Supplemental Material: "Real-Time Seamless Single Shot 6D Object Pose Prediction"

Bugra Tekin
EPFL
bugra.tekin@epfl.ch

Sudipta N. Sinha
Microsoft Research
sudipta.sinha@microsoft.com

Pascal Fua
EPFL
pascal.fua@epfl.ch

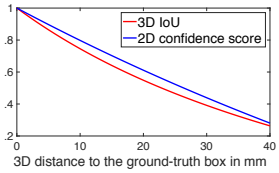
In the supplemental material, we provide details on how the training images were prepared and on the proposed confidence function and the weighted prediction step. We also present qualitative results on OCCLUSION [1] and LINEMOD [5].

Training Images. As discussed in the main paper, we segment the foreground object in the images of the training set, using the segmentation masks provided and paste the segmented image over a random image as in [2, 6, 7]. Examples of such images, which are given as input to the network at training time are shown in Figure 1. This operation of removing the actual background prevents the network from overfitting to the background, which is similar for training and test images of LINEMOD. When we train a model without eliminating the background, in practice, we observe about 1% improvement in the 2D projection score.

Figure 1. Using segmentation masks given in LINEMOD, we extract the foreground objects in our training images and composite them over random images from PASCAL VOC [4]. We also augment the training set by combining images of multiple objects taken from different training images.

Confidence function. We analyze in Figure 2 our confidence function in comparison to 3D cube IoU in terms of its value and runtime. We show that our confidence function closely approximates the actual 3D cube IoU while being much faster to compute.

Confidence-weighted prediction. In the final step of our method, we compute a weighted sum of multiple sets of predictions for the corners and the centroid, using associated confidence values as weights. On LINEMOD, this gave a 1–2% improvement in accuracy with the 2D projection metric. The first step involves scanning the full 17×17 grid to find the cell with the highest confidence for each potential object. We then consider a 3×3 neighborhood around it on the grid and prune the cells with confidence values lower than the detection threshold of 0.5. On the remaining cells, we compute a confidence-weighted average of the associated predicted 18-dimensional vectors, where the eight corner points and the centroid have been stacked to form the vector. The averaged coordinates are then used in the PnP method. This *s*ub-pixel refinement on the grid usually improves the pose of somewhat large objects that occupy several adjoining cells in the grid. Figure 3 shows an example where the *ape* object lies between two adjoining cells and the confidence weighting improves the pose accuracy.



Method	Runtime per object (ms)
3D IoU	5.37
2D Conf. Score	0.18

Figure 2. Comparison of the 3D IoU and our 2D confidence score in value (Left) and runtime (Right). The model for the *Cam* object is shifted in x-dimension synthetically to produce a distorted prediction and projected on the image plane with randomly chosen 20 transformation matrices from LINEMOD. Scores are computed between the ground-truth references and distorted predictions. Results are averaged over all the trials. The runtime for 3D IoU is computed using the optimized PyGMO library that relies on [3].

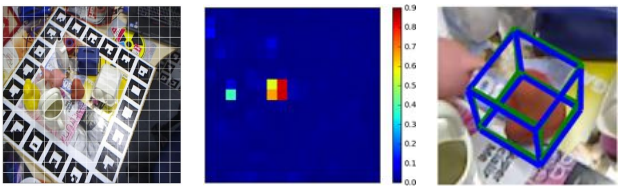


Figure 3. (Left) The 17×17 grid on a 544×544 image. (Middle) Confidence values for predictions of the *ape* object on the grid. (Right) Cropped view of our pose estimate (shown in blue) and the ground truth (shown in green). Here, three cells next to the best cell have good predictions and their combination gives a more accurate pose than the best prediction alone (best viewed in color).

Qualitative Results. We show qualitative results from the OCCLUSION [1] and LINEMOD [5] datasets in Figures 4 to 9. These examples show that our method is robust to severe occlusions, rotational ambiguities in appearance, reflections, viewpoint change and scene clutter.

Figure 4. Results on the OCCLUSION dataset. Our method is quite robust against severe occlusions in the presence of scene clutter and rotational pose ambiguity for symmetric objects. (left) Input images, (middle) 6D pose predictions of multiple objects, (right) A magnified view of the individual 6D pose estimates of six different objects is shown for clarity. In each case, the 3D bounding box is rendered on the input image. The following color coding is used – APE (gold), BENCHVISE (green), CAN (red), CAT (purple), DRILLER (cyan), DUCK (black), GLUE (orange), HOLEPUNCHER (blue). In addition to the objects from the OCCLUSION dataset, we also visualize the pose predictions of the *Benchvise* object from the LINEMOD dataset. As in [7], we do not evaluate on the *Eggbox* object, as more than 70% of close poses are not seen in the training sequence. This image is best viewed on a computer screen.

References

[1] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *ECCV*, 2014.

[2] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, and C. Rother. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In *CVPR*, 2016.

[3] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry: Theory and Applications*, 43:601–610, 2010.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

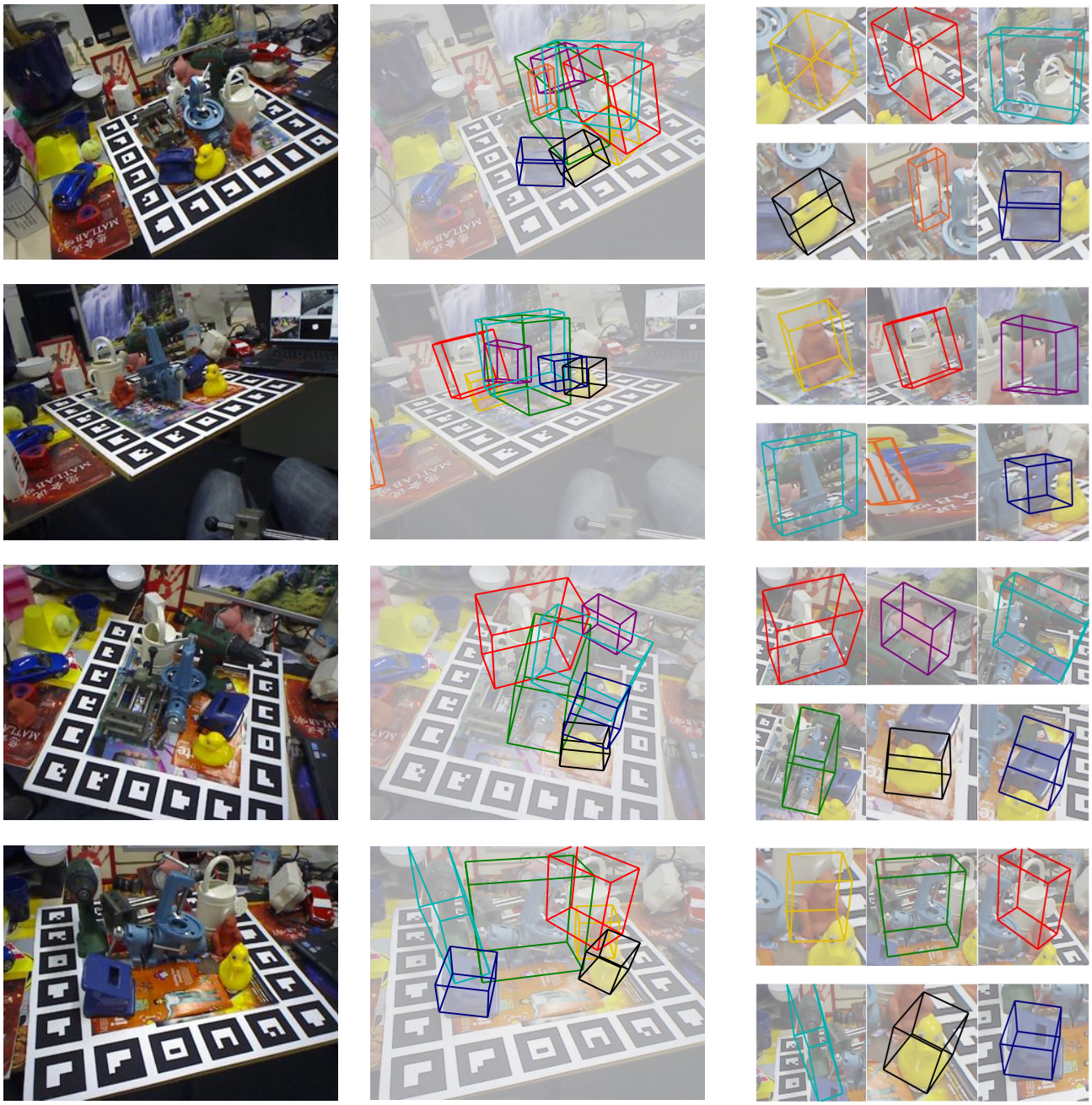


Figure 5. Results on the OCCLUSION dataset. Our method is quite robust against severe occlusions in the presence of scene clutter and rotational pose ambiguity for symmetric objects. (left) Input images, (middle) 6D pose predictions of multiple objects, (right) A magnified view of the individual 6D pose estimates of six different objects is shown for clarity. In each case, the 3D bounding box is rendered on the input image. The following color coding is used – APE (gold), BENCHVISE (green), CAN (red), CAT (purple), DRILLER (cyan), DUCK (black), GLUE (orange), HOLEPUNCHER (blue). In addition to the objects from the OCCLUSION dataset, we also visualize the pose predictions of the *Benchvise* object from the LINEMOD dataset. As in [7], we do not evaluate on the *Eggbox* object, as more than 70% of close poses are not seen in the training sequence. This image is best viewed on a computer screen.

[5] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model Based Training, Detection and Pose Estimation of Texture-less 3D Objects in Heavily Cluttered Scenes. In *ACCV*, 2012.

[6] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *ICCV*, 2017.

[7] M. Rad and V. Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. In *ICCV*, 2017.

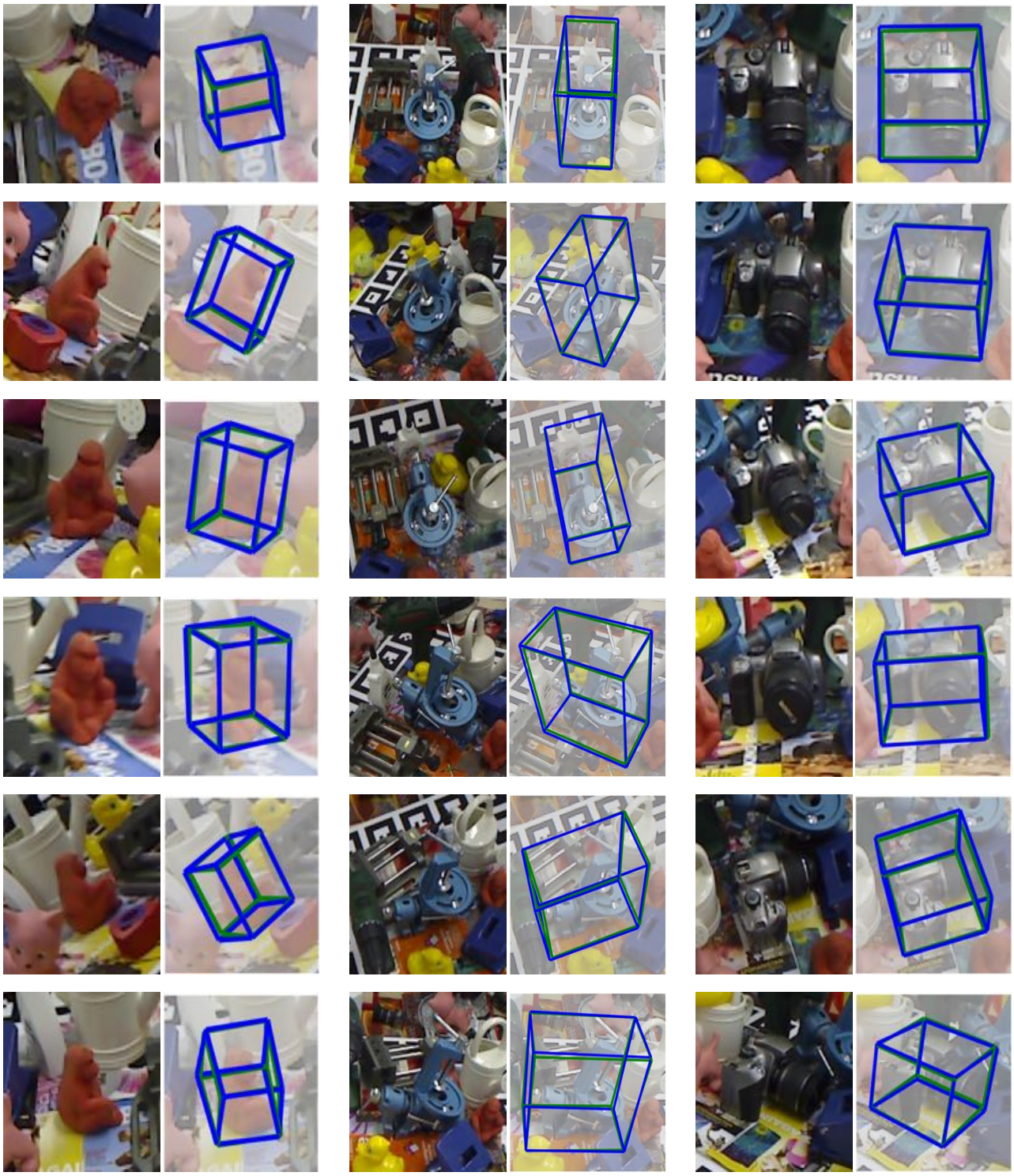


Figure 6. Example results on the LINEMOD dataset: (left) APE, (middle) BENCHVISE, (right) CAM. The projected 3D bounding boxes are rendered over the image and they have been cropped and resized for ease of visualization. The blue cuboid is rendered using our pose estimate whereas the green cuboid is rendered using the ground truth object pose. Note that the input image dimension is 640×480 pixels and the objects are often quite small. Noticeable scene clutter and occlusion makes these examples challenging.

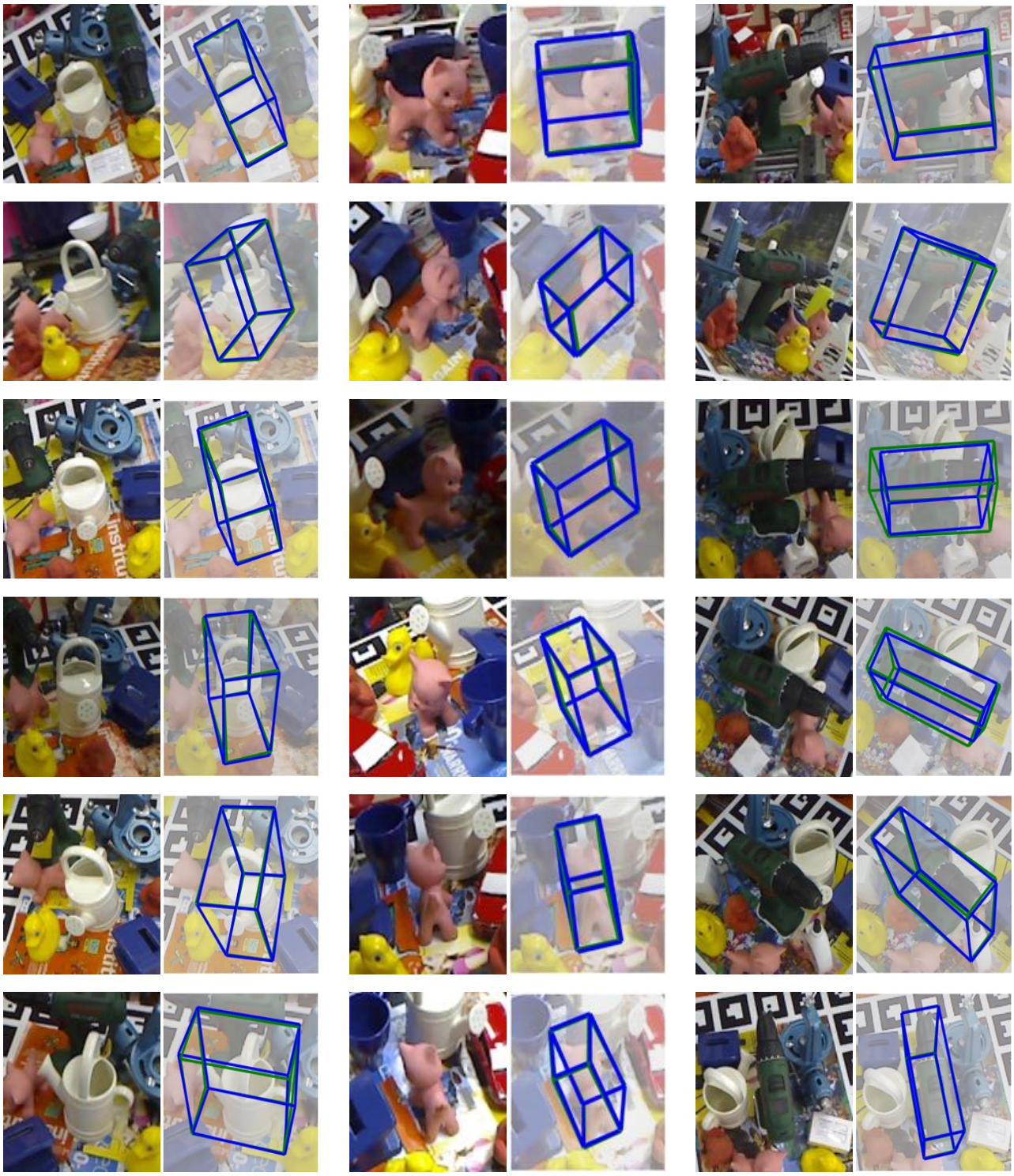


Figure 7. Example results on the LINEMOD dataset: (left) CAN, (middle) CAT, (right) DRILLER. The projected 3D bounding boxes are rendered over the image and they have been cropped and resized for ease of visualization. The blue cuboid is rendered using our pose estimate whereas the green cuboid is rendered using the ground truth object pose. Note that the input image dimension is 640×480 pixels and the objects are often quite small. Noticeable scene clutter and occlusion makes these examples challenging.

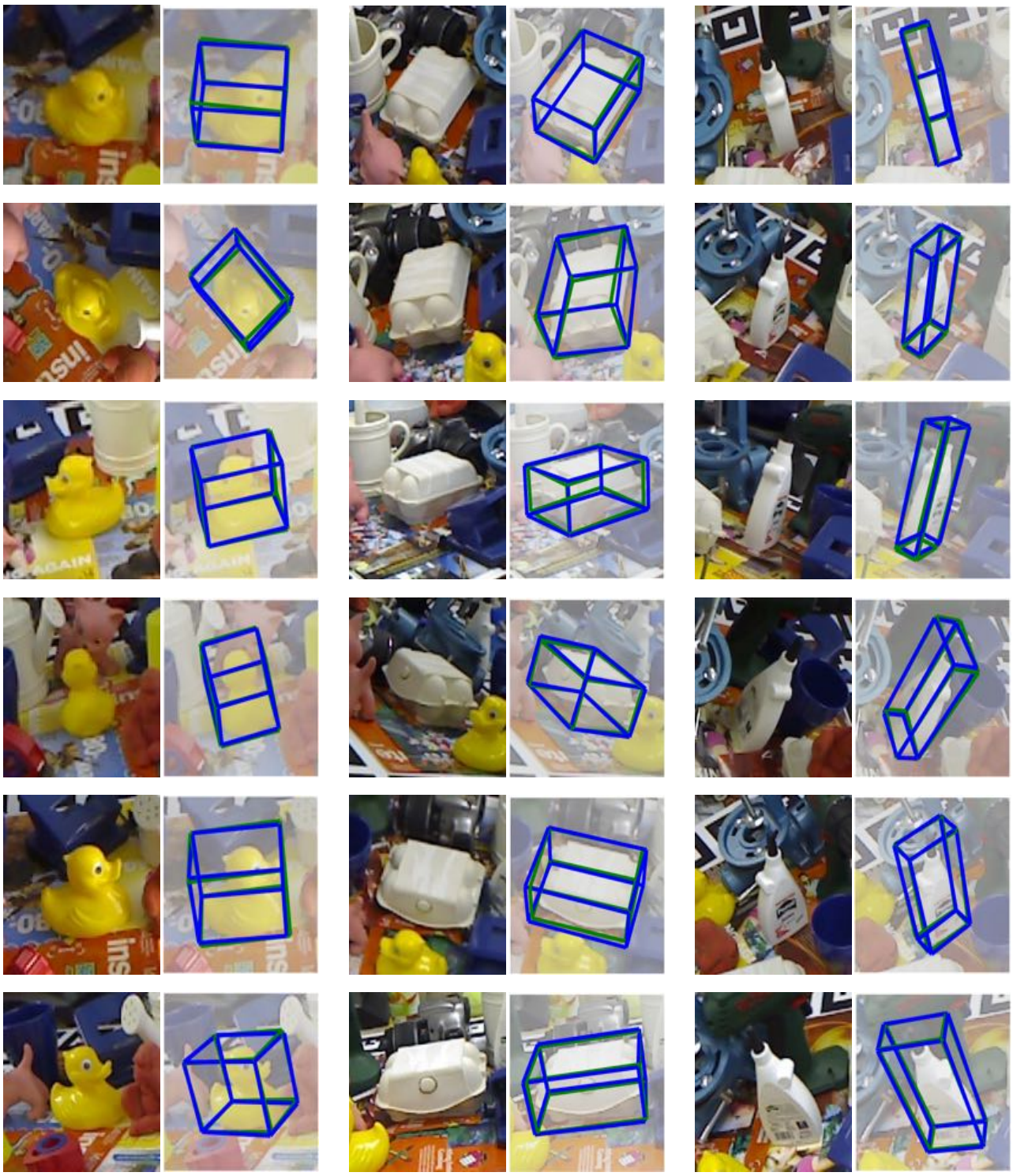


Figure 8. Example results on the LINEMOD dataset: (left) DUCK, (middle) EGGBOX, (right) GLUE. The projected 3D bounding boxes are rendered over the image and they have been cropped and resized for ease of visualization. The blue cuboid is rendered using our pose estimate whereas the green cuboid is rendered using the ground truth object pose. Note that the input image dimension is 640×480 pixels and the objects are often quite small. Noticeable scene clutter and occlusion makes these examples challenging.

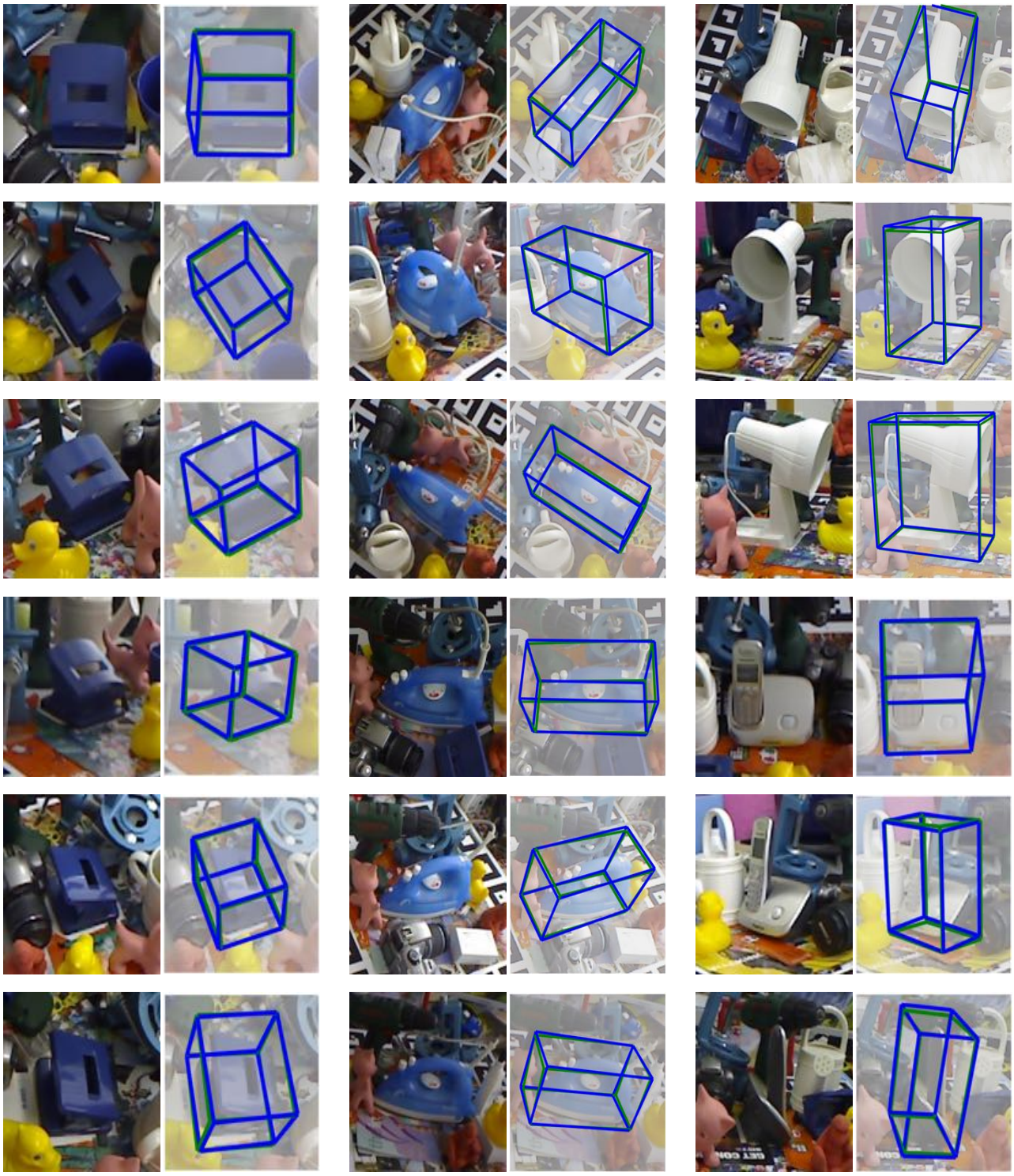


Figure 9. Example results on the LINEMOD dataset: (left) HOLEPUNCHER, (middle) IRON, (right) LAMP and PHONE. The projected 3D bounding boxes are rendered over the image and they have been cropped and resized for ease of visualization. The blue cuboid is rendered using our pose estimate whereas the green cuboid is rendered using the ground truth object pose. Note that the input image dimension is 640×480 pixels and the objects are often quite small. Noticeable scene clutter and occlusion makes these examples challenging.