Supplementary Material of
ZebraPose: Coarse to Fine Surface Encoding for 6DoF Object Pose Estimation

Yongzhi Su[1,2*] Mahdi Saleh[3*] Torben Fetzer[2] Jason Rambach[1]
Nassir Navab[3] Benjamin Busam[3] Didier Stricker[1,2] Federico Tombari[3,4]

[1] German Research Center for Artificial Intelligence (DFKI) [2] TU Kaiserslautern
[3]Technische Universität München [4]Google

{yongzhi.su; jason.rambach; torben.fetzer; didier.stricker}@dfki.de
{m.saleh; b.busam; nassir.navab}@tum.de, tombari@in.tum.de

1. Hyper-parameters in the Pose Solver

For RANSAC/PnP [7], we set the threshold value for reprojection error as 2 pixels, and execute 150 iterations. For Progressive-X [1], we also set the threshold value for the reprojection error as 2 pixels, and execute 400 iterations. The additional parameters for Progressive-X are "neighborhood_ball_radius=20", "spatial_coherence_weight=0.1", "maximum_tanimoto_similarity=0.9".

2. BOP Challenge

We submitted the results on 4 datasets of the BOP challenge and will test our method on the rest 3 datasets. The results are online in BOP Leaderboards with the submission name "zebrapose".

3. YCB-V Evaluation per Object

We present a more detailed result on the YCB-V dataset [10] in Tab. 1 and Tab. 2. As the Tab. 1 shows, in the evaluation of the estimate pose w.r.t ADD(-S) metric, we show major improvement over the state of the art.

In Tab. 2, we carefully calculated the AUC with all-points interpolation algorithm with the maximum threshold of 10 cm. If we calculate the AUC with 11-points interpolation, we will reach AUC of ADD-S of 94%, and AUC of ADD(-S) of 89.8%.

4. Qualitative Results

4.1. Vertex Code Prediction LM-O

We visualized the predicted binary code of the "duck" object in LM-O dataset [2] with a few examples in Fig. 1. Due to the size limits, we only show the predicted binary code till the 11-th bits. We render the object with the predicted pose on top of the original input ROI. To make the predicted pose more visible in the figure, we set the colour of the object model as red just for this figure. So the duck appears with the orange colour (red + yellow) in the last row. We can see that the rendered object overlapped the object in the original image quite well, indicating that our predicted pose is very accurate.

4.2. Pose Prediction LM-O

Qualitative Results on LM-O [2] can be found in Fig. 2. We render the objects with estimated pose on top of the original images. The presented confidence scores are from the 2D object detection with FCOS detector [8].

4.3. Pose Prediction YCB-V

Qualitative Results on YCB-V [10] are available in Fig. 3. We render the objects with estimated pose on top of the original images. The presented confidence scores are from the 2D object detection with FCOS detector [8].

[*]The authors contributed equally to this paper
Code: https://github.com/suyz526/ZebraPose

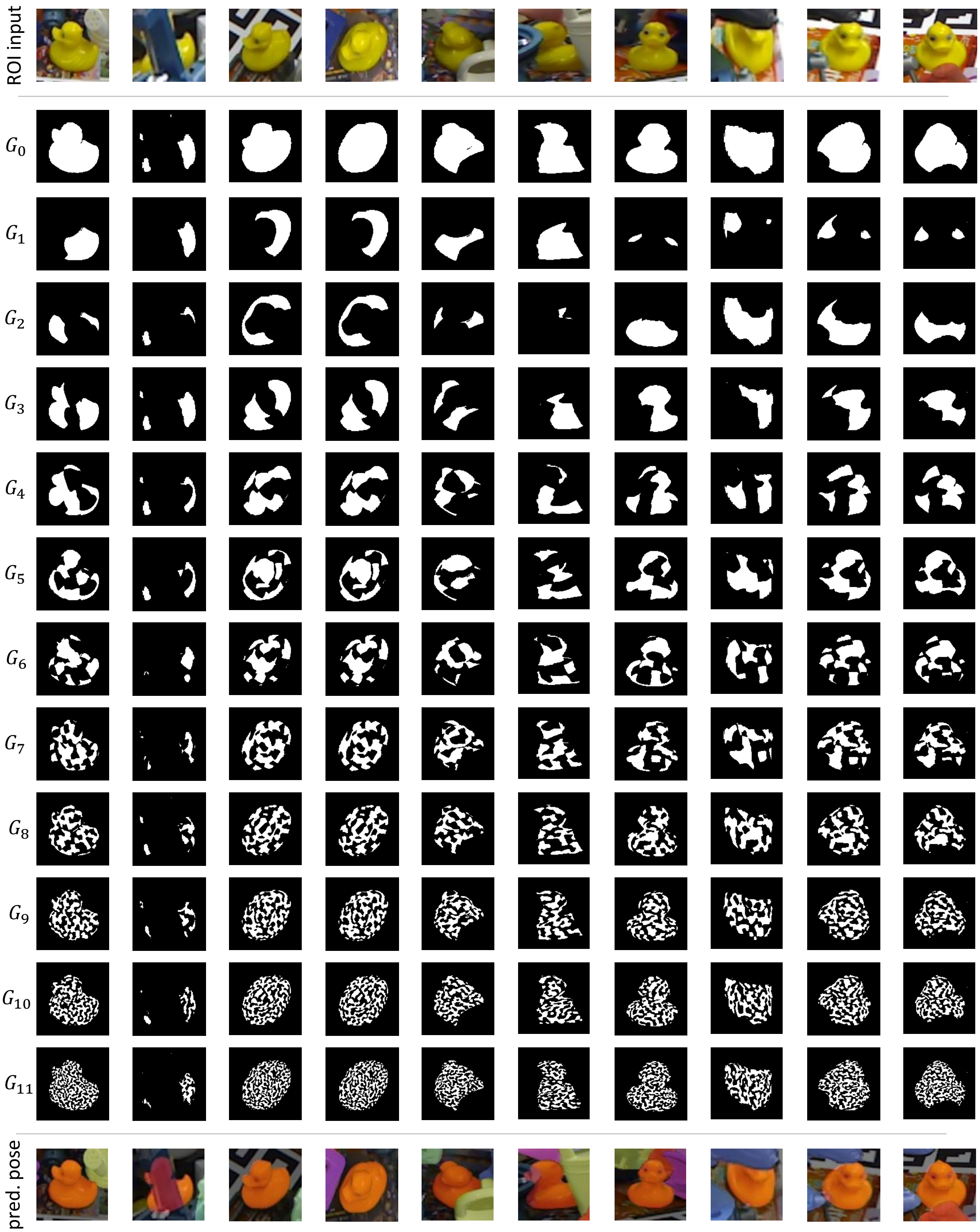

Figure 1. We visualized the predicted binary code of the "duck" in LM-O dataset [2] with a few examples. Due to the size limits, we only show the predicted binary code till the 11-th bit. We set the colour of the object model as red and render the object with the predicted pose on the top of the input ROI. We can see that the rendered object overlaps the object in the image quite well.

Figure 2. **Qualitative Results on LM-O [2]**: We render the objects with estimated pose on top of the original images. The presented confidence score are from the 2D object detection with FCOS detector [8].

Method	SegDriven [4]	Single-Stage [3]	RePose [5]	GDR-Net [9]	**Ours**
002_master_chef_can	33.0	-	-	41.5	**62.6**
003_cracker_box	44.6	-	-	83.2	**98.5**
004_sugar_box	75.6	-	-	91.5	**96.3**
005_tomato_soup_can	40.8	-	-	65.9	**80.5**
006_mustard_bottle	70.6	-	-	90.2	**100.0**
007_tuna_fish_can	18.1	-	-	44.2	**70.5**
008_pudding_box	12.2	-	-	2.8	**99.5**
009_gelatin_box	59.4	-	-	61.7	**97.2**
010_potted_meat_can	33.3	-	-	64.9	**76.9**
011_banana	16.6	-	-	64.1	**71.2**
019_pitcher_base	90.0	-	-	99.0	**100.0**
021_bleach_cleanser	70.9	-	-	73.8	**75.9**
024_bowl*	30.5	-	-	**37.7**	18.5
025_mug	40.7	-	-	61.5	**77.5**
035_power_drill	63.5	-	-	78.5	**97.4**
036_wood_block*	27.7	-	-	59.5	**87.6**
037_scissors	17.1	-	-	3.9	**71.8**
040_large_marker	4.8	-	-	7.4	**23.3**
051_large_clamp*	25.6	-	-	69.8	**87.6**
052_extra_large_clamp*	8.8	-	-	90.0	**98.0**
061_foam_brick*	34.7	-	-	71.9	**99.3**
mean	39.0	53.9	62.1	60.1	**80.5**

Table 1. **Comparison with State of the Art on YCB-V**. We report the Average Recall of ADD(-S) in % and compare with state of the art. (*) denotes symmetric objects, (-) denotes the results missing from the original paper.

Method	PoseCNN [10]		CosyPose [6]		GDR-Net [9]		**Ours**	
Metric	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
002_master_chef_can	84.0	50.9	-	-	96.3	65.2	93.7	75.4
003_cracker_box	76.9	51.7	-	-	97.0	88.8	93.0	87.8
004_sugar_box	84.3	68.6	-	-	98.9	95.0	95.1	90.9
005_tomato_soup_can	80.9	66.0	-	-	96.5	91.9	94.4	90.1
006_mustard_bottle	90.2	79.9	-	-	100	92.8	96.0	92.6
007_tuna_fish_can	87.9	70.4	-	-	99.4	94.2	96.9	92.6
008_pudding_box	79.0	62.9	-	-	64.6	44.7	97.2	95.3
009_gelatin_box	87.1	75.2	-	-	97.1	92.5	96.8	94.8
010_potted_meat_can	78.5	59.6	-	-	86.0	80.2	91.7	83.6
011_banana	85.9	72.3	-	-	96.3	85.8	92.6	84.6
019_pitcher_base	76.8	52.5	-	-	99.9	98.5	96.4	93.4
021_bleach_cleanser	71.9	50.5	-	-	94.2	84.3	89.5	80.0
024_bowl*	69.7	69.7	-	-	85.7	85.7	37.1	37.1
025_mug	78.0	57.7	-	-	99.6	94.0	96.1	90.8
035_power_drill	72.8	55.1	-	-	97.5	90.1	95.0	89.7
036_wood_block*	65.8	65.8	-	-	82.5	82.5	84.5	84.5
037_scissors	56.2	35.8	-	-	63.8	49.5	92.5	84.5
040_large_marker	71.4	58.0	-	-	88.0	76.1	80.4	69.5
051_large_clamp*	49.9	49.9	-	-	89.3	89.3	85.6	85.6
052_extra_large_clamp*	47.0	47.0	-	-	93.5	93.5	92.5	92.5
061_foam_brick*	87.8	87.8	-	-	96.9	96.9	95.3	95.3
mean	75.9	61.3	89.8	84.5	91.6	84.3	90.1	85.3

Table 2. **Comparison with State of the Art on YCB-V**. We report the Average Recall w.r.t AUC of ADD(-S) and AUC of ADD-S in % and compare with state of the art. (*) denotes symmetric objects, (-) denotes the results missing from the original paper.

Figure 3. **Qualitative Results on YCB-V [10]**: We render the objects with estimated pose on top of the original images. The presented confidence score are from the 2D object detection with FCOS detector [8].

References

[1] Daniel Barath and Jiri Matas. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3780–3788, 2019. 1

[2] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and others. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3364–3372, 2016. 1, 2, 3

[3] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020. 4

[4] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3385–3394, 2019. 4

[5] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3303–3312, 2021. 4

[6] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020. 5

[7] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 1

[8] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 1, 3, 6

[9] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021. 4, 5

[10] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 5, 6