Supplementary Material of
GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation

Gu Wang[1,2], Fabian Manhardt[2], Federico Tombari[2,3], Xiangyang Ji[1]
[1] Tsinghua University, BNRist [2] Technical University of Munich [3] Google
wangg16@mails.tsinghua.edu.cn, fabian.manhardt@tum.de,
tombari@in.tum.de, xyji@tsinghua.edu.cn

Abstract

In this supplementary material, we provide i) details of PnP/RANSAC, ii) detailed evaluation results on YCB-V [17], iii) results on LM-O [1] and YCB-V [17] under BOP [5] setup, and iv) qualitative results for LM [2], LM-O [1] and YCB-V [17].

A. Details of PnP/RANSAC

The implementation and hyper-parameters of PnP/RANSAC follow the state-of-the-art method CDPN [11] for all our experiments. Specifically, we leverage EPnP [9] together with 100 RANSAC iterations using a reprojection error threshold of 3 and confidence threshold of 0.99.

B. Detailed Results of YCB-V

We present detailed evaluation results on YCB-V [17] for our GDR-Net in Tab. B.1 and Tab. B.2 and compare them to state-of-the-art approaches w.r.t. ADD(-S) and AUC of ADD-S/ADD(-S), respectively. As for methods trained simultaneously for all objects, our GDR-Net clearly outperforms all other state-of-the-art methods. Furthermore, when GDR-Net is trained separately for each individual object, we can even surpass refinement-based methods such as DeepIM [10] w.r.t. AUC of ADD-S/ADD(-S) metric.

C. BOP Results on LM-O and YCB-V

In the main paper, we have presented the results on LM-O and YCB-V following the most commonly used evaluation protocol following another learned PnP [6] and many other works such as [17, 13, 14, 18, 10, 8]. Nevertheless, the evaluation protocol of BOP Challenge [4, 5] has recently become more popular. Therefore, we also present the results of our GDR-Net on LM-O and YCB-V under the BOP setup.

The BOP evaluation protocol differs from the former in three main aspects as follows. i) No real data should be used for LM-O, thus we only employ the provided synthetic `pbr` data [5] for training on LM-O; ii) The number of test images for both LM-O and YCB-V is smaller, *i.e.*, they only contains a subset of the original test images; iii) The evaluation metric is different. Thereby, for each dataset, an Average Recall (AR) score is reported by calculating the mean Average Recall of three different metrics: $\text{AR} = (\text{AR}_{\text{MSPD}} + \text{AR}_{\text{MSSD}} + \text{AR}_{\text{VSD}})/3$. Please refer to [5] for the detailed explanation of these metrics.

Tab. C.3 presents the results of our GDR-Net on LM-O and YCB-V compared with other state-of-the-art RGB-based methods under BOP setup. Since our method is built on top of CDPN [11], we follow [11] to train one network per object for the sake of fairness. We utilize the publicly available detections from FCOS [16] [1] following CDPNv2 [11]. We can see that our GDR-Net significantly outperforms all other state-of-the-art methods without refinement. It is worth noting that most of these top-performing methods [12, 3, 11] rely on the indirect PnP/RANSAC solver, while ours directly regresses the 6D object pose leveraging geometric guidance, which again demonstrates the effectiveness of our proposed learning-based Patch-PnP. Our GDR-Net even outperforms the state-of-the-art refinement-based method CosyPose [8] on LM-O. On YCB-V, ours is worse than CosyPose but far better than all other methods without refinement. Nevertheless, our method runs much faster than CosyPose as no refinement step is needed. Moreover, our method can be combined with an additional refiner such as CosyPose to achieve better results.

[1] https://github.com/LZGMatrix/BOP19_CDPN_2019ICCV

Method	PoseCNN [17]	SegDriven [7]	Single-Stage [6]	GDR-Net (**Ours**)	
P.E.	1	1	N	1	N
002_master_chef_can	3.6	33.0	-	51.7	41.5
003_cracker_box	25.1	44.6	-	45.1	83.2
004_sugar_box	40.3	75.6	-	83.9	91.5
005_tomato_soup_can	25.5	40.8	-	48.3	65.9
006_mustard_bottle	61.9	70.6	-	92.2	90.2
007_tuna_fish_can	11.4	18.1	-	29.1	44.2
008_pudding_box	14.5	12.2	-	39.7	2.8
009_gelatin_box	12.1	59.4	-	34.6	61.7
010_potted_meat_can	18.9	33.3	-	36.3	64.9
011_banana	30.3	16.6	-	60.2	64.1
019_pitcher_base	15.6	90.0	-	96.3	99.0
021_bleach_cleanser	21.2	70.9	-	73.0	73.8
024_bowl*	12.1	30.5	-	35.0	37.7
025_mug	5.2	40.7	-	39.3	61.5
035_power_drill	29.9	63.5	-	57.7	78.5
036_wood_block*	10.7	27.7	-	50.8	59.5
037_scissors	2.2	17.1	-	6.6	3.9
040_large_marker	3.4	4.8	-	13.7	7.4
051_large_clamp*	28.5	25.6	-	40.3	69.8
052_extra_large_clamp*	19.6	8.8	-	35.3	90.0
061_foam_brick*	54.5	34.7	-	61.1	71.9
MEAN	21.3	39.0	53.9	49.1	**60.1**

Table B.1: **Detailed results on YCB-V [17] w.r.t. ADD(-S).** P.E. means whether the method is trained with 1 pose estimator for the whole dataset or 1 per object (N objects in total). (*) denotes symmetric objects and "-" denotes unavailable results.

D. Qualitative Results

We demonstrated additional qualitative results for LM [2], LM-O [1], and YCB-V [17] in Fig. D.1, Fig. D.2 and Fig. D.3, respectively. Thereby, in Fig. D.1, we visualize the 6D pose by overlaying the image with the corresponding transformed 3D bounding box. In Fig. D.2 and Fig. D.3, we illustrate the estimated 6D poses by rendering the 3D models on top of the input image and highlighting the respective contours. Note that while *Blue* constitutes the ground-truth poses, we demonstrate in *Green* the predicted poses from GDR-Net. For better visualization we cropped the images and zoomed into the area of interest.

References

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *European Conference on Computer Vision (ECCV)*, pages 536–551, 2014. 1, 2, 3, 6

[2] 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. In *Asian Conference on Computer Vision (ACCV)*, pages 548–562, 2012. 1, 2, 5

[3] Tomas Hodan, Daniel Barath, and Jiri Matas. EPOS: Estimating 6D Pose of Objects with Symmetries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11703–11712, 2020. 1, 3

[4] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. BOP: Benchmark for 6D Object Pose Estimation. In *European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 1

[5] Tomas Hodan, Martin Sundermeyer, Bertram Drost, Yann Labbe, Eric Brachmann, Frank Michel, Carsten Rother, and Jiri Matas. BOP Challenge 2020 on 6D Object Localization. *European Conference on Computer Vision Workshops (ECCVW)*, 2020. 1, 3

[6] 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. 1, 2

[7] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6D Object Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3385–3394, 2019. 2

[8] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent Multi-view Multi-object 6D Pose Estimation. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[9] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision (IJCV)*, 81(2):155, 2009. 1

[10] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation.

Method	w/o Refinement							w/ Refinement			
	PoseCNN [17]		PVNet [13]	GDR-Net (**Ours**)				DeepIM [10]		CosyPose [8]	
P.E.	1		N	1		N		1		1	
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	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
002_master_chef_can	84.0	50.9	81.6	96.6	71.1	96.3	65.2	93.1	71.2	-	-
003_cracker_box	76.9	51.7	80.5	84.9	63.5	97.0	88.8	91.0	83.6	-	-
004_sugar_box	84.3	68.6	84.9	98.3	93.2	98.9	95.0	96.2	94.1	-	-
005_tomato_soup_can	80.9	66.0	78.2	96.1	88.9	96.5	91.9	92.4	86.1	-	-
006_mustard_bottle	90.2	79.9	88.3	99.5	93.8	100.0	92.8	95.1	91.5	-	-
007_tuna_fish_can	87.9	70.4	62.2	95.1	85.1	99.4	94.2	96.1	87.7	-	-
008_pudding_box	79.0	62.9	85.2	94.8	86.5	64.6	44.7	90.7	82.7	-	-
009_gelatin_box	87.1	75.2	88.7	95.3	88.5	97.1	92.5	94.3	91.9	-	-
010_potted_meat_can	78.5	59.6	65.1	82.9	72.9	86.0	80.2	86.4	76.2	-	-
011_banana	85.9	72.3	51.8	96.0	85.2	96.3	85.8	91.3	81.2	-	-
019_pitcher_base	76.8	52.5	91.2	98.8	94.3	99.9	98.5	94.6	90.1	-	-
021_bleach_cleanser	71.9	50.5	74.8	94.4	80.5	94.2	84.3	90.3	81.2	-	-
024_bowl*	69.7	69.7	89.0	84.0	84.0	85.7	85.7	81.4	81.4	-	-
025_mug	78.0	57.7	81.5	96.9	87.6	99.6	94.0	91.3	81.4	-	-
035_power_drill	72.8	55.1	83.4	91.9	78.7	97.5	90.1	92.3	85.5	-	-
036_wood_block*	65.8	65.8	71.5	77.3	77.3	82.5	82.5	81.9	81.9	-	-
037_scissors	56.2	35.8	54.8	68.4	43.7	63.8	49.5	75.4	60.9	-	-
040_large_marker	71.4	58.0	35.8	87.4	76.2	88.0	76.1	86.2	75.6	-	-
051_large_clamp*	49.9	49.9	66.3	69.3	69.3	89.3	89.3	74.3	74.3	-	-
052_extra_large_clamp*	47.0	47.0	53.9	73.6	73.6	93.5	93.5	73.3	73.3	-	-
061_foam_brick*	87.8	87.8	80.6	90.4	90.4	96.9	96.9	81.9	81.9	-	-
MEAN	75.9	61.3	73.4	89.1	80.2	**91.6**	**84.3**	88.1	81.9	89.8	**84.5**

Table B.2: **Detailed results on YCB-V [17] w.r.t. AUC of ADD-S and ADD(-S).** As in [17], ADD-S uses the symmetric metric for all objects, while ADD(-S) only uses the symmetric metric for symmetric objects. P.E. means whether the method is trained with 1 pose estimator for the whole dataset or 1 per object (N objects in total). (*) denotes symmetric objects and "-" denotes unavailable results.

Method	Ref.	LM-O [1]				YCB-V [17]				Time (s)
		AR_{MSPD}	AR_{MSSD}	AR_{VSD}	AR	AR_{MSPD}	AR_{MSSD}	AR_{VSD}	AR	
AAE [15]		25.4	9.5	9.0	14.6	41.0	41.3	30.7	37.7	0.190
Pix2Pose [12]		55.0	30.7	23.3	36.3	57.1	42.9	37.2	45.7	1.168
EPOS [3]		65.9	38.0	29.0	44.3	78.3	67.7	62.6	69.6	0.530
CDPNv2 [11]		*81.5*	*61.2*	44.5	62.4	63.1	57.0	39.6	53.2	*0.153*
GDR-Net (**Ours**)		**86.4**	**65.2**	**50.2**	**67.2**	*84.2*	*75.6*	*66.8*	*75.5*	**0.065**
CosyPose [3]	✓	81.2	60.6	*48.0*	*63.3*	**85.0**	**84.2**	**77.2**	**82.1**	0.395

Table C.3: **Results on LM-O and YCB-V under BOP [5] setup.** The results for other methods are obtained from https://bop.felk.cvut.cz/leaderboards/. The time (s) is the average image processing time averaged over the datasets. Ref. stands for refinement. For each column, we denote the best score in **bold** and the second best score in *italics*.

International Journal of Computer Vision (IJCV), pages 1–22, 2019. 1, 3

[11] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7678–7687, 2019. 1, 3

[12] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7668–7677, 2019. 1, 3

[13] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4561–4570, 2019. 1, 3

[14] Chen Song, Jiaru Song, and Qixing Huang. HybridPose: 6D Object Pose Estimation Under Hybrid Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 431–440, 2020. 1

[15] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 3

[16] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS:

Fully Convolutional One-Stage Object Detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9627–9636, 2019. 1

[17] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *Robotics: Science and Systems (RSS)*, 2018. 1, 2, 3, 7

[18] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6D Pose Object Detector and Refiner. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1941–1950, 2019. 1

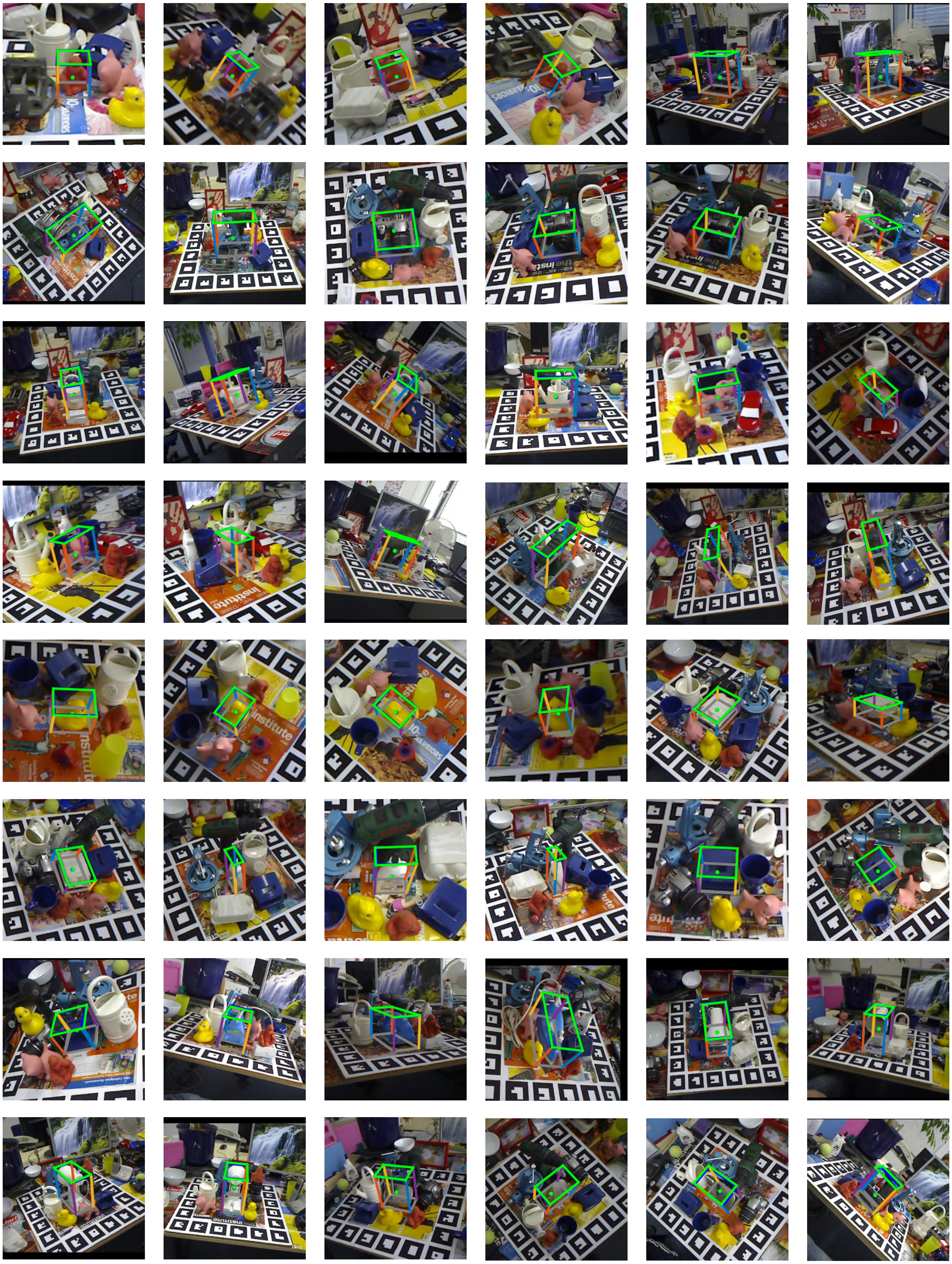

Figure D.1: **Qualitative Results on LM [2].** We visualize the 6D pose by overlaying the image with the corresponding transformed 3D bounding box. We demonstrate in *Blue* and *Green* the ground-truth pose and the predicted pose, respectively.

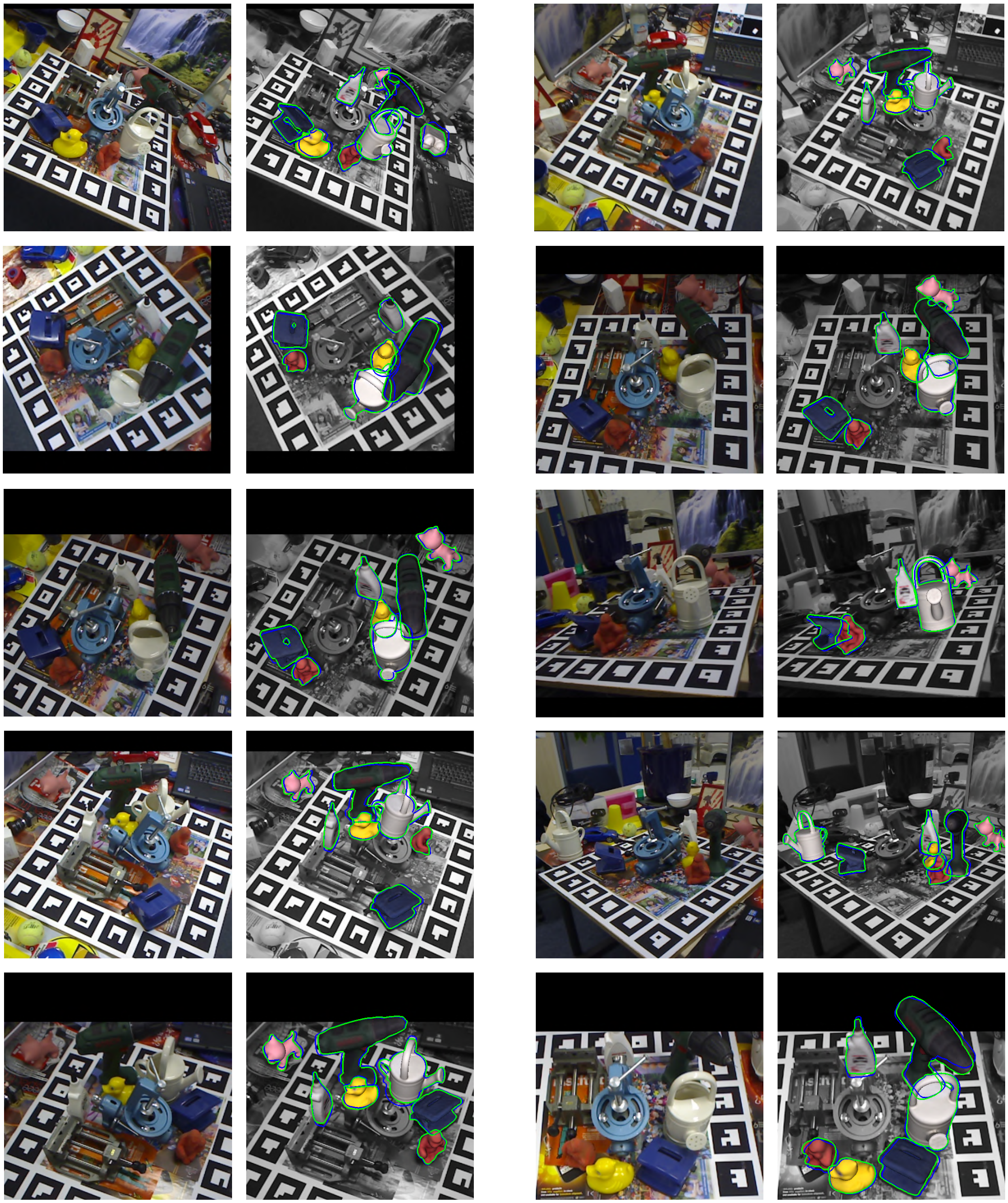

Figure D.2: **Qualitative Results on LM-O [1].** For each image, we visualize the 6D poses by rendering the 3D models and overlaying the contours on the right. We demonstrate in *Blue* and *Green* the ground-truth pose and the predicted pose, respectively.

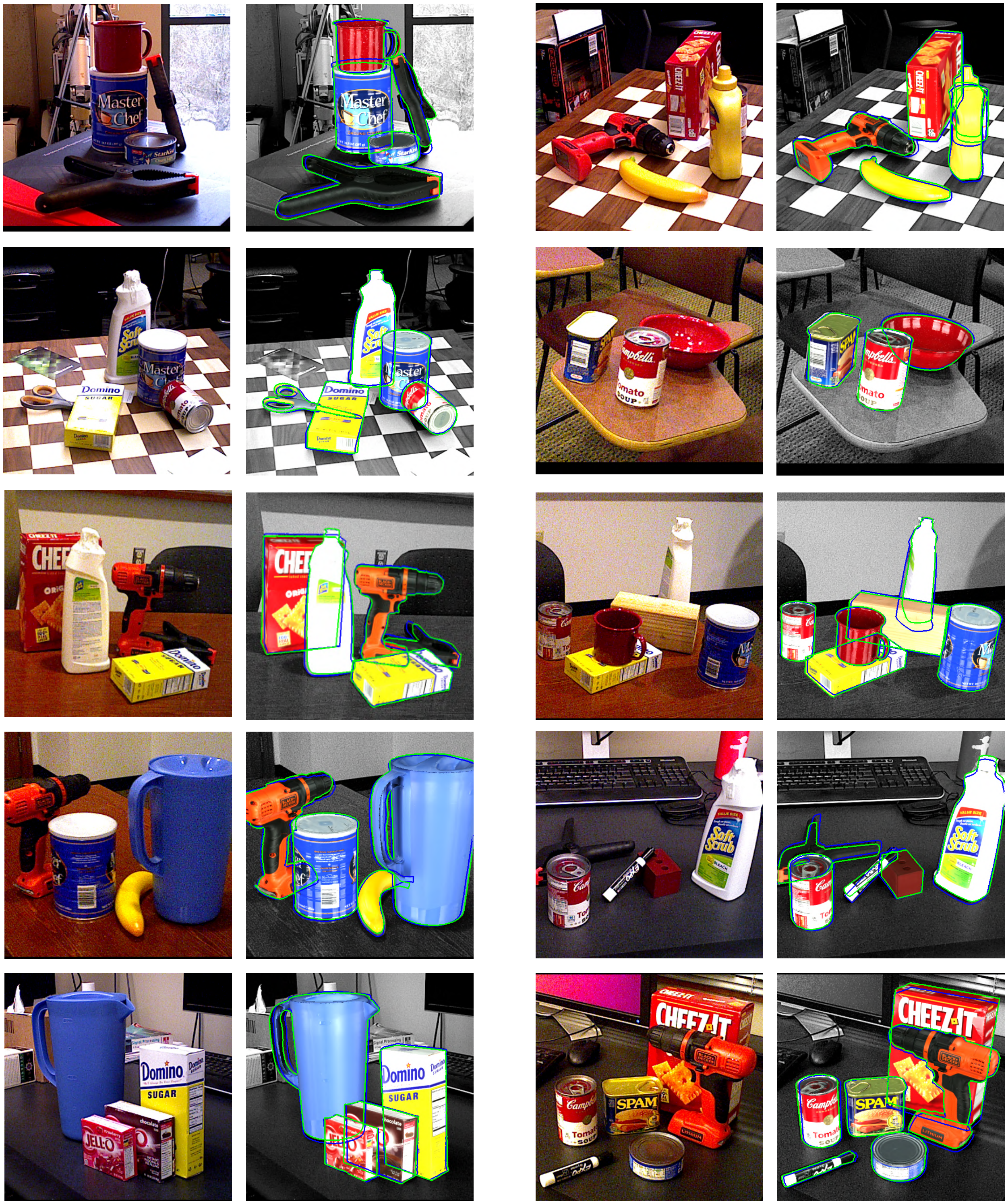

Figure D.3: **Qualitative Results on YCB-V [17].** For each image, we visualize the 6D poses by rendering the 3D models and overlaying the contours on the right. We demonstrate in *Blue* and *Green* the ground-truth pose and the predicted pose, respectively.