Occupancy Networks:
Learning 3D Reconstruction in Function Space

Andreas Geiger

Autonomous Vision Group
University of Tübingen / MPI for Intelligent Systems
Occupancy Networks: Learning 3D Reconstruction in Function Space

[Mescheder, Oechsle, Niemeyer, Nowozing & Geiger, CVPR 2019]
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:

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

**Concurrent work:**
- DeepSDF [Park et al., CVPR 2019]
- IM-NET [Chen et al., CVPR 2019]
Training Objective

Occupancy Network: Variational Occupancy Encoder:

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

- **K**: Randomly sampled 3D points \((K = 2048)\)
- **BCE**: Cross-entropy loss
- **\(q_\psi\)**: Encoder

Results

Input | 3D-R2N2 | PSGN | Pix2Mesh | AtlasNet | Ours

![Input](image1) ![3D-R2N2](image2) ![PSGN](image3) ![Pix2Mesh](image4) ![AtlasNet](image5) ![Ours](image6)

Can we also learn about object appearance?
Texture Fields

Texture Fields

Texture Fields

\[ t_\theta : \mathbb{R}^3 \times \mathcal{S} \times \mathcal{Z} \rightarrow \mathbb{R}^3 \]

\[ \mathcal{N}(0, I) \]

\[ z \]

\[ D \]

\[ u_i \]

\[ K^{-1} \]

\[ p_i \]

\[ \mathcal{N} \]

\[ S \]

\[ X \]

\[ \hat{X} \]

\[ \text{GAN Discriminator} \]

\[ \text{Adversarial Loss} \]

Color Legend:
- **Red**: Conditional Model
- **Green**: GAN Model
- **Blue**: VAE Model

Representation Power

▶ Ground truth vs. Texture Field vs. Voxelization

Results

<table>
<thead>
<tr>
<th>Projection</th>
<th>NVS</th>
<th>Texture Field</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Car" /></td>
<td><img src="image" alt="Car" /></td>
<td><img src="image" alt="Car" /></td>
</tr>
<tr>
<td><img src="image" alt="Chair" /></td>
<td><img src="image" alt="Chair" /></td>
<td><img src="image" alt="Chair" /></td>
</tr>
<tr>
<td><img src="image" alt="Airplane" /></td>
<td><img src="image" alt="Airplane" /></td>
<td><img src="image" alt="Airplane" /></td>
</tr>
<tr>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
<td><img src="image" alt="Table" /></td>
</tr>
</tbody>
</table>

Condition (2D) | Prediction (3D) | Condition (2D) | Prediction (3D)

What about object 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 $s$ and velocity $v$ given by (differentiable) ODE:

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

$$L_{\text{recon}}$$

Input $$x$$

Occupancy $$f_\theta^x$$

Velocity $$v_\theta^x$$

Solve ODE for $$t$$ from $$\tau$$ to 0

$$t = 0$$

$$t = \tau$$

Results

No correspondences needed ⇒ implicitly established by our model!

Summary
Thank you!

http://autonomousvision.github.io