Deep Image Processing for Object Pose Estimation

PoseCNN and Deep Object Pose Estimation (DOPE)

Marcus Rottschäfer

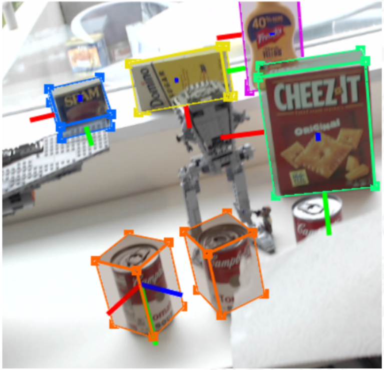
University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Technical Aspects of Multimodal Systems

11. June 2020

What is Object Pose Estimation good for?

- Estimate the 6D pose of objects from an image
- With the 6-DoF pose we can perform robotic manipulation
- Awareness of the surrounding: 3D position and orientation of objects in the environment
- pick-and-place, hand-over from a person, imitation learning



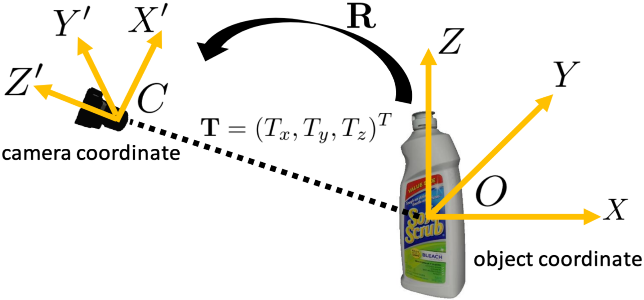
Tremblay et al. 2018

Outline

1. Motivation
2. **Foundations of Object Pose Estimation**
3. PoseCNN
4. Deep Object Pose Estimation (DOPE)
5. Conclusion

What is Object Pose Estimation?

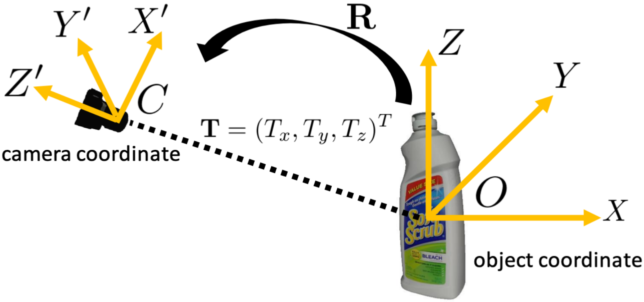
- We are talking about 6D Object Pose Estimation
- Find the 6-DoF (degrees of freedom) pose:



What is estimated in 6D pose estimation? Inspired by Xiang et al. 2018.

What is Object Pose Estimation?

- We are talking about 6D Object Pose Estimation
- Find the 6-DoF (degrees of freedom) pose:



What is estimated in 6D pose estimation? Inspired by Xiang et al. 2018.

- (Typically from a set of predefined object categories)

Two approaches to Object Pose Estimation

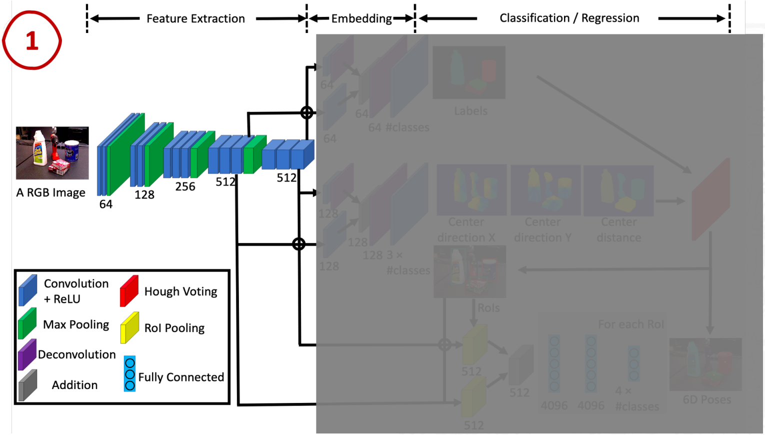
Methods can be roughly classified into two approaches (Xiang et al. 2018):

- **Template-based approaches:**
  - Create a template (e.g. 2D render of 3D object model) and match it to different regions in the image
  - Use ideas from 2D object detection (matches) and augment to 6D (e.g. YOLO or SSD for 6D)
  - Works good with texture-less objects, bad with occlusions between objects!
- **Feature-based approaches:**
  - Matching image features (points-of-interest, pixelwise) on features of 3D object model
    $\Rightarrow$ 2D-3D correspondences allow recovery of 6D pose
  - Requires textures on objects for meaningful features
  - More robust to occlusions due to feature-based matching

Outline

PoseCNN - Introduction

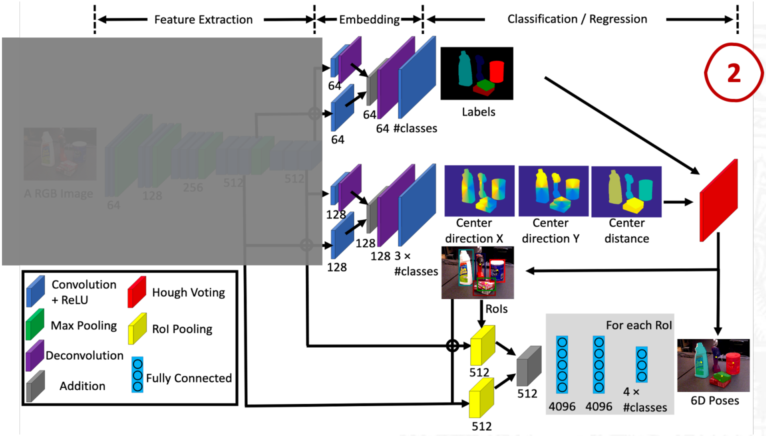
- PoseCNN is a DNN for 6D Object Pose Estimation
- Combines the advantages of both approaches
- Split into two stages:



First stage in PoseCNN. Extracting shared image features. Xiang et al. 2018.

PoseCNN - Introduction

- PoseCNN is a DNN for 6D Object Pose Estimation
- Combines the advantages of both approaches
- Split into two stages:



Second stage in PoseCNN, extracting task specific features. Xiang et al. 2018.

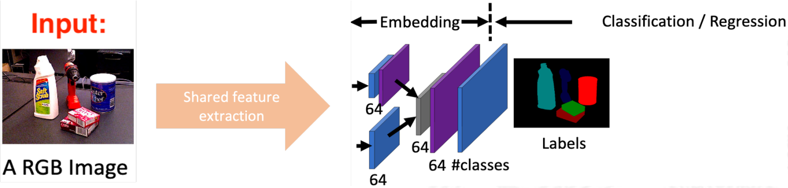
PoseCNN - Breakdown into three Tasks

PoseCNN breaks down the 6D pose estimation into 3 tasks:

1. Semantic labeling
2. 3D translation estimation
3. 3D rotation regression

PoseCNN - Semantic Labeling (1/3)

- First branch of the network, used for object detection
- Richer information about object shape than e.g. bounding box

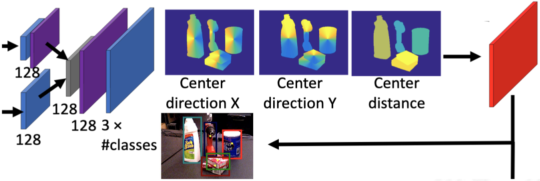


CNN architecture for semantic labeling in PoseCNN. Xiang et al. 2018.

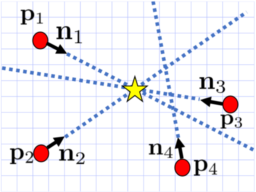
- Semantic labeling of individual objects
- Additionally helps for 3D translation estimation

PoseCNN - 3D Translation Estimation (2/3)

- Estimate the 3D translation $\mathbf{T} = (T_x, T_y, T_z)^T$ (object origin in camera coordinate system)
- Recover $\mathbf{T}$ from 2D object center $\mathbf{C}$ and T_z ($\rightarrow$ projection equation)



Architecture for the 3D translation estimation. Xiang et al. 2018.

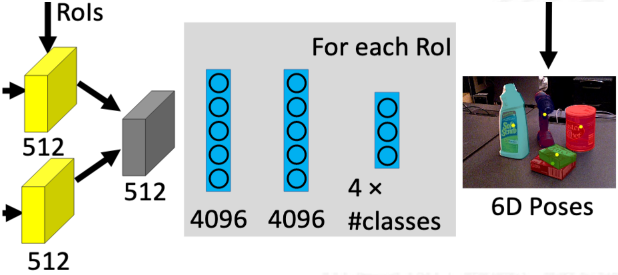


Pixelwise center voting in PoseCNN. Xiang et al. 2018.

- Hough voting layer outputs center points. Depth T_z is mean of pixelwise-depth prediction

PoseCNN - 3D Rotation Regression (3/3)

- We know which object, we know its 3D Translation → need the 3D rotation of the object
- Input: Image features, BBox contents, regress to quaternion representation



PoseCNN architecture branch for the 3D rotation regression. Xiang et al. 2018.

PoseCNN - Training

- Training on YCB-Video, subset of LINEMOD and 80k synthetic images of the YCB set.

21 YCB objects used for training. Xiang et al. 2018.

Outline

Problems with PoseCNN

PoseCNN achieve state-of-the-art results on YCB-Video, LINEMOD and Occluded-LINEMOD.

1. Manually labeled 3D object detection datasets are prohibitive
2. Test data highly corrolated to training data
3. Explicitly challenging to generalize
   - same camera intrinsics
   - same background biases
   - similar (restricted) lighting conditions

Problems with PoseCNN

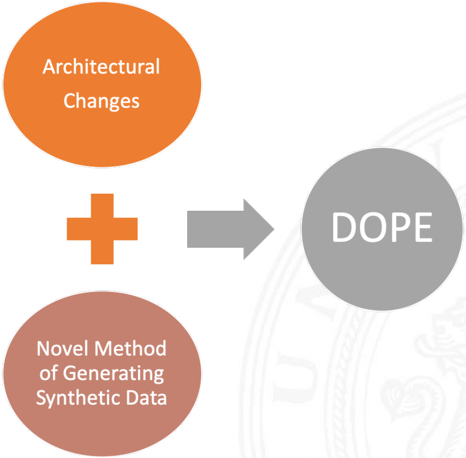
PoseCNN achieve state-of-the-art results on YCB-Video, LINEMOD and Occluded-LINEMOD.

1. Manually labeled 3D object detection datasets are prohibitive
2. Test data highly correlated to training data
3. Explicitly challenging to generalize
   - same camera intrinsics
   - same background biases
   - similar (restricted) lighting conditions

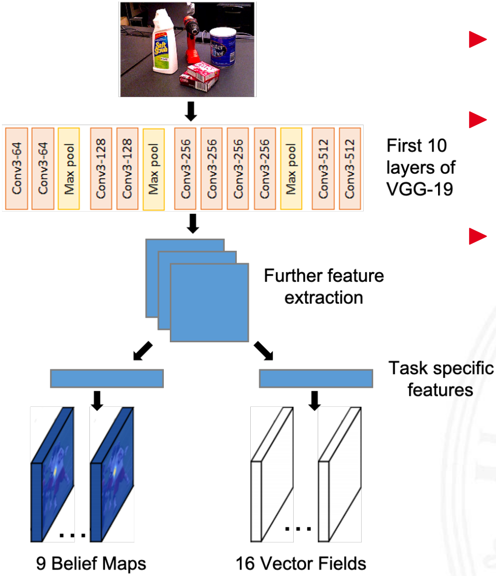
In practice, restricts use of PoseCNN.

Deep Object Pose Estimation (DOPE)

- Single-Shot Deep Neural Net for 6D Object Pose Estimation



DOPE - 1. Architectural Changes



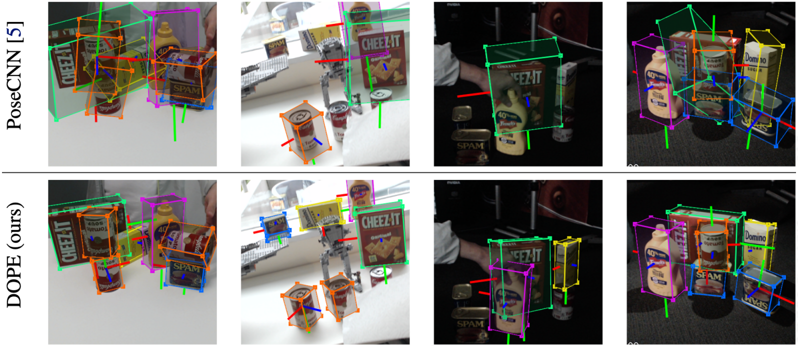
- (Belief Maps, Vector Fields) $\rightarrow$ Vertices Estimation
- Object Pose: Vertices correspond to 3D bounding box edges
- (Projected vertices, camera intrinsics, object dimensions) $\rightarrow$ **PnP-Algorithm** $\rightarrow$ **6D Pose**

DOPE - 2. Synthetic Datasets Only!

Examples for the domain randomized and photorealistic image datasets. Tremblay et al. 2018.

- Training on 60k domain-randomized, 60k photorealistic images
- Vary camera position, background, light, contrast, texture, distractors, orientation, etc.

PoseCNN vs. DOPE - Results



PoseCNN vs. DOPE estimation of YCB objects on data showing extreme lighting conditions. Tremblay et al. 2018.

- On-par with/better than PoseCNN on YCB-Video dataset
- Better generalization, e.g. extreme lighting conditions, new backgrounds

Outline

So in Conclusion:

- PoseCNN and DOPE achieve state-of-the-art in 6D Object Pose Estimation (2018)
- DOPE superior generalization to new environments
- DR + photorealistic data promising technique for data generation

DOPE on Trixi

Back-Up Slides

PoseCNN - Projection Equation

- 3D translation $\mathbf{T} = (T_x, T_y, T_z)^T$ can be recovered based on the following equation:

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} f_x \frac{T_x}{T_z} + p_x \\ f_y \frac{T_y}{T_z} + p_y \end{bmatrix}$$

- Where:

  $\mathbf{C} = (c_x, c_y)^T$ is the estimated 2D object center (projection of $\mathbf{T}$ on the image)

  T_z is the estimated depth of $\mathbf{C}$

  f_x, f_y are the focal lengths of the camera

  $(p_x, p_y)^T$ is the principal point

Evaluation Metric - ADD

Average Distance Metric (ADD) for evaluation of 6D pose estimation:

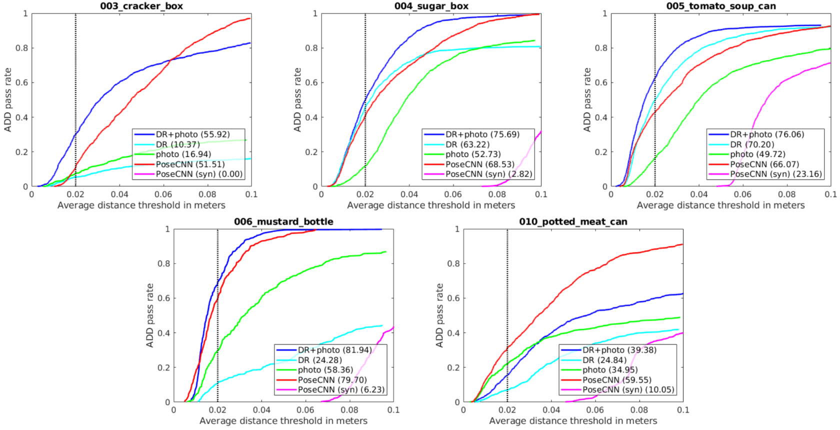
$$\mathrm{ADD} = \frac{1}{m} \sum_{\mathbf{x} \in \mathcal{M}} \|(\mathbf{R}\mathbf{x} + \mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{T}})\|$$

$$\text{ADD-S} = \frac{1}{m} \sum_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \|(\mathbf{R}\mathbf{x}_1 + \mathbf{T}) - (\tilde{\mathbf{R}}\mathbf{x}_2 + \tilde{\mathbf{T}})\|$$

Where:

- **R** and **T** are the ground-truth rotation and translation
- $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$ are the estimated rotation and translation
- $\mathcal{M}$ denotes the set of 3D model points, *m* is the number of points

PoseCNN vs. DOPE - Accuracy-threshold Curves



Accuracy-threshold curves for 5 YCB objects on the YCB-Video dataset. Tremblay et al. 2018.

- Numbers display the area under the curve (AUC)