# Online Object Model Reconstruction and Reuse for Lifelong Improvement of Robot Manipulation

Shiyang Lu[1], Rui Wang[1], Yinglong Miao[1], Chaitanya Mitash[1], Kostas Bekris[1]

*Abstract*— **This work proposes a robotic pipeline for picking and constrained placement of objects without geometric shape priors. Compared to recent efforts developed for similar tasks, where every object was assumed to be novel, the proposed system recognizes previously manipulated objects and performs online model reconstruction and reuse. Over a lifelong manipulation process, the system keeps learning features of objects it has interacted with and updates their reconstructed models. Whenever an instance of a previously manipulated object reappears, the system aims to first recognize it and then register its previously reconstructed model given the current observation. This step greatly reduces object shape uncertainty allowing the system to even reason for parts of objects, which are currently not observable. This also results in better manipulation efficiency as it reduces the need for active perception of the target object during manipulation. To get a reusable reconstructed model, the proposed pipeline adopts: i) TSDF for object representation, and ii) a variant of the standard particle filter algorithm for pose estimation and tracking of the partial object model. Furthermore, an effective way to construct and maintain a dataset of manipulated objects is presented. A sequence of real-world manipulation experiments is performed. They show how future manipulation tasks become more effective and efficient by reusing reconstructed models of previously manipulated objects, which were generated during their prior manipulation, instead of treating objects as novel every time.**

## I. Introduction

General purpose and flexible robot manipulators should be able to manipulate object instances they have never seen before. Once an object has been manipulated, however, the robot should be able to leverage its prior experience for future encounters of a similr object. Such abilities allow the deployment of robots that self-learn to precisely manipulate objects and improve their performance over time.

Recent work on manipulating novel objects either completes their shape based on category-level reasoning [1], [2] or utilizes physical constraints [3]. Single-view 3D shape completion, however, is often not precise and safe enough for many manipulation tasks, such as pick and place in constrained spaces. To ensure safety during manipulation of novel objects, recent prior work [4] considers a conservative estimate of an object's volume. The conservative estimate includes the observed surface of that object together with the volume attached to it, which has not yet been observed bounded by physical constraints, such as the presence of a support surface. To achieve this, the prior work uses a simple volumetric object representation similar to occupancy-grids [5], and proposes action primitives, such as *handoffs* and *active perception*, to reduce shape uncertainty of novel objects during manipulation. While this prior system can deal with constrained placement tasks for novel objects, its success rate and efficiency sometimes suffers. This is due to the fact that every object is treated as novel, even if instances of the same object have been seen before. The current work aims to improve the efficiency of such manipulation pipelines by recognizing previously manipulated objects and reusing the models that it has constructed online given prior manipulation operations. During the development of the proposed work, it was identified that occupancy-grid representations are not precise enough for reusing the reconstructed object model in future tasks.

To address these issues, this paper proposes a manipulation pipeline that performs object picking and constrained placement via life-long object model reconstruction and reuse. It is based on the hypothesis that some objects will reappear over multiple manipulation tasks. For instance, this can occur in logistics setups where singulated objects are dropped from a conveyor belt and then need to be picked and placed in a container to be shipped. To achieve more accurate reconstruction and model reuse, the Truncated Signed Distance Function (TSDF) is adopted to represent partial models. Similarly, a variant of a standard particle filter [6], [7], [8] is used for performing pose estimation and tracking of the partial TSDF models. This variant prunes pose hypotheses that violate viewpoint or physical constraints, and rejects models of falsely recognized objects from being reused. It achieves speed advantages by rendering objects in a region of interest instead of a full image. Furthermore, this work presents an effective way to construct a dataset on the fly that stores partially reconstructed object models for future tasks. This dataset also stores a set of color features for each object that are the output of a clustering algorithm given previous object viewpoints. The clustering is experimentally shown to result in efficient and accurate object recognition.

A sequence of real-world experiments with a manipulator and a set of objects is performed to show that more successful and more efficient robot manipulation can be achieved over time by proper reconstruction and reuse of object models. Compared to the baseline [4], the proposed robotic system not only achieves higher success rate (by a 13% margin), but also significantly improves efficiency. In particular, it reduces *handoff* actions by 31%, and reduces *active perception* actions by 49% over the same sequence of manipulation tasks against the baseline.

[1] The authors are affiliated with the Department of Computer Science at Rutgers, the State University of New Jersey, New Brunswick, NJ, 08901, USA. Email: shiyang.lu@rutgers.edu; kostas.bekris@cs.rutgers.edu

This work is partially supported by NSF NRI award IIS 1734492 and HDR TRIPODS award 1934924.

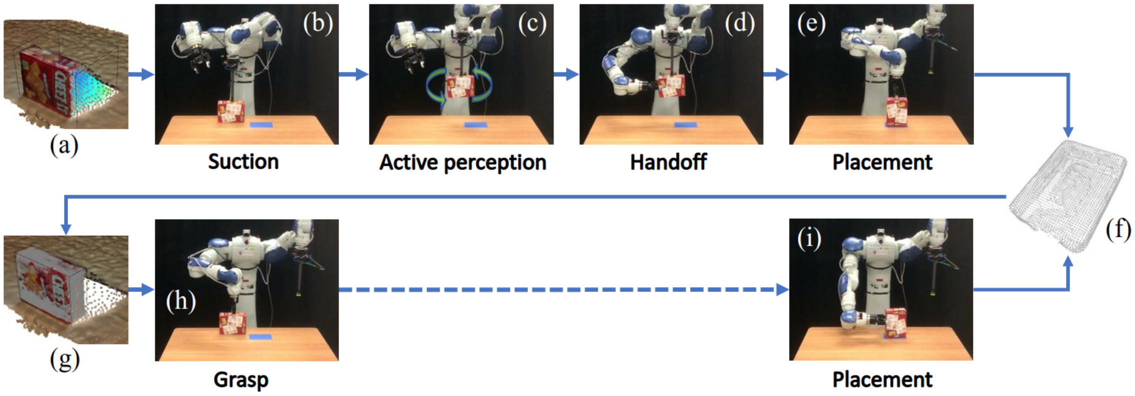

Fig. 1. (top row) A manipulation sequence for an unknown object observed for the first time. Model reconstruction is executed on the fly to get an object model. (bottom row) The same object is recognized and its reconstructed model registered against the observation resulting in a more efficient manipulation sequence. The task involves picking (images b and h), and placing the object (e and i). Without a known geometric model, this task may not be directly solvable and may require multiple intermediate actions, such as *active perception* (c), or reorientation through actions, such as *handoffs* (d). Reconstructing and reusing object models during manipulation allows to avoid these actions in future episodes due to reduced uncertainty. In this lifelong process, the object models (f), are continuously updated over time given new viewpoints.

## II. Related Work

### A. Pick-and-place Manipulation for Novel Objects

Robot manipulation systems for tight packing [9] or placement in constrained spaces [10] often assume the availability of complete 3d object models. For novel object instances, a majority of recent work focuses on task-agnostic picking [11], [12] while others resort to shape completion, performed either via category-level reasoning [1], [2] or given physical consistency constraints [3]. Nevertheless, the output of shape completion may not be precise enough and lead to collisions when the object is placed in a constrained space. Recent work [4] proposes to use a conservative shape representation for pre-pick planning to ensure safe manipulation and reconstruct the shape of the object in-hand, if the task requires it. In certain scenarios, the conservative estimate might be too restrictive for the constrained placement task. To address this, the current work proposes to recognize previously seen objects and perform life-long model reconstruction over many manipulation runs.

### B. Simultaneous Tracking and Object Reconstruction

Object models are often generated by using a turntable [13], or via manual scanning [14] or a robotic arm [15] and post-processed. These models are then used for single-shot pose estimation [16], [17] or model-based object tracking [18]. Model-free manipulation research has used local scan matching [19] with an occupancy grid structure [4] to simultaneously track and reconstruct a conservative object volume. Another popular surface reconstruction technique often used in SLAM is Truncated Signed Distance Function (TSDF) [20], [21]. It fuses multiple depth observations from a moving sensor and maintains a signed distance to the closest zero-crossing (representing the surface). The current work leverages TSDF in a particle filter for simultaneous tracking and reconstructing a manipulated object.

### C. Object Identification and Pose Estimation

Previous work [22], [23] has shown that features trained on large-scale classification datasets allows for image matching. The current work leverages such pre-trained features to store object viewpoints and re-identify object instances. Pose estimation based on particle filters [17] has been used before for matching complete object models with object segments in the scene. The current work utilizes a similar technique but with partial object models that were constructed from past manipulations.

## III. Problem Setup and Notation

**Object Representation** The $i^{th}$ rigid object to be manipulated is defined by the volume it occupies $O^i \in \mathbb{R}^3$ in a local reference frame. Given a pose $P^i \in SE(3)$, the 3D region occupied by $O^i$ in the global frame is denoted as $O^i_{P^i}$. Note that the ground truth geometric model of this object is not available, i.e., $O^i$ is unknown. The estimated object representation $\hat{O}^i$ is composed of the object's surface $S^i$ and a conservative volume $U^i$, which has not yet been viewed but may contain part of the target object, i.e. $\hat{O}^i = S^i \cup U^i$.

**Object Recognition and Pose Estimation:** One singulated object $O^i$ appears for picking for each task $i$. The robot first determines whether $O^i$ has been manipulated before given the initial RGB-D observation $I^i_{init}$. If $O^i$ is recognized as in the same category as $O^j$, where $j < i$, then the reconstructed model $\hat{O}^j$ is reused to initialize $\hat{O}^i$, which is the registered output of the current object point cloud in $I^i_{init}$ and $\hat{O}^j$. The initial pose $P^i_{init}$ is set to be $\hat{P}^i$, which is the estimated pose of the object model $\hat{O}^i$ given $I^i_{init}$. If $O^i$ is recognized as a novel object, then $\hat{O}^i$ is directly initialized from the object's point cloud in $I^i_{init}$.

**Constrained placement** Given an object at a pose $P^i_{init}$, the goal of the constrained placement is to move $O^i$ to a pose $P^i_{target}$, such that $O^i_{P_{target}} \subset R^i_{place}$ where $R^i_{place} \in \mathbb{R}^3$ is a target placement region. To accomplish this, a sequence of manipulation actions, i.e. *pick*, *handoff*, *place* are performed. Since the ground truth $O^i$ is unknown, perceptive actions that sense the object may also be needed.

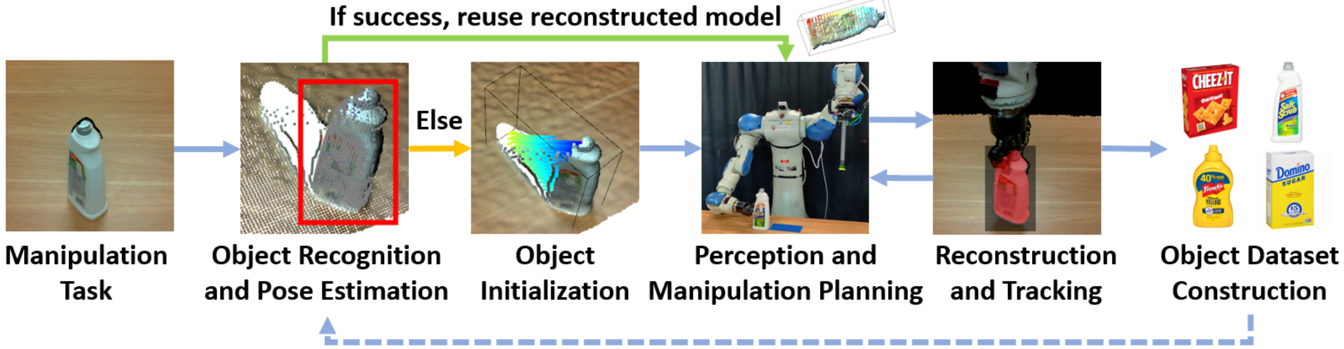
If success, reuse reconstructed model
Else
Manipulation Task
Object Recognition and Pose Estimation
Object Initialization
Perception and Manipulation Planning
Reconstruction and Tracking
Object Dataset Construction

Fig. 2. Proposed pipeline: For each object, recognition is first executed given an object dataset. If the object is recognized, pose estimation given its existing (perhaps partial) model is performed and the model is reused. If not, a new, partial model is initialized from current data. Either way, a perception and manipulation process is executed. If the model is not completed enough, additional sensing and reorientation action may be needed to place the object in a constrained area. Visual tracking is executed to dynamically update the object model in parallel. The latest model is stored in the dataset.

## IV. System Design and Implementation

The proposed pipeline is shown in Fig. 2. At the beginning of each task, the robot first determines whether the target object has been seen before. If an object is considered novel, then its model will be initialized based on the current RGB-D observation. Otherwise, a previously reconstructed model, which may be partially complete, is registered to the current observation and reused in the current task. An integrated perception and manipulation planning process is performed thereafter to accomplish a constrained placement task. During manipulation, the object is tracked and its model is dynamically updated. The reconstructed model is also used by the manipulation planning process, which forms a feedback loop. A dataset, which stores object information, is updated after each manipulation task to benefit future tasks.

### A. Object Initialization

An object model is initialized when the target object is considered unknown. Then, a truncated signed distance function (TSDF) [24] representation is used as the object model. TSDF has been widely used for high-quality scene reconstruction [25], [26]. Each voxel in a TSDF volume stores the signed distance $d$ to its closest surface, where the sign of $d$ indicates whether the voxel is in free space $(d > 0)$ or in a conservative estimate of the object's volume $(d < 0)$. The surface point cloud $S^i$ of the object can be extracted at the zero crossings of the function either by ray-casting or a marching cubes algorithm [27]. An object's TSDF volume is initialized given a minimum oriented 3D bounding box that encloses the observed point cloud of an object and its occluded region. Standard methods are used to approximately compute this 3D bounding box [28], [29]. The point cloud segment of the object can be easily obtained since each task only contains one object. The voxel size of a TSDF volume is set to be 1mm in the accompanying implementation, which is small enough to capture object details for both 3D registration and grasp pose detection.

### B. Dataset Construction

A dataset is constructed from scratch to store information of manipulated objects. For each object instance, a reconstructed TSDF model and a set of RGB features are stored. Cropped RGB images captured during manipulation are fed to a neural network for extracting features representing the object. The implementation uses ResNet50 [30] pretrained on ImageNet [31] as the feature extractor similar to related work [32]. The choice of the specific feature extractor is not the main focus of this work and can be replaced with alternatives. Since an object may look very different from different viewpoints, a set of features vectors are computed via clustering to represent an object instead of a single feature vector. In particular, the mini-batch KMeans algorithm that has been designed for efficient incremental clustering of new data [33], [34], is adopted to cluster features from similar viewpoints. It is called after each manipulation task. Given an ablation study (Sec. V-B.4), a value of $K = 64$ for the number of clusters used for an object's features provides good viewpoint diversity and near instant fitting speed.

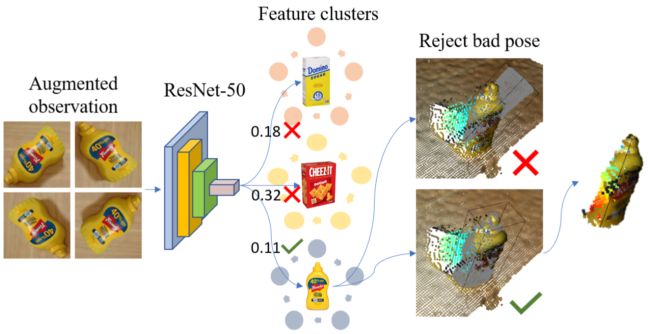

Fig. 3. Object Recognition and Pose Estimation. A segmented observation is first augmented by rotation and then fed to a neural network for feature extraction. The features are compared with centers of feature clusters of objects in the dataset. The object with the closest cosine distance that is less than a threshold ($\delta = 0.15$) is considered as a matching candidate. Pose estimation with viewpoint constraints is performed to reject false positives.

### C. Object Recognition and Pose Estimation

Given an initial observation $I^i_{init}$ for task $i$, the robot first attempts to recognize $O^i$ and estimate its pose $P^i_{init}$. Despite notable progress in object recognition and 6D pose estimation, these problems remain challenging in the considered setup, since: 1) the dataset is constructed from scratch and data collected from one task can be insufficient to train a deep model; 2) retraining a deep model after each task is time consuming and violates the objective of performing efficient manipulation. This work proposes a two-stage method that performs object recognition and pose estimation without retraining a feature extractor multiple time as in Fig. 3.

First, a cropped observation is augmented rotation-wise 8 times and fed to a feature extractor. The features for each rotated image are then compared using cosine distance against the K cluster centers of the feature sets selected to represent each object. Among the nearest neighbors of all rotated images within a cosine distance $d < \delta = 0.15$, the most similar nearest neighbor is selected as a matching candidate $\bar{O}^i = \hat{O}^j$. If no nearest neighbor has a distance $d < \delta$, then this object is considered to be novel. An ablation study of the threshold $\delta$ is performed in section V-B.4.

**Algorithm 1** 6DoF Pose Estimation

```
Require: Observed depth image $I$, object TSDF volume $V_{obj}$, extrinsic matrix $E$, intrinsic matrix $C$, number of particles $M$, number of iterations $K$, pixel depth inlier threshold $d_{thres}$, rejection ratios $\beta_1, \beta_2$.
 1: Generate scene TSDF volume $V_{scene}$ from $I$, $E$, and $C$.
 2: Filter table and get object region of interest for rendering.
 3: Compute 3D centroid $c$ of $I_{roi}$ projected in 3D space.
 4: Initialize a set of $M$ particles at $t = 0$, $\mathcal{X} = \{x_t^1, ..., x_t^M\}$ located at position $c$ with random orientation in $SO(3)$.
 5: for $t = 1$ to $K$ do
 6:     $\hat{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
 7:     for $m = 1$ to $M$ do
 8:         Diffuse $x_t^m \sim p(x_t | u_t, x_{t-1}^m)$          ▷ $u_t$ is zero.
 9:         Render object depth $I_r$ in RoI given $V_{obj}$ and $x_t^m$
10:         $w_t^m$ = count_pix_inliers($I_{roi}$, $I_r$, $d_{thres} = 1cm$)
11:         $\hat{\mathcal{X}}_t = \hat{\mathcal{X}}_t + \langle x_t^m, w_w^m \rangle$
12:     for $m = 1$ to $M$ do
13:         Draw $x_t^i$ with probability $\propto w_t^i$
14:         $\mathcal{X}_t = \mathcal{X}_t + x_t^i$
15: Sort $\mathcal{X}_K$ based on particle weights (descending order)
16: $x_{best} \leftarrow$ None
17: for $m = 1$ to $M$ do
18:     Render object $I_r$ at $x_K^m$ and project to $V_{scene}$
19:     $\beta_{free} \leftarrow$ pts_ratio_in_freespace($I_r, V_{scene}, C, E$)
20:     $\beta_{collide} \leftarrow$ pts_ratio_below_table($I_r, V_{scene}, C, E$)
21:     if $\beta_{free} < \beta_1$ and $\beta_{collide} < \beta_2$ then
22:         $x_{best} \leftarrow x_K^m$
23:         break
24: Return $x_{best}$
```

Then, a particle filter variant is used to estimate the pose of the object candidate $\bar{O}^i$, which can help reject potential false positives in recognition. The variant is detailed in Alg. 1 and is adapted from existing Monte Carlo localization methods [6], [7], [8], with the following differences: 1) It can work on partially reconstructed TSDF volumes other than complete mesh models; 2) Rendering is only performed in a Region of Interest (RoI - referred to $I_{roi}$ in the algorithmic), which is the smallest 2D bounding box of the conservative volume estimate augmented by a 30% margin. This makes the algorithm more efficient ($\sim 30ms/iter$ vs $\sim 60ms/iter$ [8]) with the same number of particles ($N = 625$); 3) Two rejection criteria are introduced to prune bad pose hypotheses that either violate viewpoint constraints or physical constraints, i.e. a registered model should not lie in the free space or collide with the supporting plane. If the number of pixel violating these criteria over the total number of pixels in the ROI is less than $\beta_1$ and $\beta_2$ respectively, where $\beta_1$ and $\beta_2$ are predefined thresholds ($\beta_1 = \beta_2 = 0.95$ in the accompanying implementation), then it is considered a good registration. Otherwise, $O^i$ is considered novel and will be initialized from scratch. These additional pose rejection criteria minimize the chance of a falsely recognized object to be registered and fail a manipulation task. The algorithm is implemented using PyCUDA [35], and the authors have open sourced on Github as a 6DoF pose annotation tool.

### D. Simultaneous Reconstruction and Tracking

The object model $\hat{O}^i$ is tracked and reconstructed over time. The same particle filter variant as in Alg. 1 is reused with the following changes: 1) The object transition model (in line 8) between two time steps is set to be: $u_t = \Delta E^i_{t-1:t}$, instead of 0, where $E^i_t$ is the end-effector's pose computed via forward kinematics; 2) One iteration is performed for each new observation; 3) The RoI is computed by first rendering the object at $\hat{x} = u_t \cdot \hat{x}_{t-1}$, where $\hat{x}_{t-1}$ is the most likely estimate, and then finding its minimum 2D bounding box augmented by a 30% size increase; 4) The rejection criteria are not used for tracking. New observations are then integrated to the object's TSDF volume after the arm and the end-effector are filtered. Since the ground-truth, frame-to-frame tracking pose of a model under reconstruction is not available, tracking quality is implicitly evaluated by the final object reconstruction (Sec. V-B.5).

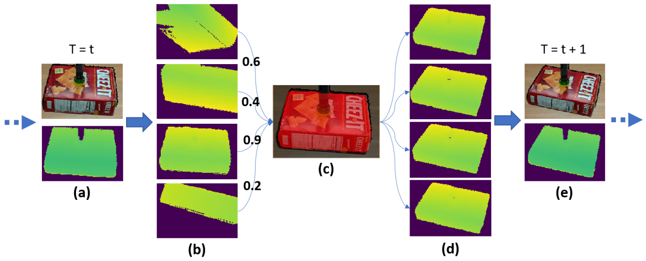

Fig. 4. Illustration of Simultaneous reconstruction and tracking. Given a new observation at $T = t$ (a), a transition model given a diffusion process is applied on particle estimates, which are sampled given the previous time step, and rendered at the region of interest (b). The particle with the largest weight is selected as the pose prediction at $T = t$ (c). The current observation is integrated to the object representation and particles are then resampled (d) to become available for the next time step (e).

### E. Perception and Manipulation Planning

Given a specified constrained area $R^i_{place}$, the robot will first check if the target object can be directly grasped by its end-effectors given the object's conservative volume. The dual-arm robot prioritizes the two-finger gripper as the orientation of the suction gripper is limited. Grasp poses are computed over the surface representation $\hat{S}^i$ to ensure stable geometric interaction [36]. For the suction gripper, suction points are sampled on the object surface $\hat{S}^i$, where the normalized normal $N$ is close to the global $z$ axis i.e.,

$|N_z| > 0.8$. Suction points are further ranked in quality according to their distances from the center of the surface.

Given the constrained area $R^i_{place}$ and the object representation $\hat{O}^i$, two bounding boxes are computed: 1) the maximum bounding box inside the constrained area, i.e., $B^i_{place} \subset R^i_{place}$, 2) a minimal bounding box $B_{O^i}$ that encloses $\hat{O}^i$. A discrete set of configurations (= 24) for the object's bounding box are computed by placing $B_{O^i}$ at the center of $B^i_{place}$ and are validated by all axis-aligned rotations. If no placement can be found, an active perception action will be taken to move the object to the front of the camera and rotate along the z-axis by 180 deg. to reduce model uncertainty, and recompute the object bounding box and placement. If there exists a placement but the target pose is beyond the reachability limits of the suction gripper, a *handoff* action will transfer the object from the suction to the parallel gripper.

As part of the planning framework, a probabilistic roadmap (PRM$^*$) [37], [38] is pre-computed for each of the arms, which takes into account collisions with static obstacles, such as the table. To generate informative paths for the considered setup, the configurations along the roadmap are sampled so that the end-effector is within the camera's view. This allows the object to be tracked during arm movement. Based on the precomputed PRM$^*$ roadmap, a lazy version of the A$^*$ algorithm is used online for computing a shortest path on the roadmap, where lazy collision checks with the object are performed after an initial solution path is found. Once a collision has been detected, the roadmap is modified and a new A$^*$ query is triggered. The loop continues until a valid path is confirmed.

## V. Experiments

Experiments are designed to showcase the effectiveness and efficiency of the proposed robotic manipulation system. It is compared with a baseline system [4], which was designed to perform similar manipulation on unknown objects but considers all objects as novel without learning object information from tasks or reuse reconstructed object models in future tasks. In addiition, evaluation and ablation studies are performed to show the efficacy of the system's submodules. In particular, the proposed system is evaluated from the following perspectives: 1) Success rate, 2) system efficiency, 3) reduced shape uncertainty after registration, 4) object recognition, and 5) object reconstruction.

### A. Hardware and Experimental Setup

The hardware setup is shown in Fig. 5. It comprises of a dual-arm manipulator (Yaskawa Motoman) with two 7-dof arms. The left arm is fitted with a suction gripper, while the right arm is fitted with a Robotiq 2-fingered gripper. A single RGB-D sensor (RealSense L515) is mounted on the robot torso overlooking the workspace. The camera is configured to capture 480p RGB-D images at a frequency of 30 Hz.

Randomly ordered constrained placement tasks are performed given 4 objects (shown in Fig. 7). Each task requires the robot to pick up an object from the table and place it in a constrained area. The target object is randomly positioned on the 2D plane within the reach of both grippers. The objects are placed on different sides and rotated along the z-axis by a predefined angle for each experiment. Since the *bleach* object can't be placed stably on its side but either stand or lie flat on the table, only 10 experiments were performed for it. For *cheezit*, *sugar*, and *mustard*, 15 experiments are performed for each. In total, 110 real world pick and place experiments were executed for both systems for comparison.

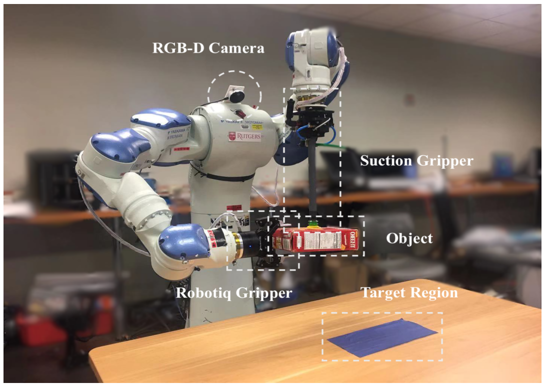

Fig. 5. The hardware setup involving a dual-arm manipulator with a suction and a parallel gripper manipulating objects in the presence of a static RDG-D camera so as to achieve placement in a target region.

### B. Results

*1) Success Rate:* There are certain object configurations that correspond to a large initial conservative estimate of an object, and thus no safe grasps or top-down suction points can be found, e.g., a standing bleach cleanser. In these cases, the task will fail if uncertainty is not reduced. In a sequence of 55 constrained placement experiments, eight tasks failed for the baseline experiment, while only one task failed for the proposed experiment, which is due to a standing *mustard* not being recognized. Table I provides the statistics.

| | **Success Rate** | # Handoffs | # Active Perception |
|---|---|---|---|
| Baseline [4] | 47/55(85%) | 37/55(67%) | 35/55(63%) |
| Proposed | 54/55(98%) | 20/55(36%) | 8/55(14%) |

TABLE I
Task success rate (higher is better)
and ratio of primitive actions used (lower is better)

*2) System Efficiency:* This is evaluated by counting the number of times two action primitives are used, i.e. *active perception* and *handoff*. *Active perception* means that the robot moves the object in front of the camera and rotates it along $Z$ to reduce shape uncertainty. *Handoff* means the robot transfers the object between grippers to achieve a grasp that allows placement. Both actions can be potentially avoided given a better model. The fewer times these actions are taken, the more efficient the manipulation process is.

*3) Reduced Shape Uncertainty:* By registering a previously reconstructed model, the initial conservative estimate of an object is greatly reduced. This allows a robot to detect more potential grasps and increases the collision-free space for motion planning. As a reminder, the conservative estimate of an object's volume is defined as the ground truth volume of that object together with the volume attached to it, which has not yet been observed. Then, the shape uncertainty is defined to be the ratio of the conservative estimate of an object's volume after a model has been registered over the conservative estimate of that object given only the current observation. The result for each experiment is shown in Fig. 6. This smaller this ratio is, the more uncertainty is reduced by registering against a previously constructed model. This ratio is reduced by $32\%$ on average, and up to $75\%$ in some cases for all the experiments by reusing a partially reconstructed model.

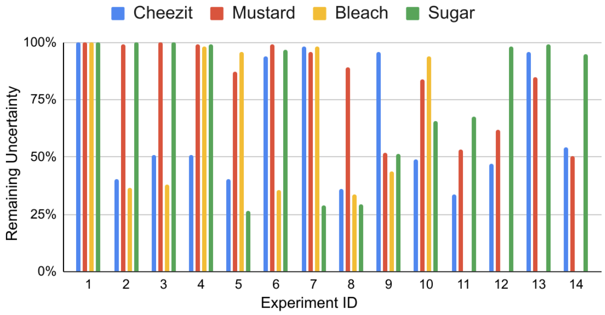

Fig. 6. Remaining uncertainty of each experiment.

*4) Object Recognition:* The target object is correctly recognized in $49/55$ of the real experiments. In $5/55$ of the experiments the object is erroneously not recognized as a previously manipulated object (false negative). Only one object was initially falsely recognized as a previously manipulated object (false positive), but was then rejected in the pose estimation stage. A small cosine distance threshold $\delta = 0.15$ is used for feature matching to minimize false positives (as in Sec. IV-C). And a value $K = 64$ is used for the mini-batch K-Means clustering approach (as in Sec. IV-B), which represents the features of each manipulated object from different viewpoints. False negatives typically occur when the currently observed object part has not be seen in previous experiments. While false positives may cause task failures, false negatives only decrease efficiency as the object will be considered novel. To make the results more statistically meaningful, the data collected from this sequence of experiments were shuffled and an ablation study was performed for the values of the $\delta$ and $K$ parameters. Table II shows how these two values affect precision and recall. The results are computed from an average of 30 randomly shuffled sequences.

| | K = 1 | K = 16 | K = 64 | K = 128 |
|---|---|---|---|---|
| $\delta$=0.14 | 0.995/0.131 | 0.975/0.755 | 0.981/0.831 | 0.988/0.844 |
| $\delta$=0.15 | 0.996/0.211 | 0.955/0.842 | 0.958/0.886 | 0.964/0.891 |
| $\delta$=0.16 | 0.978/0.315 | 0.920/0.892 | 0.932/0.927 | 0.940/0.923 |
| $\delta$=0.17 | 0.931/0.484 | 0.848/0.936 | 0.862/0.961 | 0.844/0.964 |

TABLE II

ABLATION STUDY OF $\delta$ AND $K$. EACH CELL SHOWS THE PRECISION/RECALL OF THE RECOGNITION.

Table II shows that a smaller $\delta$ tends to increase precision but decrease recall. A good balance is achieved when precision is high ($> 0.95$), while keeping recall at a good level for manipulation efficiency. Simply using the mean feature (K = 1) to represent an object is not ideal as the recall is very low. Increasing K is often beneficial, but the benefit diminishes when K becomes very large (e.g., K=128), while also increasing training time increases.

*5) Object Reconstruction:* Results of shape reconstruction are shown in Fig. 7. The first row shows the ground truth mesh models of objects, and the second row shows reconstructed models after a sequence of manipulation tasks. For objects that are stored multiple times in the dataset due to failures in recognition, this figure only shows the most completed one. Fig. 7 also presents the quantitative evaluation by comparing the distance between the aligned ground truth model $P_{gt}$ and the reconstructed model $P_{rec}$ using Chamfer distance, i.e.

$$D(P_{gt}, P_{rec}) = \frac{1}{|P_{gt}|} \sum_{p_i \in P_{gt}} d(p_i, p_r),$$

where $p_r$ is the closest point in $P_{rec}$ to $p_i$. It can be seen that the reconstructed model is close to the ground truth (Chamfer distance $D < 5mm$ for all four objects) with a few noisy points inside. Such noise is caused by tracking errors when an object is highly occluded, but it does not affect tracking or pose estimation since it will not be rendered by ray casting, and can be further removed by post-processing.

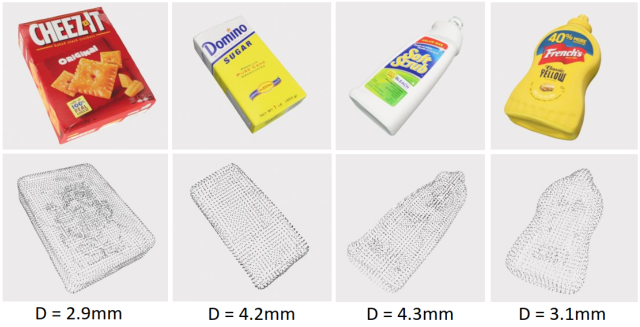

Fig. 7. Qualitative results of object reconstruction after 55 experiments. Ground truth models are shown in the first row and reconstructed models are shown in the second row. $D$ is the Chamfer distance between the ground truth model and the reconstructed model.

## VI. CONCLUSION

This work proposes a robotic system, which utilizes object model reconstruction and reuse for achieving lifelong robot manipulation. By using TSDF representations of objects and a particle filter approach for simultaneous reconstruction and tracking, object models are incrementally reconstructed over a sequence of manipulation tasks. An efficient object dataset construction is proposed to store the color and geometry information of manipulated objects, which makes models reusable and assists future manipulation tasks. Real world experiments show the efficiency of the proposed pipeline.

While this pipeline has been designed to work for most novel rigid objects, it faces challenges with certain objects, such as bowls, which have thin surfaces. This is mainly because the TSDF representation is not suitable for reconstructing thin structures, and may be improved by considering alternatives. Future work will focus on manipulation tasks in cluttered scenes, improving tracking and reconstruction for objects with thin surfaces, and task planning that maximizes information gain while placing novel objects.

# References

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

[2] M. Gualtieri and R. Platt, "Robotic pick-and-place with uncertain object instance segmentation and shape completion," *RA-L*, 2021.

[3] W. Agnew, C. Xie, A. Walsman, O. Murad, C. Wang, P. Domingos, and S. Srinivasa, "Amodal 3d reconstruction for robotic manipulation via stability and connectivity," *arXiv preprint arXiv:2009.13146*, 2020.

[4] C. Mitash, R. Shome, B. Wen, A. Boularias, and K. Bekris, "Task-driven perception and manipulation for constrained placement of unknown objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5605–5612, 2020.

[5] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 116–121.

[6] C. Choi and H. I. Christensen, "Rgb-d object tracking: A particle filter approach on gpu," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1084–1091.

[7] Y. Liu, A. Costantini, R. I. Bahar, Z. Sui, Z. Ye, S. Lu, and O. C. Jenkins, "Robust object estimation using generative-discriminative inference for secure robotics applications," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3240765.3243493

[8] X. Chen, R. Chen, Z. Sui, Z. Ye, Y. Liu, R. I. Bahar, and O. C. Jenkins, "Grip: Generative robust inference and perception for semantic robot manipulation in adversarial environments," 2019.

[9] R. Shome, W. N. Tang, C. Song, C. Mitash, H. Kourtev, J. Yu, A. Boularias, and K. E. Bekris, "Towards robust product packing with a minimalistic end-effector," in *ICRA 2019*.

[10] J. A. Haustein, K. Hang, J. Stork, and D. Kragic, "Object placement planning and optimization for robot manipulators," *IROS*, 2019.

[11] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, 2019.

[12] 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*. IEEE.

[13] 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.

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

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

[16] C. Mitash, A. Boularias, and K. Bekris, "Robust 6d object pose estimation with stochastic congruent sets," *arXiv:1805.06324*, 2018.

[17] Z. Sui, Z. Zhou, Z. Zeng, and O. C. Jenkins, "Sum: Sequential scene understanding and manipulation," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 3281–3288.

[18] C. Choi and H. I. Christensen, "Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 498–519, 2012.

[19] P. J. Besl and N. D. McKay, "Method for Registration of 3D Shapes," *International Society for Optics and Photonics*, 1992.

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

[21] 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.

[22] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[23] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *European conference on computer vision*. Springer, 2014, pp. 584–599.

[24] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[25] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 2011, pp. 127–136.

[26] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[27] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[28] G. Malandain and J.-D. Boissonnat, "Computing the diameter of a point set," *International Journal of Computational Geometry and Applications*, vol. 12, no. 6, pp. 489 – 510, December 2002.

[29] G. Barequet and S. Har-peled, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," in *In Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, 2001, pp. 38–91.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[31] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[32] A. Zeng, S. Song, K. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morena, P. Qu Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3750–3757.

[33] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 1177–1178.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[35] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih, "PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation," *Parallel Computing*, vol. 38, no. 3, pp. 157–174, 2012.

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

[37] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

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