Force-based Simultaneous Mapping and Object Reconstruction for Robotic Manipulation

Joao Bimbo, *Member, IEEE*, Andrew S. Morgan, *Student Member, IEEE*, Aaron M. Dollar, *Senior Member, IEEE*

***Abstract*—This paper presents an algorithm to simultaneously reconstruct the geometry of an unknown object and its environment via physical interactions. Applications involving highly cluttered or occluded workspaces prevent the effective use of vision. To address some of the challenges that arise, we propose an approach that instead utilizes force and torque measurements at the robot end-effector to solve for possible contact locations and probabilistically determine the occupancy likelihood on a 3D map. Our procedure constructs two occupancy maps: one fixed that represents the environment and another map that moves with the robot end-effector and reconstructs the grasped object shape, where each map informs the probability updates on the other. The algorithm is applied and tested on two scenarios: retrieving a tangled object from a scene and reconstructing the geometry of an object. We compare the results against a configuration space planner and a reinforcement learning algorithm, with our method requiring fewer collisions with the environment to extract the object.**

***Index Terms*—Manipulation Planning, Robot Grasping, Mapping, Perception for Grasping and Manipulation**

I. Introduction

ROBOTS operating in unknown or uncertain environments may be required to manipulate objects whose geometry is not known *a priori*. In order to navigate these environments, a robot must make sense of the geometry of the grasped object and of the environment surrounding it. In this paper we present a method that uses force information at the end-effector to reconstruct the grasped object and map the robot environment through a probabilistic approach. Two 3D maps are created and each informs the update on the other: one map is fixed to the world and represents the obstacles in the environment, and the other moves with the robot end-effector, reconstructing the geometry of the grasped object. When contact between the object and the environment occurs, the algorithm solves for the possible locations of the collision and updates the probability that the relevant voxels (the three-dimensional analogue of a pixel) are occupied. Moreover, when the robot moves in free space, the occupancy of the voxels is also modified, since we know that no collision is taking place.

We show that it is possible for a robotic system to navigate uncertain environments, even when reduced sensing information is available, and demonstrate the utility of the proposed approach. Possible applications of this work lie in the areas of search and rescue, if a robot needs to declutter a scenario where there is debris, or removing objects from a pile (such as in a waste sorting or recycling facility). We present the rationale for the two probability laws that are implemented, and apply the method in two contexts: object retrieval and reconstruction. We compare it to both traditional planning and reinforcement learning methods, and validate the approach on a 6 DOF manipulator with force-torque sensing capabilities at the end-effector.

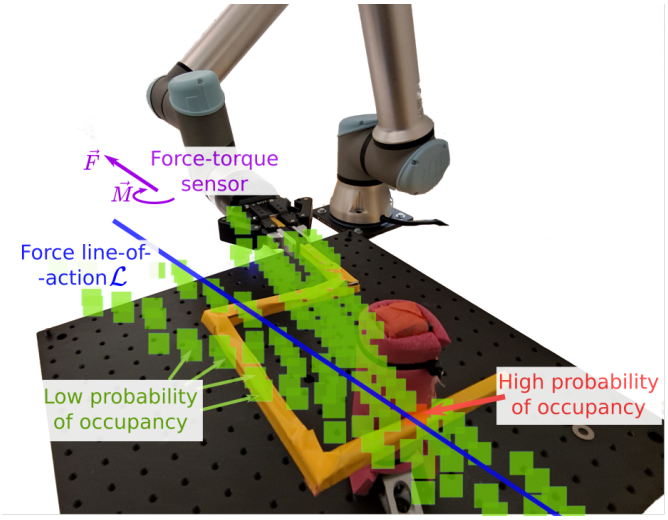


Fig. 1. 3D maps of the environment and grasped object can be built using the line-of-action of interaction forces, obtained from force-torque measurements.

A. Related Work

1) Contact Sensing: The idea of identifying the contact location from force measurements can be traced back to Bicchi *et al* [1], who named this approach *intrinsic* contact sensing. In [1], the contact location on a surface is resolved through solving the force-moment balance and the surface equation, assuming a soft finger model and a convex surface. This idea was further developed by Liu *et al* [2], who extended it to deformable surfaces and, more importantly, Kurita *et al* [3] who developed a method to compute the contact location on an arbitrary shaped surface using a series of measurements. In [3], a concave shell was placed over a mobile robot and the line-of-action of the collision force was intersected with the shell geometry. In cases where there are multiple intersections, the contact location is determined after a number

Manuscript received: September, 9, 2021; Revised January, 6, 2022; Accepted January, 31, 2022.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments.

The authors are with the Mechanical Engineering and Materials Science Department of Yale University. Contact: {joao.bimbo, andrew.morgan, aaron.dollar}@yale.edu

This work was supported by the US National Science Foundation under Grants IIS-1928448 & IIS-190068

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

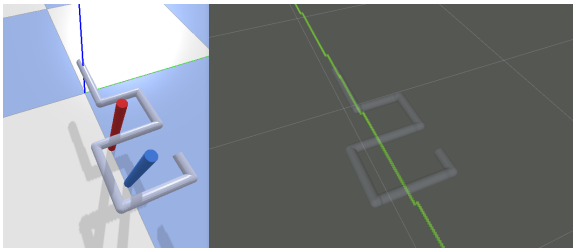
of measurements, choosing the location where the position variability is lowest. Research on detecting a collision on the body of a serial robot has also been the subject of extensive research [4], [5]. These approaches typically use the torque measurements at the robot joints, and determine the location using approaches such as the Particle Filter. The limitation in these approaches, however, is that the shape of the end-effector needs to be known *a priori*, since the force and torque measurements can only, and under some assumptions, provide information about the line-of-action of a force. Determining the actual contact location on that line requires additional information, such as the shape of the surface in contact.

2) Mapping and Navigation: Navigating through an unknown environment is a problem typically studied within the field of mobile robotics. The SLAM problem consists precisely of reconstructing an unknown map and locating the robot within it, with most approaches using either cameras [6], [7] or LIDAR [8] sensors. In [9], the authors introduce the "Blindfolded Traveler's Problem" where a serial robot navigates an unknown environment based only on contact sensing, obtained from joint torques. It tests three approaches for constructing the planning graph: one uses a Bayesian Filter (Manifold Particle Filter), one uses the sets of occupied and free voxels (Collision Hypothesis Sets), and finally a Mixture of Experts of the two. These approaches require the knowledge of the robot geometry in order to reconstruct the environment. Other approaches that aim to navigate a robot on a cluttered, unknown environment usually rely on tactile sensors distributed on the robot body [10], [11], [12], such that the contact location(s) are readily available, or machine learning with human demonstrations to navigate a social environment [13], or reinforcement learning with tactile data for object insertion [14].

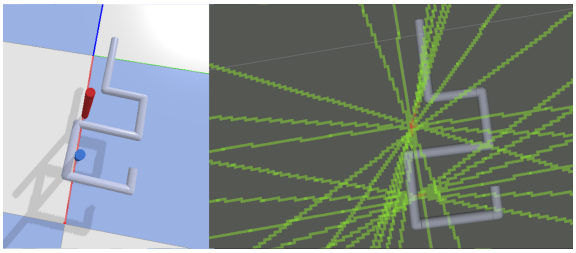
3) Object Reconstruction and Pose Estimation: The ability to perceive contact locations on the environment can also be used to reconstruct the geometry of an object. In [15], tactile sensing was used to reconstruct the object geometry as a point cloud. More detailed reconstruction can be achieved using high-resolution tactile images that provide a detailed rendering of the object geometry [16]. In [17], tactile sensing and vision were combined in a fully data-driven fashion to reconstruct the geometry, whereas in [18], a glove with electromagnetic trackers was used. Collisions between a grasped object and the environment has been used to estimate the pose of the object [19], but required both the geometry of the object and of the environment to be known *a priori*.

B. Problem Presentation

This paper introduces an algorithm that simultaneously maps the environment and reconstructs a grasped object using the contact forces that arise during the interaction. It builds two three-dimensional probabilistic occupancy maps, each of which informs the construction of the other. Two types of events are used to update the maps: *1)* when the robot measures an external force (there is a collision between the grasped object and the environment), the possible contact locations are determined and the occupancy probability of the relevant voxels on each map is computed; *2)* when the robot moves in free space and no collision is detected, this information is also used to update the maps by reducing the occupancy probability of overlapping voxels. The probability laws for each of these cases are based on an analysis of the possible contact locations and the prior occupancy probabilities associated with them on each map. We name this approach SMORE (Simultaneous Mapping and Object REconstruction).



(a) Discretization along the force line-of-action $\mathcal{L}$



(b) Obstacles are detected after a number of contacts

Fig. 2. Simulated environments of a 'S'-shaped object and two vertical peg obstacles. (a) Collisions between the object and an obstacle generate a force, and its line-of-action can be computed. (b) As multiple collisions occur along different line-of-actions, we can probabilistically determine the likelihood of occupancy as these lines intersect.

We present the two probability laws in the next section, and find a number of applications in Section III to illustrate the proposed method, comparing the results with other possible approaches. Section IV discusses the results and presents future research directions.

II. Methods

A. Method Overview

The idea behind the proposed method is to construct and populate two 3D maps. Map $\mathcal{M}_w$ is fixed to a world frame and describes the obstacles in the environment, while $\mathcal{M}_o$ is moving with the robot end-effector and reconstructs the geometry of the grasped object. Each map is defined by a pose $T \in SE(3)$ and set of voxels that discretize the 3D space: $\mathcal{M} = \{p_1, p_2, ..., p_n\}$, with each voxel having a fixed size and containing its spatial coordinates and a probability of occupancy: $p = \{x, y, z, P(O^p)\}$. Since we will be performing the same operations on both maps, these are interchangeable and we can simply refer to one map as $\mathcal{M}$ and the other as $\mathcal{M}'$. At any given time, the point $p' \in \mathcal{M}'$, coincident with point $p \in \mathcal{M}$, can be obtained by multiplying it by the homogeneous transformation T between $\mathcal{M}$ and $\mathcal{M}'$, resolved from the robot's forward kinematics.

$$p' = T^{-1} \cdot p \tag{1}$$

When contact occurs and a force is measured at the robot end-effector, the occupancy probability of the possible contact locations increases, and when the robot is moving in free space, this probability decreases. The probability laws for each of these cases take into account the current knowledge of both maps to continuously update the belief in the occupancy of the relevant voxels. On contact, a higher occupancy probability of a possible contact location p results in a higher confidence that the contact happened at this point, in turn increasing the occupancy probability of the corresponding voxel p'. Similarly, in free space, the more confidence we have that p is occupied, the bigger the decrease in the probability of occupancy of p'.

B. Probability Laws

1) Probability law for contact: When the grasped object collides with its environment, the resulting force and torque at the end-effector can be used to calculate the line-of-action of that force according to the formula:

$$\vec{r} = \frac{\vec{F} \times \vec{M}}{\vec{F} \cdot \vec{F}} \text{ , and } h = \frac{\vec{F} \cdot \vec{M}}{\vec{F} \cdot \vec{F}}, \tag{2}$$

where $\vec{F}$ and $\vec{M}$ are the measured force and moment, $\vec{r}$ is a vector to a point on that line, and h is the magnitude of the moment around the net force. It should be noted that when using a *soft finger* contact model, the admissible moment transmitted through the contact is normal to the surface [1], whereas in (2) the moment is parallel to the resultant force. Discretizing the 3D space, we can find the subset of voxels $\mathcal{L} \subset \mathcal{M}$ that are intersected by this line. Figure 2 shows the line-of-action on a 3D map as the grey object collides with the red obstacle. An equivalent set of voxels $\mathcal{L}'$ is computed for a coincident line expressed in the frame of $\mathcal{M}'$.

We start by making the distinction and stating the relationship between the probability of a voxel at point p being occupied $P(O^p)$ and the probability that the contact happened at that point $P(C^p)$. In order for the collision to happen at p, both voxel p and its coincident point p' must be occupied. If we assume that there is at most a single contact location at any given time, then:

$$P(C^p) \propto P(O^p) \cdot P(O^{p'}) \tag{3}$$

We find the new probability of occupancy of each voxel p using the law of total probability. We obtain it by summing the probability of the contact happening at that location and the probability that the voxel is occupied, but the contact happened elsewhere:

$$\begin{aligned} P(O^p)_{t+1} =& P(O^p \mid C^p) \cdot P(C^p) \\ &+ P(O^p \mid \neg C^p) \cdot P(\neg C^p) \end{aligned} \tag{4}$$

Given the current occupancy probability of the voxels in $\mathcal{L}$ and $\mathcal{L}'$, we first compute the total probability of all admissible cases. Since we know that a contact occurred somewhere along line $\mathcal{L}$, there is a location where both p and p' are occupied. Thus the admissible cases can be obtained from the union of the probability of contact anywhere on that line:

$$\begin{aligned} \bigcup_{n \in \mathcal{L}} P(C^n) = & P(C^1) + P(C^2) + ... + P(C^k) \\ & - P(C^1 \cap C^2) - P(C^1 \cap C^3) - ... \\ & + P(C^1 \cap C^2 \cap ... C^k) \end{aligned} \tag{5}$$

The calculation in (5) is combinatorial and thus computationally very expensive as the number of voxels in $\mathcal{L}$ increases. It is simpler to find the probability that no collision happens anywhere in $\mathcal{L}$ and subtract it from 1, according to De Morgan's laws:

$$\bigcup_{n \in \mathcal{L}} P(C^n) = 1 - \prod_{n \in \mathcal{L}} (1 - P(O^n) \cdot P(O^{n'})) \tag{6}$$

The probability of the contact happening at point p is then the probability of both p and p' being occupied, divided by the total probability obtained in (6):

$$P(C^p) = \frac{P(O^p) \cdot P(O^{p'})}{\bigcup_{n \in \mathcal{L}} P(C^n)} \tag{7}$$

The second part of (4) is the probability of p being occupied and the contact happening somewhere else in $\mathcal{L}$, except p. We use the result in (6) to calculate this probability:

$$\bigcup_{n \in \mathcal{L} \setminus \{p\}} P(C^n) = 1 - \frac{\prod_{n \in \mathcal{L}} (1 - P(O^n) \cdot P(O^{n'}))}{1 - P(O^p) \cdot P(O^{p'})} \tag{8}$$

Combining the result in (8) with the probability that p is occupied and p' is free, and dividing by the possible cases in (6) yields

$$P(O^p \cap \neg C^p) = \frac{\bigcup_{n \in \mathcal{L} \setminus \{p\}} P(C^n)}{\bigcup_{n \in \mathcal{L}} P(C^n)} \cdot P(O^p) P(\neg O^{p'}) \tag{9}$$

The probability of occupancy in (4) can then be obtained by combining (7) and (9), and the fact that $P(O^p \mid C^p) = 1$:

$$P(O^p)_{t+1} = \frac{PP' + \left(1 - \dfrac{\Pi}{1 - PP'}\right) \cdot P\overline{P'}}{1 - \Pi} \tag{10}$$

where

$$P = P(O^p) \text{ , } \overline{P'} = P(\neg O^{p'})$$

$$\Pi = \prod_{n \in \mathcal{L}} (1 - P(O^n) \cdot P(O^{n'}))$$

Figure 3 shows how the occupancy of the voxels in a map would evolve as multiple lines-of-action of a sequence of contacts intersect at a point. In this simplified two-dimensional case, every line intersects at one voxel in the environment map, which is likely occupied by an obstacle. As for the reconstruction of the object geometry in the bottom row of Figure 3, several locations are likely to be occupied, particularly the ones which have overlapped with the obstacle in the environment during contact. Of course, realistic scenarios are

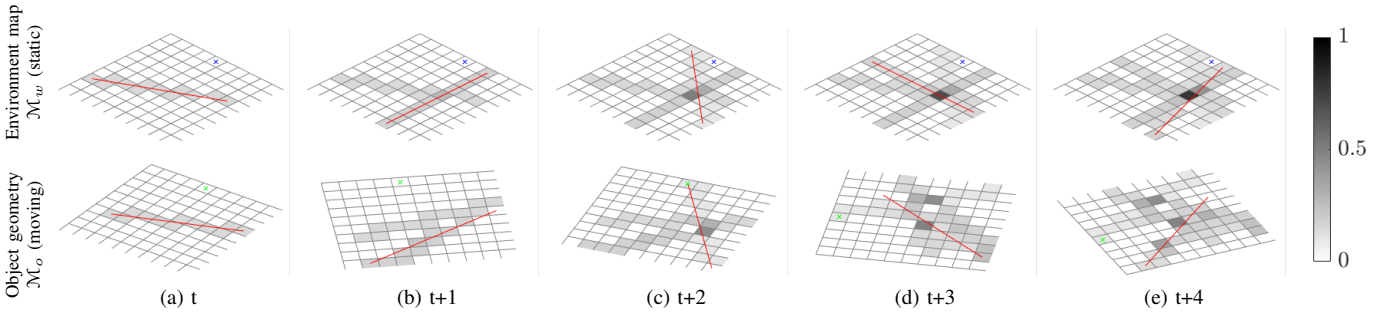


Fig. 3. Voxel occupancy (2D example). The top row shows the map of the environment and the bottom row depicts the object geometry. As the object moves and multiple lines-of-action of forces intersect at a location, the occupancy probability increases according to the probability law in (10).

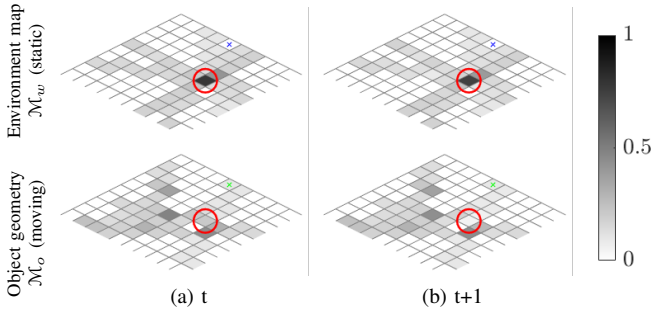


Fig. 4. Probability law in free space. The top row shows the environment map and the bottom row shows the object geometry. Left: before; right: after applying (11).

not as clear-cut, as it is unlikely that the robot collided at the same location either on the environment or the object several times in a row.

2) Probability Law for free space: At every instance where there is no measured force, we can use this information to reduce the probability of occupancy of overlapping pairs of voxels p and p' in maps $\mathcal{M}$ and $\mathcal{M}'$, respectively. Since there is no contact, we know that the admissible cases are simply $P(\neg C^p)$ for every pair of voxels. Also, the voxel at point p is only occupied if point p' is free. Thus the new probability assigned to the occupancy of p is:

$$\begin{aligned} P(O^p)_{t+1} &= \frac{P(O^p \cap \neg O^{p'})}{P(\neg C^p)} \\ &= \frac{P(O^p) \cdot (1 - P(O^{p'}))}{1 - P(O^p) \cdot P(O^{p'})} \end{aligned} \tag{11}$$

Finally, the converse operation is applied to p' to obtain $P(O^{p'})_{t+1}$. Figure 4 shows the result of applying the probability law for free space. The voxels circled in red show a location that is likely occupied in the environment $P(O^p) = .79$, whereas the corresponding voxel in the object geometry has $P(O^{p'}) = .21$. After applying the probability law in (11), $P(O^p)_{t+1} = .75$ and $P(O^{p'})_{t+1} = .05$. Simply put, if we are confident that there is an obstacle at a location in the environment and, at the robot's current pose, there is no collision detected, we become more confident that the coincident point on the object map is free.

III. Results and Discussion

A. Experimental Setup

We demonstrate two of several applications for the proposed method, namely, navigating through a cluttered environment to retrieve an object, and reconstructing the geometry of an object. The goal of object retrieval is to move an unknown object from a starting pose to a goal pose, based only on the robot state and measured external force. This application combines the proposed mapping approach with a geometric planner to find suitable paths that reach the end goal without passing through states where voxels with high occupancy probability overlap.

We experimentally validate our approach using both simulation and a real setting, in order to evaluate limitations associated with inherent uncertainties and noise. The real world trials were done using a commercially available Universal Robots UR5e, which is a 6 DOF manipulator with force-torque sensing capabilities at the end-effector. A Robotiq parallel jaw gripper was used to firmly grasp objects and the environment was set in an optical board to allow firm and accurate placement of the obstacles.

B. Implementation

The proposed algorithm was implemented in C++, using the ROS framework [20], and OpenCL [21] to increase performance via paralellization. The maps used the Octomap [22] data structure, which provides an efficient way to store 3D grid maps containing an occupancy probability, as well as utilities such as ray tracing, which is useful to determine the voxels intersected by the force line-of-action. We used the OMPL (Open Motion Planning Library) and FCL (Flexible Collision Library) [23] for planning collision-free paths, and PyBullet for simulation [24]. The source code is available at https://github.com/joaobimbo/fsmore.

C. Object Retrieval

The main application intended for this method is to retrieve objects with unknown shape from a cluttered scene. In these situations, visibility is usually reduced, and the shape of the object and the placement of obstacles in the environment are not available from vision. We tested this scenario in simulation to facilitate a systematic comparison between approaches. A

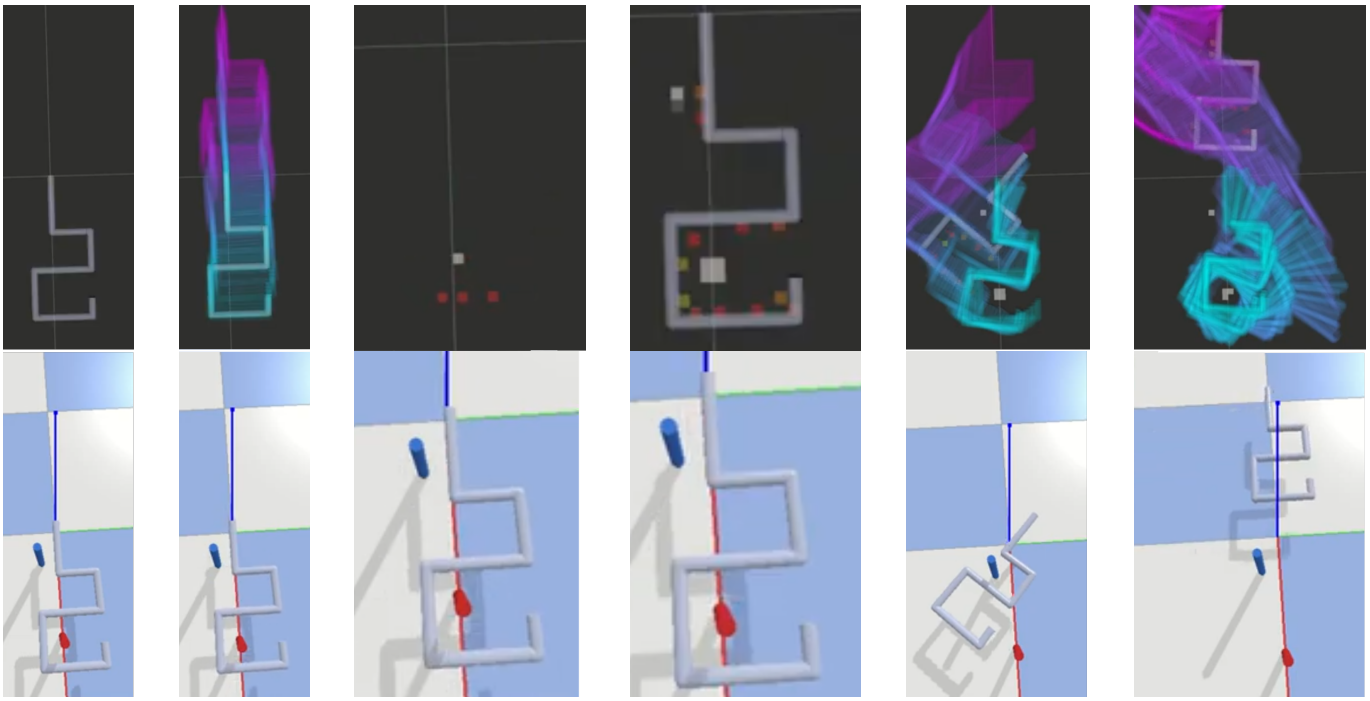


Fig. 5. Object untangling: proposed method to map the environment and reconstruct the grasped object. Top row: Object reconstruction in green/red, mapped obstacles in black/white, planned path in cyan/purple. Only the voxels with a probability of occupancy higher than 50% are displayed. Bottom row: Simulation environment.

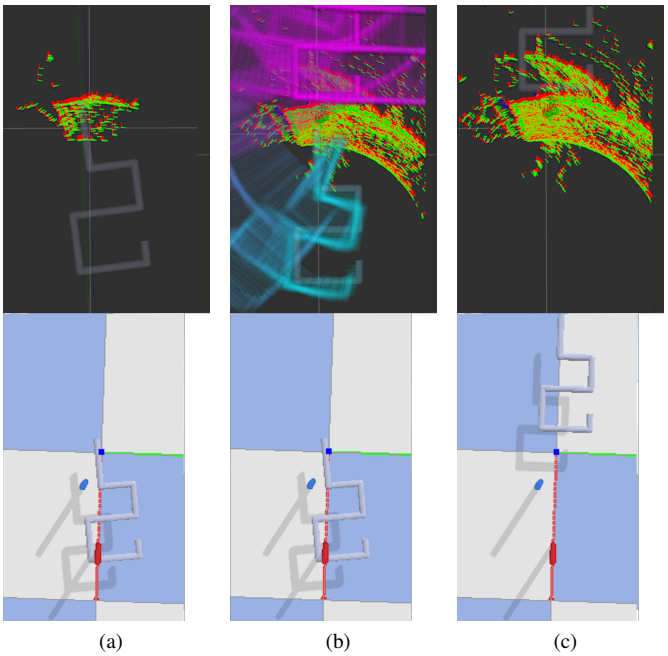


Fig. 6. Object untangling - C-space planner. Top row: Red-green-blue axes show invalid poses (poses where a collision between object and environment happened). Bottom row: Simulation environment.

virtual robot was instantiated, consisting of 6 joints (3 prismatic and 3 rotational), so that the end-effector Jacobian was a 6×6 identity matrix, allowing movement in every direction. This ensured that issues related with the robot workspace did not affect the comparison. While all the calculations, mapping, and collision detection are done in three dimensions, we constrained the movement of the robot in this case to a two-dimensional plane. This allows easier visualization of the maps and the planned path, as well as reducing the time to solve the problem. The goal of this task was to move the object from an initial to an end pose. Every time there was a collision above a certain threshold force (5 N), the robot/object was reset to the initial position. Our approach was compared with two other methods to plan and move the object away from the obstacles and reach a desired goal pose. The first method is based on a C-space planner [25], which kept track of the configuration of the virtual robot whenever there was a collision and treated that configuration as invalid, i.e. in collision. The planner would then connect the initial pose to the goal pose while avoiding any invalid states. The second approach we compared to was based on Deep Reinforcement Learning, particularly the Soft Actor-Critic (SAC) architecture [26]. To evaluate this method, we develop a representative Open AI Gym environment and utilize hyperparameters presented in the original work [27].

An example of our approach (SMORE) is shown in Figure 5: An s-shaped object is placed such that it is tangled in an environment containing two pegs (lower row, in blue and red). The goal is to move the object -1m in the x-direction. A geometric planner (RRT) finds a collision-free path between the current pose and the goal. We consider a voxel to be occupied if its occupancy probability $P(O^p) \geq 0.5$. Initially, since no obstacles are known, a straight path is planned in Fig. 5b. After a few failed attempts at moving in an approximately straight line, one of the pegs and a part of the object can

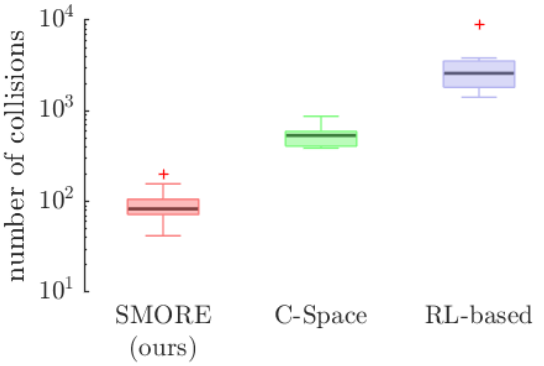
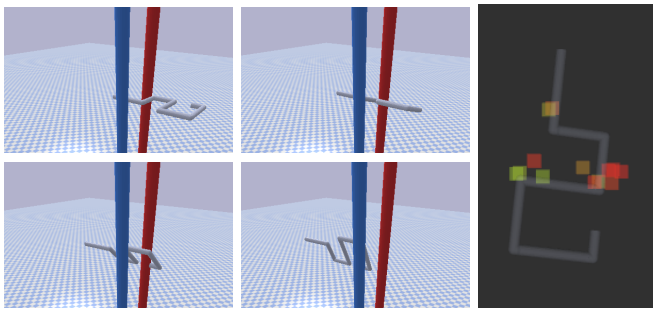


Fig. 7. Comparison of number of collision needed to untangle an object from a cluttered scene between methods (log scale) N=10



(a) (b)

Fig. 8. Experiment in simulation using RRT-Connect. The object is retrieved through a narrow space. Left: Object passing through slit Right: Resulting object reconstruction.

be seen in Fig. 5c. In Fig 5d most of the "cage" part of the object has been reconstructed, as well as both obstacles. In Fig. 5e, the planner attempts a plan around these obstacles, but is unsuccessful since there are still parts of the object that have not yet been reconstructed. Finally, in Fig. 5f, a plan is found that is able to retrieve the object completely, away from the two obstacles.

For comparison, the evolution of the C-space planner map construction is shown in Fig. 6. For each collision, the position and orientation of the object is saved. Fig. 6a shows the location of the first few collisions. Fig. 6b shows a plan around the known invalid configurations, whereas Fig. 6c shows the successful completion of the task, where the object has been removed from the obstacles and reached the goal pose. We also implemented a Deep Reinforcement Learning (RL) method based on SAC, which is generally regarded as state-of-the-art in model-free RL and has seen several extensions in the literature, e.g. [28], [29]. Similar to other evaluations, once an interaction occurs with sufficiently large application forces (>5N), the environment was reset. We define a simple reward function according to the Euclidean distance between the object pose and the goal pose, with a small penalty when contact interaction occurs. The system runs until the goal configuration is achieved, and we log the number of interactions required. The resolution of the octomap was set to 3 cm, and the planner step size was set to the same value. For the C-space planner, a pose was deemed invalid if the sum of the linear and angular distance to an invalid pose was under 0.03 (where linear distance was measured in meters and angular distance in radians). For SAC, the observation space was comprised of the difference between object pose and goal pose. Actions were continuous with a maximum magnitude of 0.03m / 0.3 radians.

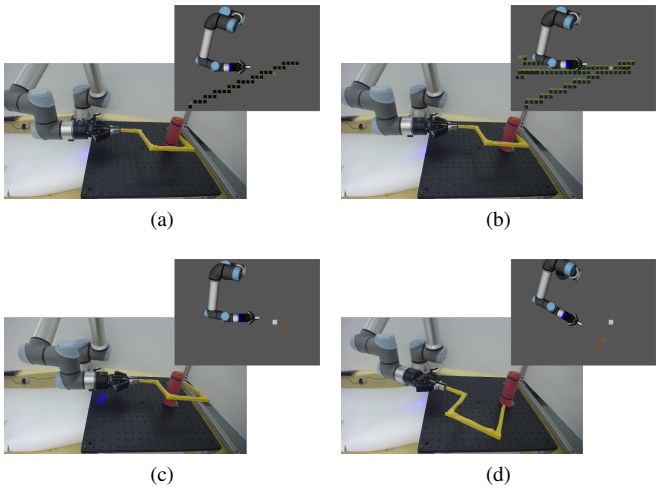
Figure 7 shows the performance comparison when extracting the object in the scenario presented in Figures 5 and 6. We compare the average number of collision that happened before the object was extracted once. These were 96 for our proposed method, 548 for the C-space planner, and 3136 for the Reinforcement Learning-based approach. It should be noted that, in practice, this task would not require this large amount of collisions if, on contact, the robot was allowed to move away from the obstacle and re-plan from that pose. We choose to reset the object to its initial pose to have a more unbiased comparison between approaches. It is also important to keep in mind that a trial was deemed successful if the object reaches the end pose once. It does not follow, necessarily, that subsequent trials will also arrive at the goal without collisions, mainly due to the stochasticity of both the RL approach and RRTs, but also to the fact that a valid plan may be found even if the environment and the object are not fully mapped yet.

Further validation was carried out using the same object as in Fig. 5 on a different environment. In this experiment, shown in Fig. 8, the robot was allowed to move and rotate in all three dimensions. Furthermore, a different, more sophisticated planner was tested (RRT-Connect [30]) together with our mapping approach. It can be seen that the object was successfully retrieved after a number of collisions required to reconstruct the object and the environment.

A similar but simpler scenario was tested on the real setup shown in Fig. 9. Given the lower accuracy of the force-torque sensor relative to the simulation, the resolution of the map was decreased to 0.05m and a single peg was used. The initial lines of action are shown in Figures 9a and 9b, where the location of the peg is already apparent. In Fig. 9c the object reconstruction is now sufficient for the planner to find the solution around the peg shown in Fig. 9d.

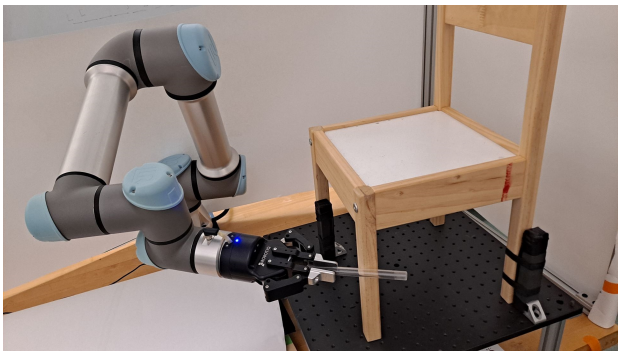
D. Object Reconstruction

The proposed method can also be applied for the purpose of reconstructing the geometry of a grasped object or of an object in the environment. We explore this application using a simple rod and manually poking an object at different points to perceive its shape. The robot was put in gravity compensation mode and a user moved it around both for safety and to ensure that different parts of the objects were touched. Figure 10a shows the setup, which consisted of a small chair and a grasped rod, and Fig. 10b shows the reconstruction result as the voxels with a probability of occupancy $P(O) \geq 0.5$, with red voxels having a higher occupancy probability. The distance between each occupied voxel and its closest point in the chair was computed using the chair model shown in Fig. 10b. The voxel resolution in this case was 0.04m, and the majority of the

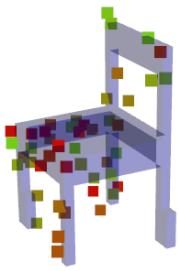


(a) (b) (c) (d)

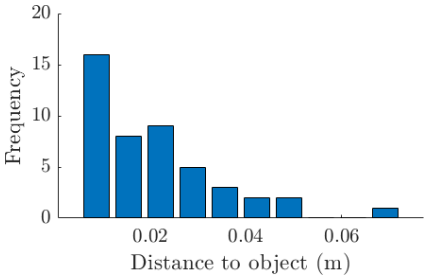
Fig. 9. Experiment on a real scenario with one peg. Environment map in black/white and object reconstruction in green/red (white and red are higher occupancy probability)



(a) Experimental setup



(b) Object Geometry



(c) Reconstruction accuracy

Fig. 10. Reconstruction of the geometry of a chair (voxel resolution = 0.04m). The purple chair was overlaid for verification

reconstructed voxels is within that distance, as shown Fig. 10c.

E. Discussion

The proposed algorithm was able to successfully reconstruct a grasped object and the surrounding environment at the same time. This enabled more effective navigation on this environment, requiring less collisions to retrieve an object with a complex shape from a situation where it was tangled with the environment. Compared to a traditional planning approach or more sophisticated machine learning methods, the proposed algorithm performed five and thirty times better, respectively. Reconstruction of an object of unknown geometry was also demonstrated to be a possible application of this method. This was done simply by poking it with an unknown object at different points.

The main shortcoming of this approach arises when the object becomes "jammed" between multiple obstacles. Since we are measuring the total force acting on the object, if there are multiple contact locations the force line-of-action is not guaranteed to be correct. A mitigating approach to solve this problem was to move the robot in small steps to ensure that only one contact is occurring. In realistic scenarios of high clutter, it cannot be guaranteed that there is a single contact location, and further investigation efforts should be focused into at least identifying such cases.

An important aspect that was not investigated in depth in this work was the noticeable sensitivity of the method's performance to a number of parameters, namely the chosen map resolution, the force threshold to consider a contact, and the force measurement sampling frequency. Another solution that was not investigated in this work was the inclusion of prior information on the map, obtained, for example, from a 3D vision system. If the geometry of one of the maps is known beforehand, either partially or completely, that information can be used to speed up the reconstruction of the other, without any modification to the method by just adding the known points to the respective octomap with $\sim 99.9\%$ occupancy probability.

IV. Conclusions

In this work we presented a method to map an environment and reconstruct the shape of a grasped object using only a robot's end-effector pose and the force measurements of the collisions between the object and the environment. On collision, the method finds the possible contact location by computing the line-of-action of the interaction force. It then computes the probability of occupancy along that line on two 3D maps – one that is static and maps the environment, and one that moves together with the robot end-effector. On the other hand, when the robot moves without colliding, the maps are also updated, since we know that there is no overlap of occupied voxels between the two maps. We provide two example applications for the method: navigating an object through a cluttered scene to remove it, and reconstruction of an object shape. We show that our approach outperforms traditional planning and a strategy based on reinforcement learning and validate our approach in a real setup.

In the future, a number of possible extensions can be thought of to improve the performance of the algorithm. In our current approach, the order at which the measurements arrive influence the computed probabilities. The solution to this could be to recursively update every line that intersects every pixel that is updated. In practice, this step would require a significant amount of computation time, since the number of lines on a typical 3D map with medium or high resolution easily reaches the thousands. Also, care should be taken to avoid falling into infinite update loops. For the object retrieval application, another possible improvement could be to use an optimal planner that takes into account the occupancy probability and

optimizes for paths along states where only voxels with low occupancy probability intersect. Current implementation only distinguishes valid and invalid states – a state is invalid if two voxels with $P(O^p) \geq 0.5$ intersect. Another improvement could follow an approach similar to the one proposed in [31] to be able to deal with multiple, sequentially occurring contacts. Finally, the contact forces that occur while retrieving an object might displace it in the robot hand, requiring that object pose uncertainty is addressed. This would probably require object pose to be estimated in parallel with the proposed algorithm, following existing strategies [32], [33], or a modification of the proposed method that accounts for this uncertainty. Finally, we intend on integrating a vision system that can provide at least partial information of the object and environment's geometry that can be complemented by the proposed approach.

ACKNOWLEDGMENT

The authors would like to thank Prof. Brian Scassellati and Nicole Salomons for their help in carrying out the experiments.

REFERENCES

[1] A. Bicchi, J. K. Salisbury, and D. L. Brock, "Contact sensing from force measurements," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 249–262, 1993.

[2] H. Liu, K. C. Nguyen, V. Perdereau, J. Bimbo, J. Back, M. Godden, L. D. Seneviratne, and K. Althoefer, "Finger contact sensing and the application in dexterous hand manipulation," *Autonomous Robots*, vol. 39, no. 1, pp. 25–41, 2015.

[3] N. Kurita, S. Sakaino, and T. Tsuji, "Whole-body force sensation by force sensor with end-effector of arbitrary shape," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5428–5433.

[4] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 1623–1630.

[5] J. Bimbo, C. Pacchierotti, N. G. Tsagarakis, and D. Prattichizzo, "Collision detection and isolation on a robot using joint torque sensing," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7604–7609.

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] A. S. Morgan, R. L. Baines, H. McClintock, and B. Scassellati, "Unstructured terrain navigation and topographic mapping with a low-cost mobile cuboid robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3597–3602.

[8] D. Droeschel and S. Behnke, "Efficient continuous-time slam for 3d lidar-based online mapping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5000–5007.

[9] B. Saund, S. Choudhury, S. Srinivasa, and D. Berenson, "The blindfolded robot: A bayesian approach to planning with contact feedback," in *International Symposium on Robotics Research (ISRR)*, 2019.

[10] A. Albini, F. Grella, P. Maiolino, and G. Cannata, "Exploiting distributed tactile sensors to drive a robot arm through obstacles," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4361–4368, 2021.

[11] 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.

[12] A. Gruebele, M. A. Lin, D. Brouwer, S. Yuan, A. Zerbe, and M. Cutkosky, "A stretchable tactile sleeve for reaching into cluttered spaces," *IEEE Robotics and Automation Letters*, 2021.

[13] G. S. Martins, R. P. Rocha, F. J. Pais, and P. Menezes, "Clusternav: Learning-based robust navigation operating in cluttered environments," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9624–9630.

[14] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-rl for insertion: Generalization to objects of unknown geometry," *arXiv preprint arXiv:2104.01167*, 2021.

[15] Z. Yi, R. Calandra, F. Veiga, H. van Hoof, T. Hermans, Y. Zhang, and J. Peters, "Active tactile object exploration with gaussian processes," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4925–4930.

[16] M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3811–3817.

[17] E. J. Smith, D. Meger, L. Pineda, R. Calandra, J. Malik, A. Romero, and M. Drozdzal, "Active 3d shape reconstruction from vision and touch," *arXiv preprint arXiv:2107.09584*, 2021.

[18] D. R. Faria, J. A. Prado, P. Drews Jr, and J. Dias, "Object shape retrieval through grasp exploration." in *ECMR*, 2009, pp. 43–48.

[19] F. Von Drigalski, S. Taniguchi, R. Lee, T. Matsubara, M. Hamaya, K. Tanaka, and Y. Ijiri, "Contact-based in-hand pose estimation using bayesian state estimation and particle filtering," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7294–7299.

[20] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA) WS on Open Source Robotics*. IEEE, May 2009, https://www.ros.org.

[21] J. E. Stone, D. Gohara, and G. Shi, "Opencl: A parallel programming standard for heterogeneous computing systems," *Computing in Science Engineering*, vol. 12, no. 3, pp. 66–73, 2010.

[22] 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, http://octomap.github.io.

[23] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, https://ompl.kavrakilab.org.

[24] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation in robotics, games and machine learning," 2017.

[25] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.

[27] C. D'Eramo, D. Tateo, A. Bonarini, M. Restelli, and J. Peters, "Mushroomrl: Simplifying reinforcement learning research," *Journal of Machine Learning Research*, vol. 22, no. 131, pp. 1–5, 2021.

[28] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," in *Advances in Neural Information Processing Systems*, 2019.

[29] A. S. Morgan, D. Nandha, G. Chalvatzaki, C. D'Eramo, A. M. Dollar, and J. Peters, "Model predictive actor-critic: Accelerating robot skill acquisition with deep reinforcement learning," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021.

[30] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.

[31] 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.

[32] J. Bimbo, S. Luo, K. Althoefer, and H. Liu, "In-hand object pose estimation using covariance-based tactile to geometry matching," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 570–577, 2016.

[33] G. Vezzani, U. Pattacini, G. Battistelli, L. Chisci, and L. Natale, "Memory unscented particle filter for 6-dof tactile localization," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1139–1155, 2017.