# Making Objects Graspable in Confined Environments through Push and Pull Manipulation with a Tool

Sarah Elliott, Michelle Valente and Maya Cakmak

*Abstract*— **Grasping objects in confined environments, such as shelves, fridges, or drawers, is challenging due to the difficulty of avoiding gripper and arm collisions with the surfaces surrounding the object. In this paper we explore the use of a tool to reconfigure objects in such environments so as to make them graspable. The proposed tool has a simple form that allows it to be used in confined environments and a high friction tool tip that enables not only *pushing* objects but also *pulling* them. Our approach involves learning predictive models of pre-defined object-directed tool actions from experience. For each action, we train a multi-modal regressor that maps the initial state of an object to changes in that state, such that future states of the object can be estimated. These allow the robot to choose a sequence of tool actions that yield graspable configurations. We demonstrate that our approach enables a PR2 robot to grasp five different objects from different, initially ungraspable, configurations on a shelf.**

## I. INTRODUCTION

Grasping is one of the most important and well-studied capabilities for robots with manipulators and grippers. It allows a robot to gain full control over an object, enabling its transportation, reconfiguration, or modification. Despite the tremendous progress in grasping research, driven by new perception and planning algorithms, grasping objects in confined and cluttered environments is still difficult.

One approach explored in the literature for dealing with such challenging environments is to use non-prehensile manipulation of objects to reconfigure them in a way that facilitates or avoids grasping. This can involve pushing the target object before grasping it [4] or pushing other objects away to enable grasping of the target object [5], [12]. Previous work exploring this approach has focused on the use of the robot's gripper or arm to accomplish the non-prehensile manipulation. In this work, we propose using a simple tool specifically designed to manipulate objects in confined environments. The size and shape of the proposed tool is chosen to be ideal for *pushing* objects through point and surface contacts within confined spaces. At the same time, the high friction end-effector of the tool allows for *pulling* objects without requiring an articulated form that reaches behind the object.

We present a method to apply the tool on non-graspable objects so as to make them graspable. Our method is based on predictive models of object-relative tool actions. These models are multi-modal regressors that are learned from experience and allow the robot to predict the change in the state of an object due to a tool action. We design ten tool actions and systematically evaluate our approach in the context of grasping objects from a shelf with a PR2 robot. We characterize the behavior of the tool actions, evaluate the effectiveness of the learned models, and demonstrate that the robot is able to appropriately use the tool to make objects graspable.

The authors are with the Computer Science and Engineering Department, University of Washington, Seattle, WA 98195, USA.

## II. RELATED WORK

Previous work has explored the use of a robot's end-effector for reconfiguring objects through non-prehensile manipulation (*i.e.,* without achieving a rigid grasp on the object and control it in 6D) [15]. The majority of this work focuses on the planning problem of efficiently computing a sequence of various robot actions to move an object from an initial configuration to a target configuration [1], [3], [6], [16], [8], [21], [18], [20], [14], [23], [5]. Many of these techniques extend sampling-based planners to different task spaces with alternative models of actions or object-manipulator interactions. For instance Barry *et al.* propose a sampling-based algorithm for planning with a combination of manipulation actions such as picking and pushing [1]. Dogar *et al.* address the problem of picking things up from a cluttered shelf by planning motions that simultaneously move a target object into the gripper while pushing clutter away [3].

Models of a robot's manipulation actions can be at different levels of fidelity, from physics based models that involve force interactions on objects [19], [12], [3], [14] to high-level abstract models in the robot's perceptual space [11], [20], [10], [8]. Others avoid modeling the the effect of actions at the level of state changes but rather directly predict the quality of the expected next state [6]. Methods exist both for continuous or parametrized action spaces [3], [8] and discrete sets of primitive actions [6], [17], [1].

Much of previous research involves empirically acquiring parameters of action models. For example, in Dogar's work, the response of a cylindrical object in response to pushing it with the robot's gripper from different initial positions is obtained empirically [4]. Mericli *et al.* learn how moveable furniture behave in response to pushing with a mobile robot [17]. Similar to our approach, Scholz *et al.* create empirical models of how objects on a table move in response to six predefined push actions [20]. Hermans *et al.* learn how objects of arbitrary shape will behave in response to pushing from different directions [7]. Most work focuses on non-prehensile manipulation using the robot's gripper or differently shaped end-effector. Although using a tool that is rigidly connected to the robot's gripper is

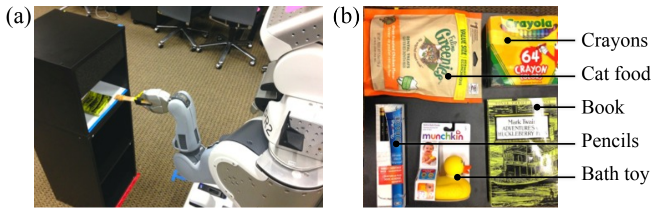

Fig. 1. (a) Experimental setup with the shelf in front of the robot. (b) Five objects used in experiments.

equivalent to controlling the robot's end-effector, Stoytchev's work, investigates the use of differently shaped tools, such as hooks, for manipulating objects [22].

Our work differs from previous work in several ways. Unlike most work our focus is not on planning; rather, we focus on the empirical modeling of tool actions and exploit the expressiveness of pre-defined tool actions using a very simple planning approach. We propose the use of a novel tool specifically designed for manipulation in confined environments. While most previous work aims to enable general object-reconfiguration or to characterize the object through manipulation, we focus on reconfiguring objects specifically for the purpose of grasping them afterwards.

## III. APPROACH

### A. Domain

The scenario considered in this paper involves picking up objects from a shelf. The starting point for this scenario is the Amazon Picking Challenge (APC)[1] which involves picking up Amazon.com items from small shelves in a warehouse setting. One challenge for picking up objects in this scenario is to find a grasp plan that avoids collisions between the shelf and the robot's gripper and arm. Many other real world scenarios involving manipulators, such as fetching items from a fridge, drawer, cabinet, bin, or shelf, involve the same challenge. In this paper we consider a more common class of shelves, different front the APC shelf in that it does not have an incline and lip at the edge to prevent objects from falling when the shelf moves. The shelf has a cell with a width of 38 cm, a height of 26 cm and a depth of 38 cm, shown in shown in Fig. 1(a).

We use a subset of five objects from the APC shown in Fig. 1(b): a box of *crayons*, a box of *pencils*, a *book*, a reclosable pack of *cat food*, and a *bath toy* in its original packaging. The objects are chosen to be diverse in size, shape, and default orientation (standing up versus lying on a surface), hence creating a different set of challenges for grasping.

### B. Platform

The robot platform used in this research is the PR2 (Personal Robot 2) which is a dual-arm mobile manipulator with an omnidirectional base. The PR2's arms have 7 degrees of freedom (DOFs). The last joint positioned at the wrist is

[1]http://amazonpickingchallenge.org/

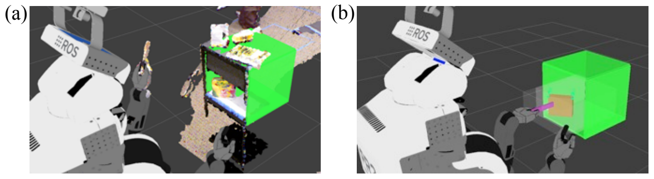

Fig. 2. (a) Point cloud of shelf and object scene as seen by the robot; localized shelf shown in green. (b) Shelf cell (transparent gray), detected object bounding box (yellow), and model of the tool (pink).

capable of 360 degree continuous rotations. The arms have 1-DOF under-actuated grippers with parallel fingertips that open to a max distance of 8 cm. For perception of objects the robot uses a Kinect sensor mounted on the robot's pan-tilt head.

### C. Perception

Grasping objects and manipulating them with a tool requires segmenting the object from its background and localizing it relative to the robot. In this work, we assume that the robot has an accurate model of the shelf on which the items are placed and is able to localize the shelf. To segment and localize objects that are on a certain cell of the shelf, the robot crops the point cloud obtained from the Kinect to only include points within the localized shelf cell. Then it clusters these points into smaller point clouds, based on $k$-means clustering, with each point cloud corresponding to an object. In this work we assume that there is only one object on the shelf or the target object is indicated by the user.

The grasp planning algorithm used in this work directly takes the segmented point cloud as input. Tool actions, on the other hand, are defined relative to the smallest *bounding box* (i) that encloses the object and (ii) whose axes are aligned with the principle components of the point cloud projected onto the horizontal plane.

### D. Grasping

The grasping process has two steps. First the robot detects potential gripper positions in which closing the gripper would result in a rigid grasp (*i.e.,* a net zero force on the object). Since the gripper only has two contact points, this relies on the friction force generated at the contact points. To detect such grasp poses, we use an implementation of the heuristic grasp generation algorithm in [9], which is provided in the *PR2 Gripper Grasp Planner Cluster*[2] package for the Robot Operating System. This algorithm takes the segmented object point cloud as input. It generates candidate gripper positions near the object at different approach angles. A candidate is considered a potential grasp point if it has a number of object points between the two fingertips and no points intersecting with the gripper. Next, the robot eliminates some of the potential grasps if they collide with obstacles in the environment, *i.e.,* other objects or the shelf.

Given a set of potential non-colliding grasps, the robot then tries to find a sequence of joint angles that lead

[2]wiki.ros.org/pr2_gripper_grasp_planner_cluster

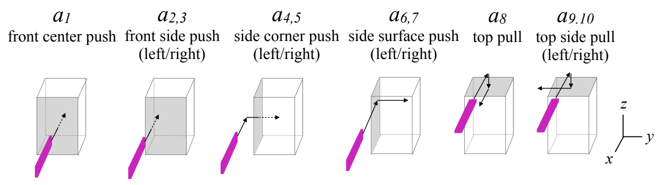
$a_1$
front center push
$a_{2,3}$
front side push (left/right)
$a_{4,5}$
side corner push (left/right)
$a_{6,7}$
side surface push (left/right)
$a_8$
top pull
$a_{9,10}$
top side pull (left/right)
z
y
x

Fig. 3. Ten pull and push tool actions used for reconfiguring the object.

to placing the gripper at the desired pose while avoiding collisions along the way. To that end we use the MoveIt[3] implementation of a motion planning algorithm called RRT-Connect [13]. If no such plan exists, the corresponding grasp is eliminated. If at least one such plan exists, the object is considered graspable. If the object is not visible on the shelf, the input point cloud is empty and the object is not considered graspable.

## E. Tool Actions

The robot reconfigures non-graspable objects using a simple tool shown in shown in Fig. 1(a). The tool is a rectangular prism with dimensions 160 mm length, 30 mm width, and 5 mm thickness. On one end it has a handle shaped like the PR2s fingertips, with edges that prevent it from sliding. On the opposite end it has a textured silicon area with a high friction constant to enable pulling with the tool. The tool sits on a mount on the PR2's shoulder where the robot is able to grasp the tool with a pre-specified motion.

Tool actions are represented as a sequence of 6D tool poses relative to the bounding box of the object. We implement ten different tool actions (illustrated in Fig. 3):

- **Front center push** $(a_1)$: (i) approach the object from the front surface, centered in the $y$-direction with tool tip facing the object; (ii) move tool towards the object in $x$-direction past beyond the front surface; (iii) revert.
- **Front side push (R/L)** $(a_2, a_3)$: (i) approach the object from the front surface, on one side (right/left) of the object in the $y$-direction with tool tip facing the object; (ii) move tool towards the object in $x$-direction past beyond the front surface; (iii) revert.
- **Side corner push (R/L)** $(a_4, a_5)$: (i) approach the object from one of the side surfaces, close to the front edge; (ii) move tool towards the object in $y$-direction past beyond the side surface; (iii) revert.
- **Side surface push (R/L)** $(a_6, a_7)$: (i) approach the object from one of the side surfaces, past beyond the back edge; (ii) move tool towards the object in $y$-direction past beyond the side surface; (iii) revert.
- **Top pull** $(a_8)$: (i) approach object from the top surface, centered in the $y$-direction, (ii) move tool down until contact is made, (iii) move tool away from the object in the $x$-direction, (iv) move tool up to break the contact.
- **Top side pull (R/L)** $(a_9, a_{10})$: (i) approach object from the top surface, centered in the $y$-direction, (ii) move tool down until contact is made, (iii) move towards one of the sides in the $y$-direction, (iv) move tool up to break the contact.

Given the object bounding box, the robot considers the surface between the two leftmost edges as the left surface and the two rightmost edges as the right surface. The front and back are assigned accordingly. Different parameters of the tool actions, such as *pushing height* (for $a_1$–$a_7$) or *pushing/pulling distance* (for $a_1$–$a_{10}$), were tuned empirically through experiments with objects of various shapes. Since the robot always holds the tool in the same configuration from the tool handle, the transform from tool poses to end-effector poses is fixed. As a result the robot can use the same motion planner, as the one used for grasping, to achieve the action-specific tool pose sequence. The tool is added to the robot model for this planning process to avoid collisions of the tool with the shelf and the target object (except for poses that are intended to contact the object). If a no-collision plan does not exist for a given object configuration, the action is considered unavailable.

## F. Learning Tool Action Models

The core problem addressed in this paper is finding the right sequence of tool actions to make a non-graspable object graspable. To that end, the robot needs a predictive model of how a tool action will effect an object. The method used In this paper is to learn such a model from experience.

We represent the state of an object on the shelf with the pose of its bounding box. Since most actions are intended to translate and rotate the object on the shelf surface, we only consider its 3D planar pose rather than its 6D pose. To account for actions that might tip the object over, we also consider the dimensions of the object as part of its state. Hence the complete state representation is still 6 dimensional: $s = (x, y, \theta, h, w, d)$[4].

The problem is then framed as learning a function that can predict the final state of an object $s'$ or the change in the object state $s' - s$ when the action $a$ is applied to the object in state $s$. Since we have a discrete set of actions, we choose to learn a separate function for each action; $e = s' - s = f_a(s)$. As $s$ and $s'$ are multi dimensional continuous variables, this is a multi-modal regression problem.

In this paper, we explored the use of two multi-modal regression methods: Linear regression (LR) with ordinary least squares and Gaussian Process regression (GPR). In both of these techniques, the output variables of the regressor (*i.e.,* individual dimensions of $e$) are assumed to be independent. LR is chosen for its simplicity and interpretability of the learned model. GPR is chosen for its ability to capture non-linearities and for providing a confidence measure (mean squared error) associated with each prediction. They have been demonstrated to be effective in learning transition models (similar to the tool action models in this paper) in the reinforcement learning framework [2]. We used the scikit[5] implementation of both regression methods.

[3]www.moveit.org

[4]This representation is equivalent to the 6D pose of the object, but we preferred it for its ease of interpretation.

[5]http://scikit-learn.org/

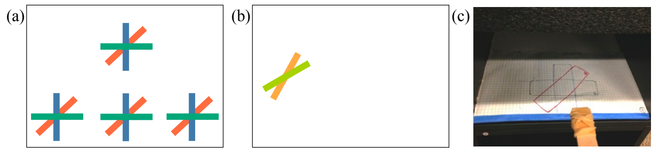

Fig. 4. Top view of initial object configurations on the shelf for datasets (a) $\mathcal{D}_1$ and (b) $\mathcal{D}_2$. (c) Example sheet showing object configuration outline used during data collection.

## G. Planning Tool Action Sequences

The predictive models of the tool actions allow the robot to estimate the new pose of an object after the action is applied. To evaluate whether the object is graspable or not in a predicted state, the robot transforms the original point cloud of the object to the predicted pose and performs a grasp evaluation as described previously.

To use the learned predictive models as part of planning a sequence of actions that make an object graspable, we propose a simple forward planning algorithm. First, the robot creates a prediction tree that has the current state of the object at the root and expands the tree by adding child nodes that correspond to predicted states after all possible actions are applied to the state in the parent node. Each edge in the tree corresponds to a tool action. Each node corresponds to the predicted state of the object after the actions on the path from the root to the node have been applied to the object. Nodes have a confidence, $p(s_t)$, associated with them. These are computed by accumulating the confidence of each prediction along the plan; $p(s_t) = \prod_{i=0}^{t} p(s_i)$ where $s_0, .., s_t$ is the unique path on the tree that leads to the node $s_t$ through a sequence of actions $a_1, .., a_t$.

After expanding the tree to a fixed depth, the robot evaluates the graspability of the object in the predicted poses corresponding to each node. To allow comparisons of alternative plans, we use a continuous measure of graspability, $g(s_t)$, computed as the number of possible alternative grasps on the object in a predicted state. To select a plan from the expanded tree, we assign each node a utility $u(s_t) = p(s_t)g(s_t)$ and choose the plan leading to the node that maximizes this utility; $s_t^* = \arg\max_{s_t \in T}(u(s_t))$. This allows a balance between our confidence in reaching the predicted state and the quality of the grasp in the predicted state.

## IV. EVALUATION

We evaluate our approach in the domain described in Sec. III-A. We first describe the data collected for training and testing the predictive models, then present an analysis of the predictive model performance, and lastly analyze the performance of the overall system in finding plans.

## A. Data

To train and test the multi-modal regressors we collected a dataset consisting of $(s, a, s')$ tuples, where $s$ is the initial state, $a \in \{a_1, .., a_{10}\}$ is the tool action, and $s'$ is the state of the object observed after the action is applied. We refer to the collection of each tuple as a *trial*. We collected two datasets (summarized in Table II):

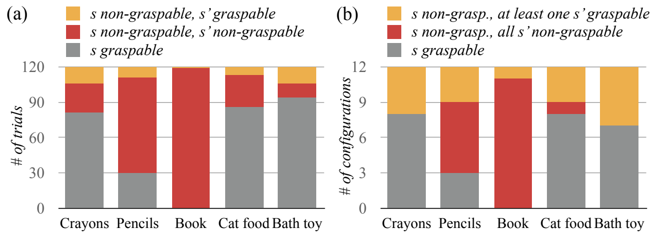

Fig. 5. (a) Distribution of trials in the $\mathcal{D}_1$ dataset (out of 120 for each object) in terms of whether objects were graspable or not before and after the tool actions. (b) Distribution of configurations in $\mathcal{D}_1$ according to whether the object was graspable (in majority of the trials) or not before the tool action and whether there was at least one action that made the object graspable.

- $\mathcal{D}_1$ systematically covers different positions and orientations on the shelf
- $\mathcal{D}_2$ involves edge cases in which the object is unlikely to be initially graspable

These datasets are specific to the particular tool and shelf we used. For $\mathcal{D}_1$ we varied $s$ by placing each of the five objects in four different positions on the shelf in three different orientations as illustrated in Fig. 4(a). $\mathcal{D}_2$ involved positioning each object adjacent to the shelf in two intermediate orientations illustrated in Fig. 4(b). We used different subsets or combinations of these datasets throughout our evaluation.

TABLE I
SUMMARY OF COLLECTED DATASETS (SEE ALSO FIG. 4).

| name | objects | positions | orientations | actions | total |
|---|---|---|---|---|---|
| $\mathcal{D}_1$ | 5 | 4 | 3 | 10 | 600 |
| $\mathcal{D}_2$ | 5 | 1 | 2 | 10 | 100 |

Before moving on to learning results, we characterize the behavior of the tool actions explored in this paper based on $\mathcal{D}_1$. We saw that 48.5% (291/600) of trials in $\mathcal{D}_1$ have an initial state $s$ in which the object is graspable. Out of the trials with non-graspable initial configurations, 14.5% (45/309) became graspable after a single tool action. When objects are already graspable, tool actions can have a destructive effect: 9% (54/600) of configurations in $\mathcal{D}_1$ were initially graspable and the object became non-graspable after the tool action.

The distribution of trials in terms of whether objects were graspable or not before and after the tool actions is further broken down by objects in Fig. 5(a). We see that the pencils and the book were particularly challenging: they were not initially graspable in a larger set of trials and fewer tool actions made them graspable. As seen in Fig. 5(b), these two objects had a higher number of configurations in which none of the actions made the object graspable. The cat food also has one configuration in which none of the tool actions worked, while the crayons and the bath toy could be made graspable by at least one action in all configurations.

To allow an investigation of regularities in the effects of different tool actions, Fig. 6 provides $s' - s$ (for the first three dimensions only) grouped by actions. We observe that

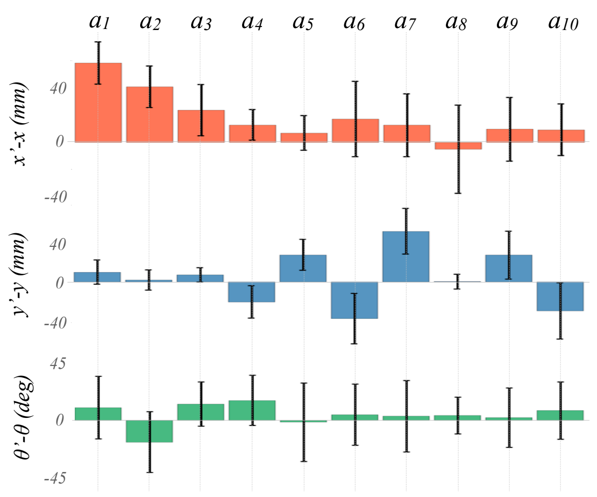

Fig. 6. Effects of the ten tool actions on the object state (changes in $x$, $y$, and $\theta$) based on $\mathcal{D}_1$. Error bars show standard deviation.

the average effects of actions are as expected. For example, $a_1$ (front center push) causes a large change in the $x$-axis; it moves the object away. $a_8$ (forward pull) has the opposite effect of pulling the object forward, with a negative change in the $x$-axis. $a_2$ and $a_3$ (front side push) rotate the object in opposite directions (in $\theta$), while also pushing it away. Similarly, the action pairs $a_4$–$a_5$ (side corner push), $a_6$–$a_7$ (side surface push), and $a_9$–$a_{10}$ (side pull) move the object in opposite directions towards the sides ($y$-axis).

Although tool actions show regularities on average, we observe a large variance in their effects. This implies that the effect of a tool action may depend on the particular object and its state. The goal of the learned predictive models, analyzed in the next section, is therefore to predict the particular effect within the observed variance.

## B. Analysis of Learning Predictive Models

*1) Regression errors:* As described in Sec. III-F, we explored two different multi-modal regression techniques to address our problem of learning predictive models of tool actions. Fig. 7 shows the average errors of the trained regressors on all dimensions of the output space ($s' - s$). We see that Gaussian Process (GP) regression leads to a very small error on the training set, as compared to the Linear regressor, as it is able to better capture the non-linearities in the data due to tipping of objects or hitting the edges of the shelf. The error of the Linear regressor is nonetheless low (around 1cm in $x$ and $y$-axis translation, 10 degrees in yaw rotation). When tested with configurations that were not in the training data ($\mathcal{D}_2$) GP regressors resulted in higher error, showing evidence of over-fitting. The Linear regressor generalized better to unseen configurations, with errors remaining within a similar range. Despite this finding, our analysis focuses on GP regressors as it is the method that supports our planning approach.

When the prediction errors are broken down by action, as shown in Fig. 8 we see that the errors reflect the way in which the test data is different from the training data.

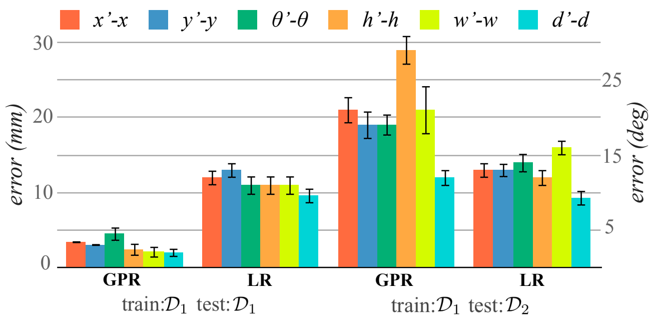

Fig. 7. Error for Gaussian Process and Linear regressors trained with the $\mathcal{D}_1$ dataset and tested on (left) $\mathcal{D}_1$ and (right) $\mathcal{D}_2$. $\theta$ errors are in *degrees* and all other dimensions are in *mm*s. Error bars indicate standard error.

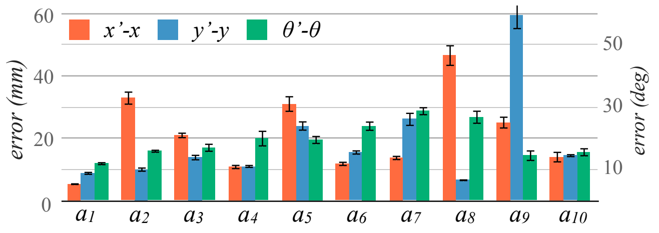

Fig. 8. Error for the Gaussian Process regressors trained with $\mathcal{D}_1$ and tested on $\mathcal{D}_2$ split by the ten tool actions. $x$ and $y$ errors are in *mm*s and $\theta$ errors are in *degrees*. Error bars indicate standard error.

In particular, the objects being placed near one of the sides of the shelf (Fig. 4(b)) caused the actions to have effects that were different from expected. For example, $a_9$ (side pull, left) normally makes the object move towards the left. However, in the test data all objects hit the wall to their left, preventing them to move along the $y$-axis. As a result the prediction error in this dimension was particularly high for $a_9$. Similarly, $a_2$ (front side push, left) normally results in a backwards movement in addition to a rotation. However, in the test data the rotation made the object contact the wall which in turn blocked the movement along the $x$-axis, causing a larger prediction error.

Breaking the errors down by object also helps further understand the sources of large error (Fig. 9(a)). We observe that the last two objects (cat food and bath toy) have a larger overall error. Unlike the first three objects these objects do not have a rectangular prism shape, making the bounding box a less accurate representation of the object. As a result, we qualitatively observed during data collection that tool actions (which are relative to the bounding box) are less consistent in the effects they have on these two objects with more complex shapes. For instance, neither of these objects have a proper top surface that allow pulling motions ($a_8 - a_{10}$) to have the intended effects. Similarly, $a_4 - a_5$ are relative to the front edge on each side, whereas the cat food package does not fill those parts of the bounding box. As a result these actions most often fail in making contact with the object.

To further explore our observation that differences among objects can be a source of error, we tried training regressors with data from $\mathcal{D}_1$ for *only one object* and test it with data from $\mathcal{D}_2$ for the same object (Fig. 9(b)). This resulted in

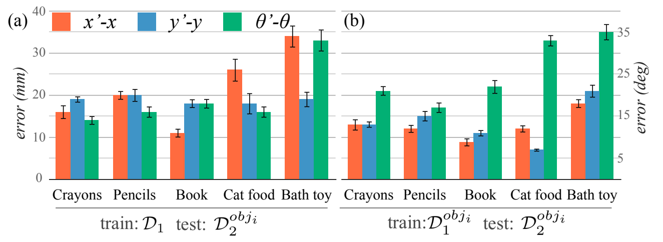

Fig. 9. Error for Gaussian Process regressors (a) trained with $\mathcal{D}_1$ and tested on $\mathcal{D}_2$ broken down by object, and (b) trained with individual object datasets $\mathcal{D}_1^{obj_i} \in \mathcal{D}_1$ and tested on $\mathcal{D}_2^{obj_i} \in \mathcal{D}_2$ for the same object. $x$ and $y$ errors are in *mms* and $\theta$ errors are in *degrees*. Error bars indicate standard error.

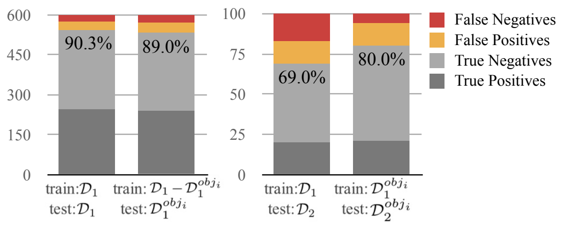

Fig. 10. Accuracy of predicting whether an object will be graspable or not after a tool action (shown in percentage), with different configurations of training and testing datasets.

overall more accurate regressors, particularly for predicting translational changes as compared to rotational changes. Consistent with our earlier observation, the errors were still relatively higher for objects with more complex shapes that yield inconsistent changes in response to tool actions.

*2) Grasp prediction accuracy:* While predictive action models can be used for planning specific motion trajectories for an object, the focus in this paper is to make the object graspable. Hence it is important to know if the predictive model enables correct assessment of future object states in terms of their graspability. To that end, we compared the observed graspability of objects in our datasets with the assessed graspability of predicted future states (computed as described in Sec. III-G). Fig. 10 shows the accuracy of such predictions for regressors trained in different ways, over different test sets.

The accuracy is reasonable overall. We see that the prediction errors of the regressors are reflected in the grasp predictions; *i.e.,* test cases with larger prediction error have lower grasp prediction accuracy. We include an additional analysis in which the regressors are trained with data from four objects and tested on the fifth object, giving a relatively high accuracy of 89%. This demonstrates the potential for good transfer to new objects within the space of configurations observed in the dataset.

*3) Single action grasp creation:* As discussed earlier with reference to Fig. 5, in some cases an object that is not graspable can be made graspable with a single action. While our goal is to find a *sequence* of actions that can take an object from *any* configuration to a graspable configuration (discussed in the next section), we take a moment to characterize ways in which a single tool action moves an object

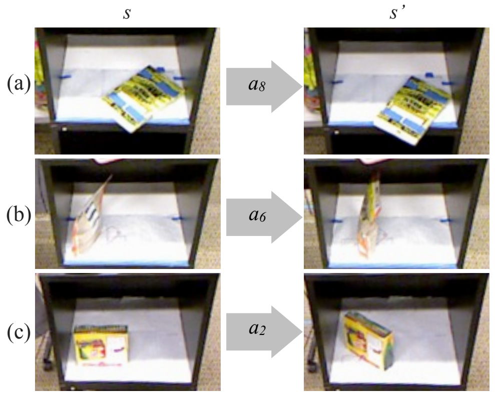

Fig. 11. Example trials from $\mathcal{D}_1$ in which the initial state $s'$ of the object is not graspable and the final state is graspable and correctly predicted as being graspable.

from an ungraspable configuration to a graspable one.

Fig. 11 shows three examples of before and after states from trials in $\mathcal{D}_1$. Each trial has an $s$ that is not graspable and $s'$ that is graspable and correctly predicted to be so. We observe three ways in which tool actions rescue objects from ungraspable configurations:

- Fig. 11(a): By pulling an object towards the front edge of the shelf, tool actions can create a graspable area on objects that cannot be grasped when laid on a flat surface (*e.g.,* book).
- Fig. 11(b): By moving objects away from the side surfaces of the shelf, tool actions can reposition objects in a way that allows the robot's bulky gripper to fit around the object.
- Fig. 11(c): By rotating objects, tool actions can reposition objects in ways that expose graspable portions of the object in the front side of the shelf enabling collision free access to grasps on those portions.

## *C. Analysis of Planning Tool Action Sequences*

In some configurations objects cannot be made graspable with a single tool action. The planning approach described in Sec. III-G targets such scenarios. Although the proposed planning technique is computationally expensive for large plan depths, we empirically found that (i) all configurations that are initially ungraspable (in $\mathcal{D}_1$ and $\mathcal{D}_2$) have at least one plan of depth=2 that is predicted to make the object graspable, and (ii) the utility metrics used in our paper do not favor longer plans over these two-step plans. Hence our analysis was limited to two-step plans.

We first tested our planning approach on all ten configurations in $\mathcal{D}_2$. Fig. 12(a) shows the number of one-step and two-step plans: we see that only three configurations did not have a one-step plan that resulted in at least one predicted grasp. Out of the 100 possible two-step plans we saw that 10 to 30 had at least one potential grasp. Fig. 12(b) shows the average number of grasps, considered as our metric for the quality of an object configuration in terms of graspability, in one-step and two-step plans. We see that two step plans tend to have more grasps in the final state; however, the difference

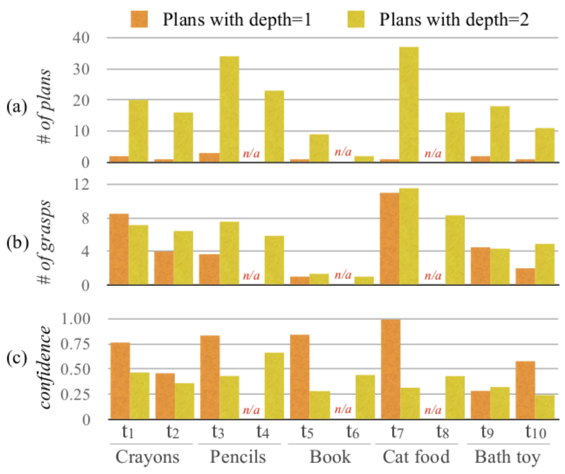

Fig. 12. (a) Number of plans with one or two actions that have at least one grasp; (b) average number of grasps and (c) average confidence ($p(s_t)$) in plans with one or two actions for 10 test configurations ($t_1 - t_{10}$, Fig. 13(a)).

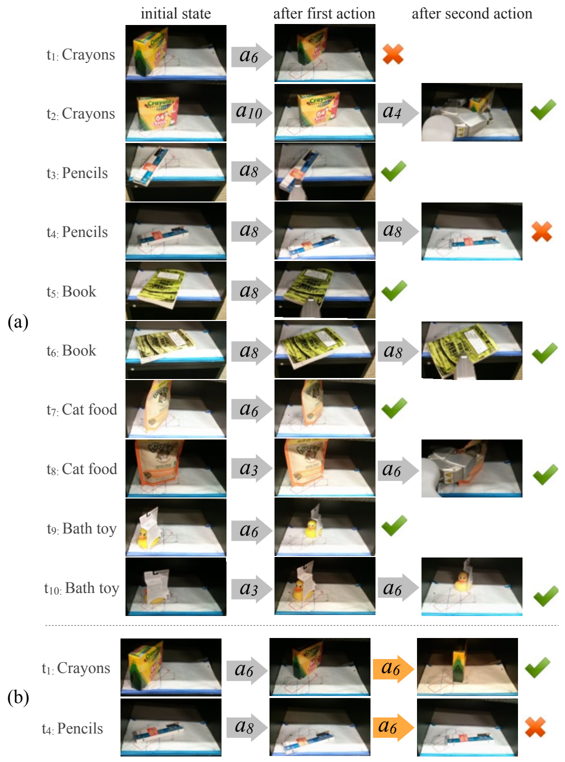

Fig. 13. (a) Action plans based on $u(s_t)$ in 10 test configurations. The grasp selected by the robot is also shown if the gripper does not preclude seeing the final object state. Check marks indicate successful final grasp. (b) Dynamic plans that were different from the original plans.

is not high. Fig. 12(c) shows the average confidence for one-step and two-step plans, which most of the time reduces with more steps, as expected.

Table II shows the plans selected for the ten test configurations (out of 110 candidates in the plan tree of depth 2) based on three different utility metrics: (i) solely based on quality of the predicted configuration (*i.e.,* number of grasps $g(s_t)$), (ii) solely based on confidence ($p(s_t)$), and (iii) based on the combined utility metric ($u(s_t)$, Sec. III-G). Although the three metrics can yield the same action plan, plans chosen based on $g(s_t)$ and $p(s_t)$ are often different and plans chosen based on $u(s_t)$ is equivalent to one or the other. In some configurations ($t_2$, $t_{10}$), two-step plans are favored even though one-step plans exist (Fig. 12(a)).

TABLE II

PLANS CHOSEN BASED ON DIFFERENT METRICS.

| test configuration | $u(s_t)$ | $g(s_t)$ | $p(s_t)$ |
|---|---|---|---|
| $t_1$ | $a_6$ | $a_4$ | $a_6$ |
| $t_2$ | $a_{10}$, $a_4$ | $a_{10}$,$a_4$ | $a_{10}$, $a_6$ |
| $t_3$ | $a_8$ | $a_8$ | $a_8$ |
| $t_4$ | $a_8$, $a_8$ | $a_8$, $a_8$ | $a_8$, $a_6$ |
| $t_5$ | $a_8$ | $a_8$ | $a_8$ |
| $t_6$ | $a_8$, $a_8$ | $a_8$, $a_8$ | $a_8$, $a_8$ |
| $t_7$ | $a_6$ | $a_6$ | $a_6$, $a_4$ |
| $t_8$ | $a_3$, $a_6$ | $a_3$, $a_4$ | $a_3$, $a_6$ |
| $t_9$ | $a_6$ | $a_4$ | $a_6$ |
| $t_{10}$ | $a_3$, $a_6$ | $a_6$ | $a_3$, $a_6$ |

Fig. 13(a) shows the execution of the plans chosen based on $u(s_t)$ in the ten test configurations. We see that most plans result in the robot being able to grasp the object at the end. In the first test that failed in execution ($t_1$) the chosen one step plan was intended to rotate the crayons but failed due to the proximity to the shelf. When the robot was allowed to re-plan during execution (Fig. 13(b)) it chose to repeat the same action and this time succeeded to rotate the object to a graspable configuration. In the second test that failed ($t_4$) chosen plan was intended to pull the object towards the shelf edge in two steps, but failed to have the expected effect in the second step. When the robot was allowed to re-plan during execution (Fig. 13(b)) it chose a different action for the second step, but that action also failed to have the intended effect. Note that both the original and the dynamically generated plans for $t_4$ are viable and are likely to succeed some of the time. This experiment demonstrates that plan executions are subject to uncertainty. This supports the approach of dynamic re-planning during execution.

In the plans with successful executions, we see a variety in the use of different tool actions. $a_8$ (pulling object forward) is the most preferred action for flat objects (book and pencils). We observe uses of alternative actions to achieve similar effects; for instance $a_3$ and $a_4$ for rotating the object; and $a_6$ and $a_{10}$ for pushing or pulling the object towards the right.

We also tested our planning approach on configurations in $\mathcal{D}_1$ where no action made the object graspable in a single step (shown as red in Fig. 5). There were 18 such configurations in total, for three of the objects (pencils:6, book:11, cat food:1). We saw that the object could be recovered with two-step plans in all of these configurations. There was an average of 7.4 two-step plans (pencils:14, book:5, cat food:8.2) and the plans had an average of 2.7 potential grasps (pencils:11.5, book:3.3, cat food:1.6).

## V. DISCUSSION

Our results demonstrate an overall success in achieving what we set out to do: using a simple tool to help grasp objects in confined environments. Our method enables a simple forward planning approach that effectively uses the tool to reconfigure an object in very few steps (one or two actions). We believe that part of this success can be attributed to the design of our tool and our tool actions. Our tool had a size and shape ideal for being used in confined spaces and particularly unique in enabling pulling actions with the help of friction forces. The tool actions were carefully designed to avoid collisions with the shelf while being useful and applicable to different objects in a range of configurations.

The idea of empirically modeling tool action effects proved to be sensitive to the way in which multi-model regressors are trained and tested. The problem was made more challenging by (i) the testing data having different action behaviors due to unseen configurations (*e.g.,* objects hitting walls), (ii) having multiple objects that behave differently in the same situation, and (iii) uncertainty in the action effects that can be drastically different (*e.g.,* tipping over). Nonetheless, the scale regression errors were not detrimental on the robot's ability to predict future graspability and find productive tool action plans.

Our paper focused on using predictive models to reconfigure a *target* object; however, the learned models can also be used to reconfigure objects that act as clutter. In that case the robot would explore plans that reconfigure the clutter, then chose a plan that make the target object graspable, for instance, by pushing clutter away from the object or deeper into the shelf. This is a simple extension of the work presented in this paper, but will likely require longer plans.

A more involved extension of our work would be to consider exploration of tool action *parameters* instead of keeping them fixed. In that case, the parameters would be used as input to the regressors which could discover correlations between input parameters and action effects. Unlike other inputs to the regressor (*i.e.,* object state), parameters would not be observed during execution. Therefore, the planning approach would need to be modified to intelligently select action parameters to achieve the best effect.

## VI. CONCLUSION

This paper explores the idea of using a simple tool to reconfigure an object in confined environments so as to make it graspable. We propose using a novel tool with a high friction tool tip that enables *pulling* actions as well as *pushing* actions. Our approach involves learning to predict the change in the continuous state of the object in response to predefined tool actions, by training a multi-modal regressor with experience data. We present a thorough analysis of different components of our system, considering alternative regression techniques and plan utility metrics. We demonstrate that our approach enables a PR2 robot to pick up five different objects from various configurations on a shelf.

## ACKNOWLEDGMENT

The authors thank Becky Leslie for helping collect the datasets used in experiments presented in this paper.

## REFERENCES

[1] J. Barry, K. Hsiao, L.P. Kaelbling, and T. Lozano-Pérez. Manipulation with multiple action types. In *Experimental Robotics*, pages 531–545. Springer, 2013.

[2] M. Deisenroth, C. Rasmussen, and D. Fox. Learning to control a low-cost manipulator using data-efficient reinforcement learning. In *Proceedings of Robotics: Science and Systems*, 2011.

[3] M.R. Dogar, K. Hsiao, M. Ciocarlie, and S.S. Srinivasa. Physics-based grasp planning through clutter. In *Robotics: Science and Systems (RSS)*, 2012.

[4] M.R Dogar and S.S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2123–2130, 2010.

[5] M.R. Dogar and S.S. Srinivasa. A planning framework for non-prehensile manipulation under clutter and uncertainty. *Autonomous Robots*, 33(3):217–236, 2012.

[6] M. Gupta and G.S. Sukhatme. Using manipulation primitives for brick sorting in clutter. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3883–3889. IEEE, 2012.

[7] T. Hermans, F. Li, J.M. Rehg, and A.F. Bobick. Learning stable pushing locations. In *IEEE Intnl. Conf. on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–7. IEEE, 2013.

[8] T. Hermans, J.M. Rehg, and A. Bobick. Guided pushing for object singulation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4783–4790. IEEE, 2012.

[9] Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E. Gil Jones. Contact-reactive grasping of objects with partial shape information. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1228–1235, Oct 2010.

[10] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 272–277. IEEE, 2008.

[11] C.C. Kemp and A. Edsinger. Robot manipulation of human tools: Autonomous detection and control of task relevant features. In *Proc. of the Fifth Intl. Conference on Development and Learning*, 2006.

[12] J.E. King, J.A. Haustein, S.S. Srinivasa, and T. Asfour. Nonprehensile whole arm rearrangement planning on physics manifolds. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[13] J.J. Kuffner and S.M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 995–1001. IEEE, 2000.

[14] K. M Lynch and M.T. Mason. Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

[15] K.M. Lynch. *Nonprehensile robotic manipulation: controllability and planning*. Carnegie Mellon University, 1996.

[16] Y. Maeda, H. Kijimoto, Y. Aiyama, and T. Arai. Planning of graspless manipulation by multiple robot fingers. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 3, pages 2474–2479. IEEE, 2001.

[17] T. A Mericli, M. Veloso, and H. L. Akin. Achievable push-manipulation for complex passive mobile objects using past experience. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 71–78. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[18] D. Nieuwenhuisen, M. H Overmars, et al. Path planning for pushing a disk using compliance. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 714–720. IEEE, 2005.

[19] J. Scholz, M. Levihn, C.L. Isbell, H. Christensen, and M. Stilman. Learning non-holonomic object models for mobile manipulation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015.

[20] J. Scholz and M. Stilman. Combining motion planning and optimization for flexible robot manipulation. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 80–85. IEEE, 2010.

[21] M. Stilman, J. Schamburek, J. Kuffner, and T. Asfour. Manipulation planning among movable obstacles. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3327–3332. IEEE, 2007.

[22] A. Stoytchev. Behavior-grounded representation of tool affordances. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3060–3065. IEEE, 2005.

[23] N. B. Zumel and M.A Erdmann. Nonprehensile manipulation for orienting parts in the plane. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 3, pages 2433–2439. IEEE, 1997.