2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Learning-based Robotic Bin-picking with Flexibly Customizable Grasping Conditions

Hiroki Tachikake[1] and Wataru Watanabe[1]

***Abstract*— A practical robotic bin-picking system requires a high grasp success rate for various objects. Also, the system must be capable of coping with various constraints and their changes flexibly. To resolve these issues, this study proposes a novel deep learning-based method that exploits a simulator to generate desired grasping actions. The features of this method are as follows: (1) Grasping conditions for any object can be flexibly customizable in the simulated environment to improve the real-world grasping actions. (2) Sensor input (RGB image) is directly regressed to grasping actions by using convolutional processing. Owing to these features, the system using the proposed method can grasp objects with geometric variations, semi-transparent objects, and objects with a biased center of gravity. Experimental results on a real robot system show that the proposed method exhibits a high grasp success rate for four types of objects with different physical and geometric properties as well as additional constraints of grasping condition.**

I. INTRODUCTION

Bin picking is a task wherein a single object is picked out of a cluttered bin. It is not a core part of assembly lines. Manual bin-picking is a time-consuming task; therefore, industrial warehouses must automate it to achieve enhanced productivity. To realize a practical bin-picking system, it is necessary to achieve a high grasping success rate for various objects with different physical properties such as the geometric shape (irregular), mass distribution, and surface friction. In addition, to improve the quality of post-processing work, it is necessary for the system to be capable of coping with various constraints and their changes flexibly; these constraints can be enforced on parameters such as the grasping position, grasping posture, and posture change after grasping (Fig. 1).

Several studies have been conducted to resolve the problems discussed earlier. For example, there are methods that estimate the posture of an object loaded in bulk by three-dimensional matching of the sensor data with a geometric-shape model of the object [1], [2]; then, the grasping posture is determined based on analytic grasping quality metrics [3], [4]. These methods are not effective when the object shapes are irregular because it is difficult to allow geometric variations owing to the inherent properties of three-dimensional matching. In contrast, a method proposed in a prior study matches the geometric shape of a hand with sensor data to grasp various objects [5]. This method does not consider the geometry of the entire object. Therefore, it is difficult

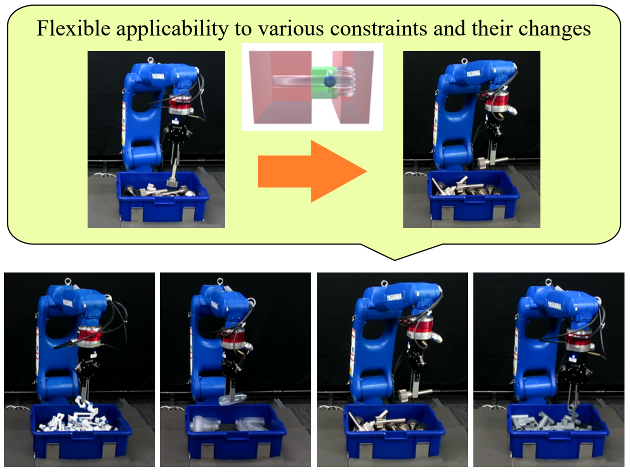

Flexible applicability to various constraints and their changes
Capability to grasp various objects

Fig. 1. The difficulty of a practical bin-picking in the industrial field. For a practical bin-picking system, the system is necessary to grasp various objects and be able to cope with various constraints and their changes. In this paper, we provide a solution for this problem.

to grasp objects without the information necessary for successful object grasping that depends on the geometric shape of the entire object (e.g., a center of gravity). Further, the method cannot adapt to additions or changes in constraints on the grasping condition.

Several other studies have investigated deep learning-based methods [6], [7]. Many studies have demonstrated that using this method is a good approach to successfully grasping a variety of objects. Furthermore, features suitable for different scenes can only be changed by retraining. To reduce the cost incurred by learning-data collection, some studies have leveraged the power of simulations [8], [9], [10], [11]. The purpose is to collect diverse datasets and generalize a grasping technique to realize robust grasping of unknown objects, as needed in logistics applications. From another point of view, the proposed method proves that grasping can be re-planned in response to changes in environmental conditions where the state is known. If this is explicitly used, the practicality of industrial bin-picking can be improved. In these works, successful grasping was realized by maximizing the criteria represented by learned deep neural networks. For this reason, the generated grasp is often different from that determined in advance. Therefore, it is difficult to apply when the grasp that satisfies the pre-determined constraint is required.

Therefore, the purpose of this study is to propose a novel learning-based method for a practical robotic bin-picking

[1]Hiroki Tachikake and Wataru Watanabe are with the Development Technology Department, AI Cube Inc., 3-14 Nihonbashi, Horidome-cho, 2 Chome, Chuo-ku, Tokyo 103-0012 Japan {Hiroki.Tachikake/Wataru.Watanabe}@ai3cube.co.jp

system. To this end, we propose a system that enables a robot to successfully perform real-world grasping tasks by planning the grasp through simulations. We combine the solutions of the following: (1) building a successful grasp space via an analytic method using a simulator, (2) learning a neural-network model to predict the grasp space from the input, and (3) realizing domain adaptation to generalize the simulation model for the real world. In addition, an RGB image is used as an input to neural networks because it is less affected by different objects.

In this study, we compare the grasping success rates of methods proposed in previous studies with that of the method proposed in this study using four types of objects with different physical properties. We also verify whether the required grasp can be achieved in two cases commonly encountered in the industrial field, i.e., grasping fragile and heavy objects when the robot is not allowed to change the object posture after grasping. Experimental results indicate that the proposed method is capable of grasping various types of objects and adapting to various constraints and their changes.

II. PROBLEM STATEMENT

In this section, we formulate the problem of bin-picking.

A. System Configuration and Assumption

Fig. 2 illustrates the system configuration. The system is composed of a robot arm, a parallel-jaw gripper, objects, and a camera. Σ_w, Σ_c are the coordinate frames attached to the base of the robot arm and the camera, respectively.

We make the following assumptions in this paper:

- The parameters of transformation ${}^{w}\boldsymbol{T}_c$ from Σ_w to Σ_c and camera's intrinsic parameters are known.
- For efficient grasping performance, a parallel-jaw grasp is limited with four degrees of freedom (DoF), including translational movements in X-Y-Z axes of Σ_w and rotations around Z-axis of Σ_w. The direction of Z-axis of Σ_c is aligned with the direction of $-Z$-axis of Σ_w.

B. Parameterization

The grasp parameters are illustrated in Fig. 3. Let $\boldsymbol{g}^s = (\boldsymbol{p}^s, \phi^s, \omega^s)$ for $s \in \{w, c\}$ denote a grasp, where $\boldsymbol{p}^s = [x^s, y^s, z^s]^T \in \mathbb{R}^3$ is a center position of the grasp, ϕ^s is a rotation angle, and ω^s is a stroke of the gripper. The subscripts w and c mean that grasp is defined Σ_w and Σ_c, respectively. Let $\boldsymbol{c}_1^s, \boldsymbol{c}_2^s \in \mathbb{R}^3$ for $s \in \{w, c\}$ denote contact positions between the gripper fingers and the object. Let $\boldsymbol{n}_1^s$ and $\boldsymbol{n}_2^s$ denote normal vectors at the contact points. $\boldsymbol{p}^s$ is calculated by $\boldsymbol{p}^s = 0.5\left(\boldsymbol{c}_1^s + \boldsymbol{c}_2^s\right)$. We assume the geometry of the cross-section of the gripper finger as rectangle, where width is l_1, height is l_2, and depth is l_3.

Let $\boldsymbol{I} \in \mathbb{R}^{H \times W}$ denote RGB Image with height H and width W. Let $\boldsymbol{g}^i = \left(\boldsymbol{p}^i, \phi^i, \omega^i\right)$ denote $\boldsymbol{g}^c$ projected on $\boldsymbol{I}$, which is parameterized by the center position of the grasp $\boldsymbol{p}^i$, a rotation angle, ϕ^i, and stroke of the gripper. Let l_1^i,l_2^i denote l_1,l_2 projected on $\boldsymbol{I}$, respectively. $\boldsymbol{g}^c$ can be converted into $\boldsymbol{g}^i$ by applying a sequence of known transforms ($\boldsymbol{g}^i =$

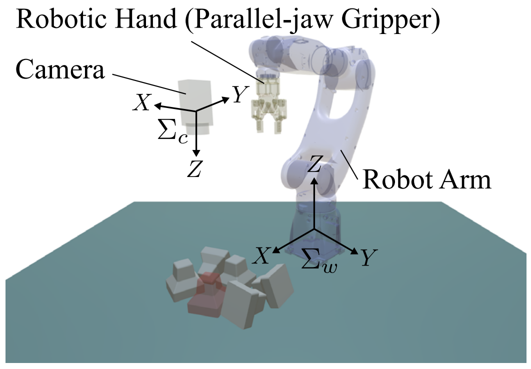


Fig. 2. System configuration and coordinate system.

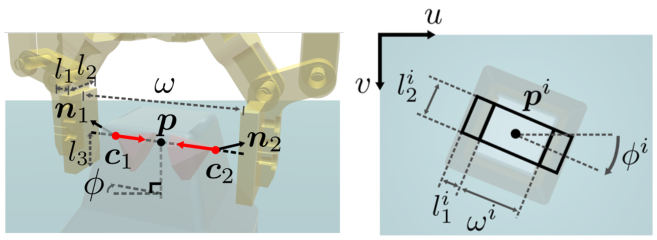


Fig. 3. Illustration of parameterization

${}^{i}\boldsymbol{T}_c\left(\boldsymbol{g}^c\right)$) and the inverse transformation can be calculated ($\boldsymbol{g}^i = {}^{i}\boldsymbol{T}_c\left(\boldsymbol{g}^c\right)$).

C. Objective

The objective of this study is to generate successful grasp space G^* for various objects, as well as to cope with constraints and their changes flexibly. However, it is challenging to build or rebuild G^* in a real world. A practical alternative is to build G^* in the simulation that requires the solution to three complex problems.

1) Existence of Domain Shift between Simulation and Real World: There is a domain distribution shift between the simulation and the real world. We assume a covariate shift, and the problem is to minimize the discrepancy in the marginal distribution of $\boldsymbol{I}$ between simulation and real world.

2) Regression to the Successful Grasp Space: Given $\boldsymbol{I}$, building G^* is the problem to estimate the conditional probability $p\left(G^* | \boldsymbol{I}\right)$. To represent the distribution, we use a deep convolutional neural network $G^*\left(\boldsymbol{I}; \boldsymbol{\theta}\right) : \boldsymbol{I} \rightarrow G^*$ parameterized by $\boldsymbol{\theta}$ and we formulate the problem as following optimization problem:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \, \mathcal{L}\left(G^*, G^*\left(\boldsymbol{I}; \boldsymbol{\theta}\right)\right), \tag{1}$$

where $\mathcal{L}\left(\cdot, \cdot\right)$ indicates a loss function.

3) Building of the Successful Grasp Space: When problems 1) and 2) are resolved, building the appropriate grasp space in simulator such that the mapped grasp space $G^*\left(\boldsymbol{I}; \boldsymbol{\theta}\right)$ converges a subspace of G^* is required.

III. METHOD

In this section, we propose a system for solving three problems that we defined in the previous section. Fig. 4 illustrates the schematic of the system. The system inputs are

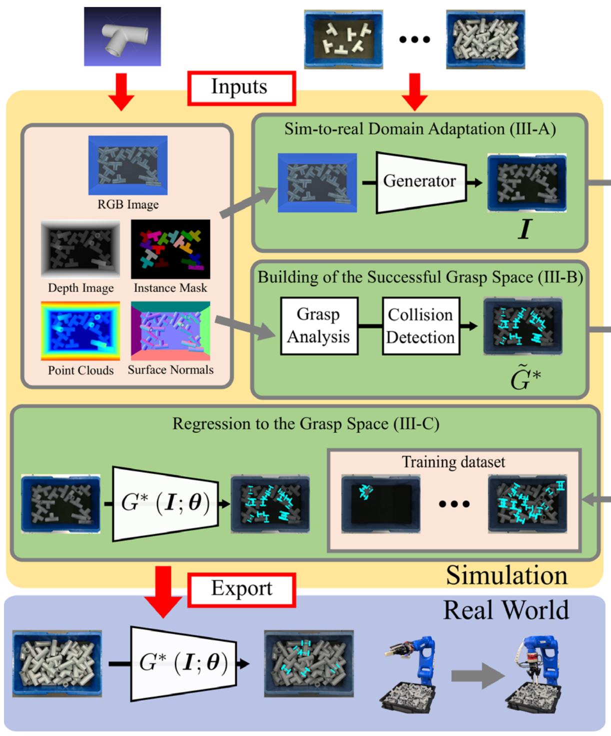


Fig. 4. Schematic overview of our proposed bin-picking system. The system inputs are a small number of RGB images in the real world, an object mesh, a scene configuration, and constraints. If these input data are changed, the picking operation of the robot is changed according to the data by the same process. Please refer to the supplementary video, which gives an overview of our system.

a small number of real-world RGB images, a polygon mesh of the object, a scene configuration (i.e., camera parameters, a bin shape, etc.), and additional constraint conditions parameterized by the gripper's graspable positions on the object. The system is divided into some of its functional components. A physical simulation of randomly placing objects in a bin is conducted, and several data are generated (e.g., RGB Image and Depth Image). The synthetic RGB image is transferred to the real-world domain (see Section III-A). An analytical method is employed to build a successful grasp space considering the constraints, from which grasp labels are created (see Section III-B). The tuple of the transferred RGB Image and the grasp labels are used to train $G^*(\boldsymbol{I};\boldsymbol{\theta})$ (see Section III-C). In the real world, a robotic arm can grasp the object by using $G^*(\boldsymbol{I};\boldsymbol{\theta})$, which is trained in the simulation.

Comparing to the other 3-D matching-based systems that require the query model, the proposed method directly infers the grasp pose from the image without explicit modeling. This feature enables the system to handle a greater variety of graspable objects. Using RGB images as the input of $G^*(\boldsymbol{I};\boldsymbol{\theta})$ also reduces the influence of the difference between objects on the sensor performance. Furthermore, by changing the additional constraints in the simulator, the system can cope with various constraints in the real world.

A. Sim-to-Real Domain Adaptation

To minimize a discrepancy in the marginal distribution of $\boldsymbol{I}$ between the simulation and the real world is known as *domain adaptation*. In this paper, we apply the method proposed by Zhu et al. [12], which is one of the unsupervised domain adaptation methods.

B. Building of the Successful Grasp Space

We propose the method to build a successful grasp space. Let $\tilde{G}^*$ be "semi-suitable" grasping space, where "semi-suitable" means is not always a subspace of G^*. We define $\tilde{G}^*$ as a finite set of grasps in the pixel coordinate frame:

$$\tilde{G}^* = \left\{\boldsymbol{g}_t^i | t \in \{1,\dots,N_t\}\right\}, \tag{2}$$

where $\boldsymbol{g}_t^i$ is t-th grasp in the pixel coordinate frame. N_t is a cardinal number. To build $\tilde{G}^*$, we assume an ideal situation for a successful grasp and define the grasp space which satisfies the situation as $\tilde{G}^*_{\text{base}}$. In addition, we define the grasp space which is necessary for additional constraints as $\tilde{G}^*_{\text{additional}}$. $\tilde{G}^*$ is an intersection of these grasp spaces:

$$\tilde{G}^* = \tilde{G}^*_{\text{base}} \cap \tilde{G}^*_{\text{additional}}. \tag{3}$$

By changing $\tilde{G}^*_{\text{additional}}$, the ground-truth grasp labels generated from $\tilde{G}^*$ and the output of $G^*(\boldsymbol{I};\boldsymbol{\theta})$ are refined to satisfy the constraints, which means that the system can easily deal with various constraints and corresponding changes.

1) Building $\tilde{G}^*_{\text{base}}$: We define the following condition as the ideal state for grasping:

- The ideal state before grasping
  (1) Low risk for collision with other objects and the bin.
- The ideal state after grasping
  (2) No breaking the pile of objects stacked in a bin.
  (3) High ability to resisting external forces such as gravity.

$\tilde{G}^*_{\text{base}}$ is built to satisfy the above conditions in the simulator. First, to satisfy condition (2), i.e., to prevent grasped object applying force on other objects, we select the object that has low overlap with other objects. Specifically, we sort each object in the simulation according to height and select the objects that exhibit a lower overlap with other instance masks than a given threshold. Next, to satisfy condition (3), for each selected object, we generate grasp candidates that satisfy *force closure* [15], [16], [17]. Force closure of parallel-jaw is expressed as follows:

$$\|\boldsymbol{n}_1 + \boldsymbol{n}_2\|_2^2 \leq \alpha, \tag{4}$$

$$\cos^{-1}\left(\frac{\boldsymbol{n}_1^T(\boldsymbol{c}_1 - \boldsymbol{c}_2)}{\|\boldsymbol{c}_1 - \boldsymbol{c}_2\|_2}\right) < \tan^{-1}\mu, \tag{5}$$

$$\cos^{-1}\left(\frac{\boldsymbol{n}_1^T(\boldsymbol{c}_2 - \boldsymbol{c}_1)}{\|\boldsymbol{c}_2 - \boldsymbol{c}_1\|_2}\right) < \tan^{-1}\mu, \tag{6}$$

where μ is a static friction coefficient. $\alpha \in \mathbb{R}$ is a constant to relax the constraint that the normal vectors are opposite [17]. To reduce the processing time, we calculate the contact position and normal vector pair for a range where the Depth Image gradient is larger than a given threshold. Significant differences between the actual and simulated shapes will affect the close force calculation. Thus, we assume that the shape of a polygon mesh resembles the actual dimensions of the real-world object. Finally, to satisfy condition (1), we calculate the possibility of collision of the grasp candidates with the objects and the bin. Let $\boldsymbol{D}$ denote Depth Image. Let $D(u,v)$ denote the value of $\boldsymbol{D}$ at the position $[u,v]^T$ in the pixel coordinate frame. Let $\boldsymbol{D}_{\text{collision}}$ denote the collision region of grasp with the objects and the bin. The value of $\boldsymbol{D}_{\text{collision}}$ is expressed as follows:

$$D_{\text{collision}}(u,v) = \begin{cases} 1 & \text{if} \quad D(u,v) \le z^c + l_3 \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

We define the region of the gripper rectangle which is projected to image plane as $\boldsymbol{D}_{\text{gripper}}$, which takes the value of 1 inside of the region and 0 outside of the region. We select the grasp with the low intersection $\boldsymbol{D}_{\text{collision}} \cap \boldsymbol{D}_{\text{gripper}}$ as collision-free. The set of collision-free grasp candidates is used as ground-truth grasps for training (see Section III-C).

2) Building $\tilde{G}^*_{\text{additional}}$: To cope with various constraints and their changes, $\tilde{G}^*_{\text{additional}}$ can be constructed arbitrarily. In this paper, we consider two application scenes: (1) A grasp is limited to a designated area, and (2) a grasp is limited around the center of gravity. We deal with two constraints by restricting the graspable region. Specifically, among grasps belonging to $\tilde{G}^*_{\text{base}}$, we use grasps whose contact points are in the region as ground-truths.

C. Regression to the Successful Grasp Space

We propose a model architecture of $G^*(\boldsymbol{I};\boldsymbol{\theta})$ and the method for optimizing (1) by a training dataset of a pair $\boldsymbol{I}$ and $\tilde{G}^*$.

1) Model Architecture: Fig. 5 illustrates a model architecture. Given $\boldsymbol{I}$, the model infers a finite set of grasps in the pixel coordinate frame:

$$G^*(\boldsymbol{I};\boldsymbol{\theta}) = \left\{ \boldsymbol{g}^i_d(\boldsymbol{I};\boldsymbol{\theta}) \,|\, d \in \{1,\ldots,N_d\} \right\}, \tag{8}$$

where $\boldsymbol{g}^i_d(\boldsymbol{I};\boldsymbol{\theta}) = \left(\boldsymbol{p}^i_d(\boldsymbol{I};\boldsymbol{\theta}), \phi^i_d(\boldsymbol{I};\boldsymbol{\theta}), \omega^i_d(\boldsymbol{I};\boldsymbol{\theta}) \right)$ is d-th grasp in the pixel coordinate frame parameterized by $\boldsymbol{\theta}$. N_d is a cardinal number. To represent $G^*(\boldsymbol{I};\boldsymbol{\theta})$, we define "default grasp" and associate a set of default grasps with each feature map cell at the top of the network (Fig. 6). In other words, for each feature map cell, the network outputs an offset and an angle from the default grasp. The transformation from these network outputs to $\boldsymbol{g}^i_d(\boldsymbol{I};\boldsymbol{\theta})$ of each feature map is expressed as follows:

$$\boldsymbol{p}^i_d(\boldsymbol{I};\boldsymbol{\theta}) = \begin{bmatrix} W\left(b_{u_o,d} + b_{w,d} l_{u,d}(\boldsymbol{I};\boldsymbol{\theta})\right) \\ H\left(b_{v_o,d} + b_{h,d} l_{v,d}(\boldsymbol{I};\boldsymbol{\theta})\right) \end{bmatrix}, \tag{9}$$

$$\phi^i_d(\boldsymbol{I};\boldsymbol{\theta}) = \frac{1}{2}\tan^{-1}\left(\frac{l_{\sin 2\phi^i,d}(\boldsymbol{I};\boldsymbol{\theta})}{l_{\cos 2\phi^i,d}(\boldsymbol{I};\boldsymbol{\theta})} \right), \tag{10}$$

$$\begin{aligned} \xi = \max\big(& W b_{w,d} \exp\left(l_{w,d}(\boldsymbol{I};\boldsymbol{\theta})\right), \\ & H b_{h,d} \exp\left(l_{h,d}(\boldsymbol{I};\boldsymbol{\theta})\right) \big) \end{aligned}, \tag{11}$$

$$\omega^i_d(\boldsymbol{I};\boldsymbol{\theta}) = \begin{cases} \dfrac{\xi}{-\sin\phi^i_d(\boldsymbol{I};\boldsymbol{\theta})} & \text{if} -\dfrac{\pi}{2} \le \phi^i_d(\boldsymbol{I};\boldsymbol{\theta}) < -\dfrac{\pi}{4} \\ \dfrac{\xi}{-\cos\phi^i_d(\boldsymbol{I};\boldsymbol{\theta})} & \text{if} -\dfrac{\pi}{4} \le \phi^i_d(\boldsymbol{I};\boldsymbol{\theta}) < 0 \\ \dfrac{\xi}{\cos\phi^i_d(\boldsymbol{I};\boldsymbol{\theta})} & \text{if } 0 \le \phi^i_d(\boldsymbol{I};\boldsymbol{\theta}) < \dfrac{\pi}{4} \\ \dfrac{\xi}{\sin\phi^i_d(\boldsymbol{I};\boldsymbol{\theta})} & \text{if } \dfrac{\pi}{4} \le \phi^i_d(\boldsymbol{I};\boldsymbol{\theta}) \le \dfrac{\pi}{2} \end{cases}, \tag{12}$$

where $l_{m,d}(\boldsymbol{I};\boldsymbol{\theta})$ for $m \in \left\{u,v,w,h,\sin 2\phi^i, \cos 2\phi^i\right\}$ are the network outputs. In addition, the network outputs a score that indicates "graspable" for each map cell. The transformation is as follows:

$$S^k_d(\boldsymbol{I};\boldsymbol{\theta}) = \frac{\exp\left(l^k_{S,d}(\boldsymbol{I};\boldsymbol{\theta})\right)}{\sum_k \exp\left(l^k_{S,d}(\boldsymbol{I};\boldsymbol{\theta})\right)}, k \in \{0,1\}, \tag{13}$$

where $l^k_{S,d}(\boldsymbol{I};\boldsymbol{\theta})$ is the network output. $S^k_d(\boldsymbol{I};\boldsymbol{\theta})$ is the score, which indicates graspable. k = 0 corresponds to ungraspable and k = 1 corresponds to graspable.

2) Training: To train the network, we need to assign ground-truth $\boldsymbol{g}^i_t$ to specific outputs of $G^*(\boldsymbol{I};\boldsymbol{\theta})$. We calculate the Jaccard overlap between box of $\boldsymbol{g}^i_t$ and box of $\boldsymbol{g}^i_d$ in the same way is proposed in image recognition literature [18], [19]. We assign label "Pos" with Jaccard overlap higher than a threshold (0.5), and assign label "Neg" with the others. After assigned these labels, we optimize the following loss function :

$$\mathcal{L}\left(G, G(\boldsymbol{I};\boldsymbol{\theta})\right) = \frac{1}{N_d}\left(\mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{angle}} + \mathcal{L}_{\text{ability}}\right), \tag{14}$$

where $\mathcal{L}_{\text{loc}}$ is loss for offset, $\mathcal{L}_{\text{angle}}$ is loss for angle, $\mathcal{L}_{\text{ability}}$ is loss for score. There loss term are expressed as follows:

$$\mathcal{L}_{\text{loc}} = \sum_{d \in \text{Pos}, t, m \in m_1} \mathbb{I}^t_d \, \mathcal{L}_{L_1}\left(l_{m,d}(\boldsymbol{I};\boldsymbol{\theta}) - l^t_{m,d} \right), \tag{15}$$

$$\mathcal{L}_{\text{angle}} = \sum_{d \in \text{Pos}, t, m \in m_2} \mathbb{I}^t_d \, \mathcal{L}_{L_1}\left(\tanh\left(l_{m,d}(\boldsymbol{I};\boldsymbol{\theta})\right) - m \right), \tag{16}$$

$$\begin{aligned} \mathcal{L}_{\text{ability}} = & -\sum_{d \in \text{Pos}} \sum_t \mathbb{I}^t_d \log\left(S^1_d(\boldsymbol{I};\boldsymbol{\theta})\right) \\ & -\sum_{d \in \text{Neg}} \log\left(S^0_d(\boldsymbol{I};\boldsymbol{\theta})\right), \end{aligned} \tag{17}$$

where $\mathbb{I}^t_d$ is an indicator for matching the d-th default grasp with the t-th ground-truth grasp. $\mathcal{L}_{L_1}(\cdot)$ indicates *Smooth L1 loss* [18]. $m_1 = \{u,v,w,h\}$, $m_2 = \left\{\sin 2\phi^i, \cos 2\phi^i\right\}$. During Training, most of the default grasp poses are label "Neg." Therefore, we apply *hard negative mining* proposed by Liu et al. [19].

3) Inference: During inference, we select grasps with $S^1_d(\boldsymbol{I};\boldsymbol{\theta})$ higher than 0.8. Transformation ${}^w\boldsymbol{T}_c\left({}^c\boldsymbol{T}_i\left(\boldsymbol{g}^i_d, z^c_d\right)\right)$ requires z^c. In this paper, we use depth value from the depth sensor.

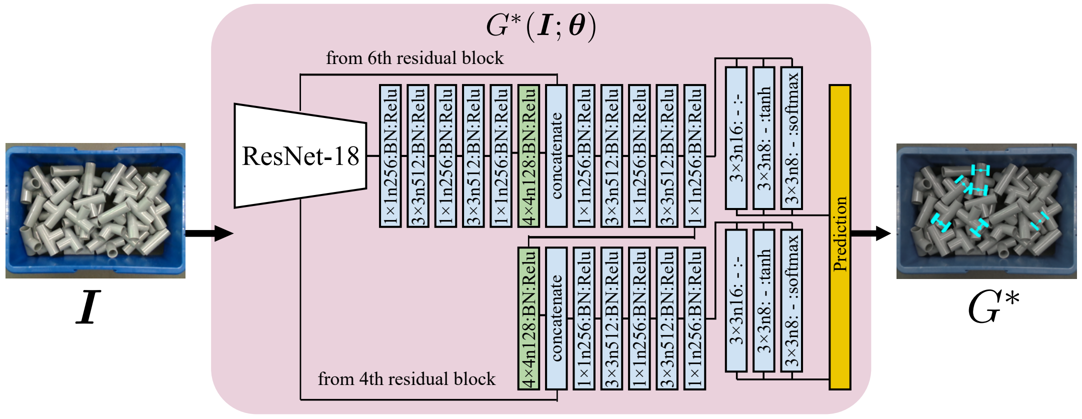


Fig. 5. Architecture of $G^*(\boldsymbol{I};\boldsymbol{\theta})$. The architecture uses the one-shot detection method [6]. An input image $\boldsymbol{I}$ is processed through a Feature Pyramid Networks (FPN)-style architecture [13] to extract multi-scale feature maps. Several convolutional layers and a prediction layer process these feature maps to generate a set of grasps and scores. The ResNet [14] is used as a backbone network. The blue boxes denote a convolution/normalization/activation layer and the green boxes denote a transposed convolution/normalization/activation layer. The prediction layer (orange box) denotes the transformations defined by (9)–(13).

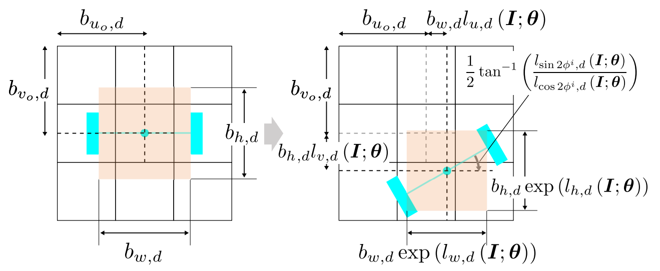

$b_{u_o,d}$
$b_{v_o,d}$
$b_{h,d}$
$b_{w,d}$
$b_{w,d}l_{u,d}(\boldsymbol{I};\boldsymbol{\theta})$
$\frac{1}{2}\tan^{-1}\left(\frac{l_{\sin 2\phi^i,d}(\boldsymbol{I};\boldsymbol{\theta})}{l_{\cos 2\phi^i,d}(\boldsymbol{I};\boldsymbol{\theta})}\right)$
$b_{h,d}l_{v,d}(\boldsymbol{I};\boldsymbol{\theta})$
$b_{h,d}\exp(l_{h,d}(\boldsymbol{I};\boldsymbol{\theta}))$
$b_{w,d}\exp(l_{w,d}(\boldsymbol{I};\boldsymbol{\theta}))$

Fig. 6. Illustration of "default grasp" in the case of 3×3 feature map cells. Sky-blue colored areas and orange-colored areas denote the default grasp and a bounding box surrounding the grasp, respectively. In each cell, the default grasp is represented by an angle and the bounding box which is parameterized by the position $(b_{u_o,d}, b_{v_o,d})$ and size $(b_{w,d}, b_{h,d})$ (left). $G^*(\boldsymbol{I};\boldsymbol{\theta})$ generates the grasp pose by inference these deviations (right).

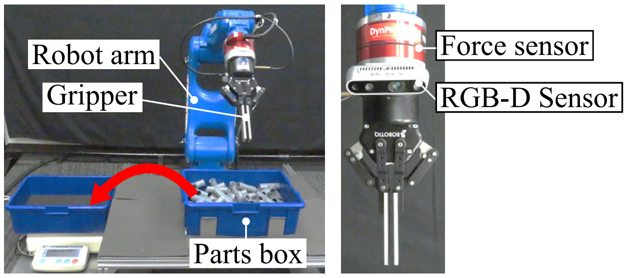


Fig. 7. Experimental setup. RGB-D camera: Intel RealSense D435, Robot arm: YASKAWA GP-7, Gripper: RobotiQ 2F-85 (toe length 100 [mm]), Force sensor: WACOH-TECH WEF-6A1000-30-RCD-B.

IV. Experiments

We evaluated and compared the performances of *Fast Graspability Evaluation* (FGE) (Domae et al. [5]) as a non-learning-based method, and *Dex-Net* (Mahler et al. [9]) as a learning-based method. For Dex-Net, we used the code published by the authors, and for FGE, we reproduced and implemented [5]. The parameter values for both methods were set in the code according to literature.

A. Experimental Settings

The experimental setup is shown in Fig. 7. The robot performed the pick and place tasks. Ten sets of ten consecutive tasks were performed, i.e., a total of 100 grasps were performed for each object. After the ten consecutive tasks had been performed, the state of the piled objects in the parts box was reset (the number of objects in the box was restored). Whether or not the operation was successful was decided automatically according to the total weight of the other box on which the picked objects were placed. In addition, when the force sensor Z-axis value of the world coordinate frame was 5 N or more, the grasp was counted as failure. When a grasp failed, or no grasp inference result was obtained, the experiment was continued by changing the state of objects pose.

Four types of objects with different physical and geometric properties were used to evaluate the performances (Fig. 8). The simulation was built by *Unity*, which is an off-the-shelf game engine. The parameter of the environment in the simulation was set from the actual dimensions of the experimental environment. The polygon mesh generated for each object employs 3-D scan data or CAD data. The number of vertices in each mesh in (a)–(d) was 38,623, 31,660, 32,248, and 3,083, respectively. The number of the domain adaptation training data was five times the number of objects, and that of the training the grasp space was 2,000.

We assumed that the scenes that grasped (b) and (d) had the following restrictions:

- The object (b) with fragile points had a limited range of graspable positions.
- Since the grasp force of the gripper is limited, the center of gravity should be taken into account while grasping the heavy object (d), i.e., graspable positions are limited to a certain distance from the center of gravity of object.

$\tilde{G}^*_{\text{additional}}$ was built to meet these constraints, as shown in Fig. 9.

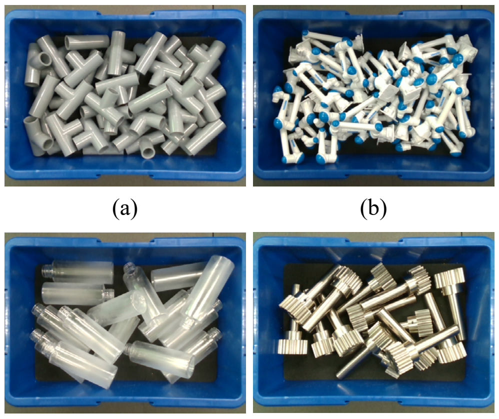


Fig. 8. Four types of objects having different geometric and physical properties: (a) Fixed rigid objects (40 pieces, approx. 30 g); (b) Objects whose geometries change locally (30 pieces, approx. 38 g); (c) Semi-transparent objects (15 pieces, approx. 12 g); (d) Heavy object with a biased center of gravity (15 pieces, approx. 550 g).

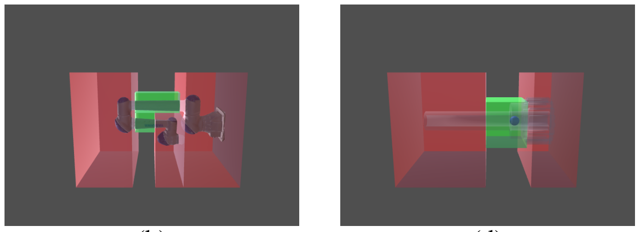


Fig. 9. Setting the graspable area: The green region indicates the graspable area, and the red region indicates the ungraspable area. (b) and (d) correspond to object (b) and object (d) respectively, as shown in Fig. 8.

TABLE I

SUCCESS RATE FOR THE PICK-AND-PLACE TASK

Method	Success Rate (Grasp of Multiple Objects): (a)	(b)	(c)	(d)
FGE [5]	74 (74)	59 (68)	35 (35)	38 (38)
Dex-Net [9]	71 (71)	62 (66)	34 (34)	63 (63)
Ours	**91** (**91**)	**83** (**86**)	**99** (**99**)	**86** (**86**)

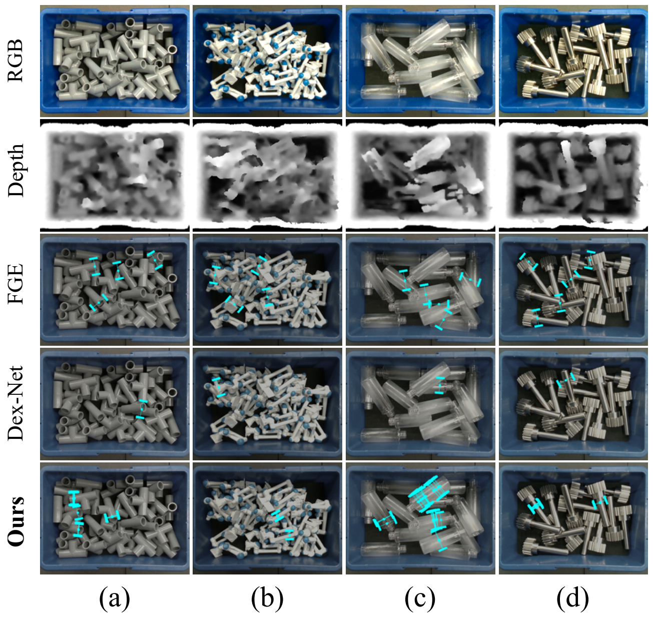


Fig. 10. Grasping pose estimation results for four types of objects by using different methods. (a)–(d) correspond to the objects (a)–(d) shown in Fig. 8. Sky-blue colored areas denote the gripper poses.

B. Experimental Results

1) Performance Comparison of Various Objects: Table I shows the comparative results of the grasp success rate. The *success rate (multiple)* includes cases where multiple objects were picked up at the same time. The proposed method achieved the highest grasp success rate across all objects, and the average grasp success rate was 89.8 % (90.5 %). Fig. 10 shows the results of the grasping positions and orientations in each method. In FGE and Dex-Net, there is a tendency to detect points at high positions where the amount of change in depth is outstanding. Unlike this, in the proposed method, the grasp pose on the image plane that has a high possibility of a successful grasp was obtained as output.

Achieving a high grasp success rate for different geometric and physical properties suggests that the proposed system could be applied to various objects. Furthermore, the cause of failure was different between the proposed method and other methods. As shown in Fig. 10, the comparison method outputs the grasping pose at the edges or corners of the objects. Because of this, grasping often failed due to instability or collisions with other objects. On the other hand, the main factors that caused the proposed method to fail were system errors such as errors in the depth information and hand-object slippage. This result suggests that building $\tilde{G}^*$ in a simulator to be a subspace of the target grasp space G^* is useful for grasping in real-world scenarios. It also suggests that the systematic and accidental errors inherent to the system and the objects affect the grasping performance, and it is necessary to consider them in future works.

Next, we discuss the objects (c) and (d), that have large differences in the grasp success rate. For the semi-transparent object (c), there was a 60 % difference in the success rate between the proposed method and the comparison methods. It is considered that this difference was caused by the missing data of the depth image as shown in Fig. 10. This also shows the superiority of the proposed method using only RGB images as input. In object (d), the grasp success rate of FGE was extremely low, compared to the other two methods. This was caused by the result of the extraction of candidate regions where FGE estimated grasp poses. The effect of this extraction was more critical for heavy the object (d), in which the grasped position affects the success or failure, compared to lightweight objects such as (a) and (b). This may be improved by applying a different instance recognition, but it is necessary to solve additional tasks by sensing in real-world scenarios. On the contrary, learning-based methods that implicitly recognize instances did not show such a tendency among (a), (b), and (d). This suggests that the method is robust against the differences between objects and sensor performances.

2) Effect of Grasp Conditions Customization: In this section, the definition of the grasp success judgment is changed based on the constraints of the scene, and the adaptability of the proposed method to different scenes is evaluated by the performance difference with and without $\tilde{G}^*_{\text{additional}}$. Specifically, if object (b) is grasped of the specified region, it is considered as a failure. Also, if object (d) is grasped and its pose is changed after being lifted, even if it is successfully transported as shown in Fig. 11, the task is considered as a failure.

Fig. 12 shows a comparison of the grasping pose estimation results for object (c) and object (d), by using $\tilde{G}^*_{\text{base}}$ and $\tilde{G}^*_{\text{base}} \cap \tilde{G}^*_{\text{additional}}$. As a result of adding $\tilde{G}^*_{\text{additional}}$, the grasping pose was changed to satisfy the constraints defined in Section IV-A. Table II shows the comparison results of the grasp success rate (the value in parentheses indicates the percentage evaluated by the criteria of IV-B.1). When $\tilde{G}^*_{\text{additional}}$ was added, the grasp success rate was improved, compared to when only $\tilde{G}^*_{\text{base}}$ was applied. In addition, compared with other methods, the case where $\tilde{G}^*_{\text{additional}}$ was applied achieved the highest grasp success rate, and the average grasp success rate was 81.5 % (84.5%). These results indicate that the proposed method coped with the change of constraints in the real world by only modifying how to generate the grasp space in the simulation environment.

In contrast, the other methods showed comparatively low grasping success rates, particularly in the case of FGE. Since FGE is a non-learning-based method, it does not have enough ability to adapt the new scenes. Learning-based methods that automatically extract the feature amount show the capability of adapting to the change of application scenes.

C. Discussion

The experimental results show that the proposed system enables the grasping of objects with different physical properties, and show that the grasp planned or re-planned in the simulator is transferred appropriately to the real-world robotic system. The wide-object-coverage and ease of changing grasp motions according to their constraint are valuable for bin-picking in industrial settings. However, this study has some limitations.

Another limitation is its customizability. We proposed a customizable bin-picking system; this means we can plan or re-plan a real-world grasp in the simulator, which is safer, cheaper, and provides more information than real-world situations can. On the other hand, further discussion on how to customize a successful grasp space is necessary. In this study, the semi-suitable grasp space is built by considering force closure proposed in [15], [16]. Several other force closure conditions have been proposed, such as studies that consider the shape uncertainty [17] and the contact surface flexibility [21]. Applying these more robust force closure conditions to build a grasp space in the simulation could improve the real-world grasp performance. However, it should be noted that a successful grasp in the simulation is only a necessary condition for a successful grasp in the real-world environment, and the reality gap still remains to be filled in the motion space.

One such limitation is the covariate shift assumption. In the physical robot experiments, that the grasp appeared to succeed on RGB Image; however, it sometimes failed, which indicates that there is a "reality gap" between the simulation and the real world in motion space. Improving the simulation performance to fill this reality gap is expensive.

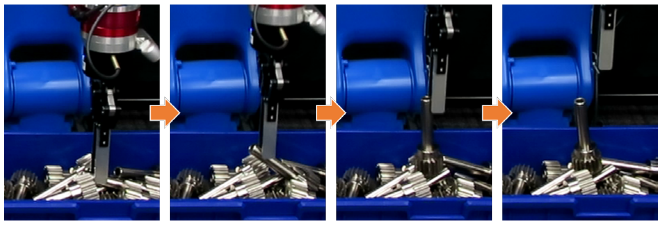
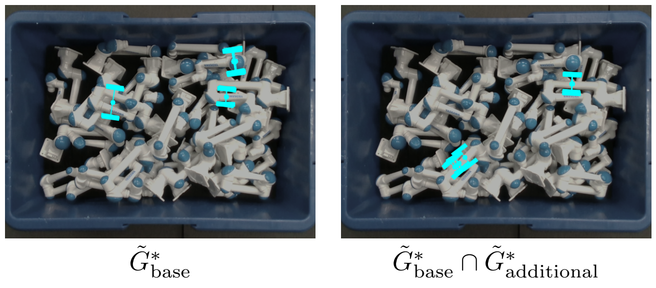
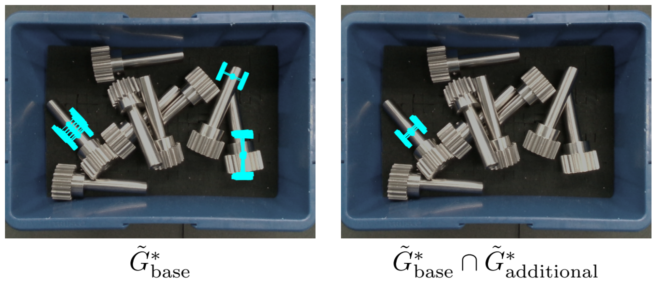


Fig. 11. Failure to grasp heavy object with a biased center of gravity. The grasp was far from the center of gravity, and the grasping force was insufficient. Hence, the object slipped and dropped.


$\tilde{G}^*_{\text{base}}$
$\tilde{G}^*_{\text{base}} \cap \tilde{G}^*_{\text{additional}}$

(b)


$\tilde{G}^*_{\text{base}}$
$\tilde{G}^*_{\text{base}} \cap \tilde{G}^*_{\text{additional}}$

(d)

Fig. 12. Comparison of grasping pose estimation results for object (b) and object (d) by using $\tilde{G}^*_{\text{base}}$ and $\tilde{G}^*_{\text{base}} \cap \tilde{G}^*_{\text{additional}}$. Sky-blue colored areas indicate the gripper poses.

TABLE II
SUCCESS RATE FOR THE PICK-AND-PLACE TASK WITH THE ADDITIONAL CONSTRAINT OF GRASP QUALITY

Method	Successful Rate (Include worse grasp):	
	(b)	(d)
FGE [5]	15 (59)	24 (38)
Dex-Net [9]	8 (62)	34 (63)
$\tilde{G}^*_{\text{base}}$	67 (78)	43 (81)
$\tilde{G}^*_{\text{base}} \cap \tilde{G}^*_{\text{additional}}$	**83** (**83**)	**80** (**86**)

Thus, it is necessary and more practical to learn more successful grasp motions in real-world grasp trials via a model that has been previously trained in the simulator. This problem has been recently defined as "micro-data learning" by Chatzilygeroudis et al. [20].

Finally, we have to consider the cost of building the simulation environment. Our system needs the collection of real-world RGB images, which is a tedious task. Domain Randomization [11], [22] could be an attractive alternative for reducing the collection of real-world data. However, unplanned domain randomization can be a more time-consuming task than domain adaptation via real-world data collection.

V. Conclusions

In this paper, we proposed a novel robotic bin-picking system that has the potential of being applied to various objects and application scenes in industries. Directly inferring the grasp space from raw sensor data with deep neural networks, explicit modeling of the object was eliminated, which expands the range of objects to which the system can be applied. In addition, to reduce the influence of the difference between objects on the sensor performance, an RGB image was taken as the input of the network. Furthermore, we proposed a method of analytically generating a training dataset of successful grasp spaces in a simulator. Combining these and minimizing the discrepancy between simulations and real-world scenarios, a successful grasp space built in the simulator was realized in the real-world. The proposed method achieved an average grasp success rate of about 89.8 % for four objects with different geometrical and physical properties in real-world grasp evaluation. The method was also able to cope with additional constraints in the real-world, i.e., the restriction of the graspable area, by only modifying the grasp space that satisfied these constraints in the simulation environment (the average success rate was 81.5 %). The wide coverage of systems as shown in these results is very useful in industrial applications, where cost-effectiveness is a desired quality. In a future work, we wish to extend our method to take into account errors inherent in systems and objects that were not considered in this study, and thereby improve the performance.

Acknowledgment

We would like to thank Prof. K. Harada of Osaka University for helpful discussions on robot grasp research.

References

[1] A.T.Miller, S.Knoop, H.I.Christensen, and P.K.Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, Sept. 2003, pp. 1824-1829.

[2] K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, and Y.Kawai, "Probabilistic approach for object bin-picking approximated by cylinders," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2013, pp. 3742-3747.

[3] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, May 1992, pp. 2290-2295.

[4] A. T. Miller and P. K. Allen, "Graspit: A versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[5] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin-picking with general grippers," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2014, pp.1997–2004.

[6] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2015, pp. 1316–1322.

[7] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2016, pp. 3406–3413.

[8] 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," in *Proc. Robotics: Science and Systems (RSS)*, July 2017.

[9] J. Mahler and K. Goldberg, "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences," in *Proc. Conf. on Robot Learning (CoRL)*, Nov. 2017, pp. 515–524.

[10] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V.Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, May 2018, pp. 4243–4250.

[11] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan,J.Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efcient robotic grasping via randomized-to-canonical adaptation networks," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 12619-12629.

[12] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int'l Conf. on Computer Vision (ICCV)*, Oct. 2017, pp. 2242–2251.

[13] T. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 936-944.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Dec. 2016, pp. 770-778.

[15] I-Ming Chen and J. Burdick, "Finding antipodal point grasps on irregularly shaped objects," *IEEE Trans. Robotics Automat.*, vol. 9, no. 4, pp. 507–512, Aug. 1993.

[16] V. Nguyen, "Constructing force-closure grasps," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Apr. 1986, pp. 13681373.

[17] J. Mahler, S. Patil, B. Kehoe, J. Van Den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *Proc. IEEE Int ' l Conf. Robotics and Automation (ICRA)*, June 2015, pp. 4919-4926.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A.C. Berg, "SSD: Single shot multiBox detector," in *Proc. 14th European Conference on Computer Vision (ECCV)*, 2016, pp. 21-37.

[20] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon and J. Mouret, "A survey on policy search algorithms for learning robot controllers in a handful of trials," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 328–347, Apr. 2020.

[21] K. Harada, T. Tsuji, K. Nagata, N. Yamanobe, K. Maruyama, A. Nakamura, and Y. Kawai, "Grasp planning for parallel grippers with flexibility on its grasping surface," in *Proc. IEEE Int'l Conf. on Robotics and Biomimetics (ROBIO)*, Dec. 2011, pp. 1540–1546.

[22] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, Sept. 2017, pp. 23–30.