Differentiable Rendering for Mesh and Implicit Surface

Weikai Chen
Tencent America

NeurIPS 2019
33rd Conference on Neural Information Processing Systems

GAMES
Graphics And Mixed Environment Seminar

ICCV 2019
Seoul, Korea

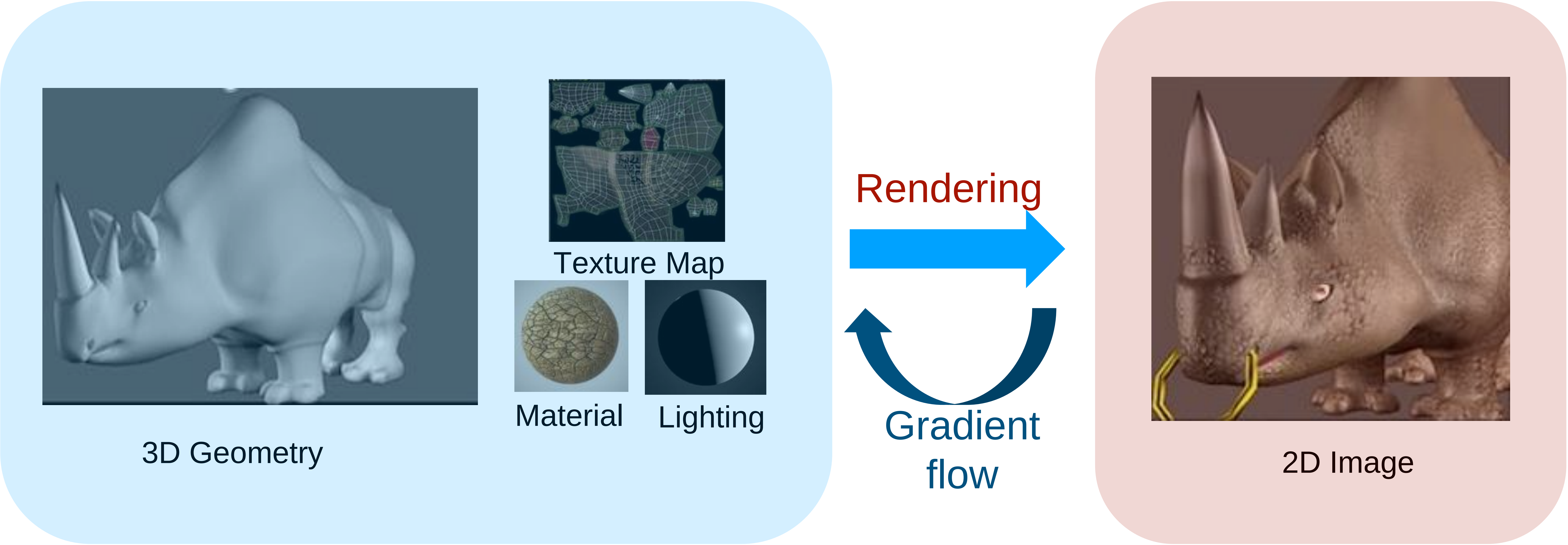
Outline

- Motivation
- SoftRas: A Differentiable Renderer for Triangular Mesh (ICCV’19)
- Learning to Infer Implicit Surfaces without 3D Supervision (NeurIPS’19)
- Conclusions

Motivation

Why Differentiable Rendering?

Rendering can be viewed as the "***bridge***" connecting 3D graphics and 2D vision



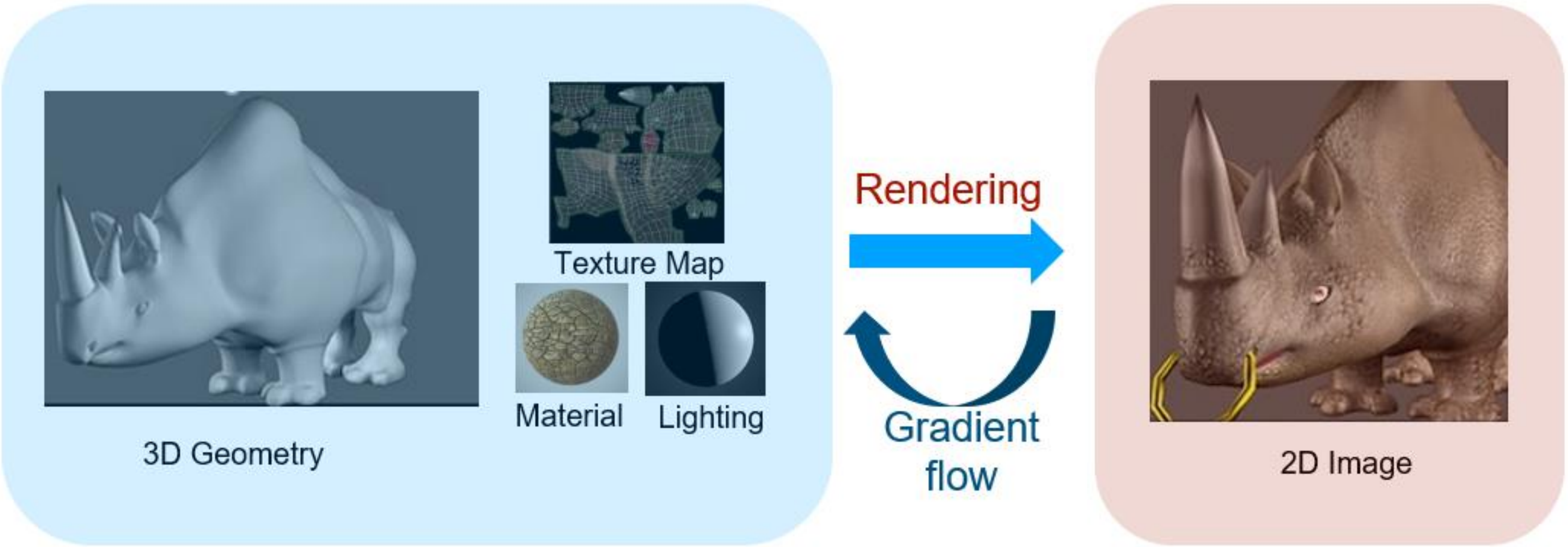
3D Graphics

2D Vision

Differentiable rendering enables direct optimization of 3D properties based on **image-based supervision** -- gradients flowing from image pixels to 3D!

Motivation

Why Differentiable Rendering?



3D Graphics **2D Vision**

Applications

Pose estimation
3D reconstruction
Material Inference
Lighting Estimation
….

ALL Image-based 3D Reasoning Tasks!

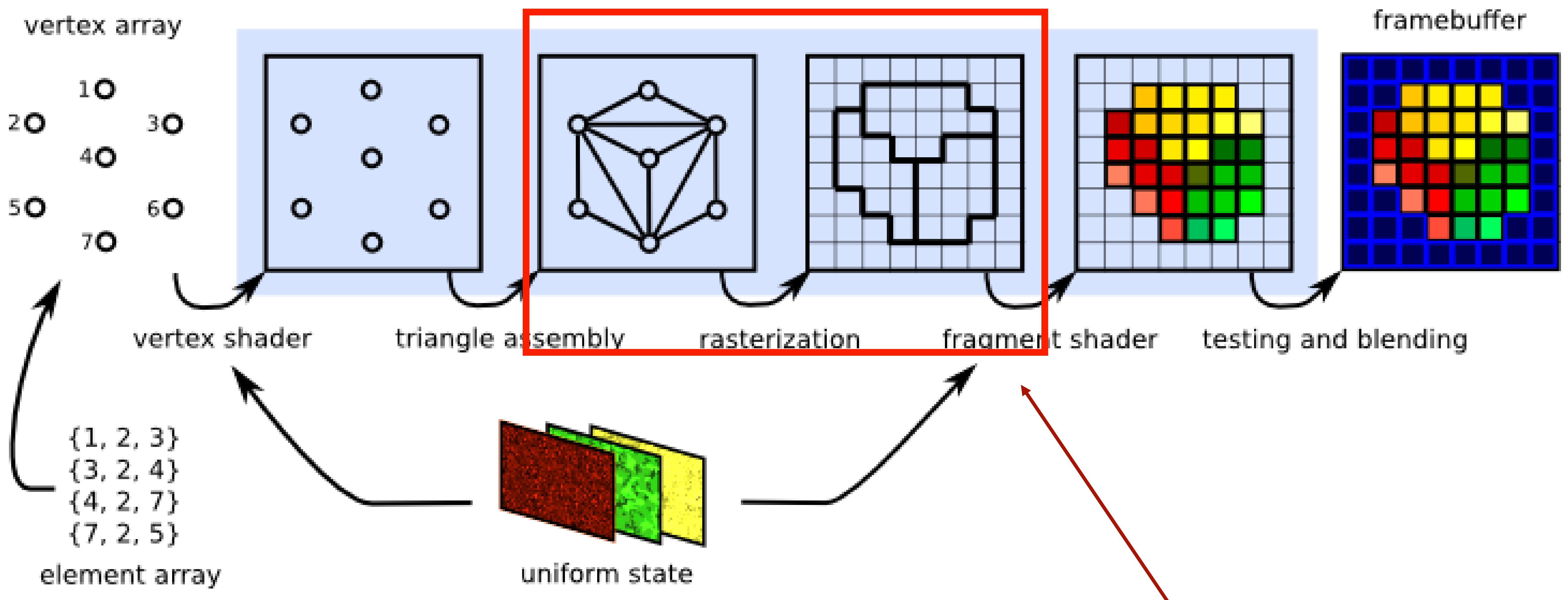
3D Unsupervised Learning!

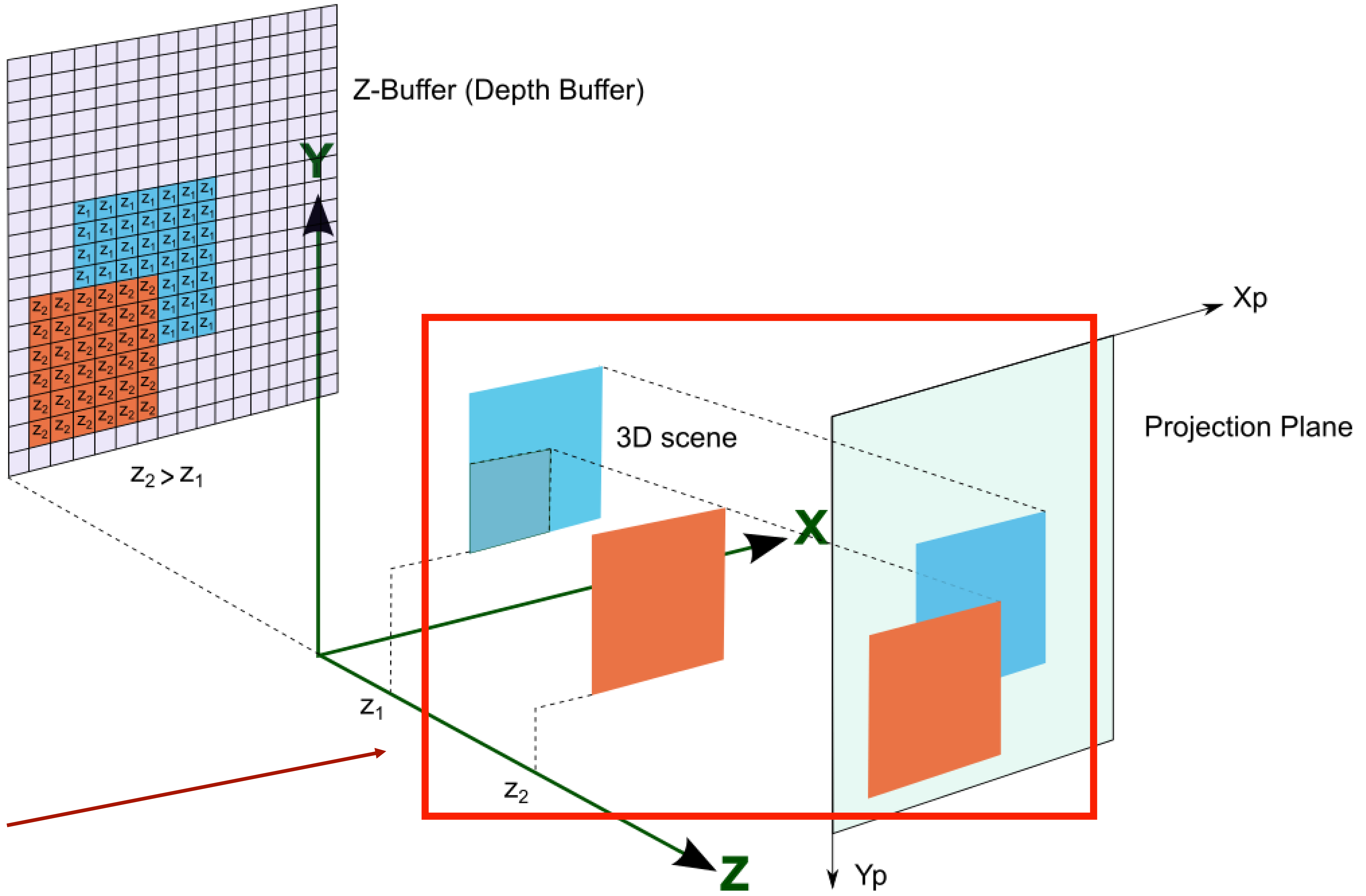
Differentiable Rendering for Meshes

Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning, ICCV’19 (Oral)

Challenges

Standard Graphics Rendering is NOT Differentiable



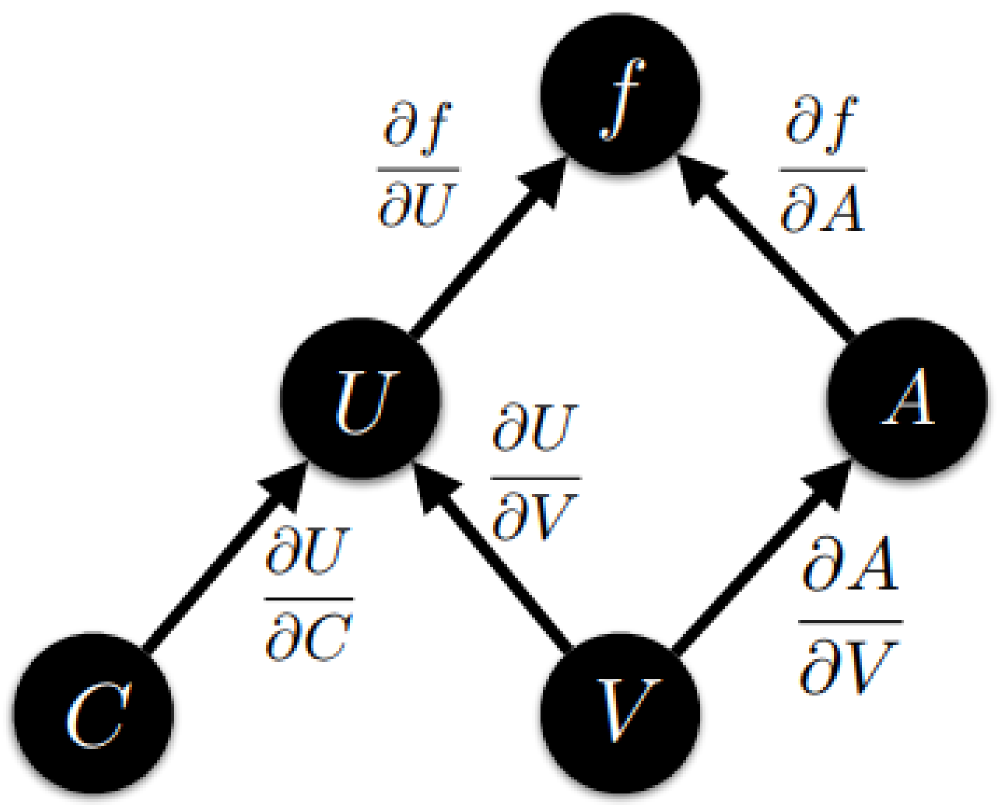


Discrete sampling

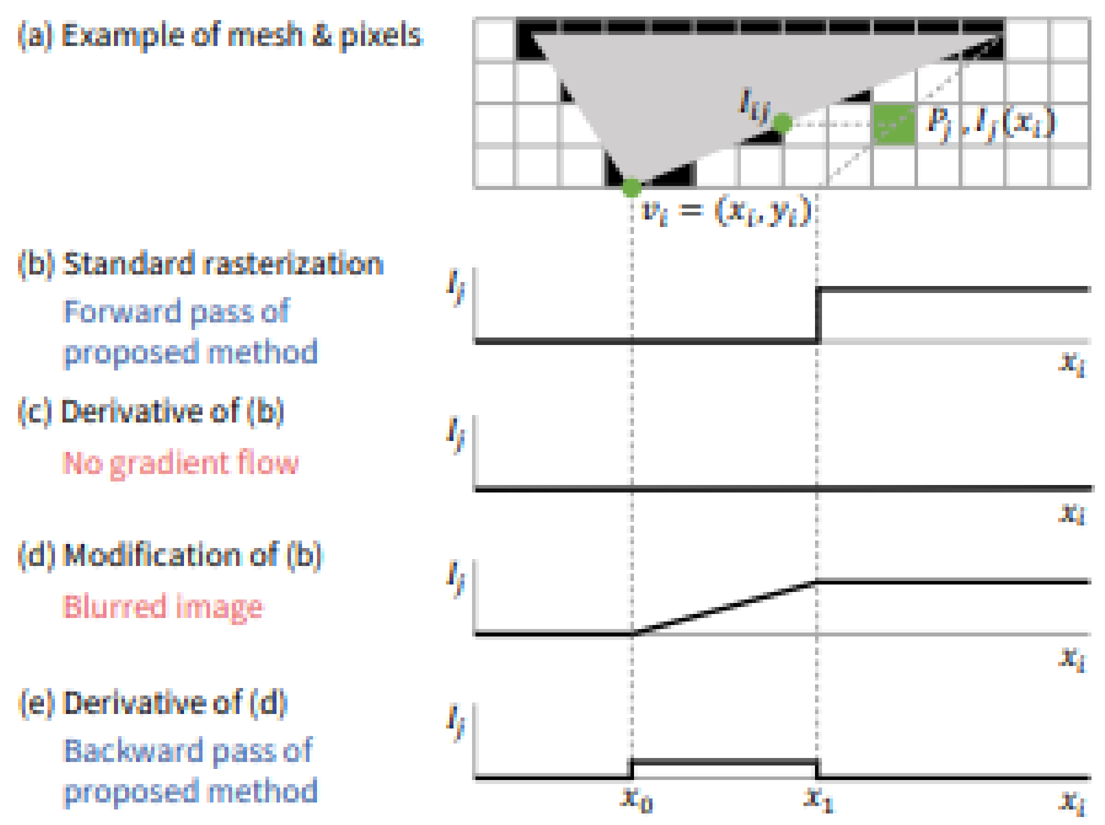
Rasterization
(XY plane)

Z-Buffering
(Z/depth direction)

Previous Works



OpenDR [Loper et al. 14]

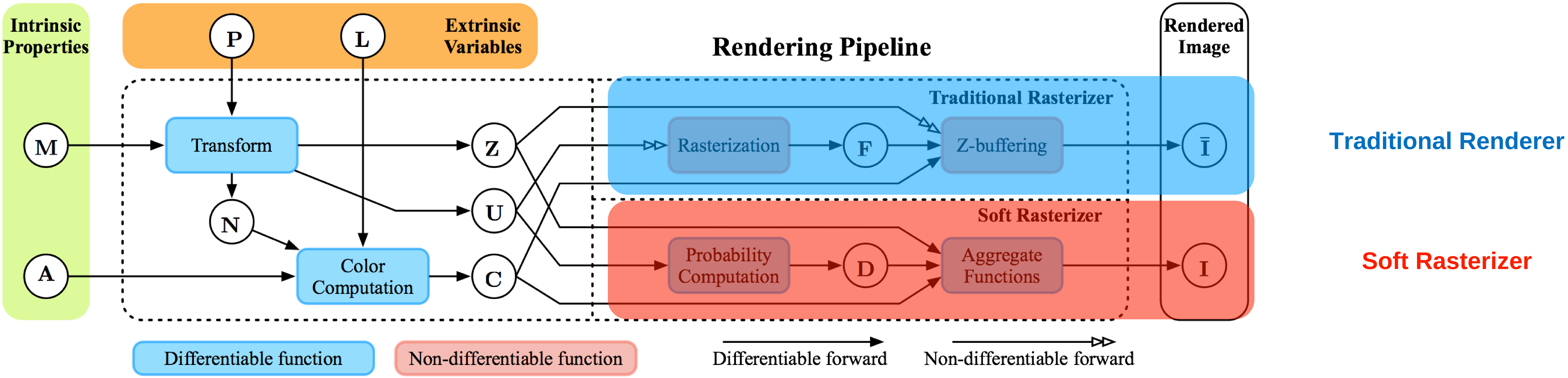


Neural 3D Mesh Renderer [Kato et al. 18]

Both directly use OpenGL in the forward rendering
and approximate the rendering gradient using hand-crafted functions.

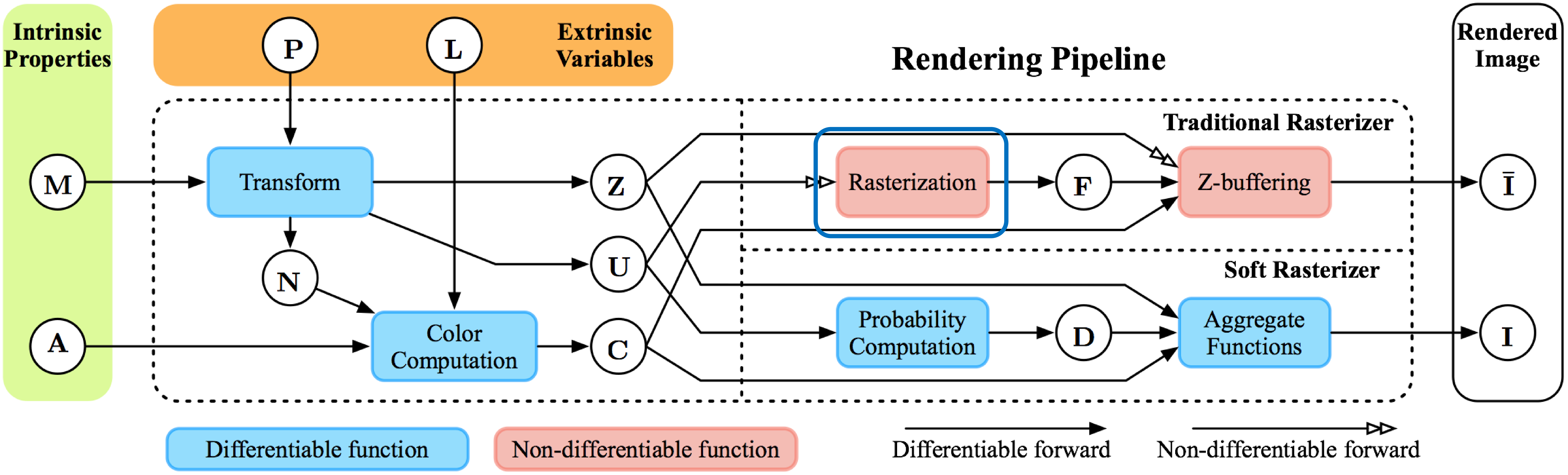
Problem: the gradient is not consistent with the forward rendering

Proposed Rendering Pipeline

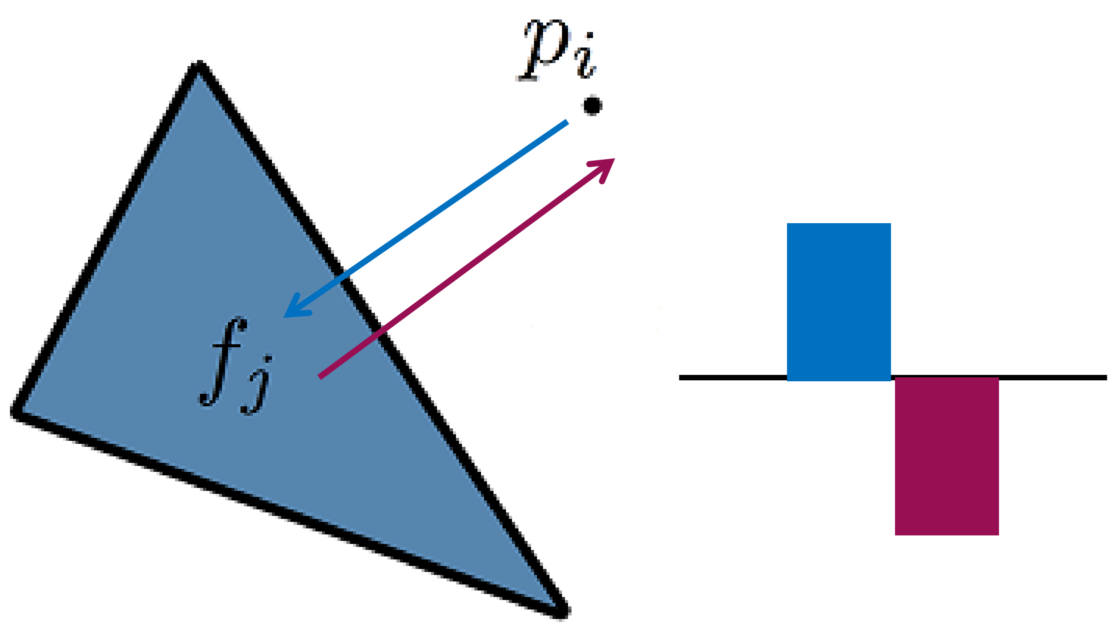


Rasterization and z-buffering are non-differentiable functions

Differentiable Rasterization



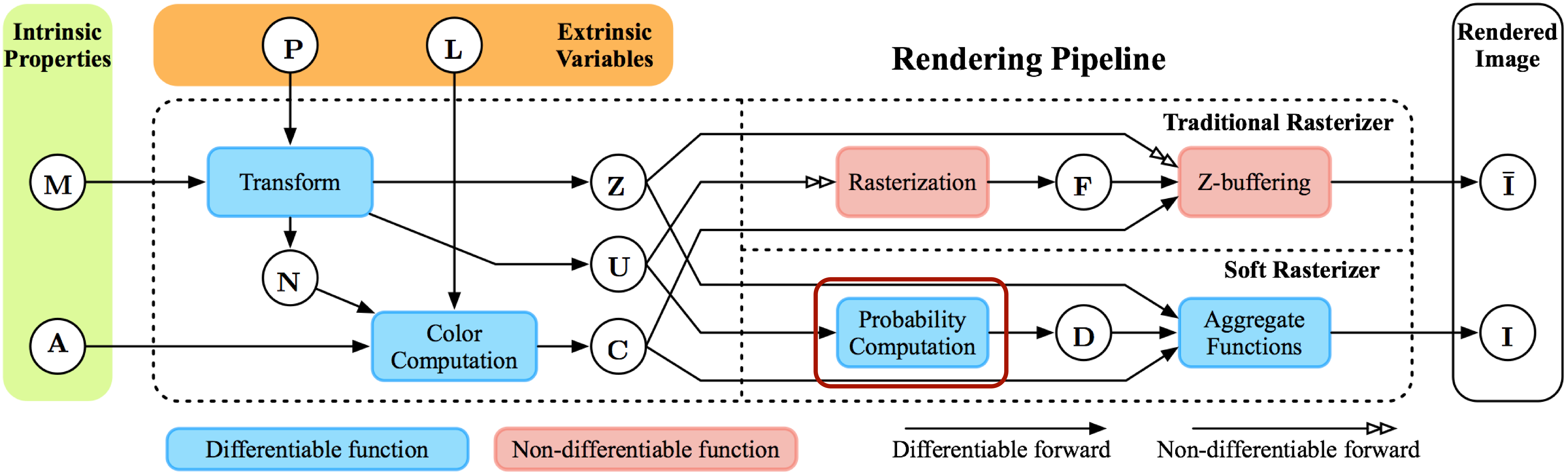
Traditional Rasterization



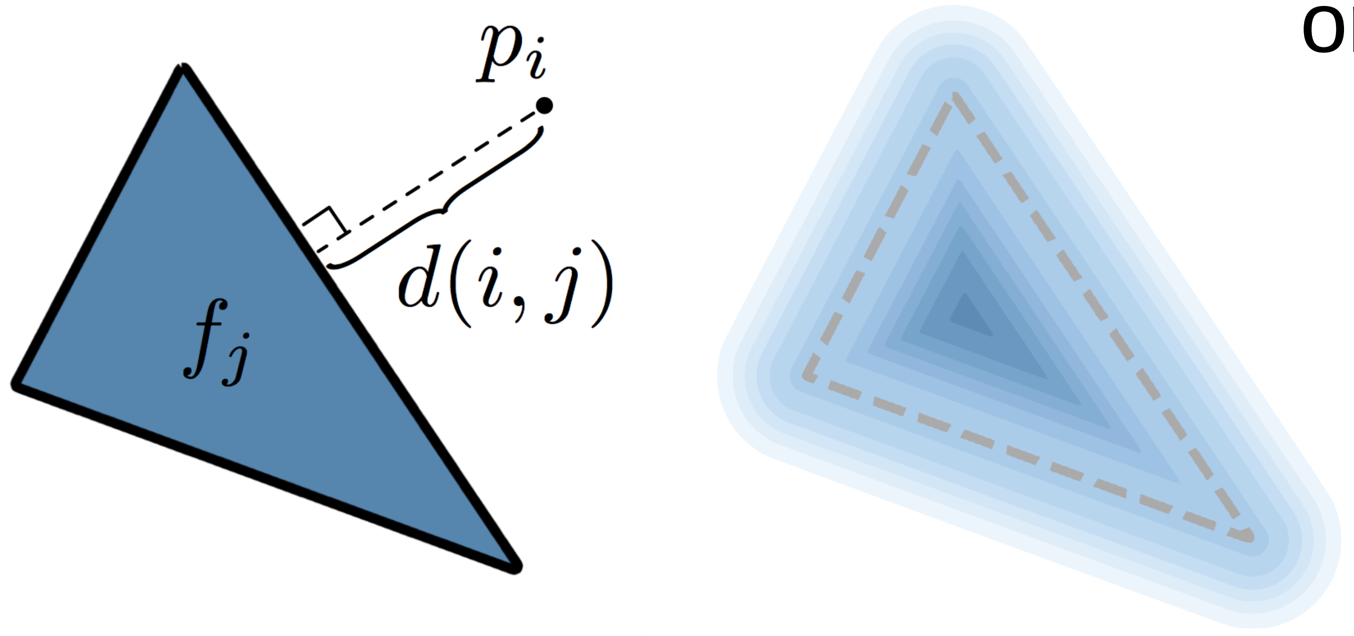
Color of p_i suffers from a sudden change when cross the edge of the triangle f_j

zero gradient in almost everywhere in the space

Differentiable Rasterization

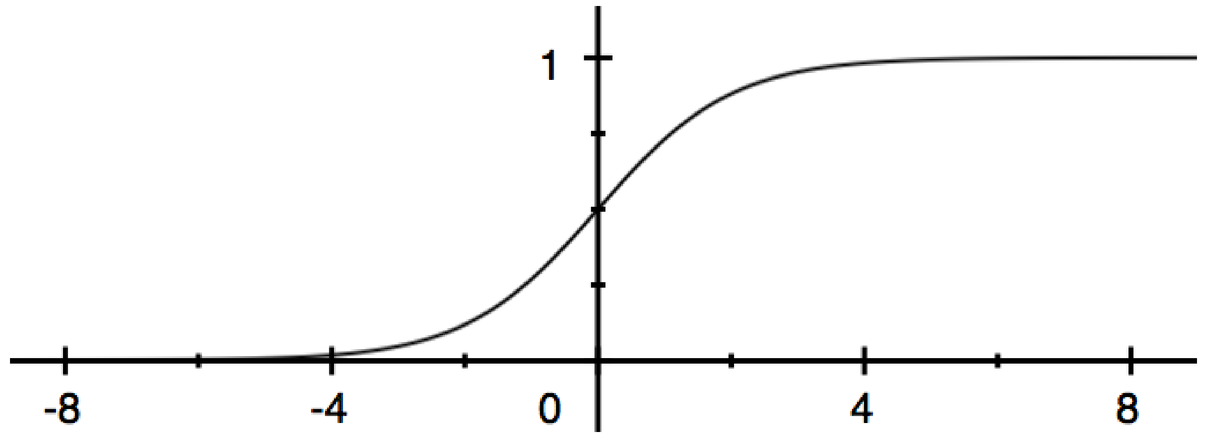


Soft Rasterization

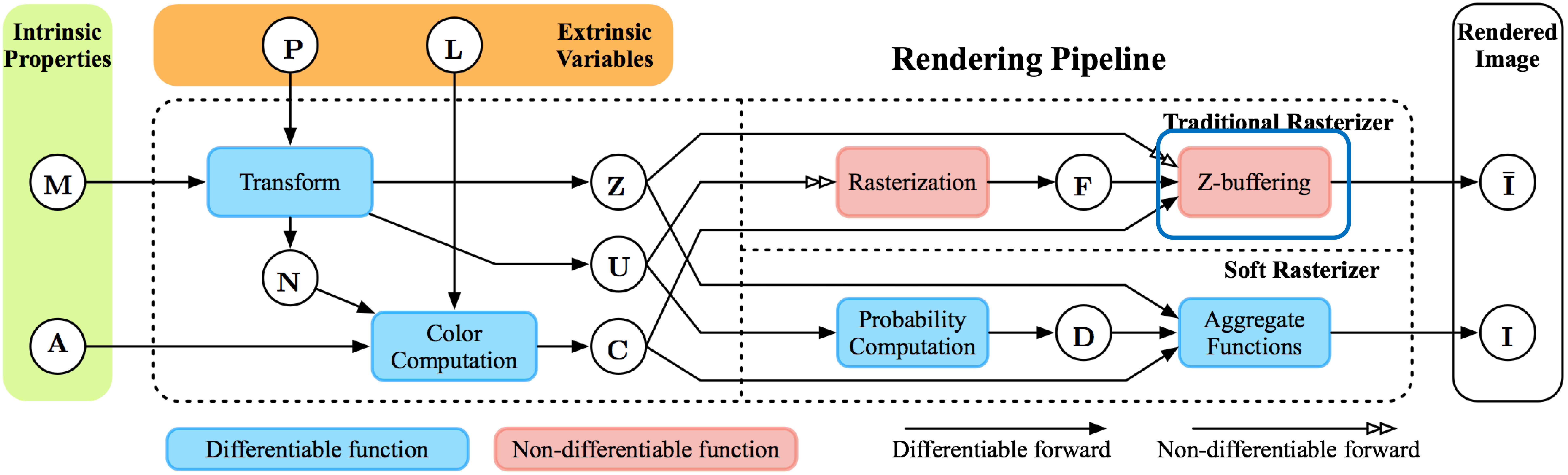


Change of color is formulated in a probabilistic way depending on *the distance between the pixel and triangle edge*.

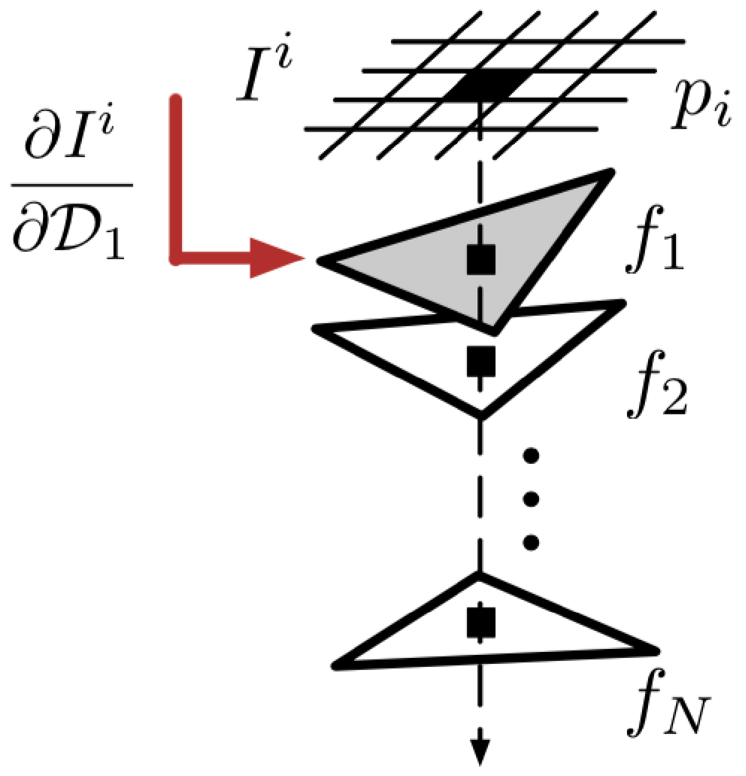
$$\mathcal{D}_j^i = sigmoid(\delta_j^i \cdot \frac{d^2(i,j)}{\sigma})$$



Differentiable Z-Buffering



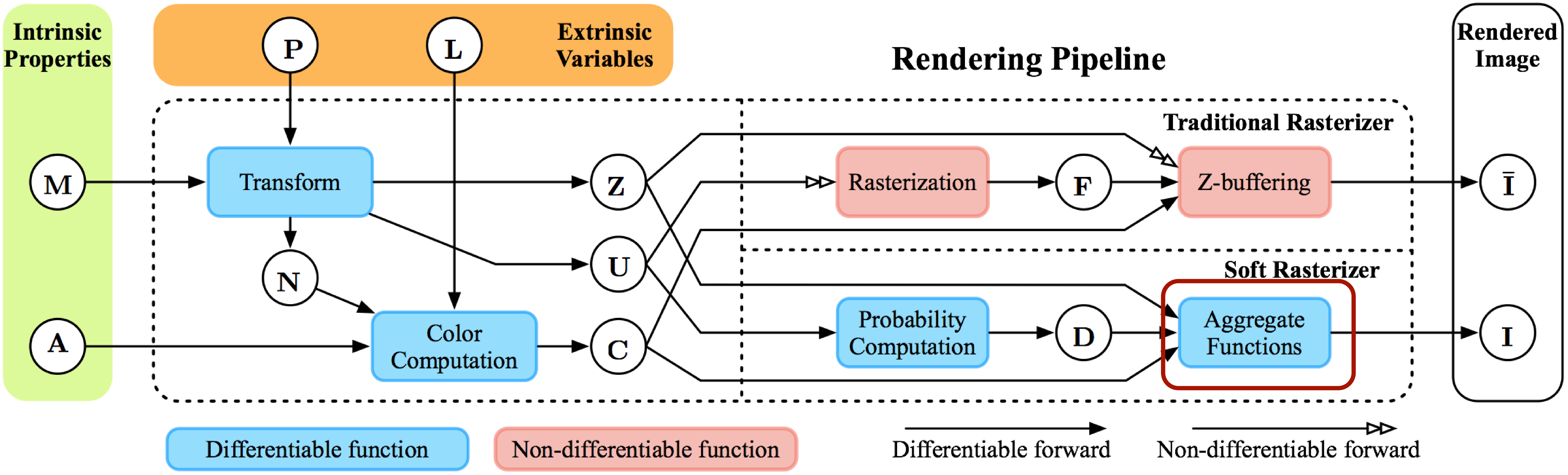
Traditional Z-Buffering



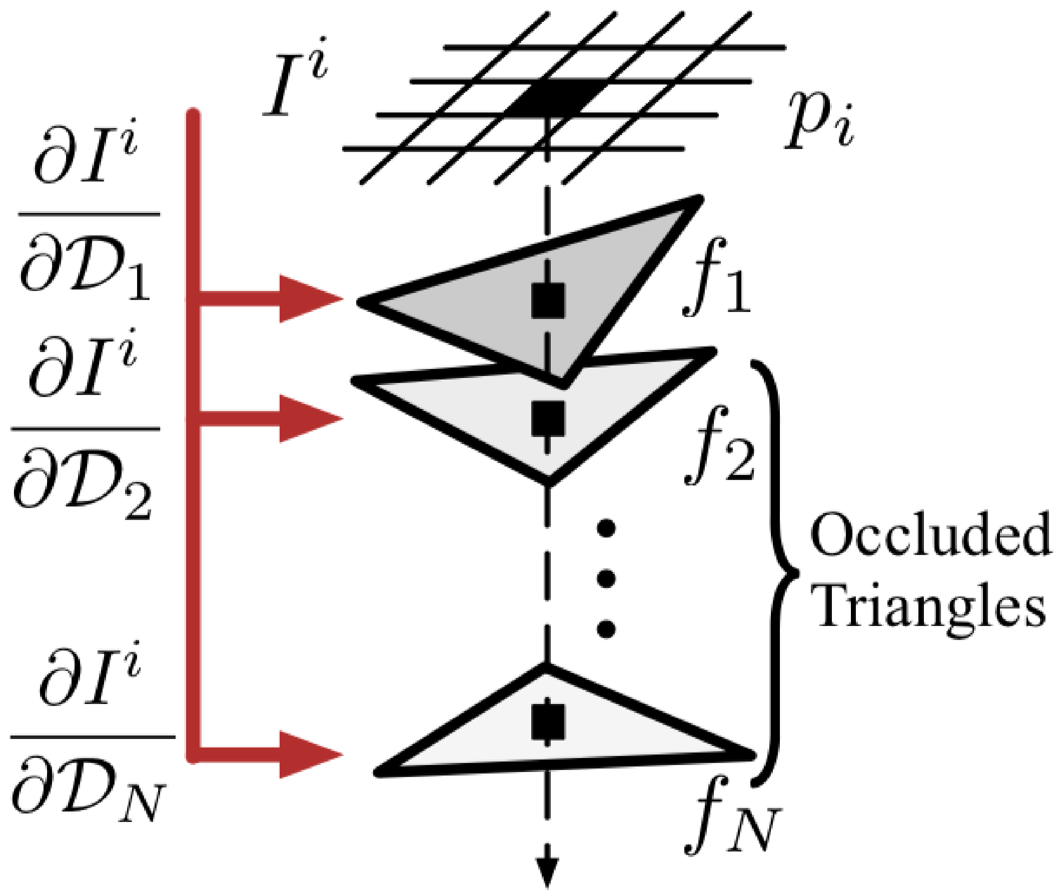
Color is determined by the nearest triangle

Non-differentiable **One-hot Voting**!

Differentiable Z-Buffering



Soft Z-Buffering



The final color is the **probabilistic aggregation** of **all possible triangles along the Z/depth direction** depending on their relative depth.

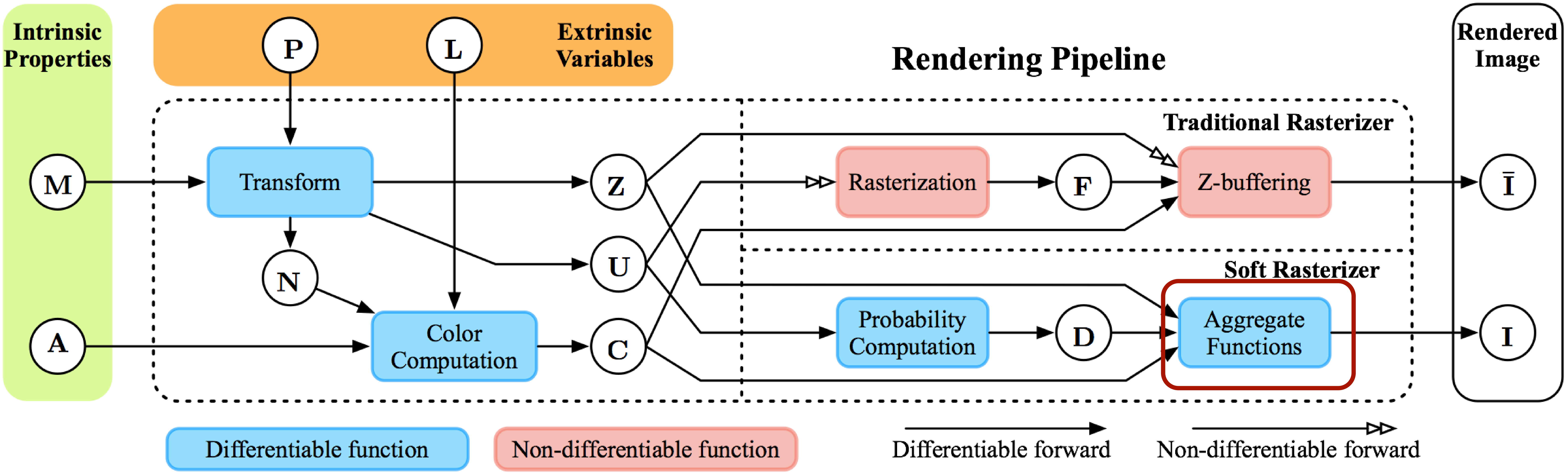
Aggregation Function

$$I^i = \mathcal{A}_S(\{C_j\}) = \sum_j w_j^i C_j^i + w_b^i C_b$$

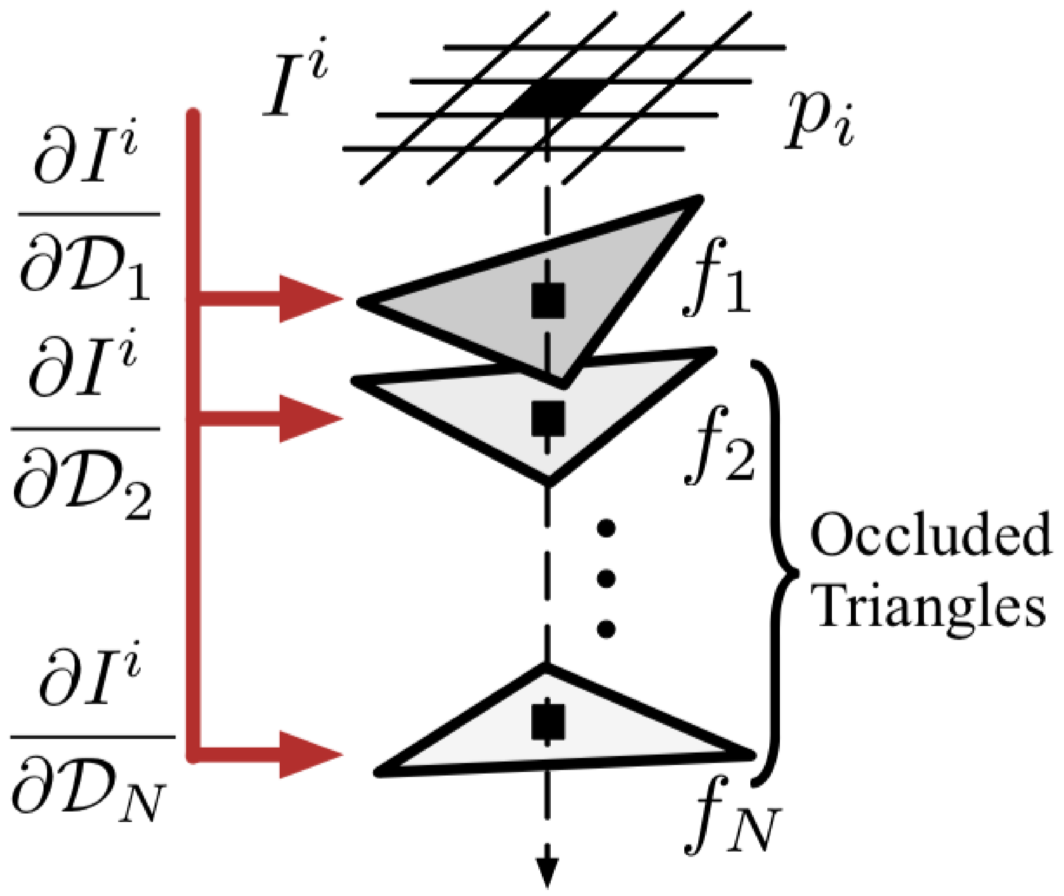
$$w_j^i = \frac{\mathcal{D}_j^i \exp(z_j^i/\gamma)}{\sum_k \mathcal{D}_k^i \exp(z_k^i/\gamma) + \exp(\epsilon/\gamma)}$$

1) Triangles closer to the image plane has higher contribution/gradient during optimization.
2) Enable gradient to flow into occluded triangles.

Differentiable Z-Buffering



Soft Z-Buffering



Aggregation Function can have different forms!

$$I^i = \mathcal{A}_S(\{C_j\}) = \sum_j w_j^i C_j^i + w_b^i C_b$$ — Color

$$I_s^i = \mathcal{A}_O(\{\mathcal{D}_j\}) = 1 - \prod_j (1 - \mathcal{D}_j^i)$$ — Silhouette

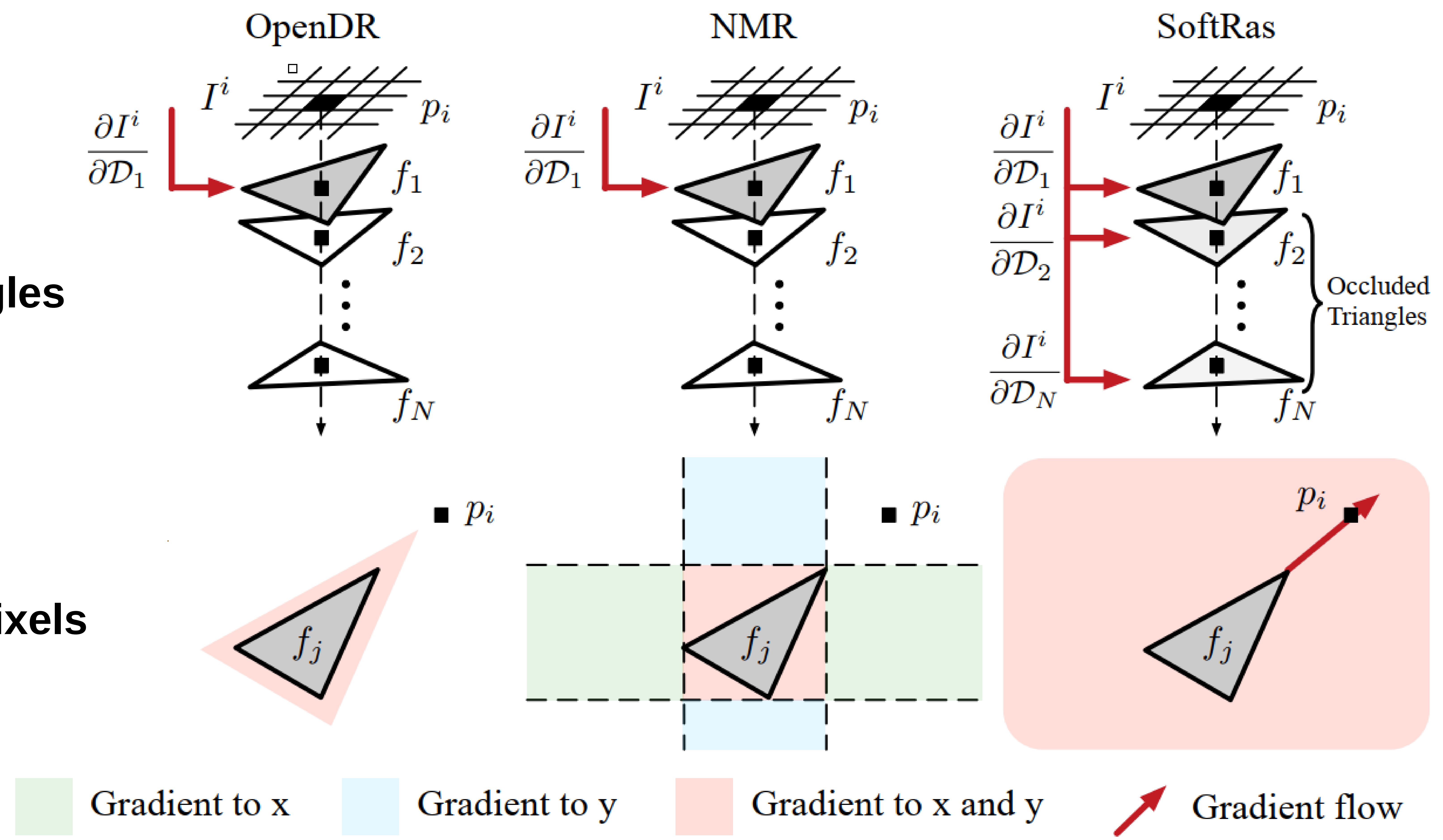
$\mathcal{A}_N$ = Neural Network — Color Silhouette

⋮

Comparisons of Different DRs

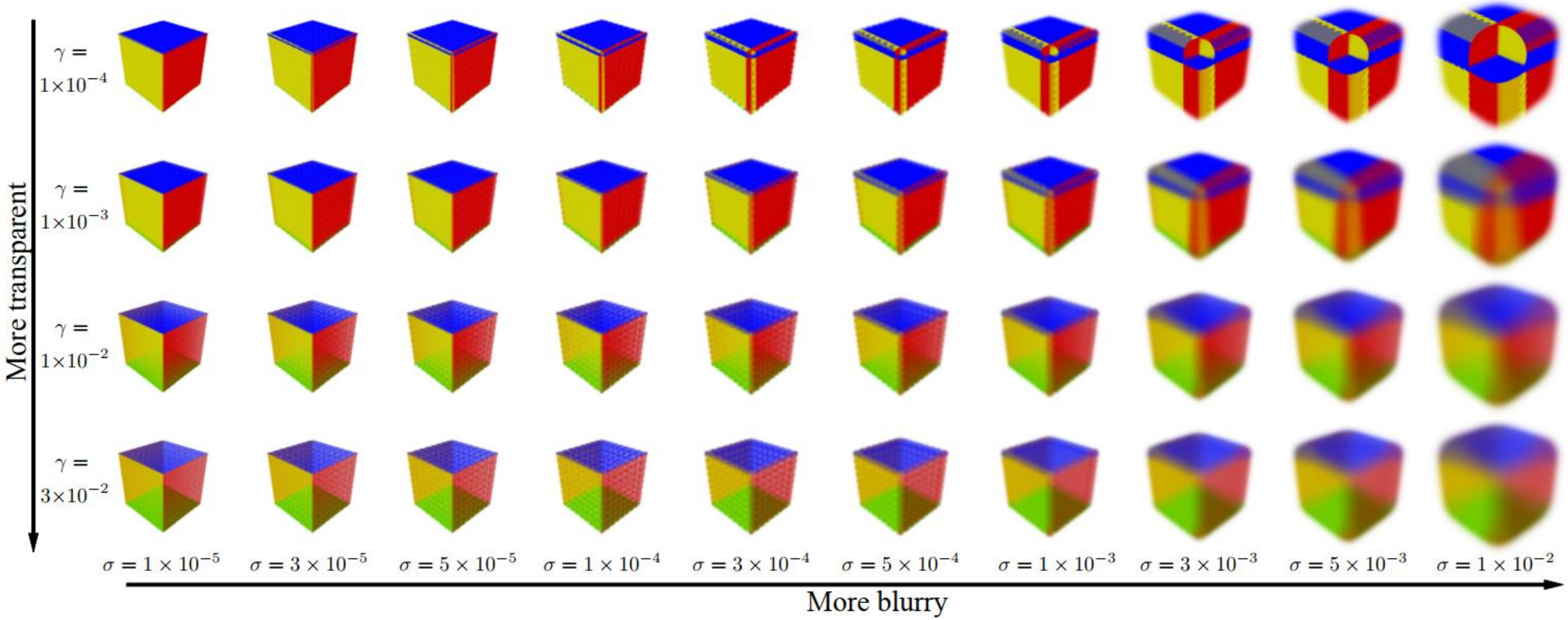
Gradient from pixels to triangles

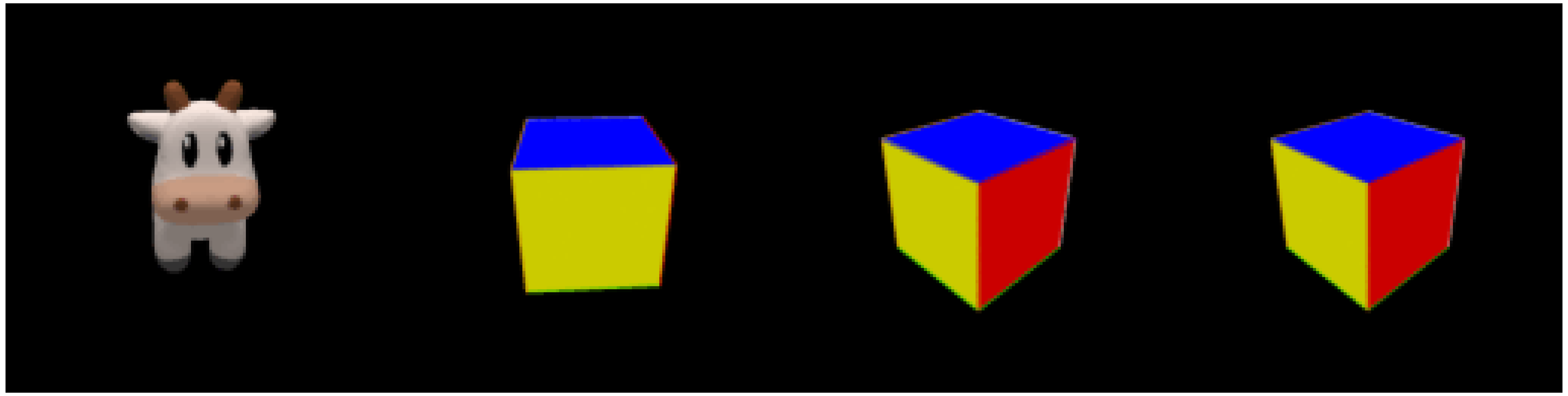
Screen-space gradient from pixels to vertices



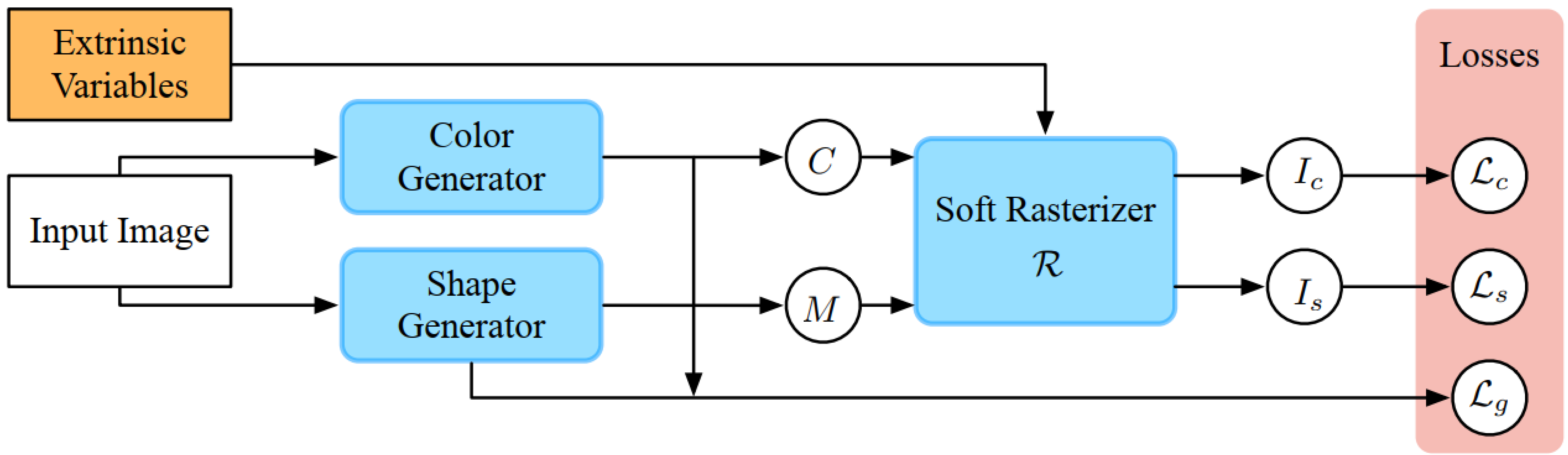
Applications – Forward Rendering

Controllable Rendering Effect

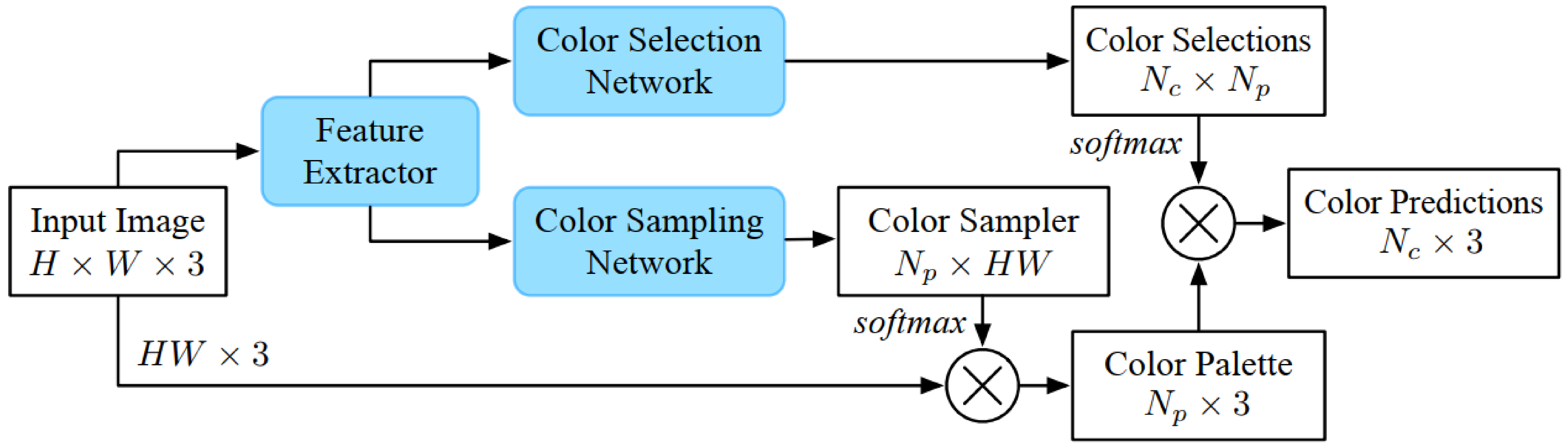




Applications – Single-view Mesh Reconstruction



Single-view Reconstruction Network



Color Generator

Loss function

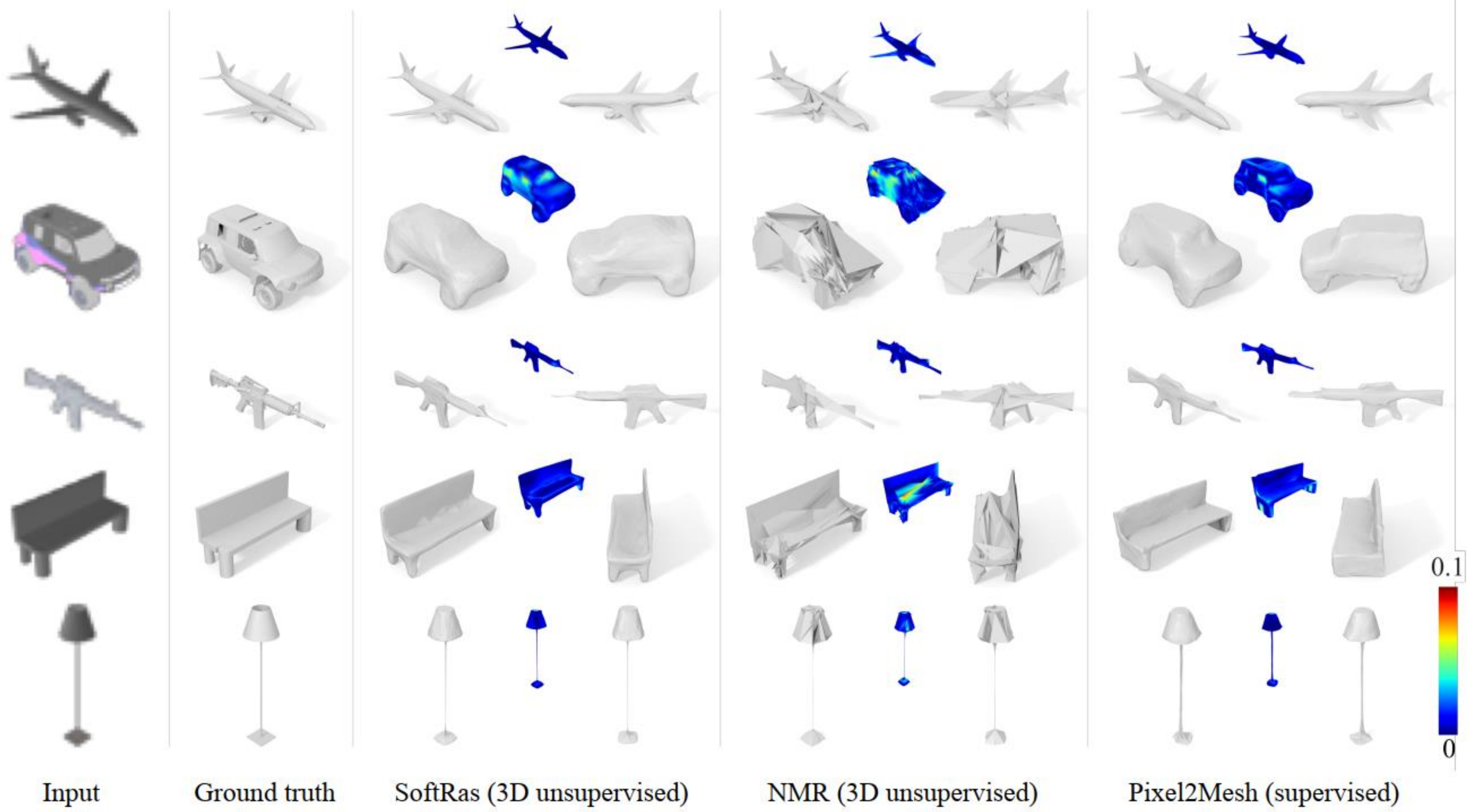
$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_c + \mu \mathcal{L}_g$$

Silhouette loss Color loss Geometry regularizer

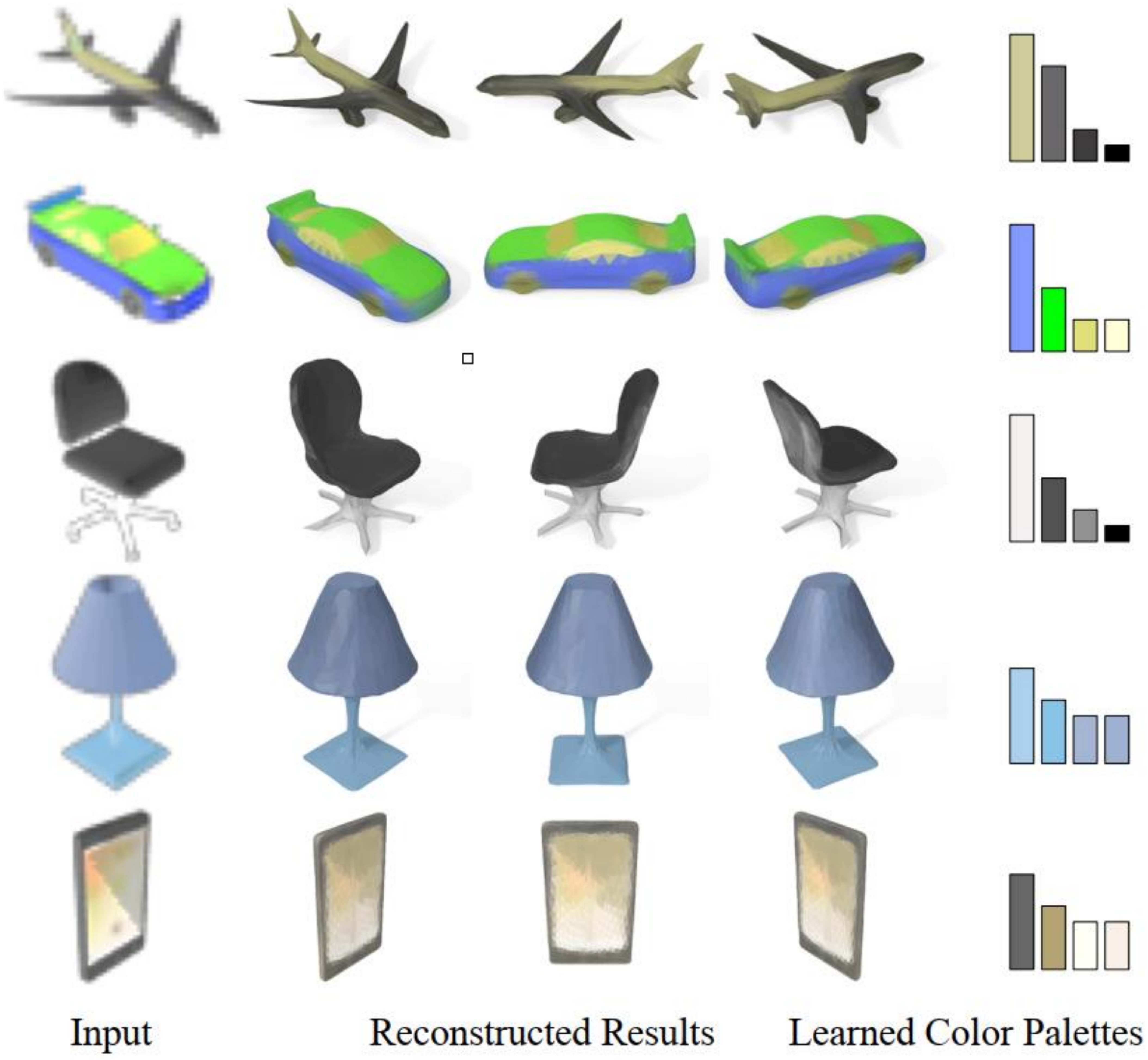
3D Unsupervised Learning!

Applications – Single-view Mesh Reconstruction

Qualitative Comparison



Applications – Single-view Mesh Reconstruction



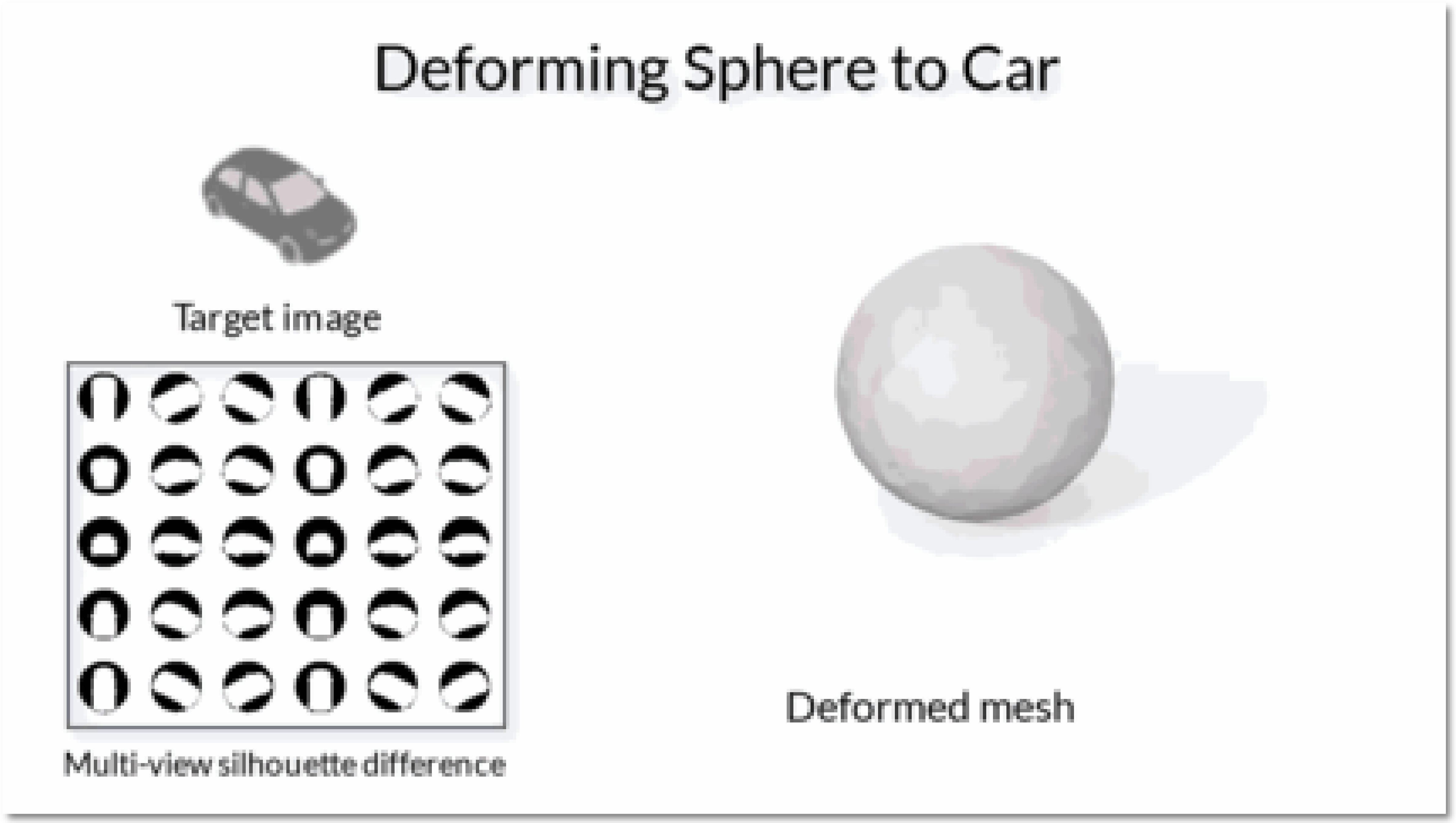
Color Reconstruction

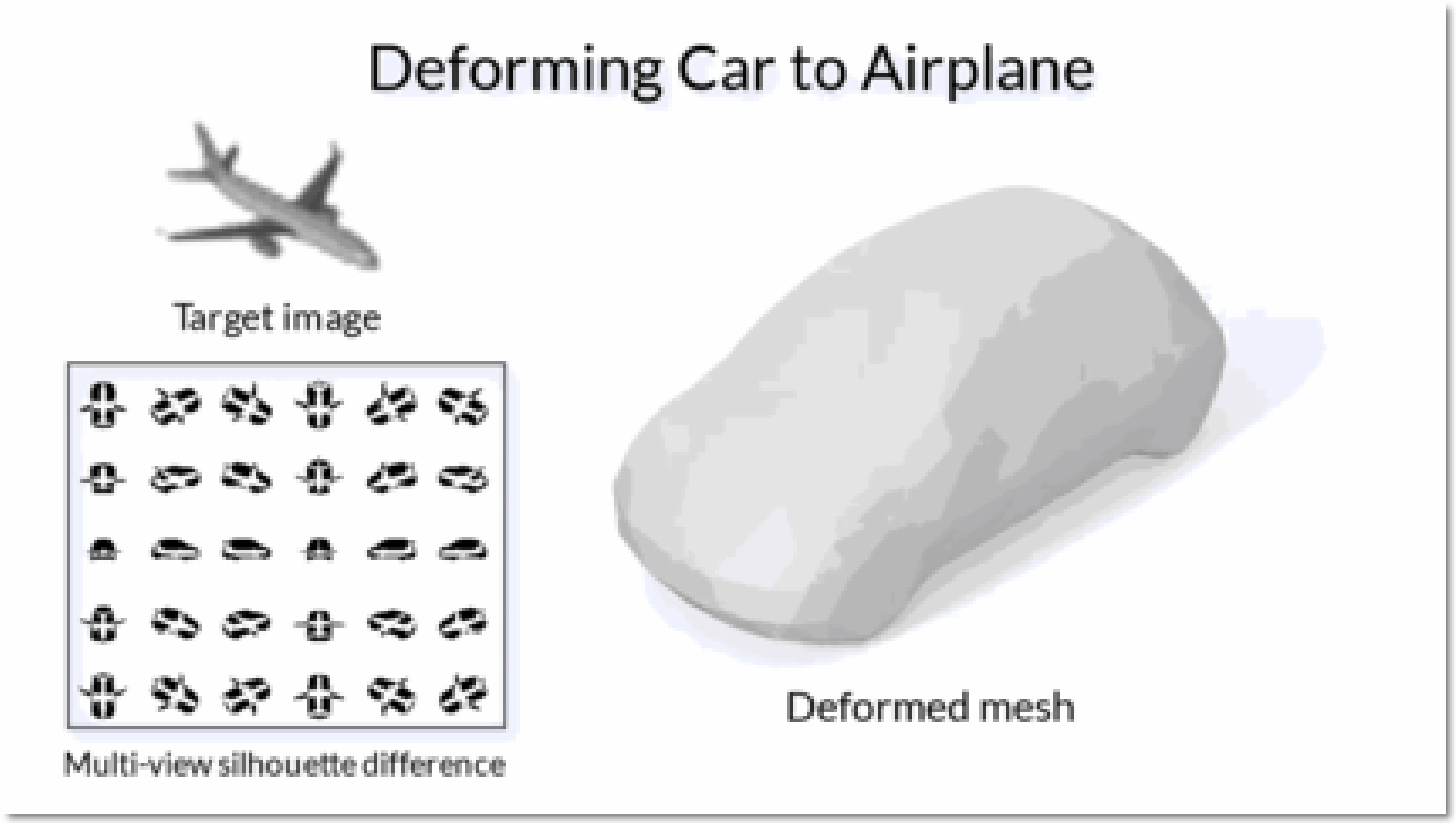
ShapeNet Dataset

Category	Airplane	Bench	Dresser	Car	Chair	Display	Lamp
Retrieval [47]	0.5564	0.4875	0.5713	0.6519	0.3512	0.3958	0.2905
Voxel [47]	0.5556	0.4924	0.6823	0.7123	0.4494	0.5395	0.4223
NMR [19]	0.6172	0.4998	0.7143	0.7095	0.4990	0.5831	0.4126
Ours (sil.)	0.6419	0.5080	0.7116	0.7697	0.5270	0.6156	**0.4628**
Ours (full)	**0.6670**	**0.5429**	**0.7382**	**0.7876**	**0.5470**	**0.6298**	0.4580
Category	Speaker	Rifle	Sofa	Table	Phone	Vessel	Mean
Retrieval [47]	0.4600	0.5133	0.5314	0.3097	0.6696	0.4078	0.4766
Voxel [47]	0.5868	0.5987	0.6221	0.4938	0.7504	0.5507	0.5736
NMR [19]	0.6536	0.6322	0.6735	0.4829	0.7777	0.5645	0.6015
Ours (sil.)	0.6654	**0.6811**	0.6878	0.4487	0.7895	0.5953	0.6234
Ours (full)	**0.6807**	0.6702	**0.7220**	**0.5325**	**0.8127**	**0.6145**	**0.6464**

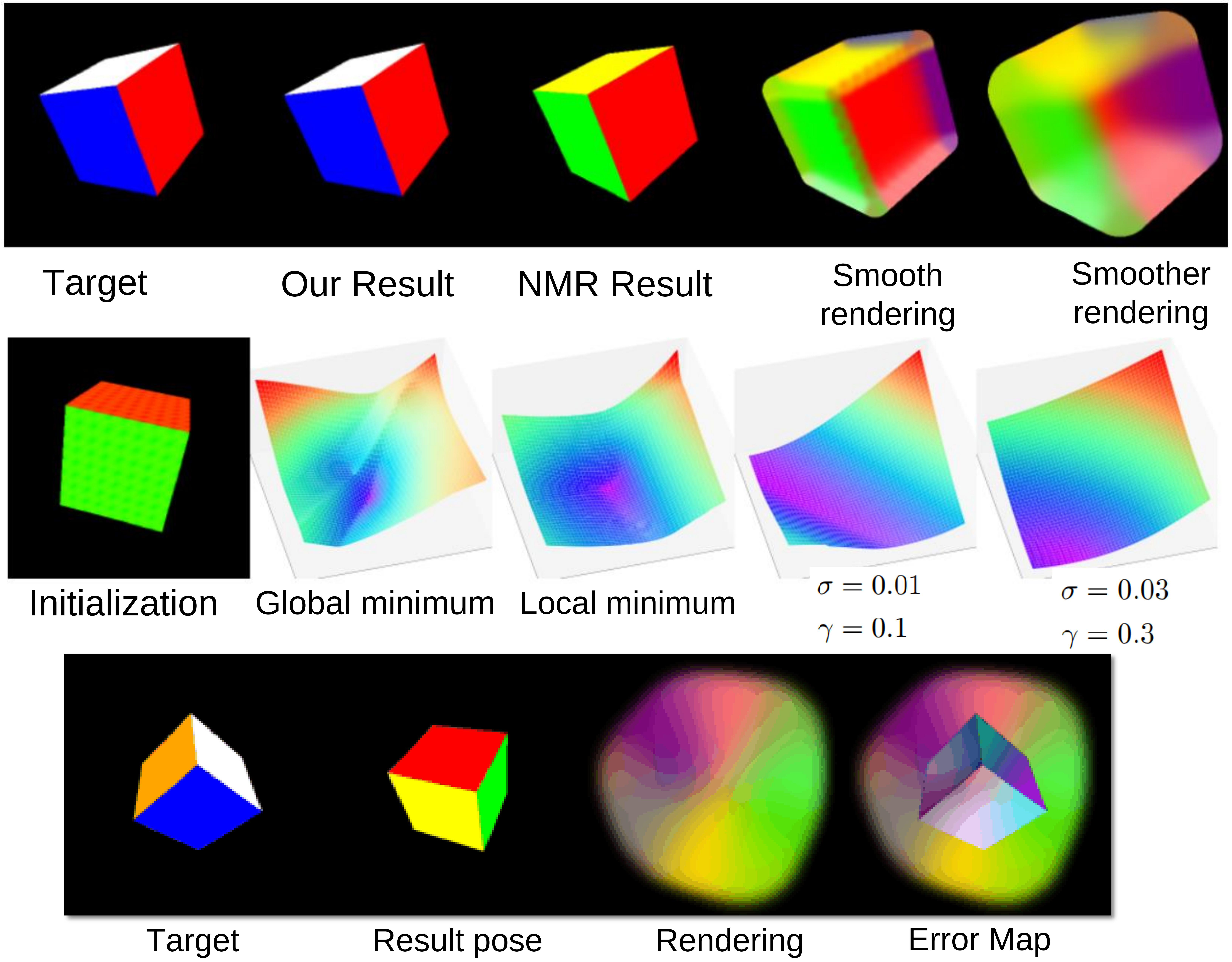
Quantitative Comparison

Applications – Shape Deformation

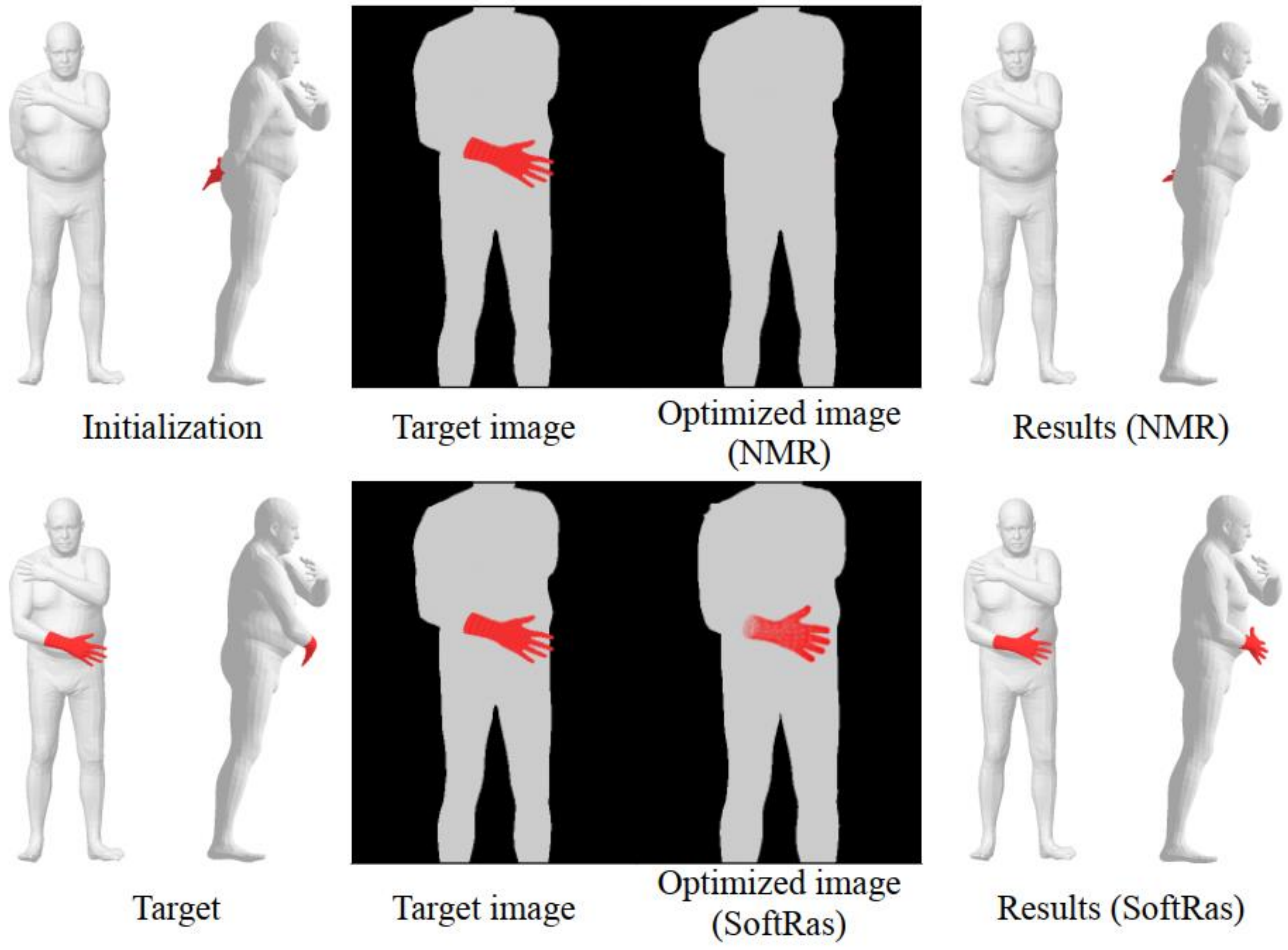


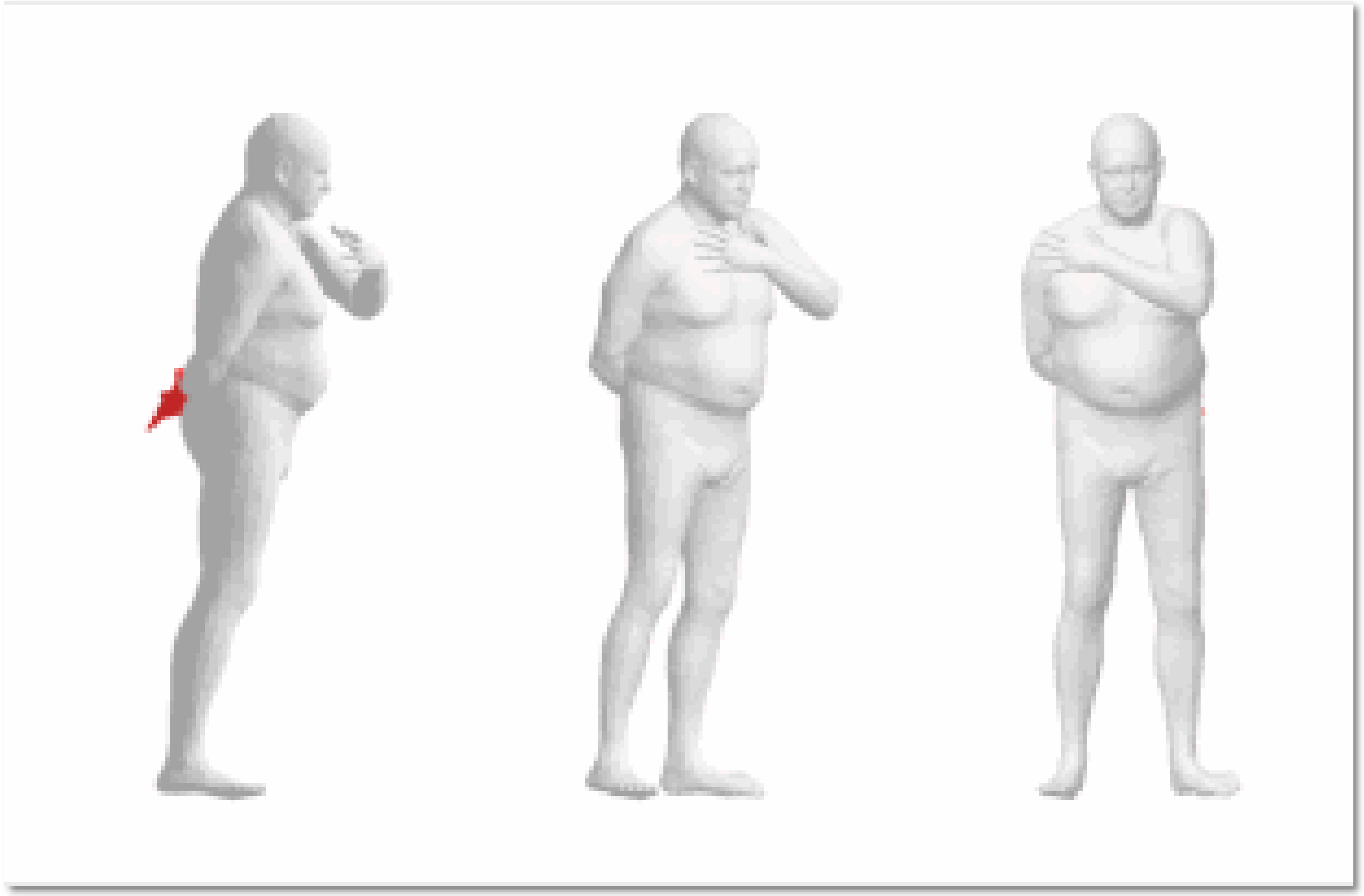


Applications – Rigid Pose Estimation



Applications – Non-Rigid Pose Estimation





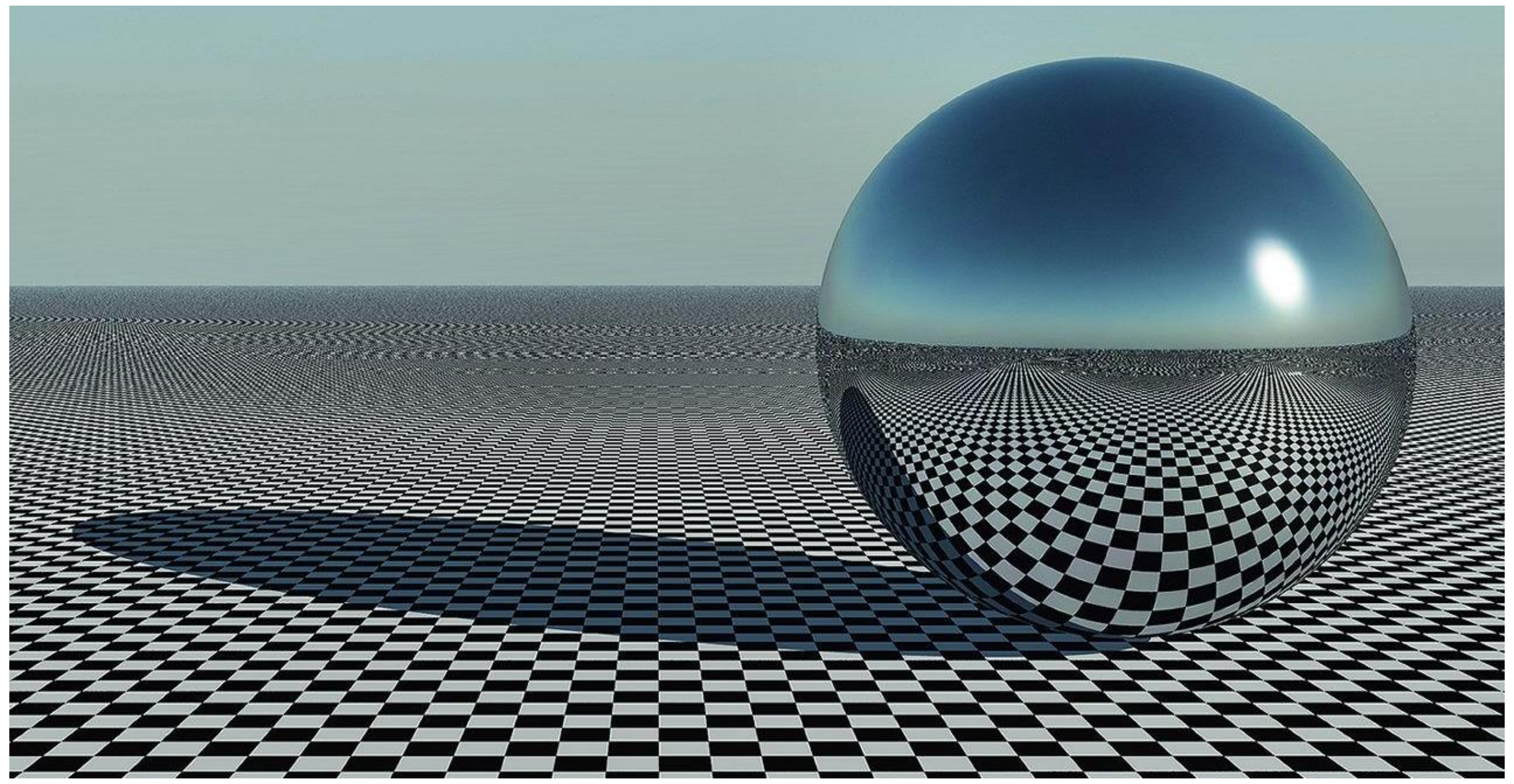
Non-Rigid Pose Optimization

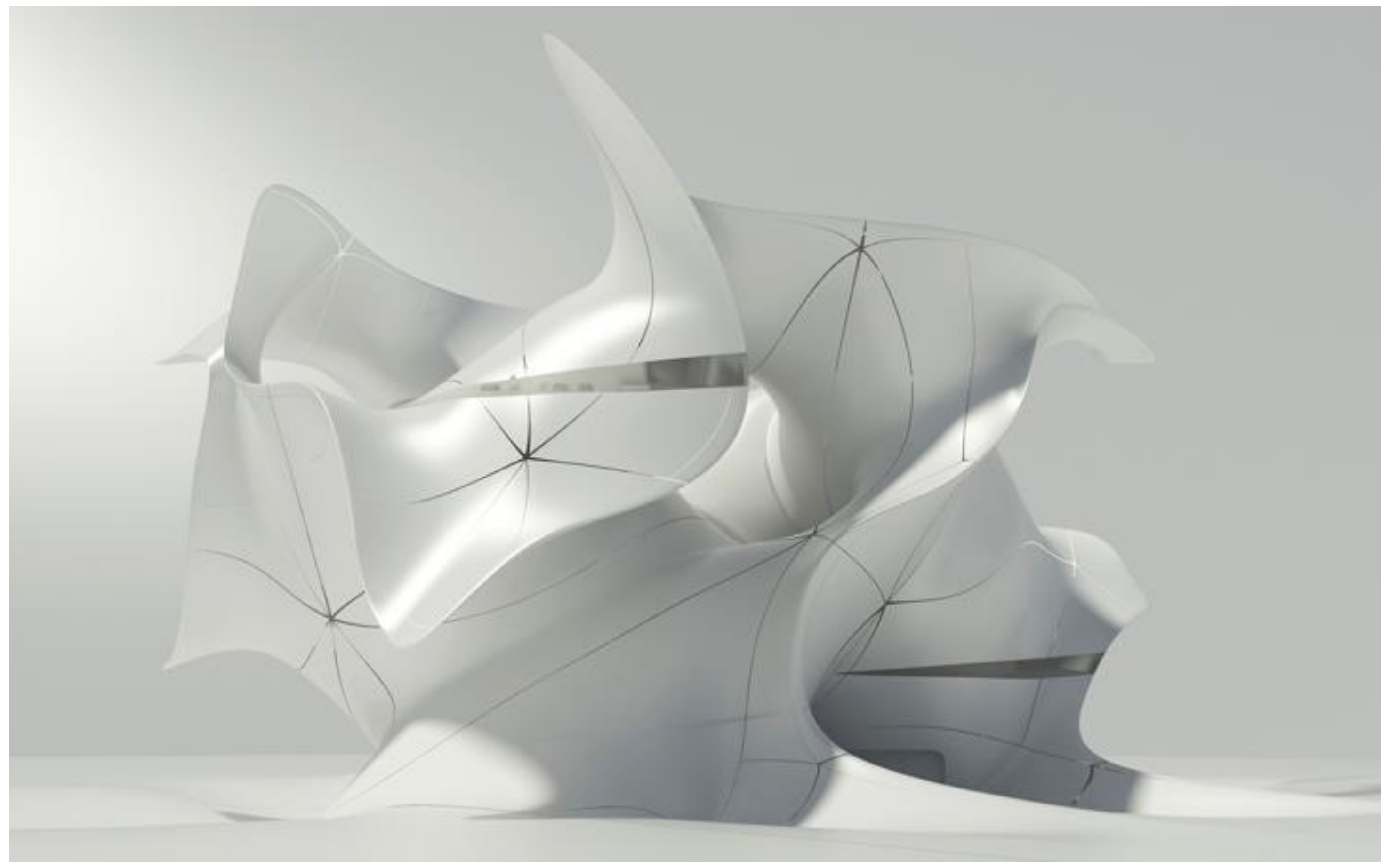
Differentiable Rendering for Implicit Surface

Learning to Infer Implicit Surfaces without 3D Supervision, NeurIPS'19

What is Implicit Surface?

How to define a unit sphere?

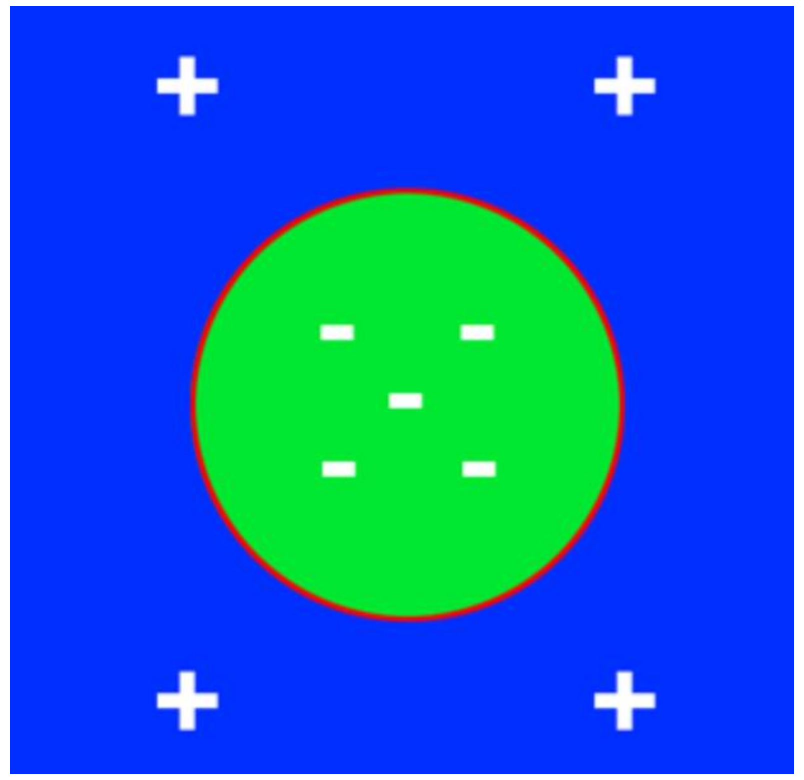


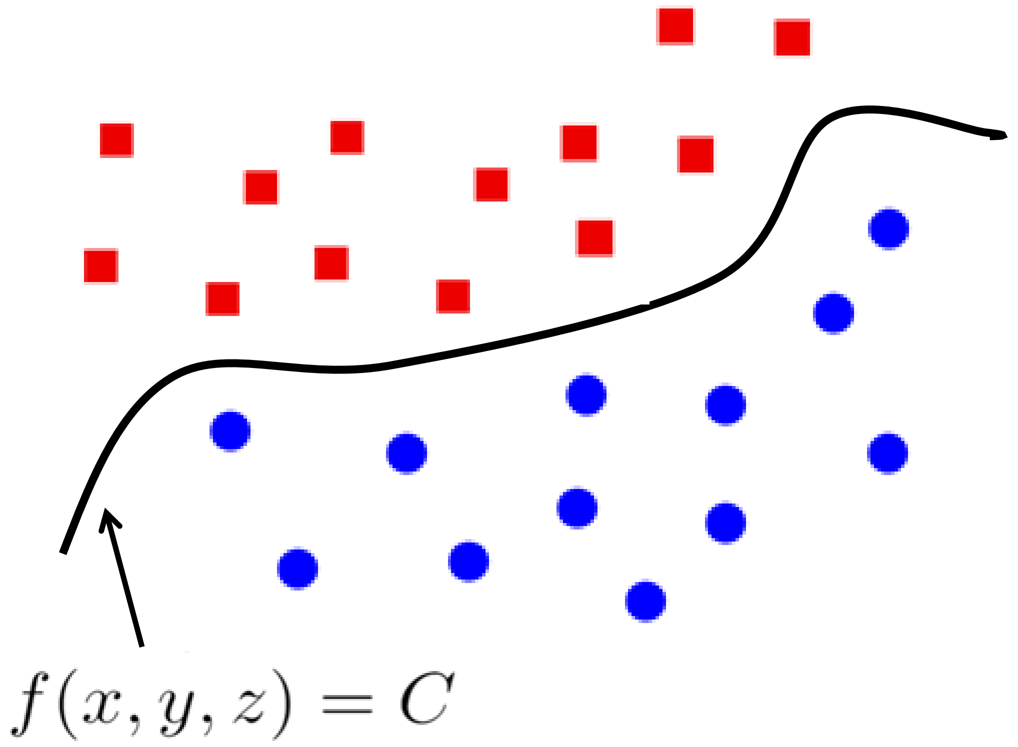


$$\underline{x^2 + y^2 + z^2 - 1} = \underline{0}$$

$$f(x, y, z) = C$$

Iso-surface

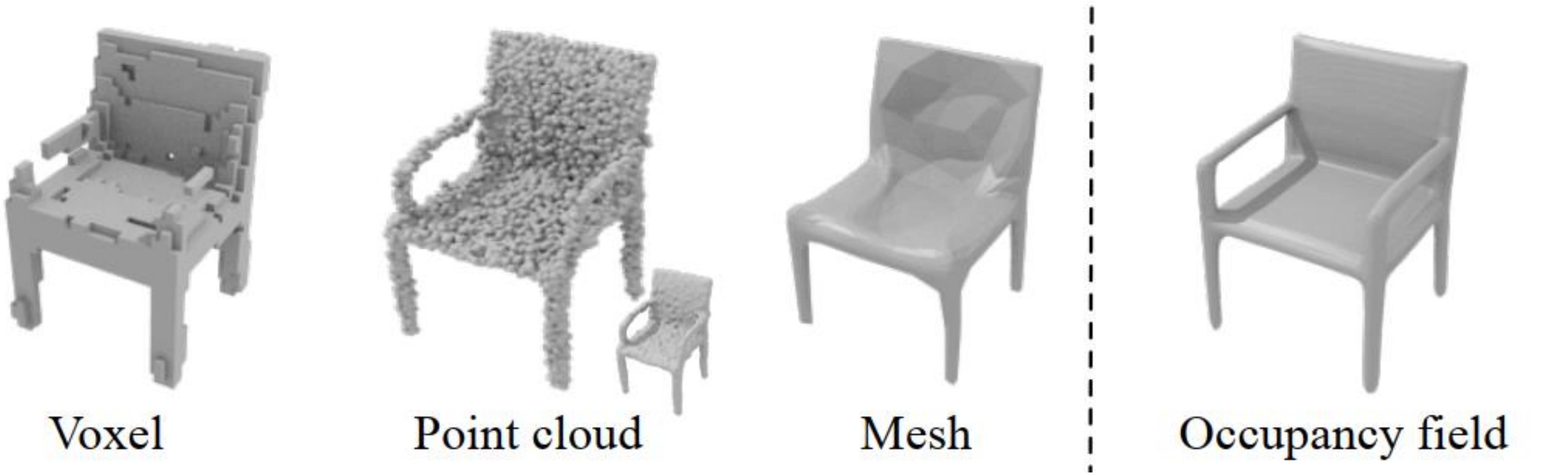




Implicit surface can be instantiated as mesh using Marching Cube algorithm.

Implicit Surface v.s. Explicit Representations

Explicit Representations | **Implicit Surface**



Voxel: +Topology, -Fidelity

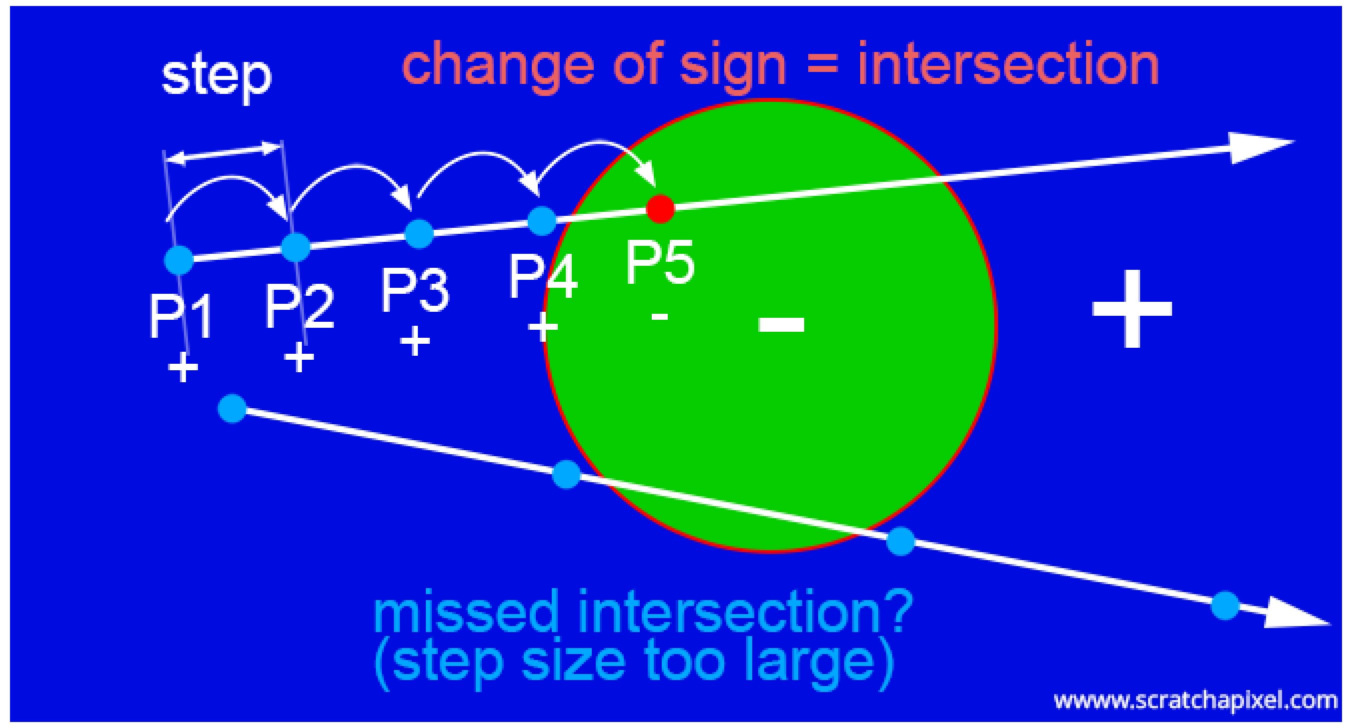
Point cloud: +Topology, -Fidelity

Mesh: -Topology, +Fidelity

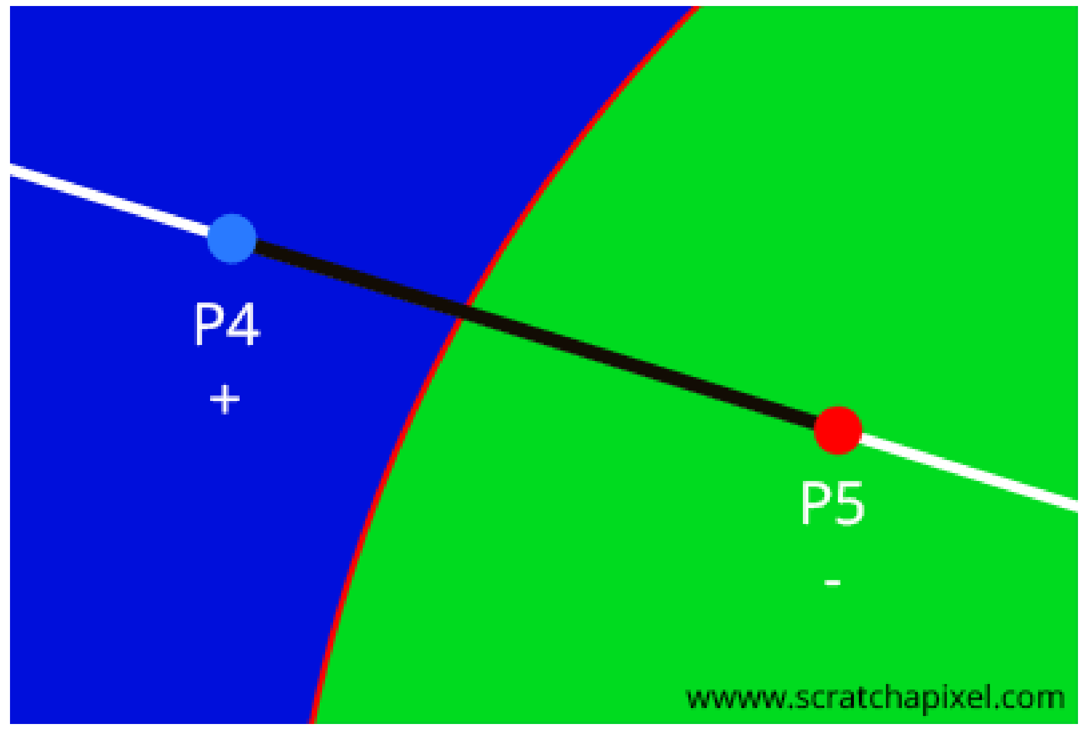
Occupancy field: +Topology, +Fidelity

Conventional Technique for Implicit Surface Rendering

Ray marching

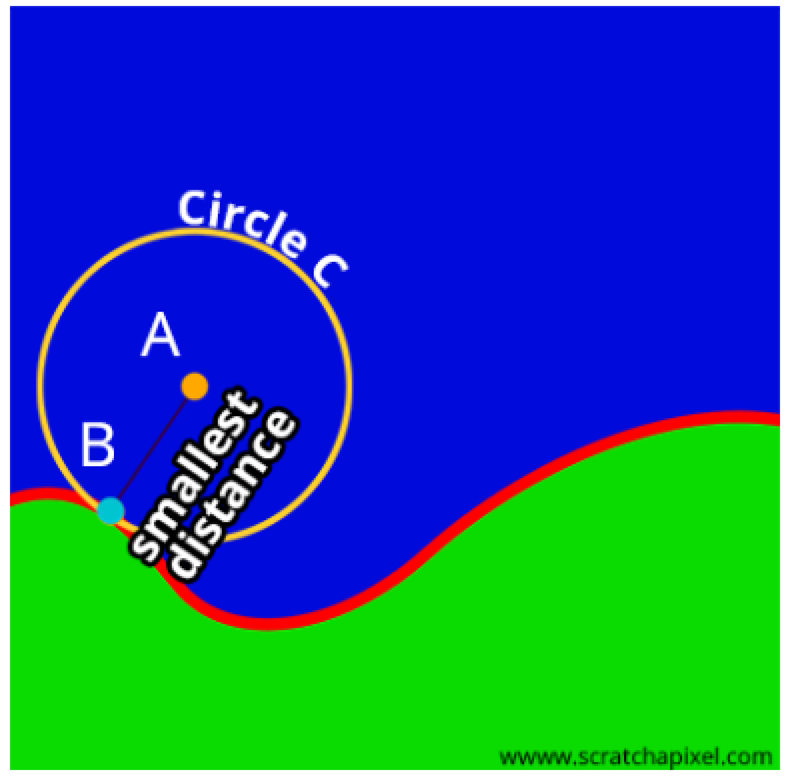


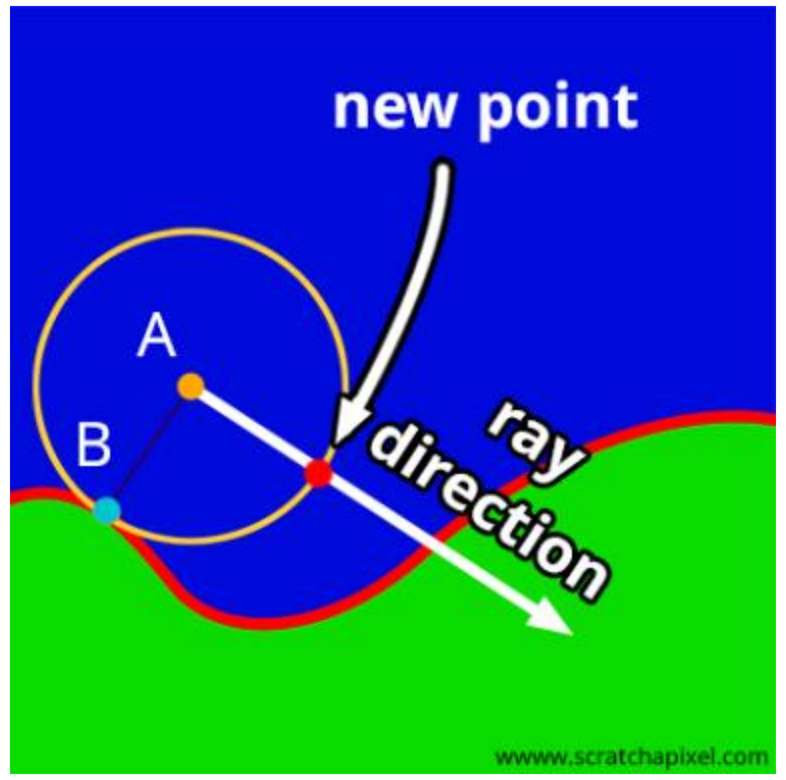
line search

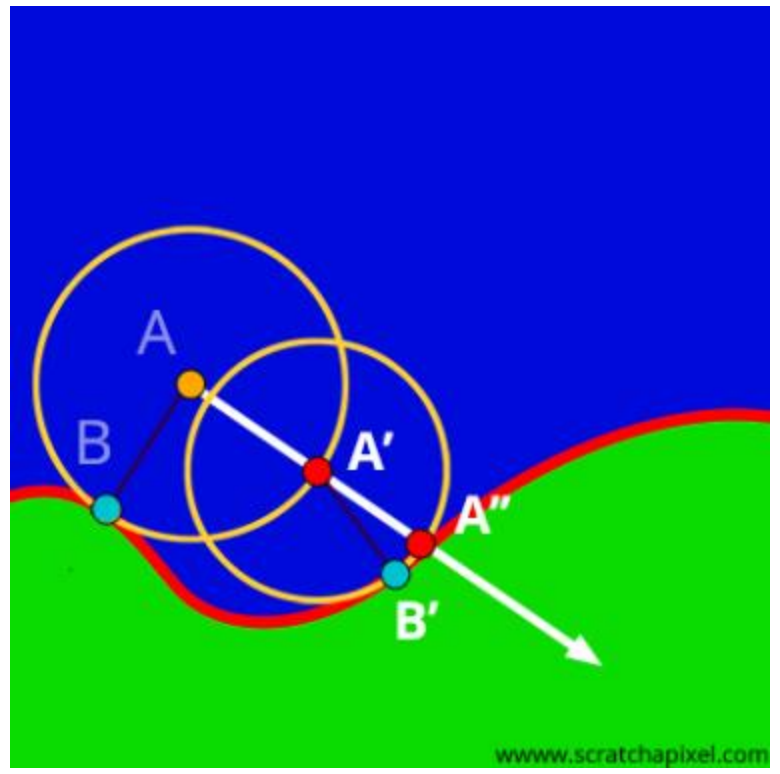


binary search

Sphere tracing





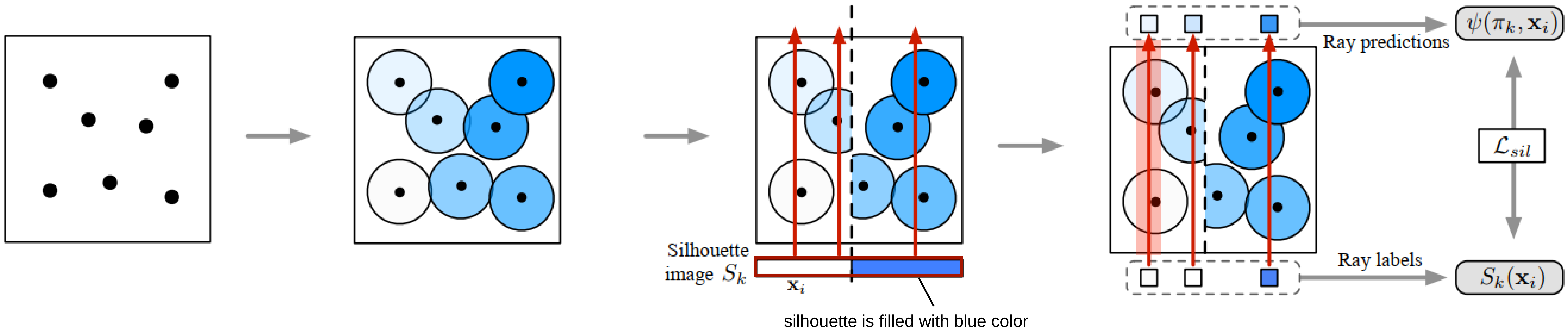


Time consuming and Non-differentiable!

Image credit: www.scratchapixel.com

Proposed Differentiable Implicit Renderer

Ray-based Field Probing Technique



Distribute 3D anchor points

Occupancy field evaluation

1) Sense the field (deeper blue -> higher value)
2) Each anchor point has spherical supporting region for computing ray-anchor intersection

Probing ray casting with boundary-aware assignment

Ray passes pixels inside/outside silhouette

Anchor points lying outside/inside of the 3D object are ignored

Aggregating intersected anchors along rays

1) Aggregate the information from the intersected anchor points via max pooling
2) Compare the prediction with the GT label in the image space

Importance Sampling

How to effectively sample anchor points and probing rays?

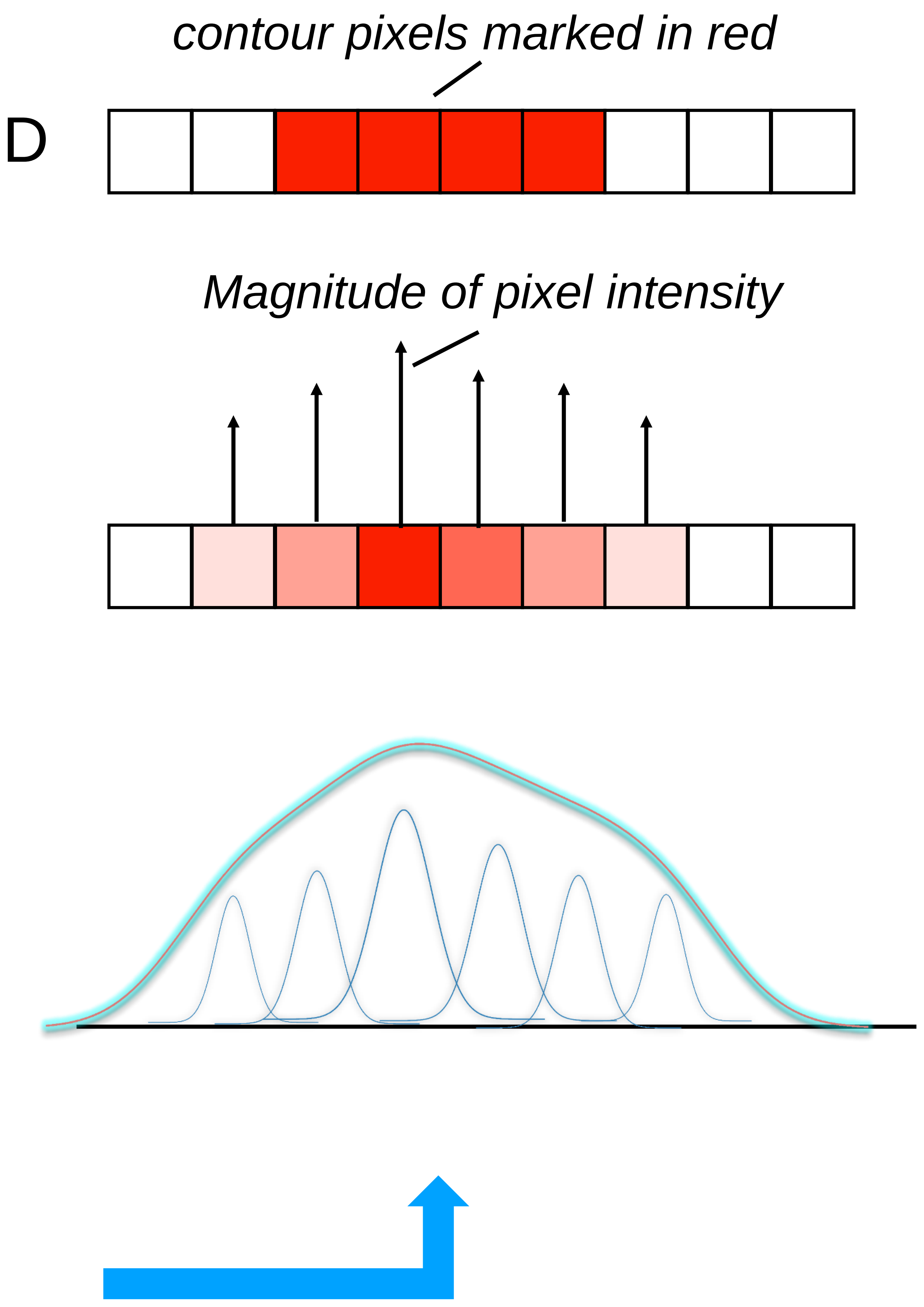
Sampling on 2D Image → **Importance sampling based on Gaussian mixture distribution computed from 2D object silhouette**

2D Contour map visualized in 1D

Apply Gaussian smoothing

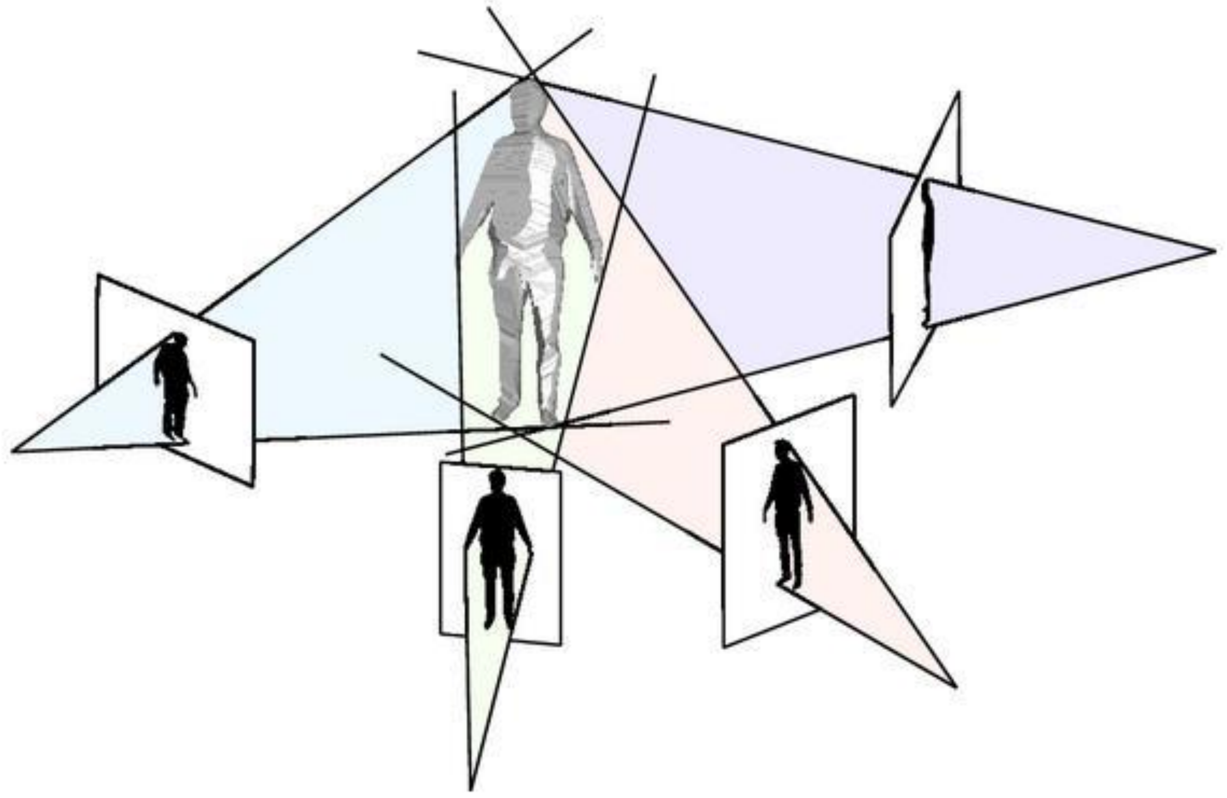
Generate Gaussian mixture distribution based on the obtained pixel intensity

Draw 2D samples from the resulted distribution



Similar Sampling Strategy applied to 3D Anchor points

3D Contour is computed as the boundary of the visual hull



Geometric Regularization for Implicit Surface

Regularizing geometric properties of implicit surface is challenging due to the lack of explicit geometric entity.

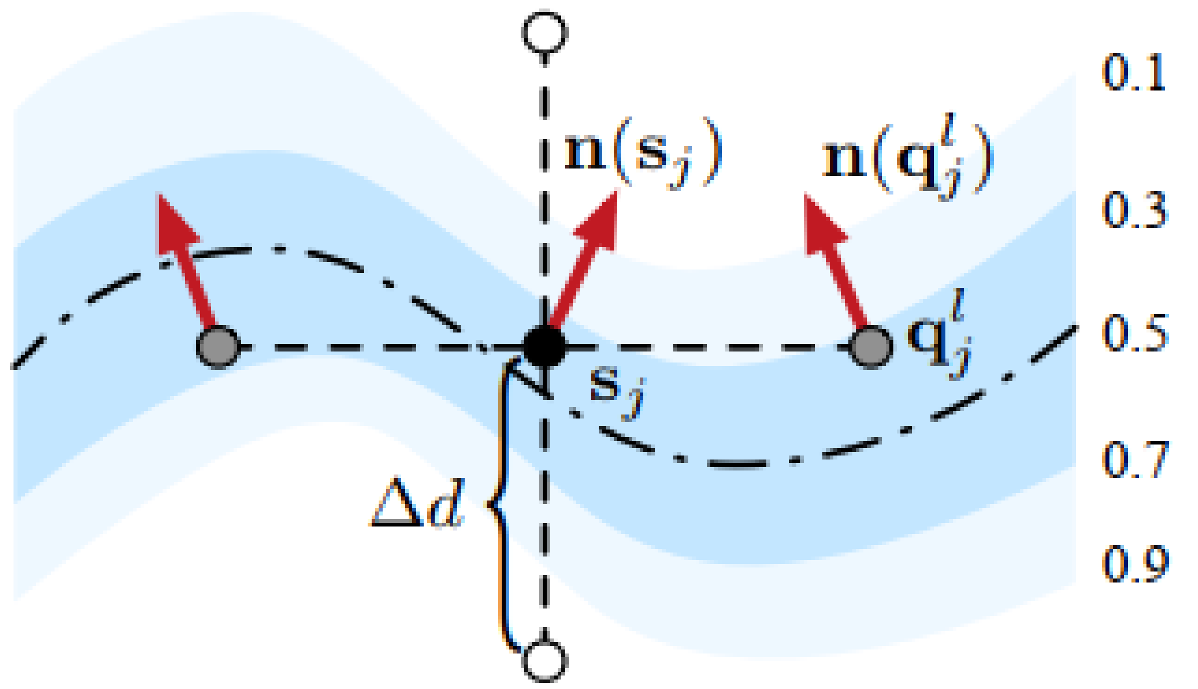
Implicit surface derivatives based on finite difference:

$$\frac{\delta^n \phi}{\delta \mathbf{p}_j^n} = \frac{1}{\Delta d^n} \sum_{l=0}^{n} (-1)^l \binom{n}{l} \phi(\mathbf{p}_j + (\frac{n}{2} - l)\Delta d)$$

Used to compute normal and other high-order derivatives at point $\mathbf{p}_j$

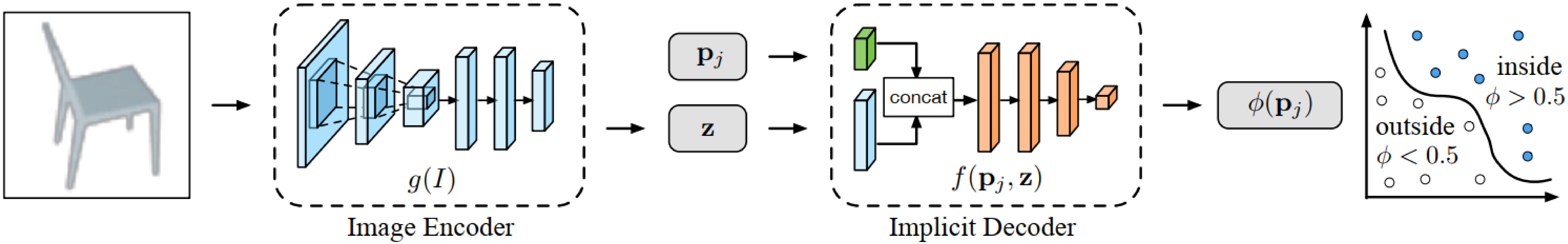
Geometric regularization based on Importance Weighting:

$$\mathcal{L}_{geo} = \frac{1}{N_p} \sum_{j=1}^{N_p} W(\phi(\mathbf{s}_j)) \frac{\sum_{l=1}^{6} W(\phi(\mathbf{q}_j^l)) \|\mathbf{n}(\mathbf{s}_j) - \mathbf{n}(\mathbf{q}_j^l)\|_p^p}{\sum_{l=1}^{6} W(\phi(\mathbf{q}_j^l))}$$



Unsupervised Learning of Implicit Surfaces

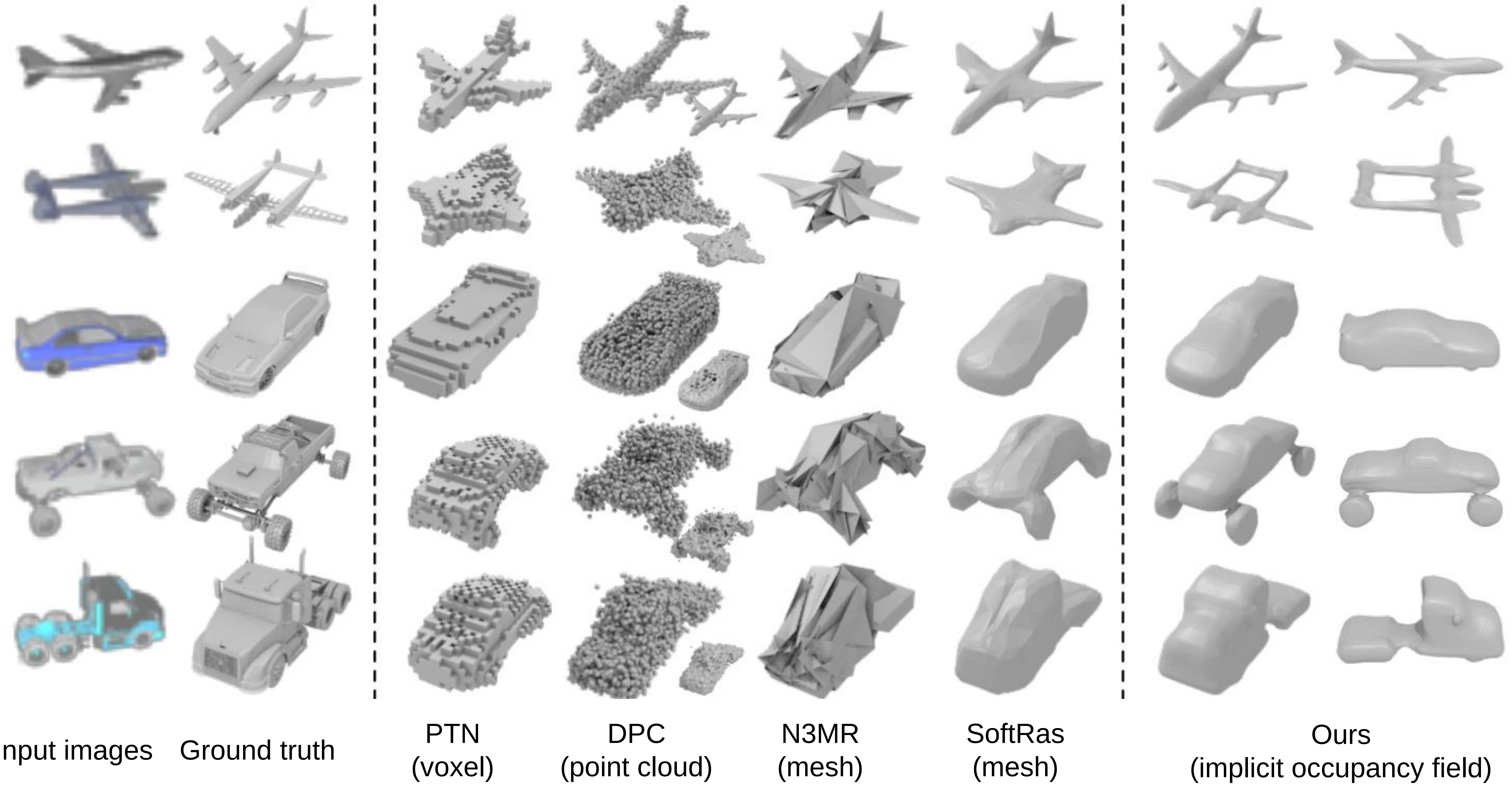
Network Structure



Loss function $$\mathcal{L} = \mathcal{L}_{sil} + \lambda\mathcal{L}_{geo}$$

Results

Qualitative Results of Single-view Reconstruction using Different Surface Representations

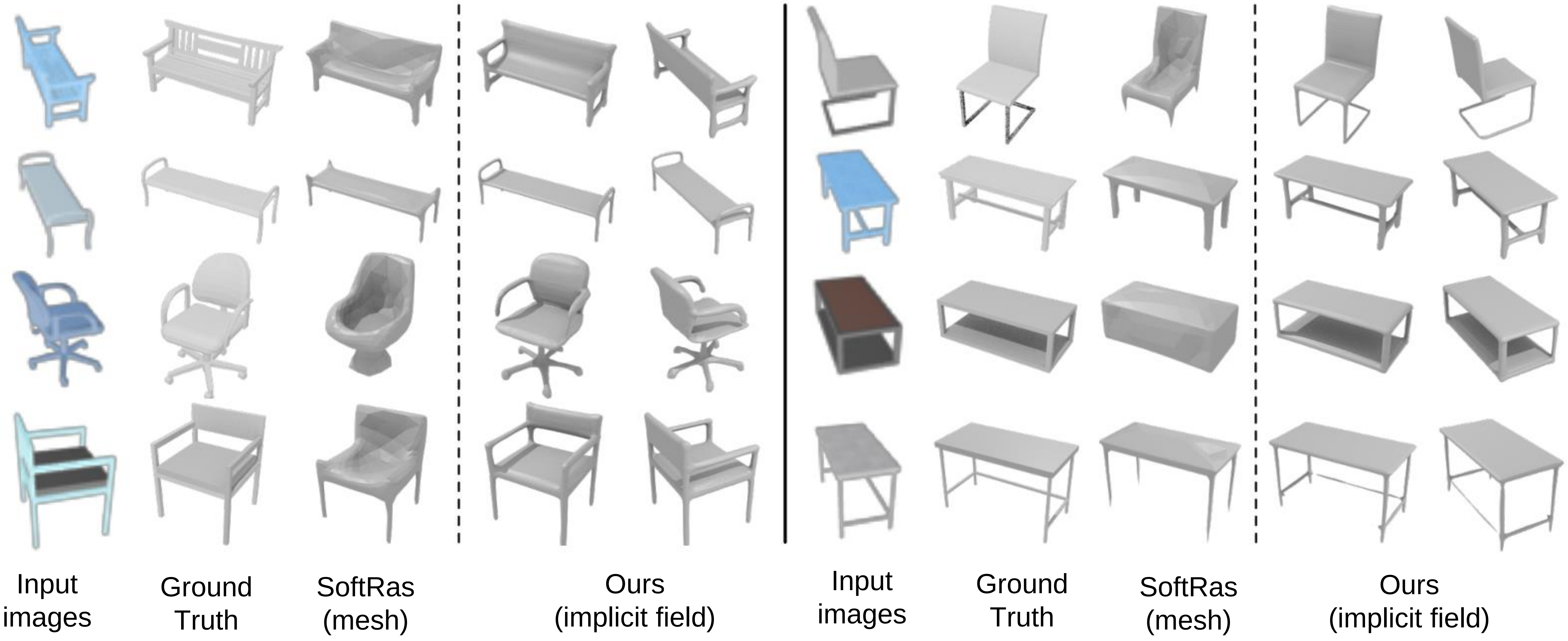


Results

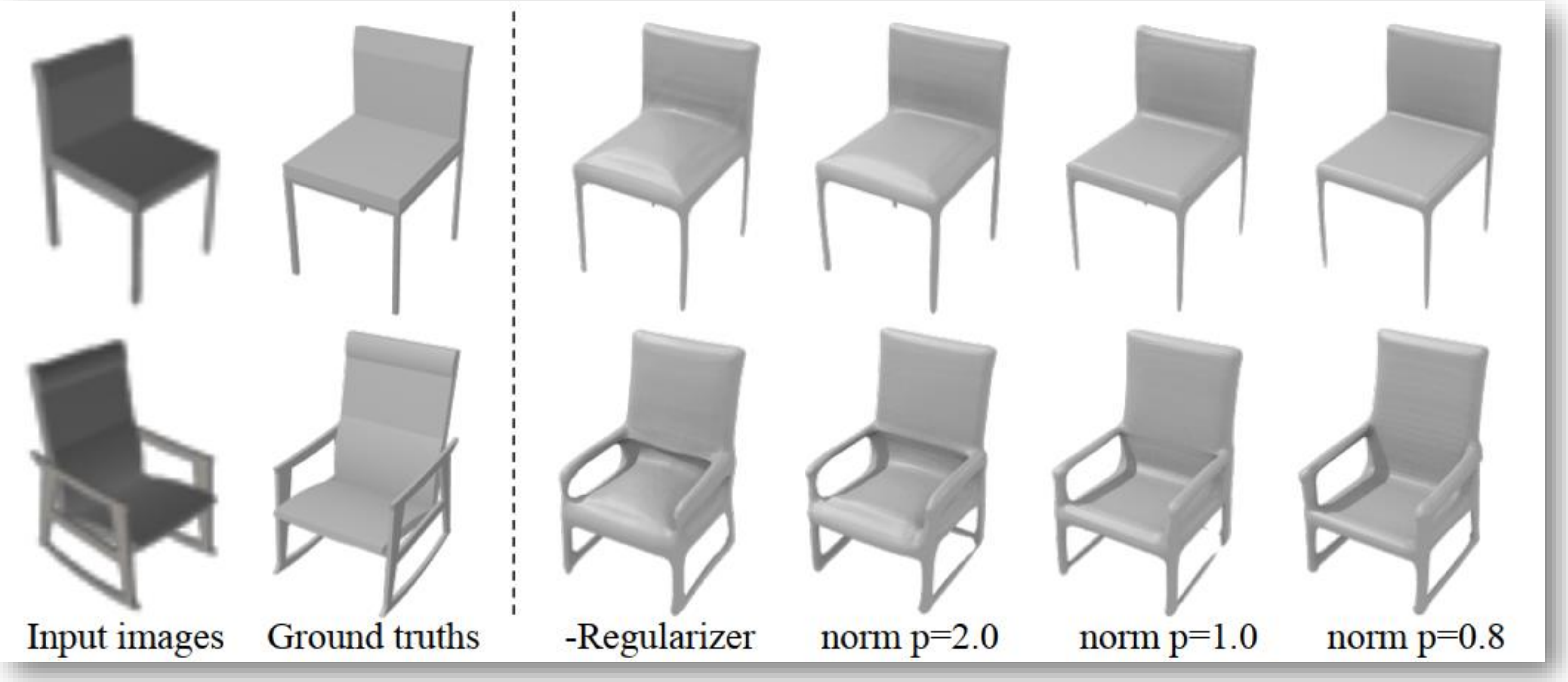
Comparisons of 3D IoU with Other Unsupervised Methods

Category	Airplane	Bench	Table	Car	Chair	Boat	Mean
PTN [4]	0.5564	0.4875	0.4938	0.7123	0.4494	0.5507	0.5417
NMR [1]	0.6172	0.4998	0.4829	0.7095	0.4990	0.5953	0.5673
SoftRas [2]	0.6419	0.5080	0.4487	0.7697	0.5270	**0.6145**	0.5850
Ours	**0.6510**	**0.5360**	**0.5150**	**0.7820**	**0.5480**	0.6080	**0.6067**

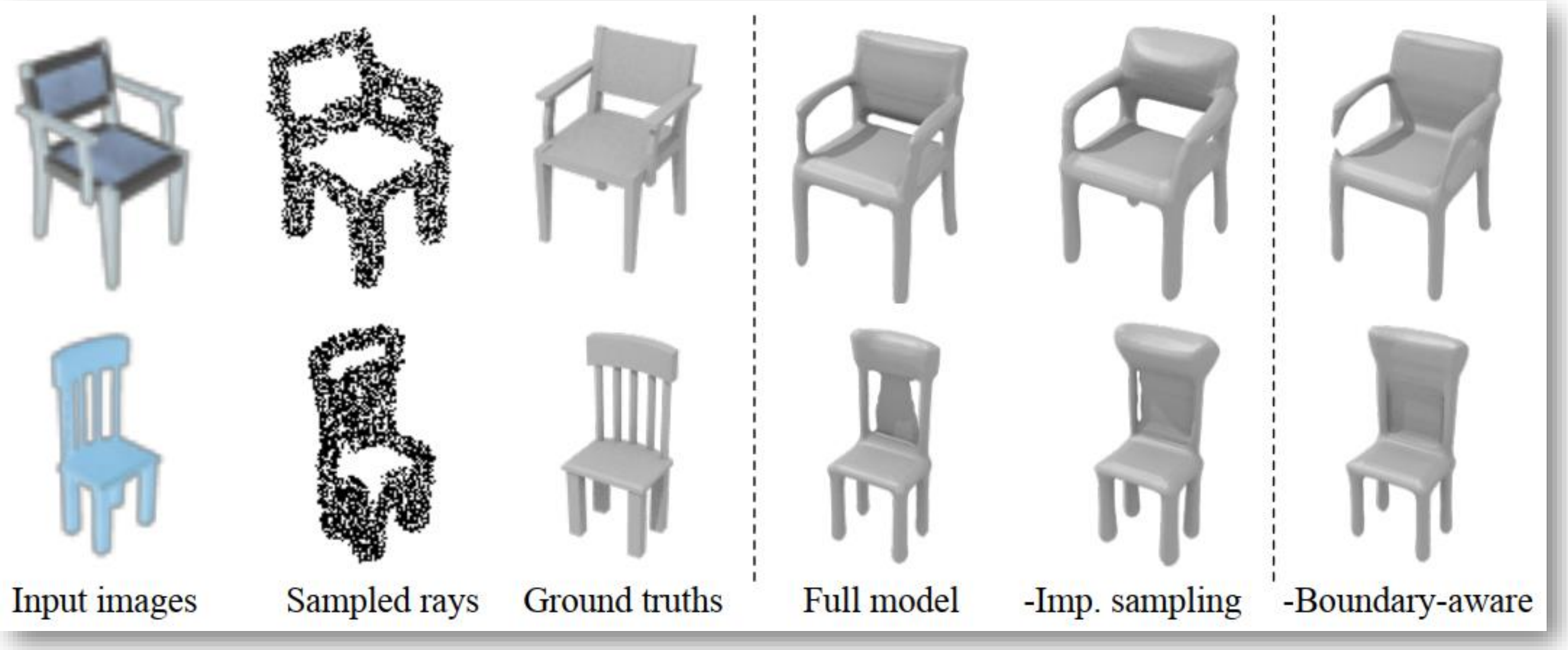
Qualitative comparisons with mesh-based approach in term of modeling capability



Ablation Analysis



Qualitative evaluations of geometric regularization by using different configurations



Qualitative analysis of importance sampling and boundary-aware assignment for single-view reconstruction

Conclusions

Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning

- A new differentiable rendering framework that can directly render a given mesh in a fully differentiable manner
- Formulate the conventional discrete operations – rasterization and z-buffering, as differentiable probabilistic processes
- Can flow gradients from image to unseen vertices and the z coordinates of the mesh triangles
- Applied to 3D unsupervised single-view reconstruction and image-based shape fitting

Learning to Infer Implicit Surfaces without 3D Supervision

- A new framework that enables learning of implicit surfaces for shape modeling without 3D supervision
- A novel field probing approach based on anchor points and probing rays that efficiently correlates the implicit field and the observed images
- An efficient point and ray sampling method for implicit surface generation from image-based supervision
- A general formulation of geometric regularization that can constrain the geometric properties of a continuous implicit surface

We have open sourced the code of SoftRas!
https://github.com/ShichenLiu/SoftRas

PYTORCH

Thank you!