SuperQ-GRASP: Superquadrics-based Grasp Pose Estimation on Larger Objects for Mobile-Manipulation

Xun Tu and Karthik Desingh
University of Minnesota Twin Cities

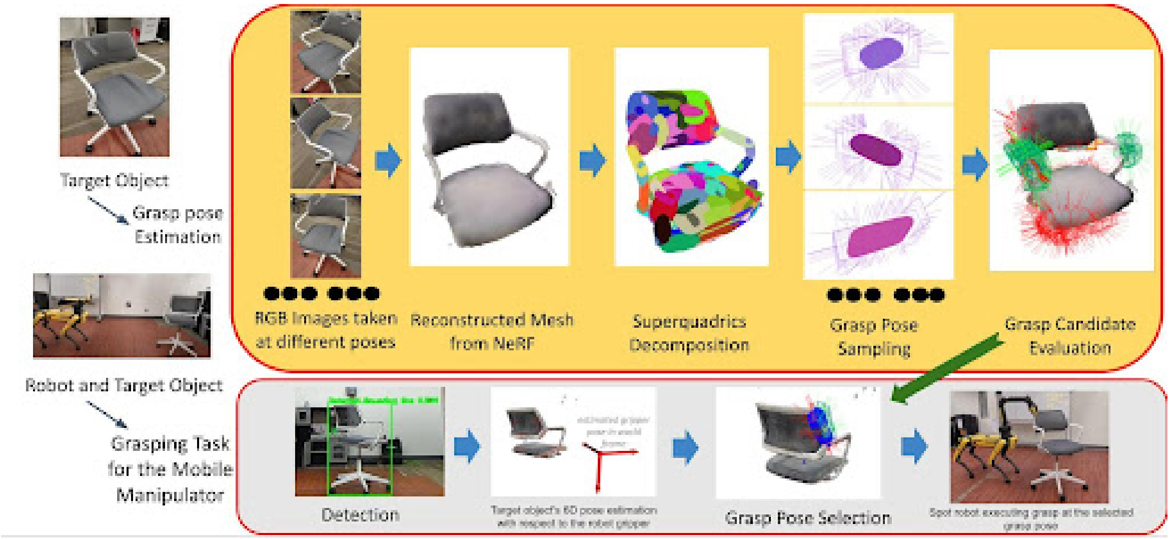
Target Object
Grasp pose Estimation
Robot and Target Object
Grasping Task for the Mobile Manipulator
RGB Images taken at different poses
Reconstructed Mesh from NeRF
Superquadrics Decomposition
Grasp Pose Sampling
Grasp Candidate Evaluation
Detection
estimated gripper pose in world frame
Target object's 6D pose estimation with respect to the robot gripper
Grasp Pose Selection
Spot robot executing grasp at the selected grasp pose

Fig. 1: An overview of our pipeline

Abstract*—Grasp planning and estimation has been a long-standing research problem in robotics, with two main approaches to find graspable poses on the objects. 1) geometric approach, which rely on 3D models of objects and gripper to estimate valid grasp poses, and 2) data-driven, learning-based approach, with models trained to identify grasp poses on raw sensor observations. The later assumes comprehensive geometric coverage during training phase, however are typically biased toward table-top scenarios and struggle to generalize to out-of-distribution scenarios with larger objects (e.g. chair). Additionally, raw sensor data (e.g. RGB-D data) from a single view of these larger objects is often incomplete and noisy and necessitates additional observations. K: Come back here to connect previous and later In this paper, we take a geometric approach, leveraging advancements in object modeling (e.g. NeRF) to build an implicit model by taking RGB images from views around the target object. This model enables the extraction of explicit mesh model while also capturing the visual appearance from novel viewpoints that is useful for perception tasks like object detection and pose estimation. We further decompose the NeRF-reconstructed 3D mesh into superquadrics (SQs) - parametric geometric primitives, each mapped to a set of precomputed grasp poses, allowing grasp composition on the target object based on these primitives. Our proposed pipeline overcomes the problems: a) noisy depth and incomplete view of the object, with a modelling step, b) generalization to objects of any size. K: Emphasize on the results

I. Introduction

To serve people, it would be helpful in many tasks for the robots to have the ability to manipulate the target object automatically, such as arranging furniture, moving heavy boxes, etc. Among many phases that might be part of a manipulation task, the grasping phase is the first and foremost to establish a firm contact with the object. Here, the grasping phase refers to the process of the robot arm holding the object at a specific location and orientation. More specifically, the robot needs to execute its gripper at the target grasp pose, which refers to the pose of the gripper before the robot closes the gripper to make contact with the target object. After the gripper is closed at the grasp pose, the robot should still hold the object stably in the downstream tasks. This paper focuses only on one step of the grasping phase specifically, which is to estimate the available grasp poses with respect to the target object.

Currently, there are mainly two approaches to estimate the valid grasp poses on the target object. One can be categorized into a geometric-based approach and the other a data-driven and raw sensor observation-based approach. The geometric-based approach is to take the 3D model of the object, study the geometric properties analytically, and estimate the grasping poses [10], [29], [42]. Typically, this method can generate precise grasp poses deterministically but usually requires an accurate mesh or point cloud of the target object, which is not always available. The key idea of the second approach - data-drive and sensor-based approach - is to train a deep learning network to estimate the grasp poses directly on the mesh or point cloud of the target object [24], [41], [47]. Through a long enough training process, the network can behave robustly to the outliers in the training dataset. However, the grasp poses predicted by a learned model are often biased to the distribution in the training dataset. Unfortunately, most of the existing works following the idea [2], [9], [40], [41], [47],

[52] do suffer from this issue. The datasets used to train these models are restricted to tabletop scenarios. The observed target objects are often convex in shape or of a low genus. Also, since usually the scenario is to grab the items from the table, the viewpoints to observe the objects are mostly top-down onto the table.

Now, we consider manipulation tasks in the context of mobile manipulators capable of moving larger objects that are typically more complex in shape and larger in size than tabletop scenarios. In addition, to grasp them, the viewpoints will vary in the 3D space instead of only going top-down. To deal with these objects, we find that the existing geometric-based and data-driven approaches are insufficient. As explained before, the accurate mesh or point cloud of the target object is not always available for the geometric-based approach. For the data-driven method, the networks are biased to the current training datasets containing mostly tabletop items, so they are appropriate for the target larger objects.

Therefore, the primary goal of our project is to enable a robot to automatically grasp a larger target object that is usually uncommon in a tabletop scenario, such as a chair or a table, as the fundamental step for the downstream mobile manipulation tasks. In this paper, we present a novel pipeline to achieve this goal. The whole pipeline mainly consists of four steps, inspired by [16], [49]. The first step is *3D Mesh Model Reconstruction*. To reconstruct the 3D mesh model of the target object from multiview RGB images. Here, we build our module based on NeRF modeling [28], [32]. The second step is *Primitives Decomposition*. That takes in the reconstructed 3D mesh model of the target object and decomposes it into several primitive shapes known as *Superquadrics*. The third step is the *Grasp Pose Estimation*. That takes in the superquadric representation of the target object and estimates grasp poses for each of the individual superquadrics as candidates. Finally, the *Grasp Candidate Validation* is to check the plausibility and collision of the grasp candidates concerning the original mesh model. Thus, the whole pipeline can give valid and stable grasp poses on a target object.

In summary, the contributions of the paper are:

1) Given a mesh of the target object, we propose a novel superquadrics-based method to estimate valid grasp poses on the target object;
2) For the larger target object, we propose a comprehensive pipeline that can take only RGB images on the target large objects as input and output valid poses for the downstream mobile manipulation tasks
3) We carry out experiments on both synthetic data in open3d [54] and real-robot trials using Boston-Dynamics Spot robot to evaluate the robustness of the pipeline;

II. Related Work

In this section, we focus on the related works about robotic grasping and shape abstraction. The section mainly consists of two parts. We first review how the existing works process the different input data formats to predict grasp poses on the target objects in Sec. II-A and Sec. II-B. Secondly, we revisit the current methods in primitive shape decomposition of object mesh and how this idea is introduced into robotic grasping in Sec. II-C.

A. Input Data Format for Robotic Grasping Task

So far, to generate the grasping poses for the robot to reach, different formats of input data have been used. On the one hand, the existing works have used visual input in the form of images [2], [5], [12], [13], [46], [52]. For example, [39], [52] has a multiview setup. The collected images is fed into a deep learning pipeline, the output of which will guide the robot to place its gripper. On the other hand, some work [4], [19], [53] process depth data, or a point cloud of the target object, to predict the grasping poses. For example, [53] uses a depth camera over the target object to collect point cloud data and convert it further into a Truncated Signed Distance Function (TSDF) to feed into deep learning networks. In addition, there are also some works [5], [18], [26], [37], [38], [40] that incorporate both ideas and use RGB-D images to enhance the performance.

However, most of the existing methods are collecting different data formats in a small space, such as the tabletop, which limits their application to a broader range of situations. For the larger objects in this paper, the collected depth information and point cloud data are usually noisy, incomplete, and only partial in view.

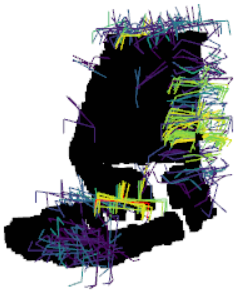

Fig. 2: An example of incomplete, noisy depth data observation on the chair (the arm rest is missing)

Thus, motivated by the progress in 3D scene reconstruction [14], [15], [28], [32], we propose a pipeline that takes RGB images on the target object at different angles as the input, and relies on the built-in functionalities of instant-NGP [32] to reconstruct the mesh of the target object.

B. Grasp Pose Estimation on target Object

To estimate the grasp poses on the target object, there are two main approaches: a geometric-based approach that depends on the analytical method to predict grasp poses on the target object and a data-driven approach that feeds the raw sensor observation into an end-to-end deep learning model for the same purpose. Traditionally, given the point cloud or the mesh of the target object, the geometric-based approach [6], [10], [29], [30], [33], [35], [42] will first study the geometric properties of the object and predict grasp poses based on these properties. For example, [10], [33], [42] study the object's curvature to determine the graspable regions of the object at first and then estimate the grasp poses. These approaches can provide deterministic results satisfying the physical constraints but usually require an accurate mesh or point cloud of the object. Although the object's precise mesh or point cloud can be obtained through 3D laser scanning [48], significant hardware is required. Alternatively, several cheaper methods exist to get the object mesh from only visual inputs [14], [32]. However, the noises in the final results may still significantly affect the geometric-based approach's performance.

Recently, methods employing trained deep learning models have attracted attention. As is shown in [4], [9], [18], [24], [26], [41], [47], [51], [53], the raw sensor observations, such as RGB-D images or a point cloud, will be preprocessed and fed into deep learning networks. The networks' output will be the target grasp poses for the gripper to execute. One obvious limitation is that these end-to-end models are usually trained on datasets containing mostly small objects that can be placed on the tables. One issue is that most items in the training dataset may be convex or of low genus. For example, the objects in YCB dataset to train Contact Graspnet [41], such as the tennis balls, cubes, bottles, etc., are different than the common large objects in daily life in shape, such as the chair, cart, luggage case, etc. Another issue is that the items on the table are always observed top-down, while the viewing direction to observe a large object can vary in 3D space. Therefore, the networks' performance on the larger objects that are uncommon in a tabletop scenario is affected. One possible solution to this problem is to construct a new training dataset through physical simulation. However, this is still challenging. The current physical simulation engines [25], [43], [50] are not designed to capture the complex inertial properties and dynamic behaviors of non-tabletop objects. So, not only the dataset construction task is costly, but also the quality of the constructed dataset is not guaranteed.

Thus, mainly inspired by the works in [3], [49], we propose a simple, non-deep learning method to deal with the limitation. It depends on an existing method robust to the outliers in the object's mesh. Also, the grasp pose estimation process does not depend on any deep learning network, so the problem bias in the training dataset will not exist.

C. Primitive Decomposition in Grasp Pose Estimation

Primitive Decomposition refers to arranging several primitive shapes in a certain way so that the final combination is close to the target object in shape. Traditionally, [44], [55] have used deep learning networks to split the target mesh into cuboids. Recently, inspired by the works in [1], [21], [34], [22] have used a more expressive primitive, i.e superquadrics to do the job. So far, their works are still on the visual level, without any manipulation task. In our pipeline, we extend their works to predict valid grasp poses based on the primitives they have split from the target object mesh. Our pipeline follows some ideas from GraspIt [29], but we don't require the user to input the estimated primitive shape manually. The works mostly similar to ours are [45], [49], but they only deal with small objects on the table. To handle the more complex large objects in space, we use a more efficient and robust primitive decomposition and have designed a more sophisticated method to predict the grasp poses on the selected primitive shape.

III. Problem Statement

The part of the entire pipeline of most novelty is the grasp pose estimation step. As for this step, the input is the mesh M of the target object. The goal is to predict several valid grasp poses $\{(R_i, t_i)\}$. Here, ”validness” demands two requirements. One is that there should be no collision between the body of the robot's gripper $B(g)$ and the mesh of the target object M. The other is that the near-antipodal metrics [42] should be satisfied for a valid grasp pose. We refer to IV-D for more details.

In our pipeline, we assume that the mesh of the target object M can be extracted manually from the background scene in instant-ngp [32]. Also, we assume that the quality of the reconstructed mesh from instant-NGP is good enough for the following grasp pose prediction task. To obtain the mesh, we feed several images $\{(I_k, P_k)\}$ at different poses into the NeRF pipeline to reconstruct the scene, and depend on instant-NGP's built-in functionalities to crop out the target object and build up the mesh. In the following grasp pose estimation step, we use a parallel gripper to approximate the complicated shape of the real-world gripper of SPOT [11].

IV. Methodology

The architecture of our system is given in 3. It mainly consists of four steps. The first step is to generate a mesh of the target object. We rely on the existing works in instant-NGP [28], [32]. It can take several RGB images taken on the target object at different poses, and reconstruct a mesh. The second step is decomposing the mesh into several primitive shapes called *Superquadrics*. This step is based on an optimization-based method called Marching Primitives [22]. Next, the third step is to sample grasp poses on the generated individual superquadrics as candidates based on our novel analytical method. Finally, the fourth step is to check the plausibility of grasp pose candidates based on collision tests and antipodal metrics [42] concerning the original mesh, where the invalid ones will be further filtered out.

A. Superquadrics

Superquadrics is a series of shapes that are used in shape abstraction due to its high expressiveness [21], [34]. Mathematically, the implicit function for a superquadric is given as

$$\left(\left(\frac{x}{a_x}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_y}\right)^{\varepsilon_2}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_z}\right)^{\frac{2}{\varepsilon_1}} = 1 \qquad (1)$$

Therefore, a superquadric can be encoded by only 5 parameters: a_x, a_y, a_z as the length of the principal axes, and

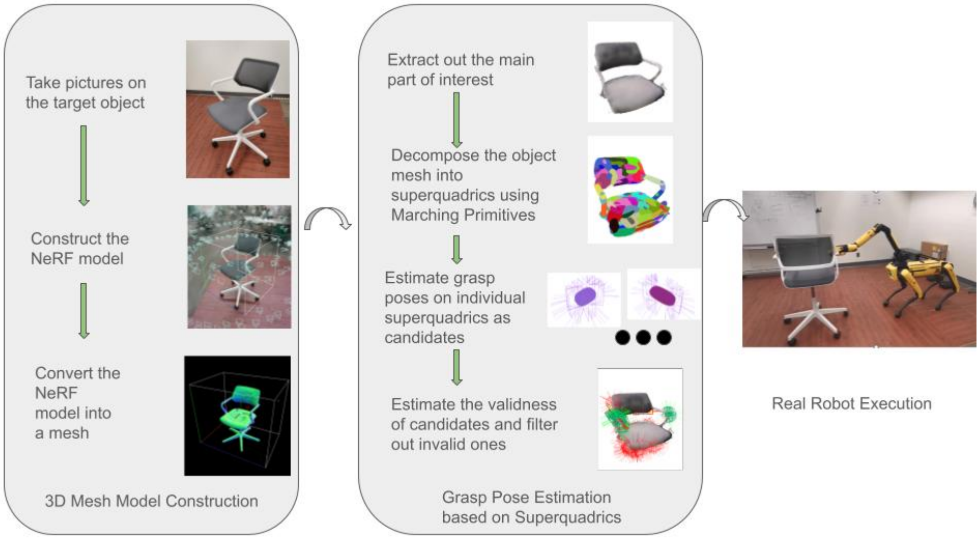
Take pictures on the target object
Construct the NeRF model
Convert the NeRF model into a mesh
3D Mesh Model Construction
Extract out the main part of interest
Decompose the object mesh into superquadrics using Marching Primitives
Estimate grasp poses on individual superquadrics as candidates
Estimate the validness of candidates and filter out invalid ones
Grasp Pose Estimation based on Superquadrics
Real Robot Execution

Fig. 3: System Architecture

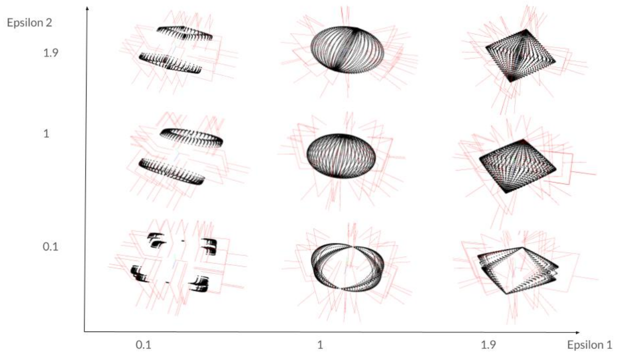

Fig. 4: Grasp pose on individual superquadrics of different parameters

$\varepsilon_1, \varepsilon_2 \in [0, 2]$ as the parameters to determine the shape. It can also be expressed in the format of spherical product:

$$\mathbf{r}(\eta, \omega) = \begin{bmatrix} a_x \cos^{\varepsilon_1} \eta \cos^{\varepsilon_2} \omega \\ a_y \cos^{\varepsilon_1} \eta \sin^{\varepsilon_2} \omega \\ a_z \sin^{\varepsilon_1} \eta \end{bmatrix} \tag{2}$$

where $\eta \in [-\pi/2, \pi/2], \omega \in [-\pi, \pi]$. Though the mathematical expressions are relatively simple, they can still cover a range of different shapes (See 4), so we decide to use them as the format of the decomposed primitive shapes.

B. Primitives Decomposition

There are already several methods [34], [45], [49] that can decompose the target mesh into superquadrics. Compared to them, we finally decide to implement the idea of Marching Primitives, considering the quality of the final decomposition and the efficiency of processing. This is a method similar to the idea of Marching Cubes [23], which splits the target mesh into several regions, and models the challenge as a nonlinear least square optimization problem with bounded inputs. We refer to [22] for more details.

C. Grasp Pose Sampling

After the decomposition is completed, we sample grasp poses directly on the individual superquadrics as grasp pose candidates. When one specific superquadric is selected, we firstly find the 5 parameters to determine its shape (See IV-A), i.e. $(a_x, a_y, a_z, \varepsilon_1, \varepsilon_2)$. Next, we find the cross-sectional plane perpendicular to the shortest axis. For example, suppose a_z is the shortest axis, we choose the part of the superquadric on xy-plane. According to Eq.(2), the equation will be given as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_x \cos^{\varepsilon_2} \omega \\ a_y \sin^{\varepsilon_2} \omega \end{bmatrix} \tag{3}$$

When $\varepsilon_2 \geq 1$, there is no discontinuity, so we generate grasp poses given at $[(a_x + l_t)\cos^{\varepsilon_2}, (a_y + l_t)\sin^{\varepsilon_2}]$, where l_t is the threshold value. When $\varepsilon_2 < 1$, there will be discontinuities if we use Eq.(2) (See). To overcome it, we calculate the derivatives in the first quarter $(x > 0, y > 0, \omega \in [0, \pi/2])$ as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -a_x \varepsilon_2 \sin^{\varepsilon_2 - 1} \omega \\ a_y \varepsilon_2 \cos^{\varepsilon_2 - 1} \omega \end{bmatrix} \tag{4}$$

Notice that when $\varepsilon_2 < 1$, $x' \to \infty$ when $\omega \to 0$, and $y' \to \infty$ when $\omega \to \pi/2$. These ”jumps” in derivatives lead to the discontinuity. Therefore, we calculate the boundary points as the start of the ”jumps” of derivatives given by ω_1, ω_2 as

$$x'(\omega_1) = -4, y'(\omega_2) = 4 \tag{5}$$

We then connect a line between the two boundary points and sample grasp poses along the line (See 5). Finally, we generate the grasp candidates in other quarters by symmetry and combine all grasp poses together as the grasp candidates associated to this specific superquadric.

D. Grasp Candidate Validation

After grasp candidates are generated, we further validate them based on the two metrics. One is that at the specified

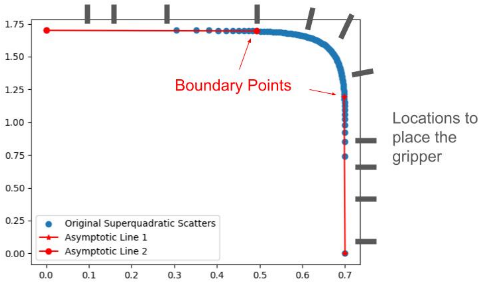

Fig. 5: Illustration of boundary points and locations of grasp poses on individual superquadrics

pose, there should be no collision between the gripper and the original object mesh. Quantitatively, the minimum signed distance between the points on the body of the gripper $B(g)$ and the object mesh M should be larger than a threshold ε:

$$\forall p \in B(g), d(p, M) \geq \varepsilon$$

The other is that the antipodal metrics in [42] need to be satisfied so the grasping can be stable:

For thresholds $k \in \mathbb{N}$ and $\theta \in [0, \pi/2]$, there exists several points $p_1, p_2, \cdots, p_k \in C(g) \cap V$, and $q_1, q_2, \cdots q_k \in C(g) \cap V$, where $C(g)$ is the closing region of the gripper and V is the set of vertices of M, such that $\hat{n}(p_i)^T \hat{f}(g) \geq \cos\theta$ and $\hat{n}(q_i)^T \hat{f}(h) \leq -\cos\theta$., where $\hat{n}(p_i), \hat{n}(q_i)$ are the normal vectors on the object mesh and $\hat{f}(g)$ is the closing direction of the gripper.

Due to computation cost, in practice we just select the closest superquadric to the gripper iteratively until valid grasp poses are found for the downstream task. However, the users can always select any number of superquadrics at desired locations according to their specific demands.

V. Experimental Setup

To demonstrate the effectiveness of our pipeline, on the one hand, we evaluate the quality of estimated grasp poses on the simulation data. The data are taken from both the existing PartNet-Mobility dataset [7], [31], [50] and real-world scene. On the other hand, we validate its performance in mobile manipulation tasks through several real-world experiments using SPOT [11].

A. Dataset

Firstly, we select 15 synthetic objects from PartNet dataset and 5 objects from real-world scene to evaluate our method. They are representative common large objects in daily life. Also, the chosen objects have covered enough hierarchies.

To fully simulate the process, we pick each of the 15 objects from PartNet-Mobility (see 6) and place its ground-truth meshes in the simulation environment called Pyrender [27]. We then take images at several poses and obtain the reconstructed mesh of the target objects through instant-NGP [32]. For the 5 objects from the real world, we follow the guidance in instant-NGP to collect their reconstructed mesh.

B. Grasp Pose Evaluation on Synthetic Data

To evaluate the effectiveness of our method for various viewpoints, for each of the reconstructed mesh, we place a gripper at eight poses selected randomly from the two semi-spheres centered at the object but of different radius. Then, we predict 50 potential grasp pose candidates at each gripper location using our pipeline as well as the two baseline methods:

- Contact GraspNet [41] the entire object mesh
- Contact GraspNet on the one-shot depth image

Since it is hard to evaluate the quality of the grasp poses through simulation, due to the complex physical mechanism of large objects and noisy reconstructed mesh, we have adopted our own two metrics:

1) The number of valid grasp poses among the 50 grasp candidates based on metrics mentioned in IV-D
2) The relative transformation between the closest grasp pose to the current gripper pose. We hope that the transformation would be as small as possible to minimize the energy cost and collision risk in mobile manipulation

C. Real-world Mobile Manipulation Experiment

To validate the performance of the pipeline in real practices, we place each of the 5 real-world in a specified location with arbitrary orientation and command the SPOT robot to estimate the object's pose and find graspable pose, and execute a reach and grasp action (See 1. To estimate the object's pose, from [8] we use its idea of feature matching and 2D-3D correspondance. Also, to improve the pose estimation accuracy, we use GroundingSAM [17], [20], [36] to filter out the background. After the robot executes the gripper at the predicted pose and close the gripper, We record if the robot succeeded in grasping this object. We repeat the process for 15 times, and record the success rate. For more qualitative demonstrations, see the supplementary material.

VI. Results

A. Grasp Pose Evaluation on Synthetic Data

To evaluate the effectiveness of our method for various view point, we select eight poses randomly and evaluate the performance of our pipeline and the baseline based on the number of valid grasp poses and the relative transformation between the gripper pose and grasp pose. These are a part of the results in Table I and Table II

TABLE I: Mean number of valid grasp poses for each object (part)

	Chair	Suitcase	Cart	Table	Chair (real)
CG+Mesh	7.00	4.62	9.62	12.62	3.50
CG+Depth	8.00	2.37	**24.5**	1.87	7.00
Ours	**15.75**	**12.12**	18.37	**17.50**	**20.25**

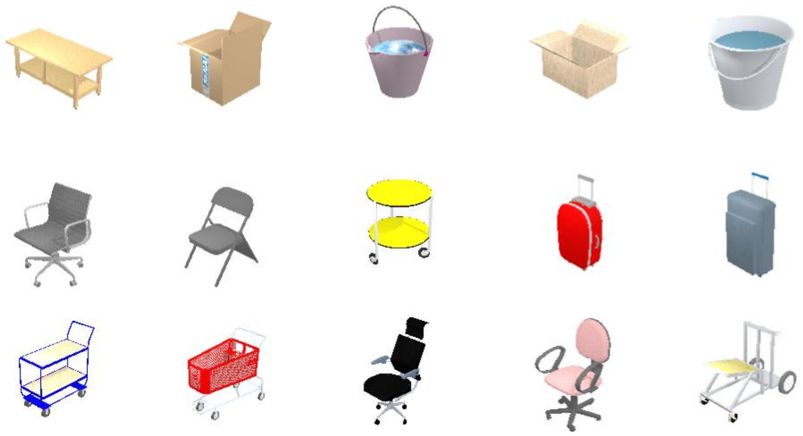

Fig. 6: Synthetic Objects used for Experiments

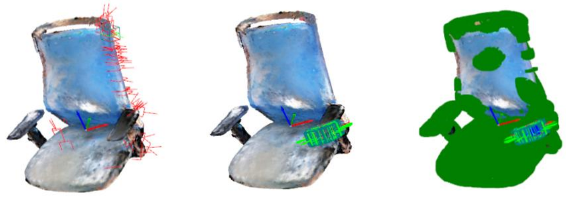

Fig. 7: Qualitative Results of predicted grasp poses

TABLE II: Relative Transformation between target grasp pose & camera pose (part); mRD: mean Rotational difference; mTD: mean Translation distance

	Chair		Suitcase		Chair (real)	
CG + mesh	mRD (°)	mTD	mRD (°)	mTD	mRD (°)	mTD
	13.66	1.61	21.73	1.82	6.74	1.75
CG + depth	mRD (°)	mTD	mRD (°)	mTD	mRD (°)	mTD
	24.39	1.68	21.30	1.44	17.23	1.58
Ours	mRD (°)	mTD	mRD (°)	mTD	mRD (°)	mTD
	24.85	**1.53**	**6.92**	**1.42**	**4.68**	**1.55**

Despite only focusing on predicting grasp poses on one local region, i.e., one superquadric, our method can still generate comparable, even larger number of valid grasp poses than the two baseline methods. Also, our method performs better in the minimum translational distance between the predicted grasp pose and the gripper pose, because ours focuses more on the local region than the baseline method to feed the whole mesh to Contact Graspnet. The baseline method to feed only one partial view of depth data to Contact Graspnet can generate minimum translational distance closer to ours. Still, it focuses too much on the local features without any idea of the full mesh, so it cannot compete with the number of valid grasp poses.

B. Real-world Mobile Manipulation Experiment

For the real-world experiments, we record the number of successful trials. In this part, we only compare our pipeline against the baseline where a Contact GraspNet model is applied on the whole mesh because, from the experiments based on synthetic data, this baseline method has a better performance. Here are the results in Table III. Our process has a significantly higher success rate. For more results, please go to our website.

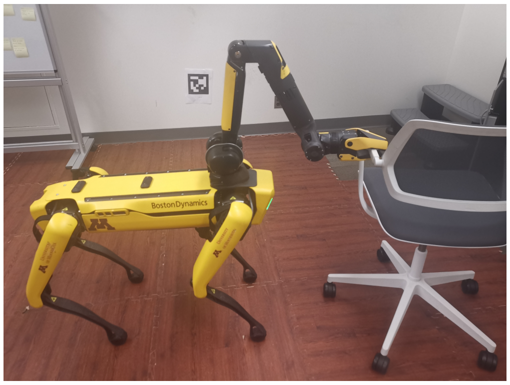

Fig. 8: Qualitative Result on real-world experiments

TABLE III: Number of Successful Trials

	obj 1	obj 2	obj 3	obj 4	obj 5
CG+Mesh	/15	/15			
Ours	/15	/15			

VII. Conclusion

In this project, we have proposed a pipeline that can take only RGB inputs of the target large object that is usually not common in a tabletop scenario, and then estimate several valid grasp poses for mobile manipulation. The robot can grasp the target object by placing its gripper at those locations and may continue the downstream tasks. The pipeline is mainly based on our novel method to split the mesh into several primitives and estimate grasp poses on each individual primitive separately. By combining it with the existing works on object detection and pose estimation, we have verified its effectiveness through real-world experiments.

References

[1] Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[2] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning, 2023.

[3] Irving Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32(1):29–73, 1985.

[4] Ondrej Biza, Skye Thompson, Kishore Reddy Pagidi, Abhinav Kumar, Elise van der Pol, Robin Walters, Thomas Kipf, Jan-Willem van de Meent, Lawson L. S. Wong, and Robert Platt. One-shot imitation learning via interaction warping, 2023.

[5] Ben Burgess-Limerick, Chris Lehnert, Jurgen Leitner, and Peter Corke. An architecture for reactive mobile manipulation on-the-move, 2022.

[6] Junhao Cai, Jingcheng Su, Zida Zhou, Hui Cheng, Qifeng Chen, and Michael Yu Wang. Volumetric-based contact point detection for 7-dof grasping. In *Conference on Robot Learning (CoRL)*. PMLR, 2022.

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[8] Ronghan Chen, Yang Cong, and Yu Ren. Marrying nerf with feature matching for one-step pose estimation, 2024.

[9] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics (T-RO)*, 2023.

[10] B. Faverjon and J. Ponce. On computing two-finger force-closure grasps of curved 2d objects. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 424–429 vol.1, 1991.

[11] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion, 2022.

[12] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation, 2023.

[13] Ethan Kroll Gordon, Amal Nanavati, Ramya Challa, Bernie Hao Zhu, Taylor Annette Kessler Faulkner, and Siddhartha Srinivasa. Towards general single-utensil food acquisition with human-informed actions. In *7th Annual Conference on Robot Learning*, 2023.

[14] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *arXiv preprint arXiv:2311.12775*, 2023.

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[16] Justin Kerr, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. Evo-neRF: Evolving neRF for sequential robot grasping of transparent objects. In *6th Annual Conference on Robot Learning*, 2022.

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[18] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps, 2014.

[19] Yulong Li, Andy Zeng, and Shuran Song. Rearrangement planning for general part assembly, 2023.

[20] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[21] Siqi Liu, Yong-Lu Li, Zhou Fang, Xinpeng Liu, Yang You, and Cewu Lu. Primitive-based 3d human-object interaction modelling and programming, 2023.

[22] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory Chirikjian. Marching-primitives: Shape abstraction from signed distance function. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] William Lorensen and Harvey Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21:163–, 08 1987.

[24] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[25] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[26] Jiayuan Mao, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Learning reusable manipulation strategies, 2023.

[27] Matthew Matl. Pyrender. https://github.com/mmatl/pyrender, 2019.

[28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *CoRR*, abs/2003.08934, 2020.

[29] A.T. Miller and P.K. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Automation Magazine*, 11(4):110–122, 2004.

[30] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 2, pages 1824–1829 vol.2, 2003.

[31] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[33] Andreas ten Pas and Robert Platt. *Localizing Handle-Like Grasp Affordances in 3D Point Clouds*, pages 623–638. Springer International Publishing, Cham, 2016.

[34] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] Justus Piater. Learning visual features to predict hand orientations. 07 2000.

[36] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[37] Gautam Salhotra, I-Chun Arthur Liu, and Gaurav Sukhatme. Learning robot manipulation from cross-morphology demonstration, 2023.

[38] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation, 2022.

[39] Anthony Simeonov, Ankit Goyal, Lucas Manuelli, Lin Yen-Chen, Alina Sarmiento, Alberto Rodriguez, Pulkit Agrawal, and Dieter Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement, 2023.

[40] Matan Sudry, Tom Jurgenson, Aviv Tamar, and Erez Karpas. Hierarchical planning for rope manipulation using knot theory and a learned inverse model. In *7th Annual Conference on Robot Learning*, 2023.

[41] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. 2021.

[42] Andreas ten Pas and Robert Platt. *Using Geometry to Detect Grasp Poses in 3D Point Clouds*, pages 307–324. Springer International Publishing, Cham, 2018.

[43] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[44] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Computer Vision and Pattern Regognition (CVPR)*, 2017.

[45] Giulia Vezzani, Ugo Pattacini, and Lorenzo Natale. A grasping approach based on superquadric models. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1579–1586, 2017.

[46] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play, 2023.

[47] Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15964–15973, 2021.

[48] Jing Wang, Juan Zhang, and Qingtong Xu. Research on 3d laser scanning technology based on point cloud data acquisition. In *2014 International Conference on Audio, Language and Image Processing*, pages 631–634, 2014.

[49] Yuwei Wu, Weixiao Liu, Zhiyang Liu, and Gregory S. Chirikjian. Learning-free grasping of unknown objects using hidden superquadrics, 2023.

[50] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[51] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross embodiment skill discovery. In *7th Annual Conference on Robot Learning*, 2023.

[52] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields, 2023.

[53] Xuechao Zhang, Dong Wang, Sun Han, Weichuang Li, Bin Zhao, Zhigang Wang, Xiaoming Duan, Chongrong Fang, Xuelong Li, and Jianping He. Affordance-driven next-best-view planning for robotic grasping. In *7th Annual Conference on Robot Learning*, 2023.

[54] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[55] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.