

Article

# A Practical Approach for Picking Items in an Online Shopping Warehouse

Nataliya Nechyporenko <sup>1,\*</sup> , Antonio Morales <sup>1,\*</sup> , Enric Cervera <sup>1</sup>  and Angel P. del Pobil <sup>1,2</sup> 

<sup>1</sup> Robotic Intelligence Lab., Universitat Jaume I, 12071 Castellón, Spain; nataliya.nechyporenko@gmail.com (N.N.); eervera@uji.es (E.C.); pobil@uji.es (A.P.d.P.)  
<sup>2</sup> Department of Interaction Science, Sungkyunkwan University, Seoul 03063, Korea  
\* Correspondence: morales@uji.es

**Abstract:** Commercially viable automated picking in unstructured environments by a robot arm remains a difficult challenge. The problem of robot grasp planning has long been around but the existing solutions tend to be limited when it comes to deploy them in open-ended realistic scenarios. Practical picking systems are called for that can handle the different properties of the objects to be manipulated, as well as the problems arising from occlusions and constrained accessibility. This paper presents a practical solution to the problem of robot picking in an online shopping warehouse by means of a novel approach that integrates a carefully selected method with a new strategy, the centroid normal approach (CNA), on a cost-effective dual-arm robotic system with two grippers specifically designed for this purpose: a two-finger gripper and a vacuum gripper. Objects identified in the scene point cloud are matched to the grasping techniques and grippers to maximize success. Extensive experimentation provides clues as to what are the reasons for success and failure. We chose as benchmark the scenario proposed by the 2017 Amazon Robotics Challenge, since it represents a realistic description of a retail shopping warehouse case; it includes many challenging constraints, such as a wide variety of different product items with a diversity of properties, which are also presented with restricted visibility and accessibility.

**Keywords:** robotics; grasping; warehouse automation; manipulation



**Citation:** Nechyporenko, N.; Morales, A.; Cervera, E.; del Pobil, A.P. A Practical Approach for Picking Items in an Online Shopping Warehouse. *Appl. Sci.* **2021**, *11*, 5805. <https://doi.org/10.3390/app11135805>

Academic Editor: Manuel Armada

Received: 4 May 2021  
Accepted: 8 June 2021  
Published: 23 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Large depots that contain millions of different items are becoming more common as online retail services offer huge catalogs to worldwide potential customers. This demands novel solutions for automation on increasingly varied conditions. Robot picking is not an exception: incoming items need to be stowed, stored temporarily and later retrieved to attend to customer orders. The autonomous manipulation of a large variety of different manufactured products is still a major challenge that cannot be easily achieved with a unique solution, either a universal gripper or a single-grasp planning algorithm.

The problem of grasp planning has been around in the robotics community for a long time [1]; however, the existing solutions tend to be limited when it comes to deploying them in open-ended realistic scenarios. Practical picking systems are called for that can first cope with the different properties of the objects to be manipulated: different shapes, rigid vs. articulated vs. soft objects and textured vs. untextured vs. transparent surfaces. Second, if items are not stored separately but in clutters, then recognition and location systems have to deal with occlusions and constrained accessibility. These types of scenarios require the solution of complex sub-problems such as reliable object modeling, recognition and location, grasp and path planning in the presence of uncertainty and obstacles, robust execution and many others.

Nechyporenko and co-workers conducted a study comparing the AGILE (Antipodal Grasp Identification and Learning) and HAF (Height Accumulated Features) methods for robot grasping [2]. This paper builds on that preliminary work; here we focus on the

problem of planning the grasp and its execution, under the conditions mentioned above. *Grasp synthesis* refers to the problem of finding a grasp configuration that satisfies a set of criteria relevant for the grasping task [1]. The generation, evaluation and selection of grasps can be performed in different ways; in the following paragraphs we review the state-of-the-art on the topic of robot grasping but limiting its scope to approaches that are pertinent to picking objects from a container, such as a tote or a bin.

Task-based grasping has been separately studied in the context of Bayesian networks for encoding the probabilistic relations among various task-relevant variables [3]. The synthesis of the category and task has been performed based on 2D and 3D data from low-level features [4,5]. Instead of relying on sensor data points, another proposal is to synthesize grasps based on semantic content with the hope of yielding more stable grasps that are functionally suitable for specific object manipulation tasks [6,7].

Miller et al. used shape primitives such as spheres, cones and boxes to approximate object shape in the *GraspIt* simulation environment [8]. In another approach, supervised learning with local patch-based image and depth features was used for grasping novel objects in cluttered environments [9]. Height maps for the representation of features have been proposed for grasp planning with a reported 92% success rate for single object grasping and taking only 2–3 s [10,11]; the implementation on the Baxter robot is remarkable since it has a precision of only 1cm, and a simple gripper is used. Using features based on geometry, a combination of analytical and data-driven methods was also proposed [12]. Finally, another approach also uses features but instead of a 3D sensor, it relies on the supervised deep learning of 2D RGB images [13]; as is common for deep learning, this method requires both a large dataset and a long training time. A proposal to alleviate this last drawback has been recently put forward for a related problem in warehouse automation [14].

In recent years, a solid trend has emerged to apply machine and deep learning to deal with object recognition, grasp planning and other components of the *pick-and-place* pipeline. In some cases, convolutional networks are applied exclusively to image processing [15], whereas in other cases, learning techniques aim to obtain models that link visual perception to grasp planning. Some exhaustive techniques execute in simulation millions of grasps on artificially generated virtual objects in order to learn the relationship between shapes and successful grasps [16], or even try to connect simulated point clouds to promising suction placements on object surfaces [17,18]. Reinforcement learning approaches have also proved to be appropriate for this kind of applications. In some cases, demonstrated trajectories have been used as training data for the path planning of the arm [19,20]. In other cases, a visual or task success reward has been used to adapt the grasping strategies [21,22]. In general, learning-based approaches are very task specific or require extensive and exhaustive computational training.

Still, commercially viable automated picking in unstructured environments by a robot arm remains a difficult challenge. This paper proposes a new approach to this problem based on the experience gathered by our participation in two editions of the *Amazon Robotics Challenge*. Indeed, a preliminary version of our system [2] was used in combination with an object recognition module [23] for successful participation in the *Amazon Robotics Challenge 2017, (ARC'17)* [24]. As a testbed for the experiments described in this paper, we use the scenario defined by that edition of the competition, which includes realistic constraints of an online shopping warehouse.

The scenario of the *Amazon Robotics Challenge* has been used as a framework to test a number of approaches. Four specific grasping strategies were proposed depending on the shape of the object and the type of end effector—either gripper or suction cup [18]. Learning from demonstration was used to compute the reach-to-grasp action and heuristically suggest the best contact points [20]. D'Avella et al. used a Baxter dual-arm robot to pick objects from a box by means of a depth analysis of the RGB-D image of the scene, along with a custom designed jamming end-effector [25].

Our solution leverages state-of-the-art grasp planning methods integrated with a new ad hoc algorithm. The main contribution of this paper is a practical solution to the problem of robot picking in unstructured environments by means of a novel approach that integrates a carefully selected method with a new strategy, the centroid normal approach (CNA), on a cost-effective dual-arm robotic system with two grippers specifically designed for this purpose. Objects are matched to the grasping techniques and grippers to maximize success, and extensive experimentation provides clues as to what are the reasons for success and failure. In terms of the developed methods, the main contribution of our research is the creation and testing of the CNA algorithm that, using the point cloud and the major graspable component of the object, is able to find the centroid and its normals corresponding to the flattest part of the object's point cloud; then, grasps are rotated around the vertical z-axis so that the final grasp is most adequate for the robot's kinematics.

The paper is organized as follows. Section 2 describes our global system along with the tasks to be performed by this system in agreement with the specifications of the ARC'17. It has to be noted that ARC'17 was the last edition of the *Amazon Robotics Challenge*. The rationale to choose this scenario is that it has become a de facto benchmark in the robotics community. Section 3 describes in detail the grasp planning algorithms that constitute the core of our approach. Section 4 describes the experiments carried out to evaluate the system, and the results are discussed in Section 5. Systematic tests under the benchmarking conditions of the Amazon Challenge are described and discussed in Section 6. Finally, Section 7 summarizes our contributions.

## 2. System Description

Amazon Robotics aims to automate the task of customer order placement and delivery of its products. Amazon's automated warehouses successfully remove the walking and searching for the object, but automated picking still remains a difficult challenge. The *Amazon Robotics Challenge* was organized in order to spur the advancement of these fundamental technologies that, in the end, could be used at warehouses all over the world [26]. The challenge entrants could use their own robot hardware and software to attempt to solve somewhat simplified versions of the general tasks of picking and stowing items on warehouse shelves.

The challenge event consisted of two tasks: picking and stowing, first independently and then combined in a final task. The robots are scored based on how many items are picked and stowed, in a fixed amount of time, from a storage system into a box/bin and vice versa. Some items are known in advance, so they are available for experimentation and training beforehand; some other items are unknown, meaning that they are only shown to the teams some minutes before starting the actual challenge. After the competition is complete, the teams share and disseminate their approach to improve future challenge results and industrial implementations [27–31].

The items to be picked by the robot have been selected not only because of their common occurrence in warehouses and households but also because of their varied shape and nature. Figure 1 shows the full set of forty benchmark items from ARC'17; all these items are known in advance. In terms of grasping, the difficulty lies in item dimensions, texture and point cloud representation. Some items, such as the bath sponge can easily slip through the fingers of the gripper and others, such as the marbles, lets through vacuum air pressure. These complications call for algorithms and gripper combinations that are robust to changes in object orientation, shape and texture.

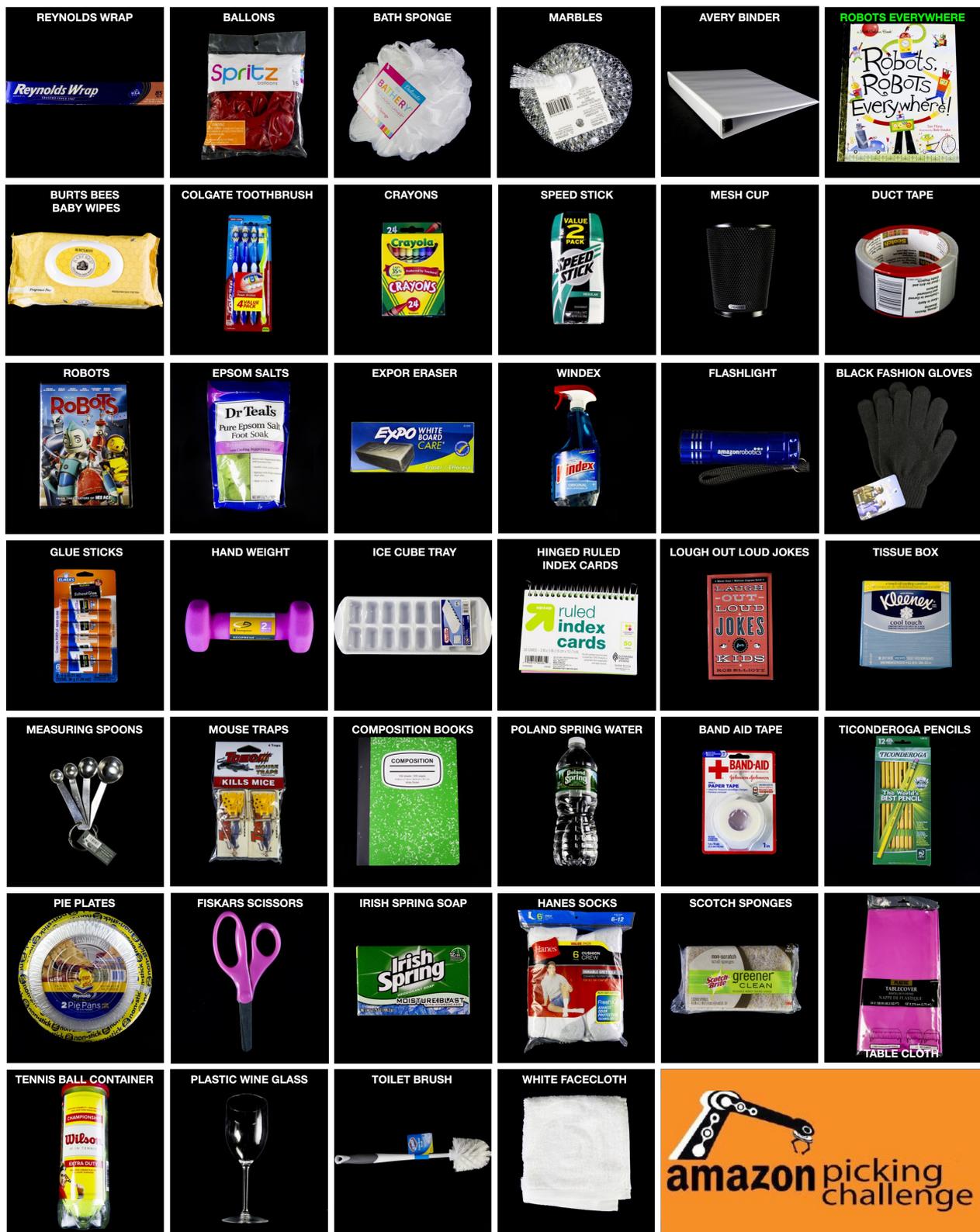


Figure 1. Official set of benchmark items from ARC'17. All these items are known in advance.

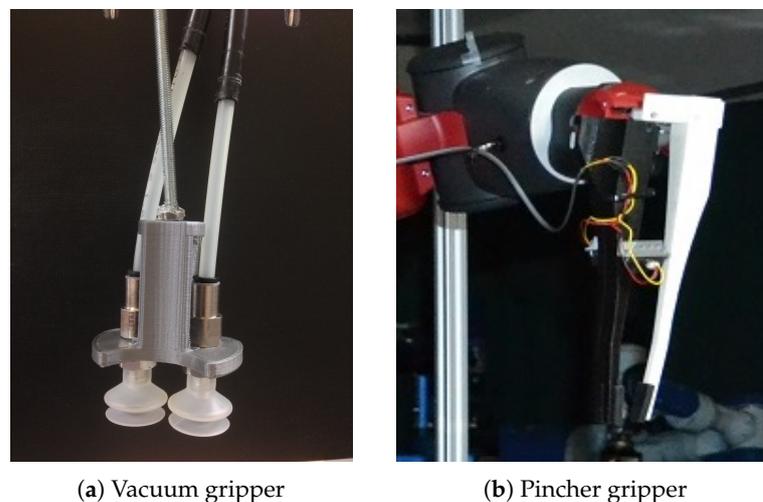
The RobinLab system is based on a dual-arm Baxter manipulator by Rethink Robotics with two custom-made grippers (Figure 2) and a shelving design based on a reliable industrial solution [24]. Bins can smoothly slide on a system of free-rotating rollers that are actuated by an external mechanism attached to the robot system. The purpose of this storage system is to allow a compact packing of the items on the bins and, at the same time,

provide a convenient access for the grippers and cameras from the top. The whole system setup can be seen in Figure 3.

The function of the vision system is to identify and localize the objects in the storage system in such a way that the grasping system can retrieve specific target items. The primary design goal was to converge into an accurate set of working algorithms for object recognition as well as creating a point cloud of one specific item. Different libraries used to accomplish these tasks are integrated into the system. The vision system diagram is shown in Figure 4; the hardware provides an RGB image from which objects are recognized, along with a depth image from which a point cloud is generated. The vision pipeline module handles the recognition of the objects by means of the combination of three methods (SIFT, SegNet and ResNet) in a way that maximizes the number of objects found and minimizes the number of false positives [23].

The ARC'17 task is in line with applications in a warehouse, production line, laboratory or household in which a robot has to analyze a scene in front of it and then manipulate objects. The robot has to safely operate in a restricted work space, requiring precise yet simple kinematic configurations to allow for predictability and operability. There is low visibility within the environment and the robot must handle objects in a clutter inside a small box. Given that the objects will be both known and unknown and the robot needs to operate in real time, the computation has to be performed quickly and efficiently. Low-cost convenient grippers with a fast-acting algorithm has to be paired for the highest success of object picking.

In addition, we added a vacuum system in order to offer alternatives when gripping is unreliable or not possible. This kind of mechanism has been presented with complete practical and theoretical analyses for other related approaches [27,30,31].



**Figure 2.** End effectors.

The input that the grasping module receives from the vision module is composed of the object that has been identified, along with its location as given by an approximate point cloud [23]. The grasping algorithm does not perform any further verification of the identity of the object. Its purpose is to compute the position and orientation of one or several grasp configurations that can be transferred to manipulation control.

The robot then has to move the arm in such a way that the terminal element ends up in the desired location to grasp the object. The two grippers that have been mounted on the Baxter robot are depicted in Figure 2. The first gripper is a vacuum gripper that uses air pressure to pick up objects. The second gripper is a two-finger gripper with a limited opening width, that has been named the *Pincher*. The goal now is to develop a grasp planning approach that optimizes the performance of these grippers in a warehouse environment.

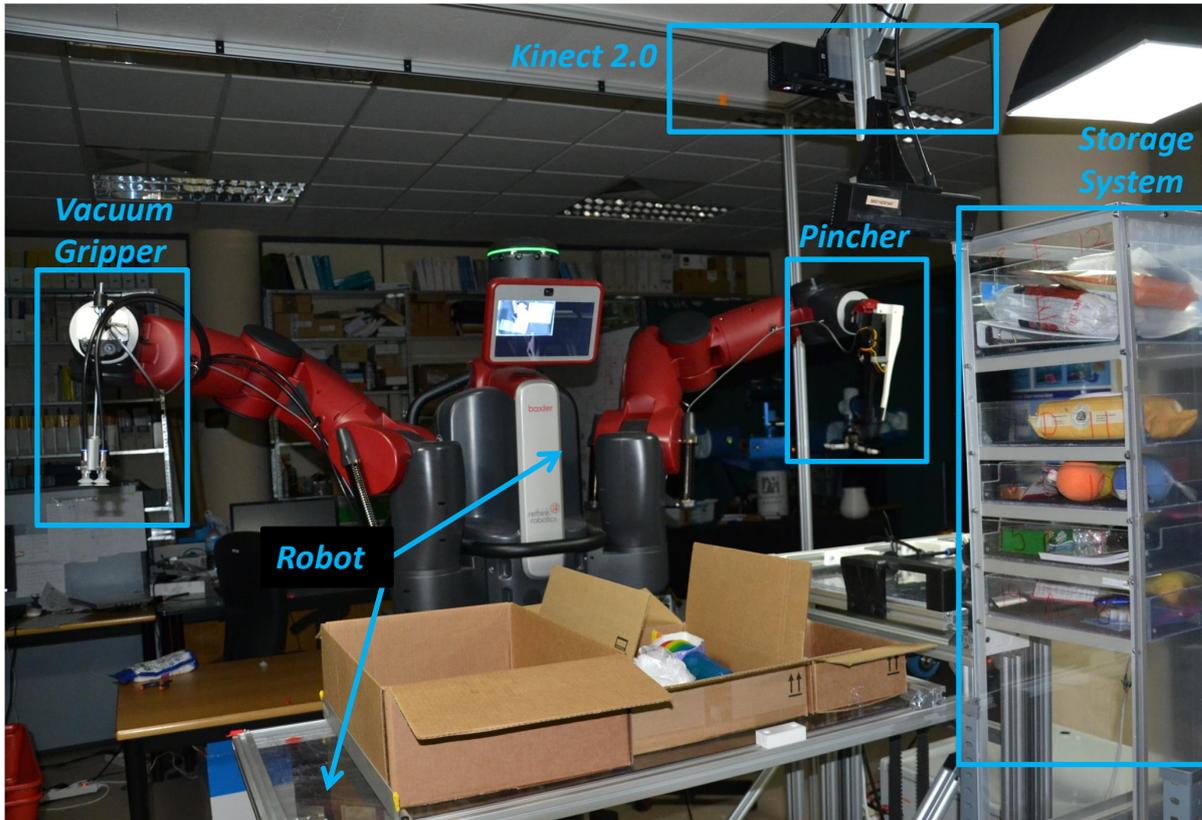


Figure 3. UJI RobinLab robot platform setup.

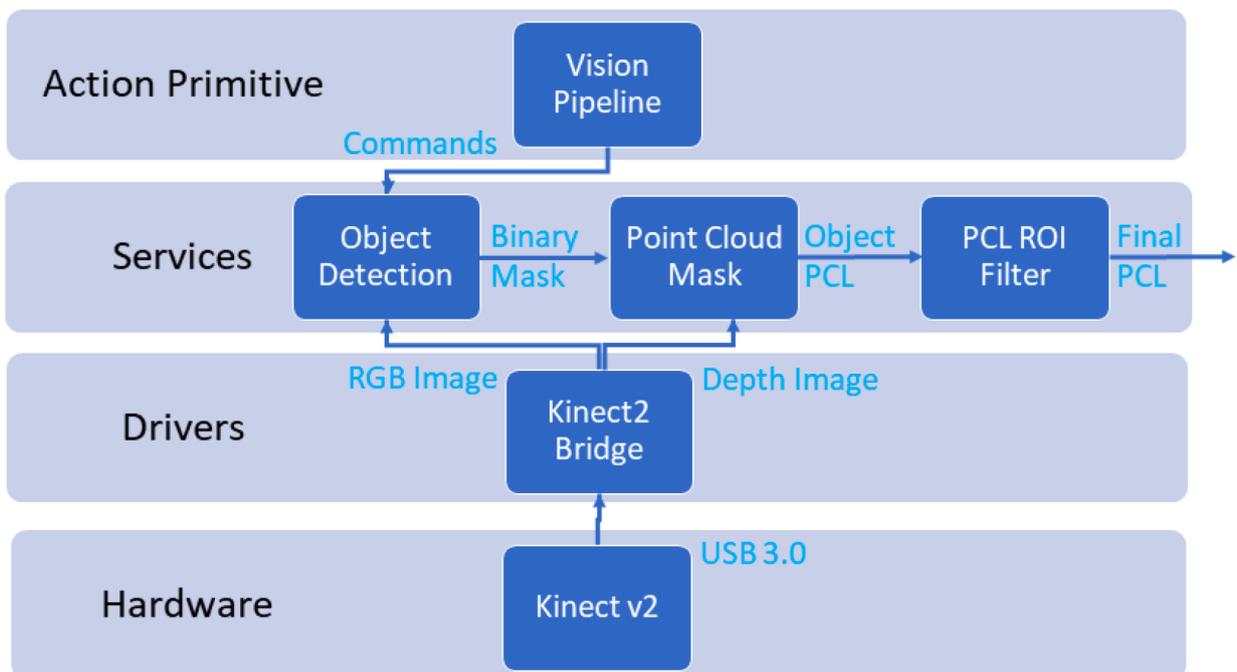


Figure 4. Vision system diagram.

### 3. Grasp Planning

Two state-of-the-art algorithms for the two-finger gripper have been selected, analyzed and compared. The most suitable one was subjected to further testing based on the analysis of time, robustness and success.

In this section we provide theoretical knowledge for the two algorithms that were considered. From a bird's eye perspective, the separate algorithms have their individual environments in which each one can exhibit its strength. HAF takes into account the height of the objects and is used with a top grasp thereby reducing the dimensionality of the problem [10,11]. AGILE (antipodal grasp identification and learning) explores the geometry of the whole object in order to find handle-like sections to exploit for grasping and thus it often aims at side grasps [12]. These feature-based algorithms can easily be compared with our proposed method, CNA (centroid normals approach), which is computationally less heavy and thus faster and also more adept at grasping flat objects such as books or folders for which a two-finger gripper can only succeed with a side grip [32].

#### 3.1. Height Accumulated Features (HAF)

As the name suggests, the HAF algorithm utilizes the heights of surface points, gathered from the point cloud data, relative to their neighbors in order to learn how to grasp the objects. The authors stress three important advantages of the algorithm; *segmentation independent*, *integrated path planning* and use of *known depth regions* [10,11].

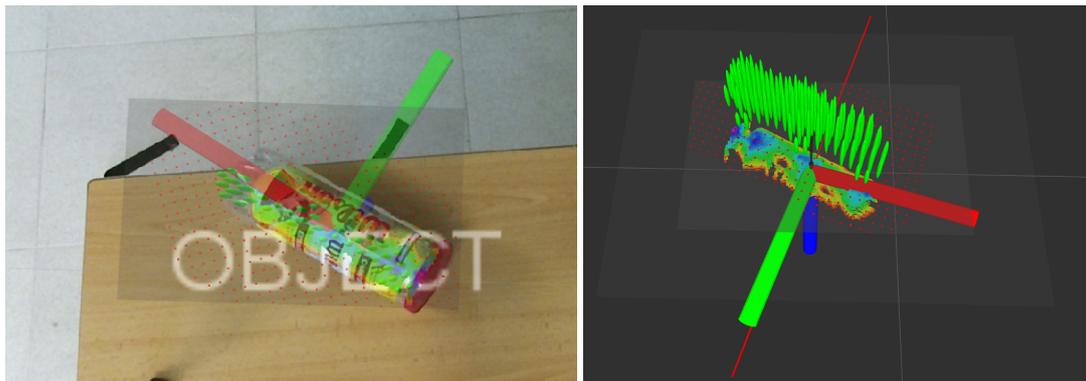
The term *height* refers to the measure of the perpendicular distance from the table plane to the points on the top surface of the object. The input point cloud is first discretized and the height grid  $H$  now contains a  $1 \times 1 \text{ cm}^2$  cell that saves the highest z-valued points with corresponding x and y values [33]. HAF features are defined similarly to Haar Basis functions. All height grid values of each region,  $R_i$ , on a height grid  $H$ , are summed up. The sums  $r_i$  are individually weighted by  $w_i$  and then summed up. The regions and weights are dependent on the HAF feature that are defined by an SVM classification. A feature value,  $f_j$ , is defined as the weighted sum of all regions. The  $j$ th HAF value  $f_j$  is calculated as:

$$f_j = \sum_{i=1}^{nrRegions_j} w_{i,j} \cdot r_{i,j} \quad (1)$$

$$r_{i,j} = \sum_{k,l \in \mathbb{N}: H(k,l) \in R_{i,j}} H(k,l) \quad (2)$$

The paper claims to have tested 71,000 features (70,000 of which are automatically generated) and finally selected 300 to 325 with an F-score selection [11]. Figure 5 visualizes the calculations in process for our implementation of the HAF algorithm; it corresponds to one of the objects in ARC'17.

The green bars indicate the identified potential grasps, whereas the height of the bars represents the grasp evaluation score. The frame (green, red and blue segments) represents the final grasp hypothesis chosen by the algorithm and indicates the final position where the end effector should go.

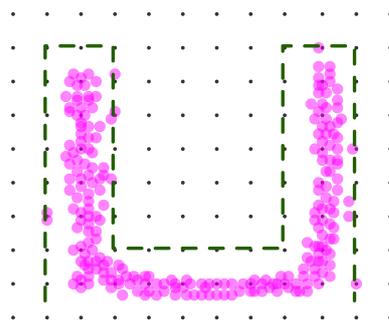


**Figure 5.** HAF visualization for the tennis ball container: (left) view from the Kinect 2.0; (right) view from Baxter.

### 3.2. Antipodal Grasp Identification and Learning (AGILE)

AGILE grasping is an algorithm that uses a point cloud to predict the presence of geometric conditions that are indicative of good grasps on an object [12]. First, geometry is used to reduce the size of the sample space by applying the conditions required for a grasp to exist: the hand must be collision-free, part of the object surface must be contained between two fingers and the grasp is antipodal. A pair of point contacts with friction is *antipodal* if and only if the line connecting the contact points lies inside both friction cones [34]. Then, the remaining grasps are classified using machine learning; geometry is used in order to automatically label the training set.

Grasp geometry is quantified by certain parameters. The reason why this algorithm is easy to implement is that these parameters are easy to tune depending on the dimensions of the two-finger gripper. The gripper is specified by the parameters  $\theta = (\theta_l, \theta_w, \theta_d, \theta_t)$  which, respectively, stand for gripper length, width, the distance between two fingers and the thickness of fingers. The method relies on features: classification of hand hypothesis uses a feature descriptor of a hand hypothesis as seen in Figure 6. In the histogram of gradients (HOG) feature descriptor, the distribution (histograms) of directions of gradients (oriented gradients) are used as features. Gradients (x and y derivatives) of an image are useful because the magnitude of gradients is large around edges and corners (regions of abrupt intensity changes) and edges and corners pack in a lot more information about object shape than flat regions [35].



**Figure 6.** An example of the HOG feature representation Adapted from [36].

### 3.3. Centroid Normals Approach (CNA)

The idea behind the new centroid normals approach (CNA) method comes from the observation that many manufactured objects are symmetric and have a large central surface that is most suitable for creating an air sealed grasp. The main idea is to receive a point cloud and downsample it using a voxel grid. Then, a cylinder or a plane is extracted, depending on the most prominent object shape. Finally, using the extracted shape, grasps located in its center are found by using surface normals and Euler to quaternion rotations. Quaternions are used for convenience with the purpose of simplifying the analysis of the

orientation of an object that can be either lying on a flat surface or leaning against a wall. Figure 7 shows the logic flow of the approach.

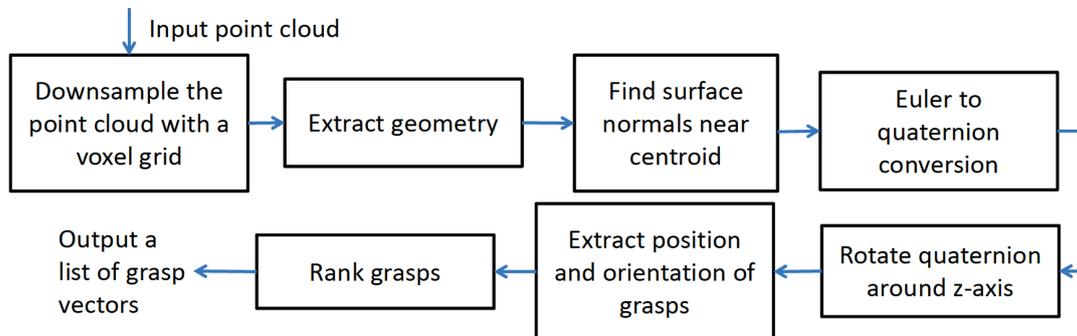


Figure 7. CNA logic flow.

The PCL library contains the following options for creating a model from a set of point cloud data points: random sample consensus (RANSAC), least median of squares (LMEDS), M-estimator sample consensus (MSAC), randomized RANSAC (RRANSAC), randomized MSAC (RMSAC), maximum likelihood estimation sample consensus (MLESA) and progressive sample consensus (PROSAC). A thorough comparison based on accuracy, computing, time and robustness was performed between RANSAC and its descendants as well as other consensus models by [37,38]. Based on the analysis of this work, PROSAC has been chosen for implementation.

Once the relevant shape has been extracted, which would be planar (in the case of a book) or cylindrical (e.g., for the tennis ball container), the algorithm finds the centroid or the 3D average of all the points fed into it. Then it calculates the surface normals closest to the centroid.

### 3.3.1. Kinematic Constraints

When the grippers were mounted onto the Baxter robot in such a way that they are perpendicular with respect to the wrist, the twist of the gripper was moved from joint W2 to joint W1: Figure 8 shows the joints of the Baxter robot, the original configuration (using W2 for twisting around Z-axis) and the chosen configuration (using W1).



Figure 8. Configuration of the wrist joints: (left) Baxter joints; (center) standard configuration using W2 for Z-axis twist; (right) our configuration using W1 for Z-axis twist.

The restriction of twist means that the angular roll along the z-axis of the gripper is also restricted. Table 1 summarizes the angular range in each joint. Note that the angular motion has been reduced by 140.5 degrees by reorienting the gripper as seen on Figure 3. Outside of the movement range, the robot will not be able to move and the inverse kinematic (IK) solution will not be found.

**Table 1.** Joint range comparison.

Gripper Orientation	$lim_1$ (deg.)	$lim_2$ (deg.)	Range (deg.)
Classical	175.25	184.75	350.50
UJI RobinLab	120.00	270.00	210.00

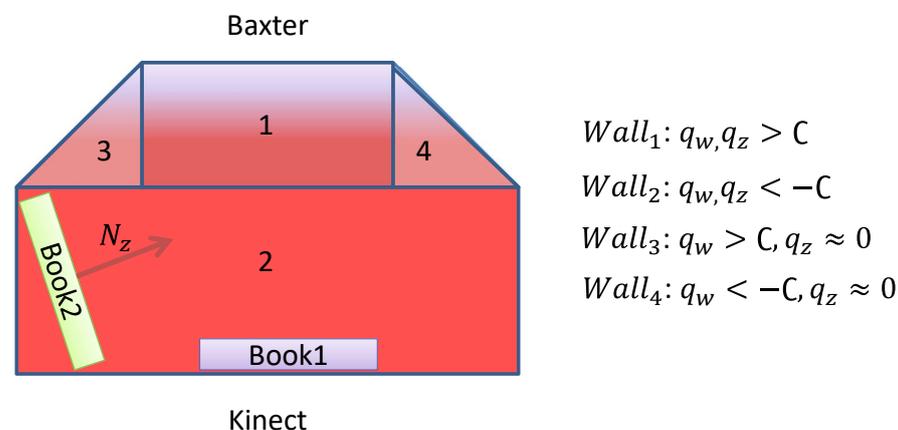
Although this decision was made in order to increase the number of possible vertical grasps where the z-axis is pointing directly downwards, the sacrifice was the limitation on the gripper. The position of the gripper on the robot arm poses two important challenges:

1. If the orientation of the object to be grasped requires the roll of the arm to be within the impossible limits of the wrist joint, then the object cannot be grasped although the grasp vector can be deemed to be successful.
2. The final position and orientation of the grasp determines greatly the position of the arm and its configuration. The grasps should try to put the robot arms similar to that when it is in the “tuck-arms” position.

### 3.3.2. Cluttered Scenario in the Stowing Task

For the pick task the idea is to have the robot approach as vertical as possible with a predefined rotation about z-axis to meet challenge 1 above and successfully pick up the identified item for a particular order. However, the stow task is different since the objects might be tightly pressed against the wall, hence further calculations need to be performed to meet challenge 2 specified above (compare the pose of *book2* and *book1* in Figure 9). The tote has been divided into South ( $W_1$ ), North ( $W_2$ ), East ( $W_3$ ), West ( $W_4$ ) and Planar grasp orientations, which originate from the sections of the tote as seen on the figure. The idea behind this classifications comes from the result of the normals.

The normals provided by the point cloud library are vectors that have the z-axis (normal) to the object however the x and y-axis are random based on the computation and reconstruction of the surface. If the robot sees a normal to the object to be sufficiently vertical and sufficiently close to the wall, then the grasp vector will resort to the predetermined arm position in the location specified. Four basis frames have been established, going out about  $45^\circ$  from the surface of each wall (such as  $N_z$ ). From the basis frames the robot has an option of  $+/- 30^\circ$  about the normal. These predetermined angles guarantee an IK solution. The way to find out which wall is being addressed has been achieved through statistical analysis of the components of the quaternion,  $q = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}$ . Knowing the location and threshold  $C$  of how tilted the grasp needs to be to transfer from Planar to a Wall position, the conditions shown in Figure 9 will classify roughly estimates the orientation of the normal within the bin.

**Figure 9.** Labeling of the tote.

Two rotation quaternions can be combined into one equivalent quaternion by the relation:  $q' = q_2q_1$ . In which  $q'$  corresponds to the rotation  $q_1$  followed by the rotation  $q_2$  ans is used to obtain rotations about the normals to the wall. Another role is the location within the tote. The tilted grasp would be sent to the gripper only if its tilt is not too close to the walls. This appears in the algorithm as a simple threshold.

### 3.3.3. Grasp Selection

Finally, a scoring algorithm has been devised in order to prioritize grasps. The grasp score  $S_i$  is calculated as follows. Knowing the pose of all the grasps, first find the average of all the  $x$  and  $y$  points, then subtract the  $x_i$  and  $y_i$  values of the current grasp from the averages,  $x_{av}$  and  $y_{av}$  to find how far away it is from the center.  $x_{dist} = |x_{av} - x_i|$  and consequently  $y_{dist} = |y_{av} - y_i|$ . The next in consideration is the  $z$ , which will be subtracted from the previous score. The  $w$  values are scalars that can adjust how much each score should matter. In the case applied for our system,  $w_x = w_y = 0.7$  and  $w_z = 0.3$ . These values have been empirically determined based on our experiments.

$$S_i = (w_x, w_y) \cdot (x_{dist}, y_{dist}) - (w_z) \cdot (z_i) \quad (3)$$

The right-hand side of Figure 10 shows the input point cloud of another object in the ARC'17 data set (the mesh cup). The left side shows the same point cloud but with the violet dots showing the post-processed point cloud; the yellow lines show the surface normals; the green circle shows the  $x$  and  $y$  position of the point cloud centroid; the purple central surface normals around the centroid correspond to the CNA vectors having the relevant  $z$ -axis that will be used for processing the outputted grasps. The frame on the right side shows one of the final grasp frames for the end effector.

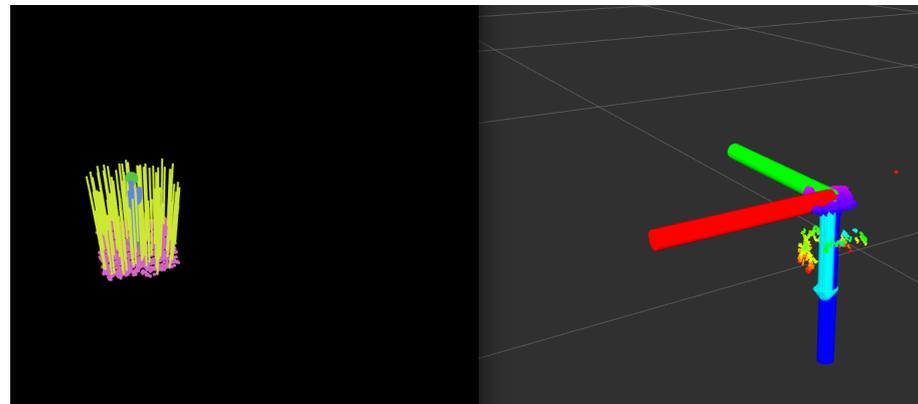


Figure 10. CNA algorithm visualization for the mesh cup.

## 4. Experimentation and Results

### 4.1. Experimentation Procedures

This section presents the experimental setup for testing the grasping algorithms as well as the results. During the pick and stow tasks, the grasping algorithms receive a single point cloud of an object to be manipulated. The algorithm for grasping has to output a suitable grasp vector. During the stow task, the tote contains 20 objects in a mixed jumble. The vision pipeline, in this case, does not guarantee the segmentation of a single object and can include parts of other objects. These noisy data require the algorithms to be robust. The grasp must also be calculated quickly and successfully. Knowing which algorithm performs best under which circumstance is key to devising a final structure that can pick up the maximum amount of objects.

#### 4.2. Preliminary Object to Gripper Matching

Before applying the testing of algorithms it is important to know which gripper works well with which object. In a perfect scenario, the best grasp algorithm will output a similar result to what a human would choose.

The procedure for finding out whether the object can be grasped by the vacuum or two-finger gripper consists of an empirical test, by guiding manually the Baxter robot and trying to grasp the object with each gripper:

1. Guide the Baxter robot and approach the item with the gripper.
2. Attempt to lift the gripper along with the item.
3. If the item can be lifted easily then it is graspable by the given gripper.
4. If the object falls down or slips then the object cannot be grasped by the given gripper.

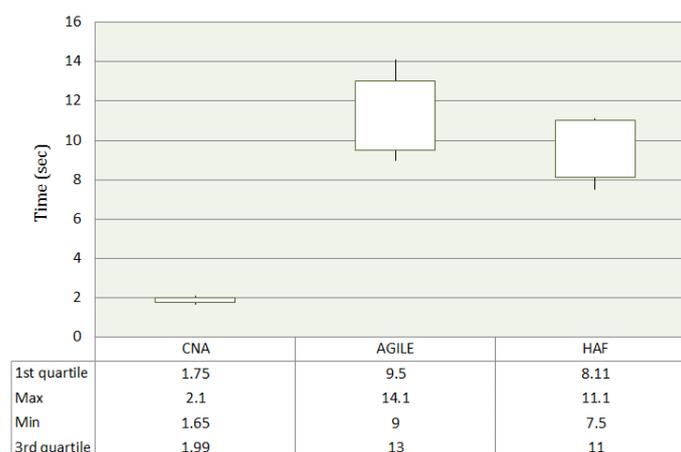
This empirical test allows the task planning procedures to know which arm to use for object manipulation. If the object can be lifted using vacuum pressure on most sides of the object then the right arm will be used with suction. If the object is best grasped with the gripper then the left arm will be used to manipulate the object.

#### 4.3. Preliminary Implementation of Algorithms

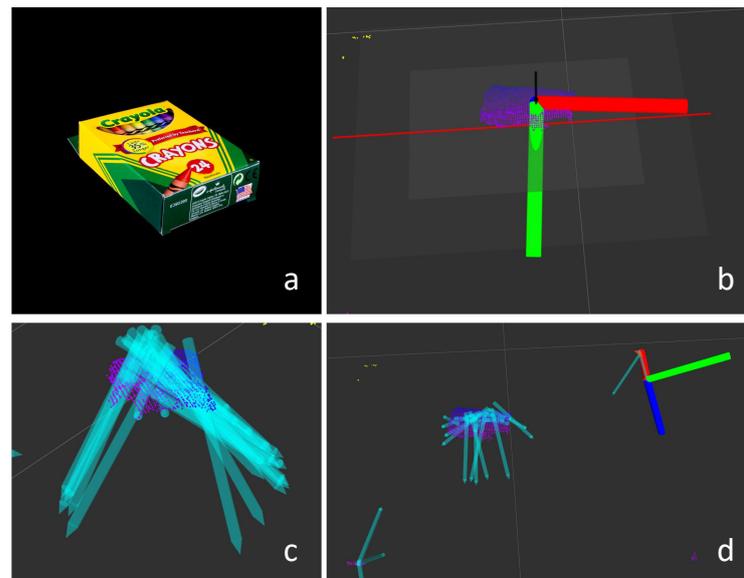
For object manipulation, the algorithm options are either AGILE, HAF or CNA. The AGILE and HAF grasping are used with the same gripper and produce similar results. In order to understand the potential of the grasping algorithms, it is first important to implement them and obtain results to see whether the algorithm is capable of dealing with the point cloud presented by the vision pipeline. The goal of this testing is to see how the algorithm behaves and fits once it is integrated into the whole system.

It is important to see the output of the final obtained approach vector as well as the time taken to perform the calculation of the given vector or vectors. The execution time has been computed with the open multi-processing library (OMP).

Figure 11 shows the approximate calculation time of each algorithm. With this information further steps have been taken to simplify the grasping optimization. Furthermore, Figure 12 shows the qualitative analysis of robustness.



**Figure 11.** Time each algorithm takes to calculate a set of grasps, based on 10 function calls for 10 different objects.



**Figure 12.** Robustness comparison of AGILE and HAF for the crayons box (a). Results for HAF (b) and AGILE (c,d) are shown.

#### 4.4. Discarding AGILE

In Figure 11 it can be seen that the AGILE grasping algorithm takes up to 14 s to calculate, HAF takes up to 11 while CNA takes up to 2 s. This is a huge drawback to computing height and geometrical features since the time-per-object increases greatly. In order to minimize time and thus maximize the performance of the system, we decided to limit the number of objects that will be used with the HAF or AGILE grasping algorithms. If an object has an option of being picked with CNA then this algorithm has the priority.

After a few preliminary testings with time, it was determined that having both AGILE and HAF grasping to be applied to the same gripper increases complexity and adds redundancy. From the first step, it is known that AGILE takes more time. The second important step is determining the robustness and success of the algorithms. As an illustrative example, Figure 12 shows the case of the crayons box. From its point cloud HAF computed a downward facing grasp vector as seen in Figure 12b. The grasp is unique with limited variations of twist around any other axis but it is stable and the result is repeatable. By choosing the vertical approach rather than the one at an angle, the robot is less likely to collide with the bin or with the tote. This saves time for not executing grasps that are likely to fail; it also shows robustness since the grasps with low scores often result in a polluted point cloud and are eliminated.

AGILE grasping, as can be seen in Figure 12c, presents the result with many possibilities. The possibilities are often at an angle away from the vertical z-axis since the edges of objects have more geometrical features. Unfortunately, the gripper often cannot pick up an object from the sideways position since there are no feasible non-friction fingers to be able to do so. AGILE grasping also takes into consideration the many parts of the point cloud that are not relevant, as can be seen in Figure 12d; these grasps can potentially cause a collision if executed.

Table 2 shows the strengths and weaknesses of the methods in comparison with each other. It was concluded for further testing to discard the AGILE grasping from the system architecture as the lack of robustness reduces success and increases the possibility of whole system failure.

**Table 2.** Comparing AGILE and HAF.

Criteria	AGILE	HAF
Time	–	+
Robustness	–	+
Success	+	+

#### 4.5. Testing Objects in Isolation

Testing objects in isolation is relevant for the problem since the point cloud given from the vision pipeline is an imperfect point cloud of a single object. The evaluation was performed based on 10 grasping attempts, each of which was scored on a binary 0/1 system to mark whether the object has been grasped and lifted or not. The setting of grasping is exactly as would be during the competition and in the warehouse. If the point cloud or hardware is not perfect, then the algorithm has to be able to deal with this. The scenario is not fully controlled but rather as realistic as possible. The successful grasps were added up and the  $\frac{\text{success}}{\text{num.attempt}} \times 100\%$  was calculated. For demonstration videos of the robot grasping, please visit <https://vimeo.com/grasps> accessed on 4 May 2021. The experiment ran according to the following rules:

1. In principle, all objects originating from the ARC'17 competition, can be grasped by the gripper in their respective category.
2. The point cloud is as provided by the vision pipeline. No changes are made to improve how the vision system perceives the object. The testing is performed with full integration of the whole system architecture.
3. The environment is exactly as the system would predict it to be in a real-world setting. A failure of gripper arm orientation or positioning is considered a failure for grasping.

In the case of HAF most failures are due to the grasp not being reachable, whereas CNA fails mostly due to the object slipping/dropping from the gripper, or not being properly fixed to the vacuum cups.

#### 4.6. Object to Algorithm Classification

Next, given a set of repeated testing and analysis of success, Table 3 was created to show which algorithm was used for which object. This table maximizes the success of the object picking. Number 1 stands for the fact that the first try is to be attempted with the algorithm in that column, and number 2 stands for the second try and this algorithm has the second priority. This has been performed because an object such as the duct tape can be picked up with either the vacuum (vac.) or the two-finger gripper (grip.) depending on its orientation.

**Table 3.** Optimization table for ARC'17 items.

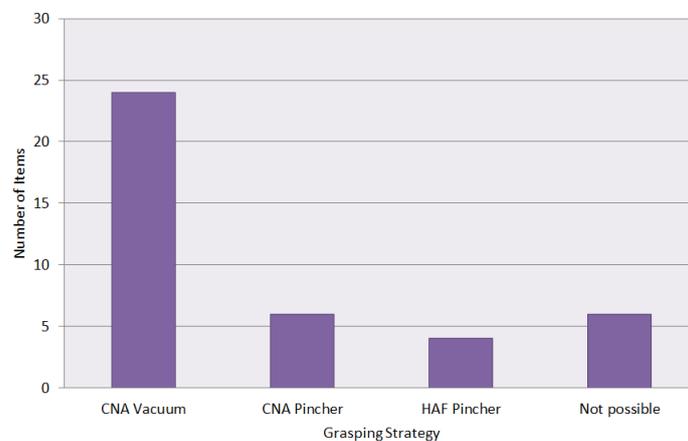
#	Object	CNA Vac.	CNA Grip.	HAF Grip.
1	avery_binder	1		
2	balloons		1	
3	burts_bees_baby_wipes	1		
4	toilet_brush		2	1
5	colgate_toothbrush_4pk	1		
6	crayons	1		
...	...	...	...	
40	flashlight	1		
	Total	29/40	8/40	5/40

## 5. Evaluation

### 5.1. Maximizing Use of CNA

As can be seen, there was a maximization of the use of the CNA algorithm, first of all, because it is successful, second because it is fast and third because it is not restricted by the workspace of the robot arm.

After an extensive amount of testing, the final optimized table has been created in order to maximize success for grasping with various algorithms and two grippers. The CNA algorithm works because it takes the center of the object given the point cloud and exploits the consistent texture quality of objects, such as the gloves. These deformable objects can be grasped in the same way that a book is grasped. There is no difference in computation and the hardware of the Pincher adapts well to the object. Note that some objects can be picked up with both grippers depending on which way the orientation is or which part of the object is visible. In this case, the robot is programmed to try twice with different algorithms. With this exploitation, it can be seen in Figure 13 that over 75% of the objects will be picked up using the CNA algorithm, whether with vacuum or with Pincher.

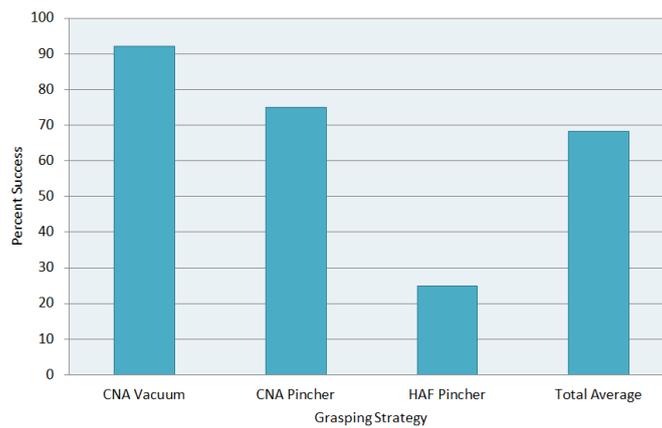


**Figure 13.** Number of items per grasping strategy.

### 5.2. Failure Analysis

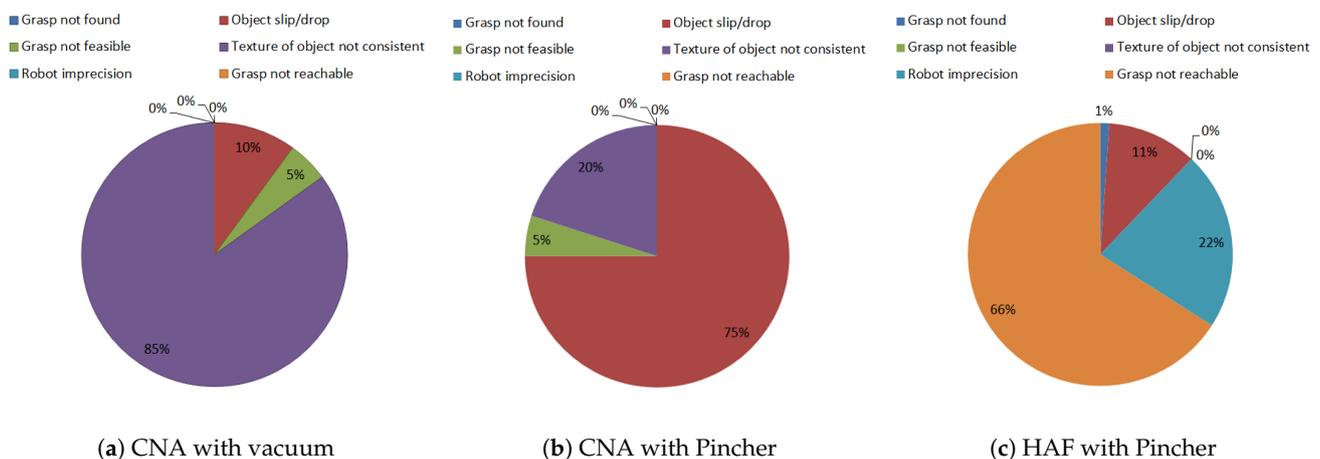
Figure 14 shows that CNA with vacuum is the most used and is also the most reliable approach. So if it is deemed that vacuum is most useful for a certain object then there is over 90% chance that it will be successfully picked up. The set of pie charts provides important information as to what happened during testing with the unsuccessful grasps. The pie charts document the reasons for failure. The most notable reasons relevant to grasping that were seen during testing are listed below.

- Grasp not found: The algorithm does not output a grasp based on the input point cloud.
- Grasp not feasible: The executed grasp will not result in lifting the object.
- Robot imprecision: The grasp is feasible and can be executed; however, the robot misses the location of grasp due to calibration or actuation errors.
- Object slip/drop: Object is grasped but then slips between the fingers or escapes from the vacuum pressure.
- Texture of object not consistent: The grasp from the point cloud looks potentially successful; however, a label or a bump in the object obstructs the lifting of the object.
- Grasp not reachable: The restrictions to robot's wrist motion deem the grasp not executable.



**Figure 14.** Percent success per grasping strategy with isolated objects; each success rate has been determined for the corresponding subset of objects. The total average is weighted with the percentage of items per strategy, including the 6 blacklisted not\_possible items.

In the case of CNA with vacuum, the main reason, as seen in Figure 15a, is the inconsistency of object texture. For example, the pie plates item has a back side that is perfectly flat and has 100% chance of being picked up; however, testing was performed on all sides and the other side is curved and air pressure slips between the holes. On the other side, the pie plates item has been picked up 0/5 times; as such, the total success can only be 50%. Figure 15b shows the reason behind the failure of CNA with Pincher. The main reason is that the object slips. The material of an object such as table cloth is very thin and thus drops unexpectedly. Another major reason is that the texture is inconsistent. An object such as black fashion gloves has a label and if one of the fingers touches this label then it glides across the label without grabbing any of the material. Note that in the above case the CNA algorithm is successful despite the hardware restrictions. This is due to the preassigned angular rolls around the z-axis of the robot wrist. This technique in CNA allows for the configuration to always be comfortable for Baxter and the IK to always find a solution. Hence the strength of this algorithm is its independence of the orientation of the wrist. Figure 15c shows the reasons for the failure of the HAF algorithm with Pincher. This combination shows a low success rate of 25%. The reason behind this low success rate can be attributed to the hardware restrictions: a grasp can always be calculated, even if it is not the best, but it can rarely be executed. Another reason, especially relevant for the hand weight item, is robot imprecision: the Pincher opening allows about a millimeter of clearance with the center of the hand weight and the robot has 1cm accuracy, hence the robot simply misses the correct grasp location.



**Figure 15.** Reasons of failure for the different methods.

The conditions for testing have been as realistic as possible. It is very important to evaluate a grasp in this perspective because grasping criteria often only results in success with regards to the object, not to the scene. Despite the setback described above, the total success rate, as shown in Figure 14, shows a 69% success for object grasping.

To illustrate the results, Figure 16 shows the successful grasps of three different items with CNA and the vacuum gripper, Figure 17 shows other successful grasps of flexible and deformable items with CNA and the Pincher, and, finally, a successful grasp of one of the difficult items (the toilet brush) is achieved with HAF and the Pincher (Figure 18).



Figure 16. CNA with vacuum; successful grasp of the Reynolds' wrap, tennis ball container and Irish spring soap.

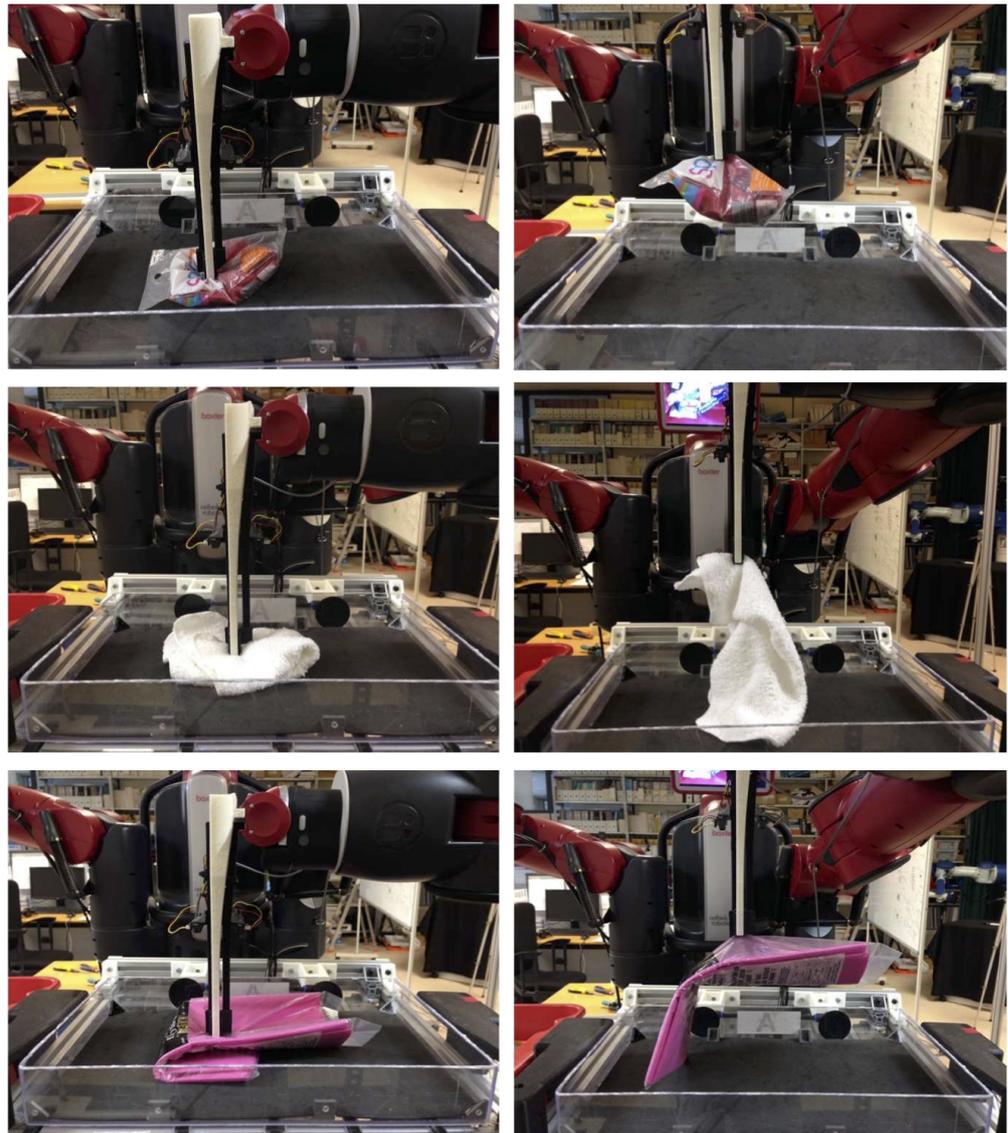


Figure 17. CNA with Pincher; successful grasp of balloons, face cloth and table cloth.

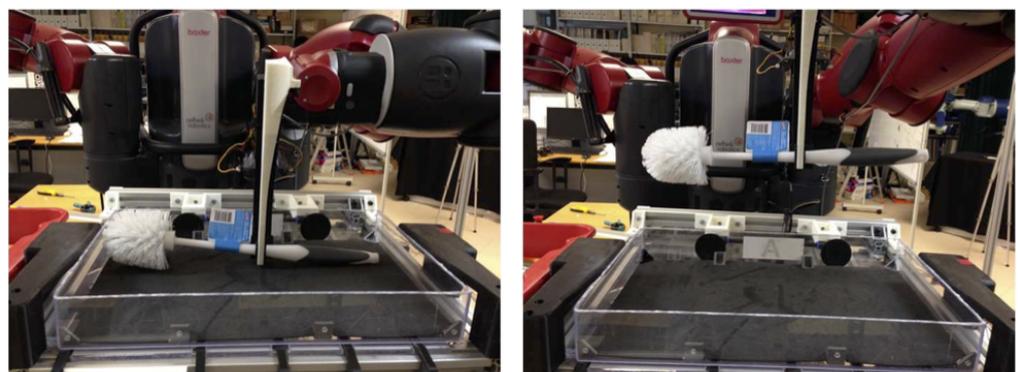


Figure 18. HAF with Pincher; successful grasp of the toilet brush.

## 6. System Performance Tests Based on ARC'17 Benchmark

Eight tests with different setups were performed in order to test the system performance in the picking task, replicating the conditions of the ARC'17 as a benchmark. For each test, 10 target items, distributed among the 5 bins of the storage system, had to be

identified by the vision system, and a correct grasp had to be planned to pick them up. In each test, the 32 items in the bins were a combination of known items (from the set of 40 objects shown in Figure 1) and unknown items. Since these unknown items were not provided by Amazon, we used the 12 items shown in Figure 19 as our unknown set; as in the competition, they are similar, but different, to some of the known items. Further, they were only available to the system 30 min before each test.

The first two tests are explained in detail below to illustrate the testing conditions and make observations about the system. The information gathered from all the tests is summarized later on.



Figure 19. Set of unknown items. These items are only available 30 min before each test.

6.1. Test 1

The initial setup for this test can be seen in Figure 20. The 10 target items to be grasped are as follows: In bin A there is the unknown target item ‘big duck’, and the known ‘glue sticks’ and ‘balloons’. In bin B, there is the unknown ‘glue’. In bin C, both the ‘crayola cera’ and ‘Adventures of Huckleberry Finn’, which are unknown. In bin D, the ‘glasses’ and ‘brush’ are unknown, and the ‘bath sponge’ and ‘windex’ are known. There are no target items in bin E, therefore it is ignored for this test.



Figure 20. Initial setup for Test 1.

In this case, all the target objects were successfully picked up with the exception of the 'big duck' item that was not recognized by the vision subsystem. The reason was that it was overlapped by the 'small duck' item that was flipped over it by the Pincher gripper when the 'balloons' were picked up.

### 6.2. Test 2

The initial setup for this test can be seen in Figure 21. In bin A there are the known target items 'flashlight' and 'black fashion gloves'. In bin B, there are the unknown 'big duck', and known 'white facecloth' and 'marbles'. In bin C, both the 'ruban isolant' and 'balls', which are unknown. In bin D, the unknown 'brush'. In bin E, the known 'mesh-cup' and 'duct-tape'.

In this case, three items were not successfully picked up: 'duct tape' was not recognized, and even though 'balls' and 'marbles' were correctly identified, picking up with the vacuum gripper and the Pincher gripper, respectively, failed.



Figure 21. Initial setup for Test 2.

### 6.3. Summary of Conducted Tests

The summary of the eight conducted tests can be seen in Table 4. The number of items that were successfully picked up is shown for each test. In the second and third columns, respectively, the percentage of the target known items that were correctly picked is shown, along with that percentage for the unknown items. The time for the task to finish is also shown (note that there is a limitation of 15 min for each test). Finally, the score according to the *Amazon Robotic Challenge* rules is shown in the rightmost column.

A total of 62 items were successfully picked and placed in the eight tests, out of the overall 80 target items to be identified among the 256 items in the bins ( $8 \times 32$ )—the result was a 77.5% success ratio. Some of the target items were the same in various tests, but under different conditions (pose, visibility). This ratio is higher than 69% for objects in isolation, but it has to be noted that now attempts to lift an item that failed were repeated up to three times as soon as the sensors detected the error; if the second or third try were successful it is counted as a success according to the competition rules. No significant differences are to be found in the percentages for known and unknown items; this is congruent with the fact that the grasp planning approach does not rely on previous knowledge about an object, beyond the decision of which gripper and algorithm to use. For the known items, this decision was taken by looking at the optimization Table 3 above. For unknown items this information was input manually in the available minutes, and based on their similarity with known items.

**Table 4.** Summary data of conducted tests.

Test #	Successful Items	Known Items (%)	Unknown Item (%)	Time (min:s)	Amazon Score
1	9/10	100	83	8:56	135
2	7/10	66	75	14:29	80
3	8/10	66	100	8:40	105
4	8/10	75	83	8:27	95
5	8/10	83	75	11:18	90
6	7/10	50	100	11:11	110
7	9/10	100	75	10:15	100
8	6/10	60	60	14:41	75
<b>Averages</b>	<b>7.75/10</b>	<b>75</b>	<b>81</b>	<b>11:00</b>	<b>98</b>

## 7. Conclusions

Although the problem of grasp planning has been around in the robotics community for a long time [1], the existing solutions tend to be limited when it comes to deploying them in realistic scenarios. Our main contribution here has been to try and apply previous knowledge together with our own methods in a realistic scenario as proposed by Amazon for its warehouses. The hardware employed is very cost effective: from the Baxter robot itself to the in-house made suction-cup end-effector and the 3D-printed Pincher gripper. This adds extra limitations on the precision and repeatability of the system.

The UJI RobinLab team took part in the ARC'17 in order to automate this warehouse environment. One of the necessary accomplishments for the robot is to be able to grasp an object, which requires a software approach that matches the robotic hardware. Based on our experience, we present here a practical solution to the problem of robot picking in this unstructured, realistic scenario by using an approach based on two algorithms that takes a point cloud as an input and outputs a position and orientation of the gripper given the hardware specifications and restrictions.

Two state-of-the-art algorithms for two-finger grippers were analyzed since they both use a point cloud as an input, and a grasp position and orientation as output. We have implemented and systematically studied them in terms of computation time, robustness and success rate. It was concluded that AGILE was less robust and took longer than HAF. The latter gave fewer grasp options with only vertical grasps, but showed better

robustness and computational speed. Given the results, it was decided that AGILE did not add significant contribution to the task and was discarded for the system.

Another contribution of the work is the creation and testing of the centroid normals approach (CNA) algorithm. It uses the point cloud and the major graspable component of the object in order to find the centroid and its normals in the flattest part of the point cloud of the object. Then, grasps are rotated around the vertical z-axis in such a way that the final grasp is most comfortable for the Baxter robot. This approach was the most successful in combination with a two-finger gripper and a vacuum gripper.

The final contribution was to match the competition objects to the grasping techniques and grippers to maximize the number of grasped items, as shown in Table 3. This table exposes the analysis that one universal algorithm for one universal gripper has not been found and currently the best solution is to use various algorithms for various grippers. This is supported by the fact that even though our overall success rates of 69% and 77.5% for our two sets of tests, respectively, may appear to not be high enough, our detailed failure analysis provides important clues about the reasons why the system fails, which very often are unrelated to the grasping algorithms themselves (object texture, imprecision of the control and/or point cloud, kinematic restrictions, etc.). We believe that our results will provide a useful addition to the literature in the field towards the deployment of practical working systems.

**Author Contributions:** Conceptualization, A.M.; Methodology, N.N.; Project administration, A.P.d.P.; Supervision, A.M. and A.P.d.P.; Validation, E.C.; Writing—original draft, N.N.; Writing—review and editing, A.M., E.C. and A.P.d.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper describes research conducted at the UJI Robotic Intelligence Laboratory. Support for this laboratory is provided in part by Ministerio de Economía y Competitividad (DPI2015-69041-R, DPI2017-89910-R), by Universitat Jaume I (P1-1B2014-52) and by Generalitat Valenciana (PROMETEO/2020/034). The first author was recipient of an Erasmus Mundus scholarship by the European Commission for the EMARO+ Master Program.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank Monica Arias for her work on the system tests.

**Conflicts of Interest:** The authors declare that they have no conflict of interest.

## References

1. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-Driven Grasp Synthesis—A Survey. *IEEE Trans. Robot.* **2013**, *30*, 289–309. [[CrossRef](#)]
2. Nechyporenko, N.; Morales, A.; del Pobil, A.P. Grasping Strategies for Picking Items in an Online Shopping Warehouse. In *Intelligent Autonomous Systems 15*; Strand, M., Dillmann, R., Menegatti, E., Ghidoni, S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 775–785.
3. Song, D.; Ek, C.H.; Huebner, K.; Kragic, D. Task-Based Robot Grasp Planning Using Probabilistic Inference. *IEEE Trans. Robot.* **2015**, *31*, 546–561. [[CrossRef](#)]
4. Madry, M.; Song, D.; Kragic, D. From object categories to grasp transfer using probabilistic reasoning. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 1716–1723.
5. Huebner, K.; Kragic, D. Selection of robot pre-grasps using box-based shape approximation. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Nice, France, 22–26 September 2008; pp. 1765–1770.
6. Dang, H.; Allen, P.K. Semantic grasping: Planning task-specific stable robotic grasps. *Auton. Robot.* **2014**, *37*, 301–316. [[CrossRef](#)]
7. Nikandrova, E.; Kyrki, V. Category-based task specific grasping. *Robot. Auton. Syst.* **2015**, *70*, 25–35. [[CrossRef](#)]
8. Miller, A.T.; Knoop, S.; Christensen, H.I.; Allen, P.K. Automatic grasp planning using shape primitives. In Proceedings of the Robotics and Automation, 2003. Proceedings. ICRA'03, IEEE International Conference, Taipei, Taiwan, 14–19 September 2003; Volume 2, pp. 1824–1829.
9. Saxena, A.; Wong, L.L.; Ng, A.Y. Learning Grasp Strategies with Partial Shape Information. In Proceedings of the AAAI-08, Chicago, IL, USA, 13–17 July 2008; Volume 3, pp. 1491–1494.
10. Fischinger, D.; Vincze, M.; Jiang, Y. Learning grasps for unknown objects in cluttered scenes. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 609–616.

11. Fischinger, D.; Weiss, A.; Vincze, M. Learning grasps with topographic features. *Int. J. Robot. Res.* **2015**, *34*, 1167–1194. [[CrossRef](#)]
12. Gualtieri, M.; ten Pas, A.; Saenko, K.; Platt, R. High precision grasp pose detection in dense clutter. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 598–605.
13. Jiang, Y.; Moseson, S.; Saxena, A. Efficient grasping from rgb-d images: Learning using a new rectangle representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3304–3311.
14. Yoneyama, R.; Duran, A.J.; del Pobil, A.P. Integrating Sensor Models in Deep Learning Boosts Performance: Application to Monocular Depth Estimation in Warehouse Automation. *Sensors* **2021**, *21*, 1437. [[CrossRef](#)] [[PubMed](#)]
15. Hasegawa, S.; Wada, K.; Niitani, Y.; Okada, K.; Inaba, M. A three-fingered hand with a suction gripping system for picking various objects in cluttered narrow space. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1164–1171. [[CrossRef](#)]
16. Tobin, J.; Biewald, L.; Duan, R.; Andrychowicz, M.; Handa, A.; Kumar, V.; McGrew, B.; Ray, A.; Schneider, J.; Welinder, P.; et al. Domain Randomization and Generative Models for Robotic Grasping. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3482–3489.
17. Mahler, J.; Matl, M.; Liu, X.; Li, A.; Gealy, D.; Goldberg, K. Dex-Net 3.0: Computing Robust Vacuum Suction Grasp Targets in Point Clouds Using a New Analytic Model and Deep Learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 5620–5627.
18. Zeng, A.; Song, S.; Yu, K.T.; Donlon, E.; Hogan, F.R.; Bauza, M.; Ma, D.; Taylor, O.; Liu, M.; Romo, E.; et al. Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3750–3757.
19. De Coninck, E.; Verbelen, T.; Van Molle, P.; Simoens, P.; Dhoedt, B. Learning robots to grasp by demonstration. *Robot. Auton. Syst.* **2020**, *127*, 103474. [[CrossRef](#)]
20. Schwarz, M.; Lenz, C.; García, G.M.; Koo, S.; Periyasamy, A.S.; Schreiber, M.; Behnke, S. Fast Object Learning and Dual-arm Coordination for Cluttered Stowing, Picking, and Packing. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3347–3354.
21. Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. *Proc. Mach. Learn. Res.* **2018**, *87*, 651–673.
22. Berscheid, L.; Meißner, P.; Kröger, T. Self-Supervised Learning for Precise Pick-and-Place Without Object Model. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4828–4835. [[CrossRef](#)]
23. Mallick, A.; del Pobil, A.P.; Cervera, E. Deep Learning Based Object Recognition for Robot Picking Task. In *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, IMCOM '18*; ACM: New York, NY, USA, 2018; pp. 30:1–30:9.
24. del Pobil, A.P.; Kassawat, M.; Duran, A.J.; Arias, M.A.; Nechyporenko, N.; Mallick, A.; Cervera, E.; Subedi, D.; Vasilev, I.; Cardin, D.; et al. UJI RobInLab's approach to the Amazon Robotics Challenge 2017. In Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Korea, 16–18 November 2017; pp. 318–323.
25. D'Avella, S.; Tripicchio, P.; Avizzano, C.A. A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper. *Robot. Comput. Integr. Manuf.* **2020**, *63*, 101888. [[CrossRef](#)]
26. Wurman, P.R.; Romano, J.M. The amazonpicking challenge 2015. *IEEE Robot. Autom. Mag.* **2015**, *22*, 10–12. [[CrossRef](#)]
27. Yu, K.T.; Fazeli, N.; Chavan-Dafle, N.; Taylor, O.; Donlon, E.; Lankenau, G.D.; Rodriguez, A. A Summary of Team MIT's Approach to the Amazon Picking Challenge 2015. *arXiv* **2016**, arXiv:1604.03639.
28. Hernandez, C.; Bharatheesha, M.; Ko, W.; Gaiser, H.; Tan, J.; van Deurzen, K.; de Vries, M.; Van Mil, B.; van Egmond, J.; Burger, R.; et al. Team delft's robot winner of the amazon picking challenge 2016. In *RoboCup 2016: Robot World Cup XX*; Springer: Cham, Switzerland, 2016; pp. 613–624.
29. Morrison, D.; Tow, A.W.; McTaggart, M.; Smith, R.; Kelly-Boxall, N.; Wade-McCue, S.; Erskine, J.; Grinover, R.; Gurman, A.; Hunn, T.; et al. Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge. *arXiv* **2017**, arXiv:1709.06283.
30. Correll, N.; Bekris, K.E.; Berenson, D.; Brock, O.; Causo, A.; Hauser, K.; Okada, K.; Rodriguez, A.; Romano, J.M.; Wurman, P.R. Analysis and Observations from the First Amazon Picking Challenge. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 172–188. [[CrossRef](#)]
31. Eppner, C.; Höfer, S.; Jonschkowski, R.; Martín-Martín, R.; Sieverling, A.; Wall, V.; Brock, O. Four Aspects of Building Robotic Systems: Lessons from the Amazon Picking Challenge 2015. *Auton. Robot.* **2018**, 1–17. [[CrossRef](#)]
32. Prats, M.; Martínez, E.; Sanz, P.; del Pobil, A. The UJI Librarian Robot. *J. Intell. Serv. Robot.* **2008**, *1*, 321–335. [[CrossRef](#)]
33. Fischinger, D.; Vincze, M. Empty the basket—a shape based learning approach for grasping piles of unknown objects. In Proceedings of the Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 2051–2057.
34. Nguyen, V.D. Constructing force-closure grasps. In Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 7–10 April 1986; Volume 3, pp. 1368–1373.
35. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.

36. ten Pas, A.; Platt, R. Using Geometry to Detect Grasps in 3d Point Cloud. In Proceedings of the International Symposium on Robotics Research (ISRR), Genova, Italy, 12–15 September 2015.
37. Choi, S.; Kim, T.; Yu, W. Performance Evaluation of RANSAC Family. In Proceedings of the British Machine Vision Conference 2009, London, UK, 7–10 September 2009.
38. Chum, O.; Matas, J. Matching with PROSAC-Progressive Sample Consensus. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Volume 1-Volume 01; CVPR '05*; IEEE Computer Society: Washington, DC, USA, 2005; pp. 220–226.