Learning Keypoint Representations for Robot Manipulation

Yuke Zhu

IROS 2019
Key to generalization across objects

Find an abstract representation that can be shared by a family of objects
A broad range of object representations

- 6D Object Pose
  - lack of details
  - specific to instance
  - computational cheap
  - (relatively) easy to estimate

- Full 3D Model
  - geometric details
  - generic to object
  - computational expensive
  - difficult to estimate
6D Object Pose

6D pose and 3D model are two ends of a spectrum of point-based representations

Full 3D Model

2 points

35,947 points (vertices)

sparse
dense
Key idea

A handful of discriminative 3D keypoints as a compact and effective object representation.
Keypoint Representations

Compact and discriminative
[SIFT, Lowe 2004]

Handles occlusions and deformations
[KeypointNet, Suwajanakorn et al. 2018]

Robust towards object variations
[Zhou et al. 2018]

Informative for robot control
[kPAM, Manuelli et al. 2019]
Keypoint Representations

Problem:
Annotating 3D keypoints is tedious and ambiguous.

Our Solution:
Learning task-specific keypoints without manual labeling.

- Compact and discriminative
  [SIFT, Lowe 2004]

- Handles occlusions and deformations
  [KeypointNet, Suwajanakorn et al. 2018]

- Informative for robot control
  [kPAM, Manuelli et al. 2019]

- Robust towards object variations
  [Zhou et al. 2018]
6-Pack: Category-level 6D Pose Tracker with Anchor-Based Keypoints

Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv,
Cewu Lu, Li Fei-Fei, Silvio Savarese, Yuke Zhu
6D Object Pose Estimation

[ΔT; ΔR]

camera coordinate

object coordinate

Applications

Activity understanding

Motion planning

Augmented Reality
However, model-based 6D tracker assumes **known 3D model** of the object and fails to generalize to **unseen objects**.

**Related Work**

**Traditional methods**
- Hinterstoisser et al. ACCV’ 12
- Choi & Christensen IROS’ 13
- Collet et al. ICRA’ 11
- Lepetit et al. TPAMI’ 04

More robust against occlusion and illumination changes

**Learning-based methods**
- DOPE [Tremblay et al. CoRL’18]
- PoseCNN [Xiang et al. RSS’18]
- PoseRBPF [Deng et al. RSS’19]
- DeepIM [Li et al. ECCV’ 18]
- DenseFusion [Wang et al. CVPR’19]

[Image: DenseFusion, Wang et al. CVPR’19]
Pose estimation without known model?

Category-level canonical pose
Category-Level 6D Pose Estimation

**Training**
- Laptop Category
- Synthetic data with ShapeNetCore** models (90% of the training data)

**Testing**
- Real data with 3 objects (10% of the training data)
- Real data with unseen objects

* Wang et al. 2019, “Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation” CVPR2019
Category-Level 6D Pose Tracking

6D Pose Estimation
- Input: image frame
- Output: 6D pose

6D Pose Tracking
- Input: video + initial bbox
- Output: 6D pose
Category-Level 6D Pose Tracking

unseen object from training category

6D pose estimation of current frame

\[ P_t = P_0[\Delta R | \Delta T]_1 \cdots [\Delta R | \Delta T]_{t-1} \]
6-PACK: 6D Pose Anchor-based Category-level Keypoint tracker
6-PACK: 6D Pose Anchor-based Category-level Keypoint tracker

Challenge: No ground-truth object model

Idea: Use anchor box as a scaffold
each anchor point captures local information

3D anchors around the previous pose
6-PACK: 6D Pose Anchor-based Category-level Keypoint tracker

**Challenge:** Incorporate temporal information

**Idea:** Use motion model to predict the likely position of the object

- Selected anchor
- 3D anchors around the next (likely) pose
**6-PACK**: 6D Pose Anchor-based Category-level Keypoint tracker

**Challenge**: Generate 3D keypoints

**Idea**: Predict offsets from selected anchor
**6-PACK**: **6D Pose** Anchor-based **Category-level** Keypoint tracker

**Current Frame**
- Generated keypoints

**Previous Frame**
- Compute relative pose w/ matching keypoints

$[\Delta T; \Delta R]$
Our Pipeline

RGB-D$_{t-1}$

6PACK

keypoints

$\Delta R$, $\Delta t$

Pose$_t$

Output at time $t$

Input at time $t$

RGB-D$_t$

6PACK

Annotating 3D keypoints is hard!
Our Pipeline

Input at time $t$

$\text{RGB-D}_{t-1}$ → $\text{6PACK}$ → Pose$_{t-2}$

$\text{RGB-D}_t$ → $\text{6PACK}$ → keypoints

End-to-end training from pose to keypoints

$\Delta R, \Delta t$

Output at time $t$

Pose$_t$
## Evaluation Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>bottle</td>
<td>5°5cm</td>
<td>5.5</td>
<td>10.1</td>
<td>5.9</td>
<td>23.7</td>
</tr>
<tr>
<td></td>
<td>IoU25</td>
<td>48.7</td>
<td>29.9</td>
<td>23.1</td>
<td>92.0</td>
</tr>
<tr>
<td></td>
<td>R_err</td>
<td>25.6</td>
<td>48.0</td>
<td>28.5</td>
<td>15.7</td>
</tr>
<tr>
<td></td>
<td>T_err</td>
<td>14.4</td>
<td>15.7</td>
<td>9.5</td>
<td>4.2</td>
</tr>
<tr>
<td>bowl</td>
<td>5°5cm</td>
<td>62.2</td>
<td>40.3</td>
<td>16.8</td>
<td>53.0</td>
</tr>
<tr>
<td></td>
<td>IoU25</td>
<td>99.6</td>
<td>79.7</td>
<td>74.7</td>
<td>74.7</td>
</tr>
<tr>
<td></td>
<td>R_err</td>
<td>4.7</td>
<td>19.0</td>
<td>9.8</td>
<td>5.3</td>
</tr>
<tr>
<td></td>
<td>T_err</td>
<td>1.2</td>
<td>4.7</td>
<td>8.2</td>
<td>1.6</td>
</tr>
<tr>
<td>camera</td>
<td>5°5cm</td>
<td>0.6</td>
<td>12.6</td>
<td>1.8</td>
<td>8.4</td>
</tr>
<tr>
<td></td>
<td>IoU25</td>
<td>90.6</td>
<td>53.1</td>
<td>30.9</td>
<td>91.0</td>
</tr>
<tr>
<td></td>
<td>R_err</td>
<td>33.8</td>
<td>80.5</td>
<td>45.2</td>
<td>43.9</td>
</tr>
<tr>
<td></td>
<td>T_err</td>
<td>3.1</td>
<td>12.2</td>
<td>8.5</td>
<td>5.5</td>
</tr>
<tr>
<td>can</td>
<td>5°5cm</td>
<td>7.1</td>
<td>17.2</td>
<td>4.3</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td>IoU25</td>
<td>77.0</td>
<td>40.5</td>
<td>42.6</td>
<td>89.9</td>
</tr>
<tr>
<td></td>
<td>R_err</td>
<td>16.9</td>
<td>47.1</td>
<td>28.8</td>
<td>12.5</td>
</tr>
<tr>
<td></td>
<td>T_err</td>
<td>4.0</td>
<td>9.4</td>
<td>13.1</td>
<td>5.0</td>
</tr>
<tr>
<td>laptop</td>
<td>5°5cm</td>
<td>25.5</td>
<td>14.8</td>
<td>49.2</td>
<td>62.4</td>
</tr>
<tr>
<td></td>
<td>IoU25</td>
<td>94.7</td>
<td>50.9</td>
<td>94.6</td>
<td>97.8</td>
</tr>
<tr>
<td></td>
<td>R_err</td>
<td>8.6</td>
<td>37.7</td>
<td>6.5</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td>T_err</td>
<td>2.4</td>
<td>9.2</td>
<td>4.4</td>
<td>2.5</td>
</tr>
<tr>
<td>mug</td>
<td>5°5cm</td>
<td>0.9</td>
<td>6.2</td>
<td>3.1</td>
<td>22.4</td>
</tr>
<tr>
<td></td>
<td>IoU25</td>
<td>82.8</td>
<td>27.7</td>
<td>52.0</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>R_err</td>
<td>31.5</td>
<td>56.3</td>
<td>61.2</td>
<td>20.3</td>
</tr>
<tr>
<td></td>
<td>T_err</td>
<td>4.0</td>
<td>9.2</td>
<td>6.7</td>
<td>1.8</td>
</tr>
<tr>
<td>Overall</td>
<td>5°5cm</td>
<td>17.0</td>
<td>16.9</td>
<td>13.5</td>
<td>32.5</td>
</tr>
<tr>
<td></td>
<td>IoU25</td>
<td>82.2</td>
<td>47.0</td>
<td>53.0</td>
<td>39.4</td>
</tr>
<tr>
<td></td>
<td>R_err</td>
<td>20.2</td>
<td>48.1</td>
<td>30.0</td>
<td>17.1</td>
</tr>
<tr>
<td></td>
<td>T_err</td>
<td>4.9</td>
<td>10.5</td>
<td>8.4</td>
<td>3.4</td>
</tr>
</tbody>
</table>

6-PACK’s 6D pose tracking accuracy is still higher than NOCS for more than 12%. 
Qualitative Evaluation Results

Keypoints generation results
(Matchings of each keypoint from different view)

Qualitative results
(Red bounding box refers to pose error larger than 5cm 5°)
Real-time testing results on unseen objects

Tracker runs at 10 fps on an NVIDIA GTX1070 GPU
Summary

- 3D keypoints are compact object representations for 6D tracking
- End-to-end learning without manual keypoint annotations
- Real-time category-level tracking for robot interaction

Code: github.com/j96w/6PACK
KETO: Learning Keypoint Representations for Tool Manipulation

Zengyi Qin, Kuan Fang, Yuke Zhu, Li Fei-Fei, Silvio Savarese
Vision-Based Tool Manipulation

- Recognizing tools
- Understanding tools
- Manipulating tools
Vision-Based Tool Manipulation

[hammering]

[Fang, Zhu, et al., RSS’18]
Vision-Based Tool Manipulation

Tool manipulation as a two-stage process

- **Stage 1**: Grasp an object as a tool.
- **Stage 2**: Use the grasped tool to complete the goal of the task.

[Fang, Zhu, et al., RSS’18]
Vision-Based Tool Manipulation

visual observation \rightarrow representation \rightarrow action
Vision-Based Tool Manipulation

- High-dimensionality
- Lack of interpretability

[Fang, Zhu, et al., RSS’18]
**KETO**: Keypoint Representations for Tool Manipulation

![Diagram of tool manipulation with keypoints labeled as learned and fixed.](image-url)
**KETO**: Keypoint Representations for Tool Manipulation

- **Environment keypoints**
- **Effect point**
- **Function point**
- **Grasp point**
**KETO: Keypoint Representations for Tool Manipulation**

For **hammering**

1. $x_t$ is close to $x_f$
2. Direction of $v$ aligns with $z$.

\[
\max_p \ v^T z - \|x_f - x_t\|^2
\]

Solving the optimal pose of object as a QP
KETO: Keypoint Representations for Tool Manipulation

- **Visual Input**
  - Keypoint Generator
  - Keypoints
  - Action Optimizer

- **Action Execution**
  - a) Initial
  - b) Grasping
  - c) Manipulation
  - d) Completion
**KETO**: Keypoint Representations for Tool Manipulation

Decoupling perception problem and control problem
**KETO**: Keypoint Representations for Tool Manipulation

How do we get supervision?

Self-supervision from interaction
**KETO: Keypoint Representations for Tool Manipulation**

Visual input → Keypoint generator → Keypoints → Action optimizer → Action execution

- a) Initial
- b) Grasping
- c) Manipulation
- d) Completion

Training data:
- Positive examples
- Negative examples

Model update → Success / Failure
Procedural Generation of Tools

<table>
<thead>
<tr>
<th>Hammers</th>
<th>Non-hammers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>Test</td>
</tr>
</tbody>
</table>
Tool Manipulation Tasks

(a) Hammering

(b) Pushing

(c) Reaching
Results: Hammering Task
Results: Pushing Task
Results: Reaching Task

- Grasp point $x_g$
- Function point $x_f$
- Effect point $x_e$
Keypoints as intermediate representations of tools are effective.
Results: Keypoint Prediction
Composite Task: Multi-stage Tool Use

autonomous execution
Tool Creation: MacGyvering

Improvising tools for inventive problem solving

[Nair, Shrivatsav, Erickson, Chernova RSS'19]
Tool Creation: MacGyvering

Keypoints provides a scaffold for generating tools from object parts.
Hammering with the Created Tool
Hierarchical Planning with Cascade Variational Inference

\[ z \sim p(z) \]
\[ a \sim g(\cdot | s, c, z) \]
\[ c \sim p(c) \]
\[ s' \sim h(\cdot | s, c) \]

Action Generator

Dynamics

Meta-Dynamics

Fang, Zhu, Garg, Savarese, Fei-Fei. CoRL'19
Hierarchical Planning with Cascade Variational Inference

Move away obstacles

Push target object to the goal

Fang, Zhu, Garg, Savarese, Fei-Fei. CoRL’19
Conclusions

3D Keypoints are compact and effective object representations for manipulation.

1. 6-PACK: Keypoints for Category-level 6D Pose Tracking

2. KETO: Keypoints for Vision-based Tool Manipulation

Supervision can be acquired through object motion and robot interaction.

Open Question

How to integrate keypoints with other representations to incorporate fine-grained semantic, geometric, and physical information of objects and environments.
Questions?