ANISE: Assembly-based Neural Implicit Surface rEconstruction

Dmitry Petrov\textsuperscript{1†}  Matheus Gadelha\textsuperscript{2}  Radomír Měch\textsuperscript{2}  Evangelos Kalogerakis\textsuperscript{1}

\textsuperscript{1} University of Massachusetts Amherst  \textsuperscript{2} Adobe Research

\textsuperscript{†} Part of the work done as Adobe Research internship
Goal: reconstruct shapes as an assembly of neural implicit parts
Goal: reconstruct shapes as an assembly of neural implicit parts
Goal: reconstruct shapes as an assembly of neural implicit parts

Input RGB Image → ANISE → Neural implicit part representations → Output Shape
Goal: reconstruct shapes as an assembly of neural implicit parts
**Goal**: reconstruct shapes as an assembly of neural implicit parts
Prior work: neural implicit for 3D shape reconstruction

DeepSDF [Park et. al 2019]
OccupancyNet [Mescheder et. al 2019]
IM-NET [Chen et. al 2019]
**Prior work:** neural implicits for 3D shape reconstruction

\[ f(x, y, z, \psi) = \text{scalar} \]
Prior work: neural implicitics for 3D shape reconstruction

\[ f(x, y, z, \psi) = \text{scalar} \]

**Prior work:** neural implicit for 3D shape reconstruction

- **DeepSDF** [Park et. al 2019]
- **OccupancyNet** [Mescheder et. al 2019]
- **IM-NET** [Chen et. al 2019]

\[ f(x, y, z, \psi) = \text{scalar} \]

3D point \( (x, y, z) \) and input encoding \( \psi \) are fed into the neural network to produce a scalar output.
Prior work: neural implicit for 3D shape reconstruction

DeepSDF [Park et. al 2019]

OccupancyNet [Mescheder et. al 2019]

IM-NET [Chen et. al 2019]

\[ f(x, y, z, \psi) = \text{scalar} \]
Prior work: neural implicits for 3D shape reconstruction

DeepSDF [Park et. al 2019]

OccupancyNet [Mescheder et. al 2019]

IM-NET [Chen et. al 2019]

\[ f(x, y, z, \psi) = \text{scalar} \]

3D point

input encoding
e.g., SDF, UDF, occupancy

neural network
Prior work: neural implicits for 3D shape reconstruction

DeepSDF [Park et. al 2019]  
OccupancyNet [Mescheder et. al 2019]  
IM-NET [Chen et. al 2019]

\[ f(x, y, z, \psi) = \text{scalar} \]
**Prior work:** neural implicits for 3D shape reconstruction

- **DeepSDF [Park et. al 2019]**
- **OccupancyNet [Mescheder et. al 2019]**
- **IM-NET [Chen et. al 2019]**

\[ f(x, y, z, \psi) = \text{scalar} \]

**3D point**

**Input encoding**

E.g., SDF, UDF, occupancy

**Implicit Surface:**

Inside:

\[ f(x, y, z, \psi) < 0 \]

Outside:

\[ f(x, y, z, \psi) > 0 \]
Prior work: neural implicits for 3D shape reconstruction

- DeepSDF [Park et. al 2019]
- OccupancyNet [Mescheder et. al 2019]
- IM-NET [Chen et. al 2019]
- ConvOccNet [Mescheder et. al 2020]
- D2IM-Net [Li et. al 2021]
- SPAGHETTI [Hertz et al. 2022]
Prior work: assembly-based modeling

A probabilistic model of Component-based Shape Synthesis [Kalogerakis et al. 2012]
Prior work: assembly-based modeling

Probabilistic Reasoning for Assembly-Based 3D Modeling [Chaudhuri et al. 2011]
Prior work: supervised composite shape modeling

- Beta Shape Machine [Huang et al. 2015]
- GRASS [Li et al. 2017]
- Composite Shape Modeling via Latent Space Factorization [Dubrovina et al. 2019]
Prior work: supervised composite shape modeling

Beta Shape Machine [Huang et al. 2015]
Points

GRASS [Li et al. 2017]
Boxes/Voxels

Composite Shape Modeling via Latent Space Factorization [Dubrovina et al. 2019]
Voxels
Prior work: supervised composite shape modeling

PQ-NET [Wu et al. 2020]
ANISE: Contributions

• Assembly is treated as a *set of part implicits* -- order does not matter
ANISE: Contributions

- Assembly is treated as a set of part implicits -- order does not matter
- **End-to-end training**: supervision after assembling parts into full shape
ANISE: Contributions

- Assembly is treated as a set of part implicits -- order does not matter
- End-to-end training: supervision after assembling parts into full shape
- Coarse-to-fine generation – predict structure, then part geometry
ANISE: Contributions

- Assembly is treated as a **set of part implicits -- order does not matter**
- **End-to-end training**: supervision after assembling parts into full shape
- **Coarse-to-fine generation** – predict structure, then part geometry
ANISE: neural modules
ANISE: neural modules

Structure prediction

Input
ANISE: neural modules
ANISE: neural modules

Input → Structure prediction → Geometry prediction
ANISE: neural modules

- Structure prediction
- Geometry prediction
- Assembly

Input → Structure prediction → Geometry prediction → Assembly → Output
ANISE: neural modules

Structure prediction → Geometry prediction → Assembly

Input → Structure prediction

Geometry prediction → Assembly

Assembly → Output
Structure prediction module
Structure prediction module

Input

Encoder

Shape Code
Structure prediction module

Input → Encoder

ResNet18 OR PointNet

[He et. al 2015] [Qi et. al 2017]

Shape Code
Structure prediction module

Input → Encoder → Shape Code → Part Transformation Codes \( \{\theta_m\} \)
Structure prediction module

Encoder

Shape Code

MLP

Part Transformation Codes $\{\theta_m\}$
Structure prediction module

Input

Encoder

Shape Code

MLP

Part Transformation Codes \( \{\theta_m\} \)

Max number=10
ANISE: neural modules

Structure prediction → Geometry prediction → Assembly
Geometry prediction module

Input → Encoder → MLP → Part Transformation Codes $\{\theta_m\}$
Geometry prediction module

Input → Encoder → MLP → Part Transformation Codes $\{\theta_m\}$
Geometry prediction module

Encoder → MLP → Part Transformation Codes $\{\theta_m\}$ → MLP → Part Geometry Codes $\{\psi_m\}$

Input → Shape Code
ANISE: neural modules

Structure prediction → Geometry prediction → Assembly

Input → Structure prediction → Geometry prediction → Assembly → Output
Assembly module

Structure & Geometry modules

Part Geometry
Codes $\{\psi_m\}$

Part Transformation
Codes $\{\theta_m\}$
Assembly module

Structure & Geometry modules

Part Geometry Codes \( \psi_m \)

Part Transformation Codes \( \theta_m \)

\( (x, y, z) \)

MLP

Part Implicits

\( f(x, y, z, \psi_m) \)
Assembly module

Structure & Geometry modules

Part Geometry Codes \( \{\psi_m\} \)

Part Implicits \( f(x, y, z, \psi_m) \)

Part Translation Codes \( \{\theta_m\} \)

Part translation \( c_m \)

& scaling \( a_m \)
Assembly module

Structure & Geometry modules

Part Geometry Codes \( \{\psi_m\} \)

\( p \)

MLP

Part Implicit\( s f_m(p) \)

Part Translation Codes \( \{\theta_m\} \)

MLP

Part translation \( c_m \) & scaling \( a_m \)
Assembly module

Part Geometry Codes \( \{\psi_m\} \)

Part Transformation Codes \( \{\theta_m\} \)

MLP

Part Implicits \( f_m(p) \)

\[
\min_m a_m f_m \left( \frac{p - c_m}{a_m} \right)
\]

[Museth et al 2002]

Union of transformed part implicits

Part translation \( c_m \)

& scaling \( a_m \)
Assembly module

Structure & Geometry modules

Part Geometry Codes \( \{\psi_m\} \)

Part Transformation Codes \( \{\theta_m\} \)

MLP

\( p \)

Part Implicits \( f_m(p) \)

\[ \min_m a_m f_m \left( \frac{p - c_m}{a_m} \right) \]

Union of transformed part implicits

Part translation \( c_m \) & scaling \( a_m \)

Part Implicits \( f_m(p) \)
ANISE: training

Structure prediction → Geometry prediction → Assembly

Input → Structure prediction

Output
Need dataset of segmented parts => PartNet

[Mo et al. 2019]
Pre-training stage: learning part geometry codes
Pre-training stage: learning part geometry codes
Pre-training stage: learning part geometry codes

Encoder → Part Geometry Code $\psi_m$ → MLP → Part Implicit $f(p, \psi_m)$
Pre-training stage: learning part geometry codes

Encoder → Part Geometry Code $\psi_m$ → MLP → Part Implicit $f(p, \psi_m)$

SDF (L1) loss
More pre-training: geometry & structure modules

Structure & Geometry modules

Part Geometry Codes \( \{\psi_m\} \)

Part Transformation Codes \( \{\theta_m\} \)

L2 loss

\( p \)

MLP

Part Implicits \( f(p, \psi_m) \)

Part translation \( c_m \)

& scaling \( a_m \)
More pre-training: geometry & structure modules

Part Geometry Codes $\{\psi_m\}$

Part Implicit $f(p, \psi_m)$

Part Translation $c_m$ & Scaling $a_m$

Structure & Geometry modules
More pre-training: geometry & structure modules

Part Geometry Codes $\{\psi_m\}$

Part Transformation Codes $\{\theta_m\}$

L2 loss

MLP

Part Implicit $f(p, \psi_m)$

Part translation $c_m$ & scaling $a_m$

L2 loss

Structure & Geometry modules
More pre-training: geometry & structure modules

Structure & Geometry modules

Part Geometry Codes \( \{\psi_m\} \)

Part Implicit \( f(p, \psi_m) \)

L2 loss

Part Transformation Codes \( \{\theta_m\} \)

Part translation \( c_m \)

Part scaling \( a_m \)

L2 loss
Final training stage

Structure & Geometry modules

Part Geometry Codes \(\{\psi_m\}\)

Part Transformation Codes \(\{\theta_m\}\)

MLP

Part Implicits \(f(p, \psi_m)\)

Part translation \(c_m\) & scaling \(a_m\)

Union of transformed part implicits

SDF (L1) loss

\[
\min_m a_m f_m \left( \frac{p - c_m}{a_m} \right)
\]
Results

PartNet training / testing
ANISE vs other part-aware methods

Input point cloud

Ground-truth
ANISE vs other part-aware methods

Input point cloud

PQ-NET
[Wu et al. 2020]

Ground-truth
ANISE vs other part-aware methods

Input point cloud

PQ-NET
[Wu et al. 2020]

JLRD
[Uy et al. 2021]

Ground-truth
ANISE vs other part-aware methods

Input point cloud

PQ-NET
[Wu et al. 2020]

JLRD
[Uy et al. 2021]

ANISE

Ground-truth
ANISE vs other part-aware methods

Input point cloud

Ground-truth
ANISE vs other part-aware methods

Input point cloud

PQ-NET
[Wu et al. 2020]

Ground-truth
ANISE vs other part-aware methods

Input point cloud

PQ-NET [Wu et al. 2020]

JLRD [Uy et al. 2021]

Ground-truth
ANISE vs other part-aware methods

Input point cloud

PQ-NET
[Wu et al. 2020]

JLRD
[Uy et al. 2021]

ANISE

Ground-truth

[Wu et al. 2020]
[Uy et al. 2021]
ANISE vs other part-aware methods

Input RGB image

Ground-truth
ANISE vs other part-aware methods

Input RGB image

PQ-NET
[Wu et al. 2020]

Ground-truth
ANISE vs other part-aware methods

Input RGB image
PQ-NET [Wu et al. 2020]
JLRD [Uy et al. 2021]
Ground-truth
ANISE vs other part-aware methods

Input RGB image

PQ-NET [Wu et al. 2020]

JLRD [Uy et al. 2021]

ANISE

Ground-truth
## ANISE vs other part-aware methods

### Single-view reconstruction

<table>
<thead>
<tr>
<th>Method</th>
<th>Chair</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IOU (↑)</td>
<td>CD (↓)</td>
</tr>
<tr>
<td>JLRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PQ-Net</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANISE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## ANISE vs other part-aware methods

*Single-view reconstruction*

<table>
<thead>
<tr>
<th>Method</th>
<th>Chair</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IOU (↑)</td>
<td>CD (↓)</td>
</tr>
<tr>
<td>JLRD</td>
<td>31.1</td>
<td></td>
</tr>
<tr>
<td>PQ-Net</td>
<td>47.0</td>
<td></td>
</tr>
<tr>
<td>ANISE</td>
<td>56.7</td>
<td></td>
</tr>
</tbody>
</table>
## ANISE vs other part-aware methods

*Single-view reconstruction*

<table>
<thead>
<tr>
<th>Method</th>
<th>Chair</th>
<th></th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IOU (↑)</td>
<td>CD (↓)</td>
<td>F1 (↑)</td>
</tr>
<tr>
<td>JLRD</td>
<td>31.1</td>
<td>5.09</td>
<td>29.4</td>
</tr>
<tr>
<td>PQ-Net</td>
<td>47.0</td>
<td>4.27</td>
<td>29.7</td>
</tr>
<tr>
<td>ANISE</td>
<td>56.7</td>
<td>2.99</td>
<td>57.4</td>
</tr>
</tbody>
</table>
# ANISE vs other part-aware methods

## Single-view reconstruction

<table>
<thead>
<tr>
<th>Method</th>
<th>Chair</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IOU (↑)</td>
<td>CD (↓)</td>
<td>F1 (↑)</td>
<td>IOU (↑)</td>
</tr>
<tr>
<td>JLRD</td>
<td>31.1</td>
<td>5.09</td>
<td>48.1</td>
<td>29.4</td>
</tr>
<tr>
<td>PQ-Net</td>
<td>47.0</td>
<td>4.27</td>
<td>50.2</td>
<td>29.7</td>
</tr>
<tr>
<td>ANISE</td>
<td>56.7</td>
<td>2.99</td>
<td>67.0</td>
<td>57.4</td>
</tr>
</tbody>
</table>
Application: part editing

Rec. shapes

Rec. shapes
Application: part editing

Rec. shapes  Edited Shapes  Rec. shapes
**Application:** part editing

Rec. shapes  |  Edited Shapes  |  Rec. shapes
Application: part editing
Application: part editing
Application: Part-constrained shape assembly

Input Collection of Shapes
Application: Part-constrained shape assembly
Application: Part-constrained shape assembly
Application: Part-constrained shape assembly

Input point cloud → Input Collection of Shapes → Output Shape
**Application**: Part-constrained shape assembly

**Input Collection of Shapes**

**Input point cloud**
Application: Part-constrained shape assembly
Application: Part-constrained shape assembly
Application: Part-constrained shape assembly
Ablation study: importance of full-shape supervision

w/o full shape supervision

Ground truth
Ablation study: importance of full-shape supervision
# Ablation study

<table>
<thead>
<tr>
<th>Full-shape supervision</th>
<th>CD (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>2.44</td>
</tr>
<tr>
<td>✓</td>
<td>1.69</td>
</tr>
</tbody>
</table>
# Ablation study

<table>
<thead>
<tr>
<th>Geometry conditioned on structure</th>
<th>Full-shape supervision</th>
<th>CD (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>2.81</td>
</tr>
<tr>
<td>✓</td>
<td>X</td>
<td>2.44</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>1.69</td>
</tr>
</tbody>
</table>
• Neural architecture for **part-aware implicit surface reconstruction**
• **SOTA performance** compared to prior methods
• Enables **part-based editing** and **part assembly from reference shapes**
Limitations

- Needs **part supervision & part-segmented datasets**
- Transformations are limited to translation and uniform scaling -- **no rotations**
- Coarse-to-fine synthesis, yet **no multiple levels of part hierarchies**
Thank you!

Acknowledgements:

Adobe

Our project web page:
https://lodurality.github.io/ANISE