Orienting Polyhedral Parts by Pushing

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

1 Introduction

Many automated manufacturing processes require parts to be oriented prior to assembly. A part feeder takes in a stream of identical parts in arbitrary orientations and outputs them in a uniform orientation. Part feeders often use data obtained from some kind of sensing device to accomplish their task.

We consider the problem of *sensorless orientation* of parts, in which the initial pose of the part is assumed to be unknown. In sensorless manipulation, parts are positioned or oriented using passive mechanical compliance. The input is a description of the part shape and the output is a sequence of open-loop actions that moves a part from an unknown initial pose into a unique final pose. Among the sensorless part feeders considered in literature are the parallel-jaw gripper [7, 9], a single pushing jaw [2, 10, 11, 13], a conveyor belt with a sequence of (stationary) fences placed along its sides [6, 14, 15], a conveyor belt with a single rotational fence (1JOC) [1], a tilting tray [8, 12], and vibratory plates and programmable vector fields [4, 5].

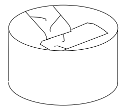

Figure 1: The three-dimensional part feeder. Plates with fences mounted to a cylinder.

The pushing jaw [2, 10, 11, 13] orients a part by an alternating sequence of pushes and jaw reorientations. The problem 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 [9] showed that any polygonal part can be oriented by a sequence of pushes. Chen and Ierardi [7] proved that any polygonal part with m vertices can be oriented by $O(m)$ pushes. They showed that this bound is tight by constructing (pathological) m-gons that require $\Omega(m)$ pushes to be oriented. Goldberg gave an algorithm for computing the shortest push plan for a polygon. His algorithm runs in $O(m^2)$ 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 of which the face on which the part rests on the surface 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. A picture of the device is given in Figure 1. We show how this device relates to genericly pushing a three-dimensional part. In this paper, give a general method which applies to many part feeders that act on three-dimensional parts. We give an algorithm with $O(n^2)$ running time which computes a sequence of $O(n^2)$ push actions, which is guaranteed to orient a polyhedral part up to symmetry. In the full paper, we present a thorough analysis of the rotational behavior of a three-dimensional part that is pushed, and give an algorithm which computes a sequence of push actions of minimal length. The drawback of the latter algorithm is its exponential running time.

*Department of Computer Science, Utrecht University, PO Box 80089, 3508 TB Utrecht, The Netherlands.

†Berretty's research is supported by the Dutch Organization for Scientific Research (N.W.O.)

2 Pushing Parts

A part P in a three-dimensional space has three rotational degrees of freedom. We denote the orientation of P by $q = (\phi, \psi, \theta)$, with (ϕ, ψ) a point on the sphere of directions (S.O.D), and θ the twist, which denotes the rotation around the axis parallel to the vector (ϕ, ψ).

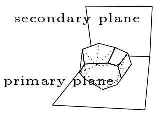

Figure 2: The generic jaw in the three-dimensional setting.

The device we use to orient this part discretizes the possible orientations of the part. A part resting on plate retains one undiscretized rotational degree of freedom. To discretize the twist of the part, we also push the part from the side by the fences across the plates.

The push actions of the plates and the fences can be generalised by letting the part rest on a plane, and letting it be aligned to a second plane orthogonal to the former plane. A picture of the generalised jaw is given in Figure 2. We call the planes of the generalised jaw the primary and secondary pushing plane.

As the part rests on the primary plane, the secondary plane pushes P at its orthogonal projection onto the primary plane. We make sure that the face on which P rests retains contact with the primary plane as the secondary plane touches P. Hence, the push action by the secondary plane changes the orientation of the projection of P, but not its shape. The application of the secondary plane to the part can, therefore, be regarded as a push operation on a two-dimensional part (namely, the projection of P). Since the plates can only touch P at its convex hull, we assume without loss of generality that P is convex.

We assume that a fixed frame is attached to P. Directions are relative to this frame. We study the *radius function* of the part, in order to identify the equilibrium orientations of the part. The radius function $r : ([0, 2\pi) \times [0, \pi)) \to \mathbb{R}^+$ maps a direction (ϕ, ψ) onto the distance from c to a supporting plane of P with contact direction (ϕ, ψ).

If the radius function of the part at the contact direction of the primary plane is not a local extremum, or saddle point, the center of mass of the part can locally reduce its distance to the primary plane, and the part will rotate towards an orientation with smaller radius. If, on the other hand, the contact direction of the primary plane corresponds to a local extremum, or saddle point of the radius function, the part will remain at this orientation.

If the parts' orientation corresponds to local minimum in the radius function of the part, a small change of the push direction will map the part back onto the same orientation. If the part is at an orientation which corresponds to a local maximum or saddle point of the radius function, a small change of the push direction will cause the part to rotate to another equilibrium orientation.

The *face push function* $p_f : [0, 2\pi) \times [0, \pi) \rightarrow [0, 2\pi) \times [0, \pi)$ links every push direction of the primary plane (ϕ, ψ) to the orientation $p_f(\phi, \psi)$ in which the part P settles after being pushed by the plane with initial contact direction (ϕ, ψ) (relative to the frame attached to P). The rotation of the part due to pushing causes the contact direction of the jaw to change. The final orientation $p_f(\phi, \psi)$ of the part is the contact direction of the jaw after the part has settled. The equilibrium push directions are the fixed points of p_f.

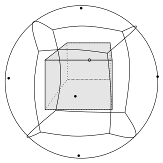

Figure 3: The projection of a cubic part onto the sphere of directions.

We derive the part's face push function from the part's radius function. To accomplish this, we place the part's center of mass at the origin of the sphere of directions, and project the edges containing local maxima or saddle points of the part's radius function onto the sphere. The projections of the edges are great arcs on the sphere of directions establishing a subdivision of the sphere of directions which captures the push function of the part. We specify the push direction of the primary plane in polar coordinates, by the tuple (ϕ, ψ).

We assume that the pushing planes are frictionless. This implies that there is no rotational component along the normal of the pushing plane. Therefore, the center of mass of the part moves along a line perpendicular to the primary pushing plane, towards this plane. As a consequence, the final orientation of the part can be computed by translating along a great arc on the sphere of directions, towards the local minimum reachable from the orientation in which we start. The twist of the part does *not* change during this alignment, and can be computed using elementary spherical trigonometry.

The push operation of the secondary plane corresponds to a push operation on a two-dimensional part. The two-dimensional push function is derived from the two-dimensional radius function.

In order to able to reach 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 of 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 analysing our device later.

We shall first identify the possible stable orientations of a polyhedral part, and then discuss the rotational behavior of a part being pushed by our device.

There are $O(n)$ stable faces, each stable face can align to $O(n)$ sides of the orthogonal projection of the part. We can construct a part with $\Omega(n^2)$ of stable orientations. The number of stable orientations of the part is, therefore, $\Theta(n^2)$.

Theorem 2.1 *A polygonal part with combinatorial complexity n has $\Theta(n^2)$ stable orientations.*

As the part aligns with the device, we identify two stages: 1. Alignment with the primary plane; 2. Alignment with the secondary plane.

We assume that only after the part has settled with the primary pushing plane, the part slides towards the secondary pushing plane, and further settles with the jaw until a stable orientation is reached.

The second part of the alignment is analogous to planar pushing, taking the projection of the part onto the primary plane as input to the radius function in the plane.

Hence, the complete three-dimensional push function $p : ([0, 2\pi) \times [0, \pi) \times [0, 2\pi)) \rightarrow ([0, 2\pi) \times [0, \pi) \times [0.2\pi))$ maps the push direction onto possible orientations of P.

Our goal is to find a sequence of push operations, which are represented by $\alpha_1, \ldots, \alpha_m$, the successive reorientations of the jaw, that maps any initial orientation onto a unique final orientation. We call such a sequence $\alpha_1, \ldots, \alpha_m$ a push plan.

3 Orienting a polyhedral part

In this section we will discuss how to orient a polyhedral part. After the first settlement with our jaw, the part can rest on any of its $O(n)$ faces, and can be aligned with the secondary pushing plane in $O(n)$ orientations per face, and can therefore have $O(n^2)$ possible orientations.

Let us for a moment assume that we know the face on which the part rests, i.e. (ϕ, ψ) is known.

The orthogonality of the two planes allows us to orient the part up to symmetry. If we rotate the jaw around the axis in the direction of (ϕ, ψ), and we tilt the jaw towards the secondary pushing plane, we have, informally speaking, turned the generic jaw into a two-dimensional jaw which acts on the orthogonal projection of our three-dimensional part onto the primary pushing plane.

The result of Chen and Ierardi [7] tells us that a two-dimensional part with m vertices can be oriented up to symmetry in $O(m)$ push operations. Hence, we can orient each single projection in $O(n)$ push operations. We can also eliminate the uncertainty in the twist for any face in contact with the primary pushing plane using $O(n^2)$ push operations, by concatenating the push sequences for the orthogonal projections of the part at each stable face.

Lemma 3.1 *Let P be a polyhedral part with n vertices. Let f be a stable face of P. Using $O(n)$ push operations we can reduce the possible orientations of P of which f is in contact with the primary pushing plane to one.*

We call the operation which orients a single face f OrientProjection(f). We shall explain how we can further reduce the number of possible orientations. To this end, we use the following lemma.

Lemma 3.2 *Let P denote a polygonal part. Let $q = (\phi, \psi, \theta)$, and $q' = (\phi', \psi', \theta')$ denote two possible orientations of P. There is a push direction p which maps q and q' onto the same face of P.*

Proof: Let (ϕ, ψ), and (ϕ', ψ') be two points on the sphere of directions. We want to find a push direction which maps these two points onto another point (ϕ'', ψ''). See Figure 4. 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'). 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. 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 reorientations of the pushing plane where the intersections must pass. These push directions correspond to push directions which map (ϕ, ψ), and (ϕ', ψ') onto the same point. $\square$

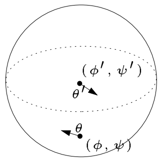

Figure 4: Two orientations on the sphere of directions. Their equator is dashed.

Combining Lemma's 3.1 and 3.2 leads to a construction of a push plan for a polyhedral part. The following algorithm orients a polyhedral part up to symmetry.

OrientPolyhedron(P)
for each face $f \in P$
    do OrientProjection(f)
$\mathcal{O} \leftarrow$ current orientations of P.
while there are with $o, o' \in O, o \neq o'$
        do map o, o' onto o''
            OrientProjection(face(o''))
            update $\mathcal{O}$.

The number of pushes used by this algorithm sums up to $O(n^2)$. Correctness follows directly from Lemma's 3.1 and 3.2.

Theorem 3.3 *Let P be a polyhedral part. P can be oriented in $O(n^2)$ push operations.*

4 Fences over plates

In this section we will apply the general framework to the part feeder we mentioned in the introduction. The feeder consists of a cylinder with plates tilted from the boundary of the cylinder downwards towards its interior. At the end of a plate, the parts fall off, onto another plate. We assume that there is no friction between the plates, the fences, and the part. Again, we assume that motion of the part is quasi-static, i.e. we ignore inertia.

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 the role of the secondary pushing plane.

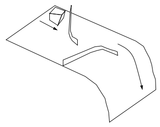

Figure 5: A part on a plate with fences.

A sequence of fences across a plate, corresponds to a sequence of push actions in the plane and are thoroughly studied [3, 6, 14, 15]. 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. 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.

In the planar case, sliding along a fence f_i, while remaining on the same plate, causes one of the edges, say e, of the projection of P to align with the fence. The curved tip of the fence [6] guarantees that e is aligned with the plate sides as P leaves the fence. So, the reorientation of the push is expressed relative to this direction. Figure 6 shows an example which explains that the reorientation of the push direction in the plane is in the range $(0, \pi/2)$ if we choose two successive left fences. A similar analysis can be done for other combinations of fences. It turns out that the type of fence f_i imposes a bound on the reorientation of push direction in the plane. Careful analysis leads to the following definition of a valid fence design [15]. A planar *fence design* is a planar push plan (i.e. angles are expressed in the plane) $\alpha_1, \ldots, \alpha_m$ satisfying for all $1 \leq i < m$: $\alpha_i \in (0, \frac{\pi}{2}) \cup (-\pi, -\frac{\pi}{2}) \Rightarrow \alpha_{i+1} \in (0, \frac{\pi}{2}) \cup (\frac{\pi}{2}, \pi)$ and $\alpha_i \in (-\frac{\pi}{2}, 0) \cup (\frac{\pi}{2}, \pi) \Rightarrow \alpha_{i+1} \in (-\frac{\pi}{2}, 0) \cup (-\pi, -\frac{\pi}{2})$.

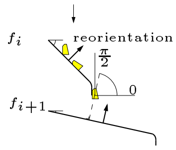

Figure 6: The reorientation of the planar push direction for two successive left fences.

As the part moves towards the end of a plate, the curved end of the plate causes the face on which the part rests to align with the vertical axis, while retaining the twist of the part. When the part leaves the plate, the next plate can 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 7 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 $(0, \pi) \times (0, \pi)$. On the next plate, the zero twisted secondary plane corresponds to a horizontal fence. Hence, the possible twist of the secondary plane is within $(-\frac{\pi}{2}, 0) \cup (0, \frac{\pi}{2})$, constructed by either a right, or a left fence to start with on the plate. The following definition summarizes the possible reorientations of the push direction.

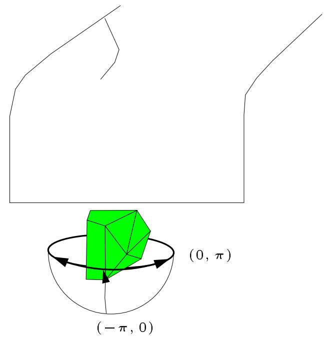

Figure 7: The next plate can only touch the lower half of the part after it left a plate.

Definition 4.1 *A* fence and plate design *is a three-dimensional push plan $\alpha_1, \ldots, \alpha_m$ satisfying for all $1 \leq i < m$:*

- $\alpha_i \in (0, 0, (0, \frac{\pi}{2}) \cup (-\pi, -\frac{\pi}{2}) \Rightarrow \alpha_{i+1} \in (0, 0, (0, \frac{\pi}{2})) \cup (\frac{\pi}{2}, \pi)) \cup ((-\pi, 0), (0, \pi), (-\frac{\pi}{2}, 0) \cup (0, \frac{\pi}{2}))$

- $\alpha_i \in (0, 0, (-\frac{\pi}{2}, 0) \cup (\frac{\pi}{2}, \pi)) \Rightarrow \alpha_{i+1} \in (0, 0, (-\frac{\pi}{2}, 0) \cup (-\pi, -\frac{\pi}{2})) \cup ((0, \pi), (0, \pi), (-\frac{\pi}{2}, 0) \cup (0, \frac{\pi}{2}))$.

Now, we show how to orient a polyhedral part with fences over plates. Recall that in Section 3 we discussed an algorithm to orient any polyhedral part with a generic jaw. In an earlier paper, we 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.

Definition 4.1 restricts our possible orientations of the primary plane in the general framework. Fortunately, Lemma 3.2 gives us two possible reorientations, say (ϕ_1, ψ_1) and (ϕ_2, ψ_2) of the primary plane Since the reorientation maps the original orientations onto the equator of the two orientations, $0 < \psi_1,\ \psi_2 < \pi$. Since $\phi_1 - \phi_2 = \pi$, one of these reorientations is in the reachable hemisphere of reorientations between two plates.

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

Theorem 4.2 *Let P be a polyhedral part. P can be oriented using $O(n^3)$ fences and plates.*

5 Conclusion

In this paper, we introduced a sensorless part feeder, consisting of cylinder with plates and fences, which is guaranteed to orient a given polyhedral part up to rotational symmetry, under the assumptions of quasi-static motion and zero friction. Also, we gave an algorithm to compute the necessary angles of the plates and the fences for a given polyhedral part. Also, we considered a general three-dimensional jaw, and gave insight into sensorless orientation in three dimensions in a more general setting. We gave an algorithm to orient a polyhedral part with our jaw. The algorithm computes a push plan of $O(n^2)$ push operations.

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 and M.T. Mason. Posing polygonal objects in the plane by pushing. In *IEEE International Conference on Robotics and Automation, Nice*, pages 2255–2268, 1992.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

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

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

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

[12] 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.

[13] 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.

[14] 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, 1988.

[15] 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.