

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/3298631>

Complete Algorithms for Feeding Polyhedral Parts using Pivot Grasps

Article in IEEE Transactions on Robotics and Automation · May 1996

DOI: 10.1109/70.488952 · Source: IEEE Xplore

CITATIONS

36

READS

74

3 authors, including:



[David Kriegman](#)

University of California, San Diego

240 PUBLICATIONS 37,123 CITATIONS

[SEE PROFILE](#)



[Kenneth Yigael Goldberg](#)

University of California, Berkeley

442 PUBLICATIONS 11,886 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



eParticipation 2.0 [View project](#)



Structural Edge Detection for Cardiovascular Modeling [View project](#)

Complete Algorithms for Feeding Polyhedral Parts using Pivot Grasps

Anil Rao*
Utrecht University

David Kriegman†
Yale University

Ken Goldberg‡
USC

Abstract

To rapidly feed industrial parts on an assembly line, Carlisle *et al.* [7] proposed a flexible part feeding system that drops parts on a flat conveyor belt, determines position and orientation of each part with a vision system, and then moves them into a desired orientation. A robot arm with 4 degrees of freedom (DoF) is capable of moving parts through 6 DoF when equipped with a passive pivoting axis between the parallel jaws of its gripper. The idea is to grasp a part with 2 hard finger contacts such that it pivots, under gravity, into a desired orientation when lifted and replaced on the table. We refer to these actions as *pivot grasps*.

This paper considers the planning problem. Given a polyhedral part shape, coefficient of friction and a pair of stable configurations as input, find pairs of grasp points that will cause the part to pivot from one stable configuration to the other. For some transitions, pivot grasps may not exist. For a part with n faces and m stable configurations, we give an $O(m^2 n \log n)$ algorithm to generate the $m \times m$ matrix of pivot grasps. When the part is star shaped, this reduces to $O(m^2 n)$. We also study a generalization that considers “capture regions” around stable configurations. Both algorithms are complete in that they are guaranteed to find pivot grasps when they exist.

1 Introduction

Achieving a desired spatial configuration of a part is a fundamental issue in robotics. For example, consider a part resting stably on a flat table. After the part is grasped in a known configuration by a robot arm, inverse kinematics can be used to achieve a desired final configuration of the part. This assumes that the grasp does not slip and that the final configuration of the part is reachable by the robot. Noting that such conditions are not always met, Tournassoud, Lozano-Perez, and Mazer proposed planning a sequence of *regrasping* operations that replace the part on the table in intermediate configurations, thereby allowing the robot to achieve a better grasp. In the presence of obstacles, they showed how to plan regrasping operations for a 6 DoF robot arm by slicing C-space but did not provide a complete algorithm [29].

Automating the grasp analysis is useful for rapid set-up of a parts feeding system using vision and a robot manipulator [7]. An efficient algorithm is particularly useful when incorporated into a solid modelling package: as the designer creates a new part, he or she can immediately test the “feedability” of this part, perhaps modifying the shape accordingly.

In an industrial setting where cost, accuracy, reliability, and speed are paramount, Carlisle *et al.* [7] considered a SCARA-type arm with only 4 active DoF to feed a stream of parts arriving on a conveyor belt (see Fig. 1). Four degree of freedom manipulators, such a SCARA arm or Automatrix Robot World module, are kinematically limited to orienting parts about the vertical axis. However, as we will show in this paper, any orientation can be achieved when the arm is equipped with a passive pivoting axis between the parallel jaws of its gripper.

Achieving an arbitrary part position and orientation using a manipulator with fewer than 6 DoF may appear counter-intuitive at first. For the type of arms considered in this paper, translations and rotations are easily separated, so only

*Department of Computer Science, Utrecht University, Postbus 80.089, Padualaan 14, 3508 TB Utrecht, The Netherlands. 31-(30) 535093, anil@cs.ruu.nl. Rao is supported by the ESPRIT Basic Research Action No. 6546 (Project PRoMotion).

†Center for Systems Science, Department of Electrical Engineering, Yale University, New Haven, CT 06520-8267, USA. 1-(203) 432-4091, FAX: 1-(203)-432-7481, kriegman@yale.edu. Kriegman was supported in part by an NSF Young Investigator Award IRI-9257990

‡Institute for Robotics and Intelligent Systems, Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0273, USA. 1-(213) 740-9080, goldberg@usc.edu. Goldberg is supported by NSF Young Investigator Award IRI-9457523 and IRI-9123747 and by Adept Technology, Inc.

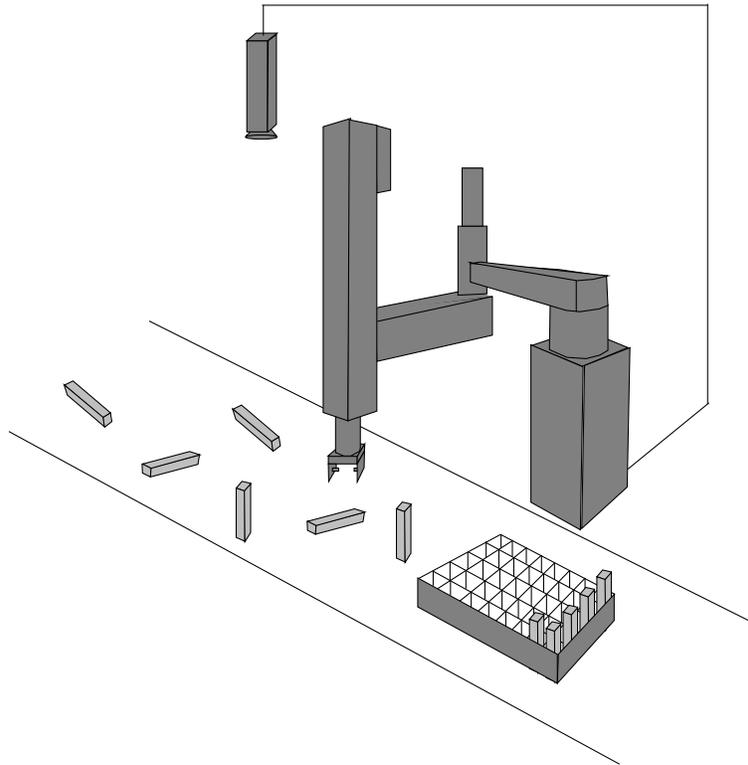


Figure 1: The Flexible Part Feeder described in Carlisle *et al.* A 4 DoF robot arm and vision system for feeding polyhedral parts.

rotations will be addressed. It is well known that any 3D rotation can be decomposed into three sequential rotations (e.g. Euler angles or roll-pitch-yaw). Now consider affixing three initially aligned coordinate systems to the gripper, to the world, and to the part with the Z axis pointing vertically and the Y axis in the direction of the pivot axis. The part is initially ungrasped. The sequence of operations considered here is:

1. Rotate the gripper about the vertical Z axis.
2. Grasp the part and lift it.
3. The grasped part is rotated about the horizontal pivot axis.
4. Rotate the grasped part about the vertical axis.

This four step procedure is equivalent performing three rotations (*i.e.* Z , Y and Z rotations in the gripper frame). Note that the first and last rotations are independent because of the change in the grasp condition. During the first Z rotation, the part moves with respect to the gripper but not with respect to the world. Whereas in the fourth step, the part does not move with respect to the gripper but moves with respect to the world frame. The change due to grasping the part is a non-integrable constraint making this a non-holonomic system as noted by Koditschek [18, 19].

The gripper is equipped with two hard finger contacts for grasps on the part. In Step 1, the gripper is rotated so as to correctly position and orientation the fingers; now squeezing the fingers establishes contact which also establishes the grasp axis since it simply is the line connecting the two contact points. Given the center of gravity of the part, the grasp axis, in turn, determines the magnitude and direction of the unactuated pivot rotation in Step 3. After stabilization, the part is finally rotated about the vertical Z axis to complete the triad of rotations.

We assume that each part is dropped onto the conveyor belt in isolation (we do not address the related problem of singulating parts). When rotations and translations in the plane are ignored, the part generally assumes one of a finite number of stable poses [22]. For a polyhedron \mathcal{P} with n faces, a pose is stable when the center of gravity lies above the face of the convex hull \mathcal{H} which contacts the support plane. In this paper, we consider pivot grasps that move a part from an initial stable configuration \hat{s} to a final stable configuration \hat{f} . The decision question is whether or not a *single* pivot grasp can accomplish this task: for this we give a $O(n \log n)$ time solution.

Using a particular class of grasps, it may not be possible to move the part between an arbitrary pair of stable configurations in a single pivot operation. Considering each stable configuration as a node in a transition graph and the particular action that will move the part between stable configurations as directed arcs, a directed graph can be defined. A path through this transition graph represents a plan which moves the part from some initial to final configuration, each edge in this path corresponding to a single pivot grasp.

If the transition graph is complete (an edge exists between any ordered pair of nodes), then only single pivot grasps need be considered. However, the transition graph may not be complete. For example, it may be impossible to reorient a part such as a pyramid resting on its base if the coefficient of static friction is too small. Similarly, it is not generally possible to rotate a part from one face to a face with an anti-parallel normal using a single pivot grasp. However, it may be possible to move a part between any pair of configurations using a *sequence* of pivot grasps.

We first consider a set of grasps where the part’s configuration prior to set-down is exactly the desired final configuration. We refer to these as “exact” pivot grasps. We show that the transition graph can be constructed in $O(m^2 n \log n)$ time where n is the number of faces of \mathcal{P} , and $m \leq n$ is the number of stable faces on \mathcal{H} . The algorithm is *complete* in the sense that whenever a sequence of exact pivot grasps exists, we are guaranteed to find it [14].

Note that the transition graph may not be strongly connected for this class of grasps; a sequence of exact pivot transitions between certain pairs of stable configurations may be impossible. Furthermore, even within a strongly connected component, the shortest sequence of transitions between a certain pair of nodes may be too long to be acceptable in practice. Therefore, with a view to improve connectivity in our transition network, we next consider a broader class of grasps. We note that when a part is placed on a supporting plane in a pose that is not stable, the force of gravity will cause the part to tumble onto one of the stable configurations. By explicitly computing the set of configurations (a capture region) which converge to a particular pose, the pivoting operation is only required to bring the part to within the capture region. Since all stable configurations are contained within a capture region, this is a richer action set, and therefore the graph will have better connectivity. We call these “capturing” pivot grasps.

We begin below by reviewing related work and then define the problem and state our assumptions in Section 3. Theory common to both types of pivot grasps is discussed in Section 4. Exact pivot grasps are covered in Section 5, and capturing pivot grasps in Section 6. We implemented the exact planner as described in Section 5.1.

A paper based on this report was to a special issue on “Assembly and Task Planning for Manufacturing” of the *IEEE Transactions on Robotics and Automation*. A condensed version is submitted to the 1995 IEEE International Conference on Robotics and Automation.

2 Related Work

Our results build on prior research in robot motion planning and grasp planning; excellent reviews can be found in [15, 26]. We consider grasps with two frictional point contacts, also known as hard finger contacts [28]. Each contact allows forces pointing into the associated friction cone. Clearly, such a grasp cannot achieve form closure: the part is free to rotate about the contact axis. For planar smooth objects, Faverjon and Ponce [11] and later, Blake [3], considered computing frictional two-finger force-closure grasps. To achieve form closure, Markenscoff *et al.* [24] showed that four hard finger contacts are necessary and sufficient for planar objects and that twelve were sufficient (seven are necessary) to grasp 3D piecewise-smooth objects without rotational symmetries.

However for reorienting parts we do not require form closure; we must insure that the part will not *translate* when lifted but will in fact *rotate* about the grasp axis. In other words, we wish to control certain motions while preventing others. Brock [4] also considered controlled motions under grasp: given an external grasp on an object, represented as contact forces and torques, what are the (infinitesimal) motions possible? However, he does not study *synthesizing* grasps for a desired motion. Trinkle and Paul [30] used controlled slip to gain an enveloping grasp of a part initially in contact with a flat support surface by squeezing contacts with a two fingered hand.

For polyhedral parts, Erdmann *et al.* [10] showed how to tilt an infinite plane to orient a given polyhedral part, regardless of its initial orientation. Both Akella and Mason [1] and Lynch [23] addressed the planar problem of planning a strategy to move a part from a *known* pose into a desired final pose using a sequence of pushing operations.

One goal of *dextrous* manipulation is to reorient a part while it is held in the hand [12]. In a multi-fingered hand, a subset of fingers grasp the part while the other fingers move to a new grasp location. After establishing a stable grasp, the first set of fingers is free to be relocated to a new grasp. Since this is analogous to “walking” across the part, such strategies are sometimes referred to as *fingerwalking* [17, 27]. Our approach uses a modified parallel-jaw gripper, gravity, and the support surface to achieve a similar goal.

3 Problem Statement

Consider the robot work cell in Figure 1; we make the following **assumptions** in developing our algorithms:

1. The worktable is a flat plane orthogonal to gravity at a known height.
2. The parallel-jaw gripper is able to translate with 3 DoF and to rotate about the gravity vector.
3. The gripper has a passive degree of freedom – a pivot axis parallel to the support plane.
4. The part is presented to the gripper in isolation. A sensing system (e.g. vision, light beams [31]) determines its exact initial configuration.
5. The gripper simultaneously makes two “hard” contacts with the part – point contacts with friction which permit rotation about the pivot axis. We assume that the part will rotate due to gravity and quickly stabilize with its center of gravity below the contact axis.

The **input** to the algorithm is:

- A polyhedral part \mathcal{P} stored as a boundary representation (B-rep).
- The part’s center of gravity taken to be the origin of the coordinate system used to define the B-rep.
- The coefficient of static friction μ_{static} .

A pivot grasp is **accessible** if both contact points are accessible in the direction of the grasp axis, *i.e.* they can be reached by fingertips moved in from $\pm\infty$ along the grasp axis. A grasp is **valid** for a given coefficient of static friction μ_{static} if it is accessible and no slippage occurs at the contact points. For the second condition, the grasp axis must lie within the friction cone at each of the two contact faces: $|\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}| \leq \cos \alpha$, where $\tan \alpha = \mu_{\text{static}}$.

The **output** of the exact pivots algorithm is the transition graph of exact pivots: a directed graph whose nodes are the m stable faces F_i of the convex hull with an arc between an ordered pair of nodes indicating the existence of a pivot grasp between them. For each arc, we compute the one-dimensional family of valid grasps affecting the transition between the corresponding nodes; each grasp from this family is described by a pair of points on the part. We may wish to do some processing over this family to pick out a grasp that is optimal under some criterion, such as the minimum coefficient of friction required. In this case, the optimum friction coefficient may also be returned as output. The entire graph can be represented by an $m \times m$ transition matrix (see Fig. 6 for an example). For capturing pivots, the arcs correspond to optimal capturing pivot grasps.

4 Part Configuration and the Grasp Axis

While a rigid body in \mathbb{R}^3 has six degrees of freedom, its mobility is reduced to five degrees of freedom when it is in contact with a plane. This five DoF Configuration space can be decomposed into: Rotation and translation in the support plane $\mathbb{R}^2 \times SO(2)$, and two other components of rotation which can be represented as a point on the unit sphere S^2 . Since planar rotations and translations are easily performed with a 4 DoF arm, we henceforth describe the part's configuration as a point on the unit sphere. As illustrated in figure 2, we affix a coordinate frame to the part with its origin at the part's center of gravity. Configurations of the part will be specified by a unit vector in the part frame; the vector is aligned with the gravity vector, and the part is just touching the work surface; *e.g.* configuration \hat{s} indicates that the face with normal \hat{s} is coincident with the support surface (unit vectors will be denoted with a hat).

The pivoting operation takes the part from a starting configuration \hat{s} to a final configuration \hat{f} . For exact pivot grasps, \hat{f} is another stable face. To rotate \hat{s} into \hat{f} , the axis of rotation (the pivot axis) must be orthogonal to both \hat{s} and \hat{f} . Let \hat{a} indicate the direction of this axis:

$$\hat{a} = \frac{\hat{s} \times \hat{f}}{|\hat{s} \times \hat{f}|}. \quad (1)$$

(Note that \hat{a} is undefined when \hat{s} and \hat{f} are parallel or anti-parallel. In these cases, exact pivot grasps are unnecessary or impossible). When lifted, the part will rotate due to gravity and settle in a configuration where the part's center of gravity is directly beneath the grasp axis. Thus the axis must be positioned such that it intersects a ray from the center of gravity in the direction $-\hat{f}$. Let λ be the distance from the axis to the center of gravity along this ray.

The family of grasp axes has the parametric equation:

$$\mathbf{a}_\lambda(t) = t\hat{a} - \lambda\hat{f}. \quad (2)$$

We use \mathbf{a}_λ to specify a particular grasp axis. Thus, the grasp axis must lie in the half-plane, \mathcal{A} , spanned by \hat{a} and $-\hat{f}$. We call this the *grasp plane*.

We next consider the grasp points on \mathcal{P} formed by the intersection of the axis with the part. Face F_i of the part lies in a plane with unit normal \hat{n}_i and at distance d_i from the origin, and is defined by $\hat{n}_i \cdot \mathbf{g} - d_i = 0$ where \mathbf{g} is a point on the face.

Substituting $\mathbf{a}_\lambda(t)$ for \mathbf{g} and solving for t , we obtain the contact point on face i :

$$\mathbf{g}_i(\lambda) = \frac{d_i + \lambda\hat{n}_i \cdot \hat{f}}{\hat{n}_i \cdot \hat{a}} \hat{a} - \lambda\hat{f}. \quad (3)$$

Note that the intersection is parameterized by λ , and over all positive λ , the intersection defines a ray. (Note: If $\hat{n}_i \cdot \hat{a} = 0$, there is no solution). For a finite polygonal face (possibly non-convex), the intersection will be a sequence of collinear segments taken from the ray, and λ will range over a disjoint set of intervals.

5 Planning Exact Pivot Grasps

Another way to view the set of grasp points is to consider the intersection of \mathcal{A} with the part: $\mathcal{A} \cap \mathcal{P}$ is an open polygon as illustrated in Fig. 3. Because the part may not be convex, $\mathcal{A} \cap \mathcal{P}$ may be composed of multiple non-convex open polygons. Consequently, a grasp axis (for a fixed value of λ) may intersect \mathcal{P} at more than two points.

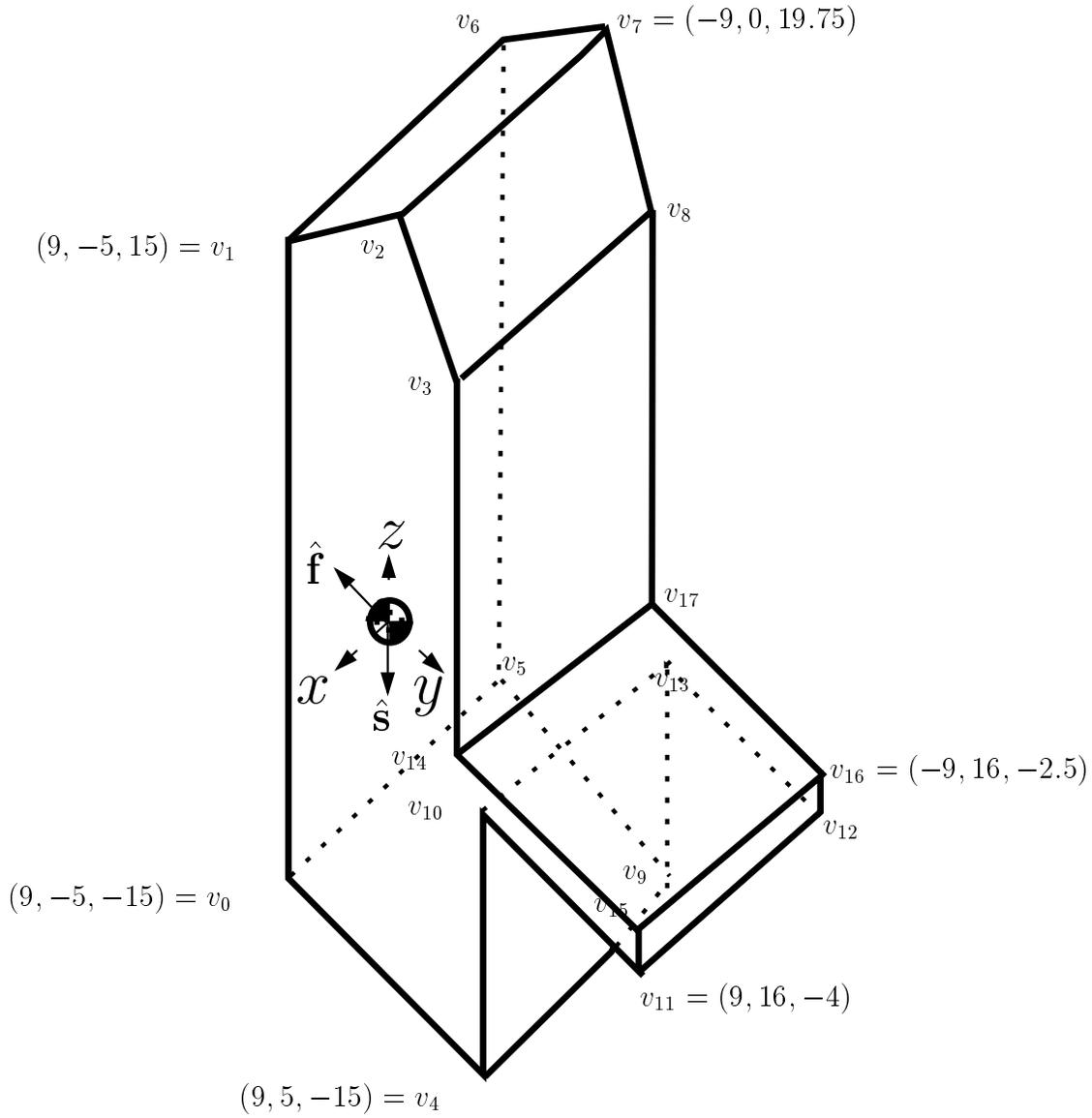


Figure 2: A polyhedral part with 18 vertices, 11 faces ($= n$), and 27 edges. We will use this part to illustrate the planning algorithms.

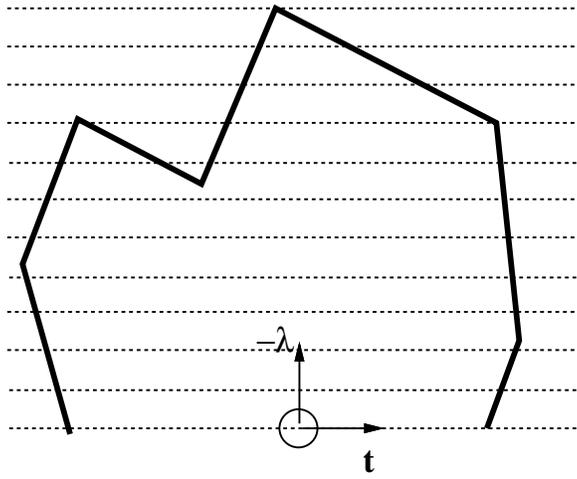


Figure 3: The open polygon formed by $\mathcal{A} \cap \mathcal{P}$. Potential grasp axes are parallel to the dotted lines.

Recall that a pivot grasp is **valid** if both contact points are accessible in the direction of the grasp axis and if the contacts will not slip under some coefficient of friction.

The outcome of these tests is a set $\Lambda = \{\lambda\}$ and a pair of maps $u: \lambda \rightarrow \mathbb{R}^3$ and $l: \lambda \rightarrow \mathbb{R}^3$ where $\forall \lambda \in \Lambda$ the pair of grasp points $(u(\lambda), l(\lambda))$ is valid; u and l are composed of g_i restricted to an interval of λ .

We now consider an efficient algorithm for computing the entire transition graph.

The algorithm:

From \mathcal{P} , compute its convex hull \mathcal{H} . A face of \mathcal{H} is stable when the projection of the center of gravity in the normal direction onto the face lies within the face; the stable faces become the nodes of the transition graph. For every ordered pair of stable faces of \mathcal{H} , whose normals are given by \hat{s} and \hat{f} , determine the set of valid grasp points (if there are any) that will pivot the part to \hat{f} as follows.

1. Determine the direction of the grasp axis \hat{a} from Eq. (1) and the grasp plane \mathcal{A} from Eq. (2).
2. Compute the intersection of \mathcal{A} with \mathcal{P} . This yields a collection of polygons \mathbf{P} in the grasp plane. Because \mathcal{P} may not be convex, $\mathcal{A} \cap \mathcal{P}$ may be composed of multiple polygons, each of which may not be convex. The size of \mathbf{P} (the number of edges and vertices in all polygons of \mathbf{P}) is $O(n)$ since each edge of \mathcal{P} can intersect \mathcal{A} at most once. Finally, since \mathcal{P} is defined by a B-rep such as a winged-edge structure, the edges of the polygons will be ordered and computable in linear time.¹
3. In the direction \hat{f} within the grasp plane, compute the upper \mathcal{U} and lower \mathcal{L} visible envelope of the polygon(s) \mathbf{P} . The visible envelope is the portion of \mathbf{P} visible from infinitely far away along $\pm\hat{a}$. Each envelope is a function of λ , and the edges of the envelope are ordered by increasing λ . By definition, points on these envelopes satisfy the accessibility condition. Each envelope can be computed in $O(n \log n)$ time [16]. However for polyhedra that are star-shaped with respect to their center of gravity, such as convex polyhedra, the intersection \mathbf{P} consists of a single simple polygon. In such a case, the envelope can be computed in linear time via a sweep technique.
4. For each edge of $\mathcal{U} \cup \mathcal{L}$ whose corresponding face has surface normal \hat{n} , determine if the face can be grasped by a point contact with friction in the direction \hat{a} according to: $\|\hat{a} \cdot \hat{n}\| \leq \cos \alpha$. This can be computed in $O(n)$ time.
5. Merge the two sorted envelopes \mathcal{U} and \mathcal{L} into a set $\Lambda = \cup \Lambda_i$ where each Λ_i is a closed interval of λ . Associated with each interval is the pair of functions $u_i(\lambda)$ and $l_i(\lambda)$ which return the grasp points. This merge step can be performed in linear time.
6. If $\Lambda \neq \emptyset$, create an arc in the transition graph between \hat{s} and \hat{f} and label it with Λ , $u(\lambda)$ and $l(\lambda)$.

¹Linear time complexity can also be seen by considering a triangulation of each face. A face with e edges can be triangulated into $O(e)$ triangles in $O(e)$ time [8]. Therefore all faces can be triangulated in $O(n)$ time into $O(n)$ triangles. Intersecting each triangle with the half-plane requires constant time.

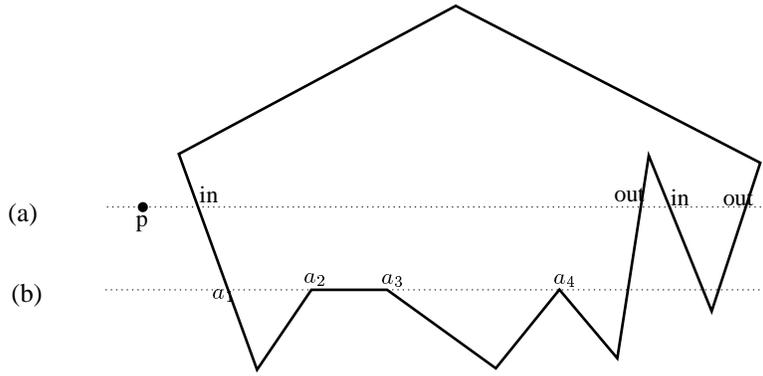


Figure 4: Intersecting a half-plane with a face. First the half-plane is intersected with every edge of the face to get a set of collinear points/subedges. In case of no degeneracies (a), one needs only to do a single point-in-face test to get the desired result. In case of degeneracies, more tests are required. Specifically, each $(a_j + a_{j+1})/2$ is tested for inclusion in the face. If it exists in the face, then the portion from a_j to a_{j+1} is part of the result.

The complexity of the algorithm is dominated by the construction of the visible envelopes; since there are $O(m^2)$ pairs of stable faces, the complexity of constructing the entire transition graph is $O(m^2 n \log n)$. For star-shaped (wrt the center of gravity) polyhedra, this reduces to $O(m^2 n)$. Notice that by setting the input $\mu_{\text{static}} = \infty$, or alternatively, by skipping Step 4 above, we obtain the family of accessible grasps instead of valid grasps.

If $\Lambda \neq \emptyset$, there are various criteria for selecting an optimal grasp from the set of valid grasps. One criterion is to select the grasp that requires the smallest coefficient of static friction μ to successfully grasp it, *i.e.* the grasp that minimizes the largest of the angles between the two surface normals and the grasp axis. Because the angles are constant over each interval Λ_i , this criterion alone returns an interval of grasps. Within this interval, the midpoint of the interval can be taken as the safest grasp; this will permit maximal error in the z -direction when positioning the gripper prior to grasping.

5.1 Implementation

We implemented the algorithm using *Maple V*, a commercial symbolic math package, and routines from a C++ geometry package developed at Utrecht University by G-J. Giezevan [13]. Given the model of a polyhedron, our program computes the transition graph: for each arc, the optimal (requiring least friction) grasp is computed. Now, consider some specifics of the implementation:

Intersecting \mathcal{A} with the input polyhedron The grasp half-plane \mathcal{A} is computed as described in Equations 1 and 2. For each of the n faces of the polyhedron, we must compute the intersection with \mathcal{A} . For face i , we compute the intersection of \mathcal{A} with every edge. If the intersection is not empty, it can be a single point on the edge or a subset of the edge.

These subedges (or points) taken over all the edges of face i are collinear and are sorted in left-to-right order by taking the projection of the center of gravity onto the plane containing face i and sorting according to the signed distance from p .

We next extract the desired intersection: the portion of \mathcal{A} contained within face i . Suppose we know whether p belongs in face i . Then, in the absence of degeneracies, one could simply do an alternation of ins and outs as shown in Fig. 4(a); the portion between an “in” and the next “out” is part of the result. However, this clearly wouldn’t work in case of Fig. 4(b). Therefore what we do is the following. We label the critical points a_1, a_2, \dots as shown. We then test point-in-face for each midpoint $(a_j + a_{j+1})/2$ in face i . If the midpoint lies in the face, we include the portion from a_j to a_{j+1} in the result. While this requires more point-in-face tests, it handles degeneracies properly.

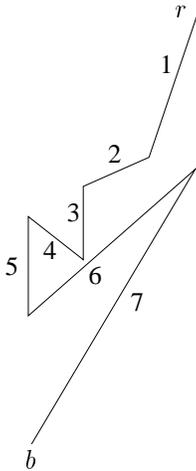


Figure 5: Computing the right envelope of a simple polygon. The figure considers computing the right envelope of the portion of the polygon from its right-most point r to its bottom-most point b . The edges are considered in the order 1,2, to 7. Depending on the current state of the envelope, a new edge is either ignored (Edge 4), is simply appended (Edges 1,2,3), a part of it is appended (Edge 5), or it replaces a portion of the current envelope and has to be *wholly* included in the new envelope (Edges 6,7).

Computing envelopes The final step in the implementation is computing the left and right envelopes of the set of edges obtained in the previous step (intersecting \mathcal{A} with the polyhedron). The intersection will be a set of polygons but the set of edges are not ordered yet. So we first order the edges into chains by doubling each edge and giving them opposite directions. We then sort these directed edges by coordinates of starting point. This enables us to compute the neighbor of each edge since neighbors share one end-point and thus we can organize the edges in polygon order. (We assume, as stated before, a simple polyhedron which implies that the polygons of intersection will be simple: no two edges cross each other except at end points and no three edges share any point in common). Next we compute the envelopes for each polygon. This is done via a simple sweep technique. Briefly, to compute the envelope from the right we begin at the right-most point r and conduct two traces: one ends at the bottom-most point b and the other at the top-most point t . Note that these three extremal points surely belong to the envelope. Consider the trace from r to b . We consider the sequence of edges one by one. Let b^* denote the the bottom-most point considered after looking at some l edges in the sequence. The $l + 1$ th edge either (i) lies fully to the left of the current right envelope (ii) fully to the right or (iii) a connected subset of it lies to the right and another connected subset lies to the left. These three cases can be distinguished by testing only the end-points of the $l + 1$ th edge. See Fig. 5. There are 7 edges from r to b . Case (i) is easy to handle: simply ignore the edge and carry on (Edge 4). In case (ii), we need to either simply append the edge (Edges 1,2,3) or we need to erase a portion of the current envelope and replace it with the $l + 1$ th edge (edge 6 and edge 7). To handle case (iii), we need to look at b^* and the portion of the $l + 1$ th edge that “peeks” from underneath b^* is the required portion to be appended to the current envelope (Edge 5).

After computing the right envelope for all the polygons, these are merged to compute the correct collective envelope. Same for the left envelope. The left and right envelopes are swept across and the midpoint of the subset of grasp point pairs that require the minimum friction angle is output.

5.2 Results

For the part shown in Fig. 2 modeled with 18 vertices and 11 faces ($= n$), the convex hull has 6 stable faces ($= m$). The transition graph, with 30 edges ($m(m - 1)$) along with the most slip-resistant exact grasp for each edge, was computed in 34 seconds on a Silicon Graphics workstation (R4400 processor running at 150 MHz, 96.5 SPECfp92, 90.4 SPECint92).

The matrix of transitions is illustrated in Fig. 6. The left column indicates the six initial poses, and the top row shows

the six possible final stable poses. The rest of the matrix displays the pivot grasps: cell (i, j) has the part drawn in its i stable configuration and specifies the pivot grasps that moves the part to the j th as follows. Thickened lines on the part indicate the left and right envelopes $(u(\lambda), v(\lambda))$, *i.e.* the set of accessible pairs of points affecting the transition. With infinite friction, each pair of points is valid; otherwise, only a subset of them is. A pair of points requiring the minimum friction is marked with disks in each cell. (See the following three figures for more detail.) The value of this minimal value of μ_{static} is at in the upper right hand corner. Notice that the cells that involve transitions between parallel faces of contact, $(4, 6)$ and $(6, 4)$, are empty: no single-step pivot grasp exists for these transitions.

We zoom into a few cells of the matrix to clarify detail. Fig. 7 shows cell $(2, 6)$: the envelopes form single segments on a pair of vertical parallel faces, and the optimal grasp consists of opposing points on these faces, and therefore any infinitesimally small friction is sufficient to affect the transition. In fact, notice from Fig. 6 that all transitions from Configuration 2 have this property. (This is also true for all transitions *to* Configuration 2, and also to Configuration 5 which is intuitively the “most stable” configuration). Fig. 8 shows Cell $(1,4)$ which is similar in that the envelopes are single opposing segments from parallel faces, in fact the same faces as in the previous figure; the computed grasp points are midpoints of these segments. However, the faces are no longer vertical and therefore the minimum coefficient of friction required is away from zero: 0.875. Finally, Cell $(4, 3)$ is shown expanded in Fig. 9. This cell is an example of a case where the envelopes consist of more than one segment. Here the optimal pair of grasp points come from orthogonal faces, and the lowest possible value that μ_{static} should be is 1.187.

6 Planning Capturing Pivot Grasps

When a part is placed in contact with the supporting plane in a configuration other than a stable one, it will tumble until it settles to one of the m stable poses. Following ideas presented by Brost [5], the space of initial contact configurations S^2 can be partitioned into a set of m disjoint capture regions, each of which contains a stable pose. From any configuration in a region $\mathcal{C}(\hat{\mathbf{f}})$, the part will converge to the corresponding stable pose $\hat{\mathbf{f}}$. Assuming only dissipative dynamics, the part’s potential energy, written as a function of the configuration $u: S^2 \rightarrow \mathbb{R}$, can be used to determine the capture regions [21]. Since u is non-smooth, stratified Morse theory can be applied to determine and classify the critical points of u ; a subset of the equipotential contours through the “saddle-like” points define the boundary of the capture regions [20].

Consider for example, the part shown in Fig. 2; contact only occurs along its convex hull shown in Fig. 10a. First, the sphere of configurations is stratified according to the generalized normal of the hull. The supporting plane only contacts a face for a single configuration. It contacts an edge along a curve of configurations given by the convex combination of the normals of the two incident faces, and it contacts a vertex along a region of sphere. Contact along an edge is depicted by the thicker arcs in Fig. 10b,c. It is shown in [20] that the boundary $\partial\mathcal{C}(\hat{\mathbf{f}})$ of the capture region $\mathcal{C}(\hat{\mathbf{f}})$ for configuration $\hat{\mathbf{f}}$ is composed of arcs of circles on the sphere. These arcs arise as configurations of constant potential energy u when a particular vertex \mathbf{v} is in contact with the supporting plane. Each circular arc can be written parametrically in t by:

$$\hat{\mathbf{q}}(t) = \cos(t)\mathbf{i} + \sin(t)\mathbf{j} + \mathbf{k} \quad (4)$$

where $|\mathbf{i}| = |\mathbf{j}| = \sqrt{1 - u^2/|\mathbf{v} - \mathbf{c}|^2}$, $\mathbf{k} = \frac{u}{|\mathbf{v} - \mathbf{c}|}(\mathbf{v} - \mathbf{c})$, and \mathbf{i}, \mathbf{j} and \mathbf{k} are orthogonal. The limits of the interval of t are determined by the configurations where rolling about the vertex leads to contact with an edge. In Figs. 10b,c, the thin curves delineate the capture regions.

Here, we extend the algorithm for exact pivoting grasps and consider the set of grasps where the part pivots to a configuration within a capture region. The set of feasible grasps then expands from being one dimensional in the case of exact grasps to being three dimensional. The extended algorithm outlined below has not been fully implemented.

6.1 Valid grasps

Recall that a pivot grasp is **valid** if both contact points are accessible in the direction of the grasp axis and if the contacts will not slip under some coefficient of friction. For capturing pivot grasps, the part’s configuration after pivot, $\hat{\mathbf{q}}$, can point anywhere within the capture region $\mathcal{C}(\hat{\mathbf{f}})$ of the stable configuration $\hat{\mathbf{f}}$. In Section 5, it was shown that the set of valid grasp axes (if non-empty) defines a one dimensional set which can be parameterized by λ . Here, we will see that the set of valid grasps is generically three dimensional.

		1	2	3	4	5	6
Final Config.							
Init. Config.							
1			0	0	0.9	0	0.9
2	0		0	0	0	0	
3	0	0		1.2	0	1.2	
4	0.9	0	1.2		0		
5	0	0	0	1		1	
6	0.9	0	1.2		0		

Figure 6: The matrix of transitions for the part with six stable configurations. Cell (i, j) indicates the family of accessible pivot grasps that will move configuration i to configuration j ; the frictionally optimal grasp from among this family is shown as a pair of disks. Numbers in the upper right-hand-corner of each cell indicate the minimal required coefficient of friction. Note that no single pivot grasp exists for cell $(4,6)$ or $(6,4)$; this transition requires a sequence of two grasps, regrasping after, for example placing the part in configuration 5.

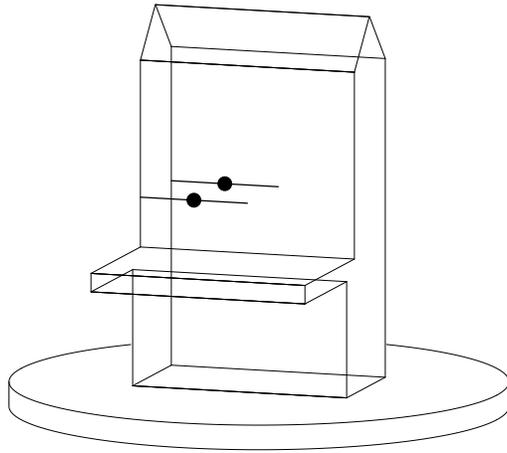


Figure 7: Transition of Cell (2,6).

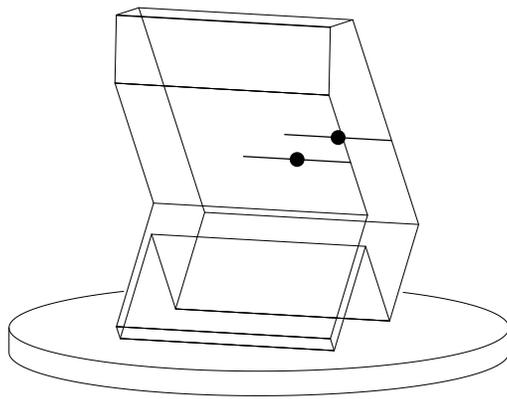


Figure 8: Transition of Cell (1-4).

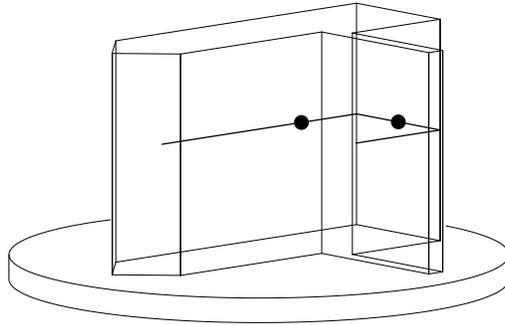


Figure 9: Transition of Cell (4,3).

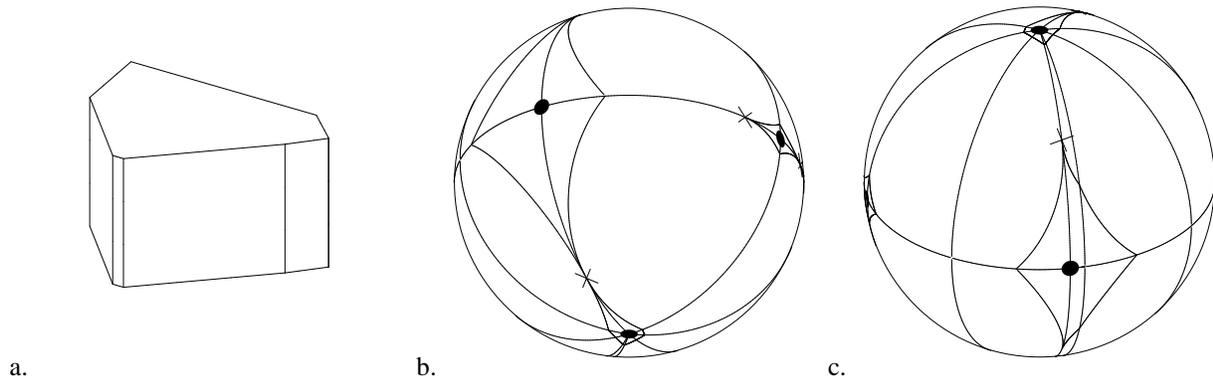


Figure 10: Capture regions: a. The convex hull for the polyhedral part in Fig. 2; b,c. The capture regions.

As discussed in Section 4 for a part resting on an initial face in configuration \hat{s} , a grasp axis can be determined from a unit vector \hat{q} (the configuration after pivoting) and a positive scalar λ . In the previous section, the configuration after pivoting was the final desired configuration or $\hat{q} = \hat{f}$. For a given initial configuration \hat{s} and a pair of faces F_i and F_j , the set of valid grasp axes can be given by:

$$\mathcal{G}_{i,j} = \{(\hat{q}, \lambda) : \hat{q} \in S^2, \lambda \in \mathbf{R}^+\}$$

subject to the following conditions.

1. \hat{q} must be in the capture region of \hat{f} . *i.e.* $\hat{q} \in \mathcal{C}(\hat{f})$.
2. $\mathbf{g}_i(\hat{q}, \lambda) \in F_i, \mathbf{g}_j(\hat{q}, \lambda) \in F_j$
3. $\mathbf{g}_i(\hat{q}, \lambda)$ and $\mathbf{g}_j(\hat{q}, \lambda)$ are accessible in the direction $\hat{a}(\hat{q})$.
4. $|\hat{a}(\hat{q}) \cdot \hat{n}_i| \leq \cos \alpha$ and $|\hat{a}(\hat{q}) \cdot \hat{n}_j| \leq \cos \alpha$
5. $\text{sign}(\hat{a}(\hat{q}) \cdot \hat{n}_i) = -\text{sign}(\hat{a}(\hat{q}) \cdot \hat{n}_j)$

where $\mathbf{g}_i(\hat{q}, \lambda)$ is given by equation (3) and $\hat{a}(\hat{q})$ is given by equation (1) with \hat{q} replacing \hat{f} .

In general each of these conditions is independent and since none of them defines an equality constraint, the valid set $\mathcal{G}_{i,j}$ will generically be empty or three dimensional. The set of all valid grasp axes is given by: $\mathcal{G} = \cup \mathcal{G}_{i,j}$. Note that Conditions 1 and 4 do not involve λ whereas Conditions 2 and 3 involve both λ and \hat{q} .

To simplify the presentation in the rest of this paper, we will only consider the polyhedron \mathcal{P} to be convex. This assumption has the following immediate implications: Every grasp axis satisfying Condition 2 will be accessible (*i.e.* it satisfies Condition 3) since every point on every face is accessible when \mathcal{P} is convex. Additionally, since there are at most 2 isolated intersections between a line and a convex polyhedron, specifying a grasp axis uniquely determines the two grasp points. Unless the grasp axis lies in a face where $\hat{a} \cdot \hat{n} = 0$, Condition 5 will be satisfied for any grasp satisfying Condition 2 since the axis intersects \mathcal{P} in only two points.

Note that constraints 1, 2 and 4 can be expressed as polynomial inequalities, and therefore the set of valid grasps is a semi-algebraic set. There are well established techniques, *e.g.* Collins's cylindrical algebraic decomposition [2] or Canny's roadmap [6], for characterizing a semi-algebraic set including determining if it is empty. Thus, a naive algorithm for determining the complete set of valid grasps which will pivot \mathcal{P} into the capture region of \hat{f} is to characterize $\mathcal{G}_{i,j}$ for all distinct pairs of faces i and j .

6.2 Optimal grasp selection

In the previous section, we observed that the set of valid grasps \mathcal{G} defines a three dimensional set. Explicitly computing this set is rather expensive, and so instead we will select a subset of \mathcal{G} by posing some notion of an optimal grasp and then optimizing the criteria over the set \mathcal{G} . There are a number of valid criteria (See [9]), and here we will consider a rather simple one; the same basic ideas can be applied to others. For a distinct pair of faces F_i and F_j , the criterion is:

$$\mathcal{O}_{i,j}(\hat{q}, \lambda) = \min(|\hat{a}(\hat{q}) \cdot \hat{n}_i|, |\hat{a}(\hat{q}) \cdot \hat{n}_j|) \quad (5)$$

The optimal grasp axis $(\hat{q}, \lambda) \in \mathcal{G}_{i,j}$ is one which maximizes $\mathcal{O}_{i,j}$, and this can be applied to all pairs of distinct faces. What does this criterion say? The dot products are related to the angle between the grasp axis and the face, and the minimum selects the larger angle. When the angle is small, the coefficient of friction μ does not have to be large for the contact to be stable. Thus, the optimal grasp also provides an upper bound on the coefficient of friction; that is, μ must be less than the optimal $\mathcal{O}_{i,j}$.

Note that for a given pair of faces F_i and F_j , the criterion (5) does not involve the grasp height λ . Thus, when pivoting directly onto \hat{f} as in Section 5, all valid grasps within an interval Λ_i (*i.e.* for a specific pair of faces) are equivalent. Furthermore, Condition 4 given above can be easily checked by selecting the optimal grasp which satisfies Conditions 1 and 2 and then checking Condition 4. As discussed above, we have assumed away Condition 3 by only considering convex polyhedra.

Note that \mathcal{G} is neither open nor closed since we have the constraint $\lambda > 0$. For the optimization to be well defined, we consider optimizing $\mathcal{O}_{i,j}$ over the closure of \mathcal{G} which will still be denoted \mathcal{G} . The maximum of $\mathcal{O}_{i,j}(\hat{q}, \lambda)$ over the closure of \mathcal{G} may either lie in the interior or on the boundary $\partial\mathcal{G}$. Since $\mathcal{O}_{i,j}$ is multimodal, a two step procedure is

required: (I) the local maxima of $\mathcal{O}_{i,j}(\hat{\mathbf{q}}, \lambda)$ over all $S^2 \times \mathbf{R}$ are computed and then only those contained within \mathcal{G} are retained. (II) $\mathcal{O}_{i,j}$ is optimized with $(\hat{\mathbf{q}}, \lambda)$ restricted to $\partial\mathcal{G}$. The maximum value of $\mathcal{O}_{i,j}$ from these two steps is the global max.

Note that when maximizing $\mathcal{O}_{i,j}$ over $\partial\mathcal{G}$, part of the boundary of \mathcal{G} is given by grasps along the edges and vertices of \mathcal{P} . In this paper, we assume point fingertip contact along a face. When considering a grasp along an edge or vertex, we assume that contact occurs on a particular face at the position of the edge or vertex. Because of the inability to precisely place a grasp point, one may want to account for the uncertainty in grasp location in practice. One approach is to only consider grasps that are further than some ϵ from any edge. Effectively, this shrinks the faces yielding smaller convex polygons.

Let us first consider step (I). Note that Eq. (5) is non-smooth, and it is maximized when either:

- i. $|\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_i|$ is a local maximum and $|\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_j| > |\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_i|$
- ii. $|\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_j|$ is a local maximum and $|\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_i| > |\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_j|$
- iii. $|\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_i| = |\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_j|$

These three conditions are subject to the two constraints: $|\hat{\mathbf{a}}| = 1$ and $\hat{\mathbf{a}} \cdot \hat{\mathbf{s}} = 0$. Conditions (i) and (ii) are symmetric, and Lagrange multipliers can be used to solve for the optimal grasp axis location subject to these two constraints: For case (i) we have:

$$\hat{\mathbf{a}} = \frac{1}{|(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{s}})\hat{\mathbf{s}} - \hat{\mathbf{n}}_i|} [(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{s}})\hat{\mathbf{s}} - \hat{\mathbf{n}}_i] \quad (6)$$

provided $|\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_j| > |\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_i|$. Geometrically, this corresponds to projecting $\hat{\mathbf{n}}_i$ onto the support plane.

Because of the fifth condition for a grasp to be valid, we can write Condition (iii) as $\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_i = -\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}_j$. The grasp axis direction satisfying this condition and the two constraints is:

$$\hat{\mathbf{a}} = \frac{1}{|\hat{\mathbf{s}} \times (\hat{\mathbf{n}}_i + \hat{\mathbf{n}}_j)|} \hat{\mathbf{s}} \times (\hat{\mathbf{n}}_i + \hat{\mathbf{n}}_j) \quad (7)$$

Geometrically, this condition corresponds to the projection of the bisector of $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_j$ onto the support plane.

The conditions given by both (6) and (7) only specify the direction of grasp axis $\hat{\mathbf{a}}$, and not the actual axis. From Section 4, the set of configurations $\hat{\mathbf{q}}$ after pivoting is orthogonal to $\hat{\mathbf{a}}$ or $\hat{\mathbf{q}} \cdot \hat{\mathbf{a}} = 0$. This defines a great circle on S^2 . The optimization criteria does not constrain λ (except by definition it is positive). Thus, the set of grasps which maximize $\mathcal{O}_{i,j}$ is two dimensional. The intersection of the great circle with the capture region can be easily computed, and since the $\mathcal{C}(\hat{\mathbf{f}})$ is not convex, this yields a set of arcs. This ensures that Condition (1) is satisfied. The set of optimal grasps is further pruned by Condition (2) that the grasp points must lie on the two faces. The intersection of the grasp axis with a face for a fixed $\hat{\mathbf{q}}$ will lie on a line parameterized by λ . Since the face is convex by assumption, the intersection will actually only lie on a single line segment, and the endpoints of this segment can be found from the edges of the face. Consequently, the optimal grasp can be specified by a pair of functions $\lambda_{min}(\hat{\mathbf{q}})$, $\lambda_{max}(\hat{\mathbf{q}})$ where $\hat{\mathbf{q}}$ is restricted to the computed arcs of the great circle.

The two dimensional set described above specifies the set of grasps that optimize the grasp criterion given in (5) for grasps confined to the interior of \mathcal{G} . We now consider the possible optimal grasps on the boundary $\partial\mathcal{G}$ of \mathcal{G} . $\partial\mathcal{G}$ is itself non-smooth, and it can be stratified into two dimensional surfaces, one dimensional curves, and zero dimensional vertices. There will be ten different cases to consider.

6.2.1 Optimizing on surfaces of $\partial\mathcal{G}$

The surfaces of $\partial\mathcal{G}$ arise when either:

1. $\hat{\mathbf{q}}$ is restricted to the boundary of the capture region and λ is free
2. One of the grasp points is restricted to an edge of a face and $\hat{\mathbf{q}}$ lies in the interior of the capture region.

In the first case, $\hat{\mathbf{q}}$ lies on an arc of a circle as given in (4), and from (1), the unnormalized grasp axis direction can be written as $\mathbf{a}(t) = \hat{\mathbf{s}} \times \hat{\mathbf{q}}(t)$. After normalizing $\hat{\mathbf{a}}(t)$, the grasps satisfying (i) (or similarly case (ii)) can be found by differentiating $\hat{\mathbf{a}}(t) \cdot \hat{\mathbf{n}}_i$ with respect to t , and finding the values of t where this vanishes. Case (iii) leads to a polynomial equation in t which can be readily solved. Once the optimal t is found, the configuration $\hat{\mathbf{q}}$ after pivoting is given by (4). λ is restricted to an interval which is easily computed by intersecting the grasp plane with faces F_i and F_j .

The second type of surface of $\partial\mathcal{G}$, is a grasp point \mathbf{g}_i on an edge of face F_i formed by the intersection with face F_k whose implicit equation is $\hat{\mathbf{n}}_k \cdot \mathbf{g} - d_k = 0$. Substituting $\mathbf{g}_i(\lambda)$ from Eq. (3) and solving for λ yields:

$$\lambda(\hat{\mathbf{q}}) = \frac{d_i(\hat{\mathbf{n}}_k \cdot \hat{\mathbf{a}}) - d_k(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}})}{(\hat{\mathbf{n}}_k \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}) - (\hat{\mathbf{n}}_i \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_k \cdot \hat{\mathbf{a}})} \quad (8)$$

Note that $\hat{\mathbf{q}}$ is not restricted, and so the optimal grasp axis direction $\hat{\mathbf{a}}$ can be computed according to the two conditions given in (6) and (7) as discussed in case 1. The set of final configurations after pivoting form arcs of a great circle $\{\hat{\mathbf{q}} : \hat{\mathbf{q}} \cdot \hat{\mathbf{a}} = 0, \hat{\mathbf{q}} \in \mathcal{C}(\hat{\mathbf{f}})\}$, and (8) provides the means to compute $\lambda(\hat{\mathbf{q}})$.

6.2.2 Optimizing on curves of $\partial\mathcal{G}$

The one dimensional strata of $\partial\mathcal{G}$ correspond to four cases:

3. $\hat{\mathbf{q}}$ lies at a vertex of the arc of the capture region, and λ is free.
4. One of the grasp points is a vertex of a face, and $\hat{\mathbf{q}}$ lies in the interior of $\mathcal{C}(\hat{\mathbf{f}})$.
5. One grasp point is along an edge of face F_i while the other grasp point is along an edge of face F_j , and $\hat{\mathbf{q}}$ is in the interior of $\mathcal{C}(\hat{\mathbf{f}})$.
6. $\hat{\mathbf{q}}$ is restricted to an arc of $\partial\mathcal{C}(\hat{\mathbf{f}})$, and one grasp point lies on an edge of a face.

Case 3 is trivial since $\hat{\mathbf{q}}$ is given and so the grasp axis direction can be directly computed from Equation (1). Note that this case is also equivalent to the conditions of the exact planning algorithm of Section 5, and so this can be handled more efficiently.

Consider case 4 where the grasp axis passes through a vertex \mathbf{v} of face i . From (3), we have that the following vector constraint

$$\mathbf{v} = \frac{d_i - \lambda \hat{\mathbf{n}}_i \cdot \hat{\mathbf{q}}}{\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}} \hat{\mathbf{a}} + \lambda \hat{\mathbf{q}}$$

from which we can derive two independent constraints by considering the component of \mathbf{v} in the $\hat{\mathbf{a}}$ and $\hat{\mathbf{q}}$ directions and a bit of manipulation:

$$\begin{cases} \lambda = \mathbf{v} \cdot \hat{\mathbf{q}} \\ (\mathbf{v} \cdot \hat{\mathbf{a}})(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}) + (\mathbf{v} \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{q}}) - d_i = 0 \end{cases} \quad (9)$$

Note that the second constraint defines a curve implicitly in $\hat{\mathbf{q}} \in S^2$ independent of λ , and can therefore be used to compute $\hat{\mathbf{a}}$. The first constraint can then be used to compute λ as a function of $\hat{\mathbf{q}}$. To find the optimal grasp axis direction on this 1D stratum of $\partial\mathcal{G}$, we have to consider condition (i)–(iii) discussed above. For cases (i) and (ii), Lagrange multipliers can be used to find the extrema of $\hat{\mathbf{n}} \cdot \hat{\mathbf{a}}$ subject to the second constraint in (9). Case (iii) is specified by a system of three polynomial equations in $\hat{\mathbf{q}}$: the second of Equations (9), $\hat{\mathbf{q}} \cdot \hat{\mathbf{s}} \times (\hat{\mathbf{n}}_i + \hat{\mathbf{n}}_j) = 0$ (from Eq. (7)), and $\hat{\mathbf{q}} \cdot \hat{\mathbf{q}} = 1$; this system can be readily solved by numerous techniques such as homotopy continuation [25].

In the fifth case, the two grasp points lie on edges formed by the intersection of F_i and F_k (as in case 2 above) and by the intersection of F_j and F_l . Since $\lambda(\hat{\mathbf{q}})$ must be the same for both grasp points, Eq. (8) can be rewritten for both edges and equated yielding:

$$\frac{(d_i(\hat{\mathbf{n}}_k \cdot \hat{\mathbf{a}}) - d_k(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}))((\hat{\mathbf{n}}_l \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_j \cdot \hat{\mathbf{a}}) - (\hat{\mathbf{n}}_j \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_l \cdot \hat{\mathbf{a}}))}{(d_j(\hat{\mathbf{n}}_l \cdot \hat{\mathbf{a}}) - d_l(\hat{\mathbf{n}}_j \cdot \hat{\mathbf{a}}))((\hat{\mathbf{n}}_k \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}) - (\hat{\mathbf{n}}_i \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_k \cdot \hat{\mathbf{a}}))} = \quad (10)$$

This is an eighth degree polynomial equation in the elements of $\hat{\mathbf{q}}$, and defines a curve of configurations on the sphere. The optimal $\hat{\mathbf{q}}$ satisfying condition (i) and (ii) can be found by Lagrange multipliers subject to the above constraint and $|\hat{\mathbf{q}}| = 1$. This yields a system of polynomial equations which is again readily solved using homotopy continuation. The optimal $\hat{\mathbf{q}}$ according to condition (iii) is clearly characterized by a system of polynomial equations. Once the optimal $\hat{\mathbf{q}}$ is computed, it must be checked that $\hat{\mathbf{q}} \in \mathcal{C}(\hat{\mathbf{f}})$, and then λ is found from (8).

In the sixth case $\hat{\mathbf{q}}$ is restricted to the boundary of the capture region, and since $\mathcal{O}_{i,j}$ is independent of λ , the optimal grasp axis direction and final configuration after pivoting $\hat{\mathbf{q}}$ can be computed as in case 1. (8) is then used to compute λ .

6.2.3 The Vertices of $\partial\mathcal{G}$

For a pair of faces F_i, F_j , the vertices of the $\partial\mathcal{G}$ can be computed as points $(\hat{\mathbf{q}}, \lambda)$ in the grasp space. Once the coordinates of the vertices are computed, $O_{i,j}$ can be directly evaluated from (5). The vertices of $\partial\mathcal{G}$ occur in the following four cases:

7. $\hat{\mathbf{q}}$ is a vertex of an arc of $\partial\mathcal{C}(\hat{\mathbf{f}})$, and one of the grasp points is at an edge;
8. One grasp point is a vertex of F_i , and the other is along an edge of F_j ;
9. One grasp point is a vertex of F_i , and $\hat{\mathbf{q}}$ is along an arc of $\partial\mathcal{C}(\hat{\mathbf{f}})$;
10. Each grasp point lies on an edge, and $\hat{\mathbf{q}}$ lies on an arc of $\partial\mathcal{C}(\hat{\mathbf{f}})$.

Case 7 is very easy to handle. The optimality criteria can be easily determined as in case 1, and the grasp point \mathbf{g}_i along the edge is computed from (8).

Grasping at a vertex of F_i is characterized by Equations (9), and from the second equation, we have an implicit constraint on $\hat{\mathbf{q}}$. Substituting the parameterized equation for an arc of $\partial\mathcal{C}(\hat{\mathbf{f}})$ given by (4) leads to a single equation in t which can be readily solved. If $\mathbf{q}(t) \in \mathcal{C}(\hat{\mathbf{f}})$, then λ can be easily determined from the first of Equations (9).

In the ninth case, one of the grasp points is vertex \mathbf{v} of face F_i and the other contact point is on an edge of F_j formed with the intersection with F_k . The grasp axis can be written as $\mathbf{v} + t\hat{\mathbf{a}}$. Writing that the grasp axis must intersect the edge and that it is orthogonal to $\hat{\mathbf{s}}$ leads to the following system of linear equations in $\mathbf{a} = t\hat{\mathbf{a}}$:

$$\begin{cases} \hat{\mathbf{n}}_j \cdot \mathbf{a} = d_j - \hat{\mathbf{n}}_j \cdot \mathbf{v} \\ \hat{\mathbf{n}}_k \cdot \mathbf{a} = d_k - \hat{\mathbf{n}}_k \cdot \mathbf{v} \\ \hat{\mathbf{s}} \cdot \mathbf{a} = 0 \end{cases}$$

Once \mathbf{a} is computed, Equations (9) along with $|\hat{\mathbf{q}}| = 1$ and $\hat{\mathbf{q}} \cdot \mathbf{a} = 0$ can be used to compute $\hat{\mathbf{q}}$ and λ .

The tenth case is like the fifth case with the restriction that $\hat{\mathbf{q}}$ lies on a boundary arc of the capture region. Substituting $\hat{\mathbf{q}}(t)$ from (4) into Equation (10) leads to a single equation in t which is easily solved. If t is within the correct interval, $\hat{\mathbf{q}}$ and λ are then readily determined from (4) and (8).

7 Discussion

This paper describes how to plan a particular class of grasps that have not previously been considered. Pivot grasps allow a robot with only 4 active DoF to move a polyhedral part through 6 DoF. The ‘‘gap’’ is closed by introducing a pivoting axis between the parallel jaws of a simple gripper and exploiting the force of gravity to rotate parts as they are lifted off a support surface.

We have presented an $O(m^2n \log n)$ algorithm that builds the transition graph of pivot grasps. The algorithm is complete in the sense that whenever a valid pivot grasp exists, it will find one.

Perhaps surprisingly, the problem of planning exact pivots is in fact *solution-complete* (borrowing terminology from [14]) *i.e.* a valid pivot grasp always exists *if* the part’s center of gravity lies inside the part and grasp points have infinite friction. The interior center of gravity condition ensures that accessible grasps that effect any transition exist; the envelopes $u(\lambda), l(\lambda)$ cannot be empty. Therefore, with infinite friction, the part can always be picked up and the transition completed. Other conditions can also insure solution-completeness, for example parts that are cuboids (rectangular parallelepipeds) with any non-zero friction. The difference between these two examples is that with infinite friction, we can always affect the transition in a single pivot grasp while for cuboids with non-zero friction, a sequence of two grasps may be required, for instance, to invert the part onto a face with an anti-parallel normal. (Remark: Performing this inversion in a single grasp is possible if we allow the manipulator to ‘‘shake’’ since this configuration is meta-stable: *i.e.* by accelerating the manipulator in almost any direction after pick-up.)

For convex polyhedra, and in general, for polyhedra that are star-shaped with respect to the center of gravity, the complexity of computing the transition graph reduces to $O(m^2n)$. An open question is whether or not this complexity can be reduced when we only need to compute paths to a single desired final face from all other stable faces (as might be the case for parts feeding).

We have tested some of these pivot grasps in the lab using an Adept robot arm (see [7]). We plan to extend the planning algorithms in several directions. Firstly, We would like to consider ‘‘active’’ pivoting, where the pivot axis is

actuated by a small motor and hence pivoting does not rely on gravity. This would allow us to remove the constraint that \hat{a} intersects $-\hat{f}$.

Secondly, in addition to implementing the computation of optimal capturing pivot grasps, we would like to broaden the definition of capture regions. For example rather than requiring that the part settles to a unique final configuration, we might consider requiring that the part settle onto one of a *set* of stable configurations. For example, consider a pair of stable faces defined by two local minima separated by a saddle point. There is a larger region including these two capture regions which is guaranteed to converge to one of these two stable faces, but we do not know to which one. The pair of stable faces can be treated as a super node. Sensing can be used to determine which of the two stable faces is achieved after pivoting. Using grasps that pivot into these “super-capture regions” can improve the connectivity of the transition graph.

One issue we have neglected is possible collisions with other parts during execution of a pivot grasp. Collisions could be easily predicted by the vision system. One way to avoid collisions would be to maintain alternative pivot grasps for each desired transition and then choose one that would avoid collisions, if possible, during execution. Note that the flexible feeding system envisioned by Carlisle et al would recirculate parts that cannot be reoriented due to such interference.

To reduce run-time, we are planning to re-implement the exact pivot algorithm in a compiled language. We will integrate this into a larger system that will aid in the rapid setup of flexible part feeders. This system will also contain routines to predict feeder throughput by estimating the statistical distribution of stable poses for a given part. As stated at the outset, our long-range goal is to incorporate this analysis into a solid modelling package: as the designer creates a new part, he or she can immediately test the “feedability” of this part, perhaps modifying the shape accordingly.

Acknowledgments: We thank Brian Carlisle and John Craig for helpful feedback on this work, Otfried Schwarzkopf for discussions on computing envelopes, and Geert-Jan Giezeman for help with the implementation.

References

- [1] S. Akella and M. T. Mason. An open-loop planner for posing polygonal objects in the plane by pushing. In *Intl. Conf. on Robotics and Automation*. IEEE, May 1992.
- [2] D. Arnon, G.E. Collins, and S. McCallum. Cylindrical algebraic decomposition I and II. *SIAM J. Comput.*, 13(4):865–889, November 1984.
- [3] A. Blake. Computational modelling of hand-eye coordination. *Phil. Transactions, Royal Society of London B*, 337:353–360, 1992.
- [4] D. L. Brock. Enhancing the dexterity of a robot hand using controlled slip. Technical Report AI-TR 992, MIT, May 1987.
- [5] R.C. Brost. *Analysis of Planning of Planar Manipulation Tasks*. PhD thesis, Carnegie Mellon, 1991.
- [6] J.F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1988.
- [7] B. Carlisle, K. Y. Goldberg, A. S. Rao, and J. Wiegley. A pivoting gripper for feeding industrial parts. In *Intl. Conf. on Robotics and Automation*, pages 1650–1655. IEEE, May 1994.
- [8] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6:485–524, 1991.
- [9] I. Chen and J. W. Burdick. Finding antipodal grasps on irregularly shaped objects. *IEEE Trans. on Robotics and Automation*, pages 507–511, August 1993. Also in *Proc. Intl. Conf. Robotics & Automation*, 1992.
- [10] M. A. Erdmann, M. T. Mason, and G. Vaněček. Mechanical parts orienting: The case of a polyhedron on a table. *Algorithmica*, 10(2), August 1993. Special Issue on Computational Robotics.
- [11] B. Faverjon and J. Ponce. On computing two-finger force-closure grasps of curved 2d objects. In *International Conference on Robotics and Automation*. IEEE, May 1991.

- [12] R. S. Fearing. Simplified grasping and manipulation with dextrous robot hands. *IEEE Journal of Robotics and Automation*, RA-2(4), December 1986.
- [13] G.-J. Giezeman. *SpaGeo—A Library for Spatial Geometry*. Users Manual, Dept. Comput. Sci., Utrecht Univ., Utrecht, the Netherlands, January 1994.
- [14] K. Y. Goldberg. Completeness in robot motion planning. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, San Francisco, February 1994.
- [15] R. A. Gruben, T. C. Henderson, and I. D. McCammon. A survey of general purpose manipulation. *International Journal of Robotics Research*, 8(1), 1989.
- [16] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inform. Process. Lett.*, 33:169–174, 1989.
- [17] J. Hong, G. Lafferriere, B. Mishra, and X. Tan. Fine manipulation with multifinger hands. In *International Conference on Robotics and Automation*. IEEE, May 1990.
- [18] D. E. Koditschek. Robot assembly: Another source of nonholonomic control problems. In *American Control Conference*, pages 1627–1632, 1991.
- [19] D. E. Koditschek. An approach to autonomous robot assembly. *Robotica*, 1993.
- [20] D. Kriegman. Capture regions of curved objects and polyhedra. *Int. J. Robot. Res.*, 1994. In preparation.
- [21] D. Kriegman. Let them fall where they may: Computing capture regions of curved 3D objects. In *Intl. Conf. on Robotics and Automation*, pages 595–601. IEEE, May 1994.
- [22] D. J. Kriegman. Computing stable poses of piecewise smooth objects. *CVGIP: Image Understanding*, 55(2), March 1992.
- [23] K. Lynch. The mechanics of fine manipulation by pushing. In *International Conference on Robotics and Automation*. IEEE, May 1992.
- [24] X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *Int. J. Robot. Res.*, 9(1), February 1990.
- [25] A. Morgan. *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems*. Prentice Hall, Englewood Cliffs, 1987.
- [26] J. Pertin-Troccaz. Grasping: A state of the art. In *The Robotics Review I*, pages 71–98. MIT Press, 1989. edited by O. Khatib, J. J. Craig, and T. Lozano-Perez.
- [27] D. Rus. Dextrous rotations of polyhedra. In *International Conference on Robotics and Automation*. IEEE, May 1992.
- [28] J. K. Salisbury. *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Department of Mechanical Engineering, Stanford University, May 1982. Published in *Robot Hands and the Mechanics of Manipulation*, MIT Press, 1985.
- [29] P. Tournassoud, T. Lozano-Perez, and E. Mazer. Regrasping. In *Intl. Conf. on Robotics and Automation*. IEEE, May 1987.
- [30] J. C. Trinkle and R. P. Paul. Planning for dextrous manipulation with sliding contacts. *Int. J. Robot. Res.*, June 1990.
- [31] A. S. Wallack and J. F. Canny. A geometric matching algorithm for beam scanning. In *SPIE symposium on optical tools for manufacturing and advanced automation*, Boston, 1993.