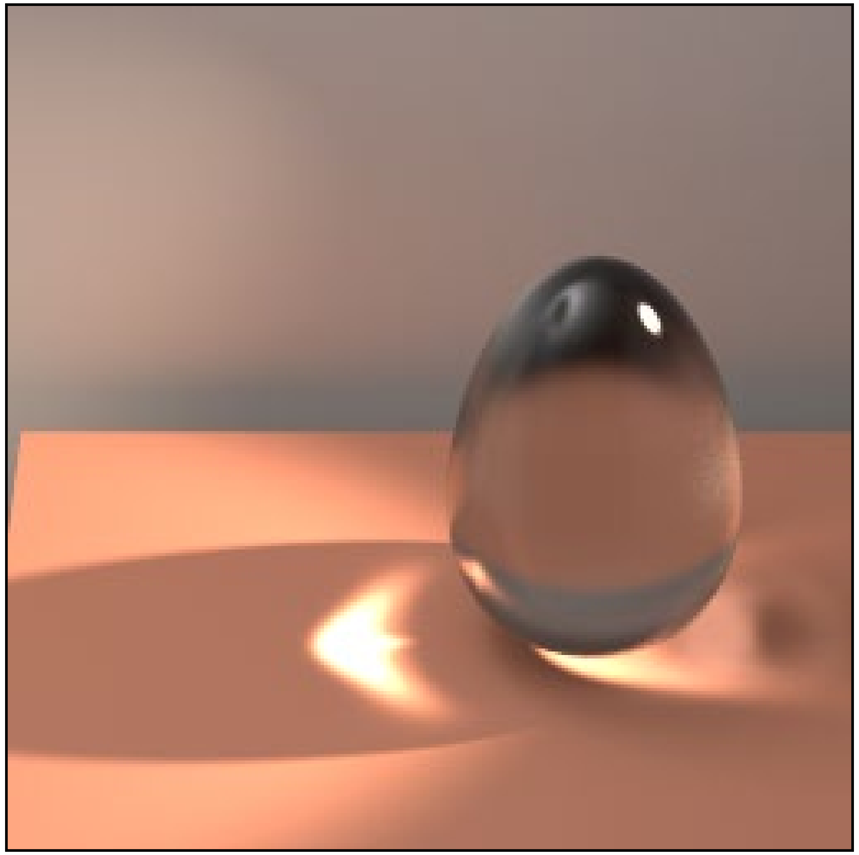
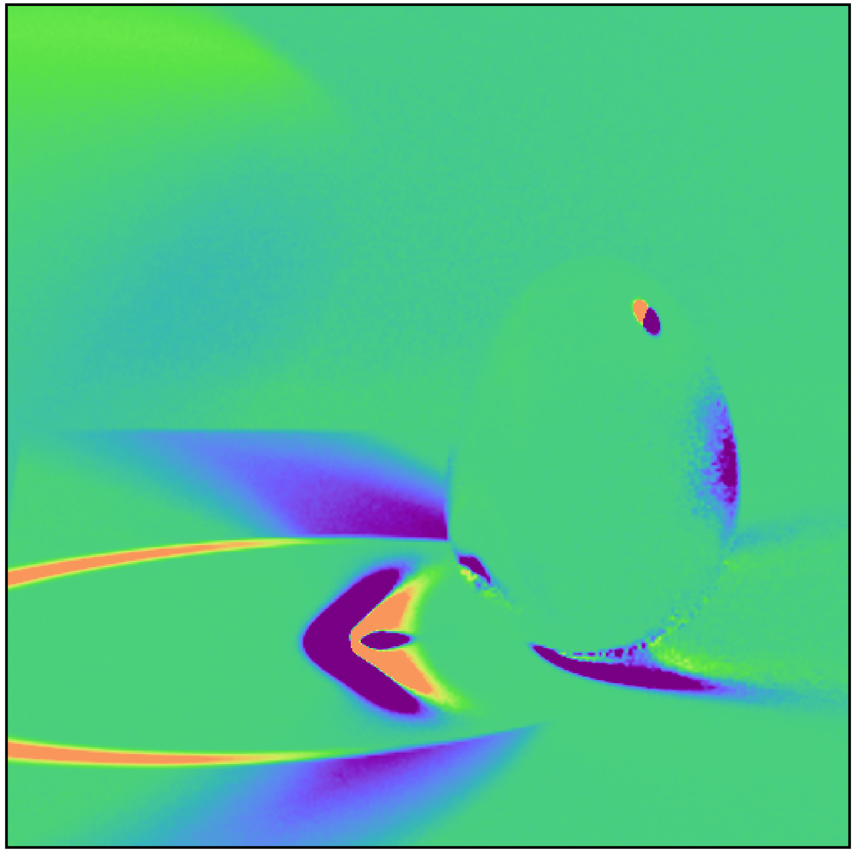
# Inverse and differentiable rendering





15-468, 15-668, 15-868
Physics-based Rendering
Spring 2021, Lecture 22

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

# Course announcements

- Take-home quiz 10 posted, due May $11^{th}$, 11:59 pm.

- Remember: Extra lecture tomorrow, noon – 1:30 pm.

- This week's reading group.
  - We'll cover non-exponential radiative transfer (same topic as last Friday).

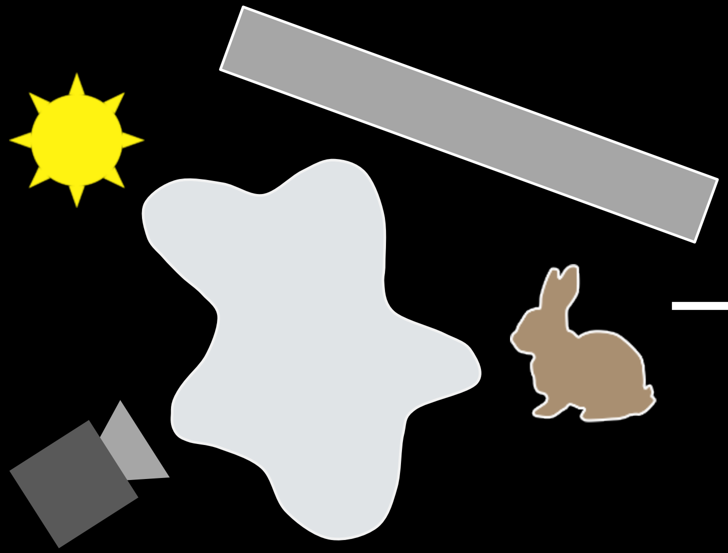
# Take the course evaluation surveys!

- CMU's Faculty Course Evaluations (FCE): https://cmu.smartevals.com/
- CMU's TA Evaluations: https://www.ugrad.cs.cmu.edu/ta/S21/feedback/
- An end-of-semester survey specific to 15-468/668/868: https://docs.google.com/forms/d/e/1FAIpQLSdxnAPIUg-Oy2IUH5OvP7GTRv3XhS0O5P0W4_NlnQp1jQ9X1A/viewform

# Overview of today’s lecture

- Inverse rendering.
- Differentiable rendering.
- Differentiating local parameters.
- Differentiating global parameters.
- Path-space differentiable rendering.
- Reparameterizations.

# Forward rendering

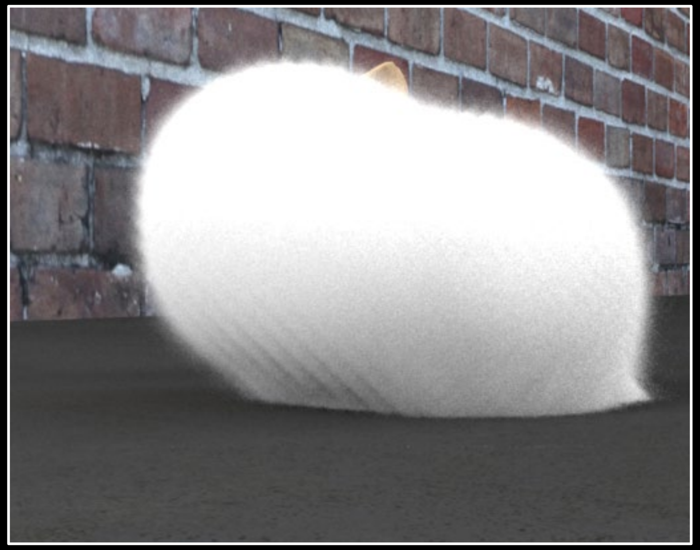


digital scene specification
(geometry, materials,
optics, light sources)

→

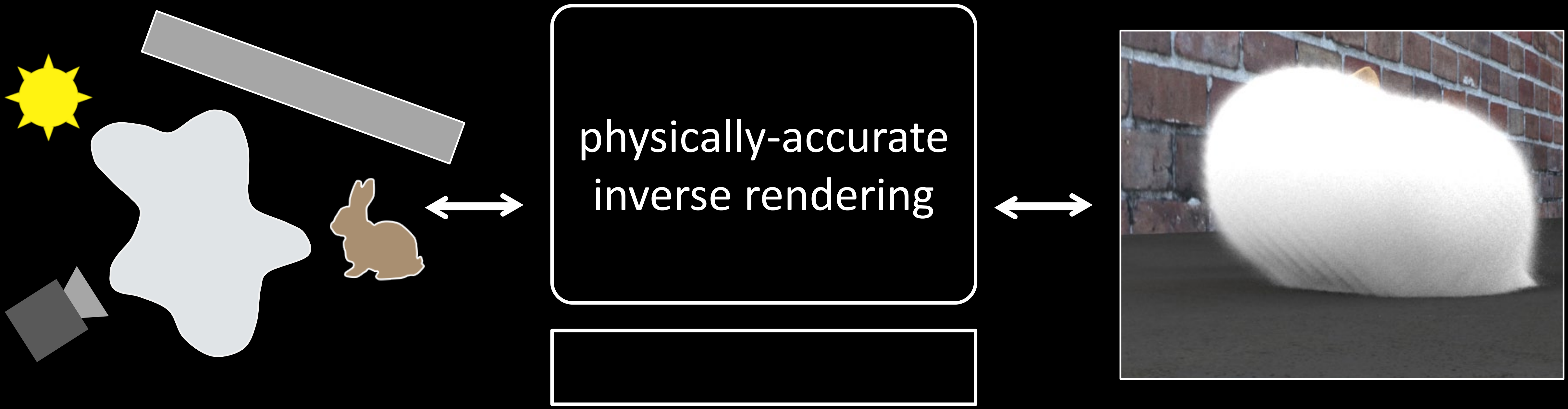
physically-accurate
rendering

→



photorealistic
simulated image

# Inverse rendering



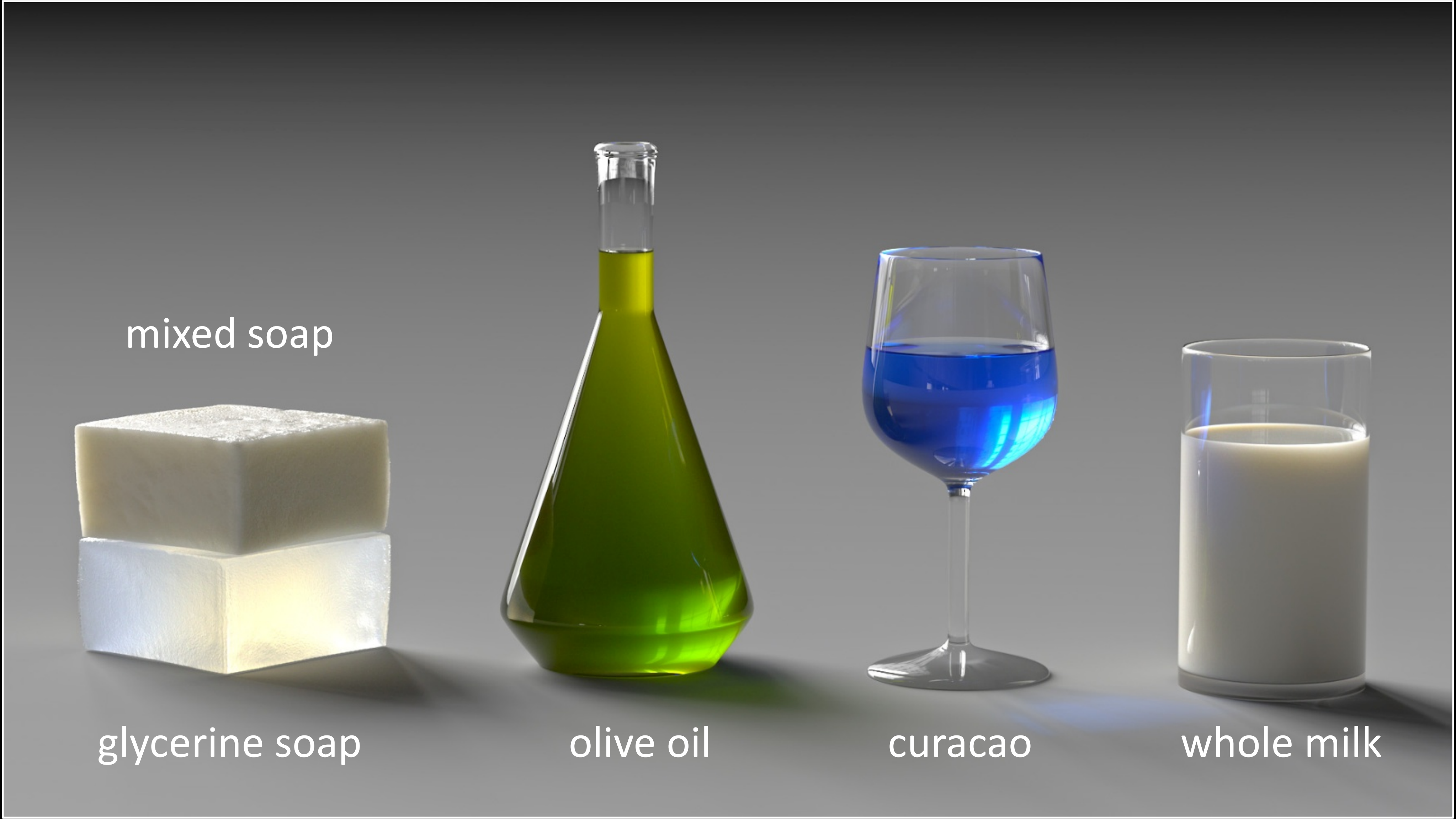

digital scene specification
(geometry, materials,
camera, light sources)

photorealistic
synthetic image

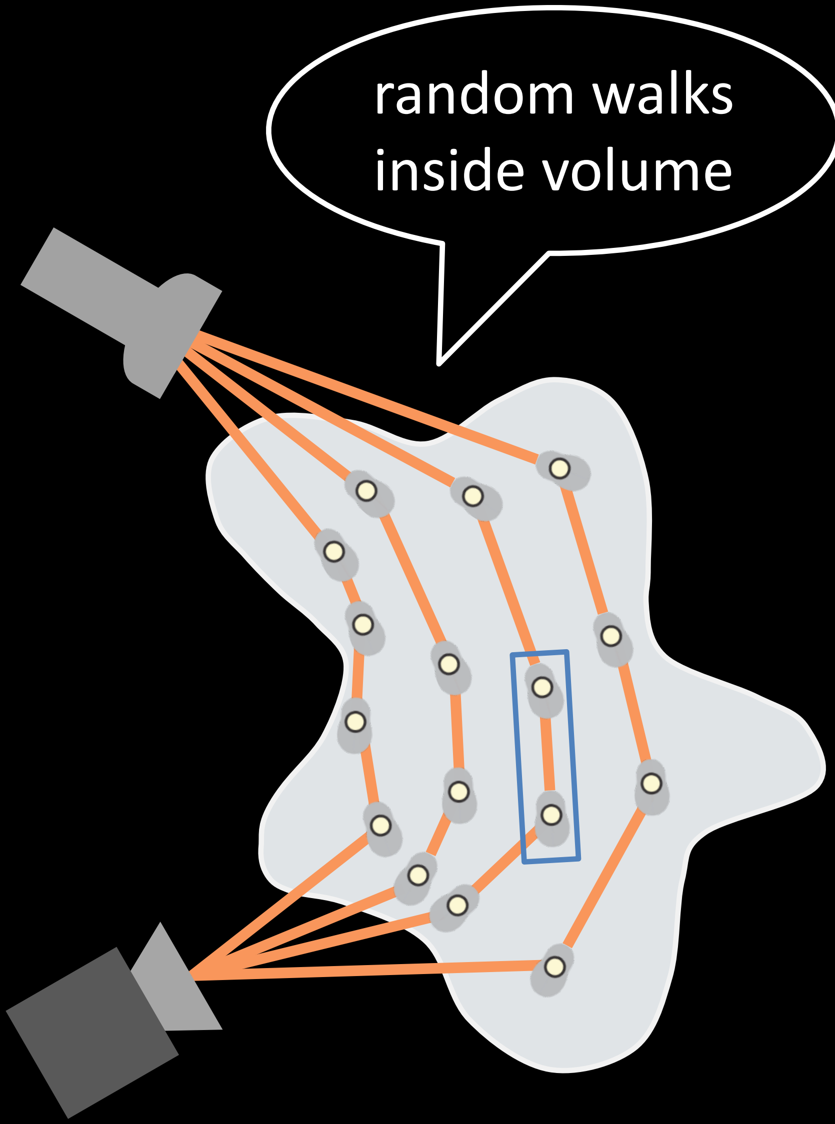
image
measurements

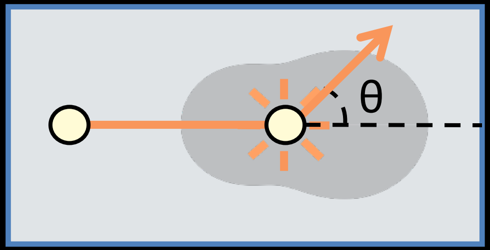
# What I was doing in 2013

# I wanted to make images such as this one



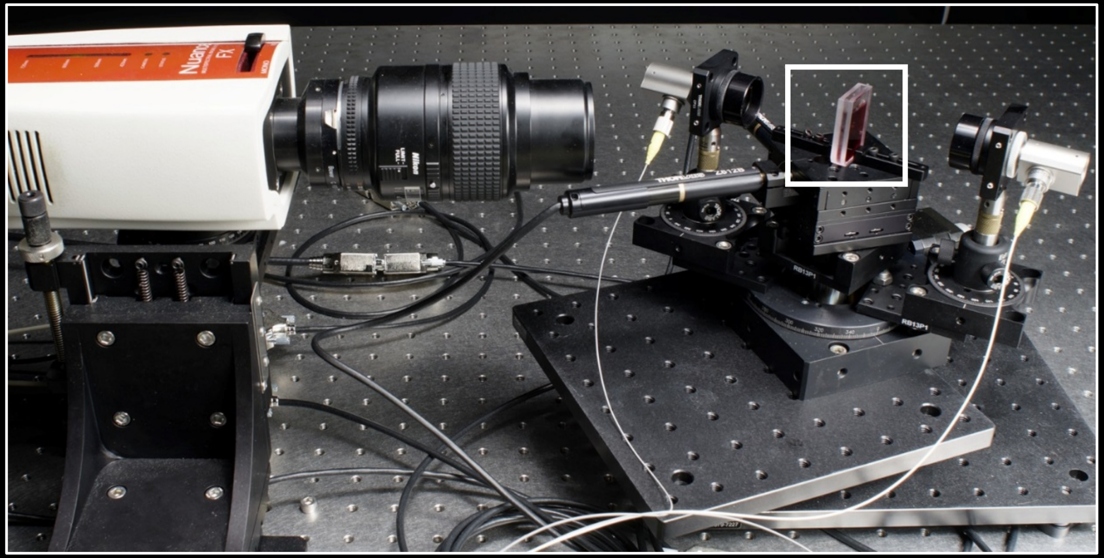
# Scattering: extremely multi-path transport

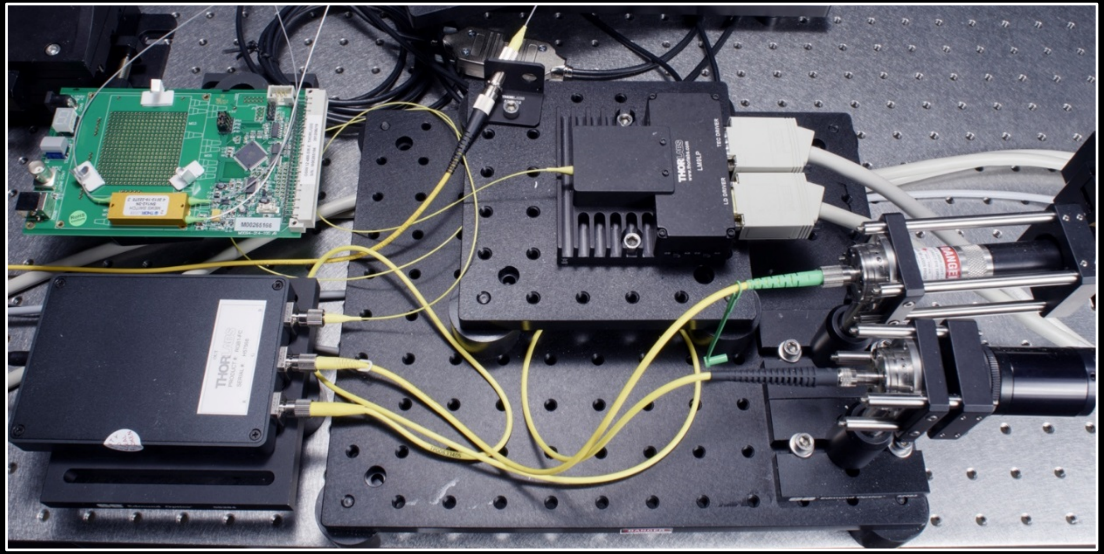


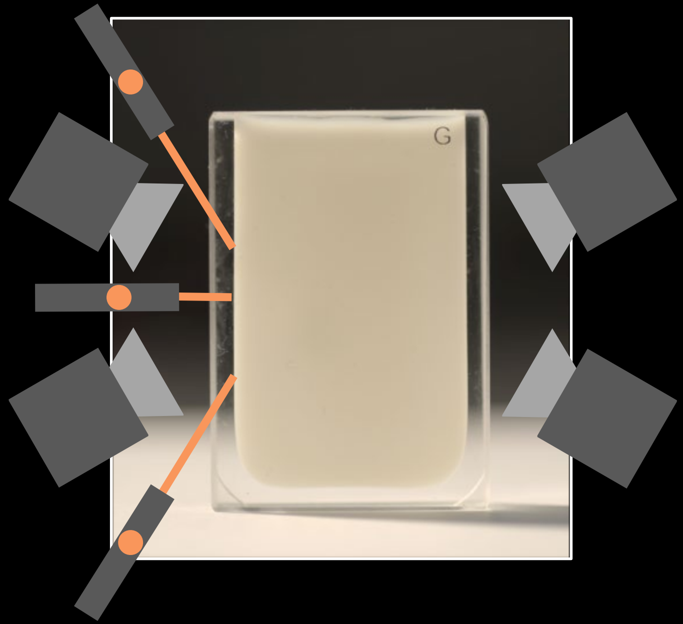



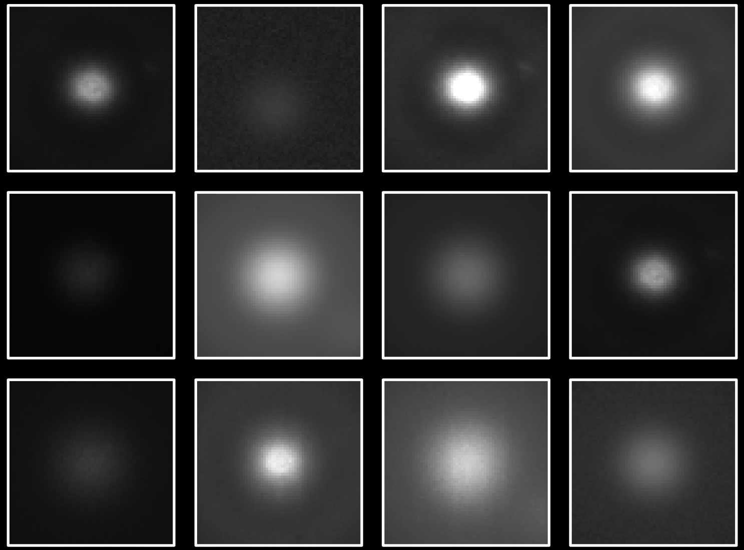

| | |
|---|---|
| volumetric density | $\sigma_t$ |
| scattering albedo | $a$ |
| phase function | $f_r$ |

# Acquisition setup





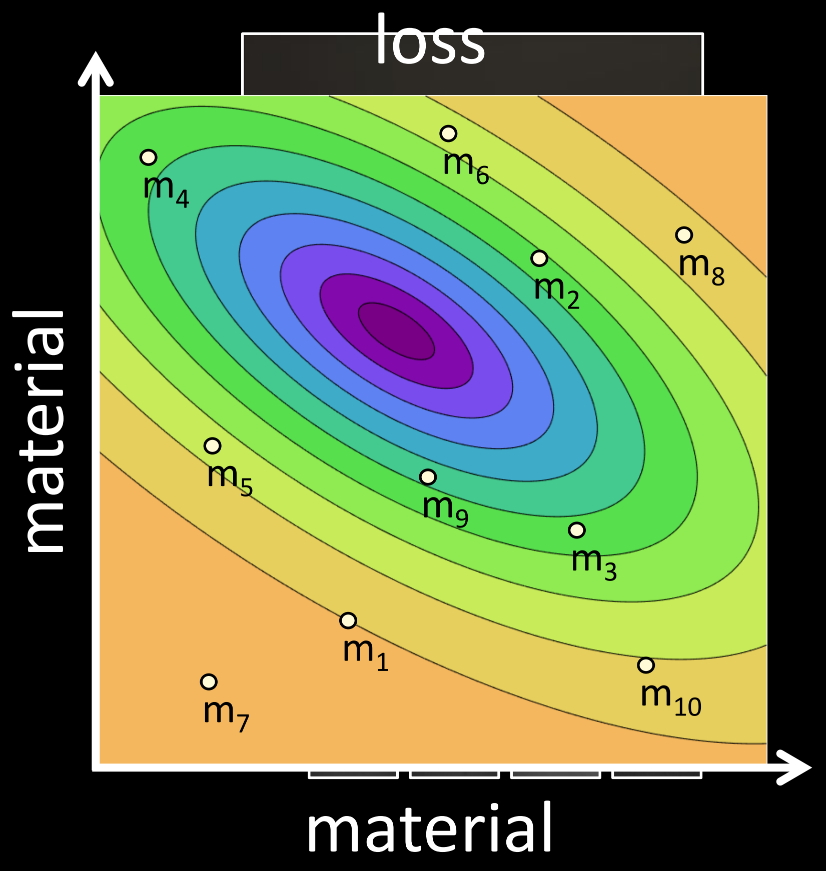




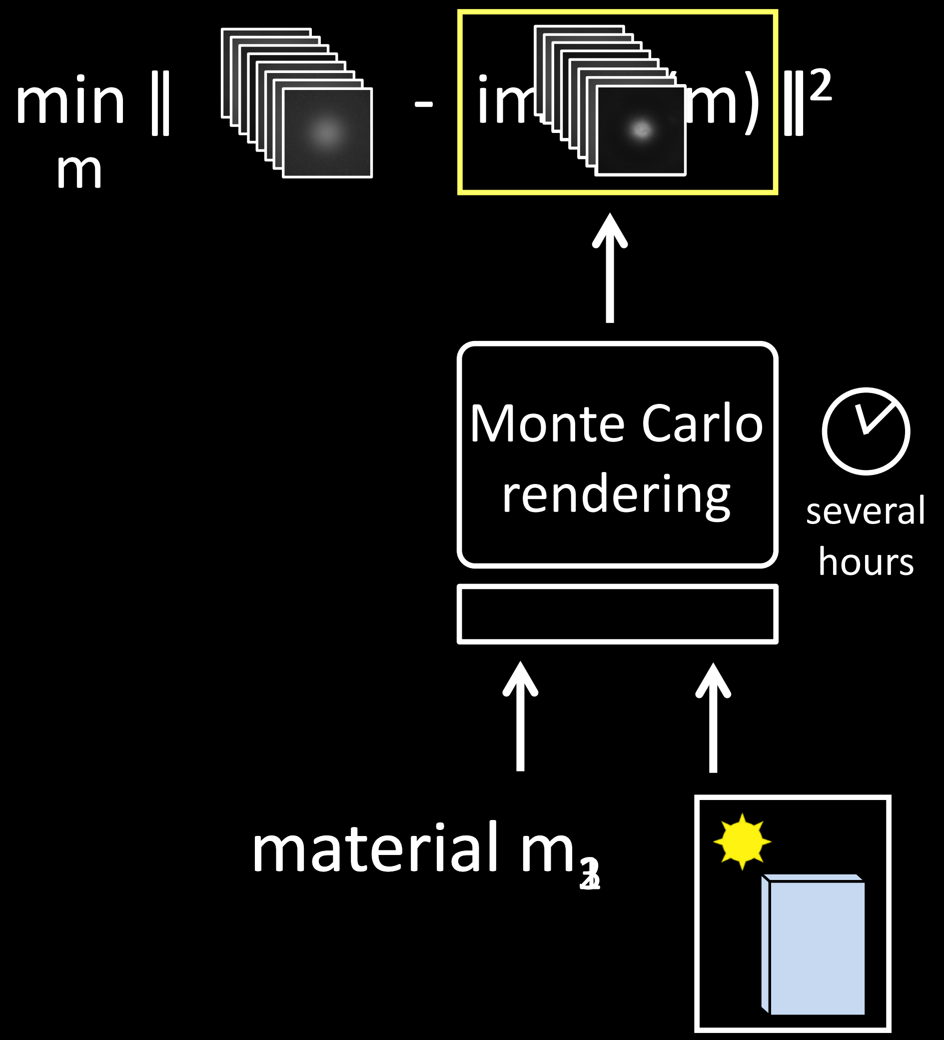
# Analysis by synthesis (a.k.a. inverse rendering)

not scalable

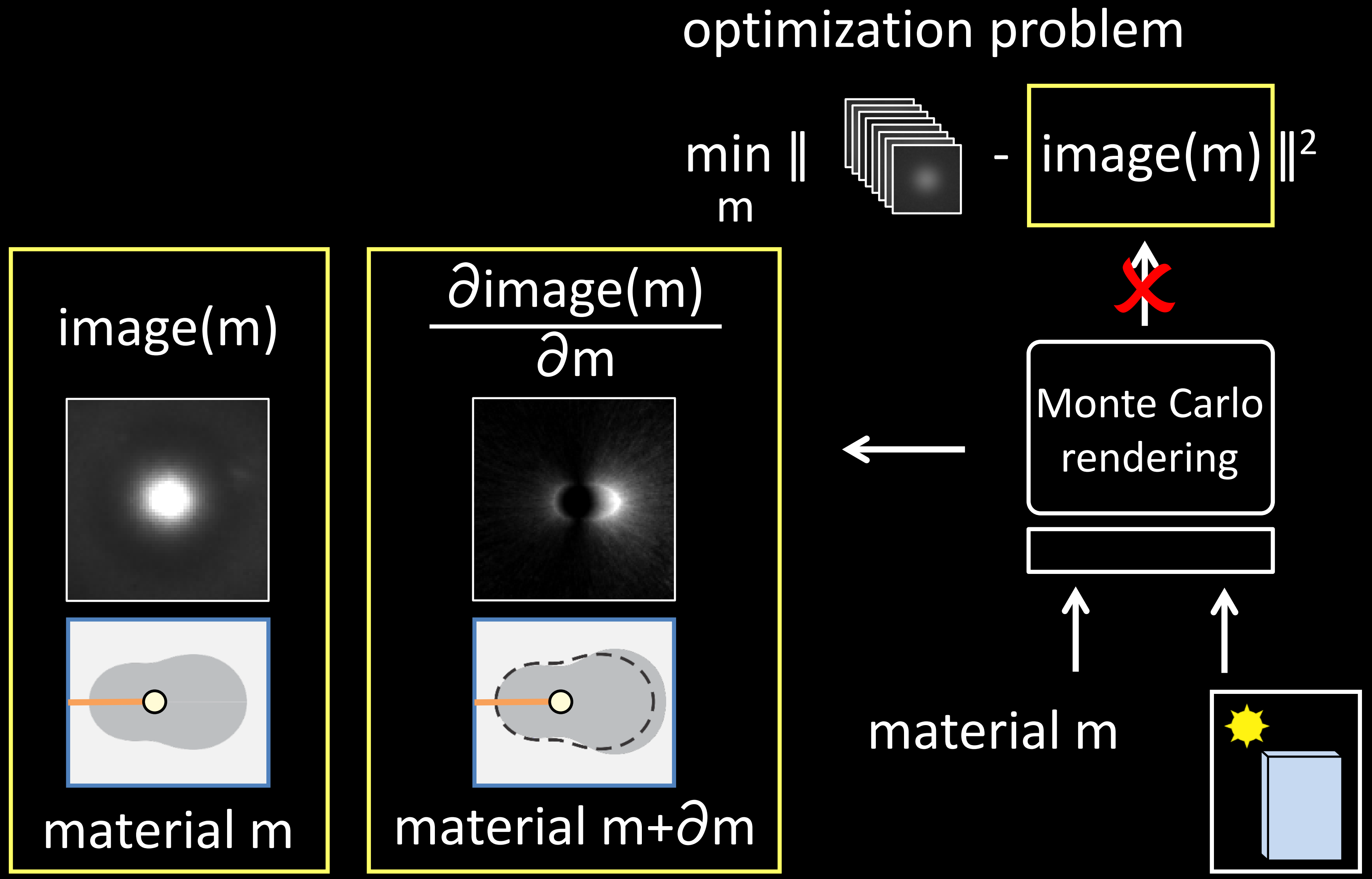
~~solve by exhaustive search?~~




optimization problem



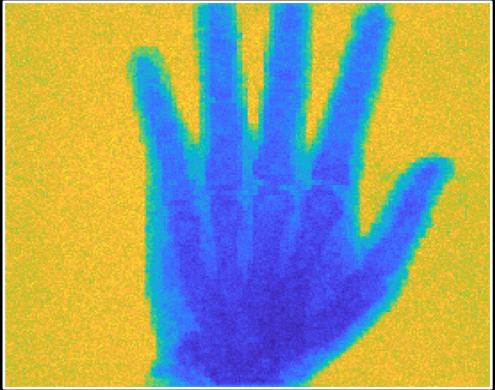
# Analysis by synthesis (a.k.a. inverse rendering)



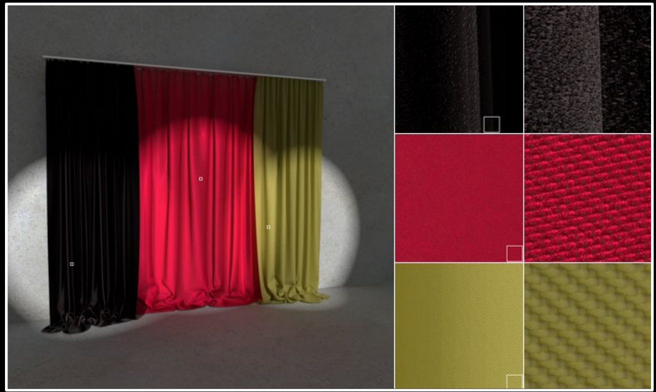
# Other scattering materials

everyday materials
[Gkioulekas et al. 2013]

industrial dispersions
[Gkioulekas et al. 2013]

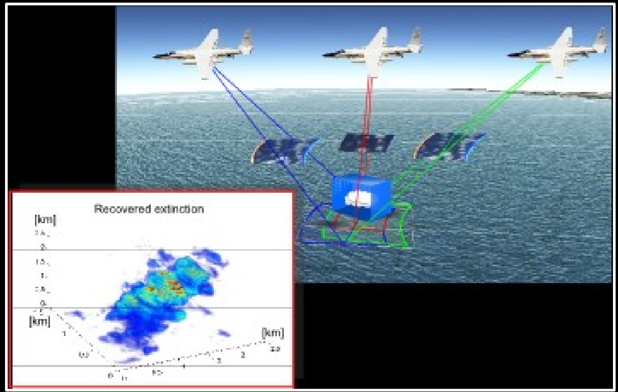


computed tomography
[Geva et al. 2018]



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

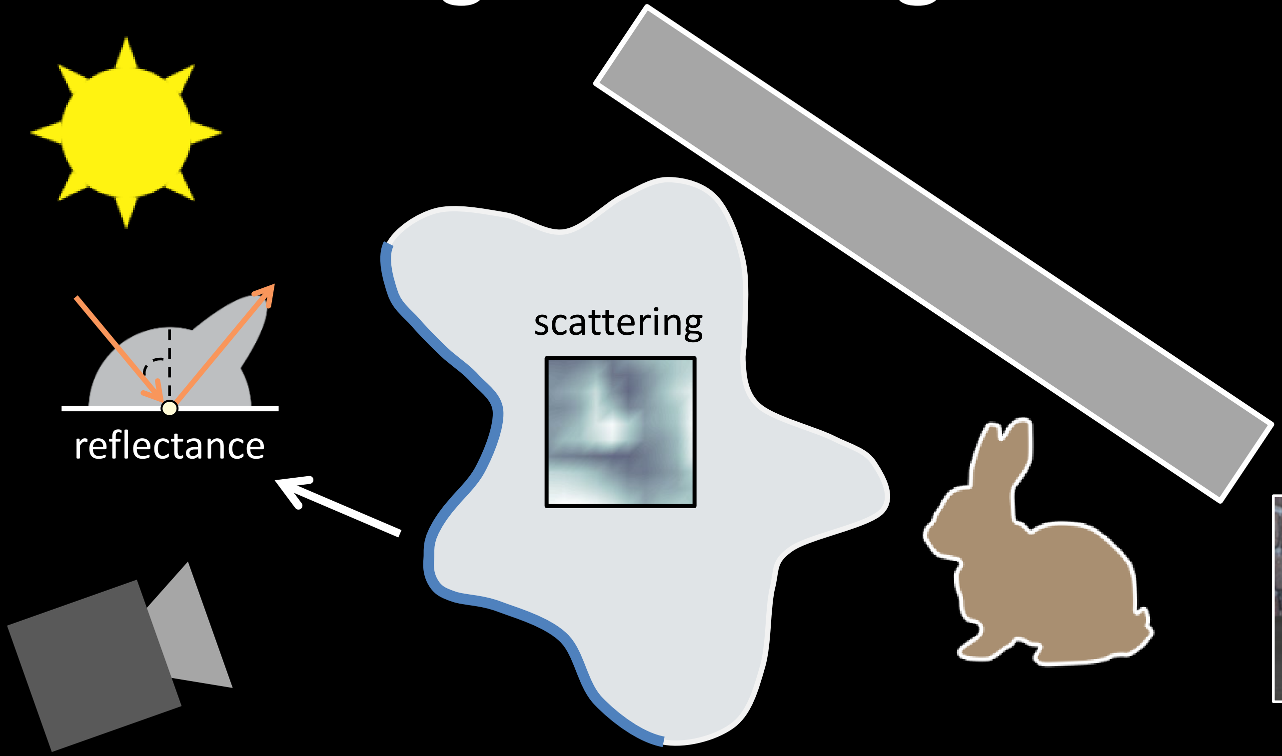
3D printing
[Elek et al. 2017, 2019]




clouds
[Levis et al. 2015, 2017]

optical
tomography
[Gkioulekas et al.
2016]

# Making sense of global illumination

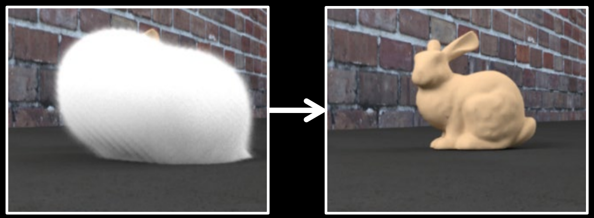



X: 3D shape

X: surface reflectance

X: occluded imaging

X: illumination



analysis by synthesis

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

stochastic gradient descent

while (not converged)

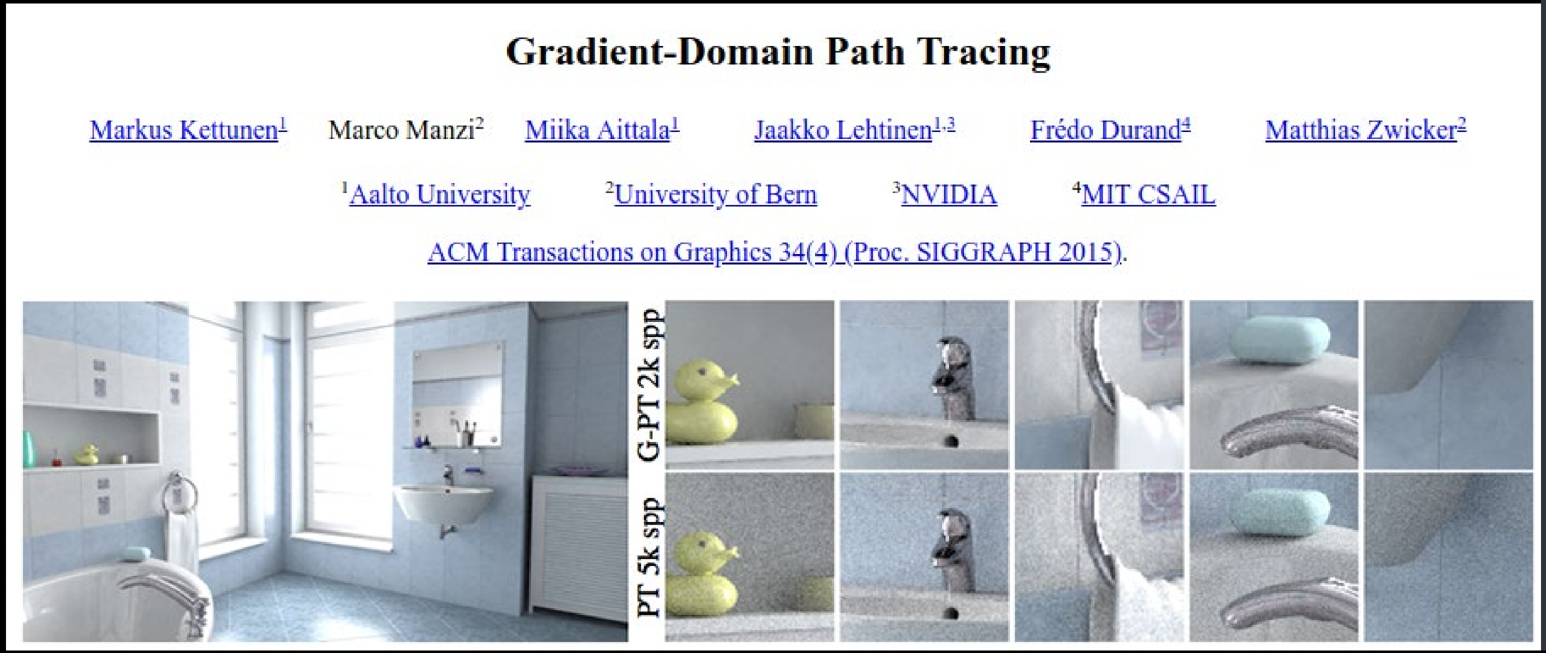
update X with $\dfrac{\partial \widetilde{\text{loss}}(X)}{\partial X}$

Monte-Carlo rendering

differentiable rendering: image gradients with respect to arbitrary X

# Differentiable rendering

## Not related to:





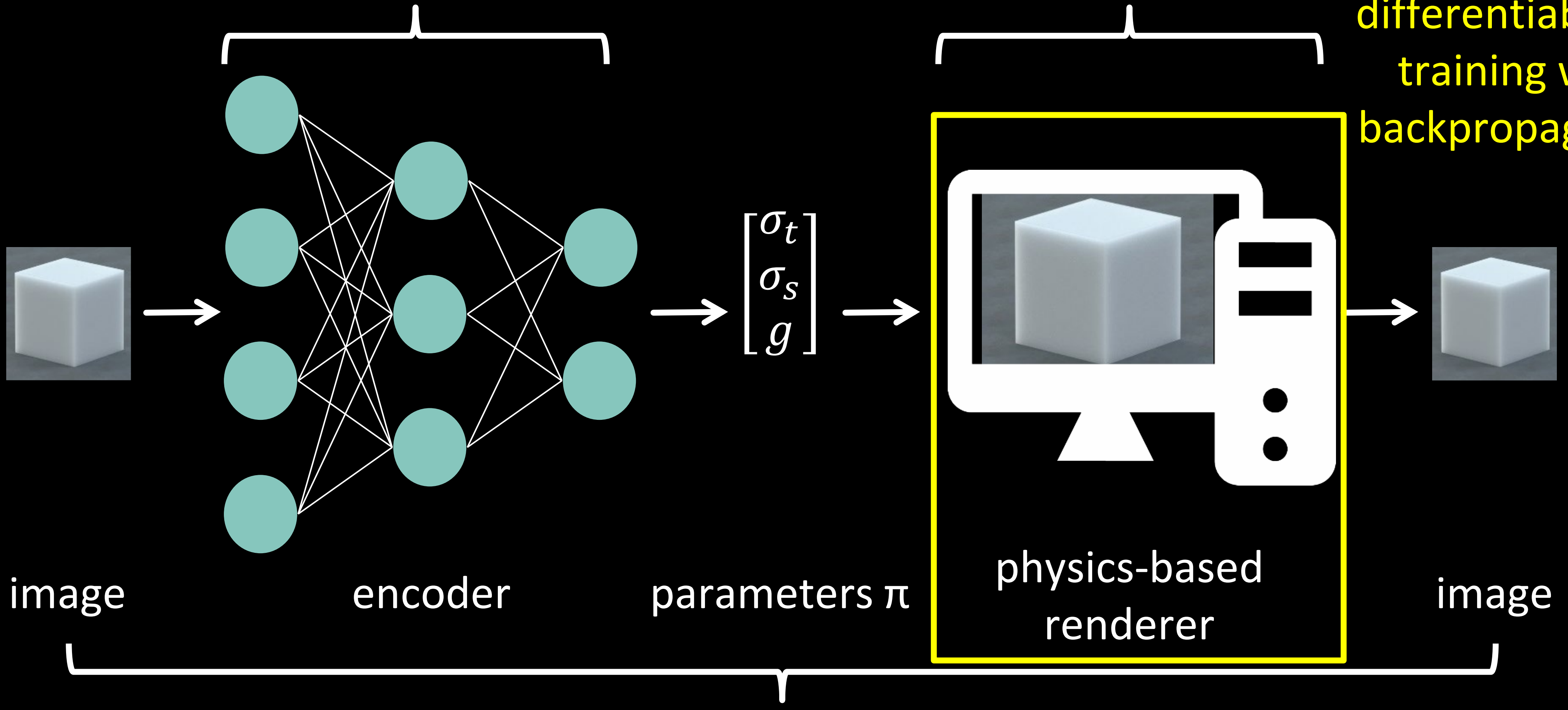
"Gradient" in their case refers to image edges.

# Differentiable rendering and deep learning

$\pi = (R_{physics})^{-1}(Img)$

$Img = R_{physics}(\pi)$

needs to be differentiable for training with backpropagation




force input and output images to be the same

# Quick reminder from calculus

# Basic differentiation rules

$$\frac{\partial}{\partial \pi}\int_a^b f(x;\pi)\mathrm{d}x = ?$$

# Basic differentiation rules

$$\frac{\partial}{\partial\pi}\int_a^b f(x;\pi)\mathrm{d}x = \int_a^b \frac{\partial}{\partial\pi} f(x;\pi)\mathrm{d}x$$

what is this rule called?

# Basic differentiation rules

$$\frac{\partial}{\partial \pi} \int_a^b f(x;\pi)\mathrm{d}x = \int_a^b \frac{\partial}{\partial \pi} f(x;\pi)\mathrm{d}x$$

differentiation under the integral sign

$$\frac{\partial}{\partial \pi} \int_{a(\pi)}^{b(\pi)} f(x;\pi)\mathrm{d}x = ?$$

# Basic differentiation rules

$$\frac{\partial}{\partial \pi}\int_a^b f(x;\pi)\mathrm{d}x = \int_a^b \frac{\partial}{\partial \pi} f(x;\pi)\mathrm{d}x$$

differentiation under the integral sign

$$\frac{\partial}{\partial \pi}\int_{a(\pi)}^{b(\pi)} f(x;\pi)\mathrm{d}x = \int_{a(\pi)}^{b(\pi)} \frac{\partial}{\partial \pi} f(x;\pi)\mathrm{d}x$$
$$+ f(b(\pi);\pi)\frac{\partial b(\pi)}{\partial \pi} - f(\alpha(\pi);\pi)\frac{\partial \alpha(\pi)}{\partial \pi}$$

what is this rule called?

# Basic differentiation rules

$$\frac{\partial}{\partial \pi}\int_a^b f(x;\pi)\mathrm{d}x = \int_a^b \frac{\partial}{\partial \pi} f(x;\pi)\mathrm{d}x$$

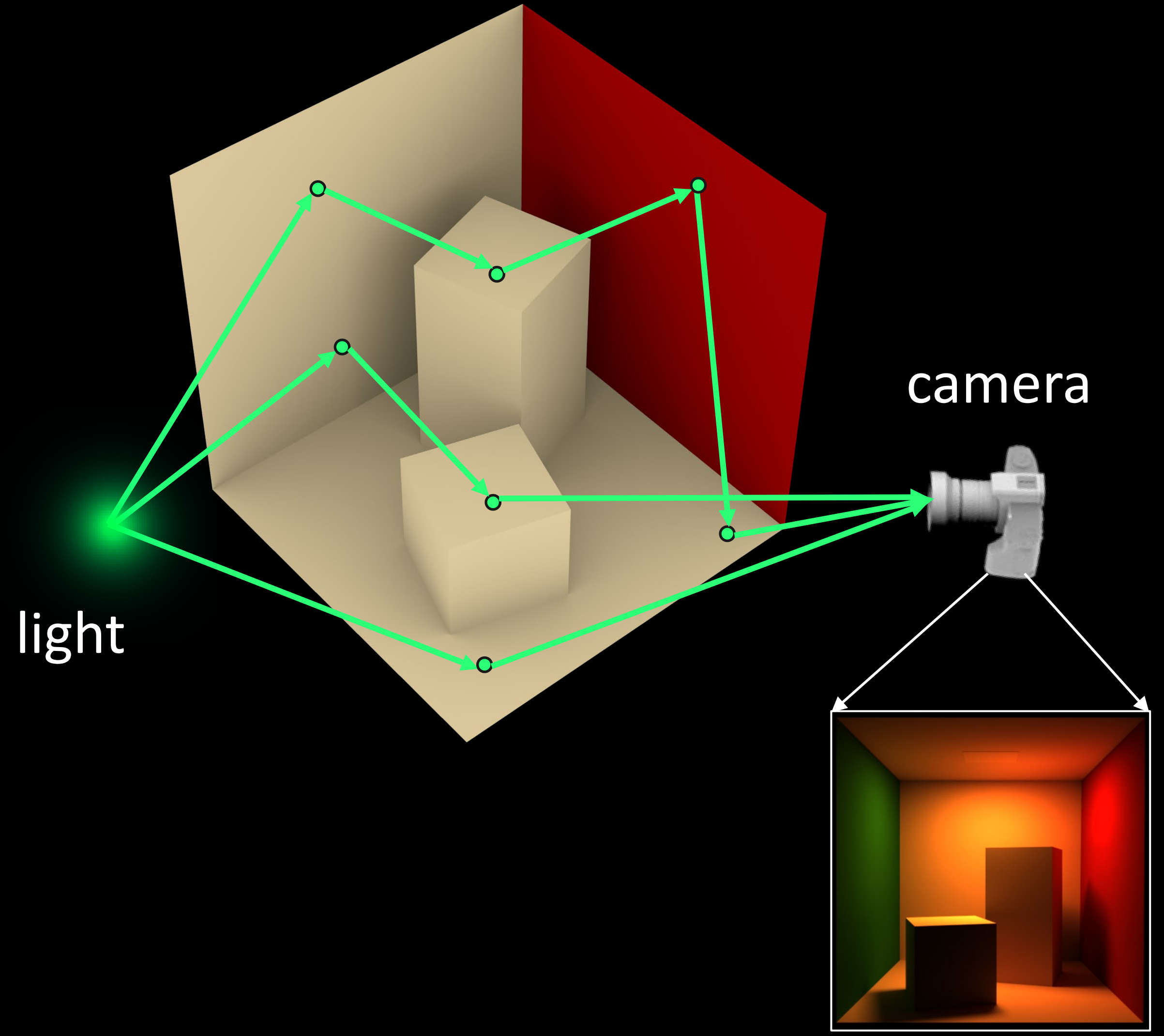
differentiation under the integral sign

$$\frac{\partial}{\partial \pi}\int_{a(\pi)}^{b(\pi)} f(x;\pi)\mathrm{d}x = \int_{a(\pi)}^{b(\pi)} \frac{\partial}{\partial \pi} f(x;\pi)\mathrm{d}x$$
$$+ f(b(\pi);\pi)\frac{\partial b(\pi)}{\partial \pi} - f(\alpha(\pi);\pi)\frac{\partial \alpha(\pi)}{\partial \pi}$$

Leibniz integral rule

# Trivial differentiable rendering

# Images as path integrals




$$I(\pi) = \int_{\mathbb{P}} f(\bar{\mathbf{x}}; \pi) \mathrm{d}\bar{\mathbf{x}}$$

$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → Space of valid paths

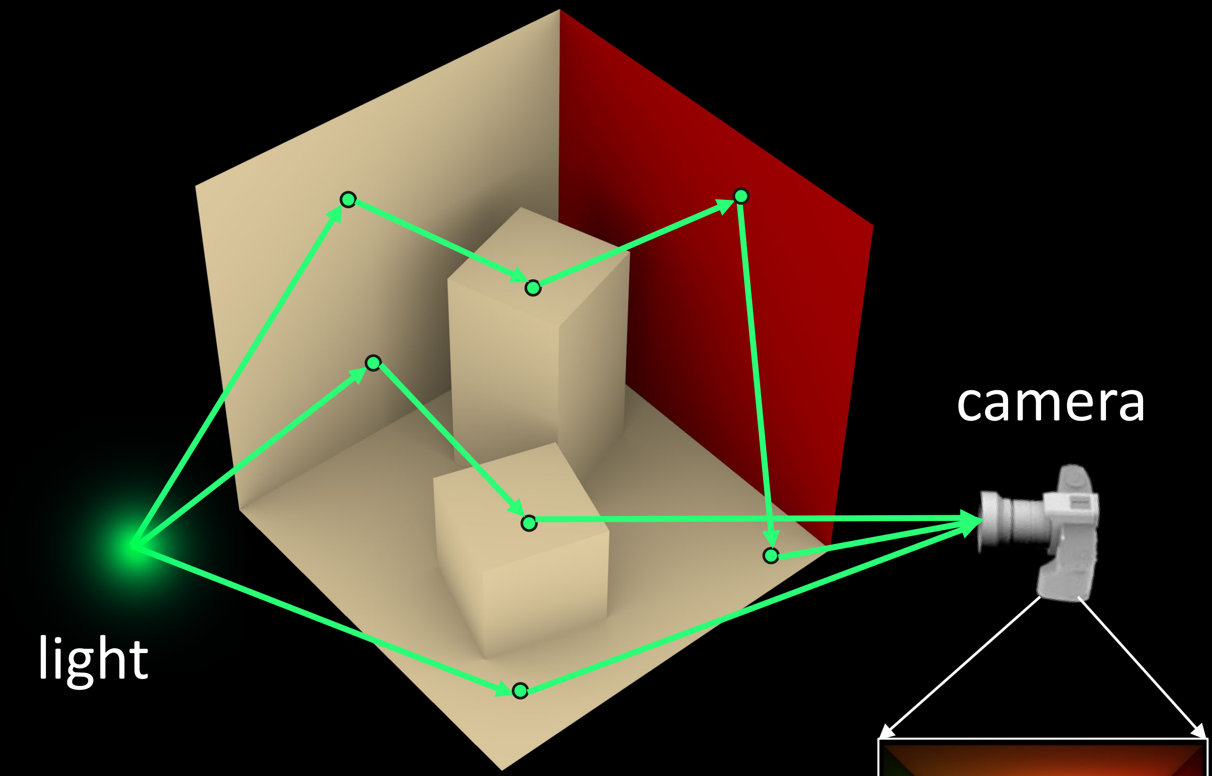
$f(\bar{\mathbf{x}})$ → Path contribution,
includes geometric terms (visibility, fall-off) &
local terms (BRDF, foreshortening, emmision)

$\bar{x} = x_0 \to x_1 \to \dots \to x_B$

$f(\bar{x}) = W_e(x_0 \to x_1) L_e(x_{B-1} \to x_B)$

$\prod f(x_{i-1} \to x_i \to x_{i+1}) G(x_i$

# Monte Carlo rendering: approximating path integrals



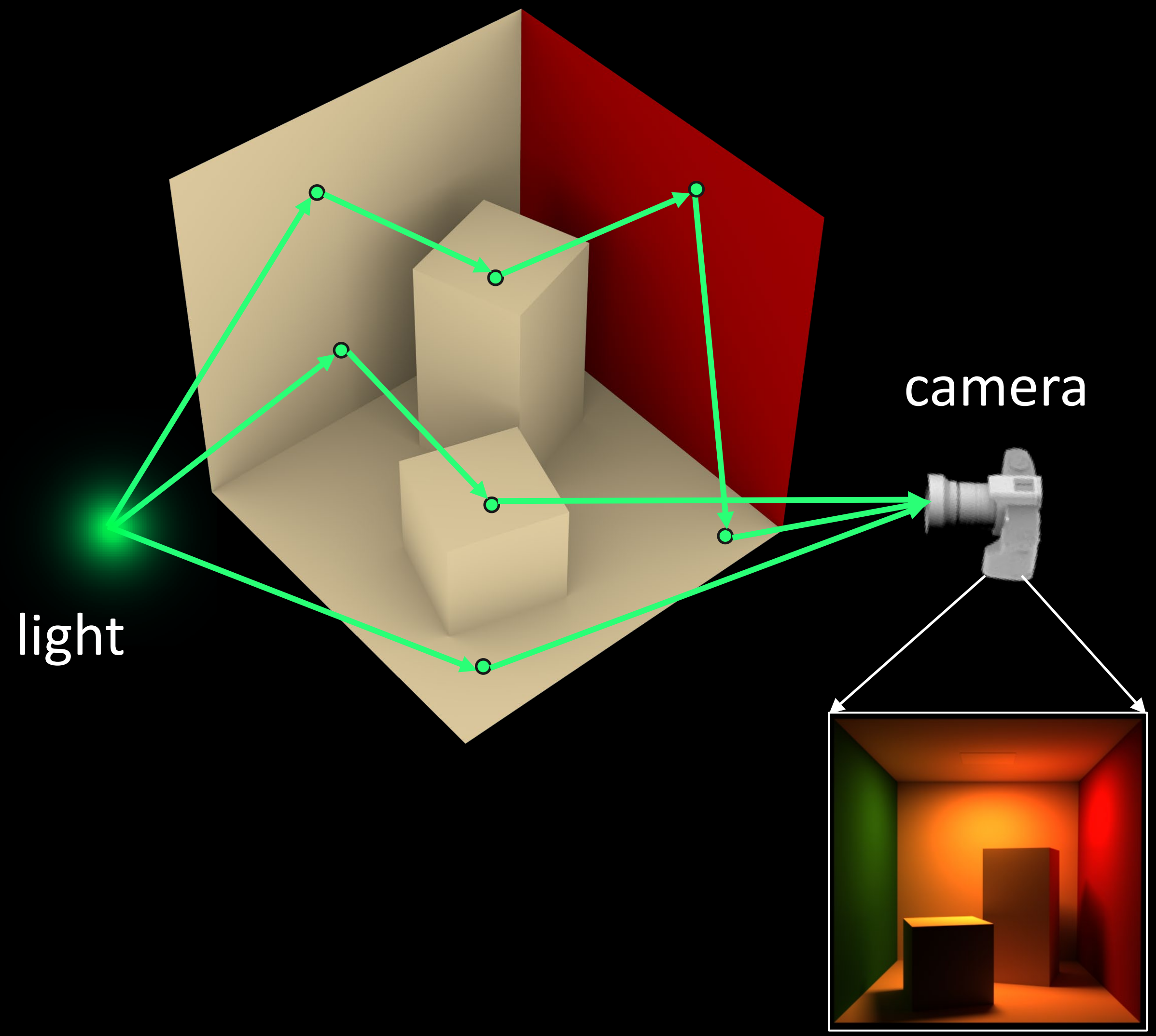

$$I(\pi) \approx \underbrace{\sum_{i=1}^{N} \frac{f(\bar{\mathbf{x}}_i;\pi)}{p(\bar{\mathbf{x}}_i)}}_{MC(\pi)}$$

$\bar{x}_i$ → Randomly sampled light paths

$p(\bar{\mathbf{x}}_i)$ → Probability of sampling a path

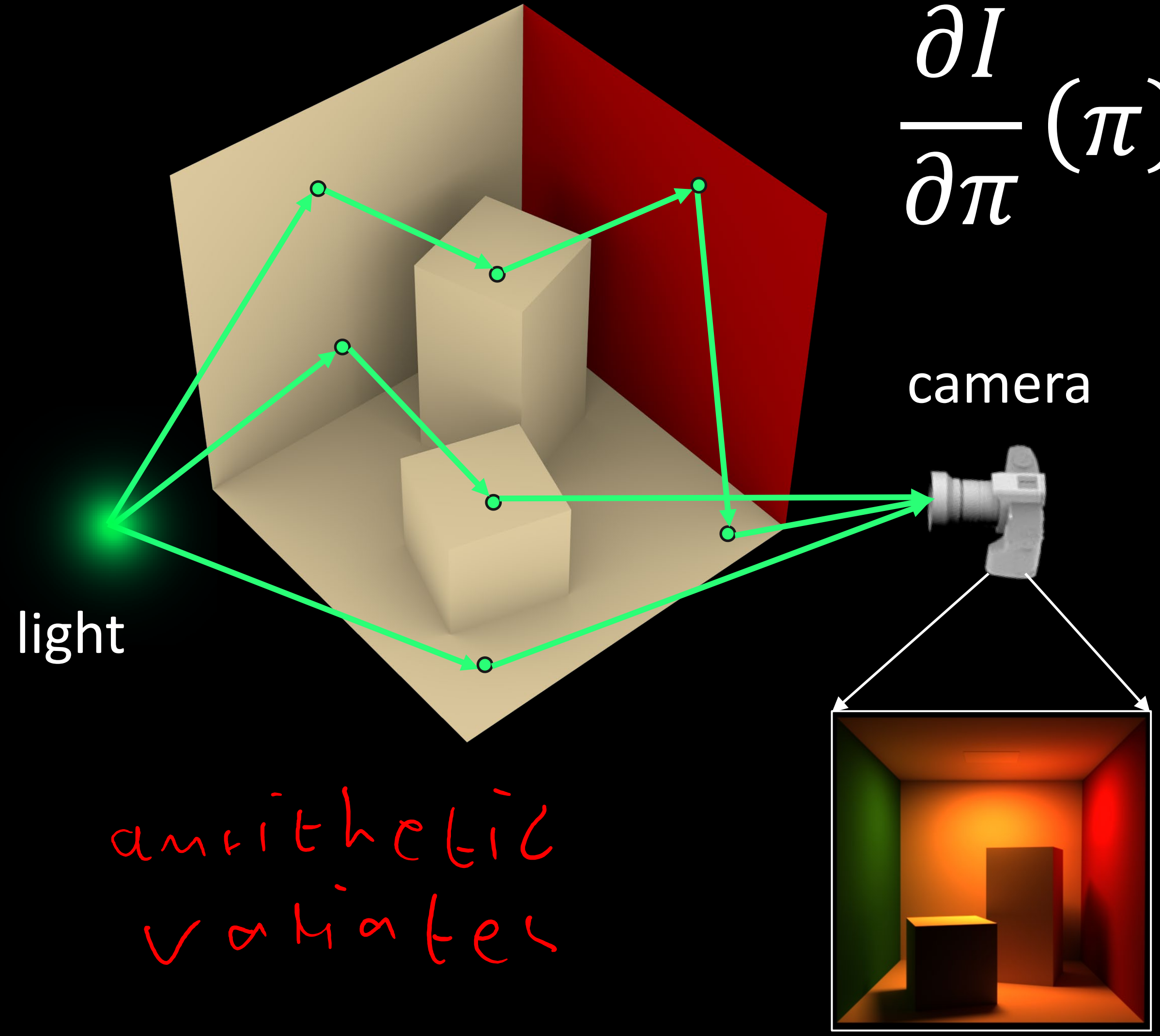
Algorithms such as path tracing, bidirectional path tracing, etc. sample paths.

# How can we approximate the derivative of the image?




$$\frac{\partial I}{\partial \pi}(\pi) \approx ?$$

# Easy approach 1: finite differences



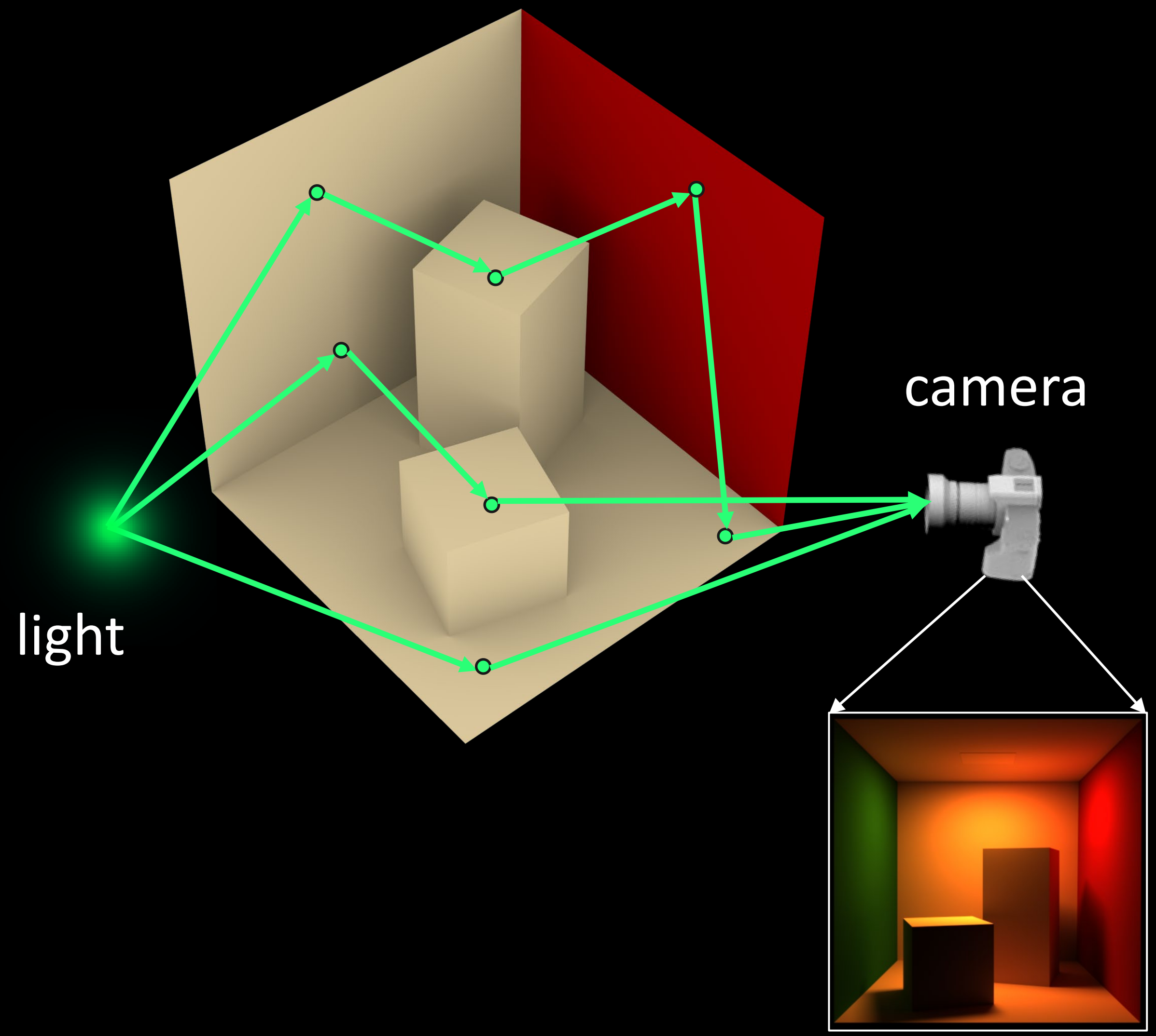

antithetic variates

$$\frac{\partial I}{\partial \pi}(\pi) \approx \frac{MC(\pi+\varepsilon) - MC(\pi-\varepsilon)}{2\varepsilon}$$

Any issues with this?

- <u>Incredibly</u> noisy for small ε
- Very inaccurate for large ε
- Techniques for noise reduction exist, but generally impractical approach

# Easy approach 2: automatic differentiation




$$\frac{\partial I}{\partial \pi}(\pi) \approx \mathrm{autodiff}(MC(\pi))$$

Any issues with this?

- Many path sampling techniques are not differentiable
- High variance (consider f(x;π) = constant)
- Rendering produces enormous, non-local computational graphs.

# OpenDR: An Approximate Differentiable Renderer

[Loper and Black 2015]

- Only direct illumination.
- Only shading parameters (normals, reflectance).

**Abstract.** Inverse graphics attempts to take sensor data and infer 3D geometry, illumination, materials, and motions such that a graphics renderer could realistically reproduce the observed scene. Renderers, however, are designed to solve the forward process of image synthesis. To go in the other direction, we propose an approximate *differentiable renderer (DR)* that explicitly models the relationship between changes in model parameters and image observations. We describe a publicly available *OpenDR* framework that makes it easy to express a forward graphics model and then automatically obtain derivatives with respect to the model parameters and to optimize over them. Built on a new autodifferentiation package and OpenGL, OpenDR provides a local optimization method that can be incorporated into probabilistic programming frameworks. We demonstrate the power and simplicity of programming with OpenDR by using it to solve the problem of estimating human body shape from Kinect depth and RGB data.

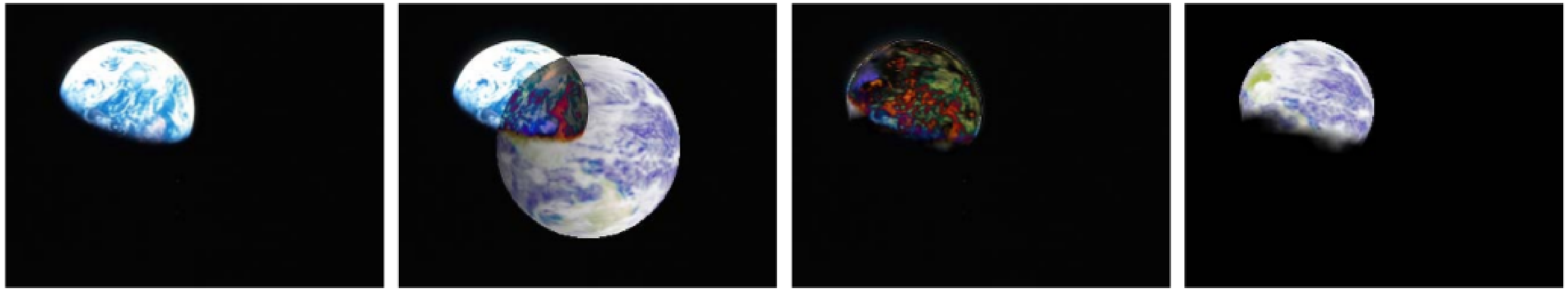
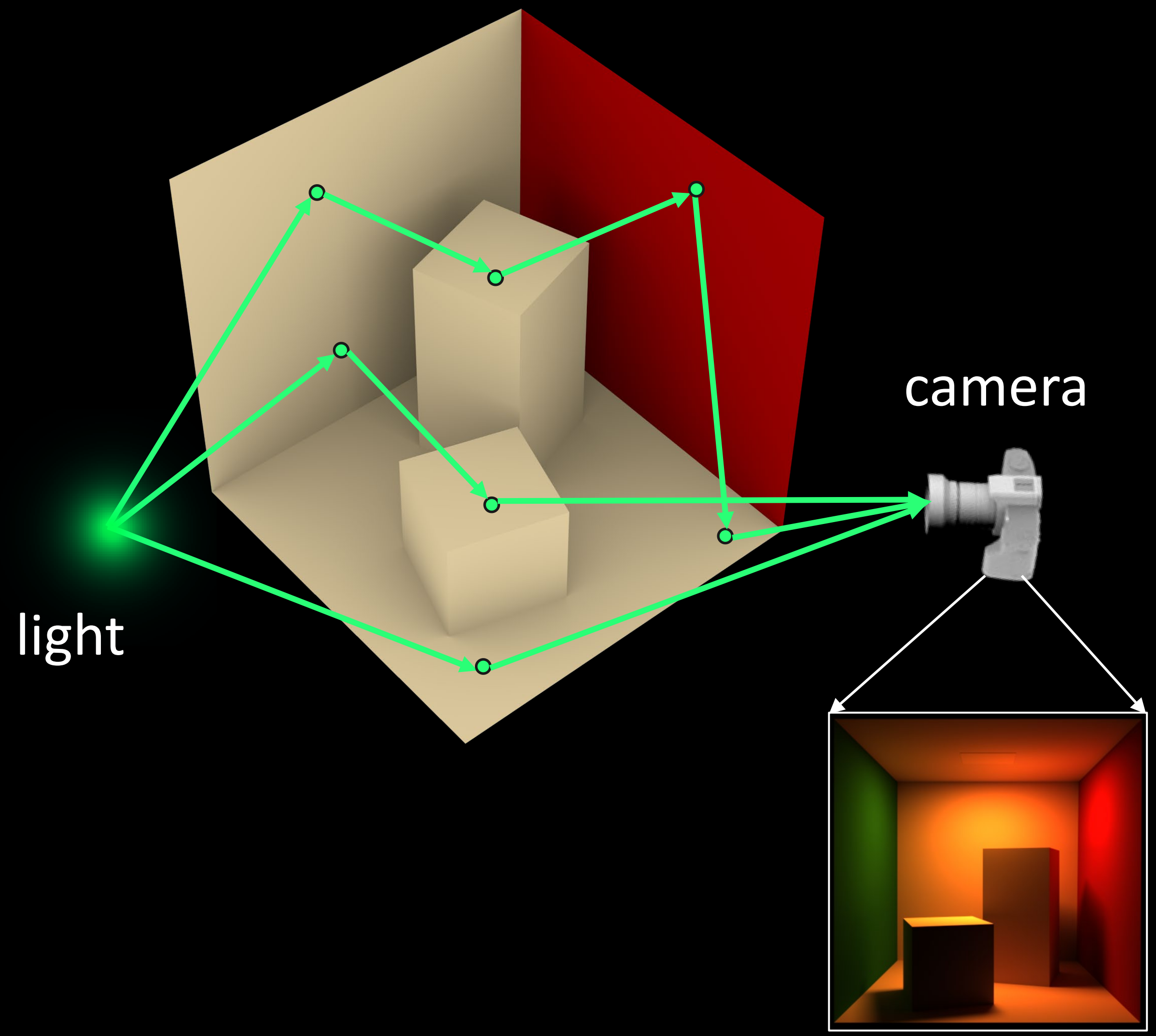


**Fig. 4.** Illustration of optimization in Figure 3. In order: observed image of earth, initial absolute difference between the rendered and observed image intensities, final difference, final result.

# Differentiable rendering for local parameters

# Images as path integrals




$$I(\pi) = \int_{\mathbb{P}} f(\bar{\mathbf{x}}; \pi)\mathrm{d}\bar{\mathbf{x}}$$

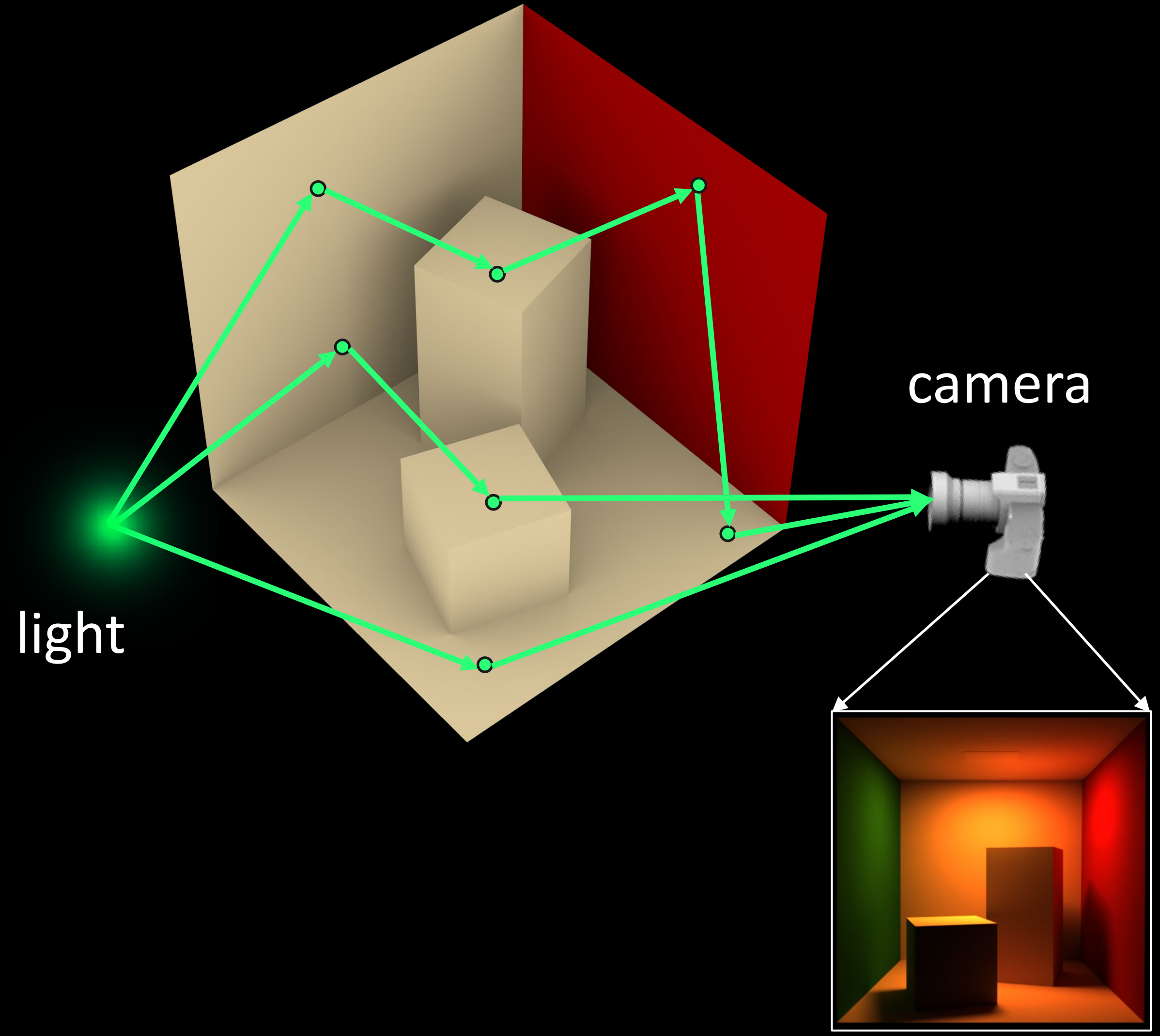
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → Space of valid paths

$f(\bar{\mathbf{x}})$ → Path contribution,
includes geometric terms (visibility, fall-off) &
local terms (BRDF, foreshortening, emission)

Assume $\mathbb{P}$ is independent of $\pi$

# Derivatives of images as path integrals




$$\frac{\partial I}{\partial \pi}(\pi) = ?$$

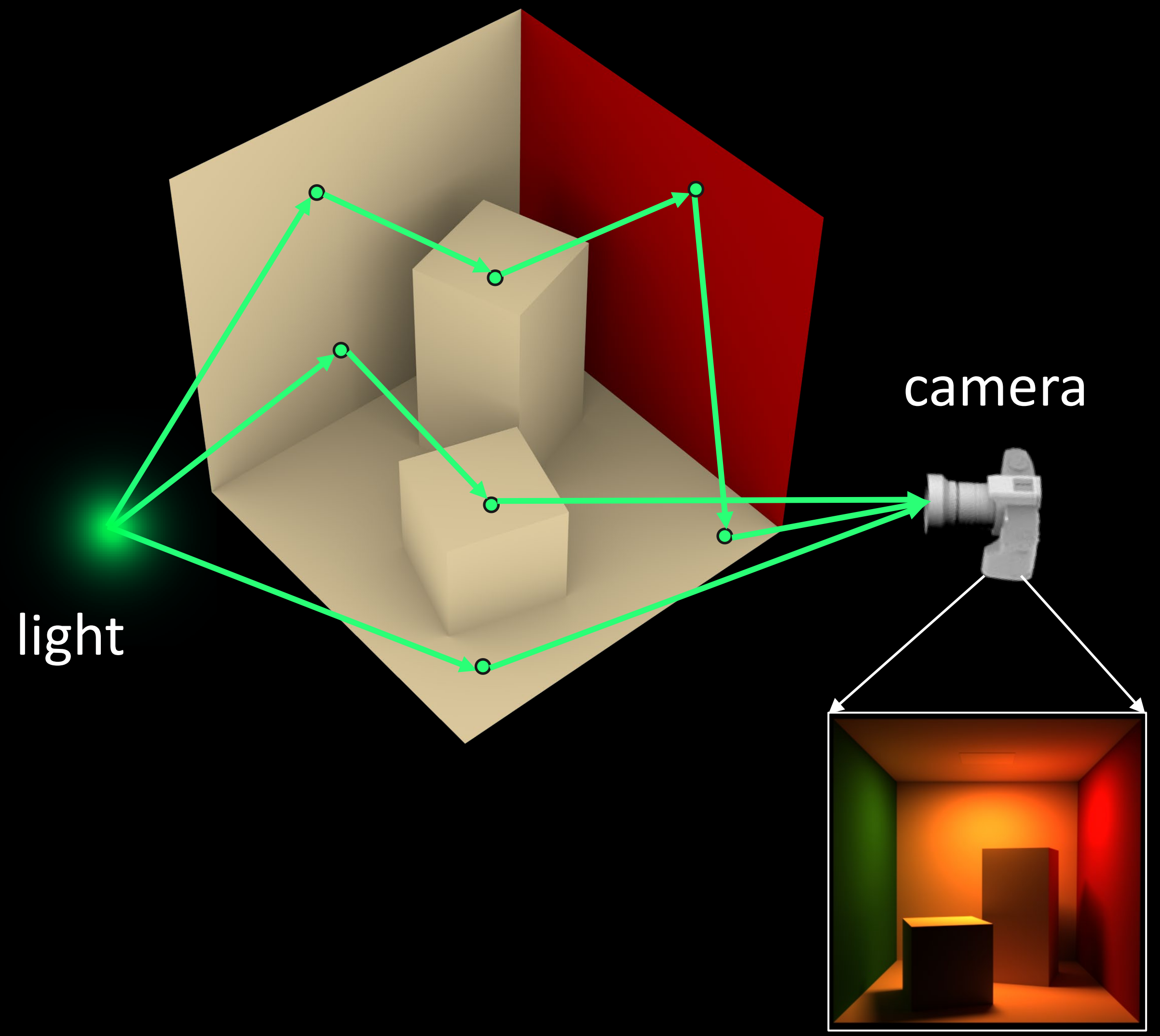
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → Space of valid paths

$f(\bar{\mathbf{x}})$ → Path contribution,
includes geometric terms (visibility, fall-off) &
local terms (BRDF, foreshortening, emission)

Assume $\mathbb{P}$ is independent of $\pi$

# Derivatives of images as path integrals




$$\frac{\partial I}{\partial \pi}(\pi) = \int_{\mathbb{P}} \frac{\partial f}{\partial \pi}(\bar{\mathbf{x}}; \pi) \mathrm{d}\bar{\mathbf{x}}$$

differentiation under the integral sign

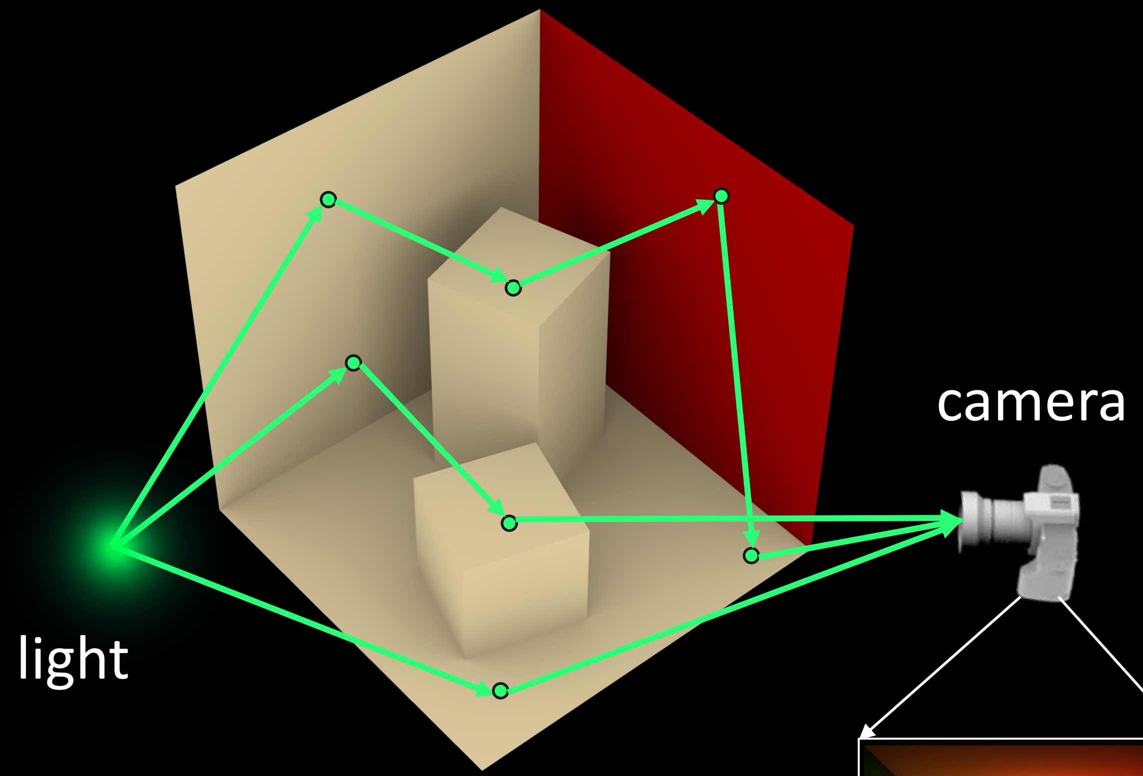
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → Space of valid paths

$f(\bar{\mathbf{x}})$ → Path contribution,
includes geometric terms (visibility, fall-off) &
local terms (BRDF, foreshortening, emission)

Assume $\mathbb{P}$ is independent of $\pi$

# Monte Carlo differentiable rendering (for local parameters)




This term is generally easy to compute during path tracing

$$\frac{\partial I}{\partial \pi}(\pi) \approx \sum_{i=1}^{N} \frac{\boxed{\frac{\partial f}{\partial \pi}(\bar{\mathbf{x}}_i;\pi)}}{p(\bar{\mathbf{x}}_i)}$$

$\bar{x}_i$ → Randomly sampled light paths

$p(\bar{\mathbf{x}}_i)$ → Probability of sampling a path

Sample paths using path tracing etc.

# Score estimator

$$f(\bar{\mathbf{x}};\pi) = \prod_{b=1}^{B} f_s(x_{b-1} \to x_b \to x_{b+1};\pi)\frac{\mathrm{V}(x_{b-1} \leftrightarrow x_b)}{\|x_{b-1} - x_b\|^2}$$

Foreshortening terms are included in the BRDF

$$\frac{\partial f}{\partial \pi}(\bar{\mathbf{x}};\pi) = \boxed{\prod_{b=1}^{B} f_s(x_{b-1} \to x_b \to x_{b+1};\pi)\frac{\mathrm{V}(x_{b-1} \leftrightarrow x_b)}{\|x_{b-1} - x_b\|^2}} \left( \sum_{b=1}^{B} \frac{\frac{\partial f_s}{\partial \pi}(x_{b-1} \to x_b \to x_{b+1};\pi)}{f_s(x_{b-1} \to x_b \to x_{b+1};\pi)} \right)$$

$f(x;\pi)$

At each path vertex:

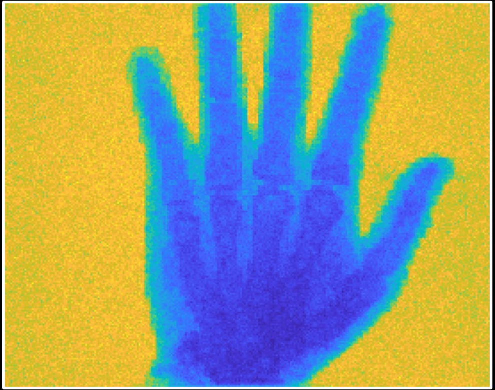
- Update product throughput using $f_s$
- Update score sum using gradient of $f_s$

Multiply the two at end of path

# This is what all these papers do

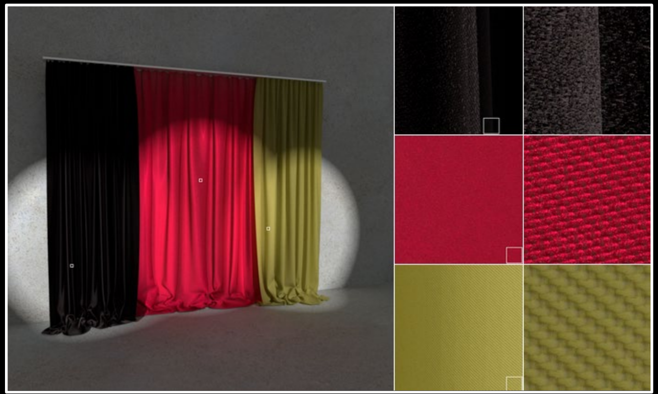
everyday materials

industrial nanodispersions



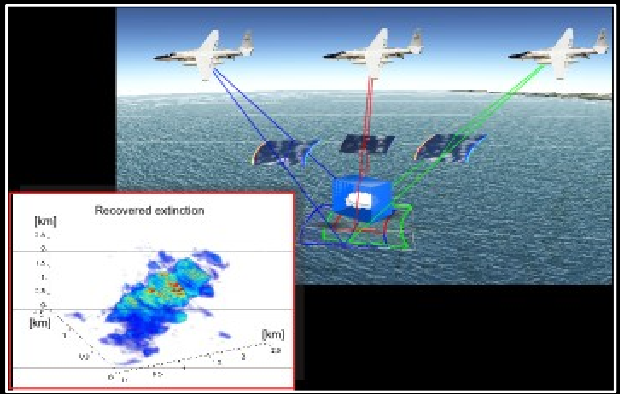
computed tomography [Geva et al. 2018]

optical tomography [Gkioulekas et al. 2016]



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

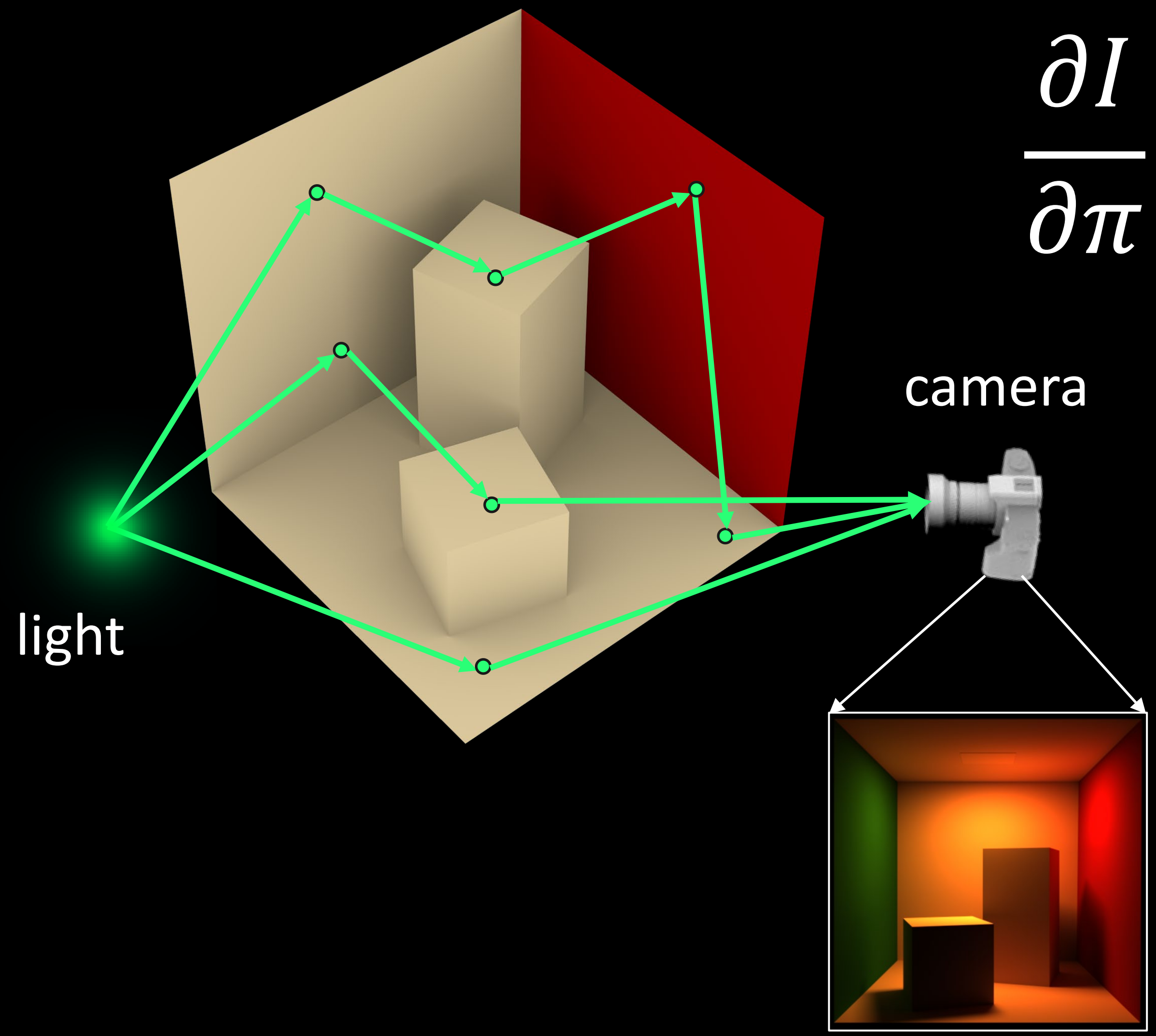
3D printing [Elek et al. 2017, 2019]




clouds [Levis et al. 2015, 2017]

# Even simpler: use autodiff

$$\frac{\partial I}{\partial \pi}(\pi) \approx \sum_{i=1}^{N} \frac{\text{autodiff}(f(\bar{\mathbf{x}}_i; \pi))}{p(\bar{\mathbf{x}}_i)}$$

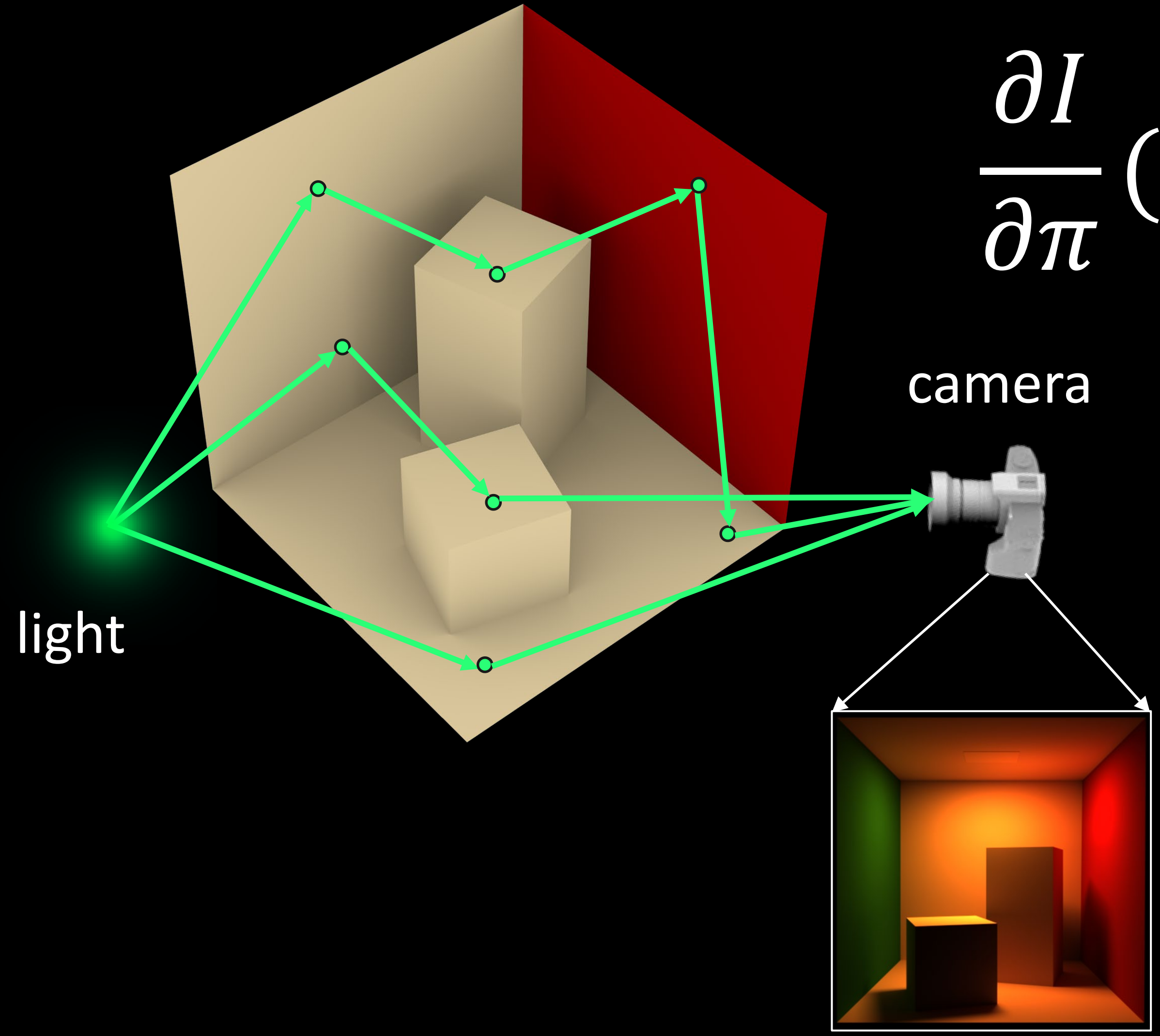



$\bar{\boldsymbol{x}}_i$ → Randomly sampled light paths

$p(\bar{\mathbf{x}}_i)$ → Probability of sampling a path

# Compare with...

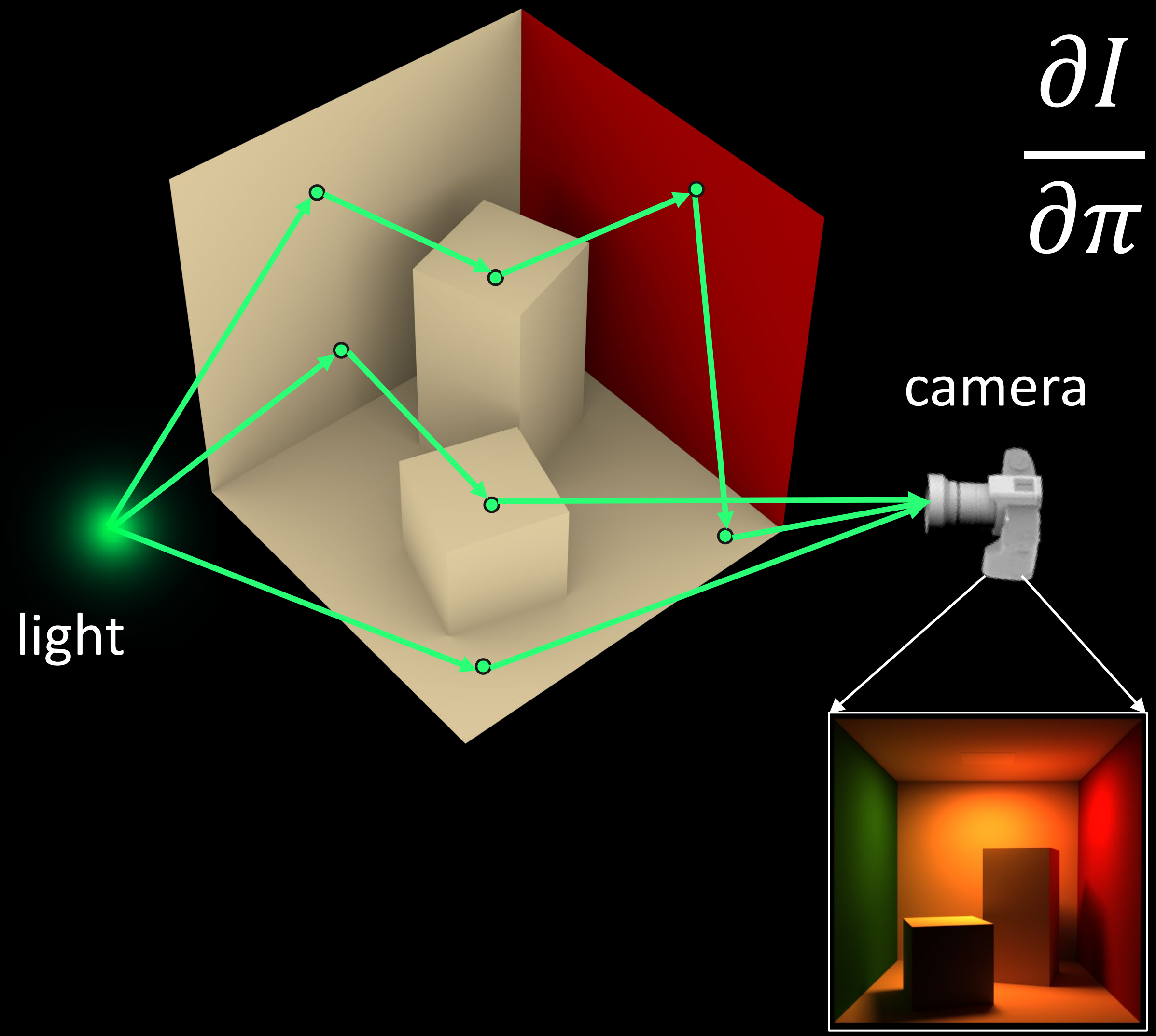
$$\frac{\partial I}{\partial \pi}(\pi) \approx \text{autodiff}\left(\sum_{i=1}^{N} \frac{f(\bar{\mathbf{x}}_i;\pi)}{p(\bar{\mathbf{x}}_i)}\right)$$




$\bar{x}_i$ → Randomly sampled light paths

$p(\bar{\mathbf{x}}_i)$ → Probability of sampling a path

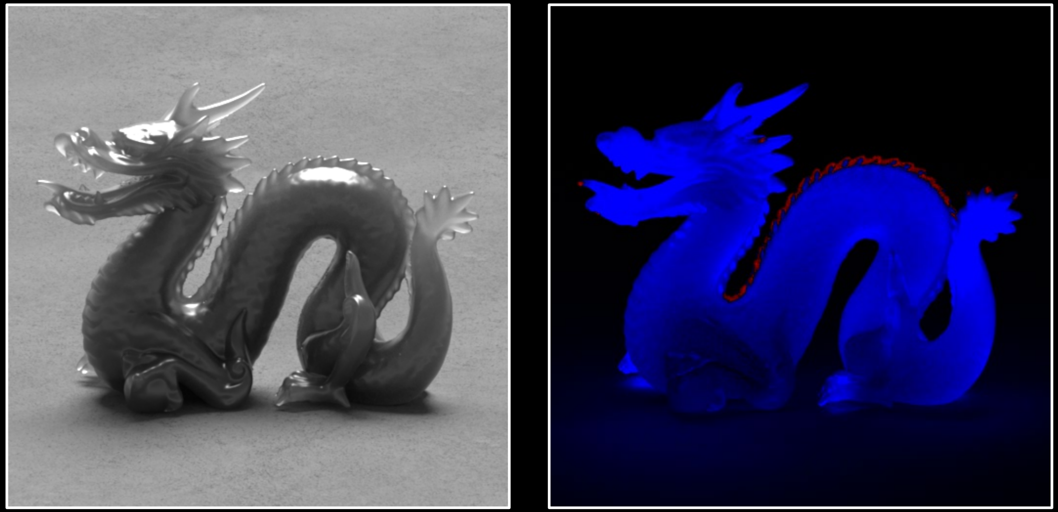
# Even simpler: use autodiff




$$\frac{\partial I}{\partial \pi}(\pi) \approx \sum_{i=1}^{N} \frac{\mathrm{autodiff}(f(\bar{\mathbf{x}}_i; \pi))}{p(\bar{\mathbf{x}}_i)}$$

- Generally lower variance.
- Remember: *Compute an estimate of the derivative, not a derivative of the estimator.*

# Compute an estimate of the derivative



derivative wrt volumetric density

**Inverse Transport Networks**

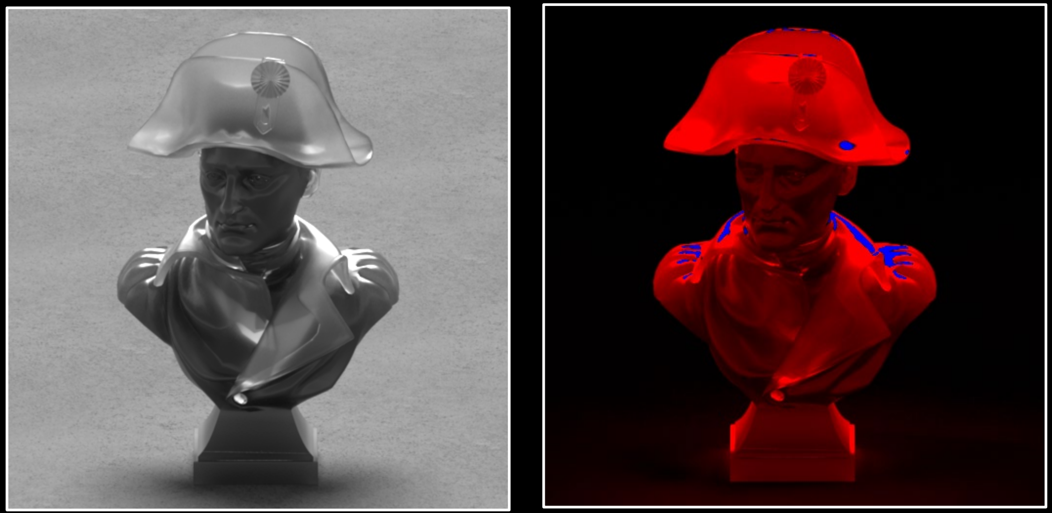

Chengqian Che
Carnegie Mellon University

Fujun Luan
Cornell University

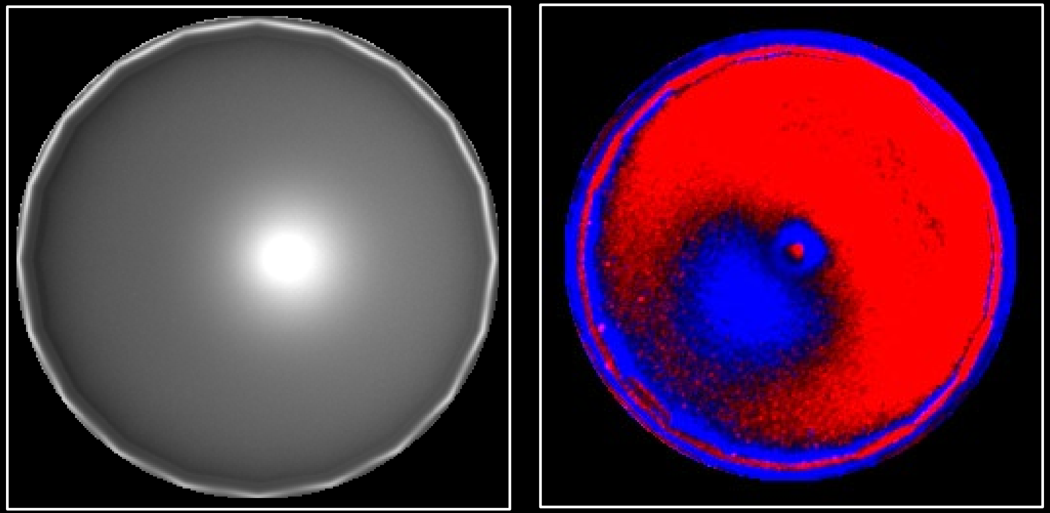
Shuang Zhao
University of California, Irvine

Kavita Bala
Cornell University

Ioannis Gkioulekas
Carnegie Mellon University

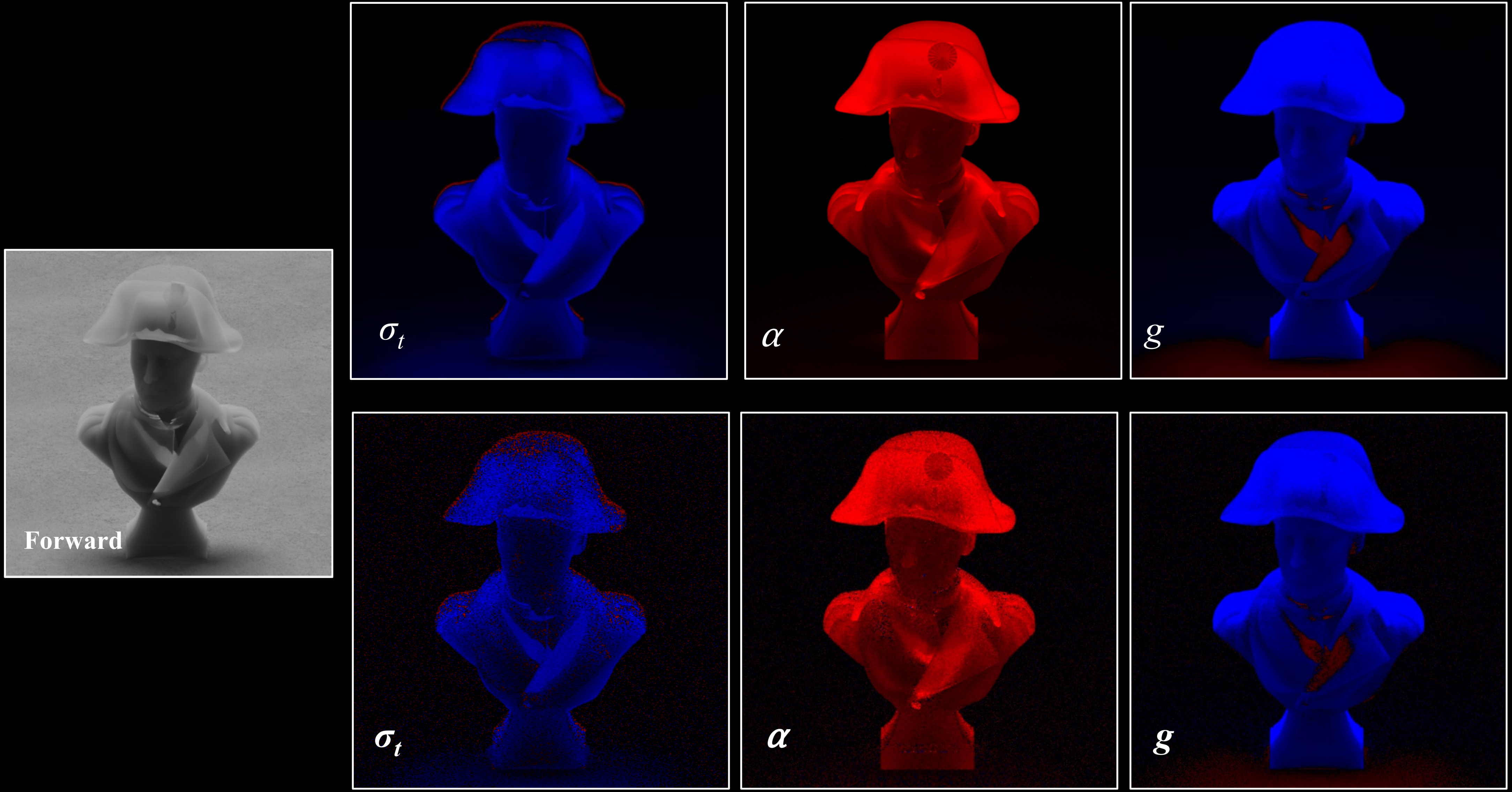



derivative wrt BRDF



derivative wrt normal

# Comparison with finite differences




rendered

finite differences

Note: Finite differences are great for testing the correctness of your gradient code.

# Compute a derivative of the estimate

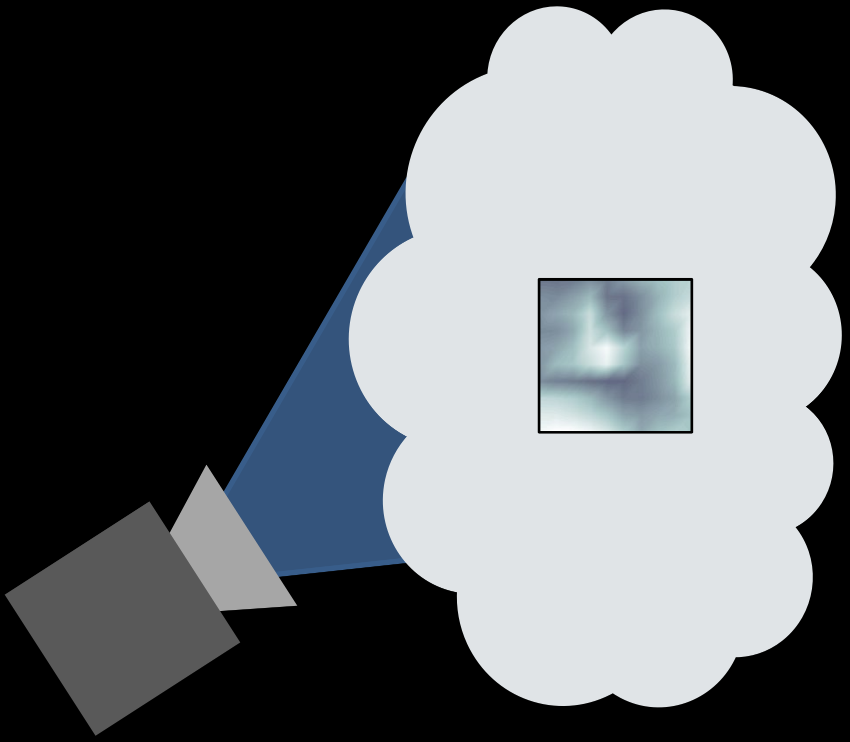
derivative wrt volumetric density

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

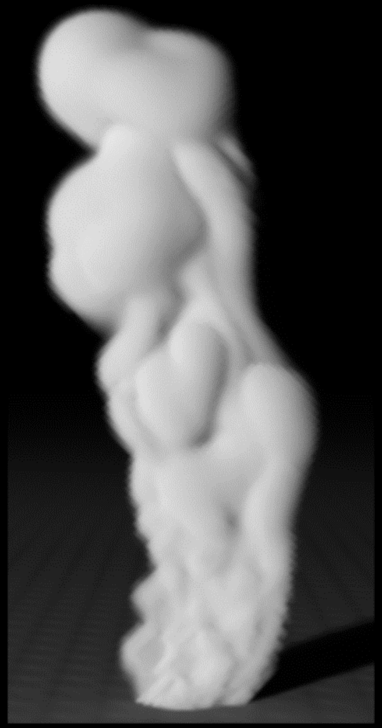
- A lot more general.
- GPU implementation.

# Looking inside scattering objects

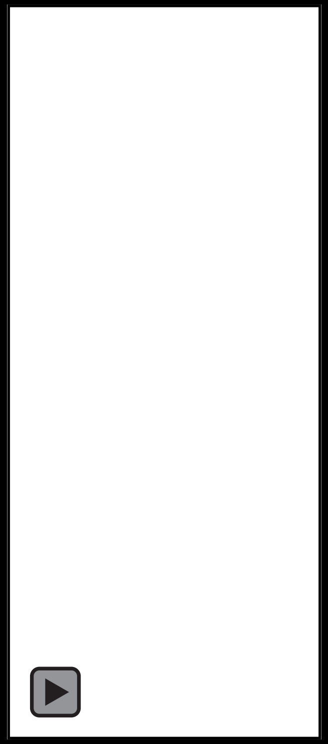


camera

thick smoke cloud



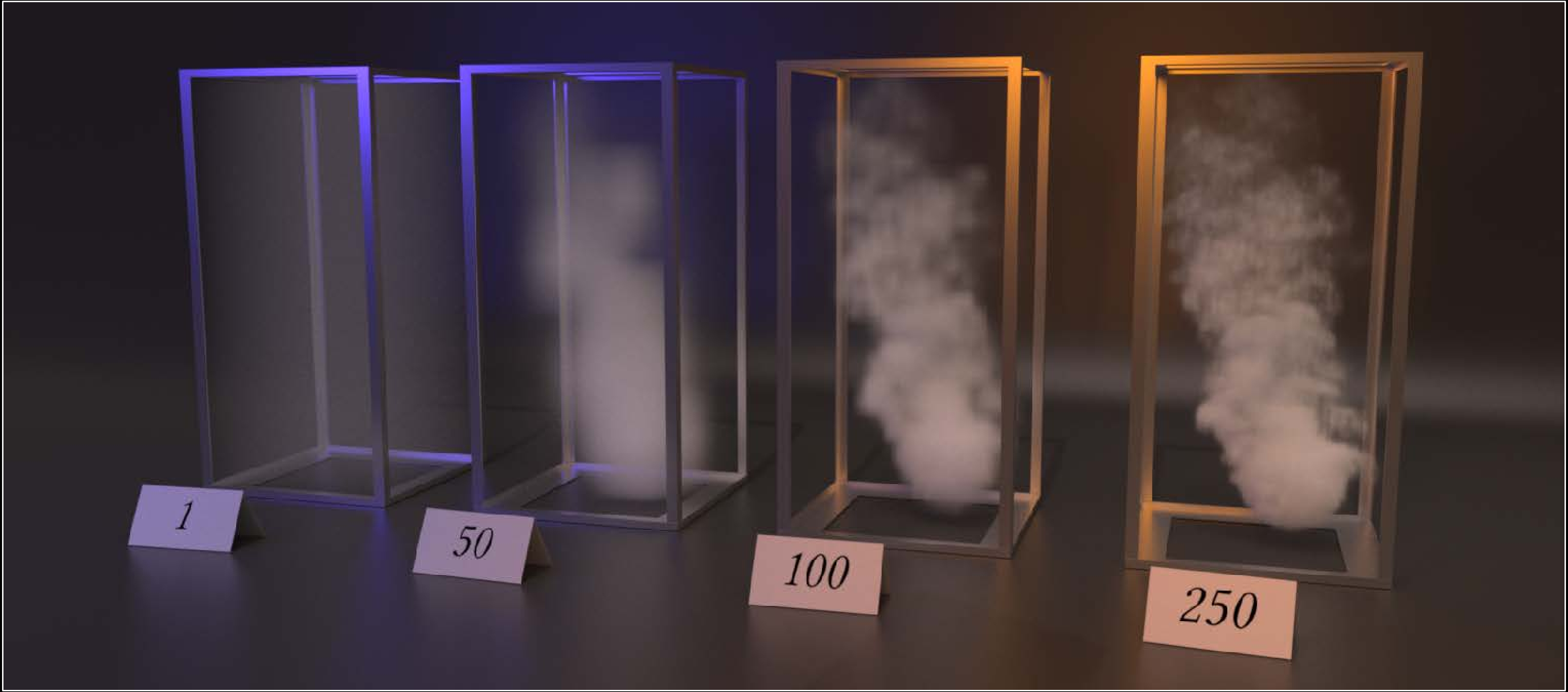
simulated camera measurements



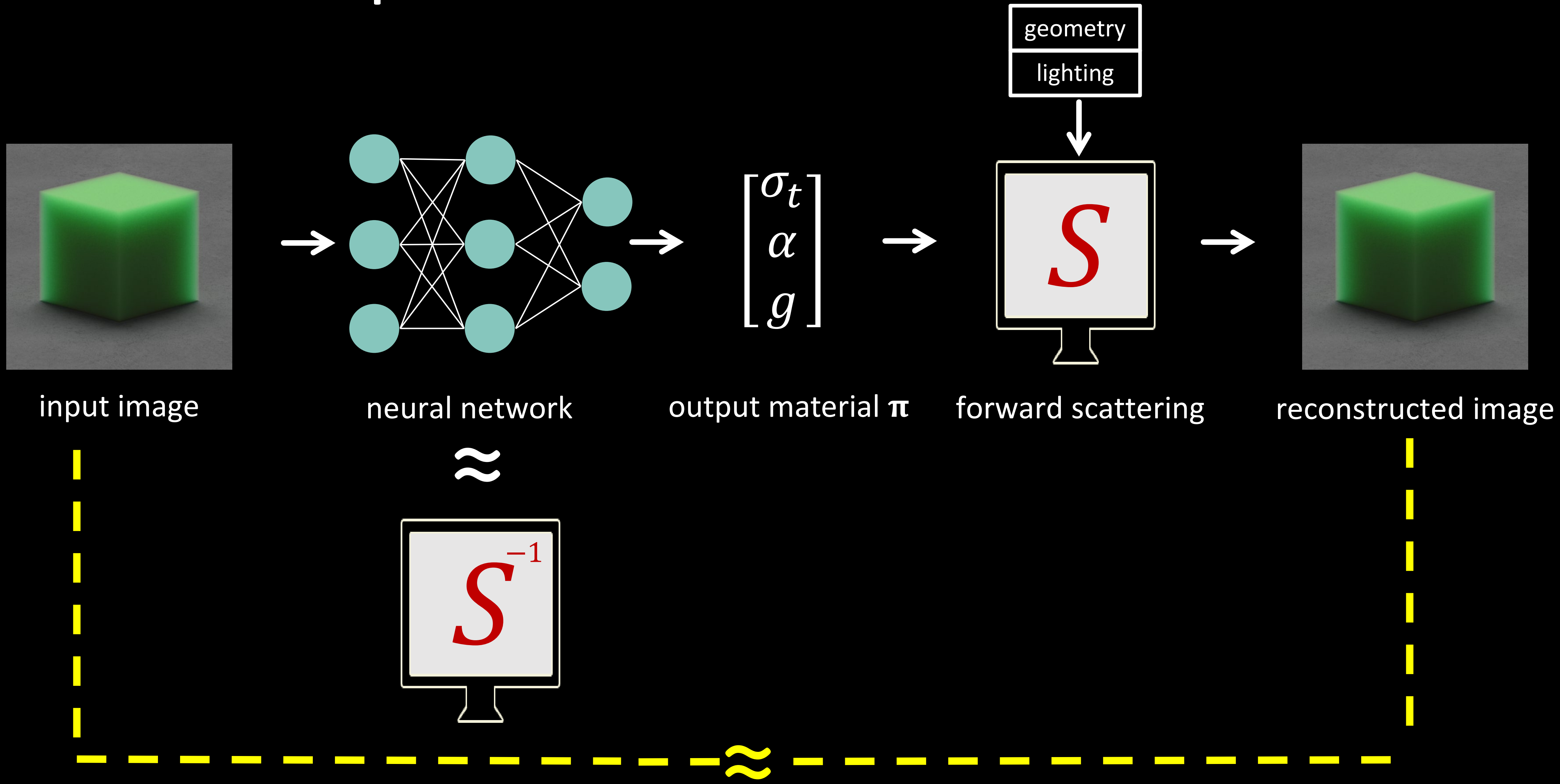
reconstructed cloud volume

slice through the cloud

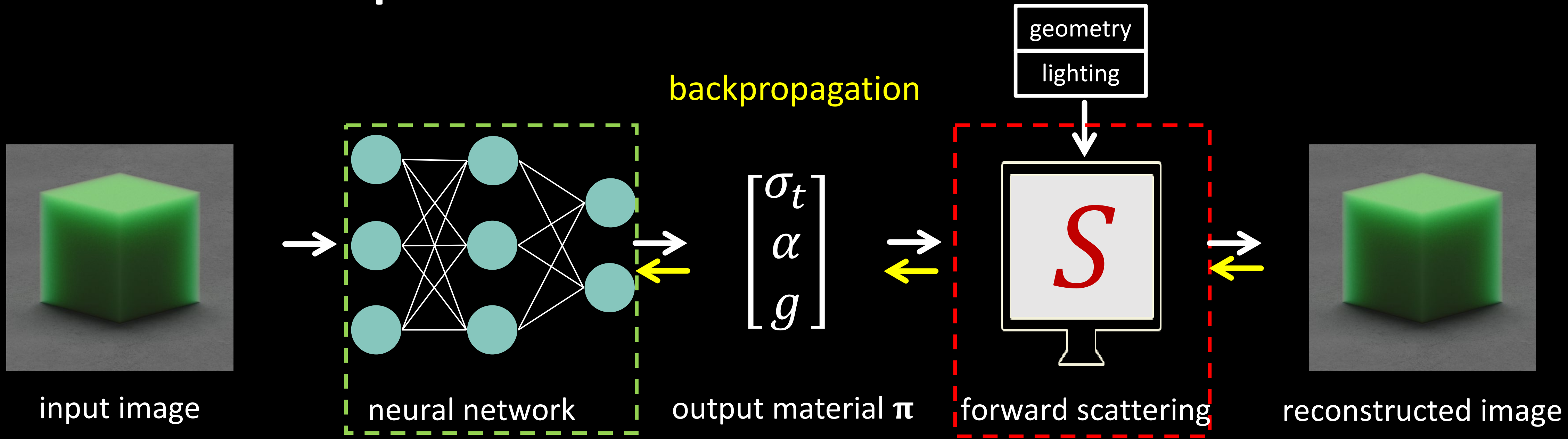
# Looking inside scattering objects



# Inverse transport network



# Inverse transport network




$$\begin{bmatrix}\sigma_t \\ \alpha \\ g\end{bmatrix}$$

## derivatives

$$\boxed{\frac{\partial \text{ neuralNet}}{\partial \textit{ weights}}}$$

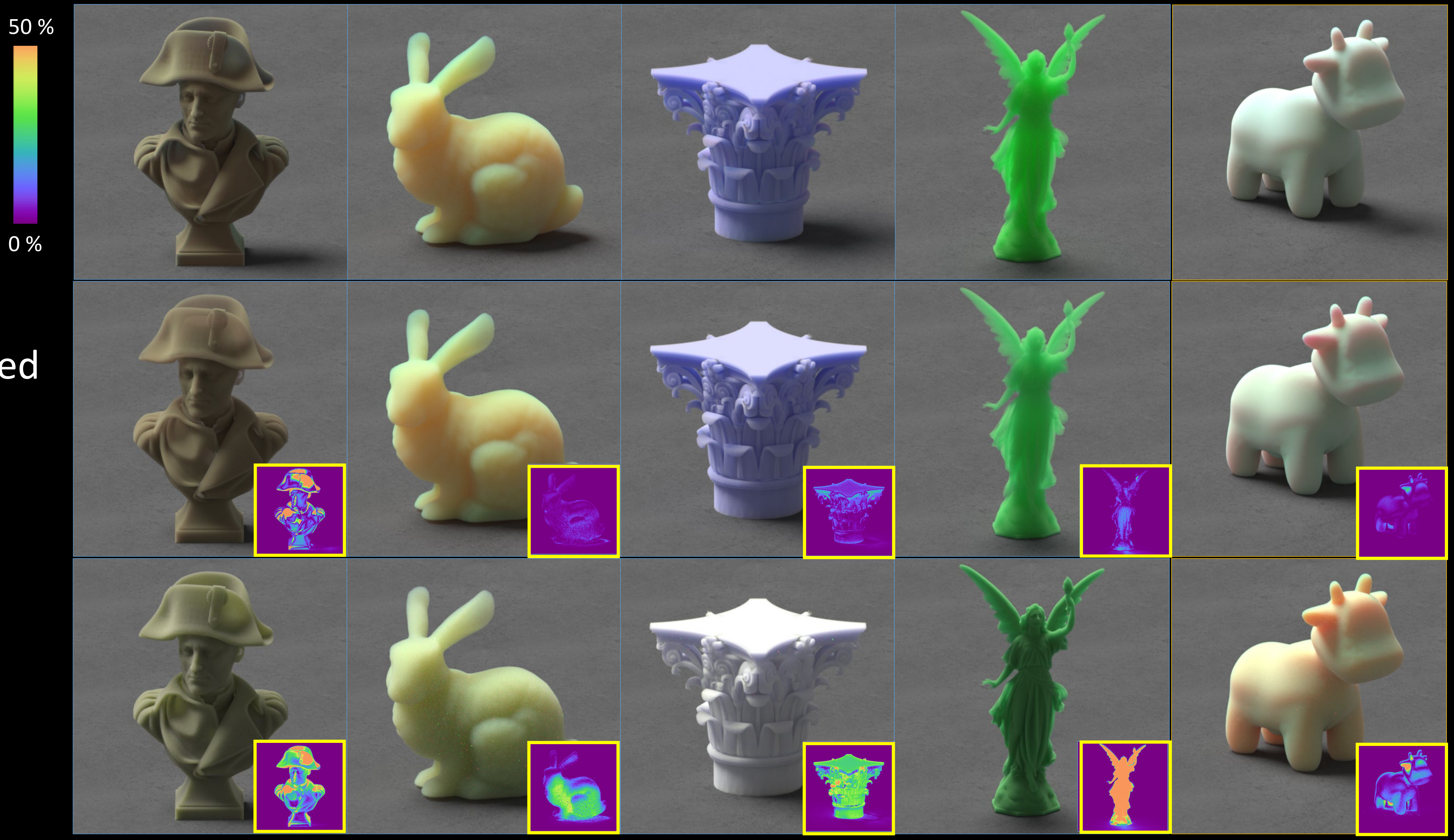
auto-diff (Torch, TensorFlow etc.)

parameter loss

$$\boxed{\frac{\partial \text{ forwardScattering}(\boldsymbol{\pi})}{\partial\, \boldsymbol{\pi}}}$$

volumetric differentiable renderer

# Examples




groundtruth

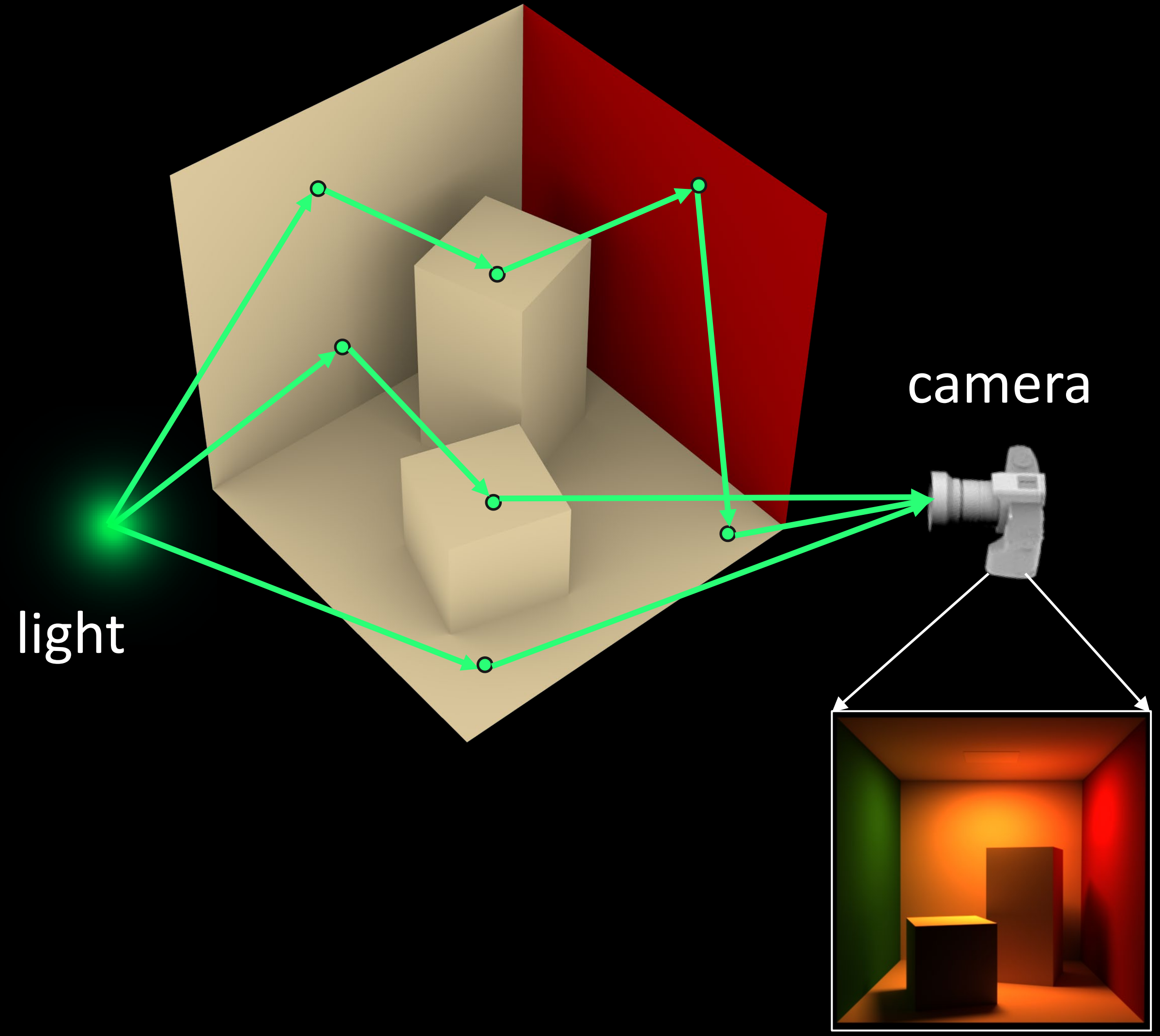
supervised and unsupervised

parameter loss: 0.60X
appearance loss: 0.40X
novel appearance loss: 0.42X

supervised only

parameter loss: 1X
appearance loss: 1X
novel appearance loss: 1X

# Derivatives of images as path integrals




$$\frac{\partial I}{\partial \pi}(\pi) = \int_{\mathbb{P}} \frac{\partial f}{\partial \pi}(\bar{\mathbf{x}}; \pi)\mathrm{d}\bar{\mathbf{x}}$$

differentiation under the integral sign

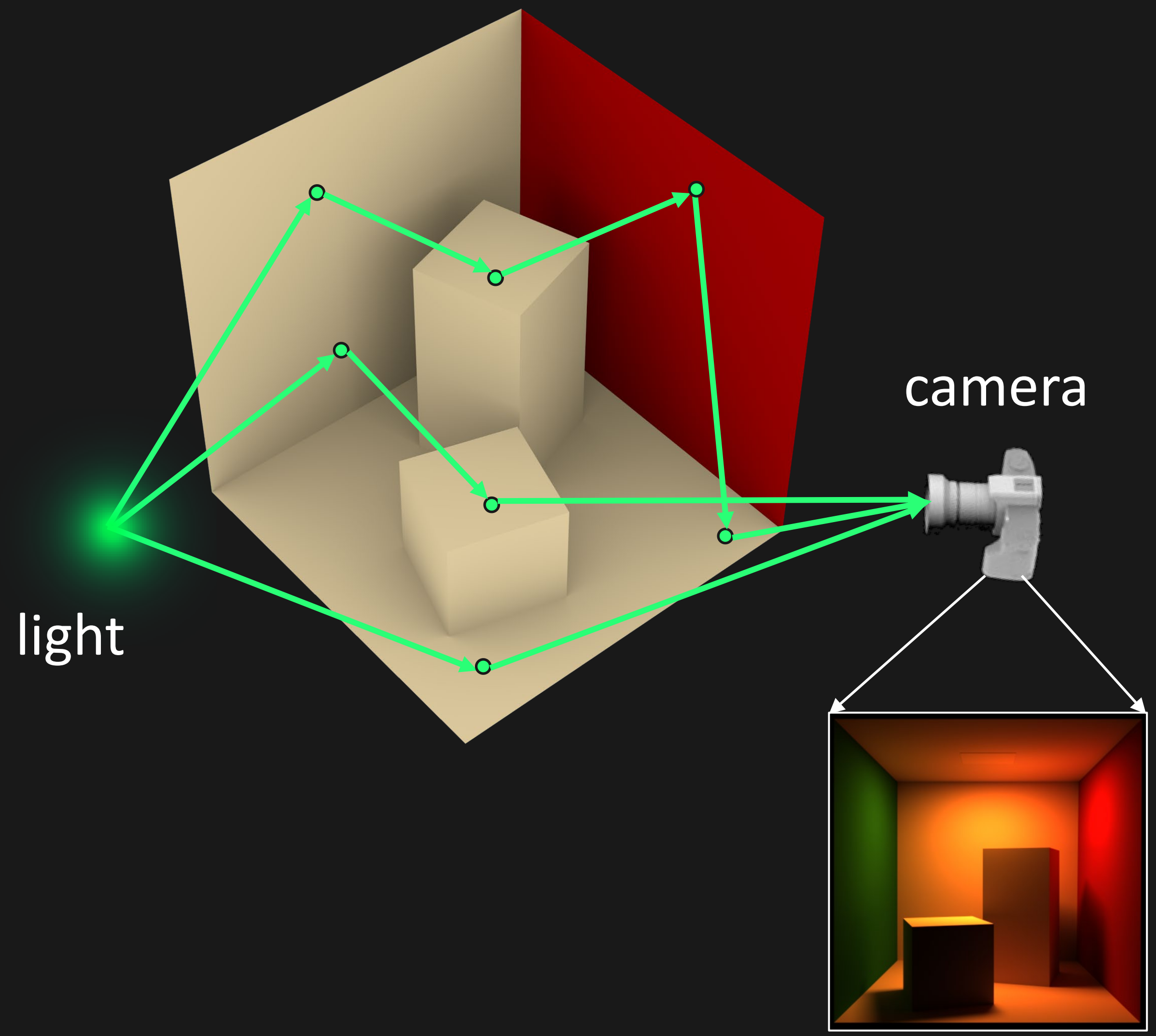
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → Space of valid paths

$f(\bar{\mathbf{x}})$ → Path contribution,
includes geometric terms (visibility, fall-off) &
local terms (BRDF, foreshortening, emission)

Assume $\mathbb{P}$ is independent of $\pi$

# Derivatives of images as path integrals




$$\frac{\partial I}{\partial \pi}(\pi) = \int_{\mathbb{P}} \frac{\partial f}{\partial \pi}(\bar{\mathbf{x}}; \pi)\mathrm{d}\bar{\mathbf{x}}$$

differentiation under the integral sign

What about parameters π that change $\mathbb{P}$?

- Location, pose, and shape of light, camera, and scene objects.

# Differentiable rendering for global geometry

# We'll work with the rendering equation

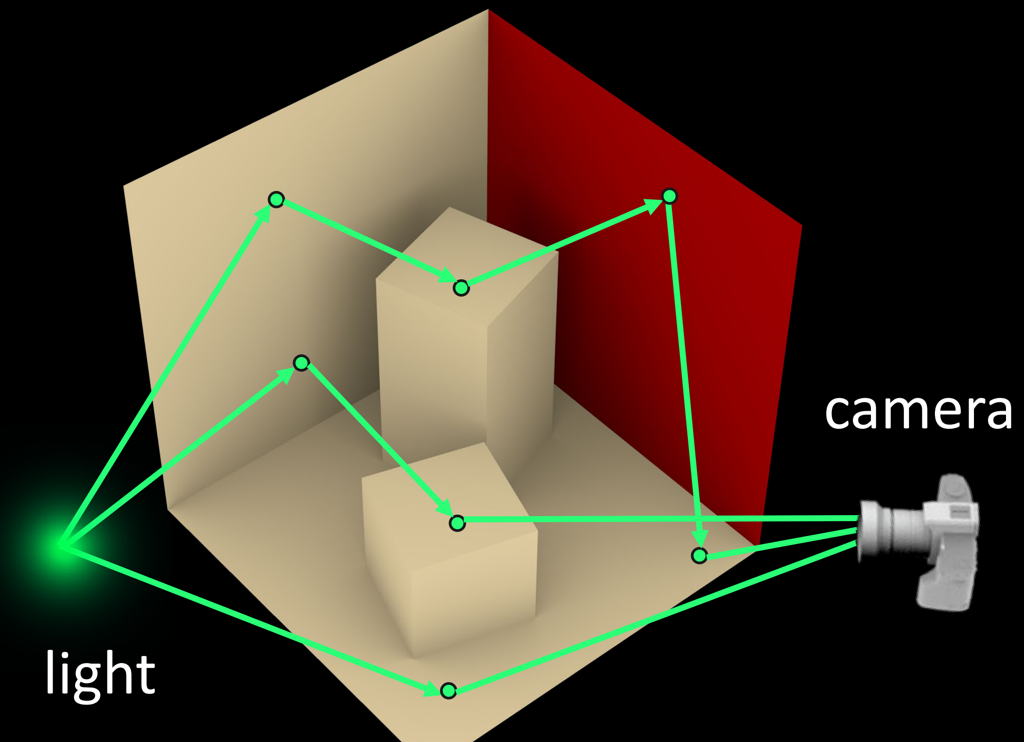
$$L(x, \omega; \pi) = \int_{G(\pi)} L(x' \to x; \pi) f(x' \to x, \omega; \pi) V(x' \leftrightarrow x; \pi) \mathrm{d}A(x')$$

$L$ → Radiance at a point and direction

$G$ → All surfaces in the scene

$f$ → Reflection, foreshortening, and fall-off

V → Visibility



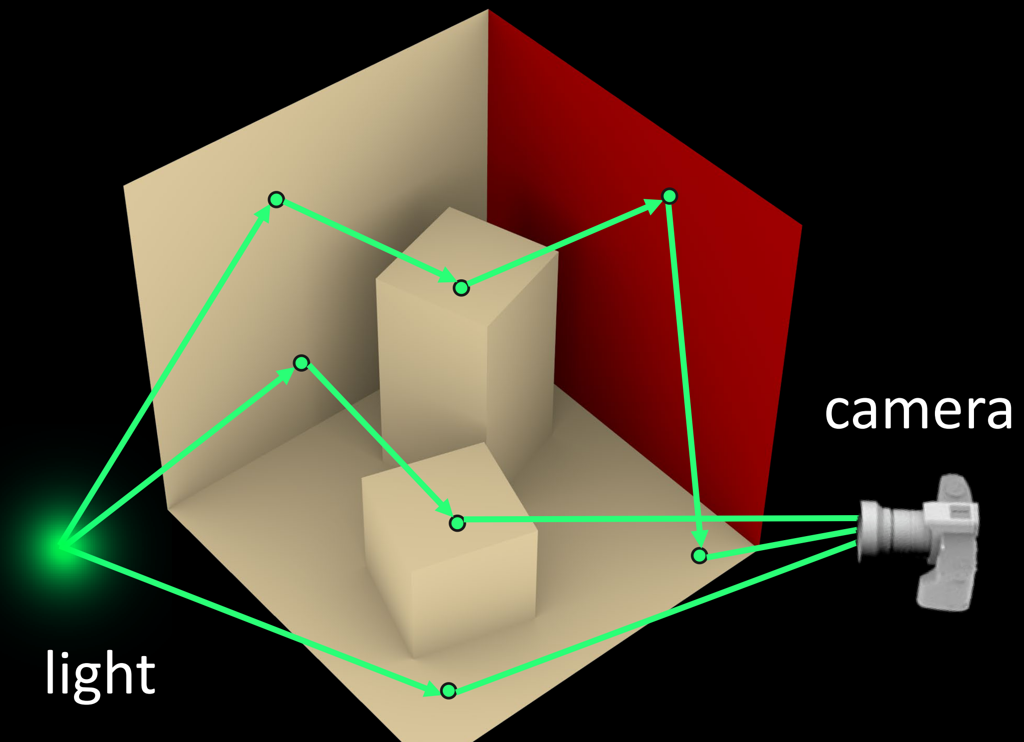
# Let's slightly rewrite the rendering equation

$$L(x, \omega; \pi) = \int_{V(x,\pi)} L(x' \to x; \pi) f(x' \to x, \omega; \pi) \mathrm{d}A(x')$$

$L$ → Radiance at a point and direction

$V$ → All <u>visible</u> surfaces in the scene

$f$ → Reflection, foreshortening, and fall-off



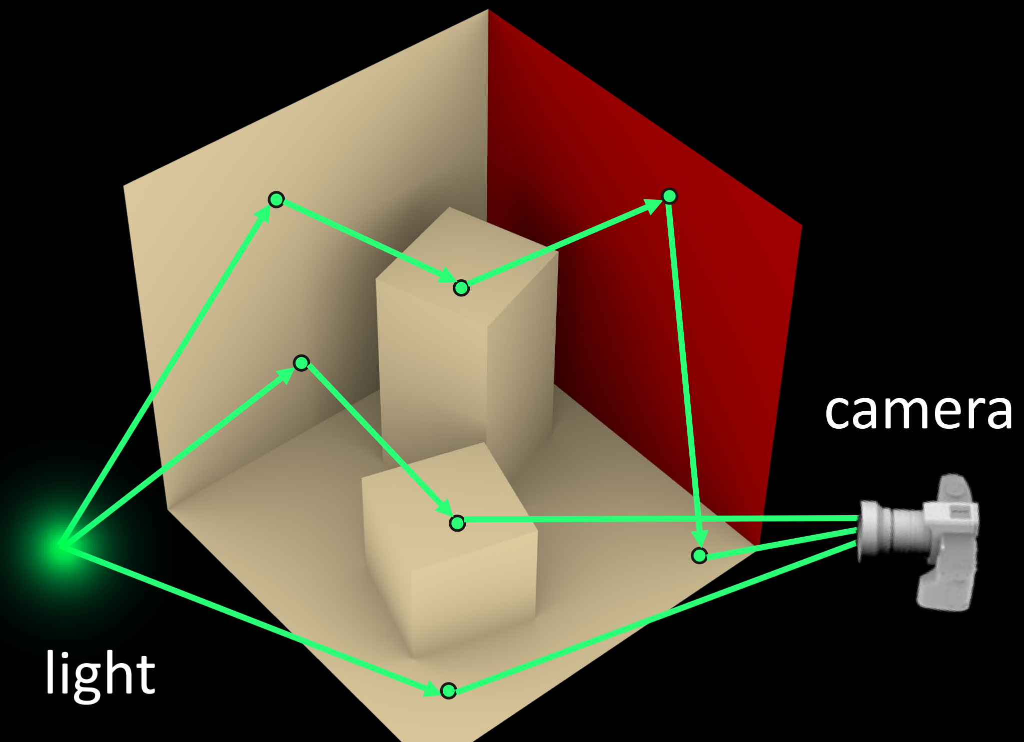
# Let's differentiate it

$$\frac{\partial}{\partial \pi} L(x, \omega; \pi) = \frac{\partial}{\partial \pi} \int_{V(x,\pi)} L(x' \to x; \pi) f(x' \to x, \omega; \pi) \mathrm{d}A(x')$$

$L$ → Radiance at a point and direction

$V$ → All visible surfaces in the scene

$f$ → Reflection, foreshortening, and fall-off




Can we just move the integral inside?

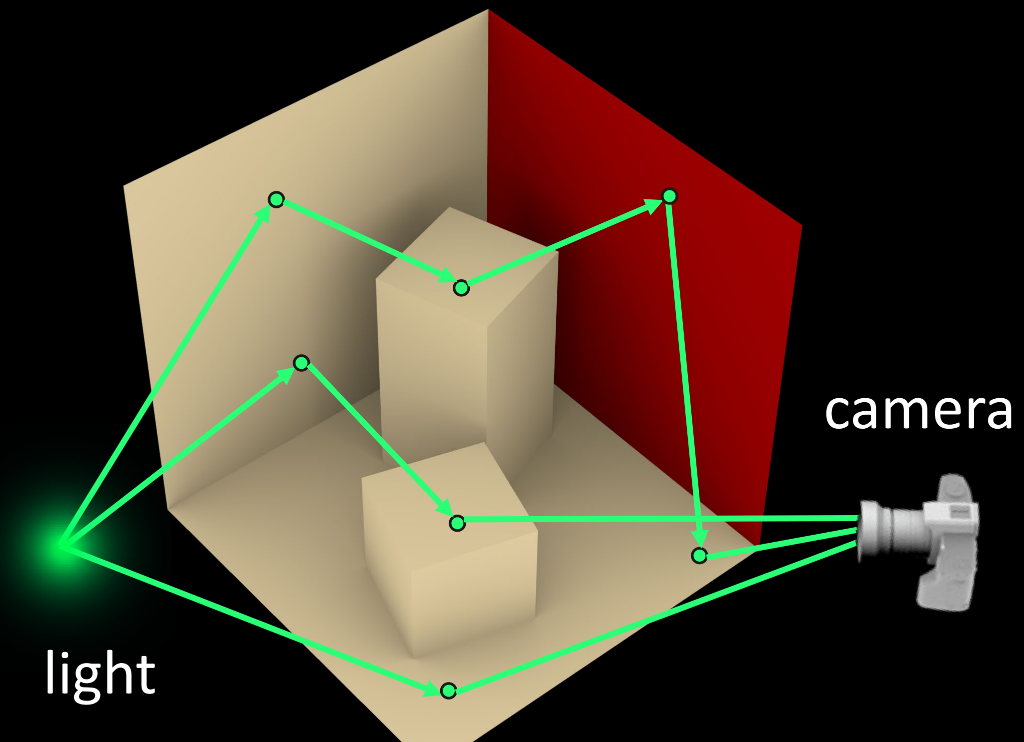
# Let’s differentiate it

$$\frac{\partial}{\partial \pi} L(x, \omega; \pi) = \frac{\partial}{\partial \pi} \int_{V(x,\pi)} L(x' \to x; \pi) f(x' \to x, \omega; \pi) \mathrm{d}A(x')$$

$L$ → Radiance at a point and direction

$V$ → All visible surfaces in the scene

$f$ → Reflection, foreshortening, and fall-off




Can we just move the integral inside?

- No. What can we do?

# Basic differentiation rules

$$\frac{\partial}{\partial \pi}\int_a^b f(x;\pi)dx = \int_a^b \frac{\partial}{\partial \pi} f(x;\pi)dx$$

differentiation under the integral sign

$$\begin{aligned}\frac{\partial}{\partial \pi}\int_{a(\pi)}^{b(\pi)} f(x;\pi)dx &= \int_{a(\pi)}^{b(\pi)} \frac{\partial}{\partial \pi} f(x;\pi)dx \\ &\quad + f(b(\pi);\pi)\frac{\partial b(\pi)}{\partial \pi} - f(\alpha(\pi);\pi)\frac{\partial \alpha(\pi)}{\partial \pi}\end{aligned}$$

Leibniz integral rule

We need a version of this for surface integrals

# Reynolds transport theorem for surfaces

$$\frac{\partial}{\partial \pi}\int_{S(\pi)} f(x;\pi)\mathrm{d}A(x) = \underbrace{\int_{S(\pi)} \dot{f}\,\mathrm{d}A(x)}_{\text{surface integral}} + \underbrace{\int_{\partial S(\pi)} f\left\langle t, \frac{\partial x}{\partial \pi}\right\rangle \mathrm{d}l(x)}_{\text{line integral on } \textit{boundary} \text{ and } \textit{discontinuities}}$$

# REYNOLDS TRANSPORT THEOREM

$$\frac{\mathrm{d}}{\mathrm{d}\pi}\int_{\Omega} f\,\mathrm{d}A = \int_{\Omega} \frac{\mathrm{d}f}{\mathrm{d}\pi}\,\mathrm{d}A + \int_{\partial\Omega} g\,\mathrm{d}l$$

**Reynolds transport theorem**
Generalization of Leibniz's rule

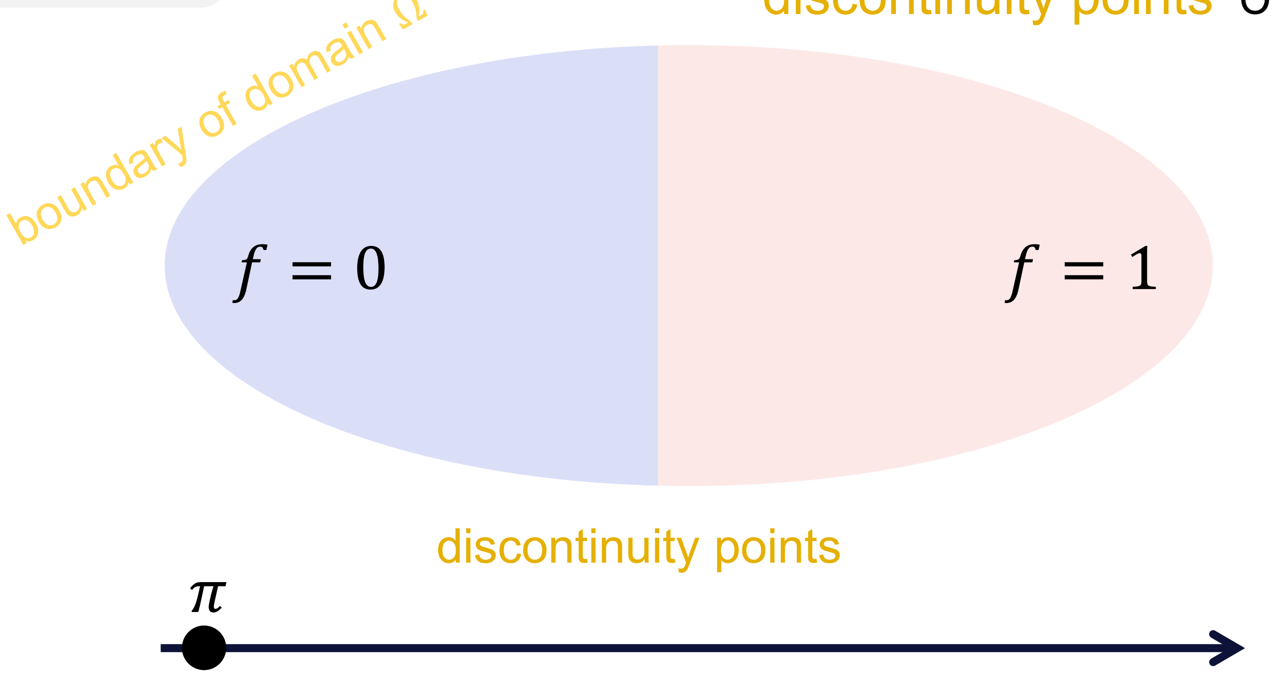
Interior integral

Boundary domain

boundary integral

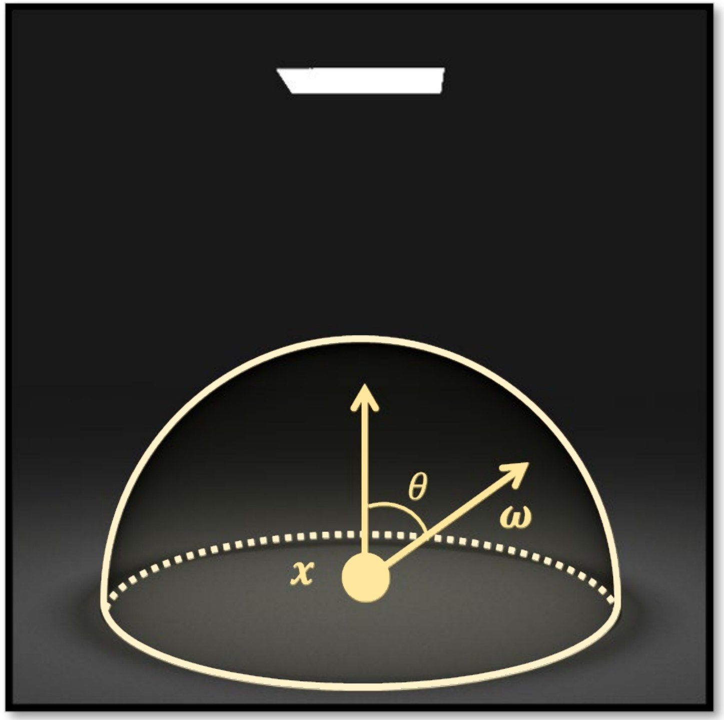
||

discontinuity points ∪ boundary of domain Ω



# REYNOLDS TRANSPORT THEOREM

$\pi$: size of the emitter




Irradiance at $\boldsymbol{x}$

$$E = \int_{\mathbb{H}^2} L_i(\boldsymbol{\omega}) \cos\theta \, \mathrm{d}\sigma(\boldsymbol{\omega})$$

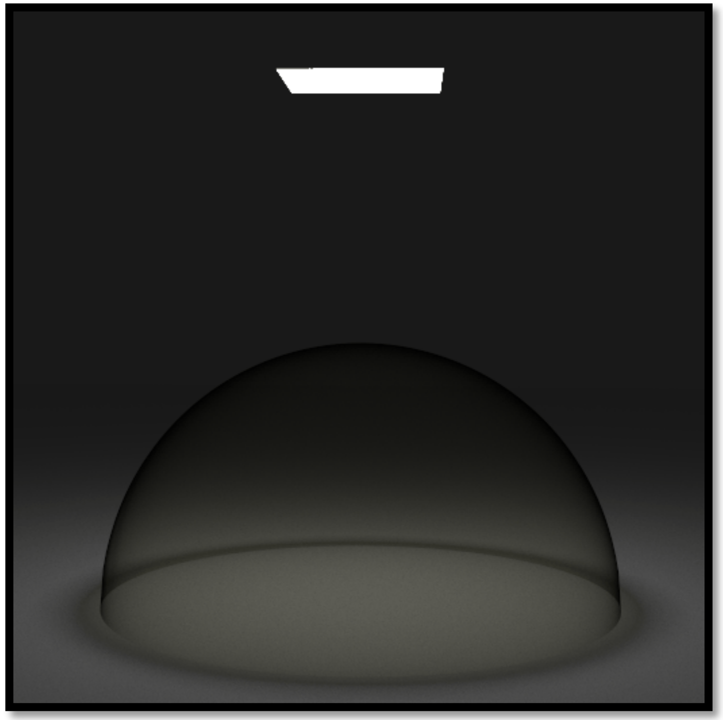
Unit hemisphere

Differential irradiance at $\boldsymbol{x}$

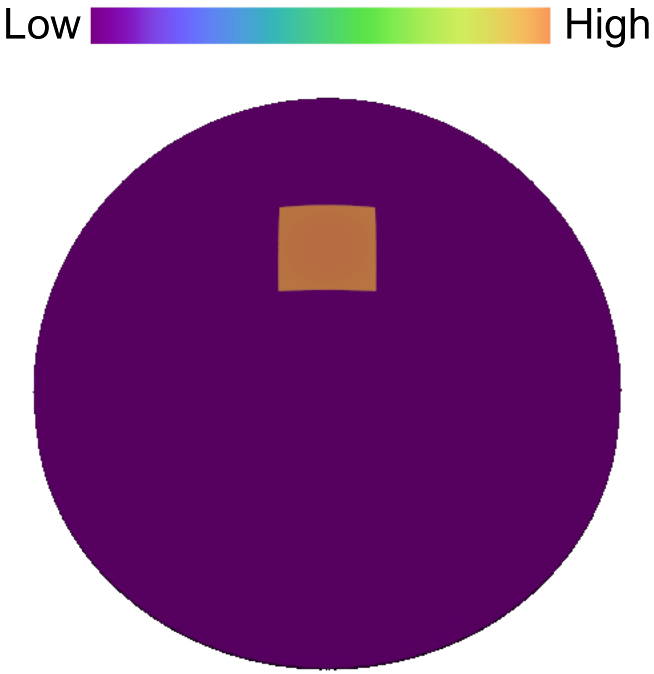
$$\frac{\mathrm{d}E}{\mathrm{d}\pi} = \frac{\mathrm{d}}{\mathrm{d}\pi} \int_{\mathbb{H}^2} L_i(\boldsymbol{\omega}) \cos\theta \, \mathrm{d}\sigma(\boldsymbol{\omega})$$

# REYNOLDS TRANSPORT THEOREM

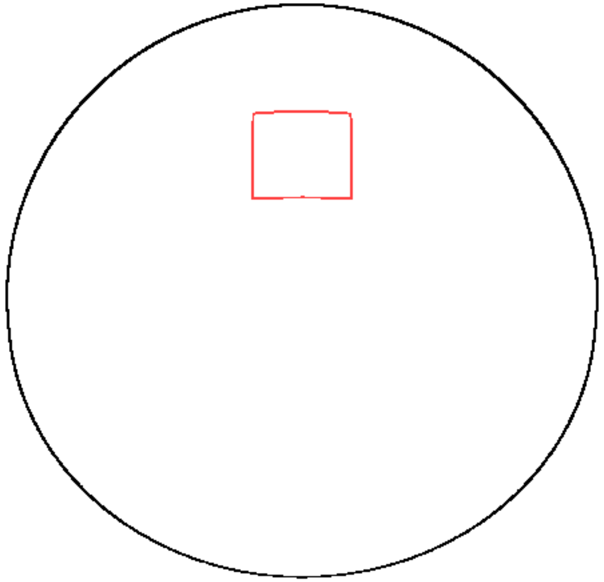
$\pi$: size of the emitter



$$E = \int_{\mathbb{H}^2} L_i(\boldsymbol{\omega}) \cos\theta \,\mathrm{d}\sigma(\boldsymbol{\omega})$$




The integrand

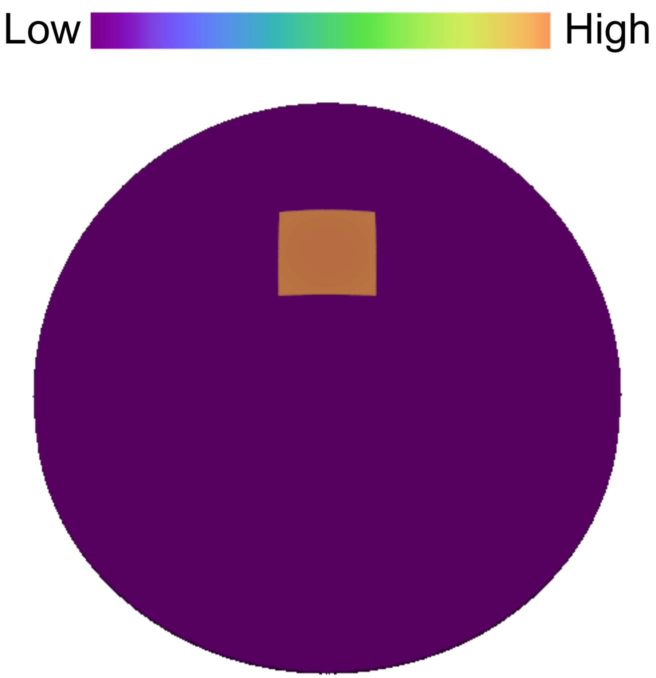


Discontinuous points
($\pi$-dependent)

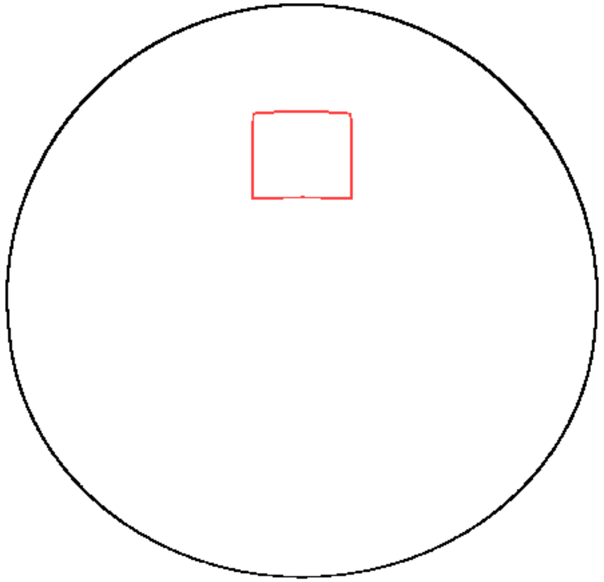
# REYNOLDS TRANSPORT THEOREM

$$E = \int_{\mathbb{H}^2} \overbrace{L_i(\boldsymbol{\omega}) \cos\theta \, \mathrm{d}\sigma(\boldsymbol{\omega})}^{f}$$

$$\frac{\mathrm{d}E}{\mathrm{d}\pi} = \int_{\mathbb{H}^2} \underbrace{\frac{\mathrm{d}f}{\mathrm{d}\pi}}_{=0} \mathrm{d}\sigma + \underbrace{\int_{\partial\mathbb{H}^2} g \, \mathrm{d}l}_{\neq 0}$$




The integrand



Discontinuous points
($\pi$-dependent)

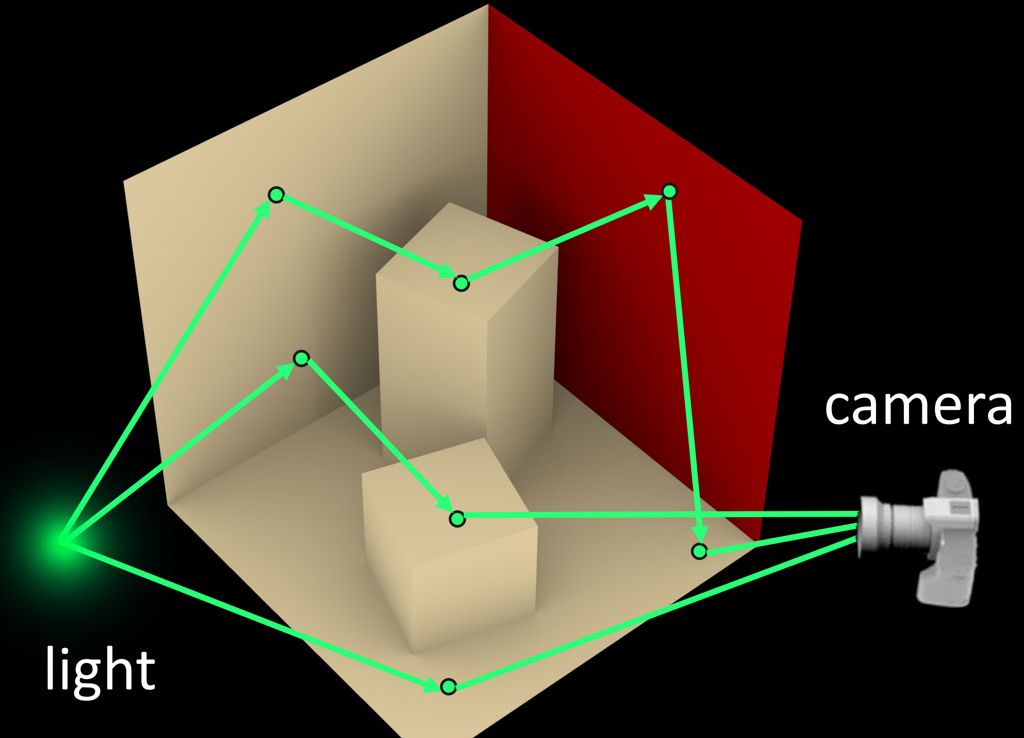
# Let's differentiate the rendering equation

$$\frac{\partial}{\partial \pi} L(x, \omega; \pi) = \frac{\partial}{\partial \pi} \int_{V(x,\pi)} L(x' \to x; \pi) f(x' \to x, \omega; \pi) \mathrm{d}A(x')$$

$L$ → Radiance at a point and direction

$V$ → All visible surfaces in the scene

$f$ → Reflection, foreshortening, and fall-off




What are the "boundary" and discontinuities of $V$?

# Boundaries

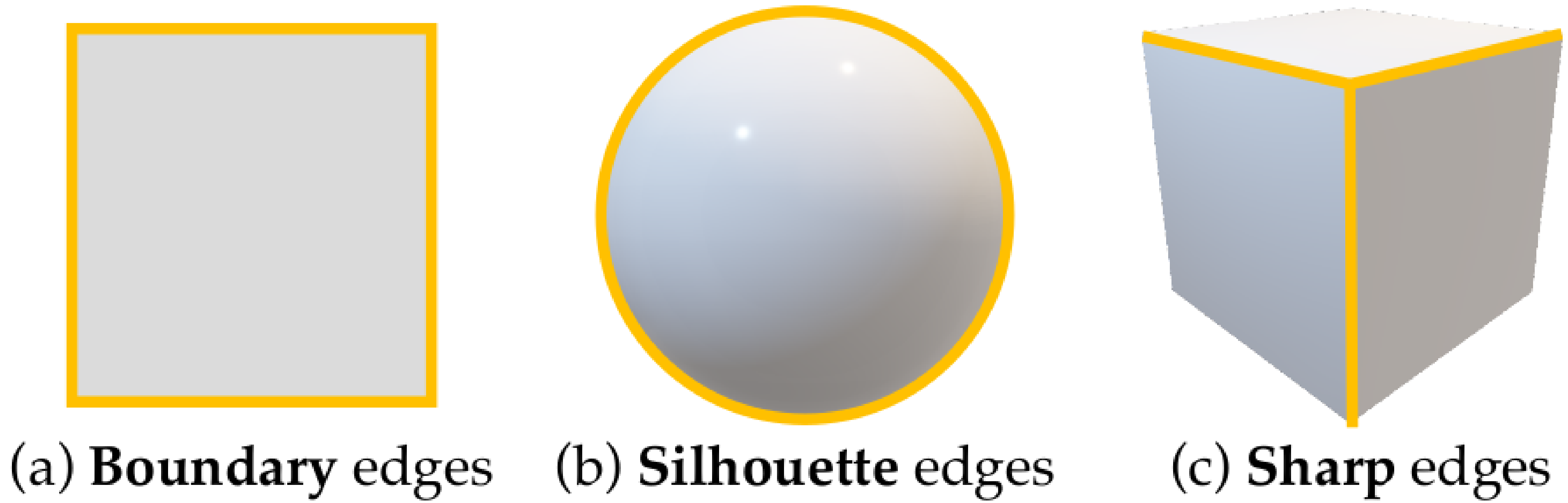
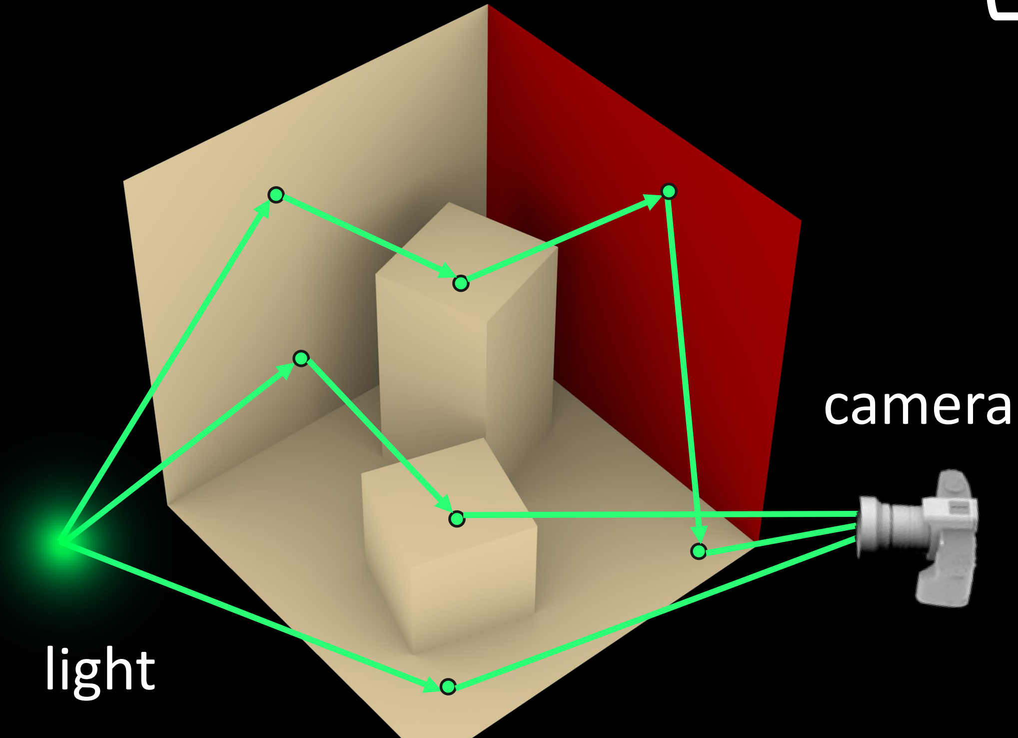



Fig. 5. Three types of edges (drawn in yellow) that can cause geometric discontinuities: (a) boundary, (b) silhouette, and (c) sharp.

# Let's differentiate it

$$\frac{\partial}{\partial \pi} L(x, \omega; \pi) =$$

$$\underbrace{\int_{V(x,\pi)} \frac{\partial}{\partial \pi} L \mathrm{d}A(x)}_{\text{recursively estimate derivative of L at some visible point}} + \underbrace{\int_{\partial V(x,\pi)} H(L) \mathrm{d}\sigma(x)}_{\text{recursively estimate radiance L at some boundary point}}$$




Not terribly good, as we ray trace, we need to:
- recompute silhouette at each vertex
- branch twice

# Global geometry differentiation

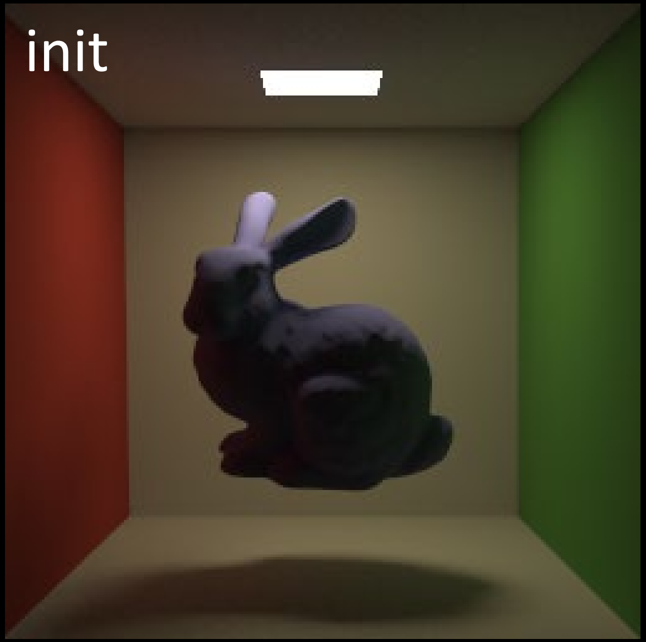
Differentiable Monte Carlo Ray Tracing through Edge Sampling

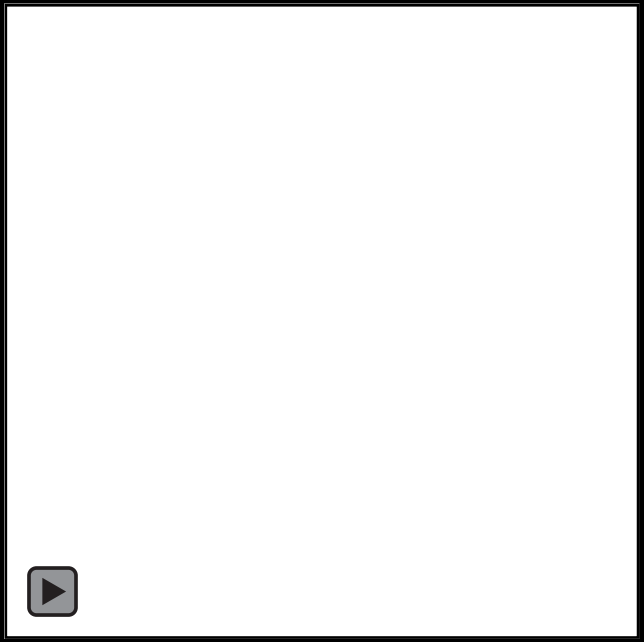
TZU-MAO LI, MIT CSAIL
MIIKA AITTALA, MIT CSAIL
FRÉDO DURAND, MIT CSAIL
JAAKKO LEHTINEN, Aalto University & NVIDIA

**Beyond Volumetric Albedo — A Surface Optimization Framework for Non-Line-of-Sight Imaging**

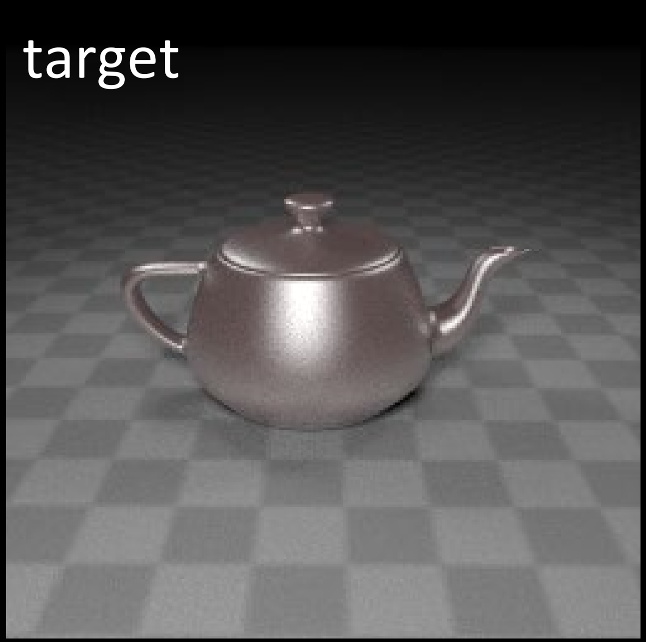
Chia-Yin Tsai, Aswin C. Sankaranarayanan, and Ioannis Gkioulekas
Carnegie Mellon University

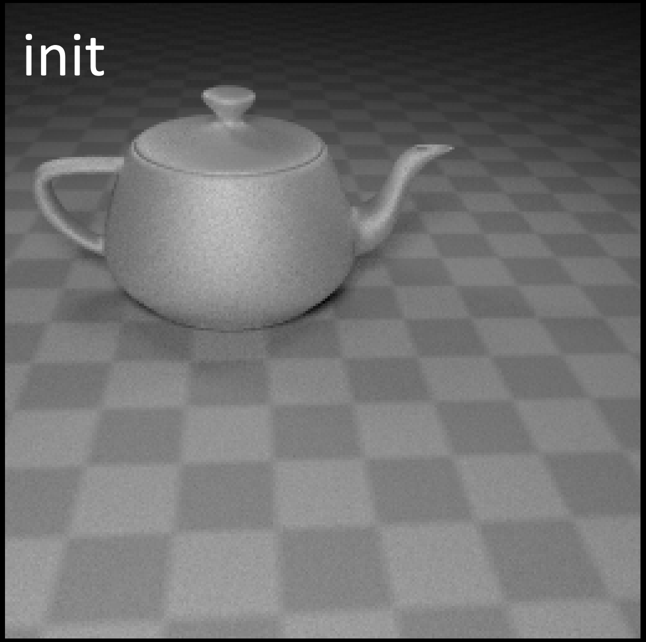
# Global geometry differentiation

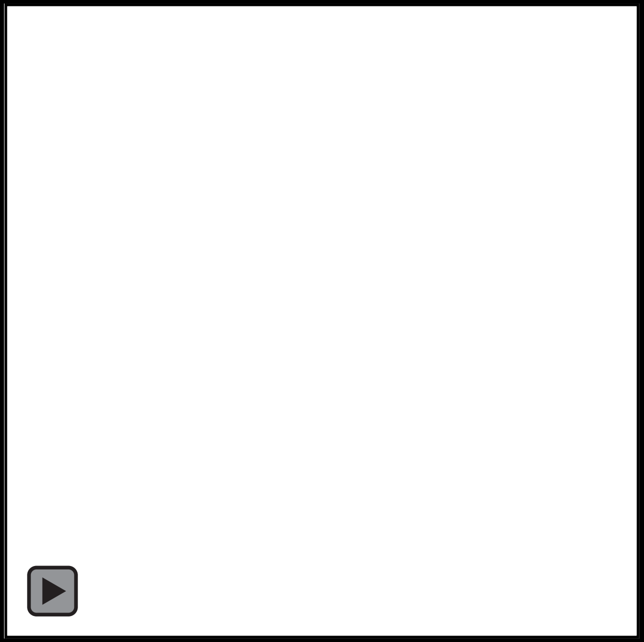



optimize
bunny
pose

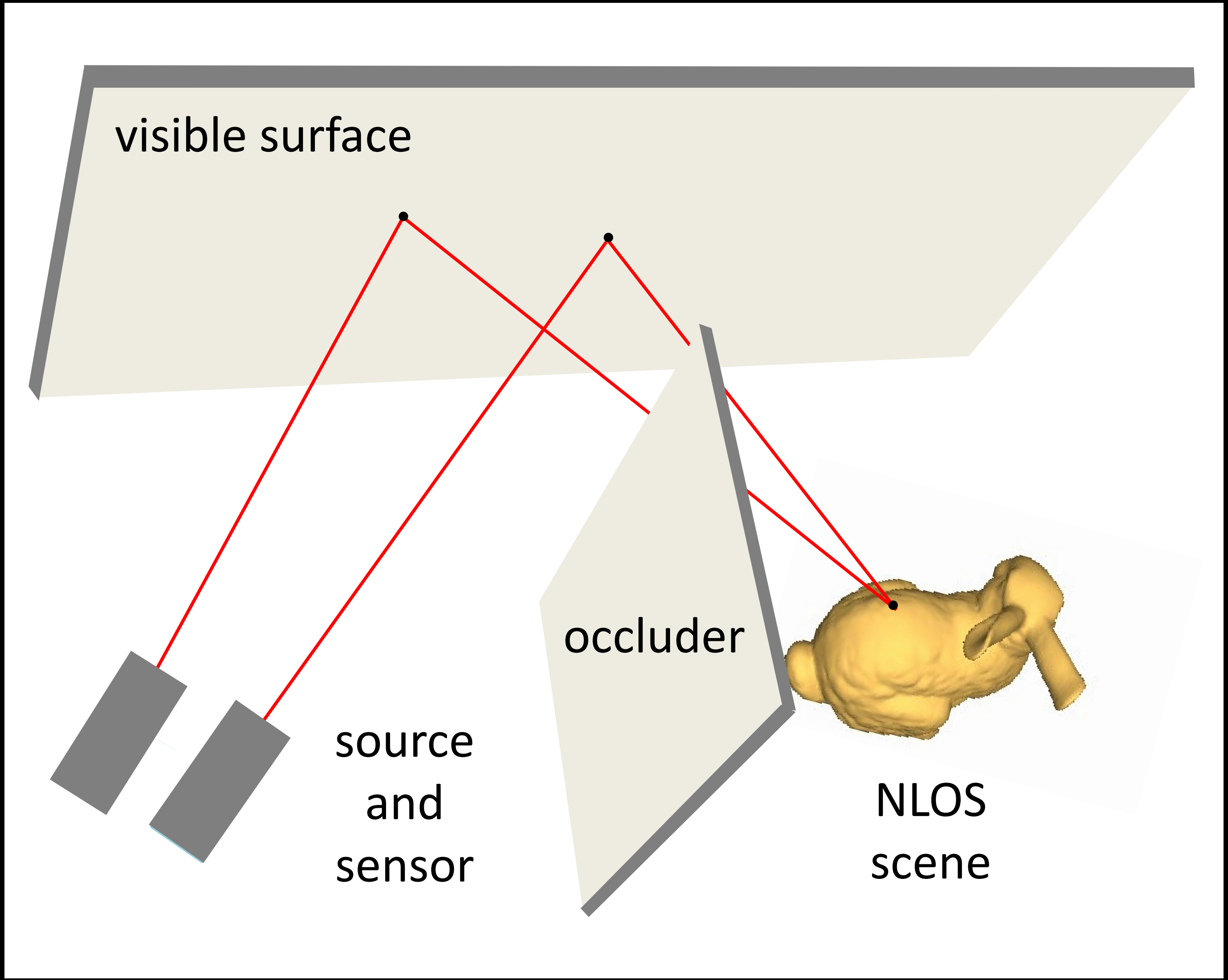


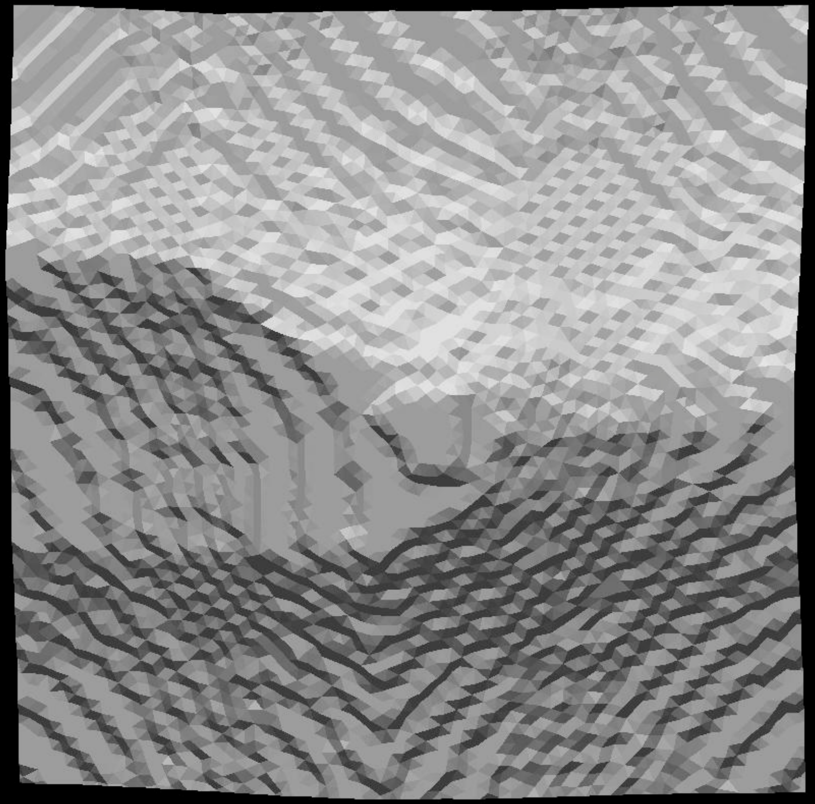



optimize
reflectance
and camera
pose

# Optimize shape



reconstruction evolution

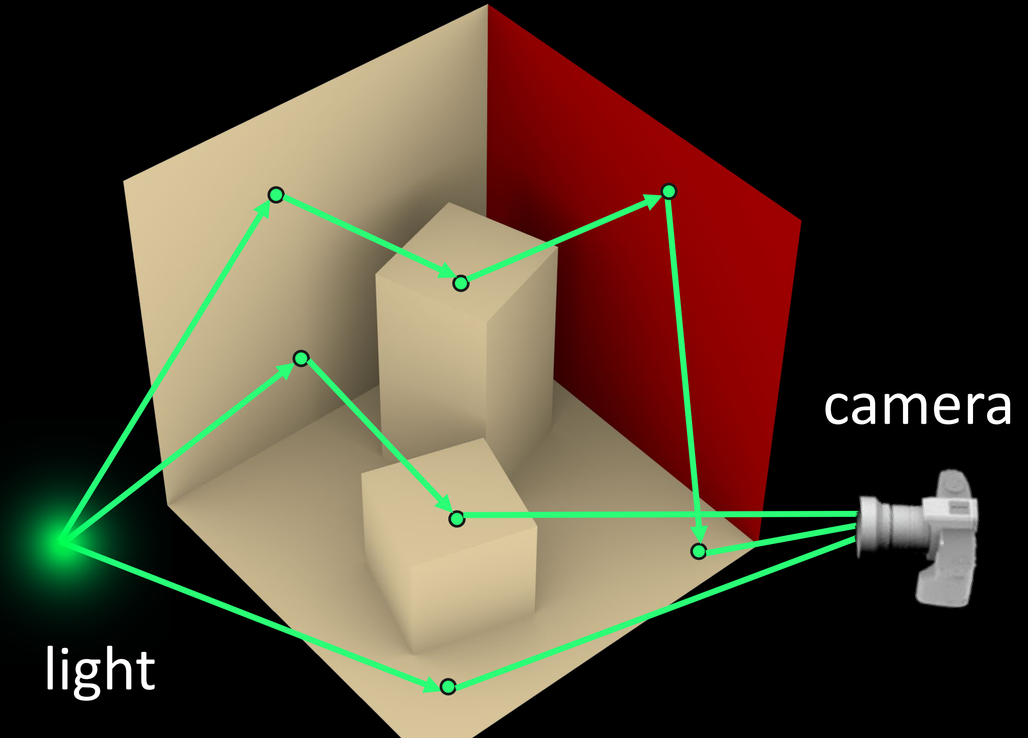
# Let's differentiate it

$$\frac{\partial}{\partial \pi} L(x, \omega; \pi)$$

$$= \int_{V(x,\pi)} F \underbrace{\left(\frac{\partial}{\partial \pi} L\right)}_{} \mathrm{d}A(x) + \int_{\partial V(x,\pi)} \underbrace{H(L)}_{} \mathrm{d}\sigma(x)$$

render derivative of L at some visible point

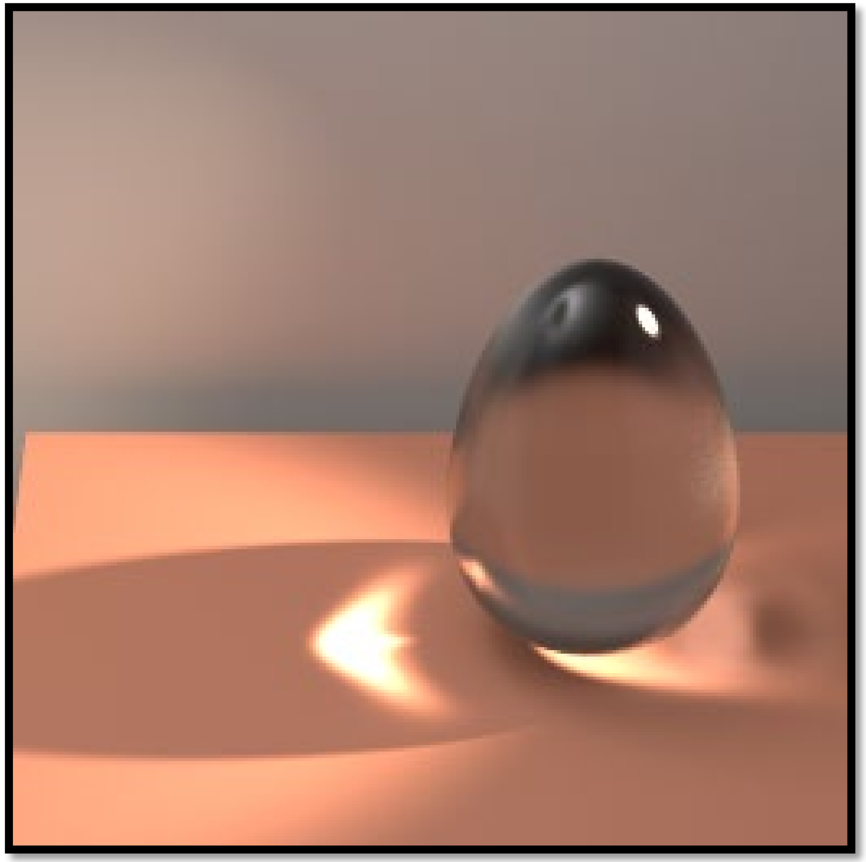
render L at some boundary (silhouette) point




Not terribly good:

- As we ray trace, we need to recompute silhouette
- Branching of two at each recursion

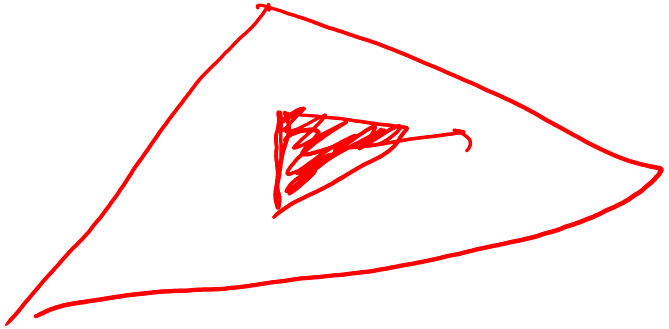
# CHALLENGES



Complex light transport effects

Complex geometry

# PATH-INTEGRAL FOR DIFFERENTIABLE RENDERING



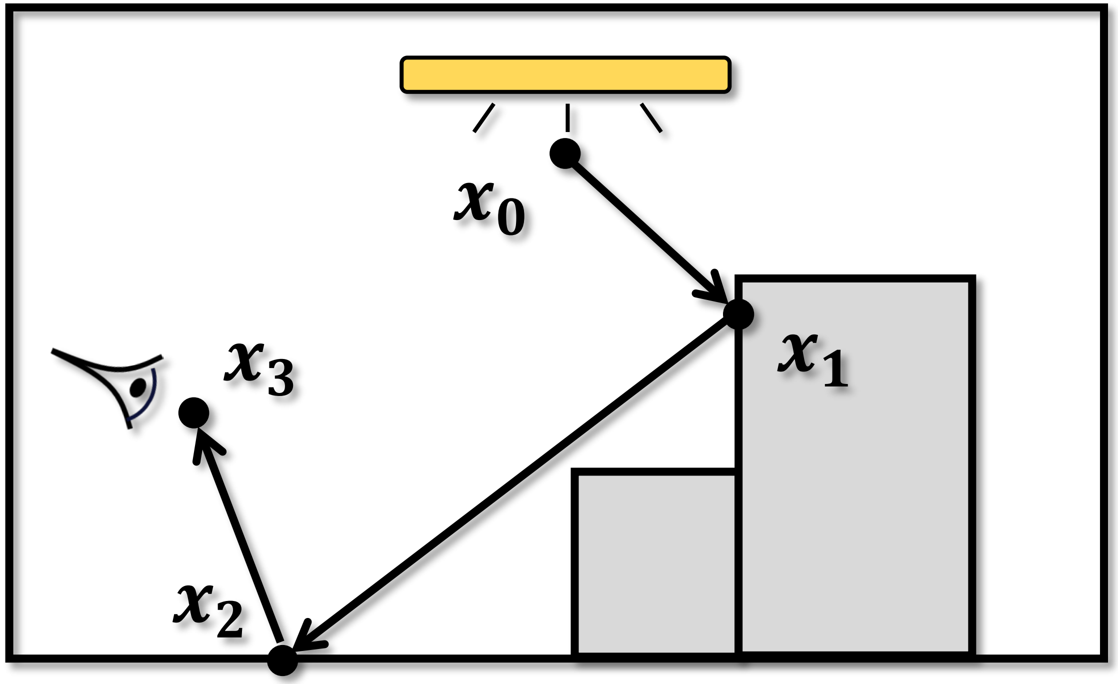
# FORWARD PATH INTEGRAL

Measurement contribution function

$$I = \int_{\Omega} f(\overline{\boldsymbol{x}}) \, \mathrm{d}\mu(\overline{\boldsymbol{x}})$$

Path space

Area-product measure




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

# DIFFERENTIAL PATH INTEGRAL

Path Integral

A generalization of Reynolds theorem

$$I = \int_\Omega f(\bar{\boldsymbol{x}})\,\mathrm{d}\mu(\bar{\boldsymbol{x}}) \quad \Rightarrow \quad \frac{\mathrm{d}I}{\mathrm{d}\pi} = \ ?$$

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

$$h_n^{(0)} := \left[\prod_{n'=n+1}^N g(\boldsymbol{x}_{n'};\, \boldsymbol{x}_{n'-2}, \boldsymbol{x}_{n'-1})\right] W_\mathrm{e}(\boldsymbol{x}_N \to \boldsymbol{x}_{N-1}), \quad (52)$$

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

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

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

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

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

and

$$\dot{h}_n(\boldsymbol{x}_n;\, \boldsymbol{x}_{n-1}) = \int_{\mathcal{M}^{N-n}} \left[\left(h_n^{(0)}\right)^{\boldsymbol{\cdot}} - h_n^{(0)} h_n^{(1)}\right] \prod_{n'=n+1}^N \mathrm{d}A(\boldsymbol{x}_{n'}) + \sum_{n'=n+1}^N \int \Delta h_{n,n'}^{(0)}\, V_{\overline{\partial\mathcal{M}}_{n'}}(\boldsymbol{x}_{n'})\, \mathrm{d}\ell(\boldsymbol{x}_{n'}) \prod_{\substack{n<i\le N \\ i\ne n'}} \mathrm{d}A(\boldsymbol{x}_i), \quad (56)$$

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 $\boldsymbol{x}_i$ with $i \ne n'$ and $\overline{\partial\mathcal{M}}_{n'}(\pi)$, which depends on $\boldsymbol{x}_{n'-1}$, for $\boldsymbol{x}_{n'}$.

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 $g_{n-1} := g(\boldsymbol{x}_n;\, \boldsymbol{x}_{n-2}, \boldsymbol{x}_{n-1})$ for all $0 < n \le N$. Then,

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

and

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

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

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

$$\dot{h}_0(\boldsymbol{x}_0) = \int_{\mathcal{M}^N} \left[\dot{f}(\bar{\boldsymbol{x}}) - f(\bar{\boldsymbol{x}}) \sum_{n'=1}^N \kappa(\boldsymbol{x}_{n'})\, V(\boldsymbol{x}_{n'})\right] \prod_{n'=1}^N \mathrm{d}A(\boldsymbol{x}_{n'}) + \sum_{n'=1}^N \int \Delta f_{n'}(\bar{\boldsymbol{x}})\, V_{\overline{\partial\mathcal{M}}_{n'}}\, \mathrm{d}\ell(\boldsymbol{x}_{n'}) \prod_{\substack{0<i\le N \\ i\ne n'}} \mathrm{d}A(\boldsymbol{x}_i). \quad (59)$$

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

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

Full derivation in the paper

# DIFFERENTIAL PATH INTEGRAL

Path Integral

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

A generalization of Reynolds theorem

Differential Path Integral

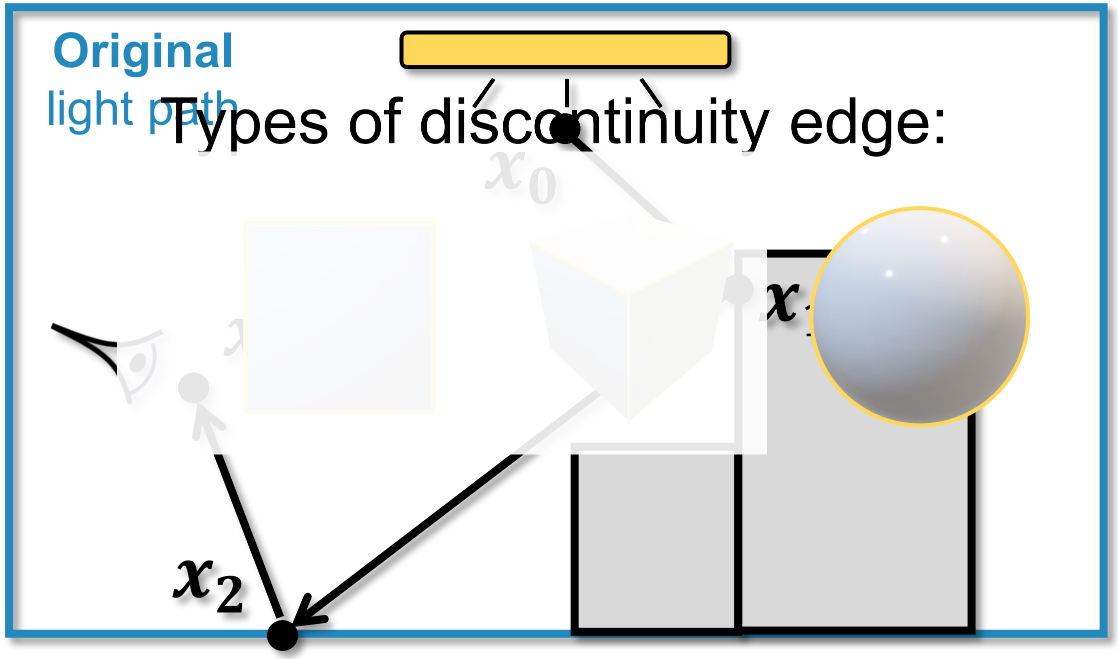
$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\Omega} \frac{\mathrm{d}}{\mathrm{d}\pi} f(\bar{\boldsymbol{x}})\,\mathrm{d}\mu(\bar{\boldsymbol{x}}) + \int_{\partial\Omega} g(\bar{\boldsymbol{x}})\,\mathrm{d}\mu'(\bar{\boldsymbol{x}})$$

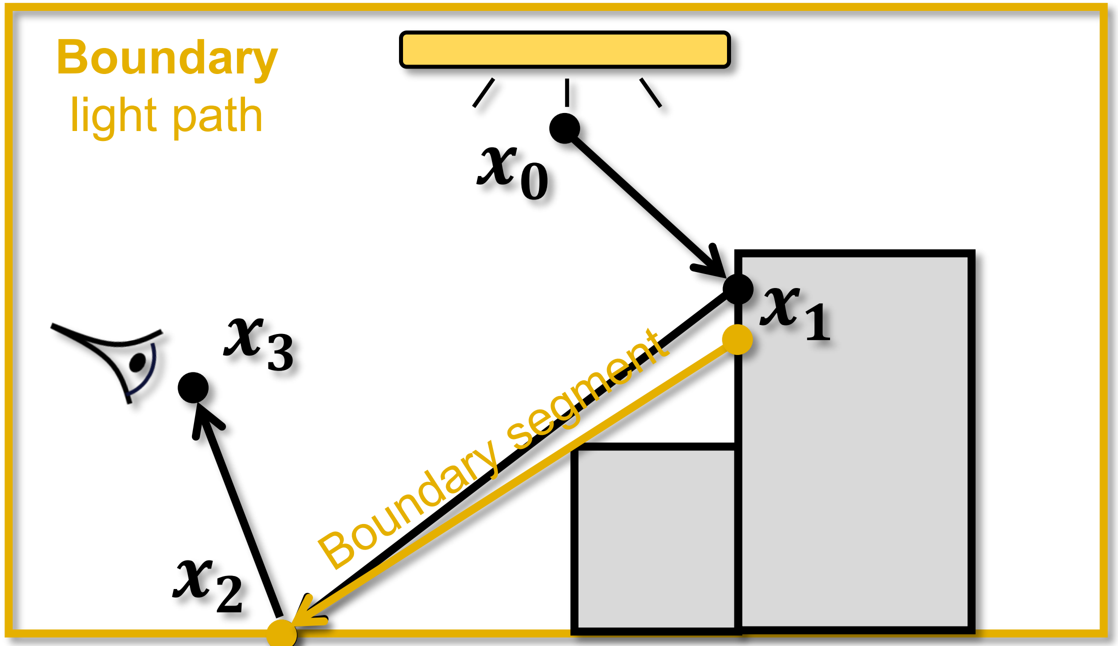
path space

Interior integral

boundary path space

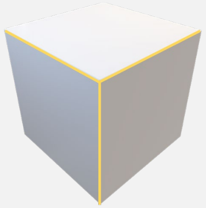
Boundary integral



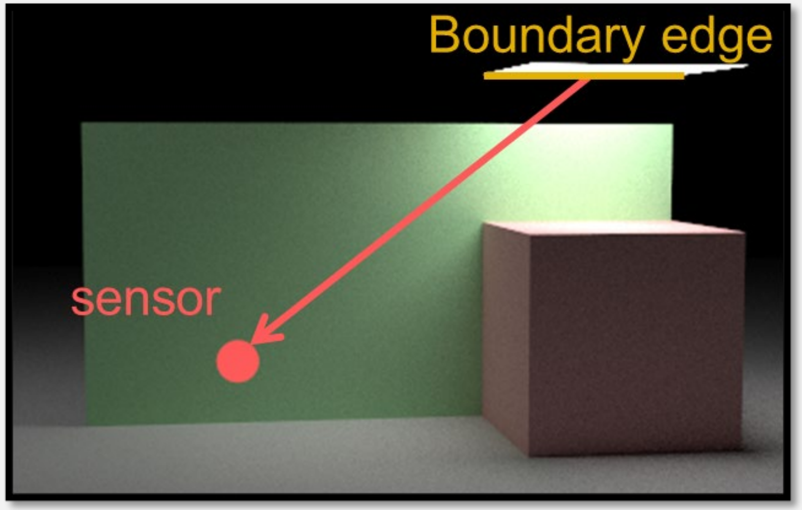



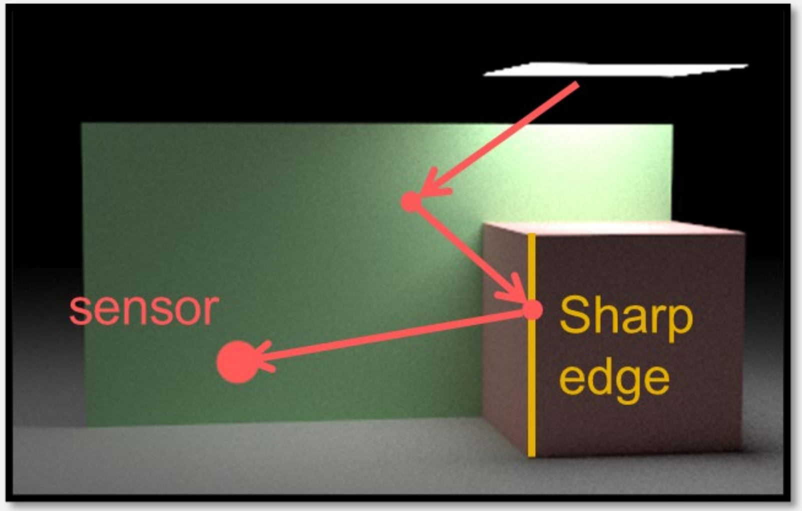
# SOURCE OF DISCONTINUITIES

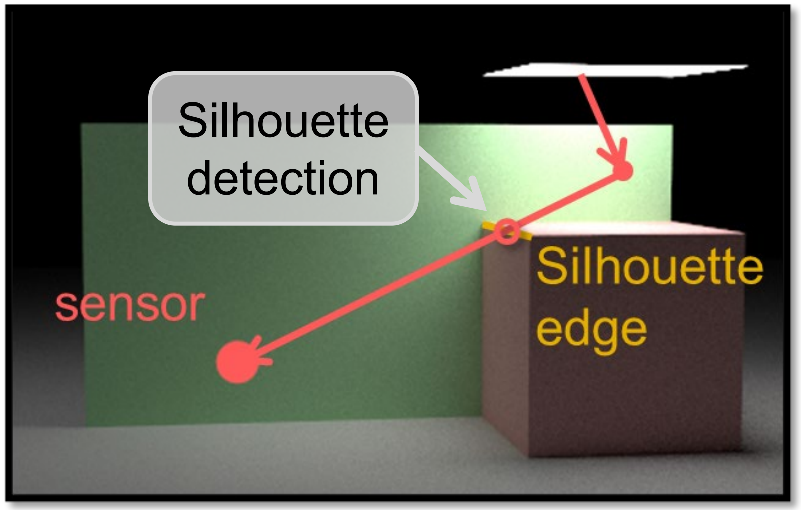
Boundary edge 

Sharp edge 

Silhouette edge 



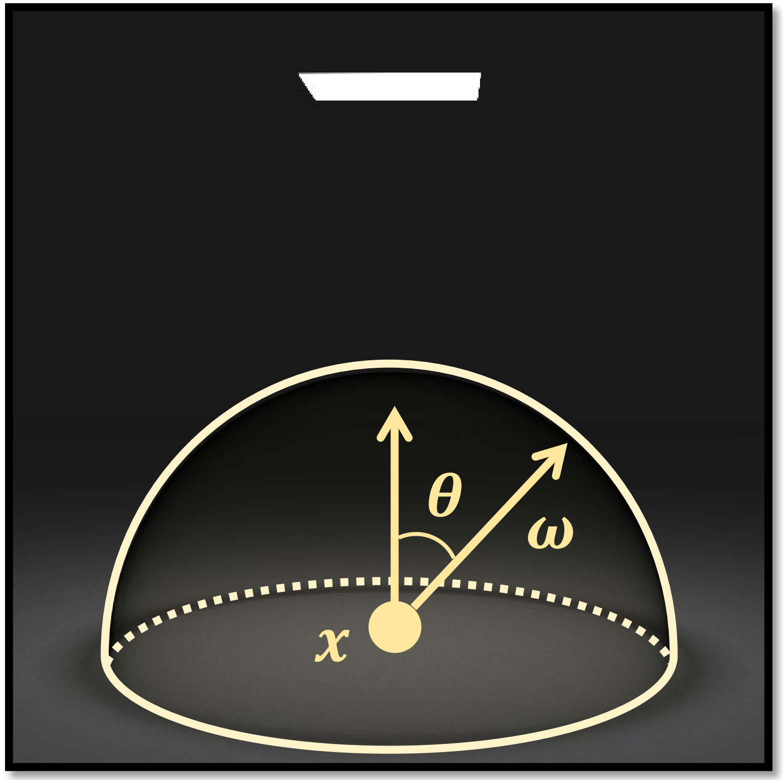

Topology-driven

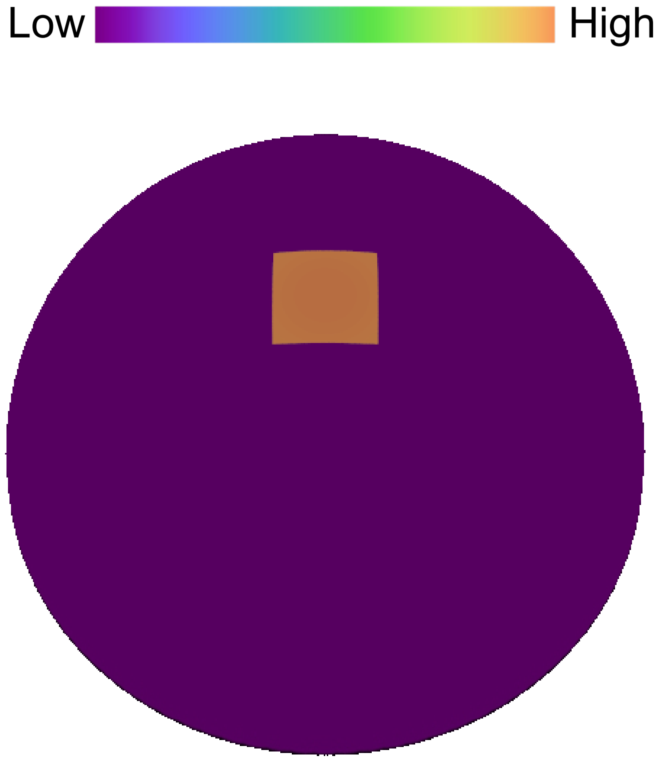
Visibility-driven

# REPARAMETERIZATIONS FOR SIMPLIFYING THE BOUNDARY TERM

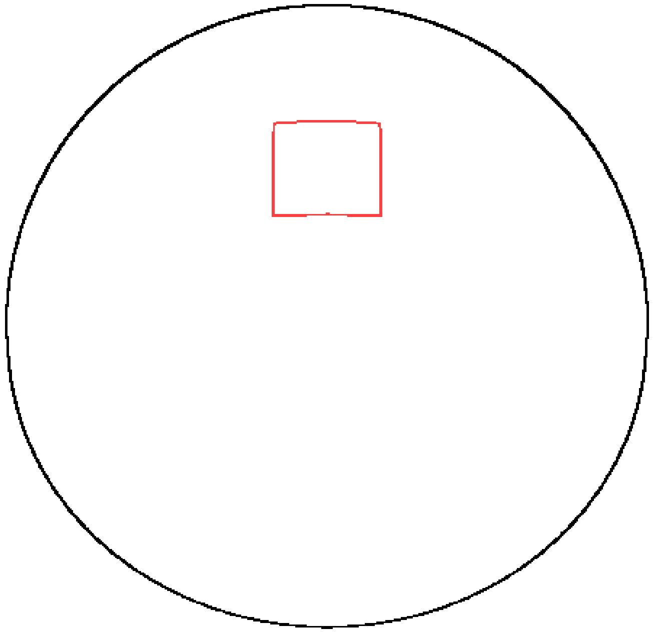
# REVISIT - DIFFERENTIAL IRRADIANCE

$\pi$: size of the emitter




Discontinuities of $f$



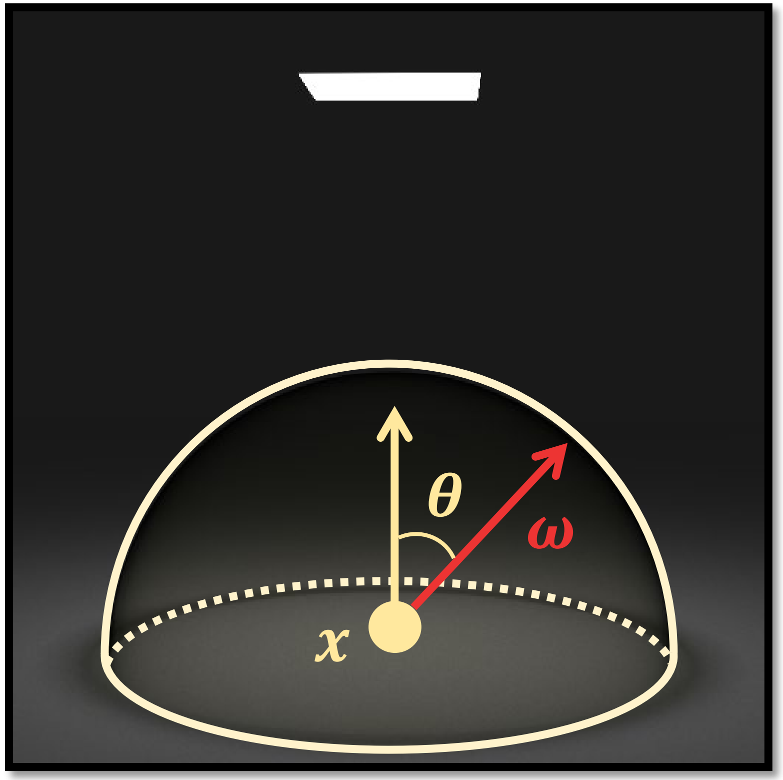
$$E = \int_{\mathbb{H}^2} \underbrace{L_i(\boldsymbol{\omega}) \cos\theta}_{f} \, \mathrm{d}\sigma(\boldsymbol{\omega})$$

Differentiation

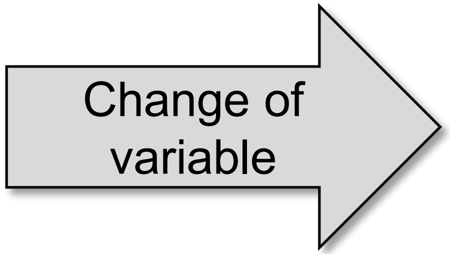
$$\frac{\mathrm{d}E}{\mathrm{d}\pi} = \int_{\mathbb{H}^2} \underbrace{\frac{\mathrm{d}f}{\mathrm{d}\pi}}_{=0} \mathrm{d}\sigma + \underbrace{\int_{\partial\mathbb{H}^2} g \, \mathrm{d}l}_{\neq 0}$$

# DIFFERENTIAL IRRADIANCE

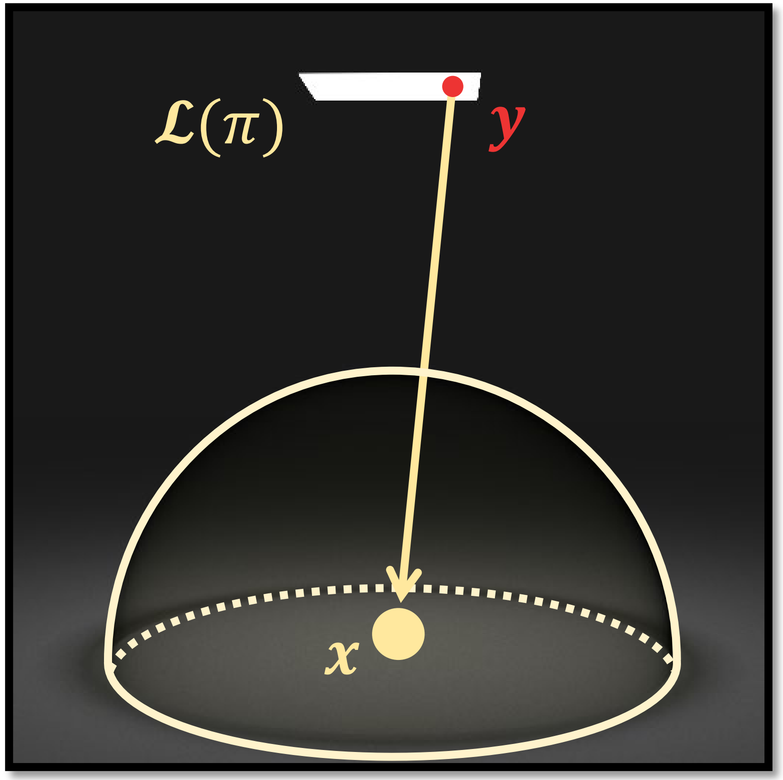
**Spherical** integral




$$E = \int_{\mathbb{H}^2} L_i(\omega) \cos\theta \, \mathrm{d}\sigma(\omega)$$



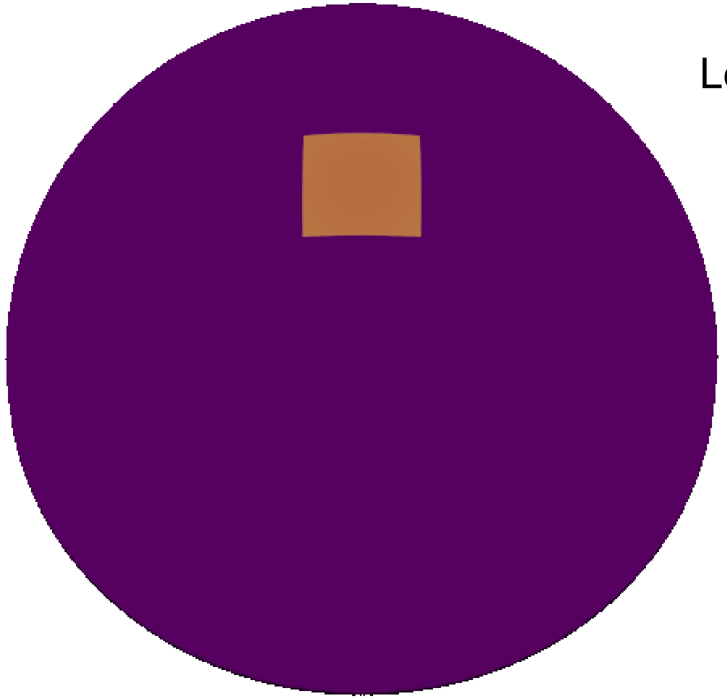

**Surface** integral




$$E = \int_{\mathcal{L}(\pi)} L_e(y \to x) \, G(x, y) \, \mathrm{d}A(y)$$

# DIFFERENTIAL DIRECT ILLUMINATION

**Spherical** integral




Change of variable

**Surface** integral

discontinuous

$$E = \int_{\mathbb{H}^2} \boxed{L_i(\boldsymbol{\omega}) \cos\theta} \, \mathrm{d}\sigma(\boldsymbol{\omega})$$

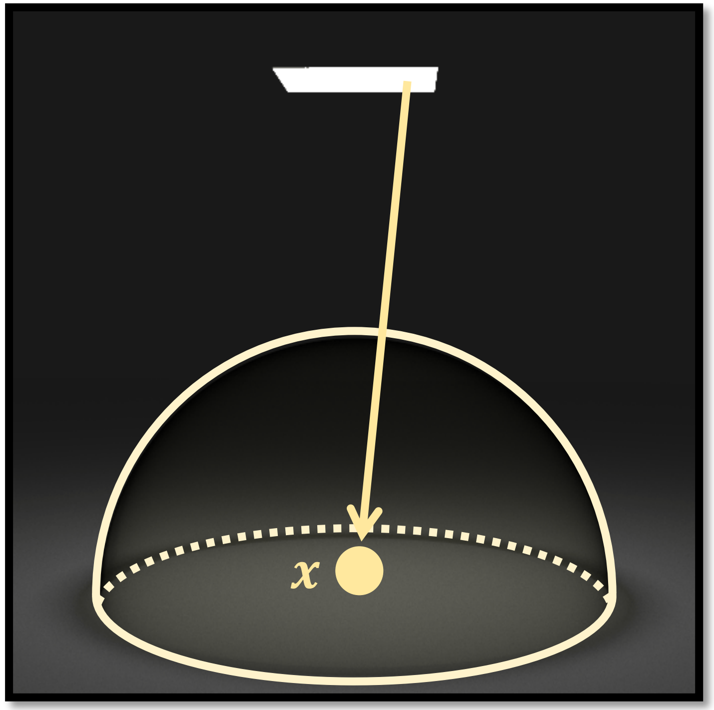
constant domain

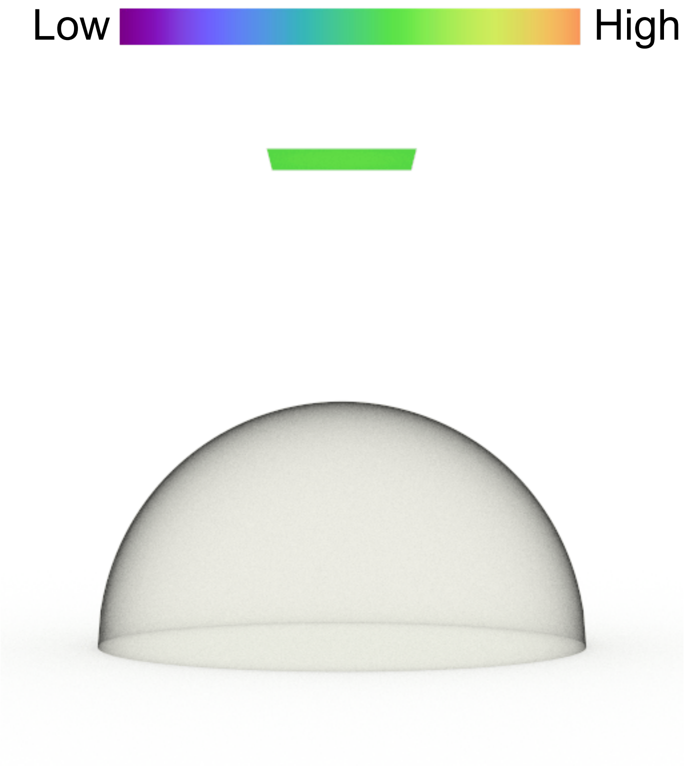
continuous

$$E = \int_{\mathcal{L}(\pi)} \boxed{L_e(\boldsymbol{y} \to \boldsymbol{x}) \, G(\boldsymbol{x}, \boldsymbol{y})} \, \mathrm{d}A(\boldsymbol{y})$$

evolving domain

# DIFFERENTIAL IRRADIANCE




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

$$E = \int_{\boldsymbol{\mathcal{L}}(\pi)} \overbrace{L_e(\boldsymbol{y} \to \boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}A(\boldsymbol{y})}^{f}$$

Differentiation

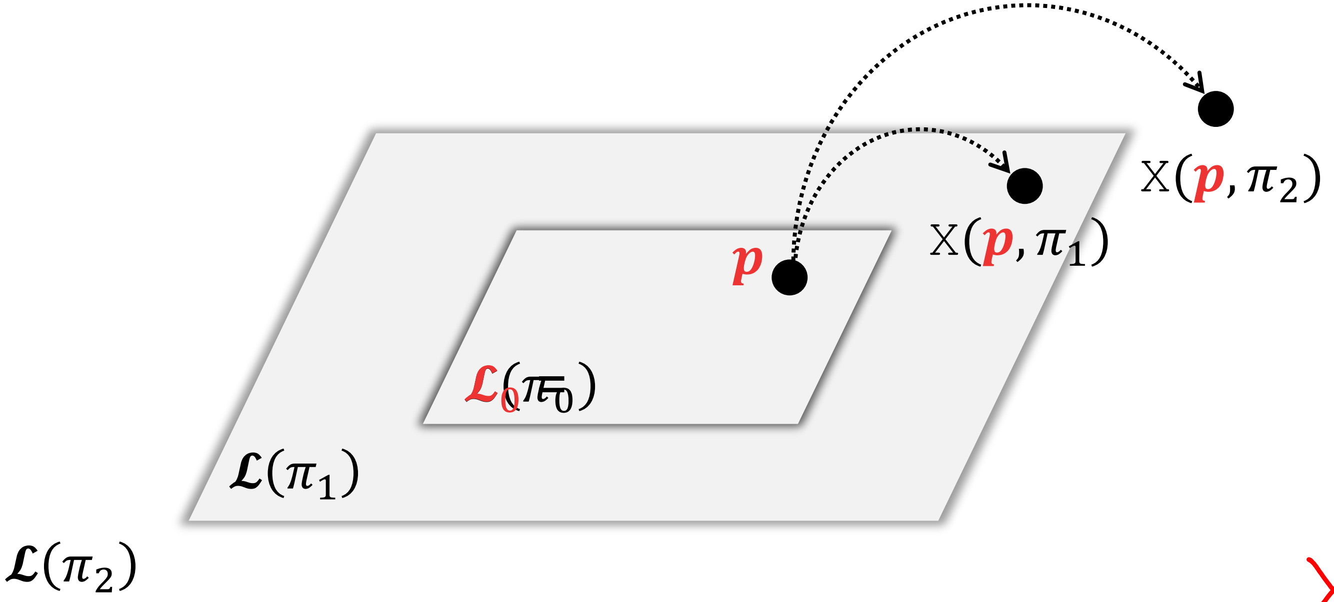
Reynolds theorem

$$\frac{\mathrm{d}E}{\mathrm{d}\pi} = \underbrace{\int_{\boldsymbol{\mathcal{L}}(\pi)} \frac{\mathrm{d}f}{\mathrm{d}\pi} \mathrm{d}A}_{\text{Interior}} + \underbrace{\int_{\partial \boldsymbol{\mathcal{L}}(\pi)} g \, \mathrm{d}l}_{\text{Boundary}}$$

$\neq 0$

# REPARAMETERIZATION

$$E = \int_{\mathcal{L}(\pi)} L_e(\boldsymbol{y} \to \boldsymbol{x})\, G(\boldsymbol{x}, \boldsymbol{y})\, \mathrm{d}A(\boldsymbol{y})$$




Parameterize $\mathcal{L}(\pi)$ using some fixed $\mathcal{L}_0$:

$$\boldsymbol{y} = \mathrm{X}(\boldsymbol{p}, \pi)$$

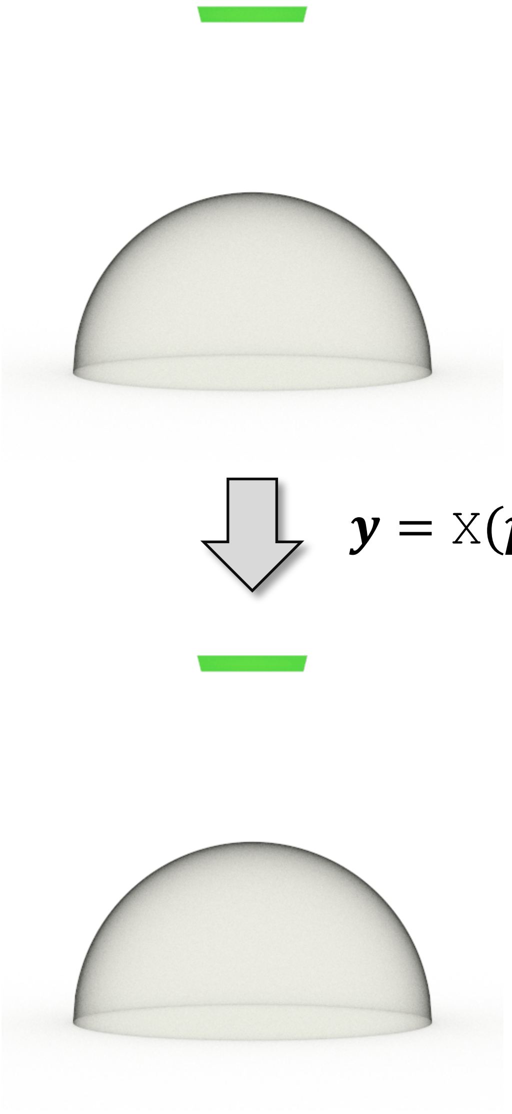
where $\mathrm{X}(\cdot\,, \pi)$ is one-to-one and continuous

$X(p, \pi)$

Reparameterization with $\boldsymbol{y} = \mathrm{X}(\boldsymbol{p}, \pi)$:

$$E = \int_{\mathcal{L}_0} L_e(\boldsymbol{y} \to \boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{y}) \underbrace{\left| \frac{\mathrm{d}A(\boldsymbol{y})}{\mathrm{d}A(\boldsymbol{p})} \right|} \mathrm{d}A(\boldsymbol{p})$$

# REPARAMETERIZATION



$\boldsymbol{y} = \mathbb{X}(\boldsymbol{p}, \pi)$

$$E = \int_{\mathcal{L}(\pi)} \overbrace{L_e(\boldsymbol{y} \to \boldsymbol{x})\, G(\boldsymbol{x}, \boldsymbol{y})}^{f}\, \mathrm{d}A(\boldsymbol{y})$$

$$\frac{\mathrm{d}E}{\mathrm{d}\pi} = \underbrace{\int_{\mathcal{L}(\pi)} \frac{\mathrm{d}f}{\mathrm{d}\pi}\, \mathrm{d}A}_{=\,0} + \underbrace{\int_{\partial\mathcal{L}(\pi)} g\, \mathrm{d}l}_{\neq\,0}$$

$$E = \int_{\mathcal{L}_0} \overbrace{L_e(\boldsymbol{y} \to \boldsymbol{x})\, G(\boldsymbol{x}, \boldsymbol{y}) \left| \frac{\mathrm{d}A(\boldsymbol{y})}{\mathrm{d}A(\boldsymbol{p})} \right|}^{f_0}\, \mathrm{d}A(\boldsymbol{p})$$

$$\frac{\mathrm{d}E}{\mathrm{d}\pi} = \underbrace{\int_{\mathcal{L}_0} \frac{\mathrm{d}f_0}{\mathrm{d}\pi}\, \mathrm{d}A}_{\neq\,0} + \underbrace{\int_{\partial\mathcal{L}_0} g_0\, \mathrm{d}l}_{=\,0}$$

# REPARAMETERIZATION

Reparameterization for irradiance

$$E = \int_{\mathcal{L}(\pi)} L_e(\boldsymbol{y} \to \boldsymbol{x}) G(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}A(\boldsymbol{y}) \xrightarrow{\boldsymbol{y} = \mathrm{X}(\boldsymbol{p}, \pi)} E = \int_{\mathcal{L}_0} L_e(\boldsymbol{y} \to \boldsymbol{x})\, G(\boldsymbol{x}, \boldsymbol{y}) \left| \frac{\mathrm{d}A(\boldsymbol{y})}{\mathrm{d}A(\boldsymbol{p})} \right| \mathrm{d}A(\boldsymbol{p})$$

Fixed surface

Reparameterization for path integral

$$I = \int_{\Omega(\pi)} f(\overline{\boldsymbol{x}})\, \mathrm{d}\mu(\overline{\boldsymbol{x}}) \xrightarrow{\overline{\boldsymbol{x}} = \mathrm{X}(\overline{\boldsymbol{p}}, \pi)} I = \int_{\Omega_0} f(\overline{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\overline{\boldsymbol{x}})}{\mathrm{d}\mu(\overline{\boldsymbol{p}})} \right| \mathrm{d}\mu(\overline{\boldsymbol{p}})$$

Fixed path space

$$\left| \frac{\mathrm{d}\mu(\overline{\boldsymbol{x}})}{\mathrm{d}\mu(\overline{\boldsymbol{p}})} \right| = \prod_i \left| \frac{\mathrm{d}A(\boldsymbol{x}_i)}{\mathrm{d}A(\boldsymbol{p}_i)} \right|$$

# DIFFERENTIAL PATH INTEGRAL

## Original

$$I = \int_{\Omega(\pi)} f(\bar{\boldsymbol{x}})\, \mathrm{d}\mu(\bar{\boldsymbol{x}})$$

$$\bar{\boldsymbol{x}} = \mathrm{X}(\bar{\boldsymbol{p}}, \pi)$$

## Reparameterized

$$I = \int_{\Omega_0} f(\bar{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\bar{\boldsymbol{x}})}{\mathrm{d}\mu(\bar{\boldsymbol{p}})} \right| \mathrm{d}\mu(\bar{\boldsymbol{p}})$$

## Original

$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\Omega(\pi)} \frac{\mathrm{d}f(\bar{\boldsymbol{x}})}{\mathrm{d}\pi}\, \mathrm{d}\mu(\bar{\boldsymbol{x}}) + \int_{\partial\Omega(\pi)} g(\bar{\boldsymbol{x}}) \mathrm{d}\mu'(\bar{\boldsymbol{x}})$$

**Pro:** No global parametrization required

**Con:** More types of discontinuities

## Reparameterized

$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\Omega_0} \frac{\mathrm{d}}{\mathrm{d}\pi} \left( f(\bar{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\bar{\boldsymbol{x}})}{\mathrm{d}\mu(\bar{\boldsymbol{p}})} \right| \right) \mathrm{d}\mu(\bar{\boldsymbol{p}}) + \int_{\partial\Omega_0} g(\bar{\boldsymbol{p}}) \mathrm{d}\mu'(\bar{\boldsymbol{p}})$$

**Con:** Requires global parametrization $\mathrm{X}$

**Pro:** Fewer types of discontinuities

# DIFFERENTIAL PATH INTEGRAL

Differential path integral

$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\Omega(\pi)} \frac{\mathrm{d}f(\bar{x})}{\mathrm{d}\pi}\,\mathrm{d}\mu(\bar{x}) + \int_{\partial\Omega(\pi)} g(\bar{x})\,\mathrm{d}\mu'(\bar{x})$$

$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\Omega_0} \frac{\mathrm{d}}{\mathrm{d}\pi}\left(f(\bar{x})\left|\frac{\mathrm{d}\mu(\bar{x})}{\mathrm{d}\mu(\bar{p})}\right|\right)\mathrm{d}\mu(\bar{p}) + \int_{\partial\Omega_0} g(\bar{p})\,\mathrm{d}\mu'(\bar{p})$$

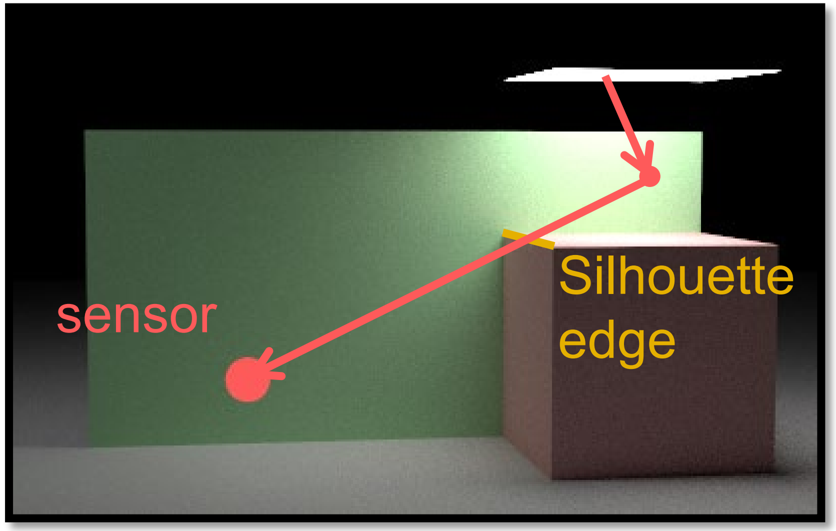
Topology-driven







Visibility-driven



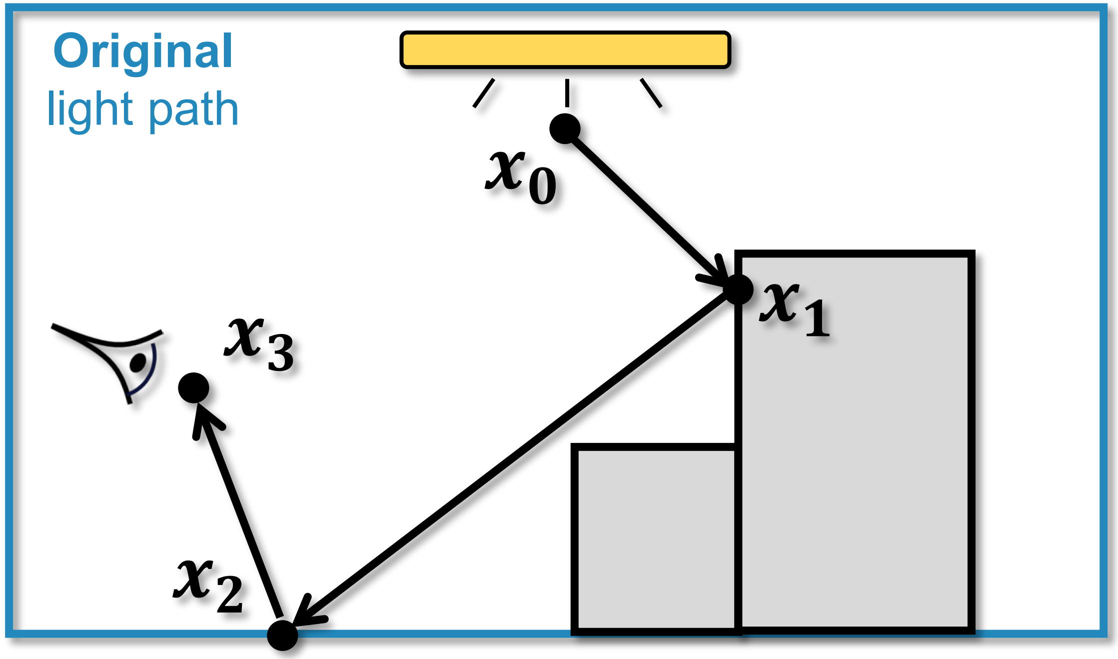
# MONTE CARLO ESTIMATORS

# ESTIMATING INTERIOR INTEGRAL

(Reparameterized)
Differential path Integral

$$\frac{\partial I}{\partial \pi} = \underbrace{\int_{\Omega_0} \frac{\partial}{\partial \pi}\left( f(\bar{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\bar{\boldsymbol{x}})}{\mathrm{d}\mu(\bar{\boldsymbol{p}})} \right| \right) \mathrm{d}\mu(\bar{\boldsymbol{p}})}_{\text{Interior integral}} + \underbrace{\int_{\partial\Omega_0} g(\bar{\boldsymbol{p}}) \mathrm{d}\mu'(\bar{\boldsymbol{p}})}_{\text{Boundary integral}}$$

Different MC estimators



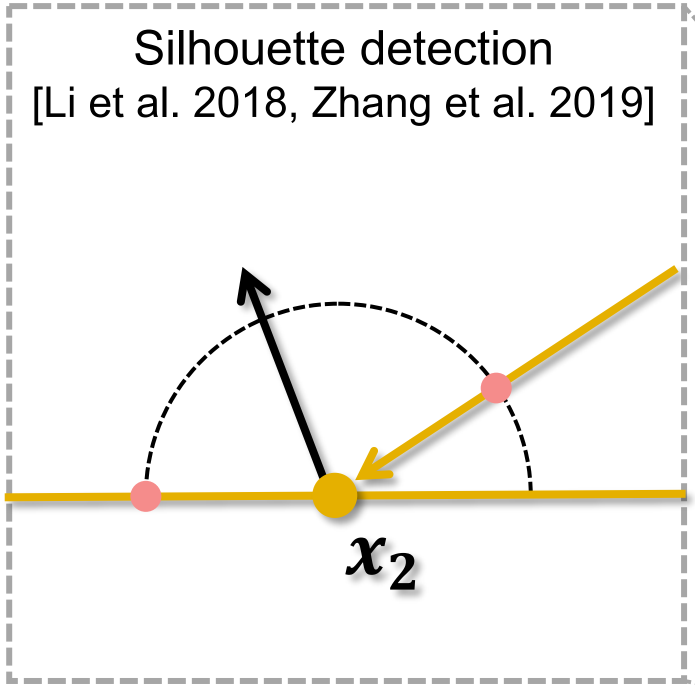

- Can be estimated using identical path sampling strategies as forward rendering
  - Unidirectional path tracing
  - Bidirectional path tracing
  - …

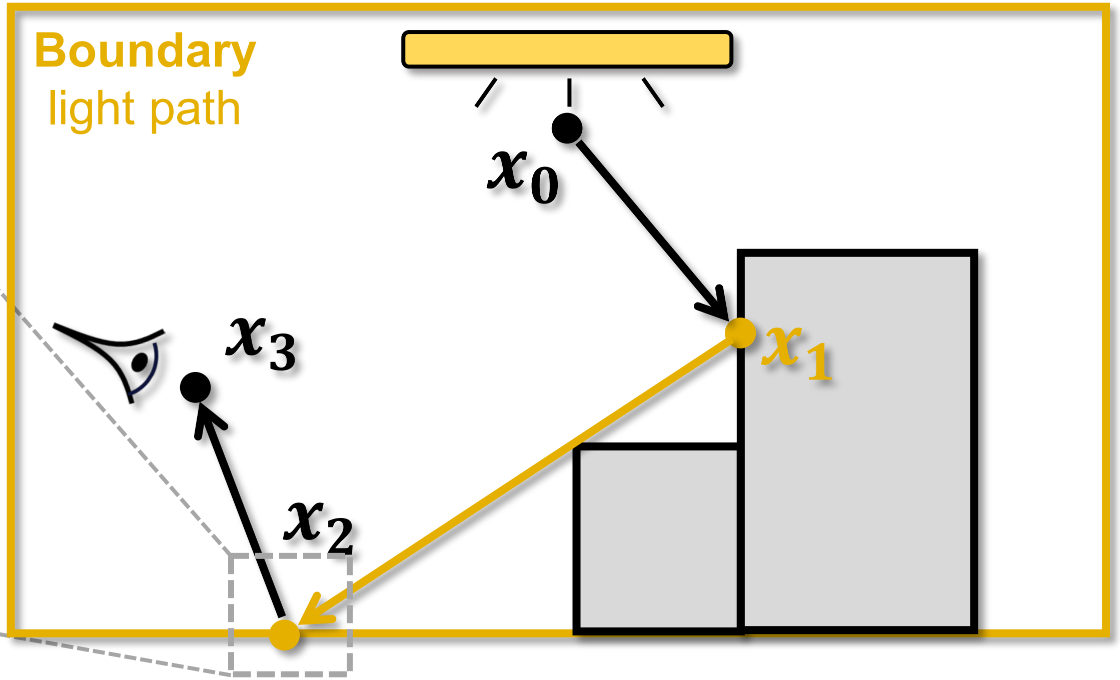
# ESTIMATING BOUNDARY INTEGRAL

(Reparameterized)
Differential path Integral

$$\frac{\partial I}{\partial \pi} = \int_{\Omega_0} \frac{\partial}{\partial \pi}\left( f(\overline{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\overline{\boldsymbol{x}})}{\mathrm{d}\mu(\overline{\boldsymbol{p}})} \right| \right) \mathrm{d}\mu(\overline{\boldsymbol{p}}) + \boxed{\int_{\partial\Omega_0} g(\overline{\boldsymbol{p}}) \mathrm{d}\mu'(\overline{\boldsymbol{p}})}$$

Boundary integral



# ESTIMATING BOUNDARY INTEGRAL

(Reparameterized)
Differential path Integral

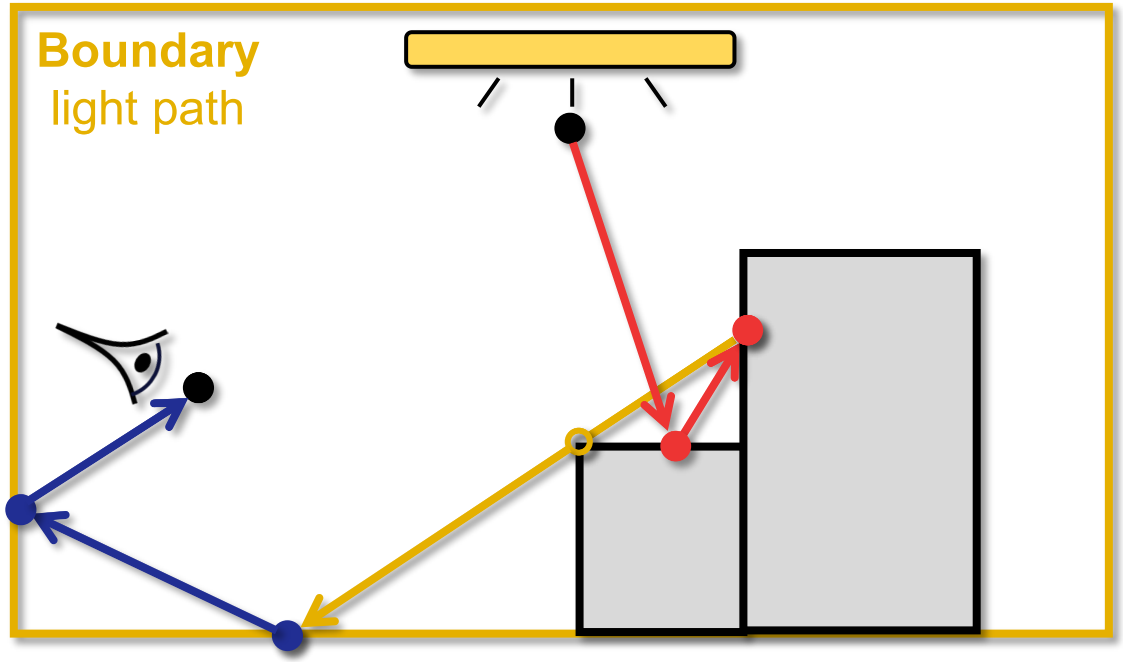
$$\frac{\partial I}{\partial \pi} = \int_{\Omega_0} \frac{\partial}{\partial \pi}\left( f(\overline{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\overline{\boldsymbol{x}})}{\mathrm{d}\mu(\overline{\boldsymbol{p}})} \right| \right) \mathrm{d}\mu(\overline{\boldsymbol{p}}) + \boxed{\int_{\partial\Omega_0} g(\overline{\boldsymbol{p}}) \mathrm{d}\mu'(\overline{\boldsymbol{p}})}$$

where $\overline{\boldsymbol{x}} = \mathrm{X}(\overline{\boldsymbol{p}}, \pi)$

Boundary integral

- Construct boundary segment
- Construct source and sensor subpaths

- To improve efficiency
  - Next-event estimation
  - Importance sampling of boundary segments



# OUR ESTIMATORS

**Unidirectional** estimator

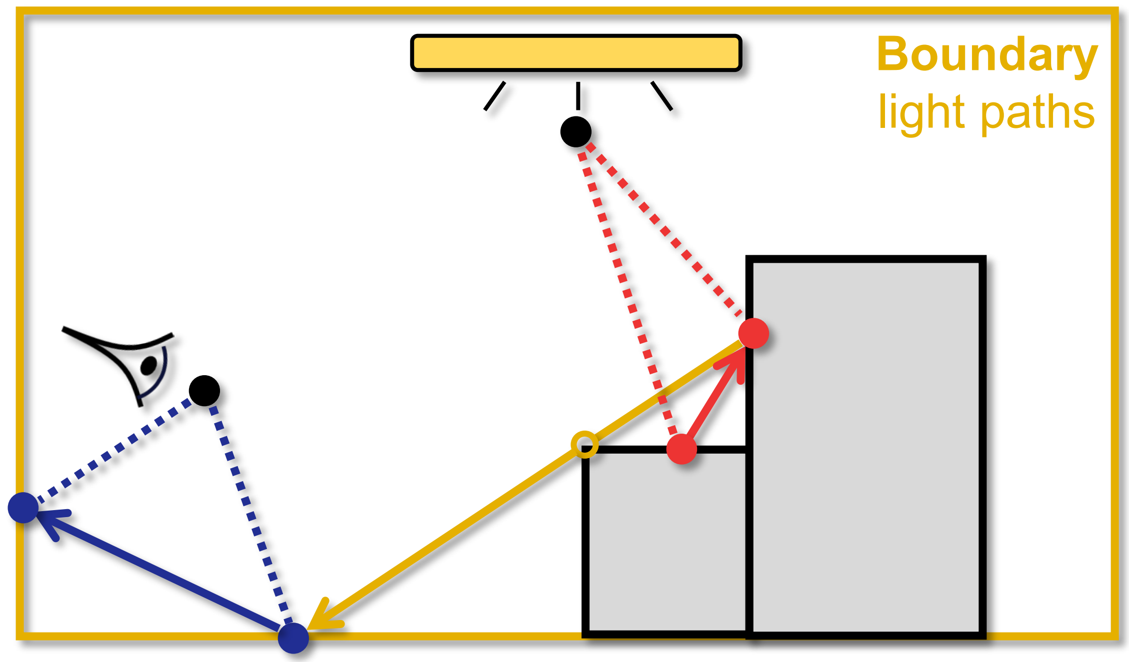
Interior: **unidirectional** path tracing

Boundary: **unidirectional** sampling of subpaths

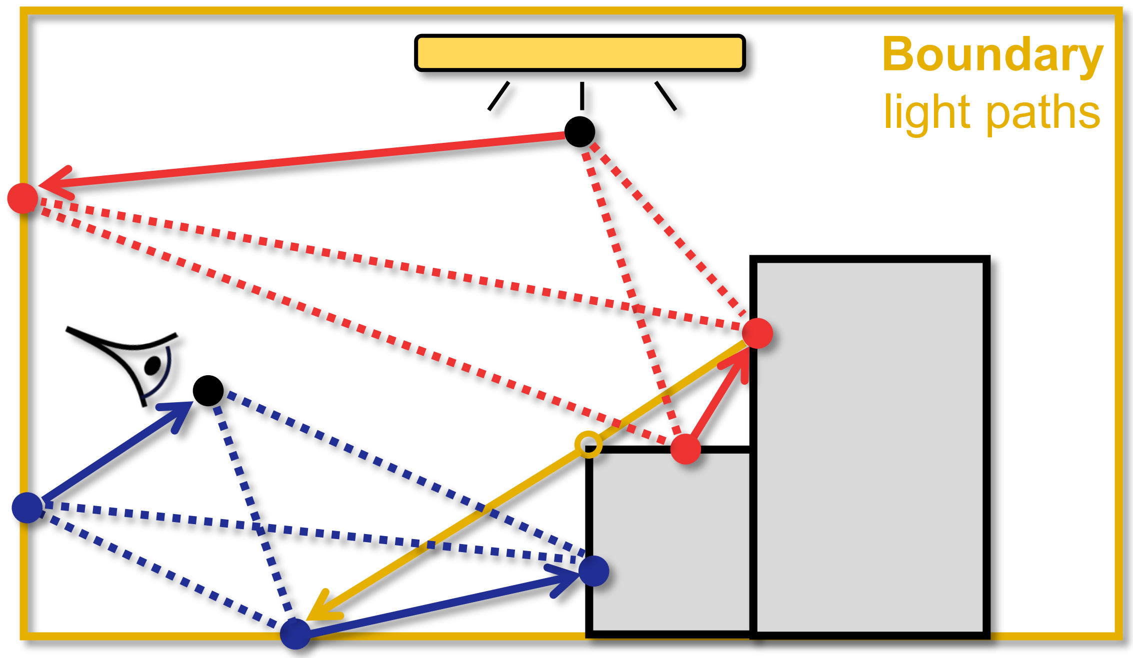
**Bidirectional** estimator

Interior: **bidirectional** path tracing

Boundary: **bidirectional** sampling of subpaths



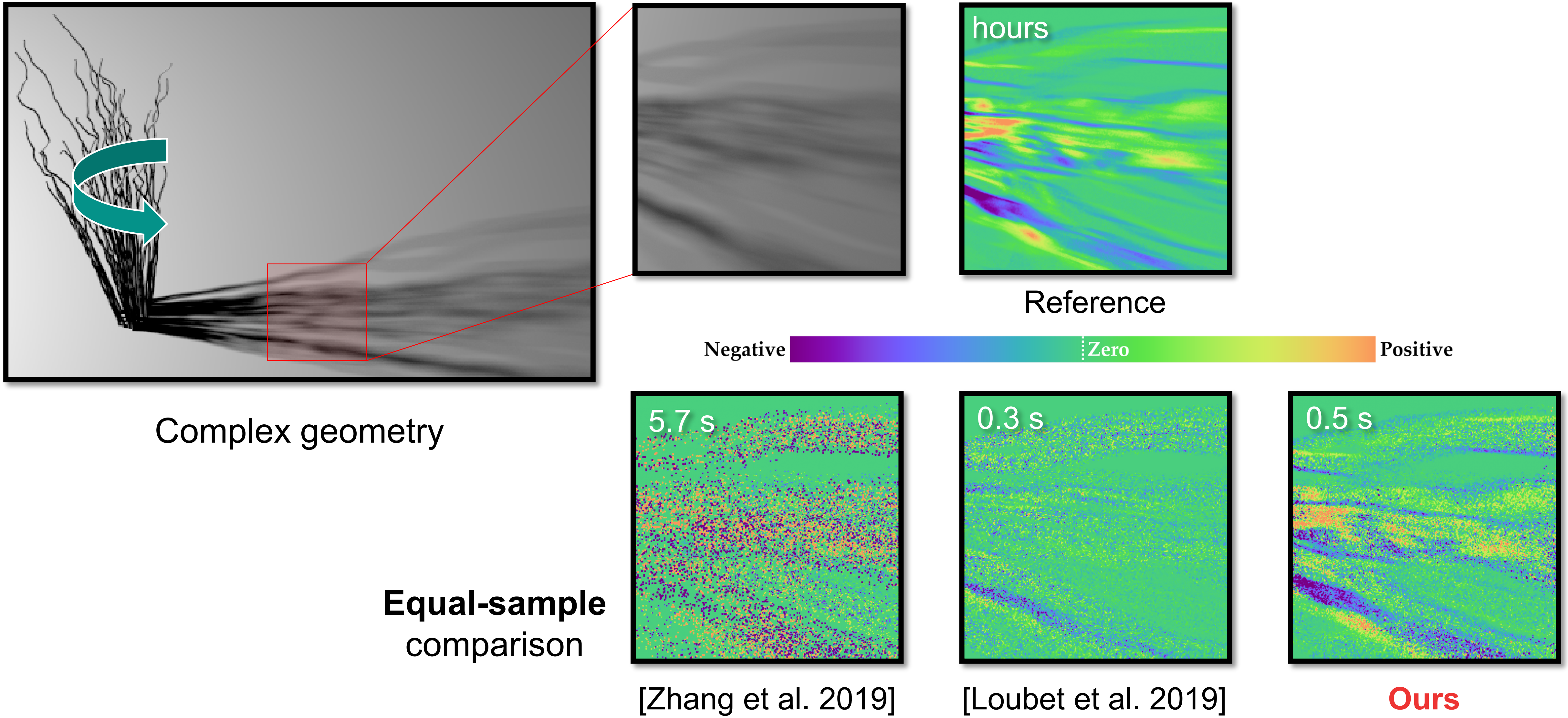

**Unidirectional** path tracing + NEE




**Bidirectional** path tracing

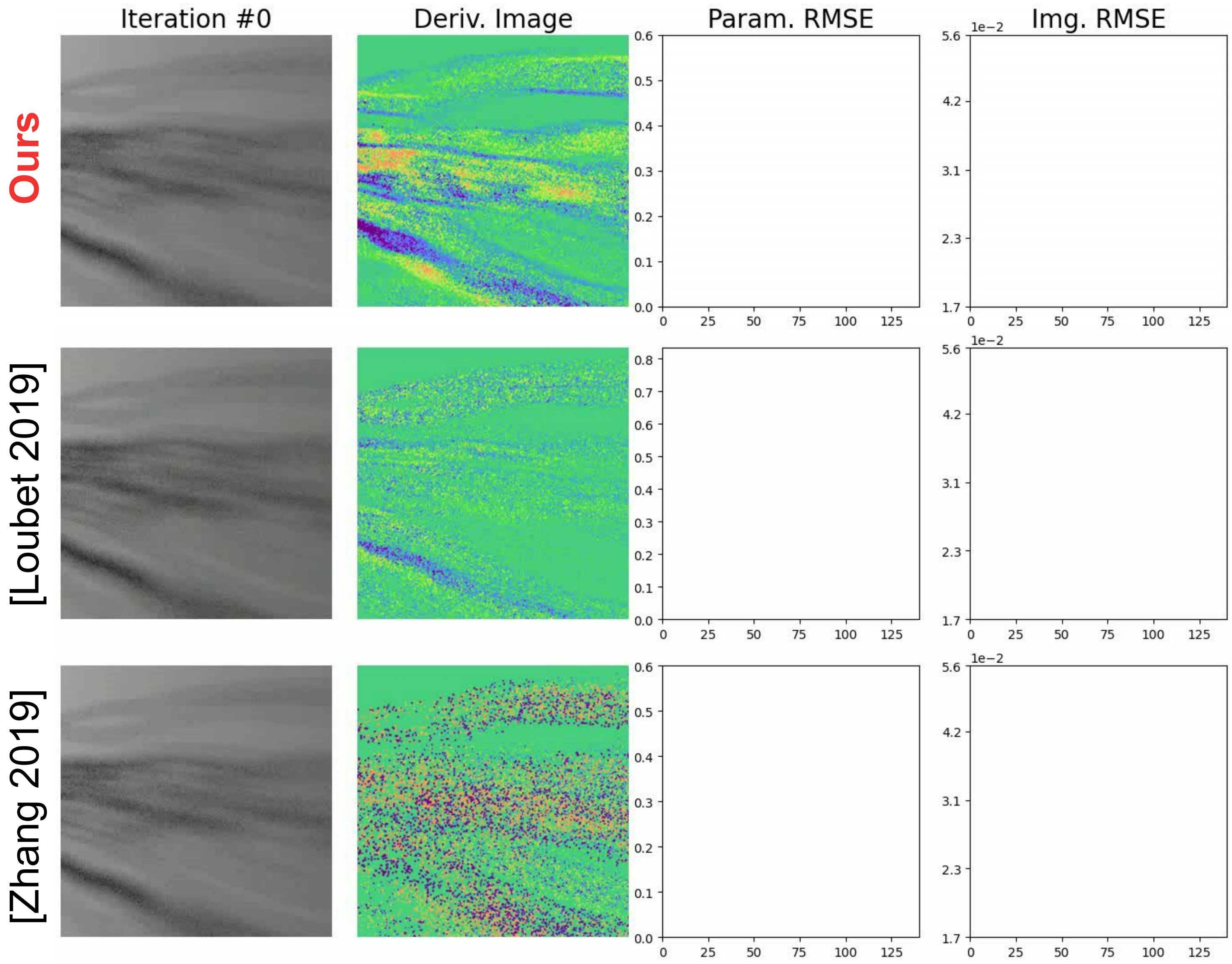
# SOME RESULTS

# HANDLING COMPLEX GEOMETRY



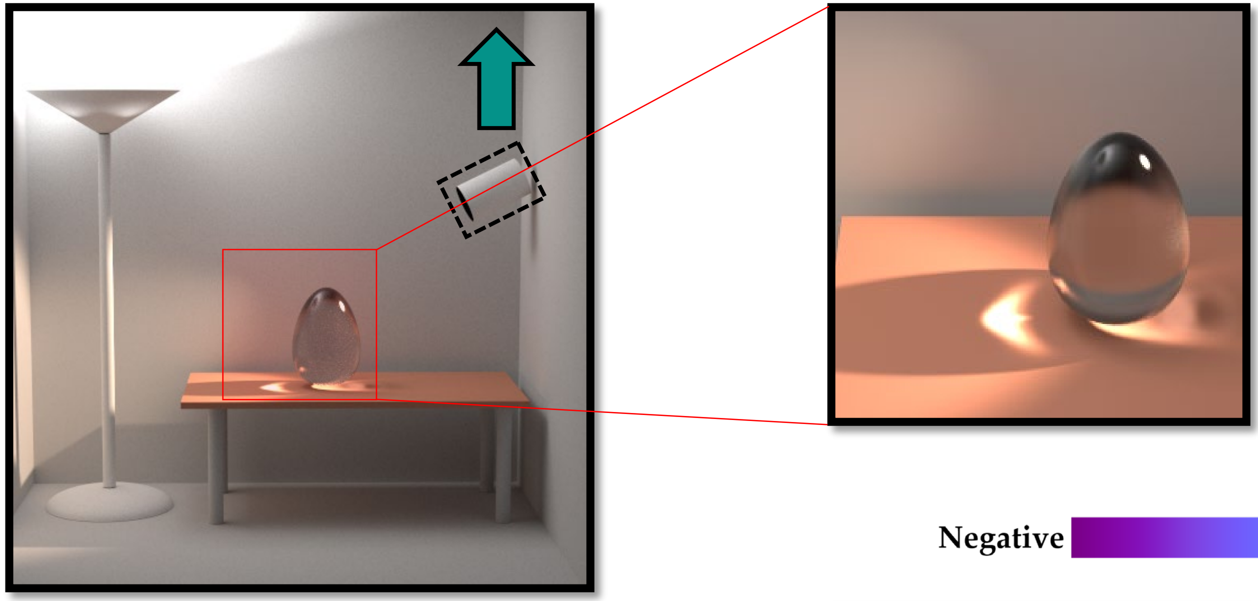
# HANDLING COMPLEX GEOMETRY

Target image

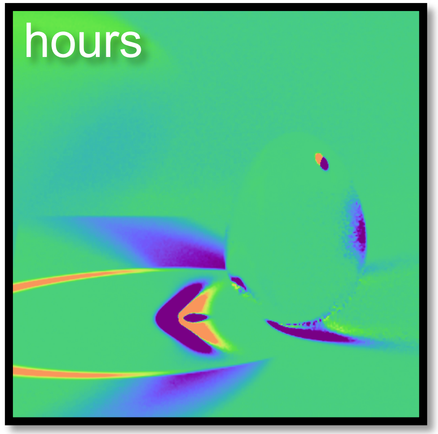



- Optimizing *rotation angle*
- **Equal-sample** per iteration
- **Identical** optimization setting
  - Learning rate (Adam)
  - Initializations

# HANDLING CAUSTICS

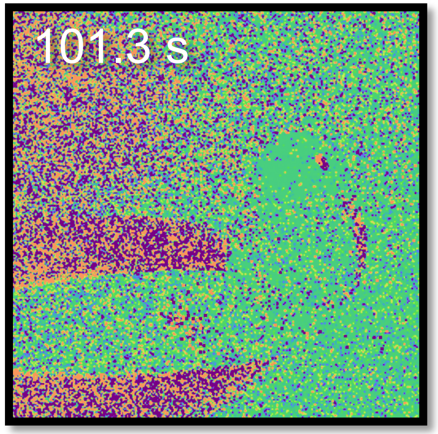


Complex light transport effects

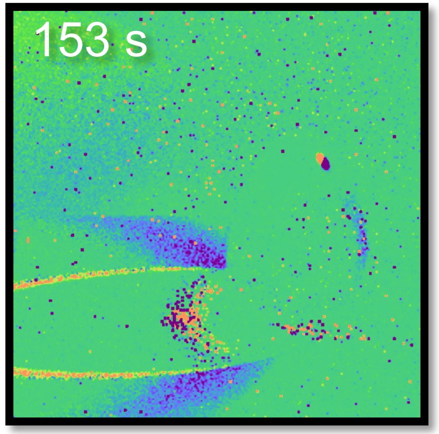



Reference

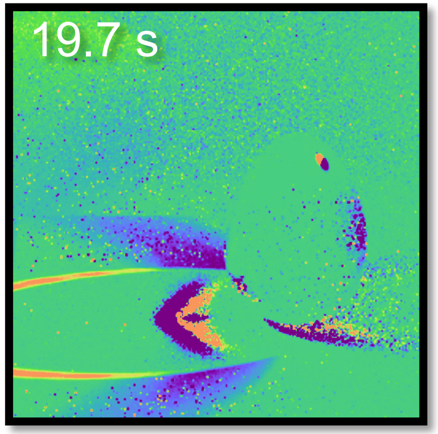

**Equal-sample**
comparison




[Zhang et al. 2019]

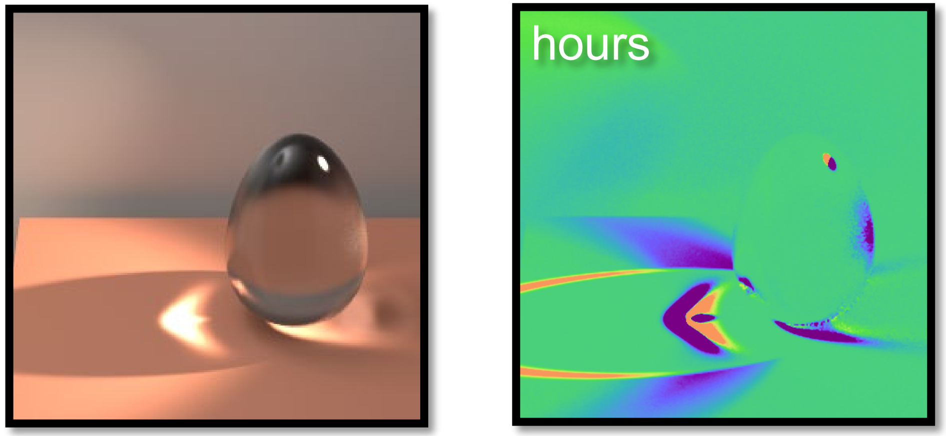



[Loubet et al. 2019]



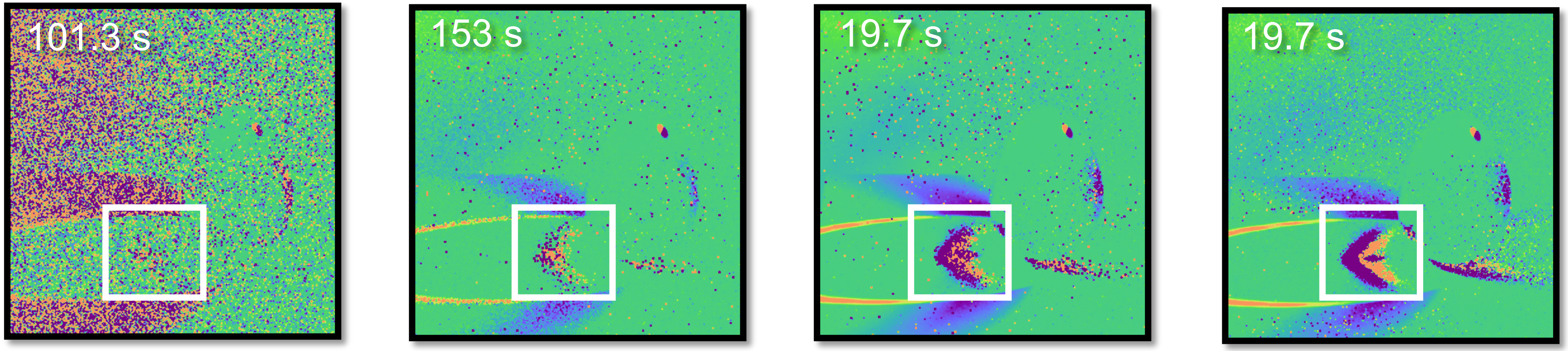

**Ours**

# HANDLING CAUSTICS




Reference

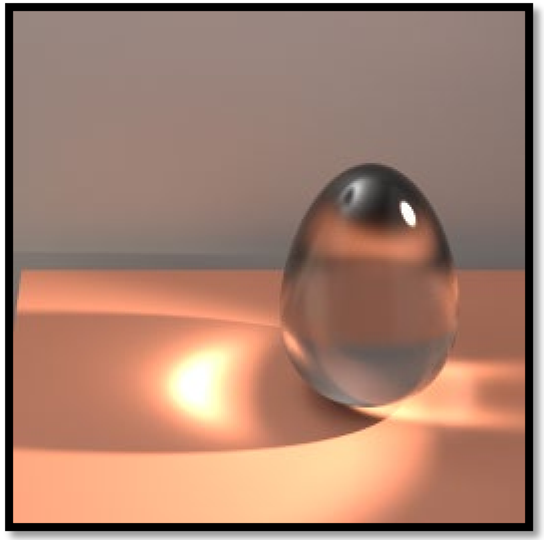
**Equal-sample** comparison

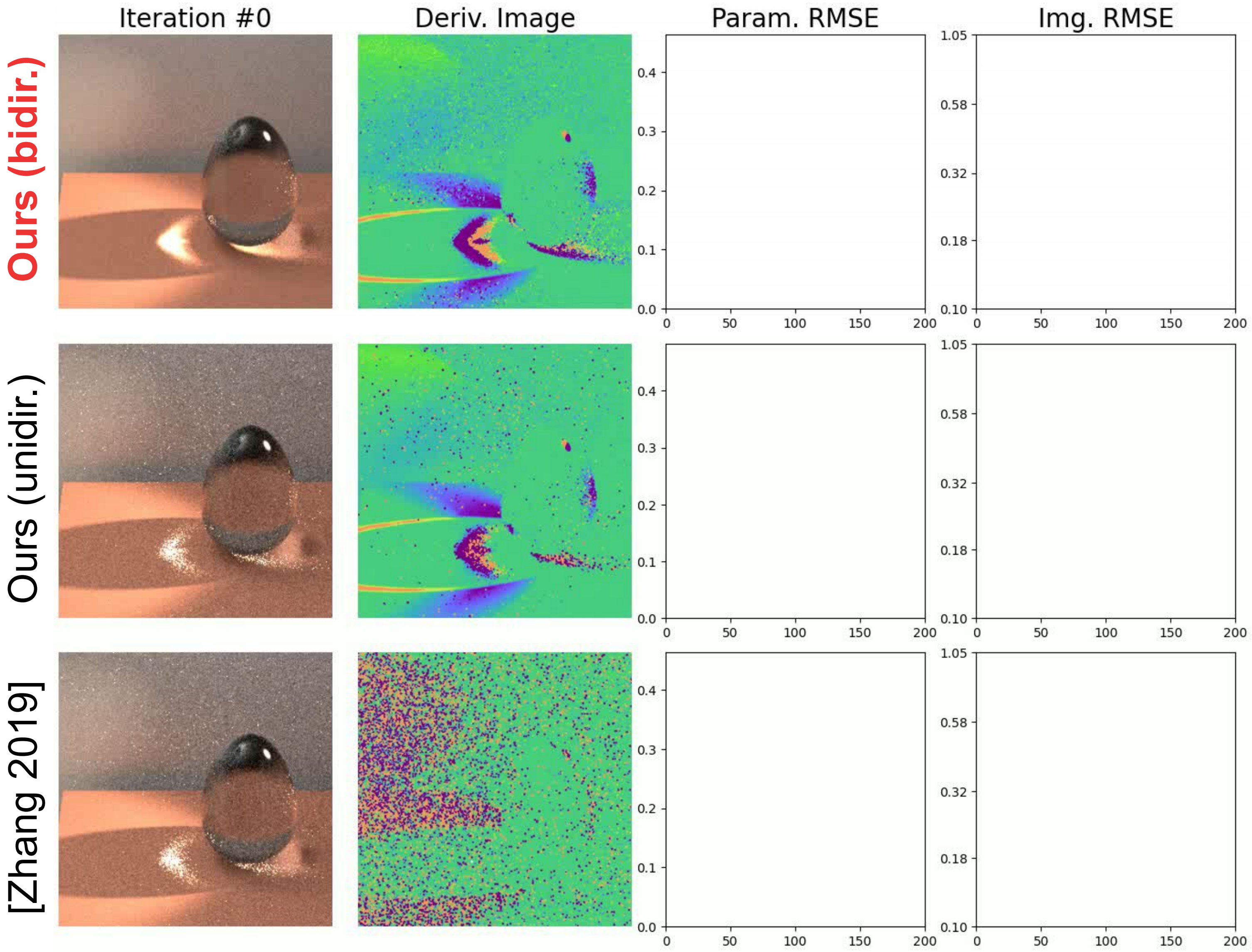




[Zhang et al. 2019] [Loubet et al. 2019] **Ours (unidirectional)** **Ours (bidirectional)**

# HANDLING CAUSTICS

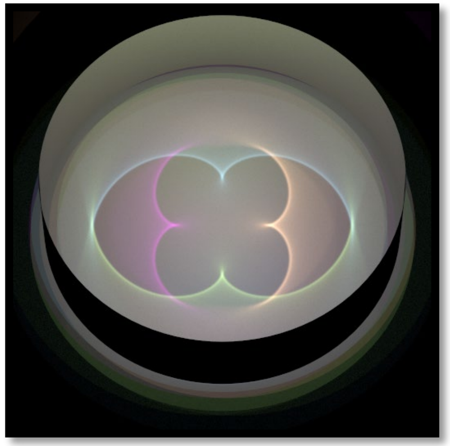
Target image



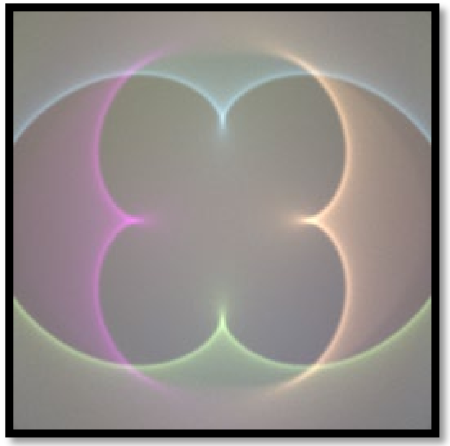



- Optimizing
  - Glass IOR
  - Spotlight position
- **Equal-time** per iteration
- **Identical** optimization setting

# SHAPE OPTIMIZATION

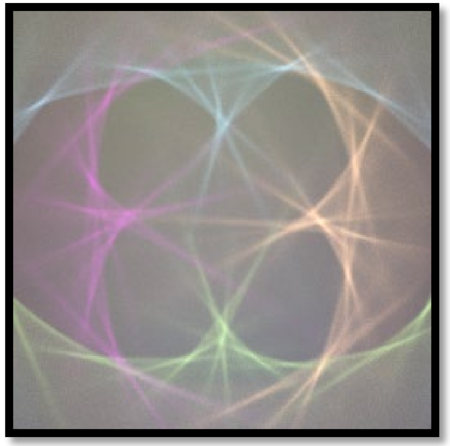


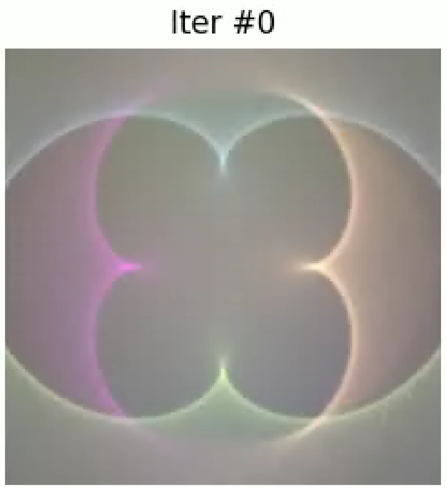
Initial

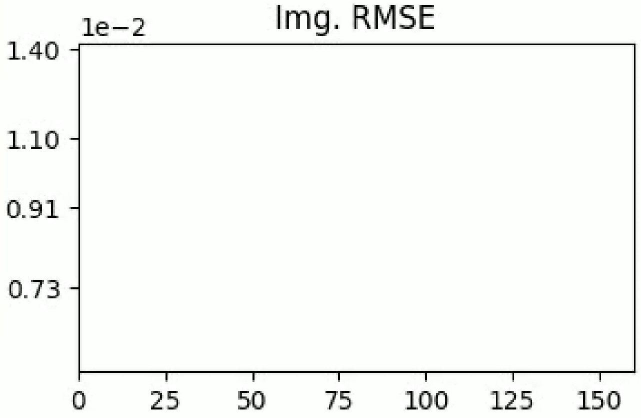


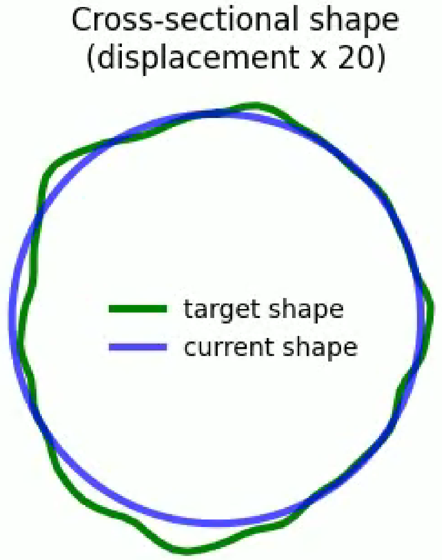
Optimizing **cross-sectional** shape (100 variables)

Target image



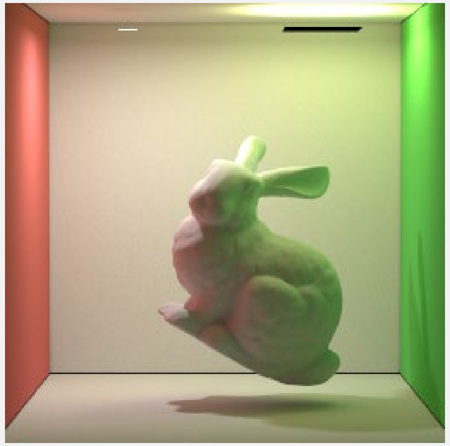


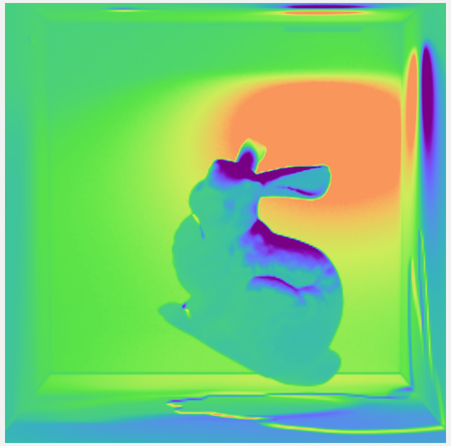



# RESULTS

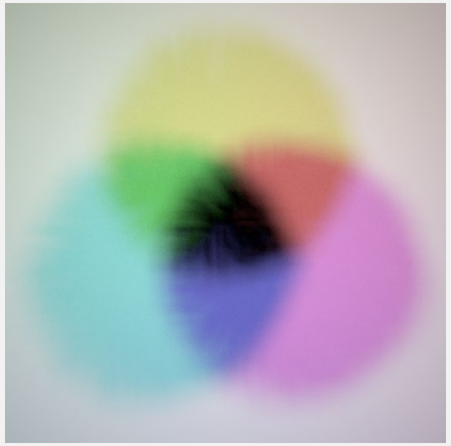
Original image



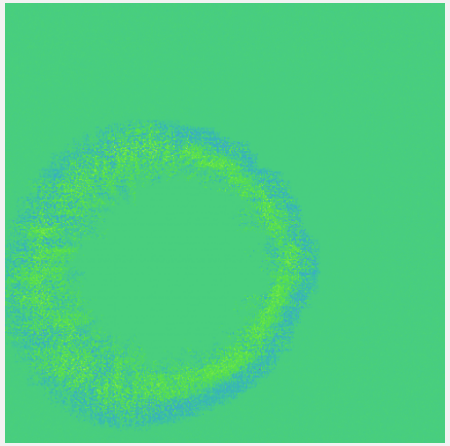
Derivative image



Original image

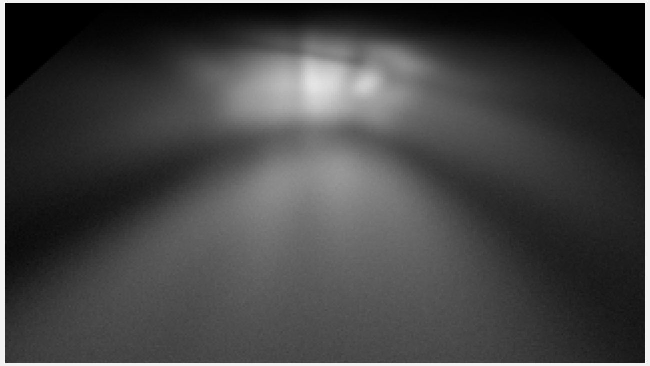


Derivative image

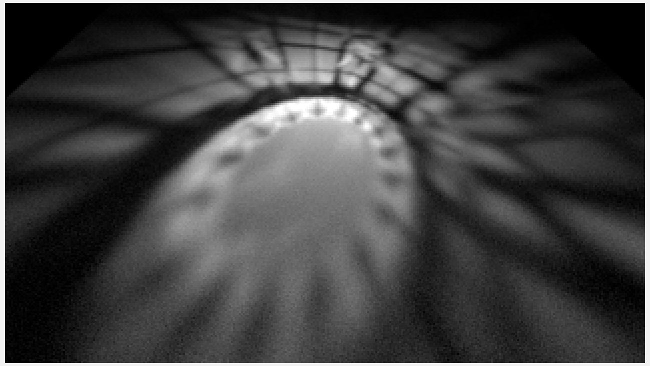


Config.

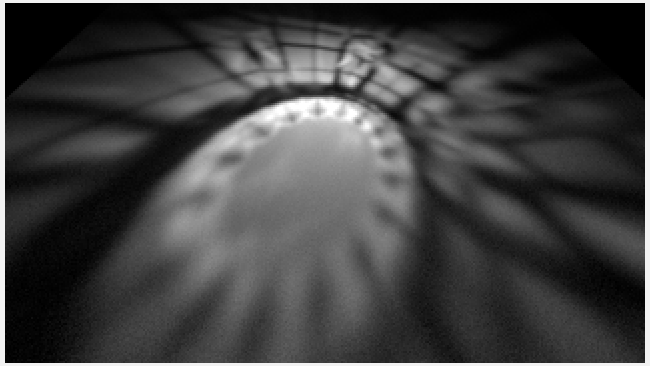
Optimize (initial)



Optimize (final)



Target



# Stuff we are missing

We need path sampling algorithms tailored to differentiable rendering:

- Some simple versions exist for local differentiation (Gkioulekas et al. 2013, 2016).
- We need to take into account diff. geometric quantities in global case.
- We need to take into account loss function.

We need theory that can handle very low-dimensional path manifolds:

- We can’t easily incorporate specular and refractive effects into arbitrary pipelines.
- Doable in isolation (Chen and Arvo 2000, Jakob and Marschner 2013, Xin et al. 2019).

# Some more general thoughts

Initialization is <u>super</u> important:

- Approximate reconstruction assuming direct lighting is usually good enough.
- Coarse-to-fine schemes work well.

Parameterizations are <u>super</u> important:

- Loss functions very non-linear and change shape easily.
- Working with meshes is a pain (topology is awful and not (easily?) differentiable).

# Parameterization matters

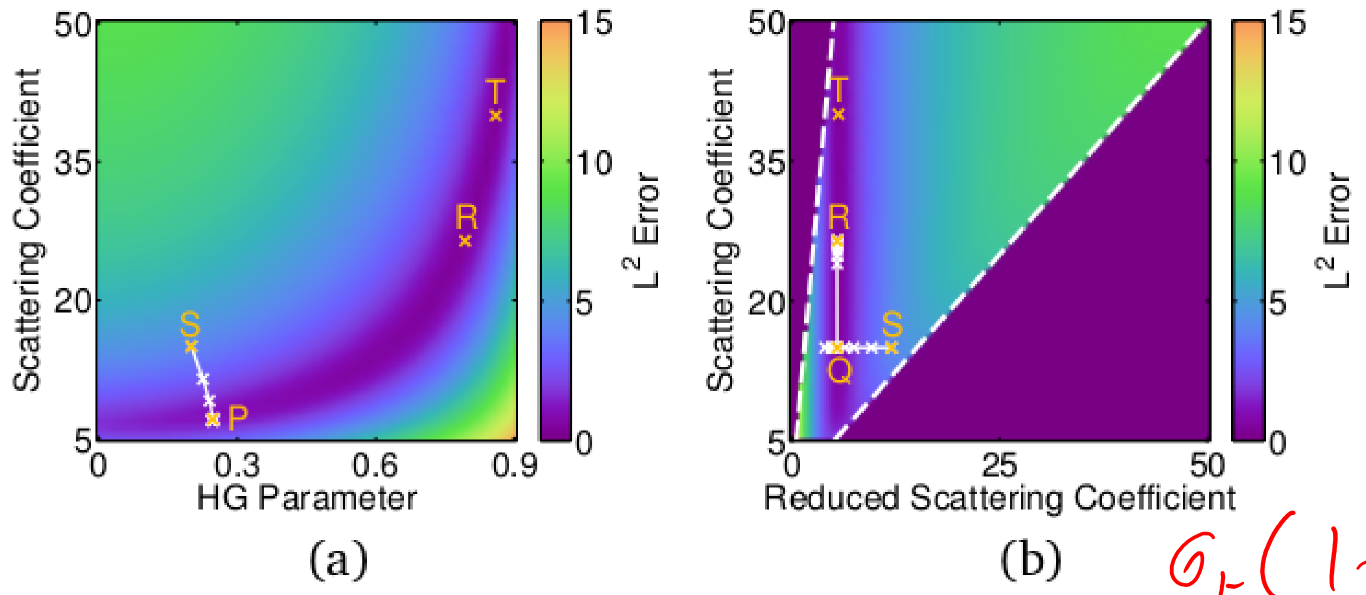



**Figure 4:** *Search spaces for an inverse rendering problem: (a) the original space; (b) the reparameterized space. The plotted region in (a) maps to the area enclosed by the dashed lines in (b). Using the original space, the stochastic gradient descent (SGD) algorithm starting from point S is trapped at point P, which is far from the real solution T. Using the reparameterized space, the algorithm is able to find point R that is much closer to the real solution.*

$\sigma_t(1-g)$

volumetric density $\sigma_t$

scattering albedo $a$

phase function $f_r$

# Some more general thoughts

Initialization is super important:

- Approximate reconstruction assuming direct lighting is usually good enough.
- Coarse-to-fine schemes work well.

Parameterizations are super important:

- Loss functions very non-linear and change shape easily.
- Working with meshes is a pain (topology is awful and not (easily?) differentiable).

You don’t always need Monte Carlo differentiable rendering:

- If you don’t have strong global illumination, just use direct lighting.
- A lot of research in computer vision on differentiable rasterizers.

Remember that you are doing optimization:

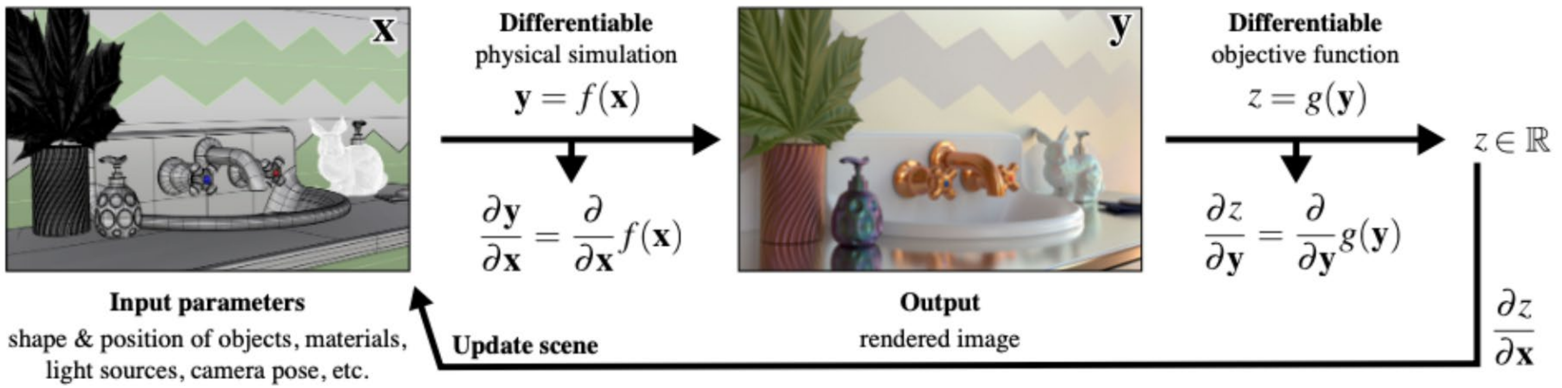
- Unbiased and consistent gradients are very expensive to compute.
- Biased and/or inconsistent gradients can be very cheap to compute.
- Often, biased and/or inconsistent gradients are enough for convergence.
- Stochastic gradient descent matters a lot.

# Reference material

**Physics-Based Differentiable Rendering**
**A Comprehensive Introduction**


Shuang Zhao[1], Wenzel Jakob[2], and Tzu-Mao Li[3]
[1]University of California, Irvine [2]EPFL [3]MIT CSAIL


SIGGRAPH 2020 Course




# CVPR 2021 Tutorial Proposal

**Title:** Tutorial on Physics-Based Differentiable Rendering

**Proposers' Names, Titles, Affiliations, and Primary Contact Emails:**


Shuang Zhao
Assistant Professor, CS
University of California, Irvine
shz@ics.uci.edu

Ioannis Gkioulekas
Assistant Professor, RI
Carnegie Mellon University
igkioule@cs.cmu.edu