Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects

by Tremblay, To, Sundaralingam, Xiang, Fox, Birchfield

Christian M. Salamut

08.12.2022

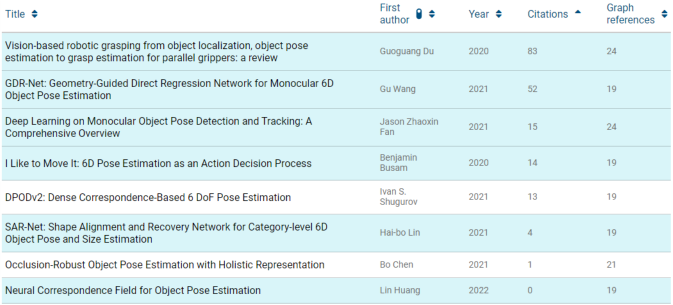
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Overview

1. Introduction
2. Contributions
3. Methods
   - Domain Randomization
   - Photorealistic Data
   - Deep Object Pose Estimation (DOPE)
4. Experiments and Results
5. State-of-the-Art
   - State-of-the-Art 6-DOF Object Detection
6. Conclusion
7. References

Paper Statistics

- published in year 2018
- 2nd Conference on Robot Learning (CoRL 2018), Zurich, Switzerland.
- people associated with NVIDIA
- around 500 citations



Title	First author	Year	Citations	Graph references
Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review	Guoguang Du	2020	83	24
GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation	Gu Wang	2021	52	19
Deep Learning on Monocular Object Pose Detection and Tracking: A Comprehensive Overview	Jason Zhaoxin Fan	2021	15	24
I Like to Move It: 6D Pose Estimation as an Action Decision Process	Benjamin Busam	2020	14	19
DPODv2: Dense Correspondence-Based 6 DoF Pose Estimation	Ivan S. Shugurov	2021	13	19
SAR-Net: Shape Alignment and Recovery Network for Category-level 6D Object Pose and Size Estimation	Hai-bo Lin	2021	4	19
Occlusion-Robust Object Pose Estimation with Holistic Representation	Bo Chen	2021	1	21
Neural Correspondence Field for Object Pose Estimation	Lin Huang	2022	0	19

Figure: Papers that cited DOPE in blue-underlaid. [1]

2-Dimensional Object Detection

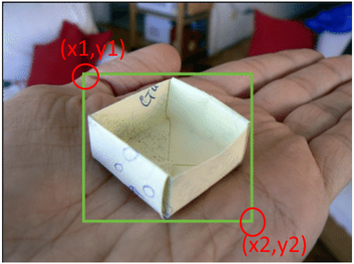
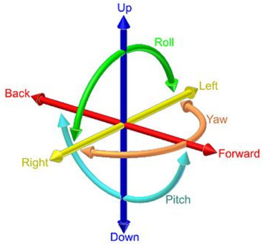


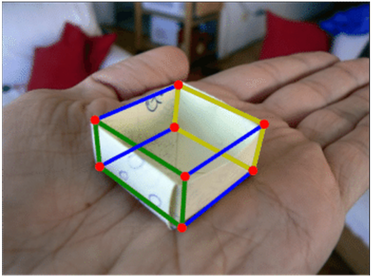
Figure: 2-D Object Annotation [2]

- rectangular bounding box shape
- created using two coordinates on the image

6 DOF / 3-D Object Detection



(a) 6-Degree of Freedom



(b) 3-Dimensional Object Annotation [2]

- three dimensions (x, y, z axes); plus three rotational axes (roll, pitch, yaw) (a)
- 8 vertices needed (b)

Sum up 2-D to 3-D

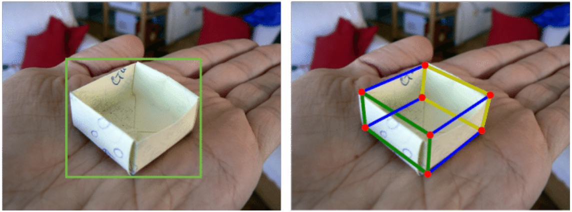


Figure: 2-D to 3-D [2].

YCB Objects

- Yale-CMU-Berkeley (YCB)
- daily life objects

Figure: YCB objects [3].

Motivation

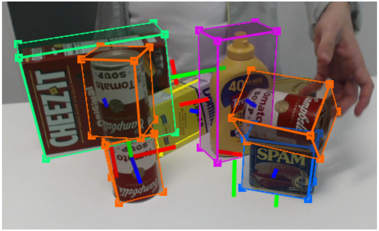


Figure: [4]

- find meshes in a coordinate system
- find pose (position and orientation) relative to the camera
- implicit representations of the above

Reality Gap

- lighting conditions
- noise (e.g. camera)
- richness of images (e.g. background textures)
- performance issues on real data

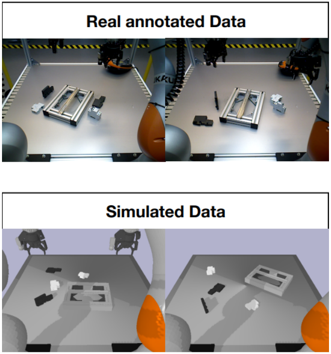


Figure: Simulated vs. Real Data [5].

Contributions

- propose photorealistic data for training
- combined with Domain Randomization
- propose DOPE (Deep Object Pose Estimation) algorithm

Domain Randomization

- existing method dealing with reality gap
- random camera positions
- lighting conditions
- objects positions
- non-realistic textures
- distractor objects

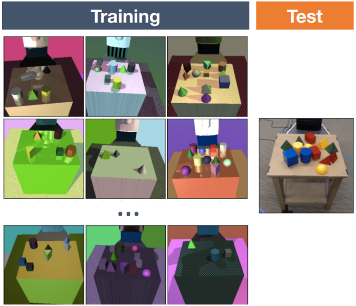


Figure: Domain Randomization. Train and Test data. [6].

Photorealistic Data

- placing the foreground objects in 3D background scenes with physical constraints
- standard backgrounds from UnrealEngine4
- YCB Objects
- allowed to fall and to collide within the scene
- changing camera position while falling
- Falling Things (FAT) data set

Figure: Photorealistic Data [4].

DOPE Framework

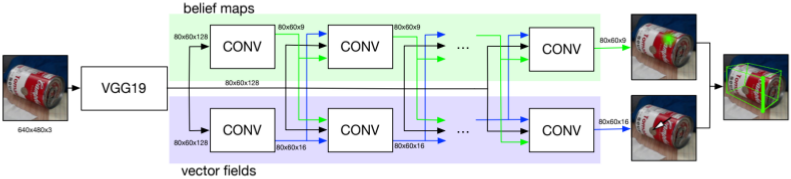


Figure: DOPE Network Architecture [7].

- Image features computed VGG-19 network
- build belief maps (8 vertices + 1 centroid)
- build 8 vector fields directing to centroid of an object
- process is done in multiple stages (field of reception)

Perspective-n-Point Extraction

- find local peaks in the belief maps above a threshold
- evaluate vector field direction and assign to closest centroid (angular threshold)
- estimate 6-DOF using a PnP algorithm (Rotation and Translation matrix)

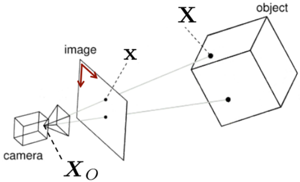


Figure: PnP-Problem [8].

Training

- 60k domain-randomized image frames mixed with 60k photorealistic image frames
- 60k DR (class specific) + 60k photorealistic
- calculate L2 for belief maps and vector fields after each stage

Average Distance Metric (ADD)

- ADD and ADD matching score

$$\text{ADD} = \frac{1}{|M|} \sum_{x \in M} ||(\mathbf{Rx} + \mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{T}})|| \qquad (1)$$

$$\text{ADD-S} = \frac{1}{|M|} \sum_{x_1 \in M} \min_{x_2 \in M} ||(\mathbf{Rx_1} + \mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x_2} + \tilde{\mathbf{T}})|| \qquad (2)$$

- **R** and **T** are **ground truth** rotation and translation matrices
- $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{T}}$ are **estimated** rotation and translation matrices

Results

- authors limited to cracker box, sugar box, tomato soup can, mustard bottle and potted meat
- reasons: graspability and texture
- threshold 2cm

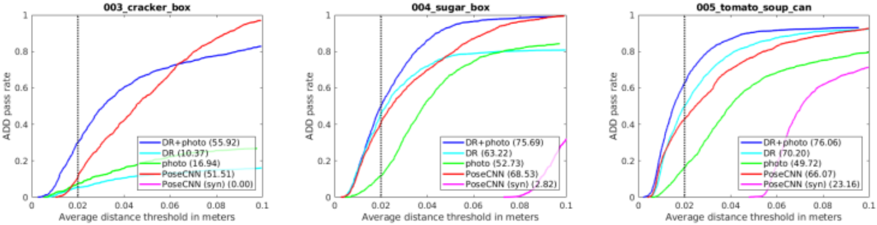


Figure: Accuracy results cracker box, sugar box, tomato soup can. [4].

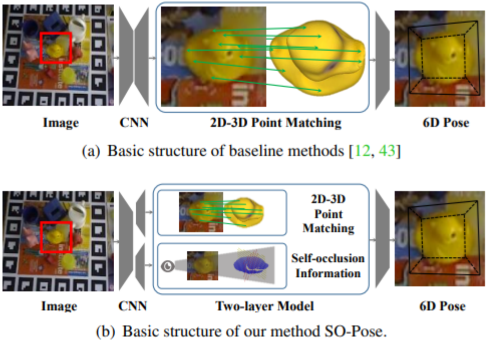
Live Demo

Figure: Live demo video [9].

Discussion

- not provided complete results
- limited to certain items (best results)
- no result table
- grasp items from different angles

Self-Occlusion Pose Intuition

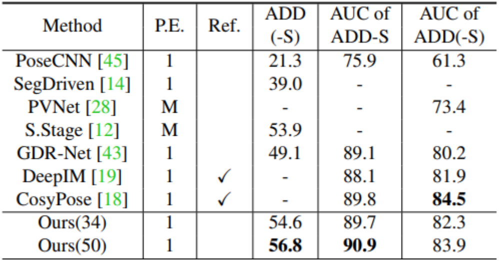


(a) Basic structure of baseline methods [12, 43]

(b) Basic structure of our method SO-Pose.

Figure: Basic Self-Occlusion-Pose [10].

State-of-the-Art 6-DOF Object Detection



Method	P.E.	Ref.	ADD (-S)	AUC of ADD-S	AUC of ADD(-S)
PoseCNN [45]	1		21.3	75.9	61.3
SegDriven [14]	1		39.0	-	-
PVNet [28]	M		-	-	73.4
S.Stage [12]	M		53.9	-	-
GDR-Net [43]	1		49.1	89.1	80.2
DeepIM [19]	1	✓	-	88.1	81.9
CosyPose [18]	1	✓	-	89.8	**84.5**
Ours(34)	1		54.6	89.7	82.3
Ours(50)	1		**56.8**	**90.9**	83.9

Figure: Results on YCB-V [10]

- ADD(-S) percentage of transformed model points whose deviation from ground truth lies below 10% of the object's diameter (0.1d).
- For symmetric objects, ADD(-S) measures the deviation to the closet model point [10]
- Area Under the Curve accuracy when using different thresholds

Conclusion

- YCB objects is a robotics data set widely used
- domain randomization is good
- but photorealistic data improves the results a lot
- DOPE is a model for predicting 6-DOF poses
- SO-Pose state-of-the-art

End Frame

Motivation

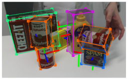


Figure: [4]

- find meshes in a coordinate system
- find pose (position and orientation) relative to the camera
- implicit representations of the above

Photorealistic Data

- placing the foreground objects in 3D background scenes with physical constraints
- standard backgrounds from UnrealEngine4
- YCB Objects
- allowed to fall and to collide within the scene
- changing camera position while falling
- Falling Things (FAT) data set

Figure: Photorealistic Data [4].

DOPE Framework

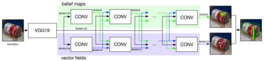


Figure: DOPE Network Architecture [7].

- Image features computed VGG-19 network
- build belief maps (8 vertices + 1 centroid)
- build 8 vector fields directing to centroid of an object
- process is done in multiple stages (field of reception)

Results

- authors limited to cracker box, sugar box, tomato soup can, mustard bottle and potted meat
- reasons: graspability and texture
- threshold 2cm

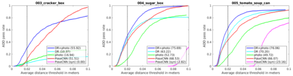


Figure: Accuracy results cracker box, sugar box, tomato soup can. [4].

[1] *Find and explore academic papers*. URL: https://www.connectedpapers.com/.

[2] Prash. *2D/ 3D object detection-bounding boxes-in the Autonomous Systems Domain*. May 2021. URL: https://simpleprash.medium.com/2d-3d-object-detection-bounding-boxes-in-the-autonomous-systems-domain-a6f867aa14a0.

[3] Berk Calli et al. "Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set". In: *IEEE Robotics & Automation Magazine* 22.3 (2015), pp. 36–52. DOI: 10.1109/MRA.2015.2448951.

[4] Jonathan Tremblay et al. "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects". In: *CoRR* abs/1809.10790 (2018). arXiv: 1809.10790. URL: http://arxiv.org/abs/1809.10790.

[5] *Reality gap*. URL: https://wiki.tum.de/display/ldv/Reality+Gap.

[6] Joshua Tobin et al. "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World". In: *CoRR* abs/1703.06907 (2017). arXiv: 1703.06907. URL: http://arxiv.org/abs/1703.06907.

[7] Umesh Thillaivasan. *Real world 3D object pose estimation and the Sim2Real gap*. 2021. URL: http://cs230.stanford.edu/projects_fall_2021/reports/103058918.pdf.

[8] URL: https://www.ipb.uni-bonn.de/photo12-2021/.

[9] nvidia. *Research at Nvidia: Deep object pose estimation for semantic robotic grasping of household objects*. Oct. 2018. URL: https://www.youtube.com/watch?v=yVGViBqWtBI.

[10] Yan Di et al. "SO-Pose: Exploiting Self-Occlusion for Direct 6D Pose Estimation". In: *CoRR* abs/2108.08367 (2021). arXiv: 2108.08367. URL: https://arxiv.org/abs/2108.08367.

[11] In: (). URL: https://tams.informatik.uni-hamburg.de/people/goerner/images/dope%202019-08-22%2013-13-15.png.

[12] Mei Jin, Jiaqing Li, and Liguo Zhang. "DOPE++: 6D pose estimation algorithm for weakly textured objects based on deep neural networks". In: *PLOS ONE* 17.6 (June 2022), pp. 1–21. DOI: 10.1371/journal.pone.0269175. URL: https://doi.org/10.1371/journal.pone.0269175.

Last visited all links on 04.12.2022.

Distractors

number and types of distractors, selected from a set of 3D models (cones, pyramids, spheres, cylinders, partial toroids, arrows, etc.)

Complete Results Accuracy curves

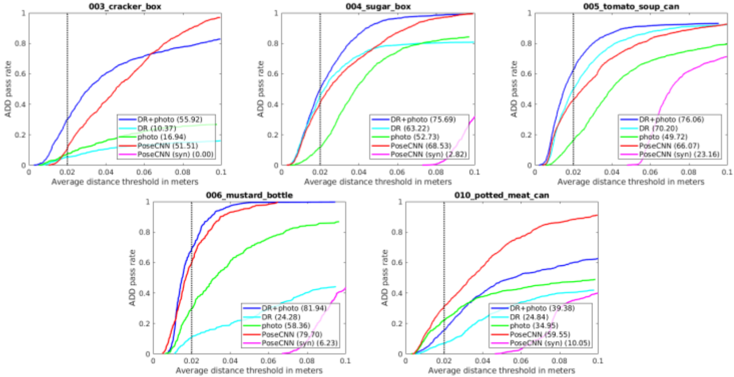


Figure: [4]

True Belief Maps, True Vector Fields

- "The ground truth belief maps were generated by placing 2D Gaussians at the vertex locations with SIGMA = 2 pixels"
- "The ground truth vector fields were generated by setting pixels to the normalized x- and y-components of the vector pointing toward the object's centroid " [4]
- => Assume Generate True locations on 400x400x8 and downsample to 50x50x8
- assume W=400, H=400. Vector from one vertex to centroid (x=100, y=200). Normalized:(0.25,0.5)

Results visualized

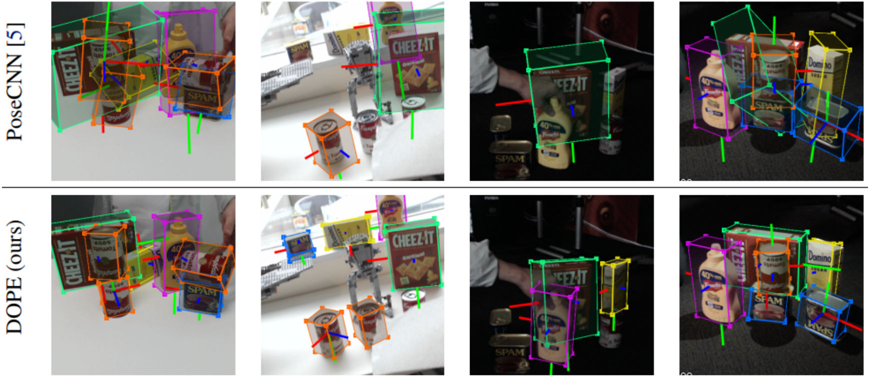
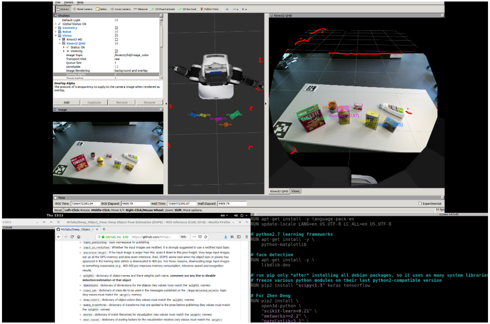


Figure: Pose estimation of YCB objects. Different lighting conditions. [4].

Live DOPE at TAMS



Kinect2 QHD
Views
Time
ROS Time: 1566472395.04
ROS Elapsed: 4409.79
Wall Time: 1566472395.07
Wall Elapsed: 4409.79
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.
□ Experimental
31 fps
Thu 13:13
NVlabs/Deep_Object_Pose: Deep Object Pose Estimation (DOPE) – ROS inference (CoRL 2018) - Mozilla Firefox
GitHub, Inc. (US)
https://github.com/NVlabs/
90%

- topic_publishing : topic namespace for publishing
- input_is_rectified : Whether the input images are rectified. It is strongly suggested to use a rectified input topic.
- downscale_height : If the input image is larger than this, scale it down to this pixel height. Very large input images eat up all the GPU memory and slow down inference. Also, DOPE works best when the object size (in pixels) has appeared in the training data (which is downscaled to 400 px). For these reasons, downscaling large input images to something reasonable (e.g., 400-500 px) improves memory consumption, inference speed *and* recognition results.
- weights : dictionary of object names and there weights path name, **comment out any line to disable detection/estimation of that object**
- dimensions : dictionary of dimensions for the objects (key values must match the weights names)
- class_ids : dictionary of class ids to be used in the messages published on the /dope/detected_objects topic (key values must match the weights names)
- draw_colors : dictionary of object colors (key values must match the weights names)
- model_transforms : dictionary of transforms that are applied to the pose before publishing (key values must match the weights names)
- meshes : dictionary of mesh filenames for visualization (key values must match the weights names)
- mesh_scales : dictionary of scaling factors for the visualization meshes (key values must match the weights

```
RUN apt-get install -y language-pack-en
RUN update-locale LANG=en_US.UTF-8 LC_ALL=en_US.UTF-8

# python2.7 learning frameworks
RUN apt-get install -y \
    python-matplotlib

# face detection
RUN apt-get install -y \
    libdlib-dev

# run pip only *after* installing all debian packages, so it uses as many system libraries
# freeze various python modules on their last python2-compatible version
RUN pip2 install "scipy<1.3" keras tensorflow

# For Zhen Deng
RUN pip2 install \
    open3d-python \
    "scikit-learn<0.21" \
    "networkx==2.2" \
    "matplotlib<3.1" \
```

Figure: Provided by Michael Görner. [11]

Dope Pick and Place

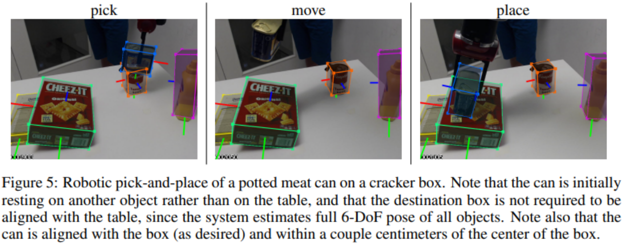


Figure 5: Robotic pick-and-place of a potted meat can on a cracker box. Note that the can is initially resting on another object rather than on the table, and that the destination box is not required to be aligned with the table, since the system estimates full 6-DoF pose of all objects. Note also that the can is aligned with the box (as desired) and within a couple centimeters of the center of the box.

Figure: Pick and place [4].

DOPE++

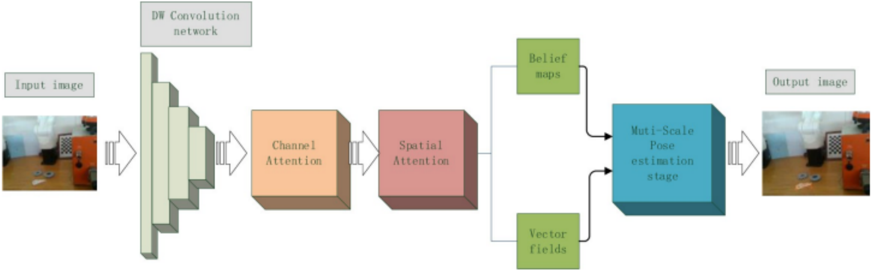


Figure: DOPE++ [12].

SO-Pose Architecture

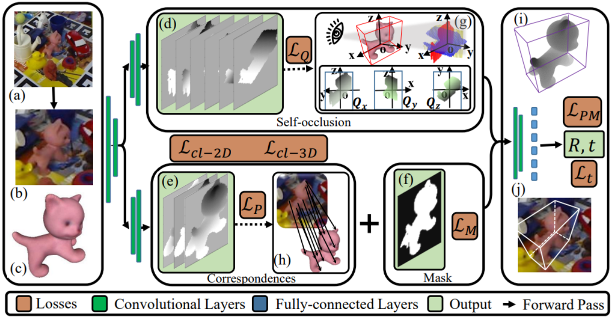


Figure: Self-Occlusion Pose Architecture [10].

Planar Non-planar

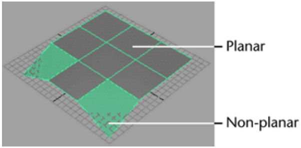


Figure: src knowledge.autodesk.com