Complete Algorithms for Feeding Polyhedral Parts using Pivot Grasps

Anil Rao, David Kriegman, Ken Goldberg

Abstract— **To rapidly feed industrial parts on an assembly line, Carlisle *et. al.* 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 [1]. When a part is grasped with two hard finger contacts and lifted, it pivots under gravity into a stable configuration. We refer to the sequence of picking up the part, allowing it to pivot, and replacing it on the table as a *pivot grasp*. We show that under idealized conditions, a robot arm with 4 degrees of freedom (DoF) can move (feed) parts arbitrarily in 6 DoF using 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 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 $O(m^2 n)$. Since pivot grasps may not exist for some transitions, multiple steps may be needed. Alternatively, we consider the set of grasps where the part pivots to a configuration within a "capture region" around the stable configuration; when the part is released, it will tumble to the desired configuration. Both algorithms are complete in that they are guaranteed to find pivot grasps when they exist.

Keywords— **Robotic Manipulation, Parts Feeders, Configuration Space, Motion Planning, Degrees of Freedom, Stability, Pose Planning, Friction, Parallel Jaw Gripper, Solid Modeling, Algorithms, Geometry.**

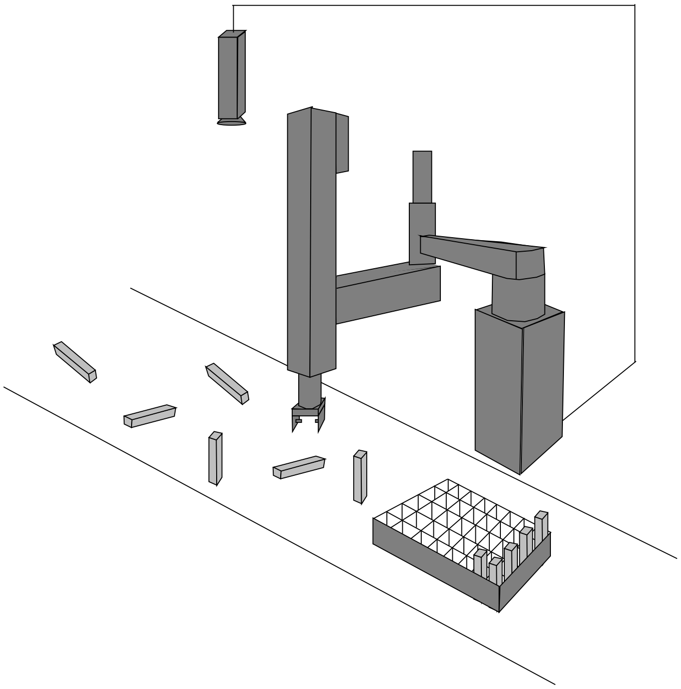

Fig. 1. The Flexible Part Feeder described in Carlisle *et al.* [1]. A 4 DoF robot arm and vision system for feeding polyhedral parts.

I. 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 six degree of freedom (DoF) robot arm by slicing the configuration space but did not provide a complete algorithm [2].

Automating grasp analysis is useful for rapid set-up of a parts feeding system using vision and a robot manipulator [1]. 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.* [1] 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 DoF manipulators, such as a SCARA arm or Robot World module, are kinematically limited to orienting parts about the vertical axis. However, we show that under idealized conditions, general reorientations of a part can be kinematically achieved with a single pivot grasp actuated by a 4 DoF. In practice, grasp accessibility and stability with respect to friction may prevent grasping along desired axes. Our planning algorithm checks these conditions and builds

This work was done when A. Rao was with the Department of Computer Science, Utrecht University, the Netherlands. Rao was supported by the ESPRIT Basic Research Action No. 6546 (Project PRoMotion). The author is now with Qualcomm, Inc., 6455 Lusk Blvd., San Diego, CA 92121-2779, arao@qualcomm.com

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

K. Goldberg is with the Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720-1735, USA., goldberg@ieor.berkeley.edu. Goldberg is supported by NSF Young Investigator Award IRI-9457523 and IRI-9123747 and by Adept Technology, Inc.

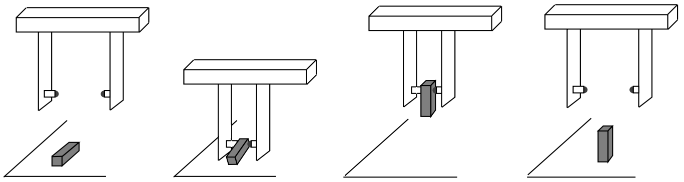

Fig. 2. A pivot grasp.

a matrix that can be used to plan multi-step pivot grasps for cases where a single pivot grasp is not accessible or frictionally stable.

Achieving an arbitrary part position and orientation using a manipulator with fewer than 6 DoF may appear counter-intuitive at first. We focus on rotations, since subsequent translation of the part is trivial. Consider the sequence of operations during a pivot grasp, illustrated in figure 2.

1. Rotate the gripper about the vertical axis.
2. Grasp the part and lift it so that the part rotates about the horizontal pivot axis and stabilizes.
3. Rotate the part about the vertical axis and set it down.

There are three rotations occurring in sequence. Note that the first and last rotations are independent because of the change in the grasp condition. During the first rotation about the vertical axis, the part moves with respect to the gripper but not with respect to the world. Whereas in the third 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 [3], [4].

Consider the two rotations that move the part with respect to the world. These are the second, or pivot, rotation and the third, or vertical rotation. The pivot rotation is about a horizontal axis. The purpose of the first rotation of the gripper without the part is simply to orient the pivot axis correctly in the horizontal plane. Can a sequence of two rotations, one about some axis lying in the horizontal plane and the second about the vertical axis, cover the space of all 3D rotations? We provide a positive answer in the Appendix.

The pivoting gripper can be modelled as having two hard point fingers that contact the part during grasping. In Step 1, the gripper is rotated to correctly position and orient the fingers; now squeezing the fingers establishes contact and also defines the pivot axis as the line connecting the two contact points. In turn, the center of gravity of the part and the pivot axis determine the magnitude and direction of the unactuated pivot rotation in Step 2. After the part stabilizes, it is rotated about the vertical 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 [5]. 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 $\mathcal{H}$ that is in contact with the support plane. In this paper, we consider pivot grasps that move a part from an initial stable configuration $\hat{\mathbf{s}}$ to a final stable configuration $\hat{\mathbf{f}}$. Given $\hat{\mathbf{s}}$ and $\hat{\mathbf{f}}$, deciding whether or not a *single* pivot grasp can achieve $\hat{\mathbf{f}}$ can be done in $O(n \log n)$ time.

Due to constraints on accessibility and friction, a single pivot grasp may be insufficient to move the part between an arbitrary pair of stable configurations.

Let us define a directed graph as follows: Each node is a distinct stable configuration and each directed edge represents a pivot grasp between the corresponding stable configurations. A path though this graph represents a sequence of pivot grasps (*i.e.* a plan), to move the part from some initial to a final stable configuration. If the transition graph is not strongly connected, some stable configurations can be unreachable regardless of how many pivot grasps are available. 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.

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 of $\mathcal{H}$. The algorithm is *complete* in the sense that whenever a sequence of pivot grasps exists, we are guaranteed to find it [6].

For cases where the transition graph is not strongly connected or the shortest path between nodes requires too many regrasping operations, we 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 can be a richer action set; previously unconnected nodes in the transition graph may now be connected. We call these "capturing" pivot grasps. While the position and orientation of the part in the plane may be uncertain after tumbling, a 4 DOF robot can easily correct this after sensing.

We begin below by reviewing related work and then define the problem and state our assumptions in Section III. Theory common to both types of pivot grasps is discussed in Section IV. Pivot grasps are treated in Section V, and capturing pivot grasps in Section VI. We implemented the planner as described in Section V-A. A preliminary and condensed version of this paper appeared in [7].

II. Related Work

Our results build on prior research in robot motion planning and grasp planning [8], [9]. We consider grasps with two frictional point contacts, also known as hard finger contacts [10]. Each contact allows forces pointing into the associated friction cone. In 3D, such a grasp cannot achieve form closure: the part is free to rotate about the contact axis. In the plane, Faverjon and Ponce [11] and later, Blake [12], considered computing frictional two-finger force-closure grasps. To achieve form closure, Markenscoff *et al.* [13] 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. 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.

One goal of *dextrous* manipulation is to reorient a part while it is held in the hand [14]. 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 *fingergaiting* [15], [16].

Brock [17] considered how a part could be moved within a grasp by adjusting applied forces at the fingers or the direction of the gravity vector relative to the grasp frame. Brock proposed the *constraint state map*, which partitions the space of control variables into regions based on contact type: point with friction, point without friction, no contact. This suggests how control variables can be adjusted to move from one qualitative contact type to another but does not directly suggest an algorithm for planning grasp configurations to achieve a specific final orientation of a part. Trinkle and Paul [18] provide a quantitative analysis of how controlled slip can be used to achieve a precise enveloping grasp of a part initially in contact with a flat support surface.

For polyhedral parts, Erdmann *et al.* [19] showed how to tilt an infinite plane to orient a given polyhedral part, regardless of its initial orientation. Both Akella and Mason [20] and Lynch [21] 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.

Our approach uses a modified parallel-jaw gripper, gravity, and the support surface to achieve an inexpensive variety of dextrous manipulation.

III. Problem Statement

Consider the robot work cell in Fig. 1; we make the following assumptions:

1. The worktable is a flat plane orthogonal to gravity at a known height.
2. The robot arm can translate the parallel-jaw gripper with 3 DoF and rotate it about the gravity vector.
3. The gripper has a passive degree of freedom – a pivot axis that is always parallel to the support plane.
4. The part is presented to the gripper in isolation. A sensing system (e.g. vision, light beams [22]) determines its exact initial configuration.
5. The gripper makes two simultaneous "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 pivot axis.

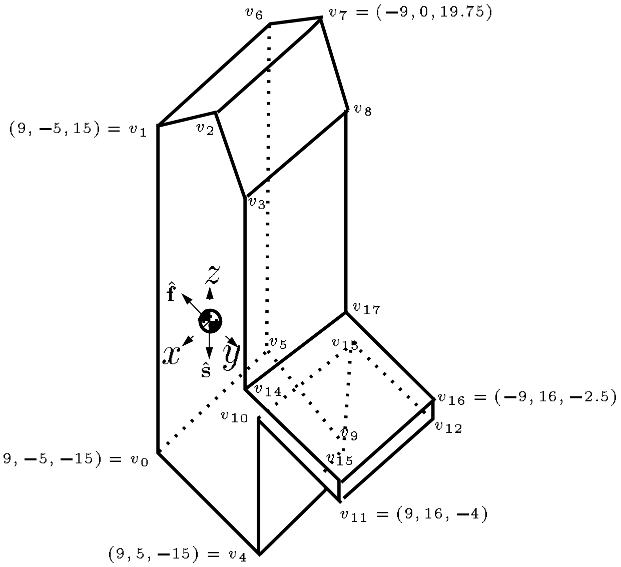

Fig. 3. A polyhedral part with 18 vertices, 11 faces (= n), and 27 edges. We will use this part to illustrate the planning algorithms.

The input to the algorithm is:

- A polyhedral part $\mathcal{P}$ stored as a boundary representation (B-rep).
- The part's center of gravity; this is taken to be the origin of the part's 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 pivot grasp is *valid* for a given coefficient of static friction 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: $|\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}| \geq \cos\alpha$, where $\tan\alpha = \mu_{\text{static}}$.

The output is the transition graph of pivot grasps: 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 a 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 to select a single grasp from this family that is optimal under some criterion, such as the minimum required coefficient of friction for the grasp to be successful. 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. 8 for an example).

IV. 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 Fig. 3, 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.

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

$$\hat{\mathbf{a}} = \frac{1}{|\hat{\mathbf{s}} \times \hat{\mathbf{f}}|} \hat{\mathbf{s}} \times \hat{\mathbf{f}}. \tag{1}$$

Note that $\hat{\mathbf{a}}$ is undefined when $\hat{\mathbf{s}}$ and $\hat{\mathbf{f}}$ are parallel or anti-parallel. In these cases, pivot grasps are either 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 $-\hat{\mathbf{f}}$. Let λ be the distance from the axis to the center of gravity along this ray. The family of grasp axes has the following equation parameterized in t:

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

We use $\mathbf{a}_\lambda$ to specify a particular grasp axis. Thus, the grasp axis must lie in the half-plane, $\mathcal{A}$, spanned by $\hat{\mathbf{a}}$ and $-\hat{\mathbf{f}}$. 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 $\hat{\mathbf{n}}_i$ and at distance d_i from the origin, and is defined by $\hat{\mathbf{n}}_i \cdot \mathbf{g} - d_i = 0$ 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:

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

The intersection is parameterized by λ, and over all positive λ, the intersection defines a ray. (Note: If $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}} = 0$, 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.

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. 4. 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.

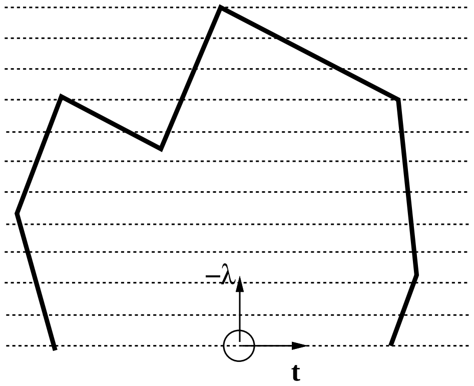

Fig. 4. The open polygon formed by $\mathcal{A} \cap \mathcal{P}$. Potential grasp axes are parallel to the dotted lines.

V. Planning Pivot 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 the given coefficient of friction. The algorithm outputs a set of valid grasps that can be represented by a set of intervals Λ of λ and a pair of maps $u\colon \lambda \rightarrow \mathbb{R}^3$ and $l\colon \lambda \rightarrow \mathbb{R}^3$ where $\forall \lambda \in \Lambda$ the pair of grasp points $(u(\lambda), l(\lambda))$ is valid; u and l are composed of $\mathbf{g}_i$ restricted to an interval of λ.

We now describe the algorithm for computing the transition graph.

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 define the nodes of the transition graph. For every ordered pair of stable faces of $\mathcal{H}$, whose normals are given by $\hat{\mathbf{s}}$ and $\hat{\mathbf{f}}$, determine the set of valid grasp points (if there are any) that will pivot the part from $\hat{\mathbf{s}}$ to $\hat{\mathbf{f}}$:

1. Determine the direction of the grasp axis $\hat{\mathbf{a}}$ from (1) and the grasp plane $\mathcal{A}$ from (2). (See Fig. 5.)
2. Compute $\mathbf{P} = \mathcal{A} \cap \mathcal{P}$. Because $\mathcal{P}$ may be nonconvex, $\mathbf{P}$ 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 $\hat{\mathbf{f}}$ within the grasp plane, compute the upper $\mathcal{U}$ and lower $\mathcal{L}$ visible envelope of $\mathbf{P}$. The visible envelope is the portion of $\mathbf{P}$ visible from infinitely far away along $\pm\hat{\mathbf{a}}$. 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 [24]. However for polyhedra that are star-shaped with respect to their center of gravity,

[1] Linear 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 [23]. Therefore all faces can be triangulated in $O(n)$ time into $O(n)$ triangles. Intersecting each triangle with the half-plane requires constant time.

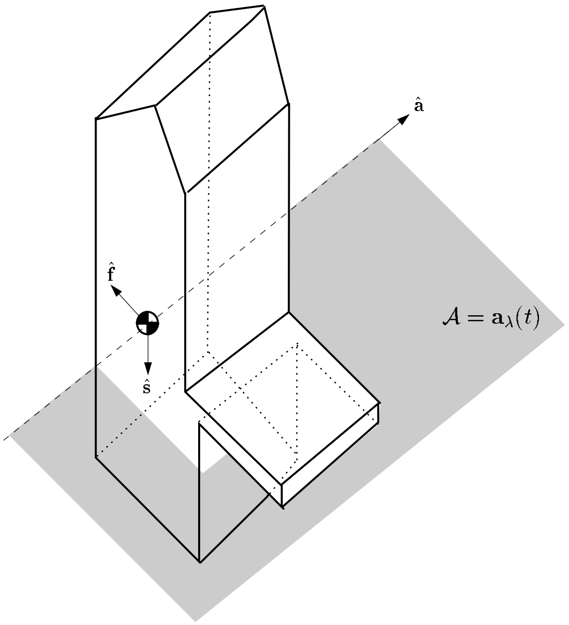

Fig. 5. The same part as before (Fig. 3) showing the vectors corresponding to the start configuration ($\hat{\mathbf{s}}$), the final configuration ($\hat{\mathbf{f}}$) and the grasp axis direction which is perpendicular to both ($\hat{\mathbf{a}}$). All three vectors pass through the center of mass. The shaded region is the grasp plane: the grasp axis is constrained to lie within this half plane and be parallel to $\hat{\mathbf{a}}$.

such as convex polyhedra, the intersection **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 $\hat{\mathbf{n}}$, determine if the face can be grasped by a point contact with friction in the direction $\hat{\mathbf{a}}$ according to: $|\hat{\mathbf{a}} \cdot \hat{\mathbf{n}}| \geq \cos\alpha$. This can be checked 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 $\hat{\mathbf{s}}$ and $\hat{\mathbf{f}}$, and label it with Λ, $u(\lambda)$ and $l(\lambda)$.

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 (with respect to the center of gravity) polyhedra, this reduces to $O(m^2 n)$.

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 representative grasp; this grasp will permit maximal gripper positional uncertainty.

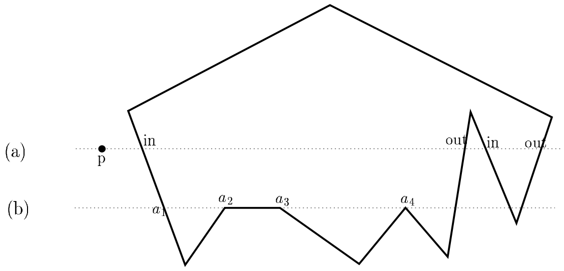

Fig. 6. Intersecting a half-plane with a face. First the half-plane is intersected with every edge of the face to get a set of collinear points/subedges. In case of no degeneracies (a), one needs only to do a single point-in-face test to get the desired result. In case of degeneracies, more tests are 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.

A. Implementation and Example

We implemented the algorithm using *Maple V*, a commercial symbolic mathematics package, and routines from a C++ geometry package developed at Utrecht University by G-J. Giezeman [25]. Given the model of a polyhedron, our program computes the transition graph: for each arc it computes the optimal grasp as defined in the preceding paragraph.

A.0.a 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. 6(a); the portion between an "in" and the next "out" is part of the result. However, this clearly would not work in case of Fig. 6(b). Instead, we label the critical points $a_1, a_2, ...$ 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.

A.0.b 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

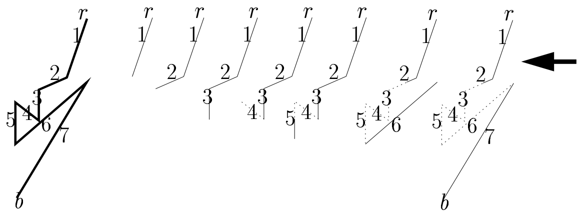

Fig. 7. Computing the right envelope of a simple polygon. The figure considers computing the right envelope of the portion of the polygon from its right-most point r to its bottom-most point b. The edges are considered in the order 1,2, to 7. Depending on the current state of the envelope, a new edge 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).

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 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. 7. 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.

For the part shown in Fig. 3 modeled with 18 vertices and 11 faces (= n), the convex hull has 6 stable faces (= m). The transition matrix, with 30 edges along with the most slip-resistant 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).

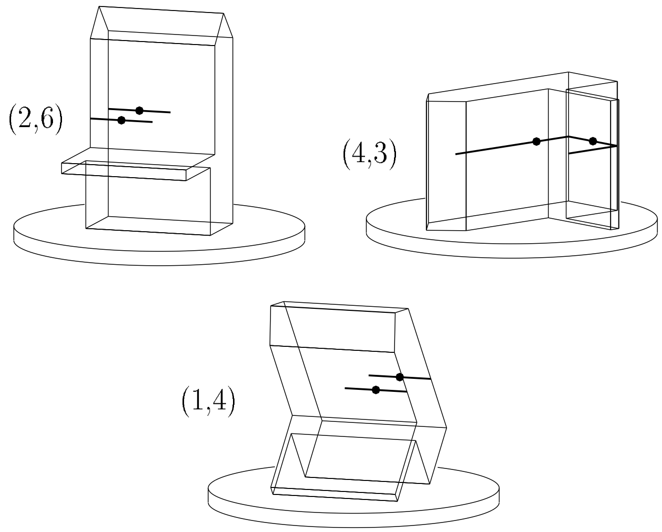

Fig. 9. Transitions in Cells (2,6), (1,4), and (4,3) of Fig. 8 shown expanded.

The matrix of transitions is illustrated in Fig. 8.

In Fig. 9, we expand a few cells of the matrix to clarify detail. In Cell $(2,6)$ the set of valid grasps lie on parallel vertical faces and include a single segment on each face. The optimal grasp consists of opposing points on these faces, and therefore any non-zero friction is sufficient to affect the transition. In fact, notice from Fig. 8 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. Cell $(1,4)$ is similar in that the set of valid grasps lie along segments on the same two parallel faces as in Cell $(2,6)$; the computed grasp points are midpoints of these segments. However, while these two faces are parallel, they are no longer vertical; therefore the minimum coefficient of friction required is nonzero: 0.875. Finally, Cell $(4,3)$ shows a case where the envelopes consist of multiple segments. Here the optimal pair of grasp points come from orthogonal faces, and the lowest possible value of μ_{static} is 1.187.

VI. 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 Brost [26], [27], 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 $u\colon S^2 \rightarrow \mathbb{R}$, can be used to determine the capture regions [28]. Since u is non-smooth, stratified

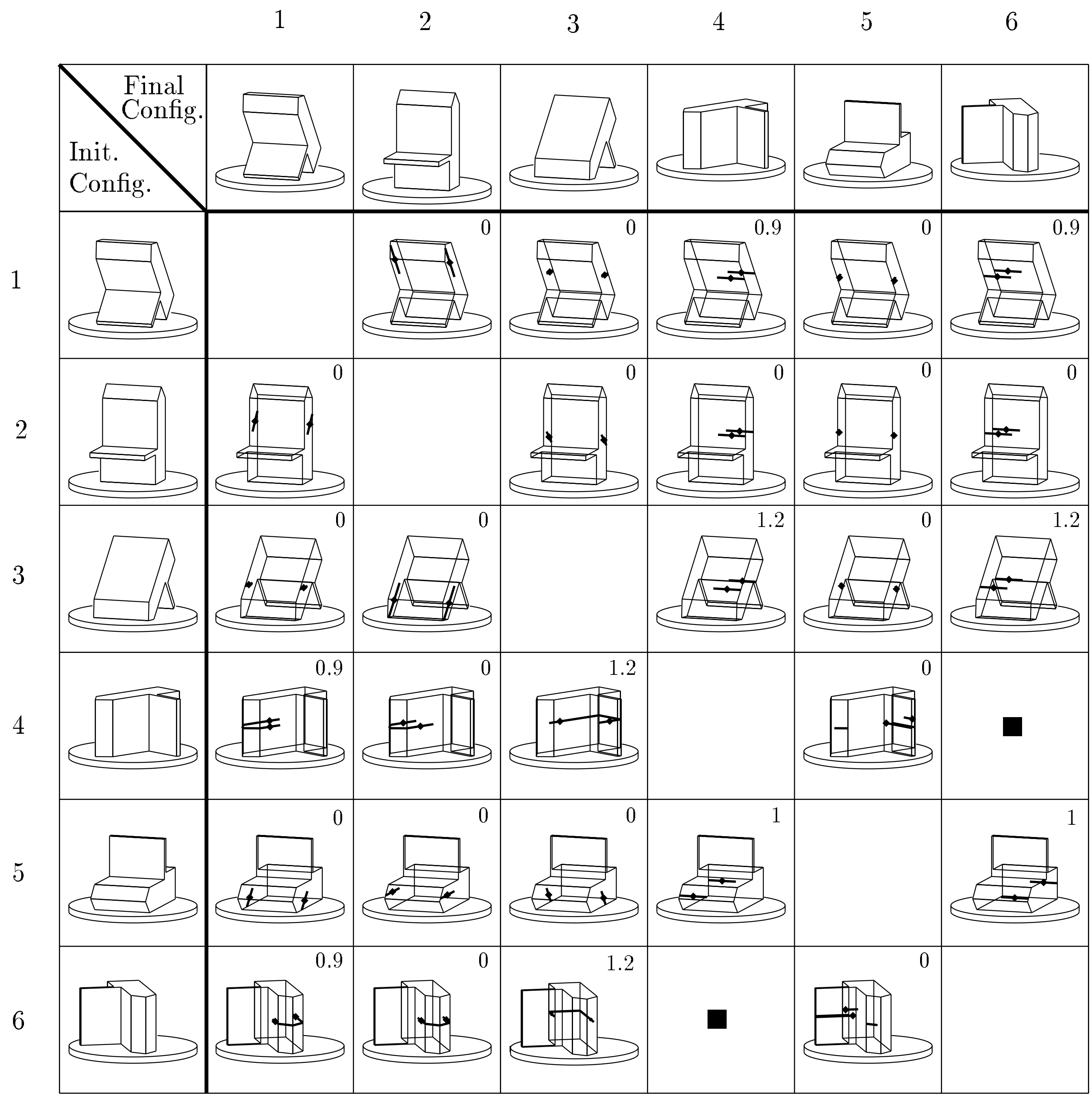

Fig. 8. The matrix of transitions for the part with six stable configurations. Cell (i, j) indicates the family of accessible pivot grasps that will move configuration i to configuration j. Thickened lines on the part indicate the left and right envelopes $(u(\lambda), v(\lambda))$, *i.e.* the set of accessible pivot grasps that will achieve the transition. The black dots indicate the frictionally optimal grasp from among this family. 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 pivot grasps through an intermediate configuration.

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 [29].

Consider for example, the part shown in Fig. 3; 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 [29] 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

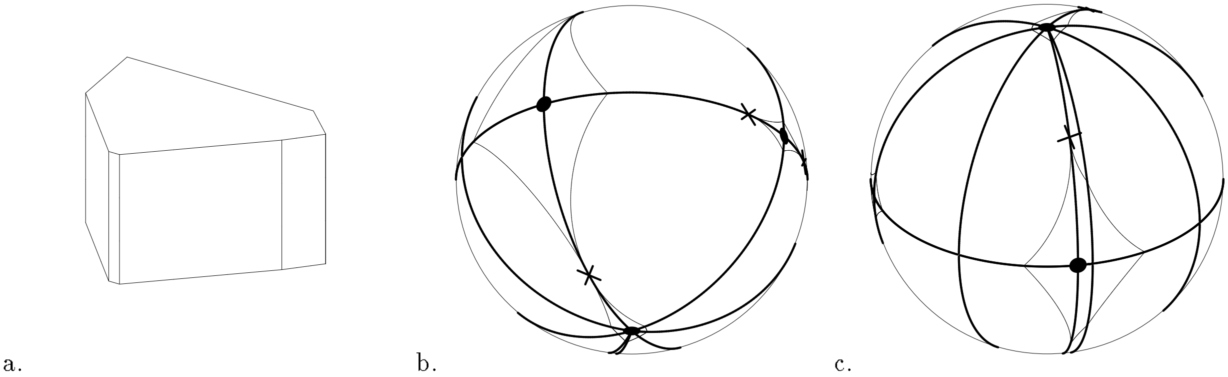

Fig. 10. Capture regions: a: The convex hull for the polyhedral part in Fig. 3; b,c: Capture regions for each stable pose are shown bounded by thin lines.

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:

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

where $|\mathbf{i}| = |\mathbf{j}| = \sqrt{1 - u^2/|\mathbf{v}|^2}$, $\mathbf{k} = \frac{u}{|\mathbf{v}|}(\mathbf{v})$, $\mathbf{v}$ is expressed in the part's coordinate system whose origin is at the center of gravity, 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 pivoting grasps and consider the set of grasps where the part pivots to a configuration within a capture region. The set of valid grasps then expands from being one dimensional (*i.e.* , the set Λ) to being three dimensional. This can be seen by noting that for each configuration $\hat{\mathbf{q}} \in \mathcal{C}(\hat{\mathbf{f}})$ within the capture region (a 2D set), the set of valid grasps that will pivot to $\hat{\mathbf{q}}$ is one dimensional (or empty). The algorithm outlined below has not been fully implemented.

A. Finding Capturing Pivot 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 $\hat{\mathbf{q}}$ after pivoting can lie anywhere within the capture region $\mathcal{C}(\hat{\mathbf{f}})$ of the stable configuration $\hat{\mathbf{f}}$. When the gripper releases the part, it will tumble to configuration $\hat{\mathbf{f}}$. Again, dropping may introduce uncertainty in the plane, but this can be easily corrected by the 4 DoF arm after another sensory step.

As discussed in Section IV for a part resting on an initial face in configuration $\hat{\mathbf{s}}$, a grasp axis can be determined from a unit vector $\hat{\mathbf{q}}$, the configuration after pivoting, and a positive scalar λ. For an initial configuration $\hat{\mathbf{s}}$ and a pair of faces F_i and F_j, the set of valid grasp axes can be represented by:

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

subject to the following conditions.

1. $\hat{\mathbf{q}}$ must be in the capture region of $\hat{\mathbf{f}}$, *i.e.* $\hat{\mathbf{q}} \in \mathcal{C}(\hat{\mathbf{f}})$.
2. $\mathbf{g}_i(\hat{\mathbf{q}}, \lambda) \in F_i$, $\mathbf{g}_j(\hat{\mathbf{q}}, \lambda) \in F_j$.
3. $\mathbf{g}_i(\hat{\mathbf{q}}, \lambda)$ and $\mathbf{g}_j(\hat{\mathbf{q}}, \lambda)$ are accessible in the direction $\hat{\mathbf{a}}(\hat{\mathbf{q}})$.
4. The grasp axis lies in the friction cone at the contact points, *i.e.* $|\hat{\mathbf{a}}(\hat{\mathbf{q}}) \cdot \hat{\mathbf{n}}_i| \geq \cos\alpha$ and $|\hat{\mathbf{a}}(\hat{\mathbf{q}}) \cdot \hat{\mathbf{n}}_j| \geq \cos\alpha$.

where the grasp points $\mathbf{g}_i(\hat{\mathbf{q}}, \lambda)$ are computed from (3), and $\hat{\mathbf{a}}(\hat{\mathbf{q}})$ is given by (1) with $\hat{\mathbf{q}}$ replacing $\hat{\mathbf{f}}$. Since $\mathcal{G}_{i,j} \subset S^2 \times \mathbb{R}^+$ is three dimensional, since none of these conditions defines an equality constraint on $S^2 \times \mathbb{R}^+$, and since each of these conditions is independent, the valid set $\mathcal{G}_{i,j}$ will generically be empty or three dimensional. The set of all valid grasp axes is given by: $\mathcal{G} = \cup \mathcal{G}_{i,j}$.

To simplify the presentation in the rest of this paper, only convex polyhedra will be considered. This assumption has the following immediate implications: Every grasp axis satisfying Condition 2 is accessible (*i.e.* it satisfies Condition 3) since every line that does not lie in a face intersects $\mathcal{P}$ at only two points. 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.

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 [30] or Canny's roadmap [31], for characterizing a semi-algebraic set and 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 $\hat{\mathbf{f}}$ is to characterize $\mathcal{G}_{i,j}$ for all distinct pairs of faces i and j.

B. Optimal grasp selection

In the previous section, we observed that the set of valid grasps $\mathcal{G}$ defines a three dimensional semi-algebraic 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 criterion over the set $\mathcal{G}$. There are a number of valid criteria [32]; here we utilize the same one as in Section V. For a distinct pair of faces F_i and F_j, the criterion is:

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

The optimal grasp axis $(\hat{\mathbf{q}}, \lambda) \in \mathcal{G}_{i,j}$ for faces i and j is the one which maximizes $\mathcal{O}_{i,j}$. The optimum grasp can then be selected over all pairs of distinct faces. What does this criterion say? The dot products are a function of the angle between the grasp axis and the normal to the face; the minimum selects the larger angle. When the angle is small, the coefficient of friction μ_{static} does not have to be large for the contact to be stable. Furthermore, the optimal grasp provides an upper bound on the coefficient of friction; that is, μ_{static} 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. 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. Because of the constraint $\lambda > 0$, $\mathcal{G}$ is neither open nor closed. 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 $\mathcal{O}_{i,j}(\hat{\mathbf{q}}, \lambda)$ 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 $S^2 \times \mathbb{R}$ 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 maximum.

Let us first consider maximizing $\mathcal{O}_{i,j}$ over the interior of $\mathcal{G}$. Note that Eq. 5 is non-smooth and multimodal. For the grasp axis direction $\hat{\mathbf{a}}$ to be a local maximum of $\mathcal{O}_{i,j}$, it must satisfy one of the following three conditions:

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 $\hat{\mathbf{a}} \cdot \hat{\mathbf{s}} = 0$.

For Condition i (and similarly Condition ii), Lagrange multipliers can be used to find the grasp axis $\hat{\mathbf{a}}$ subject to these two constraints

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

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:

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

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 IV, 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. Its intersection with the capture region yields a set of arcs that satisfy Condition 1. Furthermore, the grasp points must lie on the two faces, and the intersection of the grasp axis with a face for a fixed $\hat{\mathbf{q}}$ lies on a line parameterized by λ. Since the face is convex by assumption, the grasp points will lie on a single line segment. Consequently, the optimal grasp set 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 are ten different cases to consider.

B.1 Maximizing on the 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). 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 $\partial\mathcal{G}$ corresponds to 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:

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

Since $\hat{\mathbf{q}}$ is unrestricted, the optimal axis direction $\hat{\mathbf{a}}$ can be computed from (6) or (7) as discussed above. 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 $\lambda(\hat{\mathbf{q}})$.

B.2 Maximizing on the 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.

Consider Case 4 where the grasp axis passes through a vertex $\mathbf{v}$ of face i. From (3), we have 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:

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

Since $\hat{\mathbf{a}}$ is a function of $\hat{\mathbf{q}}$ by Eq. 1, the second constraint defines a curve implicitly in $\hat{\mathbf{q}} \in S^2$. 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 $\partial\mathcal{G}$, Conditions (i)–(iii) discussed above are applied to this curve. 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 Equation (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 [33].

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:

$$\begin{aligned}(d_i(\hat{\mathbf{n}}_k \cdot \hat{\mathbf{a}}) - d_k(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}))((\hat{\mathbf{n}}_l \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_j \cdot \hat{\mathbf{a}}) - (\hat{\mathbf{n}}_j \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_l \cdot \hat{\mathbf{a}})) = \\ (d_j(\hat{\mathbf{n}}_l \cdot \hat{\mathbf{a}}) - d_l(\hat{\mathbf{n}}_j \cdot \hat{\mathbf{a}}))((\hat{\mathbf{n}}_k \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}) - (\hat{\mathbf{n}}_i \cdot \hat{\mathbf{q}})(\hat{\mathbf{n}}_k \cdot \hat{\mathbf{a}})).\end{aligned} \tag{10}$$

This is an eighth degree polynomial equation in the elements of $\hat{\mathbf{q}}$, and defines a curve on S^2. 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 Case 6, $\hat{\mathbf{q}}$ is restricted to $\partial\mathcal{C}(\hat{\mathbf{f}})$, and since $\mathcal{O}_{i,j}$ is independent of λ, the optimal axis direction and post pivot configuration $\hat{\mathbf{q}}$ can be computed as in Case 1. Eq. 8 is then used to compute λ.

B.3 The Vertices of $\partial\mathcal{G}$

For a pair of faces F_i, F_j, the vertices of $\partial\mathcal{G}$ can be computed as points $(\hat{\mathbf{q}}, \lambda)$ in the grasp space. Once the coordinates of the vertices are computed, $\mathcal{O}_{i,j}$ 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 the equations in (9). The second equation provides an implicit constraint on $\hat{\mathbf{q}}$. Substituting the parametric 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 equation in (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, Eq. 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 $\partial\mathcal{C}(\hat{\mathbf{f}})$. 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).

VII. Discussion

This paper introduces a new class of grasps. 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 presented an $O(m^2 n \log n)$ 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, under idealized conditions, planning pivots is *solution-complete* [6] *i.e.* a valid pivot grasp always exists when the part's center of gravity lies inside the part and the grasp contacts have an infinite coefficient of friction. Requiring that the center of gravity lies within the part ensures the existence of an accessible grasp; 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 ensure solution-completeness, for example parts that are cuboids (rectangular parallelopipeds) with any non-zero friction. The difference between these two examples is that with infinite friction, we can always achieve the transition in a single pivot grasp while for cuboids with non-zero friction, a sequence of two grasps may be required. Consider for example the problem of pivoting a part onto a face with an anti-parallel normal: 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 [1]). 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 the center of gravity must lie below the grasp axis after pivoting.

In addition to implementing the computation of optimal capturing pivot grasps, we would like to broaden the notion of capture regions. In the current definition, the part settles to a unique final configuration. However, there are larger regions of S^2 from which the part the part will settle to one of a *set* of stable configurations; however, the specific configuration cannot be determined without further assumptions about the dynamics. Pivoting into this region can be represented as non-deterministic arcs in the transition graph. Sensing can then be used to determine which stable configuration has been achieved after pivoting. Using grasps that pivot into these “super-capture regions” in a multi-step plan may increase 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.

We are planning to re-implement the pivot algorithm in a compiled language and integrate it into a CAD system to aid in the rapid setup of flexible part feeders for assembly. This system will also contain routines to predict feeder throughput by estimating the statistical distribution of stable poses for a given part [34]. The long-range objective is to develop tools that can permit coordinated design of products with the assembly systems that will manufacture them.

Appendix

Can a sequence of two rotations, one about some axis lying in the horizontal plane and the second about the vertical axis, cover the space of all 3D rotations? Recall that a rotation about an axis ω by angle θ can represented by a unit quaternion $\mathbf{Q} = (q_0, \mathbf{q}) = (q_0, (q_1, q_2, q_3)^T) = (\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\omega)$ [35]. Furthermore, recall that the product of two quaternions $\mathbf{Q}$ and $\mathbf{P}$ is given by:

$$\mathbf{Q} * \mathbf{P} = (q_0 p_0 - \mathbf{q}\cdot\mathbf{p}, q_0\mathbf{p} + p_0\mathbf{q} + \mathbf{q}\times\mathbf{p})$$

Using quaternions, we will prove the following:

Proposition 1: Any rotation can be achieved by first rotating about an axis restricted to a horizontal plane by some angle ϕ followed by a rotation of angle α about the vertical axis.

Proof: We show that an arbitrary unit quaternion $\mathbf{Q}$, it is equal to the product of two quaternions: rotation about the vertical axis $\mathbf{V}$ and rotation about an axis in the horizontal plane $\mathbf{H}$. As quaternions, they can be expressed as:

$$\begin{aligned}\mathbf{V} &= (\cos\tfrac{\alpha}{2}, (0, 0, \sin\tfrac{\alpha}{2})^T)\\ \mathbf{H} &= (\cos\tfrac{\phi}{2}, (k_x\sin\tfrac{\phi}{2}, k_y\sin\tfrac{\phi}{2}, 0)^T)\end{aligned}$$

where $(k_x, k_y, 0)^T$ is the horizontal axis of rotation.

We now show that for any quaternion $\mathbf{Q}$, there exist values of α, ϕ, k_x, k_y such that $\mathbf{Q} = \mathbf{V} * \mathbf{H}$. After taking the quaternion product, we arrive at the following four scalar equations:

$$\begin{cases} q_0 = \cos\frac{\alpha}{2}\cos\frac{\phi}{2}\\ q_1 = \cos\frac{\alpha}{2}\sin\frac{\phi}{2}k_x - \sin\frac{\alpha}{2}\sin\frac{\phi}{2}k_y\\ q_2 = \cos\frac{\alpha}{2}\sin\frac{\phi}{2}k_y + \sin\frac{\alpha}{2}\sin\frac{\phi}{2}k_x\\ q_3 = \sin\frac{\alpha}{2}\cos\frac{\phi}{2}\end{cases}$$

Taking the sum of the square as well as the ratio of the fourth and first equations provides the means to solve for ϕ and α

$$\begin{aligned}\cos^2\tfrac{\phi}{2} &= q_0^2 + q_3^2\\ \tan\tfrac{\alpha}{2} &= q_3/q_0\end{aligned}$$

There are multiple valid solutions to these equations unless $q_0 = 0$; however, in this case it is easy to see from the original equations that $\cos\frac{\alpha}{2} = 0$ (i.e. $\alpha = \pi$), and so $\cos\frac{\phi}{2} = q_3$. Once α and ϕ are determined, the equations for q_1 and q_2 are linear in k_x and k_y and can be easily solved. ■

Acknowledgments

We thank Brian Carlisle, John Craig, Frank Park, and Jeff Trinkle for helpful feedback, Otfried Schwarzkopf for discussions on computing envelopes, Geert-Jan Giezeman for help with the implementation, and the anonymous reviewers for their suggestions.

References

[1] B. Carlisle, K. Goldberg, A. Rao, and J. Wiegley, “A pivoting gripper for feeding industrial parts”, in *IEEE Int. Conf. Robotics and Automation*, May 1994.

[2] P. Tournassoud, T. Lozano-Perez, and E. Mazer, “Regrasping”, in *IEEE Int. Conf. Robotics and Automation*, May 1987.

[3] D. E. Koditschek, “Robot assembly: Another source of nonholonomic control problems”, in *American Control Conference*, 1991, pp. 1627–1632.

[4] D. E. Koditschek, “An approach to autonomous robot assembly”, *Robotica*, 1993.

[5] D. J. Kriegman, "Computing stable poses of piecewise smooth objects", *CVGIP: Image Understanding*, vol. 55, no. 2, pp. 109–118, Mar. 1992.

[6] K. Y. Goldberg, "Completeness in robot motion planning", in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, San Fransisco, Feb. 1994.

[7] A. Rao, D. Kriegman, and K. Goldberg, "Complete algorithms for reorienting polyhedral parts using a pivoting gripper", in *IEEE Int. Conf. Robotics and Automation*, Nagoya, Japan, May 1995, pp. 2242–2248.

[8] R. A. Grupen, T. C. Henderson, and I. D. McCammon, "A survey of general purpose manipulation", *Int. J. Robotics Research*, vol. 8(1), 1989.

[9] J. Pertin-Troccaz, "Grasping: A state of the art", in *The Robotics Review I*, pp. 71–98. MIT Press, 1989, edited by O. Khatib, J. J. Craig, and T. Lozano-Perez.

[10] J. Kenneth 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.

[11] B. Faverjon and J. Ponce, "On computing two-finger force-closure grasps of curved 2D objects", in *IEEE Int. Conf. Robotics and Automation*, May 1991.

[12] A. Blake, "Computational modelling of hand-eye coordination", *Phil. Trans., Royal Society of London B*, vol. 337, pp. 353–360, 1992.

[13] X. Markenscoff, L. Ni, and C. H. Papadimitriou, "The geometry of grasping", *IJRR*, vol. 9(1), February 1990.

[14] R.S. Fearing, "Simplified grasping and manipulation with dextrous robot hands", *IEEE J. Robotics and Automation*, vol. 2, no. 4, pp. 188–195, 1986.

[15] J. Hong, G. Lafferriere, B. Mishra, and X. Tan, "Fine manipulation with multifinger hands", in *IEEE Int. Conf. Robotics and Automation*, May 1990.

[16] D. Rus, "Dextrous rotations of polyhedra", in *IEEE Int. Conf. on Robotics and Automation*, May 1992.

[17] D. L. Brock, "Enhancing the dexterity of a robot hand using controlled slip", Tech. Rep. AI-TR 992, MIT, May 1987.

[18] J. C. Trinkle and R. P. Paul, "Planning for dextrous manipulation with sliding contacts", *Int. J. Robot. Res.*, June 1990.

[19] M. A. Erdmann, M. T. Mason, and G. Vaněček, "Mechanical parts orienting: The case of a polyhedron on a table", *Algorithmica*, vol. 10, no. 2, August 1993.

[20] S. Akella and M. T. Mason, "Posing polygonal objects in the plane by pushing", in *IEEE Conf. Robotics and Automation*, May 1992, pp. 2255–2262.

[21] K. Lynch, "The mechanics of fine manipulation by pushing", in *IEEE Int. Conf. Robotics and Automation*, May 1992.

[22] 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.

[23] B. Chazelle, "Triangulating a simple polygon in linear time", *Discrete and Computational Geometry*, vol. 6, pp. 485–524, 1991.

[24] J. Hershberger, "Finding the upper envelope of n line segments in $O(n \log n)$ time", *Inform. Process. Lett.*, vol. 33, pp. 169–174, 1989.

[25] G.-J. Giezeman, *SpaGeo—A Library for Spatial Geometry*, Dept. Comput. Sci., Utrecht Univ., Utrecht, the Netherlands, Jan. 1994.

[26] R. C. Brost, "Dynamic analysis of planar manipulation tasks", in *IEEE Conf. on Robotics and Automation*, 1992.

[27] R. C. Brost, *Analysis of Planning of Planar Manipulation Tasks*, PhD thesis, Carnegie Mellon, 1991.

[28] D. Kriegman, "Let them fall where they may: Computing capture regions of curved 3D objects", in *IEEE Conf. on Robotics and Automation*, May 1994, pp. 595–601.

[29] D. Kriegman, "Let them fall where they may: Capture regions of curved objects and polyhedra", Center for Systems Science 9508, Yale University, Electrical Engineering, New Haven, CT 06520-8267, 1995, To appear in Int. J. Robotics Research.

[30] D. Arnon, G. Collins, and S. McCallum, "Cylindrical algebraic decomposition I and II", *SIAM J. Comput.*, vol. 13, no. 4, pp. 865–889, November 1984.

[31] J.F. Canny, *The Complexity of Robot Motion Planning*, MIT Press, 1988.

[32] I. M. Chen and J. W. Burdick, "Finding antipodal grasps on irregularly shaped objects", *IEEE Trans. on Robotics and Automation*, pp. 507–511, Aug. 1993, Also in Proc. *Intl. Conf. Robotics & Automation*, 1992.

[33] A. Morgan, *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems*, Prentice Hall, Englewood Cliffs, 1987.

[34] K. Goldberg, J. Craig, R. Zanutta, and B. Carlisle, "Estimating throughput for a flexible part feeder", in *International Symposium on Experimental Robotics*, 1995.

[35] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.

Anil Rao completed his B.Tech. degree in Computer Science from the Indian Institute of Technology, Madras, in 1988, his M.S. in Computer and Information Sciences from the Ohio State University, and his Ph.D in Computer Engineering from the University of Southern California in 1992. Between March 1993 and August 1995, he was with the Department of Computer Science at Utrecht University in the Netherlands as a research associate. His research interests lie in Robotic Manipulation and Planning, Geometry and Algorithms, and Motion Planning. Since September 1995, he is employed by Qualcomm, Inc. in San Diego.

David J. Kriegman was born in Englewood, New Jersey on October 24, 1961. He graduated *summa cum laude* from Princeton University with a B.S.E. degree in Electrical Engineering and Computer Science in 1983 where he was awarded the Charles Ira Young Award for electrical engineering research. He received the M.S. degree in 1984 and Ph.D. in 1989 in electrical engineering from Stanford University where he studied under a Hertz Foundation Fellowship.

Currently, he is an Associate Professor at the Center for Systems Science in the Departments of Electrical Engineering and Computer Science at Yale University and was awarded a National Science Foundation Young Investigator Award in 1992.

Kenneth Y. Goldberg earned a dual degree (B.S. in Electrical Engineering and B.S. in Management) from the University of Pennsylvania in 1984. He got his PhD in 1990 from the School of Computer Science at Carnegie Mellon University. He also studied at Edinburgh University and the Technion. In 1991, he joined the faculty at USC with a joint appointment in Computer Science and EE-Systems. In 1995, he moved to UC Berkeley's Department of Industrial Engineering and Operations Research, where he directs research on geometric algorithms for feeding, sorting, and fixturing industrial parts. Prof. Goldberg serves on the Advisory Board of the IEEE Society of Robotics and Automation and was named an NSF Presidential Faculty Fellow in 1995.