

RGB Matters: Learning 7-DoF Grasp Poses on Monocular RGBD Images

Minghao Gou¹, Hao-Shu Fang², Zhanda Zhu², Sheng Xu¹, Chenxi Wang¹ and Cewu Lu¹
Shanghai Jiao Tong University

Abstract—General object grasping is an important yet unsolved problem in the field of robotics. Most of the current methods either generate grasp poses with few DoF that fail to cover most of the success grasps, or only take the unstable depth image or point cloud as input which may lead to poor results in some cases. In this paper, we propose RGBD-Grasp, a pipeline that solves this problem by decoupling 7-DoF grasp detection into two sub-tasks where RGB and depth information are processed separately. In the first stage, an encoder-decoder like convolutional neural network *Angle-View Net*(AVN) is proposed to predict the $SO(3)$ orientation of the gripper at every location of the image. Consequently, a *Fast Analytic Searching*(FAS) module calculates the opening width and the distance of the gripper to the grasp point. By decoupling the grasp detection problem and introducing the stable RGB modality, our pipeline alleviates the requirement for the high-quality depth image and is robust to depth sensor noise. We achieve state-of-the-art results on GraspNet-1Billion dataset compared with several baselines. Real robot experiments on a UR5 robot with an Intel Realsense camera and a Robotiq two-finger gripper show high success rates for both single object scenes and cluttered scenes. Our code and trained model will be made publicly available.

I. INTRODUCTION

Grasping is a fundamental and important problem in robotics which is the basis for robotic manipulation. Despite its vital importance, solutions to this problem are far from satisfactory.

Traditional methods [4] [9] [5] utilize physical analysis to find suitable grasp poses. However, these methods require accurate object models which are not always available. It's also difficult to apply these algorithms to unseen objects. Besides, these methods are usually time consuming and computationally expensive.

With the development of computer vision and artificial intelligence, researchers propose data-driven or learning based methods to solve this problem. 2D planar grasping methods are firstly studied. Several datasets [17] [32] [10] are proposed and many algorithms [17] [20] [26] [35] [19] [3] [2] [7] learn to generate 2D planar grasp poses on those datasets. A few of them get a high accuracy on 2D grasping metrics. However, 2D planar grasping model imposes many restrictions on the grasp poses. The gripper can only approach the object in the top-down direction while in some cases it fails to grasp the object

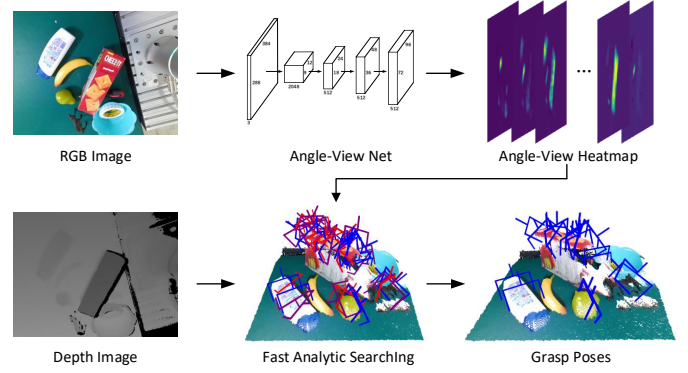


Fig. 1. An overview of our method. Top Left: The input RGB image. Top Middle: The *Angle-View Net*(AVN). Top Right: Output *Angle-View Heatmaps*(AVHs). Bottom Left: The input depth image. Bottom Middle: The collision and empty grasp detection based *Fast Analytic Searching*(FAS) module. Bottom Right: The output 7-DoF grasp poses.

along this direction. For example, it's very difficult for a gripper to grab a horizontally placed plate.

Recently, researchers begin to explore 6-DoF grasp poses detection. 6D pose estimation [31] [16] [42] allows 6-DoF grasping on known objects but cannot generalize to novel scene. GPD [29] and PointnetGPD [22] follow a two-step sampling-evaluation method. However, due to the unsatisfactory quality of the sampled results, a huge number of samples have to be evaluated to find reliable grasp poses which is time-consuming. Tian *et al.* [38], Florence *et al.* [14] and Patten *et al.* [30] transfer grasp poses from existing ones. But these methods fail when the objects are novel and the geometries are not similar to any existing one. Fang *et al.* [12], Ni *et al.* [28] and Mousavian *et al.* [27] feed partial-view point cloud captured by RGBD camera to neural networks to obtain 6-DoF grasp poses. However, depth data is not stable compared with RGB data because of potential sensor failure.

In this paper, we present RGBD-Grasp, a 7-DoF grasp detection pipeline. We take a monocular RGBD image as input and output the 6-DoF gripper pose alongside with the width of gripper as an additional DoF. As shown in Figure 1, we decouple the problem into two sub-problems. Using only the RGB image, we generate a group of heatmaps that indicate the pixel-wise orientation matrix of the gripper by an encoder-decoder network. Then we calculate the gripper opening width and the distance to the grasp point by collision and empty grasp detection guided fast analytic searching using the heatmap and the depth image.

¹{gmh2015, xs1020, wx1997, lucewu}@sjtu.edu.cn. Cewu Lu is the corresponding author, member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China

²{fhaoshu, zhandazhu}@gmail.com

The advantages of RGBD-Grasp are threefold. Firstly, it is the first attempt to generate high DoF grasp poses using a monocular RGBD image. The use of RGB image makes the method more stable and high DoF grasp poses solve the problem of planar restriction. Secondly, although the method is composed of two stages, each of the stages runs efficiently. In our experiment, it runs at 10.5 FPS which meets the requirement of real-time operation. Thirdly, benefiting from the deep CNN and large-scale data of GraspNet-1Billion dataset [12], the method not only works well with seen or similar objects, but it also generalizes to novel objects.

We evaluate our method on the GraspNet-1Billion benchmark [12]. Compared with existing methods, we improve the AP by **2.20**, **2.56** and **1.57** for seen, unseen and novel objects respectively for Kinect and **0.42**, **1.12** and **1.70** for RealSense and achieve the state-of-the-art. We also conduct real robot experiments for both single object scenes and cluttered scenes with a UR5 robot mounted with a RealSense camera. We report **91.7%** of the average success rate for single object scenes and **91.1%** for cluttered scenes.

II. RELATED WORKS

In this section, we focus on reviewing deep learning based grasping methods. Several review papers [11] [25] [18] [6] have made detailed analysis on the works in this field. According to the representation of grasp poses, previous works can be divided into mainly two types: 2D planar grasping and 6-DoF grasping.

A. 2D Planar Grasping

2D planar grasping means the gripper is constrained in the direction perpendicular to the camera plane. In this case, the grasp pose is often represented by an oriented rectangle which gives the location, orientation, width and height of the gripper from the top-down view. Jiang *et al.* [17] propose the Cornell dataset of 2D planar grasping and a searching based algorithm to find the best grasp pose. However, both the speed and performance are far from satisfactory. Lenz *et al.* [20] search over a large space to generate several candidates and use a CNN to process both RGB and depth images to evaluate them. The top ranked grasp is selected for execution. Mahler *et al.* [24] generate many 2D grasp candidates and evaluate them with GQ-CNN. Morrison *et al.* [26] propose GG-CNN to predict the quality and pose of grasps at each pixel. Redmon *et al.* [35] and Kumra *et al.* [19] directly regress the parameters for 2D planar grasp and grid the picture to obtain multiple grasp poses. Asif *et al.* [3] propose an encoder-decoder network to generate an affordance map and calculate the 2D grasp poses accordingly. Inspired by Region Proposal Network [36], Chu *et al.* [7] propose Grasp Proposal Network to detect a lot of grasps on multi-object images. Asif *et al.* [2] assemble three different types of networks and switch among them according to their confidence scores. Although the reported scores in these papers are high, the constraint of 2D planar grasp is strong so that many suitable grasps cannot be represented in this format, which limits their application.

B. 6-DoF Grasping

For 6-DoF grasping methods, the gripper can grasp objects from arbitrary directions. **Six parameters are needed to specify the 3D location and rotation. Additional DoFs such as the width or height of the gripper may also be part of the representation.** GPD [29] and PointnetGPD [22] sample grasps in the space with constraint and then use LeNet or Pointnet to evaluate these samples respectively. Mousavian *et al.* [27] sample grasp poses using a variational autoencoder and then assess and refine the sampled poses. S⁴G [34] takes a partial view point cloud as input and regresses the 6-DoF poses directly. Florence *et al.* [14], DGCM-Net [30] and Tian *et al.* [38] transfer existing grasp poses from one object to another. Yan *et al.* [41], Lundell *et al.* [23] and Varley *et al.* [39] firstly reconstruct the scene from partial view point cloud and then plan grasps on the completed scene. Recently, Fang *et al.* [12] propose the first large-scale dataset in this area and decouple the grasp poses estimation into learning approaching directions and operation parameters. Ni *et al.* [28] directly regress 6D poses after extracting the features of objects using Pointnet++ [33]. 6-DoF grasping methods eliminate the restriction on the degree of freedom and generate grasp poses with higher quality. However, most of these methods only make use of the partial view point cloud and ignore the color information, while the quality of point cloud is unstable. For dark, transparent and reflective materials, the obtained depth image can be very inaccurate and may lead to failure.

We solve this problem by incorporating both RGB and depth information. With the aid of RGB images, we can alleviate the demand for high quality depth images. Experiments in Section. VI demonstrate that our method is robust to depth noise and domain variation.

III. PROBLEM STATEMENT

A. Definitions

a) *Grasp Pose*: Grasp pose \mathcal{P} is defined by a tuple:

$$\mathcal{P} = (x, y, z, r_x, r_y, r_z, w) \quad (1)$$

in which x , y and z denote the translation of the gripper while r_x , r_y and r_z denote the rotation and w denotes the width accordingly.

b) *RGBD Image*: RGBD Image \mathcal{I} is represented by a tuple:

$$\mathcal{I} = (\mathcal{C}, \mathcal{D}) \quad (2)$$

in which $\mathcal{C} = \mathbb{R}^{3 \times H \times W}$ denotes the RGB image and $\mathcal{D} = \mathbb{R}^{H \times W}$ denotes the depth map.

c) *Gripper Configuration*: In this paper, we only take the most common parallel-jaw gripper into consideration. The gripper configuration \mathcal{G} is define by a tuple:

$$\mathcal{G} = (h, l, w_{max}) \quad (3)$$

in which h , l and w_{max} are the height, length and maximum width of the gripper respectively as shown in Figure 2.

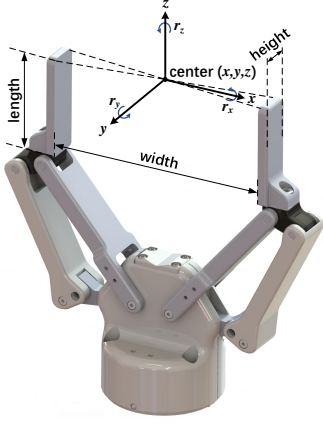


Fig. 2. The schematic diagram of a gripper. The length and height of the gripper are fixed according to the gripper configuration. Three translation coordinates, three rotation angles and the width are 7 variables to represent the pose of the gripper.

B. Problem Statement

We make a formal statement of our problem based on the definitions given in Section III-A. Let \mathcal{E} denote the environment including robot and objects, and $s(\mathcal{E}, \mathcal{I}, \mathcal{G}, \mathcal{P})$ denote a binary variable indicating grasp success or not. A grasp is successful if the object is lifted successfully. Given an RGBD image \mathcal{I} and a gripper configuration \mathcal{G} , our goal is to find a set of grasp poses $\mathbf{P} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_k\}$ that maximizes the grasp success rate given a fixed k :

$$\{\mathcal{P}_1^*, \mathcal{P}_2^*, \dots, \mathcal{P}_k^*\} = \underset{|\mathbf{P}|=k}{\operatorname{argmax}} \sum_{\mathcal{P}_i \in \mathbf{P}} \operatorname{Prob}(s = 1 | \mathcal{E}, \mathcal{I}, \mathcal{G}, \mathcal{P}_i). \quad (4)$$

This means that we hope our algorithm to predict abundant grasp poses to cover the whole scene, so that we can have different candidates for grasp execution.

IV. METHOD

A. Overview

As illustrated before, we decouple our problem into two sub-problems. Firstly, *Angle-View Net (AVN)* generates the gripper orientations in different positions of the image:

$$\mathcal{P}_{img} = (u, v, r_x, r_y, r_z, c), \quad (5)$$

where (u, v) is a location in image coordinate and c is the success confidence of each prediction. Secondly, we collect those predictions with high confidence scores and calculate their widths and distances to the image plane, given camera intrinsic parameters and a depth image based on *Fast Analytic Searching (FAS)*:

$$\mathcal{P}_{cam} = (x, y, z, r_x, r_y, r_z, w) \quad (6)$$

B. Angle-View Net

The AVN predicts the pixel-wise gripper rotation configurations. One naive way to learn the orientations is to directly regress the rotation matrices or quaternions. However, one location may have several feasible rotations to achieve robust

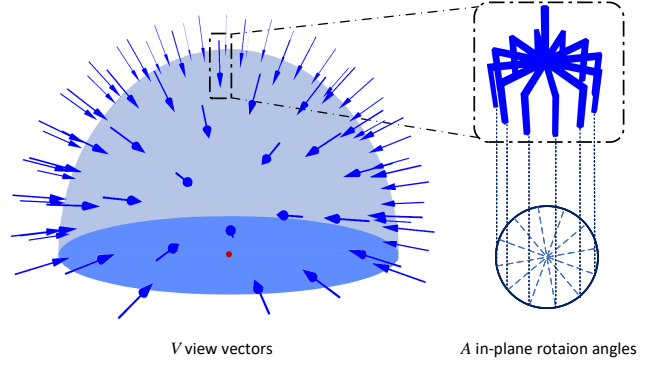


Fig. 3. The orientation of the gripper is defined by a view vector and an in-plane rotation angle. As shown in the left part of the figure, V view vectors are uniformly sampled in the upper hemisphere. And as shown in the right part of the figure, A in-plane rotation angles are sampled. In this figure, V and A are set to 120 and 6 respectively.

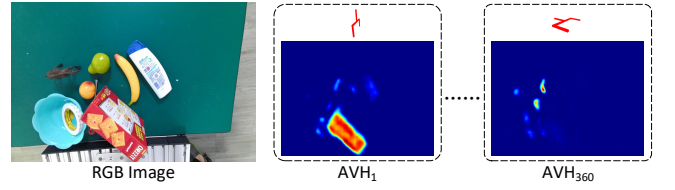


Fig. 4. The illustration of two examples of AVH. The left is the original RGB image. The figures on the right are the example AVHs. The corresponding orientation of the gripper is shown above each AVH.

grasping, which makes the regression mechanism impractical.

Instead, inspired by GraspNet-1Billion [12], we decouple the orientation into the approaching direction and in-plane rotation and treat it as a multi-class classification problem. As shown in Figure 3, we uniformly sample V views from the upper hemisphere and A in-plane rotation angles. As a result, there are a total of $V \times A$ classes of orientations.

To obtain the gripper orientations at different locations, we grid the image into $G_H \times G_W$ grids with G_H grids along the vertical direction and G_W grids along the horizontal direction. For each grid, the AVN predicts a 1-dim vector with $V \times A$ elements which indicate the confidence scores of each orientation class in that grid. We denote the final output of AVN as the *Angle-View Heatmap (AVH)* which is defined by a 3D tensor:

$$\mathbf{AVH} = \mathbb{R}^{(V \times A) \times G_H \times G_W}. \quad (7)$$

An example of AVH is given in Figure 4. To learn the mapping from \mathbf{C} to AVH: $\mathbb{R}^{3 \times H \times W} \rightarrow \mathbb{R}^{(V \times A) \times G_H \times G_W}$, we refer to AlphaPose [13, 21] and adopt an encoder-decoder like structure. As illustrated in Figure 5, a ResNet50 [15] first encodes the image to high dimensional features. Then a pixel-shuffle layer [37] and two dense upsampling convolution (DUC) layers [40] decode the features to the AVH. Let \mathbf{AVN}_θ denotes AVN with the weights of θ . Given the dataset $\{(\mathbf{C}_1, \mathbf{AVH}_1), \dots, (\mathbf{C}_n, \mathbf{AVH}_n)\}$ with $(\mathbf{C}_i, \mathbf{AVH}_i)$ the i^{th} pair of the RGB image and ground truth AVH, the network

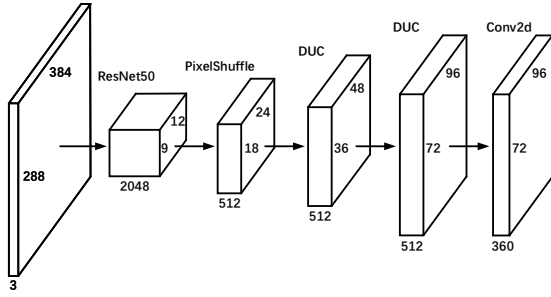


Fig. 5. The structure of the Angle-View Net (AVN). The network takes an RGB image as input and passes the image to Resnet to extract dense features. After that, pixel shuffle and Dense Upsampling Convolution (DUC) layers are used to upsample the features. Finally it generates the **AVH**.

can be trained by minimizing the L2 loss function \mathcal{L} :

$$\theta = \arg \min_{\theta} \sum_i \mathcal{L}(\text{AVN}_{\theta}(\mathbf{C}_i), \mathbf{AVH}_i), i = (1, 2, \dots, n) \quad (8)$$

The acquisition of ground truth **AVH** for an image will be detailed in Section. V. Compared with those who randomly sample poses, the quality of the generated candidates from **AVN** are higher and the computing cost for filtering is lower.

C. Fast Analytic Searching

AVN identifies 5 of the 7 DoF of the grasp pose. The width of the gripper and the distance of the gripper to the camera plane remain to be solved. We propose collision and empty grasp detection based Fast Analytic Searching to calculate the width and distance.

One naive way to obtain the width and the distance is learning them by regression or classification. However, learning faces two problems. First, both the width and the distance are dimensional quantities while the RGB values of the image are dimensionless. Thus it's hard to learn the width and the distance without any information of the scale of the image. For example, two images taken at different scales may have the same RGB values. Second, width and distance estimating is closely related to the camera intrinsic parameters. Trained network is hard to generalize to images captured by camera with different intrinsic parameters.

Thus instead, we solve the problem by filtering unreasonable grasp poses. As shown in Figure 6, for width estimation, given a gripper configuration \mathcal{G} , we uniformly sample W widths from 0 to w_{max} . For distance, we assume that the gripper should be close to the corresponding point on partial-view point cloud reconstructed by the depth image. So we sample D distances from the location below the point to location above the point. For each sampled width and distance together with \mathcal{G} and **AVH**, the gripper can be accurately modeled. As shown in Figure 6, two kinds of grasp poses are filtered out by the rules below.

- 1) There are points in the gripper occupied space.
- 2) There is no point in the grasping space.

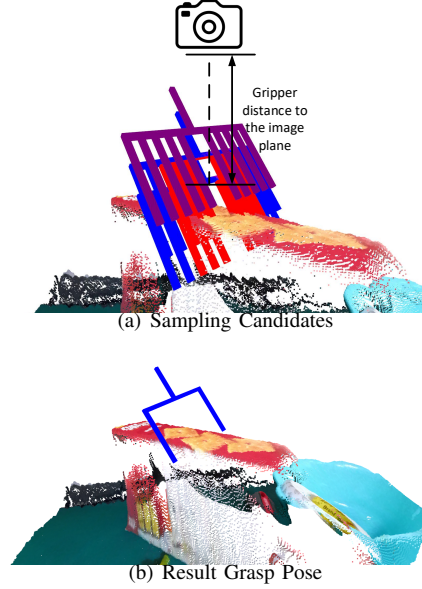


Fig. 6. The illustration of the **FAS** Module. Top Figure: The point cloud is reconstructed by the depth image. Candidates with different widths and different distances to the image plane are sampled. We conduct collision and empty checking on these candidates. Red colored candidates collide with the scene point cloud, which violates the first rule. Purple colored candidates have no point in their grasping spaces, which violates the second rule. Blue colored candidates are the remaining ones and considered good grasp poses. Bottom Figure: Among all the good grasp poses, we conduct grasp pose non-maximum suppression to find the one with the largest distance and smallest width.

V. IMPLEMENTATION DETAILS

A. Dataset

As mentioned above, we need to generate RGB image and corresponding ground truth **AVH** pairs for training **AVN**. As discussed in Section I, Jiang *et al.* [17] and Jacquard [10] provide the 2D planar grasping datasets. However, all the grasps in these datasets approach the objects in the top-down direction. They are unable to define grasps in all directions. Besides, annotations in these datasets are sparse, which lead to bad performance in generating the ground truth **AVH**.

We adopt the GraspNet-1Billion [12] dataset. To the best of our knowledge, this is the only publicly available 6-DoF grasping dataset. It is also easy to generate the ground truth **AVH** taking the advantage of the dense annotations.

The ground truth **AVHs** are generated by two steps. Firstly, we initialize an empty $\mathbf{AVH} = \mathbf{0}^{(V \times A) \times G_w \times G_H}$ that all the possible angles, views and places are labeled negative. Secondly, for each ground truth grasp pose in the GraspNet-1Billion dataset, we calculate the closest angle, view and place and then label it as positive. For all the 100 training scenes in GraspNet-1Billion dataset, there are about 110 million positive samples and 64 billion negative samples for Kinect and 143 million positive sample and 64 billion negative samples for Intel RealSense captured images.

B. Network and Training

As shown in Figure 5, for our network, the input RGB image size is 384×288 and the output **AVH** size is 96×72 .

We uniformly sample $V = 60$ views in the upper hemisphere and $A = 6$ angles, generating 360 heatmaps in total. In **FAS**, the widths are sampled from 1cm to 10cm with the step of 1cm. Depths from the place 2cm below the origin point to the place 2cm above the point with the step of 1cm are explored.

We train the network using two NVIDIA 2080Ti GPUs with the batchsize of 16 and learning rate of 10^{-3} with the ADAM optimizer. The learning rate is decayed by 10 after each 20000 batches. The network is initialized with ImageNet pretrained weights. We train on both Kinect and Intel RealSense camera captured images for 60000 batches. During training, we conduct heavy data augmentation including random crop and color jittering to avoid over-fitting.

VI. EXPERIMENTS

In this section, we show the experiments carried on both GraspNet-1Billion dataset and a UR-5 robot. We compare our results with several baselines and analyze the differences.

A. Evaluation Metric

First, we conduct experiments on GraspNet-1Billion dataset. We evaluated our result on the dataset using the standard metric [12]. This metric first conducts grasp pose non-maximum suppression(GPNMS) [1] to avoid duplicated grasp poses in a small region. Then, for the top k candidates, it labels those collided grasp poses as negative predictions based on the reconstructed scene. A binary label is assigned to the remaining grasp poses given different friction coefficients by force-closure metric [24]. Finally, it calculates the **AP** by finding the mean average precision of different friction coefficients and different k range from 1 to 50.

B. Results on GraspNet-1Billion

1) *Baselines*: We adopt GG-CNN [26], Chu *et al.* [7], Fang *et al.* [12] as baselines. We compare both the quantitative results and visualization results to show the effectiveness of our RGBD-Grasp. For GG-CNN [26] and Chu *et al.* [7], we first convert the GraspNet-1Billion data into the Cornell dataset [17] format by selecting top-down grasps and projecting them on the image. Then, for Chu *et al.* [7], we retrain the network using the converted GraspNet-1Billion dataset. For GG-CNN [26], the retrained model gets poor performance and we use the pretrained model on Cornell dataset for evaluation instead. To show the effectiveness of our **AVN**, we also conduct an ablation study by removing it from our pipeline and randomly sample candidates for **FAS**.

To verify the robustness of our method towards depth images with poor quality, we randomly add some Gaussian noise on the depth images and test the performances of RGBD-Grasp and Fang *et al.* [12]. Besides, we have also conducted cross-domain testing by training the network on one of the two cameras between Kinect and RealSense and evaluating the result on the other one. This is to verify the cross domain generalization ability of algorithms.

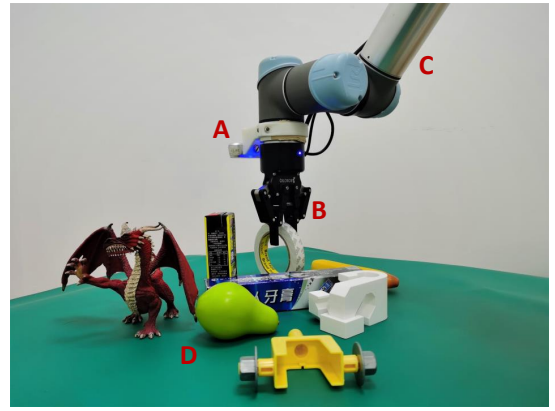


Fig. 7. Real robot experiments on a cluttered scene. A: Intel RealSense RGBD camera. B: Robotiq two-finger gripper. C: UR-5 Robot. D: Objects from GraspNet 1-Billion dataset. The robot executes the best grasp pose output by RGBD-Grasp and the objects are grabbed one by one.

2) *Results and Analysis*: The scores of several methods on the GraspNet-1Billion metric [12] are shown in Table I and some of the examples are visualized in Figure 8. The definitions of **APs** can be found in [12]. As for 2D planar grasping methods [26] [7], the restriction on DoF greatly harms their performances. And for random sampling, it outputs few reasonable grasp poses as the orientations of the grasps are randomly selected even though the **FAS** module filters some of the bad grasp poses.

RGBD-Grasp slightly outperforms Fang *et al.* on the original dataset. Moreover, when it comes to feeding depth images with noise, the score obtained by Fang *et al.* drops significantly while RGBD-Grasp only gets a small decrease.

As shown in Table II. RGBD-Grasp performs well even though it's trained on a different camera. However, Fang *et al.* relies heavily on the depth image and as a result gets low scores when evaluated in a cross-camera setting.

C. Real Robot Experiments

We also conduct real robot experiments. As shown in Figure 7, the experiment is carried on a UR-5 robot mounted with an Intel RealSense camera and a Robotiq two-finger gripper. A computer with an NVIDIA GTX 2060 GPU, an Intel i7-8750H CPU, 8G RAM and an Ubuntu 18.04 operating system is used to run our pipeline and control the robot. The trajectory is planned using MoveIt [8] to avoid collision among the robot, gripper, camera and objects.

1) *Experiment on Single Object Scenes*: We randomly select 9 objects from GraspNet-1Billion dataset and place them on the table at random place with random orientation. Objects with different difficulties are selected to analyze the performance of generalization. We conduct 20 experiments for each object and calculate the success rate. The results are shown in Table III.

2) *Experiment on Cluttered Scenes*: We also conduct experiments on cluttered scenes. In each scene, we randomly select 4-8 objects from GraspNet-1Billion and randomly place them on the table. We find and execute the best grasp pose candidate using the pipeline until all the objects are

Methods	Seen			Unseen			Novel		
	AP	AP _{0.8}	AP _{0.4}	AP	AP _{0.8}	AP _{0.4}	AP	AP _{0.8}	AP _{0.4}
Random	10.87/11.44	12.97/13.43	9.66/10.01	10.45/11.10	12.22/12.98	8.94/9.76	4.90/5.54	5.76/6.76	2.87/3.16
GG-CNN [26]	15.48/16.89	21.84/22.47	10.25/11.23	13.26/15.05	18.37/19.76	4.62/6.19	5.52/7.38	5.93/8.78	1.86/1.32
Chu <i>et al.</i> [7]	15.97/17.59	23.66/24.67	10.80/12.74	15.41/17.36	20.21/21.64	7.06/8.86	7.64/8.04	8.69/9.34	2.52/1.76
Fang <i>et al.</i> [12]	27.56/29.88	33.43/36.19	16.95/19.31	26.11/27.84	34.18/33.19	14.23/16.62	10.55/11.51	11.25/12.92	3.98/3.56
[12] with noise	22.35/24.35	27.96/30.34	12.23/14.24	21.12/21.34	28.38/27.90	11.23/12.30	7.23/8.10	8.02/9.87	2.52/2.23
Ours with noise	26.34/28.98	32.78/35.91	15.84/18.40	27.01/29.01	35.35/34.21	15.10/17.32	10.95/11.64	11.15/12.62	4.53/4.68
Ours	27.98/32.08	33.47/39.46	17.75/20.85	27.23/30.40	36.34/37.87	15.60/18.72	12.25/13.08	12.45/13.79	5.62/6.01

TABLE I

EVALUATION RESULT FOR DIFFERENT METHODS. THE TABLE SHOWS THE RESULTS ON DATA CAPTURED BY REALSENSE/KINECT RESPECTIVELY.

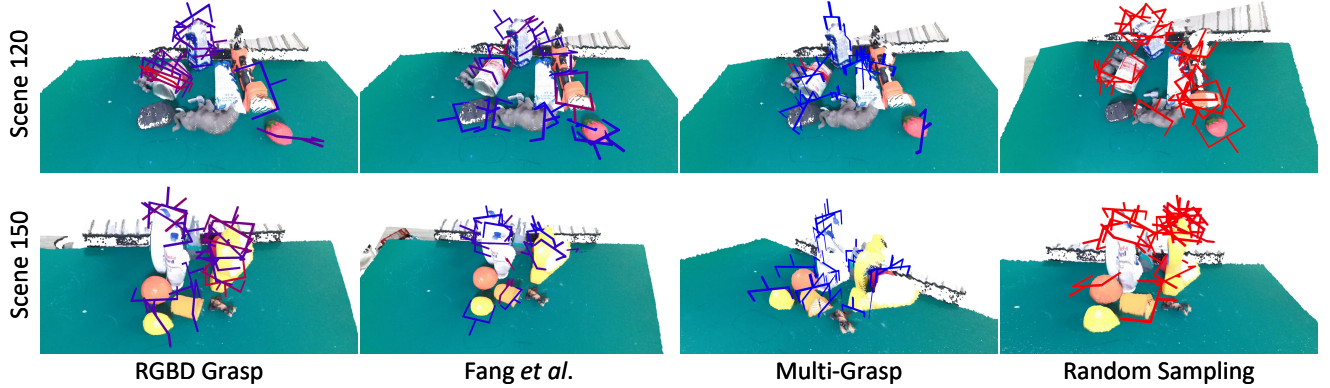


Fig. 8. Example visualization results of four different algorithms on scene 120 and scene 150 in GraspNet-1Billion dataset for Kinect captured image. The color of the gripper tells the confidence score for each grasp. For more confident grasp pose, the gripper color is closer to red.

Methods	Train on	Eval on	Seen	Similar	Novel
Fang <i>et al.</i>	Kinect	Kinect	29.88	27.84	11.51
Fang <i>et al.</i>	RealSense	Kinect	22.00	21.52	8.40
Fang <i>et al.</i>	Realsense	RealSense	27.56	26.11	10.55
Fang <i>et al.</i>	Kinect	RealSense	20.19	18.97	7.97
RGBD Grasp	Kinect	Kinect	32.08	30.40	13.08
RGBD Grasp	RealSense	Kinect	27.25	26.54	11.25
RGBD Grasp	Realsense	RealSense	27.98	27.23	12.25
RGBD Grasp	Kinect	RealSense	26.65	27.81	13.15

TABLE II

CROSS DOMAIN TESTING RESULTS.

ID	Name	Type	Attempt	Success	Success Rate
1	004_sugar_box	seen	20	19	95%
5	011_banana	seen	20	20	100%
12	013_apple	seen	20	19	95%
38	dabao_sod	similar	20	19	95%
39	soap_box	similar	20	19	95%
43	baoke_marker	similar	20	18	90%
72	peeler_cover	novel	20	17	85%
33	dragon	novel	20	18	90%
74	ice_cube_mould	novel	20	16	80%
-	Average	-	20	18.33	91.67%

TABLE III

RESULTS OF REAL ROBOT EXPERIMENTS ON SINGLE OBJECT SCENES

taken away. We define the success rate of cluttered-scene experiments by:

$$\text{success rate} = \frac{\text{number of objects}}{\text{number of attempts}}$$

The experiment results are shown in Table IV. We can see that RGBD-Grasp can achieve satisfactory success rate.

Objects ID	Number	Attempts	Success Rate
5, 15, 31, 33, 58, 68, 70, 75	8	8	100.0%
1, 2, 4, 5, 13	5	6	83.3%
17, 21, 26, 24, 32, 35, 36, 37	8	10	80.0%
35, 36, 47, 50, 55, 59	6	6	100.0%
5, 15, 26, 50, 72, 73, 77	7	8	87.5%
6, 13, 19, 21, 68, 70, 74	7	7	100.0%
Average	6.83	7.5	91.1%

TABLE IV

RESULTS OF REAL ROBOT EXPERIMENTS ON CLUTTERED SCENES

VII. CONCLUSION

In this paper, we propose a novel 7-DoF grasp pose detection pipeline named RGBD-Grasp. It decouples the problem of finding grasp poses into two sub-questions. The *Angle-View Net*(AVN) module takes a monocular RGB image as input and generates heatmaps to predict both grasping locations in the image plane and the orientations of the gripper. The *Fast Analytic Searching*(FAS) module takes the heatmaps and a depth image as input and outputs the widths and distances from the gripper centers to the image plane. We achieve state-of-the-art results on GraspNet-1Billion benchmark. We also conduct experiments using a UR-5 robot with a Robotiq two-finger gripper on both single object scenes and cluttered scenes. Both experiments report high success rates and therefore verify the effectiveness of RGBD-Grasp.

VIII. ACKNOWLEDGMENT

This work is supported in part by the National Key R&D Program of China, No. 2017YFA0700800, National Natural

Science Foundation of China under Grants 61772332, Shanghai Qi Zhi Institute, SHEITC(2018-RGZN-02046) and Baidu Fellowship.

REFERENCES

- [1] <https://graspnet.net/evaluation.html>. 2020.
- [2] Umar Asif, Jianbin Tang, and Stefan Herrer. “Densely supervised grasp detector (DSGD)”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 8085–8093.
- [3] Umar Asif, Jianbin Tang, and Stefan Herrer. “GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices.” In: *IJCAI*. 2018, pp. 4875–4882.
- [4] Antonio Bicchi and Vijay Kumar. “Robotic grasping and contact: A review”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 1. IEEE. 2000, pp. 348–353.
- [5] J. Bohg et al. “Data-Driven Grasp Synthesis—A Survey”. In: *IEEE Transactions on Robotics (T-RO)* 30.2 (2014), pp. 289–309.
- [6] Shehan Caldera, Alexander Rassau, and Douglas Chai. “Review of deep learning methods in robotic grasp detection”. In: *Multimodal Technologies and Interaction* 2.3 (2018), p. 57.
- [7] Fu-Jen Chu, Ruinian Xu, and Patricio A Vela. “Real-world multiobject, multigrasp detection”. In: *IEEE Robotics and Automation Letters (RA-L)* 3.4 (2018), pp. 3355–3362.
- [8] David Coleman et al. “Reducing the barrier to entry of complex robotic software: a moveit! case study”. In: *arXiv preprint arXiv:1404.3785* (2014).
- [9] Hao Dang and Peter K Allen. “Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 1311–1317.
- [10] A. Depierre, E. Dellandréa, and L. Chen. “Jacquard: A Large Scale Dataset for Robotic Grasp Detection”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 3511–3516.
- [11] Guoguang Du et al. “Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review”. In: *Artificial Intelligence Review* (2020), pp. 1–58.
- [12] Hao-Shu Fang et al. “GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11444–11453.
- [13] Hao-Shu Fang et al. “Rmpe: Regional multi-person pose estimation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017.
- [14] Peter Florence, Lucas Manuelli, and Russ Tedrake. “Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation”. In: *Conference on Robot Learning (CoRL)* (2018).
- [15] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2016, pp. 770–778.
- [16] Yisheng He et al. “Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11632–11641.
- [17] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. “Efficient grasping from rgb-d images: Learning using a new rectangle representation”. In: *2011 IEEE International conference on robotics and automation (ICRA)*. IEEE. 2011, pp. 3304–3311.
- [18] Kilian Kleeberger et al. “A Survey on Learning-Based Robotic Grasping”. In: *Current Robotics Reports* (2020), pp. 1–11.
- [19] Sulabh Kumra and Christopher Kanan. “Robotic grasp detection using deep convolutional neural networks”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 769–776.
- [20] Ian Lenz, Honglak Lee, and Ashutosh Saxena. “Deep learning for detecting robotic grasps”. In: *The International Journal of Robotics Research (IJRR)* 34.4-5 (2015), pp. 705–724.
- [21] Jiefeng Li et al. “Crowdpose: Efficient crowded scenes pose estimation and a new benchmark”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10863–10872.
- [22] Hongzhuo Liang et al. “Pointnetgpd: Detecting grasp configurations from point sets”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3629–3635.
- [23] J. Lundell, F. Verdoja, and V. Kyrki. “Robust Grasp Planning Over Uncertain Shape Completions”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 1526–1532.
- [24] Jeffrey Mahler et al. “Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics”. In: *Proceedings of Robotics: Science and Systems (RSS)*. Cambridge, Massachusetts, July 2017. DOI: 10.15607/RSS.2017.XIII.058.
- [25] M. Q. Mohammed, K. L. Chung, and C. S. Chyi. “Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations”. In: *IEEE Access* 8 (2020), pp. 178450–178481.
- [26] Douglas Morrison, Peter Corke, and Jurgen Leitner. “Closing the loop for robotic grasping: A real-time,

- generative grasp synthesis approach”. In: *Robotics: Science and Systems XIV* (2018), pp. 1–10.
- [27] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. “6-dof graspnet: Variational grasp generation for object manipulation”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 2901–2910.
 - [28] P. Ni et al. “PointNet++ Grasping: Learning An End-to-end Spatial Grasp Generation Algorithm from Sparse Point Clouds”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 3619–3625.
 - [29] Andreas ten Pas et al. “Grasp pose detection in point clouds”. In: *The International Journal of Robotics Research (IJRR)* 36.13-14 (2017), pp. 1455–1473.
 - [30] Timothy Patten, Kiru Park, and Markus Vincze. “DGCM-Net: Dense Geometrical Correspondence Matching Network for Incremental Experience-Based Robotic Grasping”. In: *Frontiers in Robotics and AI* 7 (2020), p. 120. ISSN: 2296-9144.
 - [31] Sida Peng et al. “Pvnet: Pixel-wise voting network for 6dof pose estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4561–4570.
 - [32] Lerrel Pinto and Abhinav Gupta. “Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours”. In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 3406–3413.
 - [33] Charles Ruizhongtai Qi et al. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in neural information processing systems (NeurIPS)*. 2017, pp. 5099–5108.
 - [34] Yuzhe Qin et al. “S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes”. In: *Conference on robot learning (CoRL)*. PMLR. 2020, pp. 53–65.
 - [35] Joseph Redmon and Anelia Angelova. “Real-Time Grasp Detection Using Convolutional Neural Networks”. In: *Proceedings IEEE International Conference on Robotics and Automation (ICRA)* (2014).
 - [36] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems (NeurIPS)*. 2015, pp. 91–99.
 - [37] Wenzhe Shi et al. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
 - [38] Hao Tian et al. “Transferring Grasp Configurations using Active Learning and Local Replanning”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 1622–1628.
 - [39] Jacob Varley et al. “Shape completion enabled robotic grasping”. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 2442–2447.
 - [40] Panqu Wang et al. “Understanding convolution for semantic segmentation”. In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2018, pp. 1451–1460.
 - [41] Xinchun Yan et al. “Learning 6-dof grasping interaction via deep geometry-aware 3d representations”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–9.
 - [42] Zelin Zhao et al. “Estimating 6d pose from localizing designated surface keypoints”. In: *arXiv preprint arXiv:1812.01387* (2018).