Soft Tracking Using Contacts for Cluttered Objects to Perform Blind Object Retrieval

Sheng Zhong[1], Nima Fazeli[1], and Dmitry Berenson[1]

Abstract—Retrieving an object from cluttered spaces such as cupboards, refrigerators, or bins requires tracking objects with limited or no visual sensing. In these scenarios, contact feedback is necessary to estimate the pose of the objects, yet the objects are movable while their shapes and number may be unknown, making the association of contacts with objects extremely difficult. While previous work has focused on multi-target tracking, the assumptions therein prohibit using prior methods given only the contact-sensing modality. Instead, this paper proposes the method Soft Tracking Using Contacts for Cluttered Objects (STUCCO) that tracks the belief over contact point locations and implicit object associations using a particle filter. This method allows ambiguous object associations of past contacts to be revised as new information becomes available. We apply STUCCO to the Blind Object Retrieval problem, where a target object of known shape but unknown pose must be retrieved from clutter. Our results suggest that our method outperforms baselines in four simulation environments, and on a real robot, where contact sensing is noisy. In simulation, we achieve grasp success of at least 65% on all environments while no baselines achieve over 5%.

Index Terms—Perception for Grasping and Manipulation, Force and Tactile Sensing.

I. INTRODUCTION

THis paper considers the problem of tracking objects in cluttered environments without visual feedback. Applications such as rummaging through a cupboard, refrigerator, or bin for a target object require tracking objects to estimate the pose of the target with limited or no visual sensing. In these scenarios contact feedback is necessary to estimate the poses of objects. A key difficulty is that the objects in these scenarios are movable, requiring the robot to estimate the poses of objects as they move. This is especially challenging because we do not assume we know the shapes of, or even the number of, objects in the environment *a priori*. Thus when two nearby contacts are detected, it is not clear if we have contacted two objects once or one object twice. This ambiguous data association makes tracking much more difficult, as we may need to change the association of past contacts with objects when we observe new data.

Previous work in this area has focused either on single target tracking from contact [1] or on visual tracking of
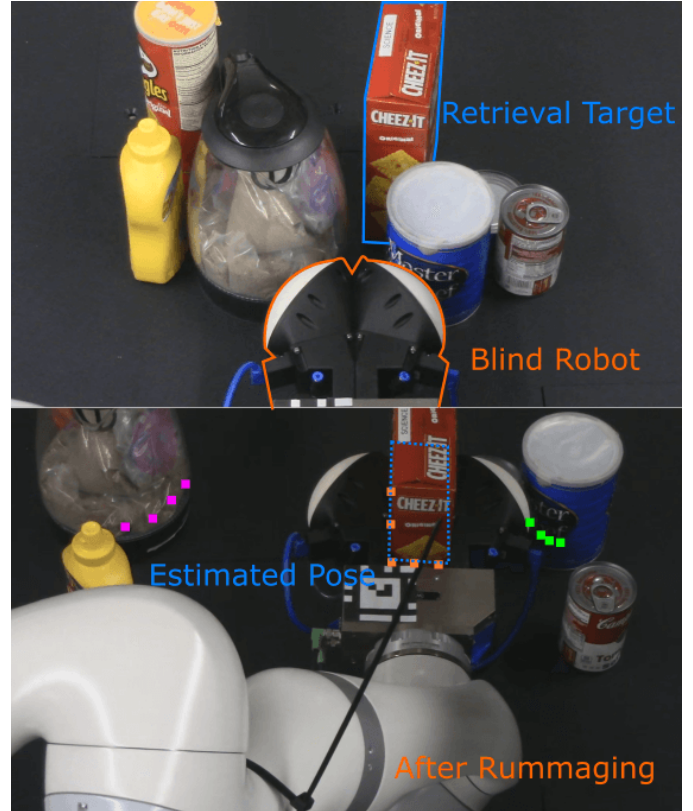
Fig. 1: Initial (top) and after rummaging (bot) cluttered environment with STUCCO allowing us to successfully estimate the pose of the cracker box and grasp it without visual perception. The segmented tracked contact points are shown in different colors.

objects [2], [3]. Work on tracking multiple targets with lidar/sonar/visual data is relevant, but relies on receiving long-range information at high frequency to be effective, which is not the case for contact. To our knowledge, this is the first paper to address the problem of tracking multiple objects using only contact feedback.

The key insight that allows us to tackle this problem is that we can efficiently propagate a belief over contact points without explicit object assignments. We can then sample from that belief to generate hypotheses of contact points and associations. We term this approach "soft tracking" to emphasize its difference with tracking explicit "hard" associations.

Given a model of the pushing dynamics (which can be very simplistic) and an existing method for localizing contact on the surface of the robot [4] (contact isolation), our method, which we call Soft Tracking Using Contacts for Cluttered Objects (STUCCO), tracks the belief over contact point locations and implicit associations using a particle filter. We propagate the belief by sampling whether each contact point moved with probability inversely proportional to its distance to the

latest contact point, then updating the particles to enforce that contact points could not have occurred inside the robot. The best estimate of contact points and a hard association of them to objects, useful for downstream tasks, can be extracted from the belief through our segmentation process.

To show the utility of STUCCO, we demonstrate how it can be used to solve the Blind Object Retrieval (BOR) problem, where a target object of known shape must be retrieved from a planar cluttered environment. We evaluate our method and baselines on both simulated and real (Fig. 1) instances of this type of problem and find that our method achieves at least 65% grasp success on all environments while no baseline achieves more than 5% grasp success on all of them.

II. RELATED WORK

When we know the number of objects in the environment and the mapping between sensing and object is unambiguous, single target tracking methods can be used, such as ones from [5] when vision is available, or the Manifold Particle Filter [1] when contact feedback is available. For single isolated objects, pose and shape estimation has been demonstrated using tactile feedback [6], [7]. Here, we focus on the much more difficult problem when data association is ambiguous and there are an unknown number of objects.

For this problem, computer vision methods have traditionally been used. The relevant problem is termed Multiple Object Tracking, with [8] providing a comprehensive survey of modern methods. In cases where vision is available, its information density makes it attractive as the primary method for object detection and tracking. Our method could be used in conjunction to resolve ambiguities and provide information around occlusions. Indeed, often the robot will occlude the target as it approaches for manipulation.

Outside of computer vision, Multiple Target Tracking is a more common term to refer to the problem and is associated with methods that are agnostic to the information source [9]. Classically, Multiple Hypothesis Tracking (MHT) [10] propagates hypotheses on associations of observations (contacts) to specific targets (objects). A relaxation of allowing association probabilities, instead of fixed associations, is Joint Probabilistic Data Association (JPDA) [11]. While there are ways to limit the combinatorial number of hypotheses to make these methods tractable, in the context of unknown object shapes, the explicit association of contact to objects is difficult. Often only much later do we have sufficient data to discriminate previous associations, so many hypotheses must be kept.

An alternative to explicitly considering associations is propagating the intensity (first-order moment) of the posterior on the number of targets and their states. This class of methods is called intensity filters [9], with the Probability Hypothesis Density (PHD) filter [12] being a notable special case. We compare against an implementation of the PHD filter as a baseline. All these methods were designed with dense information sources in mind (radar, sonar, or cameras), and their assumptions are problematic in the context of blind manipulation. Most significantly, their observation models assume each target generates an observation at each step with some state-independent probability. This is clearly not the case for contact, since we can only observe contact from objects close to the robot. Our method takes inspiration from intensity filters and propagates a belief without explicit associations that exploits the local nature of contact.

Several methods have been proposed for manipulation in cluttered environments. An RGB-D approach [3] demonstrated success in visually segmenting then retrieving objects in clutter. However, the environments they showed allow immediate segmentation of the target object without needing to rummage; additionally, the objects were often well separated from each other. A haptic approach with whole-arm tactile sensing was demonstrated in [13] to successfully reach in clutter. In contrast to their focus on robot control for navigation, allowing them to push movable objects out of the way, we focus on perceiving the objects themselves for downstream manipulation tasks. Similar to [13], our approach benefits from making numerous contacts, as each contact gives us information. While we do not have accurate localization of contact points from whole-arm tactile feedback (which is limited to very few current robots) we are able to perform our tasks using only the estimate of the external wrench at the end-effector.

III. PROBLEM STATEMENT

Let $\mathbf{x} \in$ SE(3) denote the robot end-effector pose. We are given a trajectory $\mathbf{x}_0, ..., \mathbf{x}_T$, during which the robot has made contacts with some objects. We assume that the robot is the only agent in the environment, so objects only move in direct or indirect contact with the robot. Additionally we assume that the robot moves rigidly with no compliance, the robot's geometry is known, the clutter is rigid, and that we are given a dynamics model of how objects transform for some robot motion. Our objective is to track the contact points such that they stay close to object surfaces and are segmented corresponding to the objects they belong to.

Concretely, we define *contact error* (CE) on a contact point to be the smallest euclidean distance from the tracked point to any object surface. The contact error on the trajectory is the average over all contact points. Additionally, we evaluate the segmentation quality using the Fowlkes-Mallows index (FMI) [14], which approaches 0 for random assignments (with increasing number of points) and 1 for perfect assignments.

IV. METHOD

At a high level, our approach enables downstream tasks such as object tracking and retrieval for robots "rummaging" in environments with only tactile feedback. To this end, our approach takes as input the robot trajectory $\mathbf{x}_0, ..., \mathbf{x}_T$ and a set of contact points (each one denoted $\mathbf{p} \in \mathbb{R}^3$) detected during motion. The output is the tracked set of contact points segmented based on object motions. Our method is composed of three elements: contact detection and isolation, soft tracking, and contact point segmentation, of which contact detection and isolation uses prior work while the rest are our contributions. In the following, we provide the details of each component.
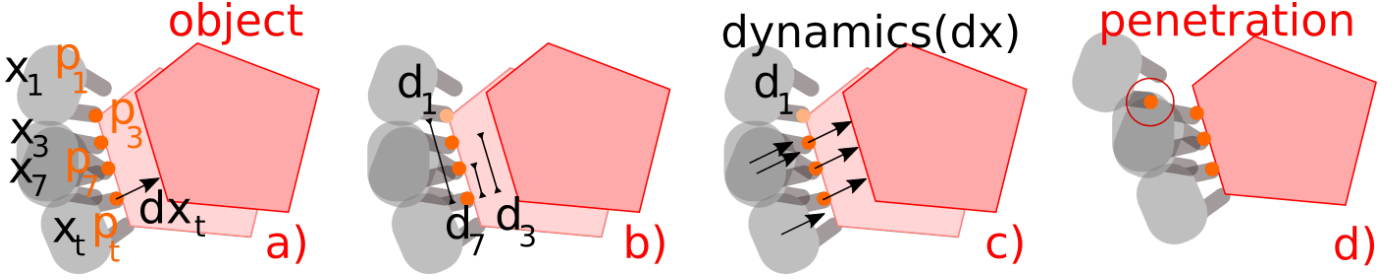
Fig. 2: Prediction (a-c) and update (d) step of one particle. (a) Initial observation of the latest contact point, end-effector pose, and change in end-effector pose. (b) Compute connection probability based on distance to $\mathbf{p}_t$ and sample that connection; point 1 was not connected. (c) Apply dynamics to all connected points. (d) Update step assigns this particle low probability due to $\mathbf{p}_1$ penetrating $\mathbf{x}_3$ and $\mathbf{x}_7$.

A. Contact Detection and Isolation

To detect contact, we utilize the momentum observer [15]. This observer estimates the external wrench applied to the robot (γ) using the robot's joint torques and dynamics model. We detect contact if a specified threshold $\bar{\epsilon}$ is exceeded, similar to [4], [16]:

$$\gamma^T \Sigma_{meas}^{-1} \gamma > \bar{\epsilon} \qquad (1)$$

where Σ_{meas}^{-1} is the precision matrix of the residual, measured by executing random actions in free space.

Once detected, we localize contact on the robot's surface using the Contact Particle Filter (CPF) [4]. This filter iteratively solves for the contact location on the robot's surface assuming a point contact that can transmit forces but no torques – commonly referred to as the Hard Finger approximation [17]. We note that the remainder of our method does not depend on the details of the contact detection and isolation algorithm. As such, advances in this area can be used to extend the functionality of our approach.

B. Soft Tracking

STUCCO maintains a belief over the positions of all contact points. One possibility is to track each contact point independently (e.g. a Kalman Filter to estimate each contact point's position); however, this approach ignores the dependence between contact points that stems from the connectivity between points that belong to the same object. To utilize this basic assumption and represent the belief, we use a particle filter where each particle represents the set of all contact point positions and associated end-effector poses for those contacts. For convenience, we refer to the pair $(\mathbf{p}, \mathbf{x})$ as a point. Alg. 1 shows how we propagate this belief while Fig. 2 depicts one step of our method for a single particle.

Our algorithm is structured in alternating prediction and update steps typical of Bayesian filters. Each particle is propagated independently, thus for simplicity we describe the process in terms of a single particle. However, in practice the process can be parallelized across particles and points.

Our method does not explicitly track point to object associations, like the iFilter from [9] and PHD filter from [12]. Instead, at each step we sample associations to predict motion, the source of the "soft tracking" name. Incorrect associations are propagated forward, resulting in low likelihood for the

---

Algorithm 1: Soft Tracking Using Contacts for Cluttered Objects

Given: N number of particles, l characteristic length, l_p penetration length, `pxpen` point to robot penetration, `pxdyn` object dynamics for change in end-effector pose

1 $P_{1..N} \leftarrow \{\}$ // `particles`
2 $t \leftarrow 0$
3 **while** *robot is in execution* **do**
4   robot executes action $\mathbf{u}_t$
5     $t \leftarrow t + 1$
6   observe $\mathbf{x}_t$ and $d\mathbf{x}_t$, change in end-effector pose while in contact
7   **if** *in contact from Eq. 1* **then**
8         $\mathbf{p}_t \leftarrow$ get latest contact point
9   **for** $n \leftarrow 1$ **to** N **do**
10             $P_n \leftarrow P_n \cup \{(\mathbf{p}_t, \mathbf{x}_t)\}$
11             $d_{0..t} \leftarrow \|\mathbf{p}_{n,0..t} - \mathbf{p}_t\|_2$
12             $p_{connect,0..t} = e^{-d_{0..t}^2/l}$
13             $p_{sample,0..t} \sim \mathcal{U}(0,1)$ indep.
14             $adj \leftarrow p_{sample,0..t} < p_{connect,0..t}$
            // `predict step`
15             $P_{n,adj} \leftarrow \texttt{pxdyn}(P_{n,adj}, d\mathbf{x}_t)$
            // `update step`
16             $\epsilon \leftarrow \sum_{i=1}^{|P_n|} \sum_{j=1}^{|P_n|} \texttt{pxpen}(\mathbf{x}_{n,i}, \mathbf{p}_{n,j})$
17             $p_{obs,n} \leftarrow e^{-\epsilon^2/l_p}$;
18   **else**
19   **for** $n \leftarrow 1$ **to** N **do**
20             $\epsilon \leftarrow \sum_{j=1}^{|P_n|} \texttt{pxpen}(\mathbf{x}_t, \mathbf{p}_{n,j})$
21             $p_{obs,n} \leftarrow e^{-\epsilon^2/l_p}$;
22     $\texttt{ImportanceResample}(P, p_{obs})$
23     $\texttt{ReplaceInconsistentPoints}(P, \texttt{pxpen})$

---

particle when the update step detects inconsistencies arising from some past mistake.

In detail, the prediction step (lines 11 to 15 in Alg. 1) estimates how each contact point moves for an observed change in robot end-effector pose $d\mathbf{x}$. Since we assume that the robot is the only agent in the environment, we only predict motion when in contact. Contact points belong to objects; however, the likelihood of two contact points belonging to the same object scales inversely w.r.t. their relative distance. Line 12 from

---

Algorithm 2: `ReplaceInconsistentPoints`

Given: $P_{1..N}$ particles, pxpen point to robot penetration

1   **for** $n \leftarrow 1$ **to** N **do**
2   **for** $j \leftarrow 1$ **to** $|P_n|$ **do**
      // how inconsistent each point is
3       $\epsilon_j \leftarrow \sum_{i=1}^{|P_n|} \text{pxpen}(\mathbf{x}_{n,i}, \mathbf{p}_{n,j})$
4     $incon \leftarrow \epsilon > 0$ // indices
5   **for** $j \in incon$ **do**
6       $k \leftarrow \arg\min_{i \in \neg incon} ||\mathbf{p}_{n,i}, \mathbf{p}_{n,j}||_2$
7       $P_{n,j} \leftarrow \{\mathbf{p}_{n,k}, \mathbf{x}_{n,k}\}$ // as well as weight

---

Alg. 1 encodes this using a characteristic length l parameter. To determine each contact point's adjacency (belonging to the same object) to the most recently encountered one, we randomly sample proportional to the likelihood provided by their relative distance. Contact points on the same object and their associated end-effector poses move together according to the given dynamics function pxdyn on line 15. Thus a single contact point $\mathbf{p}_{n,i}$ at time $i < t$ would only move if it is adjacent to $\mathbf{p}_t$.

The update step in lines 21 and 17 evaluates the likelihood of each particle in realizing the most recent observation, $p_{obs,n}$. We utilize the fact that contact can only occur on the robot surface to evaluate each particle. To this effect, we define the function pxpen$(\mathbf{x}, \mathbf{p})$ which outputs 0 if $\mathbf{p}$ is outside the robot when the end-effector is in pose $\mathbf{x}$, and otherwise $\min_{\mathbf{p}_s \in S(\mathbf{x})} ||\mathbf{p} - \mathbf{p}_s||_2$, where $S(\mathbf{x})$ is the set of points on the robot surface at $\mathbf{x}$.

When in contact, the predicted movement of the contact points may result in penetration between any pair of $\mathbf{x}$ and $\mathbf{p}$. Thus in line 16 we sum the penetration between all pairs in the particle. In contrast, in line 20 when out of contact, we only need to evaluate the observed $\mathbf{x}_t$ against all contact points since there is no predicted movement.

In both cases, the computation of $p_{obs,n}$ parallels our computation for adjacency during the prediction step, with a separate length parameter l_p scaling with the expected contact isolation error (actual contact point's distance to the estimated contact point). A lower value will result in more false positives of penetration while a higher value will result in more false negatives. With p_{obs}, we perform the standard particle filter importance resampling (line 22 of Alg. 1).

Even after resampling, particles may still have penetration inconsistencies. This could be due to none of the particles sampling a consistent prediction, or from errors in the contact isolation or contact dynamics. To address this, we call `ReplaceInconsistentPoints` after resampling, detailed in Alg. 2. A point is inconsistent and discarded if its $\mathbf{p}$ incurs any penetration, and replaced with the closest consistent point in terms of $\mathbf{p}$ Euclidean distance in lines 6 and 7 of Alg. 2.

C. Segmenting into Objects

Many useful applications of tracking require a single estimate of the contact points as well as hard assignments to

---

Algorithm 3: Segment a particle into objects

Given: P_n a single particle, l characteristic length, α probability threshold for each edge

1   **for** $i \leftarrow 1$ **to** $|P_n|$ **do**
2   **for** $j \leftarrow 1$ **to** $|P_n|$ **do**
3       $d \leftarrow ||\mathbf{p}_{n,i} - \mathbf{p}_{n,j}||_2$
4       $A_{i,j} = e^{-d^2/l} > \alpha$

5   **return** connected components of adjacency matrix A

---

objects. To achieve this, our method selects the most likely particle (MAP) according to the particle weights (updated each step with p_{obs}). Alg. 3 details how the MAP particle is segmented into groups of contact points that are estimated to belong to the same object.

Similar to line 12 from Alg. 1, we compute the connection probability and compare it against a threshold α to determine if an edge between two points exists. The resulting adjacency matrix A describes a graph over all the points, from which we find connected components. Each connected component is an object. Our segmentation is a form of agglomerative clustering (such as with BIRCH [18] or DBSCAN [19]), which is well suited for irregular and elongated shapes, such as the set of points belonging to surfaces of objects. A common weakness of these methods is combining two clusters when noise or an error creates a data point between them. Our update process mitigates this weakness when the robot's configuration overlaps with the erroneous contact point (depicted in Fig. 7) and it is deemed inconsistent, but it remains an issue if our robot does not explore that location.

V. EXPERIMENTS

In this section, we evaluate and benchmark the performance of our approach on: i) tracking and segmentation of contact points under a "blind rummaging" policy; and ii) a downstream task of Blind Object Retrieval (BOR) – both in cluttered environments. To this end, we first describe our baselines. Next, we describe the robot environment and training data. Then, we formalize the downstream BOR task that uses contact tracking. Lastly, we quantitatively evaluate our method and baselines on BOR in simulated and real-world cluttered environments.

For all tasks, we used the following pxdyn:

$$\text{pxdyn}(P_n, d\mathbf{x}) = \{(\mathbf{p} + F(d\mathbf{x}), \mathbf{x} + d\mathbf{x}) | (\mathbf{p}, \mathbf{x}) \in P_n\} \quad (2)$$

where F extracts the linear translation of the pose change. This motion model implicitly assumes that objects translate together with the robot when in contact without rotation. A more sophisticated motion model may be used if object properties such as size, shape, or pressure distribution are known *a priori*; however, this is not the case in our experiments. Here, we demonstrate that our method is able to partially mitigate errors from this approximation since some of its predictions result in contact point penetration.

To speed up our method, we implemented Alg. 1 to process each particle in parallel. In particular, pxpen was implemented as a parallel lookup of a pre-computed discretized (resolution 1mm) signed distance field of the end-effector in

link frame. The transform of contact points from world to link frame was also implemented to be parallel.

For contact isolation, we used the Single-CPF from [4] which assumes each detected contact occurred at only one contact point. We note that while there are inherent ambiguities in isolating contact from externally-applied wrenches, our method is robust to these errors. On the real robot, we additionally consider contacts detected by each of the two soft-bubble sensors [20] (seen in Fig. 1). This distributed tactile sensing modality significantly mitigates ambiguities from using only wrench estimates.

A. Baselines

We compare against baselines that maintain a single estimate of all $\mathbf{p}$, clustering on $\mathbf{p}$ at each step in contact, and applying the dynamics function to all points in the same cluster as $\mathbf{p}_t$. The baselines differ in their clustering methods, with BIRCH [18] and DBSCAN [19] by default not needing to specify the number of clusters. A k-means baseline was implemented that starts with a single cluster and increases the cluster number by 1 if doing so reduces the inertia (what k-means minimizes) sufficiently. Additionally, we consider a Gaussian Mixture (GM) implementation of the PHD filter [12]. As introduced in Section II, this method propagates the intensity (first-order moment) of the posterior on the number of objects and their positions. The intensity is integrated over to extract discrete targets (objects), which we clustered the contact points to using nearest neighbours then propagated in the same way as the clustering methods.

B. Training Set for Tuning

For simulation, we use a floating Franka Emika (FE) gripper from the PANDA arm (see Fig. 4) with a fixed height and constrained orientation. The gripper is simulated in PyBullet [21] and takes discrete action steps in the form of desired dx, dy, with a maximum per step movement of 0.03m along each dimension. Each simulation time step is 1/240s, and we moved slowly to avoid bouncing objects off the robot. The residual γ used for contact detection and isolation here is the measured force torque on the gripper provided by the simulator.

Note that our method takes contact points as input and is not limited to planar systems. However, restricting to planar motion simplifies the data collection, contact isolation, and the downstream task of BOR.

Our training set consists of 40 trials of randomized start and goal positions for each of the 4 environments depicted in Fig. 3. We generated trajectories using a greedy controller that entered a random walk of length 6 upon contact. Trajectories that were in contact less than 5% of the time were discarded without replacement, yielding a total of 129 valid trajectories.

Tuning consisted of parameter sweeping to maximize median FMI and minimize median CE on the whole training set. For our method, the primary parameter to tune was the characteristic length l, which was larger on the real robot to handle a large kettle. See Tab. I for our tuned parameters. BIRCH was tuned to have threshold 0.08, DBSCAN was tuned

Fig. 3: Training environments in simulation. Immovable walls are colored grey while the darker the movable object is, the more massive it is.

TABLE I: STUCCO parameters for Blind Object Retrieval.

Parameter	sim	real
l characteristic length	0.02	0.006
l_p penetration length	0.002	0.002
$\bar{\epsilon}$ residual threshold	1	5
N number of particles	100	100
α connection threshold	0.4	0.4

to have eps 0.05 and minimum neighbourhood size of 1, k-means was tuned to need an inertia improvement of 5 times to increase the number of clusters, and the GMPHD filter was tuned to have birth probability of 0.001, spawn probability of 0, and detection probability of 0.3.

The tuned performances of all methods on the training set are shown in Fig. 5 (top), where our method outperforms all baselines in CE. Since we are interested in manipulation in clutter, Fig. 5 (bottom) shows the performance on runs that had ambiguous contact assignments. For each step, this was computed using the minimum distance from the robot to the second closest object, with an ambiguity score of 1 corresponding to a distance of 0 and a score of 0 corresponding to a distance of 0.15m or more. The bottom figure shows runs with an average ambiguity of at least 0.3. On these, our method outperforms the baselines in both FMI and CE by an even larger margin, demonstrating that our method is well suited for clutter.

C. Blind Object Retrieval

We present the problem of Blind Object Retrieval: pose estimation and grasping of a target object with known geometry using no visual perception. To perform this task, the robot rummages in clutter to collect contact points that it can segment into objects using our method. Using the segmented objects, the robot runs iterative closest point (ICP) [22] between the set of contact points of each object and model points sampled from the known surface of the target object. ICP is run 30 times from random initial poses and the object with the lowest variance in position estimation is selected (see Fig. 6). An important source of variance is uncertainty in orientation due to contacts being unevenly distributed across the object surface. From the ICP estimates of the selected object, we further select the one that penetrates $\mathbf{x}_t$ the least, and on ties choose the lowest ICP matching error.

To evaluate success, we attempt a grasp at the estimated pose after executing a given rummaging policy. Grasp success is an important metric to evaluate on for two reasons. First, it avoids the need to combine position and orientation errors. These metrics are typically combined using radius of gyrations which may not be available during run time. Second, it does not penalize small pose errors that may not be relevant to the
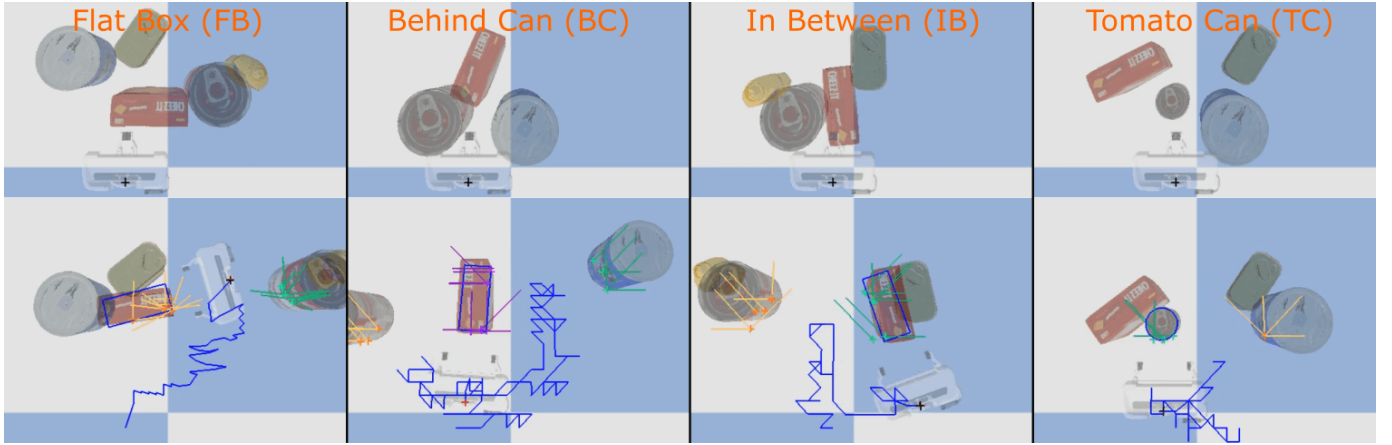
Fig. 4: Simulated BOR task in 4 different environments, with each starting condition (top) and after executing actions (bottom), with the trail in blue. Overlaid is STUCCO's best estimate of segmented objects, with propagated contact points as crosses and associated actions taken with a different color for each object. The pose estimate of the target object is represented by a blue outline.
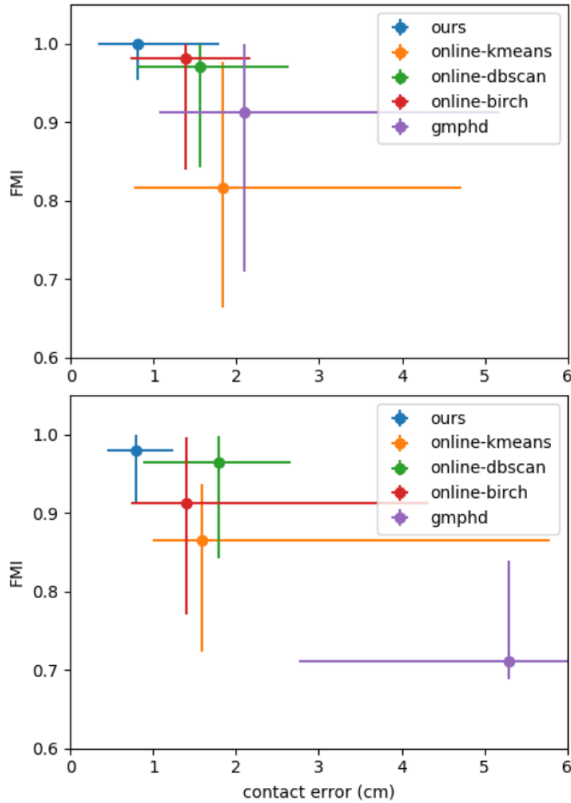

Fig. 5: Tracking metrics evaluated on the training set, with the median plotted and error bars indicating 20-80th percentile. (top) Results for the whole data set, and (bot) results for only runs with an ambiguity score of at least 0.3. Ideal performance is an FMI of 1 and CE of 0 corresponding to points in the upper left.
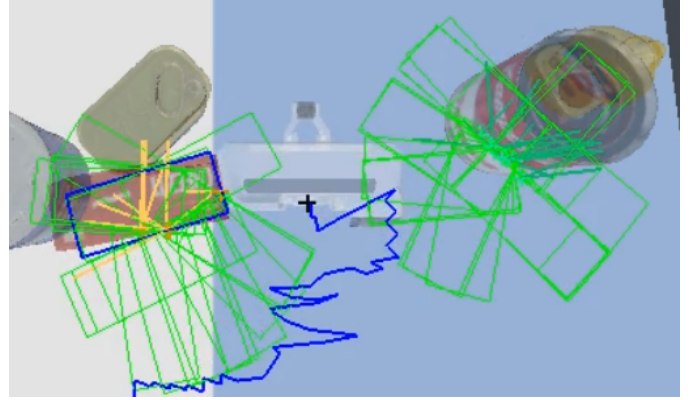

Fig. 6: Iterative closest point pose estimates from 30 random starts plotted in green. ICP run on the left set of points (corresponding to the cracker box) result in lower positional variance in the estimate than the ICP results run on the set of points corresponding to the can (right).

A grasp was successful if it closed on the two long sides of the box and not on a corner, or around more than half of the tomato can. The control sequence was manually created to make contact with multiple objects that were initially close together, while making sufficient contacts to identify the target. When replayed, each action was perturbed with uniformly random noise of up to ± 0.5mm (compared to max action step of 30mm).

We performed 20 runs of each task (same random seed used for each baseline so they are evaluated under the same actions), with the statistical comparison of our method against baselines summarized in Tab. II. Our method was the only one that achieved 65% grasp success or higher on all tasks, and achieved significantly lower contact error than all baselines.

Our method also achieved better FMI than baselines. The overall lower FMI scores compared to the training set seen in Fig. 5 attests to the difficulty of the BOR tasks. Importantly, the lowered FMI (indicating more assignment errors) for our method did not translate to increased CE. This is a key strength of our method and can be attributed to the particle update penalizing contact penetration

task and can safely be ignored while penalizing those beyond a gripper dependent threshold that will always result in grasp failure.

D. Blind Object Retrieval in Simulation

In simulation, we designed 4 cluttered environments (see Fig. 4) with YCB objects [23], with the target being the cracker box in FB, BC, and IB, and the tomato can in TC.
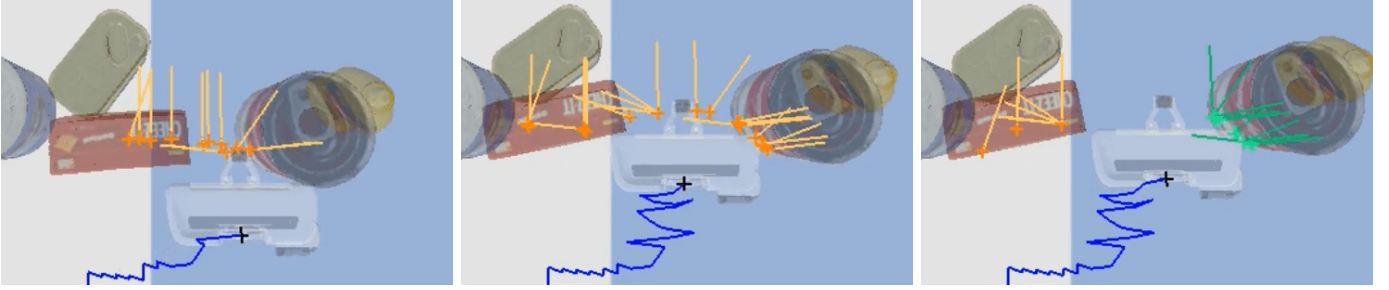
Fig. 7: Steps during a BOR run with (left) initially wrong associations of contact points to objects, (mid) moving to right before entering the gap between the objects, and (right) resolving the previous ambiguity from moving through the gap and penalizing particles with points in between. Tracked contact points are in orange for the first object and green for the second one.

and also the `ReplaceInconsistentPoints` function. Specifically, the action sequences often moved back and forth between two objects, eventually opening a gap and moving through it, as captured in Fig. 7. The oscillatory motion initially left particles with contact points between the gap, but after moving through it, the update process assigned high likelihood to particles that separated the contact points to either side. Additionally, `ReplaceInconsistentPoints` replaced remaining points in between with points on a side while occasionally making an assignment error.

The baselines have no mechanisms for correcting associations in hindsight, so the lower FMI translated to higher CE and lower grasp success. However, despite the high errors in CE, the baselines sometimes achieved grasp success due to the ICP eliminating many wrong pose estimates.

E. Real Robot Blind Object Retrieval

We applied our method on a real 7DoF KUKA LBR iiwa arm with two soft-bubble tactile sensors [20] on the gripper for the BOR task depicted in Fig. 1. Similar to simulation, we restricted our motion to be planar, with a max step of 20mm implemented using a Cartesian impedance controller. KUKA's on-board software estimated the externally applied wrench at the end-effector using the measured joint torques. To accommodate the limited sensitivity of this measurement, We filled the YCB objects to increase their mass and reduce the effect of measurement noise.

TABLE II: Quantitative comparison of our method against baselines on 20 runs of blind object retrieval in different simulated cluttered environments and 5 runs on the real environment in Fig. 1. GS is grasp success (%), and CE is contact error (cm). Top values per category are in bold while standard deviations are in parentheses. FB, BC, IB, and TC are depicted in Fig. 4.

task		ours	BIRCH	DBSCAN	k-means	GMPHD
	GS	**70**	5	5	45	25
FB	FMI	**0.69 (0.08)**	0.62 (0.07)	0.66 (0.10)	0.63 (0.05)	0.56 (0.05)
	CE	**0.72 (0.25)**	4.63 (0.53)	3.52 (1.54)	1.49 (0.40)	3.90 (0.61)
	GS	**80**	0	10	5	0
BC	FMI	**0.92 (0.05)**	0.84 (0.09)	0.89 (0.02)	0.83 (0.04)	0.57 (0.04)
	CE	**1.33 (0.13)**	6.49 (0.36)	6.56 (0.31)	7.04 (0.41)	7.33 (0.44)
	GS	**65**	0	60	10	0
IB	FMI	**0.78 (0.07)**	0.71 (0.05)	0.75 (0.06)	0.47 (0.23)	0.46 (0.06)
	CE	**1.04 (0.33)**	3.27 (0.36)	2.31 (0.58)	4.84 (1.81)	5.62 (0.85)
	GS	**85**	0	0	25	20
TC	FMI	**1.00 (0.00)**	0.79 (0.07)	**1.00 (0.00)**	0.54 (0.17)	0.54 (0.17)
	CE	**0.31 (0.90)**	4.20 (0.22)	4.14 (0.09)	6.23 (0.72)	8.09 (0.22)
Real	GS	**100**	20	0	20	0

Contact isolation was performed independently by the left and right soft-bubble sensors, then by the CPF if neither of them detected contact. Each bubble had a depth camera inside that measured surface deformation. Pixels with deformations greater than 4mm were considered deformed. We then averaged all deformed pixel coordinates and projected that point to the camera frame then rigidly transformed it to produce a contact point in the world frame. Due to the deformable nature of the sensors, we adjusted `pxpen` to ignore the first 10mm of penetration.

To extend Alg. 1 to multiple contact points per step, each contact point's distance in line 11 is measured against their closest new contact point, and dynamics in line 15 is performed with their associated $d\mathbf{x}$. Note that $d\mathbf{x}$ for each new contact point may be distinct due to making contact at different times during the action.

VI. DISCUSSION

Proprioceptive and tactile driven object state-estimation is an important functionality for autonomous robotic systems in highly unstructured environments. Here, we discuss important extensions that can further generalize our method to more challenging instances of "blind" object state-estimation for downstream tasks such as object retrieval. These extensions include generalizing beyond point contacts and generating the rummaging policies.

A. Generalizing Beyond Object Translation

Our update step can use inconsistency in object motion to correct for errors in the dynamics function, such as assuming that objects translate without rotation. However, it is unable to handle higher dimensional pose changes such as those induced by toppling or deforming. To address this, more rich information beyond point contacts can be extracted from each contact (e.g., incipient slip from the soft-bubbles) together with more sophisticated object dynamics models.

B. Generalization Beyond Single Point Contacts

Alg. 1 can generalize beyond single point contacts without major changes. Indeed, as shown in our real robot experiment with essentially three contact detectors, we can easily generalize to multiple contacts per step. The soft-bubble sensors provide rich contact information that we hope to exploit in

future work. Advanced representations of contact patches such as meshes and non-uniform rational B-splines (NURBS) [24] could directly replace or exist alongside contact points in Alg. 1 as long as we have efficient pairwise distance functions between them.

C. Rummaging Policy

In this paper, our method assumed a prescribed action sequence that makes sufficient contact with our target object to uniquely identify it. Future work could generate this policy using active perception. For example, after estimating the object pose using a probabilistic method (or approximations such as using multiple runs of ICP to estimate the target pose and associated uncertainty), an action sequences can be chosen to reduce uncertainty while maintaining distance from other contact point sets to minimize ambiguity.

VII. CONCLUSION

We presented STUCCO, a contact tracking method that maintains a belief over contact point locations to enable corrections in hindsight. The method is based on the basic assumptions that points closer together are more likely to be on the same object, and that contact only occurs on the surface of the robot. We showed that it performs well on a variety of Blind Object Retrieval tasks in clutter and demonstrated its application on a real robot. Specifically, it is capable of handling cases where contact is initially made on different objects close together, and later correct their tracking when they are moved apart. In contrast, we showed that clustering and the PHD filter baselines struggle in these scenarios. Finally, the failure of baselines that maintain a single estimate of the contact points on our tasks suggests that it is beneficial to maintain a belief over them to allow corrections in hindsight. Future work will focus on representing rich contact patches and generating the rummaging policy.

REFERENCES

[1] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pose estimation for planar contact manipulation with manifold particle filters," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 922–945, 2015.

[2] M. Betke, D. E. Hirsh, A. Bagchi, N. I. Hristov, N. C. Makris, and T. H. Kunz, "Tracking large variable numbers of objects in clutter," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[3] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 437–451, 2018.

[4] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5062–5069.

[5] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[6] K.-T. Yu and A. Rodriguez, "Realtime state estimation with tactile and visual sensing for inserting a suction-held object," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1628–1635.

[7] S. Suresh, M. Bauza, K.-T. Yu, J. G. Mangelson, A. Rodriguez, and M. Kaess, "Tactile slam: Real-time inference of shape and pose from planar pushing," *arXiv preprint arXiv:2011.07044*, 2020.

[8] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, p. 103448, 2020.

[9] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*. Artech House, 2013.

[10] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[11] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[12] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[13] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *The International Journal of Robotics Research*, vol. 32, no. 4, pp. 458–482, 2013.

[14] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[15] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 999–1004.

[16] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[17] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer handbook of robotics*. Springer, 2016, pp. 955–988.

[18] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," *ACM sigmod record*, vol. 25, no. 2, pp. 103–114, 1996.

[19] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[20] N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasey, T. Ikeda, and R. Tedrake, "Soft-bubble grippers for robust and perceptive manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9917–9924.

[21] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[22] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[23] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

[24] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 1996.