Grasp Pose Detection Based On Point Cloud Shape Simplification

H W Liu¹ and C Q Cao²,3*
¹School of Mechanical Engineering, Nanjing University of Science & Technology, Nanjing 210094, China;
²Anhui Polytechnic University, Wuhu 241000, China;
³HIT Wuhu Robot Technology Research Institute Co., LTD, Wuhu 241007, China;
*caochuqing@hitrobot.com.cn

Abstract. In the robot grasping environment, grasping unknown objects that have neither model data nor RGB data is very important, we present a new approach for grasping unknown objects. Firstly, depth camera is used to obtain the partial 3D data of target object. Then, we perform 3D segmentation and the segmented parts are simplified to a cylinder, a sphere, an ellipsoid or a parallelepiped based on the geometric and semantic shape characteristic. Grasping constraints are used to obtain grasp candidates based on the simplified shape. The grasp descriptions are used to describe each candidates and CNN is used for grasping training. We evaluate our grasp networks both in simulation and robotic experiment, and the experiment result shows that the grasp quality score using simplified constraints is more robust than without simplified constraint when the gripper posture is uncertain.

1. Introduction

Autonomous grasping of robotic is one of the core technologies of intelligent robotic. However, the uncertainty of shape and attitude of target object limits the application of robotic grasping technology. To address this problem, recent research has focused on building a large-scale grasping dataset [1][2] and using neural networks for robotic grasping training, using images data [3] or point cloud data [4] acquired by sensors as input of grasping network. Most current grasping methods use two-dimension (image) data or 2.5D (depth image) data [5] as the input of grasping training. And very few people take the three-dimensional geometry information [6] of objects into consideration. Intuitively, when the shape and posture of the grasped object are uncertain, a single grasping direction may limit the grasping quality of robotic [7]. The grasping quality is usually related to the interaction between the robot gripper and the surface of target object, the lack of object geometric analysis may have a negative effect on grasping.

In order to address the above problems, we use the 3D data of the target object for training. However, computing a stable grasp is a “big-spending” when a large number of novel objects are added to the grasp database. Therefore, reusing the grasps of similar shapes is promising [8][9], as opposed to computing stable grasps for the novel objects from scratch. Those works take the appropriate parts of a given object for grasping, depending on the similar topologies and geometric shapes.

In this paper, we propose a method to detect grasp pose based on the three-dimensional point cloud simplification. As illustrated in Figure 1, first, the objects are segmented based on geometric and semantic shape characteristics, and then the main part is simplified as a kind of sphere, ellipsoid, cylinder and parallelepiped. The grasping pose is constrained based on the simplified shape feature.
Under the grasp constraint, we built a grasp dataset and trained it in YCB dataset [10] – an online dataset that includes depth scanning data of real objects. In our previous work, we have realized the segmentation and simplification of three-dimensional point cloud. Here, we summarize the contributions of this paper in the field of robotic grasping.

- We recommend applying the simplified algorithm based on the basic form to describe objects. In order to avoid the singularity of object description and make a more vivid description of object shape characteristics, the sphere, the ellipsoid, the cylinder and the parallelepiped are used to describe objects.
- We propose a grasp strategy to constrain the grasping position based on the simplified shape characteristics and suggest to grasp the main parts of the object. And acquiring grasp candidates based on the grasping constraints.
- We build a new grasp description that incorporates grasp heightmaps and angle graph of object surface to measure the grasp similarity.

2. Related work

The autonomous grasping depends on the recognition of object and the description of grasped object shape. The description of the object shape provides an effective way to select grasping strategy. In this section, we give a brief introduce of previous research work on robotic grasping, especially similar objects grasping and part-based grasping.

2.1. Similar objects grasping

In the grasping experiment on similar objects, the grasped objects can be classified into the same categories based on different characteristics, such as usage, application, or geometric shape. Goldfeder et al. [11] proposed a method of using hyperquadric curve tree to automatically approach the object geometry; Lei Q et al. [12] synthesized an executable grasp used cylinder searching on a single point cloud for unknown object. In [13], [14], the authors used the shape similarity of objects to transfer the optimal grasp position to the novel object. Tian H et al. [15] segmented the target objects according to the shape, marked the optimal grasp posture of each sample objects and mapped it to the parts with the same semantics in the target object to obtain the optimal grasp posture.

2.2. Part-based grasping

Many scholars have been proposed to segment and grasp the target parts based on some grasping characteristics. Kai et al. [16] proposed a segmentation algorithm based on the minimum volume bounding box, which decomposition the 3D points into simple objects that composed of multiple bounding boxes; Aleotti et al. [17] proposed a programming-based demonstrated method and a 3D segmentation method, they segmented the grasped object based on Reeb graphs which were used to generate the topological representation. At present, many researches use part information of objects for grasping training. In [18], [19], the authors design a GPD method to normalizes the point cloud and extract the projection features of 3-channels, 12-channels and 15-channels in the grasp closing region,
and construct a grasping classification model based on CNN. Liang H et al. [20] proposed PointNetGPD grasping method based on PointNet [20], taking the original point cloud as input and directly processing the three-dimension data that locates in the jaw for grasping detection and evaluation.

3. Grasp pose detection

3.1. Definitions
Given a set of point cloud and a description of the grasp configuration, the grasp pose detection problem is to identify grasp configurations from which a stable grasping quality would be formed if the robot’s gripping were to close. Given a specified object \( \mathcal{O} \), let \( \mathcal{Q} \subset \mathbb{R}^3 \) denote the robot workspace. The input of the algorithm is a set of 3D point cloud, \( \mathcal{C} \subset \mathcal{Q} \). Each point in the 3D point data is real-time 3D data acquired by the camera and matching at least one camera viewpoint acquisition data, \( \mathcal{V} : \mathcal{C} \rightarrow \mathcal{V} \), where \( \mathcal{V} \subset \mathbb{R}^3 \) indicates the set of camera viewpoints, \( \mathcal{CV} = (\mathcal{C}, \mathcal{V}) \) indicates the point cloud after multi-viewpoint data combination and viewpoints respectively. Let \( g = (p, r) \subset \mathbb{R}^6 \) denote a set of grasp configuration in 3D robot workspace, where \( p = (x, y, z) \subset \mathbb{R}^3 \) denote the position of the gripper, \( r = (r_x, r_y, r_z) \subset \mathbb{R}^3 \) denote the orientation of the gripper.

The expected grasping result is that if the robotic gripper is moved to a given gripper pose \( g \), when the gripper figure are closed, the force closure of the grasped object can be achieved. In this paper, we only consider two-fingered parallel jaw gripper and assume a depth camera to obtain real-time depth data.

3.2. Outline of the algorithm
In order to achieve the robust grasp and obtain the appropriate grasp pose, firstly, we pretreated the 3D data to achieve segmentation and simplification. Grasp constraints are used to obtain grasp candidates based on the simplified shape. The grasp descriptions in this paper are used to describe each candidates and CNN is used for grasping training. Finally, executing the grasp posture with highest score. The specific steps are shown in Figure 1.

In our previous work [21], we have completed the steps of pretreatment and successfully simplified the segmented parts into one of sphere, ellipsoid, cylinder and parallelepiped. As shown in Figure 2, our algorithm can be implemented in different objects, such as regular objects (plane, bottle et al) and irregular objects (cat, chicken model et al), the shown model in Figure 2 is acquired in SHREC’15 datasets [22].

It should be noted that in this grasping algorithm, we only simplify the main part of segmentation parts in order to reduce the running time, and focus on the next steps.

3.3. Sampling grasp candidates
The goal of this section is to obtain a larger set of grasp candidates (6-DOF hand pose), which may
contain one or more optimal grasps. When selecting grasp candidates, we hope they can be evenly distributed among the effective parts, that is, the graspable part of object surface. In this experiment, we defined each grasp candidate, $h \in \mathbb{Z}$, as a simple “hand”, where $\mathbb{Z}$ is the region of interest.

In the pretreatment step, we have simplified the main part of target object. We set the main part as the region of interest and constrained the grasping pose based on the simplified shape features to obtain the grasp candidates. The following experiment confirmed this method is effective for robot grasping. The grasp constraint is shown below:

- Simplified into sphere: the center position $p_{cen}$ of the gripper is located in the center of the ball, the orientation $r_{cen}$ of the gripper is a random direction vector (shown in Figure 4(a)).
- Simplified into ellipsoid: the center position $p_{cen}$ of the gripper is located in the central long axis of the ellipsoid, the orientation $r_{cen}$ is the same as one of the two short central axis (shown in Figure 4(b)).
- Simplified into cylinder: the center position $p_{cen}$ of the gripper is at the central axis, the orientation $r_{cen}$ is parallel to the central axis (shown in Figure 4(c)).
- Simplified into parallelepiped: the center position $p_{cen}$ of the gripper is at the center long axis, the orientation $r_{cen}$ is the same as one of all edges except the shortest edge of the parallelepiped (shown in Figure 4(d)).

We define $D(h) \in \mathbb{Q}$ as the volume occupied by the robotic gripper when it is fully open. Defined closing region, $C(h) \in \mathbb{Q}$, as the volume area swept by the gripper when the fingers are closed. In the vertical direction of each gripper, we “push” the gripper until the following conditions are met, and Figure 3 shows the above conditions occurred when grasping object.

**Condition 1.** The grabs hand and the point cloud will not collide and contact during the process of “push” hand: $D(h) \cap C = \emptyset$, and the center of the $C(h)$ is $p_{cen}$.

**Condition 2.** The side of the gripper hand or the bottom of the hand collides with the point cloud: $D(h) \cap C \neq \emptyset$, then stops pushing forward.

**Figure 3.** Illustrate of the conditions present in the paper. (a) Condition 1 occurred when grasping. (b) Condition 2 occurred when grasping.

On the basis of grasping constraints, considering the noise of the input cloud and obtaining sufficient grasp candidates, all the gripper positions $p_{cen}$ have random errors of 0–1 cm and all the gripper orientations $r_{cen}$ have random errors of 0–15 degree. As illustrated in Figure 4, we obtain grasping candidates though the above grasp constraints.
3.4. Classifying grasp candidates

The grasp training network in this paper is defined as a binary classification task that the input is the feature representation of a set of grasp candidates and the output is a prediction of whether or not the grasp candidate is graspable. The network structure we use is the same as the LeNet-5 [23] CNN structure: three convolutional layers, two pooling layers and a fully connected layer with a softmax on the output. Except the size of network learning rate is 0.01, the size of output, pooling strides and kernel, etc. are all the same as the LeNet network framework provided in Caffe [24].

3.5. Grasp candidate representation

We use the geometric of the mesh surfaces that contained in the closing region \( C(h) \in \mathbb{Q} \) to represent a grasp candidate. The two fingers of the robotic grab can be close to or far away from each other, because they are modeled as rectangles in this experiment, and the shape of the closing region \( C(h) \in \mathbb{Q} \) is a cuboid. Figure 5(a) shows a grasp candidate generated with three-dimensional mesh data from YCB dataset [10]. The region shown in red is the mesh points within closing region \( C(h) \). We define \( C(h) = (c, v) \) be a closing region, \( C(h) \) parameter is composed of the gripper centroid in 3D space \( c \in \mathbb{R}^3 \) and the approach direction of the gripper, or direction along which the fingers close, \( v \in \mathbb{C}^2 \).

First, the mesh points in the closing region \( C(h) \) are transformed into the unified grasp coordinate, as introduced in Figure 5(b) where z axis is parallel to \( v \), Figure 5(b) shows the closing region \( C(h) \) and local gripper coordinate.

Because we evaluate each grasp candidate as a high grasping quality or not by judging whether it is a force closure grasp [25], we only consider the surface that is in contact with two fingers of the jaw. To measure grasp similarly, we embed each grasp candidate \( C(h) = (c, v) \) in a feature space based on 2D projection heightmaps \( h(x, y) \) and angle graph \( a(x, y) \) of the local surface on left and right fingers of the jaw. Take the right jaw finger as an example, voxel the closing region in the xy plane to \( 30 \times 30 \).

As shown in Figure 5(c), let \( r \subset \mathbb{R} \) be a minimum projection distance long z direction, and the heightmaps \( h(x, y) \) is obtained through equation 1. Let the \( v_i \subset \mathbb{C}^2 \) be the normal vector of the minimum projection distance point, and calculate the angle \( w_i \subset \mathbb{R} \), \( w_i \in [0, 90] \) between \( v_i \) and the z axis, and obtain the angle graph \( a(x, y) \) (equation 2).

\[
h(x, y) = \frac{\max(r) - r(x, y)}{\max(r)} \times 255
\]
The two images, $h$ and $a$, are 30×30 images (Figure 5(d,e)). In summary, for each of the projection in two directions, we have two channels of feature representation respectively, and a total of 4 channels.

3.6. Generating training labels

We trained network in the follow form: First, we use the method presented in Section 3.3 to constrain the grasping pose based on the simplified shape to obtain the grasp candidates. Then, we obtain the corresponding surface features of each grasp candidate by using the method described in Section 3.5, and attach a label to each grasp candidate to indicate whether or not the candidate is an effective grasp pose by evaluating a force closure grasp [24] would be obtained if the grippers were to close, that is, each of the surface normals at the contacts between gripper and grasped object is opposite to the finger closing direction $v$ and collinear with the connection line between the contacts (see Figure 6).

Unfortunately, it is very tricky to determine whether a grasp candidate is a force closure grasp, because the real-time mesh is noise-containing. For example, the mesh data in the YCB dataset is noise-containing because it is reconstructed from the actual sensor. We “softening” the force closure condition in order to address the above problems. In particular, we consider the point within 2mm of the contact point between two fingers when the fingers are closed after voxelized the point cloud. When the normal of more than 1/4 point that exists in each contact is less than 20 degree with the finger closing direction $v$, it can be judged as graspable. The above method aims to search for a force closure grasp in small regions that may establish contact.

4. Experiment

We carry out the simulation experiment and the robotic grasping experiment on the method proposed in this paper respectively. In the simulation experiment, we mainly evaluate the performance of pretreatment method and grasp classification network. In the robotic experiment, we conduct multiple robotic grasp experiments to determine whether our grasping model can be implemented in real world.

4.1. Simulation experiments

In the simulation experiment, the grasping dataset we use is based on the YCB dataset, which contains the mesh points of 77 different objects formed by scanning and processing in real-time, and matching the grasp candidate representation obtained by the YCB dataset with the corresponding grasp labels that indicate whether or not the candidate is graspable. We defined robot grip as a parallel jaw that can grasp objects up to 11cm. The compute system we use consists of 16 GB of system memory, Corei7-8700k CPU, and an NVIDIA GeForce RTX 2080 graphic card. We build a dataset with 100k candidates, in which the positive and negative samples are 80k/20k respectively, and it takes 40min to train. In order to verify the effectiveness of the proposed method, we analyze the pretreatment method and grasping training method respectively.

1) Comparison between pretreatment and unpretreatment: The method we proposed in this paper pretreatments the object and then builds a new grasp description. In order to evaluate the positive effects of the pretreatment method proposed in this paper, we performed two sets of candidate experiments respectively using YCB dataset. In addition to our previous 100k grasp candidates, we recreated a dataset of grasp candidates that randomly acquired approximate 2k grasp candidates for
each of the 77 YCB objects, with a balance between positives and negatives of the grasp description, we obtained a total of 150k candidates. We obtained approximately 94% accuracy over 77 objects in the YCB dataset when training 100k grasp candidates with pretreatment (the red line in Figure 7). We obtain only 89% accuracy (the blue line in Figure 7) when we train 150k grasp candidates without pretreatment. The accuracy of two grasping methods is demonstrated in Figure 7.

**Figure 7.** Grasp detection accuracy without pretreatment (blue); grasp detection accuracy with pretreatment (red)

From Figure 7 we can see that the training model with pretreatment obtains better result than without pretreatment. It allows us to use pretreatment to acquire varying degrees of prior knowledge, which has a positive effect on the robotic grasping judgement. In the simulation experiment with pretreatment, since we have obtained grasping candidates which have a higher probability of grasping based on the object shape, we obtained higher grasp accuracy with fewer grasp candidates. Depending on the results described above, pretreatment plays a positive role in robotic grasping judgement.

2) Compared with other grasp methods: In the comparative experiment, we mainly want to compare the advanced grasping methods with our proposed methods in terms of performances on grasping quality classification. We chose GPD [18] [19] and PointNetGPD [20] as baseline because they are all the works closest to ours and using 3D point clouds as input and training. In the experiment, since we cannot obtain the three-dimension data of the target object’s unobserved area from the camera location to calculate the 15-channel version of GPD, we only compare the 3-channel, 12-channel of GPD methods and PointNetGPD method. We train at different training set size for the same dataset (YCB dataset), and list the best grasp accuracy result and average grasp accuracy result among the 200 epochs in Table 1. We compare these representations by evaluating training set size and accuracy with which they can predict grasps.

| Training model | Training set size | Grasp accuracy          |
|----------------|-------------------|-------------------------|
|                |                   | Best result             | Average result          |
| Our method     | 0.1M              | 95.74%                  | 94.06%                  |
| PointNetGPD    | 1.6M              | 92.18%                  | 91.81%                  |
| GPD (12channel)| 3.64M             | 84.29%                  | 83.50%                  |
| GPD (3-channel)| 3.63M             | 82.50%                  | 81.38%                  |

From the above experimental results, we can draw the following conclusions. First, under the same dataset, our and PointNetGPD methods perform significantly better than all the GPD methods in grasp accuracy. Both of them have more than 90% grasp accuracy, and require less data for training than GPD. Furthermore, although we have similar results with the grasp accuracy of PointNetGPD, the performance of our method is better than PointNetGPD method in some fields, we need fewer training set size compared with PointNetGPD method in training set size. And the method we proposed has an average 3% improvement over the PointNetGPD baseline. As Table 1 illustrated that our proposed method manifests a higher grasp accuracy in best result and average result, which indicates that our
proposed grasping model can obtain more accurate graspable region of the target object based on the geometric characteristics of local point clouds.

4.2. Robotic experiment
To validate our algorithm, we use the Baxter robotic for grasping in the experiment. The robotic has 7-DOF arm and is equipped with electric parallel jaws with a width limit of 3 to 11cm. Each gripper is connected with two force sensors to measure the force between the two gripper arms. In order to obtain the 3D data of grasped object, we mount Kinect v1 sensor to the robot’s left wrist, and the robot’s right jaw for grasping (shown in Figure 8(a)). The robot communicates with the PC through the robot operating system (ROS), and collision-free inverse kinematics solver IK [24] is used for solving inverse kinematics.

We select 12 common household items, and 6 of which are contained in the grasp dataset, such as peach, chips can et al, while the rest are novel objects, such as shampoo, chess model et al. The objects are shown in Figure 8(b). Each object for grasping is placed on the desktop separately. In order to get as much multi-angle point cloud data as possible, when using depth camera to collect point cloud, the robotic arm always performs a collision-free trajectory movement [26] on a hemisphere centered on the target object. The radius of the hemisphere is 80cm and the arm rotates 100 degrees around the hemisphere center. At all times, the lens of depth sensor always points towards the center point of desktop workspace and the angle to the desktop is 50 degrees. We obtain real-time 3D reconstruction data of grasped scene through KinectFusion [27]. And we pretreatment the 3D points cloud of target object (shown in Figure 9(b)) by fitting a plane to the table using RANSAC [28] and extracting the point (shown in Figure 9(c)) above the table based on Euclidean distance clustering. Then, we use the pretreatment method and the grasping training neural network mentioned above to generate best grasps for testing. According to shape constraints, candidate with the highest grasping quality is selected as the best grasp (shown in Figure 9(e)). An example of grasping chess model is demonstrated in Figure 9.

In this experimental, we ran 120 clutter experiments. We performed grasp tests on the object shown in Figure 8(b). Each object is grasped 10 times, and the position and posture of object in each grasp are
random and different. In the grasping experiment, we defined a “successful raise” as the jaw grasped and raised the object 20cm without dropping it. And a grasp is successful if (1) the gripper successfully grasps the object and (2) it is a “successful raise”. Table 2 demonstrates the accuracy of grasping dataset objects and novel objects respectively.

Table 2. Accuracy of grasping dataset and novel objects

| Object type     | Accuracy |
|-----------------|----------|
| Dataset objects | 93.3%    |
| Novel objects   | 91.6%    |
| Total           | 92.5%    |

Among the 60 grasping of dataset objects, high grasping success accuracy 100% has been generated because the objects are regular, such as chips can and pudding box. There were several unsuccessful grasps when grasping peach and screwdriver, the narrow width of claw and low height of the acquired point cloud probably lead to several failed grasps. In the experiment of grasping novel objects, there were several failed grasps when grasping chicken model and elephant model, the poor quality and the irregular shape may lead to several failed grasps. It can be concluded that our algorithm can get robust when grasping regular shape objects. For irregular objects, we may get wrong grasping result, but the failed rate is limited. Overall the grasping success accuracy of robotic remains above 90%, which indicates that our proposed grasp model can effectively grasp objects.

5. Conclusion

In this paper, we train robotic grasping based on previous pretreatment work that first segmentation and simplification point cloud, and a large set of grasp candidates are obtained by grasping constraint based on pretreatment result and then the grasping classification is trained by neural network. We proposed three ways to increase the robotic understanding of target object. First, in order to avoid the singularity of object description and to make a more vivid description of object shape characteristics, we proposed a pretreatment method to apply the simplification of the basic form to the description of objects. Second, we propose a grasp strategy based on the basic shape and suggest to grasp the main parts of the object. Third, we build a new grasp description to measure the grasp similarity. Compared with current grasps quality classification methods, our experiments in this paper show an increase of about 3% in grasp accuracy and reach 94% grasp accuracy. This result emphasizes that the way we proposed to enhance the classification accuracy and robotic grasping accuracy.

6. References

[1] Mahler J, Pokorny F T and Hou B 2016 Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a Multi-Armed Bandit model with correlated rewards IEEE Int. Conf. on Robotics & Automation
[2] Mahler J, Liang J and Niyaz S 2017 Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics arXiv preprint
[3] Lenz I, Lee H and Saxena A 2015 Deep learning for detecting robotic grasps The Int. J. of Robotics Research 34(4-5) pp 705-724.
[4] Ten Pas A and Platt R 2015 Using Geometry to Detect Grasps in 3D Point Clouds Computer Science
[5] Johns E, Leutenegger S and Davison A J 2016 Deep Learning a Grasp Function for Grasping under Gripper Pose Uncertainty IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) pp 4461-68
[6] Yan X, Hsu J and Khansari M 2018 Learning 6-DOF Grasping Interaction via Deep Geometry-aware 3D Representations IEEE Int. Conf. on Robotics and Automation (ICRA) pp 1-9
[7] Ten Pas A, Gualtieri M and Saenko K 2017 Grasp Pose Detection in Point Clouds[J]. The International Journal of Robotics Research The Int. J. of Robotics Research 36(13-14) pp 1455-73
[8] Vahrenkamp N, Westkamp L and Yamanobe N 2016 Part-based grasp planning for familiar objects IEEE-RAS Int. Conf. on Humanoid Robots pp 919-925
[9] Stouraitis T, Hillenbrand U and Roa M A 2015 Functional power grasps transferred through warping and replanning IEEE Int. Conf. on Robotics & Automation pp 4933-40

[10] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel and Aaron M D 2017 Yale-CMU-Berkeley dataset for robotic manipulation research The Int. J. of Robotics Research 36 pp 261-268

[11] Goldfeder, Allen and Lackner 2007 Grasp Planning via Decomposition Trees IEEE Int. Conf. on Robotics & Automation pp 4679-84

[12] Lei Q and Wisse M 2017 Fast grasping of unknown objects using cylinder searching on a single point cloud 9th Int. Conf. on Machine Vision 10341 pp 1034108

[13] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen 2009 The columbia grasp database IEEE Int. Conf. on Robotics and Automation pp 1710-16

[14] Y. Li, J. L. Fu, and N. S. Pollard 2007 Data-driven grasp synthesis using shape matching and task-based pruning IEEE Trans. on Visualization and Computer Graphics 13 pp 732–747

[15] Tian H, Wang C and Manocha D 2019 Transferring Grasp Configurations using Active Learning and Local Replanning Int. Conf. on Robotics and Automation pp 1622-28

[16] Kai H, Ruthotto S and Kragic D 2008 Minimum volume bounding box decomposition for shape approximation in robot grasping IEEE Int. Conf. on Robotics & Automation pp 1628-33

[17] Aleotti and S. Caselli 2011 Part-based robot grasp planning from human demonstration IEEE Int. Conf. on Robotics and Automation pp 4554–4560

[18] Ten Pas A, Gualtieri M and Saenko K 2017 Grasp Pose Detection in Point Clouds The Int. J. of Robotics Research 36 pp 1455-73

[19] Gualtieri M, Ten Pas A and Saenko K 2016 High precision grasp pose detection in dense clutter IEEE/RSJ Int. Conf. on Intelligent Robots and Systems pp 598-605

[20] Liang H, Ma X and Li S 2018 PointNetGPD: Detecting Grasp Configurations from Point Sets Int. Conf. on Robotics and Automation pp 3629-35

[21] Liu H W, Cao C Q and Wang Y J 2019 Autonomous grasping method based on non-structural basic composition analysis Robot 41(05) pp 583-590

[22] Pickup D, Sun X and Rosin P L 2015 Canonical Forms for Non-Rigid 3D Shape Retrieval Eurographics Workshop on 3d Object Retrieval

[23] Haykin S and Kosko B 2009 Gradient Based Learning Applied to Document Recognition Wiley-IEEE Press

[24] Murray R M, Sastry S S and Zexiang L 2017 A Mathematical Introduction to Robotic Manipulation CRC Press

[25] Schulman J, Ho J and Lee A X 2013 Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization Robotics: science and systems 9(1) pp 1-10.

[26] S. Starke, N. Hendrich, D. Krupke, and J. Zhang 2017 Evolutionary Multi-Objective Inverse Kinematics on Highly Articulated and Humanoid Robots IEEE Int. Conf. on Intelligent Robots and Systems pp 6959-66

[27] Newcombe R A, Izadi S and Hilliges O 2012 KinectFusion: Real-Time Dense Surface Mapping and Tracking Ismar Basel Switzerland Oct 4(6) pp 127-136.

[28] Li L, Yang F and Zhu H 2017 An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells Remote Sensing 9(5) p 433

Acknowledgments
Our research has been supported in part by National Natural Science Foundation of China (No.61503186).