REGNet: REgion-based Grasp Network for Single-shot Grasp Detection in Point Clouds

Binglei Zhao, Hanbo Zhang, Xuguang Lan, Haoyu Wang, Zhiqiang Tian and Nanning Zheng

***Abstract*— Learning a robust representation of robotic grasping from point clouds is a crucial but challenging task. In this paper, we propose an end-to-end single-shot grasp detection network taking one single-view point cloud as input for parallel grippers. Our network includes three stages: Score Network (SN), Grasp Region Network (GRN) and Refine Network (RN). Specifically, SN is designed to select positive points with high grasp confidence. GRN coarsely generates a set of grasp proposals on selected positive points. Finally, RN refines the detected grasps based on local grasp features. To further improve the performance, we propose a grasp anchor mechanism, in which grasp anchors are introduced to generate grasp proposals. Moreover, we contribute a large-scale grasp dataset without manual annotation based on the YCB dataset. Experiments show that our method significantly outperforms several successful point-cloud based grasp detection methods including GPD, PointnetGPD, as well as S^4G.**

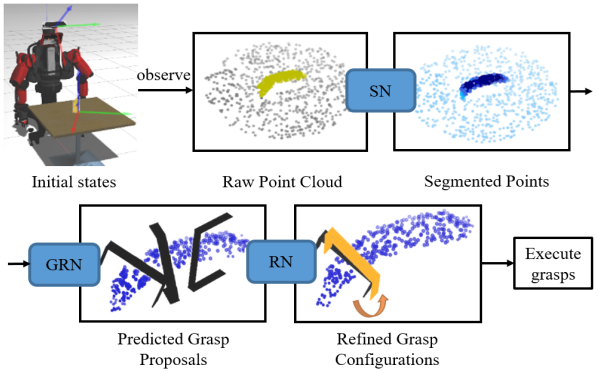

Fig. 1. Illustration of the pipeline in this paper. SN uses the point cloud as input and outputs segmented points. Then GRN predicts grasps from grasp regions. After RN refining the predicted grasps, we execute grasping.

I. Introduction

Robotic grasping is a widely and actively investigated field in robotics since it plays a crucial and fundamental role in manipulation and interaction with the outside world. However, reliable robotic grasping in real-world scenarios is still a challenging task due to the uncertainty caused by unstructured environments, various object geometries, and sensor noise. Most grasp detection algorithms aim at generating stable grasp configurations with high-quality scores. Nevertheless, their performance is far behind human beings.

Traditional methods utilizing physical analysis [6], [14] can generate proper grasps, but it's difficult to obtain grasps on generic objects without 3D models. Recent results show the superiority of data-driven methods that can predict grasps on unseen objects [33]. In [8], [10], the grasp represented by a rectangle is detected through a network based on RGB or RGD images. Nevertheless, due to lacking consideration of geometric information and grasp quality metrics, they struggle to find the optimal grasp and limit the way of grippers contacting objects (e.g. grippers usually perpendicular to the table). On the other hand, some methods [16], [17] assess quality scores of grasp candidates by classification networks. However, the running time will go up quickly as the number of candidates increases. Recently, S^4G [18] directly regresses grasps using single-point features based on PointNet++ instead of sampling and evaluation. Since PointNet++ does not explicitly model the local spatial distribution of points, the features contain less shape awareness [31], [32]. When grasping is performed on an area, the less shape awareness leads to less accurate grasp detection.

In this paper, we present an end-to-end grasp detector to address the problems. As illustrated in Fig. 1, our network consists of three stages: Score Network (SN), Grasp Region Network (GRN) and Refine Network (RN). Specifically, at the beginning, we utilize PointNet++ [4] to extract features from the single-view point cloud. Then **SN** predicts the point grasp confidence from the features to assess whether each point is suitable as a grasp center and segments the points into negative and positive. Afterwards, **GRN** regresses grasps from grasp regions, which are spheres centered on positive points. Compared to a single point, the grasp region provides a larger receptive field, leading to more effective local feature aggregation. Since most grasp orientation is closely relevant to the grasp center and is only feasible in a small range, we introduce the grasp anchor mechanism that predefines grasp anchors with assigned orientations as the reference to more effectively predict grasps in GRN. To generate more accurate grasps, **RN** refines proposals using the fusion features of points within the predicted grasps and the grasp regions.

In most cases, data-driven methods require grasp datasets for training. Many manually labeled datasets, such as Cornell [28] and VMRD [23] dataset, include no specific information about grasp quality metrics. In contrast, PointNetGPD [17] and Dex-net [11] utilize physics analysis to generate grasps. However, they include grasps with quality metrics, but don't

*This work was supported in part by the key project of Trico-Robot plan of NSFC under grant No.91748208, National Science and Technology Major Project under grant No. 2018ZX01028-101, key project of Shaanxi province under grant No.2018ZDCXLGY0607, and NSFC under grant No.61973246.

Xuguang Lan is with the Institute of Artificial Intelligence and Robotics, the National Engineering Laboratory for Visual Information Processing and Applications, School of Electronic and Information Engineering, Xi'an Jiaotong University, No.28 Xianning Road, Xi'an, Shaanxi, China. xglan@mail.xjtu.edu.cn

include points with grasp confidence, which indicates the grasp probability of a point as the grasp center. The grasp confidence is important to learn which area is suitable for grasping. Therefore, we propose a method to calculate the point grasp confidence and construct a large-scale grasp dataset with the grasp confidence based on YCB dataset [20].

Another issue is how to evaluate performance. Jaccard Index used in previous works is not suitable for 3D space since a large overlap area cannot ensure a stable grasping. Moreover, it is computationally complex. On the other hand, some methods use the classification accuracy for performance evaluation [11], [17], which cannot directly evaluate the detection performance and is diffiult to extend to more algorithms. Therefore, we present the Valid Grasp Ratio (VGR), which is easy to compute and intuitively describes the possibility of the predicted grasp to be high-quality. Moreover, it is convenient to be transfered to other algorithms, showing its potential as a benchmark for 3D space grasp detection. Our model significantly outperforms other methods and can be generalized to unseen objects. Our contributions could be summarized as follows.

- Our proposed single-shot region-based grasp network is a state-of-the-art method for grasp detection in 3D space, which outperforms several successful methods including GPD, PointnetGPD, as well as S^4G.
- We generate a large-scale grasp dataset in 3D space with point grasp confidence to evaluate the grasp probability of points as the grasp centers.
- We present the metric of VGR to assess the performance of grasp detection in 3D space, which can be conveniently transferred to other algorithms.

II. Related Work

A. Grasp Detection

Existing grasp detection methods are generally divided into two categories: model-based and model-free. The model-based methods usually use physical analysis tools [6], [7], [14] to generate grasps on models and then register observed points with the objects. Some works improve performance by improving the pose estimation accuracy [21], however, it is difficult to generalize these methods to generic objects without 3D models. In contrast, model-free methods based on deep learning can generalize to novel objects.

Some model-free works detect grasps by assessing grasp robustness of sampled grasps. Dex-net [11] uses GQ-CNN to learn a robustness metric based on depth images. GPD [16] and PointNetGPD [17] both perform a classification to identify graspable regions based on point clouds. Mousavian et al. [22] use a variational autoencoder to sample grasps and an evaluator to assess these grasp candidates.

And some methods are inspired by object detection based on RGB or RGB-D images. Lenz et al. [9] propose a sliding window approach, which utilizes a classifier to predict if a patch contains a potential grasp. However, repetitive scanning patches or candidates and classifying take lots of time. Redmon et al. [8] propose a one-shot detection method using an end-to-end network instead of sliding windows based on RGD images. Following it, several works [10], [23]–[27] make a series of improvements and achieve better performance. Nevertheless, these methods struggle to find the optimal grasp because of the lacking consideration of surface geometric information and grasp quality metrics.

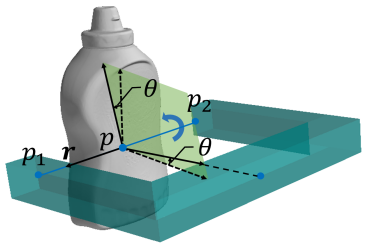

Fig. 2. The grasp configuration in the world coordinate system. When the gripper is placed as a grasp, the grasp orientation $\mathbf{r}$ is the connection of two fingertips, and grasp angle θ is the angle of the gripper rotating around the orientation axis starting parallel to the ground.

The recent work, S^4G [18] directly regresses grasps from single-point features extracted by PointNet++. Though the features contain group information, they acquire less shape awareness by lack of modeling the local spatial layout [31]. When grasping is performed on an area, the less shape awareness leads to less accurate grasp detection. We present a detection network based on region features, which have a large receptive field to effectively combine local features.

B. Grasp Dataset Generation

Most deep learning methods require datasets for the training process. The manually labeled datasets, such as Cornell [28] and VMRD [23] grasp dataset, include no specific information about the grasp quality metrics. In contrast, some methods automatize grasp generation based on physic simulation or random trials [29], [30]. Some works also generate datasets by analyzing the geometry information between grippers and objects. Dex-net [11] generates binary grasp quality scores using robust quasi-static GWS analysis. And PointNetGPD [17] provides continuous grasp quality scores generated by force-closure and GWS analysis. However, they are difficult to evaluate the grasp probability of a point as the grasp center, which is called the grasp confidence. The grasp confidence is important to learn which area is suitable for grasping. Therefore, we build a grasp dataset with the grasp quality metrics and the point grasp confidence.

III. Problem Statement

Given an observed single-view point cloud P, we aim at learing parallel-jaw grasp configurations g in 3D space. On the basis of GPD [19], we define the grasp as $g = (p, \boldsymbol{r}, \theta) \in \mathbb{R}^7$, where $p = (x, y, z) \in \mathbb{R}^3, \boldsymbol{r} = (r_x, r_y, r_z) \in \mathbb{R}^3$ and $\theta \in [-\pi/2, \pi/2]$ represent the grasp center, orientation and the approach angle, respectively. Fig. 2 illustrates the meaning of each parameter in grasp configuration. For each grasp configuration g, the corresponding *grasp quality metric* is defined as s_g, which means the grasp probability of g. Furthermore, since s_g only describes the grasp probability

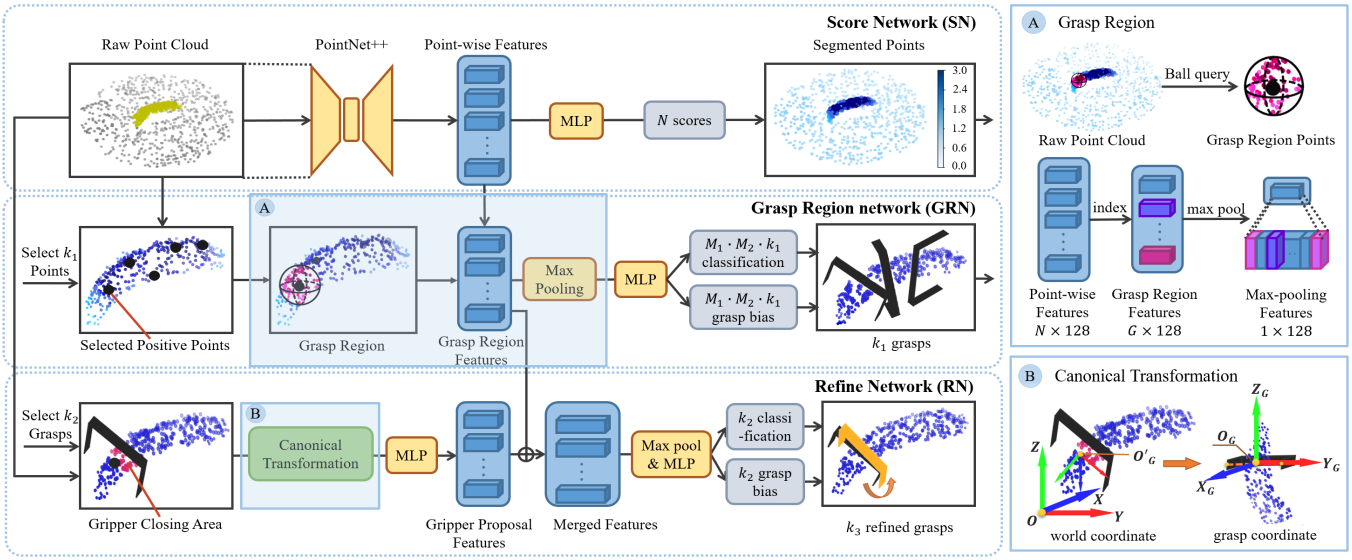

Fig. 3. Architecture of the REGNet. SN takes raw point clouds as input and outputs segmented points with grasp confidence. The picture of segmented points shows that the darker the blue, the higher the confidence. GRN takes point clouds and segmented points from SN as input and outputs coarse predicted grasp proposals. RN takes point clouds and predicted grasps from GRN as input and outputs the refined grasps. (A) shows the meanings of the grasp region and the grasp region features. (B) demonstrates how to transform points from the world coordinate system to the grasp coordinate system.

of one grasp, we define the *point grasp confidence* c_{pc} to assess the grasp probability of each point as the grasp center and to learn which area in P is suitable for grasping.

IV. Proposed Approach

We present the **RE**gion-based **G**rasp **N**etwork to detect grasps in 3D space from the single-view point cloud. The overall architecture includes three parts, Score Network (SN), Grasp Region Network (GRN) and Refine Network (RN), which is illustrated in Fig. 3. Specially, we utilize PointNet++ [4] to extract features that are shared with the three stages.

A. Score Network for Grasp Confidence Evaluation

The *Score Network* (SN) takes point clouds as input to estimate point grasp confidence, which is the grasp probability of each point as a grasp center. To accurately extract point-wise features of the raw point cloud, we utilize PointNet++ [4] as our backbone network. Given a certain number of points, PointNet++ encodes these points into group features and then decodes the group features into point-wise features through distance interpolation. It makes the features contain contextual informations and benefits grasp generation. Then we design a binary segmentation head to evaluate the grasp confidence of each point from the extracted features. After the segmentation, all points will be labeled as positive and negative. We define the SN loss L_1 using the cross-entropy loss, which is formulated as (1).

$$L_1 = -\frac{1}{N}\sum_{pc \in P} c^i_{pc} log \widehat{c}^i_{pc}\ , i \in \{neg, pos\} \qquad (1)$$

where N is the number of points in the raw point cloud P, $c^i_{pc} \in \{0, 1\}$ is the ground-truth confidence of the point pc, which is generated from c_{pc} using the method described in Section V, and $\widehat{c}^i_{pc}$ is the predicted score of the i^{th} category (negative or positive). If $\widehat{c}^{pos}_{pc} > \widehat{c}^{neg}_{pc}$, pc is proper as a regression point of the grasp center. In the following, $P_{pos} = \{pc | \widehat{c}^{pos}_{pc} > \widehat{c}^{neg}_{pc}, pc \in P\}$ is called *positive point set*.

B. Grasp Region Network for Grasp Proposal Generation

Since the positive points are highly informative for predicting their associated grasps, the *Grasp Region Network* (GRN) uses these points segmented by the SN as regression points to effectively regress grasp proposals.

Since an object has various grasps, it's not necessary to use all points in P_{pos} as regression points. In P_{pos}, we only keep a subset containing k_1 points using the farthest point sampling method (FPS) [4]. FPS ensures that our network can cover as many points as possible with different location structures. Then we get the grasp regions of k_1 points, which are spheres centered on these points. Considering that grasp orientation is mostly closely relevant to the grasp center and is only feasible in a small range, we introduce the grasp anchor mechanism that predefines grasp anchors with assigned orientations. GRN obtains the features of the k_1 grasp regions and regresses one proposal on each of them, totally k_1 proposals, based on the grasp anchor mechanism.

Grasp region. In the research of 2D object detection, "Region" is often considered to be a rectangular region [3]. Nevertheless, in this paper, *Grasp Region* is a sphere centered on a positive point $p_a \in P_{pos}$, which is shown in Fig. 3(A). Noticeably, P_{pos} keeps k_1 selected points in it.

Given N points as input, ball query [4] finds all points that are within a radius ϕ to the positive point p_a, which guarantees that the obtained points have a fixed region scale. To ensure the fixed-dimensional input, we randomly sample and keep G points in the grasp region. Based on PointNet++, we obtain the features of these points, which are called *grasp region features* in the following. As illustrated in Fig. 3(A),

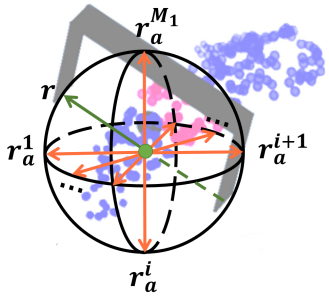

Fig. 4. Illustration of sampling orientations in anchors. $\boldsymbol{r_a^i}$ is a vector, which can be sampled from p_a to any point on the spherical surface. Given a ground-truth orientation $\boldsymbol{r}$ (green line), the category is determined by the minimum angle between it and the assigned orientation $\boldsymbol{r_a^i}$ (orange line).

taking k_1 grasp region features as input, the symmetric max-pooling operation outputs new features that are invariant to the input order [5]. Then through applying multi-layer perception (MLP) on the max-pooling features, we regress k_1 grasp proposals using the grasp anchor mechanism.

Grasp anchor mechanism. Instead of direct regression, the anchor-based classification and regression can achieve high localization accuracy [34]. Since a grasp configuration is represented as $g = (p, \boldsymbol{r}, \theta) \in \mathbb{R}^7$, which can be predicted for each positive point p_a, we define the grasp anchor as $g_a = (p_a, \boldsymbol{r_a^i}, \theta_a^j) \in \mathbb{R}^7, (1 \leqslant i \leqslant M_1, 1 \leqslant j \leqslant M_2)$.

Considering that small changes in θ have little effect for grasp detection, which means θ has a large fault tolerance rate, it is permitted that there is an acceptable regression error in θ. The definition of the anchor can be simplified as $g_a = (p_a, \boldsymbol{r_a^i}, 0) \in \mathbb{R}^7, (1 \leqslant i \leqslant M_1)$, which predefines grasp anchors with assigned orientations as the reference. Each anchor is centered on p_a, and is associated with a 3-dimensional orientation $\boldsymbol{r_a^i}$ and a zero angle. In other words, for a positive point p_a, there are M_1 corresponding anchors. As shown in Fig. 4, the anchor orientation $\boldsymbol{r_a^i}$ is a vector from the sphere center to the surface, which can be sampled from a unit sphere centered on p_a. For uniform sampling, the angle between each $\boldsymbol{r_a^i}$ should be equal.

The loss function for orientation estimation consists of two terms, one term for classification of the assigned orientation $\boldsymbol{r_a^i}$, and the other for residual regression within $\boldsymbol{r_a^i}$. For center estimation, we utilize smooth L1 loss for regression since the distance between p_a and its corresponding ground-truth center is within a small range. The optimized targets of grasp orientation and center are defined as:

$$
\begin{aligned}
c^{(p_a)} &= \arg\min_i \left\langle \boldsymbol{r_a^i}, \boldsymbol{r^{(p_a)}} \right\rangle, \ 1 \leqslant i \leqslant M_1 \\
res_r^{(p_a)} &= \frac{\boldsymbol{r^{(p_a)}}}{\|\boldsymbol{r^{(p_a)}}\|} - \frac{\boldsymbol{r_a^{c^{(p_a)}}}}{\|\boldsymbol{r_a^{c^{(p_a)}}}\|} \\
res_p^{(p_a)} &= \left(p^{(p_a)} - p_a\right)/S
\end{aligned}
\tag{2}
$$

where $c^{(p_a)}$ is the ground-truth orientation's category, $res_p^{(p_a)}$ and $res_r^{(p_a)}$ are the ground-truth residuals of center and orientation within the assigned category, $p^{(p_a)}$ and $\boldsymbol{r^{(p_a)}}$ are the ground-truth center and orientation, $\boldsymbol{r_a^i}$ and $\boldsymbol{r_a^{c^{(p_a)}}}$ are the i^{th} and $c^{(p_a)}$ assigned orientation, and S is the maximum of length, width and height of the gripper. During caculation, the unitization method guarantees that $\boldsymbol{r^{(p_a)}}$ and $\boldsymbol{r_a^{c^{(p_a)}}}$ can be unit vectors.

Since the assigned angle is *0*, we directly use smooth L1 loss to estimate θ. The overall GRN loss L_2 is defined as:

$$
\begin{aligned}
L_2 &= \frac{1}{N_{pos}} \left(\lambda_{cls} \cdot L_{cls} + \sum_{u \in \{p,r,\theta\}} \lambda_u \cdot L_u \right) \\
L_{cls} &= \sum_{p_a \in P_{pos}} F_{cls}\left(\hat{c}^{(p_a)}, c^{(p_a)}\right) \\
L_u &= \sum_{p_a \in P_{pos}} F_{reg}\left(\hat{res}_u^{(p_a)}, res_u^{(p_a)}\right), \ u \in \{p, r, \theta\}
\end{aligned}
\tag{3}
$$

where L_{cls}, L_u are the losses of orientation's classification and residual regressions, N_{pos} is the number of points in P_{pos}, which is equal to k_1, $\hat{c}^{(p_a)}$ and $c^{(p_a)}$ are the predited and ground-truth category of orientation, $\hat{res}_u^{(p_a)}$ and $res_u^{(p_a)}$ are the predited and ground-truth residuals of center, orientation and angle. Specially, $res_\theta^{(p_a)}$ is equal to $\theta^{(p_a)}$, since the assigned angle $\theta_{aj} = 0$. F_{cls} and F_{reg} denote the cross-entropy classification loss and smooth L1 loss. Considering the different magnitudes of these losses, we set $\lambda_{cls} = 0.2$, $\lambda_p = 10$, $\lambda_r = 5$ and $\lambda_\theta = 1$ in practice.

For each p_a, we predict the grasp only using a unique positive anchor and the corresponding residual term. Finally, we will obtain k_1 predicted grasp proposals.

C. Refine Network for Grasp Refinement

To generate more accurate grasps, we propose the *Refine Network* (RN) to refine the proposals. Compared to the grasp region, the area within a grasp predicted by GRN, which is defined as *gripper closing area*, contains information closer to the ground truth. A gripper closing area containing fewer points has less information about the ground truth. Hence, we only select grasps containing more than 50 points in their gripper closing areas for refinement. Specifically, we use k_2 to denote the number of selected grasps. Firstly, we transform the points in the selected gripper closing areas from the world coordinate to the grasp coordinate systems to fully utilize proposals generated from GRN. RN then combines the features of grasp regions and gripper closing areas extracted by MLP to obtain better fusion features. Finally, RN refines k_2 previous predicted grasps using the fusion features.

Canonical transformation. For one grasp $\hat{g} = (\hat{p}, \hat{\boldsymbol{r}}, \hat{\theta})$ predicted from GRN, we transform the points in the gripper closing area from the world coordinate to the grasp coordinate system. As illustrated in Fig. 3(B), the *grasp coordinate system* defines that: (1) the origin O_G is located at the predicted center $\hat{p}$; (2) Y_G axis is along the direction of the orientation $\hat{\boldsymbol{r}}$; (3) X_G axis is obtained by rotating X' around Y_G by θ, (X' axis is parallel to the ground in the world coordinate system and perpendicular to Y_G); (4) Z_G axis is perpendicular to both X_G and Y_G axes. Each point pc within

the gripper closing area will be transformed to the grasp coordinate system as $\widetilde{pc}$ through canonical transformation.

Feature fusion for refinement. MLP is able to extract better local features from the transformed points $\widetilde{pc}$ in gripper closing areas, which are called *gripper closing area features*. And for k_2 selected grasps, there are also k_2 grasp region features extracted by PointNet++. RN concatenates the grasp region and gripper closing area features to get the fusion features. The features are used to refine k_2 grasps through max-pooling and MLP layers for classification and regression. The classification y_i is for classifying if the i^{th} predicted grasp from GRN is close to the ground truth. When the differences between the predicted and ground-truth orientation and angle are less than $2\pi/9$ and $\pi/3$, y_i will be set as 1. If else, $y_i = 0$. The residual targets for regression are only for grasps with $y_i = 1$, which are defined as:

$$
\begin{aligned}
res_{Rr}^{(i)} &= \frac{\boldsymbol{r^{(i)}}}{\|\boldsymbol{r^{(i)}}\|} - \frac{\hat{\boldsymbol{r}}_{\boldsymbol{i}}}{\|\hat{\boldsymbol{r}}_{\boldsymbol{i}}\|},\ \hat{\boldsymbol{r}}_{\boldsymbol{i}} = r\hat{e}s_r^{(i)} + \boldsymbol{r}_{\boldsymbol{a}}^{\hat{\boldsymbol{e}}^{(\boldsymbol{i})}} \\
res_{Rp}^{(i)} &= \left(p^{(i)} - \hat{p}_i\right)/S,\ \hat{p}_i = r\hat{e}s_p^{(i)} \cdot S + p_a \\
res_{R\theta}^{(i)} &= \theta^{(i)} - r\hat{e}s_\theta^{(i)}
\end{aligned}
\tag{4}
$$

where $res_{Rr}^{(i)}$, $res_{Rp}^{(i)}$, $res_{R\theta}^{(i)}$ are the i^{th} true residuals of the orientation, center and angle in RN, $\hat{\boldsymbol{r}}_{\boldsymbol{i}}, \hat{p}_i, r\hat{e}s_\theta^{(i)}$ are values predicted from GRN, and $\boldsymbol{r^{(i)}}, p^{(i)}, \theta^{(i)}$ are the ground truth.

Since predicted values are close to the ground truth, RN directly regresses residuals and the RN loss L_3 is defined as:

$$
\begin{aligned}
L_3 &= \frac{1}{k_2} \cdot \lambda'_{cls} \cdot L'_{cls} + \frac{1}{k'_3} \cdot \sum_{u \in \{p,r,\theta\}} \lambda'_u \cdot L'_u \\
L'_{cls} &= \sum_{i=0}^{k_2} F_{cls}\left(\hat{y}_i, y_i\right)\ ,y_i \in \{0,1\} \\
L'_u &= \sum_{\substack{i=0 \\ y_i=1}}^{k_2} F_{reg}\left(r\hat{e}s_{Ru}^{(i)}, res_{Ru}^{(i)}\right),\ u \in \{p,r,\theta\}
\end{aligned}
\tag{5}
$$

where $L'_{cls},\ L'_u$ are losses of classification and regressions, F_{cls}, F_{reg} denote the cross-entropy loss and smooth L1 loss, y_i, $res_{Ru}^{(i)}$ are the ground-truth category and residuals of the i^{th} grasp proposal, $\hat{y}_i$, $r\hat{e}s_{Ru}^{(i)}$ are the predicted category and residuals, and k_2, k'_3 are the number of grasps from GRN and grasps with $y_i = 1$ (regression losses are only computed for grasps with $y_i = 1$). In practice, we set λ'_{cls}, λ'_p, λ'_r and λ'_θ all to 1. RN finally obtains k_3 refined grasps and k_3 is equal to the number of predicted positives.

V. Grasp Dataset Generation

To train REGNet, we need to generate a large-scale grasp dataset G_{pos}. The process of generating data includes sampling grasp candidates, scoring grasp quality metrics s_g, and finally calculating the point grasp confidence c_{pc}.

We use the same sampling method as PointNetGPD [17] to generate grasp candidates of objects in the YCB dataset [20]. Then we use force-closure analysis and collision detection to evaluate a binary grasp quality metrics s_g for each grasp. The s_g is formulated as $s_g = min(s_g^a, s_g^c)$, where s_g^a and s_g^c are the corresponding binary antipodal score and binary collision score. G_{pos} only contains grasps with $s_g = 1$. Finally, we propose a method to calculate the point grasp confidence to represent the grasp probability of a point as the grasp center.

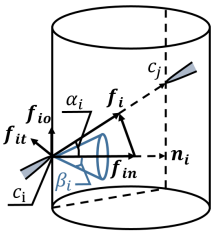

Fig. 5. α_i is the angle between the contact force and the normal vector at p_i. β_i is the angle of the friction cone at p_i.

A. Antipodal and Collision-free Grasps Construction

Force closure is a basic constraint in robotic grasping, which is widely used to construct antipodal grasps [2]. Assume that the contact type between grippers and objects is point contact with friction, the force exerted to the object $\boldsymbol{f_i}$ can be decomposed into a normal force $\boldsymbol{f_{in}}$ and two tangential forces $\boldsymbol{f_{io}}, \boldsymbol{f_{it}}$ [1]. Force-closure requires $\boldsymbol{f_i}$ to satisfy the constraint $\sqrt{\boldsymbol{f_{io}^2} + \boldsymbol{f_{it}^2}} \leqslant \mu_i \|\boldsymbol{f_{in}}\|$, where μ_i is the friction coefficient. We simplify the constraint to $\alpha_i \leqslant \beta_i$, where $\alpha_i = \langle \boldsymbol{n_i}, \boldsymbol{f_i} \rangle, \beta_i = arctan\mu_i$, and $\boldsymbol{n_i}$ is the normal vector. We use s_g^a to assess whether a grasp is force closure. As shown in Fig. 5, for each grasp g, there are two contact points c_i, c_j. Since μ_i is unknown, we choose $\mu = 0.6$ as a threshold. We compute $\alpha_i, \alpha_j, \beta_i$ and β_j to judge whether g is force closure when $\mu_i = \mu$. If g is force closure ($\alpha_i,\ \alpha_j \leqslant arctan\mu$), we set $s_g^a = 1$. If not, $s_g^a = 0$.

The collision score s_g^c is generated from the collision detection. The collision detection simulates the grasping process to detect whether the gripper collides with objects. If a grasp g does not collide with the object, we set $s_g^c = 1$, which guarantees the grasp is collision-free. If not, $s_g^c = 0$.

B. Point Confidence Generation

Noticeably, the point grasp confidence c_{pc} describes the grasp probability of each point as a grasp center. Since c_{pc} intuitively indicates the density of grasps with $s_g = 1$ near the point pc, it can help learn which area in P is suitable for grasping. Specifically, we count all randomly generated grasps in G_{pos} to calculate c_{pc}, which is defined as:

$$
\begin{aligned}
c_{pc} &= \sum_{g_i \in G_{pos}} \sigma_{g_i},\ pc \in P \\
\sigma_{g_i} &= \begin{cases} 0 & dis(pc, p_i) \geqslant d_{th} \\ 1 - dis(pc, p_i)/d_{th} & \text{else} \end{cases}
\end{aligned}
\tag{6}
$$

where pc is a point in P, g_i is a grasp in G_{pos}, d_{th} is the distance threshold, and $dis(pc, p_i)$ is the distance between pc and the center of g_i. In practice, d_{th} is set as $0.02m$. Intuitively, c_{pc} of a point will be higher as there are more

grasps near it. We set a threshold $c_t = 0.6$ to divide all points in P into positive and negative according to c_{pc}. The ground truth of SN will be set as $c_{pc}^{neg} = 1, c_{pc}^{pos} = 0$ when $c_{pc} \leqslant c_t$.

VI. Experiments

In this section, we concentrate on the grasp detection performance of our proposed method and the effectiveness of each component. Firstly, we present the *Valid Grasp Ratio* (VGR), which is a more general and direct metric for performance evaluation of grasp detection in 3D space. Then the results evaluated on our dataset demonstrate that REGNet achieves the highest VGR of 92.47% on the seen objects, and is able to be successfully generalized to unseen ones. Moreover, ablation studies show that every component of REGNet has an inextricable effect on the final performance, making REGNet a state-of-the-art grasp detection algorithm based on point clouds.

A. Evaluation Metrics for 3D Grasp Detection

In this part, we propose the Valid Grasp Ratio (VGR), Valid Antipodal Grasp Ratio (VAGR) and Valid Collision-free Grasp Ratio (VCGR) to evaluate the performance of grasp detection methods in 3D space.

IOU or Jaccard, the similarity measure metrics are widely used in 2D detection methods to evaluate performance. Nevertheless, their calculation in 3D space is complicated. Moreover, since a large overlap area between the predicted grasp and the ground truth does not ensure a stable grasping trial, they cannot be used as accurate metrics for 3D grasp detection evaluation. On the other hand, some grasp detection methods in 3D space simply use the classification accuracy as the measurement [11], [17], which is not convenient to transfer to other types of methods.

Therefore, we the present *Valid Grasp Ratio* (VGR), *Valid Antipodal Grasp Ratio* (VAGR), *Valid Collision-free Grasp Ratio* (VCGR) to evaluate the performance. Specifically, we firstly transform k_3 predicted grasps from the world coordinate system to the object coordinate system. Then their quality metrics are obtained through the method described in Section V. We use k_T to denote the number of antipodal and collision-free grasps, whose s_g^a and s_g^c are both equal to 1. The metric of VGR is defined as $VGR = k_T/k_3$, which is expressed as the quotient of the number of grasps with high-quality metrics and all predicted grasps. Similar to VGR, k_T^a and k_T^c denote the number of antipodal grasps whose $s_g^a = 1$ and collision-free grasps whose $s_g^c = 1$. VAGR and VCGR can be defined as $VAGR = k_T^a/k_3$ and $VCGR = k_T^c/k_3$.

There are several advantages of VGR as the performance measurement. Firstly, VGR is convenient to be computed by simply counting the ratio of high-quality metric grasp configurations. Secondly, VGR can intuitively describe the grasp detection performance since it gives out the possibility of the predicted grasp to be a robust one. Finally, VGR is easy to be transferred to other grasp detection methods, which means that it has the potential to be a benchmark for grasp detection in 3D space. Based on VGR, we also propose VAGR and VCGR for more comprehensive evaluation, which can assess the antipodal grasp and collision-free grasp generation ability, respectively.

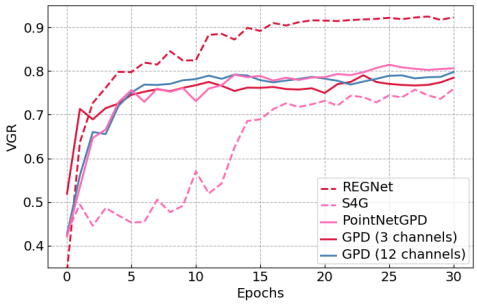

Fig. 6. Test VGR of various methods. (Both GPD and PointNetGPD are trained for 200 epochs. Their best test results are shown in Table I.)

B. Network Details

In this part, we will give an introduction to the details of our network designation.

For SN, we randomly sample 20000 points from the raw point cloud as input, which means $N = 20000$, and the input is 6-dimension, including location (x, y, z) and color (r, g, b) information. For extracting features of the input, we use the same architecture as PointNet++ [4] with MSG (multi-scale grouping). Through three hierarchical set abstraction and three feature propagation layers, we get $N \times 128$ features. Then we use one conv1d layer to obtain segmentation results.

For GRN, $k_1 = 64$ during training. k_1 can be changed during test because it isn't affected by the training process. When getting grasp regions, we set ϕ as the half maximum of the parallel-jaw gripper's length, width and height, and $G = 256$. M_1, the number of assigned orientations is set as 8. After getting k_1 grasp regions, we use 4 conv1d layers to obtain the classification and regression results.

For RN, we use 3 conv1d layers to extract gripper closing area features. And we also use 3 conv1d layers to get classification and regression results from the merged features.

The above three networks are trained simultaneously for 30 epochs with batch size 4 and learning rate 0.001 in the beginning. The strategy for adjusting the learning rate is dividing the learning rate by 2 every 5 epochs. We use Adam as the optimizer.

C. Evaluation of Grasp Detection

Dataset split. We use the method described in section V to generate our grasp dataset. We choose six objects in YCB dataset [20] and generate 400 grasps with high-quality metrics for each object. The training and test datasets are divided by a ratio of $4:1$, which contain 2880 and 720 point clouds from various perspectives, respectively.

Baselines. The compared methods include 3-channel and 12-channel versions of GPD, 2-class single-view PointNetGPD, as well as S^4G. For a fair comparison, in GPD [16] and PointNetGPD [17], we sample $k_1 = 64$ grasps and construct binary classifiers to evaluate quality metrics. k_3 is the number of the predicted positive class that contains grasps with $s_g = 1$ and k_T can be simplified to the number

TABLE I

COMPARISON OF PERFORMANCE

	Grasp quality			Time efficiency	
	VAGR	VCGR	VGR	Forward time	Computing time
GPD (3 channels)	/	/	79.34%	**2.31ms**	2077.12ms
GPD (12 channels)	/	/	80.22%	2.67ms	2502.38ms
PointNetGPD	/	/	81.72%	4.77ms	1965.60ms
S4G	87.78%	78.83%	77.63%	559.81ms	**679.04ms**
REGNet	**98.69%**	**93.13%**	**92.47%**	556.29ms	686.31ms

The forward and computing time are the forward passing and total running time. The forward time of feature extraction in REGNet is 553.04ms. Since the negative class of GPD and PointNetGPD contains $s_g^a = 0$ or $s_g^c = 0$ grasps, whose distribution significantly affects the metrics of VAGR and VCGR, we don't compare them separately.

TABLE II

PERFORMANCE ON SEEN AND UNSEEN OBJECTS

Objects		VAGR	VCGR	VGR
Seen objects	mustard bottle	94.88%	99.46%	94.41%
	gelatin box	99.35%	99.76%	99.30%
	banana	99.96%	99.95%	99.92%
	peach	99.83%	87.35%	87.28%
Unseen objects	sugar box	95.55%	88.62%	87.06%
	pudding box	99.00%	98.36%	97.45%
	golf ball	86.18%	86.23%	85.76%
	plum	94.40%	71.67%	70.16%
	screwdriver	74.76%	89.75%	72.44%

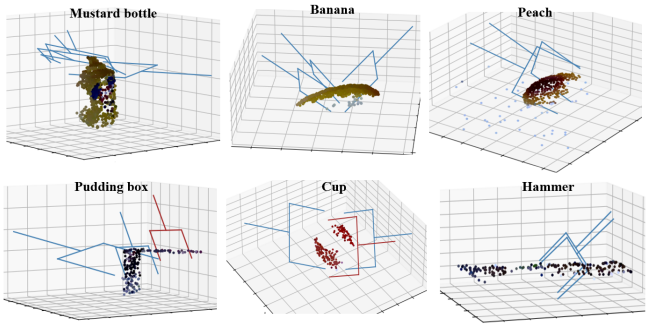

Fig. 7. Some grasp detection results. Blue grasps have high-quality metrics, while red grasps are not collion-free.

of true positives. For S^{4}G [18] and our method, we directly compute the VGR metric.

Results and analysis. In this part, we compare the performance of our method with baselines in the aspects of VGR and time efficiency. The results demonstrate that REGNet significantly outperforms all baselines and the speed is on a par with the fastest method. Moreover, we also show that it can successfully generalize to unseen objects.

In Table I, REGNet increases the VGR by 10.75% compared with PointNetGPD, which indicates its effectiveness for grasp detection in 3D space. As for the computing time, REGNet is comparable to the fastest algorithm, S^{4}G. The forward time of REGNet is mainly spent on feature extraction based on PointNet++. We can choose alternative feature extraction networks [5], [31], [32] as the backbone to reduce the forward time. On the other hand, GRN and RN act in a nearly cost-free way by sharing features with SN, leading to an elegant and effective solution.

TABLE III

RESULTS OF ABLATION ANAYSIS

	VAGR	VCGR	VGR
Ours (direct-single)	96.05%	87.73%	85.57%
Ours (direct-region)	96.73%	89.94%	87.91%
Ours (w/o RN)	98.62%	91.98%	91.36%
Ours (w/ RN)	**98.69%**	**93.13%**	**92.47%**

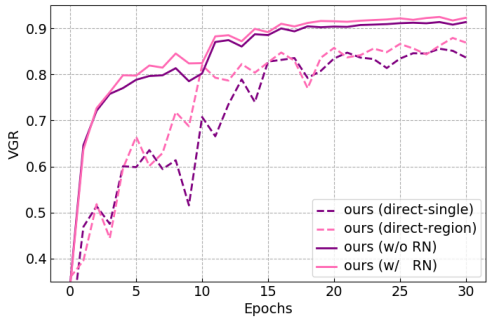

Fig. 8. Test VGR of ablation experiments.

As illustrated in Table II, REGNet can effectively predict grasps on the unseen objects. However, the performance on unseen ones is limited in: (1) the object's length, width, and height are close to the gripper width, in which a small deviation will cause collision, such as the plum; (2) the structures of unseen objects are dissimilar to the seen ones. For example, the existing model cannot accurately predict the grasps of the screwdriver since it includes a slender cylindrical structure that doesn't appear in the training dataset.

D. Ablation Analysis and Discussion

We conduct a series of extensive ablation experiments to analyze the effectiveness of main components of REGNet including the grasp region, grasp anchor mechanism and refine network. The results demonstrate that all 3 parts contribute inextricably to the performance.

Evaluation of the Grasp Region. In this part, we analyze the effort of the grasp region on the final performance. In detail, for a fair comparison, we demonstrate the performance of direct-single and direct-region versions of REGNet, which directly regress grasps from single-point and grasp region features, respectively. The two versions both include the SN stage and the only difference between them is the feature

used for regression in GRN. In Table III, the VGR drops without the grasp region, which illustrates that the grasp region improves the efficiency of grasp detection.

Evaluation of grasp anchor mechanism. In order to analyze the effort of the grasp anchor mechanism, we compare the performance of direct-region and regression based on anchor mechanism (w/o RN) versions, in which the grasps are generated from GRN. The only difference between them is whether to use the grasp anchor mechanism, i. e., ours (direct-region) directly regresses grasps while ours (w/o RN) regresses grasps based on the grasp anchor mechanism. The grasp anchor mechanism increases the VGR by 3.45% and contributes inextricably to the performance.

Evaluation of Refine Network. To analyze the effectiveness of RN, we compare the performance of versions without (w/o RN) and with the RN stage (w/ RN). Removing the RN decreases the VGR by 1.11%, which demonstrates the advantages of grasp refinement based on fusion features.

VII. CONCLUSIONS

We present the REGNet, an end-to-end single-shot network based on the single-view point cloud to detect grasps in 3D space. Our method significantly improves the detection performance and can be generalized to detect grasps on unseen objects. The ablation studies demonstrate each component in REGNet is effective and contributes inextricably to the final performance.

In future work, we will generate a synthetic scene of many objects to extend our dataset. Based on the synthetic data, our work will effectively detect grasps in dense clutter.

REFERENCES

[1] Yu Zheng and Wen-Han Qian. Coping with the grasping uncertainties in force-closure analysis. In *International Journal of Robotics Research (IJRR)*, 24(4):311-327, 2005.

[2] Van-Duc Nguyen. Constructing force-closure grasps. In *International Journal of Robotics Research (IJRR)*, 7(3):3-16, 1988.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91-99, 2015.

[4] Charles R. Qi, Li Yi, Hao Su, Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[5] Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Borst C, Fischer M, Hirzinger G. Grasp Planning: How to Choose a Suitable Task Wrench Space. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[7] Andrew Miller and Peter K. Allen. Graspit!: A Versatile Simulator for Robotic Grasping. In *IEEE Robotics and Automation Magazine*, 11(4):110-122, 2004.

[8] Redmon Joseph, Angelova Anelia. Real-Time Grasp Detection Using Convolutional Neural Networks. In *IEEE International Conference on Robotics and Automation (ICRA)*,2015.

[9] Ian Lenz, Honglak Lee, Ashutosh Saxena. Deep Learning for Detecting Robotic Grasps. In *Robotics: Science and Systems Conference (RSS)*, 2013.

[10] Xinwen Zhou, Xuguang Lan, et al. Fully Convolutional Grasp Detection Network with Oriented Anchor Box. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7223-7230. IEEE, 2018.

[11] Mahler, Jeffrey, Liang, Jacky, Niyaz, Sherdil, et al. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. In *Robotics: Science and Systems Conference (RSS)*, 2017.

[12] Domenico Prattichizzo, and Jeffrey C. Trinkle. Grasping. In *Springer handbook of robotics*, pages 671-700. Springer, 2008.

[13] Jonathan Weisz and Peter K. Allen. Pose error robust grasping from contact wrench space metrics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 557-562. IEEE, 2004.

[14] Andrew Miller, Peter Kirby Allen. Examples of 3D Grasp Quality Computations. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[15] David Fischinger, Markus Vincze, Yun Jiang. Learning grasps for unknown objects in cluttered scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[16] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, Robert Platt. Grasp Pose Detection in Point Clouds[J]. *International Journal of Robotics Research (IJRR)*, 2017.

[17] Hongzhuo Liang, Xiaojian Ma, et al. PointNetGPD: Detecting Grasp Configurations from Point Sets. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[18] Yuzhe Qin, Rui Chen, et al. S4G: A modal Single-view Single-Shot SE(3) Grasp Detection in Cluttered Scenes. In *Conference on Robot Learning (CoRL)*, 2019.

[19] Marcus Gualtieri, Andreas ten Pas, Kate Saenko, Robert Platt. High precision grasp pose detection in dense clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[20] Berk Calli, Arjun Singh, James Bruce, and Aaron Walsman. Yale-cmu-berkeley dataset for robotic manipulation research. In *International Journal of Robotics Research (IJRR)*, 36(4):261-268, 2017.

[21] Zeng Andy, Yu Kuan-Ting, et al. Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[22] Arsalan Mousavian, Clemens Eppner, Dieter Fox. 6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[23] Hanbo Zhang, Xuguang Lan, Site Bai, et al. ROI-based Robotic Grasp Detection for Object Overlapping Scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019

[24] Hanbo Zhang, Xinwen Zhou, Xuguang Lan, et al. A Real-time Robotic Grasping Approach with Oriented Anchor Box. In *IEEE Trans. Syst. Man Cybern*, 2019.

[25] Umar Asif, Jianbin Tang, Stefan Harrer. GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices. In *International Joint Conferences on Artificial Intelligence Organization (IJCAI)*, 2018.

[26] Di Guo, Fuchun Sun, Huaping Liu, et al. A hybrid deep architecture for robotic grasp detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1609-1614. IEEE, 2017.

[27] Sulabh Kumra and Christopher Kanan. Robotic Grasp Detection using Deep Convolutional Neural Networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[28] "Cornell grasping dataset," http://pr.cs.cornell.edu/grasping/rect_data/data.php, accessed: 2013-09-01

[29] Amaury Depierre, Emmanuel Dellandrea, and Liming Chen. Jacquard: A Large Scale Dataset for Robotic Grasp Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[30] Lerrel Pinto and Abhinav Gupta. Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[31] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-Shape Convolutional Neural Network for Point Cloud Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[32] Jiaxin Li, Ben M Chen, Gim Hee Lee. SO-Net: Self-Organizing Network for Point Cloud Analysis. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, 2018.

[33] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-Driven Grasp Synthesis - A Survey. In *IEEE Transactions on Robotics (TRO)*, 2013.

[34] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object Detection in 20 Years: A Survey. arXiv preprint arXiv:1905.05055, 2019.