Practical aspects of detection and grasping objects by a mobile manipulating robot

Bartłomiej Kulecki, Kamil Młodzikowski, Rafał Staszak and Dominik Belter
Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poznan, Poland

Abstract

Purpose – The purpose of this paper is to propose and evaluate the method for grasping a defined set of objects in an unstructured environment. To this end, the authors propose the method of integrating convolutional neural network (CNN)-based object detection and the category-free grasping method. The considered scenario is related to mobile manipulating platforms that move freely between workstations and manipulate defined objects. In this application, the robot is not positioned with respect to the table and manipulated objects. The robot detects objects in the environment and uses grasping methods to determine the reference pose of the gripper.

Design/methodology/approach – The authors implemented the whole pipeline which includes object detection, grasp planning and motion execution on the real robot. The selected grasping method uses raw depth images to find the configuration of the gripper. The authors compared the proposed approach with a representative grasping method that uses a 3D point cloud as an input to determine the grasp for the robotic arm equipped with a two-fingered gripper. To measure and compare the efficiency of these methods, the authors measured the success rate in various scenarios. Additionally, they evaluated the accuracy of object detection and pose estimation modules.

Findings – The performed experiments revealed that the CNN-based object detection and the category-free grasping methods can be integrated to obtain the system which allows grasping defined objects in the unstructured environment. The authors also identified the specific limitations of neural-based and point cloud-based methods. They show how the determined properties influence the performance of the whole system.

Research limitations/implications – The authors identified the limitations of the proposed methods and the improvements are envisioned as part of future research.

Practical implications – The evaluation of the grasping and object detection methods on the mobile manipulating robot may be useful for all researchers working on the autonomy of similar platforms in various applications.

Social implications – The proposed method increases the autonomy of robots in applications in the small industry which is related to repetitive tasks in a noisy and potentially risky environment. This allows reducing the human workload in these types of environments.

Originality/value – The main contribution of this research is the integration of the state-of-the-art methods for grasping objects with object detection methods and evaluation of the whole system on the industrial robot. Moreover, the properties of each subsystem are identified and measured.

Keywords Grasping, 3D perception, Mobile manipulation

Paper type Research paper

1. Introduction

Mobile manipulating platforms become popular in industry (Dömel *et al.*, 2017). They also find application in homes and hospitals to support people in daily activities (Jain and Argall, 2016). Mobile manipulating platforms combine the positive properties of mobile robots and industrial robotic arms. They can transport goods and manipulate objects at the same time. However, to obtain full autonomy of such robots in a real industrial environment or continuously changing domestic surroundings the robot should use algorithms for autonomous navigation (Droeschel *et al.*, 2017), motion planning (Huang *et al.*, 2000) and grasping (Morrison *et al.*, 2018; Mahler *et al.*, 2019).

Typical mobile-manipulating robots in the industry are precisely docked and positioned in the workspace before they start objects manipulating. They can be docked mechanically with the workstation (gon Roh *et al.*, 2008) or they can use visual markers to determine the pose of the robot with respect to the pose of the workstation (Andersen *et al.*, 2013). In this research, we are interested in different scenarios. We assume that the

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/0143-991X.htm**

Industrial Robot: the international journal of robotics research and application
48/5 (2021) 688–699
Emerald Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/IR-10-2020-0242]

This work was supported by the National Centre for Research and Development (NCBR) through project LIDER/33/0176/L-8/16/NCBR/2017. The authors would like to thank Justyna Ataman for the support during the implementation of the GG-CNN on the UR5 robot and during experiments on the robot.

Received 31 October 2020
Revised 11 December 2020
15 January 2021
Accepted 16 January 2021

workspace does not have artificial markers and the environment is unstructured. Thus, the objects are not positioned by external tools and their poses are *a priori* unknown.

In this article, we consider the following scenario. The robot moves autonomously in the environment. It can move between workspaces and on the given position the robot should perform the manipulation tasks. We focus on the pick-and-place tasks where the robot has to grasp objects and place them in the given position. The example scenario is presented in Figure 1. We consider two types of grasping tasks. In the first type of task, the robot cleans the table and does not know the category of objects. The robot grasps all objects from the table and places them in the container (Figure 1). In the second scenario, the robot should know the category of the objects and should grasp the objects which are related to the task. In this case, the robot should recognize the objects on the scene and then grasp them.

The practical aspect of this work is related to grasping various objects. We are looking for methods that allow the robot to grasp both known and unknown categories of objects. The robot should be able to move freely in the environment and manipulate objects related to the tasks. If the robot is asked to bring the object from the given category, the robot should be able to find and grasp the found object.

To grasp the object of the given category, we verified two approaches. The first method uses point clouds to find the objects on the scene and determine the configuration of the gripper which allows grasping the object. In the second approach, we use *state-of-the-art* Convolutional Neural Network-based (CNN-based) methods for grasping. We combine the CNN-based object detection and grasping methods to obtain the new system which can be used to grasp objects from the given categories related to the tasks performed by the robot. In this approach, the robot does not have to be precisely positioned with respect to the workstation. The visual feedback allows finding and grasping objects independently on the pose of the objects with respect to the robot.

Figure 1 Robot collecting objects using CNN for object detection and grasping

2. Related work

The example mobile manipulating platforms used in research are Rollin' Justin (Dietrich *et al.*, 2016), Cosero (Stückler *et al.*, 2016), Armar (Berns *et al.*, 2000), PR2 (Bohren *et al.*, 2011), Centauro (Klamt *et al.*, 2020) and Little Helper (Hvilshøj *et al.*, 2009). Also, many commercial platforms like Apas, Tiago, Talos and Fetch Mobile Manipulator robots are available. The extended review of the mobile manipulating platforms is available in Klamt *et al.* (2020). The researches on these platforms focus on localization (Droeschel *et al.*, 2017), mapping (Droeschel *et al.*, 2017), objects detection (Huang *et al.*, 2017), objects pose estimation and manipulation (Mahler *et al.*, 2019).

In this article, we focus on grasping objects by a robot equipped with a two-fingered gripper and an RGB-D sensor for 3 D perception. To grasp the objects, the robot should detect the objects in the environment. Current *state-of-the-art* methods for object detection are based on the CNN. The YOLO detector predicts the position of the bounding boxes on the image and classifies the object inside this region (Redmon and Farhadi, 2020). The Single Shot Detector (Liu *et al.*, 2016) uses more convolutional layers than YOLO to operate on feature maps with various resolutions using additional computational resources. Additionally, the Mask R-CNN (He *et al.*, 2017) performs segmentation inside the bounding box to determine which pixels belong to the detected objects. Pose estimation of objects in the 3 D space is also solved using CNN. Because the estimation of the object orientation is more challenging than the estimation of the translation these problems are decoupled (Xiang *et al.*, 2018). Also, using depth data helps with the 6D pose estimation of objects.

Most of the perception systems of the robots, which enables the robot to adapt to the changing state of the environment and configuration of the objects, are vision-based. These methods use RGB, depth images, point clouds or 3D meshes (Fischinger and Vincze, 2012; Kootstra, 2012; ten Pas and Platt, 2014). The 2D representations like images are efficient in searching the grasping points but the 3D models like point clouds or 3 D meshes allow searching for kinematically feasible, collision-free configurations of dextrous robotics hands (Kopicki *et al.*, 2019). In this article, we compare two methods. The first method uses a 3D point cloud to determine the configuration of the griper. The second method uses depth images only.

Most research is focused on grasping various objects independently on the categories of the objects. These methods might be represented by probabilistic inference using point cloud (Kopicki *et al.*, 2015). The grasping algorithm uses local geometric properties (features) of the objects to define grasping points. The algorithm also takes into account the configuration of the hand and collisions with the objects. Further clusterization of the data and grasps allows inferring about the configuration of the robotic hand from a single viewpoint without direct reconstruction of the object (Kopicki *et al.*, 2019). The main advantage of the probabilistic-based methods is that they require a few examples only to train the robot (Song

et al., 2015). It is also straightforward to incorporate the uncertainty of the data in the model (Johns *et al.*, 2016). Other grasping frameworks utilize searching methods like graph search (Hang *et al.*, 2017) or sampling-based search (Liu and Carpin, 2015) to find the force-closure stable grasps.

Recently, CNN-based grasping methods became popular. The neural network can find the relation between the input image and the grasping point directly from the training data. The main problem with these methods is data collection. To collect the training data synthetic models can be sampled (Mahler *et al.*, 2017; Satish *et al.*, 2019). The latest version of the Dex-Net 4.0 used to collect objects from the container achieves reliability greater than 95% (Mahler *et al.*, 2019). Also, real robots can be used to collect training data (Levine *et al.*, 2017; Pinto and Gupta, 2016) but this approach is time-consuming and costly. The time needed to collect the data and the cost might be significantly reduced by using physics simulators (Johns *et al.*, 2016).

Our work is based on the Generative Grasping Convolutional Neural Network (GG-CNN) proposed by Morrison *et al.*, 2018). The Cornell dataset with real images annotated with grasp poses for a two-fingered gripper is used to train the neural network (Lenz *et al.*, 2015). In contrast to the Dex-Net (Mahler *et al.*, 2017), which requires sampling the objects, the GG-CNN computes a grasp quality factor, grasp angle and grasp width for each pixel of the input depth image. The GG-CNN is used to provide on-line feedback for the arm controller and can be used to grasp static and moving objects.

Grasping objects is much easier with mechanical feedback. Soft grippers adapt to the shape of the objects (Shepherd *et al.*, 2011; Deimel and Brock, 2014). This approach compensates the inaccuracies of the perception system and grasps planning methods. Adaptive grippers allow gentle manipulation of soft objects like fruits without destroying the objects (Abeach *et al.*, 2017). Also, feedback from the tactile sensors allows adapting the configuration of the gripper to the state and shape of the objects (Hogan *et al.*, 2018; Tian *et al.*, 2019). In this research, we use force feedback only during grasping objects. This approach is slower than mechanical feedback, but the grasping is more precise and the success depends mainly on the performance of the visual-based grasp planning investigated in this research.

The CNN-based methods for object detection are relatively new and developed independently. A set of state-of-the-art general object detection methods is available. Most accurate and efficient are YOLO (Redmon and Farhadi, 2020), SDD (Liu *et al.*, 2016) or Mask R-CNN (He *et al.*, 2017). Also, generic methods for object grasping are well studied. Among them we can find the GG-CNN (Morrison *et al.*, 2018), Dex-Net (Mahler *et al.*, 2019), multi-object grasp detection (Chu *et al.*, 2018), probabilistic-based methods (Kopicki *et al.*, 2015) or reinforcement learning-based methods (Levine *et al.*, 2017). The object detection and pose estimation methods are rarely developed together or integrated into a single system that allows grasping defined types of objects. First implementations use traditional techniques for object detection and grasping. In Wei and Chen (2020) the contours of objects are utilized to find them on the RGB image. Then, the detected contours are matched with the template model to estimate the pose of the object and determine the grasping points. The previous work by Ferran Rigual *et al.* investigates the problem of object detection for object manipulation (Rigual *et al.*, 2012). In this case, the method that uses local descriptors and matching the model to the current view (Collet *et al.*, 2009) is applied for object detection. Even though the object detection method also returns the pose of the object, the grasping point is determined using data from a depth camera and assuming that objects have a cuboid shape. This assumption is not always valid in a real-life scenario. The realistic application of the CNN-based Masked R-CNN method for object detection and pose estimation is presented by Shin *et al.* (2019). The gripper position is determined using obtained masks. The obtained success rate varies from 50 to 90%. However, little work has been done to integrate CNN-based object detection methods with CNN-based general grasping algorithms.

2.1 Approach and contribution

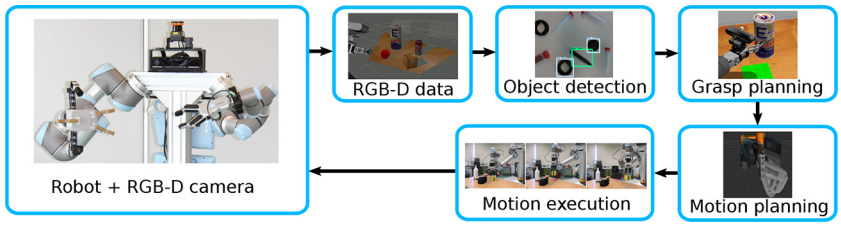
In this article, we investigate the role of the object detection and grasping modules in the perception system of a mobile-manipulating platform. Despite the number of general object detection methods, a little work is related to tasks when a robot has to grasp defined types of objects. This scenario is closer to industrial applications where the robot has to manipulate objects related to the given task. In this article, we integrated the category-independent GG-CNN method with a CNN-based object detector. The obtained system is used to detect and grasp objects defined by the user. For comparison, we implemented a method that determines the grasp pose of the gripper using operations on the point clouds. We compare both approaches to show the properties of the CNN and point cloud-based methods for grasping objects. We determined the advantages and disadvantages of both methods which are important for the autonomy of mobile manipulating industrial robots.

3. Grasping detected objects

3.1 System architecture

The proposed system's architecture is presented in Figure 2. The method is designed for our mobile manipulating platform, but it can be also used on the stationary robotics arm. Our robot is equipped with a mobile differential-drive base which allows the robot to move autonomously between workstations. We use the SLAMTEC Mapper 2D localization system (Nitta *et al.*, 2020) to localize the robot in the environment. SLAMTEC Mapper is more accurate and reliable than Hector SLAM (Nitta *et al.*, 2020). Our experience shows that it also works better than GMapping software (Grisetti *et al.*, 2007) running with Hokuyo UTM-30LX laser rangefinder. We also use Dynamic Window Approach motion planner (Fox *et al.*, 1997) for robot navigation. The robot is equipped with the Universal Robots UR5 arm and the OnRobot RG6 gripper for manipulating objects on the workstation. The robot is equipped with three RGB-D sensors. The first sensor (Kinect Xbox One) is mounted on the base of the robot and is used mainly for collision avoidance. The second sensor (Kinect Xbox One) is mounted on the head of the robot and is used to build a 3 D model of the scene. The third RGB-D camera (Intel RealSense D435) is mounted on the gripper and is used to

Figure 2 Architecture of the proposed controller of the robot



precisely measure the position of the objects on the workstation.

The calibration of the perception system is important in the presented system. We use the dedicated software provided by the vendor to perform the internal calibration of two Kinect Xbox One sensors. We also use this software to find the transformation between the RGB and depth camera. The transformation between the camera which is located on the gripper and the camera in the head of the robot can be found using easy hand-eye calibration software from Robot Operating System that implements the method presented in Tsai and Lenz (1989). However, we applied our implementation which uses artificial marker (checkerboard) and gradient-based optimization with the Adam algorithm (Kingma and Ba, 2015) to find the transformations between the cameras, the robot frame (base of the arm) and the frame associated with the localization system (SLAMTEC Mapper) (Piasek *et al.*, 2019).

The RGB-D data from the Intel RealSense D435 mounted on the gripper are used to detect objects and determine the grasping pose of the gripper. First, we use an object detection module to find objects on the table (Figure 2). Then, the grasp planning module determines the pose of the gripper which allows grasping the selected object. Finally, the motion planning module, which is based on the MoveIt! software, plans and executes the motion of the arm. The system is implemented in the Robot Operating System.

3.2 Objects detection

Having a precise knowledge about the objects' representation is a key element in the proposed pipeline. The objects visible in the scene have to be extracted from the background which constitutes redundant information. Therefore, a Single Shot Detector (SSD) based on the neural network Inception V2 architecture (Szegedy *et al.*, 2014) has been used to find bounding boxes of the objects that are required to perform the given task. The detector has been trained and adjusted to detect a set of articulated objects using weights pre-trained on the MS COCO dataset (Lin *et al.*, 2014).

In the experiment, there is an established set of objects that the module recognizes: stand type 1, stand type 2, ring, sumo figurine, plastic probe, tape, button, dispenser, sponge, plastic shaft. For that reason, we have collected a data set consisting of 1,500 samples and manually labeled the images in terms of object bounding boxes and object classes. The scenes containing objects have been diversified to allow more robust performance when it comes to object detection in various conditions.

Right upon RGB-D data acquisition the object detection module determines where the objects are located in the 2 D image. The next step is to pass the obtained color data to the object detection module and retrieve both bounding boxes and object classes present in the image. Ultimately, the constrained areas containing object representations are extracted from the whole captured images and passed to the grasp planning module.

3.3 Grasping with the point cloud

The point cloud obtained from a single perspective does not contain enough information about the shape of the object. The RGB-D camera acquires information about the front side of the objects and information about full 3D shape is missing. To obtain a reliable model of the objects, the robot performs the scanning procedure. The robot moves a camera to various predefined positions around the objects. In these positions, the input point clouds are accumulated to obtain a complete representation of objects on the table. The procedure is time-consuming but allows us to obtain a full model of the objects. The example configurations of the robot during scanning the objects on the table are presented in Figure 3.

Grasp planning method based on the point cloud requires processing the input data from the RGB-D camera. The goal of the proposed methods is to extract objects from the scene, detect instances of the objects, estimate their pose and define the pose of the gripper that allows grasping the selected object. We use the Point Cloud Library to process the point cloud. The results of the point cloud processing procedures are illustrated in Figure 4. In Figure 4(a) we show the raw point cloud data from the sensor. The first step of processing is cropping data to the Region of Interest (ROI) to remove points in the background [Figure 4(b)]. Then, the RANSAC method is applied for detecting the table plane [Figure 4(c)]. The RANSAC algorithm fits the plane model to the data to find the surface of the table. Finally, we extract points representing objects [Figure 4(d)] by removing the detected plane from the filtered input cloud.

In the next stage, the instances of objects are found in the point cloud. We use the Euclidean segmentation method to cluster the data and separate each object. The clustering

Figure 3 Example configurations of the robot during scanning the objects on the table

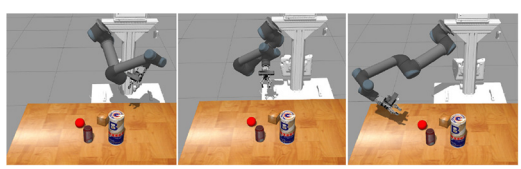
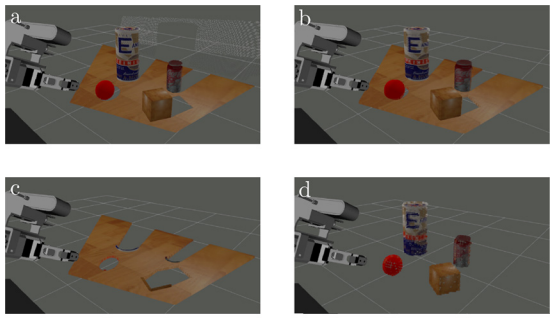


Figure 4 Input point cloud processing



Notes: (a) Input point cloud; (b) removed background; (c) the estimated surface of the table; (d) extracted objects

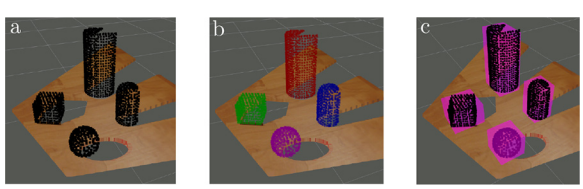
procedure uses the Euclidean distance as a threshold so the objects which are close to each other form the same cluster. Then, the robot estimates the pose of the objects. First, the centroids are calculated for each cluster (object). Then, the Oriented Bounding Box (OBB) fitting is performed. To obtain the best fit of the OBB, we use the Principal Component Analysis method (PCA), which finds three directions of objects point cloud with the greatest variance. These directions correspond to the edges of OBB. An example result of this process is presented in Figure 5.

The estimated poses of the bounding boxes give us information about the position and alignment of the object in the 3D space. This information is utilized for the grasp planning process. The centroid of the object is used as a gripper target position. The orientation of the gripper is determined depending on the dimensions of the bounding box. If the height of the box is smaller than the width and length of the box, the robot tries to grasp the object from the top. If the height of the box is larger than the remaining dimension, the reference orientation of the gripper is horizontal. This approach stabilizes the grasp and increases the success rate of the method in experiments on the real robot.

3.4 Grasping with convolutional neural network

As a representative object-independent grasp synthesis method, we have chosen the Generative GG-CNN (Morrison *et al.*, 2018). In contrast to the previous *state of the art* CNN-based grasping methods, the GG-CNN predicts grasp quality and gripper configuration for each pixel. This approach allows for avoiding time-consuming sampling of the input space (image). Moreover, the GG-CNN is fast and can be used to re-compute the gripper configuration during the motion of the robot and grasp moving objects (Morrison *et al.*, 2018).

Figure 5 Results of the (b) segmentation and (c) bounding box fitting procedures for the (a) example input point cloud

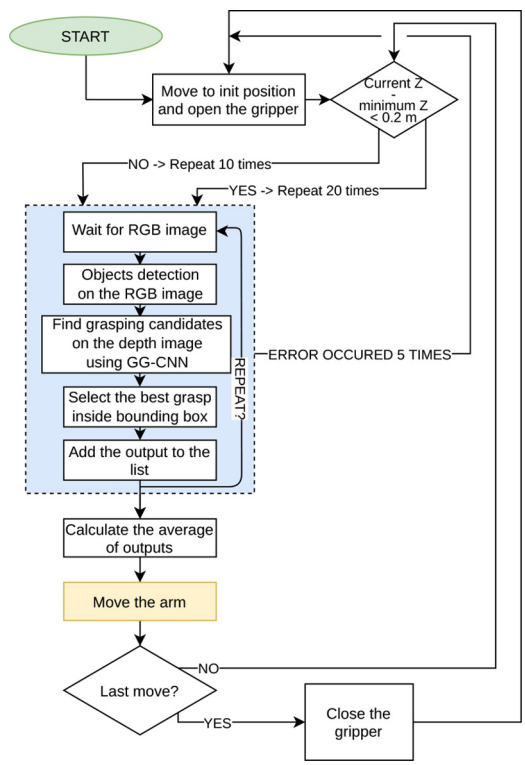


In this research, we modified the original software so it can work with the Universal Robots UR5 arm and the Intel RealSense D435 RGB-D camera. We use the information about the gripper's opening (gripper width) at the end of the sequence only to avoid its unnecessary motions. Instead, we open the gripper at the beginning of the procedure and we close the gripper at the end of the motion. While closing the gripper, we use force feedback. The gripper stops the motion if the reaction force is larger than the threshold. This strategy compensates for the inaccuracies of the perception system, stabilizes the grasp and increases the success rate of the grasping method.

On top of the GG-CNN and arm controller, we have built a procedure, which closes the loop from the perception systems and allows grasping the selected object. This procedure is presented in Figure 6. In the beginning, the robot moves to the initial configuration. In this configuration, the camera is tilted down and observes the environment in front of the robot. Because the GG-CNN is designed to work in three-dimensional space (GG returns horizontal motion of the gripper and planar rotation) the controller keeps the gripper and the camera tilted down during the execution of the reference motion. The motion of the gripper is limited by the predicted height of the table to prevent damaging the gripper caused by an inaccurate output from the neural network.

We divided the behavior of the procedure according to the distance to the grasping object. If the camera and the gripper are far from the object, we repeat the object detection and grasping methods 10 times and we compute the average output from the GG-CNN. Our experience from the experiments

Figure 6 Block diagram visualizing the grasping procedure with the GG-CNN method



shows that this approach stabilizes the results from the module and increases the success rate. The robot stops the procedure and moves to the initial configuration if the object is not detected at least five times in a sequence of 10 measurements. The controller works in the loop and executes the part of the reference motion. Then, the perception procedure is repeated. If the camera is closer to the object and the distance to the table is less than 20 cm, we repeat the perception procedure 20 times to increase the accuracy. Then, we execute the robot's last motion to the goal position. The threshold set to 20 cm comes from the camera field of view and the minimal range of the sensor (0.1 m).

3.5 Integration of the generative grasping convolutional neural network with a convolutional neural network-based object detector

The crucial modification of the GG-CNN is presented in the blue box in Figure 6. The current image from the camera is provided to the input of the Single Shot Detector to find the objects on the image. Independently, the depth image is provided to the input of the GG-CNN. The GG-CNN is general and provides grasp candidates for all objects on the scene. To grasp the selected object, we apply the mask (bounding box for the selected object) from the SSD to the output from the GG-CNN to find the best grasp candidate for the object.

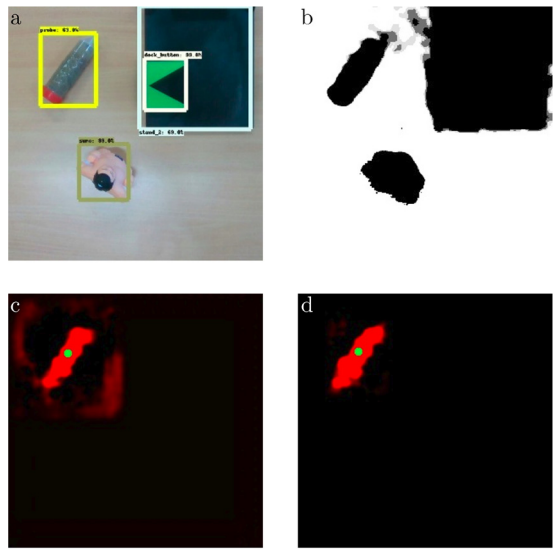
Alternatively, we verified the second method of integrating results from object detection with the GG-CNN method. In the second approach, the object interesting for the given task is cropped from the depth image. The remaining pixels in the depth image are set to 0. The modified depth image is provided as an input of the GG-CNN to find the best candidate grasp. The example result for this method is presented in Figure 7(c). For the comparison, the output from the GG-CNN obtained with the method used in this research is presented in Figure 7(d). The best grasping points (green points in the image) are almost in the same place for both cases. However, when the bounding box from the object detector is used to crop the depth image, the GG-CNN returns candidate grasp on the edges of the bounding box. The cropping operation on the depth image introduces edges which might cause improper behavior of the method. Thus, in the experiments on the real robot, we run GG-CNN on the whole depth image and select the best grasping point inside the bounding box given by the SSD.

The example sequence of grasping an object is presented in Figure 8. First, the camera observes the scene from the initial configuration of the robot. Then, the robot gradually executes the planned path and repeats the perception procedures (SSD and GG-CNN). When the distance between the table and the camera is smaller than 20 cm, the robot executes the final trajectory to the goal position given by the GG-CNN and closes the gripper. In the visualization presented in Figure 8 the robot moves to the initial configuration after a successful grasp.

4. Results

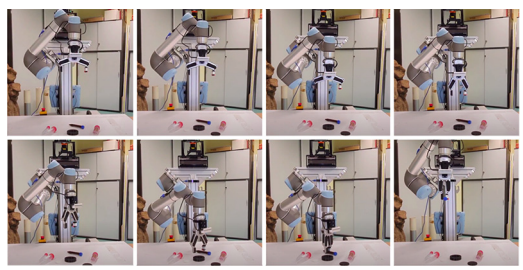
In the first experiment, we verified the accuracy of the procedure which estimates the position and dimensions of the objects on the table [1]. During experiments on the real robot, the ground truth position of the objects is unknown so we

Figure 7 Example output for two methods of integrating object detection results with the GG-CNN



Notes: (a) Input RGB image with detected objects; (b) depth image; (c) results from the GG-CNN when the mask is applied on the depth image; (d) mask applied on the output from the GG-CNN

Figure 8 Example sequence of grasping the defined object using GG-CNN



obtained quantitative results in the Gazebo simulator. In the experiment, the four objects (a ball, a can, a box and a wooden box) are randomly located on the table and the proposed method is used to estimate their position and dimensions. The results are presented in Table 1. The experiment is performed 20 times and the mean error values and standard deviation are computed. The average position error for all objects is smaller than 6 mm. The average dimension error of the objects is smaller than 9 mm. The standard deviations for both values

Table 1 Results of the point cloud-based object pose and dimensions estimation: average MSE for the position e_p and dimensions e_d and corresponding standard deviations σ_p and σ_d

Parameter	Ball	Can	Box	Wooden box	Average
e_p [m]	0.0062	0.0067	0.0055	0.0046	0.0057
σ_p [m]	0.0009	0.0024	0.0010	0.0008	0.0013
e_d [m]	0.0055	0.0029	0.0069	0.0198	0.0088
σ_d [m]	0.006	0.0012	0.0037	0.0105	0.0040

remain small (below 2 mm for the position of the objects and 4 mm for the dimensions of the objects).

The results presented in Table 1 are obtained in the simulation environment with the perfect depth sensor. The errors result from the geometrical properties of the perception system and the incomplete model of the objects obtained after the scene scanning phase. We expect that these error values are larger if the real RGB-D camera is used. Moreover, the position and dimension errors accumulate and might cause collisions during grasp execution. We deal with this problem during the experiments on the real robot by increasing the opening of the gripper and using force feedback while closing the gripper. By this relatively simple strategy, we deal with small inaccuracies of the estimated pose and dimensions of the objects.

Subsequently, we performed the same experiment on the real robot with data from the Intel RealSense D435. The example results are presented in Figure 9. Because the objects' real poses are unknown, we evaluate results by using the whole pipeline for perception and grasping. We performed 46 trials. The goal was to detect the object given by the operator, grasp the object, and place the object in the given position. The average success rate for all objects is 73.91% taking into account the whole task. However, the average success rate for grasping the object is 93.88%. The success rate for the yellow foam and black case is 100% and drops to 81.82% for the sumo figurine. This is caused mainly by the dimensions of this object. The sumo figurine is narrow on the top and the gripper misses the object. In this case, the sensor error and pose estimation error accumulate and cause a smaller success rate.

In some trials, the robot failed because the robot hit the table despite the fact that we use OctoMap (Hornung *et al.*, 2013) to detect collisions between the robot and the environment. In some cases, the trajectory planning using MoveIt for the robot failed and as a result, the robot performs strange motions of the arm and collides with the objects in the environment. The example execution of the motion of the robot is presented in Figure 10. Another interesting property of the method which comes from these experiments is the way the robot grasps the objects. In Figure 9, two example bottle poses are presented. When the bottle pose is horizontal the robot grasps these objects from the top. If the pose of the bottle is vertical, the reference pose of the gripper is horizontal.

In the course of the data collection process, we have selected a set of data to evaluate the efficacy of the detection system. Testing data have not been used in the training phase allowing us to validate the reliability of the detection module as a part of a robotic system. We follow the standard methodology of calculating Average Precision and mean Average Precision,

Figure 9 Example bounding boxes found during experiments on the real robot

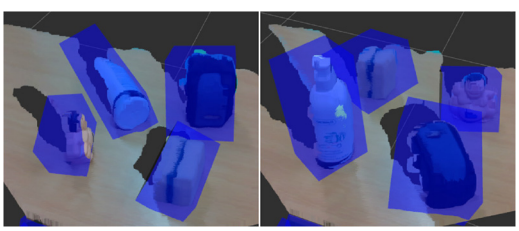
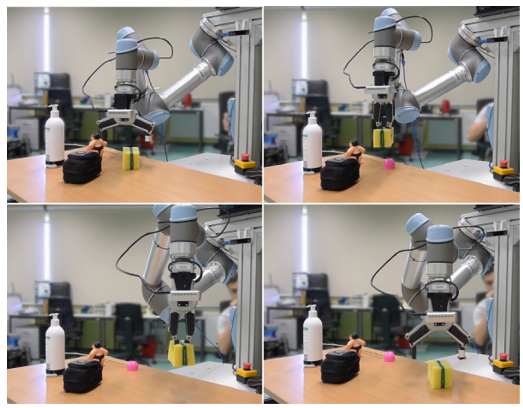


Figure 10 Robot grasping and moving the object detected using the point cloud-based method



widely described as AP and mAP, respectively. We calculate the aforementioned metrics based on 282 test images including 14 classes. The Average Precision metric is strongly dependent on the type of object. The more explicit and distinct the object is in terms of color or shape, the more precisely it can be detected by the neural network. It is clearly visible in Figure 11, where objects with characteristic features achieve a better AP score. On the other hand, the objects that do not have distinct features (white dispenser, transparent probe) turn out to be detected properly in less number of cases. In some of the cases, we might face a false positive detection, which occurs mostly when the objects are similar to each other or to the environment. Hence, the classes stand_1, stand_2, and probe happen to have more incorrect detections, as it can be observed in Figure 12. The achieved mAP of 77.96% satisfies the need in our use case, where we can carry out the detection based on multiple frames and make sure that no object is missed in the grasping phase.

In the first experiment with the GG-CNN, the goal of the robot was to collect the objects from the table and put them in the box. The example configurations of the objects are presented in Figure 13. For the configuration of the objects presented in Figure 13(a) the robot collected all objects in 7

Figure 11 AP and mAP score across 14 classes based on the test data set

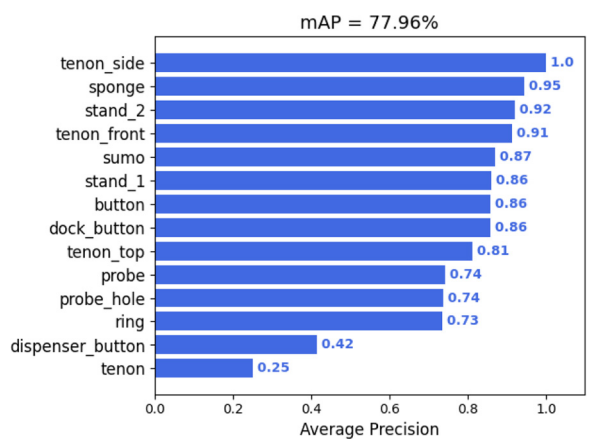


Figure 12 Results of the detection carried out against the collected test dataset

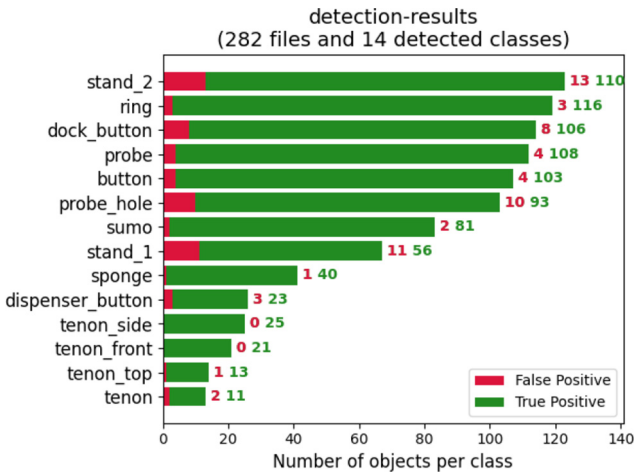
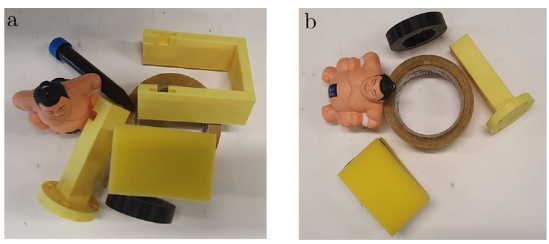


Figure 13 Example configurations of the objects during the "cleaning the table" task

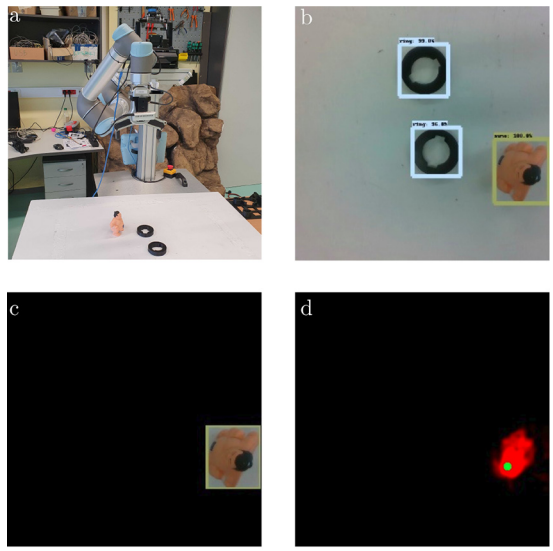


trials. It means that the GG-CNN failed in three attempts. For the configuration presented in Figure 13(b) all grasping trials were successful. We conclude that with the GG-CNN, the robot can clean the table successfully without the knowledge about the objects' categories. In contrast to the point cloud-based method, the objects can occlude each other. However, raw GG-CNN cannot be applied to the tasks where the category of objects is important.

In the next experiment, we show the output from the modules used during grasping objects with the GG-CNN. In Figure 14(a), we show the experimental set. The objects are on the table and the camera is facing down. The objects detected by the SSD are presented in Figure 14(b). Then, the proposed method crops and selects the object from the image [Figure 14(c)]. The output from the GG-CNN is presented in Figure 14(d). The grasping point is indicated by the green dot. In this scenario, the robot follows the defined objects and grasps the objects according to the output provided by the GG-CNN.

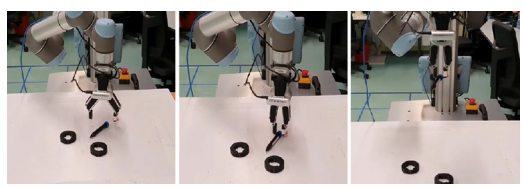
The example execution of the grasp planned with the use of GG-CNN is presented in Figure 15. The robot repeats the object detection and the grasp planning in the loop until it reaches the minimal distance depending on the sensor range and at the end grasps the object. Finally, the controller adjusts the gripper's orientation according to the orientation of the selected object (tube in Figure 15). In contrast to the original GG-CNN, the algorithm ignores the remaining objects on the scene.

Figure 14 Example output from the modules used while grasping objects using GG-CNN



Notes: (a) Experimental set; (b) object detection results; (c) object cropped from the RGB image; (d) output from the GG-CNN

Figure 15 Experiment with the GG-CNN: the robot detects the tube and successfully grasps the object



In the last experiment, we compare the point cloud and GG-CNN-based approaches to grasping. The algorithms are used to grasp the same set of objects. The selected objects are presented in Figure 16. In the experiment, the robot tries to grasp the objects (single object on the table for each trial) using defined methods. The results are presented in Table 2. The compared methods perform the best for the box-like object, so we selected objects which are problematic for both methods. The CNN-based and point cloud-based methods have a similar success rate. The point cloud-based method performs better for the dispenser bottle. The GG-CNN tries to grasp this object always from the top which is difficult if the bottle is standing. In contrast, the point cloud-based method adapts the orientation of the gripper depending on the orientation of objects. The black ring is also problematic for the GG-CNN. The neural network prefers points that are on the edges of the object. As a result, the GG-CNN returns the grasping point on the edge of the ring. In the successful scenario, one finger of the gripper is outside the ring and the second finger is in the center of the ring. Unfortunately, the opening of the gripper is very often inaccurate. The error related to the grasping point and the error related to the opening of the gripper accumulate. In this case, the finger which is inside the ring causes collisions and the grasp

Figure 16 The objects used to compare point cloud-based method with the GG-CNN: sponge, dispenser bottle, sumo figurine, tape, plastic probe, tape, ring, I-shape part and plastic shaft

attempt fails. On the other hand, the point cloud-based method fails when the ring is horizontal. The height of the ring is within the noise range of the sensor and the object is hardly visible on the depth image. The plane segmentation in most cases removes this object from the scene which results in only one successful grasp from 20 trials.

We also analyze the execution time for both methods. The average segmentation time for the point cloud-based method is 74.5 ms. The average value is obtained for 1,000 measurements. The time required for generating bounding boxes depends on the number of objects in the scene. The average time needed to generate bounding boxes is 38.7 ms when four objects are observed by the robot. This time decreases to 22.9 ms when two objects only are detected on the scene. Finally, the computation of the reference gripper pose takes an average of 50 ms. However, the point cloud-based method suffers from the scene scanning procedure. The scanning procedure varies from 95 to 105 s and 79 to 82 s when the speed of the robot is set to 35 and 50% of the maximal speed, respectively. All these procedures are performed in a single step when GG-CNN is applied. The GG-CNN does not require environment scanning or scene segmentation. The whole perception, which computes the grasping point, takes 32.8 ms only for the GG-CNN (inference takes 7.6 ms, and searching for the maximum inside the bounding box takes 25.2 ms on average). For both methods, the execution time takes less than 30 s, but we do not use the full speed of the robot. Our robot works in the fully autonomous mode in the experiments presented in the article and for safety reasons (of the robot and people in the environment), we reduce speed to 35–50% of the maximal available speed.

5. Conclusions

In this article, we focus on the RGB-D based grasp planning for robotic arms. We used two category-free grasping methods which are based on different principles. The first method uses a 3 D point cloud to extract objects from the background, estimate the poses of the objects and determine the pose of the gripper. The second method based on the GG-CNN uses depth image only to find the optimal grasping point and configuration of the gripper. We integrated the grasping methods with the CNN-based object detection method to obtain a system for grasping defined objects. We closed the loop from the perception system of the robot and performed a comparative study to investigate the properties of both methods. We determine the benefits and drawbacks of the obtained systems.

The properties of the point cloud-based method for grasping the objects depend on the assumptions made during implementation. The segmentation of the objects depends on the Euclidean distance so the objects which are close to each other create a single segment. As a result, the robot tries to grasp a few objects at once. Also, small objects can't be grasped by these methods. This comes from the fact that the depth measurements are noisy. If the dimensions of the objects are comparable with the sensor noise, they are removed from the scene during segmentation. Despite these drawbacks, the implemented method is efficient and the robot can deal with grasping various objects. Moreover, thanks to the visual feedback, even if a single grasp fails, the robot repeats the procedure until the objects are successfully moved to the goal position.

The CNN-based method is much faster than the method that the point cloud-based method. This comes from the fact that the method does not segment the scene directly nor extracts objects. The GG-CNN looks for all points on the scene which are good grasp candidates. We limit this behavior by integrating GG-CNN with the object detection method and looking for grasping points inside the bounding box related to the found object. The main drawback of the GG-CNN is related to the produced output. The method looks for the good local grasping points and ignores the whole shape of the object.

Table 2 Comparison of grasping efficiency for the point cloud-based and GG-CNN methods

	PCL			GG-CNN		
Object name	Successful attempts	Total attempts	Efficiency [%]	Successful attempts	Total attempts	Efficiency [%]
Sponge	17	20	85	16	20	80
Dispenser bottle	13	15	86.7	6	15	40
Sumo	14	20	70	16	20	80
Tape	15	20	75	18	20	90
Ring (vertical)	16	20	80	14	20	70
Ring (horizontal)	1	20	5	8	20	40
I-shape part	12	20	60	15	20	75
Plastic shaft	2	20	10	8	20	40
ALL	90	155	**58.1**	101	155	**65.2**

This situation is well visible during experiments with the rings and results in many failures grasps. However, similarly to the point cloud-based method, collecting the objects from the table is performed with a reasonable success rate due to the feedback and capability of the system to repeat the failure trial.

In this article, we focus on the integration of grasping and object detection methods. The main contribution of this work lies in:

- Integration of the category-free grasping methods with CNN-based object detection – the new system allows grasping objects detected on the scene depending on the task.
- Implementation of a reference method that defines the grasping position of the gripper using operations on the point cloud – the method also extracts objects from the scene and determines the pose of the objects on the scene. The implemented method is used as a reference for the proposed CNN-based grasping and object detection method and used for comparison.
- Implementation of the perception, grasping and motion execution methods on the real robot to show the properties of the system in various scenarios.
- Comparison of the grasping methods in the experiments on the mobile manipulating robot.

In the future, we are going to work on the identified limitations of the presented methods. The point cloud-based grasping methods require careful scanning of the environment to build the model of the scene. We are going to develop methods that reconstruct the scene from a single camera image (Piaskowski *et al.*, 2019). We are also going to use the information about the object category to improve the CNN-based grasping method.

Note

1 Short video from experiments is available at https://youtu.be/hkBHKmyCXqw

References

Abeach, L.A.T.A., Nefti-Meziani, S. and Davis, S. (2017), "Design of a variable stiffness soft dexterous gripper", *Soft Robotics*, Vol. 4 No. 3.

Andersen, R.S., Damgaard, J.S., Madsen, O. and Moeslund, T.B. (2013), "Fast calibration of industrial mobile robots to workstations using qr codes", *Ieee Isr 2013*, pp. 1-6.

Berns, K., Asfour, T. and Dillmann, R. (2000), "Design and control of the humanoid robot armar", in Morecki, A., Bianchi, G. and Rzymkowski, C. (Eds), *Romansy 13*, Springer Vienna, Vienna, pp. 307-312.

Bohren, J., Rusu, R.B., Gil Jones, E., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mösenlechner, L., Meeussen, W. and Holzer, S. (2011), "Towards autonomous robotic butlers: lessons learned with the pr2", *2011 IEEE International Conference on Robotics and Automation*, pp. 5568-5575.

Chu, F., Xu, R. and Vela, P.A. (2018), "Real-world multiobject, multigrasp detection", *IEEE Robotics and Automation Letters*, Vol. 3 No. 4, pp. 3355-3362.

Collet, A., Berenson, D., Srinivasa, S.S. and Ferguson, D. (2009), "Object recognition and full pose registration from a single image for robotic manipulation", *2009 IEEE International Conference on Robotics and Automation*, pp. 48-55.

Deimel, R. and Brock, O. (2014), "A novel type of compliant, underactuated robotic hand for dexterous grasping", *Proceedings of Robotics: Science and Systems*, Berkeley.

Dietrich, A., Bussmann, K., Petit, F., Kotyczka, P., Ott, C., Lohmann, B.L. and Albu-Schäffer, A. (2016), "Whole-body impedance control of wheeled mobile manipulators", *Autonomous Robots*, Vol. 40 No. 3, pp. 505-517.

Dömel, A., Kriegel, S., Kaßecker, M., Brucker, M., Bodenmüller, T. and Suppa, M. (2017), "Toward fully autonomous mobile manipulation for industrial environments", *International Journal of Advanced Robotic Systems*, Vol. 14 No. 4, pp. 1-19.

Droeschel, D., Schwarz, M. and Behnke, S. (2017), "Continuous mapping and localization for autonomous navigation in rough terrain using a 3d laser scanner", *Robotics and Autonomous Systems*, Vol. 88, pp. 104-115.

Fischinger, D. and Vincze, M. (2012), "Empty the basket – a shape based learning approach for grasping piles of unknown objects", *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE*, pp. 2051-2057.

Fox, D., Burgard, W. and Thrun, S. (1997), "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine*, Vol. 4 No. 1, pp. 23-33.

Gon Roh, S., Park, J.H., Lee, Y.H., Song, Y.K., Yang, K.W., Choi, M., Kim, H.-S., Lee, H. and Choi, H.R. (2008), "Flexible docking mechanism with error-compensation capability for auto recharging system of mobile robot", *International Journal of Control, Automation, and Systems*, Vol. 6 No. 5, pp. 731-739.

Grisetti, G., Stachniss, C. and Burgard, W. (2007), "Improved techniques for grid mapping with rao-blackwellized particle filters", *IEEE Transactions on Robotics*, Vol. 23 No. 1, pp. 34-46.

Hang, K., Stork, J., Pollard, N. and Kragic, D. (2017), "A framework for optimal grasp contact planning", *IEEE Robotics and Automation Letters*, Vol. 2 No. 2, pp. 704-711.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017), "Mask r-cnn", *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980-2988.

Hogan, F.R., Bauza, M., Canal, O., Donlon, E. and Rodriguez, A. (2018), "Tactile regrasp: grasp adjustments via simulated tactile transformations", *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2963-2970.

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C. and Burgard, W. (2013), "OctoMap: an efficient probabilistic 3D mapping framework based on octrees", *Autonomous Robots*, Vol. 34 No. 3, pp. 189-206.

Huang, Q., Tanie, K. and Sugano, S. (2000), "Coordinated motion planning for a mobile manipulator considering stability and manipulation", *The International Journal of Robotics Research*, Vol. 19 No. 8, pp. 732-742.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. and Murphy, K. (2017), "Speed/accuracy trade-offs for modern convolutional object detectors", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296-3297.

Hvilshøj, M., Bøgh, S., Madsen, O. and Kristiansen, M. (2009), "The mobile robot 'little helper': concepts, ideas and working principles", *2009 IEEE Conference on Emerging Technologies Factory Automation*, pp. 1-4.

Jain, S. and Argall, B. (2016), "Grasp detection for assistive robotic manipulation", *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2015-2021.

Johns, E., Leutenegger, S. and Davison, A. (2016), "Deep learning a grasp function for grasping under gripper pose uncertainty", *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE*, pp. 4461-4468.

Kingma, D.P. and Ba, J. (2015), "Adam: a method for stochastic optimization", in Bengio, Y. and LeCun, Y. (Eds), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA*, May 7-9, 2015, Conference Track Proceedings.

Klamt, T., Schwarz, M., Lenz, C., Baccelliere, L., Buongiorno, D., Cichon, T., DiGuardo, A., Droeschel, D., Gabardi, M., Kamedula, M., Kashiri, N., Laurenzi, A., Leonardis, D., Muratore, L., Pavlichenko, D., Periyasamy, A.S., Rodriguez, D., Solazzi, M., Frisoli, A., Gustmann, M., Roßmann, J., Süss, U., Tsagarakis, N.G. and Behnke, S. (2020), "Remote mobile manipulation with the centauro robot: full-body telepresence and autonomous operator assistance", *Journal of Field Robotics*, Vol. 37 No. 5, pp. 889-919.

Kootstra, G. (2012), "Enabling grasping of unknown objects through a synergistic use of edge and surface information", *The International Journal of Robotics Research*, Vol. 34, pp. 26-42, doi: 10.1177/0278364915594244.

Kopicki, M.S., Belter, D. and Wyatt, J.L. (2019), "Learning better generative models for dexterous, single-view grasping of novel objects", *The International Journal of Robotics Research*, Vol. 38 Nos 10/11, pp. 1246-1267.

Kopicki, M., Detry, R., Adjigble, M., Stolkin, R., Leonardis, A. and Wyatt, J.L. (2015), "One shot learning and generation of dexterous grasps for novel objects", *The International Journal of Robotics Research*, Vol. 35 No. 8, doi: 10.1177/0278364915594244.

Lenz, I., Lee, H. and Saxena, A. (2015), "Deep learning for detecting robotic grasps", *The International Journal of Robotics Research*, Vol. 34 Nos 4/5, pp. 705-724.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J. and Quillen, D. (2017), "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection", *International Journal of Robotics Research*, doi: 10.1177/0278364917710318.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. (2014), "Microsoft coco: common objects in context", European conference on computer vision, Springer, pp. 740-755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A.C. (2016), "Ssd: single shot multibox detector", in: Leibe, B., Matas, J., Sebe, N. and Welling, M. (Eds), *Computer Vision – ECCV 2016*, Springer International Publishing, Cham, pp. 21-37.

Liu, S. and Carpin, S. (2015), "Global grasp planning using triangular meshes", *IEEE International Conference on Robotics and Automation*, IEEE, pp. 4904-4910.

Mahler, J., Matl, M., Satish, V., Danielczuk, M., DeRose, B., McKinley, S. and Goldberg, K. (2019), "Learning ambidextrous robot grasping policies", *Science Robotics*, Vol. 4 No. 26, p. eaau4984.

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Aparicio, J. and Goldberg, K. (2017), "Dex-net 2.0: deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics", Robotics Science and Systems (RSS).

Morrison, D., Corke, P. and Leitner, J. (2018), "Closing the loop for robotic grasping: a real-time, generative grasp synthesis approach", *Proc. of Robotics: Science and Systems (RSS)*.

Nitta, Y., Yenet Bogale, D., Kuba, Y. and Tian, Z. (2020), "Evaluating slam 2d and 3d mappings of indoor structures", in Osumi, H. , Furuya, H. and Tateyama, K. (Eds), *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC), International Association for Automation and Robotics in Construction (IAARC)*, Kitakyushu, pp. 821-828.

Piasek, J., Staszak, R., Piaskowski, K. and Belter, D. (2019), "Multi-sensor extrinsic calibration with the adam optimizer", 12th International Workshop on Robot Motion and Control (RoMoCo), pp. 209-214.

Piaskowski, K., Staszak, R. and Belter, D. (2019), "Generate what you can't see – a view-dependent image generation", *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5903-5909.

Pinto, L. and Gupta, A. (2016), "Supersizing self-supervision: learning to grasp from 50k tries and 700 robot hours", *IEEE International Conference on Robotics and Automation', IEEE*, pp. 3406-3412.

Redmon, J. and Farhadi, A. (2020), "A mobile manipulation system for one-shot teaching of complex tasks in homes", ArXiv arXiv:1804.02767.

Rigual, F., Ramisa, A., Alenyà, G. and Torras, C. (2012), *Object Detection Methods for Robot Grasping: Experimental Assessment and Tuning*, CCIA.

Satish, V., Mahler, J. and Goldberg, K. (2019), "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks", *IEEE Robotics and Automation Letters*, Vol. 4 No. 2, pp. 1357-1364.

Shepherd, R., Ilievski, F., Choi, W., Morin, S., Stokes, A., Mazzeo, A.D., Chen, X., Wang, M. and Whitesides, G. (2011), "Multigait soft robot", *Proceedings of the National Academy of Sciences of Sciences*, Vol. 108 No. 51, pp. 20400-20403.

Shin, H., Hwang, H., Yoon, H. and Lee, S. (2019), "Integration of deep learning-based object recognition and robot manipulator for grasping objects", *2019 16th International Conference on Ubiquitous Robots (UR)*, pp. 174-178.

Song, D., Ek, C., Huebner, K. and Kragic, D. (2015), "Task-based robot grasp planning using probabilistic inference", *IEEE Transactions on Robotics*, Vol. 31 No. 3, pp. 546-561.

Stückler, J., Schwarz, M. and Behnke, S. (2016), "Mobile manipulation, tool use, and intuitive interaction for cognitive service robot cosero", *Frontiers in Robotics and AI*, Vol. 3, pp. 1-20.

Szegedy, C., Reed, S., Erhan, D., Anguelov, D. and Ioffe, S. (2014), "Scalable, high-quality object detection", arXiv preprint arXiv:1412.1441.

ten Pas, A. and Platt, R. (2014), "Localizing handle-like grasp affordances in 3d point clouds", *International Symposium on Experimental Robotics*.

Tian, S., Ebert, F., Jayaraman, D., Mudigonda, M., Finn, C., Calandra, R. and Levine, S. (2019), "Manipulation by feel: touch-based control with deep predictive models", *IEEE International Conference on Robotics and Automation*, pp. 818-824.

Tsai, R.Y. and Lenz, R.K. (1989), "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration", *IEEE Transactions on Robotics and Automation*, Vol. 5 No. 3, pp. 345-358.

Wei, A.H. and Chen, B.Y. (2020), "Robotic object recognition and grasping with a natural background", *International Journal of Advanced Robotic Systems*, Vol. 17 No. 2, doi: 1729881420921102.

Xiang, Y., Schmidt, T., Narayanan, V. and Fox, D. (2018), "Posecnn: a convolutional neural network for 6D object pose estimation in cluttered scenes".

Corresponding author

Dominik Belter can be contacted at: dominik.belter@put.poznan.pl