Unbiased Warped-Area Sampling for Differentiable Rendering

Sai Praveen Bangaru
Tzu-Mao Li
Frédéric Durand
3D Scene Description
Parameters $\theta$

Vertex List
- $v_1$ 0.92 -0.9 1.8
- $v_2$ 0.42 -0.9 1.8
- $v_3$ -0.92 0.9 1.8
- $v_4$ 0.65 -0.9 1.8
- $v_5$ 1.21 -0.9 1.8
- $v_6$ -1.21 -0.9 1.8
- $v_7$ 1.21 -0.9 1.8
- $v_8$ 1.21 -0.9 1.8
- $v_9$ 1.21 -0.9 1.8

Index List
- $p_1$ 0 1 2
- $p_2$ 2 3 4
- $p_3$ 4 1 2

Light List
- $L_1$ 0.5 0.0 -0.5
- $L_2$ -0.5 0.0 -0.5
- $L_3$ 0.5 0.0 -0.5

Rendering Method
$\text{render}(\theta)$

2D Image
DIFFERENTIABLE RENDERING

3D Scene Description
Parameters $\theta$

Vertex List
- $v_1: 0.92, -0.9, 1.8$
- $v_2: 0.42, -0.9, 1.8$
- $v_3: -0.92, 0.9, 1.8$
- $v_4: 0.65, -0.9, 1.8$
- $v_5: 1.21, -0.9, 1.8$
- $v_6: -1.21, -0.9, 1.8$
- $v_7: 1.21, -0.9, 1.8$
- $v_8: 1.21, -0.9, 1.8$
- $v_9: 1.21, -0.9, 1.8$

Index List
- $p_1: 0, 1, 2$
- $p_2: 2, 3, 4$
- $p_3: 3, 4, 1$

Light List
- $L_1: 0.5, 0.0, -0.5$
- $L_2: -0.5, 0.0, -0.5$
- $L_3: 0.5, 0.0, -0.5$

Rendering Method
$\text{render}(\theta)$

Differentiable Rendering Method
$\text{render}'(\theta)$

2D Image $I$

Differential $\frac{\partial I}{\partial \theta}$
WHY DIFFERENTIABLE RENDERING?

Motivation 1

3D Scene Description
- Vertex List
  - v1 0.92 -0.9 1.8
  - v2 0.42 -0.9 1.8
  - v3 -0.92 0.9 1.8
  - v4 0.65 -0.9 1.8
  - v5 1.21 -0.9 1.8
  - v6 -1.21 -0.9 1.8
  - v7 1.21 -0.9 1.8
  - v8 1.21 -0.9 1.8
  - v9 1.21 -0.9 1.8

- Index List
  - p1 0 1 2
  - p2 1 2 3
  - p3 3 4 1

- Light List
  - L1 0.5 0.0 -0.5
  - L2 -0.5 0.0 -0.5
  - L3 0.5 0.0 -0.5

Parameters θ

Inverting rendering process

\[ \text{render}^{-1}(\theta) \]

\[ \theta_0 \rightarrow \theta_i \]

Gradient descent on \( \theta_i \)

Analysis-by-Synthesis

Image \( I \)

\[ \text{render}(\theta) \]

\[ \delta \]

\[ \text{inverse renderer} \]

\[ \text{render-1(I)} \]
WHY DIFFERENTIABLE RENDERING?

Motivation 2  →  Deep Learning (adversarial robustness, etc..)

3D Scene

- Vertex List:
  - v1: 0.92 -0.9 1.8
  - v2: 0.42 -0.9 1.8
  - v3: -0.92 0.4
  - v4: 0.65
  - v5: 1.21
  - v6: -1.21

- Index List:
  - p1: 0 1 2
  - p2: 2 3 4
  - p3: 4 1 2

- Light List:
  - L1: 0.5 0.0 -0.5
  - L2: -0.5 0.0 -0.5
  - L3: 0.5 0.0 -0.5

2D Image

- render(θ)

Backpropagation
SCENE PARAMETER DERIVATIVES

Image $I$

θ: Object Translation

θ: Vertex position

θ: Camera Rotation

Differential $\frac{\partial I}{\partial \theta}$
AUTO-DIFF HAS A VISIBILITY PROBLEM

Sudden discontinuity $\rightarrow$ Auto-diff fails due to edges

Smooth function $\rightarrow$ Auto-diff computes correct derivative
RASTERIZATION APPROACHES ARE LIMITED

Key Idea: **Analytical occupancy**

- [Jalobeanu 2004]

- [de La Gorce 2008]

Key Idea: **Approximating visibility**

- **Soft Rasterizer** [Liu 2019]

- **Neural 3D Mesh Renderer** [Kato 2017]
RENDERING AS AN INTEGRAL

Goal: Differentiate these integrals
MONTE CARLO ESTIMATION

\[ \int_D \approx \sum_{\{\cdot\}} = \text{RGB pixel value} \]

*Integral over pixel space D*

*Summation over area-samples*
DISCONTINUOUS INTEGRANDS

\[ \partial_\theta \int_D \]

*Integral over pixel space D*

\[ \sum_{\{\theta\}} \partial_\theta \]

*Summation over area-samples*

*(Incorrect)* Attempt 1 \[\rightarrow\] Apply auto-diff to summation
Challenges for Edge-Sampling

Arbitrary silhouette sampling is hard!
EDGE-SAMPLING HAS TROUBLE WITH SPECULAR REFLECTIONS

(Near-)Perfect Mirror

Rendering Caustics

Manifold-Exploration
MLT
[Jakob 2012]

Natural Constraint Representation for MLT
[Kaplanyan 2014]
AREA-SAMPLING

Reparameterizing Discontinuous Integrands for Differentiable Rendering [Loubet 2019]

Transform samples with $\theta$. Avoids discontinuities.

Heuristic Approximation! May not work for all samples.
SUMMARY OF METHODS

**Rasterization**
- Approximate visibility
- No secondary effects

**Edge-sampling**
- Fast
- Exact derivative
- Depth complexity
- No perfect specularities
- Complex data structures

**Area-sampling**
- Fast (No complex sampling)
- Approximate derivative
OUR APPROACH
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

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

is estimated through edge-samples
THE DIVERGENCE THEOREM

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

[Gauss 1813]
APPLYING THE DIVERGENCE THEOREM TO THE EDGE INTEGRAL

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

Solution: Rewrite $\int_{\partial D} f \vec{v} \cdot \vec{n}$ into $\int_D \nabla \cdot (\vec{\nabla}_0 f)$

$\int_D \nabla \cdot (\vec{\nabla}_0 f)$ can be estimated through area-samples.
QUICK RECAP

• Used *Reynolds transport theorem* to find the boundary integral
  \[ \int_{\partial D} f \mathbf{v} \cdot \mathbf{n} \]

• Rewrote \[ \int_{\partial D} f \mathbf{v} \cdot \mathbf{n} \] to \[ \int_{D} \nabla \cdot (\mathbf{v}_\theta f) \] using the *divergence theorem*.

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

$\omega \in \Omega$, the domain of integration

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

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

$\mathcal{V}_\theta :$ defined over $D$

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

Rule 1: Continuous

$\nu_\theta(\omega)$

$\nu_\theta(\omega)$
Rule 2: Boundary Consistent
INTERPOLATION WITHOUT KNOWLEDGE OF BOUNDARIES

Available quantities
- Origin point
- Ray
- Intersection
- Primitive

No access to discontinuity points
CONSTRUCTING $\vec{V}_\theta$

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

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

At all points (not just boundaries)

+ Boundary consistent
- Not continuous

(Incorrect)
CONSTRUCTING $\vec{V}_\theta$

Attempt 2 Filter Attempt 1 with a Gaussian filter

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

\[ k(\cdot, \cdot) = \text{Gaussian filter} \]

+ Continuous

- Not boundary consistent
BOUNDARY-AWARE WEIGHTING

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

Ideal weighting function

Approach Dirac delta near boundaries
BOUNDARY-AWARE WEIGHTING

Implicit Boundary through geometric normals

\[ \langle \omega, n \rangle = 0 \]

at boundaries
CONSTRUCTING $\vec{V}_\theta$

Our Approach $\rightarrow$ Filter Attempt 1 with harmonic weights

$$k(\omega, \omega') = \frac{1}{D(\omega, \omega') + B(\omega')}$$

- Distance function
- Boundary test

+ Boundary consistent
+ Continuous
1. Sample **path** using path tracer \((N \text{ paths})\)

For each bounce:

2. Sample **auxiliary** rays \((N' \text{ rays})\)

3. Compute boundary term \(B()\) locally

4. Compute weight \(k(,,)\) and \(\partial_\theta \omega\)

5. Find weighted mean
Russian Roulette

\[ \sum \frac{k(\omega, \omega'_i)g(\omega'_i)}{\sum k(\omega, \omega'_i)} \]

\[ N' \sim \text{GEOM}(p) \]

Variance Reduction

Relationship with Reparameterization

\[ \mathcal{V}_\theta(\omega) \leftrightarrow \mathcal{T}(\omega; \theta) \]

\[ \mathcal{V}_\theta(\omega) = [\partial_\theta \mathcal{T}(\omega; \theta)]_{\theta=\theta_0} \]
RESULTS
VARIANCE COMPARISON WITH EDGE-SAMPLING

Pot

Image $I$

Reference Derivative

Li et al. 2018

Ours without Russian roulette

Ours with Russian roulette

HEDGE
Rotating cylindrical objects present a complicated scenario for area-sampling
BIAS COMPARISON WITH REPARAMETERIZATION

Extremely complex geometry like foliage can cause heuristic to fail
POSE ESTIMATION CAN FAIL WITH BIASED GRADIENTS

Multiple Initializations
CONTRIBUTIONS

**Edge-integral to Area-integral**

**Warp field conditions**

**Harmonic interpolation**

\[ k(\omega, \omega') = \frac{1}{D(\omega, \omega') + B(\omega')} \]
ACKNOWLEDGEMENTS

This project was funded by the Toyota Research Institute.

Code available on redner
Code *coming soon* on mitsuba-2