applied sciences

MDPI

Article

# Object-Independent Grasping in Heavy Clutter

**HyunJun Jo and Jae-Bok Song ***

School of Mechanical Engineering, Korea University, Seoul 136-713, Korea; jhj0630@korea.ac.kr
* Correspondence: jbsong@korea.ac.kr; Tel.: +82-2-923-3591

Received: 22 November 2019; Accepted: 21 January 2020; Published: 23 January 2020


**Abstract:** When grasping objects in a cluttered environment, a key challenge is to find appropriate poses to grasp effectively. Accordingly, several grasping algorithms based on artificial neural networks have been developed recently. However, these methods require large amounts of data for learning and high computational costs. Therefore, we propose a depth difference image-based bin-picking (DBP) algorithm that does not use a neural network. DBP predicts the grasp pose from the object and its surroundings, which are obtained through depth filtering and clustering. The object region is estimated by the density-based spatial clustering of applications with noise (DBSCAN) algorithm, and a depth difference image (DDI) that represents the depth difference between adjacent areas is defined. To validate the performance of the DBP scheme, bin-picking experiments were conducted on 45 different objects, along with bin-picking experiments in heavy clutters. DBP exhibited success rates of 78.6% and 83.3%, respectively. In addition, DBP required a computational time of approximately 1.4 s for each attempt.

**Keywords:** grasping; manipulation; robotics; machine learning

## 1. Introduction

For a robot to grasp a target object in a cluttered environment successfully, where many objects are stacked in a small space such as a box, the gripper should not collide with surrounding objects or the walls of the box. Because of the recent development of artificial neural networks (ANN), grasping algorithms can provide excellent performance if sufficient data are provided for learning [1–3]. However, learning often requires several robots and devices to compute the vast amount of data needed. Additionally, in cases where the target objects change frequently (such as in the logistics industry), ANN-based grasping algorithms have to be retrained, which is inefficient [4]. Therefore, an algorithm that allows the robot to grasp unknown objects without excessive learning is necessary.

Many grasping algorithms use either a geometry-based or data-driven method. The former is a traditional method in which the grasp pose is estimated by predicting the exact three-dimensional (3D) position of an object [5,6] or by matching the 3D point cloud using known 3D computer-aided design (CAD) models [7–9]. Therefore, applying this method to a new object is cumbersome, because an accurate CAD model is needed and cannot always be obtained. Thus, the estimated pose of the target object may be inaccurate. Because of this, methods that estimate the pose of the objects in 3D environments without CAD models have been recently proposed [10]. Though geometry-based grasping methods are often used because the CAD models of manufactured objects are available, the logistics industry is unlikely to have CAD models of the products. Thus, these methods are hardly applied in logistics.

In contrast, in the data-driven method, the grasp poses of the objects are estimated using an ANN-based learning scheme. This method generally has a higher success rate than the traditional geometry-based methods. In this method, RGB images [11,12], depth images [3,13], or both [14] can be used. However, a data-driven method requires a large amount of data that are manually

labeled; thus, collecting the training dataset is time-consuming and costly, and the training time is long. To solve these issues, methods such as obtaining the data from simulations [13] and using generative adversarial networks (GANs) [15] have been proposed. However, simulations may have low success rates because of the difference to reality, and the GAN-based method requires a long training time [16] and is hard to be trained. There are also reinforcement learning-based schemes. These, similar to deep learning-based methods, have good performance when properly taught. However, their learning requires an enormous amount of data. For instance, Google collected more than 0.9 million grasp data over several months using 14 robots [1,2]. Using a single robot, this would have taken years. Furthermore, even when these data are collected, the algorithm can hardly operate if they are not obtained in different environments.

Herein, we propose a bin-picking scheme based on the depth difference image (DDI), which estimates the graspability by analyzing the space around the object to be grasped. By DDI-based bin picking (DBP), a robot with a two-finger gripper can grasp unknown objects in a cluttered environment. This does not require a learning process (which requires a substantial amount of data) or CAD models of the target objects. Therefore, the most significant contribution of this study is to provide a generalized grasp solution that does not need prior information, including CAD models and training data.

DBP consists of a grasp candidate generator, grasp pose evaluator, and grasp pose modifier. The grasp candidate generator considers the shape of an object and the surrounding space, generating a group of candidates for the robot to attempt grasping. The grasp pose evaluator determines the most appropriate grasp candidate using a Gaussian mixture model (GMM) and DDI. The grasp pose modifier obtains the final grasp pose by adjusting that determined by the grasp pose evaluator. Experiments involving the bin picking of different objects in a two-dimensional (2D) clutter and of one type of object in a heavy clutter revealed that this method is effective.

The remainder of this paper is organized as follows. In Section 2, the overall structure of DBP and the individual modules are described in detail. Section 3 presents the experimental results and Section 4 analyzes the experimental results. Finally, Section 5 presents the conclusions.

## 2. DBP

The DBP structure used in this study is shown in Figure 1. DBP consists of three elements: a grasp candidate generator, a grasp pose evaluator, and a grasp pose decider. The grasp candidate generator processes the image obtained by a depth sensor and generates a group of grasp pose candidates. The grasp pose evaluator selects the most appropriate candidate among those obtained from the grasp candidate generator by analyzing the shape of the target object and the surrounding space. The grasp pose decider adjusts the grasp pose to obtain a more appropriate one. Using the foregoing procedure, robotic grasping can be performed without learning using devices such as a graphics processing unit (GPU).

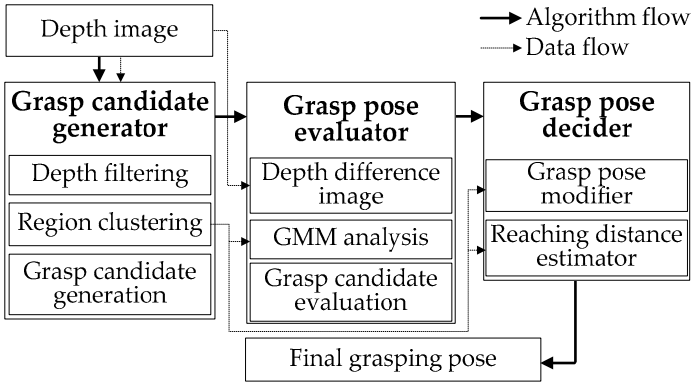


**Figure 1.** Structure of DBP (DDI-based bin picking).

### 2.1. Grasp Candidate Generator

The grasp candidate generator provides grasp poses that are likely to lead to a successful object grasping. In this, three processes are involved: depth filtering, region clustering, and grasp candidate generation.

First, depth filtering removes the image, maintaining only the data of the lowest $p$%, using the height. This is because objects located higher are generally easier to grasp. The lowest $p$% is selected because high objects appear closer to the camera of the robot.

Then, region clustering divides the filtered depth image into several object regions. Here, the depth image is clustered using a density-based spatial clustering of applications with noise (DBSCAN) algorithm on the filtered $p$% data. Figure 2 shows an example of DBSCAN. If a circle of radius $\varepsilon$ is drawn around points A and F, a minimum of five points falls in the circle. Because points A and F are in the same circle, they belong to the same cluster and are called the core points. When points B, C, D, E, and G are centered, instead, less than five points are in the circle; thus, these are called border points. Point H is never included; thus, it is called a noise point. Although DBSCAN does not need to set the number of clusters in advance, it can detect clusters with geometric shapes and outliers. In this study, $\varepsilon$ was set to 10 pixels, and the minimum number of samples for one cluster was set to 5. However, these parameters can be changed depending on the environment, e.g., the number and shape of the target objects and the resolution of the depth sensor.

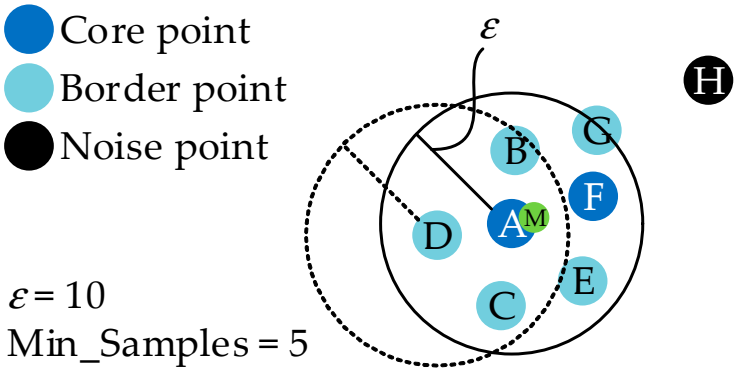


**Figure 2.** Example of DBSCAN (density-based spatial clustering of applications with noise).

Finally, grasp candidates are generated for each cluster estimated by DBSCAN. Figure 3 shows an example of the whole operation. In the rightmost figure, each cluster has 10 grasp candidates. Grasp candidates consist of the locations, grasp angles, and width of the gripper. The locations are determined from the centroids of the clusters. For example, in Figure 2, the centroid of the cluster is point M. The grasp angles are simply multiples of (180/$n$). The gripper width is equal to the smallest dimension of the width and height of a rectangle surrounding the cluster. In Figure 3, for example, the gripper width is $h_0$.

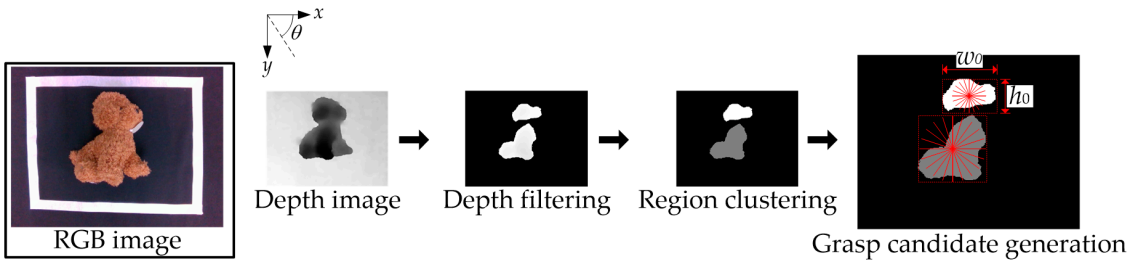


**Figure 3.** Operation of the grasp candidate generator.

### 2.2. Grasp Pose Evaluator

The grasp pose evaluator identifies the most appropriate grasp candidate among those provided by the grasp candidate generator considering the object shape and surrounding space. It performs a DDI analysis, GMM analysis, and graspability evaluation through a cost function with three parameters.

#### 2.2.1. DDI

The DDI is computed using the maximum depth difference between adjacent pixels in the depth image. This novel method produces large values at the boundary between the objects and the surrounding environment, and small values in areas exclusively belonging to either of them.

The DDI can be obtained as follows. First, an $m \times m$ region is filtered from the upper-left corner of the depth image. Here, the largest difference between the central pixel and the adjacent pixels is used as the new output. Then, the filter moves one pixel to the right according to the sliding-window approach and repeats the operation. At the end of the row, the filter moves to the next column. The corresponding pseudo-code is presented in Algorithm 1 for the case of $m = 3$, and a sample DDI is shown in Figure 4.

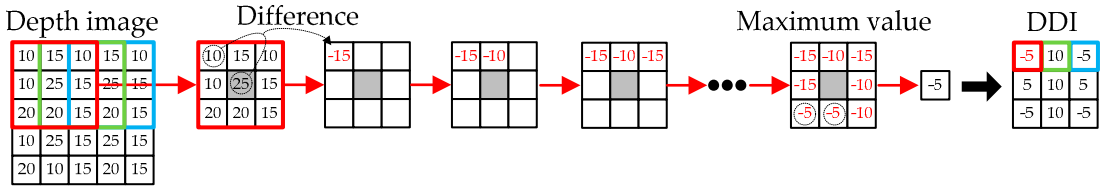


**Figure 4.** Example of DDI (Depth difference image).

The size $m$ can be any odd number except 1. The difference in the DDI for different $m$ is not large, mainly being that for larger $m$, the size of the DDI is smaller. In this study, $m$ was set to 3 to approximate the size of the resulting image to that of the input image. However, $m$ can be safely set to another odd number.

**Algorithm 1** DDI

**Data**: Depth image $\boldsymbol{d}_{\text{in}}$, $d_{i,j} \in \boldsymbol{d}_{\text{in}}$
**Result**: Depth difference image $\boldsymbol{D}$, $D_{i,j} \in \boldsymbol{D}$
**for** $i = 1, 2, 3, \ldots, (width\_of\_\boldsymbol{D}_{\text{in}} - 2)$ **do**
  **for** $j = 1, 2, 3, \ldots, (height\_of\_\boldsymbol{D}_{\text{in}} - 2)$ **do**

$$D_{i,j} \leftarrow \max(d_{i,j} - d_{i+1,j+1}, d_{i+1,j} - d_{i+1,j+1}, d_{i+2,j} - d_{i+1,j+1}, d_{i,j+1} - d_{i+1,j+1}, d_{i+2,j+1} - d_{i+1,j+1}, d_{i,j+2} - d_{i+1,j+1}, d_{i+1,j+2} - d_{i+1,j+1}, d_{i+2,j+2} - d_{i+1,j+1})$$

As shown in Figure 5, the DDI has large values at the contour of the object, similar to contour extraction, e.g., using a Sobel operator. However, there is a significant difference. Because a Sobel operator outputs only small values (e.g., 0–10), the depth difference cannot be properly represented. In contrast, the DDI can display both the contour of the object and the depth difference between neighboring pixels. This feature is used to estimate the graspability of the grasp candidates.

#### 2.2.2. Evaluation Model by GMM

To evaluate the grasp candidates, a model based on the GMM was designed, using three Gaussian models and DDI values corresponding to the grasp candidates, as shown in Figure 6. Additionally, a cost function was developed using the three parameters defined in the following. In Figure 6, the three Gaussian models are ordered according to their $x$ values and denoted 1, 2, and 3. $G_2$ can be interpreted as the area where the object exists, and $G_1$ and $G_3$ can be considered as spaces to the left and right of the object, respectively. According to the Gaussian models obtained from the GMM,

the proportion difference, height difference, and width are defined, and the cost function is constructed by multiplying or dividing them.

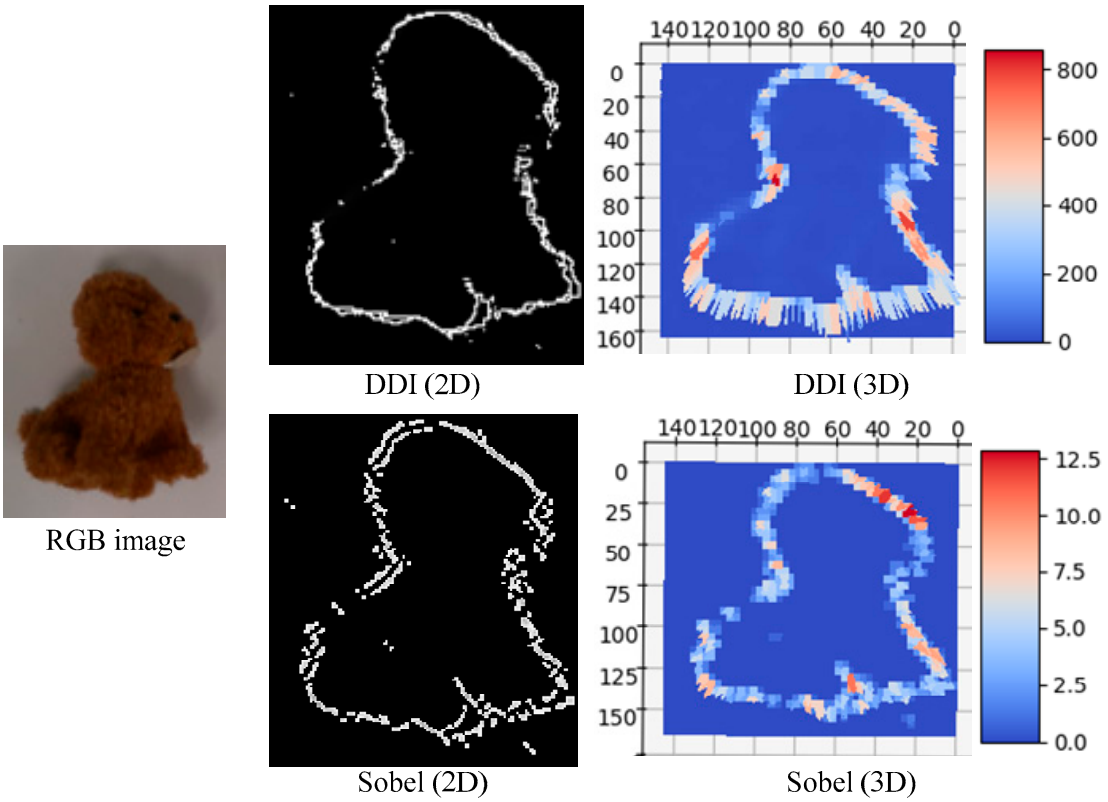


**Figure 5.** DDI and result of the Sobel operator in two and three dimensions.

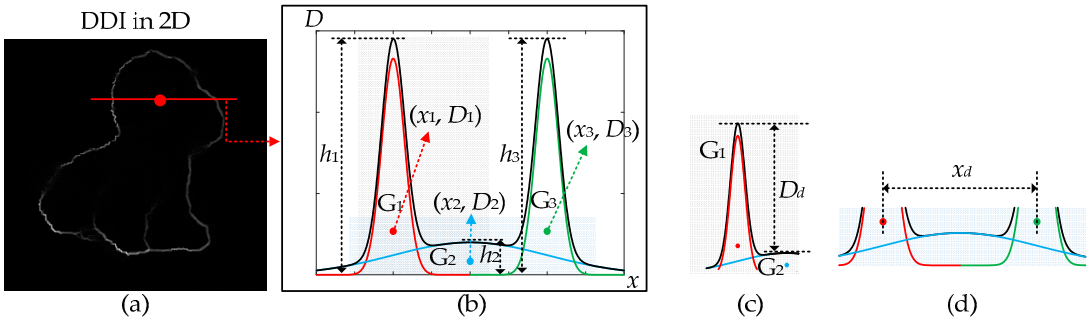


**Figure 6.** Evaluation model composed of three Gaussian models: examples of (**a**) the DDI, (**b**) the DDI profile along the grasp candidate, (**c**) the height difference $D_d$, and (**d**) the width $x_d$.

Figure 6a,b shows the DDI and its profile along the grasp candidate indicated by the red line in Figure 6a, respectively. In (b), point $(x_i, D_i)$ and $h_i$ are the average point and the maximum value of each Gaussian model estimated by the GMM, respectively.

The proportion difference index $d_p$ is defined as the ratio of $G_1$ to $G_3$. The proportion $p$ is defined: as

$$p_i = \frac{\text{number of data belonging to Gaussian model } i}{\text{number of total data}}; \quad (1)$$

thus, $p_1 + p_2 + p_3 = 1$. For example, in Figure 6, $p_1 = 0.25$, $p_2 = 0.5$, and $p_3 = 0.25$. If $p_1$ and $p_3$ have different values, there is space only on one side (left or right) of the object. Therefore, to select the cases in which both the left and right sides of the object are wide, the proportion difference $d_p$ is defined as:

$$d_p = \frac{|p_1 - p_3|}{\max(p_1, p_2, p_3)}, \tag{2}$$

where the difference between $p_1$ and $p_3$ is divided by the largest value among $p_1$, $p_2$, and $p_3$ for normalization. Thus, $d_p$ indicates whether $p_1$ and $p_3$ are similar. However, even if they are, the spaces may not be large enough. Then, the height difference, which is the second evaluation index, is used.

The height difference index $d_h$ is defined as:

$$d_h = \frac{D_d}{\max(h_1, h_2, h_3)} = \frac{|h_2 - \min(h_1, h_3)|}{\max(h_1, h_2, h_3)}, \tag{3}$$

where the height difference $D_d$ of Figure 6c is divided by the largest value among $h_1$, $h_2$, and $h_3$ for normalization. Thus, $d_h$ indicates the depth of the space around the object. A larger $d_h$ indicates a deeper space around the object. In summary, $d_p$ and $d_h$ indicate the width and depth of the space around the object, respectively.

The width index $w_c$ is defined by the difference between the $x$ values of $G_1$ and $G_3$ of Figure 6d, i.e.,

$$w_c = \begin{cases} \infty \text{ for } x_d = 0 \\ -\log(x/c) + 1 \text{ for } 0 < x_d < c \\ 1 \text{ for } x_d = c \\ 2^{x/c} - 1 \text{ for } c < x_d < x_l \\ \infty \text{ for } x_d > x_l \end{cases} \quad \text{where } x_d = x_3 - x_1. \tag{4}$$

Here, $x_d$ represents the distance between $G_1$ and $G_3$, and $x_l$ represents the maximum width of the gripper. $c$ is proportional to the width of the gripper and the camera resolution and inversely proportional to the distance. Because the opening of the gripper is limited, $w_c$ is infinite when $x_d > x_l$. Additionally, it is assumed that the gripper has an optimal width to grasp an object; thus, $w_c$ is minimized at a specific $c$ value and increases rapidly as $c$ deviates from this value. The expressions for $w_c$ for $0 < x_d < c$ and $c < x_d < x_l$ were initially designed as linear functions, but they were replaced with exponential functions to optimize the opening width of the gripper.

#### 2.2.3. Evaluation function

The evaluation function $e$ is defined according to the three foregoing evaluation indices as:

$$e = \frac{d_p \times w_c}{d_h}, \tag{5}$$

where $d_p$ and $d_h$ determine whether there is enough space around the object, and $w_c$ determines whether the width of the object is appropriate for grasping according to the width of the gripper. Therefore, the evaluation function determines the graspability according to the surrounding space and the shape of the object. Because, when the object can be appropriately grasped, $d_p$ and $w_c$ are small, and $d_h$ is large, the optimal grasp pose corresponds to the smallest $e$.

### *2.3. Grasp Candidate Decider*

The grasp pose decider consists of the grasp pose modifier and the reaching distance estimator. The grasp pose modifier determines the final grasp pose of the robot, and the reaching distance estimator determines the distance that the robot must travel downward for grasping.

The grasp pose modifier updates the location and width of the gripper. First, the width of the gripper is estimated according to the optimal grasp pose and shape of the cluster in the depth image. In Figure 7a, the width of the gripper is larger than the target object. To avoid collisions with other objects in the clutter, the grasping width should be reduced, and the location should be modified according to the new grasping width. In the clustered depth image, the width has been reduced to fit the boundaries of the cluster. A new grasping width is obtained by adding a margin to this value. Additionally, as shown in Figure 7b, the half-width of the newly estimated gripper is set as the new center position of the gripper.

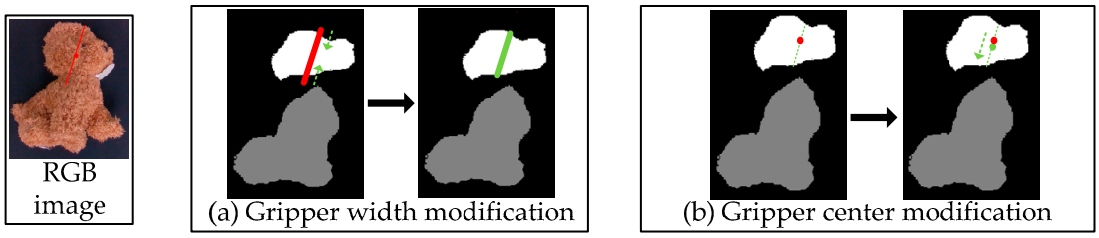


**Figure 7.** Grasp pose modification of (**a**) the gripper width and (**b**) the gripper center.

Next, the reaching distance estimator determines the height at which the robot should approach the object. Because an RGB-D camera can only see one side of the object, the distance to approach before grasping must be determined according to the partial depth data of the object. The height is determined from the maximum and minimum depth ($h_{max}$ and $h_{min}$, respectively) of the cluster in the clustered depth image as:

$$h = h_{\max} + k \times (h_{\max} - h_{\min}), \tag{6}$$

where $k$ is a factor that indicates the correspondence of the $h_{max} - h_{min}$ difference to the reaching distance. In fact, though $h_{max}$ is the deepest of the filtered points, it is not large enough to grasp the object, because only the depth information close to the camera remains after filtering. Therefore, for reliable grasping, the robot must reach beyond the value of $h_{max}$, which is obtained by introducing $k$ (set to 0.5–1 in this study).

## 3. Experiments

Several experiments were conducted to determine whether the proposed DBP is effective for grasping objects that have a complex piling structure. For this purpose, the performance of DBP was compared with that of three grasping algorithms. The first one used a random method. In this, the grasp position was the central coordinate of the cluster found by the grasp candidate generator, and the grasp angle was determined randomly. Thus, compared to the DBP method, the grasp position was the same, and the grasp angle was different. The second algorithm was based on principal component analysis (PCA) [17]. In this method, the grasp center position was set to one of the center points of the clusters in the clustered depth image, and the grasp angle was obtained by PCA. Thus, a narrow part of the target object was used for grasping. The third algorithm was based on an ANN [18]. In this algorithm, the ANN received the depth image and used it to estimate the grasp pose. Note that the algorithm did not previously learn the objects to be used in the experiments.

The three algorithms and DBP were tested with different objects both in a 2D cluttered environment and in 3D bin picking. In the 2D clutter, grasping was performed for 20 types of objects in an area delimited by white lines. In 3D bin picking, the target objects were placed in a 390 mm × 480 mm × 250 mm box. In this case, the parameters of the DBP algorithm were $p = 0.1$, $n = 20$, $c = 80$, and $k = 0.5$.

Figure 8 shows the experimental setup, which comprised a UR5 robot, a RealSense D435 RGB-D sensor mounted on the robot arm, and a Robotiq two-finger gripper. The main central processing unit (CPU) was an Intel Core i9-7940X, and the GPU was a GeForce GTX 1080 Ti. Figure 8a shows the 45 different objects used in the experiments. In the 2D cluttered environment shown in Figure 8b,

20 objects were randomly selected among the 45 objects and stacked in the outlined area. In bin picking, as shown in Figure 8c, the 45 objects were used. The objects comprised the Australian Center for Robotic Vision (ACRV, Brisbane, Australia) picking benchmark (APB) [19], the Yale-CMU-Berkeley (YCB) benchmark [20], the World Robot Summit (WRS) 2018 set, and household items.

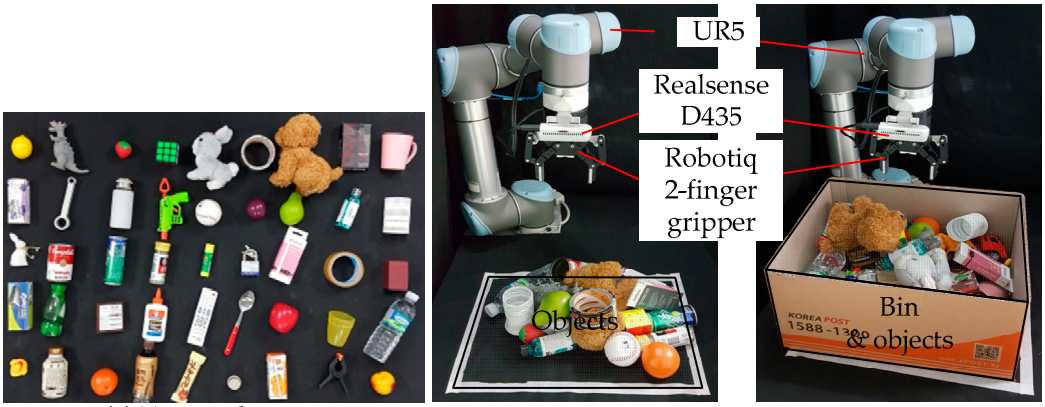


**Figure 8.** (**a**) Target objects; (**b**) grasping in 2D clutter; (**c**) grasping in 3D bin.

To test the DBP scheme in a heavy clutter, a 330 mm × 450 mm × 260 mm box was filled with small cosmetic containers, as shown in Figure 9a. Without proper consideration of the space around the object, grasping is difficult. In this experiment, the parameters of the DBP algorithm were $p$ = 0.05, $n$ = 20, and $c$ = 80. In contrast to the previous experiments, the width of the gripper was fixed, because only one type of object was targeted.

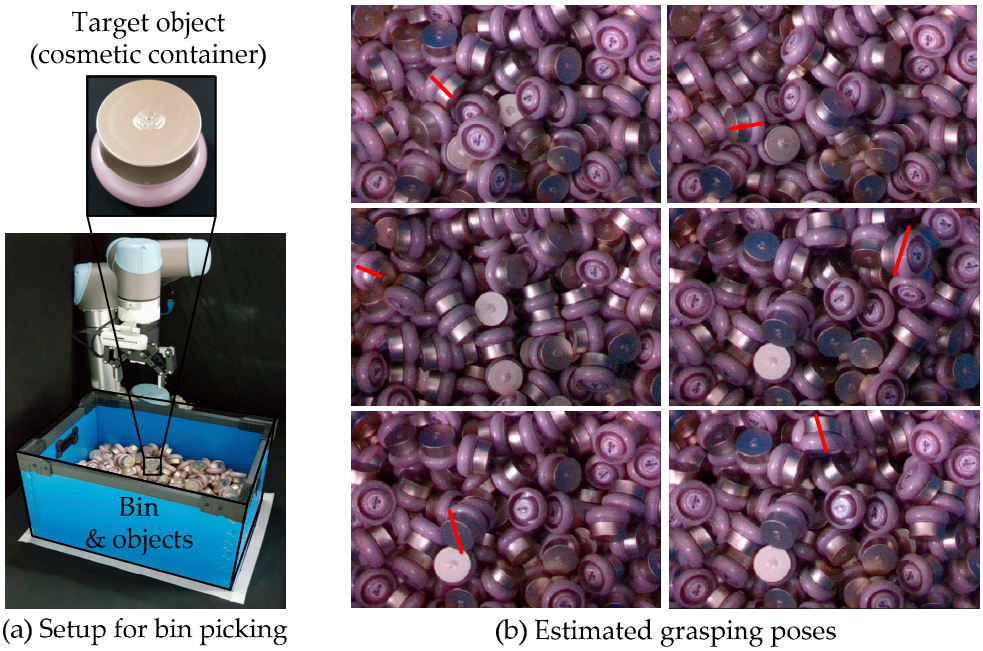


**Figure 9.** (**a**) Bin picking filled by cosmetic containers and (**b**) estimated grasp poses in heavy clutter.

## 4. Discussion

### *4.1. Comparisons with Other Algorithms*

As seen in Table 1, DBP exhibited the highest success rate in all the experiments. The learning-based methods have a lower success rate, though they show similar performance. In particular, the PCA-based

grasping performed similarly to DBP in the 2D and 3D clutters with different objects, because the gripper collided rarely with other objects during the grasp attempts owing to the large space between the objects in those experiments. In fact, the success rate in the heavy clutter of cosmetic containers, where DBP outperformed the other algorithms, supports this assumption. Thus, the experiments indicated that grasping without collisions is important in heavy-clutter environments. Examples of grasp poses estimated by DBP are shown in Figure 9b.

**Table 1.** Grasping success rates in the three environments.

| Experiment | Method | Success Rate (%) | Time (s) |
|---|---|---|---|
| 2D clutter | Random | 56.9 | 0.4 |
| | PCA | 81.1 | 0.7 |
| | ANN | 71.1 | 0.6 |
| | DBP | 84.5 | 1.4 |
| Bin picking with 45 objects | Random | 41.2 | 0.4 |
| | PCA | 78.4 | 0.8 |
| | ANN | 76.0 | 0.6 |
| | DBP | **78.6** | 1.4 |
| Bin picking with cosmetic containers | Random | 58.8 | 0.4 |
| | PCA | 42.1 | 0.7 |
| | ANN | 73.3 | 0.6 |
| | DBP | **83.3** | 1.4 |

In all the experiments, the PCA-based grasping was twice as fast as DBP, and the ANN-based grasping was in turn 1.16 times faster than PCA. However, DBP needed 1.4 s to estimate the grasp pose, which is sufficiently fast for practical applications. After the robot grasps an object, it needs time to move the object to the designated position. Additionally, if an eye-to-hand camera is used instead of the eye-in-hand camera used in our experiments, the robot can estimate a new grasp pose while moving an object.

In summary, the PCA grasping method had a good performance if the space around the object to be grasped was sufficiently large, and the performance deteriorated otherwise. The performance of the learning-based grasping was similar in all environments, but worse than DBP. In fact, DBP, which considers both the space around and the shape of the target object, obtained a good grasping success rate even when the space around the target object was small. Thus, DBP can be applied more widely than other grasping methods, because its performance is good in different environments.

### *4.2. Causes of Failures*

Though the DBP algorithm demonstrated the highest success rate, it showed 16.7% failures in grasping cosmetic containers. These failures are most likely caused by the environmental changes that occur after estimating the grasp pose. When a gripper approaches the object, its fingers often contact the surrounding objects. Such contact is likely to change the surrounding environment and the pose of the target object. Therefore, unless the grasp pose is corrected accordingly, the chance of successful grasping is reduced.

Another cause is related to the top-down grasping path that is used by most two-finger grippers. In this, the gripper first moves over the target object, then descends vertically to grasp the object. In this way, grasping objects near the wall is very difficult. Because, in grasp pose estimation, avoiding a collision with the wall has priority over grasping the object, the estimation of the correct grasp pose is not easy, especially for small bins.

## 5. Conclusions

A novel difference image-based bin-picking method was proposed for generalized grasping in heavy clutter. We introduced a DDI to analyze the geometry around the object to be grasped in the absence of a CAD model, a graspability evaluation method based on the DDI, and a DBP structure consisting of a grasp candidate generator, grasp pose evaluator, and grasp pose decider. The DBP method aims to estimate the optimal grasp pose in a short time with a cost-efficient process, grasping novel objects even in space-constrained environments. The performance of DBP was verified by grasping experiments in 2D clutter and 3D bin-picking environments. In the experiments, DBP exhibited better performance than other grasping methods. In particular, the success rate of DBP in heavy clutter with small objects was 83.3%, approximately 1.4 times higher than that of the other algorithms. Moreover, the computation time was 1.4 s, which is sufficiently fast for industrial and logistics applications.

**Author Contributions:** Conceptualization, H.J.; methodology, H.J.; software, H.J.; validation, H.J.; formal analysis, H.J.; investigation, H.J.; resources, H.J.; data curation, H.J.; writing—original draft preparation, H.J.; writing—review and editing, J.-B.S.; visualization, H.J.; supervision, J.-B.S.; project administration, J.-B.S.; funding acquisition, J.-B.S. All authors have read and agree to the published version of the manuscript.

**Funding:** This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Ministry of Science and ICT (MSIT, Sejong city, Republic of Korea). (No. 2018-0-00622, Robot manipulation intelligence to learn methods and procedures for handling various objects with tactile robot hands).

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

## References

1. Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* **2017**, *37*, 421–436. [CrossRef]
2. Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhouke, V.; et al. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In Proceedings of the 2018 Conference on Robot Learning (CoRL), Zurich, Switzerland, 29–31 October 2018; pp. 651–673.
3. Mahler, J.; Liang, J.; Niyaz, S.; Laskey, M.; Doan, R.; Liu, X.; Ojea, J.A.; Goldberg, K. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv* **2017**, arXiv:1703.09312.
4. Pharswan, S.; Vohra, M.; Kumar, A.; Behera, L. Domain-Independent Unsupervised Detection of Grasp Regions to Grasp Novel Objects. In Proceedings of the 2019 IEEE International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 640–645.
5. Papazov, C.; Haddadin, S.; Parusel, S.; Krieger, K.; Burschka, D. Rigid 3D geometry matching for grasping of known objects in cluttered scenes. *Int. J. Robot. Res.* **2012**, *31*, 538–553. [CrossRef]
6. Hernandez, C.; Bharatheesha, M.; Ko, W.; Gaiser, H.; Tan, J.; Deurzen, K.V.; Vries, M.D.; Bil, B.V.; Egmond, J.V.; Burger, R.; et al. Team delft's robot winner of the amazon picking challenge 2016. *arXiv* **2016**, arXiv:1610.05514.
7. Ciocarlie, M.; Hsiao, K.; Jones, E.G.; Chitta, S.; Rusu, R.B.; Sucan, I.A. Towards reliable grasping and manipulation in household environments. In *International Symposium on Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–12.
8. Hinterstoisser, S.; Holzer, S.; Cagniart, C.; Ilic, S.; Konolige, K.; Navab, N.; Lepetit, V. Multimodal Templates for Real-Time Detection of Texture-Less Objects in Heavily Cluttered Scenes. In Proceedings of the 2011 International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 858–865.
9. Kehoe, B.; Matsukawa, A.; Candido, S.; Kuffner, J.; Goldberg, K. Cloud-Based Robot Grasping with the Google Object Recognition Engine. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 4263–4270.
10. Dogar, M.; Hsiao, K.; Ciocarlie, M.; Srinivasa, S. Physics-based grasp planning through clutter. In *Robotics: Science and Systems VIII (RSS)*; MIT Press: Cambridge, MA, USA, 2012.
11. Benvegnu, L. *3D Object Recognition without Cad Models for Industrial Robot Manipulation*; Univerisità Degli Studi di Padova: Padua, Italy, 2017.

12. Pinto, L.; Gupta, A. Supersizing Self-Supervision, Learning to Grasp from 50k Tries and 700 Robot Hours. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3406–3413.
13. Lenz, I.; Lee, H.; Saxena, A. Deep Learning for Detecting Robotic Grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [CrossRef]
14. Johns, E.; Leutenegger, S.; Davison, A.J. Deep Learning a Grasp Function for Grasping under Gripper Pose Uncertainty. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4461–4468.
15. Bousmails, K.; Irpan, A.; Wohlhart, P.; Bai, Y.; Kelcey, M.; Kalakrishnan, M.; Downs, L.; Ibarz, J.; Pastor, P.; Konolige, K.; et al. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 4243–4250.
16. Asif, U.; Bennamoun, M.; Sohel, F.A. RGB-D Object Recognition and Grasp Detection Using Hierarchical Cascaded Forests. *IEEE Trans. Robot.* **2017**, *33*, 547–564. [CrossRef]
17. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* **1996**, *96*, 226–231.
18. Da-Wit, K.; HyunJun, J.; Jae-Bok, S. Grasping Method in a Complex Environment using Convolutional Neural Network Based on Modified Average Filter. In Proceedings of the 2019 International Conference on Ubiquitous Robots (UR), Jeju, Korea, 24–27 June 2019; pp. 113–117.
19. Leitner, J.; Tow, A.; Sunderhauf, N.; Dean, J.; Durham, J.; Cooper, M.; Eich, M.; Lehnert, C.; Mangels, R.; McCool, C.; et al. The ACRV Picking Benchmark: A Robotic Shelf Picking Benchmark to Foster Reproducible Research. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4705–4712.
20. Calli, B.; Walsman, A.; Singh, A.; Srinivasa, S.; Abbeel, P.; Dollar, A. Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols. *IEEE Robot. Autom. Mag.* **2015**, *22*, 36–52. [CrossRef]