Task-driven Perception and Manipulation for Constrained Placement with No Shape Priors

Chaitanya Mitash, Rahul Shome, Bowen Wen, Abdeslam Boularias and Kostas Bekris

***Abstract*— Recent progress in robotic manipulation has dealt with the case of no prior object models in the context of relatively simple tasks, such as bin-picking. Existing methods for more constrained problems, however, such as deliberate placement in a tight region, depend more critically on shape information to achieve safe execution. This work introduces a possibilistic object representation for solving constrained placement tasks without shape priors. A perception method is proposed to track and update the object representation during motion execution, which respects physical and geometric constraints. The method operates directly over sensor data, modeling the seen and unseen parts of the object given observations. It results in a dynamically updated conservative representation, which can be used to plan safe manipulation actions. This task-driven perception process is integrated with manipulation task planning architecture for a dual-arm manipulator to discover efficient solutions for the constrained placement task with minimal sensing. The planning process can make use of handoff operations when necessary for safe placement given the conservative representation. The pipeline is evaluated with data from over 240 real-world experiments involving constrained placement of various unknown objects using a dual-arm manipulator. While straightforward pick-sense-and-place architectures frequently fail to solve these problems, the proposed integrated pipeline achieves more than 95% success and faster execution times.**

I. Introduction

Object placement in tight spaces is a challenging problem often encountered in robotic manipulation. It corresponds to a task where constraints on the placement pose of the object are imposed. It occurs, for instance, in logistics applications, such as packing objects in boxes or packages for shipping. It also appears in service robotics where, for example, a book needs to be placed on a bookshelf and inserted in a small space amongst other books.

Some of the recent work has focused on variants of this problem, such as the bin-packing problem [1], [2] and table-top placement in clutter [3]. Nevertheless, a geometric 3D model for the manipulated object is assumed to be known. This assumption is invalid in several scenarios due to the wide variety of objects to be manipulated as well as the time and effort required for obtaining the shape models. Some recent robotic manipulation pipelines [4], [5] have shown the capacity of picking novel and previously unseen objects from clutter. These methods, however, assume no constraints on the object placement pose. Therefore, the object is grasped with any feasible and stable grasp without reasoning about placement. Some alternatives consider placement constraints and do not require exact models of objects but operate with category-level prior information. Examples include an approach based on sparse keypoint representations [6] and deep reinforcement learning [7]. While these approaches are good to guide manipulation planning solutions, the employed representations do not allow safe and generalizable manipulation planning as no deterministic geometric or physical constraints are considered. Moreover, task-relevant features may not be easy to define based only on the category of the manipulated object.

The authors are with the Computer Science Department of Rutgers University in Piscataway, New Jersey, 08854, USA. Email: {cm1074, rs1123, bw344, ab1544, kb572}@rutgers.edu

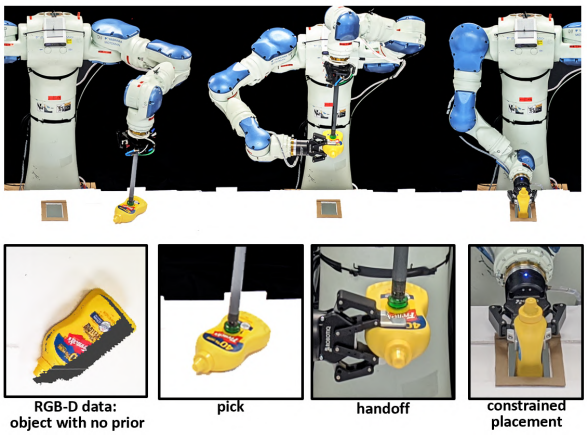

Fig. 1. A pick-handoff-place solution computed by the proposed pipeline for inserting an object in a constrained space. The object is previously unknown. The approach reasons about its shape before manipulating it to fit it into the constrained space, where only few object poses succeed.

This work develops both a representation and a framework that are *general* and can be easily used with task specifications that assume no priors, such as dialogue and deictic expressions in the form of "Put that there" [8]. It should also be *flexible* to allow the incorporation of additional semantic constraints, such as "Put the mugs upright on the shelf" [6]. Thus, in order to deal with the many real-world scenarios where object priors are not available, this work:

- Proposes a *possibilistic* representation to deal with shape uncertainty within a *manipulation planning pipeline*. The representation models the unobserved part of the object to ensure computation of safe manipulation actions when no prior object information is provided. The representation can be tracked and updated while maintaining geometric and physical consistency.
- Develops a robotic system for picking an object (with no priors) from a table-top and placing it in a constrained space as in Fig. 1. The system comprises of a dual-arm manipulator with a single RGB-D sensor as well as a vacuum-based and an adaptive, finger-based hand.

Preprint submitted to IEEE Robotics and Automation Letters as Submission for RA-L and IROS
Received February 24, 2020 23:44:36 PST

- Demonstrates the use of *handoffs* for transferring objects between two arms in a real-world setup. Handoffs allow regrasping and more flexibility in solving constrained placement as the initial pick may not directly allow a placement. A handoff is a challenging constrained placement task by itself as the object is inserted in the confined space between the robotic hand's fingers.
- Proposes a *task-driven* perception and manipulation planning pipeline that exploits the strength of the proposed object representation for:
  1. computing safe to execute manipulation actions (picks, handoffs and placements) even without priors;
  2. planning high-quality action sequences, which may include handoffs and re-sensing, given the representation;
  3. minimizing the number of sensing actions needed to complete the placement task; and
  4. performing visual tracking and closed-loop execution to counter the effects of stochastic in-hand motions that often result from unmodeled physical forces, such as gravity, inertia and grasping contacts.

Over 240 real-world manipulation experiments are performed to study and evaluate the performance of the proposed system. The experiments demonstrate that the proposed pipeline is robust in handling objects with no shape priors within the limitations of the end-effector and the sensor. It achieves a task success rate of 95.82%, which is much higher than straightforward pick-sense-and-place alternatives, while reducing requirements for sensing operations and achieving faster execution times.

II. Related Work

Instance-level object manipulation: Most existing efforts in robotic manipulation operate over the assumption that an exact, 3D geometric model of the manipulated object is known in advance. The pose corresponding to the object model [9], [10] is perceived in the scene and manipulation actions [11], [12] are performed on the object to move it from source pose to a pre-specified target pose. The assumption of known geometric models may not always be true when there are a wide variety of objects and obtaining models for each of them is expensive.

Category-level object manipulation: In some scenarios, prior knowledge about the objects might be available at the category-level. There has been recent progress in category-level pose estimation [13], [14], [15], [16], [17] but given large intra-class shape variation in certain scenarios, it is hard to capture the shape in a single category-level pose representation. This often leads to planning manipulation actions that end up in physically-unrealistic configuration for certain object instances. A recent effort [6] proposed using semantic keypoints as category-level object representation for planning pick-and-place manipulation actions. While the representation successfully addresses the issue of intra-class variation, it does not have the dense geometric information needed for safe manipulation. Another recent work, [7] performs pick and place on objects by training an end-to-end deep reinforcement learning framework within the task context. Given that it is hard to interpret the learnt policies, it is not clear how the policies learnt with rewards coming from a specific task can be generalized to other similar tasks, configurations and objects.

Task-agnostic grasping: There has also been considerable effort and success in learning grasp quality metrics [18], [19], [20], [21], [22] to generate robust grasps given partial observations of novel objects. This, however, does not take into account the task context and can often not even provide enough choices for robust grasps required for the purpose of manipulation planning.

Placement reasoning: Most manipulation pipelines for novel objects do not address the problem of constrained placement [4], [5]. Some related efforts learn the stable placements of objects by defining features on pointclouds and learning a placement score function [23]. Alternatively, the problem of motion planning for placing grasped objects in clutter typically requires know geometric models [3]. Few efforts consider both picking and constrained placement for novel objects.

Shape reconstruction and shape completion: Some works reconstruct 3D models of in-hand objects by focusing on object tracking and video segmentation respectively [24], [25]. Such reconstructions aim to generate good quality object models as opposed to this work, which performs good-enough models to solve the current task. Given the recent progress in learning-based shape completion [26], [27], [28], this tools is being increasingly used in the context of manipulation [29], [30]. Nevertheless, such techniques typically require access to prior knowledge to complete the object's shape and the output is often too noisy for manipulation planning in constrained places.

III. Problem Setup and Notation

Rigid object: A rigid object geometry is defined by a region occupied by the object $O^* \subset \mathbb{R}^3$ in its local reference frame that defines its shape. Given a pose $P \in SE(3)$, the region occupied by the object at P is denoted by O^*_P.

Constrained placement: Given an object at an initial pose $P_{\text{init}} \in SE(3)$, the goal of the constrained placement problem is to transfer the object O^* to a pose $P_{\text{target}} \in SE(3)$, such that $O^*_{\text{P}_{\text{target}}} \subset R_{\text{place}}$ where $R_{\text{place}} \subset \mathbb{R}^3$ is the target placement region.

Manipulation actions: The robotic arms utilize end-effectors to alter the pose of objects once they are grasped. The scope of this work is limited to prehensile manipulation including the following actions:

a) *Pick:* A motion that ends at a configuration that allows the end-effector to attach to and immobilize the object.

b) *Placement:* A motion that ends with releasing the object once it attains a desired pose.

c) *Handoff:* Involving two arms, a handoff comprises of a simultaneous *pick* by an arm, and *placement* by another.

Manipulation planning for constrained placement: Computing a sequence of manipulation actions that can move the object O^* from P_{init} to a target pose P_{target}, which successfully solves a constrained placement task.

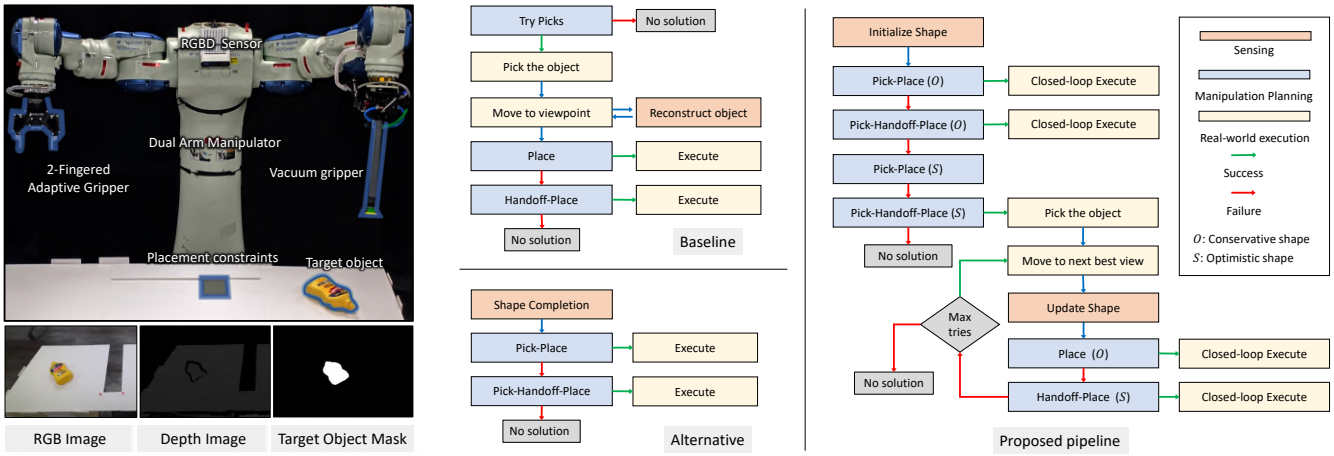

Fig. 2. System components (left-top), alternative (center) and proposed pipelines (right) for the placement of a target object within a constrained region. The problem takes as input (left-bottom) RGB-D sensor data and a 2d segmentation mask of the target object.

Such a solution consists of motions of the arms denoted by Π, which is parametrized by the time of the motions. $\Pi(0)$ is the initial arm configuration, and $\Pi(1)$ has an arm placing the object at P_{target}.

Perception actions: It should be noted that no prior knowledge is assumed about the object to be manipulated, i.e., O^* is unknown. Thus, an object representation O is defined over which manipulation planning can operate. In general $O \neq O^*$. O will typically be derived from an initial view of the object. This is typically incomplete, and might not be sufficient to solve the constrained placement problem. This necessitates a separate class of actions, namely perception actions, which can update the object representation O with additional sensing information.

The key issue relates to how O is estimated and used. For instance, given the initial view, when object models are available, object pose estimation can provide an O (and P_{init}) that is close to the entire geometry of $O^*_{P_{\text{init}}}$. Without models, shape completion methods try to estimate O^* as well. The shape representation needs to be accurate enough to estimate the occupied volume of the object, in order to facilitate a successful solution to the planning problem.

Now it is possible to describe the overall problem, which combines perception and manipulation actions.

Task-driven Perception and Manipulation Planning: Computing a sequence of perception and manipulation actions that are necessary for the successful completion of a constrained placement task.

Let P_{final} be the resultant pose of the object after executing the sequence of actions returned by *task-driven perception and manipulation planning*. In general $P_{\text{final}} \neq P_{\text{target}}$. The constrained placement task fails if the true object geometry O^* at P_{final} is not within R_{place}. This could be a result of:
i) errors in the computation of P_{target} due to limitations of the representation O w.r.t. O^*;
ii) errors in execution leading to $P_{\text{final}} \neq P_{\text{target}}$.

There is a need for an object representation O that is accurate enough w.r.t. O^* and guarantees a solution that can be tracked and updated to ensure P_{final} is close enough to the correct P_{target}, such that $O^*_{P_{\text{final}}} \in R_{\text{place}}$.

IV. System Design and Baseline

Fig. 2 (left) shows the hardware setup used in this work. The robot being used is a dual-arm manipulator (*Yaskawa Motoman SDA10f*). The current work allows the control of its two 7 $d.o.f$ arms C_{l} and C_{r}. The left arm of the robot is fitted with a narrow, cylindrical end-effector with a vacuum gripper attached to the end; and the right arm is fitted with a Robotiq 2-fingered adaptive gripper. A single RGB-D sensor (Microsoft Azure Kinect) is mounted on the robot overlooking both the picking and the placement regions. The sensor is configured in the Wide-FOV mode to capture images at 720p resolution with a frequency of up to 20Hz. Fig. 2 (center) illustrates two alternative pipelines and Fig. 2 (right) highlights the proposed pipeline.

Baseline - Complete Shape Reconstruction: The baseline is designed to pick the object with a task-agnostic pick (i.e., any pick that works) and then reconstruct the entire object by moving to pre-defined viewpoints. After the reconstruction, manipulation planning is performed on the reconstructed shape to find and execute a solution for constrained placement.

A drawback of this approach is that *committing to a task-agnostic pick* might preclude solutions, which might have been possible with a different pick. For instance, the initial pick might not allow a direct placement or in some cases even obstruct handoffs. Another drawback is that the amount of object reconstruction required depends on the task. It can be inefficient to fully reconstruct the object if a *robust solution with partial information* can be found. Finally, even with a large number of perception actions, some parts of the objects might be missing, which can still lead to execution failures. For instance, this can happen if say the bottom surface is not reconstructed and fingered grasps interact with the unmodeled part of the object during execution.

Alternative - Shape completion: An alternative is to perform shape completion to fill in the part of the object that is occluded from a single sensor. This is typically performed via learning priors on objects or categories or via assumptions, such as object symmetry. Picks and placements are computed over this completed shape. Any action computed over such

an output *cannot guarantee safe execution*, since there are no guarantees on the parts of the shape being completed. For prehensile manipulation and constrained placement, any noise in shape completion can lead to collisions.

Even if the computed plans work in simulation for either of the alternatives, there might be execution failures due to unmodeled motions of object, such as within-hand motion (which violate the prehensile assumption). Thus, a *closed-loop execution* that would keep track of the object during manipulation and perform online adjustments is essential.

V. Proposed Pipeline

This section outlines the primary contributions of the current work: a) **a shape representation**, b) **an online pose and shape tracking method** that can leverage the shape representation, and c) **a pipeline** (Fig 2, right), which composes different modules to solve the constrained placement task. The design choices made in the pipeline address the issues highlighted in the above pipelines. The proposed pipeline begins with initializing an object shape representation.

Object Shape Representation: An object is represented as a point set O that consists of two mutually exclusive sets of points $\mathcal{S}$ and $\mathcal{U}$ in $\mathbb{R}^3$. $\mathcal{S}$ is a set of *seen* points on the surface of the object that are observed by the RGB-D sensor. $\mathcal{U}$ is a set of *unseen* points in space that have not been observed by the sensor given its observations but *have a non-zero probability of belonging to the target object*. Thus $O = \mathcal{S} \cup \mathcal{U}$, where, $\mathcal{S} \cap \mathcal{U} = \phi$. The conservative representation ensures that the optimal object shape $O^* \subseteq O$.

Given the input RGB-D images $(I_{\text{rgb}}, I_{\text{depth}})$ and the target object mask T_{mask}, the object representation O is initialized with it's origin $\mathbf{0}^O$ at the centroid of the 3D segment corresponding to T_{mask} and the reference frame at identity rotation with respect to the camera frame. A voxel grid is initialized at $\mathbf{0}^O$ and each voxel is classified as either 1) *observed and occupied* $\mathcal{S}$, 2) *unobserved* $\mathcal{U}$, or 3) observed and unoccupied, i.e., empty voxels that are implicitly modeled as a set of points $\{p \in \mathbb{R}^3 \mid p \notin \mathcal{S} \cup \mathcal{U}, \|p - \mathbf{0}^O\| < D_{\text{max}}\}$, for a maximum dimension parameter $D_{\text{max}} = 30cm$.

Grasp computation: Grasp sets $\mathcal{G}_{\text{l}}$ and $\mathcal{G}_{\text{r}}$ are computed over the object shape representation O by ensuring stable geometric interaction with the observed part of the object $\mathcal{S}$ and being collision-free with *both* $\mathcal{S}$ *and* $\mathcal{U}$, thereby ensuring *safe* and successful execution. It is also crucial for the success of manipulation planning to have large, diverse grasp sets at its disposal. This is distinct from the typical objective of grasp generation modules that primarily focus on the quality of the top (few) returned grasps. For instance, in Fig. 3 top-right, the grasps are spread out over O with different approach directions, which provide options to manipulation planning and aid solution discovery.

Vacuum grasps $\mathcal{G}_{\text{l}}$ are computed by uniformly sampling pick points and their normals from $\mathcal{S}$, and ranked in quality by their distance from the shape centroid. The grasp set $\mathcal{G}_{\text{r}}$ for the fingered gripper samples a large set of grasps over O according to prior work [19]. Sampled grasps are then pushed forward along the grasp approach direction until the fingers collide with either points from $\mathcal{S}$ or $\mathcal{U}$, and ranked by the alignment between the finger and contact region on $\mathcal{S}$.

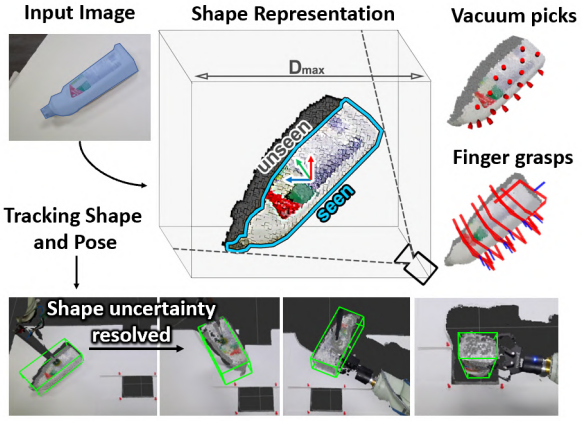

Fig. 3. The object is represented by a set of occupied, unseen and empty voxels. This representation is tracked during manipulation such that it can be updated based on new viewpoints to reduce the uncertainty in shape while maintaining the physical consistency of the representation such as the empty voxels in the object frame can never be occupied and unseen voxels can only reduce in size by being replaced by occupied or empty voxels.

Placement Computation: Given the placement region R_{place}, and the object representation O, two boxes are computed, 1) the maximum volume box B_{place} within R_{place} and 2) the minimal volume box B_{O} that encloses O. Candidate placement poses (P_{target}) correspond to configurations of B_{O} which fit within B_{place}. A discrete set (24) of configurations for the box is computed by placing B_{O} at the center of B_{place} and validating all axis-aligned rotations. The total set of poses returned is $\mathcal{P}_{\text{place}}$, such that any pose in it is a candidate P_{target}.

Manipulation Planning: The input to manipulation planning is the estimated object representation O, the grasp sets available for both arms, $\mathcal{G}_{\text{l}}, \mathcal{G}_{\text{r}}$, and the placement poses $\mathcal{P}_{\text{place}}$. Manipulation planning returns a sequence of prehensile manipulation actions that ensure a collision free movement (Π) of the arms and O such that the object is transferred from P_{init} to some $P_{\text{target}} \in \mathcal{P}_{\text{place}}$. In the absence of any errors, the execution of these actions solves the constrained placement task.

As a part of the task planning framework, a probabilistic roadmap [31] consisting of 5000 nodes is constructed using the PRM* algorithm [32], for each of the arms. The grasps and placements for each arm can be attained by corresponding grasping, and placement configurations of the arms, obtained using *Inverse Kinematics* solvers. Beginning with the initial configuration of the arms, the high-level task planning problem becomes a search over a sequence of the manipulation actions, achievable by the *pick, place or handoff* configurations. This is described in the form of a forward search tree [33] which operates over the same roadmap [34] by invalidating edges (motions) that collide with the object, or the other arm. The search tree is further focused by only expanding *pick-place* and *pick-handoff-place* action sequences. Each such sequence can be achieved through a

combination of different choices of *grasping, handoff, and placement* configurations. The search traverses the set of options for grasps in the descending order or quality, and returns the first discovered solution that successfully achieves a valid target placement ($P_{\text{target}} \in \mathcal{P}_{\text{place}}$).

Composition into Pipeline: As in Fig 2(right) manipulation planning is first performed over the grasps and placements computed on the conservative estimate of the shape ($O = \mathcal{S} \cup \mathcal{U}$). If the planning fails to find a solution, grasps and placements are re-computed over an optimistic shape ($\mathcal{S}$). If no placements are achievable at this point, the problem is *not solvable* since $\mathcal{S} \subset O \subset O^*$. The solution with the optimistic shape provides the immediate picking action. After the pick, the object is moved to the *next best view* to update the conservative shape estimate O with a manipulation planning attempt for the updated shape. The solution is then executed in a *closed-loop* fashion using tracking. The proposed pipeline benefits from the fact that manipulation planning could be performed early in the process *without compromising on the safety of execution*. This also allows it to solve the problem *with minimal perception actions*.

The following modules describe the steps involved in tracking, shape-update and closed-loop execution.

Shape and Pose Tracking: The object pose P^t changes over time with the gripper manipulating it, where $E^t \in SE(3)$ denotes the gripper pose at time t. Between consecutive timestamps for a perfect prehensile manipulation, $\Delta P^{t-1:t} = \Delta E^{t-1:t}$ which is the change in the gripper's pose. Tracking is introduced to account for non-prehensile within-hand motions which violates this nicety.

The object segment at any time s^t is computed from a) points lying in a pre-defined region of interest ($D_{\max}$) in the reference frame of the gripper, and b) by eliminating the points corresponding to the gripper's known model. An update to the pose of the object, $\Delta P^{t-1:t}$ is computed (and updated) in three steps:

1) Assuming rigid attachment of the object with the end-effector, the transformation, $\Delta E^{t-1:t}$ is applied to the object segment in previous frame s^{t-1} to obtain the expected object segment at time t, s'^t.

2) To account for any within hand motion of the object, a transformation is computed between s'^t and the observation s^t via ICP. While $\Delta P^{t-1:t} = \Delta E^{t-1:t} * \Delta P_{\text{ICP}}$ provides a good estimate of relative pose between consecutive frames, accumulating such transforms over time can cause drift.

3) A final pointset registration process is utilized to locally refine the pose. An ICP registration step with a strict correspondence threshold is performed between the object representation (O) at pose $P^t = P^{t-1} \cdot \Delta P^{t-1:t}$, and the current observation s^t. The resulting transformation is applied to $\Delta P^{t-1:t}$, and correspondingly P^t.

During manipulation, when a new viewpoint is encountered, the output of pose tracking is utilized to update the object's shape which assists tracking in future frames.

Update Shape: The object shape is updated with every new viewpoint. As mentioned above this might be necessary to reduce the conservative estimate of the shape and help with pose tracking. For the case when the update is invoked as a perception action, the first step is to compute the *next best view* amongst a set of discrete viewpoints that exposes the most number of unobserved voxels (in $\mathcal{U}$). In the implementation rotations about the global *Z-axis* are evaluated. This is found by rendering $\mathcal{S}$ at each of the viewpoints and computing the count of $\mathcal{U}$ that are unoccluded in the renderings. This viewpoint is most-likely to reduce the conservative volume of the object. The object is then moved to this viewpoint and O is updated.

The size of the set O (and thus the conservative volume) is the largest at initialization. Any update to O either removes a point $p \in \mathcal{U}$ (if it is observed to be empty) or p can be moved from $\mathcal{U}$ to $\mathcal{S}$. To update O, the observed segment s^t is transformed to the object's local frame based on the pose P^t. For each point p on the transformed point cloud, its nearest neighbor $p^{\mathcal{S}} \in \mathcal{S}$ and $p^{\mathcal{U}} \in \mathcal{U}$ are found. If $\mid p^{\mathcal{S}} - p \mid < \delta_c$ where δ_c is the correspondence threshold, p is considered to be already present. Otherwise, if $\mid p^{\mathcal{U}} - p \mid < \delta_c$, $p^{\mathcal{U}}$ is removed from $\mathcal{U}$ and added to $\mathcal{S}$. Finally all points in $\mathcal{U}_{P_t}$ are iterated over to remove points in $\mathcal{U}$ and thus in O which belong to the empty part of space based on the currently observed depth image. Applying these physical and geometric constraints in the update process significantly reduces the drift that occurs in simultaneous updates to the object's pose and shape.

Closed Loop Execution: Given a manipulation planning solution describing the motions (Π) of the arms, and the object over time, it is the objective of closed-loop execution to ensure that any errors in execution or non-prehensile grasping interactions are adjusted for. At any time t, $\Pi(t)$ describes how the arms are configured, and assuming prehensile grasps, the object pose P^{t*}. Tracking returns the current estimate P^t. If $P^t \neq P^{t*}$ the remainder of the motion has to be adjusted to account for $\Delta P = P^{t*} - P^t$. Large reported errors in terms of ΔP might need re-planning, which is not part of the current work. In our implementation this adjustment is performed before *handoffs*, and *placements* by doing local changes to Π.

Fig. 4. (left) Change in grasp as the object is being picked due to the center of mass of the object being away from the pick point. This combined with other nuances of real-world manipulation, such as unmodeled contact parameters during handoff leads to the situation where the object does not reach the planned placement pose. Thus adjustment is made based on the closed loop execution module (right) before handoffs and placement.

VI. Evaluation

More than **240** real-robot experiments are performed using different unknown objects and placement constraints to assess the proposed system. The experimental setup and results are reported in the following section.

Design Choices: The experiment setup design was carefully chosen to fairly evaluate the efficacy of the current contributions - the shape representation and tracking, and robust pipeline that leverages the two.

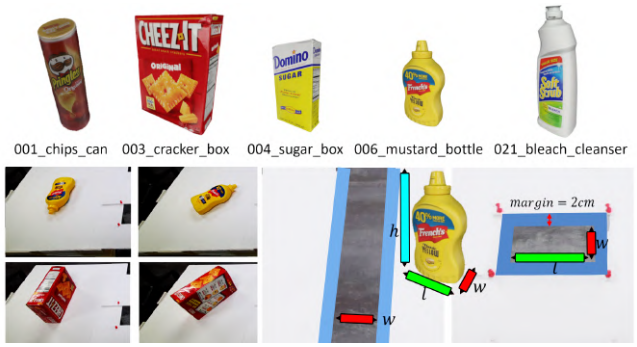

Fig. 5. Objects used in the experiments (top). Examples of initial configurations (left). Examples of placement constraints (right).

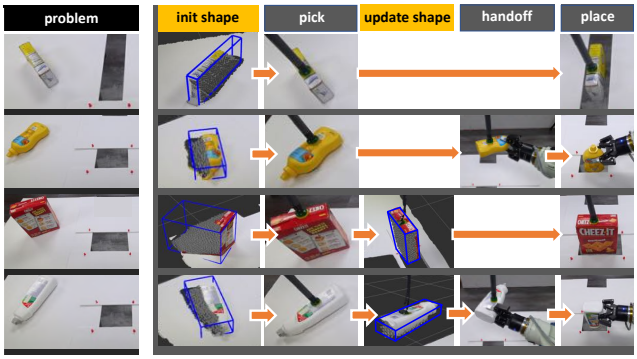

Fig. 6. Qualitative results indicating different solution modes of the proposed pipeline.

A dual-arm manipulator described in Fig 2(left) provides a powerful manipulation task-solving capability via a rich set of actions, different end-effectors, handoff interactions, extended reachability etc.

Objects: Experiments are performed over 5 YCB [35] objects (Fig. 5) of different shapes and sizes. It should be noted that *no models* are made available to the method.

The objects are placed on a table-top scene in front of the left arm (vacuum gripper), with the target placement region centrally aligned in front of the robot, reachable by both arms. The constrained placement solutions here can therefore involve a direct placement by the left arm, or a handoff-placement with the right arm. Different initial configurations of the object will affect the nature of the task planning solution by virtue of a) different available initial picks, and b) different conservative shape representation based on how much of the object is unseen at the configuration.

Initial Configuration: For each stable resting pose of the object in front of the left arm, rotations were uniformly sampled along the axis perpendicular to the plane of the table. Configurations with limited reachable grasps are ignored.

Placement Region: An opening is created on the table surface where the object needs to be *placed*. This corresponds to the placement task. Two placement scenarios are evaluated as shown in Fig 5 (bottom right). Using the measures of three canonical dimensions measured from the object, the first class of opening size allows four out of six approach directions for placement to fit, while the other only allows two approach directions. An error tolerance of $2.00cm$ is considered in the dimension of the opening.

The idea is that more constraints (lesser approach directions) need deliberate planning to choose precise grasp and handoff sequences that allow the placement. The low error tolerance also motivates the use of tracking to adjust within-hand motions of the object in closed-loop execution.

Evaluation metrics: The following metrics are reported for the manipulation trials and used for evaluating the task success rate and efficiency of the manipulation pipelines. *Success (S)* denotes the percentage of trials that resulted in collision-free, successful insertion of objects within the constrained opening, while *Marginal Success (MS)* records trials where the object grazes the boundaries of the constrained space during a successful insertion. The failures include *Placement failures* where the final act of placement fails to insert the object, *Handoff failures* where executing the transfer of object between the arms fails, and *No Solution* cases when planning fails and nothing is executed.

In terms of quality metrics, *Task planning time* records open-loop manipulation planning, *Move time* records the time the robot is in motion, and *Sensing actions* counts the number of times the robot actively reconfigures the object to acquire sensor data from a new viewpoint.

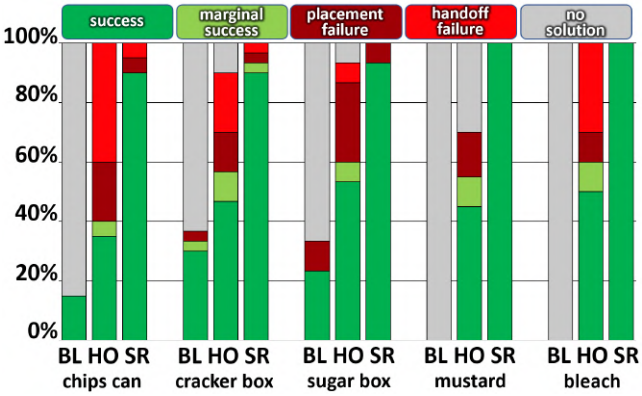

Fig. 7. Figure indicates the split of outcomes of experiments within success and various failure cases for each category.

Comparison: Shape completion, and reconstruction have been mentioned as potential alternative solutions to the problem. The focus of this work is on manipulation of objects with no prior information. Shape completion can only operate under the existence of priors, or with heuristic assumptions like symmetry etc. Our proposed pipeline reflects an integrated perception, planning, and execution paradigm that is an improvement upon both alternatives, and leverages the benefits of our shape representation and tracking. As such, we choose to perform an ablation study that improves upon the more model-agnostic and generalizable of the alternatives - shape reconstruction as the *Baseline* (BL).

240 trials are performed with combinations of *object sets*, *initial configurations* and *placement constraints*. Out of these, 120 experiments use the *Baseline* pipeline shown in Fig. 2 and the rest use the proposed pipeline.

Baseline (BL): The baseline corresponds to the shape reconstruction pipeline but *without the option for handoffs*. Once picked with a task-agnostic grasp, the object is moved in front of the sensor at a predefined pose, and RGB-D

TABLE I

Object	#Experiments	Baseline		(+) Handoff		(+) Shape-representation	
		S (%)	S + MS (%)	S (%)	S + MS (%)	S (%)	S + MS (%)
001_chips_can	20	15.00	15.00	35.00	40.00	90.00	90.00
003_cracker_box	30	30.00	33.33	46.66	56.66	90.00	93.33
004_sugar_box	30	23.33	23.33	53.33	60.00	93.33	96.66
006_mustard_bottle	20	0.00	0.00	45.00	55.0	100.00	100.00
002_bleach_cleanser	20	0.00	0.00	50.00	60.00	100.00	100.00
Overall	120	15.83	16.66	46.66	55.00	94.16	95.82

Evaluating the task success rate of the proposed manipulation pipeline against a baseline. Overall 240 manipulation trials were executed, where the results corresponding to *Baseline* and *Baseline + handoff* are derived from the first set and the *shape representation* results are derived from the second set. *S* indicates successful insertion in the constrained space, and *MS* stands for marginal success, where the object made contact with the boundary of the constrained space but the task still succeeded.

TABLE II

	Baseline + Handoff			Shape-representation				
	sense-place	sense-hoff-place	overall	place	sense-place	hoff-place	sense-hoff-place	overall
#instances	20.0	46.0	66.0	18.0	22.0	51.0	24.0	115.0
tp time (s)	4.29 ± 3.59	5.87 ± 2.88	5.39 ± 3.20	1.10 ± 0.47	6.69 ± 4.15	5.41 ± 3.14	13.50 ± 8.69	6.67 ± 6.22
move time (s)	9.92 ± 1.04	19.91 ± 1.87	16.88 ± 4.88	6.13 ± 2.76	7.24 ± 1.24	18.12 ± 2.02	18.22 ± 1.67	14.18 ± 5.79
sense actions	4.0 ± 0.0	4.0 ± 0.0	4.0 ± 0.0	0.0 ± 0.0	1.36 ± 0.56	0.0 ± 0.0	1.41 ± 0.57	0.59 ± 0.81

Comparing the quality and computation time for the solutions found with the baseline and the proposed approach. The data is presented only for successful executions within each category.

images are captured from 4 different viewpoints by rotating the object along the global *Z-axis* by an angle of $\pi/2$. All 4 views are merged to obtain the object's reconstruction. Manipulation planning is then invoked to find a pick-and-placement (no handoff) solution with the left arm if it exists. The baseline achieves a very low success rate (Table. I) and the most dominant failure mode is *No Solution* (Fig. 7) since the initially chosen grasp might not allow task completion.

(+) Handoff (HO): An improvement over BL, this allows the manipulator an additional option of transferring the object to the fingered gripper which can then be used for reorientation and placement in the constrained space. The overall success rate increases significantly when additional handoff actions are available. Nonetheless, the handoff by itself can be seen as a constrained placement problem, and as this approach commits to a pick for object reconstruction without manipulation planning, it could still lead to *No solution* cases specially for relatively smaller sized objects such as for the *Mustard bottle* (Fig. 7). The grasps with the fingered gripper are computed assuming that the reconstructed geometry is indeed the complete model of the object. However, views across a single rotation are not sufficient to complete the object shape. This causes grasps to collide with the unmodeled parts of the object during execution (*Handoff failures*). Handoff actions often disturb the within-hand object pose and can cause *Placement failures*.

(+) Shape representation (SR): The proposed pipeline reflects Fig 2 (right). The method discovers four different classes of solutions (Fig 6) which compose a sequence of *shape initialization, picks, updates, handoffs and placements*. The key benefit is that our pipeline chooses the mode of operation based on the problem at hand, and tries to *(a)* perform the minimum number of sensing actions *(b)* with a minimum number of manipulation actions *(c)* in a robust fashion that accounts for non-prehensile errors *(d)* while guaranteeing safe execution and successful task completion.

The results reflect that our pipeline achieves all of the above by leveraging the proposed *shape representation*, *integrated perception and planning* in the pipeline, and *closed loop execution* to achieve a success rate of **95.82%**.

SR eliminates the cases of *No Solution* by performing manipulation planning with a *large, diverse, and robust* set of grasps. It ensures successful execution of the task by conservative modeling of the unseen parts of the object to avoid collision and by tracking the shape representation to account for any in-hand motion of the object and adjusting the computed plan. The failure cases for this approach are due to failures in tracking. If the within-hand motion is too drastic motion plans might not be found for local adjustments to the initially computed solution.

As indicated in Fig. 6 (left) and Table. II, the proposed solution can find one of the four solution modes with varying solution quality. The advantage in terms of efficiency comes from the fact that the proposed solution requires additional sensing in only 38% of the runs and the mean number of sensing actions is 1.36 as opposed to the 4 additional sensing actions in every run for the baseline approach. Also the fact that the shape representation allows task planning before picking with multiple grasping options increases the number of single-shot pick-and-place solutions with less motion time in addition to avoiding *No solution* scenarios. The overall execution time reduces significantly due to the combination of these factors. The proposed pipeline shows clear benefits across all the metrics over the extensive real-world trials.

Demonstrations and Publicly-shared Data: On top of the benchmark, additional experiments are performed to demonstrate the proposed system. The first demonstration is performed over mugs, some with and some without handles, with the handles being occluded in the first viewpoint. Such a case imposes ambiguity for shape completion approaches, but is solved with the proposed pipeline as demonstrated in the accompanying video. The second demonstration presents the task of flipping objects and placing them on the table. For objects with no geometric models, tasks specification for object placement can either be relative to constraints in the environment or relative to the initial pose. The following

data items corresponding to all the manipulation runs for the proposed solution are made publicly available at https://cs.rutgers.edu/~cm1074/task-driven.html. 1) Task specification: Initial RGB-D data, object segment, placement region. 2) RGB-D data at 20Hz for the executed trajectory. 3) Robot arm transformations and end-effector grasping status for both grippers. 4) Pose estimates of the initial segment returned by the tracking module for every frame. The data can be used as a manipulation benchmark or to study tracking shapes and poses of objects in-hand during manipulation.

Fig. 8. Demonstrations of the proposed pipeline's operation (left) in the presence of shape ambiguity (right) on the object flipping task.

VII. Limitations and Future Work

The current work paves the way for the paradigm of task-driven perception and manipulation using a possibilistic object representation for solving constrained placement tasks. The results show performance benefits from the design principles adopted in the representation, tracking, and pipeline proposed in the current work.

There are some limitations to the current work that can be addressed in future research. The pick/grasp computation is not the focus here. General grasping strategies on such shape representations can prove useful. The depth sensor used in this study is not suited for reflective and thin objects, and shows significant distortion and smoothing. The bounding box representation for placement is an approximation that is computationally efficient and sufficient for most cases but not ideal. This can be resolved by performing mesh reconstruction and sampling configurations for placement based on those. Segmentation in the presence of clutter is challenging. It is interesting to study the effect of segmentation noise and occlusion due to the clutter on this process. Finally, it is often not possible or safe to insert the object completely in a narrow opening, and in such cases it can be dropped from some height. This process is significantly affected by the object's weight distribution and needs to be modeled or addressed using a controller.

References

[1] F. Wang and K. Hauser, "Robot packing with known items and nondeterministic arrival order," *RSS*, 2019.
[2] R. Shome, W. N. Tang, C. Song, C. Mitash, C. Kourtev, J. Yu, A. Boularias, and K. Bekris, "Towards robust product packing with a minimalistic end-effector," in *ICRA*, 2019.
[3] J. A. Haustein, K. Hang, J. Stork, and D. Kragic, "Object placement planning and optimization for robot manipulators," *arXiv preprint arXiv:1907.02555*, 2019.
[4] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, 2019.
[5] 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 *ICRA*, 2018.
[6] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *ISRR*, 2019.
[7] M. Gualtieri, A. Ten Pas, and R. Platt, "Pick and Place without Geometric Object Models," in *ICRA*, 2018.
[8] M. Alikhani, B. Khalid, R. Shome, C. Mitash, K. Bekris, and M. Stone, "That and there: Judging the intent of pointing actions with robotic arms," in *AAAI*, 2020.
[9] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," in *RSS*, 2018.
[10] C. Mitash, A. Boularias, and K. Bekris, "Robust 6d object pose estimation with stochastic congruent sets," in *BMVC*, 2018.
[11] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE RAM*, vol. 11, 2004.
[12] P. S. Schmitt, W. Neubauer, W. Feiten, K. M. Wurm, G. V. Wichert, and W. Burgard, "Optimal, sampling-based manipulation planning," in *ICRA*, 2017.
[13] 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," in *CVPR*, 2019.
[14] C. Sahin and T.-K. Kim, "Category-level 6d object pose recovery in depth images," in *ECCV*, 2018.
[15] B. Burchfiel and G. Konidaris, "Hybrid bayesian eigenobjects: Combining linear subspace and deep network methods for 3d robot vision," in *IROS*, 2018.
[16] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *ICRA*, 2017.
[17] D. Rodriguez, C. Cogswell, S. Koo, and S. Behnke, "Transferring grasping skills to novel instances by latent space non-rigid registration," in *ICRA*, 2018.
[18] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *RSS*, 2017.
[19] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, "Grasp Pose Detection in Point Clouds," *IJRR*, vol. 36, 2017.
[20] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *ICRA*, 2019.
[21] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *ICRA*, 2015.
[22] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *IJRR*, vol. 34, 2015.
[23] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *IJRR*, vol. 31, 2012.
[24] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in hand model acquisition," in *ICRA*, 2010.
[25] F. Wang and K. Hauser, "In-hand object scanning via rgb-d video segmentation," in *ICRA*, 2019.
[26] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *IROS*, 2017.
[27] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[28] X. Zhang, Z. Zhang, C. Zhang, J. Tenenbaum, B. Freeman, and J. Wu, "Learning to reconstruct shapes from unseen classes," in *NIPS*, 2018.
[29] W. Gao and R. Tedrake, "kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion," *arXiv:1909.06980*, 2019.
[30] A. Price, L. Jin, and D. Berenson, "Inferring occluded geometry improves performance when retrieving an object from dense clutter," *arXiv:1907.08770*, 2019.
[31] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE TRA*, 1996.
[32] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," *IJRR*, vol. 30, 2011.
[33] K. Hauser and V. Ng-Thow-Hing, "Randomized Multi-Modal Motion Planning for a Humanoid Robot Manipulation Task," *IJRR*, vol. 30, 2011.
[34] W. Vega-Brown and N. Roy, "Asymptotically optimal planning under piecewise-analytic constraints," in *WAFR*, 2016.
[35] 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 RAM*, vol. 22, 2015.