**Graspness Discovery in Clutters for Fast and Accurate Grasp Detection**

Chenxi Wang*, Hao-Shu Fang*, Minghao Gou, Hongjie Fang, Jin Gao, Cewu Lu†
Shanghai Jiao Tong University
wcx1997@sjtu.edu.cn, fhaoshu@gmail.com, {gmh2015, galaxies}@sjtu.edu.cn,
gao3944677@126.com, lucewu@sjtu.edu.cn

1. Introduction

As a fundamental problem in robotics, robust grasp pose detection for unstructured environment has been fascinating our community for decades. It has a broad spectrum of applications in picking [10], assembling [40], home serving [11], etc. Advancing the generality, accuracy and efficiency is a long pursuit of researchers in this field.

For grasp pose detection in the wild, it can be regarded as a two-stage problem: given a single-view point cloud, we first find locations with high graspability (where stage) and then decide grasp parameters like in-plane rotation, approaching depth, grasp score and gripper width (how stage) for a local region.

Previous methods for 6-DoF grasp pose detection in cluttered scenes mainly focused on improving the quality of grasp parameter prediction, i.e., the how stage, and two lines of research are explored. The first line [41, 27, 31] adopts a sampling-evaluation method, where grasp candidates are uniformly randomly sampled from the scene and evaluated by their model. The second line [36, 13, 32] proposes end-to-end networks to calculate grasp parameters for the whole scene, where point clouds are sampled before [32] or during [36, 13] the forward propagation. For all these methods, the where stage is not explicitly modeled (i.e., they do not perform a filtering procedure in a first stage) and candidate grasp points distribute uniformly in the scene.

However, we find that such uniform sampling strategy greatly hinders the performance of the whole pipeline. There are tremendous points in 3D contiguous space, while positive samples are concentrated in small local regions. Take GraspNet-1Billion [13], the current largest dataset in grasp pose detection as an example. We statistically find that, even with object masks, the graspable points are less...
than 10% among all the samples, not to mention the candidate points in the whole scene. Such an imbalance causes a large waste of computing resources and degrades the efficiency.

To tackle the above bottleneck in grasp pose detection, we propose a novel geometrically based quality, **graspness**, for distinguishing graspable area in cluttered scenes. One might think that we need complex geometric reasoning to obtain such graspness. However, we discover that a simple look-ahead search by exhaustively evaluating possible future grasp poses from a point can well represent its graspness. Statistical results demonstrate the justifiability of our proposed graspness measure, where the local geometry around points with high graspness are distinguished from those with low scores. Fig. 1 gives an illustration of our graspness for a cluttered scene.

Furthermore, we develop a graspness model that approximates the above process in practice. Given a point cloud input, it predicts point-wise graspness score, which is referred to as **graspable landscape**. Benefiting from the stability of the local geometry structures, our graspness model is object agnostic and robust to variation of viewpoint, scene, sensor, etc., making it a general and transferable module for grasp point sampling. We qualitatively evaluate its robustness and transferability in our analysis. Tremendous improvements in both speed and accuracy for previous sampling-evaluation based methods are witnessed after equipping them with our graspness model.

Based on our graspness model, we also propose Graspness-based Sampling Network (GSNet), an end-to-end two-stage network with a graspness-based sampling strategy. Our network takes a dense scene point cloud as input, which preserves the local geometry cues. The sampling layer firstly selects the points with high graspness. Remaining points are discarded from the forward propagation to improve the computation efficiency. Such two-stage design is beneficial to network convergence and also the final accuracy by providing more positive samples during training.

We conduct extensive experiments to evaluate the effectiveness of our proposed graspness measure, model and the end-to-end network. Several baseline methods equipped with our graspness model outperform their vanilla counterparts by a large margin in both speed and accuracy. Moreover, our GSNet outperforms previous methods to a large extent. Our code and models will be made publicly available to facilitate researches in related area.

### 2. Related Work

In this section, we first briefly review previous methods on grasping in cluttered scenes, followed by concluding the common strategies they have used to sample grasp candidates. Finally, we surveyed some literature in cognitive science area where grasp recognition is witnessed in human perception.

#### Grasping in Cluttered Scenes

For cluttered scene grasp pose detection, previous research can be mainly divided into two categories: planar based grasp detection and 6-DoF based grasp detection. The research in the first category [1, 22, 25, 29, 30, 37, 24, 8] mainly took RGB images or depth images as inputs and output a set of rotated bounding boxes to represent the grasp poses. Due to the limitation of low DoF, their applications were usually restricted. Another line of research aimed to predict full DoF grasp poses. Among them, two different directions were explored. The first direction [42, 41, 27, 31] adopted the sampling-evaluation based two-step policy, where grasp candidates were densely uniformly sampled in the scene and evaluated using a deep quality model. The second direction [13, 36, 32, 4] adopted the end-to-end strategy, where point clouds of the scene were directly processed by end-to-end networks. For each input point, the network attempted to predict the most feasible grasp pose. All the mentioned methods focused on improving the quality of grasp parameters and the problem of where to grasp was not investigated.

#### Grasp Sampling Strategies

Several kinds of sampling strategies can be concluded from the above methods. The most common used strategy is the uniform sampling, which is adopted by [41, 27, 32, 36]. Specifically, GPD [41] and PointNetGPD [27] uniformly sampled grasp points in the scene point cloud and estimated the rotation by darboux frame. Some end-to-end models [36, 32] down-sampled the input point cloud by voxel grid to avoid memory explosion. A similar strategy, farthest point sampling, is adopted by other end-to-end model [13]. Some optimization based methods are also explored. Ciocarlie et al. [9] and Hang et al. [19] adopted the simulated annealing method, while Mahler et al. [29] proposed cross-entropy methods. In [31], a grasp sampler network first sampled possible grasp poses on partial object point cloud and conducted iterative refinement by a grasp evaluator based on its gradient. In a recent paper by Clemens et al. [12], several sampling methods for grasp dataset generation are reviewed. However, all the previous methods ignore the geometric cues for graspable point sampling. In this paper, we propose a novel graspness measure based on local geometry for graspable point sampling, which is much more efficient than previous uniform sampling and optimization based methods.

#### Graspness in Cognitive Science

In cognitive area, researchers have studied the visual attention during grasping for a long period. Many literature [2, 3, 15, 20, 38] demonstrated that human bias the allocation of available perceptual resources, named as affordance attention, towards the region with the highest graspbility. And such attention
usually precedes the action preparation stage [2]. Such discovery corresponds to our graspsness concept and motivates us to apply it in the grasp sampling strategy.

3. Graspness Discovery

3.1. Preliminary

As mentioned above, we decouple the grasp pose detection problem into two stages. Before the common practice in previous research that directly calculates the grasp parameters, we first sample points and views with high graspsness. Computational resources will be allocated to these areas thereafter to improve computational efficiency.

To determine the suitable grasp locations and the feasible approach directions with high graspspability, we define two kinds of graspsness in a high dimensional space to represent parallel attention in point locations and approach directions. Before detailing our graspsness measure, we first introduce some basic notations.

For a point sets \( P = \{p_i | i = 1, ..., N\} \), we assume \( V \) approach directions uniformly distributed in a sphere space \( V = \{v_j | j = 1, ..., V\} \).

Two kinds of graspsness scores are discussed in this paper. The first is the point-wise graspsness scores denoted as

\[
S_p = \{s_p^i | s_p^i \in [0, 1], i = 1, ..., N\},
\]

where \([0, 1]\) denotes that our graspsness for each point ranges from 0 to 1. The second is the view-wise graspsness scores denoted as

\[
S_v = \{s_v^i | s_v^i \in [0, 1]^V, i = 1, ..., N\},
\]

where \([0, 1]^V\) denotes \(V\)-dim graspsness ranging in \([0, 1]\).

In the following section, we illustrate how we measure graspsness for both single object and the cluttered scene.

3.2. Graspness Measure

Single Object Graspness Given an object point cloud, we aim to generate graspsness for each point where higher activation denotes larger possibility for successful grasping. Assuming there is an oracle \(1(\cdot)\) that tells whether an arbitrary grasp is successful, and \(G_{i,j}\) denotes the set of all feasible grasp poses for view \(v_j\) centered at point \(p_i\), then the graspsness score \(\tilde{s}_p^i\) and \(\tilde{s}_v^i\) can be obtained by an exhaustive look-ahead search:

\[
\tilde{s}_p^i = \sum_{j=1}^{V} \sum_{g \in G_{i,j}} 1(g) \frac{1}{|G_{i,j}|}, i = 1, ..., N,
\]

\[
\tilde{s}_v^i = \left\{ \sum_{g \in G_{i,j}} \frac{1(g)}{|G_{i,j}|} \middle| 1 \leq j \leq V \right\}, i = 1, ..., N.
\]

By doing so, we guarantee that higher graspsness value always denote higher possibility of successful grasping.

In practice, such an oracle \(1(\cdot)\) does not exist, and \(G_{i,j}\) can contain infinite grasp poses in a continuous space.

Figure 2. Graspness scores. The left image shows the graspness without collision detection while the right image shows the graspsness with collision detection

Thus, we make an approximation to the above process. For view \(v_j\) of point \(p_i\), we generate \(L\) grasp candidates \(G_{i,j} = \{g_{k,j}^{i,j} | k = 1, ..., L\}\) by grid sampling along gripper depths and in-plane rotation angles. For each grasp \(g_{k,j}^{i,j}\), we calculate a grasp quality score \(q_{k,j}^{i,j}\) using a force analytic model [29]. A threshold \(c\) is manually set to filter out unsuccessful grasps. Then, the relaxation form of Eqn. 1 is:

\[
\tilde{s}_p^i = \sum_{j=1}^{V} \sum_{k=1}^{L} \frac{1(q_{k,j}^{i,j} > c)}{|G_{i,j}|}, i = 1, ..., N,
\]

\[
\tilde{s}_v^i = \left\{ \sum_{k=1}^{L} \frac{1(q_{k,j}^{i,j} > c)}{|G_{i,j}|} \middle| 1 \leq j \leq V \right\}, i = 1, ..., N.
\]

Scene-Level Graspness After defining the object-level graspsness, we extend it to cluttered scenes by first discussing the gap between them and then redefining the graspsness in cluttered scenes.

A cluttered scene contains multiple objects and the irrelevant background. As shown in Fig. 2(a), the simplest way to compute scene-level graspsness is directly projecting the object-level graspsness score to the scene by object 6D poses. However, this solution ignores the differences between an object model and a scene cloud captured from RGB-D camera. Firstly, a valid grasp of a single object may collide with background or other objects when placing in cluttered manner and becomes a negative grasp. Secondly, as the depth camera provides single-view partial point clouds, we need to associate the scene point cloud with the projected object point.

To deal with the collision problem, we follow [13] to reconstruct the scene using object 3D models and corresponding 6D poses. Each grasp \(g_{k,j}^{i,j}\) is evaluated by a collision checking process and assigned a collision label \(c_{k,j}^{i,j}\). Our graspsness scores are then updated as:

\[
\tilde{s}_p^i = \sum_{j=1}^{V} \sum_{k=1}^{L} \frac{1(q_{k,j}^{i,j} > c) \cdot 1(c_{k,j}^{i,j})}{|G_{i,j}|}, i = 1, ..., N,
\]

\[
\tilde{s}_v^i = \left\{ \sum_{k=1}^{L} \frac{1(q_{k,j}^{i,j} > c) \cdot 1(c_{k,j}^{i,j})}{|G_{i,j}|} \middle| 1 \leq j \leq V \right\}, i = 1, ..., N.
\]

After that, we project the object points to the scene by object 6D poses. For each point in the scene, we obtain its
graspness scores by nearest neighbor search and associate it with the nearest projected object point.

Finally, to obtain a coherent representation for the scene-level graspness scores, we perform a normalization for each scene:

\[ S^p = \left\{ \frac{s_i^p - \min(S^p)}{\max(S^p) - \min(S^p)} \mid i = 1, ..., N \right\}, \]

\[ S^v = \left\{ \frac{s_i^v - \min(S^v)}{\max(S^v) - \min(S^v)} \mid i = 1, ..., N \right\}, \]

where \( \min(\cdot) \) denotes column wise minimum:

\[ \min(S^v) = \left\{ \min_{i=1}^{N} s_i^v \mid j = 1, ..., V \right\}, \]

and so does \( \max(\cdot) \). Fig. 2(b) shows an example of scene-level graspness scores.

3.3. Justification

In order to justify our grasp measure, we analyze the local geometry for regions with different graspness to find out whether they are really distinguishable geometrically. For a single-view point cloud, the cascaded graspness model detailed in Sec. 4.1 is used to extract the local feature vector of each point. The points with graspness more than 0.3 are treated as positive samples, and negative ones of the same size are sampled with graspness less than 0.1. Fig. 3 shows the t-SNE [28] visualization of the encoded local geometry (feature vectors of each point produced by backbone network) for all the scenes in GraspNet-1Billion [13] training/testing set respectively. We can observe that regions with different graspness are quite distinguishable. It demonstrates that our graspness measure is rational and reveals the potential of learning graspness from point cloud.

4. GSNet Architecture

After defining the grasp measure, we introduce the end-to-end grasp pose detection network, GSNet, where our graspness is learned by an independent module and can be applied to other methods.

4.1. Cascaded Graspness Model

Given a dense single view point cloud \( P \), graspness model needs to learn two approximations: \( f^p : P \rightarrow S^p \) and \( f^v : P \rightarrow S^v \).

It is challenging to find a direct mapping from point coordinates to graspness scores due to the large domain gap between these two spaces. Instead, we decompose the whole process into two sub-functions. Consider a high dimensional feature set \( F \):

\[ F = \{ f_i \mid f_i \subset \mathbb{R}^C, i = 1, ..., N \}, \]

where \( \mathbb{R}^C \) denotes \( C \)-dim feature space. The point set is firstly transformed to the feature set by \( h^t : P \rightarrow F \). Graspable landscapes are then generated by \( h^p : F \rightarrow S^p \) and \( h^v : F \rightarrow S^v \). Hence, we model the grasp scores by

\[ f^p = h^p \circ h^t, \quad f^v = h^v \circ h^t, \]

where \( \circ \) denotes function composition, and the feature set \( F \) is shared by both \( h^p \) and \( h^v \).

Although \( h^p \) and \( h^v \) can be learned simultaneously, the computation overhead is quite expensive since \( S^v \) is in high dimensional space. Meanwhile, it is not necessary to compute the view-wise graspable landscapes for all points since most of the points are not graspable at point level. Hence, we propose cascaded graspness model to learn \( h^t, h^p \) and \( h^v \) step by step, where points are sampled by the output of \( h^p \) before learning \( h^v \) to reduce computation cost.

Backbone Network Approximation of \( h^t \) requires a strong backbone network for extraction of both global and local point features. We adopt ResUNet14 built upon MinkowskiEngine [6] because it can flexibly process point sets of any size with sparse convolution and has shown excellent performance in multiple tasks of 3D deep learning [7, 14, 18, 5]. The network can also be replaced by other point-wise networks, such as PointNet [34, 35], PointCNN [26] and SSCNs [17].

The network adopts a U-shape architecture with residual blocks, which obtains point features using 3D sparse (transposed) convolutions and skip-connections. For a point cloud of size \( N \times 3 \), it extracts a \( C \)-channel feature vector set, and outputs a point set of size \( N \times (3 + C) \) for graspable sampling and grasp generation.

Graspable Farthest Point Sampling The modeling for \( h^p \) is implemented with a multi-layer perceptron (MLP) network to generate point-wise graspable landscape. Specifically, the output contains a prediction for the graspable landscape of size \( N \times 1 \) and a binary objectness classification scores of size \( N \times 2 \), resulting a total output of size \( N \times 3 \). Graspness scores of non-object points are set to 0.
After obtaining the point-wise graspable landscape, we select points with graspness score larger than $\delta_p$ and adopt farthest point sampling (FPS) to maximize distances among sampled points. $M$ seed points are sampled with $(3+C)$-dim features, where 3 denotes the point coordinates and $C$ denotes the features output by the backbone network.

**Graspable Probabilistic View Selection** $h^v$ is also modeled by an MLP. We apply it to the sampled seed points and output $M \times V$ vectors for view-wise graspable landscapes and $M \times C$ residual features for grasp generation. $V$ views are sampled from a unit sphere using Fibonacci lattices [16].

After obtaining the view-wise graspness scores, we select the best view for afterward predictions during inference. For training, we adopt probabilistic view selection (PVS) that normalizes the graspness scores of all views on a seed point to $(0,1)$ and regard them as probability scores, which are the sum of features output by graspable FPS and graspable PVS. The grouped point sets of size $M \times K \times (3+C)$ are called grasp candidates, where $K$ stands for the number of sampled points in each group.

**Grasp Generation from Candidates** We use a shared PointNet [34] for grasp generation. Grasp candidates are processed by an MLP network and a max-pooling layer, and be output as feature vectors of size $M \times C'$. Finally we get grasp configurations by a new MLP network.

The output of GSNet contains scores and widths for different (in-plane rotation)-(approach depth) combinations. We pick the combination with the highest score as the grasp prediction. The output size is $M \times (A \times D \times 2)$, where $A$ denotes the number of in-plane rotation angles, $D$ denotes the number of gripper depths and 2 denotes the score and the width.

**Grasp Score Representation** We use the minimum friction coefficient $\mu$ under which a grasp is antipodal to evaluate the quality of the grasp. Based on this, we define the grasp score as

$$q_i = \begin{cases} \frac{\ln (\mu_{\text{max}}/\mu_i)}{\ln (\mu_{\text{max}}/\mu_{\text{min}})} & g_i \text{ is positive,} \\ 0 & g_i \text{ is negative.} \end{cases} \tag{5}$$

All scores are normalized to $[0, 1]$. Smaller $\mu_i$ indicates higher score $q_i$ and more probability to succeed.
4.2.1 Loss Function
Cascaded graspness model and grasp operation model are trained simultaneously with multi-task losses:
\[ L = L_o + \alpha(L_p + \lambda L_v) + \beta(L_s + L_w), \]
where \( L_o \) is for objectness classification, \( L_p, L_v, L_s \) and \( L_w \) are for regressions of point-wise graspable landscape, view-wise graspable landscape, grasp scores and gripper widths respectively. \( L_p \) and \( L_s \) are calculated only if the related points are on objects, \( L_v \) is calculated for views on seed points and \( L_w \) is calculated for grasp poses with ground truth scores \( > 0 \). We use softmax for classification tasks and smooth-L1 loss for regression tasks.

5. Experiments
5.1. Implementation Details

Benchmark Dataset GraspNet-1Billion [13] is a large-scale dataset for grasp pose detection, which contains 190 scenes with 256 different views captured by two cameras (RealSense/Kinect). The testing scenes are divided into three splits according to the object categories (seen/similar/novel). A unified evaluation metric is proposed to benchmark both image based methods and point cloud based methods. We adopt this benchmark as it aligns well with real-world grasping.

Data Processing and Augmentation The point cloud is downsampled with voxel size 0.005m before being fed into the network, and contains only XYZ in camera coordinates. Input clouds are augmented on the fly by random flipping along YZ plane and random rotation around Z axis in \( \pm 30^\circ \).

Implementations To obtain graspness for scenes in GraspNet-1Billion, we follow the process illustrated in Sec. 3.2 since it contains abundant grasp pose annotations. For each point, it densely labels grasp quality score for 300 different views and 48 grasps for each view. Thus, our approach directions \( V \) and grasp candidates per view \( L \) are set as 300 and 48.

For our network, the backbone network adopts an encoder-decoder architecture and outputs feature vectors of channel \( C = 512 \). In visual selection module, \( M = 1024 \) seed points and \( V = 300 \) views are sampled, and the threshold \( \delta^p \) is set to 0.1. The size of MLP used for \( h^p \) is \((512, 3)\) and \( h^s \) is \((512, 512, 300)\). In cylinder-grouping, we sample \( K = 16 \) seed points in the cylinder space with radius \( r = 0.05m \) and height range of \([-0.02m, 0.04m]\). We divide in-plane rotation angles into \( A = 12 \) classes (15° per class) and use \( D = 4 \) classes for approaching distances (0.02m, 0.03m, 0.04m). The two MLPs used to process attentional proposals and output grasp scores and gripper widths have the size of \((512, 256, 256)\) and \((256, 256, 96)\) respectively. Finally the network outputs grasp scores and gripper widths for \( A \times D = 48 \) classes. In loss functions, we set \( \alpha, \beta, \lambda = 10, 10, 10 \).

Training and Inference Our model is implemented with PyTorch and trained on Nvidia GTX 1080Ti GPUs for 10 epochs with Adam optimizer [23] and the batch size of 4. The learning rate is 0.001 at the first epoch, and multiplied by 0.95 every one epoch. The network takes about 1 day to converge. During training, we use one GPU for model updating and one GPU for label generation. In inference, we only use one GPU for fast prediction.

5.2. Performance of Cascaded Grasps Model
Cascaded grasps model is proposed to distinguish graspable areas in various scenes, thus the generality and stability across different domains are important for the model. Here we design an experiment to illustrate its generality and stability.

Evaluation Metric The ranking error is used to quantitatively evaluate the function approximation ability of the model. We divide the range of graspness score into \( K \) bins uniformly and convert the contiguous scores to discrete ranks. The ranking error is defined as the mean rank distances between predictions and labels:
\[ e_{\text{rank}} = \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{|\hat{r}_i - r_i|}{K}, \]
where \( r_i, \hat{r}_i \in \{0, 1, ..., K - 1\} \) stand for the ranks for predictions and labels respectively, and \( N_r \) is the number of predictions. We set \( K = 20 \) in experiments. \( e_{\text{rank}}^v \) and \( e_{\text{rank}}^p \) are used to denote the ranking error of point-wise graspness score and view-wise graspness score respectively.

Inference in Different Domains We conduct three groups of experiments where the dataset is split by object categories, viewpoints and cameras respectively (detailed in Tab. 1). In the first group, we train the model on scene 0-99, and test it on scenes with three object categories (seen, similar and novel). The second group divides the 256 viewpoints into 3 sets, trains the model on viewpoint 0-127, and tests on three viewpoint sets respectively. The third group trains the model on Kinect captured data, and tests the performance on data captured by RealSense.

The results are summarized in Tab. 1. For point-wise grasps model prediction, we can see that the difference between \( e_{\text{rank}}^v \) of seen and novel categories is not obvious. View variation also has a low impact on point-wise grasps model prediction. The \( e_{\text{rank}}^v \) of RealSense is higher than Kinect, but the distance is still in an acceptable range. For view-wise grasps model prediction, \( e_{\text{rank}}^v \) in all groups are nearly un-
changed. These experiments prove the stability and generality of the cascaded graspness model when transferred to new domains.

5.3. Comparing with Representative Methods

We compare our method with previous representative methods. GG-CNN [30] and Chu et al. [8] are rectangle-based methods which take images as input. GPD [41] and Liang et al. [27] classify grasp candidates generated by rule-based point cloud sampling. Fang et al. [13] propose an end-to-end network which predicts grasp poses directly from scene point clouds.

We test our method in three object categories respectively and report the results in Tab. 2. The models for Realsense and Kinect are trained separately. Our method outperforms previous methods by a large margin on both cameras without any post-processing. Compared with Fang et al., the previous state-of-the-art method, GSNet improves the performance by ∼2x on AP metric [13]. Notably, on the most difficult metric AP0.4, GSNet still achieves a great relative improvement (> 140%) on all categories. Fig. 5 presents the qualitative results of our network. The top-1 grasp accuracy on three categories are 78.2/27.64, 62.88/57.64 and 28.97/24.04 for Realsense/Kinect input.

We also report the results after simple collision detection using a parallel-jaw gripper model, where all grasps collided with scene points are removed. The results are improved by 1.42/2.31 AP, 1.06/1.79 AP and 0.33/0.77 AP on the three categories respectively.

5.4. Boosting with Cascaded Graspness Model

We apply the cascaded graspness model(CGM) to GPD, Liang et al., and Fang et al. directly and compare the results with the original methods. For Fang et al., we simply replace ApproachNet with our module. For GPD and Liang et al., we first determine the grasp candidate points using our predicted point-wise graspable landscape, followed by their post processing of Darboux frame estimation and grasp images/clouds classification.

In the middle of Tab. 2, we show the results after adding the CGM. Both the two-step methods and the end-to-end method achieve significant performance gains, proving the effectiveness of cascaded graspness model. Graspable landscapes can not only improve candidate qualities, but also reduce the huge computation time caused by densely sampling.

5.5. Analysis

Effects of Graspable FPS/PVS In Sec. 4.1 we use graspable FPS to sample seed points from graspable landscapes, while other sampling methods can also be applied to the network. We compare our sampling method with three alternatives: a) random sampling from the whole point cloud; b) FPS from the whole point cloud; c) random sampling from graspable landscapes. Tab. 3 shows the results of the models trained using different sampling methods. FPS outperforms random sampling by at least 4.98 AP and sampling with graspable landscapes improves the results by over 7 AP for both FPS and random sampling, which proves the effectiveness of graspable FPS.
Scene 108  Scene 121  Scene 125  Scene 148  Scene 175

Figure 5. Qualitative results of GSNet. Top 50 grasps after grasp-NMS[13] are displayed.

<table>
<thead>
<tr>
<th>Point Sampling</th>
<th>View Sampling</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>graspable PVS</td>
<td>46.17</td>
</tr>
<tr>
<td>FPS</td>
<td>graspable PVS</td>
<td>51.15</td>
</tr>
<tr>
<td>graspable random</td>
<td>graspable PVS</td>
<td>53.32</td>
</tr>
<tr>
<td>graspable FPS</td>
<td>normal</td>
<td>55.63</td>
</tr>
<tr>
<td>graspable FPS</td>
<td>top-1 score</td>
<td>58.34</td>
</tr>
<tr>
<td>graspable FPS</td>
<td>graspable PVS</td>
<td>59.70</td>
</tr>
</tbody>
</table>

Table 3. Comparison of different sampling methods. "top-1 score" stands for selecting the view with the highest graspness score.

<table>
<thead>
<tr>
<th>Landscape</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>object-level</td>
<td>55.33</td>
</tr>
<tr>
<td>scene-level</td>
<td>59.70</td>
</tr>
</tbody>
</table>

Table 4. Landscape types.

<table>
<thead>
<tr>
<th>Camera</th>
<th>CGM</th>
<th>GOM</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealSense</td>
<td>0.08s</td>
<td>0.02s</td>
<td>0.10s</td>
</tr>
<tr>
<td>Kinect</td>
<td>0.10s</td>
<td>0.02s</td>
<td>0.12s</td>
</tr>
</tbody>
</table>

Table 6. Inference speed on GraspNet-1Billion. “CGM” is cascaded graspness model and “GOM” is grasp operation model.

Selection of Landscape Representations In Sec. 3.2 we extend object-level graspness scores to cluttered scenes. Tab. 4 shows that sampling from scene-level graspness performs better than object-level counterpart. The representation for graspness score also has multiple choices. We replace the original definition ratio of feasible grasps with mean and maximum grasp quality scores respectively in the calculation of view-wise graspness scores, and the results in Tab. 5 shows that feasible grasp ratio performs the best.

Model Speed Tab. 6 shows the inference time of our method. Cascaded graspness model achieves a high speed on RealSense/Kinect data, which can also provide accurate sampling for various grasp detection methods. GPD and PointNetGPD take >1s while ours takes only ~0.1s.

5.6. Real Grasping Experiments

We also conduct grasping experiments for cluttered scenes in the real-world setting. The configuration of our experimental setup is illustrated in supplementary materials. The experiments are conducted on a UR-5 robotic arm with an Intel RealSense D435 camera and a Robotiq two-finger gripper. During experiments, we only keep the points on table workspace for speed up.

We conduct grasping experiments in six cluttered scenes. Each scene contains 6-8 objects selected from GraspNet-1Billion. Objects are put together randomly and we repeat the grasping pipeline until the table are cleaned. The success rate is defined as the ratio of object number and attempt number. Tab. 7 reports the grasping performance, which proves the effectiveness of our method. A comparison with other baselines is detailed in supplementary materials.

6. Conclusion

In this paper, we propose a novel geometrically based quality named graspness. A look-ahead searching method is adopted as our graspness measure and we statistically demonstrate its effectiveness and rationality. An end-to-end network is developed to incorporate graspness into grasp pose detection problem, wherein an independent model learns the graspable landscapes. We conduct extensive experiments and demonstrate the stability, generality, effectiveness and robustness of our graspness model. Large margin of improvements are witnessed for previous methods after equipping with our graspness model, and our final network sets a high record for both accuracy and speed.

Acknowledgement 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 and Shanghai Qi Zhi Institute, SHEITC(2018-RGZN-02046).
References


