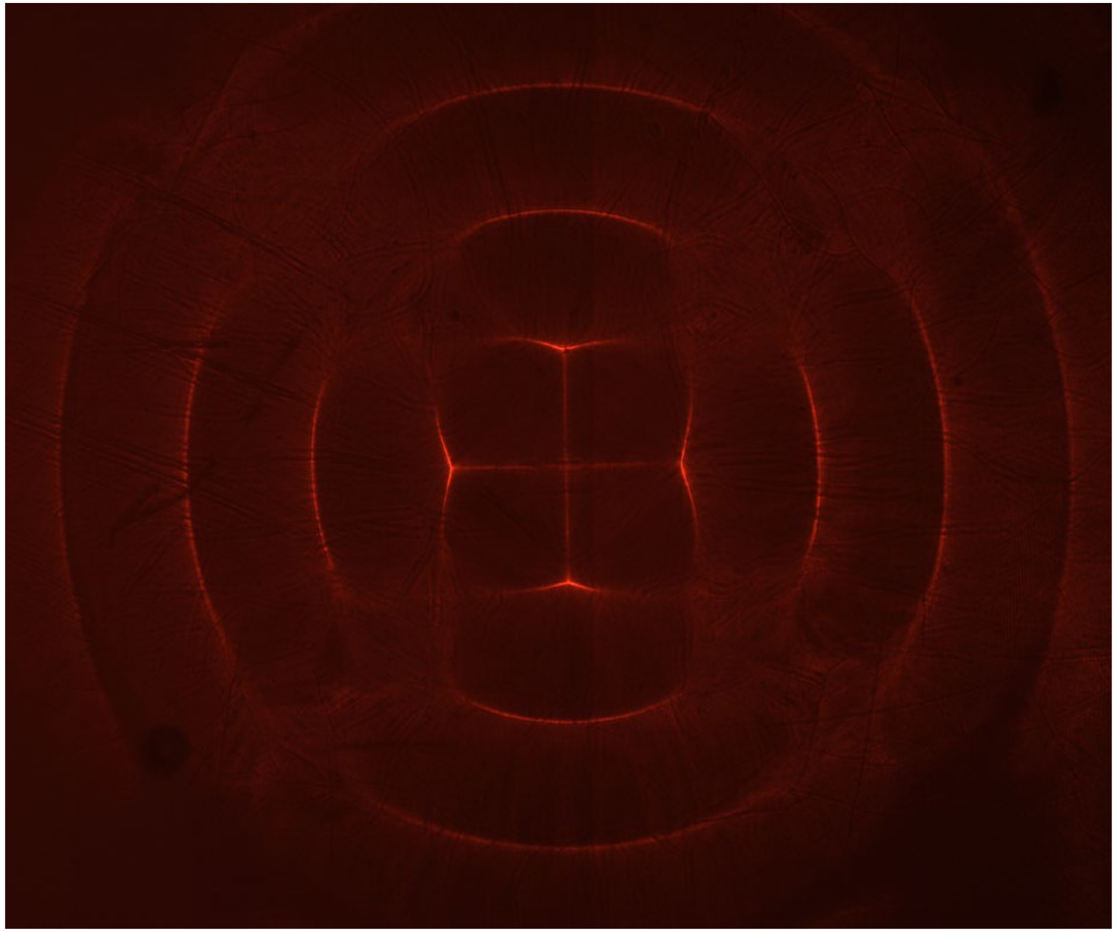
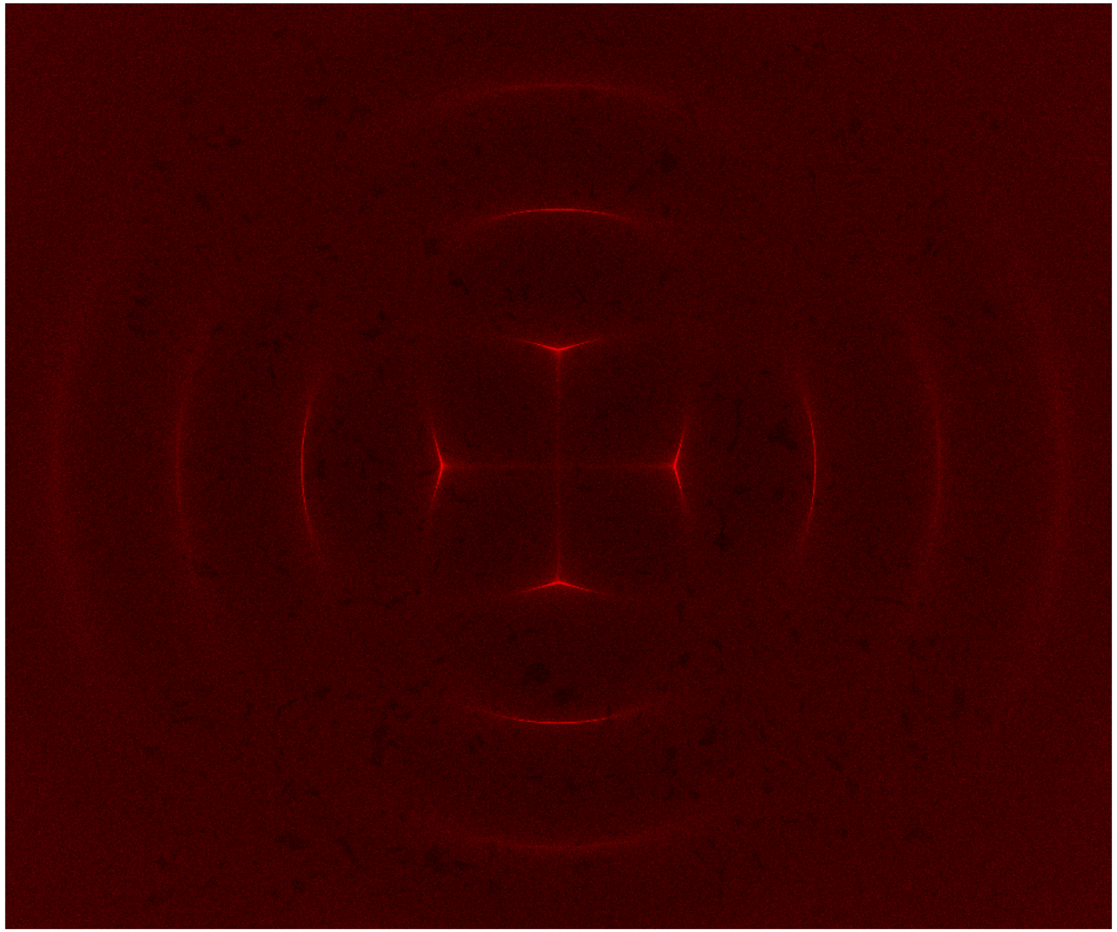
Rendering for scientific imaging applications





15-468, 15-668, 15-868
Physics-based Rendering
Spring 2025, Lecture 16

https://graphics.cs.cmu.edu/courses/15-468

Course announcements

- We’re all done with homework!

Overview of today’s lecture

- Rendering continuous refraction.
- GRIN optics.
- Rendering the refractive radiative transfer equation.
- Acousto-optics.
- Rendering speckle.
- Fluorescence microscopy.

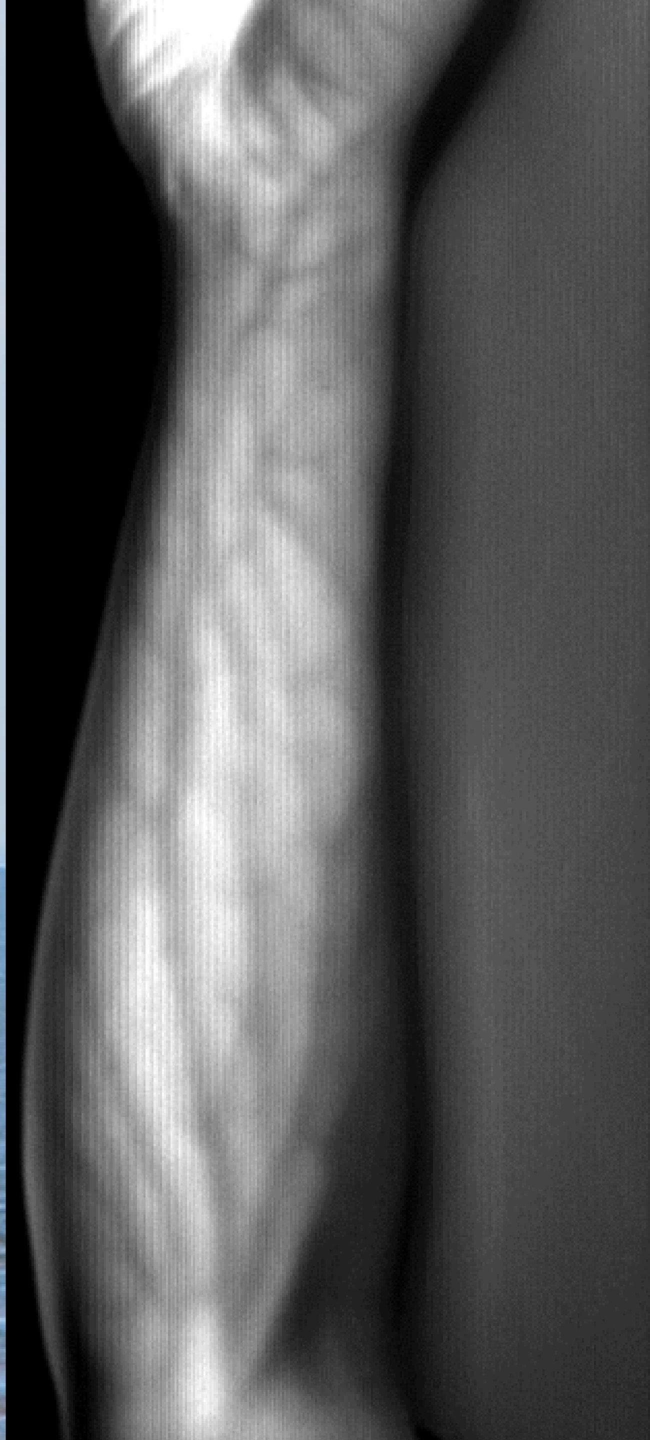
Slide credits

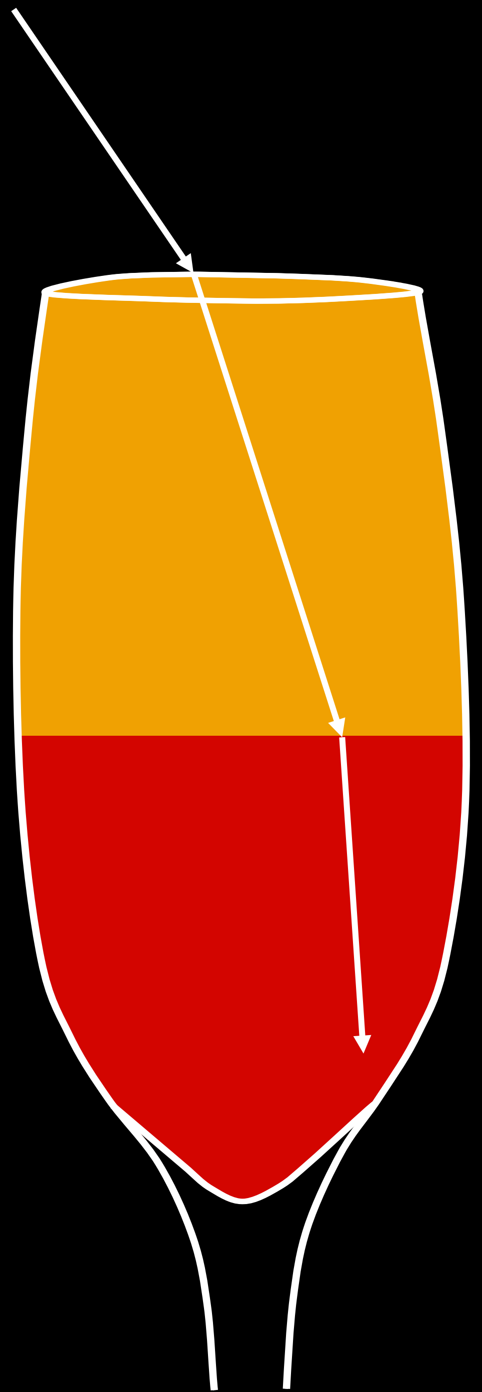
Many of these slides were directly adapted from:

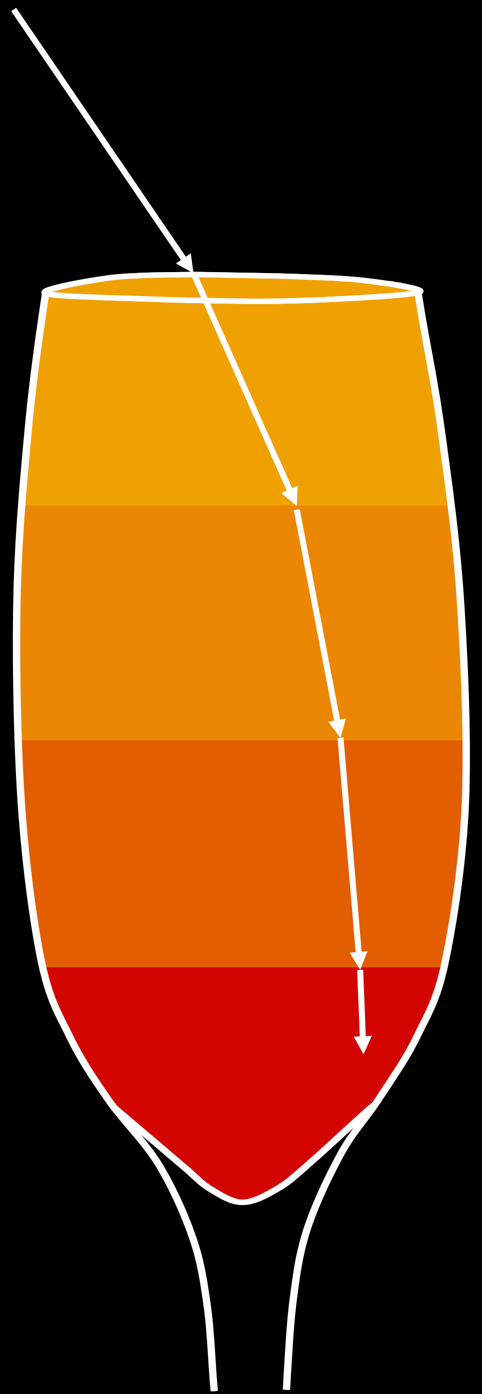
- Adithya Pediredla (CMU).
- Arjun Teh (CMU).
- Chen Bar (Technion).

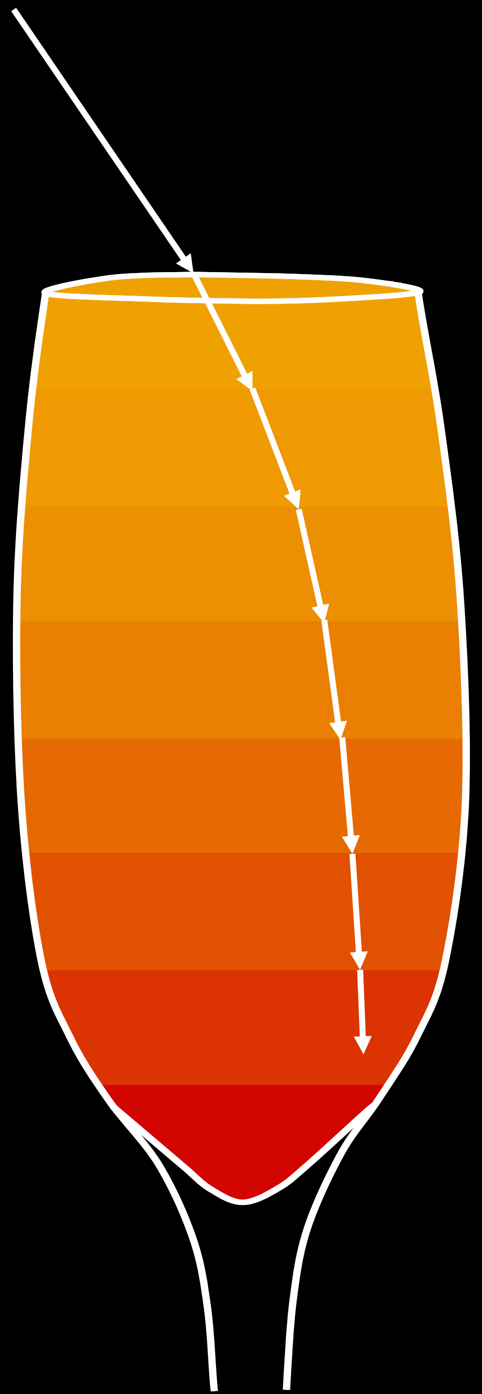
Media with continuously varying refractive index and scattering

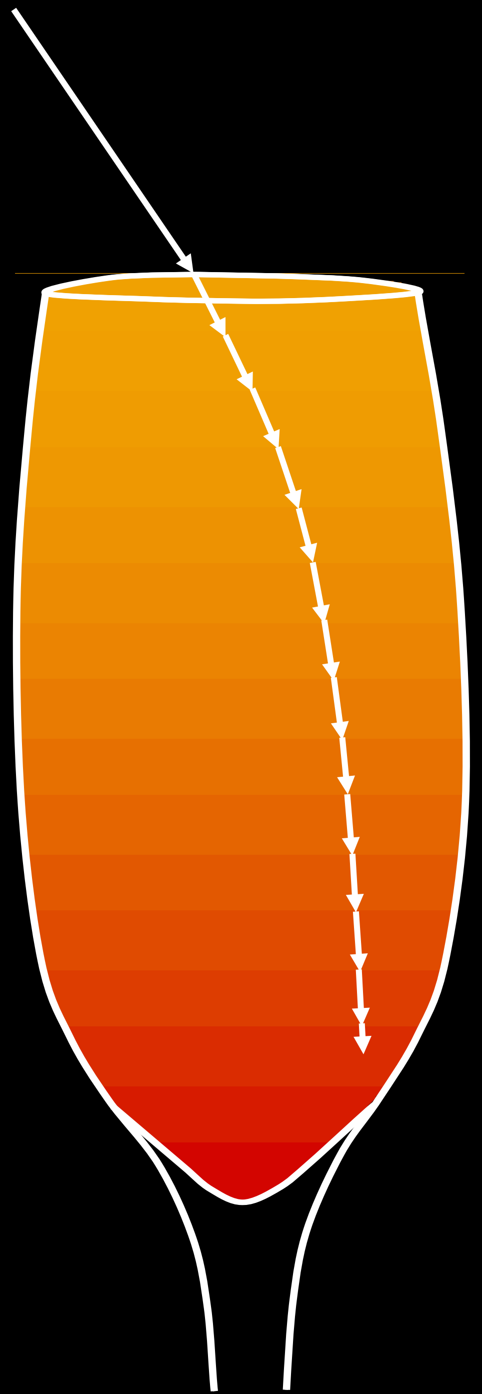


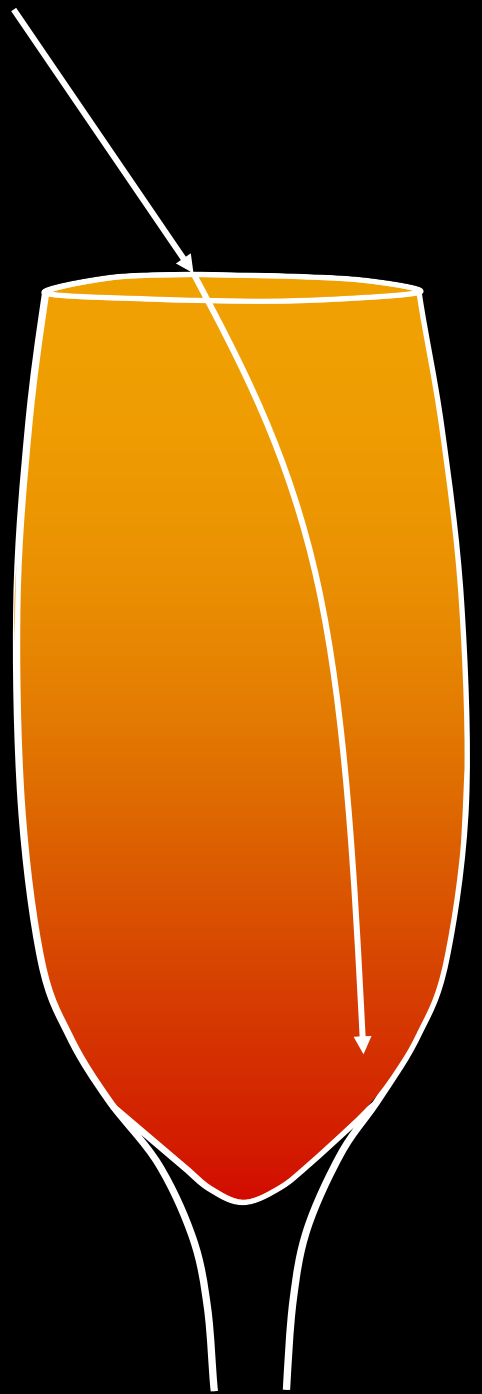









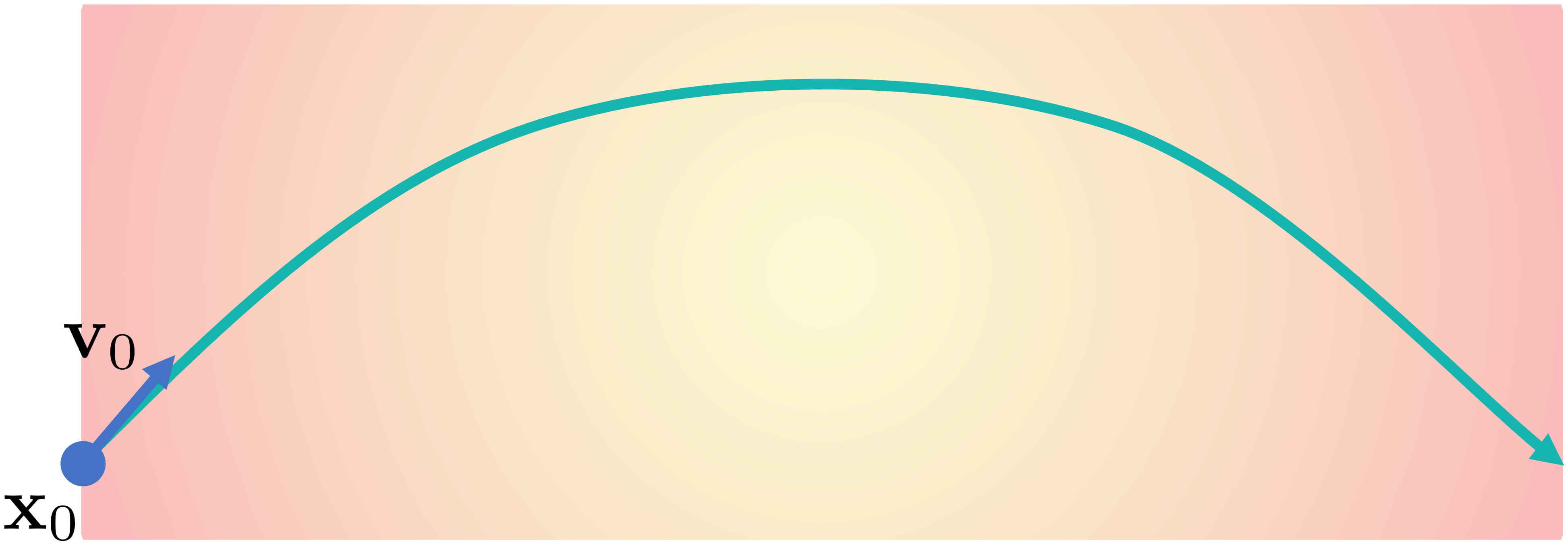




Nonlinear Ray Tracing

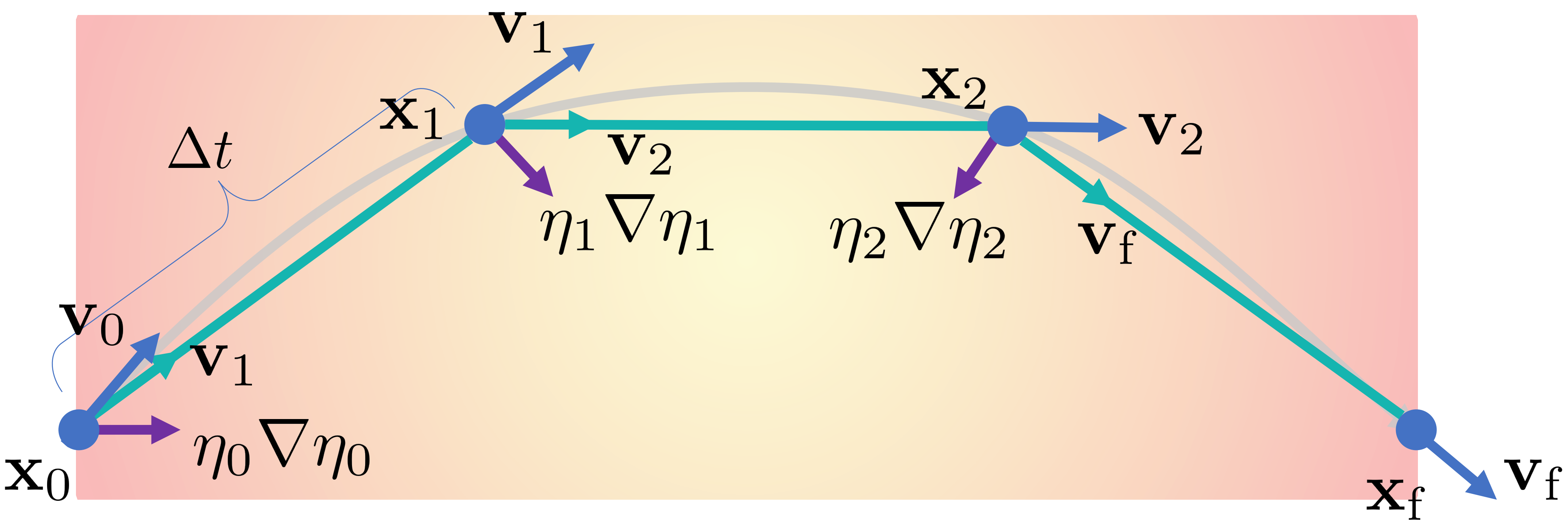
$\eta(\mathbf{x})$: refractive index of the volume at location, $\mathbf{x}$

Nonlinear Ray Tracing



$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v} \qquad \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \eta \nabla \eta$$

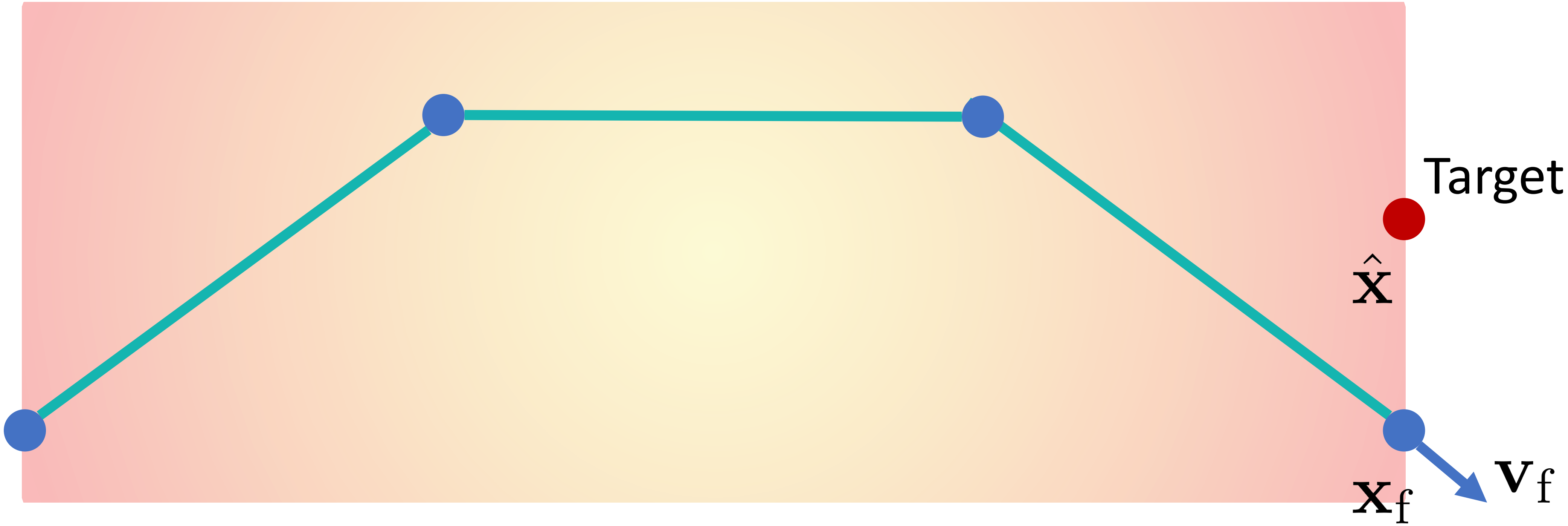
Nonlinear Ray Tracing



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

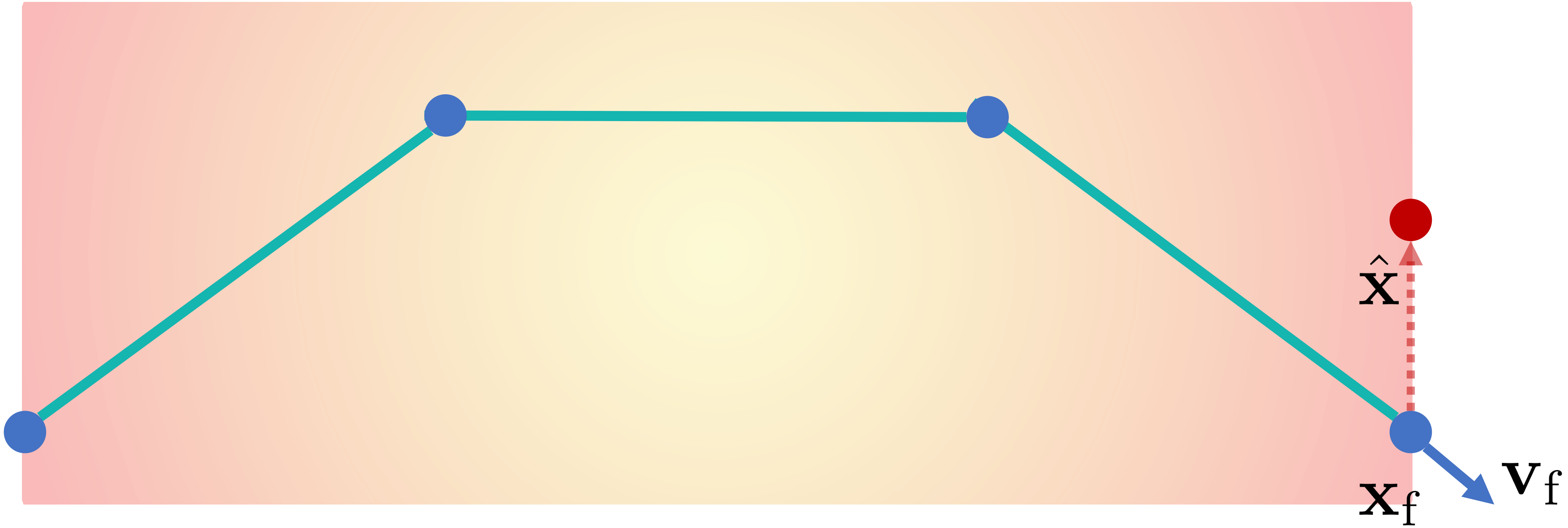
$$\mathbf{v}_i = \mathbf{v}_{i-1} + \eta_{i-1} \nabla \eta_{i-1} \Delta t$$

Nonlinear Ray Tracing



$$\min_{\eta} \|\hat{\mathbf{x}} - \mathbf{x}_\mathrm{f}\|^2$$

Nonlinear Ray Tracing



$$\frac{\mathrm{d}}{\mathrm{d}\eta}\|\hat{\mathbf{x}} - \mathbf{x}_\mathrm{f}\|^2$$

Nonlinear Ray Tracing in reverse

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

$$\begin{aligned}
\text{s.t. } \dot{\mathbf{x}}\left(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0\right) &= \mathbf{v}, \quad \forall \sigma \in \left[0, \sigma_f\right], \\
\dot{\mathbf{v}}\left(\sigma; \eta, \mathbf{x}_0, \mathbf{v}_0\right) &= \eta \nabla \eta, \quad \forall \sigma \in \left[0, \sigma_f\right], \\
\mathbf{x}\left(0; \eta, \mathbf{x}_0, \mathbf{v}_0\right) &= \mathbf{x}_0, \\
\mathbf{v}\left(0; \eta, \mathbf{x}_0, \mathbf{v}_0\right) &= \mathbf{v}_0,
\end{aligned} \tag{15}$$

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

$$\dot{\boldsymbol{\mu}} = -\boldsymbol{\lambda}, \quad \forall \sigma \in \left[0, \sigma_f\right] \tag{20}$$

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

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

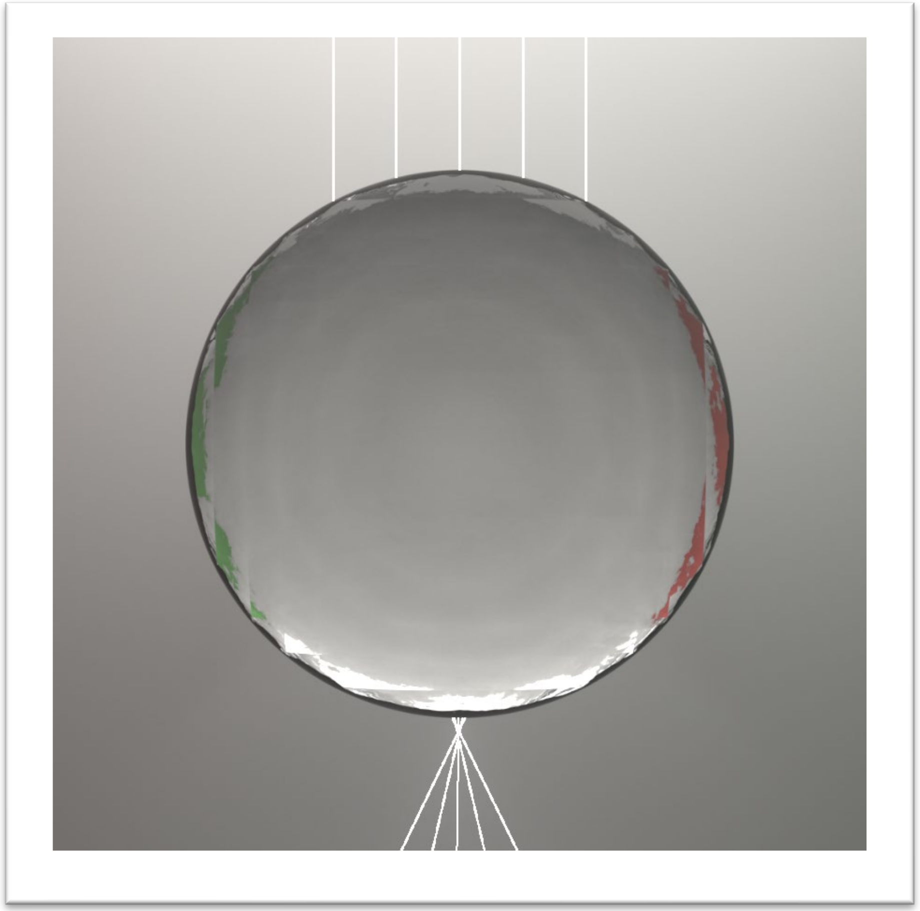
$$\mathbf{x}_{i-1} = \mathbf{x}_i - \mathbf{v}_i \Delta\sigma, \tag{28}$$

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

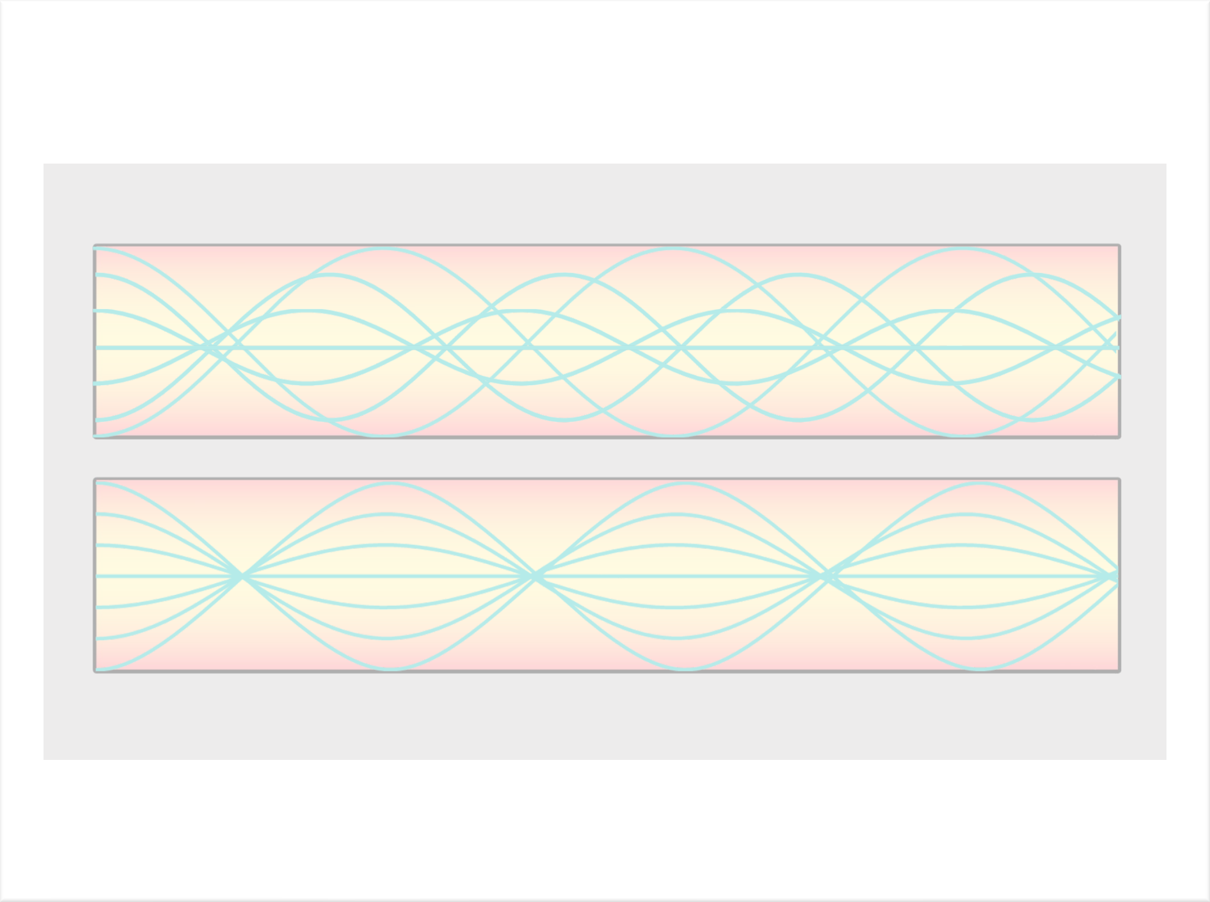
$$\begin{aligned}
\boldsymbol{\lambda}_{i-1} &= \boldsymbol{\lambda}_i \\
&\quad + \left(\nabla\eta\left(\mathbf{x}_{i-1}\right)\left(\nabla\eta\left(\mathbf{x}_{i-1}\right)\right)^{\top} + \eta\left(\mathbf{x}_{i-1}\right)\mathrm{Hess}\left(\eta\left(\mathbf{x}_{i-1}\right)\right)\right)\boldsymbol{\mu}_i \Delta\sigma,
\end{aligned} \tag{30}$$

$$\boldsymbol{\mu}_{i-1} = \boldsymbol{\mu}_i + \boldsymbol{\lambda}_{i-1}\Delta\sigma. \tag{31}$$

Optimizing Gradient-Index (GRIN) Optics

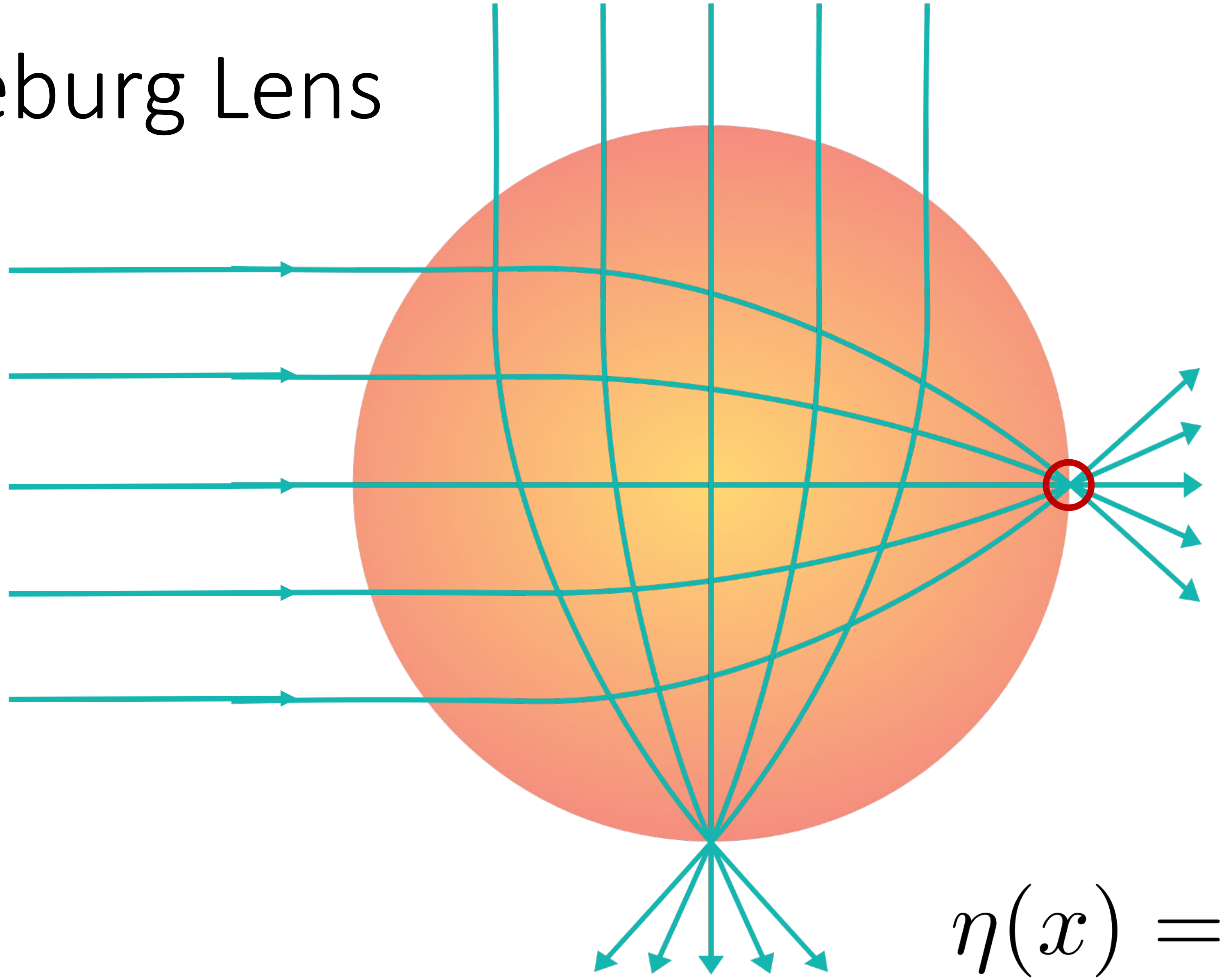


Luneburg Lens



GRIN Fiber

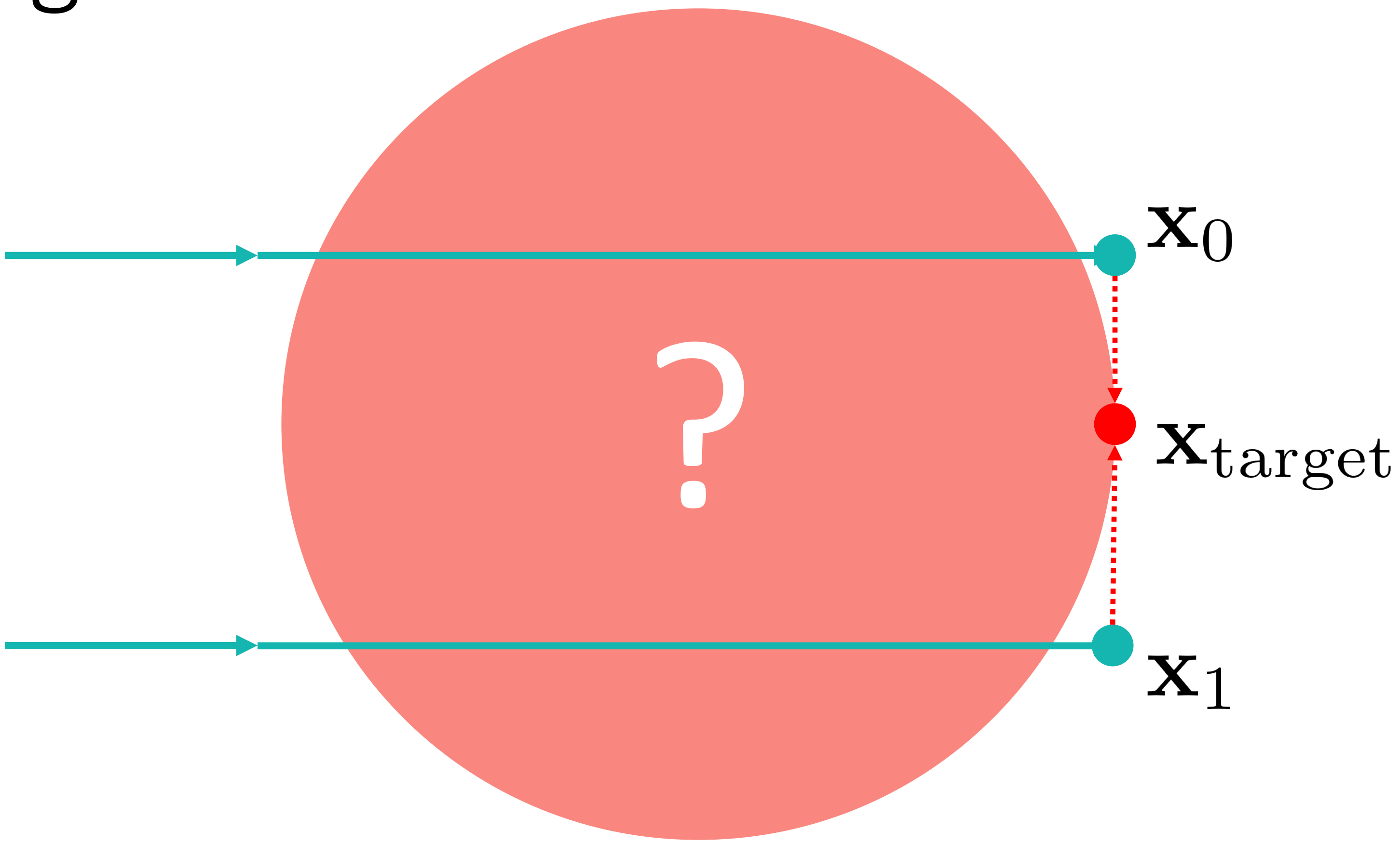
Luneburg Lens



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

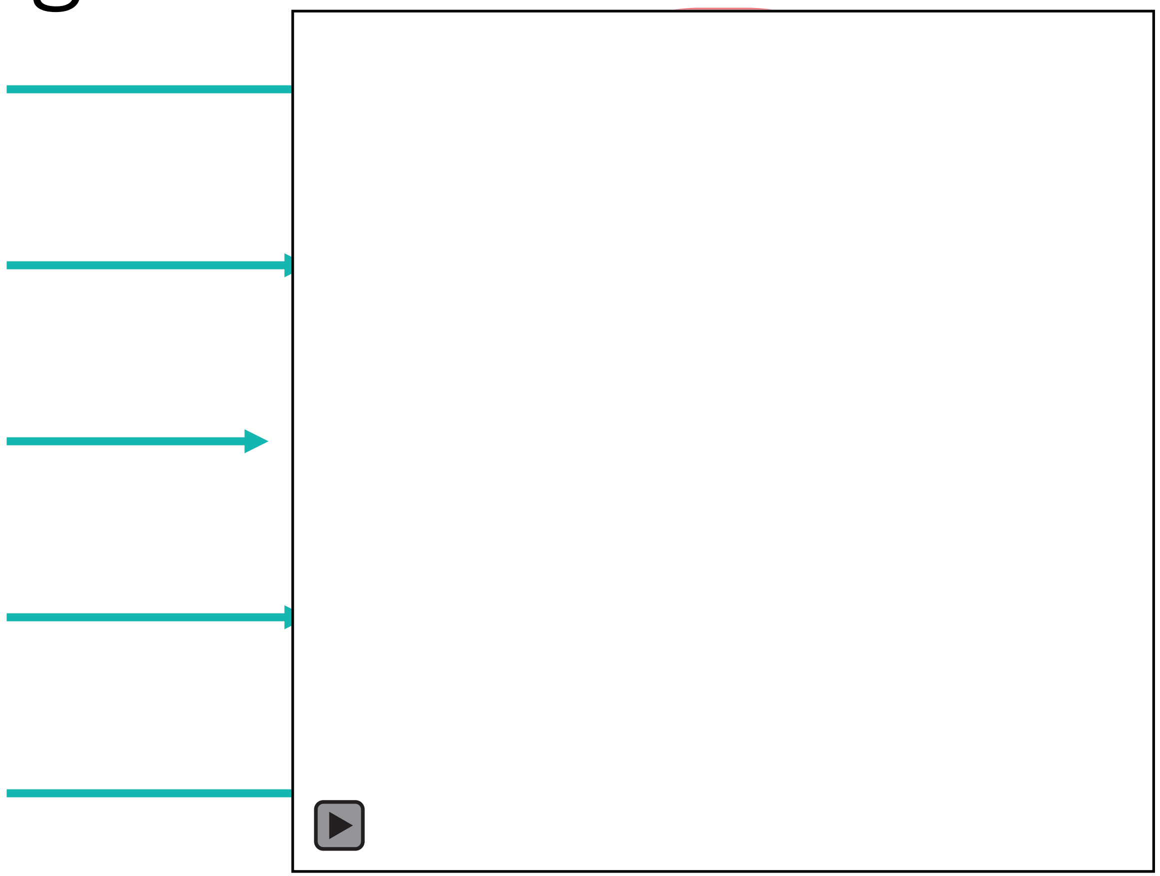
[Luneburg, R. K. 1944]

Luneburg Lens

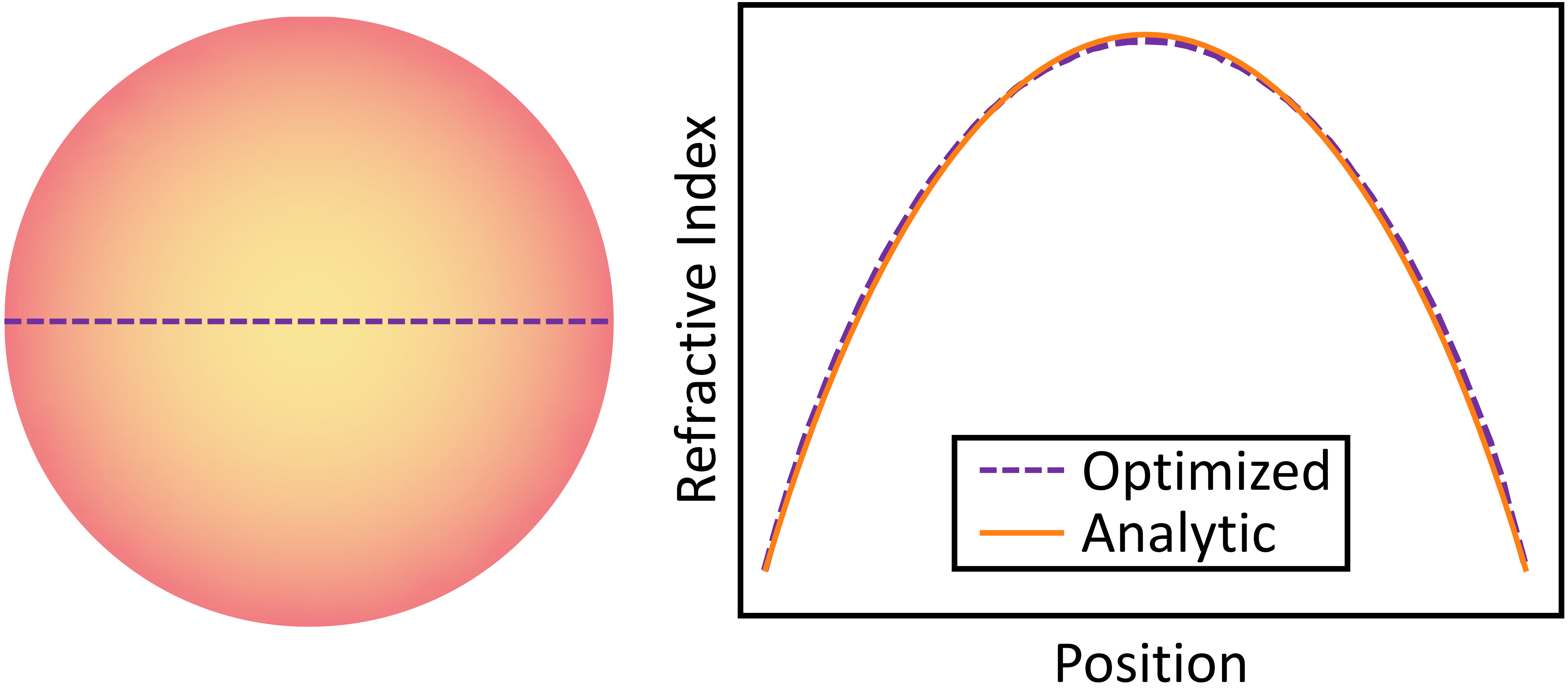


$$\min_{\eta} \sum_{i \in \mathrm{rays}} \left\| \mathbf{x}_{\mathrm{target}} - \mathbf{x}_{\mathrm{i}} \right\|^2$$

Luneburg Lens

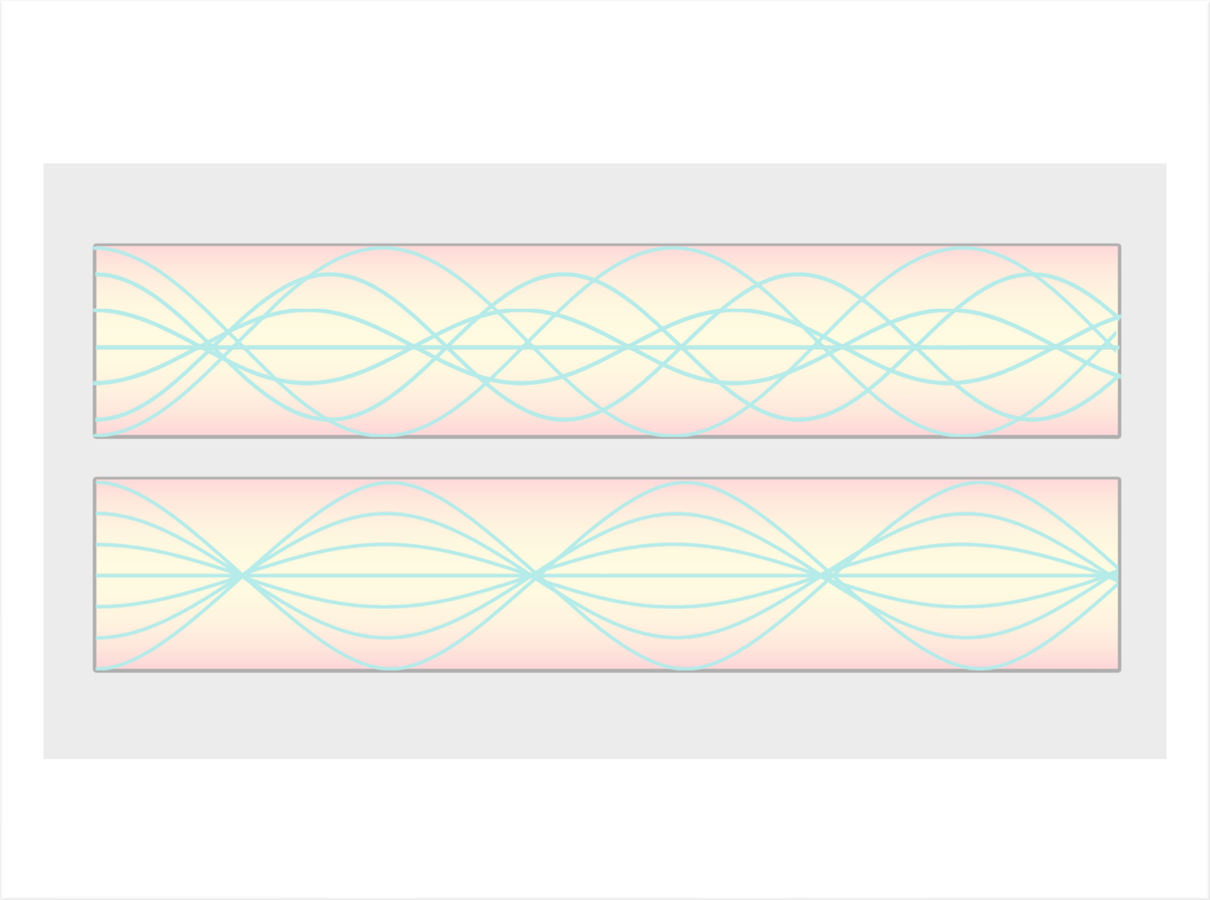


Luneburg Lens



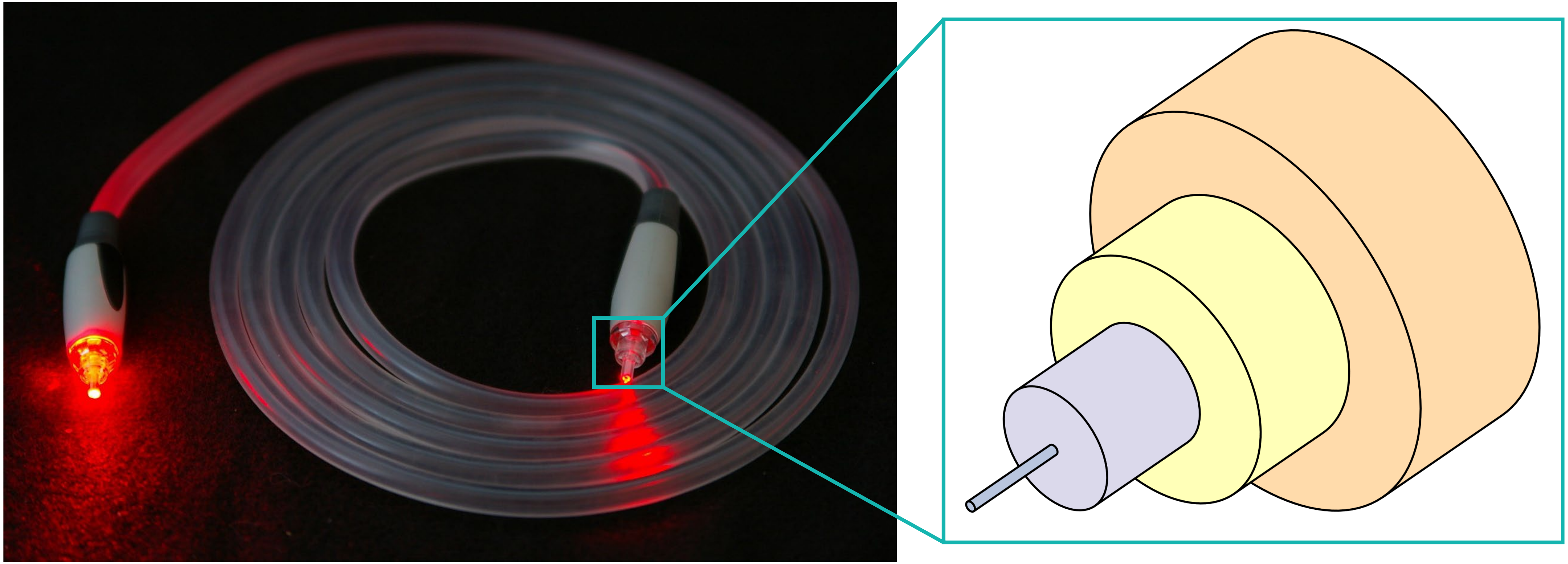
Optimizing Gradient-Index (GRIN) Optics

Luneburg Lens



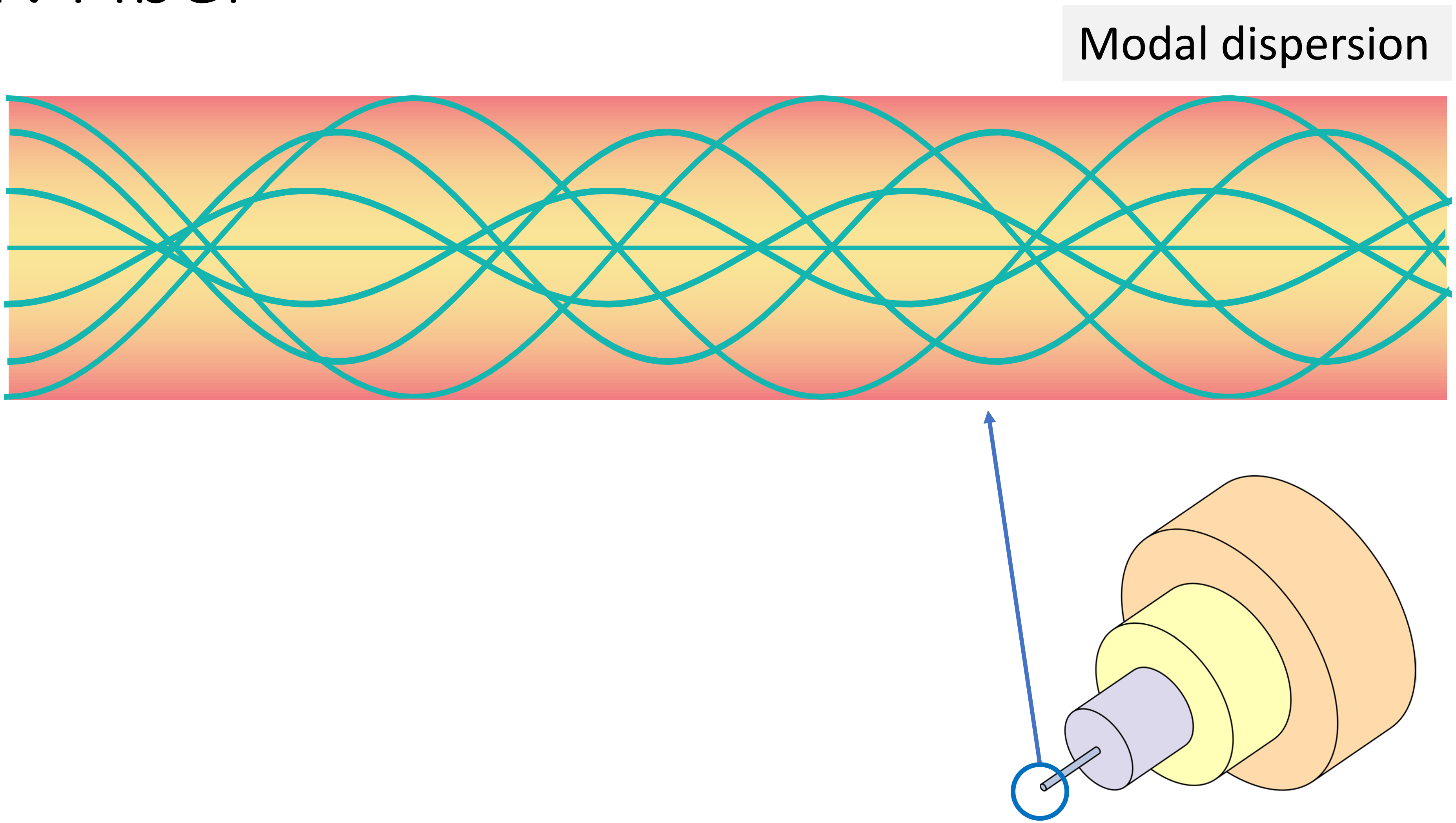
GRIN Fiber

GRIN Fiber

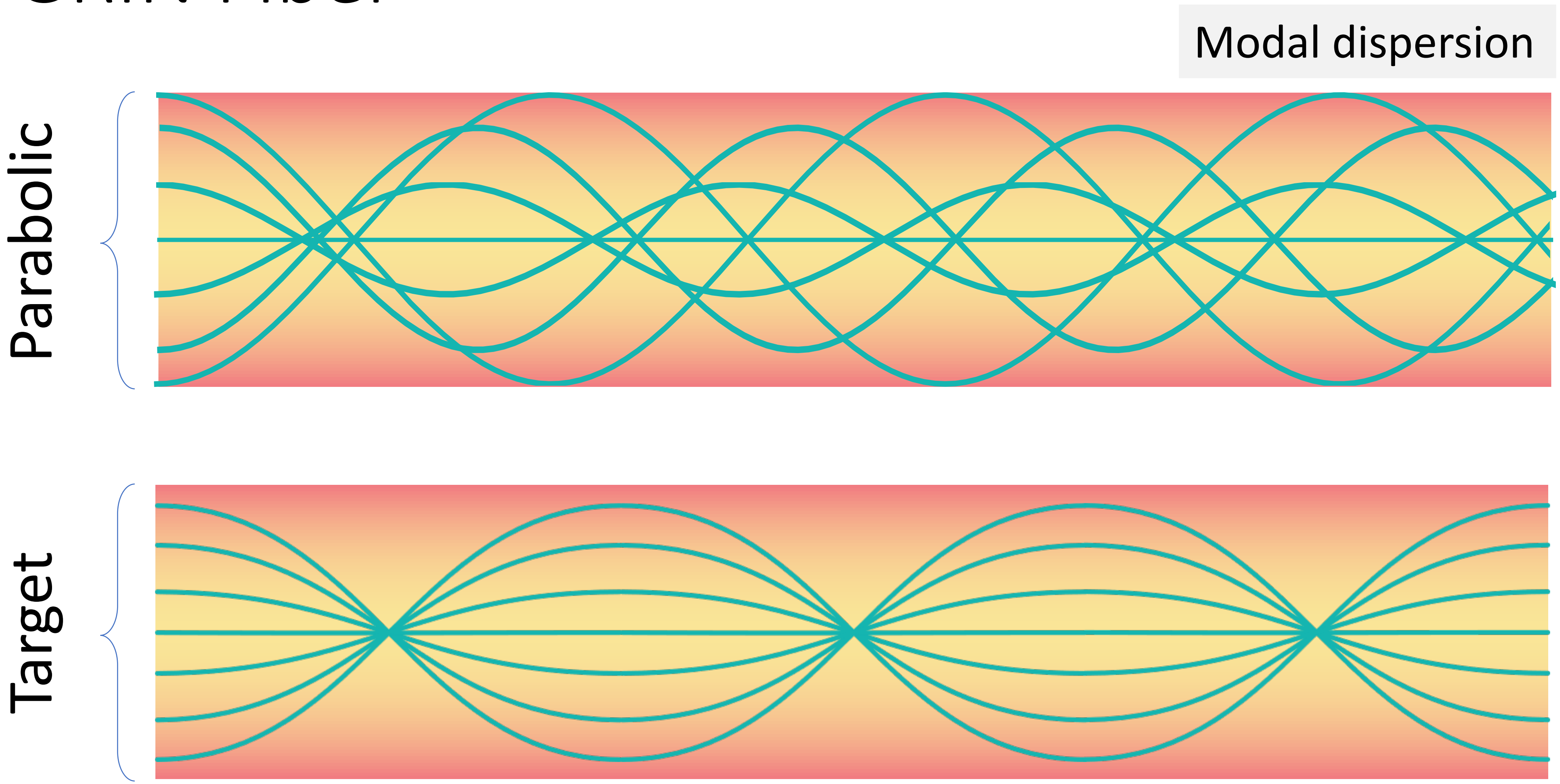


https://en.wikipedia.org/wiki/Optical_fiber

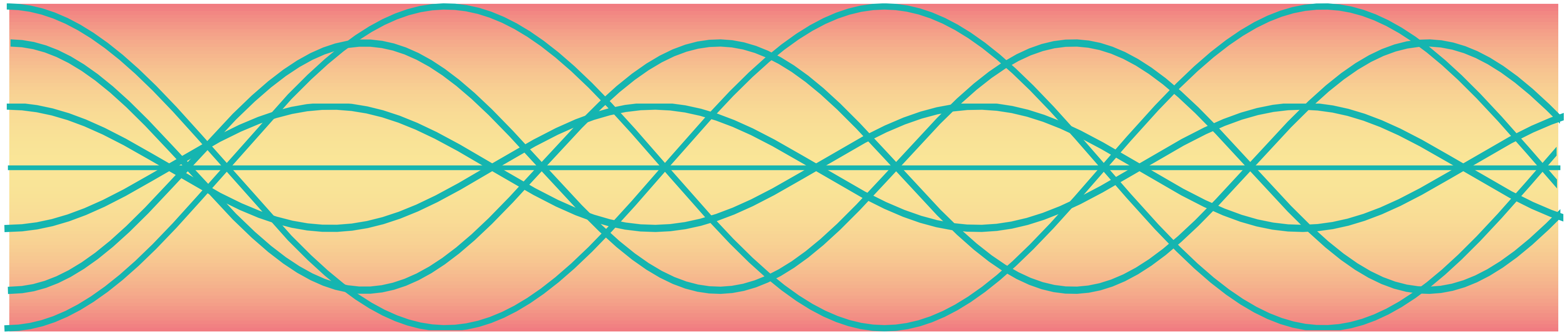
GRIN Fiber



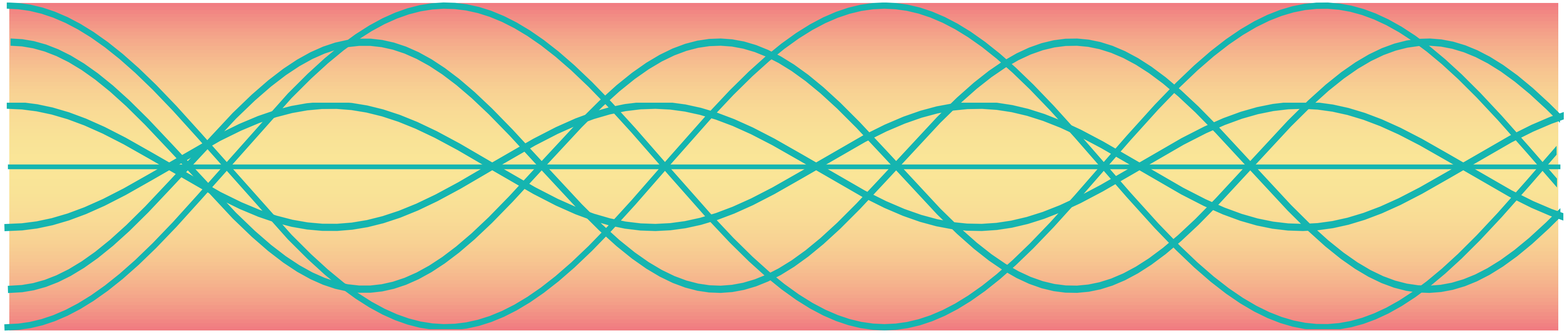
GRIN Fiber

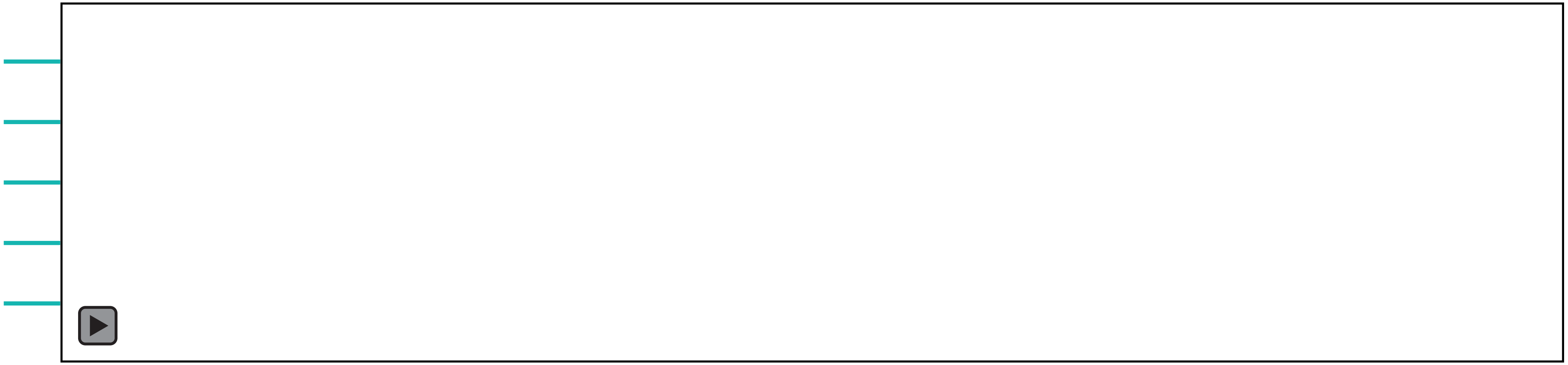


GRIN Fiber

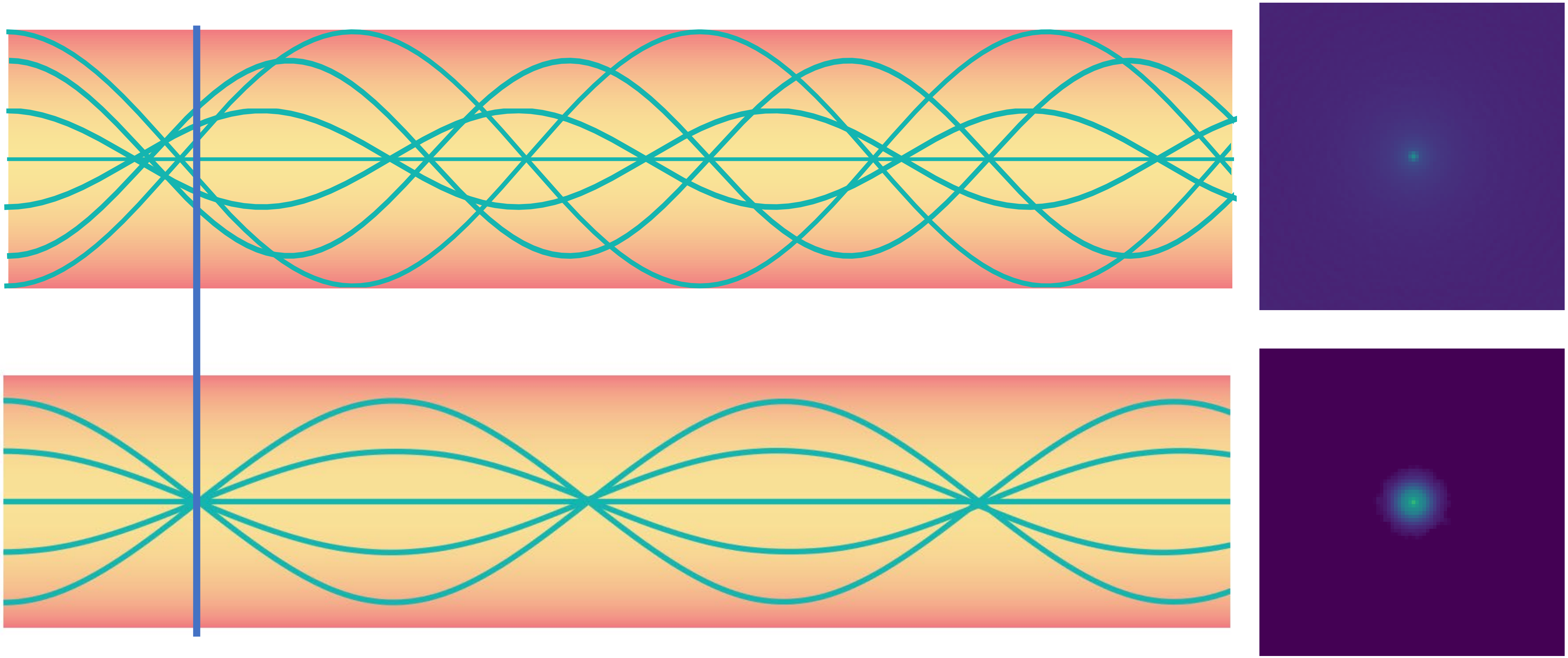


GRIN Fiber

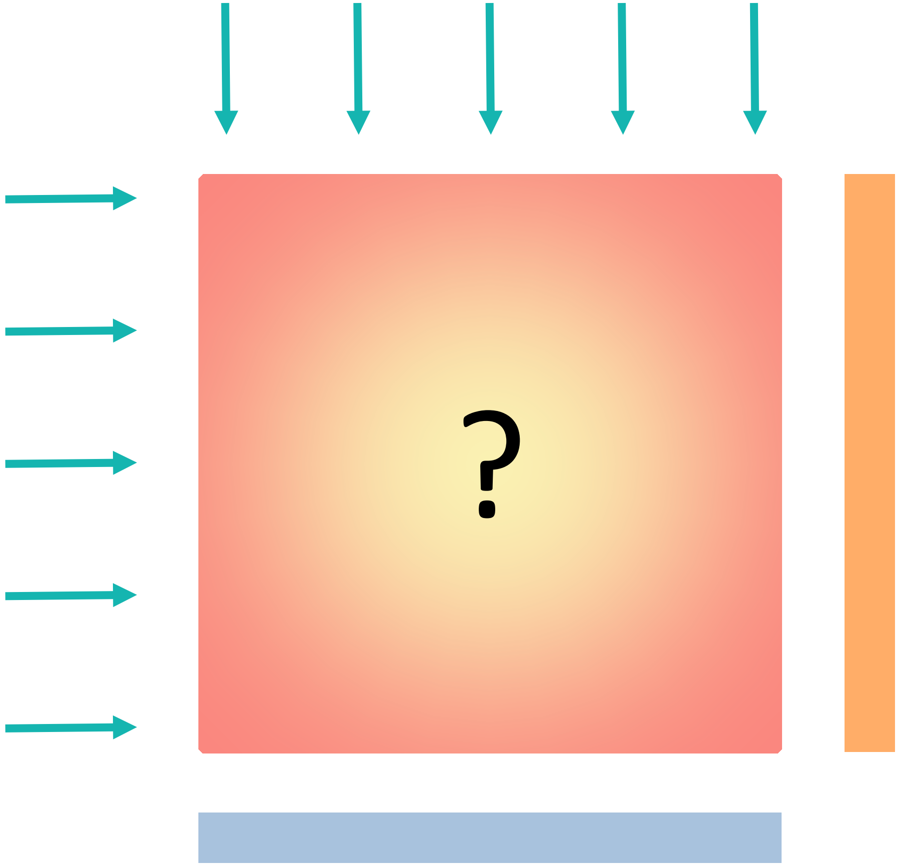


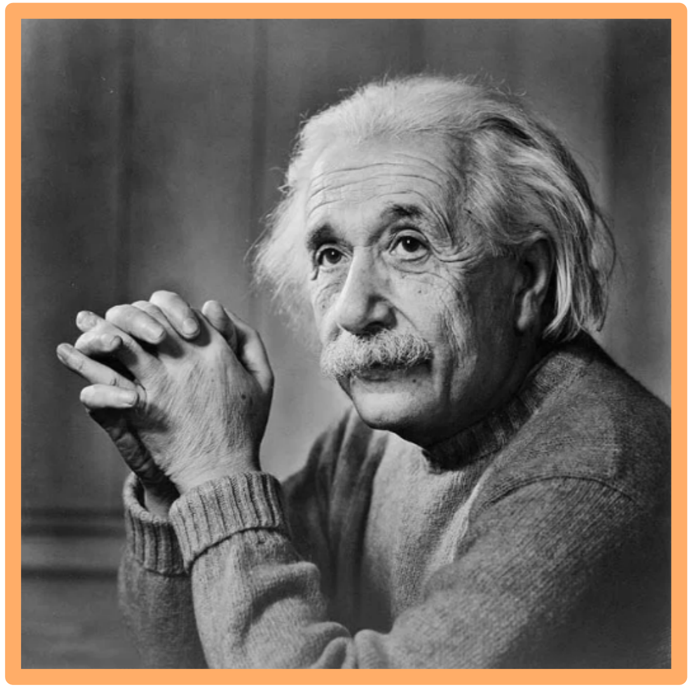


GRIN Fiber

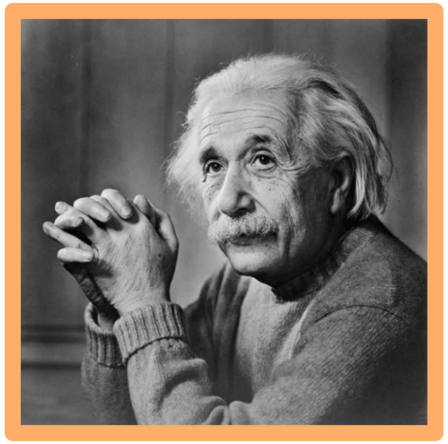


Multiview Display

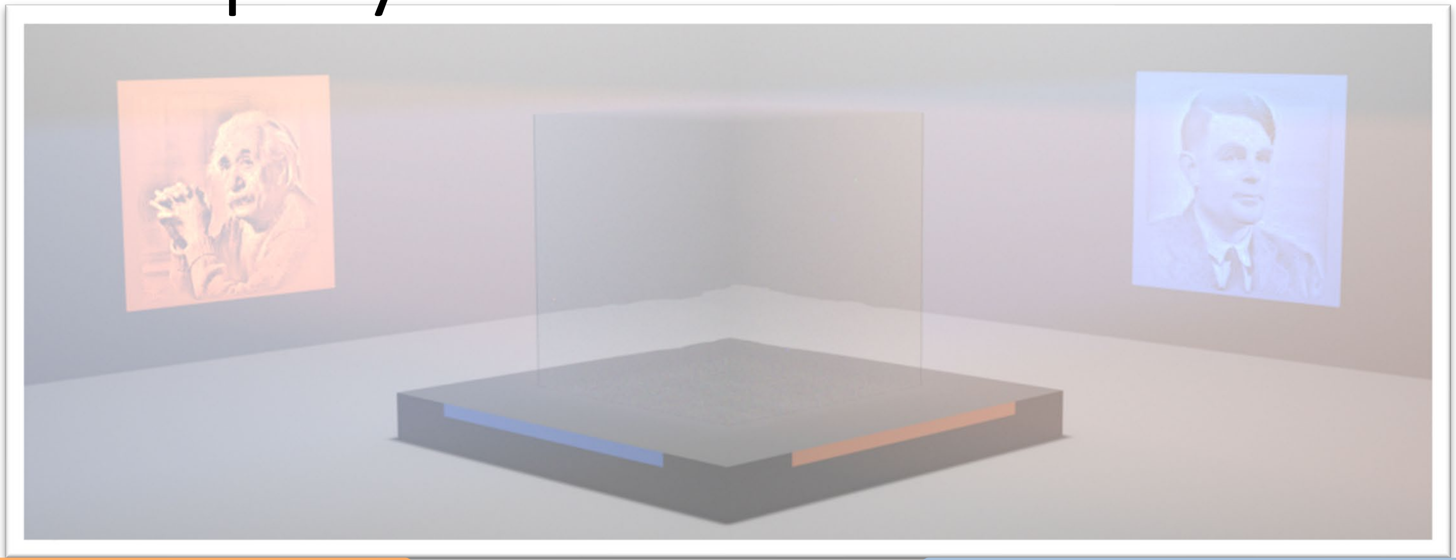


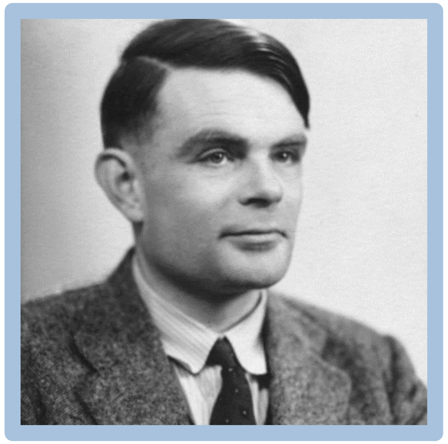


Multiview Display

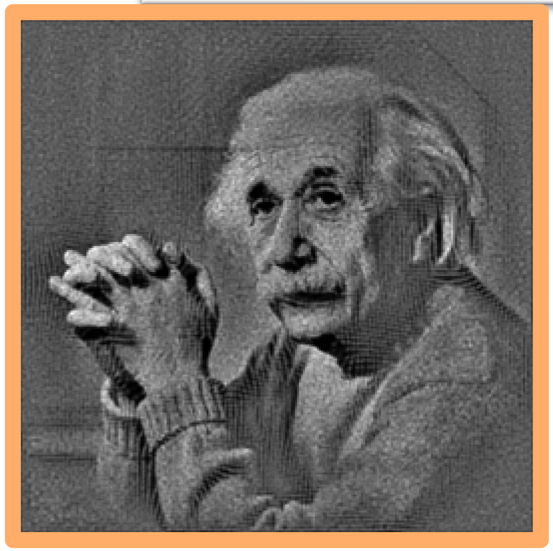


Target





Target

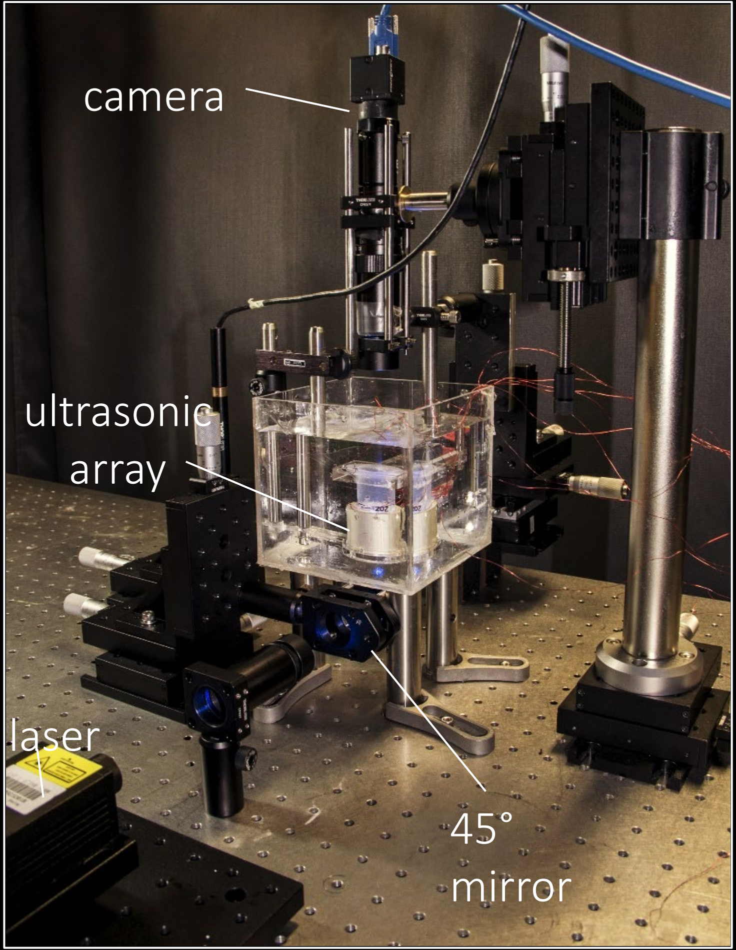


optimization results

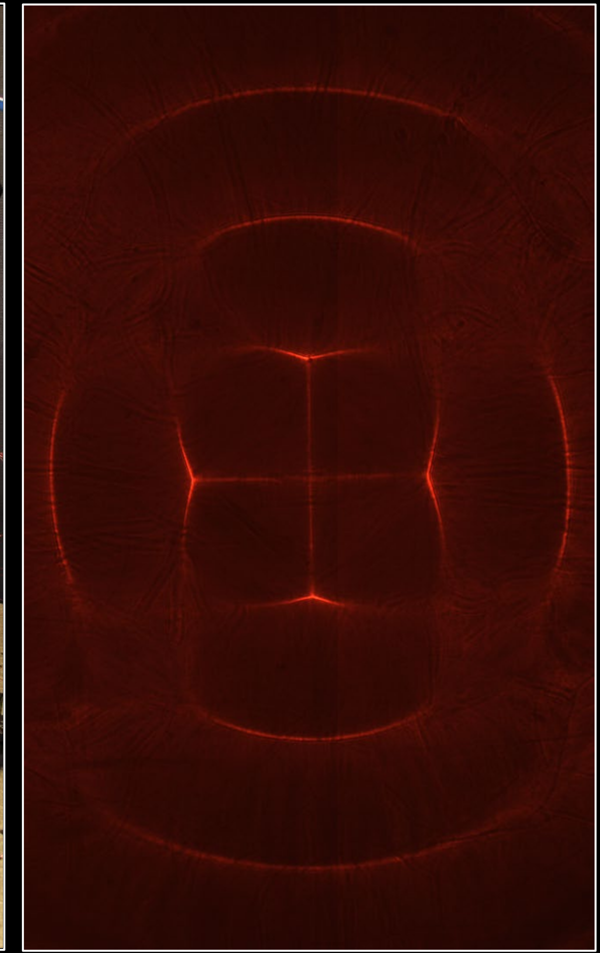
Media with continuously varying refractive index and scattering

unbiased techniques for scientific imaging

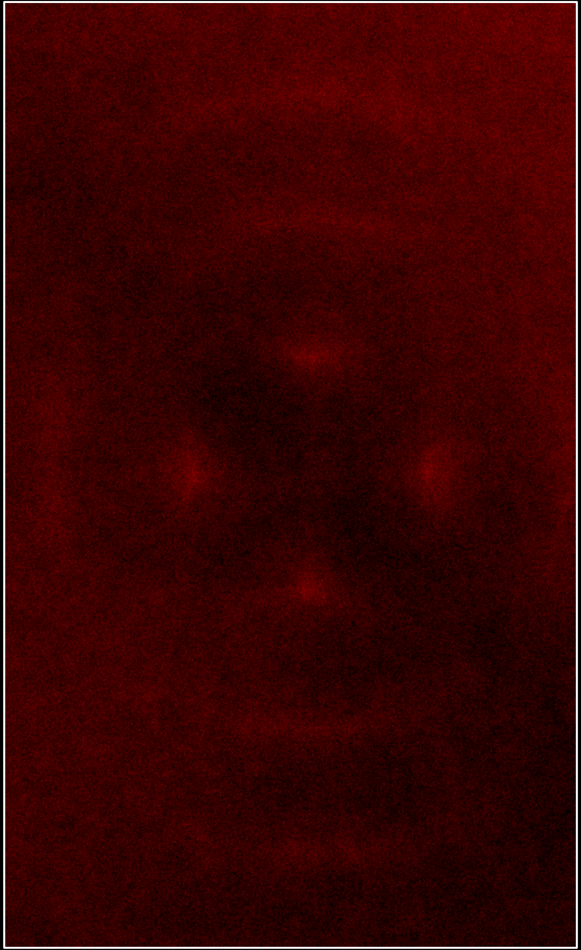
experimental hardware



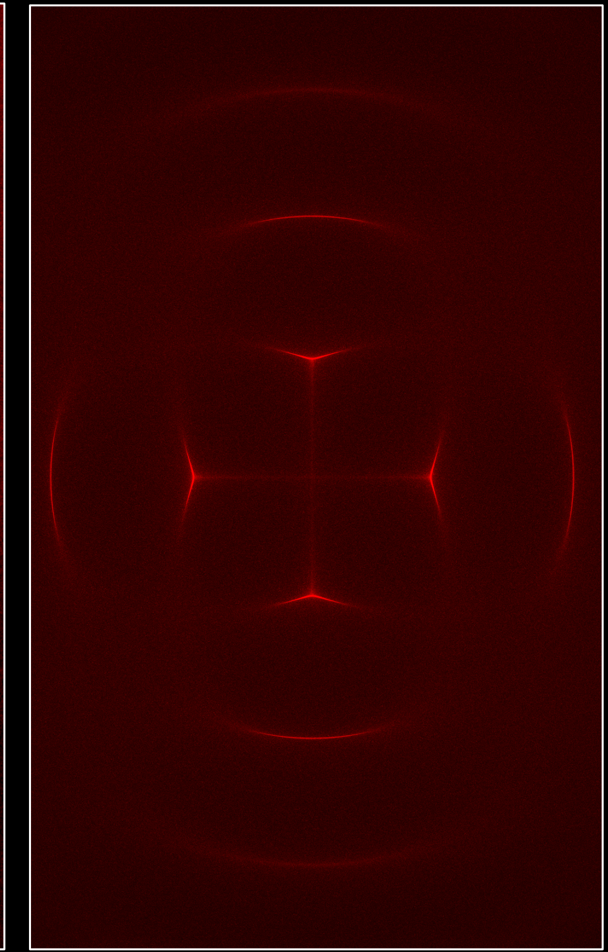
experimental capture

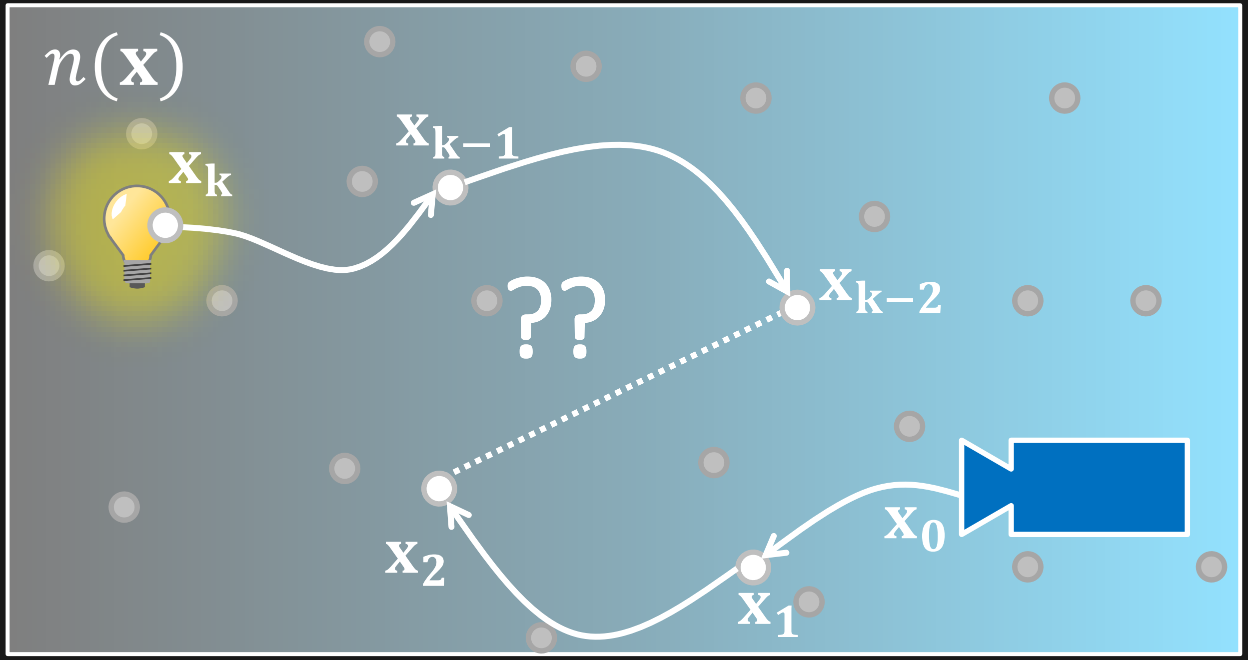


photon mapping

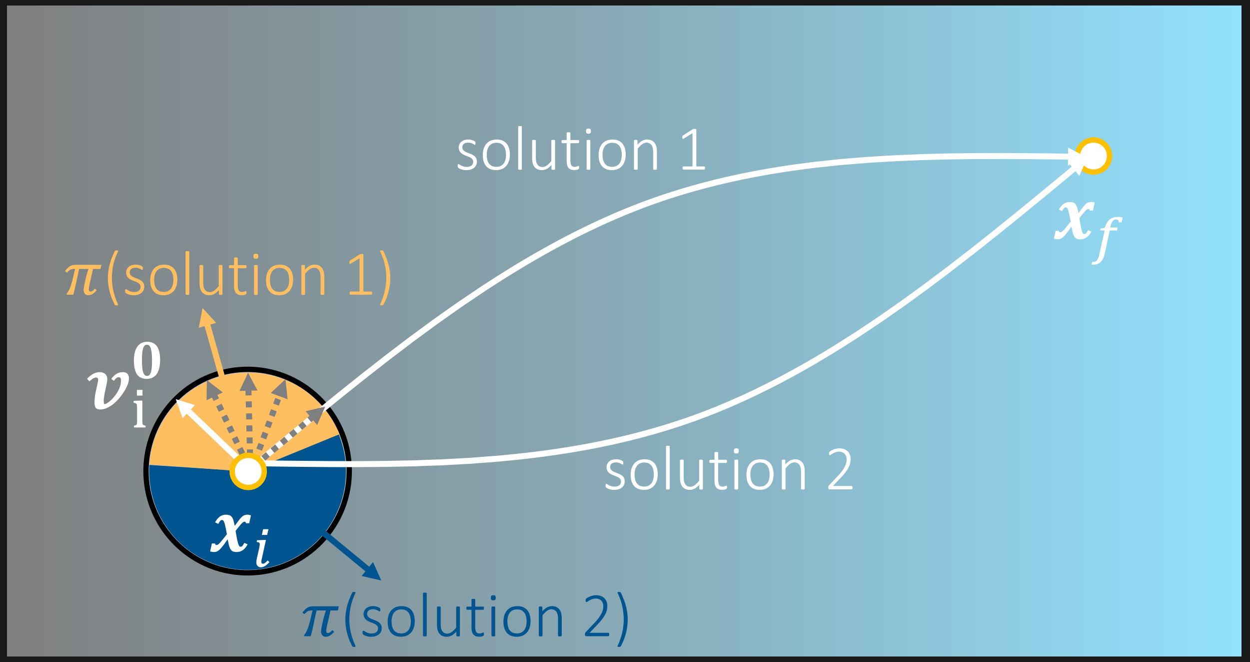


unbiased (ours)

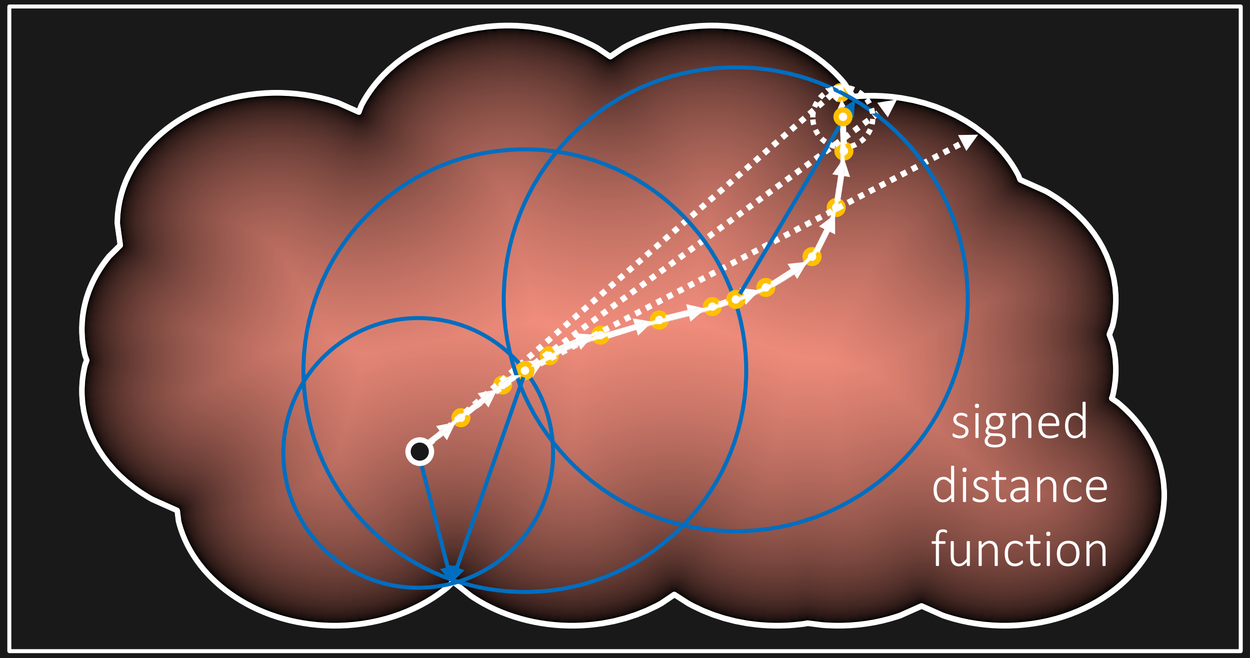




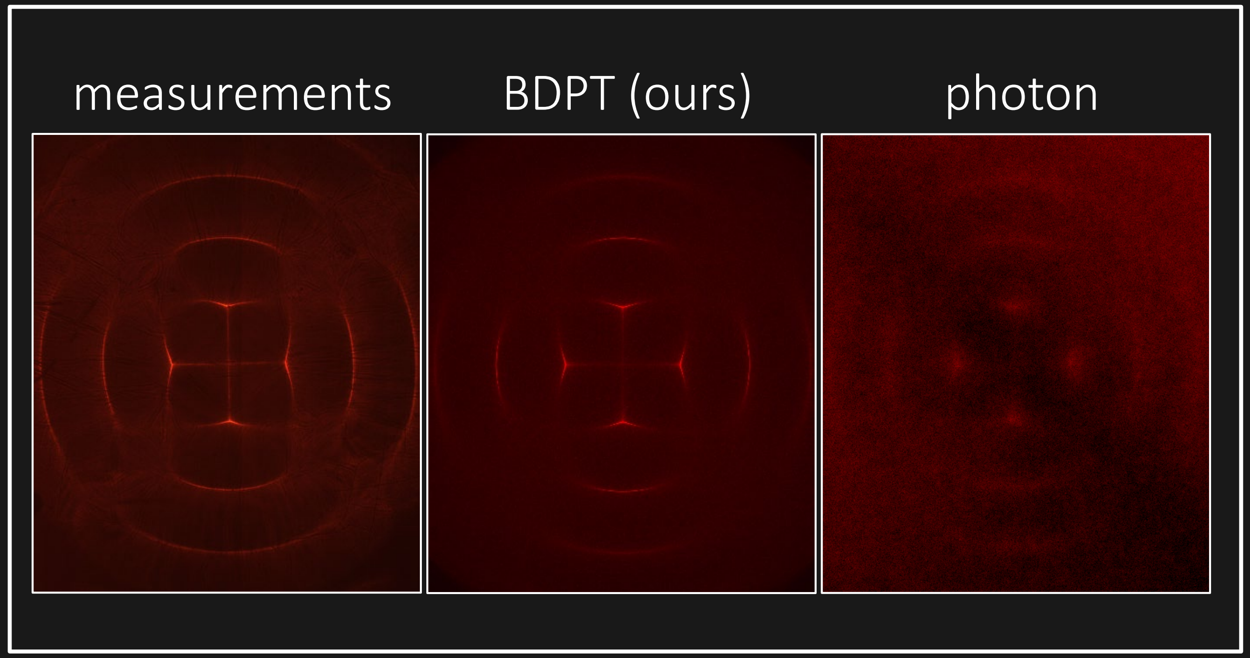
1. background on refractive radiative transfer equation



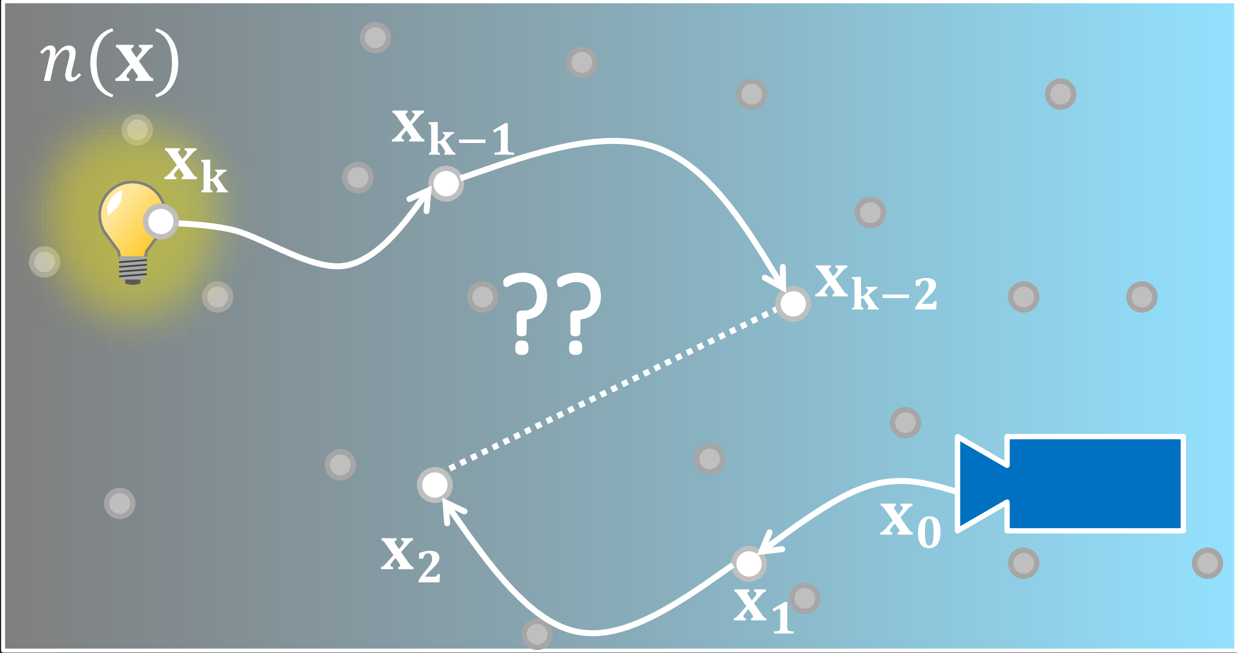
2. direct connections: our solution to unbiased rendering



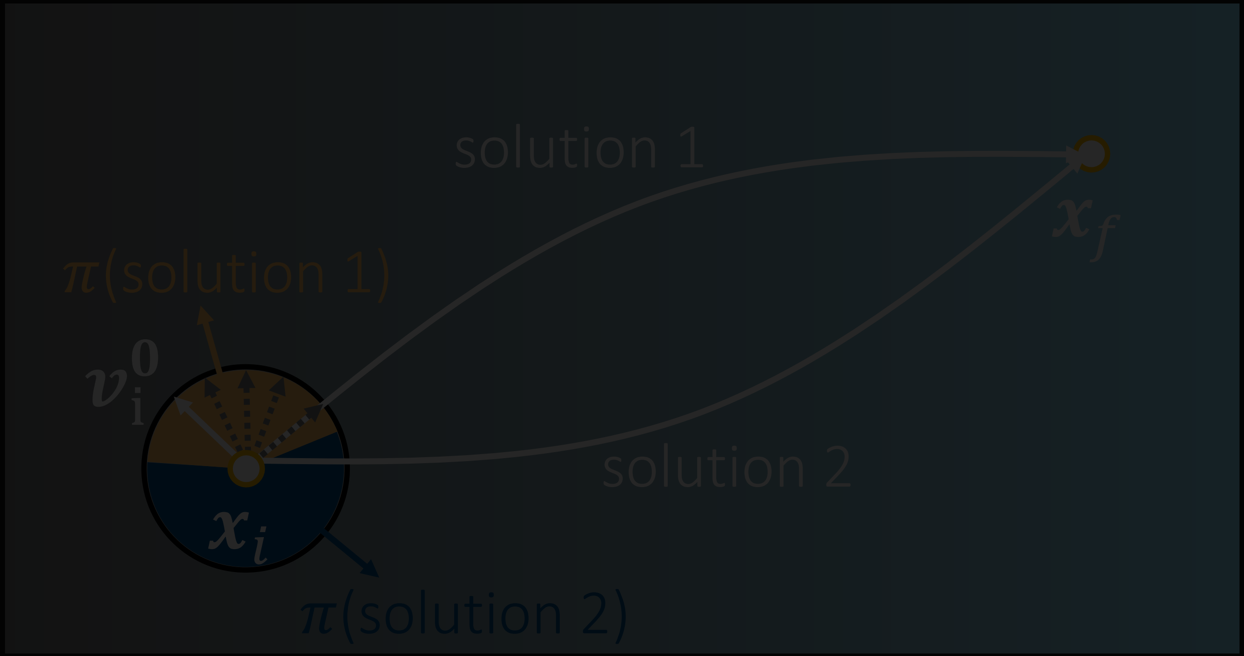
3. acceleration techniques



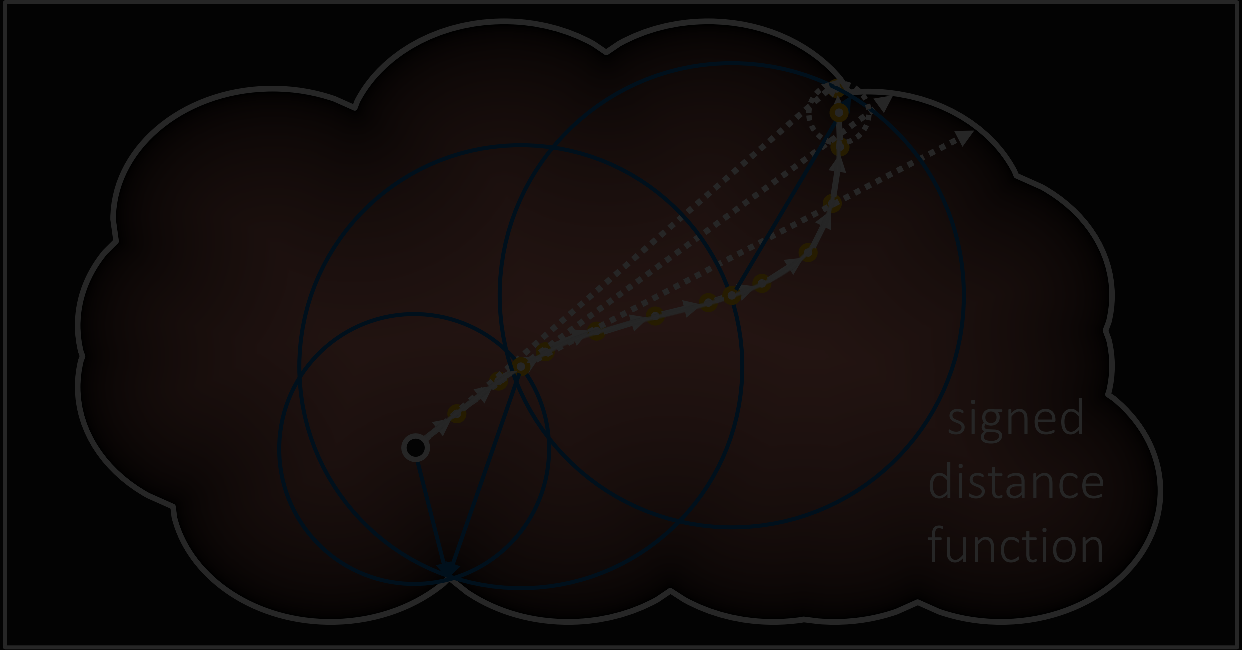
4. experiments



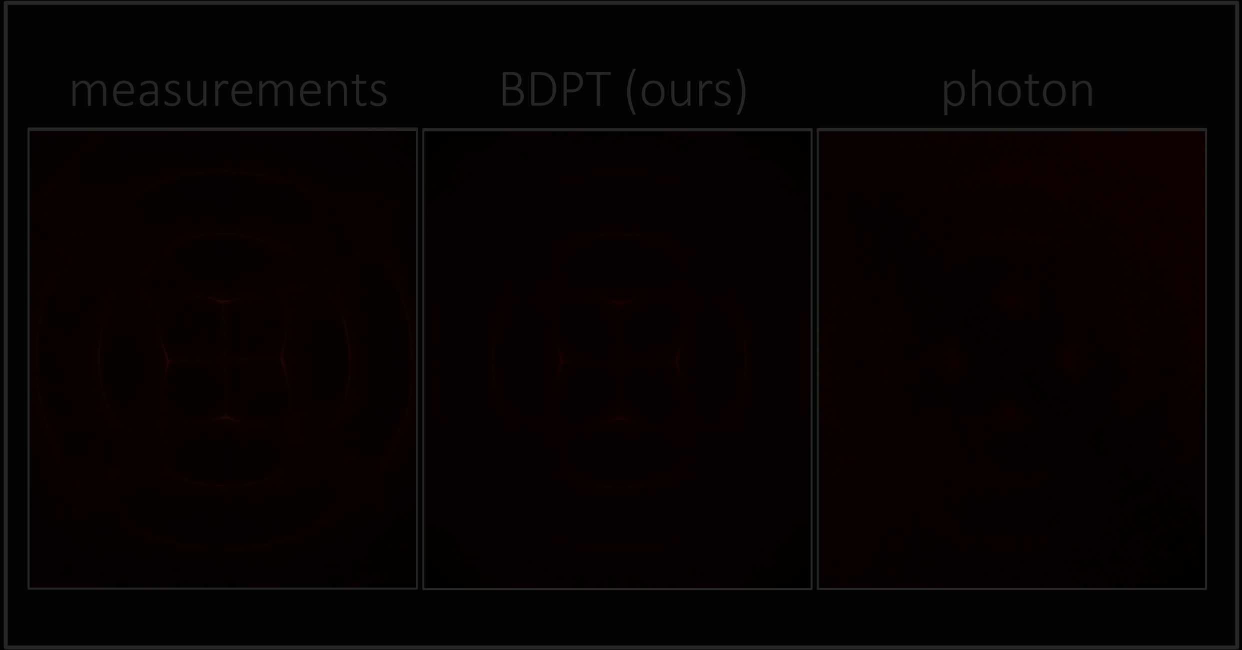
1. background on refractive radiative transfer equation



2. direct connections: our solution to unbiased rendering

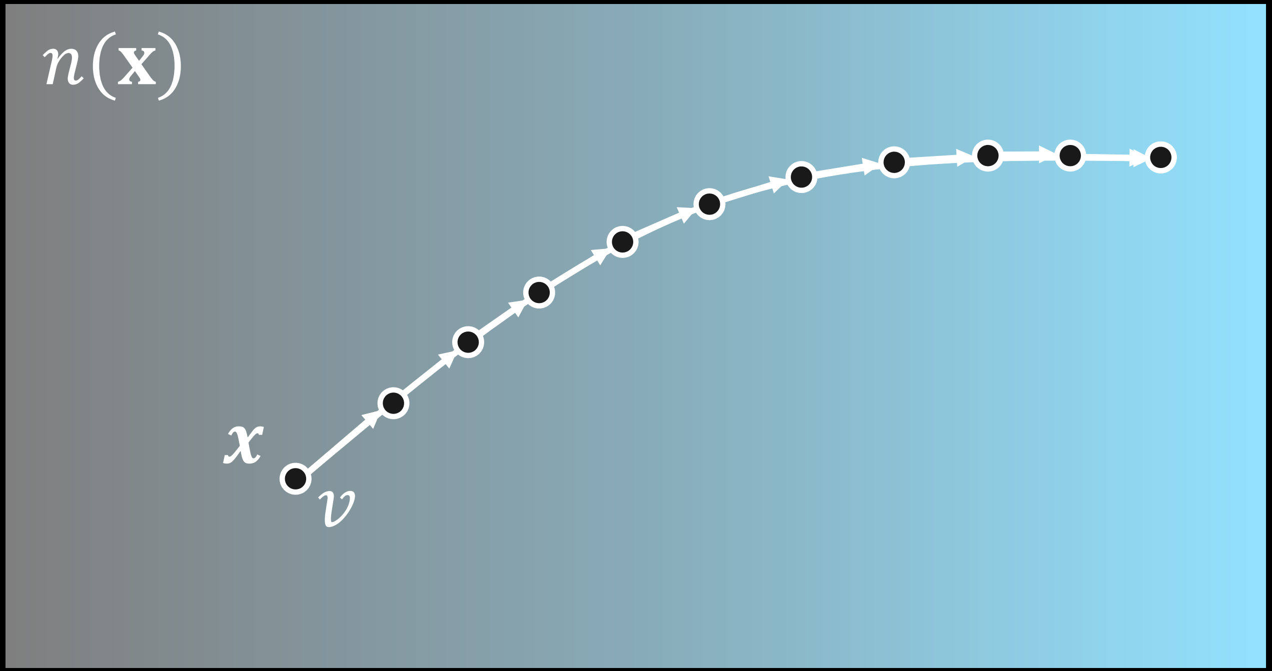


3. acceleration techniques



4. experiments

continuous refraction and no scattering

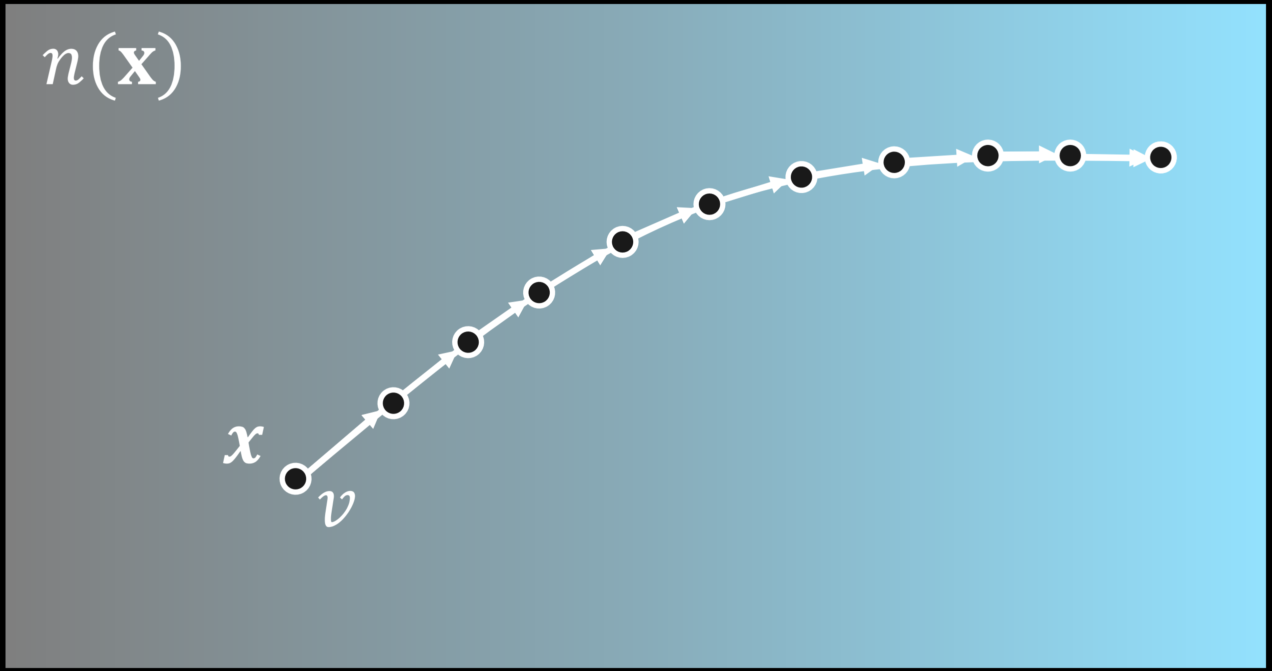


Hamilton's equations for refractive ray tracing

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x})$$

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

continuous refraction and no scattering



Hamilton's equations for refractive ray tracing

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x})$$

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

solved using symplectic integration

scattering and no continuous refraction

radiative transfer equation (RTE)

$$\frac{\mathrm{d}L}{\mathrm{d}s} = \sigma_a L_e - (\sigma_a + \sigma_s) L + \frac{\sigma_s}{4\pi} \int f_s(\omega', \omega) L \mathrm{d}\omega'$$

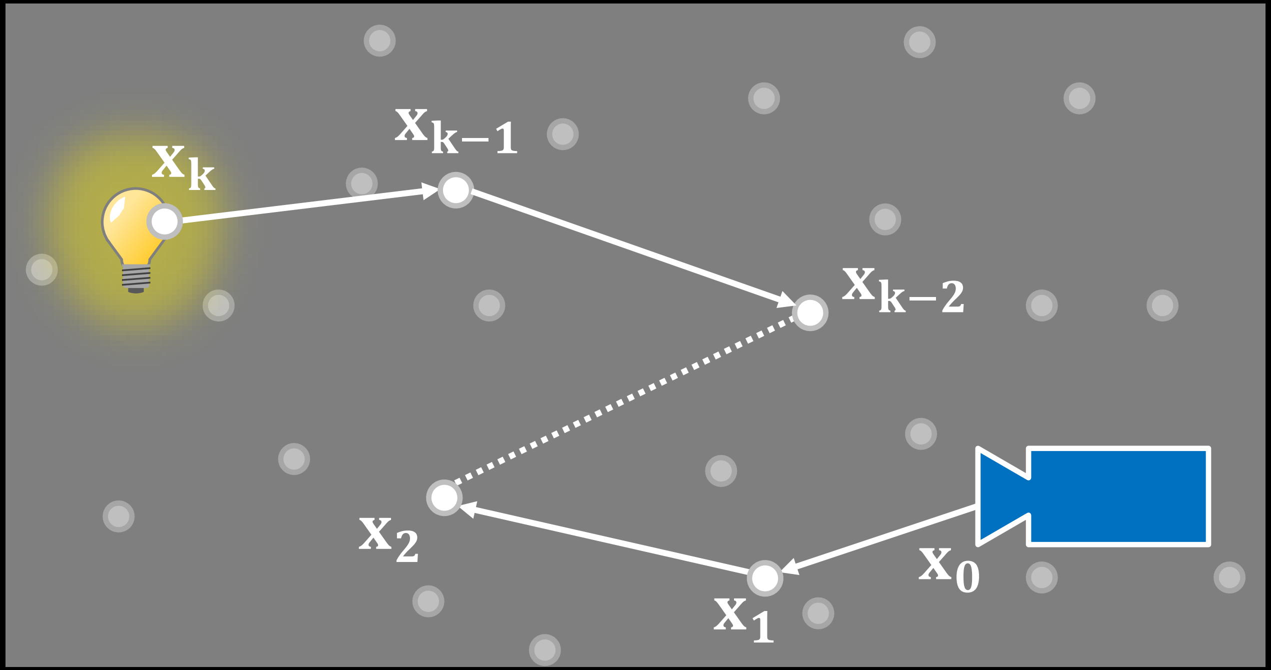
scattering and no continuous refraction

radiative transfer equation (RTE)

$$\frac{\mathrm{d}L}{\mathrm{d}s} = \sigma_a L_e - (\sigma_a + \sigma_s) L + \frac{\sigma_s}{4\pi} \int f_s(\omega', \omega) L \,\mathrm{d}\omega'$$

solved using Monte Carlo integration

scattering and no continuous refraction



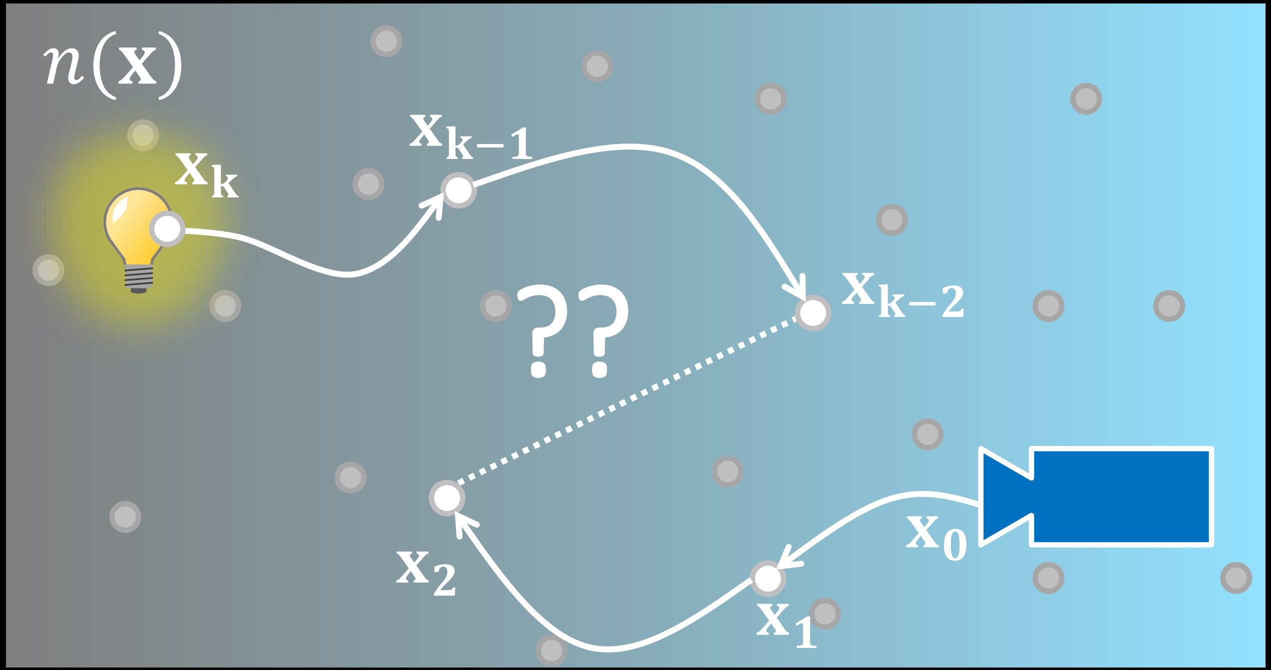
bidirectional path tracing (BDPT):

1.trace a random sensor subpath

2.trace a random emitter subpath

3.join vertices with a straight line

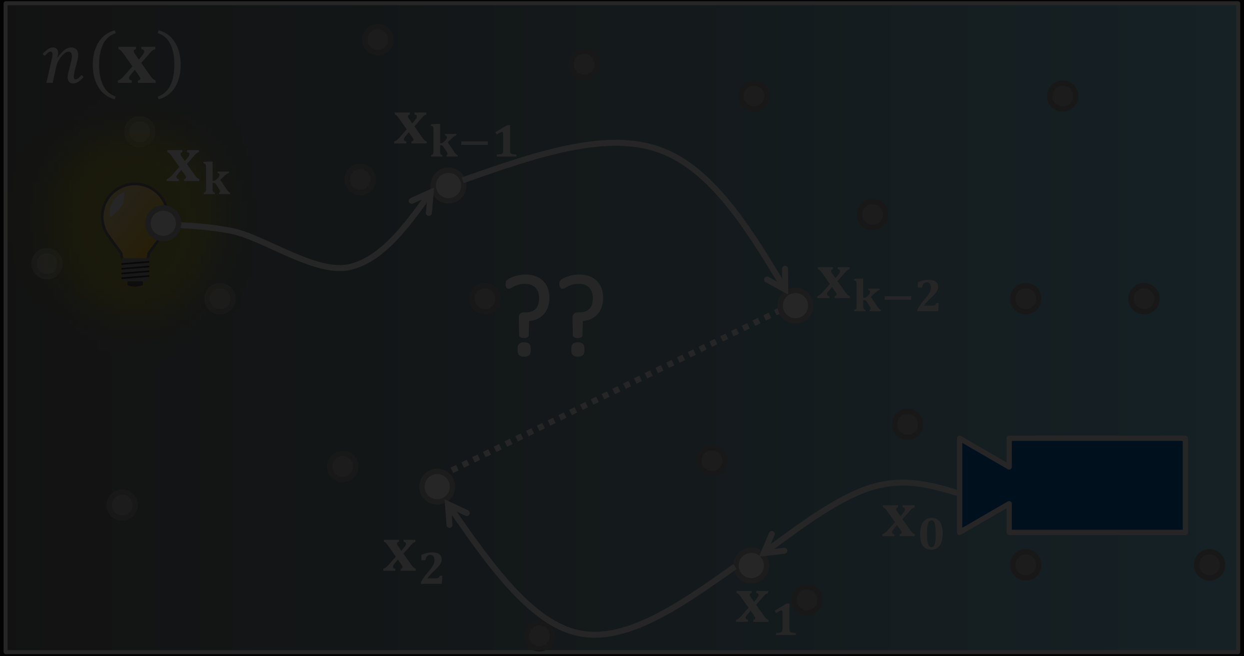
continuous refraction and scattering



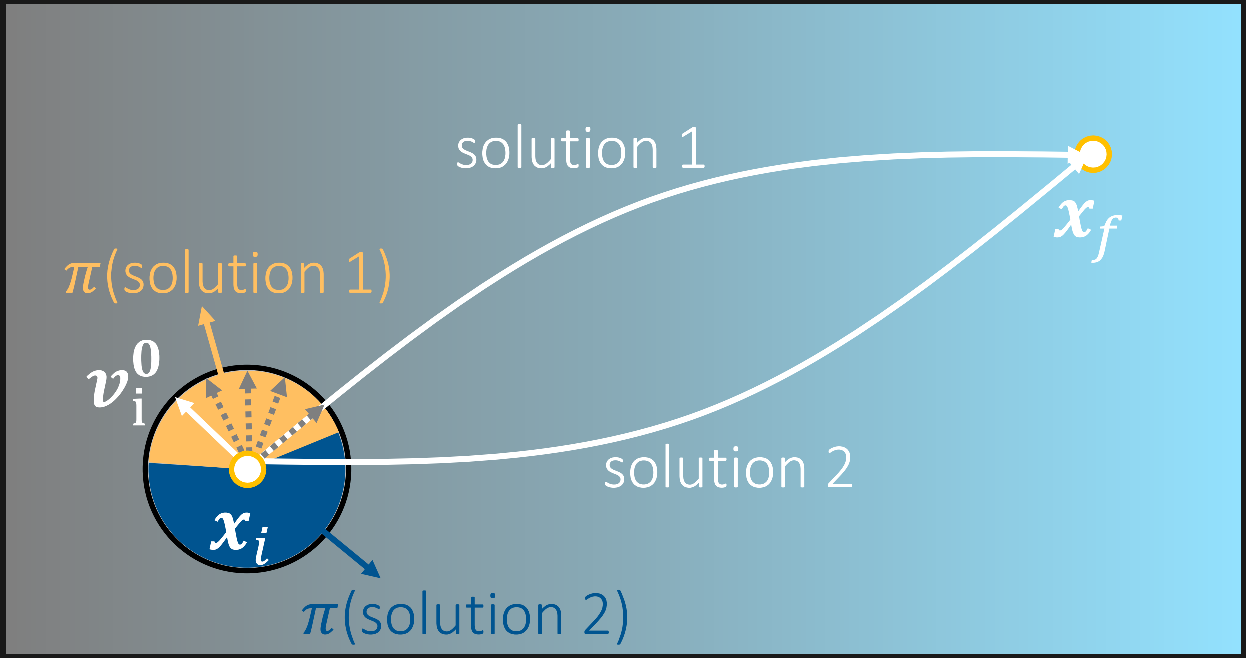
bidirectional path tracing (BDPT):

1. trace a random sensor subpath
   use refractive ray tracing
2. trace a random emitter subpath

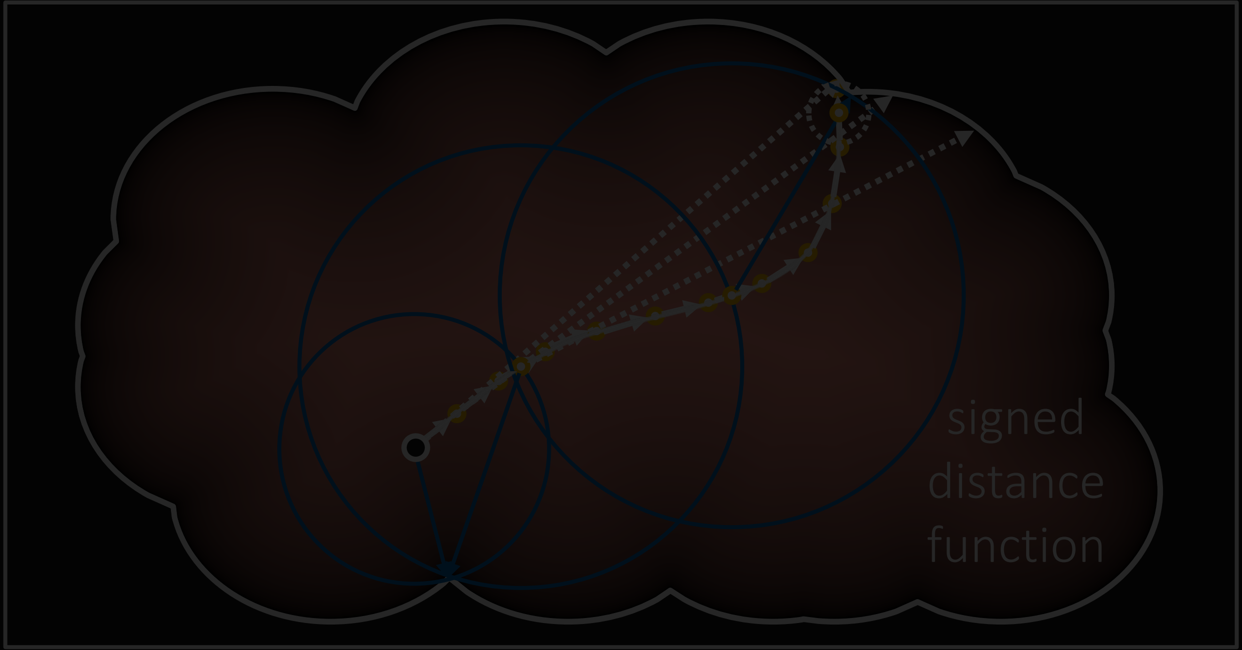
3. join vertices with a ~~straight line~~ curve



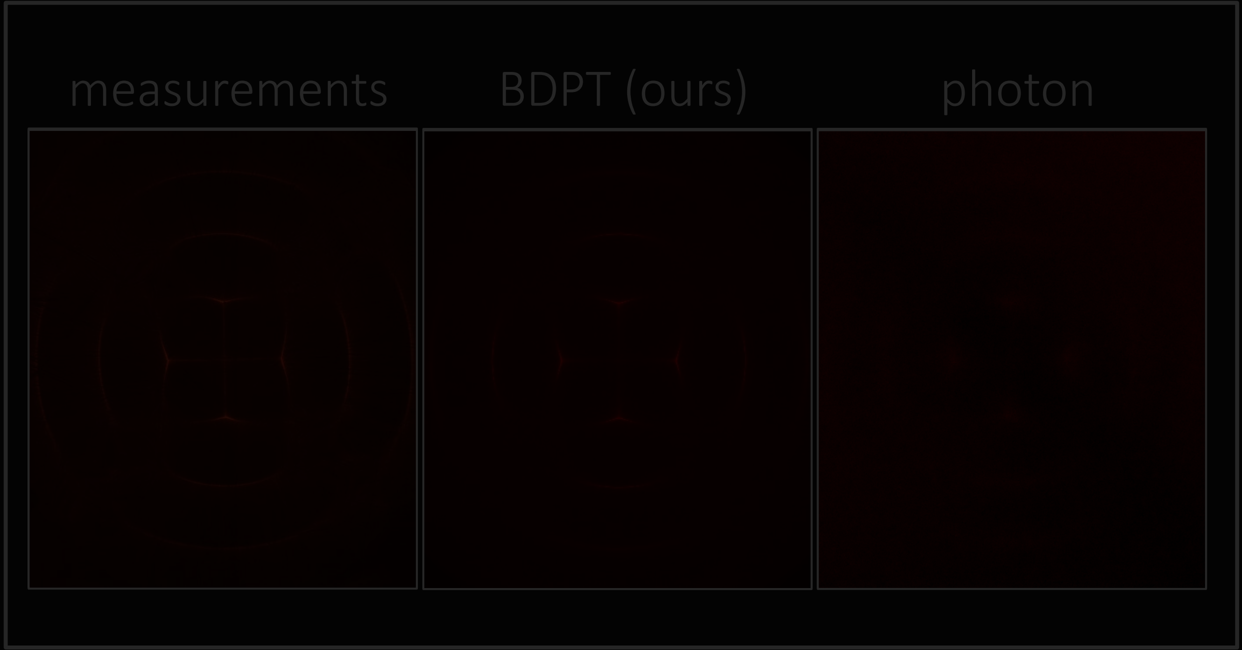
1. background on refractive radiative transfer equation



2. direct connections: our solution to unbiased rendering



3. acceleration techniques



4. experiments

direct connections

we have to solve this:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x}), \quad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{x}_f$

boundary value problem (BVP)

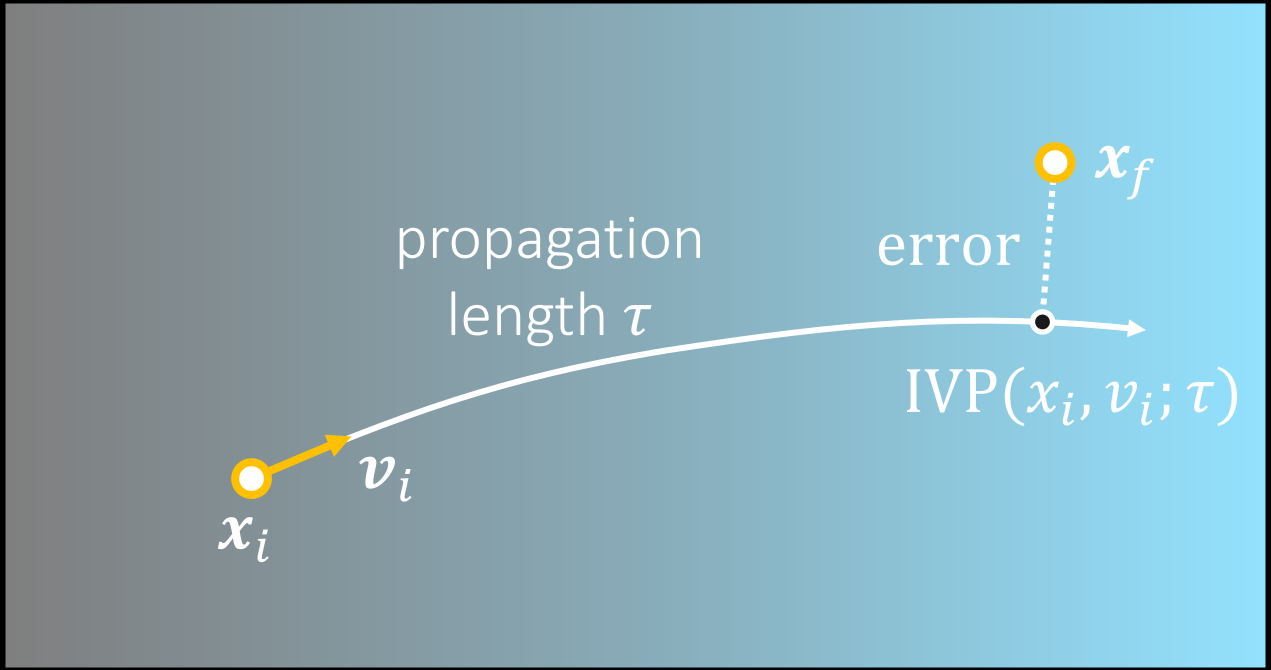
we know how to solve this:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x}), \quad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{v}_i$

initial value problem (IVP),
a.k.a. refractive ray tracing

direct connections



$$\text{error}(x_f, x_i, v_i) \equiv \min_{\tau} \left\| x_f - \text{IVP}(x_i, v_i; \tau) \right\|^2$$

we have to solve this:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x}), \quad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{x}_f$

boundary value problem (BVP)

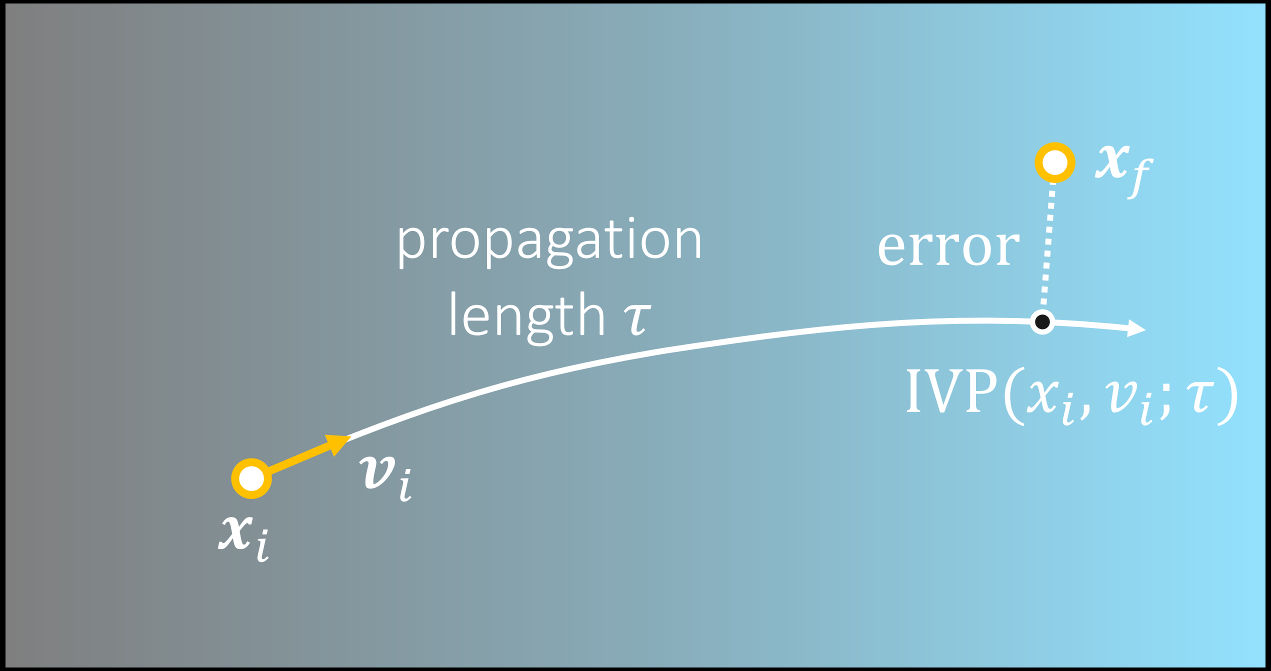
we know how to solve this:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x}), \quad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{v}_i$

initial value problem (IVP),
a.k.a. refractive ray tracing

direct connections



$$\mathrm{error}(x_f, x_i, v_i) \equiv \min_{\tau} \left\| x_f - \mathrm{IVP}(x_i, v_i; \tau) \right\|^2$$

we have to solve this:

$$\min_{v_i} \mathrm{error}(x_f, x_i, v_i)$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{x}_f$

boundary value problem (BVP)

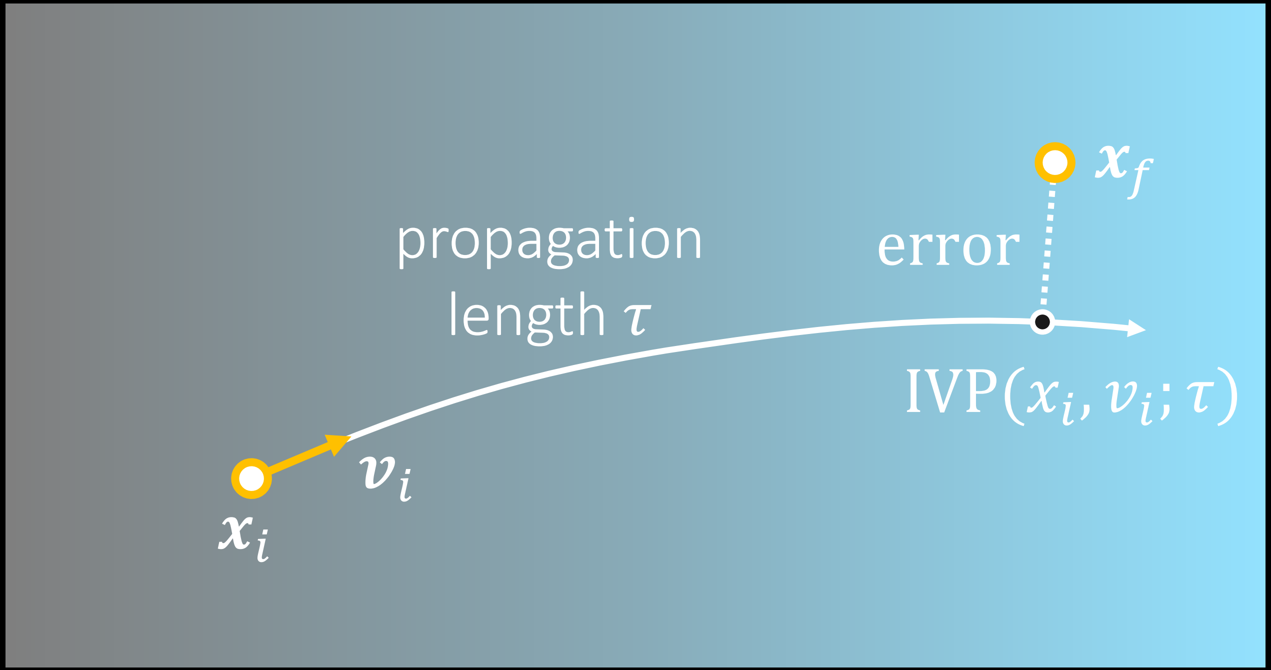
we know how to solve this:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x}), \quad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{v}_i$

initial value problem (IVP),
a.k.a. refractive ray tracing

direct connections



differentiable

$$\text{error}(x_f, x_i, v_i) \equiv \min_{\tau} \left\| x_f - \text{IVP}(x_i, v_i; \tau) \right\|^2$$

we have to solve this:

$$\min_{v_i} \text{error}(x_f, x_i, v_i)$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{x}_f$

boundary value problem (BVP)

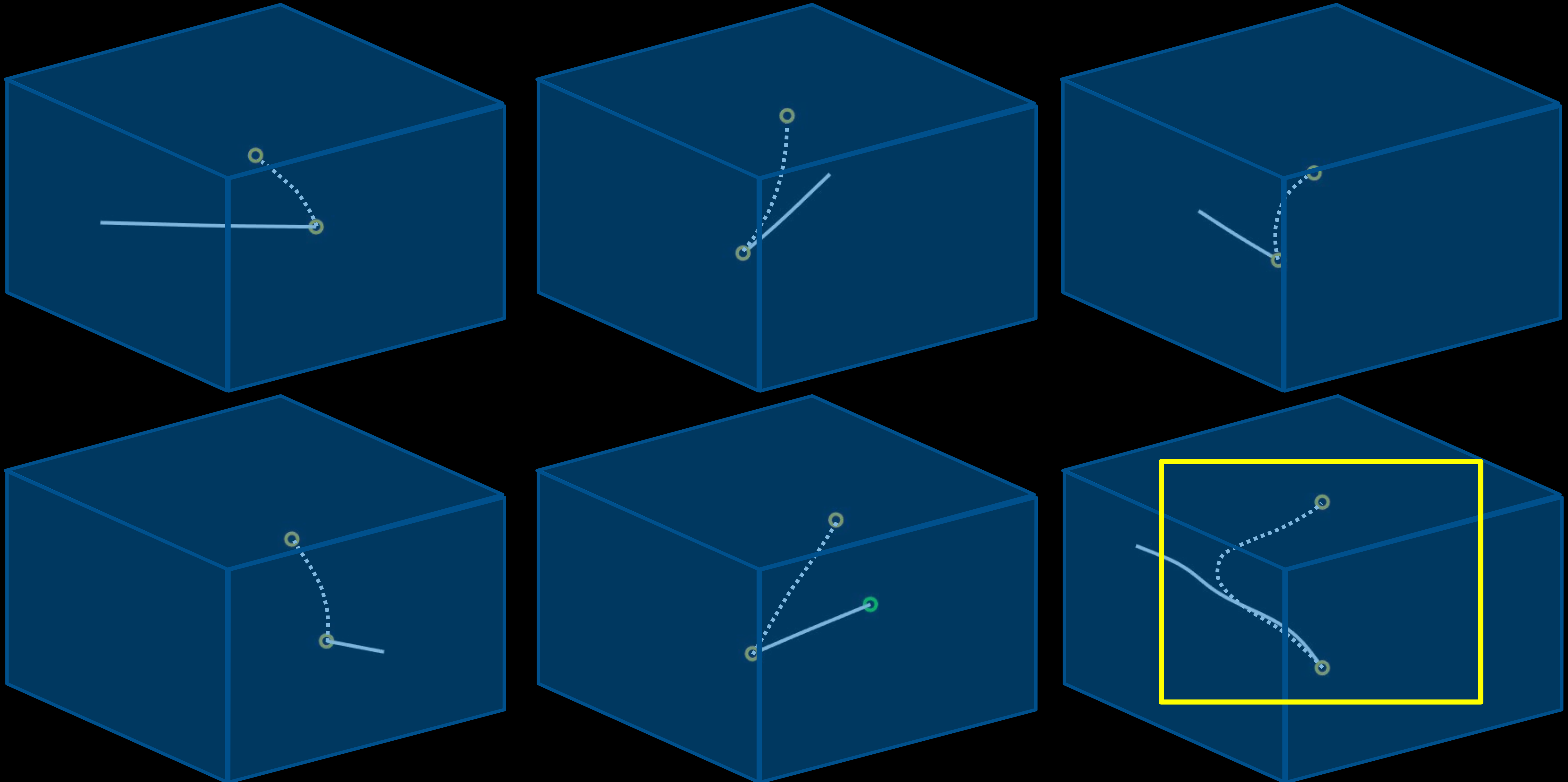
differentiable

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}s} = \nabla_x n(\boldsymbol{x}), \quad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\boldsymbol{v}}{n(\boldsymbol{x})}$$

boundary conditions: $\boldsymbol{x}_i, \boldsymbol{v}_i$

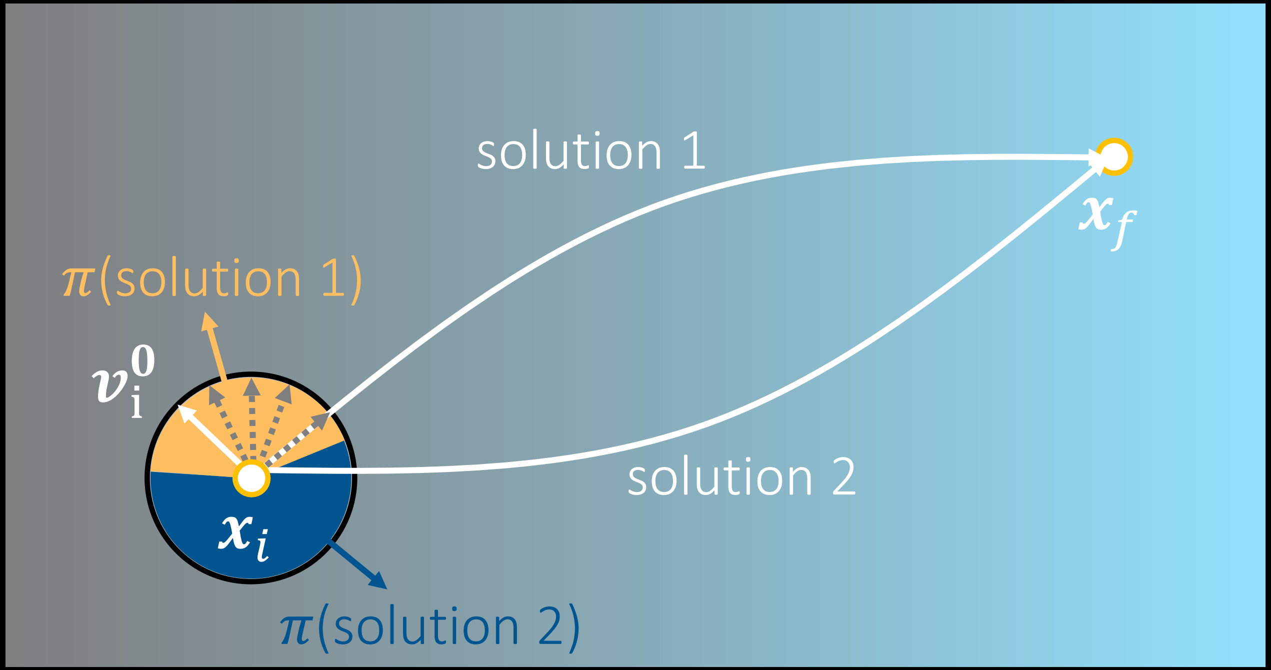
initial value problem (IVP),
a.k.a. refractive ray tracing

direct connections



multiple direct connections

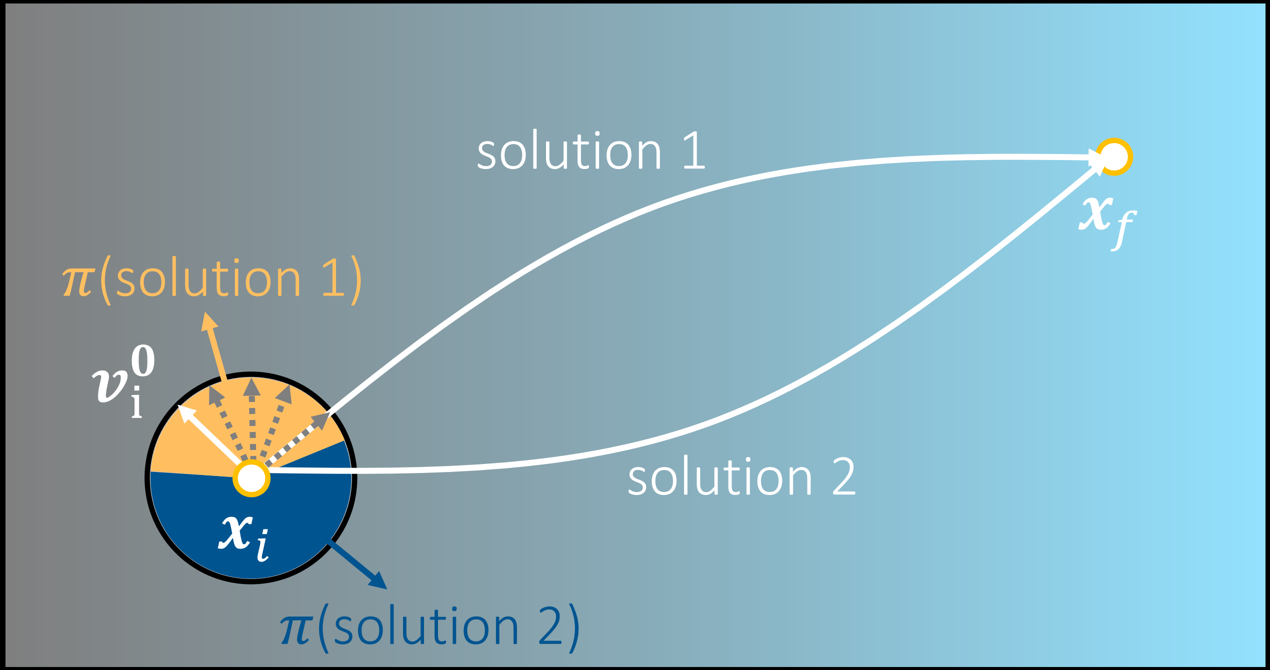
$$\text{total throughput} = \sum_{\text{all solutions}} \text{throughput(solution)}$$



approach 1:
exhaustively enumerate all solutions

multiple direct connections

$$\text{total throughput} = \sum_{\text{all solutions}} \text{throughput(solution)}$$



approach 1:
exhaustively enumerate all solutions

impractical

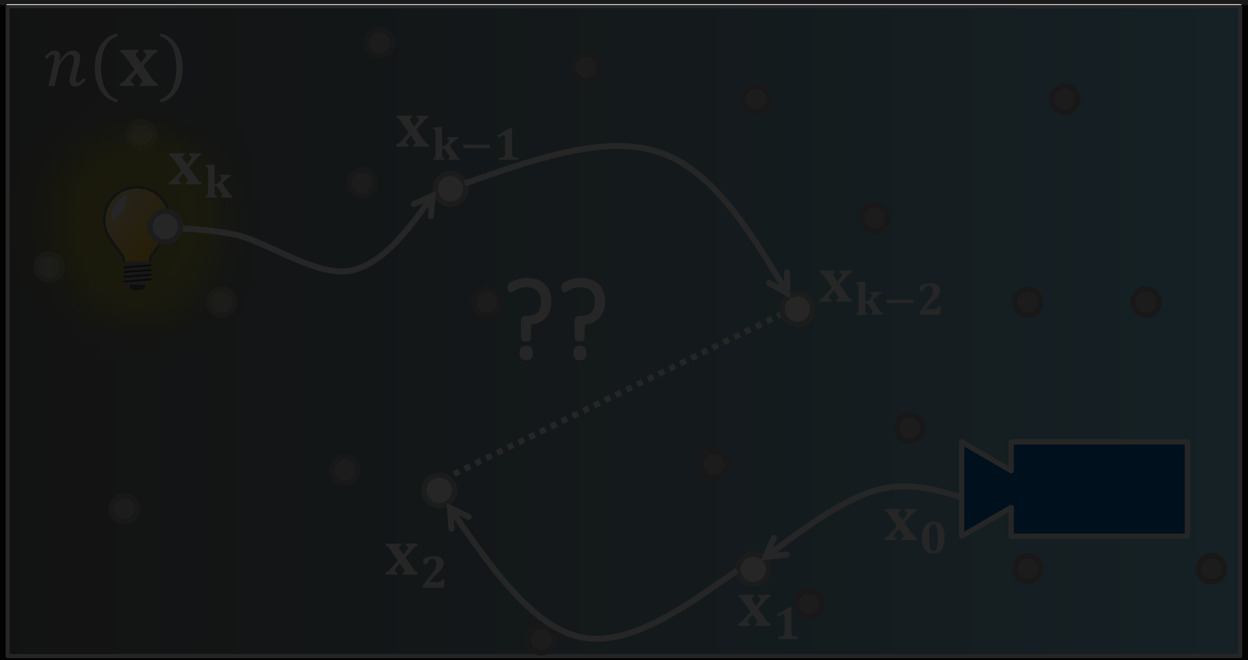
approach 2:
unbiased single-sample Monte Carlo

1. randomly sample initial direction
2. solve BVP
3. form estimate

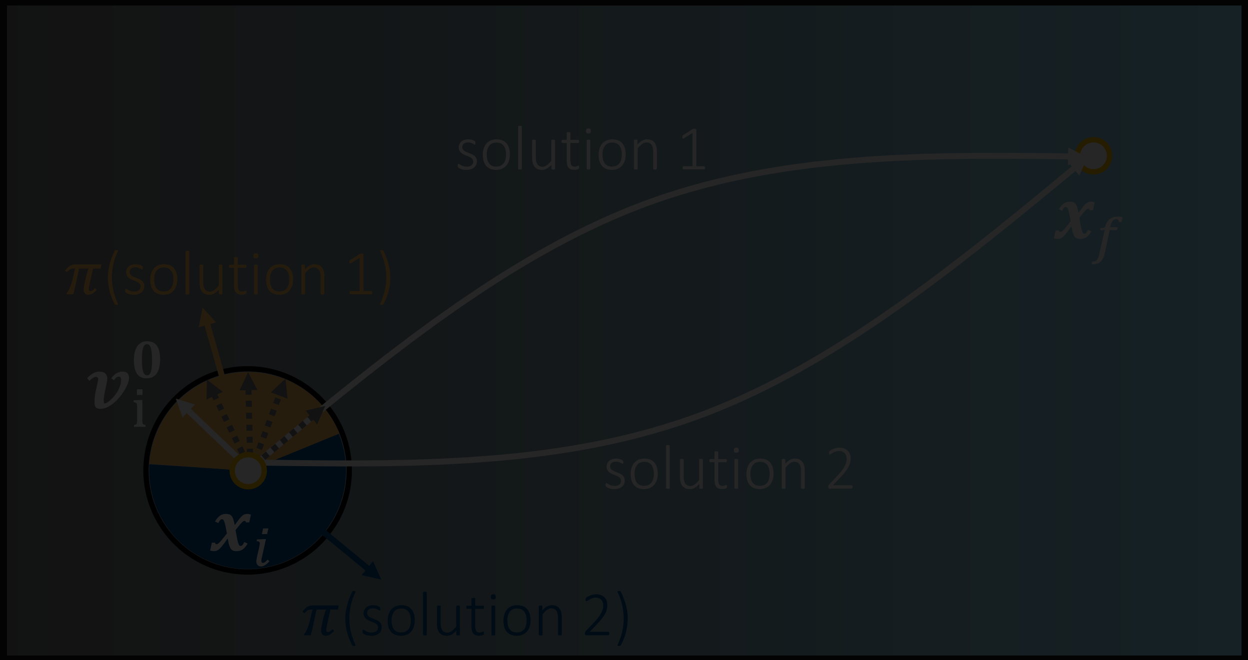
$$\text{total throughput} \approx \frac{\text{throughput(solution)}}{\text{probability(solution)}}$$

set of initial directions that converge to the solution

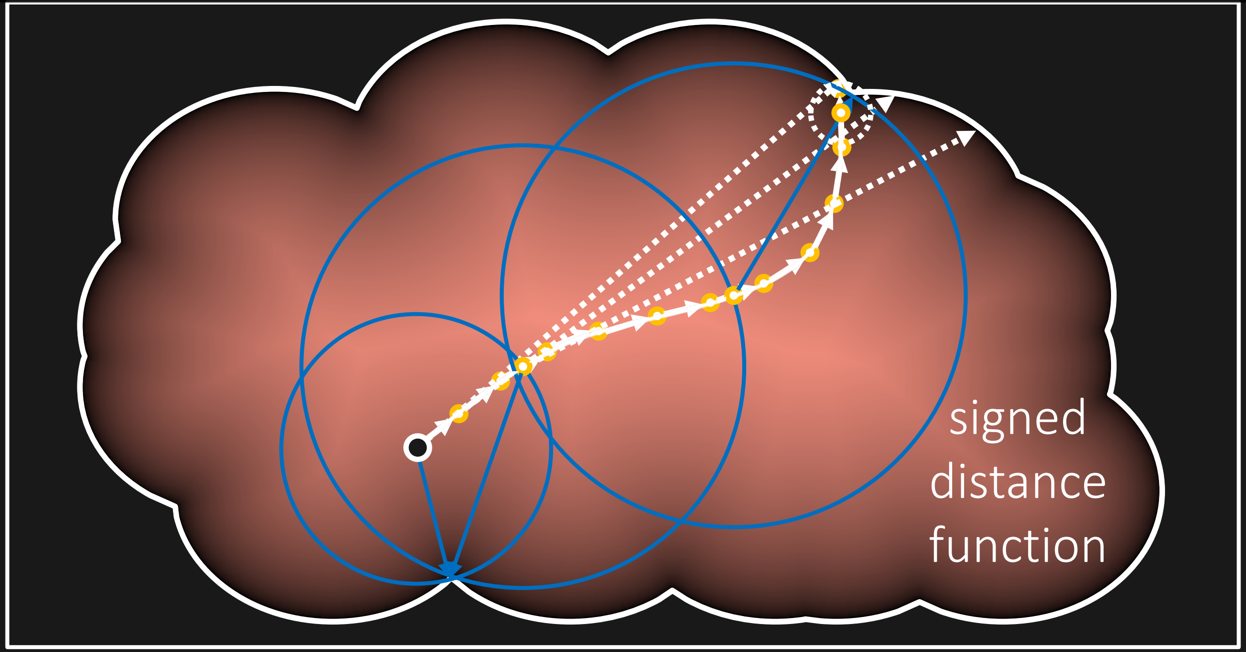
Zeltner et al. “Specular manifold sampling for rendering high-frequency caustics and glints”, TOG 2020



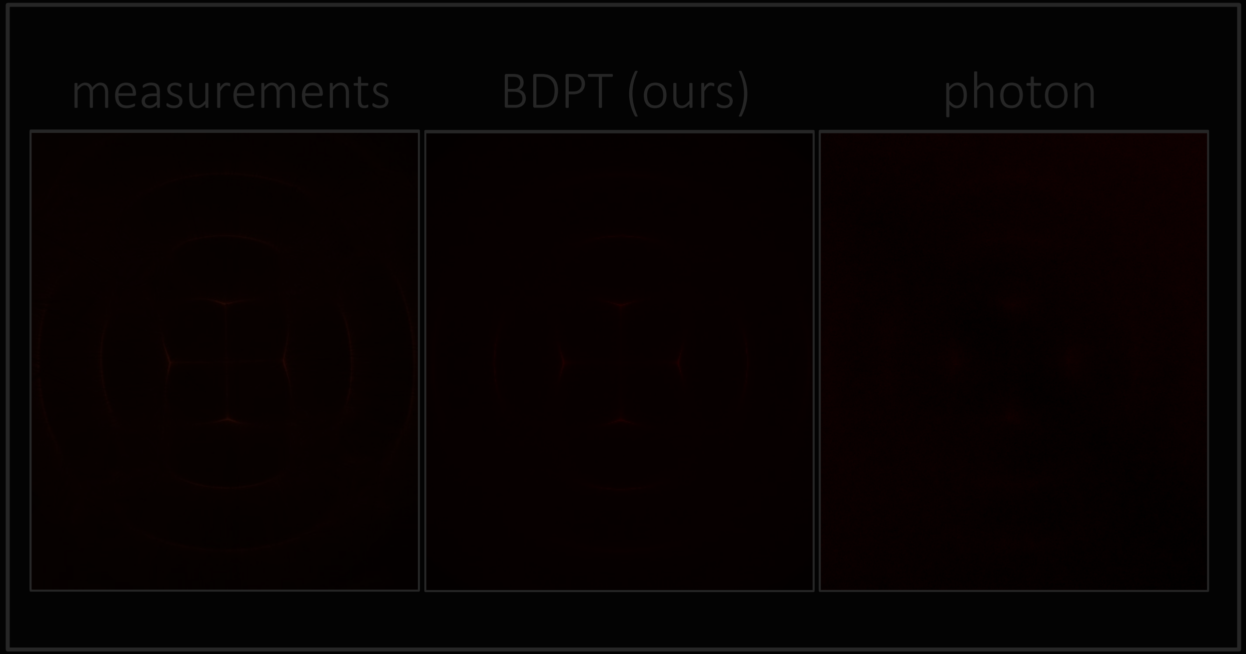
1. background on refractive radiative transfer equation



2. direct connections: our solution to unbiased rendering



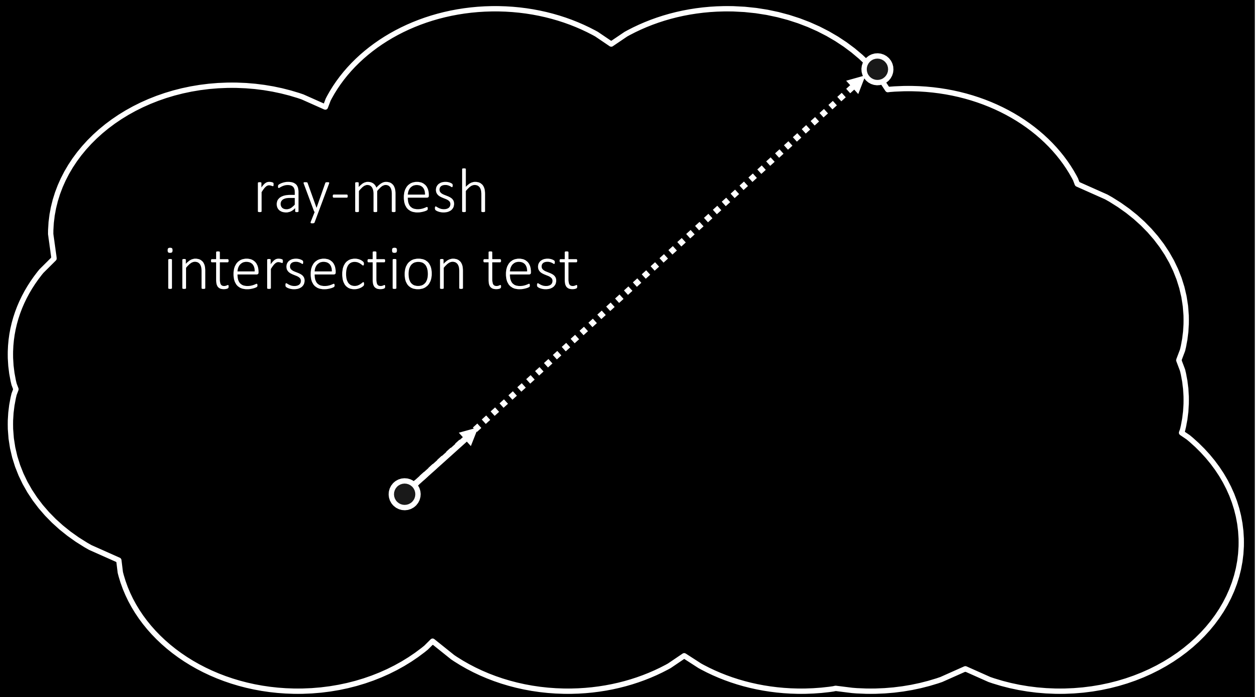
3. acceleration techniques



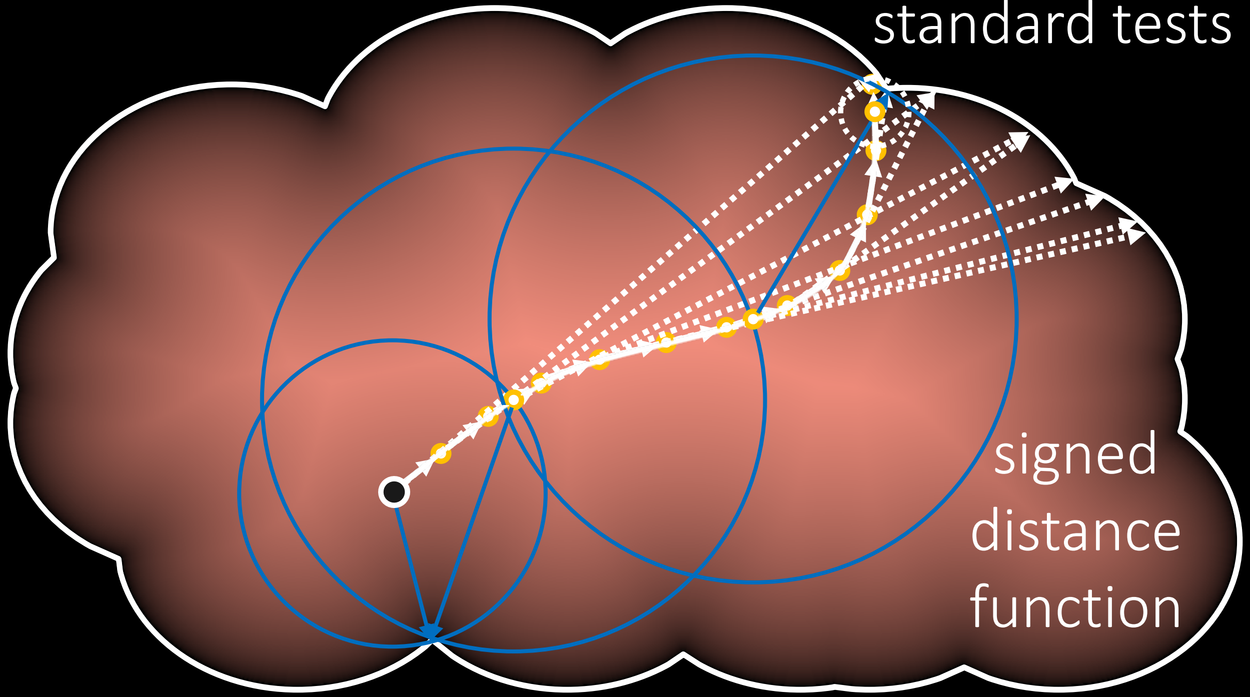
4. experiments

acceleration: sphere tracing

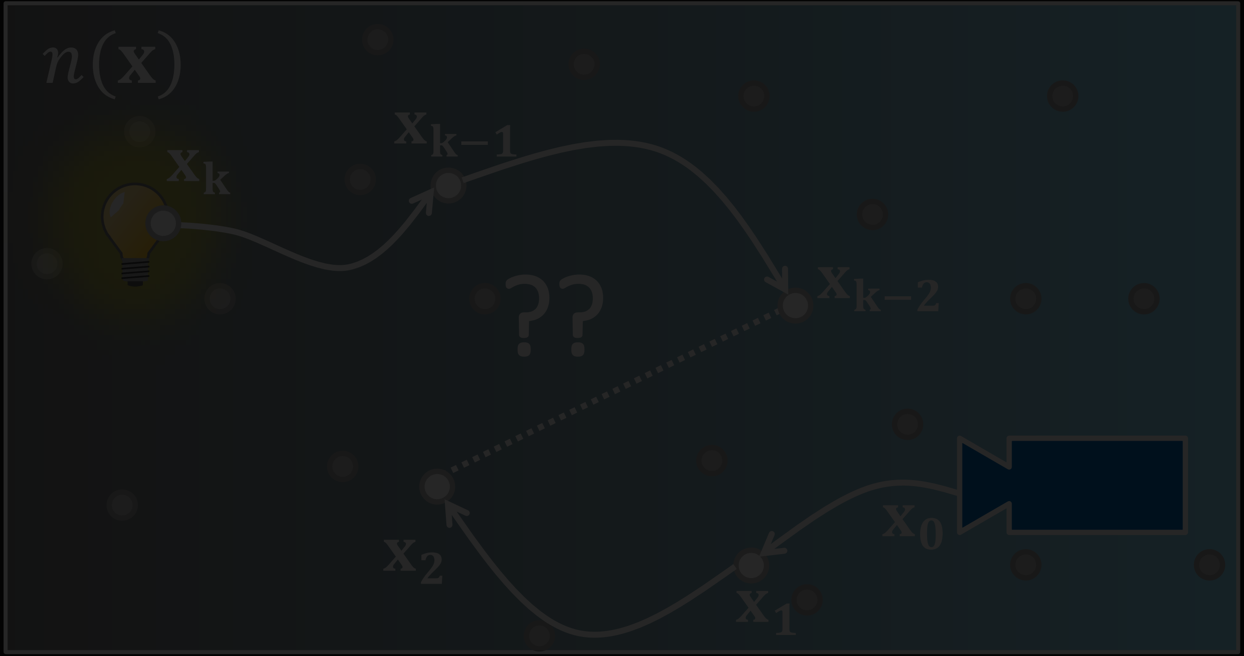
standard ray tracing



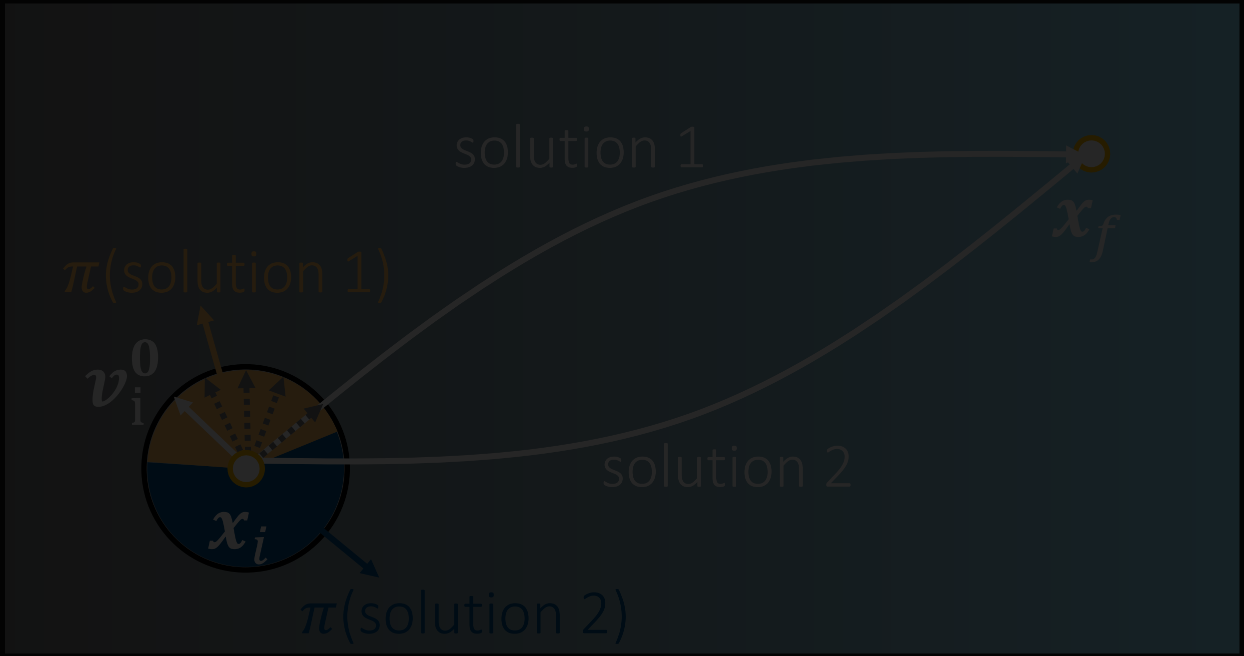
refractive ray tracing



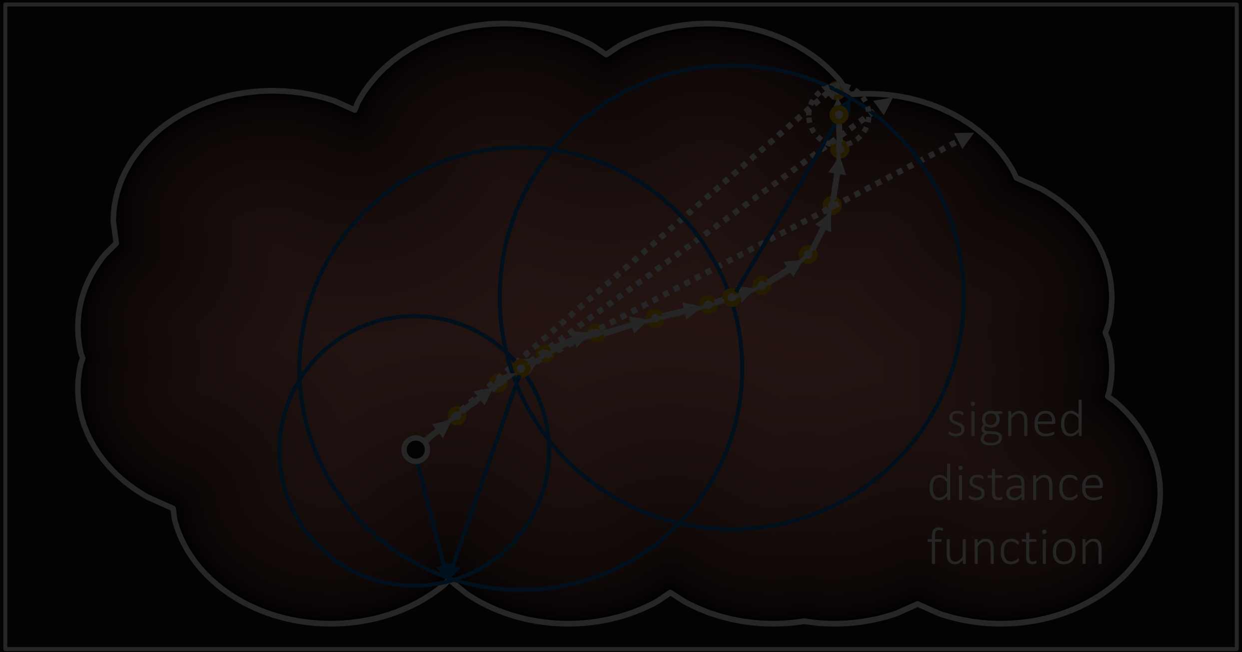
does not introduce bias



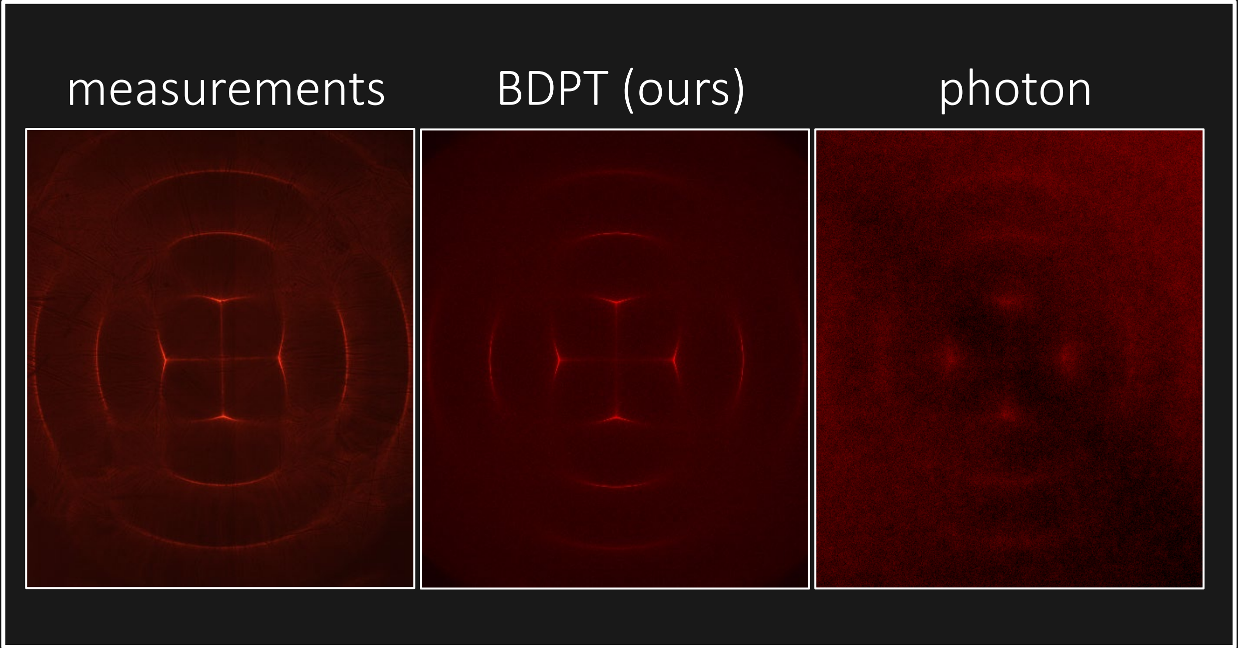
1. background on refractive radiative transfer equation



2. direct connections: our solution to unbiased rendering

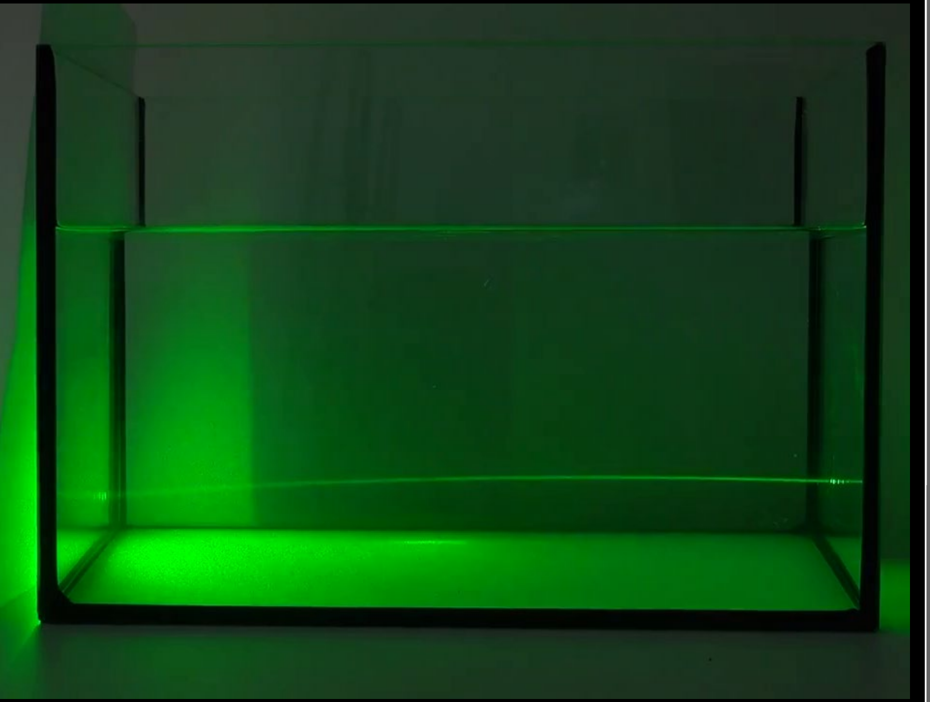


3. acceleration techniques

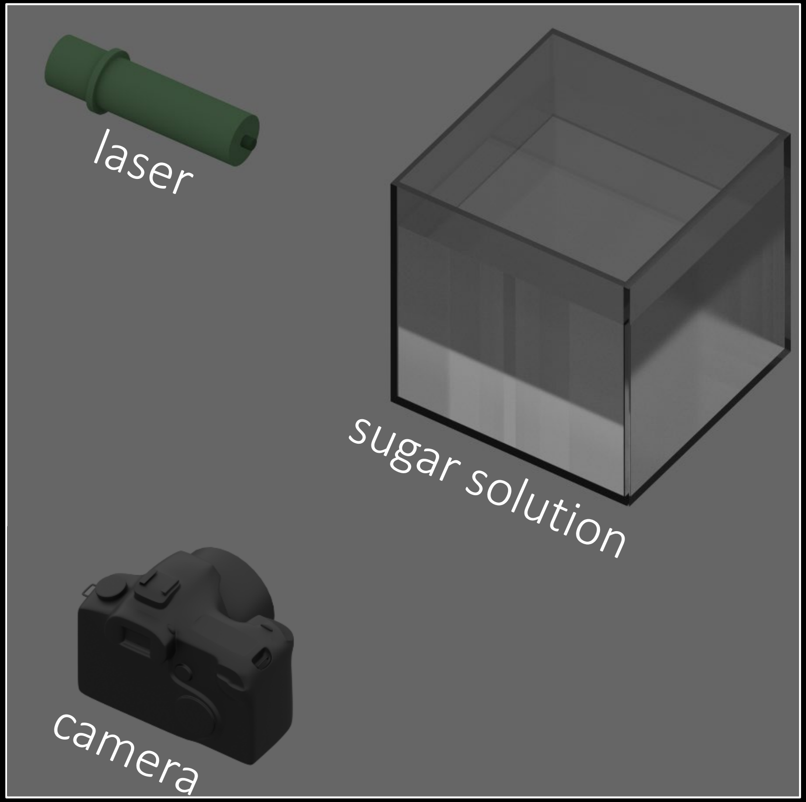


4. experiments

continuously refractive media and scattering



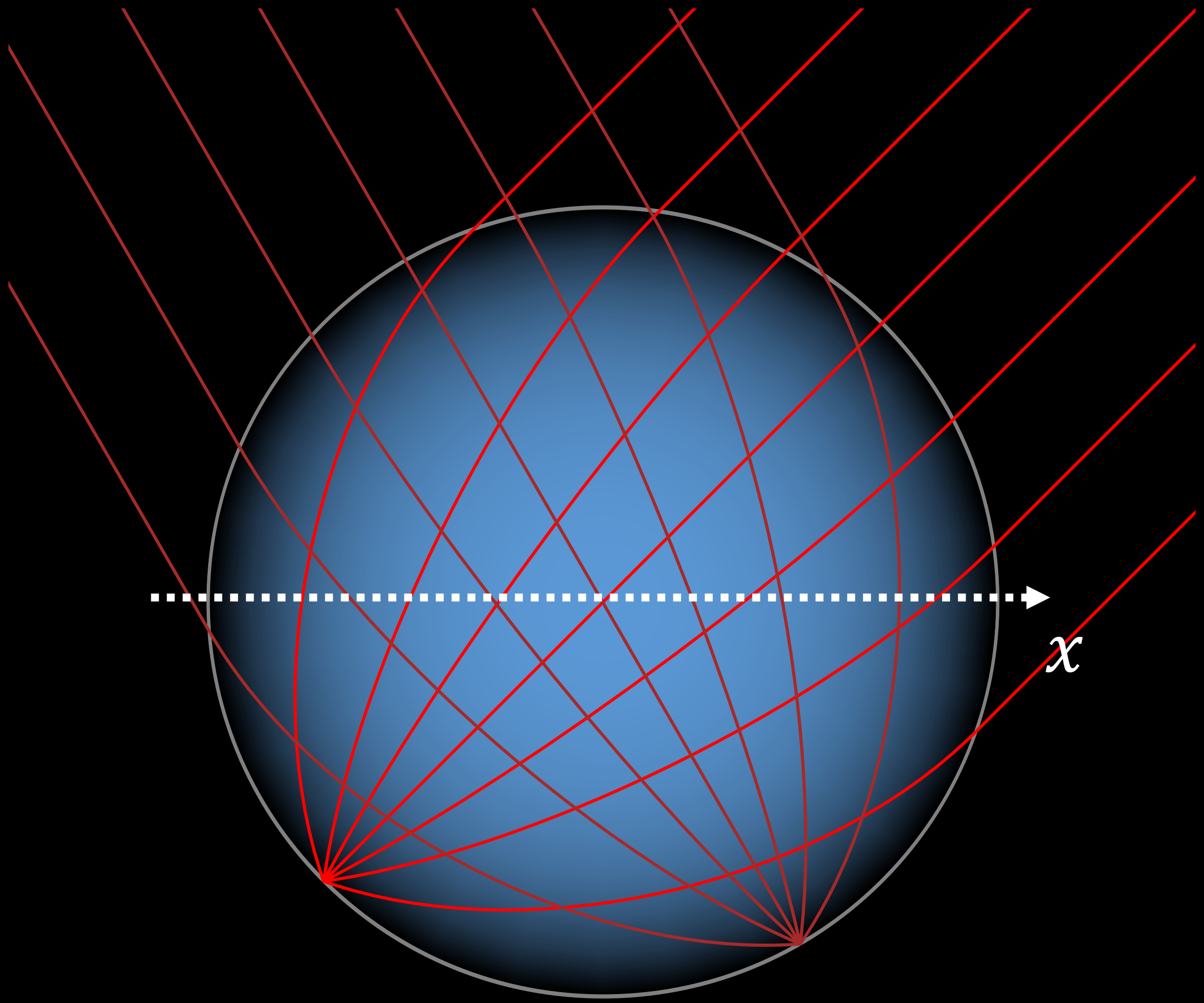
real scene

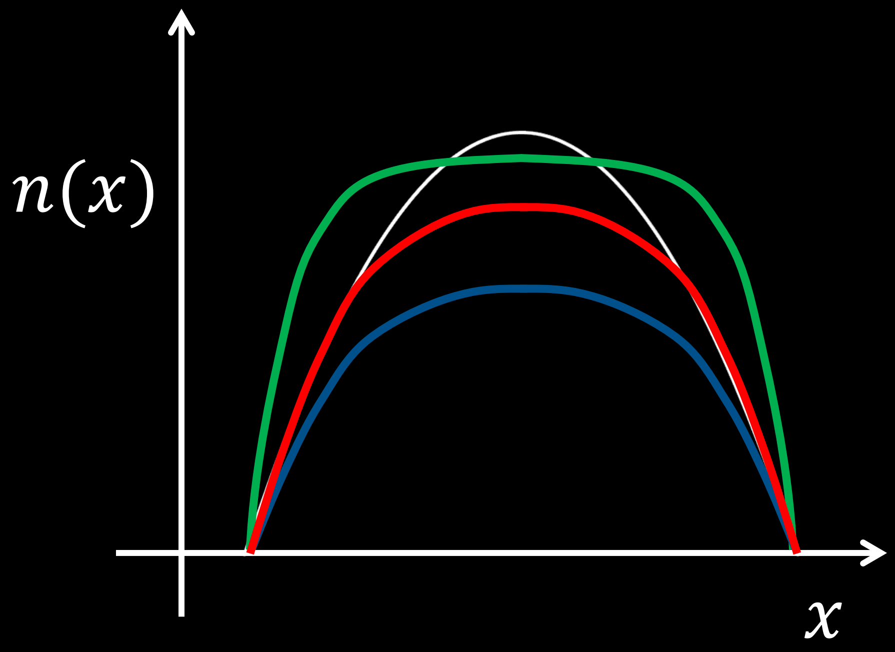


set up

light propagation

Luneburg lenses

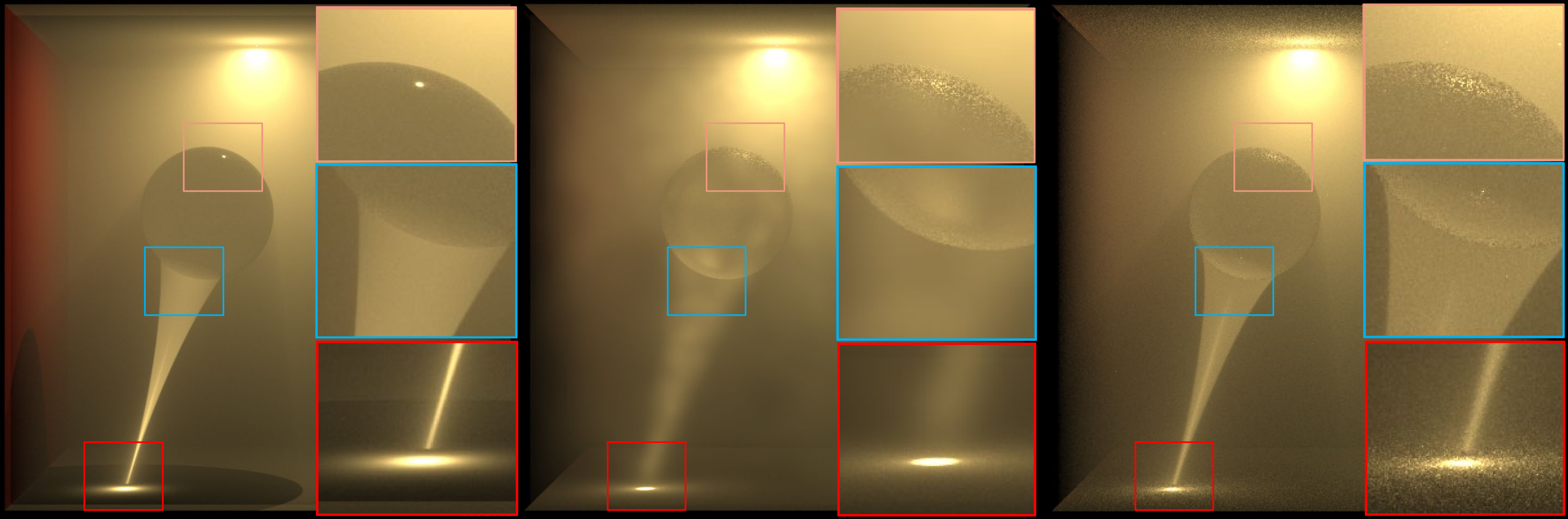




https://en.wikipedia.org/wiki/Luneburg_lens


Luneburg lenses
power equal to standard lens
rendering time: 10 mins

comparison with photon mapping



BDPT (ours)

photon mapping
(default parameters)

photon mapping
(optimized parameters)

BDPT is 5x faster than photon mapping

rendering time: 10 min

transient rendering (videos)

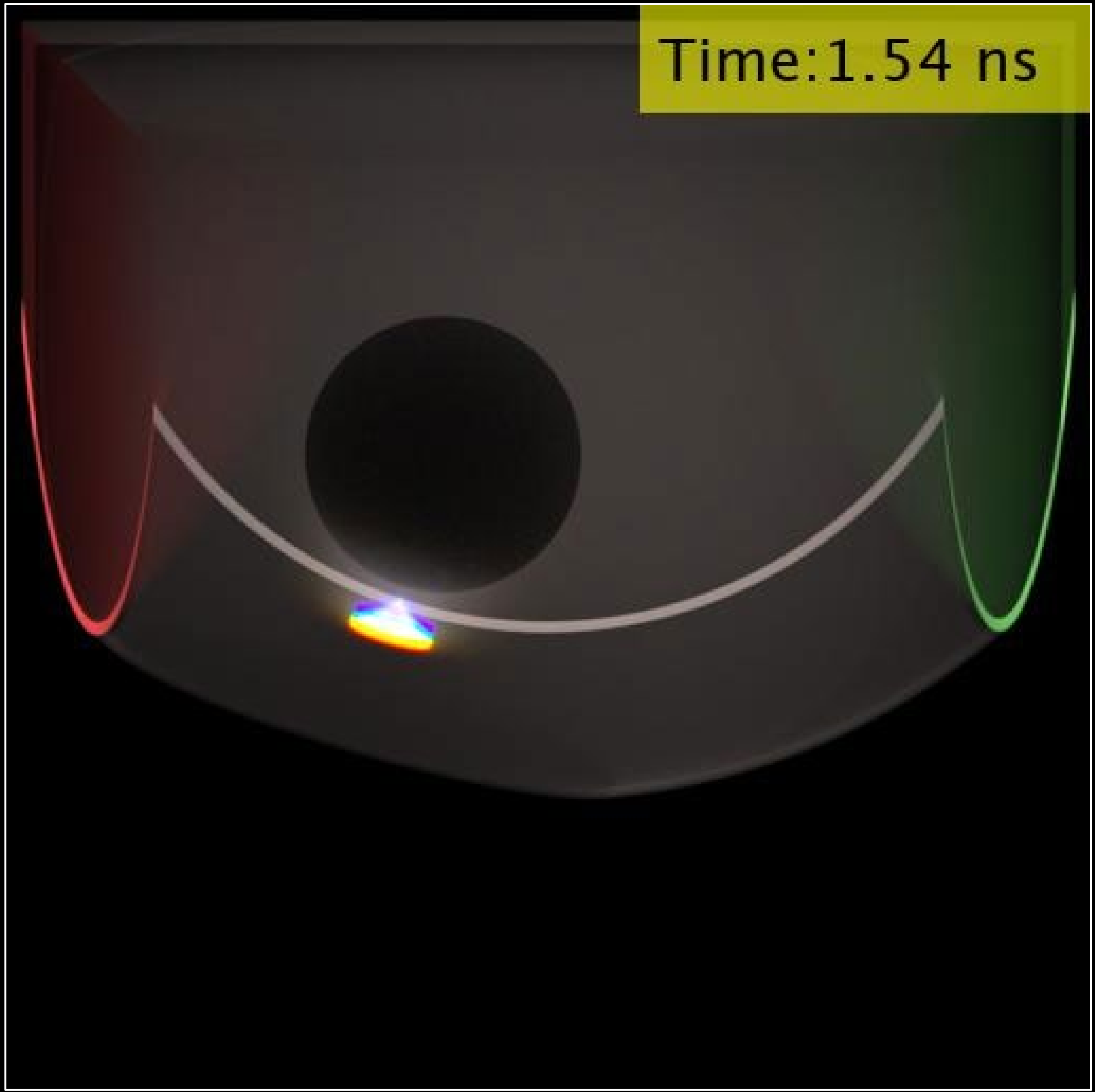
constant refractive index

continuous refractive index

time:
urs

transient rendering

constant refractive index



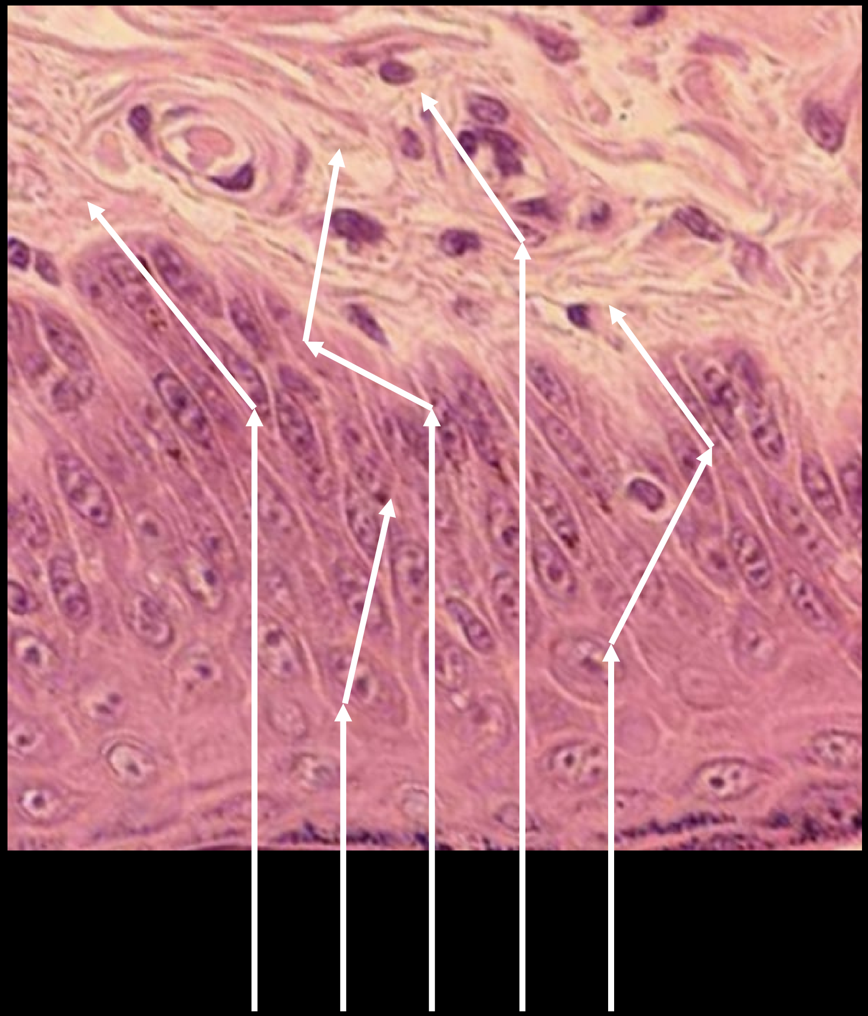
continuous refractive index

transient rendering

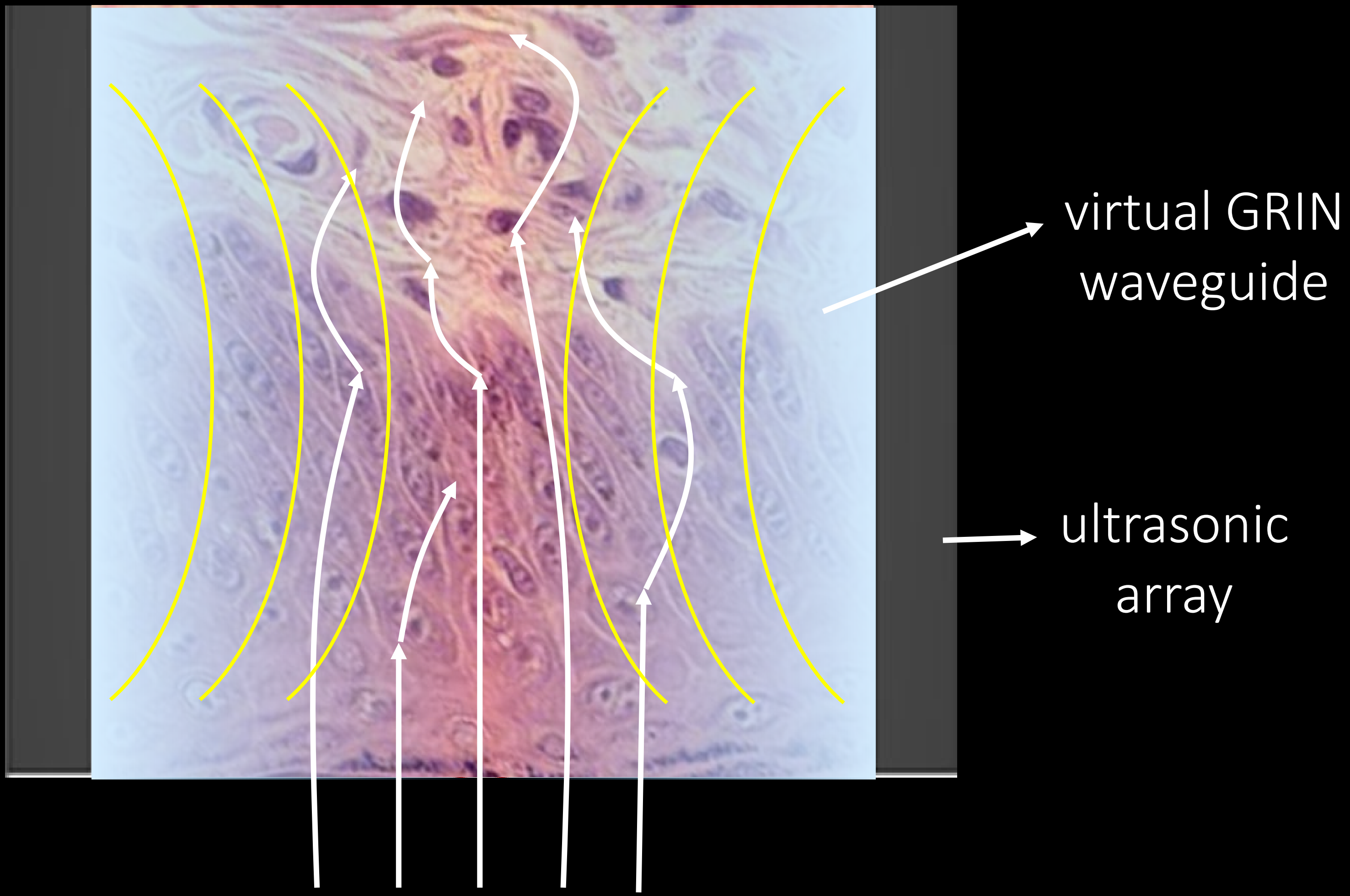
constant refractive index

continuous refractive index

virtual ultrasonic waveguides

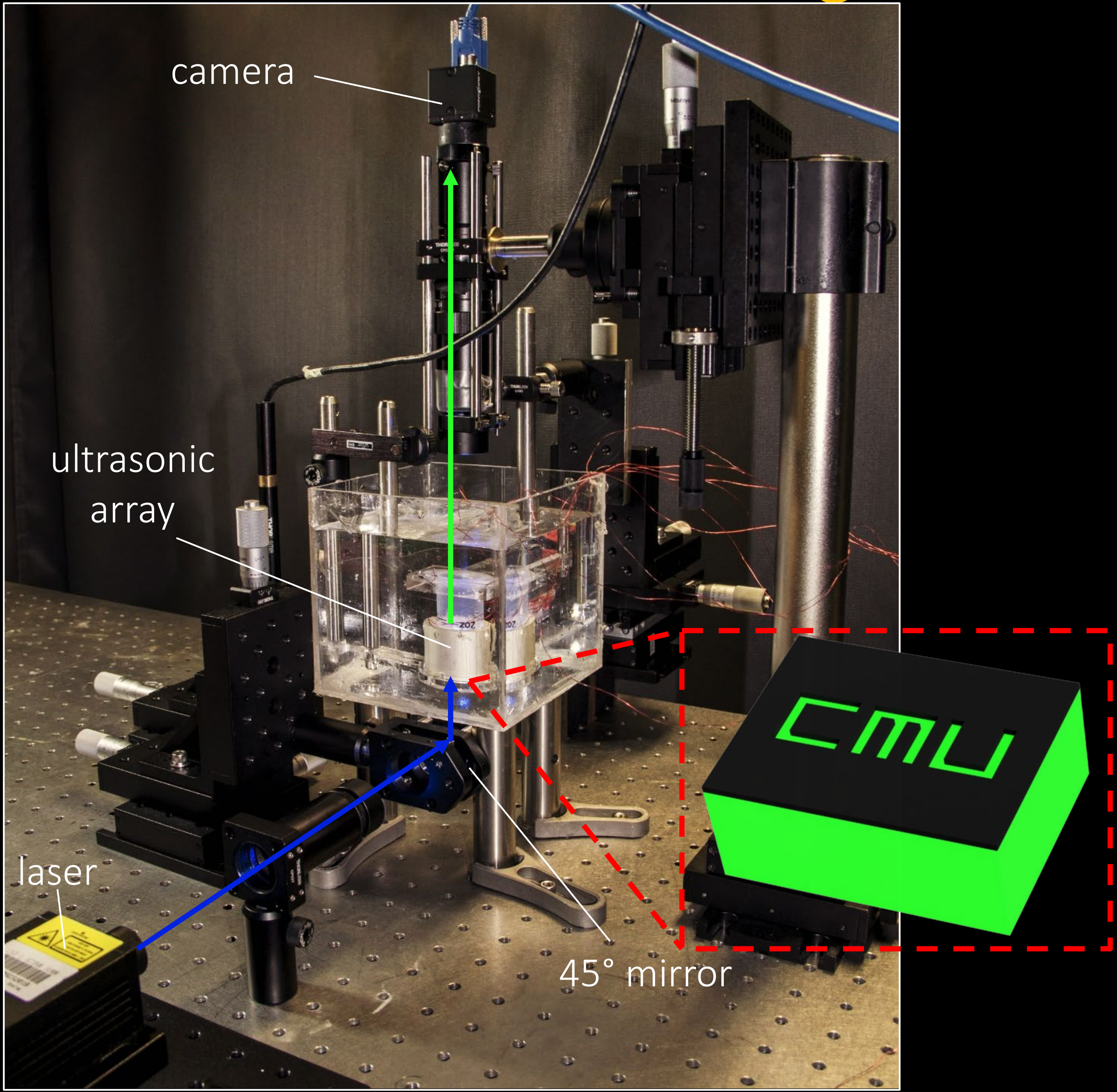


virtual ultrasonic waveguides

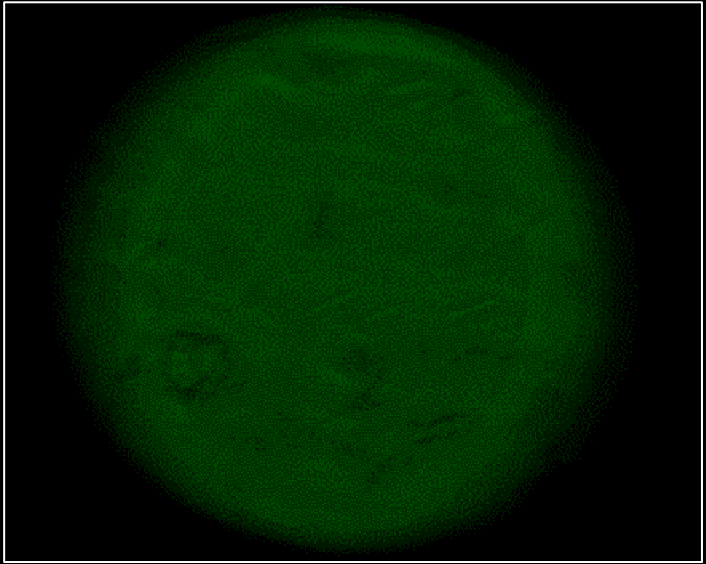


Chamanzar et al. “Ultrasonic sculpting of virtual optical waveguides in tissue”. Nature communications, 2019
Scopelliti et al. “Ultrasonically sculpted virtual relay lens for in situ microimaging”. Light: Science and Applications, 2019
Karimi et al. “In situ 3D reconfigurable ultrasonically sculpted optical beam paths”. Optics express, 2019

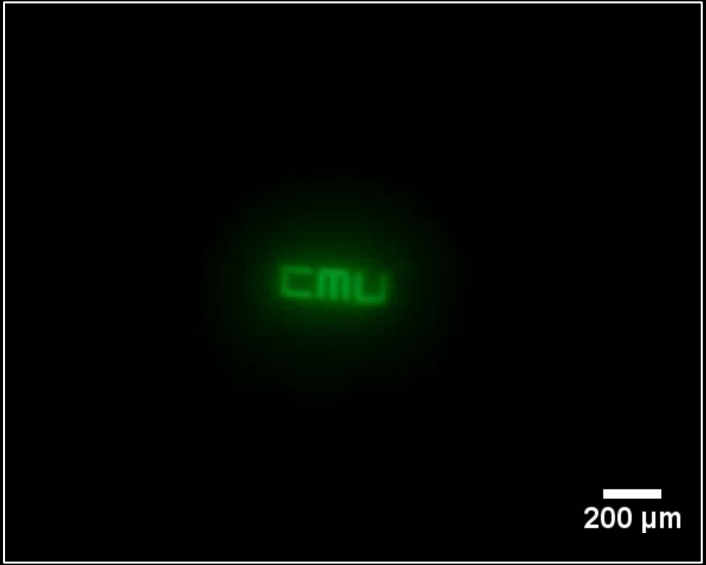
virtual ultrasonic waveguides



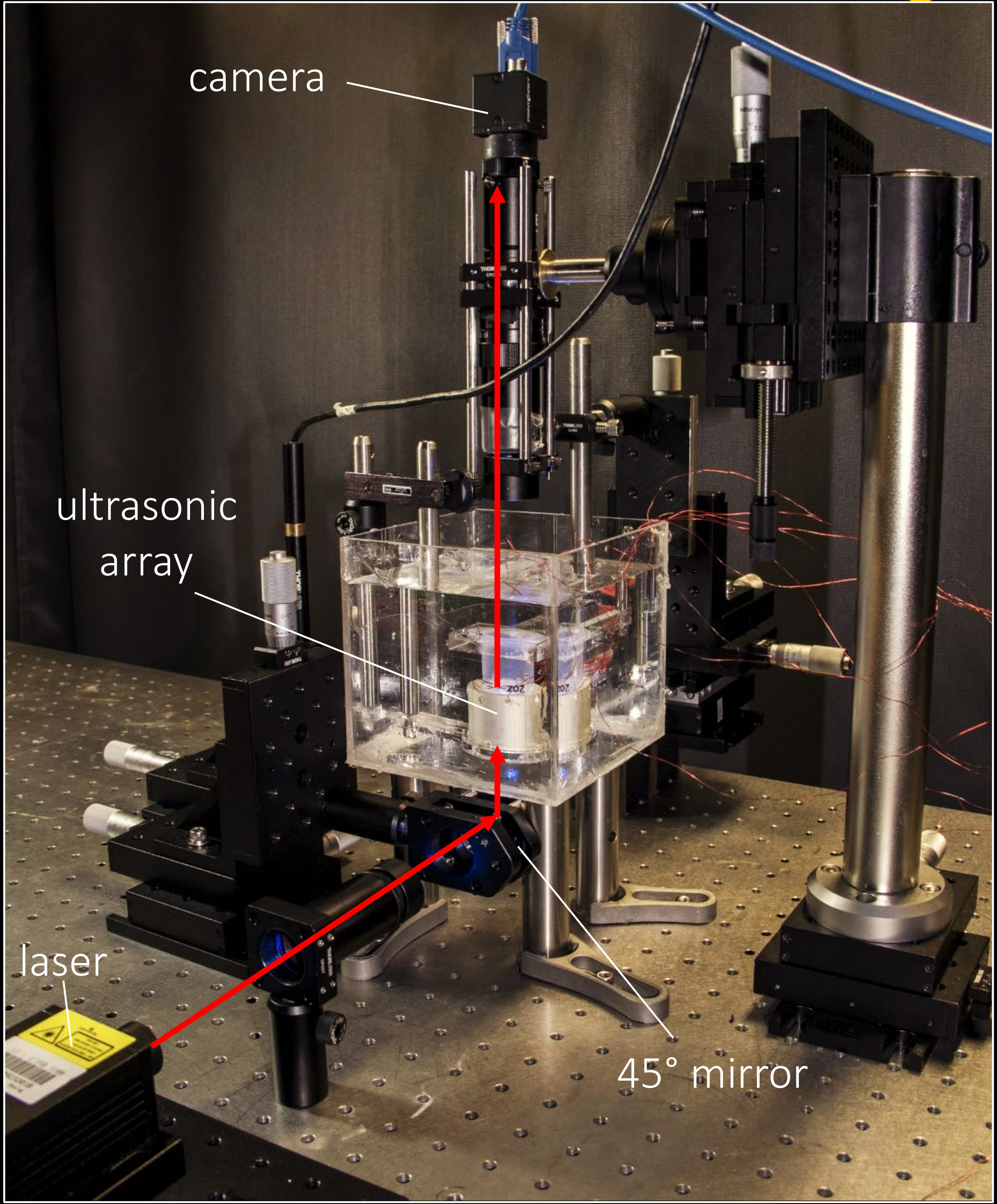
no waveguide

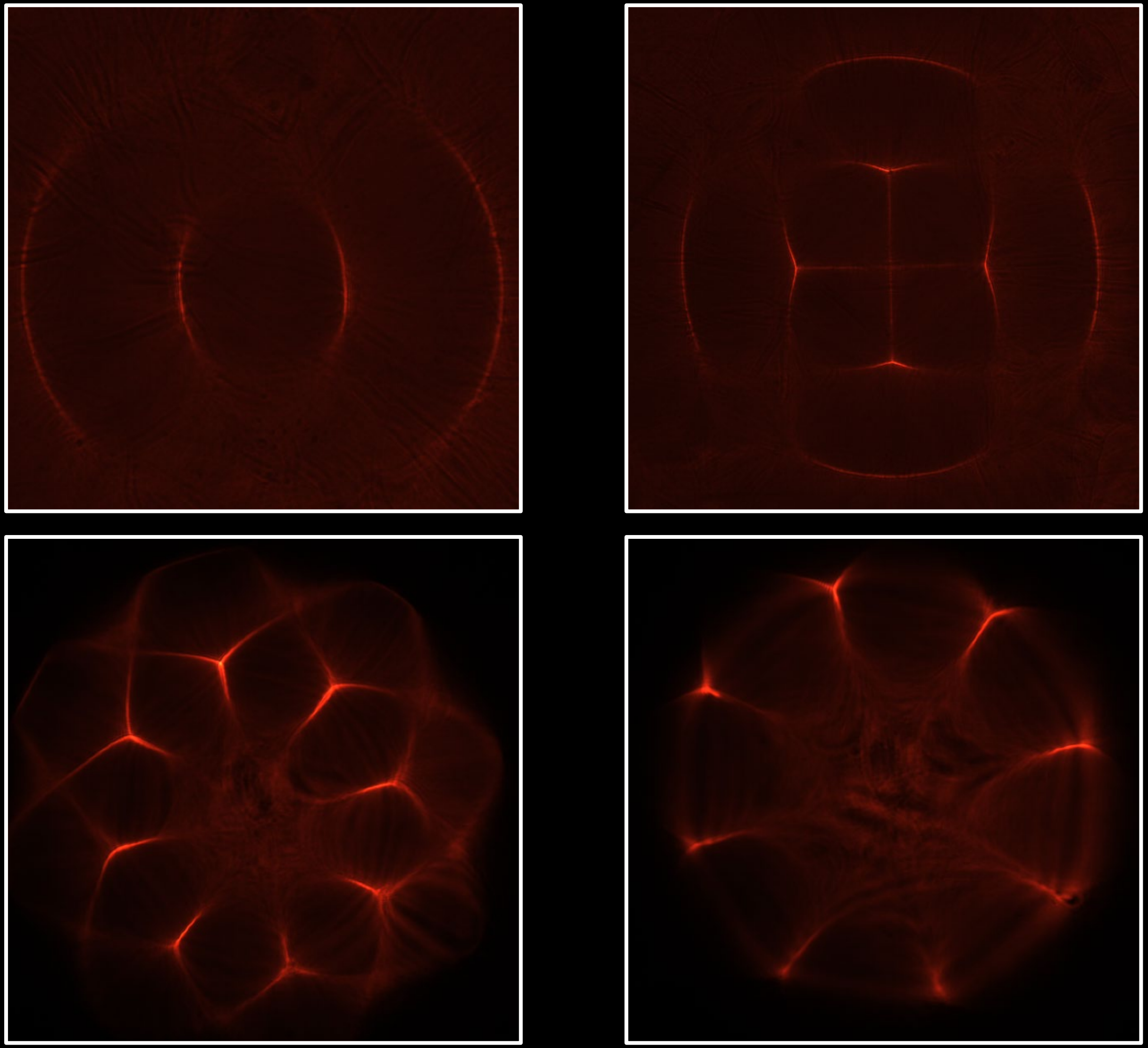


virtual waveguide



virtual ultrasonic waveguides

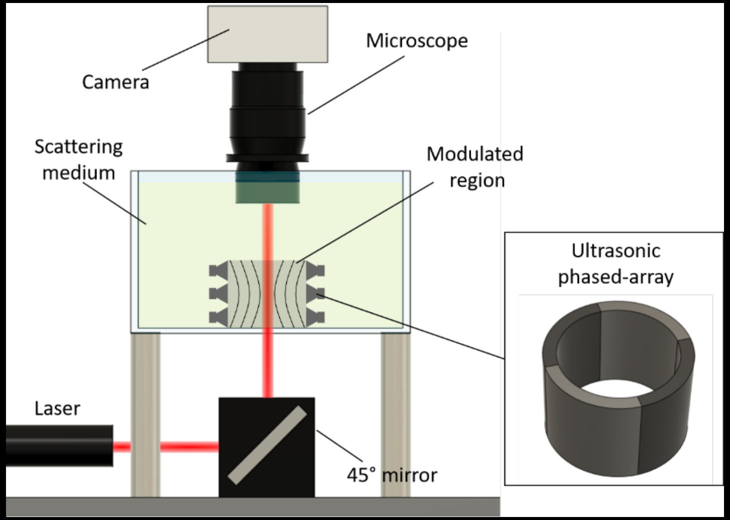




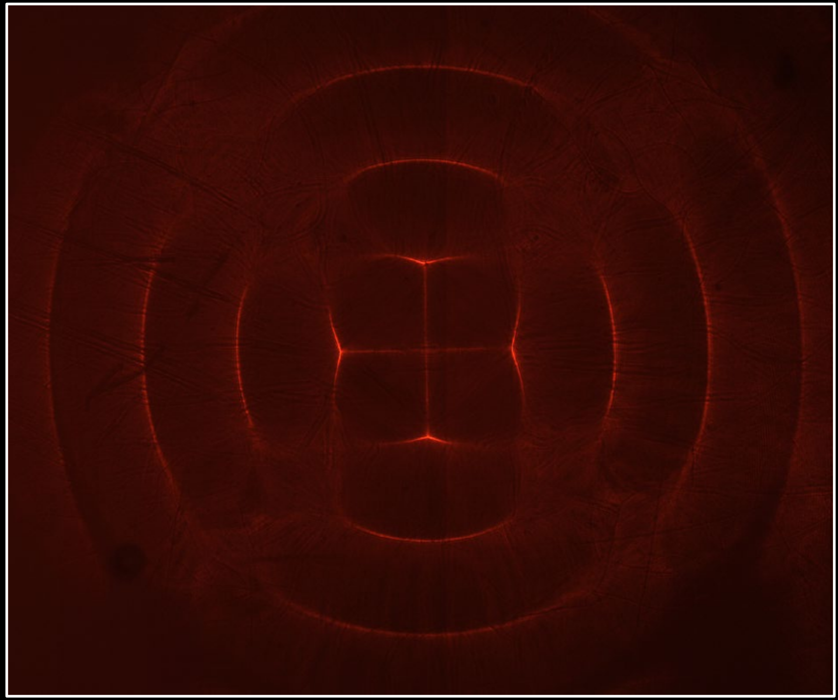
Karimi et al. “In situ 3D reconfigurable ultrasonically sculpted optical beam paths”. Optics express, 2019

Rendering acousto-optics

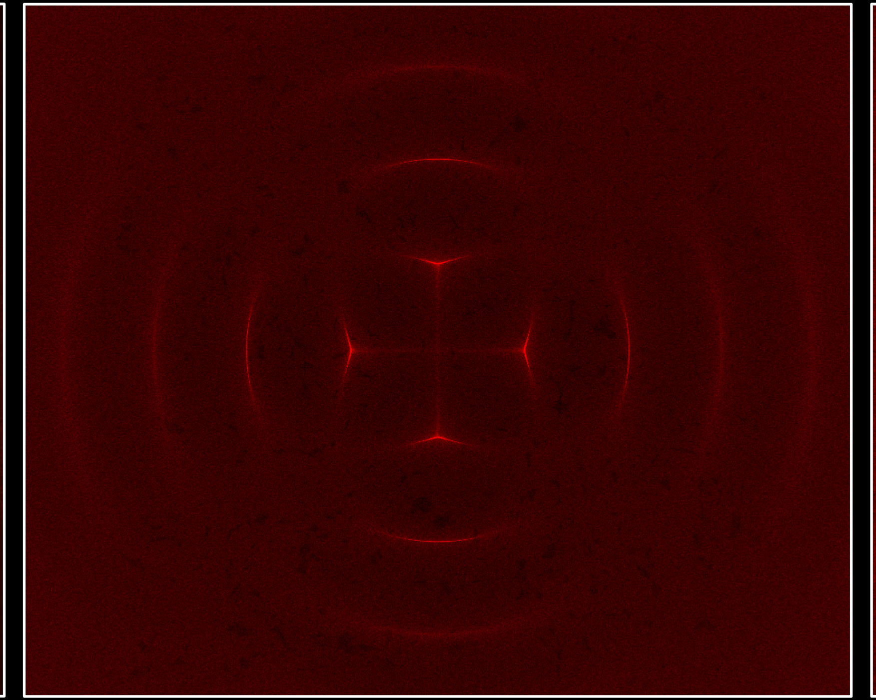
setup for ultrasonic lensing in scattering



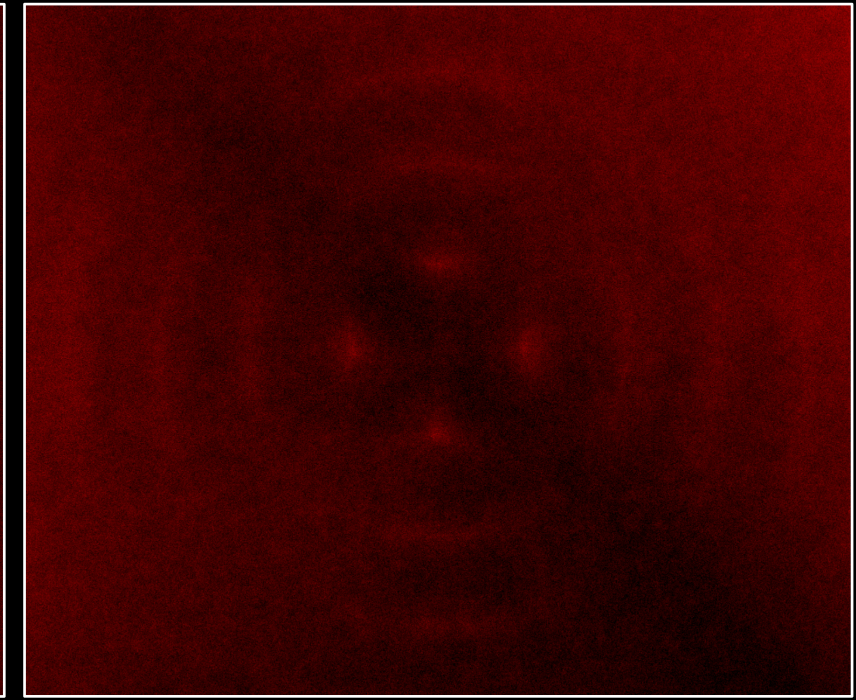
real capture



our algorithm

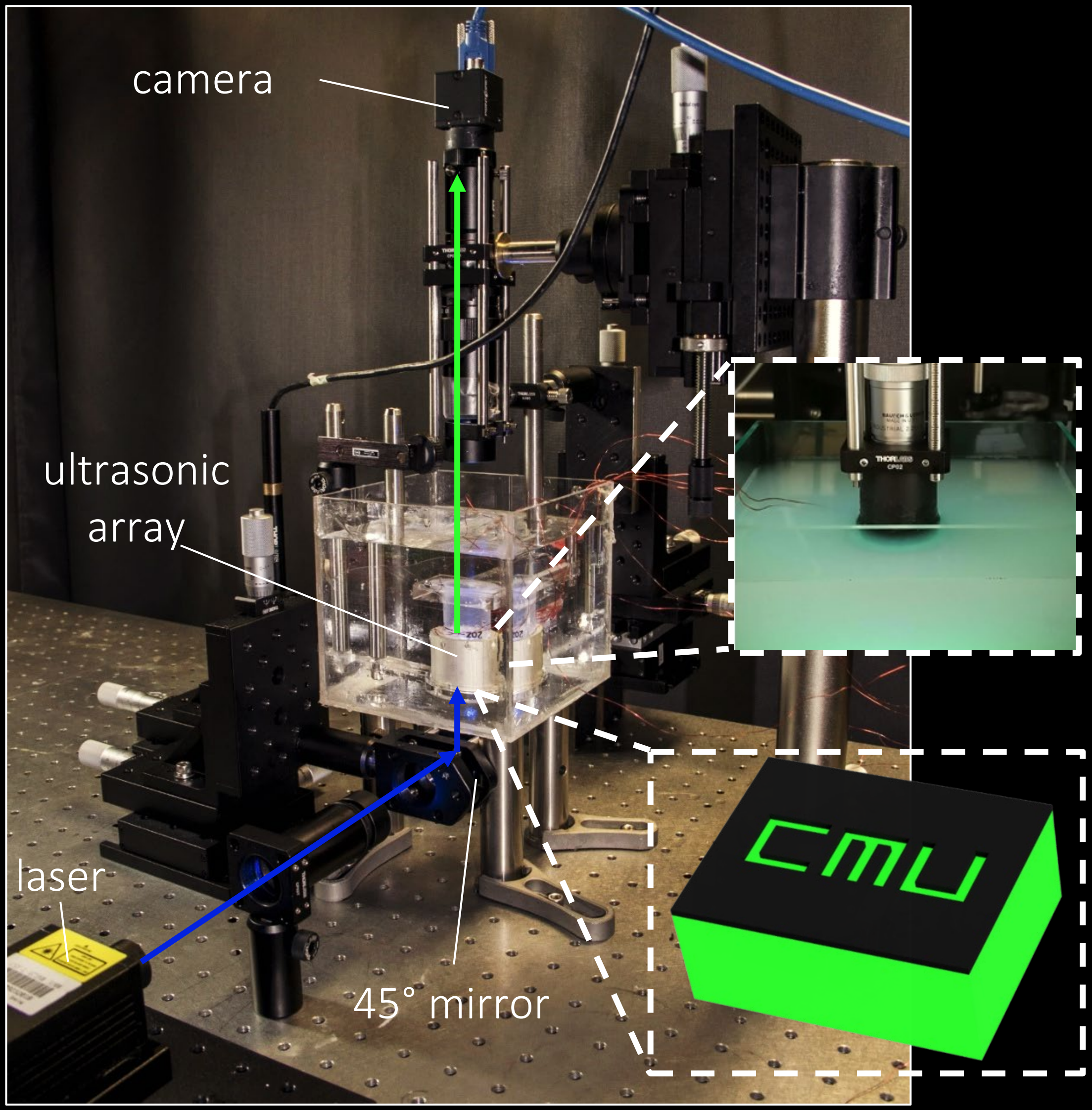


previous algorithm



[Pediredla et al. Transactions on Graphics 2020]

Ultrasonic light guiding inside tissue



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

- ultrasound frequency
- ultrasound voltage
- shape of waveguides
- placement of transducers
- sensor size
- and more…

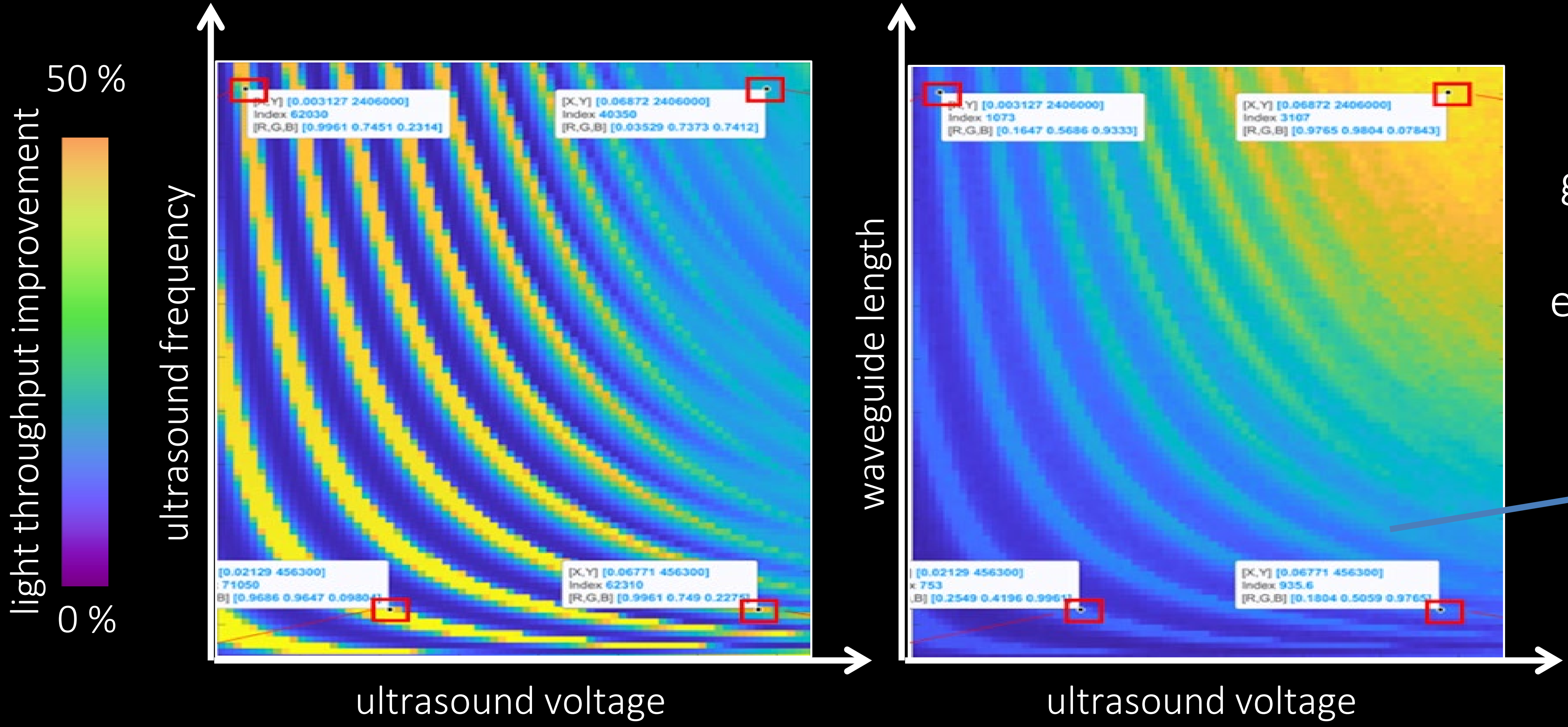
Guiding performance strongly affected by different parameter values

Painstaking experiments:

- several hours of work to test one set of parameter values

Optimizing ultrasonic GRIN waveguides

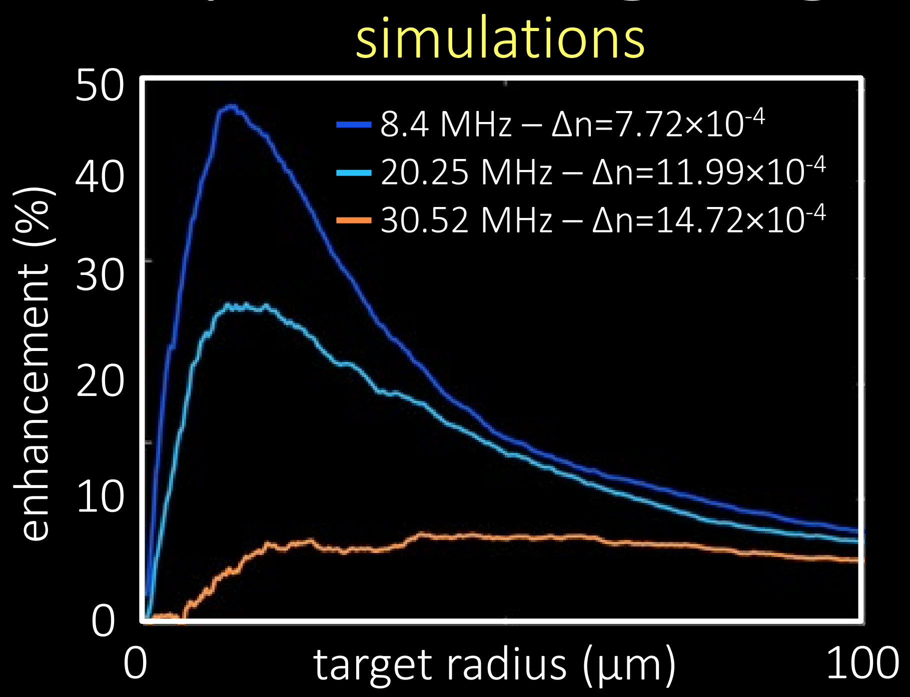
- Hundreds of thousands of virtual experiments.

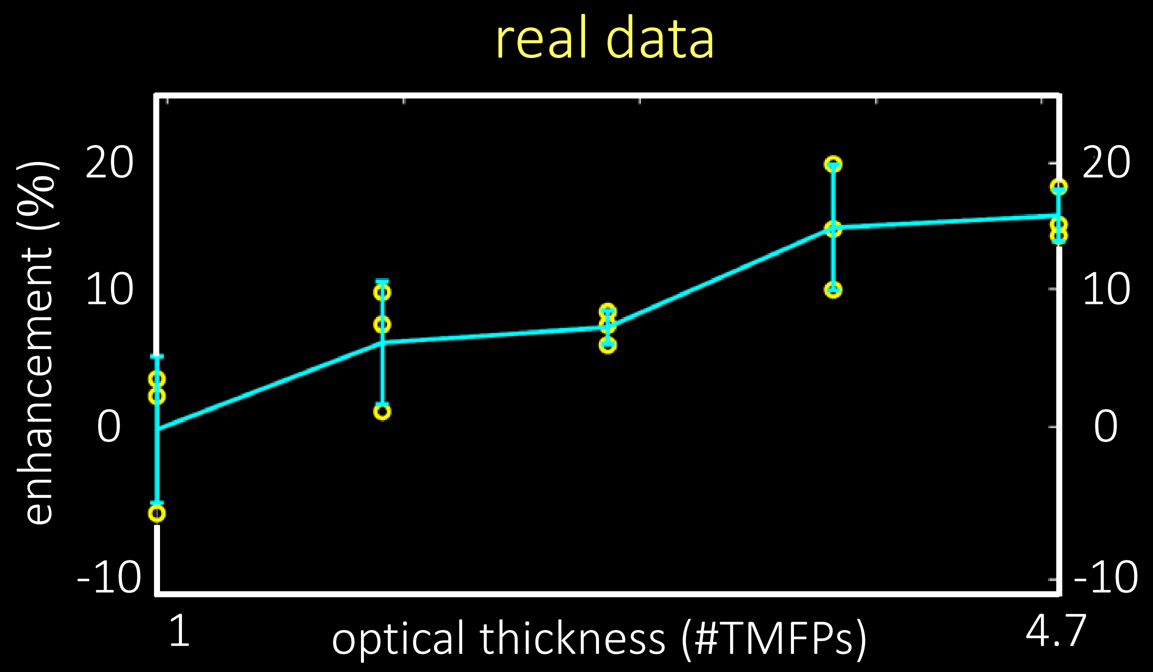


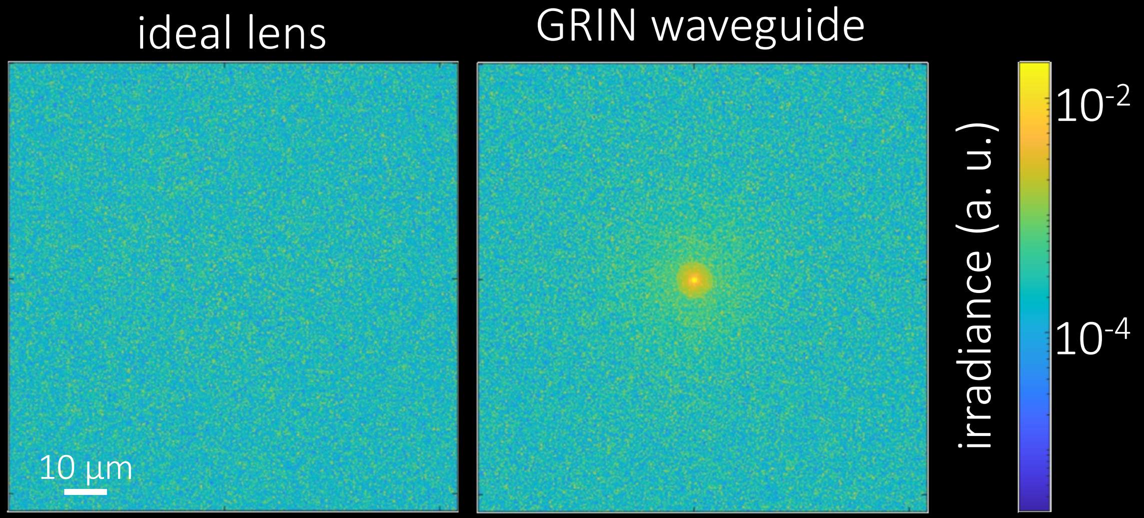
Each dot on these graphs would have been a real experiment taking a PhD student a full day's work

[Pediredla et al., submitted to Nature Communications 2021]

Improved light guiding in human bladder







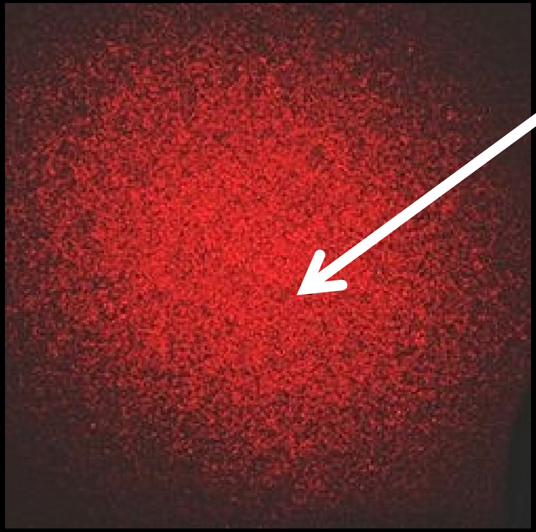
Improved light guiding performance by

- 200% compared to unoptimized waveguides
- 50% compared to external optics

Simulation predictions verified experimentally

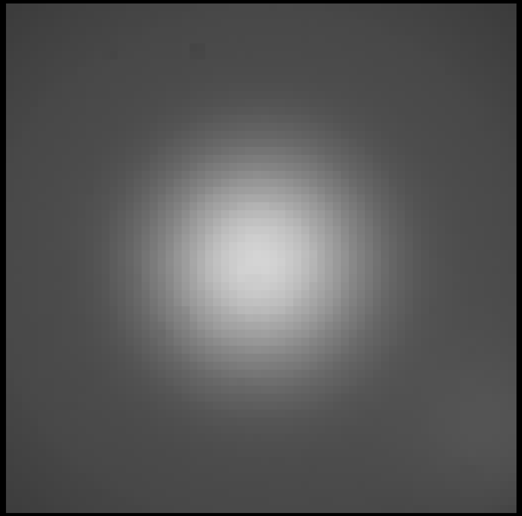
[Pediredla et al., submitted to Nature Communications 2021]

Speckle and memory effect

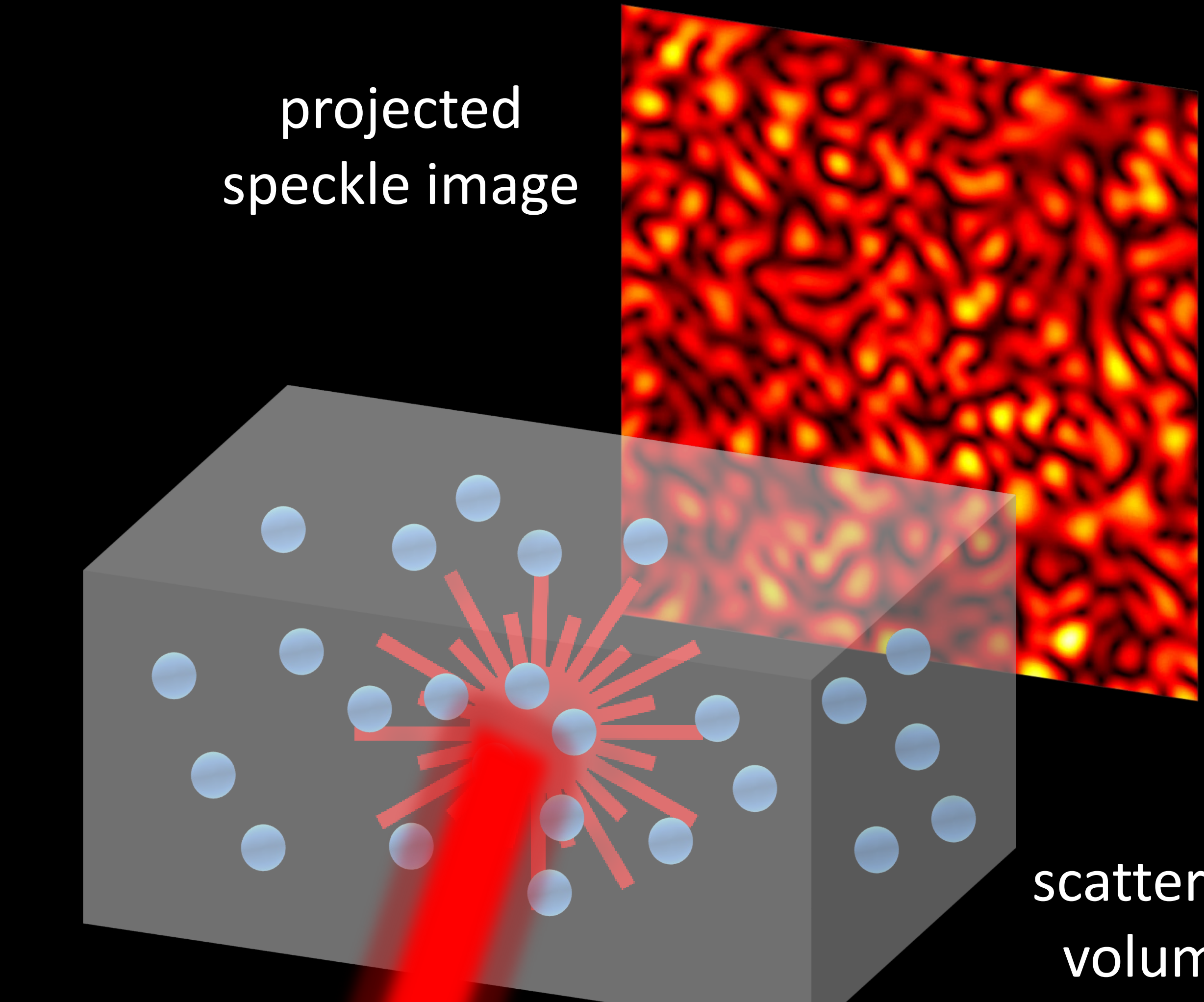


speckle: noise-like pattern

what real laser images look like

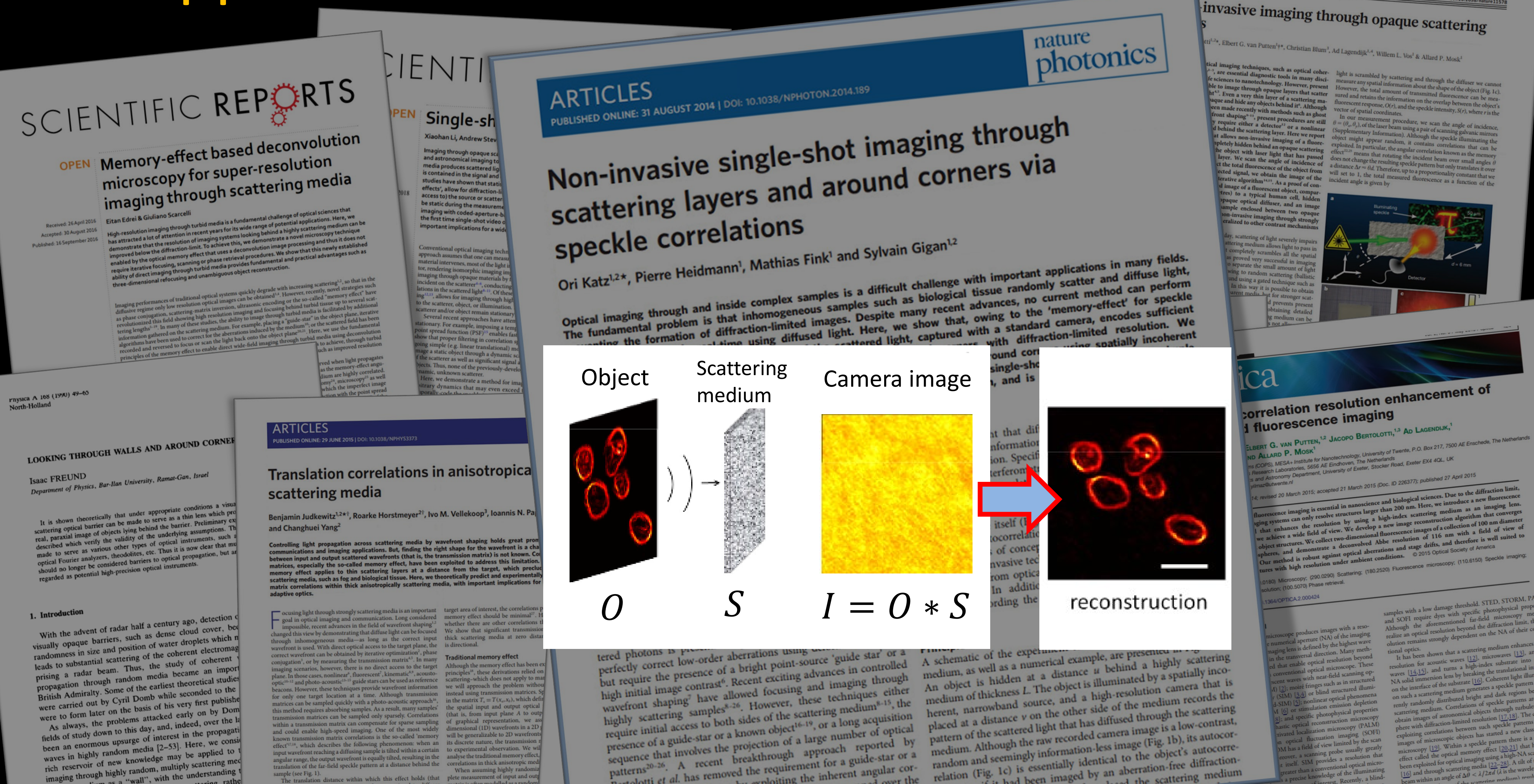


what standard rendered images look like



scattering volume

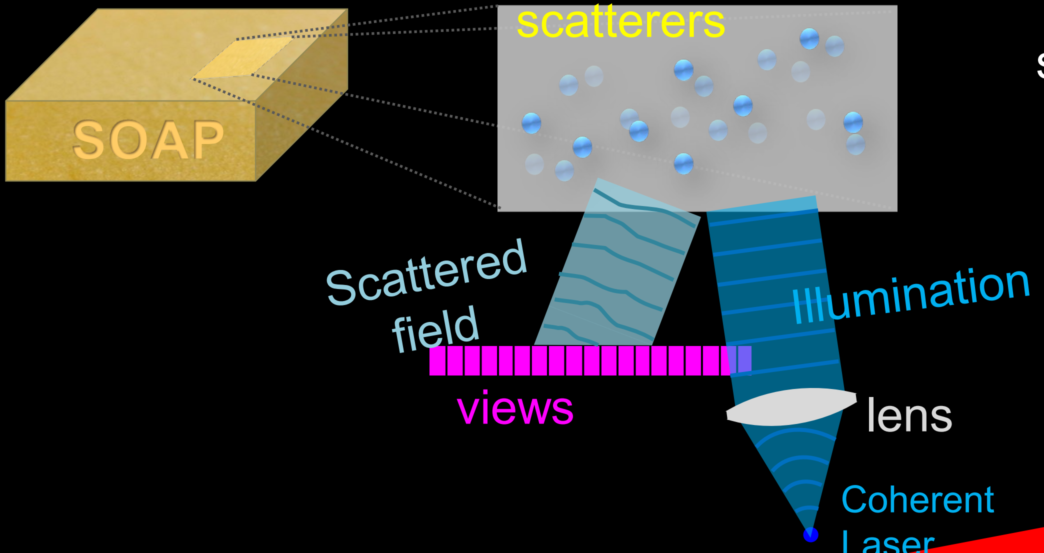
Applications and Related Work



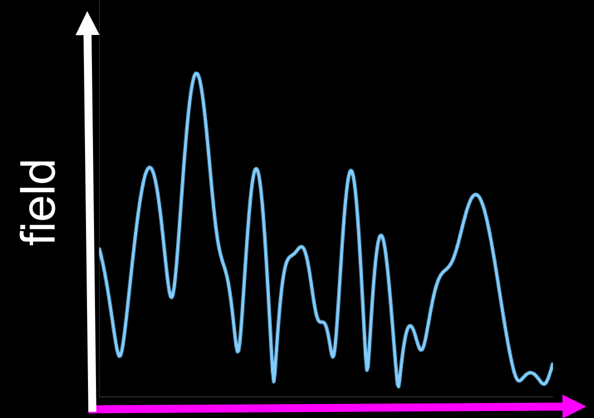
Simulating speckles

inefficient → Specify exact (sub-wavelength) position of scatterers

⇨ In graphics we describe materials by ***statistical*** bulk parameters, as the ***density*** of scatterers



solve wave equations



Wave equation solvers

- Differential equation F
- Integral equation (e.g.,

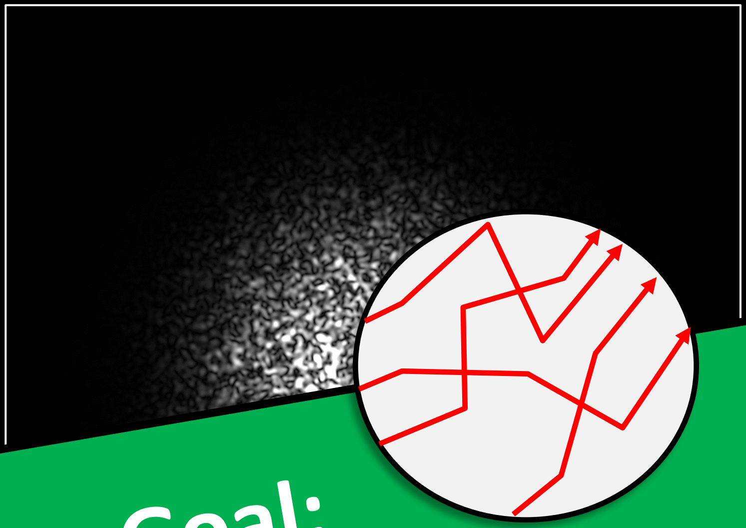
Slow

Practical only for tiny or optically thin media

For simplicity: **Flatland**
Scattering medium is **2D**
Sensor is **1D**
Speckle pattern is **1D**

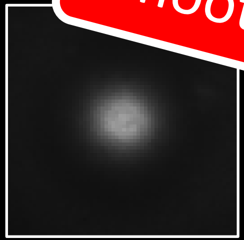
Monte Carlo (MC) Simulation of Speckles

MC Advantage:
1. **Fast**
2. input is scatterer **density** rather than exact scatterer locations

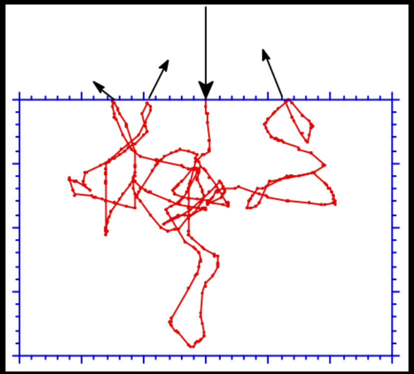


Our Goal:
Extend efficient MC tools to evaluate speckles and their coherent statistics

Standard intensity MC

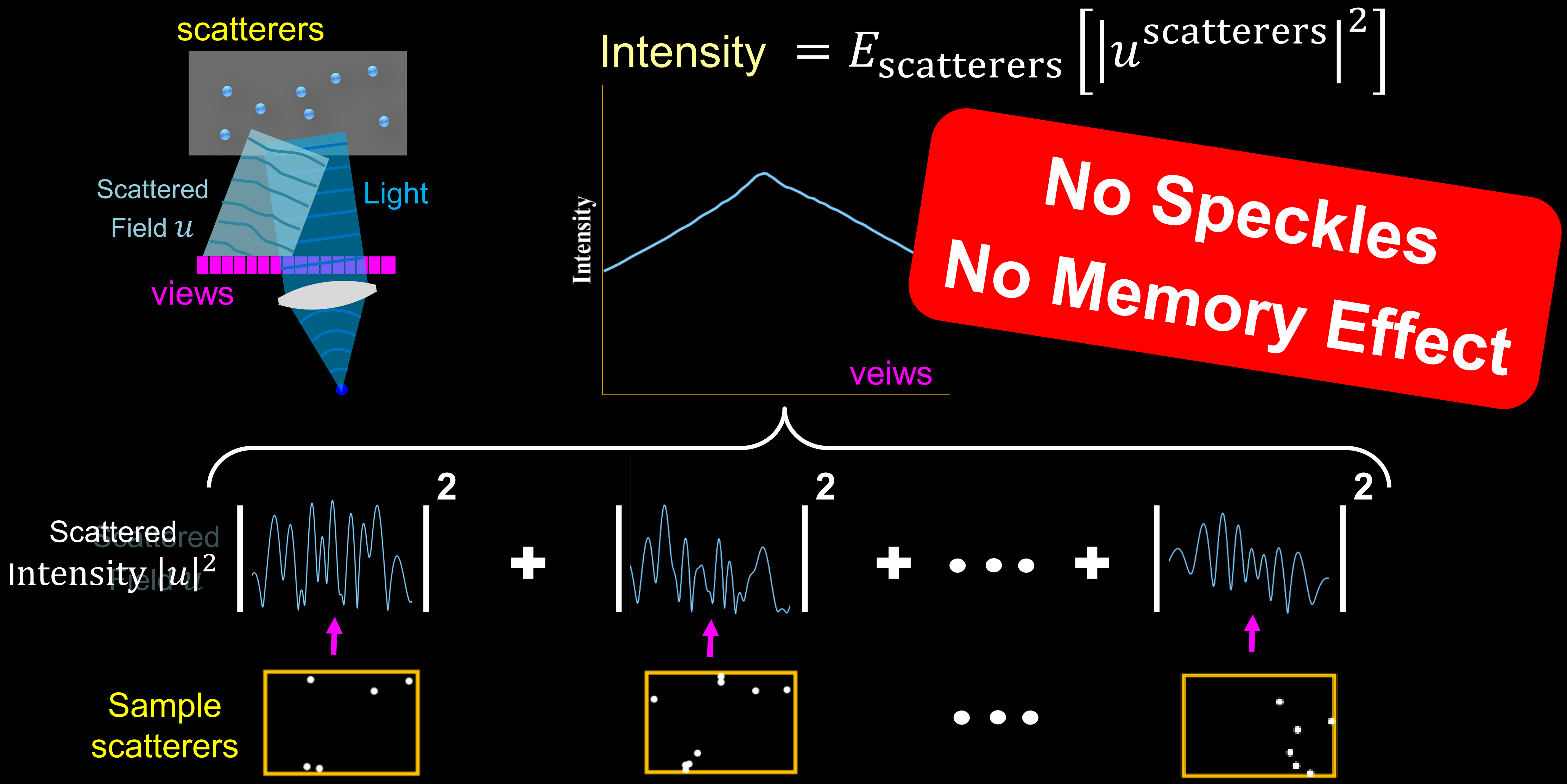


Smooth, no speckles



Monte Carlo Modeling of Light Transport in Multi-layered Tissues,
Wang & Jacques, 1992

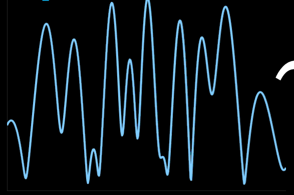
Wave Solution v.s. Monte Carlo



MC requires the scatterers density – no need for exact positions

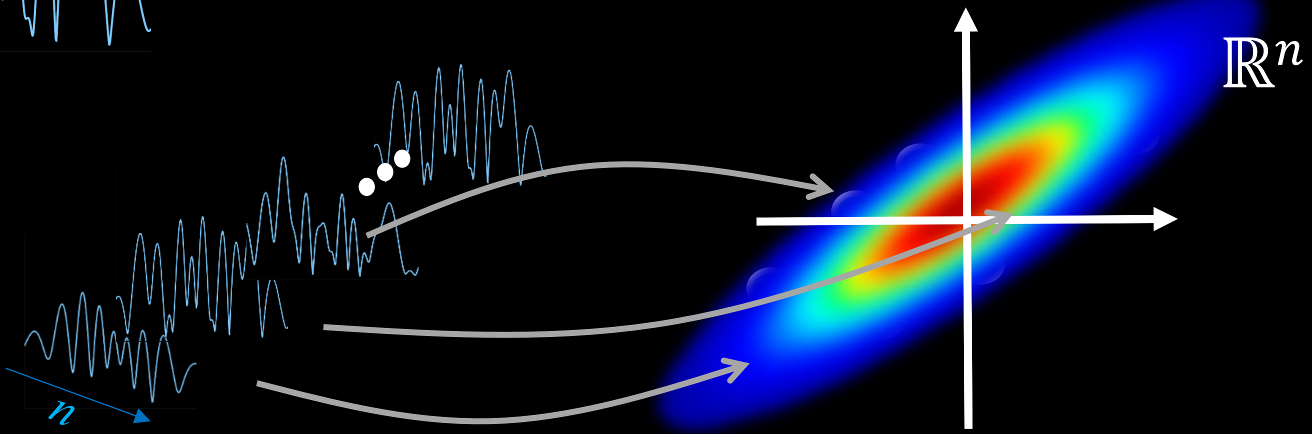
Speckle Statistics

Speckles



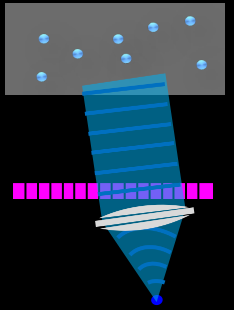
$\sim \mathcal{N}$ (Mean, Covariance)

Sufficient Statistics



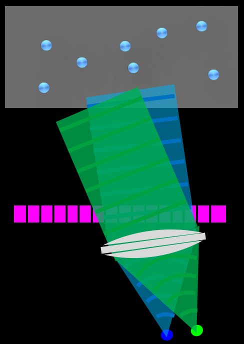
1st moment

Intensity Mean $= \mathrm{E}_{\text{scatterers}}\left[\left|u^{\text{scatterers}}\right|^2\right]$ ⇒ Incoherent Summation



Cross-Illumination

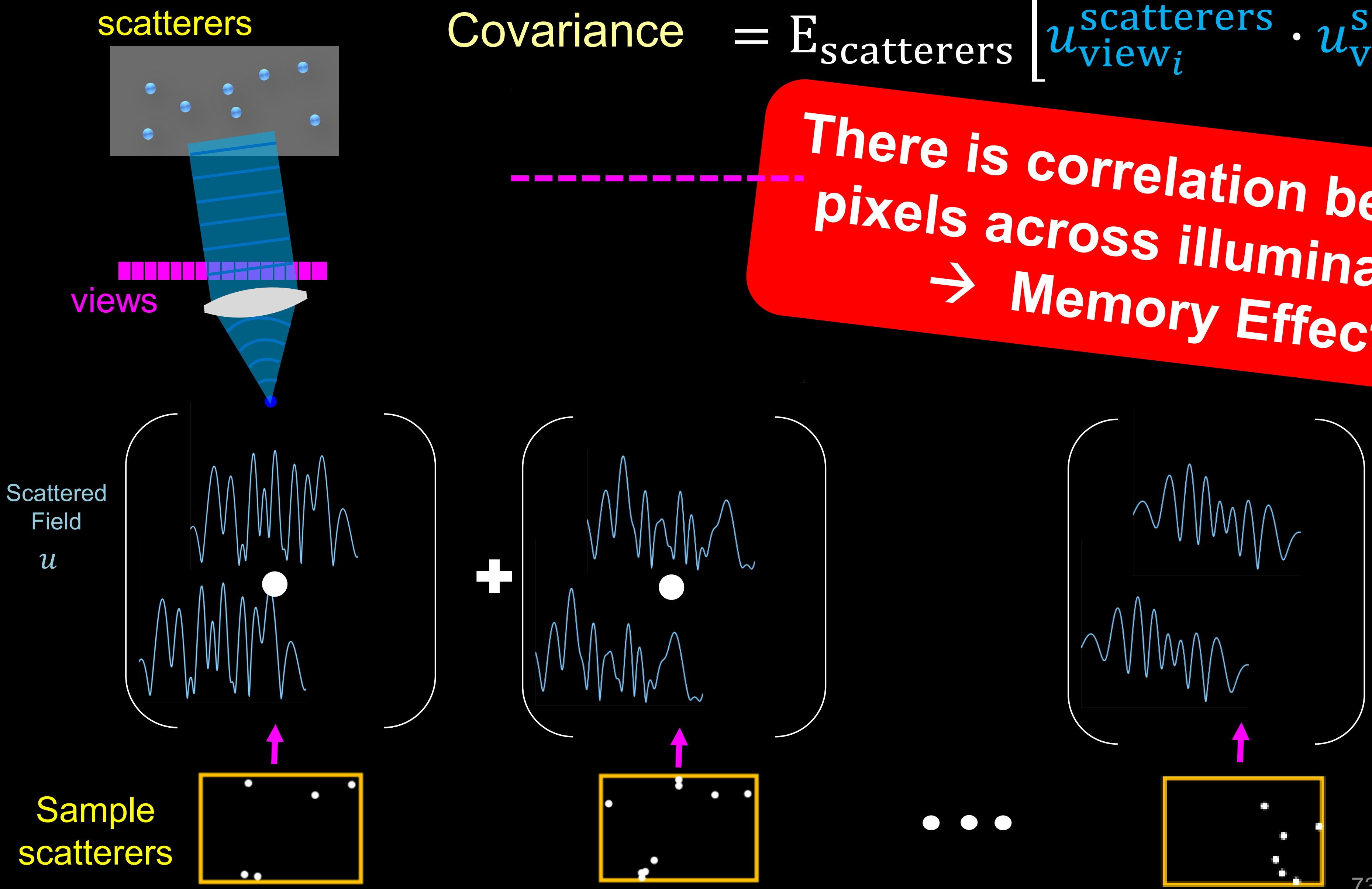
Field Covariance $= \mathrm{E}_{\text{scatterers}}\left[u_{\text{view}_i}^{\text{light}_1,\text{scatterers}} \cdot u_{\text{view}_j}^{\text{light}_2,\text{scatterers}^*}\right]$



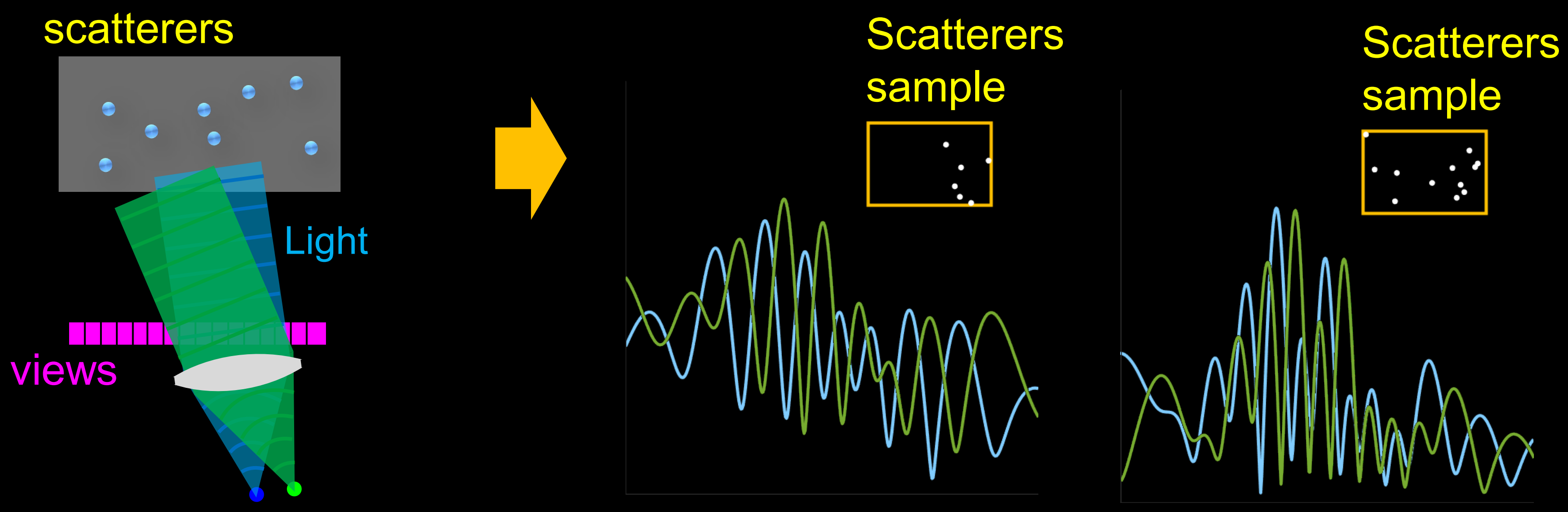
2^{nd} Moment - Covariance

Covariance $= \mathrm{E}_{\text{scatterers}}\left[u^{\text{scatterers}}_{\text{view}_i} \cdot u^{\text{scatterers}\,*}_{\text{view}_j}\right]$

There is correlation between pixels across illuminations → Memory Effect



Cross –illumination statistics

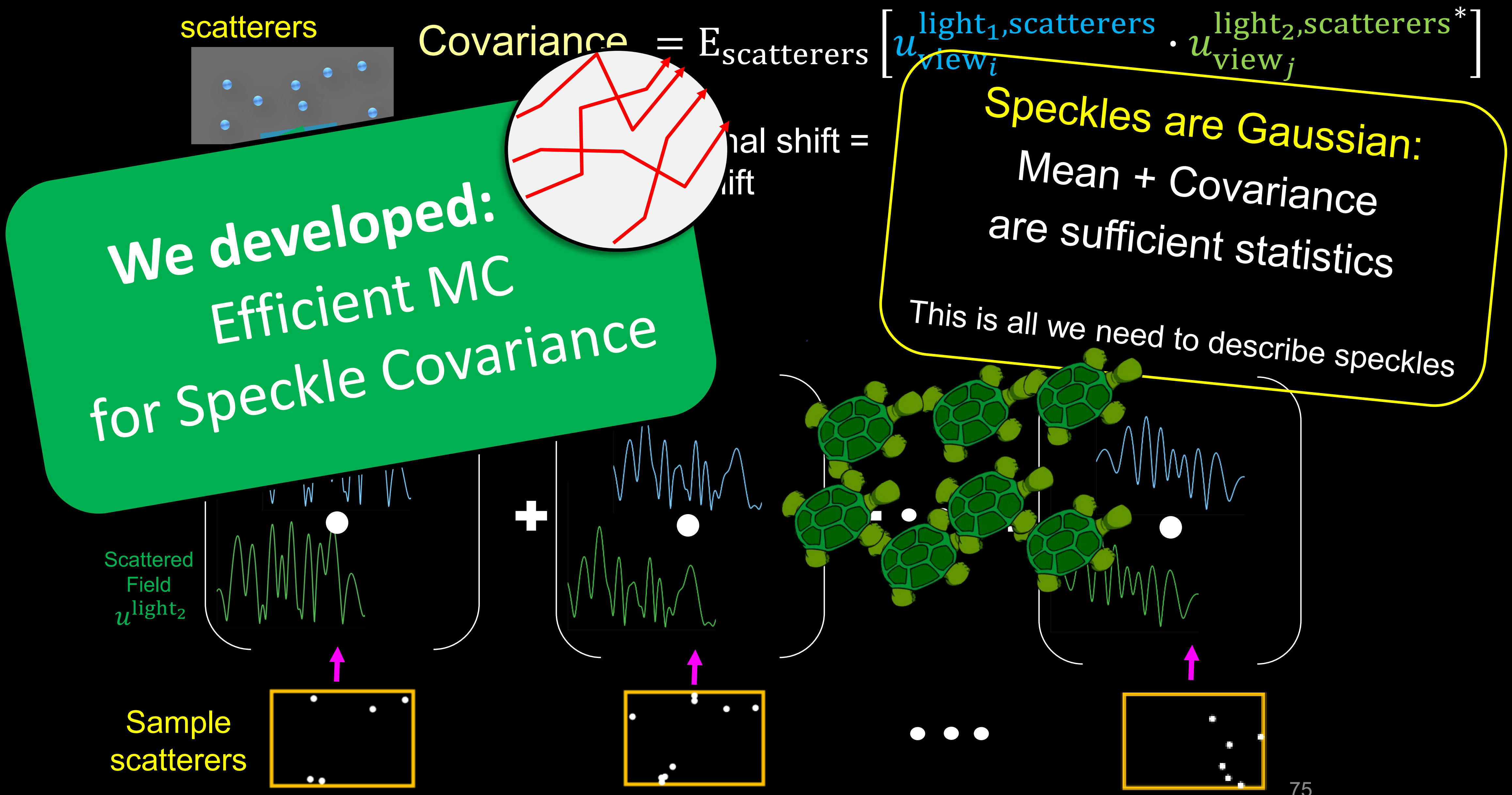


Memory Effect:

tilting illumination results in highly correlated shifted speckles

Next: Cross Illumination Covariance

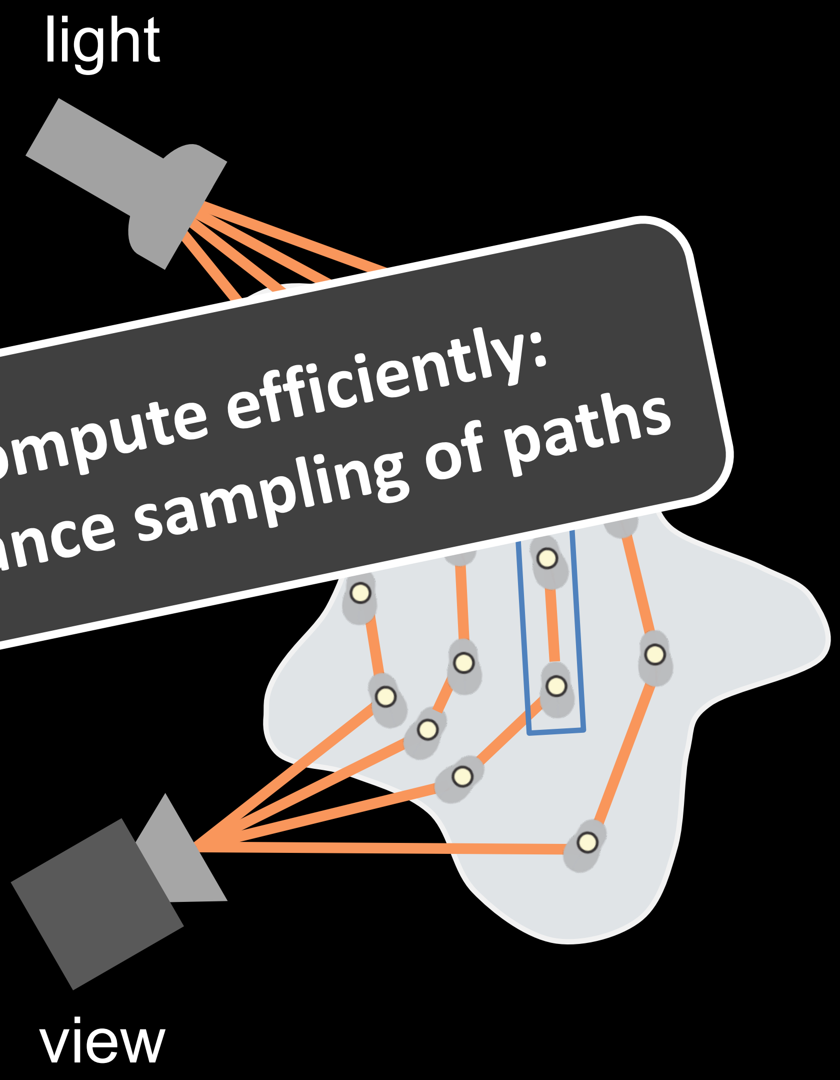
Cross –illumination statistics



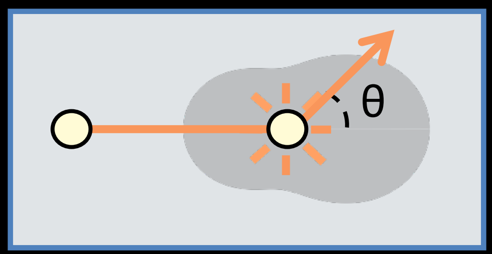
Monte Carlo Rendering 101

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

Throughput that acts on each path, depends on the scattering material



To compute efficiently: importance sampling of paths



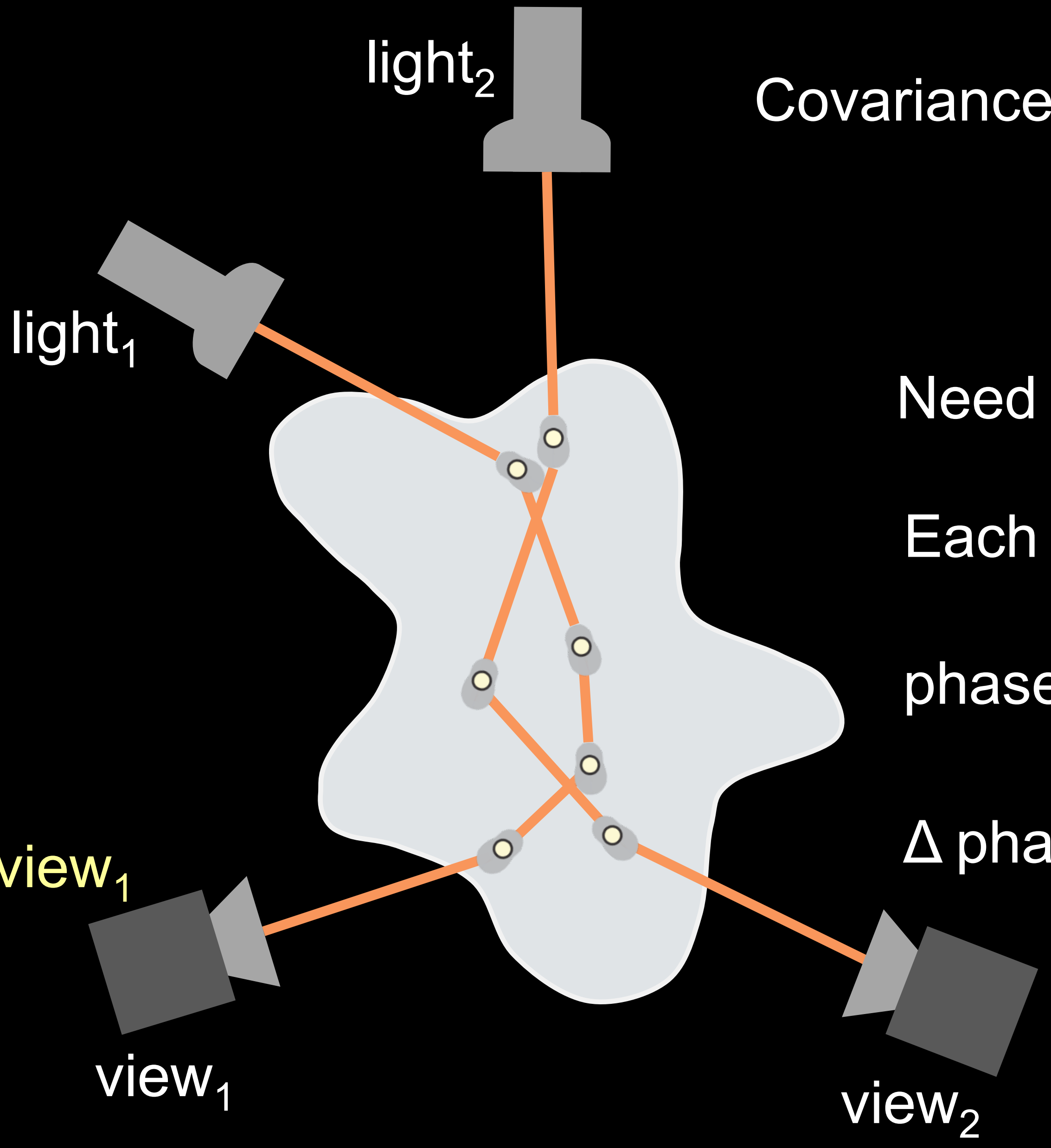
material =

volumetric density (extinction coefficient) σ

scattering albedo a

phase function p_θ

Covariance Rendering



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

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

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

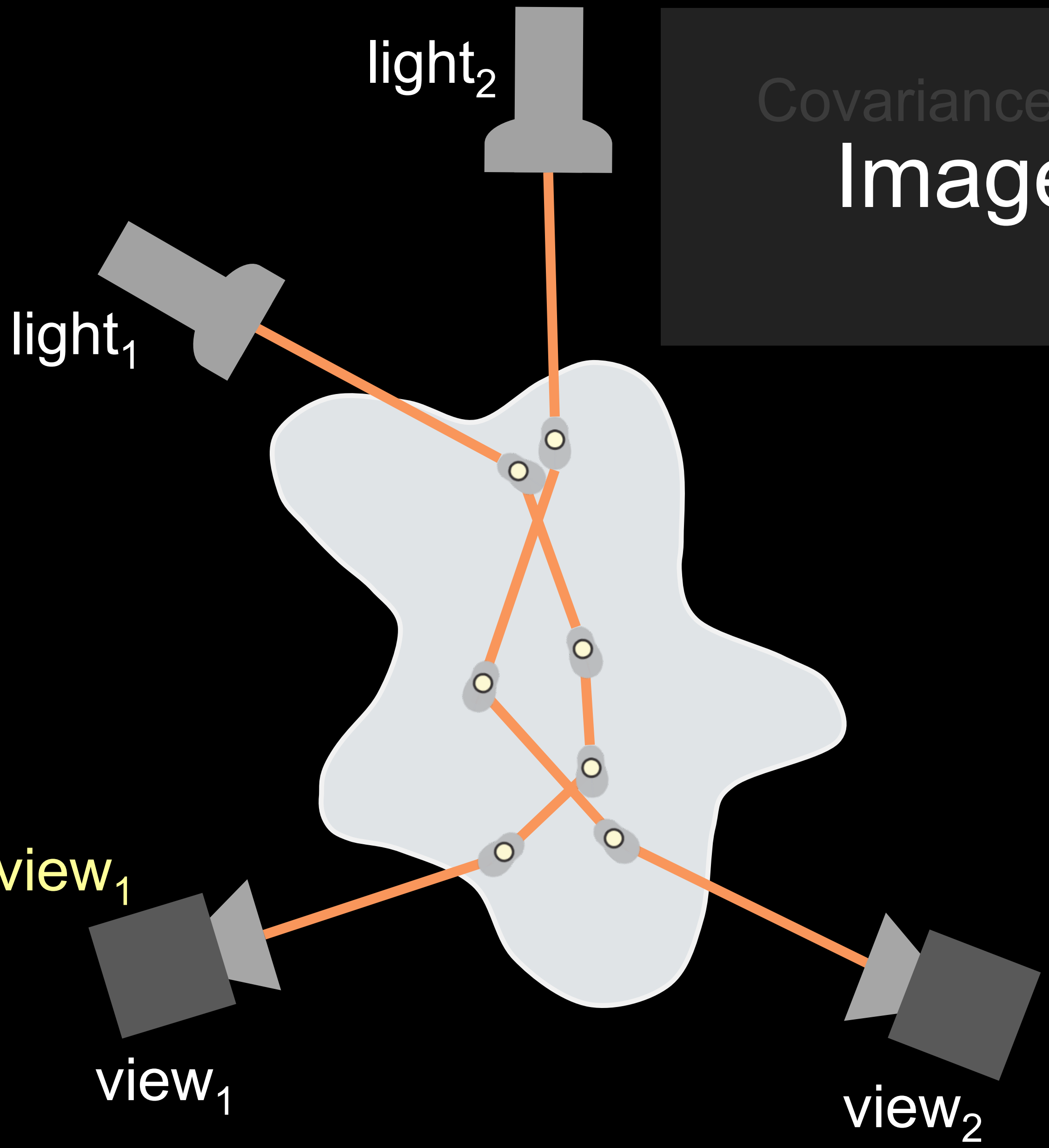
Each path contributes a complex number u

phase ∝ Length (path)

Δ phase ∝ Length (path$_1$) - Length (path$_2$)

Covariance Rendering

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



$\text{path}_1 = \text{path}_2$

⇩

Same complex contribution

$u(\text{path}_1) = u(\text{path}_2)$

⇩

$\Delta\,\text{phase} = 0$

⇩

$u(\text{path}) \cdot u^{*}(\text{path}) = f\,(\text{path})$

Real

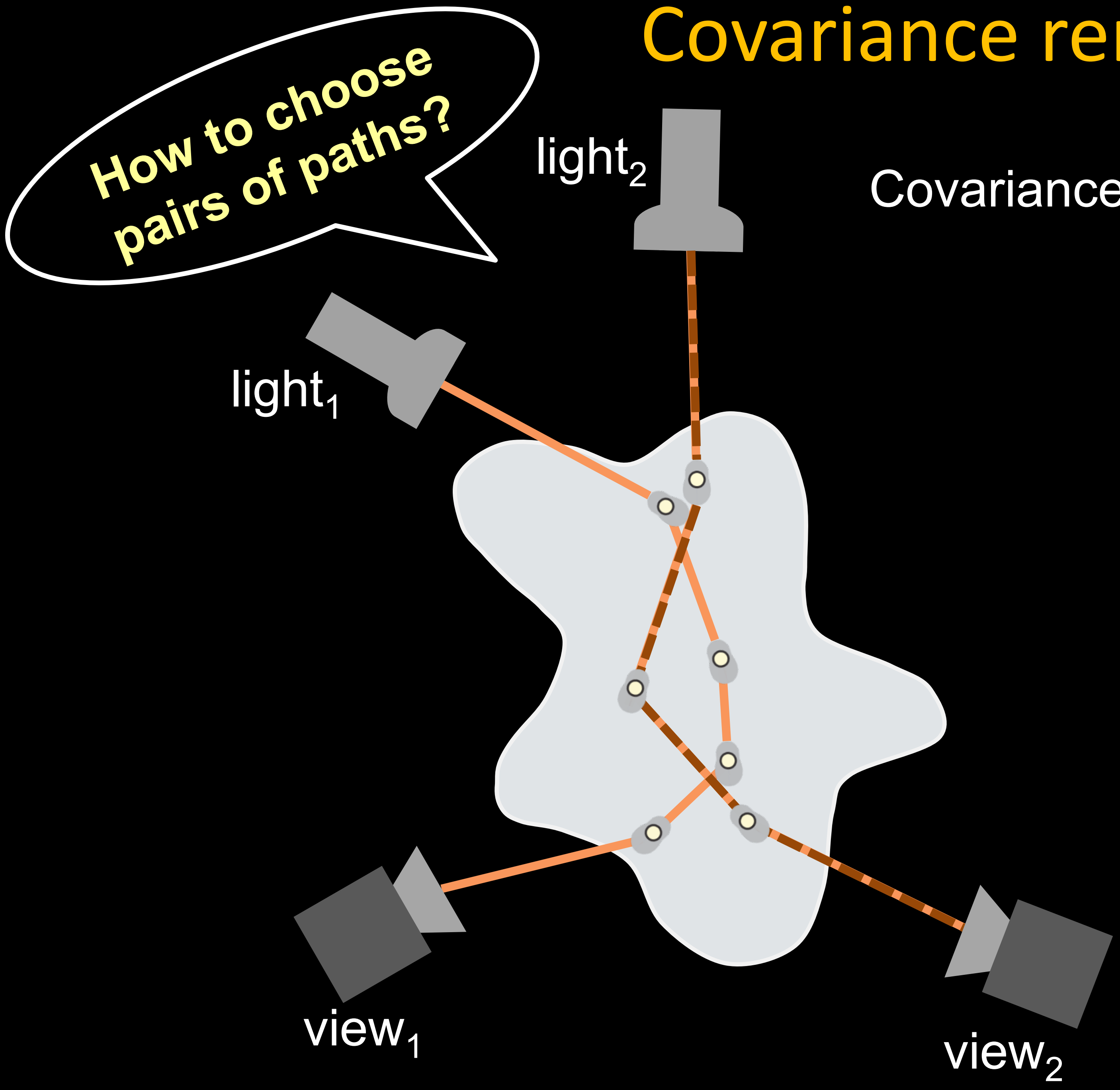
path$_1$:
light$_1$ → view$_1$

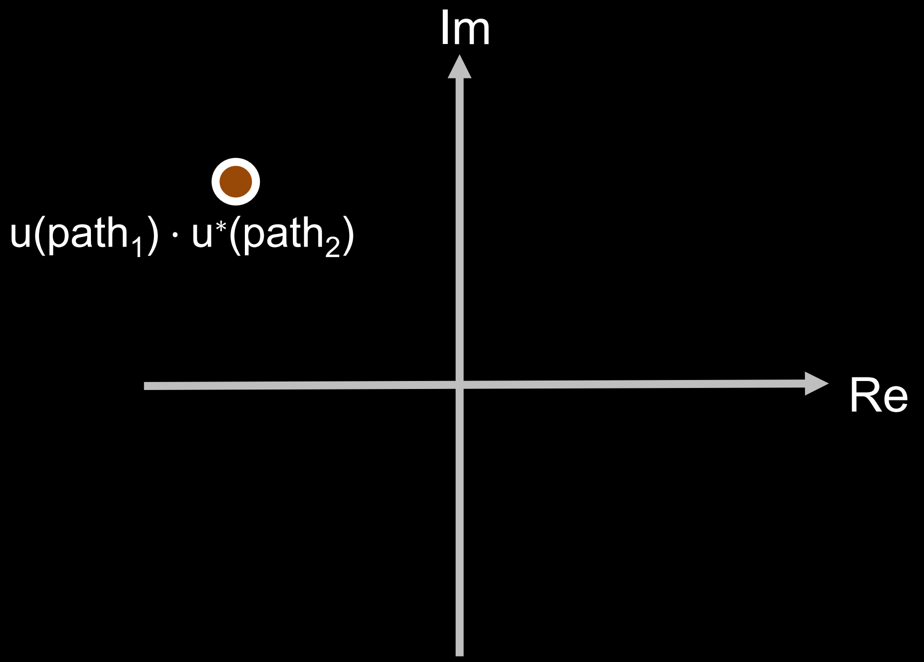
path$_2$:
light$_2$ → view$_2$

Covariance rendering

$$\text{Covariance} = \int_{path_1, path_2} u(path_1) \cdot u^*(path_2)$$

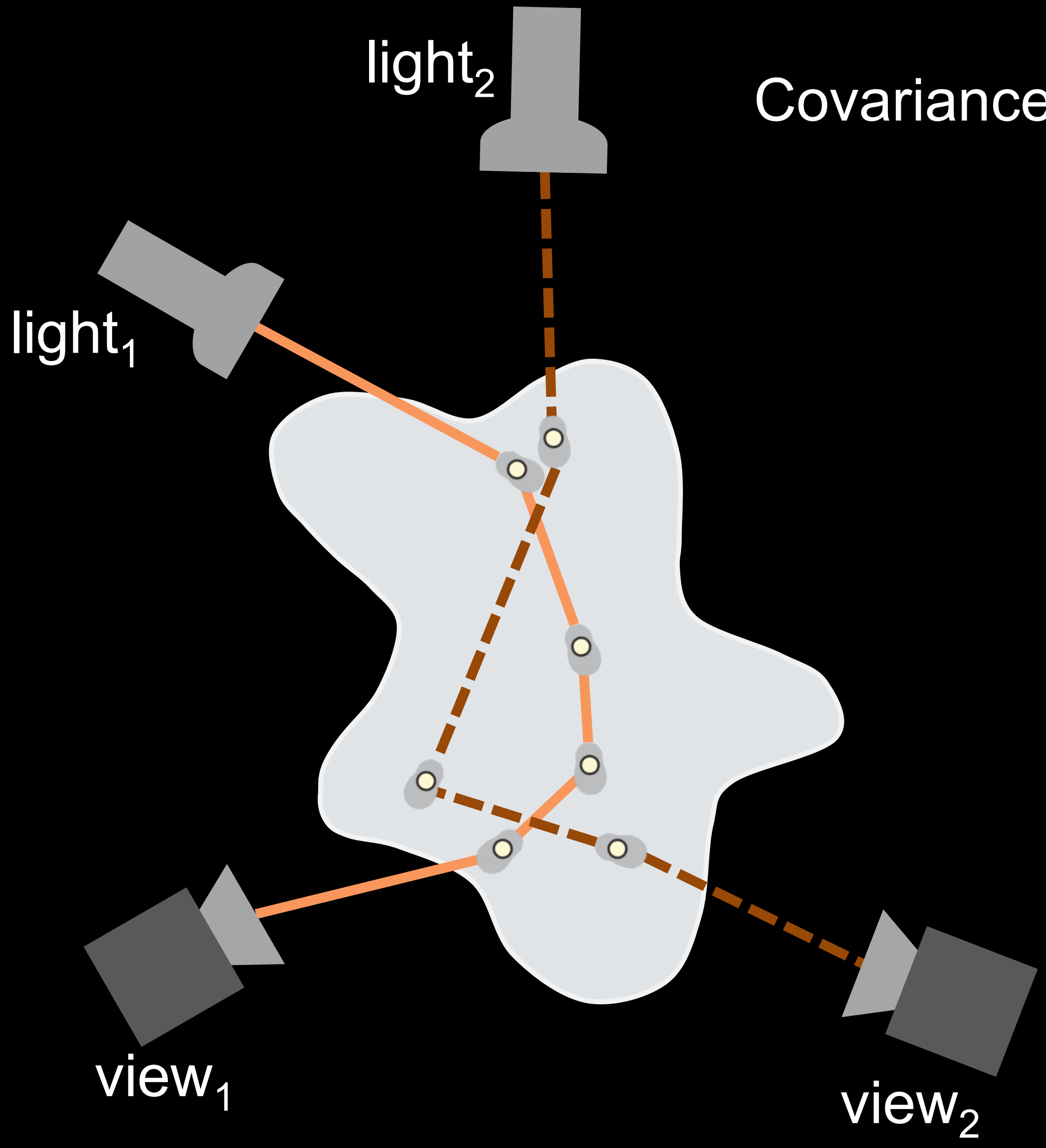
$u = |u|\, e^{i \cdot phase}$

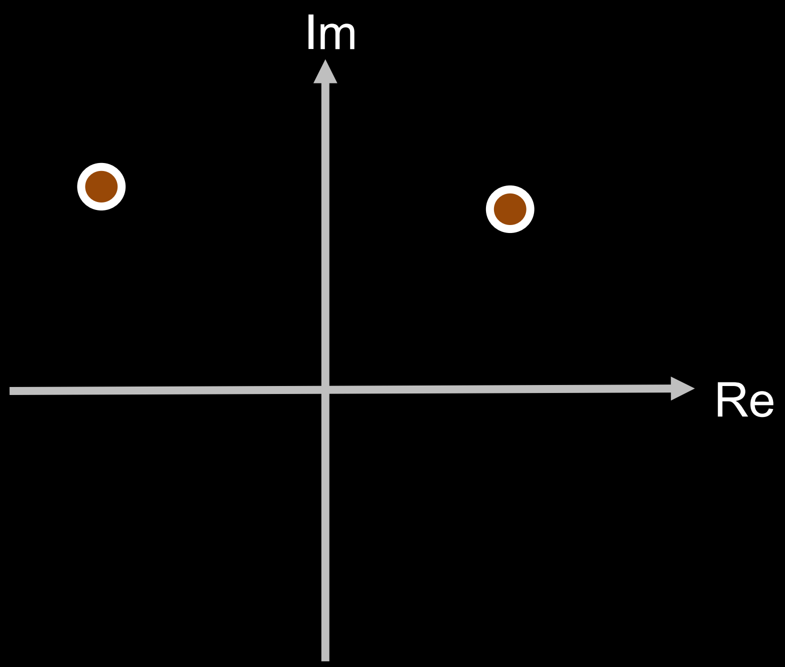




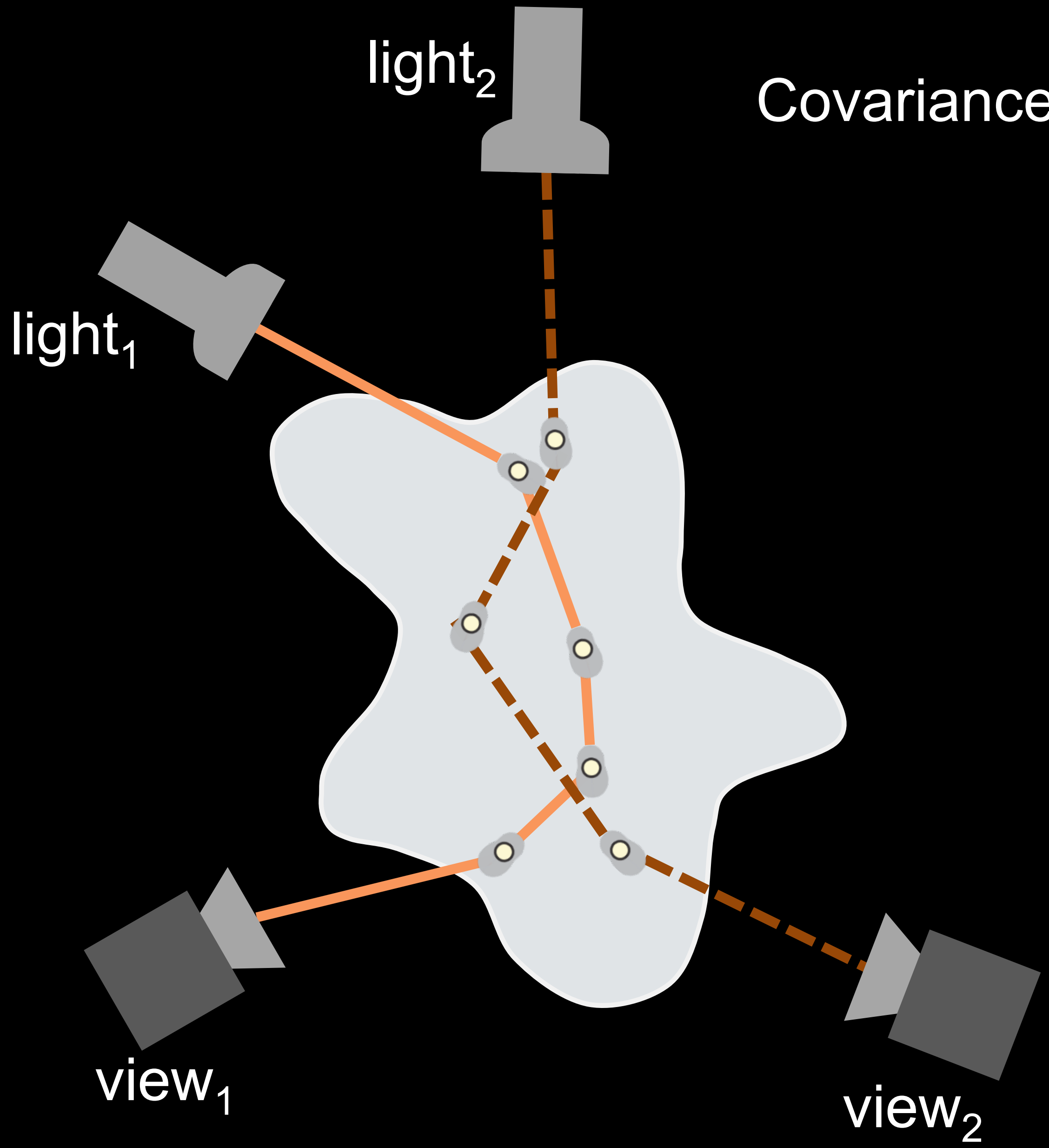
Covariance rendering

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

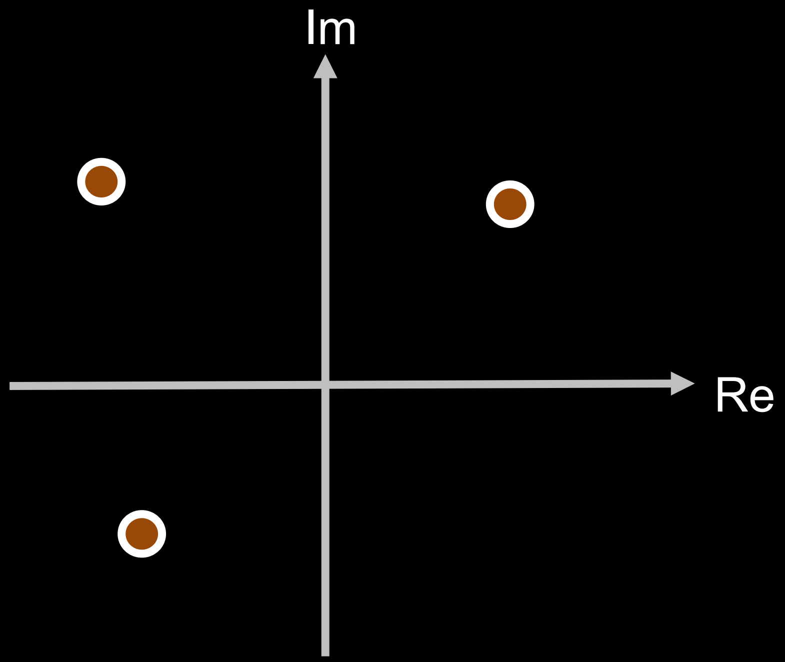




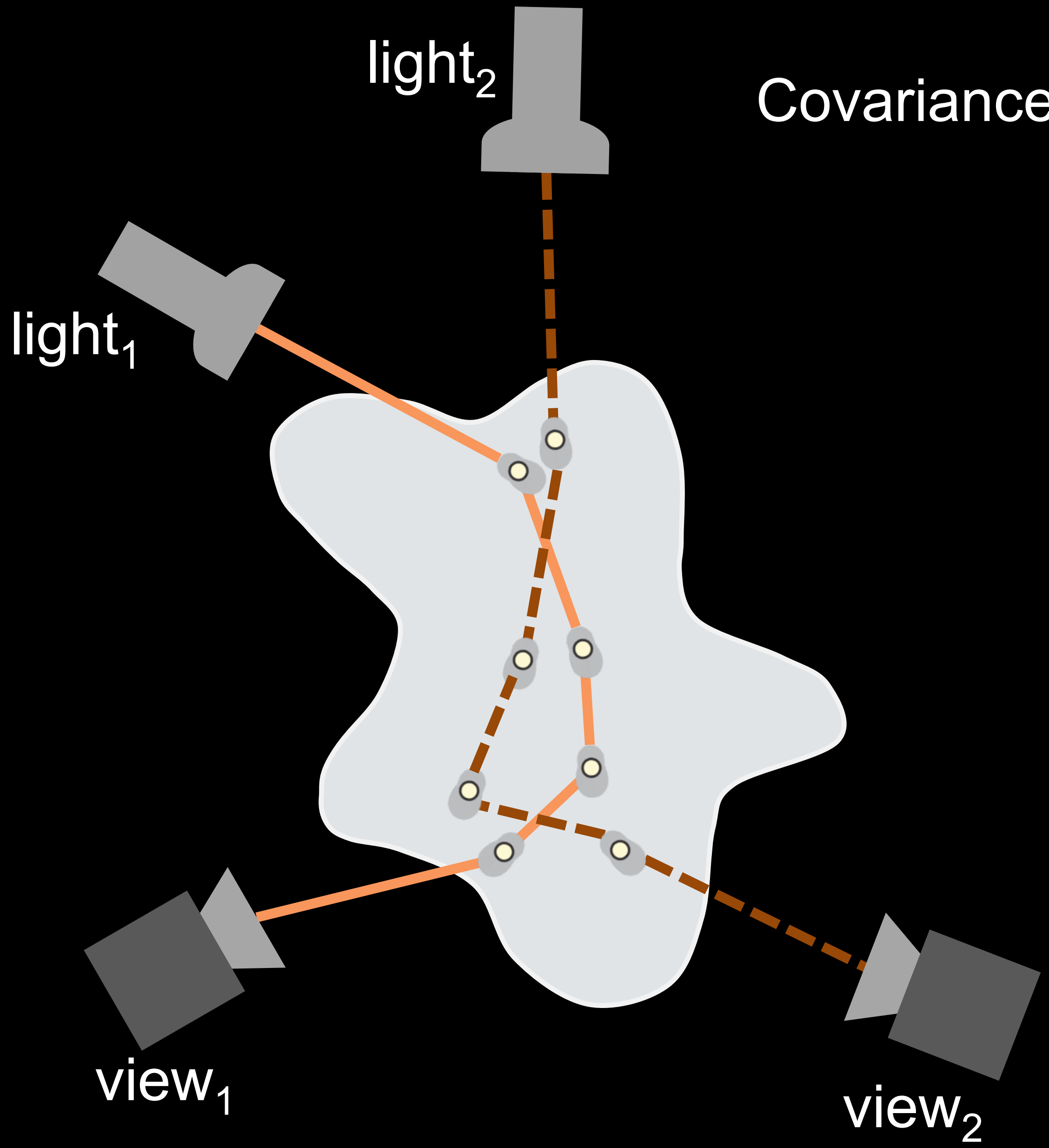
Covariance rendering



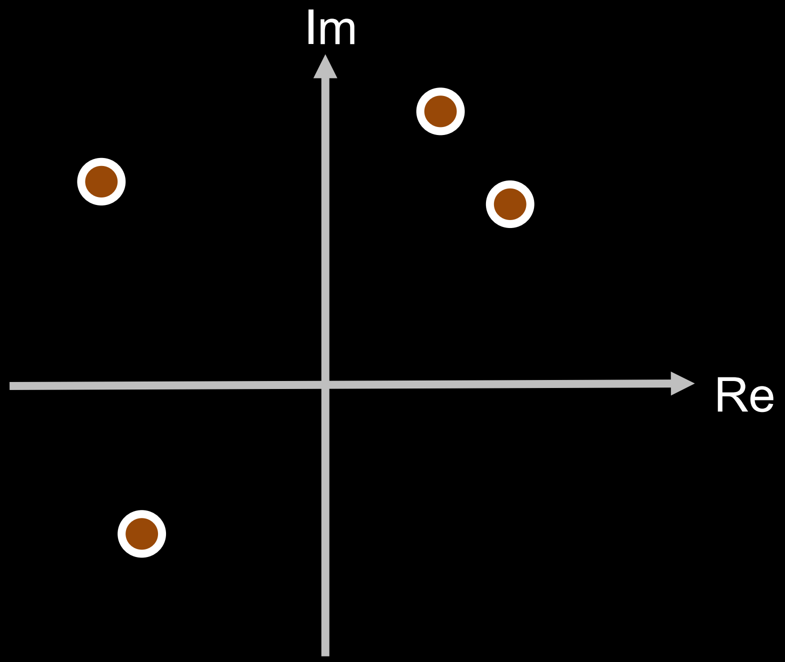
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



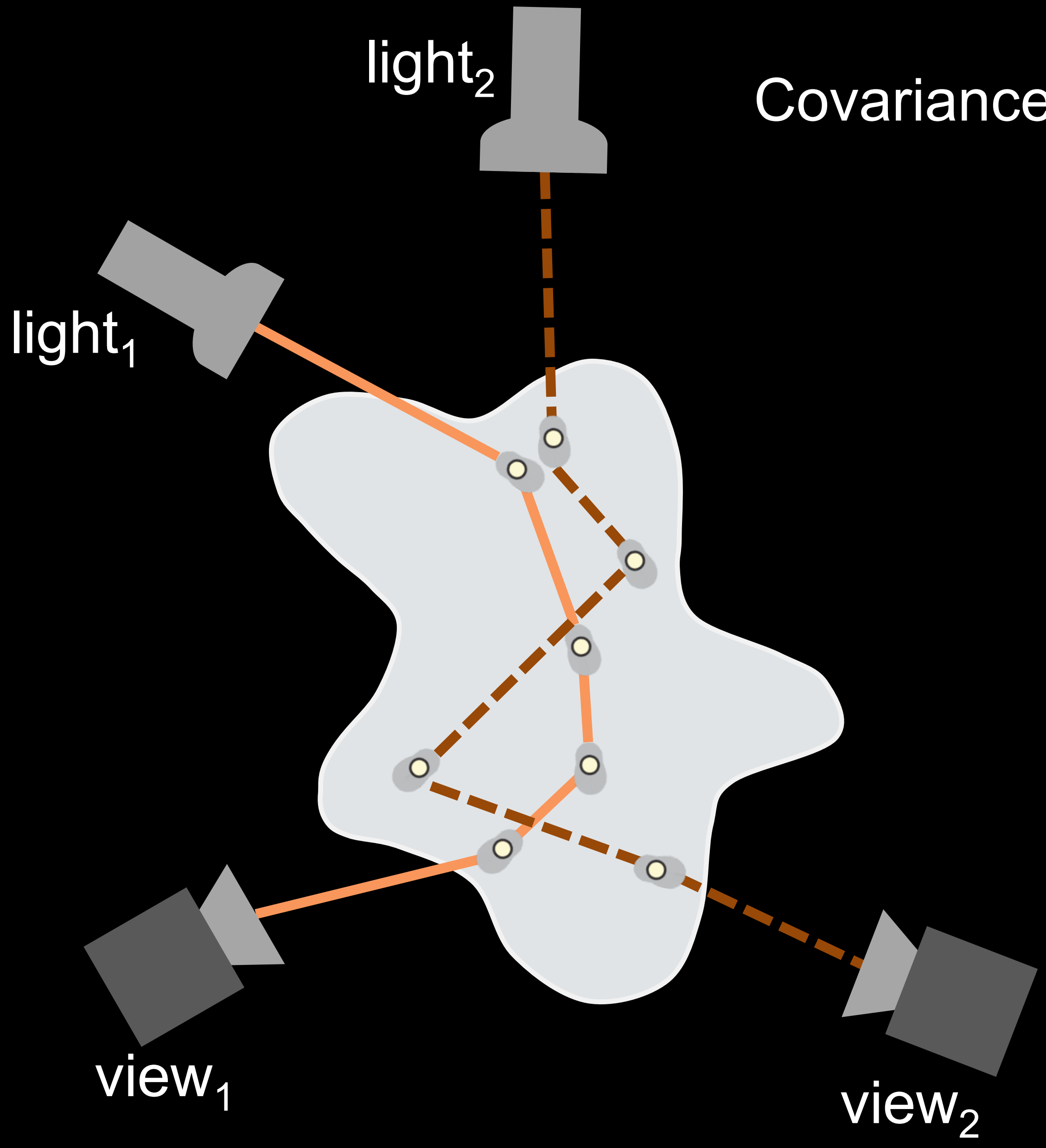
Covariance rendering



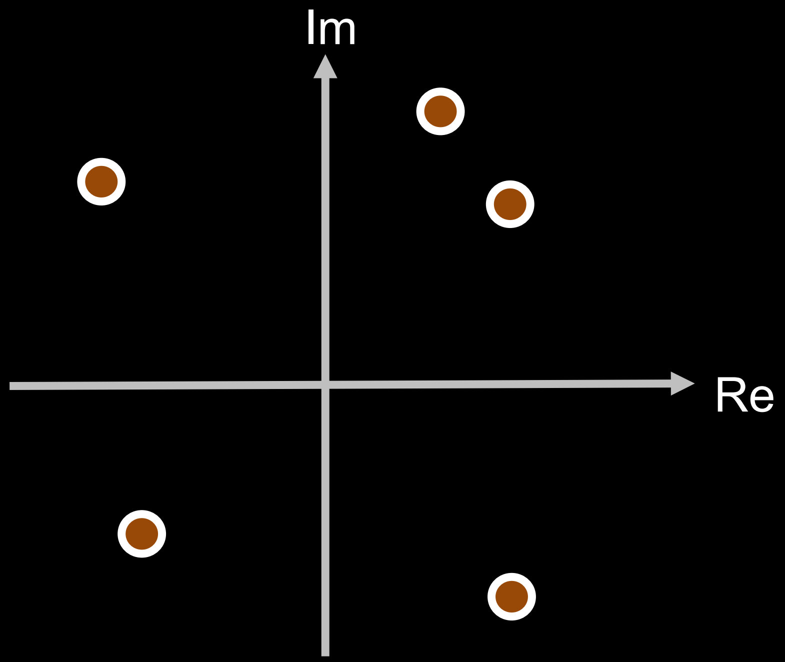
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



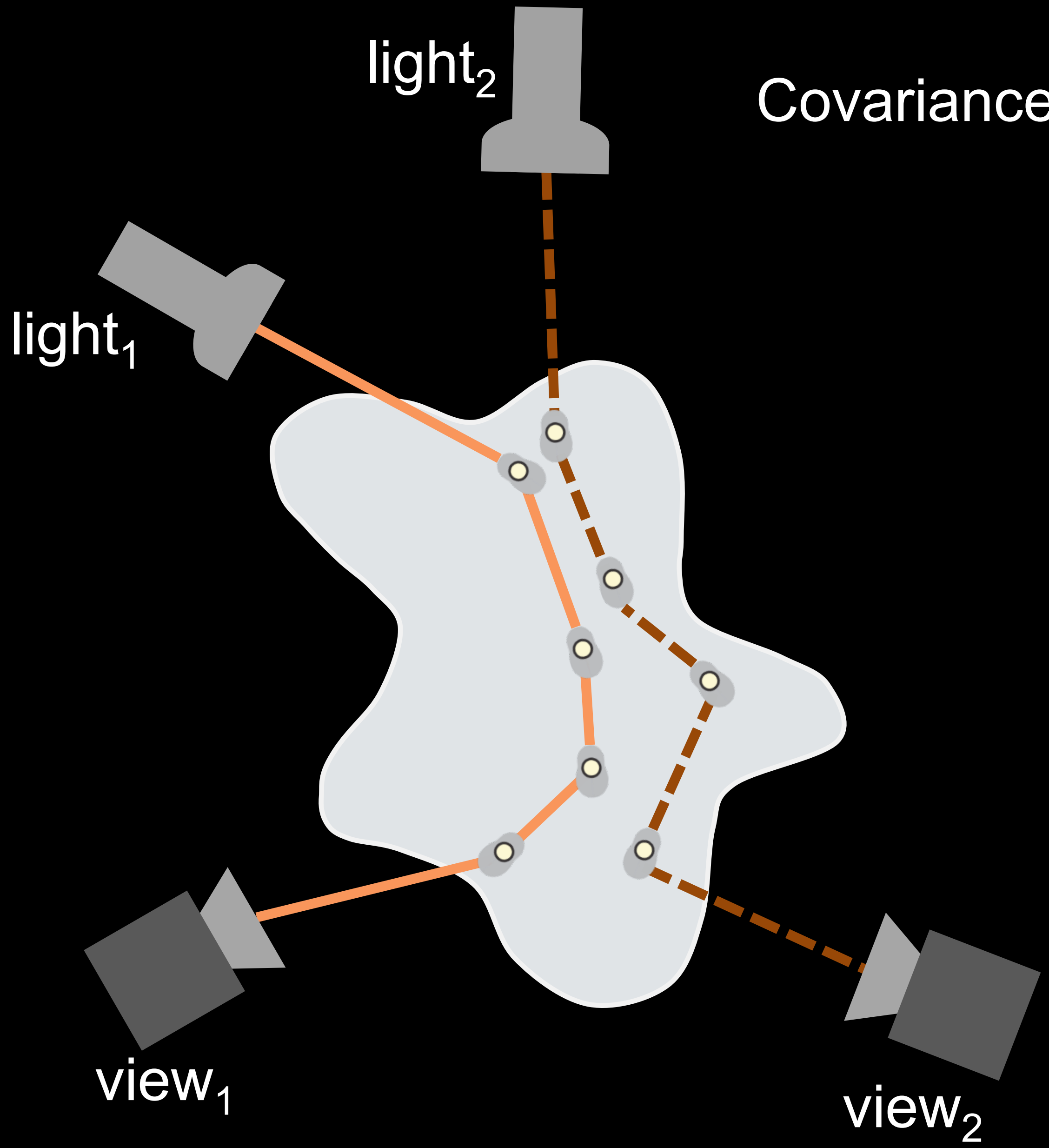
Covariance rendering



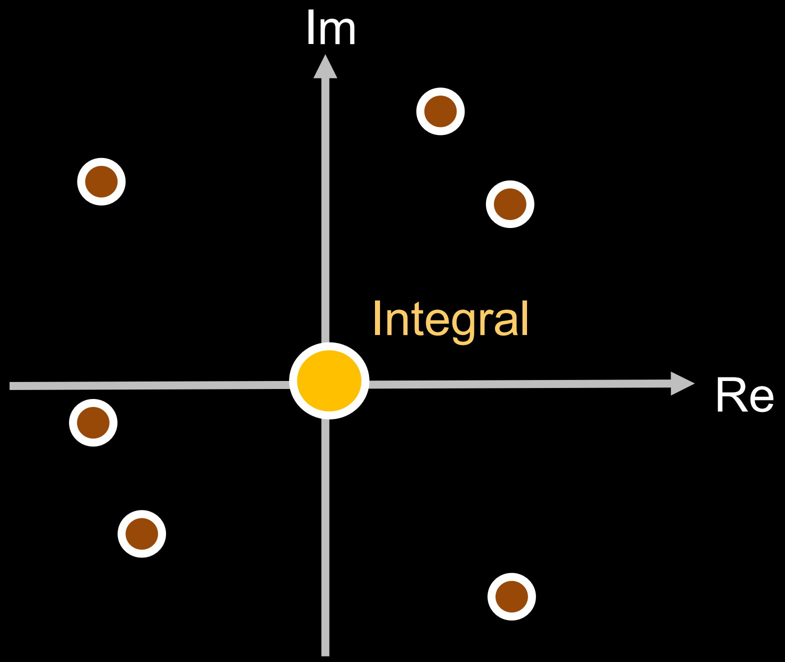
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



Covariance rendering

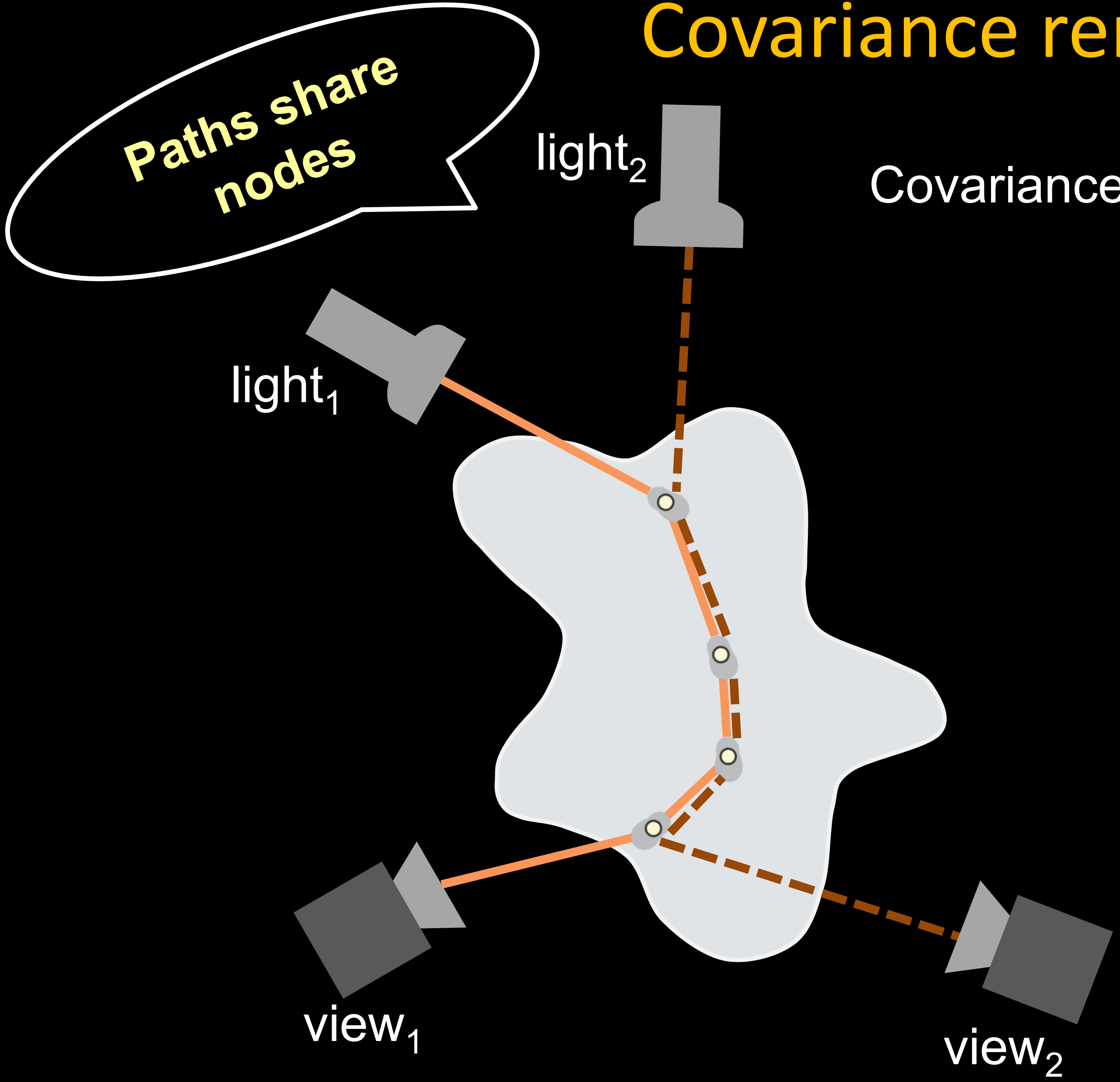


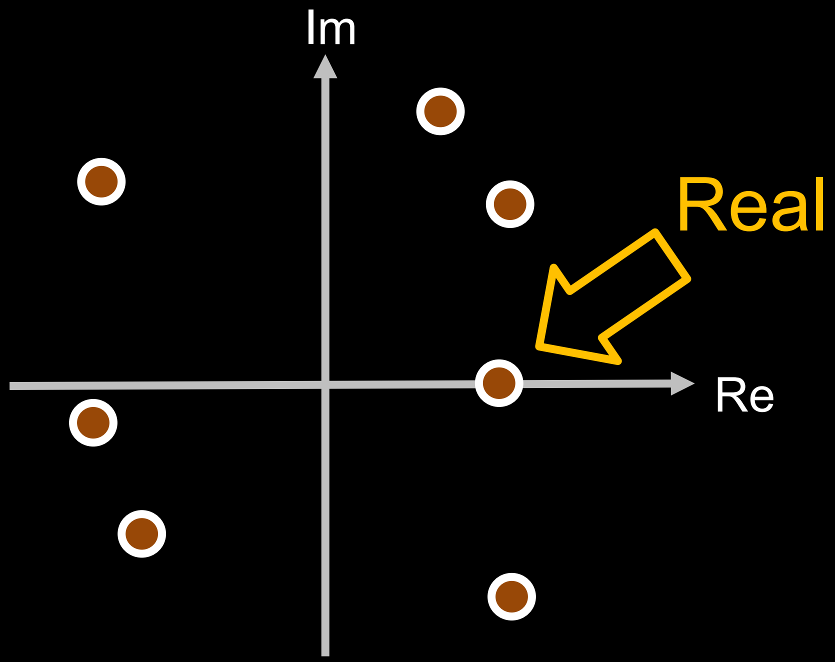
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



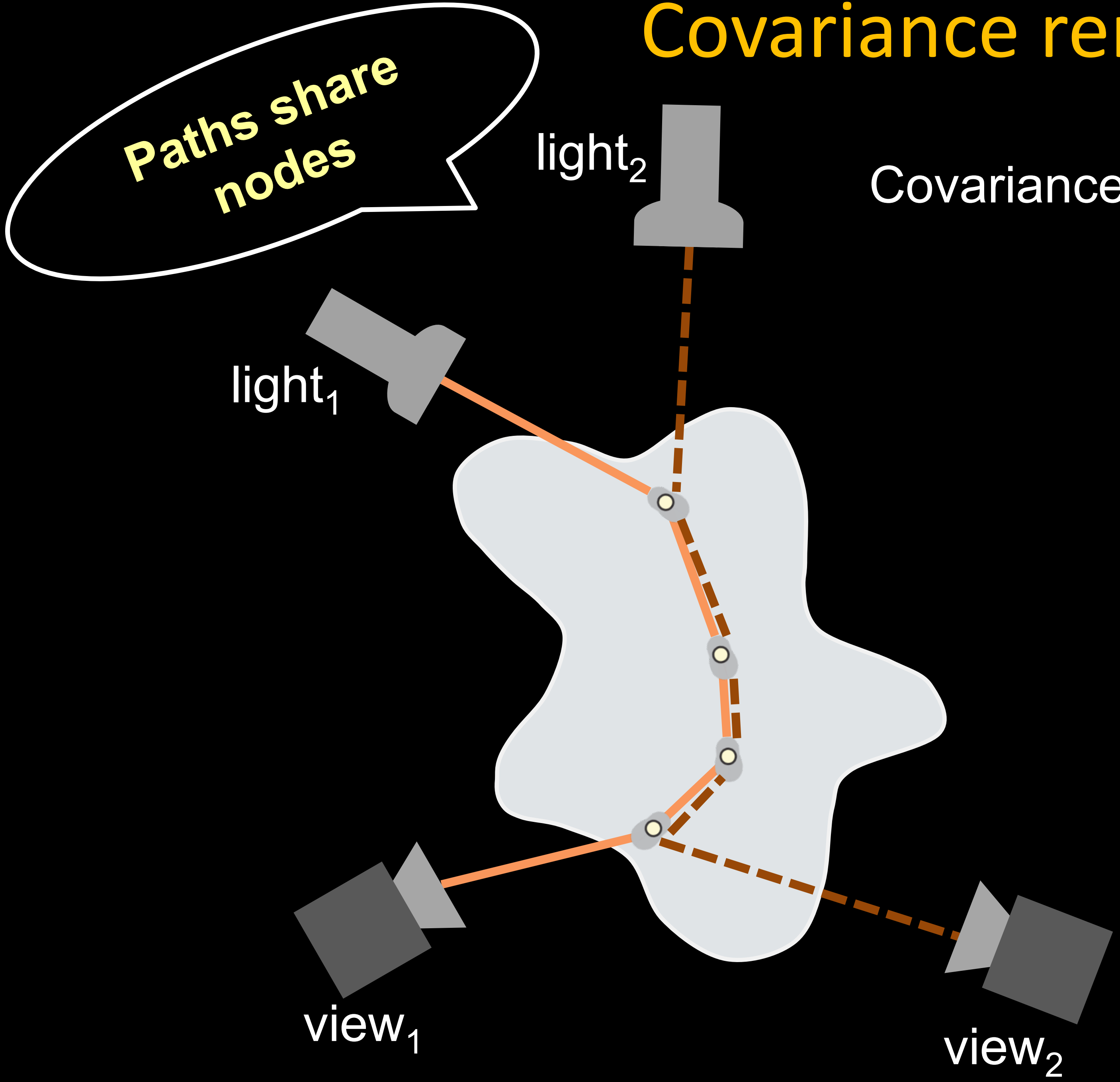
Covariance rendering

$$\text{Covariance} = \int_{path_1,path_2} u(path_1) \cdot u^*(path_2)$$

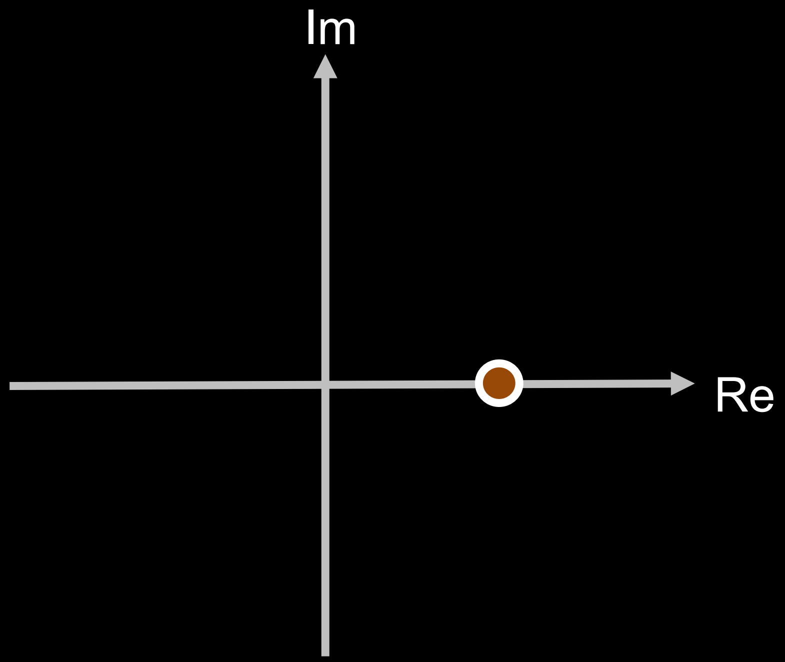




Covariance rendering

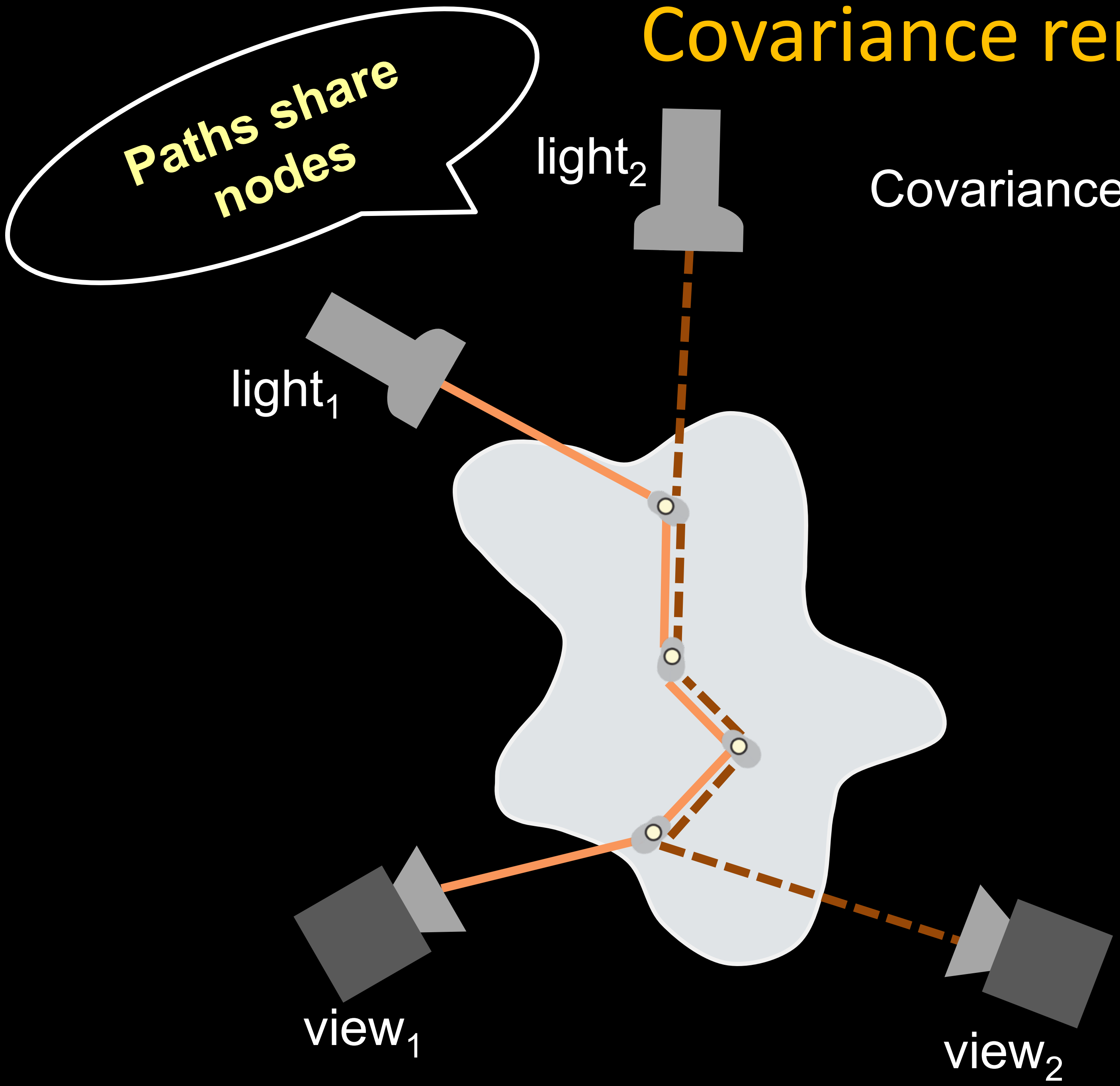


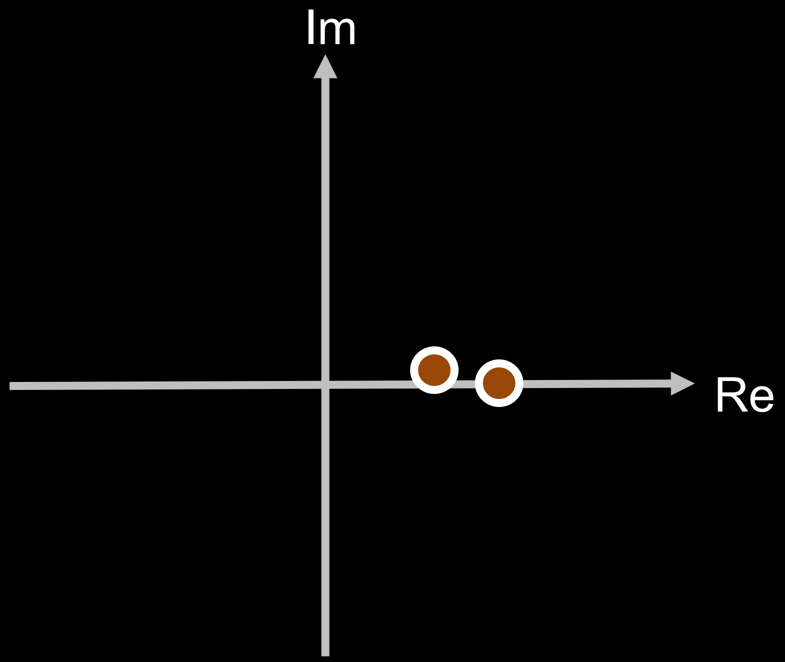
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



Covariance rendering

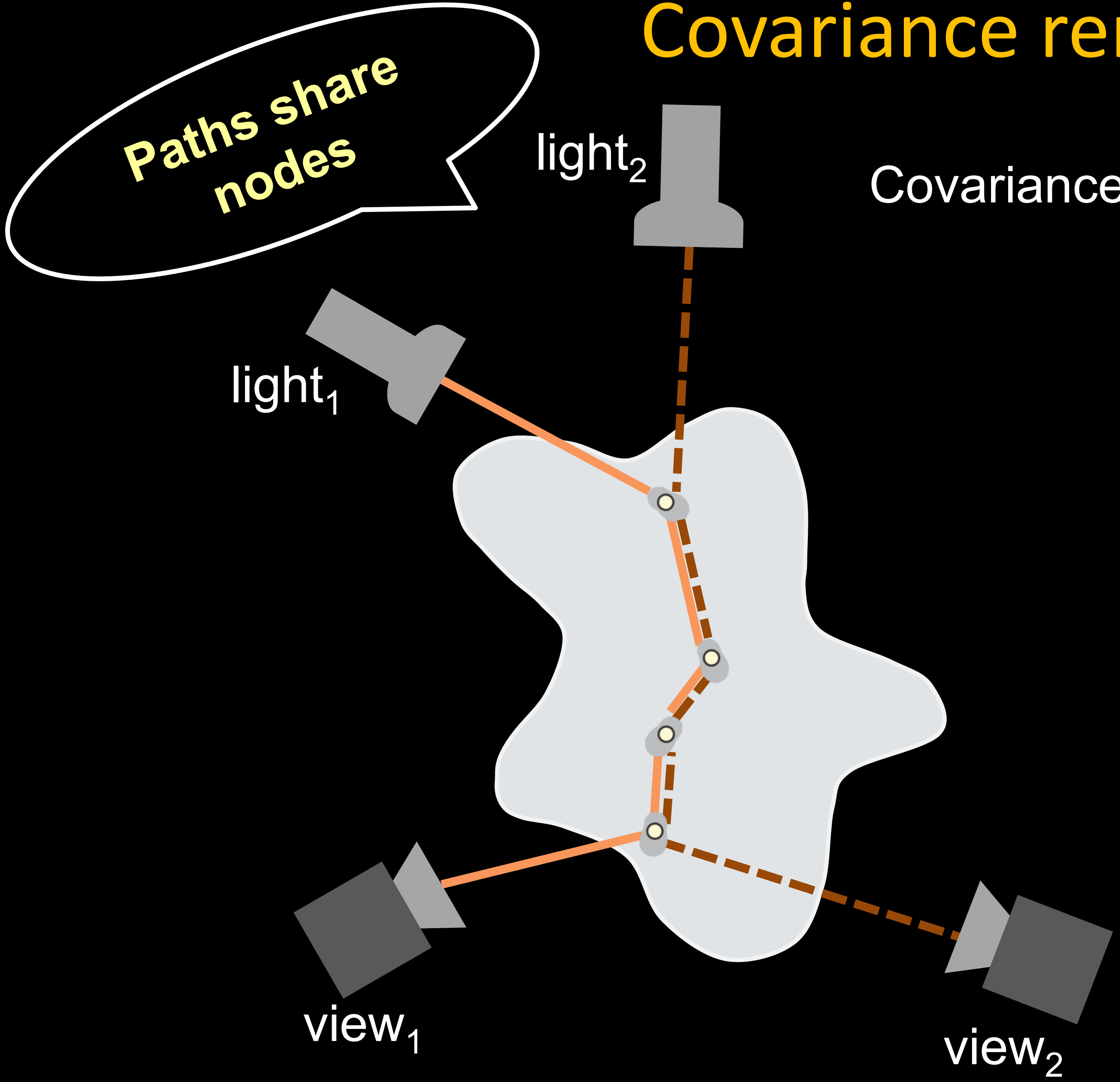
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

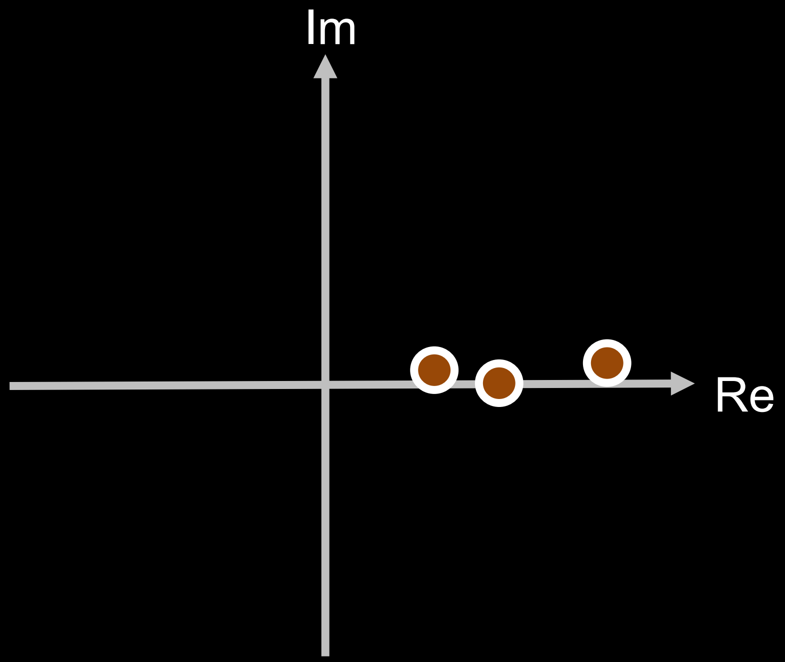




Covariance rendering

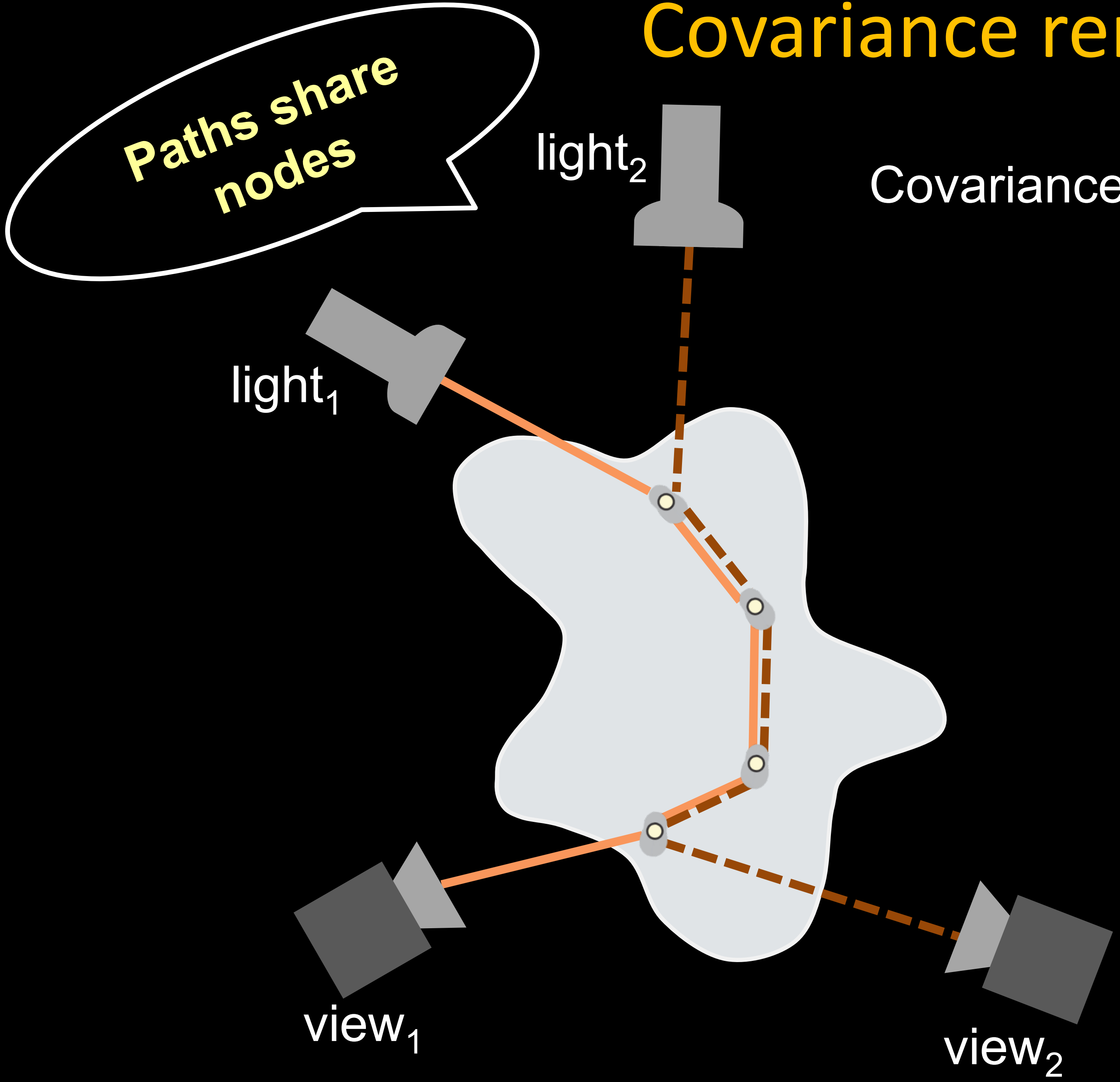
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$

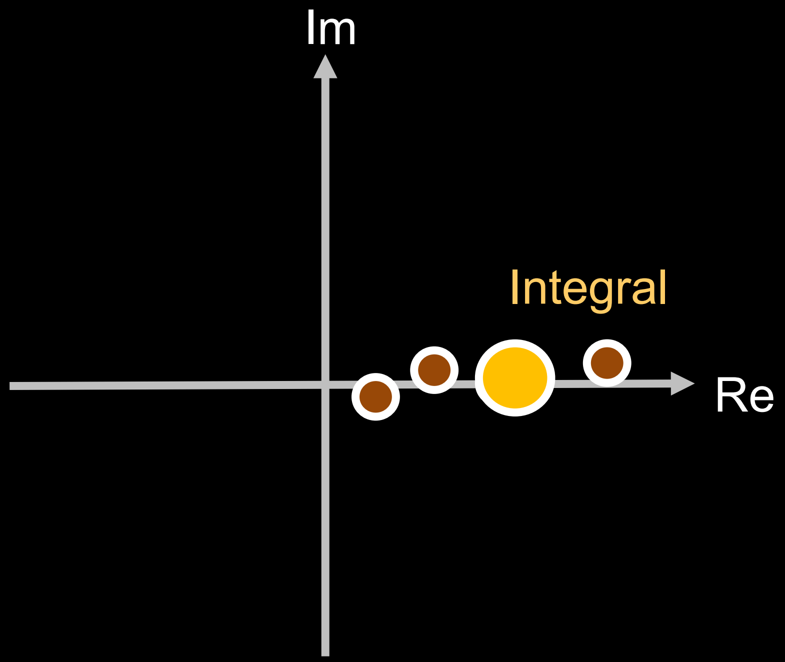




Covariance rendering

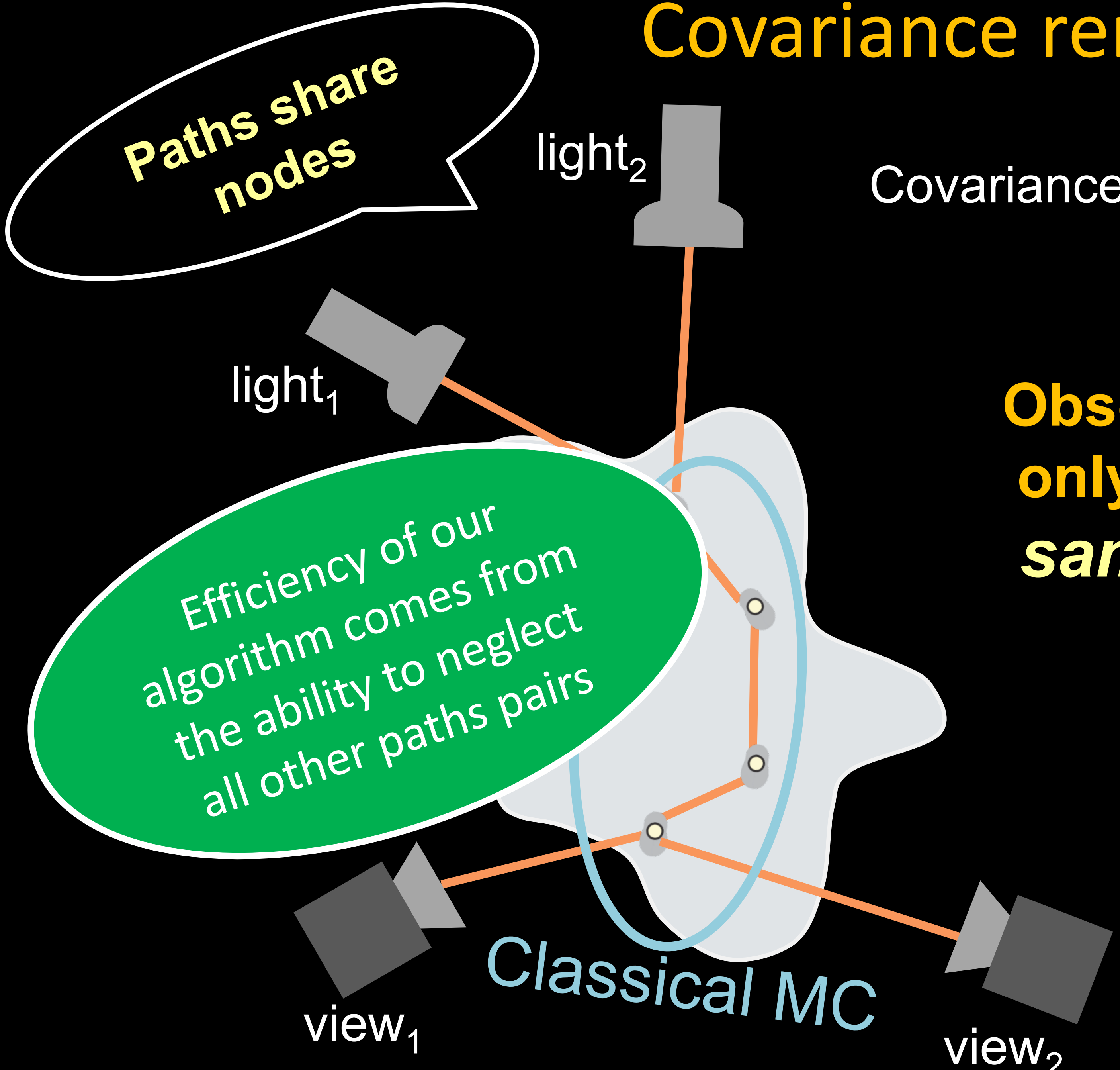
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



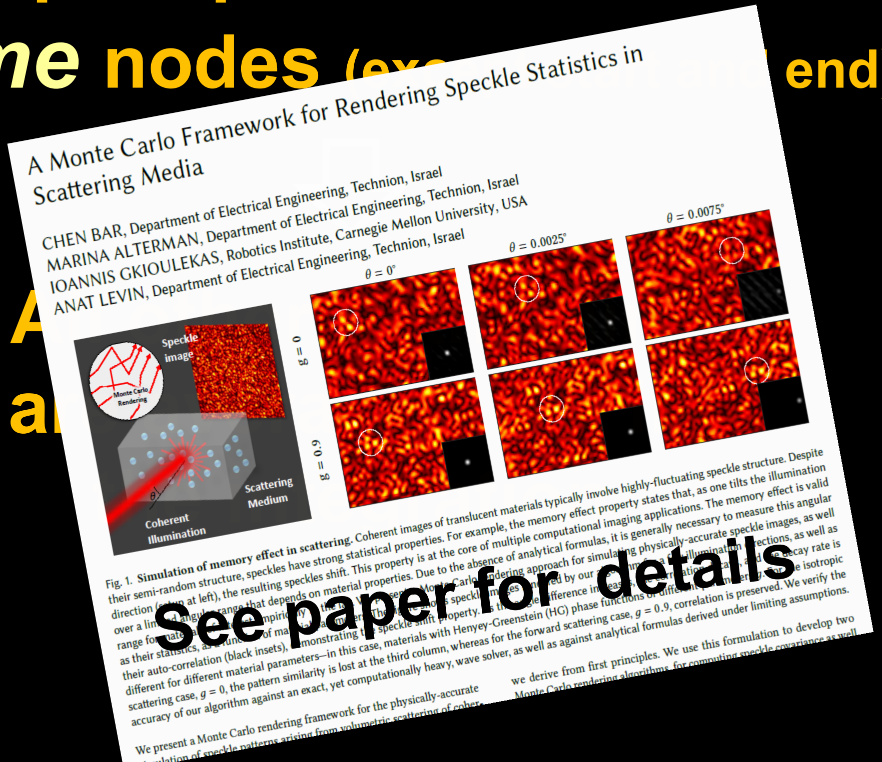


Covariance rendering

$$\text{Covariance} = \int_{path_1, path_2} u(path_1) \cdot u^*(path_2)$$

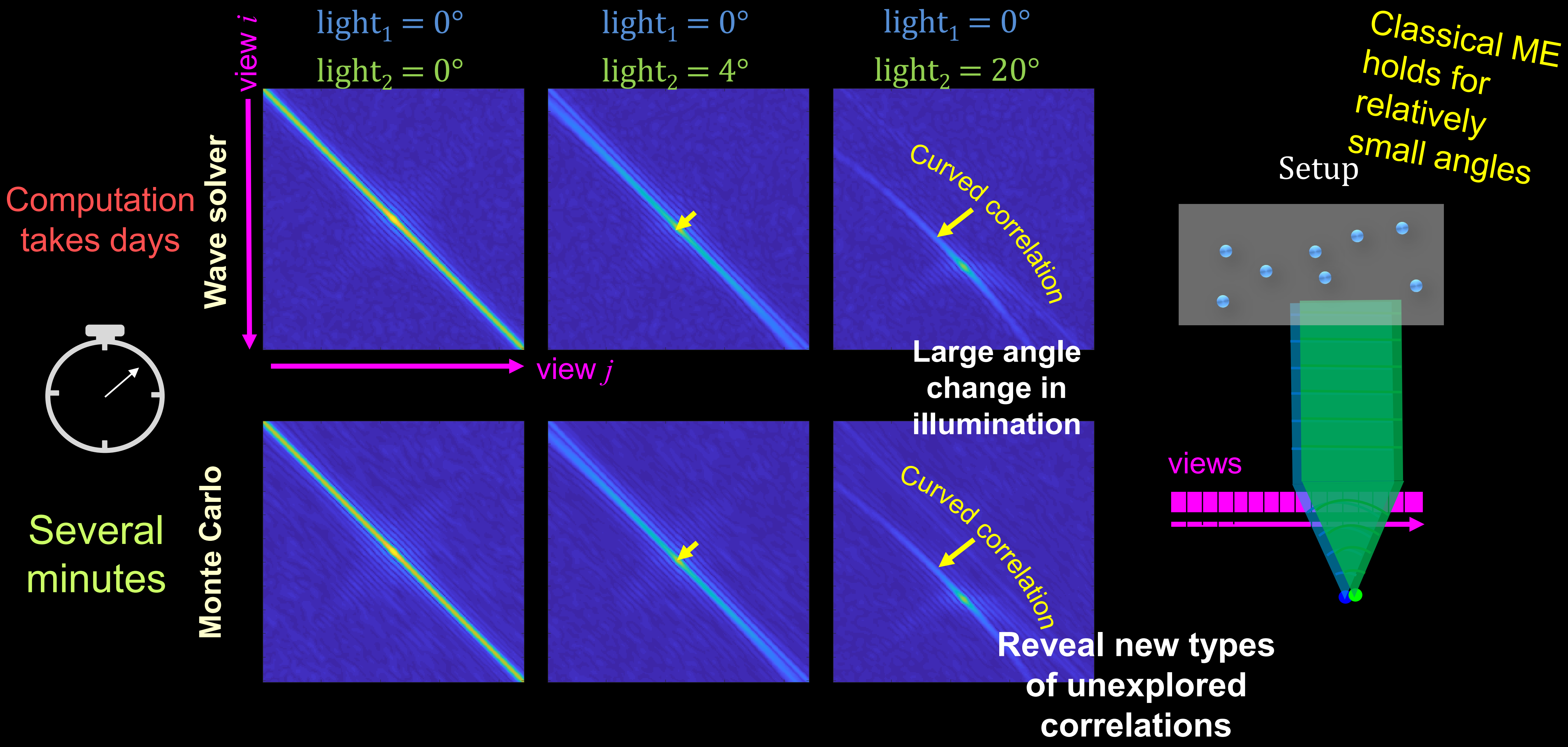


Observation: need to consider only path pairs that share the *same* nodes (ex... end)

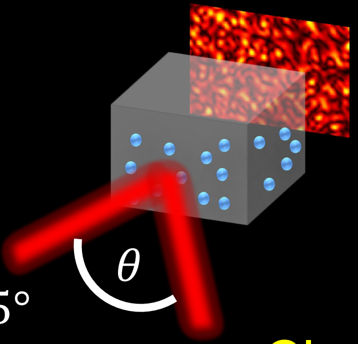

A Monte Carlo Framework for Rendering Speckle Statistics in Scattering Media

CHEN BAR, Department of Electrical Engineering, Technion, Israel
MARINA ALTERMAN, Department of Electrical Engineering, Technion, Israel
IOANNIS GKIOULEKAS, Robotics Institute, Carnegie Mellon University, USA
ANAT LEVIN, Department of Electrical Engineering, Technion, Israel

Validation: Wave Equation Covariances v.s. MC

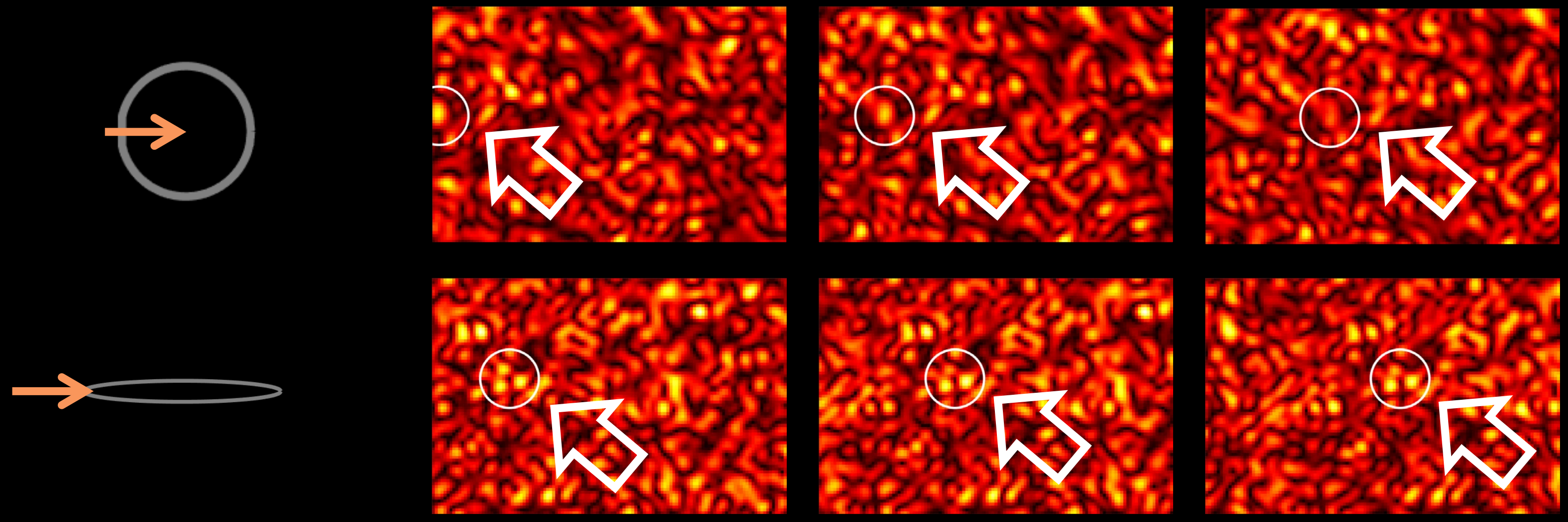


Rendering Speckles



Phase Function

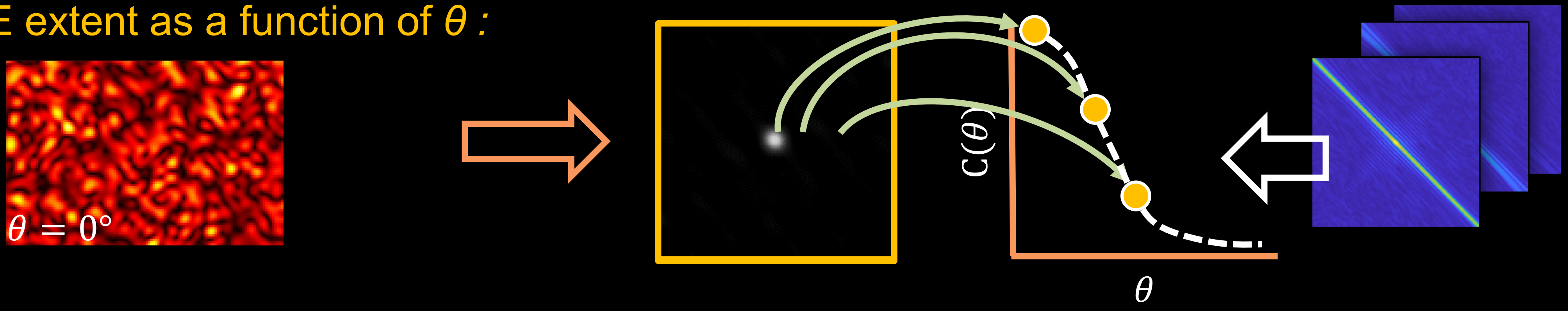
$\theta = 0°$ | $\theta = 0.0025°$ | $\theta = 0.005°$



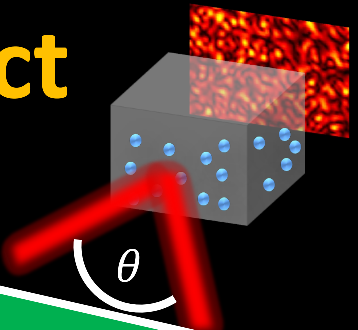
Classical ME holds for relatively small angles

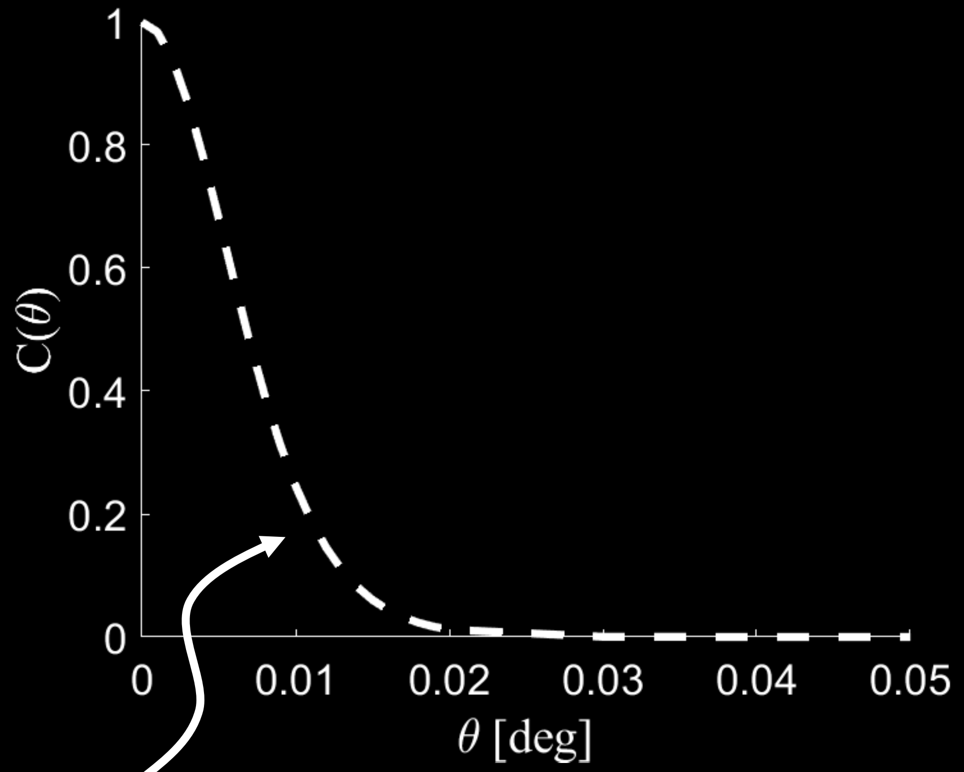
Exact ME extent is different for different materials.

Computing ME extent as a function of θ :



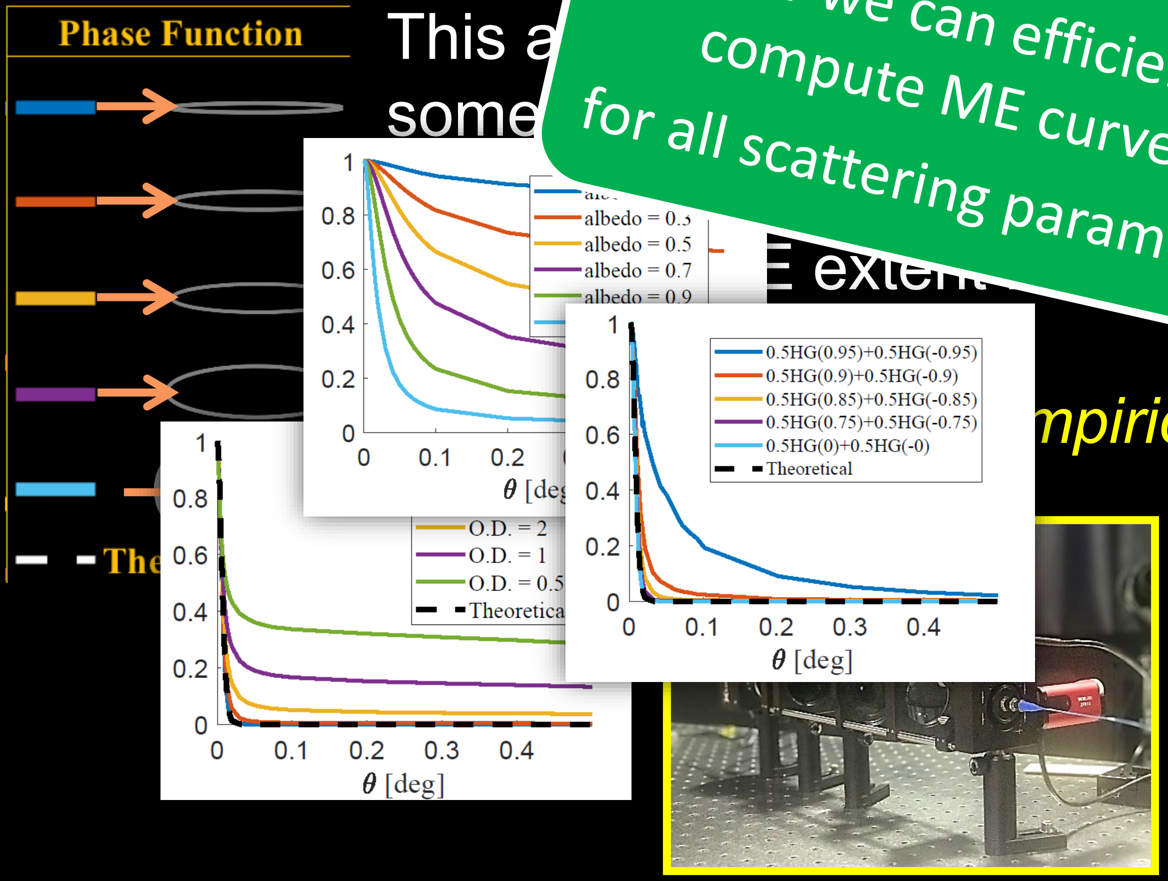
Evaluating the Memory Effect





Analytical solution based on diffusion approximation

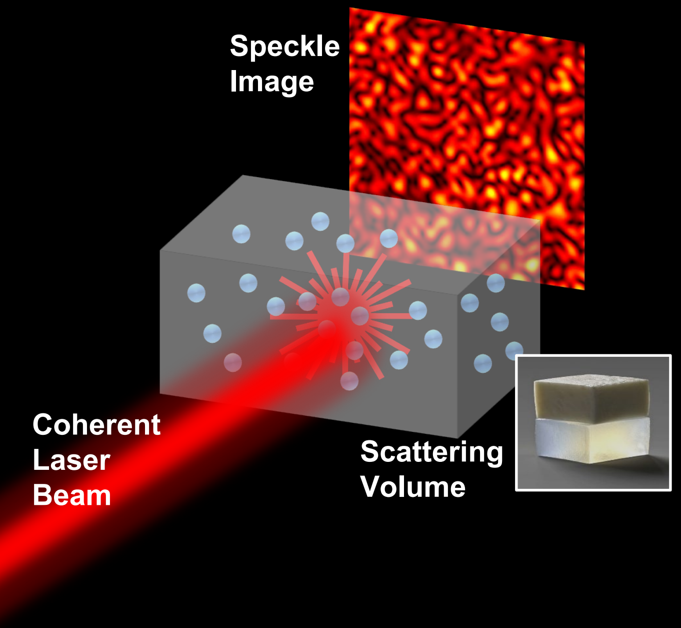
This a
some
E extent
er
mpirically in



Now we can efficiently compute ME curves for all scattering parameters

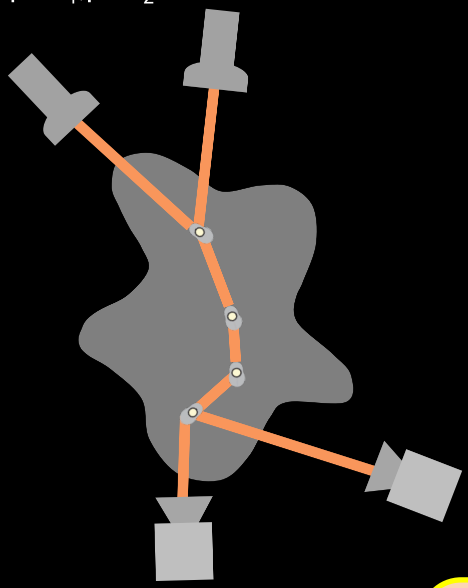
Summary

Problem:
Coherent Scattering

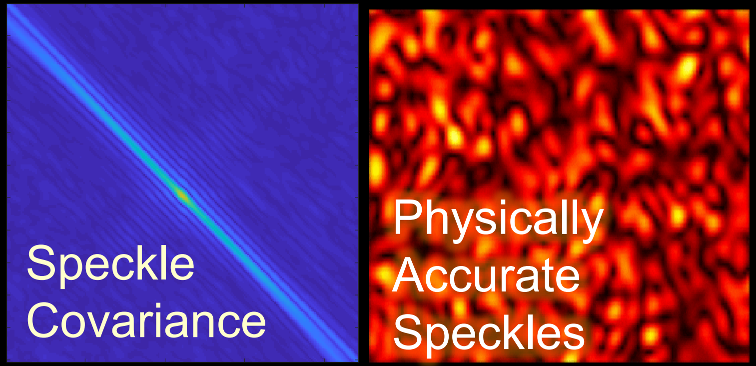


Path-integral formulation for speckle covariance

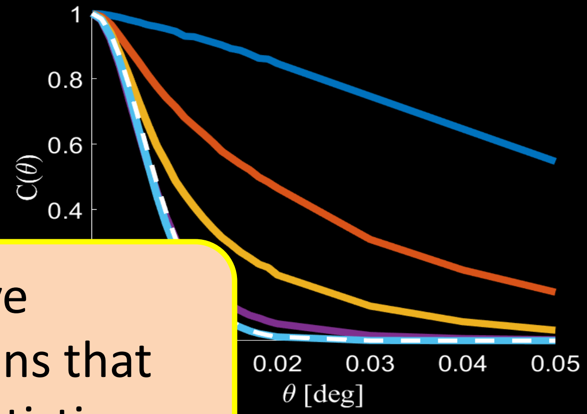
$$\text{Covariance} = \int_{\text{path}_1,\text{path}_2} u(\text{path}_1) \cdot u^*(\text{path}_2)$$



Efficient MC Rendering

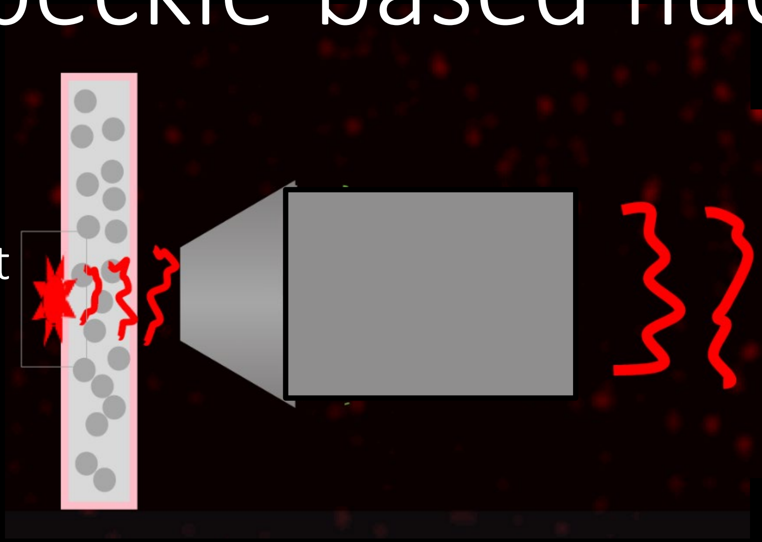


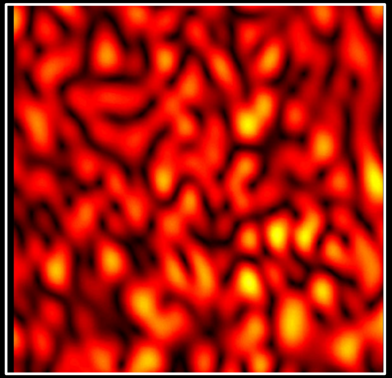
Memory Effect Evaluation

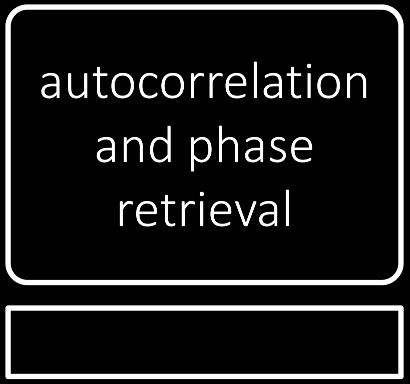


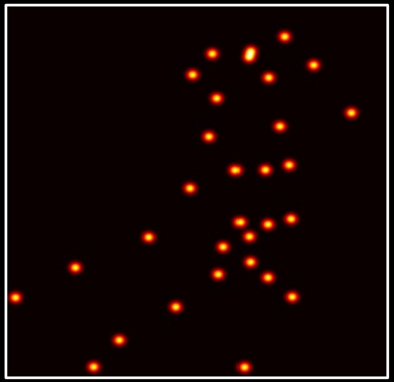
Potentially improve imaging applications that rely on speckle statistics

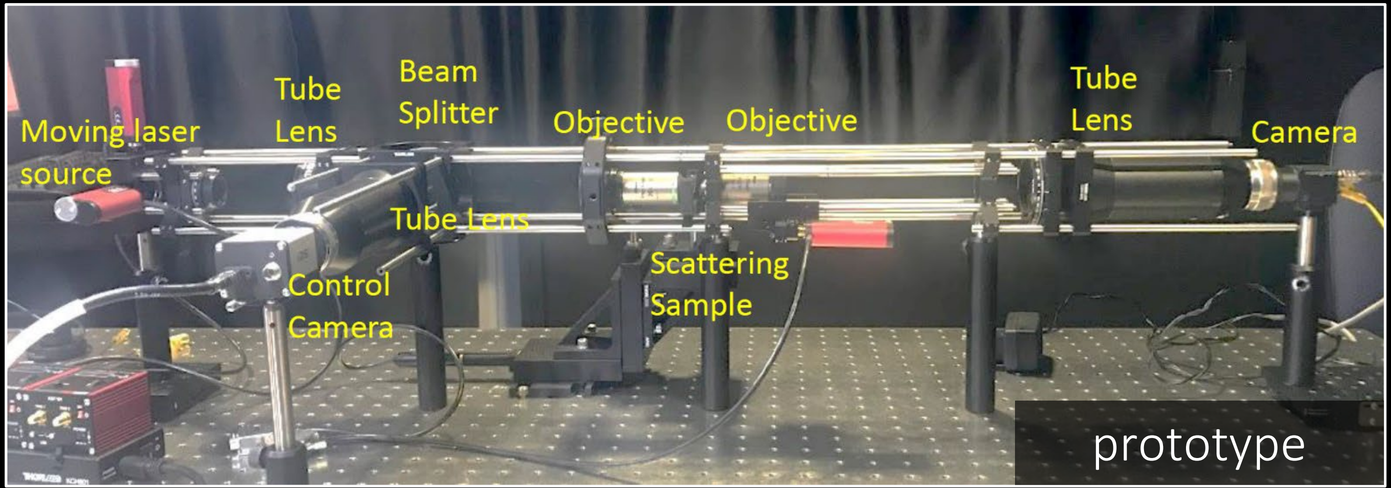
Speckle-based fluorescence microscopy











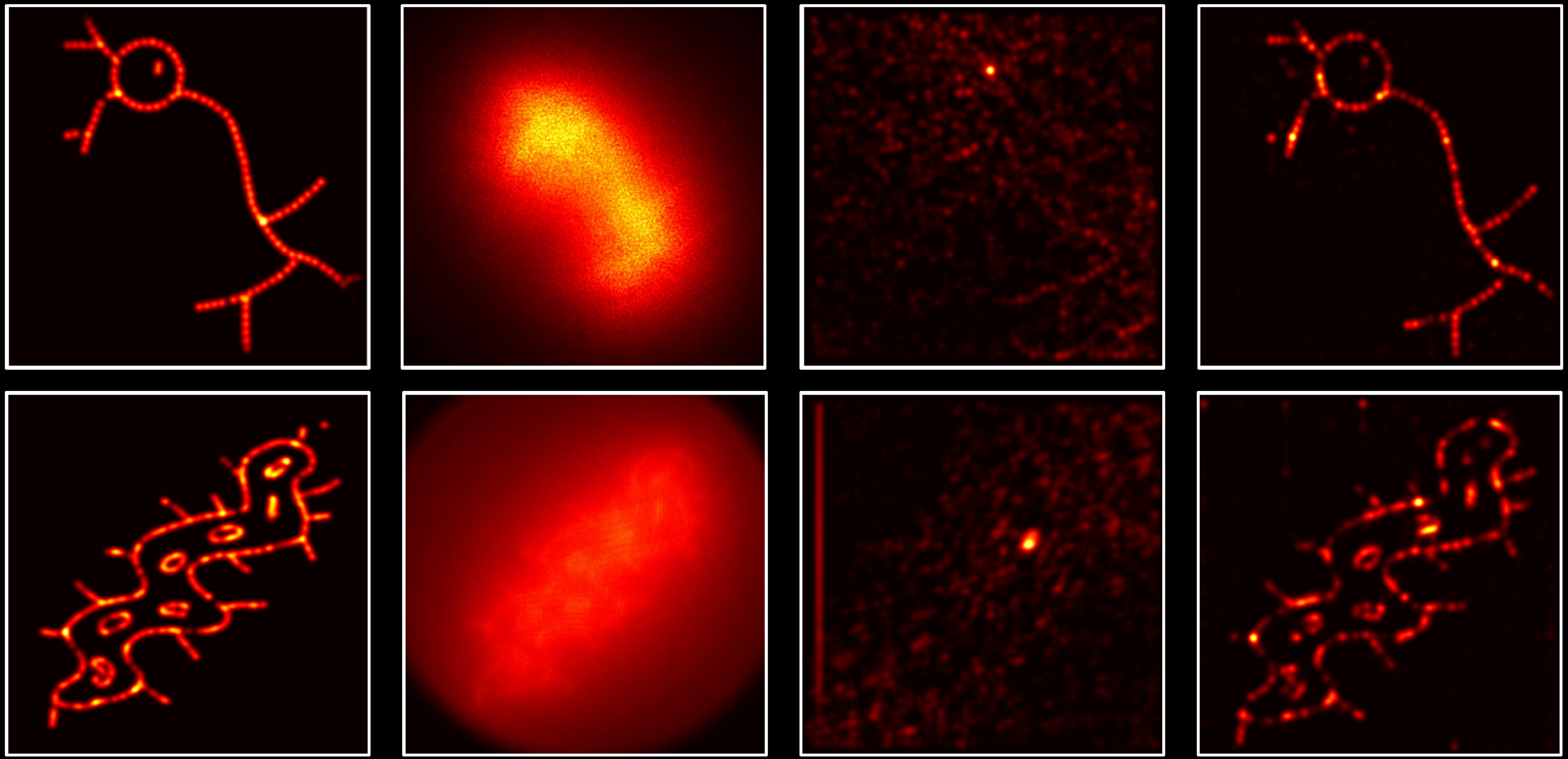
Performance strongly depends on:

- speckle statistics
- image priors
- tissue parameters

[PIs: Gkioulekas, Levin]

Better algorithms for fluorescence microscopy

groundtruth | input image | prior algorithm | our algorithm



[Alterman et al. Transactions on Graphics 2021]

Acquisition of scattering materials

Use differentiable speckle rendering to recover material parameters from speckle images