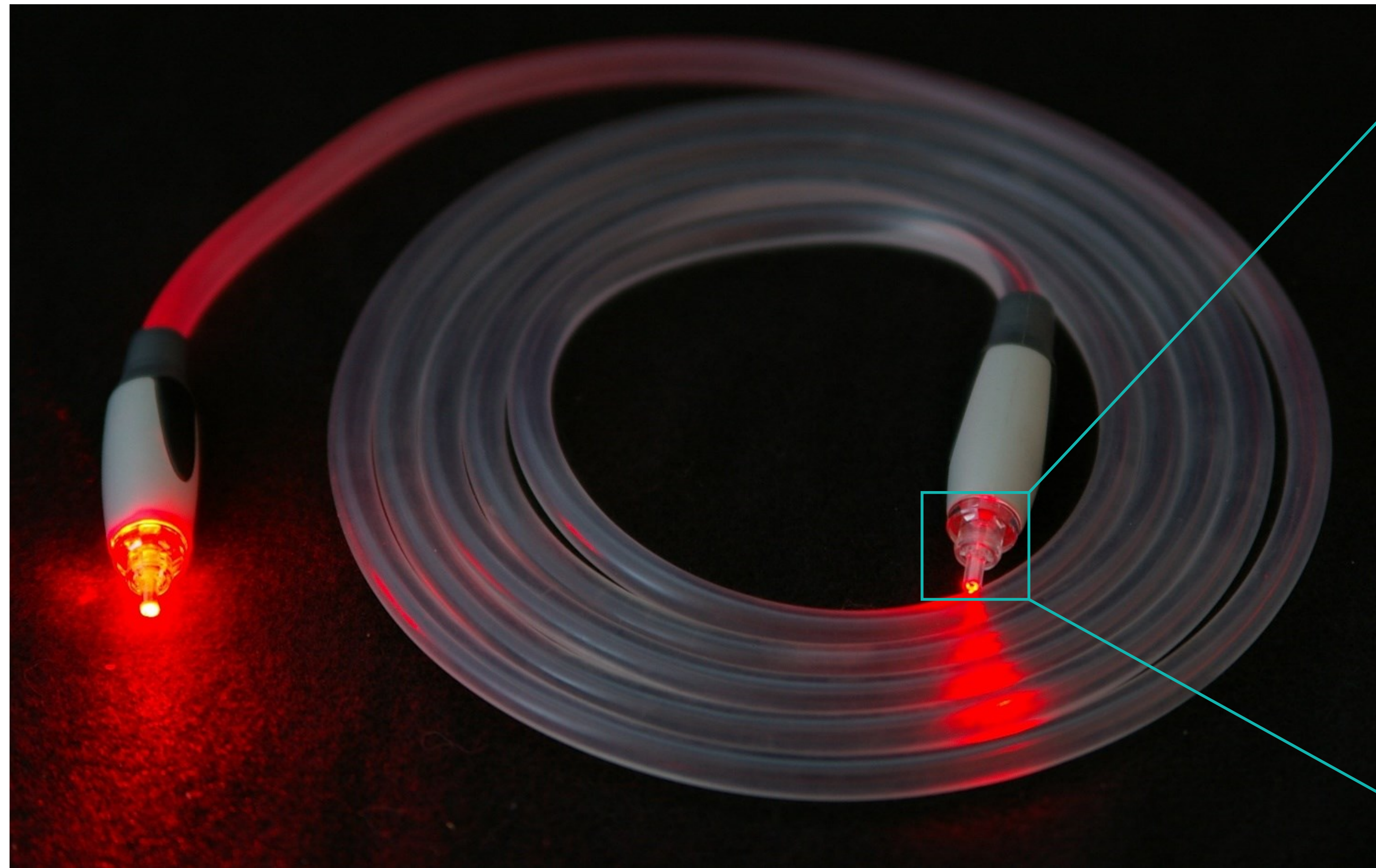
Luneburg Lens



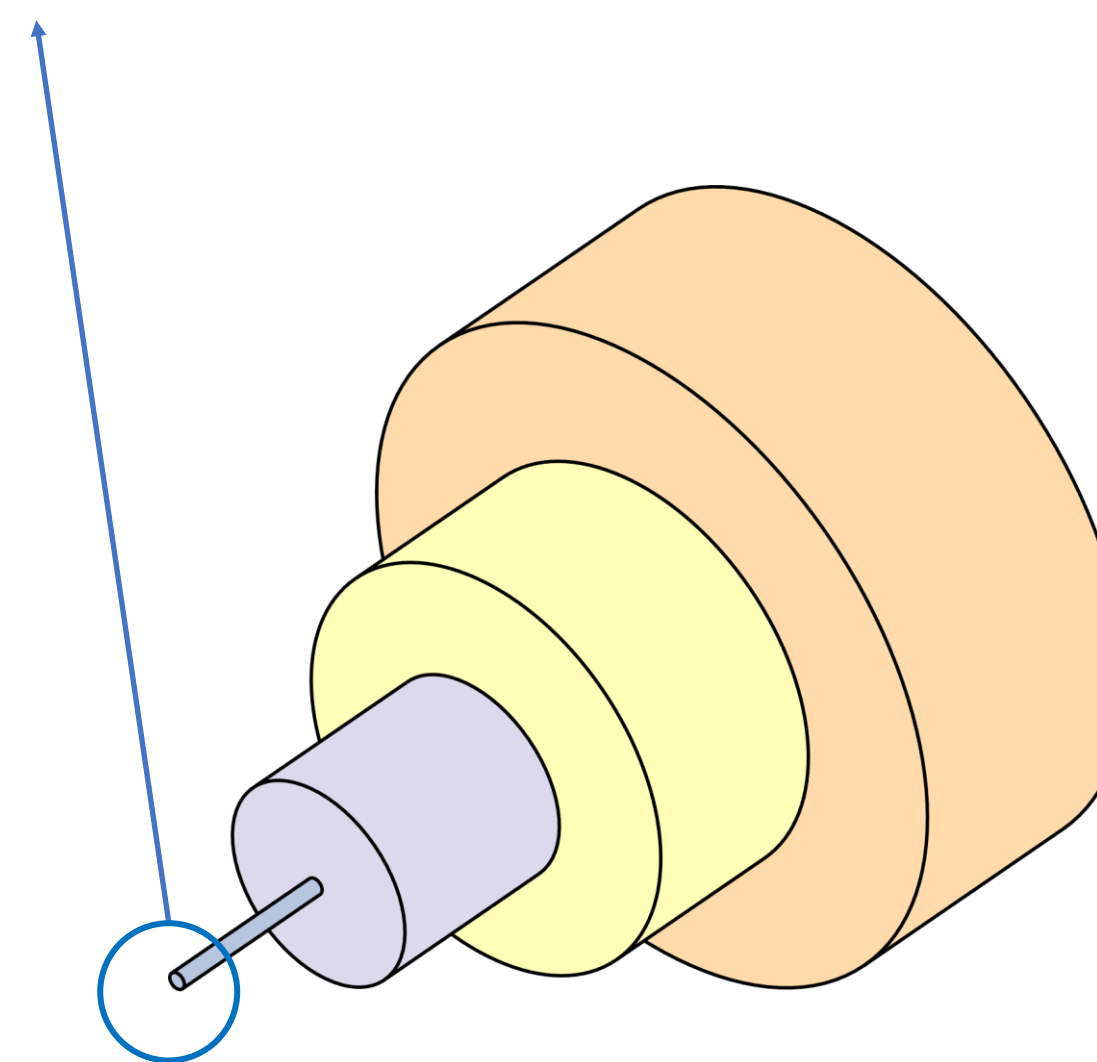
Luneburg Lens



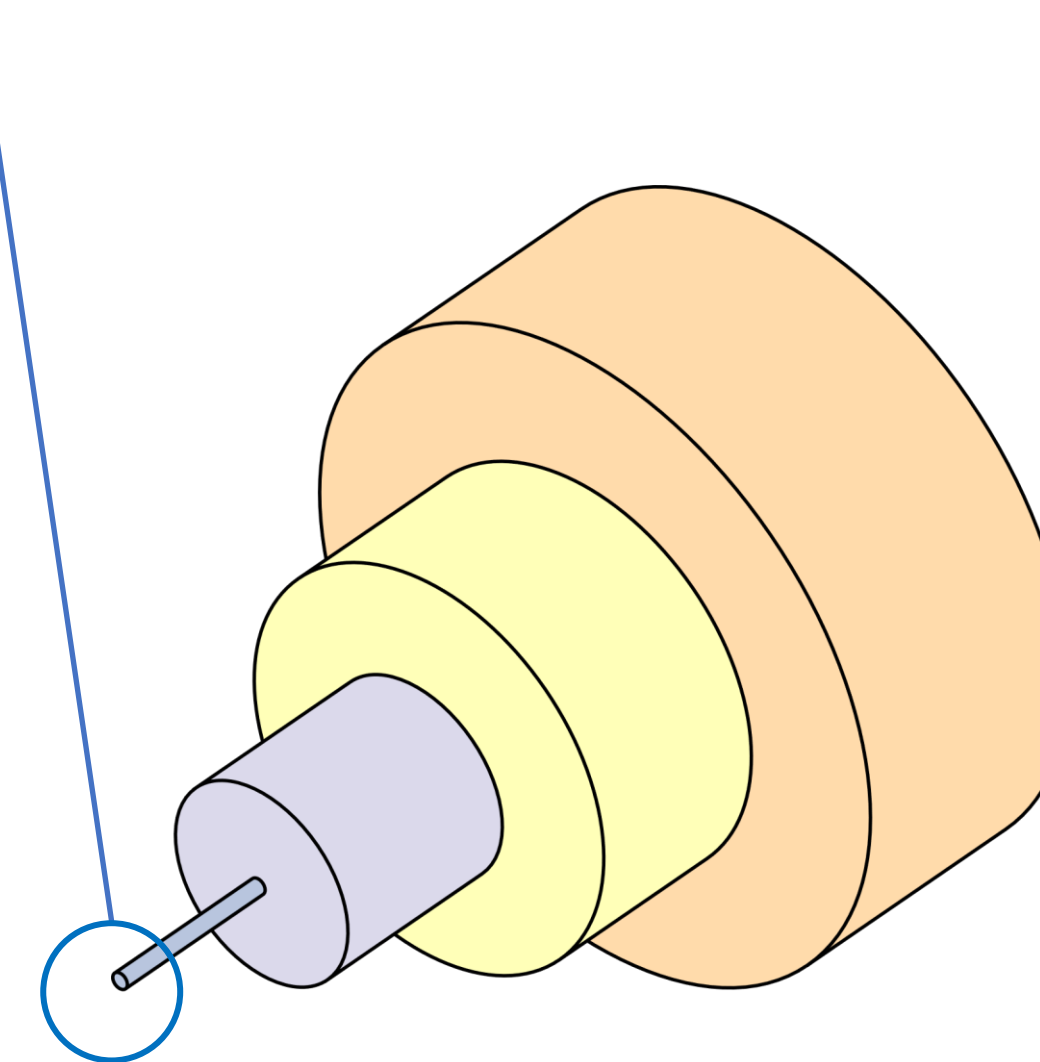
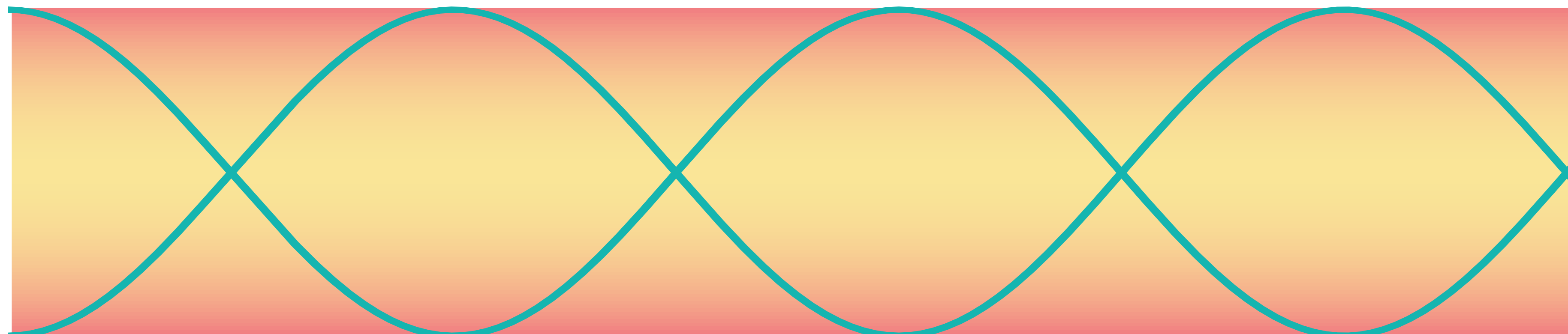
GRIN Fiber



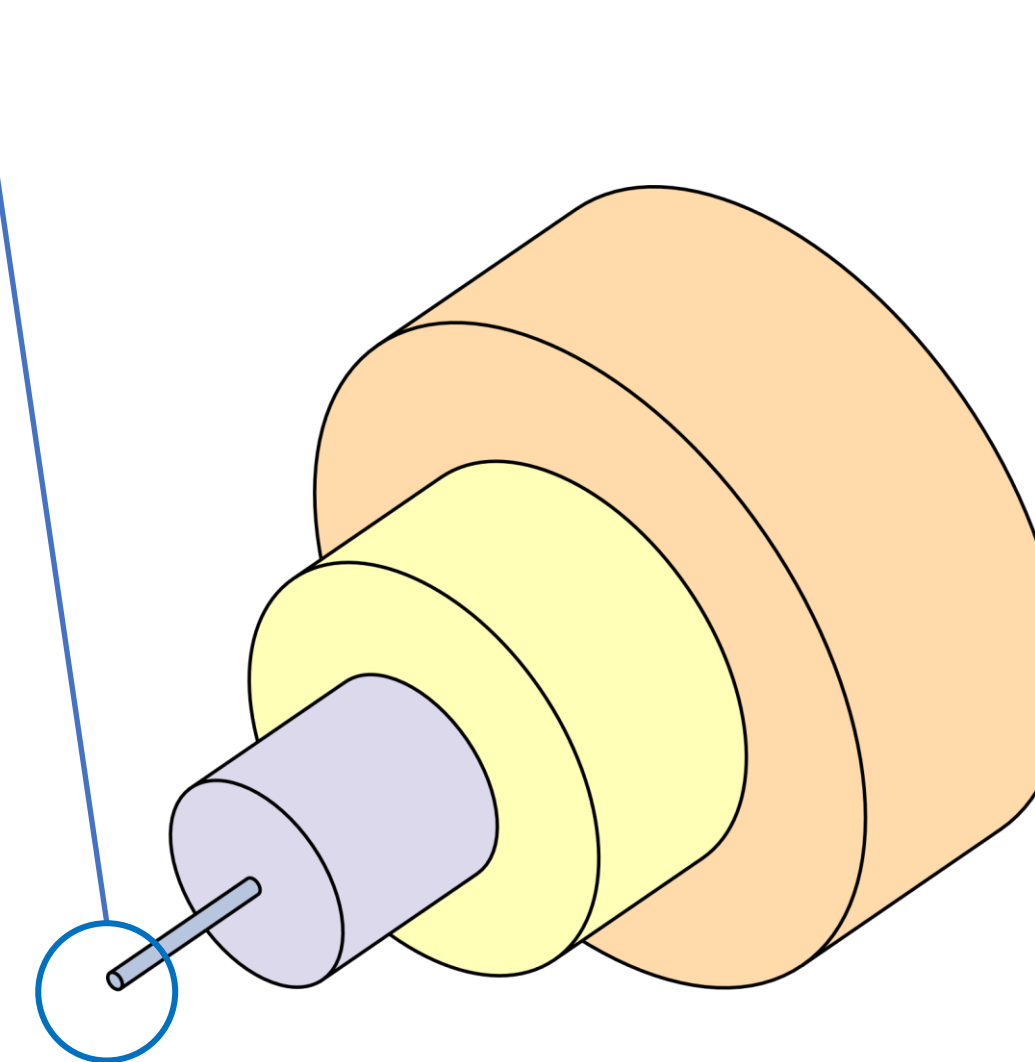
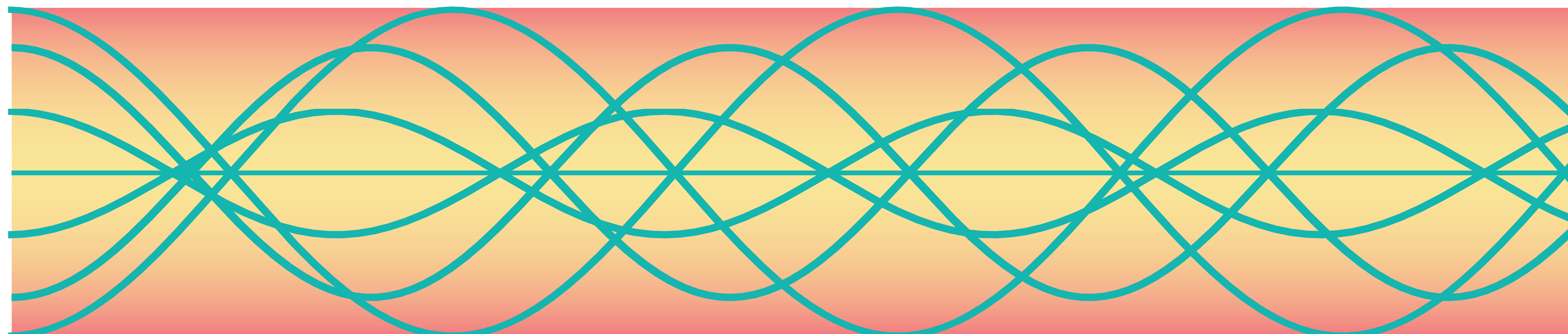
GRIN Fiber



GRIN Fiber

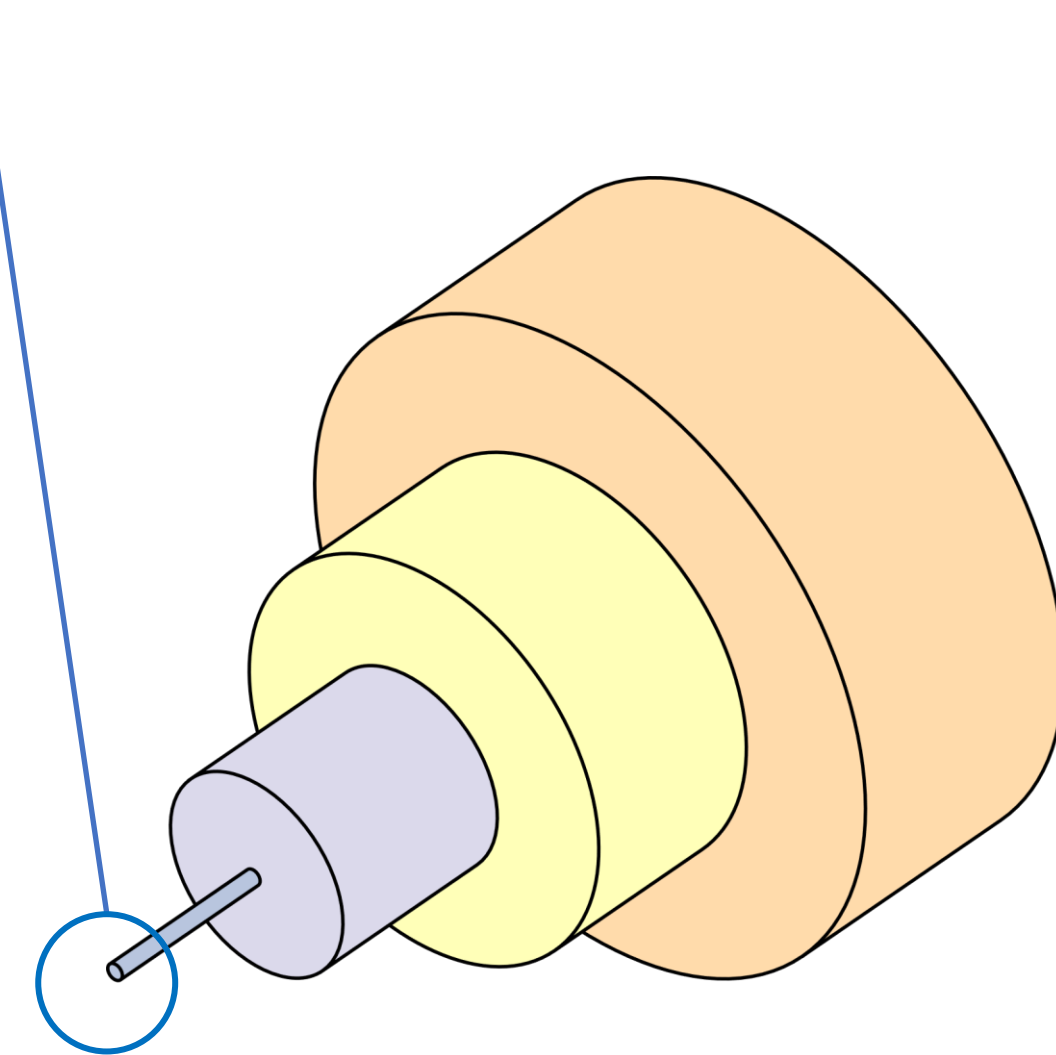
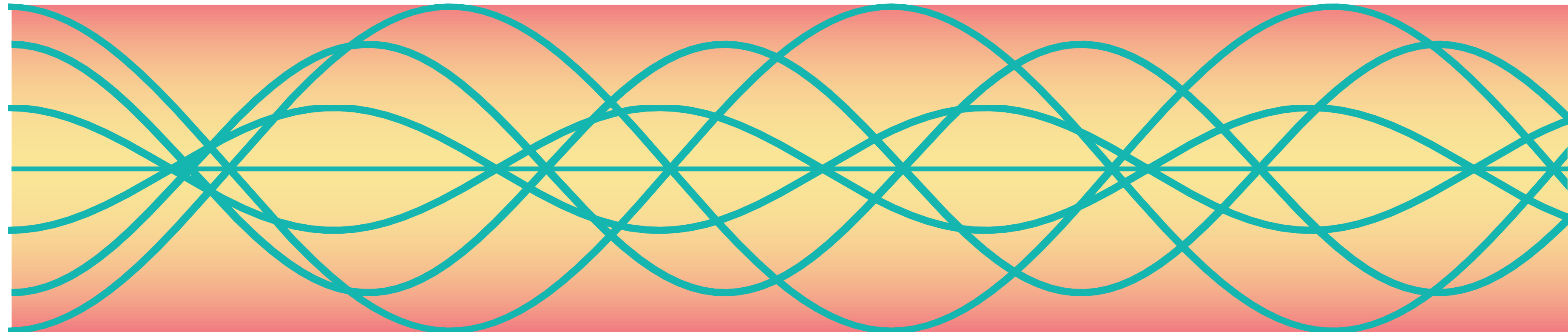


GRIN Fiber



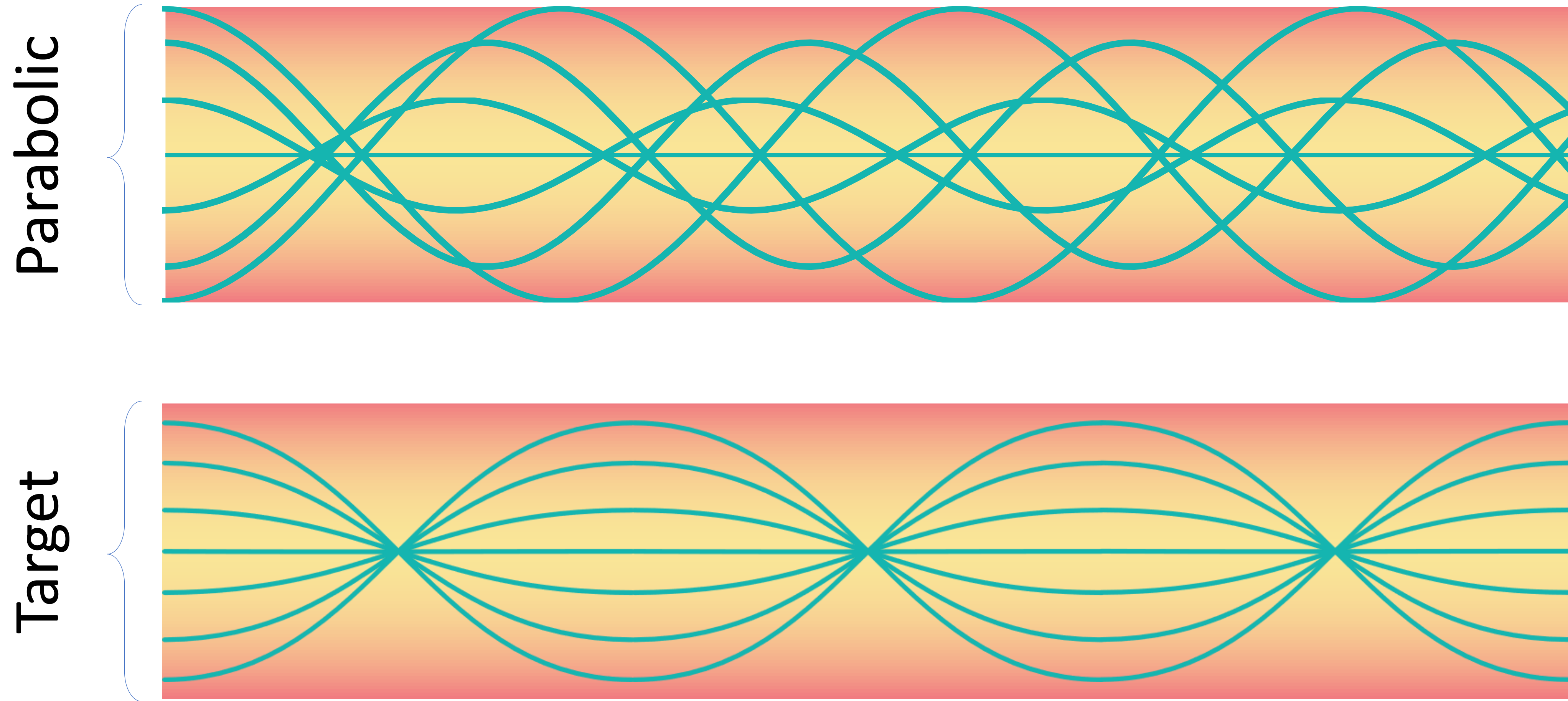
GRIN Fiber

Modal dispersion

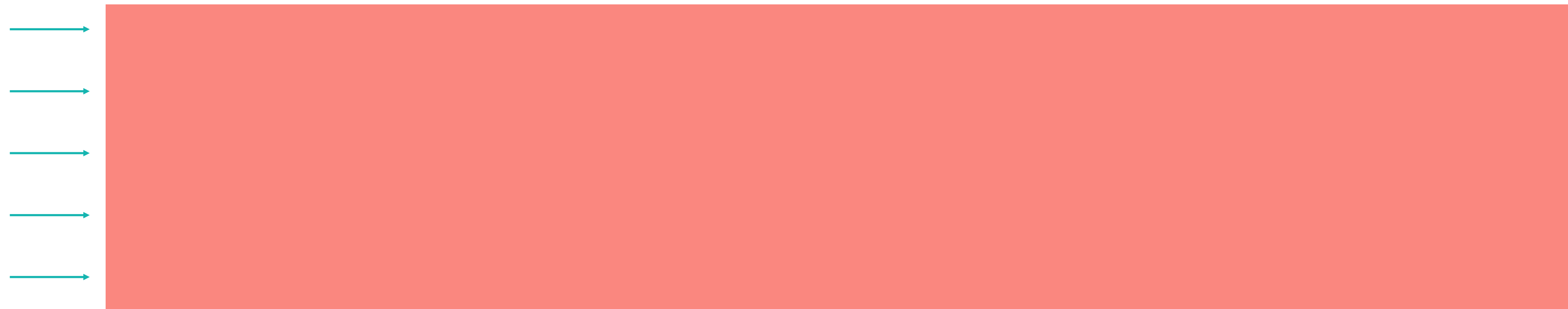
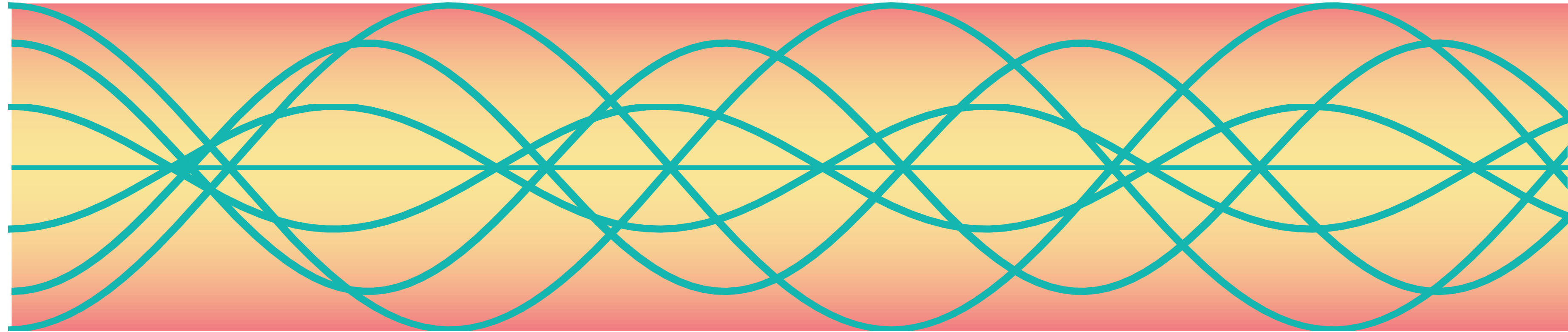


GRIN Fiber

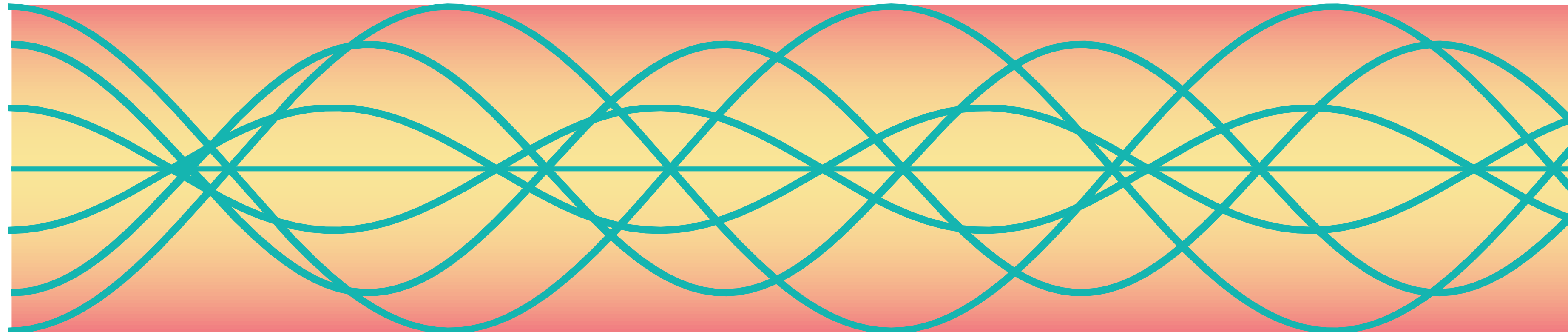
Modal dispersion



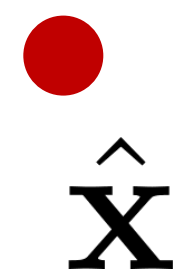
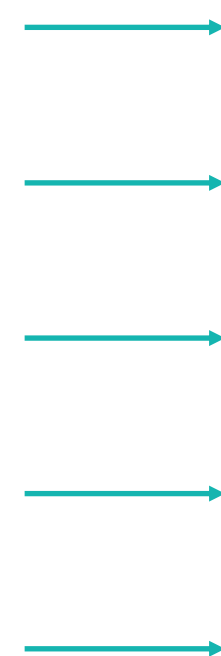
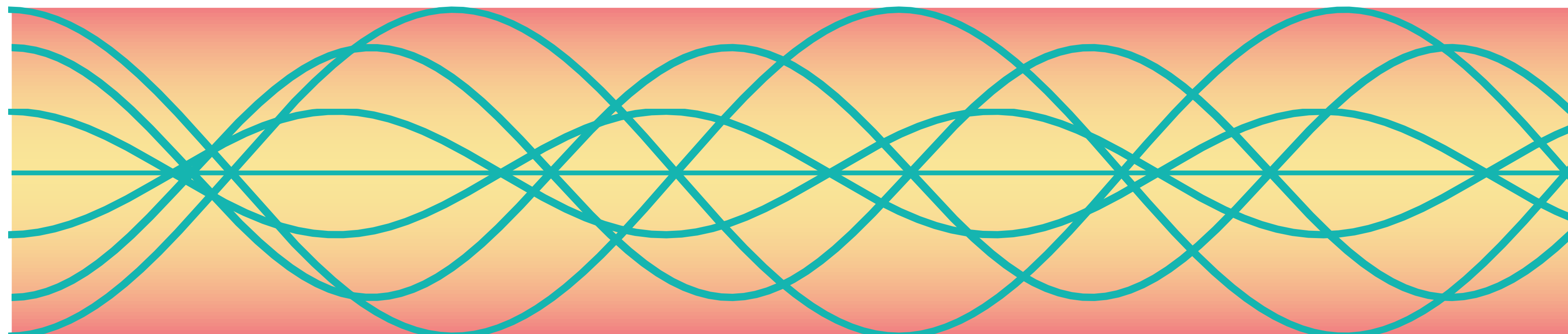
GRIN Fiber



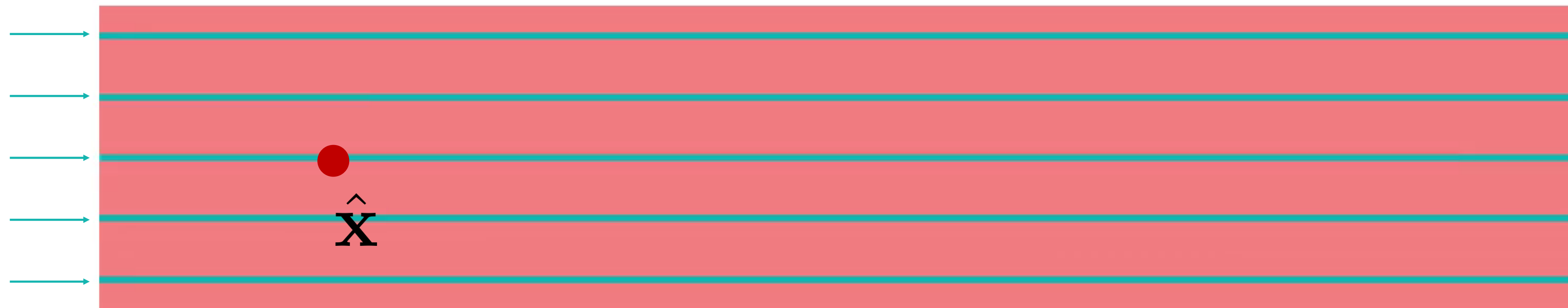
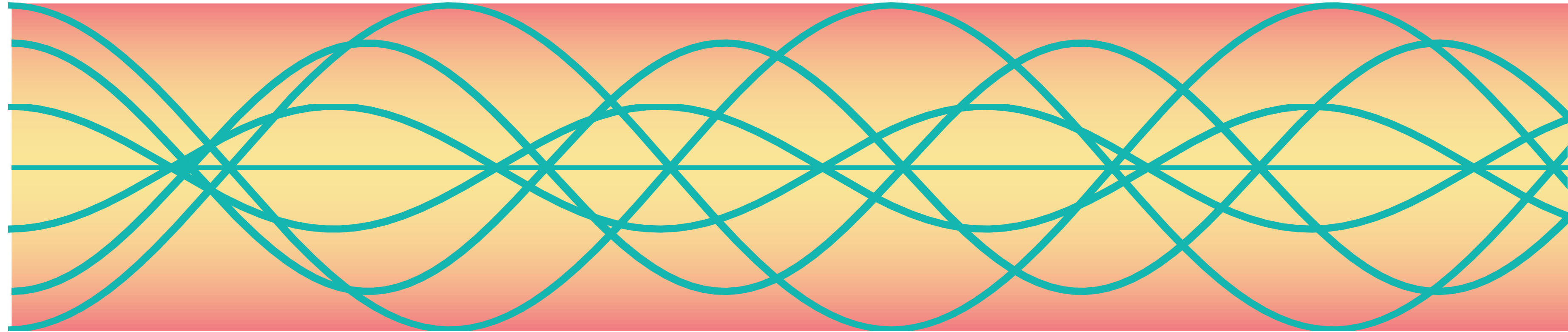
GRIN Fiber



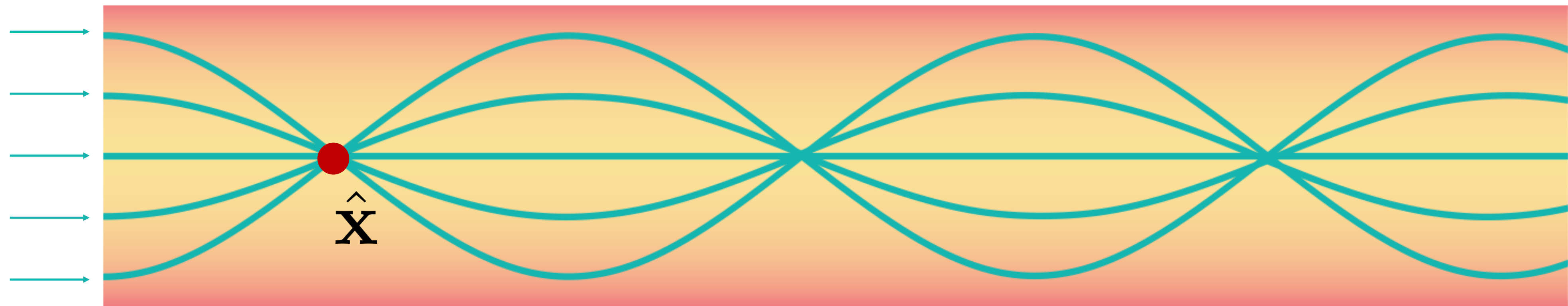
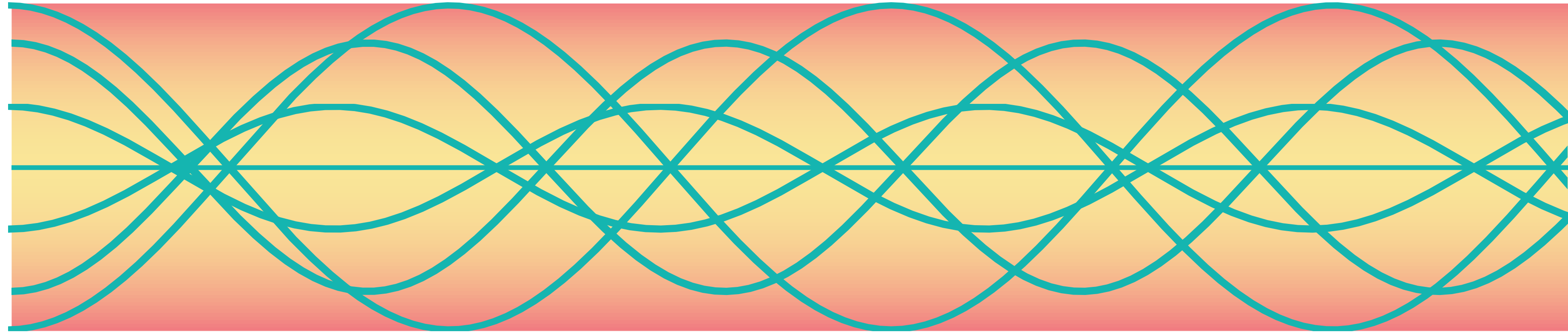
GRIN Fiber



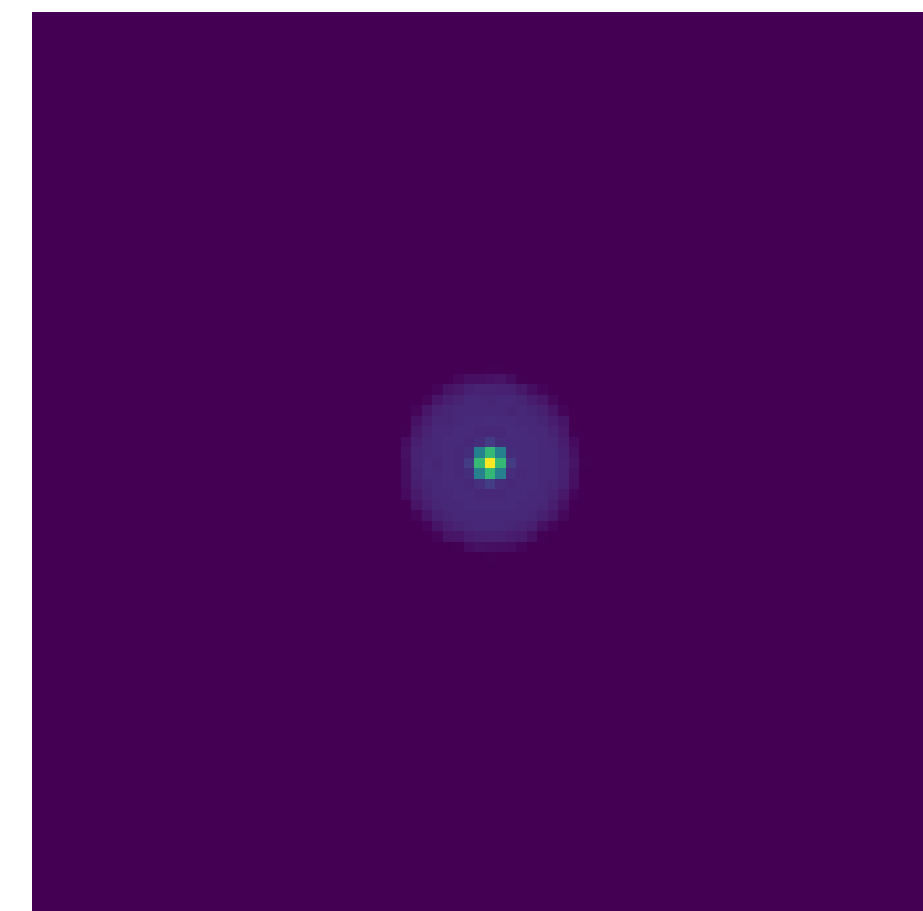
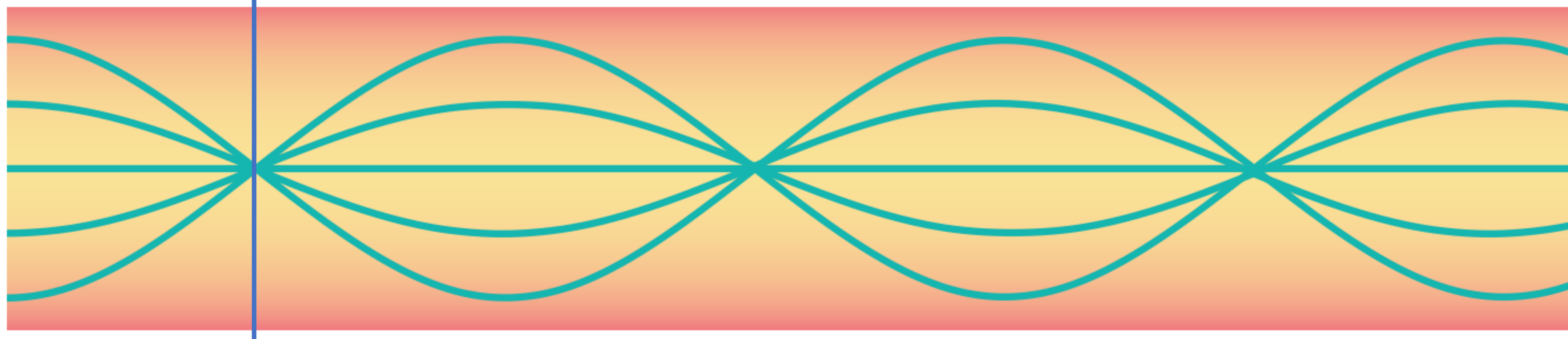
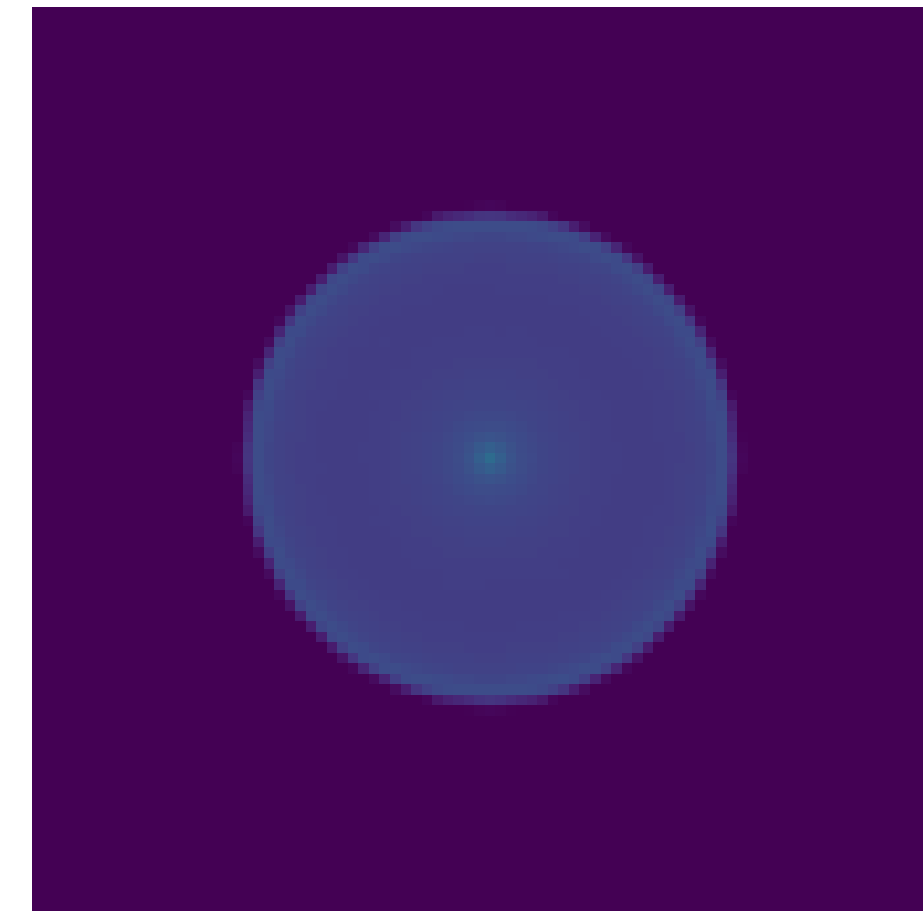
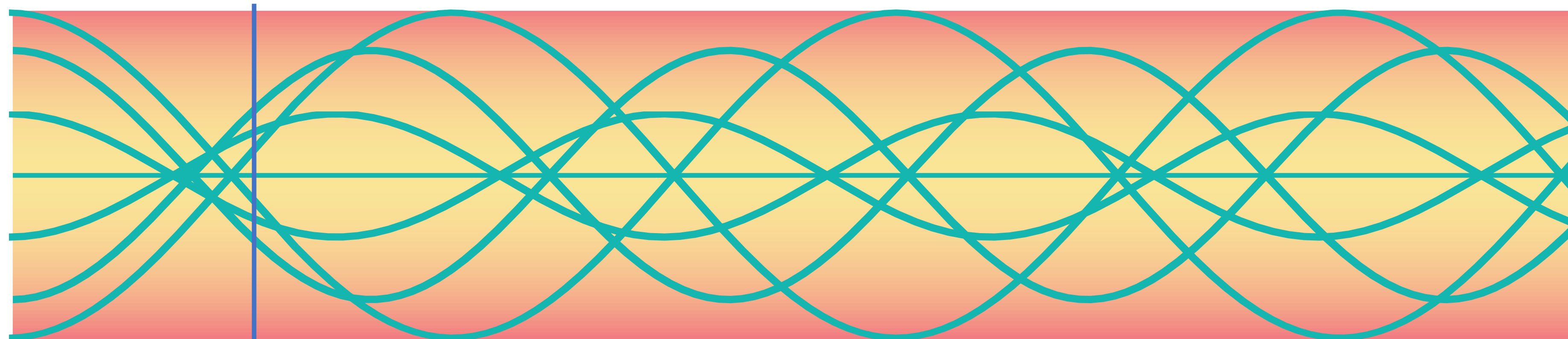
GRIN Fiber



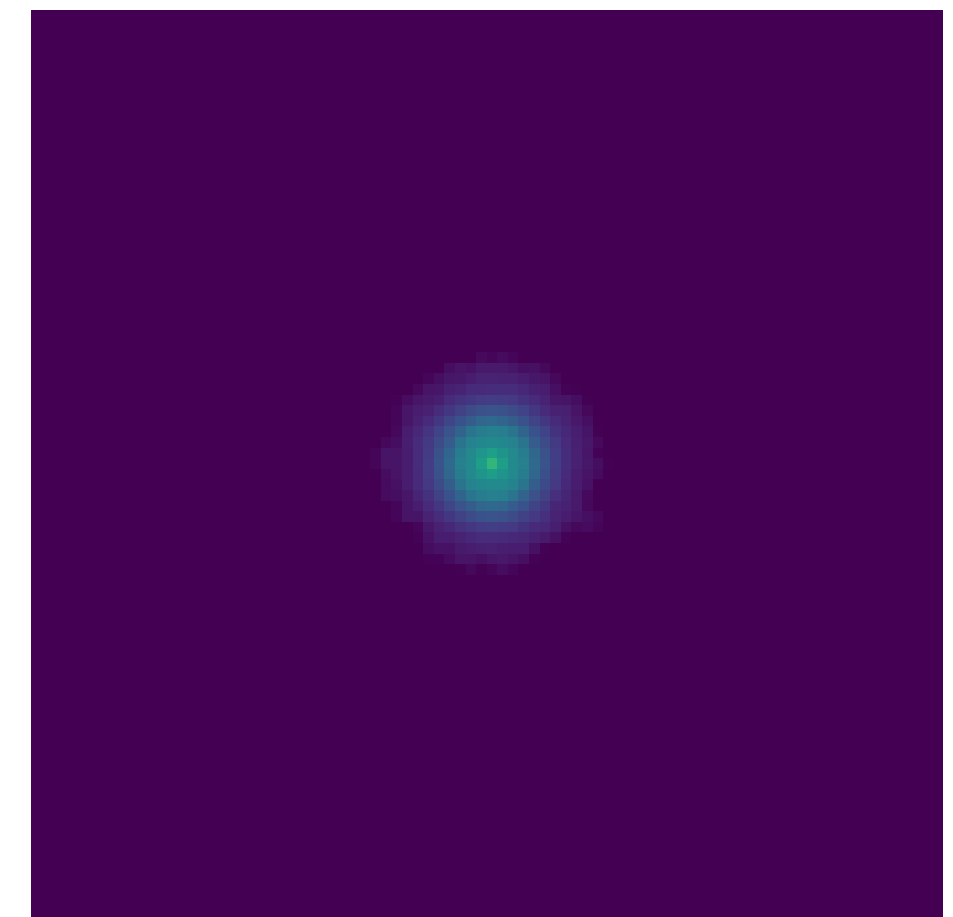
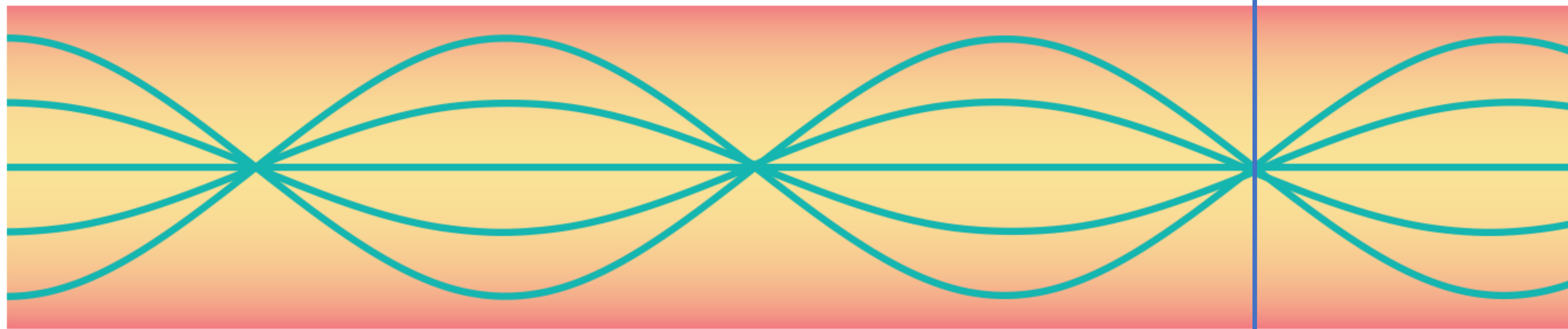
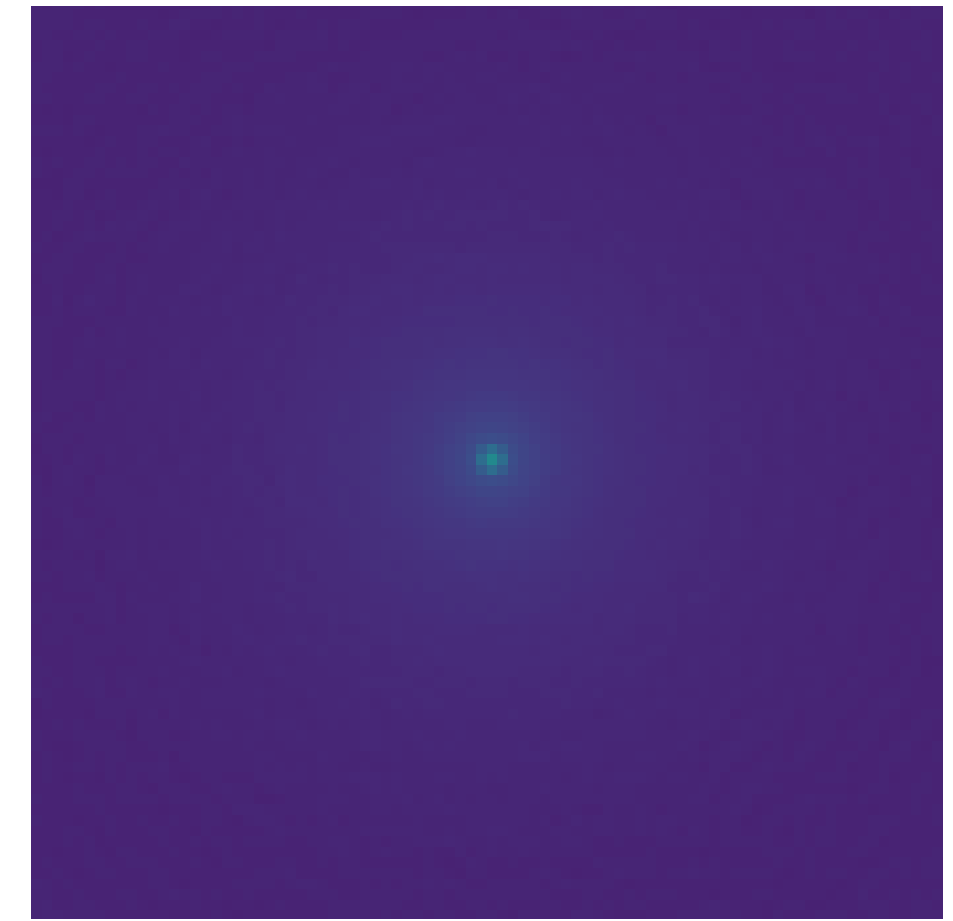
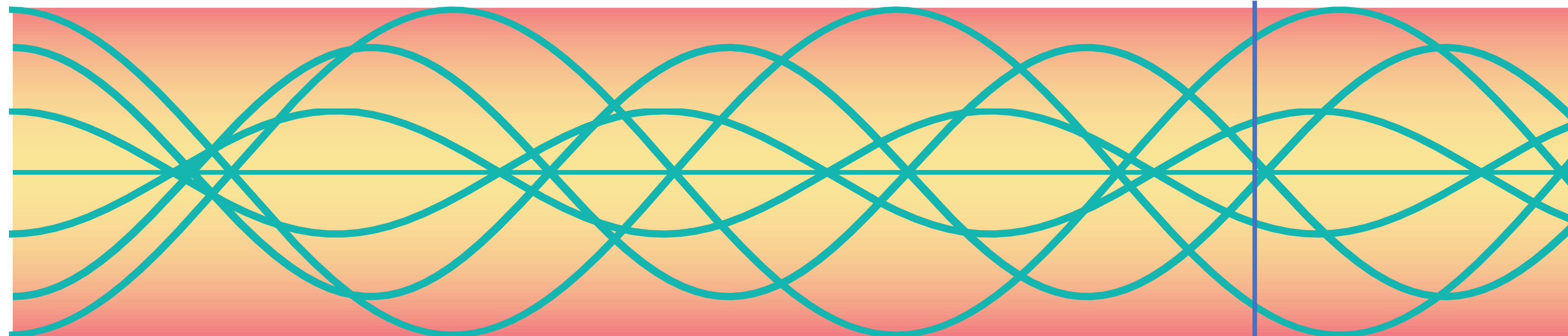
GRIN Fiber



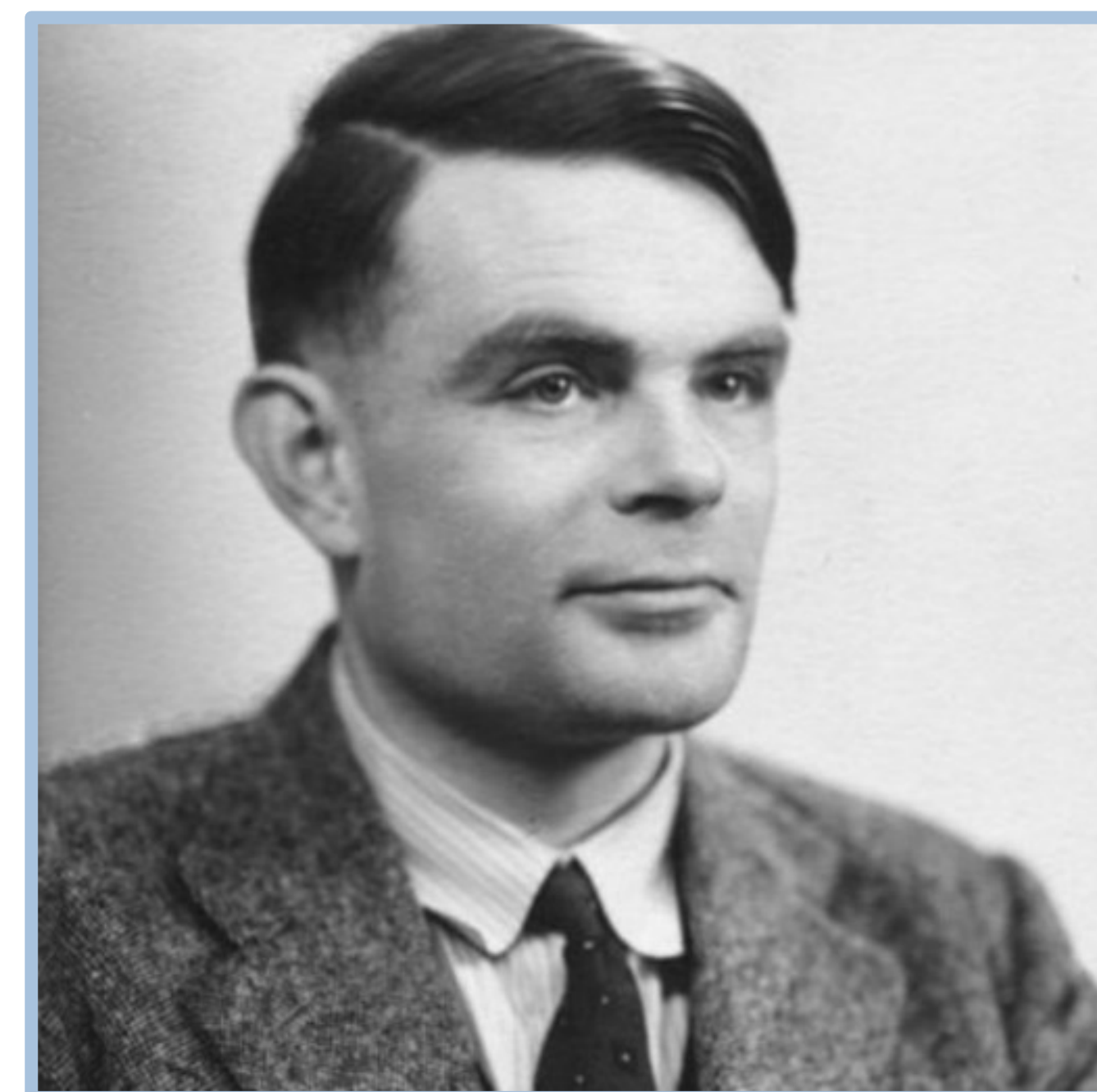
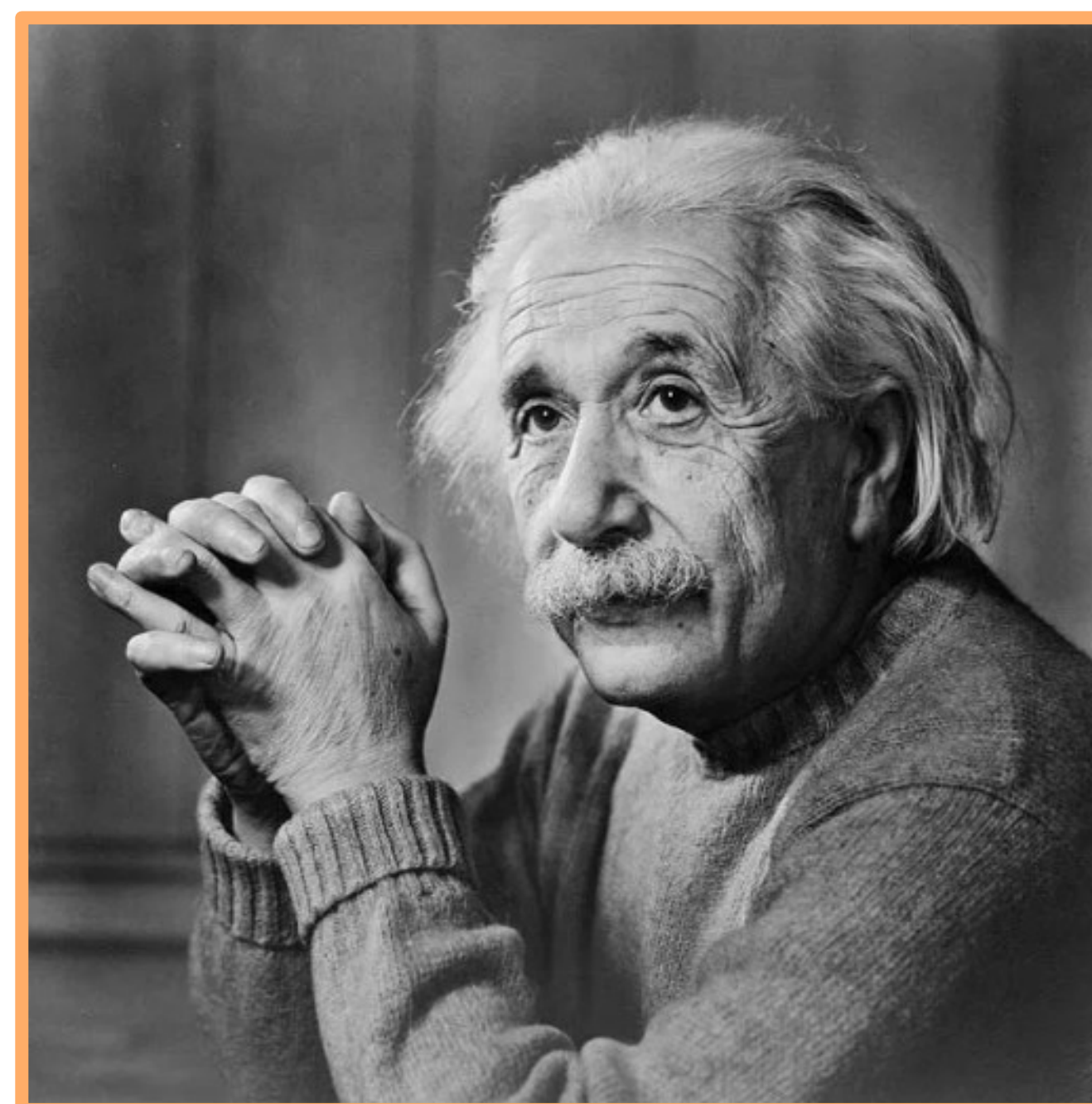
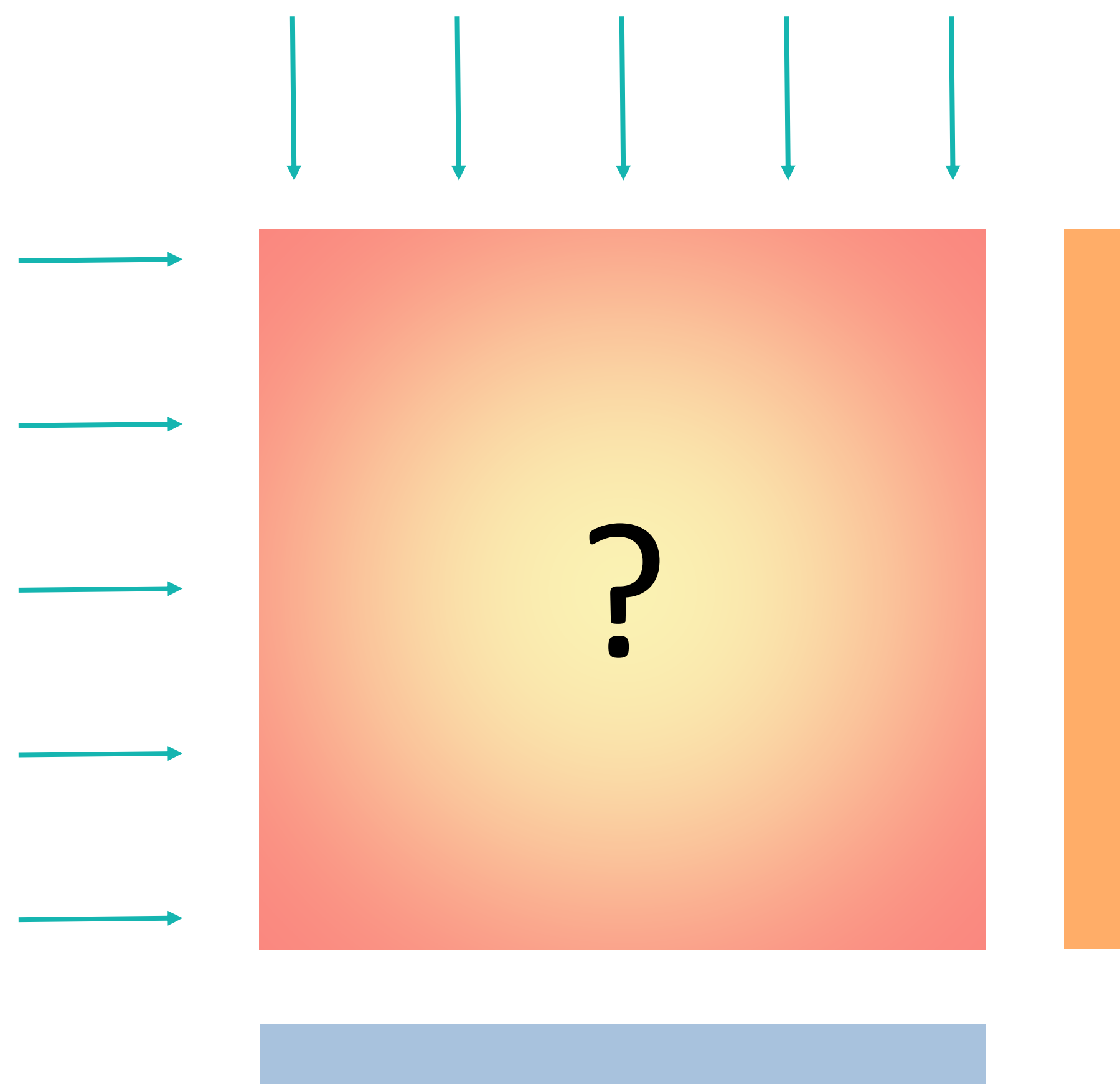
GRIN Fiber



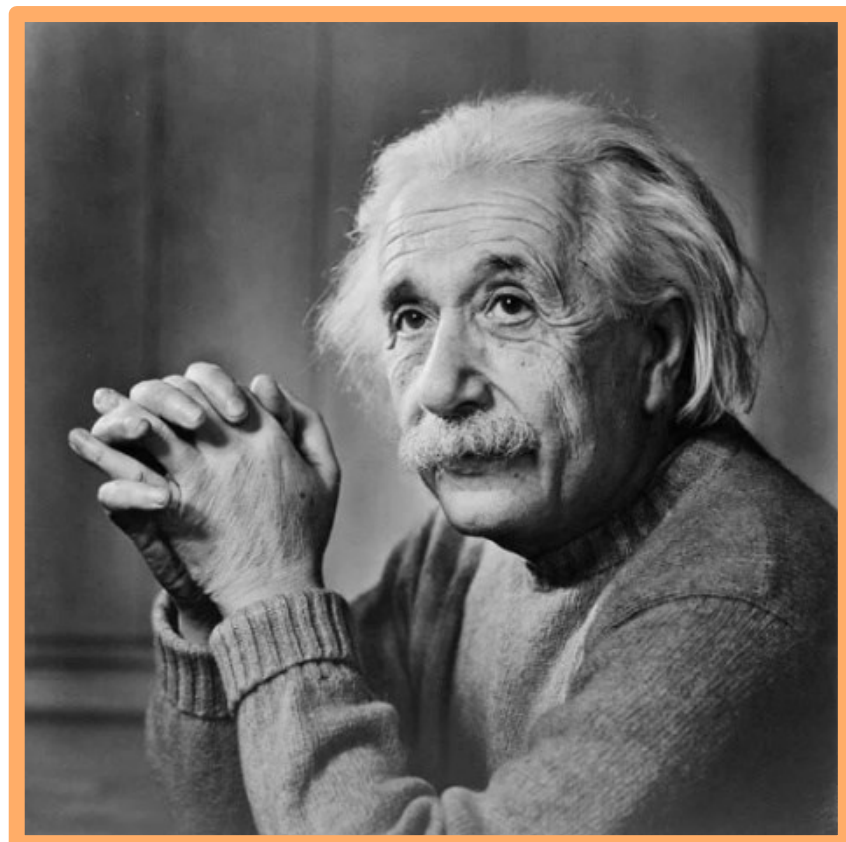
GRIN Fiber



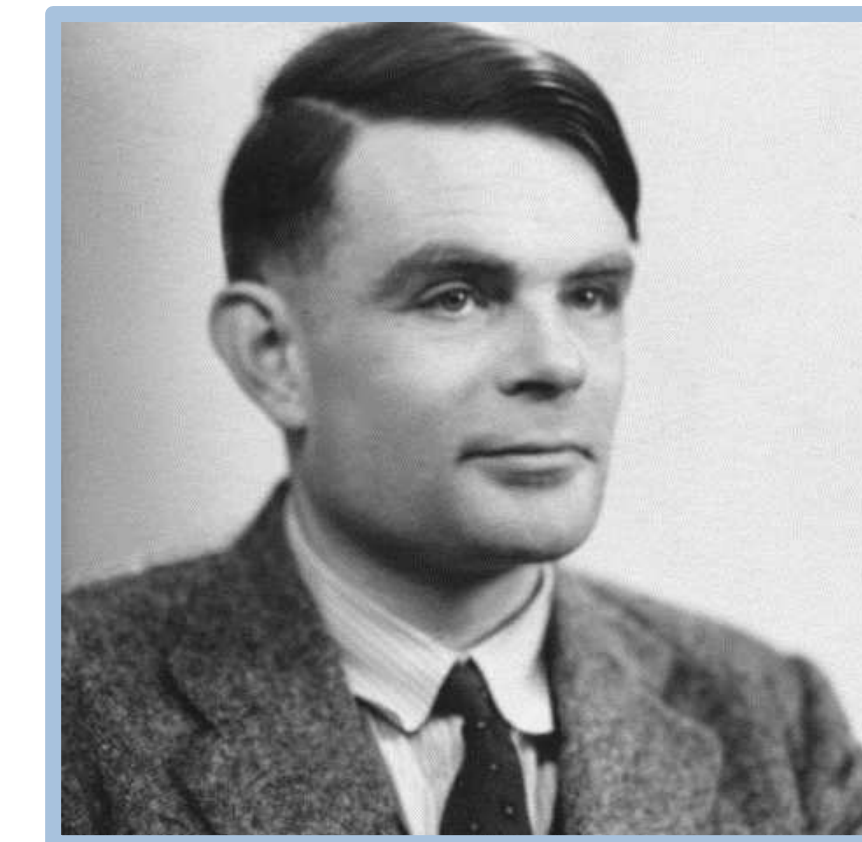
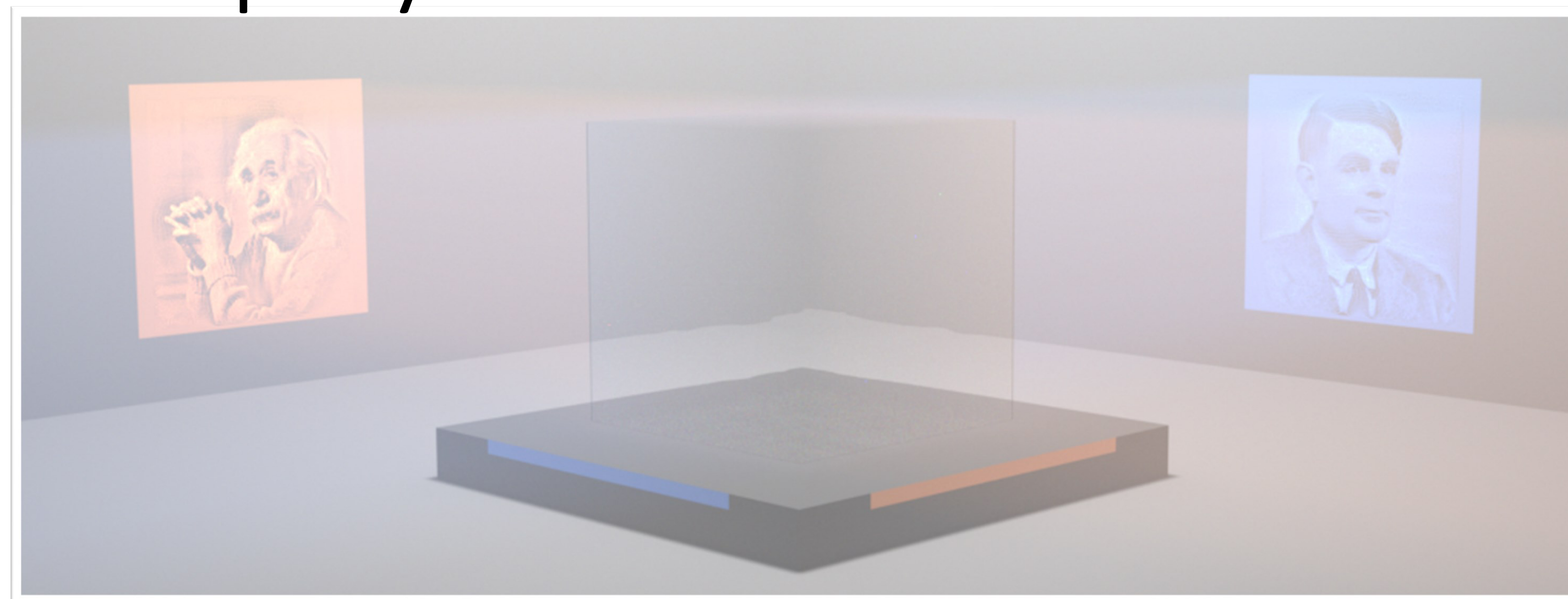
Multiview Display



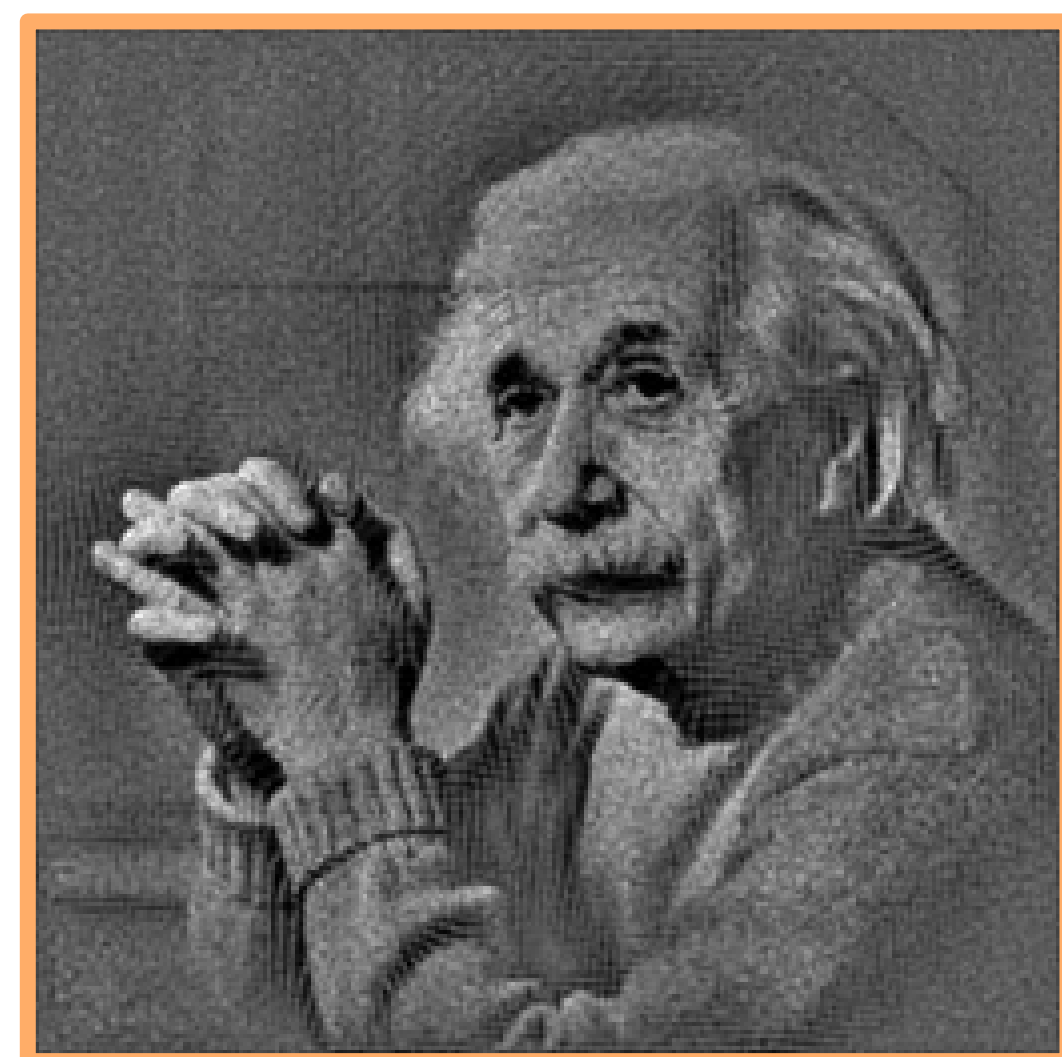
Multiview Display



Target



Target

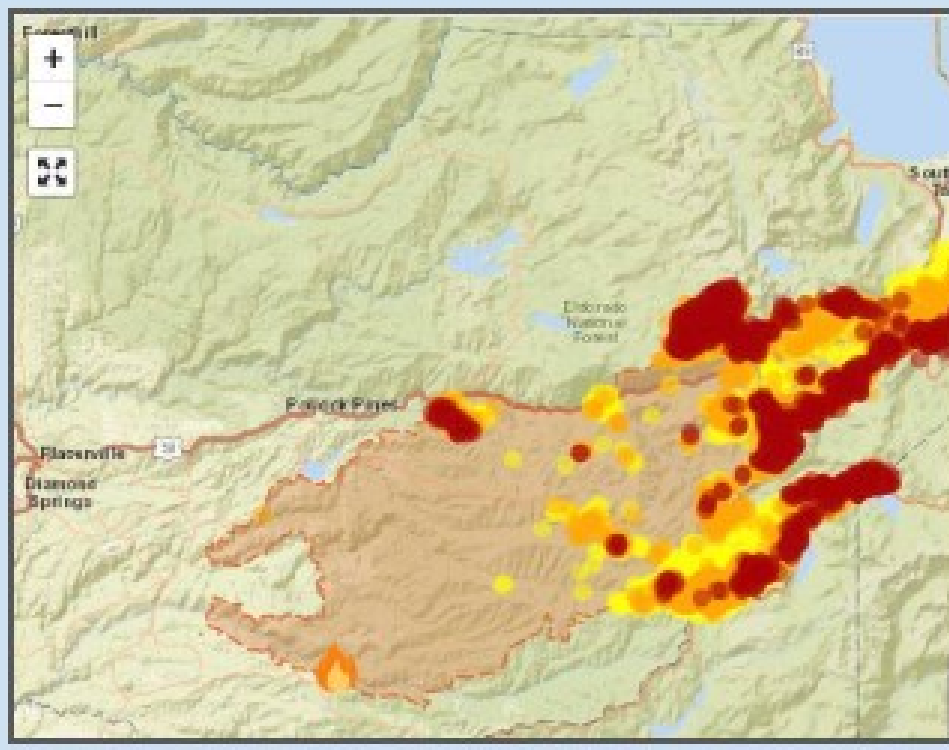


optimization
results

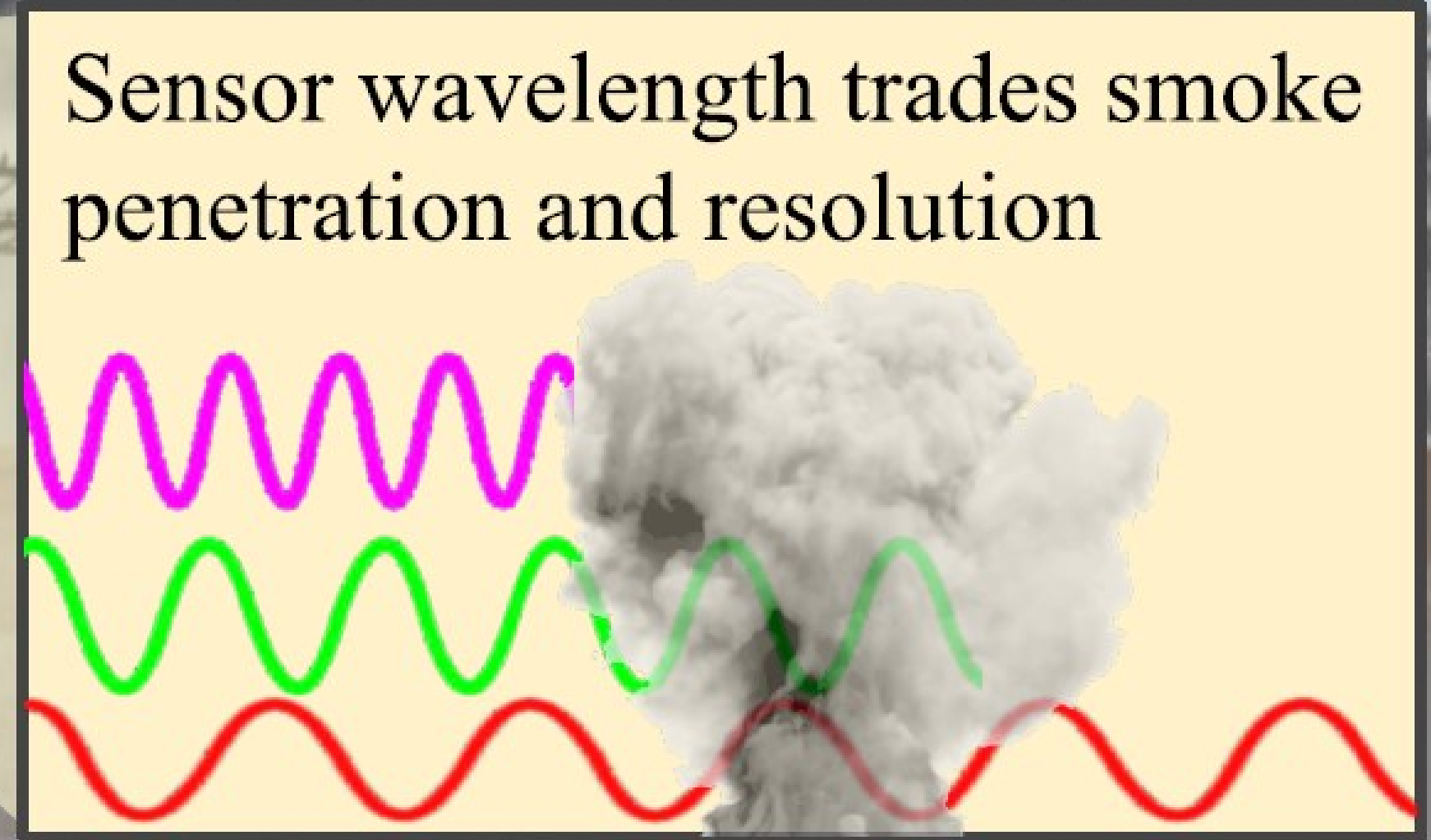


Differentiable rendering for wildfire monitoring

Other's product: 2D fire map, low-res



Ours: Low-flying, granular resolution; team scouts fire plume and environment up-close



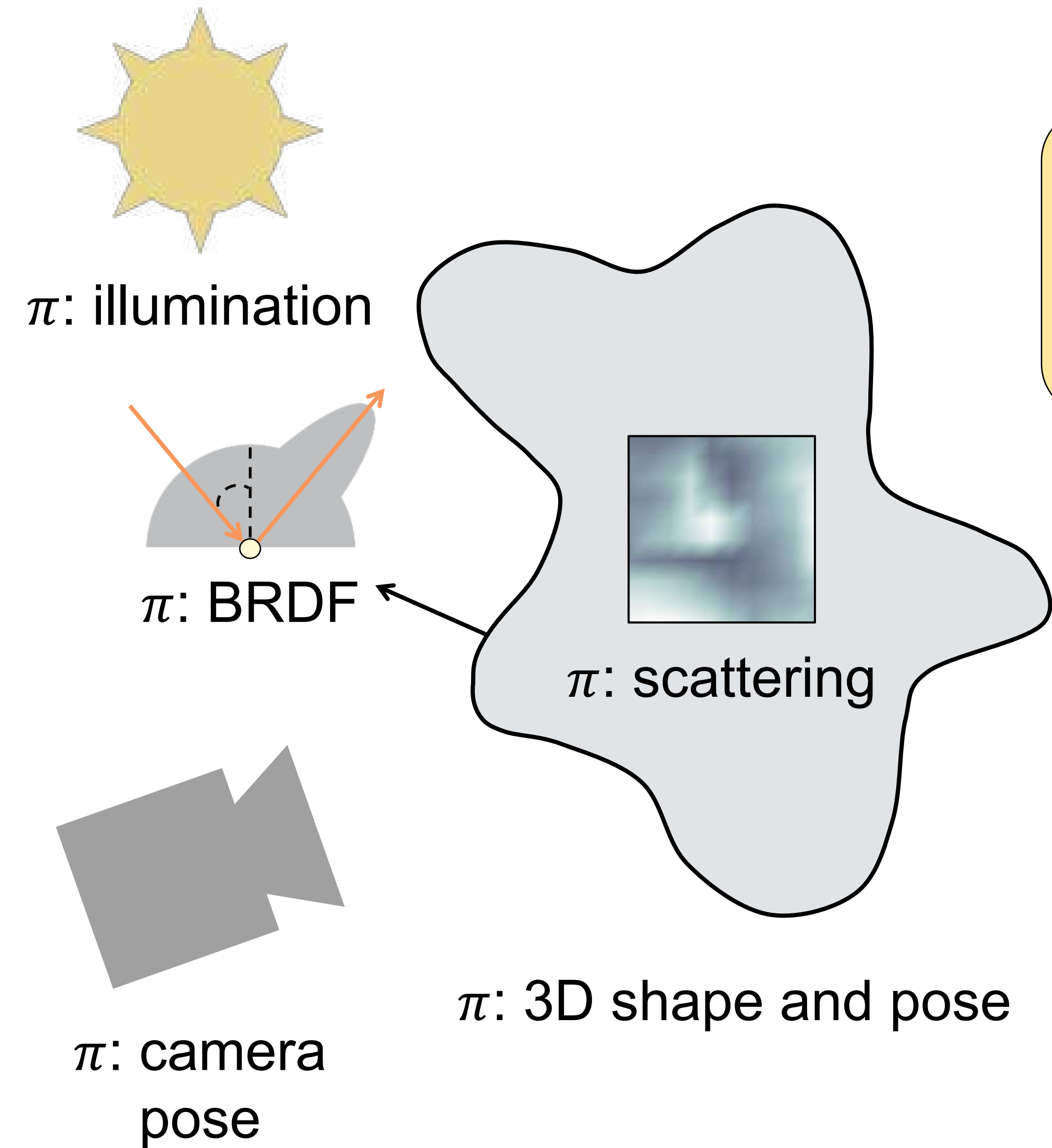
What differentiable rendering does
not give us

Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{img}, \text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

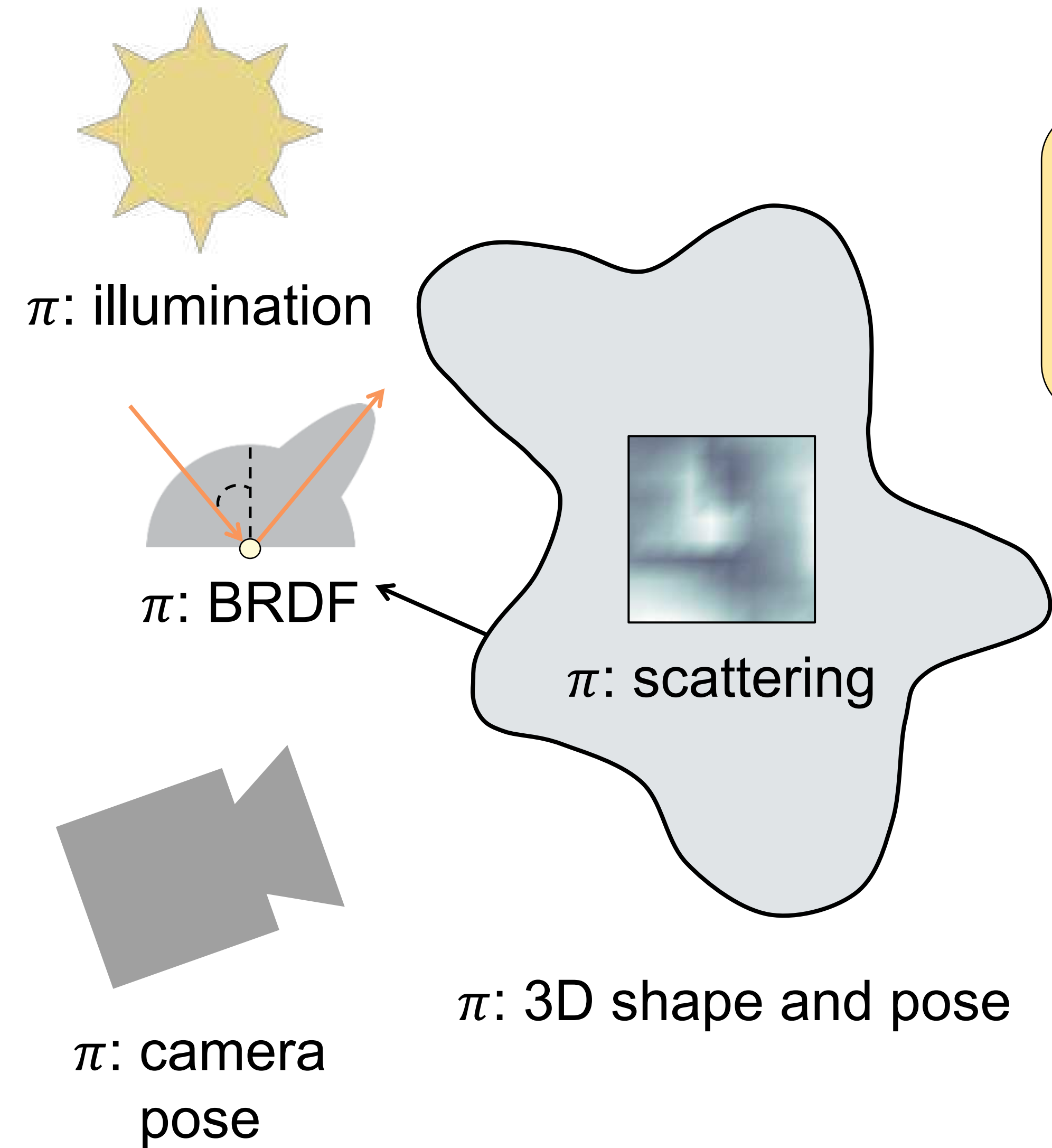

Inverse rendering (a.k.a. analysis by synthesis)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{reference image}, \text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

Inverse rendering (a.k.a. analysis by synthesis)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{reference image}, \text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

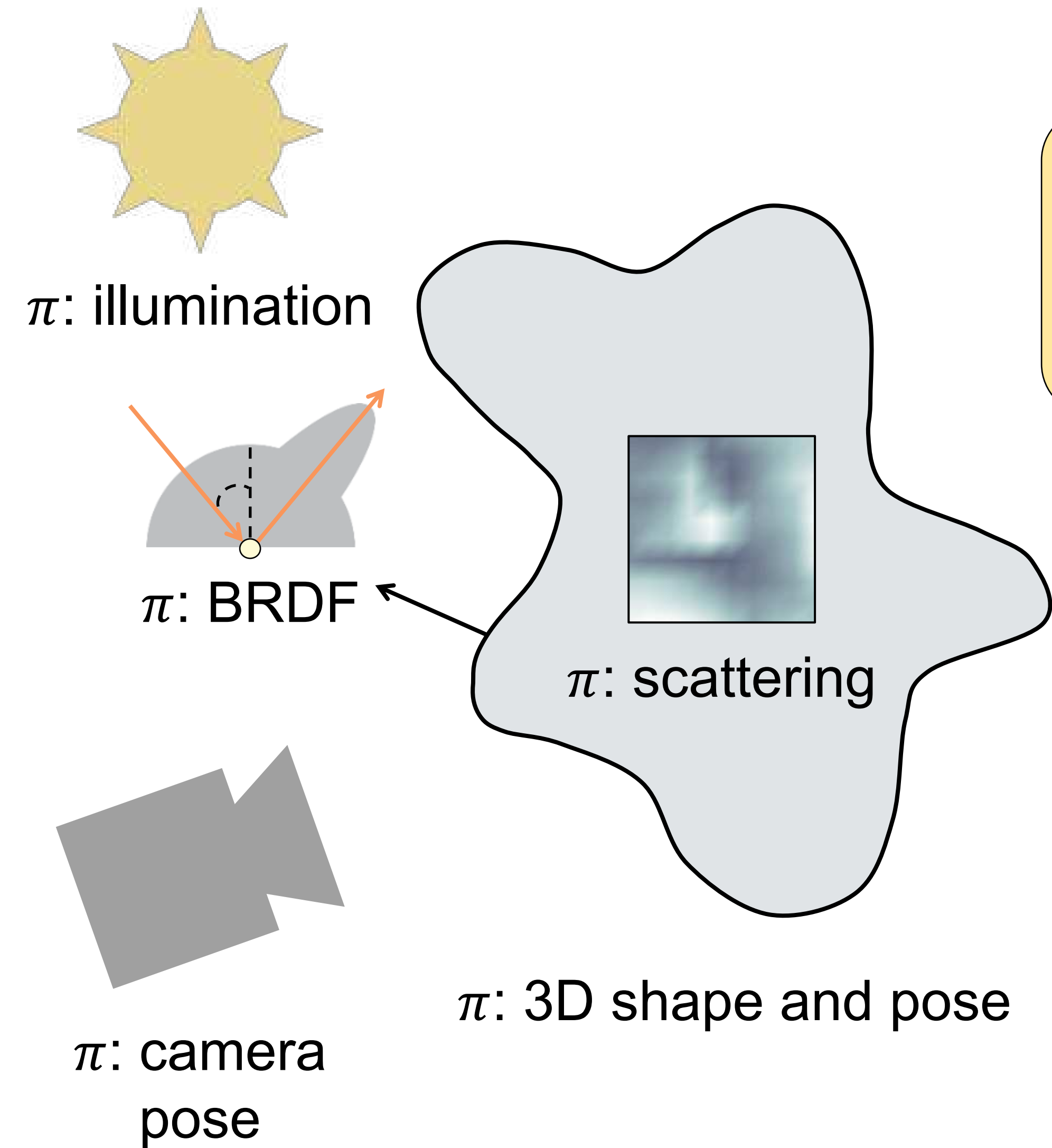
Stochastic gradient descent (e.g., Adam):

initialize $\pi \leftarrow \pi_0$

while (not converged)

$$\text{update } \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$$

Inverse rendering (a.k.a. analysis by synthesis)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{img}, \text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

Stochastic gradient descent (e.g., Adam):

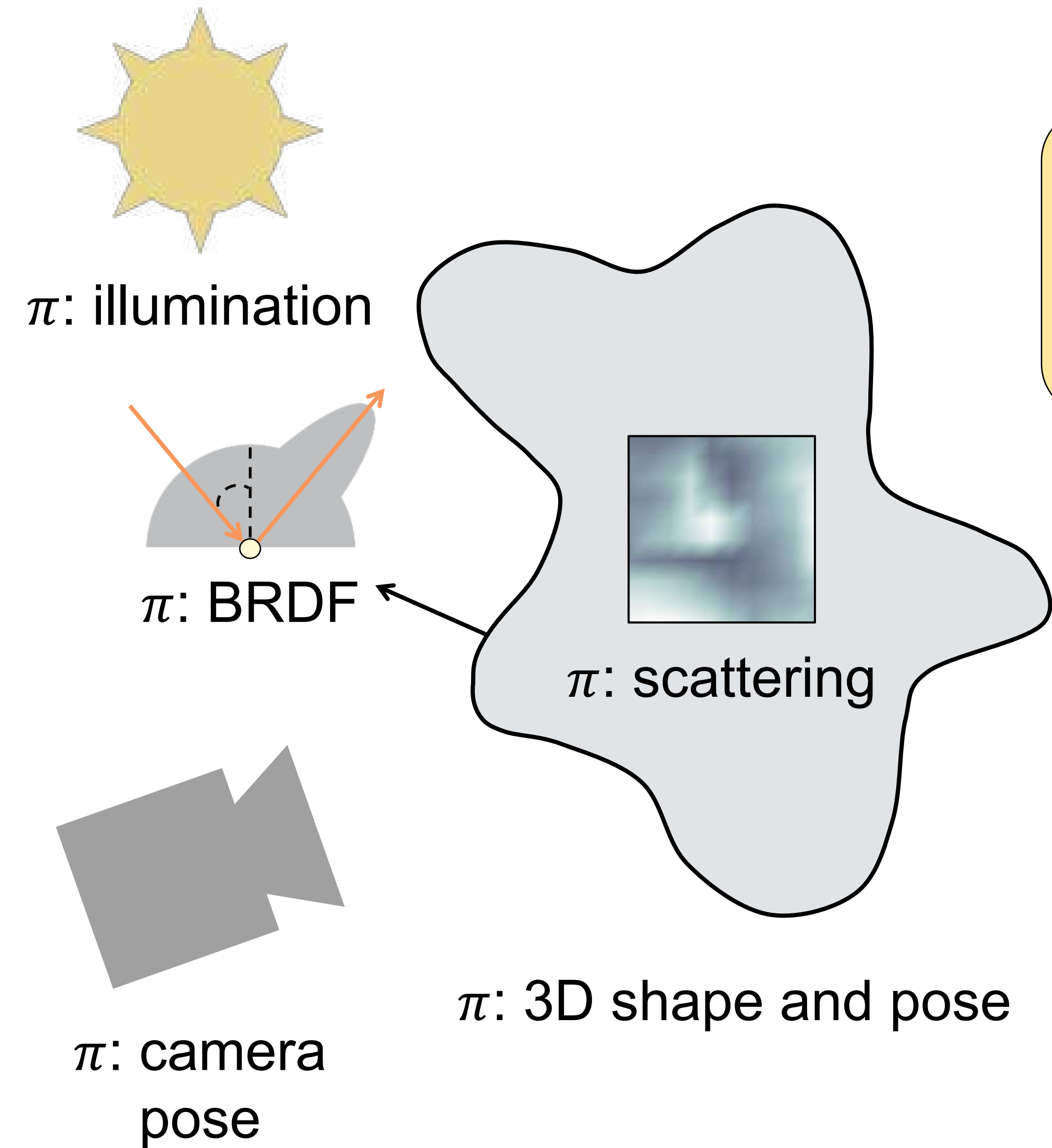
initialize $\pi \leftarrow \pi_0$

while (not converged)

update $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Differentiable rendering

Inverse rendering (a.k.a. analysis by synthesis)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

Stochastic gradient descent (e.g., Adam):

initialize $\pi \leftarrow \pi_0$

while (not converged)

update $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Differentiable rendering

Why we need good initializations

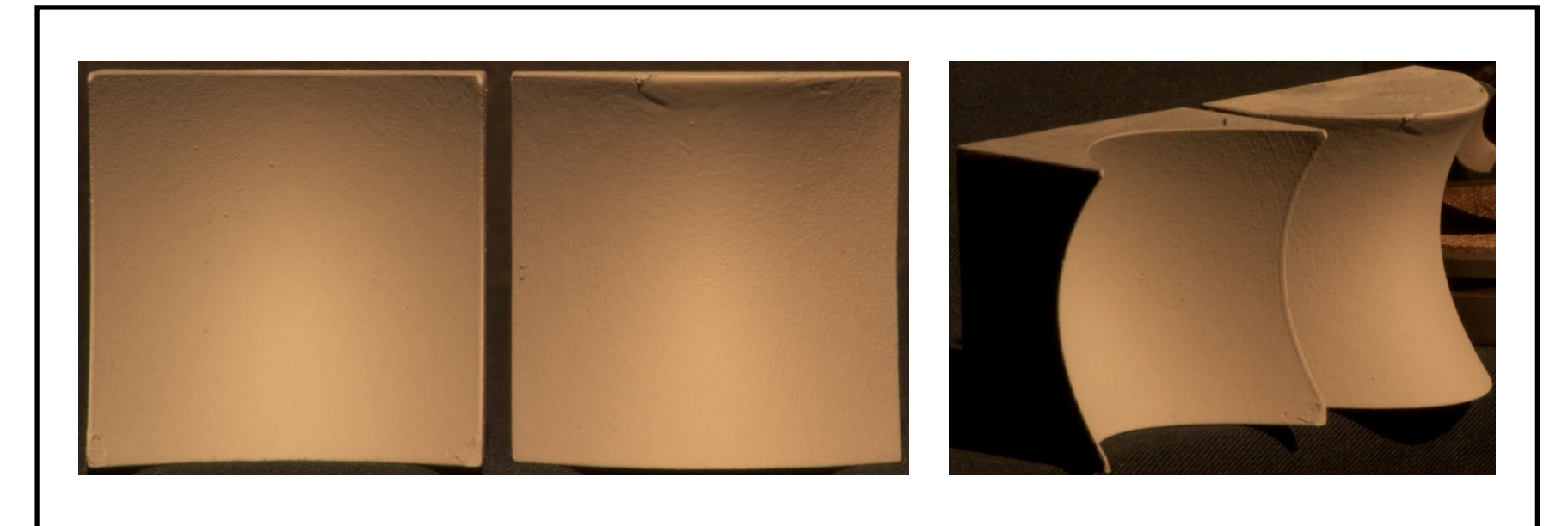
- Analysis-by-synthesis objectives are highly non-convex, non-linear
 - Multiple *local* minima

Why we need good initializations

- Analysis-by-synthesis objectives are highly non-convex, non-linear
 - Multiple *local* minima
- Ambiguities exist between different parameters
 - Multiple *global* minima

Why we need good initializations

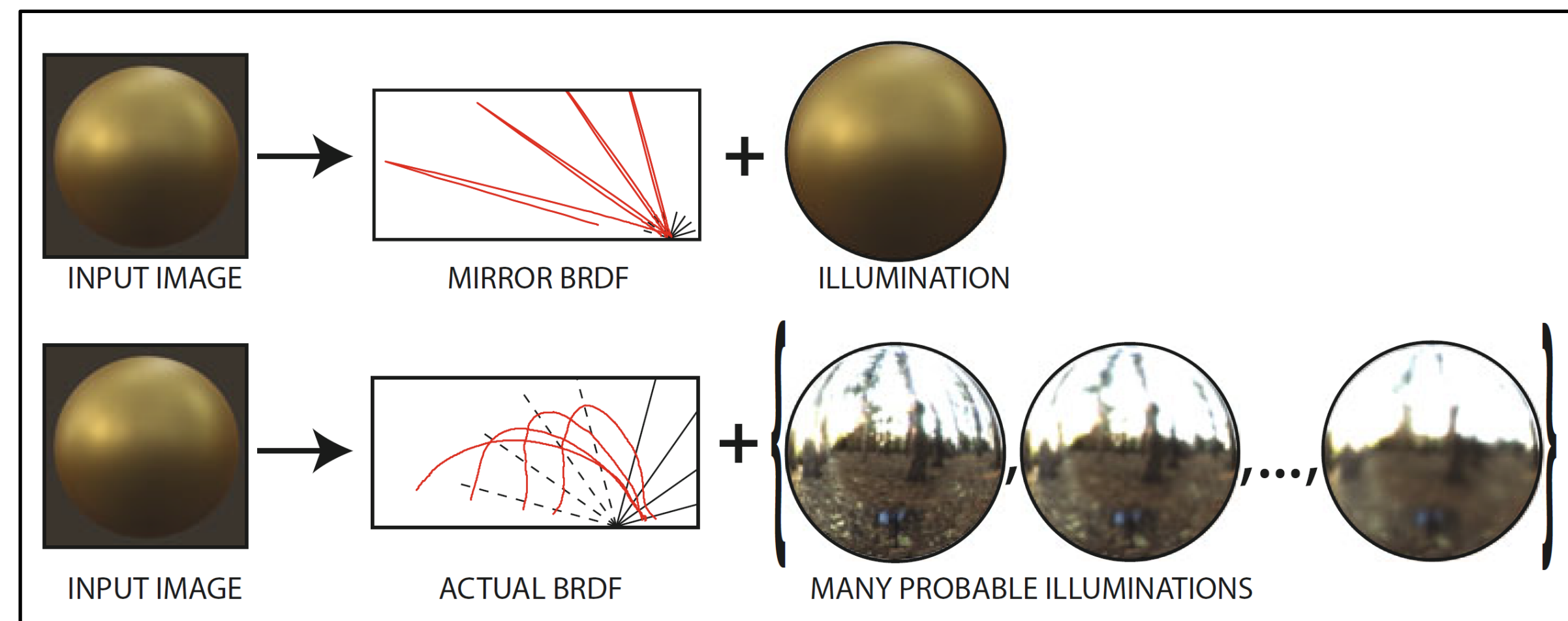
- Analysis-by-synthesis objectives are highly non-convex, non-linear
 - Multiple *local* minima
- Ambiguities exist between different parameters
 - Multiple *global* minima



Ambiguities between shape and lighting
[Xiong et al. 2015]



Ambiguities between scattering parameters [Zhao et al. 2014]



Ambiguities between BRDF and lighting
[Romeiro and Zickler 2010]

Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$


Stochastic gradient descent (e.g., Adam):

initialize $\pi \leftarrow \pi_0$

while (not converged)

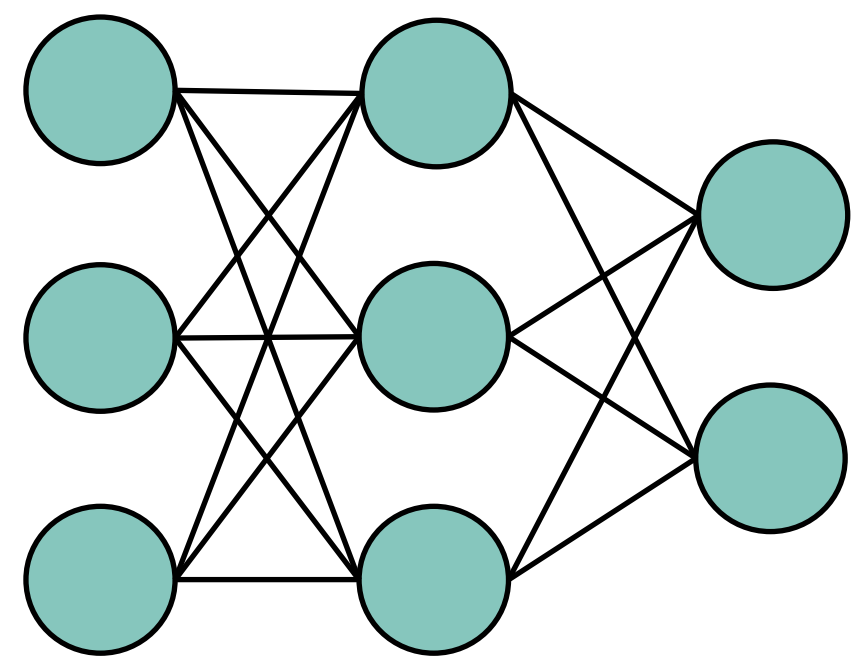
update $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Differentiable rendering

Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

Neural network



Stochastic gradient descent (e.g., Adam):

initialize $\pi \leftarrow \pi_0$

while (not converged)

update $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Differentiable rendering

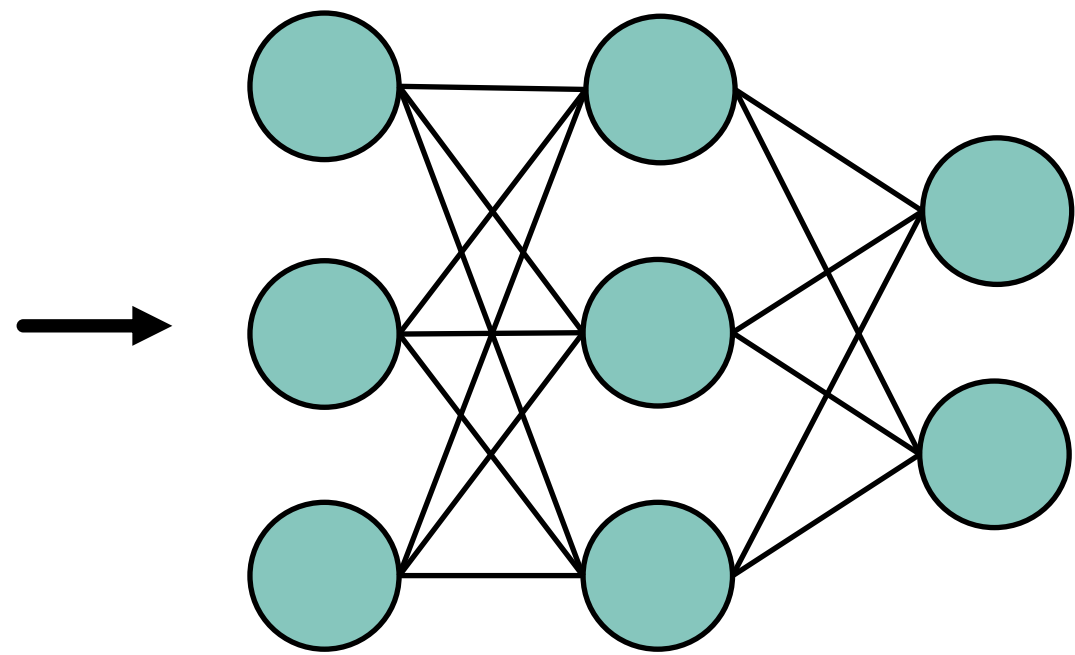
Inverse rendering (a.k.a. analysis by synthesis)

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$


Learned initializations help:

- avoid local minima
- accelerate convergence



Neural network

Stochastic gradient descent (e.g., Adam):

initialize $\pi \leftarrow \pi_0$

while (not converged)

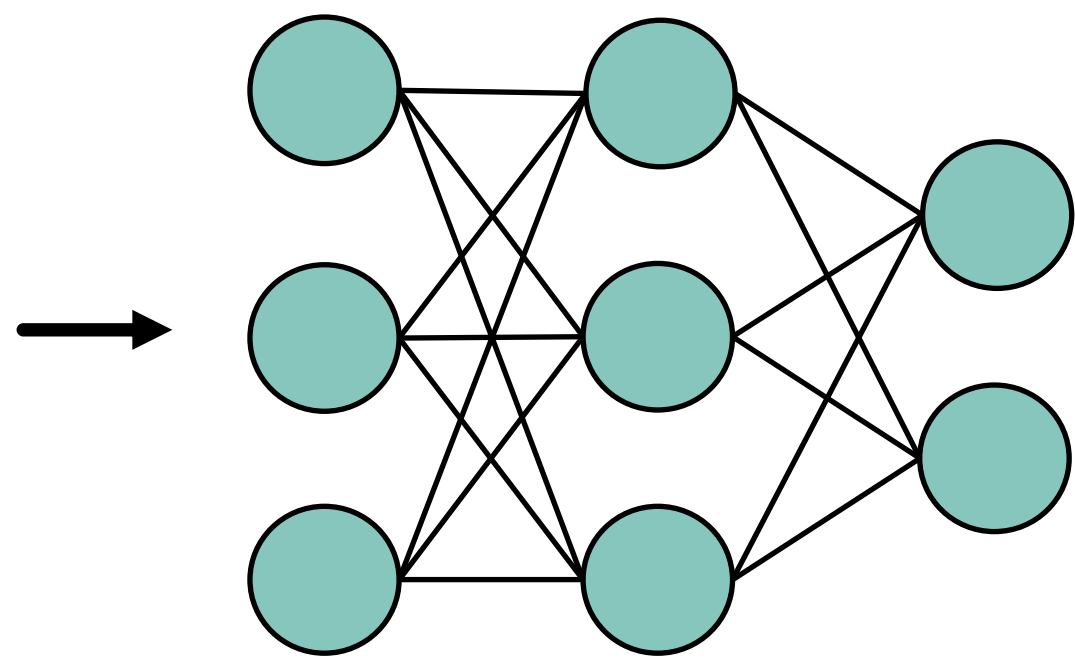
update $\pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi}$

Differentiable rendering

Inverse rendering (a.k.a. analysis by synthesis)

Learned initializations help:

- avoid local minima
- accelerate convergence



Neural network

Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

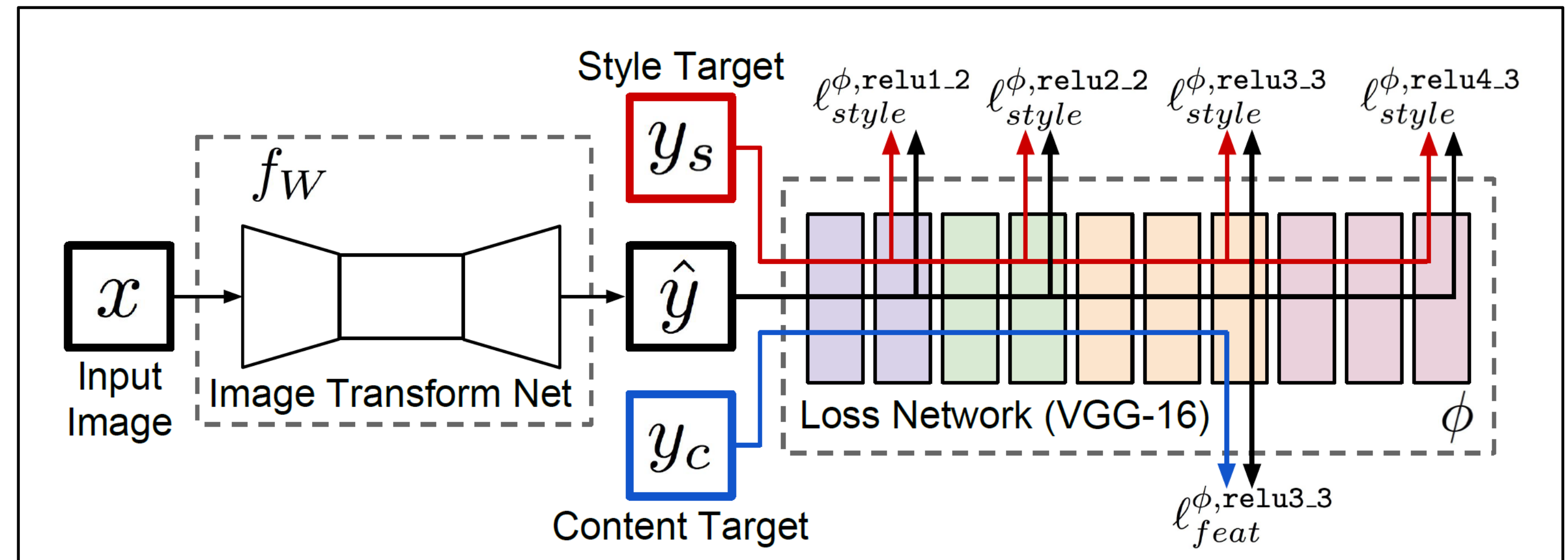
Stochastic gradient descent (e.g., Adam):

$$\begin{aligned} &\text{initialize } \pi \leftarrow \pi_0 \\ &\text{while (not converged)} \\ &\quad \text{update } \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \end{aligned}$$

Differentiable rendering

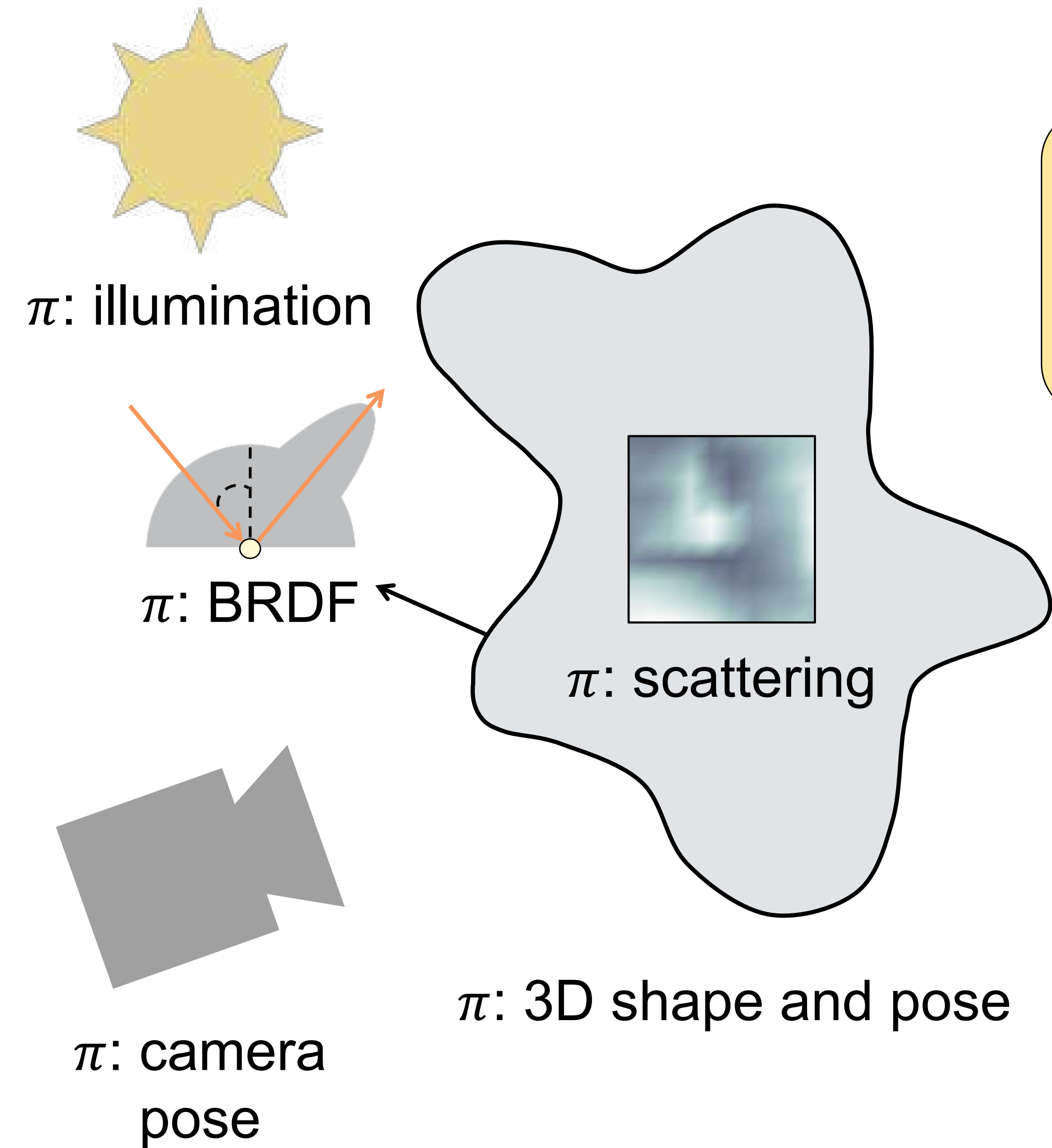
Why we need discriminative loss functions

- Well-designed loss functions can help reduce ambiguities
- Perceptual losses can help emphasize design aspects that matter
- Differentiable rendering can be combined with any loss function that can be backpropagated through



VGG-based *perceptual loss* [Johnson et al. 2016]

Inverse rendering (a.k.a. analysis by synthesis)



Analysis-by-synthesis optimization:

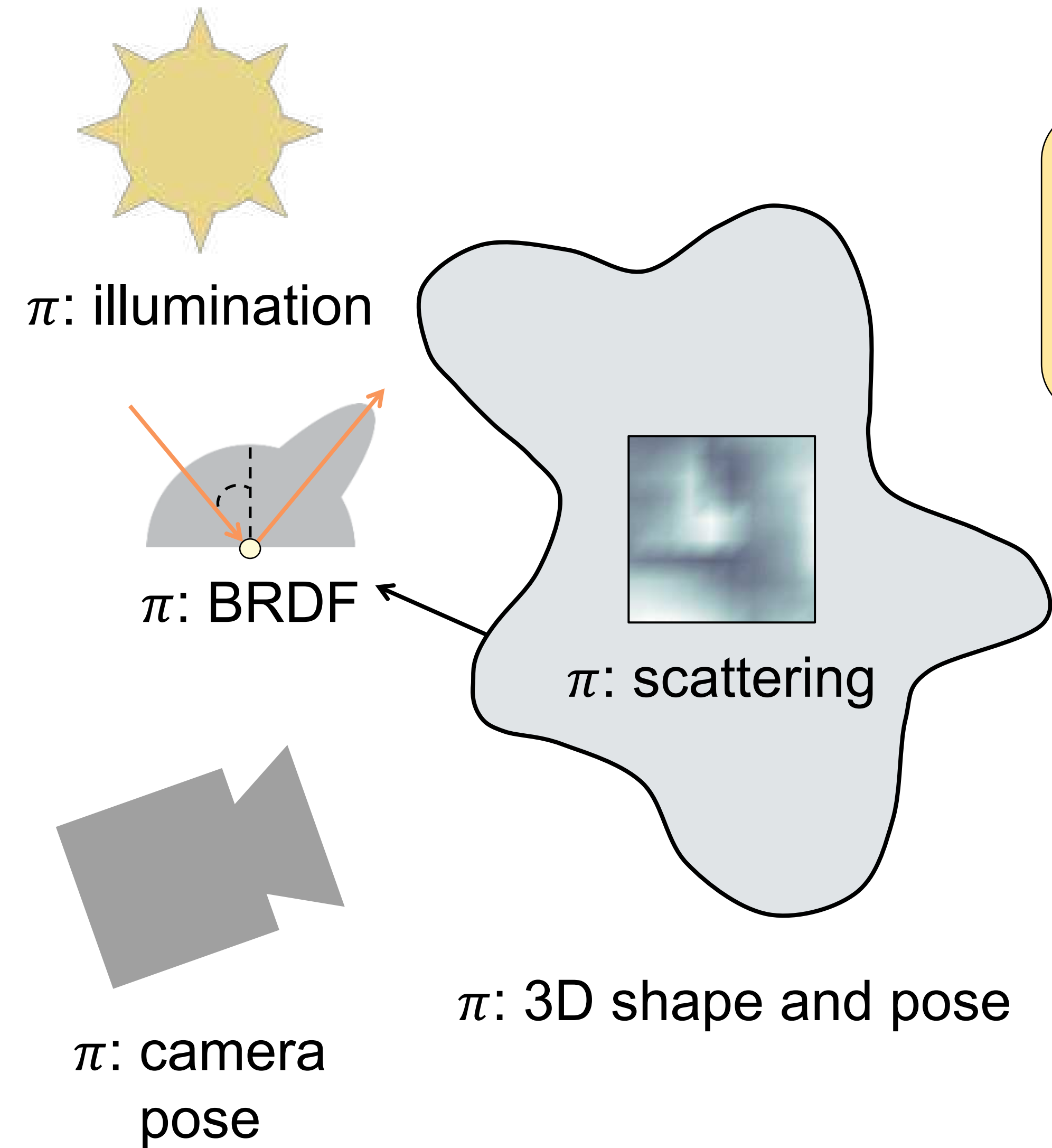
$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$


Stochastic gradient descent (e.g., Adam):

$$\begin{aligned} &\text{initialize } \pi \leftarrow \pi_0 \\ &\text{while (not converged)} \\ &\quad \text{update } \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \end{aligned}$$

Differentiable rendering

Inverse rendering (a.k.a. analysis by synthesis)



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss} \left[\text{image}, \text{render} \left(\begin{array}{c} \text{scene} \\ \text{unknowns } \pi \end{array} \right) \right]$$

Stochastic gradient descent (e.g., Adam):

$$\begin{aligned} &\text{initialize } \pi \leftarrow \pi_0 \\ &\text{while (not converged)} \\ &\quad \text{update } \pi \leftarrow \pi + \eta \cdot \frac{d\text{loss}(\pi)}{d\pi} \end{aligned}$$

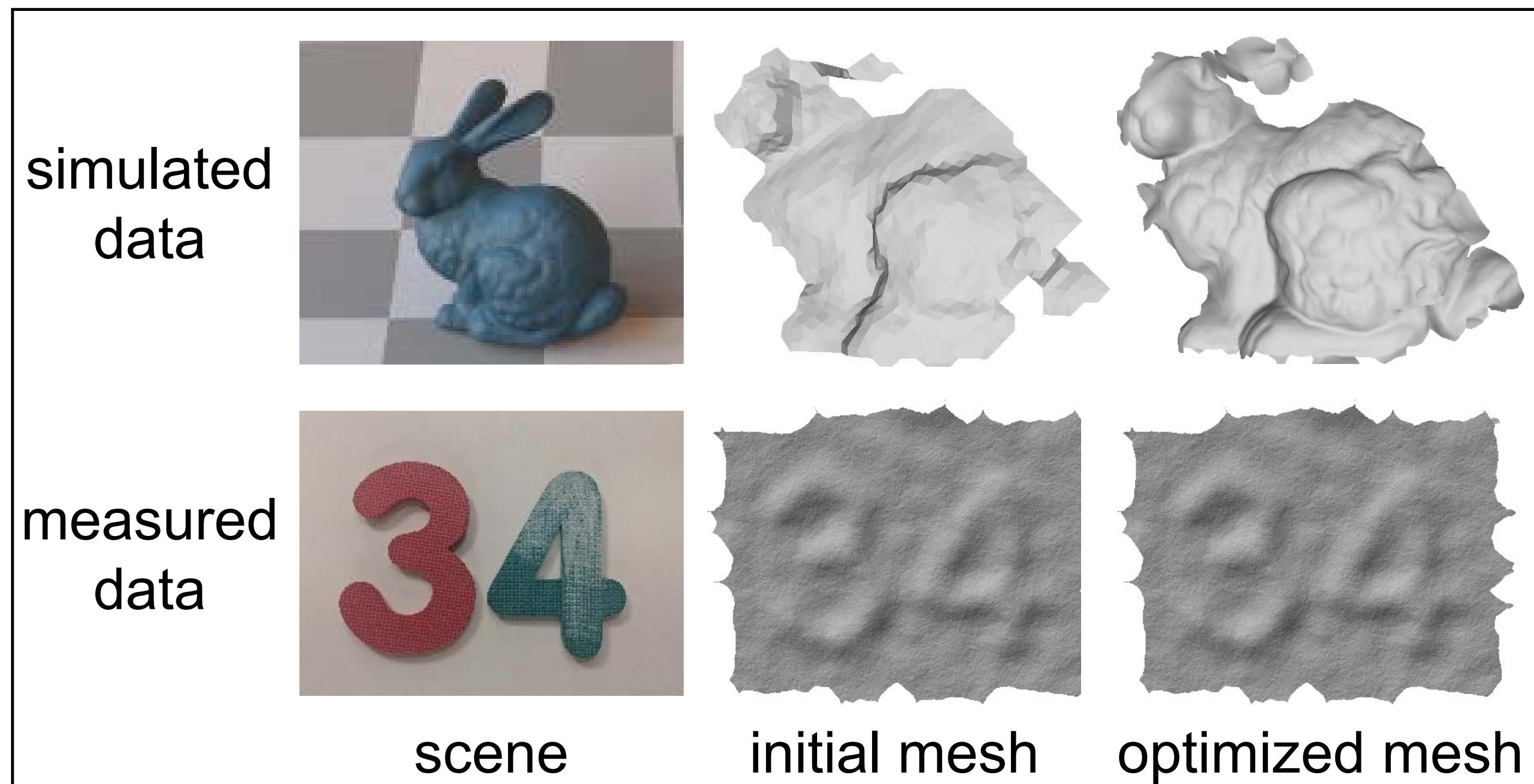
Differentiable rendering

High signal-to-noise ratio is critical

- The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements

High signal-to-noise ratio is critical

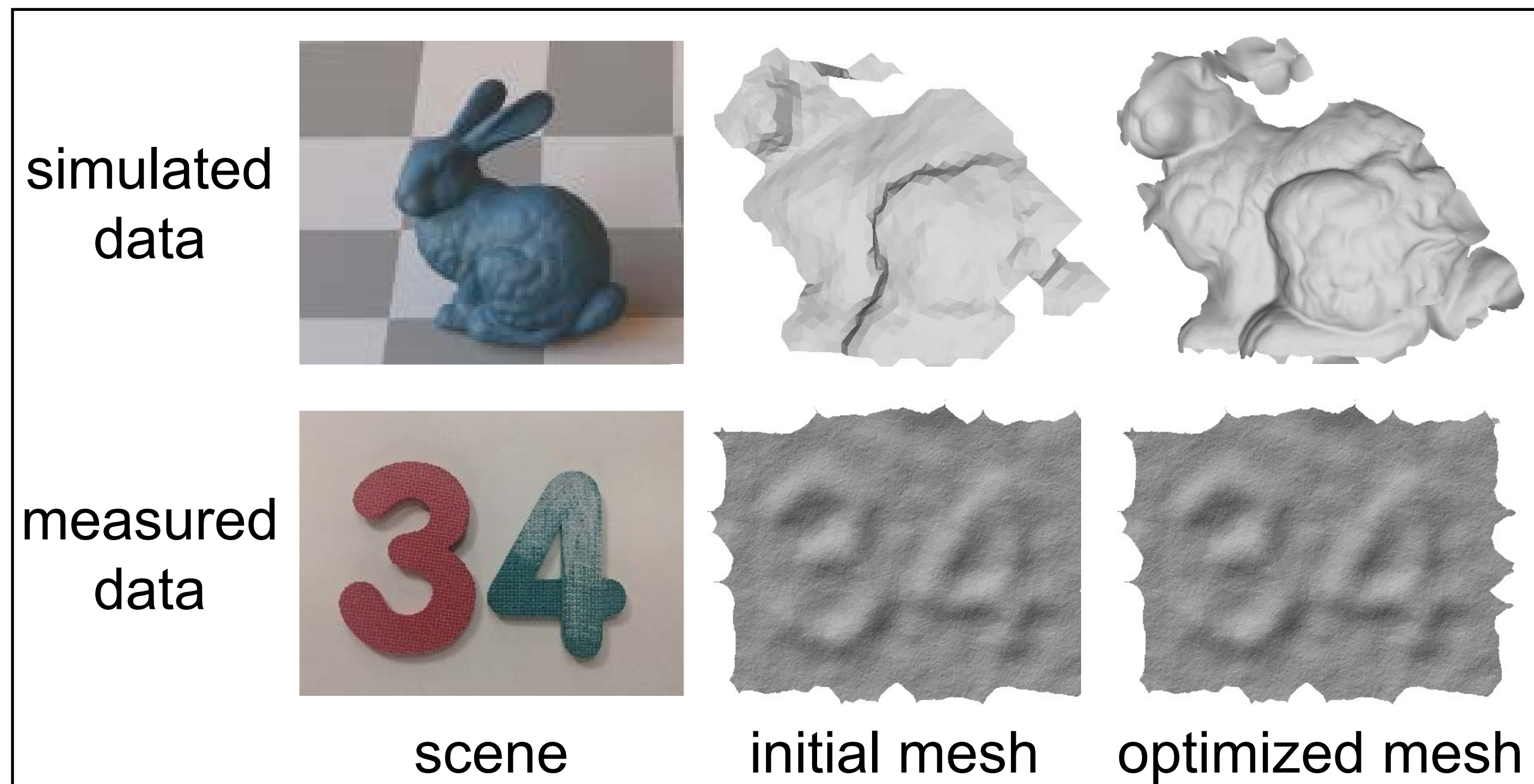
- The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements



Non-line-of-sight imaging [Tsai et al. 2019]

High signal-to-noise ratio is critical

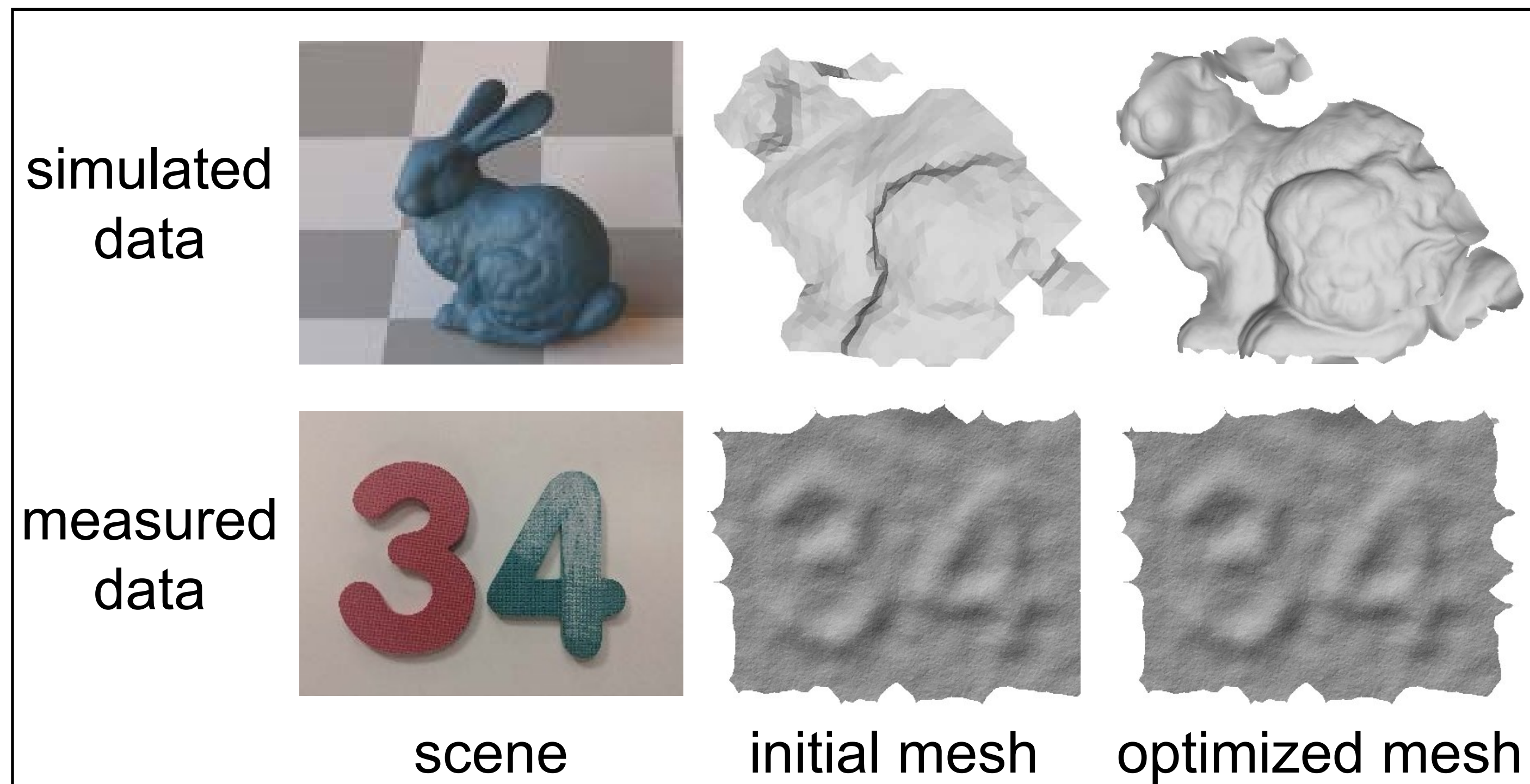
- The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements
- We need reliable camera models (noise, aberrations, other non-idealities)



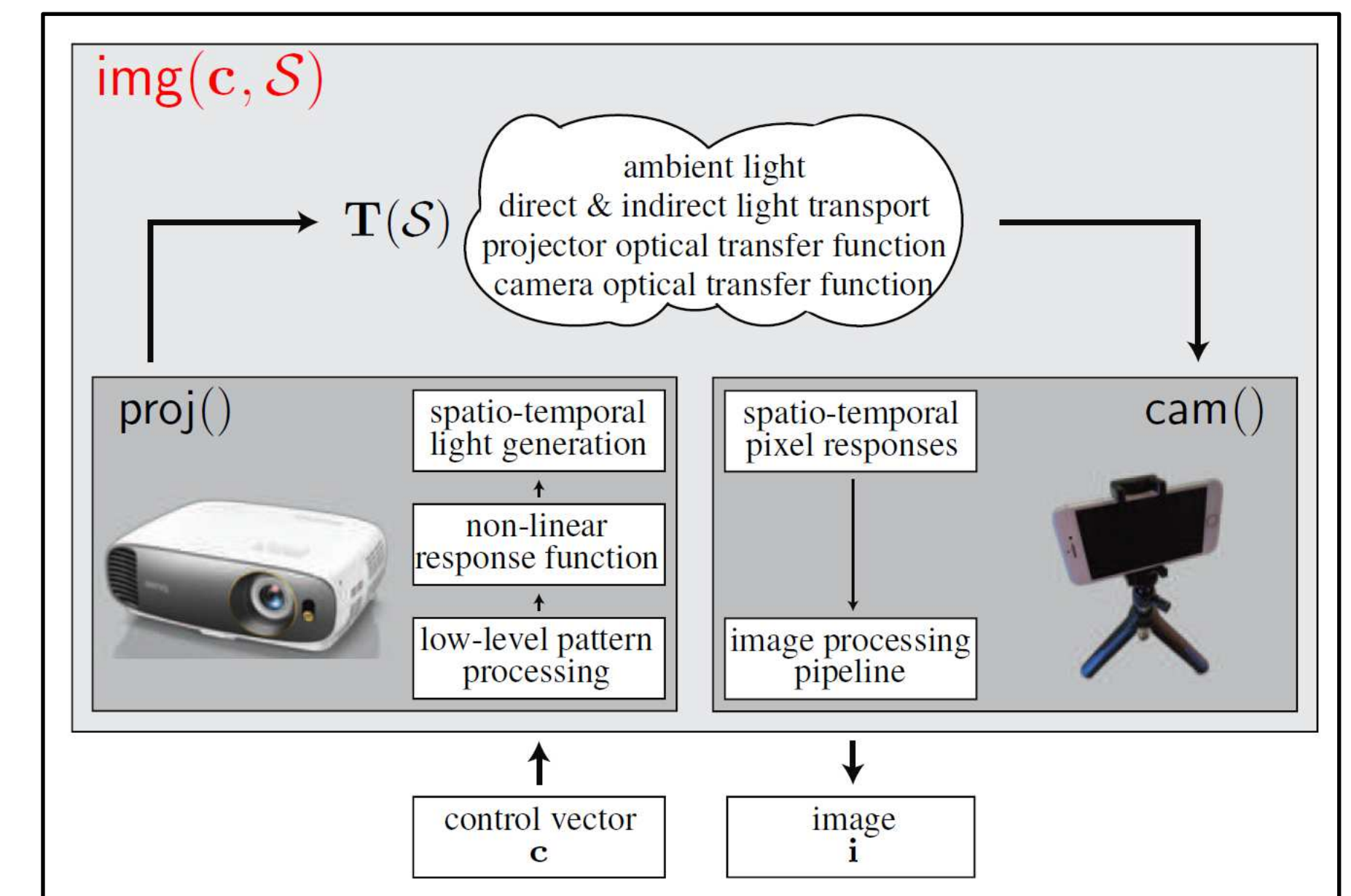
Non-line-of-sight imaging [Tsai et al. 2019]

High signal-to-noise ratio is critical

- The extent to which we can improve upon an initialization strongly depends on the signal-to-noise ratio of our measurements
- We need reliable camera models (noise, aberrations, other non-idealities)



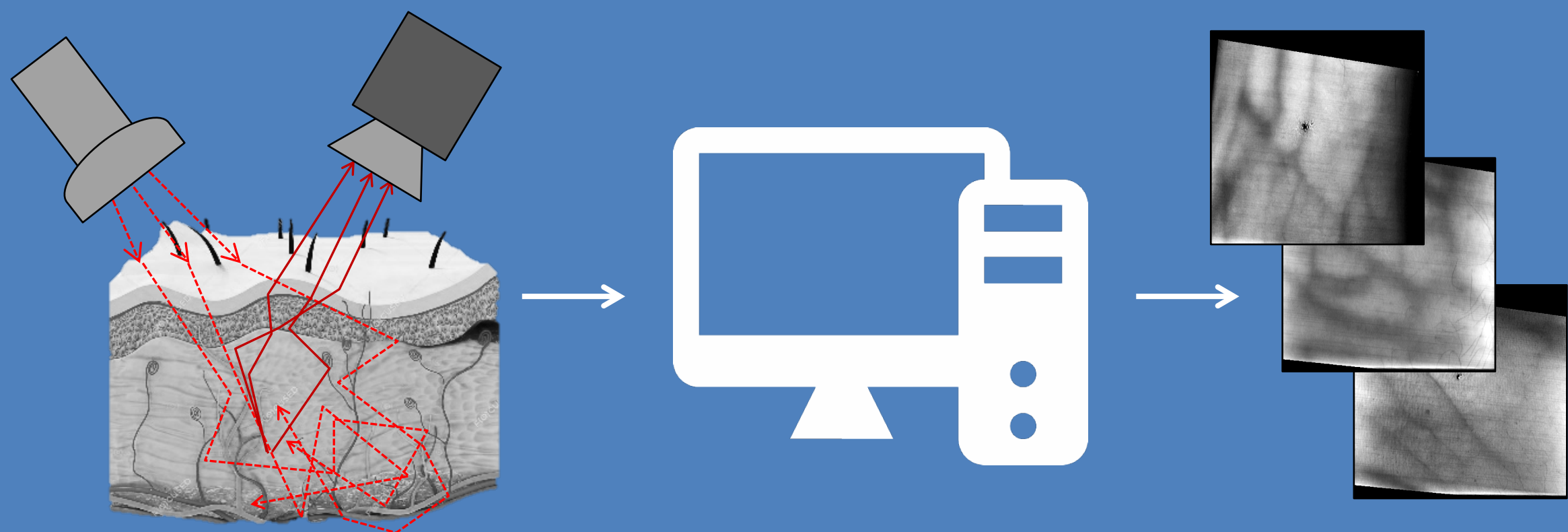
Non-line-of-sight imaging [Tsai et al. 2019]



Optical gradient descent [Chen et al. 2020]

Physics-based rendering and its applications to computational imaging

forward rendering

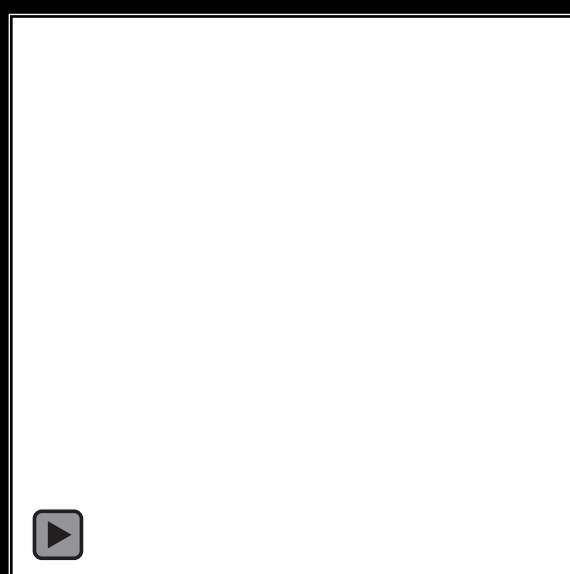


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

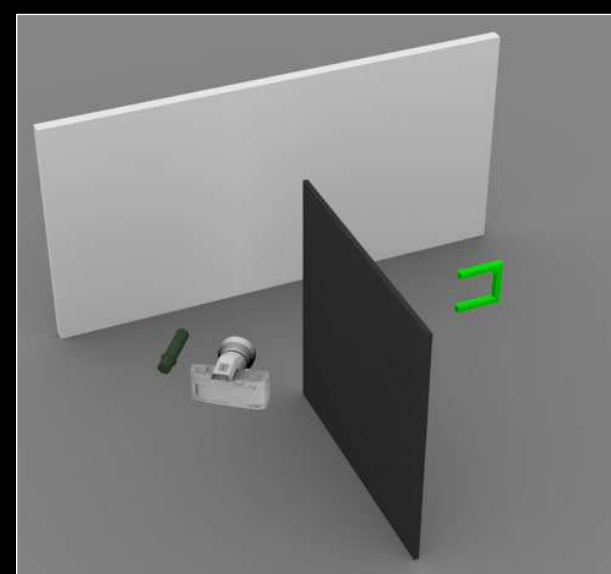
inverse rendering



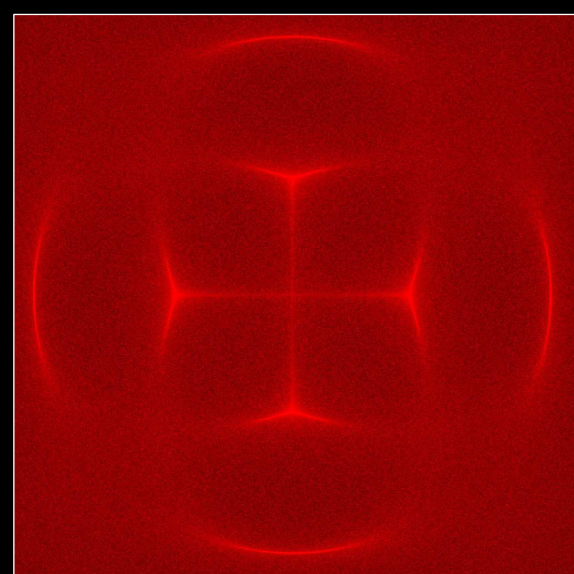
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



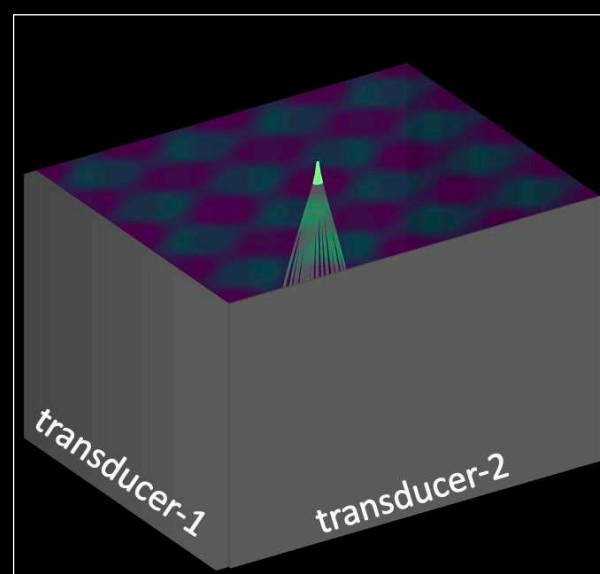
time-of-flight
imaging



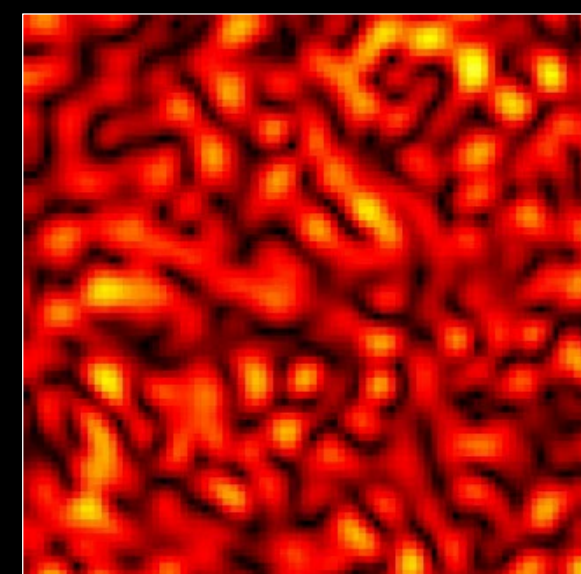
non-line-of-sight
imaging



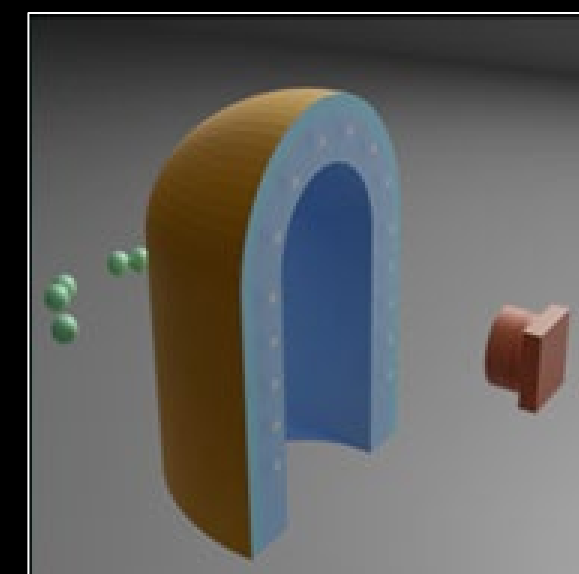
acousto-optic
lensing



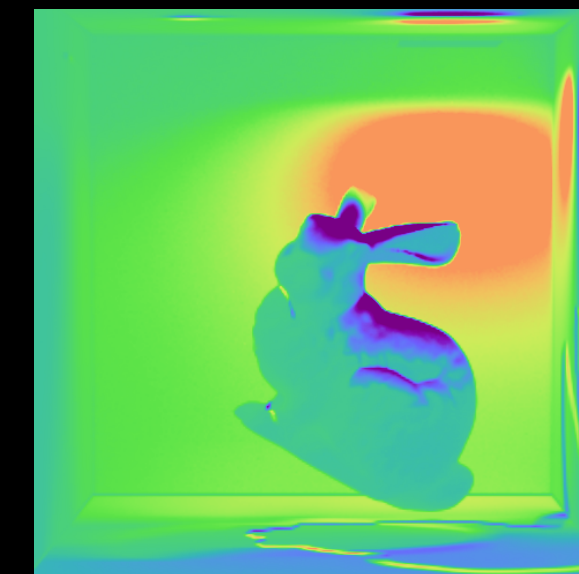
ultrafast light
scanning



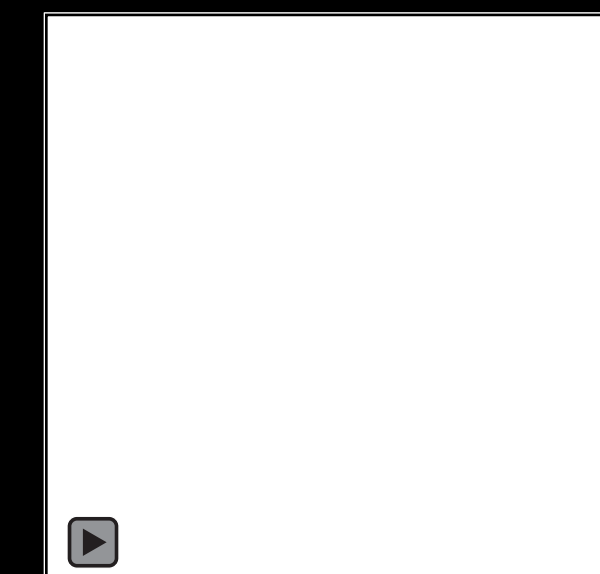
speckle
imaging



tactile sensor
design



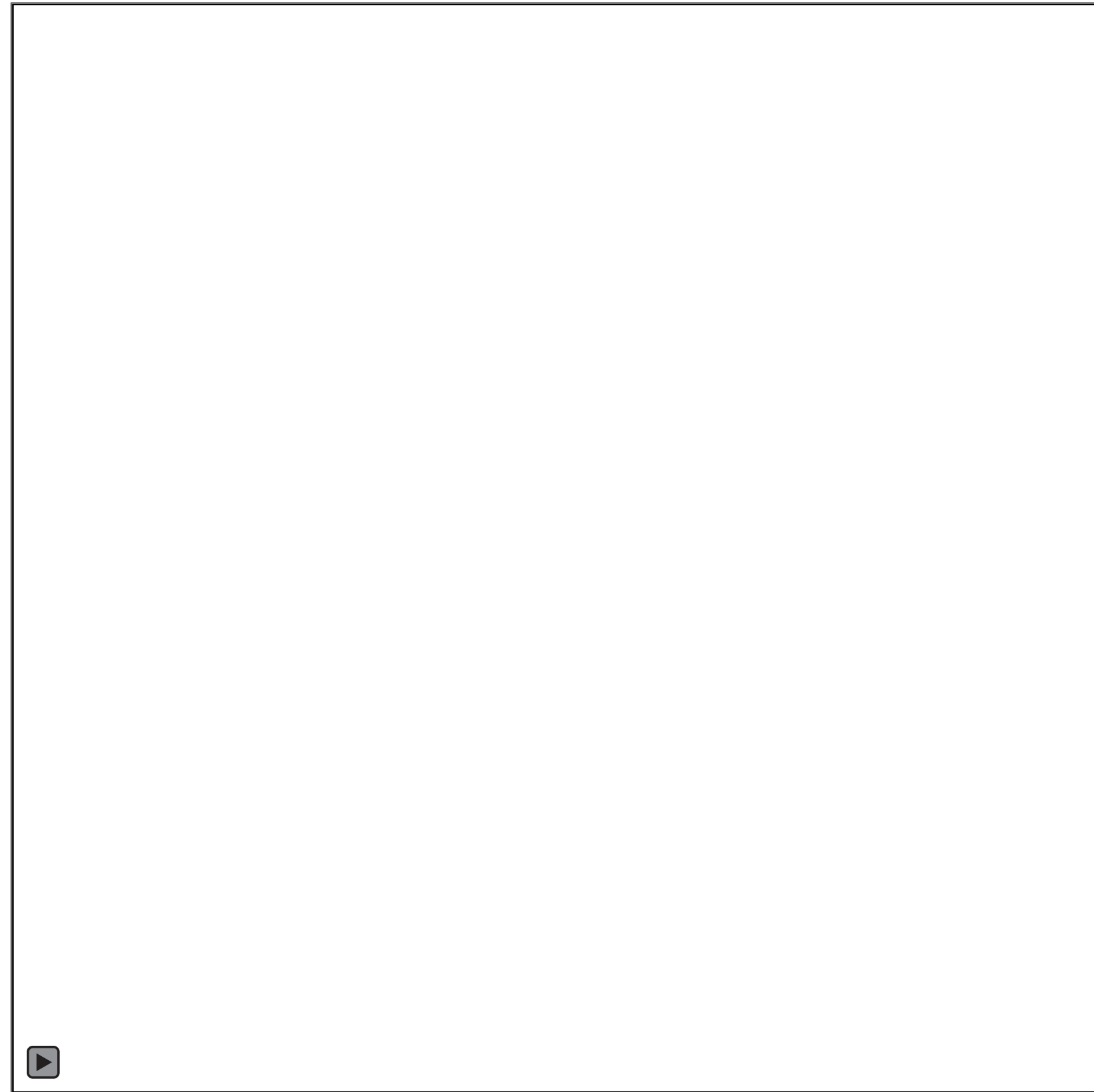
differentiable
rendering



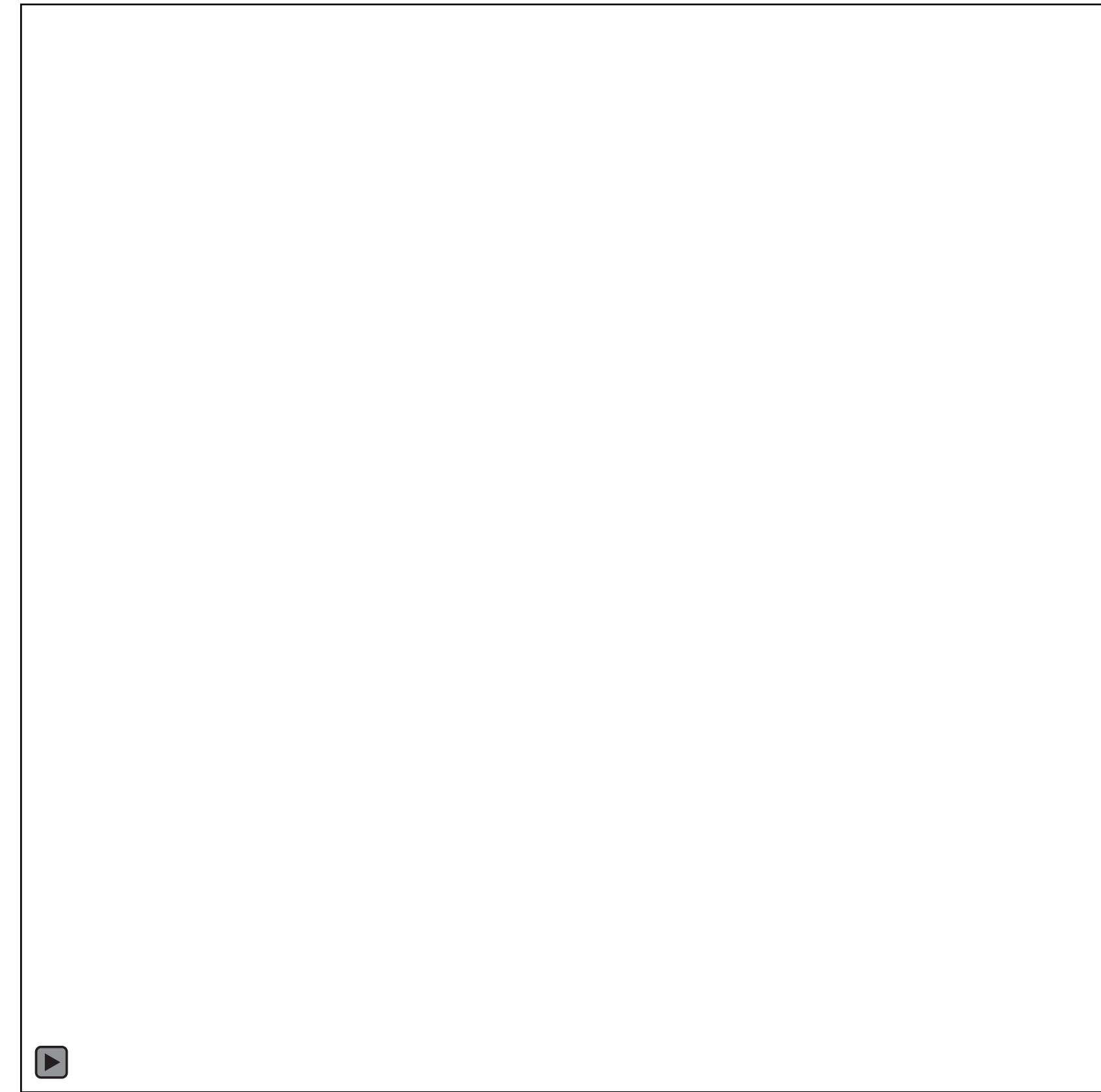
inverse
problems

Take-Home Messages

- Great progress has been made in physics-based rendering
 - Capable of handling **multiple types of imaging systems beyond RGB cameras (e.g., time-of-flight, sonar, tactile sensors)**.



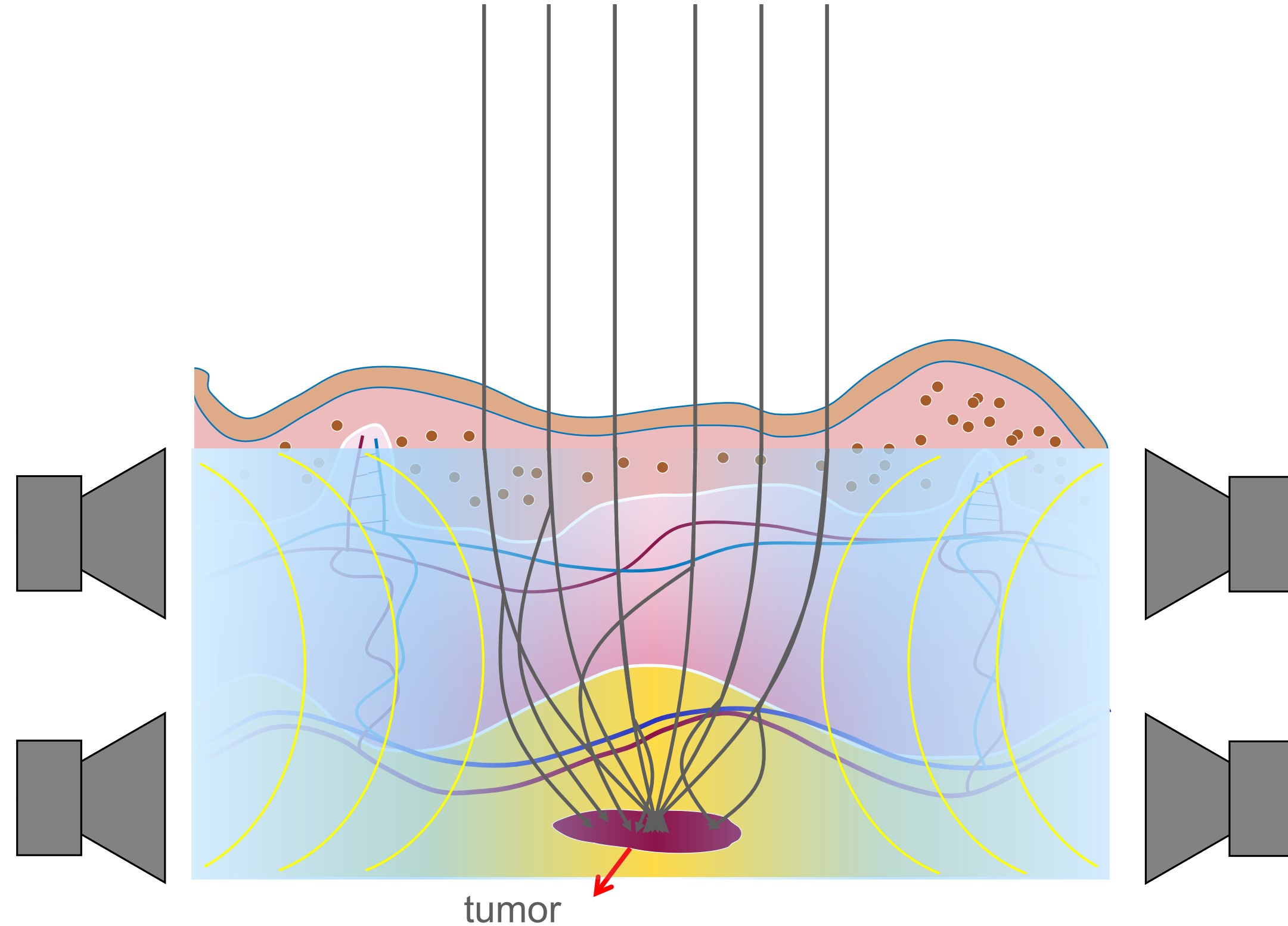
transient rendering of static scene



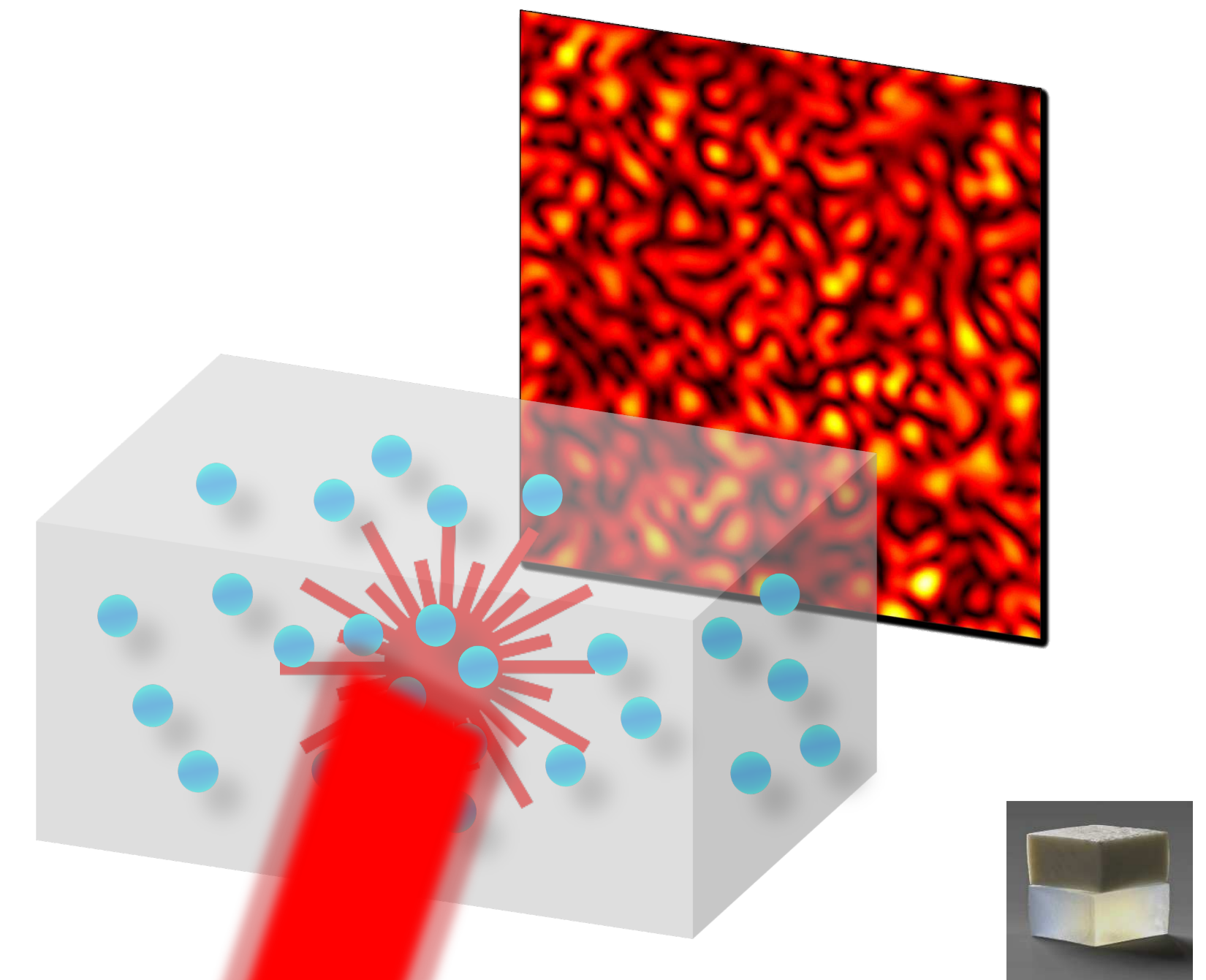
time-gated rendering of dynamic scene

Take-Home Messages

- Great progress has been made in physics-based rendering:
 - Capable of handling **multiple types of imaging systems beyond RGB cameras (e.g., time-of-flight, sonar, tactile sensors)**.
 - Capable of handling **more general scene models and light-matter interactions (e.g., speckle, continuous refraction and scattering)**.



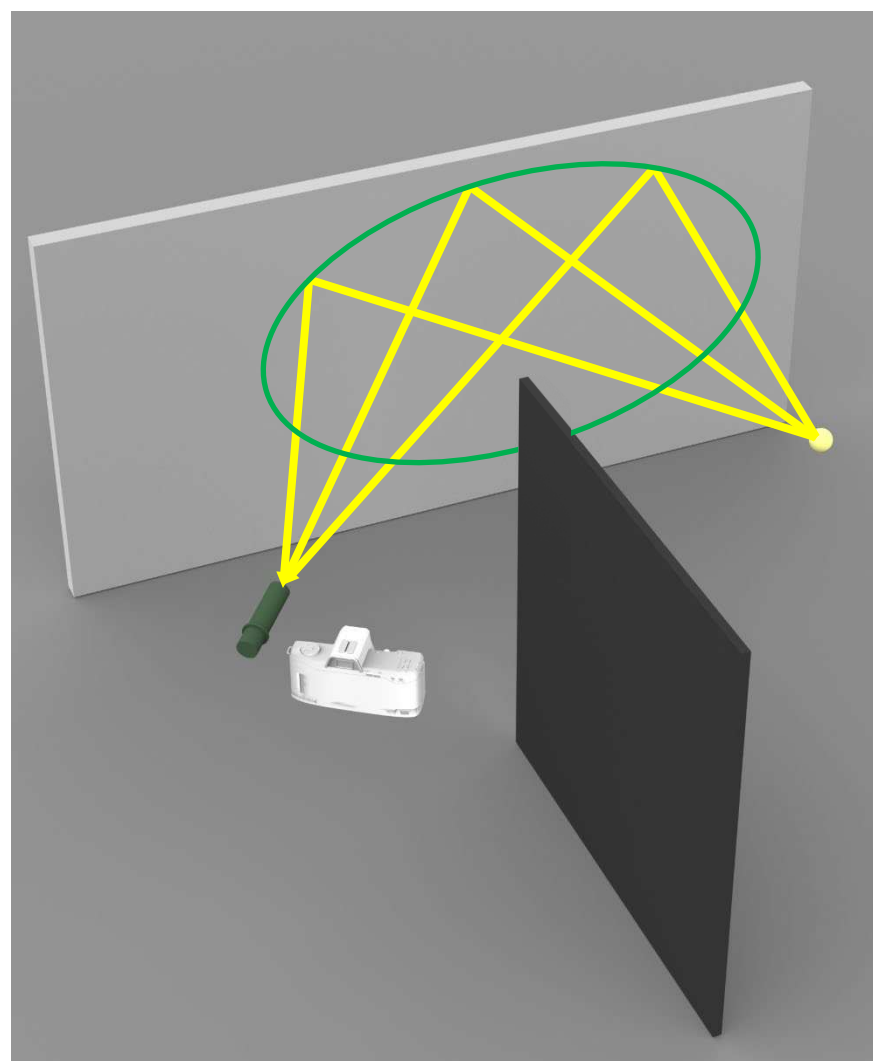
continuous refraction



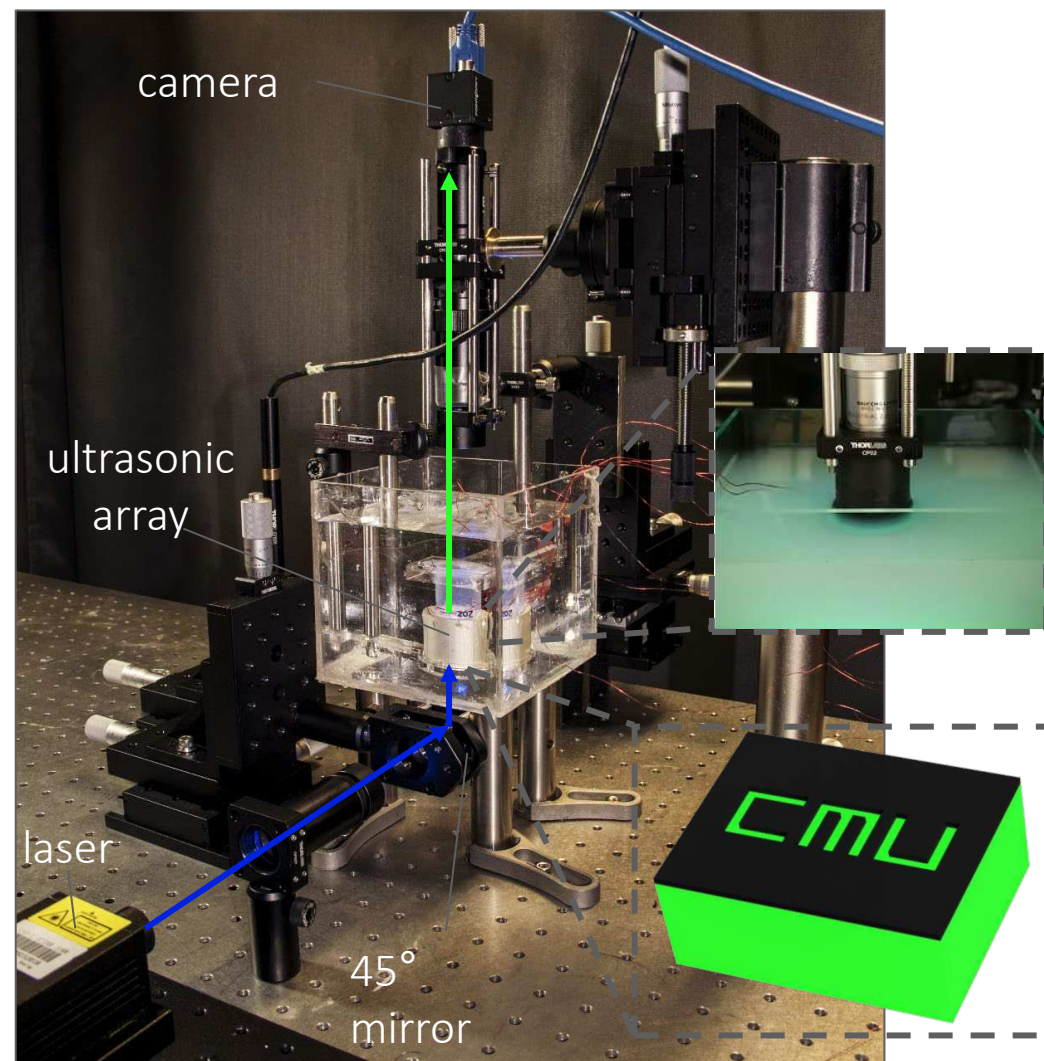
speckle-rendering

Take-Home Messages

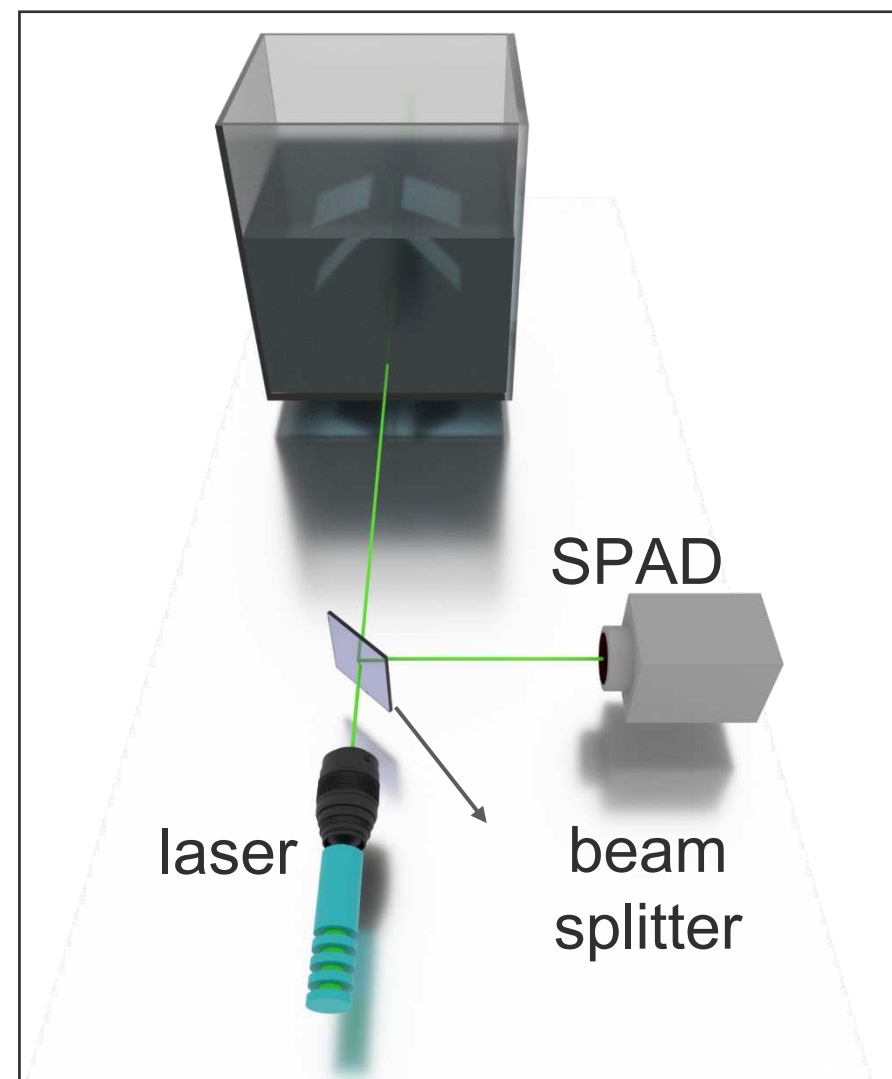
- Great progress has been made in physics-based rendering
 - Capable of handling **multiple types of imaging systems beyond RGB cameras (e.g., time-of-flight, sonar, tactile sensors)**.
 - Capable of handling **more general scene models and light-matter interactions (e.g., speckle, continuous refraction and scattering)**.
 - Capable of acting as **digital twins** for scientific imaging applications.



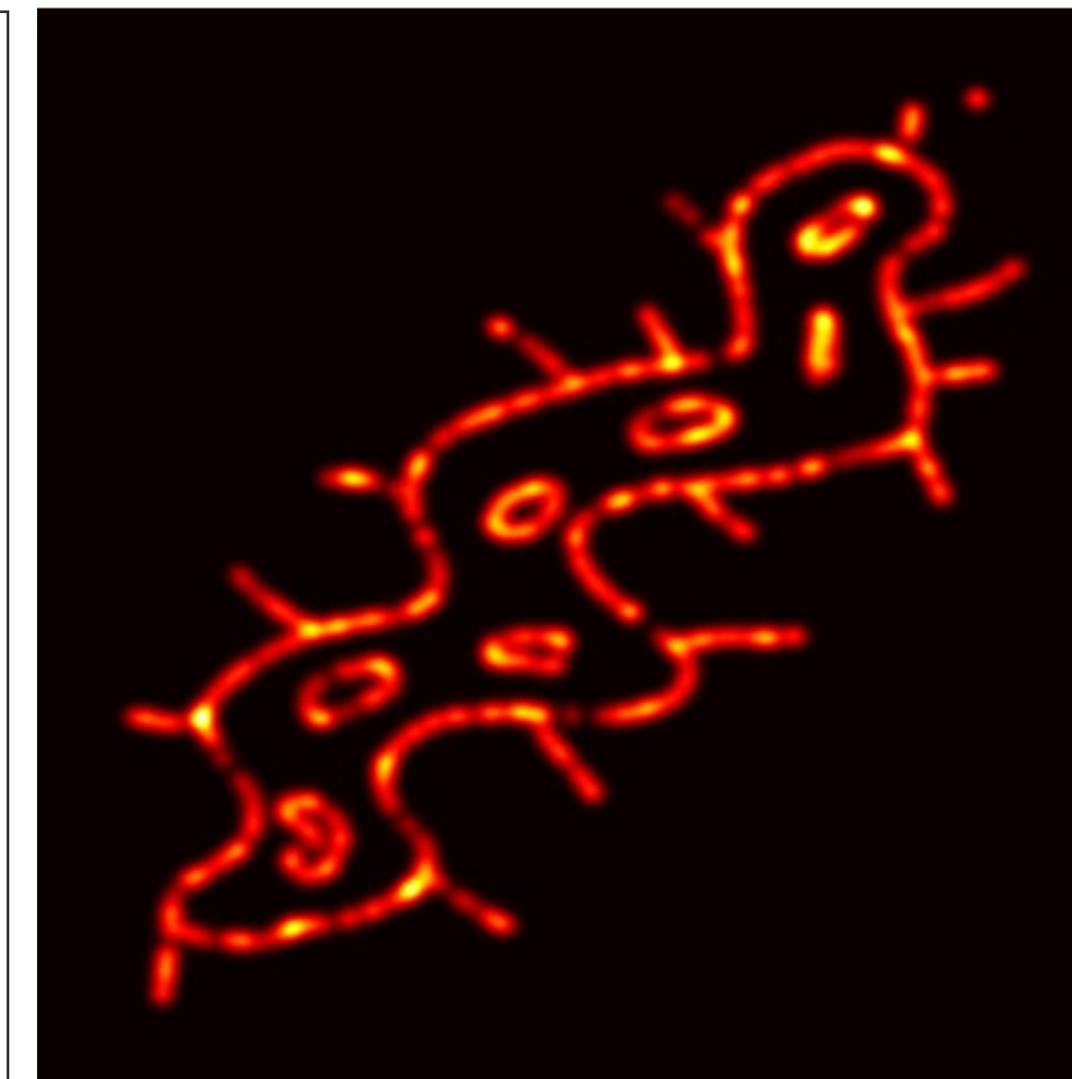
non-line-of-sight
imaging



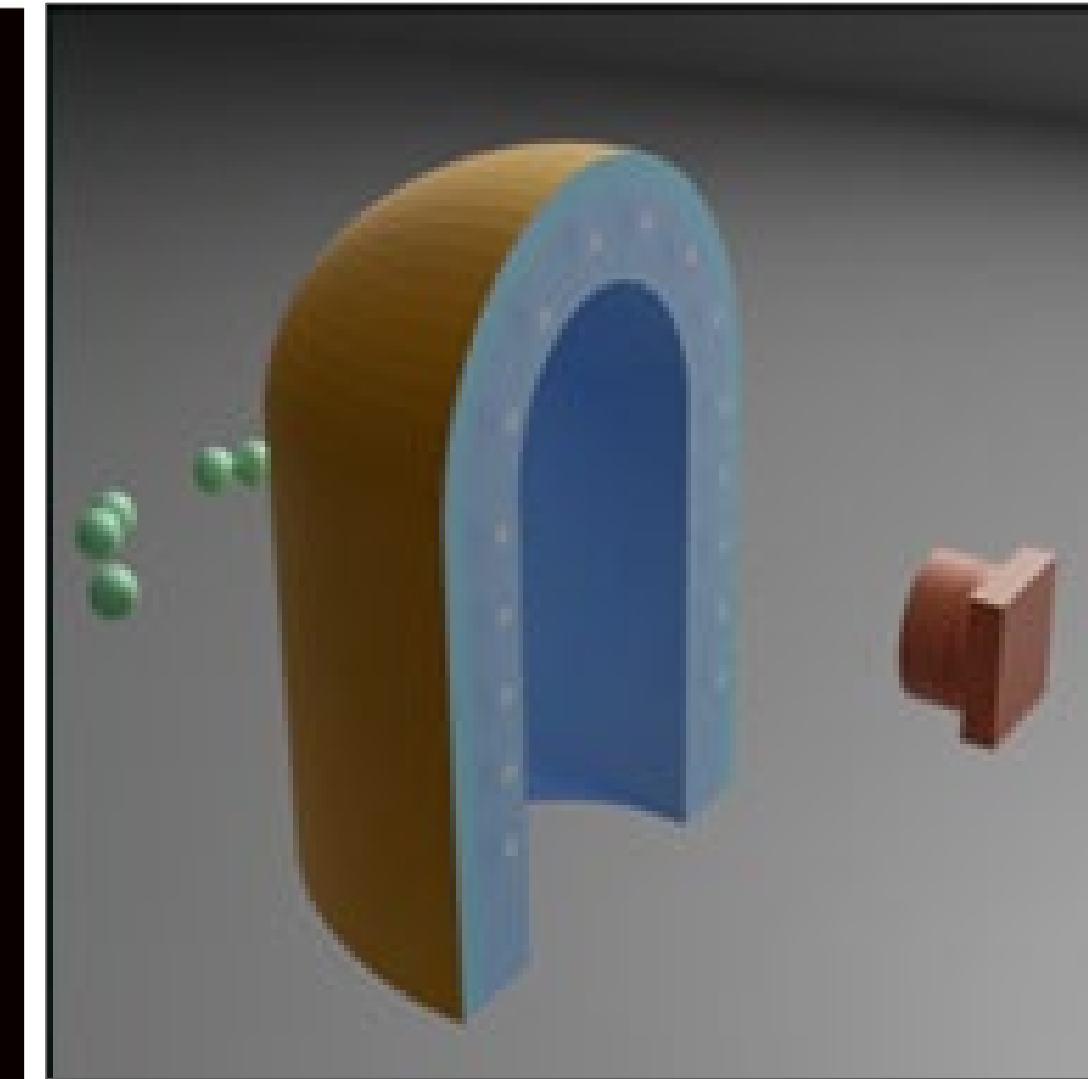
light throughput
enhancement



ultrafast
light scanners



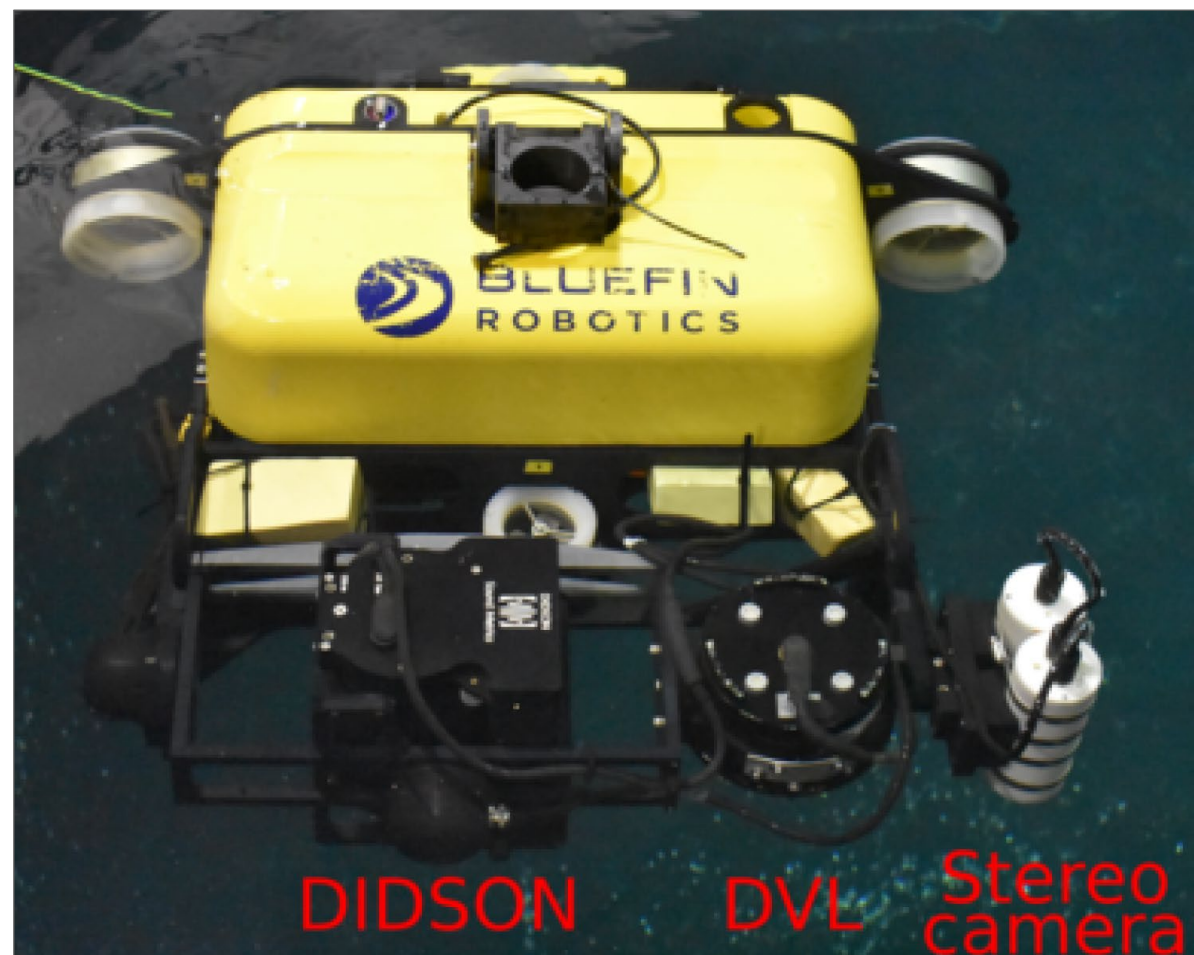
imaging through
scattering media



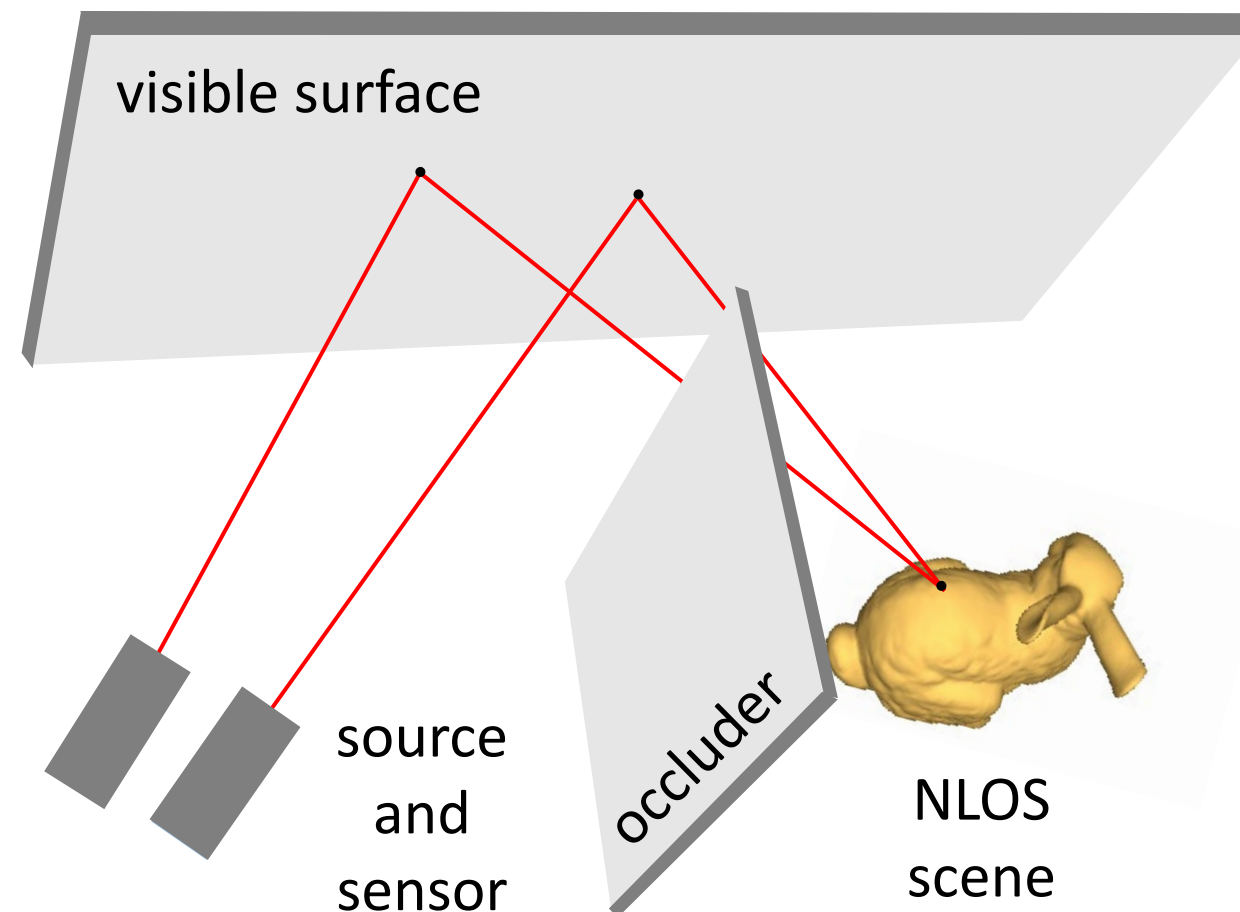
tactile
sensor design

Take-Home Messages

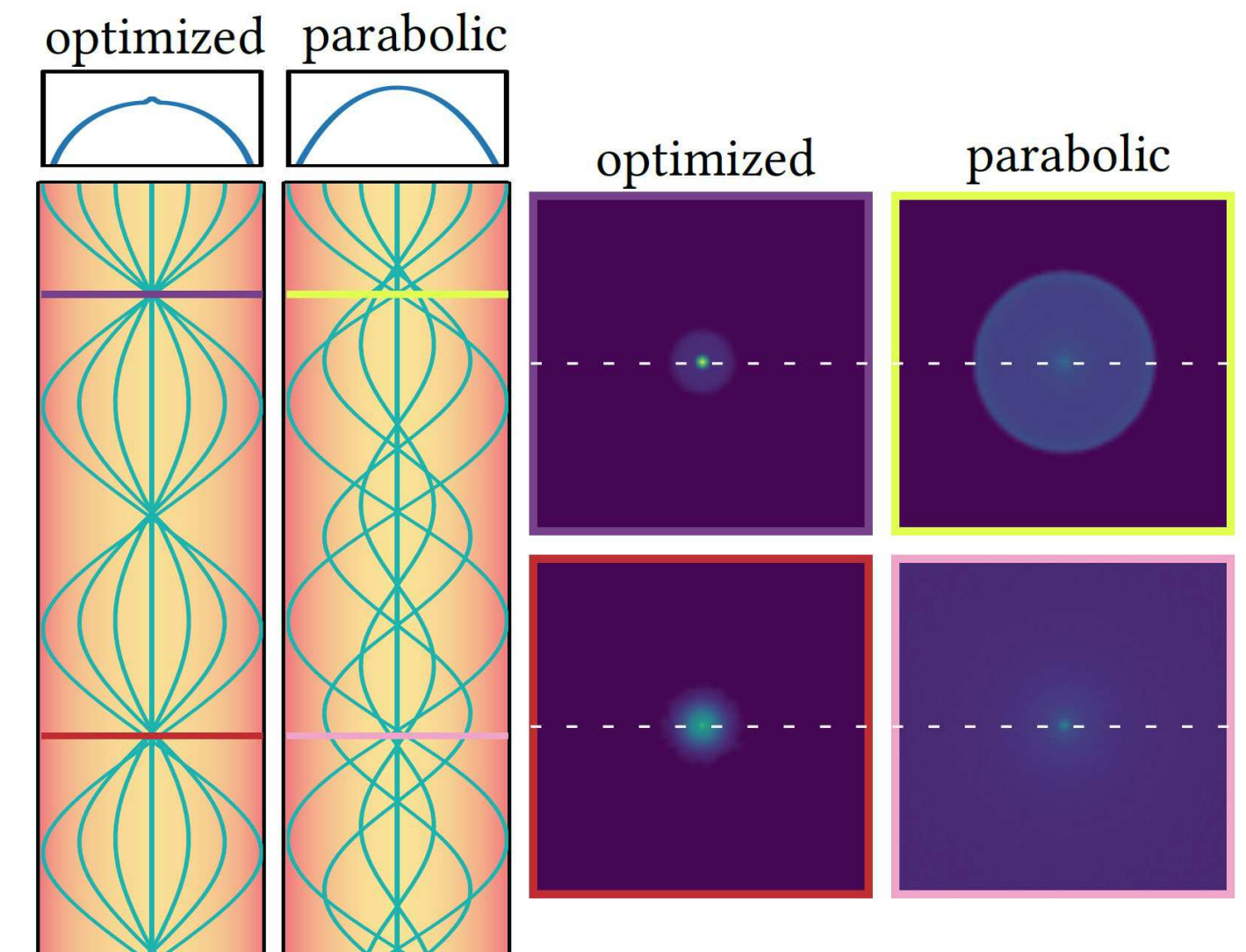
- Great progress has been made in physics-based rendering
 - Capable of handling **multiple types of imaging systems beyond RGB cameras** (e.g., time-of-flight, sonar, tactile sensors).
 - Capable of handling **more general scene models and light-matter interactions** (e.g., speckle, continuous refraction and scattering).
 - Capable of acting as **digital twins** for scientific imaging applications.
 - Capable of **differentiation** for general inverse rendering problems.



underwater
sonar

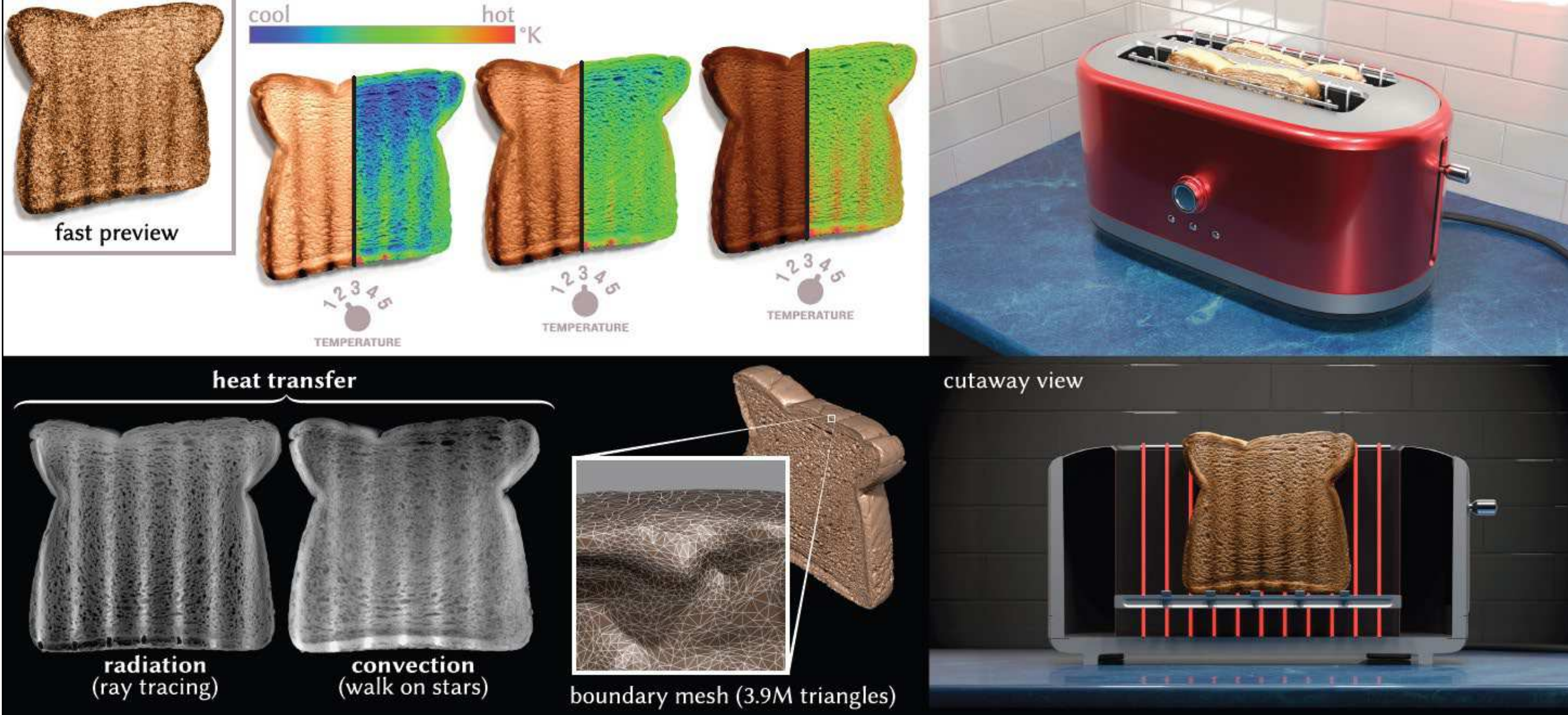


non-line-of-sight
imaging



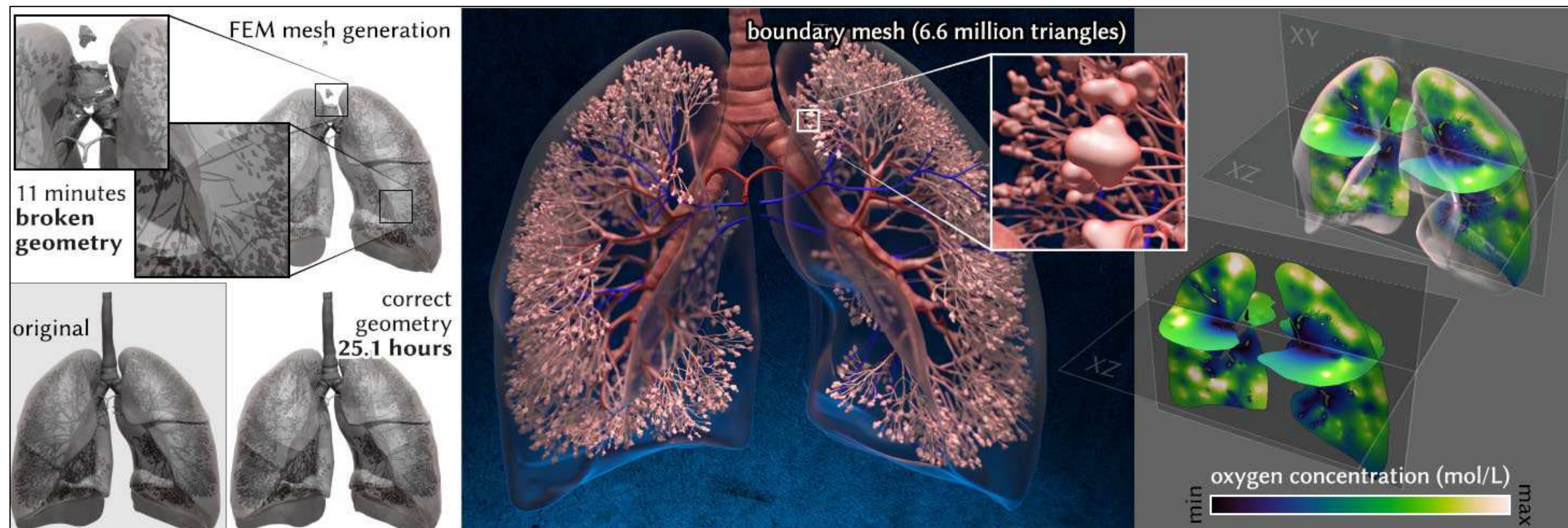
GRIN optic
design

Monte Carlo rendering for more general physics and sensing



Simulation of general diffusion processes like heat transfer and oxygen flow

Joint work with Rohan Sawhney, Bailey Miller, Keenan Crane SIGGRAPH 2023





Many thanks to our collaborators

Many thanks to our sponsors



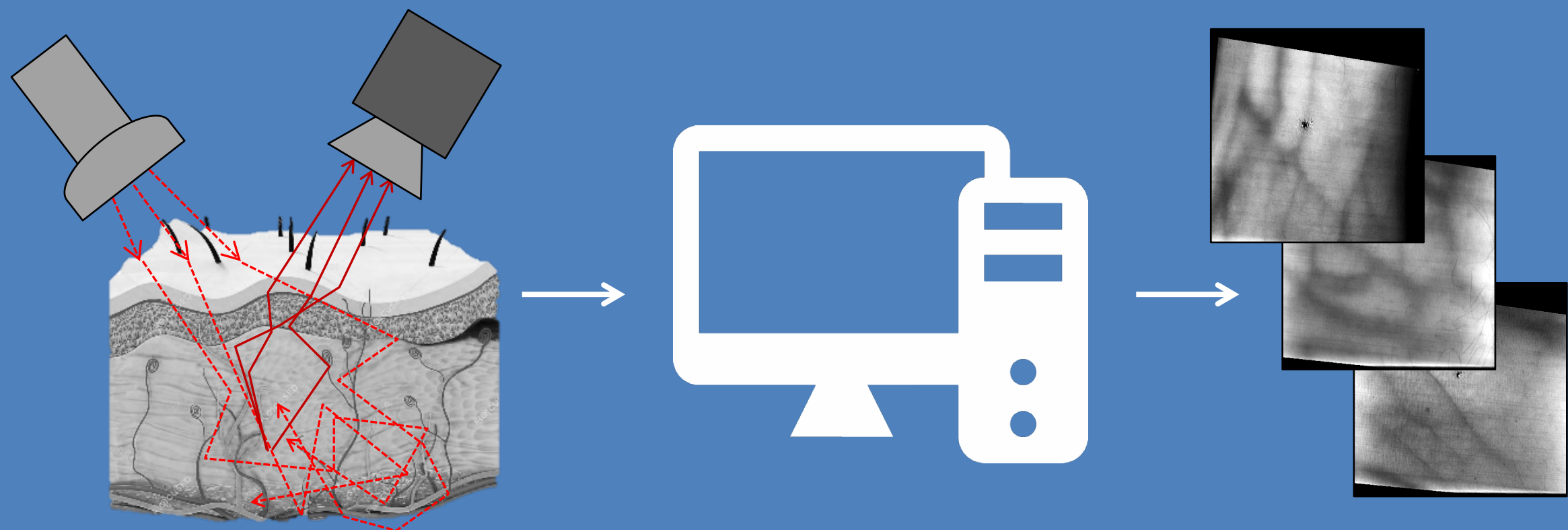
ALFRED P. SLOAN
FOUNDATION



SEE BELOW THE SKIN

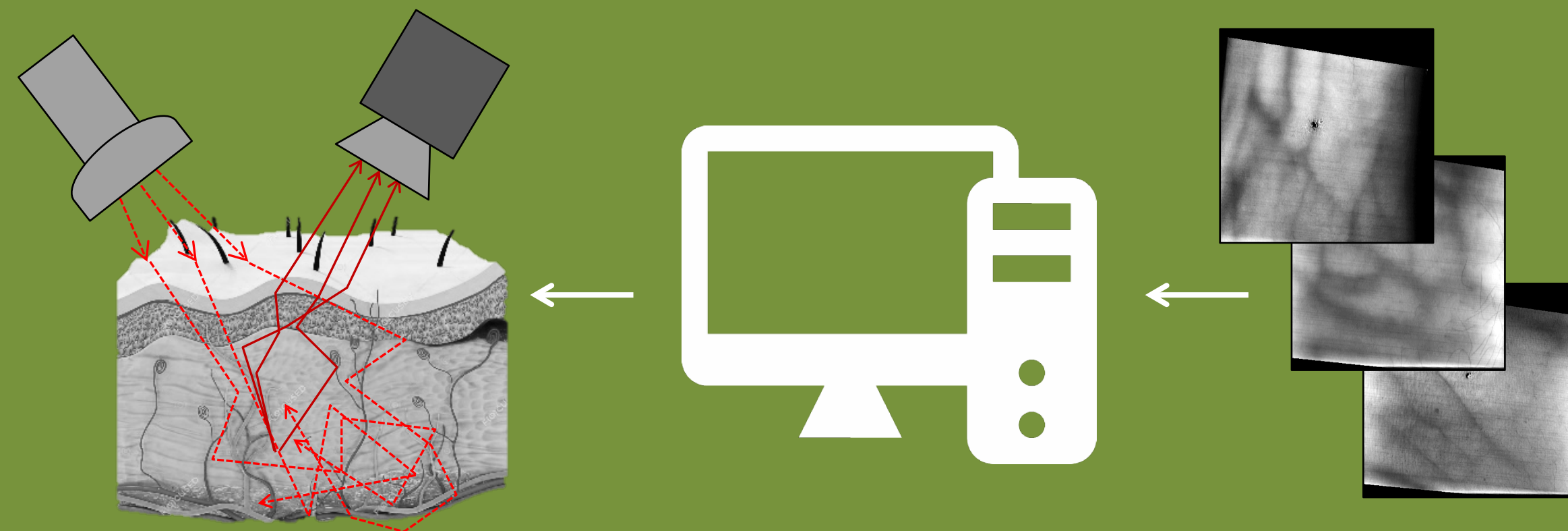
Physics-based rendering and its applications to computational imaging

forward rendering

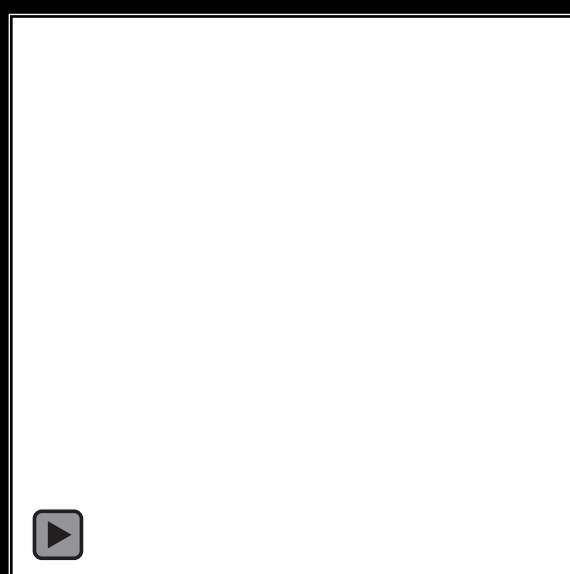


- accurate and efficient simulation
- virtually design sensors, optics, and algorithms

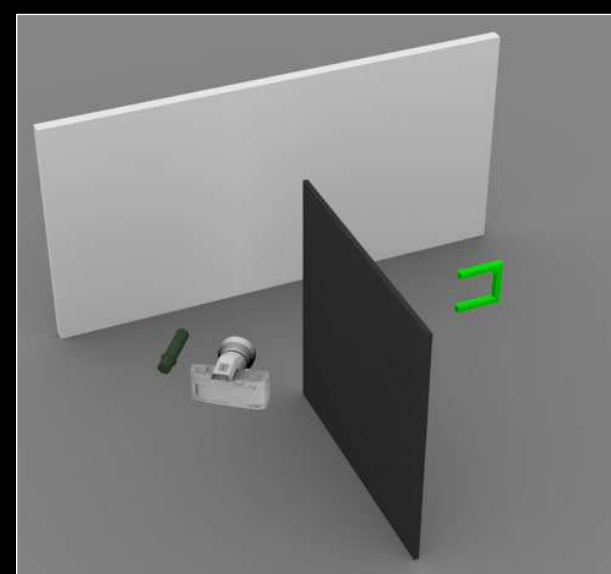
inverse rendering



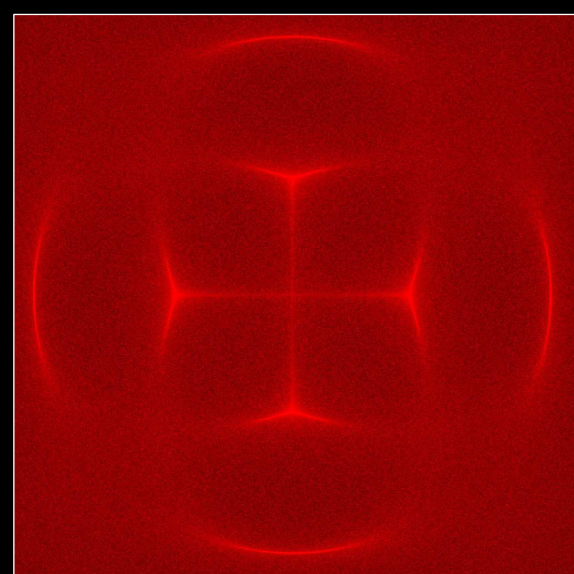
- accurate and efficient differentiable simulation
- tractably solve general inverse problems



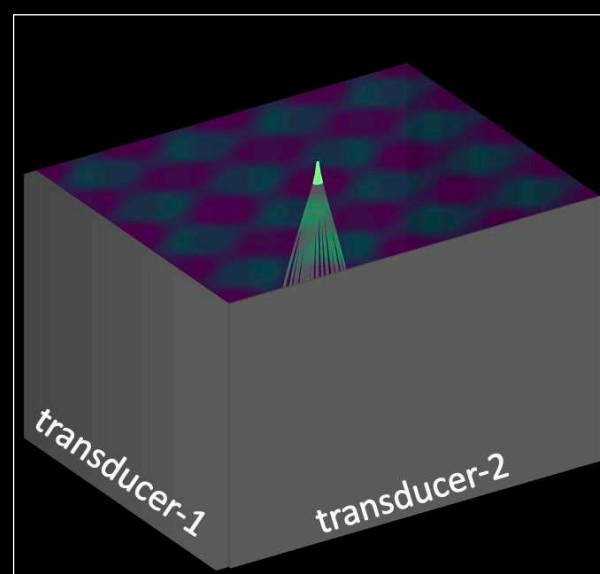
time-of-flight
imaging



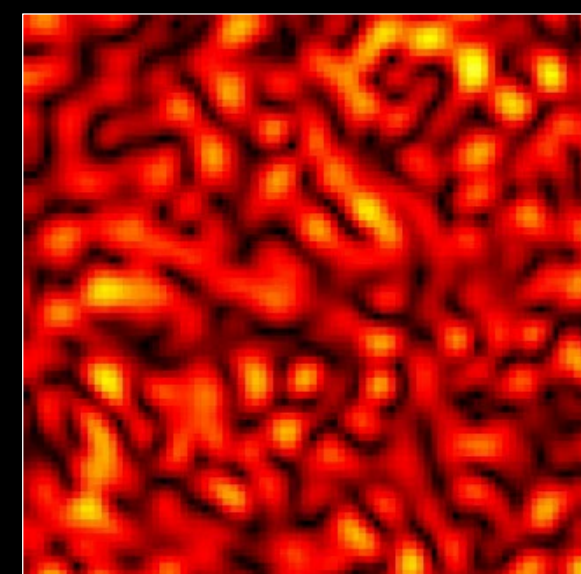
non-line-of-sight
imaging



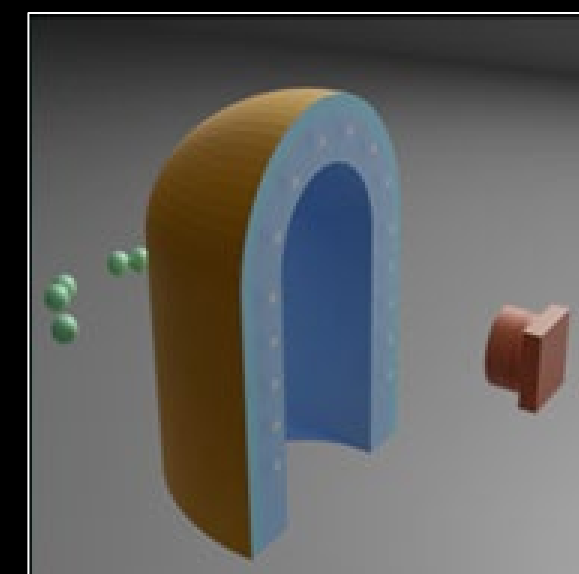
acousto-optic
lensing



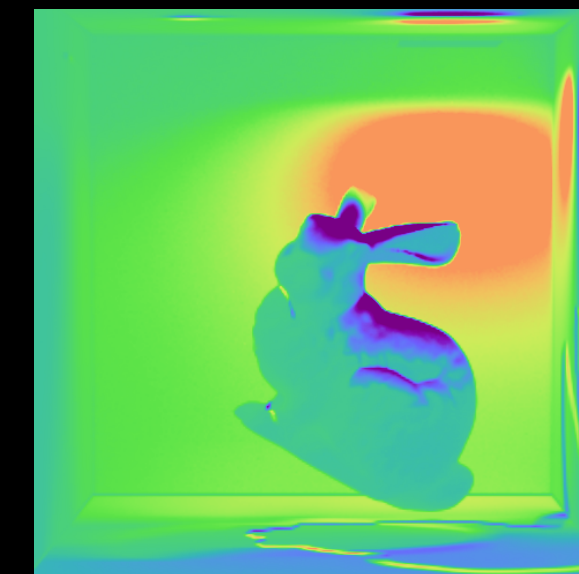
ultrafast light
scanning



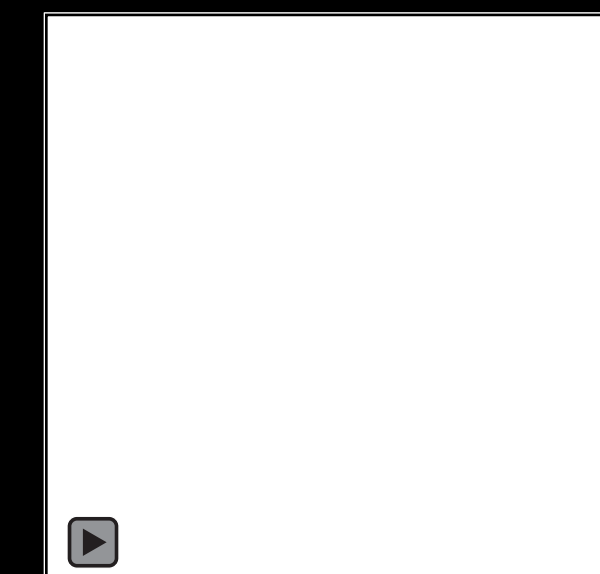
speckle
imaging



tactile sensor
design



differentiable
rendering



inverse
problems