OSOP: A Multi-Stage One Shot Object Pose Estimation Framework
—
Supplementary Material

Ivan Shugurov[1,3], Fu Li[1,2], Benjamin Busam[1], Slobodan Ilic[1,3]

[1] TU Munich [2] NUDT [3] Siemens AG
{ivan.shugurov, fu.li, b.busam, slobodan.ilic}@tum.de

1. Pose Hypothesis Selection

Pose hypothesis selection is done by rendering the object in the predicted pose and measuring a VSD-like pose consistency score [8]. If only RGB data is available, the score is defined over per-pixel correspondence error. If depth is available, per-pixel depth discrepancy is measured. For consistency, depth values and object sizes are expressed in millimeters.

Similarly to the main paper, let $C : \mathcal{M} \times SE(3) \rightarrow [0,1]^{W \times H \times 3}$ denote 2D-3D correspondences for the object rendered in the given pose. Correspondences are defined as Normalized Object Coordinates (NOCS) [19]. Its inverse C^{-1} recomputes correspondences to the unormalized object coordinates in the object coordinate system. Let $\hat{S}$ be a predicted binary segmentation with $\hat{S}_p$ indicating whether a pixel p belongs to the object or not. Function $S : \mathcal{M} \times SE(3) \rightarrow [0,1]^{W \times H}$ renders a segmentation mask for a given object pose. Function $D : \mathcal{M} \times SE(3) \rightarrow \mathbb{R}^{W \times H}$ renders per-pixel depth for a given object pose, whereas $\hat{D}$ represents the observed depth map. $\mathbf{T} \in SE(3)$ is object pose.

Then, in case of RGB, pose consistency is defined as

$$\text{cons}(\tau) = \underset{p \in \hat{S} \cap S(\mathbf{T})}{\text{avg}} \begin{cases} 1, & \text{if } \left\| \hat{C}_p - C^{-1}\left(C\left(\mathbf{T}\right)_p\right) \right\|_2 < \tau \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

In case of depth, pose consistency is defined as

$$\text{cons}(\tau) = \underset{p \in \hat{S} \cap S(\mathbf{T})}{\text{avg}} \begin{cases} 1, & \text{if } \left\| \hat{D}_p - D(\mathbf{T})_p \right\|_2 < \tau \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

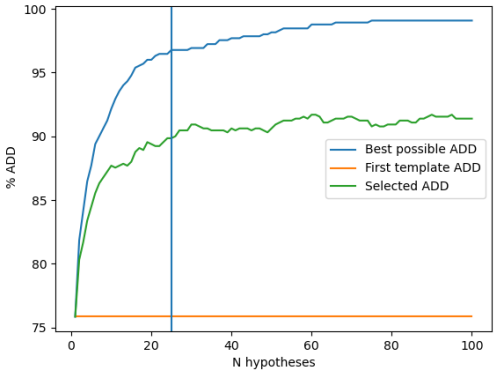
Consistency is averaged over thresholds τ from 1 to 5mm, and the pose hypothesis with the highest average consistency is selected as the final pose.

Ablation studies presented in Figures 1 and 2 illustrate how the ADD score changes depending on the minimal distance between templates in the set of pose hypotheses and the number of hypotheses. The experiments were conducted on a small random subset of the data. "First template ADD" denotes the ADD reached on the subset of data using the standard OSOP pipeline without multiple hypothesis. "Best possible ADD" denotes the average ADD among all images in the subset, where for each image the best ADD among the predicted poses was picked. This sets the upper bound on what ADD the method can reach with the given number of pose hypothesis. "Selected ADD" denotes the ADD of the poses chosen with the proposed pose selection method. Experiments on Linemod and Occlusion demonstrate that the 15 degrees threshold on the distance between templates works the best, as it eliminates duplicate templates, that have a low angular distance from each other, and ensures more diverse hypothesis set. The plots also show that the ADD of chosen poses stops improving after approximately 25 pose hypotheses. Therefore, we used the threshold of 15 degrees and 25 templates in the experiments.

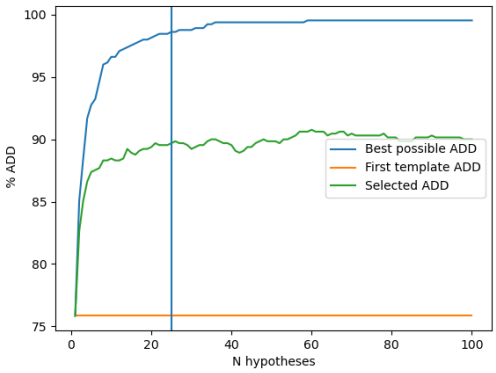
2. Architecture

In all experiments we used ResNet50 as the feature extractor. For the first and the third stages, we use feature maps after layers number 10, 22 and 40. The network is trained and tested on the full resolution images of size 480×640. For the image descriptor, this corresponds to $\mathbf{f}^k$ of sizes $120 \times 160 \times 256$, $60 \times 80 \times 512$ and $30 \times 40 \times 1024$. We used 2880 templates for the localization network, which corresponds to 576 camera locations with 5 in-plane rotations. The object descriptors $\mathbf{o}^k$ thus has the dimensions $16 \times 36 \times 5 \times 256$, $16 \times 36 \times 5 \times 512$ and $16 \times 36 \times 5 \times 1024$. The detailed network architecture is visualized in Figure 4 with the detailed architecture of the attention block visualized in Figure 3.

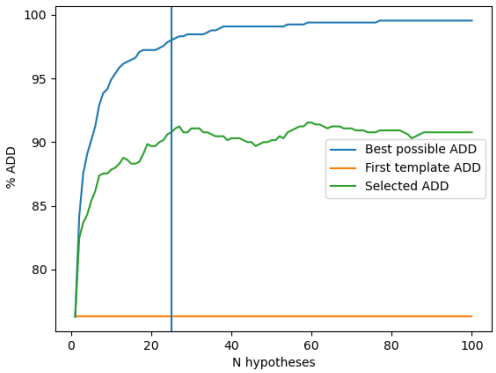
For the second stage, we used approximately 90K templates as suggested in AAE [17, 18], and map each of them to the latent space of size $8 \times 8 \times 256$. Even though the resulting descriptor is of higher dimension than in [18, 22], all



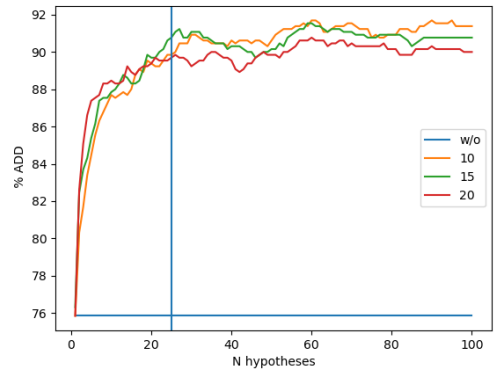
(a) 10 degrees template threshold.



(b) 20 degrees template threshold.



(c) 15 degrees template threshold.



(d) Comparison of ADD with different template thresholds.

Figure 1. Ablation studies on pose hypothesis selection on Linemod dataset dataset [6].

descriptors for all templates still fit on a single GPU, which enables fast inference. During training, we convert the rotation matrix from the egocentric to allocentric coordinate system following [13]. This conversion ensures that the visual appearance of the object is dependent exclusively on the rotational component of the $SE(3)$ pose. The angle between rotations is computed as an arcos of quaternions representing them. Symmetric objects are ignored during training. The third stage operates on images of sizes 128×128. The detailed network architecture is visualized in Figure 5.

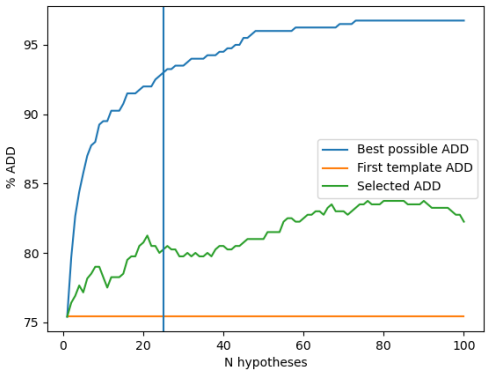
3. Implementation Details

The detector was implemented using Pytorch [15]. A pre-trained ResNet50 [5] served as feature extractor F_{FE} in all three stages. The feature extractor was unchanged for the segmentation and 2D-2D correspondence networks to overcome the domain gap problem. Only the last block of the ResNet was fine-tuned for the second stage in the proposed pipeline. We used the MAGSAC [1] implementation from OpenCV [3] and point-to-plane ICP from Open3D [24]. All our experiments were conducted on an Intel Core i9-9900K CPU 3.60GHz with NVIDIA Geforce RTX 2080 TI GPU. We trained the networks with the Adam optimizer [11]. The localization network and the 2D-2D matching network were trained for 50 epochs which took approximately one day on a single GPU. The second and third stage networks were trained for 10 epochs, which took approximately 2 hours. We render templates at 25 FPS with 128x128 resolution and models down-sampled to 5K faces. It takes around one hour to render 90K templates, 40 seconds to render 1K templates and 200 seconds to render 5K templates.

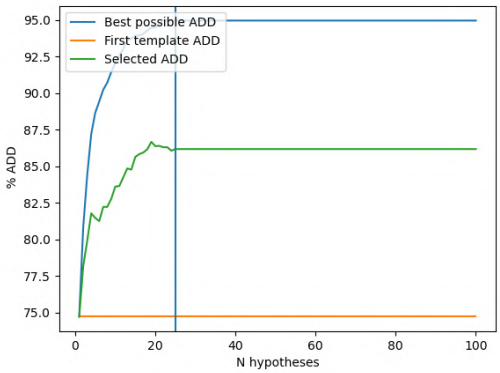
4. Additional Results and Visualisations

Table 1 provides per-object add score of the proposed and state of the art methods on the Linemod dataset.

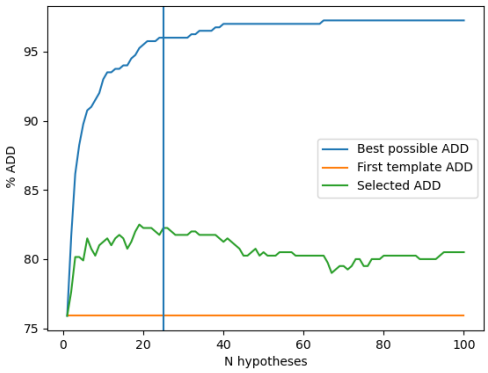
Table 2 shows evaluation of our method on the TLESS dataset [7]. We followed the Multi-Path AAE [17] evaluation pipeline and ran the 2nd and the 3rd stages of OSOP on detections from Multi-Path AAE. OSOP convincingly outperforms Multi-Path AAE and PPF on RGB and RGBD data respectively. This proves that the matching strategy does not suffer from object symmetries and heavy occlu-



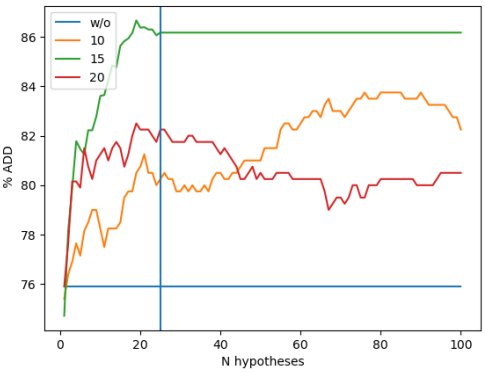
(a) 10 degrees template threshold.



(c) 15 degrees template threshold.



(b) 20 degrees template threshold.



(d) Comparison of ADD with different template thresholds.

Figure 2. Ablation studies on pose hypothesis selection on Occlusion dataset dataset [2]

Table 1. Percentages of correctly estimated poses w.r.t. the ADD on the Linemod [6] dataset for methods trained on synthetic data. In case of only RGB input, our method relies on PnP+RANSAC to estimate the pose. Kabsch+RANSAC is used if RGBD data is available.

	RGB								RGBD						
Method	**SSD6D** [10]	**PfS** [21]	**AAE** [18]	**SSD6D** [10]	**OURS**	**DPOD** [23]	**Ours**	**DPOD** [23]	**AAE** [18]	**Ours**	**Ours**	**PPF** [6]	**Ours**	**Ours**	**SSD6D** [10]
Refinement	-	-	-	**DL** [12]	-	-	**Mult. Hyp.**	**DL** [23]	**ICP**	-	**ICP**	**ICP**	**Mult. Hyp.**	**Mult. Hyp. + ICP**	**ICP**
Ape	2.6	7.5	4	-	22.57	35.1	26.05	**52.12**	24.35	75.64	81.31	**86.50**	85.19	86.09	-
Bvs.	15.1	25.1	20.9	-	50.41	59.4	55.59	**64.67**	89.13	88.05	91.02	70.70	93.82	**94.64**	-
Cam	6.1	12.1	30.5	-	32.30	15.5	**36.21**	22.23	82.1	64.03	65.28	**78.60**	68.94	69.34	-
Can	27.3	11.3	35.9	-	42.89	48.8	52.17	**77.51**	70.82	65.80	68.47	80.20	**80.43**	**80.43**	-
Cat	9.3	15.4	17.9	-	34.43	28.1	42.57	**56.49**	72.18	77.02	80.07	85.40	87.44	**87.96**	-
Driller	12	18.6	24	-	43.94	59.3	49.57	**65.23**	44.87	72.14	76.35	**87.30**	78.45	79.46	-
Duck	1.3	8.2	4.9	-	20.08	25.6	22.16	**49.04**	54.63	68.28	79.9	46.00	88.27	**92.73**	-
Eggbox	2.8	100	81	-	**73.50**	51.2	72.38	62.21	96.62	98.09	**98.17**	97.00	98.00	**98.17**	-
Glue	3.4	81.2	45.5	-	42.63	34.6	**52.28**	38.94	**94.18**	69.18	69.58	57.20	69.43	69.50	-
Holep.	3.1	18.5	17.6	-	18.19	17.7	18.59	**25.55**	51.25	70.65	74.05	77.40	86.34	**92.72**	-
Iron	14.6	13.8	32	-	69.27	84.7	72.30	**98.43**	77.86	98.25	98.61	84.90	99.39	**99.40**	-
Lamp	11.4	6.5	60.5	-	27.14	45	27.87	**58.35**	86.31	43.52	51.26	**93.30**	39.19	48.49	-
Phone	9.7	13.4	33.8	-	33.63	20.9	**39.58**	33.79	86.24	62.10	63.88	**80.70**	65.81	66.05	-
Mean	9.1	25.5	31.4	34.1	39.31	40.5	43.64	**54.20**	71.58	73.29	76.76	78.86	80.05	81.92	**90.9**
Time (ms)	-	-	24	-	96	36	1343	-	224	60	68	-	722	749	100

sions. We used the same networks as in the Linemod experiments.

Tables 3, 4,5 and 6 provide per-object detection statistics on all four datasets used in the paper.

Figure 6 provides more insights about the detection quality of the proposed method, by comparing it to the Precision-Recall curves of OS2D [14] and YOLOv3 [16]. The plots clearly demonstrate the superiority of our method to OS2D.

Figures 7, 8, 9 10 compare ground truth object outlines with predicted object outlines.

Figure 11, 12 and 13 compare the naive pixel-wise at-

Table 2. Results on the TLESS- dataset [7] reported according to the Average Recall (AR) metric of the BOP challenge [8] on the BOP challenge subset of test images. All methods apart from ours and PPF [4] require prior training on RGB renderings of target objects.

Method	Train data	Refinement	AR
CosyPose	synt	ICP	0.640
EPOS		-	0.467
AAE		ICP	0.487
Multi-Path AAE		-	0.310
DPOD		-	0.081
Ours + Kabsch	-	-	0.532
Drost, PPF		ICP	0.444
Ours + Kabsch		ICP	0.435
Drost, PPF		ICP	0.404
Ours + PnP		-	0.403

tentions versus the thresholded and conditioned attentions as proposed in the paper. This visual comparison proves the the usefullness of the proposed operations as they effectively reduce the noise and allow the network to focus on the true regions of interest.

Figures 14, 15, 16 and 17 provide a visual comparison of predicted 2D-3D correspondences to the ground truth. Correspondences are color-coded according to their NOCS coordinates.

Finally, Figures 18, 19, 20 and 21 demonstrate the quality of estimated poses.

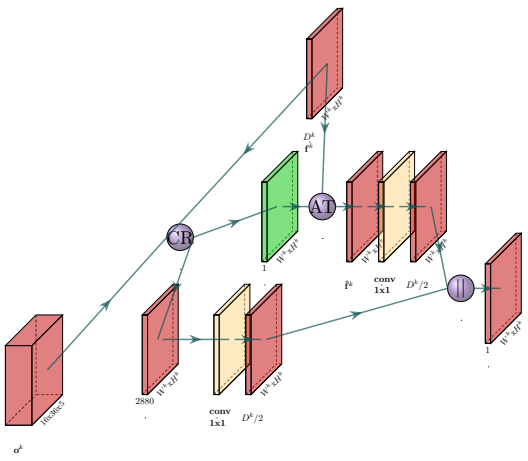


Figure 3. Architecture of the attention block, that stacks features extracted from the attended image features and features from the correlation between the image and object descriptors. Gray color represents ResNet blocks, yellow color represents convolutional layers, red color represent feature maps and descriptors. Better viewed digitally.

References

[1] Daniel Barath, Jiri Matas, and Jana Noskova. MAGSAC: marginalizing sample consensus. In *CVPR*, 2019. 2

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014. 3, 8

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 2

[4] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010. 4, 8

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[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 *ACCV*, 2012. 2, 3, 8

[7] Tomas Hodan, Pavel Haluza, Stepan Obdrzalek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *WACV*, 2017. 2, 4

[8] Tomas Hodan and Antonin Melenovsky. Bop: Benchmark for 6d object pose estimation: https://bop.felk.cvut.cz/home/, 2019. 1, 4

[9] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In *ICCVW*, 2019. 8

[10] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *ICCV*, 2017. 3

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[12] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep model-based 6d pose refinement in rgb. In *ECCV*, 2018. 3

[13] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. Cps++: Improving class-level 6d pose and shape estimationfrom monocular images with self-supervised learning. *arXiv preprint arXiv:2003.05848v3*, 2020. 2

[14] Anton Osokin, Denis Sumin, and Vasily Lomakin. Os2d: One-stage one-shot object detection by matching anchor features. In *ECCV*, 2020. 3, 8

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner,

Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019. 2

[16] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 3, 8

[17] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O Arras, and Rudolph Triebel. Multi-path learning for object pose estimation across domains. In *CVPR*, 2020. 1, 2

[18] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*, 2018. 1, 3

[19] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019. 1

[20] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 8

[21] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from shape: Deep pose estimation for arbitrary 3d objects. *arXiv preprint arXiv:1906.05105*, 2019. 3

[22] Sergey Zakharov, Wadim Kehl, Benjamin Planche, Andreas Hutter, and Slobodan Ilic. 3d object instance recognition and pose estimation using triplet loss with dynamic margin. In *IROS*, 2017. 1

[23] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *ICCV*, 2019. 3

[24] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 2

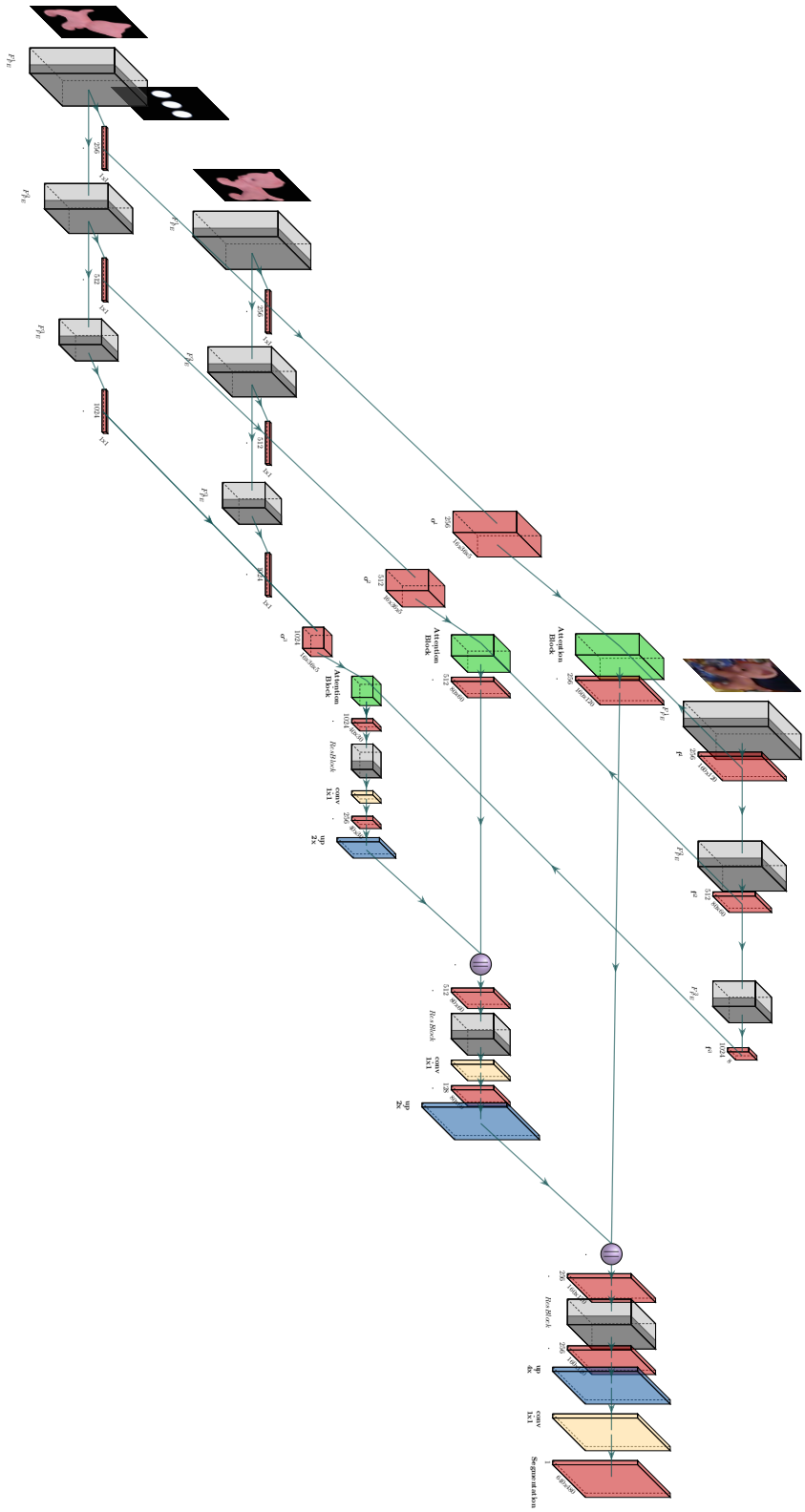


Figure 4. Architecture of the stage 1 network. Gray color represents ResNet blocks, yellow color represents convolutional layers, red color represent feature maps and descriptors and green represents attention blocks. Better viewed digitally.

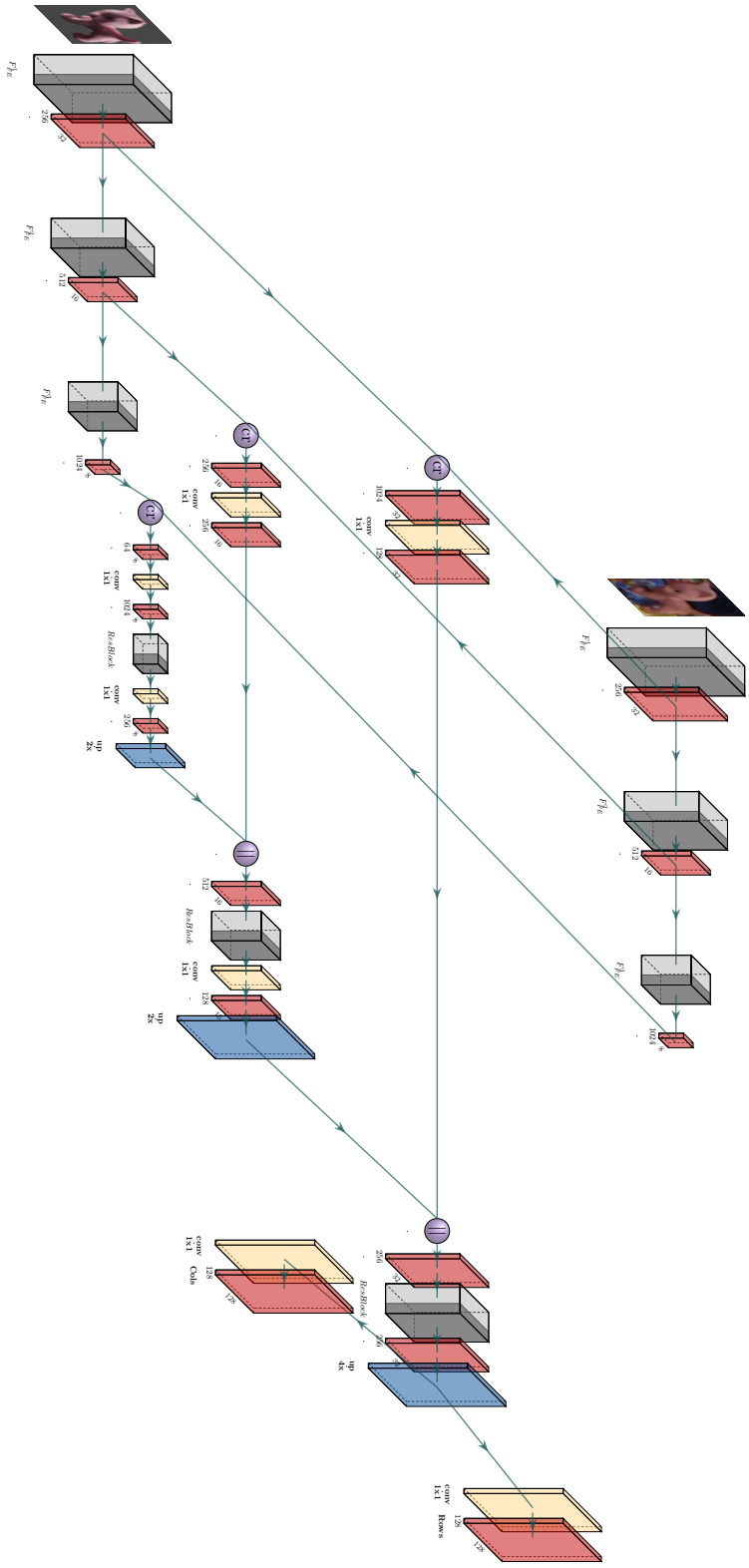


Figure 5. Architecture of the stage 3 network for dense correspondence matching. Gray color represents ResNet blocks, yellow color represents convolutional layers, red color represent feature maps and descriptors and green represents attention blocks. Better viewed digitally.

Table 3. One shot segmentation on all images from the Linemod [6] dataset.

Object	1	2	4	5	6	8	9	10	11	12	13	14	15
Bbox Precision	0.42	0.59	0.31	0.59	0.5	0.29	0.63	0.98	0.44	0.37	0.75	0.21	0.23
Bbox Recall	0.87	0.95	0.71	0.96	0.91	0.83	0.97	0.78	0.7	0.93	0.99	0.69	0.68
Pixel IoU	0.77	0.75	0.58	0.76	0.71	0.74	0.81	0.83	0.62	0.72	0.79	0.64	0.63

Table 4. One shot segmentation on BOP subset of images from the Occlusion [2] dataset.

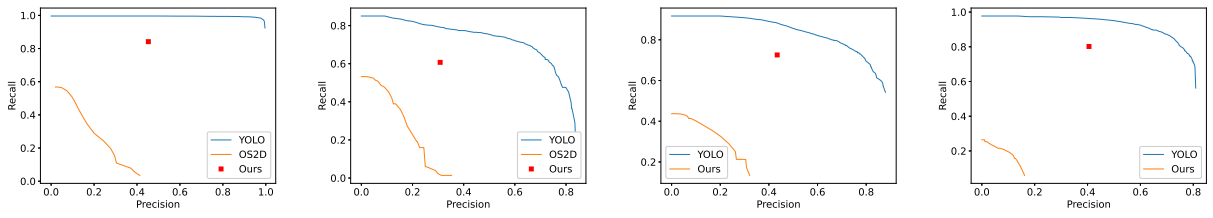
Object	1	5	6	8	9	10	11	12
Bbox Precision	0.27	0.26	0.43	0.14	0.55	0.36	0.15	0.28
Bbox Recall	0.55	0.56	0.68	0.5	0.86	0.68	0.25	0.76
Pixel IoU	0.51	0.58	0.61	0.68	0.75	0.58	0.38	0.63

Table 5. One shot segmentation on BOP subset of images from the Homebrewed [9] dataset.

Object	1	3	4	8	9	10	12	15	17	18	19	22	23	29	32	33
Bbox Precision	0.23	0.63	0.65	0.52	0.47	0.05	0.39	0.65	0.29	0.09	0.92	0.31	0.58	0.9	0.15	0.51
Bbox Recall	0.63	0.92	0.92	0.79	0.91	0.12	0.89	0.22	0.61	0.24	100	0.82	0.99	0.96	0.38	0.74
Pixel IoU	0.71	0.83	0.77	0.79	0.75	0.64	0.82	0.71	0.55	0.63	0.89	0.72	0.83	0.89	0.57	0.84

Table 6. One shot segmentation on BOP subset of images from the YCB-V [20] dataset.

Object	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Bbox Precision	0.44	0.21	0.51	0.59	0.51	0.55	0.31	0.36	0.27	0.72	0.24	0.18	0.38	0.62	0.09	0.31	0.13	0.81	0.18	0.17	0.9
Bbox Recall	0.98	0.53	0.98	0.91	1	1	0.92	1	0.65	1	0.69	0.63	1	1	0.34	0.76	0.2	1	0.63	0.57	1
Pixel IoU	0.61	0.55	0.75	0.77	0.71	0.85	0.38	0.75	0.49	0.88	0.58	0.53	0.78	0.66	0.58	0.67	0.8	0.77	0.64	0.59	0.85



(a) Linemod [4] (b) Occlusion [2] (c) Homebrewed [9] (d) YCB-V [20]

Figure 6. Precision-Recall curves for four datasets used in the evaluation on a subset of image. We compare out method to OS2D [14] and YOLOv3 [16].

Figure 7. Segmentation quality on the Linemod dataset. Left column visualizes ground truth outlines of the visible object parts. Right column visualizes predicted outlines of the visible object parts.

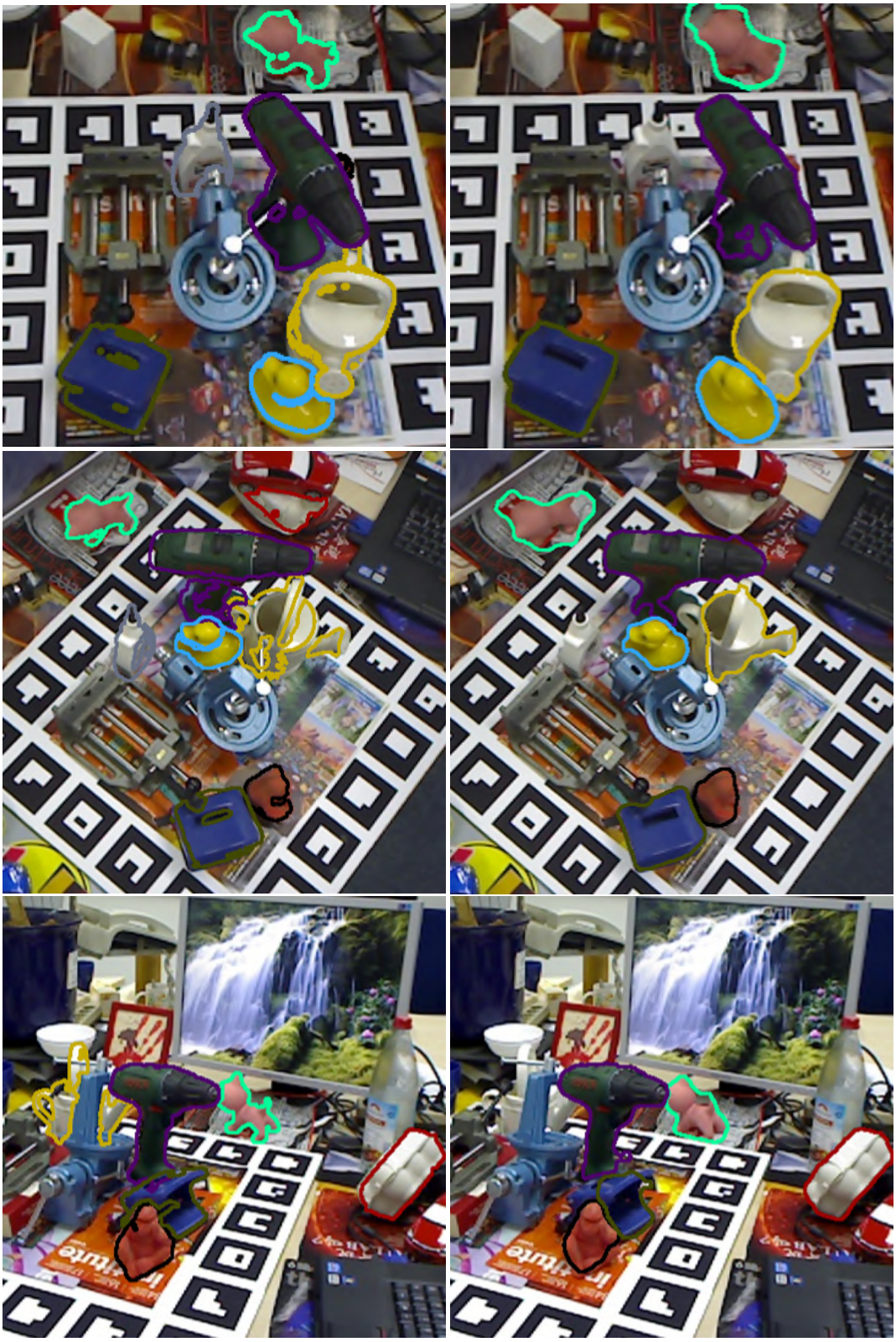


Figure 8. Segmentation quality on the Occlusion dataset. Left column visualizes ground truth outlines of the visible object parts. Right column visualizes predicted outlines of the visible object parts.

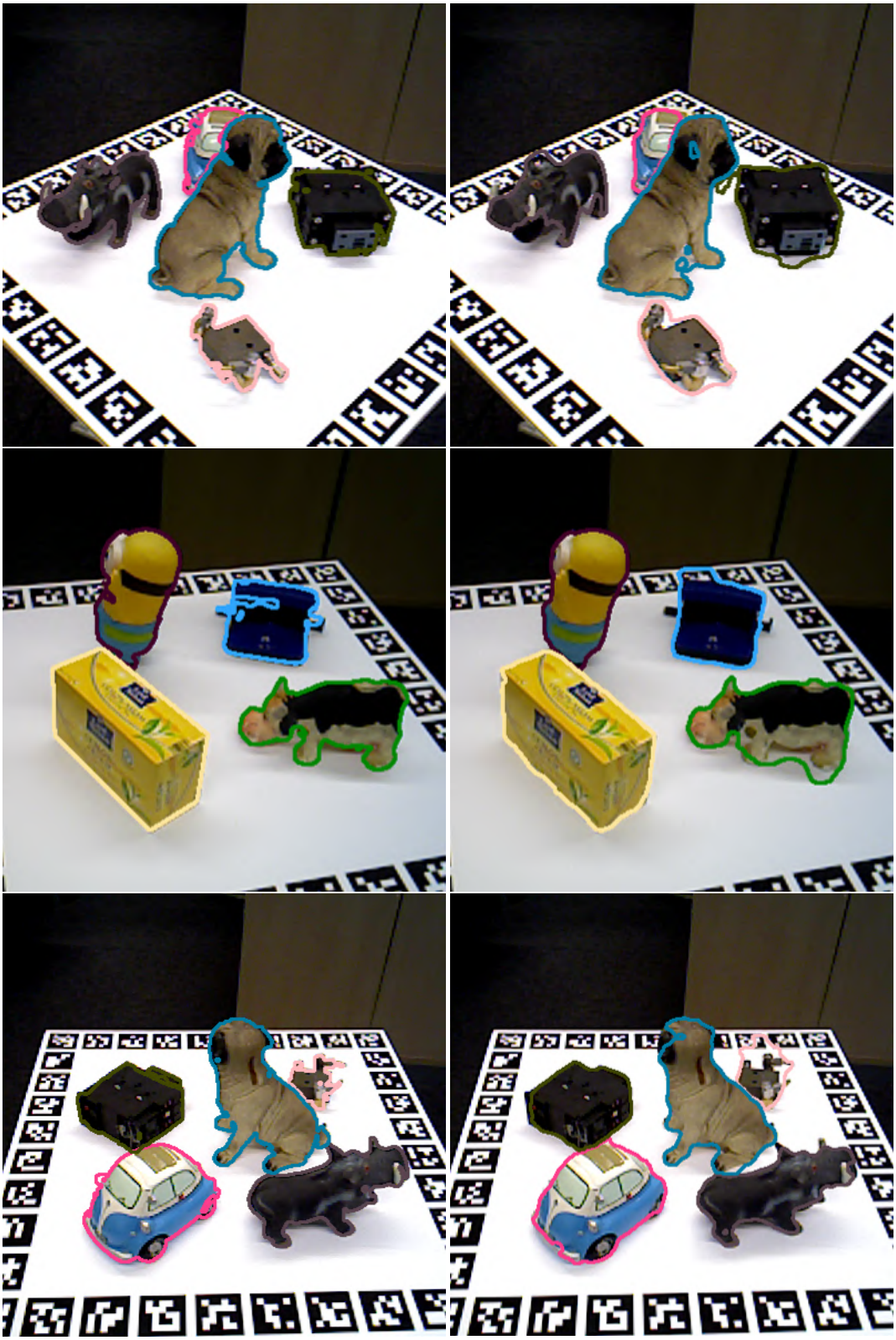


Figure 9. Segmentation quality on the Homebrewed dataset. Left column visualizes ground truth outlines of the visible object parts. Right column visualizes predicted outlines of the visible object parts.

Figure 10. Segmentation quality on the YCB-V dataset. Left column visualizes ground truth outlines of the visible object parts. Right column visualizes predicted outlines of the visible object parts.

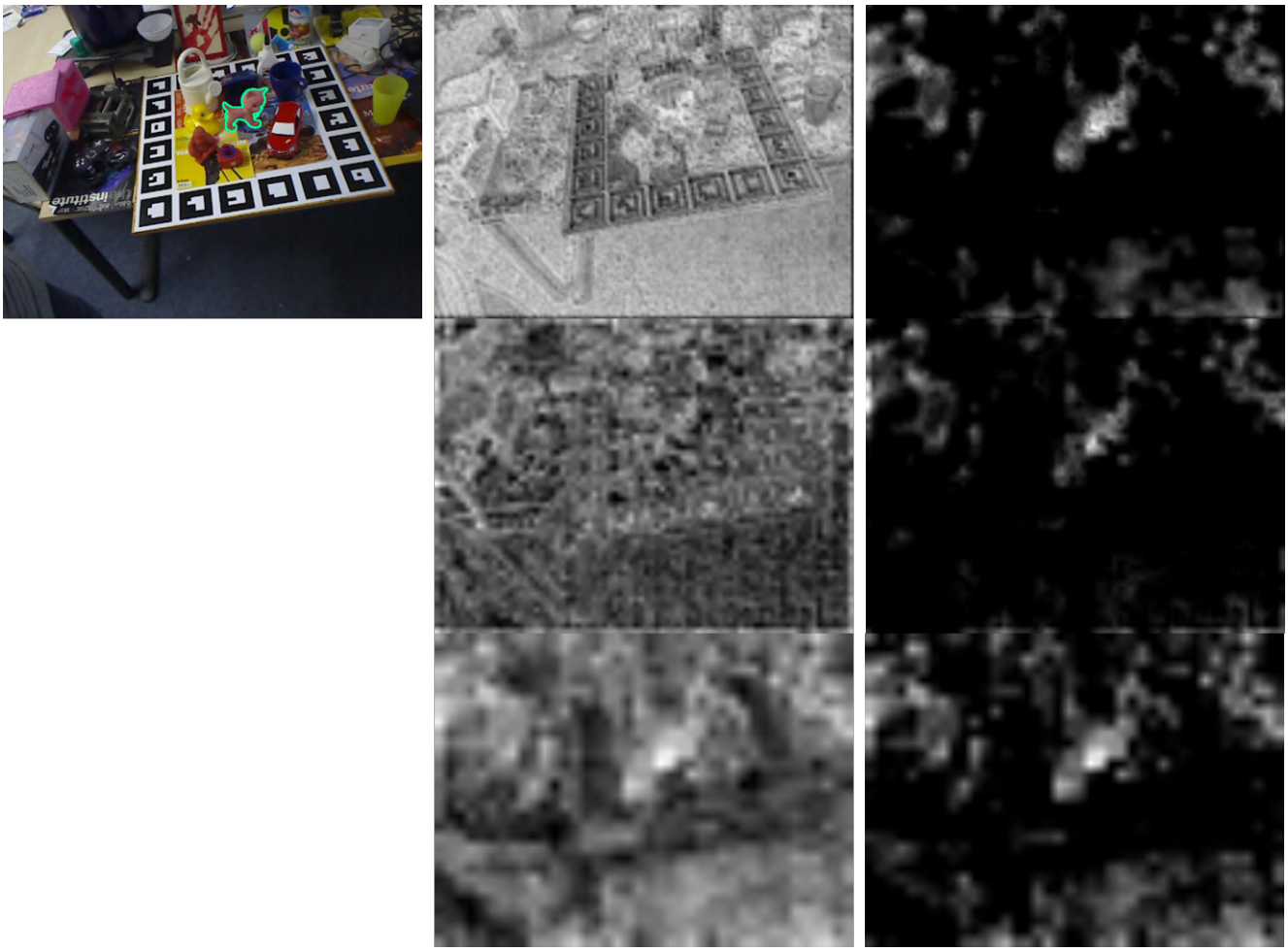


Figure 11. Visualization of attention values on the Linemod dataset. The left column provides the original RGB image with the highlighted target object. The central column shows the raw attentions and the right column visualizes the thresholded and conditioned, as described in the paper. Rows correspond to different depth of feature extractor.

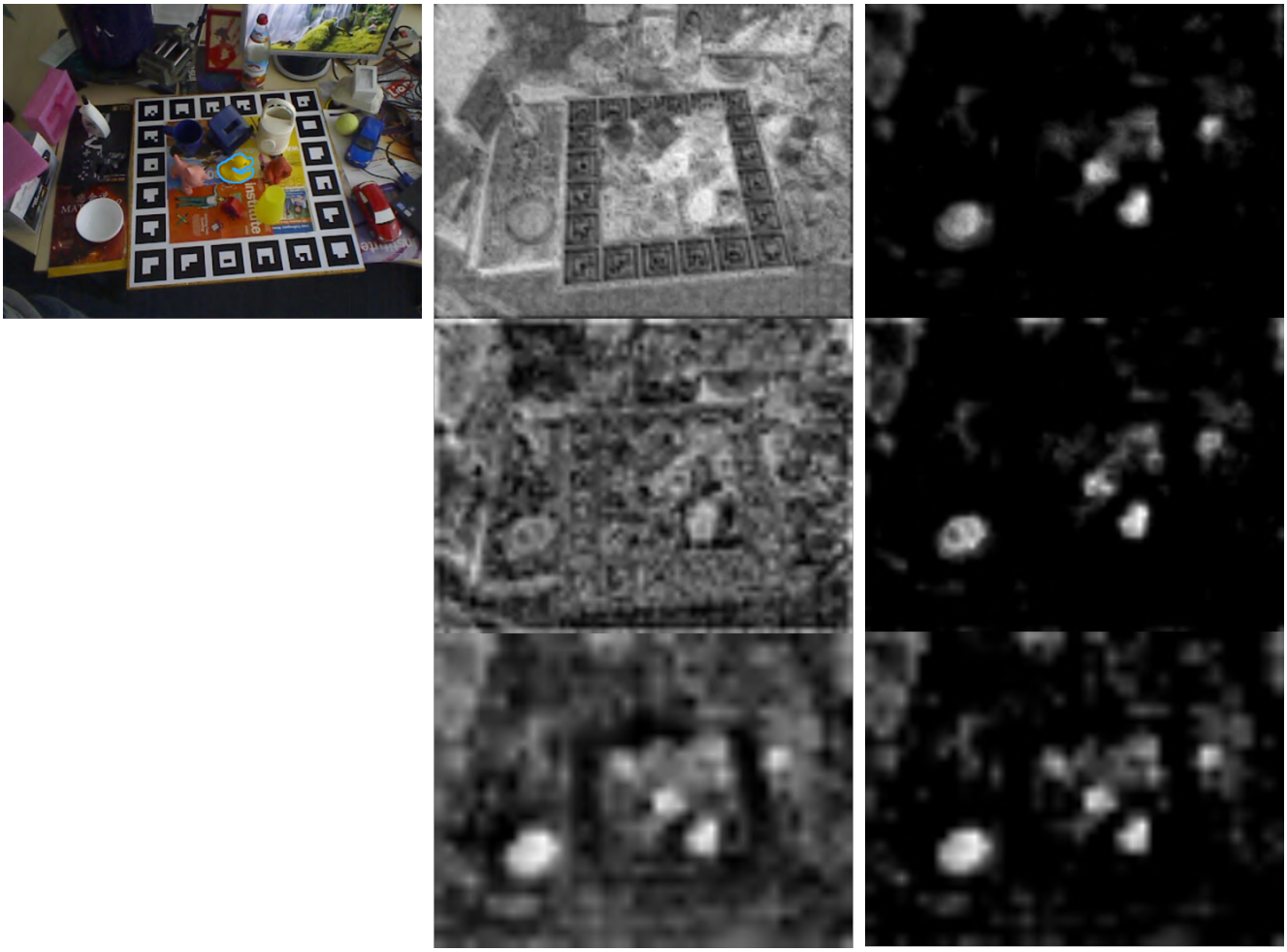


Figure 12. Visualization of attention values on the Linemod dataset. The left column provides the original RGB image with the highlighted target object. The central column shows the raw attentions and the right column visualizes the thresholded and conditioned, as described in the paper. Rows correspond to different depth of feature extractor.

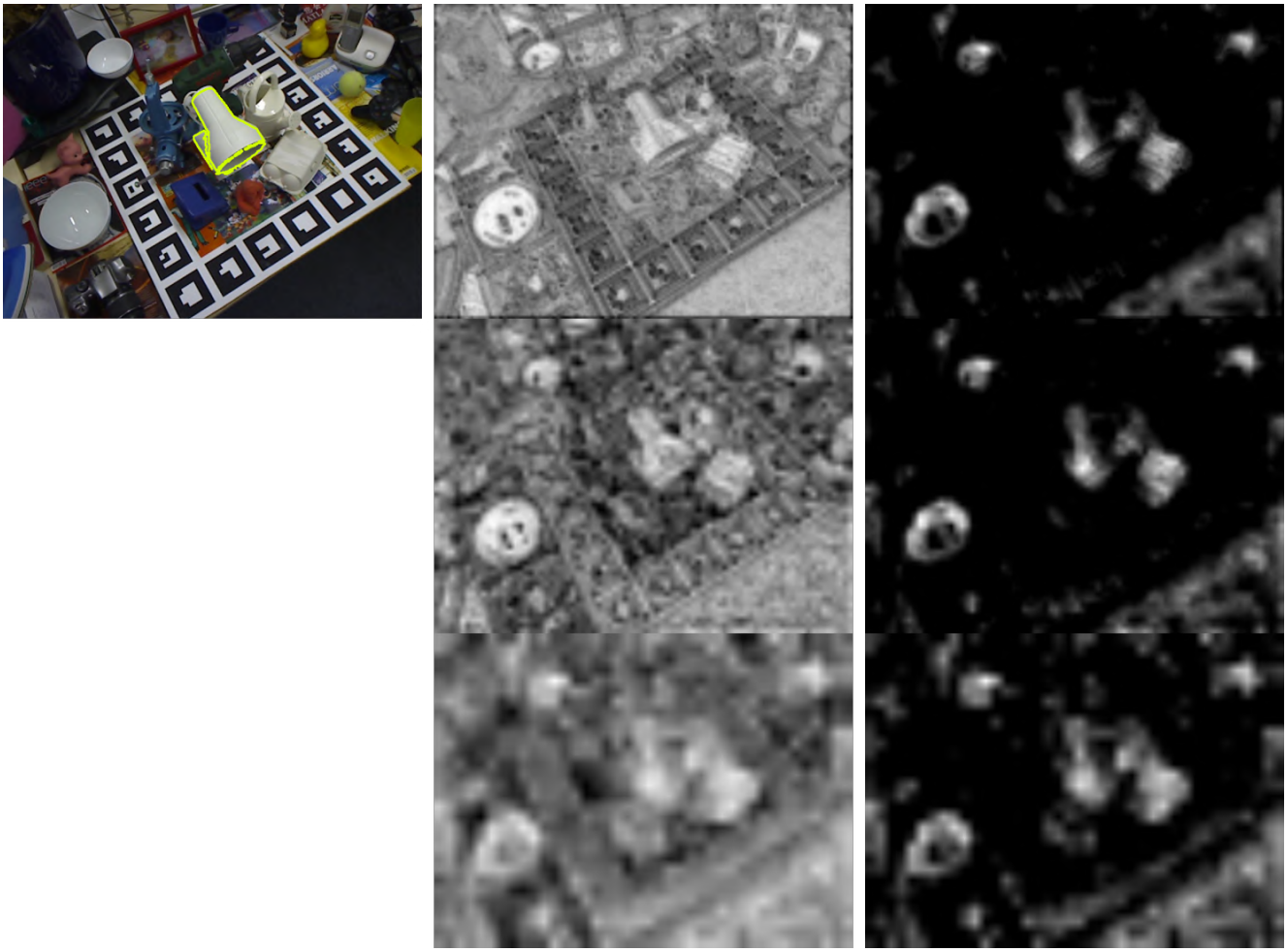


Figure 13. Visualization of attention values on the Linemod dataset. The left column provides the original RGB image with the highlighted target object. The central column shows the raw attentions and the right column visualizes the thresholded and conditioned, as described in the paper. Rows correspond to different depth of feature extractor.

Figure 14. Correspondence quality on the Linemod dataset. The left row shows input RGB images with highlighted target objects. Next two columns show color-coded ground truth and predicted NOCS maps.

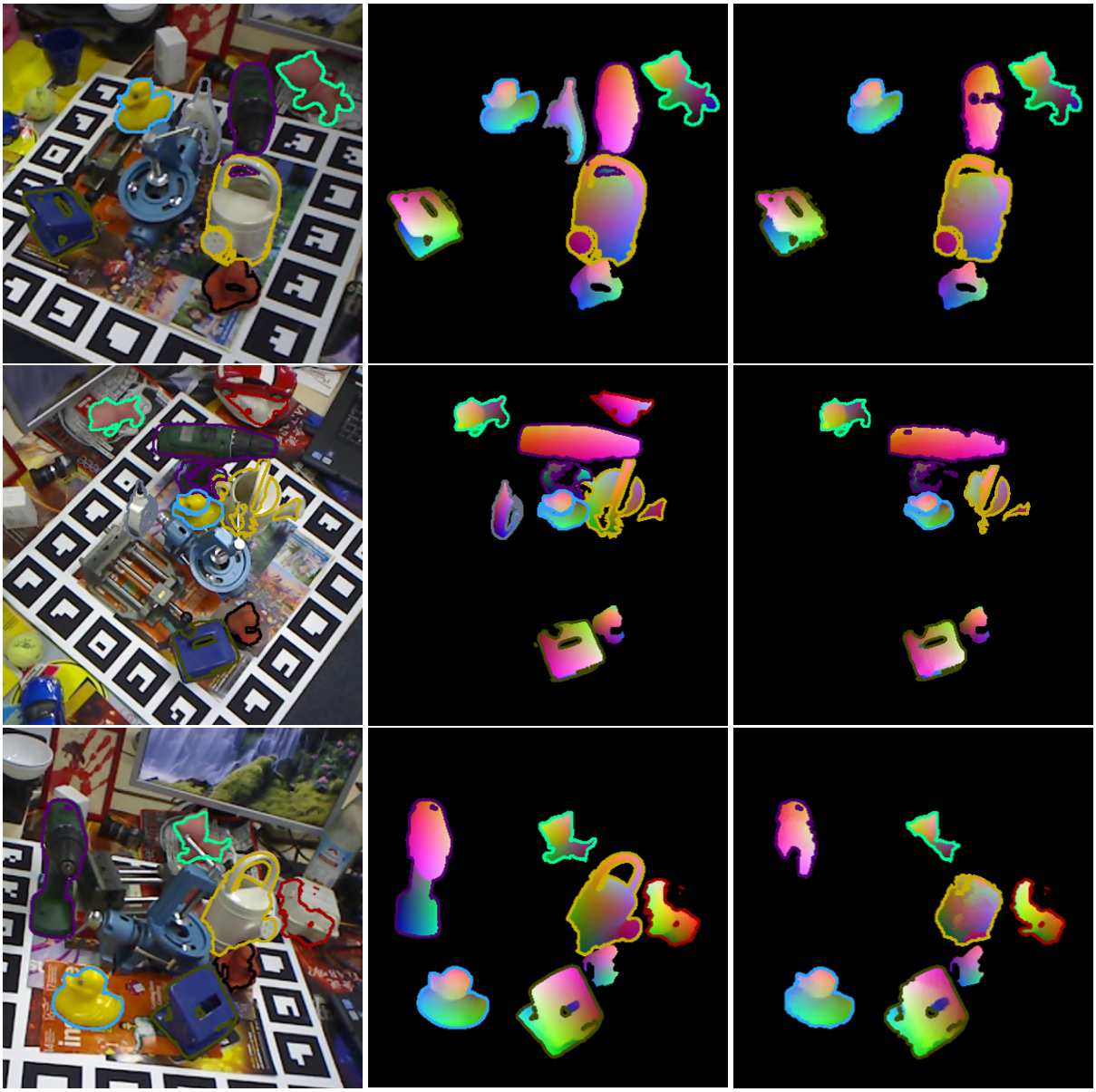


Figure 15. Correspondence quality on the Occlusion dataset. The left row shows input RGB images with highlighted target objects. Next two columns show color-coded ground truth and predicted NOCS maps.

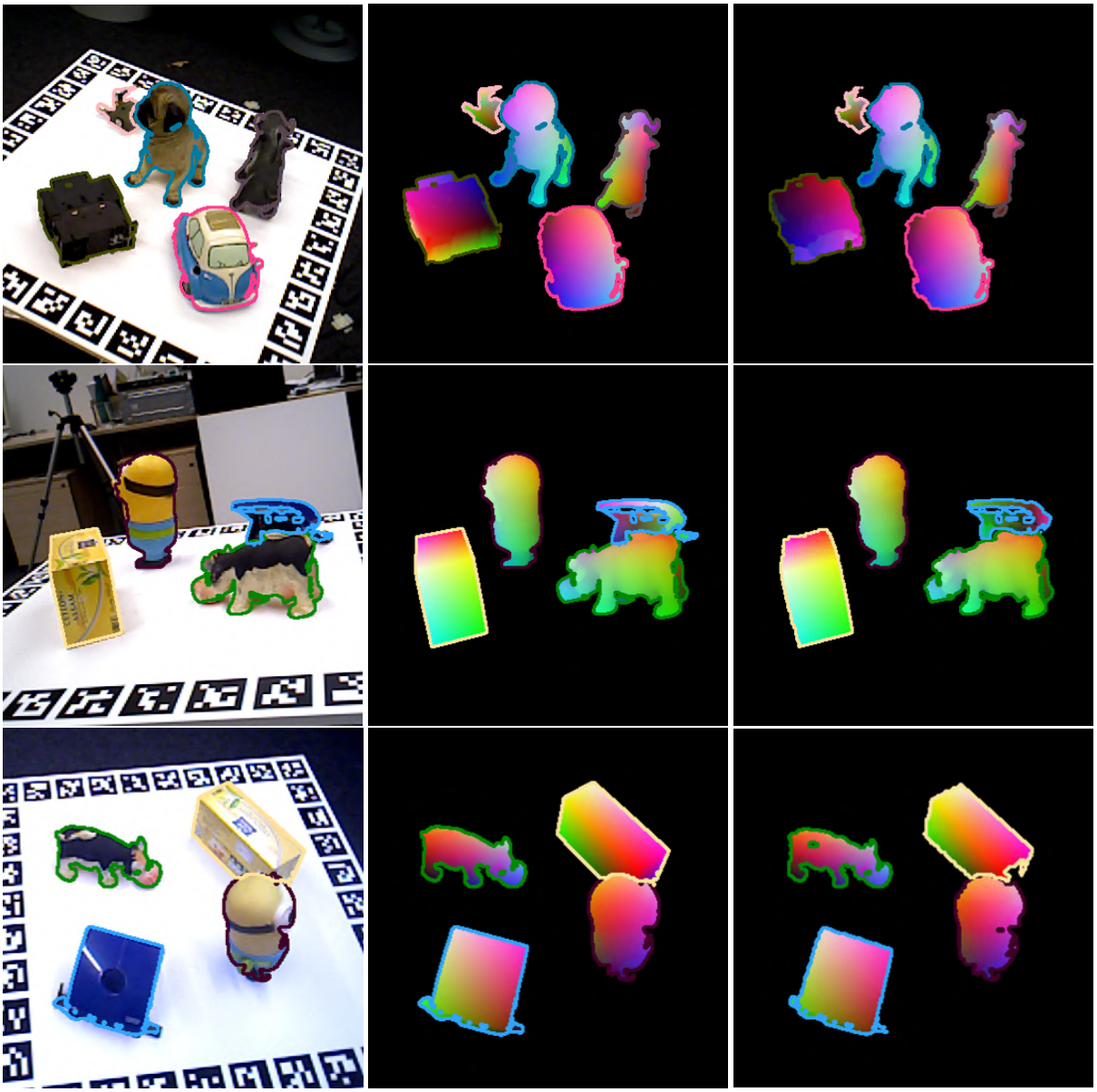


Figure 16. Correspondence quality on the Homebrewed dataset. The left row shows input RGB images with highlighted target objects. Next two columns show color-coded ground truth and predicted NOCS maps.

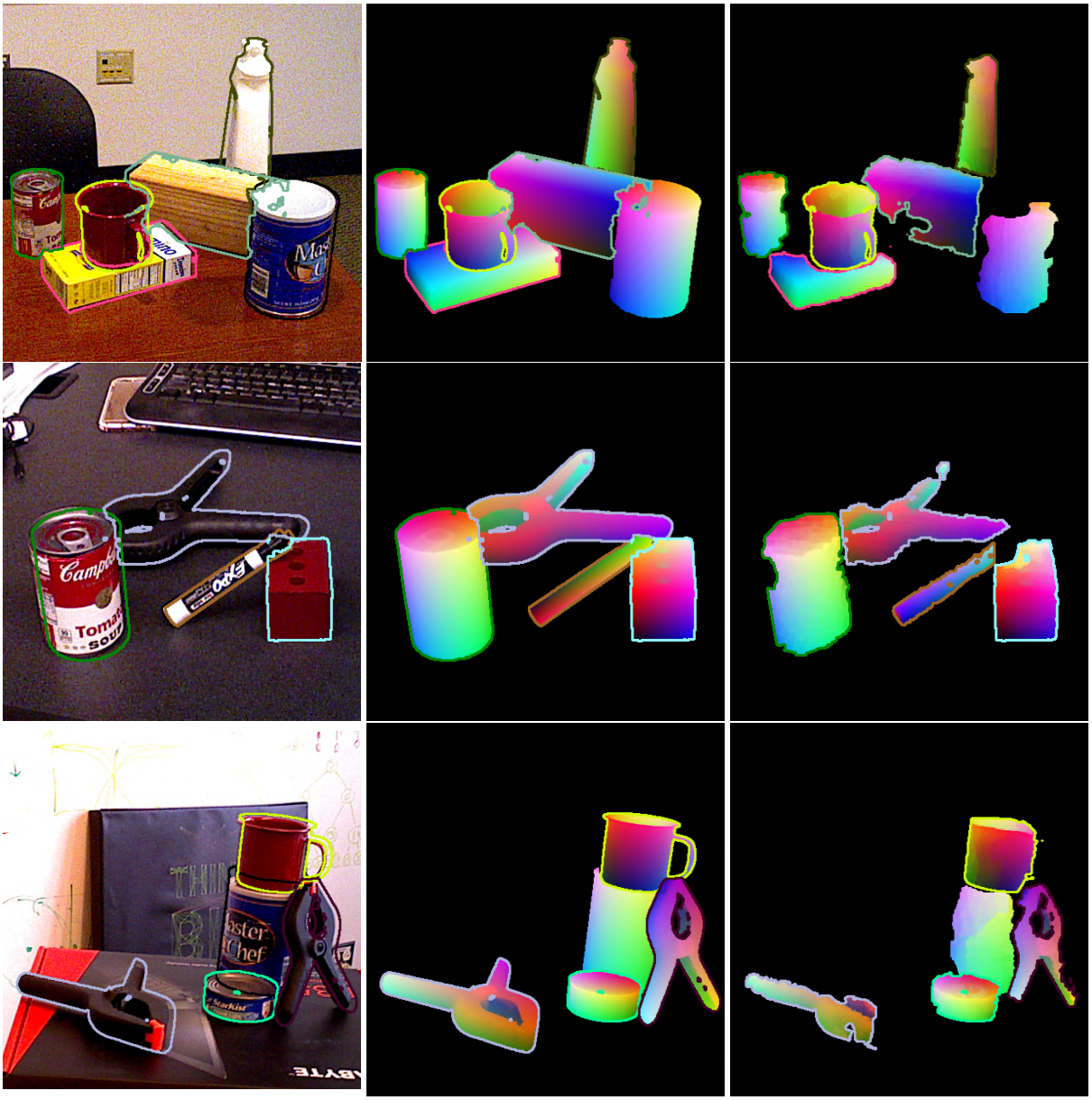


Figure 17. Correspondence quality on the YCB-V dataset. The left row shows input RGB images with highlighted target objects. Next two columns show color-coded ground truth and predicted NOCS maps.

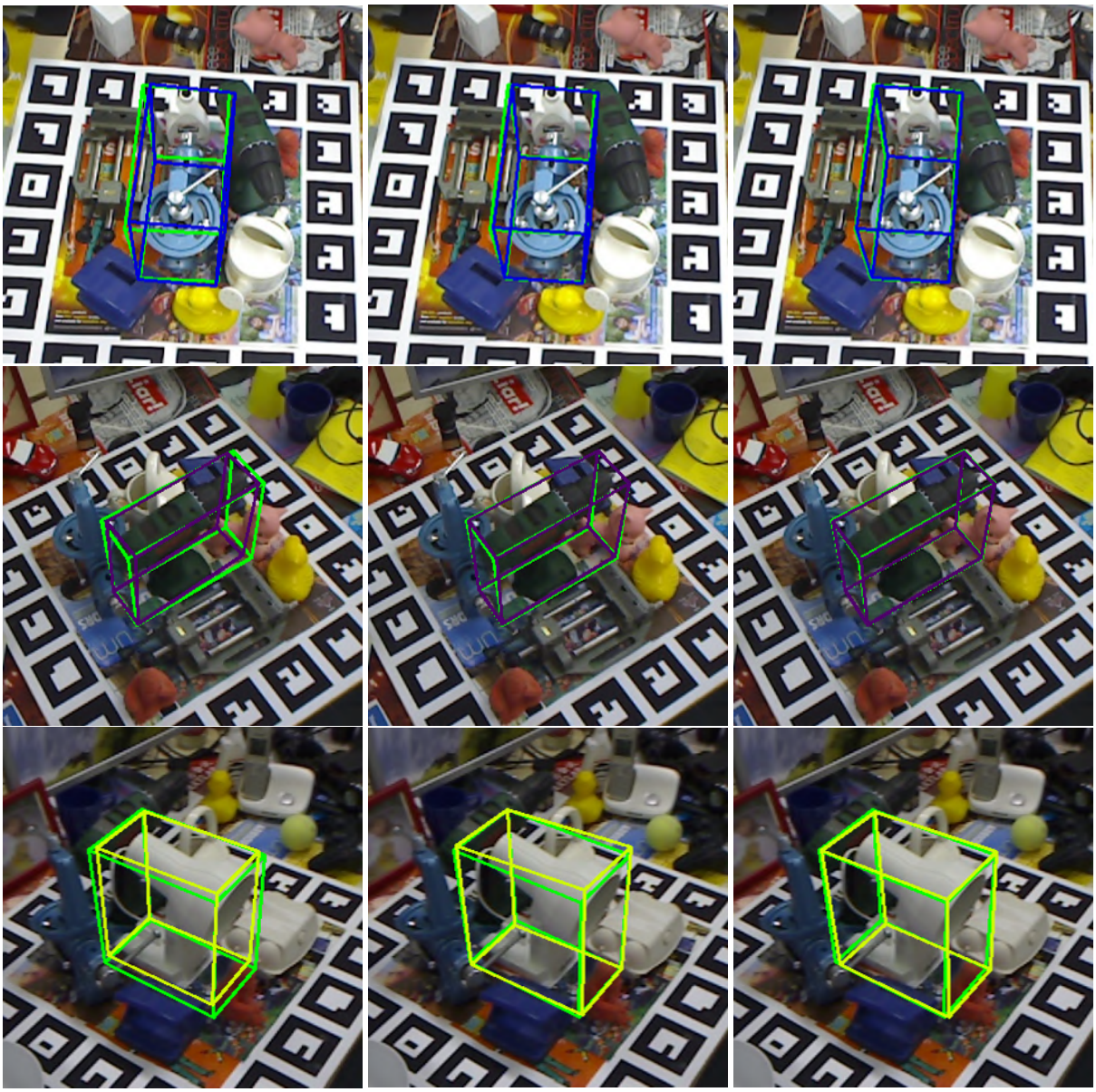


Figure 18. Pose quality on the Linemod dataset. Green 3D bounding boxes visualize the ground truth poses of the object. Color-coded bounding boxes visualize the predictions. The left column contain the poses obtained with the PnP algorithm, the central column - with Kabsch, and the poses in the right column are refined with ICP.

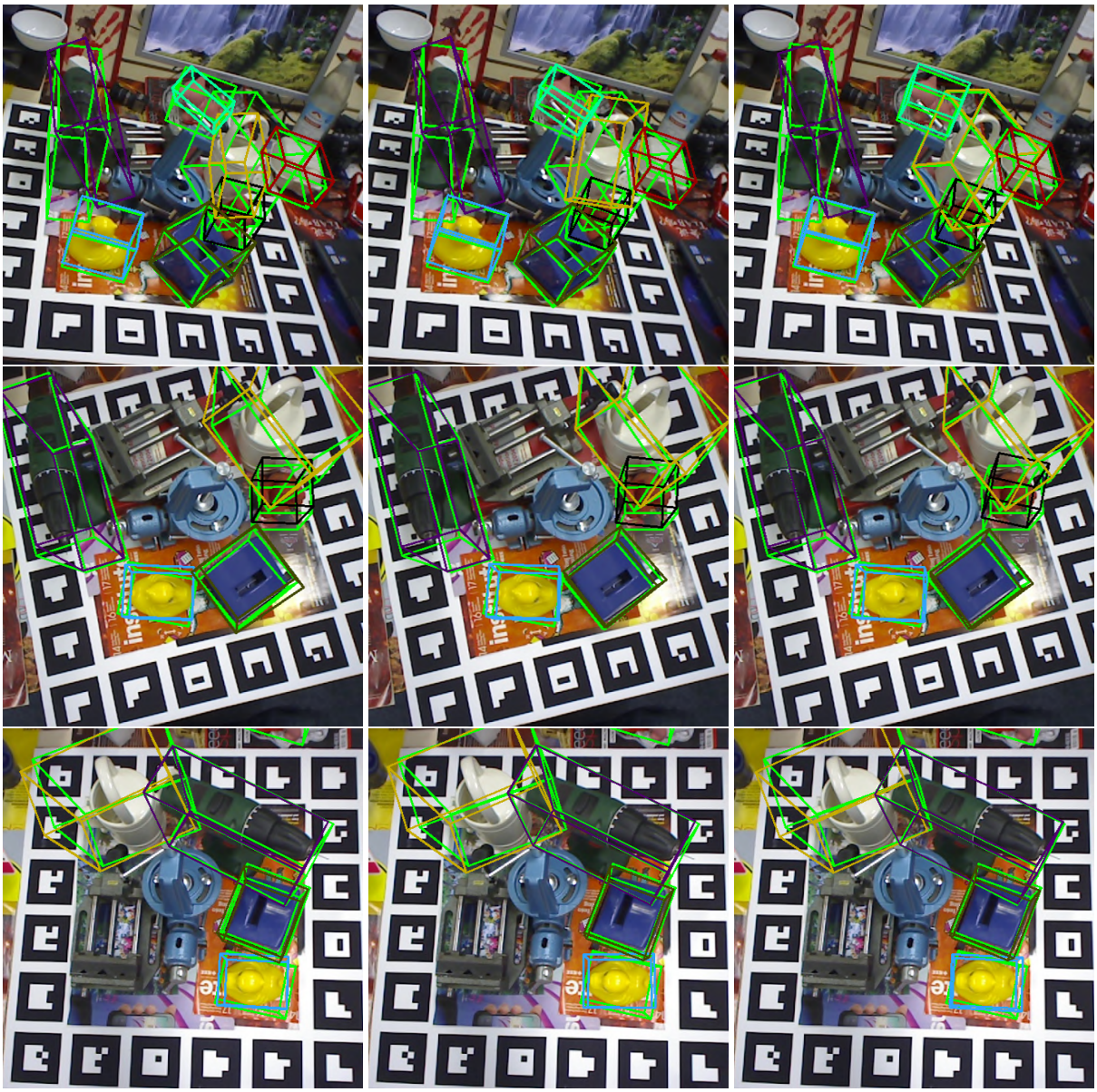


Figure 19. Pose quality on the Occlusion dataset. Color-coded bounding boxes visualize the predictions. The left column contain the poses obtained with the PnP algorithm, the central column - with Kabsch, and the poses in the right column are refined with ICP.

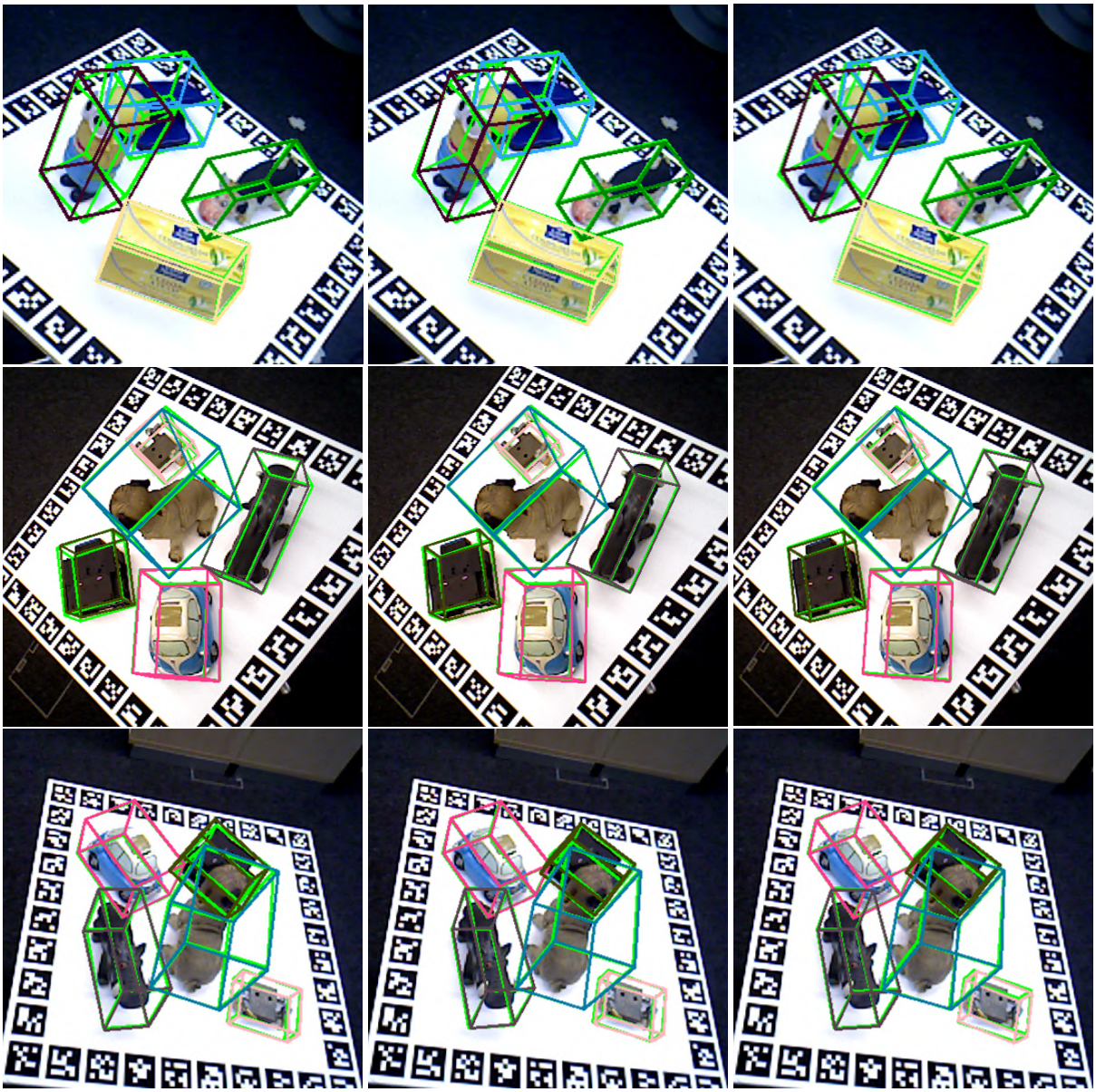


Figure 20. Pose quality on the Homebrewed dataset. Color-coded bounding boxes visualize the predictions. The left column contain the poses obtained with the PnP algorithm, the central column - with Kabsch, and the poses in the right column are refined with ICP.

Figure 21. Pose quality on the YCB-V dataset. Color-coded bounding boxes visualize the predictions. The left column contain the poses obtained with the PnP algorithm, the central column - with Kabsch, and the poses in the right column are refined with ICP.