

KLT Bin Detection and Pose Estimation in an Industrial Environment

Aleksei Beloshapko[1(✉)], Christian Knoll[2], Bilel Boughattas[3], and Vladimir Korkhov[1]

[1] Saint-Petersburg State University, 7/9 Universitetskaya nab., St. Petersburg 199034, Russia
beloshapko-alexey@rambler.ru, v.korkhov@spbu.ru
[2] Robert Bosch GmbH, Robert-Bosch-Campus 1, 71272 Renningen, Germany
Christian.Knoll@de.bosch.com
[3] University of Stuttgart, Postfach 10 60 37, 70049 Stuttgart, Germany
bilel.boughattas@gmail.com

Abstract. In order for Automated Guided Vehicles (AGV's) to handle KLT bins (Kleinladungsträger, Small Load Carrier) in a flexible way, a robust bin detection algorithm has to be developed. This paper presents a solution to the KLT bin detection and pose estimation task. The Mask R-CNN network is used to detect a KLT bin on color images, while a simple plane fitting approach is used to estimate its 5DoF position. This combination gives promising results in a typical use case scenario when the KLT bin is aligned with the camera view.

Keywords: KLT bin picking · Object detection · Instance segmentation · Pose estimation · Plane fitting · Industry 4.0

1 Introduction

A constant evolution of the information technologies brings changes to the industry. There is a huge trend for improving flexibility, transparency and automation of the production plants, which is a part of adoption of the Industry 4.0 paradigm. In particular, new technologies allow for better customization and flexibility of shop-floor production and logistics. Therefore, they loosen some of the restrictions imposed to allow for automation in the past, and generate a bulk of new tasks to be solved.

An integration of AGVs (Automated Guided Vehicles) for internal production plant logistics allowed a better flexibility in comparison to conveyors. However, these vehicles require smart software to perform the task of goods and material picking and delivery. One of the common ways to keep the workpieces and products is to use KLT bins (Kleinladungsträger, also known as a Small Load Carrier). This creates a task of a localization of such transport containers in an industrial environment in order to pick them.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 105–118, 2020.
https://doi.org/10.1007/978-3-030-58817-5_9

2 Related Work

The task of KLT bin detection has attracted little attention of researchers. In most applications where KLT bins are used, the software can assume its position to be fixed. This includes an extensively investigated task of "order picking", see [5,9,15] for an example.

This work is heavily related to [4], where the case of KLT bin detection with fisheye cameras is investigated. The main difference is the type of camera in use and a different use case scenario: in that work, the target distance to the KLT bin was assumed extremely small. In this work, we use a Color-Depth camera, and do not make any strict assumptions about KLT bin position.

Another attempt to solve this task was made by [14]. They concentrated on a particular case where the KLT bin is observed from the top, so that only edges of vertical sides of a bin are visible. They developed an algorithm based on KLT bin top edges detection, followed by a RANSAC-based procedure to choose the best hypothesis.

A related investigation was made by [20] to detect pallets in an industrial environment with range data only. The main idea was to use deep learning instead of classical computer vision approaches to detect the object of interest in a range data. We also use deep learning for object detection, and the neural networks in use are similar. However, we use the color image for object detection, and the range image is used for pose estimation only.

The task is also related to more general fields of research: object detection and object pose estimation.

2.1 Object Detection

In certain situations, the classical computer vision approach could be successfully applied for object detection. If the 3D shape of the object is known in advance, a shape matching algorithm like [8] can be applied. If the appearance of the object is known, then there is an option to use a keypoint feature matching approach like SURF [3] or ORB [23]. If the object does not have enough texture for feature matching, but has enough discriminative contours, then a 2D shape-based matching [28] can be applied.

However, a KLT bin lacks for 3D shape information as it consists of plane segments. In addition, no texture information is available. The contours of the object consist of straight lines, so in this sense a KLT bin is easily blended into an industrial environment. Thus, it is not possible to rely on some single aspect of object appearance. This leads us to the development of a detector that accepts the raw object image as the input. The use of machine learning methods is the most promising option that addresses these issues.

Nowadays the research on object detection on color and range images is largely driven by neural networks. In this work we use Mask R-CNN network [11] to perform KLT bin detection and segmentation. This solution avoids hardcoding the object appearance, but requires to collect a large training dataset instead. The network itself remains a state-of-the-art solution for instance segmentation

tasks, but possibly could be optimized by using a more recent and optimized backbone classifier like EfficientNet [25].

2.2 Object Pose Estimation

There is a wide choice of methods for pose estimation. In this work, we rely on box shape of a KLT bin, thus using a plane segmentation and fitting approach [24] as a base algorithm. If a 3D shape matching algorithm was used for object detection, then the pose estimate could be obtained after an iterative closest-point refinement [29]. There are also a number of methods based on deep learning that handle the task of pose estimation, see [13,26,27].

3 The KLT Bin Detection and Pose Estimation Task

In this paper, we present an approach to KLT bin detection and pose estimation using a RGB-D camera [18]. We define three distinct scenarios that we handle:

1. Horizontal view case: The camera and the KLT bin are aligned and only the side part of the bin is visible
2. Vertical view case: The camera and the KLT bin are aligned and only the top part of the bin is visible
3. General view case: The camera and the KLT bin are not aligned so that more than one side of the bin is visible

The KLT bin and the camera are said to be aligned if the optical axis of the camera is almost parallel to some of the KLT bin sides (with tilt tolerance up to 5° of rotation). The aligned positions of a KLT bin are of the special interest for us, because these scenarios are a typical case in the industrial environment.

The task is to localize all the KLT bins in the 3D space that are visible to the camera. The task is solved in two stages: a KLT bin detection and a KLT bin pose estimation. The detection stage uses the Mask R-CNN network to process the color image and generate region proposals that contain KLT bins. The second stage processes point clouds that correspond to these regions and generates the pose estimates. The core algorithm for this is the RANSAC-based plane detection followed by a least-squares refinement. The pose estimation step is only performed for the aligned case. See Fig. 1 for an overview.



Fig. 1. Overview of the pipeline for KLT bin detection and pose estimation

4 KLT Bin Detection with Mask R-CNN

Our solution is based on the object detection capabilities of the Mask R-CNN [11] neural network. This network takes a color image as an input, and produces object bounding boxes alongside with their object segmentation masks and class labels. We modified the network to detect objects of only two classes: a KLT bin class and not-a-bin class (referred to as a "background" class). For each bounding box with KLT bin class label, we get a segmentation mask that we later use for pose estimation.

4.1 Mask R-CNN

The network itself consists of four major image processing stages, which are depicted in Fig. 2. The first step is to use a classification network in a fully-convolutional manner to generate a pyramid of feature maps. We use the ResNet-101 network [12] as a classification backbone. The second stage is to produce a set of rough bounding boxes that might contain an object using the top feature map of the pyramid. This is done with a Region Proposal Network [21]. At the third stage the location of generated bounding boxes is being refined. For each rough bounding box estimate, the network predicts deltas for top-left corner location and for box size. Finally, the last stage generates a class label and a segmentation mask for each of the refined object bounding box.

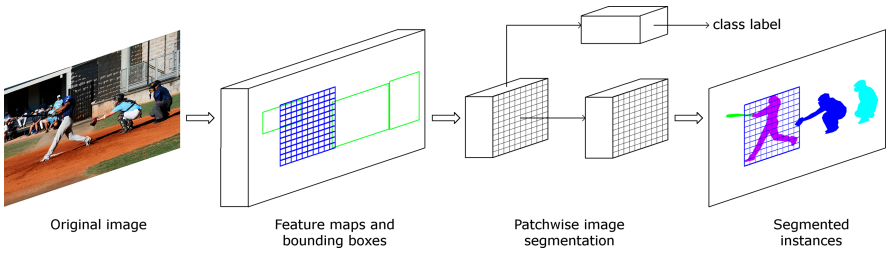

Fig. 2. The Mask-RCNN network pipeline [11]

4.2 Training Dataset

To train such a network, we created a training dataset for KLT bin instance segmentation. The raw data was obtained with Intel RealSense cameras (D415 and D435). The labeling was performed either manually, or based on color thresholding, or with a semi-automatic approach. Each labeled image went through our data augmentation pipeline to enlarge the training dataset and improve data variety.

Semi-automatic Labeling Procedure. The semi-automatic procedure was introduced to label a large amount of images with less effort than just manual labeling. The idea is to capture a video sequence of the object of interest and then use a surface reconstruction method to create a 3D mesh of the captured area. Then labeling of this 3D mesh is equivalent to labeling of the complete video sequence.

The procedure consists of 5 steps:

1. A Color-Depth video sequence of a KLT bin is captured
2. The video sequence is processed with a surface reconstruction method to produce a 3D mesh of the area and a camera relative pose estimate for each image frame
3. The 3D mesh labeling is performed
4. The labeled parts of the mesh are back-projected to each of the images of the sequence. This is possible with the corresponding camera pose estimates. The projection of a labeled part generates an object segmentation mask.
5. The resulting segmentation masks are refined for each image of the sequence.

At Step 1 a human captures a video sequence while going around an area with some KLT bins. It is better to make a complete circle, or a path with shortcuts to ensure loop closures at the reconstruction stage.

At Step 2, the depth images sequence is fed to some 3D surface reconstruction method. We used KinectFusion [17] and Open3D [6] for mesh generation. While KinectFusion is faster, it does not use loop closures, so it quickly accumulates a camera pose drift. Open3D reconstruction pipeline is completely offline, but uses loop closures to improve the quality of the mesh and the camera trajectory estimate. Therefore, KinectFusion is more suitable for short sequences, while Open3D is able to handle long image sequences, but takes more time to process.

At Step 3, one has to manually specify the parts of the mesh that belong to KLT bins. We used Meshlab [7] to simply crop out parts that are KLT bins. As this step is manual, the complete procedure is semi-automatic. To make the procedure fully automatic, one has to automate this particular step.

At Step 4, the KLT bin meshes are back-projected to the original images of the video sequences to form the segmentation masks. As for each image frame there is a camera pose estimate, the back-projection is done simply by projecting all mesh faces to the camera frame.

At Step 5, all the segmentation masks are refined via a Grab-cut based procedure. For this step we used an OpenCV implementation of the Grab-cut [22] algorithm. For each segmentation mask, the inner part of the region is treated as “sure-foreground”, while the outer part as “sure-background”. The area near the region boundary was treated as either “probably-foreground” or “probably-background”. With this initialization, the Grab-cut algorithm was run for a single iteration on the color image. This step removes some specific artifacts of the segmentation, such as decimation effect due to the voxel-based nature of the mesh.

As a result, for each image in a video sequence the KLT bin segmentation masks are generated.

Data Augmentation Pipeline. The data augmentation primary target is to make a more representative dataset. Usually neural networks trained with the use of data augmentation are more robust to certain challenging situations. Our augmentation pipeline perform color augmentation image rotation and background substitution.

Color augmentation is performed along the hue axis. This ensures that the training dataset contains KLT bins of any color.

Random image rotation is performed in a standard way: the image is rotated in the image plane around its center. The complete rotation angle range of 360° is used.

After these three transformations are applied, the resulting KLT bin region is cut-and-pasted onto some other background image. This ensures background variation, as we captured the KLT bins images at just several background scenes.

4.3 Experiments

The training data consists of 11000 labeled images. 4200 of them are obtained with the semi-automatic labeling procedure, 6500 with background subtraction method and 300 were labeled manually. 10% of the semi-automatically labelled images were used for training as they are, while the rest went through the data augmentation pipeline, with each image producing 10 new ones. Thus, the final training data set had 106220 images with mask labels. See Fig. 3 for some training images samples.

The Mask R-CNN network was trained for 20 epochs. Each epoch had 1200 optimization steps with the batch size of 1. We used a 3-stage training scheme, where during the first two stages the ResNet weights were partially fixed. Batch Normalization [16] was used to regularize the network.

The test dataset consisted of 390 manually labeled images captured at 8 different scenes. See Fig. 4 for samples.

The mean Intersection over Union (IoU) metric is used to evaluate the quality of image segmentation. This metric is one of the standard ways to evaluate the image segmentation accuracy. The IoU metric has the value in range $[0, 1]$ and the value of 1 means perfect match of the prediction mask with the ground truth.

We also estimate the object detection performance with a confusion matrix. Object detection is implemented by prediction the image bounding boxes. This functionality is a part of the Mask R-CNN pipeline.

We estimate these three entries of the confusion matrix: the True Positive (TP) rate, the False Positive (FP) rate and the False Negative (FN) rate. The TP rate is the number of correct bounding box predictions divided by the number of bin bounding box predictions. The False Positive rate is the number of incorrect bounding box predictions divided by the total number of bin bounding box predictions. The False Negative rate is the number of ground truth bin regions that were not detected divided by the total number of ground truth bin regions.

This way the FP rate indicates the responses on the non-existent bin, while FN rate indicates real bins that were not detected. The sum of the True Positive and the False Positive rate is always 1.



Fig. 3. Samples of images that were generated with a data augmentation pipeline and used to train the Mask R-CNN network for KLT bin detection



Fig. 4. Some examples of the test images. Left image relates to the vertical view scenario, the center image - to the horizontal view scenario, right image shows a general position case

Table 1 reports the results. It can be seen that Mask R-CNN handles well the vertical and general view cases, while failing a lot in the horizontal view case. A data inspection revealed that the majority of the detection misses belong to a single bin type that appears a lot in the test dataset, but does not in the training dataset. This indicates that the network works poorly on the bin types that were not trained. See Fig. 5 for an example of bin detection.

The histogram Fig. 6 shows the IoU metric value distribution across regions generated by the Mask R-CNN. Most of the detections fall in the range of 0.8–1.0, having few outliers.

Table 1. Evaluation of the Mask R-CNN network over test datasets

Test set	Total images	TP rate	FN rate	FP rate	mIoU
Vertical view	250	0.960	0.035	0.039	0.939
Horizontal view	160	0.989	**0.399**	0.010	0.919
General view	59	0.983	0.016	0.016	0.912



Fig. 5. An example of an image processed with the Mask R-CNN pipeline. The detected bin instances are labeled with a bounding box rectangle and painted with the predicted mask region. Note that the blue bin in the right part of the image is not detected (Color figure online)

5 KLT Bin Pose Estimation Using Plane Fitting

For the aligned use cases when only one side of a KLT bin is observed, its relative position can be estimated by fitting a plane into the KLT bin point cloud. See Fig. 7 for a KLT bin point cloud sample. This cloud can be generated out of the depth image and the segmentation provided by Mask R-CNN, assuming that camera intrinsic parameters are known.

To conduct the experiment, we used the implementation of the plane fitting algorithm from Point Cloud Library [24]. The method accepts a point cloud and does a RANSAC-based plane fitting followed by least-squares refinement. As a result, we obtain a plane center point and a normal vector estimate for each KLT bin region.

The accuracy of such approach is validated against another pose estimation method that relies on tracking of a fixed checkerboard pattern.

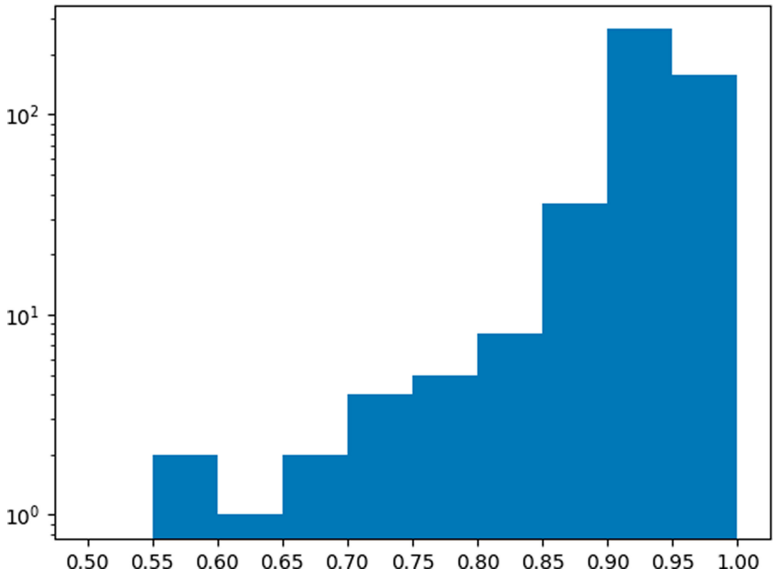

Fig. 6. A log-scale histogram of IoU metric values for the regions generated with Mask R-CNN on the test image set

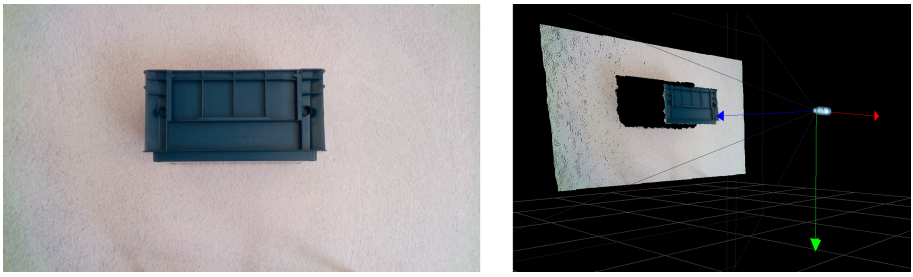

Fig. 7. The image of a KLT bin over uniform background and the corresponding colored pointcloud captured by the RGB-D camera. Figure made with the Intel RealSense Viewer. (Color figure online)

5.1 Camera Pose Estimation via Charuco Marker Tracking

The ground truth for camera pose estimation is obtained with the Charuco marker [10] tracking approach. The Charuco marker is the planar checkerboard pattern with AruCo markers [2] inside the checkerboard cells. The advantage of using such a pattern instead of a regular checkerboard pattern is that it is less ambiguous when the pattern is partially occluded. When tracking a checkerboard pattern, it is usually required to have all checkerboard corners visible. The use of AruCo markers makes it possible to recover feature point correspondences even if some of the feature points are occluded.

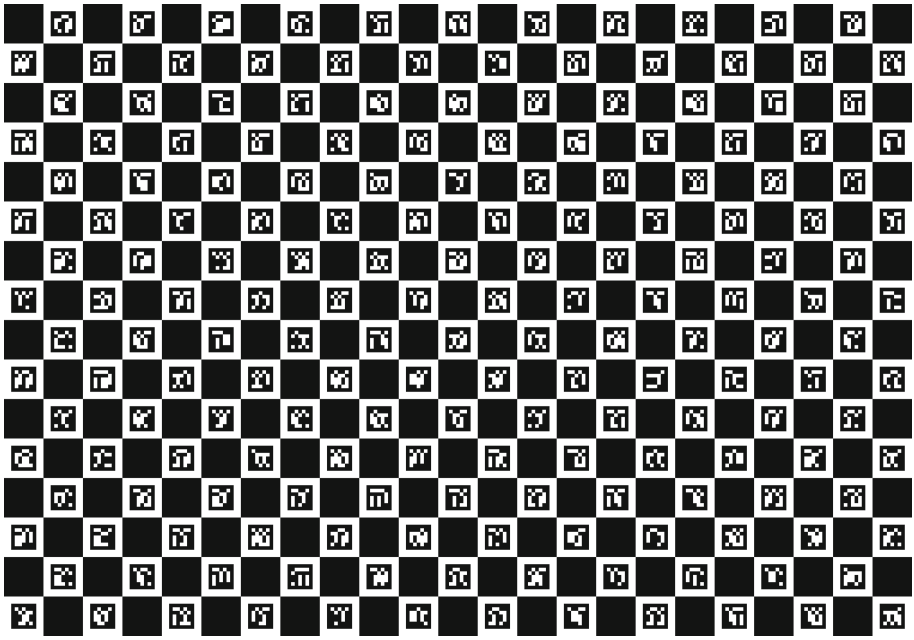

Fig. 8. The Charuco marker used for camera pose estimation

The use of the checkerboard pattern allows feature detection with subpixel accuracy. Having accurate correspondences between 3D marker points and their projections on the color image frame, it is possible to formulate a Perspective-n-Point problem [19], that could be solved by minimizing the feature point reprojection error. Given a sufficient number of points, this method achieves state-of-the-art accuracy [1].

We use a Charuco board with 24×17 cells. The checkerboard cell size is 51 mm, and each of the internal white cell contains an AruCo marker. AruCo marker dimensions is 0.8 times smaller than the hosting checkerboard cell. A total of 184 AruCo markers are in use. See Fig. 8.

The AruCo marker dictionary was generated with the procedure described in [10]. In short, we set a minimum hamming distance threshold between AruCo markers being equal to 3. Then we randomly sample the AruCo markers and add them to a dictionary if its distance to all markers in the dictionary is less than the threshold. This way, we obtain a collection of AruCo markers that have significant difference between each other.

Of course, the hamming distance threshold of 3 is quite a low requirement. This comes as a trade-off of having a large AruCo marker dictionary.

The KLT bin is placed directly on the Charuco marker, see Fig. 9.



Fig. 9. Sample images used to evaluate the pose estimation precision. Left and center - KLT bin side (horizontal) views, right - top (vertical) view

The KLT bin occludes some part of the marker, but the use of internal markers makes the corner identification still possible.

5.2 Experiments

We simulated three different scenarios: horizontal views on sides of a KLT bin and a vertical view, see Fig. 9. The camera was moved around being pointed to the KLT bin and capturing Color-Depth images. For each of such images, we obtain a full 6DoF pose with Charuco marker tracking approach.

For each of the images that have the pose estimated with sufficient accuracy, we run our KLT bin detection pipeline. First, Mask R-CNN network estimates the region of the image where the KLT bin appears. Second, the plane fitting is run on the part of point cloud that corresponds to that region. This way we obtain a center point and the normal vector of the KLT bin. Finally, the center point estimate is expressed in the global coordinate frame defined by the Charuco marker.

It has to be noted that the ChAruCo pattern as a background creates an additional challenge for object detection and instance segmentation. This pattern is similar to bin regions as it also contains little texture information and a large amount of straight lines. This influences the segmentation accuracy, which in turn has an influence on the pose estimation accuracy.

We evaluate the precision bin centre estimates. To do so, we estimate a centroid point of all these estimates, and then compute the average deviation. Table 2 shows the details.

Table 2. KLT bin center point estimation precision evaluation. All precision values are given in meters

Scenario	Total images	Average center deviation	Max center deviation	Camera viewpoint variation	Max camera viewpoint shift
Side view #1	33	**0.0321**	0.105	0.2107	0.5938
Side view #2	29	**0.0316**	0.0876	0.2155	0.5488
Top view	193	**0.0360**	0.1490	0.1939	0.7082

Given these precision values, it is clear that the pipeline gives a rough position estimate of a bin. However, these evaluation conditions are more challenging than the typical use case, so we treat these precision estimates as an upper precision boundary.

6 Conclusion

The presented method can be applied for KLT bin detection and pose estimation in an industrial environment. The strong sides of the proposed solution is the flexibility of the bin detection algorithm that is based on machine learning techniques, along with the simplicity of the pose estimation approach.

The weak spots are the bin detection rate and the segmentation accuracy. The use case of a horizontal view on a KLT bin has to be additionally addressed. The direct way to reduce the False Negative rate is to populate the training dataset with more images of non-detected bins.

The plane fitting approach used for pose estimation is a simple and effective option, but it depends heavily on the region proposal accuracy generated by the neural network. One of the easiest way to improve pose estimation precision is to perform a time-based filtering with a Kalman filter. This is an option in the case when the camera trajectory is known. Another way is to introduce a more sophisticated pose estimation algorithm. The techniques based on machine learning could be the preference to avoid further constraints on the object appearance.

References

1. An, G.H., Lee, S., Seo, M.W., Yun, K.J., Cheong, W.S., Kang, S.J.: Charuco board-based omnidirectional camera calibration method. Electronics **7**, 421 (2018)
2. Babinec, A., Jurišica, L., Hubinský, P., Duchoň, F.: Visual localization of mobile robot using artificial markers. Procedia Eng. **96**, 1–9 (2014). https://doi.org/10.1016/j.proeng.2014.12.091
3. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_32
4. Beloshapko, A., Korkhov, V., Knoll, C., Iben, U.: Industrial fisheye image segmentation using neural networks. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11622, pp. 678–690. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24305-0_50
5. Buchholz, D., Kubus, D., Weidauer, I., Scholz, A., Wahl, F.M.: Combining visual and inertial features for efficient grasping and bin-picking. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 875–882 (2014)
6. Choi, S., Zhou, Q.Y., Koltun, V.: Robust reconstruction of indoor scenes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5556–5565 (2015)
7. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: MeshLab: an open-source mesh processing tool. In: Eurographics Italian Chapter Conference, vol. 1, pp. 129–136 (2008). https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136

8. Drost, B., Ilic, S.: 3D object detection and localization using multimodal point pair features. In: 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission, pp. 9–16 (2012)
9. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3D object recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 998–1005 (2010)
10. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., Marín-Jiménez, M.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn. **47**, 2280–2292 (2014). https://doi.org/10.1016/j.patcog.2014.01.005
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. CoRR abs/1703.06870 (2017). http://arxiv.org/abs/1703.06870
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385
13. Hodan, T., Haluza, P., Obdrzálek, S., Matas, J., Lourakis, M.I.A., Zabulis, X.: T-LESS: an RGB-D dataset for 6d pose estimation of texture-less objects. CoRR abs/1701.05498 (2017). http://arxiv.org/abs/1701.05498
14. Holz, D., Behnke, S.: Fast edge-based detection and localization of transport boxes and pallets in RGB-D images for mobile robot bin picking. In: Proceedings of ISR 2016: 47st International Symposium on Robotics, pp. 1–8 (2016)
15. Holz, D., et al.: Active recognition and manipulation for mobile robot bin picking. In: Röhrbein, F., Veiga, G., Natale, C. (eds.) Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe. STAR, vol. 94, pp. 133–153. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-03838-4_7
16. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167 (2015). http://arxiv.org/abs/1502.03167
17. Izadi, S., et al.: Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST 2011, New York, NY, USA, pp. 559–568. Association for Computing Machinery (2011). https://doi.org/10.1145/2047196.2047270.https://doi.org/10.1145/2047196.2047270
18. Keselman, L., Woodfill, J.I., Grunnet-Jepsen, A., Bhowmik, A.: Intel realsense stereoscopic depth cameras. CoRR abs/1705.05548 (2017). http://arxiv.org/abs/1705.05548
19. Lu, X.: A review of solutions for perspective-n-point problem in camera pose estimation. J. Phys. Conf. Ser. **1087**, 052009 (2018). https://doi.org/10.1088/1742-6596/1087/5/052009
20. Mohamed, I.S., Capitanelli, A., Mastrogiovanni, F., Rovetta, S., Zaccaria, R.: Detection, localisation and tracking of pallets using machine learning techniques and 2D range data. CoRR abs/1803.11254 (2018). http://arxiv.org/abs/1803.11254
21. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR abs/1506.01497 (2015). http://arxiv.org/abs/1506.01497
22. Rother, C., Kolmogorov, V., Blake, A.: Grabcut-interactive foreground extraction using iterated graph cuts. In: ACM Transactions on Graphics (SIGGRAPH), August 2004. https://www.microsoft.com/en-us/research/publication/grabcut-interactive-foreground-extraction-using-iterated-graph-cuts/

23. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision, pp. 2564–2571 (2011)
24. Rusu, R., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA 2011), May 2011. https://doi.org/10.1109/ICRA.2011.5980567
25. Tan, M., Le, Q.V.: EfficientNet: rethinking model scaling for convolutional neural networks. CoRR abs/1905.11946 (2019). http://arxiv.org/abs/1905.11946
26. Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S.: Deep object pose estimation for semantic robotic grasping of household objects. CoRR abs/1809.10790 (2018). http://arxiv.org/abs/1809.10790
27. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes. CoRR abs/1711.00199 (2017). http://arxiv.org/abs/1711.00199
28. Xu, X., Zhang, X., Han, J., Wu, C.: HALCON application for shape-based matching. In: 2008 3rd IEEE Conference on Industrial Electronics and Applications, pp. 2431–2434 (2008)
29. Zinsser, T., Schmidt, J., Niemann, H.: A refined ICP algorithm for robust 3-D correspondence estimation. In: Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429), vol. 2, p. II-695 (2003)