Orienting Polyhedral Parts by Pushing

Robert-Paul Berretty*† Mark H. Overmars*
A. Frank van der Stappen*

Abstract

A common task in automated manufacturing processes is to orient parts prior to assembly. We consider sensorless orientation of an asymmetric polyhedral part by a sequence of push actions, and show that is it possible to move any such part from an unknown initial orientation into a known final orientation if these actions are performed by a jaw consisting of two orthogonal planes. We also show how to compute an orienting sequence of push actions.

We propose a three-dimensional generalization of conveyor belts with fences consisting of a sequence of tilted plates with curved tips; each of the plates contains a sequence of fences. We show that it is possible to compute a set-up of plates and fences for any given asymmetric polyhedral part such that the part gets oriented on its descent along plates and fences.

1 Introduction

An important task in automated assembly is orienting parts prior to assembly. A part feeder takes in a stream of identical parts in arbitrary orientations and outputs them in a uniform orientation. Usually, part feeders use data obtained from some kind of sensing device to accomplish their task. We consider the problem of *sensorless orientation* of parts, in which no sensors but only knowledge of the geometry of the part is used to orient it from an unknown initial orientation to a unique final orientation. In sensorless manipulation, parts are positioned or oriented using open-loop actions which rely on passive mechanical compliance.

A widely used sensorless part feeder in industrial environments is the bowl feeder [9, 8]. Among the sensorless part feeders considered in the scientific literature are the parallel-jaw gripper [12, 17], the single pushing jaw [3, 18, 19, 21], the conveyor belt with a sequence of (stationary) fences placed along its sides [10, 22, 26], the conveyor belt with a single rotational fence [1, 2], the tilting tray [16, 20], and vibratory plates and programmable vector fields [6, 7].

*Institute of Information and Computing Sciences, Utrecht University, PO Box 80089, 3508 TB Utrecht, The Netherlands.

†Supported by the Dutch Organization for Scientific Research (N.W.O.)

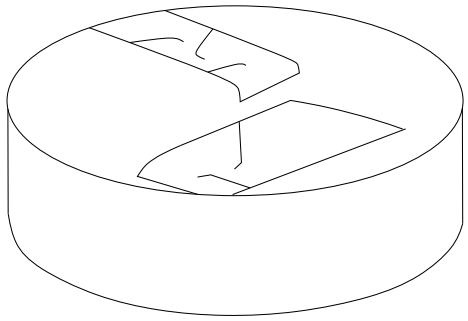


Figure 1: The three-dimensional part feeder. Plates with fences mounted to a cylinder.

Scientific literature advocates a risc (Reduced Intricacy in Sensing and Control) approach to designing manipulation systems for factory environments. These systems benefit from their simple design and do not require guru programming skills [11]. The pushing jaw [3, 18, 19, 21] orients a part by an alternating sequence of pushes and jaw reorientations. The objective of sensorless orientation by a pushing jaw is to find a sequence of push directions that will move the part from an arbitrary initial orientation into a single known final orientation. Such a sequence is referred to as a *push plan.* Goldberg [17] showed that any polygonal part can be oriented by a sequence of pushes. Chen and Ierardi [12] proved that any polygonal part with n vertices can be oriented by $O(n)$ pushes. They showed that this bound is tight by constructing (pathological) n-gons that require $\Omega(n)$ pushes to be oriented. Eppstein [15] observed that for a special class of feeders, polynomial planning algorithms exist. Goldberg gave an algorithm for computing the shortest push plan for a polygon. His algorithm runs in $O(n^2)$ time. Berretty *et al.*[5] gave an algorithm for computing the shortest sequence of fences over a conveyor belt that orients a part as it slides along the fences. The algorithm runs in $O(n^3 \log n)$ time.

The drawback of the majority of the achievements in the field of sensorless orientation is that they only apply to flat, two-dimensional parts, or to parts where the face the part rests on is known beforehand. In this paper, we narrow the gap between industrial feeders and the scientific work on sensorless orientation, by introducing a feeder which orients three-dimensional parts up to rotational symmetry. This is the first device which can be proven to correctly feed three-dimensional parts. The device we use is a cylinder with plates tilted toward the interior of the cylinder attached to the side. Across the plates, there are fences. The part cascades down from plate to plate, and slides along the fences as it travels down a plate. A picture of the feeder is given in Figure 1. The goal of this paper is to compute the set-up of plates and fences that is guaranteed to move a given asymmetric polyhedral part towards a unique final orientation. Such a set-up, consisting of a sequence of plate slopes, and for each

plate a sequence of fence orientations is referred to as a *(plate and fence) design.*

The alignment of the part with plates and fences strongly resembles the alignment of that part with a jaw consisting of two orthogonal planes: finding a design of plates and fences corresponds to finding a constrained sequence of push directions for the jaw. This relation motivates our study of the behavior of a part pushed by this jaw. We show that a three-dimensional polyhedral part P can be oriented up to rotational symmetry by a (particular) sequence of push actions, or push plan for short, of length $O(n^2)$, where n is the number of vertices of P. Furthermore, we give an $O(n^3 \log n)$ time algorithm to compute such a push plan. We shall show how to transform this three-dimensional push plan to a three-dimensional design. The resulting design consists of $O(n^3)$ plates and fences, and can be computed in $O(n^4 \log n)$ time.

The paper is organized as follows. We first discuss the device we use to orient parts, introduce the corresponding jaw, and study the behavior of a part being pushed in Section 2. We then show, in Section 3, that the jaw can orient any given polyhedral part up to symmetry. In Section 4 we show how to compute a sequence of push actions to orient a given part. In Section 5 we show how the results for the generic jaw carry over to the cylinder with plates and fences. In Section 6, we conclude and pose several open problems.

2 Pushing Parts

A polyhedral part in three-dimensional space has three rotational degrees of freedom. There are numerous ways to represent orientations and rotations of objects in the three-dimensional world. We assume that a fixed reference frame is attached to P. We denote the orientation of P relative to this reference frame by $\sigma = (\phi, \psi, \theta)$, where (ϕ, ψ) are the polar coordinates of a point on the sphere of directions, and θ the roll, which is a rotation about the ray emanating from origin, intersecting (ϕ, ψ). See Figure 2 for a picture. This representation will be shown to be appropriate considering the rotational behavior of the part as it aligns to our feeder. We discuss our feeder in Section 2.1. The rotational behavior of P in contact with the feeder is discussed in Section 2.2.

2.1 Modeling the feeder

A part in three-dimensional space can have infinitely many orientations. The device we use to orient this part discretizes the set of possible orientations of the part. The feeder consists of a series of bent plates along which the part cascades down. Across a plate, there are fences which brush against the part as it slides down the plate. A picture of a part sliding down a plate is given in Figure 3(a). The plate on which the part slides discretizes the first two degrees of freedom of rotation of the part. A part in alignment with a plate retains one undiscretized rotational degree of freedom. The rotation of the part is determined up to its roll, i.e. the rotation about the axis perpendicular to the plate. The fences, which are mounted across the plates, push the part from the side, and discretize

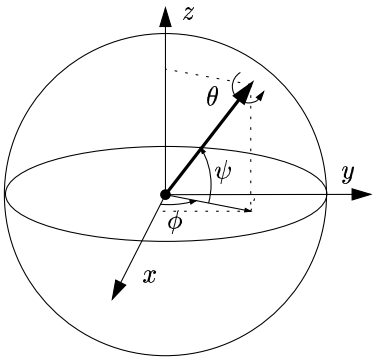


Figure 2: The rotation is specified by a point (ϕ, ψ) on the sphere of directions, and a rotation θ about the vector through this point.

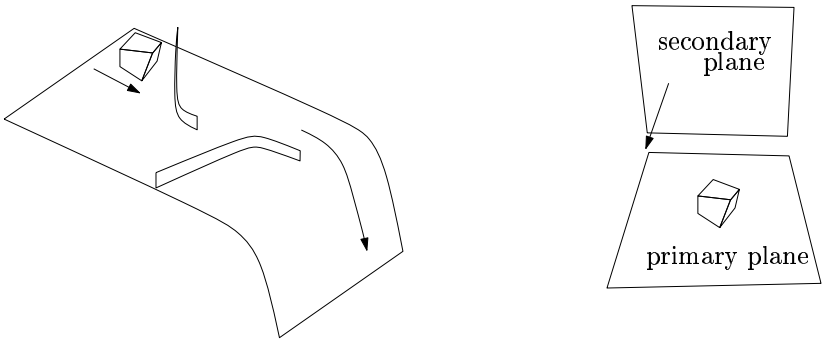


Figure 3: (a) A part sliding down a plate with fences. (b) The same part on the jaw.

the roll of its rotation. We assume that P first settles on the plate before it reaches the fences which are mounted across the plate, and there is only rotation about the roll axis as the fences brush the part.

We look at the push actions of the plates and the fences in a more general setting. We generalize the cylindrical feeder by substituting the plate along which the part slides by a plane on which the part rests. We substitute the fences by an orthogonal second plane, which pushes the part from the side. We call the planes the primary and secondary (pushing) plane, respectively. A picture of the resulting jaw is given in Figure 3(b).

Since the planes can only touch P at its convex hull, we assume without loss of generality that P is convex. We assume that the center-of-mass of P, denoted by c, is inside the interior of P. Analogously to the cylindrical feeder,

we assume that only after P has aligned with the primary plane, we apply the secondary plane. As the part rests on the primary plane, the secondary plane pushes P at its orthogonal projection onto the primary plane. We assume that the feature on which P rests retains contact with the primary plane as the secondary plane touches P. We assume that for any equilibrium orientation, which is an orientation for which P rests on the jaw (see Section 2.2 for a definition of an equilibrium orientation), the projection of P onto the primary plane has no rotational symmetry. We refer to a part with this property as being asymmetric.

In order to be able to approach the part from any direction, we make the (obviously unrealistic) assumption that the part floats in the air, and assume that we can control some kind of gravitational field which attracts the part in a direction towards the jaw. Also, we assume that the part quasi-statically aligns with the jaw, i.e. we ignore inertia. Studying this unrealistic situation is useful for analyzing our feeder later.

In order to be able to determine a sequence of push directions that orients P, we need to understand the rotational behavior of P when pushed by the jaw. We analyze this behavior below.

2.2 The push function

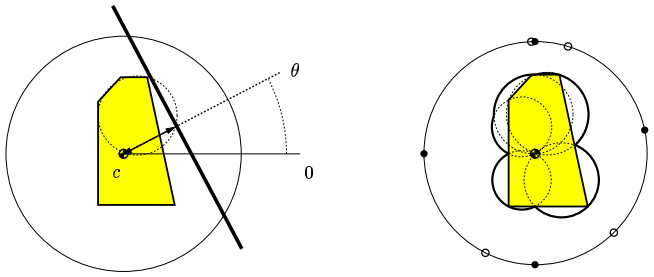
A basic action of the jaw consists of directing and applying the jaw. The result of a basic action for a part in its reference orientation is given by the *push function*. The push function $\varpi : [0, 2\pi) \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, 2\pi) \rightarrow [0, 2\pi) \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, 2\pi)$ maps a push direction of the jaw relative to P in its reference orientation onto the orientation of P after alignment with the jaw. The orientation of P after a basic action for a different initial orientation than its reference orientation is equal to the push function for the push direction plus the offset between the reference and the actual initial orientation of P.

We dedicate the next three subsections to the discussion of the push function for P in its reference orientation. As P aligns with the device, we identify two subsequent stages; namely alignment with the primary plane, and alignment with the secondary plane.

Since we assume that we apply the secondary plane only after the part has aligned with the primary pushing plane, we shall separately discuss the rotational behavior of the part during the two stages. In the next two subsections we discuss the first stage of alignment. The last subsection is devoted to the second stage of alignment.

2.2.1 Alignment with the primary plane

The part P will start to rotate when pushed unless the normal to the primary plane at the point of contact passes through the center-of-mass of P [19]. We refer to the corresponding direction of the contact normal as an *equilibrium* contact direction or orientation.



(a) (b)

Figure 4: (a) The radius for contact direction θ. (b) The bold curves show the radius function for a planar part P. The dots depict local minima, the circles local maxima of the radius function.

The contact direction of a supporting plane of P is uniquely defined as the direction of the normal of the plane pointing into P. We study the *radius function* of the part, in order to explain the alignment of P with the primary plane. The radius function $r : ([0, 2\pi) \times [-\frac{\pi}{2}, \frac{\pi}{2}]) \rightarrow \mathbb{R}^+$ maps a direction (ϕ, ψ) onto the distance from c to the supporting plane of P with contact direction (ϕ, ψ).

We first study the planar radius function for a planar part P_p with center-of-mass c_p. The planar radius function easily generalizes to the radius function for a three-dimensional part. The planar radius function $r_p : [0, 2\pi) \rightarrow \mathbb{R}^+$ maps a direction θ onto the distance from c to the supporting line of P_p with contact direction θ, see Figure 4(a). With the aid of elementary trigonometry, we derive that the distance of c to the supporting line of P_p in contact with a fixed vertex v for contact direction θ equals the distance of c to the intersection of the ray emanating from c in direction θ and the boundary of the disc with diameter (c, v). Combining the discs for all vertices of P_p gives a geometric method to derive r_p. The radius $r_p(\theta)$ is the distance of c to an intersection of the ray emanating from c in direction θ and the boundary of a disc through a vertex of P_p. If there are multiple discs intersecting the ray, $r_p(\theta)$ equals the maximum of all distances from c to the intersection with any disc—a smaller value would not define the distance of a supporting line of P, but rather line intersecting P. In conclusion, $r_p(\theta)$ equals the distance from c to the intersection of the boundary of union of discs for each vertex of P with the ray emanating from c in direction θ. In Figure 4(b), we show a planar part with for each vertex v, the disc with diameter (c, v). The boundary of the discs is drawn in bold.

The three-dimensional generalization of a disc with diameter (c, v) is a ball with with diameter (c, v). The three-dimensional radius function $r(\phi, \psi)$ is the distance of c to the intersection of the ray emanating from c in direction (ϕ, ψ) with the union of the set of balls for each vertex of P. We call the boundary of the union the *radius terrain*; it links every contact direction of the primary

plane to a unique distance to c.

The radius terrain contains maxima, minima, and saddle points. If the contact direction of the primary plane corresponds to a local extremum, or saddle point of the radius function, the part is at an equilibrium orientation, and the contact direction of the primary plane remains unchanged. If, on the other hand, the radius function of the part for a contact direction of the primary plane is not a local extremum, or saddle point, the gravitational force will move the center-of-mass closer to the primary plane, and the contact direction will change. We assume that, in this case, the contact direction traces a path of steepest descent in the radius terrain until it reaches a equilibrium contact direction. In general, the part can pivot along different features of the part, as the contact direction follows the path of steepest descent towards an equilibrium.

Different types of contact of the primary plane correspond to different features of the radius terrain. The contact directions of the primary plane with a vertex of P define a (spherical) patch in the terrain, the contact directions of the primary plane with an edge of P define an arc, and the contact direction of the primary plane with a face of P defines a vertex. In Figure 5, we show different types of contacts of P with the primary plane. Figure 5(a) shows an equilibrium contact direction with the primary plane in contact with vertex v_1 of P. The contact direction corresponds to a maximum in the radius terrain. Figure 5(b) shows a vertex contact which is not an equilibrium. Figure 5(c) shows an equilibrium contact direction for edge (v_3, v_4) of P. Figure 5(d) shows a non-equilibrium contact for edge (v_5, v_6). In Figure 5(e) we see a degenerated non-equilibrium contact for edge (v_7, v_8), which actually corresponds to a non-equilibrium vertex contact with the primary plane in contact with vertex v_8. The direction of steepest descent in the radius terrain corresponds to a rotation about v_8. Figure 5(f) shows a stable equilibrium face contact. The contact direction corresponds to a local minimum of the radius terrain. In Figure 5(g) we see a degenerated face contact which corresponds to an edge contact for edge (v_{18}, v_{19}) of P. Figure 5(h) shows a degenerated face contact which corresponds to a vertex contact for vertex v_{15}.

The alignment of the part to the primary plane is a concatenation of simple rotations, i.e. a rotation about a single vertex or edge. The path of a simple rotation in the radius terrain is either a great arc on a balls with radius (c, v) for a vertex of P, or a part of a intersection of two balls (c, v_1), (c, v_2) for two vertices which is a part of the boundary of a disc. It is easy to see that the projection of the arcs in the radius terrain of any of the simple rotations project to great arcs on the sphere of directions. Hence, during a simple rotation, the contact direction of the primary plane traces a great arc on the sphere of contact directions. During each single stage of alignment, we assume that there is no (instantaneous) rotation about the roll axis.

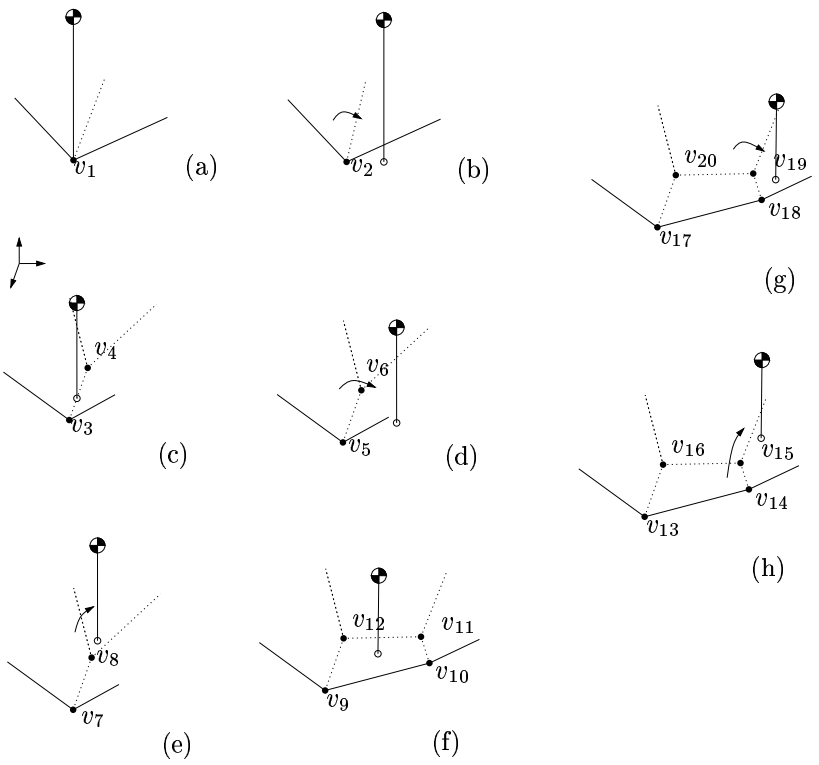


Figure 5: Different contacts for the primary plane, with a projection of c onto the primary plane. The primary plane is assumed at the bottom of the pictures.

2.2.2 Computation of the roll after alignment with the primary plane

The mapping of Section 2.2.1 only tells us which feature of the part will be in contact with the primary plane after rotation. It leaves the change in the part's roll out of consideration. Nevertheless, we need to keep track of the roll as P aligns with the primary plane. We remember that the alignment with the primary plane is a concatenation of simple rotations each corresponding to a great arc on the sphere of contact directions of the primary plane.

With the aid of spherical trigonometry, it is possible to compute the change in roll caused by a reorientation of the primary plane (prior to pushing). Subsequently, we can compute the change in roll for a simple rotation of P. Since the alignment of the part can be regarded as a concatenation of such simple rotations, we obtain the final roll by repeatedly applying the change in the roll of P for each simple rotation in the alignment to the primary plane.

2.2.3 Alignment with the secondary plane

Let us assume that P is in equilibrium contact with the primary plane. The next step in the application of the jaw is a push operation of the secondary (orthogonal) plane. The push action by the secondary plane changes the orientation of the projection of P onto the primary plane. The application of the secondary plane to the part can, therefore, be regarded as a push operation on the two-dimensional orthogonal projection of P onto the primary plane.

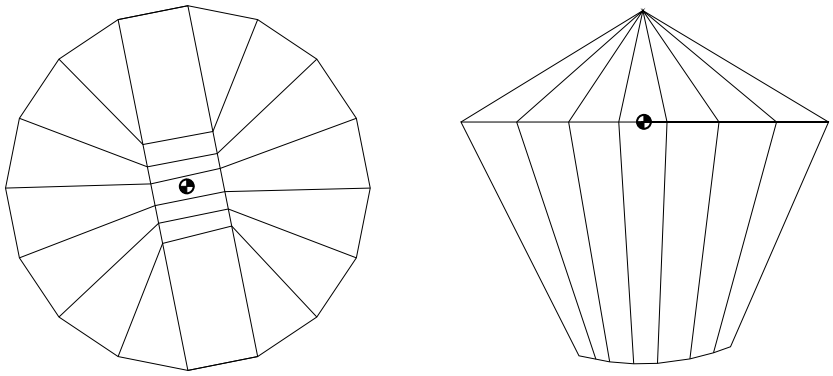
The *planar push function* for a planar projection of P $\varpi_{\text{proj}} : [0, 2\pi) \rightarrow [0, 2\pi)$ links every orientation θ to the orientation $\varpi_{\text{proj}}(\theta)$ in which the part P_{proj} settles after being pushed by a jaw with initial contact direction θ (relative to the frame attached to P_{proj}). The rotation of the part due to pushing causes the contact direction of the jaw to change. The final orientation $\varpi_{\text{proj}}(\theta)$ of the part is the contact direction of the jaw after the part has settled. The equilibrium push directions are the fixed points of ϖ_{proj}.

Summarizing, we can compute the orientation of P after application of the jaw. In the next section, we shall show we can always orient P up to symmetry in the push function by means of applications of the jaw.

3 Orienting a polyhedral part

In this section we will show that any polyhedral part P can be oriented up to rotational symmetry in the push functions of the projections of P onto the primary plane. The part P has at most $O(n)$ equilibria with respect to the primary plane, and any projection of P onto the primary plane has $O(n)$ vertices. Hence, the total number of orientations of P compliant to the jaw is $O(n^2)$. Figure 6 shows an example of a part with $\Omega(n^2)$ possible orientations.

Lemma 1 *A polyhedral part of with n vertices has $O(n^2)$ stable orientations. This bound is tight in the worst case.*



(a) (b)

Figure 6: A part with $\Omega(n^2)$ equilibrium orientations. (a) Bottom view. (b) Side view.

From the previous section, we know that the pushing jaw rotates P towards one of its equilibrium orientations with respect to the primary plane, and the secondary plane. Let us, for a moment, assume that the contact direction (ϕ, ψ) of the primary plane is known.

We can now redirect and apply the secondary plane. We remember that we assume that applying the secondary plane has no influence on the contact direction of the primary plane. Consequently, the rotations of the part, due to applications of the secondary pushing plane, are fully captured by the planar push function of the projection of the part onto the primary plane. Chen and Ierardi [12] show that a two-dimensional part with m vertices can be oriented up to symmetry by means of planar push plan of length $O(m)$. Consequently, we can orient P in equilibrium contact with the primary plane up to symmetry in the projection of the part onto the primary plane by $O(n)$ applications of the secondary plane.

Lemma 2 *Let P be an asymmetric polyhedral part with n vertices. There exists a plan of length $O(n)$ that puts P into a given orientation (ϕ, ψ, θ) from any initial orientation (ϕ, ψ, θ').*

We call the operation which orients P for a single equilibrium contact direction of the primary plane (ϕ, ψ) CollideRollsSequence(ϕ, ψ). We can eliminate the uncertainty in the roll for any equilibrium contact direction of the primary plane. The initialization of the push plan that orients P reduces the number of possible orientations to $O(n)$ by a concatenation of CollideRollsSequence for all equilibrium contact directions of P. Lemma 3 will give us a push operation to further reduce the number of possible orientations.

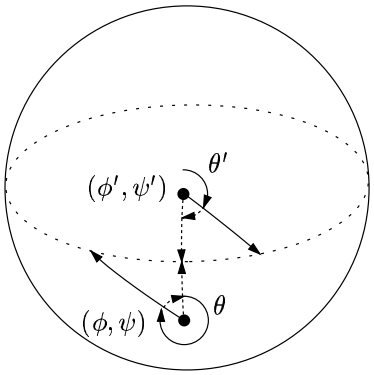


Figure 7: Two orientations on the sphere of directions. Their equator is dashed. A desired reorientation of the primary plane is dotted.

Lemma 3 *For every pair of orientations (ϕ, ψ, θ), and (ϕ', ψ', θ') of a polyhedral part there exist two antipodal reorientations of the primary plane which map these orientations onto $(\tilde{\phi}, \tilde{\psi}, \tilde{\theta})$ and $(\tilde{\phi}', \tilde{\psi}', \tilde{\theta}')$, such that $\tilde{\phi} = \tilde{\phi}'$ and $\tilde{\psi} = \tilde{\psi}'$.*

Proof: We will prove that there is a reorientation of the primary plane for which the resulting contact directions of the primary plane for P in initial orientation σ and σ' are the same. We focus on the first two parameters of the orientations σ and σ': (ϕ, ψ) and (ϕ', ψ') represent two points on the sphere of directions. We want to find a push direction that maps these two points onto another point (ϕ'', ψ''). See Figure 7. Let E denote the great circle consisting of all points equidistant to (ϕ, ψ) and (ϕ', ψ'). E divides the sphere of directions into a hemisphere containing (ϕ, ψ) and a hemisphere containing (ϕ', ψ'). Any reorientation of the primary plane corresponds maps (ϕ, ψ) and (ϕ', ψ') onto contact directions which are equidistant to these original contact directions. Let r denote the ray emanating from (ϕ, ψ), in the direction of θ, and r' denote the ray emanating from (ϕ', ψ') in the direction of θ'. Points on the rays (with equal distance δ to the origins) correspond to a reorientation of the primary pushing plane by $(0, \delta)$. Both rays intersect E. We aim for a push direction (ϕ, δ'), with δ' such that the the jaw touches P at an orientation in E. The component ϕ of the push direction changes the direction of the rays emanating from (ϕ, ψ) and (ϕ', ψ') to $\theta + \phi$ and $\theta' + \phi$ resp. We will show that there is ϕ, such that for both orientations the push direction touches the part at the same point. If their first intersection with E is in the same point, we have found a push direction which maps both orientations onto the same face. Since the orientations are in different hemispheres, increasing ϕ will move the intersections of the rays with E in opposite direction along E. This implies that there are two antipodal reorientations of the primary plane where the intersections must pass. These push directions correspond to push directions which map (ϕ, ψ), and (ϕ', ψ') onto the same point. □ □

We call the basic operation which collides two orientations onto the same equilibrium for the primary plane CollidePrimaryAction. Combining Lemma 2 and 3 leads to a construction of a push plan for a polyhedral part. The following algorithm orients a polyhedral without symmetry in the planar projections of P for equilibrium contact directions of the primary plane.

OrientPolyhedron(P):
▷ After initialization $|\Sigma| = O(n)$
while $|\Sigma| > 1$
  do pick (ϕ, ψ, θ), $(\phi', \psi', \theta') \in \Sigma$
    plan $\leftarrow$ CollidePrimaryAction($(\phi, \psi, \theta), (\phi', \psi', \theta')$)
    ▷ Lemma 3;
    ▷ plan$(\phi, \psi, \theta) = (\phi'', \psi'', \theta'')$, and plan$(\phi', \psi', \theta') = (\phi'', \psi'', \theta''')$
    for all $(\tilde{\phi}, \tilde{\psi}, \tilde{\theta}) \in \Sigma$
      $(\tilde{\phi}, \tilde{\psi}, \tilde{\theta}) \leftarrow$ plan$(\tilde{\phi}, \tilde{\psi}, \tilde{\theta})$.
    plan $\leftarrow$ CollideRollsSequence(ϕ'', ψ'')
    ▷ Lemma 2
    for all $(\tilde{\phi}, \tilde{\psi}, \tilde{\theta}) \in \Sigma$
      $(\tilde{\phi}, \tilde{\psi}, \tilde{\theta}) \leftarrow$ plan$(\tilde{\phi}, \tilde{\psi}, \tilde{\theta})$.

The number of pushes used by this algorithm sums up to $O(n^2)$. Correctness follows directly from Lemma's 2 and 3.

Theorem 4 *Any asymmetric polyhedral part can be oriented by $O(n^2)$ push operations by two orthogonal planes.*

4 Computing a Push Plan

In this section, we present an algorithm for computing a push plan for a three-dimensional part. We know from Section 3 that such a plan always exists for asymmetric parts. The push plans of Section 3 consist of two stages. During the initialization stage of the algorithm we reduce the number of possible orientations to $O(n)$ different equilibrium contact directions of the primary plane with a unique roll each. The initialization consists of $O(n^2)$ applications of the secondary plane. In the second stage, we run algorithm OrientPolyhedron which repeatedly decreases the number of possible orientations of the part by one, by means of a single application of the primary plane followed by $O(n)$ applications of the secondary plane, until one possible orientation remains. Summing up, a push plan of Section 3 corresponds to $O(n)$ applications of the primary plane, and $O(n^2)$ applications of the secondary plane.

We maintain the $O(n)$ different orientations which remain after the initialization stage in an array. During the execution of the second stage, we update the entries of the array. Hence, for each application of either of the two planes of the jaw, we compute for $O(n)$ orientations of the array the orientation after application of the jaw.

In order to compute the orientation of P after application of the primary plane, we need to be able to compute the path of steepest descent in the radius terrain. In order to determine the orientation of P after application of the secondary plane, we need to be able to compute the planar projection of P onto the primary plane for stable orientations of P, and we need to compute planar push plans.

We start by discussing the computation of the path of steepest in the radius terrain from the initial contact direction of the primary plane. The path is a concatenation of great arcs on the sphere of contact directions of the primary plane. Lemma 5 bounds the complexity of the radius terrain.

Lemma 5 *Let P be a convex polyhedral part with n vertices. The complexity of the radius terrain of P is $O(n)$.*

Proof: There exist bijections between the faces of P and the vertices of the radius terrain, the vertices of P and the patches of the radius terrain, and the edges of P and the edges of the radius terrain. Hence, the combinatorial complexity of the radius terrain equals the combinatorial complexity of P, which is $O(n)$. □ □

In a piecewise-linear terrain with combinatorial complexity n, the complexity of a path of steepest descent can consist of $\Omega(n^2)$ pieces [13]. We shall show, however, that a path of steepest descent in the radius terrain had complexity $O(n)$.

Lemma 6 *Let P be convex polyhedral part. A path of steepest descent in the radius terrain of P has combinatorial complexity $O(n)$.*

Proof: A steepest-descent path in the radius terrain consist of simple subpaths connecting vertices and points on arcs. Thus, the complexity of the path depends on the number of visits of vertices and crossings of arcs. We prove the theorem by showing that the number of visits of a single vertex, and the number of crossings of a single arc is bounded by a constant.

A vertex of the terrain—which corresponds to a face contact—can be visited only once. If the path crosses a vertex, the radius must be strictly decreasing. Hence the path will never reach the height of the vertex again.

We shall show that the path crosses an arc—which corresponds to an edge contact—of the terrain (from one patch to a neighboring patch) at most once. Let us assume that the part is crossing the arc in the terrain which corresponds to a contact of the primary plane to edge (v_1, v_2) of the part. Let us assume that the path in the terrain first travels through the patch of v_1, and then through the patch of v_2. In this case, the part first rotates about v_1, until the edge (v_1, v_2) reaches the primary plane. Instead of rotating about (v_1, v_2), the part subsequently rotates about v_2—the primary plane immediately breaks contact with v_1. Since we assume that the center-of-mass follows the path of steepest descend in the radius terrain, the primary plane can only break contact with v_1 if the distance of v_1 to c is greater than the distance of v_2 to c. See Figure 8.

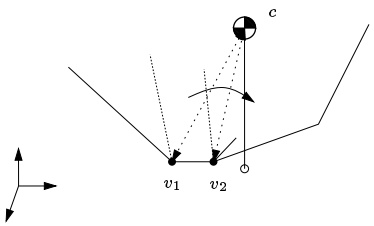


Figure 8: The path of steepest descent, crossing an edge of the radius terrain. The distance from v_1 to c is greater than the distance from v_2 to c.

Hence for each arc crossing, the part pivots on a vertex with smaller distance to c, and consequently crosses each arc at most one time.

Since the number of arcs and vertices of the radius terrain is bounded by $O(n)$, the proof follows. □ □

In order to compute the path of steepest descent, we need not compute the radius terrain. We can suffice with a decomposition of the sphere of contact directions—of which the cells correspond to primary plane-vertex contacts—together with the position of the corresponding vertices on the sphere of directions.

We assume that P is given as a doubly-connected edge list. A doubly connected edge list consists of three arrays which contain the vertices, the edges, and the faces of the part. We refer the reader to [24, 14] for a detailed description, and [4] for a discussion on the implementation of the doubly-connected edge list to represent polyhedra. For our purposes, it suffices to know that the doubly-connected edge list allows us to answer all relevant adjacency queries in constant time.

We compute the decomposition of the sphere of contact directions from the doubly-connected edge list of P. We recall that the cells of the arrangement on the sphere of contact directions correspond to plane-vertex contacts. For contact directions at the boundary of a cells, the primary plane is in contact with at least two vertices, and thus with an edge or face of P.

We use the aforementioned correspondence between the part and the arrangement to efficiently compute the latter from the former. For each edge of the part, we add an edge to the arrangement. The vertices of the edge correspond to the contact directions of the primary plane at the faces of the part neighboring the edge. These contact directions are computed in constant time from the edge and a third vertex on the boundary of the face. The connectivity captured by the representation of the part, easily carries over to the connectivity of the arrangement. Hence, the computation of the doubly-connected edge list representing the arrangement on the sphere of directions can be carried out in $O(n)$ time. With each cell of the arrangement, we store the corresponding vertex of the part. Figure 9(a) shows the decomposition of the sphere of contact directions for a cubic part.

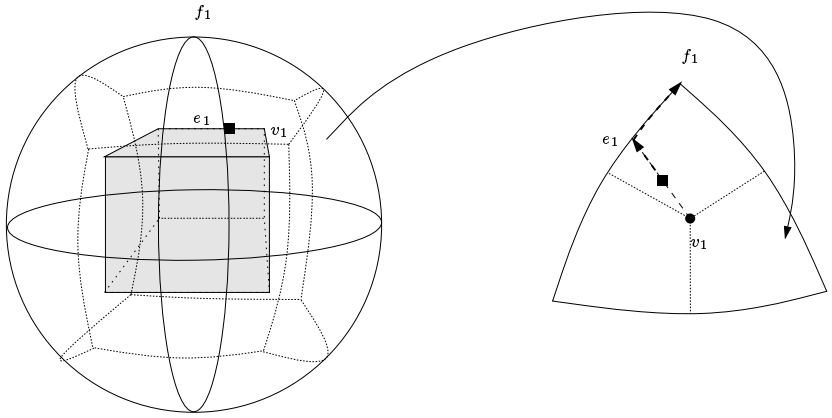


Figure 9: (a) The decomposition of the sphere of directions (solid), together with the projection of the part (dotted). (b) The face for which the primary plane is in contact with v_1. The arrows show the contact directions of the primary plane starting at squared the contact point until the part settles on face f_1.

In the example each face, each edge, and each vertex of the cube has an equilibrium contact direction of the primary plane. As a consequence, any contact direction which corresponds to a face contact is an equilibrium contact direction and the pivoting stops after a constant number of steps. In Figure 9(b), we show the great arcs on the sphere of directions which correspond to the simple rotations of the alignment of the part to the primary plane. Firstly, the part rotates about vertex v_1, until edge e_1 reaches the primary plane. The part continues to rotate about edge e_1, until it finally reaches face f_1.

In order to determine the orientation for a given initial contact direction, we need to determine the contact vertex. In other words, we need to determine which cell of the arrangement corresponds to the contact direction. It is not hard to see that this can be accomplished in linear time, by walking through the arrangement.

Lemma 7 *Let P be a polyhedral part with n vertices in its reference orientation. Let $(\tilde{\phi}, \tilde{\psi})$ be a push direction of the primary plane. We can determine the orientation (ϕ, ψ, θ) of the part after application of the primary plane in $O(n)$ time.*

Computing an orthogonal projection of P onto the primary plane can be carried out in linear time per equilibrium by means of an algorithm of Ponce *et al.* [23], which first finds the leftmost vertex of the projection through linear programming, and then traces the boundary of the projection.

The planar push function of a given projection can be computed in $O(n)$ time by checking its vertices. Querying the planar push function can be carried out in $O(\log n)$ time by performing a binary search on the initial orientation.

Lemma 8 *Let P be a polyhedral part with n vertices in equilibrium orientation (ϕ, ψ, θ). Let $\tilde{\theta}$ be a push direction of the secondary plane. We can determine the orientation (ϕ, ψ, θ') of the part after application of the secondary plane in $O(\log n)$ time.*

For almost all parts, the computation of a planar push plan of linear length can be done in $O(n)$ time using an algorithm due to Chen and Ierardi [12]. Chen and Ierardi show that there are pathological parts for which they only give an $O(n^2)$ algorithm for computing a push plan. So, the best upper bound on the running time to compute CollideRollsSequence for $O(n)$ projections of P is $O(n^3)$. It remains open whether a polyhedral part can have $\Omega(n^2)$ pathological projections, or to improve the bound on the running time in an other way. Computing the push direction of the primary plane which maps two different faces onto the same equilibrium with respect to the primary plane (CollidePrimaryAction) can be done in constant time.

Summarizing, the total cost of computing the reorientations of the jaw takes $O(n^2)$ time. The cost of the necessary maintenance of $O(n)$ possible orientations of P is the sum of $O(n^2)$ updates for applications of the secondary plane which take $O(n \log n)$ time each, and $O(n)$ updates for applications of the primary plane, which take $O(n^2)$ maintenance time each. Theorem 9 gives the main result of this section.

Theorem 9 *A push plan of length $O(n^2)$ for an symmetric polyhedral part with n vertices can be computed in $O(n^3 \log n)$ time.*

5 Plates with fences

In this section we will use the results from the preceding sections to determine a design for the feeder consisting of tilted plates with curved tips, each carrying a sequence of fences. The motion of the part effectively turns the role of the plates into the role of the primary pushing plane, and the role of the fences into the role of the secondary pushing plane. We assume that the part quasi-statically aligns to the next plate, similar to the alignment with the primary plane of the generic jaw. Also, we assume that the contact direction of the plate does not change as the fences brush the part, i.e. the part does not tumble over.

The fact that the direction of the push, i.e., the normal at the fence, must have a non-zero component in the direction opposite to the motion of the part, which is pulled down by gravity, imposes a restriction on successive push directions of the secondary plane. Fence design can be regarded as finding a constrained sequence of push directions. The additional constraints make fence design in the plane considerably more difficult than orientation by a pushing jaw.

As the part moves towards the end of a plate, the curved end of the plate causes the feature on which the part rests to align with the vertical axis, while retaining the roll of the part. When the part leaves the plate, the next plate can

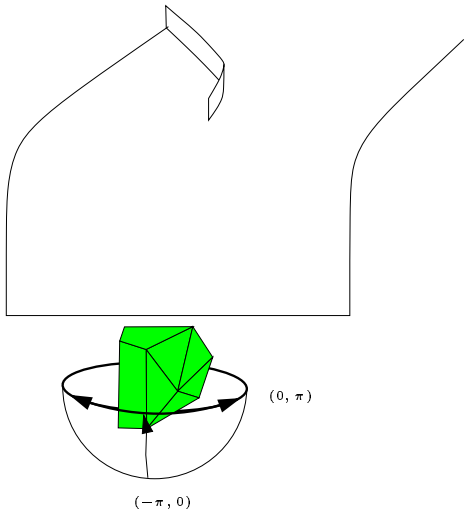


Figure 10: The next plate can only touch the lower half of the part after it left a plate.

only push the part from below. This draws restrictions on the possible reorientations of the primary plane, in the model with the generic three-dimensional jaw.

From Figure 10 follows that the reorientation of the primary plane is within $(-\pi, 0) \times (0, \pi)$ when the last fence of the last plate was a left fence. Similarly, for a last right fence, the reorientation of the primary plane is within $(0, \pi) \times (0, \pi)$.

Berretty *et al.*[5] showed that it is possible to orient a planar polygonal part (hence a polyhedral part resting on a fixed face) using $O(n^2)$ fences. The optimal fence design can be computed in $O(n^3 \log n)$ time.

The gravitational force restricts our possible orientations of the primary plane in the general framework. Fortunately, Lemma 3 gives us two antipodal possible reorientations of the primary plane. It is not hard to see that one of these reorientations is in the reachable hemisphere of reorientations between two plates.

This implies we can still find a fence and plate design, which consists of $O(n^3)$ push operations.

Theorem 10 *An asymmetric polyhedral part can be oriented using $O(n^3)$ fences and plates. We can compute the design in $O(n^4 \log n)$ time.*

6 Conclusion

We have shown that sensorless orientation of an asymmetric polyhedral part by a sequence of push actions by a jaw consisting of two orthogonal planes is

possible. We have shown that the length of the sequence of actions is $O(n^2)$ for parts with n vertices, and that such a sequence can be determined in $O(n^3 \log n)$ time.

We have proposed a three-dimensional generalization of conveyor belts with fences [5]. This generalization consists of a sequence of tilted plates with curved tips, each carrying a sequence of fences. A part slides along the fences of a plate to reach the curved tip where it slides off onto the next plate. Under the assumptions that the motion of the part between two plates is quasi-static and that a part does not tumble from one face onto another during its slide along one plate, we can compute a set-up of $O(n^3)$ plates and fences in $O(n^4 \log n)$ time that will orient a given part with n vertices. (As in the two-dimensional instance of fence design, the computation of such a set-up boils down to the computation of a constrained sequence of push actions.)

Our aim in this paper has been to gain insight into the complexity of sensorless orientation of three-dimensional parts rather than to create a perfect model of the behaviour of pushed (or sliding) and falling parts. Nevertheless, we can relax some of the assumptions in this paper. First of all, in a practical setting, a part which does not rest on a stable face, but on a vertex or edge instead, will most likely change its contact direction with the primary plane if it is pushed from the side. Hence, we want to restrict ourselves orientations of P which have stable equilibrium contact directions of the primary plane. After the first application of the jaw, it might be the case that P is in one of its unstable rather than stable equilibria. A sufficiently small reorientation of the jaw in an appropriate direction, followed by a second application of the jaw, will move the part towards a stable orientation though, allowing us to start from stable orientations only.

The computation of the reorientation of the primary plane results in two candidate reorientations. Although extremely unlikely, these reorientations could both correspond to unstable equilibrium contact directions. As mentioned, in a practical situation one wants to avoid such push directions. It is an open question whether there exist parts which can not be oriented without such unstable contact directions.

Our approach works for parts which have asymmetric projections onto the primary plane for all equilibrium contact directions of primary plane. It is an open problem to exactly classify parts that cannot be fed by the jaw.

It is interesting to see how the ideas from this paper can be extended to other feeders such as the parallel jaw gripper, which first orients a three-dimensional part in the plane, and subsequently drops it onto another orientation. Rao *et al.*[25] show how to compute contact directions for a parallel jaw gripper to move a three-dimensional part from a known orientation to another one. We want to see if this method generalizes to sensorless reorientation.

The algorithm of this paper generates push plans of quadratic length. It remains to be shown whether this bound is asymptotically tight. Also, it is interesting to find an algorithm which computes the shortest push plan that orients a given part. Such an algorithm would need to decompose the space of possible reorientations of the jaw for P in its reference orientation into regions

which map onto different final orientations of P. This requires a proper algebraic formulation of the push function, and a costly computation of corresponding the arrangement in the space of push directions. In contract to the planar push function, the three-dimensional push function is not a monotonous transfer function. Eppstein [15] showed that, in general, finding a shortest plans is NP-complete. It is an open question whether we can find an algorithm for computing a shortest plan for the generic jaw that runs in polynomial time.

References

[1] S. Akella, W. Huang, K.M. Lynch, and M.T. Mason. Sensorless parts feeding with a one joint robot. *Algorithms for Robotic Motion and Manipulation* (J.-P. Laumond and M. Overmars (Eds.)), A.K. Peters, pages 229–238, 1996.

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

[3] Srinivas Akella and Matthew T. Mason. Posing polygonal objects in the plane by pushing. In *IEEE International Conference on Robotics and Automation*, 1992.

[4] G. Barequet. DCEL: A polyhedral database and programming environment. *Internat. J. Comput. Geom. Appl.*, 8:619–636, 1998.

[5] Robert-Paul Berretty, Ken Goldberg, Mark Overmars, and A. Frank van der Stappen. Computing fence designs for orienting parts. *Computational Geometry, Theory and Applications*, 10(4):249–262, 1998.

[6] K.-F. Böhringer, V. Bhatt, and K.Y. Goldberg. Sensorless manipulation using transverse vibrations of a plate. In *Proc. IEEE Int. Conf. on Robotics and Automation, Nagoya, Japan*, pages 1989–1996, 1995.

[7] K.-F. Böhringer, B.R. Donald, and N.C. MacDonald. Upper and lower bounds for programmable vector fields with applications to mems and vibratory plate part feeders. *Algorithms for Robotic Motion and Manipulation,* J.-P. Laumond and M. Overmars (Eds.), A.K. Peters, pages 255–276, 1996.

[8] G. Boothroyd and P. Dewhurst. *Design for Assembly – A Designers Handbook.* Department of Mechanical Engineering, University of Massachusetts, Amherst, Mass., 1983.

[9] G. Boothroyd, C. Poli, and L. Murch. *Automatic Assembly.* Marcel Dekker, Inc., New York, 1982.

[10] M. Brokowski, M.A. Peshkin, and K. Goldberg. Optimal curved fences for part alignment on a belt. *ASME Transactions of Mechanical Design*, 117, March 1995.

[11] J. F. Canny and K. Y. Goldberg. Risc for industrial robotics: Recent results and open problems. In *Proc. IEEE Internat. Conf. Robot. Autom.*, May 1994.

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

[13] M. de Berg, P. Bose, K. Dobrint, M. van Kreveld, M. Overmars, M. de Groot, T. Roos, J. Snoeyink, and S. Yu. The complexity of rivers in triangulated terrains. In *Proc. 8th Canad. Conf. Comput. Geom.*, pages 325–330, 1996.

[14] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Berlin, 1997.

[15] D. Eppstein. Reset sequences for monotonic automata. *SIAM Journal of Computing*, 19(5):500–510, 1990.

[16] M.A. Erdmann and M.T. Mason. An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4:367–379, 1988.

[17] Ken Goldberg. Orienting polygonal parts without sensors. *Algoritmica*, 10(2):201–225, 1993.

[18] K. M. Lynch and Matthew T. Mason. Stable pushing: Mechanics, controllability, and planning. *International Journal of Robotics Research*, 1996. to appear, (An earlier version appeared in the First Workshop on the Algorithmic Foundations of Robotics, A. K. Peters, Boston, 1995.).

[19] M. Mason. *Manipulator grasping and pushing operations.* PhD thesis, MIT, 1982. published in *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, 1985.

[20] B.K. Natarajan. An algorithmic approach to the automated design of parts orienters. In *IEEE Annual Symposium on Foundations of Computer Science*, pages 132–142, 1986.

[21] M.A. Peshkin and A.C. Sanderson. The motion of a pushed sliding workpiece. *IEEE Journal of Robotics and Automation*, 4(6):569–598, 1988.

[22] M.A. Peshkin and A.C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 1988.

[23] Jean Ponce, Steve Sullivan, Attawith Sudsang, Jean-Daniel Boissonnat, and Jean-Pierre Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *Internat. J. Robot. Res.*, 16(1):13–35, 1997.

[24] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 3rd edition, October 1990.

[25] Anil Rao, David Kriegman, and Ken Goldberg. Complete algorithms for reorienting polyhedral parts using a pivoting gripper. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C22–C23, 1995.

[26] J. Wiegley, K. Goldberg, M. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. *Assembly Automation*, 17(2):129–136, August 1997.