Neural Implicit Representations for 3D Vision and Beyond

Prof. Dr.-Ing. Andreas Geiger
Autonomous Vision Group

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MAX-PLANCK-GESELLSCHAFT

ellis
European Laboratory for Learning and Intelligent Systems

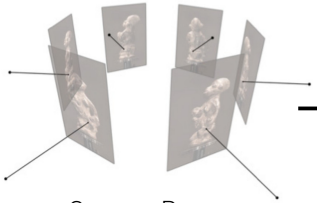
Agenda

1. Implicit Neural Representations
2. Differentiable Volumetric Rendering
3. Neural Radiance Fields
4. Generative Radiance Fields
5. Further Applications

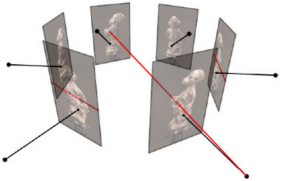
1
Implicit Neural Representations

Traditional 3D Reconstruction Pipeline

Input Images

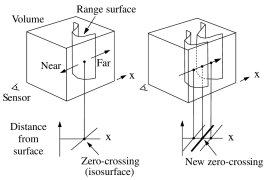


Camera Poses

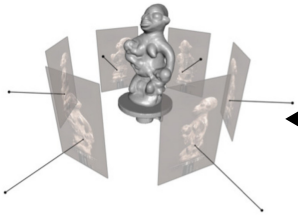


Dense Correspondences

Depth Maps

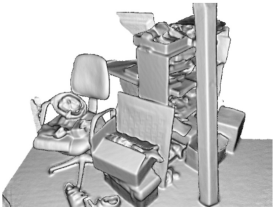


Depth Map Fusion

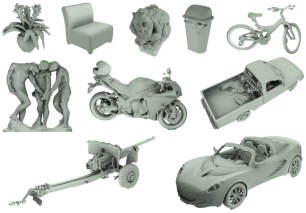


3D Reconstruction

3D Datasets and Repositories



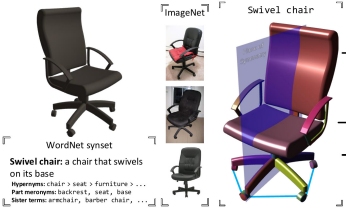
[Newcombe et al., 2011]



[Choi et al., 2011]

[Dai et al., 2017]

[Wu et al., 2015]

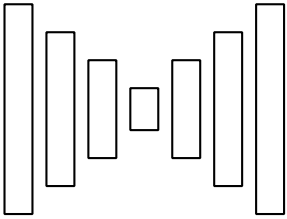


[Chang et al., 2015]

[Chang et al., 2017]

Can we learn 3D Reconstruction from data?

Input Images

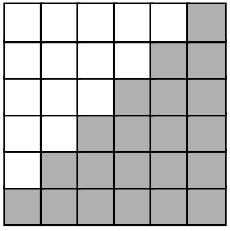


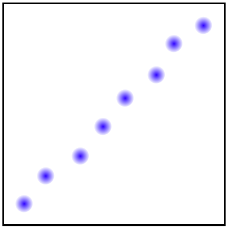
Neural Network

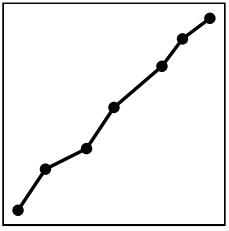
3D Reconstruction

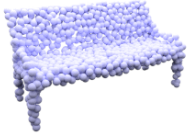
What is a good **output** representation?

3D Representations









Voxels

[Maturana et al., IROS 2015]

Points

[Fan et al., CVPR 2017]

Meshes

[Groueix et al., CVPR 2018]

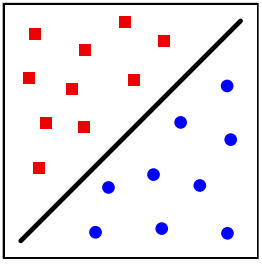
Occupancy Networks

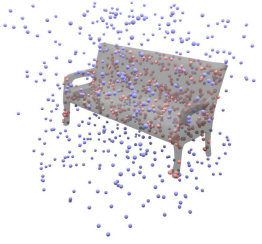
Key Idea:

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

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

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





Remarks:

- The function f_θ models an **occupancy field**

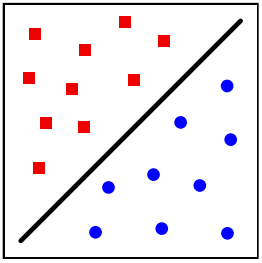
Occupancy Networks

Key Idea:

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

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

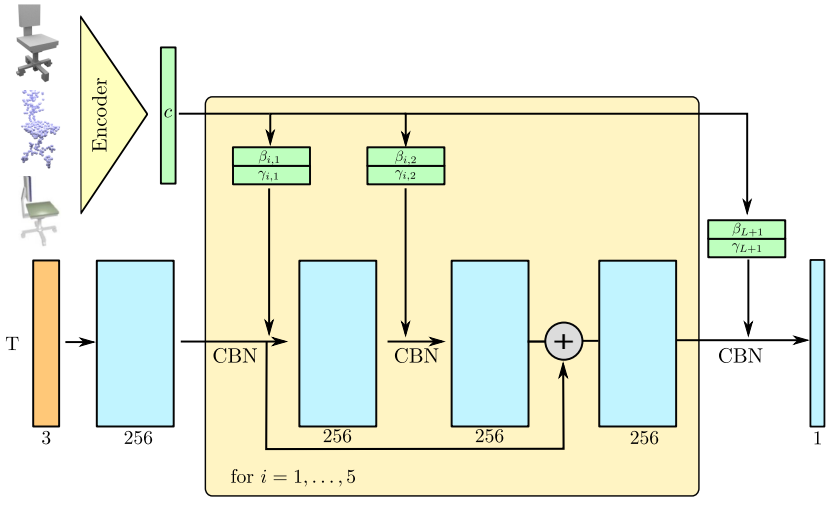
3D Location | Condition (eg, Image) | Occupancy Probability



Remarks:

- The function f_θ models an **occupancy field**
- Also possible: **signed distance field** [Park et al., 2019]

Network Architecture



Training Objective

Occupancy Network:

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

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

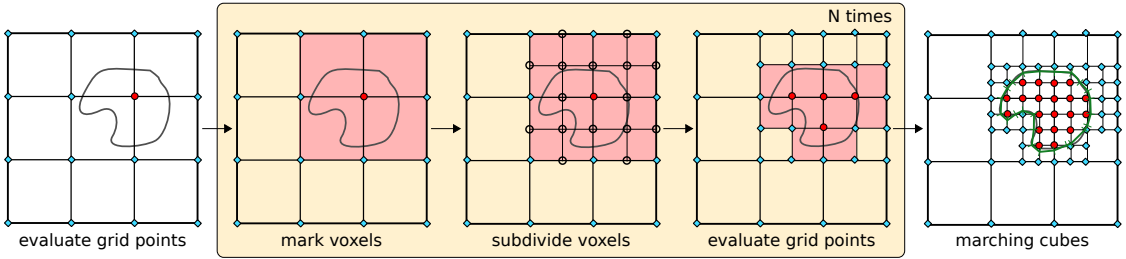
Training Objective

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

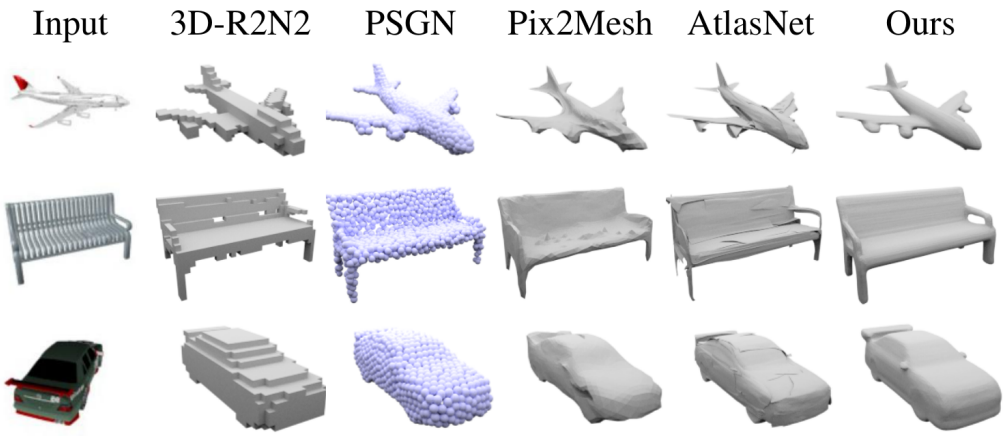
Occupancy Networks



Multiresolution IsoSurface Extraction (MISE):

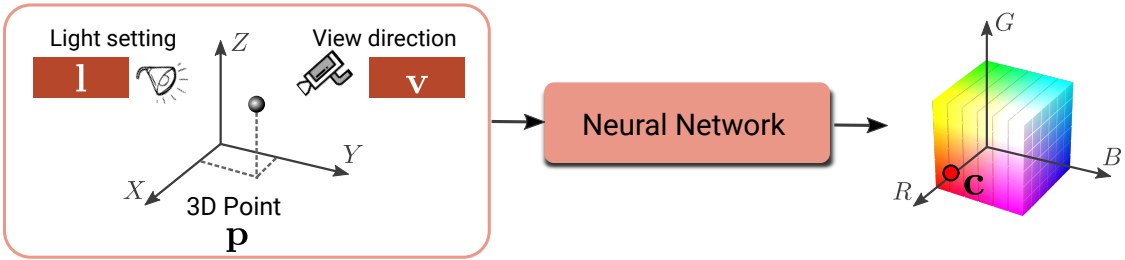
- Build octree by incrementally querying the occupancy network
- Extract triangular mesh using marching cubes algorithm (1-3 seconds in total)

Results



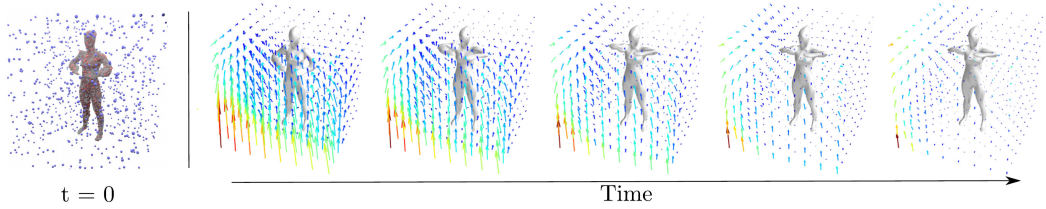
Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

Representing View-Dependent Appearance



Oechsle, Niemeyer, Mescheder, Strauss and Geiger: Learning Implicit Surface Light Fields. 3DV, 2020.

Representing Motion



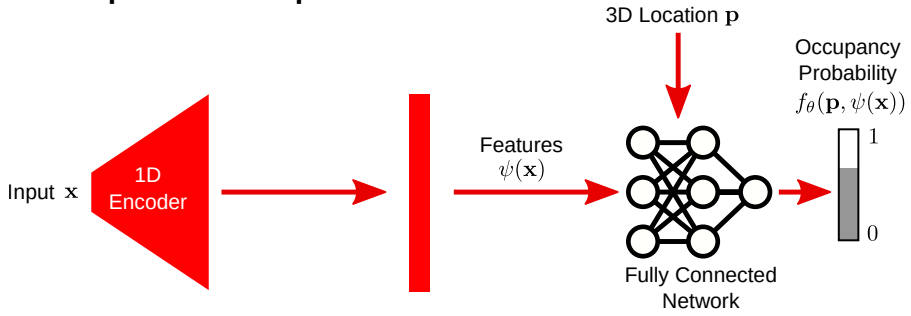
- Extending Occupancy Networks to 4D is hard (curse of dimensionality)
- Represent **shape** at $t = 0$ using a 3D Occupancy Network
- Represent **motion** by temporally and spatially continuous vector field
- Relationship between 3D trajectory $\mathbf{s}$ and velocity $\mathbf{v}$ given by (differentiable)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

Limitations

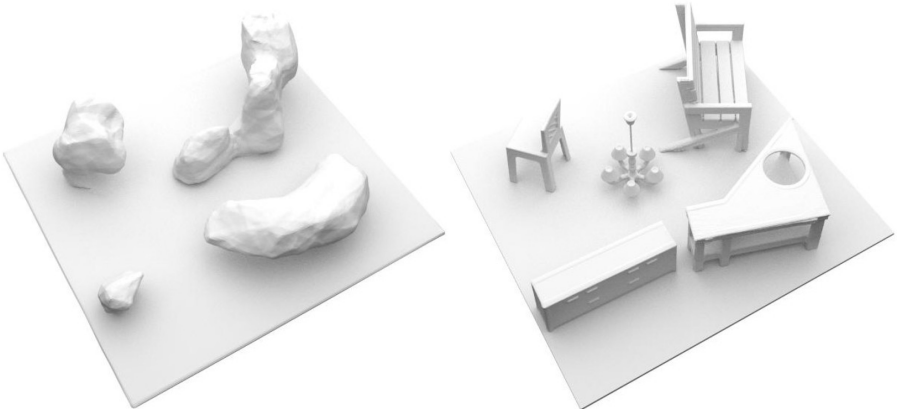
Structure of implicit neural representations:



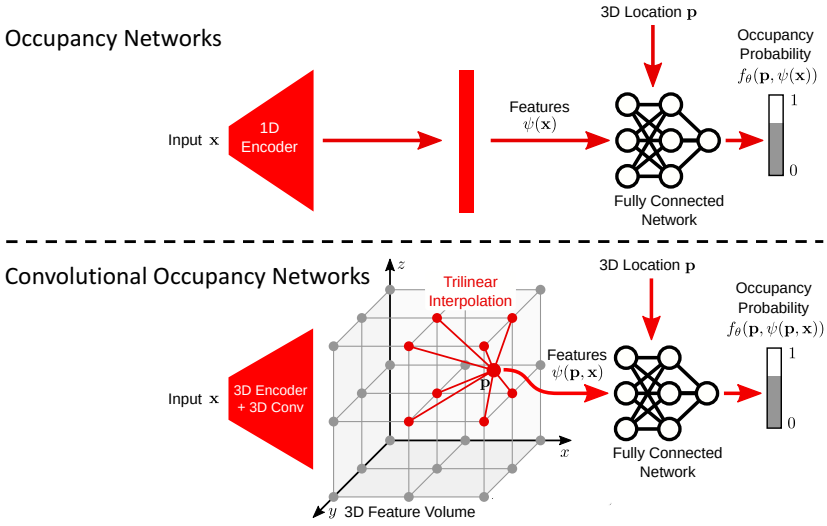
- **Global latent code** $\Rightarrow$ no local information, overly smooth geometry
- **Fully connected architecture** $\Rightarrow$ does not exploit translation equivariance

Limitations

Implicit models work well for simple objects but **poorly on complex scenes:**

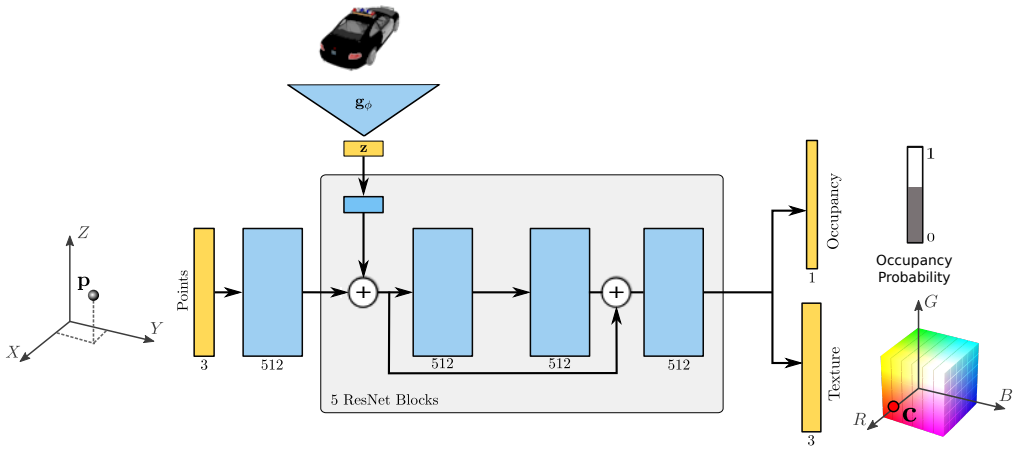


Convolutional Occupancy Networks



2
Differentiable Volumetric Rendering

DVR: Differentiable Volumetric Rendering



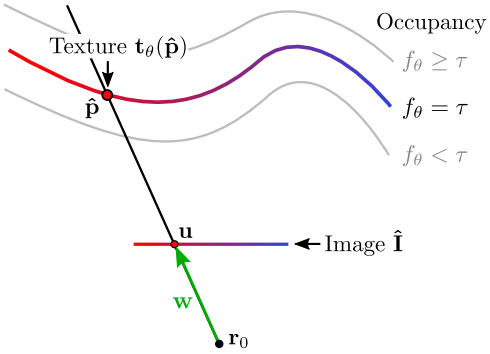
Niemeyer, Mescheder, Oechsle and Geiger: Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. CVPR, 2020.

Forward Pass
(Rendering)

DVR: 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 (secant method)
- Evaluate texture field $\mathbf{t}_\theta(\hat{\mathbf{p}})$ at $\hat{\mathbf{p}}$
- Insert color $\mathbf{t}_\theta(\hat{\mathbf{p}})$ at pixel $\mathbf{u}$



Backward Pass
(Differentiation)

DVR: Differentiable Volumetric Rendering

Backward Pass:

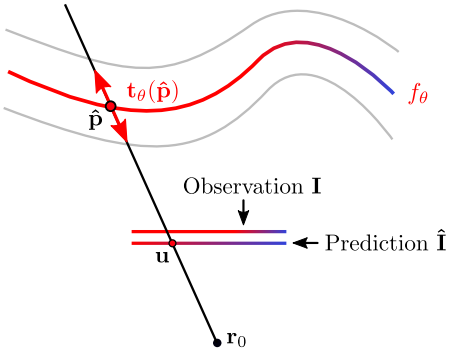
- Image Observation $\mathbf{I}$
- Loss $\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}) = \sum_{\mathbf{u}} \|\hat{\mathbf{I}}_{\mathbf{u}} - \mathbf{I}_{\mathbf{u}}\|$
- Gradient of loss function:

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

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

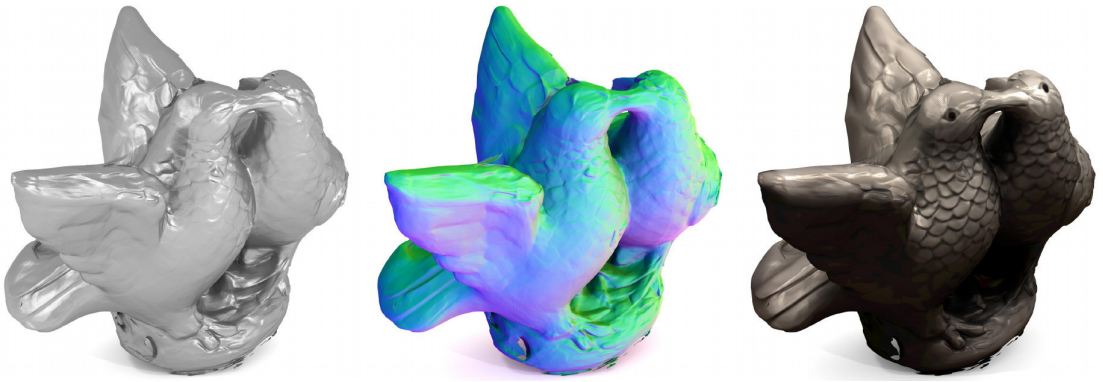
- Differentiation of $f_\theta(\hat{\mathbf{p}}) = \tau$ yields:

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



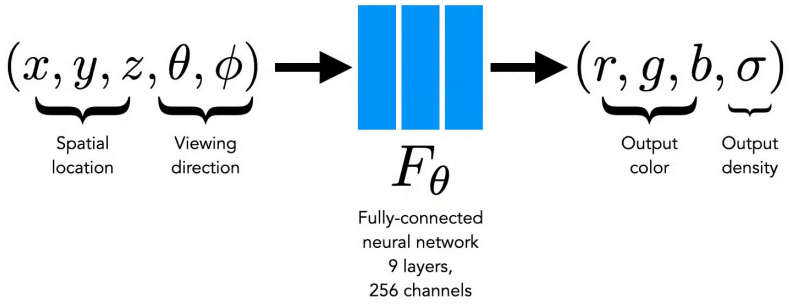
$\Rightarrow$ **Analytic solution** and **no need** for storing **intermediate results**

Results



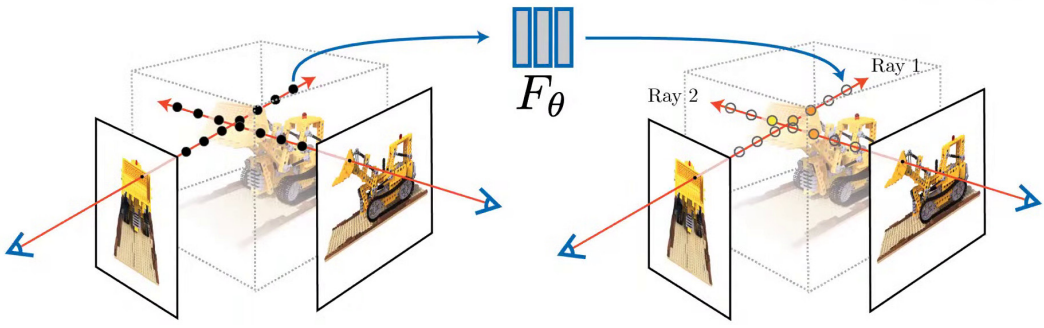
3
Neural Radiance Fields

NeRF: Representing Scenes as Neural Radiance Fields



- **Task:** Given a set of images of a scene, render image from novel viewpoint
- Vanilla **ReLU MLP** that maps from **location/view direction to color/density**
- Conditioning on view direction allows for modeling **view-dependent effects**

Volume Rendering

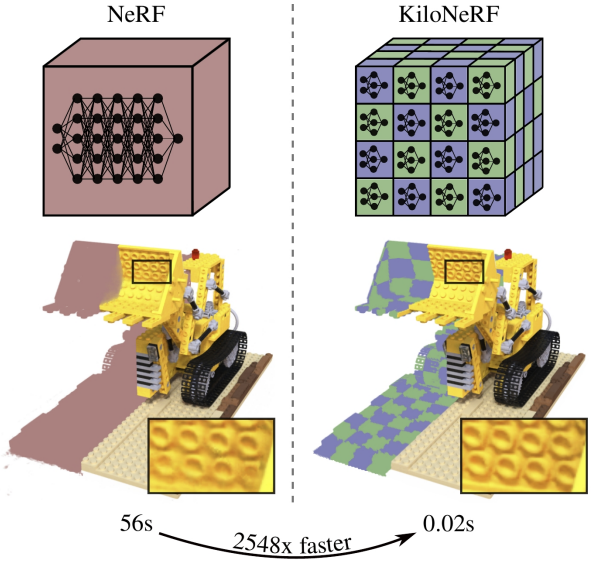


- **Volume rendering** works very similar to traditional **ray tracing** in graphics
- Shoot ray through scene, sample points, apply **alpha composition** to render pixel
- **Fourier features** improve visual fidelity (i.e., texture details) significantly

Mildenhall, Srinivasan, Tancik, Barron, Ramamoorthi and Ng: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV, 2020.

NeRF Results

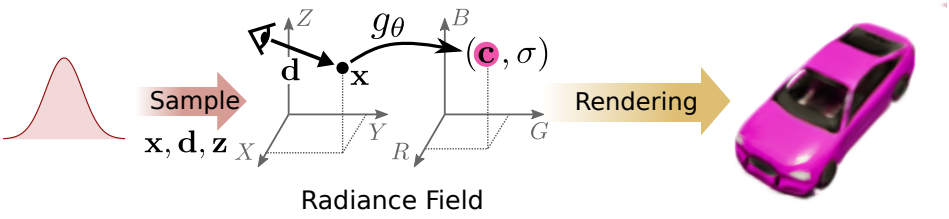
KiloNeRF



Reiser, Peng, Liao and Geiger: KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. ICCV, 2021.

4
Generative Radiance Fields

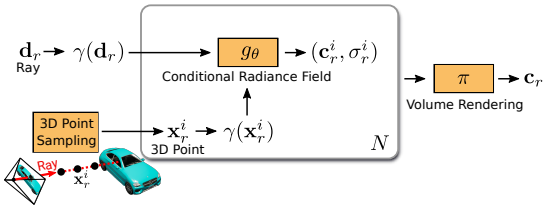
GRAF: Generative Radiance Fields



- **Generative model** for **radiance fields**
- Train from **unstructured and unposed 2D image collections**
- Challenges: Can't discriminate in 3D, but volumetric rendering is slow

Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020.

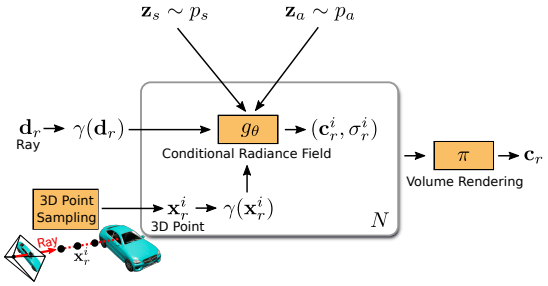
GRAF: Generative Radiance Fields



- A **radiance field** g_θ maps a 3D point $\mathbf{x}_r^i$ and viewing direction $\mathbf{d}_r$ to color/density
- By sampling N points along the ray, we can **render** a pixel's color $\mathbf{c}_r$

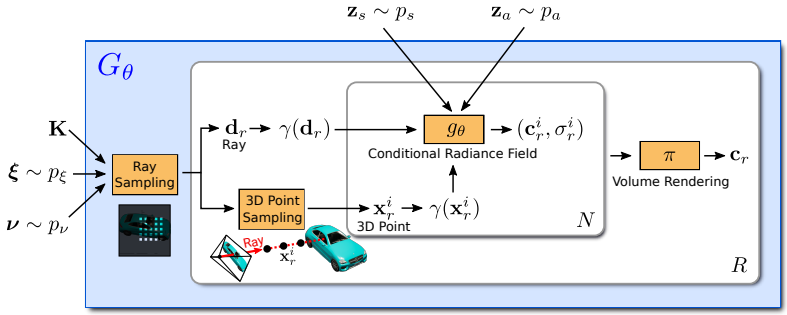
Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020.

GRAF: Generative Radiance Fields



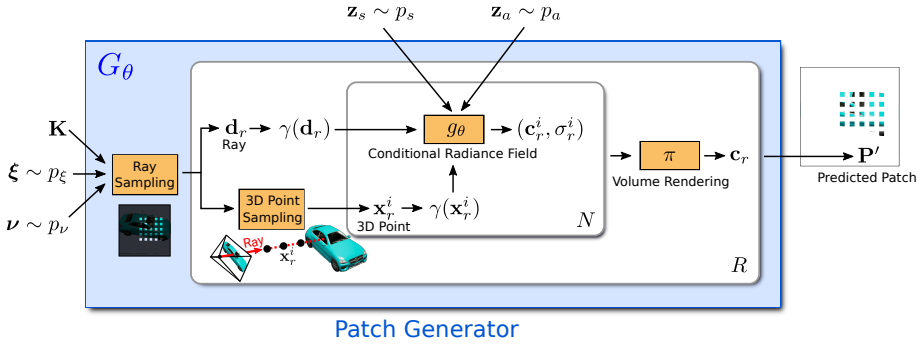
- GRAF **conditions** the radiance field g_θ on additional latent codes $\mathbf{z}$
- Here, $\mathbf{z}_s$ is a **shape latent code** and $\mathbf{z}_a$ is an **appearance latent code**

GRAF: Generative Radiance Fields



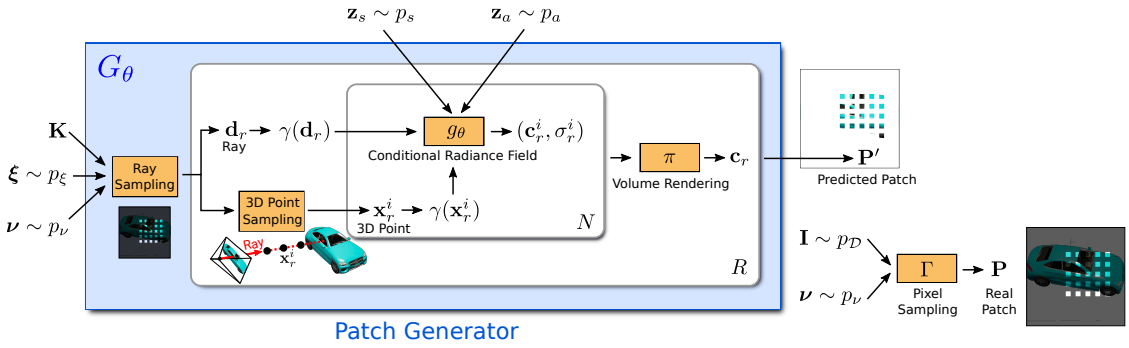
- The generator repeats this process for R rays, sparsely sampled on a 2D grid
- The camera **intrinsics** $\mathbf{K}$, **extrinsics** $\boldsymbol{\xi}$ and **2D grid** $\boldsymbol{\nu}$ are drawn randomly

GRAF: Generative Radiance Fields



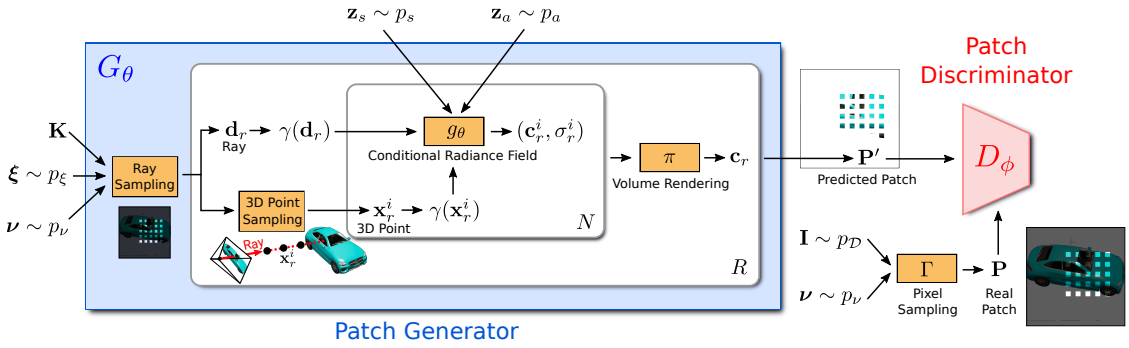
- This generates **image patches** of size 32×32 pixels
- The sampling pattern changes the **location** and **stride** (scale)

GRAF: Generative Radiance Fields



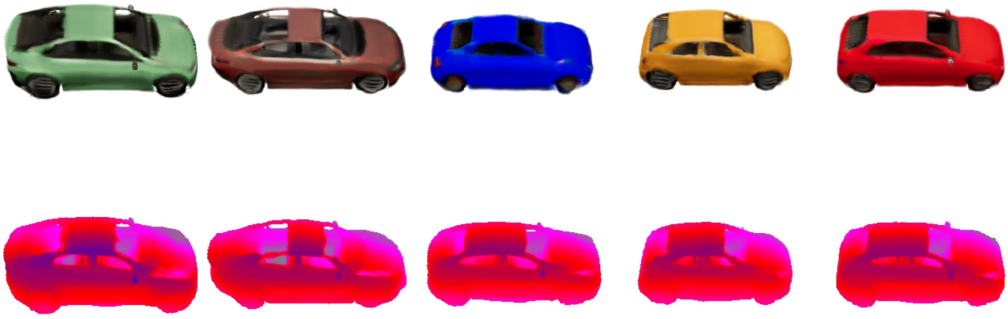
- Similarly to the generator, we can extract **real patches** of the same size
- Here $\mathbf{I}$ denotes a real image sampled from the data distribution $p_\mathcal{D}$

GRAF: Generative Radiance Fields



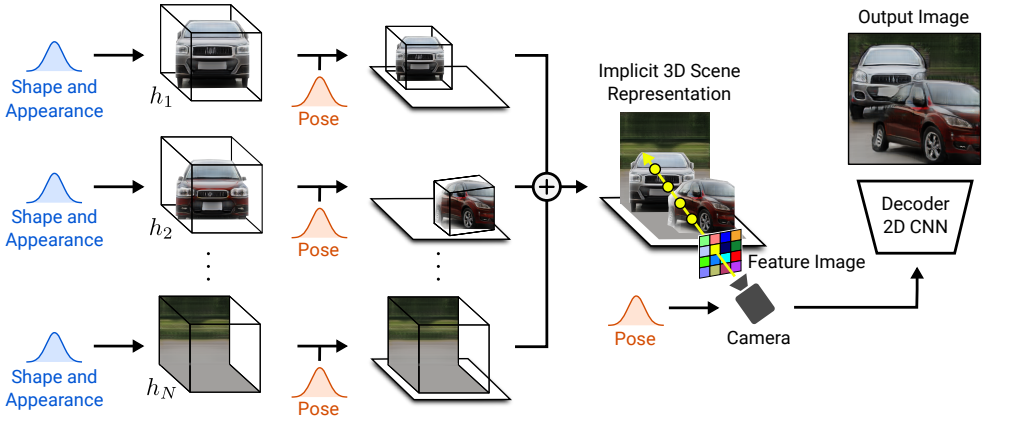
- We can now compare both patches using a **2D discriminator**
- GRAF implements the 2D discriminator as a simple **4 layer ConvNet**

GRAF: Generative Radiance Fields



Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020.

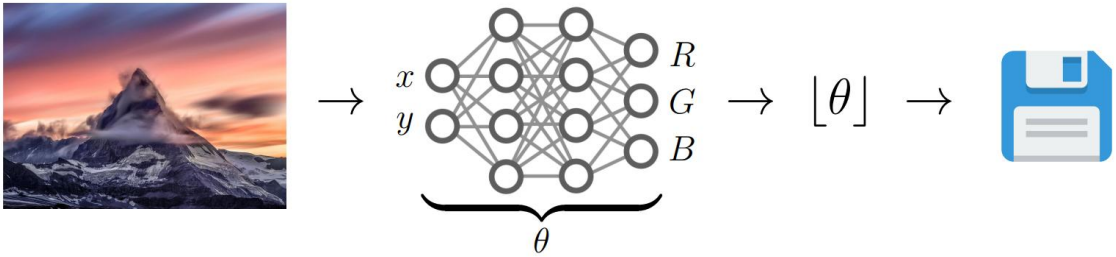
GIRAFFE: Compositional Generative Neural Feature Fields



Niemeyer and Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021. Best Paper Award.

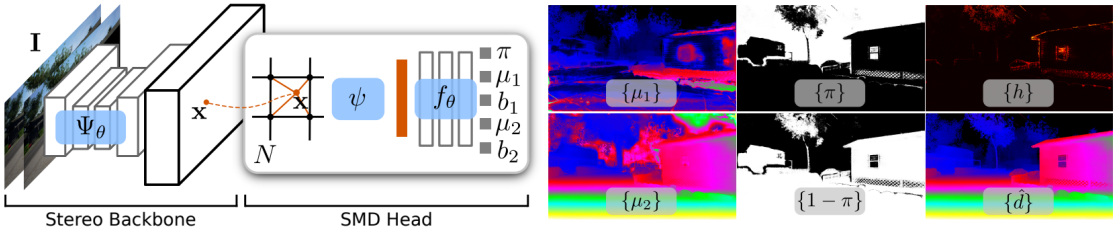
5
Further Applications

COIN: Compression with Implicit Neural representations



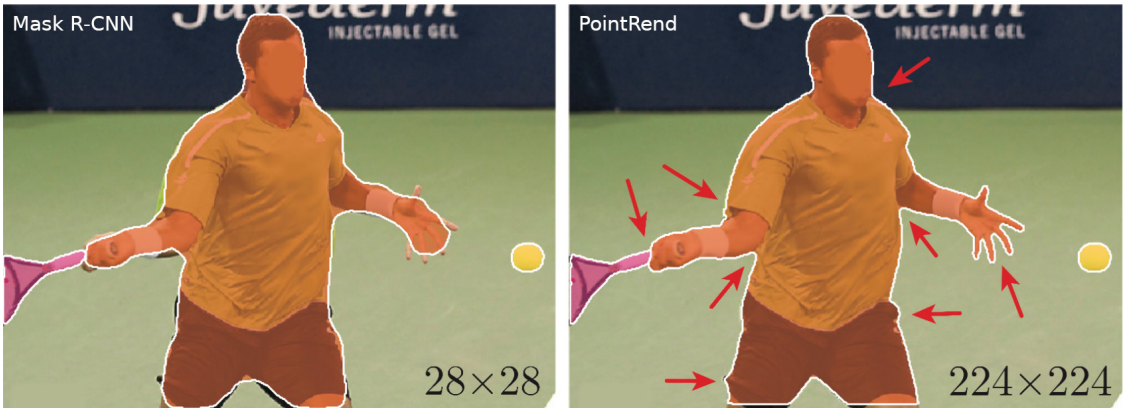
- Parameters of trained neural network act as compressed representation of data

SMD-Nets: Stereo Mixture Density Networks



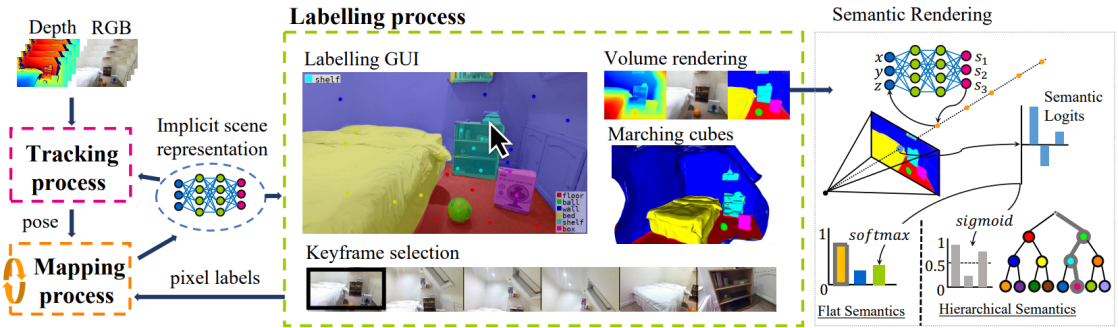
- MLP decoder enables to query stereo disparity maps at arbitrary resolution

PointRend: Image Segmentation As Rendering



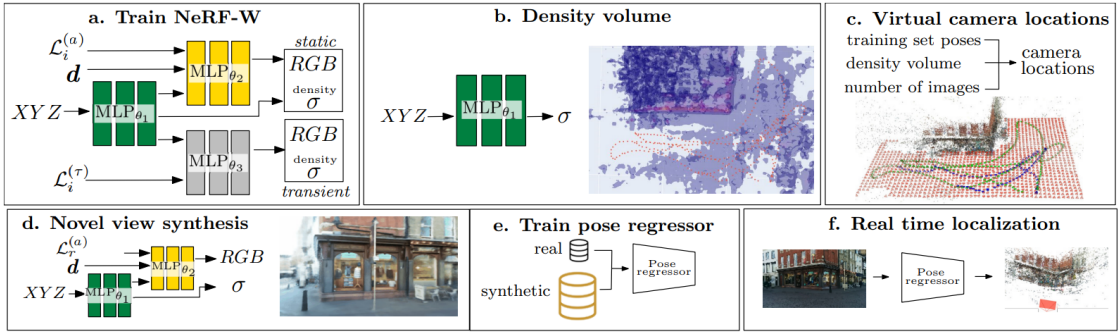
- MLP decoder allows for accurate instance segmentation in Mask R-CNN

iLabel: Interactive Neural Scene Labelling



- Predicting geometry and semantics jointly enables efficient few-shot annotation

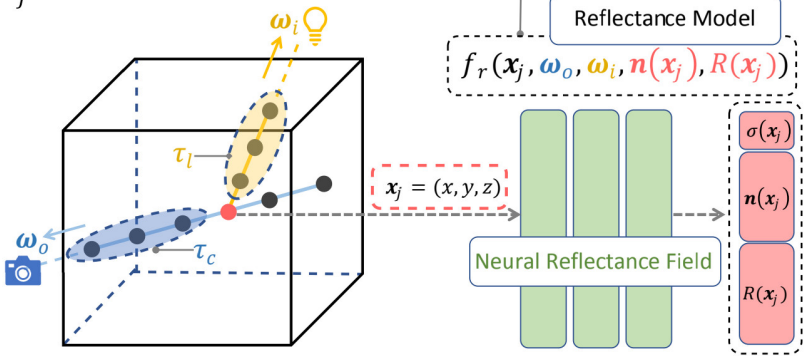
LENS: Localization enhanced by NeRF synthesis



- NVS improves downstream tasks like localization via data augmentation

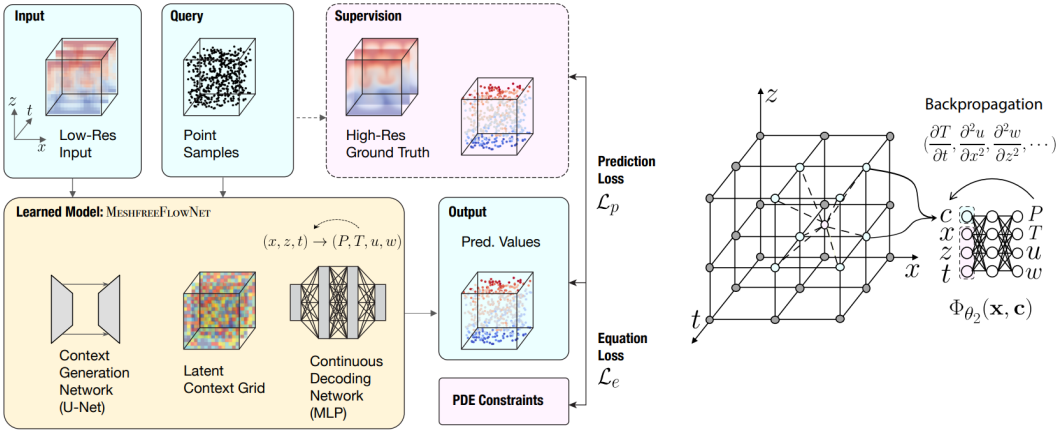
Neural Reflectance Fields for Appearance Acquisition

$$\sum_j \tau_c(\boldsymbol{x}_j)\,\tau_l(\boldsymbol{x}_j)\,\left(1-\exp\left(-\sigma(\boldsymbol{x}_j)\,\Delta t_j\right)\right) f_r(\boldsymbol{x}_j)\,L_l(\boldsymbol{x}_j)$$



- Beyond light fields: modeling material properties using coordinate-based models

MeshfreeFlowNet: Deep continuous space-time super-resolution

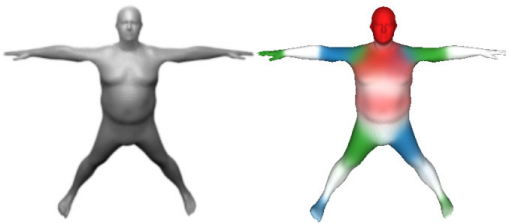


- Space-time super-resolution of turbulent flows (Rayleigh-Bénard convection)

SNARF: Differentiable Forward Skinning for Animating Neural Shapes

Input 3D Posed Meshes

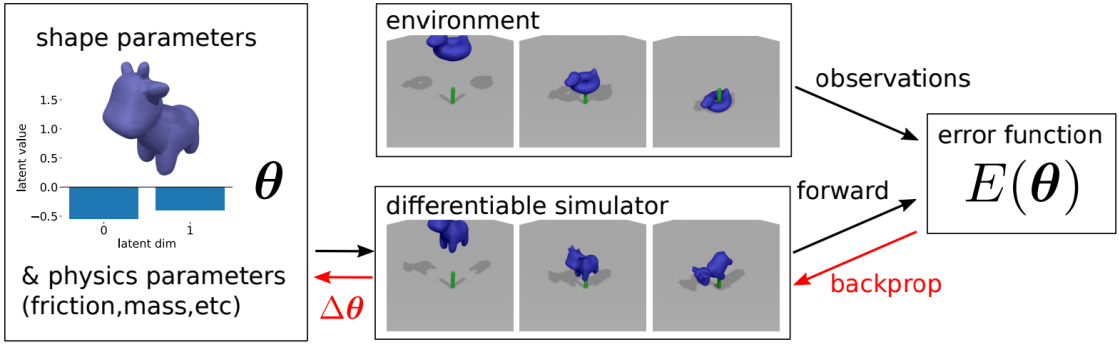
Continuous Implicit Surfaces in Unseen Poses



Learned Canonical 3D Shape and Skinning Weights

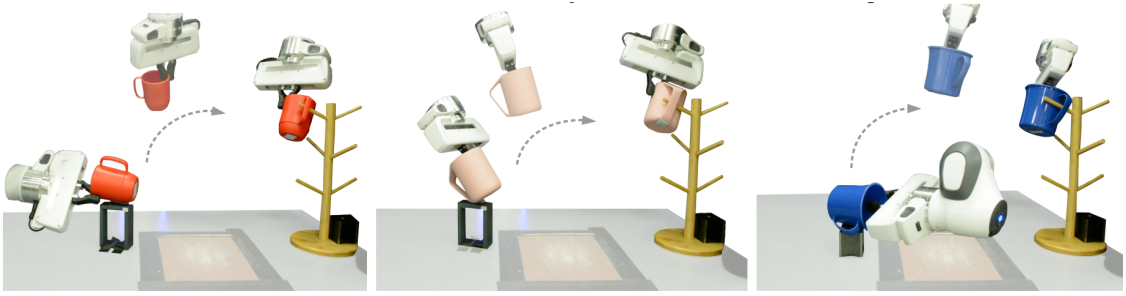
- Learning articulated human shape representations (geometry and skinning)

DiffSDFSim: Differentiable Rigid-Body Dynamics With Implicit Shapes



- Learning differentiable rigid-body dynamics (mass, friction) with implicit shapes

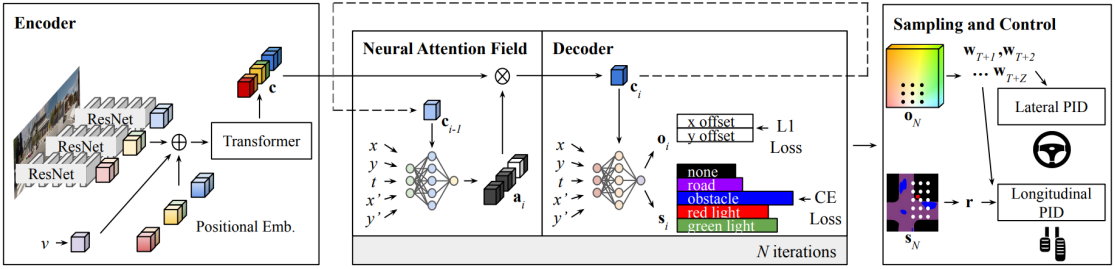
Neural Descriptor Fields



- Few-shot generalization of manipulation tasks by predicting spatial descriptors

Simeonov, Du, Tagliasacchi, Tenenbaum, Rodriguez, Agrawal and Sitzmann: Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation. Arxiv, 2021.

NEAT: Neural Attention Fields for End-to-End Autonomous Driving

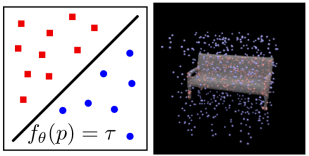


- Continuous state representation and waypoint prediction for self-driving

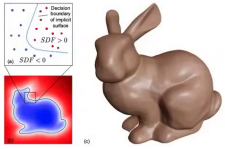
Summary

Neural Networks as Continuous Representations

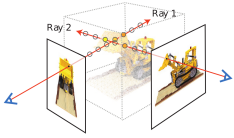
Occupancy Networks
[Mescheder et al. 2019]
$(x, y, z) \rightarrow \text{occupancy}$



DeepSDF
[Park et al. 2019]
$(x, y, z) \rightarrow \text{distance}$

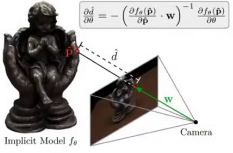


Neural Radiance Fields
[Mildenhall et al. 2020]
$(x, y, z, \theta, \phi) \rightarrow \text{color, density}$

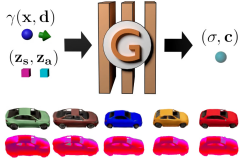


Scene Representation Networks
[Sitzmann et al. 2019]
$(x, y, z) \rightarrow \text{latent vec. (color, dist)}$

Differentiable Volumetric Rendering
[Niemeyer et al. 2020]
$(x, y, z) \rightarrow \text{color, occ.}$



Generative Radiance Fields
[Schwarz et al. 2020]
$(x, y, z, \theta, \phi, \mathbf{z}) \rightarrow \text{color, density}$



Summary

Coordinate-Based Networks:

- Effective output representation for shape, appearance, material, motion
- No discretization, model arbitrary topology
- Can be learned from images via differentiable rendering
- Many applications: reconstruction, motion, view synthesis, robotics

However:

- Geometry must be extracted in post-processing step (1 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), Better input encoding (NeRF)

Thank you!

http://autonomousvision.github.io

Federal Ministry
of Education
and Research

Federal Ministry
for Economic Affairs
and Energy

Microsoft
Research