10.24425/acs.2025.153959

Archives of Control Sciences
Volume 35(LXXI), 2025
No. 1*, pages 89–121*

Estimation of grasp type and determining grasping points for Adaptive Grasping Device

Michał CZUBENKO , Piotr ŁYCZKO and Rafał OKUŃSKI

The article describes the grasping point algorithm (GPA), aimed to determine points for picking up objects by a manipulator equipped with a proprietary gripper enabling three picking modes (suction cup, two fingers, three fingers). The paper outlines the topic of unknown object picking using various types of grippers, both from a stack of objects and stand-alone items. It describes details of four methods available in the literature. Presented GPA algorithm consists of four separate functions dedicated to appropriate grasping methods chosen based on a decision-making process. The dedicated functions has been described in detail and tested. Used methods have been tested manually on 15 different objects. GPA algorithm has been evaluated in real-world tests as well. The presented tests showed efficiency of approximately 92% in the case of moving previously unknown objects from the stack to the target cuvette.

Key words: pick-and-place operation; Industry 5.0; adaptive gripper; grip points estimation; grasping robot.

M. Czubenko (corresponding author, e-mail: micczube@pg.edu.pl) and P. Łyczko are with Intema Sp. z o.o., Gdańsk, Poland and Department of Robotics and Decision Systems, Faculty of Electronics Telecommunications and Informatics, Gdańsk University of Technology, Gdańsk, Poland.

R. Okuński is with Intema Sp. z o.o., Gdańsk, Poland.

The project was co-founded by the National Centre for Research and Development (POIR.01.01.01-00-1685/20).

M. Czubenko: Conceptualization, Supervision, Methodology, Validation, Writing-Reviewing and Editing; P. Łyczko: Software, Data curation, Writing-Original draft preparation, Investigation; R. Okuński: Software, Data curation, Writing-Original draft correction, Investigation.

The project was co-funded by the European Union from the European Regional Development Fund as part of the Smart Growth Program. The project is implemented as part of the National Centre for Research and Development (NCBR): Fast Track, activities: R+D projects for enterprises, sub-activity: Industrial research and development works carried out by enterprises; POIR.01.01.01-00-1685/20.

Received 12.08.2024. Revised 17.01.2025.

1. Introduction

In 2022, the number of annual robot installations reached 553,000 units [45]. The leading countries include China, Japan, the United States, the Republic of Korea, and Germany. These countries are responsible for approximately 80% of the installations. The observed growth is related to ongoing technological development, in particular, both the implementation of cobot (cooperative robot) technology and aspects of Industry 4.0 (basically, digitalization and network integration) [17, 42]. Note that, in the future, we may expect robotic implementations of AI systems (like in the Robotics Foundation Model by Covariant) and (far in the future) robots equipped with cognitive aspects [18, 27, 33, 34]. In general, the main reason for such rapid robotization is to increase productivity while saving costs [29], mainly through faster and more precise manipulation of objects.

Note that, the industrial definition of a robot, specified by ISO 8373:2012 norm, shows that a robot is ‘a multifunctional, reprogrammable, automatically controlled manipulator, programmable (. . .) for use in industrial automation applications’. From the above definition, it follows that a robot is mainly designed to manipulate objects. However, the stand-alone robot itself is nothing without a specified tool attached to its flange.

The tools of the production robots play a crucial role in enabling their interaction with the environment and performing various tasks. Most robots may be adapted to a certain task by changing a tool and reprogramming their trajectories. Among the most common tools, we can distinguish grippers (the most numerous group), welding, painting, and other special-purpose tools. Since the main purpose of the industrial robot is the manipulation of objects, the grippers constitute a fundamental element. Thus, in parallel with the robotization of enterprises, methods of grasping objects have also been developed.

In general, grippers, based on the type of origin, can be distinguished as anthropomimetic, biomimetic, and other (not biomimetic nor anthropomimetic) [6]. On the other hand, they can also be categorized by the gripping method: mechanical, magnetic, vacuum, with elastic chambers, air jet; or by drive type: electric, hydraulic, pneumatic, magnetic, drive-less [25]. The grippers may also be diversified by the finger types as soft or hard. However, the most commonly known classification is based on the gripper configuration, it includes [51]:

- one finger (a vacuum-based sucker),
- two fingers (most common one),
- three fingers (with different configurations),
- soft, flexible fingers,
- adaptive multi-fingers,
- grain-filled ball grippers.

Among many grippers manufacturers, the key players should be distinguished: SCHUNK, Thinkbot Solutions, KCC, OnRobot, SMC, DH-Robotics, and Festo Corporation [24]. The source shows that the 2-finger grippers are the most used on the market (about 43%), followed by 3-fingers (35%), and 4-fingers (22%). However, it does not include single or multiple vacuum grippers. The robotic grippers market can be estimated at 1.7 billion USD in 2023, and it should reach 4.3 billion by 2033 [30].

A statistical review ‘of pneumatic, parallel, two-finger, industrial grippers’ can be found in [4]. The paper shows that most of the 2-fingers grippers on the market have a similar characteristic – limited force and small stroke. Another broad review of the state of the art of robotic grippers and their applications can be found in [54]. From an agricultural point of view, a review of gripers and methods of object grasping is presented in [61]. On the other hand, there are plenty of scientific works about soft grippers. Their review can be found in [23]. Note that the soft grippers are quite adaptive, and they can be used in grasping different-shaped objects.

Despite the broad range of gripper types, there is no universal one [3]. Designing a universal grasping system capable of adapting to different objects poses a challenge from both a technical and algorithmic perspective [22].

From a mechanical point of view, several attempts have been made. Namely, [31] designed a two-finger gripper equipped with a sliding suction mechanism. Another approach, presented in [37], is based on three reconfigurable fingers that can be clasped in two different methods. A similar approach can be found in [32]. There also exists a four-finger gripper with reconfigurable fingers [40]. The authors show that their gripper can perform five types of grasping, namely: caging, parallel pinch, thumb-three finger, clasped, and t-shape grasping. Another modern approach to the design of grippers is presented in [38], where the authors show the concept of anthropomimetic fingers. Another gripper with adaptive fingers can be found in [15], where the authors show different types of two-fingers grasping, and compare them to available grippers.

Currently, there are also a few adaptive, reconfigurable grippers on the global market. These are mainly based on similar concepts (three fingers, their rotation, and two types of grasping). Among them, we can list products of the Robotiq, OnRobot, Barret, and Neobotix companies, and the open project of the T42 gripper.

Based on the websites, articles, and additional materials such as videos, both groups of adaptive grippers perform quite well when grasping a single object. A key issue for robust grasping is therefore an algorithm capable of analyzing the physical properties of an object to effectively determine the appropriate grasping technique.

1.1. Motivation and background

The primary objective of our study is to create a flexible stand that integrates an articulated robot capable of autonomously picking various unknown objects from a stack and performing automatic sorting based on their visual features. In recent research, we have designed a robotic stand based on an articulated manipulator, which provides 6 Degrees of Freedom (DoF) and collision detection [10]. For our current research, we used a Delta DRV90L manipulator. The manipulator has been equipped with a specific four-finger gripper (Fig. 1) with 3 different working modes, enabling it to pick objects from a stack located within the working area of the ZIVID-2 camera. We used a fine-tuned neural network inspired by the SF-Mask RCNN (Synthetic Fusion Mask Region-based Convolutional Neural Network) architecture to analyze the objects on a stack [2]. Such a neural network can perform RGBD (Red Green Blue Depth) image segmentation [11], allowing us to identify and isolate the topmost object in the stack. Subsequently, we calculate the gripping points for the selected object and choose the grasping type. This part is further described in the current paper. Once the object is grasped, it is transferred to the vision station. This station is equipped with three RGB cameras, which are accompanied by appropriate lighting conditions to facilitate accurate object perception. By incorporating these components and techniques, our robotic stand demonstrates the ability to effectively handle objects from a stack.

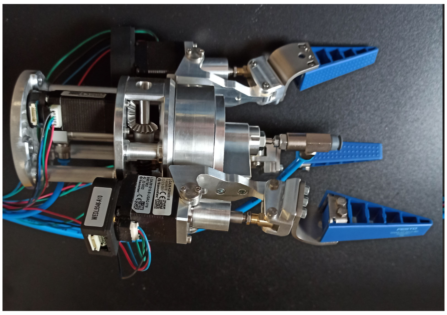


Figure 1: Our four-finger gripper with 3 different working modes: single sliding finger with a suction cup, two standard parallel fingers, or three for grasping rounded objects

1.2. Motivation and structure

This research paper involves comparing different types of grasping algorithms presented in the literature and evaluating their performance. Our main goal is to create a grasping system that is not only efficient and precise but also flexible and capable of handling diverse grasping scenarios, applicable across a wide spectrum of robot-assisted applications. Since *'It's really hard to decide the grasp when we do not have the full object geometry'* [13], we propose a new algorithm based on the analytics of object geometry seen from an RGBD camera. The algorithm is partially based on machine learning and image processing techniques. It enables the robot gripper to adapt to diverse grasping scenarios.

The paper is organized as follows. The next section describes the state of the art in systems for computing grasp points, while its subsections present a few examples of such systems in detail. The proposed algorithm is introduced in Section 3. Further on (Subsection 4.1), we will find the outcomes of comparative manual tests along with an analysis of the results. The real-world test can be found in Subsection 4.3. The article is finally concluded in the last section.

2. Related work

Since the Industrial Revolution, there has been an ongoing drive to enhance the autonomy and multifunctionality of industrial robots. Many solutions have arisen to facilitate the advancement and bolstering of these aspirations. Effectively manipulating and transferring objects requires the implementation of several algorithms: object detection, its position estimation, and designation of grasping points.

The article [13] thoroughly describes this process, providing an overview of methods for each of these elements. The authors, present the methods employed in achieving each stage, through an extensive literature review. They conclude that there are still significant challenges in each of these stages. In general, the influence of occlusions and missing geometry of the objects in the data acquisition phase is still too high on grasping algorithm decisions. 'It's really hard to decide the grasp when we do not have the full object geometry' [13]. Another very important aspect is the lack of datasets concerning object position and grasp estimation for the machine learning methods. Moreover, some machine learning algorithms show poor generalization and struggle with new objects. The last remark shows that transparent objects are very difficult to analyze by vision algorithms.

According to [59], methods for grasp point estimation may be categorized according to the availability of knowledge – full and limited. Complete knowledge means that the grasping system has a full 3D model of the objects available, thus, it knows all of the surfaces and geometries of the objects. Such an approach is used, for example, in [19, 55, 66]. On the other hand, there is a real-world approach, where data is typically derived from depth-sensing cameras. As a consequence, the grasping system does not know the shape of an object. Occlusions and obstructions of objects may affect the grasping. Examples of methods based on incomplete knowledge can be found in [21, 35, 41, 57, 64].

A broad review of different grasping methods through the years can be found in [62]. The researchers conclude that analytical methods require complete knowledge of the model and certain simplifications but are more stable than data-driven methods.

Methods for grasp point estimation can also be distinguished according to the type of algorithm into analytical, supervised learning and reinforcement learning (both supervised and reinforcement are types of machine learning). Usually, analytical algorithms go hand in hand with complete data. On the other hand, artificial intelligence methods, namely, artificial neural networks, require large datasets, extensive data annotations, but are able to deal with uncertainty in effectively detecting grasps for new objects. Methods employing neural networks for grasp computation can be found in [7–9, 35, 55, 56, 64]. Methods based on geometric analysis can be found in [19, 21, 57]. Additionally, distinct algorithms utilizing reinforcement learning can be identified [44, 46, 60]. However, this necessitates the creation of complex environments, and training is time-consuming. They lack good generalization, but are potentially useful where complete knowledge of the robot's working environment is available.

Conversely, in the work by [59], the authors tackle the time-consuming annotation problem by transferring knowledge from one annotated object to similar ones via learning templates and variations from the template, then creating a model to transfer and refine grasps for objects.

Our work focuses on utilizing point clouds to compute grasp points, similar to the studies [21, 55, 57, 64], yet the final algorithm for grasp determination does not rely on neural network models and does not leverage knowledge of object models but rather estimates complete object models and determines grasps through geometric analysis.

Based on the literature review and similarities in the problem, four algorithms, described in [1, 35, 53, 57], were selected for further tests. These algorithms involve computing grasp points for a robot with a two-fingered gripper using an RGB/RGBD camera. The availability of source code to reproduce the research results was another crucial element in the selection. Note that the code was unavailable for one of the selected works, [57], so it was implemented manually.

2.1. Deep CNN-based architecture (DNN)

The method described in [1] is based on a neural network architecture. The deep neural network architecture operates on standard color – RGB (Red Green Blue) – images only. The main backbone of this architecture is based on a slightly modified ResNet-101 (Residual Network) [20] with a Feature Pyramid Network (FPN) [39] and Inplace Activated Batch Normalization [49]. The backbone extracts feature maps from the input color image. These maps serve as inputs for grasp detection and image segmentation. Further, grasp detection is based on a Faster R-CNN [48]. It uses a Region Proposal Network (RPN) to estimate region rectangles (only axis aligned), and a grasp detection head with two outputs – grasp orientation and fixed bounding box predictions. In parallel, the segmentation branch uses Mini-DeepLab. Finally, the grasping candidates from the grasping head are refined by fusing the segment information and processing them by MLP (Multi-Layer Perceptron) with five outputs for each grasp candidate.

The authors tested their approach on the three datasets:

- Cornell Grasp Dataset – 885 RGB-D images with a resolution 640 × 480 of 240 graspable objects [28, 36].
- Jacquard Dataset – 54 k synthetic RGB-D images of 11 k graspable objects [12].
- OCID (Object Clutter Indoor Dataset) – which includes more than 1.7 k images with 11.4 k segmented objects and 75 k possible grasps [52].

In the first dataset, the authors achieved 98.2% accuracy using 5-fold cross-validation. On the second dataset, the discussed approach achieved 92.95%, while on the OCID dataset, the accuracy was 89.02%.

The accuracy was measured according to the correct grasp, defined as:

- the difference between the predicted grasp angle and the ground truth grasp angle is less than 30°,
- the IoU (Intersection over Union) calculated between the ground truth and prediction bounding rectangle is more than 25%.

Note that such a metric is highly uncertain for the real grasping purpose, especially in case of very complex objects. Moreover, the grasp accuracy measured according to the manually selected ground truth is not always translated into actual grip, in general. The most valued tests are based on the efficiency of grasping by real manipulators.

2.2. GraspNet

In the article [53], the authors propose a model for six Degree-of-Freedom (DoF) grasping of an unknown object in a cluttered space based on partial observation of a point cloud scene. They used a modified PointNet++ network [47] to build ‘an asymmetric U-shaped network’. The network has input of 20 k 3D points and can predict the grasp representations (in the form of 6-DoF). It was trained on 17.7 million synthetic grasp scenes, where the objects were based on the ACRONYM dataset (8.8k different object meshes) [16].

After training the network, the authors achieved 90% effectiveness in picking real objects based on RGB-D images from an Intel RealSense L515 LiDAR camera and a 7-DoF Franka Panda robot with a parallel-jaw gripper.

2.3. Generative Residual Convolutional Neural Network (GRCNN)

The method described in [35] relies on the Generative Residual Convolutional Neural Network, which processes n channel input. The pre-processing stage makes sure that the input is at an appropriate resolution and fixes the number of channels, thus the network can use RGB and/or Depth channels. Further, the network generates three images that correspond to the grasp angle, width and quality. The post-processing stage of the algorithm is responsible for the final grasp pose calculation.

The authors evaluated grasps on two of the standard datasets mentioned earlier: the Cornell and Jacquard datasets. They achieved, respectively, accuracies of 97.7% and 94.6%. The method was also evaluated in a real application using a 7-DoF Baxter Robot by Rethink Robotics and an Intel RealSense D435 camera. The real-world testing showed accuracies of 95.4% and 93%. The positive prediction was defined as well as in Section 2.1.

2.4. Grasp Pose Estimation (GPE)

Another method can be found in [57], which employs a four-stage grasp detection procedure based on a point cloud collected from a Kinect camera. Initially, using a point cloud, object boundary points are detected and mapped onto the image plane. Subsequently, the object's centroid and skeleton are determined. Further, multiple grip locations are identified along the object's skeleton. Finally, the best grasp is selected from the grip locations represented by drawn rectangles. The approach has been tested with the UR10 robot. The report shows results from a real test as a grasping accuracy of 88.16% and 77.03%, respectively, for distinct and cluttered objects. Part of this method was adapted in our approach for the suction cup grasping.

3. Grasping point algorithm (GPA)

The whole bin-picking process starts with image acquisition from the ZIVID-2 camera pointed at the stack of objects. The image is processed by camera software to obtain a point cloud and normal array. Further, the data are transformed into a robot coordinate system using a hand-eye calibration algorithm. The process of the Grasping Point Algorithm (GPA, presented in Fig. 2) starts after the stack image segmentation using the SF-mask RCNN described in [2, 11]. The result of the SF-mask may be depicted as m different image segments $\{P_1, P_2, ..., P_m\}$, which should usually correspond to different objects in the stack. Since the RGB image is linked to a point cloud, the segments can be also represented in 3D space. The index of objects is assigned by the SF-mask neural network and further used to estimate the grasping order by the queue method described in Section 3.1. The grasping order should reduce the possibility of collisions during object picking. Note that the first object processed by the GPA should be the highest and most coherent on the stack. The queue method also takes this into account.

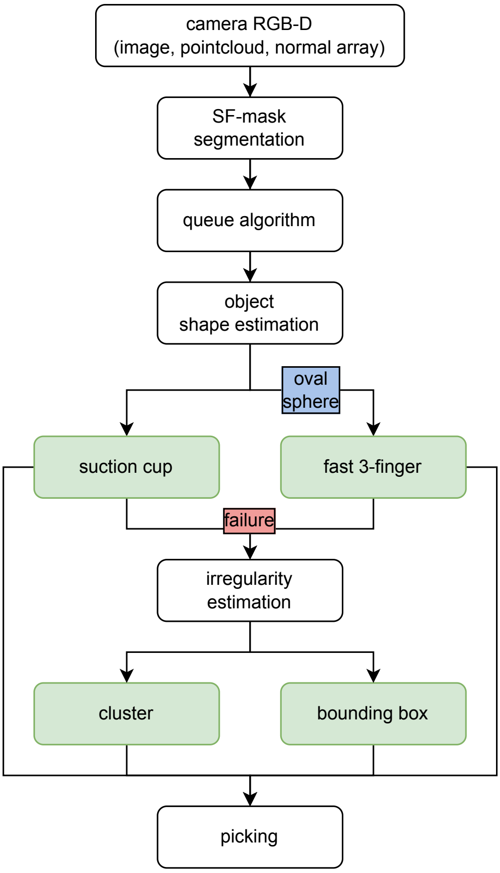


Figure 2: Block diagram of the decision-making process

Subsequently, for the best-selected object b, normal array N_b is used to calculate its sphericity and ovalness; a more detailed description is given in Section 3.2. If the object exhibits the above-mentioned characteristics, the grasping points are calculated using the Fast Three Fingers (FTF) algorithm described in Section 3.4.

Otherwise, grasping points for the Suction Cup (SC) are computed (Section 3.3), as this type of grasping is the most stable and takes less computation time for the majority of objects. If the algorithms fail to find grasping points or their quality is not good enough for the grip, another attempt is calculated, based on a two-fingered grip.

Firstly the irregularity parameter of the considered object is calculated based on its flat surfaces. If the object consists of large (in terms of object dimensions) flat surfaces, the Cluster (Section 3.5) algorithm for two-fingers is computed, as it performs well with large clusters. For irregular objects with many side walls and irregular upper surfaces, the Bounding Box (BBox) algorithm (Section 3.6) is applied.

After each method, the quality of all proposed grasping points is calculated (selectively for the type of grasping). The best points are also validated by simple collision detection with other objects. The colliding grasping points are simply removed from the list. Finally, if at least one valid point is identified, it is converted to robot tool coordinates, and the object is picked.

3.1. Establishing the order of objects for grasping

Our algorithm uses the point clouds P of all segmented objects to determine the grasping sequence. The proposed method examines points belonging to object i (P_i), and computes several parameters, namely: mean height h_i, continuity c_i, and surface v_i. The aspect of the proximity of other objects or the environment is also taken into consideration as e_i.

The height of object i is calculated based on the average of 90% of the lowest points, using their z coordinate values (distance from the robot base). The highest points are not considered to mitigate the impact of noise that may arise due to camera inaccuracy. The calculated height is normalized. This metric favors the highest objects as they have the smallest chance of collision with other items. Such objects should also have a more accurate point cloud as the other items should not obscure them.

The volume parameter v_i is estimated based on the number of points from the segment. Bigger items give more opportunities to be grasped than very small objects. They are also less likely to be obscured by other objects.

Continuity c_i is calculated with the DBSCAN algorithm [58]. The continuity value is high when an object is divided by the algorithm into one big cluster, and it is low when a few small clusters are found. This parameter highly penalizes objects lying under other items as they are less likely to be successfully picked up.

The surroundings of the object i are examined by determining an oriented bounding box parallel to the global X, Y, Z axes. The environmental parameter –

e_i – is calculated as the proportion of points inside the bounding box belonging to the object i, and the other points. If the value for the environment parameter is low, the probability of potential collision increases.

The final object parameter is calculated according to the formula:

$$g_i = (h_i + 1) \cdot c_i \cdot e_i \cdot v_i \, , \tag{1}$$

where:
g_i – parameter of the order for an object i
h_i – normalized height of the object i
c_i – continuity value of the object i
e_i – surroundings of the object i
v_i – volume parameter of the object i.
The results are sorted in descending order by the g parameter. The object with the highest g value is considered the best candidate for picking.

3.2. Sphericity and roundness of an object

The three-finger gripper option works best when handling spherical, cylindrical, or in general, rounded-shaped objects. Based on the paper [5], the sphericity parameter – ρ – can be calculated from the distribution of normal vectors for 3D points of the selected object.

The normalize normal vector for each point along each axis is assigned to corresponding intervals n_{axis} within the range $\langle -1, 1\rangle$. The number of intervals n_{axis} has been set to 200. The frequency of intervals is normalized by dividing them by the total number of points. Based on the data, three histograms – H_x, H_y, H_z – (one for each axis) are created. The function parameter ρ can be described as:

$$\rho_i = \frac{\int_{-1}^{1} (H_x + H_y + H_z)}{3} \, . \tag{2}$$

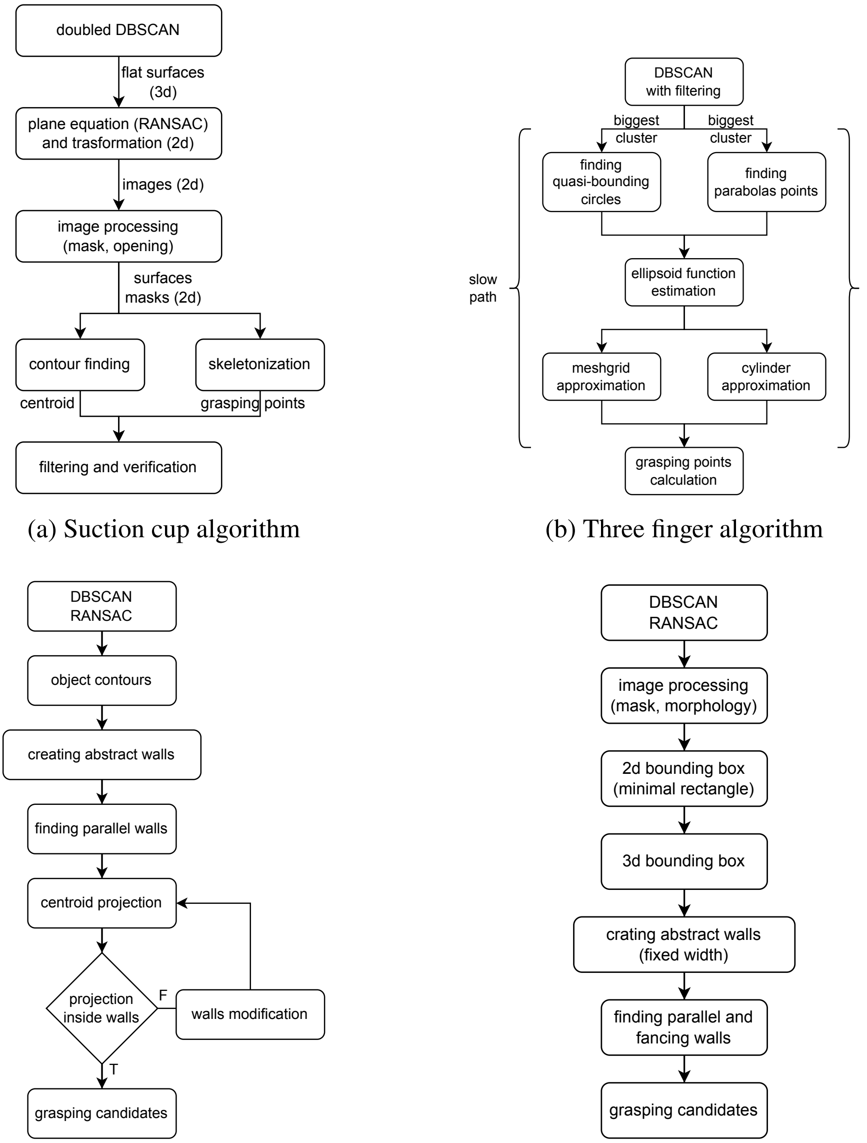
The roundness of the object, ϕ, is computed using the direct point cloud of the object. All points are projected onto a two-dimensional plane. Further, the convex hull is computed by the standard Jarvis algorithm [26]. The convex hull points are used to obtain the perimeter ($rhull_i$) and convex area ($Ahull_i$) by the shoelace algorithm. The circularity value of the object i is calculated as:

$$\phi_i = \frac{4 \cdot \pi \cdot Ahull_i}{rhull_i^2} \, . \tag{3}$$

For a perfect circle, this value is equal to 1.

3.3. Suction cup algorithm (SC)

The algorithm for determining the grip for the suction cup (one gripping point) was based on [57], as described in Section 2.4. The main differences include data source (using only point cloud data) and gripping method (only a suction cup). The scheme of the algorithm is presented in Fig. 3a.



(a) Suction cup algorithm

(b) Three finger algorithm

(c) Cluster algorithm

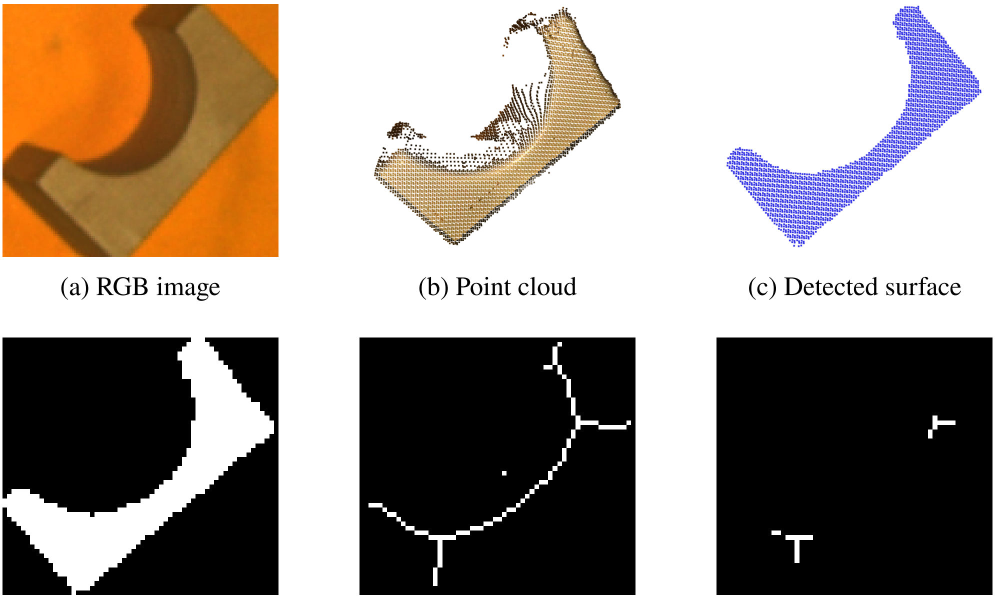
(d) Bounding box algorithm

Figure 3: Details of used algorithms

Firstly, on a segment of the i-th object, we are searching for flat surfaces without any holes or cavities. To identify these surfaces, the parallel implementation of the DBSCAN algorithm is applied [58]. The initial point cloud for the DBSCAN algorithm is pre-processed by adding an additional channel to each point – its normal vector multiplied by a large scalar. This operation helps us to divide the point cloud of an object into similar surfaces.

The resulting clusters are once again passed through the DBSCAN algorithm, this time in original form (without additional normal coefficients). Similar surfaces are divided if there is a space between them (Fig. 4c). Further, we use the RANSAC algorithm [65] to calculate the equations of the surfaces found in the previous step.

Potential grip points are determined separately for every flat face. Firstly, the points are projected onto the XY plane. Further, a two-dimensional grid mask is created, where each point from the transformed surface is assigned to a single pixel as its approximation (Fig. 4d). The resolution of the grid is chosen arbitrarily (1 mm per pixel). The obtained mask image is transformed with a morphological operator – opening (erosion followed by dilation) using a structuring rectangle with kernel shape 3 × 3. The centroid of the mask is calculated, as an obvious grasping candidate. Further, the mask is skeletonized by the algorithm in [63] (Fig. 4e). The skeleton of the surface is used exactly as in [57].


(a) RGB image

(b) Point cloud

(c) Detected surface

(d) 2D grid mask

(e) Object skeleton

(f) Filtered grasping points

Figure 4: Suction cup grasping computation steps

The suction cup is represented as a binary circle with a radius corresponding to the suction cup installed on the robot and the chosen resolution of a surface grid. For each potential grip point, the logical **and** operation is performed between the obtained mask and the binary representation of the suction cup. Such filtering is intended to eliminate any openings (within each grip point that would cause air to escape) and points lying too close to the edges of the surface (Fig. 4f). Further filtering also includes the distribution of the transformed z coordinates in the vicinity of the gripping point, which determines the irregularity of the surface. The obtained results are transformed back to the original robot coordinates. Based on the plane equation, we calculate the gripping angle for the suction cup.

Finally, the potential grasping points are validated. For the suction cup, the most critical quality parameter is the Euclidean distance from the object's centroid (CoG) – d_{CoG}. In general, a suction cup should aim to catch the object at its center of gravity. The further away from the center, the less stable the grasp may be, and more force is needed to lift the object.

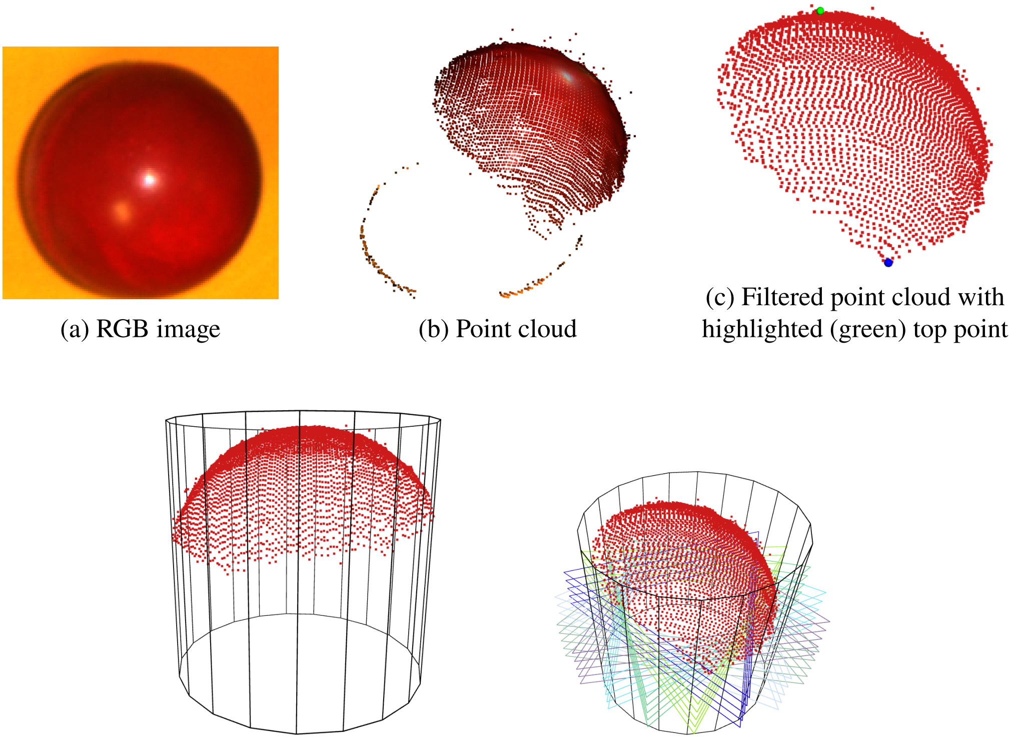
3.4. Three finger algorithm and its acceleration (FTF)

As was mentioned earlier, a three-fingered gripper can lift round objects (such as a ball, an egg, or a cylinder), which is its advantage. Such a conclusion can also be derived from the literature, e.g. [50], where the 'basic mode' works for rounded objects.

With a circular object, just like with a suction cup, the robot approaches the object at a right angle to the plane on which the grip points are located. The algorithm is presented in Fig. 3b, where the 'slow' path is an original concept, and the 'fast' path is further modified and currently working on a robot.

Our algorithm (TF – Three Finger) to calculate grasping points for rounded objects (Fig. 5a and Fig. 5b) is based on the fact that such objects can be approximated to a certain ellipsoid. Initially, the point cloud is downsampled and filtered (outliers are removed) (Fig. 5c). Subsequently, we use the DBSCAN algorithm to select the largest points cluster. The algorithm searches for quasi-bounding circles within a several fixed height range of the cluster points. The Cayley transform ellipsoid fitting (CTEF) algorithm [43] estimates the ellipsoid function from the calculated circles. Next, a mesh grid is created, and for each grid square center, a grip point is estimated with the next two points at 120° and 240° apart.

Due to the high computational complexity and long time of these calculations, the algorithm has been modified to improve its speed (FTF – Fast Three Finger). Instead of estimating a paraboloid, we use the knowledge of the object's vertex (highest point – green dot in Fig. 5c), its edges, and centroid. Thus, the problem of shape estimation can be reduced to a two-dimensional bounding ellipse calculation. The obtained ellipse provides the parameters of the bounding cylinder (Fig. 5d). To get potential grasping points, several circles are placed every fixed distance on the sidewall of the cylinder. Next, all circles are divided into slices of a fixed angle. The grasping points are the intersection points of these operations. As was mentioned above, grip points are connected in a set of three points every 120°, as shown in Fig. 5e as triangles. Note that circular objects must be gripped below its widest point.



(a) RGB image

(b) Point cloud

(c) Filtered point cloud with highlighted (green) top point

(d) Estimated cylinder walls

(e) Potential grasps

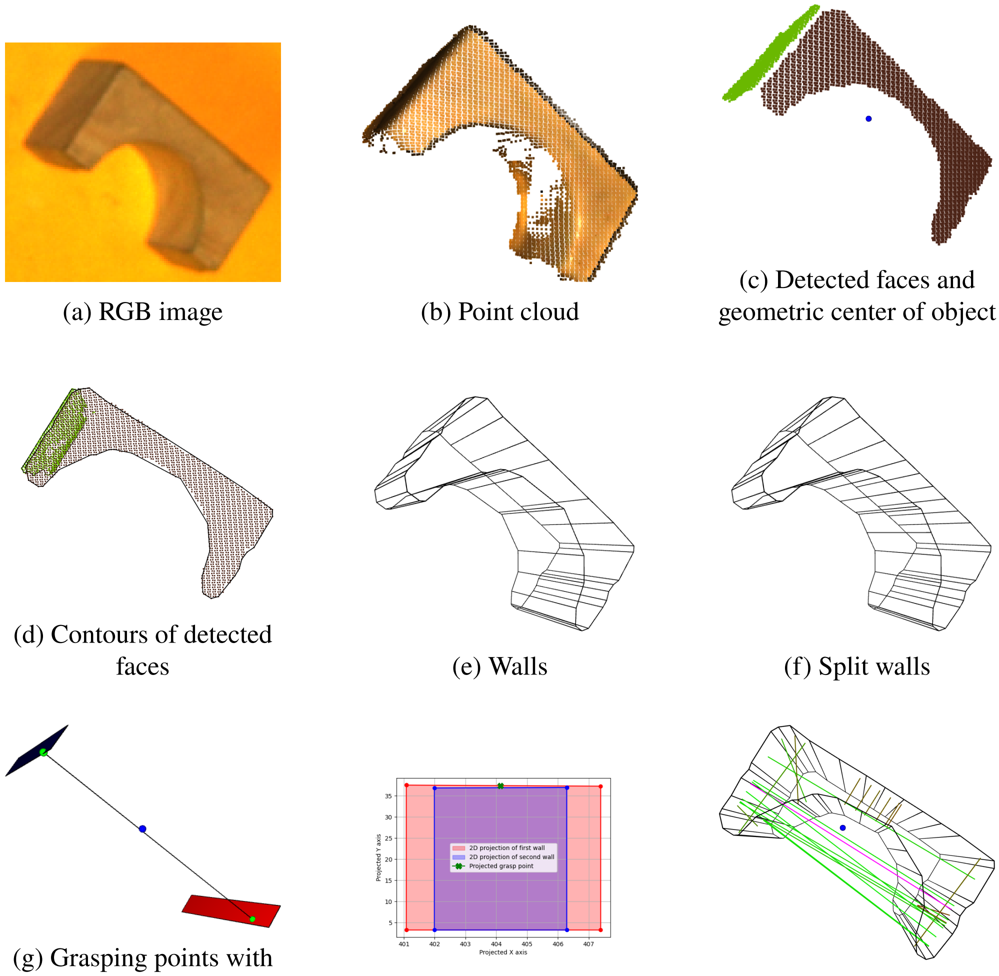
Figure 5: Three fingers grasping computation steps

This algorithm, besides spherical objects, can determine grasping points for a cylinder, skipping the height determination step. Instead of estimating the ellipse, the lowest and highest points of the cylinder are determined.

For the three-fingered grip, the grasp quality parameter primarily revolves around the depth of the grip – d_{grip}. The object must be gripped as low as possible to ensure it stays firmly in the gripper without slipping out during transportation. The grasp depth is calculated as the Euclidean distance between the highest point of the object and the mean value of the proposed grasping point.

3.5. Cluster based algorithm (Cluster)

The scheme of a cluster-based algorithm for a two-fingered gripper is presented in Fig. 3c. This algorithm, similar to the suction cup method described earlier, is based on surfaces found by DBSCAN (Fig. 6c). In this case, for a two-fingered gripper, the flatness of surfaces is not strictly necessary for computation. The surfaces found are projected onto a two-dimensional plane. For each surface, its contour is calculated by applying the alpha shape algorithm [14] (Fig. 6d). Close contours are joined together to simplify the computation, which also re-



(a) RGB image

(b) Point cloud

(c) Detected faces and geometric center of object

(d) Contours of detected faces

(e) Walls

(f) Split walls

(g) Grasping points with geometric center of object

(h) Projection points

(i) Estimated grasps

Figure 6: Cluster algorithm computation steps

duces the number of potential bad gripping points and decreases the possibility of collision between the object and the gripper's fingers.

Next, abstract vertical walls are formed based on subsequent points belonging to the contours of selected surfaces (Fig. 6e). Their height is predefined. To increase the subsequent number of grip points, wide walls are divided to match the average width of the wall (Fig. 6f). Further, the algorithm searches for pairs of quasi-parallel walls by comparing the angle between their normal vectors. Two walls are considered quasi-parallel if the value of the θ angle is within the accepted range – $\langle 0°, 15° \rangle$ and $\langle 165°, 180° \rangle$. For each pair of parallel walls, an additional check is carried out to determine if the walls intersect, using their projection into a two-dimensional plane.

The geometric center of the analyzed point cloud is also determined. Further, the algorithm analyses each pair of parallel walls. The centroid of the considered object is projected onto the walls to create so-called projection points. If the projection points lie inside both examined walls, they are considered as grasping points (Fig. 6g). If one of the projection points is not within the wall area (Fig. 6h), the top edges of the walls are rotated to make them parallel to the x-axis. Further, the median of the x coordinates is calculated along with the y coordinate for the 'upper' edge of the wall for the selected x value. Such a median is projected back to real coordinates and added to the list of potential grip points. All estimated grasping points are shown in Fig. 6i.

The quality of the grasping points is influenced by both the depth and the distance to the object's centroid. Thus, the quality parameter is calculated as an average of the distance from the centroid and the grasp depth. This refers to providing the biggest grasping area while minimizing the potential force required to lift the object.

3.6. Bounding Box (BBox) algorithm

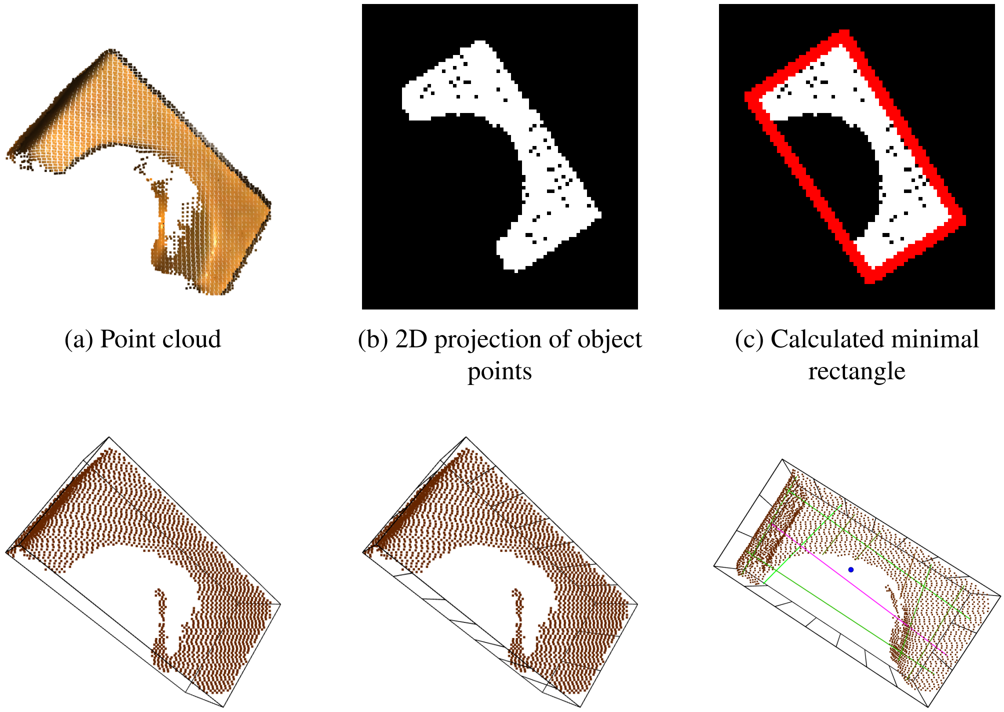
The Bounding Box algorithm involves estimating a 3D bounding box around an object. The algorithm is depicted in Fig. 3d. Similarly to the previous algorithm, to determine the boundaries, the point cloud is initially downsampled and filtered (Fig. 7a). Further, the points are projected onto a two-dimensional plane. Similarly, as in the suction cup algorithm (Section 3.3), a 2D mask is created based on a projection and the minimal quadrilateral area surrounding the resulting figure is determined. The four corners of the envelope on the two-dimensional plane are transformed back into their corresponding points in 3D space. This process determines the four upper vertices of the bounding box.

The lower vertices have the same x and y coordinates as the upper ones, while their z coordinate is determined using the geometric center CoG and the maximal

z vertices:

$$\text{lower}_z(j) = \text{upper}_z(j) - 1.8 \cdot (\max_j(\text{upper}_z) - CoG_z), \tag{4}$$

where $j \in < 1,4 >$ is the index of the corner, and $upper$ or $lower$ are the appropriate coordinates (Fig. 7d).



(a) Point cloud

(b) 2D projection of object points

(c) Calculated minimal rectangle

(d) Estimated bounding box

(e) Split bounding box

(f) Estimated grasps

Figure 7: BBox grasping computation steps

For each of the four sides of the resulting object, n finger-width walls are determined (Fig. 7e). For each pair of walls, our algorithm checks their parallelism (by comparing their normal vectors) similar to Section 3.5. Parallel walls are also checked to determine whether they lie opposite each other. Further, the median of the x coordinates of the upper edges is used as a potential grip coordinate, while its y coordinate is determined by the y_{min} and $y_{\max}$ coordinates. The grip height is simply the CoG_z.

The quality of the grasping points in the bounding box algorithm is the same as in the previous method.

4. Tests and their results

To properly assess the quality of the algorithms and the effectiveness of the final version of the proposed GPA algorithm, a number of tests were carried out. The component functions of the GPA algorithm (namely SC, FTF, Cluster, and BBox) were compared with algorithms available in the literature and Git repository. At the same time, an entire pick and place station was tested by moving 100 different stacks of 25 objects each. To thoroughly analyze the algorithm's inaccuracies, a single object transfer test was also performed.

4.1. Comparison of algorithms

To compare the described (Section 2) and implemented functions in our algorithm (Section 3), each of them was tested on objects (Fig. 8) of different shapes, using the generated grasping points. Note that the described algorithms from the state-of-the-art section are designed only for a two-fingered gripper. Thus in certain cases, e.g. a ball, they may not provide any results. As was described earlier, the quality of the grasp can be evaluated as the distance from the center of gravity and the depth of the grasp. The results of such a metric for each of the fifteen described objects are detailed in Table 1 and Table 2. Figure 9 presents examples of grasping points determined by the algorithms, indicating how the gripper approaches to pick up the object.

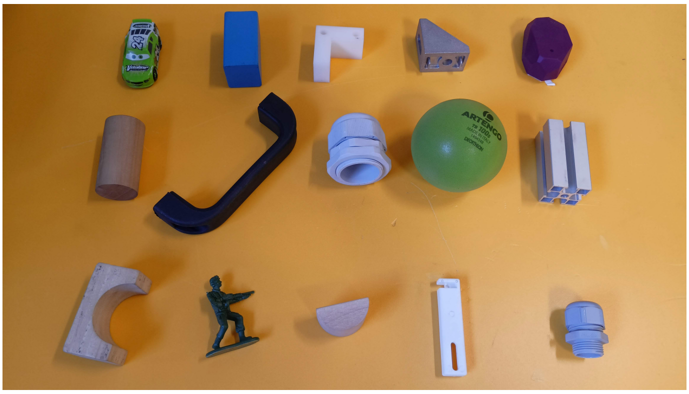
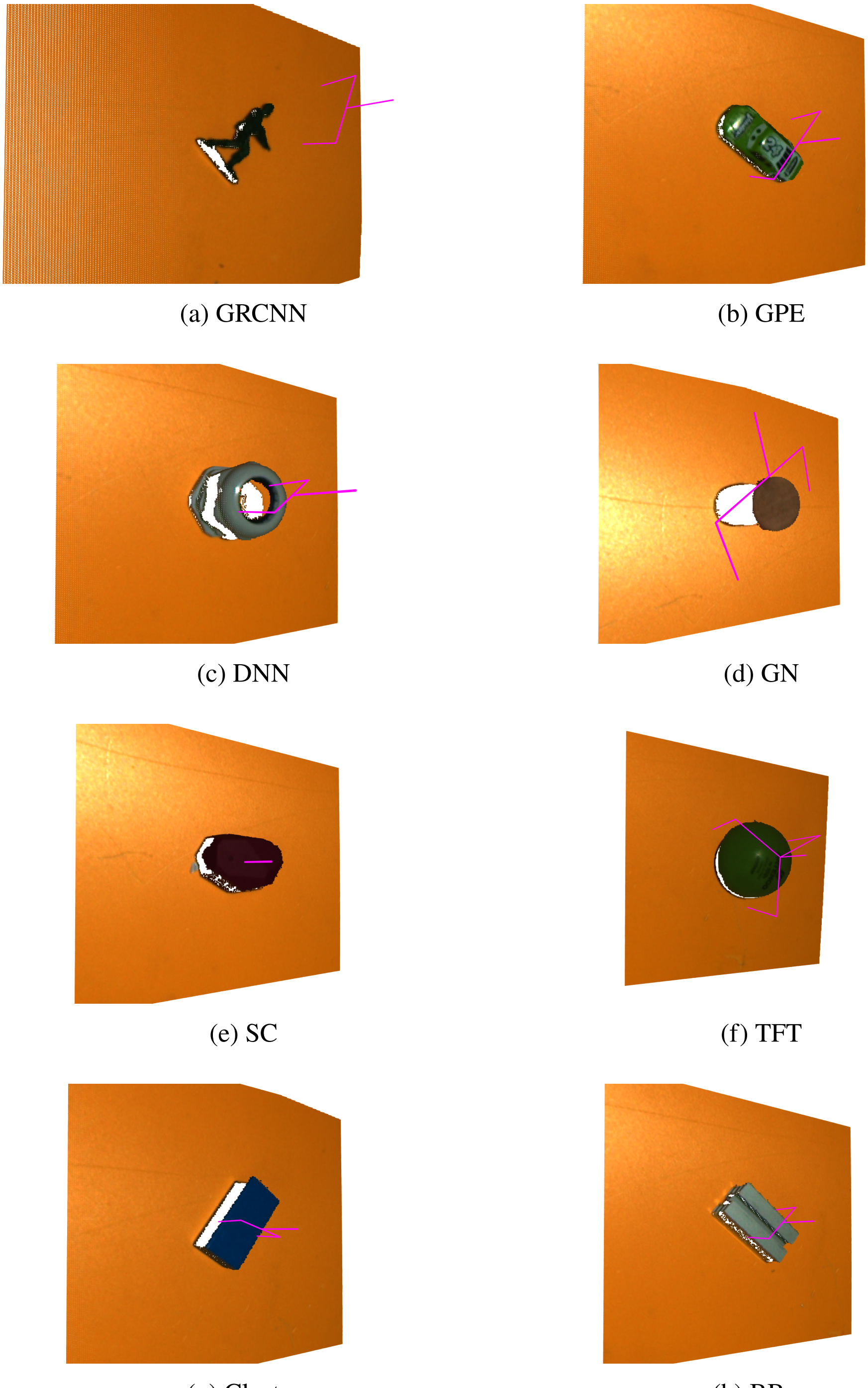


Figure 8: Objects for test algorithms

According to Table 1, the GRCNN algorithm proposes grasping points far away from the CoG of the object. The majority of such grasps are not valid and are efficient. The GPE algorithm achieves quite good results. Its distances are similar to Cluster and BBox algorithms, however, in certain cases, the GPE algorithm did not generate any points, like with the toy soldier. Note that the toy



(a) GRCNN

(b) GPE

(c) DNN

(d) GN

(e) SC

(f) TFT

(g) Cluster

(h) BBox

Figure 9: Results of the estimated points to grasp by each of algorithms

soldier is the hardest example to process. The DNN algorithm did not perform well. Our assumption in that case is that the neural network had been trained for different cases. On the other hand, the GraspNet had been trained on a bigger/more varied dataset since it worked better for 12 cases.

Table 1: Average distance in mm of grasps from CoG of the objects for different algorithms (the lower the better). The number of positive estimations (any estimation near object) out of 5 attempts is given in brackets

Object no.	GRCNN	GPE	DNN	GN	SC	FTF	Cluster	BBox
1 (toy car)	47.62 (2)	4.51 (5)	16.94 (1)	– (0)	15.33 (5)	1.61 (5)	4.91 (5)	7.94 (5)
2 (cube block)	50.41 (2)	5.61 (4)	– (0)	11.72 (3)	0.81 (5)	0.74 (5)	4.81 (5)	3.25 (5)
3 (white corner)	21.12 (2)	– (0)	– (0)	8.88 (5)	13.18 (5)	3.23 (5)	6.27 (5)	8.64 (5)
4 (aluminium corner)	48.79 (1)	3.22 (3)	– (0)	15.91 (2)	1.39 (5)	3.2 (5)	9.66 (5)	4.67 (5)
5 (beacon)	41.58 (2)	3.36 (5)	– (0)	4.8 (1)	4.86 (5)	1.8 (5)	3.9 (5)	4.45 (5)
6 (cylinder block)	9.75 (1)	2.5 (4)	– (0)	10.04 (5)	2.07 (5)	1.24 (5)	1.94 (5)	3.47 (5)
7 (handle)	47.4 (1)	16.02 (4)	– (0)	41.57 (1)	29.77 (2)	11.94 (5)	21.7 (5)	40.9 (5)
8 (PG29 gland)	16.58 (2)	15.5 (2)	2.55 (4)	3.57 (5)	10.66 (1)	2.58 (5)	2.24 (3)	1.9 (5)
9 (ball)	52.74 (1)	20.73 (2)	– (0)	5.08 (5)	– (–)	4.27 (5)	6.49 (5)	21.71 (5)
10 (aluminium frame)	56.91 (1)	2.65 (3)	– (0)	10.98 (3)	11.29 (4)	1.69 (5)	3.88 (5)	3.11 (5)
11 (bridge block)	32.35 (3)	8.57 (3)	– (0)	26.55 (5)	12.9 (5)	3.06 (5)	5.73 (5)	5.7 (5)
12 (toy soldier)	60.42 (1)	– (0)	– (0)	– (0)	– (–)	2.33 (5)	4.13 (5)	4.69 (5)
13 (half of cylinder)	46.12 (1)	3.37 (4)	– (0)	14.21 (4)	1.82 (5)	2.73 (5)	2.81 (5)	1.93 (5)
14 (plastic detail)	- (0)	7.1 (1)	21.15 (2)	– (0)	3.98 (5)	0.75 (5)	1.64 (5)	9.2 (5)
15 (M20×1.5 gland)	50.92 (1)	7.45 (1)	3.69 (2)	6.44 (5)	– (–)	1.04 (5)	1.3 (5)	1.3 (5)

For the Suction Cup algorithm, the distance from the CoG is crucial, however, the most important parameter is that the grasping point should lie on a flat, nonperforated surface. In some cases, the metric is similar to the FTF algorithm. The FTF algorithm achieves the minimal distance from the CoG, which is natural for the case of three fingers. The Cluster and BBox algorithms stay in the middle, showing (for most cases) quite a good distance from the CoG.

The grasp depth shown in Table 2 is quite an important parameter. In theory, the best grasping points are the deepest ones. In our cases, the grasping points should lie on the bottom half of the objects. The gasping points near the top surfaces are most vulnerable to accidentally slipping out of the grip. Note that since the Suction Cup works only for top surfaces, its grasping depth as a parameter is useless.

Table 2: Average grasp depth in mm of grasps for different algorithms. The depth of each object is presented in column ‘h’. Note that the grasp depth is measured before the system checks the environmental conditions, such as obstacles

Object no.	h	GRCNN	GPE	DNN	GN	FTF	Cluster	BBox
1 (toy car)	18.2	8.58	6.75	14.26	–	22.49	10.89	6.56
2 (cube block)	28.6	2.16	7.54	–	12.96	36.77	7.33	4.3
3 (white corner)	38.6	8.77	–	–	27.79	35.96	23.71	15.22
4 (aluminium corner)	24.4	12.54	5.68	–	21.04	27.91	6.43	4.78
5 (beacon)	22.8	5.77	6.53	–	34.8	32.16	7.26	5.1
6 (cylinder block)	59.2	29.79	5.42	–	20.09	38.72	6.81	7.45
7 (handle)	22.7	0.01	6.72	–	21.26	31.53	6.56	4.55
8 (PG29 gland)	47.2	2.85	9.64	3.16	25.17	33.03	18.11	18.45
9 (ball)	65.0	49.7	4.35	–	34.78	33.2	22.27	5.11
10 (aluminium frame)	29.2	15.17	13.09	–	19.79	34.05	8.27	5.94
11 (bridge block)	29.9	14.96	6.07	–	16.83	34.11	6.43	5.34
12 (toy soldier)	17.8	11.34	–	–	–	20.39	11.2	11.02
13 (half of cylinder)	29.5	15.07	6.09	–	18.41	32.8	7.68	5.77
14 (plastic detail)	9.2	–	9.72	7.66	–	13.81	7.33	3.35
15 (M20 × 1.5 gland)	35.9	38.43	44.34	38.71	14.9	19.74	11.99	11.99

For objects 4, 6, and 9–13, the GRCNN shows the depth of the grasping points in the middle of the objects, which is very good behavior. However, for the other objects, the grasp is too shallow. The GPE grasping points in general are also too shallow. On the other hand, the depths of GraspNet’s points is quite good. The grasping points do not extend the depth of the objects and they are more than 10 mm, thus the probability of slipping out decreases.

The FTF algorithm shows the grasping points at a depth bigger than the object’s height. This phenomenon is caused by the fact that the gripper should catch objects in the middle of its fingers. However, the grasp depth is measured before the system checks the environmental conditions, such as obstacles. Thus, the next step of the GPA algorithm is to check the obstacles and fix the depth of the grasping points so that they do not exceed the depth of the object. The Cluster and BBox algorithms show the depth at quite a nice level, sometimes too shallow (like in the case of the aluminum corner).

In Table 3, the quality of the grasping point was evaluated manually on a scale of:

0) the algorithm did not generate any grasping point,

1) the gripper will not catch the item (impossible grasp),

2) there is a big chance that the gripped object will slide out,
3) the object will probably slide out during the movement,
4) there is a small chance of sliding out,
5) the gripper will catch the object without it sliding out (perfect grasp).

Table 3: Human evaluation of best grasp, graded 1–5, where 1 means that object can't be gripped while 5 is a perfect fitted grip

Object no.	GRCNN	GPE	DNN	GN	SC	FTF	Cluster	BBox
1 (toy car)	1	3.8	0.4	0	2.4	3	2.6	3.8
2 (cube block)	1	3.6	0	2.5	5	1.2	3.4	4.2
3 (white corner)	0.8	0.2	0	2.2	4	1	1.2	1.2
4 (aluminium corner)	1	1.4	0	0.8	5	3.4	2.2	2
5 (beacon)	0.6	3.6	0	0.4	5	3.8	1.6	3.6
6 (cylinder block)	0.4	2.4	0	2	5	5	2.2	1.8
7 (handle)	1	3.6	0	0.4	1.2	2.6	2	1.2
8 (PG29 gland)	0.4	1.2	1.6	2.8	0	5	2.2	3.4
9 (ball)	0.5	0.5	0	2.25	0	5	1.5	1
10 (aluminium frame)	1	3.25	0	0.5	2.75	3.25	3.75	4.5
11 (bridge block)	1	3.75	0	1	4	2.5	2.75	3
12 (toy soldier)	1	0	0	0	0	2	1.2	1.2
13 (half of cylinder)	1	2.25	0	1.5	5	3.5	2.5	2.75
14 (plastic detail)	1	0.75	0.5	0	4	1	2.5	3.25
15 (M20 × 1.5 gland)	1	1	1.6	3.6	0	5	3.2	3.2
average	0.85	2.09	0.27	1.33	2.89	3.15	2.32	2.67

Each object was analyzed five times in different positions and angles in a workspace (the orientation of the objects did not change). The results of such tests are presented in Table 3. In general, the GRCNN method almost always generates a grasping point, however it is usually outside of the object. The GPE method is most useful from the external ones. Its average score is similar to the Cluster and BBox algorithms. The score of the DNN method is quite low since it had been trained on a certain closed dataset where the objects were completely different from those tested. This is the biggest disadvantage of using neural networks. On the other hand, in several cases, the Contact GraspNet (GN) gives the best results. Especially in the case of the gland, GN proposes grasping points perpendicular to the planes of the nut but at a certain angle to the gland, which is a very interesting approach.

For our partial algorithms, the results show that the FTF always works well with round objects (which was the main assumption of this algorithm). If there is a flat plane and a regular shape of the object, we should use a suction cup. The Cluster and BBox algorithms work well for rectangular objects, however they have some issues with irregular shapes. Moreover, they are not complementary to each other. In most cases, they give similar grasping points. Thus, in the future, they should be merged into one algorithm.

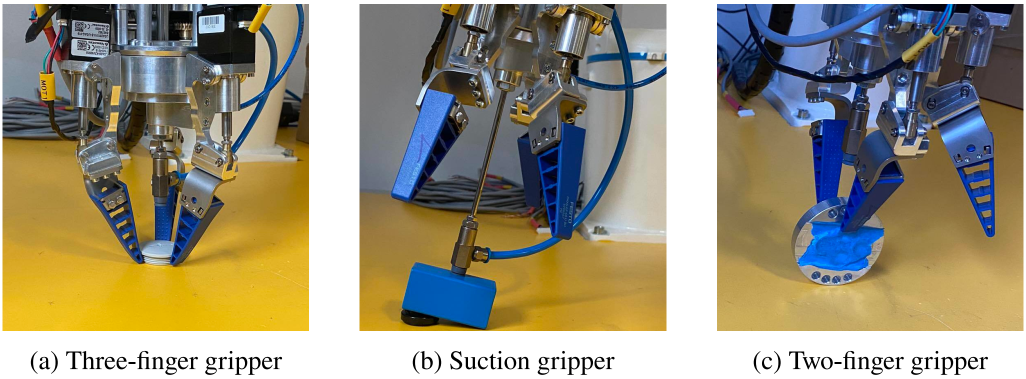
The state-of-the-art methods can only determine gripping points for two-fingered grippers. Our expert's evaluation shows that they strongly outperform our methods in only two cases, for objects 1 and 7. Note that the GRCNN method relies on a neural network [35], and in the case of objects other than those in the training set, the grip may be inaccurately determined. The model lacks the ability to generalize, which is a drawback and renders it unsuitable for these studies due to the goal of adaptability to each type of object. Similarly, GraspNet works quite well for several objects, while in other cases, the effect of the finite learning dataset is also visible. On the other hand, the GPE method [57], based on geometric analysis, performs better than any other state-of-the-art method.

Note that our final algorithm, GPA, consists of all four functions, which complement each other depending on the type of object.

4.2. The environment

The Delta DRV90L robot with its controller box was used in the tests. RGBD images were taken using a ZIVID2 camera. The robot and gripper were connected to the host PC via the Modbus protocol, while the camera used Ethernet communication via the ZIVID API. The host PC was equipped with an Intel i7-11700F processor, 32 GB of RAM and an RTX3060 graphics card.

Our tool, presented in Fig. 1, is an originally designed gripper with three possible ways of grasping objects: a three-fingered, a two-fingered, or a suction cup grip. The suction cup is centrally located on a pneumatic actuator. It extends below the gripper's fingers when it picks up an object and hides over the fingers otherwise. The three fingers of the gripper are positioned at 120° intervals and close by clamping towards the center. One of the gripper's fingers can be rotated around the gripper's center axis, which allows it to align parallel to one of the remaining fingers, creating the two-fingered grip. Examples of object manipulation by the gripper are depicted in Fig. 10.



(a) Three-finger gripper (b) Suction gripper (c) Two-finger gripper

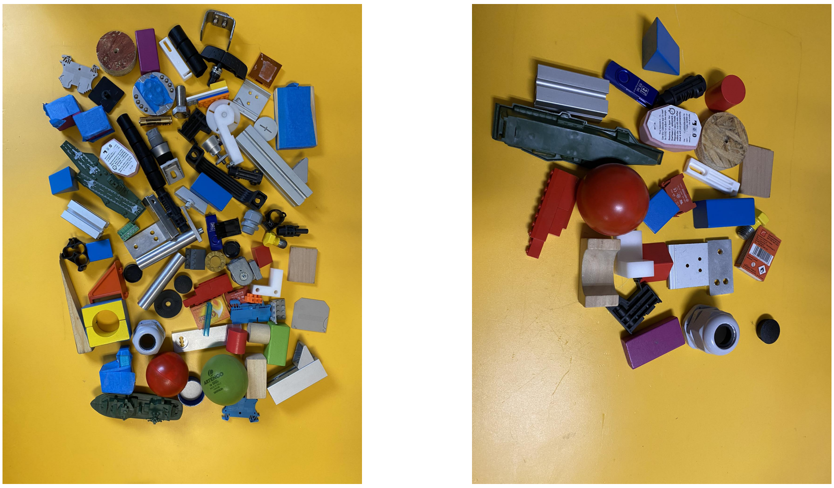
Figure 10: Sample grips using different configurations

4.3. GPA algorithm test

Experiments involving the gripper's object handling were conducted through two scenarios: picking up individual objects from the workspace, and unloading a stack of objects. In both cases, the mentioned manipulator was used. All tests and grasping attempts were carried out under consistent environmental conditions in a single location with constant lighting.

4.3.1. Individual test

In the first scenario, 80 diverse objects in terms of shape and color were chosen (Fig. 11a). For each object, five attempts were made to pick it up. Thus,



(a) Individual objects (b) The objects from the stack

Figure 11: Objects used in tests

the robot performed 400 grasping attempts. In each attempt, the objects were positioned randomly.

For the individual object grasping experiment, a successful grasp involved the robot transferring the object from the initial placement to predetermined location – target storage box – and then releasing the object there. An unsuccessful grasp was noted if the algorithm failed to determine the grasp, the gripper did not pick up the object, the object slipped during transfer, or if it was not released at the destination point.

After 400 attempts, the successful transfer of a single object was found to occur in 91.25% of cases. The results divided into the gripper mode are presented in Table 4. Table 5 contains information about the grasping mode selected for different objects. Note that only 4 object were grasped in 5 attempts by three different modes. The same table also shows that one object was very hard to grasp – only one attempt was successful.

Table 4: Number of attempts made with a particular gripper mode

Gripper mode	Number of attempts	Successful attempts[%]
Suction cup	223	95.51
Two-finger	144	84.72
Three-finger	33	90.90

Table 5: Number of objects grasped by each gripper and number of successful attempts at grasping an object

Selected gripper mode/s	Objects	Successful attempts	Objects
Suction cup	27	5	51
Two-fingers	14	4	24
Three-fingers	3	3	4
Suction cup, Two-fingers	25	2	1
Suction cup, Three-fingers	1	1	0
Two-fingers, Three-fingers	6	0	0
Suction cup, Two-fingers, Three-fingers	4	–	–

4.3.2. Stack unloading

In the second scenario, we tried to unload 100 different stacks composed of 25 distinct objects (Fig. 11b). We achieved a success rate of 92.24%. On average, we moved about 23 objects from each stack. Our algorithm and robot achieved full success only 8 times – moved all objects from the stack to the destination. The details of the test are presented in Table 6.

Table 6: Number of attempts for which a specific number of successful transfers occurred

Successfully placed objects	Number of attempts
21/25	3
22/25	10
23/25	71
24/25	6
25/25	8

4.4. Discussion

Determining the appropriate gripper mode for objects and defining their sequence works optimally, but there is always room for improving the algorithm's quality. A significant challenge was encountered with small irregular objects, where the algorithm failed to identify the object's top surface or struggled to determine the bounding box, thus unable to ascertain a suitable grasp. Small objects pose greater difficulty due to their less precise point cloud compared to larger or medium-sized objects. Objects grasped using the suction cup present an issue when their surface is not perfectly flat but this deviation is too subtle to be detected reliably by the software after capturing the image.

The three-fingered gripper performs excellently with spherical objects. Errors here stemmed from misjudging whether an object is spherical or not. Identifying an object's spherical or round shape works optimally, but achieving maximum accuracy solely from a single camera placed statically cannot be guaranteed. Additionally, in several instances, image segmentation failed, categorizing one object as two or not detecting parts of an object, especially if an object had multiple colors.

The tests on individual objects revealed that some errors arise from the camera's inaccuracies, segmentation issues, or the specific nature of some objects that are challenging for geometric analysis. The suction cup performed best at grasping objects, followed by the three-fingered gripper, while the two-fingered gripper showed the weakest performance. Issues with the suction cup arose not only from the objects' imperfectly flat surfaces at the grasp point but also from any excessive distance from the objects' geometric center. Establishing the distance limit for the suction cup from the geometric center poses a challenge as it varies for each object depending on its shape and mass distribution. The subpar results of the two-fingered gripper stem from attempting to grasp irregular, challenging objects complicated by data noise. This version of our gripper lacks built-in sensors on its fingers, leading to instances where it did not apply sufficient grip force or incorrectly aligned the fingers relative to the object's surface (as the fingers do not always align parallel due to irregular surfaces or miscalculations caused by data noise). The three-fingered gripper functioned properly when the objects' spherical or round nature was correctly identified.

Regarding unloading stacks of objects, the algorithm achieved an average transfer rate of 92.24% of objects. Typically, two objects out of 25 remained at the end. These objects, characterized by unsuccessful attempts, exhibited irregular shapes, lacked a flat surface, were small (approximately 1 cm tall), and often caused collision issues between the calculated gripping points by the two-fingered gripper and the surface. As a precaution against the risk of collision with the surface, the robot refrained from attempting to grasp these objects. In several instances, the segmentation failed to detect these objects, leading to the premature termination of the robot's automated operation. There were multiple unsuccessful attempts due to slight shifts in the object during gripping, yet subsequent attempts with refined grasp points often led to successful pickups.

5. Conclusions

Selecting the right gripper for objects and determining their order optimally works well, but there is always room for improving the algorithm's performance. An important factor to minimize potential errors involves developing a robot trajectory algorithm that is not automated by its software, as this might cause unforeseen movements or attempts to navigate the robot beyond its working area. Enhancing the approach angle of the robot's suction cup towards the object can reduce the risk of finger contact with the surface. If the angle is too steep, avoiding using the suction grip altogether might prevent finger impact with the surface. Although such an error did not occur during testing, there is a potential risk based on the robot's behavior and algorithm. Improving segmentation and placing the camera on the robot instead of statically above it could enhance the point cloud quality and reduce the data noise. Installing additional sensors on the fingers to verify if an object is gripped would simplify the estimation of grasp points, as it would not require pinpointing exact points on the object, only around it.

References

[1] S. Ainetter and F. Fraundorfer: End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13452–13458. IEEE, 2021. DOI: 10.1109/ICRA48506.2021.9561398

[2] S. Back, J. Kim, R. Kang, S. Choi, and K. Lee: Segmenting unseen industrial components in a heavy clutter using rgb-d fusion and synthetic data. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 828–832. IEEE, 2020. DOI: 10.1109/ICIP40778.2020.9190804

[3] A. Billard and D. Kragic: Trends and challenges in robot manipulation. *Science*, **364**(6446), (2019), eaat8414. DOI: 10.1126/science.aat8414

[4] L. Birglen and T. Schlicht: A statistical review of industrial robotic grippers. *Robotics and Computer-Integrated Manufacturing*, **49** (2018), 88–97. DOI: 10.1016/j.rcim.2017.05.007

[5] V. Blåsjö: The isoperimetric problem. *The American Mathematical Monthly*, **112**(6), (2005), 526–566. DOI: 10.2307/30037526

[6] T.J. Cairnes, C.J. Ford, E. Psomopoulou, and N. Lepora: An overview of robotic grippers. *IEEE Potentials*, **42**(3), (2023), 17–23. DOI: 10.1109/MPOT.2023.3236143

[7] H. Cao, G. Chen, Z. Li, Y. Hu, and A. Knoll: Neurograsp: multimodal neural network with euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Transactions on Instrumentation and Measurement*, **71**, (2022), 1–11. DOI: 10.1109/TIM.2022.3179469

[8] H. Cheng, Y. Wang, and M.Q.-H. Meng: A robot grasping system with single-stage anchor-free deep grasp detector. *IEEE Transactions on Instrumentation and Measurement*, **71**, 1–12, 2022. DOI: 10.1109/TIM.2022.3165825

[9] H. Cheng, Y. Wang, and M.Q.-H. Meng: A vision-based robot grasping system. *IEEE Sensors Journal*, **22**(10), (2022), 9610–9620. DOI: 10.1109/JSEN.2022.3163730

[10] M. Czubenko and Z. Kowalczuk: A simple neural network for collision detection of collaborative robots. *Sensors*, **21**(12), (2021), 4235. DOI: 10.3390/s21124235

[11] M. Czubenko, A. Chrzanowski, and R. Okuński: Instance segmentation of stack composed of unknown objects. *Engineering Applications of Artificial Intelligence*, **126** (2023), 106942. DOI: 10.1016/j.engappai.2023.106942

[12] A. Depierre, E. Dellandréa, and L. Chen: Jacquard: A large scale dataset for robotic grasp detection. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3511–3516. IEEE, 2018. DOI: 10.1109/IROS.2018.8593950

[13] G. Du, K. Wang, S. Lian, and K. Zhao: Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review*, **54**(3), (2021), 1677–1734. DOI: 10.1007/s10462-020-09888-5

[14] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel: On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, **29**(4), (1983), 551–559. DOI: 10.1109/TIT.1983.1056714

[15] N. Elangovan, L. Gerez, G. Gao, and M. Liarokapis: Improving robotic manipulation without sacrificing grasping efficiency: a multi-modal, adaptive gripper with reconfigurable finger bases. *IEEE Access*, **9** (2021), 83298–83308. DOI: 10.1109/ACCESS.2021.3086802

[16] C. Eppner, A. Mousavian, and D. Fox: Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227. IEEE, 2021. DOI: 10.1109/ICRA48506.2021.9560844

[17] G. Erboz: How to define industry 4.0: Main pillars of industry 4.0. *Managerial Trends in the Development of Enterprises in Globalization Era*, **761** (2017), 761–767.

[18] E. Gibney: The ai revolution is coming to robots: how will it change them? *Nature*, **630**(8015), (2024), 22–24. DOI: 10.1038/d41586-024-01442-5

[19] N. Guo, B. Zhang, J. Zhou, K. Zhan, and S. Lai: Pose estimation and adaptable grasp configuration with point cloud registration and geometry understanding for fruit grasp planning. *Computers and Electronics in Agriculture*, **179** (2020), 105818. DOI: 10.1016/j.compag.2020.105818

[20] K. He, X. Zhang, S. Ren, and J. Sun: Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016. DOI: 10.1007/978-3-319-46493-0_38

[21] M. Hegedus, K. Gupta, and M. Mehrandezh: Efficiently finding poses for multiple grasp types with partial point clouds by uncoupling grasp shape and scale. *Autonomous Robots*, **46**(6), (2022), 749–767. DOI: 10.1007/s10514-022-10049-6

[22] J. Hernandez, M.S.H. Sunny, J. Sanjuan, I. Rulik, M.I.I. Zarif, S.I. Ahamed, H.U. Ahmed, and M.H. Rahman: Current designs of robotic arm grippers: a comprehensive systematic review. *Robotics*, **12**(1), (2023), 5. DOI: 10.3390/robotics12010005

[23] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida: Soft manipulators and grippers: A review. *Frontiers in Robotics and AI*, **3**, (2016), 69. DOI: 10.3389/frobt.2016.00069

[24] B.R. insights. Centric grippers market report overview. Technical report, 2024. URL https://www.businessresearchinsights.com/market-reports/centric-grippers-market-103894

[25] V. Ivanov, V. Andrusyshyn, I. Pavlenko, J. Pitel', and V. Bulej: New classification of industrial robotic gripping systems for sustainable production. *Scientific Reports*, **14**(1), (2024), 295. DOI: 10.1038/s41598-023-50673-5

[26] R.A. Jarvis: On the identification of the convex hull of a finite set of points in the plane. *Information processing letters*, 2 (1): 18–21, 1973, DOI: 10.1016/0020-0190(73)90020-3

[27] E. Jezierski, P. Łuczak, P. Smyczyński, and D. Zarychta: Human–robot cooperation in sorting of randomly distributed objects. *Archives of Control Sciences*, **29** (2019). DOI: 10.24425/acs.2019.131228

[28] Y. Jiang, S. Moseson, and A. Saxena: Efficient grasping from rgbd images: Learning using a new rectangle representation. In *2011 IEEE International conference on robotics and automation*, pages 3304–3311. IEEE, 2011. DOI: 10.1109/ICRA.2011.5980145

[29] M.D. Johnson, S. Berman, R. Azen, W. Otieno, and N. Campbell-Kyureghyan: Robotization of industrial processes: Motivational differences between companies with and without existing robotic processes. *IISE Transactions on Occupational Ergonomics and Human Factors*, pages 1–14, 2023. DOI: 10.1080/24725838.2023.2278794

[30] N. Kaitwade: Robotic grippers market. Technical report, Newark, Delaware, United States, 2023. URL https://www.futuremarketinsights.com/reports/robotic-grippers-market

[31] L. Kang, J.-T. Seo, S.-H. Kim, W.-J. Kim, and B.-J. Yi: Design and implementation of a multi-function gripper for grasping general objects. *Applied Sciences*, **9**(24), (2019), 5266. DOI: 10.3390/app9245266

[32] L. Kang, Y. Yang, J. Yang, and B.-J. Yi: A three-fingered adaptive gripper with multiple grasping modes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6097–6103. IEEE, 2021. DOI: 10.1109/IROS51168.2021.9636758

[33] Z. Kowalczuk and M. Czubenko: Intelligent decision-making system for autonomous robots. *International Journal of Applied Mathematics and Computer Science*, (4), (2011), 671–684. DOI: 10.2478/v10006-011-0053-7

[34] Z. Kowalczuk and M. Czubenko: Cognitive motivations and foundations for building intelligent decision-making systems. *Artificial Intelligence Review*, **56**(4), (2023), 3445–3472. DOI: 10.1007/s10462-022-10255-9

[35] S. Kumra, S. Joshi, and F. Sahin: Antipodal robotic grasping using generative residual convolutional neural network. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9626–9633. IEEE, 2020. DOI: 10.1109/IROS45743.2020.9340777

[36] I. Lenz, H. Lee, and A. Saxena: Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, **34**(4-5), (2015), 705–724. DOI: 10.1177/0278364914549607

[37] G. Li, C. Fu, F. Zhang, and S. Wang: A reconfigurable three-finger robotic gripper. In *2015 IEEE International Conference on Information and Automation*, pages 1556–1561. IEEE, 2015. DOI: 10.1109/ICInfA.2015.7279534

[38] J. Li, T. Liao, H. Nigatu, H. Guo, G. Lu, and H. Dong: Under-actuated robotic gripper with multiple grasping modes inspired by human finger. *arXiv preprint arXiv:2403.12502*, 2024, DOI: 10.48550/arXiv.2403.12502

[39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie: Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. DOI: 10.48550/arXiv.1612.03144

[40] Z. Lu, H. Guo, W. Zhang, and H. Yu: Gtac-gripper: A reconfigurable under-actuated four-fingered robotic gripper with tactile sensing. *IEEE Robotics and Automation Letters*, **7**(3), (2022), 7232–7239. DOI: 10.1109/LRA.2022.3181370

[41] K. Makihara, Y. Domae, I.G. Ramirez-Alpizar, T. Ueshiba, and K. Harada: Grasp pose detection for deformable daily items by pix2stiffness estimation. *Advanced Robotics*, **36**(12), (2022), 600–610. DOI: 10.1080/01691864.2022.2078669

[42] D. Masłowski and M. Czubenko: System bezpieczeństwa dla współpracującego robota przemysłowego na bazie kamer głębi. *Pomiary Automatyka Robotyka*, **23**(4), (2019), 41–46. DOI: 10.14313/PAR_234/41

[43] O. Melikechi and D. B. Dunson: Ellipsoid fitting with the cayley transform. *IEEE Transactions on Signal Processing*, **72** (2023), 70–83. DOI: 10.1109/TSP.2023.3332560

[44] M.Q. Mohammed, K.L. Chung, and C.S. Chyi: Pick and place objects in a cluttered scene using deep reinforcement learning. *International Journal of Mechanical and Mechatronics Engineering*, **20**(04), (2020), 50–57.

[45] C. Müller: World robotics 2023 – industrial robots. Technical report, Frankfurt am Main, Germany, 2023. URL https://ifr.org/wr-industrial-robots/

[46] A. Orsula, S. Bøgh, M. Olivares-Mendez, and C. Martinez: Learning to grasp on the moon from 3d octree observations with deep reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4112–4119. IEEE, 2022. DOI: 10.1109/IROS47612.2022.9981661

[47] C.R. Qi, L. Yi, H. Su, and L.J. Guibas: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, **30** (2017). URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

[48] S. Ren, K. He, R. Girshick, and J. Sun: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, **28** (2015). URL https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[49] S. Rota Bulò, L. Porzi, and P. Kontschieder: In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. DOI: 10.48550/arXiv.1712.02616

[50] A.S. Sadun, J. Jalani, and F. Jamil: Grasping analysis for a 3-finger adaptive robot gripper. In *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 1–6. IEEE, 2016. DOI: 10.1109/ROMA.2016.7847806

[51] Z. Samadikhoshkho, K. Zareinia, and F. Janabi-Sharifi: A brief review on robotic grippers classifications. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–4. IEEE, 2019. DOI: 10.1109/CCECE.2019.8861780

[52] M. Suchi, T. Patten, D. Fischinger, and M. Vincze: Easylabel: A semi-automatic pixel-wise object annotation tool for creating robotic rgb-d datasets. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6678–6684. IEEE, 2019. DOI: 10.1109/ICRA.2019.8793917

[53] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox: Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021. DOI: 10.1109/ICRA48506.2021.9561877

[54] K. Tai, A.-R. El-Sayed, M. Shahriari, M. Biglarbegian, and S. Mahmud: State of the art robotic grippers and applications. *Robotics*, **5**(2), (2016), 11. DOI: 10.3390/robotics5020011

[55] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt: Grasp pose detection in point clouds. *The International Journal of Robotics Research*, **36**(13-14), (2017), 1455–1473. DOI: 10.1177/0278364917735594

[56] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield: Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018. DOI: 10.48550/arXiv.1809.10790

[57] M. Vohra, R. Prakash, and L. Behera: Real-time grasp pose estimation for novel objects in densely cluttered environment. In *2019 28th IEEE International Conference on Robot and*

Human Interactive Communication (RO-MAN), pages 1–6. IEEE, 2019. DOI: 10.1109/RO-MAN46459.2019.8956438

[58] Y. Wang, Y. Gu, and J. Shun: Theoretically-efficient and practical parallel dbscan. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD'20, page 2555–2571, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367356, DOI: 10.1145/3318464.3380582

[59] H. Wen, J. Yan, W. Peng, and Y. Sun: Transgrasp: Grasp pose estimation of a category of objects by transferring grasps from only one labeled instance. In *European Conference on Computer Vision*, pages 445–461. Springer, 2022, DOI: 10.1007/978-3-031-19842-7_26

[60] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser: Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018. DOI: 10.1109/IROS.2018.8593986

[61] B. Zhang, Y. Xie, J. Zhou, K. Wang, and Z. Zhang: State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review. *Computers and Electronics in Agriculture*, **177** (2020), 105694. DOI: 10.1016/j.compag.2020.105694

[62] H. Zhang, J. Tang, S. Sun, and X. Lan: Robotic grasping from classical to modern: A survey. *arXiv preprint arXiv:2202.03631*, 2022. DOI: 10.48550/arXiv.2202.03631

[63] T.Y. Zhang and C.Y. Suen: A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, **27**(3), (1984), 236–239,

[64] L. Zheng, W. Ma, Y. Cai, T. Lu, and S. Wang: Gpdan: Grasp pose domain adaptation network for sim-to-real 6-dof object grasping. *IEEE Robotics and Automation Letters*, 2023, DOI: 10.1109/LRA.2023.3286816

[65] Q.-Y. Zhou, J. Park, and V. Koltun: Open3d: A modern library for 3d data processing. *CoRR*, abs/1801.09847, 2018, DOI: 10.48550/arXiv.1801.09847. URL http://arxiv.org/abs/1801.09847

[66] M. Zhu, K.G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis: Single image 3d object detection and pose estimation for grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943. IEEE, 2014. DOI: 10.1109/ICRA.2014.6907430