CSC2457 3D & Geometric Deep Learning

Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision

Michael Niemeyer Lars Mescheder Michael Oechsle Andreas Geiger

Feb 16th, 2021

Presenter: Sara Sabour

Instructor: Animesh Garg

UNIVERSITY OF TORONTO

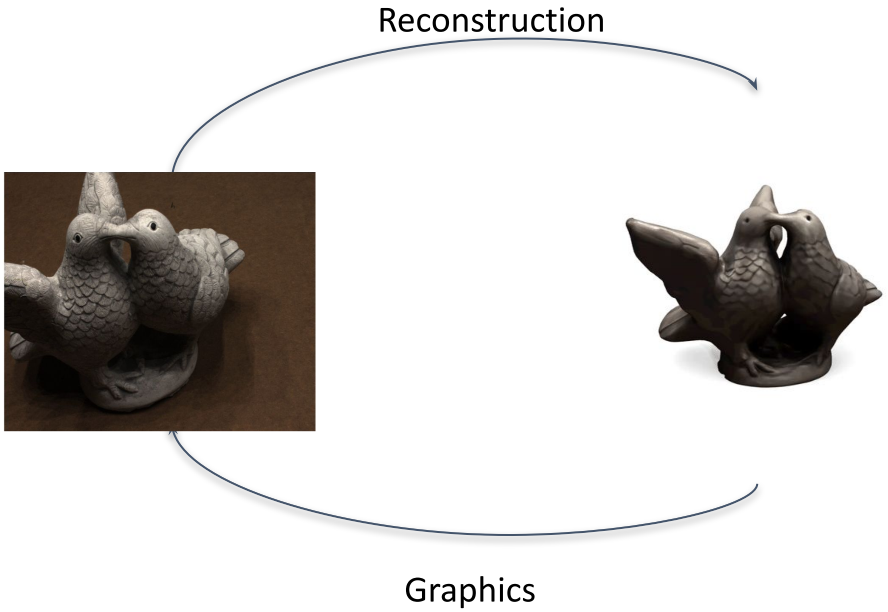
Main Problem

3D reconstruction without 3D supervision

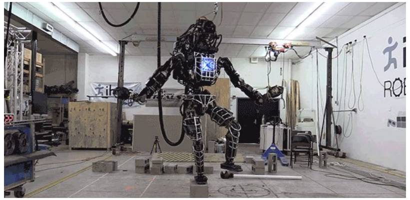
- Generate a full 3D model
    - Implicit function
- Train only with single RGB images

    + Camera intrinsics and extrinsics

    + Object masks



Motivation



Why do we want a 3D model as opposed to just rendering?

Real world is 3D -> interaction requires a model

- Robotic applications
- Autonomous driving

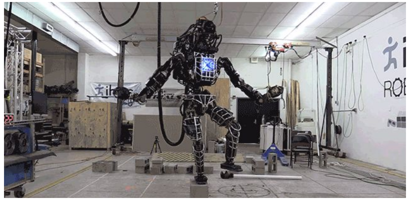
Motivation

- Why do we need a 3D model as opposed to just rendering?

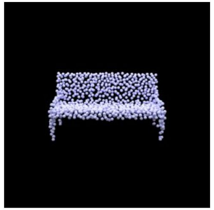
    Real world is 3D -> interaction requires a model

    - Robotic applications
    - Autonomous driving

- Why Implicit representations?
    - Infinite resolution with fixed footprint
    - Perfect surface rendering without template



Voxel (Choy et al. 2016)



Point Cloud (Fan et al. 2017)

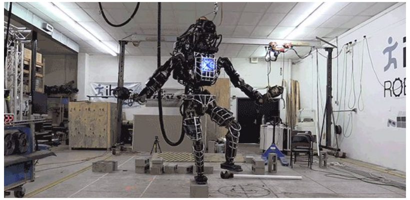
Mesh (Groueix et al. 2017

Motivation

- Why do we need a 3D model as opposed to just rendering?

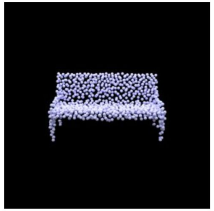
    Real world is 3D -> interaction requires a model

    - Robotic applications
    - Autonomous driving



- Why Implicit representations?
    - Infinite resolution with fixed footprint
    - Perfect surface rendering without template

Voxel (Choy et al. 2016)



Point Cloud (Fan et al. 2017)

Mesh (Groueix et al. 2017)

- Why unsupervised?
    - Real world 3D supervision is not easy to gather

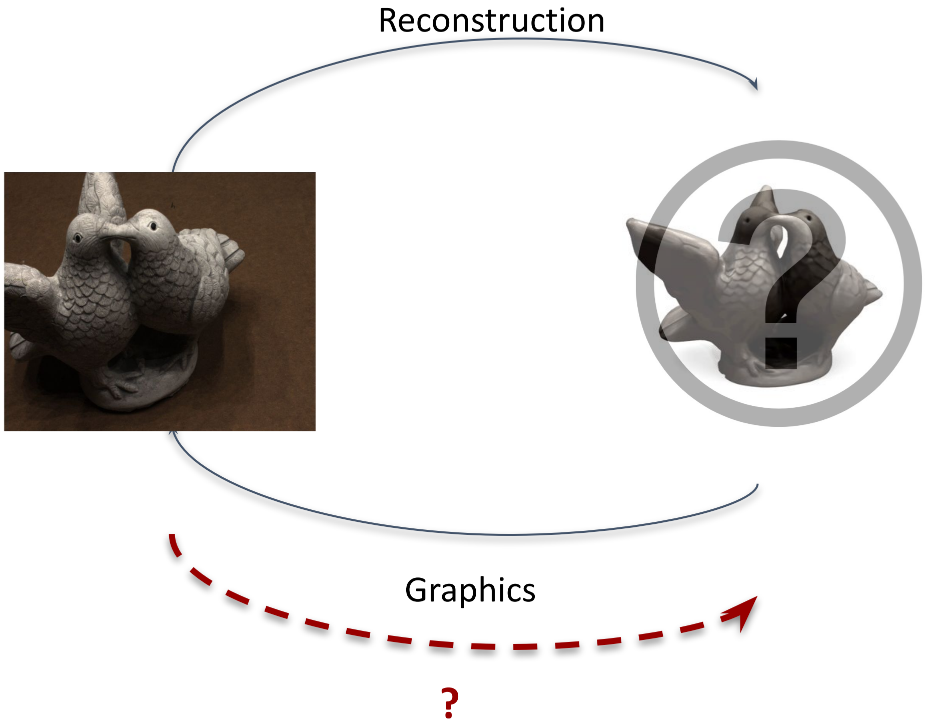
Why is it hard and not already solved?

Unsupervised Implicit Model:

- Requires good regularizers.
- Requires rendering back to image.
- Previous work mainly focused on shape and ignored texture.

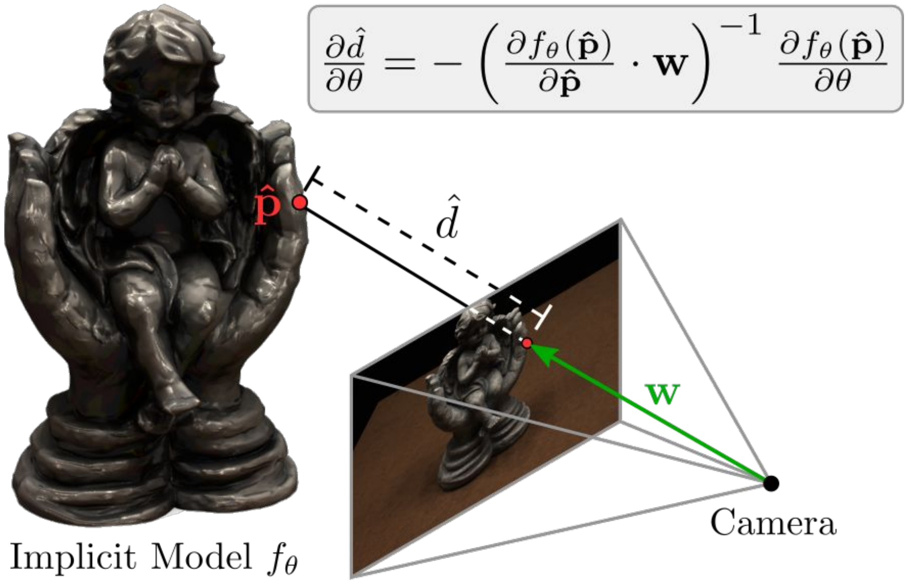
Unsupervised Implicit Model:

- Gradient through an implicit function rendering (ray tracing) was costly, infeasible, inaccurate.



Contributions

1. Novelty (method)
    a. They propose an analytic derivation for the gradient of the implicit function rendering.
    b. Their model incorporates texture as well as shape.
2. Results
    a. SOTA on unsupervised Shapenet.
    b. Realistic dataset results.

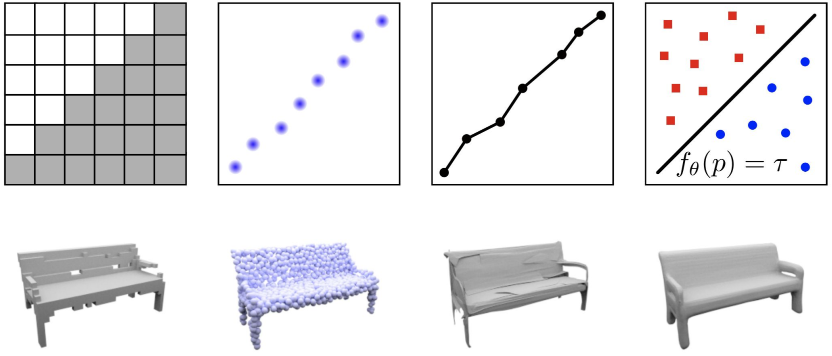


General Background: Implicit Function

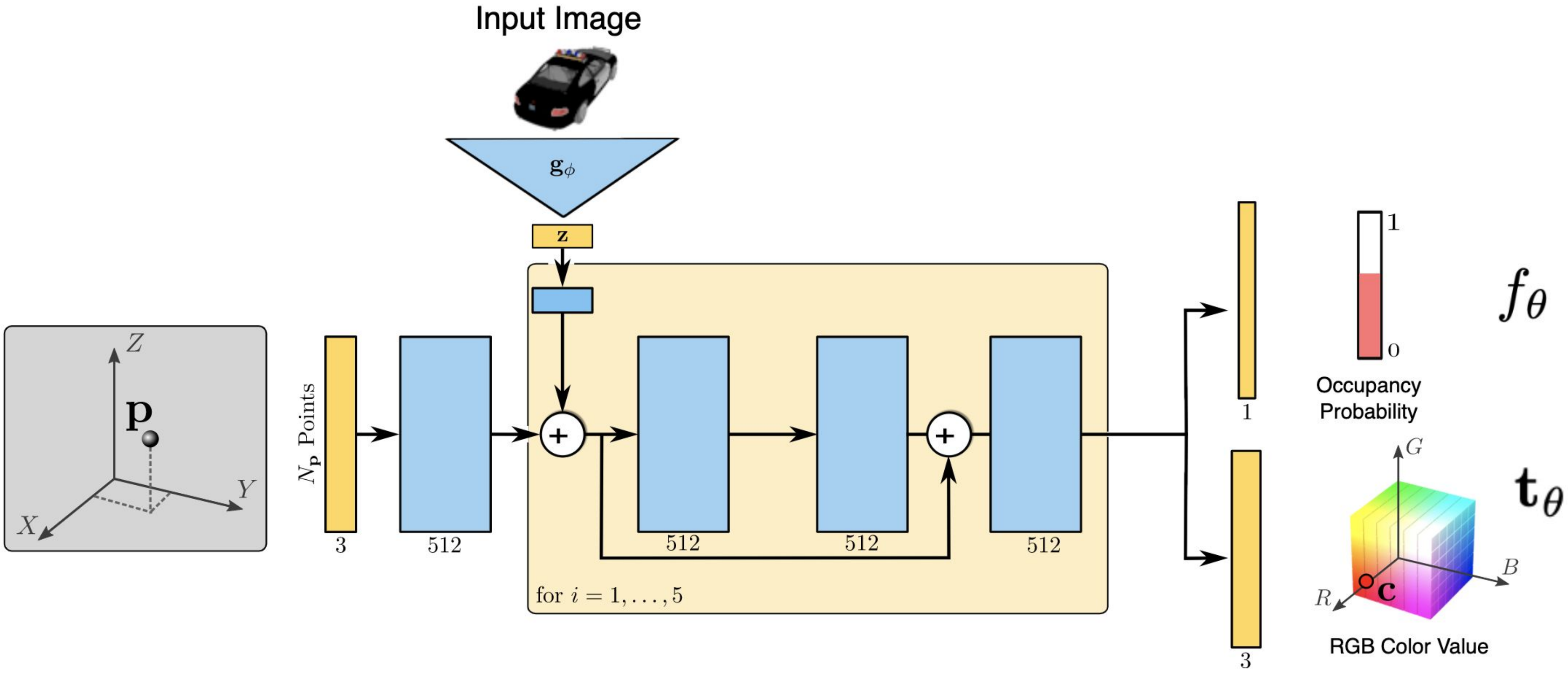
The Surface is modeled as the Root of a parametric function.

$$f_\theta : \mathbb{R}^3 \times \mathcal{Z} \rightarrow [0, 1]$$

$$\mathbf{t}_\theta : \mathbb{R}^3 \times \mathcal{Z} \rightarrow \mathbb{R}^3$$



Implicit Function Architecture



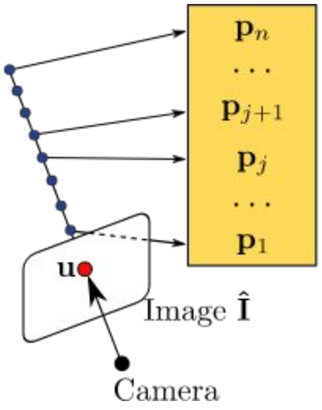
Rendering with an Implicit function

Why f_θ **is a 3D model?**

Given f_θ we can render it from any camera viewpoint.

How?

Ray tracing!



Take **n** equidistant candidate points that would be projected to **u** for this camera angle.

$$\mathbf{r}(d) = \mathbf{r}_0 + d\mathbf{w}$$

$$\mathbf{p}_j^{\text{ray}} = \mathbf{r}(j\Delta s + s_0)$$

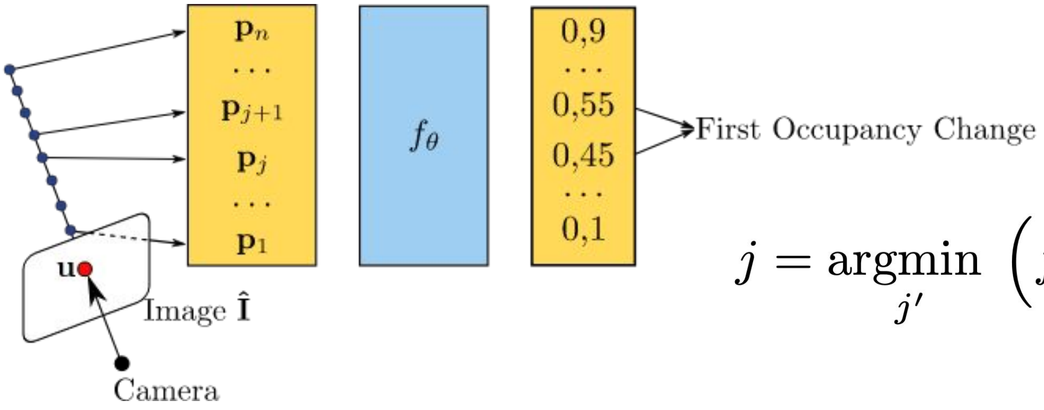
Rendering with an Implicit function

Why f_θ is a 3D model?

Given f_θ we can render it from any camera viewpoint.

How?

Ray tracing!



$$j = \underset{j'}{\operatorname{argmin}} \left(f_\theta(\mathbf{p}^{\text{ray}}_{j'+1}) \geq \tau > f_\theta(\mathbf{p}^{\text{ray}}_{j'}) \right)$$

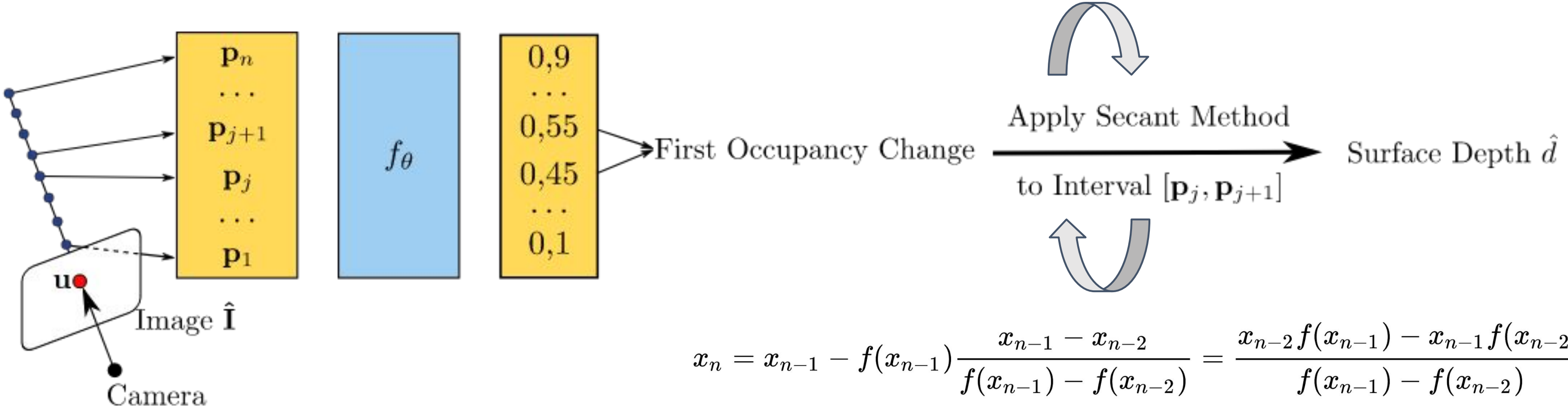
Rendering with an Implicit function

Why f_θ is a 3D model?

Given f_θ we can render it from any camera viewpoint.

How?

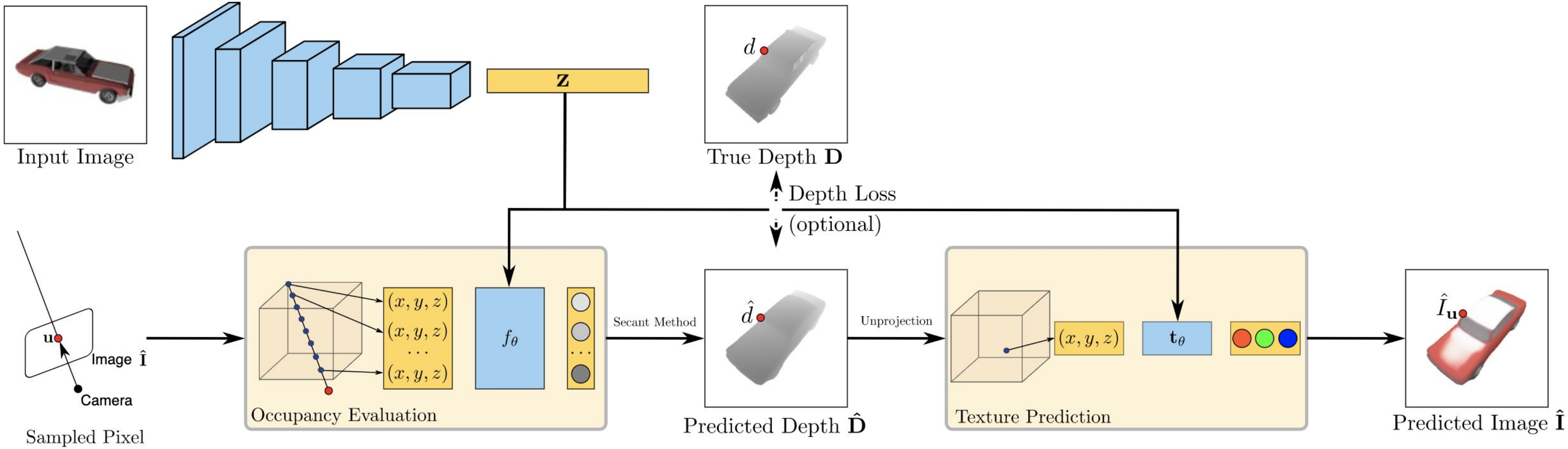
Ray tracing!



$$x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})} = \frac{x_{n-2} f(x_{n-1}) - x_{n-1} f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})}$$

Method overview

How to use implicit rendering for image autoencoding?

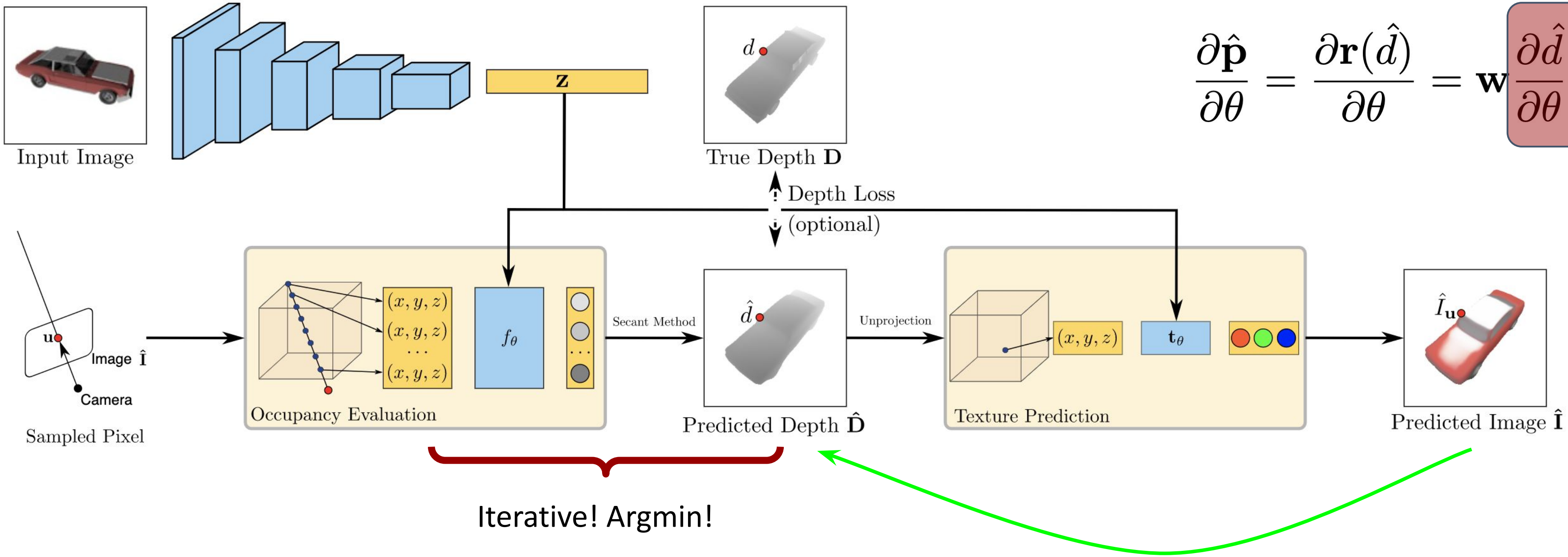


What is the issue?

How to backpropagate efficiently?!

$$\frac{\partial \hat{\mathbf{I}}_{\mathbf{u}}}{\partial \theta} = \frac{\partial \mathbf{t}_\theta(\hat{\mathbf{p}})}{\partial \theta} + \frac{\partial \mathbf{t}_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \theta}$$

$$\frac{\partial \hat{\mathbf{p}}}{\partial \theta} = \frac{\partial \mathbf{r}(\hat{d})}{\partial \theta} = \mathbf{w} \frac{\partial \hat{d}}{\partial \theta}$$



Main contribution: Analytical Derivation

Observation: Gradients only needs to be calculated at the surface: $f_\theta(\hat{\mathbf{p}}) = \tau$

Main contribution: Analytical Derivation

$$f_\theta(\hat{\mathbf{p}}) = \tau$$

Chain Rule!

$$\frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta} + \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \theta} = 0$$

Chain Rule!

$$\frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta} + \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \mathbf{w} \frac{\partial \hat{d}}{\partial \theta} = 0$$

Main contribution: Analytical Derivation

$$f_\theta(\hat{\mathbf{p}}) = \tau$$

Chain Rule!

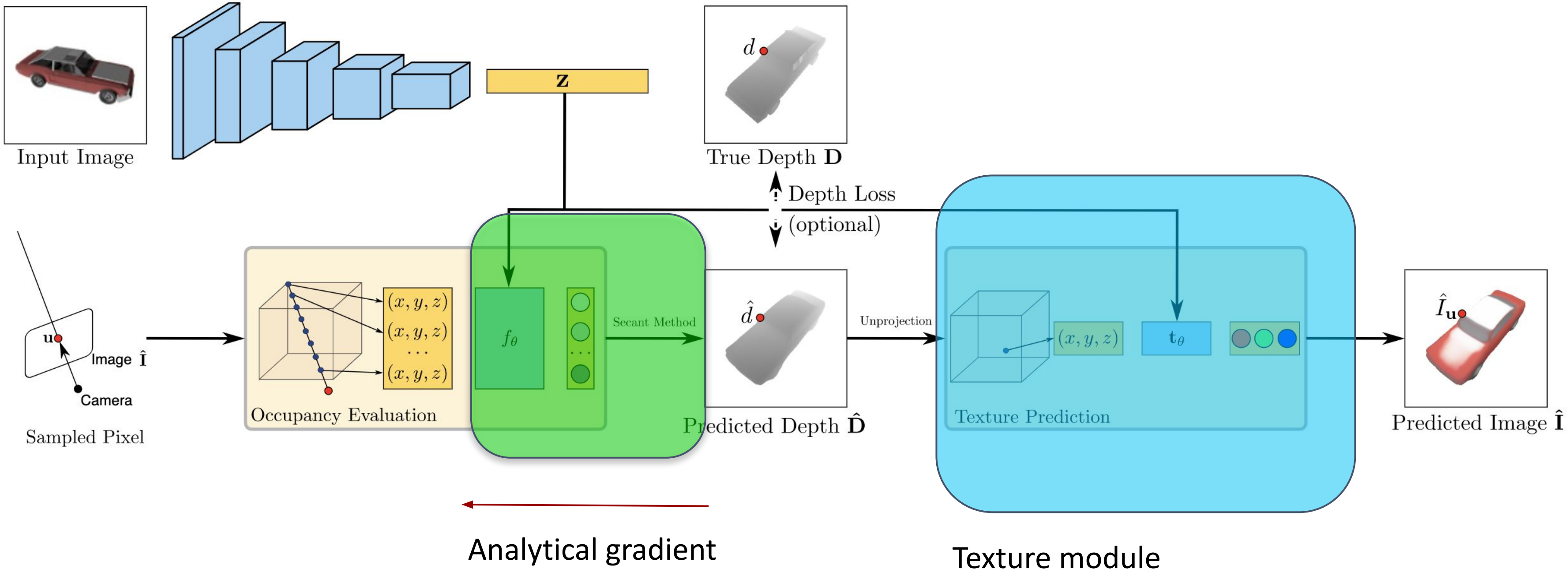
$$\frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta} + \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \theta} = 0$$

Chain Rule!

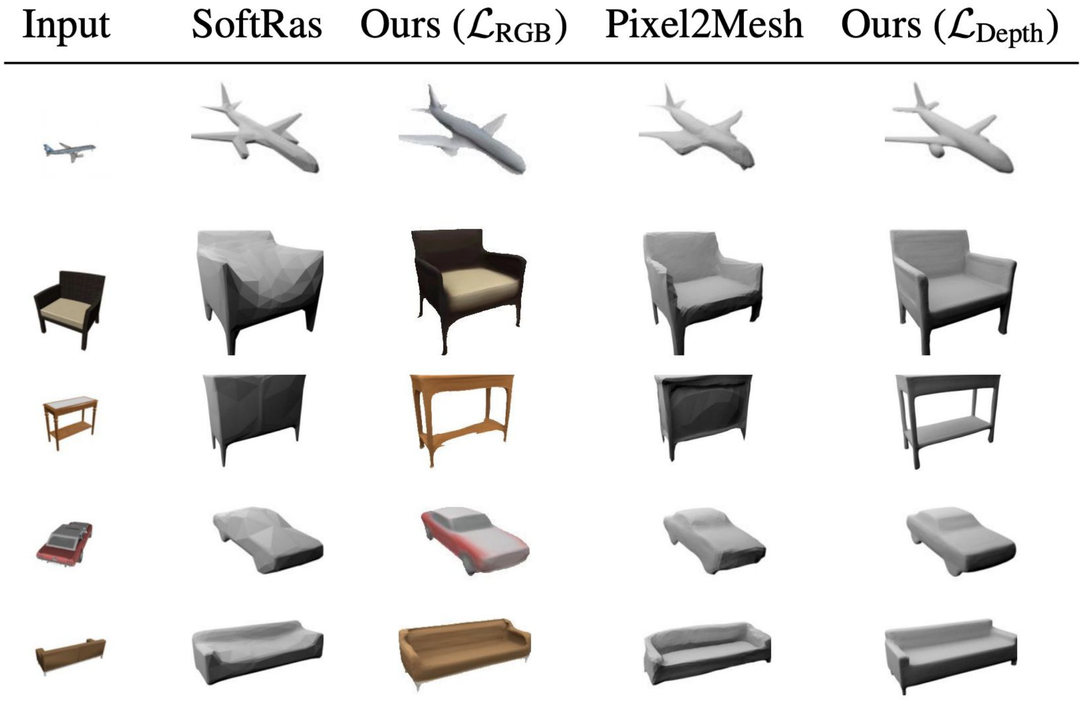
$$\frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta} + \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \mathbf{w} \frac{\partial \hat{d}}{\partial \theta} = 0$$

$$\frac{\partial \hat{d}}{\partial \theta} = -\left(\frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \mathbf{w} \right)^{-1} \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta}$$

Main contribution



Experimental Results: qualitative ShapeNet



Experimental Results: Chamfer Distance on ShapeNet

category	2D Supervision			2.5D Supervision		3D Supervision		
	DRC (Mask) [79]	SoftRas [44]	Ours ($\mathcal{L}_{\text{RGB}}$)	DRC (Depth) [79]	Ours ($\mathcal{L}_{\text{Depth}}$)	3D R2N2 [13]	ONet [48]	Pixel2Mesh [80]
airplane	0.659	**0.149**	0.190	0.377	**0.143**	0.215	**0.151**	0.183
bench	-	0.241	**0.210**	-	**0.165**	0.210	**0.171**	0.191
cabinet	-	0.231	**0.220**	-	**0.183**	0.246	**0.189**	0.194
car	0.340	0.221	**0.196**	0.316	**0.179**	0.250	0.181	**0.154**
chair	0.660	0.338	**0.264**	0.510	**0.226**	0.282	**0.224**	0.259
display	-	0.284	**0.255**	-	**0.246**	0.323	0.275	**0.231**
lamp	-	**0.381**	0.413	-	**0.362**	0.566	0.380	**0.309**
loudspeaker	-	0.320	**0.289**	-	**0.295**	0.333	0.290	**0.284**
rifle	-	**0.155**	0.175	-	**0.143**	0.199	0.160	**0.151**
sofa	-	0.407	**0.224**	-	**0.221**	0.264	0.217	**0.211**
table	-	0.374	**0.280**	-	**0.180**	0.247	**0.185**	0.215
telephone	-	**0.131**	0.148	-	**0.130**	0.221	0.155	**0.145**
vessel	-	**0.233**	0.245	-	**0.206**	0.248	0.220	**0.201**
mean	0.553	0.266	**0.239**	0.401	**0.206**	0.277	0.215	**0.210**

Experimental Results: Qualitative DETU

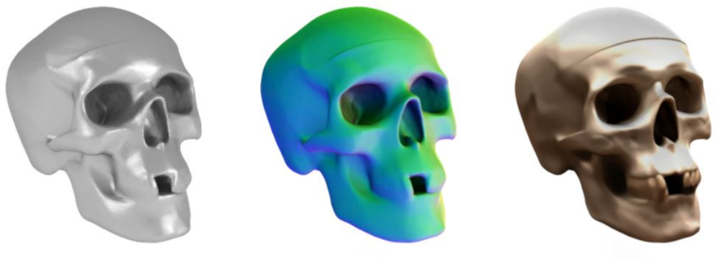
Compared with Poisson surface reconstruction (sPSR) on mesh based approaches with a trimming of 5 or 7.

(a) Colmap 5 (b) Colmap 7 (c) Ours

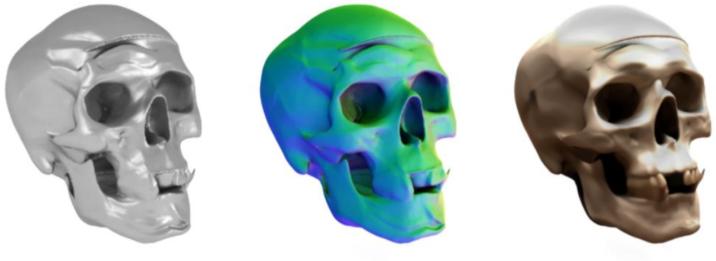
Experimental Results: Quantitative DETU

	Trim Param.	Chamfer-L_1
Tola [78] + sPSR	0	1.826
Furu [18] + sPSR	0	1.517
Colmap [67] + sPSR	0	**1.303**
Camp [9] + sPSR	0	1.441
Tola [78] + sPSR	5	1.399
Furu [18] + sPSR	5	1.311
Colmap [67] + sPSR	5	**1.091**
Camp [9] + sPSR	5	1.331
Tola [78] + sPSR	7	0.910
Furu [18] + sPSR	7	0.839
Colmap [67] + sPSR	7	**0.733**
Camp [9] + sPSR	7	1.092
Ours ($\mathcal{L}_{\text{RGB}}$)	-	0.907
Ours ($\mathcal{L}_{\text{RGB}} + \mathcal{L}_{\text{Depth}}$)	-	**0.782**

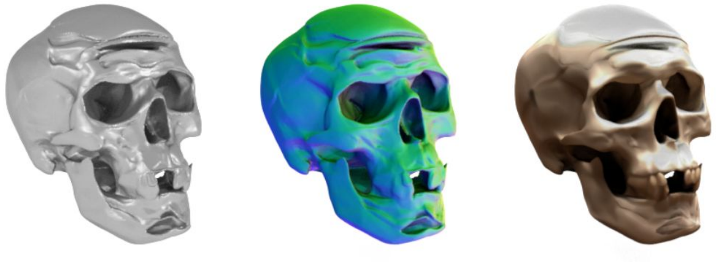
Effect of adding a surface smoothness loss



(a) Our model ($\mathcal{L}_{\text{RGB}}$) with $\lambda_2 = 1$



(b) Our model ($\mathcal{L}_{\text{RGB}}$) with $\lambda_2 = 0.1$



(c) Our model ($\mathcal{L}_{\text{RGB}}$) with $\lambda_2 = 0.$

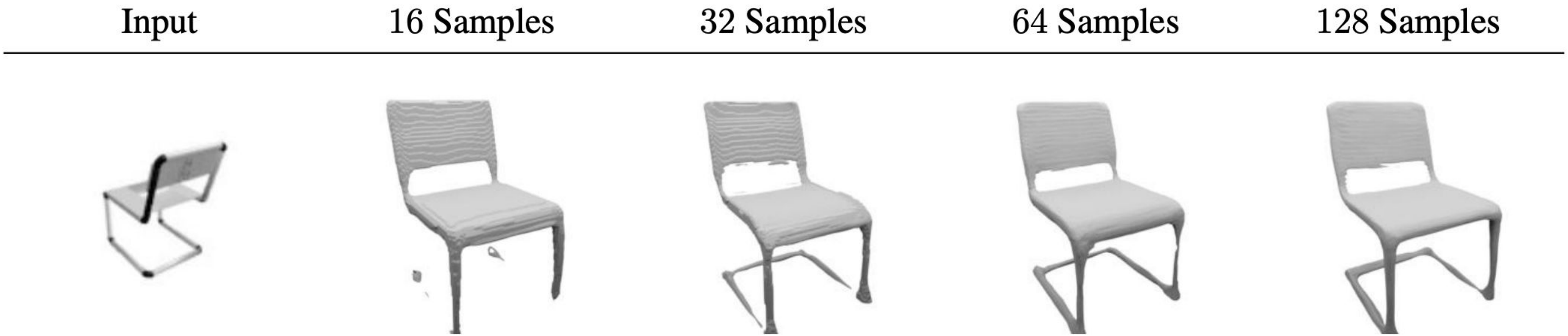
Effect of adding a supervised Depth signal

(a) Colmap [17] + sPSR

(b) Ours ($\mathcal{L}_{\mathrm{RGB}}$)

(c) Ours ($\mathcal{L}_{\mathrm{RGB}} + \mathcal{L}_{\mathrm{Depth}}$)

Effect of number of samples



Discussion of results

- On both of datasets they **outperform previous unsupervised methods**.
    - So implicit functions are superior models to Voxel, point clouds and Mesh based models.

- They do not have any **real world scene examples** from Autonomous Driving or Game Engines that simulate those.
    - So they have not bridged the gap between real world (background, multiple objects) and synthetic data yet.
    - It is only object centric reconstruction.

Critique / Limitations / Open Issues

- Relying on **camera intrinsics and object masks** could be as unrealistic as having 3D model or depth maps.
  - Add canonicalization, unsupervised alignment?
  - Add compositionality?

- They fail on narrow/sharp geometrics.
  - Smarter ray tracing? A prior? Part decomposition?

Contributions (Recap)

- Successful 3D unsupervised reconstruction
- Scalable to real world interactive domains (only object level)
- Better Chamfer Distance and scalability to real world images
- Key insight: integrate texture, use analytical derivatives