COPE: End-to-end trainable Constant Runtime Object Pose Estimation Supplementary Material

Stefan Thalhammer
Automation and Control Institute, TU Vienna
Gusshausstrasse 27-29
thalhammer@acin.tuwien.ac.at

Timothy Patten
Robotics Institute, UTS
81 Broadway, Building 11
timothy.patten@uts.edu.au

Markus Vincze
Automation and Control Institute, TU Vienna
Gusshausstrasse 27-29
vincze@acin.tuwien.ac.at

This supplementary manuscript provides the reader with an overview of the network layout in Section 1, additional training details in Section 2, discussion of the performance in comparison to the state of the art in Section 3, a highlight of the capabilities of COPE in Section 4 and concludes with a presentation of error cases in Section 5.

1. Network Design

ResNet101 [5] in conjunction with PFPN [12] is used for multi-scale feature extraction. This is followed by three modules, one for location classification, bounding box estimation and keypoint estimation that are shared over feature pyramid levels. Each of the modules consist of four convolution layers with Mish [11] activation. The convolution layers of the classification module have 256 feature channels and those of the bounding box and the keypoint estimation module have 512 feature channels. Linearly activated output layers convolve the feature maps to a one-hot encoding for the object class, bounding box corners and the number of keypoints. Feature maps are not spatially reduced when passing through these modules. The estimated keypoints are fed to a module that directly learns 6D pose estimation, consisting of two convolution layers with 512 and 256 feature channels with Mish activation as well as a linearly activated output convolution with $a{\cdot}9$ output parameters, where a is the number of classes. Thus, the outputs are 3 for translation and 6 for rotation for each dataset object separately. The parameter d in Equation 2 of the submitted manuscript is set to 3 in all experiments.

2. Training Details

2.1. Backbone

The performance reduction occurring when estimating poses under domain shift is partially alleviated by setting low-level stages of the backbone to non-trainable [8, 7, 14]. EPOS [8] freezes the majority of the backbone, i.e. early and middle flow of Xception-65 [2] when training exclusively on synthetic data. We experienced this strategy to be infeasible for feature pyramid-based approaches since-prediction are made from different feature map resolutions taken also from early and intermediate feature maps of the backbone. Thus, freezing the weights of all layers up to the output layers reduces the pose estimation performance since feature learning for the present task is limited. Hence, further investigating the adaption of the backbone to have deeper early stages might lead to improved domain transfer.

2.2. Hyperparameters

All results are presented with the same set of hyperparameters, the only exception is Table 2 in the submitted manuscript. For the ablations, the parameter settings are mentioned in the table.

Training: The relevant hyperparameters are, apart from the network layout and optimization itself, the base of the logarithm for location sampling, the visibility threshold for foreground samples and the color and image space augmentation hyperparameters. A visual comparison for location sampling on physically-based rendering images of the LM [6] objects is provided in Figure 1. The top row shows our physics-based scheme, the bottom shows using all pyramid levels for training. Without providing quantitative analysis it is already visible that using all feature pyramid levels

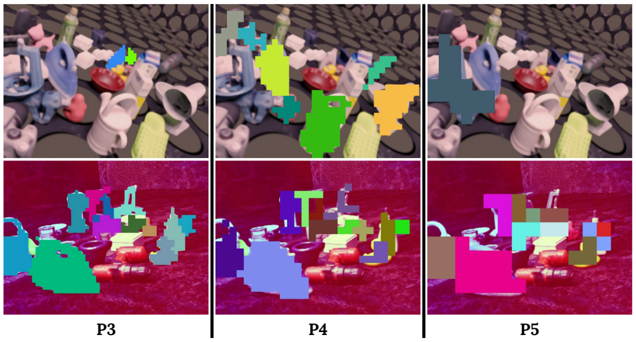

Figure 1. **Comparison of Training Location Sampling:** Utilizing the object dimensions for sampling the pyramid level to train on (**top**) leads to a similar amount of true locations per object and prevents ambiguous training locations. Sampling small objects in coarse feature map resolutions results in aliasing effects that are detrimental to convergence (**bottom**). Images are cropped to improve visibility. Best viewed on a screen.

leads to aliasing effects in the lower resolved pyramid levels, while explicitly using object depth and dimension samples a similar amount of true locations per object.

Training on objects with too much occlusion is detrimental to pose estimation performance. To overcome this issue we set the threshold for foreground samples to 0.25 of the object visibility when computing L_{cls}; all other losses are computed for objects with at least 0.5 object visibility. We apply affine color space transformations to improve the domain transfer, as also mentioned in the manuscript, parameters and ranges are provided in Table 1. Additionally we randomly scale training images by 5% to improve translation equivariance of our trained models.

Inference: Parameters that require manual assignment are the detection threshold, the Intersection-over-Union (IoU) for clustering hypotheses, number of hypotheses to use for the pose computation per instance and the maximal number of instances per image. All image locations with a detection threshold above 0.5 are considered as foreground, thus as true locations containing objects of interest and are consequently used for hypotheses clustering and pruning. An IoU of 0.5 is used for clustering instance hypotheses. An ablation for the amount of hypotheses to derive the final pose is presented in Table 3 in the submitted manuscript. The hyperparameter is set to 10 for all experiments apart from those ablating its influence. For the setting of the manuscript, the number of maximal instances to detect is set to 100. Yet, despite there being little restrictions for that parameter, increasing this threshold contributes little to nothing since 100 instances to detect per image is already a considerably large number. After clustering and pruning hypotheses, those that are not supported by another hypothesis are discarded.

2.3. Evaluation Metrics

Comparison to the state of the art is provided using the performance score of the BOP challenge [9]. The deviation of the estimated pose $\hat{P}$ to the ground truth P is projected to a scalar value using the average recall of three error metrics. These are the Visual Surface Discrepancy, the Maximum Symmetry-Aware Surface Distance and the Maximum Symmetry-Aware Projection Distance:

$$e_{VSD} = \underset{p \in \hat{V} \cup V}{avg} \begin{cases} 0 & \text{if } p \in \hat{V} \cap V \wedge |\hat{D}(p) - D(p)| < \tau \\ 1 & \text{otherwise} \end{cases},$$

$$e_{MSSD} = \underset{s \in S_i}{min}\ \underset{m \in M_i}{max} ||\hat{P}m - Ps||_2,$$

$$e_{MSPD} = \underset{s \in S_i}{min}\ \underset{m \in M_i}{max} ||proj_{3D \rightarrow 2D}(\hat{P}m) - proj_{3D \rightarrow 2D}(Psm)||_2, \quad (1)$$

where $\hat{V}$ and V are sets of image pixels; $\hat{D}$ and D are distance maps and τ is a misalignment tolerance. Distance maps are rendered and compared to the distance map of the test image to derive $\hat{V}$ and V. S_i is a set of symmetry transformations that depend on the visual ambiguities of the object mesh. M_i is a subset of the mesh vertices and $proj_{3D \rightarrow 2D}(.)$ denotes the projection to the image space. For each of these metrics the average recall (AR) is measured when comparing errors to multiple error thresholds (and τ in the case of e_{VSD}). Results are then reported as the Average Recall: $AR = (AR_{VSD} + AR_{MSSD} + AR_{MSPD})/3$.

Ablations are evaluated using the ADD(-S) recall [6]:

$$e_{ADD} = \underset{m \in M_i}{avg} ||\hat{P}m - Pm||, \quad (2)$$

$$e_{ADDS} = \underset{m_1 \in M_i}{avg}\ \underset{m_2 \in M_i}{min} ||\hat{P}m_1 - Pm_2||. \quad (3)$$

ADD measures the average deviation of models points using the corresponding point distance. For objects exhibiting symmetric transformations, the ADD-S error, using the closest point distance, is calculated. We report the fraction of poses below the commonly used error threshold of 10% of the object diameter.

Results for object detection are reported using the the mean Average Precision (mAP) of the Microsoft COCO object detection challenge [10].

3. Comparing Detection Performance

The results reported in Table 2 in the submitted manuscript indicate that the detection performance of COPE is inferior to that of FCOS [13]. However, this conclusion has to be drawn with caution since FCOS only performs 2D Detection. The network size of FCOS is ~ 50 million parameters just for object detection while COPE additionally predict geometric correspondences and direct 6D

Table 1. Color space augmentations applied during training.

Augmentation	Chance (per channel)	Range
gaussian blur	0.2	$\sigma \sim \mathcal{U}(0.0, 2.0)$
average/median/motion blur	0.2	$\sigma \sim \mathcal{U}(3, 7)$
bilateral blur	0.2	$\sigma \sim \mathcal{U}(1, 7)$
hue/saturation	0.5	$\mathcal{U}(-15, 15)$
grayscale	0.5	$\mathcal{U}(0.0, 0.2)$
add	0.5 (0.5)	$\mathcal{U}(-0.04, 0.04)$
multiply	0.5 (0.5)	$\mathcal{U}(0.75, 1.25)$
gamma contrast	0.5 (0.5)	$\mathcal{U}(0.75, 1.25)$
sigmoid contrast	0.5 (0.5)	$\mathcal{U}(0, 10)$
logarithmic contrast	0.5 (0.5)	$\mathcal{U}(0.75, 1.0)$
linear contrast	0.5 (0.5)	$\mathcal{U}(0.7, 1.3)$

poses with only $\sim$ 17 million parameters more. Additionally, FCOS uses an input image resolution with up to 1333 pixels for the larger image side while COPE uses 640×480 input images. Thus, COPE solves twice as many tasks with higher complexity from images with half of the input resolution.

4. Highlights of COPE

Handling Multiple Mutually Occluding Objects The top row of Figure 2 shows accurate bounding box and pose estimates on IC-BIN's [4] *Juice*. Due to the end-to-end trainability and the parallel learning of detection and pose estimation, COPE learns to effectively handle multiple mutually occluding instances of the same object. Increased mutual occlusion of instances of *Coffeecup* is displayed in the middle row, which again shows accurate bounding box and pose estimates for all the visible object instances. The bottom row shows a scenario where both *Juice* and *Coffeecup* are present. Ultimately, a false positive detection of *Juice* occurs in the center of the bulk due to the heavy mutual occlusion of multiple instances.

Handling Occlusion in Clutter Figure 3 displays accurate bounding box and pose estimates on occluded examples of LM-O's [1] *Ape*, *Can* and *Eggbox*. The middle row shows similarly occluded examples of *Drill*, *Holepunch* and *Glue* and the bottom row for *Cat* and *Duck*. Despite only training one model for all of LM's objects COPE is robustly handling each object, even under occlusion.

5. Error Cases

Figure 4 presents recurring errors on IC-BIN. The top row shows an instance of *Juice* in top-view not being detected, indicated with a red and white circle. Despite the high relative visibility of *Juice* these reduced views are not often sampled during training data generation and are thus difficult to detect during runtime. The middle row displays a similar error occuring for *Coffeecup*, again indicated with red and white circles. Multiple top-view orientated instances are not detected and thus result in false negative detections. The bottom row shows one instance of each *Coffeecup* and *Juice* not being detected despite providing rich visual features. Assigning true training locations on objects with too low visibility leads to reduced performance during inference. As such we treat objects that are largely occluded, i.e. with less than 25% relative object visibility, as background during training. However, this leads to cases where discriminative object portions are visible in the image, yet are treated as background, as can be seen here. Further investigation is required to consider the abundance of features during training target sampling to overcome such issues.

Figure 5 presents common error cases on LM-O, again indicated with red and white circles. The top row displays a false positive detection of the *Holepunch* on a toy car with the same color and very similar material properties as the object of interest. The middle row shows a similar false-positive detection of the object *Duck*. In the bottom row an example of *Eggbox* with an occlusion pattern that is unlikely to be similarly sampled when randomizing object placements using physical modelling is displayed. Since it is unlikely that objects of roughly the same size end up being stacked, these cases are not experienced during training [3].

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 *Proceedings of the European Conference on Computer Vision*, pages 536–551, 2014.

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

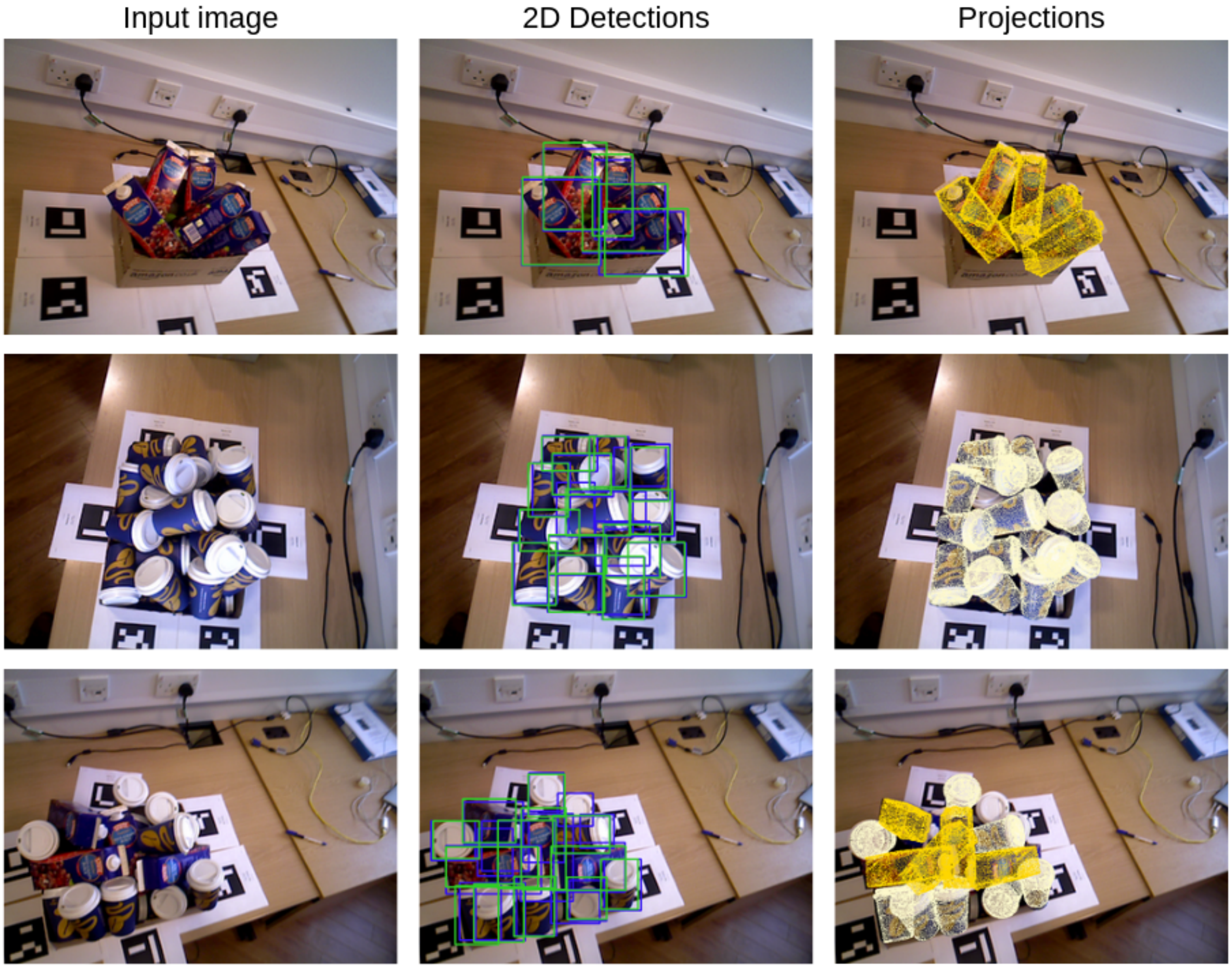

Figure 2. **Multiple Object Instances:** Columns are, from left to right, input image, 2D detections and reprojected object mehses based on the estimated poses. Each instance is indicated with a specific color. Green and blue bounding boxes correspond to estimates and ground truth, respective. Best viewed on screen.

[3] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *CoRR*, abs/1911.01911, 2019.

[4] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3583–3592, 2016.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] 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 *Proceedings of the Asian Conference on Computer Vision*, pages 548–562, 2012.

[7] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In *Proceedings of the European Conference on Computer Vision Workshops*, pages 0–0, 2018.

[8] 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*, pages 11703–11712, 2020.

[9] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *Proceedings of the European Conference on Computer Vision Workshops*, 2020.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

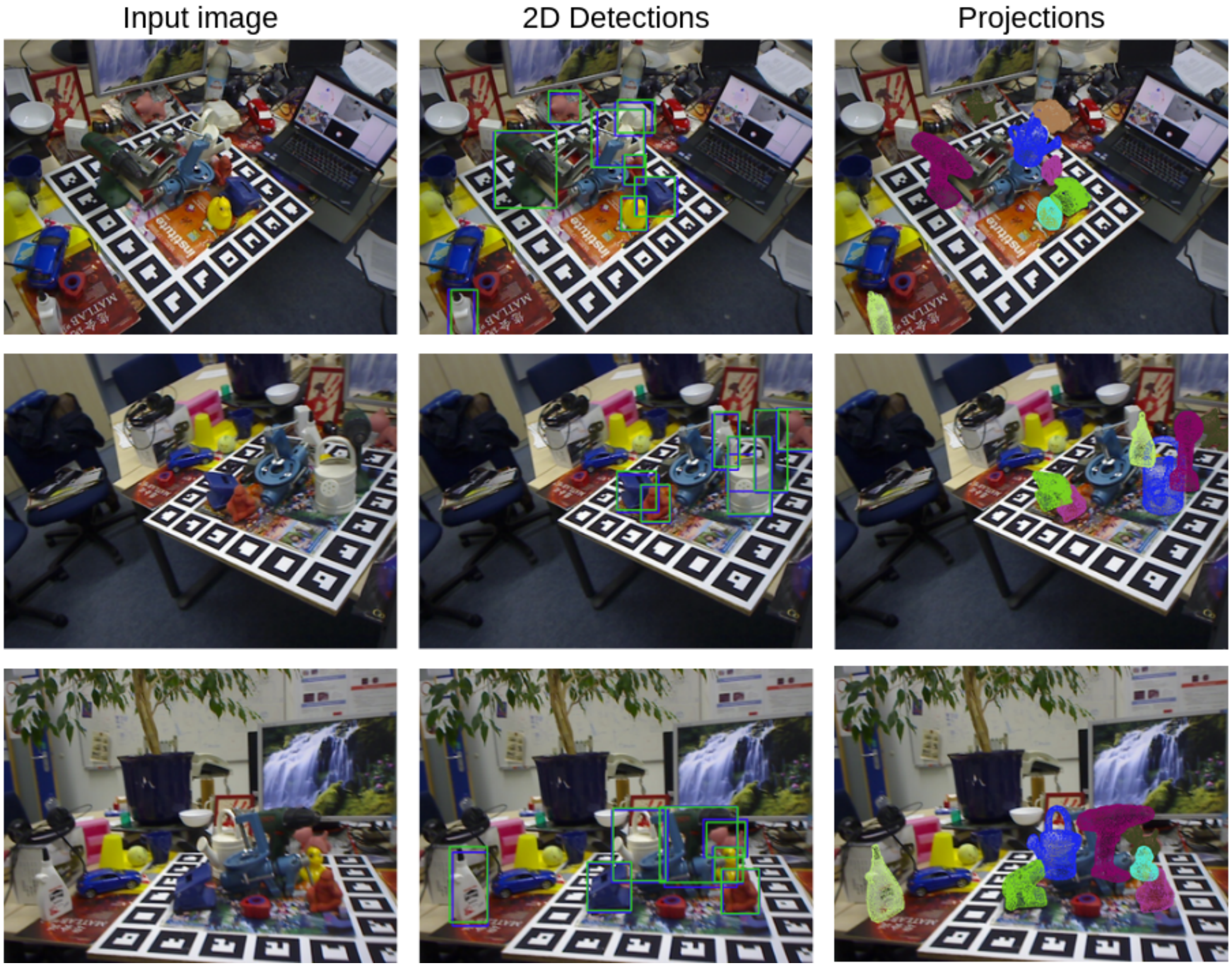

Figure 3. **Occlusion Handling on LM-O:** Columns are, from left to right, input image, 2D detections and reprojected object mehses based on the estimated poses. Each instance is indicated with a specific color. Green and blue bounding boxes correspond to estimates and ground truth, respective. Best viewed on screen.

[11] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[12] Stefan Thalhammer, Markus Leitner, Timothy Patten, and Markus Vincze. Pyrapose: Feature pyramids for fast and accurate object pose estimation under domain shift. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2021.

[13] 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.

[14] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019.

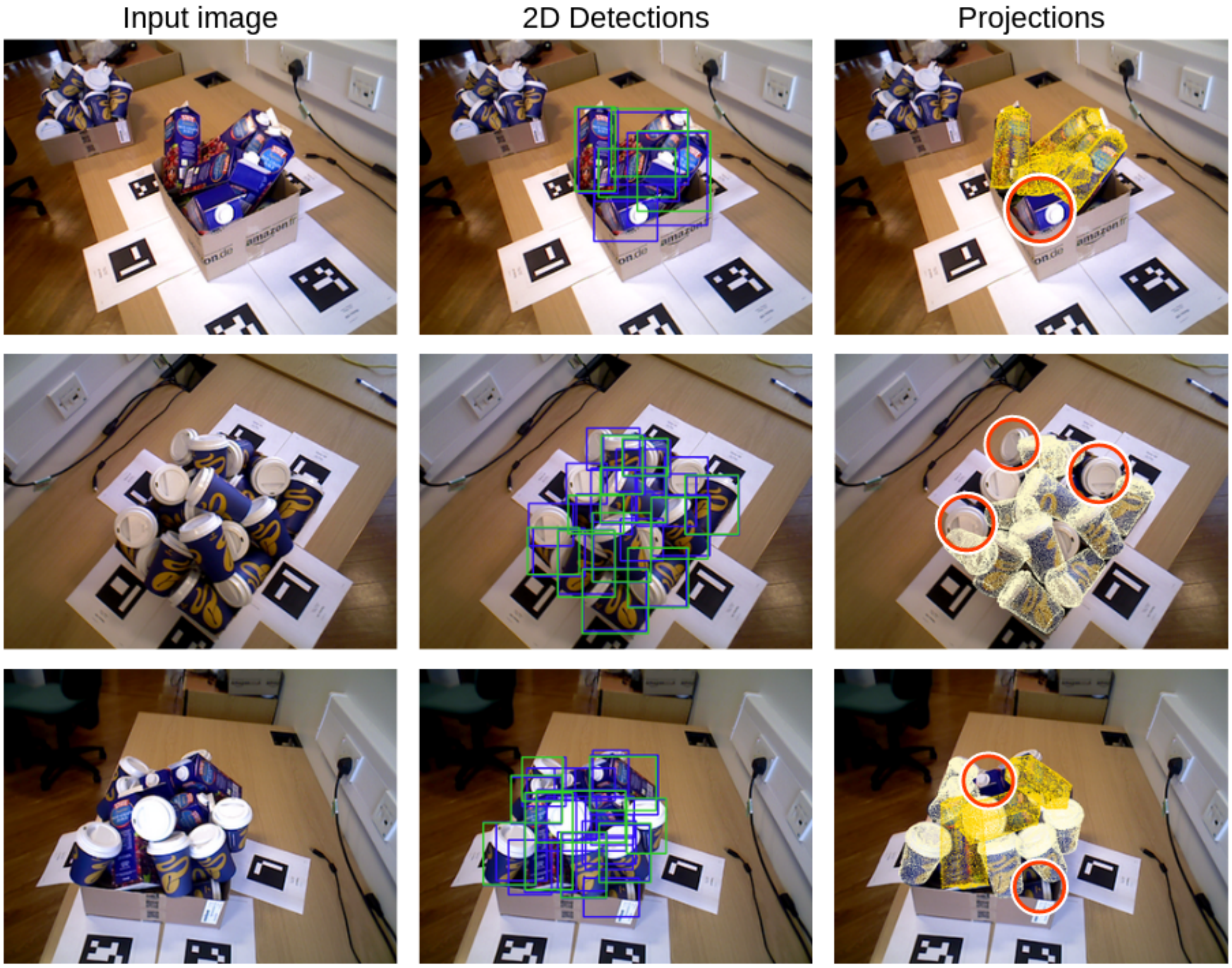

Figure 4. **Error Cases on IC-BIN:** Columns are, from left to right, input image, 2D detections and reprojected object mehses based on the estimated poses. Each instance is indicated with a specific color. Green and blue bounding boxes correspond to estimates and ground truth, respective. Errors are indicated with a red and white circle. Best viewed on screen.

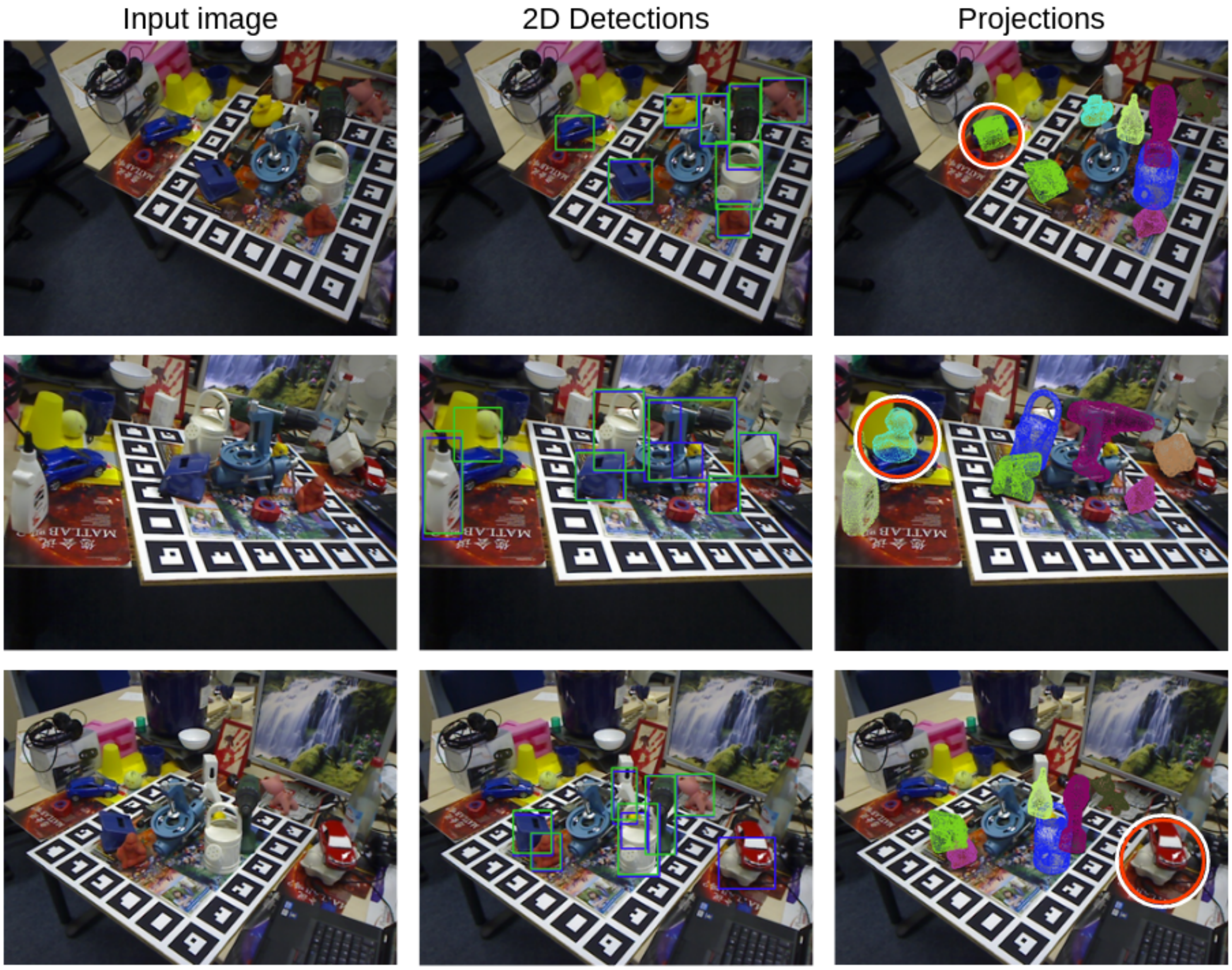

Figure 5. **Error Cases on LM-O:** Columns are, from left to right, input image, 2D detections and reprojected object mehses based on the estimated poses. Each instance is indicated with a specific color. Green and blue bounding boxes correspond to estimates and ground truth, respective. Errors are indicated with a red and white circle. Best viewed on screen.