UNIVERSITY OF LEEDS

This is a repository copy of *Non-Prehensile Manipulation in Clutter with Human-In-The-Loop*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/157973/

Version: Accepted Version

Proceedings Paper:
Papallas, R and Dogar, MR orcid.org/0000-0002-6896-5461 (2020) Non-Prehensile Manipulation in Clutter with Human-In-The-Loop. In: Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA 2020). IEEE International Conference on Robotics and Automation (ICRA 2020), 31 May - 31 Aug 2020, Paris, France. IEEE , pp. 6723-6729. ISBN 978-1-7281-7395-5

https://doi.org/10.1109/ICRA40945.2020.9196689

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Uploaded in accordance with the publisher's self-archiving policy.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Non-Prehensile Manipulation in Clutter with Human-In-The-Loop

Rafael Papallas and Mehmet R. Dogar

***Abstract*— We propose a human-operator guided planning approach to pushing-based manipulation in clutter. Most recent approaches to manipulation in clutter employs randomized planning. The problem, however, remains a challenging one where the planning times are still in the order of tens of seconds or minutes, and the success rates are low for difficult instances of the problem. We build on these control-based randomized planning approaches, but we investigate using them in conjunction with human-operator input. In our framework, the human operator supplies a high-level plan, in the form of an ordered sequence of objects and their approximate goal positions. We present experiments in simulation and on a real robotic setup, where we compare the success rate and planning times of our human-in-the-loop approach with fully autonomous sampling-based planners. We show that with a minimal amount of human input, the low-level planner can solve the problem faster and with higher success rates.**

I. Introduction

We propose a human-operator guided planning approach to pushing-based manipulation in clutter. We present example problems in Figs. 1 and 2. The target of the robot is to reach and grasp the green object. To do this, however, the robot first has to push other objects out of the way (Fig. 1b to Fig. 1e). This requires the robot to plan which objects to contact, where and how to push those objects so that it can reach the goal object. We present an approach to this problem where a human-in-the-loop provides a *high-level plan*, which is used by a low-level planner to solve the problem.

These *reaching through clutter* problems are difficult to solve due to several reasons: First, the number of objects make the state space of high-dimensionality because the planner needs to reason about the robot state and all the movable objects. Second, this is an underactuated problem, since the objects cannot be controlled by the robot directly. Third, predicting the evolution of the system state requires running computationally expensive physics simulators, to predict how objects would move as a result of the robot pushing. Effective algorithms have been developed [1]–[12], however, the problem remains a challenging one, where the planning times are still in the order of tens of seconds or minutes, and the success rates are low for difficult problems.

Further study of the reaching through clutter problem is important to develop approaches to solve the problem more successfully and faster. It is a problem that has a potential for major near-term impact in warehouse robotics (where robots need to reach into shelves to retrieve objects) and personal home robots (where a robot might need to reach into a fridge or shelf to retrieve an object). The Amazon Picking Challenge [13] was a competition which gained particular attention for this potential near-term impact of robotic manipulation to warehouse robotics. The algorithms that we currently have, however, are not able to solve reaching through clutter problems in the real world in a fast and consistent way. Here, we ask the question of whether human-operators can be used to provide a minimal amount of input that results in a significantly higher success rate and faster planning times.

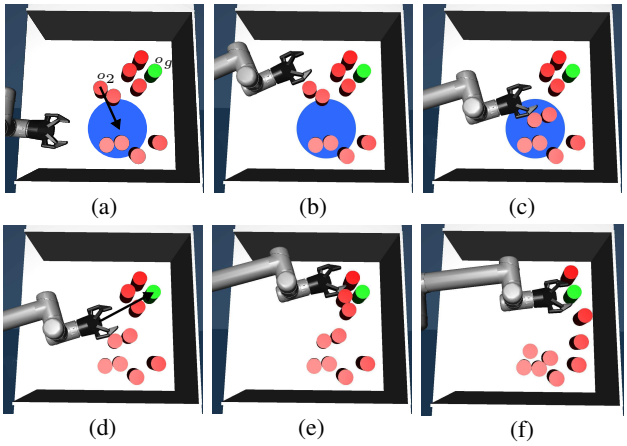


Fig. 1: A human-operator guiding a robot to reach for the green goal object, o_g. Arrows indicate human interaction with the robot. In (a) the operator indicates o_2 to be pushed to the blue target region. From (a) to (c) the robot plans to perform this push. In (d) the operator indicates to the robot to reach for the goal object. From (d) to (f) the robot plans to reach the goal object.

Most recent approaches to the reaching through clutter problem employs the power of randomized *kinodynamic planning*. Haustein et al. [4] use a kinodynamic RRT [14], [15] planner to sample and generate a sequence of robot pushes on objects to reach a goal state. Muhayyuddin et al. [5] use the KPIECE algorithm [16] to solve this problem.

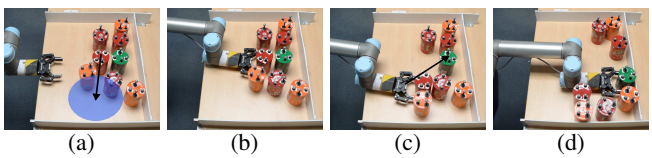


Fig. 2: Human-operator guiding a robot in the real-world.

This research has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grants agreement No. 746143, and from the UK Engineering and Physical Sciences Research Council under grant EP/N509681/1, EP/P019560/1 and EP/R031193/1.

Authors are with the School of Computing, University of Leeds, United Kingdom {r.papallas, m.r.dogar}@leeds.ac.uk

These planners report some of the best performance (in terms of planning times and success rates) in this domain so far.

We build on these kinodynamic planning approaches, but we investigate using them in conjunction with human-operator input. In our framework, the human operator supplies a *high-level plan* to make the underlying planners solve the problem faster and with higher success rates.

For example in Fig. 1a, the human operator supplies the high-level action of first pushing the object o_2 to the blue region. A key point here is that pushing o_2 in Fig. 1a to its target region is itself a problem which requires kinodynamic planning through clutter, since the object and the robot may need to contact and push other objects during the motion. The robot uses a kinodynamic planner to approach o_2 in Fig. 1b and to push it to the target region in Fig. 1c. In Fig. 1d the operator points directly the actual goal object (green). The kinodynamic planner in Fig. 1e finds a way to push other objects out of the way and in Fig. 1f successfully reaches the goal object. The human operator's role in the system is not to guide the robot all the way to the goal, but to provide key high-level actions to help the robot. At any point during the interaction, even at the very beginning, the operator can decide not to provide any further high-level actions (either because the scene is easy enough for the low-level planner or because the operator is busy) and she can command the system to plan directly for the actual goal object. The system degrades nicely to state-of-the-art kinodynamic planning if no high-level actions are provided.

The use of high-level plans is related to recent work in robotic hierarchical planning [2], [17]–[20] and task-and-motion planning (TAMP) [21]–[23]. This line of work shows that with a good high-level plan for a task, the search of the low-level motion planner can be guided to a relevant but restricted part of the search space, making the planner faster and more successful. Particularly relevant to our problem is the work from Stilman et al. [17], which formulates the problem of *manipulation/navigation among movable obstacles (NAMO)* as a high-level search over the orderings of objects to be moved, combined with a low-level motion planner that pick objects up and move in that order. We use a similar high-level plan structure, i.e. an ordering of objects, but we focus on non-prehensile manipulation of objects, rather than pick-and-place.

The hierarchical/TAMP planners above generate high-level plans autonomously. Motivated by existing work in human-in-the-loop planning [24]–[29], in this work we investigate the potential of using a human operator to suggest high-level plans. The existing work in human-in-the-loop planning focuses on path planning and providing clues to a planner to guide it through the collision-free space. We explore a similar approach, but in the context of *non-prehensile pushing-based planning*, where human physical intuition can be useful. Other human-in-the-loop systems have been investigated for pick-and-place tasks [30]–[33] but to the best of our knowledge, a human-in-the-loop approach has not been applied to non-prehensile physics-based manipulation before.

We compare our method to using kinodynamic methods without any high-level plans, e.g. KPIECE and RRT. We also compare our method to hierarchical methods which generate high-level plans autonomously. For the latter, we implemented a non-prehensile variation of the NAMO planner as well as an approach which uses a straight-line motion heuristic to generate candidate objects for the high-level plan. We performed experiments in simulation and on a real robot, which shows that the human-in-the-loop approach produces more successful plans and faster planning times. This gain, of course, comes at the expense of a human operator's time. We show that this time is minimal and to evaluate this further, we experiment with a single human operator providing high-level plans in-parallel to multiple robots and present an analysis. We discuss whether such an approach may be feasible in a warehouse automation setting. To support reproducibility, we provide the source code of all our algorithms and experiments in an open repository[1].

II. Problem Formulation

Our environment is comprised of a robot r, a set of movable obstacles O, and other static obstacles. The robot is allowed to interact with the movable obstacles, but not with the static ones. We also have $o_g \in O$ which is the *goal* object to reach.

We are interested in problems where the robot needs to reach for an object in a cluttered shelf that is constrained from the top, and therefore we constrain the robot motion to the plane and its configuration space, Q^r, to $SE(2)$. The configuration of a movable object $i \in \{1, \ldots, |O|\}$, q^i, is its pose on the plane (x, y, θ). We denote its configuration space as Q^i. The configuration space of the complete system is the Cartesian product $Q = Q^r \times Q^g \times Q^1 \times \cdots \times Q^{|O|-1}$.

Let $q_0 \in Q$ be the initial configuration of the system, and $Q_{goals} \subset Q$ a set of possible goal configurations. A goal configuration, $q_n \in Q_{goals}$, is defined as a configuration where o_g is within the robot's end-effector (see Fig. 1f).

Let U be the control space comprised of the robot velocities. Let the system dynamics be defined as $f : Q \times U \to Q$ that propagates the system from $q_t \in Q$ with a control $u_t \in U$.

We define the *Reaching Through Clutter* (RTC) problem as the tuple $(Q, U, q_0, Q_{goals}, f)$. The solution to the problem is a sequence of controls from U that move the robot from q_0 to a $q_n \in Q_{goals}$.

III. Sampling-based Kinodynamic Planners

Two well known sampling-based kinodynamic planners are Rapidly-exploring Random Trees (RRT) [14], [15] and Kinodynamic Motion Planning by Interior-Exterior Cell Exploration (KPIECE) [16]. Both RRT and KPIECE have been used before in the literature to solve problems similar to the RTC problem [4], [5], [34], [35]. We use kinodynamic RRT and KPIECE in our work in two different ways: (1) as baseline RTC planners to compare against, and (2) as the low-level planners for the Guided-RTC Framework that accepts high-level actions (explained in Sec. IV).

Kinodynamic RRT: RRT is a sampling-based planner that builds a tree from the initial state, $q_0 \in Q$, and then samples

[1] https://github.com/rpapallas/hitl_clutter

a random state $q_{rand} \in Q$ and tries to extend the tree from the nearest neighbor $q_{near} \in Q$. Kinodynamic RRT samples controls to bring q_{near} near to q_{rand}.

KPIECE: KPIECE is a sampling-based planner that operates well in problems with complex dynamics [16]. KPIECE builds a tree from the state and control space until a goal is reached. KPIECE uses the notion of space coverage to guide its exploration in the state space by constructing a discretization of the state space. KPIECE samples a control from U and it tries to expand the tree and update the discretization.

In this work, when we plan with a kinodynamic planner (either RRT or KPIECE) we will use the notation *kinodynamicPlanning(q_{start}, goal)* with a start configuration of the system, q_{start}, and some *goal* input.

IV. Guided-RTC Framework

In this section, we describe a *guided* system to solve RTC problems. A Guided-RTC system accepts high-level actions. A high-level action can suggest to push a particular obstacle object into a certain region, or it may suggest to reach for the goal object. We formally define a high-level action with the triple (o_i, x_i, y_i), where $o_i \in O$ is an object, and (x_i, y_i) is the centroid of a *target region* that o_i needs to be pushed into. The target region has a constant diameter d. When $o_i = o_g$, this is interpreted as the high-level action to reach for the goal object, and the centroid can be ignored. The high-level actions may be suggested by an automated high-level planner or by a human-operator.

Consider Fig. 1 as an example. A human-operator suggests a high-level action $(o_2, 1.15, 0.4)$ (Fig. 1a), where $(1.15, 0.4)$ is the centroid of the blue target region. The Guided-RTC system finds the controls to push o_2 into the target region (Fig. 1c). When the human-operator suggests to reach for the goal object o_g (Fig. 1d), the system finds the controls to perform this action (Figs. 1e and 1f).

In this work, we investigate how a Guided-RTC system with a human-in-the-loop performs when compared with (a) solving the original RTC problem directly using kinodynamic approaches (Sec. III), and (b) using Guided-RTC systems with automated ways of generating the high-level actions.

In Sec. IV-A we present a generic algorithm to implement the Guided-RTC framework which is agnostic to how the high-level actions are generated. Then we present different approaches to generate the high-level actions, including a human-in-the-loop approach in Sec. IV-B, as well as two other automated approaches in Secs IV-C and IV-D.

A. A Generic approach for Guided-RTC Planning

We present a generic algorithm for Guided-RTC in Alg. 1. The initial configuration of the problem is assumed to be the current configuration, $q_{current}$, of the system (line 2). The next high-level action is decided based on the current configuration (line 4). If the object in the high-level action is not the goal object (line 5), then it is pushed to the target region between lines 6 and 11, and a new high-level action is requested. If it is the goal object, the robot tries to reach it between lines 13 and 15 and the system terminates.

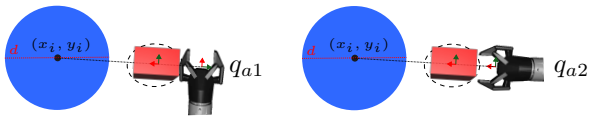


Fig. 3: Approaching states: The blue circle is the target region, the red rectangle the object to manipulate. We compute two approaching states, q_{a1} and q_{a2}.

We plan to push an object to its target region in two steps. In line 7 we plan to an intermediate *approaching state* near the object, and then in line 9, we plan from this approaching state to push the object to its target region. Specifically, given an object to push, o_i, we compute two approaching states q_{a1} and q_{a2} (line 6). Fig. 3 shows how these approaching states are computed, based on the object's current position, the centroid (x_i, y_i) and the minimum enclosing circle of the object. The approaching state q_{a1} encourages side-ways pushing, where q_{a2} encourages forward pushing. We also experimented with planning without first approaching the object but we found that approaching the object from a good pose yields to faster pushing solutions. Using both approaching states as the goal we plan to move to one of them (multi-goal planning) in line 7. Then, from the approaching state reached (either q_{a1} or q_{a2}) we push o_i to its target region (line 9). If any of the two planning calls in lines 7 and 9 fails, then the algorithm proceeds to the next high-level action (line 4). Otherwise, we execute the solutions sequentially in line 11, which changes the current system configuration $q_{current}$.

We use kinodynamic planners (e.g. kinodynamic RRT or KPIECE) to support the planning in lines 7, 9 and 13.

Alg. 1 runs up to an overall time limit, $T_{overall}$, or until a goal is reached. The pushing planning calls in lines 7 and 9 have their own shorter time limit, $T_{pushing}$, and they should find a valid solution within this limit. The planning call in line 13 is allowed to run until the overall time limit is over.

Algorithm 1 Guided-RTC

```
1:  procedure GRTC(Q, U, q_0, Q_goals)
2:    q_current ← q_0
3:    do
4:      o_i, x_i, y_i ← NextHighLevelAction(q_current)
5:      if o_i ≠ o_g then
6:        q_a1, q_a2 ← compute approaching states to o_i
7:        kinodynamicPlanning(q_current, {q_a1, q_a2})
8:        if planning fails then continue
9:        kinodynamicPlanning(q_a1 or q_a2, (o_i, x_i, y_i))
10:       if planning fails then continue
11:       q_current ← execute solutions from lines 7 and 9
12:   while o_i ≠ o_g
13:   kinodynamicPlanning(q_current, Q_goals)
14:   if planning succeeds then
15:     q_current ← execute solution from line 13
```

B. Guided-RTC with Human-In-The-Loop (GRTC-HITL)

Guided-RTC with Human-In-The-Loop (GRTC-HITL) is an instantiation of the GRTC Framework. A human-operator, through a graphical user interface, provides the high-level actions. In Alg. 2 we present GRTC-HITL NEXTHIGHLEVELACTION function (referenced in Alg. 1, line 4).

The human provides high-level actions until she selects the goal object, o_g. The GRTC framework (Alg. 1) plans and executes them. The state of the system changes after each high-level action and the human operator is presented with the resulting state each time ($q_{current}$). Note here that *the operator can decide not to provide any guidance* (by selecting the goal object straightaway), which would be equivalent to running a state-of-the-art kinodynamic planning on the original RTC problem.

We developed a simple user interface to communicate with the human-operator. The operator at every step is presented with a window showing the environment and the robot. The operator, using a mouse pointer, provides the input by first clicking on the desired object and then a point on the plane (Fig. 1a) that becomes the centroid of the target region.

The approach we propose here uses a human-operator to decide on the high-level plan. One question is whether one can use automatic approaches, and how they would perform compared to the human suggested actions. To make such a comparison, we implemented two automated approaches.

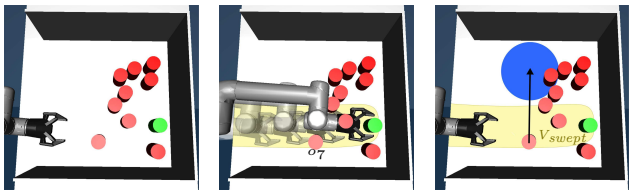
C. Guided-RTC with NAMO

We adapted the NAMO algorithm described by Stilman et al. [17] to our problem as an alternative, autonomous, way to generate a high-level plan. NAMO has originally been used for pick-and-place manipulation. We adapted it to work for non-prehensile tasks by using a kinodynamic planner as the low-level planner, instead of collision-free motion planners as in the original work.

To determine the ordering of objects to manipulate and where to place them, i.e. the high-level plan, NAMO uses backward planning. It starts by running the low-level planner to reach the goal, assuming the robot can travel through other movable objects. The resulting volume of space swept by the robot to reach the goal object is then checked to see which movable objects intersect with it. These objects are added to a queue to be moved out of this swept volume. The algorithm then pops out an object from this queue and makes a recursive call to reach and move that object. This process continues until the queue is empty, meaning that (1) there is a plan to reach and move every object out of the way, and (2) there is a position to place every object out of the accumulated robot swept volume. The last object planned for is the first one to be moved during execution. For a more detailed explanation of NAMO, we refer the reader to Stilman et al. [17].

Since NAMO plans backward, to decide on the first object to be moved, it needs to determine all the objects to be moved and their target positions. While this means NAMO can offer theoretical guarantees when a plan exists, it also means that in highly cluttered environments like ours, NAMO can quickly run out of space to place objects, before it resolves all the constraints. In our experimental setting which includes a high number of objects in a restricted shelf space, NAMO failed in all cases by filling up the space with the robot swept volume before a plan for all objects in the queue have been found.

This motivated us to design a heuristic approach similar to NAMO, but one that plans forward, by directly identifying the first object to move out of the way.



(a) (b) (c)

Fig. 4: GRTC-Heuristic: (a) Initial state. (b) The robot moves on a straight line to the goal object, o_g, to obtain the first blocking obstacle (o_7) and the swept volume (yellow area). (c) The heuristic produces a high-level action for o_7 indicated by the arrow and the target region (blue). This process is repeated until V_{swept} contains no blocking obstacle.

D. Guided-RTC with Straight Line Heuristic (GRTC-Heuristic)

We present this approach in Alg. 3 and illustrate it in Fig. 4. This heuristic assumes the robot moves on a straight line from its current position towards the goal object (Fig. 4b). The first blocking object, o_b in line 2, is identified as the next object to be moved. During the straight line motion, we capture the robot's swept volume, V_{swept} (Fig. 4b). We randomly sample a collision-free target region centroid outside V_{swept} (Alg. 3 line 4 and Fig. 4c). The object and the centroid are then returned as the next high-level action (Alg. 3 line 5). The centroid sampling happens 30cm around the object's initial position to maximize the chance of a successful pushing. If there is no collision-free space around o_b, then we sample from the entire space.

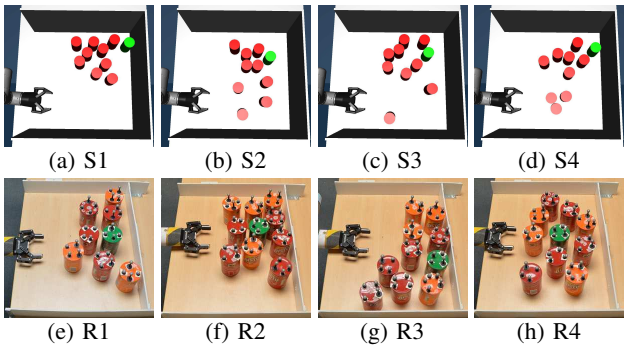
After every high-level action suggested by the heuristic, the

Algorithm 2 GRTC-HITL

1: **function** NEXTHIGHLEVELACTION($q_{current}$)
2: $o_i \leftarrow$ get object selection from human operator
3: **if** $o_i \neq o_g$ **then**
4: $x_i, y_i \leftarrow$ get region centroid from human operator
5: **return** o_i, x_i, y_i
6: **return** o_g

Algorithm 3 GRTC-Heuristic Planner

1: **function** NEXTHIGHLEVELACTION($q_{current}$)
2: $o_b \leftarrow$ find the first blocking obstacle to o_g
3: **if** there exists a blocking obstacle o_b **then**
4: $x_b, y_b \leftarrow$ find collision-free placement of o_b
5: **return** o_b, x_b, y_b
6: **return** o_g ▷ No blocking obstacle, reach the goal



(a) S1 (b) S2 (c) S3 (d) S4

(e) R1 (f) R2 (g) R3 (h) R4

Fig. 5: Initial states of different problems in simulation (S1-S4) and real world (R1-R4). Goal object is in green.

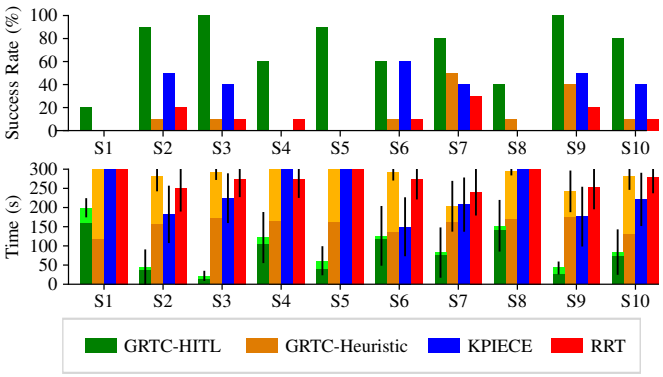


Fig. 6: Simulation results, for each scene (S1-S10): (Top) Success rate. (Bottom) Mean planning time. The error bars indicate the 95% CI. For GRTC-HITL and GRTC-Heuristic, the dark shade indicates the planning time where the light shade indicates the time it took to produce the high-level actions (for GRTC-HITL this is a fraction of the time).

Guided-RTC framework (Alg. 1) plans and executes it and the state of the system is updated ($q_{current}$). The heuristic then suggests a new high-level action from $q_{current}$ until there is no blocking obstacle (Alg. 3 line 6).

V. Experiments & Results

The algorithms we evaluate are: (1) GRTC-HITL which is the main algorithm we propose and uses a human operator to obtain the high-level actions (Sec. IV-B), (2) GRTC-Heuristic which uses a straight line heuristic to automatically determine the high-level actions (Sec. IV-D) and (3) Kinodynamic RRT and KPIECE (Sec. III) which plan to reach for the goal object without a high-level plan. As explained in Sec. IV-C, NAMO failed to find solutions in our problems and therefore we did not include results for it here.

For all experiments, we use the Open Motion Planning Library (OMPL) [36] implementation of RRT and KPIECE. We use MuJoCo[2] [37] to implement the system dynamics, f. For all planners, the overall planning time limit, $T_{overall}$, is 300 seconds, after which it was considered a failure. For GRTC-HITL and GRTC-Heuristic, $T_{pushing}$ is 10 seconds. For GRTC-HITL, the human-interaction time was included in the overall time limit. The same human-operator, who was experienced with the system, was used in all experiments. Since we are interested in an industrial/warehouse scenario where human-operators would be trained to use the system, we are mainly interested in the performance of trained operators, rather than novices.

In Sec. V-A we present simulation results comparing GRTC-HITL with Kinodynamic RRT, KPIECE, and GRTC-Heuristic. In Sec. V-B we show results where the human operator guides multiple robots in parallel, in simulation. In Sec. V-C we present real-world experiments comparing kinodynamic planners with GRTC-HITL on 10 different scenes. A video with some of these experiments is available on `https://youtu.be/nfr1Fdketrc`.

A. Simulation Results

We evaluated each approach 100 times by running them 10 times in 10 different, randomly-generated, scenes. We use a randomizer that places the goal object at the back of the shelf and then incrementally places the remaining (nine) objects in the shelf such that no object collides with each other. Some of the scenes are presented in Figs. 5a to 5d.

For GRTC-HITL, the human-operator interacted with each scene *once* and from the last state left by the human-operator we ran the planner (Alg. 1 line 13) to reach for the goal object *10 times*. For GRTC-Heuristic we ran all 100 experiments with both RRT and KPIECE as the low-level planners and we present the better performing one. For GRTC-HITL and GRTC-Heuristic the low-level planner is RRT.

Fig. 6 summarizes the results of our experiments for each of the random scenes (S1-S10). Fig. 6-Top shows that GRTC-HITL yields to more successes per scene than any other approach except for S6 which was as successful as KPIECE. The overall success rate for each approach is 72% for GRTC-HITL, 11% for RRT, 28% for KPIECE and 14% for GRTC-Heuristic. Fig. 6-Bottom shows that GRTC-HITL improved the planning time in *all* scenes.

Table I summarizes the guidance performance for GRTC-HITL and GRTC-Heuristic for all ten scenes. Proposed Actions indicates the total number of high-level actions proposed. This number includes the successful actions (actions that the planner managed to satisfy) and failed actions (actions that the planner could not find a solution for). Guidance Time indicates the time spent on generating the high-level actions in seconds (in case of GRTC-HITL the time the human-operator was interacting with the system and for GRTC-Heuristic the time took for the heuristic to generate the high-level actions). On average, the human proposed around 5 actions, of which around 3 were successful. On the other side, GRTC-Heuristic proposed on average around 88 actions, of which only 3 were successful. The human operator spent on average 14 seconds interacting with the system while GRTC-Heuristic spent on average 124 seconds proposing high-level actions.

[2]On a computer with Intel Core i7-4790 CPU @ 3.60GHz, 16GB RAM.

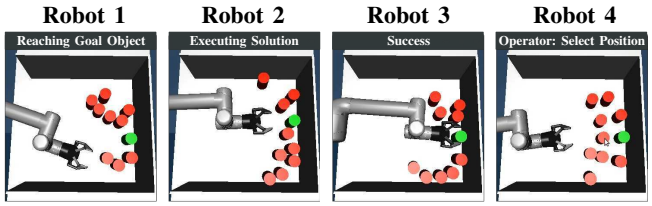


Fig. 7: Parallel Guidance: The first robot is planning to reach for the goal object, the second one executes a solution, the third robot successfully reached the goal object, the fourth robot is waiting for human input (operator's main focus).

B. Parallel Guidance

Since the human spends only a small amount of time guiding the robot using GRTC-HITL, a single human-operator can be used to guide multiple robots simultaneously, e.g. in a warehouse where a small number of operators remotely guide a large number of robots. We present an example in Fig. 7. We performed experiments to test how the performance was affected when the human-operator guides multiple robots in parallel. We tested guiding four robots in parallel. We generated 20 new random scenes and divided them into five groups of four. We compare the performance when the robots were guided individually in 20 different runs, to the performance when four robots were guided in parallel in five different runs. We repeated this process two times (i.e. 40 individual runs and 10 parallel runs). Note that in the worst case with a time limit of 300 seconds, running 20 scenes individually requires 100 minutes of human-operator time; whereas five parallel runs require just 25 minutes.

Table II summarizes these results. The success rate of parallel guidance is 60% and for individual guidance 70%. This efficient use of the human-operator's time comes with the cost of slightly increased planning time and lower success rate. We also measured the time the system was waiting for human input which was on average 15 seconds.

TABLE I: Simulation results.

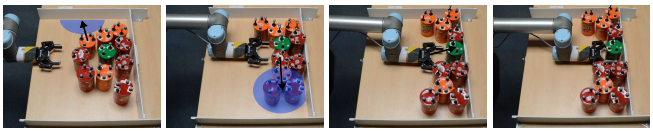
	GRTC-HITL		**GRTC-Heuristic**	
	μ	σ	μ	σ
Proposed Actions	4.9	3.3	88.4	58.2
Successful Actions	3.1	1.0	3.0	1.4
Guidance Time (s)	13.6	10.0	124.3	81.7

TABLE II: Parallel vs. Individual Guidance.

	Parallel		**Individual**	
	μ	σ	μ	σ
Guidance Time (s)	21.1	28.0	13.0	14.4
Overall Planning Time (s)	139.7	117.9	122.8	120.0
Planner Idle Time (s)	14.8	9.4	0.0	0.0

TABLE III: Real-world results.

	GRTC-HITL	**KPIECE**	**RRT**
Successes	7	1	2
Planning Failures	2	4	8
Execution Failures	1	5	0



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

Fig. 8: A guided planning demonstration in the real world.

C. Real-robot results

We performed experiments using a UR5 manipulator on a Ridgeback omnidirectional base. We used the OptiTrack motion capture system to detect initial object/robot poses and to update the state in the human interface after every high-level action.

We evaluated RRT, KPIECE and GRTC-HITL performance in *ten* different problems in the real world. We show some of the scenes in Figs. 5e to 5h.

Table III summarizes the success rate of each approach in the real world. When we say that the robot failed during execution, we mean that although the planner found a solution, when the robot executed the solution in the real-world, it either failed to reach the goal object, or it violated some constraint (hit the shelf or dropped an object to the floor). These execution failures were due to the uncertainty in the real world: The result of the robot's actions in the real-world yield to different states than the ones predicted by the planner.

The success rate for GRTC-HITL, RRT and KPIECE is 70%, 20%, and 10% respectively. GRTC-HITL failed 20% during planning and 10% during execution. KPIECE was more successful during planning than RRT but failed most of the times during execution. RRT, on the other, hand accounts for more failures during planning than any other approach.

In Figs. 2 and 8 we show two examples. In the first example, the human operator provides the first high-level action in Fig. 2a and then indicates the goal object in Fig. 2c which is reached in Fig. 2d. In the second example, the human-operator provides initially two high-level actions (Fig. 8a and Fig. 8b). The operator in Fig. 8c indicates the goal object and the robot reached the goal object in Fig. 8d.

VI. Conclusions

We introduced a new human-in-the-loop framework for physics-based non-prehensile manipulation in clutter (GRTC-HITL). We showed through simulation and real-world experiments that GRTC-HITL is more successful and faster in finding solutions than the three baselines we compared with. We also presented experiments where a single human-operator guides multiple robots in parallel, to make best use of the operator's time. We made the source code of our framework and of the baselines publicly available.

To the best of our knowledge, this is the first work to look into non-prehensile manipulation with human-in-the-loop. In the future, we would like to build on this work to evaluate the parallel control of higher numbers of robots, by minimizing the time the system is idle.

REFERENCES

[1] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, "Physics-based grasp planning through clutter," in *Robotics: Science and Systems*, 2012.

[2] G. Havur, G. Ozbilgin, E. Erdem, and V. Patoglu, "Geometric rearrangement of multiple movable objects on cluttered surfaces: A hybrid reasoning approach," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 445–452.

[3] N. Kitaev, I. Mordatch, S. Patil, and P. Abbeel, "Physics-based trajectory optimization for grasping in cluttered environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3102–3109.

[4] J. A. Haustein, J. King, S. S. Srinivasa, and T. Asfour, "Kinodynamic randomized rearrangement planning via dynamic transitions between statically stable states," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3075–3082.

[5] M. ud din, M. Moll, L. Kavraki, J. Rosell *et al.*, "Randomized physics-based motion planning for grasping in cluttered and uncertain environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 712–719, 2018.

[6] W. Bejjani, R. Papallas, M. Leonetti, and M. R. Dogar, "Planning with a receding horizon for manipulation in clutter using a learned value function," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[7] C. Nam, J. Lee, Y. Cho, J. Lee, D. H. Kim, and C. Kim, "Planning for target retrieval using a robotic manipulator in cluttered and occluded environments," *arXiv preprint arXiv:1907.03956*, 2019.

[8] J. E. King, M. Cognetti, and S. S. Srinivasa, "Rearrangement planning using object-centric and robot-centric action spaces," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3940–3947.

[9] W. Bejjani, M. R. Dogar, and M. Leonetti, "Learning physics-based manipulation in clutter: Combining image-based generalization and look-ahead planning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[10] W. C. Agboh and M. R. Dogar, "Real-time online re-planning for grasping under clutter and uncertainty," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–8.

[11] E. Huang, Z. Jia, and M. T. Mason, "Large-scale multi-object rearrangement," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 211–218.

[12] K. Kim, J. Lee, C. Kim, and C. Nam, "Retrieving objects from clutter using a mobile robotic manipulator," in *2019 16th International Conference on Ubiquitous Robots (UR)*. IEEE, 2019, pp. 44–48.

[13] C. Eppner, S. Höfer, R. Jonschkowski, R. M. Martin, A. Sieverling, V. Wall, and O. Brock, "Lessons from the amazon picking challenge: Four aspects of building robotic systems." in *Robotics: Science and Systems*, 2016.

[14] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[15] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[16] I. A. Şucan and L. E. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," in *Algorithmic Foundation of Robotics VIII*. Springer, 2009, pp. 449–464.

[17] M. Stilman, J.-U. Schamburek, J. Kuffner, and T. Asfour, "Manipulation planning among movable obstacles," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3327–3332.

[18] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical planning in the now," in *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[19] M. Dogar and S. Srinivasa, "A framework for push-grasping in clutter," *Robotics: Science and systems VII*, vol. 1, 2011.

[20] G. Lee, T. Lozano-Pérez, and L. P. Kaelbling, "Hierarchical planning for multi-contact non-prehensile manipulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 264–271.

[21] F. Lagriffoul, D. Dimitrov, J. Bidot, A. Saffiotti, and L. Karlsson, "Efficiently combining task and motion planning using geometric constraints," *The International Journal of Robotics Research*, vol. 33, no. 14, pp. 1726–1747, 2014.

[22] A. Wells, N. Dantam, A. Shrivastava, and L. Kavraki, "Learning feasibility for task and motion planning in tabletop environments," *IEEE Robotics and Automation Letters*, 2019.

[23] J. Lee, Y. Cho, C. Nam, J. Park, and C. Kim, "Efficient obstacle rearrangement for object manipulation tasks in cluttered environments," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 183–189.

[24] Y. K. Hwang, K. R. Cho, S. Lee, S. M. Park, and S. Kang, "Human computer cooperation in interactive motion planning," in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*. IEEE, 1997, pp. 571–576.

[25] O. B. Bayazit, G. Song, and N. M. Amato, "Enhancing randomized motion planners: Exploring with haptic hints," *Autonomous Robots*, vol. 10, no. 2, pp. 163–174, 2001.

[26] M. Taïx, D. Flavigné, and E. Ferré, "Human interaction with motion planning algorithm," *Journal of Intelligent & Robotic Systems*, vol. 67, no. 3-4, pp. 285–306, 2012.

[27] J. Denny, J. Colbert, H. Qin, and N. M. Amato, "On the theory of user-guided planning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4794–4801.

[28] F. Islam, O. Salzman, and M. Likhachev, "Online, interactive user guidance for high-dimensional, constrained motion planning," *arXiv preprint arXiv:1710.03873*, 2017.

[29] J. Denny, R. Sandström, and N. M. Amato, "A general region-based framework for collaborative planning," in *Robotics Research*. Springer, 2018, pp. 563–579.

[30] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 1–8.

[31] S. Muszynski, J. Stückler, and S. Behnke, "Adjustable autonomy for mobile teleoperation of personal service robots," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 933–940.

[32] T. Witzig, J. M. Zöllner, D. Pangercic, S. Osentoski, R. Jäkel, and R. Dillmann, "Context aware shared autonomy for robotic manipulation tasks," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5686–5693.

[33] M. Ciocarlie, K. Hsiao, A. Leeper, and D. Gossow, "Mobile manipulation through an assistive home robot," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5313–5320.

[34] R. B. Rusu, I. A. Şucan, B. Gerkey, S. Chitta, M. Beetz, and L. E. Kavraki, "Real-time perception-guided motion planning for a personal robot," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 4245–4252.

[35] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stückler, A. Berner, J. Li, R. Klein, and S. Behnke, "Mobile bin picking with an anthropomorphic service robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2327–2334.

[36] 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, http://ompl.kavrakilab.org.

[37] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5026–5033.