

# Shape from Diameter: Positive Results

Anil S. Rao and Kenneth Y. Goldberg

RUU-CS-93-17

April 1993



**Utrecht University**

**Department of Computer Science**

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

# Shape from Diameter: Positive Results

Anil S. Rao and Kenneth Y. Goldberg

Technical Report RUU-CS-93-17  
April 1993

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

ISSN: 0924-3275

# Shape from Diameter: Positive Results\*

Anil S. Rao<sup>†</sup> and Kenneth Y. Goldberg<sup>†</sup>

## Abstract

Our objective is to automatically recognize parts in a structured environment (such as a factory) using inexpensive and widely-available hardware. We consider the planar problem of determining the convex shape of a polygonal part from a sequence of projections. Projecting the part onto an axis in the plane of the part produces a scalar measure, the *diameter*, which is a function of the angle of projection. The diameter of a part at a particular angle can be measured using an instrumented parallel-jaw gripper.

Previously presented negative results motivate us to consider the problem of *recognizing* a part from a known (finite) set of parts. Given a set of polygonal parts with a total of  $N$  faces, can we find the shortest sensing plan for disambiguating the parts? Only diameters at  $n \leq N$  stable faces can be measured. We construct an internal representation of these stable diameters in  $O(N + n^3)$  time and then give two planning algorithms: one constructs an optimal sensing plan in time  $O(n^4 2^n)$ , the other constructs a suboptimal sensing plan in time  $O(n^2 \log n)$ . Neither plan requires more than  $n$  measurements.

## 1 Introduction

In automated manufacturing it is often useful to sort parts according to shape. A common approach is to use machine vision, which can be sensitive to lighting conditions and requires coordination with a programmable manipulator. In this paper we explore an alternative approach that uses an inexpensive modification to the parallel-jaw gripper. For the class of convex parts with constant polygonal cross section (2.5 D parts), we consider the following problem: recover the shape of a part's cross section by grasping the part with a parallel-jaw gripper and measuring the distance between the jaws: the *diameter* of the part at some angle. Can a sequence of such measurements be used to determine part shape? Note that the answer is clearly negative if we admit curved parts, for example a circle cannot be distinguished from a Reuleaux triangle (cf. the Wankel rotary engine) merely by grasping and measuring its diameter.

Negative results that shape cannot be uniquely recovered even for polygonal parts are presented in [25]: for a given set of diameter measurements: there is an (uncountably) infinite set of polygonal shapes consistent with these measurements.

These results motivate us to consider the problem of *recognizing* a part from a known (finite) set of parts. The challenge is to plan a sequence of grasp angles to efficiently recognize the part, taking into account the induced rotation of the part caused by the gripper. In other words, given a set of polygonal parts with a total of  $N$  faces, find the shortest sensing plan for disambiguating the parts using a parallel-jaw gripper. See Fig. 1 for an example.

---

\*This material is based upon work supported by the National Science Foundation under Award No. IRI-9123747 and ESPRIT Basic Research Action No. 6546 (project PROMotion).

<sup>†</sup>Department of Computer Science, Utrecht University, Padualaan 14, Postbus 80.089, 3508 TB Utrecht, Nether-

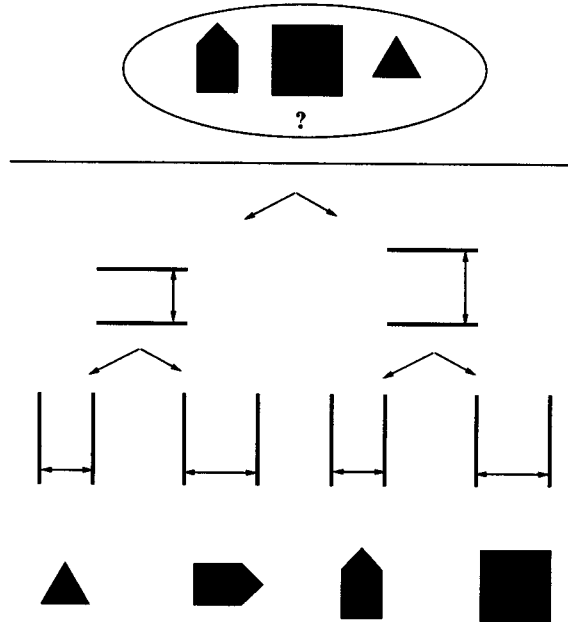


Figure 1: An example grasp plan for distinguishing the three parts shown at the top.

Since diameters at only stable orientations can be measured, we build a graph representation,  $G$ , of the  $n \leq N$  stable diameters in  $O(N + n^3)$  time. Our positive results are two planning algorithms: one constructs an optimal sensing plan in time  $O(n^4 2^n)$ , the other constructs a suboptimal sensing plan in time  $O(n^2 \log n)$ . Neither plan requires more than  $n$  on-line measurements.

Our solution is an example of *active sensing* [1], where the sensor is purposefully moved to acquire appropriate data. More specifically, it is an example of *sensor-based manipulation planning* [32], where the part is moved in the process of acquiring sensor data. Thus the planner requires a model of mechanics to predict part motion. In [32], a brute-force search was used to find optimal plans for the related problem of determining part orientation with a standard parallel-jaw gripper. Here, we consider a *frictionless* parallel-jaw gripper, modified with a linear bearing to insure a finite range of possible diameters [15]. We analyze the computational complexity of finding optimal and suboptimal plans for recognizing a part from a known set.

## 1.1 Preliminaries

For sake of completeness, we present some preliminary definitions similar to the ones presented in [25]. Let  $S^1$  denote the space of planar orientations  $[0, 2\pi)$  and  $\mathcal{R}_+$  the set of positive reals. Given a fixed part  $P$  in an  $x - y$  coordinate frame, the diameter function  $d : S^1 \rightarrow \mathcal{R}_+$  of  $P$  can be formally defined as follows. Imagine two (infinite) parallel lines  $l, h$  (supporting lines) both making angle  $\phi$  with the  $x$ -axis, just touching  $P$  so that  $P$  lies entirely in the region between the two lines. In such a case we say that the supporting lines  $l, h$  are at *orientation*  $\phi$  with respect to the (fixed) polygon  $P$ . The diameter of  $P$  at orientation  $\phi$ ,  $d(\phi)$ , is the distance between the two lines that

---

lands. anil@cs.ruu.nl, Tel: 31-30-533922, FAX: 31-30-513791.

<sup>†</sup>Department of Computer Science, Institute of Robotics and Intelligent Systems, 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.

are at orientation  $\phi$ . The diameter function is continuous and has period  $\pi$ . See Fig. 2. Also, the diameter function of a part is the diameter function of its convex hull. Therefore we can only seek shape recovery of the *convex hull of a part* from its diameter function [13].

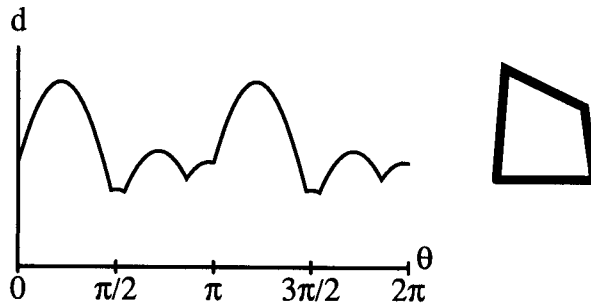


Figure 2: The diameter function for the four-sided part shown at the right.

Call a diameter function *valid* if it is the diameter function of a polygon. Valid diameter functions are characterized as a class of piecewise sinusoidal functions in [25]. The negative results in [25] for shape recovery assume that no *a priori* knowledge about the part except that it is a polygon. Section 2 assumes that the given part has one among a known set of polygonal cross sections. We also assume a frictionless parallel jaw gripper [15] equipped with a linear position sensor (possibly noisy) for taking diameter measurements. We briefly discuss mechanics of the parallel jaw gripper grasping a polygonal object [13]. We only consider *stable* diameter measurements, *i.e.* we assume that, upon grasping, the part settles into a stable equilibrium orientation before the diameter at that orientation is measured. The aim is to compute a “grasp strategy”, *i.e.* a sequence of stable diameter measurements, that would distinguish between the parts up to part distinguishability. We give algorithms that consider tradeoffs between preprocessing time and the length of the grasp strategy produced.

The results of this technical report and those in [25] will appear in the International Journal of Robotics Research [26].

## 1.2 Related work

Our work has relationship with the concept of geometric probing [9]. Under the Cole-Yap model of probing,<sup>4</sup> Belleville [2] proves that the problem of determining whether  $K$  probes are sufficient to identify a polygon from among a set is *NP-hard*.

A complete description of results on geometric probing appear in [29]. Skiena [30] provides a summary of the basic results and open problems in this area. However, some of the basic assumptions common in geometric probing are quite unrealistic in practice. Some examples of such assumptions are that the part is not disturbed by the probes, the probes are accurate (travel in straight lines and return error-free information), and probes are powerful (can recover information like normals from the contact face, *etc.* Some of the more relevant papers in probing are discussed briefly below.

[18] investigates tactile exploration of objects using a parallel jaw gripper. He considers how a parallel-jaw gripper equipped with a tactile array can be used to measure physical properties of a part such as weight and stiffness. [17] discusses an algorithm for the approximative reconstruction of

<sup>4</sup>A probe is a line directed at a polygon. The result of the probing is the point of contact.

a planar convex body from its projections in a finite number of directions. However, by “projections” of a body in a particular direction  $\phi$ , they understand the length of the intersection of a line (with orientation orthogonal to  $\phi$ ) with the body as a function of the distance of the line from the origin. Thus, their idea of “projections” is quite different from “projection probes.” *Projection probes* [19] are a special type of *hyperplane probes* [10]. A projection probe slides a straight line in a direction normal to itself over the plane until it hits the polygon. When it does, it returns the projection (or image) of the polygon on the line. From  $3n - 2$  projection probings, [19] shows that you can recover the complete shape of the polygon. If the sliding straight line makes an angle  $\phi$  with the  $x$ -axis, then the *length* of the projection is the diameter at orientation  $\phi \pm \pi/2$ . Boissonat and Yvinec [4] consider probing of non-convex objects – both polygons and polyhedra. They give algorithms that exactly recover shape using powerful projection probes that return not only point of contact with object but also the direction of normal there. Our negative results [25] therefore imply that for shape recovery, both the length of the projection, and the offset information (namely where the projection lies with respect to some fixed point on the sliding line) are necessary, and merely the length of the projection (diameter information) is not sufficient.

Wallack and Canny consider shape and pose recognition by scanning light beam projections that are orthogonal to the plane of the part [33]. Only a few high resolution scan lines are used and the pose algorithm runs in  $O(n)$  time (on the average) for an  $n$ -sided polygonal part. In [35] they consider light beams that are in the plane of the part which can be used to measure diameter without affecting part orientation. In contrast, grasping actions with a parallel jaw gripper can be used to measure diameter *after rotating the part into a stable orientation* prior to the measurement.<sup>5</sup> [34] considers the problem of determining part orientation by planning an sequence of diameter measurements. Our’s is the more general problem of recognizing a part from a known set. Also, we explore the computational complexity of planning an optimal sequence.

Our work on recognizing part shape also has some relation with robotic inspection [31] and the work of Ellis [11] concerned with tactile data for recognition. Chen and Ierardi [7] consider the problem of recognizing a set of polygonal parts from diameter measurements. However, they do not consider optimal plans. Kang and Goldberg [16] consider a similar problem where the diameter measurements are corrupted by Gaussian noise. They show that data from a sequence of random grasps can be processed using a Bayesian estimator. However random grasping has low average-case performance when part shapes are similar. The relation between planning manipulation strategies and searches through trees was observed in [12, 21, 32]. The notion of *active vision*, servoing sensors so as to obtain information efficiently is popular in vision recognition strategies [27]. Ours is also an active sensing strategy in that we seek a best next grasp for the part based on the previous ones.

Finally, our deployment of the parallel-jaw gripper and simple diameter sensing is in tune with Canny’s and Goldberg’s Reduced Intricacy in Sensing and Control, or RISC, robotics [6]. They observe that for certain industrial tasks, complex (high degree of freedom) manipulators and sensors are unnecessary and that simple, robust, and inexpensive hardware is called for. In this paper, we demonstrate a RISC approach to shape recognition.

## 2 Positive results : recognizing polygonal parts from a known set

The previous sections discussed some negative results for shape recovery from diameter function. However, note that we assumed nothing about the shape of the part except for it being a polygon. In this section, we consider the following problem. We have a part  $P$  whose shape is unknown but

---

<sup>5</sup>We use such measurements in Section 2

for being one of  $k$  planar polygonal parts  $Q_1, \dots, Q_k$  of known shapes. Furthermore,  $P$  is known to lie flat on a table between the jaws of a frictionless parallel jaw gripper equipped with a linear position sensor capable of measuring, up to an error  $\epsilon$ , the distance between the jaws. The gripper can be rotated about an axis perpendicular to the table on which the part rests. The problem is to determine the shape of  $P$  using a minimum number of diameter measurements on the part.

The determination of  $P$  is up to  $\sigma$ -equivalence between parts – a notion that will be defined formally in Section 2.2. For now we could understand it in light of the negative results [25]. For example, parts that have the same diameter function and are indistinguishable to diameter measurements alone. Also, the sensor error  $\epsilon$  can play a factor in reducing distinguishability between parts.

Section 2.4 gives a simple test to check whether two parts are  $\sigma$ -equivalent. This test may be applied *a priori* to the set of given parts and we replace each subset of  $\sigma$ -equivalent parts by a single representative part (any single part from the subset). Thereafter we may deal with the problem of recognizing parts from among a set of parts, no two of which are  $\sigma$ -equivalent.

We begin in section 2.1 with a brief description on the mechanics of the parallel jaw gripper<sup>6</sup> and by defining the notion of a *stable grasp* on the part by the gripper. The only diameter measurements allowed are those in which the part is in a stable grasp by the gripper. There are at most  $r$  stable grasps on an  $r$ -sided polygonal part. Section 2.4 discusses the representation of all stable grasps of the system (*i.e.* all stable grasps of the known parts  $Q_1, \dots, Q_k$ ) as a graph data structure. If  $N$  is the total number of faces in all the polygons, and  $n \leq N$  the number of stable orientations, the computation of this data structure requires  $O(N+n^3)$  time. We use the terms “planning algorithm” and “grasp strategy” to denote different entities. “Planning algorithm” (or “planner”) refers to the off-line preprocessing done on the graph data structure. The output of this preprocessing is a “grasp strategy” (or “grasp plan” or just “plan”) which is run on-line. The grasp strategy gives a sequence of angles at which grasps must be applied on the part in order to recognize it. In general, the next grasp angle will depend on the current diameter measured. The grasp strategy is “tree-like,” every node of the tree being associated with some subgraph of  $G$ . The root is associated with the whole of  $G$  and children of a node are associated with disjoint portions of the subgraph of the parent. This partition of the subgraph associated with the parent is determined by the current diameter value measured. We begin at the root, and travel down along a path of the tree progressively refining the search by pruning the graph. At an intermediate node of the path, we choose the correct child to visit next depending on the diameter value measured. Search trees are popular in the AI world. For example see [28]. By the *length* of the grasp strategy (or plan), we understand the maximum depth (number of grasps) of any leaf in the tree representing the strategy.

There are several measures of complexity here : the preprocessing time (complexity of the “planning algorithm”); worst case number of required grasps of the grasp plan to identify the object (“height” of the tree representing the grasp plan); and amount of on-line work to be done per grasp. Section 2.5 discusses tradeoffs between preprocessing time and length of the plan. Given the data structure representing the stable orientations of the parts, we present two planning algorithms. One (Section 2.5.1) requires  $O(n^2 \log n)$  time and produces a non-optimal grasp plan of length at most  $n$ . The second planning algorithm (Section 2.5.2) runs in  $O(2^n n^4)$  time and produces a grasp plan of optimal length. Both grasp strategies require  $O(\log n)$  on-line time per grasp. This can be improved to constant time by using efficient hashing techniques.

---

<sup>6</sup>Further details, including the design of such a device, may be found in [13]

## 2.1 Gripper and the grasping process

By a *parallel jaw gripper* (denoted by  $J$ ) we understand a gripper consisting of two linear jaws arranged in parallel. Let  $L, H$  denote the lower jaw and upper jaw, respectively. Further assumptions:

1. The part  $P$  is a rigid planar polygonal object resting flat on a table.
2. The part's initial position is unconstrained as long as it wholly lies between the two jaws. The part remains between the jaws throughout grasping.
3. The motion of the gripper  $G$  is orthogonal to the jaws  $L, H$ .
4. All motion occurs in the plane and is slow enough that inertial forces are negligible. The scope of this *quasi-static* model is discussed in [20, 22].
5. Both jaws make contact simultaneously (pure squeezing).
6. Once contact is made between a jaw and the part, the two surfaces remain in contact throughout the grasping motion. The action continues until further motion would deform the part. The part and the gripper are now said to be in the *grasped state*.
7. The gripper can be rotated about an axis orthogonal to the table.
8. There is zero friction between the part and the jaws. See [13] for a design validating this assumption. Essentially, this can be achieved by incorporating one of the jaws with a frictionless bearing.
9. In a grasped state, the distance between  $L, H$  is measurable by a linear position sensor up to an error  $\epsilon$ .

Most of these assumptions are essentially the same as those made in [14, 24] (work dealing with orienting parts) except Assumption 9 which is new. In Assumption 7, it is not necessary that the gripper be rotatable precisely. Some margin in error of part orientation with respect to the gripper is allowed. This issue is studied in [14]. Also, looking ahead at the planning algorithms that follow in Section 2.5, we may note that we always choose orientations to rotate the gripper from among a set of intervals of orientations such that any orientation from this set is applicable. Thus, if we have an inaccurately rotating gripper, we may choose the middle most orientation in the largest interval to allow for maximal error in rotation. Assumption 5 can be relaxed to *push-grasp* actions [5] in which one of the jaws, say  $L$ , first makes contact with the part and pushed the part against the other jaw before squeezing. For simplicity, of presentation, we retain this assumption of pure squeezing. See Fig. 3.

### 2.1.1 The transfer function $\Gamma$

The transfer function of a part  $\Gamma : S^1 \rightarrow S^1$  records the change in orientation of a part after it is grasped (refer to assumptions 4-6 above). That is, if  $\theta$  denotes an orientation of  $P$  with respect to  $G$  in the contact state,  $\Gamma(\theta)$  denotes the orientation in the grasped state. The connection between the diameter and transfer functions is the following rule.

*General principle of frictionless squeezing:* The grasping process tends to (locally) minimize diameter.

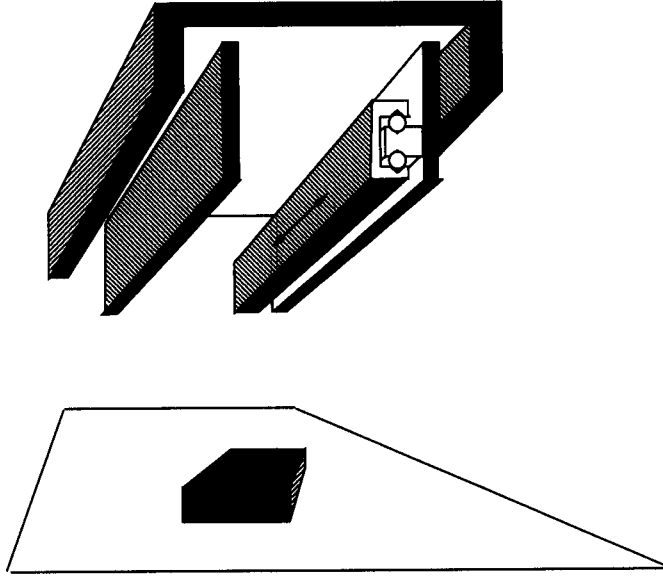


Figure 3: Schematic of the frictionless parallel jaw gripper poised above a rectangular part.

This gives a simple procedure for computing the transfer function from diameter function. Let  $x, y$  be adjacent local maxima in  $d$  and let  $z$  be the unique local minima between them. Then all initial orientations in the region between  $x$  and  $y$  fall to  $z$  after the grasp. Each such region between two adjacent local maxima is called a *step*. In [23] we show that, with a minor modification to the grasping process (replacing one grasp with three mini-grasps), all steps may be assumed left-closed and right-open. Thus  $\forall \phi \in [x, y), \Gamma(\phi) = z$ . For polygonal parts  $\Gamma$  is a step function with a fixed point<sup>7</sup> in the interior of each step. The points of discontinuity in  $\Gamma$  correspond to local maxima in  $d$  and fixed points correspond to local minima. See Fig. 4 for an example.

Orientations that are local minima in the diameter function, or equivalently, fixed points in the transfer function, are called stable (equilibrium) orientations. An  $n$ -gon has at most  $n$  stable orientations [25].

## 2.2 The sensor function and $\sigma$ -equivalence

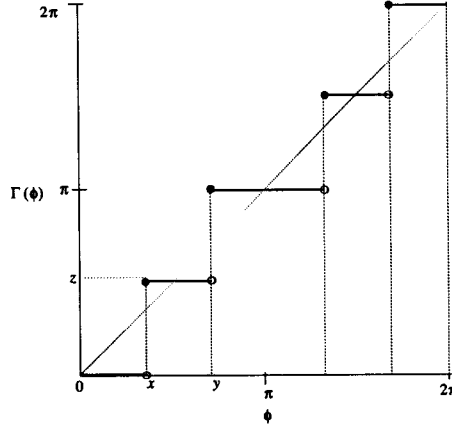
Given a part  $P$  with known diameter function  $d$  and transfer function  $\Gamma$ , its *sensor function*,  $\sigma : S^1 \rightarrow \mathbb{R}_+$  is the composition of  $d$  and  $\Gamma$ , *i.e.*  $\forall \theta, \sigma(\theta) = d(\Gamma(\theta))$ . For polygonal parts, if  $|\Gamma|$  denotes the number of steps in transfer function  $\Gamma$ , then  $\sigma$  is a step-function with at most  $|\Gamma|$  steps.<sup>8</sup> Given  $P$  initially in orientation  $\theta$  with respect to the gripper, a grasp action at  $\alpha$  will return a measurement of  $\sigma(\alpha + \theta)$ . The new orientation,  $\theta'$  of the part with respect to the gripper will be  $\Gamma(\alpha + \theta)$ .

Let  $f$  be some function whose domain is  $S^1$ . Then  $f^{(x)}$  denotes the function  $f$  normalized with respect to orientation  $x$ , *i.e.*

$$\forall \theta \in S^1, f^{(x)}(\theta) = f(\theta + x \bmod 2\pi).$$

<sup>7</sup>An orientation  $z$  such that  $\Gamma(z) = z$ .

<sup>8</sup> $\sigma$  could have less than  $|\Gamma|$  steps if two adjacent steps in  $\Gamma$  result in the same diameter.



Discontinuities in the transfer function occur at local maxima in the diam. function. Every step in the tr. function has a fixed point at the intermediate local minimum.

Figure 4: Transfer function of a rectangular part.

Two parts  $P_A, P_B$  with sensor functions  $\sigma_A, \sigma_B$  are  $\sigma$ -equivalent if there exist stable orientations  $\beta_A, \beta_B$  of  $P_A, P_B$ , respectively, such that

$$\forall \theta \in S^1, \sigma_A^{(\beta_A)}(\theta) = \sigma_B^{(\beta_B)}(\theta).$$

If two parts  $P_A, P_B$  are  $\sigma$ -equivalent, we refer to the orientations  $\beta_A, \beta_B$  in the above definition as the witnessing orientations for the  $\sigma$ -equivalence. Alternatively, we say that the parts  $P_A, P_B$  are  $\sigma$ -equivalent at orientations  $\beta_A, \beta_B$ . Replacing  $\sigma$  by  $d$  or  $\Gamma$ , we obtain definitions for parts being  $d$ -equivalent and  $\Gamma$ -equivalent, respectively. Notice that  $d$ -equivalence implies  $\Gamma$ -equivalence and also  $\sigma$ -equivalence. However,  $\sigma$ -equivalence does not imply  $\Gamma$ -equivalence. See Fig. 5.

### 2.3 Problem statement

The shape identification problem is defined below.

Given a set of  $k$  polygonal parts,  $\{P_1, P_2, \dots, P_k\}$ , no pair of which are  $\sigma$ -equivalent, with a total of  $N$  faces and  $n$  stable equilibrium orientations,  $n \leq N$ ; find a sensing plan consisting of parallel-jaw gripper grasp actions for identifying each part such that, if  $X_i$  is the maximum length of a sequence of grasp actions to identify part  $P_i$ , then  $\max_i X_i$  is minimized.

We call a tree of grasp actions as a *grasp plan* or grasp strategy.  $\max_i X_i$  is called the *grasp length* of the grasp plan. The problem as stated above asks for the *optimal grasp plan*, a grasp plan of shortest grasp length. See Fig. 1 for an example in which each  $X_i = 2$ . Our results are two planning algorithms: one that constructs an optimal grasp plan (Section 2.5.2) in time  $O(n^4 2^n)$ , and the other constructs a suboptimal sensing plan (Section 2.5.1) in  $O(n^2 \log n)$  time. Neither plan requires more than  $n$  on-line diameter measurements. Both the planning algorithms operate on an internal representation of the stable equilibrium orientations which is computable in  $O(N + n^3)$  time (Section 2.4).

The problem as stated above requires no two parts in  $\{P_1, P_2, \dots, P_k\}$  to be  $\sigma$ -equivalent. The parts are identified uniquely in a known final orientation. While it is extremely unlikely that a given

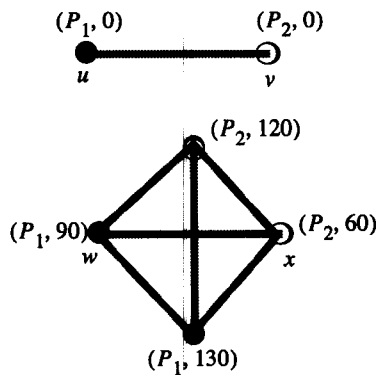
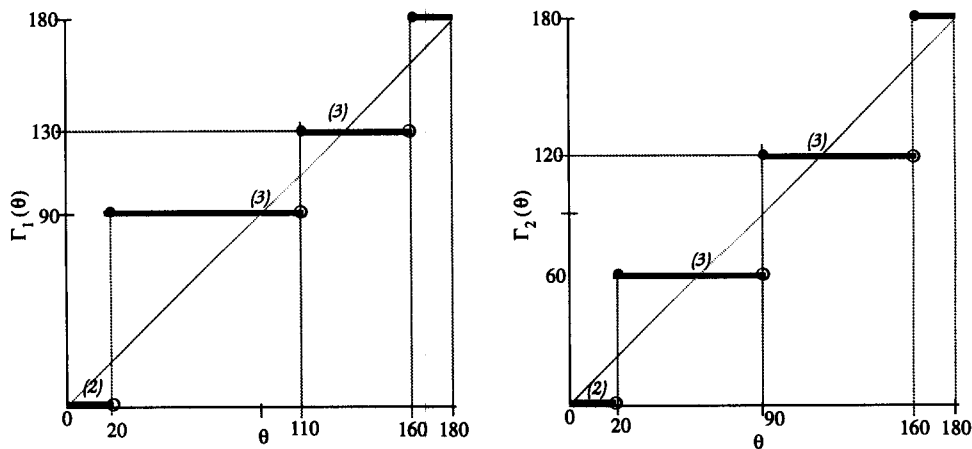


Figure 5:  $\sigma$ -equivalence. At the top are different transfer functions with three steps each. Only the portion between  $[0, \pi)$  (the period of symmetry) is shown. The transfer function on the left,  $\Gamma_1$  has fixed points  $\{0, 90, 130\}$  degrees in and for  $\Gamma_2$  they are  $\{0, 60, 120\}$ . The diameter value to which each step collapses to is shown in parenthesized italics just above the step. Let the parts that correspond to these transfer functions be  $P_1, P_2$ , respectively. The graph of stable diameters for these two parts,  $G(P_1, P_2)$  is shown below. A vertex is shown as a (part, stable orientation) tuple. For example,  $u$  is the vertex  $(P_1, 0)$  and  $v$  is  $(P_2, 0)$ .  $R((u, v))$  can be verified to be empty.  $P_1, P_2$  are  $\sigma$ -equivalent with  $(0, 0)$  being the witnessing orientations. Given parts  $P_1$  and  $P_2$ , both in orientation 0, no grasp action can distinguish between them.  $\{u, v\}$  is one maximal clique in  $G(P_1, P_2)$ . The other maximal clique consists of four vertices and all the six edges  $e$  in that clique have non-empty  $R(e)$ .

set of factory parts will contain  $\sigma$ -equivalent parts, detecting  $\sigma$ -equivalent pairs is straightforward after construction of our graph representation of the parts (Section 2.4).

## 2.4 Representation of stable orientations

Let the error in the linear position sensor be  $\epsilon > 0$ . All of the analysis that follows is based on a representation of the  $k$  parts as a *graph of stable diameters*,  $G_\epsilon$ , which we now define.  $G_\epsilon = (V, E_\epsilon)$  is an undirected graph with its vertex set  $V$  in a 1-1 correspondence with the set of stable equilibrium orientations (within  $[0, \pi)$ ) of all the  $k$  polygons. Assume that every vertex  $v$  in  $G$  has the following information associated with it:  $\phi_v$ , the stable orientation it corresponds to;  $P_v$ , the associated part;  $d_v$ , the diameter function of  $P_v$ ; and  $\Gamma_v$ , the transfer function of  $P_v$ . Define the operation,  $\stackrel{\epsilon}{=}$  (equality up to  $\epsilon$ ) as  $a \stackrel{\epsilon}{=} b$  if  $|a - b| \leq \epsilon$ . An edge between two vertices exists if and only if the corresponding two stable orientations have diameter values equal up to  $\epsilon$ . That is

$$E_\epsilon = \{(u, v) \mid u, v \in V, d_u(\phi_u) \stackrel{\epsilon}{=} d_v(\phi_v)\}$$

The graph becomes more dense as  $\epsilon$  increases. In future, we drop the subscript  $\epsilon$  on  $G, V$  and  $E$  for convenience.

Let  $G$  have  $n$  vertices and  $m$  edges. Let  $|T|$ , where  $T$  is an induced subgraph of  $G$ , denote the number of vertices in  $T$ . If  $\epsilon$  is sufficiently small,<sup>9</sup> then  $G$  partitions itself into disjoint maximal cliques (completely connected subset of vertices). Otherwise, every connected component of  $G$  need not be a clique. For simplicity, in future we assume that all connected components are cliques. Very similar analysis applies to non-clique components.

The maximal cliques of  $G$  can be found in  $O(n + m)$  time by finding connected components. Each maximal clique  $C$  has an associated diameter value  $\delta_C$  which could be chosen as the average of the elements of the set  $\{d_v(\phi_v) \mid v \in C\}$ .

**Attributes on edges** Unequality up to  $\epsilon$ ,  $\stackrel{\epsilon}{\neq}$ , is defined so that  $a \stackrel{\epsilon}{\neq} b$  if  $|a - b| > \epsilon$ . Let  $e = (u, v)$  be an edge in  $G$ . Then, the attribute associated with edge  $e$  is a subset of  $S^1$ ,  $R(e)$ , defined by

$$R(e) = \{\phi \mid d_u(\Gamma_u(\phi_u + \phi)) \stackrel{\epsilon}{\neq} d_v(\Gamma_v(\phi_v + \phi))\}.$$

$R(e)$  may be termed the *set of resolving orientations* for edge  $e = (u, v)$  for the following reason. Suppose we have one of the two part-orientation pairs:  $P_u$  in orientation  $\phi_u$  or part  $P_v$  in orientation  $\phi_v$ . These have the same diameter (up to  $\epsilon$ ) (because of the existence of edge  $e$ ). However, applying a grasp action taken from  $R(e)$  will result in different (up to  $\epsilon$ ) diameters identifying the part uniquely. The next lemma gives the complexity of computing  $R(e)$ . Let  $|\Gamma|$  be the number of steps in transfer function  $\Gamma$ .

**Lemma 1**  $R(e)$ , where  $e = (u, v) \in E$ , can be constructed in  $O(|\Gamma_u| + |\Gamma_v|)$  time.

**Proof:**  $R(e)$ , where  $e = (u, v)$ , can be seen to be

$$R(e) = \{\phi \in S^1 \mid \sigma_u^{(\phi_u)}(\phi) \neq \sigma_v^{(\phi_v)}(\phi)\}.$$

Therefore, first construct functions  $\sigma_u^{(\phi_u)}$  and  $\sigma_v^{(\phi_v)}$ . Scan them from left to right to determine  $R(e)$ .  $\square$

---

<sup>9</sup>more precisely, if  $\epsilon < X/n$ , where  $X$  is the smallest gap among the set of distinct stable diameters

**Theorem 1** Given a set of planar polygonal parts  $P_1, \dots, P_k$  with  $N$  faces in total (and  $n \leq N$  stable orientations), its internal representation, i.e. the graph of stable diameters,  $G$ , consists of  $n$  vertices and can be computed in  $O(N + n^3)$  time.

**Proof:** In  $O(N)$  time it is possible to compute  $n$ , the number of stable orientations of the parts. This determines the vertex set of  $G$ . The associated information for each vertex  $v$  can be computed in  $O(n \log n)$  time. Summing over all vertices, we get  $O(n^2 \log n)$  time. The edge set can now be computed in  $O(n^2)$  time. Computation of a each  $R(e)$  takes  $O(n)$  time (Lemma 1) and therefore, computation of all the  $R(e)$  takes  $O(n^3)$  time.  $\square$

**Testing  $\sigma$ -equivalence** If  $R(e) = \emptyset$ , for some edge  $e = (u, v) \in G$ , then the parts  $(P_u, P_v)$  are  $\sigma$ -equivalent at  $\phi_u, \phi_v$ , and vice versa. This gives a test for  $\sigma$ -equivalence between parts.

## 2.5 Planning algorithms and grasp strategies

For an induced subgraph  $G' \subset G$ , the cardinality of  $\{P_v | v \in G'\}$  is called its heterogeneity. If this number is one, then the subgraph is termed homogeneous, and heterogeneous otherwise.

The first grasp on the part, which is randomly applied, and returns a diameter value which is “hashed” into one of the maximal cliques  $C \subset G$ . If this clique  $C$  is homogeneous, the part is identified. Otherwise additional grasping is required to resolve the heterogeneous clique. The next grasp should a useful grasp giving us new information that will help identifying homogeneous sub-cliques. To determine useful grasps we need the attributes on the edges defined before.

**Resolving edges.** To resolve a heterogeneous 2-clique  $\{u, v\}$  connected by the edge  $e$ , where  $R(e) \neq \emptyset$ , simply rotate the gripper by any angle<sup>10</sup> in  $R(e)$ , squeeze the part and measure the diameter. By the definition of  $R(e)$ , one can now identify the part based on this last value measured.

If  $R(e) = \emptyset$ , the two parts are  $\sigma$ -equivalent and no grasp can distinguish between them.  $\sigma$ -equivalent parts include parts with identical diameter function.

**Resolving cliques with more than 2 vertices.** First, extend  $R$ , the resolving set of orientations, to an induced subgraph or any set of edges  $Z$  by defining<sup>11</sup>

$$R(Z) = \bigcap_{e \in Z} R(e).$$

If  $R(C) \neq \emptyset$ , for a clique  $C$ , do as before, i.e. rotate the gripper by any<sup>12</sup> orientation in  $R(C)$  and squeeze. The measured diameter, by the definition of  $R(C)$ , can identify the part. However, if  $R(C) = \emptyset$ , a single additional grasp will not suffice.

<sup>10</sup>or by a “best” orientation  $\phi_{\max}$  maximizing the absolute diameter difference,  $|d_u(\Gamma_u(\phi_u + \phi)) - d_v(\Gamma_v(\phi_v + \phi))|$ , over all  $\phi \in S^1$ . This would allow for maximum sensor error and is easy to determine during the computation of  $R(e)$  given in the proof of Lemma 1 in section 2.4. However, if we want to allow for maximal error in gripper rotation, we can choose to take the middle orientation in the largest continuous interval of orientations in  $R(e)$ .

<sup>11</sup>Strictly, if  $Z$  has heterogeneity  $h < |Z|$ , it is sufficient to consider the subset of edges  $Z' = \{(u, v) | u, v \in Z, P_u \neq P_v\}$  and define  $R(Z) = \bigcap_{e \in Z'} R(e)$ . The given definition assumes the worst case of  $h = |Z|$ .

<sup>12</sup>or “best” orientation – but in this case best is harder to compute. In fact it reduces to a maximin problem : find  $\phi \in R(C)$  that maximizes

$$\min_{u, v \in C} |d_u(\Gamma_u(\phi_u + \phi)) - d_v(\Gamma_v(\phi_v + \phi))|.$$

However, if we want to allow for maximal error in gripper rotation, we can choose to take the middle orientation in the largest continuous interval of orientations in  $R(C)$ .

Resolving cliques  $C$  with  $R(C) = \emptyset$  is the subject of the rest of this paper. Analogously, *distinguishability between parts*  $P_u, P_v, P_w, \dots$  corresponding to a clique  $C$  of vertices  $u, v, w, \dots$  can be understood as “distinguishability up to empty  $R(C)$ .”

### 2.5.1 An efficient planning algorithm resulting in a short but suboptimal strategy

We first present an planning algorithm that requires an off-line  $O(n^2 \log n)$  preprocessing time and produces a grasp plan requiring no more than  $q'$  grasps, where  $q' \leq n$  is the size of the maximal clique in  $G$  that the part 'hashes' onto after the first grasp.

The off-line planning algorithm computing the grasp plan is presented now. Let two lists  $A_v, D_v$ , both initialized to  $\emptyset$ , be associated with every vertex  $v \in G$ . Let  $Q$  be a queue of cliques initialized to  $\emptyset$ . Operations on  $Q$  are “PUSH”, inserting a clique at the tail of the queue; and “POP”, removing a clique from the head of the queue. Let  $A \bowtie a$  denote that element  $a$  being inserted at the tail of list  $A$ . Let disjoint union of two sets  $T_1, T_2$  be denoted as  $T_1 \uplus T_2$ .

**Lemma 2** *If the vertices  $\{a, b, c\}$  form a 3-clique (possibly non-maximal) with  $R((a, b)) \neq \emptyset$ . Then, for every orientation  $x \in R((a, b))$ ,  $x$  exists in one or both of  $R((a, c)), R((b, c))$ .*

**Proof:** Let diameter value  $D_a = d_a(\Gamma_a(x + \phi_a))$ .  $D_b, D_c$  are similarly defined. Since  $x \in R((a, b))$ ,  $D_a \neq D_b$ . This implies that  $D_c$  is unequal to at least one of  $D_a$  and  $D_b$ , say  $D_c \neq D_a$ . Then,  $x \in R((a, c))$  by definition of  $R((a, c))$ .  $\square$

#### (Off-line) Planning Algorithm SUBOPTIMAL

Let  $G$  be decomposed into  $s$  maximal cliques  $S_1, S_2, \dots, S_s$ .

for  $i = 1$  to  $s$  do:

  if  $(|S_i| \neq 1)$  do:

    PUSH clique  $S_i$  onto the queue  $Q$ .

    while  $(Q \neq \emptyset)$  do:

      1. POP a clique  $T$  from  $Q$ .

      2. Pick arbitrary edge  $e = (u, w)$  in  $T$  satisfying  $R(e) \neq \emptyset$ . If no such  $e$ , go to step 1.

      3. Pick an arbitrary orientation  $x_e \in R(e)$ .

      4. For every vertex  $v \in T$ ,

        4.1 (compute new orientation)  $\phi_v \leftarrow \Gamma_v(\phi_v + x_e)$ ; Let  $d'(v) \leftarrow d_v(\phi_v)$ .

        4.2 (update associated lists)  $A_v \bowtie x_e$ ;  $D_v \bowtie d'(v)$ .

      5. Sort the vertices  $v$  of  $T$  by their  $d'$  value.

      6. Let the partition induced on  $T$  by  $d_v$  be  $T_1 \uplus T_2 \uplus \dots \uplus T_l$ .

      7. PUSH each sub-clique  $T_j, 1 \leq j \leq l$ , with  $|T_j| > 1$ , onto  $Q$ .

$\square$

*Analysis:* Because of the choice of  $x_e$  and Lemma 2, there will be at least two distinct  $d'(v)$  values arising from the vertices of the clique  $T$  (Step 4.1). That is,  $l \geq 2$  (Step 6) and  $|T_j| \geq 1, 1 \leq j \leq l$ . Computing the new orientation (Step 4.1) could be understood as a vertex migration: vertex  $(\phi_v, P_v, d_v, \Gamma_v)$  before Step 4.1 behaves as vertex  $(\Gamma(\phi_v + x_e), P_v, d_v, \Gamma_v)$  after this step. Therefore, after Step 4.1, notice that we could get duplicated vertices, *i.e.* more than one vertex corresponding to the same orientation of the same part. In such a case, we eliminate all but one of these vertices. The queue always consists of mutually disjoint cliques only. In Step 2, if there is no edge  $e$  with  $R(e) \neq \emptyset$ , then it implies that the clique  $T$  is unpartitionable, *i.e.* the set of parts  $\{P_u | u \in T\}$  are pairwise pseudo-twins all at their witnessing orientations. However, Step 2 can be accomplished in linear time because:

**Lemma 3** Let  $v$  be some vertex in a  $p$ -clique  $T$ . If all the  $p - 1$  edges  $e$  from  $T$  incident on  $v$  have  $R(e) = \emptyset$ , then every edge  $f$  in  $T$  has  $R(f) = \emptyset$ .  $\square$

**Proof:**

Suppose there is an edge  $f = (u, w)$  with  $R(f) \neq \emptyset$ . Let  $x \in R(f)$ . Then, by Lemma 2,  $x \in R((u, v))$  or  $x \in R((w, v))$  which contradicts  $R(e) = \emptyset$  for every edge incident on  $v$  in  $T$ .

$\square$

Step 2 can therefore be implemented in linear time. The sorting in Step 5 becomes the dominant operation in each iteration. Hence, the worst case time complexity to empty the queue starting with the  $p$ -clique  $T$  is  $O(p \log p + (p - 1) \log(p - 1) + \dots) = O(p^2 \log p)$  time. Repeating this process for every maximal clique  $S_i$  gives us the overall complexity of  $O(n^2 \log n)$  time for this planning algorithm. Correctness essentially follows from the Lemma 2 that guarantees a partition of  $T$  by grasp action  $x_e$ .

The lists  $A_v, D_v$  associated with a vertex  $v$  of a clique  $C$  give the grasp plan to resolve the clique  $C$ . This on-line grasp plan is now presented.

Let  $A_v = \{x_{1,v}, x_{2,v}, \dots\}$  be the list of angles associated with vertex  $v$ .  $D_v = \{d_{1,v}, d_{2,v}, \dots\}$  is the corresponding list of associated diameter values.

#### (On-line) Grasp Strategy SUBOPTIMAL

INPUT: Planar part  $P$  lying flat on table.

Grasp  $P$  and measure the diameter  $D$ . Hash on to the appropriate (maximal) clique  $C_1$ .

*i.e.*  $C_1 = \{v \in G \mid d_v(\phi_v) = D\}$ ,  $|C_1| = q'$ .

Set  $i \leftarrow 1$ .

**repeat:**

0. Let  $v$  be any vertex in  $C_i$ ,
1. Rotate the gripper by angle  $X = x_{i,v}$ .
2. Squeeze and measure the diameter  $D$ .
3.  $C_{i+1} = \{v \in C_i \mid d_{i,v} = D\}$ .
4.  $i \leftarrow i + 1$ .

**until**  $|C_{i+1}| = |C_i|$ .

$\nu \leftarrow i - 1$ . Let  $C_\nu = \{v_1, \dots, v_z\}$  be of heterogeneity  $w$ .

Let  $\{P_1, P_2, \dots, P_w\}$  be the parts associated with the vertices  $\{v_1, \dots, v_z\}$ .

Declare that the part  $P$  is one of  $P_1, P_2, \dots, P_w$ .

$\square$

*Analysis:* The subclique selection step, Step 3, is the dominant operation in every grasp and can be implemented in  $O(\log n)$  time per grasp. The grasp length of the plan,  $\mu$ , is bounded above by the size of the  $A_v, D_v$  lists, which is at most  $q' = |C_1| \leq q \leq n$ . ( $q$  is the size of the maximum clique in  $G$ ).

### 2.5.2 Optimal grasp strategies

The reason why the Planning Algorithm SUBOPTIMAL of section 2.5.1 does not produce the shortest grasp plan is that the partitions  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  (Step 6) were chosen on basis of an arbitrary cutting edge  $e$  and arbitrary cutting orientation  $x_e$  (Steps 2,3).

Let  $E(T_1, T_2, \dots, T_l)$  denote the set of edges with one end point in one of the  $T_i$  and the other end-point in one of the other  $T_j$ . That is,

$$E(T_1, T_2, \dots, T_l) = \{e = (u, v) \mid u \in T_i, v \in T_j, 1 \leq i, j \leq l, i \neq j\}.$$

To get the optimal number of grasps, one approach is to consider every partition of every subclique. However, even the number of 2-partitions of an  $i$ -clique is very large ( $\Theta(2^i)$ ). Therefore this naive approach would lead to an algorithm that requires at least

$$\sum_{i=1}^{q-1} \binom{q}{i} \Omega(2^i) = \Omega(3^q)$$

operations. However, notice that we only need to consider “good” partitions  $T_1 \uplus T_2 \uplus \dots \uplus T_l$ , i.e. those satisfying  $R(E(T_1, T_2, \dots, T_l)) \neq \emptyset$ . The following lemma gives a polynomial bound on the number of good partitions.

**Lemma 4** *There are only  $O(n^3)$  good partitions  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  of any maximal clique  $T$  in  $G$ . Furthermore, the only good partitions  $T'_1 \uplus T'_2 \uplus \dots \uplus T'_l$  of a subclique  $T' \subset T$  are the aforementioned  $O(n^3)$  partitions of  $T$  restricted to the vertices of  $T'$ . Finally, these  $O(n^3)$  partitions of maximal clique  $T$  can be computed in  $O(n^4 \log n)$  time.*

**Proof:** Lemma 1 implies that every  $R(e)$  has a complexity of  $O(n)$ . This implies that there exists a constant  $c$  so that each  $R(e)$  has no more than  $cn$  intervals<sup>13</sup> over  $[0, \pi)$ . Consider the set of left end points and right end points of all the  $R(e)$  as a sorted (in increasing order) set of points  $A = \{\alpha_1, \alpha_2, \dots, \alpha_t\}$ . Note that  $t$  is at most  $cn \frac{n(n-1)}{2} = O(n^3)$  because there are  $\frac{n(n-1)}{2}$  edges  $e$ . Now consider an open interval  $A_i = (\alpha_i, \alpha_{i+1})$ . It is easy to see that, for every edge  $e$ , either  $A_i \subset R(e)$  or  $A_i \cap R(e) = \emptyset$ .

By “a (good) partition of  $T$  generated by  $x$ ”, let us understand the following. Let  $\{D_1, \dots, D_l\}$  be the set of diameter values (ignoring duplicates) in  $\{d_v(\Gamma_v(x + \phi_v)) \mid v \in T\}$ . Then, the partition of  $T$  generated by  $x$  is  $T_1 \uplus T_2 \uplus \dots \uplus T_l$ , where for  $1 \leq i \leq l$ ,  $T_i = \{v \in T \mid d_v(\Gamma_v(x + \phi_v)) = D_i\}$ . The major claim is that all good partitions  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  in  $T$  are included among those generated by some  $x \in A$ . To verify this claim observe that the end-points of every interval of  $R(E(T_1, T_2, \dots, T_l))$  are present in  $A$  and also that *any* point<sup>14</sup> in  $R(E(T_1, T_2, \dots, T_l))$  suffices to generate the partition  $T_1 \uplus T_2 \uplus \dots \uplus T_l$ . This proves the first statement of the lemma. The second statement follows from the simple fact that  $R(E(T'_1, T'_2, \dots, T'_l))$ , for the partition  $T'_1 \uplus T'_2 \uplus \dots \uplus T'_l$  of any sub-clique  $T'$ , also has all its interval end-points in  $A$  (due to simple inclusion).

This gives us a simple procedure to list out all the  $n^3$  partitions of  $G$ , namely, form the set  $A = \{\alpha_1, \alpha_2, \dots, \alpha_t\}$  (can be done in  $O(n^3 \log n)$  time by sorting once the  $R(e)$  have been compute) and determine the partitions generated by every  $\alpha_i$ ,  $1 \leq i \leq t$ . Each such partition would require  $O(n)$  time to compute, and  $O(n \log n)$  time to sort by  $D_i$  value. Therefore, all the good partitions can be computed in  $O(n^4 \log n)$  time.  $\square$

**Corollary 1** *A partition  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  of  $T$  is good if and only if it is generated by some  $x \in A$ , where  $A$  and “generation of a partition” are as defined in the proof of Lemma 4.  $\square$*

<sup>13</sup>We may assume all closed intervals.

<sup>14</sup>including the end-points

Let us assume that all the (at most  $t = O(n^3)$ ) good partitions of the cliques in  $G$  are stored in a table as shown in Fig. 6. The columns correspond to the  $n$  vertices of  $G$ . The rows give the various good partitions. For considering the good partitions of a subset of vertices, simply restrict your vision to the columns corresponding to those vertices (second statement of Lemma 4). Each good partition also stores its generator alongside. For example,  $x_1$  generates the good partition  $\{v_1, v_2, v_5\} \uplus \{v_3\} \uplus \{v_4\}$  for the clique  $\{v_1, v_2, v_3, v_4, v_5\}$ . More than one generator might exist for the same partition: notice that  $x_4$  also generates the same partition as  $x_1$  for  $\{v_1, v_2, v_3, v_4, v_5\}$ . A partition which was good for a clique may not be good for its subcliques:  $x_1$  and  $x_4$  do not generate a partition for  $\{v_1, v_2, v_5\}$ .

Our approach towards the optimal algorithm is to consider every clique in  $G$ , beginning with 1-cliques, then 2-cliques, working upwards until the maximal clique. When considering an  $i$ -clique  $T$  the aim is to compute its optimal grasp plan. This is done by considering every good partition  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  of  $T$ . The optimal grasp strategies for each of  $T_1, T_2, \dots, T_l$  have already been computed by this time because we have worked bottom-up. Let  $h(T_i)$  be the grasp length of the optimal grasp plan for  $T_i$ . Define the *grasp length of the good partition*  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  to be

$$h(T_1, T_2, \dots, T_l) = \max_{i=1}^l (h(T_i)).$$

Define the *best partition* of  $T$  to be the good partition of shortest grasp length. The optimal grasp plan for  $T$  can be computed from the optimal grasp plans of  $T_1, T_2, \dots, T_l$ , where  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  is the best partition for  $T$ .  $h(T)$  will be 1 plus the grasp length of its best partition.

Here, we store optimal grasp plans as a rooted tree, each node of the tree alongside a subclique (rather than lists alongside vertices as done previously). The root,  $\rho(T)$ , of the subtree storing the so-far-optimal grasp strategy for clique  $T$  is labelled by the vertices of  $T$ . The root is also associated with some angle  $x$  which is the grasp angle that generates the so-far-best partition of  $T$ . Let  $\{D_1, \dots, D_l\}$  be the set of diameter values (ignoring duplicates) in  $\{d_v(\Gamma_v(x + \phi_v)) \mid v \in T\}$ . Let  $\{T_j\}, 1 \leq j \leq l$  be the partition of  $T$  induced by the diameter values. The edges out of the root are labelled by the diameter values  $D_1, \dots, D_l$  and the children of the root are labelled by the vertices of the individual  $T_j$  and will be associated by an appropriate grasp angle.<sup>15</sup> The leaves of this tree consist of vertices that all belong to the same part or a clique of vertices such that every edge  $e$  between two vertices associated with distinct parts has  $R(e) = \emptyset$ . Let the subtree whose root  $\rho(T)$  is labelled by the vertices of clique  $T$  be denoted as  $\text{TREE}(T)$ . Additionally, let  $h(T)$  denote the depth of  $\text{TREE}(T)$ . It is easy to see that, at the end of the computation, this definition of  $h(T)$  and that given in the previous paragraph are the same. Initially, the tree for every clique (including the non-maximal ones) consists of a single unconnected node labelled by the vertices of that clique.

Let  $N_i$  be the number of  $i$ -cliques in  $G$ .  $q$  is size of maximum clique in  $G$ .

---

<sup>15</sup>Each internal node is just like for the root – it stores the grasp strategy for the  $T_j$  it is labelled by.

## Planning Algorithm OPT\_GRASP

```

for  $i = 1$  to  $q$  do:
  for  $j = 1$  to  $N_i$  do:
    Let  $C_{j,i}$  be the  $j$ th  $i$ -clique in  $G$ .
    if  $i = 1$  or  $2$ 
      Do the needful [Trivial Cases]
    else do: [ $i > 2$ ]
      1.  $w \leftarrow |C_{j,i}|$ ;  $h(C_{j,i}) \leftarrow w$ .
      2. Consider all good partitions of  $C_{j,i}$  in turn.
        2.1 Let  $T_1 \uplus T_2 \uplus \dots \uplus T_l$  be one such good partition.
        2.2 Let  $x \in R(E(T_1, T_2, \dots, T_l))$ .
        2.3 if  $(1 + h(T_1, T_2, \dots, T_l) < w)$  do:
          2.3.1  $w \leftarrow 1 + h(T_1, T_2, \dots, T_l)$ ;  $h(C_{j,i}) \leftarrow w$ .
          2.3.2 Let  $\{D_1, \dots, D_l\} \leftarrow \{d_v(\Gamma_v(x + \phi_v)) \mid v \in C_{j,i}\}$ 
          2.3.3 For  $1 \leq r \leq l$ ,  $T_r = \{v \in C_{j,i} \mid d_v(\Gamma_v(x + \phi_v)) = D_r\}$ .
          2.3.4 Remove all edges out the tree node  $X(C_{j,i})$ .
          2.3.5 Associate node  $X(C_{j,i})$  by grasp angle  $x$ .
          2.3.6 Create  $l$  edges out of node  $X(C_{j,i})$  with labels  $D_1, \dots, D_l$ .
          2.3.7 Let the  $r$ th edge,  $1 \leq r \leq l$ , point to  $X(T_r)$ .

```

□

*Analysis:* Assume that at any stage of the computation, given a clique  $T$ , the information associated with it, namely,  $h(T)$  and a pointer to  $\rho(T)$ , can be obtained by random access in  $O(1)$  time. Also assume that we have precomputed the good partitions of every maximal clique and stored them in tabular form (as described in the proof Lemma 4 and Fig. 6). Steps 2 and 2.1 can be implemented by loading in the next row (partition) from the table. The test for “goodness” of this partition with respect to  $C_{j,i}$  consists of checking whether the partition loaded in has at least two distinct labels for the columns corresponding to vertices in  $C_{j,i}$ . This check can be done in  $O(i)$  time. Assume that the partition also brings with it its generator which may be used as  $x$  in Step 2.2 (without computation of  $E(T_1, \dots, T_l)$  and  $R(E(T_1, \dots, T_l))$ ). At this time subcliques  $T_1, \dots, T_l$  have been processed and so  $h(T_1), \dots, h(T_l)$  may be accessed and their maximum,  $h(T_1, \dots, T_l)$  computed in  $O(l) = O(i)$  time for Step 2.3. A single iteration of steps 2.3.1–2.3.7 can be implemented in  $O(l) = O(i)$  time. Let  $T(p)$  represent the preprocessing time complexity to resolve a  $p$ -clique. Since there are at most  $O(n^3)$  good partitions (Lemma 4), it is easy to see that

$$T(p) = \sum_{i=1}^{p-1} \binom{p}{i} O(in^3). \quad (1)$$

This gives  $T(p) = O(n^4 2^p)$ . Total preprocessing complexity is the sum of complexities for each maximal clique, which is upper bounded by  $\frac{n}{q} n^4 2^q$ , where  $q$  is the size of the maximum clique in  $G$ . The on-line grasp plan still runs in  $O(\log n)$  per grasp. However, because to the representation of grasps directly as trees, its implementation is straightforward and is not described.

This exponential time complexity isn’t as bad as it sounds since the size of the largest clique in the graphs we consider is likely to be small even if we have large number of parts. The correctness follows from the discussion before presentation of the planning algorithm and the fact that, at the end of the processing for  $C_{j,i}$ , the best partition for it would have been determined correctly (Step 2.3).

Partition Generators	Vertices								
	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	•	•	$v_n$
$x_1$	2	2	1	3	2	3			4
$x_2$	2	3	5	4	5	7			1
$x_3$	1	1	1	1	2	2			2
$x_4$	1	1	4	5	1	1			2
•									
•									
•									
$x_t$	1	5	a	5	a	a			5

$t = O(n^3)$

Figure 6: Storing good partitions for a graph in tabular form. The columns are the vertices of the graph and the rows are the good partitions. In considering good partitions for a particular clique of vertices  $C$ , look only at the columns under vertices in  $C$ . Partitions are specified by attaching labels to vertices. In a particular partition, for a particular set of vertices, two vertices belong to the same partition if and only if they have the same label associated with them in that partition (the actual label itself is not important).  $\{v_1, v_2\} \uplus \{v_3\}$  is a good partition for  $\{v_1, v_2, v_3\}$  generated by  $x_1$ .  $x_1$  also generates the good partition  $\{v_1, v_2, v_5\} \uplus \{v_3\} \uplus \{v_4\}$  for  $\{v_1, \dots, v_5\}$ .  $x_1$  does not generate a good partition for  $\{v_1, v_2\}$ . (Why? Because  $x_1 \notin R((v_1, v_2))$ .) The number of good partitions (number of rows),  $t$ , is  $O(n^3)$ .

### 2.5.3 An example

Consider Fig. 7. Three polygonal parts  $P_1, P_2, P_3$  are shown shaded and in zero orientation.  $P_1$  is an equilateral triangle with altitude  $d_1$ ;  $P_2$ , a house shaped part with two stable orientations, one with diameter  $d_1$  and another with a  $d_2 > d_1 + \epsilon$ ; and  $P_3$  is a square of dimension  $d_2$ .  $N$ , the total number of faces of all the parts is 12. The graph of stable diameters for this set of parts,  $G$ , has  $n = 7$  vertices,  $v_1, v_2, \dots, v_7$ .  $G$  is shown just below  $P_1$  in the figure. Vertices of  $G$  are indicated by solid circles: those associated with the stable orientations of  $P_1$  with the smallest circles,  $P_2$  slightly larger, and  $P_3$  the largest. Each vertex  $v_i$  is shown associated with the following information: the part  $P_i$ , orientation  $\phi_i$  and the diameter of  $P_i$  at orientation (in degrees)  $\phi_i$ .

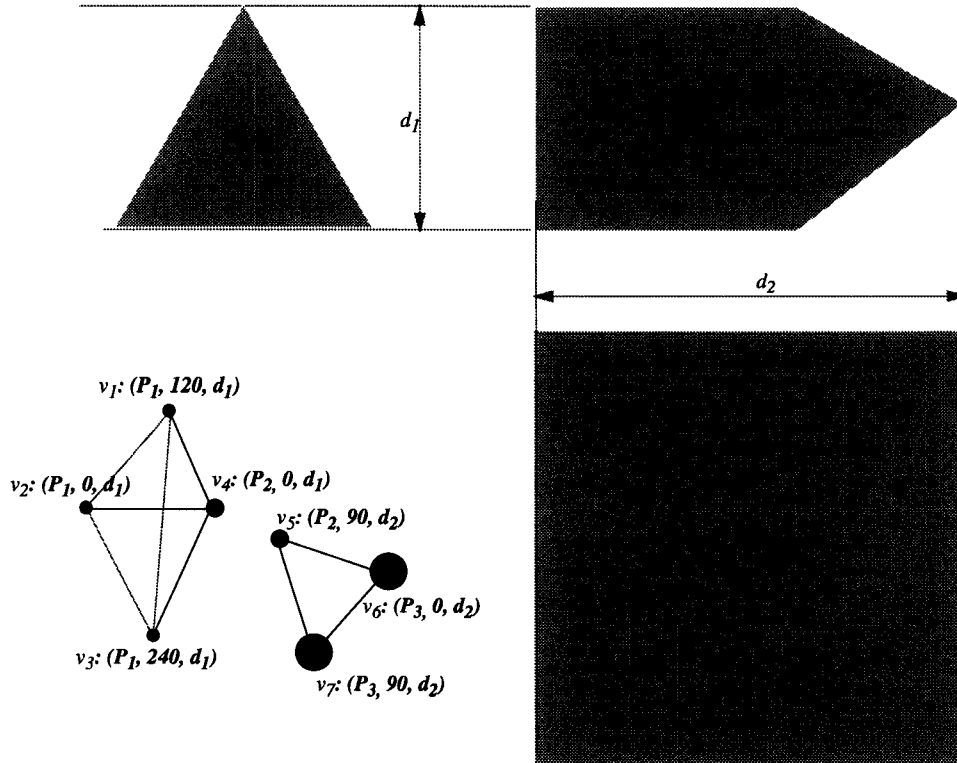


Figure 7: The graph of stable diameters for the three parts  $P_1, P_2, P_3$ .

Recall that edges in  $G$  are between two stable orientations having the same diameter value. Thus  $G$  can be decomposed into two maximal cliques:  $C_1 = \{v_1, v_2, v_3, v_4\}$  and  $C_2 = \{v_5, v_6, v_7\}$ . For any edge  $e$  in  $\{v_1, v_2, v_3\}$ ,  $R(e) = \emptyset$ . Same is true for any edge in  $\{v_6, v_7\}$ . The  $R(e)$  for other edges are not shown individually but it is easy to note that each of these  $R(e)$  is non-null and includes the orientation<sup>16</sup> 90 degrees.

The graph for resolving the cliques in  $G$  is shown in Fig. 8. The nodes of the graph strategy tree are indicated as little square boxes such as  $\square$ . There are 7 nodes in all forming a complete binary tree of depth 2. Each node is associated with a subgraph (shown enclosed in a rectangular box) of  $G$ , children associated with disjoint portions of the parent and the root associated with all of  $G$ . Every internal node is also associated with a orientation angle (shown enclosed in parenthesis)

<sup>16</sup>Possibly not the best orientation : "best" meaning that allowing maximal sensor or rotation error.

which indicates the angle the gripper must be rotated before the next grasp. For the root this can be any angle, wlog 0, since the first grasp is random. The possible choices for diameter measurement from the first grasp is  $d_1$  or  $d_2$ . The two edges out of the root are therefore labelled  $d_1$  and  $d_2$ , respectively. If in the first grasp we measure  $d_1$ , we travel down the left edge to a node associated with clique  $C_1$  (and analogously if we measure  $d_2$ ). The node associated with  $C_1$  discovers that  $R(\{v_1, v_2, v_3\}) = \emptyset$  and therefore computes the partition  $\{v_1, v_2, v_3\} \uplus \{v_4\}$  generated by orientation  $x = 90$ . So we rotate the gripper by 90 degrees and grasp again. If measured diameter is still  $d_1$ , we have part 1, otherwise part 2. Likewise along other paths down the tree.

So, in this case of the three polygons, the complete strategy is as follows. Grasp, rotate by 90 and grasp again. If diameter values measured are  $(d_1, d_1)$ , then we have part 1, if  $(d_2, d_2)$  its part 3, and part 2 otherwise. This was also shown in Fig. 1 earlier.

### 3 Future work

We would like to address the following issues in the future.

**Length of grasp strategy** The first planning algorithm we presented for recognizing polygonal parts returned a grasp plan with length no worse than  $q'$ ,  $q'$  being the size of the maximum clique in  $G$ . However, we believe that the average length of this grasp plan to be much better than  $q'$ , perhaps closer to  $\log q'$ . The reason is that we believe the partition of  $S_i$  generated by a arbitrary  $x$  from a arbitrary non-empty  $R(e)$  (Steps 2,3 in Planning Algorithm SUBOPTIMAL) to be fairly even (*i.e.* no set of the partition having more than  $c|S_i|$  vertices, for some fixed  $c < 1$ ). Thus, for practical implementations, perhaps this planning algorithm is more suitable. We would like to compare the length of the grasp plan computed by the two planning algorithms. Application of randomized constructions in geometry [8, 3] would be useful in proving good expected behavior.

**Identifying  $\sigma$ -equivalent parts** Our planning algorithms can identify any part (and its final orientation up to symmetry) from among a set of parts as long as no two are  $\sigma$ -equivalent. Testing for  $\sigma$ -equivalence of parts is straightforward from the graph of stable diameters  $G$  (Section 2.4). However, can we distinguish between  $\sigma$ -equivalent parts  $P_u, P_v$  (that are not  $d$ -equivalent)? While the chance that two industrial parts are  $\sigma$ -equivalent is very remote, this is nevertheless a theoretically interesting problem.<sup>17</sup> Let  $\phi_u, \phi_v$  be the orientations that witness the  $\sigma$ -equivalence For the edge  $e = (u, v)$  in  $G$ ,  $R(e) = \emptyset$ . Therefore, consider the set  $R_2(e) \subset S^1 \times S^1$  defined by

$$R_2(e) = \{(\phi_1, \phi_2) \mid d_u(\Gamma_u(\Gamma_u(\phi_1 + \phi_u) + \phi_2)) \neq d_v(\Gamma_v(\Gamma_v(\phi_1 + \phi_v) + \phi_2))\}.$$

If  $R_2(e)$  is non-empty, then a 2-tuple of orientations from  $R_2(e)$  can be applied to disambiguate between the  $\sigma$ -equivalent parts. In fact, we can run the planning algorithms substituting  $R_2(e)$  for  $R(e)$ .  $R_2(e)$  can be computed in  $O(|\Gamma_u||\Gamma_v|)$  time. If  $R_2(e)$  is also empty, then  $R_3(e) \subset (S^1)^3$  needs to be computed. Given the  $R_2(e)$  for all edges,  $R_3(e)$  can be computed again in  $O(n^2)$  time and so on. Surely, for  $d$ -equivalent parts, all the  $R_i(e)$  will be empty (a trivial bound on  $i$  is  $n^2$ ) and no grasp plan can possibly distinguish between them. However, we conjecture that for  $\sigma$ -equivalent parts that are not  $d$ -equivalent, there exists a constant  $c$  such that  $R_c(e) \neq \emptyset$ . If true, this would result in an  $O(n^2 \log n)$  planning algorithm, with  $O(n^4)$  preprocessing, that gives a plan of length at most  $cn$  to distinguish between any set of parts as long as no two of them are  $d$ -equivalent.

<sup>17</sup>Our algorithms in this paper can distinguish between  $\sigma$ -equivalent parts as long as the parts do not enter into orientations that witness their  $\sigma$ -equivalence.

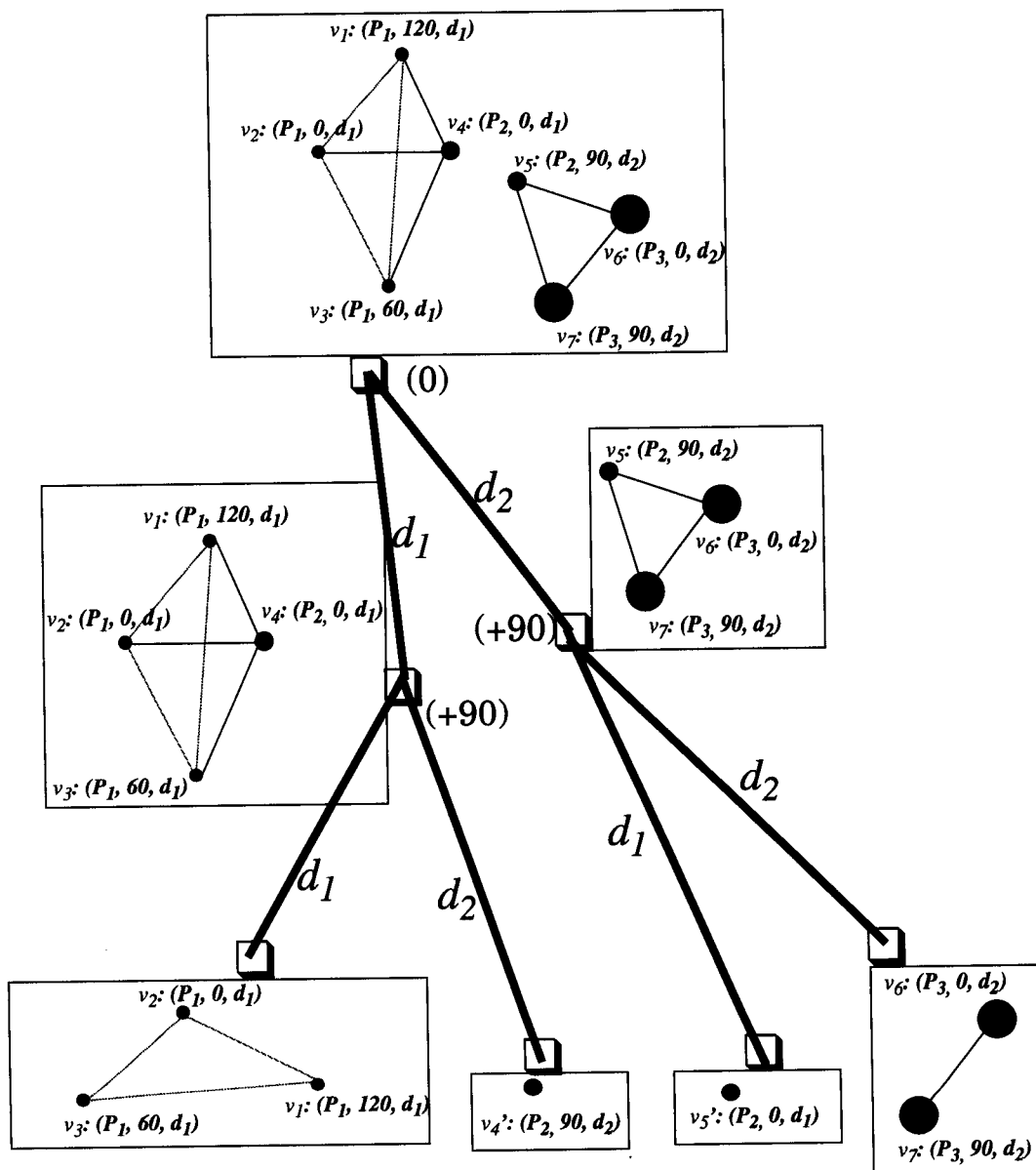


Figure 8: Grasp plan to resolve cliques in the graph of stable diameters for the parts  $P_1, P_2, P_3$  shown in the last figure.

## 4 Conclusion

We have considered the planar problem of determining the convex shape of a polygonal part from a sequence of measurements taken with a frictionless parallel-jaw gripper. We have derived both negative [25] and positive results.

The negative results are related to the study of curves of constant width (*Gleichdicke*) in classical geometry, establishing a new link between computational geometry and robotics. The positive results are examples of sensor-based manipulation planning, but differ from previous work in that we use a frictionless gripper and give a computational analysis of the planning problem. The required hardware is inexpensive and well-known. This is consistent with our aim to develop new software for “old” hardware.

**Acknowledgments** We thank Randy Brost, Duk Kang, Mike Erdmann, Doug Ierardi, Matt Mason, Babu Narayanan, Mark Overmars, Viktor Prasanna, Govindan Rajeev, Ari Requicha, Günter Rote, Russ Taylor, and Chee Yap for rewarding discussions on this subject. We also thank the anonymous referees for their constructive criticism and useful suggestions.

## References

- [1] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8), 1988.
- [2] P. Belleville. Probing polygons minimally is hard (extended abstract). In *2nd Canadian Conference on Computational Geometry*, pages 50–53, 1990.
- [3] J. D. Boissonnat, D. Devillers, R. Schott, M. Teillaud, and M. Yvinec. Application of random sampling to on-line algorithms in computational geometry. *Discrete and Computational Geometry*, 8:51–71, 1992.
- [4] J. D. Boissonnat and M. Yvinec. Probing a scene of non-convex polyhedra. *Algorithmica*, 8:321–342, 1992.
- [5] R. C. Brost. Automatic grasp planning in the presence of uncertainty. *The International Journal of Robotics Research*, 8(1), February 1988.
- [6] John Canny and Ken Goldberg. A RISC paradigm for industrial robotics. Technical Report ESRC 93-4/RAMP 93-2, University of California at Berkeley, Engineering Systems Research Center, February 1993.
- [7] Y-B. Chen and D. J. Ierardi. Distinguishing polygons by sensing diameters. Technical Report USC-CS-92-503, University of Southern California, Dept. of Computer Science, December 1991.
- [8] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete and Computational Geometry*, 4:387–421, 1989.
- [9] R. Cole and C. K. Yap. Shape from probing. *Journal of Algorithms*, 8(1):19–38, 1987.
- [10] D. P. Dobkin, H. Edelsbrunner, and C. K. Yap. Probing convex polytopes. In *Symposium on Theory of Computing (STOC)*, pages 424–432. ACM, 1986.
- [11] R. Ellis. Acquiring tactile data for the recognition of planar objects. In *IEEE International Conference on Robotics and Automation*, 1987.

- [12] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. In *IEEE International Conference on Robotics and Automation*, 1986.
- [13] K. Y. Goldberg. *Stochastic plans for robotic manipulation*. PhD thesis, Carnegie-Mellon University, School of Computer Science, August 1990.
- [14] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, August 1993. (Special issue on Computational Robotics).
- [15] K. Y. Goldberg and M. Furst. Low friction gripper. U.S. Patent # 5,098,145, March 1992.
- [16] D. Kang and K. Y. Goldberg. Shape recognition by random grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, July 1992.
- [17] D. Kolzow, A. Kuba, and A. Volcic. An algorithm for reconstructing convex bodies from their projections. *Discrete and Computational Geometry*, 4:205–237, 1989.
- [18] A. Koutsou. Object exploration using a parallel-jaw gripper. GRASP Lab Tech Report MS-CIS-88-48, University of Pennsylvania, 1988.
- [19] S-Y. R. Li. Reconstruction of polygons from projections. *Information Processing Letters*, 28:235–240, 12 Aug. 1988.
- [20] 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.
- [21] 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.
- [22] M. A. Peshkin. *Planning Robotic Manipulation Strategies for Sliding Objects*. PhD thesis, Carnegie-Mellon University, Department of Physics, Pittsburgh, Pennsylvania, Nov 1986. Also published as a book: *Robotic Manipulation Strategies*, Prentice Hall, 1990, New Jersey.
- [23] A. S. Rao and K. Y. Goldberg. Grasping planar curved parts with a parallel-jaw gripper. Technical Report # 299, University of Southern California, Institute of Robotics and Intelligent Systems (IRIS), Los Angeles, Calif. 90089-0273, August 1992. Submitted to the *IEEE Transactions on Robotics and Automation*.
- [24] A. S. Rao and K. Y. Goldberg. Orienting generalized polygons. In *International conference on Robotics and Automation (ICRA)*, Nice, France, May 1992. IEEE.
- [25] A. S. Rao and K. Y. Goldberg. Shape from diameter: Negative results. Technical Report RUU-CS-93-16, Utrecht University, Department of Computer Science, Padualaan 14, Postbus 80.089, 3508 TB Utrecht, 1993.
- [26] A. S. Rao and K. Y. Goldberg. Shape from diameter: recognizing polygonal parts with a parallel-jaw gripper. *International Journal of Robotics Research*, 1993. Accepted for publication.
- [27] D. Raviv. A quantitative approach to camera fixation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 386–392, Maui, Hawaii, June 1991. IEEE.

- [28] E. Rich. *Artificial Intelligence*. series in AI. McGraw Hill, New York, 1983.
- [29] S. S. Skiena. *Geometric probing*. PhD thesis, University of Illinois, Dept. of Computer Science, Urbana, Ill., 1988.
- [30] S. S. Skiena. Problems in geometric probing. *Algorithmica*, pages 599–605, 1989.
- [31] A. J. Spyridi and A. A. G. Requicha. Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines. In *International Conference on Robotics and Automation*. IEEE, May 1990.
- [32] 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.
- [33] A. S. Wallack and J. F. Canny. Linear time algorithm for object localization using scanning. manuscript, September 1991.
- [34] A. S. Wallack and J. F. Canny. Object localization using finger gap sensing. Technical Report ESRC 92-3/RAMP 92-2, Univ. of California, Berkeley, Dept. of Computer Science, February 1992.
- [35] A. S. Wallack and J. F. Canny. Object localization using light beam sensing. (submitted), March 1992.