Inverse and differentiable rendering

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

• Take-home quiz 10 posted, due 4/23, worth 100 points.
• Will try to have feedback for all proposals by Friday.
Overview of today’s lecture

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

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

physically-accurate inverse rendering

photorealistic synthetic image
What I was doing in 2013:

- Mustard
- Whole milk
- Hand cream
- Shampoo
- Olive oil
- Robitussin
- Curacao
- Mixed soap
- Milk soap
- Liquid clay
- Reduced milk
- Coffee
- Wine
I wanted to make images such as this one

mixed soap

glycerine soap

olive oil

curacao

whole milk
Scattering: extremely multi-path transport

random walks inside volume

volumetric density $\rho$

scattering materialbedo $a$

phase function $f_r$
Acquisition setup
Analysis by synthesis (a.k.a. inverse rendering)

not scalable
solve by exhaustive search?

optimization problem

\[
\min_{m} \| \begin{array}{c}
\text{material} m \\
\text{render}(m)
\end{array} - \text{image data} \|^2
\]

Monte Carlo rendering

material \(m_2\)

several hours

not scalable
Analysis by synthesis (a.k.a. inverse rendering)

\[ \min_{m} \left\| \text{render}(m) \right\|^2 \]

- **Monte Carlo rendering**
- **material m**
- **image(m)**
- **material m**
- **material m + \partial m**

**Optimization problem**
Other scattering materials

- Everyday materials
  - [Gkioulekas et al. 2013]

- Industrial dispersions
  - [Gkioulekas et al. 2013]

- Computed tomography
  - [Geva et al. 2018]

- Woven fabrics
  - [Khungurn et al. 2015, Zhao et al. 2016]

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

- Clouds
  - [Levis et al. 2015, 2017]

- Optical tomography
  - [Gkioulekas et al. 2016]
Making sense of global illumination

reflectance

scattering

differentiable rendering: image gradients with respect to arbitrary X

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 \[ \frac{\partial \text{loss}(X)}{\partial X} \]

Monte-Carlo rendering
Differentiable rendering and deep learning

\[ \pi = (R_{\text{physics}})^{-1}(\text{Img}) \]

\[ \text{Img} = R_{\text{physics}}(\pi) \]

needs to be differentiable for training with backpropagation

force input and output images to be the same
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

\[ \frac{d}{d\pi} \int_{a(\pi)}^{b(\pi)} f(x, \pi) \, dx \quad ? \quad \int_{a(\pi)}^{b(\pi)} \frac{d}{d\pi} f(x, \pi) \, dx \]

Move derivative inside integral

Account for changes in integration limits

\[ + f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi} \]

Account for discontinuities of integrand that depend on \( \pi \)

\[ + \sum_{i} \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi} \]
A simple example

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

\[
\frac{d}{d\pi} \int_{0}^{4\pi} f(x, \pi) dx = \int_{0}^{2\pi} \frac{d}{d\pi} 0 dx + \int_{2\pi}^{4\pi} \frac{d}{d\pi} 1 dx + \frac{d(4\pi)}{d\pi} - 0 \frac{d0}{d\pi} + (0 - 1) \frac{d(2\pi)}{d\pi}
\]

Account for changes in integration limits

Account for discontinuities of integrand that depend on \( \pi \)

Move derivative inside integral
Leibniz integral rule

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

**Interior integral**

**Boundary terms**

- Account for changes in integration limits

\[
\int_{a(\pi)}^{b(\pi)} f(b(\pi), \pi) \frac{db(\pi)}{d\pi} - f(\alpha(\pi); \pi) \frac{da(\pi)}{d\pi}
\]

- Account for discontinuities of integrand that depend on \(\pi\)

\[
\sum_{i} \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi}
\]
Simplified Leibniz integral rule

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

\[
\frac{d}{d\pi} \int_a^b f(x, \pi) \, dx = \int_a^b \frac{d}{d\pi} f(x, \pi) \, dx
\]

**Interior integral**

\[
\int_a^b f(x, \pi) \, dx
\]

**Boundary terms**

\[
\begin{align*}
\quad & f(b(\pi), \pi) \frac{db(\pi)}{d\pi} = f(a(\pi), \pi) \frac{da(\pi)}{d\pi} \\
& \quad + \sum_{i} \left( f(c_i(\pi)^-, \pi) - f(c_i(\pi)^+, \pi) \right) \frac{dc_i(\pi)}{d\pi}
\end{align*}
\]

Move derivative inside integral

Account for changes in integration limits when:
- Integration limits are independent of \( \pi \).
- Integrand discontinuities are independent of \( \pi \).

Account for discontinuities of integrand that depend on \( \pi \).
Reynolds transport theorem

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

Reynolds transport theorem [1903]
Generalization of the Leibniz rule

Interior integral

Boundary integral

boundary of domain \( \Omega \)

discontinuity points \( \cup \) boundary of domain \( \Omega \)
(if they depend on \( \pi \))
DIFFERENTIATING DIRECT ILLUMINATION
Direct illumination integral

Radiance from $x$:

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

Monte Carlo rendering:

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

$$I \approx \sum_s f_r(\omega_i^s, \omega_o) L_i(\omega_i^s) (n \cdot \omega_i^s) \frac{1}{p(\omega_i^s)}$$
Differential direct illumination

Differential radiance from $x$:

\[
\frac{dI}{d\pi} = \frac{d}{d\pi} \int_{\mathbb{H}^2} f_r(\omega_i, \omega_o) L_i(\omega_i) (n \cdot \omega_i) \, d\sigma(\omega_i)
\]
Differential direct illumination: local parameters

\[ \frac{dI}{d\pi} = \int \int \frac{f_r((\omega_i, \omega_o)) L_i(\omega_i) (n \cdot \omega_i)}{p(\omega_i)} \, d(\omega_i) \]

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

\(\pi\): local parameters
- BRDF parameters
- shading normal
- illumination brightness

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]
Differentiate entire contribution [Zeltner et al. 2021]

\[ \frac{dI}{d\pi} \approx \int \frac{d}{d\pi} \left\{ f_r(\omega_i, \omega_o, \pi)L_i(\omega_i)(n \cdot \omega_i) \right\} d\sigma(\omega_i) \]

Just move derivative inside integral

Monte Carlo estimation:
- Sample random directions \( \omega_i^s \) from PDF \( p(\omega_i, \pi) \)
- Form estimator

\[ \frac{dI}{d\pi} \approx \sum S \frac{d}{d\pi} \left\{ f_r(\omega_i^s, \omega_o, \pi)L_i(\omega_i^s)(n \cdot \omega_i^s) \right\} \]

Differentiate entire contribution [Zeltner et al. 2021]

\( \pi \): local parameters
- BRDF parameters
Differential direct illumination: global parameters

Differential radiance from $x$:

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

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

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

Need to use full Reynolds transport theorem
Discontinuities in the integrand

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

\[ f(\omega_i) \]

\( \pi \): size of the emitter

Integrand

Discontinuous points

(\( \pi \)-dependent)
Applying the Reynolds transport theorem

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

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

Interior integral (same as for local parameters)

Boundary integral

Integrand \( f(\omega_i) \)

Discontinuous points (\( \pi \)-dependent)

[Ramamoorthi et al. 2007, Li et al. 2019]
Reparameterizing the direct illumination integral

### Hemispherical integral

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

### Surface integral

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

Includes visibility, fall-off, and foreshortening terms.
Reparameterizing the direct illumination integral

**Hemispherical integral**

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

- Discontinuous
- Constant domain

**Surface integral**

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

- Continuous
- Evolving domain

Change of variables
Differentiating the hemispherical integral

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

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

\( \pi \): size of the emitter

Low | High

Discontinuities of \( f \)
Differentiating the area integral

\[ I = \int_{\mathcal{L}(\pi)} f(y \to x) G(x, y) dA(y) \]

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

\[ \pi: \text{size of the emitter} \]
Sources of discontinuities

- **Boundary edge**
- **Sharp edge**
- **Silhouette edge**

**Topology-driven**
- Sensor pointing to a boundary edge

**Visibility-driven**
- Sensor pointing to a silhouette edge

Silhouette detection
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

- INIT. MESH: 0.0115
- SOFTRA: 0.0039
- PYTORCH3D: 0.0091
- MITSUBA 2: 0.0065
- NVDIFFRACT: 0.0022
- Luan et al. 2021: **0.0016**
- GROUND TRUTH

- INIT. MESH: 0.0878
- SOFTRA: 0.0053
- PYTORCH3D: 0.0066
- MITSUBA 2: 0.0071
- NVDIFFRACT: 0.0023
- Luan et al. 2021: **0.0010**
- GROUND TRUTH

Relative err: 0% - 30%
Sources of discontinuities

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

[Image: Silhouette edge diagram]

[Gargallo et al., ICCV 2007]
DIFFERENTIATING GLOBAL ILLUMINATION
Images as path integrals

\[ I(\pi) = \int_{\mathcal{P}} f(\bar{x}; \pi) d\bar{x} \]

- \( \bar{x} \rightarrow \) Light path, set of ordered vertices on surfaces
- \( \mathcal{P} \rightarrow \) Space of valid paths
- \( f(\bar{x}) \rightarrow \) Path contribution,
  includes geometric terms (visibility, fall-off) & local terms (BRDF, foreshortening, emission)
Monte Carlo rendering: approximating path integrals

\[ I(\pi) \approx \sum_{i=1}^{N} \frac{f(\bar{x}_i; \pi)}{p(\bar{x}_i; \pi)} \]

\[ MC(\pi) \]

\( \bar{x}_i \rightarrow \text{Randomly sampled light paths} \)

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

Algorithms such as path tracing, bidirectional path tracing, etc. sample paths.
How can we approximate the derivative of the image?

\[
\frac{\partial I}{\partial \pi}(\pi) \approx ?
\]
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 \( \varepsilon \)
- Very inaccurate for large \( \varepsilon \)
- 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 \( f(x;\pi) = \text{constant} \))
- Rendering produces enormous, non-local computational graphs.
DIFFERENTIATING GLOBAL ILLUMINATION WITH RESPECT TO LOCAL PARAMETERS
Images as path integrals

\[ I(\pi) = \int_{\mathbb{P}} f(\bar{x}; \pi) d\bar{x} \]

- \( \bar{x} \rightarrow \) Light path, set of ordered vertices on surfaces
- \( \mathbb{P} \rightarrow \) Space of valid paths
- \( f(\bar{x}) \rightarrow \) 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

\[
\frac{\partial I}{\partial \pi}(\pi) = ?
\]

\(\tilde{x}\) → Light path, set of ordered vertices on surfaces
\(\mathcal{P}\) → Space of valid paths
\(f(\tilde{x})\) → Path contribution, includes geometric terms (visibility, fall-off) & local terms (BRDF, foreshortening, emission)

Assume \(\mathcal{P}\) is independent of \(\pi\)
Derivatives of images as path integrals

\[
\frac{\partial I}{\partial \pi}(\pi) = \int_{\mathbb{P}} \frac{\partial f}{\partial \pi}(\overline{x}; \pi) d\overline{x}
\]

differentiation under the integral sign

\(\overline{x}\) \rightarrow \text{Light path, set of ordered vertices on surfaces}

\(\mathbb{P}\) \rightarrow \text{Space of valid paths}

\(f(\overline{x})\) \rightarrow \text{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)

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

This term is generally easy to compute during path tracing.

- \(\bar{x}_i\) → Randomly sampled light paths
- \(p(\bar{x}_i)\) → Probability of sampling a path

Sample paths using path tracing etc.
Score estimator

\[
f(\vec{x}; \pi) = \prod_{b=1}^{B} f_s(x_{b-1} \rightarrow x_b \rightarrow x_{b+1}; \pi) \frac{V(x_{b-1} \leftrightarrow x_b)}{\|x_{b-1} - x_b\|^2}
\]

[ Foreshortening terms are included in the BRDF ]

\[
\frac{\partial f}{\partial \pi}(\vec{x}; \pi) = \prod_{b=1}^{B} f_s(x_{b-1} \rightarrow x_b \rightarrow x_{b+1}; \pi) \frac{V(x_{b-1} \leftrightarrow x_b)}{\|x_{b-1} - x_b\|^2}
\]

[ 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

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

\( \bar{x}_i \rightarrow \text{Randomly sampled light paths} \)

\( p(\bar{x}_i) \rightarrow \text{Probability of sampling a path} \)
Compare with...

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

\(\bar{x}_i\) \rightarrow \text{Randomly sampled light paths}

\(p(\bar{x}_i)\) \rightarrow \text{Probability of sampling a path}
Even simpler: use autodiff

\[
\frac{\partial I}{\partial \pi} (\pi) \approx \sum_{i=1}^{N} \frac{\text{autodiff}(f(\overline{x}_i; \pi))}{p(\overline{x}_i; \pi)}
\]

- 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.](image)
Compute an estimate of the derivative

derivative wrt volumetric density

derivative wrt BRDF

derivative wrt normal
Comparison with finite differences

Note: Finite differences are great for testing the correctness of your gradient code.
Compute a derivative of the estimate

- A lot more general.
- GPU implementation.

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

derivative wrt volumetric density
Derivatives of images as path integrals

\[
\frac{\partial I}{\partial \pi} (\pi) = \int_{\mathcal{P}} \frac{\partial f}{\partial \pi} (\bar{x}; \pi) \, d\bar{x}
\]

differentiation under the integral sign

\( \bar{x} \) → Light path, set of ordered vertices on surfaces

\( \mathcal{P} \) → Space of valid paths

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

Assume \( \mathcal{P} \) is independent of \( \pi \)
Derivatives of images as path integrals

\[ \frac{\partial I}{\partial \pi}(\pi) = \int_{\mathbb{P}} \frac{\partial f}{\partial \pi}(\bar{x}; \pi) d\bar{x} \]

differentiation under the integral sign

What about parameters \( \pi \) 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

\[
L(x, \omega; \pi) = \int _{G(\pi)} L(x' \rightarrow x; \pi) f(x' \rightarrow x, \omega; \pi) V(x' \leftrightarrow x; \pi)dA(x')
\]

- \(L\) → 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

\[ L(x, \omega; \pi) = \int_{V(x, \pi)} L(x' \rightarrow x; \pi) f(x' \rightarrow x, \omega; \pi) dA(x') \]

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

Can we just move the integral inside?
Let’s differentiate it

\[
\frac{\partial}{\partial \pi} L(x, \omega; \pi) = \frac{\partial}{\partial \pi} \int_{V(x, \pi)} L(x' \rightarrow x; \pi) f(x' \rightarrow x, \omega; \pi) dA(x')
\]

\( L \rightarrow \) Radiance at a point and direction

\( V \rightarrow \) All visible surfaces in the scene

\( f \rightarrow \) Reflection, foreshortening, and fall-off

Can we just move the integral inside?

• No. What can we do?
Let's differentiate it

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

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

What are the “boundary” and discontinuities of $V$?
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

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

- 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
Boundary edge detection and sampling

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
Let’s differentiate it

\[
\frac{\partial}{\partial \pi} L(x, \omega; \pi) = \int_{V(x, \pi)} F \left( \frac{\partial}{\partial \pi} L \right) dA(x) + \int_{\partial V(x, \pi)} H(L) d\sigma(x)
\]

render derivative of \( L \) at some visible point

render \( L \) at some boundary (silhouette) point

Not terribly good:
- As we ray trace, we need to recompute silhouette
- Branching of two at each recursion
CHALLENGES

Complex light transport effects

Complex geometry
REPARAMETERIZATION APPROACHES
THE REYNOLDS TRANSPORT THEOREM

\[ \partial_t \int_D f = \int_D \partial_t f + \int_{\partial D} f \vec{v} \cdot \vec{n} \]

- Interior term
- Edge term

\( D \): Set of continuous points

\( \partial D \): Set of discontinuous points
CONVERTING EDGE-SAMPLES TO AREA-SAMPLES

\[ \int_{\partial D} f \vec{v} \cdot \vec{n} \] is estimated through edge-samples

Goal: Rewrite \[ \int_{\partial D} f \vec{v} \cdot \vec{n} \] into area integral \[ \int_D g \]

\[ \int_D \nabla \cdot (\vec{v}_\theta f) \] can be estimated through area-samples
THE DIVERGENCE THEOREM

\[ \int_{\partial D} \vec{f} \cdot \vec{n} \quad \iff \quad \int_{D} \nabla \cdot \vec{f} \]

[Gauss 1813]
QUICK RECAP

• Used *Reynolds transport theorem* to find the boundary integral

\[
\int_{\partial D} f \mathbf{\nabla} \cdot \mathbf{n}
\]

• Rewrote

\[
\int_{\partial D} f \mathbf{\nabla} \cdot \mathbf{n}
\]

to

\[
\int_D \nabla \cdot (\mathbf{\nabla}_\theta f)
\]
using the *divergence theorem*.

• Have to define the vector field \( \mathbf{\nabla}_\theta \) over domain \( D \)
A 2D EXAMPLE SCENE

\[ \omega \in \Omega, \text{ the domain of integration} \]

\[ \omega_1^{(b)}, \omega_2^{(b)} \in \partial \Omega, \text{ the discontinuous set} \]
VELOCITY $\vec{V}$ : THE BOUNDARY DERIVATIVE

$\partial_\theta \omega_i^{(b)}$: Derivative of boundary position w.r.t $\theta$
WARP FIELD $\nu_\theta$ : EXTENSION OF $\vec{V}$ TO ALL POINTS

$\nu_\theta(\omega)$

$\nu_\theta$ : defined over $D$

$\vec{V}$ : defined over $\partial D$
VALIDITY OF $\vec{V}_\theta$

**Rule 1: Continuous**

![Graph showing the validity of $V_\theta$ for continuous functions](image)
Rule 2: Boundary Consistent
CONSTRUCTING $\vec{V}_\theta$

Attempt 1  
Find $\partial_\theta \omega$ through *implicit derivative*

(Incorrect)

$$y = \text{INTERSECT}(\omega, \theta) \implies \partial_\theta \omega = \frac{\partial \omega y}{\partial \theta y}$$

At all points (not just boundaries)

- Boundary consistent
- Not continuous
CONSTRUCTING $\mathbf{\theta}$

Attempt 2: Filter Attempt 1 with a Gaussian filter

$\int_{\Omega'} k(\omega, \omega') \frac{\partial \omega y}{\partial \theta y}$

$k(...)$ = Gaussian filter

+ Continuous
- Not boundary consistent
BOUNDARY-AWARE WEIGHTING

Goal: Find weights $k(\omega, \omega')$ s.t. $
\n\tilde{v}_\theta = \frac{\partial \omega y}{\partial \theta y}$

at boundaries.

Ideal weighting function

Approach Dirac delta near boundaries
PATH-INTEGRAL FOR DIFFERENTIABLE RENDERING
FORWARD PATH INTEGRAL

\[ I = \int_{\Omega} f(\mathbf{x}) \, d\mu(\mathbf{x}) \]

Measurement contribution function
Area-product measure
Path space

Light path \( \mathbf{x} = (x_0, x_1, x_2, x_3) \)
Path Integral

\[ I = \int_{\Omega} f(\mathbf{x}) \, d\mu(\mathbf{x}) \]

\[ \frac{dI}{d\pi} = ? \]

Full derivation in the paper
DIFFERENTIAL PATH INTEGRAL

\[ I = \int_{\Omega} f(\mathbf{x})d\mu(\mathbf{x}) \]

A generalization of Reynolds theorem

\[ \frac{dI}{d\pi} = \int_{\Omega} \frac{d}{d\pi} f(\mathbf{x})d\mu(\mathbf{x}) + \int_{\partial\Omega} g(\mathbf{x})d\mu'(\mathbf{x}) \]

Path Integral

Differential Path Integral

Original light path

Boundary light path

Types of discontinuity edge:

Boundary segment

Path space

Boundary path space

Boundary integral

89
SOURCE OF DISCONTINUITIES

Boundary edge  

Sharp edge  

Silhouette edge  

Topology-driven  

Visibility-driven  

Boundary edge  

Sharp edge  

Silhouette detection
TEXTURE PARAMETERIZATION FOR SIMPLIFYING THE BOUNDARY TERM
REPARAMETERIZATION

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

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

\[ y = X(p, \pi) \]

where \( X(\cdot, \pi) \) is one-to-one and continuous

Reparameterization with \( y = X(p, \pi) \):

\[ E = \int_{\mathcal{L}_0} L_e(y \to x) G(x, y) \left| \frac{dA(y)}{dA(p)} \right| \, dA(p) \]
REPARAMETERIZATION

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

\[
\frac{dE}{d\pi} = \int_{\mathcal{L}(\pi)} \frac{df}{d\pi} \, dA + \int_{\partial \mathcal{L}(\pi)} g \, dl
\]

\[
y = x(p, \pi)
\]
Reparameterization for irradiance

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

\[ y = X(p, \pi) \]

\[ E = \int_{\mathcal{L}_0} L_e(y \to x) G(x, y) \left| \frac{dA(y)}{dA(p)} \right| \, dA(p) \]

Fixed surface

Reparameterization for path integral

\[ I = \int_{\Omega(\pi)} f(\bar{x}) \, d\mu(\bar{x}) \]

\[ \bar{x} = X(\bar{p}, \pi) \]

\[ I = \int_{\Omega_0} f(\bar{x}) \left| \frac{d\mu(\bar{x})}{d\mu(\bar{p})} \right| \, d\mu(\bar{p}) \]

Fixed path space

\[ \prod_i \left| \frac{dA(x_i)}{dA(p_i)} \right| \]
**DIFFERENTIAL PATH INTEGRAL**

Original

\[ I = \int_{\Omega(\pi)} f(\bar{x}) \, d\mu(\bar{x}) \]

\[ \bar{x} = \chi(p, \pi) \]

Reparameterized

\[ I = \int_{\Omega_0} f(\bar{x}) \left| \frac{d\mu(\bar{x})}{d\mu(p)} \right| \, d\mu(p) \]

Original

\[ \frac{dI}{d\pi} = \int_{\Omega(\pi)} \frac{df(\bar{x})}{d\pi} \, d\mu(\bar{x}) + \int_{\partial\Omega(\pi)} g(\bar{x}) \, d\mu'(\bar{x}) \]

**Pro:** No global parametrization required

**Con:** More types of discontinuities

Reparameterized

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

**Con:** Requires global parametrization \( \times \)

**Pro:** Fewer types of discontinuities
DIFFERENTIAL PATH INTEGRAL

Differential path integral

\[
\frac{dI}{d\pi} = \int_{\Omega(\pi)} \frac{df(x)}{d\pi} \, d\mu(x) + \int_{\partial\Omega(\pi)} g(x) \, d\mu'(x)
\]

\[
\frac{dI}{d\pi} = \int_{\Omega_0} \frac{d}{d\pi} \left( f(x) \left| \frac{d\mu(x)}{d\mu(p)} \right| \right) d\mu(p) + \int_{\partial\Omega_0} g(p) \, d\mu'(p)
\]

Topology-driven

Visibility-driven
MONTE CARLO ESTIMATORS
(Reparameterized) Differential path Integral

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

- Interior integral
- Boundary integral

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

Original light path

Different MC estimators
ESTIMATING BOUNDARY INTEGRAL

\[
\frac{\partial I}{\partial \pi} = \int_{\Omega_0} \frac{\partial}{\partial \pi} \left( f(x) \left| \frac{d\mu(x)}{d\mu(p)} \right| \right) d\mu(p) + \int_{\partial \Omega_0} g(p) d\mu'(p)
\]

Boundary integral

Silhouette detection
[Li et al. 2018, Zhang et al. 2019]
ESTIMATING BOUNDARY INTEGRAL

(Reparameterized) Differential path Integral

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

where \( \bar{x} = x(\bar{p}, \pi) \)

- 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

Equal-sample comparison

Reference

Negative

Zero

Positive

5.7 s

0.3 s

0.5 s

[Zhang et al. 2019]

[Loubet et al. 2019]

Ours
Target image

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

Complex light transport effects

Equal-sample comparison

101.3 s

153 s

19.7 s

Reference

Negative

Zero

Positive

[Zhang et al. 2019]

[Loubet et al. 2019]

Ours
**Equal-sample comparison**

- **[Zhang et al. 2019]**: 101.3 s
- **[Loubet et al. 2019]**: 153 s
- **Ours (unidirectional)**: 19.7 s
- **Ours (bidirectional)**: 19.7 s

**Reference**

*Note: The reference image is not provided in the text.*
HANDLING CAUSTICS

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

Target image

[Image of a target image with a glass object and a spotlight]
SHAPE OPTIMIZATION

Target image

Initial

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

Cross-sectional shape (displacement x 20)
Applications
Inverse scattering [Gkioulekas et al. 2013]
Acquisition setup

Invert using differentiable rendering
Synthetic renderings

- mixed soap
- glycerine soap
- olive oil
- curacao
- whole milk
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
Examples

Groundtruth

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

- Single-photon avalanche photodiode (SPAD)
- Picosecond laser
- Galvo mirror

[Image of a SPAD-based lidar system with labeled components: galvo mirror, picosecond laser, single-photon avalanche photodiode (SPAD).]
NLOS shape optimization [Tsai et al. 2019]

Simulated time-of-flight data

visible surface
source and sensor
occluder
NLOS scene

100,000 vertices
NLOS shape optimization [Tsai et al. 2019]

Measured time-of-flight data
Reflectometry from interreflections [Shem-Tov et al. 2020]

- Many measurements (2D scan of light & camera)
- Intensities map directly to BRDF entries

- 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

Groundtruth

Optimized shape

~ 6.3x better parameter recovery

~ 11.2x better parameter recovery
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:

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

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

initialize \( \pi \leftarrow \pi_0 \)

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

while (not converged)
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)

Learned initializations help:
- avoid local minima
- accelerate convergence

Analysis-by-synthesis optimization:
\[
\min_{\text{scene unknowns } \pi} \text{loss} \left[ \text{render} \left( \text{scene unknowns } \pi \right) \right],
\]

Stochastic gradient descent (e.g., Adam):
- initialize \( \pi \leftarrow \pi_0 \)
- update \( \pi \leftarrow \pi + \eta \frac{d\text{loss}(\pi)}{d\pi} \)

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:

$$\min_{\text{scene unknowns } \pi} \text{loss} [\text{render}(\text{scene unknowns } \pi)]$$

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

initialize $$\pi \leftarrow \pi_0$$

while (not converged) update $$\pi \leftarrow \pi + \eta \frac{d\text{loss}(\pi)}{d\pi}$$

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]
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 **super** important:
• Approximate reconstruction assuming direct lighting is usually good enough.
• Coarse-to-fine schemes work well.

Parameterizations are **super** important:
• Loss functions very non-linear and change shape easily.
• Working with meshes is a pain (topology is awful and not (easily?) differentiable).

You don’t always need **Monte Carlo** differentiable rendering:
• If you don’t have strong global illumination, just use direct lighting.
• A lot of research in computer vision on differentiable rasterizers.

Remember that you are doing optimization:
• Unbiased and consistent gradients are very expensive to compute.
• Biased and/or inconsistent gradients can be very cheap to compute.
• Often, biased and/or inconsistent gradients are enough for convergence.
• **Stochastic** gradient descent matters a lot.
Physics-Based Differentiable Rendering
A Comprehensive Introduction
Shuang Zhao¹, Wenzel Jakob², and Tzu-Mao Li³
¹University of California, Irvine  ²EPFL  ³MIT 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
ghz@ics.uci.edu

Ioannis Gkioulakas
Assistant Professor, RI
Carnegie Mellon University
igkioulak@cs.cmu.edu