6DoF Pose Estimation for Industrial Manipulation based on Synthetic Data

Manuel Brucker[1], Maximilian Durner[1], Zoltán-Csaba Márton[1], Ferenc Bálint-Benczédi[2], Martin Sundermeyer[1], and Rudolph Triebel[1]

Abstract We present a perception system for mobile manipulation tasks. The primary design goal of the proposed system is to minimize human interaction during system setup which is achieved by several means, such as automatic training data generation, the use of simulated training data, and 3D model based geometric matching. We employ a state-of-the art deep-learning based bounding box detector for rough localization of objects and a Point Pair Feature based matching algorithm for 6DoF pose estimation. The proposed approach shows promising results on our recently published dataset for industrial object detection and pose estimation. Furthermore, the system's performance during four days of live operation at the Automatica 2018 trade fair is analyzed and failure cases are presented and discussed.

1 Introduction and State of the Art

In this study, we evaluate a perception system for manipulation tasks that was designed based on our experiences developing autonomous robots and as part of the RobDREAM project (`http://robdream.eu/`). The perception system is integrated into our Autonomous Industrial Mobile Manipulator (AIMM) [7, 8], which is used to deliver parts to different workcells (see Fig. 1) in a shop floor logistics and kitting scenario.

Recently, deep learning (DL) has dominated vision research, and considerable work has been conducted in object detection, i.e., the simultaneous inference of bounding boxes and semantic labels of objects [25, 19, 6, 23, 18, 24]. However, for industrial use cases (and other practical tasks), the lack of labeled training data must be addressed. Here, we present our experiences with robot-aided dataset generation and using simulations of the scenario to obtain synthetic views.

[1]German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Münchner Str. 20, Oberpfaffenhofen, 82234 Weßling, Germany; e-mail: `First.Last@dlr.de` · [2]Institute for Artificial Intelligence, University of Bremen, Germany; e-mail: `balintbe@cs.uni-bremen.de`

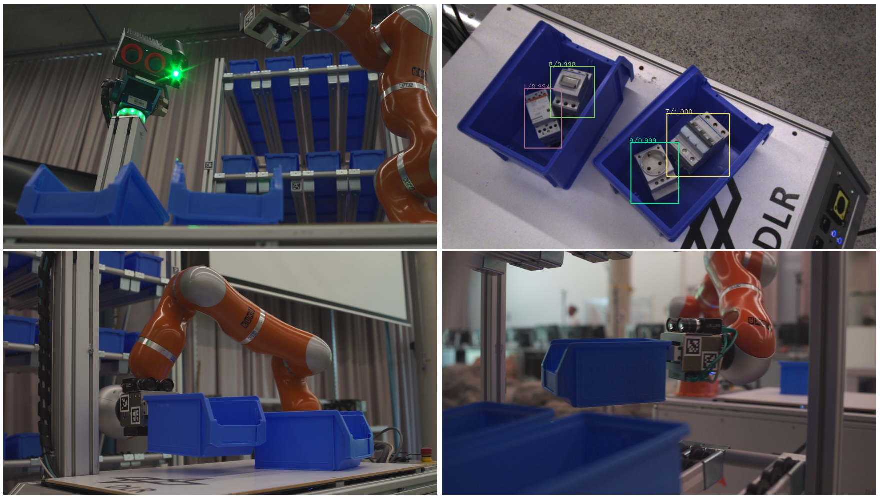


Fig. 1: AIMM platform use case: Small Load Carriers (SLCs) are detected on a workbench and moved to the robot's transport area. The content of the SLCs is detected and placed in the corresponding flow rack.

Given the previously reported mixed generalization performance of photo-realistically rendered object views (with background) for object detection and viewpoint estimation [21, 27, 20], we investigated the effect of using different combinations of real and UnrealCV [33] data to analyze the benefits of considering synthetic images when training a deep-learned object detector.

Although DL approaches for pose estimation exist [4, 22, 16, 29, 28], if reliable depth data is available and sub-framerate speed is sufficient, methods based on Point Pair Features (PPF) [31, 13], have demonstrated more robust pose estimation performance on multiple datasets. Note that a PPF-based method won the recent SIXD Challenge (`http://cmp.felk.cvut.cz/sixd/challenge_2017/`). Typically, runtimes in the range of seconds are required. Here, we use such an approach [17], which is optimized to a runtime of approximately 0.5 sec.

Our goal is to minimize human interaction throughout the system setup process or, where human interaction is unavoidable, make the setup process more intuitive and less expert-dependent.

2 Technical Approach

Figure 2 shows an overview of the perception pipeline.

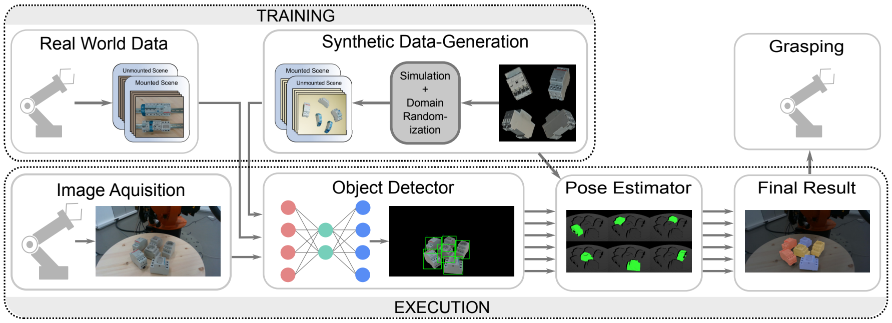


Fig. 2: Perception pipeline flowchart for the training and execution phase.

2.1 Dataset Creation

We created a freely available dataset (`www.dlr.de/rm/thr-dataset`) [10] that will be included in the next iteration of the SIXD Challenge [15].

In addition, we use an annotation tool to annotate ground truth execution data [10]. Assuming the camera position is known relative to the world model, only a single viewpoint per scene needs to be annotated. With this tool, a non-expert can produce the ground truth for image segmentation, classification, and pose estimation. The ground truth annotation tool was tested in an independent user study conducted by Tecnalia in which expert and non-expert users reported on its usefulness. The results of this user study will be publicly availabel `http://robdream.eu/`). While there remain possibilities for further automation, we found the current data acquisition and ground truth annotation setup was sufficient to label more than 6000 frames with correct segmentation and 6DoF pose information. To obtain more stable prior ground truth estimations, the information of several viewpoints can be fused to diminish drift effects of the camera position [17]. This approach has been used for the Top Hat Rail (THR) dataset and we plan to integrate it on our mobile platform.

2.2 Synthetic Data Generation

We used a previously reported system [1] to generate synthetic training data for object detection using photo-realistic rendering. Here, scene generation is based on variations of the training scenes from the THR dataset. Scenes can be extracted automatically from previously described experience logs [10] and an environment model using the RoboSherlock [3, 2] perception framework. We use UnrealCV [33] through the RobCog project (`http://www.robcog.org`), which offers extensions to communicate through ROS and a wrapper for UnrealCV to stream images over the network. The plugin developed for the RobCoG system offers basic interfaces for spawning and moving objects, as well as the virtual camera. These interfaces are implemented as standard ROS communication schemas. We generated nine variations of the original two test scenes from the THR dataset (five unmounted and four mounted) such that the appearance of each of the nine objects from the database is evenly distributed with a total of 2880 images per variation.

2.3 Object Detection and Pose Estimation

We generate 2D bounding boxes using RetinaNet [18] with a ResNet50 [12] backbone at a resolution of 800 pixels, as well as YOLOv2 [23] for comparison. In addition we use the corresponding depth data as input to the pose estimation method.

For pose estimation, we use a variant of the well-known generic PPF-based pose estimation method from depth data [9], as presented in the *SceneParser* framework [17]. The method is based on extracting parameters from multiple combinations of model surface point pairs and their corresponding normals [32]. The extracted parameters are then used as keys in per object hash tables to quickly find similar point pairs for a given candidate pair. Candidate point pairs are sampled randomly, and similar model point pairs are retrieved from the hash tables. For each combination of candidate and model point pairs, a rigid transformation can be calculated when considering the corresponding normals. Since the hash table keys are not unique, many possibly conflicting hypotheses will be generated. Therefore, another processing step is necessary, in which hypotheses are clustered and only those that are supported by a sufficient amount of candidate/model pairs are processed further. Then, quality values for the remaining hypotheses are calculated by rendering the objects in their corresponding poses, followed by a pixel-wise comparison of the resulting depth buffer to the acquired depth data. If the quality of the highest rated hypothesis exceeds a threshold, it is considered to explain the sensor data sufficiently. Finally, an ICP step is used for local registration.

3 Experiments and Results

The following two subsections focus on evaluating the detection and pose estimation components using our public benchmark dataset and during the Automatica live demo days (Fig. 3). The task of the vision system for the Automatica scenario includes marker less handling of the SLCs, automatic gathering of labeled training data of complex scenes for the detector, detection of the THR components in an SLC, and verification of the SLCs deliveries to different workstations.

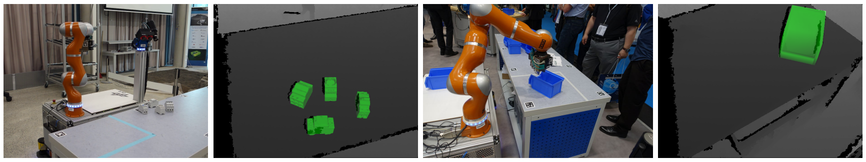


Fig. 3: AIMM detecting THR-elements and SLCs in the lab and at the Automatica fair. Objects are rendered in the estimated poses used for grasping.

Training	**Real Objects**			**Real Objects** **+ 2 Real Scenes**			**Unreal Scenes**		
Testing	All	M	U	All	M	U	All	M	U
YOLOv2	0.0576	0.0795	0.0646	0.6765	0.5544	0.7657	0.4828	0.3663	0.6392
RetinaNet	0.1471	0.0699	0.2342	0.7587	0.6563	0.8782	0.6307	0.5349	0.7163

Training	**Unreal Scenes** **+ Real Objects**			**Unreal Scenes** **+ 2 Real Scenes**			**Unreal Scenes** **+ 2 Real Scenes** **+ Real Objects**		
Testing	All	M	U	All	M	U	All	M	U
YOLOv2	0.5614	0.4196	0.7451	0.7602	0.6895	0.8129	0.8118	0.7568	0.8476
RetinaNet	0.6872	0.5939	0.7689	0.8404	0.7367	0.9091	0.8428	0.7368	0.9120

Table 1: YOLOv2 and RetinaNet mAP ($@0.5IOU$) values on the THR dataset test scenes - M and U indicate mounted and unmounted object scenes, respectively.

3.1 Benchmarking

The resulting mean average precision (mAP) of the bounding-box detection system [11] are shown in Table 1. Testing always happens on the real test scenes from the THR dataset [10]. Since the training scenes from the dataset only contain four types of objects, we train the network to detect only those objects using all other object types as negative examples.

The pose estimation results for detections with a confidence value greater than 0.5 are shown in Figure 4. In the reported results, we differentiate between results for mounted and unmounted scenes due to the step change in the difficulty between such scenes.

3.2 Robotic Experiments

For the Automatica shop floor logistics demo, we used our AIMM robot with a stereo pair on its gripper (for in-hand detection as a verification step) and an *rc_visard* by the DLR spinoff RoboCeption GmbH combined with a pattern projector, which was used for the main detection and pose estimation tasks.

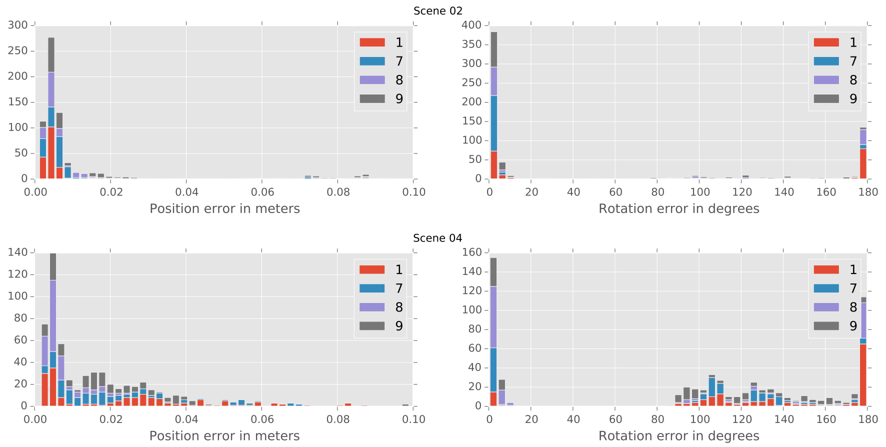


Fig. 4: Translational and rotational errors for objects 1, 7, 8, and 9 in the THR dataset test scenes. Top: Scene 2 with unmounted objects spread on a plane. Bottom: Scene 4 with objects mounted on a rail. High rotational errors of approximately 180° are due to the geometric (near-)symmetries of objects 1 and 8.

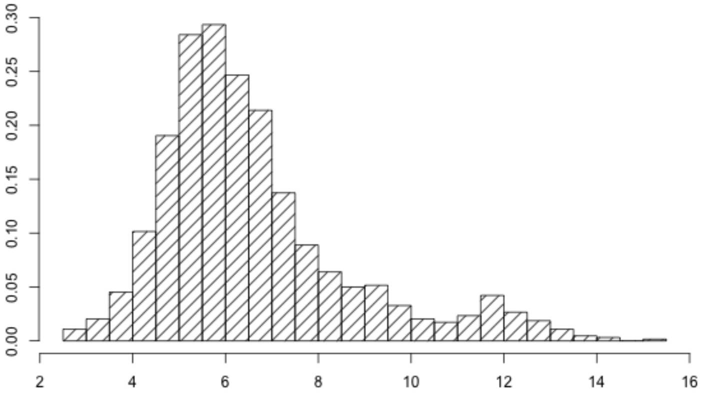


Fig. 5: For each of the 1282 SLC detections at Automatica, we rendered the best 6DoF match of the 3D model, ran Canny edge detection on it, and computed the edge pixels' mean distance to RGB edges. The distances were capped at 25 pixels to avoid false associations. The two peaks correspond to correct and flipped detections.

During the four days, our system was triggered for 1103 frames, resulting in 1282 SLC detections and corresponding poses. Since these resulted in successful grasps, we assume that most of the estimated poses were sufficiently accurate. Due to the different grasping failure detection steps, we were able to robustly handle the small amount of incorrect percepts, as well as the (intentionally introduced) human disturbance before, during, and after the vision system was triggered. In these cases, the same scenes were captured repeatedly until a grasp was successful, and no operator interaction was required. As we did not have ground truth for the scenes, we evaluated the detection and pose estimation jointly by reporting the average distance between the real edges detected in the RGB image and the rendering of the detected object in the estimated pose [26], as shown in Fig. 5, where errors of up to approximately 6-7 pixels seem visually appealing. Note that erros up to 8-9 pixels would still result in successful grasps most of the time. The small peak near 12 pixels is due to 90- or 180-degree flipped poses that sometimes occur, or due to occlusions/stacking.

While most detections resulted in pose estimates that were sufficiently accurate for successful grasping, Fig. 6 highlights several interesting cases. For each detection, 3D-based pose estimation was performed inside the bounding box without the use of prior knowledge about the expected poses. Besides misdetections, the most common mistakes were 180° (and rarely 90°) flips. As with THR elements, where geometrically similar objects are stacked s.t. their exact boundary and translational errors are difficult to detect while matching, detection and pose estimation errors are common. Perfectly aligned side-by-side SLCs typically result in a single detection and a correct pose. Similarly, in the case of stacked ones, the top (and occasionally the bottom) SLC is matched reliably.

4 Conclusions and Discussion

The experimental results show that for 2D object detection simulated data is, as of now, insufficient and a Domain Adaptation (DA) [5] step is necessary, i.e., using synthetic images combined with real data. By introducing real images during the training phase, the network is less sensitive to color nuance differences between the rendered and the real images. However, by simply generating realistic synthetic data, we achieved a 10% increase in performance on real data.

Improvements of the 2D object detection trained using only synthetic data could be obtained by generating more variations, applying more advanced transfer learning approaches, or by employing additional Domain Randomization (DR) [30, 14]. Furthermore, DA can be facilitated using deeper integration of autonomous modeling and ground truth annotation in the mobile robotic system.

For pose estimation, we can relatively often observe confusion with the geometrically symmetric view at 180° rotation because the camera image is not used. Therefore, to improve the results, we plan to add a camera-based classification step [10]. In addition, we plan to fuse our current detection pipeline with our recent DL-based one [28], as well as introduce a plausibility check based on the environment model and non-interpenetration [26].

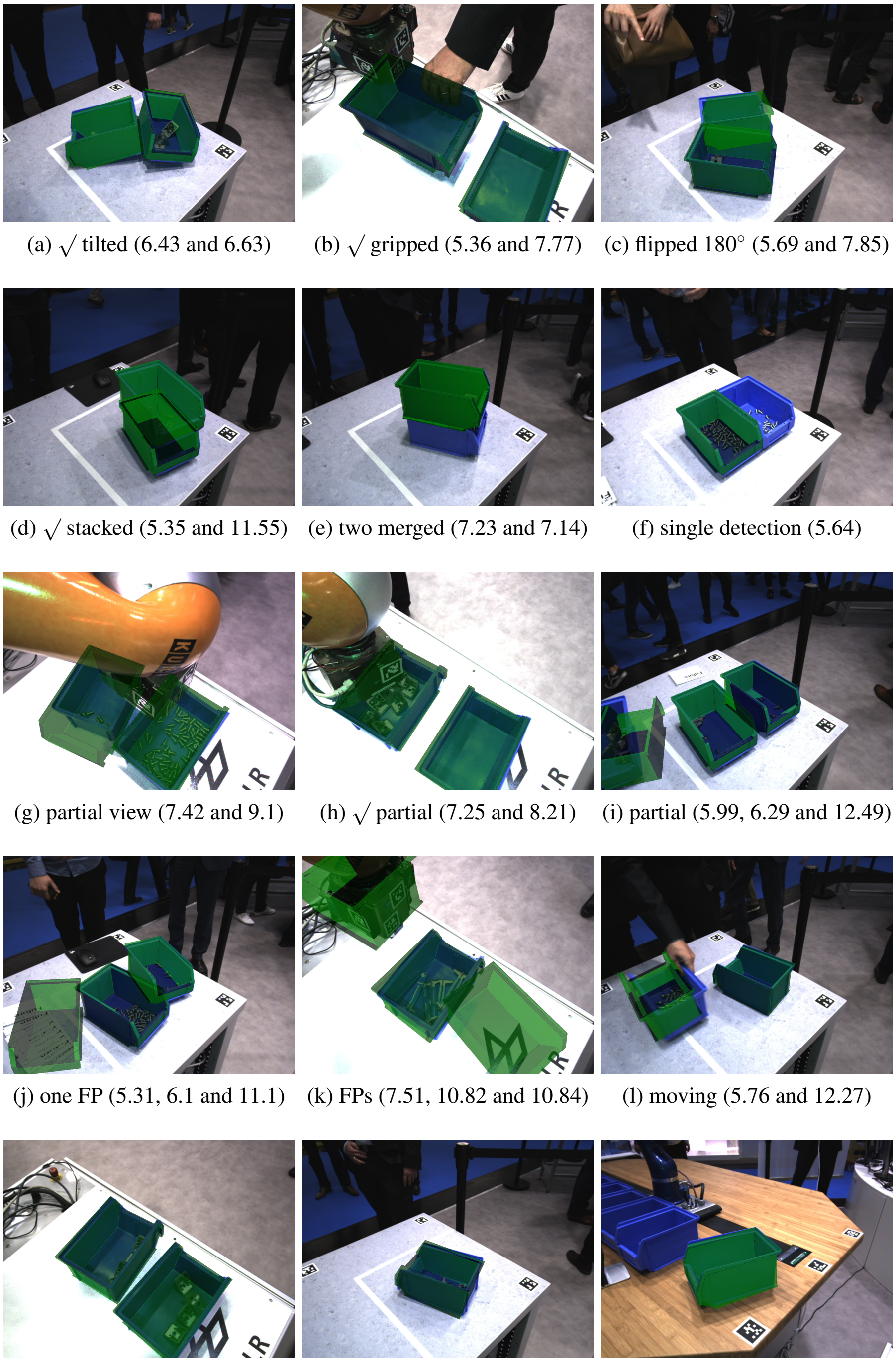
Note that the THR dataset is freely available online, the UnrealCV extension scenes will be published soon, and the scenes and detections logged during the live demo will be released to the community.

Acknowledgements We thank the entire team of the mobile manipulation Automatica demo at DLR, especially Andreas Dömel for coordination, Peter Lehner for the robotic manipulation, and Sebastian Riedel for the logging component. We also thank Amrutha Saseendran for her help with the detector, as well as the teams of the other three Factory of the Future demos.

References

1. Ferenc Balint-Benczedi and Michael Beetz. Variations on a theme: 'it is a strange kind of memory that only works backwards'. In *International Conference on Intelligent Robots (IROS)*. IEEE, 2018. under review.
2. Ferenc Balint-Benczedi, Zoltan-Csaba Marton, Maximilian Durner, and Michael Beetz. Storing and retrieving perceptual episodic memories for long-term manipulation tasks. In *18th International Conference on Advanced Robotics (ICAR)*. IEEE, 2017. Best Paper Finalist.
3. Michael Beetz, Ferenc Balint-Benczedi, Nico Blodow, Daniel Nyga, Thiemo Wiedemeyer, and Zoltan-Csaba Marton. RoboSherlock: Unstructured Information Processing for Robot Perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015. Best Service Robotics Paper Award.
4. Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. 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.
5. Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.
6. Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.
7. Andreas Dömel, Simon Kriegel, Manuel Brucker, and Michael Suppa. Autonomous pick and place operations in industrial production. In *12th International Conference on Ubiquitous Robots and Ambient Intelligence*, pages 356–356. IEEE, 2015. Best Video Paper Award.
8. Andreas Dömel, Simon Kriegel, Michael Kaßecker, Manuel Brucker, Tim Bodenmüller, and Michael Suppa. Towards fully autonomous mobile manipulation for industrial environments. *International Journal of Advanced Robotic Systems*, 2017.
9. B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 998–1005, June 2010.
10. Maximilian Durner, Simon Kriegel, Sebastian Riedel, Manuel Brucker, Zoltán-Csaba Márton, Ferenc Bálint-Benczédi, and Rudolph Triebel. Experience-based optimization of robotic perception. In *18th International Conference on Advanced Robotics (ICAR)*, pages 32–39. IEEE, 2017. Best Paper Finalist.
11. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.
12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
13. Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. In *European Conference on Computer Vision*, 2016.
14. Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. *arXiv:1710.10710*, 2017.
15. Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. In *European Conference on Computer Vision*, pages 606–619. Springer, 2016.

16. 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1529, 2017.
17. Simon Kriegel, Manuel Brucker, Zoltan Csaba Marton, Tim Bodenmüller, and Michael Suppa. Combining object modeling and recognition for active scene exploration. In *International Conference on Intelligent Robots and Systems*, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2384–2391, November 2013.
18. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.
19. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.
20. Chaitanya Mitash, Kostas E Bekris, and Abdeslam Boularias. A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 545–551. IEEE, 2017.
21. Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. How useful is photo-realistic rendering for visual learning? In *European Conference on Computer Vision*, pages 202–217. Springer, 2016.
22. Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. *arXiv preprint arXiv:1703.10896*, 2017.
23. Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
24. Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.
25. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
26. Tanner Schmidt, Katharina Hertkorn, Richard Newcombe, Zoltan Marton, and Dieter Fox. Depth-based tracking with physical constraints for robot manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 119–126, May 2015. Best Paper Award finalist.
27. Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015.
28. Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D orientation learning for 6D object detection from RGB images. In *The European Conference on Computer Vision (ECCV)*, September 2018.
29. Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. *arXiv preprint arXiv:1711.08848*, 2017.
30. Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
31. Joel Vidal, Chyi-Yeu Lin, and Robert Martí. 6d pose estimation using an improved method based on point pair features. *arXiv preprint arXiv:1802.08516*, 2018.
32. Eric Wahl, Ulrich Hillenbrand, and Gerd Hirzinger. Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification. In *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 474–481. IEEE, 2003.
33. Qiu Weichao, Zhong Fangwei, Zhanga Yi, Qiao Siyuan, Zihaom Xiao, Kim Tae Soo, Wang Yizhou, and Yuille Alan. Unrealcv: Virtual worlds for computer vision. *ACM Multimedia Open Source Software Competition*, 2017.



(a) √ tilted (6.43 and 6.63) (b) √ gripped (5.36 and 7.77) (c) flipped 180° (5.69 and 7.85)

(d) √ stacked (5.35 and 11.55) (e) two merged (7.23 and 7.14) (f) single detection (5.64)

(g) partial view (7.42 and 9.1) (h) √ partial (7.25 and 8.21) (i) partial (5.99, 6.29 and 12.49)

(j) one FP (5.31, 6.1 and 11.1) (k) FPs (7.51, 10.82 and 10.84) (l) moving (5.76 and 12.27)

(m) √ with 3.74 and 6.05 (n) × misaligned with 7.31 (o) × misaligned with 7.64

Fig. 6: Example of success (√) and failure (×) cases with mean edge pixel errors.