# Parts Orienting with Partial Sensor Information

Srinivas Akella
Beckman Institute
University of Illinois, Urbana-Champaign
Urbana, IL 61801

Matthew T. Mason
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

## Abstract

*Parts orienting, the process of bringing parts in initially unknown orientations to a goal orientation, is an important aspect of automated assembly. Bowl feeders used in industry rely on a sequence of mechanical operations, without using sensors, to orient parts. In our work, we use partial information sensors along with mechanical operations to eliminate uncertainty in part orientation. We show that sensor-based orienting plans need $O(m)$ operations, where $m$ is the maximum number of states with the same sensor value. We characterize the relation between part shape, orientability, and recognizability to identify conditions under which a single plan can orient and recognize multiple part shapes. We describe implemented planners and experiments to demonstrate generated plans.*

## 1 Introduction

A parts orienting system must bring randomly oriented parts to goal orientations for assembly. Bowl feeders, widely used in industry, rely on mechanical operations to orient parts (Boothroyd *et al.* [4]). While very effective, bowl feeders are not reconfigurable. It can take a couple of months to design a feeder for a new part or to discover the part cannot be oriented. So there is a need for flexible parts orienting systems with characterizable capabilities.

We use sensors that provide partial information on part orientation in combination with mechanical operations to flexibly generate orienting strategies with simple hardware elements. The actions bring the part to a finite set of states, and the sensors, which could be photosensors or contact sensors, help distinguish them. We focus on an implementation where a polygonal part on a moving conveyor belt is aligned against a stationary fence. A robot executes a sequence of *push-align operations* to identify the part's orientation. A push-align operation involves the robot rotating the part before it is aligned at the fence, and an LED sensor measuring the part diameter after alignment (Figure 1).

We show that sensor-based plans using push-align operations always exist to orient any polygonal part up to symmetry in a part shape dependent function. We prove the worst-case length of a sensor-based orienting plan is $2m - 1$ operations, where $m$ is the maximum number of states with the same sensor value, and present an algorithm to generate minimum length plans. We prove that even with sensing, any polygonal part cannot be distinguished from an infinite set of polygons during orienting. We identify conditions under which multiple part shapes can be oriented and recognized with a single sensor-based plan.

Portions of this work appeared earlier in (Akella [1]).

## 2 Related Work

*Sensor-based manipulation*. Grossman and Blasgen [12] used a tilted tray to bring objects to a finite number of orientations and distinguished them with a tactile probe. Our work is most closely related to that of Taylor *et al.* [19] who automatically generated sensor-based orienting plans for tray tilting and parallel-jaw grasping using AND/OR search, and that of Rao and Goldberg [17], who characterized orientability and recognizability of multiple parts by parallel-jaw grasping with diameter sensing. Canny and Goldberg [8] advocate the use of simple, inexpensive sensors and actuators for automation tasks.

*Parts feeding*. Boothroyd *et al.* [4] describe parts orienting devices for automated assembly, including vibratory bowl feeders. The SONY APOS parts feeder (Shirai and Saito [18]) uses a vibrating pallet with nests designed to trap only parts in the correct orientation. Brost [6] uses shape constraints to orient a part by designing a nest to "catch" it. Caine [7] applies motion constraints from interacting part shapes to vibratory bowl feeder track design. Krishnasamy *et al.* [13] analyze the effect of shape and vibration parameters on the entrapment efficiency of parts in an APOS-like system.

Erdmann and Mason [10] used sensorless manipulation strategies to eliminate configuration uncertainty of a part in a tray by repeatedly tilting it. Peshkin and Sanderson [16] used a sequence of fences over a conveyor to automatically orient a sliding part. Wiegley *et al.* [21] developed a complete algorithm to find the shortest sequence of frictionless curved fences to orient a polygonal part. Böhringer *et al.* [3] use a vibrating plate to position and orient parts sensorlessly. Zumel and Erdmann [22] use a frictionless two-palm manipulator to bring polygons in an unknown initial orientation to a goal orientation.

*Pushing and grasping*. Mason [15] derived rules to predict

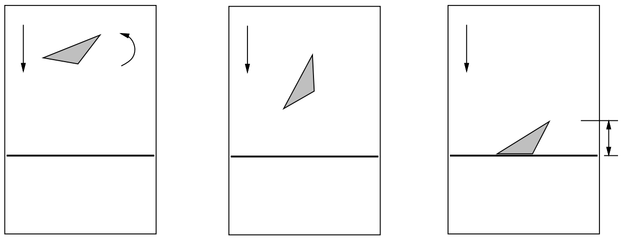

**Figure 1:** Schematic overhead view of a part on the conveyor during a push-align operation. The conveyor motion is “downward.”

the rotation direction of a pushed object. Mani and Wilson [14] used these rules to develop a system to orient parts with a sequence of fence pushes at different angles. Goldberg [11] developed an algorithm to generate optimal sensorless orienting plans for polygonal objects using a frictionless gripper. van der Stappen *et al.* [20] characterize the number of sensorless orienting actions for a part in terms of its eccentricity. The sensorless 1JOC system (Akella *et al.* [2]) uses a one degree of freedom fence to orient parts on a conveyor.

# 3 Mechanics

We assume a part in an unknown initial orientation drifts on a conveyor belt until it contacts a fence placed perpendicular to the conveyor motion (Figure 1). The part then rotates so one of its edges is aligned against the fence. An LED sensor measures the diameter of the part perpendicular to the fence. This provides partial information on the part orientation. Our goal is to find a sequence of *push-align operations* to orient the part. Each operation consists of an action followed by a sensor reading of the diameter of the aligned part. The action consists of the robot picking up an aligned part at the fence by suction, translating upstream from the fence, rotating the part by a chosen angle, and releasing it on the conveyor to be aligned at the fence. Orienting a part means identifying the part edge aligned against the fence.

In our analysis, we assume all parts are polygons of known shape, with the center of mass at a known position in the part interior. Nonconvex polygons are treated by considering their convex hulls. All motions are quasi-static, and all frictional interactions are described by Coulomb friction. There is a uniform coefficient of friction between the part and conveyor surface, and zero friction between the part and fence.

## 3.1 The Radius Function

When a part on the moving conveyor contacts the fence, it is pushed normal to the fence. The radius function (Goldberg [11]) predicts the part rotation due to such a linear normal push. The **radius function** of a polygon is a mapping $r : \mathcal{S}^1 \rightarrow \mathcal{R}^1$ from the orientation $\phi$ of a supporting line of the polygon to the perpendicular distance $r$ from the polygon center of mass to the supporting line (Figure 2). The local minima of the radius function correspond to stable edges of the part. A part rotates to achieve a minimum radius and

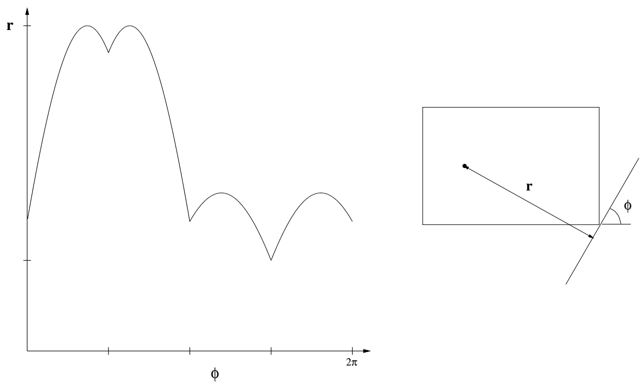


**Figure 2:** The radius function for the rectangle with its center of mass indicated by the black dot. Based on (Goldberg [11]).

comes to rest with stable edges aligned with the fence. An edge is *stable* if the center of mass projects onto its interior. An edge is *unstable* if the center of mass projects outside the edge, and is *metastable* if the center of mass projects onto a vertex of the edge. Each local maximum determines a divergent orientation. A push maps the entire interval between two divergent orientations to the enclosed stable orientation. The radius function has a period of $2\pi$. Symmetry in part shape leads to periodicity in the radius function.

## 3.2 The Push and Push-Diameter Functions

The **push function** (Goldberg [11]) is a mapping $p : \mathcal{S}^1 \rightarrow \mathcal{S}^1$ from the initial fence orientation relative to the part to the final fence orientation relative to the part after a push, assuming the part orientation is held constant. It is a monotonic step function with a period of $2\pi$. A *symmetric* push function has a period $T < 2\pi$ such that $p(\phi + T) = (p(\phi) + T) \bmod 2\pi$.

The **push-diameter function** (Figure 3) describes the part diameter resulting from a push for a given initial part orientation. It is a mapping $d : \mathcal{S}^1 \rightarrow \mathcal{R}^1$ from the initial fence orientation $\phi$ relative to the part to the part diameter $d$ perpendicular to the fence resulting from a push, assuming the part orientation is held constant. The push-diameter function is a step function with a period of $2\pi$.

Let $S$ be the set of stable orientations $\{\phi_s\}$ of the radius function and let $0 < T < 2\pi$. A *periodic* push-diameter function has a period $T$ such that $d(\phi) = d(\phi + T)$. A *quasiperiodic* push-diameter function has a period $T$ and at least one stable orientation $\phi_s$ such that $d(\phi) = d(\phi + T)$ when $\phi$ is measured relative to $\phi_s$, and at least one stable orientation $\phi_s^{'} \in S$ such that $d(\phi) \neq d(\phi + T)$ for some $\phi$ measured relative to $\phi_s^{'}$. A *symmetric* push-diameter function has a period $T$ such that $d(\phi) = d(\phi + T)$ and for every $\phi_s \in S$, there exists $\phi_s^{'} = (\phi_s + T) \bmod 2\pi$, $\phi_s^{'} \in S$.

## 3.3 Finding Representative Actions

The *resting range* of a stable orientation is the set of initial part orientations relative to the fence that converge to the stable orientation (Figure 4), and is obtained from the ra-

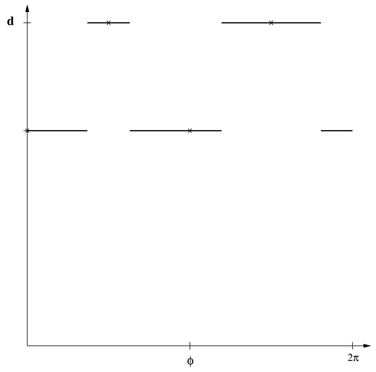


**Figure 3:** The push-diameter function for the rectangle of Figure 2.

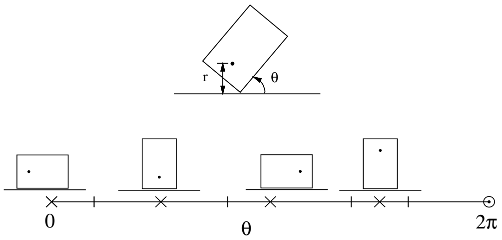


**Figure 4:** Resting ranges of the rectangle. The x's indicate stable orientations and the vertical bars indicate resting range limits. This diagram corresponds to a slice of the push stability diagram of Brost [5] along the 90 degree pushing direction.

dius function. Push-align actions are specified by the angle the part is rotated through before being pushed, and can be grouped into equivalence classes. An *action range* is an interval of rotations for which a part transitions from one edge to another (Figure 5). Every action in an action range results in the same final state when applied to the same initial state. For a part with $n$ stable states, we compute the action ranges for a set of $k$ states from the overlap of their individual action range intervals, to obtain another set of intervals, the *overlap ranges* (Figure 6). Since overlap ranges are equivalence classes, the set of *representative actions* selected from the interior of the $kn$ overlap ranges covers the set of actions.

# 4 Sensorless Orienting

A sensorless orienting plan is a sequence of push-align actions that brings any initial state to the same final state. Goldberg's polynomial-time algorithm ([11]) applied to a push function yields a minimum length plan to orient a part up to symmetry in its push function. The plans are $O(n)$ in length, where $n$ is the number of stable edges (Chen and Ierardi [9]).

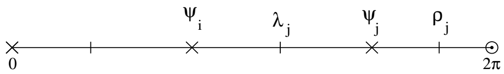


**Figure 5:** The action range for the transition from state $s_i$ to state $s_j$ is the open interval $(\lambda_j - \psi_i, \rho_j - \psi_i)$, where $\psi_i$ is the stable orientation of $s_i$, and $\lambda_j$ and $\rho_j$ are the left and right limits of the resting range of $s_j$.

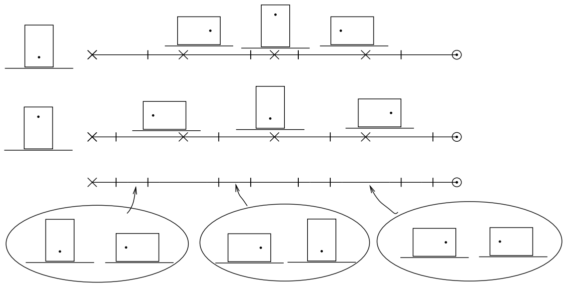

**Figure 6:** Action ranges for an indistinguishable set of states of the rectangle, with the resulting stable states. The overlap ranges are at the bottom, with the set of resulting states for three selected ranges also shown.

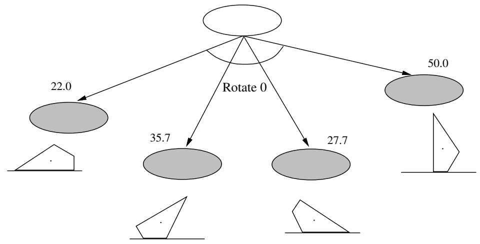


**Figure 7:** A part with unique diameter values at its stable orientations can be oriented in a single step. The arrows linked by an arc represent a push-align operation.

# 5 Sensor-Based Orienting

Sensor-based orienting plans measure the diameter of the aligned part after every push. These measurements provide only partial information on the part orientation. The sensor consists of two parallel arrays of LEDs and phototransistors perpendicular to the fence on either side of the conveyor. The LED spacing determines the sensor resolution. By determining the range of sensor values consistent with each stable state, we can capture the effect of sensor resolution and sensor noise. Two states with overlapping sensor ranges are considered *indistinguishable*. An **indistinguishable set** is a set of states where each state is indistinguishable from at least one other state. A $k$-indistinguishable set has $k$ states.

A single push-align operation identifies part orientation when every stable edge has a unique diameter value (Figure 7). When several orientations are indistinguishable, we need a conditional plan with a sequence of push-align operations (Figure 8). We seek *optimal* plans that minimize the maximum number of actions to orient a part. After proving bounds on plan length, we outline an algorithm to find optimal sensor-based orienting plans.

## 5.1 Plan Length

Plan length is the maximum number of operations required to orient a part. We prove plan length bounds in terms of $m$, the maximum number of indistinguishable stable states, and $n$, the number of stable states, using the following part classification (Figure 9).

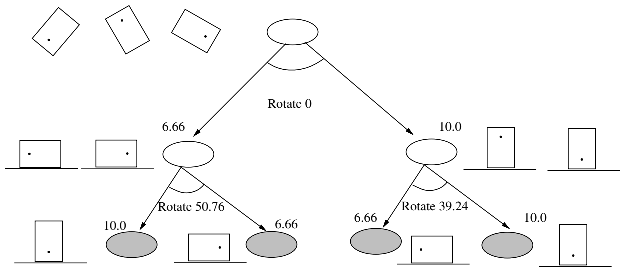


**Figure 8:** A sensor-based plan to orient the rectangle of Figure 2. Each node contains a set of indistinguishable states. Goal nodes, shown shaded, have a single orientation. The sensor value shown at a node indicates the diameter value of its indistinguishable states.

1. *Parts with all states having unique diameter values*: A single step is sufficient to identify any stable state from its diameter value.
2. *Parts with some states with unique diameter values*: Group states into indistinguishable sets. Let the part be in a state from an $m$-indistinguishable set $M$. Apply a rotation of $\theta$ to all states in $M$. Select $\theta$ so some initial state is in the resting range of a unique diameter valued state. This action reduces the number of possible states by at least one. In the worst case, we have $m - 1$ resulting states to be distinguished. We recursively continue this process on these states, eliminating at least one state with each action, until we are left with a single state. So the maximum number of actions to orient the initial set of $m$ states is no greater than $m$.
3. *Parts with multiple diameter values, none unique, and an asymmetric aperiodic push-diameter function*: Let the initial state belong to an $m$-indistinguishable set. Since the push-diameter function is asymmetric and aperiodic, there is always some action that leads the indistinguishable states to states with distinguishable diameters. This means the largest resulting indistinguishable set has at least one fewer state than the initial set. Applied recursively to the resulting set of states, this shows that the maximum number of steps to orient the part is no greater than $m$.
4. *Parts with multiple diameter values, none unique, and an asymmetric quasiperiodic push-diameter function*: An asymmetric quasiperiodic push-diameter function has some indistinguishable states that are not periodically spaced. The worst case occurs when there is only one state $s_a$ that differs from its indistinguishable states in having an asymmetric stable orientation, and $s_a$ belongs to an $m$-indistinguishable set $M_a$. Let the part initially be in a different $m$-indistinguishable set. First execute the smallest action that causes states to transition to a subset of $M_a$ that includes $s_a$. Since the action range of $s_a$ differs from other states in $M_a$, execute the smallest action that can distinguish it. If the part was in $s_a$, its orientation is now determined. Else in the worst case, we have $(m - 1)$ remaining indistinguishable states. We recursively continue eliminating states from consideration by this two step process. For $m$ initial states, a maximum of $1 + 2(m - 1)$, or $2m - 1$ steps will orient the part.
5. *Parts with multiple diameter values, a symmetric push-diameter function, and an asymmetric push function*: Having a symmetric push-diameter function and asymmetric push function implies there is at least one pair of neighboring indistinguishable stable states, the asymmetric pair, whose action ranges differ from a corresponding pair of states an integer number of periods away. The asymmetric pair may belong to a larger set of adjacent indistinguishable states, and belongs to an indistinguishable set $M_a$. Let the number of periods of the push-diameter function be $p$. Assume the part is initially in $M_a$. Perform the smallest action for a subset of states in $M_a$ to transition to another indistinguishable set. This reduces the indistinguishable set by $p$ states. For the states not in $M_a$, execute the smallest action to transition to the subset of $M_a$ that includes the asymmetric pair such that the state in the asymmetric pair is "out of phase" with the states in other corresponding pairs. Execute a second action to distinguish this asymmetric state from the other states. These two actions reduce the indistinguishable set by at least one state. Applying this sequence of actions repeatedly gives us a plan with a maximum length of $m$ steps.
6. *Parts with multiple diameter values, none unique, and symmetric push-diameter and push functions*: Let $p$ be the number of periods of the push-diameter function. This is equivalent to having a push diameter function of period $2\pi/p$ with $1/p$ the number of states. By the pre-

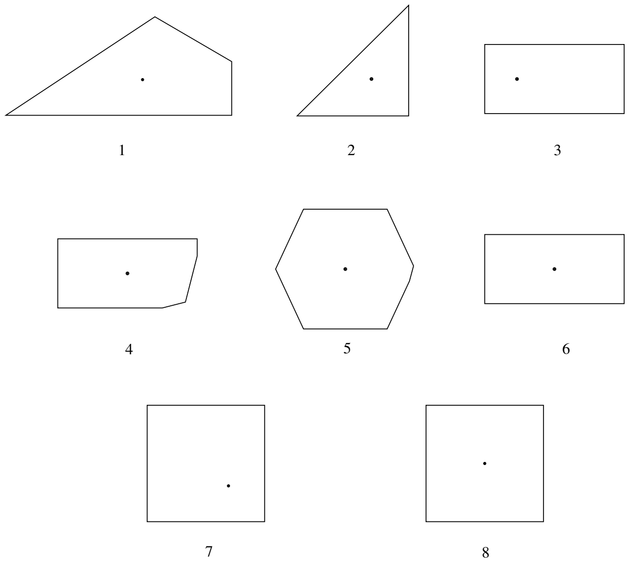


**Figure 9:** Polygons with different worst-case plan length characteristics.

vious analysis, the maximum number of steps to orient the part up to symmetry is $2m/p - 1$.

7. *Parts with all states with the same diameter value and an asymmetric push function*: This is the sensorless orienting case with a maximum plan length of $2n - 1$ steps (Chen and Ierardi [9]).
8. *Parts with all states with the same diameter value and a symmetric push function*: This is sensorless orienting where the part can be oriented up to symmetry with a maximum of $2n/p - 1$ steps, where $p$ is the number of periods of the push function.

### 5.2 Bottom-to-Top Algorithm

The bottom-to-top algorithm returns the minimum length sensor-based plan to orient a part. It begins by finding the best representative actions to distinguish all possible 2-indistinguishable sets, where the best action is the one that minimizes the maximum length over the resulting child indistinguishable sets. The length associated with an indistinguishable set is the worst-case number of actions to distinguish it, and is one plus the maximum length of its child indistinguishable sets from its best action. It then finds the best actions for all possible 3-indistinguishable sets. It sequentially works on larger indistinguishable sets, avoiding actions that lead to indistinguishable sets of the same size, until it finds the best actions for all the indistinguishable sets. At each level the algorithm has available the best actions and lengths for all indistinguishable sets of smaller size. It creates a plan by selecting the best sequence of actions to distinguish each initial indistinguishable set.

For some polygons, every $k$-indistinguishable set may not have an action that results in smaller indistinguishable sets. If no $k$-indistinguishable set has an action that leads to smaller indistinguishable sets, the part is orientable only up to symmetry. If one or more $k$-indistinguishable sets have actions that lead to smaller indistinguishable sets, the part can be brought to a distinct orientation. In such cases, we must perform a second pass of selecting actions to apply to the unresolved indistinguishable sets. The time to identify the best actions over all subsets over all sensor values is $\sum_i O(n^2 n_i^2 2^{n_i})$, where $n_i$ states share the $i$th diameter value, and the part has $n$ stable states. Therefore the time complexity of the algorithm is $O(n^3 m^2 2^m)$. Planning is performed off-line only once for each part.

The bottom-to-top algorithm is *complete* in that it finds a plan to orient the part when one exists or returns a plan up to symmetry otherwise. For each indistinguishable set, the representative actions cover the action space. The algorithm applies all representative actions of an indistinguishable set to it, and the second pass ensures the solution is found.

## 6 Orienting Multiple Part Shapes

We consider orienting any part from a known set of parts using the same plan, as may occur during product changeovers or when parts with different shapes have to be oriented on the same line. We first show that unlike the radius function, the push function and push-diameter function do not uniquely define the shape of a convex polygon.

**Theorem 1** *Two convex polygons have the same radius function if and only if they have the same shape.*

**Proof:** See (Akella [1]). □

**Theorem 2** *There is an infinite set of non-similar convex polygons having the same push function.*

**Proof:** Any convex polygon shares the same push function with an infinite set of similar polygons. We first prove that for any convex polygon, we can find another non-similar convex polygon with the same push function by creating a new transition vertex from any transition vertex of the given polygon. A *transition vertex* is one at which a local maximum of the radius function occurs. Consider a transition vertex $V$ with incident edges $e_i$ and $e_j$ defined by $VE_i$ and $VE_j$ (Figure 10). Draw perpendiculars to $e_i$ and $e_j$ from the center of mass $C$ to obtain points $P_i$ and $P_j$ respectively, which can lie in the interior or exterior of the edges. A local maximum of the radius function occurs at $V$ when the supporting line is perpendicular to $VC$. Call this line $l_\perp$. We splice in two new unstable edges that meet at a new transition vertex. Pick $Q_i$ in the interior of both $VP_i$ and $VE_i$. Draw a perpendicular to $CQ_i$ through $Q_i$ to intersect $VC$ at $R_i$. Any line segment with one endpoint in the interior of $VR_i$ and $Q_i$ as the other endpoint forms an unstable edge. Similarly find $Q_j$ and $R_j$ for $e_j$. Pick $T$ in the intersection of $VR_i$ and $VR_j$ to be the new transition vertex. $TQ_i$ and $TQ_j$ identify unstable edges $u_i$ and $u_j$ whose addition does not change the stable orientations.

To prove the divergent orientation of the push function does not change, we must prove that a local radius maximum occurs at $T$ when the supporting line is $l'_\perp$ perpendicular to $CT$. We must show that length$(CT)$ is greater than the radius values of $u_i$ and $u_j$, and that they are greater than the radius values of $e_i$ and $e_j$ respectively. Let the radius values of $e_i$ and $u_i$ be $r_i$ and $r_{u_i}$. Consider a line rotating about $Q_i$ with its initial orientation along $Q_iR_i$. The radius of the line is proportional to the cosine of its rotation angle, with the maximum at its initial orientation. Therefore length$(CQ_i) > r_{u_i} > r_i$. From $\triangle CQ_iT$, length$(CT) >$ length$(CQ_i)$, which implies length$(CT) > r_{u_i} > r_i$. Similarly, length$(CT) > r_{u_j} > r_j$. So $T$ is a transition vertex. The new polygon with vertices $Q_i$, $T$, and $Q_j$ replacing vertex $V$ has the same push function as the original polygon. Since there is an infinite set of choices for $Q_i$, $Q_j$, and $T$,

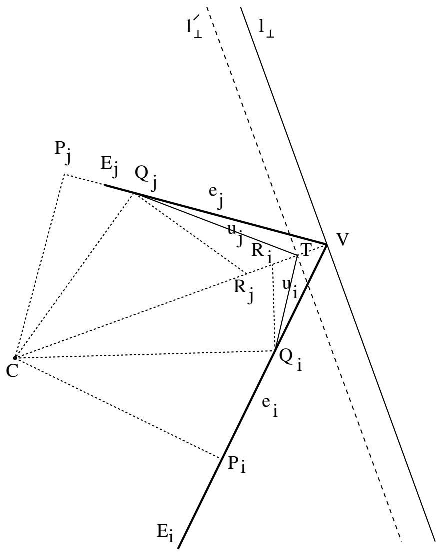


**Figure 10:** Generating a new polygon with the same push function as the given polygon by splicing in unstable edges $u_i$ and $u_j$. Edges $e_i$ and $e_j$ of the original polygon are drawn bold.

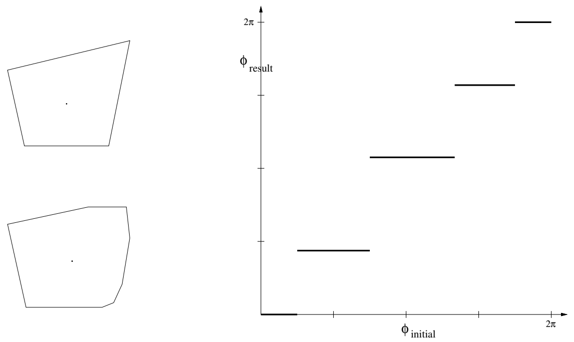


**Figure 11:** Two parts with the same push function.

we can generate an infinite set of polygons having the same push function. See Figure 11 for an example. □

**Theorem 3** *There is an infinite set of convex polygons having the same push-diameter function.*

**Proof:** Two parts have the same push-diameter function if their stable edge orientations and diameters, and their divergent orientations are identical. Our constructive proof modifies one or more transition and diameter determining vertices of the given polygon. A *diameter determining vertex* of a stable edge is the vertex with the largest perpendicular distance to the edge. Each stable edge of a convex polygon has one or two such vertices. Every polygon has at least one transition vertex. For the proof, we classify polygons as follows.

1. *Polygons with at least one transition vertex that is not a diameter determining vertex.* We can use the construction of Theorem 2 to modify this vertex without changing any stable edge diameters. Thus we can generate an infinite set of polygons with the same push function and push-diameter function as the original polygon.

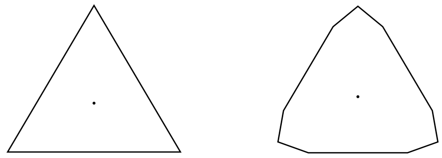

**Figure 12:** A regular polygon modified to generate another polygon with the same push-diameter function.

2. *Polygons whose transition vertices are all diameter determining vertices, with every stable edge having a single diameter determining vertex.* All diameter determining vertices may not be transition vertices, and all polygon edges may not be stable. Given a non-transition vertex, we can splice in new unstable edges and find a new non-transition vertex without modifying the push function, using constructions similar to that in Theorem 2. Modify every diameter determining vertex to reduce the diameter of each stable edge by the same percentage, with the construction for each vertex depending on whether or not it is a transition vertex. Expand this modified polygon so it has the same push-diameter function as the original polygon. We can thus generate an infinite set of polygons with the same push-diameter function. See Figure 12.

3. *Polygons whose transition vertices are all diameter determining vertices, with at least one stable edge having two diameter determining vertices.* Since the diameter determining vertices of the chosen stable edge $e_s$ are equidistant to it, they are the vertices of a parallel edge $e_d$. When either of the diameter determining vertices determines the diameter of only $e_s$, it can be modified without changing the push-diameter function. Consider the case when the vertices are diameter determining for more than one edge. The clockwise neighboring edges of $e_s$ and $e_d$ form a neighboring pair, and the counter-clockwise neighboring edges form a neighboring pair. For a given neighboring edge pair, let the interior angle between $e_s$ and its neighbor from the edge pair be $\sigma$, and the interior angle between $e_d$ and its neighbor from the edge pair be $\delta$. We can generate an infinite set of polygons with the same push-diameter function for the two cases that can occur.

   *All neighboring edge pairs have parallel edges:* There may be one or two pairs of neighboring parallel edges. Select a diameter determining vertex belonging to one of the pairs of neighboring parallel edges. This vertex can be modified without altering the push function, using the appropriate construction. This vertex is either the diameter determining vertex for only $e_s$, or for $e_s$ and its neighboring edge from the selected pair, if it is stable. Since both edges have two diameter determining vertices, the push-diameter function is also not altered. See Figure 13 for an example.

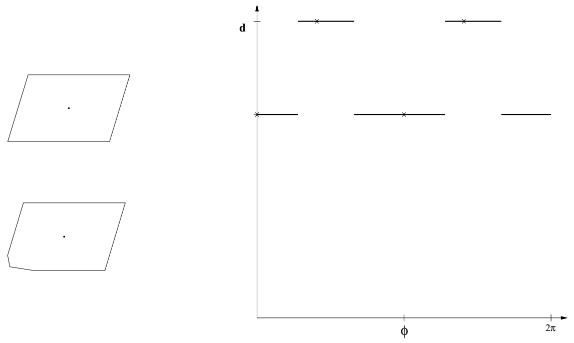


**Figure 13:** Two parts with the same push-diameter function. Their push functions are also identical.

> *At least one neighboring edge pair has non-parallel edges:* Select the neighboring pair with non-parallel edges. If both pairs have non-parallel edges, select the pair with the smaller $(\sigma - \delta)$ value. If $\sigma < \delta$, select the diameter determining vertex that belongs to $e_d$ and its neighboring edge from the pair. From the geometry of the convex polygon, this vertex can be a diameter determining vertex for only $e_s$ and can be modified without changing the push and push-diameter functions. If $\sigma > \delta$, select the vertex belonging to $e_s$ and its neighbor from the selected pair. From the polygon geometry, it is either not a diameter determining vertex or a diameter determining vertex for only edge $e_d$ if it is stable. In either case, we can modify it without changing the push-diameter function.

Hence there is an infinite set of polygons with the same push-diameter function as the original polygon. □

### 6.1 Part Orientability and Recognizability

We identify conditions for a set of parts to be orientable and recognizable by first considering pairs of parts. Two parts are **distinguishable** if there exists an action sequence such that the output strings of stable orientation diameter values for the two parts differ.

**Lemma 4 (Akella [1])** *Two parts with identical push-diameter functions are orientable but not distinguishable.*

**Theorem 5** *Two parts with different push-diameter functions are orientable and distinguishable if $|D| > 1$, where $D$ is the set of distinct diameter values of all stable orientations of the two parts.*

**Proof:** When $|D| > 1$, there are multiple distinct diameter values. When one or both parts have some diameter values unique to them, we can distinguish the parts by bringing them to the corresponding stable orientations. When the two parts have the same set of multiple diameter values, there is at least one pair of states whose action ranges differ since the push-diameter functions differ. It is most difficult to distinguish parts when they have almost identical push-diameter

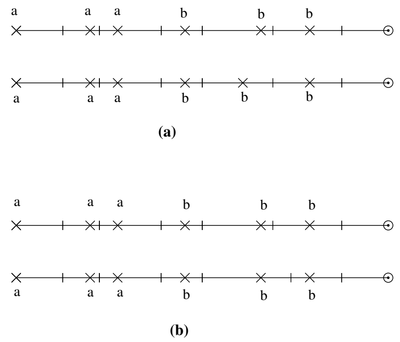


**Figure 14:** Resting ranges of two parts with almost identical push-diameter functions. Indistinguishable states have the same character. (a) Identical resting ranges, stable orientations differ. (b) Resting ranges differ, identical stable orientations.

functions. Without loss of generality, we assume each part has the same two diameter values (Figure 14). Every state in one part is paired with a corresponding state in the other. There is some action for which the parts transition from the pair of states with differing action ranges to two states that are not paired. This action causes the parts to go "out of phase," and there is then an action sequence guaranteed to bring the two parts to distinguishable states. □

A set of known parts $P$ is said to be **recognizable** if there exists an action sequence to distinguish every member of the set from every other member of the set.

**Theorem 6** *For a known set of parts $P$, if every pair of parts is orientable and distinguishable, then any subset of parts $S \subseteq P$ is orientable and recognizable.*

**Proof:** Since every pair of parts can be oriented and distinguished, no two parts have the same push-diameter function. Therefore each part has a state whose action range differs from at least one state in every other part. Also every pair of parts has multiple distinct diameter values. We can show an action sequence always exists to rotate parts "out of phase" with the others one at time and use this to identify them. □

### 6.2 Planning for Multiple Parts

We extended the bottom-to-top algorithm to generate orienting plans for multiple parts (Figure 15). A state is additionally encoded by the part it belongs to, and indistinguishable sets consist of indistinguishable states over all parts. Multiple parts can also be oriented by sensorless strategies. By Theorem 2, an infinite set of parts can have the same push function, and they can all be oriented by the same plan. We cannot recognize the parts however. For parts with different push functions, a breadth-first search planner can be used to find a sensorless plan when one exists.

## 7 Implementation

We implemented the bottom-to-top algorithm in Common Lisp to generate orienting plans. Given a part shape, the

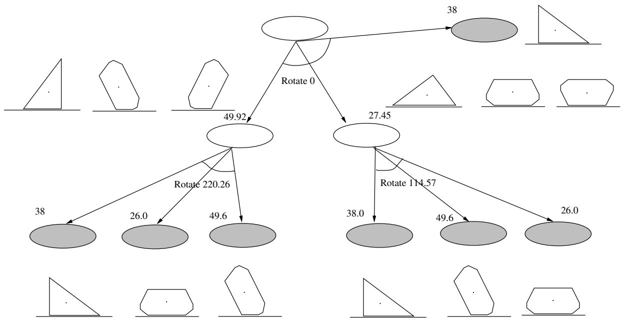


**Figure 15:** A sensor-based plan to orient two part shapes. It was generated by a modified version of the bottom-to-top planner.

planner returns a plan to orient the part uniquely when possible, and up to symmetry otherwise. The planner took an average of 0.101 secs on a Sparc ELC on a test set of parts. We also implemented a sensorless planner that uses breadth-first search. We demonstrated sensor-based and sensorless plans using an Adept 550 robot and a conveyor. We ran 20 trials of a sensor-based plan and 10 trials of a sensorless plan on the two parts of Figure 15. For the triangle, 17 of 20 trials with the sensor-based plan and 8 of 10 trials with the sensorless plan succeeded. For the 8-gon, 19 of 20 trials with the sensor-based plan and all 10 trials of the sensorless plan succeeded. We also ran 20 trials, 10 on each part, of the plan in Figure 15. The triangle was successfully oriented and recognized all 10 times and the 8-gon was successfully oriented and recognized 9 times. All failures occurred when the suction cup made insufficient contact to pick up the part.

## 8 Conclusion

In this paper we coupled simple sensing operations with mechanical operations to orient parts. We showed that simple sensors reduce the number of orienting steps from $O(n)$ to $O(m)$. This can significantly reduce execution time and the number of stages in a pipelined setup. We identified conditions under which sensors allow a feeder to orient and recognize multiple part shapes with a single plan. These results can be applied to similar tasks such as orienting by parallel-jaw grasping (Rao and Goldberg [17]) and orienting with actuated fences (Akella *et al.* [2]). Future extensions include treating a broader class of part shapes and developing planners that minimize the expected plan length.

## Acknowledgments

Funding was provided by NSF under grants IRI-9213993 and IRI-9318496 (with ARPA).

## References

[1] S. Akella. *Robotic Manipulation for Parts Transfer and Orienting: Mechanics, Planning, and Shape Uncertainty*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Dec. 1996. Robotics Institute Technical Report CMU-RI-TR-96-38.

[2] S. Akella, W. H. Huang, K. M. Lynch, and M. T. Mason. Sensorless parts orienting with a one-joint manipulator. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, Apr. 1997.

[3] K. F. Böhringer, V. Bhatt, and K. Y. Goldberg. Sensorless manipulation using transverse vibrations of a plate. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 1995.

[4] G. Boothroyd, C. Poli, and L. E. Murch. *Automatic Assembly*. Marcel Dekker, 1982.

[5] R. C. Brost. Automatic grasp planning in the presence of uncertainty. *International Journal of Robotics Research*, 7(1):3–17, Feb. 1988.

[6] R. C. Brost. Dynamic analysis of planar manipulation tasks. In *IEEE International Conference on Robotics and Automation*, Nice, France, May 1992.

[7] M. Caine. The design of shape interactions using motion constraints. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.

[8] J. F. Canny and K. Y. Goldberg. RISC for industrial robotics: Recent results and open problems. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.

[9] Y.-B. Chen and D. J. Ierardi. The complexity of oblivious plans for orienting and distinguishing polygonal parts. *Algorithmica*, 14:367–397, 1995.

[10] M. Erdmann and M. T. Mason. An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4(4):369–379, Aug. 1988.

[11] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2/3/4):201–225, 1993.

[12] D. D. Grossman and M. W. Blasgen. Orienting mechanical parts by computer-controlled manipulator. *IEEE Transactions on Systems, Man, and Cybernetics*, 5(5), September 1975.

[13] J. Krishnasamy, M. J. Jakiela, and D. E. Whitney. Mechanics of vibration-assisted entrapment with application to design. In *IEEE International Conference on Robotics and Automation*, Apr. 1996.

[14] M. Mani and W. R. D. Wilson. A programmable orienting system for flat parts. In *North American Manufacturing Research Institute Conference XIII*, Berkeley, California, May 1985.

[15] M. T. Mason. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research*, 5(3):53–71, 1986.

[16] M. A. Peshkin and A. C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5):524–531, Oct. 1988.

[17] A. S. Rao and K. Y. Goldberg. Shape from diameter: Recognizing polygonal parts with a parallel-jaw gripper. *International Journal of Robotics Research*, 13(1):16–37, Feb. 1994.

[18] M. Shirai and A. Saito. Parts supply in Sony's general-purpose assembly system "SMART". *Jpn. J. Adv. Automation Tech.*, 1:108–111, 1989.

[19] R. H. Taylor, M. T. Mason, and K. Y. Goldberg. Sensor-based manipulation planning as a game with nature. In R. C. Bolles and B. Roth, editors, *Robotics Research: The Fourth International Symposium*, pages 421–429, Cambridge, MA, 1988. MIT Press.

[20] A. F. van der Stappen, K. Y. Goldberg, and M. Overmars. Geometric eccentricity and the complexity of manipulation plans. *Algorithmica*, Accepted Nov. 1997.

[21] J. Wiegley, K. Y. Goldberg, M. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. In *IEEE International Conference on Robotics and Automation*, Apr. 1996.

[22] N. B. Zumel and M. A. Erdmann. Nonprehensile manipulation for orienting parts in the plane. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, Apr. 1997.