kPAM-SC: Generalizable Manipulation Planning using KeyPoint Affordance and Shape Completion

Wei Gao, Russ Tedrake

Abstract— **While traditional approaches to manipulation planning assume known object templates, recent approaches to "category-level manipulation" aim to manipulate a category of objects with potentially unknown instances and large intra-category shape variation. In this paper we explore an object representation to enable precise category-level manipulation, capturing a notion of the object configuration and extent, while being generalizable to novel instances. Building on our previous work, kPAM[1], we combine semantic keypoints with dense geometry (a point cloud or mesh) as the interface between the perception module and motion planner. Leveraging advances in learning-based keypoint detection and shape completion, both dense geometry and keypoints can be perceived from raw sensor input. Using the proposed hybrid object representation, we formulate the manipulation task as a motion planning problem which encodes both the object target configuration and physical feasibility for a category of objects. In this way, many existing manipulation planners can be generalized to categories of objects, and the resulting perception-to-action manipulation pipeline is robust to large intra-category shape variation. Extensive hardware experiments demonstrate our pipeline can produce robot trajectories that accomplish tasks with never-before-seen objects. The video demo is available on this link: https://sites.google.com/view/generalizable-manipulation.**

I. Introduction

This paper focuses on robotic manipulation planning at a category level. In particular, the pipeline should take raw sensor inputs and plan physically feasible robot trajectories that will move a set of objects to their target configuration. For example, the robot should "place mugs in a box" and handle potentially unknown instances in the mug category, despite the variation of shape, size, and topology. Accomplishing these types of tasks is of significant importance to both industrial applications and interactive assistant robots.

To achieve this goal, we need an object representation that (1) is accessible from raw sensor inputs, (2) with which the manipulation planner can reason about both the physical feasibility and desired object configuration, (3) can generalize to novel instances. Perhaps 6-DOF pose is the most widely-used object representation in robotic manipulation. Most manipulation planning algorithms [27], [24], [4] assume the known geometric template and 6-DOF pose of the manipulated objects. Consequently, many contributions from the vision community [23], [30], [20], [3] focus on pose estimation. However, as detailed in Sec. V-B, representing an object with a parameterized pose defined on a fixed geometric template, as these works do, may not adequately capture large intra-class shape or topology variations. On the other hand, many works [6], [19], [32], [10] focus on robot grasping planning. Although these methods enable robotic picking for arbitrary objects, it is hard to extend them to more complex tasks that involve object target configuration.

In our previous work kPAM [17], we represent objects using semantic 3D keypoints, which provides a concise way to specify the target configuration for a category of objects. Although this approach has accomplished several category-level manipulation tasks, it lacks the complete and dense geometric information of the object. Thus, kPAM [17] cannot reason about physical properties such as the collision, static equilibrium, visibility, and grasp stability of the planned robot action, despite their obvious importance in robotic applications. Authors of kPAM need to manually specify various intermediate robot configurations, which can be labor-intensive and sub-optimal. Furthermore, the pipeline can be overly confident in physically infeasible robot trajectories and send them for execution, which is rather dangerous.

Building on kPAM [17], we resolve this limitation with a new hybrid object representation which combines both (i) semantic 3D keypoints and (ii) full dense geometry (a point cloud or mesh). The dense geometry is obtained by leveraging well-established shape completion algorithms [34], [18], [21], which generalize well to novel object instances. With the combined dense geometry and keypoints as the object representation, we formulate the manipulation task as a motion planning problem that can encode both the object target configuration and physical feasibility for a category of objects. This motion planning problem can be solved by a variety of existing planners and the resulting robot trajectories can move the objects to their target configuration in a physically feasible way. The entire perception-to-action manipulation pipeline is robust to large intra-category shape variation. Extensive hardware experiments demonstrate our method can reliably accomplish manipulation tasks with never-before-seen objects in a category.

The contribution of this work is twofold. Conceptually, we introduce a hybrid object representation consists of dense geometry and keypoints as the interface between the perception module and planner. This representation has similar functionalities with the existing 6-DOF pose representation with templates in robot manipulation, while the generalizability to novel instances makes it a promising alternative. Systematically, our work is the first one we know that integrates shape completion with manipulation planning and demonstrate its generalization. This integration enables many existing manipulation planners, either optimization-based methods [24], [27] or sampling-based approaches [4],

CSAIL, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, USA. Emails: weigao@mit.edu, russt@mit.edu.

[1]kPAM [17] stands for **K**ey**P**oint **A**ffordance-based **M**anipulation.

[25], to handle a category of objects in a unified and precise way.

This paper is organized as follows: in Sec. II we review related works. Sec. III provides background knowledge about the keypoint representation in [17]. Sec. IV describes our manipulation pipeline. Sec. V demonstrates our methods on several manipulation tasks and shows generalization to novel instances. Sec. VI concludes this work.

II. Related Work

A. Pose-based Robotic Manipulation

For most manipulation applications, the object pose is the default interface between the perception and planning modules: the perception module estimates the pose from raw sensor inputs; the planning module takes the estimated pose as input and plans robot actions to accomplish some manipulation tasks. Researchers have made many contributions on both the pose estimation [23], [30], [20], [3] and manipulation planning [24], [27], [4], [25]. Some works [1], [11] integrate state-of-the-art pose estimators with trajectory planners to build fully functional manipulation pipelines and solve real-world tasks such as packing and assembly.

However, pose estimation can be ambiguous under large intra-category shape variations, and using one geometric template for motion planning and object target specification can lead to physically infeasiblity for other instances. A comparison is made in Sec. V.

B. Manipulation at a Category Level

Several existing works aim at robot manipulation for a category of object. Among them, kPAM [17] uses a modularized pipeline where the semantic keypoints are used to represent the object. However, kPAM cannot reason about physical feasibility. As a result, unsafe robot trajectories can be sent to robot execution instead of stopped early (see Sec. V-E), which can be rather dangerous. On the other hand, [5] demonstrates robotic pick-and-place across different instances using reinforcement learning. However, the reward engineering and training procedure in these algorithms limit the generalization to new target configurations and manipulation tasks.

C. Grasping and Manipulation with Shape Completion

Robot grasp planning is the task of computing a stable grasp pose that allows the robot to reliably pick up the object. Among various approaches for grasp planning, model-free methods [32], [6], [19], [15] typically train deep networks that take raw sensor observations as input and produce grasp poses as output. In contrast, model-based algorithms [33], [16], [29] estimate the grasp quality based on geometric information such as antipodal points or surface normal, then select the grasp with the best grasp quality.

As the geometry obtained from typical RGBD sensors are noisy and incomplete, several works [29], [14], [31], [28] combine shape completion with grasping planning for improved performance and robustness. [22] also shows shape completion can improve the performance of robot object searching.

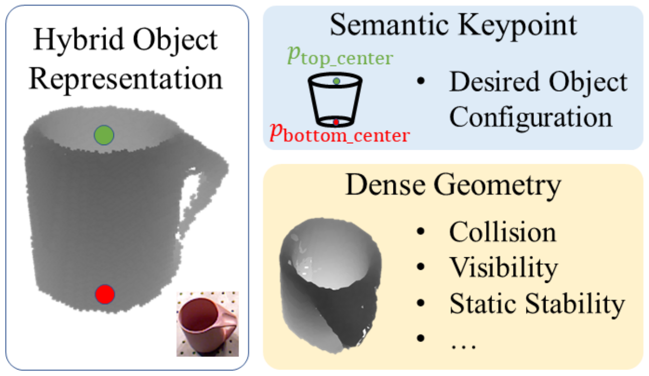

Fig. 1: An overview of the proposed hybrid object representation for category-level manipulation planning using the mug as an example. We exploit a hybrid object representation consists of semantic keypoints and dense geometry. The semantic keypoint are used to specify the desired object target configuration, while the dense geometry is used to ensure the physical feasibility of the planned robot action. Benefit from advances in learning based keypoint detection and shape completion, the proposed object representation can be obtained from raw images, and the resulting perception-to-action pipeline generalizes to novel instances within the given category.

In this work, we are interested in category-level robotic manipulation planning. This task requires reasoning about both the desired object configuration and physical feasibility, and is out of the scope for the above-mentioned methods.

III. Preliminary: kPAM

In this section we briefly recap kPAM [17]. kPAM is a framework to specify the object target configuration for manipulation. In kPAM, each object is represented by several semantic 3D keypoints $p \in R^{3 \times N}$, where N is the number of keypoints. Using the mug category as an example, we may represent the mug by $N = 2$ keypoints: $p_{\text{top\_center}}$ and $p_{\text{bottom\_center}}$, as shown in Fig. 1. These keypoints are detected from raw sensor inputs (typically RGBD images) using neural network based detectors.

In kPAM, the robot action is abstracted as a rigid transformation $T_{\text{action}} \in SE(3)$ on the object, and the transformed keypoint would be $T_{\text{action}} p$. The object target configuration is specified as a set of geometric costs/constraints on the transformed keypoint $T_{\text{action}} p$. Using the mug in Fig. 1 as an example, to place the mug upright at some target location p_{target}, the planned robot action T_{action} should satisfy

$$||T_{\text{action}} p_{\text{bottom\_center}} - p_{\text{target}}|| = 0 \tag{1}$$

After the object has been grasped, the T_{action} would be applied, which is essentially a rigid transformation of the robot end-effector.

As the dense geometry information of the object is missing, it remains unclear how to ensure the robot trajectory that apply T_{action} to the object is physically feasible. This work resolves this issue.

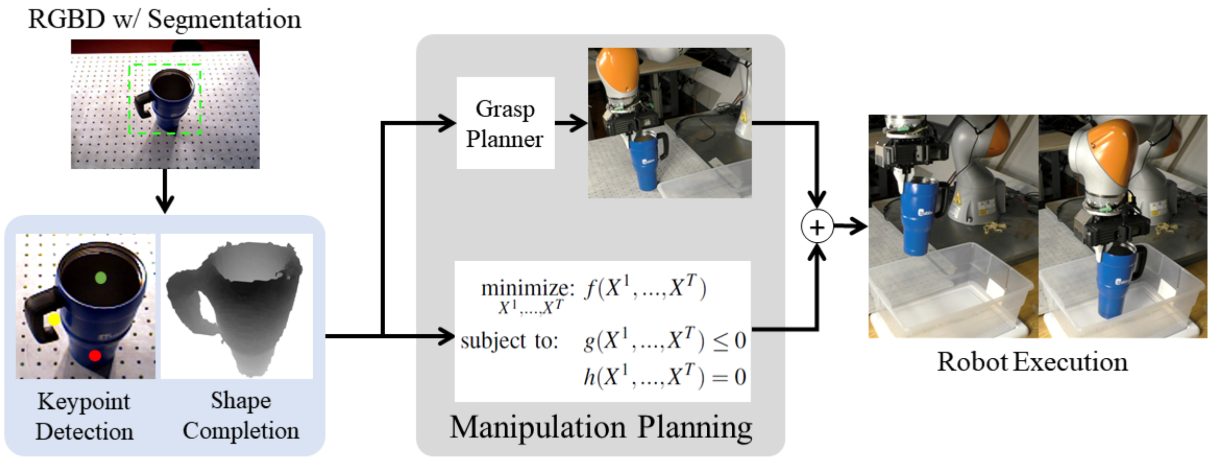

Fig. 2: An overview of the manipulation pipeline. The hybrid object representation consists of dense geometry and keypoint is used as the interface between the perception module and manipulation planner. Given a RGBD image with instance segmentation, we perform shape completion and keypoint detection to obtain dense geometry and 3D keypoints, respectively. Then, the perception result is used to plan robot trajectories that move the objects to their desired configurations while satisfying physical constraints. Semantic keypoints are used to specify the object target configuration, while dense geometry is used to ensure the physical feasibility of the planned robot action.

IV. Manipulation Framework

As mentioned in Sec. I, we need dense geometry to reason about the collision, static equilibrium, visibility, and grasp stability. There exist several possibilities to obtain it. In previous works, dense geometry is obtained by transforming the geometric template with an estimated pose. However as shown in our experiment, pose estimation does not generalize to new objects. One may also consider some coarse approximation, such as the convex hull of keypoints. However, the generality of the pipeline can be compromised if the geometry representation is not sufficiently accurate. For instance, the keypoints convex-hull approximation is not suitable for the “put mug on a rack” task in our experiment, where the non-convex mug handle geometry must be considered.

We propose to use shape completion, which directly estimated the accurate, dense and complete geometry (in the form of point cloud or mesh) from raw observation. Shape completion is extensively studied [34], [18], [21] in the vision community and well-trained shape complete networks generalize well to a category of objects with potentially unseen instances. Furthermore, we demonstrate that the data used to train the shape completion network can be collected automatically. These properties make shape completion a promising candidate for manipulation tasks.

As illustrated in Fig. 2, we use the hybrid object representation consists of dense geometry and keypoints as the interface between the perception and planning modules. The semantic keypoints are designated manually and used to specify the object target configuration, while the dense geometry is used to ensure the physical feasibility of the planned robot action. The perception part includes shape completion and keypoint detection, which is detailed in Sec. IV-A. The manipulation planning and grasp planning that use the perception results are presented in Sec. IV-B and Sec. IV-C, respectively.

A. Perception and Automatic Data Collection

The task of perception is to produce the proposed hybrid object representation from raw sensor inputs, which consists of 3D keypoints and dense geometry (point cloud or mesh). Note that although we present specific approaches used in our pipeline, any technique for keypoint detection and shape completion can be used instead. For keypoints we adopt the method in our previous work kPAM [17], and this subsection mainly focuses on the perception of dense geometry.

We use the state-of-the-art ShapeHD network [34] for 3D shape completion. ShapeHD is a fully convolutional network that takes RGBD images as input and predicts 3D volumetric occupancy. Then the completed point cloud can be extracted by taking the occupied voxel. If the object mesh is required, triangulation algorithms such as marching cubes can be used. The completed geometry are aligned with the observed object (viewer-centered in [34]) and expressed in the camera frame, we can further transform it into the world frame using the calibrated camera extrinsic parameters.

The shape completion network requires training data consists of RGBD images and corresponding ground-truth 3D occupancy volumes. We collect training data using a self-supervised method. Given a scene containing an object of interest, we first perform 3D reconstruction of that scene. Then, we perform background subtraction to obtain the reconstructed mesh of the object. Finally, we can get the occupancy volume by transforming the reconstructed mesh into camera frame and voxelization. Note that the data generation procedure does not require pre-built object template or human annotation.

In our experiment, we collect 117 scenes and over 100,000 pairs of RGBD images and ground-truth 3D occupancy volumes within four hours. Even with small amount of data we were able to achieve reliable and generalizable shape completion, some qualitative results are shown in Sec. V-B.

B. Manipulation Planning Formulation

In this subsection we discuss the formulation of manipulation planning problem. This planning problem can be solved by many existing planners [7], [4], [25] to produce robot trajectories. From another perspective, we aim to generalize these existing planners from a specific instance to a category of potentially unknown objects.

As mentioned in Sec. IV, we represent an object o_j by its 3D semantic keypoints $p_j \in R^{3\times N}$ and dense geometry (point cloud or mesh), where $1 \leq j \leq M$ and M is the number of objects. Following many existing works [7], [4], [25], we assume that the robot can change the state of an object only if the object is grasped by the robot. Furthermore, we assume the object is rigid and the grasp is tight. In other words, there is no relative motion between the gripper and the grasped object. Both the semantic keypoints and dense geometry would move with the robot end-effector during grasping. To achieve this, we need a grasp planner which is discussed in Sec. IV-C.

Given the object representation, the concatenated configuration for robot and objects at time t is defined as $X^t = [o_1^t, ..., o_M^t, q^t]$, where $1 \leq t \leq T$, T is the number of time knots and q^t is the robot joint configuration. The general planning problem can be written as

$$\underset{X^1,...,X^T}{\text{minimize:}} \ f(X^1, ..., X^T) \tag{2}$$

$$\text{subject to:} \quad g(X^1, ..., X^T) \leq 0 \tag{3}$$

$$h(X^1, ..., X^T) = 0 \tag{4}$$

where f is the objective function, g and h are the concentrated inequality and equality constraints. If optimization-based planning algorithms [24], [26] are used to solve the problem (2), f, g and h should be differentiable. On the other hand, many sampling-based planners [13], [12] or TAMP algorithms [4], [9] only need a binary predicate on whether the constraint is satisfied.

Using the proposed object representation consist of semantic 3D keypoints and dense geometry, the key benefit is that the motion planner can handle a category of objects without instance-wise tuning. In the following text, we discuss several important costs and constraints that are related to the object representation.

Object Target Configuration Let p_j^t be the keypoints of the object o_j at the time t, where $1 \leq t \leq T$. The target configuration of an object o_j can be represented as a set of costs and constraints on its semantic keypoint p_j^T, where T is the terminal time knot. For instance, to place the mug at some target location p_{target} as illustrated in Fig. 1, we need an equality constraint

$$||p_{\text{bottom\_center}}^T - p_{\text{target}}|| = 0 \tag{5}$$

where $p_{\text{bottom\_center}}^T$ is the mug bottom-center keypoint expressed at time T. Note that this constraint can handle mugs with different size, shape and topology. Many other costs and constraints can be used to specify the object target configuration. Please refer to kPAM [17] for more details.

Collision Avoidance The dense geometry information from shape completion can be used to ensure the planned trajectory is collision-free. Specifically, let B_r denote the set of rigid bodies of the objects, robot and environment, where the geometry of objects are obtained using shape completion. We need to ensure

$$\text{signed\_distance}(X^t; b_i, b_j) \geq \delta_{\text{safe}} \tag{6}$$

$$\text{for } b_i \in B_r,\ b_j \in B_r,\ i \neq j,\ 1 \leq t \leq T \tag{7}$$

where δ_{safe} is a threshold, $\text{signed\_distance}(X^t; b_i, b_j)$ is the signed distance [24] between the pair of rigid body (b_i, b_j) at the configuration X^t. Practically, it is usually unnecessary to check the collision of every rigid body pairs, as most rigid bodies except the grasped object and robot end-effector have limited movement.

Geometric Predicates In many planning algorithms [8], [27], geometric predicates are used to model the geometric relationship between the objects and the environment. Although these predicates are typically proposed in the context of known objects with geometric templates, they can benefit from shape completion and naturally generalize to a category of objects. Here we summarize several examples used in these manipulation planners.

- The static stability constraint enforces that the object placement surfaces are aligned with one of the environment placement regions. To use this predicate, it is required to extract the surfaces on the object that afford placing from the object dense geometry. Please refer to [8] for more details.
- The visibility constraint requires the line segments from the sensor to the object are not blocked by other objects or the robot. In other words, the manipulate object should not block or be blocked by other objects.
- The containment constraint enforces the convex hull of an object is included in a container. This constraint needs the convex hull of the object, which can be computed using the dense geometry from shape completion.

C. Grasping

The grasp planning module is responsible to compute a grasp pose that allows the robot to stably pick up and transfer the object. Various algorithms [19], [6], [16] have been proposed for grasp planning. Some of them [29], [14], [31] are built upon shape completion and can be easily integrated into our pipeline. These algorithms are object-agnostic and can robustly generalize to novel instances within a given category.

V. Results

In this section, we demonstrate a variety of manipulation tasks using our pipeline. The particular novelty of these demonstrations is that our method can automatically plan robot trajectories that handle large intra-category variations without any instance-wise tuning or specification. The video demo and source code are available on our this link.

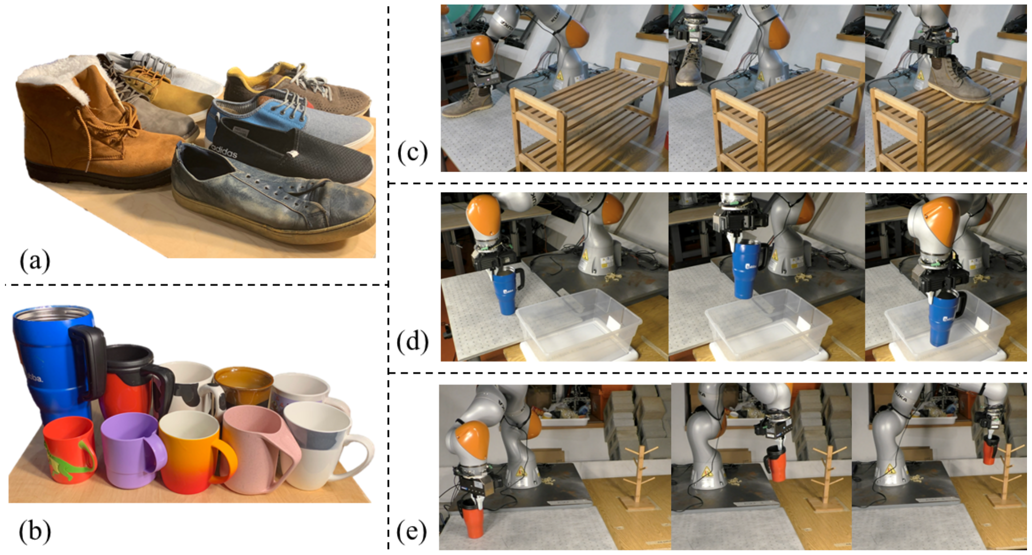

Fig. 3: An overview of our experiments. (a) and (b) are the shoes and mugs used to test the manipulation pipeline. Note that both the shoes and mugs contain substantial intra-category shape variation. We use three manipulation tasks to evaluate our method: (c) put shoes on a shelf; (d) put mugs in a container; (e) hang mugs on a rack by the mug handles. Visit this link to watch the video demo.

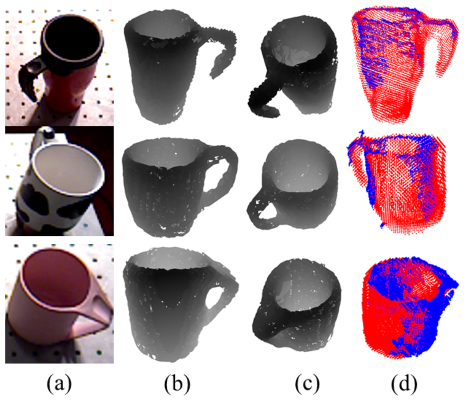

Fig. 4: The qualitative results for shape completion. (a) is the input RGB image. (b) and (c) are the dense geometry from shape completion in two viewing directions. (d) compares the point cloud from depth image and shape completion: the depth image point cloud is in blue while shape completion is in red. Although the completed geometry contains small holes and defects, the accuracy is sufficient for many manipulation tasks.

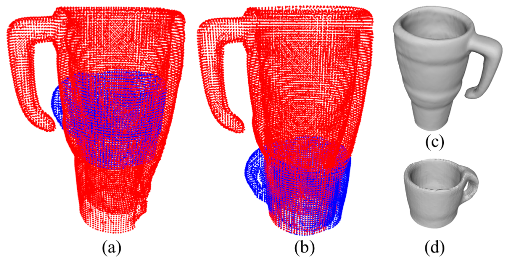

Fig. 5: Pose estimation can lead to multiple ambiguous alignment. (a) and (b) show two alignment results by [3] (variation on the random seed), where we attempt to register the mug template (d) into the observation (c). Using these pose estimation results for manipulation planning can lead to undetected physically infeasibility, as shown in Sec. V-E.

A. Experiment Setup and Implementation Details

We use a 7-DOF Kuka IIWA arm mounted with a RGBD sensor. Both the intrinsic and extrinsic of the camera are calibrated. We use 8 shoes and 10 mugs to test the manipulation pipeline, as shown in Fig. 3. Note that both shoes and mugs have substantial intra-category shape variation. Keypoints same as kPAM [17] are used to define the target configuration of the shoe and mug. Using the automatic data collection in Sec. IV-A, we collect 41000 images to train the shoe networks and 70100 network for mug networks.

The drake library [26], which provides optimization based motion planners, is used for manipulation planning. The costs and constraints include object target configuration and collision avoidance. For this work we use a fixed contact-mode sequence consists of picking, transferring and placing of the object, as the scheduling of contact-modes is not our focus. We emphasize that the proposed object representation and manipulation framework are agnostic to the concrete planner that solves (2). Many motion planners, either optimization-based methods [24], [27] or sampling-based approaches [4], [25], can be plugged in and used.

B. Perception and Comparison with Pose Estimation

In this subsection, we provide results of shape completion and compare it with the widely-used 6-DOF pose representation. Fig. 4 shows several completed dense geometry for representative mugs. The network takes input from images in Fig. 4 (a) and produces the dense geometries in Fig. 4 (b) and (c). Fig. 4 (d) compares the point cloud from depth images and shape completion. Although the completed geometry contains small holes and defects, the accuracy is sufficient for many manipulation tasks. Note that the network generalizes to instances with substantial shape variations.

TABLE I: Robot Experiment Statistics

Task	#Trails	#Failure	#Planning Failure	#Grasp Failure	#Execution Failure
shoe2shelf	50	14	12	2	0
mug2container	50	10	9	0	1
mug2rack	50	15	10	0	5

TABLE II: False-Negative (Overly Confident, see Sec. V-E) Statistics

Manipulation Pipeline	#Trails	#False Negative (Overly Confident)	False Negative Rate
6-DoF Pose	50	19	38%
kPAM	50	9	18%
Ours	50	1	2%

On the contrary, pose estimation can be ambiguous and fail to capture large shape variations. An illustration is provided in Fig. 5. where the pose estimator in [3] is used to align two mugs. Using dense geometry from pose estimation can lead to undetected physically infeasibility, as shown in Sec. V-E.

C. Manipulation Task Specifications

1) Put shoes on a shelf: The first manipulation task is to put shoes on a shoe shelf, as shown in Fig. 3 (c). The shoe should be placed in alignment with the shelf. The robot needs to deal with shoes with different appearance and geometry.

2) Place mugs into a container: The second manipulation task is to place mugs into a box without colliding with it, as shown in Fig. 3 (d). The mug should be upright and its handle should be aligned with the container.

3) Hang mugs on a rack: The last manipulation task is to hang mugs onto a mug rack by the mug handle, as shown in Fig. 3 (e). The geometry and position of the mug rack are available to the robot.

Note that although task 1) and 3) have been performed in our previous work kPAM [17], it uses various manually-specified robot trajectories. This manual specification has two major limitations: 1) it is labor-intensive and can hardly scale to more complex environments and manipulation tasks; 2) the pipeline tends to be overly confident, as it cannot detect physical infeasibility without dense geometry. A comparison is made in Sec. V-E.

D. Result and Failure Mode

The video demo is on this link. The statistics about three different tasks is summarized in Table. I. Most failure cases result from the failure of the motion planner. Since the motion planner used in this work uses non-convex optimization internally, it can be trapped in bad local minima without a good initialization. This issue can be resolved by using sampling-based planners such as RRT. These planners are globally optimal but need longer running time.

The grasp failure in Table. I means (1) the robot fails to grasp the object; or (2) the relative motion between the gripper and the grasped object is too large. The relative motion may occur during the grasping, or when the object is not rigid. This problem could be alleviated by the addition of an external camera that would allow us to re-perceive the object after grasping. The execution failure in Table. I refers to the situations such that (1) the robot makes collision (despite the perception and planning are successful); or (2) the object is not placed into the target configuration. This problem necessitates the execution monitoring described in [2].

E. Comparison with Alternative Pipelines

In this subsection, we compare our method with two alternative manipulation pipelines: 1) a manipulation pipeline based on 6-DOF pose with a geometric template; 2) the original kPAM [17] pipeline. For the 6-DOF pose based manipulation pipeline, we use the same pose estimator as the baseline (Fig. 4) in kPAM [17]: first initialize the alignment with detected keypoints, then perform ICP fitting between the observed point cloud and geometric template to get the 6-DOF pose.

Without an accurate characterization of the dense geometry, these alternatives suffer from false-negative (overly-confidence): the resulted robot trajectory can be physically infeasible even if the pipeline claims both the perception and planning succeed. Note that this false-negative is much more adversarial than the planning failure in Table. I (Sec. V-D). When the planner fails the pipeline would be stopped and it is still safe. On the contrary, when false-negative happens the unsafe trajectory would be sent to the robot for execution, which is rather dangerous.

Table. II summarizes the false-negative rate of all three methods on the "mug2container" task. We mark a trail as "false-negative" if the pipeline claims perception and planning succeeds, but the resulted trajectory leads to a collision. The false-negative rates of the two alternatives are much higher than our method, which implies our method is much safer. This highlights the benefit of accurate characterization of dense geometry and the integration of shape completion into the manipulation pipeline.

VI. Conclusion

In this paper, we focus on manipulation planning of a category of objects, where the robot should move a set of objects to their target configuration while satisfying physical feasibility. This is challenging for existing works as they assume known object templates and 6-DOF pose estimation, which doesn't generalize of novel instances within the category. Thus, we propose a new hybrid object representation consists of semantic keypoints and dense geometry as the interface between the perception module and planning module. Systematically, we contribute a novel integration of shape completion with keypoint detector and manipulation planner. In this way, both the perception and planning module generalizes to novel instance. Extensive hardware experiments demonstrate the effectiveness of our method.

References

[1] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao. Perception, planning, and execution for mobile manipulation in unstructured environments. *IEEE Robotics and Automation Magazine, Special Issue on Mobile Manipulation*, 19(2):58–71, 2012.

[2] R. Diankov. Automated construction of robotic manipulation programs. 2010.

[3] W. Gao and R. Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. *arXiv preprint arXiv:1811.10136*, 2018.

[4] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling. Sampling-base methods for factored task and motion planning. *The International Journal of Robotics Research*, 37(13-14):1796–1825, 2018.

[5] M. Gualtieri, A. ten Pas, and R. Platt. Pick and place without geometric object models. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7433–7440. IEEE, 2018.

[6] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt. High precision grasp pose detection in dense clutter. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 598–605. IEEE, 2016.

[7] K. Hauser. Task planning with continuous actions and nondeterministic motion planning queries. 2010.

[8] J. A. Haustein, K. Hang, J. Stork, and D. Kragic. Object placement planning and optimization for robot manipulators. *arXiv preprint arXiv:1907.02555*, 2019.

[9] L. P. Kaelbling and T. Lozano-Pérez. Hierarchical planning in the now. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[10] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[11] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wüthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg. Real-time perception meets reactive motion generation. *IEEE Robotics and Automation Letters*, 3(3):1864–1871, 2018.

[12] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[13] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[14] J. Lundell, F. Verdoja, and V. Kyrki. Robust grasp planning over uncertain shape completions. *arXiv preprint arXiv:1903.00645*, 2019.

[15] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[16] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1957–1964. IEEE, 2016.

[17] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019.

[18] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[19] D. Morrison, P. Corke, and J. Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *arXiv preprint arXiv:1804.05172*, 2018.

[20] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[21] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *arXiv preprint arXiv:1901.05103*, 2019.

[22] A. Price, L. Jin, and D. Berenson. Inferring occluded geometry improves performance when retrieving an object from dense clutter. *arXiv preprint arXiv:1907.08770*, 2019.

[23] C. Sahin and T.-K. Kim. Category-level 6d object pose recovery in depth images. In *European Conference on Computer Vision*, pages 665–681. Springer, 2018.

[24] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. Citeseer.

[25] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 639–646. IEEE, 2014.

[26] R. Tedrake and the Drake Development Team. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems, 2016.

[27] M. Toussaint, K. Allen, K. A. Smith, and J. B. Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning.

[28] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans. Learning continuous 3d reconstructions for geometrically aware grasping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11516–11522. IEEE, 2020.

[29] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2442–2447. IEEE, 2017.

[30] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. *arXiv preprint arXiv:1901.02970*, 2019.

[31] D. Watkins-Valls, J. Varley, and P. Allen. Multi-modal geometric learning for grasping and manipulation. *arXiv preprint arXiv:1803.07671*, 2018.

[32] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[33] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383. IEEE, 2017.

[34] X. Zhang, Z. Zhang, C. Zhang, J. Tenenbaum, B. Freeman, and J. Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems*, pages 2257–2268, 2018.