Facilitating Robotic Grasping using Pushing and Toppling

Christopher Correa

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-80
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-80.html

May 17, 2019

Facilitating Robotic Grasping using Pushing and Toppling

by

Christopher Correa

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair
Professor Ruzena Bajcsy

Spring 2019

The thesis of Christopher Correa, titled Facilitating Robotic Grasping using Pushing and Toppling, is approved:

Chair ______________________________ Date ____________

______________________________ Date ____________

______________________________ Date ____________

University of California, Berkeley

Facilitating Robotic Grasping using Pushing and Toppling

Abstract

Facilitating Robotic Grasping using Pushing and Toppling

by

Christopher Correa

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

Robots are expected to grasp complex 3D objects in a wide variety of situations. This task can be difficult when the object's pose prevents the robot from perceiving or executing grasps on the object. When robust grasps are not accessible, robots can execute non-prehensile actions such as pushing and toppling to change an object's 3D pose to provide access to robust grasps. We develop two planar pushing policies and evaluate each policy's ability to increase access to robust grasps for both parallel jaw grippers and vacuum suction grippers. Using an ABB YuMi arm, we execute each pushing policy on the same 1000 simulated and physical scenarios in which the quality of all accessible grasps is low, and measure the predicted grasp reliability before and after the push. These experiments suggest that pushing can be used effectively to expose robust parallel jaw grasps, but are less effective in exposing robust vacuum suction grasps. As a result, we explore using toppling to reveal flat object faces for vacuum suction grippers. We present a toppling model which characterizes the robustness of toppling a 3D object specified by a triangular mesh, using Monte Carlo sampling to account for uncertainty in object apose, friction coefficients, and push direction. We run 700 physical toppling experiments using the ABB Yumi arm to compare the performance of the proposed model against empirical outcomes. We find that the toppling model outperforms a baseline model by an absolute 26.9% when comparing the total variation distance between each model's predicted probability distribution and the empirical distribution. We use the robust model as the state transition function in a Markov Decision Process (MDP) to plan optimal sequences of toppling actions to expose access to robust suction grasps. Data from 20,000 simulated experiments suggests the toppling policy can increase suction grasp reliability by 33.6%.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my research advisor Ken Goldberg. Professor Goldberg challenged me to critically analyze my work, which allowed me to become the best researcher I could become.

I also would like to thank my mentors Jeffrey Mahler and Michael Danielczuk. Their guidance helped shape not only my research, but also the way I approach increasingly complex problems and I will remember their advice for all of my future endeavors.

I would also like to thank Professor Ruzena Bajcsy for her mentorship in the field of Robotics, and her advice through my academic career. I also thank Professor Bajcsy for her help as part of my Master's Thesis committee.

Finally, I would like to thank my family for their support throughout my undergraduate and graduate career at the University of California, Berkeley.

Chapter 1

Introduction

1.1 Overview

Grasping a wide variety of objects is essential in home de-cluttering, e-commerce warehouse manipulation, manufacturing, and service robotics. However, the ability to perceive and execute reliable grasps may be limited due to a lack of visibility of graspable points on the object, kinematic feasibility or environment collisions. Non-prehensile actions such as pushing and toppling can be attempted before grasping in order to reorient objects into poses with increased visibility and access to robust grasps.

Planar pushing has the the potential to separate objects from environmental obstacles, exposing antipodal surfaces for parallel jaw grasps. Previous pushing work measures the success of pushes as the degree of separation between objects and attempts to minimize the number of push actions to achieve separation [5, 8]. We explore directly using grasp confidence metrics instead of an indirect metric such as object separation for comparison of push policies, and attempt to maximize grasp confidence over a single push action. This metric differs from previous formulations because a push does not need to completely separate an object to reveal a grasp. In addition, to increase access to grasps on every object, the robot may not need to execute several pushes, as pushing to reveal a grasp, and subsequently execute that grasp could reveal more grasps of nearby objects.

We observe that planar pushing is able to expose robust parallel jaw grasps, but is less effective in exposing robust vacuum suction grasps. In contrast, toppling, the act of pushing an object into a new static resting pose with a robotic point contact, can be used to expose new surfaces for robust vacuum suction grasps.

Models of toppling for extruded 2D shapes have been developed in prior work by Lynch [17, 19] and Zhang *et al.* [37]. In this thesis, we extend these models to 3D objects by predicting planar rotations and topple actions that are not perpendicular to the topple edge, and estimate topple reliability under uncertainty in object and gripper position, friction coefficients, and push direction.

We find that it is sometimes necessary to execute intermediate topple actions in order to expose a robust suction grasp. Therefore, we develop toppling policies which plan sequences of toppling actions to increase access to robust suction grasps, using suction grasp analysis from Dex-Net 3.0 [25].

1.2 Summary of Contribution

This thesis presents a novel metric for evaluating non-prehensile actions. We evaluate a non-prehensile action based on its ability to expose robust grasps. We present a formulation of sequential pushing and toppling as a Markov Decision Process (MDP). We present two novel pushing policies based on targeting free space and diffusing clusters of objects, a toppling policy and two toppling baseline policies targeting poses with exposed flat surfaces. We use the proposed metrics to evaluate these policies by their ability to expose robust parallel jaw and vacuum suction grasps. In 20,000 simulated experiments, the proposed toppling policy increases suction grasp reliability by an absolute 28.4% over the Max-Height Baseline.

This thesis also presents a quasi-static dynamics model for the MDP that estimates topple reliability for a 3D polyhedral mesh when pushed at a given point under uncertainty in object and gripper position, friction coefficients, and push direction. We run 700 physical toppling experiments using an ABB YuMi robot and 3D printed objects. We run a parameter sweep over the uncertainty parameters for object and gripper position, friction coefficients, push direction, and object pose to match the model's predictions with empirical outcomes. Using the optimal parameters, we generate a dataset of 1,257,000 candidate toppling points on 189 3D CAD models in a total of 1257 stable poses, each labelled with associated toppling reliabilities. The proposed model outperforms the baseline model by an absolute 26.9%, when comparing the total variation of each predicted distribution with the empirical distribution.

1.3 Related Work

1.3.1 Planar Pushing

Mason pioneered research on analytic models of push mechanics [26]. Lynch and Akella described the mechanics of stable pushing and described how to create a plan for controlled pushing of an object through a series of obstacles [2, 18, 20, 21]. "Singulation" applies pushing mechanics and planning to the task of separating or extracting objects that lie close together, and it is often required for successful object recognition or grasping. Model-based approaches such as the one proposed by Cosgun *et al.* [6] planned a series of robot actions to clear space for a target object in two-dimensional, tabletop pushing scenarios. Eitel *et al.* [8] explore singulation in clutter using a push proposal convolutional neural network, showing that they can separate up to 8 objects with at least a 40% success rate in an average of 11 push actions. In contrast to their work, which seeks to minimize the number of push actions to separate all objects, we find one push at each opportunity, take into account bin walls and corners, and analyze push success based on new metrics.

1.3.2 Reorienting to a New Stable Pose

One approach to reorient objects to new stable poses designs robotic grippers with multiple contact points designed specifically for the objects of interest, in order to execute several pinching motions while closing the gripper around the object [36, 35]. These grippers perform well with a single object, but do not generalize to the arbitrary objects on which robots are expected to operate. Pivoting, another reorienting motion primitive, is the act of loosely picking up objects so that they can rotate around the parallel jaw gripper's axis due to gravity [4, 31]. Holladay *et al.* [11] introduce a pivot dynamics model and Hou *et al.* [12] combine pivoting and "rolling", rotating the arm while stiffly gripping the object, to more efficiently reorient objects. However, these papers assume two-finger grasps are available on the object, in which case toppling to find robust grasps is unnecessary. We extend these papers by reorienting when grasps are not available.

1.3.3 Toppling

Previous work has developed physics-based quasi-static models of toppling behavior on conveyor belts [17, 19, 37] and with robotic hands to move heavy objects [1]. Yamashita *et al.* [34] explore the use of multiple cooperative robots to topple objects. However, these models only predict motions of 2D extruded shapes, do not model object resistance to planar rotations, and only model pushes perpendicular to the toppling edge. Lee *et al.* [16] discretize all possible poses of the object, search for the contact points necessary to hold the object in each pose, and determine a trajectory to topple the object over. Learning-based approaches have been developed to predict object motions from just RGB images [14, 28, 15], and to learn a toppling policy from demonstrations [30]. These approaches do not require knowledge of the object geometry, but also only operate on 2D extruded shapes. In this paper, we present a model which predicts the toppling behavior of pushes in arbitrary directions for 3D objects beyond 2D extrusions.

Chapter 2

Problem Statement

Given a rigid polyhedral object or set of objects which are not graspable in their current orientation, our goal is to use non-prehensile motions like pushing or toppling to expose a grasp for a robotic parallel jaw gripper or vacuum suction gripper. We formulate this problem as a Markov Decision Process (MDP) and present policies for pushing and toppling to maximize the predicted grasp robustness following the non-prehensile actions.

2.1 Assumptions

Throughout this thesis, we make the following assumptions:

1. Quasi-static physics (i.e., inertial terms are negligible).
2. The forces and torques applied by the robot are consistent with the *point contact with friction* model [29].

Chapters 3 and 4 describe the assumptions specific to pushing and toppling respectively.

2.2 Definitions

1. **State:** Let $\mathbf{x}_t = \{(\mathcal{O}, T_o)\}_{n=0}^{N} \in \mathcal{X}$ represent the state of every object in a heap at time t. $\mathcal{O}$ represents the object's geometrical properties, material properties, and center of mass, and T_o represents the pose of the object.
2. **Actions:** Let $\mathbf{u}_t = (\mathbf{p}, \mathbf{q}) \in \mathcal{U}$ be the linear trajectory of the point pusher manipulator in 3D space between the start point $\mathbf{p} = (x, y, z)$ and the end point $\mathbf{q} = (x', y', z')$. Pushing actions are linear trajectories in which the object slides on the ground and Topple actions are linear trajectories in which the object does not slip, but rotates over an edge.

3. **Rewards:** The reward for executing each non-prehensile action is 0. After executing all actions, the robot gets a reward $V_j(\mathbf{x}_T)$ or $V_s(\mathbf{x}_T)$, the predicted reliability of the best accessible parallel jaw or suction grasp, respectively, of any object in the heap.

4. **Transition Distribution:** Let $\mathbb{P}[\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t]$ be the probability that the object heap transitions into state $\mathbf{x}_{t+1}$ when the robot executes action $\mathbf{u}_t$.

2.3 Objective

Our ultimate goal is to develop a policy which plans an action or sequence of actions $\{\mathbf{u}_t\}_{t=1}^T$ that maximizes the reliability of the best available grasp at time T. The planar pushing policies execute a single action, as grasps revealed from pushing can be executed, therefore revealing further robust grasps of the other objects. The toppling policies execute a sequence of actions, as it is often necessary to topple the object more than once to reveal an optimal face for suction grasps. We consider a subset of the toppling problem where the object is singulated and on a planar workspace to simplify the problem for the presented toppling model.

For the purpose of this analysis, the multi-step policies pick a fixed time horizon trajectory, and execute "no-op" actions $\emptyset$ if they predict no action will increase access to suction grasps.

$$\begin{aligned} \Delta V = \max_{\mathbf{u}_{0..T}} \quad & \mathbb{E}\left[V(\mathbf{x}_T)\right] - V(\mathbf{x}_1) \\ \text{s.t.} \quad & \mathbf{u}_t \in \mathcal{U} \cup \{\emptyset\} \end{aligned}$$

This formulation differs from the standard MDP formulation in that the reward is only applicable for the final action. Although the fixed horizon formulation doesn't explicitly penalize taking unnecessary actions, the policies presented in this thesis attempt to reach the goal state with the fewest actions.

We define the Parallel Jaw Grasp Confidence Gain and Suction Grasp Confidence Gain to be ΔV when $V(\mathbf{x}_t) = V_j(\mathbf{x}_t)$ and $V(\mathbf{x}_t) = V_s(\mathbf{x}_t)$ respectively. While previous work uses object separation as a metric for non-prehensile actions, we use the grasp quality gain as a metric.

To achieve this goal, we develop several policies to plan and execute non-prehensile motions with a robot. For topple actions, we develop a model to robustly estimate the transition distribution under uncertainty in contact point position, friction coefficients, and push direction. The topple policies use this model to plan future actions and compute an expected reward for the MDP.

Chapter 3

Linear Pushing

3.1 Overview

When objects in clutter cannot be grasped, particularly because of environmental collisions, pushing can be used to separate the objects and expose robust grasps. In this chapter, we will describe several pushing policies, and explore the situations in which pushing is effective in exposing robust grasps. We compare the performance of each policy against a random pushing baseline on 1000 heaps in which initial grasp confidence is low.

In the work presented in this chapter, I helped implement half of the pushing policies and helped write scripts to execute each policy on object heaps and analyze the results. My co-authors, Michael Danielczuk and Jeffrey Mahler, implemented the other half of the policies, integrated the policies in the pyBullet simulation and physical robot, and also wrote scripts to analyze results.

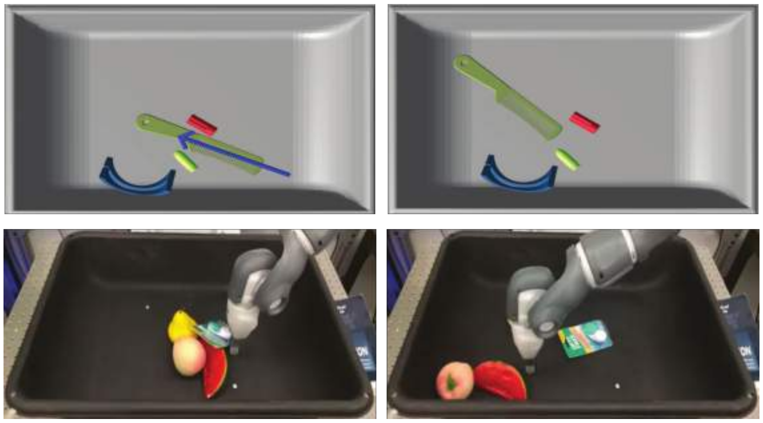

Figure 3.1: Before (left) and after (right) images of successful pushes in simulation (top) and in physical experiments with the ABB YuMi (bottom).

3.1.1 Pushing Assumptions

1. Object distances are approximated using the object center of mass for purposes of finding free regions in the bin and boundaries between objects.

2. Object center of mass in physical experiments is approximated as the centroid of each object's points from a point cloud.

3.2 Pushing Policies

3.2.1 Quasi-Random Policy

The Quasi-Random Policy generates a linear push action using the following three steps:

1. Choose one object in the heap at random,

2. Choose a direction at random, and

3. Push for a fixed length at the center of mass toward the chosen object in the chosen direction.

The push action is clipped to the bounds of the bin so that the gripper will not collide when executing the action.

3.2.2 Boundary Shear Policy

The boundary shear policy is adapted from the pushing policy introduced in Hermans et al. in [10]. It aims to separate the two closest objects in the heap by pushing one of them along the boundary between the two objects.

1. Find the two closest objects in the heap with centers of mass c_i and c_j,

2. Construct a line $\overline{c_ic_j}$ connecting the centers of mass of the two closest objects projected to the plane of the bin bottom, and a line $\overline{c_ic_j}_\perp$ perpendicular to $\overline{c_ic_j}$ that defines the vector approximating the boundary of the two objects,

3. Generate four possible push vectors, two for each object, that extend through the centers of mass of the objects in the direction $\overline{c_ic_j}_\perp$, and

4. Choose the push direction which is closest to the direction of free space and is collision free.

3.2.3 Free Space Policy

The free space policy aims to separate the two objects in the heap with closest centers of mass by pushing one of them along a direction toward the most free space, taking into account bin walls and other objects. It generates the push action using the following steps:

1. Find the two objects in the heap with closest centers of mass c_i and c_j,
2. For each object, find the free space point p_i defined above,
3. Draw lines $\overline{c_i p_i}, \overline{c_j p_j}$ from each of the centers of mass of the two closest objects to the points p_1 and p_2, respectively, with each point projected to the plane of the bottom of the bin,
4. Generate two possible push vectors, one for each object, that extend through the centers of mass of the objects in the collision-free directions closest to $\overline{c_i p_i}$ and $\overline{c_j p_j}$, and
5. Choose from the two possible collision-free push actions based on the minimum distance from the current center of mass of object i to p_i.

3.2.4 Maximum Clearance Ratio Policy

The maximum clearance policy, defined by Chang, Smith, and Fox [5], analyzes the available space for an object to be pushed into and the cluttered area it is being pushed from.

1. Calculate clearance in front of and behind each object for 16 uniform directions spanning angles between 0 and 2π by moving the objects footprint in the given direction and checking for collisions with other objects or the bin, and
2. Choose push action that maximizes ratio of space in the forward direction to space in the backward direction and is collision free.

3.2.5 Cluster Diffusion Policy

The cluster diffusion policy groups objects into clusters based on their position. It considers pushes of objects away from their corresponding cluster centers, along the vector originating from the cluster center to the object center of mass.

1. Separate objects into clusters of one to three objects and find the centroid of each cluster m_i,
2. Define pushing vectors $\overline{m_i c_i}$ that connect center of cluster to center of mass c_i of each object in its cluster, and

3. Score each of the potential push actions as their cosine similarity with the direction of most free space for the given object, and execute the push action with the highest score.

3.3 Pushing Experiments

3.3.1 Simulated Experiments

We generate heaps of 3D object meshes in a bin in which pushing has the potential to be helpful. To do this, we choose meshes from the Thingiverse, and the KIT and 3DNet datasets. We initialize simulations by sampling over distributions of heap size, 3D object models, camera pose, and friction coefficients to get an initial state $\mathbf{x}_0$. We randomly drop objects into the bin, and repeatedly execute parallel jaw and suction grasps until the bin is cleared or the grasping policy described in [23, 24, 25] fails n times in a row or has confidence below a threshold. If the bin is not cleared, we record the heap at the state in which the grasping policy fails. We then roll out each push policy on this set of heaps, and measure the performance of each policy using the metrics in Section 2.3.

With the dataset of over 1000 pushing scenarios collected, each of the policies described in Section 3.2 were rolled out on the set of heaps, using pybullet [7] for simulating gripper-object and object-object interactions in the bin, and the metrics in Section 2.3 were measured for each pushing action. We reject heaps where none of the policies (including the quasi-random policy), were able to expose a parallel jaw or vacuum suction grasp whose predicted reliability was higher than the that of the original state of the heap. This could be due either to object geometry or extreme cases of object positioning in the bin (e.g. object lying directly in a corner of the bin). These heaps reduce the performance across all policies, and rejection sampling allows us to focus on cases that highlight differences between the pushing policies. The remaining 481 pushing scenarios, termed *improvable heaps* were used to compare the pushing policies to the baseline policy.

We record the Suction Grasp Confidence Gain and Parallel Jaw Confidence Gain for each pushing action. To analyze the differences between policies statistically, we ran robust linear regressions over 1907 observations for each metric, controlling for differences between heaps. The results showed that the free space and boundary shear policies are statistically different from the baseline in the Parallel Jaw Confidence Gain metric ($p < 0.001$), but not the Suction Grasp Confidence Gain metric.

We examined many individual heaps to understand the magnitude and variance of the policies' impact. Figure 3.3 shows an example where each policy outperformed the baseline. In Figure 3.3, we can see that the non-baseline policies choose to push one of the two objects that overlap, and they all achieve a large increase in the parallel jaw metric by uncovering the red object initially lying underneath another object. The Boundary Shear and Free Space policies perform especially well, separating all of the objects. Note that the object does not need to be completely uncovered for the grasp to be available. This reflects the difference

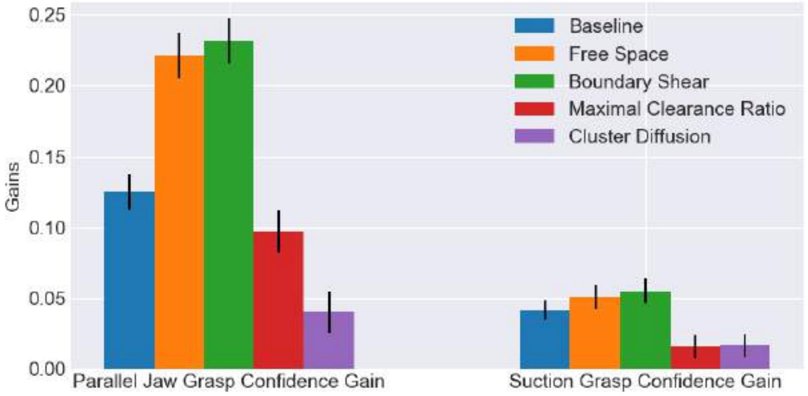

Figure 3.2: Means and standard errors of the mean for each policy and each type of end effector. These results suggest pushing has a larger effect on the parallel jaws. We speculate that this effect occurs due to suction grasps relying on faces of objects being available, and are thus less likely to be affected by pushing, whereas parallel jaw grasps are heavily affected by space around the object.

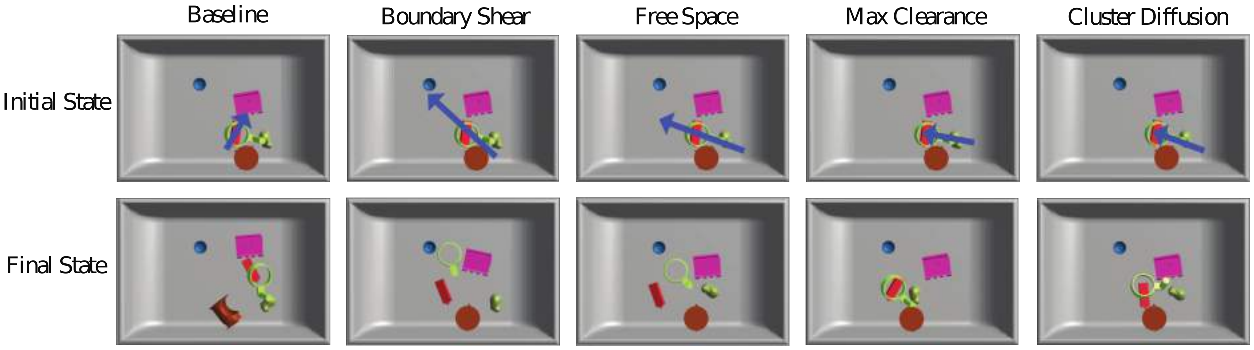

Figure 3.3: For this heap, each policy outperforms the baseline. The initial state with the planned action (top row) and final state after executing the planned action (bottom row) are shown for each policy. The blue arrow represents the planned push and the the initial gripper position is represented by the tail of the arrow, while the final position is represented by the head.

between measuring grasping metrics and object separations because in this case, the objects are still touching but a parallel jaw grasp becomes available.

3.3.2 Physical Experiments

We planned pushes for bin picking on an ABB YuMi using the Boundary Shear policy, the best performing policy in simulation, over 35 heaps of objects with varying geometry

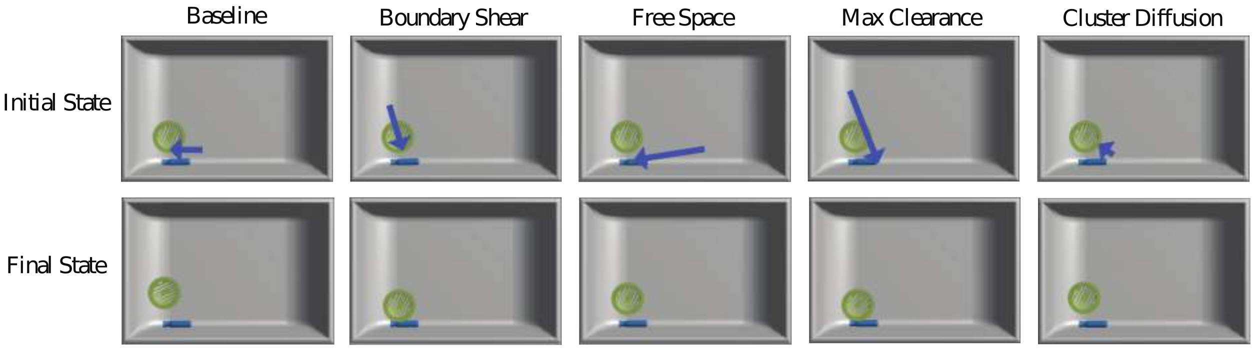

Figure 3.4: The baseline marginally outperforms all the other policies. The initial state with the planned action (top row) and final state after executing the planned action (bottom row) are shown for each policy. The blue arrow represents the planned push and the the initial gripper position is represented by the tail of the arrow, while the final position is represented by the head.

such as tools, toys, produce, and industrial parts. Each heap contained between two and ten objects in configurations with few accessible grasps, such as two bottles side-by-side. For each heap, the robot acquired a point cloud of the bin with a Photoneo PhoXi depth sensor, segmented the point cloud using Euclidean Cluster Extraction implemented in the Point Cloud Library [32], and planned a push using point clusters as objects and the cluster centroid as an estimate of the center of mass of each object. The robot then executed the push by closing the parallel jaw gripper and following the linear push trajectory. Each push took approximately 1.0 seconds to plan.

For each push, we measured the Overall Grasp Confidence Gain of parallel-jaw and suction grasps planned by a Grasp Quality Neural Network policy [24, 25]. We categorized performance based on the grasp confidence for the best grasp in the initial heap (pre-pushing). Heaps with $Q_o < 0.25$ had an Overall Grasp Quality Gain of 0.24 ± 0.07, while heaps with $Q_o < 0.5$ had an Overall Grasp Quality Gain of 0.12 ± 0.06.

3.4 Failure Modes and Limitations

In Figure 3.4, we can see that the non-baseline policies fail to find a collision-free push that can move one of the objects away from the corner of the bin. The baseline policy's action is clipped so that it does not collide with the bin, and results in it slightly increasing the parallel jaw grasp confidence by nudging the green object further from the other object. The non-baseline policies have no effect on the grasp confidence metrics. This figure illustrates one of the current failure modes with the pushing policies that we have implemented. By taking a conservative approach and avoiding collisions at all costs, we are sometimes unable to plan a push that moves the objects away from the bin edges.

In addition, the results in Figure 3.2 imply that pushing affects the parallel jaw grasps more than it affects suction grasps. Pushing actions typically move objects around the bin, but rarely topple them onto a new face or side. Suction relies on sampling grasps on the top faces of the objects; if the face does not change, then it is unlikely that the suction grasp confidence will change significantly. However, for the parallel jaws, grasp confidence depends strongly on available space around the object. Thus, pushing an object to a more free location can shift the parallel jaw grasp confidence more dramatically.

Chapter 4

Toppling

4.1 Overview

The experiments in Chapter 3 demonstrated that linear pushing was able to effectively expose areas on antipodal surfaces for parallel jaw grippers, but was not as effectively able to expose flat surfaces for suction grippers. In this chapter, we explore how toppling can be used to expose robust vacuum suction grasps. Since prior models predict the toppling behavior of extruded two dimensional shapes, we present a toppling model for three dimensional objects, and present results which suggest that this model can be used to expose robust vacuum suction grasps. Physical experiments suggest that the presented model outperforms the toppling baseline by an absolute 26.9% when comparing the total variation of each predicted distribution with the empirical distribution. We present a toppling policy which uses this model to plan topple actions and demonstrate that this policy can increase suction grasp reliability by an absolute 28.4% over a Max-Height Baseline.

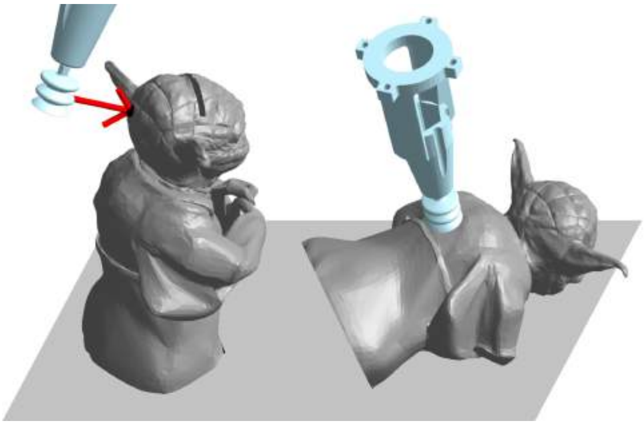

Figure 4.1: Toppling the "Yoda Piggybank" object exposes access to a robust vacuum suction grasp.

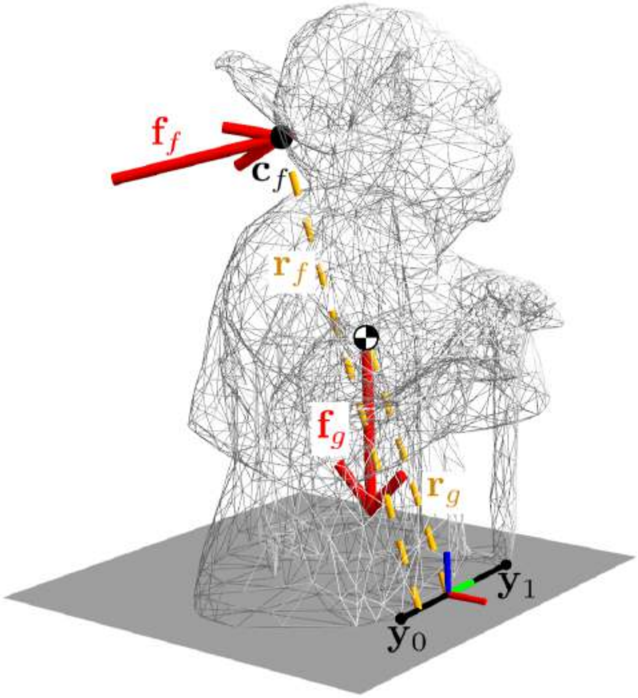

Figure 4.2: Illustration of the forces acting on the object. The point pusher produces a wrench according to the force $\mathbf{f}_f$ and the moment arm $\mathbf{r}_f$, defined between the contact point and its projection on the edge. Gravity produces a wrench according to the gravitational force $\mathbf{f}_g$ and the moment arm $\mathbf{r}_g$, defined between the center of mass and its projection onto the edge.

4.1.1 Definitions

We formally describe the conditions for toppling with the the following quantities, as shown in Figure 4.2 and 4.3.

1. **Toppling Edge**: $\mathbf{s}_n = n\mathbf{y}_0 + (1-n)\mathbf{y}_1,\ \forall n \in [0,1]$, the parametrized line representing the toppling edge.

2. **Instantaneous Velocity Vectors**: $\hat{\mathbf{v}}(\mathbf{s}_n)$, the unit instantaneous velocity vector at each point along the toppling edge of the object.

3. **Pressure Distribution**: $p(\mathbf{s}_n)$, the pressure distribution of the object along the toppling edge, determined by the distribution of the object's mass and geometry along the toppling edge.

4. **Contact Point**: $\mathbf{c}_f$, the point where the point pusher contacts the object.

5. **Forces**: $\mathbf{f}_f$ and $\mathbf{f}_g$, the forces applied to the object by the point pusher and gravity, respectively.

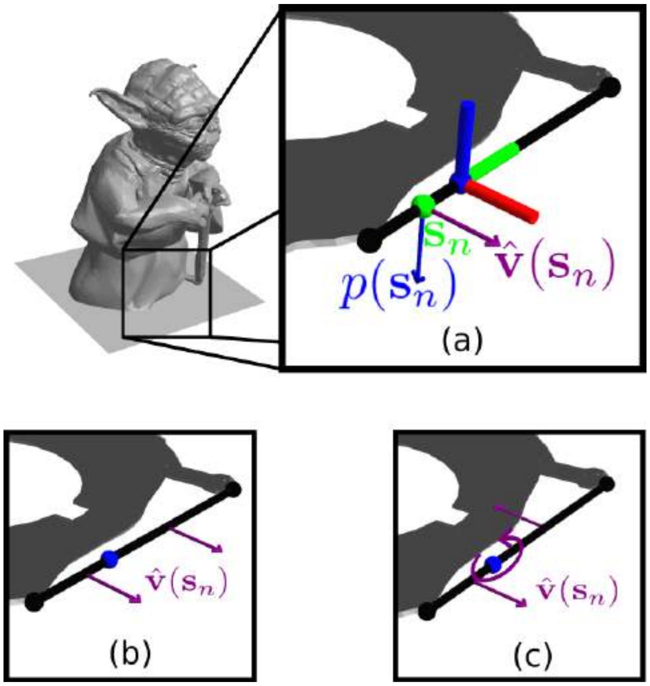
$\mathbf{s}_n$
$\hat{\mathbf{v}}(\mathbf{s}_n)$
$p(\mathbf{s}_n)$
(a)
$\hat{\mathbf{v}}(\mathbf{s}_n)$
(b)
$\hat{\mathbf{v}}(\mathbf{s}_n)$
(c)

Figure 4.3: (a) The topple action produces unit velocity vectors $\hat{\mathbf{v}}(\mathbf{s}_n)$ across the toppling edge, which the object-workspace interaction resists via its pressure distribution $p(\mathbf{s}_n)$. The coordinate frame origin is at the projection of the center of mass onto the toppling edge, and the y-axis is parallel to the toppling edge. (b) and (c) demonstrate the dichotomy between the maximal $\mathbf{f}_t$ (b) and τ_z (c) the object can resist, depending on the direction of $\hat{\mathbf{v}}(\mathbf{s}_n)$.

6. **Friction Coefficients**: μ_f and μ_T, the coefficients of friction of the object-point pusher and the object-workspace contacts, respectively.

7. **Friction Wrench**: τ_z and $\mathbf{f}_t$, the torques around the z-axis and frictional force that the object-workspace contact exerts on the object, illustrated in Figure 4.3.

8. **Moment Arms**: $\mathbf{r}_f$ and $\mathbf{r}_g$, the moment arms between the push contact point and the toppling edge, and the center of mass and the toppling edge, respectively.

4.1.2 Toppling Assumptions

1. Known object geometry (defined by a two-manifold triangular mesh) and center of mass.

2. The object rests in a stable pose [9] on a planar surface.

3. The object has a constant friction coefficient across the toppling edge and at each contact point. The choice of friction coefficient is described in Section 4.4.2.

4. The object's pressure distribution is modelled as two point contacts on the endpoints of the toppling edge.

5. The object is singulated and rests in a stable pose on a planar workspace.

4.2 Toppling Model

We develop a quasi-static models to estimate the topple reliability for a given contact point and push direction. We define the *base* of the object to be the convex hull of the object's vertices touching the workspace. For each edge on the base, we check the following three conditions, to determine if $\mathbf{u}_t$ will topple the object.

1. **Contact Slip:** The point pusher does not slip on the object.

2. **Workspace Slip:** The contact between the toppling edge and the workspace can resist the wrench applied by the point pusher.

3. **Minimum Push Force:** The wrench applied by the point pusher has a magnitude large enough to rotate the object over the given edge.

If multiple edges satisfy these conditions, the model predicts the object will topple over the edge which requires the least force applied by the point pusher. We then use quasi-static analysis to predict the pose of the object after it topples over the predicted edge [9].

The model presented in this section expands on the model by Lynch *et al.* [19] by using the Friction Limit Surface [13] to predict planar rotations and by using Monte Carlo sampling to predict topple reliability under uncertainty.

4.2.1 Condition 1: Contact Slip

If the point pusher slips on the contact point, the proposed quasi-static model predicts it will not be able to apply the required force to topple the object. The point pusher will not slip on the object as long as the it presses on the object in a direction within the object's friction cone at the point of contact.

$$\begin{bmatrix}(\mathbf{f}_f)_{x'} & (\mathbf{f}_f)_{y'} & (\mathbf{f}_f)_{z'}\end{bmatrix}^T = T_c\mathbf{f}_f$$
$$\sqrt{(\mathbf{f}_f)_{x'}^2 + (\mathbf{f}_f)_{y'}^2} \leq \mu_f(\mathbf{f}_f)_{z'}$$

$T_c \in SE(3)$ is the rigid transformation from the world coordinate system to the coordinate system defined at the contact point $\mathbf{c}_f$. $(\mathbf{f}_f)_{x'}, (\mathbf{f}_f)_{y'}, (\mathbf{f}_f)_{z'}$ are the x, y, and z components of $\mathbf{f}_f$ relative to a coordinate frame defined at the contact point.

4.2.2 Condition 2: Workspace Slip

If the object slips either tangentially on the workspace or rotationally along an axis perpendicular to the workspace, the force applied at the contact point will result in the object moving in the plane instead of toppling. To determine whether a predicted push will topple the object or cause the object to slip on the workspace, we apply the friction limit surface model described in [13]. The friction limit surface describes the set of wrenches the object-workspace contact can resist before the object slips on the workspace. If the wrench applied by the point pusher is within this set of resistable wrenches, the object will topple.

We can integrate the infinitesimal forces applied by the object along the toppling edge to find the tangential force $\mathbf{f}_t$ and torque around the z-axis τ_z which the object-workspace contact can resist due to friction:

$$\begin{aligned}\mathbf{f}_t &= -\int_{n=0}^{1} \mu_T \hat{\mathbf{v}}(\mathbf{s}_n) p(\mathbf{s}_n) dn \\ \tau_z &= \int_{n=0}^{1} \mu_T ||\mathbf{s}_n \times \hat{\mathbf{v}}(\mathbf{s}_n)||_2 p(\mathbf{s}_n) dn\end{aligned}$$

The tangential force that the object-workspace contact can resist is maximized when $\hat{\mathbf{v}}(\mathbf{s}_n)$ is constant across the toppling edge (Figure 4.3b), and the torque around the z-axis is maximized when the instantaneous velocities all produce torques in the same direction around the z-axis (Figure 4.3c), where $F_n = mg$.

$$\begin{aligned}||\mathbf{f}_t||_2 &\leq \mu_T F_n \\ \tau_z &\leq \int_{n=0}^{1} \mu_T ||\mathbf{s}_n||_2 p(\mathbf{s}_n) dn \\ &\leq (\tau_z)_{max}\end{aligned}$$

For our experiments, we assume the object's pressure distribution is defined as two point masses on the endpoints of the toppling edge, though other distributions can be substituted. In this case, $(\tau_z)_{max} = \frac{\mu_T F_n}{2}[\mathbf{y}_1 + \mathbf{y}_0]$.

We approximate the set of wrenches the friction contact can exert (the Friction Limit Surface) with an ellipse in wrench space [13], defined by the maximum tangential force and rotational torque. The friction limit surface in wrench space is visualized in Figure 4.4.

$$\frac{||\mathbf{f}_t||_2^2}{(\mu_T F_n)^2} + \frac{\tau_z^2}{(\tau_z)_{max}^2} \leq 1$$

4.2.3 Condition 3: Minimum Push Force

The applied force produces a torque around the toppling edge. If the magnitude of this torque is less than the magnitude of the torque caused by gravity, then the object will not topple. An object such as the one in Figure 4.2 will rotate at the toppling edge if $||\mathbf{r}_f \times \mathbf{f}_f'||_2 \geq ||\mathbf{r}_g \times \mathbf{f}_g||_2$, where $\mathbf{f}_f'$ is the component of $\mathbf{f}_f$ orthogonal to the toppling edge.

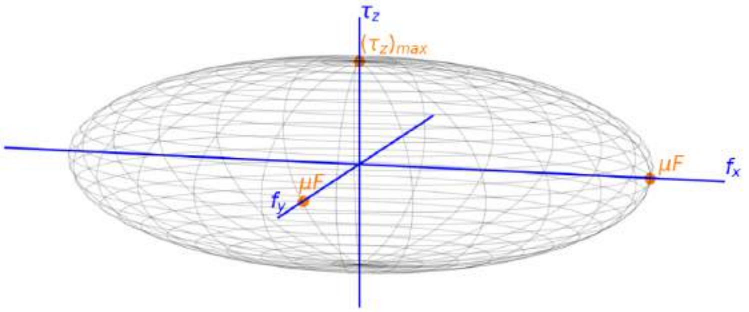

Figure 4.4: Approximation of the Friction Limit Surface (FLS) as an ellipse in wrench space. If a linear trajectory produces a wrench on the object within the FLS, the object will topple and not slip on the workspace.

4.2.4 Predicting Toppling Final Resting Pose

If the conditions in Sections 4.2.1, 4.2.2, and 4.2.3 are satisfied, we predict the object's pose after toppling by rotating the object around the toppling edge until either:

1. The center of mass passes over the toppling edge (i.e. the apex of the topple). After this point, the object is free to fall unaided. We use the analysis from Goldberg *et al.* [9] to predict which face the object settles on after it falls.

2. The topple is blocked before it reaches the apex of the topple (i.e. some point other than the toppling edge touches the workspace). In this case, we return the object to its original pose.

All trials which fail to topple the object over any edge are mapped to the starting pose. Figure 4.5 shows the predicted poses of an object when toppled over three edges from various candidate points.

4.2.5 Toppling Reliability

To account for uncertainty in pose and object geometric and material properties, we introduce a robust metric for computing toppling probabilities. We consider 100 topple trials for each candidate contact point, with Gaussian noise added to the contact position, push direction, and friction coefficients: $\mu_T \sim \mathcal{N}(0.43, 0.1)$, $\mu_f \sim \mathcal{N}(0.98, 0.2)$ $\mathbf{c}_f \sim \mathcal{N}(\mathbf{c}_f, 5.05 \cdot 10^{-4} \cdot I)$, $\mathbf{f}_f \sim \mathcal{N}(\mathbf{f}_f, 0.055 \cdot I)$, and $R_\theta \sim \mathcal{N}(0.0, 0.22)$. Section 4.4.2 describes how we choose these noise distributions. These trials serve as Monte-Carlo estimates for the true underlying toppling distribution. Figure 4.5 illustrates these Monte-Carlo estimates for each sampled contact point on the object, as well as the predicted final pose of the object if toppled at each contact point.

We sum the probabilities of toppling over edges which result in the same final pose. We use these probabilities to estimate the transition distribution $\mathbb{P}[\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t]$, defined in

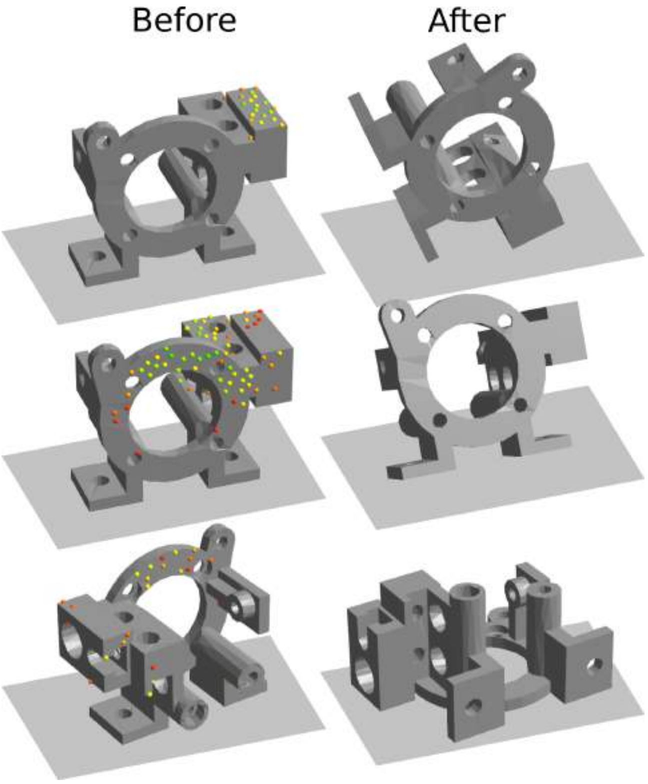

Figure 4.5: Predicted reliability of each toppling action. The color of each contact point in the left column corresponds to the model's predicted probability that the object will reach the pose in the right column when pushed normal to the surface at each contact point. Contact points with 0% predicted reliability are omitted.

Section 2.2, where $\mathbf{x}_t$ is the current state, $\mathbf{u}_t$ is the topple action, and $\mathbf{x}_{t+1}$ is the predicted final pose of the object. The probability that the object topples to a different pose is $\mathbb{P}[\mathbf{x}_{t+1} \neq \mathbf{x}_t | \mathbf{x}_t, \mathbf{u}_t]$.

4.2.6 Toppling Model Baselines

The presented model expands on prior work by predicting planar rotations, and predicting the toppling reliability under uncertainty. To evaluate the utility of these additions, we compare the model against three baselines without these additions. To predict toppling without modelling planar rotations, we assume $\tau_z = 0$. In this case, Condition 2 reduces to:

$$||\mathbf{f}_t||_2^2 \leq (\mu_T F_n)^2$$

To predict toppling without robust estimates, for each candidate topple point, we consider only one trial, with no noise added.

Model	Predicts Rotations	Robust Estimates
Baseline		
Baseline + Rotations	✓	
Baseline + Robustness		✓
Robust Model	✓	✓

Table 4.1: Robust Model and all considered baselines to evaluate performance against physical experiments.

4.3 Toppling Policies

The goal of the policies in this section is to pick a toppling action to transition 3D objects into poses with improved grasp accessibility. We use the probabilities from the Toppling Model in Section 4.2 to predict the pose of objects after each topple action $\mathbf{u}_t$ and use the grasp analysis from Mahler *et al.* [25] to predict suction grasp accessibility. We compare the proposed policy against Max-Height and Greedy Baselines.

4.3.1 Max-Height Baseline

To evaluate the utility of modelling the toppling behavior of objects in attempting to increase suction grasp access, we develop a baseline with no knowledge of the toppling transition probabilities. However, the policy does have knowledge of the predicted grasp accessibility of the object in each pose. This baseline picks the highest possible point on the object's surface with a surface normal within 15° of the workspace and pushes at this point. The Max-Height policy executes "no-ops" if the object is in the pose with the highest reliability of accessible grasps.

4.3.2 Greedy Baseline

The Greedy Baseline uses the proposed toppling model in this chapter to compute the topple probabilities for only the current time-step and picks the action which maximizes the expected suction grasp reliability $V_s(\mathbf{x}_{t+1})$ immediately after the topple action:

$$\pi(\mathbf{x}_t) = \underset{\mathbf{u}_t}{\operatorname{argmax}} \left(\mathbb{E}_{\mathbf{x}_{t+1} \sim \mathbb{P}[\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t]} \left[V_s(\mathbf{x}_{t+1}) \right] \right)$$

The policy executes "no-ops" if $V_s(\mathbf{x}_{t+1}) \leq V_s(\mathbf{x}_t)\ \forall \mathbf{x}_{t+1} \in \mathbb{P}[\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t]$. The Greedy Baseline benefits from fast computation times because it only computes the actions for the current time-step, but does not account for sub-optimal topples that could allow for better future topple actions.

4.3.3 Value Iteration Policy

The Value Iteration Toppling Policy considers sub-optimal poses by assigning a value to each pose defined as the maximum suction grasp reliability for any pose reachable with linear topple actions, according to the transition probabilities generated by the model in Section 4.2.5.

The policy generates a graph of the toppling MDP, such as the one in Figure 4.6, where the nodes represent the object's stable poses, and the edges represent topple actions. The policy assigns each action a value $Q_T(\mathbf{x}_t, \mathbf{u}_t)$, based on the discounted suction grasp reliability of future object states following action $\mathbf{u}_t$, using Value Iteration [33]:

$$Q_T(\mathbf{x}_t, \mathbf{u}_t) = \mathbb{E}_{\mathbf{x}_{t+1} \sim \mathbb{P}[\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t]} \left[V_T(\mathbf{x}_{t+1}) \right]$$
$$V_T(\mathbf{x}_t) = \max \left(V_s(\mathbf{x}_t), \gamma \max_{\mathbf{u}_t} Q_T(\mathbf{x}_t, \mathbf{u}_t) \right)$$

We choose $\gamma = 0.95$ as our discount factor. After computing Value Iteration, the policy executes the action with the highest q-value $Q_T(\mathbf{x}_t, \mathbf{u}_t)$, or "no-ops" if $V_T(\mathbf{x}_{t+1}) \leq V_T(\mathbf{x}_t)\ \forall \mathbf{x}_{t+1} \in \mathbb{P}[\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t]$.

4.4 Experiments

4.4.1 Dataset Generation

Our goal was to predict the toppling behavior of objects with dense samples of contact points across the object's surface and with a high number of Monte-Carlo trials per sampled contact point. Since this can be computationally expensive, we pre-computed the topple probabilities and final poses for every stable pose of the desired objects, and store these values in a database. The pre-computation allowed us to quickly retrieve the toppling probabilities for a wide variety of objects.

We generated a dataset of 1000 candidate topple points on 189 objects in 1257 stable poses, for a total of 1,257,000 candidate topple points. We chose objects which satisfy the following conditions: 1) The object has a stable pose with high grasp reliability ($> 50\%$) and a stable pose with low grasp reliability ($< 25\%$) and 2) the object's center of mass is higher in the high grasp reliability pose than in the low grasp reliability pose. We chose a complex set of objects whose convex hulls had an average of 1437 faces. The average time to compute the toppling probabilities and final poses for 1000 candidate topple points with 100 samples per point is 40.47 seconds per object stable pose on a desktop computer running Ubuntu 16.04 with a 3.6 GHz Intel Core i7-6850k CPU.

4.4.2 Toppling Model Experiments

To compare the predictions of the model and the baselines in Section 4.2.6 with empirical outcomes, we ran physical experiments of topple actions on the seven 3D printed objects

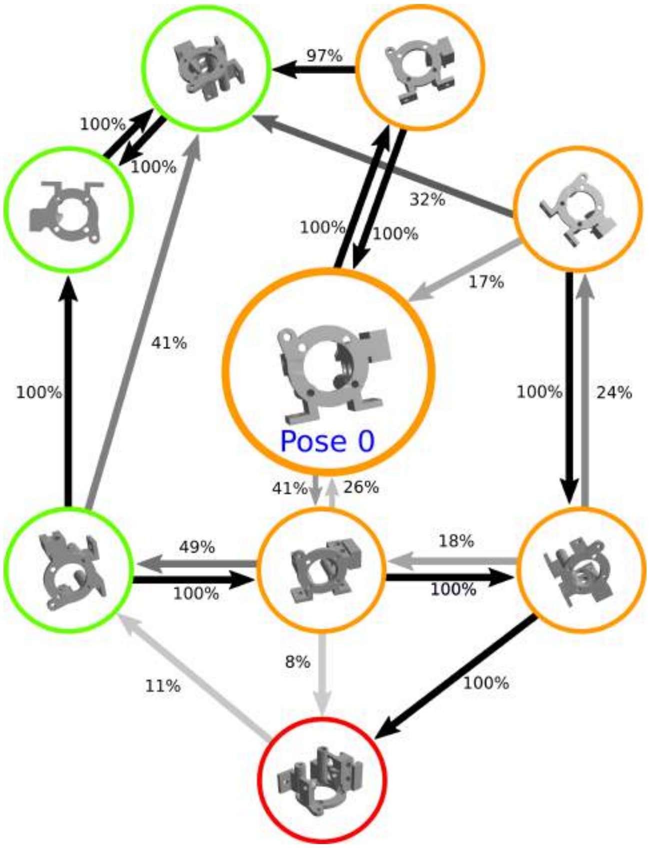

Figure 4.6: Toppling Graph: The object in Figure 4.5 starts at Pose 0. All poses that are attainable via linear topple actions are shown. The border color corresponds to the reliability of the best available suction grasp of the object in the pose, and the edges are labelled with the probability of success for the best linear topple action.

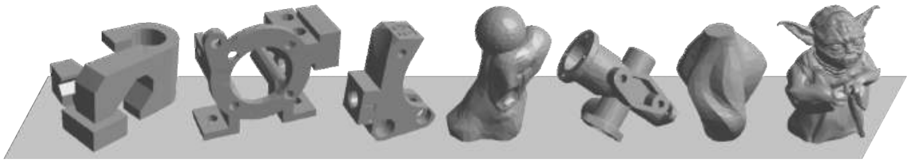

Figure 4.7: The objects used in physical experiments.

shown in Figure 4.7. We executed 10 topple actions per object and repeated each topple action 10 times.

We placed the 3D printed objects in front of an ABB YuMi robot on a planar workspace. The robot acquired a 3D point cloud of the mesh using a Photoneo PhoXi depth sensor, and used the Super4PCS algorithm [27] to match the pose of the object in simulation to the pose of the object in front of the robot.

We then chose a topple action and recorded the pose of the object before and after the

Topple Predictions	
Model	mAP
Baseline	0.741
Baseline + Rotations	0.752
Baseline + Robustness	0.852
Robust Model	**0.848**

Table 4.2: Mean Average Precision (mAP) for each model's prediction of toppling into any new pose, based on the empirical data collected. We perform k-fold cross validation, and average the mAP of each of the held-out folds.

action. Since the predicted topple reliability is 0 at most contact points, we sampled actions with probabilities proportional to the topple probabilities from Section 4.2.5, where $\epsilon = 10^{-4}$ to get topple actions with a broader range of reliabilities.

$$\mathbb{P}[\mathbf{u}_t] = \frac{\mathbb{P}[\mathbf{x}_{t+1} \neq \mathbf{x}_t | \mathbf{x}_t, \mathbf{u}_t] + \epsilon}{\sum_{\mathbf{u}'_t} \mathbb{P}[\mathbf{u}'_t]}$$

Predicting Whether the Object Will Topple

For each model and baseline, we considered 100 sets of noise distributions to perturb the contact point position, friction coefficients, and push direction. We performed 6-fold Cross Validation, each with a different object held out, so that the model comparison is not dependent on the choice of noise distributions. We averaged the performance on the held out object for each fold.

Table 4.2 shows how well each model and baseline was able to predict whether a given push will topple the object into a pose other than the start pose, when compared against empirical data.

To further analyze the effect of each parameter on the model's performance, we created a t-SNE visualization [22], shown in Figure 4.8. The models which predict rotations and topple robustness not only have a larger maximum mAP, but these models seem to be less sensitive to the choice of parameters. This suggests the proposed robust model can generalize better to new objects.

Predicting the Object Pose Distribution

We also computed an empirical distribution of final poses and estimated distributions of final poses from the proposed model and the baseline models. We computed the total variation distance between each model's predictions and the empirical distribution to quantify the accuracy of the proposed model's predicted toppling behavior. We averaged the total variation distance across every state and action:

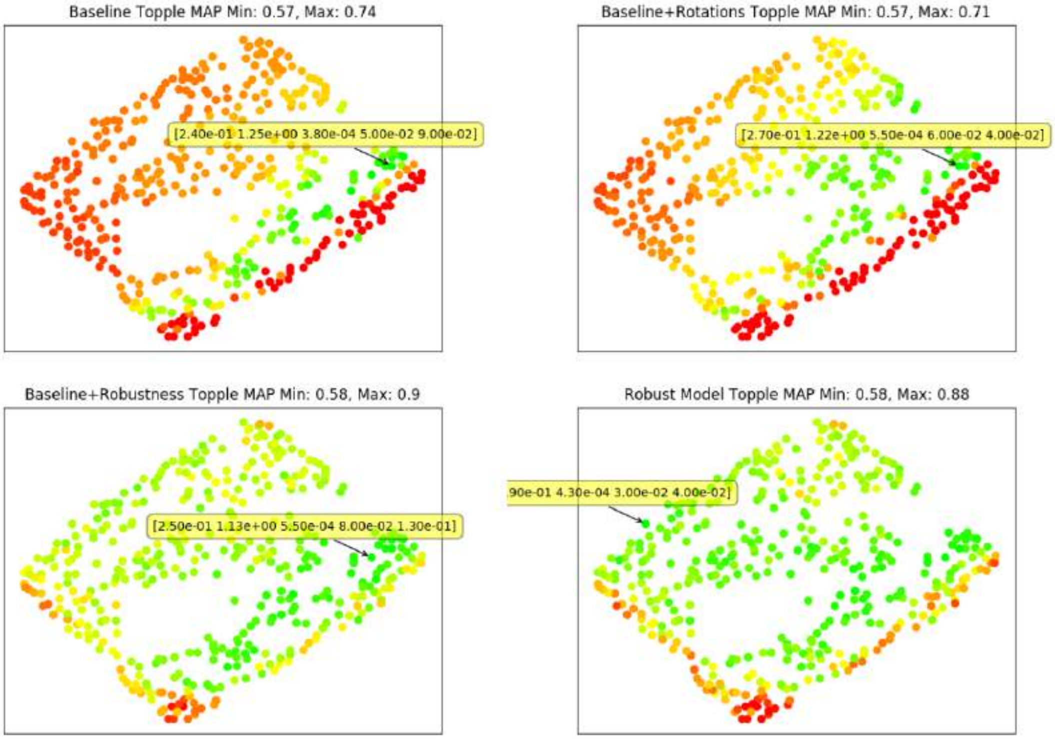

Figure 4.8: Each point in the t-SNE visualization represents a model with a different choice of model parameters. Green points represent models with higher mAP, red points represent models with lower mAP. Each sub-graph is normalized to show the difference in parameter space. An example model is labelled in each sub-graph. The numbers represent the mean of μ_T and μ_f, and the variance of $\mathbf{c}_f$, $\mathbf{f}_f$, and R_θ.

$$\text{TV} = \frac{1}{|\mathcal{X}||\mathcal{U}|} \sum_{\substack{\mathbf{x}_t \in \mathcal{X} \\ \mathbf{u}_t \in \mathcal{U}}} \sup_{\mathbf{x}_{t+1} \in \bar{\mathcal{X}}} \left| \hat{\mathbb{P}}[\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t] - \mathbb{P}[\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t] \right|$$

In the physical experiments, $\mathcal{X}$ represents every initial state, $\mathcal{U}$ represents the actions executed, and $\bar{\mathcal{X}}$ represents the set of 10 final poses the object reaches.

Again, we consider 100 models, each with different noise distribution parameters and performed 6-fold Cross Validation. Table 4.3 shows both the mean Average Precision and Total Variation distance averaged over the held out sets for the best model in each of the six folds.

In Figure 4.9, each model in the t-SNE plot is colored corresponding to the Total Variation between it's predicted final pose distribution and empirical distribution. Similar to the Topple Predictions, the models which predict rotations and topple robustness have a lower Total Variation, but also are less sensitive to the choice of model parameters.

Final Pose Predicitions		
Model	TV	mAP
Baseline	0.424	0.412
Baseline + Rotations	0.494	0.381
Baseline + Robustness	0.247	0.568
Robust Model	**0.211**	**0.589**

Table 4.3: Total variation distance (TV) and mean Average Precision (mAP) between each model and empirical toppling distributions. We perform 6-fold cross validation, and average the total variation and mAP of each of the held out folds. Model distributions with low total variation distance are consistent with the empirical distribution.

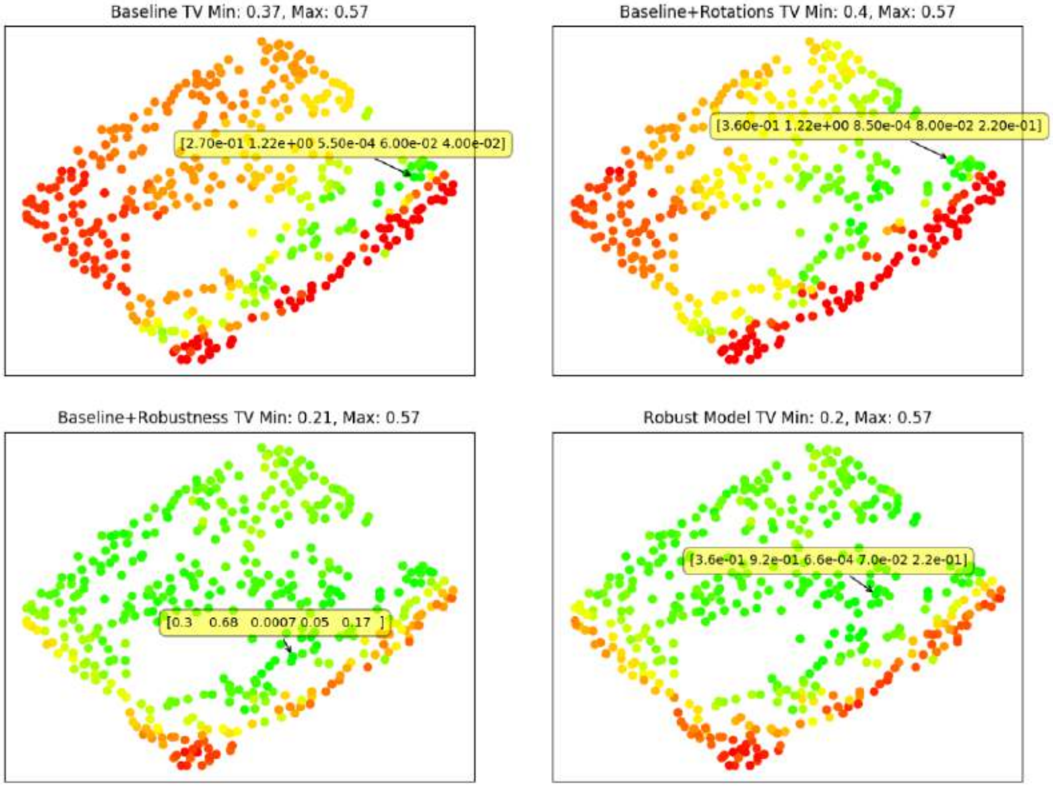

Figure 4.9: Each point in the t-SNE visualization represents a model with a different choice of model parameters. Green points represent models with higher TV, red points represent models with lower TV. Each sub-graph is normalized to show the difference in parameter space. An example model is labelled in each sub-graph. The numbers represent the mean of μ_T and μ_f, and the variance of $\mathbf{c}_f$, $\mathbf{f}_f$, and R_θ. Model distributions with low total variation distance are consistent with the empirical distribution.

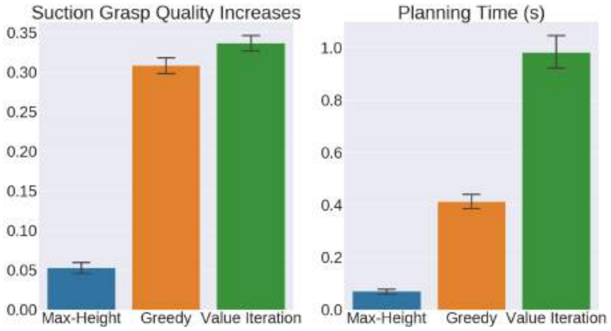

Figure 4.10: Average difference in suction grasp reliability after each policy in Section 4.3 executes topple actions on a randomly-placed mesh in 3654 simulations. The planning times are the average planning time per action of each policy.

4.4.3 Toppling Policy Experiments

To test the policies' abilities to effectively use toppling to increase suction grasp accessibility, we ran all policies in a simulated environment and recorded the predicted reliability of accessible suction grasps before and after each policy finished execution. In each of the 20,000 trials, we dropped a random mesh into a simulated environment into a random pose, weighted according to its stable pose probability [9], used each policy to plan an action on the object, and used the Toppling Model to forward simulate the action. We drew a next state according to $\mathbb{P}[\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t]$.

Of these 20,000 trials, we found toppling to be useful in 3654 trials, i.e., the object was not already dropped in the pose with the highest suction grasp access and there existed a topple action which was predicted to increase suction grasp reliability. The performance of each policy in these useful trials is illustrated in Figure 4.10. These results suggest that the situations in which toppling is useful is limited, but that toppling can be effective in uncovering robust grasps.

While pushing the object near the top of the object can marginally improve suction grasp access, the Greedy Baseline and Value Iteration Policy are able to increase the reliability of suction grasps by an absolute 25.5% and 28.4%, respectively, over the Max-Height Baseline. While the the Value Iteration Policy achieves the highest increase in access, it comes at the cost of planning time, as it has to perform more database retrievals, and compute value iteration on the toppling graph at run-time.

Figure 4.6 exemplifies a situation in which the Value Iteration Policy outperforms the Greedy Baseline. The Greedy Baseline attempts a suction grasp at Pose 0, since the predicted grasp reliability of the object in all immediately reachable poses is the same as the predicted grasp reliability of the object in Pose 0. However, the Value Iteration Policy recognizes that toppling twice can lead to a state with 100% grasp reliability.

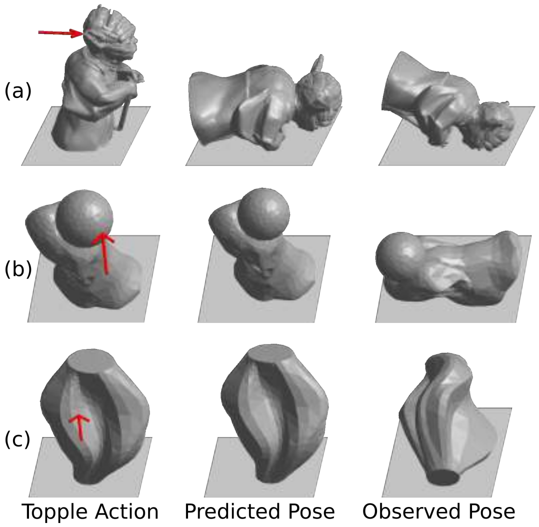

Figure 4.11: Failure modes from physical experiments. (a) Momentum causes the object to roll further than predicted (b) the model correctly predicts the point pusher will slip, but the object still topples as the point pusher slips (c) the model correctly predicts the object will slip on the workspace, but the object first rotates then topples.

4.5 Failure Modes and Limitations

To evaluate our model beyond our analysis, we examined examples in which the model significantly differs from physical outcomes. The three main limitations we identified with the proposed model are illustrated in Figure 4.11. In all three situations, the model is unable to predict the behavior because of the quasi-static assumption. In Figure 4.11a, the point pusher is able to topple the object, but momentum causes the object to roll past the predicted pose. The proposed model assumes quasi-statics during the rolling phase of the topple. In Figure 4.11b the topple action causes the object to initially slip on the workspace and then topples. The proposed model predicts a low topple reliability for this topple action, as it does not model dynamics. In a related scenario, the point pusher slips on the object surface in Figure 4.11c, but still topples the object. The model assumes the object will not topple if the point pusher slips on the object, so it incorrectly predicts this topple action fails.

Chapter 5

Discussion and Future Work

In this thesis, we explored several policies for pushing and toppling. Simulated and physical experiments suggest that pushing can be used when object grasps are occluded by environmental obstacles by separating target objects from other objects or bin walls. Pushing can be used to expose robust parallel jaw grasps, because parallel jaws require exposed antipodal faces, but is less useful in exposing robust vacuum suction grasps, because suction grippers require exposed flat surfaces at the top of the object. In contrast, toppling can be used when vacuum suction grasps are inaccessible because of rough local surface geometry, by transitioning target objects to new stable poses, exposing new flat surfaces. Simulated experiments suggest that toppling can be used to expose vacuum suction grasps. In order to plan toppling actions, we expand upon prior toppling models in this thesis, to predict toppling behavior of 3D polyhedral objects, predict planar rotations and translations, and predict toppling under uncertainty. Physical experiments comparing the proposed model and several baselines against physical outcomes suggest that the model outperforms the baseline model by an absolute 26.9% when using the Total Variation metric.

5.1 Pushing Future Work

In this work, we assume that if none of the five policies were able to prove grasp quality, then the heap is not improvable. Some heaps may be improvable by a policy not tested in this work. In the future, we will determine why some heaps are not able to be improved and seek a method for determining when heaps can be improved without testing several policies on them. For example, when objects are entangled, or cannot easily be pushed due to object pose or shape, we could attempt a different push or grasp action. Additionally, we made strong assumptions about the boundaries, geometries, and poses of the objects that were analyzed by representing them as points at their center of mass for finding free space in the bin. We seek to modify our simulations to calculate minimum distances between meshes more efficiently while still accounting for the entirety of the objects. We also will look to exploit quicker free space computation in image space as an alternative to our current object assumptions.

5.2 Toppling Future Work

In future work, we would explore toppling in clutter, where objects may be resting on other objects and not on a stable resting pose. In addition, the model will have to predict whether a topple action would result in the object toppling into a stable pose or rest onto a new object. We would also like to relax the model's assumptions, such as the approximation of the object's pressure distribution as two point masses at the endpoints of each edge. We also hope to explore more complex toppling policies such as multi-arm topple actions, such as the one presented by Yamashita *et al.* [34] and non-linear topple motions. In addition to evaluating the proposed model with physical experiments, we would like to execute topple actions in simulation and compare its performance to that of the proposed model. In certain situations, objects lie in flat stable poses, and no topple action is able to change the objects pose. We hope to design a manipulator that can flip objects over from the bottom using similar toppling analysis in order to reorient the object. Finally, we would like to explore pre-computing a dataset of the topple Q-Values, to decrease planning time for the Value Iteration Policy.

5.3 Shared Future Work

A composite policy which chooses a pushing or toppling action would be the most natural extension of the work presented in this thesis. This would involve extending toppling to clutter. When toppling in clutter, an action which fails to the object may still expose a robust grasp by clearing away environmental obstacles. Since the presented pushing policies don't predict the grasp reliability after the push, they cannot be directly compared to the toppling policies in a composite policy. Another extension would be to explore the use of non-prehensile motions in active perception [3]. Pushing and toppling are tools in order to accomplish higher level tasks, such as bin de-cluttering and mechanical search. In future work, we would like to explore how to use the policies explored in this thesis in these more complicated tasks. Finally, we would like to explore actions that are more complex than linear trajectories for pushing and toppling.

Bibliography

[1] Yasumichi Aiyama, Masayuki Inaba, and Hirochika Inoue. “Pivoting: A new method of graspless manipulation of object by robot fingers”. In: *Intelligent Robots and Systems’ 93, IROS’93. Proceedings of the 1993 IEEE/RSJ International Conference on.* Vol. 1. IEEE. 1993, pp. 136–143.

[2] Srinivas Akella and Matthew T Mason. “Posing polygonal objects in the plane by pushing”. In: *Int. Journal of Robotics Research (IJRR)* 17.1 (1998), pp. 70–88.

[3] Ruzena Bajcsy. “Active perception”. In: *Proceedings of the IEEE* 76.8 (1988), pp. 966–1005.

[4] Brian Carlisle et al. “A pivoting gripper for feeding industrial parts”. In: *ICRA*. 1994, pp. 1650–1655.

[5] Lillian Chang, Joshua R Smith, and Dieter Fox. “Interactive singulation of objects from a pile”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* IEEE. 2012, pp. 3875–3882.

[6] Akansel Cosgun et al. “Push planning for object placement on cluttered table surfaces”. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* IEEE. 2011, pp. 4627–4632.

[7] E Coumans, Y Bai, and J Hsu. *Pybullet physics simulator.* URL: `http://pybullet.org`.

[8] Andreas Eitel, Nico Hauff, and Wolfram Burgard. “Learning to Singulate Objects using a Push Proposal Network”. In: *Int. S. Robotics Research (ISRR).* 2017.

[9] Ken Goldberg et al. “Part pose statistics: Estimators and experiments”. In: *IEEE Transactions on Robotics and Automation* 15.5 (1999), pp. 849–857.

[10] Tucker Hermans, James M Rehg, and Aaron Bobick. “Guided pushing for object singulation”. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS).* IEEE. 2012, pp. 4783–4790.

[11] Anne Holladay, Robert Paolini, and Matthew T Mason. “A general framework for open-loop pivoting”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on.* IEEE. 2015, pp. 3675–3681.

[12] Yifan Hou, Zhenzhong Jia, and Matthew T Mason. “Fast Planning for 3D Any-Pose-Reorienting Using Pivoting”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1631–1638.

[13] Imin Kao, Kevin Lynch, and Joel Burdick. “Contact Modeling and Manipulation”. In: *Springer Handbook of Robotics*. Springer, 2008, pp. 946–949.

[14] Marek Kopicki, Jeremy Wyatt, and Rustam Stolkin. “Prediction learning in robotic pushing manipulation”. In: *2009 International Conference on Advanced Robotics*. IEEE. 2009, pp. 1–6.

[15] Marek Kopicki et al. “Learning to predict how rigid objects behave under simple manipulation”. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 5722–5729.

[16] Gilwoo Lee, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. “Hierarchical planning for multi-contact non-prehensile manipulation”. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE. 2015, pp. 264–271.

[17] Kevin M Lynch. “Inexpensive conveyor-based parts feeding”. In: *Assembly Automation* 19.3 (1999), pp. 209–215.

[18] Kevin M Lynch. “The mechanics of fine manipulation by pushing”. In: *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE. 1992, pp. 2269–2276.

[19] Kevin M Lynch. “Toppling manipulation”. In: *ICRA*. 1999, pp. 2551–2557.

[20] Kevin M Lynch and Matthew T Mason. “Controllability of pushing”. In: *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*. Vol. 1. IEEE. 1995, pp. 112–119.

[21] Kevin M Lynch and Matthew T Mason. “Stable pushing: Mechanics, controllability, and planning”. In: *The International Journal of Robotics Research* 15.6 (1996), pp. 533–556.

[22] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[23] Jeffrey Mahler and Ken Goldberg. “Learning deep policies for robot bin picking by simulating robust grasping sequences”. In: *Conference on Robot Learning*. 2017, pp. 515–524.

[24] Jeffrey Mahler et al. “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics”. In: *Proc. Robotics: Science and Systems (RSS)*. 2017.

[25] Jeffrey Mahler et al. “Dex-Net 3.0: Computing Robust Robot Suction Grasp Targets in Point Clouds using a New Analytic Model and Deep Learning”. In: *arXiv preprint arXiv:1709.06670* (2017).

[26] Matthew T Mason. *Mechanics of robotic manipulation.* MIT press, 2001.

[27] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. “Super 4pcs fast global pointcloud registration via smart indexing”. In: *Computer Graphics Forum.* Vol. 33. 5. Wiley Online Library. 2014, pp. 205–215.

[28] Thomas Mörwald et al. “Predicting the unobservable visual 3d tracking with a probabilistic motion model”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE. 2011, pp. 1849–1855.

[29] Richard M Murray. *A mathematical introduction to robotic manipulation.* CRC press, 2017.

[30] Nancy S Pollard and Jessica K Hodgins. “Generalizing demonstrated manipulation tasks”. In: *Algorithmic Foundations of Robotics V.* Springer, 2004, pp. 523–539.

[31] Anil Rao, David J Kriegman, and Kenneth Y Goldberg. “Complete algorithms for feeding polyhedral parts using pivot grasps”. In: *IEEE Transactions on Robotics and Automation* 12.2 (1996), pp. 331–342.

[32] Radu Bogdan Rusu and Steve Cousins. “3d is here: Point cloud library (pcl)”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* IEEE. 2011, pp. 1–4.

[33] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[34] Atsushi Yamashita et al. “Motion planning of multiple mobile robots for cooperative manipulation and transportation”. In: *IEEE Transactions on Robotics and Automation* 19.2 (2003), pp. 223–237.

[35] Mike Tao Zhang and Ken Goldberg. “Designing robot grippers: optimal edge contacts for part alignment”. In: *Robotica* 25.3 (2006), pp. 341–349.

[36] Mike Tao Zhang and Ken Goldberg. “Gripper point contacts for part alignment”. In: *IEEE Transactions on Robotics and Automation* 18.6 (2002), pp. 902–910.

[37] Tao Zhang et al. “The toppling graph: Designing pin sequences for part feeding”. In: *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on.* Vol. 1. IEEE. 2000, pp. 139–146.