



Object Pose Estimation with PointNet

Michael Haberl

Supervisor: Prof. Dr. Ulrich Bauer

Advisor: Benjamin Busam, M.Sc.

05.10.2018

Introduction

Determining the Architecture

Evaluation on Synthetic Data

Evaluation on the Linemod Dataset

Overview

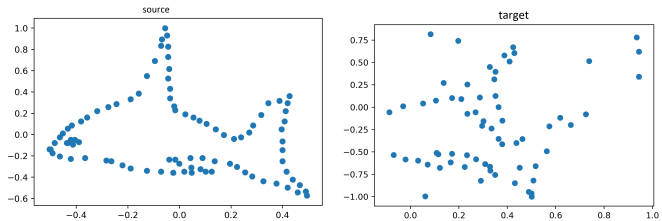
Introduction

Determining the Architecture

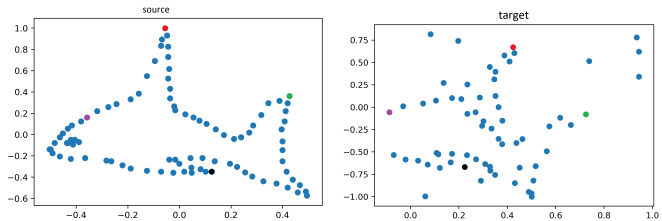
Evaluation on Synthetic Data

Evaluation on the Linemod Dataset

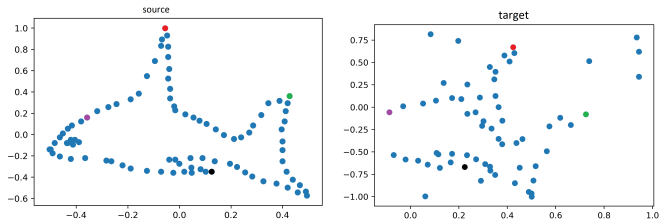
Pose estimation



Pose estimation



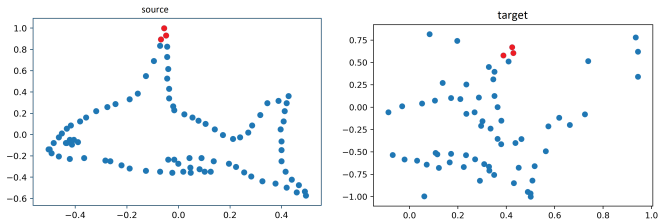
Pose estimation



Approaches

- Alternating optimisation

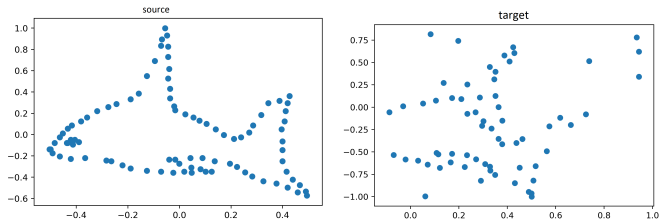
Pose estimation



Approaches

- Alternating optimisation
- Feature matching

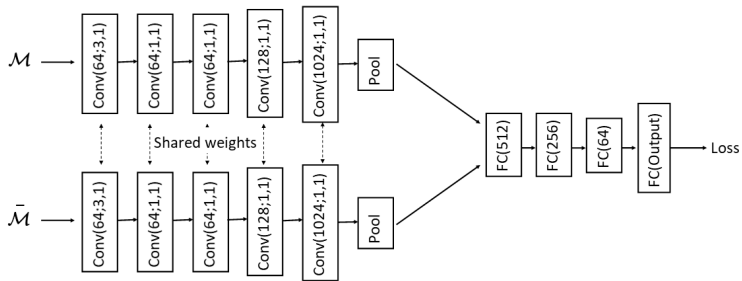
Pose estimation



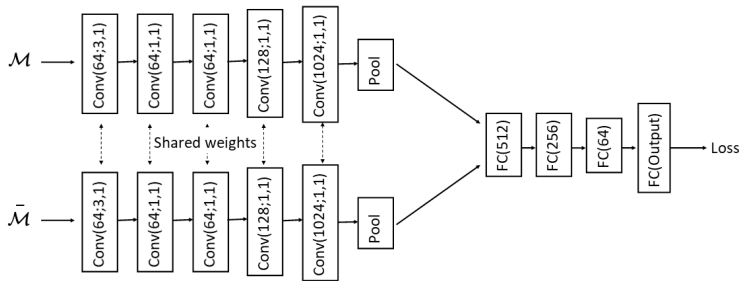
Approaches

- Alternating optimisation
- Feature matching
- Template matching

Used Architecture



Used Architecture



$$\mathcal{L}(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) = w_{rot} \cdot L_{rot} + w_{transl} \cdot L_{transl} + w_{res} \cdot L_{res} + w_{reg} \cdot L_{reg}$$

Pose parametrisation

Pose

$$\mathbf{P} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4}$$

$$\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3$$

Pose parametrisation

Pose

$$\mathbf{P} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4}$$

$$\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3$$

Rotation parametrisations

- $\mathbf{q} = (q_1, q_2, q_3, q_4) \in S^3$

Pose parametrisation

Pose

$$\mathbf{P} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4}$$

$$\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3$$

Rotation parametrisations

- $\mathbf{q} = (q_1, q_2, q_3, q_4) \in S^3$
- $\mathbf{R} \in SO(3)$

Pose parametrisation

Pose

$$\mathbf{P} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4}$$

$$\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3$$

Rotation parametrisations

- $\mathbf{q} = (q_1, q_2, q_3, q_4) \in S^3$
- $\mathbf{R} \in SO(3)$
- $\mathbf{r} = \theta \mathbf{u}$, $\mathbf{u} \in S^2$, $\theta \in (-\pi, \pi)$

Overview

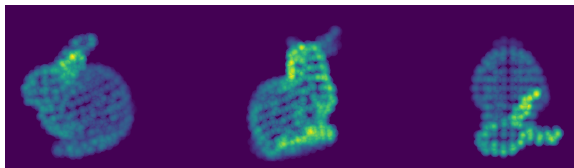
Introduction

Determining the Architecture

Evaluation on Synthetic Data

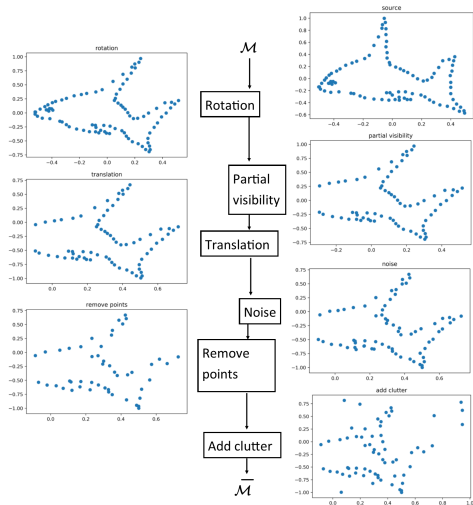
Evaluation on the Linemod Dataset

Dataset to determine the architecture



<https://www.cc.gatech.edu/~turk/bunny/bunny.html>, accessed
09.09.2018

Augmentation process for synthetic data



Parameters used to create the targets

Parameters used for creating the target point clouds

- uniformly sampled quaternions $\mathbf{q} \in \mathcal{S}_+^3$
- uniformly sampled translations with $\|\mathbf{t}\| < 0.3$
- Gaussian noise $\mathcal{N}(0, 0.01)$ clipped to $[-0.05, 0.05]$

Loss functions

$SE(3)$ loss

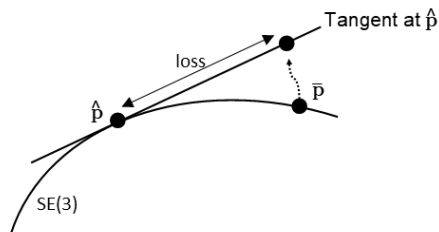
- $\varphi_{SE(3)}(\bar{\mathbf{p}}, \hat{\mathbf{p}}) = \left\| \text{Log}_{\hat{\mathbf{p}}}^Z(\bar{\mathbf{p}}) \right\|_{Z_{\hat{\mathbf{p}}}}^2$
 $\mathbf{p} = (r_1, r_2, r_3, t_1, t_2, t_3)$

B. Hou, Computing CNN Loss and Gradients for Pose Estimation with Riemannian Geometry [HMK⁺18]

Loss functions

$SE(3)$ loss

- $\varphi_{SE(3)}(\bar{\mathbf{p}}, \hat{\mathbf{p}}) = \left\| \text{Log}_{\hat{\mathbf{p}}}^Z(\bar{\mathbf{p}}) \right\|_{Z_{\hat{\mathbf{p}}}}^2$
 $\mathbf{p} = (r_1, r_2, r_3, t_1, t_2, t_3)$



B. Hou, Computing CNN Loss and Gradients for Pose Estimation with Riemannian Geometry [HMK⁺18]

Loss functions

Rotation losses

- $\varphi_1(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \arccos \left(2 \langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle^2 - 1 \right)$
- $\varphi_2(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \min \{ \|\hat{\mathbf{q}} - \bar{\mathbf{q}}\|, \|\hat{\mathbf{q}} + \bar{\mathbf{q}}\| \}$
- $\varphi_3(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \arccos (|\langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle|)$
- $\varphi_4(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = 1 - \langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle^2$

φ_1, φ_4 : Anonymous, Explaining the Ambiguity of Object Detection and Pose Estimation from Visual Data [Ano]
others: D. Huynh, Metrics for 3D Rotations: Comparison and Analysis [Huy09]

Loss functions

Rotation losses

- $\varphi_1(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \arccos \left(2 \langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle^2 - 1 \right)$
- $\varphi_2(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \min \{ \|\hat{\mathbf{q}} - \bar{\mathbf{q}}\|, \|\hat{\mathbf{q}} + \bar{\mathbf{q}}\| \}$
- $\varphi_3(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \arccos (|\langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle|)$
- $\varphi_4(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = 1 - \langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle^2$
- $\varphi_5(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \left\| \mathbf{1}_3 - \hat{\mathbf{R}}\bar{\mathbf{R}}^T \right\|_F$
- $\varphi_6(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \arccos \left(\left(\text{Tr} \left(\hat{\mathbf{R}}\bar{\mathbf{R}}^T \right) - 1 \right) / 2 \right)$

φ_1, φ_4 : Anonymous, Explaining the Ambiguity of Object Detection and Pose Estimation from Visual Data [Ano]
others: D. Huynh, Metrics for 3D Rotations: Comparison and Analysis [Huy09]

Translation and regularisation loss

Translation loss

- $L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|_2$

Translation and regularisation loss

Translation loss

- $L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|_2$

Regularisation loss

- $L_{reg}(\mathbf{q}) = L_\delta(\|\mathbf{q}\| - 1)$

Translation and regularisation loss

Translation loss

- $L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|_2$

Regularisation loss

- $L_{reg}(\mathbf{q}) = L_\delta(\|\mathbf{q}\| - 1)$
- $L_{reg}(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \|\mathbf{1}_3 - \hat{\mathbf{R}}\bar{\mathbf{R}}^T\|_F$

Evaluation metrics

Evaluation metrics

- ADD: $e_{ADD}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x} \in \mathcal{M}} \|\bar{\mathbf{P}}_{\mathbf{x}} - \hat{\mathbf{P}}_{\mathbf{x}}\|$

T. Hodan, On evaluation of 6D object pose estimation [HMO16].

Evaluation metrics

Evaluation metrics

- ADD: $e_{ADD}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x} \in \mathcal{M}} \left\| \bar{\mathbf{P}}_{\mathbf{x}} - \hat{\mathbf{P}}_{\mathbf{x}} \right\|$
- ADI: $e_{ADI}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \left\| \bar{\mathbf{P}}_{\mathbf{x}_1} - \hat{\mathbf{P}}_{\mathbf{x}_2} \right\|$

T. Hodan, On evaluation of 6D object pose estimation [HMO16].

Evaluation metrics

Evaluation metrics

- ADD: $e_{ADD}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x} \in \mathcal{M}} \|\bar{\mathbf{P}}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}\|$
- ADI: $e_{ADI}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \|\bar{\mathbf{P}}\mathbf{x}_1 - \hat{\mathbf{P}}\mathbf{x}_2\|$
- Pose correct: $e_{ADD} \leq 0.1d_{\mathcal{M}}$

T. Hodan, On evaluation of 6D object pose estimation [HMO16].

Evaluation metrics

- ADD: $e_{ADD}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x} \in \mathcal{M}} \|\bar{\mathbf{P}}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}\|$
- ADI: $e_{ADI}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \|\bar{\mathbf{P}}\mathbf{x}_1 - \hat{\mathbf{P}}\mathbf{x}_2\|$
- Pose correct: $e_{ADD} \leq 0.1d_{\mathcal{M}}$
- Rotation error:
$$e_{RE}(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \arccos\left(\left(\text{Tr}(\hat{\mathbf{R}}\bar{\mathbf{R}}^{-1}) - 1\right)/2\right) \cdot 180/\pi$$

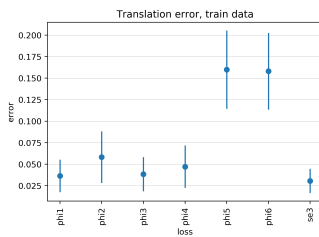
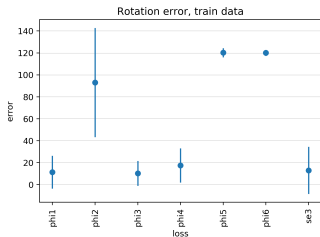
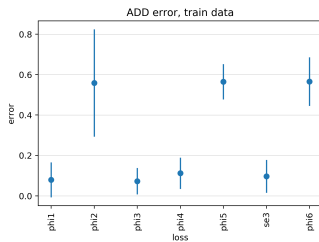
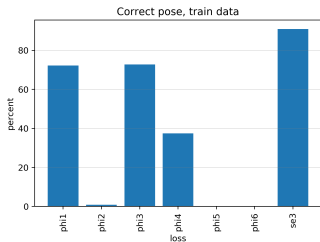
T. Hodan, On evaluation of 6D object pose estimation [HMO16].

Evaluation metrics

- ADD: $e_{ADD}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x} \in \mathcal{M}} \|\bar{\mathbf{P}}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}\|$
- ADI: $e_{ADI}(\hat{\mathbf{P}}, \bar{\mathbf{P}}; \mathcal{M}) = \text{avg}_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \|\bar{\mathbf{P}}\mathbf{x}_1 - \hat{\mathbf{P}}\mathbf{x}_2\|$
- Pose correct: $e_{ADD} \leq 0.1d_{\mathcal{M}}$
- Rotation error:
 $e_{RE}(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \arccos\left(\left(\text{Tr}(\hat{\mathbf{R}}\bar{\mathbf{R}}^{-1}) - 1\right)/2\right) \cdot 180/\pi$
- Translation error: $e_{TE}(\hat{\mathbf{t}}, \bar{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|$

T. Hodan, On evaluation of 6D object pose estimation [HMO16].

Comparing rotation losses



Rotation losses

- $\varphi_1(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \arccos \left(2 \langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle^2 - 1 \right)$
- $\varphi_2(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \min \{ \|\hat{\mathbf{q}} - \bar{\mathbf{q}}\|, \|\hat{\mathbf{q}} + \bar{\mathbf{q}}\| \}$
- $\varphi_3(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = \arccos (|\langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle|)$
- $\varphi_4(\hat{\mathbf{q}}, \bar{\mathbf{q}}) = 1 - \langle \hat{\mathbf{q}}, \bar{\mathbf{q}} \rangle^2$
- $\varphi_5(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \left\| \mathbf{1}_3 - \hat{\mathbf{R}}\bar{\mathbf{R}}^T \right\|_F$
- $\varphi_6(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \arccos \left(\left(\text{Tr} \left(\hat{\mathbf{R}}\bar{\mathbf{R}}^T \right) - 1 \right) / 2 \right)$

Overview

Introduction

Determining the Architecture

Evaluation on Synthetic Data

Evaluation on the Linemod Dataset

Linemod objects



S. Hinterstoisser, Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes [HLI⁺13]

Parameters used to create the targets

Parameters used for creating the target point clouds

- uniformly sampled quaternions $\mathbf{q} \in \mathcal{S}_+^3$
- uniformly sampled translations with $\|\mathbf{t}\| < 0.3$
- Gaussian noise $\mathcal{N}(0, 0.01)$ clipped to $[-0.05, 0.05]$

Training on 3D data

Parameters

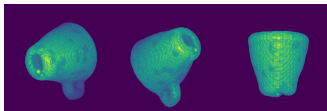
- loss:

$$\mathcal{L}(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) = w_{rot} \cdot \varphi_1(\bar{\mathbf{q}}, \hat{\mathbf{q}}) + w_{transl} \cdot L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) + w_{reg} \cdot L_{reg}(\hat{\mathbf{q}})$$

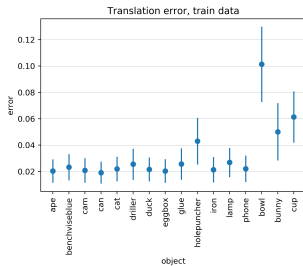
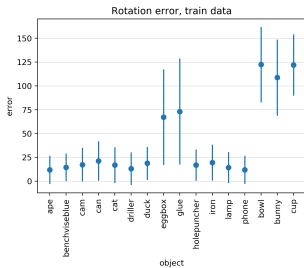
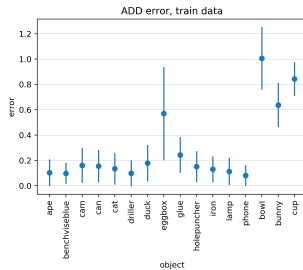
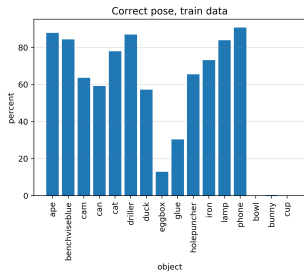
- epochs: 80000
- learning rate: 0.005
- training data: 13 of the Linemod objects

Challenges

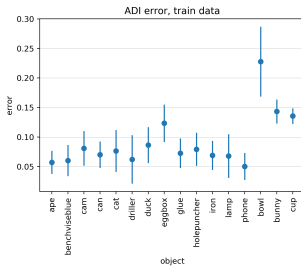
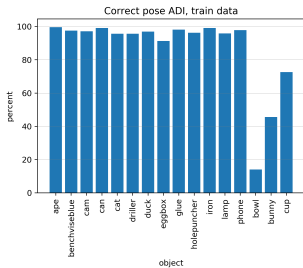
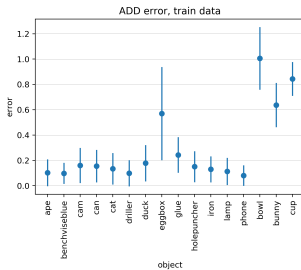
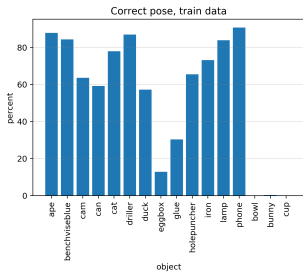
- Ambiguous objects



Evaluation

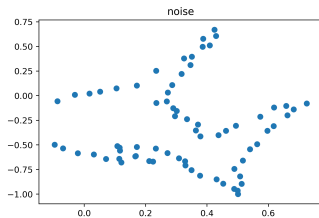
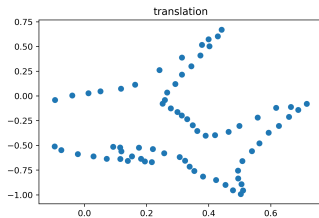


Evaluation

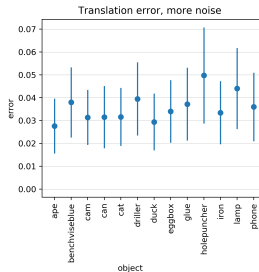
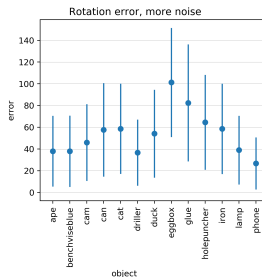
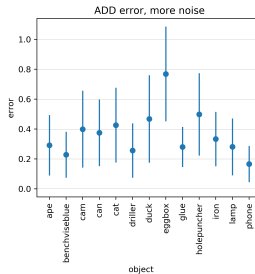
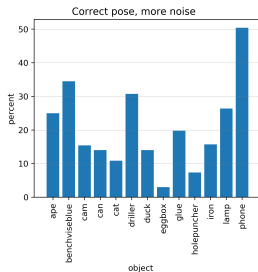


Challenges

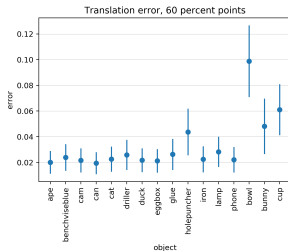
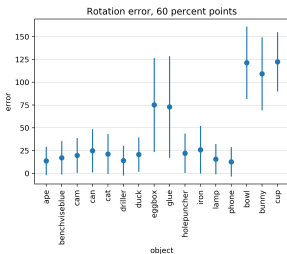
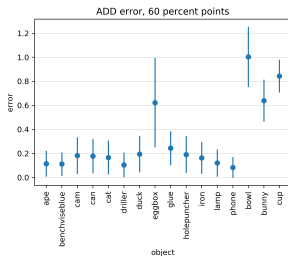
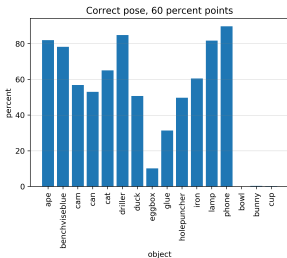
- Ambiguous objects
- Noise



Evaluation with noise

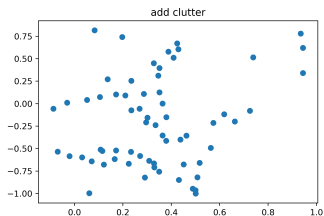
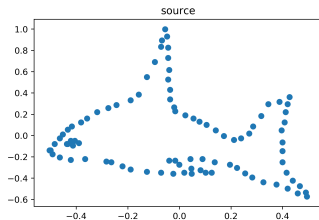


Evaluation with 40% missing points

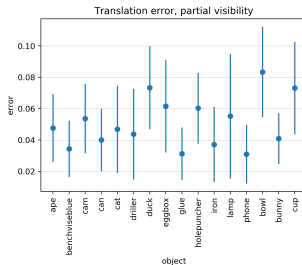
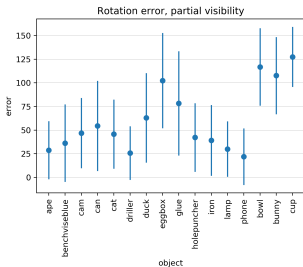
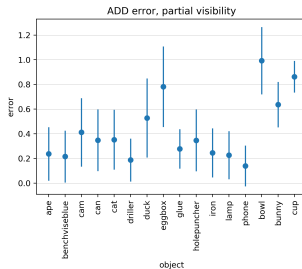
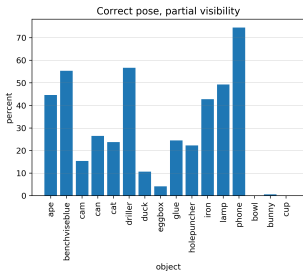


Challenges

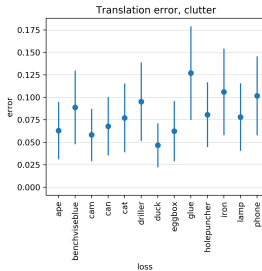
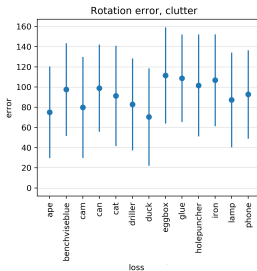
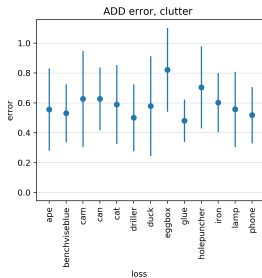
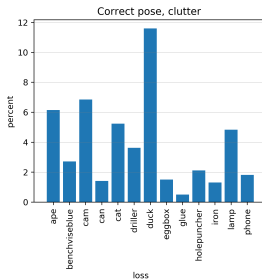
- Ambiguous objects
- Noise
- Occlusions and clutter



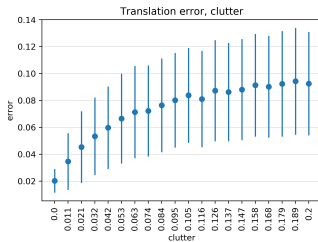
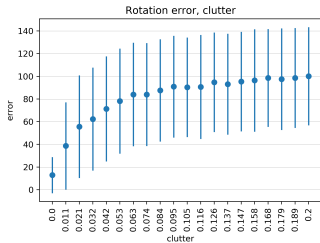
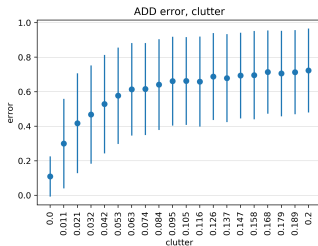
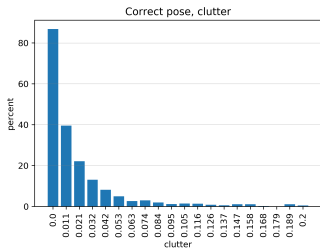
Evaluation with partial visibility



Evaluation with clutter



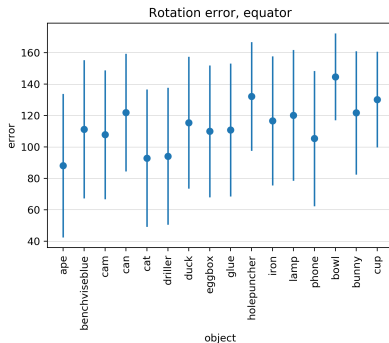
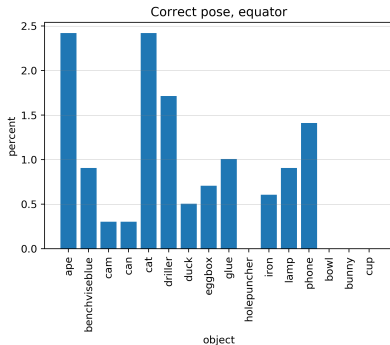
Evaluation with increasing clutter



Challenges

- Ambiguous objects
- Noise
- Occlusions and clutter
- Representation of the rotation

Evaluation around the equator

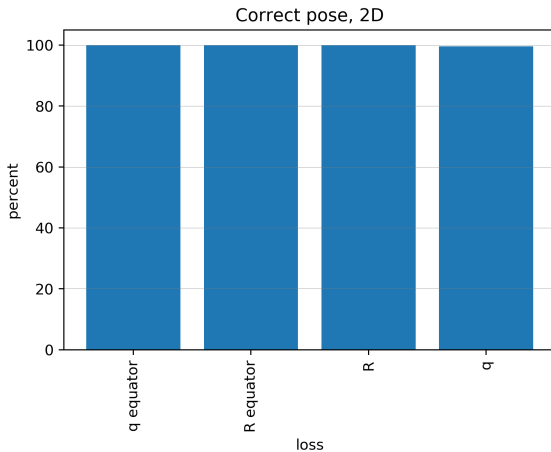


Training on 2D data

Loss functions

- $\mathcal{L}_R(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) = w_{rot} \cdot \varphi_5(\bar{\mathbf{R}}, \hat{\mathbf{R}}) + w_{transl} \cdot L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}})$
- $\mathcal{L}_q(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) =$
 $w_{rot} \cdot \varphi_1(\bar{\mathbf{q}}, \hat{\mathbf{q}}) + w_{transl} \cdot L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) + w_{reg} \cdot L_{reg}(\hat{\mathbf{q}})$
- learning rate: 0.0005

Evaluation 2D



Overview

Introduction

Determining the Architecture

Evaluation on Synthetic Data

Evaluation on the Linemod Dataset

Linemod-Dataset

Provided are:

- meshes of 15 Household objects
- for each object 1100 RGB and depth images
- ground truth poses and the bounding boxes

Linemod-Dataset



S. Hinterstoisser, Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes [HLI⁺13]

Residual losses

Residual loss

- $L_{res}(\bar{\mathcal{M}}, \hat{\mathcal{M}}) = \frac{1}{N} \sum_{\mathbf{x}_i \in \bar{\mathcal{M}}} \min_{\mathbf{x}_j \in \hat{\mathcal{M}}} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \frac{1}{N} \sum_{\mathbf{x}_i \in \bar{\mathcal{M}}} r_i^2$

P. Huber, Robust Estimation of a Location Parameter[Hub64]

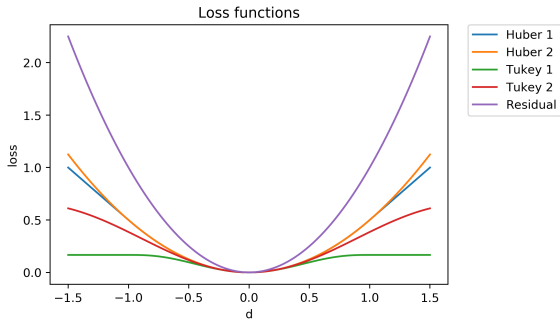
Residual losses

Residual loss

- $L_{res}(\bar{\mathcal{M}}, \hat{\mathcal{M}}) = \frac{1}{N} \sum_{\mathbf{x}_i \in \bar{\mathcal{M}}} \min_{\mathbf{x}_j \in \hat{\mathcal{M}}} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \frac{1}{N} \sum_{\mathbf{x}_i \in \bar{\mathcal{M}}} r_i^2$
- $L_\delta(r) = \begin{cases} \frac{1}{2}r^2 & , \text{for } |r| \leq \delta \\ \delta|r| - \frac{1}{2}\delta^2 & , \text{otherwise.} \end{cases}$

P. Huber, Robust Estimation of a Location Parameter[Hub64]

Comparing residual losses

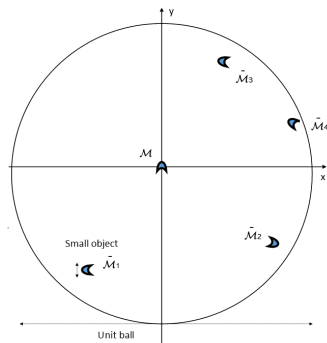


Residual loss

- $L_{res}(\bar{\mathcal{M}}, \hat{\mathcal{M}}) = \frac{1}{N} \sum_{\mathbf{x}_i \in \bar{\mathcal{M}}} \min_{\mathbf{x}_j \in \hat{\mathcal{M}}} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \frac{1}{N} \sum_{\mathbf{x}_i \in \bar{\mathcal{M}}} r_i^2$
- $L_\delta(r) = \begin{cases} \frac{1}{2}r^2 & , \text{for } |r| \leq \delta \\ \delta|r| - \frac{1}{2}\delta^2 & , \text{otherwise.} \end{cases}$
- $L_c(r) = \begin{cases} \frac{c^2}{6} \left[\left(1 - \left(\frac{r}{c}\right)^2\right)^3 \right] & , \text{if } |r| \leq c \\ \frac{c^2}{6} & , \text{otherwise.} \end{cases}$

Challenges

- Ambiguous objects
- Noise
- Occlusions and clutter
- Representation of the rotation
- Small object diameter compared to translation



Training on the Linemod dataset

Parameters

- 2 networks: translation-network, rotation-network

Training on the Linemod dataset

Parameters

- 2 networks: translation-network, rotation-network
- $L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|_2$

Training on the Linemod dataset

Parameters

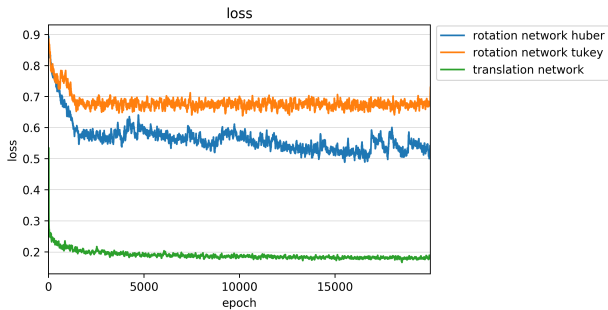
- 2 networks: translation-network, rotation-network
- $L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|_2$
- $\mathcal{L}_1(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) = w_{res} \cdot L_{hub}(\bar{\mathcal{M}}, \hat{\mathcal{M}}) + w_{rot} \cdot \varphi_1(\bar{\mathbf{q}}, \hat{\mathbf{q}}) + w_{transl} \cdot L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) + w_{reg} \cdot L_{reg}(\hat{\mathbf{q}})$
- $\mathcal{L}_2(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) = w_{res} \cdot L_{tukey}(\bar{\mathcal{M}}, \hat{\mathcal{M}}) + w_{rot} \cdot \varphi_1(\bar{\mathbf{q}}, \hat{\mathbf{q}}) + w_{transl} \cdot L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) + w_{reg} \cdot L_{reg}(\hat{\mathbf{q}})$

Training on the Linemod dataset

Parameters

- 2 networks: translation-network, rotation-network
- $L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|_2$
- $\mathcal{L}_1(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) = w_{res} \cdot L_{hub}(\bar{\mathcal{M}}, \hat{\mathcal{M}}) + w_{rot} \cdot \varphi_1(\bar{\mathbf{q}}, \hat{\mathbf{q}}) + w_{transl} \cdot L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) + w_{reg} \cdot L_{reg}(\hat{\mathbf{q}})$
- $\mathcal{L}_2(\bar{\mathbf{P}}, \hat{\mathbf{P}}, \mathcal{M}) = w_{res} \cdot L_{tukey}(\bar{\mathcal{M}}, \hat{\mathcal{M}}) + w_{rot} \cdot \varphi_1(\bar{\mathbf{q}}, \hat{\mathbf{q}}) + w_{transl} \cdot L_{transl}(\bar{\mathbf{t}}, \hat{\mathbf{t}}) + w_{reg} \cdot L_{reg}(\hat{\mathbf{q}})$
- epochs: 20000
- learning rate: 0.005
- training data: 2 of the Linemod objects (lamp, phone)

Training



Outlook

- Add colour information
- Add surface normals
- Different representation for rotation

Thank you!



Anonymous.

Explaining the Ambiguity of Object Detection and Pose Estimation from Visual Data.



Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab.

Robust optimization for deep regression.

Proceedings of the IEEE International Conference on Computer Vision, pages 2830–2838, 2015.



Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab.

Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes.

Lecture Notes in Computer Science, 7724 LNCS(Part 1):548–562, 2013.



Benjamin Hou, Nina Miolane, Bishesh Khanal, Matthew C. H. Lee, Amir Alansary, Steven G. McDonagh, Joseph V. Hajnal, Daniel Rueckert, Ben Glocker, and Bernhard Kainz.

Computing CNN loss and gradients for pose estimation with riemannian geometry.

CoRR, abs/1805.01026, 2018.



Tomáš Hodňan, Jiří Matas, and Štěpán Obdržálek.

On evaluation of 6D object pose estimation.

Lecture Notes in Computer Science, 9915 LNCS:609–619, 2016.



Peter J. Huber.

Robust estimation of a location parameter.

Annals of Mathematical Statistics, 35(1):73–101, March 1964.



Du Q. Huynh.

Metrics for 3D rotations: Comparison and analysis.

Journal of Mathematical Imaging and Vision, 35(2):155–164,
2009.