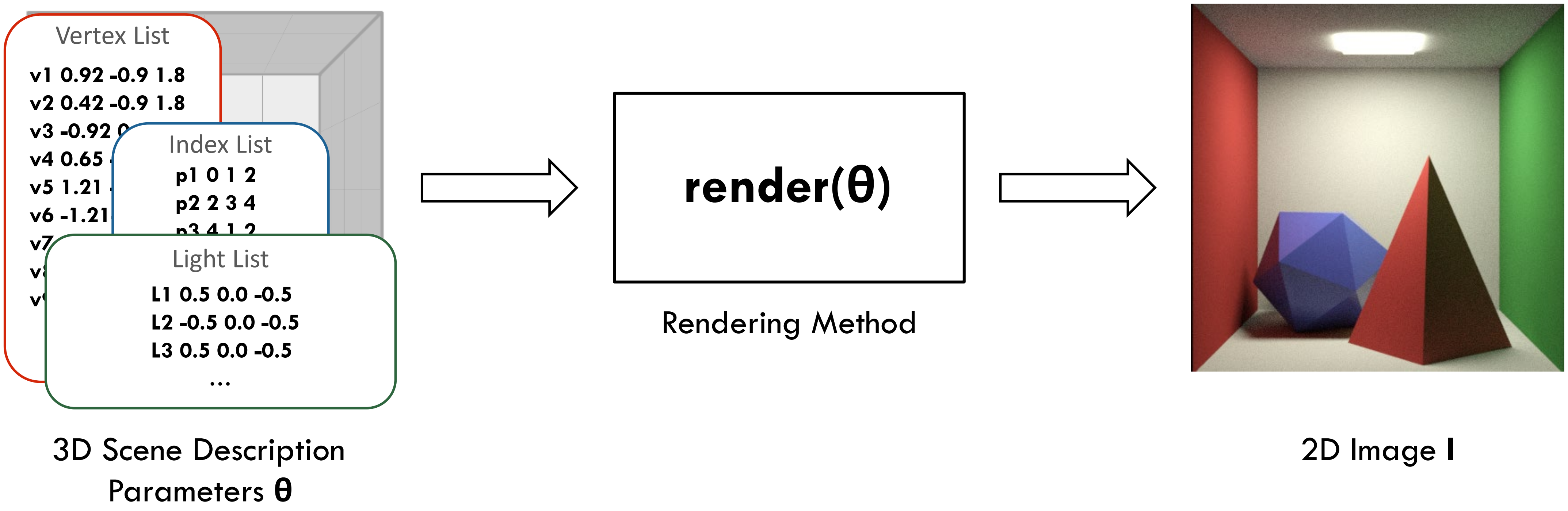
SIGGRAPH ASIA 2020 VIRTUAL
Pre-recorded sessions:
From 4 December 2020
Live sessions:
10 – 13 December 2020
SA2020.SIGGRAPH.ORG
#SIGGRAPHAsia | #SIGGRAPHAsia2020

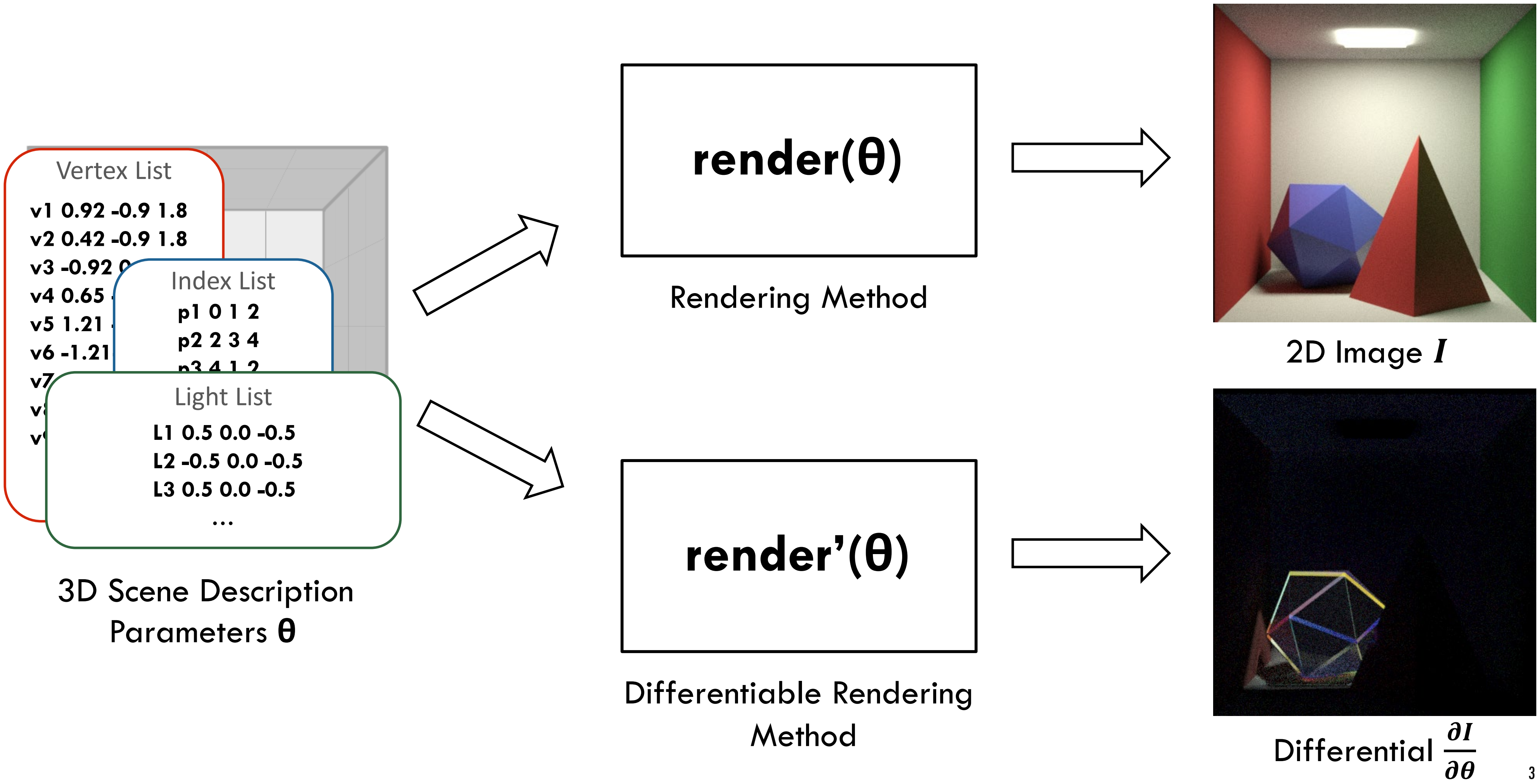
Unbiased Warped-Area Sampling for Differentiable Rendering

Sai Praveen **Bangaru**
Tzu-Mao **Li**
Frédo **Durand**

RENDERING

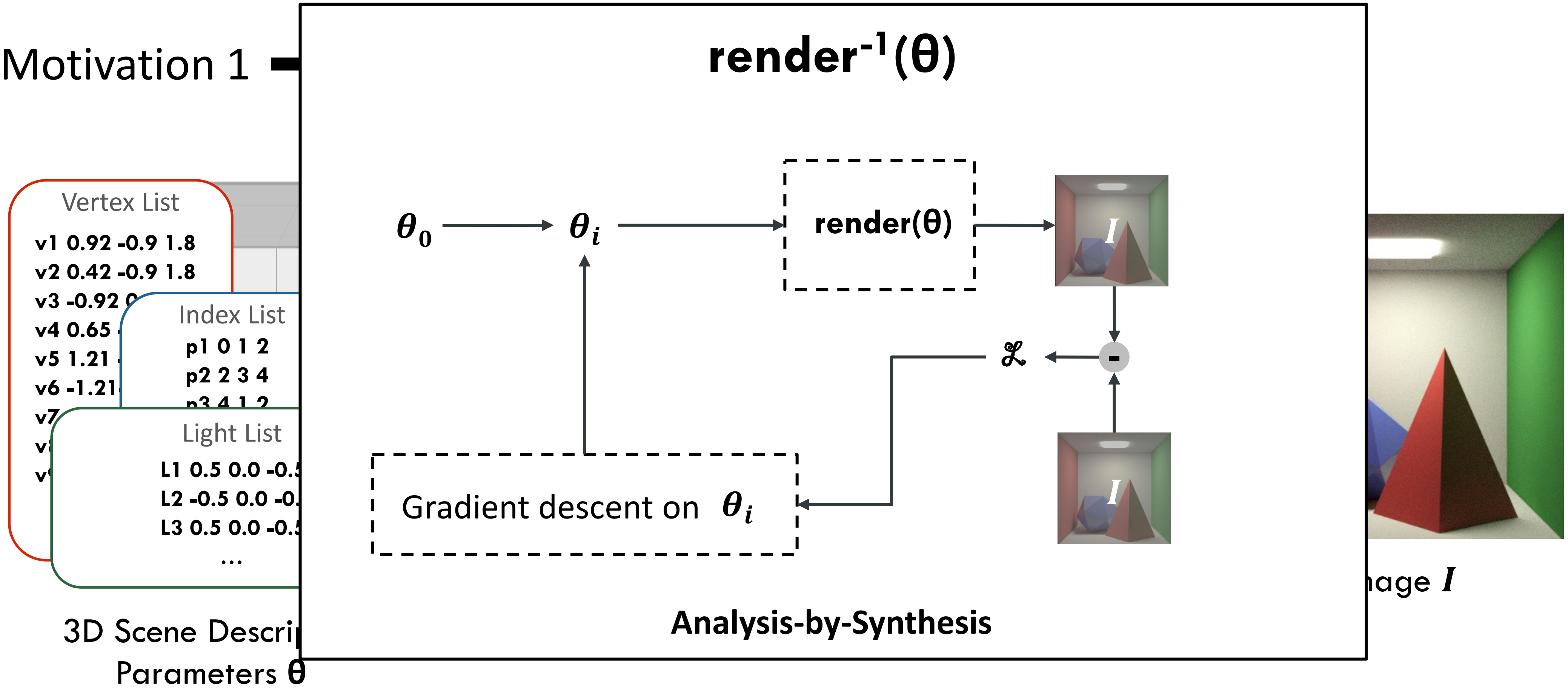

Vertex List
v1 0.92 -0.9 1.8
v2 0.42 -0.9 1.8
v3 -0.92
v4 0.65
v5 1.21
v6 -1.21
Index List
p1 0 1 2
p2 2 3 4
Light List
L1 0.5 0.0 -0.5
L2 -0.5 0.0 -0.5
L3 0.5 0.0 -0.5
...
render(θ)
Rendering Method
3D Scene Description
Parameters θ
2D Image I

DIFFERENTIABLE RENDERING



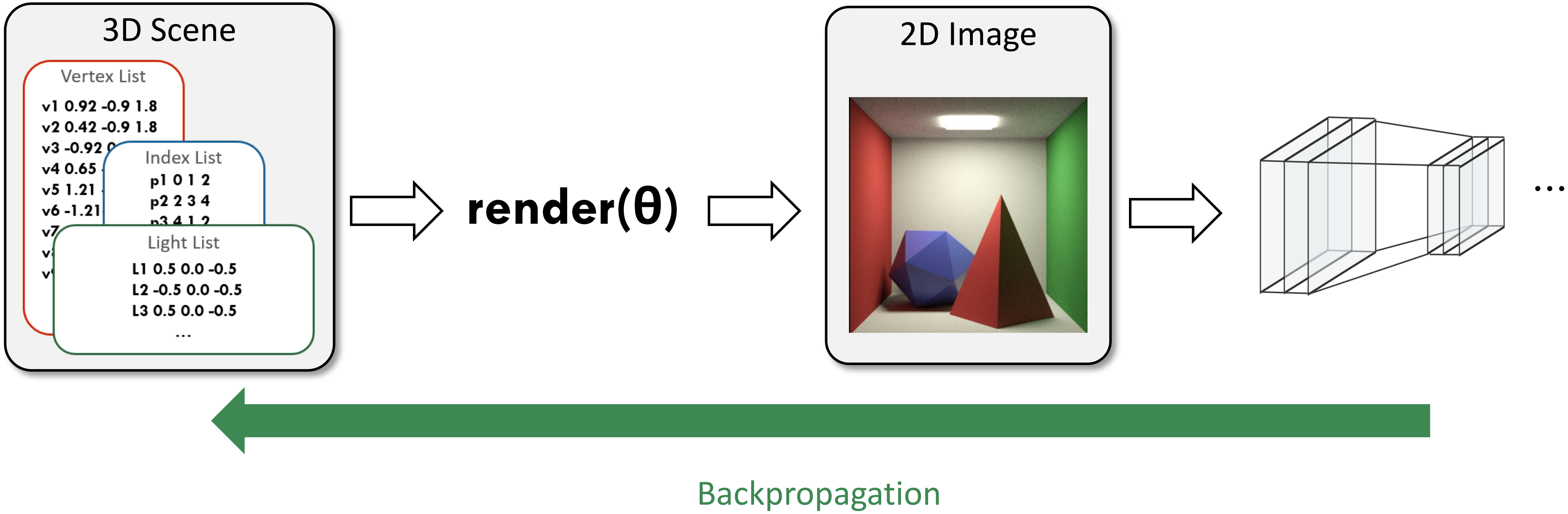
WHY DIFFERENTIABLE RENDERING?

Motivation 1



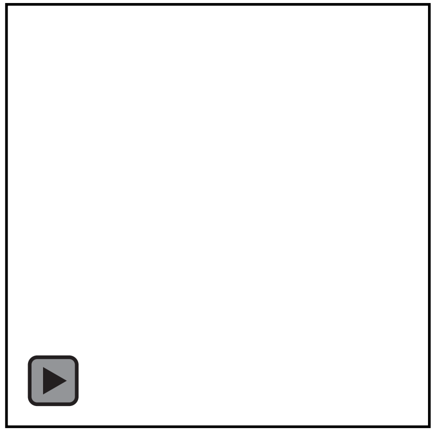
WHY DIFFERENTIABLE RENDERING?

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

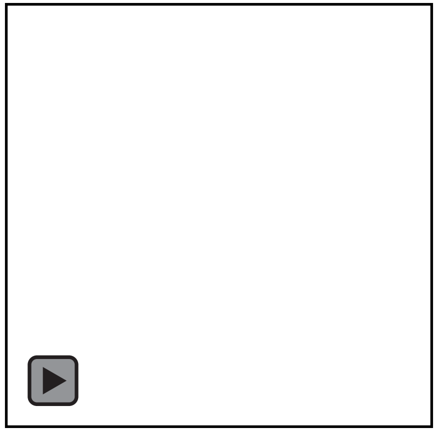


SCENE PARAMETER DERIVATIVES

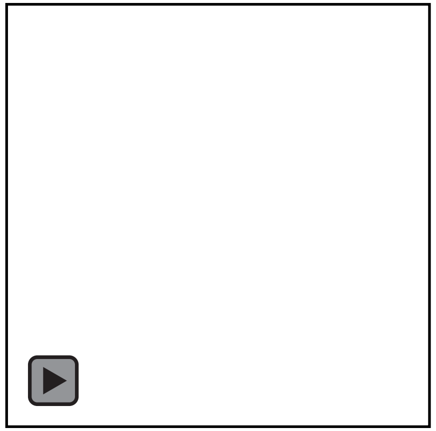
Image $\boldsymbol{I}$



θ: Object Translation

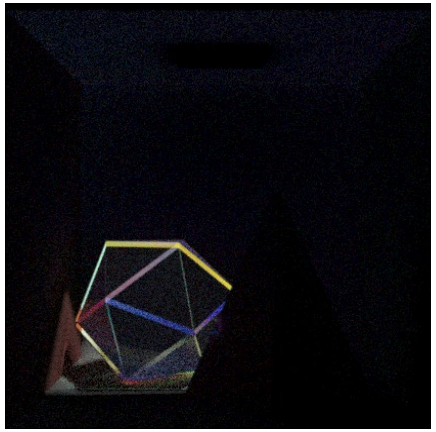


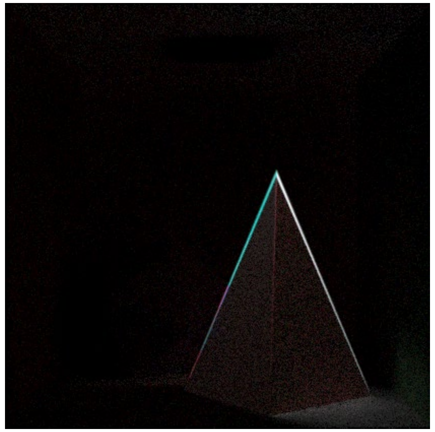
θ: Vertex position

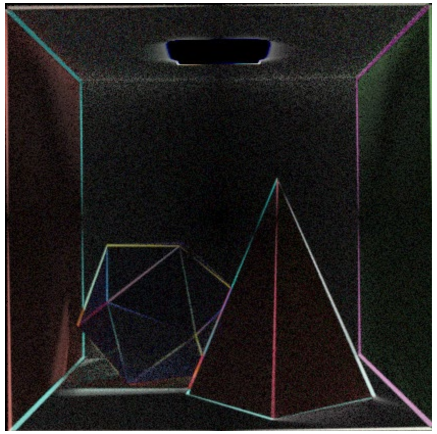


θ: Camera Rotation

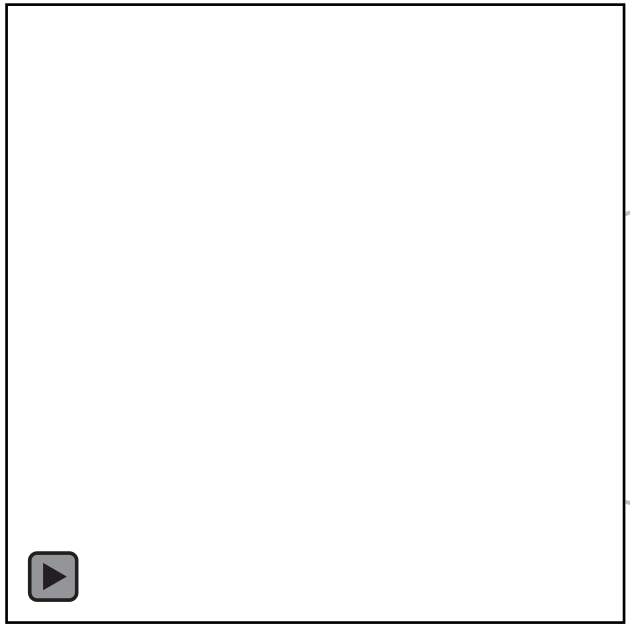
Differential $\frac{\partial \boldsymbol{I}}{\partial \boldsymbol{\theta}}$







AUTO-DIFF HAS A VISIBILITY PROBLEM



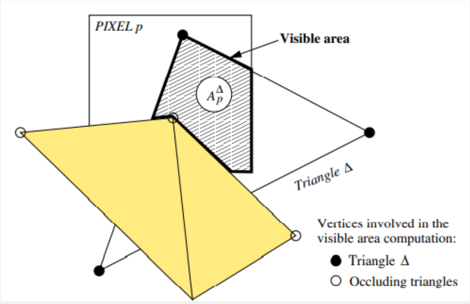
Sudden discontinuity ⇨ Auto-diff fails due to edges

Smooth function ⇨ Auto-diff computes correct derivative

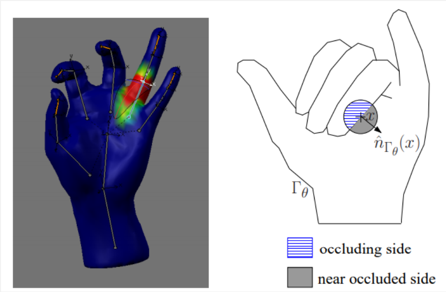
RASTERIZATION APPROACHES ARE LIMITED

Key Idea: *Analytical* occupancy

[Jalobeanu 2004]

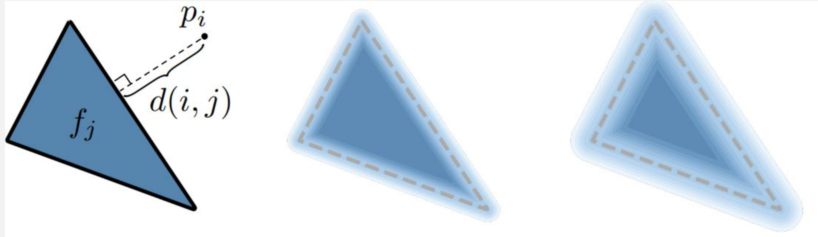


[de La Gorce 2008]

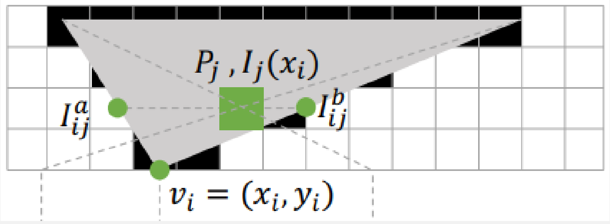


Key Idea: *Approximating* visibility

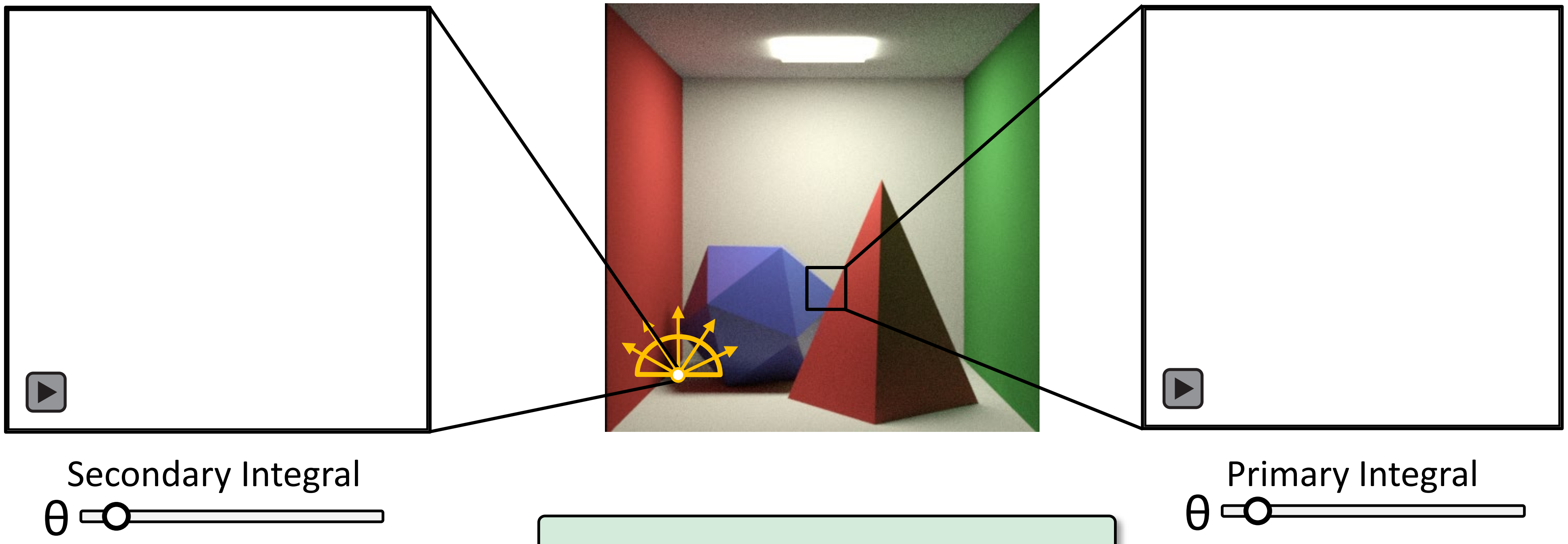
Soft Rasterizer [Liu 2019]



Neural 3D Mesh Renderer [Kato 2017]

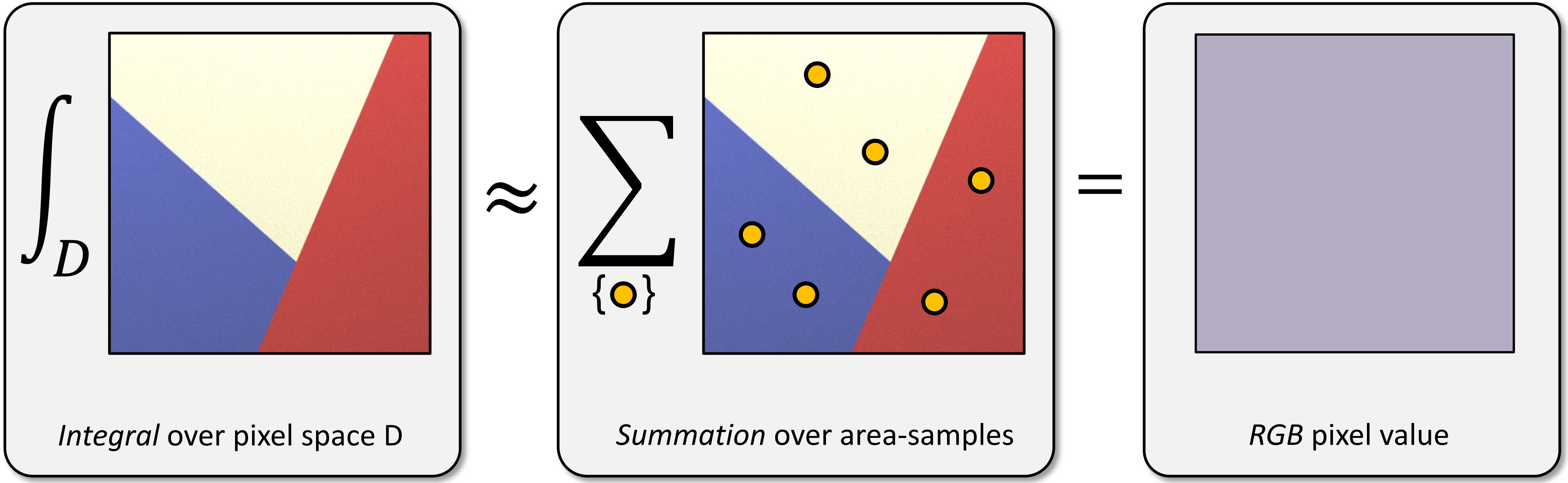


RENDERING AS AN INTEGRAL



Goal: Differentiate these integrals

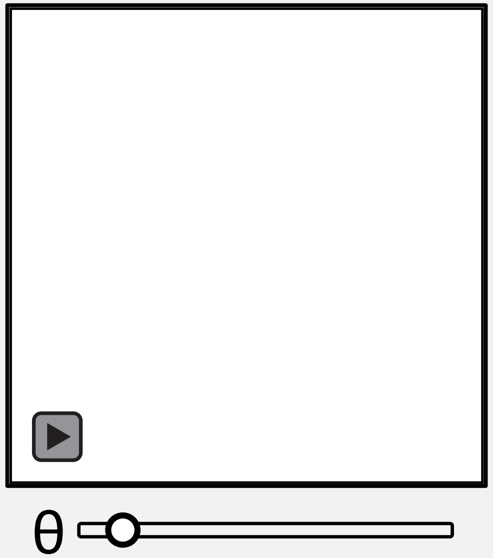
MONTE CARLO ESTIMATION



DISCONTINUOUS INTEGRANDS

(Incorrect) Attempt 1 ⟹ Apply *auto-diff* to summation

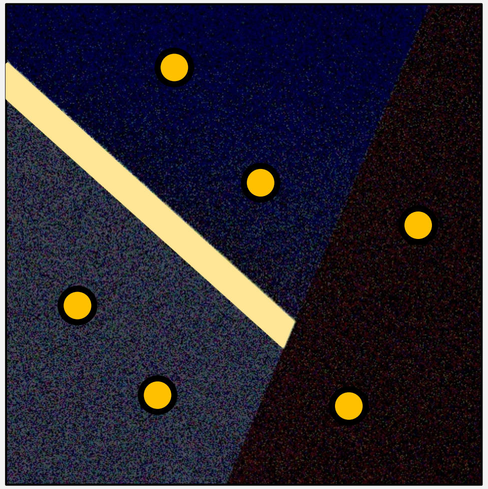
$$\partial_\theta \int_D$$



Integral over pixel space D

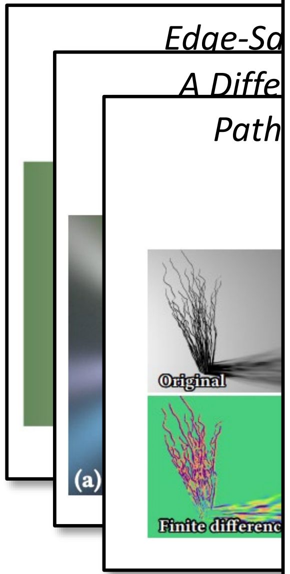
$$\not\approx$$

$$\sum_{\{\circ\}} \partial_\theta$$



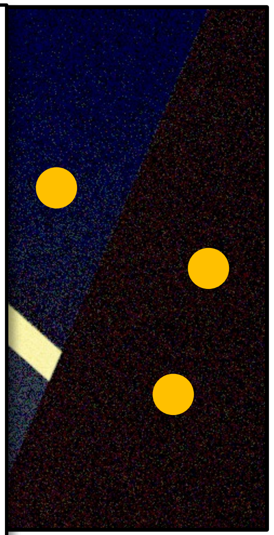
Summation over area-samples

EDGE-SAMPLING

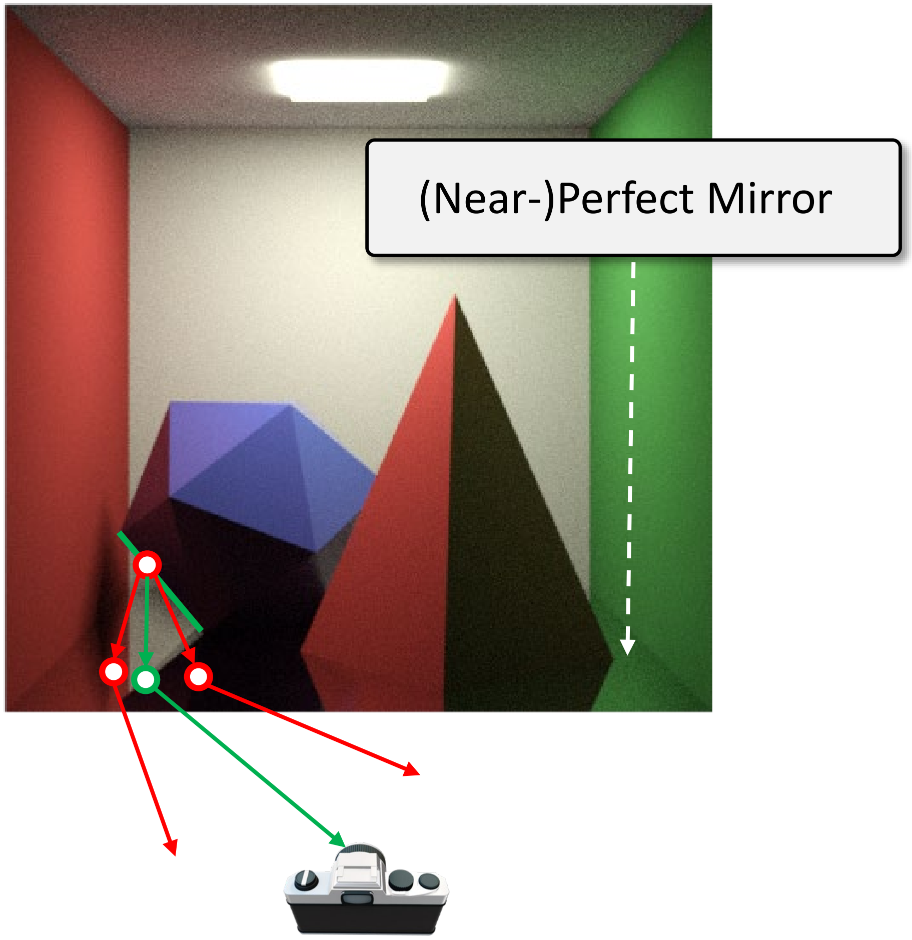


Challenges for Edge-Sampling

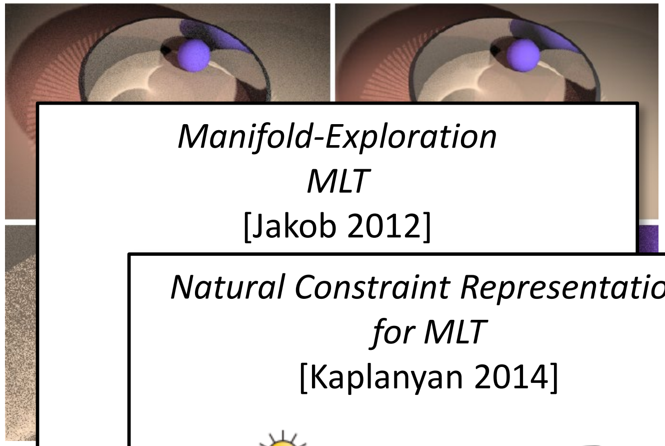
Arbitrary silhouette sampling is *hard!*



EDGE-SAMPLING HAS TROUBLE WITH SPECULAR REFLECTIONS

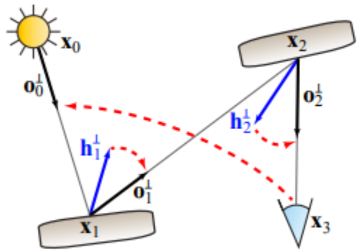


Rendering Caustics



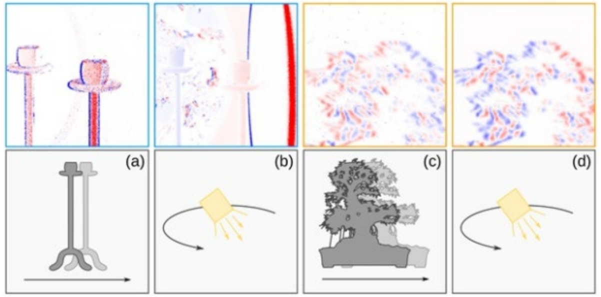
Manifold-Exploration MLT
[Jakob 2012]

Natural Constraint Representation for MLT
[Kaplanyan 2014]

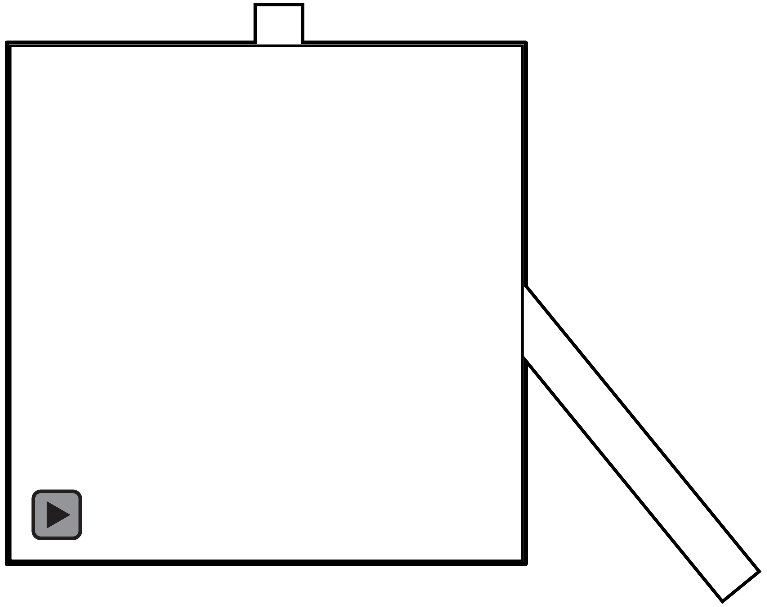


AREA-SAMPLING

Reparameterizing Discontinuous Integrands for Differentiable Rendering
[Loubet 2019]



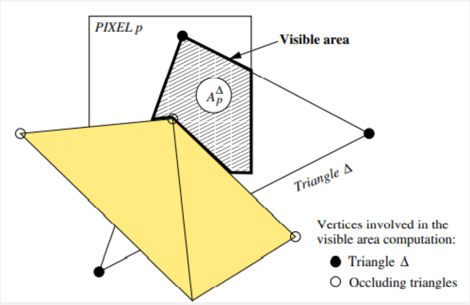
Transform samples with **θ**. Avoids discontinuities.

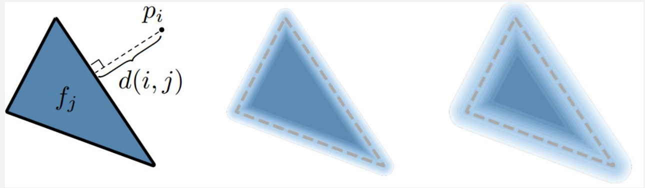


Heuristic Approximation!
May not work for all samples.

SUMMARY OF METHODS

Rasterization



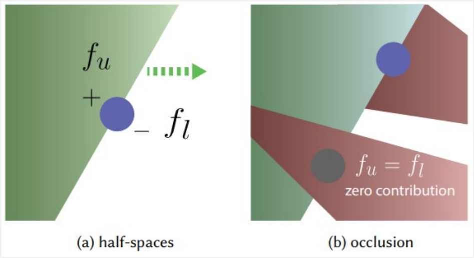


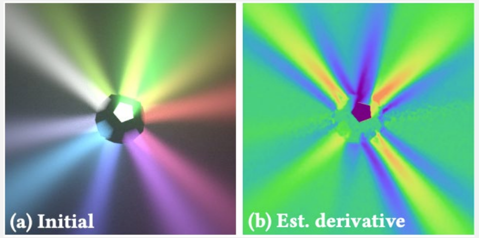
+ Fast

- Approximate visibility

- No secondary effects

Edge-sampling

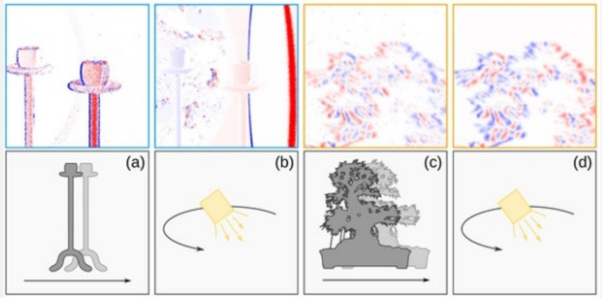




+ Exact derivative

- Depth complexity
- No perfect specularities
- Complex data structures

Area-sampling

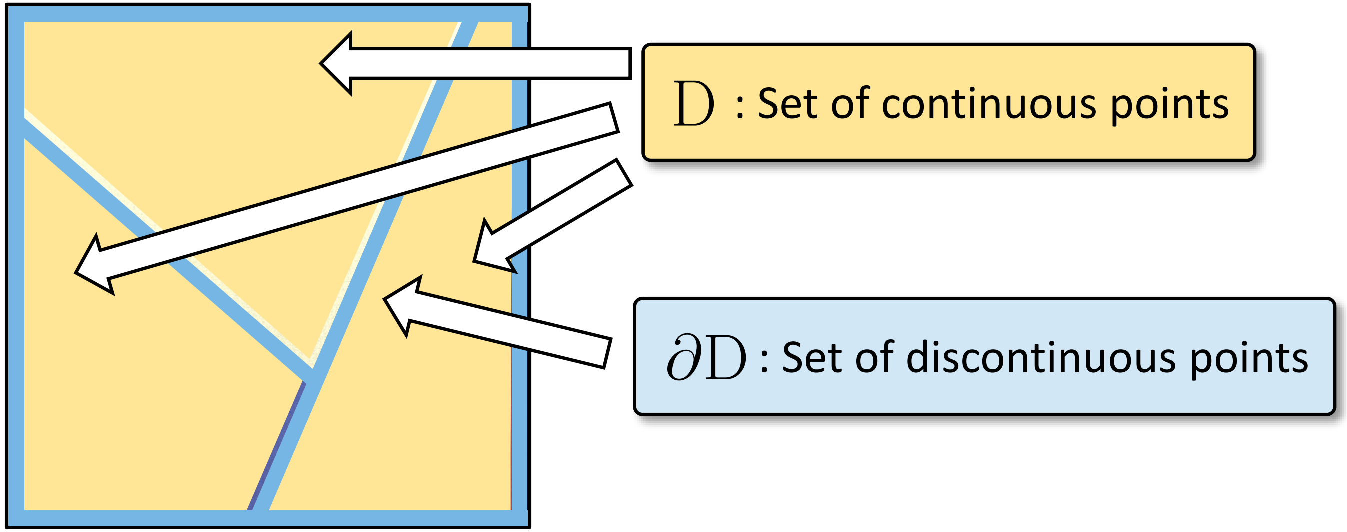


+ Fast (No complex sampling)

- Approximate derivative

OUR APPROACH

THE REYNOLDS TRANSPORT THEOREM



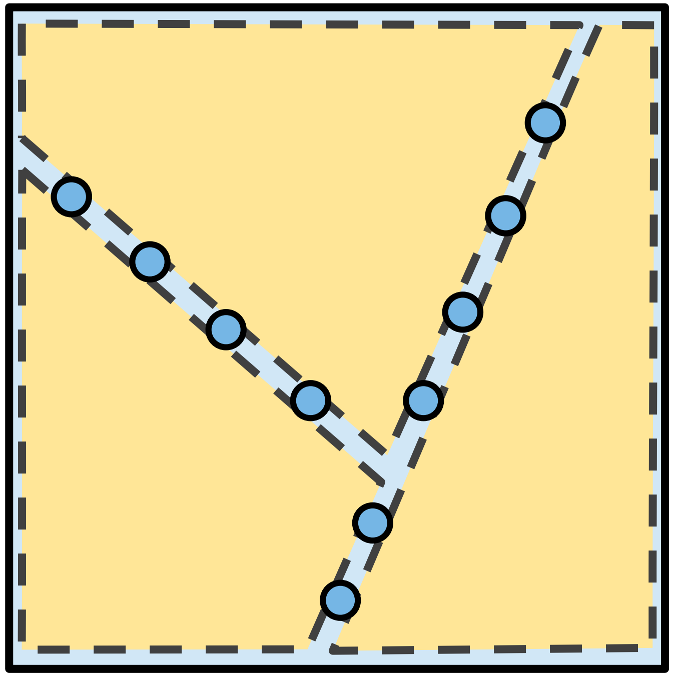
$$\partial_\theta \int_{\mathrm{D}} f = \int_{\mathrm{D}} \partial_\theta f + \int_{\partial \mathrm{D}} f \vec{\mathbf{v}} \cdot \vec{\mathbf{n}}$$

Interior term | Edge term

CONVERTING EDGE-SAMPLES TO AREA-SAMPLES

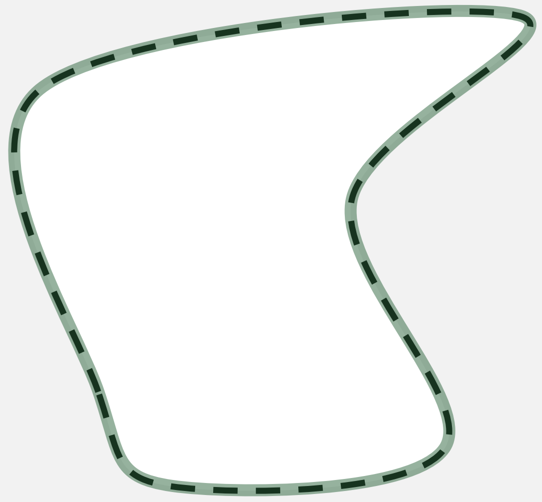
$\int_{\partial D} f\vec{\mathbf{v}} \cdot \vec{\mathbf{n}}$ is estimated through edge-samples

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



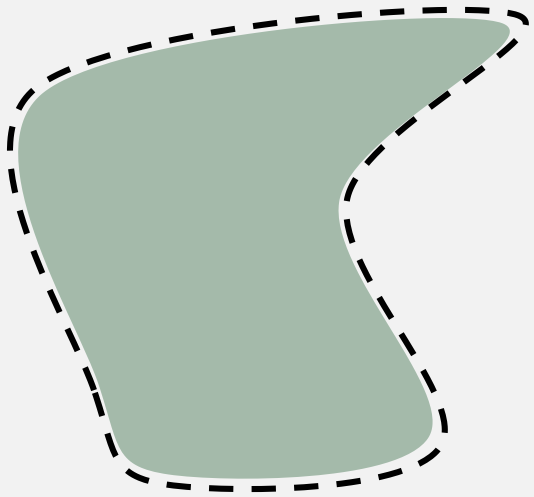
THE DIVERGENCE THEOREM

[Gauss 1813]



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

$$\Leftrightarrow$$



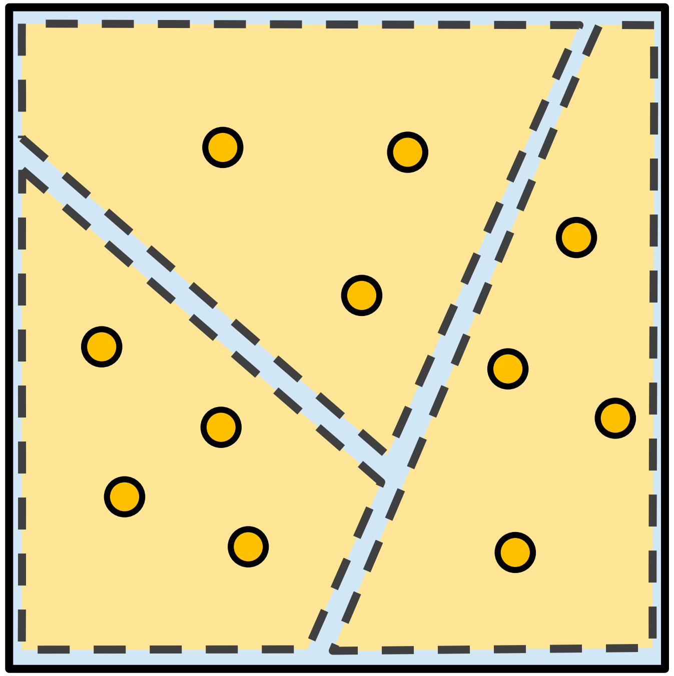
$$\int_{D} \nabla \cdot \vec{f}$$

APPLYING THE DIVERGENCE THEOREM TO THE EDGE INTEGRAL

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

Solution: Rewrite $\int_{\partial \mathrm{D}} f\vec{\mathbf{v}} \cdot \vec{\mathbf{n}}$ into $\int_{\mathrm{D}} \nabla \cdot (\vec{\mathcal{V}}_\theta f)$

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



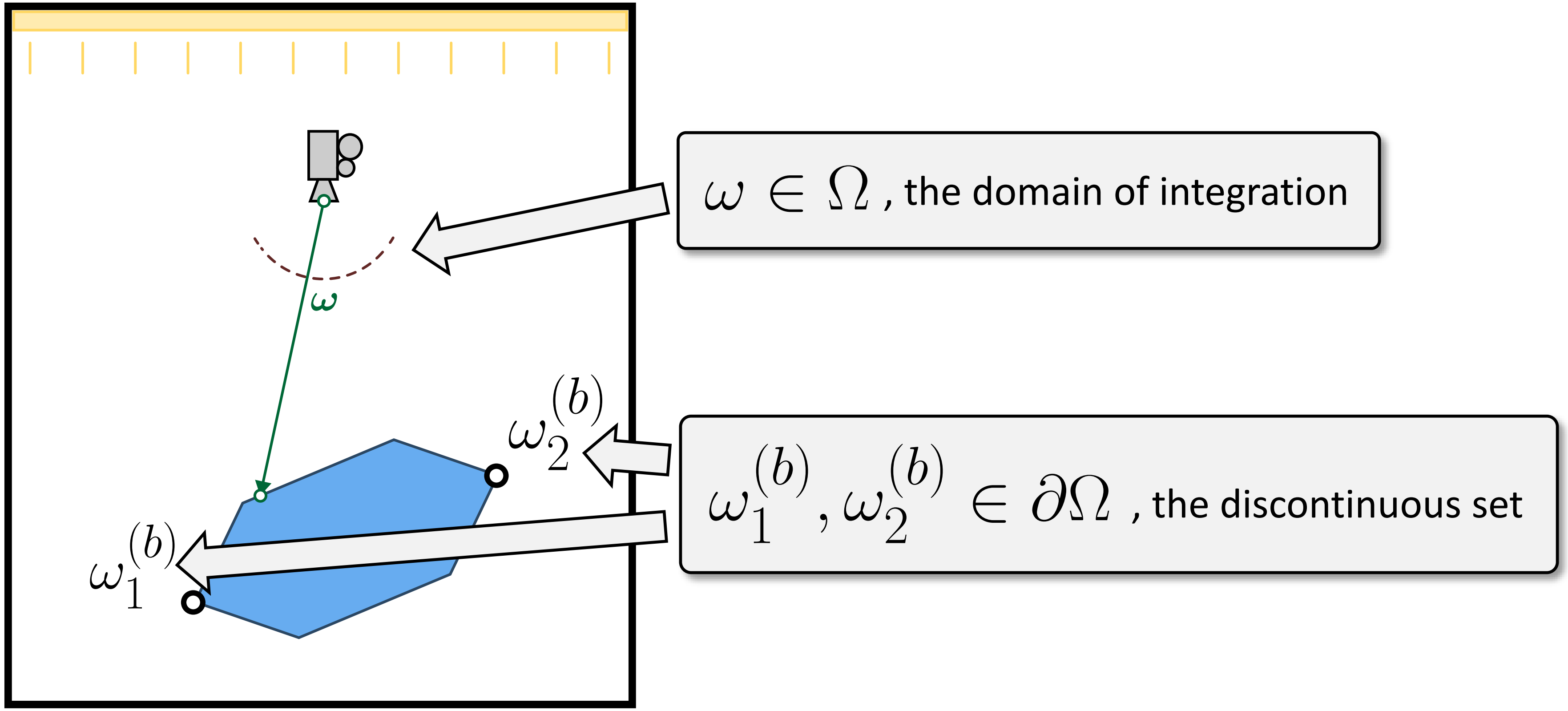
QUICK RECAP

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

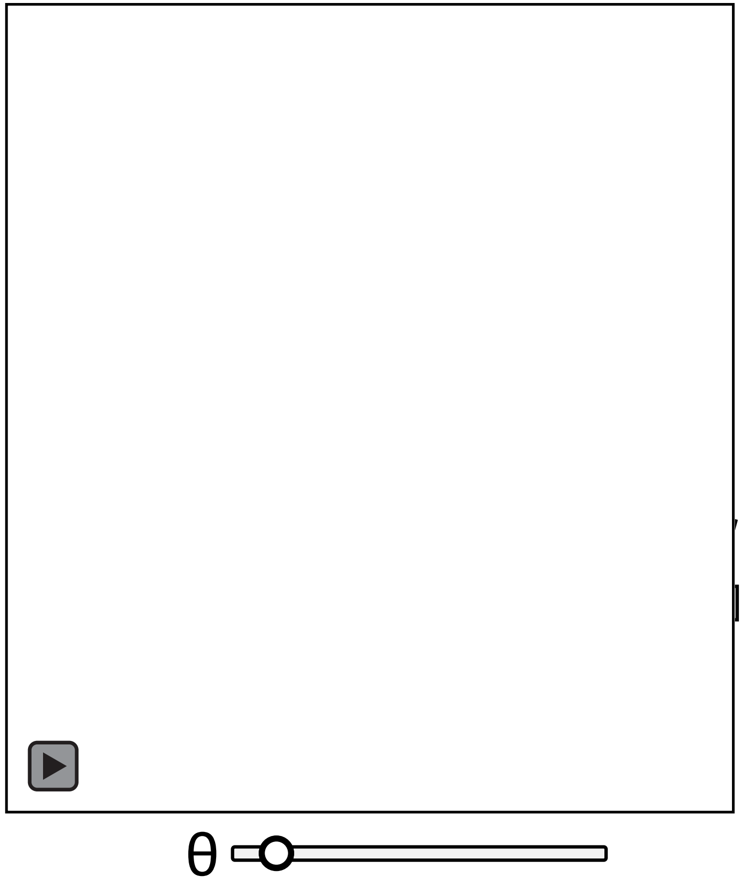
- Rewrote $\int_{\partial \mathrm{D}} f \vec{\mathbf{v}} \cdot \vec{\mathbf{n}}$ to $\int_{\mathrm{D}} \nabla \cdot (\vec{\mathcal{V}}_\theta f)$ using the *divergence theorem*.

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

A 2D EXAMPLE SCENE

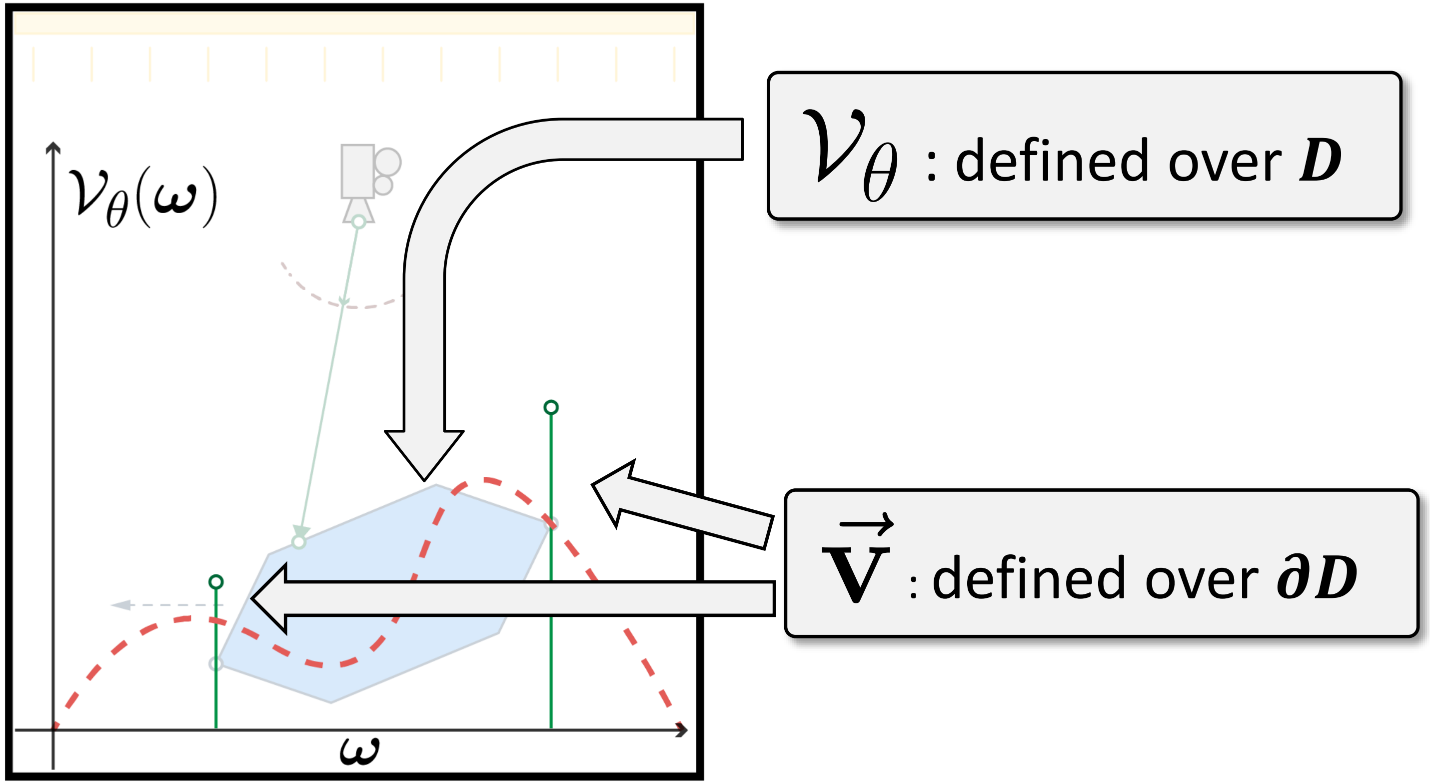


VELOCITY $\vec{\mathbf{V}}$: THE BOUNDARY DERIVATIVE



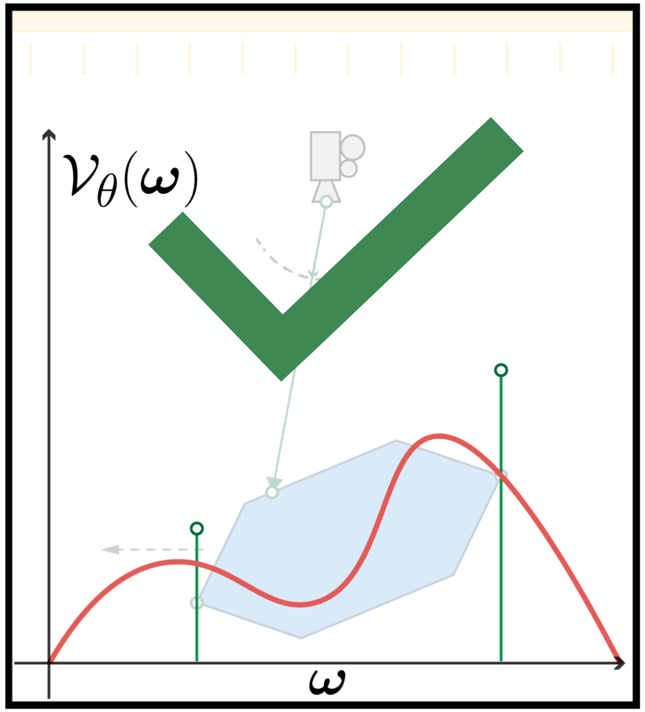
$$\partial_\theta \omega_i^{(b)}$$: Derivative of boundary position w.r.t **θ**

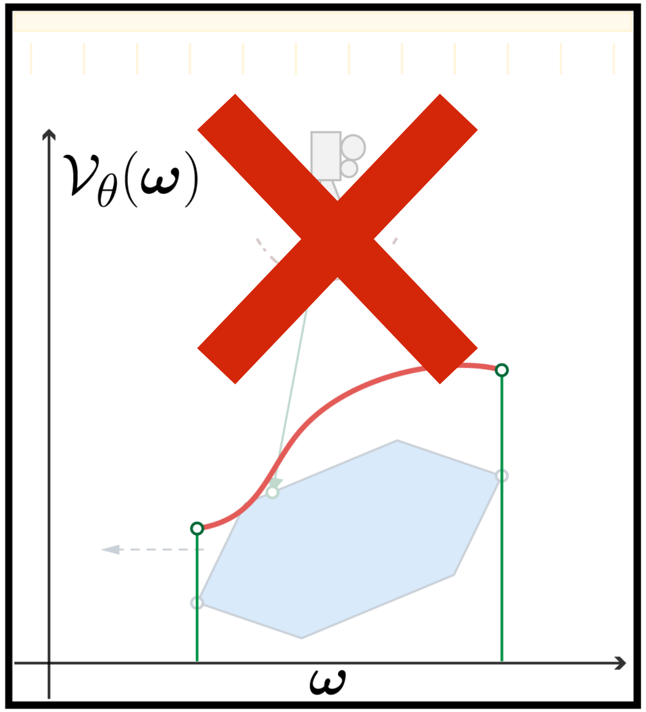
WARP FIELD $\mathcal{V}_\theta$: EXTENSION OF $\vec{\mathbf{V}}$ TO ALL POINTS



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

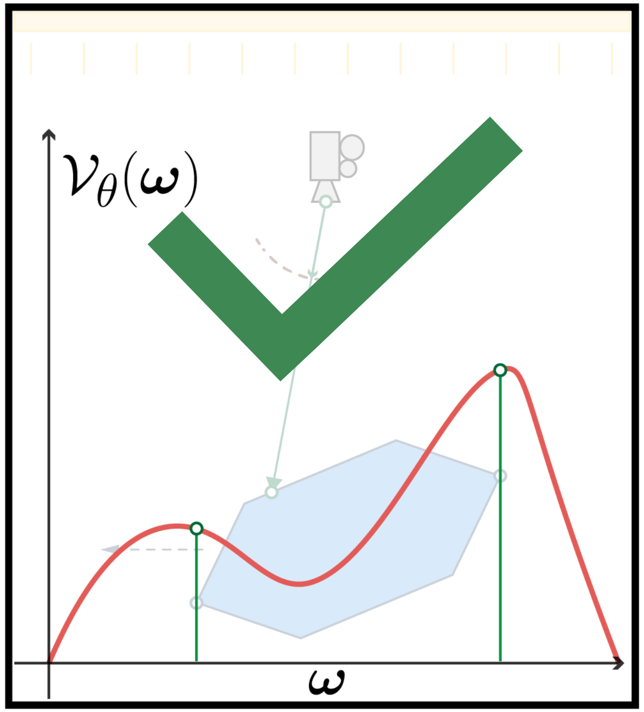
Rule 1: Continuous

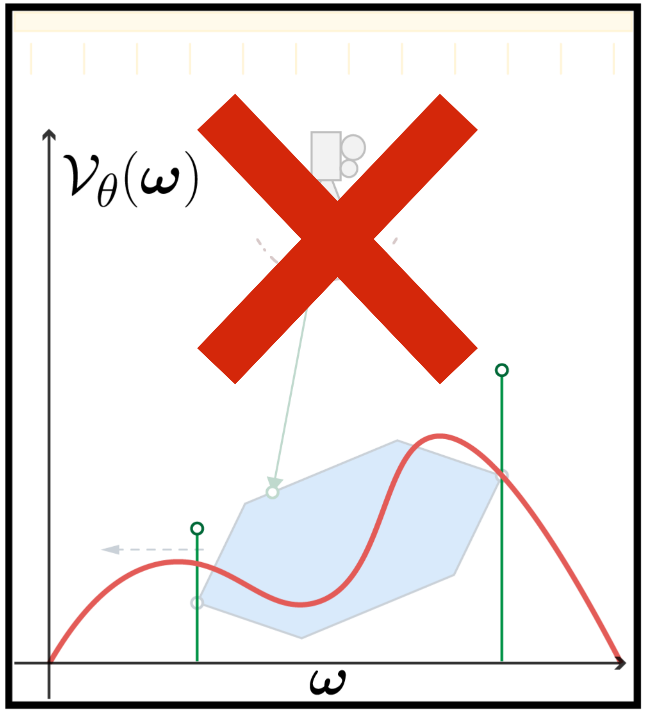




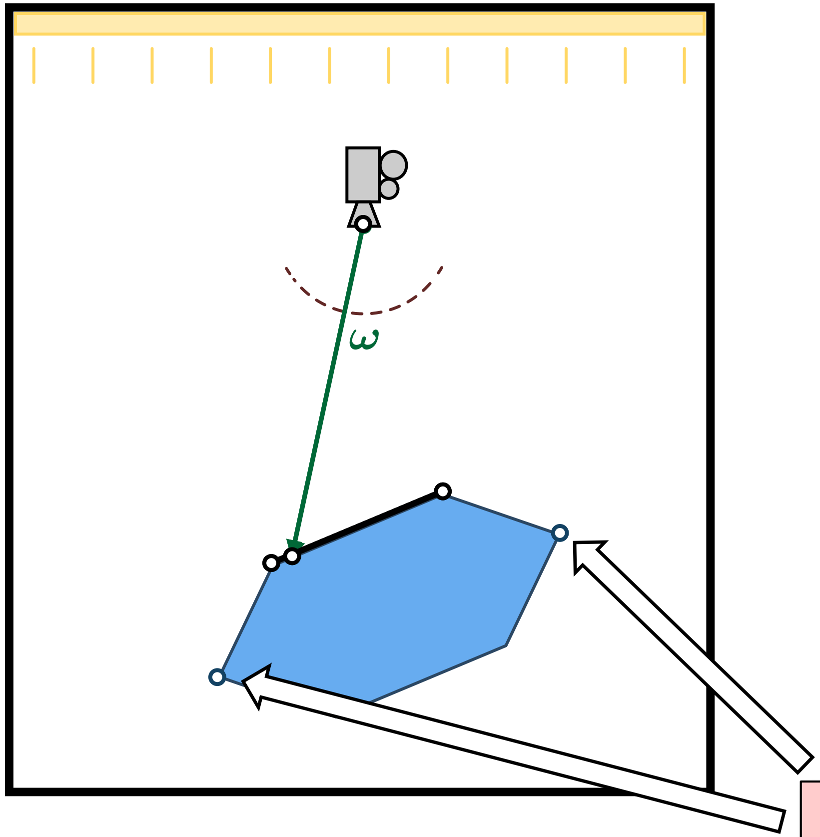
VALIDITY OF $\vec{\mathcal{V}}_\theta$

Rule 2: Boundary Consistent





INTERPOLATION WITHOUT KNOWLEDGE OF BOUNDARIES



Available quantities

- Origin point
- Ray
- Intersection
- Primitive

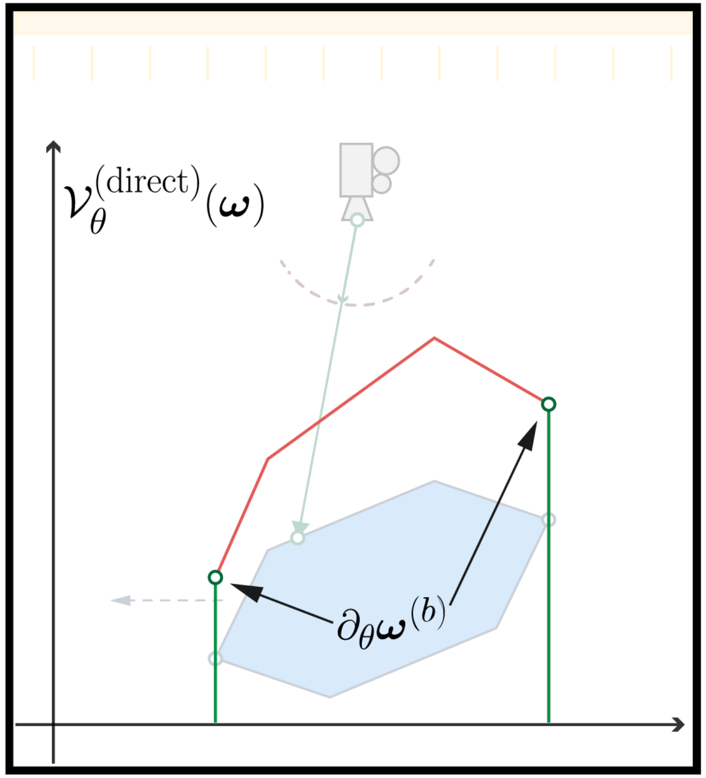
No access to discontinuity points

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

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

(Incorrect)

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



At all points (not just boundaries)

+ Boundary consistent

- Not continuous

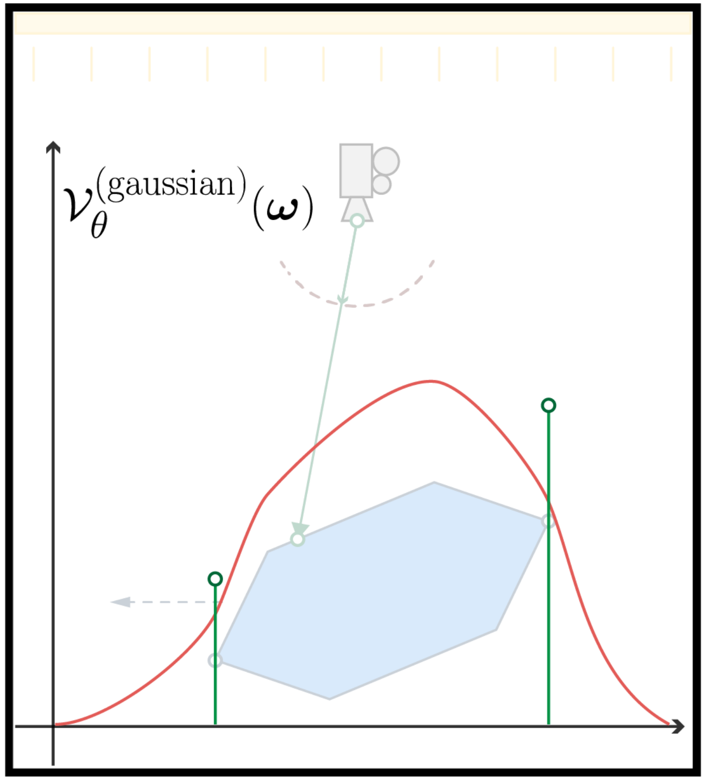
CONSTRUCTING $\vec{\mathcal{V}}_\theta$

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

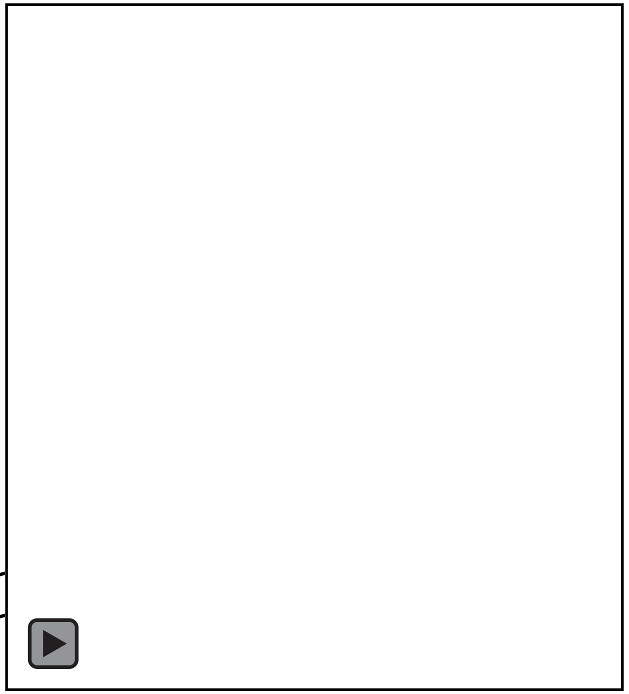
- Not boundary consistent

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

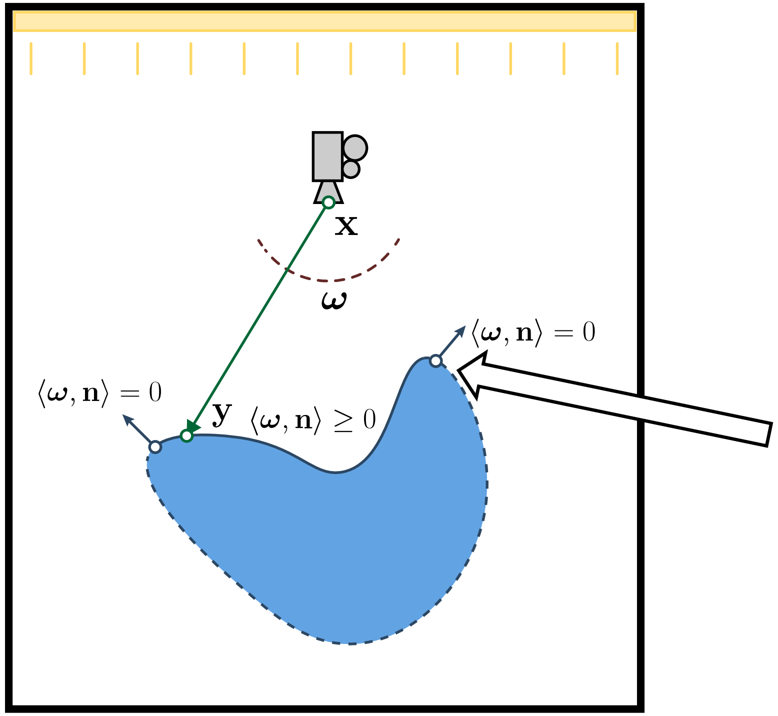
$\boldsymbol{\omega}$



Approach Dirac delta near boundaries

BOUNDARY-AWARE WEIGHTING

Implicit Boundary through *geometric normals*



$$\langle \boldsymbol{\omega}, \mathbf{n} \rangle = 0$$

at boundaries

y sampling)

boundary
g)

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

Our Approach ⟶ Filter *Attempt 1* with harmonic weights

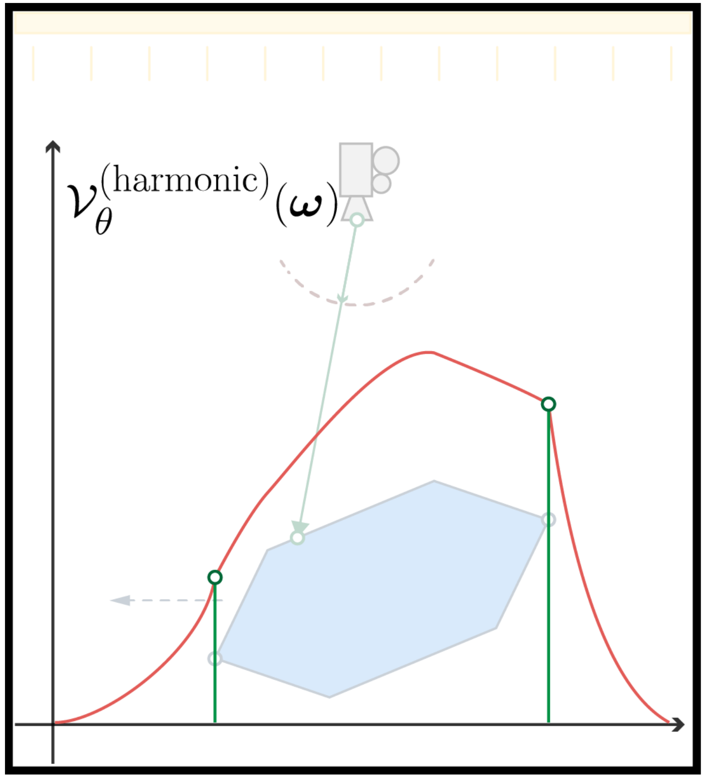
$$k(\boldsymbol{\omega}, \boldsymbol{\omega}') = \frac{1}{\boxed{\mathcal{D}(\boldsymbol{\omega}, \boldsymbol{\omega}')} + \boxed{\mathcal{B}(\boldsymbol{\omega}')}}$$

Distance function

Boundary test

+ Boundary consistent

+ Continuous



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

1. Sample **path** using path tracer *(N paths)*

For each bounce:

2. Sample **auxiliary** rays *(N' rays)*
3. Compute boundary term ***B()*** locally
4. Compute weight **k(.,.)** and $\partial_\theta \omega$
5. Find weighted mean

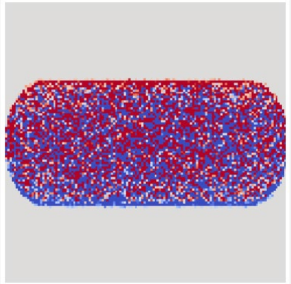
ADDITIONAL DETAILS

Russian Roulette

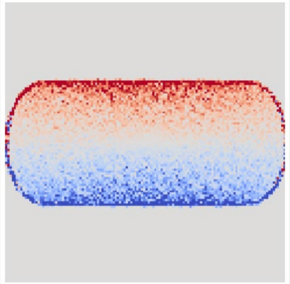
$$\frac{\sum k(\boldsymbol{\omega}, \boldsymbol{\omega}_i')g(\boldsymbol{\omega}_i')}{\sum k(\boldsymbol{\omega}, \boldsymbol{\omega}_i')}$$

$$N' \sim \textsc{Geom}(p)$$

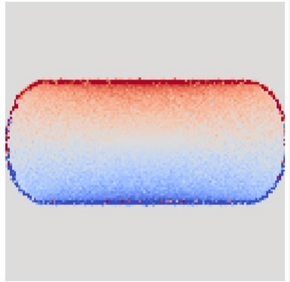
Variance Reduction



No Variance Reduction



Antithetic Variates



Antithetic Variates +
Control Variates

Relationship with Reparameterization

$$\mathcal{V}_\theta(\boldsymbol{\omega}) \Longleftrightarrow \mathcal{T}(\boldsymbol{\omega}; \theta)$$

$$\mathcal{V}_\theta(\boldsymbol{\omega}) = [\partial_\theta \mathcal{T}(\boldsymbol{\omega}; \theta)]_{\theta=\theta_0}$$

RESULTS

VARIANCE COMPARISON WITH EDGE-SAMPLING

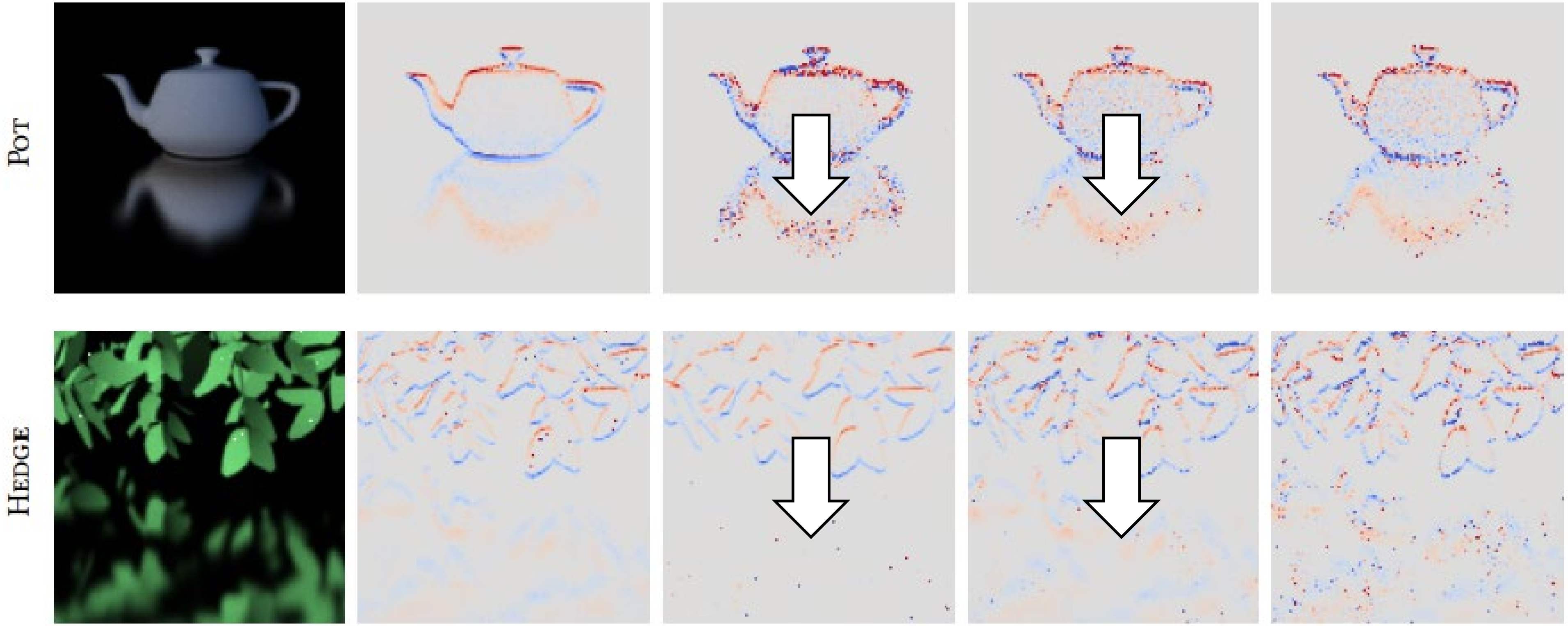


Image I

Reference Derivative

Li et al. 2018

Ours *without* Russian roulette

Ours *with* Russian roulette

BIAS COMPARISON WITH REPARAMETERIZATION

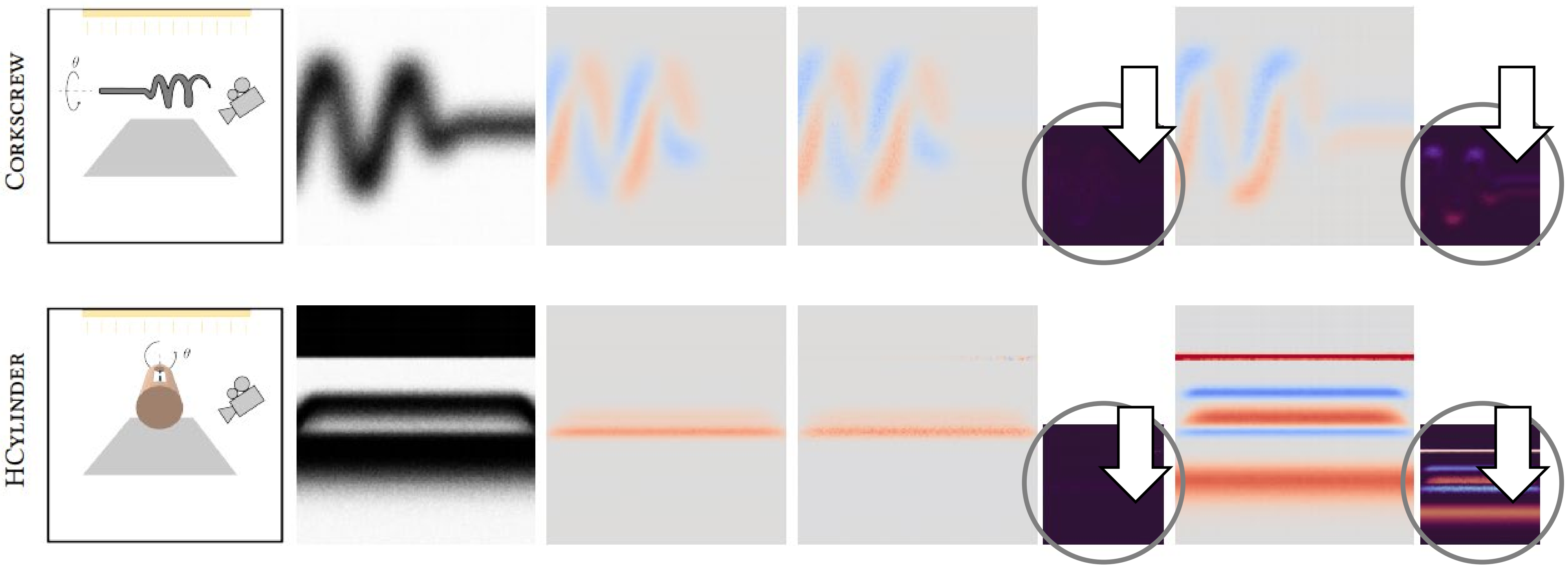


Illustration Image I Reference Ours Loubet et al. 2019

Rotating cylindrical objects present a complicated scenario for area-sampling

BIAS COMPARISON WITH REPARAMETERIZATION

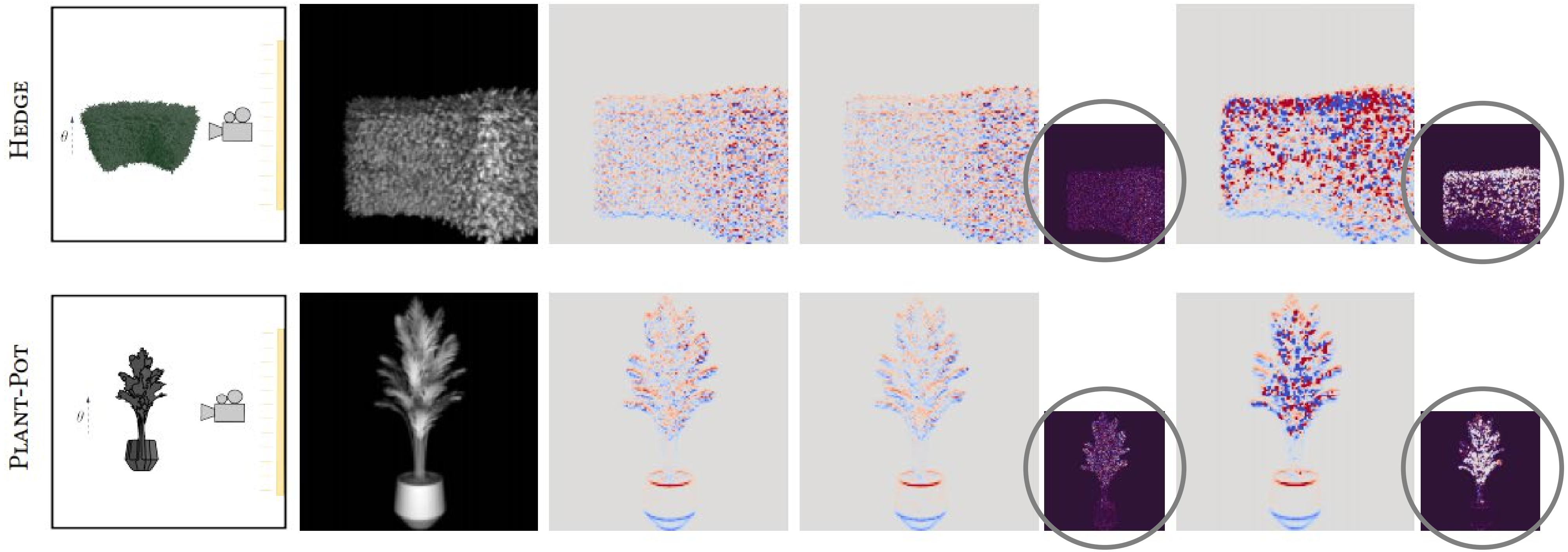
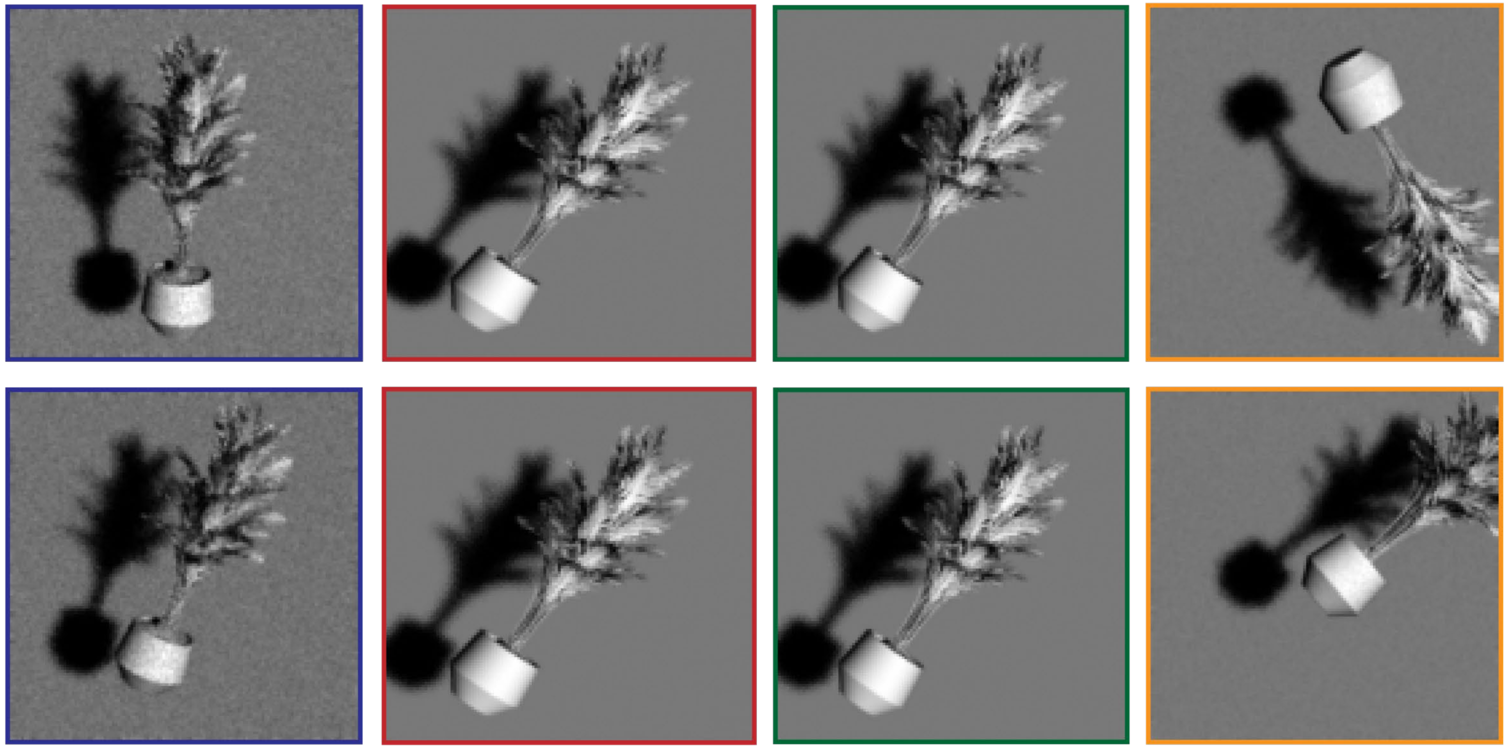


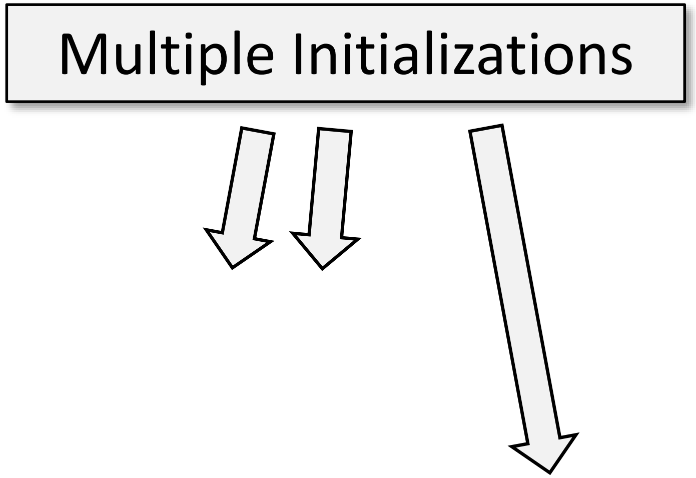
Illustration | Image I | Reference | Ours | Loubet et al. 2019

Extremely complex geometry like foliage can cause heuristic to fail

POSE ESTIMATION CAN FAIL WITH BIASED GRADIENTS

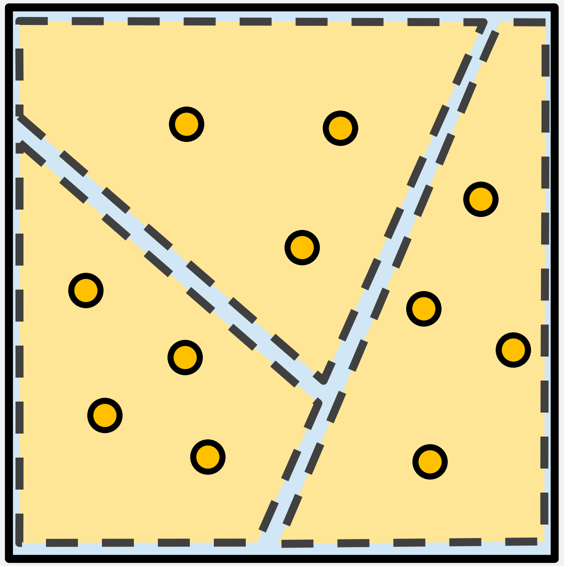


Initialization Target Ours Reparameterization

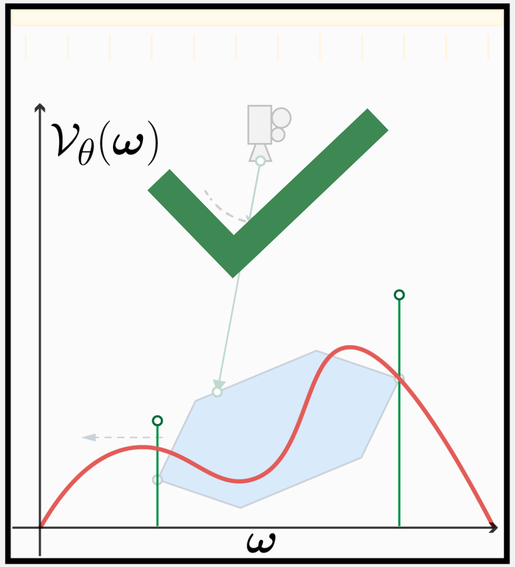


CONTRIBUTIONS

Edge-integral to *Area*-integral

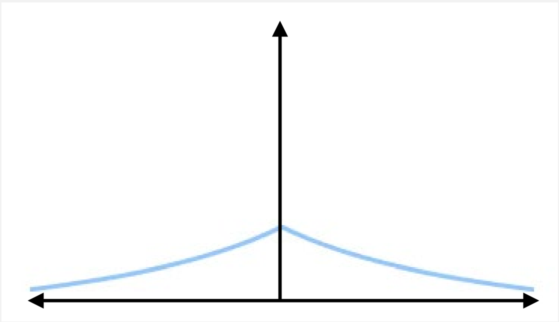


Warp field conditions



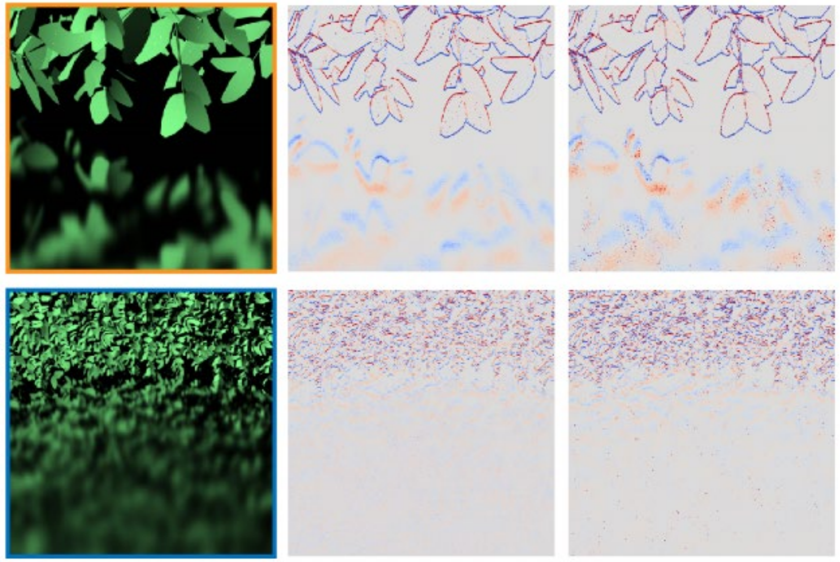
Harmonic interpolation

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



ACKNOWLEDGEMENTS

This project was funded by the **Toyota Research Institute.**



Highlighted Section   Ground Truth (FD)   Our Method

TOYOTA
RESEARCH INSTITUTE

CSAIL

Project Page

SCAN ME

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