Differentiable Volumetric Rendering

Learning Implicit 3D Representations without 3D Supervision

MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

Michael Niemeyer[1,2] Lars Mescheder[1,2,3] Michael Oechsle[1,2,4] Andreas Geiger[1,2]

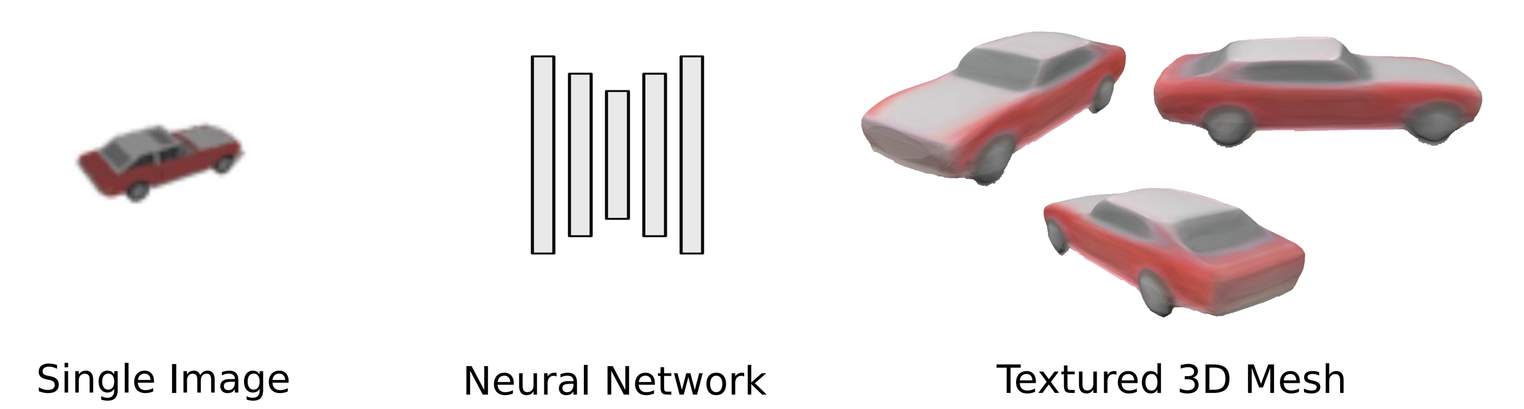
[1]Max Planck Institute for Intelligent Systems, Tübingen [2]University of Tübingen [3]Amazon, Tübingen [4]ETAS GmbH, Stuttgart

imprs-is

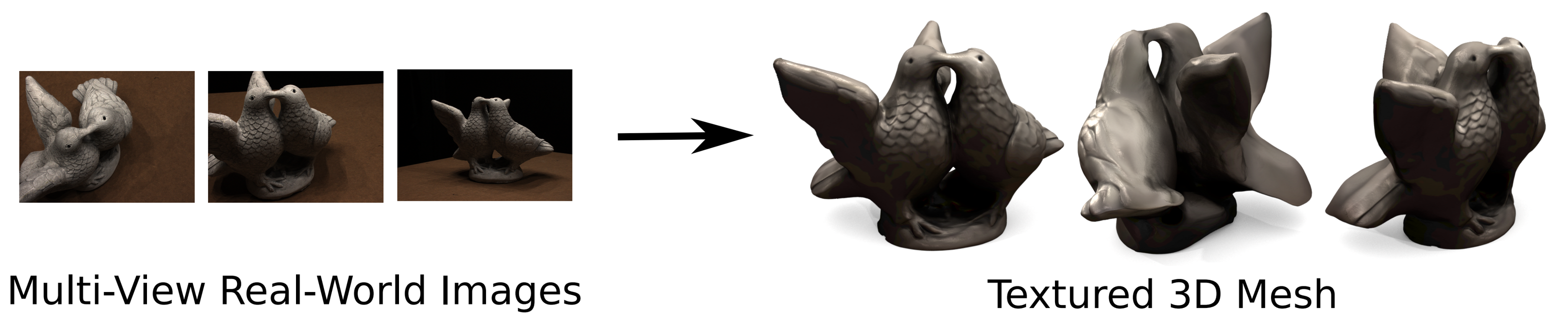
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Motivation

Our aim is to reconstruct the **shape and texture** of an object from single or multi-view images:



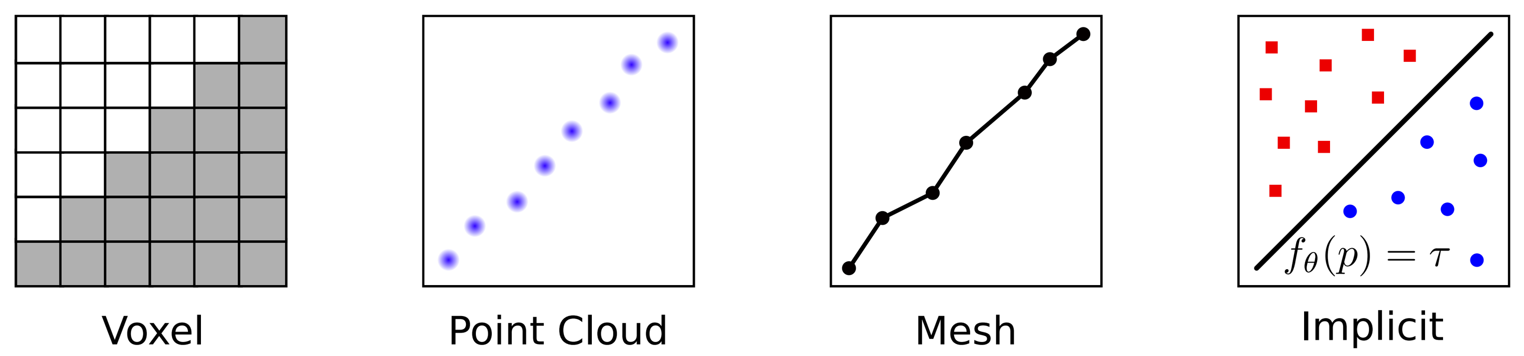
We want to apply our method to not only synthetic but also **real-world imagery**:



We want to train with **only multi-view images**, object masks and optional depth depths as supervision

Implicit Representations

Many recent learning-based reconstruction methods represent 3D geometry and texture **implicitly**



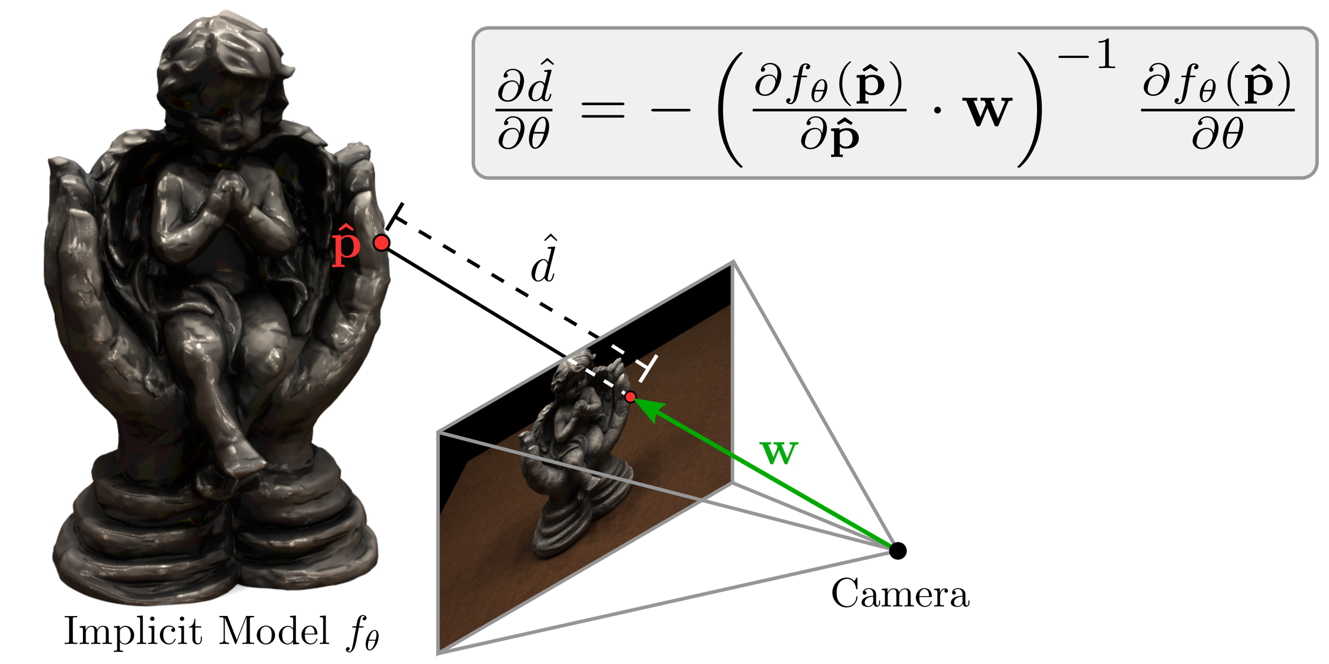
In contrast to previous representations, they do **not discretize space** and are not restricted in topology

However, existing methods require **3D ground truth information** for training

Hence, they are **restricted to synthetic data** and it is unclear how to scale to real-world imagery

How can we infer implicit 3D representations without 3D supervision?

Our Contribution

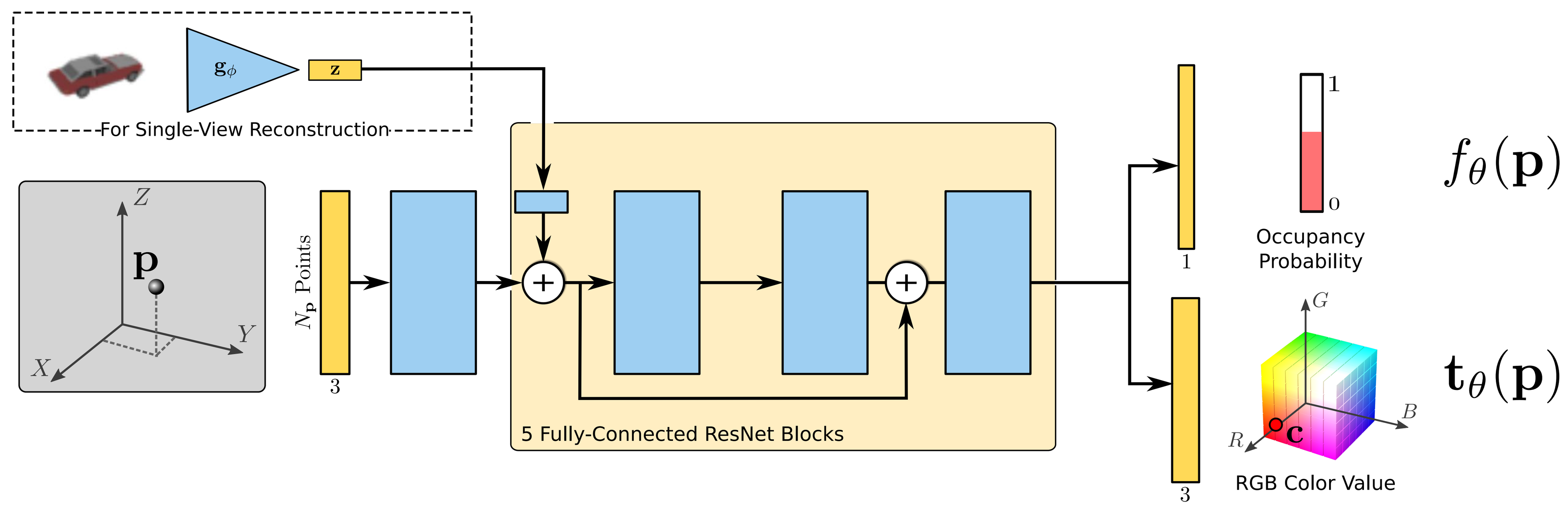


We propose to **differentiably render implicit representations**

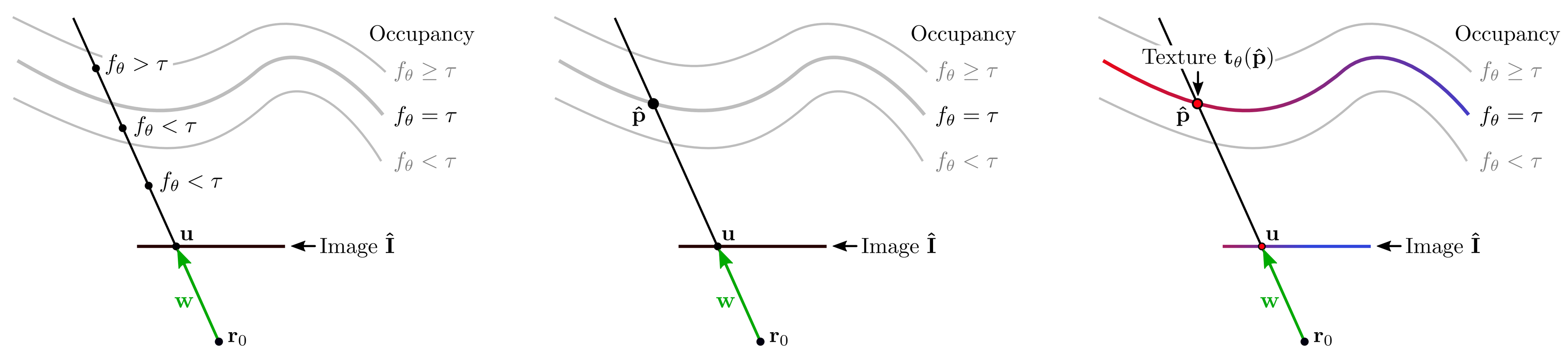
We derive an **analytic expression** for the gradient of the depth $\hat{d}$ with respect to the network parameters θ

+ We can train **without** 3D supervision
+ We get an **exact formula** without approximation
+ We do **not** need to store any intermediate steps

Our Model

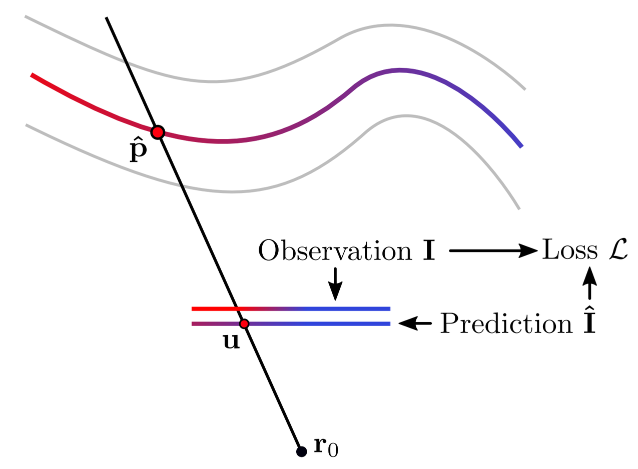


Forward Pass



1. March along ray to find first interval of occupancy change
2. Find surface point $\hat{\mathbf{p}}$ as root in this interval
3. Evaluate $\mathbf{t}_\theta(\hat{\mathbf{p}})$ to get RGB color value and insert at pixel $\mathbf{u}$

Backward Pass



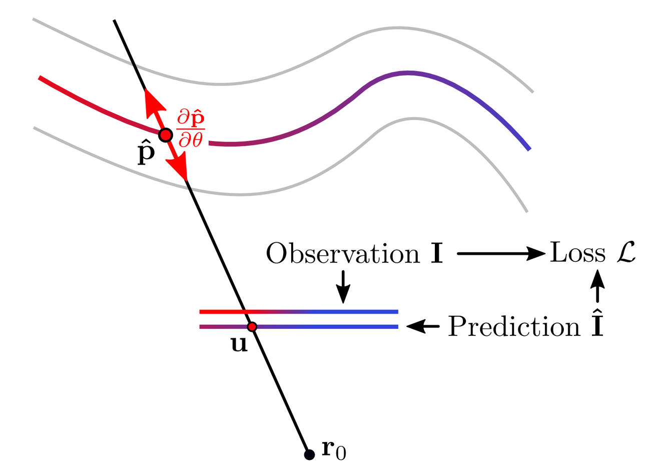
Loss Function:

$$\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}) = \sum_{\mathbf{u}} |\hat{\mathbf{I}}_{\mathbf{u}} - \mathbf{I}_{\mathbf{u}}|_1$$

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

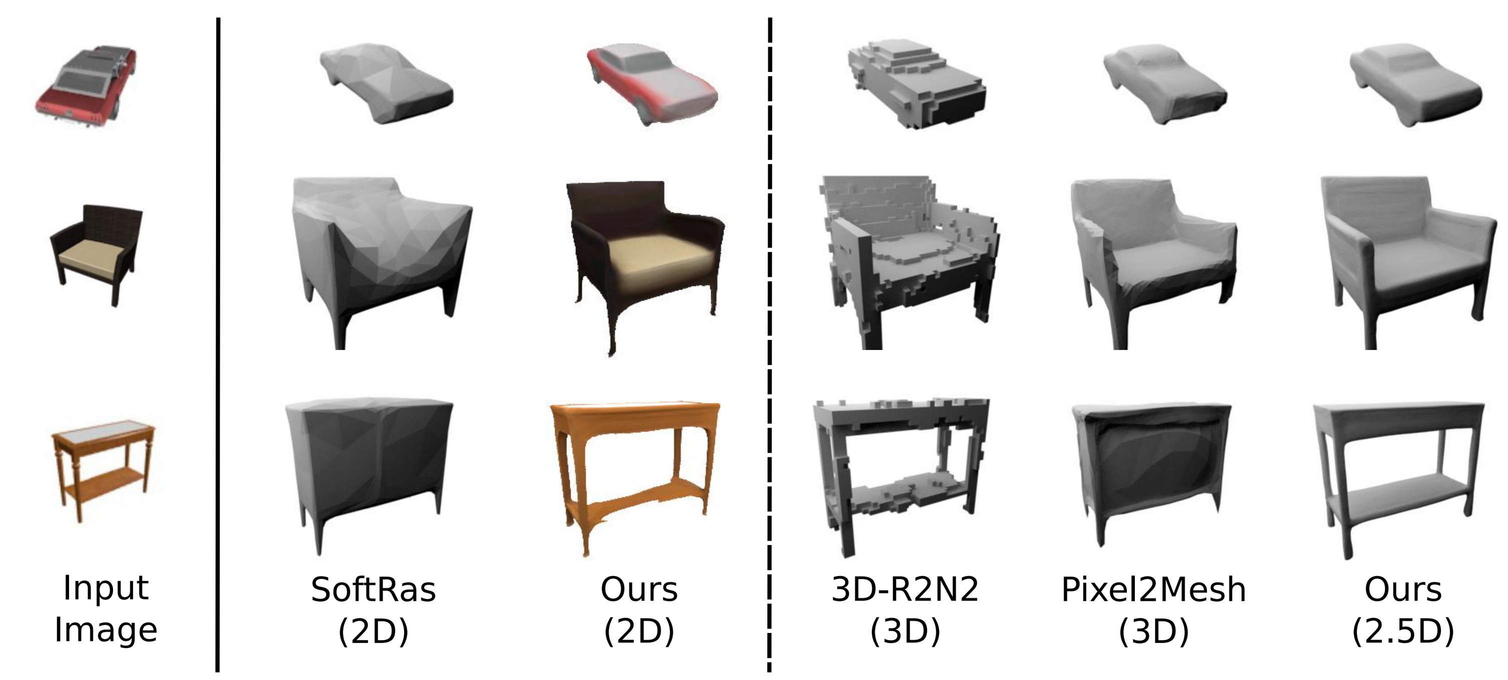
$$\frac{\partial \hat{d}}{\partial \theta} = -\left(\frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \mathbf{w}\right)^{-1} \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta}$$

Differentiation of $\hat{\mathbf{p}} = \mathbf{r}_0 + \mathbf{w} \cdot \hat{d}$:

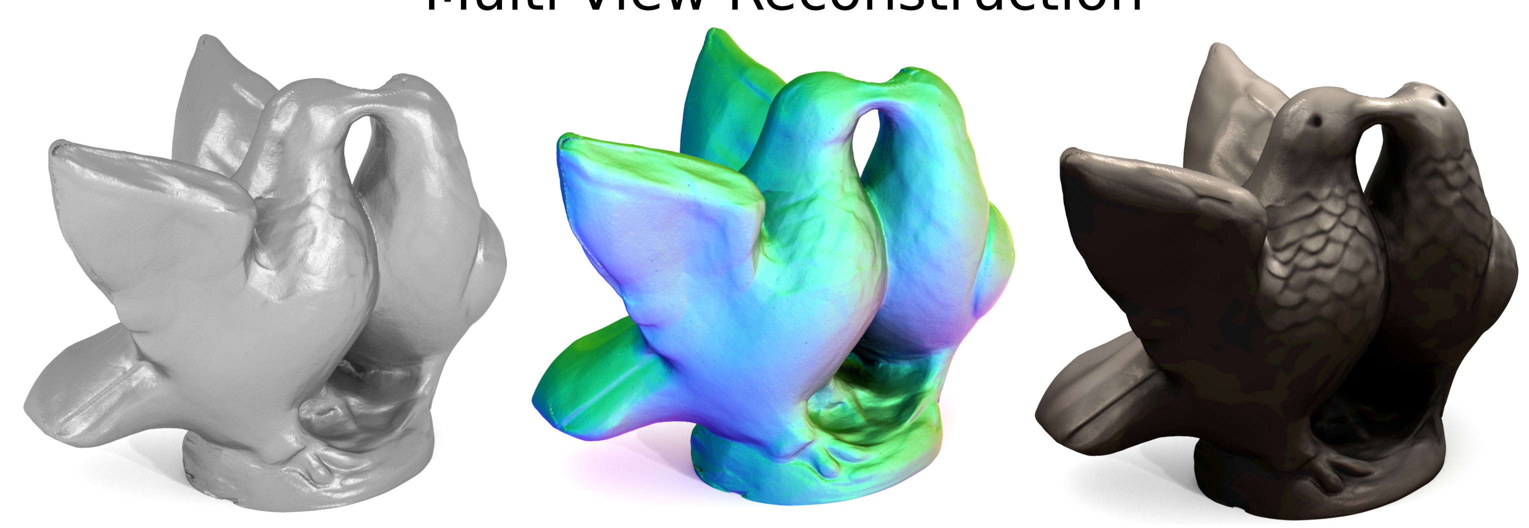
$$\frac{\partial \hat{\mathbf{p}}}{\partial \theta} = \mathbf{w} \cdot \frac{\partial \hat{d}}{\partial \theta}$$

Experiments

Single-View Reconstruction



Multi-View Reconstruction



Training Progression

https://tiny.cc/dvr-project

michael.niemeyer@tue.mpg.de