Learning 3D Reconstruction in Function Space

Andreas Geiger

Autonomous Vision Group
MPI for Intelligent Systems and University of Tübingen

June 15, 2020

University of Tübingen
MPI for Intelligent Systems
Autonomous Vision Group

Collaborators

Lars Mescheder

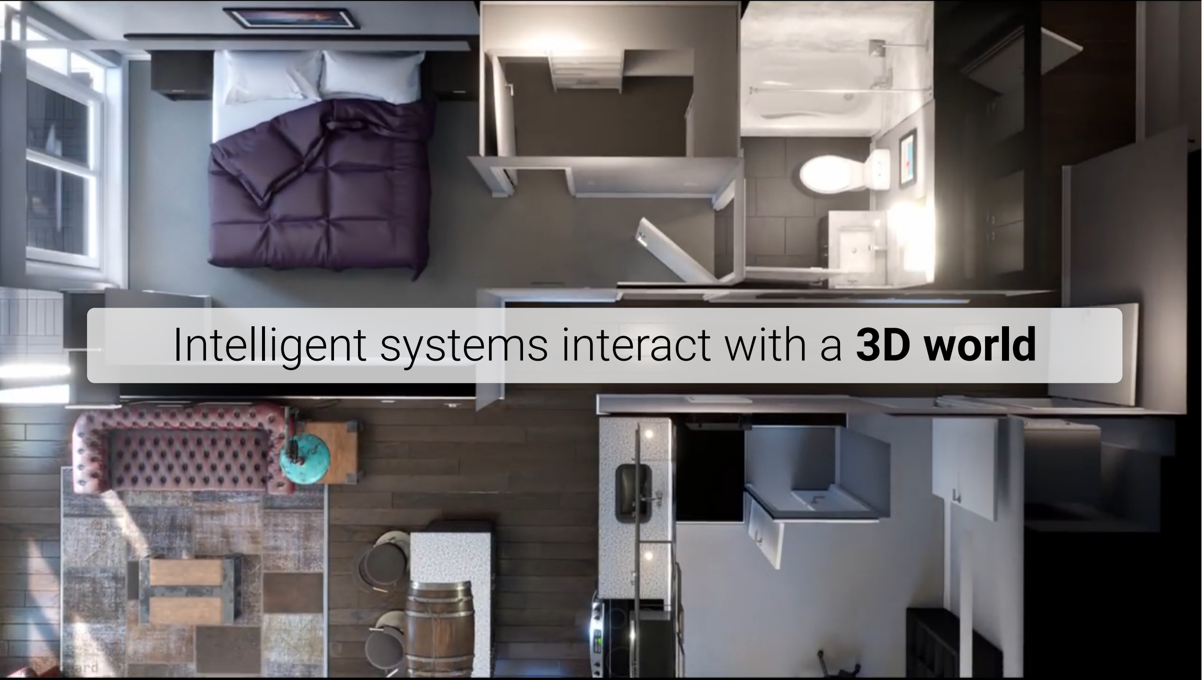
Michael Niemeyer

Michael Oechsle

Sebastian Nowozin

Andreas Geiger

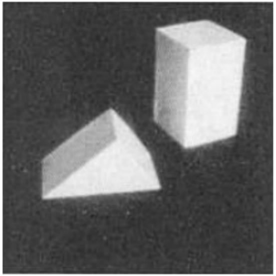
Our goal is to make **intelligent systems** more autonomous, robust and safe

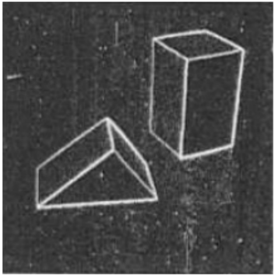


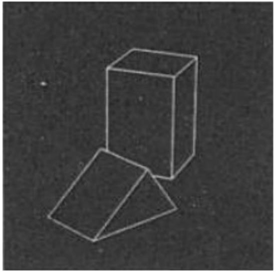
Intelligent systems interact with a **3D world**

3D reconstruction is a hard problem

1963: Blocks World



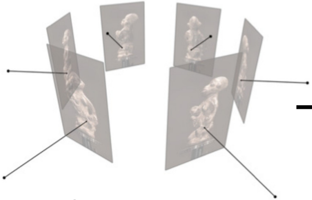




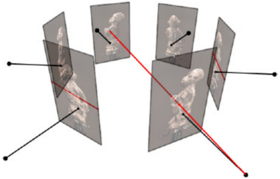
Larry Roberts: Machine Perception of Three-Dimensional Solids. PhD Thesis, MIT, 1963.

Traditional 3D Reconstruction Pipeline

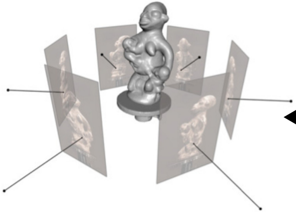
Input Images



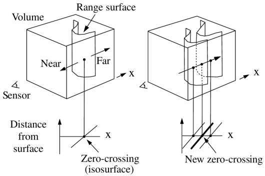
Camera Poses



Dense Correspondences



3D Reconstruction



Depth Map Fusion

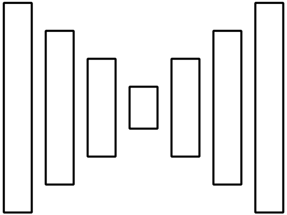
Depth Maps

Humans recognize 3D from a **single** 2D image

Can we **learn** to infer 3D **from a 2D image**?

3D Reconstruction from a 2D Image

Input Images



Neural Network

3D Reconstruction

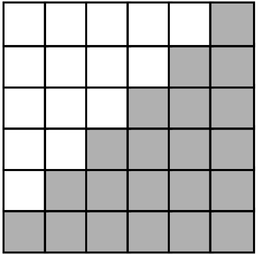
What is a good **output** representation?

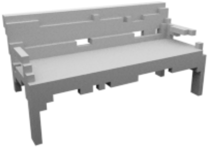
3D Representations

Voxels:

- **Discretization** of 3D space into grid
- Easy to process with neural networks
- Cubic memory $O(n^3) \Rightarrow$ limited resolution
- Manhattan world bias

[Maturana et al., IROS 2015]



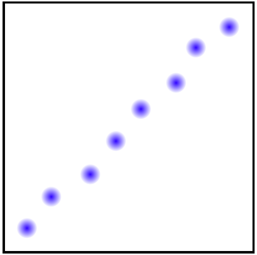


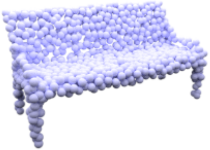
3D Representations

Points:

- **Discretization** of surface into 3D points
- Does not model connectivity / topology
- Limited number of points
- Global shape description

[Fan et al., CVPR 2017]



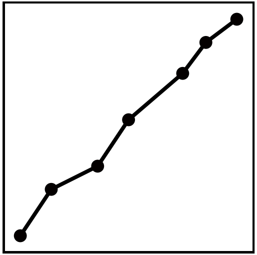


3D Representations

Meshes:

- **Discretization** into vertices and faces
- Limited number of vertices / granularity
- Requires class-specific template – or –
- Leads to self-intersections

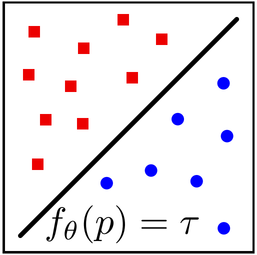
[Groueix et al., CVPR 2018]



3D Representations

This work:

- Implicit representation $\Rightarrow$ **No discretization**
- Arbitrary topology & resolution
- Low memory footprint
- Not restricted to specific class



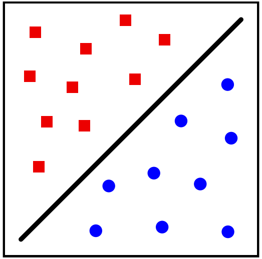
Occupancy Networks

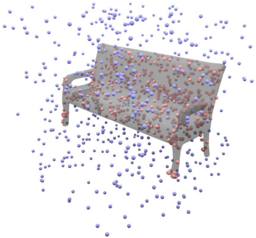
Key Idea:

- Do not represent 3D shape explicitly
- Instead, consider surface **implicitly** as **decision boundary** of a non-linear classifier:

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

3D Location — Condition (eg, Image) — Occupancy Probability





Concurrent work:

- DeepSDF [Park et al., CVPR 2019]
- IM-NET [Chen et al., CVPR 2019]

Training Objective

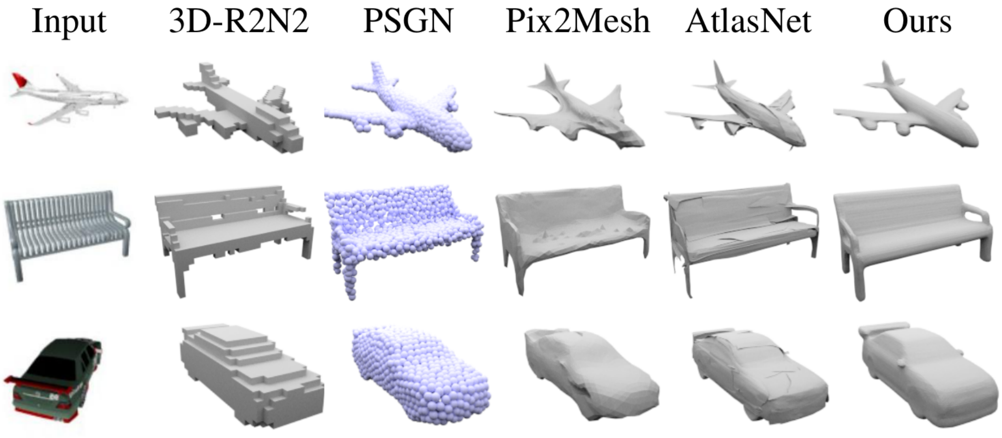
Occupancy Network:Variational Occupancy Encoder:

$$\mathcal{L}(\theta, \psi) = \sum_{j=1}^{K} \text{BCE}(f_\theta(p_{ij}, z_i), o_{ij}) + KL\left[q_\psi(z|(p_{ij}, o_{ij})_{j=1:K}) \,\|\, p_0(z)\right]$$

- K: Randomly sampled 3D points ($K = 2048$)
- BCE: Cross-entropy loss
- q_ψ: Encoder

Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

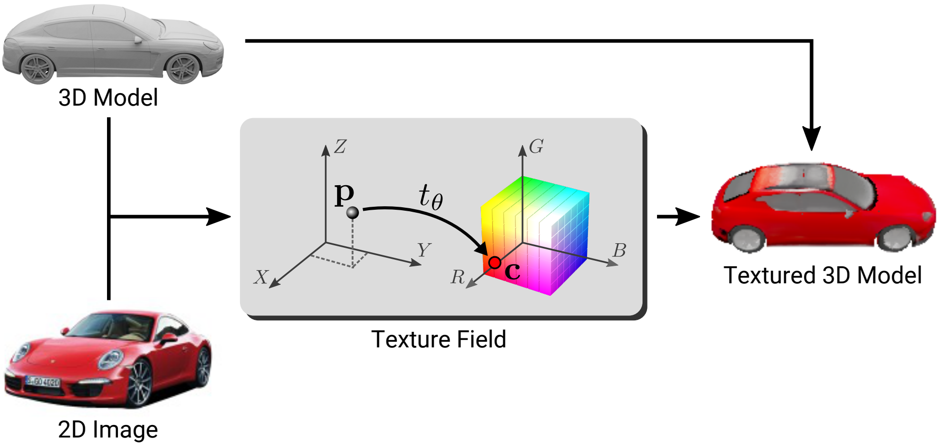
Results



Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

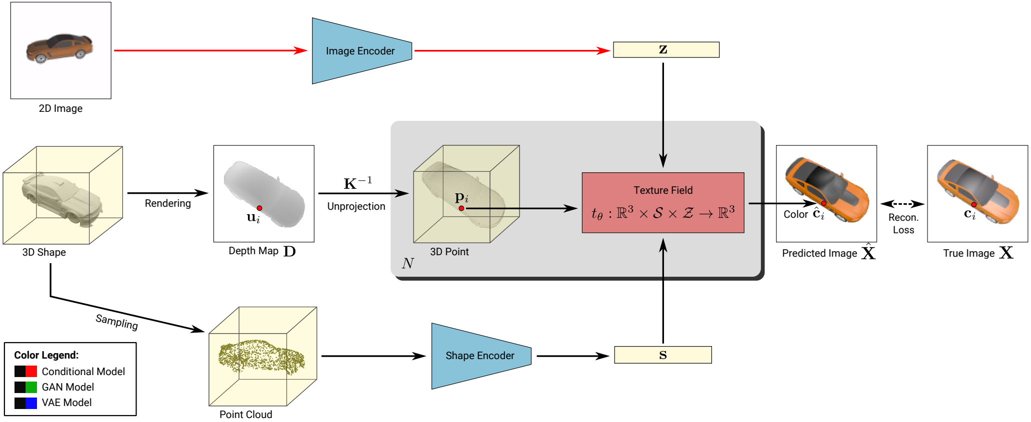
Can we also learn about object **appearance**?

Texture Fields



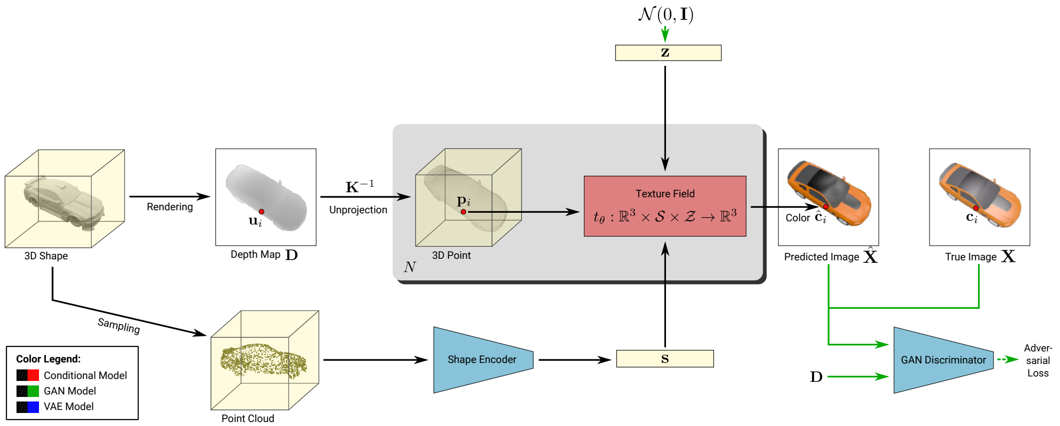
Oechsle, Mescheder, Niemeyer, Strauss and Geiger: Texture Fields: Learning Texture Representations in Function Space. ICCV, 2019.

Texture Fields



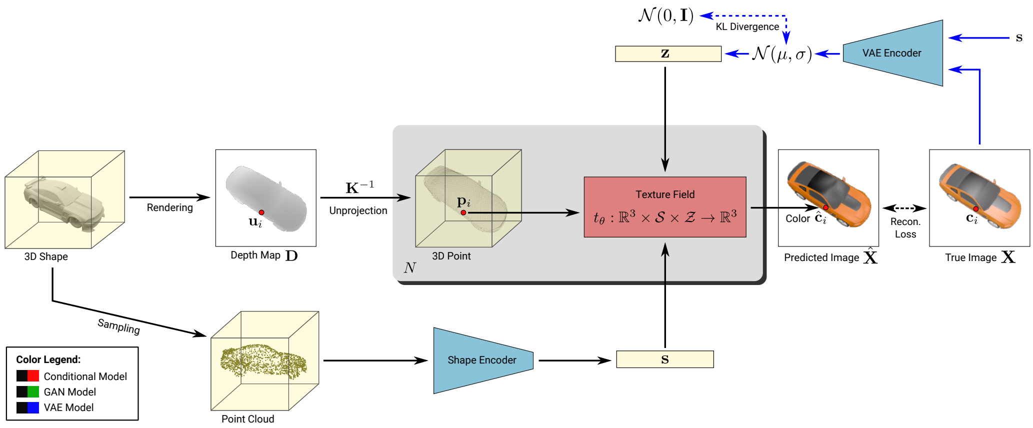
Oechsle, Mescheder, Niemeyer, Strauss and Geiger: Texture Fields: Learning Texture Representations in Function Space. ICCV, 2019.

Texture Fields



Oechsle, Mescheder, Niemeyer, Strauss and Geiger: Texture Fields: Learning Texture Representations in Function Space. ICCV, 2019.

Texture Fields



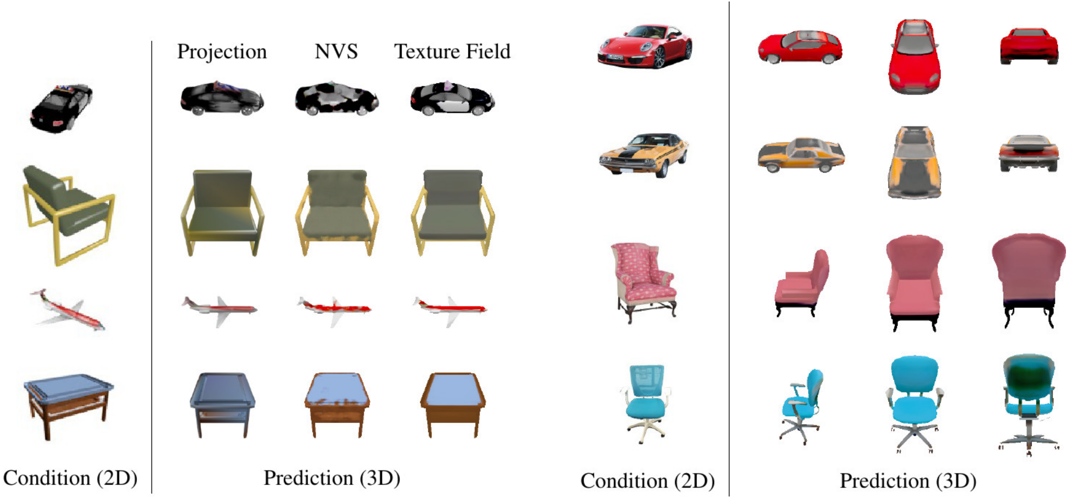
Oechsle, Mescheder, Niemeyer, Strauss and Geiger: Texture Fields: Learning Texture Representations in Function Space. ICCV, 2019.

Representation Power

- Ground truth vs. Texture Field vs. Voxelization

Oechsle, Mescheder, Niemeyer, Strauss and Geiger: Texture Fields: Learning Texture Representations in Function Space. ICCV, 2019.

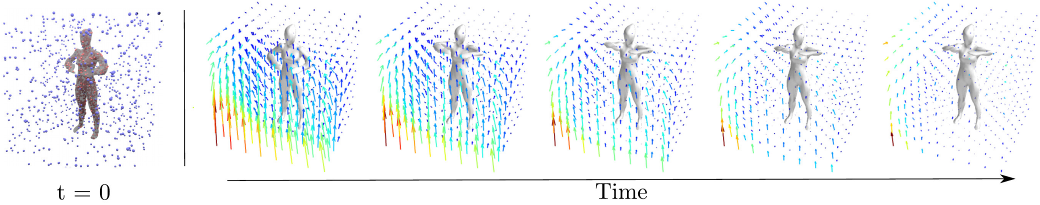
Results



Oechsle, Mescheder, Niemeyer, Strauss and Geiger: Texture Fields: Learning Texture Representations in Function Space. ICCV, 2019.

What about object **motion**?

Occupancy Flow

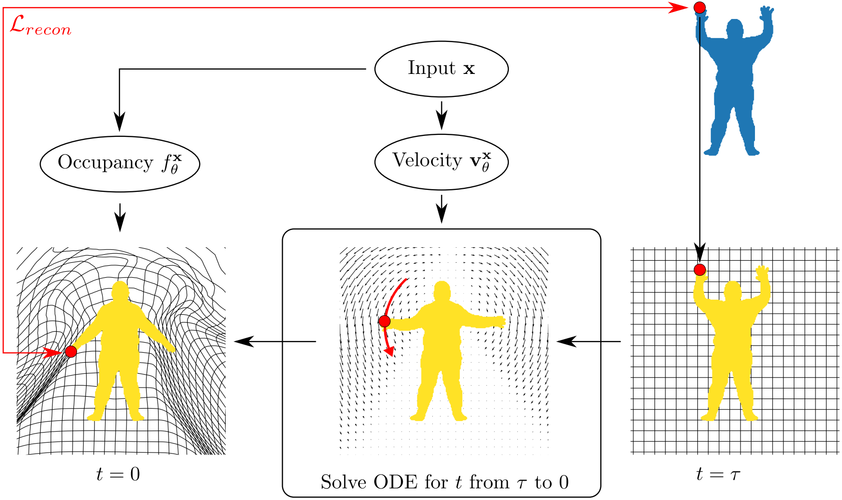


- Extending Occupancy Networks to 4D is hard (curse of dimensionality)
- Represent shape at $t = 0$ using a 3D Occupancy Network
- Represent motion by temporally and spatially continuous vector field
- Relationship between 3D trajectory $\mathbf{s}$ and velocity $\mathbf{v}$ given by (differentiable) ODE:

$$\frac{\partial \mathbf{s}(t)}{\partial t} = \mathbf{v}(\mathbf{s}(t), t)$$

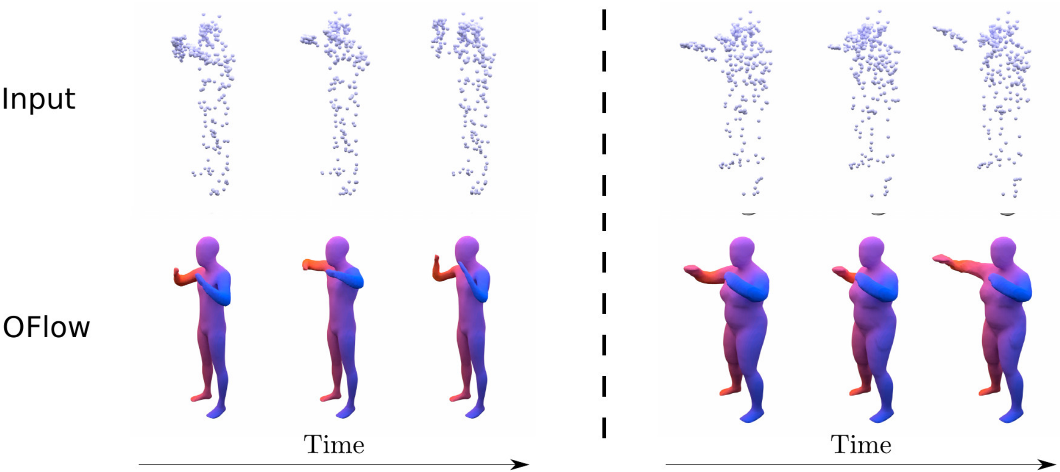
Niemeyer, Mescheder, Oechsle and Geiger: Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics. ICCV, 2019.

Occupancy Flow



Niemeyer, Mescheder, Oechsle and Geiger: Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics. ICCV, 2019.

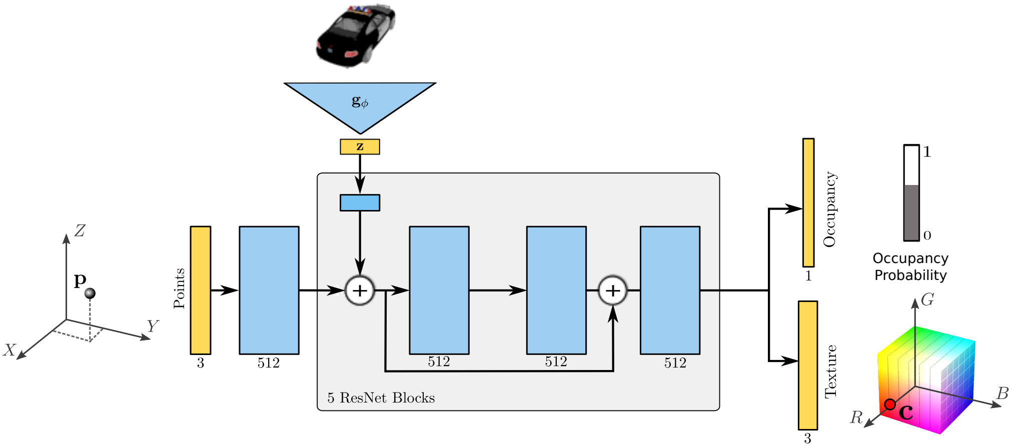
Results



- ► No correspondences needed ⇒ implicitly established by our model!

Can we **learn** implicit representations **from images**?

Architecture



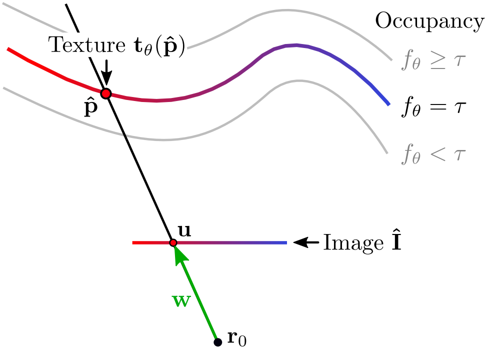
Forward Pass

(Rendering)

Differentiable Volumetric Rendering

Forward Pass:

- For all pixels $\mathbf{u}$
- Find surface point $\hat{\mathbf{p}}$ along ray $\mathbf{w}$ via ray marching and root finding
- Evaluate texture field $\mathbf{t}_\theta(\hat{\mathbf{p}})$ at $\hat{\mathbf{p}}$
- Insert color $\mathbf{t}_\theta(\hat{\mathbf{p}})$ at pixel $\mathbf{u}$



Niemeyer, Mescheder, Oechsle and Geiger: Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. CVPR, 2020.

Backward Pass

(Differentiation)

Differentiable Volumetric Rendering

Backward Pass:

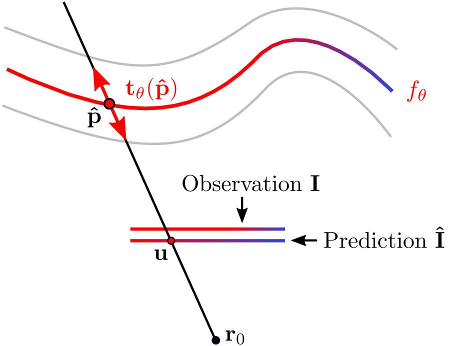
- Image Observation $\mathbf{I}$
- Loss $\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}) = \sum_{\mathbf{u}} \|\hat{\mathbf{I}}_{\mathbf{u}} - \mathbf{I}_{\mathbf{u}}\|$
- Gradient of loss function:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{\mathbf{u}} \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{I}}_{\mathbf{u}}} \cdot \frac{\partial \hat{\mathbf{I}}_{\mathbf{u}}}{\partial \theta}$$

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

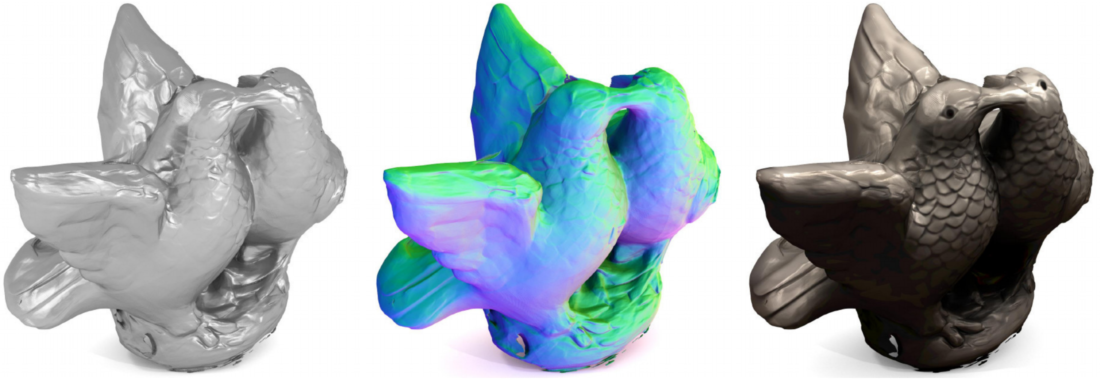
- Differentiation of $f_\theta(\hat{\mathbf{p}}) = \tau$ yields:

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



$\Rightarrow$ **Analytic solution** and **no need** for storing **intermediate results**

Results



Niemeyer, Mescheder, Oechsle and Geiger: Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. CVPR, 2020.

Summary

Summary

Neural Implicit Models:

- Effective output representation for shape, appearance, material, motion, etc.
- No discretization, model arbitrary topology
- Can be efficiently learned using 2D supervision
- Many applications: reconstruction, view synthesis, segmentation, etc.

Challenges:

- Geometry must be extracted in post-processing step (1-3 sec for ONet)
- Extension to 4D not straightforward (curse of dimensionality)
- Fully connected architecture and global condition lead to oversmooth results
- Promising: Local features (ConvONet, PiFU), Better input encoding (NeRF)

Thank you!

http://autonomousvision.github.io

European Research Council
Established by the European Commission

Federal Ministry
of Education
and Research

Microsoft Research