Computing Grasp Functions

A.S. Rao and K.Y. Goldberg

UU-CS-1994-18
April 1994

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

Computing Grasp Functions

A.S. Rao and K.Y. Goldberg

Technical Report UU-CS-1994-18
April 1994

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924-3275

Computing Grasp Functions*

Anil S. Rao† Kenneth Y. Goldberg‡

Abstract

The *grasp function* of a planar part, characterized by extrema in the part's width function, can be used to analyze grasp mechanics. In particular, it can predict the final orientation of the part when it is grasped with a parallel-jaw gripper. This information allows us to derive a sequence of grasp angles that will orient the part up to symmetry in the grasp function. In previous papers the grasp function was assumed given as input; in this paper we present a linear-time algorithm for computing the grasp function of a part bounded by n algebraic arcs given in parametric form. We also show that the algorithm can be extended to compute a related function that describes the outcome of first pushing and then grasping the part.

1 Introduction

In manufacturing, it is often necessary to orient parts prior to packing or assembly. In previous work, we showed that a modified parallel-jaw gripper can be used to orient planar parts bounded by linear [10] and algebraic curves [32]. See Fig.s 1 and 4. This approach uses mechanical compliance of the part as it is grasped; part rotation can be precisely characterized with a function, $\Gamma : S^1 \times S^1$, that we call the *grasp function*. Given an initial orientation θ of the part with respect to the gripper, $\Gamma(\theta)$ gives the part's final orientation after the jaws are fully closed (See bottom of Fig. 2).

In convex geometry is defined the concept of *width in a given direction* of a planar curve (extendible to solids and higher dimensions) as follows: "the width of a convex curve in a given direction is the distance between a pair of supporting lines perpendicular to this direction" [37]. As will be shown in Section 2, the grasp function can be characterized by extrema in this width function. Computing the width function, even for algebraic parts, precisely can be difficult. Thus we are motivated to seek direct methods of obtaining the grasp function.

This paper presents an algorithm that computes the grasp function of a part given the n arcs forming its boundary. The arcs are assumed to be algebraic: each arc is a segment from a smooth (twice-differentiable) curve represented by a polynomial or rational equation(s) of degree at most some constant k in parametric form. We call these *algebraic parts*. Such contours include a broad class of industrial curves, (and surfaces) [17]. The algorithm runs in $O(n)$ time, given the arcs in order, and is therefore optimal. The constant of proportionality is a unit of symbolic computation time defined in Section 2.2. When combined with the planning algorithm given in [32], the results in this paper provide a complete algorithm for planning sensorless strategies for orienting algebraic parts. Examples of such plans are presented in Section 3.3.

We next describe related work; Section 2 defines diameter and grasp functions and the description of the latter in terms of the former. Section 3 presents the grasp function computing algorithm.

1.1 Related Work

Computational Geometry In computational geometry, the diameter of a set is the farthest distance between any two points taken from the set; by width of a set they imply the distance between the closest pair of supporting parallel lines. These are, respectively, the (global) maximum and minimum values in the set's diameter function. For point sets in

*This research was supported by ESPRIT Basic Research Action No. 6546 (project PRoMotion), NSF Awards IRI-9123747 and DDM-9215362 (Strategic Manufacturing Initiative) and by an equipment grant from Adept Technology, Inc. The results in this paper have been submitted to the journal "Computational Geometry: Theory and Applications."

†Department of Computer Science, Utrecht University, Padualaan 14, Postbus 80.089, 3508 TB Utrecht, Netherlands. anil@cs.ruu.nl, Tel: 31-30-535093, FAX: 31-30-513791.

‡Institute of Robotics and Intelligent Systems and Department of Computer Science, PHE 204, University of Southern California, Los Angeles, California 90089-0273. goldberg@iris.usc.edu, Tel: 1-213-740-9080, FAX: 1-213-740-7877.

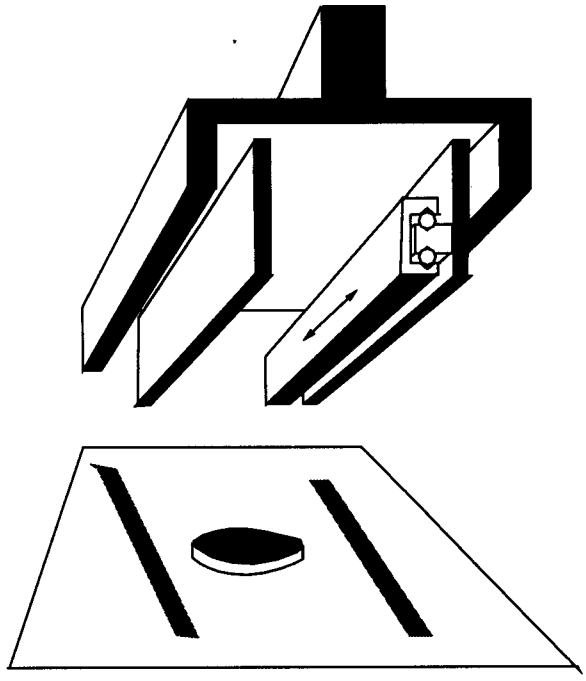


Figure 1: Schematic of parallel jaw gripper above a part.

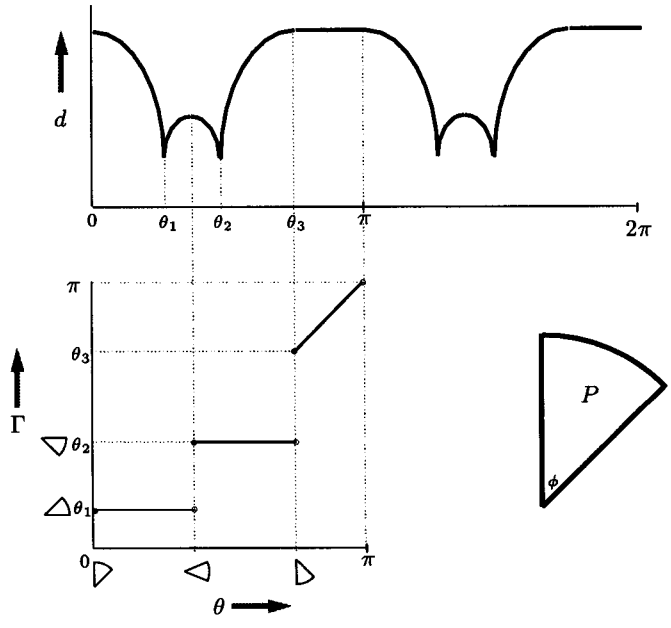


Figure 2: **(top)** The diameter function for the pie-shaped part (shown at right). During a squeeze action, the part rotates so as to reduce the diameter, terminating when the diameter reaches a local minimum. **(bottom)** The grasp function is shown for the portion $[0, \pi)$.

the plane, [1, 19, 34] give algorithms to compute and maintain the diameter and width under insertion and deletion of points. For point sets in three dimensions, efficient algorithms are presented in [18, 7] for computing their diameter and width. Partitioning point sets while maintaining some conditions on the resulting diameters is considered in [2, 15]. Diameter in the Manhattan metric is considered in [9].

Approximating point sets by n-point subsets that have the same width is considered in [12].

Lovász and Simonovits [21] consider randomized algorithms to compute the volume and diameter of a convex set, given by a well-guaranteed separation oracle, in d dimensions. While they give polynomial time algorithms for approximating the volume, they show that there cannot exist a polynomial time algorithm to approximate the diameter or width within a factor of $d^{1/4}$.

Thus we see that computing and maintaining the diameters and widths of sets is a well-studied and difficult problem. Now we refer to literature dealing with the recovery of a planar object's shape from its diameter function. The motivation for this is that measuring the diameter along a particular orientation can be accomplished with simple hardware. The existence and computation of "curves of constant width" (examples: circle or the Reuleaux triangle) has been studied since Euler who called them *gleichdicke*. They also referred to as orbiforms or spheroids. Generating pixel representations of such curves in considered in [16]. Linear time ruler and compass constructions for orbiforms with width equal to the diameter of an input polygon are presented in [5]. Thus recovery of shape from diameter function of orbiforms, and hence curved objects in general, is impossible. Therefore, in [31, 33] we considered the recovery of shape from a polygon's diameter function (which is piecewise sinusoidal) and showed that there existed infinitely many polygons having the same diameter function. The diameter along some orientation can be considered as the length of the object's orthogonal projection on a line perpendicular to this direction. If, however, we design a probe that returns not only the length of the projection but the end-points as well, then [20] shows that from $3n - 2$ projection probings, you can recover the complete shape of the polygon.

These results show that recovering shape from diameter is a difficult problem as well; many objects have the same diameter function. However, our results in [10, 32] then imply that all parts having the same diameter function (and hence the same grasp function) have the same orienting plan. In this paper, our focus is on computing the grasp function from extrema (local maxima and local minima) in the diameter function, the first step towards orienting parts by a sequence of grasps.

Computational Algebra The examples we present in Section 3.3 required solutions to high order polynomial simultaneous equations in two variables. These were implemented using the Computer Algebra package *Maple V* [6, 14].

Robotics and Grasping There is a substantial literature on the subject of grasping. See [13] and [27] for reviews. Chen and Burdick [8] search for antipodal grasp points on objects using an analysis similar to the one we use in Section 3.1. However, their interest lies in constructing force-closure regions to immobilize the object. On the other hand, our goal is to orient parts: our grasps permit (and in fact exploit) compliant motion of the object.

Pingle *et al.* [30] demonstrated that *pushing* can be used to eliminate uncertainty while grasping. Mason [23] formalized the role of pushing in robot manipulation. Building on this, Brost [4] distinguished between stable and wedged configurations for polygonal parts and gave an algorithm for achieving a stable grasp with a parallel-jaw gripper when the part's initial orientation can be described by a tolerance interval.

The idea of using a sequence of pushing or grasping motions to reduce uncertainty in part orientation was previously addressed by [22, 25, 29, 36]. Natarajan [26] ignored the mechanics of parts feeders and focused on the computational problem of planning with a given set of transfer functions.

2 Preliminaries: Mechanics and Geometry

In this section we give the definitions of diameter and grasp functions; the computation of the latter from extrema in the former; and the assumptions involved throughout. ¶ refers to the part and P to its planar projection. Let S^1 denote the space of planar orientations and $\Re_+$ the set of positive reals. An *interval* is a continuous subset of S^1.

2.1 Diameter Function

Consider two parallel lines making an orientation θ wrt the part, minimally separated and enclosing the part. The distance between the lines is the diameter of the part in orientation θ, denoted as $d(\theta)$. Thus, the *diameter function* d is a $S^1 \rightarrow \Re_+$ function.[1] See Fig.s 3, 2.

Some easily verifiable properties of the diameter function include:

- The diameter function is continuous: $\Delta d \rightarrow 0$ as $\Delta\theta \rightarrow 0$. It is single valued and positive with domain S^1. [37]
- The diameter function of a part is equal to the diameter function for its convex hull.
- The diameter function has period π because the parallel lines are interchangeable. Additional symmetry in the part can reduce the periodicity in diameter function to a fraction of π [10].
- Let p denote the perimeter of the projection P. Then

$$\int_0^\pi d(\theta) \;=\; p.$$

  This follows from Cauchy's formula restricted to 2D (see Problem 22.14 on page 173 in [3]).
- The diameter function for a polygonal part is piecewise sinusoidal [32, 33].

2.2 Assumptions

Part Assumptions The part ¶ is rigid and of known fixed cross section P which is made up of a sequence of algebraic arcs. If P is not convex, let P refer to its convex hull which may be computed in $O(n)$ time using the algorithm in [35] (given the arcs in order).

Let P consist of n arcs; we assume that the clockwise ordering $S_0, \ldots, S_{n-1}$ of the n arcs is given. Let ± 1 be modulo n. Define n vertices $V_1, \ldots, V_i$ so that vertex V_i is the point where S_i and S_{i+1} meet.

Each arc is a segment taken from an algebraic curve. The underlying curve may be implicitly represented as a polynomial equation $f(x, y) = 0$ or parametrically as functions $X(t), Y(t)$ where X, Y are functions in t giving the x, y coordinate of a point on the curve. These are typically rational functions (ratio of two polynomials) although that is not a requirement for our purposes. For simplicity, let us assume the parametric representation. That is, for each arc S, functions $X_S(t), Y_S(t)$, are given. Most algebraic curves used in engineering (including conics) have (rational) parametric representations. However, not all algebraic curves do. Our techniques work even if the curve is known only implicitly and arc end-points are known.

When the underlying curve is parametrically represented, let the parameter t vary between 0 and 1 to sweep the arc; points $\text{left}(S) = (X_S(0), Y_S(0))$ and $\text{right}(S)(X_S(1), Y_S(1))$ are therefore the end-points of the arc. A simple remapping of the parameter can ensure this. The direction of parameter sweep is clockwise: *i.e.* $\text{left}(S_{i+1}) = \text{right}(S_i)$.

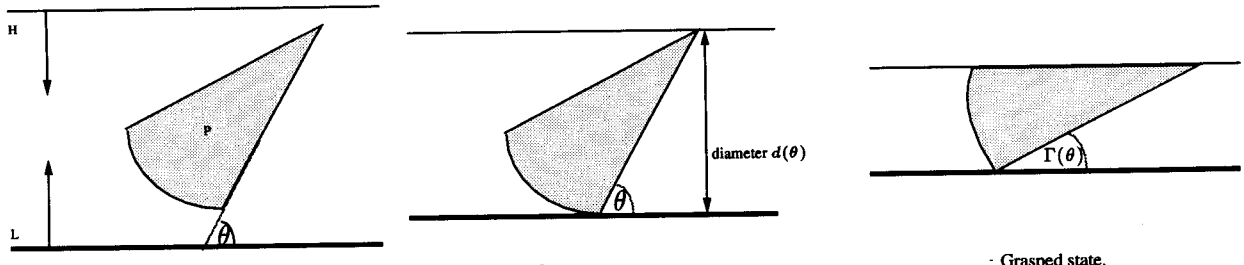
We assume, without loss of generality, that every arc is "small" in that it does not contain two points with parallel tangents. If an arc does not have this property, it can be split into two arcs with the addition of a (pseudo) vertex so that it would.

Finally, let us denote by τ a unit of symbolic computation time. More specifically, if the X_S, Y_S are ratios of two polynomials of degree at most k in t, then τ_k can be upper bounded by the time required to solve two simultaneous polynomial equations of degree at most $(2k-1)k^2$. The simultaneous equations have degree at most $2k-1$ if the X_S, Y_S are polynomials of degree at most k (rather than rationals). We treat k as some known constant and drop subscript k from τ. The rest of the paper assumes $\tau = O(1)$.

Grasp Assumptions Define a *grasp action* α as the combination of orienting the gripper at angle α wrt a fixed world frame, closing the jaws as far as possible over the part without deforming it, and then opening the jaws.

We assume a grasping process as in [10, 32, 33]. Briefly, the gripper is based on the ubiquitous parallel-jaw gripper modified to reduce friction to negligible levels [11]. The part is always flat on a horizontal table between the jaws of the gripper. The gripper has two degrees of freedom: in addition to the opening/closing motion of the jaws, they can

[1] Thus the diameter function as defined here and the width function w defined in [37] are identical except for a shift of 90 degrees, *i.e.* $w(\theta) = d(\theta + \pi/2)$.



Non-contact state.
L, H wide apart so as not to be touching P

Orientation θ

Contact state.
L, H simultaneously just touching P

Orientation θ

Grasped state.
L, H closed as far as possible without deforming P

Orientation $\Gamma(\theta)$

Figure 3: Configurations of the part P wrt the gripper G during the grasping process. In the non-contact state **(top)**, the jaws L,H are not touching P. In the contact state **(middle)**, both jaws just touch P but do not effect its orientation θ. Between the contact and grasped state **(bottom)**, the orientation of P changes to $\Gamma(\theta)$, Γ being the part's grasp function.

also together be oriented arbitrarily with respect to the table. A (squeeze) grasp action begins with the jaws widely separated and oriented at some angle with the table. See Fig. 3. Then they move towards each other and make opposing and simultaneous contact (pure squeezing) with the part at points at which the jaws are clearly tangent to the part. As the jaws continue closing motion, the part exhibits mechanical compliance changing its orientation. The part never wedges because of negligible friction between part and gripper. Gripper motion stops when further motion will violate part rigidity. The jaws then move apart. Gripper and part motion occurs in the plane and is slow enough that inertial forces are negligible (quasi-static motion as discussed in [24, 28]).

The assumption of simultaneous contact, almost never achieved in practice (Esp without sensors!) is made only for simplicity of presentation. In practice, push-grasp actions [4, 32] would be used. This is briefly discussed in the appendix.

2.3 Grasp Function

Under the assumptions described in Section 2.2, the chief among them being that of negligible friction and simultaneous contact between the jaws and gripper, it follows (this is shown in [10, 32]) that

Fact 1 *(Squeeze) Grasping minimizes diameter.*

In other words, the diameter function plays the role of a potential function when grasped at an orientation θ, the part descends to a valley (local minima orientation) in the diameter function. Let this final orientation be $\Gamma(\theta)$. This mapping from initial pre-grasp orientations to final orientations $\Gamma : S^1 \rightarrow S^1$ is the *grasp function* of the part.

Consider an interval $\theta \in (a, b)$ in which the diameter $d(\theta)$ is constant. Then from Fact 1 we clearly get that $\Gamma(\theta) = \theta$. A maximal interval in which $\Gamma(\theta) = \theta)$ (diameter is constant) is called a *ramp*. Otherwise, let a, b be adjacent local maxima in d with a single local minimum α between them. Then $\forall\theta \in (a, b)\ \Gamma(\theta) = \alpha$. Such an interval forms a *step* in Γ.

Fact 1 is ambiguous about $\Gamma(\theta)$ where θ is an end-point of a step or ramp (local maximum or end-point of an interval of constant diameter). However, we may assume steps and ramps to be left-closed and right-open. This is justified by a simple modification to the grasping process as shown in [32]. Thus a *grasp function consists of a sequence of (left-closed and right-open) steps and ramps.* The periodicity of the grasp function is the same as that of the diameter function. See again Fig. 2. $[0, \theta_3)$ contains two steps while $[\theta_3, \pi)$ is a ramp.

In [32] we prove that:

> *The diameter function of an algebraic part is differentiable at all but a finite number of points in S^1 and it contains a finite number of local maxima, local minima, and regions in S^1 of constant diameter.*

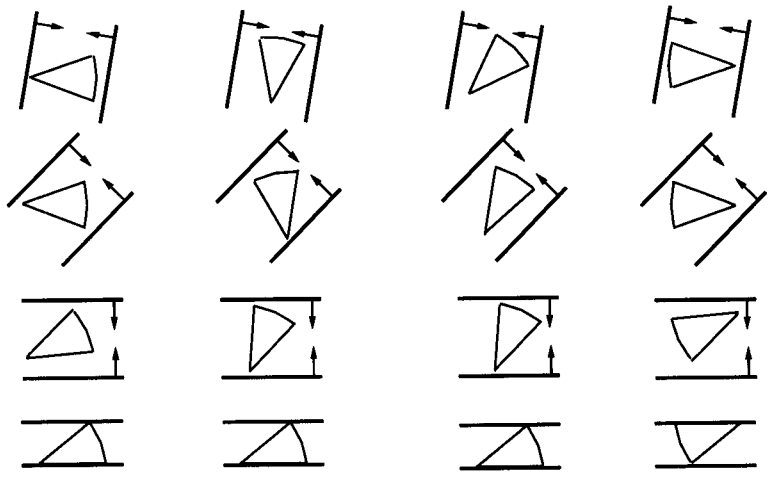


Figure 4: Top view of a three-stage plan for orienting a pie-shaped part with half angle 20° into one of two orientations. Four traces of the plan (running from top to bottom) are shown. Note that the part's initial orientation is different in each trace (top of each column). The gripper orientation at each stage, indicated by two parallel-lines, is the same for each trace; yet the same desired grasp configuration is achieved in all cases. To obtain a single final orientation, push-grasp actions may be used. See Fig. 13.

This readily implies the following.

Lemma 1 *The grasp function of an algebraic contains a finite number of steps and ramps.*

Since the diameter function is oblivious to part concavities (Section 2.1), it follows that the grasp function is too. Hence, if the true planar part shape P is not convex, we let P refer to its convex hull.

Finally, notice that the grasp function can be computed from *local maxima, local minima* in the diameter function and *end-points of regions of constant diameter*. Let us refer to these orientations as *extrema* in the diameter function.

2.4 Orienting parts

The importance of the grasp function lies in the planning algorithms presented in [10] for polygonal parts and [32] for curved parts. These algorithms, when presented with the grasp function, plan the optimal length sequence of grasp actions to orient the part up to symmetry in its diameter function starting from any arbitrary orientation. Given n steps and ramps as input, the algorithm runs in $O(\min(nN, n^2 \log n + N))$ time, where N is the length of the plan produced. Although the plan length N is optimal, there are no bounds on N in terms of n. We skip the details in this paper which is devoted to computing grasp functions. As an example, when presented with the grasp function at the bottom of Fig. 2 as input, the planning algorithm comes up with the angles $75°, 42°, 0$ which are the grasp actions shown in Fig. 4.

3 Grasp Function Computing Algorithm

In Section 2.3 we noted that to compute the grasp function Γ, we need to compute the extrema in the diameter function d. This section presents an $O(n)$ algorithm to compute the extrema in the diameter function for an algebraic part.

Any contact configuration between the jaws and the part is defined by the two (opposing) points of contact. Let us refer to such pairs of points as *contact points*. Let us refer to an arc S_i or a vertex V_j as a *feature*. The features that contain contact points are called *contact features*. Further, if the contact points define an extremum configuration, let the features that contain them be called *extremum contact features*.

In Section 3.1 we consider testing if a pair of features are extremum contact features, and if so, to obtain the extremum configuration in $O(1)$ time. This leads to a naive $O(n^2)$ algorithm by considering every feature pair. However, Section 3.2 shows that only $O(n)$ contact feature pairs exist and gives an algorithm to compute them. The

tests of Section 3.1 need be therefore applied only $O(n)$ times to obtain all the extrema in the diameter function, resulting in an $O(n)$ time algorithm.

Since there are only $O(n)$ contact configurations, there are clearly only $O(n)$ extrema. Therefore in an additional $O(n)$ time, the grasp function may be computed. This results in our main theorem:

Theorem 1 *The grasp function of an algebraic part with n arcs can be computed in $O(n)$ time.*

3.1 Computing Extrema Given Feature Pairs

There are three types of feature pairs: arc-arc (a-a), arc-vertex (a-v), and vertex-vertex (v-v) in decreasing order of complexity. A v-v feature pair obviously defines a unique contact configuration which is treated as potentially extremal. Pairs of the a-a and a-v type may result in uncountably many contact configurations. Algebraic tests are applied to these pairs to prune out a finite number of (at most $(2k-1)k^2$) potentially extremal configurations. A geometric grasp test is applied to each potentially extremal configuration as the final test for extremality.

Algebraic Conditions First we consider a-a pairs and then a-v pairs.
arc-arc pairs. Let $S(u), T(v)$ be the two arcs under consideration.

Consider the bivariate function $D^2(u,v)$ to be the square of the Euclidean distance between point $S(u)$ and point $T(v)$. That is,

$$D^2(u,v) = (X_S(u) - X_T(v))^2 + (Y_S(u) - Y_T(v))^2.$$

Note that

$$\frac{\partial(D^2)}{\partial u} = 2\left(X_S(u) - X_T(v)\right)\frac{dX_S}{du} + 2\left(Y_S(u) - Y_T(v)\right)\frac{dY_S}{du}$$

is well defined for $0 < u, v < 1$ (since S is smooth). Similarly $\frac{\partial(D^2)}{\partial v}$ is well defined in this region.

The key observation is that if the pair of points $S(u_0), T(v_0)$ is a extremum contact configuration, then (u_0, v_0) will be a local extremum in $D^2(u,v)$. To see this, consider an ϵ-neighborhood around the point (u_0, v_0), ϵ sufficiently small: all points in this neighborhood will result in a D^2 value less (more) than $D^2(u_0, v_0)$ if θ_0 is a local maximum (minimum). See Fig. 5.

In other words, u_0, v_0 is a solution of the system of two simultaneous equations in the range $(0,1)$:

$$\frac{\partial(D^2)}{\partial u} = 0; \quad \frac{\partial(D^2)}{\partial v} = 0. \tag{1}$$

Remark: If the underlying curve equations for S, T are known only implicitly as $f_S(x,y) = 0, f_T(x,y) = 0$, we get four simultaneous equations rather than two; the two extra equations are for expressing point-in-curve segment inclusion. The above analysis still holds and a configuration defined by the pair of points (x_0, y_0), (x_1, y_1) is extremal only if $f_S(x_0, y_0) = 0; f_T(x_1, y_1) = 0; \frac{df_S}{dx}(x_1 - x_0) + \frac{df_S}{dy}(y_1 - y_0) = 0; \frac{df_T}{dx}(x_1 - x_0) + \frac{df_T}{dy}(y_1 - y_0) = 0$. The solution obtained must then be tested for arc-inclusion. In the rest of this section we assume parametric representations with the understanding that we can handle implicit representations as well (albeit resulting in equations somewhat more complicated).

arc-vertex pairs. Let $U(u), W$ be an arc-vertex pair. Let the coordinates of W be (X_W, Y_W). The analysis is similar to that before and so is the result: if the pair of points $U(u_0), W$ define an extremal contact configuration, then the normal to the arc at $U(u_0)$ passes through W; *i.e.* $0 < u_0 < 1$ should satisfy (See Fig. 6):

$$\frac{dX_U}{du}(X_U(u_0) - X_W) + \frac{dY_U}{du}(Y_U(u_0) - Y_W) = 0. \tag{2}$$

Remark: Intervals of constant diameter orientations (ramps in the grasp functions) can arise either as arc-arc or arc-vertex contact configurations. In the former case, the Equations in 1 will not be independent over $u, v \in (0,1)$ and the region over which they are not independent can be computed giving the range of orientations of constant diameter. In the a-v case, Equation 2 reduces to $0 = 0$. Again, the range of contact configurations over which Equation 2 degenerates corresponds to an interval of constant diameter orientations.

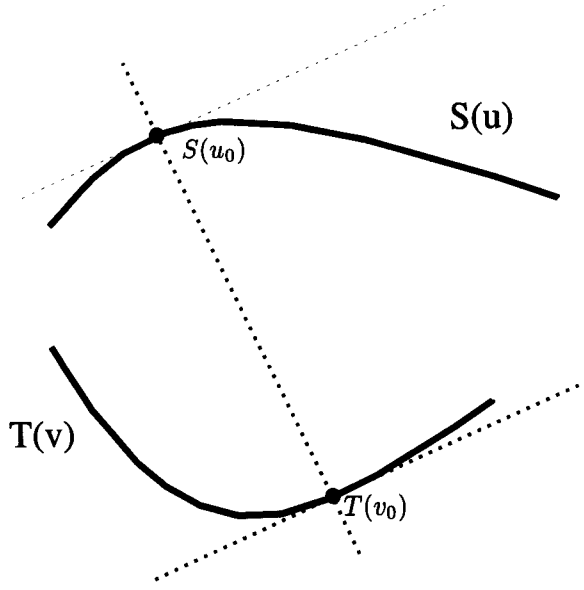


Figure 5: Pairs of curve segments $S(u), T(v)$. If $S(u_0), T(v_0)$ define an extremum orientation, then they satisfy the system of equations 1 (but not vice versa). This implies that the following three lines coincide: the normal to $S(u)$ at u_0; the normal to $T(v)$ at v_0; the line joining u_0 and v_0.

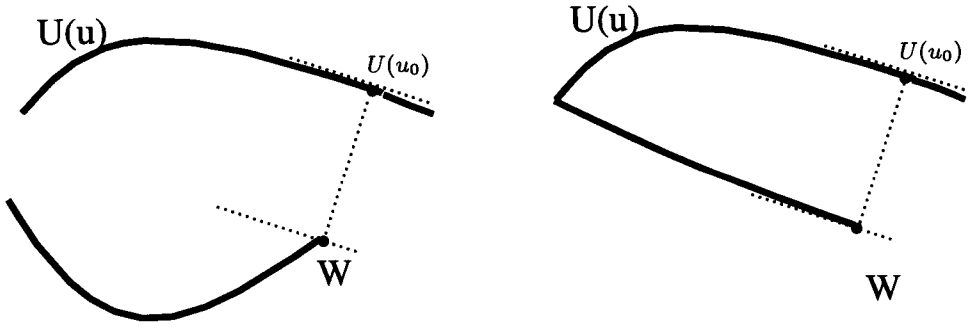


Figure 6: Potentially extremal arc-vertex pairs: U, W. The configuration $U(u_0), W$ is potentially extremal if the normal to U at u_0 passes through W. However, the configuration on the left fails grasp conditions and is therefore not extremal.

Grasp Conditions. Equations 1 and 2 give configurations that are potentially extremal. Also, all v-v pairs are potentially extremal. All potentially extremal configurations $S(u_0), T(v_0)$ have to satisfy the following jaw-contact condition to be extremal: that is, the jaws need to be able touch points $S(u_0), T(v_0)$. This is so if and only if the two lines at $S(u_0), T(v_0)$ perpendicular to the line joining the two points intersect no other point on the part boundary (only intersections with the same or adjacent features need be checked due to convexity and therefore this can be implemented in constant time). See Fig.s 6, 9.

Lemma 2 *Given two features, an extremal contact configuration, if it exists, can be computed in $O(1)$ time.*

3.2 Only $O(n)$ Feature Pairs Need Be Considered

In this section we will show that for an algebraic (convex) part boundary P, only $O(n)$ contact feature pairs exist and they can all be computed in $O(n)$ time. For a feature (arc or vertex) S, let $\beta(S)$ denote the list of (clockwise ordered) features that form contact feature pairs with S. Thus, we need to show that $\sum |\beta(S)| = O(n)$ and desire an algorithm that will compute all the $\beta(S)$ in $O(n)$ time.

Before we describe the algorithm, we need some notation and a lemma. A tangent to P at a vertex is simply a line that touches P at only the vertex and is generally not unique (unless the arcs intersecting at the vertex have parallel tangents there). Let $\lambda(S)$ denote the set of slopes of the lines tangent to (points in) S. Since P is convex, clearly $\lambda(S)$ is a single range of angles and the $\lambda(S)$ taken together are a partition of the space of planar angles $[0, 2\pi)$. Every $\lambda(S)$ can be made non-empty and distinct from the others by assuming that arcs are open at both ends.

Let p be some reference point chosen to the interior of P, and let the angle subtended by an arc S_i at p be σ_i. Let a list of features β be termed *consecutive* if it is a sublist of the circular list $[v_0, S_0, v_1, S_1, \ldots, v_{n-1}, S_{n-1}]$. Thus, if β is a consecutive list of features, then $\cup\sigma_i$, where i varies over the arc indices in β, is a single range of angles. Let us denote this range of angles as $\sigma(\beta)$. In a continuous range of angles, let the angles be ordered in increasing clockwise fashion: *i.e.* the *minimum* angle is the most CCW angle and the maximum angle is the most CW angle.

Given m consecutive feature lists $\beta_1, \beta_2, \ldots, \beta_m$, they are *well-ordered* if the list of minimum angles from $\sigma(\beta_1), \sigma(\beta_2), \ldots, \sigma(\beta_m)$ are in clockwise order; and the same holds for the list of maximum angles.

Lemma 3 *Let $[S, T, U, \ldots, W]$ be an arbitrary list of consecutive features from P.*

1. *$S \notin \beta(S)$.*
2. *$\beta(S)$ is consecutive.*
3. *$\beta(S), \beta(T)$ are well-ordered. Also, $\beta(S), \beta(T)$ have at most one feature in common.*
4. *$\beta(S), \beta(T), \beta(U)$ are well-ordered.*
5. *$\beta(S), \beta(T), \beta(U), \ldots, \beta(W)$ are well-ordered.*

Proof: 1. is clearly true of vertices. It is true for arcs from the assumption of "small" arcs made without loss of generality in Section 2.2 (Part Assumptions) because then a distinct pair of parallel lines cannot be simultaneously tangent to S.

At points defining a contact configuration, the tangents (locations of the jaws) are parallel. Consider points on P outside S (points outside S are sufficient because of 1.) that have tangents whose slope belongs to $\lambda(S)$. From convexity of P, this set of points is a continuous set. Thus $\beta(S)$ which is the list of features containing this set of points is consecutive. This proves 2.

$\beta(S), \beta(T)$ are well-ordered iff neither $\sigma(\beta(S)), \sigma(\beta(T))$ includes the other. Suppose the latter is not true: let $\sigma(\beta(S)) \subset \sigma(\beta(T))$. This implies that any point forming a contact configuration with S also forms it with T. This violates $\lambda(S)$ and $\lambda(T)$ having no intersection.

Suppose $\beta(S), \beta(T)$ have two (or more) features in common. Since $\beta(S), \beta(T)$ are well ordered, two common features include a vertex and an arc. Consider a point q on the arc infinitesimally close to the vertex. Because of 1. and 2., q forms a contact configuration with some point in S and some point in T. This implies that the slope of the tangent at q exists in both $\lambda(S)$ and $\lambda(T)$ which violates the fact that $\lambda(S)$ and $\lambda(T)$ are distinct. This completes the proof of 3.

There are several cases to consider in 5. but all are similar in proof. We'll just consider one of them. Suppose the minimum angle of $\sigma(\beta(U))$ is less than that of $\sigma(\beta(T))$ (they cannot be equal because otherwise one of these sets would include the other violating well-orderedness of $\beta(T), \beta(U)$ which is true from 3.). Then because of well-orderedness of $\beta(T), \beta(U)$, their maximum angles also have the same order. Consider the minimum angle in $\sigma(\beta(U))$ and the maximum angle in $\sigma(\beta(T))$. Neither belongs to the other's set. Let them be achieved at points q_U, q_T, respectively. That is, there is a point, call it p_U in U forming a contact configuration with q_U, and a point, call it p_T in T forming a contact configuration with q_T; and further, because of their relative placements, p_U, q_U, q_T, p_T form a clockwise ordering which is impossible by convexity. This proves 5.

Finally, 6. follows straight away from 5 by considering all (circularly) consecutive triplets of features. □

The above lemma leads to a simple algorithm for computing every $\beta(S)$.

Explicitly compute $\beta(V_0)$ by marching around the part once and checking for points that have tangents with slopes from $\lambda(V_0)$. This can be done in $O(n)$ time. Compute the other $\beta(S)$ in the order $S_0, V_1, S_1, ..., v_{n-1}, S_{n-1}$ by scanning the part in the clockwise order. This second step can be performed in $O(n + \sum|\beta(S_i)| + \sum|\beta(V_i)|)$ time. Thus, from Lemma 3, we have that the algorithm runs in $O(n)$ time.

Lemma 4

$$\sum_{i=0}^{n}(|\beta(S_i)| + \sum|\beta(V_i)|) = O(n).$$

Proof: Recall from Statements 2,3 of Lemma 3 that each $\beta(S)$ is consecutive and for adjacent features S and T, $\beta(S), \beta(T)$ share at most one common feature. The proof goes through even if $\beta(S), \beta(T)$ are not adjacent.

Therefore, consider the following notationally simpler problem. Given a circular list with n elements and n sublists $H_1, \ldots, H_n$, a sublist consisting of consecutive elements, such that no two sublists have more than one element in common. The goal is to show that $\sum_{i=1}^{n}|H_i| = O(n)$. It is obvious that the lemma is proved if this is true.

To show that all the H_i have together $O(n)$ elements, first remove all the H_i that have at most one element. Wlog let these be the last $n-j$ subsets. So now we have $H_1, H_2, \ldots, H_j$ with at least two elements each and no two intersect in more than one element. This implies that no element exists in more than two sublists. Duplicate each element that exists in exactly two sublists. Now no element exists in more than one sublist, so it is obvious that $\sum_{i=1}^{j}|H_i| \leq 2n$. Also, $\sum_{j+1}^{n}|H_i| <= n - j$. □

Remark: A similar proof holds if the size of each $H_i \cap H_j$ is required to be at most a constant $c > 1$. Then rather than duplication of elements, we may have to make up to $\Theta(c^2)$ copies of elements that repeat in more than one sublist. However, if the H_i were subsets rather than sublists (*i.e.* no ordering among the elements) then the total size of all the H_i can be $\Theta(n^{3/2})$. This is because the problem is identical to "how many edges can a bipartite graph with n vertices in each partition have, if it does not contain $K_{2,c+1}$ as a subgraph." Otfried Schwarzkopf pointed out to us that for $c = 1$, the answer is $\Theta(n^{3/2})$ edges.

3.3 Examples

Example 1 Our first example is a flint-stone- (or obelisk-) shaped part shown in Fig. 7. The three arcs p, q, r are all cubic parametric curves:

$$X_p(u) = -3u + 3u^2 + u^3; Y_p(u) = 9u + 3u^2 - 2u^3.$$

$$X_q(u) = 1 + 6u - 3u^3; Y_q(u) = 10 - 6u - 9u^2 + 5u^3.$$

$$X_r(u) = 6u - 3u^2 + u^3; Y_r(u) = -3u + 3u^2.$$

The three vertices are P(4,0), Q(0,0), R(1,10). Clearly no vertex-vertex extrema are possible (they fail the grasp conditions). The only arc-arc extrema occurs when we consider the pair of arcs (p, q). The equations to be solved simultaneously for u, v (u is the parameter for arc p and v for q) are

$$(-3 + 6u + 3u^2)(-3u + 3u^2 + u^3 - 1 - 6v + 3v^3) +$$

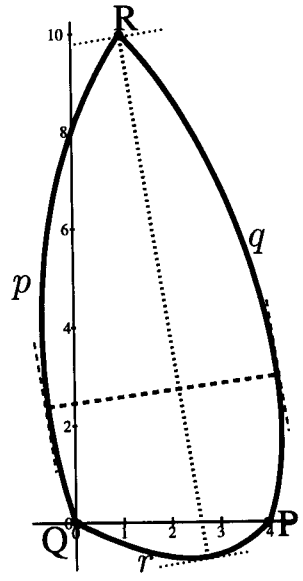


Figure 7: An algebraic part made up of three arcs p, q, r and vertices P,Q,R. Its diameter function possesses a local maximum orientation of the vertex-arc type indicated by a dotted line (making angle $-80.93°$ with the x-axis) connecting R with a point in r. The local maximum orientation is the orientation of the jaws when perpendicular to this line and is thus $90 - 80.93 = 9.07$ degrees. A local minimum orientation of the arc-arc (p, q) exists indicated by the dashed line (making angle $7.5°$ with the x-axis). The local minimum orientation, again the orientation of the jaws when perpendicular to this line, is $97.5°$.

$$\left(9 + 6\,u - 6\,u^2\right)\left(9\,u + 3\,u^2 - 2\,u^3 - 10 + 6\,v + 9\,v^2 - 5\,v^3\right) = 0; \text{ and}$$

$$\left(6 - 9\,v^2\right)\left(-3\,u + 3\,u^2 + u^3 - 1 - 6\,v + 3\,v^3\right) +$$

$$\left(-6 - 18\,v + 15\,v^2\right)\left(9\,u + 3\,u^2 - 2\,u^3 - 10 + 6\,v + 9\,v^2 - 5\,v^3\right) = 0.$$

Maple V returns the (unique, for the range $u, v \in (0, 1)$) solution:

$$\{u = 0.2475949490, v = 0.7090134540\}.$$

This corresponds to the points $(-0.54, 2.38) \in p$ and $(4.18, 3.00) \in q$. 2nd derivative checks may be applied to check that this pair of points defines a local minimum orientation.

The only arc-vertex extrema occurs when we consider the arc-vertex pair r,R. The equation to be solved (u is the parameter for r) is

$$\left(6 - 6\,u + 3\,u^2\right)\left(6\,u - 3\,u^2 + u^3 - 1\right) - -\left(-3 + 6\,u\right)\left(-3\,u + 3\,u^2 - 10\right).$$

Maple V returns the solution $u = 0.593$. This corresponds to the point $(2.71, -0.72) \in r$. The extrema orientation defined by this point and R is a local maximum orientation.

Thus the part has a single local maximum and a single local minimum orientation in its diameter function. Hence its grasp function consists of a single step.

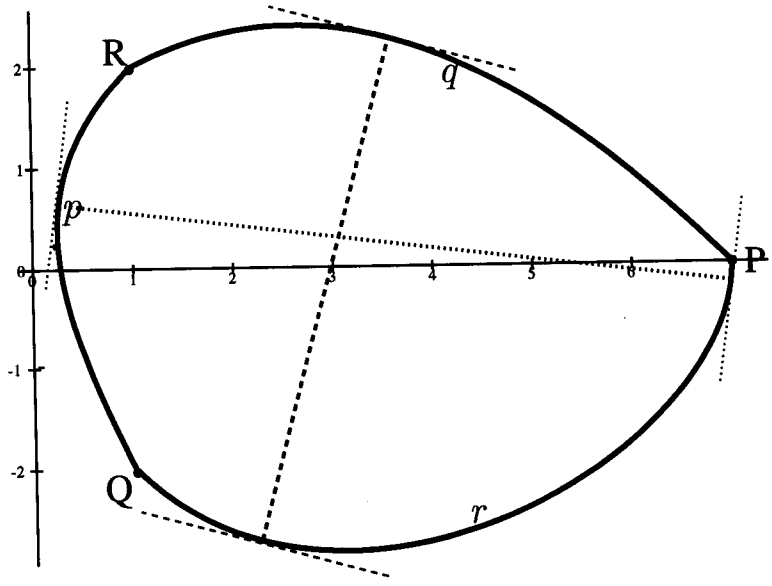


Figure 8: A boulder-shaped part showing two extrema in diameter function: a local maximum at $\sim 83°$ and a local minimum at $\sim 165°$.

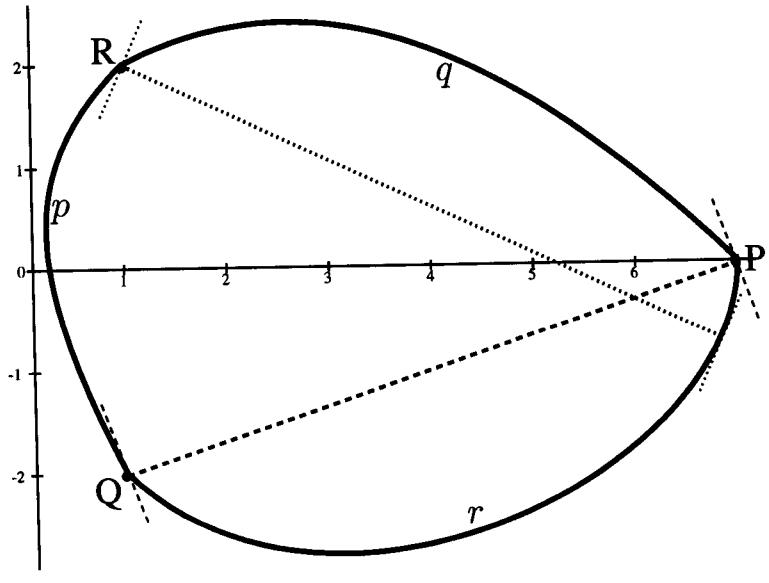


Figure 9: The same part as in Fig. 8 indicating two potential extrema configurations in diameter function. However, these fail grasp conditions and are therefore are not extrema.

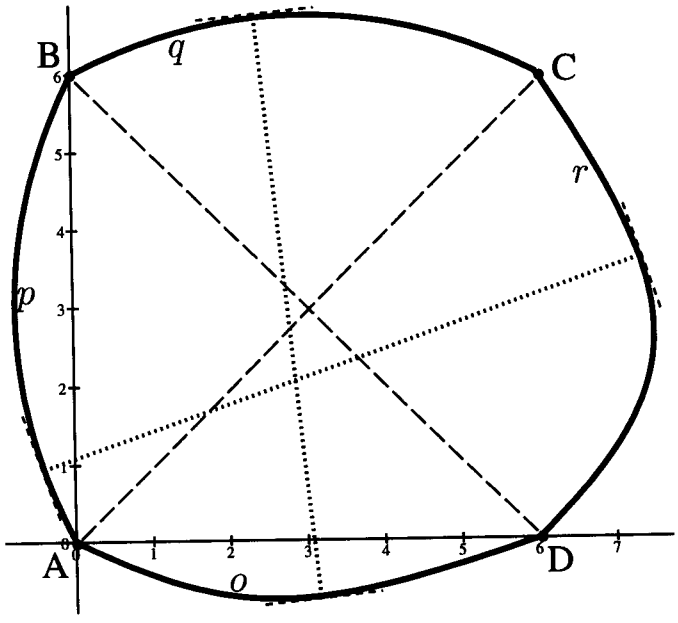


Figure 10: An algebraic part having two maxima and minima in its diameter function. Its convex boundary is made up of four arcs: o, p, q, r and four vertices A,B,C,D. The arcs are cubic parametric curves as before. Both the maxima $(\pi/4, 3\pi/4)$ are vertex-vertex type and indicated by dashed lines joining BD and AC. The minima (5.88°, 108.91°) are arc-arc type: dotted lines joining points in q, o and p, r. The grasp function is plotted in Fig. 11.

Example 2 Our second example is a boulder shaped part and, like the first, its boundary consists of three cubic parametric arcs. Its diameter function has a local maximum and local minimum, both of which are arc-arc extrema. See Fig. 8.

Unfortunately, like the first example, this part too has no other extrema. See Fig. 9. The vertex-arc pair r,R results in the equation

$$(6 + 12\,u - 18\,u^2)\,(6\,u + 6\,u^2 - 6\,u^3) - -(-6 + 24\,u - 12\,u^2)\,(-4 - 6\,u + 12\,u^2 - 4\,u^3)$$

which has a valid solution $u = 0.873$ in the range $(0, 1)$; this corresponds to the point (6.82, -0.75) as indicated in the figure. However, the orientation defined by this point and R(1,2) is not an extremum in the diameter function because it fails the grasp conditions for extremality at R. A similar situation occurs for p,P.

On the other hand, the arc-arc pair p, q and the other vertex-arc pairs result in equations that have no solutions in the required ranges. The vertex-vertex pairs all fail the grasp conditions for extremality. Fig. 9 verifies this for PQ.

Example 3 As we have seen, the two parts in Examples 1,2 have single local minimum orientations in $[0, \pi)$. The grasp function for $[0, \pi)$ therefore consists of a single step; the part will settle into this orientation (or $\pi+$ this local minimum) upon the first grasp. See Fig. 10 for a more interesting example. Its grasp function is in Fig. 11.

From the Grasp Function given in Fig. 11, the algorithm in [32] computes this plan consisting of two grasp actions: the first at $-57.4°$ and the second at 0. With these two grasp actions, the part is oriented up to a symmetry of π, *i.e.* in one of two orientations separated by π. See Fig. 12. To orient the part into a unique orientation in S^1, push grasp actions may be used. See Fig. 17. Both the squeeze grasp plan and the push grasp plans are of optimal length in that there are no plans possible of shorter length that will orient the part up to symmetry beginning from any orientation.

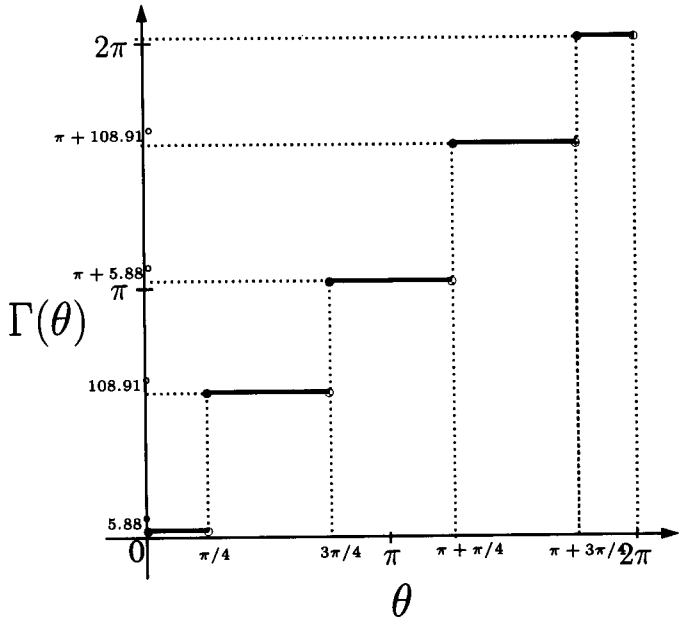


Figure 11: Grasp Function Γ for the part shown in Fig. 10. If θ is the initial orientation of the part wrt the gripper, $\Gamma(\theta)$ denotes its orientation after the grasp. The grasp function has a period of symmetry π, *i.e.* $\Gamma(\pi + \theta) = \pi + \Gamma(\theta)$. Within a period of symmetry of width π, the grasp function has two steps corresponding to the two minima (and maxima) in diameter function.

4 Conclusion

In this paper we have presented an optimal algorithm for computing the (squeeze) grasp function of an part bounded by algebraic curves. Together with the algorithm in [32], this results in a complete algorithm to orient algebraic parts, given their boundary representation, up to symmetry in their diameter functions.

Diameter functions were seen to have a period of symmetry at most π. Therefore squeeze-grasp plans constructed from diameter functions can orient parts at best into two orientations separated by π. Push-grasp plans could be used to uniquely orient most parts. This is because they work with an analogue to the diameter function called the radius function which generally has a period of symmetry 2π. Push-grasps, which also relax the assumption of simultaneous contact of the two jaws with the part, are discussed in the appendix. As an example of a push-grasp plan, see Fig. 13.

The examples we presented (Section 3.3) were implemented using the Computer Algebra package *Maple V* [6]. When they existed, solutions to 5th degree simultaneous equations were obtained in a couple of seconds or less on a Silicon Graphics workstation. Maple would report non-existence of a solution in about five seconds. Thus, the total computation for each of the examples was well under a minute.

Most industrial parts, however complicated their algebraic shape, tend to have few extrema in their diameter and radius functions. This leads to grasp functions of low complexity and small length plans showing th effectiveness of our approach. We are currently working on grasp functions and planning algorithms to orient 3D parts initially resting on a flat surface using a two-fingered gripper.

Acknowledgments
We thank Pankaj Agarwal, Doug Ierardi, Lefteris Melissaratos, Babu Narayanan, Mark Overmars, Govindan Rajeev, Otfried Schwarzkopf, and Chee Yap for several interesting discussions. Olivier Tardieu implemented the algorithm for generalized polygonal parts. Otfried Schwarzkopf's wonderful graphical editor, *Ipe*, greatly helped with the diagrams of this paper.

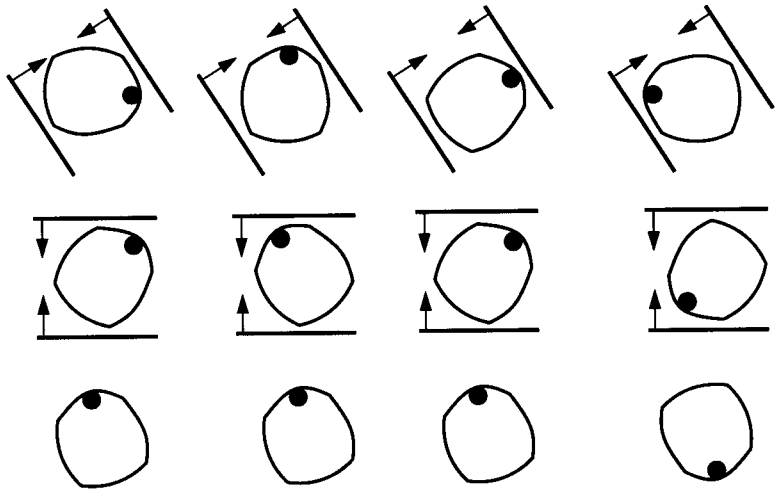


Figure 12: Top view of a two-stage plan for orienting the part in Fig. 10 into one of two orientations separated by π. The part is shown with a "registration mark" to help indicate orientation. Four traces of the plan (running from top to bottom) are shown. Note that the part's initial orientation is different in each trace (top of each column): in world coordinates, they are $0, 90, 40, 180$ degrees. The gripper is oriented at $-57.4°$ in the first stage; thus the orientations of the parts with respect to the gripper are $57.4, 147.4, 97.4, 237.4$. Thus the final orientations of the parts after the first grasp action, *with respect to the gripper* are (from the grasp function in Fig. 12): $108.91, 185.88, 108.91, 288.91$. Now, for the second stage, the jaws turn by $+57.4°$ to come into orientation 0 in world coordinates. Thus the parts are now at orientations $51.51, 128.48, 51.51, 231.51$ wrt the gripper (and in world coordinates since the gripper is at orientation 0 wrt the world frame). The second grasp thus results in the part going into orientations (from grasp function): $108.91, 108.91, 108.91, 180 + 108.91$. The first and fourth columns are 180 out of phase wrt each other at all times.

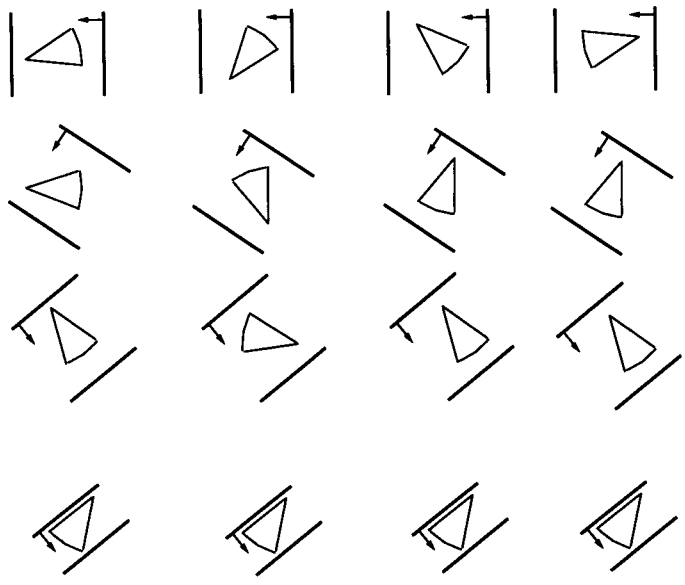


Figure 13: Four traces of 3-step push-grasp plan. The push angles are $0, -56, -128$ degrees. Line with arrow indicates pushing jaw. The initial orientations for the four traces are same as those in Fig. 4. Notice the final orientations in both figures. Initial orientations that are 180° apart (see the first and fourth columns in Fig. 4), end up 180° apart in Fig. 4 (squeeze-grasp actions). However, in this figure, they end up in the identical orientation. Push-grasping resolves the ambiguity.

References

[1] P. K. Agarwal and M. Sharir. Off-line dynamic maintenance of the width of a planar point set. *Comput. Geom. Theory Appl.*, 1:65–78, 1991.

[2] D. Avis. Diameter partitioning. *Discrete Comput. Geom.*, 1:265–276, 1986.

[3] R. V. Benson. *Euclidian Geometry and Convexity*. McGraw-Hill Book Company, 1966.

[4] R. C. Brost. Automatic grasp planning in the presence of uncertainty. *The International Journal of Robotics Research*, 8(1), February 1988.

[5] V. Chandru and R. Venkataraman. Circular hulls and Orbiforms of simple polygons. In *ACM-SIAM Symposium on Discrete Algorithms* (SODA). SIAM-ACM, 1991.

[6] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt. *Maple V Library Reference Manual*. Springer-Verlag, New York, 1992.

[7] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric searching. *Discrete Comput. Geom.*, 10:183–196, 1993.

[8] I. Chen and J. W. Burdick. Finding antipodal grasps on irregularly shaped objects. IEEE *Transactions on Robotics and Automation*, pages 507–511, August 1993. Also in Proc. *Intl. Conf. Robotics & Automation*, 1992.

[9] D. Z. Du and D. J. Kleitman. Diameter and radius in the manhattan metric. *Discrete Comput. Geom.*, 5:351–356, 1990.

[10] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2):201–225, Aug 1993. (Special issue on Computational Robotics).

[11] K. Y. Goldberg and M. Furst. Low friction gripper. U.S. Patent # 5,098,145, March 1992.

[12] P. Gritzmann and M. Lassak. Estimates for the minimal width polytopes inscribed in convex bodies. *Discrete Comput. Geom.*, 4:627–635, 1989.

[13] R. A. Grupen, T. C. Henderson, and I. D. McCammon. A survey of general purpose manipulation. *International Journal of Robotics Research*, 8(1), 1989.

[14] A. Heck. *Introduction to Maple: A computer algebra system.* Springer-Verlag, 1992.

[15] J. Hershberger. Minimizing the sum of diameters efficiently. *Comput. Geom. Theory Appl.*, 2(2):111–118, 1992.

[16] J. D. Hobby. Rasterizing curves of constant width. *J. ACM*, 36:209–229, 1989.

[17] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction.* Computer Graphics and Geometric Modeling. Morgan Kaufmann, San Mateo, CA 94403., 1989.

[18] M. E. Houle and G. T. Toussaint. Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-10(5):761–765, 1988.

[19] R. Janardan. On maintaining the width and diameter of a planar point-set online. In *Proc. 2nd Annu. SIGAL Internat. Sympos. Algorithms*, volume 557 of *Lecture Notes in Computer Science*, pages 137–149. Springer-Verlag, 1991.

[20] S-Y. R. Li. Reconstruction of polygons from projections. *Information Processing Letters*, 28:235–240, 12 Aug. 1988.

[21] L. Lovász and M. Simonovits. On the randomized complexity of volume and diameter. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 482–492, 1992.

[22] M. Mani and R. D. W. Wilson. A programmable orienting system for flat parts. In *Proc: North American Mfg. Research Inst. Conf XIII*, 1985.

[23] M. T. Mason. *Manipulator Grasping and Pushing Operations.* PhD thesis, MIT, June 1982. published in *Robot Hands and the Mechanics of Manipulation*, MIT Press, 1985.

[24] M. T. Mason. On the scope of quasi-static pushing. In O. Faugeras and G. Giralt, editors, *The Third International Symposium on Robotics Research.* MIT Press, 1986.

[25] M. T. Mason, K. Y. Goldberg, and R. H. Taylor. Planning sequences of squeeze-grasps to orient and grasp polygonal objects. Technical Report CMU-CS-88-127, Carnegie Mellon University, Computer Science Dept., Pittsburgh, PA 15213, April 1988.

[26] B. K. Natarajan. Some paradigms for the automated design of parts feeders. *International Journal of Robotics Research*, 8(6):98–109, December 1989. Also appeared in IEEE FOCS, 1986.

[27] 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.

[28] M. A. Peshkin. *Planning Robotic Manipulation Strategies for Sliding Objects.* PhD thesis, Carnegie-Mellon University, Department of Physics, Pittsburgh, Pennylvania, Nov 1986. Also published as a book: *Robotic Manipulation Strategies*, Prentice Hall, 1990, New Jersey.

[29] M. A. Peshkin and A. C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5), October 1988.

[30] K. Pingle, R. Paul, and R. Bolles. *Programmable Assembly: Three Short Examples.* Stanford AI Lab Film, 1974.

[31] A. S. Rao and K. Y. Goldberg. On the recovery of a polygon's shape from its diameter function. In *4th Canadian Conference on Computational Geometry*, pages 210–215, St. John's, Newfoundland, Canada, August 1992. Prelim. version presented at the 1st MSI workshop on Computational Geometry, Oct. 26, 1991, Stony Brook, NY.

[32] A. S. Rao and K. Y. Goldberg. Manipulating algebraic parts in the plane. Technical Report RUU-CS-93-43, Utrecht University, Department of Computer Science, December 1993. Submitted to the IEEE *Transactions on Robotics and Automation*; preliminary version appears in Proc. ICRA 1992 under the title "Orienting generalized polygonal parts." For a compressed postscript version anonymous ftp to `ftp.cs.ruu.nl` and `get /pub/RUU/CS/techreps/CS-1993/1993-43.ps.gz`.

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

[34] G. Rote, C. Schwarz, and J. Snoeyink. Maintaining the approximate width of a set of points in the plane. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 258–263, Waterloo, Canada, 1993.

[35] A. A. Schaeffer and C. J. Van Wyk. Convex hulls of piecewise-smooth Jordan curves. *Journal of Algorithms*, 8(1):66–94, 1987.

[36] R. H. Taylor, M. T. Mason, and K. Y. Goldberg. Sensor-based manipulation planning as a game with nature. In *Fourth International Symposium on Robotics Research*, August 1987.

[37] I. M. Yaglom and V.G. Boltyanskii. *Convex Figures*. Holt, Rinehart and Winston, New York, 1951.

A Radius Function and Push Grasping

Diameter functions have period at most π which implies that the best we can achieve by a sequence of squeeze grasps is one of two final orientations in S^1. To obtain a unique final orientations, push grasp actions may be used. [2] As identified by Brost [4]) a push grasp action consists of a push phase by one jaw towards the other prior to squeezing. Thus, in addition to having periods 2π (generally), push grasps are easer to justify mechanically than squeeze grasps which require simultaneous contact of the two jaws with the part.

A push grasp action requires a sufficient pushing distance [28] and the location of the part's center of mass (com). The mechanics of the push phase is captured by the *push function* defined below [23]. The distance from the part's c.o.m to a line in orientation θ tangent to the part varies with θ. Define the *radius function*, $r : S^1 \rightarrow \Re_+$, to record this variation, *i.e.* $r(\theta)$ equals the distance from the c.o.m. to a tangent line of orientation θ. From Mason [23] and Brost [4], analogous to Fact 1, we get

Fact 2 *Pushing minimizes radius.*

Thus, to compute the push function, we need the extrema in the radius function. This may be done in a manner similar to finding extrema in diameter function. In fact it is simpler. The only events to consider are com-arc and com-vertex events. For the former simply set up the equation as in Equation 2 replacing the coordinates of W by the coordinates of the com. This gives us points on the arc the normal at which passes through the com. Com-vertex events can be maxima in radius function or end points of regions of constant radius, or not extrema. Simple tests may be applied for each case. The push function can therefore also be computed in $O(n)$ time. See Fig.s 14, 15.

The push grasp function, which maps an initial orientation to the corresponding final orientation after the push and squeeze phases, is simply the composition of the squeeze grasp function and the push function because:

Fact 3 *Push-grasping first minimizes radius, and subsequently diameter.*

Since the push function and squeeze grasp function consist of $O(n)$ steps and ramps, and can each be computed in $O(n)$ time,

Theorem 2 *The push grasp function of an algebraic part with n arcs can be computed in $O(n)$ time.*

See Fig. 16. The push grasp function can be input to the algorithm given in [32] to produce a push grasp plan to orient a part up to symmetry in the push grasp function (which is usually of period of 2π). See Fig. 17 for an example push grasp plan.

[2] This is because push grasp functions usually have period 2π rather than π. However in pathological parts (very symmetrical parts with perfectly symmetrically positioned center of mass), push grasp functions can have period less than 2π.

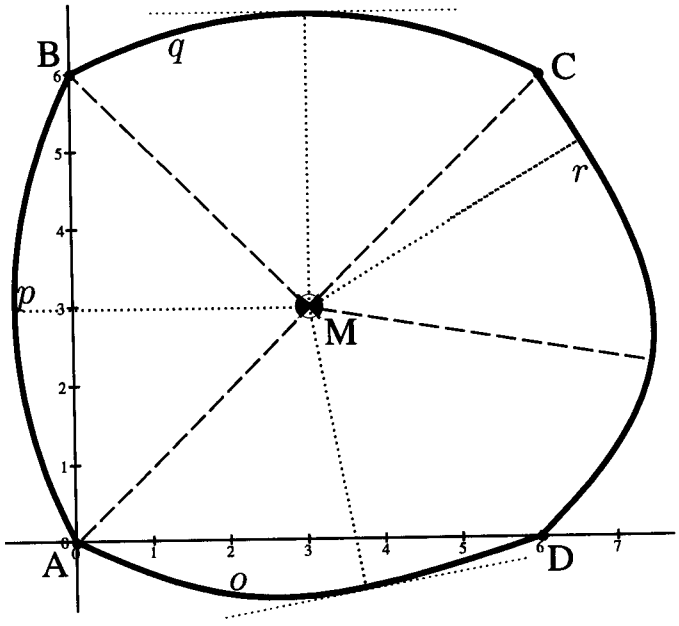


Figure 14: Radius function extrema analysis on the same algebraic part as in Fig. 10. Its center of mass (com) M is assumed to be at point (3,3) indicated by a special symbol in the figure. There are eight extrema, four local maxima and four local minima in the radius function. The local minima are at 0, 90, 190.24, 301.35 degrees and local minima configurations are indicated by dotted lines originating at the com (again, the orientation is that of the jaw perpendicular to these dotted lines). The jaw positions for the 0 and 190.24 configurations are shown in the figure. The local maxima are at 45, 135, 260.35, 315 degrees. Note that orientation 225 (corresponding to MD) is not an extremum. Based on this data, the push-function may be readily constructed (See Fig. 15.)

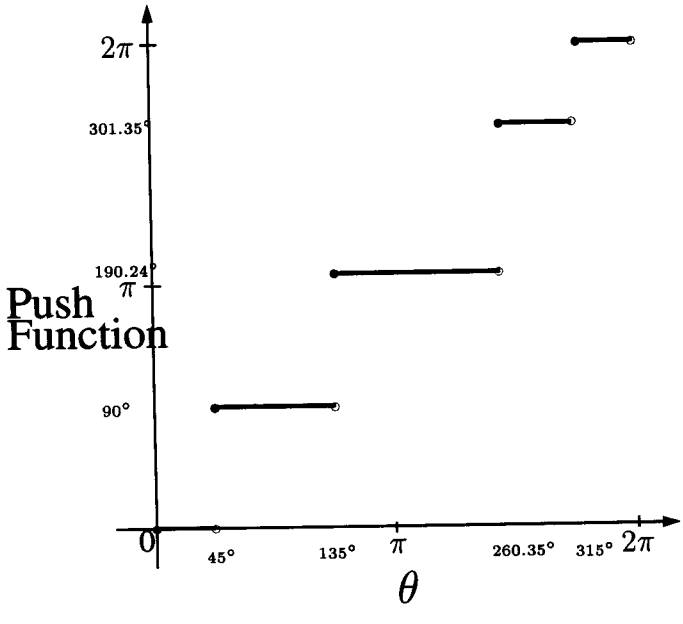


Figure 15: Push function: $S^1 \rightarrow S^1$ for the part shown in Fig. 14. It consists of four steps (the first and the last step are part of the same).

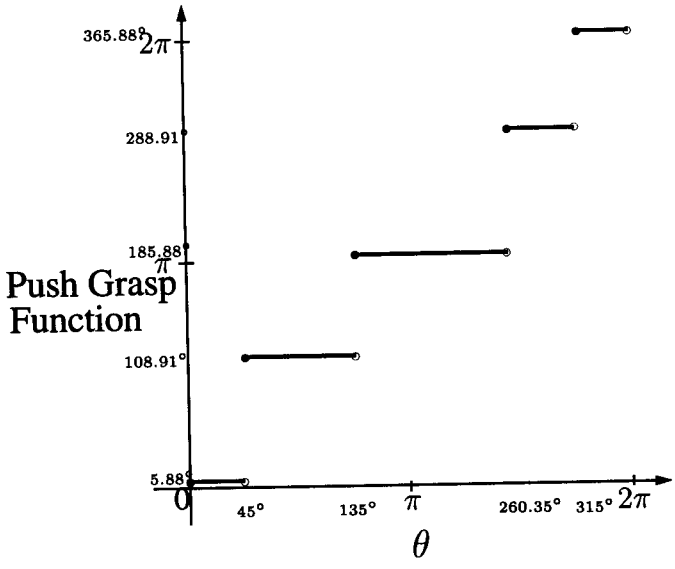


Figure 16: The push-grasp: $S^1 \rightarrow S^1$ function for the same part. It is simply the composition of the (squeeze) grasp function (Fig. 11) and the push function (Fig. 15).

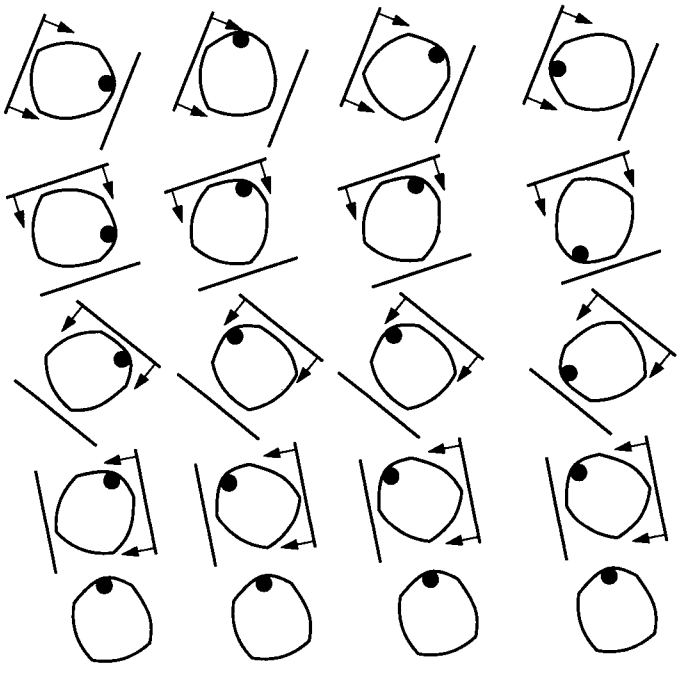


Figure 17: Top view of a four-stage plan for orienting the part in Fig. 14 into a unique orientation using push-grasp actions. Four traces are shown, one per column. The initial orientations of the part (top of each column), as in the squeeze grasp plan (Fig. 12), are $0, 90, 40, 180$ degrees in world coordinates. The first push grasp action is at 67 degrees. Arrows indicate the pushing jaw. The other three are at 17, -40, and -80 degrees. This sequence of push grasp actions orient the part into orientation 106 degrees irrespective of its initial orientation. Note that while the first and fourth columns begin 180 degrees apart here they end up in the same orientation (compare with Fig. 12).