Object-Centric Situational 3D Understanding

Leonidas Guibas
Stanford University
Towards Object Pose Estimation
Background: 3D Machine Learning
3D Needed for Navigation, Manipulation, AR/VR

Self-Driving Cars

source: Waymo

Augmented Reality

source: Microsoft HoloLens

Sometimes 2D is not enough ...
3D Point Clouds from Many Sensors

Lidar point clouds (LizardTech)

Structure from motion (Microsoft)

Depth camera (Intel, Microsoft, Google)
Deep Nets on 3D Point Cloud Data

• Close to raw sensor data
• Representationally simple
• Irregular
**Review: PointNet, PointNet++**

---

**End-to-end learning** for irregular point data

**Unified** framework for various tasks

Invariances

The model has to respect key desiderata for point clouds:

**Point Permutation Invariance**

Point cloud is a set of unordered points

**Spatial Transformation Invariance**

Point cloud rigid motions should not alter classification results

**Sampling Invariance**

Output a function of the underlying geometry and not the sampling
PointNet Basic Structure

Classification Network

Per point processing

Global symmetric aggregation

Segmentation Network
“Up-convolution” through 3D interpolation and/or pointnet.

Background: Object Data Sets and Object Knowledge

[Indoor Scene Understanding with RGB-D Images, S. Gupta et al.]
ShapeNet (>3M Models) https://www.shapenet.org/

Stanford:
Leonidas Guibas
Pat Hanrahan
Silvio Savarese

Princeton:
Tom Funkhouser
Jianxiong Xiao

UT Austin:
Qixing Huang
Object Knowledge: ShapeNet

Parts, symmetries, keywords, physical properties, materials, affordances, ...
Object Interaction Knowledge

Vector Field to Histograms

Vector Field $\mathbf{v}_i$ →

- Dilatation
- Shear Strain
- Vorticity
- Tensor Mag.
- Vector Mag.
- Vector Orien.
Crowdsourcing and Algorithmic Propagation

Integrated memory and processing

“Synapse” evolution
Initial Results: Part Annotation

~30,000 shapes
~90,000 parts

PartNet: Fine-Grained and Instance-Level Parts

PartNet: Fine-Grained Parts

- Subset of ShapeNetCore
  - 24 common indoor categories, 26,671 shapes, 573,585 parts
  - Avg 18 Part/shape, Max 230
- Human-annotated:
  - more fine-grained parts + instance-level parts
PartNet: Hierarchical and Consistent Parts

- Provide hierarchical segmentation of shapes: parts at multiple scales
- All shapes from the same category conform to a consistent part template
PartNet-Mobility: Annotate Part Motions (SAPIEN)

Objects in Context: Object Pose, Object Relationships
How to Simplify the Notion of Object Pose?
Abstraction and Canonicalization: Class-Specific Object Containers
Canonical “Containers” for Objects
NOCS: Normalized Object Coordinate Space

Plato

Canonicalize
- Position
- Orientation
- Size

RGB colors represent XYZ coordinates of shape.
Normalized Object Coordinate Spaces (NOCS)

- RGB colors represent XYZ coordinates of shape.
- Can be augmented with surface, mesh colors, affordance maps, or learned features.

NOCS Lifting Map, in lieu of Camera Pose Estimation

Interpretation

1. Partial shape & pose reconstruction of the object in NOCS
2. Dense correspondences from visible pixel to NOCS
3. Object-centered depth map
NOCS Category-Level Reasoning for Many Applications

Pose Estimation

Aggregation / Propagation

Synthesis
NOCS for 9D Object Pose and Size Estimation

Input: RGB-D Image + Category-Level CAD Model Repository

Output: 3 degrees Translation + 3 Rotation + 3 Size (9 DoF)

No object-specific CAD model
Category-level object pose can be defined for each category up to the limit of global symmetry in the category.
Exploit ShapeNet Category Co-Alignments

Abstract away, pose, size, some intra-category variation
Data: Point Clouds Complement RGB Images

- + High resolution
- + Dense coverage
- - Subject to many imaging artifacts

- + Absolute depth and scale
- - Sparse, low rez
An Image-Centric Approach: Build on Mask-RCNN
Method Overview

A Mask-RCNN-based backbone

Include depth info
Context-Aware MixEd ReAlity (CAMERA) Dataset

Context-Aware Mixed Reality Data Generation

Real Tabletop Scenes + Detected Planes → Composited RGB → Ground Truth Depth → Ground Truth Mask

ShapeNetCore

Indoor Lighting

Synthetic Objects

Ground Truth NOC Map
CAMERA Dataset

- **300K** mixed reality images are generated
  - 275K training
  - 25K validation

- **31 scenes** captured from IKEA as real backgrounds
  - 27 scenes for training
  - 4 scene for validation
  - 553 images

- **6 object categories**— bottle, bowl, camera, can, laptop and mug
  - 1085 models, 184 for validation

- Distractor objects
Real Dataset

- 8K RGB-D frames
  - training/validation/testing
  - 4300/950/2750
- 18 different real scenes
  - training/validation/testing
  - 7/5/6
- 42 unique instances
  - 7 per category
  - training/validation/testing
  - 3/1/3

Plus COCO images without pose annotation
Symmetry-aware loss function: 
\[ L_s = \min_{i=1,\ldots,|\theta|} L(\tilde{y}_i, y^*) \]
Results on Real Test Data

Training Data:
275K CAMERA
20K COCO (without NOCS)
4.3K real images

Performance:
3D IoU @ 50%: 76.4%
5°, 5cm: 10.2%
10°, 10cm: 23.1%
Qualitative Results: Real Data

Input

NOCs
ground truth

NOCs
prediction

6D pose + size 6D pose + size
ground truth

6D pose + size prediction
Voting Schemes: Object Detection and Pose Estimation
Via a voting scheme

GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES*

D. H. BALLARD
Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A.

(Received 10 October 1979; in revised form 9 September 1980; received for publication 23 September 1980)

Abstract—The Hough transform is a method for detecting curves by voting for a curve and parameters of that curve. It has been generalized to detect arbitrary shapes. (1.2) The algorithm is applied to binary edge-detection images. Specifically, it is shown how to use a shape-template, precomputed in an off-line step, to detect shapes among image features. (6) The algorithm is extended to cases of curved edges.


g. 1. Kinds of shapes detected with generalized Hough transform in binary images. (a) complex shapes.
Deep Hough Voting – A Two-Stage Approach

Input: point cloud → Seeds (XYZ + feature) → Votes (XYZ + feature) → Object center proposals

Vote clusters → Output: 3D bounding boxes

- table
- chair
VoteNet – A Two-Stage Approach

VotingNet

Input: point cloud

Seeds (XYZ + feature)

Votes (XYZ + feature)

Vote clusters

Output: 3D bounding boxes

Pointnet++

Pointnet++
Results on SUN RGB-D

Image of the scene

VotingNet prediction

Ground truth
Results on ScanNet

VotingNet prediction

Ground truth
**Quantitative Results**

<table>
<thead>
<tr>
<th>SUN RGB-D</th>
<th>Deep sliding shapes</th>
<th>Clouds of oriented gradients</th>
<th>Frustum pointnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSS [37]</td>
<td>Geo + RGB</td>
<td>44.2</td>
<td>78.8</td>
</tr>
<tr>
<td>COG [33]</td>
<td>Geo + RGB</td>
<td>58.3</td>
<td>63.7</td>
</tr>
<tr>
<td>2D-driven [17]</td>
<td>Geo + RGB</td>
<td>43.5</td>
<td>64.5</td>
</tr>
<tr>
<td>F-PointNet [30]</td>
<td>Geo + RGB</td>
<td>43.3</td>
<td>81.1</td>
</tr>
<tr>
<td>VotingNet (ours)</td>
<td>Geo only</td>
<td><strong>74.4</strong></td>
<td><strong>83.0</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>bathtub</th>
<th>bed</th>
<th>bookshelf</th>
<th>chair</th>
<th>desk</th>
<th>dresser</th>
<th>nightstand</th>
<th>sofa</th>
<th>table</th>
<th>toilet</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSS [37]</td>
<td>Geo + RGB</td>
<td>44.2</td>
<td>78.8</td>
<td>11.9</td>
<td>61.2</td>
<td>20.5</td>
<td>6.4</td>
<td>15.4</td>
<td>53.5</td>
<td>50.3</td>
<td>78.9</td>
<td>42.1</td>
</tr>
<tr>
<td>COG [33]</td>
<td>Geo + RGB</td>
<td>58.3</td>
<td>63.7</td>
<td>31.8</td>
<td>62.2</td>
<td><strong>45.2</strong></td>
<td>15.5</td>
<td>27.4</td>
<td>51.0</td>
<td><strong>51.3</strong></td>
<td>70.1</td>
<td>47.6</td>
</tr>
<tr>
<td>2D-driven [17]</td>
<td>Geo + RGB</td>
<td>43.5</td>
<td>64.5</td>
<td>31.4</td>
<td>48.3</td>
<td>27.9</td>
<td>25.9</td>
<td>41.9</td>
<td>50.4</td>
<td>37.0</td>
<td>80.4</td>
<td>45.1</td>
</tr>
<tr>
<td>F-PointNet [30]</td>
<td>Geo + RGB</td>
<td>43.3</td>
<td>81.1</td>
<td><strong>33.3</strong></td>
<td>64.2</td>
<td>24.7</td>
<td><strong>32.0</strong></td>
<td>58.1</td>
<td>61.1</td>
<td>51.1</td>
<td><strong>90.9</strong></td>
<td>54.0</td>
</tr>
<tr>
<td>VotingNet (ours)</td>
<td>Geo only</td>
<td><strong>74.4</strong></td>
<td><strong>83.0</strong></td>
<td>28.8</td>
<td><strong>75.3</strong></td>
<td>22.0</td>
<td>29.8</td>
<td><strong>62.2</strong></td>
<td><strong>64.0</strong></td>
<td>47.3</td>
<td>90.1</td>
<td><strong>57.7</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ScanNetV2</th>
<th>Input</th>
<th>mAP@0.25</th>
<th>mAP@0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSS [42, 12]</td>
<td>Geo + RGB</td>
<td>15.2</td>
<td>6.8</td>
</tr>
<tr>
<td>MRCNN 2D-3D [11, 12]</td>
<td>Geo + RGB</td>
<td>17.3</td>
<td>10.5</td>
</tr>
<tr>
<td>F-PointNet [34, 12]</td>
<td>Geo + RGB</td>
<td>19.8</td>
<td>10.8</td>
</tr>
<tr>
<td>GSPN [54]</td>
<td>Geo + RGB</td>
<td>30.6</td>
<td>17.7</td>
</tr>
<tr>
<td>3D-SIS [12]</td>
<td>Geo + 1 view</td>
<td>35.1</td>
<td>18.7</td>
</tr>
<tr>
<td>3D-SIS [12]</td>
<td>Geo + 3 views</td>
<td>36.6</td>
<td>19.0</td>
</tr>
<tr>
<td>3D-SIS [12]</td>
<td>Geo + 5 views</td>
<td>40.2</td>
<td>22.5</td>
</tr>
<tr>
<td>3D-SIS [12]</td>
<td>Geo only</td>
<td>25.4</td>
<td>14.6</td>
</tr>
<tr>
<td>VoteNet (ours)</td>
<td>Geo only</td>
<td><strong>58.6</strong></td>
<td><strong>33.5</strong></td>
</tr>
</tbody>
</table>
How to best combine geometry and appearance info?
Lifting 2D features

displacement and ray direction:

\[
\left( \frac{\Delta u}{f} z_1, \frac{\Delta v}{f} z_1, \frac{\overrightarrow{OC'}}{\|\overrightarrow{OC'}\|} \right).
\]
## Results on SUN RGB-D

<table>
<thead>
<tr>
<th>methods</th>
<th>RGB</th>
<th>bathtub</th>
<th>bed</th>
<th>bookshelf</th>
<th>chair</th>
<th>desk</th>
<th>dresser</th>
<th>nightstand</th>
<th>sofa</th>
<th>table</th>
<th>toilet</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSS [39]</td>
<td>✓</td>
<td>44.2</td>
<td>78.8</td>
<td>11.9</td>
<td>61.2</td>
<td>20.5</td>
<td>6.4</td>
<td>15.4</td>
<td>53.5</td>
<td>50.3</td>
<td>78.9</td>
<td>42.1</td>
</tr>
<tr>
<td>COG [34]</td>
<td>✓</td>
<td>58.3</td>
<td>63.7</td>
<td>31.8</td>
<td>62.2</td>
<td>45.2</td>
<td>15.5</td>
<td>27.4</td>
<td>51.0</td>
<td>51.3</td>
<td>70.1</td>
<td>47.6</td>
</tr>
<tr>
<td>2D-driven [15]</td>
<td>✓</td>
<td>43.5</td>
<td>64.5</td>
<td>31.4</td>
<td>48.3</td>
<td>27.9</td>
<td>25.9</td>
<td>41.9</td>
<td>50.4</td>
<td>37.0</td>
<td>80.4</td>
<td>45.1</td>
</tr>
<tr>
<td>PointFusion [43]</td>
<td>✓</td>
<td>37.3</td>
<td>68.6</td>
<td>37.7</td>
<td>55.1</td>
<td>17.2</td>
<td>23.9</td>
<td>32.3</td>
<td>53.8</td>
<td>31.0</td>
<td>83.8</td>
<td>45.4</td>
</tr>
<tr>
<td>F-PointNet [29]</td>
<td>✓</td>
<td>43.3</td>
<td>81.1</td>
<td>33.3</td>
<td>64.2</td>
<td>24.7</td>
<td>32.0</td>
<td>58.1</td>
<td>61.1</td>
<td>51.1</td>
<td><strong>90.9</strong></td>
<td>54.0</td>
</tr>
<tr>
<td><strong>VOTE</strong> [28]</td>
<td>✓</td>
<td>74.4</td>
<td>83.0</td>
<td>28.8</td>
<td>75.3</td>
<td>22.0</td>
<td>29.8</td>
<td>62.2</td>
<td>64.0</td>
<td>47.3</td>
<td>90.1</td>
<td>57.7</td>
</tr>
<tr>
<td><strong>+RGB</strong></td>
<td>✓</td>
<td>70.0</td>
<td>82.8</td>
<td>27.6</td>
<td>73.1</td>
<td>23.2</td>
<td>27.2</td>
<td>60.7</td>
<td>63.7</td>
<td>48.0</td>
<td>86.9</td>
<td>56.3</td>
</tr>
<tr>
<td><strong>+region feature</strong></td>
<td>✓</td>
<td>71.7</td>
<td>86.1</td>
<td>34.0</td>
<td>74.7</td>
<td>26.0</td>
<td>34.2</td>
<td>64.3</td>
<td>66.5</td>
<td>49.7</td>
<td>88.4</td>
<td>59.6</td>
</tr>
<tr>
<td><strong>IMVOTE</strong> [28]</td>
<td>✓</td>
<td><strong>75.9</strong></td>
<td><strong>87.6</strong></td>
<td><strong>41.3</strong></td>
<td><strong>76.7</strong></td>
<td>28.7</td>
<td><strong>41.4</strong></td>
<td><strong>69.9</strong></td>
<td><strong>70.7</strong></td>
<td>51.1</td>
<td>90.5</td>
<td><strong>63.4</strong></td>
</tr>
</tbody>
</table>
3D Detection with Sparse Points

**Application:** 3D detection from monocular video, using sparse SLAM keypoints.

*Picture: ORB-SLAM results*
Articulated Object Pose Estimation
Problem and Objective: Articulated Objects

**Input**
- 3D Point Cloud from a Single Depth Image
- Unknown Object
- Known Category

**Output**
- Segmentation
- Part Amodal Bounding Boxes
- Part Pose
- Joint States
- Joint Parameters/Limits

Representation: Parts and Whole H-NOCS

Articulation-Aware Normalized Coordinate Space Hierarchy

Normalized Articulated Object Coordinate Space (O-NOCS)

Normalized Part Coordinate Space (P-NOCS)

Normalize over the articulation as well
Canonicalization

Normalized Articulated Object Coordinate Space (O-NOCS)

- canonicalize articulations
- align
- cer scale
Representation: H-NOCS

Normalized Part Coordinate Space (O-NOCS)

Normalized Articulated Object Coordinate Space (O-NOCS)

Normalized Part Coordinate Space (P-NOCS)
Hierarchical Normalization in H-NOCS

Camera Space

O-NOCS

P-NOCS
Method: H-NOCS Network

- PointNet++ → O-NOCS
  - Joint Voting
- PointNet++ → Part Segmentation
  - P-NOCS
Joint parameter: revolute/prismatic axis orientation, revolute pivot location

Estimate the joint parameters in the O-NOCS, where the joints are in their canonical state

Use the predicted transformation between the O-NOCS and P-NOCS to convert the joint parameters to the P-NOCSs

Finally, use the predicted per-part pose to convert the joint parameters into camera space
<table>
<thead>
<tr>
<th>Eyeglasses</th>
<th>Oven</th>
<th>Washing Machine</th>
<th>Laptop</th>
<th>Drawer</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Eyeglasses" /></td>
<td><img src="image2.png" alt="Oven" /></td>
<td><img src="image3.png" alt="Washing Machine" /></td>
<td><img src="image4.png" alt="Laptop" /></td>
<td><img src="image5.png" alt="Drawer" /></td>
</tr>
<tr>
<td><img src="image6.png" alt="Eyeglasses" /></td>
<td><img src="image7.png" alt="Oven" /></td>
<td><img src="image8.png" alt="Washing Machine" /></td>
<td><img src="image9.png" alt="Laptop" /></td>
<td><img src="image10.png" alt="Drawer" /></td>
</tr>
<tr>
<td><img src="image11.png" alt="Eyeglasses" /></td>
<td><img src="image12.png" alt="Oven" /></td>
<td><img src="image13.png" alt="Washing Machine" /></td>
<td><img src="image14.png" alt="Laptop" /></td>
<td><img src="image15.png" alt="Drawer" /></td>
</tr>
<tr>
<td><img src="image16.png" alt="Eyeglasses" /></td>
<td><img src="image17.png" alt="Oven" /></td>
<td><img src="image18.png" alt="Washing Machine" /></td>
<td><img src="image19.png" alt="Laptop" /></td>
<td><img src="image20.png" alt="Drawer" /></td>
</tr>
<tr>
<td><img src="image21.png" alt="Eyeglasses" /></td>
<td><img src="image22.png" alt="Oven" /></td>
<td><img src="image23.png" alt="Washing Machine" /></td>
<td><img src="image24.png" alt="Laptop" /></td>
<td><img src="image25.png" alt="Drawer" /></td>
</tr>
<tr>
<td><img src="image26.png" alt="Eyeglasses" /></td>
<td><img src="image27.png" alt="Oven" /></td>
<td><img src="image28.png" alt="Washing Machine" /></td>
<td><img src="image29.png" alt="Laptop" /></td>
<td><img src="image30.png" alt="Drawer" /></td>
</tr>
<tr>
<td><img src="image31.png" alt="Eyeglasses" /></td>
<td><img src="image32.png" alt="Oven" /></td>
<td><img src="image33.png" alt="Washing Machine" /></td>
<td><img src="image34.png" alt="Laptop" /></td>
<td><img src="image35.png" alt="Drawer" /></td>
</tr>
</tbody>
</table>
Results: Pose Estimation

- **Rotation Error** (lower is better)
  - Part 0: 9.7
  - Part 1: 5.1
  - Part 2: 3.7

- **Translation Error**
  - Part 0: 0.035
  - Part 1: 0.061
  - Part 2: 0.057

- **3D IoU** (higher is better)
  - Part 0: 86.9
  - Part 1: 87.4
  - Part 2: 40.5

*Synthetic Test Data*
Our method achieves high-accuracy joint parameter estimation.
Results: Real Instance-Level Pose Estimation

RGB reference  Predicted Pose  Labeled GT Pose
Temporal-Consistent Pose and Size Tracking
Actuator-Centric Human Object Interaction Capture
Wearable Inside-Out Capture Gantry

✓ **Inside-out viewpoint**
  • Not restricted to **studio** environment

✓ **Wearable**
  • **Moves** with users, sees objects being manipulated

✓ **Multiple cameras**
  • Robust to **occlusions**

✓ **Lower cost**
  • 3D printed prototype costs around **$500** (vs. >$10,000 for outside-in capture)

✓ **Use cases**
  • Tracking hands in **close interaction** with objects
  • Acquiring large in-the-wild dataset
The Prototype

- 5 cameras
- 120 FPS
- 640 x 480px
Data Acquisition

Main Idea

• No way to annotate large-scale egocentric hand interaction data.

• Develop robust hand shape and pose from these in-the-wild images.
NOCS Maps Prediction

Input RGB

Ground Truth

Prediction
NOCS View Aggregation
From Single to Multiple NOCS Lifts

U-Net Capture Algorithm

Input: Multi-view RGB video

NOCS as a view accumulator space

Output: Hand shape, pose, and camera pose

Pose-Normalized NOCS
Summary

The power of aggregation

The power of abstraction/canonicalization
Acknowledgements

• Collaborators:
  • Current/past students: Jingwei Huang, Kaichun Mo, Charles Qi, Davis Rempe, He Wang, Eric Yi
  • Current/past postdocs: Vova Kim, Or Litany, Soeren Pirk, Srinath Sridhar
  • Senior: Sofien Bouaziz, Xinlei Chen, Kaiming He, Shuran Song, Hao Su, Jules Valentin

Sponsors:
Thank You

guibas@cs.stanford.edu