Task-driven Perception and Manipulation for Constrained Placement of Unknown Objects

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

***Abstract*—Recent progress in robotic manipulation has dealt with the case of previously unknown objects 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 deals with pick-and-constrained placement of objects without access to geometric models. The objective is to pick an object and place it safely inside a desired goal region without any collisions, while minimizing the time and the sensing operations required to complete the task. An algorithmic framework is proposed for this purpose, which performs manipulation planning simultaneously over a conservative and an optimistic estimate of the object's volume. The conservative estimate ensures that the manipulation is safe while the optimistic estimate guides the sensor-based manipulation process when no solution can be found for the conservative estimate. To maintain these estimates and dynamically update them during manipulation, objects are represented by a simple volumetric representation, which stores sets of occupied and unseen voxels. The effectiveness of the proposed approach is demonstrated by developing a robotic system that picks a previously unseen object from a table-top and places it in a constrained space. The system comprises of a dual-arm manipulator with heterogeneous end-effectors and leverages hand-offs as a re-grasping strategy. Real-world experiments show that straightforward pick-sense-and-place alternatives frequently fail to solve pick-and-constrained placement problems. The proposed pipeline, however, achieves more than 95% success rate and faster execution times as evaluated over multiple physical experiments.**

***Index Terms*—Perception for Grasping and Manipulation; Manipulation Planning; Dual Arm Manipulation**

I. Introduction

OBJECT placement in tight spaces is a challenging problem in robot manipulation. In contrast to a simpler pick-and-drop problem, only specific object poses will allow it to fit in a tight space. Such scenarios occur in logistics applications, such as packing items into boxes, or in service robotics, such as inserting a book into a gap in a bookshelf. Recent work has focused on variants of this problem, such as bin-packing [1], [2] and table-top placement in clutter [3]. Nevertheless, in many cases a geometric and textured 3D model for the manipulated object is assumed to be known. Possessing such high-fidelity models is expensive both in terms of time and effort. In several setups it becomes infeasible to build models due to the wide variety of objects to be manipulated and the resources required for obtaining the models. Some recent robot manipulation systems [4], [5] have shown the capacity of picking novel and previously unseen objects from clutter. These systems, however, typically assume no constraints for the object's placement. Therefore, the object is grasped with any feasible and stable grasp without reasoning about placement. Some alternatives 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 the employed representations can guide manipulation planning solutions, they do not account for safety as they do not consider geometric or physical constraints.

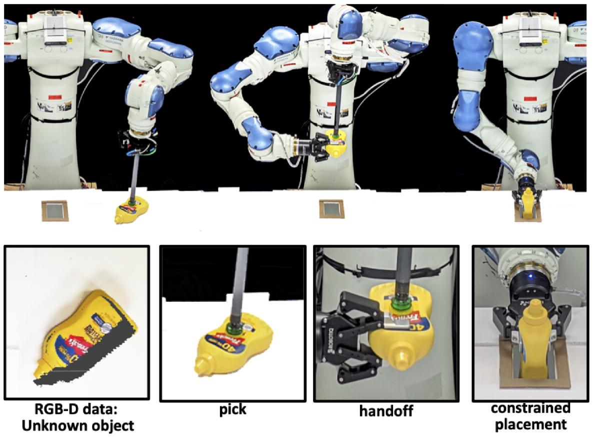

Fig. 1. A demonstration of pick-handoff-place returned by the proposed framework for inserting a previously unknown object in a constrained space. This work does not focus on the last step of precise, closed-loop insertion but how to reason about the object's shape so as to safely bring it to the opening of the placement area. Only a subset of placement poses allow the object to fit into the target area. Experiments (Table.I) consider a larger margin than the demonstration shown here (2 cm instead of 1 cm shown).

This work targets pick-and-place problems where the task imposes constraints on the placement pose. The capabilities of a manipulator impose limitations on what placement poses are reachable depending on the grasp, making certain grasps more desirable than others. This requires careful reasoning to select the pick that will allow the desired placement. This will be referred to as the *pick-and-constrained-placement* problem. In the context of this problem, it is possible that a feasible placement pose is not directly attainable using a pick-and-place operation. Instead, it may require a re-grasping of the object or a hand-off to be executed.

Solutions to such problems typically need object models for

Manuscript received: Feb, 24, 2020; Revised May, 16, 2020; Accepted June, 22, 2020.

This paper was recommended for publication by Editor Dr. Hong Liu upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the NSF, grant numbers IIS-1734492 and IIS-1723869.

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

Digital Object Identifier (DOI): see top of this page.

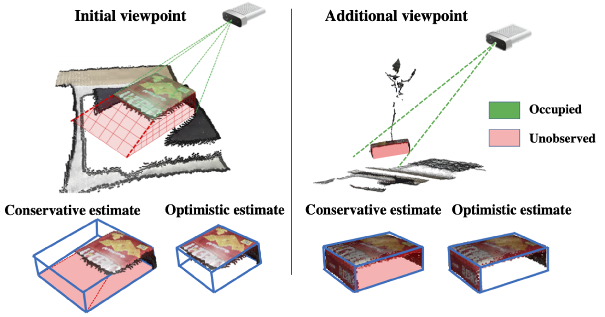

Fig. 2. (Left) Figure shows a partial view of the object from the sensor. A *conservative* estimate of the object considers both the observed and the unobserved parts of the object given knowledge of the support surface. An *optimistic* estimate considers only the observed parts. (Right) The estimates are updated as the object is manipulated to different sensor viewpoints.

collision checking, which this work does not assume. Picking, placement, and re-grasping actions need to be computed given partial viewpoints of the object acquired from the sensor, as in Fig. 2. This work approaches the pick-and-constrained-placement problem without prior object models as *integrated perception and manipulation planning*. The objective is to place the entire object safely inside the desired goal region without any collisions, while minimizing the time and sensing operations required to complete the task.

One option is to pick the object with a task-agnostic grasp that solely interacts with the visible part of the object as in pick-and-drop systems [4], [5]. After picking, the object's shape can be completely reconstructed by manipulating it to different configurations in front of the sensor. Then, geometric planning can be performed based on the reconstructed model. Nevertheless, not only will this option be very time-consuming given the object must be moved to multiple viewpoints, but using a goal-agnostic pick may not allow the constrained placement without multiple regrasps. A key point of this work is that constrained placement task can be successfully completed without a complete object model. This motivates a dynamic estimate of the object's shape to solve the problem, and a planning approach capable of using and updating such a representation on the fly.

The algorithmic solution proposed simultaneously operates over *conservative* and *optimistic* estimates of the object's 3D volume, as in Fig. 2. The *conservative* estimate considers the entire volume attached to the object, which has not been observed by the sensor as part of the object. The *optimistic* estimate considers only the object's observed region to be its complete representation. While the *conservative* estimate ensures that the manipulation is safe, the *optimistic* estimate guides the action selection when no solution can be found for the *conservative* estimate. Both estimates are dynamically updated by incorporating new viewpoints, which are selected such that a safe-to-execute constrained placement solution can be found with minimal sensing.

To efficiently obtain these dynamic estimates, this work proposes to utilize a simple volumetric representation. Similar to occupancy-grids [8] often used in the context of robot navigation to store the occupied and free space, this representation stores whether a voxel in the object's reference frame is occupied, unoccupied or unobserved. Instead of utilizing fixed-size grids or octrees to store the volumetric information, the representation maintains sets of occupied and unobserved voxels. This minimalistic representation provides efficiency at the cost of building exact models but proves to be sufficient to solve the considered problem.

The effectiveness of the proposed approach is demonstrated by developing a robotic system that picks a previously unseen object from a table-top and places it in a constrained space (Fig. 1). The system comprises of a dual-arm manipulator, an RGB-D sensor, a vacuum-based end-effector and an adaptive, finger-based hand. Additionally, the system features *handoffs* to transfer objects between the two arms and a strategy to adjust the computed motion trajectories during real-world execution given sensing updates. Handoff is a re-grasping strategy that allows more flexibility in solving constrained placement problems. Closed-loop execution handles stochastic in-hand motions of objects resulting from unmodeled physical forces like gravity, inertia and grasping contacts.

240 real-world manipulation experiments are performed to compare the proposed solution and the straightforward pick-sense-and-place alternative [1]. The experiments demonstrate that the proposed pipeline is both robust and efficient in handling objects with no prior models within the limitations of the end-effector and the sensor. It achieves a success rate of 95.82%, which is much higher than an alternative that commits to a pick without manipulation planning and performs object reconstruction from heuristic viewpoints without utilizing the conservative volumetric representation. The proposed pipeline results in fewer sensing operations and achieves faster execution times.

II. Related Work

This section discusses existing pick-and-place manipulation pipelines, object representations for these pipelines and assumptions about the object's shape and category.

Manipulation pipelines for pick-and-place: Given access to object models, previous work has addressed problems such as bin-picking [9], tight-packing [1], [2] and placement of grasped objects in clutter [3]. Most manipulation pipelines for novel objects [4], [5] focus on picking the object but do not address the problem of constrained placement. It has been demonstrated that robust grasps can be computed [10] over 3d point cloud representations of novel objects by learning local geometric features. However, constrained placement tasks require simultaneously evaluating placements and grasps over the objects, which is a relatively harder problem than task-agnostic grasping. A recent work, [7] performs pick-and-place of objects without object models, but within a single category, by training an end-to-end deep reinforcement learning framework within the task context. Given that it is hard to interpret the learned policies, it is not clear how the policies learned with rewards coming from a specific task can be generalized to other similar tasks, configurations and objects. Another recent effort [6] proposes using semantic keypoints as category-level object representation in conjunction with shape

[1] Videos and supplementary algorithmic description: https://robotics.cs.rutgers.edu/task-driven-perception/

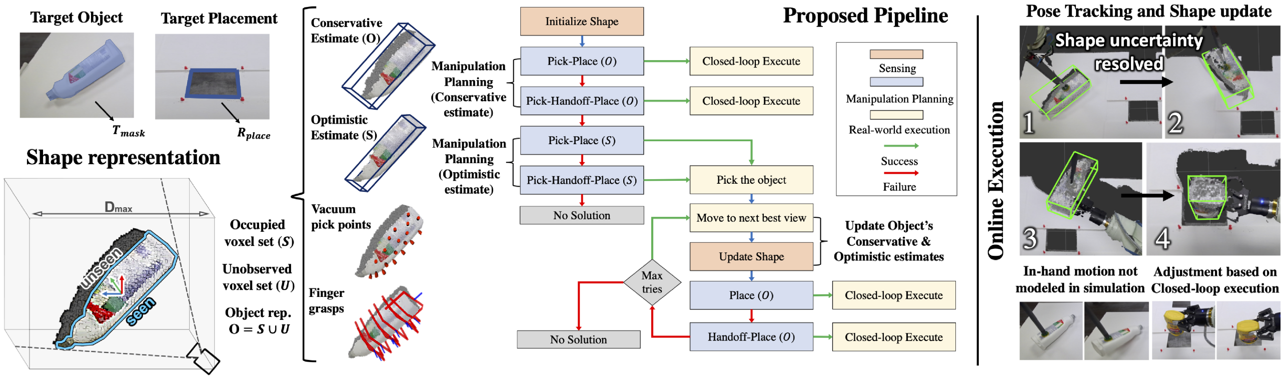

Fig. 3. (left) The proposed framework considers as input RGB-D images and the target object mask and builds a shape representation based on the observed and the occluded part of the object. (center) It simultaneously operates over a conservative and an optimistic estimate of the object's volume to compute a sequence of manipulation and sensing actions for pick-and-constrained-placement. The object is dynamically updated during the manipulation until a safe to execute sequence of action is available. (right) Online adaption is performed, which is informed by pose-tracking to counter the effect of stochastic in-hand motion of the object, which is not modeled during planning.

completion [11] to model collision geometry. Nevertheless, such techniques typically require access to prior knowledge of the object's category to complete its shape and the output is often too noisy for safe manipulation planning in constrained spaces. This work does not assume any knowledge of the object's geometry or category prior while solving pick-and-constrained placement problems.

Object representation for manipulation: Objects are often represented as mesh models that capture the surface of the object. The models are built either using a turntable setup [12], or via in-hand scanning by a human user [13] or a robotic arm [14]. A popular technique for surface reconstruction is Truncated Signed Distance Function (TSDF) [15], [16] which fuses multiple depth observations from a sensor and maintains a signed distance to the closest zero-crossing (representing the surface). Alternatively, the Surfel representation [17] is used to store local surface patches with position and normal information. Nonetheless, the objective is to generate complete meshes and often involves additional setup and post-processing steps. The complete models are then used to perform pose estimation [18], [19], [20], [21] over the online sensor data and transfer the manipulation actions that are defined over the model to the scene. Given the effort in modeling every object instance, some approaches operate at the category-level where objects are represented in a normalized object frame [22] or via a canonical model [23]. 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 instances of the category.

An alternative is volumetric shape completion that has been studied in the context of grasping [24], [25], [26], manipulation [11] and object search [27]. These approaches come up with a most-likely estimate of the object from a partial view based on assumptions, such as symmetry or category-level information. Operating over such estimates can lead to collisions if the estimated volume is smaller than the actual object. Instead, the proposed approach operates only over the sensor data without any assumptions about the object's shape. The object representation in this work is most similar to occupancy grids. Occupancy grids are often used in the context of SLAM or indoor navigation to map boolean or probabilistic occupancy properties either over fixed grid structures [8] or over a more efficient octree representation [28]. Instead of a fixed grid structure the current representation stores the *occupied* and *unobserved* voxels of the object as sets. These sets are updated based on new viewpoints. The minimalistic representation can efficiently maintain a dynamic representation of the conservative and optimistic object volume. Thus, the representation is utilized in the context of the pipeline to perform manipulation planning directly over sensor data, without any assumptions over geometric or category-level priors.

III. Problem Setup and Notation

This section formulates the *integrated perception and manipulation planning problem* for constrained placement.

Object representation: A rigid object can be defined by a region occupied by the object $O^* \subset \mathbb{R}^3$ in its local reference frame that represents its shape. Given a pose $P \in SE(3)$, the region occupied by the object at P is denoted by O^*_P. It should be noted that a geometric model is not available for the object to be manipulated, i.e., O^* is unknown. Thus, O defines an object representation over which manipulation planning can operate. In general, $O \neq O^*$. O is derived from an initial view of the object given point cloud and image segmentation. The resulting object model is typically incomplete, and may not be sufficient to safely place the object in a constrained area.

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 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 Planning: Manipulation planning for constrained placement involves computing a sequence of manipulation actions (picks, placements, re-grasps) that can move the object O^* from P_{init} to P_{target}, which successfully solves a constrained placement task. Such a solution consists of motions of the arms denoted by Π parameterized 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}.

Integrated Perception and Manipulation Planning: Given that the true object geometry O^* is unknown and planning can make use only of the partial object representation O, *perception actions* are also necessary. These actions can update the object representation O by manipulating it to desirable configurations in front of the sensor and obtaining additional sensing information. Thus, the problem involves computing a sequence of perception and manipulation actions, such that: i) the object after executing the sequence of actions ends up inside the defined constraints, i.e., $O^*_{P_{\text{final}}}$ is within R_{place}, where P_{final} is the resultant pose of the object after applying the actions; ii) and the returned sequence of perception and manipulation actions minimizes the task execution time.

IV. Proposed pipeline

This section presents the proposed pipeline as shown in Figure. 3. Given as input RGB-D image of the scene and the target object mask T_{mask}, the object representation O is initialized with it's origin at the centroid of the 3D point cloud segment corresponding to T_{mask} and the reference frame at identity rotation with respect to the camera frame. Within a voxel grid centered at the origin, each voxel is labeled 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 voxels $\{p \in \mathbb{R}^3 \mid p \notin \mathcal{S} \cup \mathcal{U}, \|p - \text{origin}(O)\| < D_{\max}\}$, for a maximum dimension parameter $D_{\max} = 30cm$. The representation is stored as a set O that consists of two mutually exclusive sets of voxels $\mathcal{S}$ and $\mathcal{U}$ in $\mathbb{R}^3$. $\mathcal{S}$ is a set of *occupied* voxels on the surface of the object that are observed by the RGB-D sensor. $\mathcal{U}$ is a set of *unobserved* voxels in space that have not been observed by the sensor given the viewpoints but *have a non-zero probability of belonging to the target object*. Thus $O = \mathcal{S} \cup \mathcal{U}$, where, $\mathcal{S} \cap \mathcal{U} = \phi$.

A set of grasps and placements are computed simultaneously over a *conservative estimate* and an *optimistic estimate* of the object's volume. The *conservative estimate* corresponds to O, while the *optimistic estimate* only considers the observed part of the object, i.e., $\mathcal{S}$. Manipulation planning is performed considering the grasps and placements computed over O. The objective is to compute a sequence of these manipulation actions and corresponding arm motions, which allow to connect a grasp to a placement pose. Any manipulation planning solution computed over O can be directly executed in the real-world as it is necessarily collision-free with respect to the true object shape O^*, given that $O^* \subseteq O$. Often no solution can be found for the task as O may significantly overestimate O^*. In such a scenario, the object is picked and manipulated to acquire new observations, thereby updating O.

The choice of picking point is critical as it might influence the solution once O has been updated. For this reason, manipulation planning is performed over the optimistic estimate of the object's volume. In this case, all actions after picking, such as re-grasps and placements are computed over $\mathcal{S}$. If no placements are achievable given $\mathcal{S}$, the problem is *not solvable*, since $\mathcal{S} \subset O^*$. If a solution is found for $\mathcal{S}$, it informs the selection of the picking point over O.

The next decision is the selection of the next best view. It is selected among a set of pre-defined discrete viewpoints with an objective of exposing the highest number of unobserved voxels in $\mathcal{U}$. This is found by rendering $\mathcal{S}$ at each of the viewpoints and computing the number of voxels in $\mathcal{U}$ that are visible, given the rendered image. The selected 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 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 at time t is transformed to the object's local frame based on the estimated 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, the method iterates over all points in $\mathcal{U}_{P_t}$ to remove points in $\mathcal{U}$, which belong to the empty part of space based on the currently observed depth image. Applying these constraints in the update significantly reduces the drift that occurs in simultaneous updates to the object's pose and shape.

Grasps and placements are re-computed over the updated object estimate and manipulation planning is performed again. This process is repeated until either a solution is found for the constrained placement task or the algorithm runs out of a maximum number of trials. This means that the pipeline does not require the object to be completely reconstructed, but only enough to compute a safe-to-execute solution for the placement task.

V. System Design & Implementation

Fig. 4 shows the hardware setup. It comprises a dual-arm manipulator (Yaskawa Motoman) with two 7-dof arms. The left arm is fitted with a narrow, cylindrical end-effector with a vacuum gripper; and the right arm is fitted with a Robotiq 2-fingered gripper. A single RGB-D sensor (Kinect Azure) is mounted on the robot overlooking both the picking and the placement regions. The sensor is configured in Wide-FOV mode to capture images at 720p resolution with a frequency of up to 20Hz. Below are the implementation details corresponding to different components of the proposed pipeline for this hardware setup.

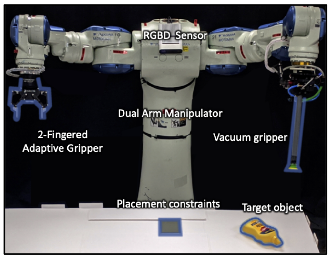

Fig. 4. Hardware Setup.

Grasp computation: Grasp sets $\mathcal{G}_{\text{l}}$ and $\mathcal{G}_{\text{r}}$ are computed over the object 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, the grasps are spread out over O with different approach directions, which provide options to manipulation planning and aid solution discovery.

Vacuum grasps $\mathcal{G}_{\mathrm{l}}$ are computed by uniformly sampling pick points and their surface normals from $\mathcal{S}$, and ranked in quality by their distance from the shape centroid. The grasp set $\mathcal{G}_{\mathrm{r}}$ for the fingered gripper samples a large set of grasps over O according to prior work [10]. Sampled grasps are pushed forward along the grasp approach direction until the fingers collide with points from $\mathcal{S}$ or $\mathcal{U}$, and ranked by the alignment between the finger and contact region on $\mathcal{S}$.

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 correspond to configurations of B_{O} which fit within B_{place}. A discrete set of configurations ($= 24$) for the box is computed by placing B_{O} at the center of B_{place} and validating all axis-aligned rotations. Any pose in the returned set $\mathcal{P}_{\text{place}}$ is a candidate P_{target}.

Manipulation Planning: The input to manipulation planning is the estimated object representation O, the grasp sets for both arms, $\mathcal{G}_{\mathrm{l}}, \mathcal{G}_{\mathrm{r}}$, 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 [29] consisting of 5000 nodes is constructed using the PRM* algorithm [30] 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 [31] which operates over the same roadmap [32] 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 ($P_{\text{target}} \in \mathcal{P}_{\text{place}}$).

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 in the reference frame of the gripper, and b) by eliminating the points corresponding to the gripper's known model. Object pose update $\Delta P^{t-1:t}$ is computed 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 point-set 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.

Reaction to Sensing Updates: Given a manipulation planning solution Π, the objective is to ensure that any errors in execution or non-prehensile grasping interactions are addressed. At any point in time t, $\Pi(t)$ describes how the arms are configured. Assuming prehensile grasps, the expected object pose P^{t*} can be estimated. 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 ΔP errors may require complete re-planning of Π. In this work these adjustments are performed before *handoffs*, and *placements* by locally adapting Π.

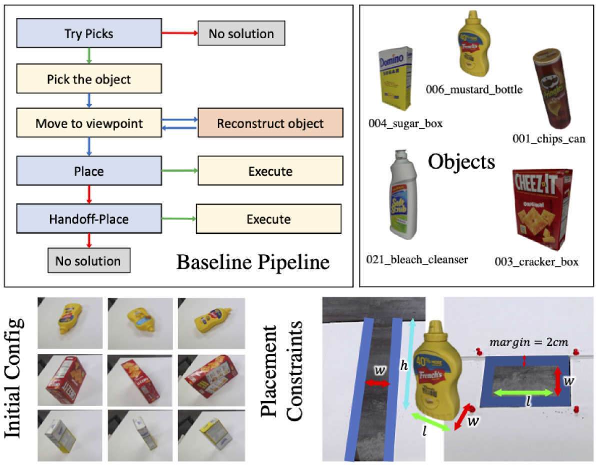

Fig. 5. Objects used in the experiments (top). Examples of initial configurations (left). Examples of placement constraints (right).

VI. Experimental Setup

This section describes the setup for the experiments performed to measure the efficacy of the proposed pipeline and the developed system in solving the *pick-and-constrained-placement* problem. Given the dual-arm manipulator, objects are placed on a table-top 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 can therefore involve a direct placement by the left arm, or a handoff-placement with the right arm. Following describe the different parameters of the setup followed by the evaluation metrics.

Objects: Experiments are performed over 5 YCB [12] objects (Fig. 5) of different shapes and sizes. It should be noted that *no models* are made available to the method.

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. 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. Configurations with limited reachable grasps are ignored. The height of the table is known in advance and it is used to obtain the initial point cloud segment for the object.

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. Evaluating the insertion with a lower margin would need to consider the sensor accuracy, errors in detection of the target placement region and the accuracy of an insertion controller, which are not the focus of this work.

Evaluation metrics: Given the task, the pipeline is responsible to pick the object and re-configure it such that it ends up within the desired placement region on top of the table. A simple control strategy is used to insert the object into the hole and measure the success of the task. It utilizes cartesian control to incrementally lower the object until the joint-limits are reached or a collision is observed. The object is then dropped. *Success (S)* denotes the percentage of trials that result 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. 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 re-configures the object to acquire sensor data from a new viewpoint.

Baseline - Complete Shape Reconstruction: The baseline (shown in Fig. 5) picks the object with a task-agnostic pick (i.e., any pick that works) and reconstructs the entire object by moving to pre-defined viewpoints. 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.

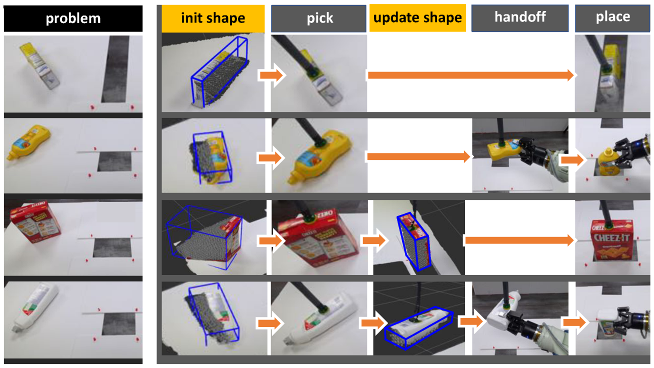

Fig. 6. Qualitative results indicating different solution modes of the proposed pipeline.

VII. Results

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. 5 and the remaining 120 use the proposed pipeline. The results for *Baseline* (BL) and *Baseline + Handoff* (HO) are derived from the same set of physical experiments. Fig. 7 shows the outcome of the experiments. 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.

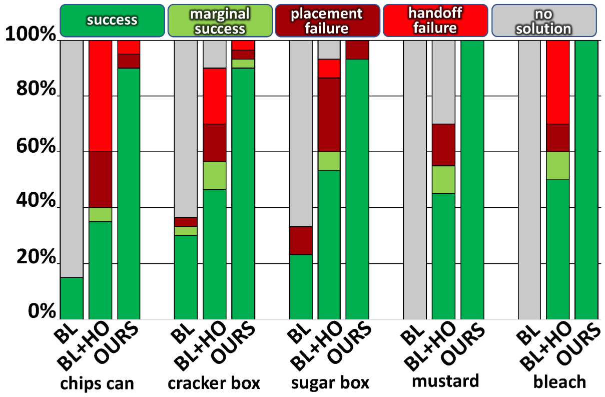

Fig. 7. Split of outcomes of experiments within success and various failure cases for each category.

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 images are captured from 4 different viewpoints by rotating the object along the global *Z-axis* by an angle of $\pi/2$. Views are merged to obtain the object's reconstruction. Manipulation planning

TABLE I

Object	#Experiments	Baseline		(+) Handoff		Proposed pipeline	
		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 results for the proposed pipeline 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			Proposed pipeline				
	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.

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. This implies that given the selected grasp, a reachable, collision-free placement configuration cannot be found for the arm.

Baseline + 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 to reorient and place it 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. Unlike the proposed approach, the baseline does not consider the unseen part of the object as a collision geometry. This causes grasps to collide with the unmodeled parts of the object during execution (*Handoff failures*). The baseline approach performs re-sensing after it picks the object. The re-sensing action prevents any inconsistency due to in-hand motion of the object during the pick. Nevertheless, any in-hand motion that occurs after the reconstruction does not get accounted for and can result in *Placement failures*. Placement can also fail if the reconstructed geometry is an under-approximation of the true object geometry.

Proposed Pipeline: The proposed pipeline discovers four classes of solutions (Fig 6) that compose a sequence of *picks, updates, handoffs and placements*. The key benefit is that it 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 it achieves all of the above by leveraging the *object representation*, *integrated perception and planning* in the pipeline, and *closed loop execution* to achieve a success rate of **95.82%**.

The proposed pipeline 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 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. Additionally, the object representation allows task planning with multiple grasping options even before picking thereby increasing the number of single-shot pick-and-place solutions with less motion time. The overall execution time reduces significantly due to the combination of these factors.

Demonstrations and Publicly-shared Data: On top of the benchmark, additional demonstrations show the capability of 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. Without models, object placement tasks can either be specified relative to constraints in the environment or relative to the initial pose. Following data items corresponding to all the manipulation runs for the proposed solution are made publicly available at https://robotics.cs.rutgers.edu/task-driven-perception/. 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 grasping status for both grippers. 4) Relative pose estimates 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.

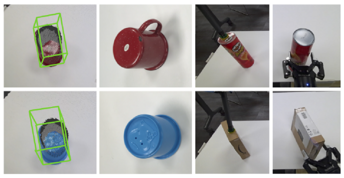

Fig. 8. Demonstrations of the proposed pipeline's operation (left) in the presence of shape ambiguity (right) on the object flipping task.

VIII. Limitations and Future Work

The current work paves the way for the paradigm of task-driven perception and manipulation for solving pick-and-constrained-placement tasks. Not assuming a category-level shape prior or known geometric models and operating directly over the sensor data makes this manipulation pipeline safe to execute and scalable. The results show performance benefits from the design principles adopted in the pipeline and the representation 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. Additionally, the end-effectors utilized in the system restrict the choice of objects that can be evaluated due to limitations based on object's weight, size or material properties. Similar restrictions are due to the depth sensor used in this study as it is not suited for reflective and transparent objects. Segmentation in the presence of clutter is challenging despite the recent progress in depth and color based segmentation [33], [34] of unknown objects. Future work could focus on dealing with segmentation noise and occlusions due to clutter. 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 mass distribution, which also needs to be modeled.

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," *IROS*, 2019.

[4] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, 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] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *ICRA*, 1985.

[9] A. Zeng, K. Yu, S. Song, D. Suo, E. Walker Jr, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *ICRA*, 2017.

[10] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, "Grasp Pose Detection in Point Clouds," *IJRR*, 2017.

[11] W. Gao and R. Tedrake, "kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion," *arXiv:1909.06980*, 2019.

[12] 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*, 2015.

[13] F. Wang and K. Hauser, "In-hand object scanning via rgb-d video segmentation," in *ICRA*, 2019.

[14] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in hand model acquisition," in *ICRA*, 2010.

[15] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH*, 1996.

[16] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR*, 2011.

[17] T. Weise, T. Wismer, B. Leibe, and L. Van Gool, "In-hand scanning with online loop closure," in *ICCV Workshops*, 2009.

[18] 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.

[19] C. Mitash, A. Boularias, and K. Bekris, "Robust 6d object pose estimation with stochastic congruent sets," in *BMVC*, 2018.

[20] B. Wen, C. Mitash, S. Soorian, A. Kimmel, A. Sintov, and K. E. Bekris, "Robust, occlusion-aware pose estimation for objects grasped by adaptive hands," *arXiv preprint arXiv:2003.03518*, 2020.

[21] C. Mitash, B. Wen, K. Bekris, and A. Boularias, "Scene-level pose estimation for multiple instances of densely packed objects," in *Conference on Robot Learning*, 2020.

[22] 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.

[23] D. Rodriguez, C. Cogswell, S. Koo, and S. Behnke, "Transferring grasping skills to novel instances by latent space non-rigid registration," in *ICRA*, 2018.

[24] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *IROS*, 2017.

[25] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap-robotic grasping under incomplete observation," in *ICRA*, 2011.

[26] A. H. Quispe, B. Milville, M. A. Gutiérrez, C. Erdogan, M. Stilman, H. Christensen, and H. B. Amor, "Exploiting symmetries and extrusions for grasping household objects," in *ICRA*, 2015.

[27] A. Price, L. Jin, and D. Berenson, "Inferring occluded geometry improves performance when retrieving an object from dense clutter," *ISRR*, 2019.

[28] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, 2013.

[29] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE TRA*, 1996.

[30] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," *IJRR*, 2011.

[31] K. Hauser and V. Ng-Thow-Hing, "Randomized Multi-Modal Motion Planning for a Humanoid Robot Manipulation Task," *IJRR*, 2011.

[32] W. Vega-Brown and N. Roy, "Asymptotically optimal planning under piecewise-analytic constraints," in *WAFR*, 2016.

[33] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation," *CoRL*, 2019.

[34] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in *ICRA*, 2019.