Grasping and Fixturing: a Geometric Study and an Implementation

by

Marek Teichmann

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science
New York University
September, 1995

Approved: ______________________________

Professor Bhubaneswar Mishra
Research Advisor

To Quyen and my parents

Acknowledgments

I would like to extend my sincere thanks to my advisor Professor Bhubaneswar Mishra, for his encouragement and guidance throughout my stay at NYU. He was always available for a quick question or a lengthy discussion, and had a seemingly inexhaustible supply of interesting open problems to work on. I would like to thank Professor Chee-Keng Yap for the many interesting discussions we had, ranging from continued fractions to randomized algorithm, and for reading this thesis.

I also wish to thank Professor Richard Pollack for reading this work, and along with Professor Janos Pach for transferring some of their enthusiasm for Discrete and Computational Geometry through their many courses on these topics. They too always found time for my questions.

My thanks go to Fred Hansen for his advice and assistance with the implementation, and Louis Pavlakos for his help with MOSAIC.

Thanks to Professor Boris Aronov, Joonsoo Choi, S. Muthukrishnan, Jürgen Sellen for the many fun discussions we had.

I would like to thank my parents for being there for me, even when I was mostly absent, particularly toward the end of this work.

Finally, I shall always be in debt to my wife Quyen for her constant understanding, support and love.

I also gratefully acknowledge support by the National Science Foundation under Grants CDA-9018673, IRI-9003986, IRI-9414862, by an NYU Technology Transfer Grant, and by postgraduate awards from NSERC, Canada and FCAR, Québec, Canada.

ABSTRACT

Grasping and Fixturing: a Geometric Study and an Implementation

Marek Teichmann

Research Advisor: Professor Bhubaneswar Mishra

The problem of immobilizing an object by placing "fingers" (or points) on its boundary occurs in the field of dexterous manipulation, manufacturing and geometry. In this dissertation, we consider the purely static problems of good grasp and fixture set synthesis, and explore their connection to problems in computational and combinatorial geometry. Two efficient randomized approximation algorithms are proposed for finding the smallest cover for a given convex set and for finding the largest magnitude by which a convex set can be scaled and still be covered by a cover of a given size. They generalize an algorithm by Clarkson [Cla93]. The cover points are selected from a set of n points. The following bounds are valid for both types of problems. For the former, c is the size of the optimal cover, and for the latter, c is the desired cover size. In both cases, a cover of size $4cd \lg c$ is returned. The running time depends on the set to be covered: covering an n-vertex polytope in $\mathbb{R}^d$ takes $O(c^2 n \log n \log c)$ expected time, and covering a ball takes $O((nc)^{1+\delta} + c^{\lfloor d/2 \rfloor + 1} \log n \log^{\lfloor d/2 \rfloor} c)$ expected time. These algorithms have applications to finding a good grasp or fixture set. An $O(n^2 \log n)$ algorithm for finding optimal 3 finger grasps for n sided polygons is also given. We introduce a new grasp efficiency measure based on a certain class of ellipsoids, invariant under rigid motions of the object coordinate system. To our knowledge, this is the first measure having this property. We also introduce a new *reactive* grasping paradigm which does not require a priori knowledge of the object. This paradigm leads to several reactive algorithms for finding a grasp for parallel jaw grippers and three finger robot hands equipped with simple sensors. We

show their correctness and discuss our implementation of one such algorithm: a parallel jaw gripper with light-beam sensors which we have built.

Contents

List of Figures

Chapter 1

Immobility

If you think you can grasp me, think again:
My story flows in more than one direction
A delta springing from the river bed
With its five fingers spread.

—*Adrienne Rich, 1989*

1.1 Introduction

The problem of immobilizing an object occurs in the field of dexterous manipulation, manufacturing and geometry. Various models of immobilization exist and are closely related. In this dissertation, we consider the purely static problems of good grasp and fixture set synthesis for polyhedral objects, and the corresponding geometric problems. An efficient randomized approximation algorithm for convex set covering is given with applications to finding a good grasp or fixture set. We introduce a new grasp efficiency measure, invariant under rigid motions of the object coordinate system. To our knowledge, this is the first measure having this property. We also introduce a new *reactive* grasping paradigm, show its correctness and discuss our implementation: a parallel jaw gripper with light-beam sensors which we have built.

Immobilizing an object consists of placing a set of "fingers" whose contact with the object prevents its motion. Such problems arise in manufacturing in pick–and–place tasks where an object is acquired by a robot hand and placed in a desired location.

Another application is fixturing, where fixture elements are used to hold a part in place for machining.

The area of dexterous manipulation studies the problems of grasping, fine manipulation and control with multi-fingered robot hands. Several hands have been developed, including Utah/MIT Dextrous Hand [JWKB84], and the hands by Asada [Asa79], Salisbury *et al.* [SR81], Okada [Oka82a], and Tomovic *et al.* [TB62]. These hands usually consist of a small number (four or five) of articulated fingers, possibly with a palmar surface. Each finger is usually an open kinematic chain. The fingers consist of a set of phalanges (i.e. links) usually with active joints and force-sensors; in addition, the fingertip and the surface of the distal phalanx may contain touch- and slippage-sensors. A variety of finger arrangements have been suggested, the more common being an anthropomorphic arrangement, which consists of an opposable 'thumb' and the others arranged to work cooperatively. But other end-effector constructions are possible, and it may be argued that some, like the parallel jaw gripper are quite sufficient for many tasks, particularly pick and place operations [GF93]. See also the recent work of Zhuang, Goldberg and Wong [ZGW94].

In manufacturing, fixtures and other workholding devices need to be placed so as to secure the stock to the tool base and withstand high cutting loads while machining takes place [HW90, Mis91, BG94b, ORSW95]. For example, *modular fixtures* are increasingly being used in flexible manufacturing and job shop machining [ZGW94].

Similar problems can be posed in a geometric setting. A set of points is placed in $\mathbb{R}^d$ and the points are forbidden to penetrate the interior of the object. Questions such as the following arise: how many points are needed and how should they be placed in order to immobilize the object. The techniques used to solve these problems bear much similarity to those used in grasping theory.

In the first part of this dissertation, we exploit the connection between grasp analysis and some theorems in Discrete and Computational Geometry first established by [MSS87]. See also [MNP90]. Such theorems include Steinitz' theorem [Ste16], and its quantitative version which is used for grasp quality measurement [BKP82, KMY92]. We examine the extension of this theory to multiple objects in contact for the analysis and quality evaluation of fixture placement. This extension was first introduced by Baraff, Mattikalli and Khosla [BMK94] who provide heuristics for minimizing the number of fixture elements needed. In addition, we examine the effects of friction, and show that the model remains valid in this case.

In the remainder of this chapter we examine these notions in detail. We show new convexity properties of the *wrench map* which is useful for computational purposes. We also establish a connection between the problem of testing whether an object is immobilized by a set of fingers and the problem of testing rigidity of bar and joint frameworks, and discuss various types of immobility in those two areas. We also analyze the effect of friction on the wrench map and show that the notion remains applicable. Finally, we briefly describe two approaches to finding grasps which do not require knowledge of the object to be grasped, including our new *reactive* approach.

In chapter 2 we examine existing grasp efficiency measures and present a new measure that is invariant under rigid motions of the coordinate system in which the object is defined. This answers an open problem posed in [KMY92, FC92]. This measure is based on a certain class of ellipsoids. We also present methods for its computation. Finally, we examine the effect of friction on grasp strength and correct a misconception which has appeared in the literature.

In chapter 3 we describe a new algorithm for finding three finger optimal grasps of an n sided polygon in time $O(n^2 \log n)$ and some approximation algorithms. In chapter 4 we provide randomized approximation algorithms for finding close to optimal grasps and fixture placements and minimizing either the number of 'fingers' or maximizing the quality of the grasp — two contradictory goals. These problems translate to purely geometric *covering* problems in three and six dimensions, but we study them in arbitrary dimension. These problems can be formally described as follows.

Let $L \subset \mathbb{R}^d$ be a convex set, and for $r > 0$, let rL represent the set $\{rx : x \in L\}$. For a set $U \subset \mathbb{R}^d$, denote the largest scaling factor r of L such that $rL \subseteq \operatorname{conv} U$ by $\rho(U)$. When L is a ball in $\mathbb{R}^d$, call $\rho(U)$ the *residual radius* of U. Also let a set of points C be a *cover* for a set L, if $L \subset \operatorname{conv} C$.

[**MinCover-L**]: Given $\rho_0 > 0$, we wish to find the smallest cover C for $\rho_0 L$

that is, a set $C \subset U$ of size c^* with $\rho(C) \geq \rho_0$.

Since this problem is difficult [MSS87, BMK94], we will solve a corresponding approximation problem: that of finding a cover of size $c^* d \log c^*$. We will call this a $d \log c^*$-approximation of the optimal cover. The companion problem is:

> [**MaxScale-L**]: Given an integer c, find the set $C \subset U$ of size c which maximizes $\rho(C)$. Let $\rho^*(c)$ denote this maximum.

In the approximation version, we ask for a $d \log c^*$-approximation of the best cover C. In this thesis, **MinCover-L** and **MaxScale-L** will refer to the corresponding approximate versions. We shall provide randomized algorithms for these approximation problems that generalize a result by Clarkson [Cla93] for polytope covering. The results hold for any convex set L for which there is a *strong violation* oracle [GLS88], also called *half-space emptiness query* [Mat92]. In our application we will replace L by various convex sets, in particular the d-dimensional ball and a polytope. Let $\gamma = 1/\lfloor d/2 \rfloor$ and δ be any positive constant. For both types of problems, when L is a polytope, we get roughly the same expected time bounds as Clarkson, and for a ball, the expected running time is $O(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + c \log(n/c)(c \log c)^{\lfloor d/2 \rfloor})$ for fixed d. We also give bounds when d is not constant. In our application $d = 6$ and this translates to $O(c^2 n \log n \log c)$ expected time when L is a n-vertex polytope, or $O((nc)^{1+\delta} + c^4 \log n \log^3 c)$ when L is a ball.

Upto now we have assumed that the shape of the object to be grasped is known in advance. This is not always easy to achieve. In chapter 5, we introduce a set of *reactive* algorithms for finding a grasp. The algorithms use simple sensors attached to the robot hand to find good gripping points on the object. We consider two types of hands: a parallel jaw gripper with light beam sensors and a three finger hand with object normal sensors. These algorithms use ideas from geometric probing [DEY86, CY87] to avoid disturbing the object. They also perform motion planning for the hand, and are robust.

In chapter 6, the implementation of the reactive control algorithm and the associated parallel jaw gripper equipped with light beam sensors is described.

1.2 The Theory of Grasping

The hand we consider in the first part of this thesis will be somewhat idealized[1]. It consists of several independently movable force-sensing fingers; this hand is used to grasp a rigid object B. The fingers will be placed at points $\mathbf{p}$ of the boundary of B, which we shall denote by ∂B. We will consider only the points of contact of this hand with B, and ignore issues such as motion planning and accessibility. Furthermore, we make the following simplifying assumptions:

- (*Smooth Body*) B is a full-bodied (i.e. no internal holes) compact subset of the Euclidean 3-space. Furthermore, B has a piece-wise smooth boundary ∂B.

- (*Point Contact*) For each finger-contact on the body, we may associate a nominal point of contact, $\mathbf{p} \in \partial B$. We let ∂B^* denote the set of points $\mathbf{p} \in \partial B$ such that the direction $\mathbf{n}(\mathbf{p})$ normal to ∂B at $\mathbf{p}$ is well-defined; by convention, we pick $\mathbf{n}(\mathbf{p})$ to be the unit normal pointing into the interior of B.

  For each such point $\mathbf{p}$, we can define a wrench system $\{, ^{(1)}(\mathbf{p}), , ^{(2)}(\mathbf{p}), \ldots, , ^{(m)}(\mathbf{p})\}$, $(0 \leq m \leq 6)$, where the number and screw-axes of the wrench system depend on the contact type. Some of these wrenches can be *bisense* (i.e. can act in either sense) and the remaining wrenches, *unisense*. (For a discussion of screw theory, and in particular, wrenches and twists, see Appendix A.2. Also see [Hun78] and [Ohw80].)

- (*Compliance*) We will consider the case when the fingers are stiff—the force/torques applied at the fingers are generated by some actuators whose mechanics need not concern us.

Many interesting special cases occur, depending on how we model the *static friction* and the *stiction* between the fingers and the body B. In the case, where the contacts are frictionless, a finger can only apply force $\mathbf{f}$ on the body in the direction $\mathbf{n}(\mathbf{p})$ at the point $\mathbf{p}$. Also if the fingers are non-sticky, then the force $\mathbf{f}$ has a non-negative magnitude,

[1] Part of this section previously appeared in [MT92].

$f = \mathbf{f} \cdot \mathbf{n}(\mathbf{p}) \geq 0$. Such grips are also known as 'positive grips'. In this case, the wrench system associated with each point is:

$$\Gamma(\mathbf{p}) = \{[\mathbf{n}(\mathbf{p}), \mathbf{p} \times \mathbf{n}(\mathbf{p})]\}$$

Thus, corresponding to a set of finger-contacts, we have a system of n wrenches,

$$\{\mathbf{w}_1, \ldots, \mathbf{w}_k, \mathbf{w}_{k+1}, \ldots, \mathbf{w}_n\},$$

the first k of which are bisense and the remaining last $n-k$ of the wrenches are unisense. Let us assume that the magnitudes of these wrenches are given by the scalars f_i's

$$\{f_1, \ldots, f_k, f_{k+1}, \ldots, f_n\},$$

where $f_1, \ldots, f_k \in \mathbb{R}$ and $f_{k+1}, \ldots, f_n \in \mathbb{R}_{\geq}$, and not all the magnitudes are zero. We call such a system of wrenches and the wrench-magnitudes, a *grip*, G, and say that this grip G generates an external wrench $\mathbf{w} = [F_x, F_y, F_z, \tau_x, \tau_y, \tau_z] \in \mathbb{R}^6$, if

$$\mathbf{w} = \sum_{i=1}^{n} f_i \ \mathbf{w}_i.$$

In matrix notation, the above equation is expressed as

$$\mathbf{w} = \mathcal{G} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix},$$

where $\mathcal{G}$ is a $6 \times n$ matrix whose columns are the corresponding n wrenches of the system $\{\mathbf{w}_1, \ldots, \mathbf{w}_k, \mathbf{w}_{k+1}, \ldots, \mathbf{w}_n\}$, associated with the contact points of the grip. The matrix $\mathcal{G}$ is called a *grip matrix* of the grip defining the system of wrenches

$$\{\mathbf{w}_1, \ldots, \mathbf{w}_k, \mathbf{w}_{k+1}, \ldots, \mathbf{w}_n\}.$$

The grip matrix has already been used by Salisbury [Sal82] for example.

1.2.1 Equilibrium and Closure Grasps

Next, we consider the concept of a *closure grasp*:

Definition 1.2.1. *A set of gripping points on an object B to which corresponds a system of wrenches $\mathbf{w}_1$, ..., $\mathbf{w}_n$ (as before) is said to constitute a* force/torque closure grasp *if and only if any arbitrary external wrench can be generated by varying the magnitudes of the wrenches (subject to the constraints imposed by the senses of the wrenches).*

This property was introduced by Reuleaux [Reu63] in the nineteenth century. Other early studies include Somoff [Som00], and more recently Lakshminarayana [Lak78]. In robotics, this concept has been used in the context of grasping since Salisbury's thesis [Sal82].

The terminology used in these works and also the work of Markenscoff *et al.* [MNP90] however is *form closure*, while other papers by Nguyen [Ngu87], Mishra *et al.* [MSS87] and the books by Murray *et al.* [MLS94] use the terminology we adopt. We reserve the term *form closure* for a dual notion, which turns out to be equivalent; see below. For this reason, we shall sometimes simply refer to force-torque closure grasps, as *closure* grasps. Note that Trinkle [Tri92] proposes yet another notion which is quite different from ours. See [Mis94] for a comparison between these notions.

A necessary and sufficient condition for a closure grasp, which was shown in [MS89], is that the (module) sum of the linear space spanned by the vectors $\mathbf{w}_1$, ..., $\mathbf{w}_k$ and the positive space spanned by the vectors $\mathbf{w}_{k+1}$, ..., $\mathbf{w}_n$ is the entire $\mathbb{R}^6$:

$$\text{lin }(\mathbf{w}_1, \ldots, \mathbf{w}_k) + \text{pos }(\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n) = \mathbb{R}^6.$$

Let us denote, by L, the linear space lin $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$, and, by $L^{\perp}$, the orthogonal complement of L in $\mathbb{R}^6$. Let π be the linear projection function of $\mathbb{R}^6$ onto $L^{\perp}$ whose kernel is L. Then it can be shown that a necessary and sufficient condition for a closure grasp is

$$\text{lin }(\mathbf{w}_1, \ldots, \mathbf{w}_k) + \text{pos }(\pi\mathbf{w}_{k+1}, \ldots, \pi\mathbf{w}_n) = \mathbb{R}^6.$$

The above equation in turn is equivalent to the following conditions:

$$\mathbf{0} \in \text{int conv }(\pi\mathbf{w}_{k+1}, \ldots, \pi\mathbf{w}_n)$$

in $L^\perp$. Here, if $k = 0$ (i.e. positive grip) then the above condition reduces to the following:

$$\mathbf{0} \in \text{int conv } (\mathbf{w}_1, \ldots, \mathbf{w}_n) .$$

Testing for closure can be done in constant time using any convex hull algorithm. See for example [Ede87]. Several implementations of such algorithms exist. See for example O'Rourke's book [O'R94], the QHULL implementation [BDH93], or the implementation by Emiris which uses a symbolic perturbation technique for removing degeneracies [EC92].

A simple linear time algorithm for finding *at least* one set of force targets that can generate a given external wrench has been presented in [MSS87]. Also, as the external wrench is varied in the course of a manipulation task, this algorithm updates the force targets in constant time.

Similarly, we have

Definition 1.2.2. *A set of gripping points on an object B to which corresponds a system of wrenches $\mathbf{w}_1$, …, $\mathbf{w}_n$ is said to achieve an* equilibrium *grasp if there exist f_1, …, $f_k \in \mathbb{R}$ and f_{k+1}, …, $f_n \in \mathbb{R}_{\geq}$, with not all the magnitudes zero such that*

$$\mathbf{0} = \sum_{i=1}^{n} f_i \ \mathbf{w}_i .$$

And the equivalent condition [MSS87] is that

$$\mathbf{0} \in \text{conv } (\mathbf{w}_1, \ldots, \mathbf{w}_n) .$$

At this point, we emphasize that the above formulation has turned a problem in mechanics into a purely geometric problem, now amenable to many interesting techniques in convexity theory and computational and combinatorial geometry. This observation and its generalizations are the basis of this thesis.

1.2.2 Form Closure

Another approach to immobility is that first introduced in [Lak78] and further examined in [Ohw80]. Each wrench associated with a contact point reduces the set of possible twists that the body can undergo. The remaining twists are exactly those whose virtual coefficient with any wrench is non-negative, since otherwise the virtual work done by such twists would be negative.

A set of wrenches associated with the contacts constitutes a *form closure* if any arbitrary twist $\mathbf{d}$ of the object is resisted by the set of contacts, i.e. if $\mathbf{d}$ is non-reciprocal to some $\mathbf{w}_i$ $(i = 1, \ldots, k)$ or contrary to some $\mathbf{w}_j$ $(j = k+1, \ldots, n)$.

It was shown in [MS89] that form closure, also referred to as *zero total freedom*, and force closure are equivalent, as they are dual to each other.

1.3 The Case of a Positive Grip

Recall that a non-frictional grip is called a *positive grip*. Note that, in this case, the fingers are assumed to be point fingers; a finger can only apply a force on the object along the surface-normal at the point of contact, directed inward. In this situation, we have a *wrench map*, , , mapping ∂B^* into the six-dimensional *wrench space* $\mathbb{R}^6$ as follows:

$$, : \partial B^* \to \mathbb{R}^6 \ : \mathbf{p} \mapsto [\mathbf{n}(\mathbf{p}), \mathbf{p} \times \mathbf{n}(\mathbf{p})].$$

Essentially, , maps $\mathbf{p}$ to the point $,(\mathbf{p})$ in the wrench space that represents the effects of applying a unit force at $\mathbf{p}$ in the direction $\mathbf{n}(\mathbf{p})$. Furthermore $,(\partial B^*)$ belongs to the unit radius "cylinder" $S^3 \oplus \mathbb{R}^3$ (or for planar objects $S^2 \oplus \mathbb{R}$). See Figure 1.1 for an illustration. We also extend this definition to denote the closure of $,(\partial B^*)$ by $,(\partial B)$.

We can say even more about the structure of the wrench map. We show that the image of a set of points on the boundary of an object lies on the boundary of its convex hull. This will be useful later when we study grasp efficiency measure. Let us begin with a purely geometric definition and lemma.

Definition 1.3.1. *A set of points $S \in \mathbb{R}^d$ is said to be in* convex position *if no point of S can be written as a convex combination of points of S with all coefficients non-zero.*

This is equivalent [MS71, p.8] to saying that no point lies in int conv S.

Let $x_1, x_2, \ldots, x_n \in \mathbb{R}^k$ be distinct points in convex position. To each x_i, we associate a set of n_i points $y_{ij} \in \mathbb{R}^l$, $j = 1, \ldots, n_i$. Let $w_{ij} = [x_i, y_{ij}] \in \mathbb{R}^{k+l}$, and $W = \{w_{ij} : i = 1, \ldots, n;\ j = 1, \ldots, n_i\}$. Also define the projection $\pi\colon \mathbb{R}^{k+l} \to \mathbb{R}^k \ : w_{ij} \mapsto x_i$.

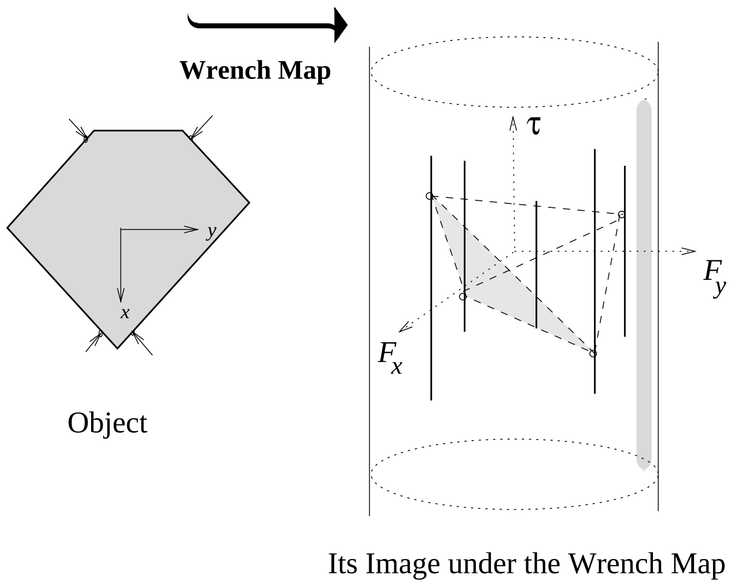


Figure 1.1: The wrench map.

Lemma 1.3.1. *For all* $i_0 \in \{1, \ldots, n\}$ *and* $j_0 \in \{1, \ldots, n_i\}$,

$$w_{i_0 j_0} \in \text{conv}\,\{w_{i_0 j} : j = 1, \ldots, n_i\}.$$

If in addition $\{y_{ij} : j = 1, \ldots, n_i\}$ *is in convex position for each* i*, then so is* W*.*

Proof. Let $q = w_{i_0 j_0}$. Clearly, $q \in \text{conv}\, W$, in other words $q = \sum_{ij} \lambda_{ij} p_{ij}$, for some λ_i, $0 \leq \lambda_{ij} \leq 1$. Then

$$\begin{aligned} \pi q &= \sum_{ij} \lambda_{ij} \pi w_{ij} = \sum_i \left(\sum_j \lambda_{ij} \right) x_i \\ &\Rightarrow \pi q \in \text{conv}\,\{x_i : i = 1, \ldots, n\}. \end{aligned}$$

But since the x_i are in convex position, $\pi q = x_s$ for some $s \in \{1, \ldots, n\}$ and we must have $s = i_0$ by equating the first k coordinates. This means $\lambda_{ij} = 0$ for $i \neq i_0$. Therefore

$$q = \sum_{j=1}^{n_{i_0}} \lambda_{i_0 j} w_{i_0 j} \tag{1.1}$$

as required. To show the second statement it is sufficient to note that if (1.1) is also satisfied then $q = w_{i_0 j_0}$ for some $j_0 \in \{1, \ldots, n_i\}$. $\square$

In particular, if $n_i = 1$, $q \in W$ must be a vertex of conv W. We also have

Corollary 1.3.2. *If for a grasp $G = \{\mathbf{p}_i : i = 1, \ldots, n\}$ with grip points $\mathbf{p}_i$, no 3 fingers have the same normal, and if $\mathbf{n}(\mathbf{p}_i) = \mathbf{n}(\mathbf{p}_j)$ then $\mathbf{p}_i - \mathbf{p}_j$ is not parallel with $\mathbf{n}(\mathbf{p}_i)$ (i.e. the lines of action of the applied forces are not identical), then $,(G)$ are vertices of* conv $,(G)$.

Proof. In Lemma 1.3.1, let $(k, l) = (3, 3)$ (or $(k, l) = (2, 1)$ for planar objects). Note that $\mathbf{n}(G)$ is in convex position, and for each normal $\mathbf{n}(\mathbf{p}_i) \in \mathbb{R}^k$, the associated points in $\mathbb{R}^l$ are also in convex position, since any two distinct points are. Then an application of Lemma 1.3.1 to $,(G)$ finishes the proof. $\square$

A slightly weaker property of $,(S)$ can be shown when S is the set of boundary points of a polyhedral object.

Theorem 1.3.3. *When B is a polyhedron with n vertices,* conv $,(\partial B)$ *is polyhedral and has at most n vertices, and $,(\partial B) \subset \partial\,$conv$\,,(\partial B)$.*

Proof. Let F be a face of B. For all $\mathbf{p} \in F$, $\mathbf{n}(\mathbf{p}) = \mathbf{n}_0$ are identical. Then $,(F)$ is a 2 dimensional polygon, possibly degenerate, by linearity of $,$. For $\mathbf{p} \in \partial B$, $,(\mathbf{p})$ belongs to the "vertical" hyperplane $h = \{\mathbf{x} \in \mathbb{R}^6 : [\mathbf{n}(\mathbf{p}), \mathbf{0}] \cdot \mathbf{x} = 1\}$. Let the positive side h^+ of h be the closed halfspace delimited by h and containing the origin. Then $,(\mathbf{p}') \in h^+$ for any point $\mathbf{p}' \in \partial B$ since $\mathbf{n}(\mathbf{p}') \cdot \mathbf{n}(\mathbf{p}) \leq 1$. Hence h is a supporting hyperplane for conv $,(B)$. Now for normal $\mathbf{n}_0$, consider the set $D = \{\mathbf{p} \in \partial B \ : \ \mathbf{n}(\mathbf{p}) = \mathbf{n}_0\}$. D is a collection of faces of B and includes F. For $\mathbf{p} \in D$, $\mathbf{p} \times \mathbf{n}_0$ lies in the 2 dimensional subspace aff F of $\mathbb{R}^3$ (one dimensional for planar objects) since any point of $,(D)$ has its first three (two) coordinate those of $\mathbf{n}_0$, and in addition, $\mathbf{p} \times \mathbf{n}_0 \perp \mathbf{n}_0$. $,(D)$ is the union of the image by $,$ of the set of polygonal faces of D which is a set of planar polygons, a set with *linear* boundary, as seen above. Since h is a supporting hyperplane for conv $,(\partial B)$ and no other points of $,(\partial B)$ lie on h but those of conv $,(D)$, conv $,(D)$ must be a 2-face (edge) of conv $,(\partial B)$. $\square$

If B is a polyhedron with N vertices and faces, then the , (∂B) has at most $O(N)$ vertices and at most $O(N^3)$ total complexity [McM70, Ede87]. Computing the convex hull of , (∂B) takes at most $O(N^3)$ time [Cha93]. If B is planar with N vertices, then the number of faces of , $(\partial B) \in \mathbb{R}^3$ is only $O(N)$, and its convex hull takes only $O(N \log N)$ time to compute. It is interesting to note that the computation of the convex hull requires only $O(N)$ time if the points are sorted by their corresponding normal. This is due to the structure of the set of points: all points lie on N vertical segments. This follows from the results of Aggarwal, Guibas, Saxe and Shor [AGSS89]. They provide a linear time algorithm for computing the convex hull of points structured as in Lemma 1.3.1 with $n_i = 2$, $k = 2$ and $l = 1$, *i.e.* points which lie “above” the vertices of a convex polygon in the xy-plane, two points per vertex. They assume the polygon is given as an ordered list. In our case, the points project onto the unit circle centered at the origin in the xy-plane, and we need to know in addition which points are the highest and lowest for each vertical segment and identify the vertical segments. A similar structure occurs in six dimensions for the case of three dimensional objects. We therefore believe that the bound on the number of faces for three dimensional objects can also be improved.

1.3.1 Number of Fingers Required for Closure

In order to obtain a particular grasp on an object, it must be determined if that grasp is achievable. It is for this reason, researchers have studied the question of how many fingers (wrenches) are required to obtain certain grasps on the object.

Here we consider the frictionless case. Reuleaux and Somoff, followed by Lakshminarayana determined that a closure grasp of a two dimensional object requires at least *four* fingers and of a three dimensional object requires at least *seven*.

Mishra, Schwartz, and Sharir [MSS87] then gave general bounds on the number of fingers in the case of a *positive grip*, for piece-wise smooth surfaces. They show that six (resp. twelve) fingers are sufficient to obtain a closure grasp of objects bounded by *non-exceptional* surfaces in the plane (resp. in three dimensions). To achieve equilibrium, only four (resp. seven) fingers are required.

A surface Σ is exceptional [MSS87] if for some wrench $[\mathbf{f}, \mathbf{t}]$, at each point $\mathbf{p}$ on Σ, the normal $\mathbf{n}(\mathbf{p})$ at $\mathbf{p}$ is perpendicular to $\mathbf{f} + \mathbf{t} \times \mathbf{p}$. It is shown in that paper that such surfaces are exactly surfaces of revolution. A similar result has been obtained by Selig and Rooney [SR89].

Let us briefly describe the technique of [MSS87] for showing these bounds. But first let us recall a fundamental theorem from combinatorial geometry[2] [Ste16].

Theorem 1.3.4 (Steinitz). *If* $X \subseteq \mathbb{R}^d$ *and* $x \in \text{int conv } X$*, then* $x \in \text{int conv } Y$ *for some* $Y \subseteq X$ *with* $|Y| \leq 2d$. $\square$

First it is shown that if a finger is placed at every point of the object, then closure is achieved for non-exceptional objects. In other words,

$$0 \in \text{int conv } \mathbf{,}\ (\partial B).$$

Then Steinitz's theorem is applied to $\mathbf{,}\ (\partial B)$ to show the bounds for closure. For the bounds on the number of fingers needed for equilibrium, Carathéodory's theorem [Car07] is used.

Theorem 1.3.5 (Carathéodory). *If* $X \subseteq \mathbb{R}^d$ *and* $x \in \text{conv } X$*, then* $x \in \text{conv } Y$ *for some* $Y \subseteq X$ *with* $|Y| \leq d + 1$. $\square$

Mishra *et al.* also provide an algorithm that finds *at least one* such grip on a polyhedral object which runs in time linear in the number of faces of the object.

Similar results have also been obtained by Markenscoff, Ni, and Papadimitriou [MNP90], although their proof techniques are quite different from those used by [MSS87].

For certain types of objects, it can be shown that *seven* fingers are sufficient thus closing the gap between the bounds for those objects. Let us first give a definition.

Definition 1.3.2. *A set of vectors in* $\mathbb{R}^d$ *is* positively spanning *if every vector in* $\mathbb{R}^d$ *can be expressed as a positive linear combination of those vectors.*

[2] We attempt to distinguish general facts in combinatorial geometry from facts about wrench spaces by using bold type for vectors in the latter.

	2D Objects	3D Objects
Equilibrium grasps		
Piecewise smooth	4	7
Smooth	3	5
Closure grasps		
Piecewise smooth	6 (excluding disks)	12 (excluding objects with a surface of revolution)
Polyhedral	4	7

Table 1.1: Bounds on the number of fingers sufficient for force/torque closure.

In [MNP90] it is shown that such a grasp is achievable for an object B in $\mathbb{R}^3$ without rotational symmetry such that there is a maximal inscribed sphere S and a set P of isolated points on $\partial B \cap S$ such that the normals to B at those points are positively spanning. This is shown using the largest inscribed sphere inside the object and a perturbation argument. For planar objects, the bound is four. In particular the result applies for polyhedral objects. For convex polyhedral objects, Meyer [Mey90] also shows that seven fingers are sufficient and provides an algorithm with $O(n^{3/2}\sqrt{\log n})$ running time to find such a grasp. In [Mey90], an initial grasp of six fingers is found. It consists of three fingers near each of the vertices forming the diameter of B. This is also done using an inscribed sphere near those vertices. Torque cannot be exerted around the line connecting the two spheres, but this is remedied by splitting one of the points into two nearby ones, again achieving form closure. Table 1.3.1 summarizes existing bounds found in [MSS87, MS89, Mey90, MT92, MLS94].

1.3.2 Algorithmic Techniques

Given the bounds on the number of fingers, and a polyhedron P of n faces, we wish to find a set of fingers and force targets that will produce a force-torque closure grasp.

Meyer's algorithm has been briefly described in the previous section. The complexity of the algorithm is due mainly to finding the diameter of the convex polyhedron. It would be interesting to determine whether another pair of vertices that is not so hard to find might suffice.

Mishra, Schwartz and Sharir [MSS87] have an algorithm with a more algebraic flavor for any polyhedral object. The algorithm works in two phases: the first phase finds a set of $O(n)$ fingers forming a force-torque closure grasp, the second then reduces the number of fingers to a constant number. In fact this algorithm may find a grasp with less than twelve (six) fingers, but this is not guaranteed.

Markenscoff, Ni, and Papadimitriou [MNP90], briefly mention that their proofs are constructive and can be applied to the case where the object is a polytope. Their algorithm consists of finding a largest inscribed sphere. For convex polytopes, this can be done in linear time. See for example [Meg84]. For non-convex polytopes, the algorithm of Milenkovik [Mil93] for computing the Voronoi diagram of a polyhedron can be used. It runs in $O(nv \log^2 b)$ time, where v is the number of vertices of the Voronoi Diagram of the n vertex polyhedron, and b is the number of desired bits of precision. This produces four or more points on the boundary of the object where the sphere touches (recall the conditions of validity.) Then finger positions, near these points must be found. For example if four points are found, three of the points are "split" appropriately, producing a seven finger grasp. Other cases are similarly covered.

1.4 Friction and Other Contact Types

The number of fingers necessary for a closure grasp can be reduced in the presence of friction. Usually the Coulomb friction model for the surface contacts is assumed, as is done for example in [SR82] and [KR86]. A friction cone is associated with each contact, and the line of action of the force transmitted through the contact must lie within this cone.

In the plane, only three fingers are required for form closure and in $\mathbb{R}^3$, form closure is possible with four fingers, and these numbers are also lower bounds. This was shown in [MNP90].

Let $\mu > 0$ be the friction coefficient. Form closure with n fingers under friction occurs when any arbitrary external wrench $\mathbf{w}$ can be expressed as

$$\mathbf{w} = \sum_{i=1}^{n} f_i \Big[\Gamma(\mathbf{p}_i) + t_i (\mathbf{n}_i^{\perp}, \mathbf{p}_i \times \mathbf{n}_i^{\perp}) \Big], \tag{1.2}$$

where $f_i \geq 0$ and $\mathbf{n}_i^{\perp}$ is a unit normal to $\mathbf{n}(\mathbf{p}_i)$ with $0 \leq t_i \leq \mu$, $(1 \leq i \leq n)$. In other words, the actual force applied at point $\mathbf{p}_i$ is a non-negative multiple of $\mathbf{n}(\mathbf{p}_i) + t_i \mathbf{n}_i^{\perp}$ and belongs to the friction cone $\mathcal{C}_\mu(\mathbf{p}_i)$, a set of vectors forming an angle of at most $\arctan(\mu)$ with $\mathbf{n}(\mathbf{p}_i)$. Let us also define the set of forces of unit magnitude in the friction cone

$$\mathcal{F}_\mu(\mathbf{p}) = \{\mathbf{f} \in \mathcal{C}_\mu(\mathbf{p}) : \|\mathbf{f}\| = 1\}.$$

We now show that the techniques for testing the closure properties of Section 1.2.1 still apply. We would like to say that since the finger can apply a force anywhere in the friction cone, we can include the image by pos Γ of the entire friction cone in the set of forces that can be generated at the contact point. Let $\Gamma(\mathbf{p}, \mathbf{f}) = [\mathbf{f}, \mathbf{p} \times \mathbf{f}]$ and define

$$\Gamma_\mu(\mathbf{p}) = \Gamma(\mathbf{p}, \mathcal{F}_\mu(\mathbf{p})). \tag{1.3}$$

Note that $\Gamma_0(\mathbf{p}) = \{\Gamma(\mathbf{p})\}$. We require the force applied from any direction to be of unit magnitude (in contrast with [FC92]) so that $f\Gamma_\mu(\mathbf{p})$ represents the effect of applying a force of magnitude f at $\mathbf{p}$. This avoids certain problems when evaluating grasp strength as discussed in chapter 2.

To generate an external wrench $\mathbf{w}$, we would like to select points from $\bigcup_i \text{pos}\, \Gamma_\mu(\mathbf{p}_i)$ each point corresponding to a finger position, direction and magnitude of applied force.

There are two potential difficulties. First, to a point in wrench space may correspond two or more fingers. This is the case for example when two fingers apply the same force at two points on the line of action of this force. In this case, one may pick either finger to generate that wrench.

The second difficulty is that we are allowed to select *at most one* point (actually ray) for each pos $\Gamma_\mu(\mathbf{p}_i)$ since a finger can apply a force only in one direction at any one time. We now show that this is always possible. Our solution will become particularly important in the next chapter where forces that fingers can apply are bounded. There, it is not sufficient to simply add all the wrenches that a finger is required to apply, since this might yield too large a force. We will need the following properties of convex sets and of the wrench map. For $S \subseteq \mathbb{R}^d$, denote by $[0,1]S$ the set $\{\lambda s \in S : 0 \leq \lambda \leq 1\}$.

Lemma 1.4.1. *Both* $\text{pos}\,\Gamma_\mu(\mathbf{p})$ *and* $[0,1]\Gamma_\mu(\mathbf{p},\mathbf{n})$ *are convex.*

Proof. To show the first statement, let $\mathbf{w}_i = [\mathbf{f}_i, \mathbf{p}_i \times \mathbf{f}_i] \in \text{pos}\,\Gamma_\mu(\mathbf{p})$, $i = 1, 2$. For $\lambda_1, \lambda_2 \geq 0$ with $\lambda_1 + \lambda_2 = 1$,

$$\lambda_1 \mathbf{w}_1 + \lambda_2 \mathbf{w}_2 = [\lambda_1 \mathbf{f}_1 + \lambda_2 \mathbf{f}_2, \mathbf{p} \times (\lambda_1 \mathbf{f}_1 + \lambda_2 \mathbf{f}_2)].$$

But $\lambda_1 \mathbf{f}_1 + \lambda_2 \mathbf{f}_2 \in \mathcal{F}_\mu(\mathbf{p})$ by convexity of the friction cone, which proves the first statement. Since $\lambda\Gamma_\mu(\mathbf{p},\mathbf{n}) = \Gamma_\mu(\mathbf{p},\lambda\mathbf{n})$, if in addition $\|\mathbf{f}_i\| \leq 1$, then we have $\|\lambda_1 \mathbf{f}_1 + \lambda_2 \mathbf{f}_2\| \leq \lambda_1 + \lambda_2 = 1$ by the triangle inequality and the second statement follows. $\square$

In two dimensions, the friction cone $\mathcal{C}_\mu(\mathbf{p})$ is bounded by two rays emanating from the origin and its image in wrench space $\Gamma_\mu(\mathbf{p})$ is a planar cone (Γ being linear) and also bounded by two rays. For three dimensional objects however, the friction cone is a circular cone in three dimension with apex at the origin. Its image under Γ is a convex cone with apex at the origin, but is no longer linear. An early analysis of the image of a friction cone was done by Ji [Ji87].

Let us also show two generalized versions of Carathéodory's theorem, which we call *thickened* versions[3].

Theorem 1.4.2. *Consider a family* $A_1, A_2, \ldots, A_n$ *of convex sets in* $\mathbb{R}^d$ *and a point* $x \in \text{conv}\,(A_1, \ldots, A_n)$. *Then there exist* $m = \min\{n, d+1\}$ *points* $y_1, y_2, \ldots, y_m$, *each belonging to a unique* A_i *such that* $x \in \text{conv}\,\{y_1, \ldots, y_m\}$.

Proof. By Carathéodory's theorem, there exist $d+1$ points $y_1, y_2 \ldots, y_{d+1} \in \bigcup_{i=1}^n A_i$ such that

$$x = \sum_{i=1}^{d+1} \lambda_i y_i \text{ with } 0 \leq \lambda_i \leq 1. \tag{1.4}$$

[3] A different generalization called a *multiplied* [GW93, p.431] version appears in [Bár82].

Assume without loss of generality that $p_1, p_2 \in A_1$ and $\lambda_1, \lambda_2 \neq 0$. Then

$$\begin{aligned} x &= \lambda_1 y_1 + \lambda_2 y_2 + \sum_{i=3}^{d+1} \lambda_i y_i \\ &= (\lambda_1 + \lambda_2) \left(\frac{\lambda_1 y_1 + \lambda_2 y_2}{\lambda_1 + \lambda_2} \right) + \sum_{i=3}^{d+1} \lambda_i y_i \\ &= (\lambda_1 + \lambda_2) q + \sum_{i=3}^{d+1} \lambda_i y_i \end{aligned}$$

for $q = \left(\frac{\lambda_1}{\lambda_1+\lambda_2}\right) y_1 + \left(\frac{\lambda_2}{\lambda_1+\lambda_2}\right) y_2 \in A_1$ by convexity of A_1. Now x lies in the convex hull of d points which resides in an affine space of dimension $d-1$. In effect, the hyperplane containing $y_3, \ldots, y_{d+1}$ is rotated around these points until it contained x. This process can be continued until the requirement of the theorem is met. □

Note that this procedure can easily be made constructive given an appropriate description of the convex sets A_i. In particular when the A_i are polytopes given by their vertices, and we are given λ_i and y_i sorted by their set number, such that (1.4) is satisfied, then it takes only $O(d)$ time to perform one step in the proof of the theorem, hence the total time is bounded by $O(nd)$. Note also that Theorem 1.4.2 becomes Carathéodory's theorem when the sets A_i are reduced to one point each.

We also have the following

Theorem 1.4.3. *Consider a family $A_1, A_2, \ldots, A_n$ of convex sets in $\mathbb{R}^d$ and a point $x \in \operatorname{conv}(0, A_1, \ldots, A_n)$. Then there exist $m = \min\{n, d\}$ points $y_1, y_2, \ldots, y_m$, each belonging to a unique A_i such that $x \in \operatorname{conv}\{0, y_1, \ldots, y_m\}$.*

Proof. Similar to that of Theorem 1.4.2 using a version of Carathéodory's theorem which allows us to specify a point (0 in this case) such that $x \in \operatorname{conv}\{0, y_1, \ldots, y_d\}$. See for example the paper by Bárány [Bár82] for (a more general version of) this theorem. □

Applying Theorem 1.4.3 to the sets pos $,_\mu(\mathbf{p}_i)$, which are convex by Lemma 1.4.1, we conclude that only one point from each is necessary to generate an external wrench (which is called x in the theorem.) Hence $\bigcup_i$ pos $,_\mu(\mathbf{p}_i)$ does indeed represent the set of possible wrenches the fingers can generate. The second problem mentioned above is solved. Therefore, to test for closure in the frictional setting, we can check if the origin is in the interior of the convex hull of the friction cones associated with the finger contacts. This is however computationally non-trivial, as $\mathrm{pos}\,\Gamma_{\mu}(\mathbf{p}_i)$ has non-linear boundaries, at least for three dimensional objects. In practice, an approximation is usually used [PSS+95]; in chapter 2 this is examined more closely. The test for closure can then be implemented as in the non-frictional case using such an approximation.

It is also interesting to note [MNP90, lemma 9] that a given grasp with positively independent finger wrenches is a form closure under friction coefficient $\mu > 0$ if and only if it is a form closure with unbounded friction coefficient. This is used to prove the finger bounds in their paper.

Two finger grasps and other contact types

In three dimensions, two finger grasps are slightly more complex. Salisbury [Sal82] and later, Murray, Li and Sastry [MLS94] show that two fingers with soft frictional contact are sufficient for attaining closure in certain cases. The condition (satisfied for smooth objects for example) is that there are two contact points $\mathbf{p}_1$ and $\mathbf{p}_2$ with surface normals $\mathbf{n}_1$ and $\mathbf{n}_2$ resp. such that the line of action of the force applied at one point goes through the other point. Since the angle is bounded by $\arctan(\mu)$ we need to find two points on the object that are 'almost antipodal'. The wrenches generated by such a finger include those above for the frictional contact, along with $[\mathbf{0}, t\mathbf{n}]$ with $|t| < \sigma f$, where f is the magnitude of the force applied by the finger, and σ is the torsional friction coefficient.

There are other contact types that should be included in the study of form closure grasps. The theory described here can be used to analyze contacts such as line contacts, planar contacts, both frictional and non frictional. These are described in detail by Salisbury [Sal82] and Cutkosky [Cut85]. A first step in this direction has recently been taken by Overmars, Rao, Schwarzkopf and Wentik [ORSW95] who consider a line contact in the plane. There are also different models of friction for materials such as rubbers, which usually display a coefficient of friction that varies as a function of the normal force [CAHK87]. Algorithms need to be developed for finding good grasps using those more general contact types.

1.4.1 Algorithmic Techniques

Two and Three finger grasps

In the plane, Nguyen [Ngu87] gives an algorithm for finding two maximal regions on a polygon, called maximal *independent* regions, on which fingers can be placed. Each finger can be placed anywhere in its region and the grasp maintains closure. Synthesizing a grasp with n independent regions, requiring c contacts to achieve closure takes $O(cn^c 2^c)$ time. A generalization to three fingers has been done by Pollard and Lozano-Pérez [PLP90] as part of a complete manipulation system. For non-polygonal objects, Chen and Burdick [CB92] and Ponce, Stam and Faverjon [PSF93] have also generalized the maximal independent regions of Nguyen for curved objects using global optimization and algebraic techniques.

Let us describe a new algorithm for finding such two finger grasps for polyhedral objects. We have seen in the previous section the conditions for two finger closure. Here we assume that both fingers cannot be placed at the vertices or edges. The following considerations yield a simple $O(n^2)$ algorithms for finding the set of all possible two finger grasps for convex polyhedral objects. Consider a polyhedron $P \in \mathbb{R}^3$ with n vertices (it also has $O(n)$ faces and edges [Ede87]). Let F_1, F_2 be two faces of P, with normals $\mathbf{n}_1, \mathbf{n}_2$ respectively.

The Minkowski sum of F_1 and the negative of the friction cone for points on F_1 represents the set of points in space where the second finger can be placed without violating the friction constraint for the first finger. This produces a curved surface with conical and planar pieces [PSS+95] of complexity no larger than $O(N)$, if N is the number of vertices of F_1. This surface, when intersected with the plane of F_2 forms a "generalized" polygon Q_2 on F_2 with elliptical and straight edges. We do the same for F_2 producing Q_1. If Q_1 or Q_2 is empty, it means we cannot place fingers on this pair of faces. Otherwise, one can simply select two points, one on each Q_i on which to place the fingers. Such polygons fall into the category of *spline-gons* which have been studied by Dobkin and Souvaine [DS90]. They show that the intersection of two such polygons with at most N (curved) edges each can be accomplished in $O(N)$ time. Hence, we can test each pair of faces in $O(N)$ time where N is the size of the largest face. If we fix one of the faces, the sum of this number over the entire polyhedron is $O(n)$ since each edge is counted only twice. The entire algorithm therefore takes $O(n^2)$ time, and finds every candidate pair of faces for two finger grasps.

Four finger grasps

Recently Ponce, Sullivan, Sudsang and Boissonnat [PSS+93, PSS+95] have classified four finger frictional grasps of three dimensional objects using screw theory and propose methods for finding four finger equilibrium and closure grasps.

They define an m-finger grasp to achieve *non-marginal* equilibrium when there exists a set of forces in the *open* friction cones at the finger contact points such that the associated wrenches sum to zero. Their results are based on

Theorem 1.4.4 (Ponce et al). *In the presence of friction, a sufficient condition for three dimensional, m-finger closure with $m \geq 3$ is non-marginal equilibrium.*

Equivalently, one could say that an equilibrium grasps with friction coefficient μ is a force/torque closure grasps with friction coefficient $\mu' > \mu$. This generalizes a result of Nguyen for the two finger case.

They apply this result to finding four finger non-marginal equilibrium grasps, which are in fact closure grasps by their theorem. This result allows them to work in object space instead of wrench space, simplifying the algorithms for finding a grasp. They also use the following classical result [Hun78]. Four lines in space which are linearly-dependent, either

1. lie in a single plane,
2. intersect at a point,
3. form two flat pencils having a line in common, or
4. form a regulus.

They categorize four finger equilibrium grasps into three classes corresponding to the last three possibilities above. Each line corresponds to the line of action of a force applied by a finger. The first possibility above does not constitute a grasp since the forces must be positively spanning to span the force space $\mathbb{R}^3$.

Ponce *et al.* give algorithms for finding independent regions corresponding to the first two types of grasps. Also independent regions for three finger grasps are found. Their algorithms are based on linear programming. The algorithms however are not guaranteed to find a grasp if it exists, since they are based on sufficient, but not necessary conditions. Sudsang and Ponce [SP95] recently gave algorithms for finding the third type grasps, but not independent regions which appear hard to find in this case.

1.5 Fixturing of Frictionless Assemblies

We now consider the case of grasping several objects that are possibly in contact with each other, using a set of fingers, or *fixture elements*. We wish to take advantage of the object to object contacts to reduce the number of fingers necessary. This problem arises in manufacturing where many assembly tasks require a set of contacting objects to be held firmly. The fingers, here called fixture elements or *fixtures* are positioned in contact with the objects to achieve this. Often there is only a finite set of possible placements due to the construction of workholding table [Mis91, ZGW94]. Typically, such a table offers a precise lattice of holes into which *clamps* and *locators* can be inserted. Locators are essentially pins of a certain diameter. Clamps can apply pressure on the object at a point on one of the lattice axes. Thus, for a fixed object position, there is a finite number of possible points on the boundary of the objects where the fixture elements can make contact. See Figure 1.2 for an illustration. The contacts considered in this section are frictionless. This is particularly justified in manufacturing where vibrations can make frictional contacts unreliable, and machining forces can be particularly large.

In the case of a single object, Brost and Goldberg [BG94b] provide an algorithm which finds all possible positions of an object on such a table, along with fixtures corresponding to those positions. Zhuang, Goldberg and Wong [ZGW94] show that for a given lattice size, there exist polygonal objects which cannot be fixtured using three locators and

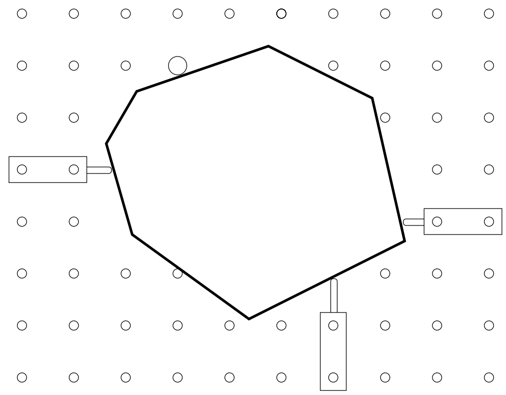


Figure 1.2: Fixturing a single object with 3 clamps and 1 locator

one clamp. Recently, Overmars, Rao, Schwarzkopf and Wentik [ORSW95] presented an algorithm to list all possible fixture sets of a polygonal object using one edge fixture, one clamp and a locator as fixtures. If k is the number of fixture configurations that are output, n the number of vertices of polygon, and p the polygon's perimeter in grid units, their algorithm runs in time $O(n(n+p)^{4/3+\epsilon}+k)$.

We consider here the case of multiple fixed objects. In this setting, forces applied to an object are both due to contacts between objects, and the fixtures. As in grasping, it is desirable to use a small number of fixtures, and/or limit the forces they must apply on the objects. The previous framework of section 1.2 generalizes to this case by essentially concatenating the force-torque vectors for each object and working in $\mathbb{R}^{6k}$ where k is the number of objects [BMK94].

Let $\mathbf{m}_i$ be the center of mass of object i. In the following definitions 0 is the 6 dimensional zero vector. For each object B_i and contact point $\mathbf{p}$ let $,_i(\mathbf{p}) = [\mathbf{n}(\mathbf{p}), (\mathbf{p} - \mathbf{m}_i) \times \mathbf{n}(\mathbf{p})]$. A fixture applied at $\mathbf{p}$ to B_i generates the generalized force-torque

$$,_*(\mathbf{p}) = [\underbrace{0, \ldots, 0}_{i-1}, ,_i(\mathbf{p}), 0, \ldots, 0] \in \mathbb{R}^{6k}.$$

The i-th position contains $,_i(\mathbf{p})$. A contact between object B_i and object B_j generates

$$,_*(\mathbf{p}) = [\underbrace{0, \ldots, 0}_{i-1}, ,_i(\mathbf{p}), \underbrace{0, \ldots, 0}_{j-i-1}, ,_j(\mathbf{p}), 0, \ldots, 0] \in \mathbb{R}^{6k}.$$

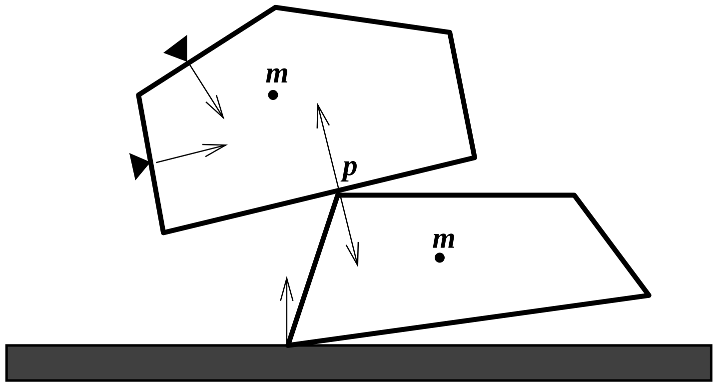


Figure 1.3: An assembly with fixtures.

The forces involved are illustrated in Figure 1.3. It is worth noting that as before this framework specializes to the case where the objects are planar; in this case, the force-torque space for each object is three-dimensional.

In this context force-torque closure can be defined analogously to the previous one-object case: a set G of fixtures (i.e. of these generalized force-torque vectors) is a force/torque closure fixture set if and only if $\mathbf{0}$ is in the interior of conv $\Gamma_*(G)$. It is easy to see that

$$\Big(\forall i \quad \mathbf{0} \in \operatorname{int}\operatorname{conv}\Gamma_i(G)\Big) \quad \text{if and only if} \quad \mathbf{0} \in \operatorname{int}\operatorname{conv}\Gamma_*(G).$$

so we have closure for each object individually exactly when the definition above is satisfied.

We note that the techniques of [MSS87] for finding a closure grasp apply in any dimension and can be used for finding closure grasps. This was observed by Baraff, Mattikalli and Khosla [BMK94]. Again, by Steinitz's theorem, one needs at most $12k$ fixtures (or $6k$ in the plane).

One problem addressed in [BMK94] is to select the smallest subset G of fixtures among a finite set S of possible fixtures that have this property. Again, there is a bound of $12k$ on the number of fixtures required to achieve force-torque closure. However such a set G does not guarantee any given grasp quality. They show that several variations of this problem are NP-hard, they also provide two heuristics for the problem described here, but without an analysis. In Chapter 4 we present a randomized algorithm which produces a number of planar fixtures that is at worst a factor of $12k \log c$ times larger than the optimal cover, if the latter has size c.

1.6 Other Types of Immobility

The analysis of grasping given so far is based on screw theory and depends only on the infinitesimal motions and forces. There is another method for analyzing the relative motion of bodies in contact. It considers the configuration space of object. For a definition of configuration space (also called c-space), see for example the book by Latombe [Lat91]. In fact there are two other possible notions of immobility and we shall examine them in this section. Here all contacts are frictionless. The purpose of this section is to show a parallel between two related fields. Certain definitions, based on which we shall not show any results, will not be given in their full mathematical precision as this would require a disproportionate amount of space. We refer the interested reader to the corresponding references.

1.6.1 First and Second Order Immobility

Recently, Rimon and Burdick [RB94] consider the configuration space of a rigid body in contact with other fixed bodies (fingers) and study possible free motions of the body. They propose two notions of immobility.

First order immobility is defined in terms of the local tangent hyperplanes in configuration space at the initial configuration. To each finger is associated an open halfspace. If these halfspaces cover the entire c-space (except for the initial configuration), there is no first order motion possible. It is shown that first order immobility is equivalent to force/torque closure.

Second order immobility depends on the local curvature of the configuration space, hence also on the local relative curvature of the fingers and the body at the contact points.

In [RB94] it is shown that a set of fingers which immobilizes an object to first order also immobilizes it to second order. Furthermore, they show that some grasps that are not force/torque closure are immobile to second order. Consider for example a triangle with fingers placed at the intersection of the triangle and its inscribed circle. The first order theory fails to distinguish between these two types of grasps.

1.6.2 Finite Immobility

Yet a different concept of immobility, which we call here *finite* immobility, has been introduced by Kuperberg [Kup90], and discussed in [O'R90] and [CSU90], [CSU91].

Definition 1.6.1. *A set of points I in $\mathbb{R}^3$ is said to* immobilize *an object B if any rigid motion of the object causes at least one point of I to penetrate the interior of B.*

Czyzowicz, Stojmenovic and Urrutia [CSU90] establish that four points suffice to immobilize any object in the plane, and $2d$ points are sufficient in $\mathbb{R}^d$ using techniques such as finding the largest sphere inscribed inside the object. The arguments bear some similarity to those used in [BFG85] and [MNP90] to show analogous results about the number of fingers required for closure grasps. They also show that polygons without parallel sides can be immobilized using three points, and provide an $O(n \log n)$ algorithm for finding such an immobilizing set [CSU91].

For smooth objects, its turns out that three points are always sufficient [MU91]. This mirrors the reduction of number of fingers required for closure grasps when one requires the object to have a smooth boundary. Similarly, Rimon and Burdick [RB94] conjecture such a reduction for second order immobility. In fact they show that a force/torque closure is sufficient to achieve immobility. Thus any algorithm which computes a frictionless closure grasp of an object also finds a set of points immobilizing the object. Although Rimon and Burdick mention the work of Czyzowicz *et al.*, they do not seem to explicitly show a connection between second order immobility and finite immobility.

Finally, testing whether a given set of points immobilizes a polygon takes $O(n)$ time [CSS92] if the points are not placed on convex vertices, otherwise the bound is $O(n^2)$.

We can summarize the relationship between the various definitions of immobility as follows. Let *Definition A* $\Longrightarrow$ *Definition B* mean *if a body is immobilized by n fingers* *under Definition A then it can be immobilized by n or fewer fingers under Definition B.* Then

$$\begin{array}{lcl} \text{Force/Torque Closure} & \Longleftrightarrow & \text{Form Closure} \Longleftrightarrow \text{First Order Immobility} \\ & \Longrightarrow & \begin{cases} \text{Second Order Immobility} \\ \text{Finite Immobility.} \end{cases} \end{array}$$

1.6.3 Rigidity of Bar and Joint Frameworks

A *framework* G in $\mathbb{R}^d$ consists of a set of v *vertices* $p_i \in \mathbb{R}^d$ or *joints* together with a set of e *edges*. An edge $\{i, j\}$ connects two vertices i and j, and represents a rigid *bar*. To such a framework, we associate a point in the configuration space of G:

$$p = (p_1, \ldots, p_v) \in \mathbb{R}^{dv}$$

which represents the positions of the vertices in $\mathbb{R}^d$. Edges are simply closed line segments $[p_i, p_j]$. Define the *edge function* $f : \mathbb{R}^{dv} \to \mathbb{R}^e$ of G by

$$f(p) = (\ldots, \|p_i - p_j\|^2, \ldots).$$

Definition 1.6.2. *The framework $G(p)$ is* flexible *if there exists a continuous function $x : [0,1] \to \mathbb{R}^{dv}$ satisfying*

1. $x(0) = p$
2. $x(t) \in f^{-1}(f(p))$ *for all* $t \in [0,1]$, *and*
3. $x(t)$ *is not an image of* p *by a rigid motion for all* $t \in (0,1]$.

Such a path is called a flexing *of $G(p)$. The framework is* rigid *if it is not flexible.*

In other words, there is a path in the configuration space that begins at p, which preserves edge lengths and which is not simply a rigid motion. See for example [Rot81, CS94], where most of this material can be found.

In this setting there are also several additional types of immobility. We will see that the definitions of various types of rigidity parallel those for immobility. This connection is formalized in the next section, at least for rigidity and immobility.

A *first-order flex* of $G(p)$ is a configuration of vectors $p' = (p'_1, \ldots, p'_v) \in \mathbb{R}^{dv}$ such that the equations

$$(p_i - p_j) \cdot (p'_i - p'_j) = 0 \qquad \text{for all } \{i, j\} \text{ bars of } G$$

are satisfied. These equations come from the formal derivative of $f(p)$. If G has pinned vertices, we will say that $G(p)$ is *first-order rigid* if its only first-order flexes satisfy $p' = 0$.

Finally, a *second-order flex* is a pair of configurations of vectors (p', p''), where p' is a first order flex of $G(p)$ and $p'' = (p''_1, \ldots, p''_n) \in \mathbb{R}^{dv}$ such that the equations

$$(p_i - p_j) \cdot (p''_i - p''_j) + (p'_i - p'_j) \cdot (p'_i - p'_j) = 0 \qquad \text{for all } \{i, j\} \text{ bars of } G$$

are satisfied and in addition $p''_i = 0$ for all pinned vertices i. One can interpret p' as velocities and p'' as accelerations permitted by the distance constraints of $G(p)$. If G has pinned vertices, $G(p)$ is *second-order rigid* if every second-order flex (p', p'') has $p' = 0$.

We are now ready to state some of the basic results in this field.

1. $G(p)$ is first order rigid, if and only if the rank of $df(p) = dv - \binom{d}{2}$ [Rot81].

2. If $G(p)$ is first order rigid, then it is rigid [AR79].

3. If $G(p)$ is first order rigid, then it is second order rigid. But the converse is not true.

4. If $G(p)$ is second order rigid, then it is rigid [Con80].

5. If $G(p)$ is in "general position" (p is a regular point of f), then $G(p)$ is rigid if and only if it is first-order rigid [Rot81].

These results are summarized below.

$$\begin{tikzcd}
& \text{First Order Rigidity} \arrow[dl] \arrow[dr] & \\
\text{Second Order Rigidity} \arrow[rr] & & \text{Rigidity} \arrow[ul, dotted, bend right, "\text{if non-degenerate}"']
\end{tikzcd}$$

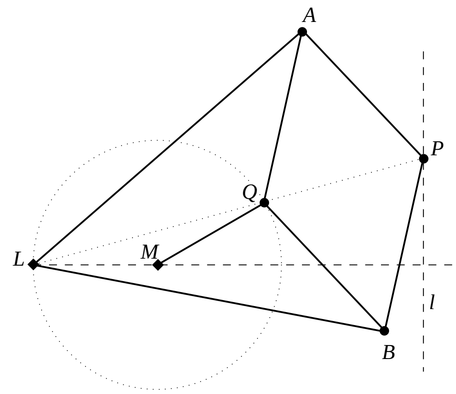


Figure 1.4: Peaucellier's Inversor.

1.6.4 Immobility and Rigidity

The previous subsections clearly show a parallel between notions of rigidity in bar and joint frameworks and notions of immobility. In fact it seems the underlying concepts are the same.

A simple construction establishes a connection between testing for finite immobility and testing framework rigidity. Namely, given a polygon and a set of points on its boundary, we give a construction that is rigid if and only if the polygon is immobilized by these points. Consider the mechanism of Peaucellier illustrated in Figure 1.4. It is described by Hilbert and Cohn-Vossen in [HCV52, page 273]. The vertices L and M are fixed to the plane and the length of bar QM and the distance LM must be identical. This mechanism allows the vertex P to move only along line ℓ, subject to the restrictions imposed by the length of the bars. We will make a slight modification to allow P to move on one side of the line (on the right side in the figure), again subject to the same restrictions. We simple introduce a joint Q' in the middle of bar MQ. This allows distance LQ to shorten and P to move away from L. First note that L, Q and P are always collinear, and that $LQ.LP = \text{constant} = (LA)^2 - (QA)^2$ [HCV52, page 273]. Also Q is on or inside the circle of radius MQ centered at M. For a given direction of LP, and for $\angle LQM < \pi/2$, Q can move closer to L by shortening MQ, but this moves P to the right of ℓ.

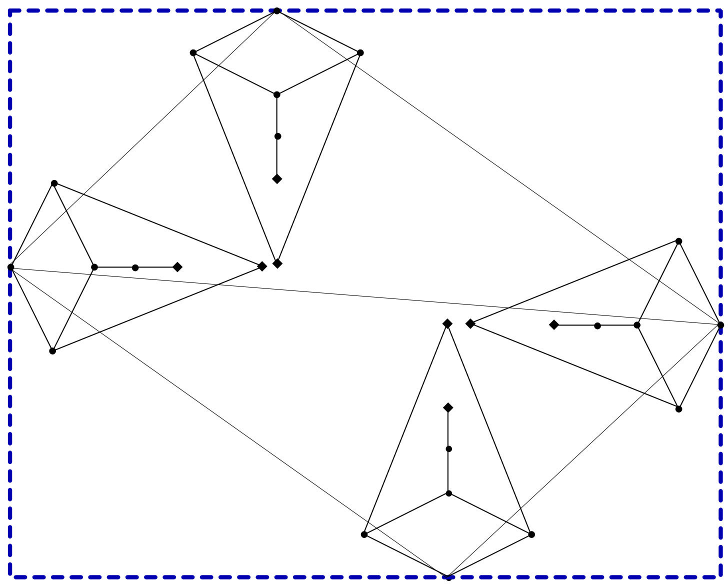


Figure 1.5: Inversors used to immobilize a rectangle.

Now imagine ℓ to be locally an edge of a polygon $\mathcal{P}$ with the interior on the side of ℓ containing L. Notice that locally P acts like a point "finger" with respect to ℓ: it can only move away from the polygon or tangentially to it. Figure 1.5 shows an example of our construction: a rectangle and four point fingers which lie on its boundary. The vertices marked by a diamond are fixed on the plane. In fact polygon $\mathcal{P}$ will be fixed on the plane, and the set of immobilizing points will be attached to the 'P' vertices of the mechanisms, and will be free to move (subject to the finger restrictions). These vertices also need be held in a rigid configuration with respect to each other by a framework representing the fact that in the original problem, they are fixed on the plane. This is easily accomplished by triangulating the set of vertices. This triangulation is rigid, and is shown by long thin lines. We have thus established that the framework given by this construction is rigid if and only the polygon is immobile.

Theorem 1.6.1. *Finite immobility in* $\mathbb{R}^2$ *can be reduced to planar rigidity.*

Note that if n is the number of immobilizing points, there are $9n + 1$ bars and $5n$

movable vertices and $2n$ vertices pinned on the plane.

The same construction does not establish a similar connection between force/torque closure and infinitesimal rigidity, because even if the P vertices are pinned down, the framework is not infinitesimally rigid due to the Q' vertices.

Finally let us note that it seems that neither purely first and second order frictionless contact analysis, nor first order frictional analysis are sufficient to completely understand the issue of immobilizing an object. In the classical example of a triangle with three fingers at the intersection with its inscribed circle, first order immobility requires friction to be successful, but second order immobility is guaranteed [RB94]. On the other hand, appropriately placing three fingers on two parallel edges (of a rectangle) also produces first order immobility with friction, but never second order immobility.

1.6.5 Immobilizing With Clamps

A variation of finite immobility was defined in the context of grasping with a *clamp*, which is a geometric idealization of a parallel jaw gripper with bounded jaw width. The definition of a clamp was made to model the "real world" gripper. Friction is allowed, but only to prevent an object from sliding in a direction parallel to the gripper. Several results exist.

Souvaine and Van Wyk [SV94], define a *stable clamp* to be a position of the gripper which prevents the object from rotating, and a small change in the position of the gripper maintains this property. These clamps usually contact the polygon at three or more points, the requirement being roughly that one of the points lies on one jaw and projects perpendicularly to the jaws onto the segment spanned by the two other points. They consider infinitesimally small gripper widths, and show that convex polygons, and certain polygons composed of convex chains can be clamped. Capoyleas [Cap93] extends their work to arbitrarily wide jaws, and some more types of polygons.

Albertson, Haas and O'Rourke [AHO90] define a slightly more general type of clamp and show that yet other classes of polygons are clampable using their definition. It is still a conjecture whether any polygon can be clamped under any of the above-mentioned definition.

Bose, Bremner, and Toussaint [BBT94] extend the definition of clamp to three dimensions and show that all convex polyhedra can be clamped with a parallel jaw gripper of sufficiently large size. They also provide an algorithm to find all such clamps for a n sided convex polytope in $O(n+k)$ time, where k is the number of antipodal "features", a feature being a vertex, edge or facet of the polytope.

1.6.6 Finger Gaiting

Once an object is grasped, certain tasks require the object to be manipulated, for example rotated [CGRW94]. Such rotations can be accomplished as in the work of Rus [Rus92]: fingers slide on the object and cause it to rotate. Here very little sensory information is used, providing a robust algorithm. A geometric planning phase takes $O(n \log n)$ time for an n sided polygonal object.

Another fundamentally different approach consist of "walking" on the object, a process called *finger gaiting*. It consist of going from a grasp to another, lifting one finger at a time, and maintaining equilibrium (closure) at all times.

Finding such a gait is non-trivial, and requires motion planning. See for example the book by Latombe [Lat91] for a description of motion planning. Hong, Laferriere, Mishra and Tan [HLMT90], used topological techniques to show that for smooth objects, there exist two and three finger gaits.

More recently, Chen and Burdick [CB93] study the space of configurations of points on the surface of an object forming a closure grasp, for the purpose of gaiting. This space is parametrized by the object surface positions, and they study the surfaces formed in this space by configurations corresponding to closure grasps for an elliptic object.

Here we consider the situation in wrench space to find valid positions for fingers. Assume we are given a hand with one more frictionless finger than is necessary for obtaining closure. Say this number is $k+1$. We now wish to manipulate the object B, and one of the fingers, say the finger at $\mathbf{p}_1$, call it F_1, is about to reach a limit imposed by the mechanism or force constraints. We wish to replace it with the free finger. Where can we place the free finger on ∂B so that closure is guaranteed when F_1 is removed?

If the grasp G is still a closure grasp without F_1, we are done. However this does not occur in general, so we assume that this is not the case. Let $G' = G \setminus \{\mathbf{p}_1\}$.

Define the set of wrenches, which when added to $\Gamma(G')$, guarantee closure:

$$C = \left\{\mathbf{w} \in \mathbb{R}^6 : \text{pos}\ (\{\mathbf{w}\} \cup \Gamma(G')) = \mathbb{R}^6\right\}.$$

We wish to find H, the points on ∂B which correspond to C:

$$H = \Gamma^{-1}(\Gamma(\partial B) \cap C).$$

But C is nothing but $-$int pos $\Gamma(G)$. To show this, let us show a geometric lemma without the baggage of wrench maps:

Lemma 1.6.2. *Let $S \subseteq \mathbb{R}^d$,* pos $S \neq \mathbb{R}^d$*. Then*

$$C = \{x \in \mathbb{R}^d : \text{pos}\ (x, S) = \mathbb{R}^d\} = -\text{int pos}\ S.$$

Proof. If $x \in -$int pos S, there is an $\epsilon > 0$ such that $-x + y \in$ pos S for all y with $\|y\| = \epsilon$. Then $x - x + y \in$ pos $\{x\} \bigcup S$ *i.e.* $y \in$ pos $\{x\} \bigcup S$, and hence pos $\{x\} \bigcup S = \mathbb{R}^d$.

Conversely, if pos $\{x\} \bigcup S = \mathbb{R}^d$, then for any $y \in \mathbb{R}^d$, $y = \sum_{i>0} \lambda_i s_i + \lambda_0 x$, for some $0 \leq \lambda_i$, $\lambda_0 \neq 0$, $s_i \in S$. Then $-x + \frac{y}{\lambda_0} \in$ pos S which implies $x \in -$int pos S, as required. □

In our application, $d = 6$, $S = \Gamma(G')$ and the set C is a polyhedral convex cone with one bounding ray per finger. Let us assume now the number of fingers is constant. A point of ∂B can be easily tested for being a valid placement for the free finger in constant time. When B is polyhedral, we have seen in Theorem 1.3.3 that $\Gamma(\partial B)$ is a collection of planar polygons in $\mathbb{R}^6$. Thus we can intersect C with $\Gamma(\partial B)$ in time proportional to the size of B. We obtain a set of polygons on the faces of B. If the free finger is placed in one of these polygons it guarantees, along with the remaining $k - 1$ fingers, force/torque closure.

1.7 Reactive Grasping

The physics–based models described so far assume that one can apply a variable force at the fingers that can be controlled. This implies sensors that can determine the force applied. More importantly, we assume complete knowledge of the object to be grasped. These are rather strong requirements, and research has been done to relax them.

One possibility is to devise a mechanism with no sensors, which is guaranteed to grasp the object without any knowledge of the object's shape. This has been done for example by Goldberg and Furst [GF93]. We take an intermediate approach. We endow a parallel jaw gripper or a three finger hand with very simple sensors, and allow the sensors to "guide" the hand toward a grasp. Our approach is non-disturbing, guarantees a grasp is found, and requires no global knowledge.

1.8 Other issues

So far, we have only discussed the static problems (i.e. grasping) in dexterous manipulation. There are still a large number of exciting and challenging questions related to the dynamic aspects of dexterous manipulation. Examples of such questions include fine-manipulation of objects by continuous finger-gaiting [HLMT90], dexterous adjustment of grasp configuration [LCS89, Rus92], prehension strategy and motion planning [SHS86, SY87] and controlled motion of grasped object [LHS89, Sil93].

A variety of issues involving high-level task planning remain largely unexplored. A thorough understanding of these questions are crucial to bring the science of dexterous manipulation into practice.

Chapter 2

Efficiency of a Closure Grasp

The techniques of Mishra *et al.* [MSS87] yield a method to synthesize at least *one* closure grasp of an object. However, in the absence of any measure of "goodness" for closures grasps, the synthesized grasp may not be very robust, thus, not useful in practice. This has motivated further research that attempts to quantify the goodness of closure grasps and to synthesize provably good closure grasps [KMY92, Mis94].

One criterion for goodness is the "efficiency" of a grasp, which is the amount of external force and torque that can be resisted by applying at most a one unit of force at each grasp point. Kirkpatrick, Mishra and Yap [KMY92] have shown such an efficiency measure and derive it from a stronger *quantitative version* of Steinitz's theorem. Ferrari and Canny [FC92] propose a slightly different measure based on Minkowski sums. They also provide generalizations of both to include friction. There are some problems with this generalization however, which we discuss in Section 2.1.4. More recently, Mishra [Mis94] proposed a framework which unifies both measures. We shall describe this framework for the case of a positive grip, and extend it to the case of friction.

2.1 "Classical" Grasp Measures

In the description of closure grasps of Chapter 1, we have made an implicit unrealistic assumption that *the magnitudes of finger forces are no way constrained.* In particular, it is quite likely that a force/torque closure grasp may resist any arbitrary external wrench;

but it may only do so by applying an unrealistically large force at a finger in response to a fairly small external wrench in some direction.

In order to alleviate this problem, we may assume that certain additional constraint is imposed on the magnitudes of the finger forces—the "*finger force constraint*" being expressible as

$$\begin{aligned} \chi &: \mathbb{R}^n \to \{0,1\} \\ &: (f_1, f_2, \ldots, f_n) \mapsto \begin{cases} 1, & \text{if the "constraint" holds;} \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Let $G = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ be a grasp. The set of external wrenches that can be generated by the grasp, subject to the finger force constraint, χ, is given by $\mathcal{W}_\chi$, called the "*feasible wrench set*:"

$$\mathcal{W}_\chi(,_{\mu}(\mathbf{p}_1), \ldots, ,_{\mu}(\mathbf{p}_n)) \quad = \quad \left\{\mathbf{w} = \sum_{i=1}^{n} f_i ,_{\mu}(\mathbf{p}_i) : \chi(f_1, f_2, \ldots, f_n) = 1\right\} \subseteq \mathbb{R}^6.$$

We also call $-\mathcal{W}_\chi$ the "*resistable wrench set*," the set of external wrenches that can be resisted by the grasp.

Note that if $\{,_{\mu}(\mathbf{p}_1), \ldots, ,_{\mu}(\mathbf{p}_n)\}$ forms a force/torque closure grasp and if

$$\text{pos}\left\{(f_1, f_2, \ldots, f_n) \in \mathbb{R}^n : \chi(f_1, f_2, \ldots, f_n) = 1\right\} = \mathbb{R}^n_{\geq 0}$$

then

$$\mathbf{0} \in \text{int } \mathcal{W}_\chi(,_{\mu}(\mathbf{p}_1), \ldots, ,_{\mu}(\mathbf{p}_n)).$$

Some natural finger force constraints that one may impose are of the following kinds:

- **Convex Constraint**:

$$\chi_{con} : f_1 \geq 0, \ldots, f_n \geq 0 \text{ and } \sum_{i=1}^{n} f_i \leq 1.$$

$\mathcal{W}_{\chi_{con}}$ is given by the convex hull of all the feasible wrenches:

$$\mathcal{W}_{\chi_{con}} = \text{conv}\{[0,1],_{\mu}(\mathbf{p}_i) : 1 \leq i \leq n\}.$$

For $\mu = 0$, this corresponds to the constraints considered by Kirkpatrick, Mishra and Yap [KMY92] and Ferrari and Canny [FC92]. This constraint allows us to bound on the total force applied by all the fingers.

- **Max Constraint**:

$$\chi_{max} : f_1 \geq 0, \ldots, f_n \geq 0 \text{ and } \max_{i \in \{1,\ldots,n\}} f_i \leq 1.$$

$\mathcal{W}_{\chi_{max}}$ is given by the Minkowski sum of the vectors $\Gamma_\mu(\mathbf{p}_1), \ldots, \Gamma_\mu(\mathbf{p}_n)$:

$$\mathcal{W}_{\chi_{max}} = \bigoplus_{i=1}^{n} \{[0,1]\Gamma_\mu(\mathbf{p}_i) : 1 \leq i \leq n\}.$$

This constraint has also been discussed in [FC92], albeit in a slightly different form. It corresponds to the case where each finger can apply a bounded force, independently of the others.

- **Hybrid Constraint**:
Let P_1, P_2, ..., P_l be a partition of the indices $\{1, \ldots, n\}$. Then

$$\chi_{hyb} : f_1 \geq 0, \ldots, f_n \geq 0 \text{ and } \sum_{i \in P_j} f_i \leq 1, \quad 1 \leq j \leq l.$$

$\mathcal{W}_{\chi_{hyb}}$ is given by the Minkowski sum of the convex hulls of the feasible wrenches corresponding to each partition P_j:

$$\begin{aligned} \mathcal{W}_{\chi_{hyb}} &= \bigoplus_{j=1}^{l} \operatorname{conv}\{[0,1]\Gamma_\mu(\mathbf{p}_i) : i \in P_j\} \\ &= \operatorname{conv} \bigoplus_{j=1}^{l} \{[0,1]\Gamma_\mu(\mathbf{p}_i) : i \in P_j\} \end{aligned}$$

This constraint corresponds for example to the case where we have several robot arms and wish to bound the total forces on each arm separately.

As in Chapter 1, for $\mu > 0$, we need to show that with the new constraints on the finger forces, we can still express an external wrench $\mathbf{v} \in -\mathcal{W}_\chi(\Gamma_\mu(\mathbf{p}_1), \ldots, \Gamma_\mu(\mathbf{p}_n))$ as a sum

$$\mathbf{v} = \sum_{i=1}^{n} f_i \mathbf{w}_i$$

with $\chi(f_1, f_2, \ldots, f_n) = 1$ and $\mathbf{w}_i \in\ ,_{\mu}(\mathbf{p}_i)$. That is, no two $\mathbf{w}_i$ are taken from the same $,_{\mu}(\mathbf{p}_j)$. This guarantees that we can select finger force targets within the force constraints.

For the case of χ_{con}, this follows from Theorem 1.4.3 applied to $[0, 1]$, $,_{\mu}(\mathbf{p}_i)$, $i = 1, \ldots, n$, which are convex sets by the second statement of Lemma 1.4.1. For the case of χ_{max}, this follows from the definition of Minkowski sum. Finally for χ_{hyb}, again from the definition of Minkowski sum, we can express the external wrench as a sum of vectors $\mathbf{w}_j$ from each partition j. For each such vector $\mathbf{w}_j$, we proceed as for χ_{con}: $\mathbf{w}_j$ can be expressed as a convex combination of vectors, with one vector from each $,_{\mu}(\mathbf{p}_i)$, $i \in P_j$ and we are done. We conclude that grasp strength can still be measured using the measures of this chapter, even in the presence of friction.

2.1.1 Grasp Quality Measures

For any set $X \subseteq \mathbb{R}^d$, let the *residual ball* of X refer to the maximal ball $\mathcal{B}(X)$ centered at the origin 0 such that $\mathcal{B}(X)$ is fully contained inside the convex hull of X. The *residual radius* of X, denoted $r(X)$ is the radius of this residual ball $\mathcal{B}(X)$.

Now define

$$r = r_{\chi}(,_{\mu}(\mathbf{p}_1), \ldots, ,_{\mu}(\mathbf{p}_n)) = r(\mathcal{W}_{\chi}(,_{\mu}(\mathbf{p}_1), \ldots, ,_{\mu}(\mathbf{p}_n))),$$

the radius of the largest ball centered at the origin, and inscribed in the feasible wrench set. We shall refer to this r as the residual radius of $\mathcal{W}_{\chi}$.

Then there exists an external wrench of magnitude only infinitesimally larger than r that cannot be generated or resisted by the grasp under consideration, if it must respect the finger force constraint χ. Let $\mathbf{w}$ be a point on the boundary of $\mathcal{B}(r)$ which touches $\mathcal{W}_{\chi}(,_{\mu}(\mathbf{p}_1), \ldots, ,_{\mu}(\mathbf{p}_n))$, then this external wrench simply corresponds to $(1 + \epsilon)\mathbf{w}$ for any $\epsilon > 0$. This value of r may thus be used to define a grasp measure. Note that we have [Mis94]

$$\mathcal{W}_{\chi_{con}} \subseteq \mathcal{W}_{\chi_{hyb}} \subseteq \mathcal{W}_{\chi_{max}} \subseteq n\, \mathcal{W}_{\chi_{con}}$$

and

$$r_{\chi_{con}} \leq r_{\chi_{hyb}} \leq r_{\chi_{max}} \leq n \; r_{\chi_{con}}.$$

Since the underlying geometric problem remains largely unchanged irrespective of the finger force constraint chosen, we shall mostly focus only on the simplest situation represented by the constraint χ_{con}. We also note that, for example in the frictionless case, the complexity of the Minkowski sum of the segments $\overline{\mathbf{0}, \; (\mathbf{p}_i)}$ is much larger than the convex hull of those segments since every segment contributes more than one vertex on the Minkowski sum. This Minkowski sum is known as a *zonotope* [Ede87]. In the six dimensional case, the upper bound on its complexity is $O(n^5)$, while for the convex hull, it is $O(n^3)$.

Another grasp metric was suggested by Jeff Trinkle [Tri92] and is based on the null vectors of the grip matrix. Li and Sastry [LS88] proposed yet a different metric based on alternate finger constraints. Mishra [Mis94] shows that both metrics may be misleading in some situations.

Alternatives

A variation on this theme may be obtained by considering some different "geometric object" inscribed in the feasible wrench set, $\mathcal{W}_{\chi}$. In Section 2.3 we replace the sphere by a special class of ellipsoids to circumvent a problem with the measures described so far. Another particularly interesting object to consider is $, \; (\partial B)$. This object represents the set of wrenches which a finger can generate by applying a force on B. Let $\lambda \in \mathbb{R}_{\geq 0}$ be a maximal positive real number such that

$$\lambda , \;_{\mu}(\partial B) \subseteq -\mathcal{W}_{\chi}. \tag{2.1}$$

Then it is clear that there is a point $\mathbf{p} \in \partial B$ such that if one "pushes" at the object B at the point $\mathbf{p}$ with a "*nasty finger*" with a force of magnitude only infinitesimally larger than λ, such a finger will be able to break the grasp. Thus $r_{\chi_{nasty}} = \lambda$ defines a grasp quality measure. In the frictionless case, this measure was defined by Meyer [Mey90], and is also described in [Mis94]. It can be particularly useful in the context of machining.

Here a “tool” is applied to the boundary and can apply potentially high forces on the object. Other force-torques due to acceleration and gravity are assumed to be either non-existent or negligible.

2.1.2 Some Existing Bounds

The question arises, how many fingers do we need to achieve a certain efficiency? In the frictionless case, and for χ_{con} some results are known. In fact these results are purely geometrical results, and we shall state them as such. We have already defined the residual radius r. Now let

$$\begin{aligned} r_d(m, X) &= \max\{r(Y) \colon Y \subseteq X \text{ and } |Y| \leq m\}, \\ r_d(m) &= \min\left\{r_d(m, X) \colon X \subseteq \mathbb{R}^d \text{ and } r(X) \geq 1\right\}. \end{aligned}$$

In the notation, we shall omit the subscript d, if the dimension is clear from the context. (In our application, the interesting case is $d = 6$ or $d = 3$ for planar objects.) Thus the original Steinitz's theorem can now be interpreted as saying that

$$r_d(2d) > 0.$$

A quantitative version of Steinitz's theorem provides more precise bounds for the number $r_d(m)$, when $m \geq 2d$.

In the case of the finger force constraint given by χ_{con}, the grasp measure for a closure grasp with m fingers can be expressed in terms of the residual radius values given by a quantitative Steinitz's theorem. It can be seen to be the value, $r_6(m, \text{conv}\ ,\ (\partial B))$. To see this, note that if we choose m points in $\mathcal{W}_{\chi_{con}} = \text{conv}\ ,\ (\partial B)$ with residual radius r then any external wrench vector $\mathbf{v}$ of magnitude at most r can be written as a convex combination of the m chosen points. So if $\mathbf{v}$ is any external wrench that is applied to the body B, and $\mathbf{v}$ lies in the residual ball of radius r, we can resist this external wrench by applying suitable forces (of magnitude at most 1) at the grasp points such that these forces sum to $-\mathbf{v}$; hence, we maintain the body in equilibrium. Thus, we see that the quantity $r_6(m)$ gives a measure for the quality of a closure grasp.

The special case where $m = 2d$ had been studied by Bárány, Katchalski and Pach [BKP82]; they showed that

$$r_d(2d) > \frac{c}{(2ed)^{\lfloor d/2 \rfloor} d^2}.$$

Kirkpatrick, Mishra and Yap [KMY92] have provided the following general bounds: For any set $X \subseteq \mathbb{R}^d$ whose convex hull contains the unit ball $\mathcal{B}^d$ centered at the origin, we can find a set $Y \subseteq X$ of at most m points with residual radius $r(Y)$ of at least

$$1 - 3d \left(\frac{2d^2}{m} \right)^{\frac{2}{d-1}},$$

where m is assumed to be sufficiently large.

Furthermore, let $X \subseteq \mathbb{R}^d$ be the set of all points on the surface of the d-dimensional unit ball centered at the origin. Then, every subset $Y \subseteq X$ of at most m points has a residual radius $r(Y)$, bounded from above by

$$1 - \frac{1}{17} \left(\frac{2d^2}{m} \right)^{\frac{2}{d-1}}.$$

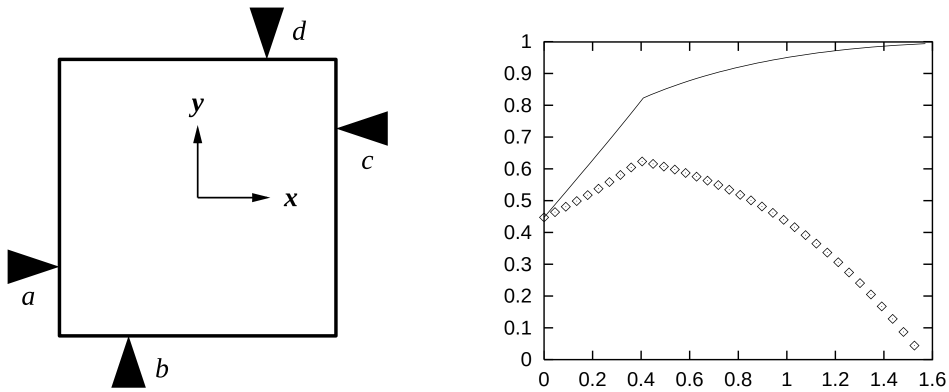
In summary, for sufficiently large m, i.e. $m \geq 13^d d^{(d+3)/2}$,

$$1 - 3d \left(\frac{2d^2}{m} \right)^{\frac{2}{d-1}} \leq r_d(m) \leq 1 - \frac{1}{17} \left(\frac{2d^2}{m} \right)^{\frac{2}{d-1}}.$$

This indicates an interesting *trade-off* between efficiency of a closure grasp and the number of fingers. These results seem to indicate that a twelve-finger positive grip, while sufficient to provide a closure grasp, may not be adequate to achieve a desirable efficiency—at least when friction is not considered.

Meyer's Result

Another negative result has been shown by Meyer [Mey90]. He considers a seven finger hand with frictionless contacts and an upper bound of 1 on magnitude of the forces it can apply. He shows that there are objects, which depend on an $\epsilon > 0$, for which $r_{nasty} \leq 5\sqrt{2}\epsilon$. The objects in question are rectangular parallelipipeds with one side of unit length and two smaller sides of length ϵ. In effect, Meyer's efficiency measure takes into account the geometry of the object.



(a) Grasp G_4

(b) Its strength (continuous curve)

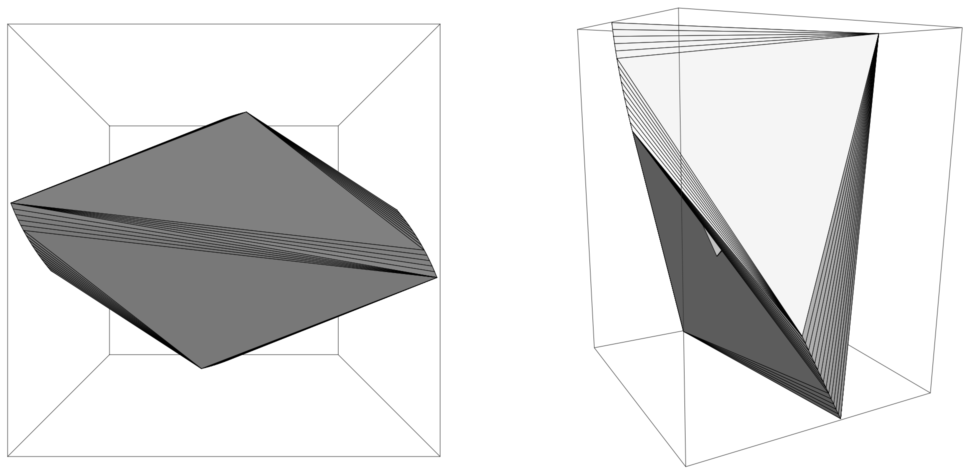
Figure 2.1: A four finger grasp and its strength as a function of the friction angle.

2.1.3 The Case of Friction

Allowing friction at the finger contacts causes the set of applicable forces to increase. Thus it is not surprising that grasp strength, as measured by the measures described earlier also increases. In fact some grasps which are not closure without friction, become closure grasps for a sufficiently large coefficient of friction. This is the case for example of two fingers placed at antipodal points on a planar object. Figure 2.1 shows a four finger closure grasp in the plane, call it G_4, and a graph showing grasp strength as measured by r_{con} as a function of the *friction angle*, the maximum angle a force can deviate from the object normal at the contact point, while maintaining contact. Corresponding to this grasp G_4, we also show in Figure 2.2 two views of the set conv , (G_4). The 'top' view is from a point on the positive torque axis.

Approximating Friction cones

The main difficulty in computing these measures, not to mention performing grasp optimization based on them, is that conv , $_\mu(\mathbf{p})$ has non-linear boundary for $\mu > 0$. It is customary in the robotics literature to approximate the boundary of the friction cone by



(a) Top view (b) Side view

Figure 2.2: The grasp of Figure 2.1 for friction coefficient 0.2.

a polyhedral cone [PSS+95]. This is sufficient for detecting the presence or absence of closure. For the purpose of computing grasp strength, this is not sufficient except for small coefficient of friction, and causes the grasp strength to be underestimated. This is discussed in Section 2.1.4.

We have already defined Γ_μ in equation (1.3) to be the image by Γ of all *unit* forces a finger can apply at a given point. This set lies on $\mathcal{S}^2 \oplus \mathbb{R}^3$ or on $\mathcal{S}^1 \oplus \mathbb{R}$ for planar objects. We can approximate this set as follows. We start by approximating the friction cone $\mathcal{C}_\mu$ as is usually done. Say we approximate a cone of angle $2\arctan(\mu)$ corresponding to a coefficient of friction of μ, by N_1 regularly spaced rays. This corresponds to approximating a unit circle by N_1 points, and by elementary geometry, we are guaranteed that this approximation contains a circle of radius $\cos\left(\frac{\pi}{N_1}\right)$. Thus the approximated friction cone is a polyhedral cone, call it C_{μ,N_1} and contains a friction cone of friction coefficient $\cos\left(\frac{\pi}{N_1}\right)\mu$. For example, if we wish to loose no more than a factor of 10^{-2} (10^{-3}, 10^{-4} resp.), we need 23 (71, 223 resp.) rays.

We now must approximate the part of the truncated friction cone $\mathcal{F}_\mu$ that lies on

$\mathcal{S}^2$. This is done using the techniques of Kirkpatrick *et al.* [KMY92]. We obtain a N_2-vertex polyhedron S_{N_2} approximating a unit sphere. This polyhedron contains a smaller concentric sphere of radius $r = 1 - 162/N_2$ for sufficiently large N_2.

This polyhedron is then intersected with the approximation of the unbounded friction cone described in the previous paragraph. The complexity of the resulting polyhedron follows from a standard amortization argument: a face of C_{μ,N_1} intersects a face of S_{N_2} in only one segment, adding at most 1 vertex to that face. Thus the total complexity of the intersection is $O(N_1 + N_2)$.

This procedure produces an approximation of $\mathcal{F}$ by a set of unit vectors which is then mapped to wrench space by , . The origin is also included. This produces a convex polytope of at most $O((N_1 + N_2)^3)$ vertices, as in Section 1.3. For the planar case, all that is needed is to approximate a planar arc on $\mathcal{S}^1 \times \mathbb{R}$. The plane of this arc goes through the origin. Let ϕ be the half-angle of the friction cone. With N equally spaced points, we get an approximation which looses a factor in the possible generated force of no more than $\cos \frac{\phi}{2N}$.

Generating Force Targets

Consider an external wrench $\mathbf{v}$ being applied to B which is being grasped by m fingers with frictional contacts. Mishra, Schwartz and Sharir [MSS87] give a linear time procedure that given a set W of vectors $\mathbf{w}_i = , (\mathbf{p}_i)$ in $\mathbb{R}^6$, $i = 1, \ldots, m$, finds a subset of 6 of them such that $\mathbf{v}$ can be expressed as a positive combination of those vectors. This can be used for finding force targets for frictionless fingers. The coefficients returned by their procedure however do not necessarily satisfy the finger force constraints.

We now describe a linear time procedure for finding force targets in the case of χ_{con} constraint. The problem is:

> Given $\mathbf{v} \in \mathbb{R}^6$, find λ_i, $0 \leq \lambda_i \leq 1$, $i = 1, \ldots, m$, such that $\mathbf{v} = \sum_i \lambda_i \mathbf{w}_i$ and $\sum_i \lambda_i = 1$.

One possible approach is as follows. It actually minimizes the coefficients λ_i when $\mathrm{conv}\,\{\mathbf{w}_i : i = 1, \ldots, m\}$ is non-degenerate (its facets have to more than 6 vertices). Consider the ray $\overrightarrow{\mathbf{0v}}$, and let F be the face of conv $\{\mathbf{w}_i, i = 1, \ldots, m\}$ which it intersects. Let $\mathbf{v}_F$ be the points of intersection. Then the vertices of F are points from W. If F has 6 or fewer vertices, $\mathbf{v}$ lies in the simplex conv $(\mathbf{0}, F) \subset \text{aff}(\mathbf{0}, F)$ and the force targets, *i.e.* the coefficients λ_i, will simply be the barycentric coordinates of $\mathbf{v}$ in this simplex. Finding $\mathbf{v}_F$ can be done using linear programming [MSS87, Sch86], and takes linear time [Meg84, Cla86]. Computing barycentric coordinates can be done by transforming the simplex into the canonical simplex with one vertex at the origin, and the others at vertices of the unit cube in $\mathbb{R}^k$, where k is the dimension of aff$(\mathbf{0}, F)$. The origin maps to the origin. This can be done by a linear transformation, see for example [MS71, p.16], and requires constant time. This transformation is non-singular and unique if both simplices are non-degenerate. The coordinates of the transformed version of $\mathbf{v}$ are the required coefficients.

Otherwise, F has more than 6 vertices, and we need to find a subset of them and again use barycentric coordinates. Now $\mathbf{v}_F$ lies in F, of dimension 5, which is one less than the dimension of the original space. Let us pick an arbitrary vertex of F, say $\mathbf{u}$, and consider the ray $\overrightarrow{\mathbf{u}\mathbf{v}_F}$. It intersects a facet H of dimension at most 4 at $\mathbf{v}_H$. We can now apply the above procedure in (at most) one dimension less, where $\mathbf{u}$ plays the role of the origin, and $\mathbf{v}_F$, the role of $\mathbf{v}$ in the description in the preceding paragraph. We obtain at most 5 coefficients which must be multiplied by λ_1 if $\mathbf{v}_F = \lambda_1 \mathbf{v}_H + \lambda_2 \mathbf{u}$. The entire procedure thus recursively finds at most 6 vertices of **,** $(\mathbf{p}_i)$, with appropriate coefficients, force targets in grasping terminology, in $O(m)$ time.

Note that only 6 fingers are in use at any given time. We also note that, just as the procedure of Mishra *et al.*, ours can also be generalized to arbitrary but constant dimension:

Theorem 2.1.1. *Given a set of points* $p_1, \ldots, p_m \in \mathbb{R}^d$, *with* $0 \in \text{conv}\,\{p_1, \ldots, p_m\}$, *and a point* $v \in \text{conv}\,\{p_1, \ldots, p_m\}$. *Then one can find* $\lambda_1, \ldots, \lambda_m$, *with* $0 \leq \lambda_i \leq 1$, *and* $\sum_{i=1}^m \lambda_i = 1$ *at most* d *of which are non-zero, such that* $v = \sum_{i=1}^m \lambda_i p_i$.

In the frictional case, one can use the above algorithm applied to an approximation of the friction cones as in Section 2.1.3, then "merge" any two points selected from the friction cones as per Theorem 1.4.3. This might be sufficient in practice.

2.1.4 An Alternative Approximation

A different method of approximating the image of a friction cone under the wrench map has been used in the literature for the purpose of computing grasp strength. Consider planar objects. It is possible to simply take the convex hull of image under , of the edges bounding the friction cone, as done in [FC92]. However this may cause a significant underestimate of the quality of a grasp.

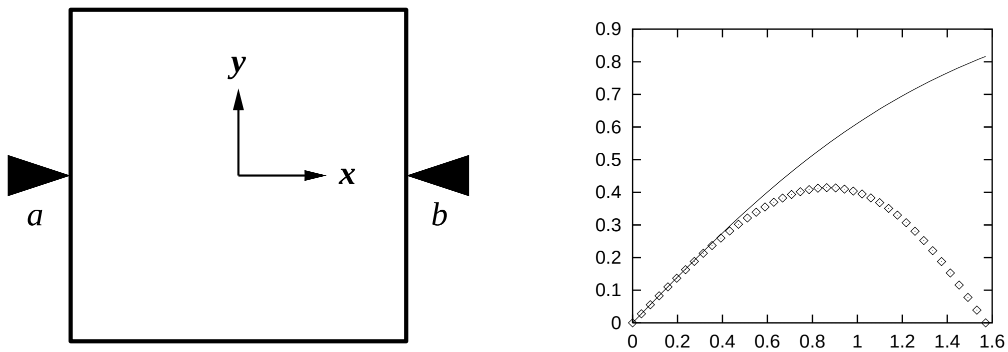
Let us consider for example the following somewhat contrived situation. A square is being grasped with four fingers placed at a, b, c, d, as in Figure 2.1. Fingers at b and d are frictionless, while finger at $a = (-2, -1)$ and $c = (2, 1)$ have a coefficient of friction μ. Let $\phi = \arctan \mu$ be the corresponding half-angle of the friction cone. The point a maps for example to two points a^+ and a^- in the wrench space. We have

$$a^{\pm} = [\cos\phi, \pm\sin\phi, \mp 2\sin\phi - \cos\phi].$$

Then as μ is increased, The projections onto the xy-plane of a^+ and a^- move close to the projection of , (d) and , (b) and the radius of the largest ball centered at the origin inside conv $\{a^{\pm}, b^{\pm}, , (c), , (d)\}$ decreases for sufficiently large angle ϕ. This is illustrated by the dotted curved in Figure 2.1(b). Thus for sufficiently large friction coefficient, increasing friction actually decreases grasp quality according to this simplified measure. A different example is as follows. Again, a square is being grasped but with two fingers placed at $a = (-1, 0), b = (1, 0)$, as in Figure 2.3(a). Both fingers benefit from coefficient of friction of μ. Again, the grasp quality decreases for very large friction coefficient, as is shown by the dotted curve in Figure 2.3(b). The continuous curve represents $r_{\chi_{con}}$. However the approximation is reasonably accurate for small μ, at least in the planar case.

2.1.5 Computation of the Measures

In the frictionless case, to compute r_{con} for a given grasp G, we simply compute conv , (G) and find the facet that is closest to the origin. If m is the number of fingers, this takes time proportional to the size of conv , (G), which is $O(m^3)$ time for spacial objects, $O(m)$



(a) Two finger grasp

(b) Its strength as a function of angle

Figure 2.3: A pessimistic approximation of friction cones (dotted curve.)

for planar objects as described in Section 1.3. For the frictionless case, one can use the approximation outlined above and replace m by the number of points generated by this approximation. See Chapter 4 for details on the computation of the r_{nasty} measure.

We have implemented the residual radius measuring algorithms by generating an approximation of $\Gamma_\mu(G)$, using the QHULL library [BDH93] of the Geometry Center, University of Minnesota. It is used to compute the convex hull of this approximation, and finally traversing the facets of the convex hull and finding the one closest to the origin. This also allows us to display the situation in wrench space, for planar objects. The program GEOMVIEW also from the Geometry Center, was used for this purpose, and to generate Figure 2.2.

2.2 Measures for Assembly Fixturing

We now introduce new grasp efficiency measures for the case of several objects. Informally the goal is again to keep the boundary of conv $\Gamma_*(G)$ bounded away from $\mathbf{0}$ for all objects. In fact the grasp efficiency measures of Section 2.1 generalize quite naturally to this case, but we work in dimension $6k$ (or $3k$ for planar objects) instead of $d = 6$ (or 3 in the planar case). The definition of residual radius still applies, we use r_{6k} (r_{3k}). The corresponding optimization problems of finding optimal grasps with respect to this measure in any dimension, discussed in Chapter 4, are **MaxScale-B** and **MinCover-B**.

For the nasty finger measure, for the inscribed set, we simply take the direct sum of all the conv $\Gamma(\partial B_i)$ for each polyhedral object. This is a special case of taking Minkowski sums of polytopes, and the result is still convex [Grü67]. Alternately, if fixtures can be placed at only a finite number of points on B_i, one can replace B_i by this set of points. The corresponding optimization problems of Chapter 4, are **MaxScale-P** and **MinCover-P**.

Finally we observe that a ball in high dimension is rather "small", and we might prefer to consider the direct sum of (5 dimensional) balls of identical radius r (which we call $\mathcal{B}(r)$), one for each object, producing $\mathcal{B}^k(r)$. This measure applied to a fixture set G is then the largest scaling factor λ such that $\lambda\mathcal{B}^k(1) \subset \text{conv}\,\Gamma_*(G)$ and the corresponding optimization problems are **MaxScale-B**^{k} and **MinCover-B**^{k}. The application at hand will determine which measure will be used.

We describe the computation of these measures in the general dimensional case in Chapter 4.

2.3 A Grasp Measure Invariant Under Rigid Motions.

The grasp efficiency measure described in the previous sections is given by the radius of the largest ball centered at the origin and inscribed in the convex hull of the wrenches $\Gamma(\mathbf{p}_i)$. It has the advantage that the radius gives a lower bound on how large external wrenches the given grasp can resist. However the facets of this convex hull move significantly as we change the coordinate system of the object, hence the measure changes its value. In fact the ball measure can be made arbitrarily small by translating the origin sufficiently far. This will be shown in Section 2.3.7.

Here, we study the transformations of wrench space as the coordinate system in the object space is changed, and derive a measure that is invariant under those changes.

2.3.1 Wrench Space Transformations

Let wrench $\mathbf{w}_0 = [\mathbf{f}, \mathbf{p}_0 \times \mathbf{f}] \in \mathbb{R}^6$. A translation of the origin in object space by a vector $-\mathbf{c} \in \mathbb{R}^3$ transforms $\mathbf{w}_0$ into $\mathbf{w}_\mathbf{c} = [\mathbf{f}, \mathbf{p}_\mathbf{c} \times \mathbf{f}] = [\mathbf{f}, (\mathbf{p}_0 - \mathbf{c}) \times \mathbf{f}]$, where subscripts indicate origin of the reference coordinate system.

Define $\widehat{\mathbf{c}}$ to be the antisymmetric matrix such that $\widehat{\mathbf{c}}\mathbf{x} = \mathbf{c} \times \mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^3$. We have

$$\widehat{\mathbf{c}} = \begin{pmatrix} 0 & -\mathbf{c}_z & \mathbf{c}_y \\ \mathbf{c}_z & 0 & -\mathbf{c}_x \\ -\mathbf{c}_y & \mathbf{c}_x & 0 \end{pmatrix}.$$

Furthermore, let

$$T(\mathbf{c}) = \begin{pmatrix} I_3 & 0 \\ \widehat{\mathbf{c}} & I_3 \end{pmatrix}$$

Then $\mathbf{w}_\mathbf{c} = T(-\mathbf{c})\mathbf{w}_0$.

$T(\mathbf{c})$ is the matrix of a "shear" type linear transformation, with the properties that $T(\mathbf{c})^{-1} = T(-\mathbf{c})$ and $T(\mathbf{c}_1)T(\mathbf{c}_2) = T(\mathbf{c}_1 + \mathbf{c}_2)$. Furthermore $\det T(\mathbf{c}) = 1$; it preserves volume. For planar objects, $\mathbf{w}_0 = [\mathbf{f}, (\mathbf{p}_0 \times \mathbf{f})_3] \in \mathbb{R}^3$ and we get

$$\mathbf{w}_\mathbf{c} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{c}_y & -\mathbf{c}_x & 1 \end{pmatrix} \mathbf{w}_0.$$

Let G be the set of grip points on the surface of the object to be grasped. Consider the grasp efficiency measure $r_{con}(\cdot)$ defined by the radius of the largest possible ball centered at the the origin in conv $\Gamma(G)$. Here it will be denoted by $r_{(0)}(\cdot)$ to emphasize the coordinate system. We now allow the origin in object space to change, and we wish to find the largest of such balls of radius $r_{(\mathbf{c})}(G)$ among all possible translations of the object coordinate system by $\mathbf{c}$. These translations modify the wrench space by $T(\mathbf{c})$. The radius $r^*(G) = \sup_\mathbf{c} r_{(\mathbf{c})}(G)$ of this optimal ball will be independent of such translations, by definition.

Instead of transforming the facets of the polytope conv $\Gamma(G)$ (which remain linear), we can transform the sphere: we get an ellipsoid of a certain class.

Let $\mathbf{w}_0 \in \mathbb{R}^6$ satisfy the equation of the ball of radius r centered at the origin:

$$\mathbf{w}_0^T \mathbf{w}_0 = r^2$$

then $\mathbf{w} = T(-\mathbf{c})\mathbf{w}_0$ satisfies

$$\mathbf{w}^T T(\mathbf{c})^T T(\mathbf{c})\mathbf{w} = r^2.$$

Let

$$Q_{\mathbf{c}} = \begin{pmatrix} I_3 + \widehat{\mathbf{c}}^T \widehat{\mathbf{c}} & \widehat{\mathbf{c}}^T \\ \widehat{\mathbf{c}} & I_3 \end{pmatrix} = T(\mathbf{c})^T T(\mathbf{c}),$$

then

$$\mathbf{w}^T Q_{\mathbf{c}} \mathbf{w} = r^2,$$

which is the equation of an ellipsoid. Note that $Q_{\mathbf{c}}$ is symmetric and positive definite, and has determinant 1. Let $E(\mathbf{c}, r)$ be the above ellipsoid in $\mathbb{R}^6$. Call r the *radius* of the ellipsoid $E(\mathbf{c}, r)$, and let $Q_{\mathbf{c}}$ be the matrix associated with $E(\mathbf{c}, r)$. Let ν_d be the volume of the unit sphere in $\mathbb{R}^d$, $\nu_d = \pi^{d/2}/, (d/2+1)$ [Cox73]. (Here , $(\cdot)$ is the Euler gamma function.) Then r^6 is the volume of the ellipsoid, divided by ν_6. $E(\mathbf{c}, r)$ is a sphere of radius r if and only if $\mathbf{c} = \mathbf{0}$. The eigenvalues of $Q_{\mathbf{c}}$ are

$$1,\ 1 + \frac{\mathbf{c}^T\mathbf{c}}{2} \pm \sqrt{\left(\frac{\mathbf{c}^T\mathbf{c}}{2}\right)^2 + \mathbf{c}^T\mathbf{c}},$$

each with multiplicity two, and are all identical exactly when $\mathbf{c} = \mathbf{0}$. Since, for an eigenvalue λ and the corresponding eigenvector $\mathbf{x}$, which is one of the principal axes of the ellipsoid, $\mathbf{x}^T Q \mathbf{x} = \lambda \mathbf{x}^T \mathbf{x}$, the point $\mathbf{w} = \frac{r\mathbf{x}}{\sqrt{\mathbf{x}^T Q \mathbf{x}}}$ belongs to the ellipsoid boundary, and $\|\mathbf{w}\| = \frac{r}{\sqrt{\lambda}}$. Thus the largest eigenvalue corresponds to the shortest principal direction, and is the radius of the largest sphere centered at the origin, inscribed in the ellipsoid. Figure 2.4 shows this ellipsoid in the planar case for $\mathbf{c} = (0, 1)$.

We now introduce rotations of the object coordinate system. A fully general unit determinant orthogonal transformation of the object coordinate system also gives a linear transformation on the wrench space, which is defined by the following matrix [MLS94]:

$$U(\mathbf{c}) = \begin{pmatrix} R & 0 \\ \widehat{\mathbf{c}}R & R \end{pmatrix}$$

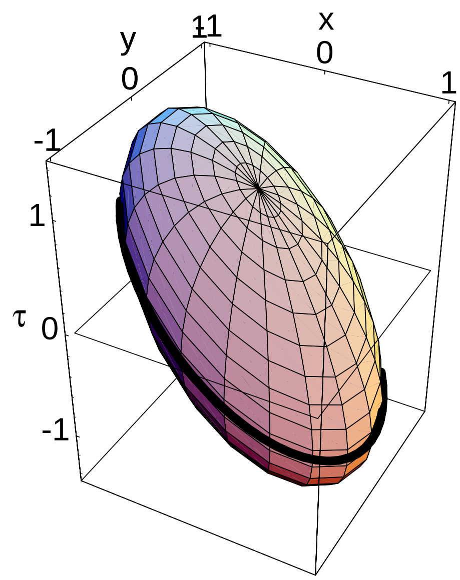


Figure 2.4: $E((0,1),1)$ with the image of the unit circle in the xy-plane highlighted.

where R^{-1} is the rotation component of the transformation with $R^T = R^{-1}$. The matrix associated with the ellipsoid becomes:

$$U(\mathbf{c})^T U(\mathbf{c}) = \begin{pmatrix} I_3 + R^T \hat{\mathbf{c}}^T \hat{\mathbf{c}} R & R^T \hat{\mathbf{c}}^T R \\ R^T \hat{\mathbf{c}} R & I_3 \end{pmatrix} = \begin{pmatrix} I_3 + \hat{\mathbf{d}}^T \hat{\mathbf{d}} & \hat{\mathbf{d}}^T \\ \hat{\mathbf{d}} & I_3 \end{pmatrix} = Q_{\mathbf{d}}$$

where $\mathbf{d} = R^T \hat{\mathbf{c}} R = \widehat{R^T \mathbf{c}}$. Hence adding rotations does not change the class of ellipsoids.

Let $r^*(G)$ be the radius of the largest ellipsoid of the above type inside conv $\Gamma(G)$:

$$r^*(G) = \sup_{\mathbf{c}} r_{(\mathbf{c})}(G).$$

We will define the new grasp quality measure *invariant under rigid motions of the coordinate system* by $r^*(G)$. It is the radius of the largest ellipsoid of the above type inside the set of feasible wrenches. It is an absolute measure.

This definition however has the consequence that the 'best' object space origin for measuring a grasp is dependent on the grasp itself.

2.3.2 Scaling

Scaling of the object coordinate system can create arbitrarily small or large torques. The wrench space transformation matrix for a scaling by s is

$$\begin{pmatrix} I_3 & 0 \\ 0 & sI_3 \end{pmatrix}.$$

When followed by a general rigid transformation, we get

$$S(\mathbf{c}) = \begin{pmatrix} R & 0 \\ \widehat{\mathbf{c}} & sR \end{pmatrix}$$

and the ellipsoid $E(\mathbf{c}, r, s) = \{\mathbf{x} \in \mathbb{R}^6 : \mathbf{x}^T Q_{\mathbf{c},s}\mathbf{x} = r^2\}$ has the matrix:

$$\begin{pmatrix} I_3 + \widehat{\mathbf{c}}^T\widehat{\mathbf{c}} & s\widehat{\mathbf{c}}^T \\ s\widehat{\mathbf{c}} & s^2 I_3 \end{pmatrix} = Q_{\mathbf{c},s}.$$

with determinant s^6. In $\mathbb{R}^3$, the corresponding determinant is s^2. The volume of this ellipsoid in $\mathbb{R}^6$ is [GLS88, p.67]

$$\nu_6 r^6/\sqrt{\det Q_{\mathbf{c},s}} = \nu_6 r^6/s^3.$$

Unfortunately the volume $\overline{r}^*(G)$ is not bounded below, but can be made arbitrarily small because of the scaling. We can fix that by setting

$$\overline{r}^*(S) = \sup_{\mathbf{c},s} r_{\mathbf{c},s}(S)s^k.$$

where $k = 3$ for spacial objects, 1 for planar objects. It not clear, however, how to balance the optimization with respect to both $\mathbf{c}$ and s, and we shall not consider scaling further.

2.3.3 Properties of $E(\mathbf{c}, r)$.

We concentrate here on rigid motions of the object coordinate system. We will need a useful lemma, which is a slight extension of a result from [GLS88, p.68].

Lemma 2.3.1. *Let $a \in \mathbb{R}^d \setminus \{0\}$ and $x \in \mathbb{R}^d$. The linear objective function $a^T x$ over the ellipsoid $x^T Q x = r^2$ is maximized for $x = \frac{r}{\sqrt{a^T Q^{-1} a}} Q^{-1} a$. The maximal value of $a^T x = r\sqrt{a^T Q^{-1} a}$.*

Also an easy calculation shows that $\nabla_{\mathbf{w}} \mathbf{w}^T Q \mathbf{w}\Big|_{\mathbf{w}_0} = 2Q\mathbf{w}_0$; this is a vector normal to the ellipsoid at $\mathbf{w}_0$, provided $\mathbf{w}_0^T Q \mathbf{w}_0 = r^2$.

Since the force space is not changed by rigid motions, the projection of $E(\mathbf{c}, r)$ onto the space of forces is a sphere in this space. Indeed, consider the points $\mathbf{w} \in \mathbb{R}^6$ on the unit radius ellipsoid which maximize $\mathbf{a}^T \mathbf{w}$, where $\mathbf{a}$ is a unit vector of the form $[\mathbf{u}^T, \mathbf{0}]^T$ in the force space, hence $\mathbf{u}$ is also unit. Furthermore it is easy to verify that

$$Q_{\mathbf{c}}^{-1} = \begin{pmatrix} I_3 & \widehat{\mathbf{c}} \\ \widehat{\mathbf{c}}^T & I_3 + \widehat{\mathbf{c}}^T \widehat{\mathbf{c}} \end{pmatrix}. \tag{2.2}$$

It is convenient to note here that $Q_{\mathbf{c}}^{-1} = T(-\mathbf{c})T(-\mathbf{c})^T$. Then by Lemma 2.3.1, the point on the ellipsoid extremal in direction $\mathbf{a}$ is

$$\frac{1}{\sqrt{\mathbf{a}^T Q^{-1} \mathbf{a}}} Q^{-1} \mathbf{a} = \frac{Q^{-1}[\mathbf{u}^T \mathbf{0}]^T}{\sqrt{\mathbf{u}^T \mathbf{u}}} = \begin{pmatrix} \mathbf{u} \\ \widehat{\mathbf{c}}^T \mathbf{u} \end{pmatrix}$$

and the projection of this vector onto the first three coordinates forming the force space is indeed on a unit sphere in that space.

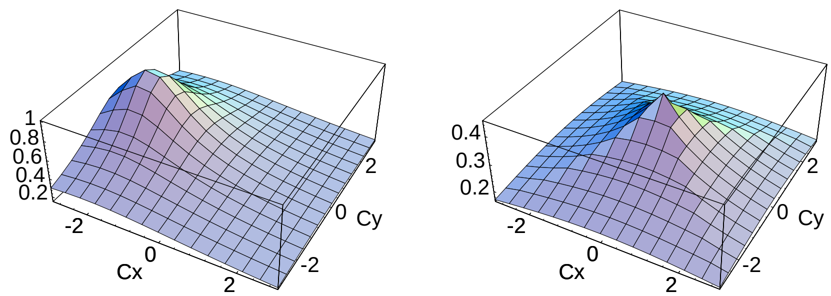
2.3.4 Largest Ellipsoid Inside a Polytope.

We start by considering the constraint that E lies in the half-space $\mathbf{a}^T \mathbf{x} \leq b$, where $b > 0$ and $\mathbf{a} \neq \mathbf{0}$. Let $Q_{\mathbf{c}}$ be the matrix of $E(\mathbf{c}, r)$. The extremal point on $E(\mathbf{c}, r)$ in the direction of the normal $\mathbf{a}$ defining the bounding hyperplane must satisfy $\mathbf{a}^T \mathbf{x} \leq b$, hence by Lemma 2.3.1

$$r\sqrt{\mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a}} \leq b, \text{ or } r \leq \frac{b}{\sqrt{\mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a}}}.$$

Define for a hyperplane $\mathbf{a}^T \mathbf{x} \leq b$,

$$r_{\mathbf{a},b}(\mathbf{c}) = \frac{b}{\sqrt{\mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a}}}. \tag{2.3}$$



(a) Constraint due to the facet $(0, 2, 1)^T\mathbf{x} = 1$.

(b) Constraint for grasp G_4.

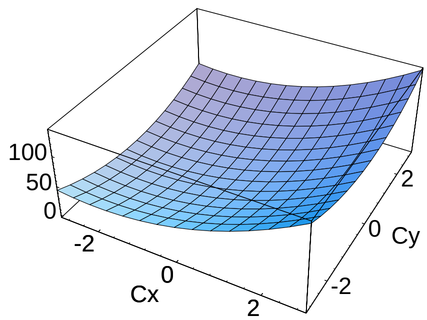
Figure 2.5: Facet constraints on r.

One such constraint is illustrated in Figure 2.5(a). From the facets of conv **,** (G) we obtain a set of constraints, in the variables $\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z$, which r must satisfy. Thus to compute r^*, we need to find the largest r subject to these these constraints. Our measure will be defined by $r^* = \sup_{\mathbf{c}} \min_{\mathbf{a},b} r_{\mathbf{a},b}(\mathbf{c})$, where $\mathbf{a}, b$ range over all hyperplanes bounding conv **,** (G) (or also conv **,** (∂B) for polyhedral B). The set of constraints for the grasp of Figure 2.1(a) is shown in Figure 2.5(b). It is not surprising that the "best" center is at the center of symmetry of this example.

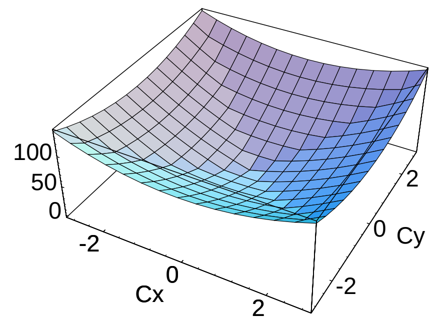
We show below that this is an optimization problem with linear objective function r and quadratic constraints, with variables $r, \mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z$, solvable in expected linear time by the Generalized Linear Programming techniques of [SW92]. We also show that this problem can be formulated as a Semidefinite Program.

2.3.5 Convexity

So far we have not shown that the measure is well defined. This will be done in this section and the next section. Although the ellipsoid of the preceding type of maximal volume is not unique, at least in the 6 dimensional case, its volume is unique. In fact the



(a) $r'_{(0,2),1}$



(b) r' for grasp G_4.

Figure 2.6: Convex versions of the constraints of Figure 2.5.

values of $\mathbf{c} = (\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z)$ corresponding to those ellipsoids forms a line segment in the parameter space and we can always select the solution which is closest to the origin. We will show that the set of constraints imposed by the halfplanes of the enclosing polytope is convex. This is not true if we consider r as the function to be optimized. But minimizing r amounts to maximizing $1/r^2$ since $r > 0$. Let

$$r'_{\mathbf{a},b}(\mathbf{c}) = 1/r^2_{\mathbf{a},b}(\mathbf{c}) = \mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a}/b^2. \tag{2.4}$$

The new constraints will then be of the form $r \geq \mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a}/b^2$, and these turn out to be convex. Figure 2.6 shows the convex constraints corresponding to the original constraints of Figure 2.5. In the following we consider only one facet and we will write $r(\mathbf{c})$ (resp. $r'(\mathbf{c})$) for $r_{\mathbf{a},b}(\mathbf{c})$ (resp. $r'_{\mathbf{a},b}(\mathbf{c})$).

We now show that the constraints (2.4) are convex.

Lemma 2.3.2. *For any* $\mathbf{a} \in \mathbb{R}^6$, $0 \leq \lambda \leq 1$, $\mathbf{c}, \mathbf{d} \in \mathbb{R}^3$,

$$\mathbf{a}^T Q^{-1}_{\lambda\mathbf{c}+(1-\lambda)\mathbf{d}} \mathbf{a} \leq \lambda \mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a} + (1-\lambda) \mathbf{a}^T Q_{\mathbf{d}}^{-1} \mathbf{a}.$$

Proof. Let $\mathbf{a} = [\mathbf{u}^T \mathbf{v}^T]^T$, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$. Then using (2.2), and the fact that for any 3 by 3 matrix M, $\mathbf{v}^T M \mathbf{u} = \mathbf{u}^T M^T \mathbf{v}$, we have

$$\begin{aligned}
&\lambda \mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a} + (1-\lambda) \mathbf{a}^T Q_{\mathbf{d}}^{-1} \mathbf{a} - \mathbf{a}^T Q_{\lambda \mathbf{c} + (1-\lambda)\mathbf{d}}^{-1} \mathbf{a} \\
&= \lambda \left[\mathbf{u}^T \mathbf{u} + \mathbf{v}^T \mathbf{v} + \mathbf{v}^T \widehat{\mathbf{c}}^T \widehat{\mathbf{c}} \mathbf{v} + 2 \mathbf{u}^T \widehat{\mathbf{c}} \mathbf{v} \right] \\
&\quad + (1-\lambda) \left[\mathbf{u}^T \mathbf{u} + \mathbf{v}^T \mathbf{v} + \mathbf{v}^T \widehat{\mathbf{d}}^T \widehat{\mathbf{d}} \mathbf{v} + 2 \mathbf{u}^T \widehat{\mathbf{d}} \mathbf{v} \right] - \mathbf{u}^T \mathbf{u} - \mathbf{v}^T \mathbf{v} \\
&\quad - \mathbf{v}^T (\lambda \mathbf{c} + (1-\lambda)\mathbf{d})^T (\lambda \mathbf{c} + (1-\lambda)\mathbf{d}) \mathbf{v} - 2 \mathbf{u}^T (\lambda \mathbf{c} + (1-\lambda)\mathbf{d}) \mathbf{v} \\
&= (\lambda - \lambda^2) \mathbf{v}^T \widehat{\mathbf{c}}^T \widehat{\mathbf{c}} \mathbf{v} + ((1-\lambda) - (1-\lambda)^2) \mathbf{v}^T \widehat{\mathbf{d}}^T \widehat{\mathbf{d}} \mathbf{v} - \\
&\quad 2\lambda(1-\lambda) \mathbf{v}^T \widehat{\mathbf{c}}^T \widehat{\mathbf{d}} \mathbf{v} \\
&\geq 0.
\end{aligned}$$

The inequality follows from the fact that $2\mathbf{v}^T \widehat{\mathbf{c}}^T \widehat{\mathbf{d}} \mathbf{v} \leq \mathbf{v}^T \widehat{\mathbf{c}}^T \widehat{\mathbf{c}} \mathbf{v} + \mathbf{v}^T \widehat{\mathbf{d}}^T \widehat{\mathbf{d}} \mathbf{v}$. □

By replacing $\mathbf{a}$ by $\mathbf{a}/b$ in the lemma, we obtain the desired result:

Corollary 2.3.3.

$$r'(\lambda \mathbf{c} + (1-\lambda)\mathbf{d}) \leq \lambda r'(\mathbf{c}) + (1-\lambda) r'(\mathbf{d})$$

and hence the feasible set determined by the set of constraints on $\mathbf{c}, r$ *is convex.*

2.3.6 Uniqueness

Even though the largest ellipsoid $E(\mathbf{c}, r)$ is not unique, its volume is. We can show this by computing the gradient of the function $r'_{\mathbf{a},b}(\mathbf{c})$ as a function of $\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z$. Let $D(\mathbf{x}) = \nabla_{\mathbf{c}} \mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a} \Big|_{\mathbf{x}}$. Then $\{\mathbf{x} \in \mathbb{R}^3 : D(\mathbf{x}) = 0\}$ is the set of points $\mathbf{c}$ for which the function r' has local minima since r' is convex. Let $\mathbf{a} = [\mathbf{u}^T, \mathbf{v}^T]^T$ with $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{u} = [u_1, u_2, u_3], \mathbf{v} = [v_1, v_2, v_3] \in \mathbb{R}^3$. Then as before

$$\mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a} = \mathbf{u}^T \mathbf{u} + \mathbf{v}^T \mathbf{v} + \mathbf{v}^T \widehat{\mathbf{c}}^T \widehat{\mathbf{c}} \mathbf{v} + 2 \mathbf{u}^T \widehat{\mathbf{c}} \mathbf{v}. \tag{2.5}$$

Clearly, if $\mathbf{v} = \mathbf{0}$ the above expression and hence the optimal volume is constant over all $\mathbf{c}$. Assume therefore $\mathbf{v} \neq \mathbf{0}$. Then a simple calculation shows that

$$D(\mathbf{x}) = \begin{pmatrix} v_2^2 + v_3^2 & -v_1 v_2 & -v_1 v_3 \\ -v_1 v_2 & v_1^2 + v_3^2 & -v_2 v_3 \\ -v_1 v_3 & -v_2 v_3 & v_1^2 + v_2^2 \end{pmatrix} \mathbf{x} - \begin{pmatrix} v_3 u_2 - v_2 u_3 \\ -v_3 u_1 + v_1 u_3 \\ v_2 u_1 - v_1 u_2 \end{pmatrix}.$$

The above matrix has determinant 0, rank 2, and a basis for its null space is $\mathbf{v}$. The latter is clear from the fact that $(\mathbf{c} + t\mathbf{v}) \times \mathbf{v} = \mathbf{c} \times \mathbf{v}$ for all $t \in \mathbb{R}$.

The rank can be shown by an elementary calculation. Consider the first two rows and assume that the first row is k times the second row, for some $k \neq 0$, and that $v_3 \neq 0$. (If $v_3 = 0$, the last two rows are clearly linearly independent.) Then from the third column, $v_1 = k v_2$. From the first column, we get $v_2^2 + v_3^2 = -k^2 v_2^2$. Substituting in the equation obtained from the second column we get

$$-k v_2^2 = k^3 (v_2^2 + v_3^2) = -k^5 v_2^2,$$

which implies $v_2^2 = 0$. But this in turn implies $v_1 = 0$ and $v_3 = 0$, a contradiction to our hypothesis. The rank of the matrix must therefore be 2.

Hence all the solutions of the system $D(\mathbf{x}) = 0$ differ by a multiple of $\mathbf{v}$ and by the preceding observation $\mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a}$ remains constant, and so does $r_{\mathbf{a},b}$.

2.3.7 Comparison with the Residual Radius Measure

We now have the tools to compare the ellipsoid based measure to the ball based measure. We show here that if the origin is translated sufficiently far, one can make the ball measure as small as desired.

Let $t > 0$ be a real parameter and consider a facet $\{\mathbf{x} : \mathbf{a}\mathbf{x} \leq b\}$. Let $\mathbf{a} = [\mathbf{u}^T \mathbf{v}^T]^T$ with $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$. To show that $r_{\mathbf{a},b}$ can be made arbitrarily small, by equation (2.3), it is sufficient to show that $\mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a}$ can be made arbitrarily large. We exhibit a direction of translation which accomplishes this for most facets. Let $\mathbf{c} = t\mathbf{v} \times \mathbf{u}$, and let $\mathbf{x} = \mathbf{c} \times \mathbf{v}$. Equation (2.5) becomes

$$\mathbf{a}^T Q_{\mathbf{c}}^{-1} \mathbf{a} = \text{const} + \mathbf{x}^T \mathbf{x} + 2\mathbf{u}^T x$$

in which the second term is a positive multiple of t^2 and

$$\begin{aligned}\mathbf{x} &= t(\mathbf{v}\times\mathbf{u})\times\mathbf{v}\\ &= t[(\mathbf{v}\cdot\mathbf{v})\mathbf{u}-(\mathbf{u}\cdot\mathbf{v})\mathbf{v}]\\ \Rightarrow \mathbf{u}^T\mathbf{x} &= t[(\mathbf{v}^T\mathbf{v})(\mathbf{u}^T\mathbf{u})-(\mathbf{u}^T\mathbf{v})(\mathbf{u}^T\mathbf{v})]\\ &= t(\mathbf{v}\times\mathbf{u})(\mathbf{v}\times\mathbf{u})\\ &\geq 0.\end{aligned}$$

And it is easily seem that if $\mathbf{u}=\mathbf{0}$ or $\mathbf{v}\times\mathbf{u}\neq\mathbf{0}$, $\lim_{t\to\infty}\mathbf{a}^TQ_{\mathbf{c}}^{-1}\mathbf{a}\to\infty$. Otherwise, if $\mathbf{v}=\mathbf{0}$, the facet is "vertical" and its distance to the wrench space origin unaffected by changes of the object space origin. However in conv , (G) there must be non-vertical facets, so we have shown that the ball measure can be made as small as desired.

2.3.8 Computation of the Measure

Using Generalized Linear Programming

The convexity of the constraints implies that we can use the probabilistic algorithm in [SW92] to compute r^* in time linear in the number of facets on conv , (G), given conv , (G). This number is at most $O(|G|^3)$ for objects in 3-space or $O(|G|)$ for planar objects, as seen previously. The computation of the convex hulls can be done in optimal time [Cha93], hence the measure can be computed in $O(|G|^3)$ expected time for objects in 3-space or $O(|G|\log|G|)$ expected time for planar objects.

Using Semidefinite Programming

Let $\{\mathbf{w}:\mathbf{a}^T\mathbf{w}\leq b\}$ be a facet of conv , (G) for some point set $G\subseteq\partial B$, and let $\mathbf{a}/b=[\mathbf{u}^T\mathbf{v}^T]^T$, for some $\mathbf{u},\mathbf{v}\in\mathbb{R}^3$, and assume $\mathbf{v}\neq\mathbf{0}$. Now $\mathbf{a}^TQ_{\mathbf{c}}^{-1}\mathbf{a}/b^2=\mathbf{u}^T\mathbf{u}+\mathbf{v}^T\mathbf{v}+\mathbf{v}^T\hat{\mathbf{c}}^T\hat{\mathbf{c}}\mathbf{v}+2\mathbf{u}^T\hat{\mathbf{c}}\mathbf{v}$, and let $\mathbf{x}=\hat{\mathbf{c}}\mathbf{v}=\mathbf{c}\times\mathbf{v}$. Then the quadratic constraint $\mathbf{a}^TQ_{\mathbf{c}}^{-1}\mathbf{a}/b^2-r'\leq 0$ can be written as

$$\mathbf{x}^T\mathbf{x}+2\mathbf{u}^T\mathbf{x}-r'+\mathbf{u}^T\mathbf{u}+\mathbf{v}^T\mathbf{v}\leq 0$$

which can be transformed into

$$\begin{pmatrix} I_3 & \mathbf{x} \\ \mathbf{x}^T & 2\mathbf{u}^T\mathbf{x} - r' + \mathbf{u}^T\mathbf{u} + \mathbf{v}^T\mathbf{v} \end{pmatrix} \succeq 0. \tag{2.6}$$

For a matrix M, $M \succeq 0$ means that M is positive semidefinite. This is a standard transformation in Semidefinite Programming, and follows directly from formulas given in [VB94].

The above equations, one for each facet of conv , (G) in turn can be written as the Semidefinite Program:

$$\begin{array}{ll} \text{minimize} & -r' \\ \text{subject to} & \text{equation (2.6) for every facet of conv , } (G)\ . \end{array}$$

The above program has dimension 4 and $3N$, if N is a the number of constraints. For a survey of Semidefinite Programming see [VB94]. Since for an $n \times n$ matrix, there exists an algorithm which takes $O(\sqrt{n}|\log \epsilon|)$ time to find a solution within ϵ of the optimal [Ali95], the above optimization takes $O(|G|^3|\log \epsilon|)$ time.

There is a minor problem however: given a solution $(\mathbf{x}, r')$ of the Semidefinite Program, we cannot recover $\mathbf{c}$ from the equation $\widehat{\mathbf{c}}\mathbf{v} = \mathbf{x}$. But from this equation, it follows that $\mathbf{c} \cdot \mathbf{x} = 0$ and

$$\mathbf{x} \cdot \mathbf{x} = (\mathbf{c} \times \mathbf{v}) \cdot \mathbf{x} = \mathbf{c} \cdot (\mathbf{v} \times \mathbf{x}),$$

and this is a system of two equations in 3 variables. (Note that this problem does not arise in 3 dimensions for planar objects.) These equations are independent when $\mathbf{v} \neq \mathbf{0}$, so we have one degree of freedom for $\mathbf{c}$. This corresponds to what has been shown in Section 2.3.6. For these possible $\mathbf{c}$'s, the volume of the ellipsoid remains constant. If $\mathbf{v} = \mathbf{0}$, the volume is the same for all $\mathbf{c}$, and no optimization is necessary. Thus any solution to this system provides the optimal value for r' and for r^*.

We have seen that it is possible to compute this metric using existing algorithms, quite efficiently. Its practical impact however remains to be studied, as well as extensions to the case of multiple objects. Also, we have discovered this measure quite recently, and have not developed algorithms to optimize grasps with respect to it. This would also be an interesting avenue of research.

Chapter 3

Computing Efficient Grasps

3.1 Optimization

Ideally, one would like to produce grips that use both small forces and require low friction coefficients. These grips should of course be stable and withstand outside forces that depend on the manipulation task. Some of these conflicting goals have been considered, mostly in two dimensions.

In the case of three fingered grasps of planar polygonal objects, Markenskoff and Papadimitriou [MP89] find a grasp that minimizes the forces required to balance a force along the third dimension applied to the center of gravity. This is done using frictional forces along the third dimension. Frictional forces in the plane are not considered. The forces are applied perpendicularly to the sides of the polygon and their lines of action meet at a point. The function minimized is a non-decreasing function of the forces, and in the presence of reflex vertices, the angles between their lines of action are minimized. This is done numerically.

Their procedure must however be repeated $O(n)$ times, where n is the number of sides of the polygon, to find the overall global optimum.

They also define and optimize the following simplified version of form closure. A set of fingers achieves form *semiclosure* of a polygon if any wrench through the center of gravity can be balanced by positive forces normal to the perimeter. To find such a set of fingers that minimizes the forces required to balance a unit wrench, they formulate an optimization problem which is then solved using linear programming and binary search. This optimization procedure must again be applied for all quadruples of sides.

Ji and Roth [JR88] consider three fingered grasps in three dimensions, but have a reduction of the problem to two dimensions for most cases. The plane considered is the one containing the three finger contact points. They assume that the fingers are given, but their angle can vary, subject to the restrictions imposed by the friction cones. These angles are chosen so that the dependence of the internal forces on friction is minimized.

More recently Ferrari and Canny [FC92] proposed algorithms for computing optimal grasps for a two and three jaw gripper. The optimality criteria are the variations of r_{con} and r_{max} described in Section 2.1.4. For a two jaw gripper and an n-sided polygonal object, an optimal grasp is found in $O(n)$ time. No time bound is given for the three jaw case. The three jaws of the three jaw gripper are at a fixed angle to each other, and will contact the object at vertices. In contrast, the algorithm given in Section 3.2 places fingers at interior points of edges. More recently Mirtich and Canny [BM94] study the case of rounded finger tips in two and three dimensions. This again allows them to place fingers at vertices, which simplifies the algorithms. They provide $O(n \log n)$ time algorithm for finding an optimal grasp for the polygonal case, $O(n^3)$ in three dimensions. It is based on finding a maximal circumscribing prism. The points of contact of this prism are taken to be the finger positions.

Kirkpatrick, Mishra and Yap [KMY92] studied the situation for planar and three-dimensional objects, in fact their algorithms apply in any dimension. Their results translate to the following: for a very large number of fingers m, and a finite set $B' \subset \partial B$ of allowable finger placements on ∂B, they can find a grasp G with quality $r_{con}(G) \geq (1 - 18\left(\frac{72}{m}\right)^{2/5}) r_{con}B'$. Of course $r_{con}B'$ is the best one can do. For planar objects, the quality is $r_{con}(G) \geq (1 - 9\frac{18}{m}) r_{con}B'$. See Chapter 4 for their result for general dimension. In addition, in Chapter 4 we describe a general technique for finding close to optimal grasps under various measures of optimality.

In the remainder of this Chapter, we describe an $O(n^2 \log n)$ algorithm for finding three finger optimal grasps in the plane for a polygonal object with n vertices.

3.2 Optimal 3 finger Grasping in the Plane

While the question of analyzing and synthesizing *a* closure grasp or *a* fixture is fairly well-studied, the question of devising efficient algorithm for synthesizing grasps of *good quality* has received relatively less attention. A systematic exploration in this direction was initiated in the work of Kirkpatrick, Mishra and Yap [KMY92], some of which was outlined earlier.

In order to better understand the underlying structure as well as to provide practical solutions in the simpler settings (as more common in manufacturing), we have directed our attention to the cases where we study lower-dimensional objects (2-D or $2\frac{1}{2}$-D) and of simpler geometry (polygonal objects) or simpler robot hands. In this chapter, we explore this problem for two-dimensional polygonal objects with hands of relatively few fingers[1]. and solve various algorithmic questions regarding the computation of an optimal three finger planar grasp. We present a novel $O(n^2 \log n)$-time algorithm to compute such an optimal grasp for an arbitrary simple n-gon. We also discuss several variations on the problem and some open questions in the area that remain unsolved.

3.2.1 Preliminary

We wish to obtain the best three-finger grasp of a planar polygonal object assuming non-frictional contacts. Note that in this case, since it is not possible to guarantee that the resulting grasp will have the force closure properties, we are willing to sacrifice the condition requiring torque-closure. In other words, we wish only to achieve a three-finger grasp such that the smallest external force such a grasp can resist is as large as possible.

More formally given a simple n-gon P, we wish to choose three distinct points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ on the interior of the edge segments of P such that the following properties hold:

1. The unit inner normals $\mathbf{n}(\mathbf{p}_1)$, $\mathbf{n}(\mathbf{p}_2)$ and $\mathbf{n}(\mathbf{p}_3)$ are concurrent.

2. The unit inner normals $\mathbf{n}(\mathbf{p}_1)$, $\mathbf{n}(\mathbf{p}_2)$ and $\mathbf{n}(\mathbf{p}_3)$ positively spans the two-dimensional

[1]The material in this section has previously appeared in [MT94].

force space, i.e.,

$$(\forall \mathbf{w} \in \mathbb{R}^2)\ (\exists f_i \geq 0, 1 \leq i \leq 3)\ \mathbf{w} = \sum_{i=1}^{3} f_i \mathbf{n}(\mathbf{p}_i).$$

3. The unit normals are "well-balanced" in the sense that

$$\begin{aligned}\min\Big\{ |\mathbf{w}| : \mathbf{w} \in \mathbb{R}^2,& \\ &(\exists f_i \geq 0, 1 \leq i \leq 3)\ \chi(f_1, f_2, f_3) = 1 \\ &\mathbf{w} = \sum_{i=1}^{3} f_i \mathbf{n}(\mathbf{p}_i) \Big\},\end{aligned}$$

   is as large as possible (among all choices of $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$). Here, $\chi(f_1, f_2, f_3)$ denotes a finger force constraint condition on the magnitude of the forces applied at the points of contact.

Thus the first property denotes the trivial torque equilibrium condition; the second property denotes the force closure condition and the third property measures the goodness of the grasp. In English, the third property says: under the condition χ_{con}, we wish to maximize the radius of a disk, centered at origin and contained in the triangle formed by (convex hull of) the points (on the unit circle) corresponding to the vectors $\mathbf{n}(\mathbf{p}_1)$, $\mathbf{n}(\mathbf{p}_2)$ and $\mathbf{n}(\mathbf{p}_3)$. Similarly, under the condition χ_{max}, we wish to maximize the radius of a disk, centered at origin and contained in the Minkowski sum of the points (on the unit circle) corresponding to the vectors $\mathbf{n}(\mathbf{p}_1)$, $\mathbf{n}(\mathbf{p}_2)$ and $\mathbf{n}(\mathbf{p}_3)$—a convex hexagon.

Let the corresponding radii be denoted as $\rho_{con}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ and $\rho_{max}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$, respectively. Note that, if the angle α_i's $(1 \leq i \leq 3)$ denote the angles between the inner normals then $\alpha_{max} = \max(\alpha_1, \alpha_2, \alpha_3) \geq 2\pi/3$ completely determines the radii

$$\begin{aligned}\rho_{con} &= \cos(\alpha_{max}/2), \quad \text{and} \\ \rho_{max} &= \sin \alpha_{max}.\end{aligned}$$

Thus both these metrics are monotonically decreasing functions of $2\pi/3 \leq \alpha_{max} \leq \pi$, and it suffices to minimize α_{max}. However, for the sake of the ease of exposition, we will

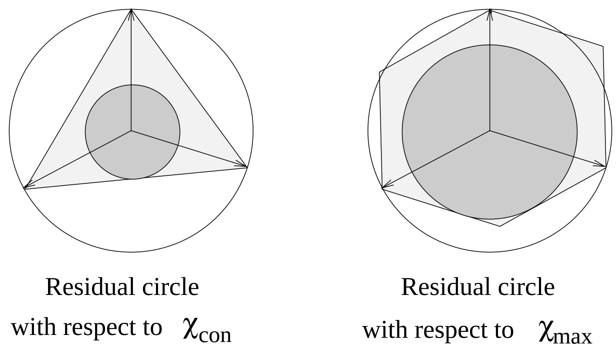


Figure 3.1: Grasp metrics associated with χ_{con} and χ_{max}.

use $\rho = \rho_{con}$, and refer to it as the "*residual radius*" of $\mathbf{n}(\mathbf{p}_1)$, $\mathbf{n}(\mathbf{p}_2)$ and $\mathbf{n}(\mathbf{p}_3)$. The optimal value of residual radius is denoted by ρ^*.

Note that given an edge $e = ab$ of the polygon P, for every point $p \in ab$, $\mathbf{n}(\mathbf{p})$ defines a unique point $q(e)$ on the unit circle in $\mathbb{R}^2$. Thus we may simply refer to this point on the unit circle by $q(e)$. Henceforth, let the edges of the n-gon be given as $E = \{e_1, e_2, \ldots, e_n\}$ and the corresponding points on the unit circle be $Q = \{q_1, q_2, \ldots, q_n\}$, where $q_i = q(e_i)$ $(1 \le i \le n)$.

3.3 A Cubic Time Algorithm

We may note at this point that there is a trivial $O(n^3)$ time algorithm to find an optimal grasp of a simple n-gon, P, by exhaustively enumerating all edge triples of P and by examining each triple successively. In order for an edge triple (e_i, e_j, e_k) to produce three necessary optimal contact points, it must be the case that (q_i, q_j, q_k) form a triangle with a positive residual radius of ρ^*—a condition that can be checked easily in $O(1)$ time. However, this is not sufficient—since we must check that there are three points $\mathbf{p}_i \in e_i$, $\mathbf{p}_j \in e_j$ and $\mathbf{p}_k \in e_k$ satisfying the torque equilibrium condition; namely, that $\mathbf{n}(\mathbf{p}_1)$, $\mathbf{n}(\mathbf{p}_2)$ and $\mathbf{n}(\mathbf{p}_3)$ are concurrent meeting at some point c.

We proceed as follows: Consider an edge ab of P. Let $HP(a, ab)$ be the open half plane containing ab and delimited by a line containing a and normal to ab and similarly, let $HP(b, ab)$ be the open half plane containing ab and delimited by a line containing b

and normal to ab. Let

$$\mathrm{slab}(e) = HP(a, ab) \cap HP(b, ab),$$

where $e = ab$.

Then it is easy to see that for a triple of edges $(e_i,\ e_j,\ e_k)$ to satisfy the torque equilibrium condition, it is necessary and sufficient that

$$\mathrm{slab}(e_i) \cap \mathrm{slab}(e_j) \cap \mathrm{slab}(e_k) = C \neq \emptyset.$$

The point of concurrency $c \in C$, and the contact points $\mathbf{p}_i$, $\mathbf{p}_j$ and $\mathbf{p}_k$ are determined by the normals from c onto the edges e_i, e_j and e_k.

Thus our previous arguments can be summarized to be saying that an edge triple $(e_i,\ e_j,\ e_k)$ defines an optimal grasp if $\mathrm{slab}(e_i) \cap \mathrm{slab}(e_j) \cap \mathrm{slab}(e_k)$ is nonempty and that the triangle formed by the corresponding points on the unit circle has a positive residual radius of ρ^*, maximal among all choices of edge triples. These considerations yield an $O(n^3)$-time algorithm.

3.3.1 Applications to Immobility and Closure Grasps

We wish to note at this point that the placements for the three fingers found by the algorithms in this section are also immobilizing sets. This follows from the following theorem of Czyzowicz, Stojmenovic and Urrutia [CSU90].

Theorem 3.3.1. *Three points immobilize a triangle if and only if the normals to the triangle at these points are concurrent.*

The grasps found by our algorithm satisfy the concurency condition by construction, and the extension of the edges at which grasp points are placed form a triangle since their normals are positively spanning, again by construction.

Furthermore, if any non zero coefficient of friction is allowed, these grasps are also good three finger closure grasps. This can be seen as follows. Let μ be this coefficient of friction, and assume μ is sufficiently small. Assume also that the origin lies at the concurrency point of the three forces. Then **,** $(\mathbf{p}_i)$ lie on the xy plane in the wrench space $\mathbb{R}^3$. For a unit vector $\mathbf{n}$ in the xy plane in $\mathbb{R}^3$, let $\mathbf{n}^\perp$ be the unit vector perpendicular

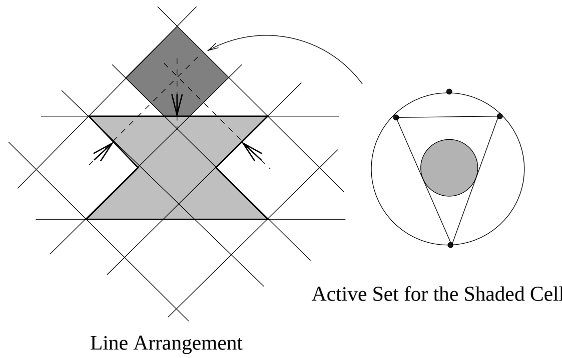


Figure 3.2: The line arrangement associated with an object.

to $\mathbf{n}$ and rotated by $\pi/2$ counter-clockwise. Let also $\mathbf{n}_i = \mathbf{n}(\mathbf{p}_i)$. The two rays bounding the friction cone at $\mathbf{p}_i$ are $\sqrt{1-\mu^2}\mathbf{n}_i \pm \mu\mathbf{n}_i^{\perp}$, and their image by , is

$$[\sqrt{1-\mu^2}\mathbf{n}_i \pm \mu\mathbf{n}_i^{\perp}, \pm|\mathbf{p}_i|\frac{\mu}{\sqrt{1+\mu}}].$$

Letting α be the half angle of the friction cone, the last term follows from the formula of the cross product which involves the sine function and the fact that $\mu = \tan\alpha = \frac{\sin\alpha}{\cos\alpha}$.

Hence the top facet of conv $\{,_{\mu}(\mathbf{p}_i) : i = 1, 2, 3\}$ will be farther from the xy plane (and from the origin) by a distance of at least $\min_i\{|\mathbf{p}_i|\frac{\mu}{\sqrt{1+\mu}}\}$. Similarly for the bottom facet. Thus when μ is sufficiently small so that no other point of conv $\{,_{\mu}(\mathbf{p}_i) : i = 1, 2, 3\}$ is closer to the origin, this quantity will be a lower bound for the quality of the grasp.

3.3.2 An Improved Subcubic Algorithm

Next, we improve upon the trivial $O(n^3)$-time algorithm. Here, we present an $O(n^2 \log n)$-time algorithm for finding the optimal three fingered planar grasp for an arbitrary simple polygon and some simple improvements for convex polygons.

We first describe the algorithm assuming that the polygon P is nondegenerate (in the sense that will be made precise later) and then remark on how the nondegeneracy can be eliminated by a simple modification to the algorithm.

The algorithm can be described as follows: First we create the two-dimensional line arrangement formed by a collection of lines consisting of three lines per edge, where the triplet of lines associated with an edge ab are: (1) the line containing the edge ab, (2) the line normal to ab, containing a and (3) the line normal to ab, containing b. Now consider a nonempty cell C of this arrangement: we say a point $q = q(e)$ on the unit circle is *active* for this cell, if $\text{slab}(e) \supseteq C$. The subset of points on the unit circle (among the points $q_1, q_2, \ldots, q_n$ of Q) that are active for this cell C, is called its active set and denoted by $\text{active}(C) \subseteq Q$. Now, if we find three points q_i, q_j and $q_k \in \text{active}(C)$, whose residual radius $\rho(C)$ is as large as possible (and positive), then it is seen that ρ^* is simply the maximum of all $\rho(C)$'s taken over all cells of the arrangement.

Note that there are at most $O(n^2)$ cells altogether and as we go from one cell C to its adjacent cell C' then the active(C') can be computed from the active(C) by adding or deleting a point on the unit circle, depending on the line containing the $C \cap C'$. Of course, here we have tacitly assumed that the polygon is nondegenerate, in the sense that all the lines on the arrangement are distinct, since otherwise $C \cap C'$ may belong to more than one line of the arrangement and thus require addition and deletion of more than one point of the set Q. Clearly, the active sets for all the cells can be computed in $O(n^2)$ time by visiting the cells of the arrangement, starting from a cell with an empty active set (such a cell exists sufficiently far away from the polygon P). However, computing the $\rho(C)$ for each cell may still take $O(n)$ time, thus forcing the entire procedure to take $O(n^3)$ time.

We circumvent this problem by the following simple trick: First of all we maintain the active(C)'s in a clockwise order in a dynamic balanced binary search tree. Since each update operation on this data structure takes $O(\log n)$ time, this increases the complexity of computing the active sets of all the cells to $O(n^2 \log n)$-time.

At any instant, we only remember $\tilde{\rho}$—the maximal residual radius seen so far. That is, $\tilde{\rho}$ is simply the maximum of those $\rho(C)$'s corresponding to only those cells C that have been visited so far. We also remember the edge triple associated with the radius value $\tilde{\rho}$. When we go from a visited cell C to an adjacent unvisited cell C', we do one of two things: If going to the next cell entails deletion of a point, q_i, on the unit circle, then we only have to update the active(C'); the maximal residual radius of C' cannot be larger than that of C and thus $\tilde{\rho}$ remains unchanged. If going to the next cell, on

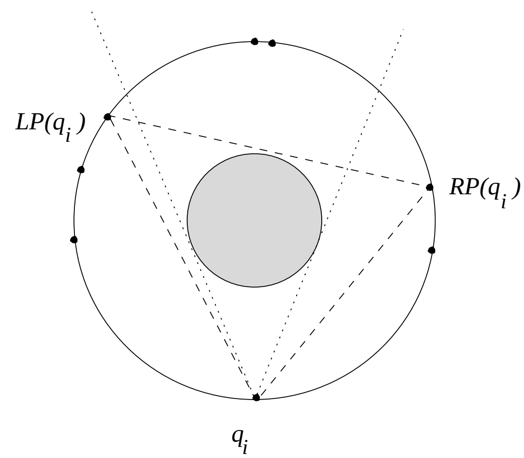


Figure 3.3: Test involving q_i and a possible residual radius value of ρ_k.

the other hand, entails addition of a point, q_i on the unit circle, then we have to both update the active(C') and check if $\tilde{\rho}$ can be improved. If the maximal residual radius of C', $\rho(C') > \tilde{\rho}$, then the associated triplet from active(C') must involve the new point q_i and two of the old points. How can we do this operation efficiently?

First note that residual radii cannot take all possible values but only one of $\binom{n}{2}$ values, each value being determined by a pair of distinct points q_l and q_m and is equal to the radius of the circle that is centered at the origin and has the line containing q_l and q_m as tangent. All these radii can be sorted in $O(n^2 \log n)$ time and are denoted by

$$0 \leq \rho_1 \leq \rho_2 \leq \cdots \leq \rho_i \leq \cdots < 1.$$

Suppose before visiting the cell C' the maximal residual radius seen so far is $\tilde{\rho} = \rho_i$. When we go to the cell C' (which requires adding the point q_i), we will successively test if it has a residual radius no smaller than ρ_{i+1}, ρ_{i+2}, etc. until we fail for some value ρ_j $(j > i)$. Each such test can be performed in $O(\log n)$ time as explained below.

Let $i < k \leq j$, and we wish to test if active(C') has three points involving q_i and of residual radius $\geq \rho_k$. Consider a circle $C(\rho_k)$ of radius ρ_k and centered at the origin. Two distinct points of active(C') are said to be *mutually visible* if the line segment connecting these two points do not intersect the interior of $C(\rho_k)$. Thus our test succeeds if we can find a pair of mutually visible distinct points among the active(C'), each of which is also mutually visible with q_i. The following technique was also used in [KMY92]. Let the *leftmost partner* of q_i be the *last* mutually visible point of q_i encountered, visiting the points of active(C') in clockwise order starting from q_i. We call this point $LP(q_i)$. Similarly, we define the the *rightmost partner* of q_i by visiting the points of active(C') in anti-clockwise order, and call it $RP(q_i)$. Since the active points of C' are kept in their sorted order in a balanced search structure, both $LP(q_i)$ and $RP(q_i)$ can be computed in $O(\log n)$ time. Then it only remains to check that $LP(q_i)$ and $RP(q_i)$ are mutually visible, a step that can be accomplished in $O(1)$ time.

Thus, we can keep track of $\tilde{\rho}$ by performing a sequence of tests per each new cell, each of which takes $O(\log n)$ time. Note that while there is no a priori bound on the number of tests we may need to perform for a new cell, it should be obvious that all but the last test succeeds and the last test fails. Thus there are at most one test per cell that fails, and the totality of all such failed tests incur a cost of $O(n^2 \log n)$. On the other hand, if we have a successful test involving a radius value ρ_k, then *we shall never perform another successful test involving ρ_k, subsequently.* Thus, the total number of successful tests are bounded by the number possible radii values ($\binom{n}{2}$ of those) and altogether they incur a cost of $O(n^2 \log n)$. Clearly, when we are done visiting all the cells, we have the global maximal residual radius ρ^* together with the edge triple, which readily give the three contact points, and we have spent $O(n^2 \log n)$ time.

If the polygon P is degenerate then the resulting arrangement may force us to add and delete many points of Q while going from a cell to its adjacent cell. If we enforce the discipline that all the deletions are performed before all the additions and each update is performed sequentially then the correctness of the algorithm still holds and the performance analysis goes through *mutatis utandis*. In summary, we have

Theorem 3.3.2. *Given an arbitrary simple n-gon P, we can compute a three finger optimal grasp of P in $O(n^2 \log n)$ time.*

Proof. The complexity analysis follows from the discussion preceding the theorem: The possible radii values can be computed and sorted in $O(n^2 \log n)$ time; the cells can all be visited with the active sets computation taking $O(n^2 \log n)$ time; the number of tests involved in going from cell to cell is no more than the sum of the number of possible radii values and the number of cells in the arrangement with each test taking $O(\log n)$ time and thus contributing only $O(n^2 \log n)$ cost to the total cost.

To see the correctness of the algorithm, note that if the computed value at the end is $\tilde{\rho}$ then clearly

$$\tilde{\rho} \leq \rho^*.$$

Conversely, consider the set of edge triples that lead to the maximal residual radius ρ^* and all the cells that are contained in the intersection of three slabs associated with each such edge triple. Among all such cells consider the one that was visited the earliest, say C'. Let the preceding visited cell be denoted C. Let the maximal residual value seen up to the time C was visited be $\tilde{\rho}'$. Thus $\tilde{\rho}' \leq \rho(C) < \rho^* = \rho(C')$. Thus active$(C')$ must have been obtained by addition of a point $q_i \in Q$. Thus, active(C') has two other points q_j and q_k such that a residual circle of radius ρ^* touches an edge of the triangle formed by q_i, q_j and q_k. Thus ρ^* is a possible residual radius and the tests at the cell C' involving possible residual radii values larger than $\tilde{\rho}'$ will all succeed up to ρ^*. Thus if the computed value at the end is $\tilde{\rho}$ then

$$\tilde{\rho} \geq \rho^*.$$

Hence they must be both equal and our algorithm correctly determines the optimal three finger grasp for P. □

3.3.3 Variations and Open Questions

There are several open questions related to the problem of finding optimal planar grasps. We briefly discuss these problems.

(1) Consider a variation on the above problem: Suppose we are given a simple polygon P with certain subset of ∂P designated as "forbidden" and its complement, "feasible." Assume that the feasible parts of the polygon consists of at most k segments (the edge segment ab being allowed to be a point a $(a = b)$, in the degenerate case). We are asked to find an optimal three-finger grasp of the polygon with none of the fingers on a forbidden

region. Using a small variation of the above algorithm, we can solve this problem in $O(k^2 \log k)$ time—only modify the line arrangement to consist of the following triple of lines per feasible edge segment $ab \subset e$, where e is an edge of P: (i) the line containing e, (ii) the line normal to e and containing a, and (iii) the line normal to e and containing b. If the edge segment is a point $a \in e$ then the above situation degenerates to two lines, one containing e and the other normal e at a.

(2) We do not know whether there is a better solution for the above problem with improved complexity. For instance, it is not even clear whether there are $O(n)$ time algorithms for objects with simpler geometry, e.g., convex objects.

(3) Is the optimal grasp found also optimal with respect to r_{con} in the presence of friction? We suspect that this is the case.

(4) Sometimes, we wish to determine not just one optimal three finger grasp but all of them. Then we may use any one of this class of optimal grasps, depending on the task at hand. Clearly, the brute force $O(n^3)$ time algorithm will succeed to do so. Note that the algorithm of the previous section cannot be easily modified into a two pass algorithm, since addition of a new point (in the process of going from one cell to an adjacent cell) may create an $O(n)$ edge triplets of residual radius ρ^*. Here, we describe an $O(n^2 \log n)$ algorithm for the special case when the object is convex.

Let P be a convex n-gon and let the possible residual radii (as in the preceding subsection) be given as

$$0 \le \rho_1 \le \rho_2 \le \cdots \le \rho_i \le \cdots < 1.$$

We shall find the optimal residual radius ρ^* by performing a binary search on the sequence of possible residual radii. For a given value of ρ_i, we can enumerate all the edge triples that lead to a residual radius of ρ_i in $O(n^2)$ as follows: Corresponding to the possible radius value ρ_i, there are at most $O(n)$ edge pairs (e_i, e_j)'s such that the corresponding points q_i and $q_j \in Q$ on the unit circle satisfy the property that the line determined by $q_i q_j$ is tangent to a circle $C(\rho_i)$ centered at the origin and of radius ρ_i. Now for each such edge pair, we need to check in $O(n)$ time if there is another edge e_k such that $q_k \in Q \setminus \{q_i,$

$q_j\}$ is mutually visible (with respect to $C(\rho_i)$) to both q_i and q_j and that

$$\mathrm{slab}(e_i) \cap \mathrm{slab}(e_j) \cap \mathrm{slab}(e_k) \neq \emptyset.$$

We can thus enumerate all the e_k's that succeed this test. The binary search only considers $O(\log n)$ different values of ρ_i's and terminates with success with the largest possible value ρ^* and enumerating all edge triples corresponding to ρ^*. It is then trivial to describe all possible three finger optimal planar grasps. Thus the algorithm has a time complexity of $O(n^2 \log n)$.

However, the algorithm applied to a nonconvex polygon leads to an $O(n^3)$-time algorithm, as in a pathological case, there may be $O(n^2)$ edge pairs to be considered for a given value of ρ_i. It is noteworthy that this algorithm is rather simple to implement and may perform well in practice. For instance, if one performs binary search on the real interval $[0, 1]$ (instead of the possible radii values), then for a random polygon this algorithm can compute in $O(n \log n \log(1/\epsilon))$ all three finger grasps whose corresponding residual radii lie in the range $[\rho, \rho^*]$ of size $< \epsilon$, for sufficiently small positive ϵ.

Chapter 4

A Randomized Algorithm

Suppose we are given a finite set of points on ∂B on which we can place fingers. Furthermore, we assume the frictionless contact model. We would like to obtain grasps of high efficiency but with few fingers. These are two conflicting goals as seen in Section 2.1.1, so in this chapter we provide algorithms for optimizing one quantity or the other. The problem of obtaining an optimal grasp with respect to the r_{con} measure has been studied Kirkpatrick, Mishra and Yap [KMY92]. They assume a large number of points on ∂B is given, and find an almost optimal subset of c of them. The number c however must be quite large: $c > 7 \times 10^7$.

We have already seen how optimal grasp synthesis problem maps into geometric optimization problems. In this chapter, we present a class of randomized algorithms which provides an approximation to a large class of the optimization problems encountered. For example, given an object with n points on its boundary where fingers can be placed, we give algorithms to select a good grasp with a minimal number c of fingers (up to a logarithmic factor) for the r_{con} and r_{nasty} measures. This grasp is the best among all the $c \log c$ finger grasps. Along similar lines, given an integer c, we find the "best" $\kappa c \log c$ finger grasp for a small constant κ. Depending on the measure, for our application the algorithms run in expected time $O(c^2 n^{1+\delta})$ or $O((nc)^{1+\delta} + c^4 \log^4 n)$. Here δ is any positive constant. This setting generalizes to higher dimensions in the context of finding sets of fixtures, and we provide randomized approximation algorithms there too.

Let us describe the setting for the optimization algorithms in full generality. Let $\rho(\cdot)$ be some grasp efficiency measure, generalized to higher dimension (see below). It has an associated geometric object L containing the origin 0, which can be scaled about 0. We define

$$\rho(U) = \max\{\lambda : \lambda L \subseteq U\}.$$

For $\rho(\cdot) = r_{con}(\cdot)$, $L = \mathcal{B}(1)$ in the appropriate dimension. One class of problems is: for a desired grasp efficiency ρ_0, select a set $G \subset \partial B$ of smallest possible size with $\rho(, (G)) \geq \rho_0$. We assume here that the set of possible points on ∂B out of which points of G are selected is finite; let its size be n. This translates to the following purely geometric problem (valid in any dimension): let U be a set of n points in $\mathbb{R}^d$ such that the origin is contained in the interior of its convex hull.

> [**MinCover-L**]: Given ρ_0, we wish to find the smallest set $C \subset U$ or *cover* of size c^* with $\rho(\operatorname{conv} C) \geq \rho_0$.

We solve a corresponding approximation problem: that of finding a cover of size $c^* d \log c^*$. We will call this a $d \log c^*$-approximation of the optimal cover.

The companion problem is: given a maximum number of fingers c, which grasp G of size c maximizes $\rho(, (G))$? The geometric version of this problem is:

> [**MaxScale-L**]: Given an integer c, find the set $C \subset U$ of size c which maximizes $\rho(C)$. Let $\rho^*(c)$ be this maximum.

In the approximation version, for which we provide an algorithm, we ask for a $d \log c^*$-approximation of the best cover C. In the sequel, we will replace L by various convex sets.

For example, the problem of finding a good grasp under the r_{con} measure corresponds to **MinCover-B** and **MaxScale-B**, and for r_{nasty} for polyhedral objects, the corresponding optimization problems are **MinCover-P** and **MaxScale-P**. The polytope P is then $, (\partial B)$.

We present an algorithmic framework derived from an algorithm by Clarkson [Cla93] for polytope covering. This framework yields efficient randomized approximation algorithms for the above problems for various types of the set L — in fact, this approach works for any set L that has a *strong violation* oracle [GLS88] or *half-space emptiness query* [Mat92] can be used for the **MinCover-P** problem.

Let $\gamma = 1/\lfloor d/2 \rfloor$. The approximation versions of **MaxScale-B** and **MinCover-B** can be solved in expected time $O((n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))}) + c\log(n/c)(c\log c)^{\lfloor d/2 \rfloor})$ when d is fixed, using sophisticated data structures. Here c represents the optimal cover size for **MinCover-B** and the desired cover size for **MaxScale-B**. In both cases a cover of size $4cd\log c$ is returned. For a polytope P of ℓ vertices, the **MinCover-P** $d\log c$–approximation problem was already solved by Clarkson's algorithm in expected time $O(n^{1+\delta} + c\ell^{1+\delta} + c(\ell c)^{1/(1+\gamma/(1+\delta))} + (nc)^{1/(1+\gamma/(1+\delta))})$ or $O(\ell c\log c + n)c\log(n/c)$ using a simpler version of the algorithm. In the same time bound, we can solve the approximation version of **MaxScale-P** (with c is an input parameter), and **MinCover-P** (with c the optimal cover size.)

4.1 Related Results

Exact algorithms for the problems mentioned in the previous section that run in polynomial time seem unlikely. In fact problems similar to these have been shown to be NP-hard [BMK94] or NP-complete [DJ90]. In [DJ90], it is shown that the problem of finding a minimal-facet separating polytope for arbitrary nested polyhedra is NP-hard.

We therefore consider approximation algorithms for the **MinCover** and **MaxScale** problems. The **MinCover-P** problem arises in a dual form in the context of separating two nested polyhedra [MS92b, Cla93]. In this problem, we are given two polyhedra L and U with $U \subseteq L$, and we want to find a separating polytope S with $U \subseteq S \subseteq L$ with small number of facets. If we ask for the smallest possible number of facets, this problem is NP-complete [DJ90]. Several approximation algorithms for this problem have been presented—both deterministic [DJ90, BG94a], and randomized [Cla93]. Consider U as the intersection of a set of half-planes. In the above-mentioned algorithms, S is found by eliminating facets (halfplanes) from U, while making sure that the new polytope is still contained in L. We assume that all polytopes involved contain the origin, and we apply the point-plane *polarity* duality transformation, with respect to the unit sphere. For future reference, call this duality $\mathcal{D}$. This duality maps a hyperplane at distance r

from the origin and with unit normal n to the point n/r and vice versa. It also maps $\mathcal{B}(r)$ to $\mathcal{B}(1/r)$ with an appropriate extension of the definition. Applying this duality transformation, we have again two convex polytopes conv L and conv U, where U is a set of points. The above separation framework amounts to covering L by the convex hull of a subset of U and this is exactly what we need.

For the **MaxScale-B** problem, Kirkpatrick *et al.* [KMY92] give an algorithm that finds a cover C of size c containing a ball of radius

$$r(C) = \left[1 - 3d\left(2d^2/c\right)^{2/(d-1)}\right] r(U),$$

for $n \geq c \geq 13^d d^{(d+3)/2}$ in time $O(\mathrm{LP}(n,d)c)$. The radius found is almost optimal for that cover size. Here $\mathrm{LP}(n,d)$ is the time required to solve a linear program of size n and dimension d. Currently the best deterministic algorithm runs in time $O(d^{7d+o(d)}n)$ time [CM93] and the best randomized algorithm in time $O(d^2n + e^{O(\sqrt{d\log d})})$. See [Gol95] for this bound and a recent survey.

4.2 The Computational Framework

Consider a set U of n points in $\mathbb{R}^d$, and a convex set L with $0 \in L \subset \text{conv } U$. We will use a routine FIND COVER which, for a given cover size c outputs a cover of size $4cd \log c$, if a cover of size c exists, otherwise it fails. Let c^* be the size of the smallest cover $C \subset U$ for L. If we wish to find c^* for fixed, un-scaled L, we use OPTIMAL COVER. The outer loop finds an approximation of c^* up to a factor of $5/4$, and goes as follows.

```
OPTIMAL COVER
Input: Set U of potential cover vertices; object to be covered.
Output: Smallest cover, up to a factor of 4d log c.

1.    for i = lg d, lg d + 1, ...
2.          FIND COVER with c = (5/4)^i
            if FIND COVER succeeded, stop.
end{OPTIMAL COVER}.
```

The algorithm FIND COVER is defined below. It needs the desired cover size c. It uses another routine FIND BAD FACET which is an implementation of the *strong violation* problem, also known as *half space emptiness query.* That is, given a half space h with a positive side, determine whether L lies entirely in the positive half-space defined by h. In fact for the **MaxScale** problems we will need more: given a direction u, we will need the supporting hyperplane for L with normal u which is the furthest in that direction.

The algorithm goes as follows. We repeatedly take a random sample R, of expected size $s = 4cd \log c$, and test whether L (or some scaled version of L) is contained in conv R using FIND BAD FACET.

Let k be a constant to be specified later. Call an iteration of the loop in FIND COVER *successful* if the weights were doubled, *i.e.* if $w(U_F) \leq w(U)/(kc)$. We also require that $|R| \approx 4cd \ln c$. Consider the following Chernoff bound.

Theorem 4.2.1 ([MR95, p.68], Theorem 4.1). *Let $X_1, \ldots, X_n$ be independent Poisson trials (coin tosses), $X \in \{0,1\}$, such that for $1 \leq i \leq n$, $\Pr[X_i = 1] = p_i$, where $0 < p_i < 1$. Then $X = \sum_{i=1}^n X_i$, $\mu = E[X] = \sum_{i=1}^n p_i$, and and $\delta > 0$,*

$$\Pr[X > (1+\delta)\mu] < \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^\mu .$$

Using the above theorem, we see that $|R|$ is no larger than a factor of 4 of this value with probability $< 0.08^{4cd \ln c}$. We simply take new random samples until the size of R is as required. This additional requirement does not change the expected asymptotic running time.

If a cover of size c is not found in the number of iterations specified, FIND COVER fails; otherwise, it returns a cover of expected size $4cd \log c$. This number is chosen in Lemma 4.2.4 to guarantee success if a cover of size c exists.

FIND COVER

Input: Size c of desired optimal cover; Max number I of iterations.

Output: Cover of size $c4d \log c$.

$s = c4d \ln c$, {*the expected size of cover we will find*}

$I = 1 + 8c \lg n/c$, {*the maximum number of iterations*}

for all $p \in U$, let $w_p = 1$

repeat for I successful iterations:

  Choose $R \subset U$ at random. (see below)

  FIND BAD FACET F of conv R.

  if no bad facet, return R.

  Let $U_F =$ points of U seeing F

  if $w(U_F) \leq w(U)/(kc)$ then

    for all $p \in U$, let $w_p = 2w_p$ {*reweight*}

    else {*not a successful iteration, try again*}

end{FIND COVER}.

The random selection of R is done by picking each point p of U independently with probability

$$\Pr(p) = 1 - \left(1 - \frac{w_p}{w(U)}\right)^s \leq s\frac{w_p}{w(U)}.$$

The expected size of R is $\sum_{p \in U} \Pr(p) \leq \sum_{p \in U} s\frac{w_p}{w(U)} = s$. Finally the heart of the work is done in FIND BAD FACET whose variations are described in the next section.

The correctness of the general algorithm follows from a series of lemmas. The following lemmas were shown in [Cla93] for L a convex polytope, but the proofs do not use the fact that L is polyhedral and actually apply to any set L. We state them here for completeness, and also to bound some constants explicitly.

Lemma 4.2.2. *Let L be any convex set not contained in* conv R*, with a point $p \in L$ on the negative side of some facet F of* conv R*. Let U_F be the set of points of U that see F. Then there is a point of the optimal cover C of L among the points of U_F.*

Proof. Assume no point of U_F is in C, and assume that a cover exists, *i.e.* that $L \subseteq$ conv U. Then there are no points of C in the closed half-space H delimited by aff F and on F's negative side. Which means conv $C \subseteq H$. But $p \in \mathbb{R}^d \setminus H$, which contradicts the assumption that C is a cover. □

This lemma is the basis of this algorithm, and its derivatives. It essentially says that by finding a set U_F, we have gained some information about C since one of its members must be in the relatively small set U_F. We restrict the size of this set to be bounded by the condition $w(U_F) \leq w(U)/(kc)$.

We say that a facet F of a polytope P is visible from a point q if for every $p \in F$, the segment $\overline{pq}$ does not meet P. Here P will be conv R. Following [Cla93], we define an *L-facet* to be a facet visible from of point of L. Then we have the following lemma, which is a slightly modified version of Clarkson's lemma 2.2. The proof follows his closely.

Lemma 4.2.3. *Given that an L-facet F is found, the probability that an iteration of* FIND COVER *will be successful (i.e. the set U_F satisfies $w(U_F) \leq w(U)/(kc)$), is at least 1/2.*

Proof. Let F be a potential facet of conv F with $w(U_F) > \frac{w(U)}{kc}$. Assume that F has d vertices. Then F is a facet of conv R exactly when its d vertices are in R and points U_F that see F are not in R. For a give F, this happens with probability

$$\begin{aligned} a = \prod_{p \in \mathrm{vert} F} \left(1 - (1 - \frac{w_p}{w(U)})^s\right) \prod_{p \in U_F} (1 - \frac{w_p}{w(U)})^s &\leq \prod_{p \in \mathrm{vert} F} s \frac{w_p}{w(U)} \cdot \prod_{p \in U_F} e^{-s \frac{w_p}{w(U)}} \\ &\leq \left(\prod_{p \in \mathrm{vert} F} w_p\right) \frac{s^d}{w(U)^d} e^{-s \frac{w(U_F)}{w(U)}}. \end{aligned}$$

We sum the first term over all potential facets F: all possible d-tuples of points of U. But the expression

$$b = \sum_{\text{potential facet } F} \prod_{p \in \mathrm{vert} F} w_p \leq \binom{n}{d} \left(\frac{w(U)}{n}\right)^d$$

with the constraint that $\sum_{p \in U} w_p = w(U)$, is maximized when all $w_p = w(U)/n$. Hence $b \leq \binom{n}{d} \left(\frac{w(U)}{n}\right)^d$. We then have, using Stirling's approximation for the factorial function, that the probability that each facet F of conv R satisfies $w(U_F) > \frac{w(U)}{kc}$ is less than

$$\begin{aligned}
\sum_{\text{potential facet } F} a &\leq \binom{n}{d}\left(\frac{w(U)}{n}\right)^d \frac{s^d}{w(U)^d} e^{-s\frac{w(U_F)}{w(U)}} \\
&\leq \frac{s^d}{n^d}\frac{(en)^d}{d^{d+1/2}} e^{-\frac{s}{kc}} \\
&= \left(\frac{er}{d}\right)^d \frac{e^{-\frac{s}{kc}w(U)}}{\sqrt{d}} = a'.
\end{aligned}$$

We let $s = 4cd \ln c$. Then it is easy to verify that $a' < 1/2$ for $k = 2$, $d > 2$ and $c > 43$, or for $k = 1.501$, $d > 2$ and $c > 6$. □

This lower bound on s determines the approximation factor for the cover. This fact does not use the geometry of the situation at all, and one wonders whether the lower bound on s could be reduced in our application where the points of U are in (non-strictly) convex position.

Finally, to obtain a bound on the running time, we need to bound the number of iterations of the loop in FIND COVER. Again this lemma is a slight variation of Clarkson's.

Lemma 4.2.4. *The number of successful iterations of the loop in* FIND COVER *before a cover is found is bounded above by* $\frac{2k}{2k-3}c\lg(n/c)$ *which is* $4c\lg(n/c)$ *for* $k = 2$, $1501c\lg(n/c)$ *for* $k = 1.501$.

Proof. At each iteration where weights are doubled, $w(U)$ increases by a factor of $1 + \frac{1}{kc} < e^{1/kc} < 2^{\frac{3}{2kc}}$. After I successful iterations, $w(U) < n2^{\frac{3I}{2kc}}$. But by Lemma 4.2.2, U_F contains an element of C whose weight is doubled. Letting t_p be the number of times the weight of p double, after I successful iterations, $w(C) \geq \sum_{p\in C} 2^{t_p}$, with $\sum_{p\in C} t_p = I$. But by convexity of the exponential function $w(C) \geq c2^{I/c}$. But $w(C) \leq w(U)$, hence

$$\begin{aligned}
c2^{I/c} &< n2^{\frac{3I}{2kc}} \\
\Rightarrow I\left(\frac{1}{c} - \frac{3}{2kc}\right) &< \log\frac{n}{c}
\end{aligned}$$

Letting $k = 3/2 + \epsilon$, we get

$$I \leq \frac{2k}{2k-3}c\lg(n/c) = 1 + \frac{3}{2\epsilon}.$$

But for $k = 2$, this implies

$$I < 4c \lg \frac{n}{c},$$

and for $k = 1.501$, this implies

$$I < 1501c \lg \frac{n}{c},$$

as required. □

A similar lemma also appears in [BG94a]. This implies together with Lemma 4.2.3 that the expected number of iterations of the loop in FIND COVER is $O(c \lg(n/c))$. It is interesting to note the tradeoff in the constants between the constant in the running time, and the constant k, which in turn determines the minimum cover size for which these lemmas are valid.

These lemmas hold for small values of c, which makes the algorithm useful for small size covers. Of course $c > d$, since we cannot have a cover of any smaller size unless the input is highly degenerate.

We can already mention here the component of the final running time incurred by finding and reweighting the points of U_F. Finding points among the n points of U that are on one side of a hyperplane can be done in time $O(dn)$ with a trivial algorithm, giving a bound of $O(cdn \log(n/c))$ for this component of the running time. This is repeated $O(c \log n/c)$ times over the entire algorithm. For fixed d, the whole process can be done in expected time $O(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))})$ using sophisticated data structures for half-plane range queries [Mat92], as described in [Cla93]. Finally, note that the expected number of oracle calls (half-plane emptiness tests) is $O(fc \log(n/c))$ where f is the number of facets of conv R. This can be avoided when L is a polytope, as described below.

4.3 Particular Measures

In this section we describe several implementations of FIND BAD FACET and give the corresponding running times of the entire algorithms.

MinCover-B:
We are given a ball $\mathcal{B}(r)$ of radius r centered at the origin, and we would like to find an

approximation of its minimal cover among the points of U. To do this, we use OPTIMAL COVER, but we define FIND BAD FACET to test whether the ball is contained in the convex hull of the random sample R. This can be done by computing the convex hull of R and finding a facet that is at a distance of less than r to the origin. This takes time $O(|R|^{\lfloor d/2 \rfloor})$ for fixed $d \geq 4$ (or $O(|R| \log |R|)$ if $d \leq 3$) [CS89, Cha93]. For variable d, we can use the algorithm in [AF92], which finds f facets of the convex hull of n points in dimension d in time $O(ndf)$.

As mentioned before, lemmas 4.2.2, 4.2.3, and 4.2.4 apply when we replace the polytope L by a ball $\mathcal{B}(r)$, and this guarantees correctness of the algorithm.

Since the expected number of iterations is $O(c \log(n/c))$ and the expected size of R is $4cd \ln c$, the expected running time of OPTIMAL COVER for $\mathcal{B}(r)$ is:

$$O\left(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + c \log(n/c)(c \log c)^{\lfloor d/2 \rfloor}\right)$$

for fixed d, or

$$O\left(ncd \log(n/c) + cd^2 2^d \log(n/c)(4cd \log c)^{\lfloor d/2 \rfloor + 1}\right)$$

for any d (and non-degenerate convex hull of the covers) using the result in [AF92] and the crude bound on the complexity of the convex hull of $d2^d v^{\lfloor \frac{d}{2} \rfloor}$ obtained from the upper bound theorem [MS71, Mul93] as follows. Let $f(j)$ be the number of j-faces of the a polytope with n vertices, then the Upper Bound Theorem says that

$$f(j) \leq \sum_{i=j}^{\lfloor \frac{d}{2} \rfloor} \binom{i}{j} \binom{v-d+i-1}{i} + \sum_{i=\lfloor \frac{d}{2} \rfloor + 1}^{j} \binom{i}{j} \binom{v-i-1}{d-i},$$

and we need an upper bound on $g = \sum_{j=0}^{d} f(j)$. Replacing the second and fourth combinations by $v^{\lfloor \frac{d}{2} \rfloor}$, we get

$$\begin{aligned} g &\leq v^{\lfloor \frac{d}{2} \rfloor} \sum_{j=0}^{d} \sum_{i=j}^{d} \binom{i}{j} \\ &\leq v^{\lfloor \frac{d}{2} \rfloor} d 2^d. \end{aligned}$$

MinCover-P:

Clarkson's original algorithm solves this approximation problem. The geometric object to be covered is a fixed polytope P with ℓ vertices containing the origin. Let the *positive* side of a facet of a polytope be the side containing the polytope. We need to find a facet F of conv R such that there is a vertex of P on the side of F not containing the origin, *i.e.* on its negative side, if P is not contained in conv R. For each point p of P, we find the facet F of conv R whose intersection point q with the line op maximizes $q \cdot p$. If p is on the the negative side of F, we return F. If no such facet exists it is easily seen that $P \subset \text{conv } R$.

Each step above can be done in $O(LP(|R|, d)) = O(LP(4cd \log c, d))$ time via linear programming. This can be done as follows.

Consider a set of points Q, and a query point q. We wish to find the facet of conv Q which is intersected by the ray stemming from the origin $\overrightarrow{0q}$. Consider the dual of Q, which is a set of hyperplanes $\mathcal{D}(Q)$. Call $\mathcal{D}^+(Q)$ the intersection of the half-spaces containing the origin and delimited by the hyperplanes of $\mathcal{D}(Q)$. In the dual, by well known properties of this duality, moving a point p along the ray $\overrightarrow{0q}$ from the origin in the direction of q corresponds to moving a hyperplane $\mathcal{D}(p)$ with normal $\overrightarrow{0q}$ from infinity towards the origin 0. When p hits a face $F = \text{conv}\,\{v_1, \ldots, v_k\}$ of conv Q, $\mathcal{D}(p)$ hits a face $f = \bigcap_{i=1}^{k} \mathcal{D}(v_i)$. When this happens, let r be the distance of $\mathcal{D}(p)$ to the origin, $r = 1/\|p\|$. Points x of f are points of $\mathcal{D}^+(Q)$ which optimize $\overrightarrow{0q} \cdot x$ since for these points x, the hyperplane $\mathcal{D}(x)$ is a supporting hyperplane for $\mathcal{D}^+(Q)$. Now define a linear program with linear constraints given by the faces of $\mathcal{D}^+(Q)$. A linear programming query with optimization direction $\overrightarrow{0q}$ returns a point v of f, which corresponds to a facet F in the primal. Hence we can test if $q \in \text{conv } Q$ using linear programming. Furthermore we also know $\lambda_Q(q) = \max\{\lambda > 0 : \lambda q \in \text{conv } Q\}$ since $\lambda_Q(q) = \frac{1}{r\|q\|}$. If $\lambda_Q(q) > 1$, q is outside conv Q. This will be particularly useful later for the **MaxScale-P** problem.

Coming back to our original problem, applying this to each vertex q of P and polytope conv R given as a set of points, we can determine whether $P \subseteq \text{conv } R$ using ℓ such linear programming queries, one for every vertex of P. The expected running time of FIND BAD FACET is then $O(\ell d^3 c \log c + \ell e^{O(\sqrt{d \log d})})$ using the linear programming algorithm of [Gol95]. The entire OPTIMAL COVER algorithm runs in expected time

$$O\Big(ncd\log(n/c) + \ell c^2 d^3 \log c \log(n/c) + \ell c \log(n/c) e^{O(\sqrt{d\log d})}\Big).$$

For fixed d, we can also use linear programming queries [MS92a]. For any arbitrary positive δ, each query takes time $O((n\log^{2d+1} n)/t^{\gamma})$ after a preprocessing step taking $O(t^{1+\delta})$ time and space ($n \le t \le n^{\lfloor d/2 \rfloor}$). By trading off preprocessing time with query time, the queries can be answered [Cla93] in $O(\ell^{1+\delta}(\ell c)^{1/(1+\gamma/(1+\delta))})$ time. Since there are $O(c\log(n/c))$ iterations on average, the entire algorithm then runs in an expected time of

$$O\left(n^{1+\delta} + c\log(n/c)\ell^{1+\delta} + c\log(n/c)(\ell c)^{1/(1+\gamma/(1+\delta))} + (nc)^{1/(1+\gamma/(1+\delta))}\right),$$

which is $O(c^2 n^{1+\delta})$.

An alternative is as follows. For fixed d, we can also use linear programming queries using the very recent batched version of Chan [Cha95]. For any arbitrary positive δ, a batch of ℓ queries on a set of m half-spaces takes time

$$O\Big(m\log\log m + m\log\ell + (m\ell)^{1-\frac{1}{\lfloor d/2\rfloor+1}} + \ell\log^{1+d} m\Big).$$

By replacing m by $4cd\log c$, the expected size of the cover, we get

$$O\Big(c\log c\log\log c + c\log c\log\ell + (c\log c\ell)^{1-\frac{1}{\lfloor d/2\rfloor+1}} + \ell\log^{1+d}(c\log c)\Big).$$

Multiplying by $4c\lg(n/c)$, the expected number of iterations, and adding the time for the half-space queries, we get an expected time of

$$O\Big(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + c^2\log n\log c\log\log c + c^2\log n\log c\log\ell + \\ c\log n(c\log c\ell)^{1-\frac{1}{\lfloor d/2\rfloor+1}} + c\log n\ell\log^{1+d}(c\log c)\Big).$$

This is also $O(n^{1+\delta}c^2 n\log n\log c)$.

MaxScale-B:

Let $r^*(s)$ be the radius of the largest ball centered at the origin and contained in a cover of size s. In this problem we are given a desired cover size c, and we would like to find

$r^*(c)$. We will not use OPTIMAL COVER but skip directly to FIND COVER. Here again we get only an $O(d \log c)$ approximation to this cover. The corresponding version of FIND BAD FACET will be to use FIND BAD FACET as for the **MinCover-B** problem, but always return the facet *closest* to the origin, *i.e.* never say that a ball of some radius has been covered. We should also remember the maximum distance to the origin of the closest facet at each iteration, and the corresponding cover. Here again we do only $4c \lg(n/c)$ successful iterations, *i.e.* $O(c \log(n/c))$ calls to FIND BAD FACET. This will guarantee that the version of FIND COVER for **MaxScale-B** finds a cover of size at most $4cd \ln c$ containing a ball of radius $r^*(c)$. This can be seen as follows. By Lemma 4.2.4 we are certain to find a cover if it exists. Hence if $\mathcal{B}(r^*(c))$ is the largest ball in covers of size c, a cover of size $4cd \ln c$ containing $\mathcal{B}(r^*(c))$ must be found. The running time is the same as for the **MinCover-B** approximation.

MaxScale-P:

The solution to this problem is very similar to the previous case. We use FIND COVER and modify FIND BAD FACET so that it determines the largest possible scaling factor λ such that $\lambda P \subseteq \text{conv}\, R$. This is done by ℓ linear programming queries as in **MinCover-P**, but in addition we compute $\lambda_{\text{conv}\, R}(v)$ for each vertex of P. Taking the minimum of these over all ℓ vertices of P gives a value $\lambda(R)$ for this cover. This value along with the corresponding facet is returned. This value corresponds to the largest scaling P can undergo while still remaining inside conv R.

After the required number of iterations in FIND COVER is done, we stop and return the maximum of all scaling factors, $\lambda^*(c)$. The running time for this version is the same as for **MinCover-P**, and the same correctness argument also applies here.

We summarize both results in

Theorem 4.3.1. *The versions of* FIND COVER *for* **MaxScale-B** *(resp.* **MaxScale-P***) find a cover of size at most $4cd \ln c$ containing a ball of radius $r^*(c)$ (resp. P scaled by $\lambda^*(c)$) in time identical to their* **MinCover-B** *and* **-P** *counterparts.*

MaxScale-L and **MinCover-L**:

When $L = \mathcal{B}^k(1)$, the scaling factor of $\mathcal{B}^k(1)$ in a polytope with f facets is easily computable in time $O(df) = O(kf)$. For general L and fixed d, if a hyperplane emptiness query for L can be answered in time h, the total expected running time for the **MinCover-L** approximation is

$$O\Big(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + hc\log(n/c)(c\log c)^{\lfloor d/2 \rfloor}\Big),$$

and if in addition, given a direction u, the supporting hyperplane for L with normal u which is the furthest in that direction, can be obtained in time h, the above time bound is also valid for the **MaxScale-L** approximation problem.

4.4 An Extension

It is possible to extend the framework to the following case: instead of a set of points U, we have a set of planar polygons. (An extension to polytopes of higher dimensions is possible, but the complexity of the algorithm increases, and our application requires only polygons.)

This corresponds to the situation in grasping where we allow fingers to be placed anywhere on ∂B. For simplicity, consider here d fixed. We have seen in Corollary 1.3.2 that for a face F of ∂B, , (F) is a planar polygon in $\mathbb{R}^6$. We can assume these polygons are triangles, since one can triangulate the faces of B in linear time [Cha91], and this does not increase the asymptotic complexity of B. We now replace the points of U by the set of triangles G_i corresponding to triangles T_i of facets of ∂B. Say B has n facets. If v is a vertex of T_i, , (v) is a vertex of G_i (the normal at v is the normal of T_i.) Now given a hyperplane h which corresponds to a facet of conv R, we intersect this hyperplane with each triangle. This creates a planar arrangement on this triangle. Each region in this arrangement is weighted by its area. Weights of regions on the negative side of h are doubled, as before. To randomly pick a point for R, we can simply pick a region G_i according to its new weight $w(G_i)\text{area}(G_i)$ as was done with points of U before, then we select any point of this region, say its centroid. This technique can be used for both **MaxScale-L** and **MinCover-L** problems. It is not clear however how to best select the point from a region: it might be advantageous to optimize, say the size of the inscribed ball inside conv R, given that we have already selected each region.

At each iteration, we have an expected number of at most $O(|R|^{\lfloor \frac{d}{2} \rfloor})$ new hyperplanes, hence over the entire run of the algorithm, the expected number of hyperplanes is at most $O((c \log c)^{\lfloor d/2 \rfloor} c \log(n/c))$. This gives a bound on the total number of regions of $O\left(n[(c \log c)^{\lfloor d/2 \rfloor} c \log(n/c)]^2\right)$, since there are $O(n)$ triangles. The running time of all the variations of the problems mentioned remain the same, with n replaced by the above number. However we do not know how to analyze the performance of this algorithm in terms of optimality of cover size, or the size of the inscribed ball, for example.

4.5 Concluding Remarks

A bottleneck in the ball version of the approximation problems is finding the facet of conv R closest to the origin. It would be interesting to find a faster method. One way might be to approximate the residual radius using the algorithm of [KMY92].

The algorithm in this chapter treats the frictionless case, or the frictional case, when the friction cones are approximated as described in Section 2.1.3. It would be interesting to devise a variation of this algorithm which does not require such an approximation.

Chapter 5

Reactive Control

Up to now, our approach to grasping has been to assume an accurate model of the object to be grasped and from such a model, an offline geometric algorithm determines a set of grip points, where the fingers are then placed. As we have already seen, this approach has been used in [MSS87, PSS$^+$93] for example. Once the grip points have been determined, the geometry of the object is deemed irrelevant and the grasp is determined and maintained by only controlling the magnitudes of the forces at the grip points. However, such a picture has not proven very useful, as in practice, obtaining such a model might not be feasible, or the exact location of the object might not be available. The main success in dexterous manipulation seems to have come from two directions: (1) telemanipulation, where a human in the loop uses much more sensory information than what is assumed theoretically necessary, and (2) simple parallel jaw grippers, where grasping algorithms can be made immune to lack of any sensory information. For example Goldberg [GF93, Gol91] constructed a parallel jaw gripper with the property that friction in the transversal direction between a jaw of the gripper and the grasped object is removed (or greatly reduced.) A similar "sensorless" strategy has been used by Rao [Rao93] to determine the shape of the object being grasped. Here, we take an approach, where the sensory information is selectively used to determine movement of the robot ("state-transitions") so that ultimately a grasp will be achieved independent of the quality of the sensory information or the reliability of the actuator movements. Furthermore, unlike

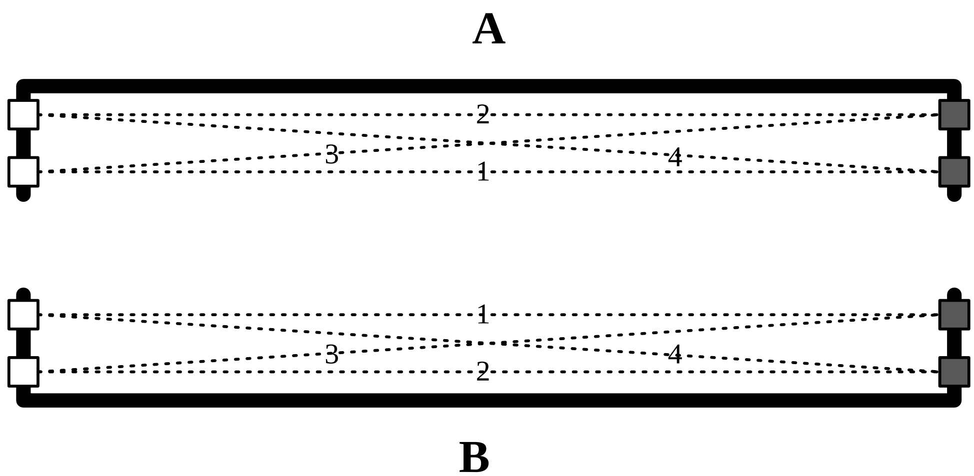


Figure 5.1: Parallel jaw gripper with sensors.

Goldberg's algorithm, our algorithms and devices are non-disturbing in the sense that it does not cause any movement of the objects except for the movements desired and can grasp objects that are kinematically constrained.

Our "algorithms" are rather different from what is usually understood by algorithms: the robots (i.e. actuators and sensors) themselves are active components of the algorithm, performing "computations" much like an analog computer in conjuncts with a digital computer. For this reason, the "algorithm design" also implies a careful choice of the participating actuators and sensors and their placement. For this reason, the algorithm does not separate in any meaningful way phases of "digital planning" interrupted by "analog actuation and sensing". Thus our reactive algorithms are much smoother in operation.

5.1 Reactive Parallel Jaw Grasping

In this section we consider a standard parallel jaw gripper and attach to each jaw two infrared light emitting diodes (IR LEDs) and two infrared detectors as depicted in Figure 5.1.

This allows us to have in fact 4 light beams which we can test for being interrupted or not by alternately illuminating the IR LEDs.

The general scheme will be to close the gripper while constantly checking the status of the sensors, and performing actions according to that status in a table driven manner.

These algorithms can be used as low level primitives for grasping applications, for example in automated manufacturing. With minor modifications, the algorithms can also be used for determining the shape of (the convex hull of) a planar part, again without disturbing the part. In contrast, Rao's scheme takes a multitude of gripping actions for obtaining information about the shape, and even then, it is not possible to reconstruct the shape in some cases. This is due to the fact that the object it moved during each grasping action.

5.1.1 Related Literature

Our work shares some similarities with Goldberg's gripper [Gol91] and is motivated by the same paradigm as Canny and Goldberg's "RISC" robotics [CG94].

In [Gol91], Ken Goldberg has modified the basic gripper to reduce the friction between the gripper and the object being grasped. This friction occurs when the gripper is being closed on an object, and this object is forced to rotate. Several results exist on finding grasps for such a parallel jaw gripper, when the object is known [RG94, RG92]. Recent results consider closing the gripper on an unknown object, and measuring the jaw distance for various orientations [Rao93]. This permits the determination of the object, but also subjects it to many motions. Another method involves grasping in random orientations to distinguish planar parts [Gol91].

In the Computational Geometry literature, there are several results on probing. Probing using finger probes [CY87], and line and other probes [DEY86] is considered. The object is to construct a model of the probed object. We are mainly interested in line probes: a line is swept (either in parallel or rotating) until the object is hit. The probe returns the position of the line where it hit the object first. Also the method of "rotating calipers" [Tou83] where a pair of parallel supporting lines rotates around a polygon has many algorithmic applications in geometry, and is closely related to the algorithm described here.

A reactive grasping algorithm for a special type of multi-fingered hand is given in [MLH93], but it uses a gripper mounted camera and a vision system. Like ours, their algorithm controls the position of the gripper interactively according to the current sensory data, but the algorithms are inherently more complex.

We propose to add probing capabilities to a basic parallel jaw gripper and to combine probing techniques with grasping using light beam sensors and proximity sensors. The goal is to produce algorithms for obtaining grasps with as little knowledge as possible about the object, and without touching the object. Recently we have devised reactive algorithms for 2 and 3 finger planar grasping using some geometric properties of minimum area triangles (for 3 fingers), where the fingers are equipped with proximity sensors. First, we present grasping algorithms for the extended parallel jaw gripper equipped with light beam sensors as described below.

5.1.2 The Parallel Jaw Algorithm

The Diameter Function

Consider a fixed polygonal object P, and define the function $d(\theta)$ to be the diameter of P in the direction θ. This is defined as the length of the projection of P (a line segment) on a line that forms an angle of θ with the x-axis. Antipodal points pairs correspond to local maxima, and local minima correspond to possible grasps. Figure 5.2 shows a polygon and its diameter function.

The current configuration of the gripper consists of

- The gripper orientation,
- The inter-jaw distance, and
- The position of (the center of) the gripper.

The position of each pair of parallel beams can be represented by a point in the graph of the diameter function. Rotations of the gripper correspond to moving the point parallel to the θ axis (horizontally), and opening and closing of the gripper to moving vertically.

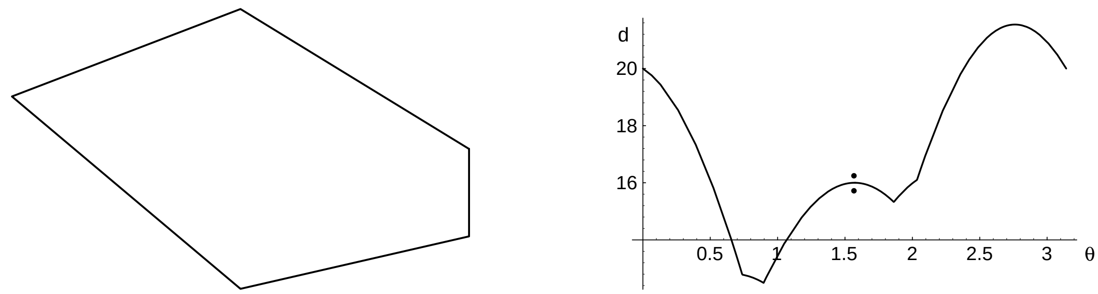


Figure 5.2: A polygon and its diameter function.

The four pairs of beams can be represented by four points in a '+' configuration. This last representation is not precise however since the translational component of the beam positions is not modeled.

The Algorithm

We assume that the initial orientation of the gripper is at angle 0 (with the x-axis, say), and that the gripper rotates around its center of symmetry.

Let A_1, A_2, A_3, A_4 be the state of the light beams on jaw A, and B_1, B_2, B_3, B_4 be the same for jaw B. In the following, *up*-ward translation means translation perpendicular to the gripper jaws and towards jaw A, *down* is towards jaw B. This also defines naturally *left* and *right* translation. Also we have a *current direction*, which is clockwise initially, and the initial position is such that the object is somewhere between the gripper jaws.

We need another primitive: *Back-up* will move the gripper to the position it was before performing the previous rotation.

The algorithm checks the current state and performs an action according to the table above, where an 'x' indicates that the corresponding beam is interrupted. The action is performed until the state changes. This is performed in a tight control loop, and hence the motions are very small. We assume that initially the object being grasped in somewhere between the parallel jaws.

The algorithm essentially attempts to keep the top of the object boundary between beams 1 and 2 of one jaw, and the bottom between the corresponding beams of the other

Algorithm 1

State	A_2	A_1	B_1	B_2	*Action*
0	.	.	.	.	close the gripper
1	.	x	.	.	move up
2	.	.	x	.	move down
3	.	x	x	.	rotate in current direction
4	x	x	x	.	move up (parallel to jaw before previous rotation)
5	.	x	x	x	move down (parallel to jaw before previous rotation)
6	x	x	.	.	move up
7	.	.	x	x	move down
8	x	x	x	x	On first entry: reverse rotation direction, back-up Otherwise: stop

jaw (state 3), and rotates the gripper while the diameter decreases.

For example, at the beginning we are in state 0 and the gripper closes until either state 1 or 2 is reached. Then a small translation occurs and immediately we are back in state 0. An initial direction of rotation is chosen arbitrarily, which might be reversed in state 8.

Let D_{12} be the sum of the distances between beams 1 and 2, for each jaw.

Let θ be a local minimum of the diameter function d, and consider the plane containing the graph of the diameter function. Define a *valley of height h around θ* as the set of points (x, y) that are above the graph of d satisfying:

$$\max(d(x), d(\theta)) \leq y \leq d(\theta) + h$$

$$\max_{x<\theta}\{d(x) \leq d(\theta)\} \leq x \leq \min_{x>\theta}\{d(x) \leq d(\theta)\}.$$

The constraint on x says that the valley stops after d dips below $d(\theta)$. See Figure 5.3. Then we have the following

Theorem 5.1.1. *For a flat polygonal object that has significantly smaller diameter than the length of the jaws, Algorithm 1 will terminate with a gripper orientation which is in* *the valley of height D_{12} that is closest to the initial orientation, and in the direction of initial decreasing slope. We say it finds the closest local minimum up to its resolution.*

Proof. At each step, we either close the gripper or translate/rotate so that the gripper can close further. We can think of this algorithm as tracking the diameter curve using two vertically positioned test point, at a distance of D_{12} apart. The bottom point corresponds to the inner pair of beams, the top point to the outer pair, and we want to keep the curve between the two points (state 3). When the bottom point gets above the curve we are in state 0, and we close (go down in the diameter function graph). When the top point moves below the curve we are in state 8. States 1,2,4,5,6,7 take care of translating the gripper. This is not modeled by the diameter graph.

To show that the algorithm converges to a local minimum, we show that at any step we make progress. In state 0 we advance down in the diameter graph, and in state 3 we advance in the chosen direction. This direction can changes only once, when the initial direction increases the diameter graph. The other states, except state 8, reduce the number of interrupted beams by translating the gripper in an appropriate direction and hence eventually allow us to get to state 0 or 3. States 4 and 5 need some explanation. These are reached after a rotation. (We assume that the gripper closes with sufficient precision so that beam 1 is always interrupted before beam 2 and we detect both events separately.) It can be seen that if we translate parallel to the jaws as they were before the previous rotation, we bring the object closer to the center of the gripper.

State 0 and 3 may increase the number of interrupted beams but we make progress by closing the gripper or rotating further, respectively. When state 8 is reached for the first time, it might be because we have chosen a wrong initial direction, and the diameter increases during that rotation. This is the only case where we rotate with all 4 beams interrupted. State 8 can only be entered from states 3, 4 and 5. If this is the first entry to state 8, we simply retrace our steps (back-up) to the state just before this state was entered. We can now perform rotations again, but in the other direction. Step 8 thus reverses the search direction and continues. Upon second entry to state 8 we are certain of having crossed a local minimum, and the diameter function increases by at least D_{12}, and we are done. □

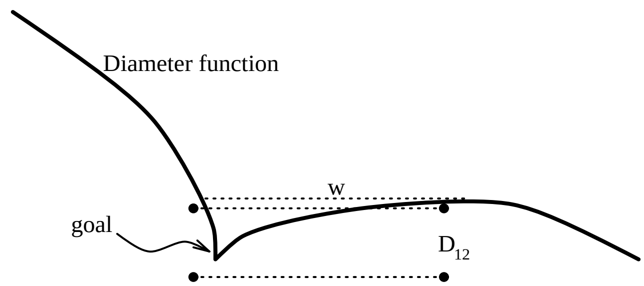


Figure 5.3: Angular error due to distance between beams.

This algorithm will miss any "small" local minima whose height is smaller than D_{12}. This is the limit of the *resolution* of the gripper. As we decrease D_{12} to zero, the algorithm misses smaller and smaller local minima, and ultimately tracks the diameter curve with total precision. This is physically impossible, so we accept a small error.

This error implies however that we don't have any maximum angle error guarantee since we can overshoot by a large angle if the diameter function increases slowly, see Figure 5.3. For example for an oval-like polygonal shape, if the difference between the minimum and the maximum diameter is too small, we might miss the minimum.

Improving accuracy

We previously had an alternative algorithm which would allow us to improve accuracy. It was based on a binary like search. Another algorithm used the crossed beams to control the rotation direction. This work has been reported in [TM94]. However in the course of experimentation, we have found that it is possible to obtain a "virtual" pair of parallel beams using a *single* LED–detector pair. Furthermore, these can be made as close to each other as desired, subject to the precision of the analog to digital converters which are used with the IR detectors to obtain the strength of the IR radiation coming from the LED. This is done by having various thresholds for the received intensity (see the section on implementation.) We therefore implemented this algorithm using this virtual pair of light beams. In this new scheme, accuracy as described does not pose a problem.

Oscillations

It is conceivable that at some point due to inaccuracies or time delays, the algorithm would oscillate between a pair of "opposite" states (ex. 1 and 2, or 4 and 5). This could also happen if the beam distance is exactly the diameter of the polygon in the current orientation. This can be avoided by checking if no gripper closing has occurred between those states and then forcing the algorithm to go to state 0 (close).

5.2 The Case of Three Fingers.

In this section we consider the same type of objects as in Section 5.1.2, but we present a reactive algorithm for a three finger hand. The sensors used are different however. Again, we use the local geometry of the object to find a set of grip points and we also solve the local motion planning problem of getting the fingers to their grasping positions. We assume in addition that the object is convex and that its boundary is smooth. Our algorithm is also non-disturbing.

Recently, Mirtich and Canny [BM94] presented algorithms for finding 2 and 3 finger grasps for known objects, optimizing various criteria. It is easy to adapt our ideas to produce reactive algorithms for generating their planar grasps reactively. Independently, Erdmann [Erd94] discusses a closely related algorithm for two finger grasps which uses sensors similar to ours.

Here however we consider three fingered grasping and two fingered grasping.

We consider a gripper that consists of 3 fingers whose endpoints move in a plane. The fingers can move to arbitrary positions within certain bounds, but their order around the triangle they are forming must remain the same. We search for the gripping points by following the object boundary in a way that is dictated by the boundary until some geometric conditions are satisfied, which will then guarantee a grasp.

Each finger is equipped with simple sensors, as described in the next section, which allow the fingers to follow the contour of the object to be grasped, and locally determine the angle of the object boundary it is close to.

Sensors

Each finger is equipped with the following sensors:

1. An omni-directional distance sensor, which returns the distance of an object in any (planar) direction. Its range can be very small, only what is required for tracking the boundary of an object that lies within close proximity to the finger.

2. An angle sensor. If the finger is close to an object as defined by the previous sensor, we require the sensor to return the angle of the object boundary at the closest point.

These sensors can be implemented by using for example an array of distance sensors similar to the optical sensor by Okada [Oka82b] or the sensor of [BSB93]. Preliminary experiments indicate that it is possible to use a pair of simple IR reflective sensors, placed a small distance apart, and measure the difference between reflected light intensity. However more work is needed.

Primitives

In order to describe our algorithm at a high level we assume that the following primitive can be called upon. The main basic operation our algorithm will need is the following of the object boundary by a finger. This has already been implemented on various robotic platforms, see for example [Jen92].

5.2.1 The Three Finger Reactive Algorithm

Our algorithm for finding a three fingered grasp is based on the paradigm of finding a locally minimal area triangle that encloses the object. A geometric algorithm for convex polygons can be found in [OAMB86].

Theorem 5.2.1 (Klee [Kle86]). *If T has a locally minimum area among triangles containing a convex body B, then the midpoint of each side of T touches B.*

It can also be shown [OAMB86] that if the midpoint of an edge e of a triangle does not touch the object, then e can be perturbed such that its midpoint after the perturbation

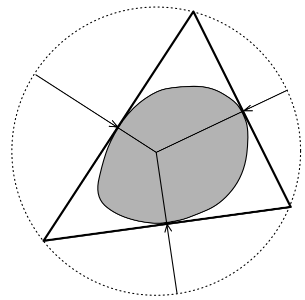


Figure 5.4: Forces applied at edge midpoints meet.

lies inside the original triangle, and that this perturbation reduces the triangle area. This is precisely how the algorithm operates. In essence we minimize a potential function defined by the triangle area, whereas in Section 5.1.2, we minimize the distance between the two jaws of the parallel jaw gripper.

Given such a triangle, we note that the lines through its edge midpoints and perpendicular to their corresponding edges are concurrent at the point which is the center of the circle circumscribing the triangle. See Figure 5.4. If we place the three fingers at the edge midpoints, we get a planar force closure grasp that does not rely on friction. The finger forces will be chosen such that they sum to 0. Planar torque closure is obtained, if we allow for sufficient friction.

We now describe the algorithm. Let B be the object to be grasped. For simplicity of exposition, when we say a finger is at point p of the boundary, it is in fact in close proximity of the object, and does not touch it. (It is on the boundary of a slightly enlarged object. The point p will be the closest point to the finger on the boundary of the original object.) The tangent to the object at that point is still well defined, by convexity. Note that given three such finger positions, the sensors at the fingers indicate the object tangents, so we know the triangle they form.

Phase 1

In the first phase we find some triangle that contains the object. We assume the object lies somewhere "between" the fingers, such that when we close the fingers, say along three

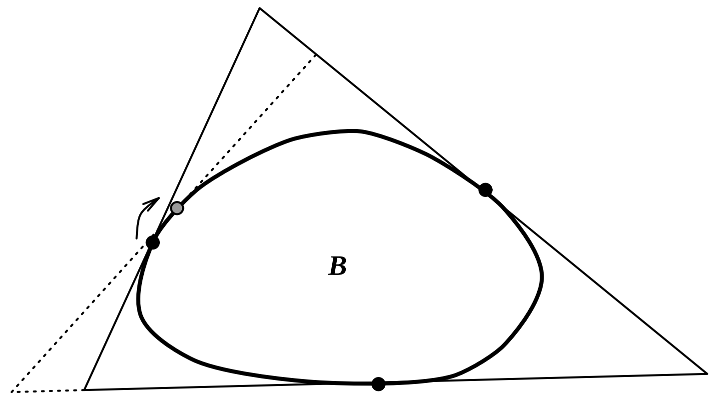


Figure 5.5: Steps of the algorithm.

concurrent lines at angles of 120 degrees with each other, each finger comes in proximity with the object.

Let f_0, f_1, f_2 be the corresponding points that lie on the boundary of B. Let t_0, t_1 and t_2 be the object tangents at those points. If the tangents form a triangle the first phase is done. Otherwise we pick an unbounded edge and "rotate" it by movings its finger along B in the unbounded direction. This procedure must lead to a finite triangle. It is also possible to arrive at a triangle with all but one edge midpoints touching, by keeping the distance of say f_1 and f_2 to t_0 the same.

Phase 2

In the second phase we find a locally minimal area triangle enclosing B. The procedure is simple: since we are looking only for a local minimum in the area, we can use local methods. Each finger divides its triangle edge into two segments. Consider the ratios of the largest segment to the total edge length. This number is between 0 and 1, and is 1/2 if the finger is at the edge midpoint. Pick the edge whose finger is proportionally the furthest from the midpoint, i.e. whose ratio is furthest to 1/2. Then move the corresponding finger along the object boundary, in the direction of the largest segment. This will reduce the ratio and it also reduces the area of the triangle since the new midpoint moves inside the old triangle. See Figure 5.5.

We iterate this step until all fingers are at their edge midpoints. At this point the normal lines at the finger points are concurrent and we found a grasp. Convergence is guaranteed by the fact that the area of the triangle decreases during every move, so an infinite loop does not occur.

5.2.2 Dealing With Uncertainty

After some point the algorithm will reach the limit of the sensor resolution and will oscillate between various nearby positions close to the local minimum. This can be solved by accepting a small error in the finger positions so that they are close to the edge midpoints. The cases where the algorithm will be the most sensitive to errors will be when the triangle is very skinny. Then a small motion of a finger will have larger consequences on the edge length and its relative position. However, since we assume some degree of friction, we simply assume that the friction is sufficient to allow for the forces to be concurrent, irrespective of sensor angular error.

This algorithm does not work for non-convex objects since for such objects the midpoint of an edge of a locally minimum area triangle does not necessarily lie on the object boundary. It might still be possible to use it locally, and we plan to investigate this. If we allow the fingers to grasp at object vertices, and if we consider that at a vertex all the possible normals between the normals of the adjacent edges are present, then this algorithm can be used on polygonal objects.

5.2.3 A Variation

Mirtich and Canny's three finger algorithm finds offline the largest equilateral triangle containing the object and the fingers are placed at the triangle edge contact points. This optimizes resistance to torques normal to the grip plane. A reactive algorithm for finding such three points can be described as follows. Again we start with all three fingers somewhere on the boundary. Fix f_0, and move the other two fingers around the boundary until the tangents form an equilateral triangle. Then rotate all the fingers around the boundary, keeping the angles between their normals 120 degrees, until the normal at f_0 has made a rotation for 120 degrees. Then move back to the position giving the largest

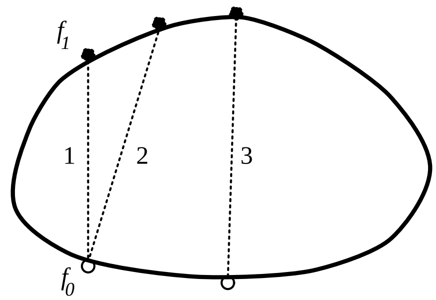


Figure 5.6: A few steps of the two finger algorithm.

area formed by the triangle t_0, t_1, t_2.

5.2.4 Two Finger Algorithm

This section describes a simple reactive algorithm for finding a two finger grasp. The hand model is as follows. The two fingers slide on an axis that is parallel to the grasping plane, and let the midpoint of the two fingers be the center of the gripper. Call the *finger normal* the direction from the finger to the midpoint. Here we assume similar sensors as previously but we do not require angle measurement, only the information on whether the angle formed by the inward pointing normal to the object at the finger position and the finger normal is 0, positive or negative. Again, we assume that the object has a smooth boundary and is convex.

There must exist a pair of points on the boundary of the object such that the normal lines at both points are in fact the same line. This can be seen by considering the pair of supporting lines, one on each side of the object, and rotating this pair until it reaches a local minimum. Then the contact points satisfy the above requirement.

We find such a pair of points as follows (refer to Figure 5.6.)

We assume that initially the object is somewhere between the two fingers. The fingers close until one of the sensors indicate proximity. Call this finger f_0, the other f_1. Then the gripper is rotated until the finger normal of f_0 is normal to the object. Such a rotation will in absolute value be by less than 90 degrees. If during this rotation f_1 becomes close to B, then it starts to track the boundary of B in the direction away from f_0. Otherwise, close the fingers along their axis (translating the center if necessary) until f_1 becomes close to B as indicated by its sensor.

Now to find the grasping points as described above, we "rotate" the gripper by following B's boundary using f_0, keeping its axis normal to B, and f_1 also following B. Then the grasp is found when f_1's normal is also normal to B. This must happen by a comment above. Note that we don't need to rotate by more than 360 degrees.

In this case also, there is a two finger reactive algorithm for finding the optimal grasp in the sense of Mirtich and Canny [BM94]. We can find the pair of points such that the distance between them is maximized by rotating as above by 180 degrees and noting all the grasp positions and the distance between the fingers for each grasp.

5.2.5 Concluding Remarks

It remains to be seen how these algorithms perform in practice. Also many variations are possible. For example, we can replace the proximity and angle sensors by force sensors, but we lose the non-disturbing quality.

It would be interesting to find algorithms that require less sensory information, for example require only the sign of the angles at the fingers instead of the exact angle for the three finger case. This would make the algorithm more robust and simplify the sensors. Generalization of these algorithms to non convex objects would be desirable.

Finally, theorem 5.2.1 is in fact valid in any dimension, (if we replace midpoint by centroid of a facet). Therefore it seems likely that these techniques can be extended to three dimensional objects and four fingers.

Chapter 6

An Implementation

6.1 The Setup

To verify the practicality of the algorithm for reactive grasping developed in Section 5.1, we have constructed a prototype of the parallel jaw gripper with sensors used in that section, and implemented the reactive algorithm. The gripper was mounted in the chuck of a robotic milling machine, MOSAIC, developed at New York University [Ash90]. A picture of the gripper alone and mounted on MOSAIC can be seen in Figure 6.1.

The MOSAIC machine is a three-axis, knee-type milling machine with pulse-width modulation motor drivers. It has a horizontal table which can move horizontally in two (xy) directions, and a spindle which can move up or down (in the z direction), and rotate around a vertical axis. Its motion controllers are installed in a VME chassis housing a 68020 CPU. The VxWorks real-time operating system runs on this CPU. The controllers can also servo the machine tool spindle. This is the special feature of MOSAIC which allows us to use it instead of a robot arm, since such an arm was not available. The only difference is that it is the table which moves, instead of the arm.

We augmented the existing software to contain an implementation of the algorithm of Section 5.1.2. The motion primitives which we used were only absolute and relative moves, for the table and for the spindle, and an *abort* command which stopped the currently executing move command.

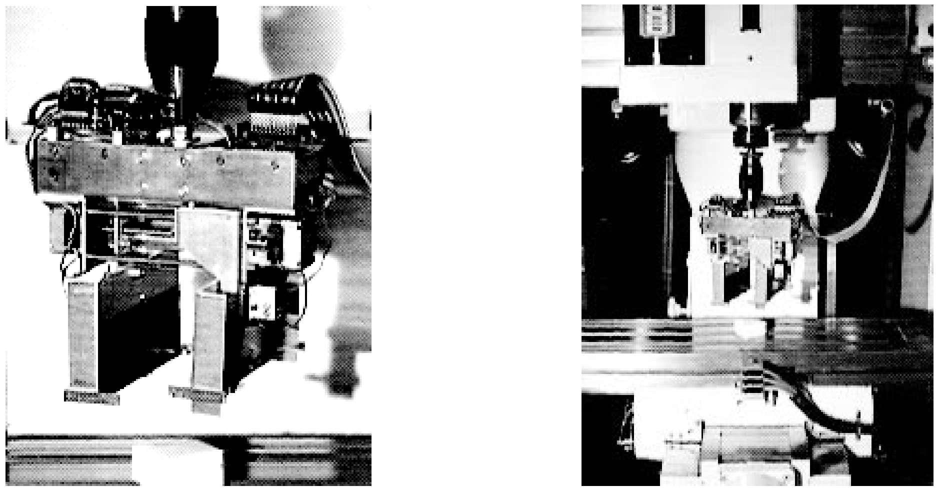


Figure 6.1: Reactive parallel jaw gripper.

6.1.1 The Reactive Gripper

Our gripper consists of three main components. A Motorola MC68332BCC single board micro-controller with 64Kb of memory and 'Time Processor Unit', a custom interface board between the micro-controller and the gripper hardware, and the gripper hardware itself. The micro-controller is connected through a 9600bps line to the VME computer.

The gripper itself consists of two platforms or 'jaws' which can move *independently*. Let's assume that the motion occurs in the xy-plane. This is in contrast with most existing grippers, and can be used to advantage to reduce the number of interactions between the gripper micro-controller and the robot. It also allows more precise control over jaw position relative to the object to be grasped. Figure 6.2 illustrates the design and shows some of the dimensions.

On each jaw a pair of infrared LED's (Motorola MLED 930) is mounted on one side, and a pair of infrared photo-transistors (Motorola MRD 300) is mounted on the other side, as depicted in Figure 6.3(a). This setup allows us to implement four light-beams, two parallel and two crossed, as depicted in Figure 5.1. We do this by turning on each LED in sequence and reading the level of infrared radiation arriving at the photo-transistors. In the current implementation, we do not use the crossed beams, however.

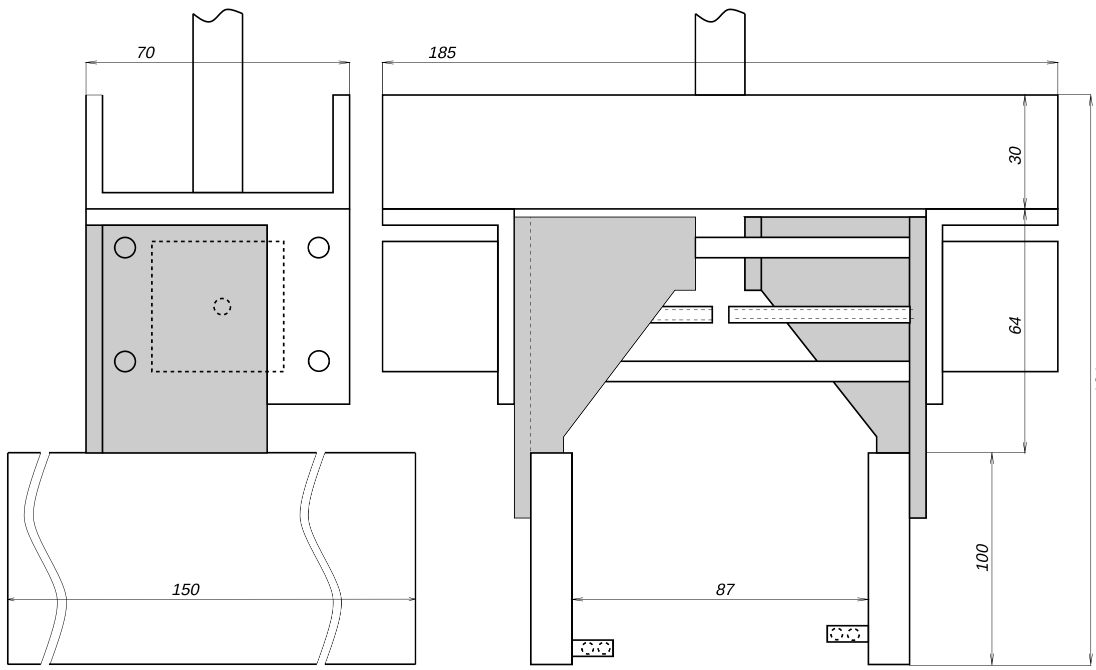
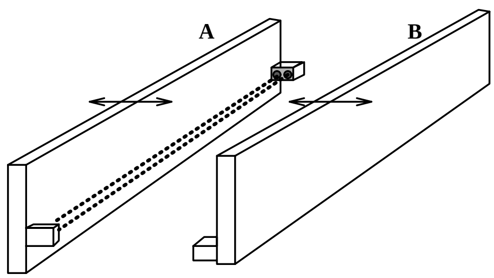
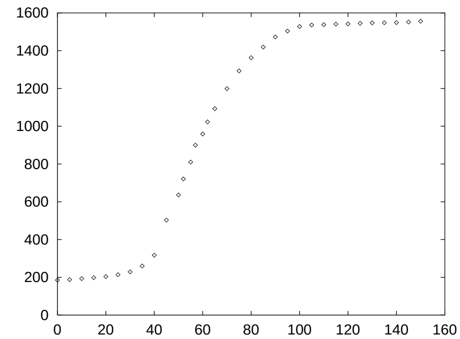


Figure 6.2: The reactive gripper.



(a) Light beam setup.



(b) A/D readings per distance traveled (10^{-3} inch).

Figure 6.3: Infrared light beams.

The photo-transistors' output are read through an 8 channel, 12 bit, serial analog to digital (A/D) converter (Maxim MAX186) connected directly to a serial channel of the micro-controller. A patent for this gripper and the algorithm is being applied for by NYU [TM95].

The software running in the micro-controller, which we have written, reads the A/D outputs periodically, at a frequency of 2kHz in the background, using a periodic interrupt feature of the Time Processor Unit of the MC68332BCC. The use of an A/D converter allows us to measure "how much" a beam is broken. The distance along which this can be done is approximately 1.5mm. In other words, we can detect the presence of an opaque object between a LED/photo-transistor pair, and measure the distance of this object to an imaginary line connecting the centers of the LED and the photo-transistor, if this distance is less than about 0.75mm. The precision of the A/D converter allows us in effect to implement two (or more) parallel light beams, in the xy-plane. This is demonstrated by a simple experiment. Consider beam A1 fixed, and an opaque object is moving in a direction perpendicular to A1 and in the xy-plane. The axis of motion goes through the midpoint between the two sensors of A1. Figure 6.3(b) shows a graphs of A/D readings for beam A1, as a function of the distance traveled by the object. The region between

40 and 100 thousands of an inch can be used for implementing the multiple beams.

The platforms are driven by two TEC type SPH-35B-12TBR stepper motors with a resolution of 1.8 deg/step. The stepper motor shaft is a threaded lead screw which drives the platforms. The combined axial travel of both fingers is 87mm. The interface to the stepper motors and to the A/D is mounted directly on the gripper. The micro-controller is sufficiently small to be also mounted on the gripper, which would permit a minimal interconnection between the gripper and the robot: a serial line and power. This is not currently done however.

The software running on the micro-controller monitors the status of the beams, and outputs through the serial line any change of status. It can also respond to queries about the status, and other commands such as *open* and *close* jaws, *follow object* (the light beams, hence the jaws, constantly track the object boundary), *initialize* (the jaws are opened until limit switches activate to zero position counters), and others. Another command is to close (or open) until a change in the beam status occurs. This is the command we use to implement the reactive algorithm. When the change occurs, the beam status is output on the serial line. The software on the VME computer monitors this information, and follows the table given in Section 5.1.2. For example when a rotate state is entered, an appropriate rotate command is performed on MOSAIC. When the light beams change status, an abort is performed for the rotate command.

The *follow object* command can be used to obtain the shape of (the convex hull of) the object, since any change in the jaw position is reported. This information, along with the current position of the gripper can be combined in a trivial way to obtain an approximation of the shape. We have not done experiments to determine the precision of the resulting polygon since this gripper is only a prototype.

6.1.2 The Experiment

In this experiment, the distance between the two light beams on each jaw was quite large (0.8mm). The main difficulty encountered was a non-trivial amount of jitter in the spindle axis. This is probably due to the fact that the controller for this axis is optimized for torque control and not position control. This caused the light beams to constantly change states, which is a problem only in state 8. Averaging successive readings and incorporating hysteresis solved the problem. We kept a log file which contained, for each state transition, the x, y coordinates (in inches in the coordinate system of the machine tool), as well as the orientation of the gripper (in revolutions), and a number indicating the width of the opening between the jaws (in the number of step-motor steps). Figure 6.4 shows graphs corresponding to the above parameters, as a function of the transition number, for one run of the algorithm. The first figure shows the initial position of the object with respect to the gripper. The initial direction of rotation was clockwise. We can see that there is an initial approach where the gripper (actually the table) is only translating (state 1)—this corresponds to the straight horizontal segment in the position graph. This phase is followed by a set of rotations with some translations, and then state 8 is entered for the first time and a reversal occurs, at transition number 159. Finally the gripper quickly reaches the other side of a valley in the diameter graph when the second transition (number 215) to state 8 occurs.

Using the *follow object* command also worked well, as this reduced interaction between the gripper and the machine.

6.1.3 A simulation

A simulation in MATHEMATICA was also written. This of course performed flawlessly. A few representative configurations of a sample run are shown in Figure 6.5. The last picture is a superposition of all the displayed configurations. The dot indicates the center of the gripper, and the lines represent the light beams, with a dashed line when the beam is broken.

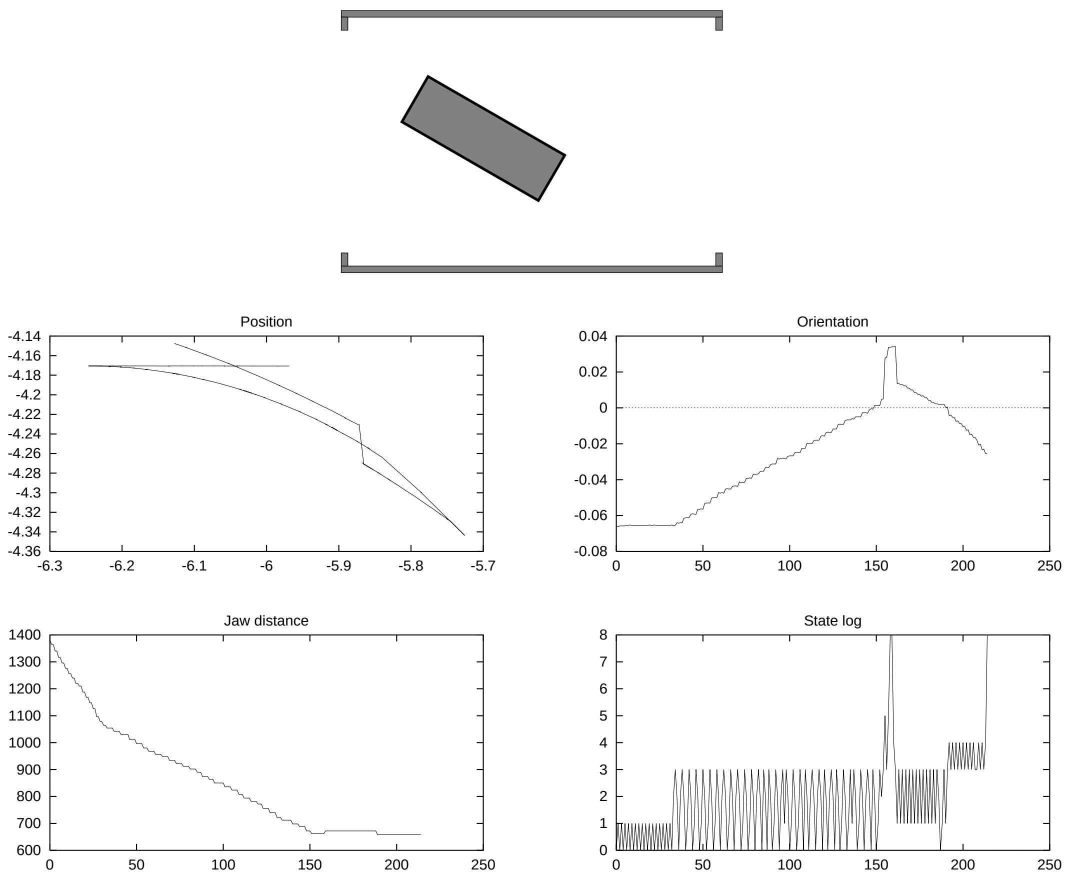


Figure 6.4: One run of the reactive algorithm.

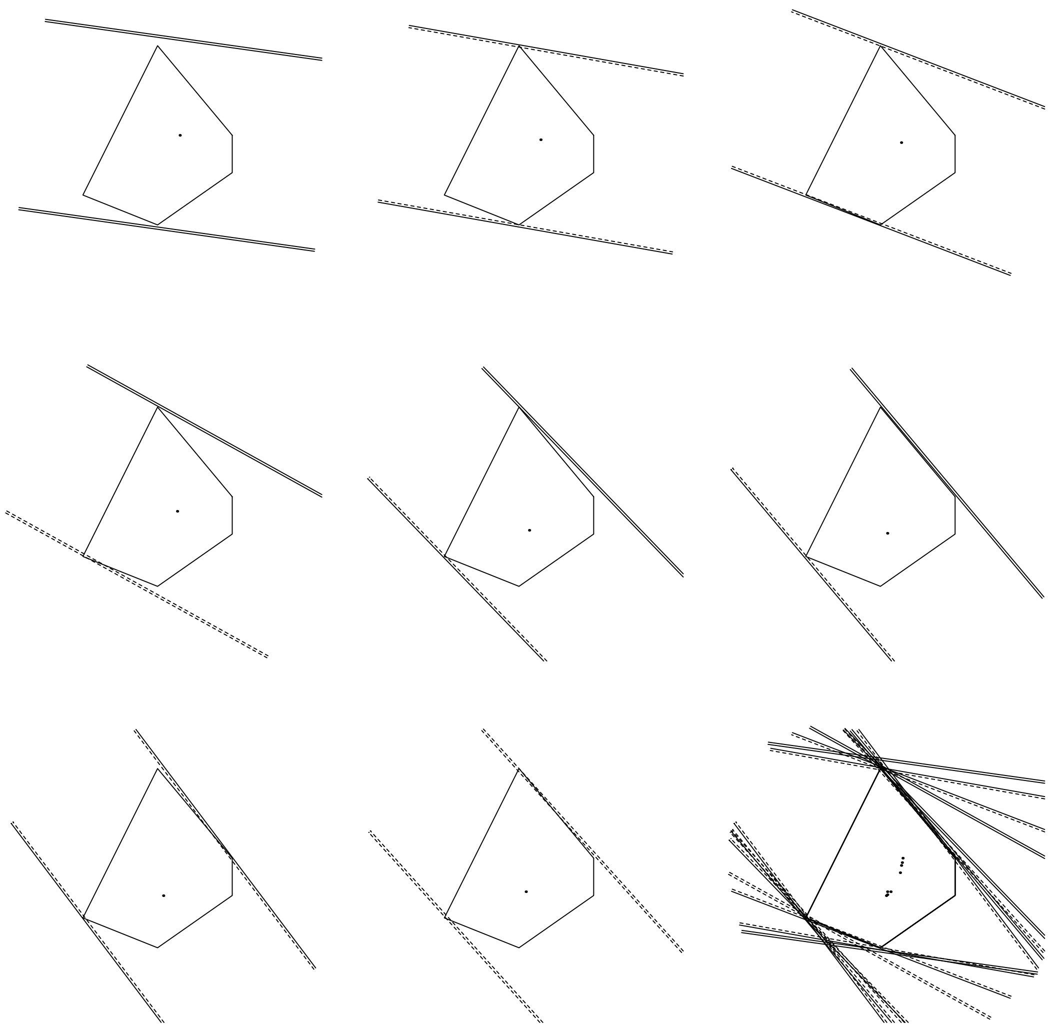


Figure 6.5: One run of the simulation.

Chapter 7

Conclusion and Open Questions

In this dissertation we explore the connections between fundamental facts in discrete geometry and grasping theory. We have extended—in a natural way, and with justifications—the current grasping theory, including grasp quality measures, to the frictional case, and pointed out some problems with one such existing extension. A new robust grasp quality measure which avoids problems with changes in object coordinate system origin was also introduced, as well as methods for its efficient computation. This measure does not vanish as does the residual radius based measure when the origin of the coordinate system is translated. Also, scaling of the coordinate system is explored, but more work is required. In addition, we compare various definitions of immobility: first-order, second-order, finite immobility and analogous notions in the field of graph rigidity. It is hoped that the analysis given will bring closer together the field of computational geometry, and that of robotics. We have attempted to make the exposition readable for readers in both areas.

We have devised a simple algorithm for computing three finger optimal grasps using a restricted version of the r_{con} measure. We allow fingers to be placed anywhere on the boundary of the object. The resulting grips are force closure and immobilizing in a geometric sense. They are also torque closure if we allow friction, no matter how small. We believe a similar algorithm for four finger grasps might be possible using the full r_{con} measure.

Existing algorithms for finding almost optimal grasps [KMY92] requires an unrealistically large number of fingers. We propose a randomized algorithm which allows one to use a substantially smaller number of fingers. The algorithm is in fact a purely geometric algorithm for set covering. We can find the almost-optimal cover of a convex set efficiently, as well as determine the almost-optimal scaling of this set, such that it is still inside the cover of a given size.

On the implementation side, we have attempted to meet J. Hollerbach's challenge: to actually implement a grasping algorithm. The goal was to devise a robust algorithm which uses little sensory input, *i.e.* no camera to detect the object's shape, and little knowledge of the object. Our inspiration came from the *probing* literature in computational geometry. Initially in fact, our ideas centered around probing algorithms of the type used in computational geometry for determining a polygon's shape. We use line probes, which are implemented as simple infrared light beams. This approach led us to design a *reactive* paradigm for the control of a parallel jaw gripper and multi-finger hand. We have designed and built a parallel jaw gripper which embodies this paradigm.

Many open questions remain, both in the theoretical domain and in the practical domain. We mention some of them below.

- The complexity of finding optimal grasps is still not well understood. This includes finding an optimal m finger frictionless grasp, given a finite set of allowable points on the object, and *a fortiori* when this set is infinite, such as when fingers can be placed anywhere on the object. It is believed that this problem is intractable [KMY92], even in the finite case. However, approximation algorithms do exist, both deterministic [KMY92], and randomized, such as the one given in chapter 4.

- Although our grasp efficiency measure can be used in the presence of friction, as we have shown, there is another parameter which can be considered for grasp quality: the coefficient of friction itself. The goals of optimizing the residual radius—a lower bound on how large forces can be resisted, and minimizing the coefficient of friction needed to achieve a given grasp quality, are contradictory. It is not clear how to balance the two goals.

- We have ignored the issue of planning the motion of the multi-finger hand from its initial configuration to the grasping configuration, which is not unique in general. The integration of motion planning algorithms with grasp synthesis algorithms remains largely unexplored.

- During the course of a complex manipulation it is likely that regrasping is necessary. We have addressed only the issue of finding a new placement for one finger. However finding an entire finger gait: a sequence of finger configurations such that the hand can go from one to another, while satisfying closure conditions is still to be studied.

- On the geometric side, we believe that the randomized framework given still has potential for more applications and generalizations. A good candidate for example would be the addition of friction.

Appendix A

Some Terminology

A.1 Geometry

A d-dimensional space, $\mathbb{R}^d$, equipped with the standard linear operations, is said to be a *linear space*.

If equipped with the Euclidean distance function, denoted by $\|\cdot\|_2$ or $\|\cdot\|$, it is a *Euclidean space*. In a Euclidean space, we denote by $\mathcal{B}^d$ the full dimensional ball centered at the origin, and by $\mathcal{S}^{d-1}$ the boundary of $\mathcal{B}^d$.

1. A *linear combination* of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $\mathbb{R}^d$ is a vector of the form
$$\lambda_1\mathbf{x}_1 + \cdots + \lambda_n\mathbf{x}_n,$$
where $\lambda_1, \ldots, \lambda_n$ are in $\mathbb{R}$.

2. An *affine combination* of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $\mathbb{R}^d$ is a vector of the form
$$\lambda_1\mathbf{x}_1 + \cdots + \lambda_n\mathbf{x}_n,$$
where $\lambda_1, \ldots, \lambda_n$ are in $\mathbb{R}$, with $\lambda_1 + \cdots + \lambda_n = 1$.

3. A *positive (linear) combination* of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $\mathbb{R}^d$ is a vector of the form
$$\lambda_1\mathbf{x}_1 + \cdots + \lambda_n\mathbf{x}_n,$$
where $\lambda_1, \ldots, \lambda_n$ are in $\mathbb{R}_{\geq 0}$.

4. A *convex combination* of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $\mathbb{R}^d$ is a vector of the form

$$\lambda_1 \mathbf{x}_1 + \cdots + \lambda_n \mathbf{x}_n,$$

where $\lambda_1, \ldots, \lambda_n$ are in $\mathbb{R}_{\geq 0}$ with $\lambda_1 + \cdots + \lambda_n = 1$.

By convention, we allow the empty linear combination (with $n = 0$) to take the value $\mathbf{0}$. We also assume that the empty linear combination is neither an affine combination nor a convex combination.

Note that affine, positive and convex combinations are all linear combinations, and a convex combination is both affine and positive combinations.

A nonempty subset $L \subseteq \mathbb{R}^d$ is said to be a

1. *linear subspace*: if it is closed under linear combinations;
2. *affine subspace (or, flat)*: if it is closed under affine combinations;
3. *positive set (or, cone)*: if it is closed under positive combinations; and
4. *convex set*: if it is closed under convex combinations.

The intersection of any family of linear subspaces of $\mathbb{R}^d$ is again a linear subspace of $\mathbb{R}^d$. For any subset M of $\mathbb{R}^d$, the intersection of all linear subspaces containing M (i.e. the smallest linear subspace containing M) is called the *linear hull* of M (or, the linear subspace *spanned* by M), and is denoted by $\operatorname{lin} M$.

Similarly, the intersection of any family of affine subspaces, or positive sets or convex sets of $\mathbb{R}^d$ is again, respectively, an affine subspace or positive set or convex set. Thus for any subset M of $\mathbb{R}^d$, we can define

1. the *affine hull* (denoted by $\operatorname{aff} M$) to be the smallest affine subspace containing M,
2. the *positive hull* (denoted by $\operatorname{pos} M$) to be the smallest positive set containing M, and
3. the *convex hull* (denoted by $\operatorname{conv} M$) to be the smallest convex set containing M.

They are also called, respectively, the affine subspace, positive set and convex set *spanned* by M.

Equivalently, the linear hull lin M can be defined to be the set of all linear combinations of vectors from M. Similarly, the affine hull aff M (respectively, the positive hull pos M, the convex hull conv M) can be defined to be the set of all affine (respectively, positive, convex) combinations of vectors from M.

A set $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of n vectors from $\mathbb{R}^d$ is said to be *linearly independent* if a linear combination

$$\lambda_1 \mathbf{x}_1 + \cdots + \lambda_n \mathbf{x}_n$$

can only have the value $\mathbf{0}$, when $\lambda_1 = \cdots = \lambda_n = 0$; otherwise, the set is said to be *linearly dependent*.

A set $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of n vectors from $\mathbb{R}^d$ is said to be *affinely independent* if a linear combination

$$\lambda_1 \mathbf{x}_1 + \cdots + \lambda_n \mathbf{x}_n \quad \text{with } \lambda_1 + \cdots + \lambda_n = 0$$

can only have the value $\mathbf{0}$, when $\lambda_1 = \cdots = \lambda_n = 0$; otherwise, the set is said to be *affinely dependent*.

A *linear basis* of a linear subspace L of $\mathbb{R}^d$ is a set M of linearly independent vectors from L such that $L = \text{lin } M$. The dimension $\dim L$ of a linear subspace L is the cardinality of any of its linear basis.

An *affine basis* of an affine subspace A of $\mathbb{R}^d$ is a set M of affinely independent vectors from L such that $A = \text{aff } M$. The dimension $\dim A$ of an affine subspace A is one less than the cardinality of any of its affine basis.

Let C be any convex set. Then by *d-interior* of C, denoted $\text{int}_d\, C$, we mean the set of points p such that, for some d-dimensional affine subspace, A, p is interior to $C \cap A$ relative to A. If c is the dim $\text{aff}\, C$, then by an abuse of notation, we write $\text{int}\, C$ to mean $\text{int}_c\, C$.

For subsets A and B of $\mathbb{R}^d$ and λ real define the *(Minkowski) sum* of A and B by

$$A + B = \{a + b : a \in A, b \in B\},$$

and let λA be

$$\lambda A = \{\lambda a : a \in A\}.$$

We shall write $A \oplus B$ instead of $A + B$ if A and B are contained in subspaces of $\mathbb{R}^d$ for which the usual direct sum exists: $A \oplus B$ is then called the *direct sum* of A and B. Call C *directly irreducible* if there is no representation of C of the form $A \oplus B$ where both A and B are different from the origin. By a decomposition theorem of Gruber, we have the result that each convex body C can be represented in the form $C_1 \oplus \cdots \oplus C_m$ where C_1, ..., C_m are directly irreducible. Such a representation is unique modulo the order of the summands.

A.1.1 Polytopes

A *hyperplane* h in $\mathbb{R}^d$ is a $d-1$ dimensional affine subspace. It can be written as $h = \{x \in \mathbb{R}^d : a^T x = b\}$, where $a \in \mathbb{R}^d$, $a \neq 0$ and $b \in \mathbb{R}$. The vector a is called the *normal* of h. A *half-space* is the set of points on one side of a hyperplane h. It is s closed half-space if it includes h, open otherwise. An example of a closed half-space is $h^+ = \{x \in \mathbb{R}^d : a^T x \leq b\}$. Hyperplane h *supports* a convex set S, if it intersects S and if $a^T s \leq b$ or $a^T s \geq b$ for every point s of S. A (convex) *polytope* is an intersection of closed half-spaces.

For $0 \leq k \leq d$, a *k-flat* is the affine hull of $k+1$ affinely independent points and has dimension k. A k-flat is also the intersection of $d-k$ hyperplanes. Thus a 1-flat is a line, a 2-flat is a plane, and a $d-1$ flat is a hyperplane.

The intersection of a polytope with a supporting hyperplane is a k-face if the affine hull of this intersection has dimension k. A 0-face is a *vertex*, a 1-face is an *edge*, and a $d-1$ face is a *facet*. Two points of a polytope P are *antipodal* if there are two distinct supporting hyperplanes of P with the same normal such that each contains its respective point.

A.2 Screw Theory

A *screw* is defined by a straight line in three-dimensional Euclidean space, called, its *screw-axis* and an associated *pitch*, p. A screw is represented by a six-dimensional vector, $\mathbf{s} = (S_1, S_2, S_3, S_4, S_5, S_6)$, known as the *screw coordinates*. The screw coordinates are interpreted in terms of the *Plücker line coordinates*, (L, M, N, P, Q, R), of the screw axis, as follows:

$$\begin{aligned} L &= S_1, \\ M &= S_2, \\ N &= S_3, \\ P &= S_4 - pS_1, \\ Q &= S_5 - pS_2, \\ R &= S_6 - pS_3, \end{aligned}$$

where L, M and N are proportional to the direction cosines of the screw axis, and P, Q and R are proportional to the moment of the screw axis about the origin of the reference frame (i.e. the cross product of a vector from the origin to a point on the axis and a unit vector, directed along the screw axis). The pitch of the screw is then given by

$$p = \frac{S_1S_4 + S_2S_5 + S_3S_6}{S_1^2 + S_2^2 + S_3^2},$$

and the magnitude of the screw is given by

$$|\mathbf{s}| = \begin{cases} \sqrt{S_1^2 + S_2^2 + S_3^2}, & \text{if } p < \infty; \\ \\ \sqrt{S_4^2 + S_5^2 + S_6^2}, & \text{if } p = \infty. \end{cases}$$

A *unit screw* is a screw with unit magnitude. Scalar multiplication and vector addition are valid for infinitesimal screws, and the screws are closed under these operations. Thus the six-dimensional space of infinitesimal screws forms a vector space. See for example [Sal82].

Sometimes, we simply consider the 2-norms of a screw (as a six-dimensional vector), disregarding its pitch:

$$|\mathbf{s}|_2 = \sqrt{\sum_{i=1}^{6} S_i^2}.$$

Given two screws $\mathbf{s}' = (S_1', S_2', S_3', S_4', S_5', s_6')$ and $\mathbf{s}'' = (S_1'', S_2'', S_3'', S_4'', S_5'', S_6'')$, we define their *virtual coefficient* as

$$\mathbf{s}' \odot \mathbf{s}'' = S_1'S_4'' + S_2'S_5'' + S_3'S_6'' + S_4'S_1'' + S_5'S_2'' + S_6'S_3''.$$

Note that the operation '$\odot$' is a commutative operation from $\mathbb{R}^6 \times \mathbb{R}^6$ into $\mathbb{R}$.

Two screws $\mathbf{s}'$ and $\mathbf{s}''$ are said to be

1. *reciprocal*: if their virtual coefficient is zero, i.e. $\mathbf{s}' \odot \mathbf{s}'' = 0$,
2. *repelling*: if their virtual coefficient is strictly positive, i.e. $\mathbf{s}' \odot \mathbf{s}'' > 0$, and
3. *contrary*: if their virtual coefficient is strictly negative, i.e. $\mathbf{s}' \odot \mathbf{s}'' < 0$.

An ensemble of screws is known as a *screw system*, and is defined by a set of $n \leq 6$ independent *basis screws*. The *order* of a screw system is equal to the number of basis screws required to define it; such a system is also called an *n-system*. The order of a screw system reciprocal to an n-system is $(6 - n)$.

With an infinitesimal rigid motion of an object in three-dimensional Euclidean space there is an associated screw called *twist* such that the body rotates about and translates along its screw axis. The screw coordinates of a twist are given by $\mathbf{t} = (T_1, T_2, T_3, T_4, T_5, T_6)$, where the first three components T_1, T_2 and T_3 correspond to the angular displacement (or angular velocity), $\overline{\omega}$, of the body and the last three components T_4, T_5 and T_6 correspond to the translational displacement (or translational velocity), $\overline{v}$, of a point fixed in the body and lying at the origin of the coordinate system. The pitch of the twist is given by

$$p = \frac{\overline{\omega} \cdot \overline{v}}{\overline{\omega} \cdot \overline{\omega}}.$$

The pitch of the twist is the ratio of the magnitude of the velocity of a point on the twist axis to the magnitude of the angular velocity about the twist axis. If the pitch of a twist is zero then the twist corresponds to a pure rotation, and if the pitch of a twist is infinite then the twist corresponds to a pure translation. The magnitude of the twist is given by

$$|\mathbf{t}| = \begin{cases} \|\overline{\omega}\|_2, & \text{if } p < \infty; \\ \\ \|\overline{v}\|_2, & \text{if } p = \infty. \end{cases}$$

Similarly, with any system of forces and torques acting on a rigid object in three-dimensional Euclidean space there is an associated screw called *wrench* such that the system of forces and torques can be replaced by an equivalent system of single force along the wrench axis and a torque about the same wrench axis. The screw coordinates of a wrench are given by $\mathbf{w} = (W_1, W_2, W_3, W_4, W_5, W_6)$, where the first three components W_1, W_2 and W_3 correspond to the resultant force, $\overline{f}$, acting on the body along the wrench axis and the last three components W_4, W_5 and W_6 correspond to the resultant torque, $\overline{\tau}$, acting on the body about the wrench axis. The pitch of the wrench is given by

$$p = \frac{\overline{f} \cdot \overline{\tau}}{\overline{f} \cdot \overline{f}}.$$

The pitch of the wrench is the ratio of magnitude of the torque acting about a point on the axis to the magnitude of the force acting along the axis. If the pitch of a wrench is zero then the wrench corresponds to a pure force, and if the pitch of a wrench is infinite then the wrench corresponds to a pure moment. The magnitude of the wrench is given by

$$|\mathbf{w}| = \begin{cases} \|\overline{f}\|_2, & \text{if } p < \infty; \\ \\ \|\overline{\tau}\|_2, & \text{if } p = \infty. \end{cases}$$

Note that the virtual coefficient of a twist $\mathbf{t} = (\overline{\omega}, \overline{v})$ and a wrench $\mathbf{w} = (\overline{f}, \overline{\tau})$ is

$$\mathbf{w} \odot \mathbf{t} = \overline{f} \cdot \overline{v} + \overline{\tau} \cdot \overline{\omega},$$

the rate of change of work done by the wrench $\mathbf{w}$ on a body moving with the twist $\mathbf{t}$.

If a twist $\mathbf{t}$ is reciprocal to a wrench $\mathbf{w}$, then the wrench does no work when the body is displaced infinitesimally by the twist. Thus for two reciprocal screws, a twist about one of the screws is possible while the body is being constrained about the other screw. Similarly, if $\mathbf{t}$ is repelling to $\mathbf{w}$, then positive work is done by the constraining wrench when the body is displaced infinitesimally by the twist. This implies that the twist can be accomplished, but then the contact of the wrench will be definitely broken. Lastly, if $\mathbf{t}$ is contrary to $\mathbf{w}$, then negative (virtual) work must be done by the constraining wrench when the body is displaced infinitesimally by the twist. This implies that such a displacement is impossible, if we assume that the objects being considered are all rigid.

For a given wrench system acting on a body, we say that the body has *total freedom*, if the body can undergo all possible twists, without breaking the contacts associated with the wrenches; we also say that the body has *total constraint*, if the body cannot undergo any twist, without breaking the contacts; otherwise, we say that the body has *partial constraint*.

Bibliography

[AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.

[AGSS89] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4(6):591–604, 1989.

[AHO90] M. Albertson, R. Haas, and J. O'Rourke. Some results on clamping a polygon. Technical Report 003, Dept. Comput. Sci., Smith College, 1990.

[Ali95] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.

[AR79] L. Asimov and B. Roth. The rigidity of graphs, 2. *J. Math. Anal. Appl.*, 68:171–190, 1979.

[Asa79] H. Asada. *Studies on Prehension and Handling by Robot Hands with Elastic Fingers.* Ph.D. thesis, Kyoto University, Japan, 1979.

[Ash90] S. Ashley. A mosaic for machine tools. *Mechanical Engineering*, pages 38–43, September 1990.

[Bár82] I. Bárány. A generalization of Carathéodory's theorem. *Discrete Math.*, 40:141–152, 1982.

[BBT94] P. Bose, D. Bremner, and G. Toussaint. All convex polyhedra can be clamped with parallel jaw grippers. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 344–349, 1994.

[BDH93] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The Quickhull algorithm for convex hull. Technical Report GCG53, Geometry Center, Univ. of Minnesota, July 1993.

[BFG85] B.S. Baker, S. Fortune, and E. Grosse. Stable prehension with three fingers. In *ACM symp. on the Theory of Computing*, pages 114–120, 1985.

[BG94a] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite vc-dimension. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 293–302, 1994.

[BG94b] R.C. Brost and K.Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 535–542, 1994.

[BKP82] Imre Bárány, Meir Katchalski, and János Pach. Quantitative Helly-type theorems. *Proc. AMS*, 86:109–114, 1982.

[BM94] J. Canny B. Mirtich. Easily computable optimum grasps in 2-d and 3-d. In *IEEE International Conference on Robotics and Automation*, pages 739–747, San Diego, CA, May 1994.

[BMK94] D. Baraff, R. Mattikalli, and P. Khosla. Minimal fixturing of frictionless assemblies: Complexity and algorithms. Report CMU-RI-TR-94-08, The Robotics Inst., Carnegie Mellon University, Pittsburgh, PA, 1994.

[BSB93] A. Bonen, K.C. Smith, and B. Benhabib. Development of a robust electro-optical proximity sensor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 989–990, Yokohama, Japan, July 1993.

[CAHK87] M.R. Cutkosky, P. Akella, R. Howe, and I. Kao. Grasping as a contact sport. In *Fourth International Symposium on Robotics Research*, Santa Cruz, California, August 1987.

[Cap93] V. Capoyleas. Clamping of polygonal objects. *Pattern Recogn. Lett.*, 14:704–714, 1993.

[Car07] C. Carathéodory. Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. *Math. Ann.*, 64:95–115, 1907.

[CB92] I. M. Chen and J. W. Burdick. Finding antipodal point grasps on irregular shaped objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2278–2283, Nice, France, 1992.

[CB93] I-M. Chen and J. Burdick. A qualitative test for N-finger force-closure grasps on planar objects with applications to manipulation and finger gaiting. In *1993 IEEE International Conference on Robotics and Automation*, pages 814–820, 1993.

[CG94] J. Canny and K. Goldberg. 'RISC' for industrial robotics: Recent results and open problems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1951–1958, 1994.

[CGRW94] B. Carlisle, K. Goldberg, A. Rao, and J. Wiegley. A pivoting gripper for feeding industrial parts. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994.

[Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[Cha93] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.

[Cha95] Timothy M. Y. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1995.

[Cla86] K. L. Clarkson. Linear programming in $O(n3^{d^2})$ time. *Inform. Process. Lett.*, 22:21–24, 1986.

[Cla93] Kenneth L. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 246–252, 1993.

[CM93] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 281–290, 1993.

[Con80] R. Connelly. Rigidity of certain cabled frameworks and the second order rigidity of arbitrary convex surfaces. *Adv. in Math.*, 37:272–298, 1980.

[Cox73] H. S. M. Coxeter. *Regular Polytopes.* Dover, New York, NY, 2nd edition, 1973.

[CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[CS94] R. Connelly and H. Servatius. Higher-order rigidity–what is the proper definition? *Discrete Comput. Geom.*, 11:193–200, 1994.

[CSS92] J. Czyzowicz, I. Stojmenovic, and T. Szymacha. On a problem of immobilizing polygons. Technical Report RR 92/05–6, Université du Québec à Hull, Québec, 1992.

[CSU90] J. Czyzowicz, I. Stojmenovic, and J. Urrutia. Immobilizing a shape. Technical Report RR 90/11–18, Université du Québec à Hull, Québec, 1990.

[CSU91] J. Czyzowicz, I. Stojmenović, and J. Urrutia. Immobilizing a polytope. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes in Computer Science*, pages 214–227. Springer-Verlag, 1991.

[Cut85] M.R. Cutkosky. *Robotic Gaping and Fine Manipulation.* Kluwer Academic Publishers, Massachusetts, 1985.

[CY87] R. Cole and C. K. Yap. Shape from probing. *J. Algorithms*, 8:19–38, 1987.

[DEY86] D. P. Dobkin, H. Edelsbrunner, and C. K. Yap. Probing convex polytopes. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 424–432, 1986.

[DJ90] G. Das and D. Joseph. The complexity of minimum convex nested polyhedra. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 296–301, 1990.

[DS90] D. P. Dobkin and D. L. Souvaine. Computational geometry in a curved world. *Algorithmica*, 5:421–457, 1990.

[EC92] I. Emiris and J. Canny. An efficient approach to removing geometric degeneracies. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 74–82, 1992.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, Heidelberg, West Germany, 1987.

[Erd94] M. Erdmann. Understanding action and sensing by designing action-based sensors. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994. Workshop on Design of Parts and Devices for Automation.

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. In *1992 IEEE International Conference on Robotics and Automation*, pages 2290–2295, Nice, France, 1992.

[GF93] K.Y. Goldberg and M.L. Furst. Low friction gripper, a design modification for the parallel-jaw gripper that improves grasp stability for polyhedral parts. U.S. Patent Number 5,186,515, February 1993.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Algorithms and Combinatorics. Springer-Verlag, 1988.

[Gol91] K.Y. Goldberg. *Stochastic Plans for Robotic Manipulation.* Ph.D. thesis, School Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1991.

[Gol95] M. Goldwasser. A survey of linear programming in randomized subexponential time. *SIGACT News*, 26(2):96–104, 1995.

[Grü67] B. Grünbaum. *Convex Polytopes.* Wiley, New York, NY, 1967.

[GW93] Peter M. Gruber and J. M. Wills, editors. *Handbook of Convex Geometry*, volume A. North-Holland, Amsterdam, Netherlands, 1993.

[HCV52] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination.* Chelsea, New York, 1952.

[HLMT90] Jiawei Hong, Gerardo Lafferriere, Bhubaneswar Mishra, and Xiaonan Tan. Fine manipulation with multifinger hands. In *1990 IEEE International Conference on Robotics and Automation*, pages 1568–1573, May 1990.

[Hun78] K.H. Hunt. *Kinematic Geometry of Mechanisms.* Clarendon Press, Oxford, 1978.

[HW90] F. Brack Hazen and Paul Wright. Workholding automation: Innovations in analysis, design and planning. *Manufacturing Review*, 3(4):224–237, December 1990.

[Jen92] J. Jennings. Model acquisition for near-sensorless manipulation using mobile robots. Invited talk, December 1992.

[Ji87] Z. Ji. *Dexterous Hands: Optimizing Grasps by Design and Planning.* Ph.D. thesis, Stanford Univ., Stanford, CA, 1987.

[JR88] Z. Ji and B. Roth. Direct computation of grasping forces for three–fingered tip–prehension grasps. *Journal of Mechanisms, Transmissions, and Automation in Design*, 110:405–513, December 1988.

[JWKB84] S.C. Jacobsen, J.E. Wood, D.F. Knutti, and K.B. Biggers. The Utah/MIT Dexterous Hand: Work in progress. *International Journal of Robotics Research*, 3(4):21–50, 1984.

[Kle86] V. Klee. Facet centroids and volume minimization. In *Fejes Tòth Festschrift*, 1986.

[KMY92] D. Kirkpatrick, B. Mishra, and C.-K. Yap. Quantitative Steinitz's theorems with applications to multifi ngered grasping. *Discrete Comput. Geom.*, 7:295–318, 1992. Also in *Proc. 22nd Annu. ACM Sympos. Theory Comput.*, pages 341– 351, 1990.

[KR86] J. Kerr and B. Roth. Analysis of multifingered hands. *International Journal of Robotics Research*, 4(4):3–17, 1986.

[Kup90] W. Kuperberg. Dimacs workshop on polytopes. Rutgers University, January 1990.

[Lak78] K. Lakshminarayana. The mechanics of form closure. *ASME 78-DET-32*, 1978.

[Lat91] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[LCS89] Z. Li, J. Canny, and S. Sastry. On motion planning for dextrous manipulation, part I: The problem formulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989.

[LHS89] Z. Li, P. Hsu, and S. Sastry. On grasping and coordinated manipulation by a multifingered robot hand. *International Journal of Robotics Research*, 8(4):31–50, 1989.

[LS88] Z. Li and S. Sastry. Task oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, RA.2-14(1):32–44, 1988.

[Mat92] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2(3):169–186, 1992.

[McM70] P. McMullen. The maximal number of faces of a convex polytope. *Mathematica*, 17:179–184, 1970.

[Meg84] N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31:114–127, 1984.

[Mey90] W. Meyer. Seven fingers allow force-torque closure grasps on any convex polyhedron. Grumman Data Systems and Adelphi University, Submitted to Algorithmica, March 1990.

[Mil93] V. Milenkovic. Robust construction of the Voronoi diagram of a polyhedron. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 473–478, Waterloo, Canada, 1993.

[Mis91] B. Mishra. Workholding – analysis, design and planning. In *IROS '91*, pages 53–57, Osaka, Japan, November 1991.

[Mis94] B. Mishra. Grasp metrics: Optimality and complexity. In K. Goldberg, D. Halperin, J.C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics*, pages 137–166. A.K. Peters, 1994.

[MLH93] T. G. Murphy, D. M. Lyons, and A. J. Hendriks. Stable grasping with a multi-fingered robot hand: A behavior-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 867–874, Yokohama, Japan, July 1993.

[MLS94] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[MNP90] X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *The International Journal of Robotics Research*, 9(1), February 1990.

[MP89] X. Markenscoff and C.H. Papadimitriou. Optimum grip of a polygon. *Int. J. of Robotics Research*, 8(2):17–29, April 1989.

[MR95] R. Motwani and P. Raghavan. *Probabilistic Algorithms.* Cambridge University Press, Cambridge, MA, 1995.

[MS71] P. McMullen and G. C. Shephard. *Convex Polytopes and the Upper Bound Conjecture.* Cambridge University Press, Cambridge, England, 1971.

[MS89] B. Mishra and N. Silver. Some discussion of static gripping and its stability. *IEEE Transactions on Systems, Man and Cybernetics*, 19(4):783–796, July/August 1989.

[MS92a] J. Matoušek and O. Schwarzkopf. Linear optimization queries. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 16–25, 1992.

[MS92b] J. S. B. Mitchell and S. Suri. Separation and approximation of polyhedral surfaces. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 296–306, 1992.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558, 1987.

[MT92] B. Mishra and M. Teichmann. On immobility. *Laboratory Robotics and Automation*, 4:145–153, 1992. Special Issue: Robot Kinematics.

[MT94] B. Mishra and M. Teichmann. Three finger optimal planar grasp. In *1994 IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Grenoble, (France), 1994.

[MU91] L. Montejano and J. Urrutia. Immobilizing figures on the plane. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 46–49, 1991.

[Mul93] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[Ngu87] V.-D. Nguyen. Constructing force-closure grasps in 3D. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 240–245, 1987.

[OAMB86] J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin. An optimal algorithm for finding minimal enclosing triangles. *J. Algorithms*, 7:258–269, 1986.

[Ohw80] M.S. Ohwovoriole. *An Extension of Screw Theory and its Applications to the Automation of Industrial Assemblies.* PhD thesis, Stanford Artificial Intelligence Laboratory, Stanford University, California, April 1980. Also Stanford Artificial Intelligence Laboratory Memo AIM-338.

[Oka82a] T. Okada. Computer control of multi-jointed finger system for precise object handling. *IEEE Transactions on Systems, Man and Cybernetics*, 1982.

[Oka82b] T. Okada. Development of an optical distance sensor for robots. *International Journal of Robotics Research*, 1(4):3–14, 1982.

[O'R90] J. O'Rourke. Computational geometry column 9. *SIGACT News*, 21(1):18–20, 1990.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994. ISBN 0-521-44592-2/Pb $24.95, ISBN 0-521-44034-3/Hc $49.95. Cambridge University Press 40 West 20th Street New York, NY 10011-4211 1-800-872-7423 346+xi pages, 228 exercises, 200 figures, 219 references. C code and errata available by anonymous ftp from grendel.csc.smith.edu (131.229.222.23), in the directory /pub/compgeom; Second Printing: Jan. 1994.

[ORSW95] Mark Overmars, Anil Rao, Otfried Schwarzkopf, and Chantal Wentink. Immobilizing polygons against a wall. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 29–38, 1995.

[PLP90] N.S. Pollard and T. Lozano-Pérez. Grasp stability and feasibility for an arm with an articulated hand. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1581–1585, Cincinatti, OH, 1990.

[PSF93] J. Ponce, D. Stam, and B. Faverjon. On computing force-closure grasps of curved two-dimensional objects. *International Journal of Robotics Research*, 12(3):263–273, 1993.

[PSS+93] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet. On characterizing and computing three and four finger closure grasps of polyhedral objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 821–827, May 1993.

[PSS+95] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet. Algorithms for computing force-closure grasps of polyhedral objects. In K. Goldberg, D. Halperin, J.C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics*, pages 167–184. A.K. Peters, 1995.

[Rao93] A. S. Rao. *Algorithmic Plans for Robotic Manipulation.* Ph.D. thesis, University of Southern California, Los Angeles, CA, 1993.

[RB94] E. Rimon and J. Burdick. Mobility of bodies in contact–I: A new second order mobility index for multiple-finder grasps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2329–2335, San Diego, CA, May 1994.

[Reu63] F. Reuleaux. *Theoretische Kinematic: Gunzüge einer Theorie des Maschinwesens.* Braunschweig, Vieweg; Also, Dover, New York, 1963.

[RG92] A. Rao and K. Goldberg. Grasping curved planar parts with a parallel jaw gripper. Technical Report 299, IRIS, August 1992.

[RG94] A. S. Rao and K. Y. Goldberg. Shape from diameter: Recognizing polygonal parts with a parallel-jaw gripper. *IRob*, 13(1):16–37, 1994.

[Rot81] B. Roth. Rigid and flexible frameworks. *American Math. Monthly*, pages 6–21, January 1981.

[Rus92] D. Rus. *Fine Motion Planning For Dexterous Manipulation.* Ph.D. thesis, Dept. of Computer Sci., Cornell University, Ithaca, NY, 1992.

[Sal82] J.K. Salisbury. *Kinematic and Force Analysis of Articulated Hands.* Ph.D. thesis, Stanford University, Stanford, CA, 1982.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience, 1986.

[SHS86] Jacob T. Schwartz, John Hopcroft, and Micha Sharir, editors. *Planning, Geometry and Complexity of Robot Motion.* Ablex Series in Artificial Intelligence, Norwood, New Jersey, 1986.

[Sil93] N. Silver. *Control of a Dexterous Robot Hand: Theory, Implementation, and Experiment.* Ph.D. thesis, New York University, New York, NY, 1993.

[Som00] P. Somoff. Über Gebiete von Schraubengeschindigkeiten eines starren Korpers bie verschiedener Zahl von Stutzflachen. *Zeitschrift für Mathematic und Physik*, 45:245–306, 1900.

[SP95] A. Sudsang and J. Ponce. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1355–1360, Japan, 1995.

[SR81] J.K. Salisbury and C. Ruoff. The design and control of a dexterous mechanical hand. In *Proceedings of the 1981 ASME Computer Conference*, Minneapolis, Minnesota, 1981.

[SR82] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated hands. *ASME J. of Mechanisms, Transmissions, and Automation in Design*, 105:35–41, 1982.

[SR89] J.M. Selig and J. Rooney. Reuleaux pairs and surfaces that cannot be gripped. *The International Journal of Robotics Research*, 8(5):79—86, 1989.

[Ste16] E. Steinitz. Bedingt Konvergente Reihen und Konvexe Systeme, 1916. J. reine angew. Math., (I) 143:128–175, 1913; (II) 144:1–48, 1914; and (III) 146:1–52.

[SV94] D. L. Souvaine and C. J. Van Wyk. Clamping a polygon. *Visual Comput.*, 10:484–494, 1994.

[SW92] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, volume 577 of *Lecture Notes in Computer Science*, pages 569–579. Springer-Verlag, 1992.

[SY87] Jacob T. Schwartz and Chee-Keng Yap, editors. *Advances in Robotics, Vol. I: Algorithmic and Geometric Aspect of Robotics.* Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1987.

[TB62] R. Tomovic and G. Boni. An adaptive artificial hand. *Trans. IRE*, AC-7:3–10, 1962.

[TM94] M. Teichmann and B. Mishra. Reactive algorithms for 2 and 3 finger grasping. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Grenoble, (France), 1994.

[TM95] M. Teichmann and B. Mishra. Reactive robotic gripper. U.S. Patent applied for, 1995.

[Tou83] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON '83*, pages A10.02/1–4, Athens, Greece, 1983.

[Tri92] J.C. Trinkle. A quantitative test for form closure grasps. Report, Department of Computer Science, Texas A&M University, College Station, TX, 1992.

[VB94] L. Vandenberghe and S. Boyd. Semidefinite programming. In *International Symposium on Math. Progr.*, August 1994.

[ZGW94] Y. Zhuang, K.Y. Goldberg, and Y.-C. Wong. On the existence of modular fixtures. In *IEEE International Conference on Robotics and Automation*, pages 543–549, San Diego, CA, 1994.