# Supplemental Material for GA-CNN: A Gripping Attention Neural Network for Random Picking with a Soft Vacuum Gripper

Hui Zhang, *member, IEEE*, Jef Peeters, Eric Demeester, Karel Kellens

## I. QP for Grasp Quality Estimation

To calculate the force-torque wrench in Fig. 4 (d) in the paper, a set of orthogonal normal is formulated in Eq. (1). Let $p > 0$ be the air-pressure differential between the gripping pad and the atmosphere, and $\mu > 0$ be the coefficient of friction between the gripping pad and the target object. In this paper, the coefficient of friction is set with a value of $\mu = 0.5$, which is a typical constant for the contact of a rubber gripping pad and unknown objects [1]–[3]. Nevertheless, how to estimate the air-pressure differential $p$ in the grasp simulation remains challenging. The value of $p$ in the real world depends on the operating curve of the fan type, the geometry of the tubing, the power of pump and so on, which is a computationally expensive problem in the simulation involving Computational Fluid Dynamics (CFD). Hence, the value of $p$ is estimated by the monotonically increasing function in Eq. (2).

Given a triangle $t_j$, $\boldsymbol{r}_z^{t_j}$ is the moment arm for the geometric center of $t_j$ towards the $z$ axis, $(n_x^{t_j}, n_y^{t_j}, n_z^{t_j})$ is the surface normal of $t_j$, and $A^{t_j}$ is the area of $t_j$. Then the sub force-torque wrench is computed by Eq. (3)-(6).

A force-torque wrench for a soft contact model is restricted by the elliptical equation [4]. Therefore, the limitations of $[\|\boldsymbol{F}_x\|, \|\boldsymbol{F}_y\|, \|\boldsymbol{F}_z\|, \|\boldsymbol{T}_z\|]^T$ can be calculated by Eq. (7)-(11), where $\boldsymbol{F}_v$ is the vacuum force on the whole gripping pad.

Furthermore, the weight value $w_j$ of each sub gripping pad is also limited by the physical properties of the used gripper. Specifically, the air-pressure differential $p$ on the contact surface is generated from a unique air-flow tube on the gripper base, and thus differential of weight values $w_j$ cannot be too large, which can be briefly restricted in Eq. (12)-(13).

Given a weight matrix $W = [w_1, w_2, \cdots, w_j, \cdots, w_n]^T$, all restrictions in Eq. (7)-(13) can be combined and converted into a formula in Eq. (14), wherein the restriction matrices $L \in \mathbb{R}^{(8+n)\times n}$ and $l \in \mathbb{R}^{(8+n)\times 1}$ are converted from Eq. (7)-(11) and Eq. (13), and $J \in \mathbb{R}^{1\times n}$ is an all-ones matrix to reformulate the constraint in Eq. (12). In other words, the physical restrictions of $[\|\boldsymbol{F}_x\|, \|\boldsymbol{F}_y\|, \|\boldsymbol{F}_z\|, \|\boldsymbol{T}_z\|]^T$ are converted into the limitations of $W$.

According to Eq. (11)-(12) in Section IV-F of the paper, it is derived that $q = e^{-min\|s(GW-\boldsymbol{\Lambda})\|} \propto -min\|GW - \boldsymbol{\Lambda}\| \propto -min\|GW - \boldsymbol{\Lambda}\|^2 \propto -min(0.5W^TGW - \boldsymbol{\Lambda}^TGW)$. Therefore, the grasp quality estimation in this paper can be seen as the minimization of $0.5W^TGW - \boldsymbol{\Lambda}^TGW$ subjected to the conditions in Eq. (14), which is solved by QP.

Corresponding author: Hui Zhang. *e-mail*: hui.zhang@kuleuven.be

$$\boldsymbol{n}_x = (1,0,0), \quad \boldsymbol{n}_y = (0,1,0), \quad \boldsymbol{n}_z = (0,0,1) \tag{1}$$

$$p = (\frac{\sum A^{t_j}}{A^{Reg.V}})^2, t_j \in Reg.V, p \subseteq [0,1] \tag{2}$$

$$\boldsymbol{f}_x^{t_j} = (pA^{t_j}n_x^{t_j} + \mu pA^{t_j}n_x^{t_j})\boldsymbol{n}_x \tag{3}$$

$$\boldsymbol{f}_y^{t_j} = (pA^{t_j}n_y^{t_j} + \mu pA^{t_j}n_y^{t_j})\boldsymbol{n}_y \tag{4}$$

$$\boldsymbol{f}_z^{t_j} = (pA^{t_j}n_z^{t_j})\boldsymbol{n}_z \tag{5}$$

$$\boldsymbol{\tau}_z^{t_j} = \boldsymbol{r}_z^{t_j} \times (\boldsymbol{f}_x^{t_j} + \boldsymbol{f}_y^{t_j}) \tag{6}$$

$$\boldsymbol{F}_v = pA^{\mathcal{G}}\boldsymbol{n}_z = \sum_{j=1}^{n} pA^{t_j}\boldsymbol{n}_z \tag{7}$$

$$|\boldsymbol{F}_x \cdot \boldsymbol{n}_x| = |\sum_{j=1}^{n} w_j \boldsymbol{f}_x^{t_j}\boldsymbol{n}_x| \leq \frac{\sqrt{3}}{3}\mu\boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{8}$$

$$|\boldsymbol{F}_y \cdot \boldsymbol{n}_y| = |\sum_{j=1}^{n} w_j \boldsymbol{f}_y^{t_j}\boldsymbol{n}_y| \leq \frac{\sqrt{3}}{3}\mu\boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{9}$$

$$0 \leq \boldsymbol{F}_z \cdot \boldsymbol{n}_z = \sum_{j=1}^{n} w_j \boldsymbol{f}_z^{t_j}\boldsymbol{n}_z \leq \boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{10}$$

$$|\boldsymbol{T}_z \cdot \boldsymbol{n}_z| = |\sum_{j=1}^{n} w_j \boldsymbol{\tau}_z^{t_j}\boldsymbol{n}_z| \leq \frac{\sqrt{3}}{3}\mu r_{\mathcal{G}}\boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{11}$$

$$\sum_{j=1}^{n} w_j = n \tag{12}$$

$$1 - \sigma_W \leq w_j \leq 1 + \sigma_W, \quad \sigma_W = 0.1 \tag{13}$$

$$LW \leq l, \quad JW = n \tag{14}$$

## II. Key Parameters for Detaset Generation

As mentioned in the paper, main parameters of the proposed 6-step grasp simulation with the adopted gripper include the radius of *Reg.V*, the radius of the soft gripper pad $\mathcal{G}$, the altitude/height of $\mathcal{G}$, the rotation step $\Delta_r$, the radius step $\Delta_\theta$, and the distance step $\Delta_d$, as defined in Table I for the dataset collection.

## III. Pseudo Code for Grasp simulation

During the grasp simulation, both $\boldsymbol{P}_{\mathcal{O}}$ and $\boldsymbol{P}_{\mathcal{G}}$ are in SE(3) for each grasp. The grasp quality is evaluated by solving the QP within 100 iterations, and the sub point cloud is desampled into 24×24 points around the contract surface. 35 high-resolution 3D meshes were selected from the YCB dataset, and their sizes were rescaled to simulate grasp scenarios with

TABLE I
MAIN PARAMETERS AND THEIR DESCRIPTIONS IN THE SIMULATION.

| Parameter | Description | Value (metric) |
|---|---|---|
| $r_v$ | Radius of *Reg.V* | 20.0 mm |
| $r_{\mathcal{G}}$ | Radius of $\mathcal{G}$ | 25.0 mm |
| $H$ | Altitude of gripping pad | 30.0 mm |
| $\Delta_\theta$ | Rotation step | $0.045\pi$ rad |
| $\Delta_r$ | Radius step | $\leq$2.0 mm |
| $\Delta_d$ | Distance step | $\leq$2.0 mm |

various objects. Over 100 K grasps were synthesized, running 200 hours on the PC introduced in the paper.

**Algorithm 1** Synthesize a Grasp Example.

**Assumptions:**

A-1. Quasi-static physics with Coulomb friction.
A-2. The object has an airtight surface and a rigid body.
A-3. Ignore the weight of a target object and the torques on the $x$, $y$ axes.

**Input:**

3D mesh of the object $\mathcal{O}$, virtual gripper $\mathcal{G}$, virtual camera $\mathcal{C}$.

**Output:**

Grasp quality $q$, point cloud $\mathcal{P}$.

**Steps:**

S-1.01: $\boldsymbol{P}_{\mathcal{O}}$ = RandomSet($\mathcal{O}$) in SE(3).
S-1.02: Build a geometric model for $\mathcal{G}$ in GCS by parameters $r_{\mathcal{G}}$, $\mathbb{V}$, $\mathbb{T}$...
S-1.03: Set geometric restrictions by $r$, $\theta$, $z$, $\Delta_r$, $\Delta_\theta$...
S-1.04: $\boldsymbol{c}$ = RandomPoint($\mathcal{O}$).
S-1.05: $\boldsymbol{d}_{\mathcal{G}}$ = RandomSet($\boldsymbol{c}, \mathcal{G}$).
S-1.06: $\boldsymbol{P}_{\mathcal{G}}$ = RandomSet($\boldsymbol{c}, \boldsymbol{d}_{\mathcal{G}}$) in SE(3).
**for** $k \leftarrow 1$ to $m$ **do**:
S-1.07: Track $\mathbb{V} \leftarrow \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k, \cdots, \boldsymbol{v}_m\}$.
**end for**
S-1.08: Update $\mathbb{T} = \{t_1, \cdots, t_j, \cdots, t_n\} \leftarrow \mathbb{V}$.
**for** $j \leftarrow 1$ to $n$ **do**:
S-1.09: Compute $\boldsymbol{f}_x^{t_j}$, $\boldsymbol{f}_y^{t_j}$, $\boldsymbol{f}_z^{t_j}$ and $\boldsymbol{\tau}_z^{t_j}$.
**end for**
S-1.10: $G \leftarrow \{\boldsymbol{f}_x^{t_j}, \cdots, \boldsymbol{f}_y^{t_j}, \cdots, \boldsymbol{f}_z^{t_j}, \cdots, \boldsymbol{\tau}_z^{t_j}, \cdots\}$.
S-1.11: Set physical restrictions for $\mu$, $p$, $\boldsymbol{F}_x$, $\boldsymbol{F}_y$, $\boldsymbol{F}_z$, $\boldsymbol{T}_z$...
S-1.12: Solve $q = e^{-min\|s(GW-\boldsymbol{\Lambda})\|}$ by QP.
S-1.13: $\boldsymbol{P}_{\mathcal{C}}$ = Set($\mathcal{C}$) in WCS.
S-1.14: Render $\mathcal{P}$ in CCS.
S-1.15: $\mathcal{P}$ = Desample(Disturb(Transform($\mathcal{P}$))).

## IV. TRAINING AND FINE-TUNING OF GA-CNN

More than 50 similar networks were trained to find the optimal architecture for the GA-CNN regarding both the prediction error and computational complexity. For each grasp example in the testing dataset, the prediction error of the GA-CNN is denoted as $|\hat{q} - q|_{abs}$. Fig. 1 presents the average prediction error and execution time affected by the jointly varying channels and depths of CBAMs in the GA-CNN. In detail, the depth of the GA-CNN varies from 18 layers to 102 layers with four different combinations of the CBAM channels.

With the increase of the depth and channels, the computational complexity of GA-CNNs consistently grows, but their prediction errors do not constantly decrease. Especially when a GA-CNN has more than 80 layers or 32-64-128-256 channels, the prediction error often enlarges due to overfitting. Consequently, the 52-layer GA-CNN with 64-128-256-512 channels keeps satisfying performance in the aspects of both the average error and computational complexity. The final GA-CNN is constructed by a 2-layer CNN with a 3×3 kernel, and 4 CBAMs with 64, 128, 256 and 512 channels respectively, containing 3.23 M parameters. It reports an average error of 0.049 and spends 0.27 ms for each synthetic grasp in the test dataset.

## V. DEFINITION OF CLUTTER LEVELS

Despite the fact that abundant benchmark objects are available online for robotic grasping tests, many items from the benchmarks are not always suitable for the investigated grippers in the aspects of sizes, weights and rigidity. A fair criterion is needed to benchmark clutters in physical grasping. In this paper, the complexities of grasp scenes are defined with nine-level metrics for the random picking with vacuum grippers.

In this research, the complexity of the object's shape is defined based on the *NSVR* in (15), where $A_{sur}$ and $V_{obj}$ are the surface area and volume of the object. Definitely, it is a property of the object and not related to any gripper. The *NSVR* keeps the same when an object is rescaled.

$$NSVR = \frac{\sqrt{A_{sur}}}{\sqrt[3]{V_{obj}}} \tag{15}$$

Fig. 2 lists some objects and their *NSVR*s from the Dex-Net [1], KIT [5] and YCB [6] grasping databases. The *NSVR*s of more than one hundred 3D meshes from these databases were calculated, and the complexities of objects can be divided into three categories based on *NSVR*s: Basic ($0 < NSVR < 2.6$), Typical ($2.6 \leq NSVR < 3.5$), and Complex ($3.5 \leq NSVR$). With these criteria, any real-world object can be classified based on the *NSVR* of its reconstructed 3D model as exhibited in Fig. 3 (a)-(c).

Moreover, the distribution of objects in a grasp scene can be classified into three levels: isolated, multiple and stacked, as shown in Fig. 3 (d)-(f). In a grasp scene with isolated objects, the objects are manually deployed with random poses on the table in SE(2), and the minimum gap between each object and its neighbors is not smaller than 5 cm. Similarly, a grasp scene with multiple objects is manually deployed on the table in SE(2), where the maximum gap between each object and its neighbors is within 1 cm. As a comparison, stacked objects in a grasp scene stay with random poses in SE(3) and touch each other, which can be deployed following the method in the existing work [7].

Therefore, the complexities of grasping clutters are divided into nine levels in Table II. Fig. 3 (d)-(f) illustrate part of the items for the subsequent physical experiments, including 5 adversarial objects from the Dex-Net dataset. In the subsequent physical experiments, every clutter consisted of 10 randomly

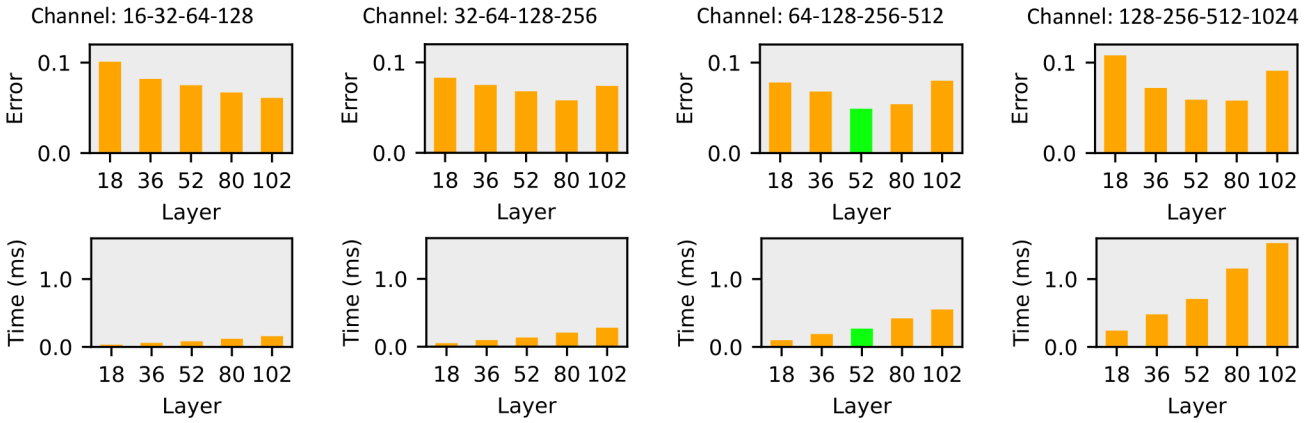


Fig. 1. Performance of the GA-CNN with different channels and depths of CBAMs. Note: Change the depths of CBAMs to get GA-CNNs with different layers, and green bars mark the selected architecture.

TABLE II
THE COMPLEXITY LEVELS OF GRASPING CLUTTERS.

| Level \ Dist. / Comp. | Isolated | Multiple | Stacked |
|---|---|---|---|
| Basic | 1 | 2 | 3 |
| Typical | 4 | 5 | 6 |
| Complex | 7 | 8 | 9 |

Note: 1) Comp. is the abbreviation of "Complexity." 2) Dist. is the abbreviation of "Distribution."

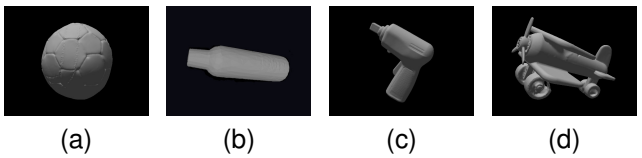


Fig. 2. Several 3D models from benchmark datasets and their *NSVR*s. (a) $NSVR = 2.22$. (b) $NSVR = 2.42$. (c) $NSVR = 2.63$. (d) $NSVR = 3.87$.

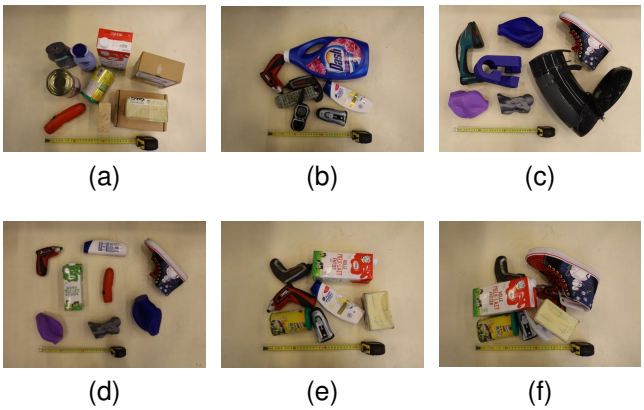


Fig. 3. Examples of objects' complexities and distributions. (a) A set of basic objects. (b) A set of typical objects. (c) A set of complex objects. (d) A set of isolated objects. (e) Multiple objects on the table in SE(2). (f) A set of stacked objects.

distributed objects, and five clutters were deployed for the test at each level (50 objects in total).

## VI. CLOSED-LOOP GRASPING

Multi-perspective grasping is developed for the proposed grasping method to improve its performance, integrating the real-time point cloud of robotic observation and the 6-DoF force-torque wrench of the gripper base to develop the closed-loop control, as shown in Fig. 1 of our paper. The role of the real-time force-torque has been explained in our published work [8]. It is mainly utilized to monitor the grasp status, like collision and grasp success, optimize the robotic motion, and minimize the moments on the gripper base during grasping.

Different from the previous work [8], the real-time point cloud $\mathbb{P}$ is evaluated at each timestep in the closed-loop control. Besides, the proposed reactive grasping method is able to detect feasible grasp poses in SE(3). As mentioned in Section III-D of the paper, a number of sub point clouds $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_i, \cdots, \mathcal{P}_u\}$ are randomly sampled and transformed into GCS for the grasp quality prediction. The grasp direction of $\mathcal{P}_i$ is calculated based on its average surface normal.

The proposed multi-perspective grasping method merges more than one point cloud from multiple viewpoints, for example, the static and wrist-mounted cameras in Fig. 4, to acquire a global point cloud and detect a globally optimized grasp pose before grasping. Afterward, the robot adjusts the gripper pose towards the globally optimized grasp pose. Then the wrist-mounted camera keeps activated to observe the clutter in a local scope, optimize grasp pose and prune the grasp motion during grasping. A multi-viewpoint point cloud ensures that the robot can overcome visual occlusions and adapt the poses of the gripper and wrist-mounted camera before grasping. The single-viewpoint point clouds during grasping ensure the proposed grasping method runs with 15 Hz real-time speed when the robot approaches a target object.

## VII. ROBOTIC SETUP

The robotic system is depicted in Fig. 4. It is composed of a 6-DoF robot (KUKA LBR IIWA 14 R820), a versatile vacuum gripper with a radius of 75 mm (FORMHAND FH-R150) that is much larger than the virtual gripper in simulation, a static camera (Microsoft Kinect Version 2), a wrist-mounted camera

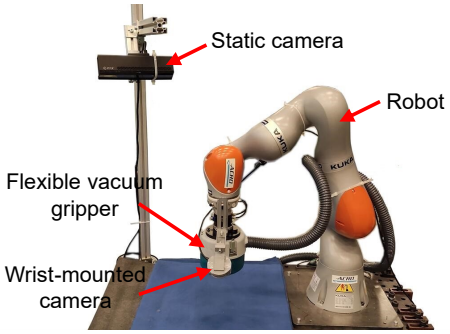


Fig. 4. Robotic setup for the experiments.

(Intel RealSense L515) and a PC, linked via ROS nodes [9]. Notably, the force-torque wrench in the closed-loop control is projected from the joint torque sensors on the KUKA IIWA, and necessary configurations and calibration are needed for the robot regarding the weights of the used gripper, wrist-mounted camera and flange.

## VIII. Failure Modes

There are four typical failure grasp cases about the proposed grasping method.

First, Fig. 5 (a) reports a failed grasp for a transparent object, since the structured light of the depth camera is unable to be reflected on a transparent surface, and the robot fails to detect the object.

Second, Fig. 5 (b) shows a failed grasp for two neighbor objects with the same height. The flat boundary cannot be detected due to the limited resolution of the depth camera. Implementing a segmentation algorithm before the grasp quality prediction is a potential solution for this problem. However, the segmentation [10] and tracking [11] of unknown objects in a clutter are still complicated.

Moreover, the proposed grasping method has mediocre performance for the random picking of slim objects, as illustrated in Fig. 5 (c)-(d). Specifically, while the GA-CNN grasping method detects the feasible grasp regions for both grasping trials, a failed grasp is still reported in Fig. 5 (d) when two slim objects are near-distributed and the neighbor object inhibits the soft gripping pad from fitting the target object. The grasp success is affected by the dimensions of objects and gripper.

Additionally, the closed-loop GA-CNN grasping merely runs within 15 Hz on the used PC due to the deep architecture of the GA-CNN. The decrease in grasping performance can be seen if objects move too fast.

In consequence, the applicability of the proposed closed-loop grasping method is also related to physical setups and target objects. Essential adjustments are needed to fit different use cases. For instance, an appropriate gripper's size should be decided regarding the objects' sizes in physical grasping trials.

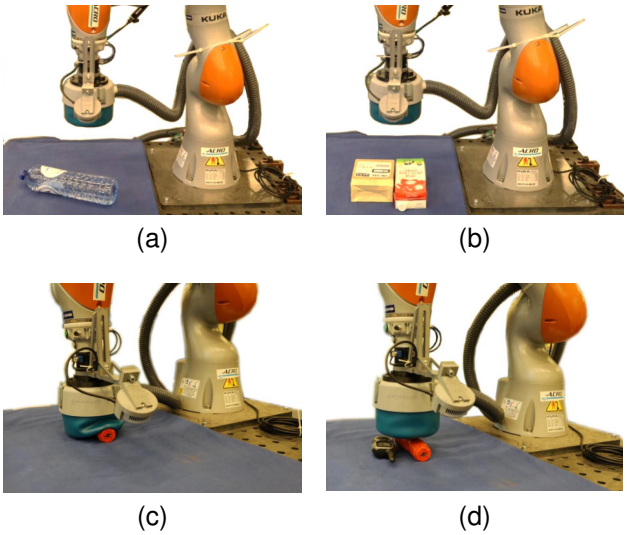


Fig. 5. Some failed grasping trials. (a) Grasping trial for a transparent object. (b) Grasping trial for two flat objects with a similar height. (c) Successful grasping trial for a single and slim object. (d) Failed grasping trial for two near-distributed and slim objects.

## References

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-Net 2.0: deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017, *arXiv:1703.09312*. [Online]. Available: https://arxiv.org/abs/1703.09312.

[2] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2018, pp. 5620–5627.

[3] H. Liang, X. Ma, S. Li, M. Gorner, S. Tang, B. Fang, F. Sun, and J. Zhang, "PointNetGPD: detecting grasp configurations from point sets," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2019, pp. 3629–3635.

[4] I. Kao, K. M. Lynch, and J. W. Burdick, "Contact modeling and manipulation," in *Springer Handbook of Robotics*. Cham, Germany: Springer, 2016, pp. 931–954.

[5] A. Kasper, Z. Xue, and R. Dillmann, "The KIT object models database: an object model database for object recognition, localization and manipulation in service robotics," *International Journal of Robotics Research*, vol. 31, no. 8, pp. 927–934, May 2012.

[6] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: using the Yale-CMU-Berkeley object and model set," *IEEE Robotics and Automation Magazine*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[7] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, Oct. 2017.

[8] H. Zhang, J. Peeters, E. Demeester, and K. Kellens, "A CNN-based grasp planning method for random picking of unknown objects with a vacuum gripper," *Journal of Intelligent and Robotic Systems*, vol. 103, no. 64, pp. 1–19, Nov. 2021.

[9] C. Hennersperger, B. Fuerst, S. Virga, O. Zettinig, B. Frisch, T. Neff, and N. Navab, "Towards MRI-based autonomous robotic US acquisitions: a first feasibility study," *IEEE Transactions on Medical Imaging*, vol. 36, no. 2, pp. 538–548, Feb. 2017.

[10] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick, "Learning to segment every thing," in *Proceedings of the IEEE Conference on Computer Vision and Patttern Recognition*, 2018, pp. 4233–4241.

[11] F. Leeb, A. Byravan, and D. Fox, "Motion-nets: 6D tracking of unknown objects in unseen environments using RGB," 2019, *arXiv:1910.13942*. [Online]. Available: https://arxiv.org/abs/1910.13942v1.