# Inverse and differentiable rendering



15-468, 15-668, 15-868 Physics-based Rendering Spring 2022, Lecture 15

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

# Course announcements

- Programming assignment 4 posted, due Friday 4/1 at 23:59.
  - How many of you have looked at/started/finished it?
  - Any questions?
- Take-home quiz 9-10 posted, due 4/26.
- Nobody was around for yesterday's recitation :-( .
- Final project details posted.

# Overview of today's lecture

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

# Forward rendering



physically-accurate rendering





digital scene specification (geometry, materials, optics, light sources) photorealistic simulated image

# Inverse rendering



physically-accurate inverse rendering



digital scene specification (geometry, materials, camera, light sources) photomægedistic synetdsætrierine agse

# What I was doing in 2013



## I wanted to make images such as this one



### Scattering: extremely multi-path transport





volumetric density  $\sigma_t$ scattenaitegrialbrecto a phase function  $f_r$ 

# Acquisition setup



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



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



## Other scattering materials







everyday materials [Gkioulekas et al. 2013]

industrial dispersions computed tomography [Gkioulekas et al. 2013]

[Geva et al. 2018]



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





clouds 3D printing [Elek et al. 2017, 2019] [Levis et al. 2015, 2017]



optical tomography [Gkioulekas et al. 2016]

#### Making sense of global illumination



X: 3D shapeX: surface reflectanceX: occluded imagingX: illumination





analysis by synthesis

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

stochastic gradient descent



differentiable rendering: image gradients with respect to arbitrary X

# Differentiable rendering

# Not related to:

#### **Gradient-Domain Path Tracing**







#### Light Transport Simulation in the Gradient Domain



"Gradient" in their case refers to image edges.



force input and output images to be the same

## Quick reminder from calculus

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

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

what is this rule called?

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

differentiation under the integral sign

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

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

differentiation under the integral sign

$$\frac{\partial}{\partial \pi} \int_{a(\pi)}^{b(\pi)} f(x;\pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{\partial}{\partial \pi} f(x;\pi) dx \qquad \text{what is this rule called} \\ + f(b(\pi);\pi) \frac{\partial b(\pi)}{\partial \pi} - f(\alpha(\pi);\pi) \frac{\partial \alpha(\pi)}{\partial \pi}$$

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

differentiation under the integral sign

$$\frac{\partial}{\partial \pi} \int_{a(\pi)}^{b(\pi)} f(x;\pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{\partial}{\partial \pi} f(x;\pi) dx \qquad \text{Leibniz integral rule} \\ + f(b(\pi);\pi) \frac{\partial b(\pi)}{\partial \pi} - f(\alpha(\pi);\pi) \frac{\partial \alpha(\pi)}{\partial \pi}$$

### Trivial differentiable rendering

Images as path integrals



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

- $\bar{\mathbf{x}} \rightarrow$  Light path, set of ordered vertices <u>on surfaces</u>
- $\mathbb{P} \rightarrow$  Space of valid paths
- $f(\bar{\mathbf{x}}) \rightarrow$  Path contribution, includes geometric terms (visibility, fall-off) & local terms (BRDF, foreshortening, emmision)

#### Monte Carlo rendering: approximating path integrals





 $\overline{x_i} \rightarrow \underline{Randomly sampled}$  light paths

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

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

#### How can we approximate the derivative of the image?





#### 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?

- <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



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

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 auto-differentiation 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



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

- $\bar{\mathbf{x}} \rightarrow$  Light path, set of ordered vertices <u>on surfaces</u>
- $\mathbb{P} \rightarrow$  Space of valid paths
- $f(\overline{\mathbf{x}}) \rightarrow$  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



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

- $\overline{\mathbf{x}} \rightarrow$  Light path, set of ordered vertices <u>on surfaces</u>
- $\mathbb{P} \rightarrow$  Space of valid paths
- f(x̄) → 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



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

differentiation under the integral sign

- $\overline{\mathbf{x}} \rightarrow$  Light path, set of ordered vertices <u>on surfaces</u>
- $\mathbb{P} \rightarrow$  Space of valid paths
- f(x̄) → 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





- $\overline{x_i} \rightarrow \underline{Randomly sampled}$  light paths
- $p(\bar{\mathbf{x}}_i) \rightarrow \text{Probability of sampling a path}$

Sample paths using path tracing etc.

#### Score estimator

$$f(\overline{\mathbf{x}};\pi) = \prod_{b=1}^{B} f_{s}(x_{b-1} \to x_{b} \to x_{b+1};\pi) \frac{V(x_{b-1} \leftrightarrow x_{b})}{\|x_{b-1} - x_{b}\|^{2}}$$

Foreshortening terms are included in the BRDF

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

$$\sum_{b=1}^{B} \frac{\partial f_{s}}{\partial \pi}(x_{b-1} \to x_{b} \to x_{b+1};\pi)}{f_{s}(x_{b-1} \to x_{b} \to x_{b+1};\pi)} \quad \text{At each path vertex:}$$

$$\cdot \text{ Update product throughput using } f_{s}$$

• 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]



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





3D printing clouds [Elek et al. 2017, 2019] [Levis et al. 2015, 2017]



optical tomography [Gkioulekas et al. 2016]

#### Even simpler: use autodiff


#### Compare with...



### Even simpler: use autodiff



#### Compute an estimate of the derivative





#### derivative wrt volumetric density

#### **Inverse Transport Networks**

Chengqian Che Carnegie Mellon University

Fujun LuanShuang ZhaoCornell UniversityUniversity of California, Irvine

Kavita Bala Cornell University Ioannis Gkioulekas Carnegie Mellon University



derivative wrt BRDF



#### derivative wrt normal

#### Comparison with finite differences



forward

rendered

finite differences

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

### Compute a derivative of the estimate



#### Mitsuba 2: A Retargetable Forward and Inverse Renderer

MERLIN NIMIER-DAVID<sup>\*</sup>, École Polytechnique Fédérale de Lausanne DELIO VICINI<sup>\*</sup>, 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.

#### derivative wrt volumetric density

## Looking inside scattering objects



camera thick smoke cloud simulated camera reconstructed cloud slice through measurements volume the cloud

## Looking inside scattering objects





## Inverse transport network



# 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



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

differentiation under the integral sign

- $\overline{\mathbf{x}} \rightarrow$  Light path, set of ordered vertices <u>on surfaces</u>
- $\mathbb{P} \rightarrow$  Space of valid paths
- f(x̄) → 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



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

differentiation under the integral sign

What about parameters  $\pi$  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

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

- $L \rightarrow$  Radiance at a point and direction
- $G \rightarrow$  All surfaces in the scene
- $f \rightarrow$  Reflection, foreshortening, and fall-off
- $\lor \rightarrow \lor$  Visibility



Let's slightly rewrite the rendering equation

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

- $L \rightarrow$  Radiance at a point and direction
- $V \rightarrow All \underline{visible}$  surfaces in the scene
- $f \rightarrow$  Reflection, foreshortening, and fall-off



### Let's differentiate it

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

- $L \rightarrow$  Radiance at a point and direction
- $V \rightarrow All \underline{visible}$  surfaces in the scene
- $f \rightarrow$  Reflection, foreshortening, and fall-off





### Let's differentiate it

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

- $L \rightarrow$  Radiance at a point and direction
- $V \rightarrow All \underline{visible}$  surfaces in the scene
- $f \rightarrow$  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

$$\frac{\partial}{\partial \pi} \int_{a(\pi)}^{b(\pi)} f(x;\pi) dx = \int_{a(\pi)}^{b(\pi)} \frac{\partial}{\partial \pi} f(x;\pi) dx \qquad \text{Leibniz integral rule} \\ + f(b(\pi);\pi) \frac{\partial b(\pi)}{\partial \pi} - f(\alpha(\pi);\pi) \frac{\partial \alpha(\pi)}{\partial \pi}$$

We need a version of this for surface integrals

### Reminder from calculus

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

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

Move derivative inside integral

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

Account for discontinuities of integrand that depend on  $\pi$  +  $\sum_{i} (f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi)) \frac{dc_i(\pi)}{d\pi}$ 

### A simple example

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

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

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

Move derivative inside integral

Account for changes in integration limits

+  $1 \frac{d(4\pi)}{d\pi} - 0 \frac{d0}{d\pi}$ +  $(0-1) \frac{d(2\pi)}{d\pi}$ 

Account for discontinuities of integrand that depend on  $\pi$ 

## Leibniz integral rule

# Differentiation under the integral sign

Also known as the Leibniz integral rule

$$\frac{1}{\mathrm{d}\pi}\int_{a(\pi)}f(x,\pi)\mathrm{d}x =$$

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

Move derivative inside integral

Account for changes in integration limits

Account for discontinuities of integrand that depend on  $\pi$ 

$$+ f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi} + \sum_{i} (f(c_{i}(\pi)^{-}, \pi) - f(c_{i}(\pi)^{+}, \pi)) \frac{dc_{i}(\pi)}{d\pi}$$

# Simplified Leibniz integral rule

# Differentiation under the integral sign

Also known as the Leibniz integral rule

$$\frac{1}{\mathrm{d}\pi}\int_{a} f(x,\pi)\mathrm{d}x =$$

#### Interior integral

$$\frac{\mathrm{d}}{\mathrm{d}\pi}f(x,\pi)\mathrm{d}x$$

Move derivative inside integral

 $da(\pi)$ 

#### Boundary terms

Account for changes in **Differention** when:  $f(b(\pi),\pi) = \frac{db(\pi)}{derivative} = f(u,\pi) \cdot \pi$ when:

• Integration limits are independent of  $\pi$ .

Account for discontinuities are independent of  $\pi$ . integrand that depend on  $\pi$ 

## Reynolds transport theorem

$$\frac{d}{d\pi} \int_{\Omega(\pi)} f(x,\pi) dA(x) \stackrel{?}{=} \int_{\Omega(\pi)} \frac{df(x,\pi)}{d\pi} dA(x) + \int_{\partial\Omega(\pi)} g(x,\pi) dl(x)$$
Boundary domain  
Generalization of the Leibniz rule
Interior integral  
Generalization of the Leibniz rule
Interior integral  
 $f = 0$ 
 $f = 1$ 
discontinuity points
 $f = 1$ 
 $f = 1$ 
 $f = 1$ 

# **Direct illumination integral**



#### Radiance from *x*:



#### Monte Carlo rendering:

- Sample random directions  $\omega_i^s$  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*:

$$\frac{\mathrm{d}I}{\mathrm{d}\pi} = \frac{\int}{\partial \Pi} \int \frac{\mathrm{d}}{\partial \Pi} \int \frac{\int}{\partial \Pi} \frac{\int}{\partial \Pi} \int \frac{\int}{\partial \Pi} \int \frac{\int}{\partial \Pi} \int \frac{\int}{\partial \Pi} \int \frac{\partial \Pi}{\partial \Pi} \int \frac{\partial \Pi}{\partial$$

#### Monte Carlo differentiable rendering:

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



 $\pi$ : *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  $\omega_i^s$  from PDF  $p(\omega_i, \pi)$
- Form estimator Differentiate entire contribution  $\frac{dI}{d\pi} \approx \sum_{s} \frac{\frac{d}{d\pi} \left\{ f_{\tilde{r}}(\omega_{\tilde{l}}^{\tilde{s}}, \omega_{\theta}, \pi) L_{\tilde{l}}(\omega_{\tilde{l}}^{\tilde{s}}) \left(n : \omega_{\tilde{l}}^{\tilde{s}})\right\}}{p(\omega_{\tilde{s}}, \pi)}$

# Differential direct illumination: global parameters



Differential radiance from *x*:

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

Need to use full Reynolds transport theorem

- *π*: *global* parameters
- shape and pose of different scene elements (camera, sources, objects)

## Discontinuities in the integrand





 $\pi$ : size of the emitter

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

Integrand  $f(\omega_i)$ 

Discontinuous points  $(\pi$ -dependent)

## Applying the Reynolds transport theorem



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

## Reparameterizing the direct illumination integral



## Reparameterizing the direct illumination integral



## Differentiating the hemispherical integral



# Differentiating the area integral



## Sources of discontinuities



#### Topology-driven

Visibility-driven

## Significance of the boundary integral

ť



**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 SDFs)



opology-driven

Visibility-driven



#### $\pi$ : size of the emitter



Irradiance at *x* 

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

Unit hemisphere

Differential irradiance at x

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

#### $\pi$ : size of the emitter





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

The integrand

Discontinuous points (π-dependent)



### Let's differentiate the rendering equation

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

- $L \rightarrow$  Radiance at a point and direction
- $V \rightarrow All \underline{visible}$  surfaces in the scene

What are the "boundary" and

discontinuities of V?

 $f \rightarrow$  Reflection, foreshortening, and fall-off



### 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

$$\frac{\partial}{\partial \pi} L(x, \omega; \pi) = \int_{V(x,\pi)} \frac{\partial}{\partial \pi} L dA(x) + \int_{\partial V(x,\pi)} H(L) d\sigma(x)$$

recursively estimate derivative of L at some visible point 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

## Let's differentiate it



### CHALLENGES





### Complex light transport effects

Complex geometry

# PATH-INTEGRAL FOR DIFFERENTIABLE RENDERING

### FORWARD PATH INTEGRAL





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

## **DIFFERENTIAL PATH INTEGRAL**



and  $\dot{h}_n(x_n; x_{n-1}) = \int_{\mathcal{M}^{N-n}} \left[ \left( h_n^{(0)} \right) \cdot - h_n^{(0)} h_n^{(1)} \right] \prod_{n'=n+1}^N \mathrm{d}A(x_{n'})$ We now derive  $\partial I_N / \partial \pi$  in Eq. (25) using the recursive relations pro-Notice that  $h_0^{(0)} = f$  and  $\Delta h_{0,n'}^{(0)} = \Delta f_{n'}$ , where  $\Delta f_{n'}$  follows the vided by Eqs. (21) and (24). Let  $\dot{h}_{n-1}(x_{n-1}; x_{n-2})$ definition in Eq. (28). Letting n = 0 in Eq. (56) yields  $+\sum_{n'=n+1}^{N}\int \Delta h_{n,n'}^{(0)} V_{\overline{\partial \mathcal{M}}_{n'}}(\boldsymbol{x}_{n'}) \, \mathrm{d}\ell(\boldsymbol{x}_{n'}) \prod_{n< i \leq N} \mathrm{d}A(\boldsymbol{x}_{i}), \quad (56)$  $= \int_{\mathcal{M}} \left[ \dot{q}_{n-1} h_n + q_{n-1} (\dot{h}_n - h_n \kappa(\mathbf{x}_n) V(\mathbf{x}_n)) \right] dA(\mathbf{x}_n)$  $\dot{h}_0(\mathbf{x}_0) = \int_{\mathcal{M}^N} \left[ \dot{f}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}}) \sum_{n'=1}^N \kappa(\mathbf{x}_{n'}) V(\mathbf{x}_{n'}) \right] \prod_{n'=1}^N \mathrm{d}A(\mathbf{x}_{n'})$  $h_n^{(0)} \coloneqq \left[\prod_{n'=n+1}^N g(\mathbf{x}_{n'}; \mathbf{x}_{n'-2}, \mathbf{x}_{n'-1})\right] W_e(\mathbf{x}_N \to \mathbf{x}_{N-1}), \quad (52)$ +  $\int_{\overline{\partial M_n}} \Delta g_{n-1} h_n V_{\overline{\partial M_n}} d\ell(\mathbf{x}_n)$  $+ \sum_{n'=1}^{N} \int \Delta f_{n'}(\bar{\mathbf{x}}) \, V_{\overline{\partial \mathcal{M}}_{n'}} \, \mathrm{d}\ell(\mathbf{x}_{n'}) \prod_{0 < i \leq N} \mathrm{d}A(\mathbf{x}_{i}).$ (59)  $h_n^{(1)} \coloneqq \sum_{n'=n+1}^N \kappa(\mathbf{x}_{n'}) \, V(\mathbf{x}_{n'}),$ (53) where the integral domain of the second term on the right-hand  $= \int_{\mathcal{M}^{N-n+1}} \left\{ \dot{g}_{n-1} h_n^{(0)} + g_{n-1} \left[ \left( h_n^{(0)} \right)^{\cdot} - h_n^{(0)} h_{n-1}^{(1)} \right] \right\} \prod_{n'=k}^{N} \mathrm{d}A(\mathbf{x}_{n'})$ side, which is omitted for notational clarity, is  $\mathcal{M}(\pi)$  for each  $x_i$  $\Delta h_{n,n'}^{(0)} := h_n^{(0)} \, \Delta g(\mathbf{x}_{n'}; \mathbf{x}_{n'-2}, \mathbf{x}_{n'-1}) / g(\mathbf{x}_{n'}; \mathbf{x}_{n'-2}, \mathbf{x}_{n'-1}),$ (54)  $+ \sum_{n'=n+1}^N \int g_{n-1} \Delta h_{n,n'}^{(0)} V_{\overline{\partial \mathcal{M}}_{n'}}(\boldsymbol{x}_{n'}) \, \mathrm{d} \boldsymbol{\ell}(\boldsymbol{x}_{n'}) \prod_{n \leq i \leq N} \mathrm{d} \boldsymbol{A}(\boldsymbol{x}_i)$ Lastly, based on the assumption that  $h_0$  is continuous in  $x_0$ , Eq. (25) with  $i \neq n'$  and  $\overline{\partial \mathcal{M}}_{n'}(\pi)$ , which depends on  $x_{n'-1}$ , for  $x_{n'}$ . can be obtained by differentiating Eq. (23): It is easy to verify that Eqs. (55) and (56) hold for n = N - 1. We for  $0 \le n < n' \le N$ . We omit the dependencies of  $h_n^{(0)}$ ,  $h_n^{(1)}$ , and  $\frac{\partial I_N}{\partial \pi} = \frac{\partial}{\partial \pi} \int_{\mathcal{M}} h_0(\mathbf{x}_0) \, \mathrm{d}A(\mathbf{x}_0)$ +  $\int \Delta g_{n-1} h_n^{(0)} V_{\overline{\partial M_n}} d\ell(\mathbf{x}_n) \prod_{n'=n+1}^N dA(\mathbf{x}_{n'})$ now show that, if they hold for some 0 < n < N, then it is also  $\Delta h_{n,n'}^{(0)}$  on  $x_{n+1}, \ldots, x_N$  for notational convenience.  $= \int_{\mathcal{M}} \left[ \dot{h}_0(\mathbf{x}_0) - h_0(\mathbf{x}_0) \,\kappa(\mathbf{x}_0) \,V(\mathbf{x}_0) \right] \,\mathrm{d}A(\mathbf{x}_0)$ the case for n - 1. Let  $g_{n-1} := g(x_n; x_{n-2}, x_{n-1})$  for all  $0 < n \le N$ .  $= \int_{\mathcal{M}^{N-n+1}} \left[ \left( h_{n-1}^{(0)} \right) \cdot - h_{n-1}^{(0)} h_{n-1}^{(1)} \right] \prod_{n'=n}^{N} \mathrm{d}A(\mathbf{x}_{n'})$ We now show that, for all  $0 \le n < N$ , it holds that Then. +  $\int_{\partial M_0} h_0(\mathbf{x}_0) V_{\partial M_0}(\mathbf{x}_0) d\ell(\mathbf{x}_0)$ (60) $+ \sum_{n'=n}^{N} \int \Delta h_{n-1,n'}^{(0)} V_{\overline{\partial \mathcal{M}}_{n'}}(\mathbf{x}_{n'}) \, \mathrm{d}\ell(\mathbf{x}_{n'}) \prod_{n \leq i \leq N} \mathrm{d}A(\mathbf{x}_i).$ (58)  $h_n(x_n; x_{n-1}) = \int_{M^{N-n}} h_n^{(0)} \prod_{n'=n+1}^N dA(x_{n'}),$  $h_{n-1}(\mathbf{x}_{n-1}; \mathbf{x}_{n-2}) = \int_{M} g_{n-1} \int_{MN-n} h_n^{(0)} \prod_{n'=n+1}^N dA(\mathbf{x}_{n'}) dA(\mathbf{x}_n)$  $= \int_{\Omega_{N}} \left[ \dot{f}(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}}) \sum_{K=0}^{N} \kappa(\mathbf{x}_{K}) V(\mathbf{x}_{K}) \right] d\mu(\bar{\mathbf{x}})$  $= \int_{MN-n+1} h_{n-1}^{(0)} \prod_{n'=n}^{N} \mathrm{d}A(\mathbf{x}_{n'}),$ +  $\sum_{K=0}^{N} \int_{\Omega_{N,K}} \Delta f_K(\bar{\mathbf{x}}) V_{\overline{\partial M}_K} d\mu'_{N,K}(\bar{\mathbf{x}}).$ and (57) Thus, using mathematical induction, we know that Eqs. (55) and (56) hold for all  $0 \le n < N$ .

Full derivation in the paper

### **DIFFERENTIAL PATH INTEGRAL**



### **SOURCE OF DISCONTINUITIES**



#### Topology-driven

Visibility-driven

# REPARAMETERIZATIONS FOR SIMPLIFYING THE BOUNDARY TERM

### **REVISIT - DIFFERENTIAL IRRADIANCE**



### **DIFFERENTIAL IRRADIANCE**



### **DIFFERENTIAL DIRECT ILLUMINATION**



### **DIFFERENTIAL IRRADIANCE**



### REPARAMETERIZATION



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

### REPARAMETERIZATION



### REPARAMETERIZATION

Reparameterization for irradiance

$$E = \int_{\mathcal{L}(\pi)} L_e(\mathbf{y} \to \mathbf{x}) G(\mathbf{x}, \mathbf{y}) dA(\mathbf{y})$$

y

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

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

$$\uparrow$$
Fixed surface

Reparameterization for path integral

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

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

$$I = \int_{\Omega_0} f(\overline{\boldsymbol{x}}) \left| \frac{\mathrm{d}\mu(\overline{\boldsymbol{x}})}{\mathrm{d}\mu(\overline{\boldsymbol{p}})} \right| \mathrm{d}\mu(\overline{\boldsymbol{p}})$$
Fixed path space 
$$II$$

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

### **DIFFERENTIAL PATH INTEGRAL**

Original Original  $\frac{\mathrm{d}I}{\mathrm{d}\pi} = \int_{\Omega(\pi)} \frac{\mathrm{d}f(\overline{\mathbf{x}})}{\mathrm{d}\pi} \,\mathrm{d}\mu(\overline{\mathbf{x}}) + \int_{\partial\Omega(\pi)} g(\overline{\mathbf{x}}) \mathrm{d}\mu'(\overline{\mathbf{x}})$  $I = \int_{\Omega(\pi)} f(\overline{\mathbf{x}}) \, \mathrm{d}\mu(\overline{\mathbf{x}})$ No global parametrization required Pro:  $\overline{\boldsymbol{x}} = X(\overline{\boldsymbol{p}}, \pi)$ More types of discontinuities Con: Reparameterized

Reparameterized

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

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

Requires global parametrization X Con: Fewer types of discontinuities Pro:

### **DIFFERENTIAL PATH INTEGRAL**

### Differential path integral



# **MONTE CARLO ESTIMATORS**

### **ESTIMATING INTERIOR INTEGRAL**

(Reparameterized) Differential path Integral

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

Interior integral

. . .

**Boundary integral** 



- Can be estimated using identical path sampling far at glese at not ward rendering
  - Unidirectional path tracing
  - Bidirectional path tracing



### **ESTIMATING BOUNDARY INTEGRAL**



### **ESTIMATING BOUNDARY INTEGRAL**

(Reparameterized) Differential path Integral

$$\frac{\partial I}{\partial \pi} = \int_{\Omega_0} \frac{\partial}{\partial \pi} \left( f(\overline{x}) \left| \frac{d\mu(\overline{x})}{d\mu(\overline{p})} \right| \right) d\mu(\overline{p}) + \int_{\partial\Omega_0} g(\overline{p}) d\mu'(\overline{p})$$
  
where  $\overline{x} = X(\overline{p}, \pi)$   
Boundary integral

- Construct boundary segment

- 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**



#### **HANDLING CAUSTICS**





Reference

#### Equal-sample comparison



#### **HANDLING CAUSTICS**

Target image



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



#### SHAPE OPTIMIZATION



#### 112

#### RESULTS











# **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

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 reconstructed measurements 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



#### 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]



## Single-image dense BRDF sampling



#### **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)



# 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 BRDF and lighting [Romeiro and Zickler 2010]



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



Ambiguities between scattering parameters [Zhao et al. 2014]

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

Analysis-by-synthesis optimization:



- avoid local minima
- accelerate convergence





Neural network





## 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)



## 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]

# 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<sup>1</sup>, Wenzel Jakob<sup>2</sup>, and Tzu-Mao Li<sup>3</sup> <sup>1</sup>University of California, Irvine <sup>2</sup>EPFL <sup>3</sup>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 <u>shz@ics.uci.edu</u> Ioannis Gkioulekas Assistant Professor, RI Carnegie Mellon University igkioule@cs.cmu.edu