

Orienting Deformable Polygonal Parts without Sensors

Shawn M. Kristek and Dylan A. Shell

Abstract—Sensorless part orienting has proven useful in manufacturing and automation, while the manipulation of deformable objects is an area of growing interest. Existing sensorless orienting techniques may produce forces which have the potential to damage deformable parts. We present an algorithm that, when provided a geometric description of the part and a deformation model, generates a plan to orient the part up to symmetry from any initial orientation. The solution exploits deformation of the object under certain configurations to help resolve ambiguity. The approach has several attractive features: (1) the resulting plan is a short sequence of such actions guaranteed to succeed for all initial configurations; (2) the algorithm operates even with a very simple model of deformation, but is extensible when specialized knowledge is available; (3) failure to find a feasible solution has precise semantics (e.g., inadequate manipulator precision). We validate the algorithm experimentally with a pair of low-precision robot manipulators, orienting 6 parts made of 4 types of materials, with the correct orientation being reached on 80% of the 192 trials. Careful analysis of the failures emphasizes the importance of low-friction conditions, that increased manipulator precision would be beneficial but is not necessary, and a simple deformation model can suffice. In addition to illustrating the feasibility of sensorless manipulation of deformable parts, we note that the algorithm has applications to manipulation of non-deformable parts without the pressure switch sensor employed in existing sensorless orienting strategies.

I. INTRODUCTION

Research on sensorless manipulation has produced a series of methods that are useful for factory assembly lines and scenarios employing automation [1], [2], [3]. These techniques, however, make use of the rigidity of the objects they manipulate. There is a growing interest in the manipulation of deformable objects (*cf.*, the book by [4] and treatment by [5]) including the widely publicized demonstration by Maitin-Shepard *et al.* [6]. Applications which bring together both lines of research are easy to identify: many of the objects we wish sorted, organized, and packed are non-rigid. Automating such operations without using sensors has the potential to decrease cost, improve reliability, or both.

Broadly speaking, prior work on manipulating deformable objects is information-centric. Robots make extensive use of sensor information, prior knowledge of the object's initial configuration, and sophisticated explicit representations, some of which also model uncertainty. Henrich and Wörn [4] actually state that sensors are necessary for useful manipulation of deformable objects, a position which appears to be widely presupposed. Moreover, the robots themselves may have other forms of complexity, *e.g.*, intricate multi-fingered grippers are usual. In contrast, this work eschews such

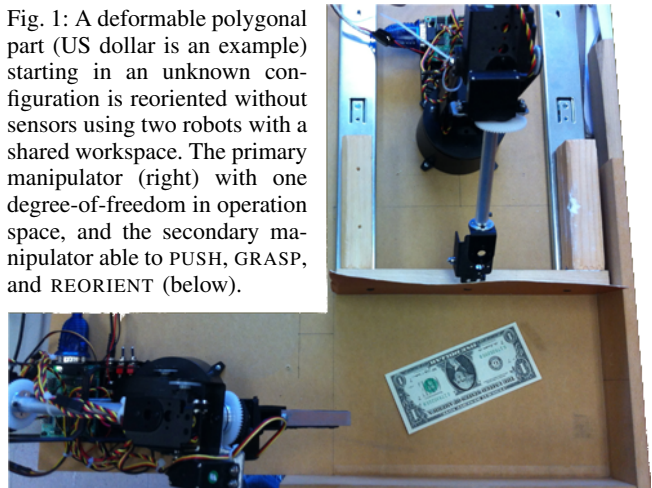
elements and employs a minimalist approach. No sensors are involved and no explicit representation of the object's configuration (or set of conceivable configurations) need be computed at run time. The robots we use are also cheap, simple, low precision devices.

At first glance it appears that doing away with sensors only makes the problem more challenging. In fact, Mason [7] observes that roboticists have reached a point of sensor dependency so that sensors are mistakenly seen as the only means of obtaining knowledge and reducing uncertainty. This work demonstrates that plans can be constructed that deform objects in order to gain information. Furthermore, because the sensorless approach avoids explicitly modeling information states of the system [8] at run time, it does not require a rich model of the way the object may deform (*e.g.*, using finite-element models, recording probabilities over a continuous shape distribution, *etc.*). Making fewer informational assumptions deemphasizes the race to build increasingly complex models to improve verisimilitude and precision; this reduces brittleness as a function of a myriad of model parameters (*e.g.*, friction coefficients, restitution constants, torsion moduli, and so on). Ultimately, this work reflects a different philosophical stance: a Pandora's box of challenges is opened once one opts to to build predictive models of complex deformable objects. Removing sensing alters the problem since, rather than attempting to observe the state of the world, one is content to understand which states are no longer possible.

II. RELATED WORK & PROBLEM SPECIFICATION

Sensorless parts orienting grew from reseach on compliant motion planning and the study of mechanical parts feeders:

Fig. 1: A deformable polygonal part (US dollar is an example) starting in an unknown configuration is reoriented without sensors using two robots with a shared workspace. The primary manipulator (right) with one degree-of-freedom in operation space, and the secondary manipulator able to PUSH, GRASP, and REORIENT (below).



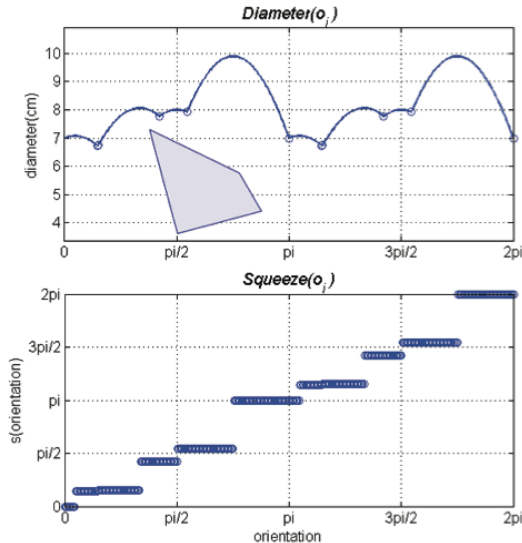


Fig. 2: $Diameter(o_i)$ is a plot of the four-gon part’s diameter for all orientations o_i . $Squeeze(o_i)$ is the final orientation after a sweep (squeeze) occurs, given an initial o_i .

the former found the idea of using mechanical compliance to reduce uncertainty, while the latter are devices specialized for repetitively orienting identical parts. This work utilizes compliant motion analysis initially developed by Brost [9] in his study of grasp planning with uncertainty for a parallel-jaw gripper and subsequently modified for sensorless parts orienting by Goldberg [1]. Brost’s push diagram represents all possible motions of a part grasped by a parallel-jaw gripper, allowing grasping plans robust to uncertainty in friction and orientation. Brost’s result uses the rule found by Mason [10] for predicting the direction a part will rotate when pushed under Coulomb friction.

Fig. 1 is an example of the scenario we consider in this work; two low precision robot manipulators were set up to share a workspace. We do not employ a reorientable parallel-jaw gripper and can not apply squeezes until the part is stably oriented because the part may deform.

III. APPROACH

A. Model Assumptions

We make explicit the assumptions of the model we employ in the table below. Essentially we use a classic model; assumptions (#1)–(#8) are from Brost [9], Mason et al. [11], and Goldberg [1], but with slight generalizations necessary for non-rigid parts: (#1) carries over from previous work, simply adding analogous “elastic forces” that arise from deformation; (#3) allows Goldberg’s analysis to be applied to deformable parts. Finally, (#9) is added to define the circumstances in which deformation occurs and these are quite natural given (#7). Controlled deformation of the object within only those stable orientations simplifies the deformation model, which otherwise could be onerous to construct.

Rather than posit a precise class objects, this paper axiomatizes the requirements as a list of assumptions. The model

for the deformation behavior itself is left unspecified, which allows it to be sophisticated or very simple, and is used for computing volumes within which the object falls. The D-space model of [14] for linearly elastic polygons (via a triangular finite element mesh model) provides circumstances in which assumption #9 holds, *i.e.*, when the orientations form *deform closure* grasps.

Assumptions
#1 Inertial forces, frictional forces, and part elastic forces are negligible in comparison to forces applied by manipulators.
#2 The sweeper and its fixed-wall can be modeled as a parallel-jaw; with all motions orthogonal to the jaws.
#3 The convex hull of the part can be treated as a semi-rigid thin planar polygon.
#4 Only one part is handled at a time.
#5 The part’s initial position is unconstrained as long as it lies somewhere between the walls. The part remains between the walls during sweeping, deforming, and pushing.
#6 The part’s center-of-mass (COM) is given and the coefficient of friction with the support surface is independent of position and velocity.
#7 There are zero frictional forces between the walls and part.
#8 Once contact is made between a wall and the part, the two surfaces remain in contact throughout the sweeping motion.
#9 Elastic deformation occurs only in <i>stable orientations</i> . This is a limited set of few orientations, each of which must be modeled via a deformation model.

B. Parts: Geometry

This work extends previous work [1], [9], [3] in which the dynamics of parts were analyzed using the parts’ polygonal convex-hulls. We refer to the hull as the geometric description. From this edges are generated, and the first of these is used as the orientation vector. We refer to the orientation vector using o_i where $0 \leq o_i \leq 2\pi$.

C. Goldberg’s Diameter and Squeeze Functions

Let $Diameter(o_i)$ be the diameter of a part in orientation o_i . Goldberg [1] defines this function as the distance separating the jaws of a parallel-jaw gripper when both jaws first make contact with the part. Interpretation in our context is slightly modified: it is the distance between the sweeper and its parallel fixed-wall when both make contact with the part. Next, $Squeeze(o_i)$ describes the rotational mappings of one orientation to another as the parallel-jaws or walls are moved closer together. $Squeeze$ received its name from the squeezing motion that occurs when a parallel-jaw gripper is closed. In this work we refer to this as sweeping rather than squeezing because a single movable wall resembles the motion of a broom towards the fixed-wall.

Observing Fig. 2 we can see that when squeezed sufficiently [1], or swept, the part will rotate to one of the orientations at one of the minimums seen in the plot of $Diameter(o_i)$. These minimums form a set O of important orientations referred to as *stable orientations* in the literature. One element of O for the four-gon part is shown in Fig. 3. As can be seen, a stable orientation occurs when at least one

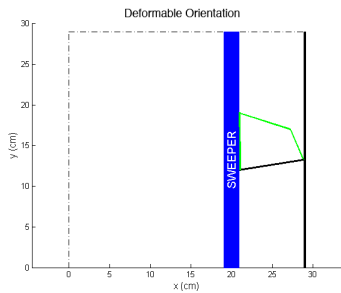


Fig. 3: A stable orientation of a four-gon part. This is an orientation in which controlled deformation of the part can occur.

edge aligns with the sweeper or its parallel fixed-wall. In this situation the rotational forces applied to the part sum to zero, preventing further rotation. With sufficient force, sweeping further from such an orientation will cause deformation of the part. For this reason, we refer to the set O as the set of deformable orientations.

Goldberg [1]’s squeeze operation involved the parallel-jaw gripper closing until the rigidity of the part meant that further reducing the diameter is impossible. This original squeeze used a pressure threshold, which would fail if the object is easily deformed, as in this work. Additionally, Goldberg would reorient his gripper at each stage of his algorithm, while our sweeper is not capable of reorienting.

Knowing the importance of the set O , we wish to exploit these whilst orientating. We need only be concerned with the squeeze portion of the sweeping action,¹ and when we perform a sweep, it begins from a region of the workspace that the part is not present. Also notice that squeezes only take place once the part has been pushed into contact with the fixed-wall. This results in the first possible squeeze occurring at the orientation with the largest diameter, and results in a natural ordering of the deformable orientations from largest to smallest diameters (Fig. 4).

Write the complete ordered set of k sets of deformable orientations as $O = \{\Theta_{k-1}, \Theta_{k-2}, \dots, \Theta_i, \dots, \Theta_0\}$. We produce controlled deformations of the object only in these orientations. Before proceeding further, we define the basic actions involved.

D. Manipulator Actions

$\text{SWEEP}(Diameter(o_i))$ moves the sweeper (primary manipulator) so that it is a distance of $Diameter(o_i)$ from the static wall. Subscript i denotes that these are values in O . Examples can be seen in Fig. 7a–7b and Fig. 7g.

$\text{DEFORM}(Diameter(o_i) - \delta_i)$ is essentially the same as the preceding operation as it is a sweeping motion that is applied only a distance of δ (recall, δ is the sweeping distance for safe and sufficient deformation) beyond how a SWEEP would do, but with the possibility of deforming the part. An implementation may also find it useful to apply additional down force in this action.

¹Owing to space limitations, we state this fact without proof; it follows from analysis similar to Goldberg’s for pushes preceding squeeze operations.

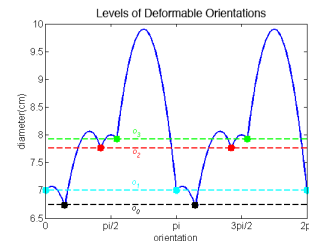


Fig. 4: The k levels of deformable orientations ordered from smallest to largest diameter. To help visualize this, one may imagine a horizontal sweep from the top downward.

$\text{PUSH}(\tilde{P}_{path})$ involves the secondary manipulator following \tilde{P}_{path} to push a deformed part; it reduces the possible positions along the axis perpendicular to the direction of sweeping action, see Fig. 7d and Fig. 7h.

$\text{GRASP}(\tilde{G}_{path})$ & $\text{REORIENT}(\tilde{R}_{path})$ work together. A part that has been deformed and subsequently pushed is in a well-defined position; the two actions first pick up the deformed part and then place it back within in the workspace within ϕ of $o_0 \in \Theta_0$ (see Fig. 7e–7f).

E. Models and Constraints

Two further elements must be introduced to completely formulate the actions and to decide whether deformation should be permitted (*i.e.*, without damaging the part).

1) The *Deformation Model*, denoted with part specific function $\text{Deformation}(\Theta_i, \delta_i) \mapsto \langle Vol_i, Com_i \rangle$, takes a set of orientations Θ_i and a deformation distance δ_i for the part and provides a volume Vol_i that is guaranteed to contain the deformed part and a spherical volume Com_i guaranteed to contain the part’s center of mass. The deformation model should indicate that a part will be damaged when δ_i is too great (formally, this can be done by returning an empty volume).

As illustrated in Fig. 5, the tent model used in the experiments follows the definition of the deformation model and provides Vol and Com by modeling the deformation with a tent or hinge model. Due to the simplicity of the tent model, these volumes can be overly large in order to be sufficiently conservative. Hirai et al. [12] worked on modeling thin deformable parts, similar to those presented in

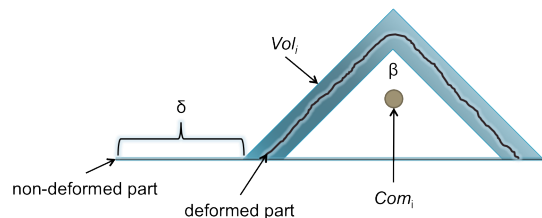


Fig. 5: A trivial deformation model for an arbitrary part we refer to as the tent model. δ is a deformation distance. β is the angle used to describe excessive deformation. Vol_i is a volume created by incorporating tolerances of the part deformation and is guaranteed to contain the deformed part. Com_i is a volume guaranteed to contain the part’s *COM*.

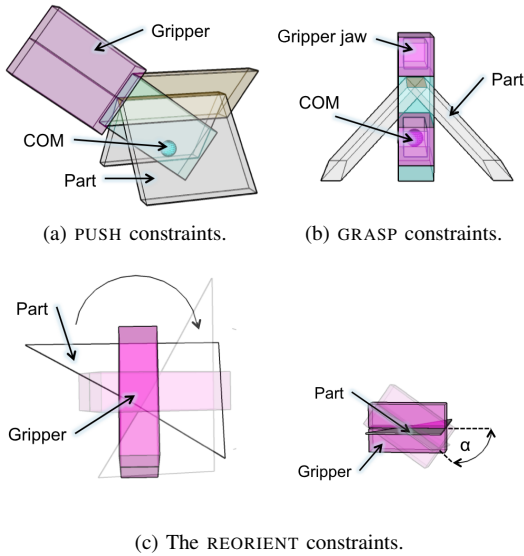


Fig. 6: End-effector constraints are necessary to constrain the trajectories found by the motion planner for the PUSH, GRASP, and REORIENT actions by the secondary manipulator.

this work, by analyzing the potential energy under geometric constraints. Such a model could be tighter, but we show that such accuracy is not always needed. Any deformation model of choice may be used.

2) *End-effector Constraints* are used for three actions in which a motion planner is necessary (PUSH, GRASP, and REORIENT). We did not want to constrain this work to any specific motion planner. Thus, we developed end-effector constraints that can use by a typical motion planner to guarantee successful actions. (These can also be refined in a straightforward way for different styles of manipulators and grippers.)

The first end-effector constraints are for PUSH, see Fig. 6a. These are volumetric constraints that guarantee a successful PUSH. Utilizing the two volumes returned by the deformation model, we need the manipulator action path planner to generate a \tilde{P}_{path} that will push the part without applying enough rotational forces to topple the part. We begin by projecting a volume, less the manipulator positioning tolerances, along the normal of the gripper. This volume must contain the Com_i provided by the deformation model. Next, we project a similar volume along the potential \tilde{P}_{path} that must contain a push-contact surface area on the part. The motion planner is then free to generate a sequence of collision free configurations that meet these constraints and will push the part to the corner. For the final $P_0 = \text{PUSH}(\tilde{P}_{path-0})$ the constraints are loosened to collision free pushing with the gripper tip pushing along the work surface.

For the GRASP end-effector constraints, see Fig. 6b, we followed the work of Smith et al. [13] in computing parallel-jaw grips for rigid polygonal parts by placing two grip points representing the jaws of the gripper collision free around the part. The placement of the grip points is constrained by the grip axis, the line joining the grip points, which must pass through the part's center-of-mass. We modify this last

requirement as follows: the grip axis is extended beyond both grip points and a volume based on the dimensions of the jaws is projected along the grip axis. This grip volume must contain the Com_i provided by the deformation model. We assume that the foam lining the parallel-jaw gripper prevents undesirable deformation of the part during grasping. For a more accurate GRASP applicable to more manipulators the work of Gopalakrishnan and Goldberg [14] should be considered, where they take into consideration factors that include deformation due to grasping.

REORIENT is a complex action akin to a pick and place operation that we have strived to define by a simple set of end-effector constraints, see Fig. 6c. Using a motion planner we place the following requirements on the manipulator configuration path. From the final GRASP manipulator configuration there must exist a manipulator configuration that holds the part at a $o_0 \pm \phi$, where $o_0 \in \Theta_0$, with the parallel-jaws at an angle α for which the gravitational forces are sufficient to pull the part to the work-surface. This assumes that the release/place phase of REORIENT is smooth and slow enough that the part will stay in $o_0 \pm \phi$. The tolerance ϕ is extracted from the *Diameter* plot and is the maximum allowed reorientation difference from the goal orientation $o_0 \in \Theta_0$ that will still guarantee successful reorientation.

IV. ALGORITHM

Input for a particular part takes the form of a geometric description and a deformation model, which are each described in more detail below. First, the algorithm computes the set O of deformable orientations by generating the *Diameter* function and then applying the *Squeeze* function (as described in Section III-C). The set deformable orientations is then group into subsets based on symmetry in *Diameter* and placed in O , which is then sorted on diameter. Working in reverse order, Algorithm IV.1 will first call upon the manipulator action path planner (Algorithm IV.2) to determine a safe and sufficient δ_i and calculate \tilde{P}_{path-i} , \tilde{G}_{path-i} , and \tilde{R}_{path-i} .

To determine δ_i and the action paths the manipulator action path planner, requires the symmetrical subset of deformable orientations Θ_i and the deformation model. The manipulator action path planner begins by initializing Δ a user specified parameter for the step size of δ_i , each of the action constraints, the work space configuration WS , a container VC for the volumes returned by the deformation model, δ_i , and the action paths.

The manipulator action path planner will pass WS , VC , and the relevant action constraints to a motion planner to determine each of the action paths. For the terminating orientation the manipulator action path planner will only find a \tilde{P}_{path-0} , and for any other Θ_i the manipulator action path planner will determine a δ_i while querying the motion planner for a \tilde{P}_{path-i} , \tilde{G}_{path-i} , and \tilde{R}_{path-i} .

Once the manipulator action path planner has returned the action paths and δ_i , the algorithm will determine if the precision is sufficient by assuring that action paths were found and that a $\text{DEFORM}(\text{Diameter}(o_i) - \delta_i)$ will not

deform $o_{i-1} \in \Theta_{i-1}$. This check of δ_i is necessary to maintain the ability to distinguish between $o_i \in \Theta_i$ and $o_{i-1} \in \Theta_{i-1}$. For the terminating orientation, the algorithm only needs to assure \tilde{P}_{path-0} exists to check for sufficient precision.

There are three reasons the algorithm might determine there is insufficient precision:

- 1) The deformation model is too pessimistic, *i.e.*, volumes that bound the deformed object are too large.
- 2) Robots are too imprecise, so the tolerances are too large and the end-effector constraints are unable to be met.
- 3) The motion planner is not complete.

If the precision is insufficient then the algorithm will return an indication that greater precision or a complete motion planner² is needed. Otherwise, when precision is sufficient, the algorithm will generate a sequence of $S_i = \text{SWEEP}(\text{Diameter}(o_i))$, $D_i = \text{DEFORM}(\text{Diameter}(o_i) - \delta_i)$, $P_i = \text{PUSH}(\tilde{P}_{path-i})$, $G_i = \text{GRASP}(\tilde{G}_{path-i})$, and $R_i = \text{REORIENT}(\tilde{R}_{path-i})$ for each deformable orientation with a diameter greater than the terminating orientation o_0 terminated by a sequence of $S_0 = \text{SWEEP}(\text{Diameter}(o_0))$, and $P_0 = \text{PUSH}(\tilde{P}_{path-0})$ for the terminating orientation o_0 . This combined sequence of actions will be returned as the Plan to be executed by the robot manipulators.

Algorithm IV.1 Constructing an Orienting Plan

Input:

- *Geometricdescription*: a counter-clockwise list of vertices of the convex-hull: $geo = \{\langle x_0, y_0 \rangle, \langle x_1, y_1 \rangle, \dots, \langle x_k, y_k \rangle\}$
- *Deformation model*: $DM(o_i, \delta_i) \mapsto \langle Vol_i, Com_i \rangle$

- 1: Find the k deformable orientations O , by utilizing *Squeeze* and grouping symmetrical deformable orientations using *Diameter*.
- 2: Order O such that $\text{Diameter}(o_0) < \dots < \text{Diameter}(o_{k-1})$
- 3: **for** $i = k - 1$ **down to** 1 **do**
- 4: $\langle \tilde{P}_{path-i}, \tilde{G}_{path-i}, \tilde{R}_{path-i} \rangle \leftarrow MAPP(\Theta_i, DM(o_i, \delta_i))$
- 5: **if** $(\tilde{P}_{path-i} = \emptyset \vee \tilde{G}_{path-i} = \emptyset \vee \tilde{R}_{path-i} = \emptyset) \vee (\text{Diameter}(o_i) - \delta_i < \text{Diameter}(o_{i-1}))$ **then**
- 6: **return**: "Insufficient precision for Θ_i or motion planner incomplete."
- 7: **end if**
- 8: $S_i = \text{SWEEP}(\text{Diameter}(o_i))$
- 9: $D_i = \text{DEFORM}(\text{Diameter}(o_i) - \delta_i)$
- 10: $P_i = \text{PUSH}(\tilde{P}_{path-i})$
- 11: $G_i = \text{GRASP}(\tilde{G}_{path-i})$
- 12: $R_i = \text{REORIENT}(\tilde{R}_{path-i})$
- 13: **end for**
- 14: $\tilde{P}_{path-0} \leftarrow MAPP(\Theta_0, DM(o_0, \delta_0 = 0))$
- 15: **if** $(\tilde{P}_{path-0} = \emptyset)$ **then**
- 16: **return**: "Insufficient precision for o_0 or motion planner incomplete."
- 17: **end if**
- 18: $S_0 = \text{SWEEP}(\text{Diameter}(o_0))$
- 19: $P_0 = \text{PUSH}(\tilde{P}_{path-0})$

Output: Plan = $\{S_{k-1}, D_{k-1}, P_{k-1}, G_{k-1}, R_{k-1}, \dots, S_0, P_0\}$

²If a probabilistically complete planner like a sample-based method is used, then this can be repeated with a policy to increase the number of samples until either a solution is found or the probability of failing to find a solution is low enough to satisfy the user.

Algorithm IV.2 Manipulator Action Path Planner (MAPP)

Input:

- Part orientation: o_i
- *Deformation model*: $DM(o_i, \delta_i) \mapsto \langle Vol_i, Com_i \rangle$
- Δ : parameter for δ_i step size

- 1: $PC_i \leftarrow \text{PUSH}(\Theta_i)_{constraints}$ {PUSH end-effector constraints $\forall o_i \in \Theta_i$ }
- 2: $PC_0 \leftarrow \text{PUSH}(\Theta_0)_{constraints}$ {PUSH end-effector constrain. $\forall o_0 \in \Theta_0$ }
- 3: $GC \leftarrow \text{GRASP}_{constraints}$ {GRASP end-effector constraints $\forall o_i > o_0$ }
- 4: $RC \leftarrow \text{REORIENT}_{constraints}$ {REORIENT end-effector const. $\forall o_i \in \Theta_i$ }
- 5: $WS \leftarrow$ configuration following previously determined actions
- 6: $VC \leftarrow \emptyset$ {container for deformation model volumes}
- 7: $\delta_i \leftarrow 0$
- 8: $\tilde{P}_{path-i} \leftarrow \tilde{G}_{path-i} \leftarrow \tilde{R}_{path-i} \leftarrow \emptyset$
- 9: **if** $o_i = o_0$ **then**
- 10: **for each** o_j in Θ_i **do**
- 11: $VC \leftarrow VC \cup DM(o_j, \delta_i)$
- 12: **end for**
- 13: $\tilde{P}_{path-i} \leftarrow \text{MotionPlanner}(WS, PC_0, VC)$
- 14: **else**
- 15: **for** $\delta_i = 0$ to $\text{Diameter}(o_i)$ **do**
- 16: $\delta_i \leftarrow \delta_i + \Delta$
- 17: **for each** o_j in Θ_i **do**
- 18: $VC \leftarrow VC \cup DM(o_j, \delta_i = 0)$
- 19: **end for**
- 20: $\tilde{P}_{path-i} \leftarrow \text{MotionPlanner}(WS, PC_i, VC)$
- 21: $\tilde{G}_{path-i} \leftarrow \text{MotionPlanner}(WS, GC, VC)$
- 22: $\tilde{R}_{path-i} \leftarrow \text{MotionPlanner}(WS, RC, VC)$
- 23: **if** $(\tilde{P}_{path-i} \neq \emptyset) \wedge (\tilde{G}_{path-i} \neq \emptyset) \wedge (\tilde{R}_{path-i} \neq \emptyset)$ **then**
- 24: **return**: $\langle \tilde{P}_{path-i}, \tilde{G}_{path-i}, \tilde{R}_{path-i}, \delta_i \rangle$
- 25: **end if**
- 26: **end for**
- 27: **end if**

Output: $\langle \tilde{P}_{path-i}, \tilde{G}_{path-i}, \tilde{R}_{path-i}, \delta_i \rangle$

V. ANALYSIS

A. Correctness

We sketch the manner of proof. To prove correctness one shows that the initial set of configurations C_k consisting of all possible $x \in X$, $y \in Y$, and $\theta \in \Theta$ is reduced in steps C_{k-1}, C_{k-2}, \dots to the set of configurations C_1 , defined as:

$$C_1 \equiv \left\{ \begin{array}{l} \langle x, y, \theta \rangle \mid x \in X \\ y \in [0, \text{Diameter}(\Theta_1)] \\ \theta \in \{\Theta \mid \text{Diameter}(\theta) < \text{Diameter}(\Theta_1)\} \end{array} \right\}.$$

This is achieved via a repeating sequence of five actions (*SDPGR*): SWEEP, DEFORM, PUSH, GRASP, REORIENT; each sequence is parameterized for a set of symmetrical orientations $\Theta_i \in O$, where each Θ_i is a set because orientations are defined up to symmetry. Once the set of configurations has been reduced to C_1 a sequence of two actions parameterized for Θ_0 , $(SP)_0$, will reduce it further to the terminating set of configurations C_0 :

$$C_0 = \langle 0, 0, \theta \in \Theta_0 \rangle. \quad (1)$$

For the block of five actions $(SDPGR)_i$, each sweep reduces the Y set of conceivable locations of the part; it marches steadily downwards to each diameter, for each i . An object that will deform can only be in a deformable orientation, and so the deformation by δ_i allows exactly items in those orientations to be pushed (reducing their X), picked up, and reoriented. Parts in non-deformable orientations are unaffected. As progress is made, the index i decreases, the feasible orientations are orientations below

the $Diameter(\Theta_i)$. This can be envisioned as a repeatedly interrupted and resumed “squeeze” operation. The set C_1 is reached when exactly k steps have been reached, with any part being deformed and reoriented at most once.

B. Plan length

The algorithm generates a plan which consists of a sequence of k (*SDPGR*)-type actions, one for each set Θ_i of symmetrical deformable orientations. Fewer than k cannot guarantee to orient a deformable part from all initial orientations and poses. To prove this, remove any action (say i), then one can construct a scenario (by put part in configuration $C = \langle x \in X, y = 0, \theta \in \Theta_i \rangle$, which places the part in a deformable orientation along the fixed-wall) which will be incorrectly handled as the part will either be incorrectly oriented finally or be deformed more than δ_i (i.e., potentially violating the safety constraints). Thus, the plan is short in that it has length $\in \mathcal{O}(|O|)$.

VI. EXPERIMENTAL VALIDATION

The experimental setup, shown in Fig. 1, has two low precision Lynxmotion manipulators. They were set up at right angles to one another with fixed walls on opposite sides of a $29cm^2$ rectangular workspace. We basically replicated the effect of a parallel-jaw gripper by making a moveable wall (the sweeper) with a parallel fixed-wall and only allowed the sweeper to move orthogonal to its parallel fixed wall. The primary manipulator was set up to move the 1 degree-of-freedom sweeper, and the secondary manipulator was set up with seven degrees-of-freedom and a parallel-jaw gripper that allowed it to perform all of the reorientation actions: PUSH, GRASP, and REORIENT. To remain entirely sensorless, the inverse kinematics were used to calculate pulse commands that were sent serially to the manipulator control boards, creating open-loop motions for each action.

The parts used for the experiments were made of card stock, dollar bills, a textile patch, or foam material, each of which was readily available, and proved easy to alter into different shaped parts (see Table VI.1). Each of the four materials had similar deformation characteristics that allowed the use of the simplistic linear tent deformation model. Of the nine parts used for the experiments, the algorithm generated plans for six of them, the other three being determined to be infeasible with the devices we employed. These plans were then executed thirty-two times for each part with the part being placed in a different initial orientation and pose each time. The successes, failures, and reasons for failures were recorded and are shown in Table VI.1. Note that we have two measures for success: complete success is when a part is correctly oriented and posed at the execution of the plan without any violations of the assumptions or failures of manipulator actions, and partial success is when the part is correctly oriented after executing the plan, but some assumptions were violated or actions fail during execution of the plan (e.g., the part may stick slightly, or slip under the sweeper, but still end up correctly oriented).

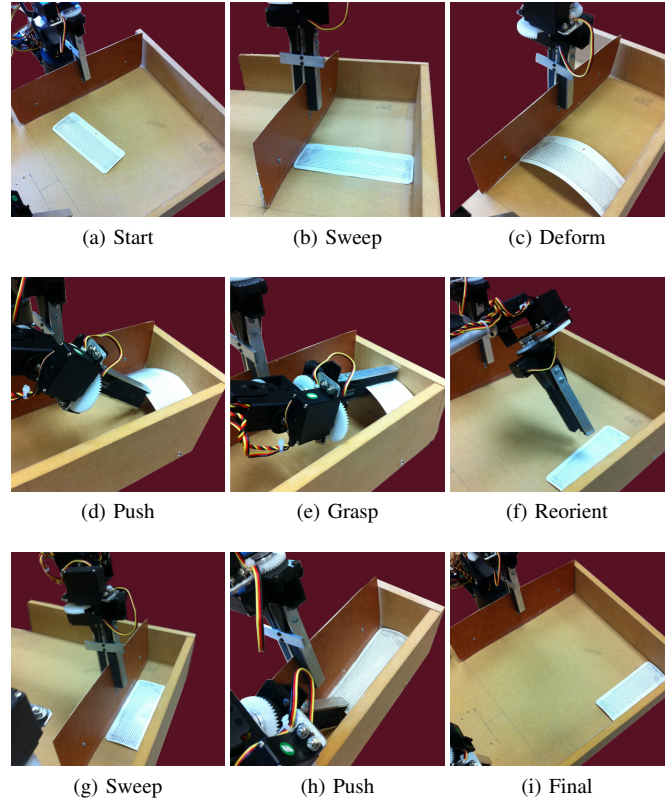


Fig. 7: Complete plan for Card part.

Fig. 8 illustrates some of the data from Table VI.1 as a whole and broken down by part.

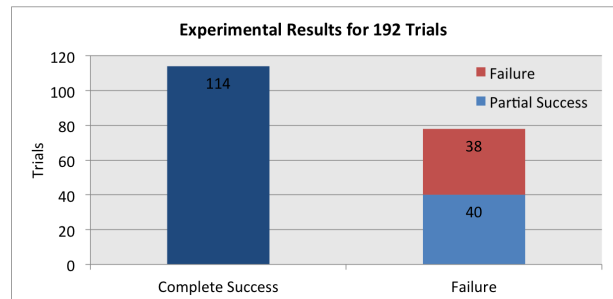











Fig. 8: The complete successes, partial successes, and failures of the experiments.

VII. APPLICATION TO TRULY SENSORLESS ORIENTING OF RIGID PARTS

The algorithm generates plans that guarantee a deformable part will be oriented up to symmetry in its convex-hull. Here we describe possible modifications that would allow the algorithm to orient rigid parts, but would do away with the pressure threshold employed by Goldberg in his method. In order to exploit the *stable orientations* for rigid parts we propose the following alterations to this work. A vertical degree of freedom is added which allows the surface of workspace to be lowered and raised. First, we replace DEFORM with LOWERWORKSURFACE and reorder

TABLE VI.1: The experimental results of 192 trials of 6 parts.

Part	Material	Complete Successes	Failures			
			Friction	Manip. Precision	Deform. Model Accuracy	Complete
 Card	card stock	22	6	4	0	4
 Dollar	dollar bill	7	11	3	11	14
 Patch	textile	23	6	1	2	3
 Foam1	foam	18	7	5	1	9
 Foam2	foam	22	7	3	1	7
 Foam3	foam	22	9	1	0	1
 Foam4	"Precision insufficient." →Due to overlapping deformation.					
 Foam5	"Precision insufficient." →Failed to meet PUSH end-effector constraints.					
 Foam6	"Precision insufficient." →Failed to meet REORIENT end-effector constraints.					

the sequence of actions to: SWEEP, PUSH, LOWERWORKSURFACE, GRASP, RAISEWORKSURFACE, and REORIENT. LOWERWORKSURFACE along with the assumption vertical friction forces exist between the sweeper and the part, but horizontal frictional forces do not (approximated by use of horizontal roller-bearings) allows SWEEP($Diameter(o_i)$) to apply enough force to hold the rigid part in a *stable orientation* in place for a GRASP and then a REORIENT to occur.

VIII. CONCLUSION

Our two-tier definition of success shows that even with a degree of pessimism, sensorless orienting of deformable parts is possible. On aggregate, the approach described works the majority of the time. Naturally, the success rate depends on the part itself. The first salient point to address in Table VI.1 is that of the failures due to friction. The essential assumption of no friction between the part and parallel-jaws (adopted widely in previous work) was found to be even more important in this work where the part slides along the walls (in a PUSH). It is understood that the zero friction assumption an approximation for when frictional forces are merely negligible, but this is more often a cause of concern for our experiments. Nevertheless, the assumption is workable under the conditions that can be realized physically. The second leading type of failure is a result of the manipulators having not only low precision, but also non-linear variance of the precision throughout the manipulator's configuration space. The last type of failure was a result of over simplification of the deformation model for parts such as the dollar that had more variation in the deformation.

REFERENCES

- [1] K. Y. Goldberg, "Orienting Polygonal Parts without Sensors," *Algorithmica*, vol. 10, no. 3, pp. 201–225, 1993.
- [2] M. Erdmann and M. T. Mason, "An Exploration of Sensorless Manipulation," *IEEE Journal of Robotics and Automation*, vol. 4, no. 1, pp. 369–379, August 1991.
- [3] S. Akella and M. T. Mason, "Posing Polygonal Objects in the Plane by Pushing," in *IEEE International Conference on Robotics and Automation*, 1992, pp. 2255–2262.
- [4] D. Henrich and H. Wörn, *Robot Manipulation of Deformable Objects*. London, UK: Springer, 2000.
- [5] S. Hirai, T. Tsuboi, and T. Wada, "Robust Grasping Manipulation of Deformable Objects," in *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, 2001, pp. 411–416.
- [6] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth Grasp Point Detection Based on Multiple-view Geometric Cues with Application to Robotic Towel Folding," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 2308–2315.
- [7] M. T. Mason, "Kicking the Sensing Habit," *AI Magazine*, vol. 14, no. 1, pp. 58–59, 1993.
- [8] S. M. LaValle, "Sensing and filtering: A tutorial based on preimages and information spaces," in *Foundations and Trends in Robotics*, 2011.
- [9] R. C. Brost, "Automatic Grasp Planning in the Presence of Uncertainty," *International Journal of Robotics Research*, vol. 7, no. 1, pp. 3–17, 1988.
- [10] M. T. Mason, "Manipulator Grasping and Pushing Operations," Ph.D. dissertation, Dept. of Computer Science, MIT, Cambridge, MA, June 1985.
- [11] M. T. Mason and K. Y. Goldberg and R. H. Taylor, "Planning Sequences of Squeeze-Grasps to Orient and Grasp Polygonal Objects," in *Seventh CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators*, 1988.
- [12] S. Hirai, H. Wakamatsu, and K. Iwata, "Modeling of Deformable Thin Parts for Their Manipulation," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, vol. 4, May 1994, pp. 2955–2960.
- [13] G. Smith, E. Lee, K. Goldberg, K. Bohringer, and J. Craig, "Computing Parallel-jaw Grips," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 3, May 1999, pp. 1897–1903.
- [14] K. Gopalakrishnan and K. Goldberg, "D-Space and Deform Closure Grasps of Deformable Parts," *International Journal of Robotics Research*, vol. 24, no. 11, November 2005.