2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

CMG-Net: An End-to-End Contact-based Multi-Finger Dexterous Grasping Network

Mingze Wei[1,*], Yaomin Huang[2,*], Zhiyuan Xu[3], Ning Liu[3], Zhengping Che[3], Xinyu Zhang[1], Chaomin Shen[2], Feifei Feng[3], Chun Shan[4], and Jian Tang[3]

***Abstract*— In this paper, we propose a novel representation for grasping using contacts between multi-finger robotic hands and objects to be manipulated. This representation significantly reduces the prediction dimensions and accelerates the learning process. We present an effective end-to-end network, CMG-Net, for grasping unknown objects in a cluttered environment by efficiently predicting multi-finger grasp poses and hand configurations from a single-shot point cloud. Moreover, we create a synthetic grasp dataset that consists of five thousand cluttered scenes, 80 object categories, and 20 million annotations. We perform a comprehensive empirical study and demonstrate the effectiveness of our grasping representation and CMG-Net. Our work significantly outperforms the state-of-the-art for three-finger robotic hands. We also demonstrate that the model trained using synthetic data perform very well for real robots.**

I. Introduction

Grasping unknown objects from a cluttered environment is a fundamental problem for autonomous robotic manipulation, arising in a wide range of applications such as industrial automation (e.g., bin-picking, quality inspection, and warehouse automation), commercial places (e.g., book picking or placing in a library and healthcare) and household (e.g., folding laundry, playing billiards, and fetching a beer in a refrigerator). Despite the exciting progress in pose prediction and object manipulation for parallel-jaw grippers [1]–[6], robotic grasping for multi-finger hands with high DoFs remains challenging. Parallel-jaw grippers have relatively low complexity in manipulation due to their one DoF structure. However, their structures also limit their deployment in more complicated scenarios with relatively big, irregular, or spherical objects.

Unlike parallel-jaw grippers, multi-finger dexterous hands significantly improve adaptability for object shapes due to higher DoFs and flexibility. However, the difficulties in finding valid hand configurations and grasp poses for multi-finger robotic hands are greatly increased due to high-dimensional search space and discontinuous grasp space [7]–[9].

[1]School of Software Engineering, East China Normal University, China. 51205902058@stu.ecnu.edu.cn, xyzhang@sei.ecnu.edu.cn

[2]School of Computer Science, East China Normal University, China. 51205901049@stu.ecnu.edu.cn, cmshen@cs.ecnu.edu.cn

[3]Midea Group, China. {xuzy70, liuning22, chezp, feifei.feng, tangjian22}@midea.com

[4]School of Electronics and Information, Guangdong Polytechnic Normal University, China. shanchun@gpnu.edu.cn

*The first two authors contributed equally. This work was done when Mingze Wei and Yaomin Huang took internships at Midea Group.

Corresponding authors: Xinyu Zhang and Jian Tang.

To handle these issues, previous work for multi-finger grasping can be divided into two categories. *Traditional analytical methods* [7], [9]–[12] explore the potential grasping space through stochastic search and sampling. These algorithms are often computationally expensive, typically requiring tens or even hundreds of iterations per object. Moreover, they heavily rely on precise object representation, thus not applicable to unknown objects in a cluttered environment. *Data-driven approaches* have attracted attention in recent years. Lu *et al.* proposed a grasp planner and grasp evaluator based on an efficient deep neural network [13]. This work still assumes the object models are known. Lundell *et al.* proposed a coarse-to-fine model to predict grasps [14]. This work targets a single object instead of many objects in a cluttered environment. Li *et al.* presented an end-to-end network [15] that can predict grasp poses and configurations. It simply discretizes and classifies multi-finger hand configurations manually.

In this paper, we propose a novel approach to grasp unknown objects in a cluttered environment. Our contribution includes

- A new grasp representation that projects 10-DoFs grasps to only 6-DoFs based on contact points. This significantly reduces the potential grasping search space, facilitates the process of learning and improves the grasping quality.
- Based on this novel representation, we propose an end-to-end deep neural network, CMG-Net, which outputs multi-finger hand configurations and grasp poses for an input single-shot viewpoint in a cluttered scene.
- To demonstrate the benefits using CMG-Net, we build a synthetic grasping dataset, consisting of 5000 cluttered scenes with 80 object classes and 20 million grasp annotations (i.e., hand poses and hand configurations).
- Our experimental results show that CMG-Net outperforms the state-of-the-art in terms of both grasping success rate and grasping quality. Moreover, we also demonstrate that CMG-Net works very well for robotic grasping in a real-world environment.

II. Related Work

Grasping is a fundamental problem for robotic manipulation and has been extensively studied. Most work focuses on parallel-jaw grippers [1], [2], [4]–[6] due to their simplicity, low DoFs, and computational efficiency. However, parallel-jaw grippers are less efficient and less reliable

for manipulating arbitrary-shaped objects. To achieve user-friendly interaction, multi-finger robotic hands and dexterous grasping remain a hot research topic in the field of robotic manipulation [16]. This research can be briefly divided into two categories: the traditional analytical sampling-based method and the data-driven method.

Traditional analytical sampling-based methods [7], [9]–[12] sampled various grasp candidates and evaluated them based on certain metrics considering the physical properties of objects such as wrench space [17]. In general, both the object model and environment are assumed to be known in advance [18]. Eigengrasp [7] reduced the dimensions of grasp search space by performing principal component analysis (PCA) on grasping pose and configuration data. Although the reduction increases the efficiency of generating grasps, the search space of the random sampling process for grasps is still very huge. As a result, these sampling-based methods are less efficient in practical use.

Data-driven methods fall into one of two primary types. The one is an extension of the traditional sampling-based method [17], [19]. Instead of computing physical metrics, this method directly estimates grasp quality metrics from trained deep models. The grasp success rate can be greatly improved since traditional metrics cannot be computed accurately from an incomplete view of a novel object without any contact feedback. However, they are still dependent on known object models and exhibit the problem of huge sampling and search space. The other data-driven method is performed in an end-to-end manner [8], [9], [19]–[23]. Specifically, this method takes the image or point cloud data of a grasped object as input and outputs a high-quality grasp. These approaches are able to effectively generate grasps and are robust to unknown objects. However, many can only handle a single object. Grasping may often fail due to the potential collision between the gripper and the environment. Some recent work [14], [15], [24] predicts collision-free 6-DoF grasping in clutter using multi-finger grippers. They only classify the grasp types and do not take into account of the properties of multi-finger grasps. Our approach considers the gripper's physical structure and does not rely on the grasp types. Using a novel grasping representation and an end-to-end deep neural network based on contacts, our approach significantly reduces the search space for grasping and can generate reliable grasp poses.

III. Overview

A. Problem Statement

We address the problem of grasp generation for unknown objects in a cluttered environment with a multi-finger robotic hand. We take a single-shot of point cloud $\mathbf{PC}$ from a given viewpoint as input and predict a set of high-quality grasps $\mathbf{G} = \{\mathbf{p}, \mathbf{q}\}$ where $\mathbf{p}$ is hand pose and $\mathbf{q}$ is hand joint configuration.

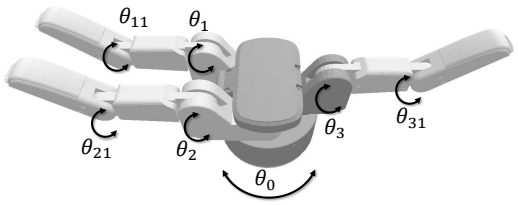


Fig. 1. A three-finger robotic hand.

In this paper, we use a *BarrettHand* as our multi-finger dexterous hand, as shown in Fig.1. The hand's pose consists of translation and orientation, denoted by $\mathbf{p} = \{x, y, z, \phi, \gamma, \varphi\}$. A hand joint configuration $\mathbf{q}$ consists of four inner joint angles, denoted by $\mathbf{q} = \{\theta_0, \theta_1, \theta_2, \theta_3\}$. Here, θ_0 is the spread joint angle. θ_1, θ_2 and θ_3 are the inner joint angles. The four outer joint angles θ_{11}, θ_{21} and θ_{31} are dependent on their individual inner joints.

B. Our Approach: Overview

Fig. 2 illustrates the overview of our approach. For a cluttered environment, we aim to grasp an unknown object using a multi-finger robotic hand. First, we obtain a point cloud captured from any viewpoint. Second, an end-to-end network, CMG-Net, is trained using our synthetic grasp dataset for the three-finger robotic hand based on contact representation. Third, we use the trained CMG-Net to obtain the final hand pose and grasp configuration. Finally, we execute the grasping task in real-world scenarios.

IV. Grasp Representation

Due to the ambiguous and discontinuous distribution of multi-finger grasps, it is difficult to directly regress a 6-DoF hand pose and a 4-DoF hand configuration in such a high dimensional space [4], [14]. Therefore, we define a novel grasp representation that can generate a constrained grasp space for our learning-based data-driven method.

Multi-finger grasp representation: For human hand grasp, Cutkosky [25] suggested a taxonomy of two main categories: power and precision. We observe that power grasp is not friendly to grasp objects on the desktop. In many scenarios, it requires the palm and fingers to entirely enclose an object, but this can easily cause collisions with the table during the hand closing action. In a real-world scenario, this may cause damage to expensive robotic hands. Therefore, we consider the precision grasp when executing a grasping task. This requires each finger to have contact with the object and but the contact between the palm, and the object is unnecessary. For a dexterous hand like BarrettHand, whose two finger joints are coupled by a single motor, the contact points with the object for the precision grasp are usually at the end of fingertips.

To describe the contact between the fingertips and the object, we fit a fingertip end with a circle (Fig. 3-(a)). The circle center at each fingertip (i.e., fingertip center) and its radius can be computed by the given gripper model. We denote this circle as the *fingertip circle*. The local vector from the fingertip center to the origin of the outer joint is denoted

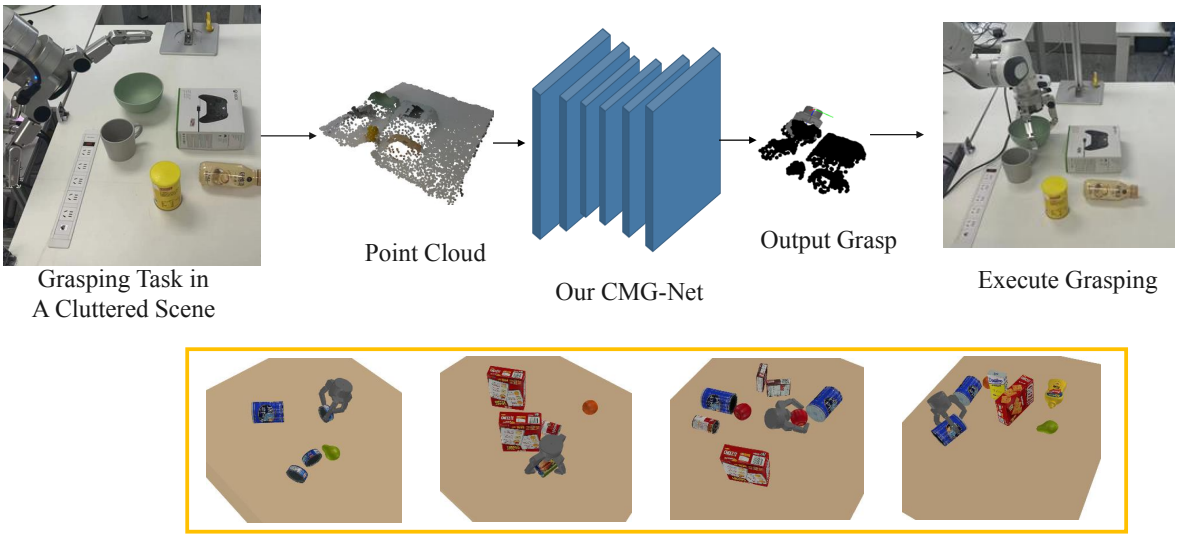


Fig. 2. The overview of our approach.

as the fingertip vector $\mathbf{v}_{finger}$. Based on this denotation, we generate a novel multi-finger grasping representation from the contact point to the corresponding finger joint pose to the hand pose and finally to the hand configuration. We assume that the fingertip model of the dexterous hand in the side view can be matched with a particular circle.

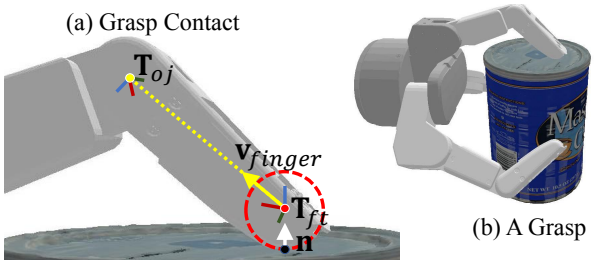


Fig. 3. Grasp representation based on contacts. (a) A contact between a fingertip and an object to be grasped. The black dot represents the contact point. The white arrow indicates the surface normal vector at the contact. The red circle highlights fingertip circle, the red point is the fingertip center and a coordinate frame $\mathbf{T}_{ft}$ is attached to the fingertip. The yellow arrow indicates the fingertip vector $\mathbf{v}_{finger}$. The yellow point is outer-joint position and a coordinate frame $\mathbf{T}_{oj}$ is attached. (b) A grasp example for a three-finger robotic hand.

Given a suitable contact point predicted on the object surface, the fingertip center $\mathbf{t}_{ft}$ in the world coordinate frame is computed as

$$\mathbf{t}_{ft} = \mathbf{t}_{contact} + r\mathbf{n} \tag{1}$$

where $\mathbf{t}_{contact}$ is the position of the contact point, $\mathbf{n}$ is its surface normal vector and r is the circle radius. To compute the hand pose from the fingertip center, we attach a coordinate frame at the fingertip center. At the contact point, The z-axis is the same as the normal vector. To avoid confusion, we denote this vector as $\mathbf{v} = [v_1, v_2, v_3]^T$. Then the rotation matrix $\mathbf{R}_{ft}$ can be calculated by

$$\mathbf{R}_{ft} = \left[\mathbf{R}^1, [0, -v_3, v_2]^T, \mathbf{v}\right], \tag{2}$$

where $\mathbf{R}^1 = [0, -v_3, v_2]^T \times \mathbf{v}$. Then the hand pose $\mathbf{T}_{ft}$ is represented by the rotation matrix $\mathbf{R}_{ft}$ and the position of fingertip center $\mathbf{t}_{ft}$.

Then we compute a fixed-length vector from the fingertip center to the outer-joint position ($\mathbf{v}_{finger}$), by finger projections $x \in [-1, 1]$ and $y \in [-1, 1]$. x and y are the projections of the fingertip vector onto the x axis and y axis respectively in the fingertip center coordinate frame. The outer-joint position $\mathbf{t}_{oj}$ can be computed

$$\mathbf{t}_{oj} = \|\mathbf{v}_{finger}\| \left(\mathbf{R}_{ft} \cdot [x, y, z]^T\right) + \mathbf{t}_{ft} \tag{3}$$

where $z = \sqrt{1 - x^2 - y^2}$. Please refer to Fig.3-(a) for illustration. Here, z axis of the outer-joint coordinate frame is orthogonal to the fingertip vector $\mathbf{v}_{finger}$ and the z axis $\mathbf{v}_z$ of the fingertip center coordinate frame. Then we have the rotation matrix $\mathbf{R}_{oj}$ of the outer-joint coordinate frame

$$\mathbf{R}_{oj} = \left[\mathbf{v}_{finger}, \mathbf{R}^2, \mathbf{v}_{finger} \times \mathbf{v}_z\right] \cdot \mathbf{R}_0 \tag{4}$$

where $\mathbf{R}^2 = \mathbf{v}_{finger} \times [\mathbf{v}_{finger} \times \mathbf{v}_z]$. $\mathbf{R}_0$ is a transformation matrix with a fixed angle of rotation around the z axis.

To further obtain the gripper pose from the outer-joint coordinate frame, we divided the four DoFs of the Barrett hand into two main joints(θ_{ms}, θ_m) and two supporting joints(θ_{s1}, θ_{s2}) according to whether they are involved in the calculation of the grasp pose or not. For example, if the contact as shown in Fig.3-(b) corresponds to finger 3, the main joint θ_m is the joint θ_3 and θ_{ms} is the spread joint θ_0. Then the supporting joints θ_{s1} and θ_{s2} are θ_1 for finger 1 and θ_2 for finger 2, respectively. Therefore, the grasp pose

is computed as

$$\mathbf{T}_{pose} = \mathbf{T}_{oj} \cdot (\mathbf{T})^{-1} \tag{5}$$

where $\mathbf{T}_{oj}$ is composed by $\mathbf{R}_{oj}$ and $\mathbf{t}_{oj}$. $\mathbf{T}$ is the transformation matrix from the base coordinate frame of the gripper to the out-joint coordinate frame. $\mathbf{T}$ can be computed from the main joints(θ_{ms}, θ_m) using forward kinematics. Finally, we can combine the supporting joints to obtain the final grasp configuration. As a result, 10 dimensional grasp space is reduced to 6 dimensions, represented by

$$\mathbf{G} = \{x, y, \theta_{ms}, \theta_m, \theta_{s1}, \theta_{s2}\} \tag{6}$$

Using such a grasp representation, we can significantly reduce the grasp space to be searched, which greatly facilitates the learning process by analyzing the characteristics of the multi-finger hand structure. Object shape and hand structure are connected through contacts. This allow more reasonable predictions.

V. Dataset Generation

In this section, we explain the steps to generate our grasp dataset used in training. For a given three-finger robotic hand, we first generate high-quality grasps for each object using *GraspIt!* [18] and keep those grasps that satisfy the grasp quality requirements. We obtain a grasp dataset consisting of contact information for each object. Then we select a number of objects, randomly place them on the table, and examine all the grasps in a simulator. Those collision-free grasps and valid grasp poses remain. Finally, we place a camera in the scene and obtain a single-shot point cloud from a viewpoint. We map the grasp pose from the world coordinate system to the camera coordinate system.

A. Single Object Grasp Dataset With Contacts

To improve the diversity in shape, texture, and size, we select 80 objects from existing datasets [1], [26]–[28]. Then we generate our grasp dataset for these objects in two steps.

First, we generate grasp poses and configurations for a single object. To uniformly sample the points on a single object for grasping, We down-sample the object mesh models to achieve a uniform distribution of sampling points with their normal in voxel space. For a sample, grasp candidates are searched in three dimensions $S_1 \times S_2 \times S_3$, where S_1 is the gripper depths, S_2 is the in-place rotation angle, and S_3 is the angle of spread joint. Given a set of $S_1 \times S_2 \times S_3$, we let fingers close in until they touch the object. We compute its ϵ-*quality* [29] and save its grasp pose and configuration as a grasp annotation if the ϵ-*quality* is greater than a specified threshold. In our dataset, we generate 15000 high-quality grasp annotations for each object.

Second, we generate the contacts between the robotic hand and the objects for those high-quality grasps. Here, we ignore a grasp if its fingertips have no contact with the given object. We use the k-means algorithm to compute the clustering of the contact points and thus generate the corresponding contact point for each fingertip.

B. Scene Grasp Dataset Generation

To generate grasps in a scene from an arbitrary viewpoint, we first take a number of objects randomly from the object dataset and place them on the table in a stable pose. Second, for each object in the scene, we retrieve grasps from the grasp dataset and perform grasping tasks in the simulation environment. We filter out those grasps exhibiting collisions and invalid poses. Third, we place a camera to capture the scene from any viewpoint. The captured point cloud is used to obtain the grasp poses $\mathbf{G}$ and contact positions in the camera coordinate system.

VI. Grasp Network: CMG-Net

In this section, we introduce a simple grasp pose estimation network: CMG-Net (Contact-based Multi-finger Grasping Network). For an input point cloud, we aim to generate a set of grasp poses. Fig. 4 shows the CMG-Net's structure, consisting of three stages, 1) grasp points segmentation, which extracts features from the point cloud by PointNet++ [30], and then classifies the contact points to determine whether they are graspable. And we predict the finger ID (e.g., finger 1, 2, 3) and finger projection corresponding to the possible contact points. 2) preliminary grasp pose prediction, which predicts an initial pose using our proposed grasp representation, and 3) grasp pose refinement, which refines the grasping pose by predicting the supporting joints.

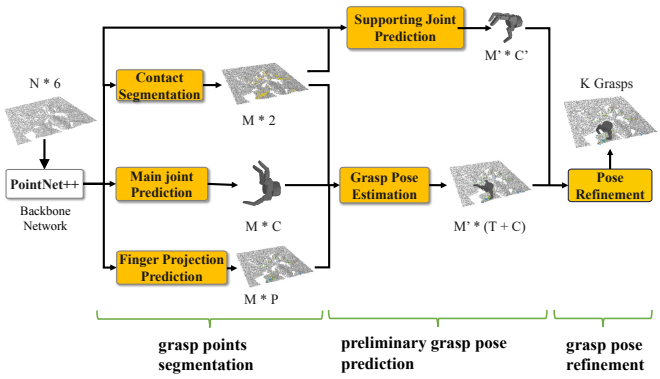


Fig. 4. The CMG-Net framework.

A. Learning to predict grasp points

1) Predicting contact points: We use PointNet++ [30] as our backbone due to its simplicity and high success rate on various challenging tasks. We use its set-abstraction (SA) layers and feature propagation (FP) layers to build an asymmetric U-shaped network with an encoder and a decoder. The encoder uses multiple SA blocks to extract abstract features from point clouds and the decoder utilizes an equal number of FP blocks to gradually interpolate the abstracted features.

We also incorporate the point normals into the network.The actual input data size is increased to $N \times 6$, where N is the number of points. The extracted point cloud

features, as shown in Fig. 4, are used by the *contact segmentation module* to predict whether a point is graspable. We design a cross-entropy loss $\mathcal{F}_{cls}$ for classification training:

$$\mathcal{L}_{gp} = \mathcal{F}_{cls}\left(c^p, c^g\right), \tag{7}$$

where c^p is the prediction result of segmentation head for each grasp sample and c^g is its corresponding label.

B. Grasp Pose Estimation

We further predict the corresponding joint angles of the given dexterous hand based on the graspable points and combine them with finger projection to calculate a preliminary grasp pose using Eq. 5. Specifically, We use *Finger Projection Prediction*, which contains a multi-layer perceptron (MLPs) network with fully connected layers, ReLU, and batch normalization, to predict finger projection x^p and y^p. The finger projection x and y will be predicted as a group xy^p. The loss function is

$$\mathcal{L}_{fp} = \frac{1}{N_{\mathrm{c}}} \sum_i \left\| xy_i^p - xy_i^l \right\| \delta_{\mathrm{c}}, \tag{8}$$

where N_{c} is the total number of contact points. xy_i^l is the ground truth finger projection from the contact points, δ_c is an indicator, and it yields 1 if xy_i corresponds to contact points and 0 otherwise. The joint angle θ_m is evenly divided into n_m bins. In our implementation, we set the subdivision angle $\phi_m = \frac{7\pi}{9}$. For each grasp sample, we calculate its bin classification label bin_m^l and residual label res_m^l as follows

$$\begin{aligned} bin_m^l &= \left\lfloor \frac{\theta_m}{\phi_m} \right\rfloor \\ res_m^l &= \frac{1}{\phi_m}\left(\theta_m^l - \left(bin_m \phi_m + \frac{\phi_m}{2}\right)\right). \end{aligned} \tag{9}$$

The loss of main joint is formulated as

$$\mathcal{L}_m = \mathcal{F}_{cls}\left(bin_m^p, bin_m^l\right) + \mathcal{F}_{res}\left(res_m^p, res_m^l\right), \tag{10}$$

where bin_m^l and res_m^l are ground-truth bin assignment and residual for a given grasp p. bin_m^p and res_m^p are their corresponding predicted values.

Analogically, the spread joints θ_{ms} has the same form

$$\mathcal{L}_{ms} = \mathcal{F}_{cls}\left(bin_{ms}^p, bin_{ms}^l\right) + \mathcal{F}_{res}\left(res_{ms}^p, res_{ms}^l\right). \tag{11}$$

According to Eq. 5, we finally obtain a preliminary grasp pose $\mathbf{G} = \{x, y, \theta_m, \theta_{ms}\}$.

C. Grasp pose refinement

To improve grasp quality and create realistic grasp poses, a joint prediction layer is used to guarantee the accuracy of supporting joints θ_{s1} and θ_{s2}. More specifically, θ_{s1} and θ_{s2} are uniformly divided into n_{s1} and n_{s2} bins, respectively. For each point p_i, we compute the bin classification label and the residual label as follows

$$\mathcal{L}_{s_i} = \mathcal{F}_{cls}\left(bin_{s_i}^p, bin_{s_i}^l\right) + \mathcal{F}_{res}\left(res_{s_i}^p, res_{s_i}^l\right), \tag{12}$$

where $i \in \{1, 2\}$ represents one supporting joint. bin_{s_i} and res_{s_i} are classification and residual labels, respectively.

D. Total loss

The total loss function for training consists of three major components: grasp point prediction $\mathcal{L}_{gp}$, preliminary grasp generation using finger projection $\mathcal{L}_{fp}$, and grasp refinement using joint prediction $\mathcal{L}_{joint}$, which can be formally expressed as,

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{gp} + \beta \mathcal{L}_{fp} + \gamma \mathcal{L}_{joint}, \tag{13}$$

where

$$\mathcal{L}_{joint} = \gamma_1 \mathcal{L}_m + \gamma_2 \mathcal{L}_{ms} + \gamma_3 \mathcal{L}_{s_i}. \tag{14}$$

In our implementation, we set $\alpha = 1$, $\beta = \gamma = 5$, $\gamma_1 = \gamma_2 = \gamma_3 = 1$, and use the cross-entropy loss [31] and Huber (smooth-L1) loss [32] for all classification and regression tasks, respectively.

VII. Experiments and Results

In this section, we first describe the experimental setup, including the dataset usage, evaluation metric, and implementation details. Then we demonstrate our experimental results in simulation and real scenarios. In our experiments and comparison, we observe that our approach can generate dense and robust grasps with high success and completion rates compared to the baseline approaches.

A. Dataset Usage

Our dataset has more than 80 categories distributed among 5000 different scenarios. The network outputs a point-wise grasp pose and hand configurations for a single viewpoint cloud. We consider a grasp attempt is performed successful if the object can lift at least $30cm$ high.

B. Simulation Experiments

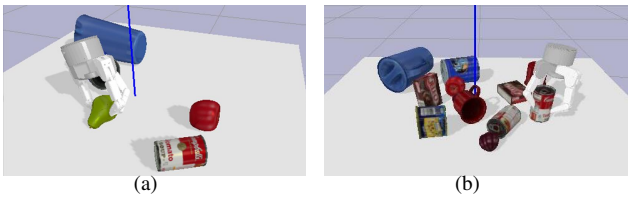
We implement our CMG-Net using Pytorch [33] on NVIDIA GPUs. Our end-to-end network is optimized using the Adam optimizer with a batch size of 32 and a learning rate of 0.004. Our input is a point cloud converted from the depth map captured using a depth camera. We randomly down-sample the point cloud to retain a reasonable number of points (e.g., 20,000).

TABLE I
Comparison in Simulation

	SR(%)	CR(%)	Quality
GraspIt!	48	51	0.63
Multi-FinGan	56	64	0.76
Ours	76	81	0.86

1) Evaluation Metrics: To demonstrate the performance, we introduce three metrics [14], [24], [34]: *Grasp Success Rate (SR)* is the rate of the number of successful grasps to the number of total attempts. *Grasp Completion Rate (CR)* is the rate of the number of objects grasped to the total number of objects after 1.5 times the number of grasp attempts. *Grasps Quality* often refers to ϵ*-quality* metric [29], representing the radius of the largest 6D ball centered on the origin that can be surrounded by the convex hull of the wrench space [35].

2) Results: The experimental results (Table I) show that our network can be well incorporated with the proposed representation. In comparison, our approach shows significant improvement against $GraspIt!$ and Multi-FinGan in terms of success rate, completion rate, and grasp quality. Fig. 5 shows a few simulation experiment results, demonstrating our method can handle small objects very well and generate high-quality grasps for complex scenes.



(a) (b)

Fig. 5. (a) grasp pose for small object; (b) grasp pose in cluttered scenes.

3) Ablation Studies: To improve the quality of handling small objects, we combine the SA and FP layers in PointNet++ through a U-shape network for feature extraction. In addition, we reduce the required regression dimensions by predicting the finger projection, which significantly improves the prediction quality of grasps and reduces the training cost. Table II shows the enhancements resulting from each of the two modules. From the results, we can see that our proposed finger projection can achieve significant improvement.

TABLE II
ABLATION STUDY

	U-shaped	Finger-Pro	SR(%)	Quality
baseline			39	0.56
+U-shaped	✓		46	0.65
+Finger-Pro		✓	72	0.80
Ours	✓	✓	76	0.86

C. Real-world Experiments

We demonstrate that the network trained from the simulation data works well in a real environment. We use a *Franka Emika Panda* equipped with a three-finger dexterous hand in our real-world experiment. We use an *Intel RealSense D435* camera to capture the depth images. In order to remove the background and desk points for grasping prediction, we perform depth image segmentation using a segmentation network [36]. Then the object point clouds are fed into our CMG-Net to obtain the final grasp. For each input, we keep the top 20 best grasps tried in the real field scene. A grasp is successful when an object can be picked up and stably moved to a specified location. Fig. 6 shows a snapshot of the real-world experiment environment and the objects grasped during the experiment. There are three experimental scenarios with 3, 6, and 9 objects to be grasped, respectively. We obtain the average of 6 trials for each scenario. In these real scenarios, the objects are all novel for robotic hand.

The experimental result is shown in Table III. We obtained a success rate up to 74.4% and a completion rate up to 86.1% in the 6-object scenario. The results look similar even for the 9-object scenario. Our work demonstrates a good adaptability for a cluttered environment with many unknown objects.

Fig. 6. A real-world experimental environment for object manipulation using a three-finger robotic hand and the given objects.

TABLE III
REAL HARDWARE EXPERIMENT RESULTS

objects	3	6	9
SR(%)	76.5	74.4	72.0
CR(%)	88.9	86.1	83.3

VIII. CONCLUSION

We present a novel contact-based grasp representation for a three-finger robotic dexterous hand. We propose an end-to-end neural network, CMG-Net, to predict the grasp pose for unknown objects from only one single-shot point cloud in a cluttered environment. To train CMG-Net, we generate a large-scale synthetic dataset for three-finger grasps. We have compared our approach against the state-of-the-art methods and observed 20% performance improvements in success rate.

There are a few limitations in our work. We mainly consider the grasping for static and rigid objects. This may limit its applications, especially in household scenes. Our approach requires the fingertips having contacts with the given object. However, in some grasps, this requirement may be not necessary. In the future, we would like to investigate the grasping representation and prediction for dynamical and soft objects. though it is much more difficult to formulate the problem. We would like to extend our approach to handle arbitrary grasps (i.e., with or without fingertip contacts).

Moreover, other possible future work may incorporate the grasping network into robotic manipulation skills, achieving more complex tasks with multi-finger hand, e.g., assembling or house cleaning.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China under grant 2021ZD0114501, the Science and Technology Innovation Action Plan of Shanghai under grant 22511105400, and partially supported by Ascend AI Computing Platform and CANN (Compute Architecture for Neural Networks).

REFERENCES

[1] H. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1Billion: A large-scale benchmark for general object grasping," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 441–11 450.

[2] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGB-D images: Learning using a new rectangle representation," in *IEEE International conference on robotics and automation*, 2011, pp. 3304–3311.

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

[4] A. Mousavian, C. Eppner, and D. Fox, "6-DoF GraspNet: Variational grasp generation for object manipulation," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2901–2910.

[5] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-DoF grasping for target-driven object manipulation in clutter," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 6232–6238.

[6] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 13 438–13 444.

[7] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," in *Robotics: Science and systems manipulation workshop*, 2007.

[8] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Deep differentiable grasp planner for high-DoF grippers," in *Robotics: Science and Systems*, 2020.

[9] K. Hang, J. A. Stork, and D. Kragic, "Hierarchical fingertip space for multi-fingered precision grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1641–1648.

[10] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4679–4684.

[11] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.

[12] R. Pelossof, A. T. Miller, P. K. Allen, and T. Jebara, "An SVM learning approach to robotic grasping," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 3512–3518.

[13] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "ContactGrasp: Functional multi-finger grasp synthesis from contact," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 2386–2393.

[14] J. Lundell, E. Corona, T. N. Le, F. Verdoja, P. Weinzaepfel, G. Rogez, F. Moreno-Noguer, and V. Kyrki, "Multi-FinGAN: Generative coarse-to-fine sampling of multi-finger grasps," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 4495–4501.

[15] Y. Li, W. Wei, D. Li, P. Wang, W. Li, and J. Zhong, "HGC-Net: Deep anthropomorphic hand grasping in clutter," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 714–720.

[16] E. Rimon and J. Burdick, *The Mechanics of Robot Grasping*. Cambridge University Press, 2019.

[17] C. Borst, M. Fischer, and G. Hirzinger, "Grasp planning: How to choose a suitable task wrench space," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 319–325.

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

[19] J. Varley, J. Weisz, J. Weiss, and P. K. Allen, "Generating multi-fingered robotic grasps via deep learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 4415–4420.

[20] Ü. R. Aktas, C. Zhao, M. S. Kopicki, A. Leonardis, and J. L. Wyatt, "Deep dexterous grasping of novel objects from a single view," *CoRR*, vol. abs/1908.04293, 2019.

[21] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Generating grasp poses for a high-DoF gripper using neural networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 1518–1525.

[22] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 4304–4311.

[23] 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," *arXiv preprint arXiv:1703.09312*, 2017.

[24] J. Lundell, F. Verdoja, and V. Kyrki, "DDGC: generative deep dexterous grasping in clutter," *CoRR*, vol. abs/2103.04783, 2021.

[25] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 269–279, 1989.

[26] 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 & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[27] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015.

[28] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, *et al.*, "SAPIEN: A simulated part-based interactive environment," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.

[29] C. Ferrari and J. F. Canny, "Planning optimal grasps." in *ICRA*, vol. 3, no. 4, 1992, p. 6.

[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Annual Conference on Neural Information Processing Systems*, 2017, pp. 5099–5108.

[31] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[32] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Annual Conference on Neural Information Processing Systems*, 2019, pp. 8024–8035.

[34] Y. Qin, R. Chen, H. Zhu, M. Song, J. Xu, and H. Su, "S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes," in *Conference on robot learning*. PMLR, 2020, pp. 53–65.

[35] C. Borst, M. Fischer, and G. Hirzinger, "Grasp planning: How to choose a suitable task wrench space," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 319–325.

[36] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning RGB-D feature embeddings for unseen object instance segmentation," in *Conference on Robot Learning*, 2020, pp. 461–470.