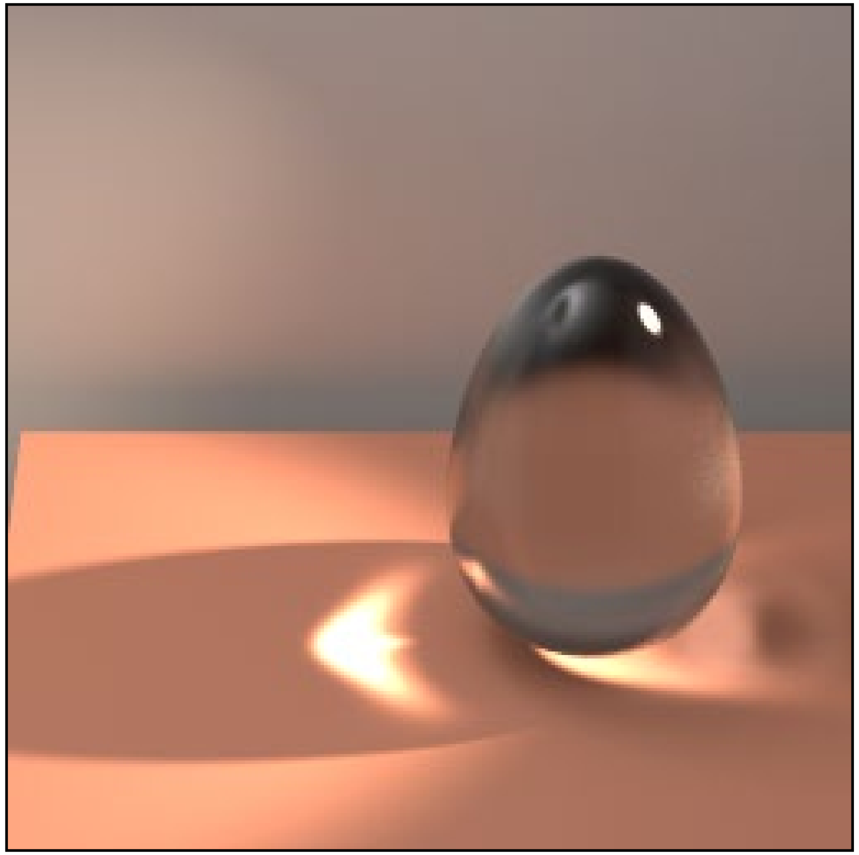
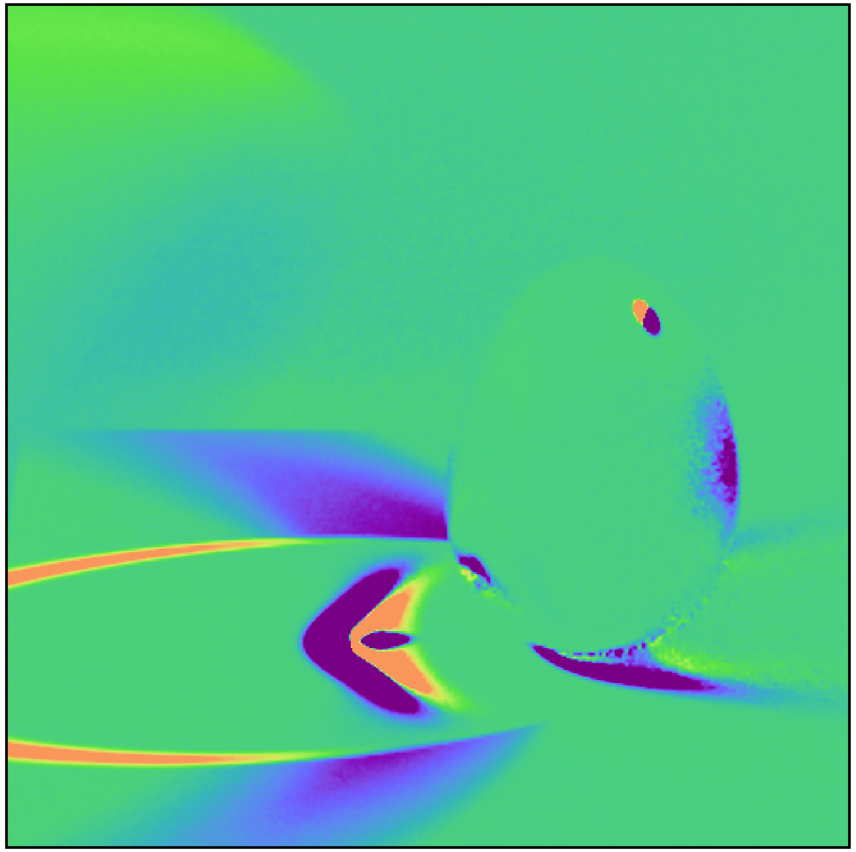
Inverse and differentiable rendering





15-468, 15-668, 15-868
Physics-based Rendering
Spring 2025, Lecture 7

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

Course announcements

- Feedback for all proposals on Canvas.
- Extra recitation today, 3:30 – 5 pm.

Overview of today's lecture

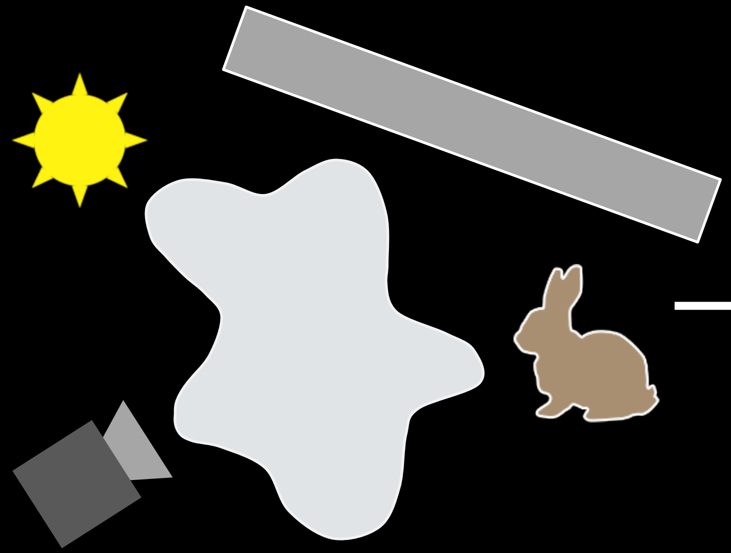
- Inverse rendering.
- Differentiable rendering.
- Differentiating local parameters.
- Differentiating global parameters.
- Path-space differentiable rendering.
- Reparameterizations.

Slide credits

Many of these slides were directly adapted from:

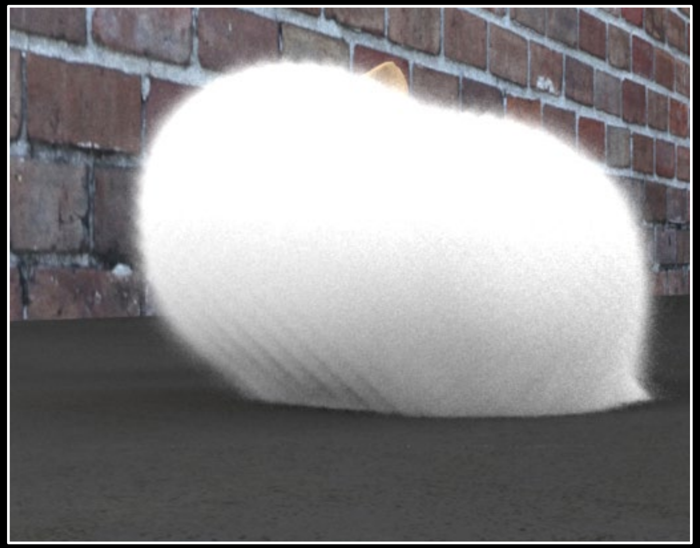
- Shuang Zhao (UC Irvine).
- Tzu-Mao Li (UCSD).
- Sai Praveen Bangaru (MIT).

Forward rendering



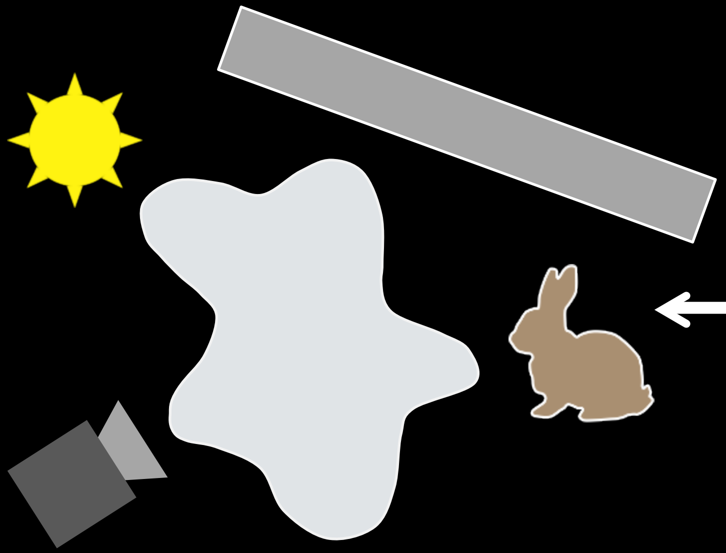
digital scene specification
(geometry, materials,
optics, light sources)

physically-accurate
rendering

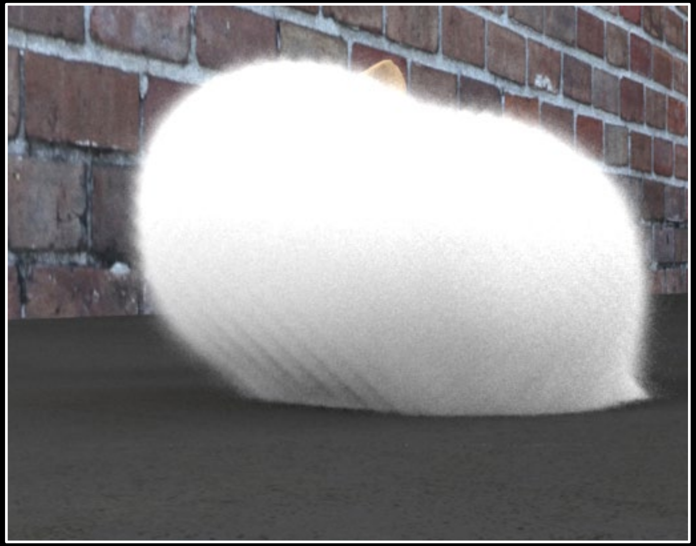


photorealistic
simulated image

Inverse rendering



physically-accurate
inverse rendering



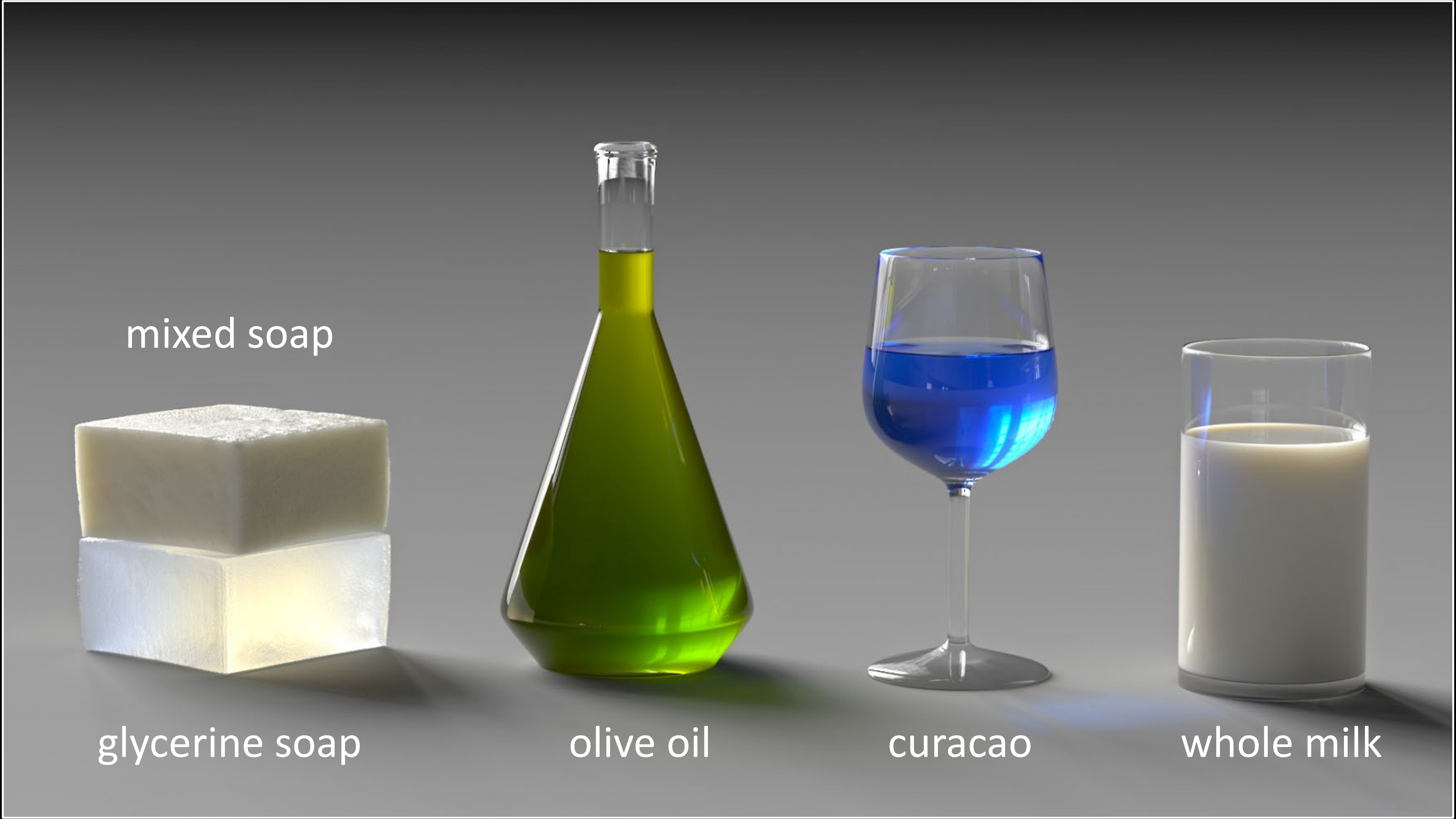
digital scene specification
(geometry, materials,
camera, light sources)

photorealistic
synthetic image

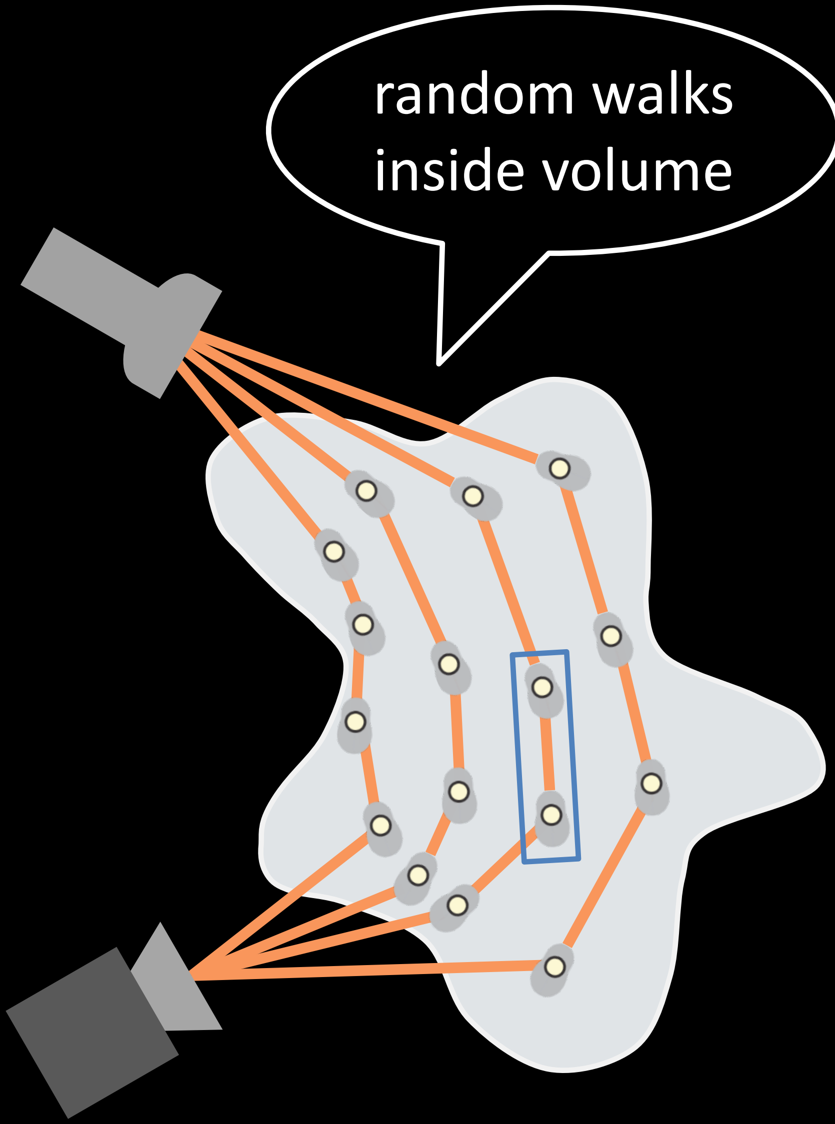
image
measurement

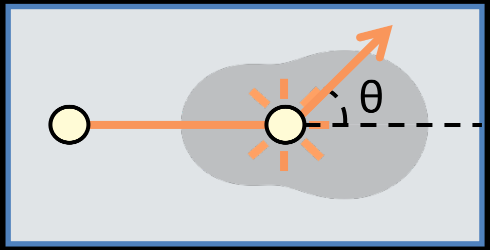
What I was doing in 2013

I wanted to make images such as this one



Scattering: extremely multi-path transport





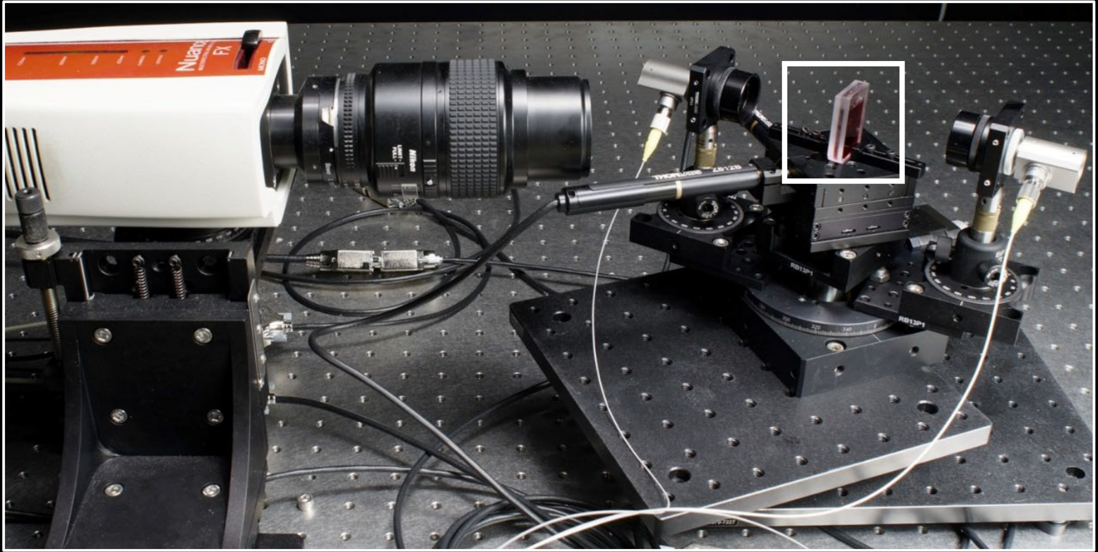
volumetric density σ_t

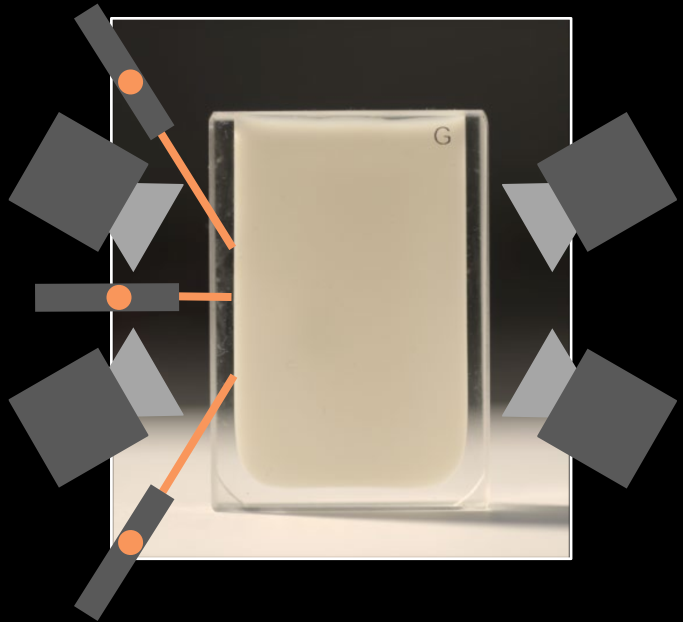
scattering albedo a

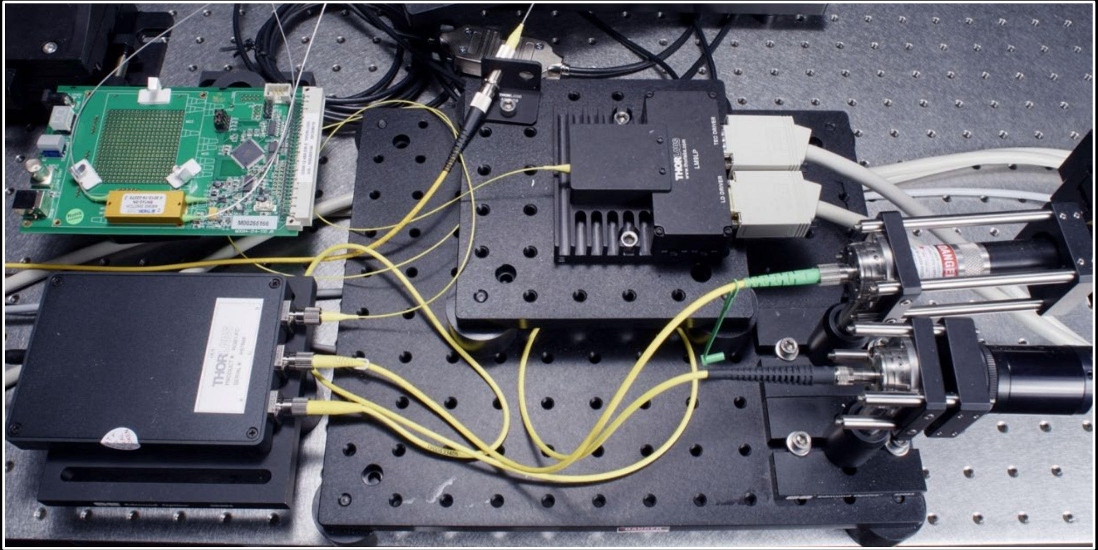
material =

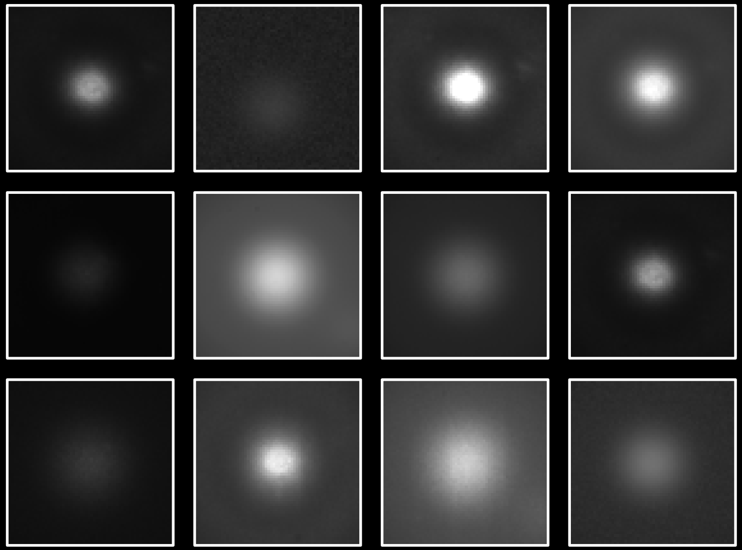
phase function f_r

Acquisition setup





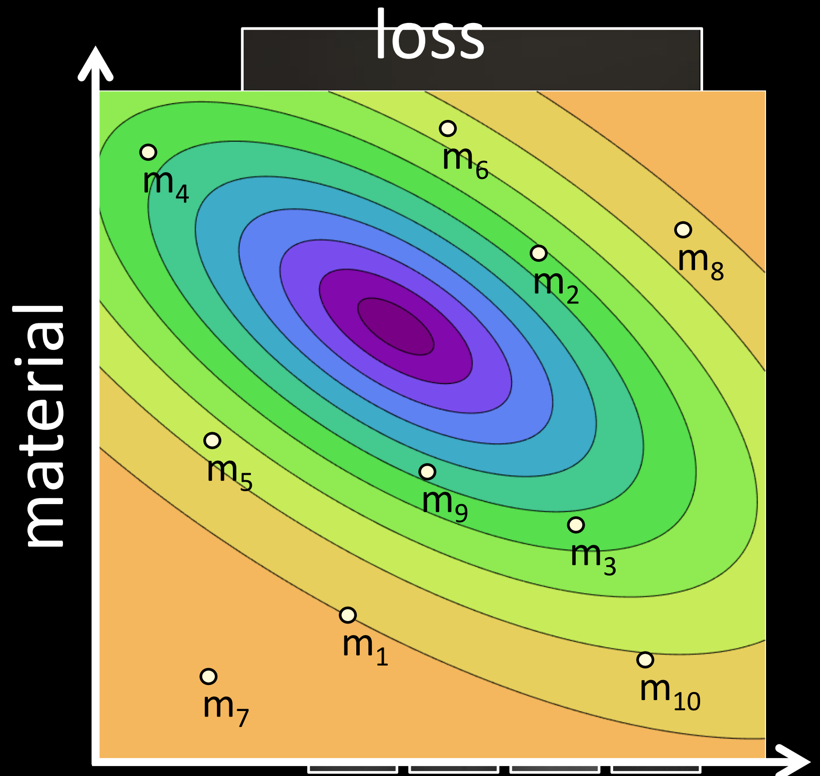




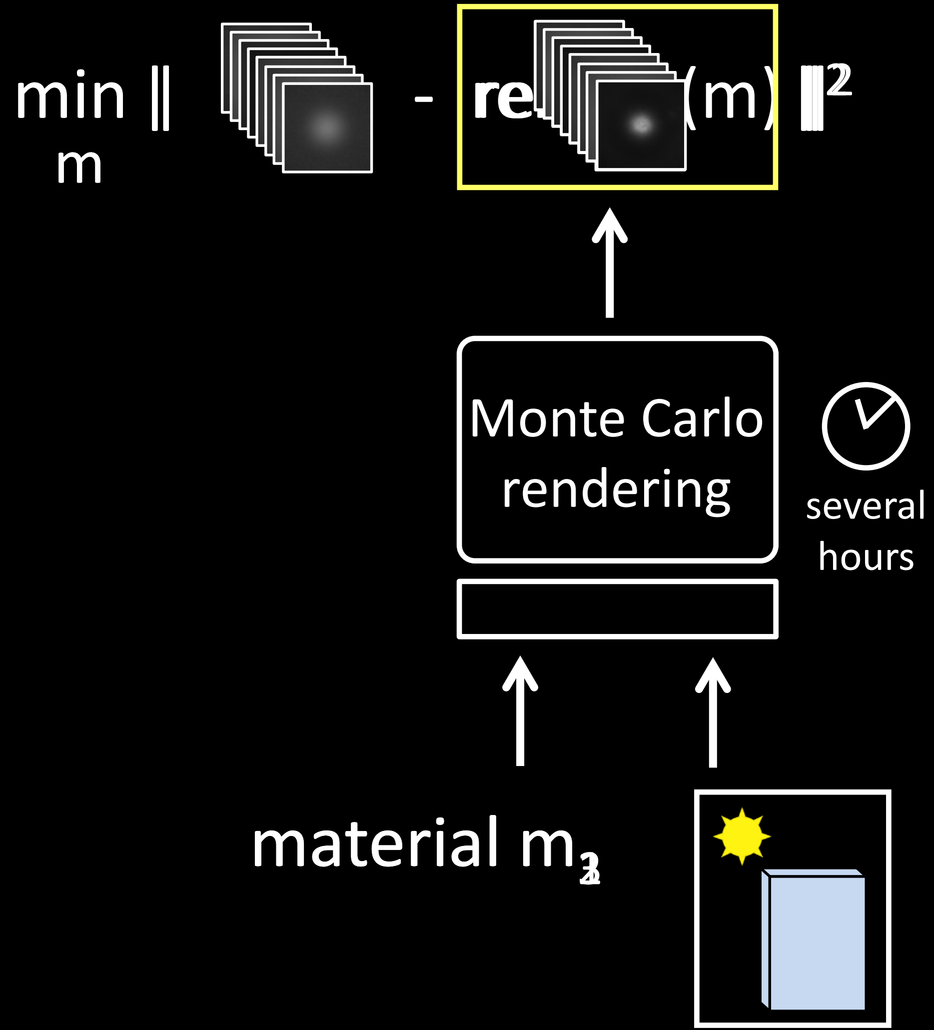
Analysis by synthesis (a.k.a. inverse rendering)

not scalable

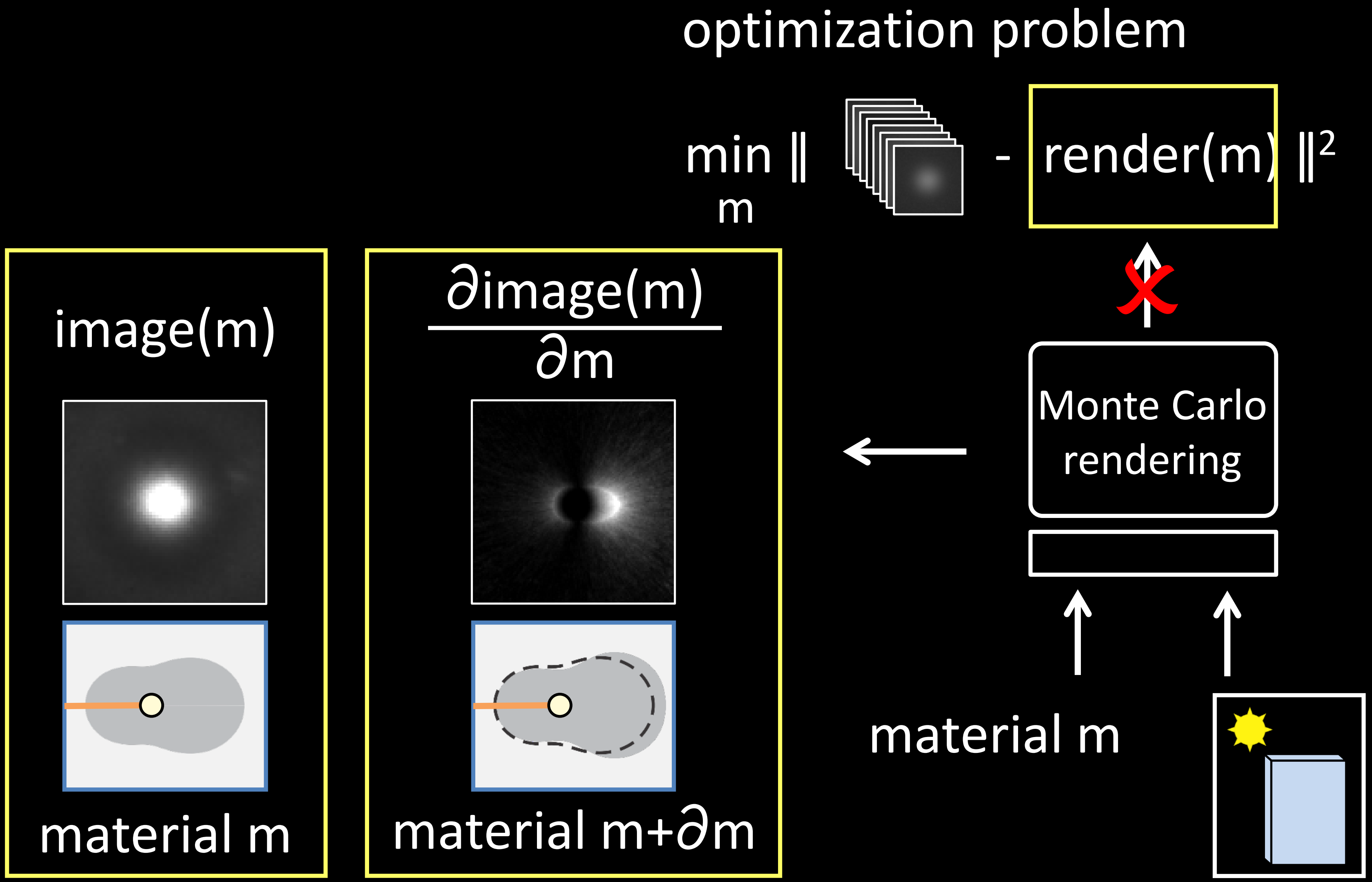
solve by exhaustive search?



optimization problem



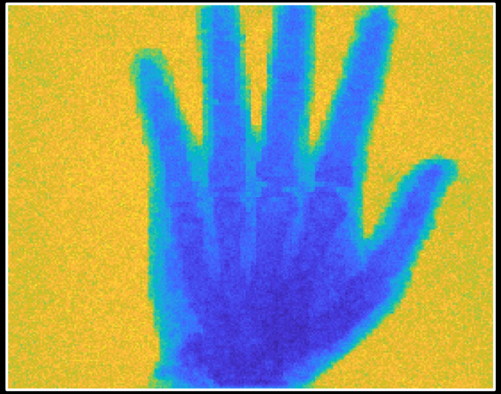
Analysis by synthesis (a.k.a. inverse rendering)



Other scattering materials

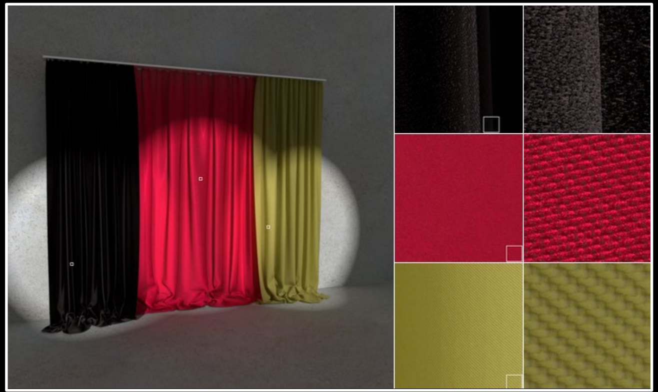
everyday materials
[Gkioulekas et al. 2013]

industrial dispersions
[Gkioulekas et al. 2013]



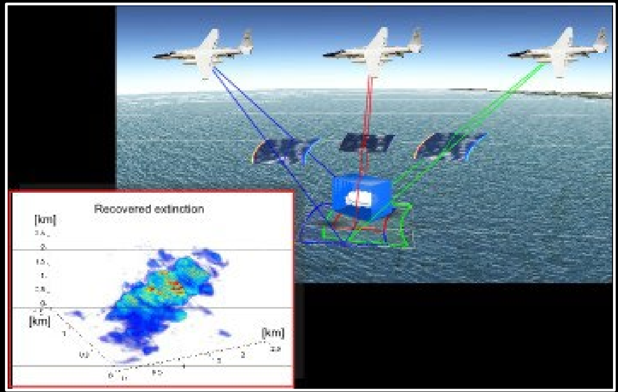
computed tomography
[Geva et al. 2018]

optical
tomography
[Gkioulekas et al.
2016]



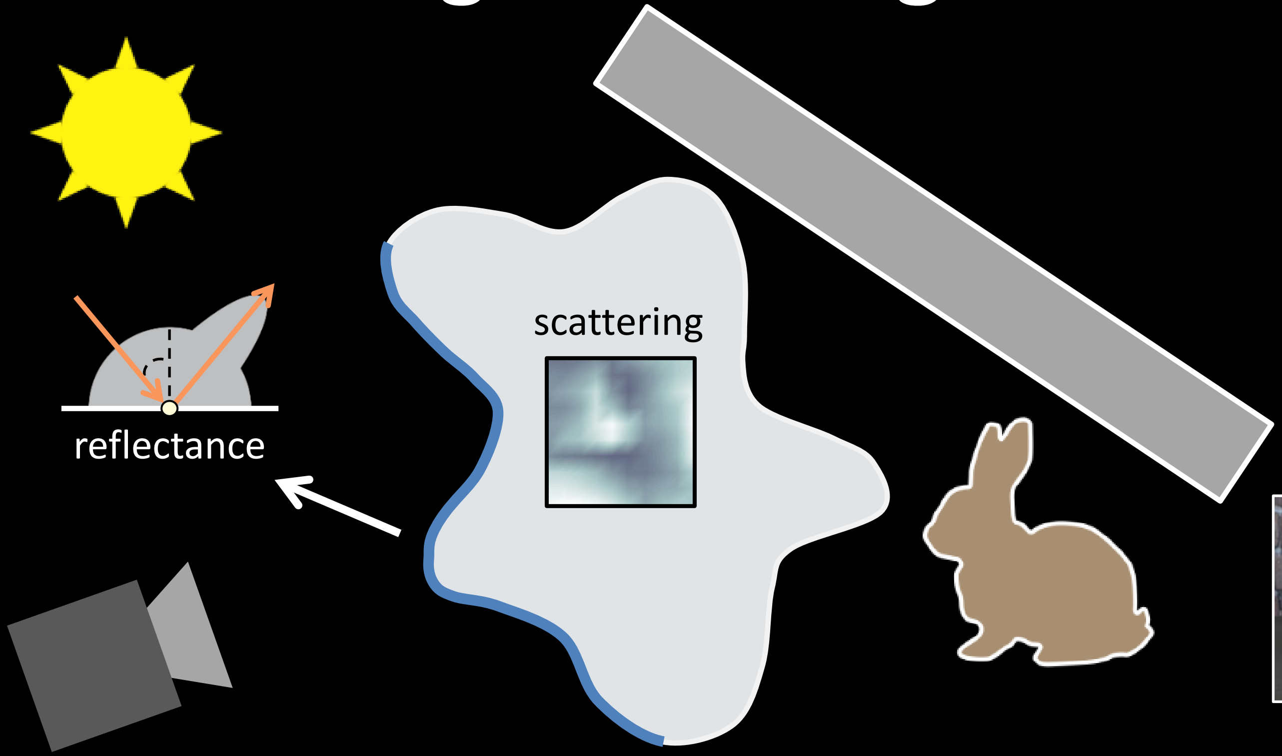
woven fabrics
[Khungurn et al. 2015,
Zhao et al. 2016]

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



clouds
[Levis et al. 2015, 2017]

Making sense of global illumination

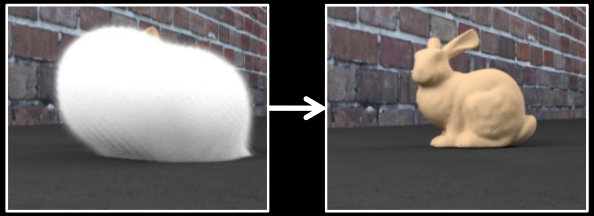


X: 3D shape

X: surface reflectance

X: occluded imaging

X: illumination



analysis by synthesis

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

stochastic gradient descent

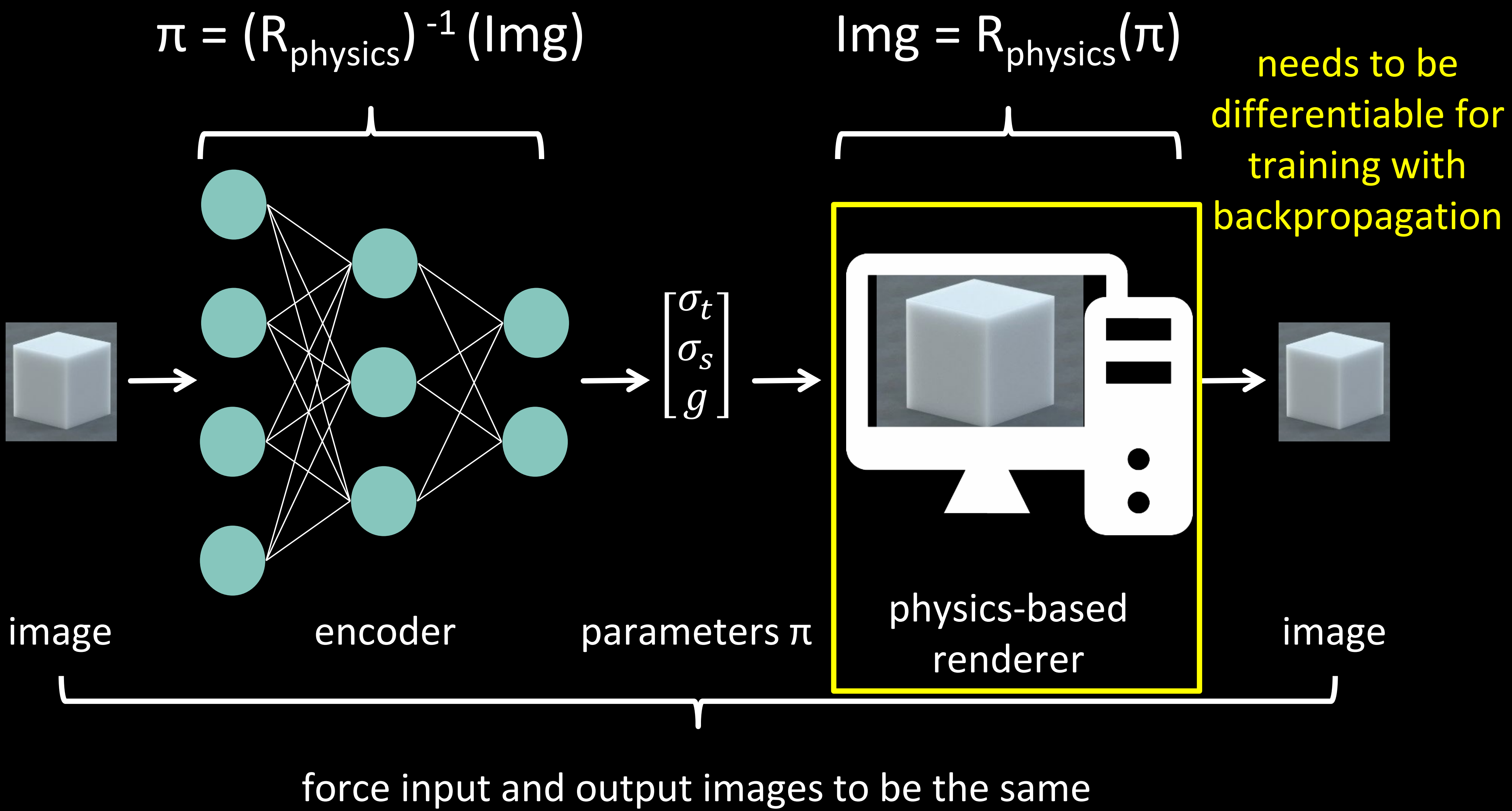
while (not converged)

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

Monte-Carlo rendering

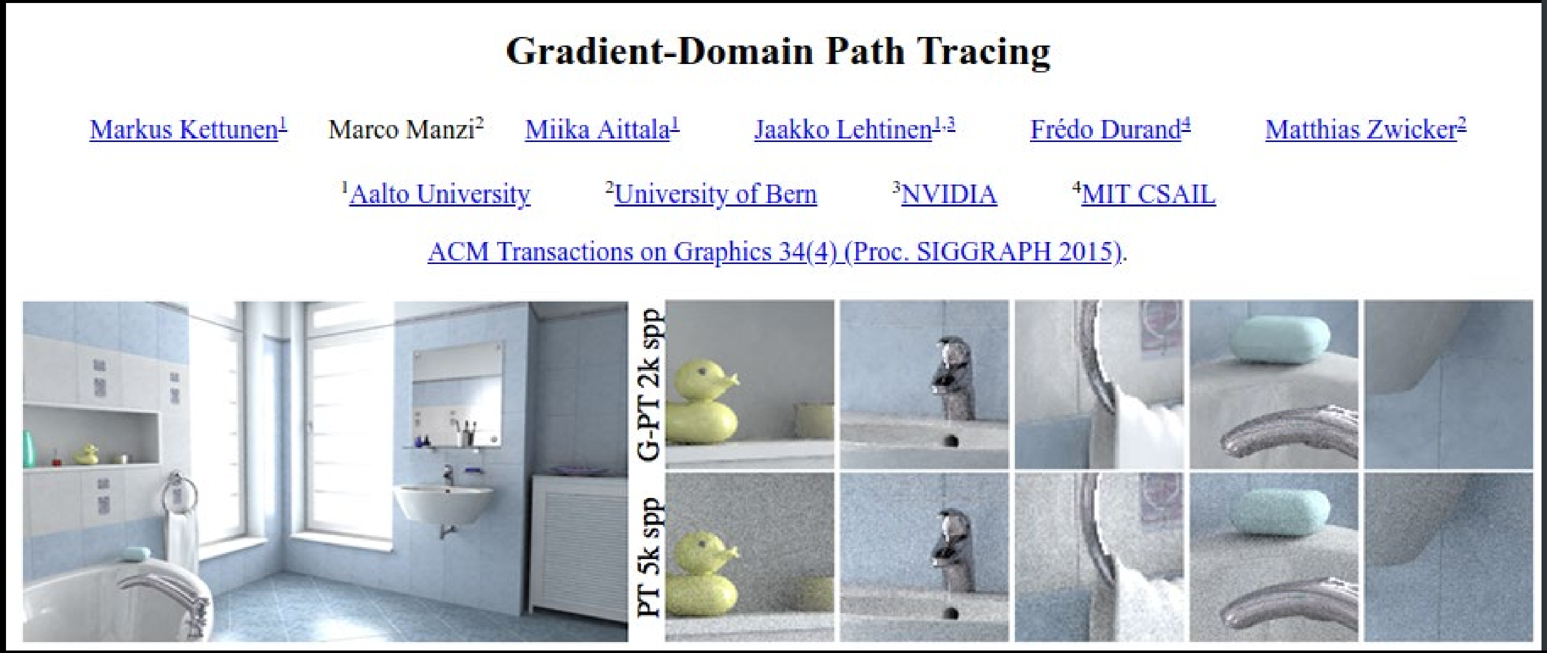
differentiable rendering: image gradients with respect to arbitrary X

Differentiable rendering and deep learning



Differentiable rendering

Not related to:



“Gradient” in their case refers to image edges.

REMINDER (?) FROM CALCULUS

Reminder from calculus

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

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

Move derivative inside integral

Account for changes in integration limits

$$+\ f(b(\pi),\pi)\frac{\mathrm{d}b(\pi)}{\mathrm{d}\pi} - f(a(\pi);\pi)\frac{\mathrm{d}a(\pi)}{\mathrm{d}\pi}$$

Account for discontinuities of integrand that depend on π

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

A simple example

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

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

Move derivative inside integral

Account for changes in integration limits

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

Account for discontinuities of integrand that depend on π

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

Leibniz integral rule

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

Interior integral

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

Move derivative inside integral

Boundary terms

Account for changes in integration limits

$$+\ f(b(\pi),\pi)\frac{\mathrm{d}b(\pi)}{\mathrm{d}\pi} - f(a(\pi);\pi)\frac{\mathrm{d}a(\pi)}{\mathrm{d}\pi}$$

Account for discontinuities of integrand that depend on π

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

Simplified Leibniz integral rule

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

Interior integral

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

Move derivative inside integral

Boundary terms

~~Account for changes in integration limits~~

$$\cancel{+ f(b(\pi);\pi)\frac{\mathrm{d}b(\pi)}{\mathrm{d}\pi} - f(a(\pi);\pi)\frac{\mathrm{d}a(\pi)}{\mathrm{d}\pi}}$$

Account for discontinuities of integrand that depend on π

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

Differentiation wrt $\boldsymbol{\pi}$ still includes the discontinuity terms!

Simplified Leibniz integral rule

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

Interior integral

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

Move derivative inside integral

Boundary terms

Account for changes in integration limits

$$+ \; f(b(\pi),\pi)\frac{\mathrm{d}b(\pi)}{\mathrm{d}\pi} - f(a(\pi),\pi)\frac{\mathrm{d}a(\pi)}{\mathrm{d}\pi}$$

Account for discontinuities of integrand that depend on π

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

Differentiation wrt $\boldsymbol{\pi}$ simplifies to just moving derivative inside integral when:

- Integration limits are independent of $\boldsymbol{\pi}$.
- Integrand discontinuities are independent of $\boldsymbol{\pi}$.

We need *both* conditions!

Reynolds transport theorem

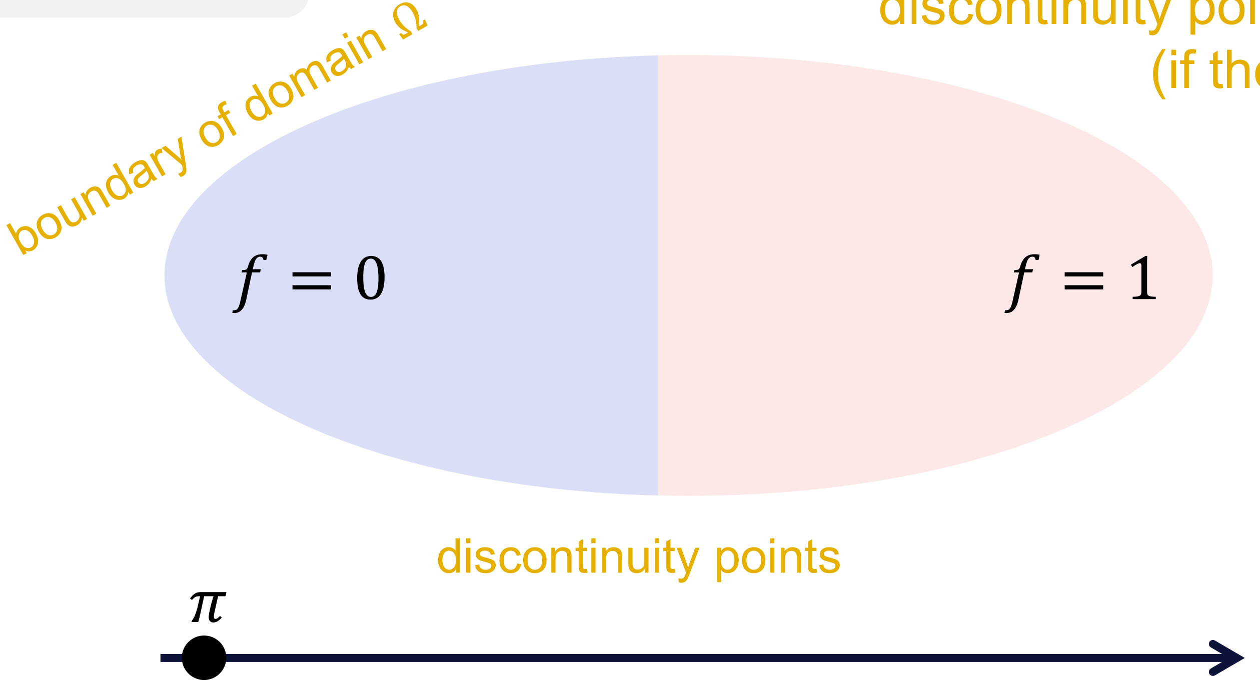
$$\frac{\mathrm{d}}{\mathrm{d}\pi}\int_{\Omega(\pi)} f(x,\pi)\mathrm{d}A(x) \overset{?}{=} \int_{\Omega(\pi)} \frac{\mathrm{d}f(x,\pi)}{\mathrm{d}\pi}\mathrm{d}A(x) + \int_{\partial\Omega(\pi)} g(x,\pi)\mathrm{d}l(x)$$

Reynolds transport theorem [1903]
Generalization of the Leibniz rule

Interior integral

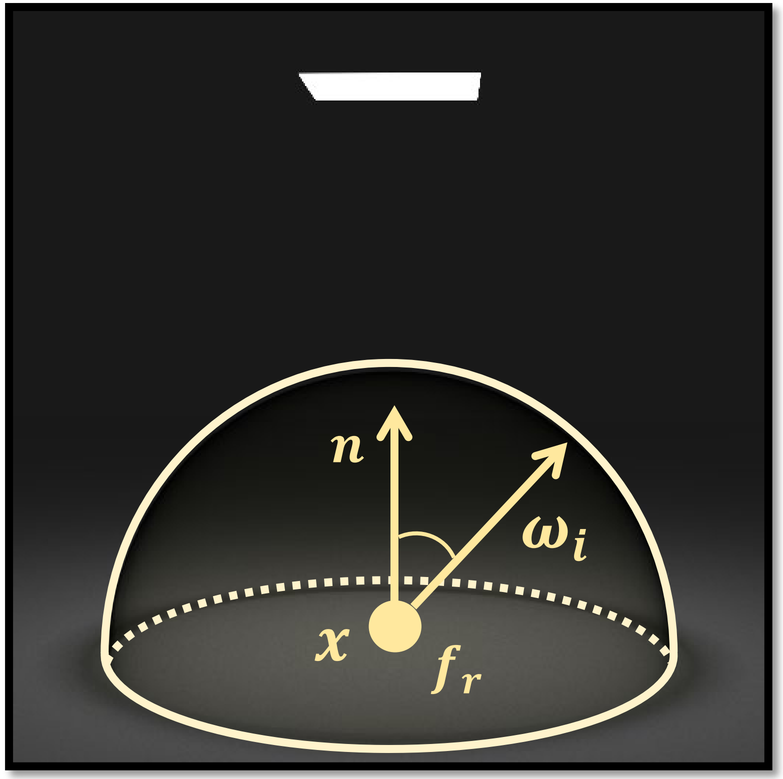
Boundary integral

Boundary domain
||
discontinuity points ∪ boundary of domain Ω
(if they depend on π)



DIFFERENTIATING DIRECT ILLUMINATION

Direct illumination integral



Radiance from x:

Reflectance (BRDF) · Incident radiance · Shading wrt normal $\boldsymbol{n}$

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

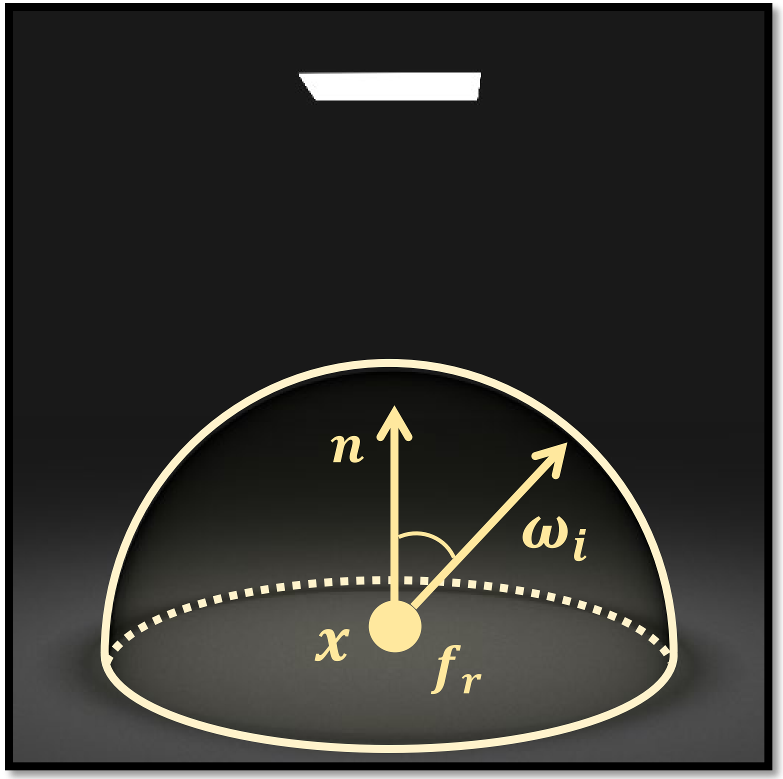
Unit hemisphere

Monte Carlo rendering:

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

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

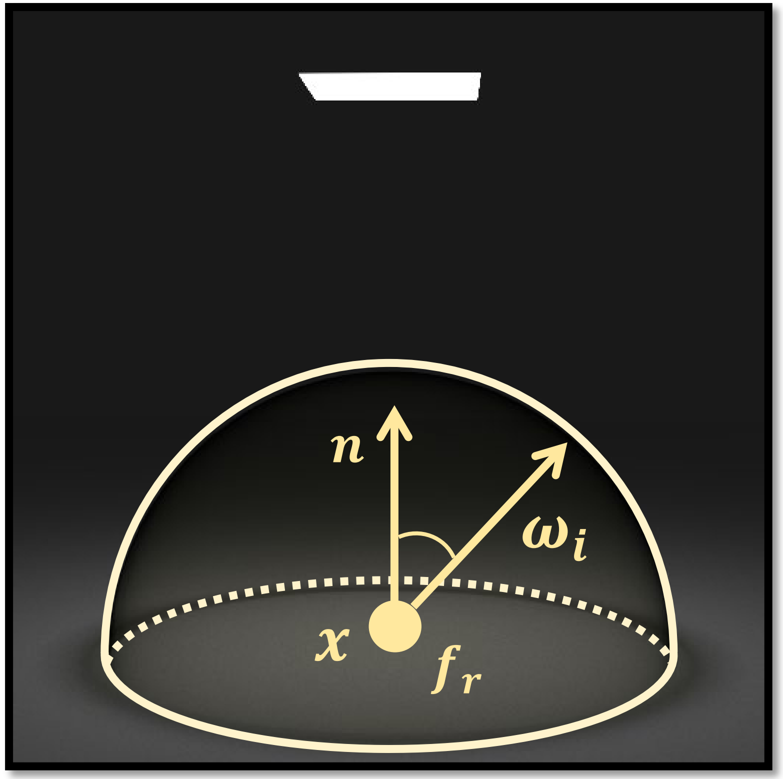
Differential direct illumination



Differential radiance from x:

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

Differential direct illumination: local parameters



π: *local* parameters

- BRDF parameters
- *shading* normal
- illumination brightness

Differential radiance from x:

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

Just move derivative inside integral

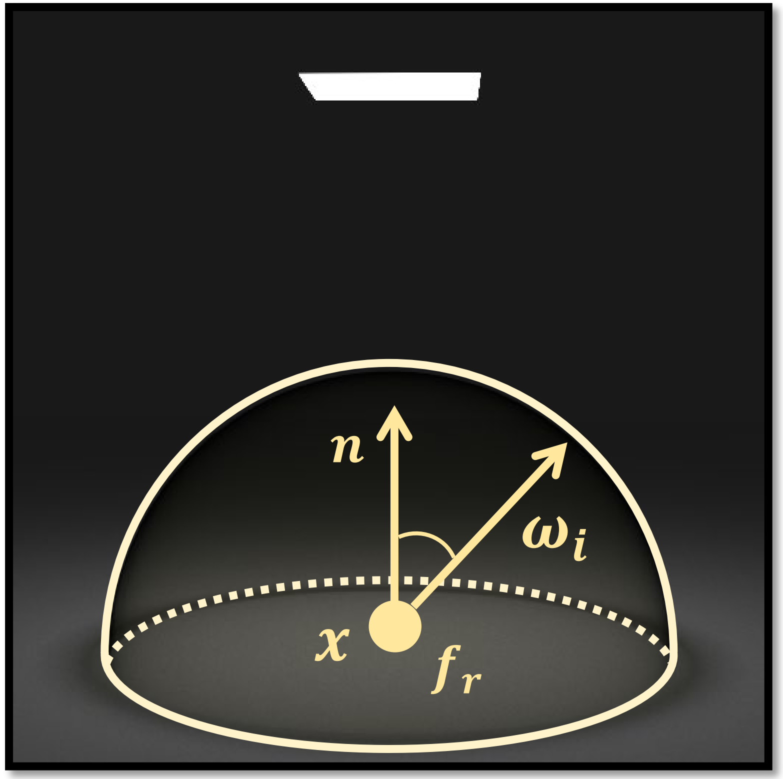
Monte Carlo differentiable rendering:

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

Just differentiate numerator
[Khungurn et al. 2015, Gkioulekas et al. 2015]

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

Alternative estimator



π: *local* parameters

- BRDF parameters

Differential radiance from x:

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

Just move derivative inside integral

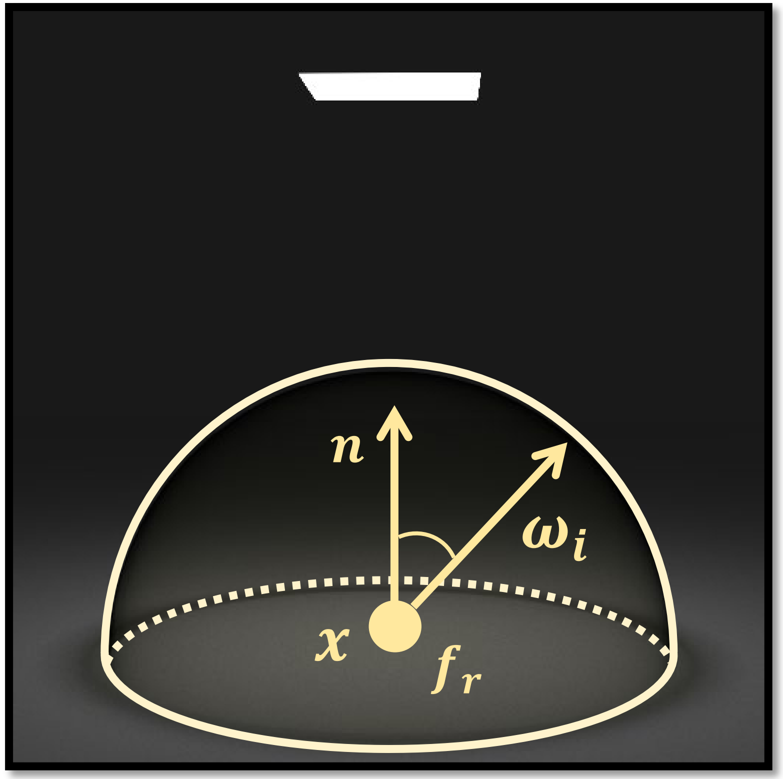
Monte Carlo estimation:

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

Differentiate entire contribution [Zeltner et al. 2021]

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

Differential direct illumination: global parameters



π: *global* parameters

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

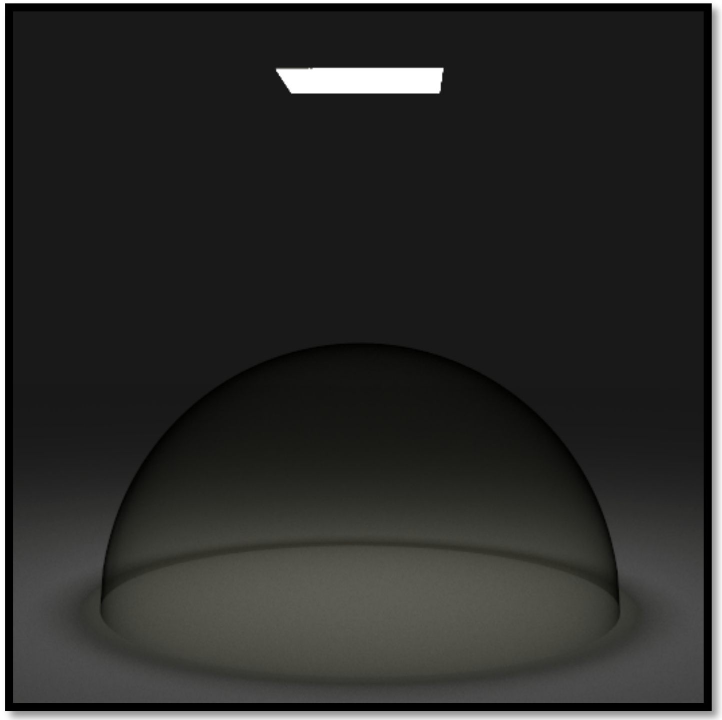
Differential radiance from x:

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

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

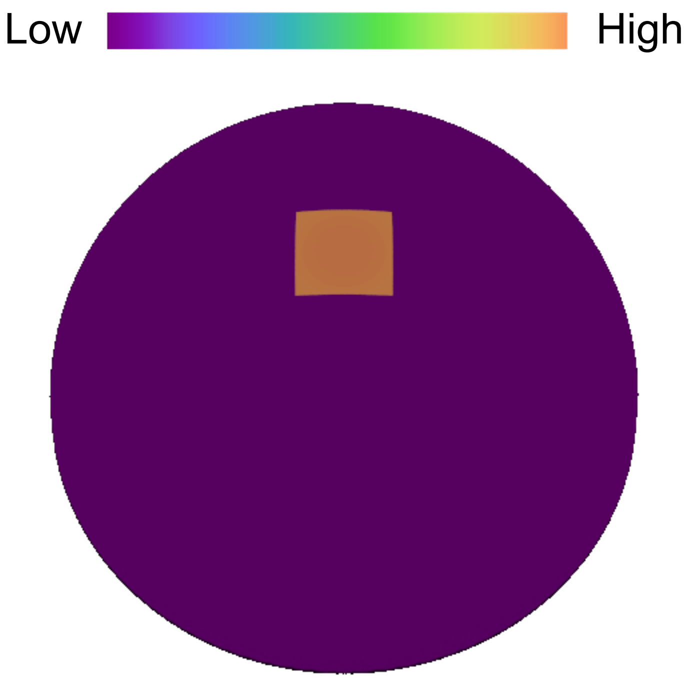
Need to use full Reynolds transport theorem

Discontinuities in the integrand

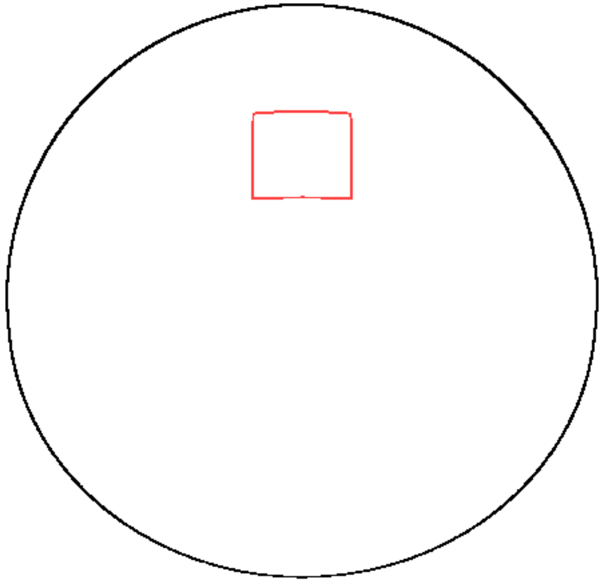


π: size of the emitter

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



Integrand
$f(\omega_i)$



Discontinuous points
(π-dependent)

Applying the Reynolds transport theorem

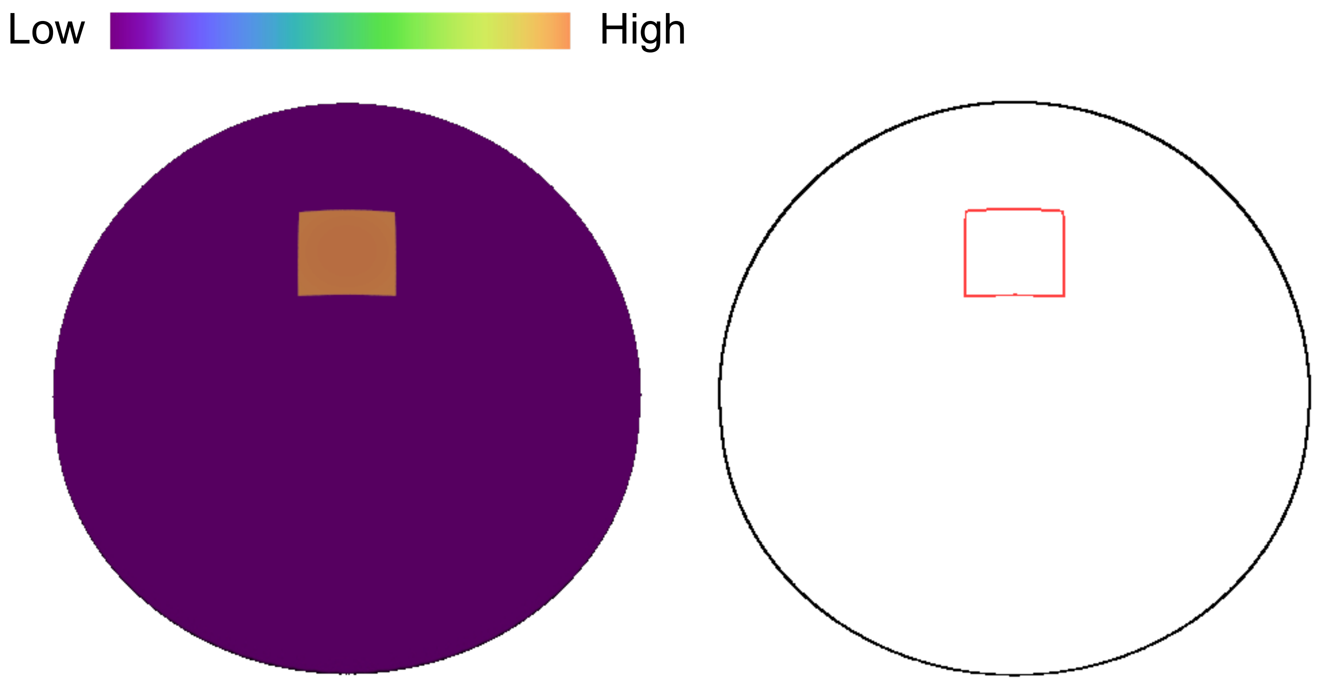
$$I = \int_{\mathbb{H}^2} f(\omega_i, \omega_o) \mathrm{d}\sigma(\omega_i)$$

$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\mathbb{H}^2} \frac{\mathrm{d}f}{\mathrm{d}\pi} \mathrm{d}\sigma + \int_{\partial\mathbb{H}^2} g \, \mathrm{d}l$$

Interior integral (same as for local parameters)

Boundary integral

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

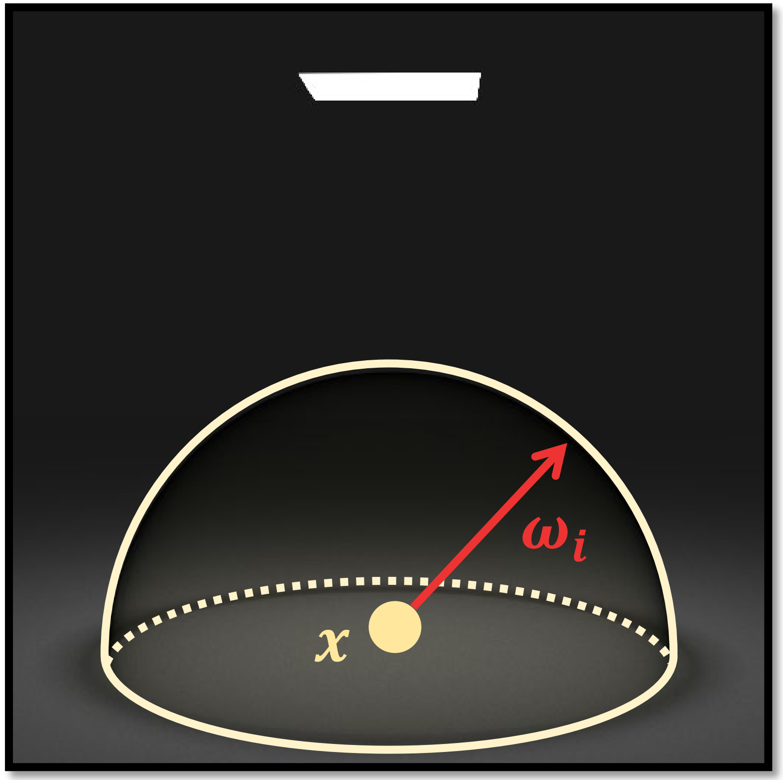


Integrand $f(\omega_i)$

Discontinuous points (π-dependent)

Reparameterizing the direct illumination integral

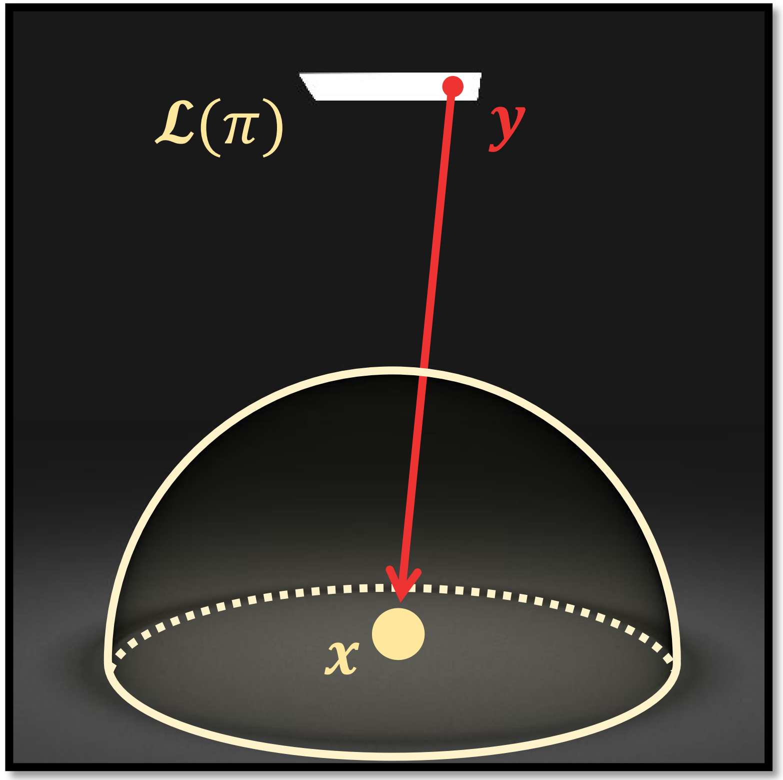
Hemispherical integral



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

Change of variables

Surface integral

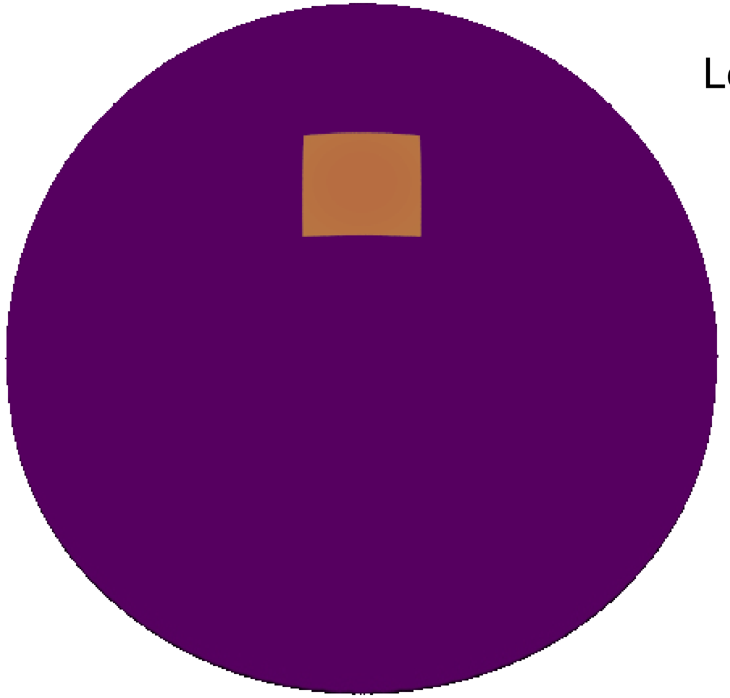


$$I = \int_{\mathcal{L}(\pi)} f(\boldsymbol{y} \to \boldsymbol{x})\, \boxed{G(\boldsymbol{x}, \boldsymbol{y})}\, \mathrm{d}A(\boldsymbol{y})$$

Includes visibility, fall-off, and foreshortening terms

Reparameterizing the direct illumination integral

Hemispherical integral



Change of variables

Surface integral

discontinuous

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

constant domain

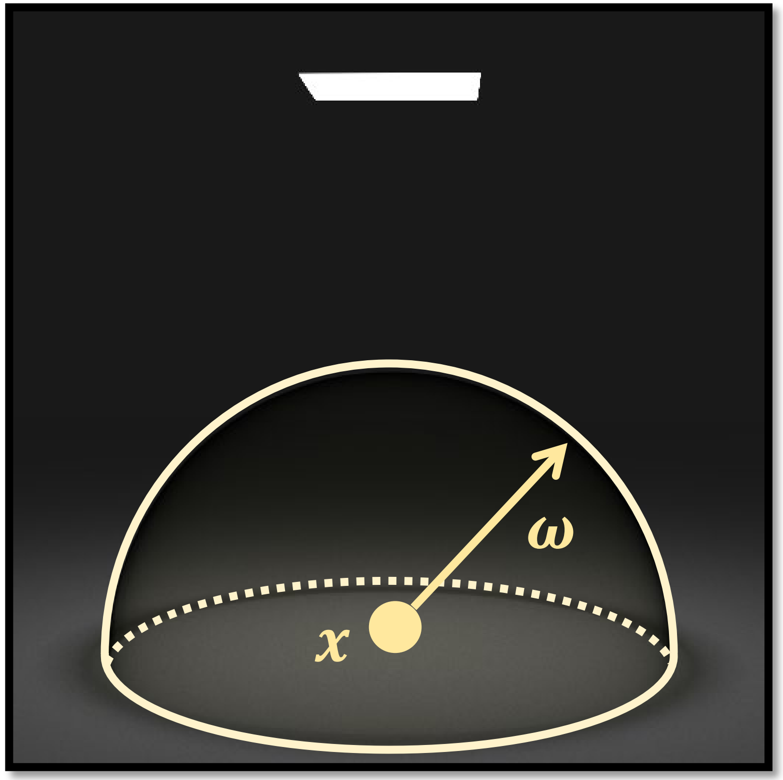
continuous

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

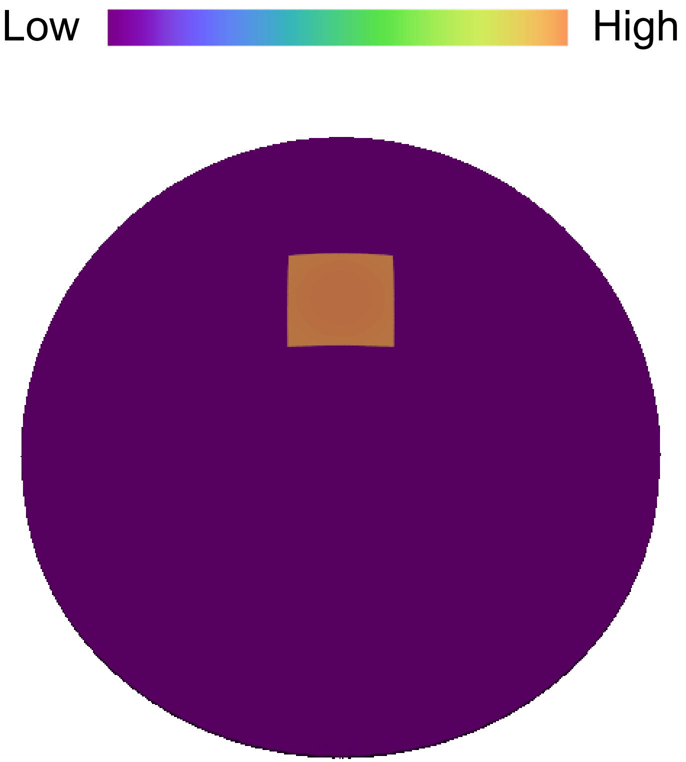
evolving domain

Differentiating the hemispherical integral

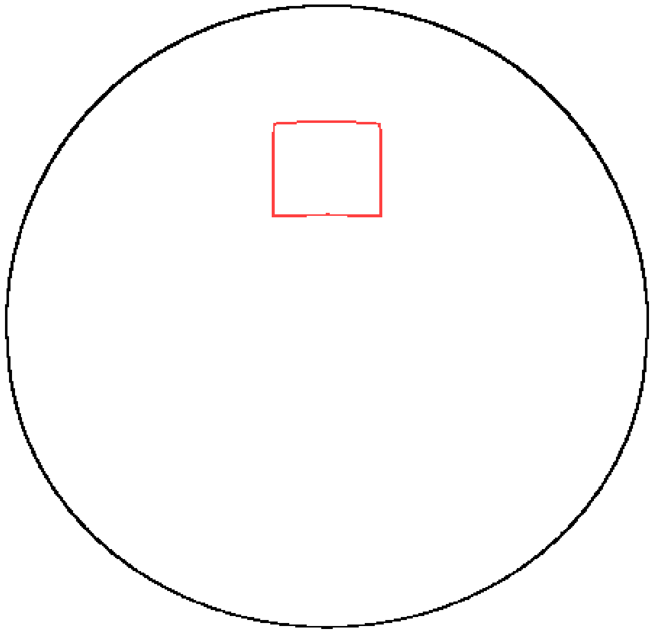
π: size of the emitter



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



Discontinuities of f



Differentiation

Reynolds transport theorem

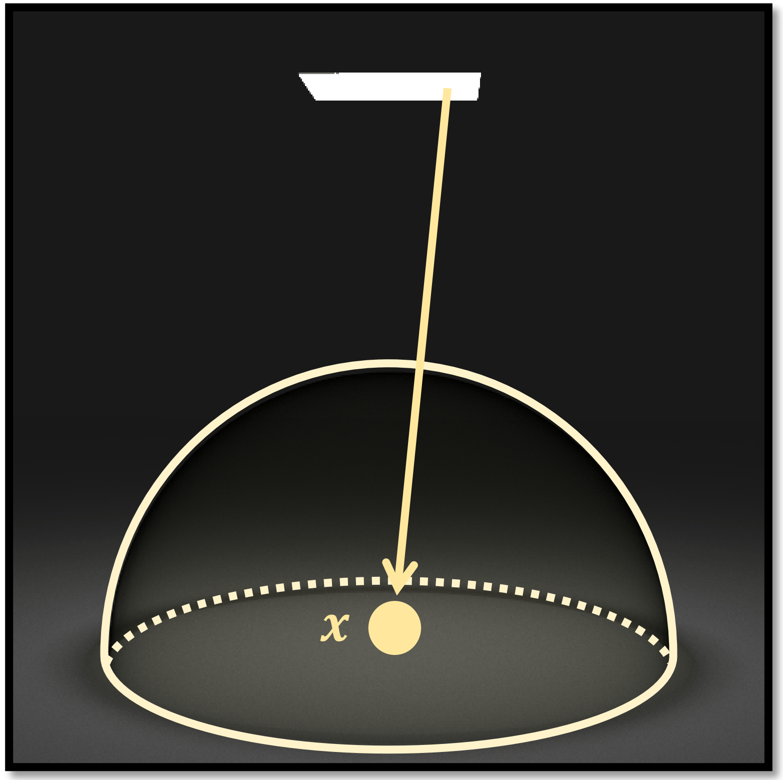
$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\mathbb{H}^2} \frac{\mathrm{d}(f)}{\mathrm{d}\pi}\,\mathrm{d}\sigma + \int_{\partial\mathbb{H}^2} g\,\mathrm{d}l$$

Interior

Boundary

Differentiating the area integral

π: size of the emitter



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

Low  High

Differentiation

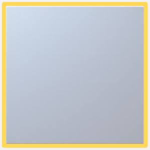
Reynolds transport theorem

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

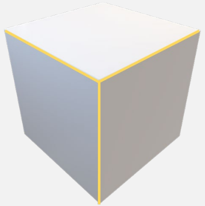
$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \underbrace{\int_{\mathcal{L}(\pi)} \frac{\mathrm{d}(fG)}{\mathrm{d}\pi} \mathrm{d}A}_{\text{Interior}} + \underbrace{\int_{\partial\mathcal{L}(\pi)} g \, \mathrm{d}l}_{\text{Boundary}}$$

Sources of discontinuities

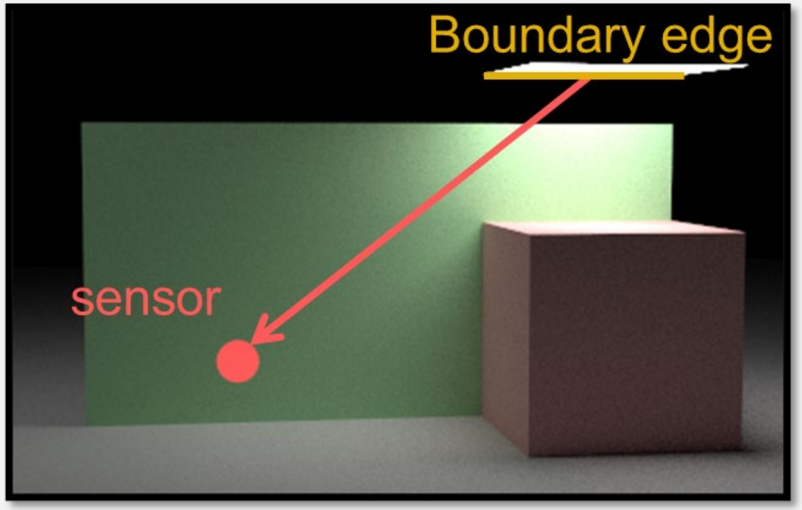
Boundary edge

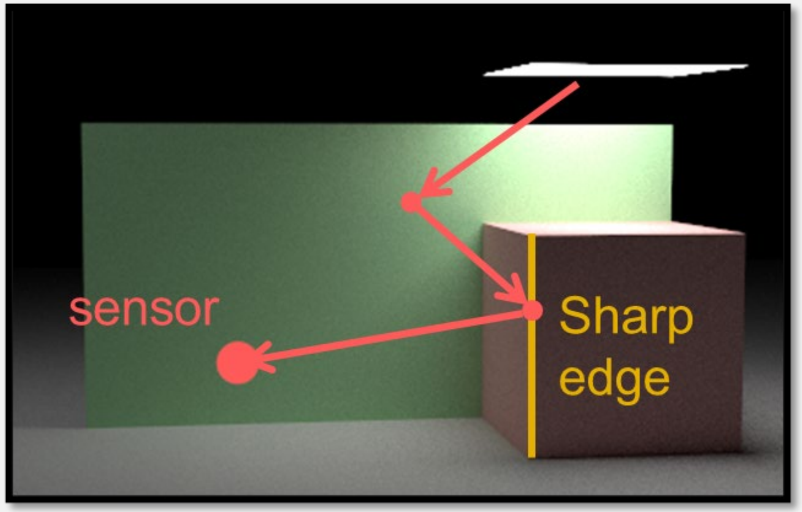


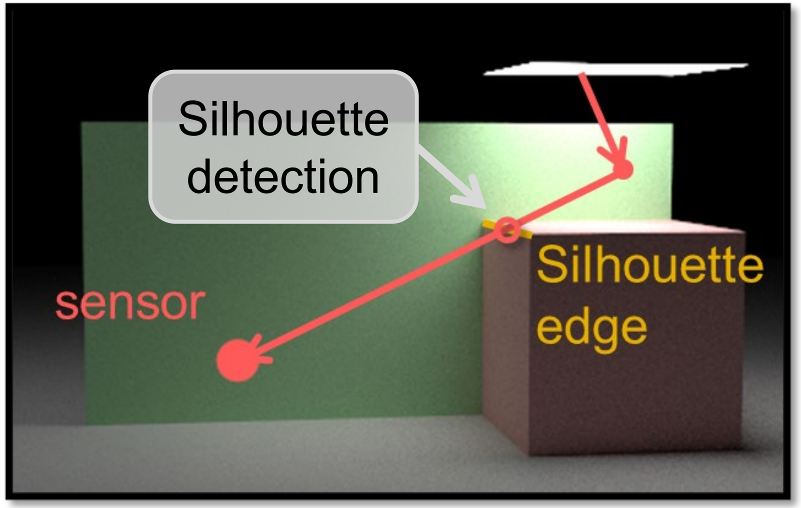
Sharp edge



Silhouette edge



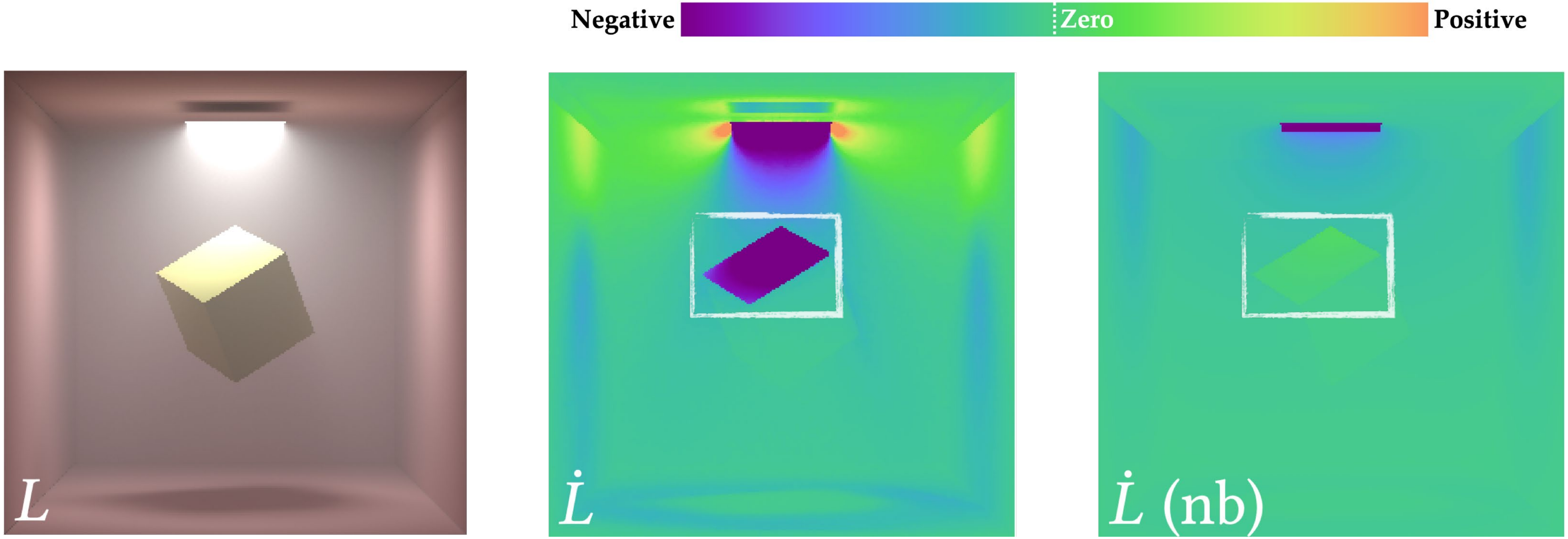




Topology-driven

Visibility-driven

Significance of the boundary integral



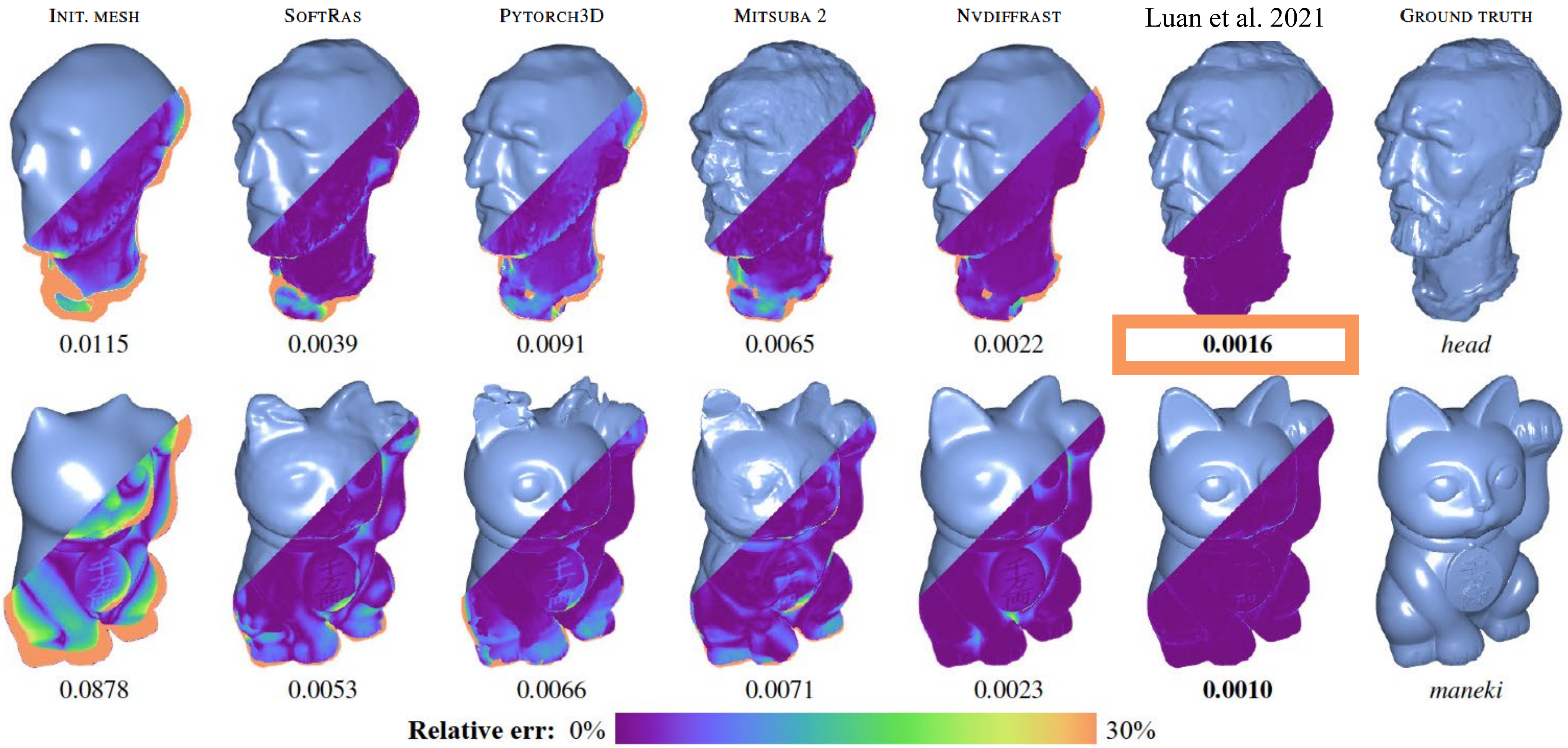
Original image

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

Derivative image
w/o boundary integral

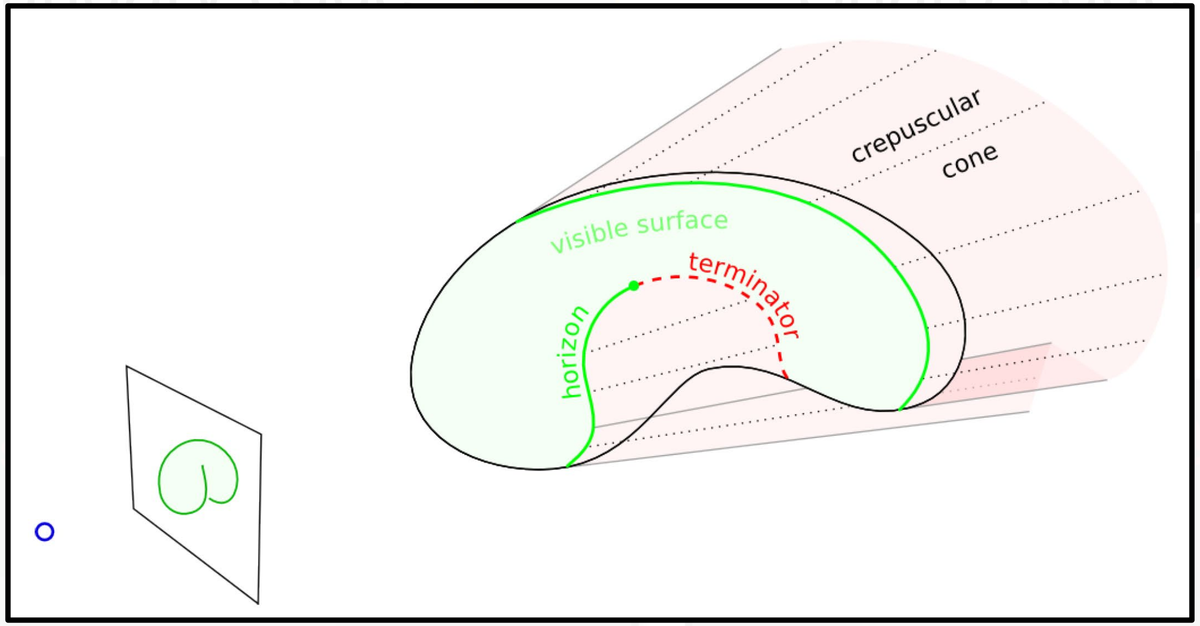
Gradient Accuracy Matters

Inverse-rendering results with *identical* optimization settings



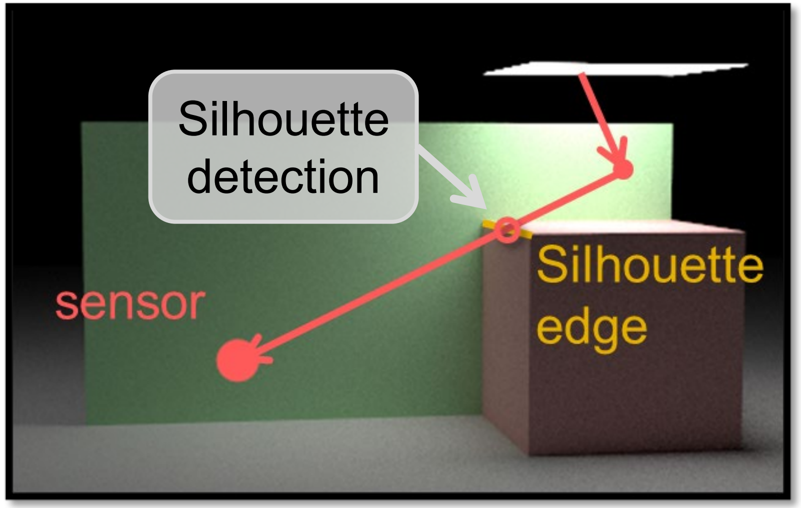
Sources of discontinuities

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



[Gargallo et al., ICCV 2007]

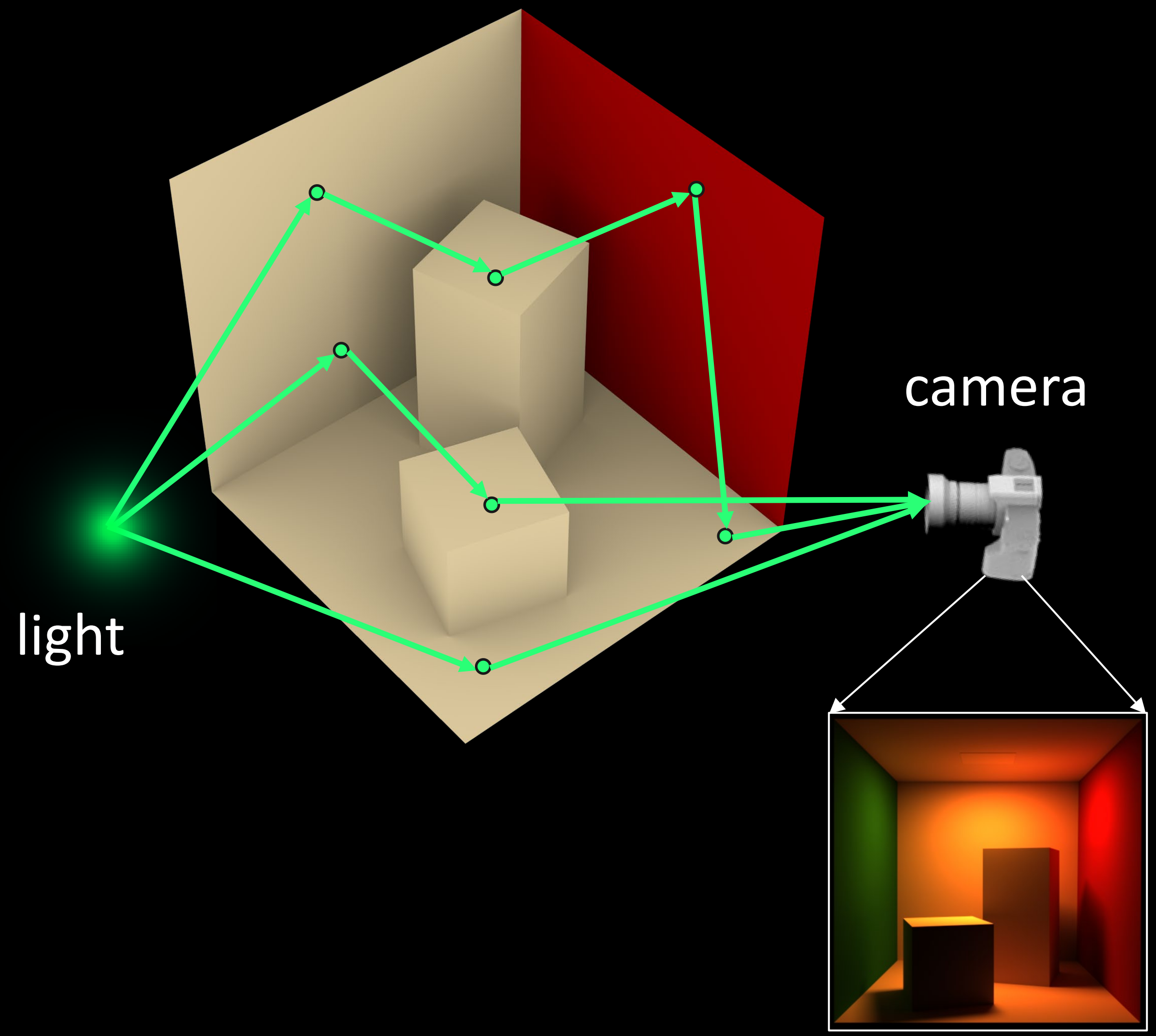
Silhouette edge



Visibility-driven

DIFFERENTIATING GLOBAL ILLUMINATION

Images as path integrals



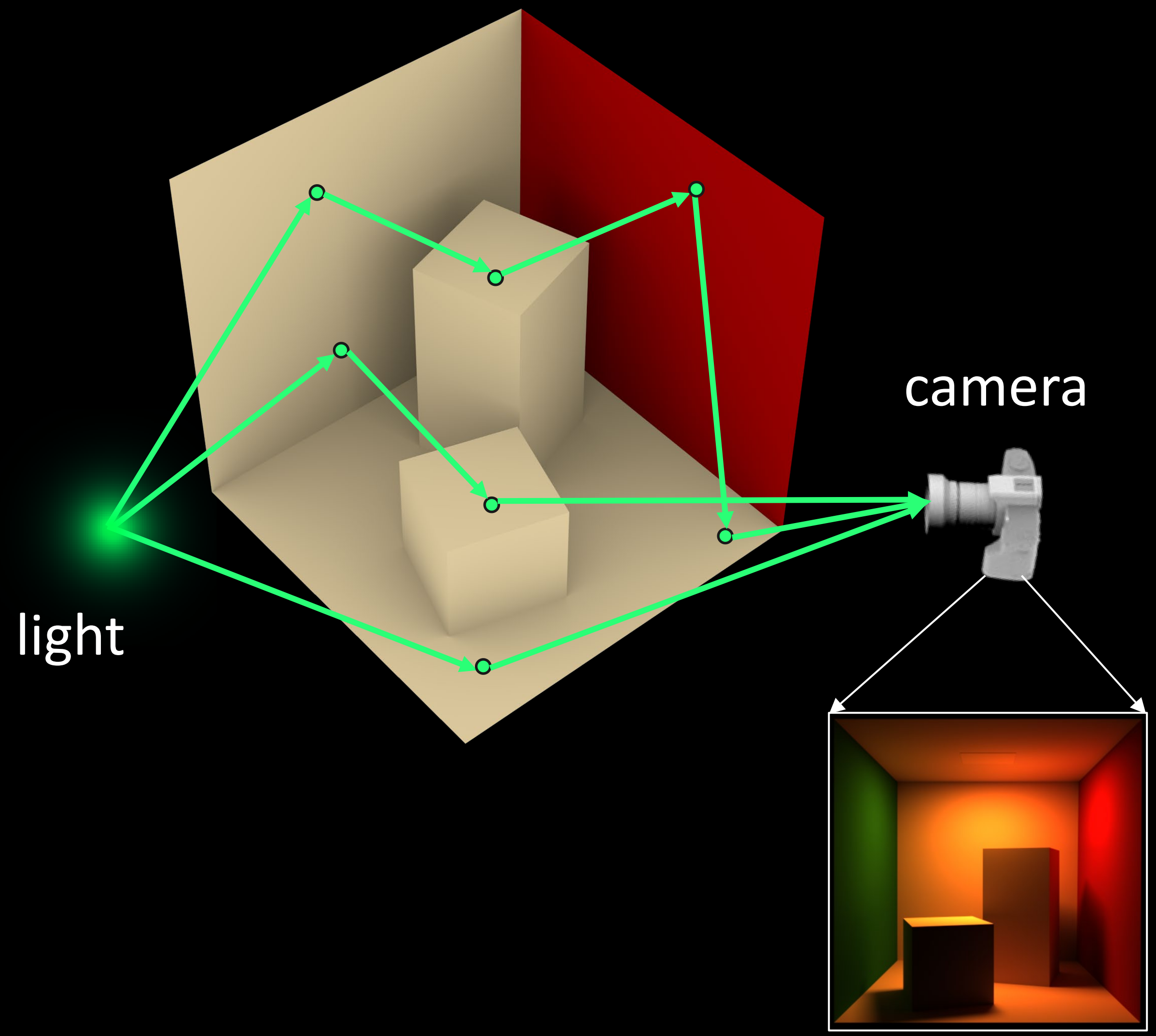
$$I(\pi) = \int_{\mathbb{P}} f(\bar{\mathbf{x}};\pi)\,\mathrm{d}\bar{\mathbf{x}}$$

$\bar{\mathbf{x}}$ → 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)

Monte Carlo rendering: approximating path integrals



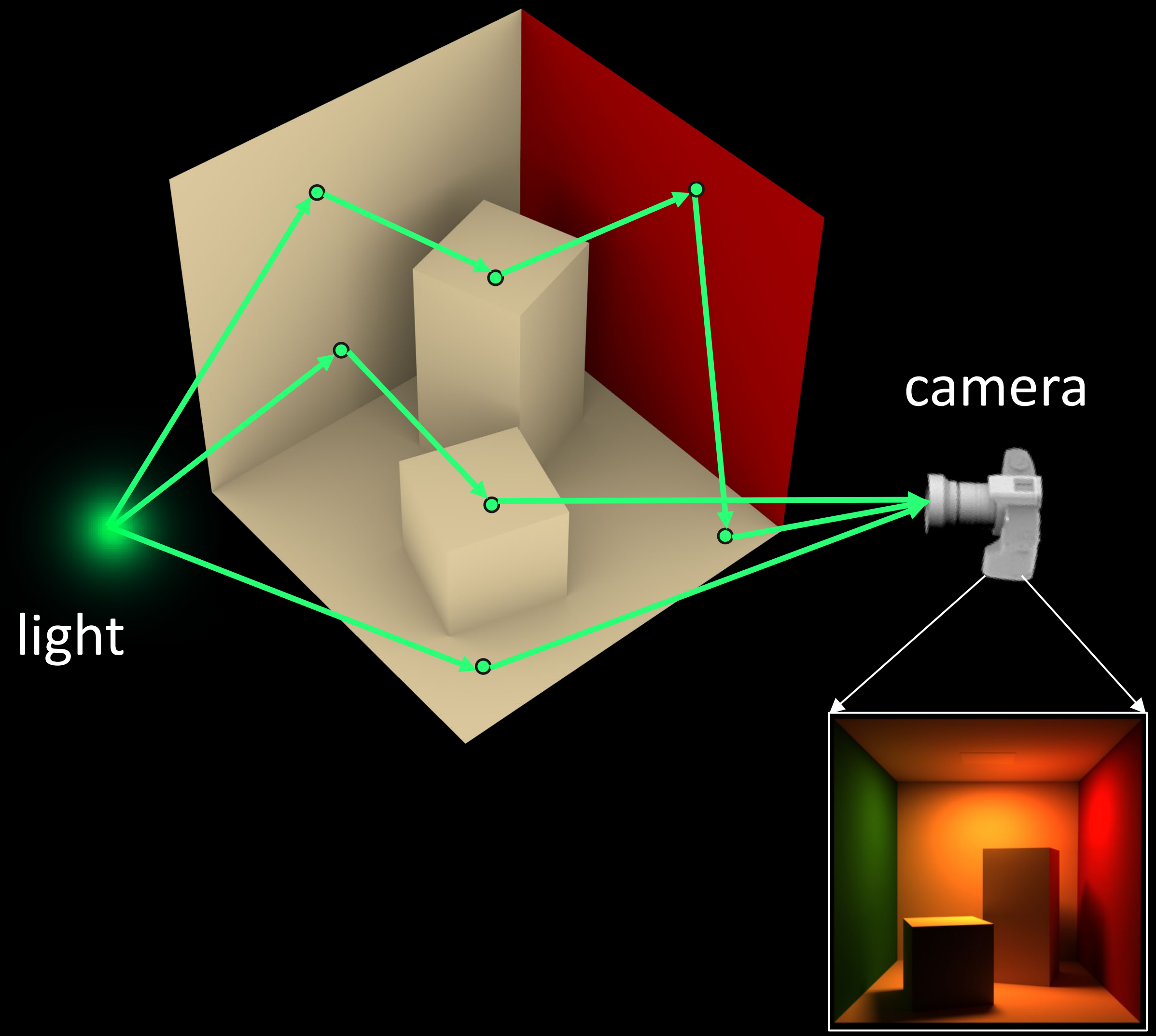
$$I(\pi) \approx \underbrace{\sum_{i=1}^{N} \frac{f(\bar{\mathbf{x}}_i;\pi)}{p(\bar{\mathbf{x}}_i;\pi)}}_{MC(\pi)}$$

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

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

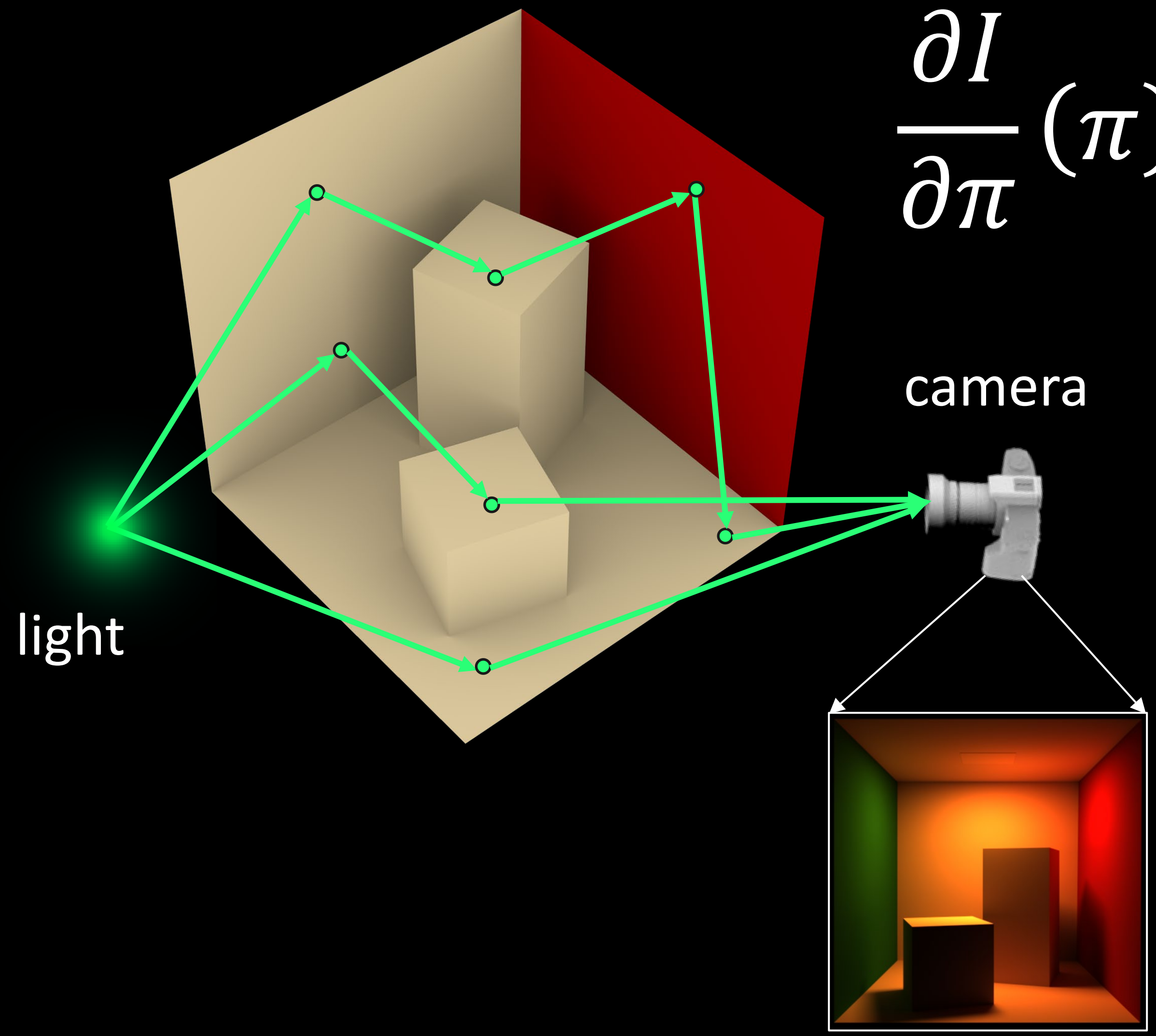
Algorithms such as path tracing, bidirectional path tracing, etc. sample paths.

How can we approximate the derivative of the image?



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

Easy approach 1: finite differences

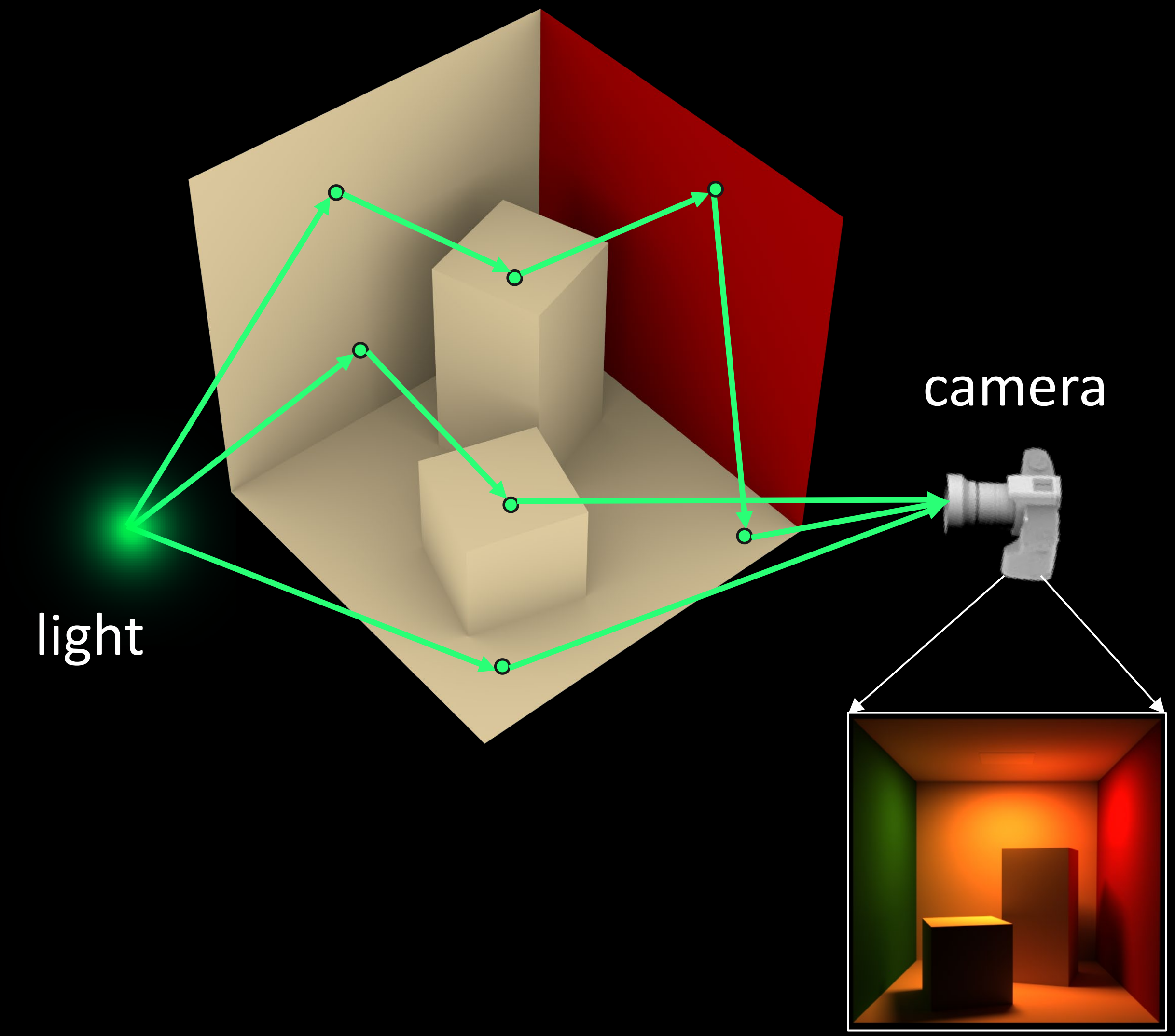


$$\frac{\partial I}{\partial \pi}(\pi) \approx \frac{MC(\pi + \varepsilon) - MC(\pi - \varepsilon)}{2\varepsilon}$$

Any issues with this?

- Incredibly noisy for small ε
- Techniques for noise reduction exist, but generally impractical approach
- Very expensive as number of differentiated parameters increases
- Very inaccurate for large ε, especially in the presence of discontinuities

Easy approach 2: automatic differentiation



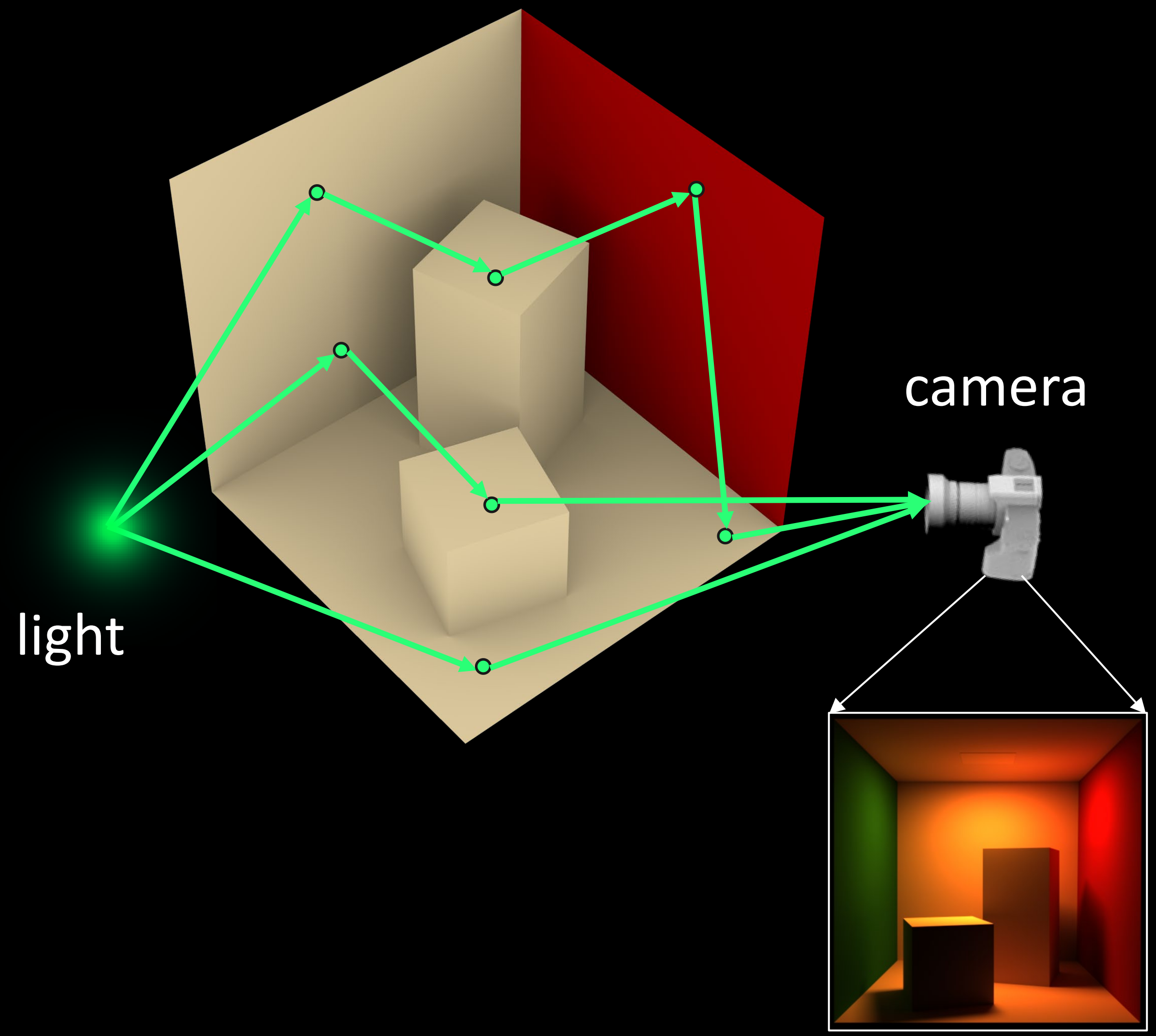
$$\frac{\partial I}{\partial \pi}(\pi) \approx \text{autodiff}(MC(\pi))$$

Any issues with this?

- Path contributions often include parameter-dependent discontinuities
- Path sampling techniques often include parameter-dependent discontinuities
- No flexibility in selecting path sampling techniques for differentiable rendering
- Rendering can produce enormous, non-local computational graphs.

DIFFERENTIATING GLOBAL ILLUMINATION WITH RESPECT TO LOCAL PARAMETERS

Images as path integrals



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

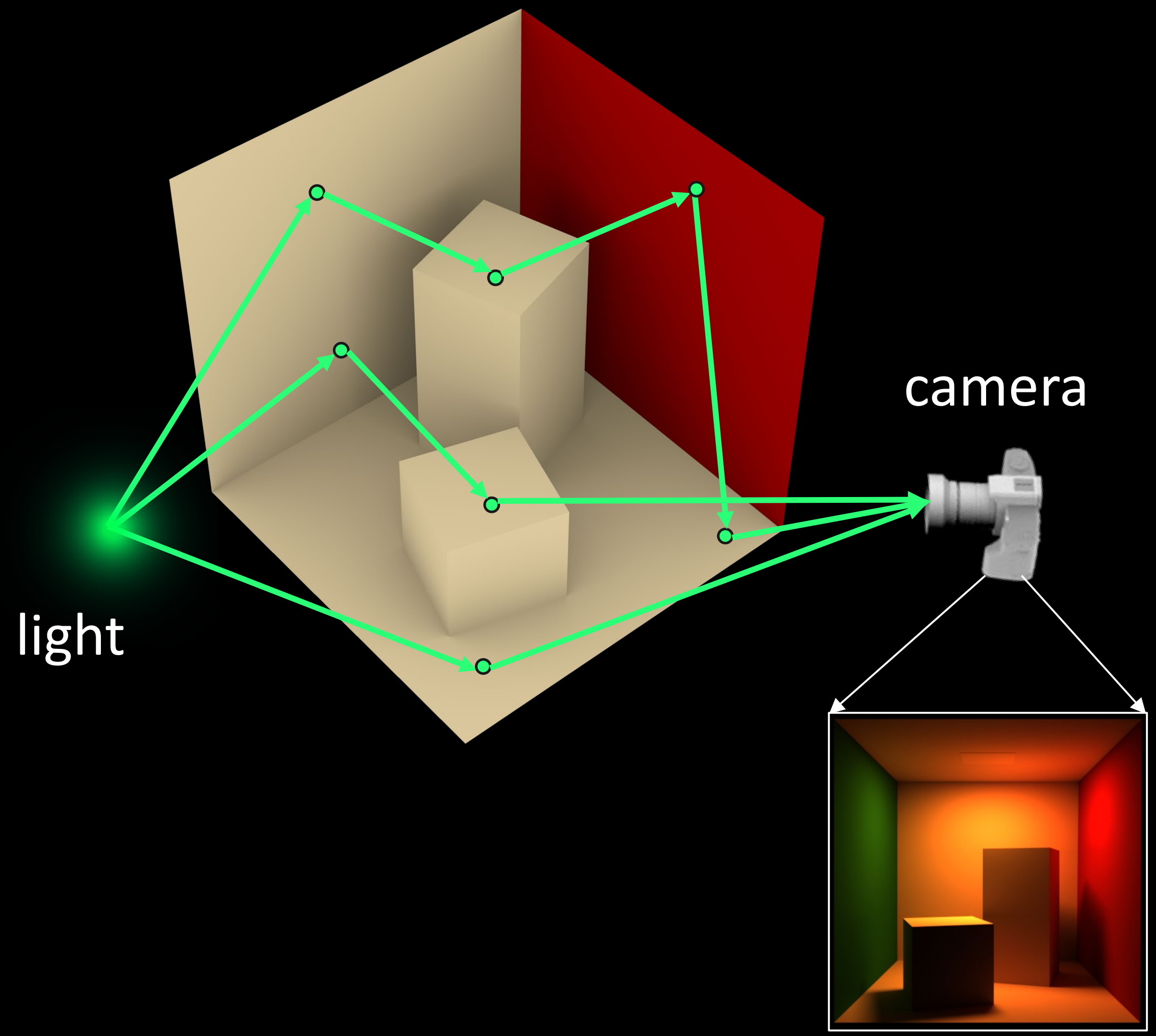
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → 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 π

Derivatives of images as path integrals



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

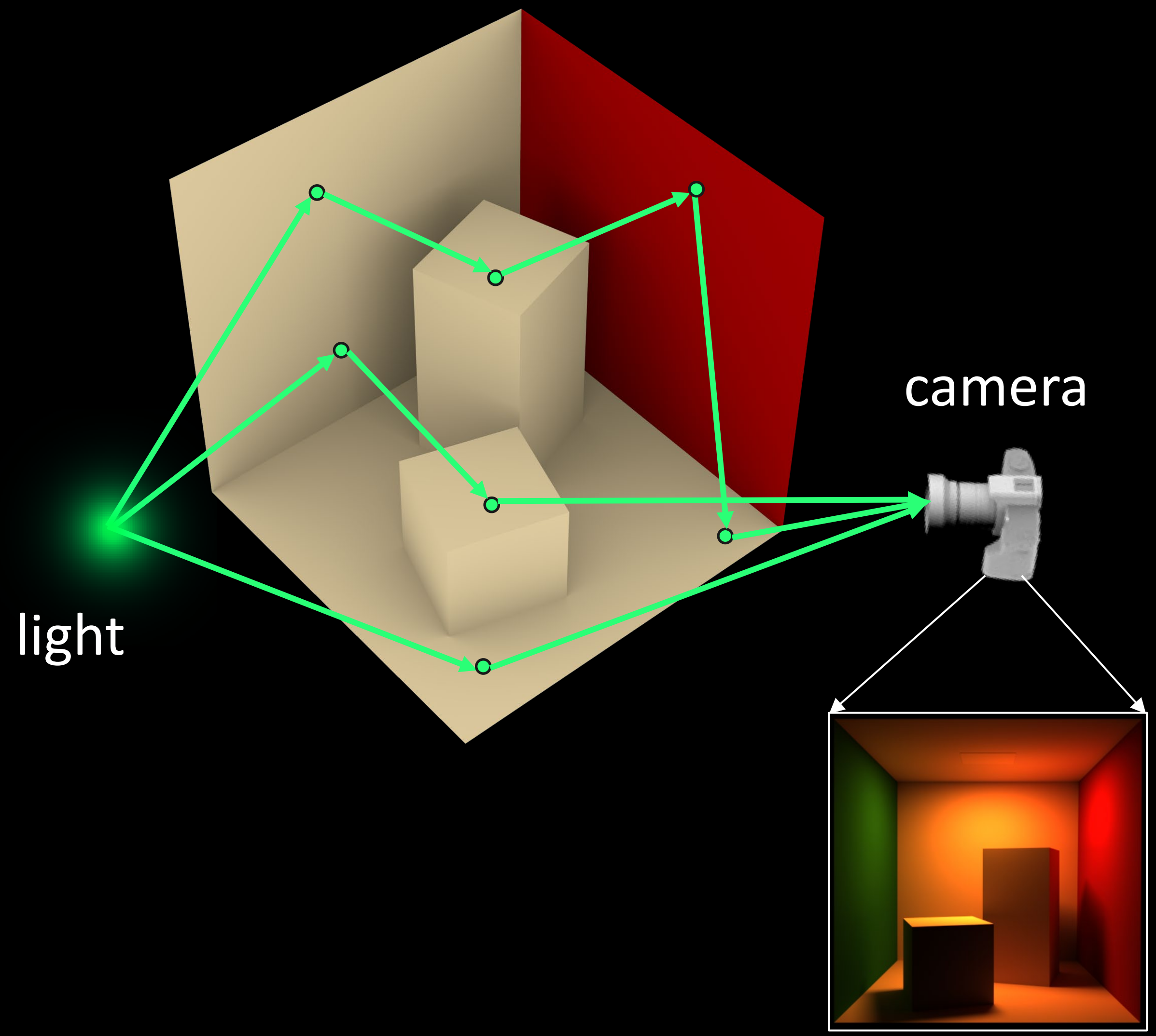
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → 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 π

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

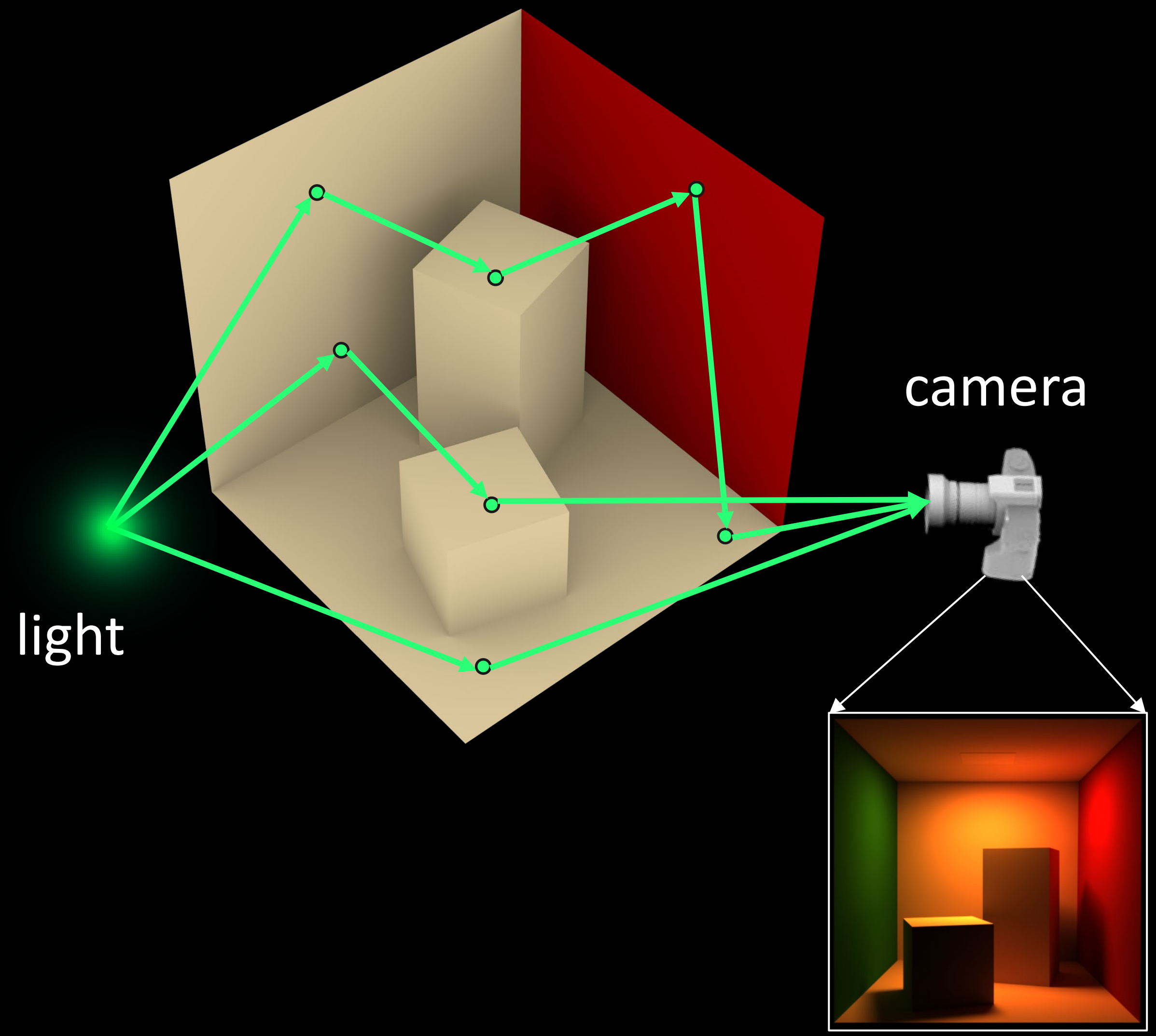
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices <u>on surfaces</u>

$\mathbb{P}$ → 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 π

Monte Carlo differentiable rendering (for local parameters)



This term is generally easy to compute during path tracing

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

$\bar{x}_i$ → 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

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

$$\sum_{b=1}^{B} \boxed{\frac{\frac{\partial f_s}{\partial \pi}(x_{b-1} \to x_b \to x_{b+1};\pi)}{f_s(x_{b-1} \to x_b \to x_{b+1};\pi)}}$$

Score function of f_s

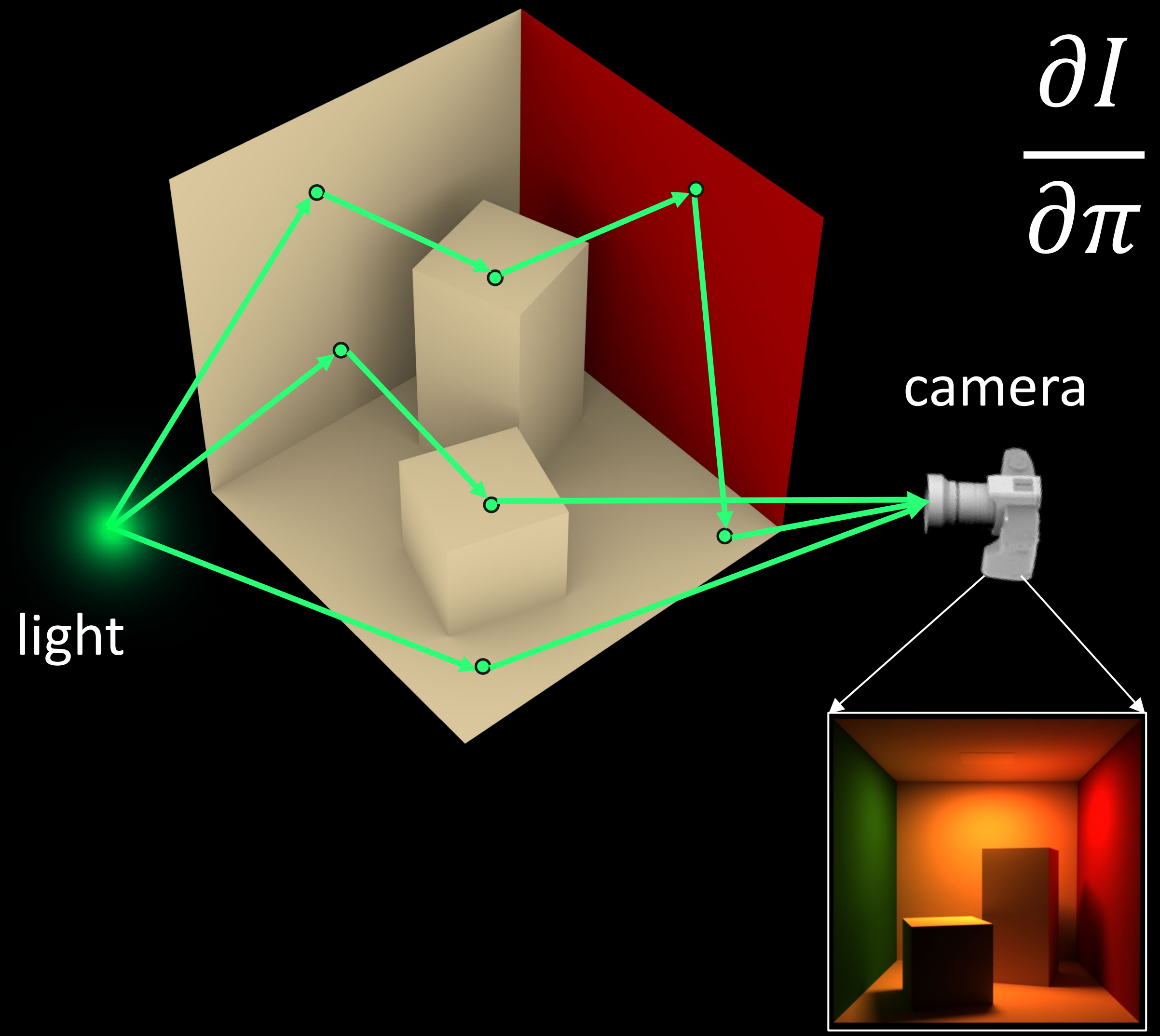
At each path vertex:

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

Multiply the two at end of path

Even simpler: use autodiff

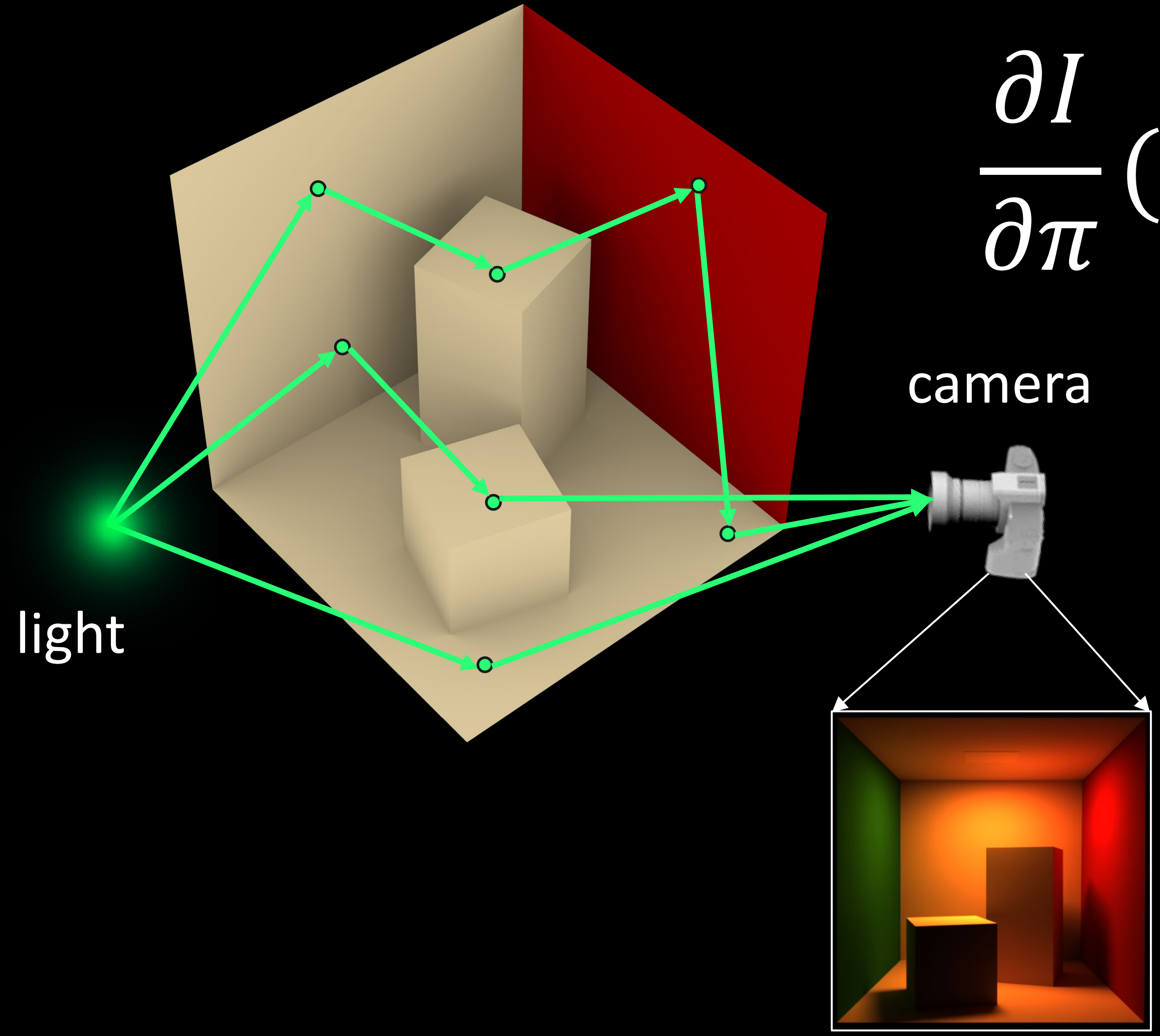
$$\frac{\partial I}{\partial \pi}(\pi) \approx \sum_{i=1}^{N} \frac{\text{autodiff}(f(\bar{\mathbf{x}}_i;\pi))}{p(\bar{\mathbf{x}}_i;\pi)}$$



$\bar{\boldsymbol{x}}_i$ → <u>Randomly sampled</u> light paths

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

Compare with...

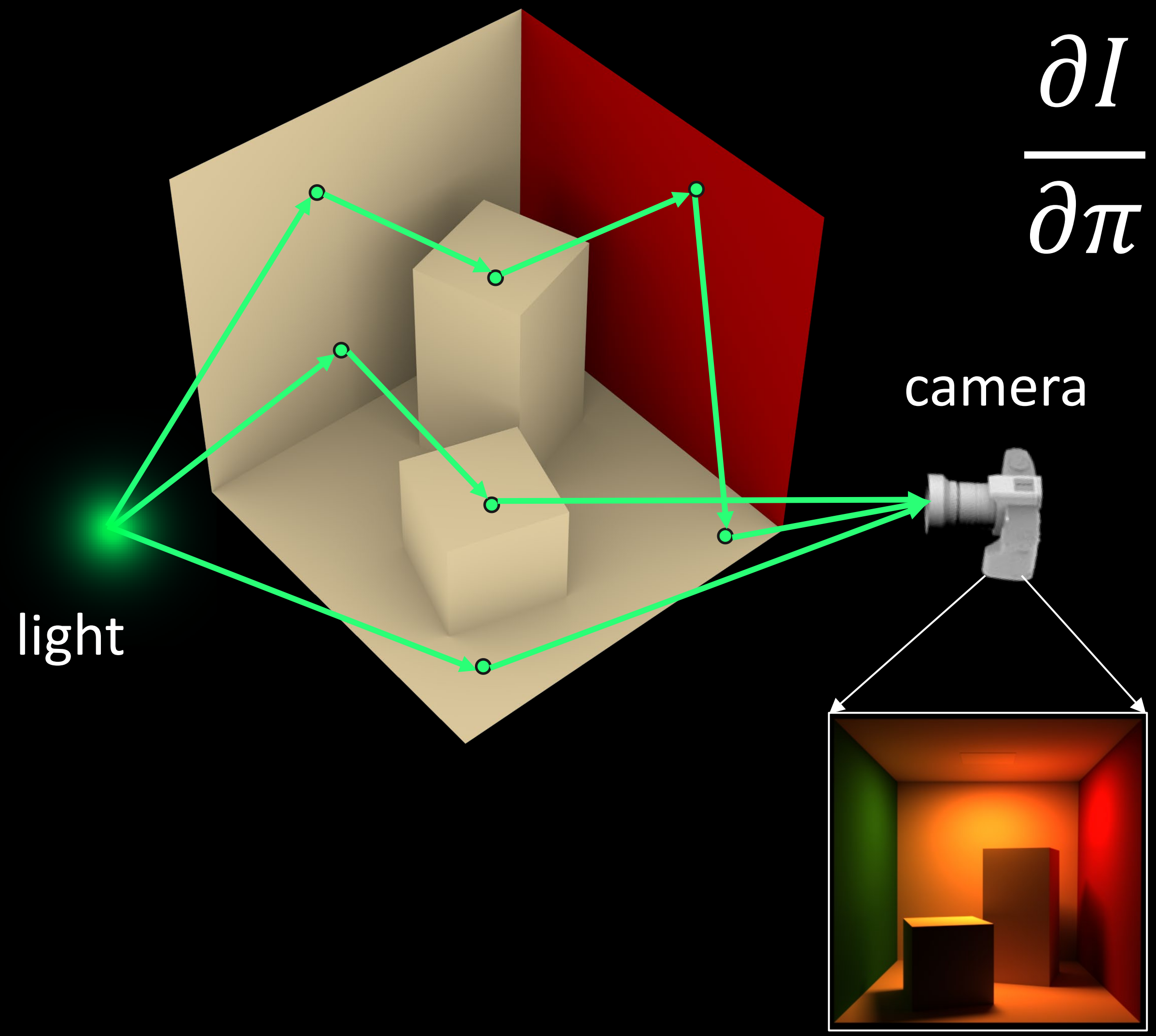


$$\frac{\partial I}{\partial \pi}(\pi) \approx \text{autodiff}\left(\sum_{i=1}^{N} \frac{f(\bar{\mathbf{x}}_i;\pi)}{p(\bar{\mathbf{x}}_i;\pi)}\right)$$

$\bar{x}_i$ → <u>Randomly sampled</u> light paths

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

Even simpler: use autodiff



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

- Depending on how badly p approximates f, can have much lower variance.
- Remember: *Compute an estimate of the derivative, not a derivative of the estimator.*

OpenDR: An Approximate Differentiable Renderer

[Loper and Black 2015]

- Approach: autodiff of the entire renderer.
- 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.

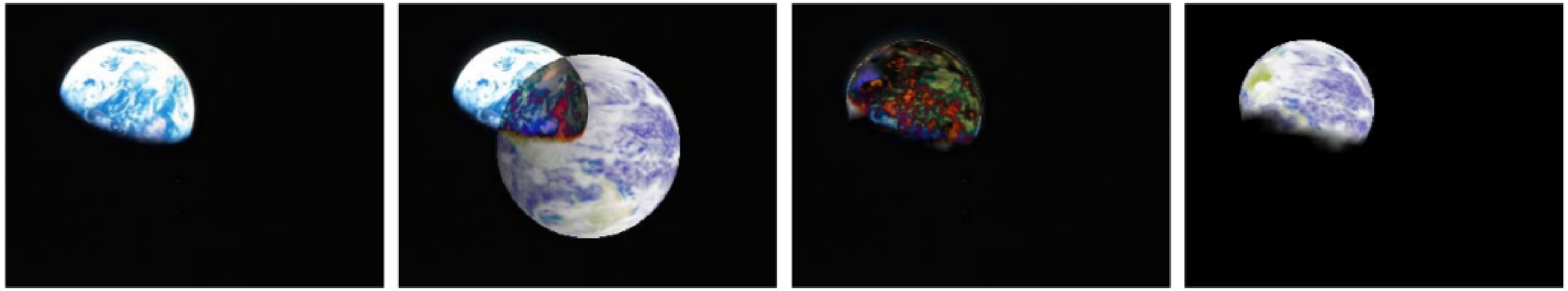
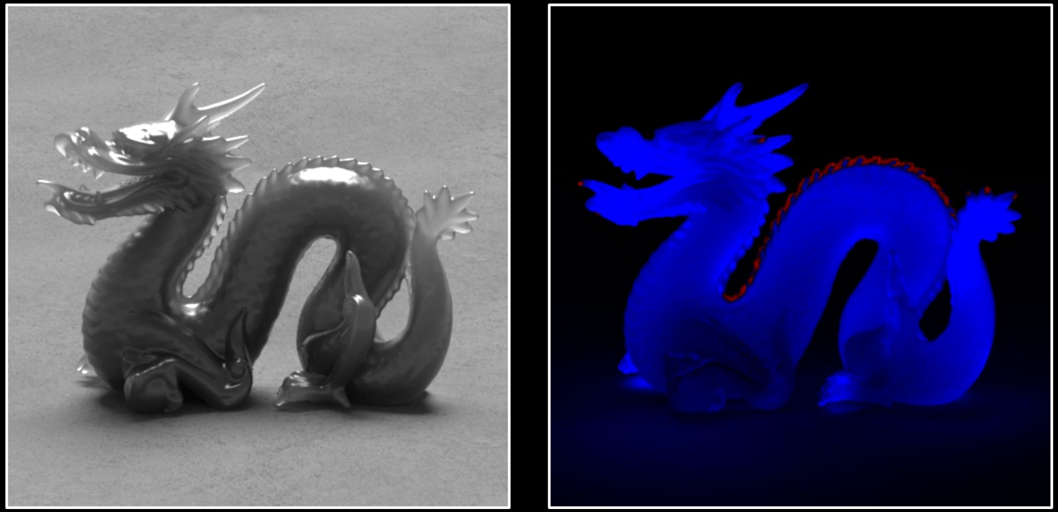
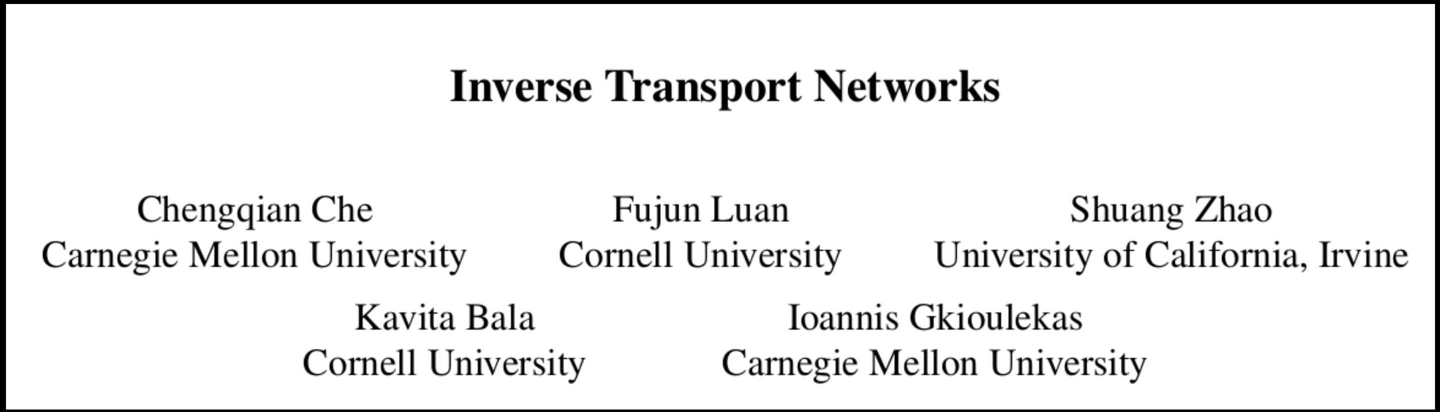


Fig. 4. Illustration of optimization in Figure 3. In order: observed image of earth, initial absolute difference between the rendered and observed image intensities, final difference, final result.

Compute an estimate of the derivative



derivative wrt volumetric density



Inverse Transport Networks

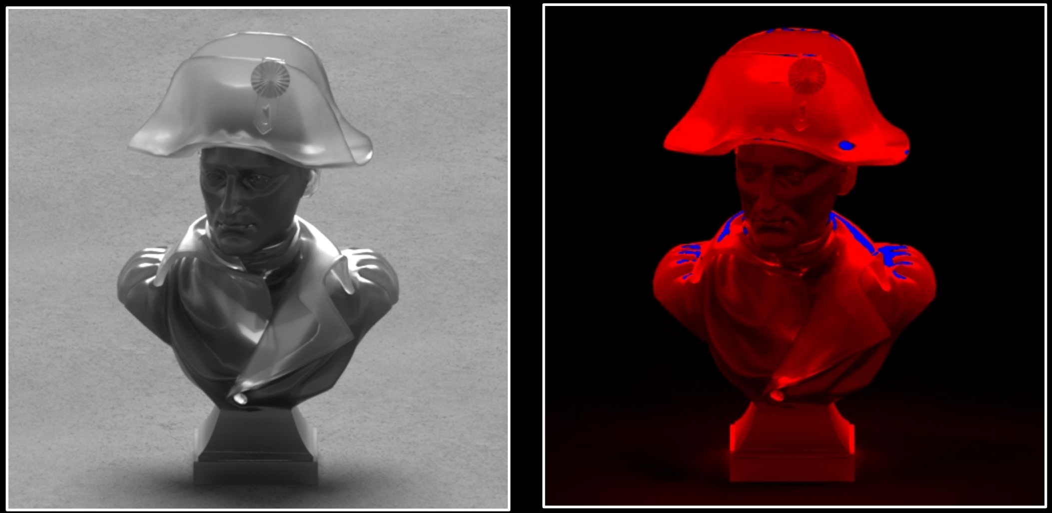
Chengqian Che
Carnegie Mellon University

Fujun Luan
Cornell University

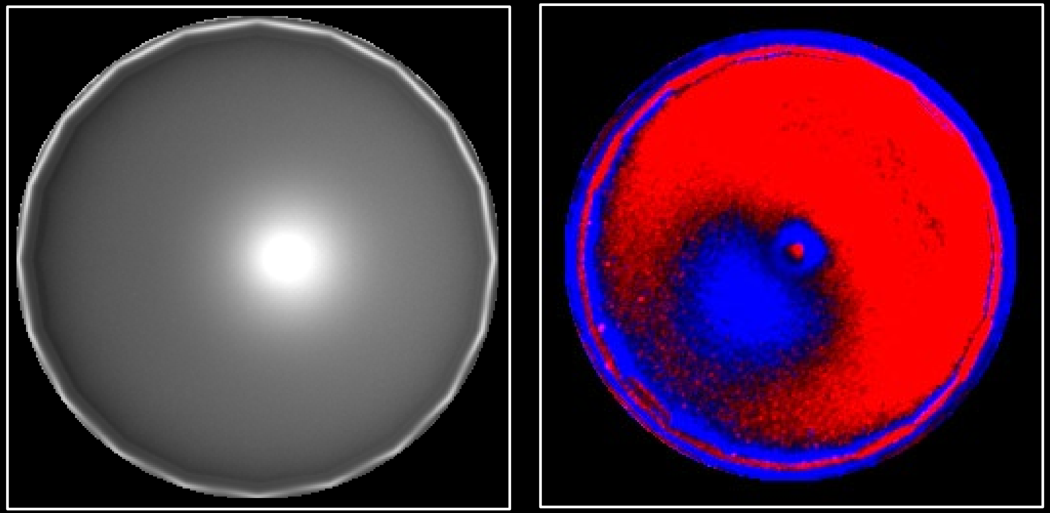
Shuang Zhao
University of California, Irvine

Kavita Bala
Cornell University

Ioannis Gkioulekas
Carnegie Mellon University

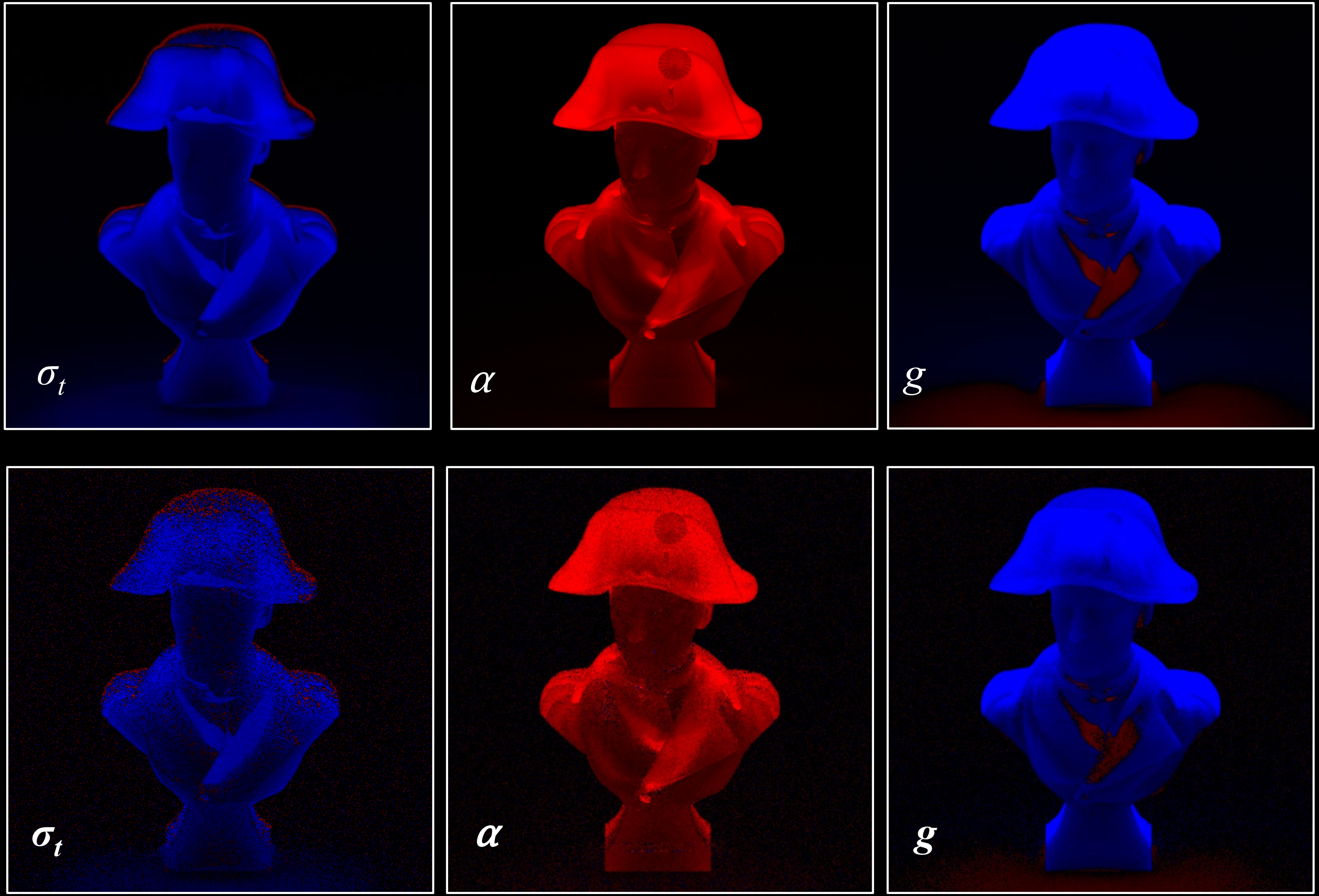


derivative wrt BRDF



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

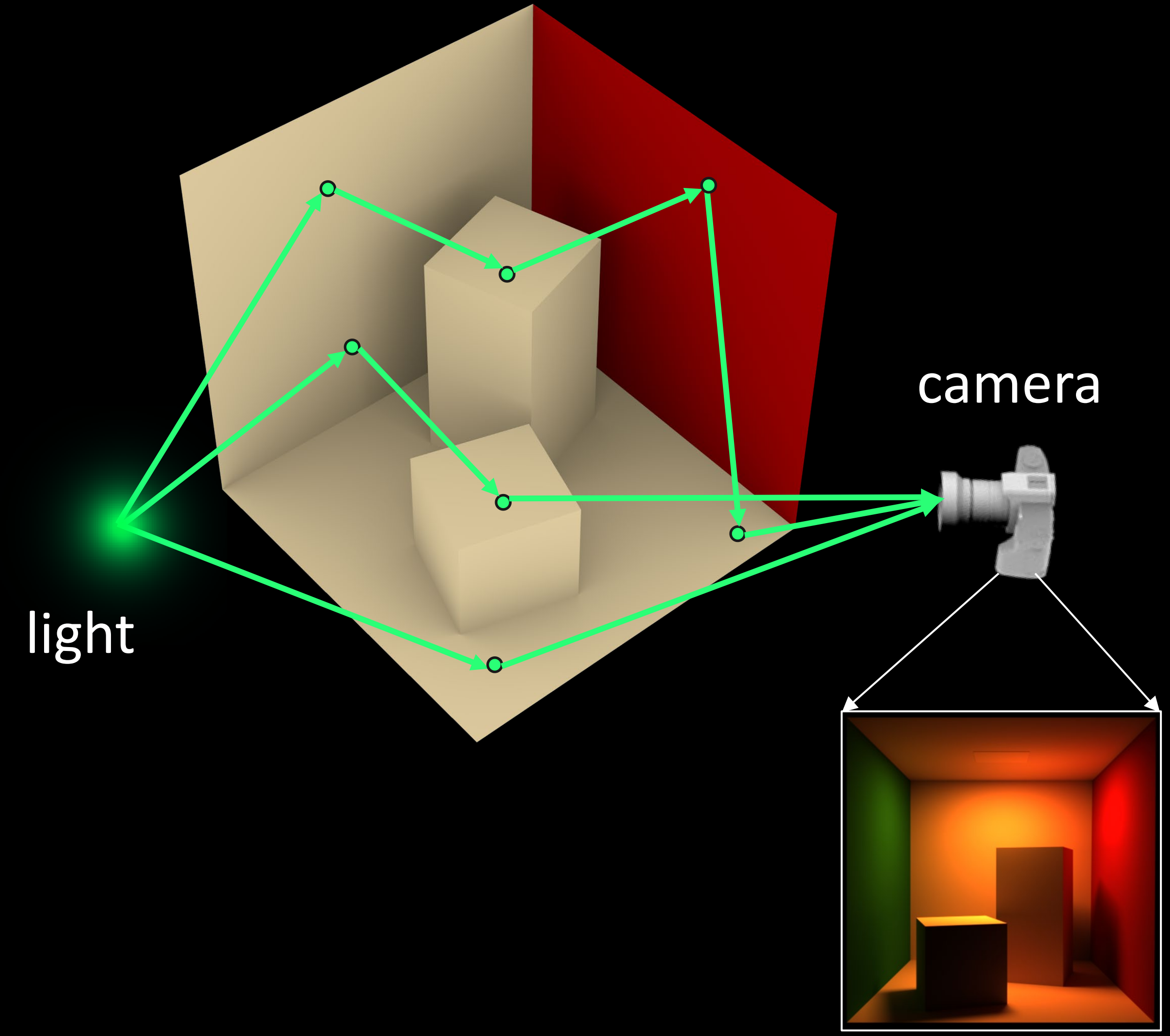
derivative wrt volumetric density

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.

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

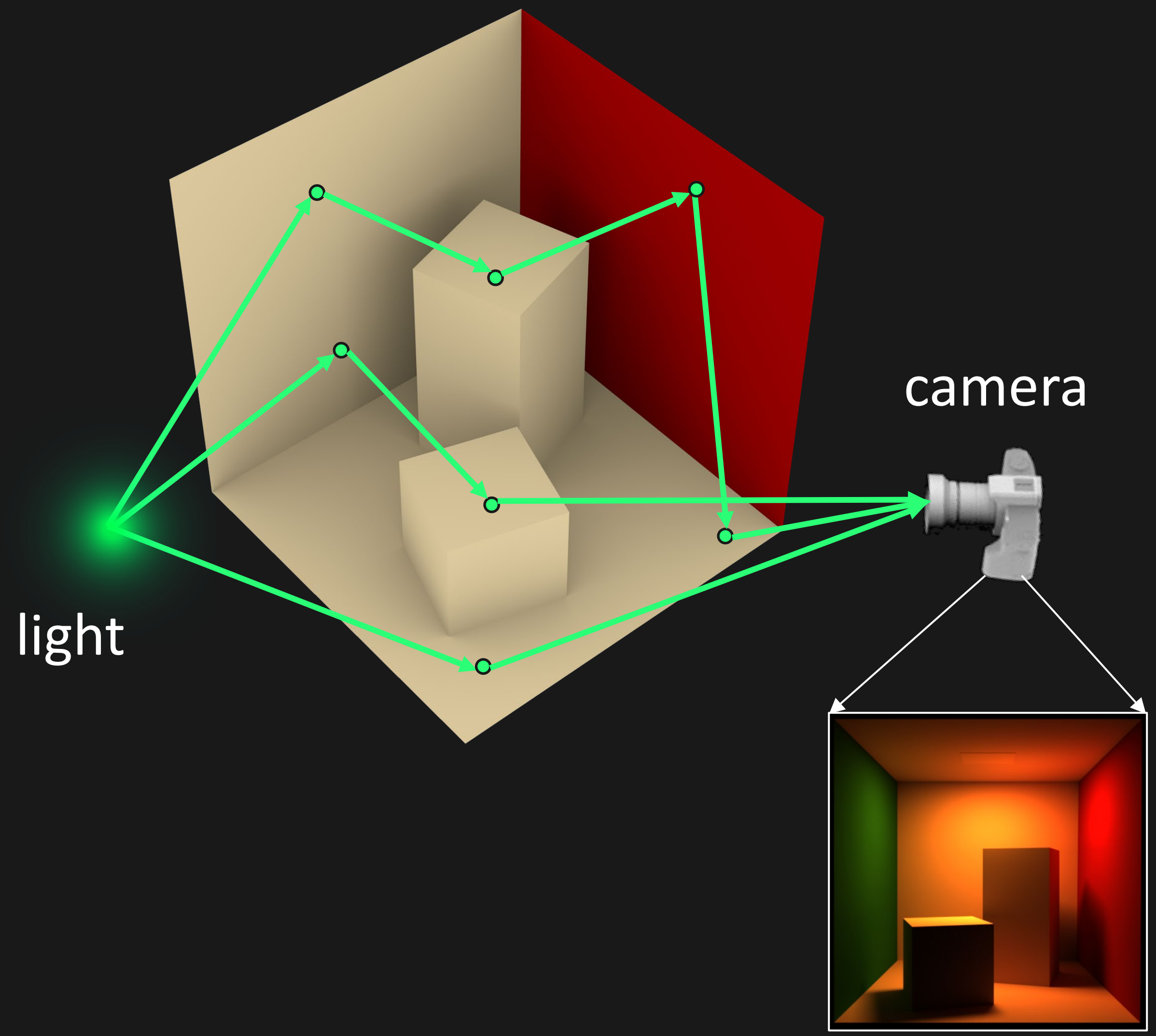
$\bar{\mathbf{x}}$ → Light path, set of ordered vertices on surfaces

$\mathbb{P}$ → 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 π

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.

DIFFERENTIATING GLOBAL ILLUMINATION WITH RESPECT TO GLOBAL PARAMETERS

We'll work with the rendering equation for a few

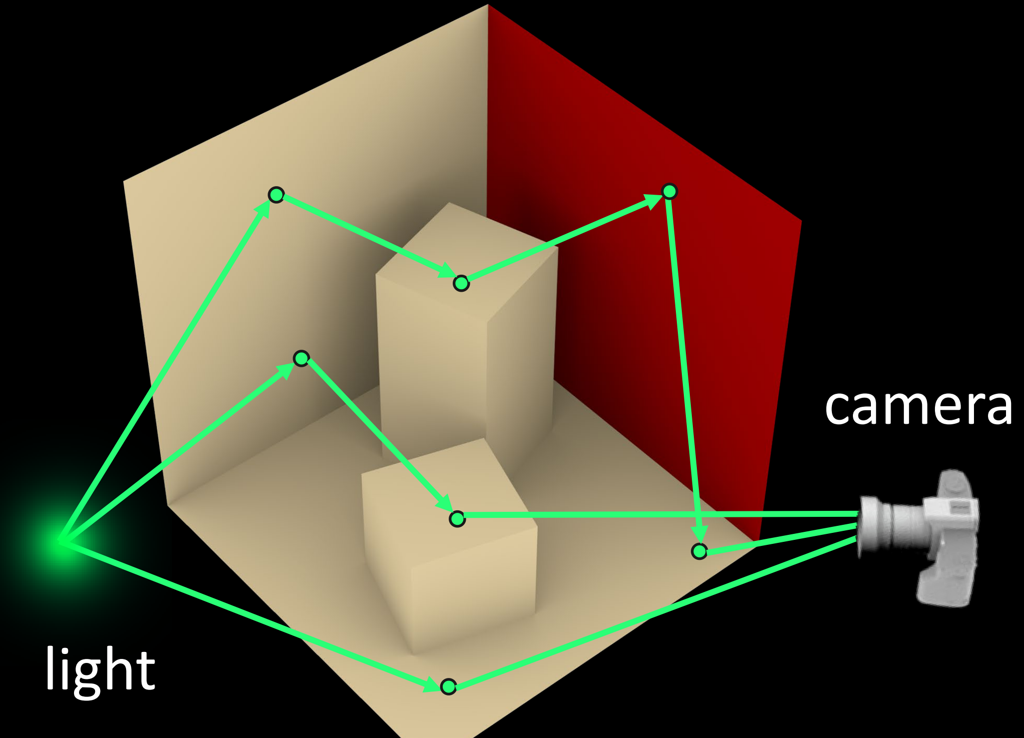
$$L(x,\omega;\pi) = \int_{G(\pi)} L(x' \to x;\pi) f(x' \to x,\omega;\pi) V(x' \leftrightarrow x;\pi) \mathrm{d}A(x')$$

L → Radiance at a point and direction

G → All surfaces in the scene

f → Reflection, foreshortening, and fall-off

V → Visibility



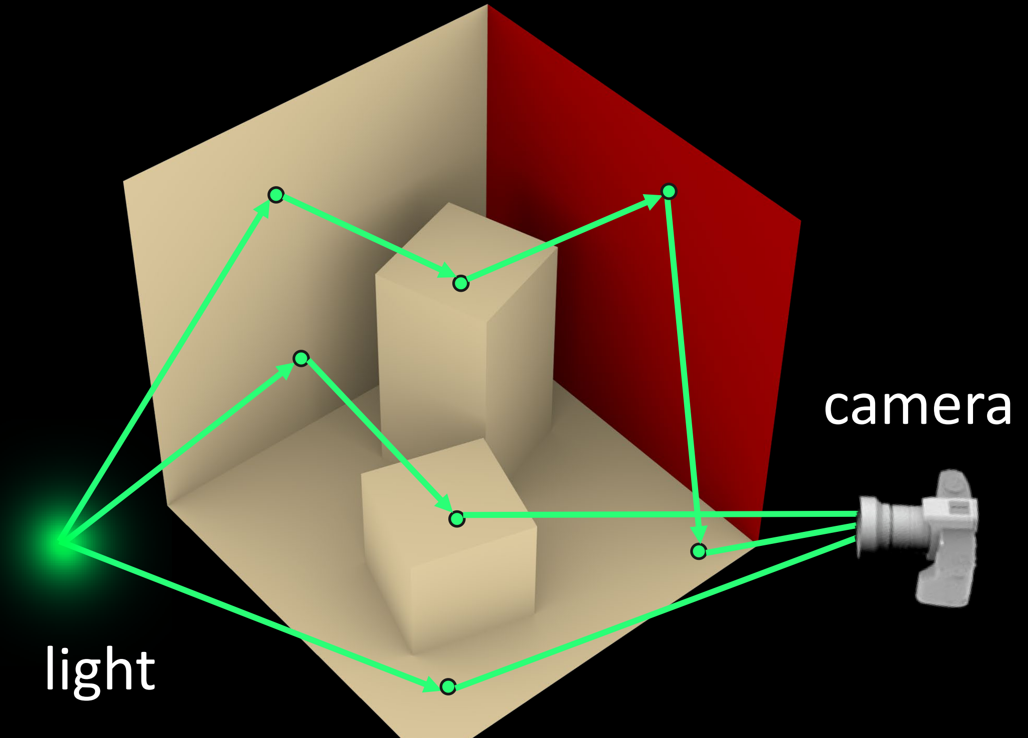
Let's slightly rewrite the rendering equation

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

L → Radiance at a point and direction

V → All visible surfaces in the scene

f → Reflection, foreshortening, and fall-off



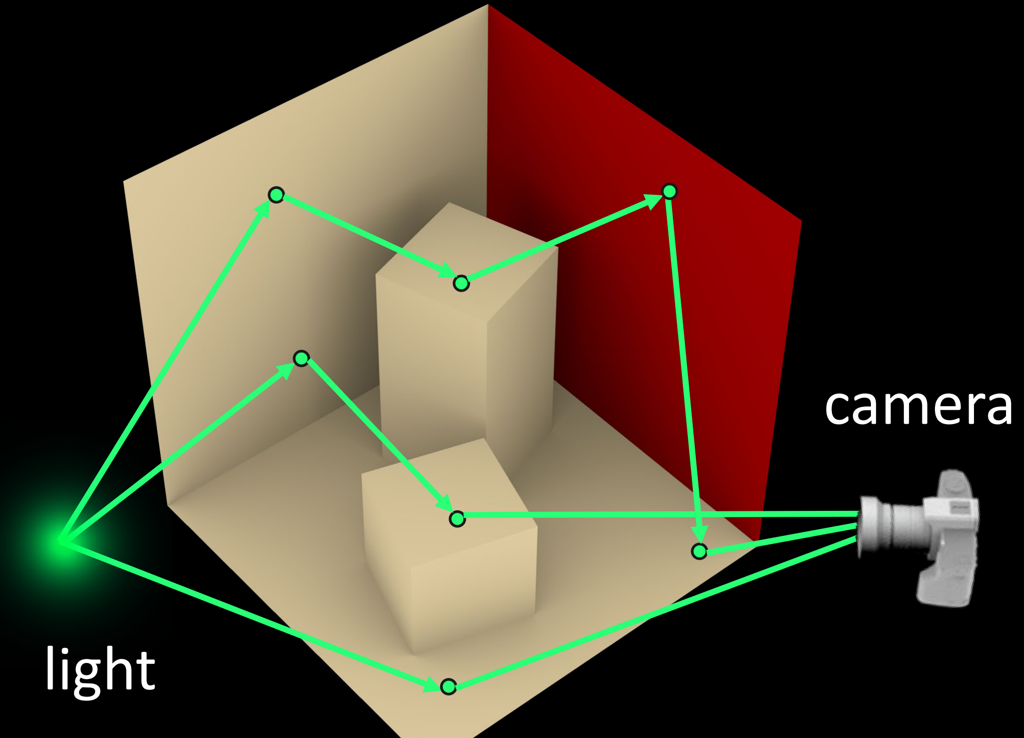
Let's differentiate it

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

L → Radiance at a point and direction

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

f → Reflection, foreshortening, and fall-off



Can we just move the integral inside?

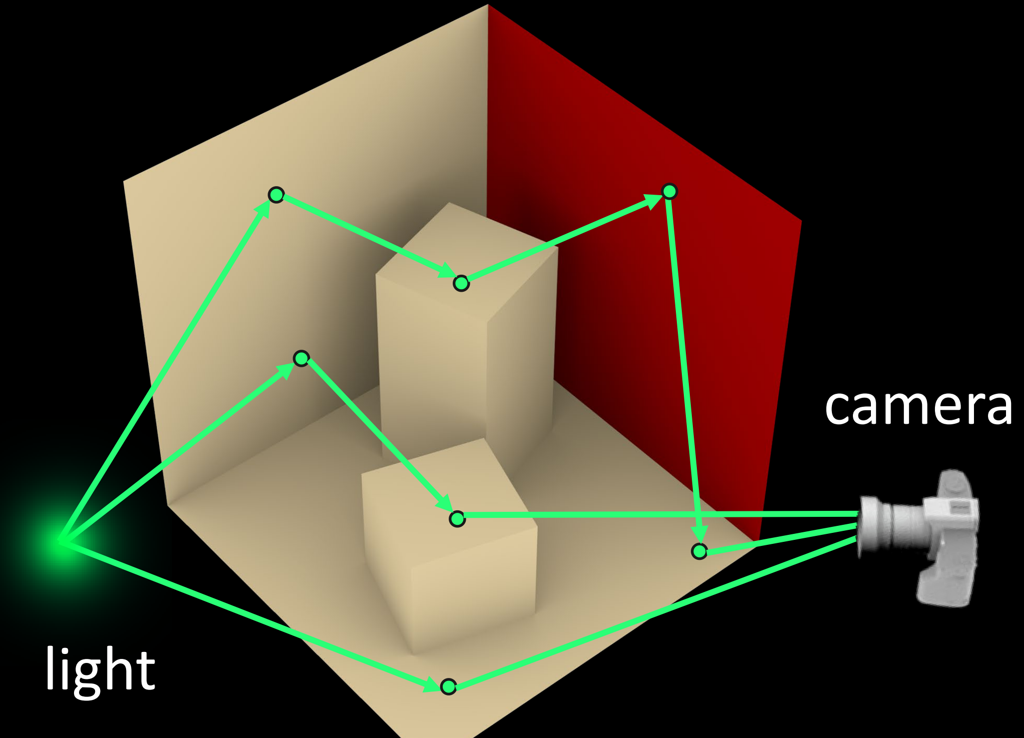
Let's differentiate it

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

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?

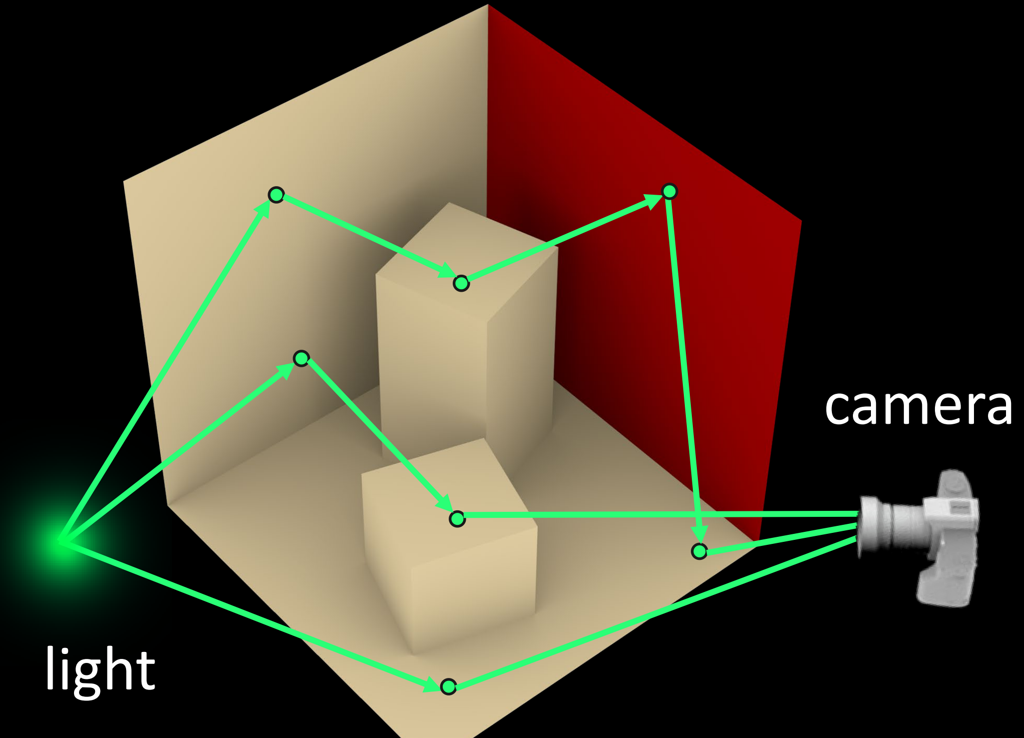
Let's differentiate it

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

L → Radiance at a point and direction

V → All visible surfaces in the scene

f → Reflection, foreshortening, and fall-off



What are the "boundary" and discontinuities of V?

Boundaries

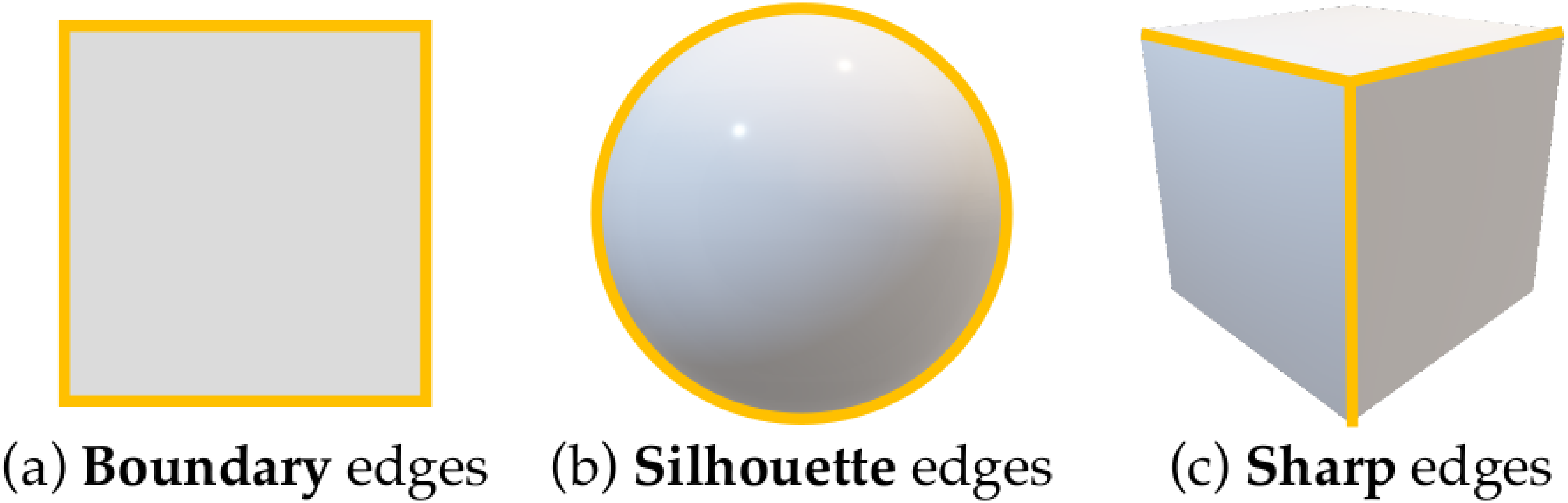
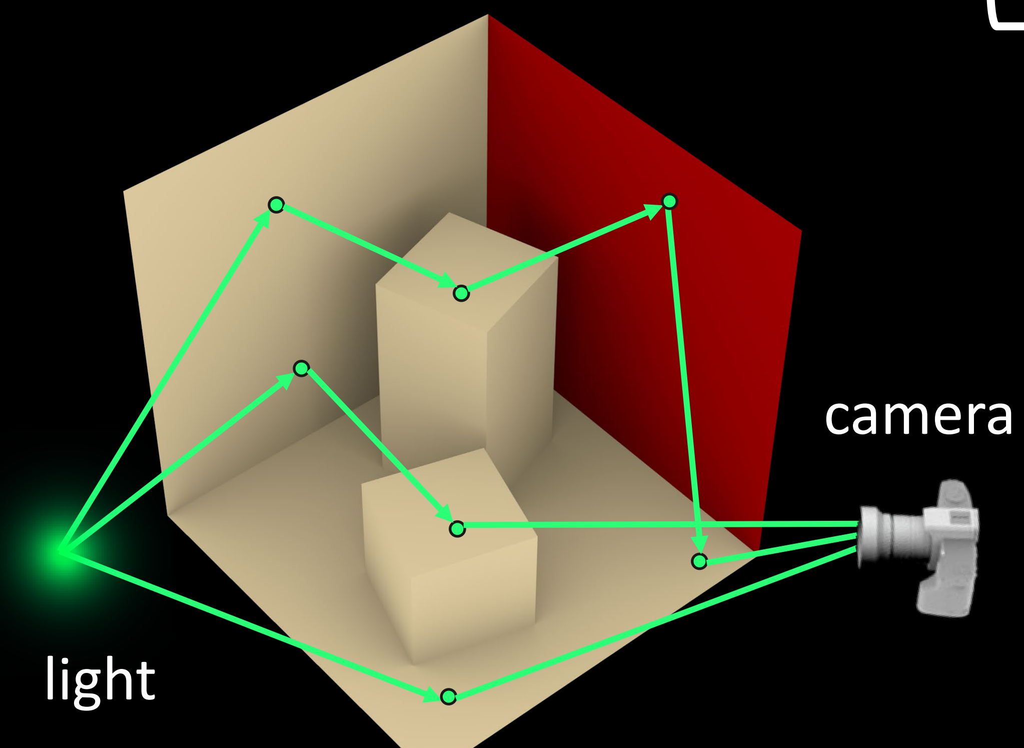


Fig. 5. Three types of edges (drawn in yellow) that can cause geometric discontinuities: (a) boundary, (b) silhouette, and (c) sharp.

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

Boundary edge detection and sampling

Not terribly good, as we ray trace, we need to:

- recompute silhouette at <u>each</u> vertex
- branch twice

Global geometry differentiation

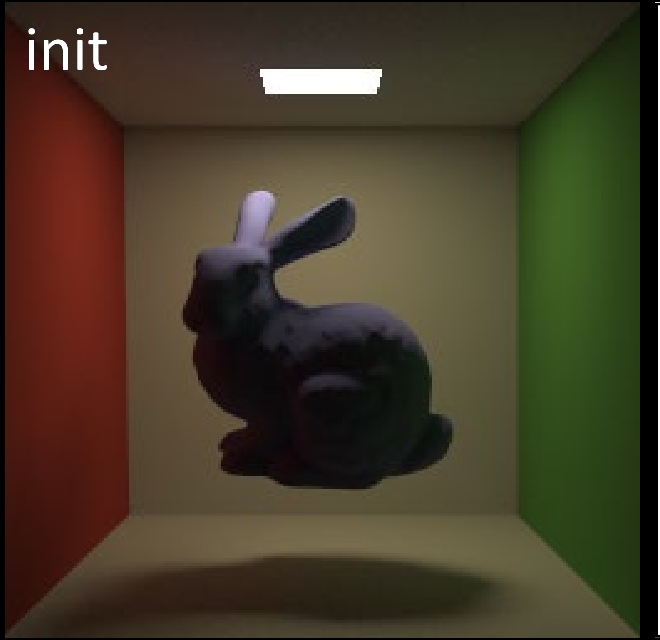
Differentiable Monte Carlo Ray Tracing through Edge Sampling

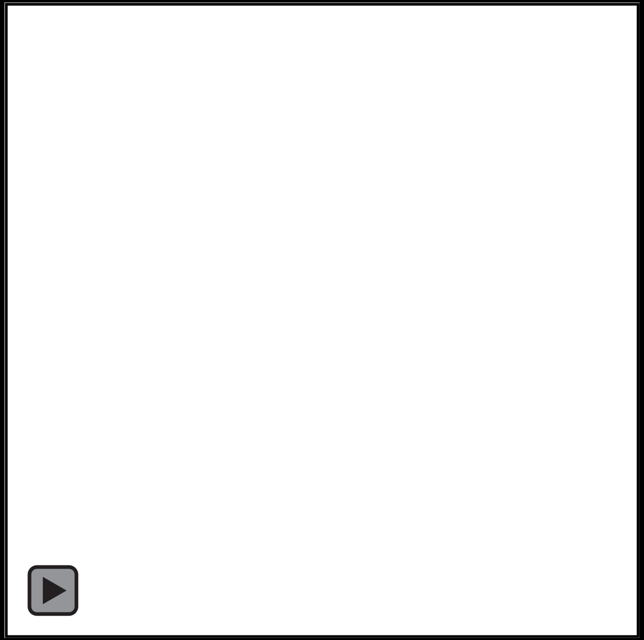
TZU-MAO LI, MIT CSAIL
MIIKA AITTALA, MIT CSAIL
FRÉDO DURAND, MIT CSAIL
JAAKKO LEHTINEN, Aalto University & NVIDIA

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

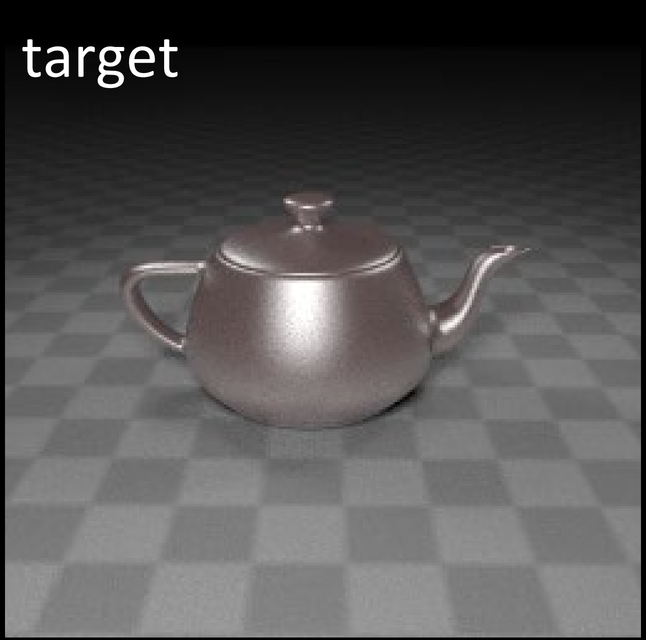
Chia-Yin Tsai, Aswin C. Sankaranarayanan, and Ioannis Gkioulekas
Carnegie Mellon University

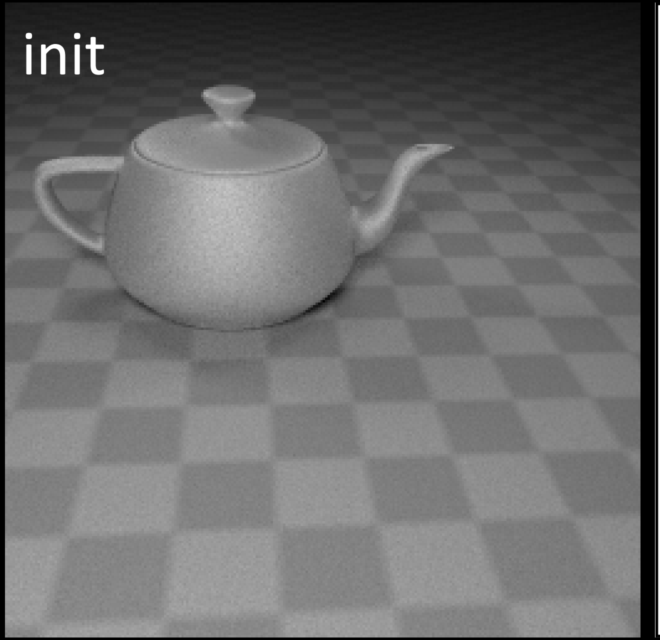
Global geometry differentiation

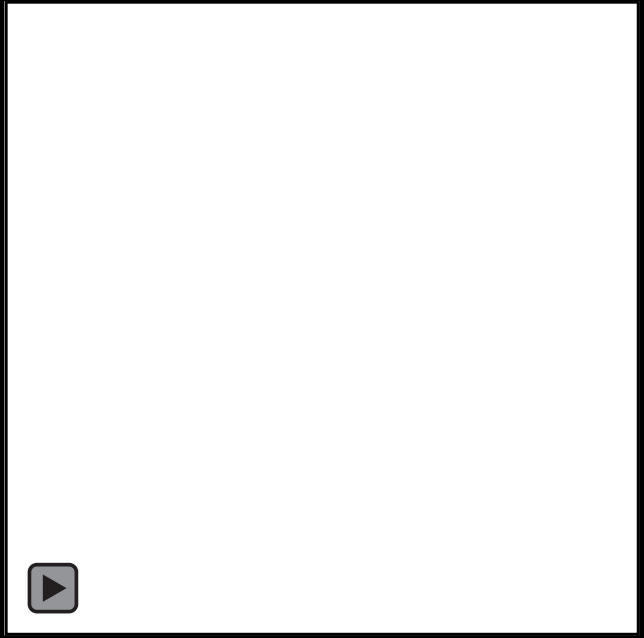




optimize
bunny
pose



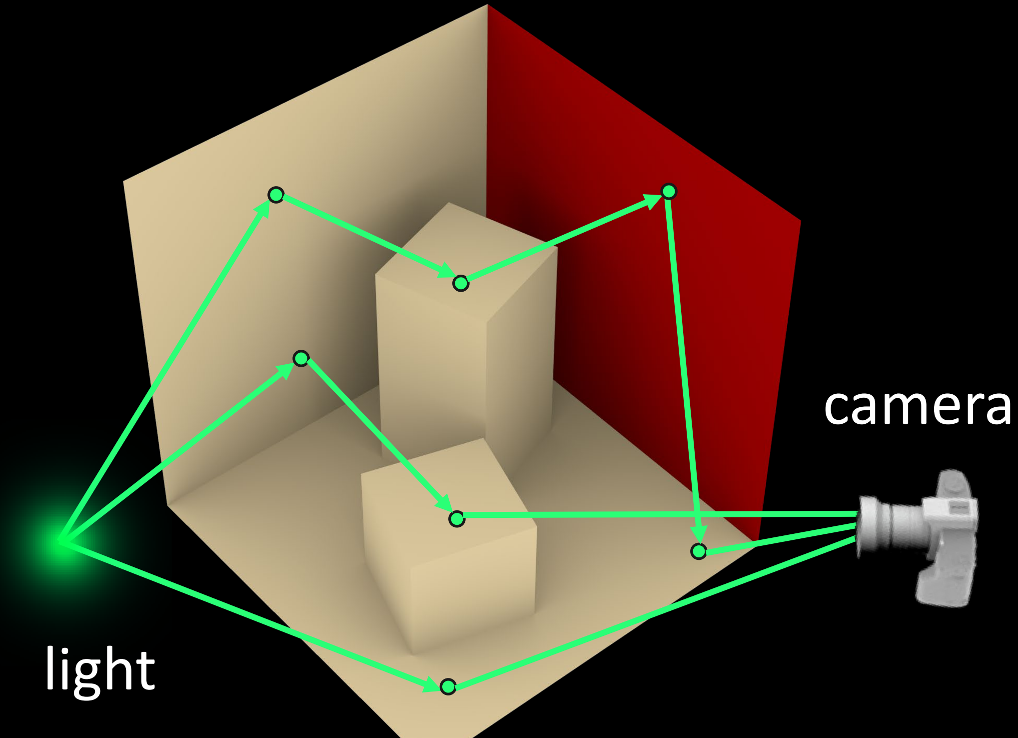




optimize
reflectance
and camera
pose

Let's differentiate it

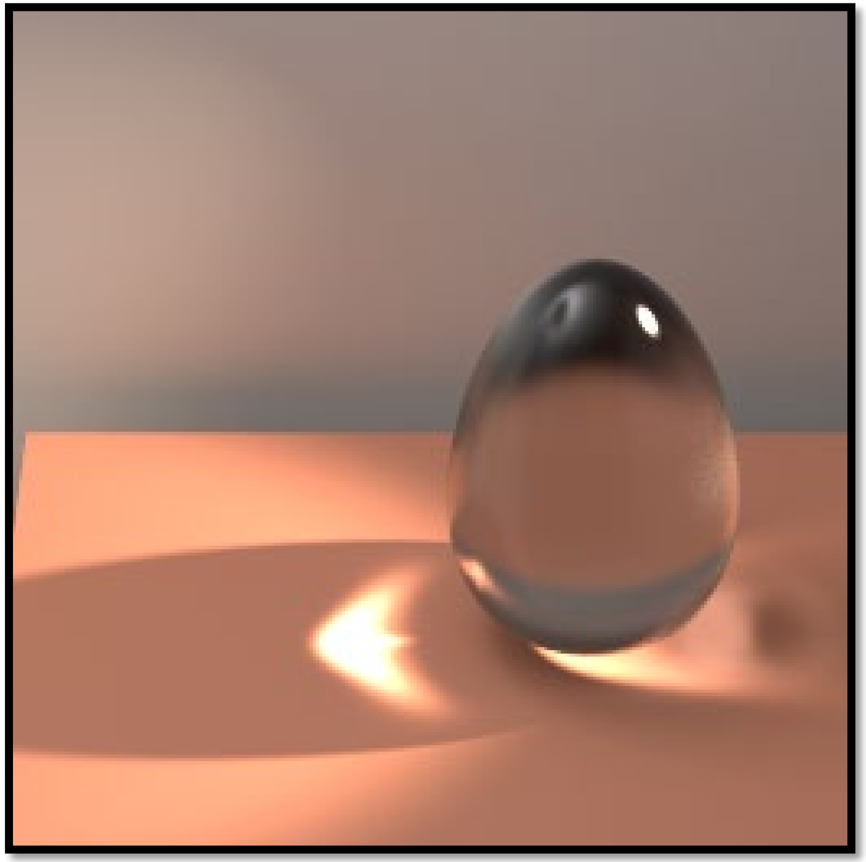
$$\frac{\partial}{\partial \pi} L(x, \omega; \pi) = \int_{V(x,\pi)} F\underbrace{\left(\frac{\partial}{\partial \pi} L\right)}_{\text{render derivative of L at some visible point}} \mathrm{d}A(x) + \int_{\partial V(x,\pi)} \underbrace{H(L)}_{\text{render L at some boundary (silhouette) point}} \mathrm{d}\sigma(x)$$



Not terribly good:

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

CHALLENGES

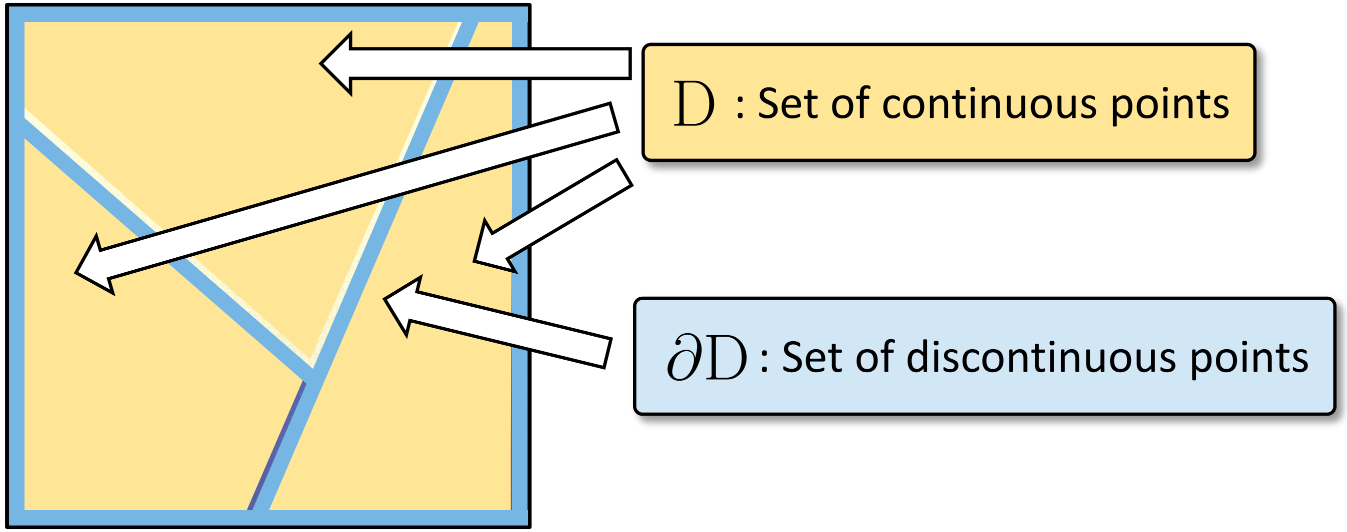


Complex light transport effects

Complex geometry

REPARAMETERIZATION APPROACHES

THE REYNOLDS TRANSPORT THEOREM



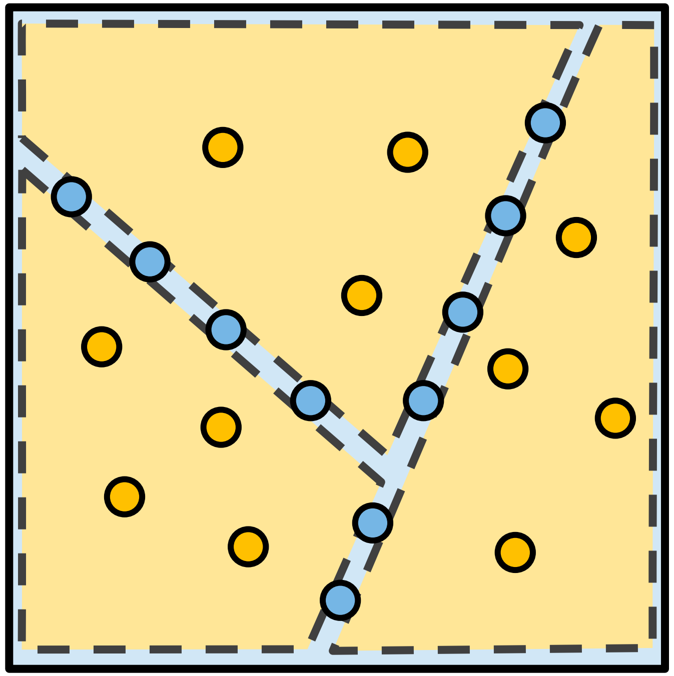
$$\boxed{\partial_\theta \int_{\mathrm{D}} f} = \underbrace{\boxed{\int_{\mathrm{D}} \partial_\theta f}}_{\text{Interior term}} + \underbrace{\boxed{\int_{\partial\mathrm{D}} f \vec{\mathbf{v}} \cdot \vec{\mathbf{n}}}}_{\text{Edge term}}$$

CONVERTING EDGE-SAMPLES TO AREA-SAMPLES

$\int_{\partial \mathrm{D}} f \vec{\mathbf{v}} \cdot \vec{\mathbf{n}}$ is estimated through edge-samples

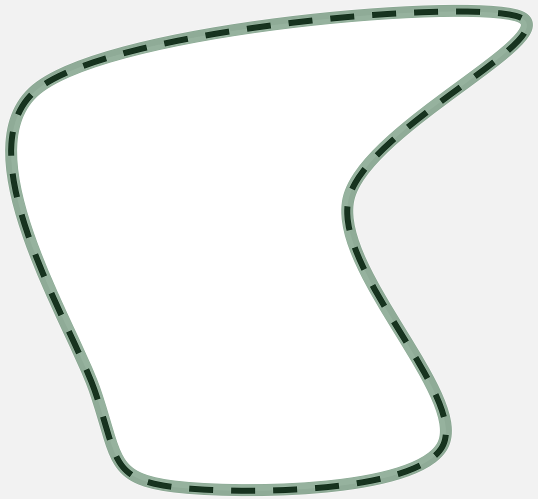
Goal: Rewrite $\int_{\partial \mathrm{D}} f \vec{\mathbf{v}} \cdot \vec{\mathbf{n}}$ into area integral $\int_{\mathrm{D}} g$

$\int_{\mathrm{D}} \nabla \cdot (\vec{\mathcal{V}}_\theta f)$ can be estimated through area-samples



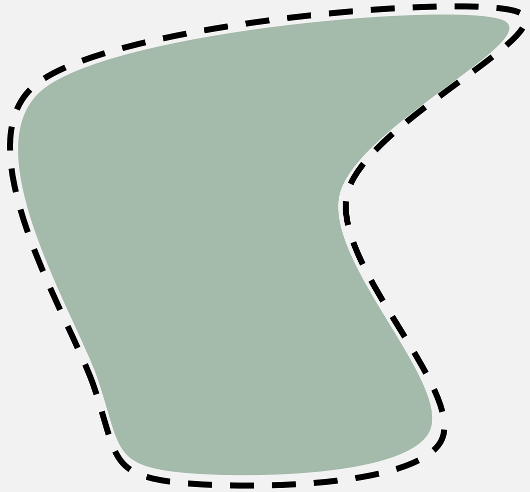
THE DIVERGENCE THEOREM

[Gauss 1813]



$$\int_{\partial D} \vec{f} \cdot \vec{n}$$

$$\Longleftrightarrow$$



$$\int_{D} \nabla \cdot \vec{f}$$

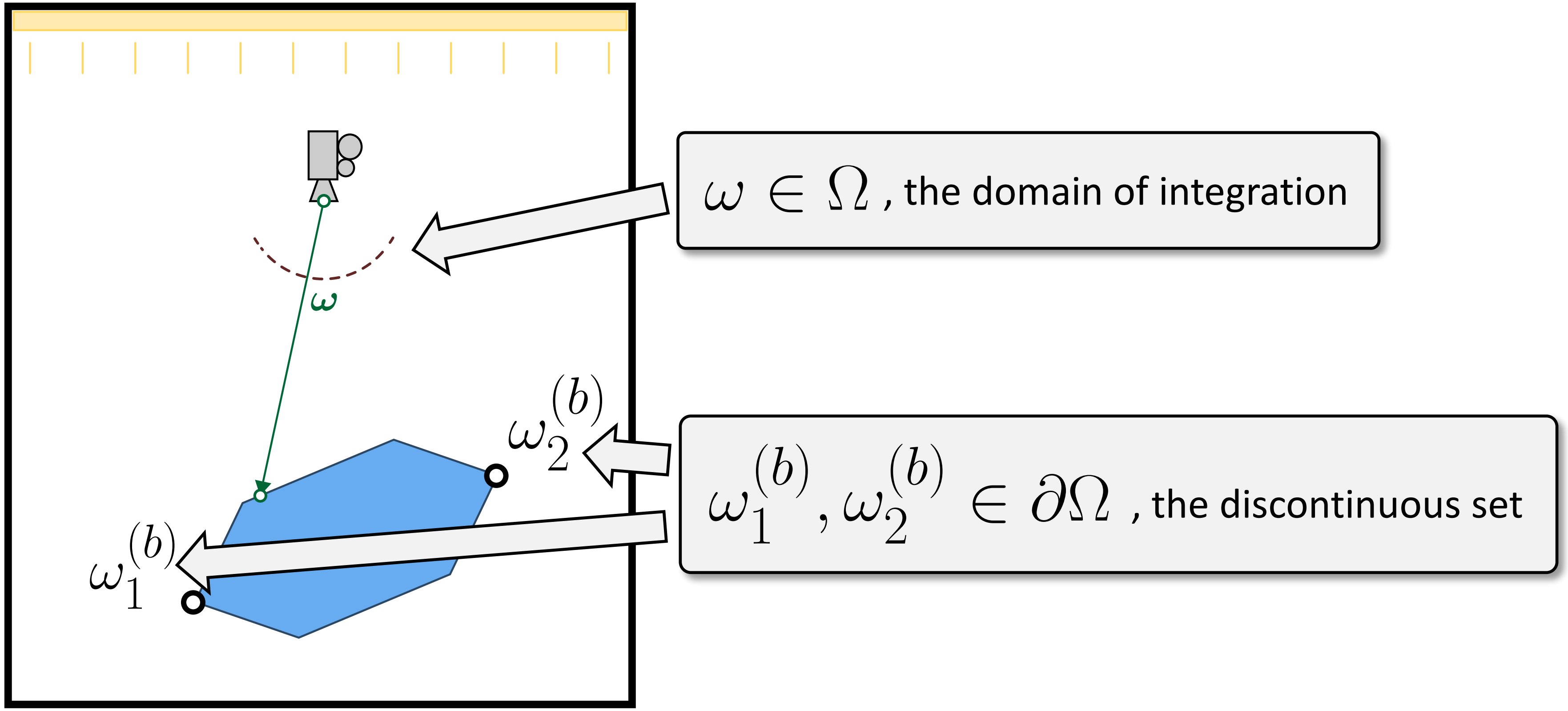
QUICK RECAP

- Used *Reynolds transport theorem* to find the boundary integral $\boxed{\int_{\partial \mathrm{D}} f \vec{\mathbf{v}} \cdot \vec{\mathbf{n}}}$

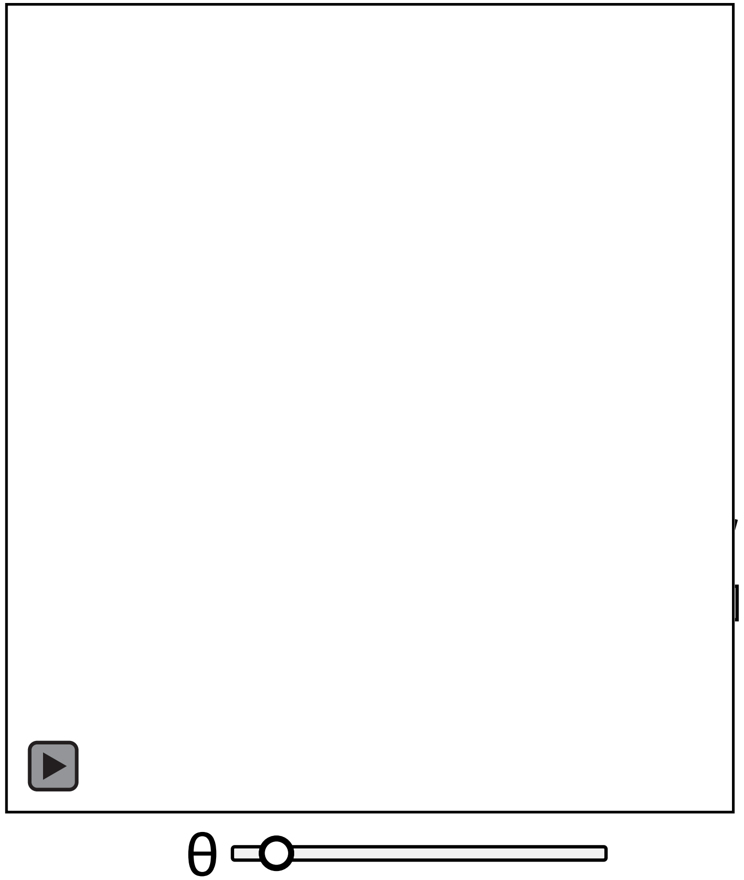
- Rewrote $\boxed{\int_{\partial \mathrm{D}} f \vec{\mathbf{v}} \cdot \vec{\mathbf{n}}}$ to $\boxed{\int_{\mathrm{D}} \nabla \cdot (\vec{\mathcal{V}}_\theta f)}$ using the *divergence theorem*.

- Have to define the *vector field* $\vec{\mathcal{V}}_\theta$ over domain D

A 2D EXAMPLE SCENE

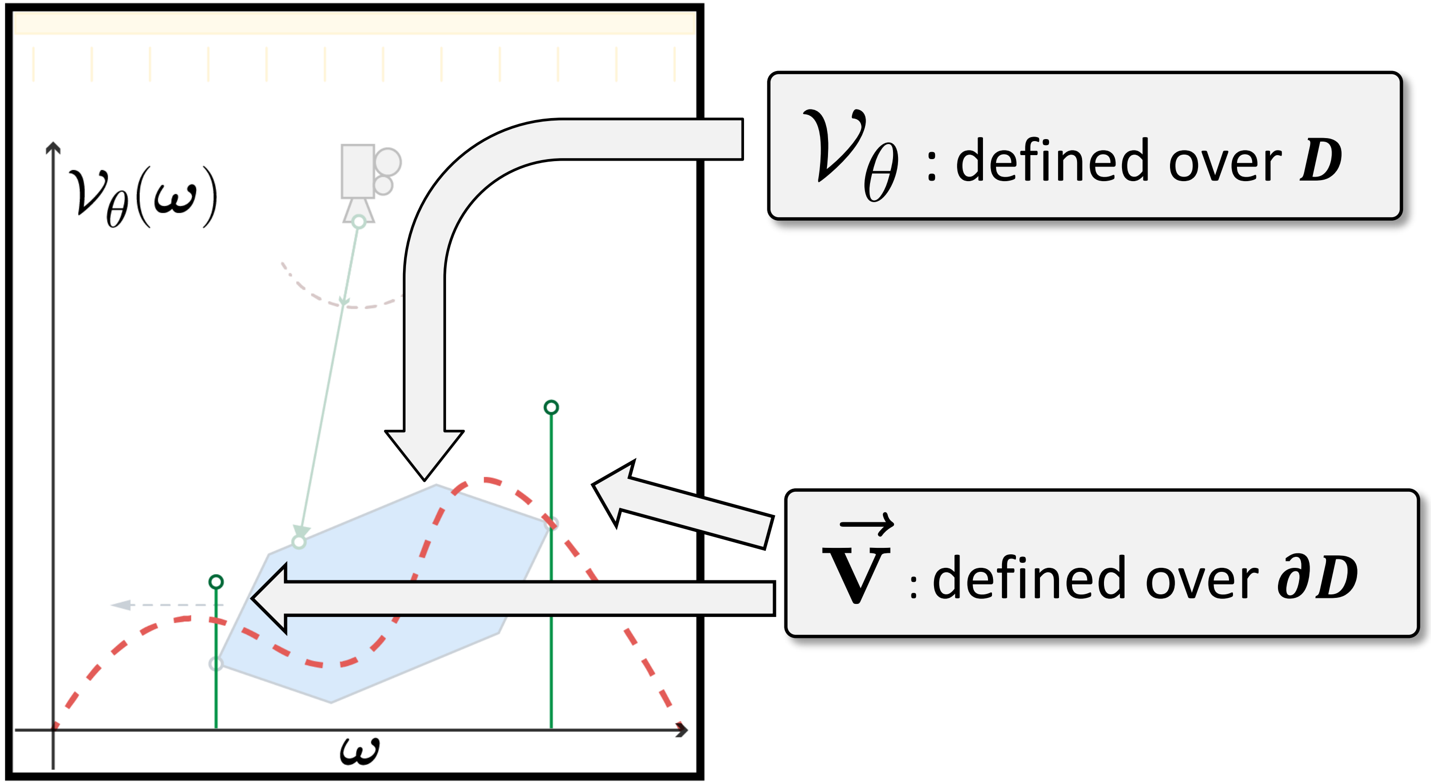


VELOCITY $\vec{\mathbf{V}}$: THE BOUNDARY DERIVATIVE



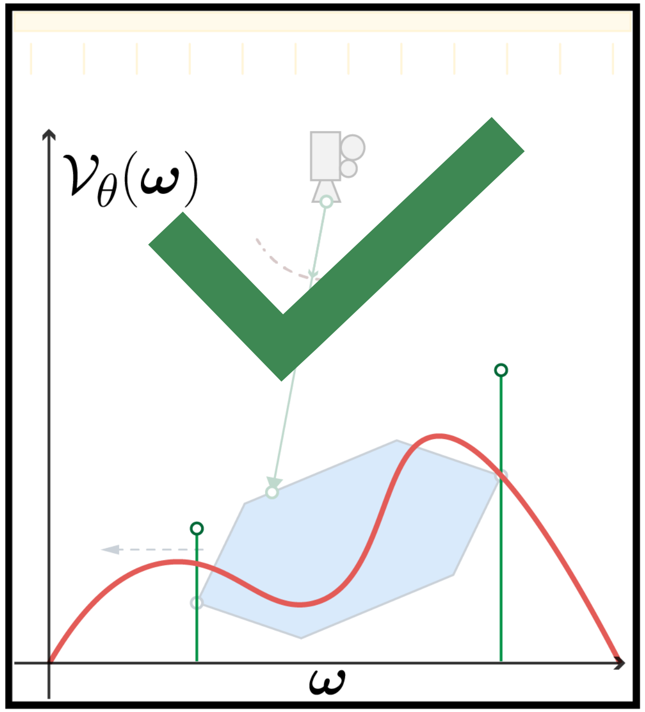
$\partial_\theta \omega_i^{(b)}$: Derivative of boundary position w.r.t **θ**

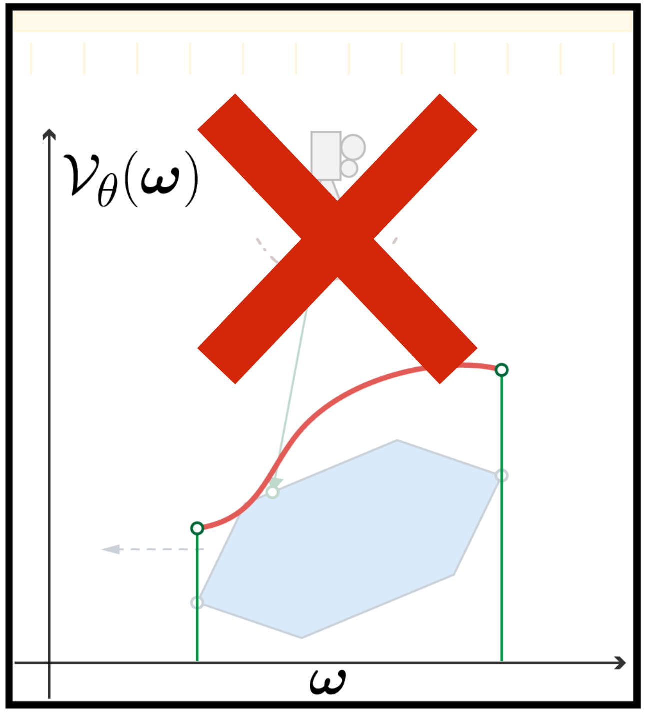
WARP FIELD $\mathcal{V}_\theta$: EXTENSION OF $\vec{\mathbf{V}}$ TO ALL POINTS



VALIDITY OF $\vec{\mathcal{V}}_\theta$

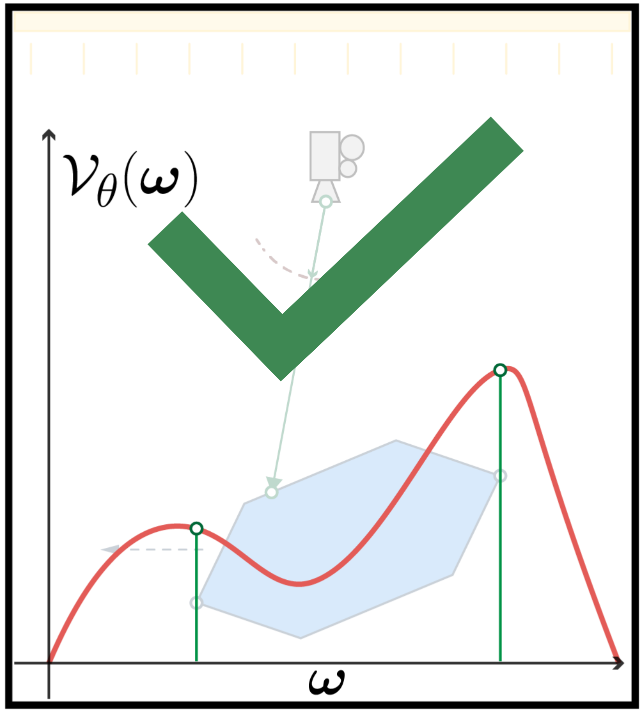
Rule 1: Continuous

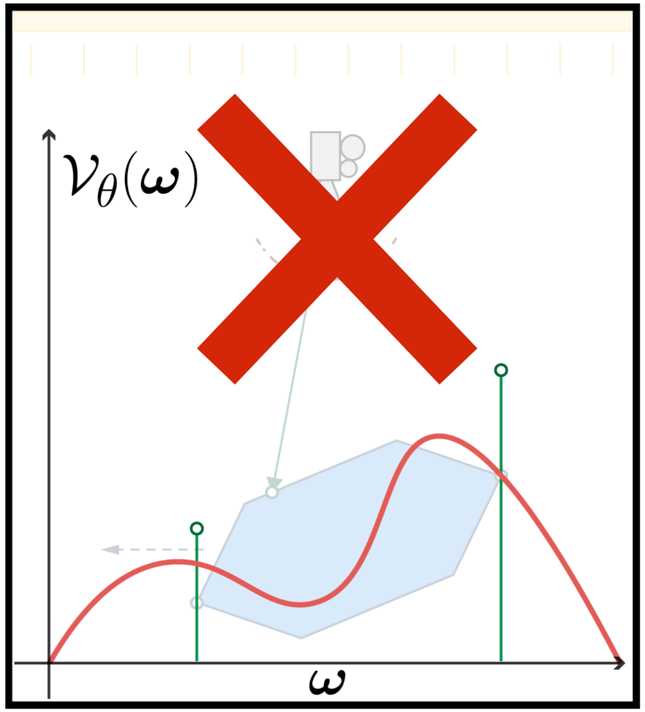




VALIDITY OF $\vec{\mathcal{V}}_\theta$

Rule 2: Boundary Consistent



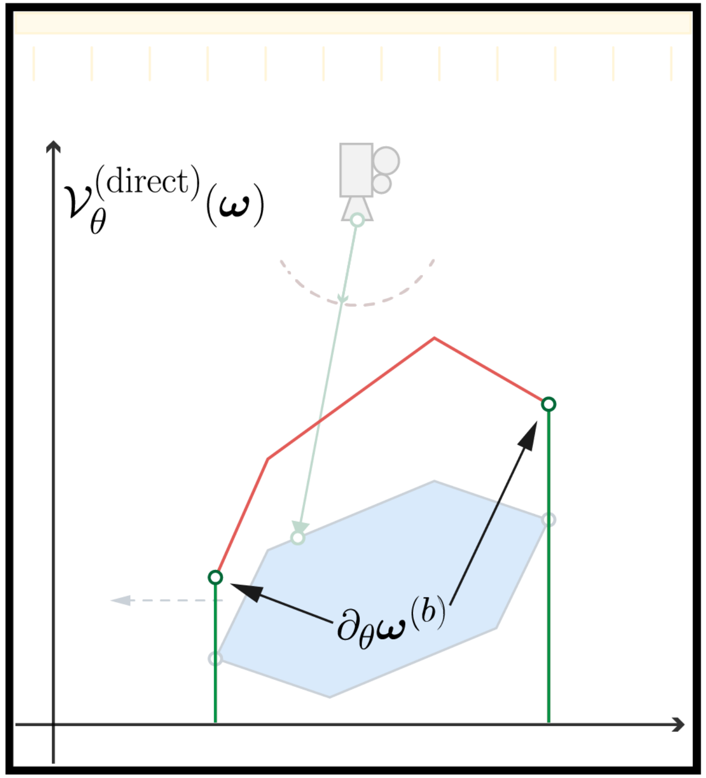


CONSTRUCTING $\vec{\mathcal{V}}_\theta$

Attempt 1 ➡ Find $\partial_\theta \boldsymbol{\omega}$ through *implicit derivative*

(Incorrect)

$$\mathbf{y} = \textsc{Intersect}(\boldsymbol{\omega}, \theta) \implies \partial_\theta \boldsymbol{\omega} = \frac{\partial_{\boldsymbol{\omega}} \mathbf{y}}{\partial_\theta \mathbf{y}}$$



At all points (not just boundaries)

+ Boundary consistent

- Not continuous

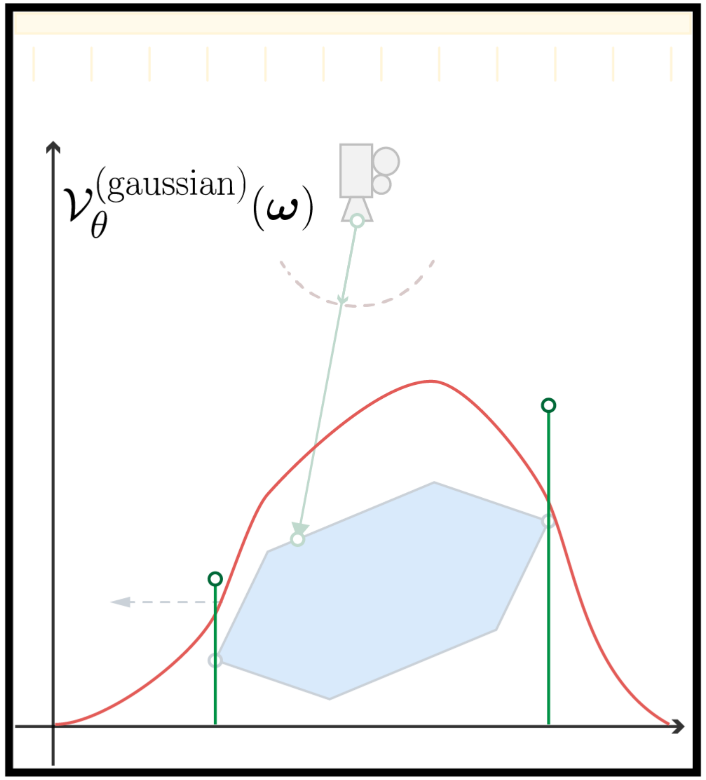
CONSTRUCTING $\vec{\mathcal{V}}_\theta$

Attempt 2 ⟶ Filter *Attempt 1* with a Gaussian filter

(Incorrect)

$$\int_{\Omega'} k(\boldsymbol{\omega}, \boldsymbol{\omega}') \frac{\partial_{\boldsymbol{\omega}} \mathbf{y}}{\partial_{\theta} \mathbf{y}}$$

k(.,.) = Gaussian filter



+ Continuous

- Not boundary consistent

BOUNDARY-AWARE WEIGHTING

Goal: Find weights $k(\boldsymbol{\omega}, \boldsymbol{\omega}')$ s.t. $\vec{\mathcal{V}}_\theta = \frac{\partial_\omega \mathbf{y}}{\partial_\theta \mathbf{y}}$ at boundaries.

Ideal weighting function

$\boldsymbol{\omega}$

Approach Dirac delta near boundaries

PATH-INTEGRAL FOR DIFFERENTIABLE RENDERING

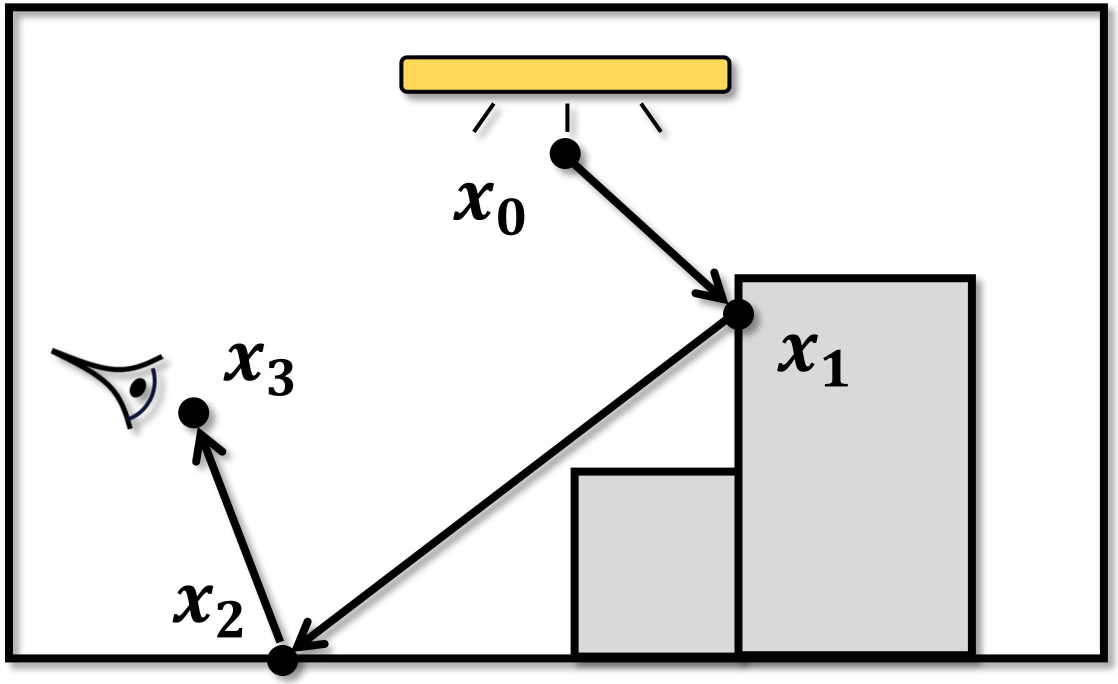
FORWARD PATH INTEGRAL

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

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

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

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

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,

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

and

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

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):

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

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

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

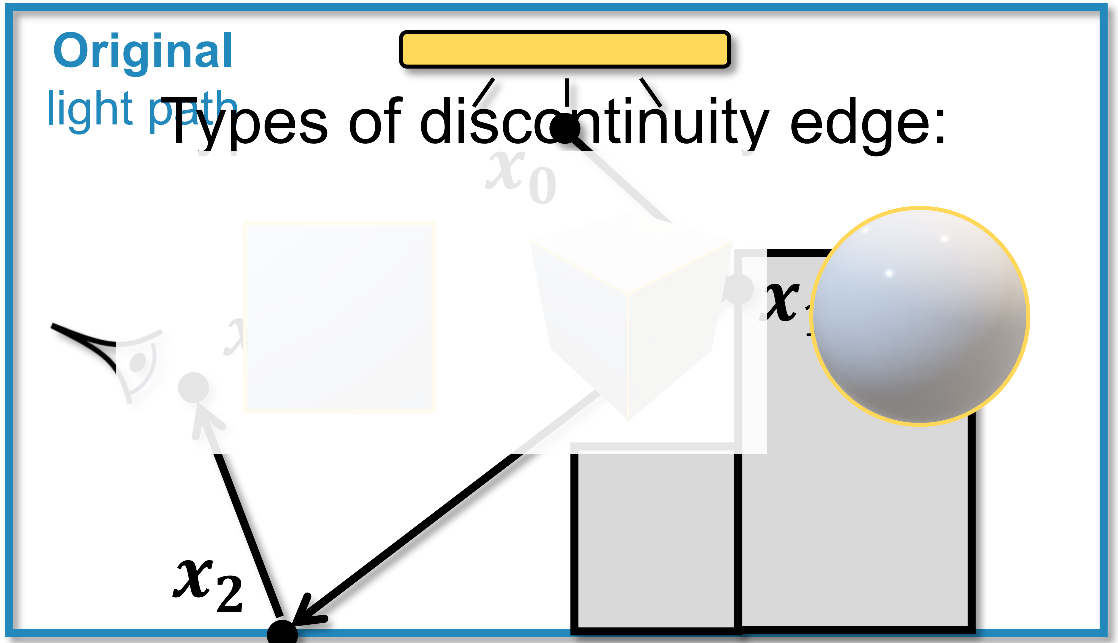
path space

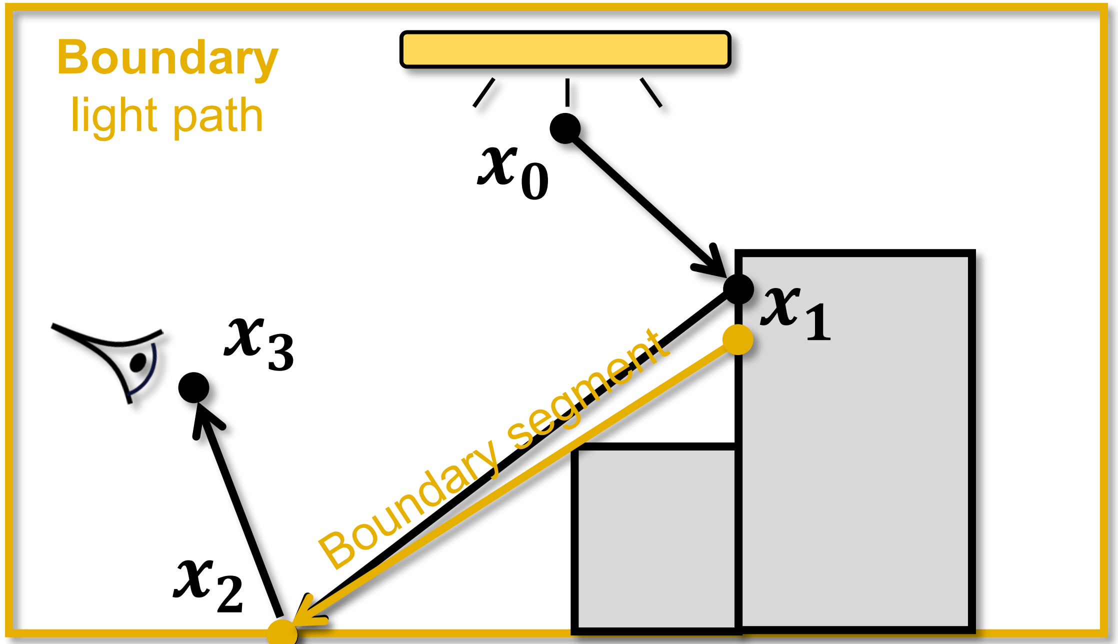
Interior integral

boundary path space

Boundary integral

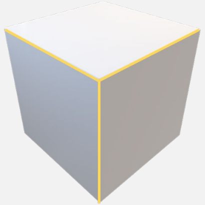
Types of discontinuity edge:



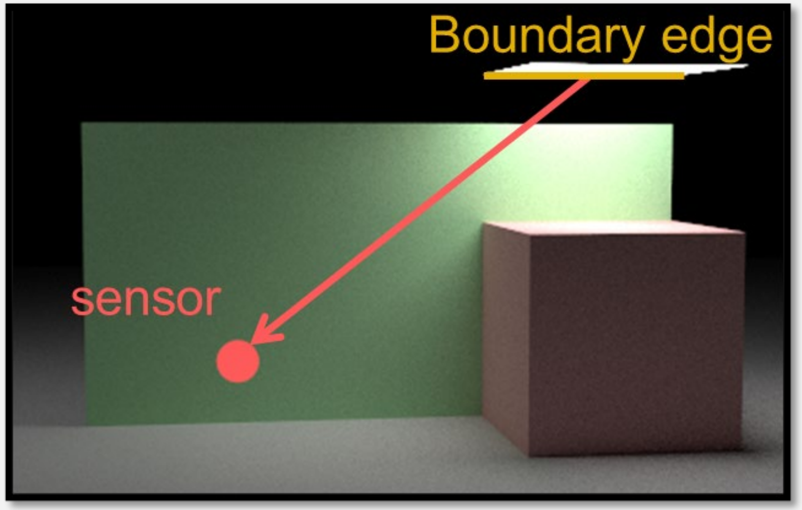


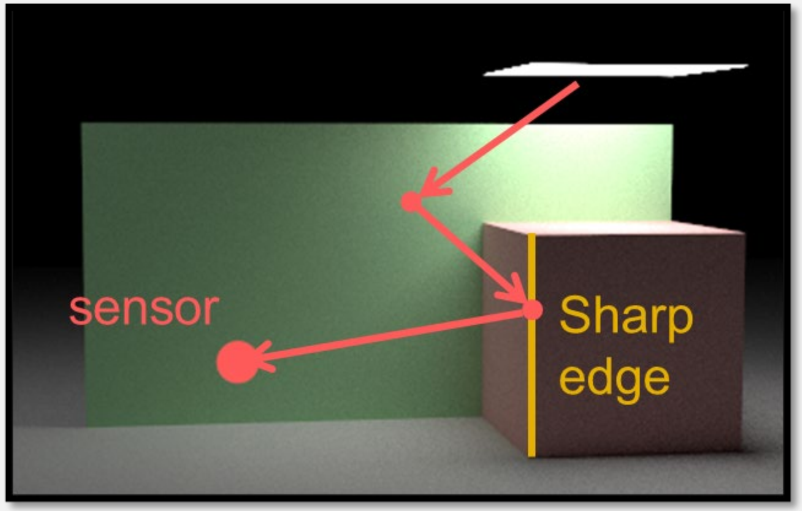
SOURCE OF DISCONTINUITIES

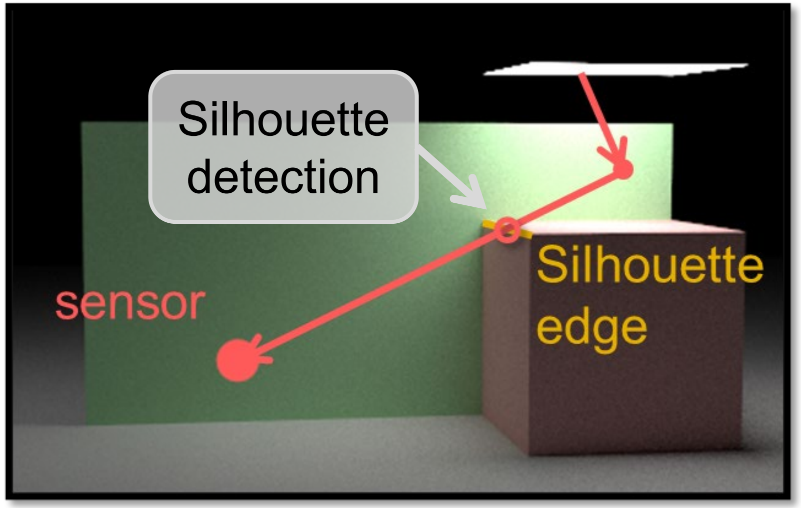
Boundary edge 

Sharp edge 

Silhouette edge 







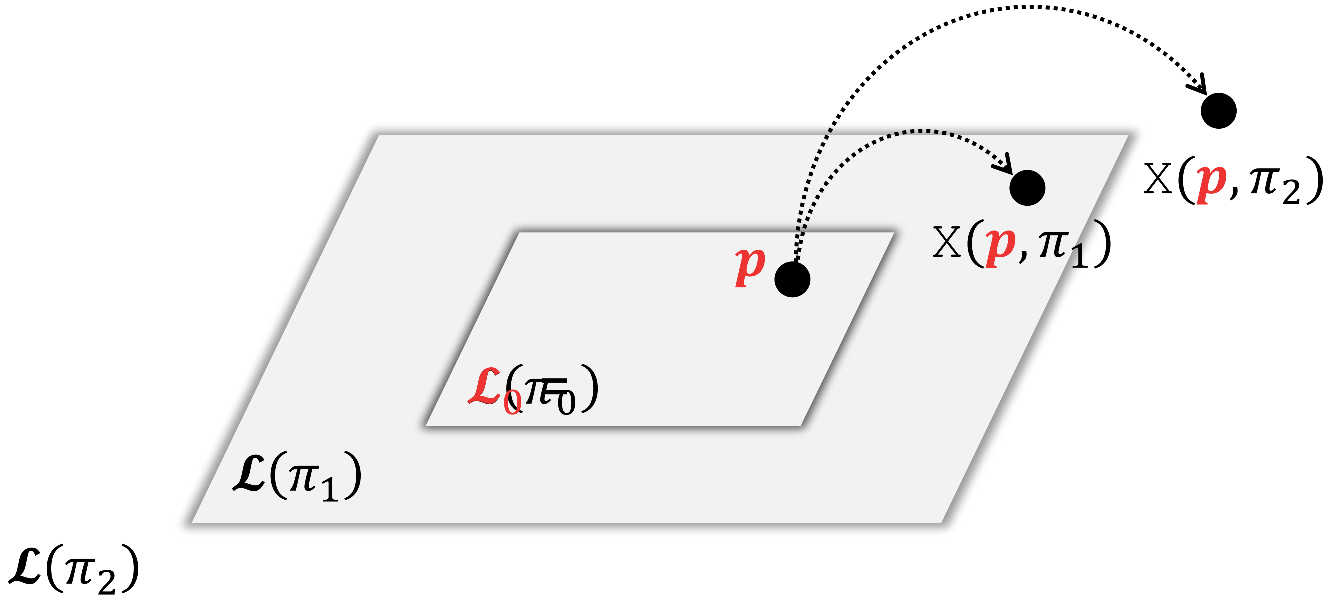
Topology-driven

Visibility-driven

TEXTURE PARAMETERIZATION FOR SIMPLIFYING THE BOUNDARY TERM

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

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

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

REPARAMETERIZATION

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

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

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

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

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

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 X
Pro: Fewer types of discontinuities

DIFFERENTIAL PATH INTEGRAL

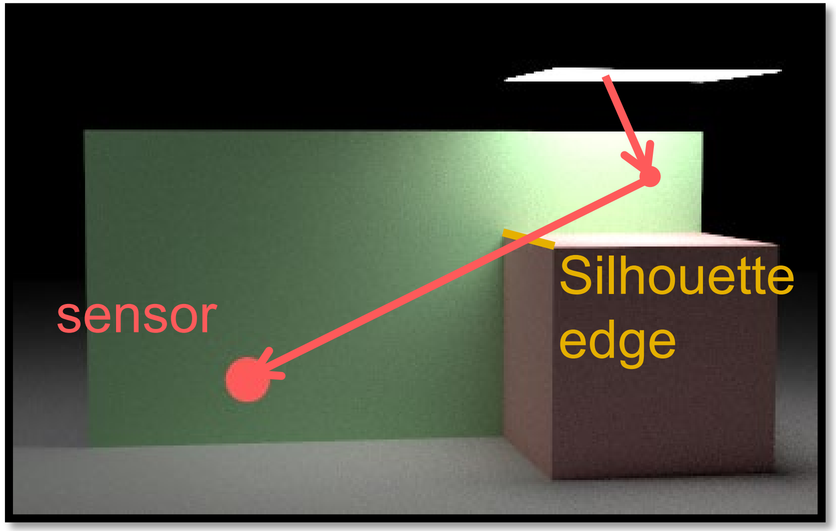
Differential path integral

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

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

Topology-driven

Visibility-driven



MONTE CARLO ESTIMATORS

ESTIMATING INTERIOR INTEGRAL

(Reparameterized)
Differential path Integral

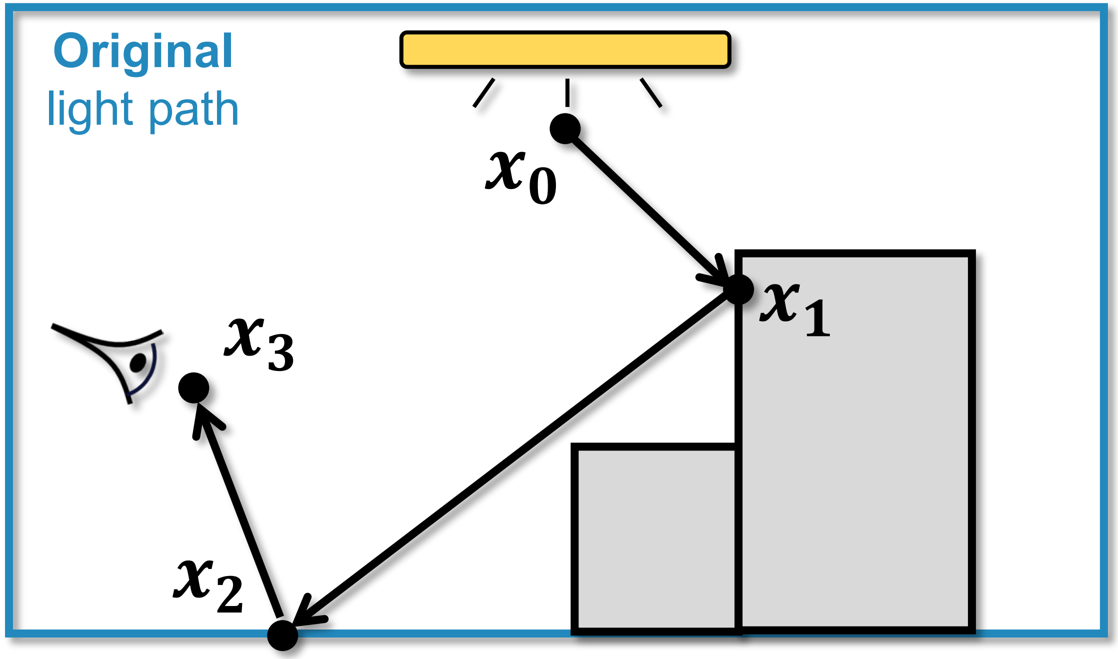
$$\frac{\partial I}{\partial \pi} = \int_{\Omega_0} \frac{\partial}{\partial \pi}\left(f(\overline{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\overline{\boldsymbol{x}})}{\mathrm{d}\mu(\overline{\boldsymbol{p}})} \right| \right) \mathrm{d}\mu(\overline{\boldsymbol{p}}) + \int_{\partial\Omega_0} g(\overline{\boldsymbol{p}})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

Interior integral

Boundary integral

Different MC estimators

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

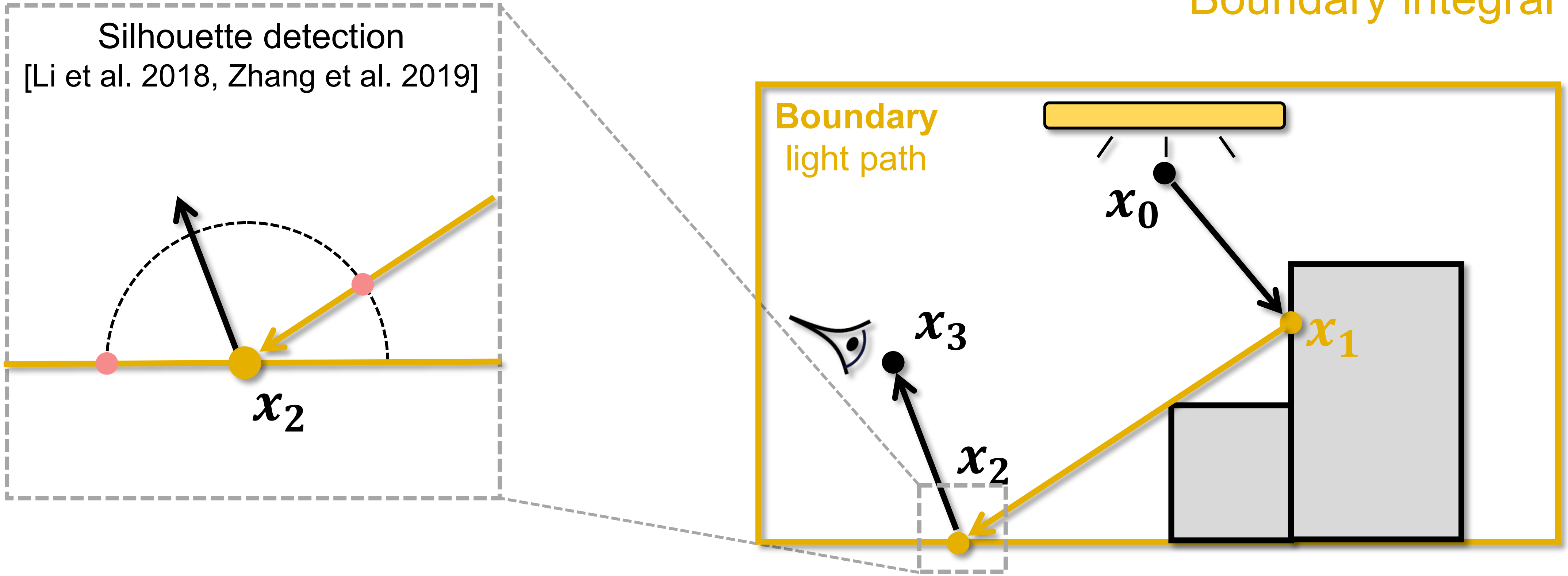


ESTIMATING BOUNDARY INTEGRAL

(Reparameterized)
Differential path Integral

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

Boundary integral



ESTIMATING BOUNDARY INTEGRAL

(Reparameterized)
Differential path Integral

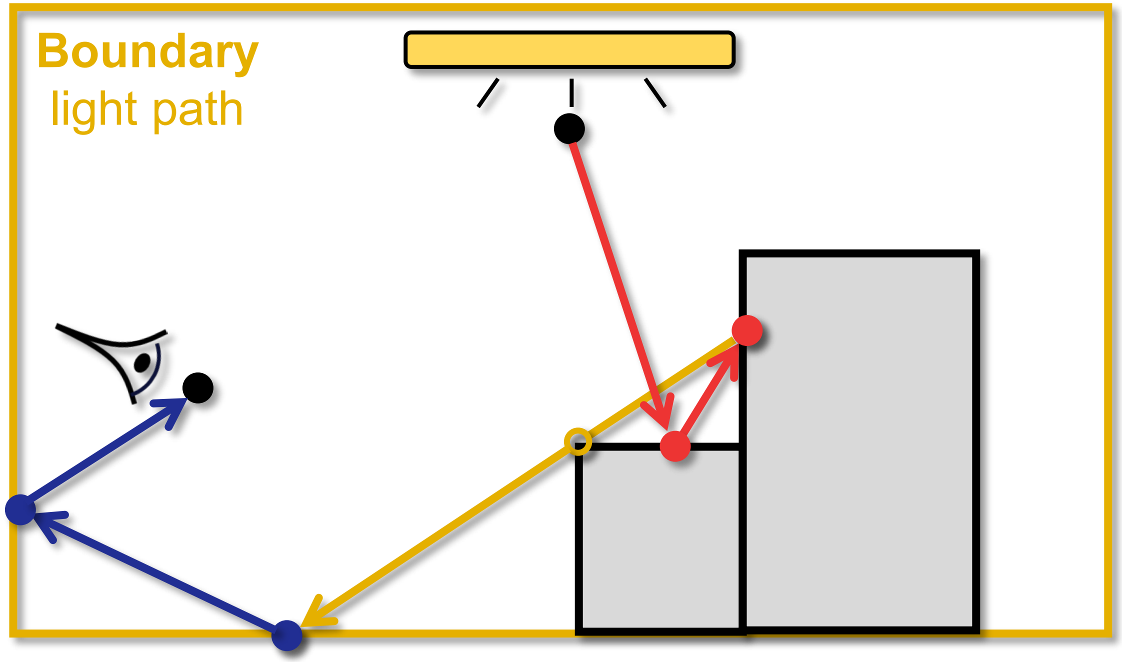
$$\frac{\partial I}{\partial \pi} = \int_{\Omega_0} \frac{\partial}{\partial \pi}\left(f(\overline{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\overline{\boldsymbol{x}})}{\mathrm{d}\mu(\overline{\boldsymbol{p}})} \right| \right) \mathrm{d}\mu(\overline{\boldsymbol{p}}) + \boxed{\int_{\partial\Omega_0}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

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

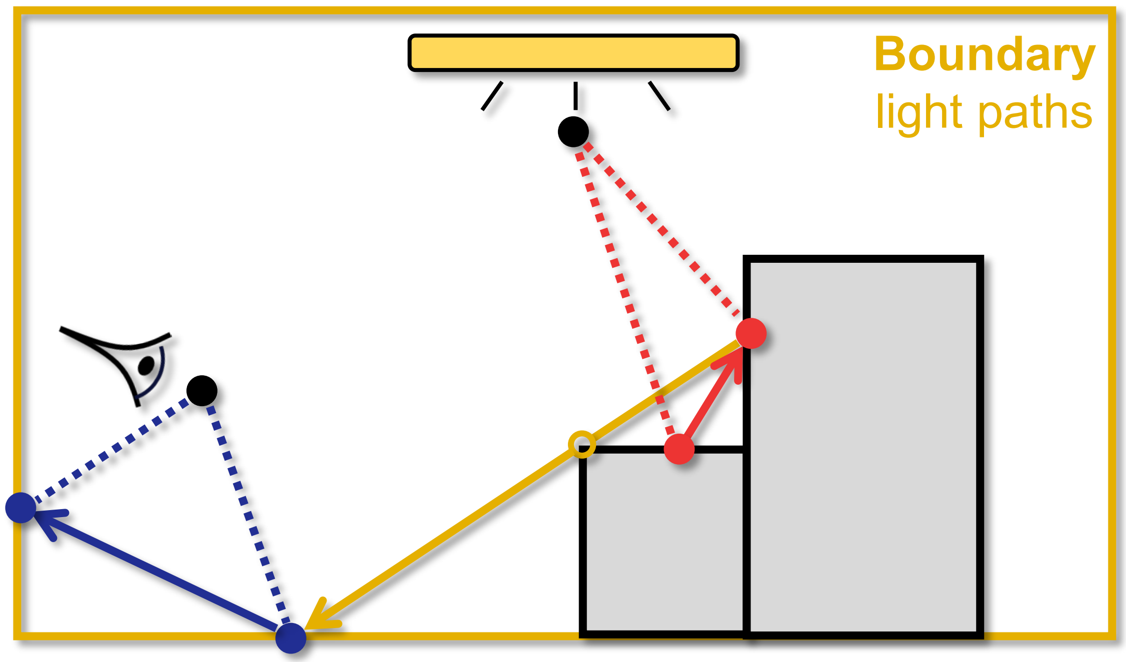
Interior: **unidirectional** path tracing

Boundary: **unidirectional** sampling of subpaths

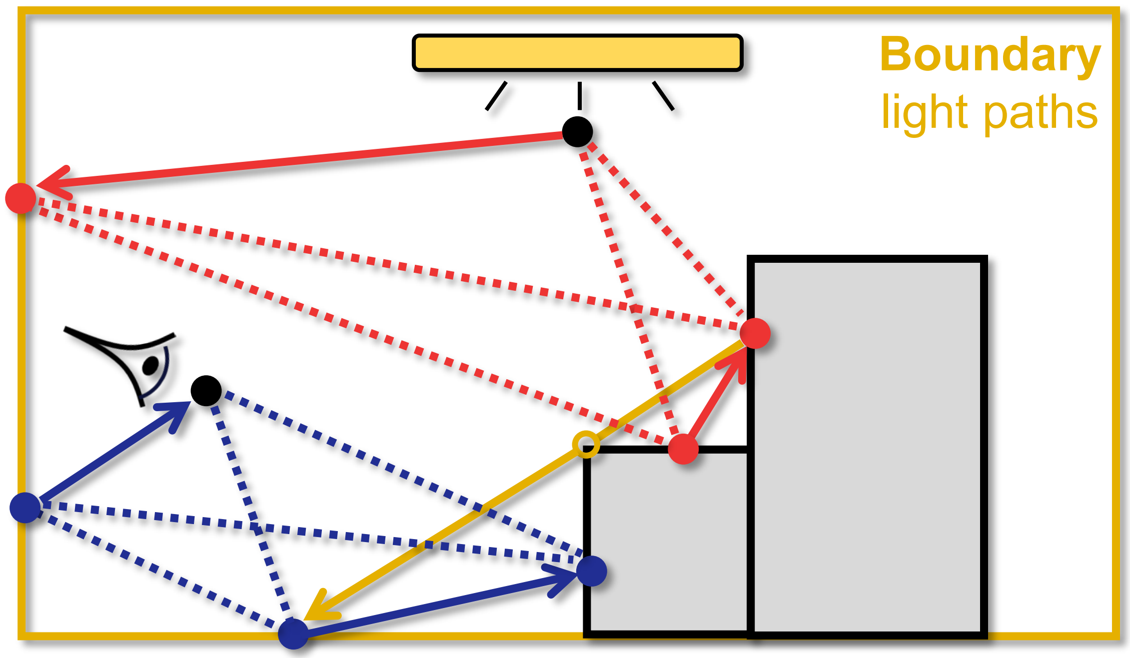
Bidirectional estimator

Interior: **bidirectional** path tracing

Boundary: **bidirectional** sampling of subpaths



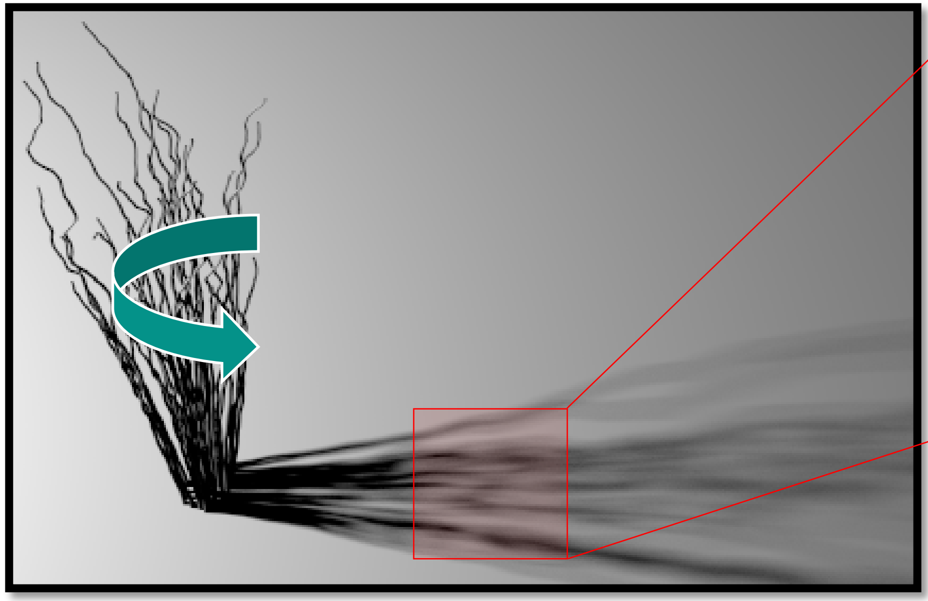
Unidirectional path tracing + NEE



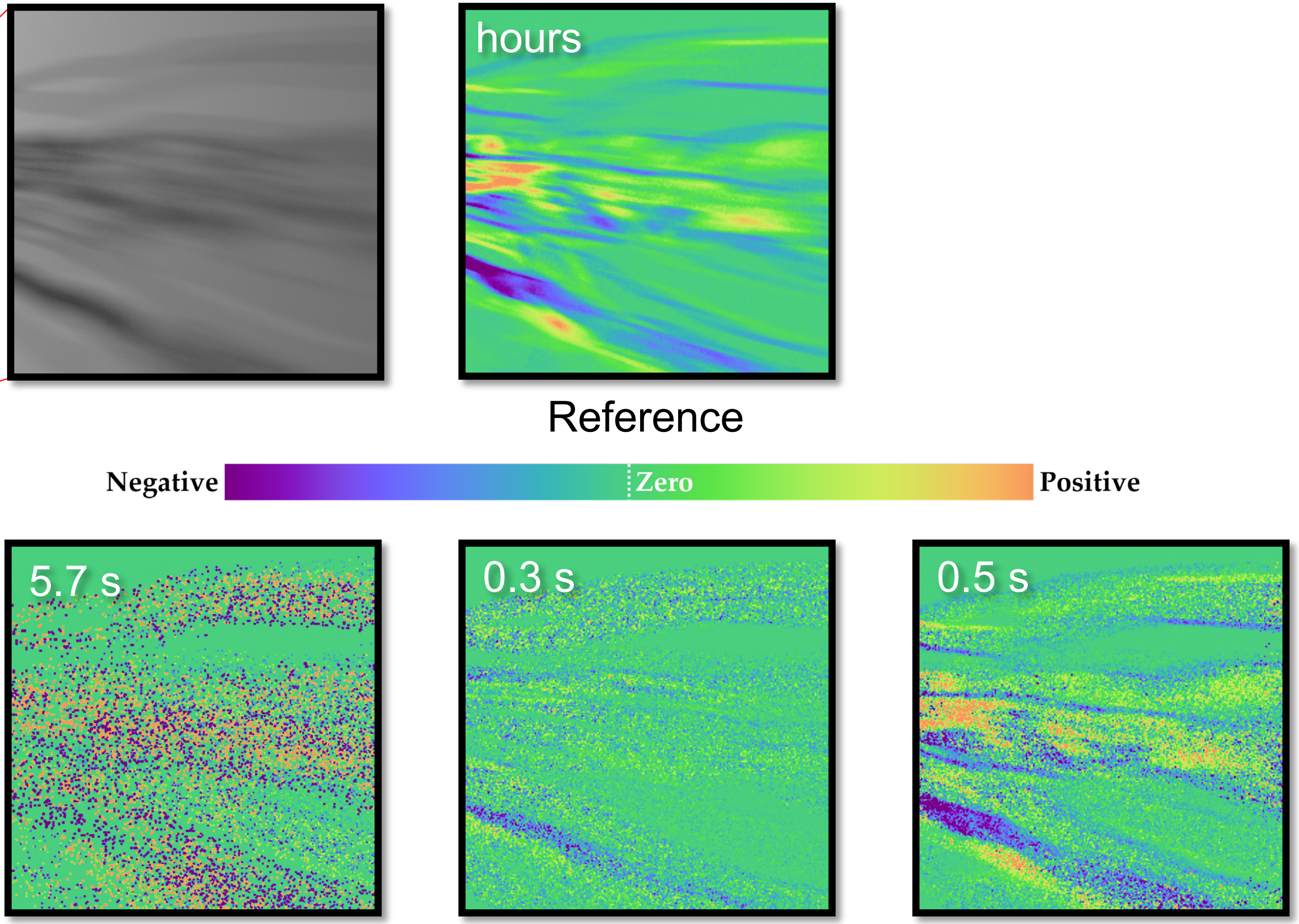
Bidirectional path tracing

SOME RESULTS

HANDLING COMPLEX GEOMETRY



Complex geometry



Reference

Equal-sample
comparison

[Zhang et al. 2019]

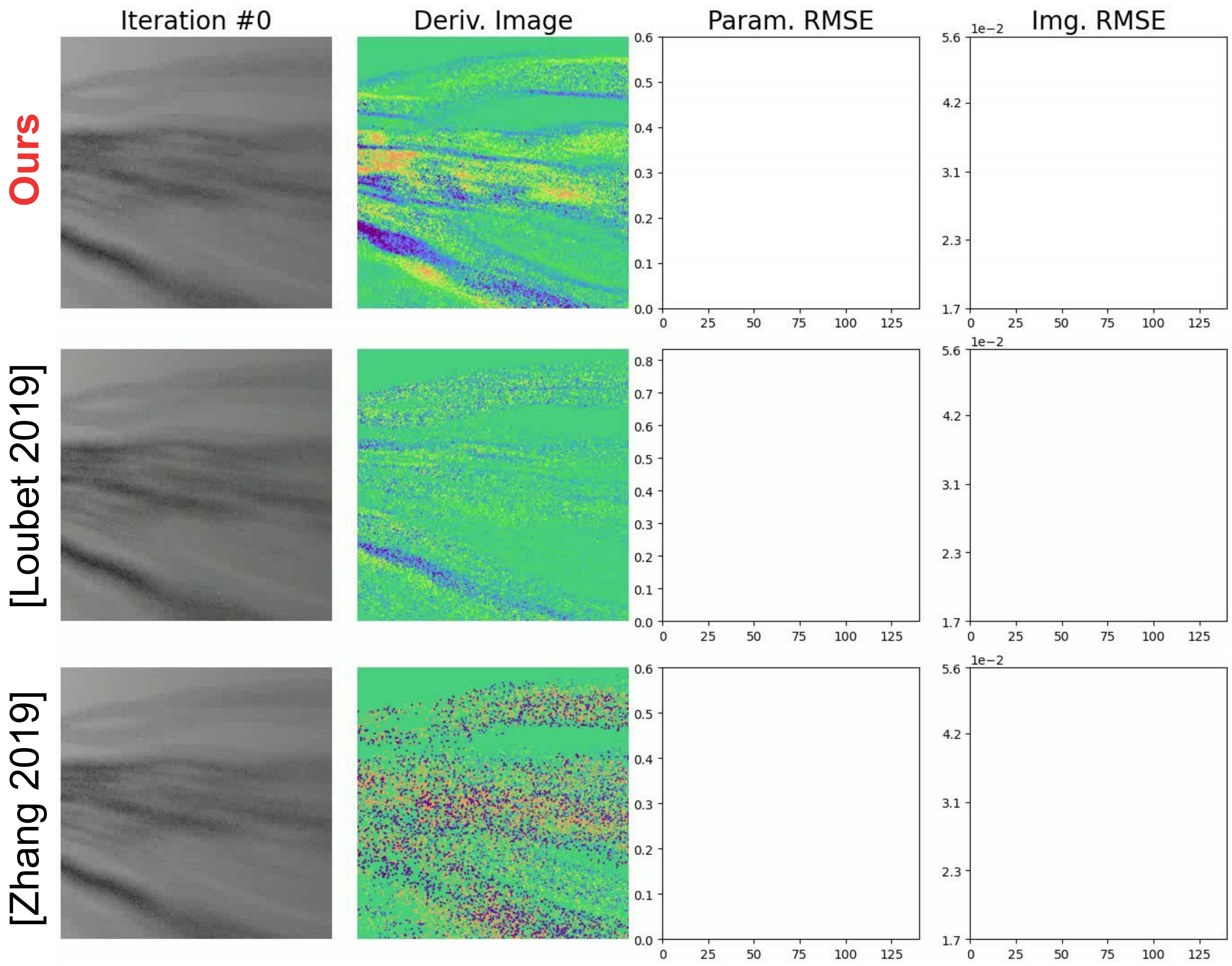
[Loubet et al. 2019]

Ours

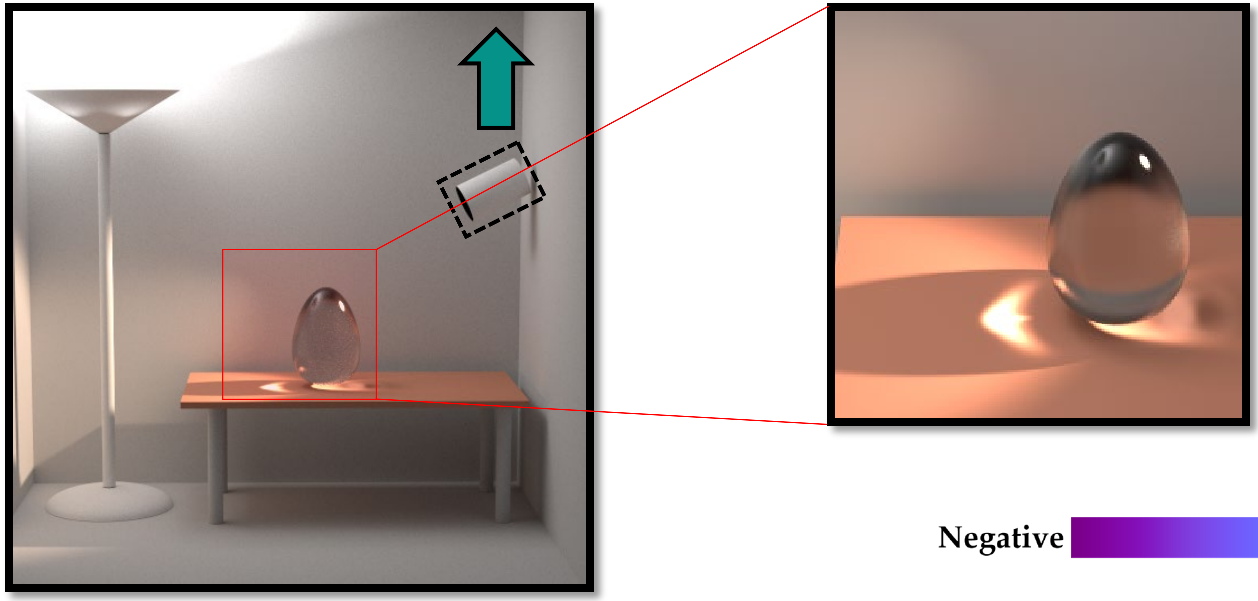
HANDLING COMPLEX GEOMETRY

Target image

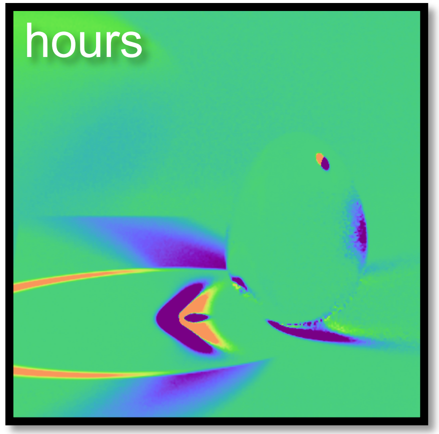
- Optimizing *rotation angle*
- **Equal-sample** per iteration
- **Identical** optimization setting
  - Learning rate (Adam)
  - Initializations



HANDLING CAUSTICS

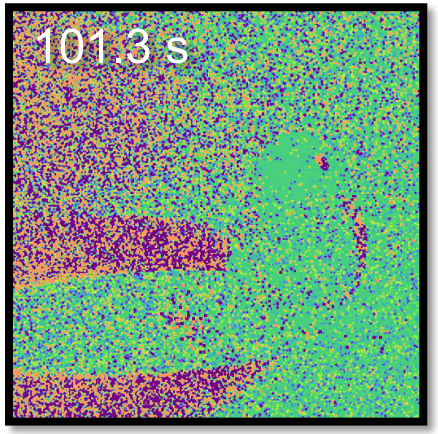


Complex light transport effects

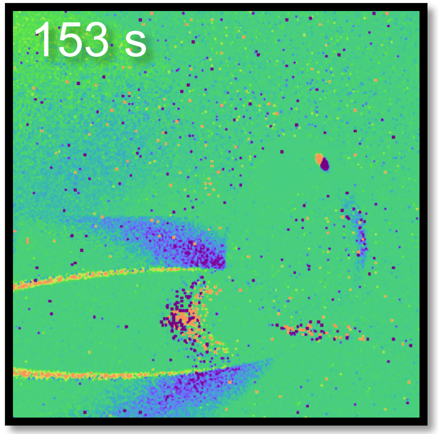


Reference

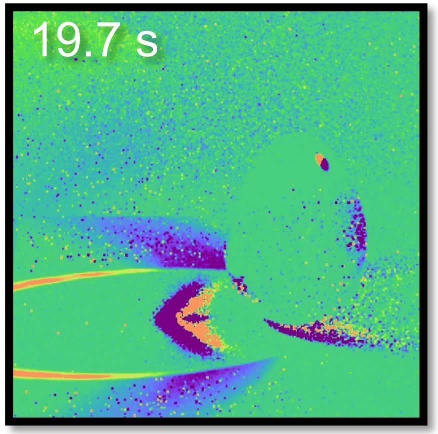
Equal-sample
comparison



[Zhang et al. 2019]

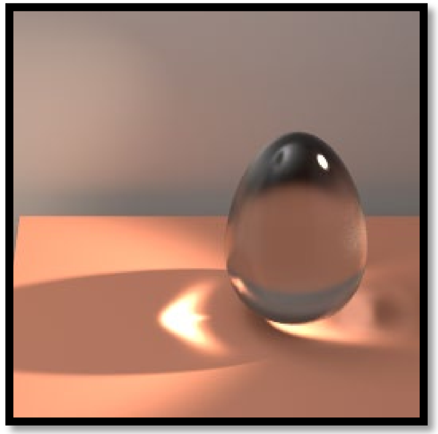


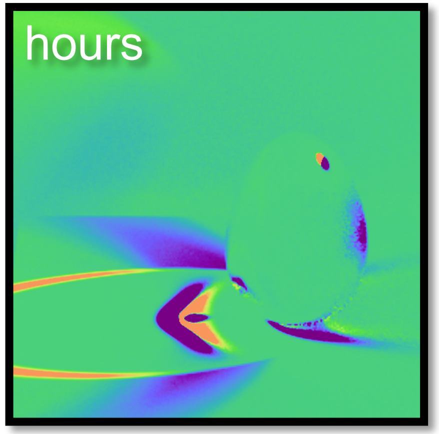
[Loubet et al. 2019]



Ours

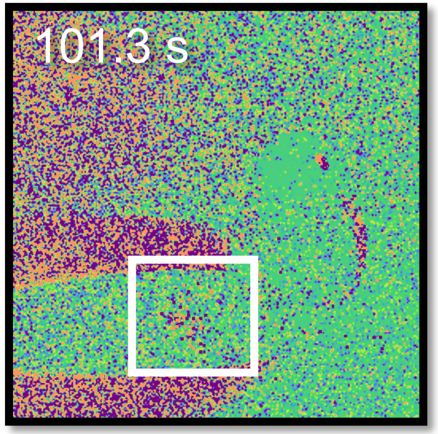
HANDLING CAUSTICS



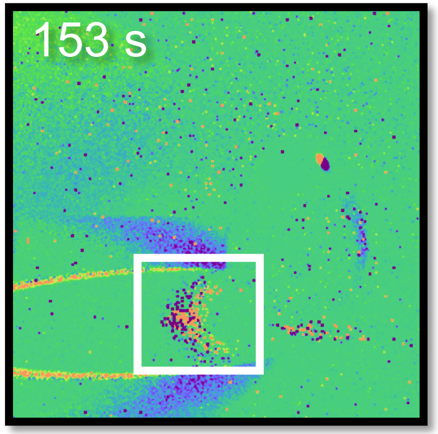


Reference

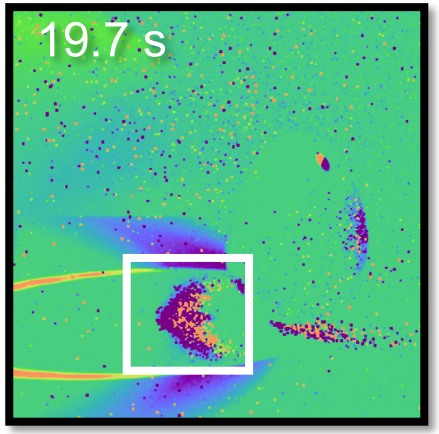
Equal-sample comparison



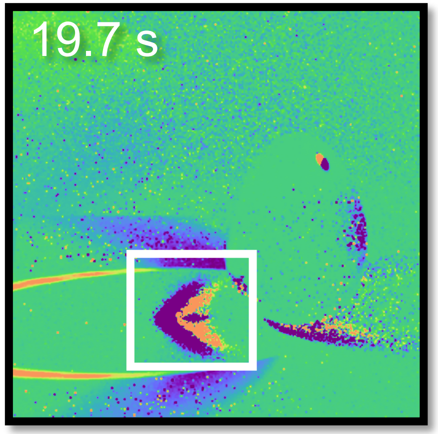
[Zhang et al. 2019]



[Loubet et al. 2019]



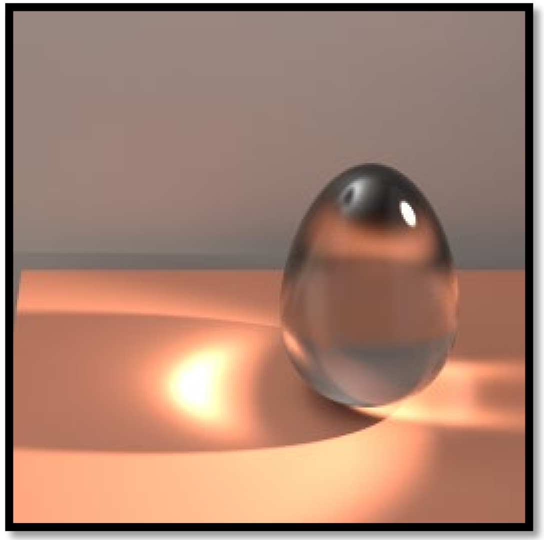
Ours (unidirectional)



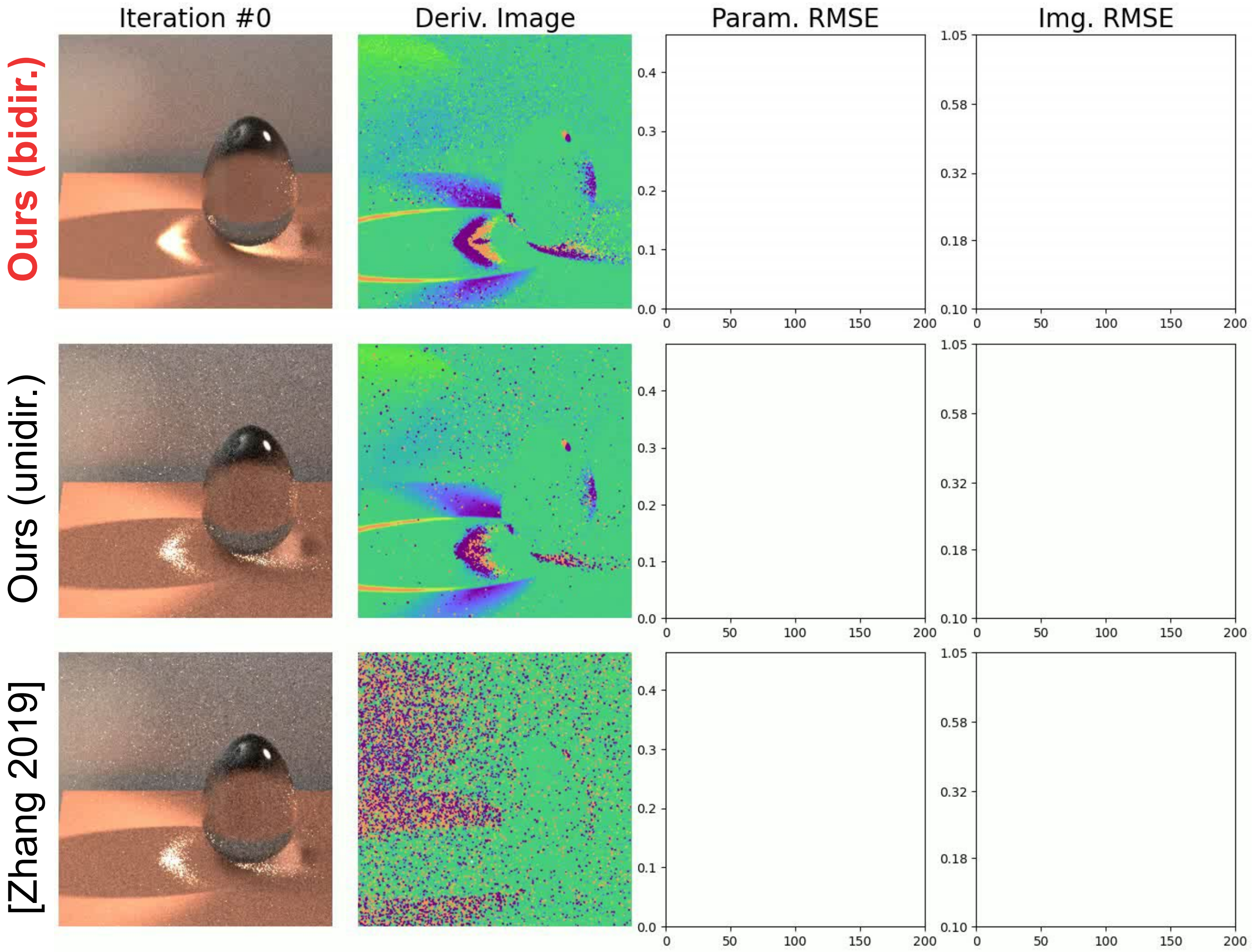
Ours (bidirectional)

HANDLING CAUSTICS

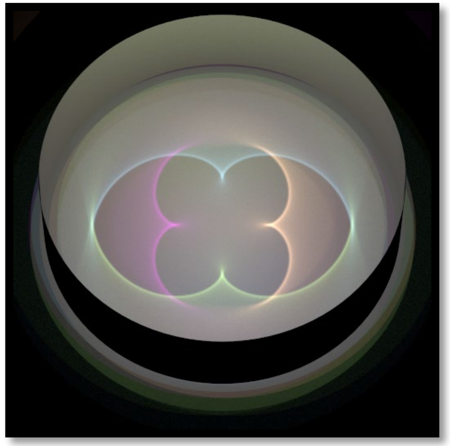
Target image



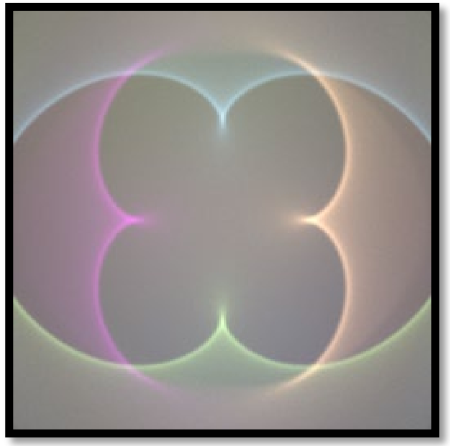
- Optimizing
  - Glass IOR
  - Spotlight position
- **Equal-time** per iteration
- **Identical** optimization setting



SHAPE OPTIMIZATION

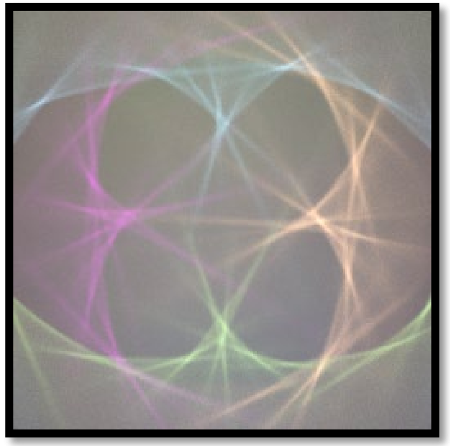


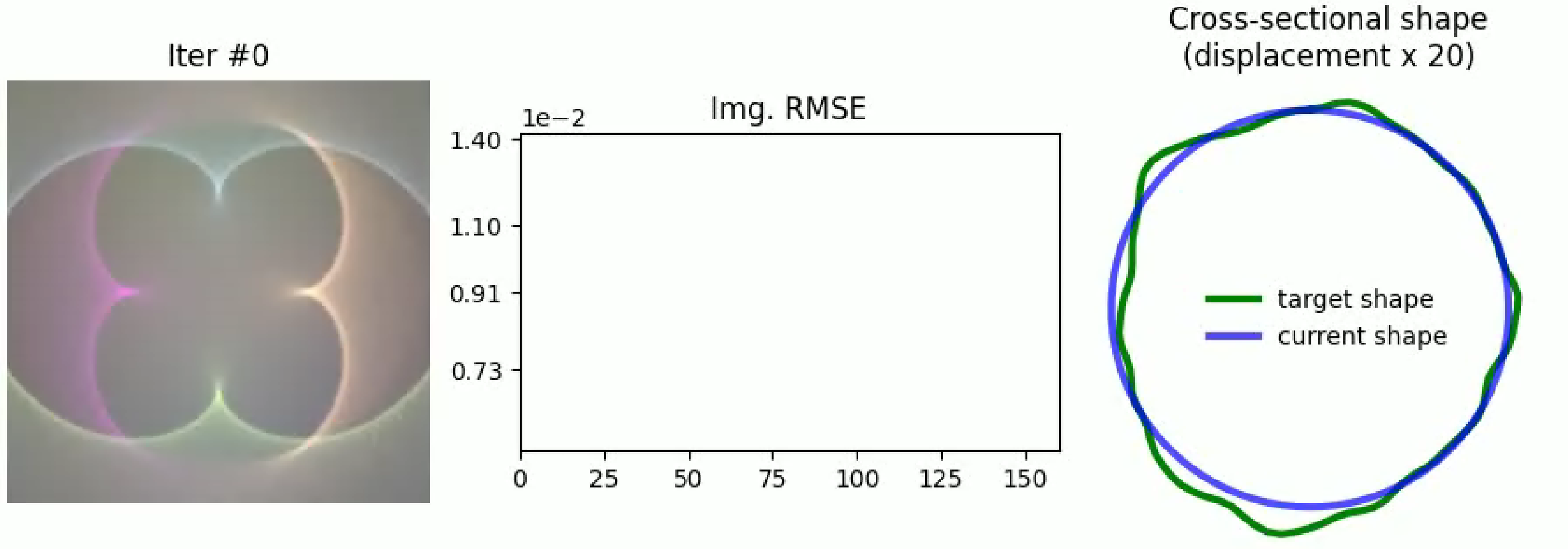
Initial



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

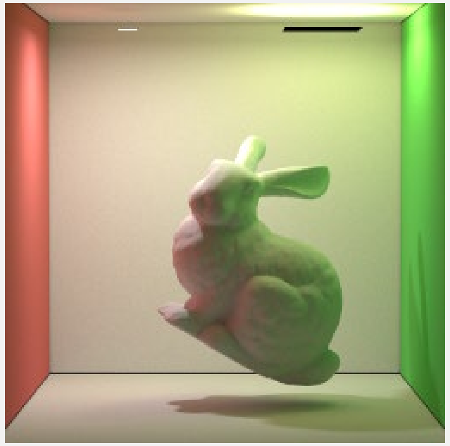
Target image



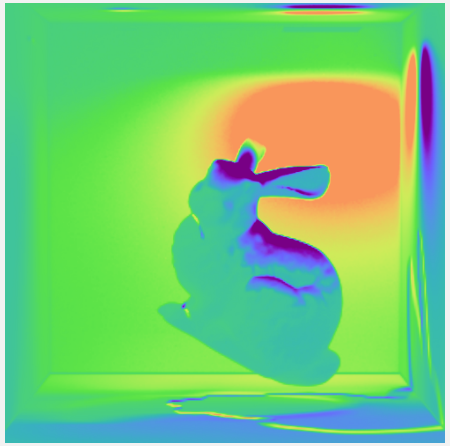


RESULTS

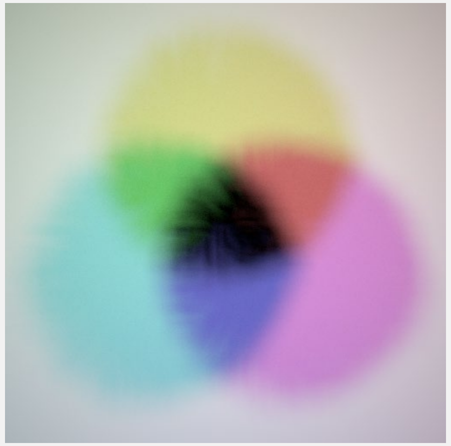
Original image



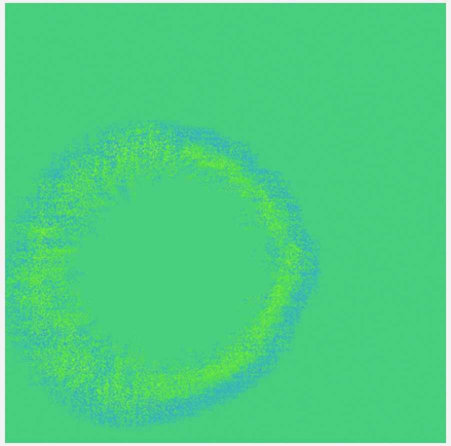
Derivative image



Original image

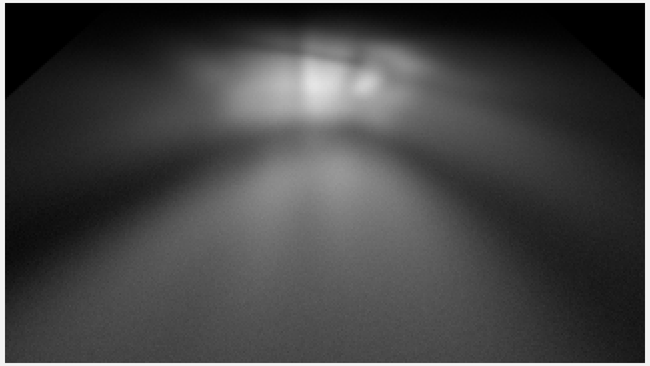


Derivative image

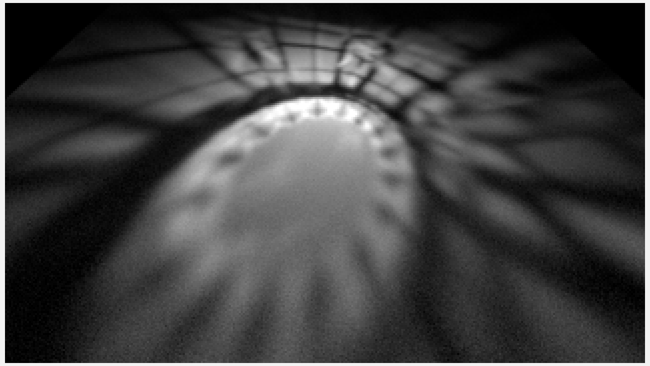


Config.

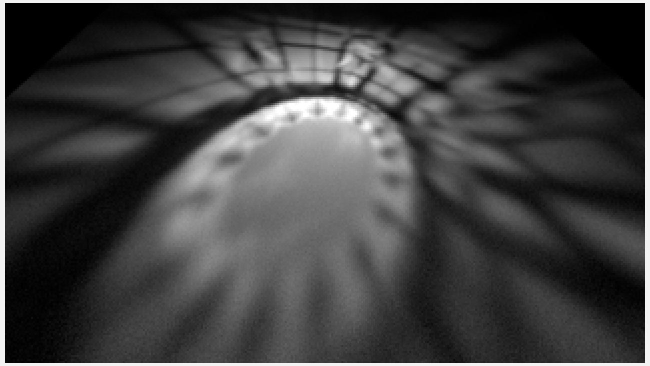
Optimize (initial)



Optimize (final)



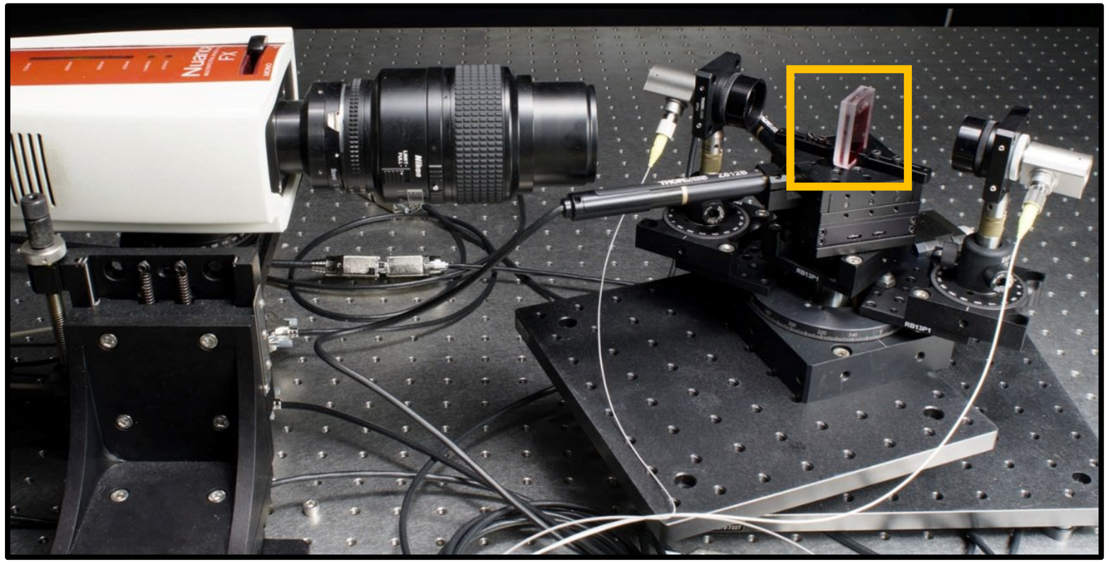
Target

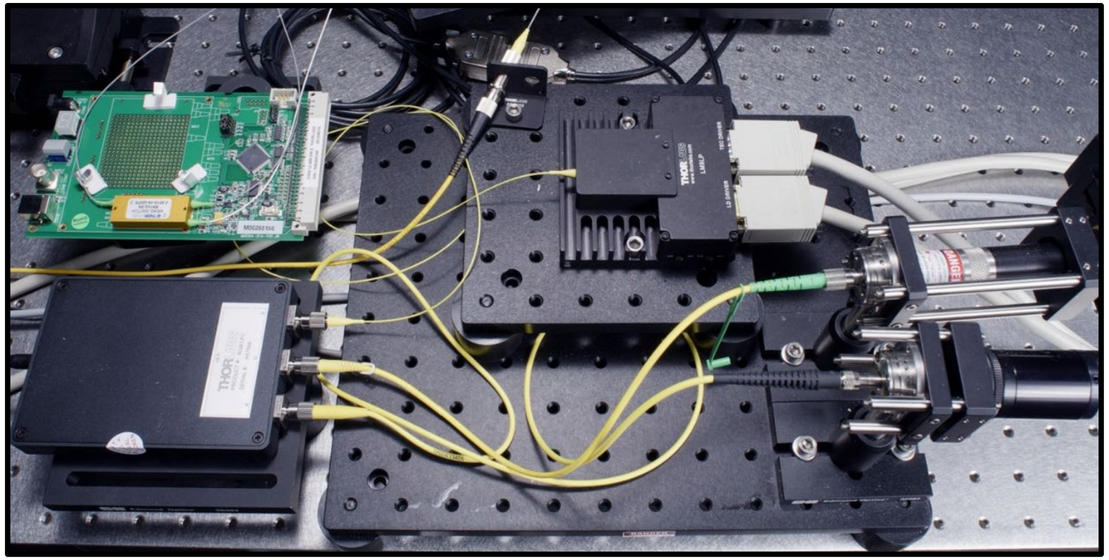


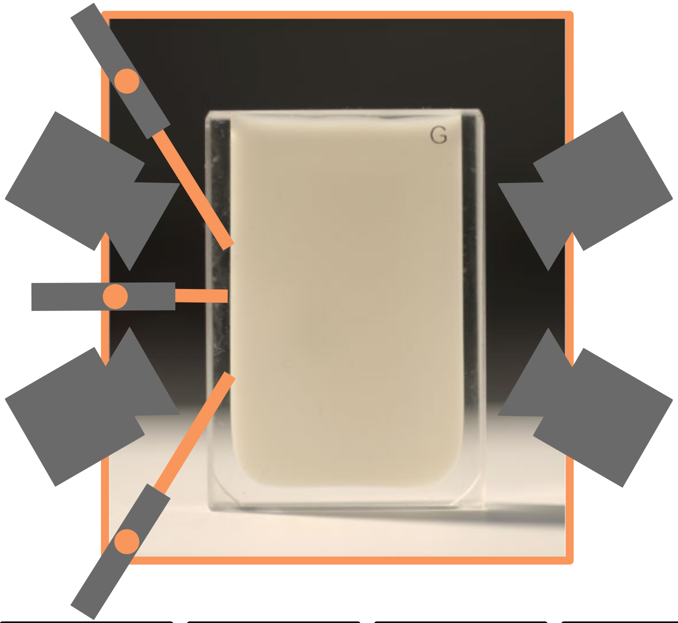
Applications

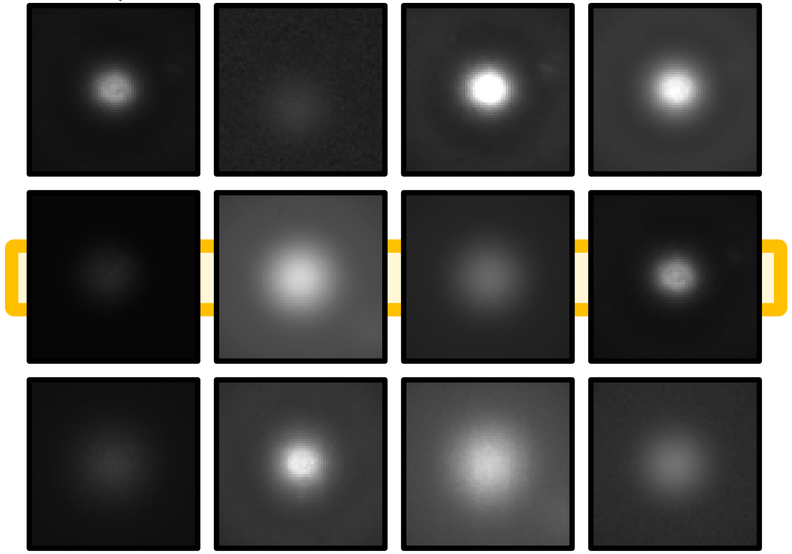
Inverse scattering [Gkioulekas et al. 2013]

Acquisition setup



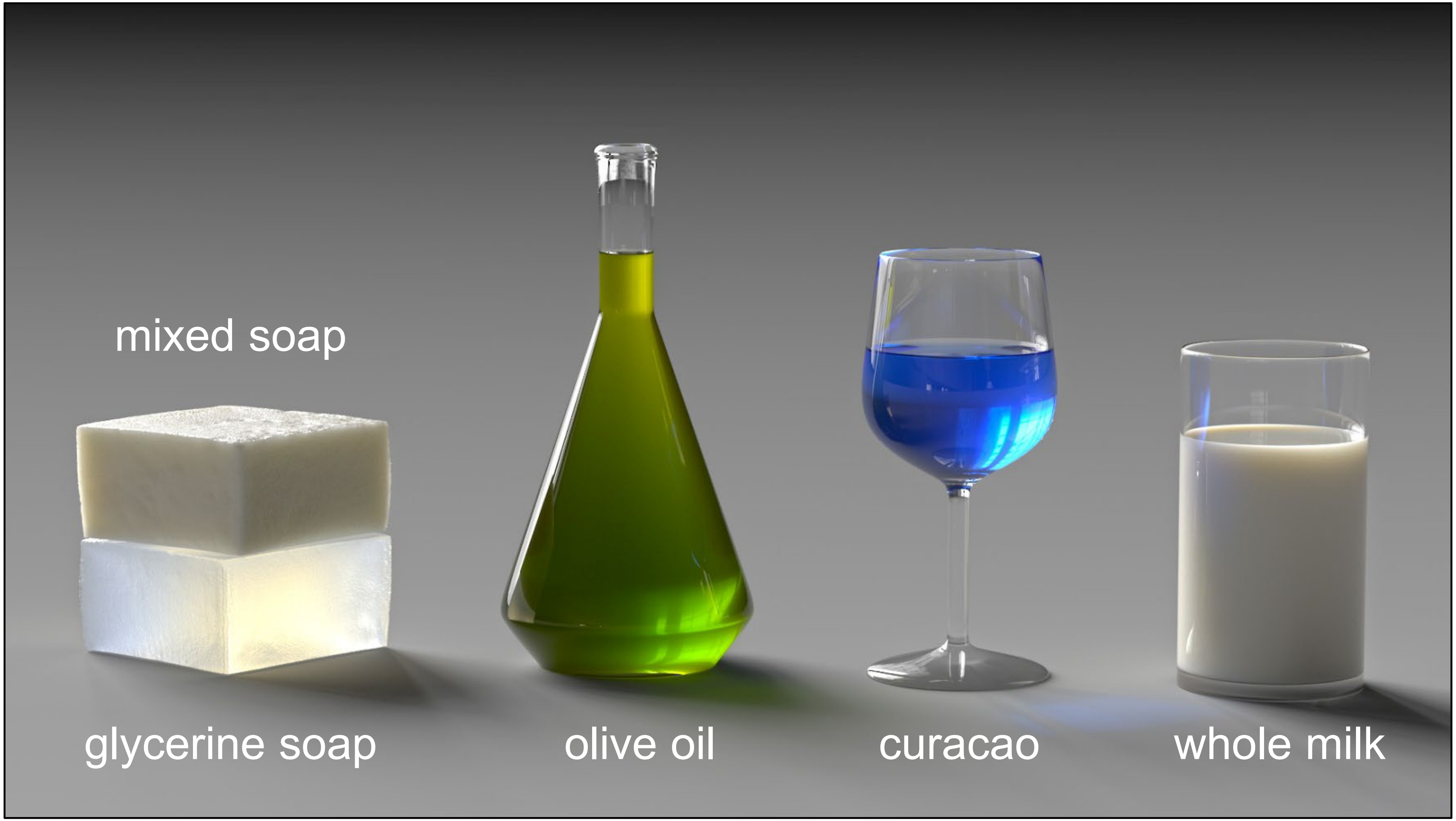






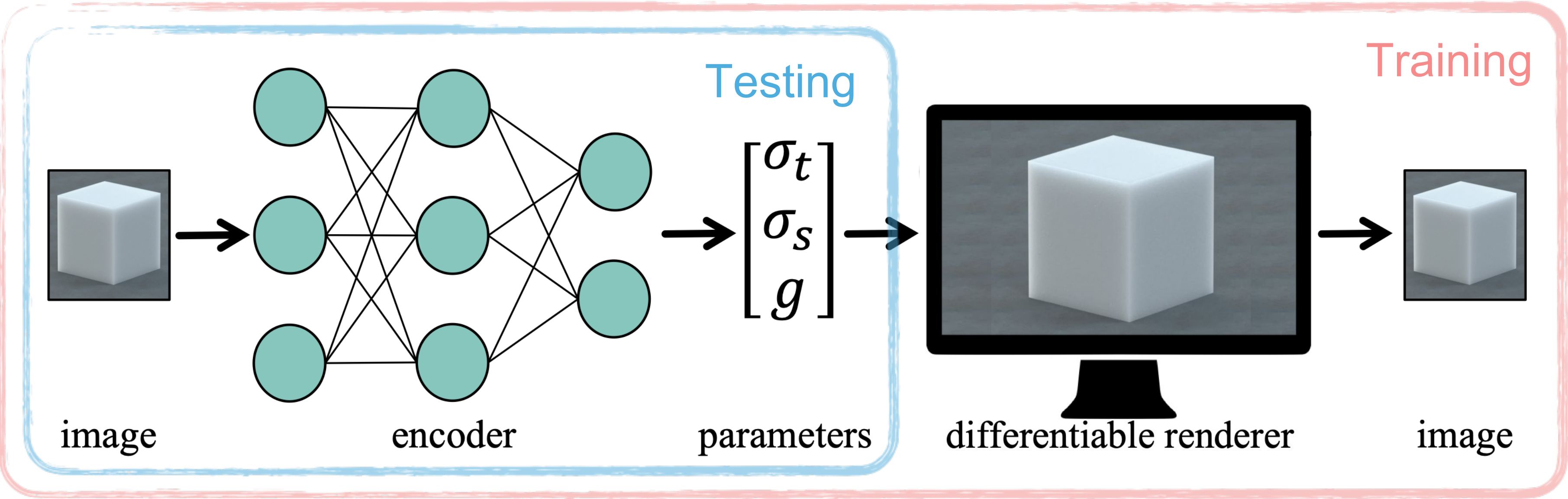
Invert using differentiable rendering

Synthetic renderings

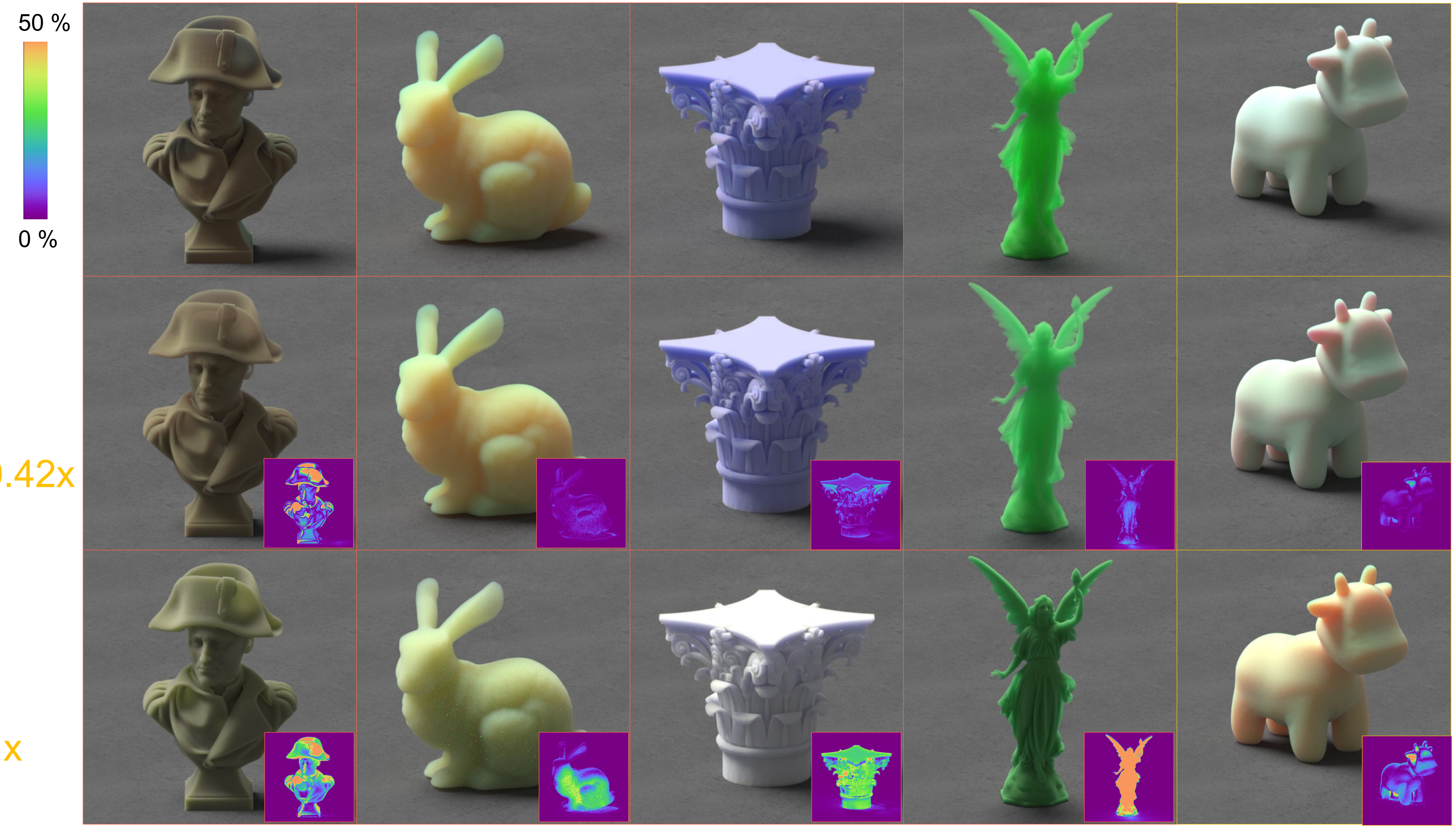


Inverse transport networks [Che et al. 2020]

- Integrate physics-based rendering into **machine learning** pipeline
- Predict scattering parameters from images

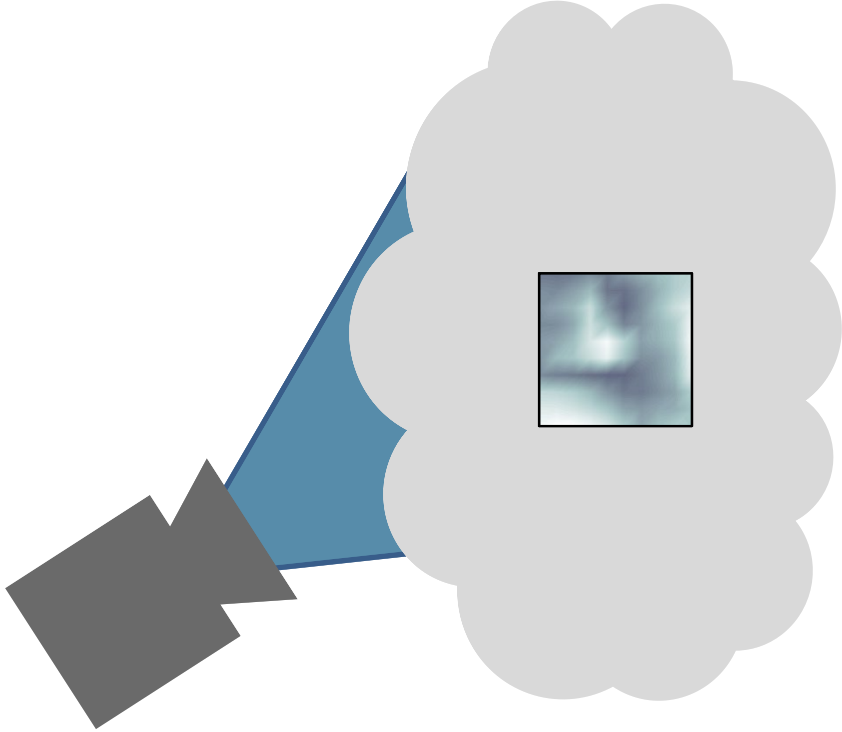


- Utilize *image loss* provided by a volume path tracer to regularize training
- Use the trained encoder to perform inverse scattering during testing



Groundtruth
50 %
0 %
Inverse transport network
parameter loss: 0.60x
appearance loss: 0.40x
novel appearance loss: 0.42x
Baseline
parameter loss: 1x
appearance loss: 1x
novel appearance loss: 1x

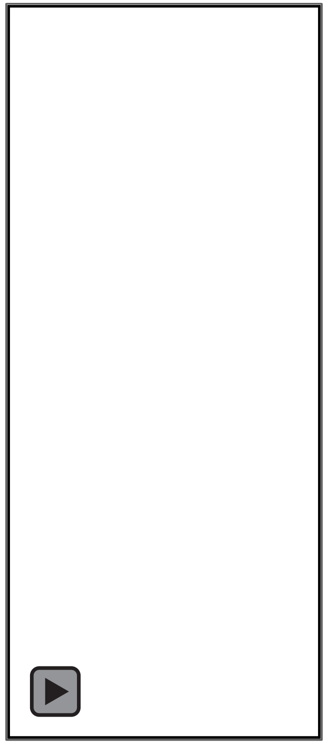
Optical tomography [Gkioulekas et al. 2015]



camera

thick smoke cloud

simulated camera measurements

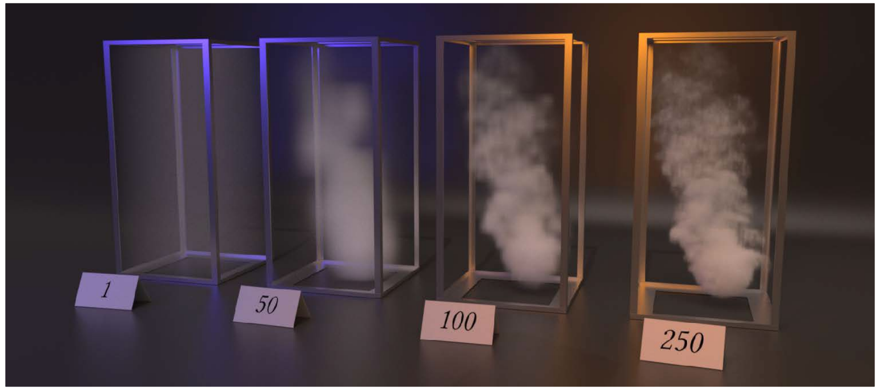


reconstructed cloud volume

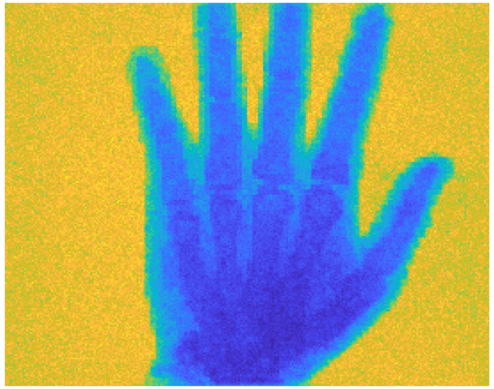
slice through the cloud

Active area of research

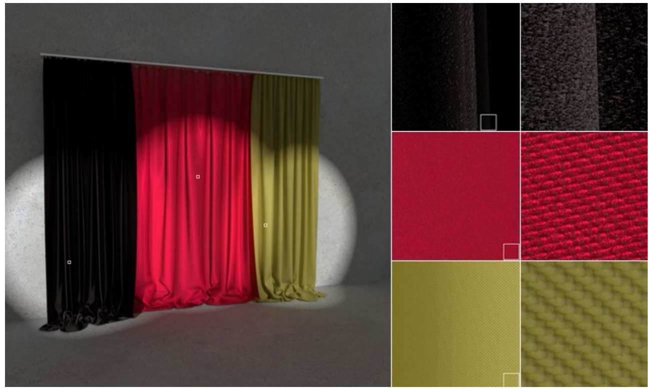
industrial dispersions
[Gkioulekas et al. 2013]



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

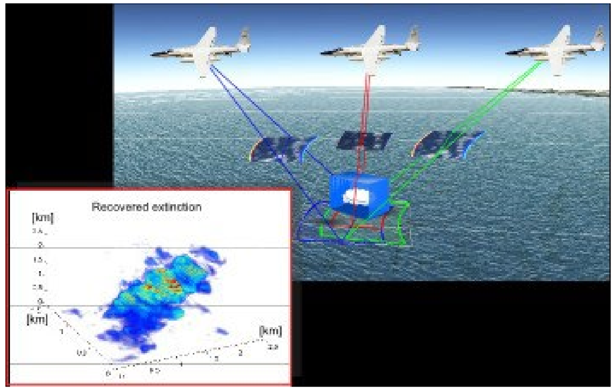


computed tomography
[Geva et al. 2018]



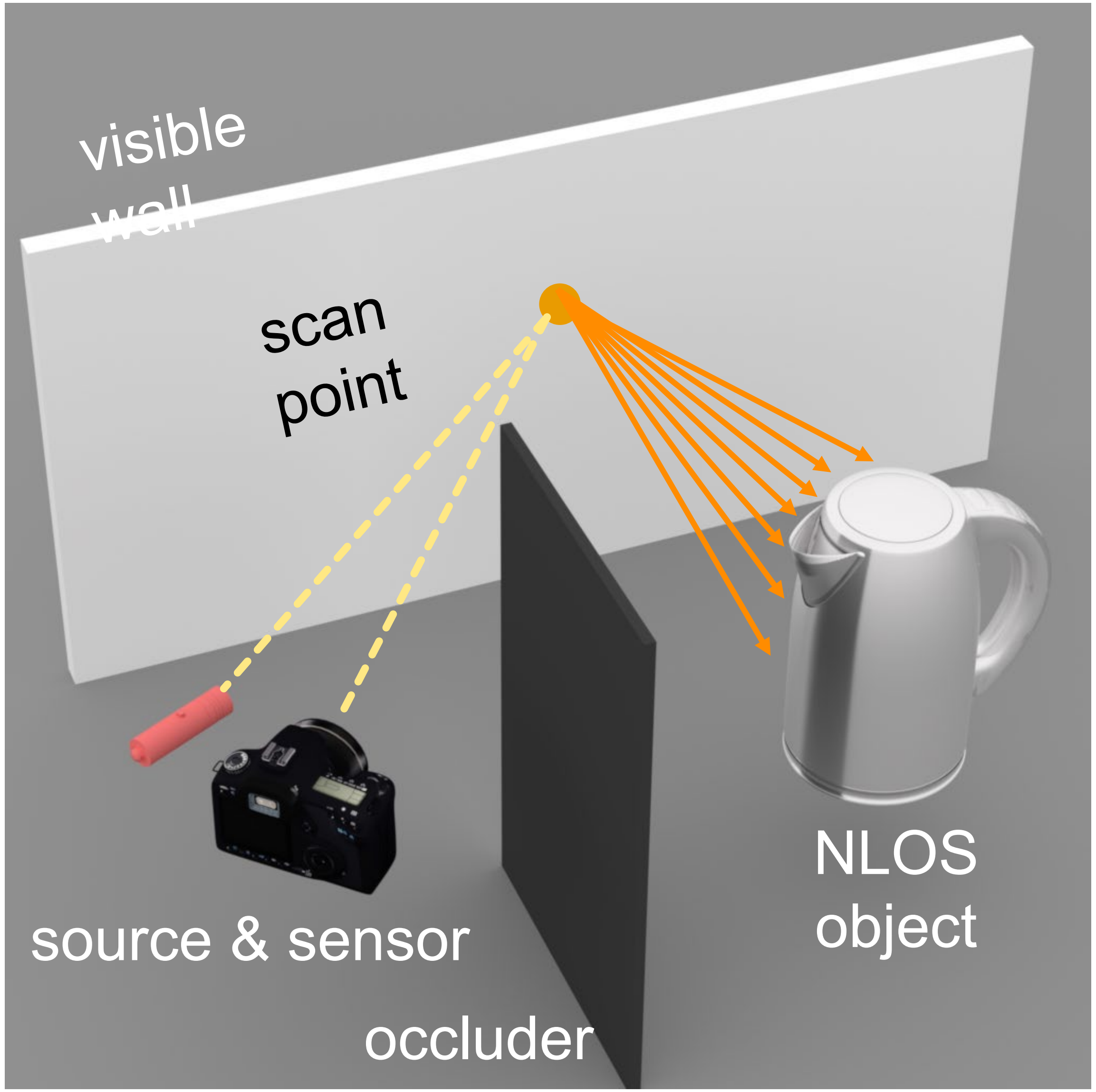
woven fabrics
[Khungurn et al. 2015,
Zhao et al. 2016]

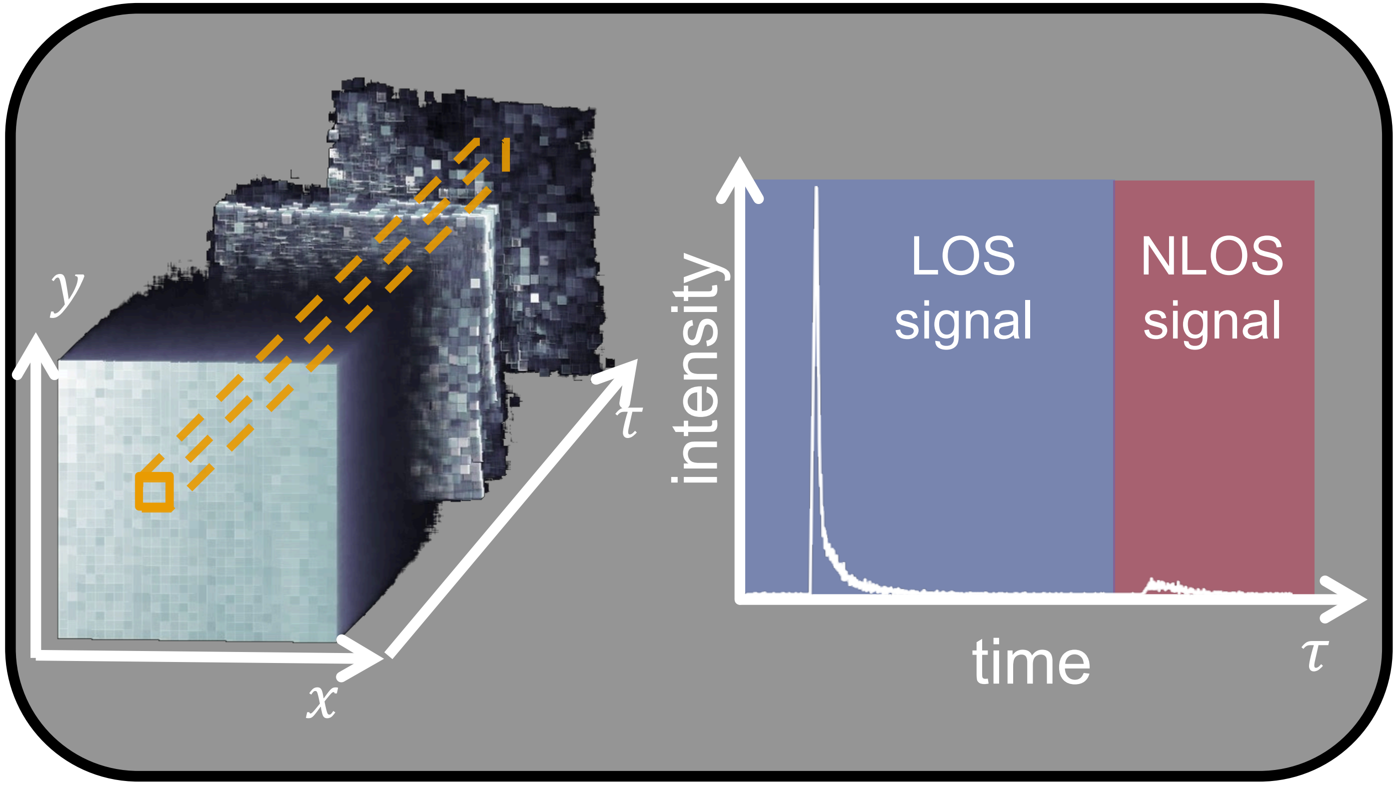
3D printing
[Elek et al. 2019,
Nindel et al. 2021]



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

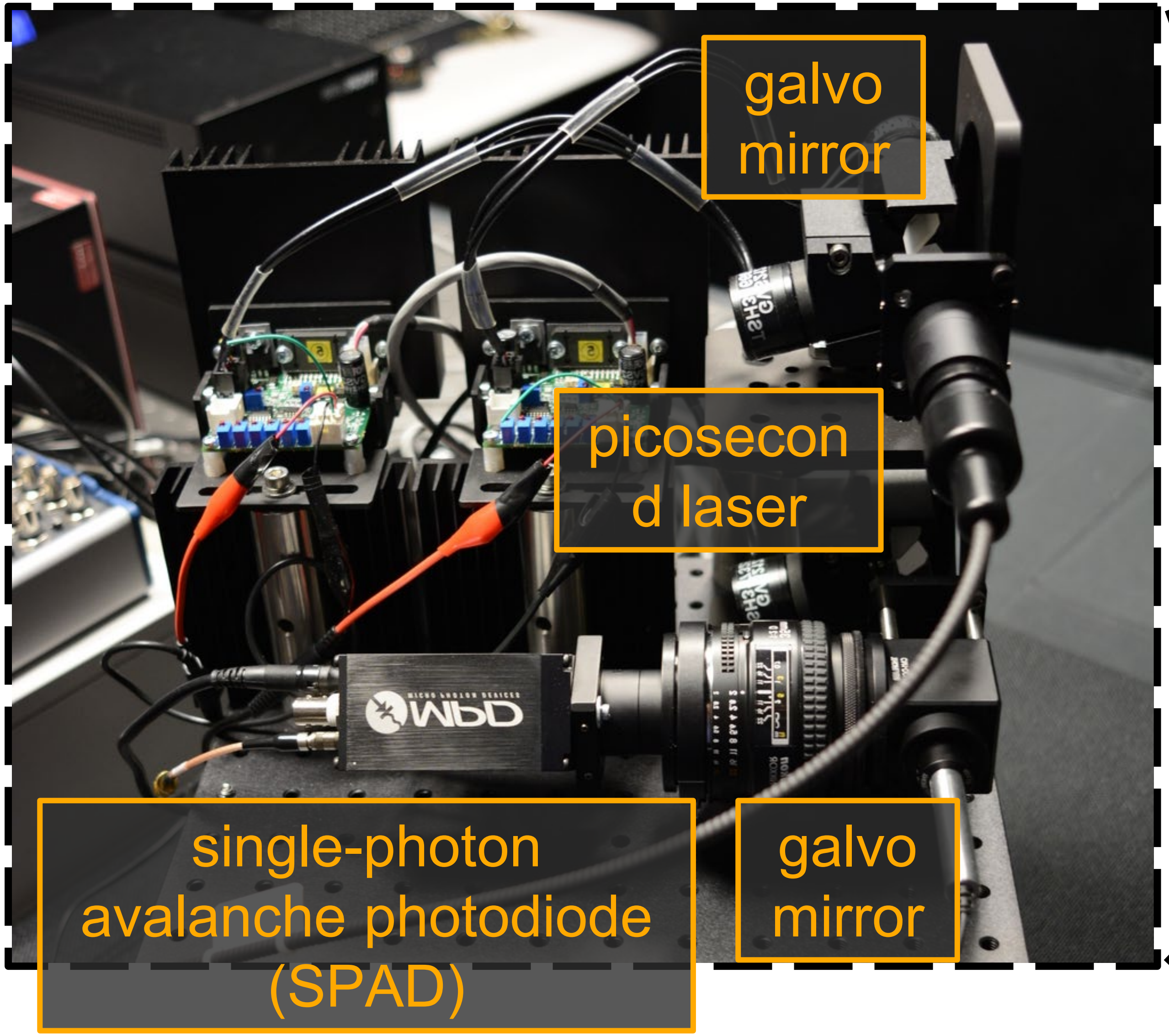
Non-line-of-sight (NLOS) imaging

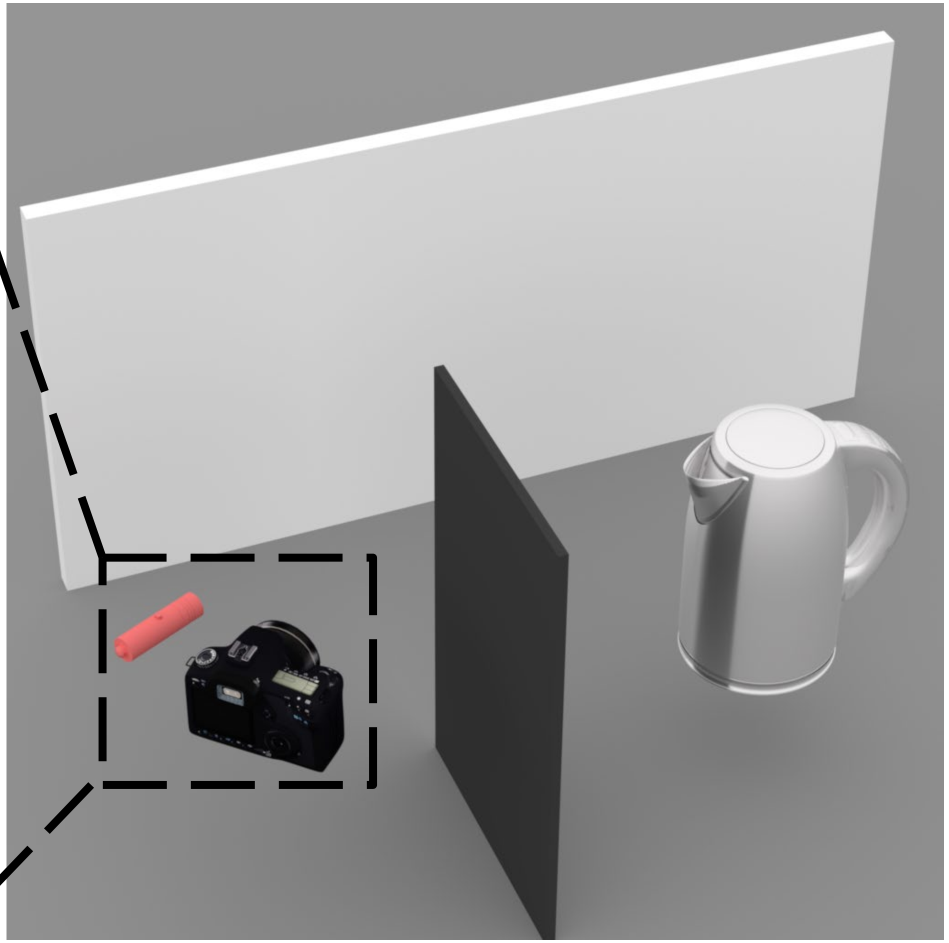




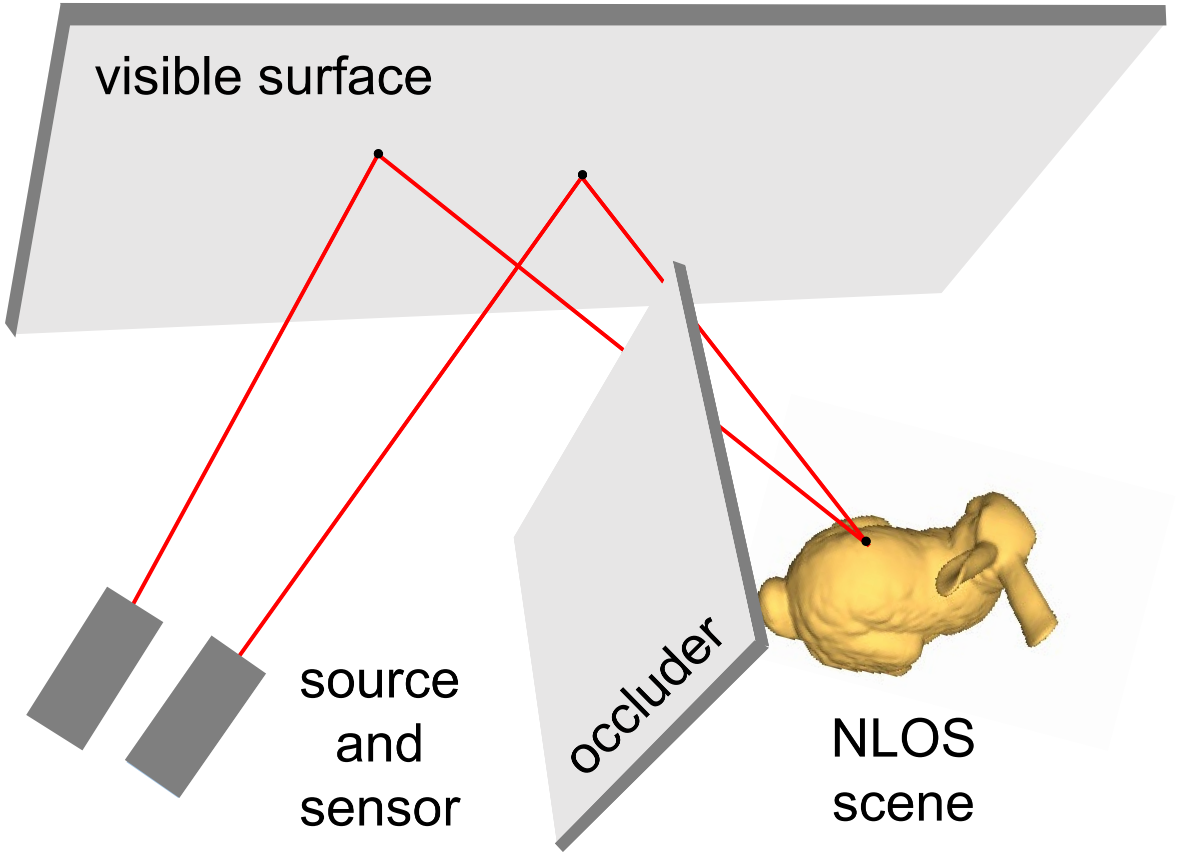
Time-of-flight measurements

SPAD-based lidar

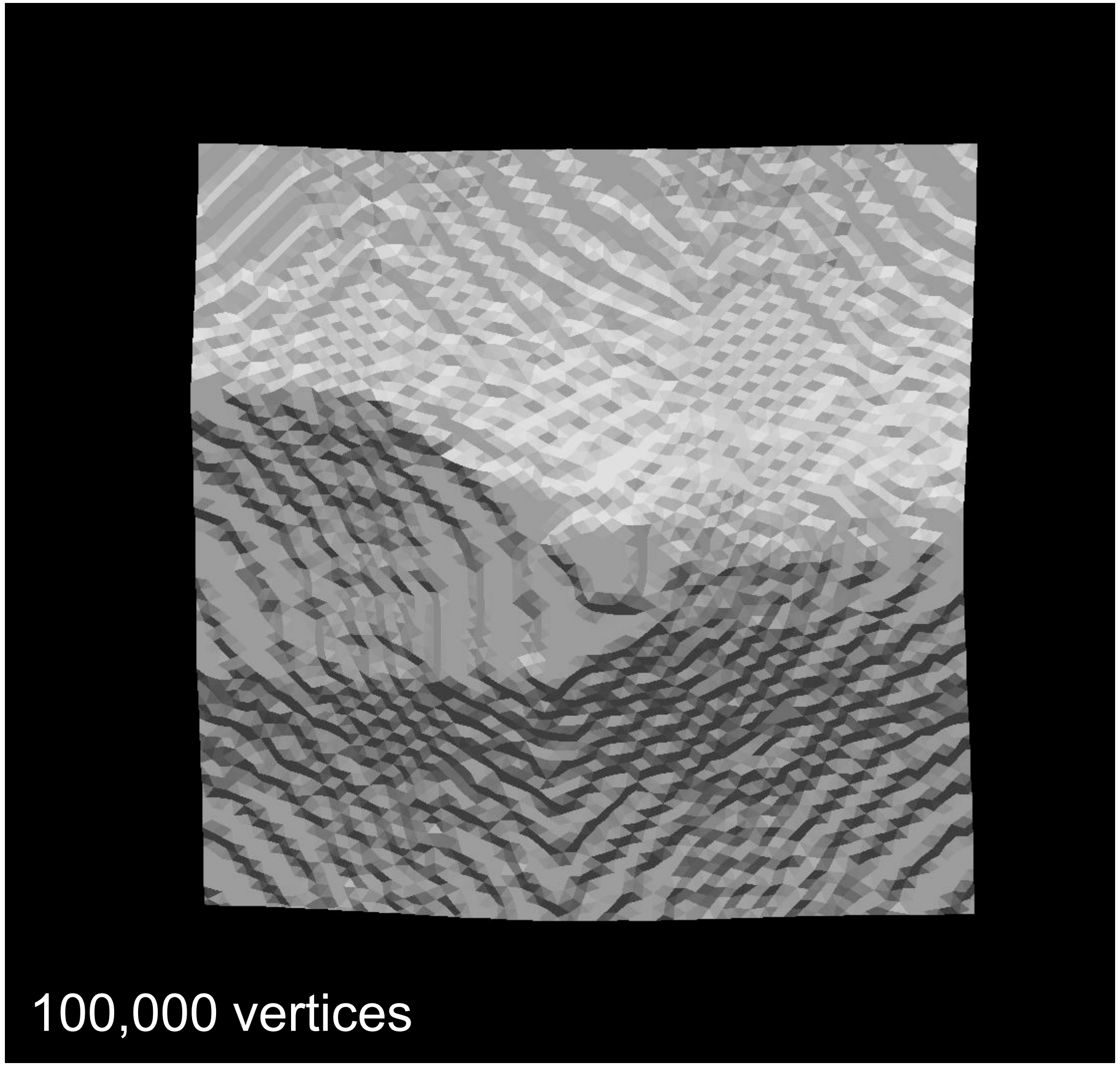




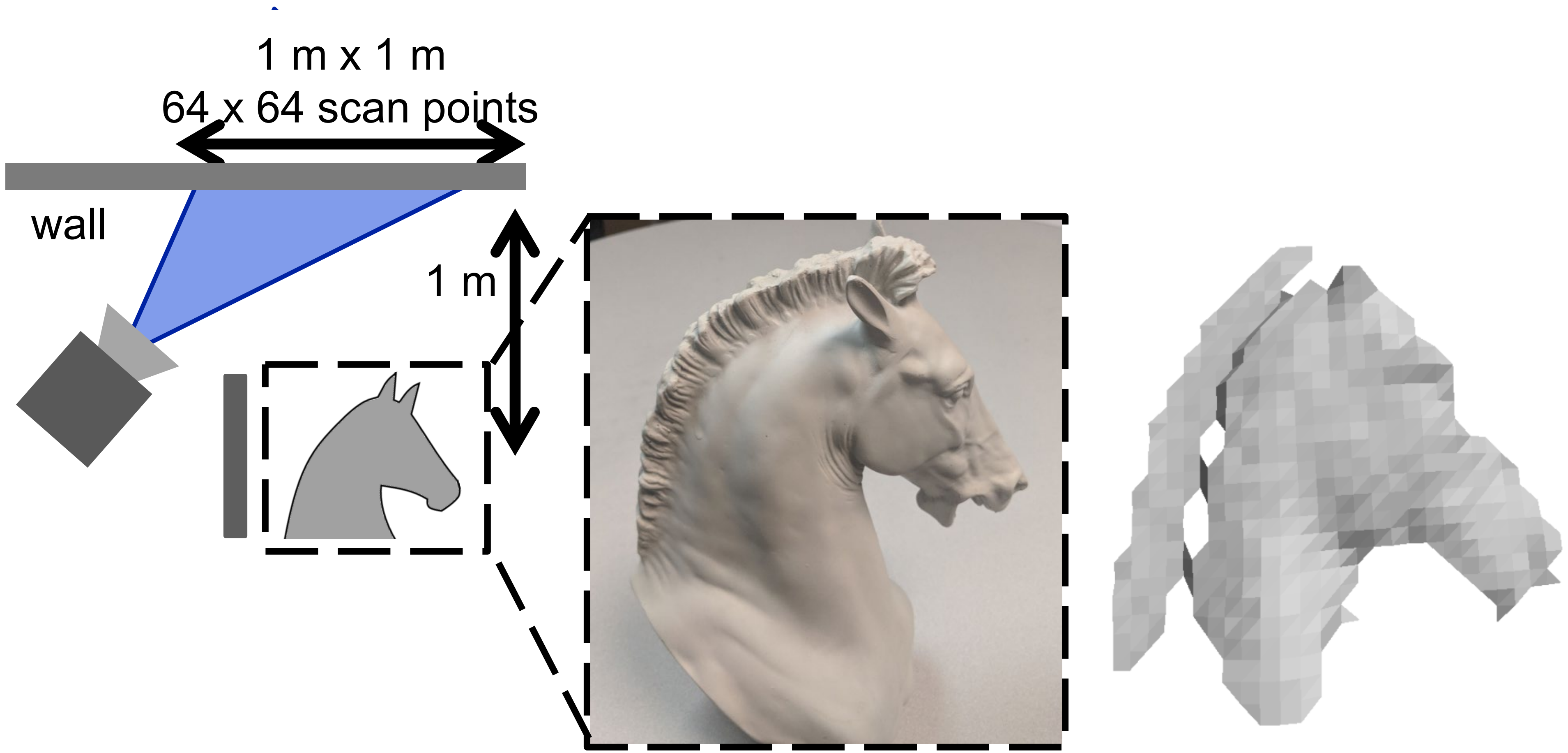
NLOS shape optimization [Tsai et al. 2019]



Simulated time-of-flight data



NLOS shape optimization [Tsai et al. 2019]



scene

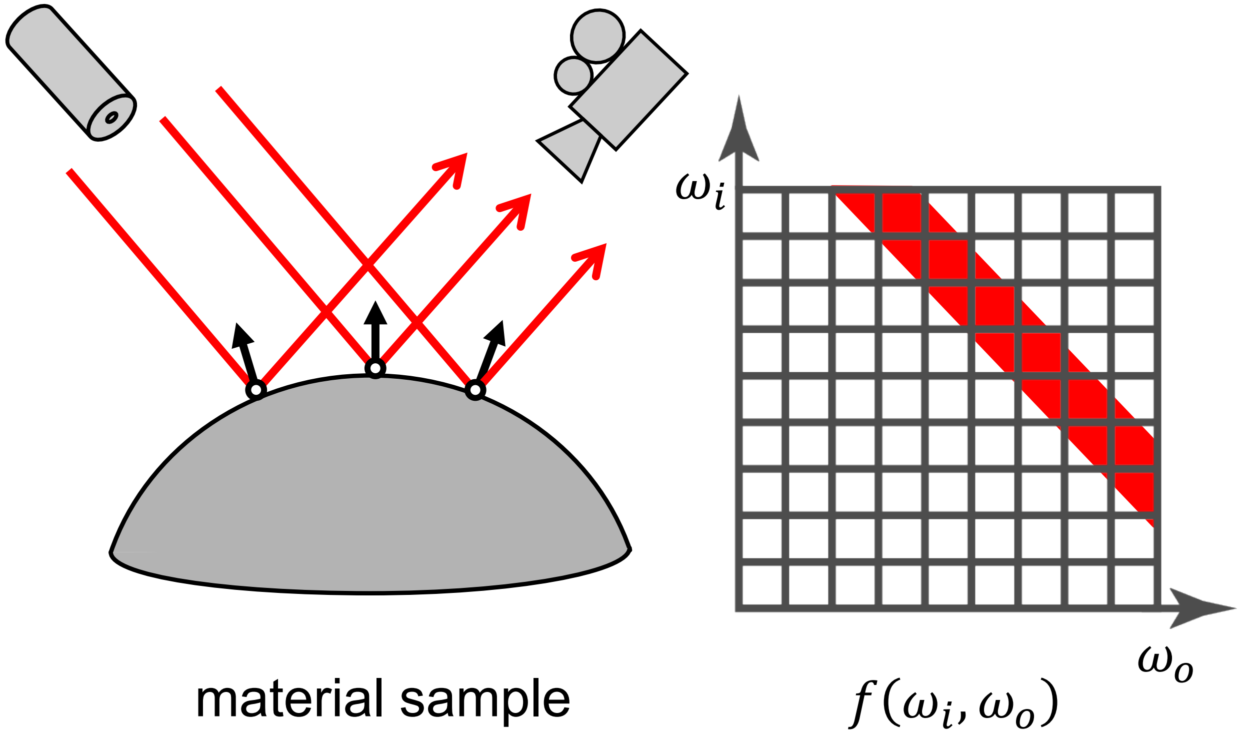
initial mesh
[O'Toole et al. 2018]

optimized
mesh

Measured time-of-flight
data

Reflectometry from interreflections [Shem-Tov et al. 2020]

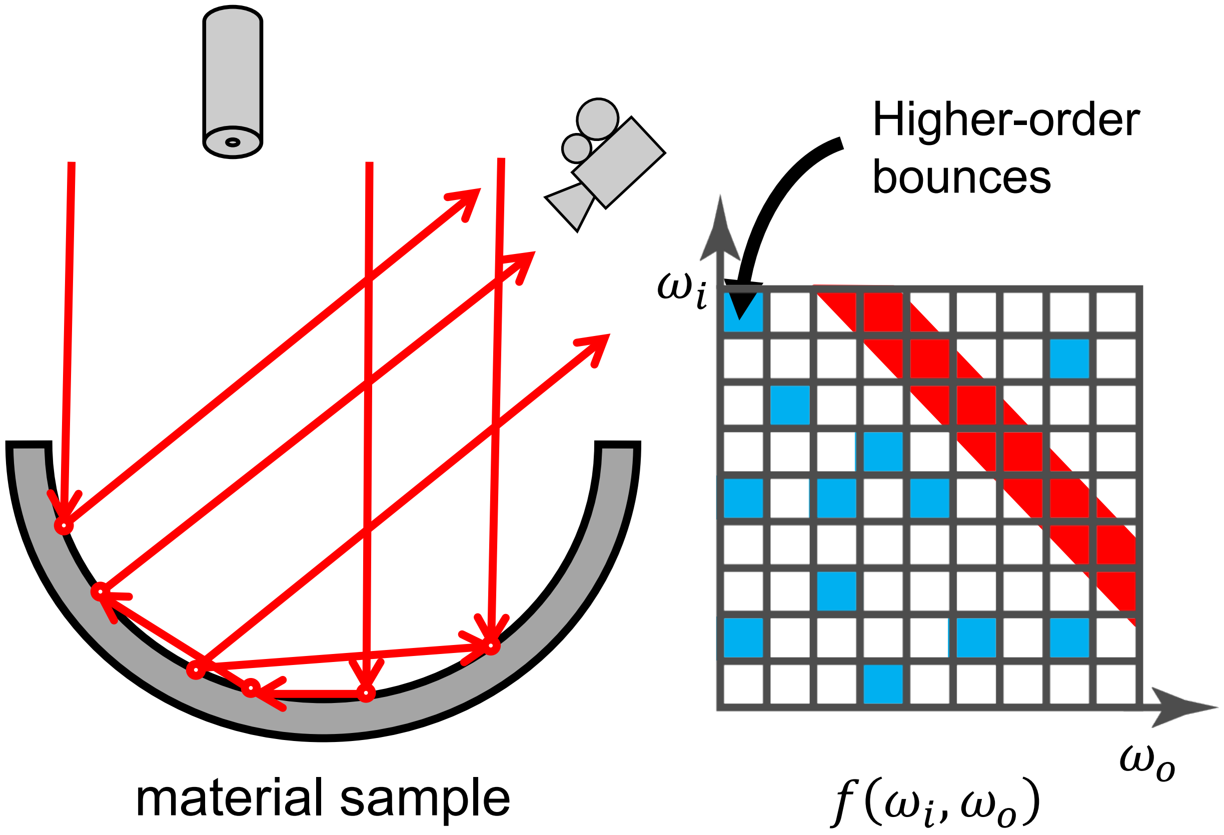
Direct illumination measurements



+ Intensities map directly to BRDF entries

- Many measurements (2D scan of light & camera)

Global illumination measurements

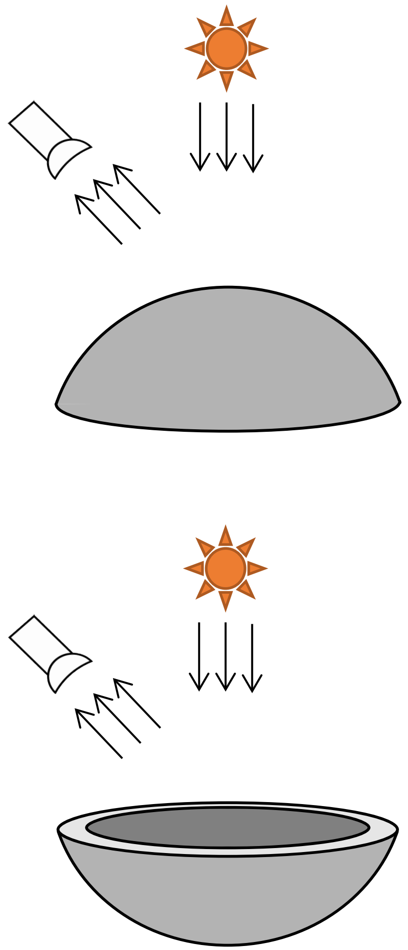


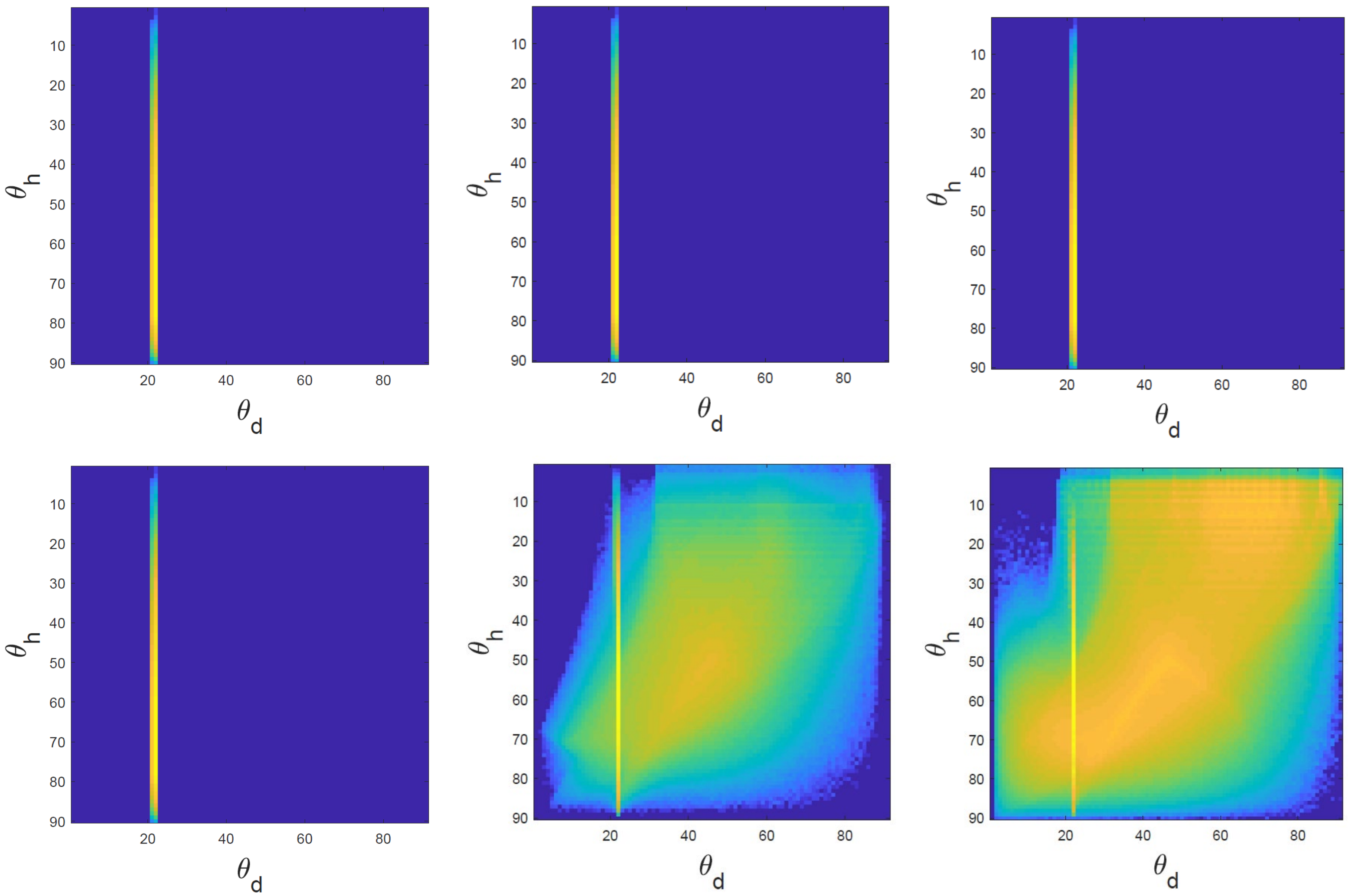
+ Fewer measurements (single image)

- Non-linear analysis-by-synthesis optimization

Solvable using differentiable rendering

Single-image dense BRDF sampling



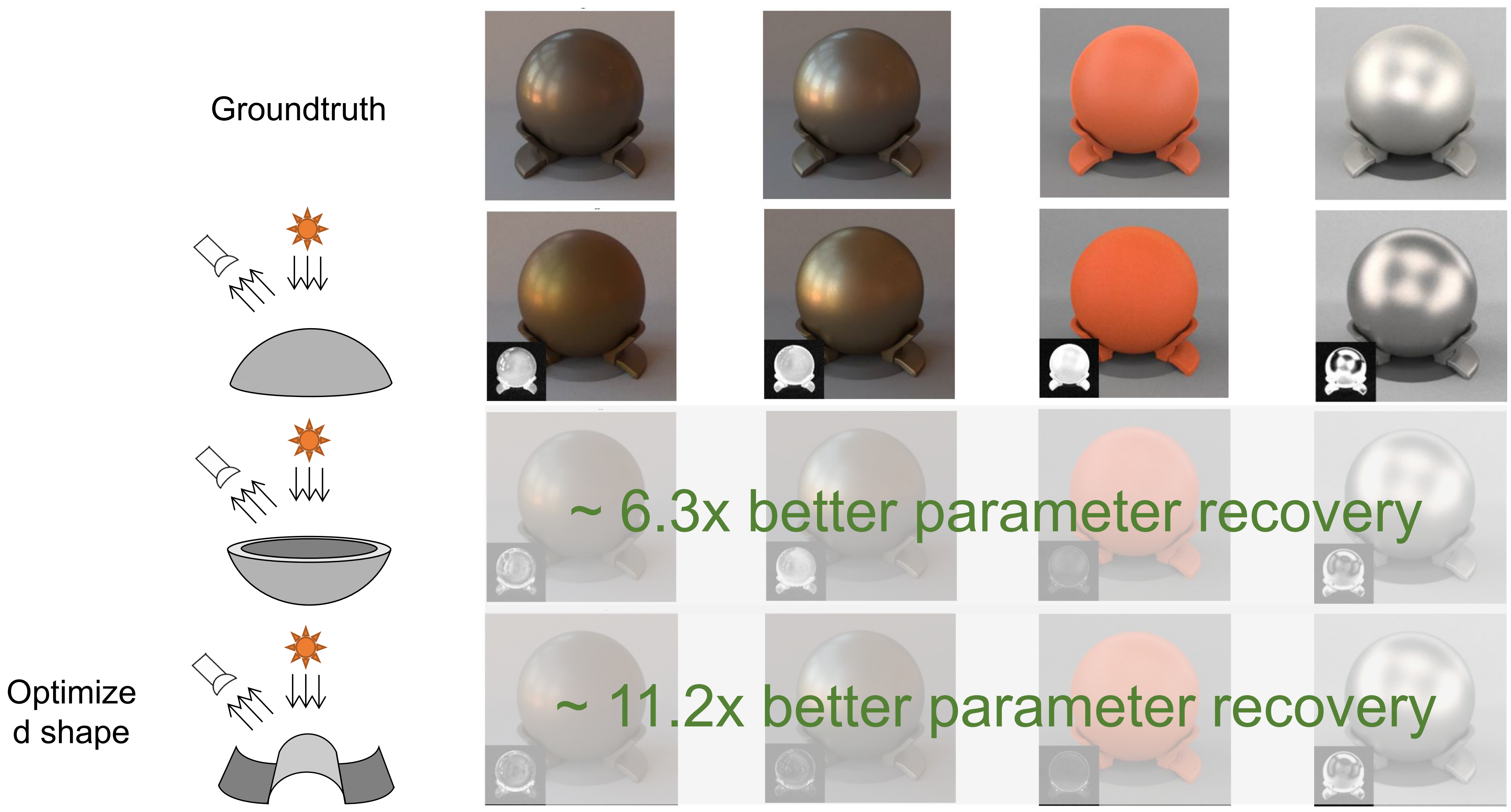


Single-bounce paths

Two-bounce paths

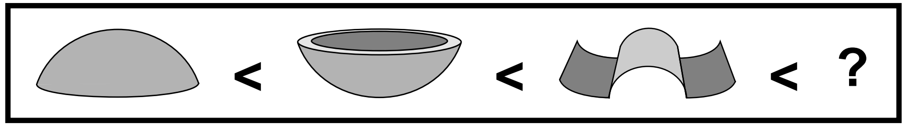
All-bounce paths

Results on MERL dataset

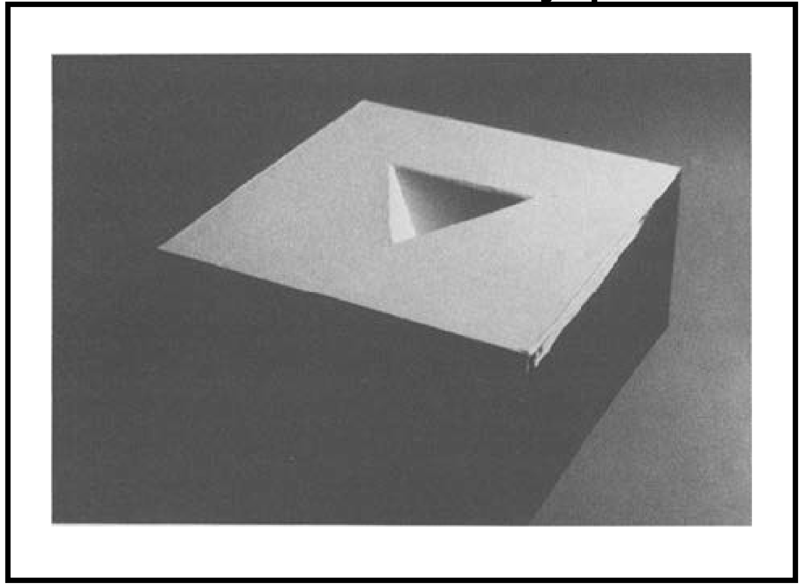


Global illumination can help…

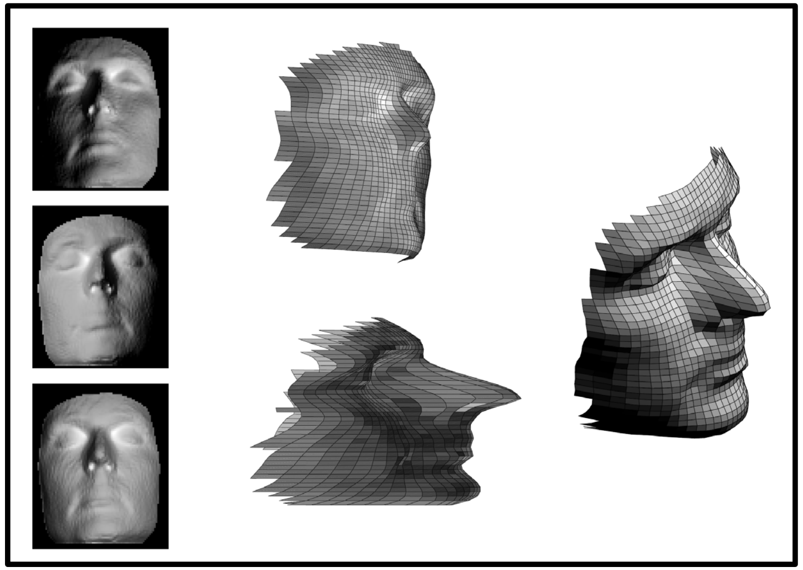
- Reduce number of measurements required for inverse rendering
  - We should rethink “optimal” acquisition systems



- Resolve ambiguities between different types of parameters
  - We should revisit theory problems on uniqueness results



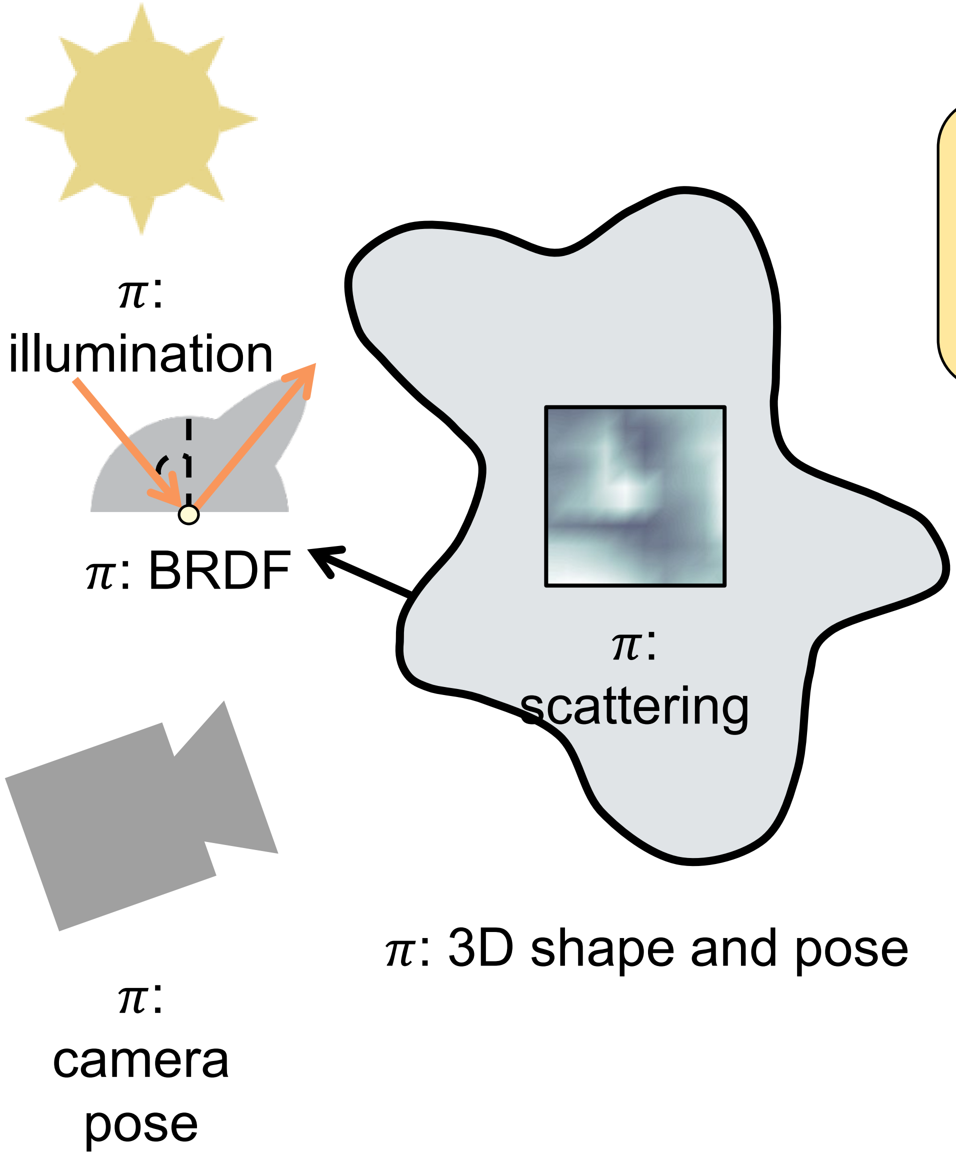
Shape from interreflections
[Nayar et al. 1990, Marr Prize]



Interreflections resolve the GBR ambiguity
[Chandraker et al. 2005]

What differentiable rendering does not give us

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



Analysis-by-synthesis optimization:

$$\min_{\substack{\text{scene}\\ \text{unknowns } \pi}} \text{loss} \left[\right.$$

$$\left. , \text{render}\left(\begin{matrix}\text{scene}\\ \text{unknowns } \pi\end{matrix}\right) \right]$$

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

$\boxed{\text{initialize } \pi \leftarrow \pi_0}$

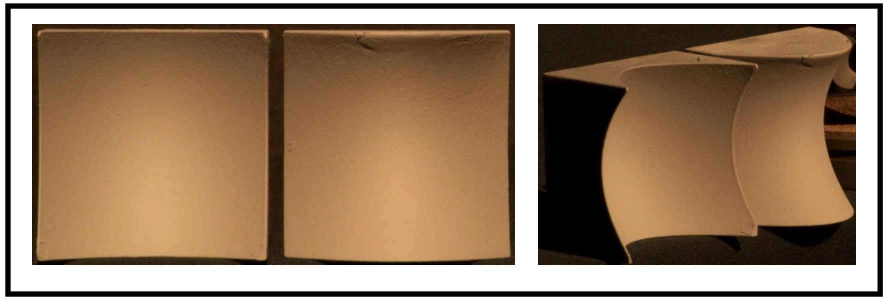
while (not converged)

update $\pi \leftarrow \boxed{\pi + \eta \cdot \frac{\mathrm{dloss}(\pi)}{\mathrm{d}\pi}}$

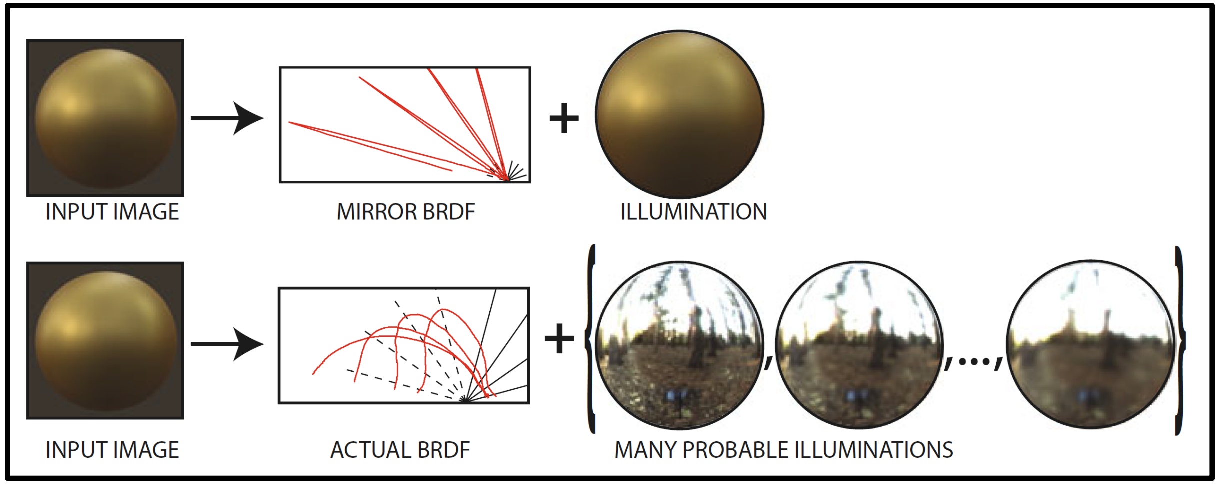
Differentiable rendering

Why we need good initializations

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



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



Ambiguities between BRDF and lighting [Romeiro and Zickler 2010]

Ambiguities between scattering parameters [Zhao et al. 2014]

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

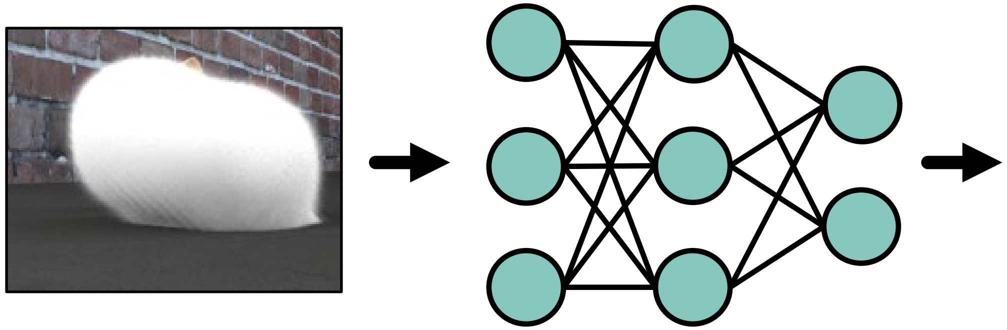
Analysis-by-synthesis optimization:

$$\min_{\substack{\text{scene}\\ \text{unknowns } \pi}} \boxed{\text{loss}}\left[\right.$$

$$\left., \text{render}\left(\text{scene unknowns } \pi\right)\right]$$

Learned initializations help:

- avoid local minima
- accelerate convergence



Neural network

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

$\boxed{\text{initialize } \pi \leftarrow \pi_0}$

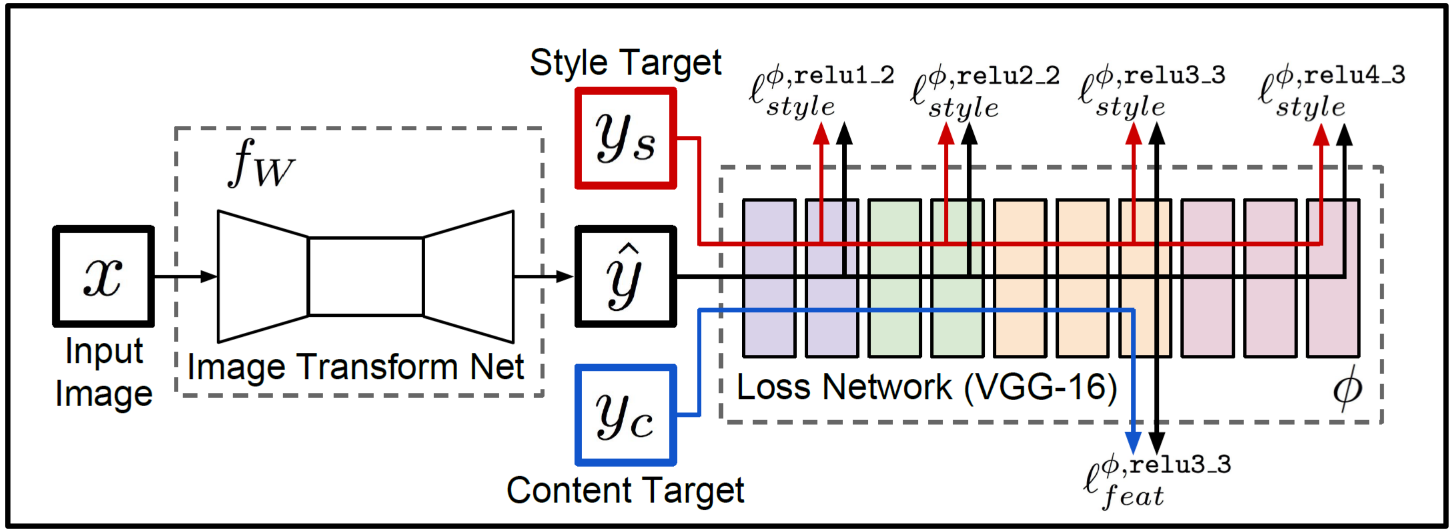
while (not converged)

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

Differentiable rendering

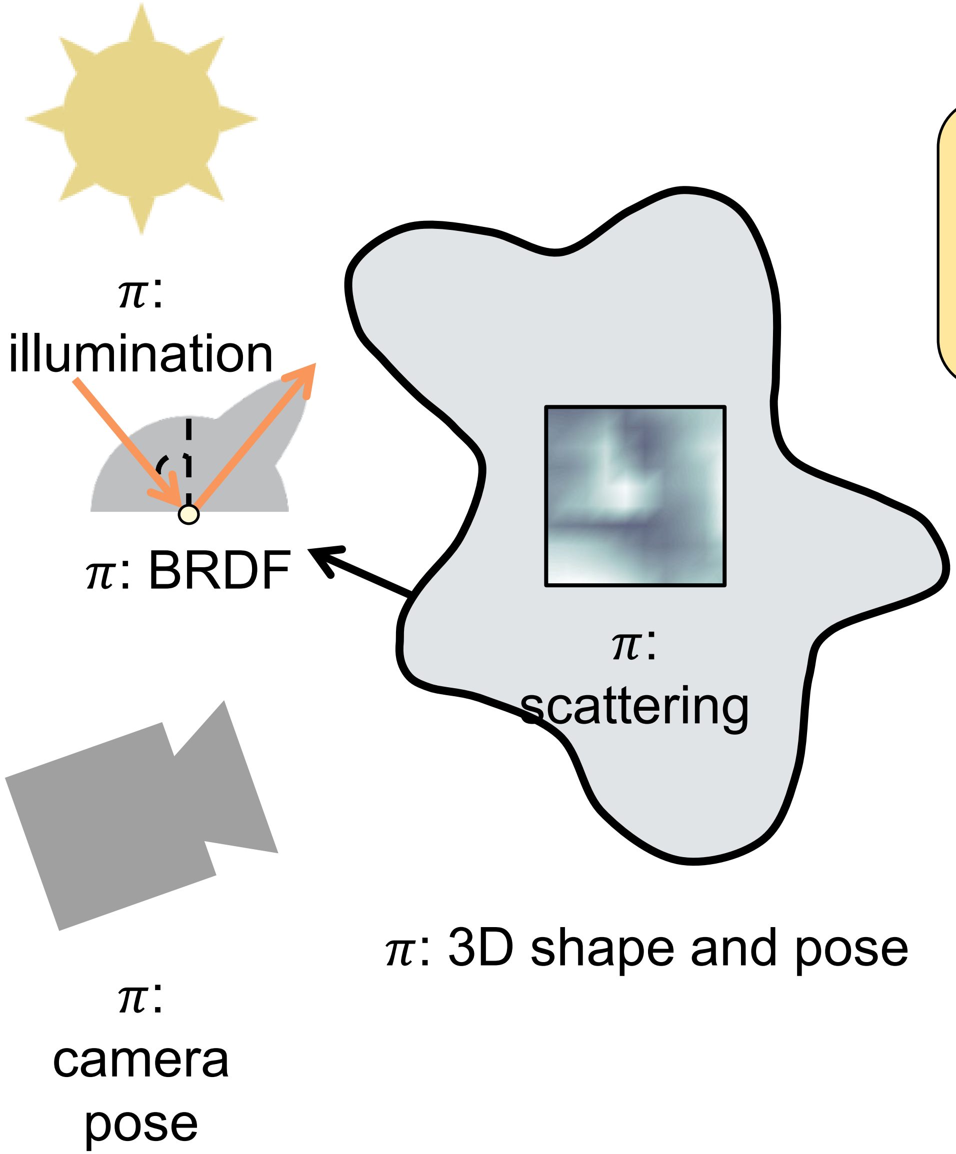
Why we need discriminative loss functions

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



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

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



Analysis-by-synthesis optimization:

$$\min_{\text{scene unknowns } \pi} \text{loss}\left[\right.$$

$$\left. , \text{render}\left(\text{scene unknowns } \pi\right)\right]$$

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

initialize $\pi \leftarrow \pi_0$

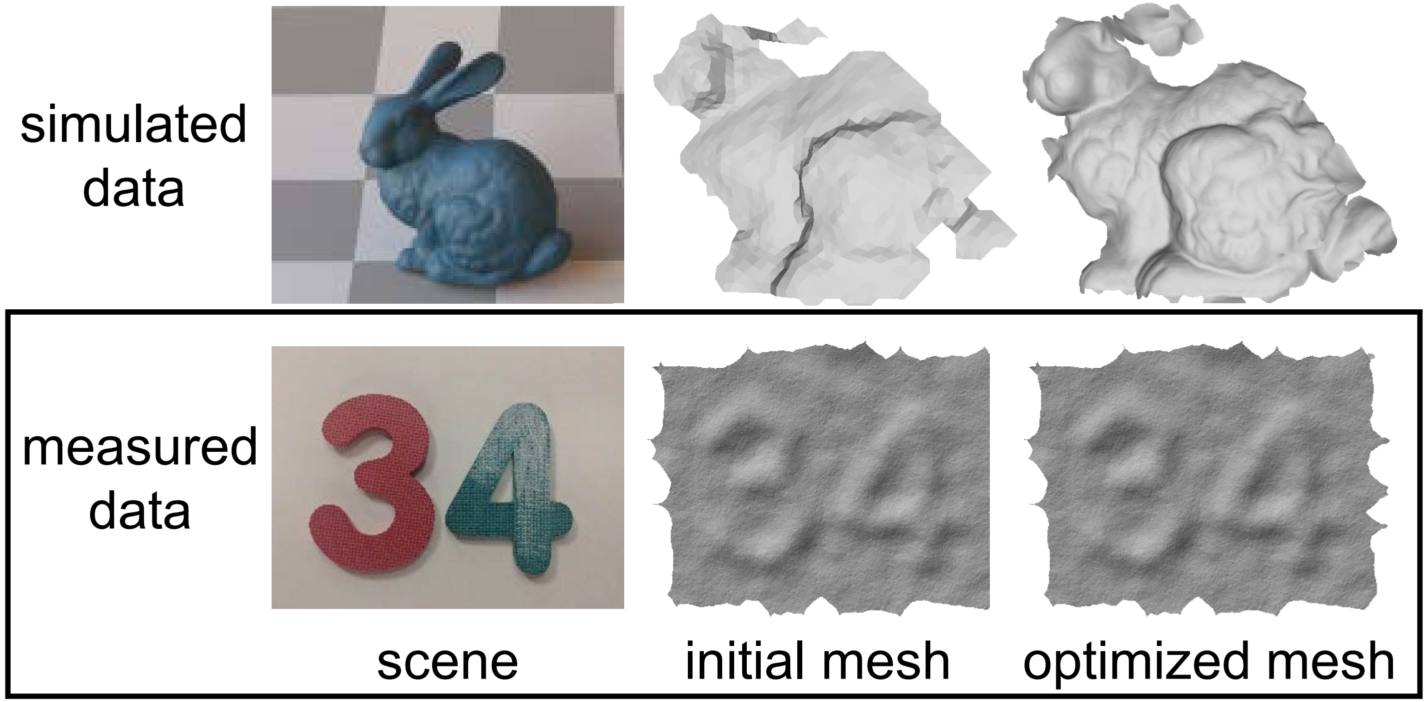
while (not converged)

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

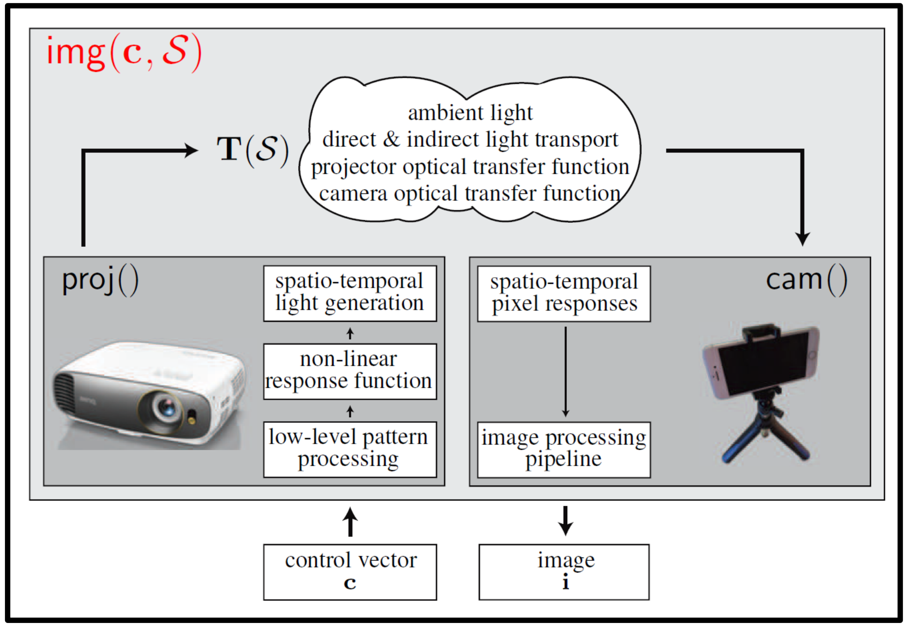
Differentiable rendering

High signal-to-noise ratio is critical

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



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



Optical gradient descent [Chen et al. 2020]

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).

You don’t always need <u>Monte Carlo</u> 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:

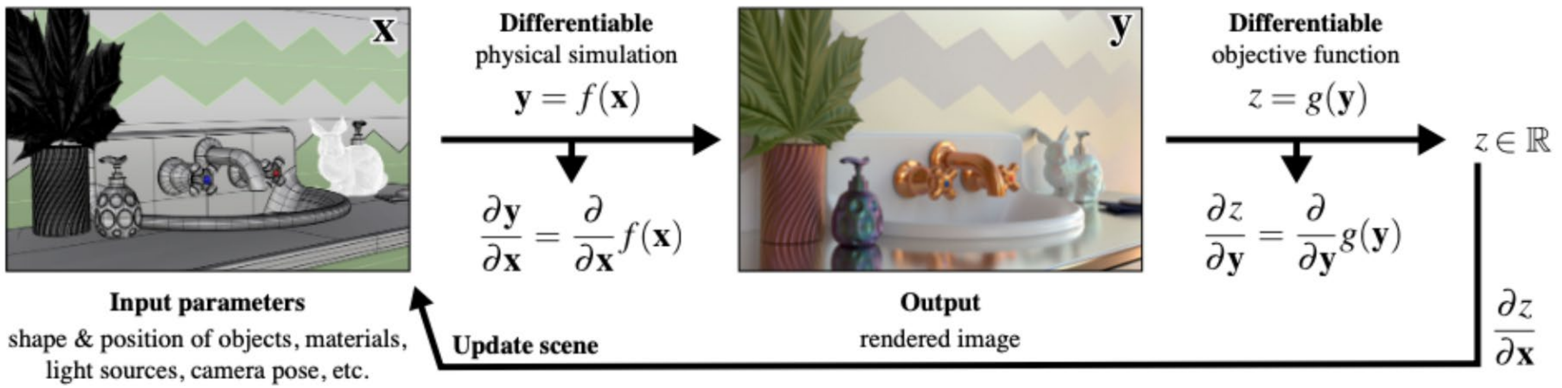
- Unbiased and consistent gradients are very expensive to compute.
- Biased and/or inconsistent gradients can be very cheap to compute.
- Often, biased and/or inconsistent gradients are enough for convergence.
- <u>Stochastic</u> gradient descent matters a lot.

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