Caging Polygons with Two and Three Fingers

Caging van Polygonen met Twee en Drie Vingers

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof. dr. J.C. Stoof, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op woensdag 16 september 2009 des middags te 12.45 uur

door

Mostafa Vahedi

geboren op 22 september 1979
te Khoramshahr, Iran

Promotor: Prof. dr. M.H. Overmars
Co-promotor: Dr. ir. A.F. van der Stappen

This work was supported by the
Netherlands Organisation for Scientific Research NWO,
grant number 612.065.306.

Contents

Chapter 1

Introduction

1.1 Introduction

The term *robot* is first used by the science fiction writer Karel Čapek as an artificial superhuman. The word "robot" is derived from the czech noun "robota" meaning "labor". Nowadays we mean autonomous machines by robots. *Robotics*, coined by science fiction writer Isaac Asimov, is the study of robots. This field of study has become a vast field that involves knowledge from multiple disciplines such as electrical engineering, mechatronics, computer science, mechanical engineering, and applied mathematics.

Robotic manipulators are designed to perform a wide variety of tasks in production lines of diverse industrial sectors, such as assembly or part orienting. To perform these tasks the robot arm has to first grasp the object in a good way. Robotic grasping has a long history of more than one hundred and twenty years of research. The analysis of mechanical fixtures goes back to the work of Reuleaux [1876]. He defined *form-closure* which is a condition under which an object or a part can be completely restrained.

In a firm grasp (such as form-closure grasp) it is guaranteed that the grasped object cannot move and be moved (even with force) among the grasping fingers. A firm grasp

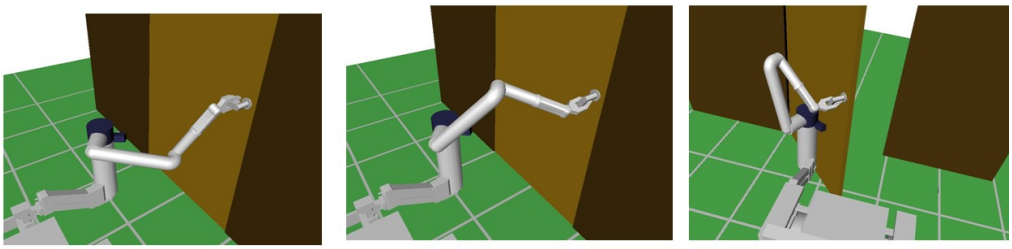


Figure 1.1: Example of an arm mounted on a wheel chair opening a door.

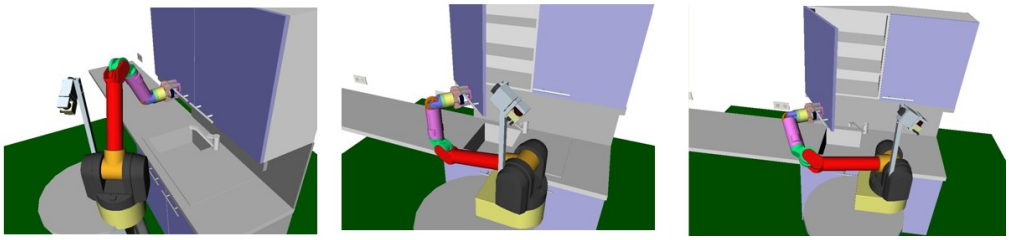


Figure 1.2: Example of a WAM arm opening a cupboard.

is not necessary for some operations. Robot researchers and designers so far have used firm grasp to perform pick and place and transportation with robotic manipulators. Human being, whom robotic researchers have tried to mimic in many areas, often uses loose grasps to perform pick and place, and also transportation. For example, to do the transportation it is not necessary to grasp the part in a way that it becomes immobilized among the grasping fingers. Instead, it is enough that the grasped object cannot escape through the fingers. In this thesis we formalize the idea of a loose grasp by the notion of caging. The amount of literature and also its short history of about ten years suggest that little is known about caging. Moreover initial results (Erickson et al.'s work [Erickson et al., 2007]) show that caging is a difficult problem.

An object is caged by a number of fingers when it is impossible to take the object to arbitrary placement far from its initial placement without penetrating any finger. We refer to a placement of a number of fingers as an arrangement of those fingers. A caging arrangement for an object is a placement of the fingers such that the object cannot escape. A caging arrangement guarantees that no matter how the object moves (translates and/or rotates) among the fingers of a manipulator, it remains within the reach of the manipulator and cannot escape. Similarly a non-caging arrangement is a placement of the fingers such that the object can be taken to arbitrary placement far from its initial placement without penetrating any finger. In this thesis we assume that the fingers are disks of radius zero or more. Figure 1.3 displays a three-finger caging arrangement and, Figure 1.4 displays a three-finger non-caging arrangement and illustrates how the polygon among the fingers can escape.

People have proposed algorithms that compute a subset of the set of whole solutions for the caging problem. People have also proposed algorithms that compute the set of whole solutions for the caging problem. In this thesis we compute the set of whole solutions for the problems we have studied. There could be many scenarios for which one might need to know the whole solution. One such scenario is that we need to transport an object grasped with fingers amidst a number of obstacles. We would like to find a certain possible grasp of the object for which the fingers avoid colliding with obstacles. Another such scenario is that we want to release a grasped object in a way that the object settles down in a specific orientation. Given all arrangements we can select one that when released settles down in a specific orientation.

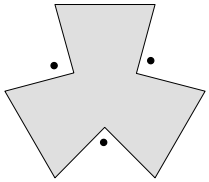


Figure 1.3: A caging arrangement of three fingers.

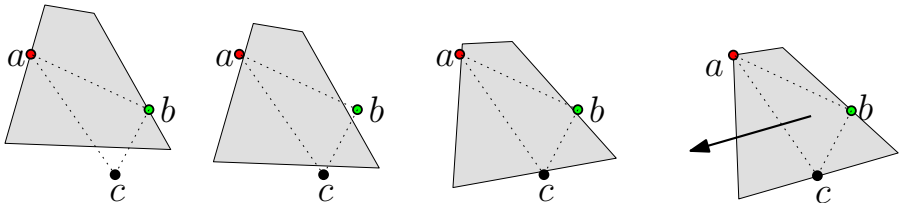


Figure 1.4: The first picture from left displays a non-caging arrangement of three fingers. From left to right we rotate and translate the polygon until it is possible to escape the polygon through the fingers which is displayed in the last picture.

It is clear that not all objects can be caged with two fingers. Convex polygons[1], for example, can always escape two fingers regardless of their placement. Figure 1.5 illustrates a non-convex polygon that cannot be caged with two fingers. Therefore, we also consider three-finger caging arrangements.

Briefly, in this thesis we study the problem of caging polygons with two and three fingers. For two-finger caging we study algorithms that report all caging arrangements by two fingers. In addition, we compute a data structure to answer whether or not a given arrangement of two fingers is caging. For three-finger caging we consider three

[1] A set in Euclidean space $\mathbb{R}^2$ is convex if it contains all the line segments connecting any pair in the set.

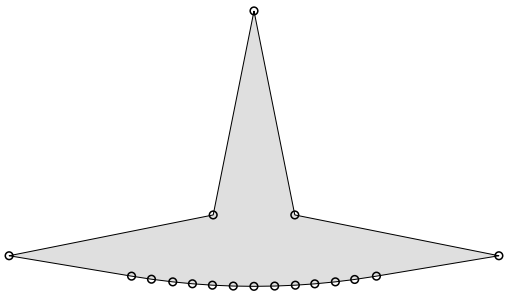


Figure 1.5: A non-convex polygons that cannot be caged with two fingers.

problems.

- First, given a polygon and placements of two fingers we compute all possible placements of the third finger that together with the two fingers cage the polygon. (Figure 1.9 shows two examples of solution sets for a convex polygon and for a non-convex polygon.) For this problem we study both convex and non-convex polygons. For convex polygons, however, we prove a better upper bound on the complexity of the solution set. We study caging convex polygons with three fingers in more detail. We consider the next two problems only for convex polygons.

- Second, given a convex polygon we compute a data structure to answer whether or not a given arrangement of three fingers is caging.

- Third, given a convex polygon we compute a data structure. With the data structure, for a given placement of two fingers —referred to as the base fingers— we compute all possible placements of the third finger that together with the two fingers cage the polygon more quickly. For this problem we consider two cases: (a) the distance between the base fingers is fixed and is given in addition to the convex polygon, and (b) the general situation in which the distance is not given.

In Sections 1.3 and 1.4 we will go into more detail about the caging problem and the exact problems we will consider.

We refer to the arrangement of the fingers that cage an object as a *caging arrangement*. In contrast, a grasp is defined as a set of contacts on the boundary of the object. Since the finger tips and the caged object do not necessarily contact each other we can safely assume that the finger tips are frictionless; our results are correct even when the friction is taken into account.

Caging has two interesting applications.

1. Cage-based manipulation: caging can be used to manipulate objects. As we fix the configuration of the manipulator, wherever we move the manipulator the caged object moves with the manipulator, because it is guaranteed that the caged object cannot escape through the fingers.

   In manipulation, caging arrangements can be used when an object should be picked up and moved to some destination. There is no need to prevent all motions of the object as it is done in the immobilizing grasps and fixturing; instead the fingers should just hold the object such that moving the fingers as a rigid body the object moves with the fingers, though the object may have some freedom among the fingers. Despite the possible imprecision in finger placement with respect to the object, an appropriate caging arrangement can guarantee that the object cannot escape through the fingers whenever these fingers move rigidly. Therefore, moving the fingers to the destination will move the object to the destination despite the relative freedom of the object among the fingers.

   Cage-based manipulation has advantages over immobilizing grasps. For an immbolizing grasp, if the fingers are slightly displaced or the shape of the object at

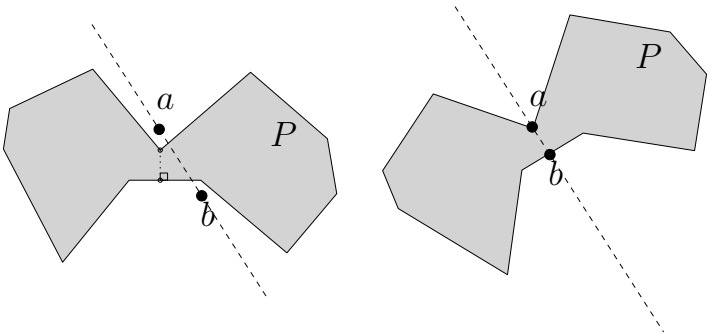


Figure 1.6: Left picture displays a two-finger caging arrangement. Right picture displays the final grasp achieved after blindly closing the fingers.

the contact points slightly changes, the resulting grasp is not immobilizing any more. Therefore, cage-based manipulation is more robust to imprecision in:

(a) object/finger position,

(b) and object/finger shape.

2. Cage-based grasping: caging can be used as a first step toward achieving a kind of firm grasp called *immobilizing* [Rimon and Blake, 1996]. From some cages there exists a way to blindly close (or open) the fingers to achieve an immobilizing grasp while preserving the cage. Figure 1.6 displays such a two-finger caging arrangement and the achieved final immobilizing grasp after closing the fingers.

Caging is related to robotics manipulation and in particular grasping. Robotics manipulation is about moving objects by machines and consists of a number of different processes such as grasping, carrying, pushing, and so on.

Caging an object suggests a number of interesting problems as follows.

- Analysis: What is the formal definition of caging? Is a given object caged with a given placements of fingers?

- Existence: What classes of objects can be caged? How many number of fingers are required to cage different objects?

- Synthesis: Given an object and a collection of fingers, can we efficiently construct a single caging arrangement, a number of caging arrangements, or even all caging arrangements of the object with these fingers?

We redefine these questions in the next section for a number of special cases.

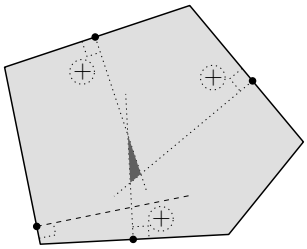


Figure 1.7: A form-closure grasp formed by four point contacts.

1.2 Grasping

A grasp is defined as a set of contacts on the boundary of the grasped object. Caging is related to the notions of form (and force) closure grasps (see for example Mason's text book [Mason, 2001]), and equilibrium and immobilizing grasps [Rimon and Burdick, 1998]. A review on grasping and related problems can be found in Bicchi and Kumar's paper [Bicchi and Kumar, 2000]. In this section we briefly define each of these kinds of grasps and the existing literature about it. In the next chapter we discuss these grasps and their formal definitions in more detail.

In manufacturing processes, immobilization of a part is used when the part needs to be held firmly and rigidly by some fixturing device, so that a machine or a human can perform industrial operations (for example drilling) on it. Clearly, the immobilization should prevent any possible motion of the part. Both form-closure grasps and second-order immobilizing grasps provide such immobilizations of parts. We explain about form-closure grasps and second-order immobilizing grasps in Subsections 1.2.1 and 1.2.2. Although equilibrium grasps are not suitable for practical manipulation, equilibrium is a necessary condition for form-closure and second-order immobilization. In Subsection 1.2.3 we explain about equilibrium grasps and we also explain how caging relates to equilibrium grasps.

1.2.1 Form closure

Form closure, introduced by Reuleaux more than a century ago, is a way of kinematically defining the notion of "firm grip" of the object without relying on friction [Markenscoff et al., 1990]. A rigid body grasped by some fingers is said to be in form closure if they constrain all *finite* and *infinitesimal* motions of the body. Figure 1.7 illustrates a form-closure grasp formed by four fingers. When friction is taken into account such grasps are often called force-closure grasps. With frictional fingers, fewer figners are required to achieve force closure compared with form closure. We do not consider force-closure grasps in the thesis.

First by Somoff [1897] and then later by Lakshminarayana [1978] it was proven that at least seven frictionless contacts are necessary to achieve form closure of three-dimensional objects. Markenscoff et al. [1990] proved that four point fingers are suffi-

cient and often necessary to put a planar object in form closure. They also showed that seven fingers are necessary and sufficient to put a three-dimensional object in form closure. There exist objects that cannot be immobilized with frictionless fingers at all, such as circles, cylinders, and screws.

Mishra et al. [Mishra et al., 1987] provided an algorithm to construct a form-closure grasp of a planar object with four to six fingers, and a form-closure grasp of a three dimensional object with seven to twelve fingers. Markenscoff et al. [Markenscoff et al., 1990] considered the maximum inscribed circle to construct a form-closure grasp of a planar object with four fingers. Van der Stappen et al. [van der Stappen et al., 2000] proposed the first algorithm to report all edge quadruples of a polygon each of which contains at least one form-closure grasp induced by four point-contacts. The running time of the algorithm is $O(n^{2+\epsilon} + K)$, where n is the number of edges of the polygon, ϵ is an arbitrary small positive value, and K is the number of reported quadruples. Gopalakrishnan and Goldberg [Gopalakrishnan and Goldberg, 2002] studied form-closure grasps induced by placing two point-contacts at two external or internal concave vertices. Meyer [Meyer, 1993] presented an algorithm to construct a form-closure grasp of a convex polyhedron with seven fingers. The running time of the algorithm is $O(n^{3/2}\sqrt{\log n})$, where n is the number of faces of the polyhedron. Ding et al. [Ding et al., 2000] proposed an algorithm to construct a form closure for a three-dimensional object, and later the results were generalized to three-dimensional curved objects [Ding et al., 2002]. Cheong et al. presented output-sensitive algorithms[2] to compute all form-closure grasps of a simple polygon with few fingers [Cheong et al., 2003, 2006]. Cheong and van der Stappen have presented output-sensitive algorithms to compute all form-closure grasps of a planar semi-algebraic object with at most four fingers [Cheong and van der Stappen, 2005]. In her thesis, Cheong presented an efficient algorithm to compute all form-closure grasps of a rectilinear polyhedron with at most seven fingers [Cheong, 2006].

1.2.2 Second-order immobilizing grasps

Rimon and Burdick [Rimon and Burdick, 1998] showed that there are equilibrium grasps that are not form-closure grasps, but nevertheless completely immobilize the object by preventing any *finite* motion of the object through curvature effects in configuration space (forming an immobilizing grasp). These immobilizing grasps are referred to as second-order immobilizing grasps, and form-closure grasps are referred to as first-order immobilizing grasps. Figure 1.8 displays a second-order immobilizing grasp of a polygon formed by three fingers.

Czyzowicz et al. [Czyzowicz et al., 1999] provided a necessary and sufficient geometric condition for a simple polygon to be immobilized by three frictionless point contacts. Rimon and Burdick showed that three sufficiently-flat fingers can immobilize generic piecewise-smooth planar objects [Rimon and Burdick, 1995]. Rimon extended these results to three dimensional objects by showing that four sufficiently-flat fingers

[2] An output-sensitive algorithm is an algorithm whose running time depends not only on the size of the input but also on the size of the output.

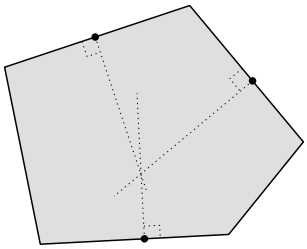


Figure 1.8: A second-order immobilizing grasp of a polygon formed by three point contacts.

can immobilize generic piecewise-smooth three-dimensional objects [Rimon, 2001].

Czyzowicz et al. [Czyzowicz et al., 1999] gave an algorithm that reports a second-order immobilizing grasp of a polygon with three fingers in $O(n \log n)$ time. The algorithm takes $O(n)$ time for convex polygons. Ponce et al.'s solution reports all immobilizing grasps of a part bounded by polynomial splines [Ponce et al., 1995]. Cheong et al. have proposed output-sensitive algorithms to report all second-order immobilizing grasps with two and three fingers [Cheong et al., 2006].

1.2.3 Equilibrium grasps

An equilibrium grasp is a grasp whose grasping fingers can exert forces and moments (not all of them zero) through the grasping points to balance the object [Mason, 2001]. There are a number of results on analyzing the equilibrium of forces in a grasp with different types of contact [Salisbury, 1982], with flexible contacts [Cutkosky, 1984], with friction [Abel et al., 1985], for three-dimensional and four fingers [Ponce et al., 1997], or for polygonal objects with three fingers [Li et al., 2003]. There are also a number of works on finding a good grasp based on some goal function such as optimum for internal forces [Kerr, 1985], or security of grasp [Jameson, 1985].

Equilibrium grasps are usually analysed to provide ways to synthesize form-closure, force-closure, or second-order immobilizing grasps, We mention the work done by Nguyen [Nguyen, 1986], by Ponce et al. on three-dimensional polyhedral objects with four fingers [Ponce et al., 1997], and by Li et al. on planar objects with three fingers [Li et al., 2003].

Interestingly, caging arrangements and equilibrium grasps are shown to be related by Rimon and Blake [Rimon and Blake, 1996]. They showed that in a multi-finger one-parameter gripping system, the hand's configuration at which the cage is broken corresponds to an equilibrium grasp. That means that as we start from a caging arrangement and continuously change the parameter (either increasing or decreasing the single parameter) at certain value of the parameter the corresponding hand's configuration becomes non-caging. That specific hand's configuration, at which the caging property changes from caging to non-caging, corresponds to an equilibrium grasp.

1.3 Caging

In this section we elaborate on the history of caging and its applications. We review the related work done on computing two- and three-finger caging arrangements.

The earliest work on trapping objects was done by Besicovitch [Besicovitch, 1957] and Shephard [Shephard, 1965]. Besicovitch showed that if a net of inextensible string encloses a sphere of unit radius in such a way that the sphere cannot slip out, then the length of the string is strictly greater than 3π, and this is false with any greater constant replacing 3π. Shephard and Besicovitch both worked on the problem of trapping a sphere with a crate [Shephard, 1965]. A crate is a set of edges of a convex polyhedron in three-dimensional Euclidean space. They worked on constructing a crate with minimum possible length holding a sphere of unit radius that does not allow the sphere to slide out.

The caging problem for planar objects was first formally defined by Kuperberg [Kuperberg, 1990] as a problem of finding a set of placements of fingers that prevents a polygon from moving arbitrarily far from its given position.

> "Let P be a polygon in the plane, and let C be a set of k points which lies in the complement of the interior of P. The points capture P if P cannot be moved arbitrarily far from its original position without at least one point of C penetrating the interior of P. Design an algorithm for finding a set of capturing points for P."

Rimon and Blake [Rimon and Blake, 1996] introduced the notion of the *caging set* as the maximal connected set of caging configurations that contains a given immobilizing grasp configuration. They have shown that for a one-parameter gripper the limit configurations where the caging property changes from caging to non-caging correspond to equilibrium grasps of the object. They have computed the caging set for a two-finger one-parameter gripper of planar objects. This result was later improved by Davidson and Blake as they proposed a more efficient algorithm to compute the caging set [Davidson and Blake, 1998a]. In a separate work, they extended the result to a three-finger one-parameter gripper [Davidson and Blake, 1998b].

There are number of works that have applied caging to a number of problems in manipulation. Sudsang and Ponce applied the caging concept to motion planning for three disk-shaped planar robots manipulating an object [Sudsang and Ponce, 2000, Sudsang et al., 1999]. They proposed a geometrical solution to compute a conservative approximation of the so called *inescapable configuration space* regions. Diankov et al. have also recently applied caging arrangements to manipulation planning such as opening doors and drawers by robots such as humanoid robots or mobile manipulators [Diankov et al., 2008]. Figures 1.1 and 1.2 display two robot arms, one opening a door and the other one opening a cupboard[3].

Sudsang et al. have applied caging to grasping and in-hand manipulation [Sudsang et al., 1998, Sudsang, 2000]. Pereira et al. applied caging to decentralized multi-robot

[3]Both pictures are taken from a paper by Diankov et al. [Diankov et al., 2008].

manipulation [Pereira et al., 2004]. They proposed a conservative, online, and decentralized test to maintain caging, by using the geometrical description of the robots. Other works on decentralized multi-robot manipulation include the work done by Fink et al. [2007, 2008].

1.3.1 Caging with two fingers

Rimon and Blake [Rimon and Blake, 1996] considered the problem of determining the caging set K, of all hand configurations which maintain a given object (not necessarily polygonal) caged such that from any initial hand configuration in K there exists a continuous path in K which leads to a given desired immobilizing grasp. In this thesis, however, we are interested in *all* possible two-finger caging arrangements of simple polygonal objects. (Figure 1.10 illustrates two examples of two-finger caging arrangements.) This problem was first tackled by Sudsang and Luewirawong [Sudsang and Luewirawong, 2003]. Their idea is to consider the immobilizing grasps at pairs of two concave vertices, or at a concave vertex and an edge. Taking into account the incident edges only, a local distance was computed for every immobilizing grasp that kept the fingers caging (neglecting the rest of the body of the polygon). As a result, the algorithm is incomplete as it reports only a subset of all caging arrangements of two disk fingers. Pipattanasomporn and Sudsang [Pipattanasomporn and Sudsang, 2006], independently from our paper, have recently solved the problem for two point fingers in $O(n^2 \log n)$ time, and also constructed a data structure capable of answering queries in $O(\log n)$ time. They suspected that every caging arrangement is a so-called squeezing caging arrangement or a so-called stretching caging arrangement without proving this non-obvious claim. (Squeezing and stretching notions are defined in Section 1.4.) Moreover, generalizing the results to disk fingers was not discussed in their paper. There is a recent work done by Pipattanasomporn et al. [Pipattanasomporn et al., 2007] on computing only the set of all squeezing caging arrangement of polygons and polyhedra with two point fingers using a convex decomposition technique. However, these results are neither easily or straightforwardly extendable from point fingers to disk fingers, nor from squeezing caging arrangements to stretching caging arrangements.

Clearly convex polygons and certain non-convex polygons cannot be caged with two fingers. Figure 1.5 illustrates a non-convex polygon that cannot be caged with two fingers. Therefore, it is important to also consider caging arrangements that involve more than two fingers.

1.3.2 Caging with three fingers

In the problem of caging a polygon with three fingers, the placements of two fingers, to which we refer as the base fingers, are given. It is required to find all placements of the third finger, such that the resulting three-finger arrangements cage the polygon. Such placements of the third finger form a number of connected components in the plane, to which we will jointly refer as the *caging set*. Figure 1.9 shows two exam-

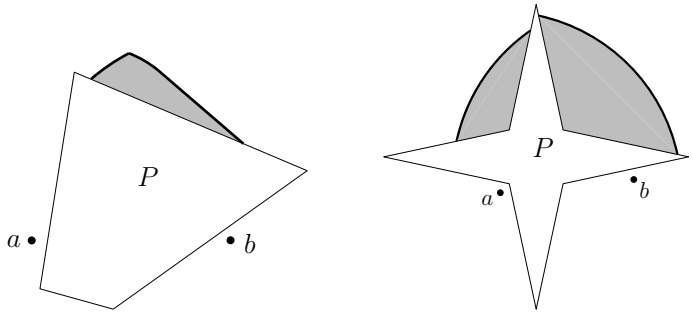


Figure 1.9: Two examples of caging sets of a convex and a non-convex polygon.

ples of caging sets of a convex polygon and a non-convex polygon. Before Erickson et al.'s work [Erickson et al., 2007] the previous complete algorithms had been limited to robotic systems with a single degree of freedom [Davidson and Blake, 1998b, Rimon and Blake, 1996] whereas efforts to tackle robotic systems with multiple degrees of freedom had been limited to approximate algorithms that assume each finger can only interact with a single object edge [Sudsang, 2000, Sudsang and Ponce, 2000, Sudsang et al., 1999]. Given the placements of the base fingers which are required to be on the boundary of a convex polygon, Erickson et al. [Erickson et al., 2007] provided the first complete algorithm for computing the caging set for such polygons in $O(n^6)$ time. However, the problem of computing the caging set when the base fingers are not necessarily on the boundary of the polygon, or when the polygon is not convex, remained open.

1.4 Contributions and organization

In Chapter 3 we present an algorithm that computes all caging arrangements of two disk fingers in $O(n^2 \log n)$ time. In addition a data structure is constructed that requires $O(n^2)$ space and is capable of answering in $O(\log n)$ time whether a given two disk-finger arrangement is caging. Our work on two-finger caging extends and improves the result by Pipattanasomporn and Sudsang [Pipattanasomporn and Sudsang, 2006] by providing an algorithm for computing all caging arrangements of two disk fingers by taking the geometry of the entire polygon into account. Therefore, our algorithm computes the complete caging set. To do this we have proven that every caging placement is squeezing or stretching caging, and thus established a relation between two-finger caging arrangements and immobilizing grasps of polygons. A caging arrangement is a squeezing caging arrangement when the fingers cannot be moved to a placement in which the fingers coincide while keeping the distance between the fingers at most equal to the distance between the two finger placements of the given caging arrangement. A caging arrangement is a stretching caging arrangement when the fingers cannot be moved to a placement in which the fingers are far from each other with respect to the polygon while keeping the distance between the fingers at least

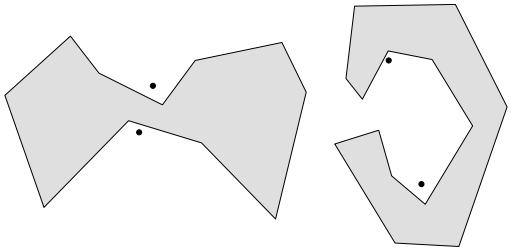


Figure 1.10: Left picture displays a two-finger squeezing caging arrangement. Right picture displays a two-finger stretching caging arrangement.

equal to the distance between the fingers of the given caging arrangement. Figure 1.10 illustrates examples of a two-finger squeezing caging arrangement and a two-finger stretching caging arrangement.

Relevent publications of Chapter 3 are as follows:

- M. Vahedi and A. F. van der Stappen. Caging polygons with two and three fingers. In *Seventh Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 71–86, 2006.
- M. Vahedi and A. F. van der Stappen. Caging polygons with two and three fingers. *Int. J. Robotics Res.*, 27(11/12):1308–1324, 2008.

The combinatorial complexity of the set of all two-finger caging arrangements can vary greatly. On the one hand, there are non-convex polygons that cannot be caged with two fingers and therefore the set is empty, while on the other hand there exist polygons for which the complexity of the set of caging arrangements is $\Omega(n^2)$. In Chapter 4 we take a first-step towards output-sensitive caging arrangement computation by proposing an algorithm for two disk fingers that runs in time $O(n^{4/3}\log^{2+\epsilon} n + M\log n + c\log(c+M))$, in which M is the number of so-called minimum or maximum grasps, c is the complexity of the set of reported caging arrangements, and ϵ is an arbitrarily small positive constant. If the set of caging arrangements turns out to be empty, our algorithm will report so in $O(n^{4/3}\log^{2+\epsilon} n)$ time. Compared to existing algorithms, which all have a quadratic dependency on the complexity of the part, we have traded considerable sensitivity to the size of the input for desirable output-sensitivity. Moreover, the output of the algorithm can also be efficiently queried to check whether a given *arbitrary* two-finger arrangement is caging in $O(\log n)$ time.

Relevent publication of Chapter 4 is as follows:

- M. Vahedi and A. F. van der Stappen. Towards output-sensitive computation of two-finger caging grasps. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 73–78, 2008.

In Chapter 5 we present a solution for computing the caging set for convex and non-convex polygons for a given placements of the base fingers. It is shown that the

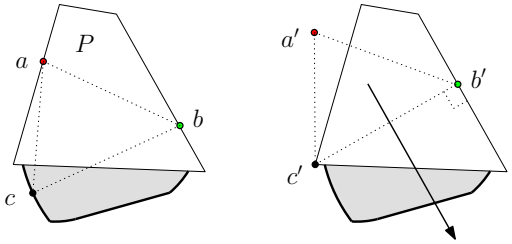


Figure 1.11: Left picture displays a three-finger arrangement in which the third finger is on the boundary of the caging set. Right picture displays the corresponding equilibrium grasp.

sections of the boundary of the caging set (the caging set is a set of two-dimensional regions) that do not belong to the polygon boundary correspond to equilibrium grasps. Figure 1.11 displays a three-finger arrangement in which the third finger is on the boundary of the caging set; the caging set is displayed with gray color; the bold curves display the boundary of the caging set that do not belong to the polygon boundary; the corresponding equilibrium grasp, which is a two-finger equilibrium grasp, is displayed at right. Our work on three-finger caging uses this fact to extend the results by Erickson et al. [Erickson et al., 2007] from convex polygons to non-convex polygons, and also from the placement of the base fingers on the polygon boundary to arbitrary placements. The running time of our proposed three point-finger caging algorithm is $O(n^6 \log^2 n)$.

Relevent publications of Chapter 5 are as follows:

- M. Vahedi and A. F. van der Stappen. Caging polygons with two and three fingers. In *Seventh Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 71–86, 2006.

- M. Vahedi and A. F. van der Stappen. Caging polygons with two and three fingers. *Int. J. Robotics Res.*, 27(11/12):1308–1324, 2008.

In the next four chapters, 6, 7, 8, and 9 we focus on caging convex polygons with three fingers. In Chapter 6 we introduce some definitions and assumptions used in next three chapters and also provide some geometric facts about caging convex polygons with three fingers.

Relevent publication of Chapter 6 is as follows:

- M. Vahedi and A. F. van der Stappen. Geometric properties and computation of three-finger caging grasps of convex polygons. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 404–411, 2007.

In Chapter 7 we prove that the worst-case complexity of the caging set of convex polygons is $O(\lambda_{10}(n^3))$, which significantly improves the already known upper bound

of $O(n^6)$, as $\lambda_{10}(n^3)$ is known to be $O(n^3 \log^* n)$ [Sharir and Agarwal, 1996][4] and thus very close to $O(n^3)$. Erickson et al. [Erickson et al., 2007] showed that the caging set is the visible scene of $O(n^3)$ constant-complexity surfaces in a three-dimensional space. To establish the upper bound, firstly we have narrowed down the types of surfaces introduced by Erickson et al. that play a role in caging set complexity. Secondly, we have formulated a new way to compute the caging set using those surfaces. In addition, we develop an efficient algorithm to compute the caging set in $O(\lambda_{10}(n^3) \log n)$ time using a divide-and-conquer technique.

Relevent publication of Chapter 7 is as follows:

- M. Vahedi and A. F. van der Stappen. On the complexity of the set of three-finger caging grasps of convex polygons. In *Robotics: Sience and Systems Conference*, 2009.

In Chapter 8 we assume that the convex polygon and the distance between the base fingers are given. We compute a data structure in $O(n^7)$ time. The data structure can be queried to report the caging set for *any* given placement of the base fingers more efficiently. When the base fingers are on the boundary of the polygon the query time turns out to be largely proportional to the combinatorial complexity of the reported caging set. Moreover using the same idea we consider another query; for a given placement of the base fingers and a placement of the third finger, we give an algorithm that determines in $O(\log n)$ time whether the resulting arrangement cages the polygon.

Relevent publication of Chapter 8 is as follows:

- M. Vahedi and A. F. van der Stappen. Geometric properties and computation of three-finger caging grasps of convex polygons. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 404–411, 2007.

In Chapter 9 we abandon the restriction of the distance between the base fingers by generalizing the data structure of the Chapter 8 so that it can be queried for *arbitrary* distances between the base fingers.

Relevent publication of Chapter 9 is as follows:

- M. Vahedi and A. F. van der Stappen. Caging convex polygons with three fingers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1777–1783, 2008.

[4] $\log^* n = \min\{i \geq 0 : \log^{(i)} n \leq 1\}$, where i is a nonnegative integer, and $\log^{(i)} n$ is the logarithm function applied i times in succession, starting with argument n.

Chapter 2

Grasp Analysis and Preliminaries

In this chapter we introduce the notions of form-closure, second-order immobility, and equilibrium grasps. We also present a number of algorithms and data structures used in this thesis. The chapter is minimalist in the sense that we only explain the facts and concepts that we use in the thesis in detail.

2.1 Notation

The configuration space for a jointed robot $\mathcal{R}$ moving among obstacles, is the set of parametric representations of all possible placements of $\mathcal{R}$. A point p in this configuration space corresponds to a certain placement $\mathcal{R}(p)$ of the robot in work space. The set of points in configuration space corresponding to placements where $\mathcal{R}$ intersects the obstacles is called forbidden configuration space, or forbidden space for short. The rest of the configuration space, which consists of the points corresponding to free placements—i.e. placements where $\mathcal{R}$ does not intersect the obstacles—is called the free configuration space, or free space for short.

In this thesis, we study the so called *caging arrangements* of two-dimensional polygonal objects with two and three fingers. The robot $\mathcal{R}$ is a robot hand that consists of a number of disk-shaped fingers, but, we do not consider or formulate the links between the fingers. Moreover, there is only one obstacle, which is a rigid polygonal object. In a caging arrangement, the fingers of $\mathcal{R}$ should be arranged around the polygonal object in a way such that they do not allow the polygonal object to escape through the fingers by rotation and/or translation, thus form a cage around the object.

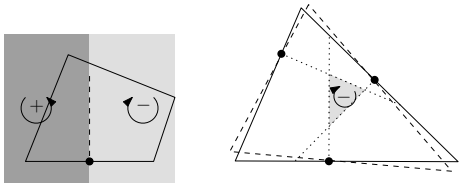


Figure 2.1: (a) When a point finger is in the interior of an edge. (b) The possible centers of rotations for a number of point fingers all in the interior of edges.

2.1.1 Form closure and immobilization

When fingers *immobilize* an object, any rigid motion of the object (rotation and translation) causes at least one finger to penetrate the interior of the object. Reuleaux [1876] formulated the notion of *form closure* for a body which is a sufficient condition for immobilization with frictionless fingers. He provided a geometric analysis for form closure of a planar object which is described (partially) in this section. Form closure of two-dimensional and three-dimensional objects can be also analyzed in so called *wrench space* that encodes forces and moments. Rimon and Burdick [1998] generalized the idea of incorporating curvature into the theory of *second-order immobility*. This theory is based on an analysis in configuration space of the part when the fingers are regarded as obstacles. According to this analysis an object is immobilized if and only if the current placement of the object is an isolated point in the configuration space. (In their terms form closure is *first-order immobility* that uses only the directions of motion of the object with respect to the fingers in the analysis, but not curvature.) Czyzowicz et al. [1999] independently provided a necessary and sufficient condition for a restricted case in which a simple polygons is immobilized with three frictionless point fingers by taking the relative curvature of the polygon at the fingers into account. We will use and therefore summarize some results from their paper.

Given a collection of point fingers on the boundary of an object, Reuleaux provided a conservative analysis of the object's possible motions; the analysis can be applied only to planar objects. Every *infinitesimal* motion can be seen as a rotation around a point in the projective plane in either counterclockwise or clockwise direction. (Translations are thus seen as rotations about points at infinity.) Consider Figure 2.1.a. When a point finger is in the interior of a straight edge, the locally inward directed line orthogonal to the edge at the finger divides possible centers of counterclockwise and clockwise rotations. The half-plane left of this line has possible centers of counterclockwise rotations and the half-plane right of this line has those of clockwise rotations. Reuleaux' analysis conservatively assumes that both clockwise and counterclockwise rotations are possible about points on the orthogonal line. Therefore given a number of point fingers on the boundary of an object, it can be rotated around any point inside the intersection of the oriented half planes induced by the point fingers (Figure 2.1.b). When the intersections of the oriented half planes are empty, the object is said to be in form closure.

Czyzowicz et al. [1999] have refined Reuleaux's analysis for the case of polygons. They have derived necessary and sufficient conditions for immobilization with three point fingers. Based on their work, the following theorem states the necessary and sufficient conditions of an immobilizing grasp on P.

Theorem 2.1.1. *Let a, b, and c be points on the boundary of P. Point contacts at a, b and c along the different incident edges γ_a, γ_b and γ_c of P respectively immobilize P if and only if:*

- *the extensions of γ_a, γ_b, and γ_c enclose P, and*
- *the orthogonals to γ_a, γ_b, and γ_c at the points a, b, and c respectively meet at a common point.*

Lemma 2.1.2 constrains the motion of P when the orthogonals at the fingers do not meet at a point.

Lemma 2.1.2. *If three fingers are placed along the interiors of edges whose extensions enclose P, then P cannot translate.*

Proof. Since the extensions of the edges enclose P, Reuleaux's analysis shows that all possible centers of rotations form a bounded triangle. Therefore no possible center of rotation is at infinity. Hence it it is not possible to translate P. □

Note that the enclosing condition of the edges is just one of the conditions that should be satisfied so that the grasp be immobilizing by Theorem 2.1.1. A polygon is immobilized when it is not possible to rotate and translate the polygon. To rule out rotations the orthogonals at the fingers must also meet at a common point.

2.1.2 Caging and equilibrium grasps

The *wrench* w induced by a finger of a point p along the boundary of the planar polygon P is defined by $w = (n, p_x n_y - p_y n_x) \in \mathbb{R}^3$, where n is a normalized inward-directed normal at p. The notion of *positive spanning* is important to analyze equilibrium grasps [Mason, 2001]. The positive linear span $\text{pos}(\{v_i\})$ of a set of vectors $\{v_i\}$ is defined as:

$$\text{pos}(\{v_i\}) = \{\sum k_i v_i | k_i \geq 0\}.$$

A set of vectors $\{v_i\}$ positively spans the entire space $\mathbb{R}^n$ if and only if $\text{pos}(\{v_i\}) = \mathbb{R}^n$. The set $\text{pos}(\{v_i\})$ should at least contain $n+1$ vectors to positively span $\mathbb{R}^n$. The convex hull $\text{conv}(\{v_i\})$ of a set of vectors $\{v_i\}$ is defined as:

$$\text{conv}(\{v_i\}) = \{\sum k_i v_i | k_i \geq 0, \Sigma k_i = 1\}.$$

The set $\text{pos}(\{v_i\})$ positively spans $\mathbb{R}^n$ if and only if the origin is in the interior of $\text{conv}(\{v_i\})$.

A three-finger grasp inducing wrenches w_1 induced by a finger at p_1 with a line of action n_1, w_2 induced by a finger at p_2 with a line of action n_2, and w_3 induced by a finger at p_3 with a line of action n_3 is an equilibrium grasp if and only if

$$k_1 w_1 + k_2 w_2 + k_3 w_3 = 0$$

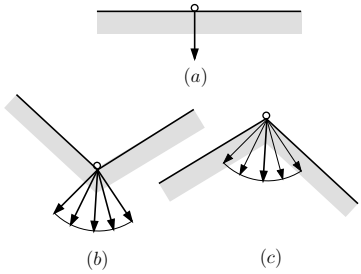


Figure 2.2: Wrenches at point contacts.

with $k_1, k_2, k_3 \geq 0$, admits a non-trivial solution, i.e., a solution in which not $k_1 = k_2 = k_3 = 0$.

Clearly there are two possibilities:

1. exactly one of k_1, k_2, k_3 equals zero, or

2. k_1, k_2, k_3 are all nonzero.

Case 1, assuming without loss of generality that $k_3 = 0$, implies that $k_1 w_1 = -k_2 w_2$, or $w_1 = -k w_2$ for some $k > 0$, which directly implies that the lines of action at p_1 and p_2 coincide but are oppositely directed. These are two-finger equilibrium grasps with all three fingers on the boundary of the polygon. Clearly, there are also two-finger equilibrium grasps such that one of the three fingers is *not* on the boundary of the polygon.

Case 2, resembles positive spanning because all of k_1, k_2, k_3 must be positive. The line of actions n_1, n_2, n_3 must positively span $\mathbb{R}^2$ and p_1, p_2, p_3 are such that the lines of action have a common point. These are the three-finger equilibrium grasps.

Our model to compute the wrenches on a given polygon differs from Markenscoff et al. [1990] model in case of a point contact and a convex vertex. See Figure 2.2 in which the small circles are the places of the point contacts and the gray sections indicate the interior of a polygon. Let b be a point on the boundary of a given polygon P. If b is on a line segment, then the wrench is the singleton set containing the wrench generated by the unit vector normal to the tangent, and directed toward the interior of P (case a). If b is on a concave vertex, then there are wrenches generated by the two inward-directed normals applied on the adjacent edges of the vertex. The wrenches for this point, is the set of all convex combinations of these two wrenches (case b). In our model, if b is on a convex vertex, then the *possible* wrenches for this point, similar to the concave case, contains the set of all convex combinations of the two wrenches (case c, notice the bold vectors in cases b and c). But the wrench for this point at one time, is just one member of that set. The intuition behind this model, is that an ϵ-radius disk-shaped finger on a vertex finger can apply any wrench, being a combination of the two normal wrenches with a small adjustment. See Figure 2.3 which displays a small disk-shaped finger at a convex vertex in two cases and the wrench for each case

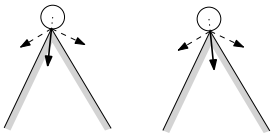


Figure 2.3: This picture displays a small disk-shaped finger at a convex vertex in two cases and the wrench for each case is displayed with a bold arrow.

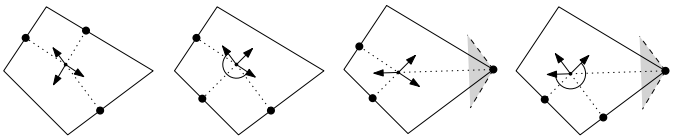


Figure 2.4: From left to right the first and the third grasps are equilibrium grasps. The second and the fourth grasps do not satisfy the angle condition.

is displayed with a bold arrow. Theorem 2.1.3 below summarizes both two- and three-finger equilibrium grasps into a single condition.

Theorem 2.1.3. *A two- or three-finger grasp along the boundary of a two-dimensional object is an equilibrium grasp if and only if the lines of action meet at a common point and the angle between every two consecutive lines of action is at most* π.

The Theorem 2.1.3 states a condition on the angle between every two consecutive lines of action which is referred to as the *angle condition*. Figure 2.4 shows some examples of both equilibrium and non-equilibrium grasps.

Using stratified Morse theory, Rimon and Blake [1996, Proposition 3.3] showed that in a multi-finger one-parameter gripping system, the hand's configuration at which the cage is broken corresponds to an equilibrium grasp. That means that as we start from a caging arrangement and continuously change the parameter (either increasing or decreasing the single parameter) at certain value of the parameter the corresponding hand's configuration becomes non-caging. That specific hand's configuration, at which the caging property changes from caging to non-caging, corresponds to an equilibrium grasp. Throughout the thesis we use the term *cage is broken* to denote that the caging property changes from caging to non-caging.

2.2 Geometric tools

In this section we define a number of algorithms and data structures we have used in this thesis.

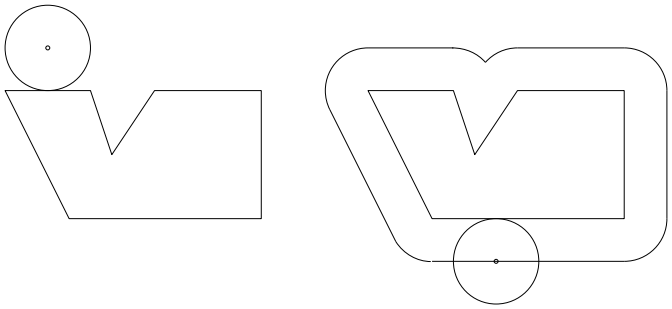


Figure 2.5: The Minkowski sum of the disk and polygon displayed in the left is shown in the right.

2.2.1 Minkowski sum

The Minkowski sum of two sets $S_1 \subset \mathbb{R}^2$ and $S_2 \subset \mathbb{R}^2$, denoted by $S_1 \oplus S_2$, is defined as

$$S_1 \oplus S_2 = \{p + q : p \in S_1, q \in S_2\},$$

where $p + q$ denotes the vector sum of the vectors p and q [de Berg et al., 2008]. Consider Figure 2.5 in which the Minkowski sum of a disk with a polygon is displayed. Placing a disk finger D_r of radius r around P is equivalent to placing point fingers around the generalized polygon $P \oplus D_r$. A generalized polygon is a shape bounded by straight segments and circular arcs. The asymptotic complexity of $P \oplus D_r$ equals that of P and it can be computed in linear time with respect to the size of P [de Berg et al., 2008]. In this thesis we propose an algorithm in Chapter 3 that computes all two point-finger caging arrangements for a generalized polygon. By using the Minkowski sum, the same algorithm can be used to solve the same problem for a simple polygon and disk fingers.

2.2.2 Arrangements

Given a finite collection $\mathcal{S}$ of geometric objects such as hyperplanes or spheres in $\mathbb{R}^d$ the arrangement $\mathcal{A}(\mathcal{S})$ is the decomposition of $\mathbb{R}^d$ into connected open cells of dimensions $0, 1, \ldots, d$ induced by $\mathcal{S}$ [Halperin, 2004].

Theorem 2.2.1. *[Halperin, 2004] Given a collection $\mathcal{S}$ of n surfaces in $\mathbb{R}^d$, as defined above, the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{S})$ is $O(n^d)$. There exists arrangements of $\theta(n^d)$ complexity.*

2.2.3 Vertical decomposition

A raw arrangement may still be an inappropriate structure as cells may have complicated shapes, and many neighboring subcells. It is often desirable to decompose the cells of the arrangement into subcomponents so that each subcomponent has a

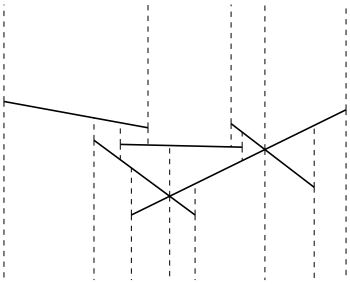


Figure 2.6: Trapezoidal decomposition of a set of line segments is displayed.

constant descriptive complexity. One such decomposition is vertical decomposition or trapezoidal decomposition. Figure 2.6 shows a trapezoidal decomposition of an arrangement induced by a set of line segments. In a trapezoidal decomposition of a set of two-dimensional arcs a vertical line segment is extended upward and downward from each vertex of the arrangement and also from each vertical tangency point of the arcs until it either hits another segment or extends to infinity. The complexity of a vertical decomposition for two-dimensional arrangements is asymptotically the same as that of the underlying arrangement. Theorem 2.2.2 bounds the complexity in three-dimensional space.

Theorem 2.2.2. *[Agarwal et al., 1996] Given a collection $\mathcal{S}$ of n algebraic surfaces in $\mathcal{R}^3$, then for any $\epsilon > 0$, a vertical decomposition of size $O(n^{3+\epsilon})$ for the arrangement $\mathcal{A}(\mathcal{S})$ can be constructed in $O(n^{3+\epsilon})$ time, so that a point-location query can be answered in $O(\log n)$ time.*

2.2.4 Point location

Given a set of disjoint geometric objects (planar regions or three-dimensional objects), the point-location problem asks for the object containing a query point. For three-dimensional spaces the theorem 2.2.2 is mentioned in Subsection 2.2.3 that provides us with a data structure to answer point-location queries. For two-dimensional spaces the set of planar regions are decomposed into simpler cells (pseudo-trapezoids) of constant complexity by vertical decomposition, and a data structure (a directed acyclic graph) is constructed on top of these cells, allowing to locate the pseudo-trapezoid containing a query point in expected logarithmic time. The search-structure and the vertical decomposition can be computed in $O(N \log N)$ expected time where N is the complexity of the vertical decomposition [Mulmuley, 1990, de Berg et al., 2008].

2.2.5 Triangulation

A triangulation of a polygon is a decomposition of polygon into triangles by a set of non-intersecting diagonals. A diagonal of a polygon is a line segment that connects two vertices of the polygon and lies inside the polygon. Triangulations are desirable

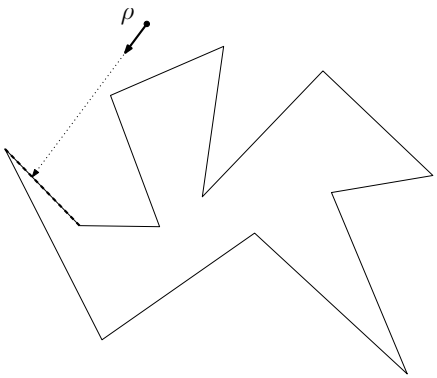


Figure 2.7: The first edge of the given polygon intersected by the ray ρ is displayed with a bold dashed line segment.

decompositions because the complexity of each triangle is constant and also because the total number of triangles is linear in the complexity of the triangulated polygon. Triangulations are usually not unique and a polygon can be triangulated in several different ways. The total number of triangles in a triangulation is $n-2$ in which n is the number of vertices of the triangulated polygon.

Theorem 2.2.3. *[de Berg et al., 2008] A simple polygon with n vertices can be triangulated in $O(n \log n)$ time with an algorithm that uses $O(n)$ storage.*

Chazelle [1991] presented a more efficient algorithm that computes a triangulation of a simple polygon in linear time. However, the above mentioned simple algorithm is sufficient for our purposes. We use triangulation in Chapter 5 to eventually decompose the set of all non-intersecting three point-finger arrangements of a simple polygon into a set of constant complexity cells.

2.2.6 Ray shooting

Consider a collection $\mathcal{S}$ of geometric objects. Consider a query ray ρ—a ray is a half line emanating from a point in a direction. It is required to find the first object in $\mathcal{S}$ intersected by ρ. In Figure 2.7, $\mathcal{S}$ is the set of edges of the displayed polygon. The first edge intersected by the ray ρ is displayed with a bold dashed line segment. In two-dimensional space, the required data structure for a given polygon for the ray shooting queries can be computed in $O(n)$ time, and every ray shooting query can be answered in $O(\log n)$ time [Hershberger and Suri, 1995]. We use ray shooting data structure in Chapter 4 to find out where the fingers hit the polygon for the first time when we squeeze or stretch a two-finger arrangement along the line connecting the two finger-placements.

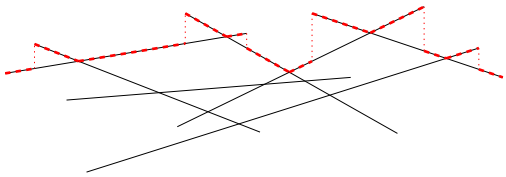


Figure 2.8: Upper envelope of a set of line segments is displayed with bold dashed line segments.

2.2.7 Upper envelope

Given a set of planar curves, which can be regarded to consist of a number of x-monotone pieces, which in turn can be seen as functions, their upper envelope (lower envelope) is the point-wise maximum (minimum) of this set of functions. See Figure 2.8 in which the upper envelope of a set of line segments is displayed. To analyze the complexity of upper envelopes we have to introduce the Davenport-Schinzel sequences.

A Davenport-Schinzel sequence, $DS(m,s)$-sequence, is a sequence of m symbols in which no two symbols alternate more than s times and the same symbol does not occur twice in a row. For instance, the sequence

$$1,2,1,3,1,3,2,4,5,4,5,2,3$$

is a $DS(5,3)$-sequence: it contains alternating subsequences of length four, such as

$$\ldots 1,\ldots 2,\ldots 1,\ldots 2,\ldots$$

(which appears in four different ways as a subsequence of the whole sequence) but it does not contain any alternating subsequences of length five.

The upper envelope (or lower envelope) of m two-dimensional x-monotone curve segments in which no two curve segments intersect each other more than $s-2$ times is a $DS(m,s)$-sequence. The maximum length of a $DS(m,s)$-sequence is $\lambda_s(m)$ [Sharir and Agarwal, 1996]. The function $\lambda_s(m)$ is known to be $O(m \log^* m)$ and thus very close to $O(m)$ for constant values of s. The function $\log^* n$ is defined as $\min\{i \geq 0 : \log^{(i)} n \leq 1\}$, where i is a nonnegative integer, and $\log^{(i)} n$ is the logarithm function applied i times in succession, starting with argument n.

The upper envelope (or lower envelope) of m two-dimensional x-monotone curve segments of maximum degree d ($d \geq 1$) is a $DS(m, 2d+2)$-sequence. The reason is that a pair of curve segments of maximum degree d intersect each other at most at $2d$ points. Therefore the maximum length of the upper envelope of m such curve segments is $\lambda_{2d+2}(m)$.

We use the complexity bounds of upper envelopes to prove an upper bound on the complexity of the set of three-finger caging arrangements of convex polygons in Chapter 7.

Chapter 3

Caging Polygons with Two Disk Fingers

In the problem of caging a polygon with two disk fingers, it is required to compute all arrangements of two disk fingers that cage a polygonal object with n edges in the plane. This problem was first tackled by Sudsang and Luewirawong [2003]. Their idea is to consider the immobilizing grasps at pairs of two concave vertices, or at a concave vertex and an edge. Taking into account the incident edges only, a local distance was computed for every immobilizing grasp that kept the fingers caging (neglecting the rest of the body of the polygon). As a result, the algorithm is incomplete as it reports only a subset of all caging arrangements of two disk fingers. Pipattanasomporn and Sudsang [2006] independently, have solved the problem for two fingers in $O(n^2 \log n)$ time, and also constructed a data structure capable of answering queries in $O(\log n)$ time. However their solution can be applied only to point fingers. Our work on two-finger caging in this chapter extends and improves their result by providing an algorithm for computing all caging arrangements of two disk fingers by taking the geometry of the entire polygon into account. Therefore, our algorithm computes the complete caging set. To do this we have proven that every caging arrangement is a squeezing or a stretching caging, and therefore established a relation between two-finger caging arrangements and immobilizing grasps of polygons without parallel edges. A caging arrangement is a squeezing caging arrangement when the fingers cannot be moved to a placement in which the fingers coincide while keeping the distance between the fingers at most equal to the distance between the two finger-placements of the given caging arrangement. When the fingers coincide outside the polygon, both fingers can be taken to any place, even to infinity, without colliding with the polygon. A caging arrangement is a stretching caging arrangement when the fingers cannot be moved to a placement in which the fingers are far from each other with respect to the polygon while keeping the distance between the fingers at least equal to the distance between the two finger placements of the given caging arrangement. When the fingers are far from each other with respect to the polygon, both fingers can be moved to any place,

even to infinity, without colliding with the polygon.

In Section 3.1 we introduce some notations and provide some definitions and explain some concepts we have used in this chapter. In Section 3.2 we give an overview of the algorithm that computes the set of all two-finger caging arrangements. We prove in Subsection 3.2.1 that every caging arrangement is a squeezing caging arrangement or a stretching caging arrangement (or both); using this fact we establish a relation between caging arrangements and immobilizing grasps in Subsection 3.2.2. In Section 3.3 we explain our algorithm that computes the set of all two-finger caging arrangements.

3.1 Preliminaries

In Subsection 3.1.1 we introduce some notations and in Subsection 3.1.2 we provide some definitions and explain some concepts we have used in this chapter.

3.1.1 Notations

This chapter addresses the problem of caging a polygon P with two disk fingers. Formally, P is caged with a number of fingers when its placement lies in a compact valid region of its free configuration space of P regarding the fingers as obstacles. Informally, P is caged with a number of fingers when the fingers make it impossible to take P to infinity without penetrating any finger. In general it is easier for the explanation to consider the polygon fixed and to move the fingers instead while keeping their mutual distances fixed. Therefore, P is caged when it is impossible to rigidly move the fingers to infinity without penetrating P.

The given simple polygon P is bounded by n edges. Let P_r denote the Minkowski sum of P and a disk of radius r centered at the origin. (Recall that the Minkowski sum of two sets A and B is the set $\{a + b | a \in A, b \in B\}$.) A disk of radius r intersects P if and only if its center lies in P_r. Placing disk fingers of radius r around P is equivalent to placing point fingers around the generalized polygon P_r. A generalized polygon is a shape bounded by straight segments and circular arcs. Define $F_r = \mathbb{R}^2 \setminus \text{int}(P_r)$. The set F_r is the set of all possible placements of a disk finger with radius r not intersecting P. Even if P itself does not contain holes, the set F_r may consist of more than one component of which exactly one is unbounded. The set F_r is closed; in particular, it includes the boundary of P_r.

A two-finger *arrangement* is a pair (a, b) of points in the plane, where each point is the center of a finger. In this chapter, the *admissible space* for two disk fingers with radii r and s is the set of all possible placements of the two fingers that do not intersect the polygon P. Formally $\mathcal{F} = F_r \times F_s$.

This section addresses the problem of caging a simple polygon P with two disk fingers. We assume that the fingers have radii r and s. Since in this chapter we only consider two-finger arrangements we generally omit the use of "two-finger".

3.1.2 Definitions

Let T_r be a vertical decomposition of F_r into a set of pseudo trapezoids. (See Subsection 2.2.3 for more information on vertical decompositions.) Here, a pseudo-trapezoid is a region bounded by two (possibly degenerate or unbounded) vertical segments, referred to as the left and right vertical walls, and two circular arcs or non-vertical segments belonging to the boundary of F_r. Similarly let T_s be a vertical decomposition of F_s into a set of pseudo trapezoids. Define $\mathcal{T} = T_r \times T_s$ which is a decomposition of the four-dimensional admissible space $\mathcal{F}$ into cells of constant complexity. A *δ-arrangement* is a two-finger arrangement for which the distance between the fingers is $\delta \in \mathbb{R}^+$. Let

$$\mathcal{F}_\delta = \{(p,q) \in \mathcal{F} | \; ||p-q|| = \delta\},$$

which is the set of all δ-arrangements. A *δ-min-arrangement* is a two-finger arrangement for which the distance between the fingers is at least $\delta \in \mathbb{R}^+$, and similarly a *δ-max-arrangement* is a two-finger arrangement for which the distance between the fingers is at most $\delta \in \mathbb{R}^+$. Let

$$\mathcal{F}_{\geq\delta} = \{(a,b) \in \mathcal{F} \mid ||a-b|| \geq \delta\},$$

which is the set of all δ-min-arrangements. Similarly, let

$$\mathcal{F}_{\leq\delta} = \{(a,b) \in \mathcal{F} \mid ||a-b|| \leq \delta\},$$

which is the set of all δ-max-arrangements. According to the definitions $\mathcal{F} = \mathcal{F}_{\geq\delta} \cup \mathcal{F}_{\leq\delta}$ and $\mathcal{F}_\delta = \mathcal{F}_{\geq\delta} \cap \mathcal{F}_{\leq\delta}$. A component of $\mathcal{F}_{\leq\delta}$ is *lower-bounded* if it contains no two-finger arrangement whose finger placements coincide. A two-finger arrangement is called a *δ-squeezing caging arrangement* if and only if it is a point of a lower-bounded component of $\mathcal{F}_{\leq\delta}$. A component of $\mathcal{F}_{\geq\delta}$ is *upper-bounded* if it contains no two-finger arrangement whose finger placements are far from each other with respect to P. A two-finger arrangement is called a *δ-stretching caging arrangement* if and only if it is a point of an upper-bounded component of $\mathcal{F}_{\geq\delta}$.

If a δ-arrangement is non-caging then there is a way to take the two fingers to a two-finer placement whose finger placements are far from P keeping the distance between the two fingers exactly equal to δ. If a δ-arrangement is not δ-squeezing caging then there is a way to take the two fingers to a two-finer placement whose finger placements are far from P keeping the distance between the two fingers at most equal to δ, and there freely squeeze the two fingers to coincide. When the two fingers coincide the distance between the two fingers is zero. Similarly, if a δ-arrangement is not δ-stretching caging then there is a way to take the two fingers to a two-finer placement whose finger placements are far from P keeping the distance between the two fingers at least equal to δ, and there freely stretch the two fingers so that both fingers get far from P and from each other. When the two fingers are far from P and from each other then the distance between the two fingers can be arbitrarily large.

A δ-squeezing caging arrangement is a caging arrangement because the containing component is lower-bounded, and thus it is not possible to take the two fingers to a coinciding two-finer placement even if we allow the distance between the fingers to

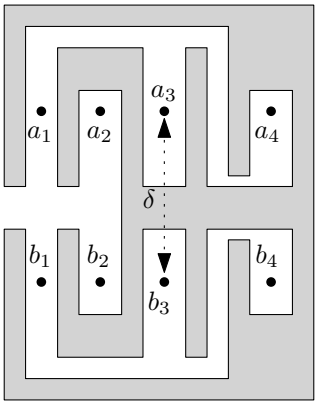


Figure 3.1: Reachability notions and caging types

be at most δ (i.e. the distance can be any value less than or equal to δ). Similarly a δ-stretching caging arrangement is a caging arrangement because the containing component is upper-bounded, and thus it is not possible to take the two fingers to a two-finer placement whose finger placements are far from P and from each other even if we allow the distance between the fingers to be at least δ (i.e. the distance can be any value more than or equal to δ).

Every δ-squeezing or δ-stretching caging arrangement is a caging arrangement. Our algorithm for two-finger caging is based on the surprising observation that the converse is also true—every δ-caging arrangement is δ-squeezing or δ-stretching.

In Figure 3.1 a shaded polygon and four δ-arrangements (a_1, b_1), (a_2, b_2), (a_3, b_3) and (a_4, b_4) are displayed. The two-finger arrangement (a_1, b_1) is not caging, (a_2, b_2) is δ-stretching caging, (a_3, b_3) is δ-squeezing caging, and (a_4, b_4) is both δ-stretching and δ-squeezing caging.

Two two-finger arrangements are *δ-reachable* if they are δ-arrangements and they lie in the same component of $\mathcal{F}_\delta$. Two two-finger arrangements are *δ-max-reachable* if they are δ-max-arrangements and they lie in the same component of $\mathcal{F}_{\leq\delta}$, and *δ-min-reachable* if they are δ-min-arrangements and they lie in the same component of $\mathcal{F}_{\geq\delta}$. When two two-finger arrangements are δ-reachable, it is possible to move the two-finger hand between the arrangements keeping the distance between the fingers exactly equal to δ. When two two-finger arrangements are δ-max-reachable, it is possible to move the two-finger hand between the arrangements keeping the distance between the fingers at most δ. Similarly, when two two-finger arrangements are δ-min-reachable it is possible to move the two-finger hand between the arrangements keeping the distance between the fingers at least δ. In Figure 3.1 no two displayed arrangements are δ-reachable, (a_1, b_1) and (a_2, b_2) are δ-max-reachable, and (a_1, b_1) and (a_3, b_3) are δ-min-reachable.

It is our aim to compute the sets $\mathcal{F}_{\leq\delta}$ and $\mathcal{F}_{\geq\delta}$ for all values of δ. It is easy to see that $\mathcal{F}_{\leq\delta} \subset \mathcal{F}_{\leq\delta'}$ for all $\delta \leq \delta'$. This monotonicity property suggests an approach where we

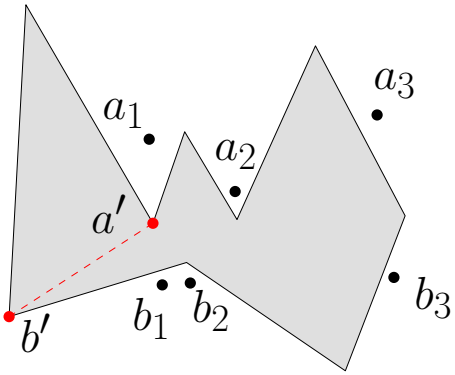


Figure 3.2: Critical maximum distances of (a_1, b_1) and (a_2, b_2) are equal to $||\overline{a'b'}||$ and critical maximum distance of (a_3, b_3) is zero.

increase δ from 0 to ∞, and consider the critical values of δ, at which existing components merge, or new components appear. Monotonicity implies that components do not split or disappear.

The *critical maximum distance* of a two-finger arrangement g is the smallest value of δ such that g does not lie in a lower-bounded component of $\mathcal{F}_{\leq\delta}$ or, equivalently, the supremum value δ for which g is a δ-squeezing caging arrangement. Figure 3.2 illustrates a number of two-finger arrangements and their critical maximum distances. We define the critical maximum distance of a two-finger arrangement g to be ∞ when the finger placements of g lie in disjoint components of F_r and F_s. Every two-finger arrangement in a lower-bounded component of $\mathcal{F}_{\leq\delta}$ has the same critical maximum distance. Every critical maximum distance is a critical distance for $\mathcal{F}_{\leq\delta}$, at which some lower-bounded component of $\mathcal{F}_{\leq\delta}$ merges with a component that is not lower-bounded. When g is a two-finger squeezing caging arrangement, then for all distances δ less than the distance between the finger placements of g, g lies in a lower-bounded component of $\mathcal{F}_{\leq\delta}$. Moreover, the critical maximum distance of g is more than the distance between the finger placements of g.

The set $\mathcal{F}_{\geq\delta}$ also grows monotonically when δ is decreased from ∞ to 0. Similarly, the *critical minimum distance* of a two-finger arrangement g is the largest value of δ such that g does not lie in an upper-bounded component of $\mathcal{F}_{\geq\delta}$ or, equivalently, the infimum value δ for which g is a δ-stretching caging arrangement. Figure 3.3 illustrates two arrangements and their critical minimum distances. If both finger placements of g lie in a bounded component of F_r and F_s respectively, we define its critical maximum distance to be 0. Every two-finger arrangement in the same upper-bounded component of $\mathcal{F}_{\geq\delta}$ has the same critical minimum distance. Every critical minimum distance is a critical distance for $\mathcal{F}_{\geq\delta}$, at which some upper-bounded component of $\mathcal{F}_{\geq\delta}$ merges with a component that is not upper-bounded. When g is a two-finger stretching caging arrangement, then for all distances δ more than the distance between the finger placements of g, g lies in an upper-bounded component of $\mathcal{F}_{\geq\delta}$. Moreover, the critical min-

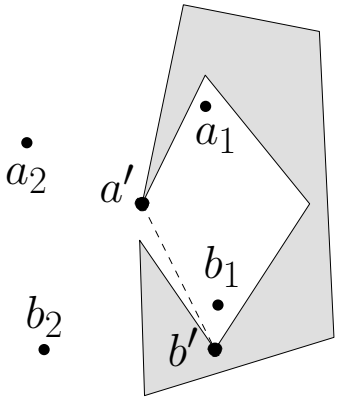


Figure 3.3: Critical minimum distance of (a_1, b_1) is equal to $||\overline{a'b'}||$ and critical minimum distance of (a_2, b_2) is ∞.

imum distance of g is less than the distance between the finger placements of g.

The set $\mathcal{F}_\delta$ does not have a monotonicity property when δ is continuously increased or is decreased. This lack of monotonicity makes it difficult to process $\mathcal{F}_\delta$ into a data structure for caging queries.

3.2 Two disk-finger caging

Let δ be the distance between the fingers of a two-finger hand. We prove in Subsection 3.2.1 that every caging δ-arrangement is a δ-squeezing caging arrangement or a δ-stretching caging arrangement (or both); using this fact we establish a relation between caging arrangements and immobilizing grasps in Subsection 3.2.2. In addition, we use this fact to report all caging arrangements by reporting all squeezing caging arrangements and stretching caging arrangements separately. To do that efficiently we use the monotonicity of $\mathcal{F}_{\leq\delta}$ and $\mathcal{F}_{\geq\delta}$. Since the squeezing and stretching caging arrangements can be computed similarly, we focus only on the computation of squeezing caging arrangements.

In Subsection 3.2.3, we define and construct *δ-max connectivity graph* to represent $\mathcal{F}_{\leq\delta}$ as a union of constant-complexity subcells. Every connected component of the graph corresponds to exactly one component of $\mathcal{F}_{\leq\delta}$. The pseudo-trapezoidations T_r and T_s of F_r and F_s respectively induce a decomposition of $\mathcal{F}_{\leq\delta}$ into constant-complexity four-dimensional cells. The δ-max connectivity graph is the adjacency graph on these four-dimensional cells.

Briefly, the algorithm works as follows: for all possible values of δ, we represent $\mathcal{F}_{\leq\delta}$ with the δ-max connectivity graph as a union of constant-complexity subcells to report the lower-bounded components, which consist of squeezing caging arrangements. We compute a sequence of distances *in increasing order* based on $\mathcal{T}$, as a superset of the critical distances of $\mathcal{F}_{\leq\delta}$, such that at each distance, the δ-max connectivity

graph can be updated by applying a constant number of changes. As we consider the distances of the sequence, we report a component of the δ-max connectivity graph as a set of squeezing caging arrangements, when the component corresponds to a lower-bounded component of $\mathcal{F}_{\leq\delta}$ and δ becomes equal to the critical maximum distance of all arrangements in that component. We present the algorithm and its running time in Section 3.3. We also obtain a data structure, based on the connectivity graph, that can be used to determine whether a given arrangement is caging.

3.2.1 Squeezing and stretching caging arrangements

In this section we prove that any caging δ-arrangement is a δ-squeezing or δ-stretching caging arrangement (or both). Using this fact we prove that a polygon with no pair of parallel edges can be caged with two fingers if and only if it can be immobilized with two fingers. First we provide some definitions and explain some concepts we have used in the rest of this section.

For any δ-arrangement (a, b), let $\mathcal{F}_{\leq\delta}(a, b)$ denote the connected component of $\mathcal{F}_{\leq\delta}$ that contains (a, b), and let $\mathcal{F}_{\geq\delta}(a, b)$ denote the connected component of $\mathcal{F}_{\geq\delta}$ that contains (a, b). The set $\mathcal{F}_{\geq\delta}(a, b)$ contains all δ-min arrangements that are in the same connected component of $\mathcal{F}_{\geq\delta}$ as (a, b) or, equivalently, all arrangements that are δ-min reachable from (a, b). Similarly, the set $\mathcal{F}_{\leq\delta}(a, b)$ contains all δ-max arrangements that are in the same connected component of $\mathcal{F}_{\leq\delta}$ as (a, b) or, equivalently, all arrangements that are δ-max reachable from (a, b).

Let α and β be two paths in F_r with the same endpoints, parametrized as functions from $[0, 1]$ to F_r. A *homotopy* between α and β is a continuous map $h : [0, 1]^2 \mapsto F_r$ such that $h(0, t) = \alpha(t)$ and $h(1, t) = \beta(t)$ for all $t \in [0, 1]$, and $h(s, 0) = \alpha(0) = \beta(0)$ and $h(s, 1) = \alpha(1) = \beta(1)$ for all $s \in [0, 1]$. If there is a homotopy between α and β, we say that α and β are *homotopic* and write $\alpha \simeq \beta$. (See e.g. Munkres's text book [Munkres, 1984].) The pair of paths α and β are *δ-max* if $||\alpha(t) - \beta(t)|| \leq \delta$ for any $0 \leq t \leq 1$, and are *δ-min* if $||\alpha(t) - \beta(t)|| \geq \delta$ for any $0 \leq t \leq 1$, and are *δ-exact* if $||\alpha(t) - \beta(t)|| = \delta$ for any $0 \leq t \leq 1$.

In the following two lemmas we prove that any caging δ-arrangement is δ-squeezing or δ-stretching caging arrangements (or both). In brief, if this is not the case then there is a caging δ-arrangement (a, b) that is neither δ-squeezing nor δ-stretching. We show that both $\mathcal{F}_{\leq\delta}(a, b)$ and $\mathcal{F}_{\geq\delta}(a, b)$ contain a non-caging δ-arrangement (a', b') at a distant location from P. Therefore, (a, b) is both δ-max reachable and δ-min reachable from (a', b'). Therefore, there is a δ-max pair of paths α_1 and β_1, in F_r and F_s respectively, such that α_1 starts from a and ends at a', and β_1 starts from b and ends at b'. Similarly, there is a δ-min pair of paths α_2 and β_2, in F_r and F_s respectively, such that α_2 starts from a and ends at a', and β_2 starts from b and ends at b'. We use these two pairs of paths to construct a δ-exact pair of paths α and β in F_r and F_s respectively, such that α starts from a and ends at a', and β starts from b and ends at b'. The existence of the δ-exact paths contradicts the assumption that there is a caging δ-arrangement that is neither δ-squeezing nor δ-stretching.

First we prove that the δ-exact paths α and β exist by assuming that $\alpha_1 \simeq \alpha_2$ and

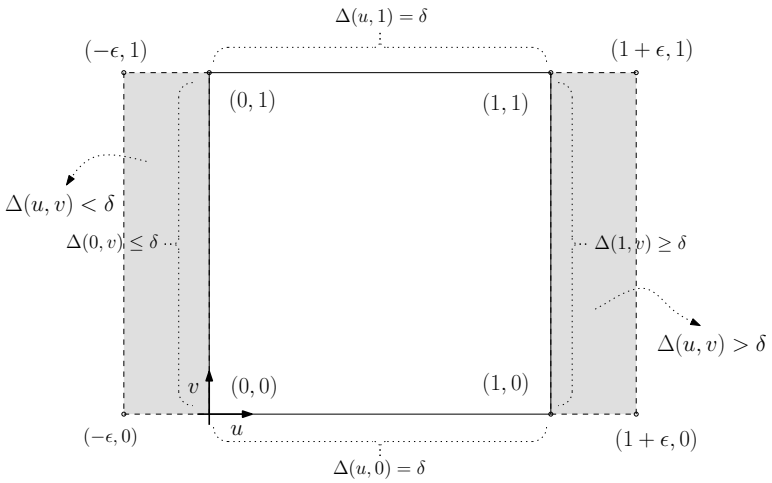


Figure 3.4: Illustration of Lemma 3.2.1 that displays the value of function Δ for some points of $\square$.

$\beta_1 \simeq \beta_2$.

Lemma 3.2.1. *Let (a,b) and (a',b') be arrangements in $\mathcal{F}_\delta$. Let α_1 and α_2 be paths in F_r from a to a', and let β_1 and β_2 be paths in F_s from b to b', such that $\alpha_1 \simeq \alpha_2$, $\beta_1 \simeq \beta_2$, α_1 and β_1 are δ-max, and α_2 and β_2 are δ-min. Then there is a path α in F_r from a to a' and a path β in F_s from b to b', such that $\alpha \simeq \alpha_1$, $\beta \simeq \beta_1$, and α and β are δ-exact.*

Proof. Fix an arbitrary homotopy $h_1 : [0,1]^2 \mapsto F_r$ from α_1 to α_2 and an arbitrary homotopy $h_2 : [0,1]^2 \mapsto F_s$ from β_1 to β_2. We observe that $h_1(u,0) = a$, $h_1(u,1) = a'$, $h_2(u,0) = b$, $h_2(u,1) = b'$, $h_1(0,v) = \alpha_1(v)$, $h_2(0,v) = \beta_1(v)$, $h_1(1,v) = \alpha_2(v)$, and $h_2(1,v) = \beta_2(v)$.

Consider the function $\Delta : [0,1]^2 \mapsto \mathbb{R}^+$ where $\Delta(u,v) = ||h_1(u,v) - h_2(u,v)||$. Figure 3.4 displays the value of function Δ for some inputs in $[0,1]^2$. Since both h_1 and h_2 are continuous, Δ is continuous as well. Moreover, we have $\Delta(0,v) \leq \delta$ and $\Delta(1,v) \geq \delta$ for all $v \in [0,1]$. The reason is that $\Delta(0,v) = ||\alpha_1(v) - \beta_1(v)||$ and $\Delta(1,v) = ||\alpha_2(v) - \beta_2(v)||$ for all $v \in [0,1]$. Similarly, we have $\Delta(u,0) = \Delta(u,1) = \delta$ for all $u \in [0,1]$. The reason is that $\Delta(u,0) = ||a - b||$ and $\Delta(u,1) = ||a' - b'||$ for all $u \in [0,1]$.

To simplify the proof, we will extend the function Δ to the slightly larger rectangular domain $\square := [-\epsilon, 1+\epsilon] \times [0,1]$ by defining $\Delta(u,v) = (1+u).\Delta(0,v)$ for all $u < 0$ and $\Delta(u,v) = u.\Delta(1,v)$ for all $u > 1$. The function Δ is continuous over this larger domain. Moreover, we observe that $\Delta(u,v) < \delta$ for all $u < 0$, and $\Delta(u,v) > \delta$ for all $u > 1$. Figure 3.4 displays the value of function Δ for some points of $\square$.

Let X be the set of points $(u,v) \in \square$ such that $\Delta(u,v) \geq \delta$, and let X_0 be the component of X containing point $(0,0)$. We observe that the points $(0,0)$ and $(0,1)$ are on the boundary of X_0. Now let Y denote the closure of $\square \setminus X_0$, and let Y_0 be the component of Y that contains the point $(0,0)$. Figure 3.5 displays the connected components X_0 and Y_0. Since X_0 is a single connected component and contains points on

the boundary of $\square$, the set Y_0 does not contain X_0, and thus is a connected component that does not contain holes; in particular, its boundary ∂Y_0 consists of a single cycle. Moreover, Y_0 contains the entire rectangle $[-\epsilon, 0] \times [0, 1]$ and is contained in the rectangle $[-\epsilon, 1] \times [0, 1]$. The reason is that X_0 does not contain any point in the interior of the rectangle $[-\epsilon, 0] \times [0, 1]$, but it contains the whole rectangle $[1, 1 + \epsilon] \times [0, 1]$. We again observe that the points $(0, 0)$ and $(0, 1)$ are on the boundary of Y_0. Thus, the set $Y_0 \cap [0, 1]^2$ is a path from $(0, 0)$ to $(0, 1)$. Let $\pi : [0, 1] \mapsto [0, 1]^2$ be an arbitrary parametrization of this path.

Finally, consider the paths $\alpha : [0, 1] \mapsto F_r$ and $\beta : [0, 1] \mapsto F_s$ where $\alpha(t) = h_1(\pi(t))$ and $\beta(t) = h_2(\pi(t))$. Our definitions imply that $||\alpha(t) - \beta(t)|| = \Delta(\pi(t)) = \delta$ for all $t \in [0, 1]$. The reason is that, all those points are on the common boundary of X_0 and Y_0. Moreover, α is homotopic to both α_1 and α_2, and β is homotopic to both β_1 and β_2. $\square$

In the following theorem, we construct two other δ-min paths α_2' and β_2', such that $\alpha_1 \simeq \alpha_2'$ and $\beta_1 \simeq \beta_2'$; then we apply Lemma 3.2.1 to prove that any caging δ-arrangement is a δ-squeezing or δ-stretching caging arrangement.

Theorem 3.2.2. *Given a polygon and a caging δ-arrangement, the arrangement is a δ-squeezing caging arrangement or a δ-stretching caging arrangement.*

Proof. We prove this by contradiction. Assume that there is a caging arrangement $(a, b) \in \mathcal{F}$ such that the distance between the fingers is δ, and it is neither a δ-squeezing nor a δ-stretching caging arrangement. Consider a δ-arrangement (a', b') at a remote location from P, which is neither a δ-squeezing caging arrangement, nor a δ-stretching caging arrangement. We will show that both $\mathcal{F}_{\leq\delta}(a, b)$ and $\mathcal{F}_{\geq\delta}(a, b)$ contain (a', b').

First, we observe that a must be in the unbounded component of F_r and b must be in the unbounded component of F_s. Assume for a contradiction that this is not the case; if both a and b are in bounded components, then (a, b) is a δ-stretching caging arrangement, and if exactly one of a and b is in a bounded component, then (a, b) is a δ-squeezing caging arrangement, contradicting the assumption that (a, b) is neither δ-squeezing nor δ-stretching caging arrangement.

The set $\mathcal{F}_{\leq\delta}(a, b)$ contains (a', b') because of the following two reasons. Firstly, $\mathcal{F}_{\leq\delta}(a, b)$ contains an arrangement where the finger placements coincide. Secondly, both a and b are in the unbounded components of F_r and F_s respectively. In this case, one can move the fingers at (a, b), keeping their distance at most δ, until they coincide, and then move the coinciding fingers to a location close to (a', b'), and finally to (a', b') itself.

Similarly, the set $\mathcal{F}_{\geq\delta}(a, b)$ contains (a', b'). Firstly, $\mathcal{F}_{\geq\delta}(a, b)$ contains an arrangement whose fingers are far from each other with respect to P. Secondly, both a and b are in the unbounded components of F_r and F_s respectively. In this case, one can move the fingers at (a, b), keeping their distance at least δ, until they are far enough from P and from each other, and finally to (a', b') itself.

Since both (a, b) and (a', b') are in $\mathcal{F}_{\leq\delta}(a, b)$ there is a δ-max pair of paths α_1 and β_1 in F_r and F_s respectively, such that α_1 starts from a and ends at a' and β_1 starts from b and ends at b'. Similarly, since both (a, b) and (a', b') are in $\mathcal{F}_{\geq\delta}(a, b)$ there is a δ-min

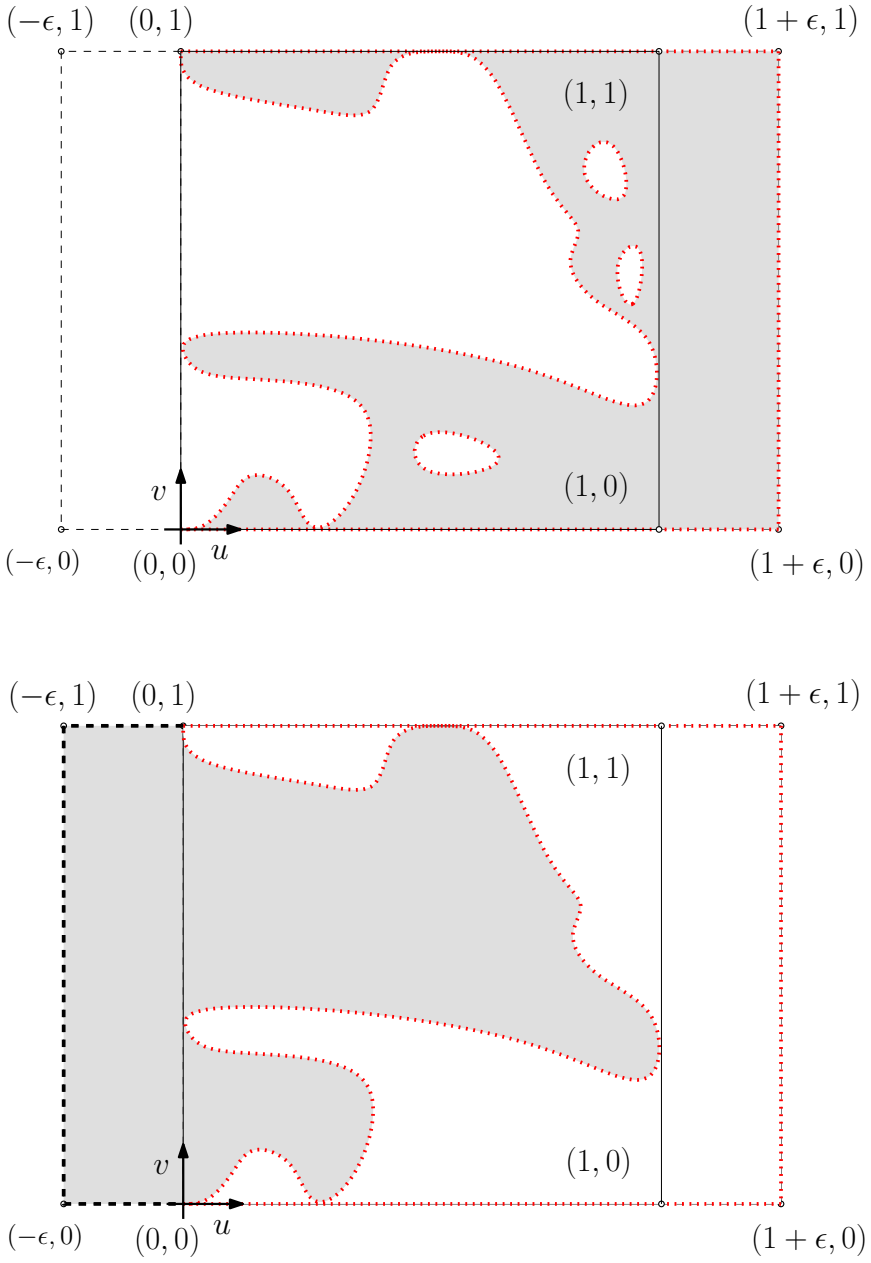


Figure 3.5: Illustration of Lemma 3.2.1 that displays the connected components X_0 (at the top) and Y_0 (at the bottom) with gray color.

pair of paths α_2 and β_2 in F_r and F_s respectively such that α_2 starts from a and ends at a' and β_2 starts from b and ends at b'.

The paths α_1 and α_2 have the same endpoints, but they are not necessarily homotopic. Similarly, the paths β_1 and β_2 have the same endpoints, but they are not necessarily homotopic. We construct two other δ-min paths α_2' and β_2' in F_r and F_s respectively, such that $\alpha_1 \simeq \alpha_2'$ and $\beta_1 \simeq \beta_2'$. Then we apply Lemma 3.2.1 to prove the theorem.

Let H_r be the convex hull[1] of P_r. Without loss of generality, there is a value t_0 (possibly equal to 0) such that $\alpha_2(t_0)$ is either on or outside H_r. If necessary, reparametrize α_2 and β_2 so that $t_0 = 1/4$. Let π be a path from $\alpha_2(1/4)$ to a' whose distance to $\beta_2(1/4)$ is always at least δ, and so that the path $\alpha_2[0, 1/4] + \pi$ is homotopic to α_1. For example, π could move directly away from $\beta_2(1/4)$ to a large circle surrounding P, around this circle as many times as necessary, and finally directly to a'. Without loss of generality, there is a value t_1 that the path β_1 intersects the circle centered at a' with radius δ for the first time (possibly equal to 1). If necessary, reparametrize α_1 and β_1 so that $t_1 = 1/2$. Let π' be a path from $\beta_1(1/2)$ to b' that moves around the circle, so that $\beta_1[0, 1/2] + \pi'$ is homotopic to β_1. Define new paths α_2' and β_2' as follows.

$$\alpha_2'(t) = \begin{cases} \alpha_2(t) & \text{if } 0 \le t \le 1/4 \\ \pi(4t-1) & \text{if } 1/4 \le t \le 1/2 \\ a' & \text{if } 1/2 \le t \le 1 \end{cases}$$

$$\beta_2'(t) = \begin{cases} \beta_2(t) & \text{if } 0 \le t \le 1/4 \\ \beta_2(1/4) & \text{if } 1/4 \le t \le 1/2 \\ \beta_2(3/4-t) & \text{if } 1/2 \le t \le 3/4 \\ \beta_1(4t-3) & \text{if } 3/4 \le t \le 7/8 \\ \pi'(8t-7) & \text{if } 7/8 \le t \le 1. \end{cases}$$

We easily verify that the pair of paths α_2' and β_2' are δ-min such that $\alpha_1 \simeq \alpha_2'$ and $\beta_1 \simeq \beta_2'$. As we consider α_1, α_2', β_1, and β_2', by Lemma 3.2.1 there is a δ-exact pair of paths from (a, b) to (a', b'). However, (a', b') is a non-caging arrangement and (a, b) is a caging arrangement. The existance of the δ-exact pair of paths contradicts the assumption that (a, b) is a caging arrangement, as there is a way to move the two fingers from (a, b) to (a', b') keeping the distance between the two fingers equal to δ, that shows (a, b) is a non-caging arrangement. □

3.2.2 Caging and immobilization

We establish a relationship between caging arrangements and immobilizing grasps for polygons that have no parallel edges. A grasp is called a *squeezing minimal grasp* if the distance between the fingers cannot be decreased locally; therefore (1) both finger placements are on the boundary of the polygon, (2) the grasp is a local minimum grasp with respect to the distance between the fingers, and (3) the line segment connecting the two finger placements locally intersects the polygon at both endpoints. Similarly,

[1]The convex hull of a set X in Euclidean space $\mathbb{R}^2$ is the smallest convex set containing X.

a grasp is called *stretching maximal grasp* if the distance between the fingers cannot be increased locally; therefore (1) both finger placements are on the boundary of the polygon, (2) the grasp is a local maximum grasp with respect to the distance between the fingers, and (3) both outward half lines emanating from the two finger placements locally intersect the polygon at both endpoints.

Lemma 3.2.3. *Every squeezing minimal grasp of a polygon without parallel edges is an immobilizing grasp.*

Proof. Consider a squeezing minimal grasp (a, b). Because no two edges of P are parallel, either a is at a concave vertex of P_r or b is at a concave vertex of P_s. Moreover, because the circular arcs on the boundary of P_r and P_s are convex outward, neither a nor b can lie in the interior of a boundary arc. Without loss of generality assume that a is at a vertex. Consider the circle centered at b that passes through a. Since the distance between a and b is more than $r + s$, within a small neighborhood of a, both edges (or arcs) of P_r incident to a are outside this circle. Therefore, both angles between the two tangent lines at a of the two incident features (edges or circular arcs) and the line segment ab are at least $\pi/2$. When an incident feature is a circular arc the angle is more than $\pi/2$.

Based on the features on which b is located there are two cases:

1. If b lies on an edge of P_s, that edge is perpendicular to segment ab. Therefore, the angle between the edge and ab is $\pi/2$. Since there is no pair of parallel edges, the angles between the two tangent lines at a and ab are more than $\pi/2$.

2. If b is at a vertex of P_s, then using the same argument both angles between the two tangent lines at b of the two incident features and ab, are at least $\pi/2$. Since there is no pair of parallel edges at most one of the four angles can be $\pi/2$.

Czyzowicz et al. [1999, Theorem 4] prove that any grasp satisfying these conditions is an immobilizing grasp. □

Lemma 3.2.4. *Every stretching maximal grasp of a polygon without parallel edges is an immobilizing grasp.*

Proof. Consider a stretching maximal grasp (a, b). Since no two edges are parallel, both a and b are at vertices of P_r and P_s respectively. Consider a circle with the line segment ab as its diameter. Within a small neighborhood of a, both edges (or arcs) of P_r incident to a are inside this circle. Similarly, within a small neighborhood of b, both edges (or arcs) of P_s incident to b are inside this circle. Gopalakrishnan and Goldberg [2002, Theorem 1] prove that any grasp satisfying these conditions is an immobilizing grasp. □

Corollary 3.2.5. *Let P be a simple polygon without parallel edges. The grasp in a lower-bounded component of $\mathcal{F}_{\leq\delta}$ that minimizes the distance between the fingers exists and it immobilizes P. Similarly, the grasp in an upper-bounded component of $\mathcal{F}_{\geq\delta}$ that maximizes the distance between the fingers exists and it immobilizes P.*

Lemma 3.2.6. *A simple polygon without parallel edges can be caged with two fingers if and only if it can be immobilized with two fingers.*

Proof. By Lemma 3.2.2 every caging arrangement is a squeezing caging arrangement or a stretching caging arrangement. Using Corollary 3.2.5 the claim follows. □

3.2.3 Connectivity graph

In this section we define a graph called *δ-max connectivity graph* to represent $\mathcal{F}_{\leq\delta}$ as a union of constant-complexity subcells. Recall from Section 3.1.2 that $\mathcal{T}$ is a decomposition of the four-dimensional admissible space $\mathcal{F}$ into cells of constant complexity. Intersecting each subcell of $\mathcal{T}$ with the set $\mathcal{F}_{\leq\delta}$ gives us a cell decomposition of $\mathcal{F}_{\leq\delta}$, which we denote $\mathcal{T}_{\leq\delta}$. The intersection of $\mathcal{F}_{\leq\delta}$ with a single subcell $\tau \in \mathcal{T}$ can be disconnected; we consider each connected component of $\mathcal{F}_{\leq\delta} \cap \tau$ to be a distinct subcell in $\mathcal{T}_{\leq\delta}$. We easily observe that the subcells of $\mathcal{T}_{\leq\delta}$ also have constant complexity.

The *δ-max connectivity graph* $G_{\leq\delta}$ is defined as follows. The vertices of $G_{\leq\delta}$ are the four-dimensional subcells in $\mathcal{T}_{\leq\delta}$. Two vertices are joined by an edge in $G_{\leq\delta}$ if and only if the interior of the union of the corresponding (closed) subcells in $\mathcal{T}_{\leq\delta}$ is connected.

Since every pseudo-trapezoid in T_r and T_s is adjacent to a constant number of pseudo-trapezoids, every subcell (t_1, t_2) of $\mathcal{T}$ is adjacent to a constant number of subcells (t_1', t_2') in $\mathcal{T}$; hence the total number of edges in $G_{\leq\delta}$ is linear in the total number of its nodes. Therefore, if there are $O(n)$ pseudo-trapezoids in T_r and T_s, there will be $O(n^2)$ nodes and edges in $G_{\leq\delta}$.

Every component of $\mathcal{F}_{\leq\delta}$ that is not lower-bounded is induced by a pair of intersecting components of F_r and F_s. Inside every pair of intersecting components of F_r and F_s we consider an arbitrary arrangement whose finger placements coincide. The resulting set of representative arrangements are contained in a set of nodes in $G_{\leq\delta}$ to which we refer as the representative nodes. Clearly, every representative node belongs to exactly one of the components of $\mathcal{F}_{\leq\delta}$ that are not lower-bounded. We label all nodes of the connected components of the representative nodes as nodes that are not lower-bounded. Then we label the rest of the nodes as lower-bounded nodes.

Although the associated δ-max-arrangements of the non-lower-bounded nodes are not δ-squeezing caging arrangements, some of them may still be caging arrangements. Recall that one δ-arrangement may belong to a lower-bounded node at distance $\delta_1 > \delta$ in $G_{\leq\delta_1}$ while it may belong to a non-lower-bounded node at a larger distance $\delta_2 > \delta_1$ in $G_{\leq\delta_2}$.

We can find the pseudo-trapezoid of T_r that contains a given point in $O(\log n)$ time. (See Subsection 2.2.4 on point location.) Similarly, we can find the pseudo-trapezoid of T_s that contains a given point in $O(\log n)$ time. Therefore, we can find the corresponding node of a given δ-max-arrangement in $G_{\leq\delta}$ each time in $O(\log n)$ time.

Lemma 3.2.7. *Given a polygon P and a distance δ, it is possible to compute $G_{\leq\delta}$ and whether each node is contained in a lower-bounded component in $O(n^2)$ time.*

Lemma 3.2.8. *After $O(n^2)$ preprocessing time, we can determine in $O(\log n)$ time whether a given two-finger δ-max-arrangement is a δ-squeezing caging arrangement.*

3.3 Two disk-finger caging algorithm

In this section we continue to focus on squeezing caging arrangements. We present an algorithm that reports all two-finger squeezing caging arrangements. The output consists of a set of constant-complexity four-dimensional cells corresponding to squeezing caging arrangements. Each point inside each reported cell corresponds to a two-finger squeezing caging arrangement of P. In addition a data structure is computed that can be used to answer whether a given two-finger arrangement is a squeezing caging arrangement of P.

Briefly, we consider all values of δ, and report the lower-bounded components of $\mathcal{F}_{\leq\delta}$ as the squeezing caging arrangements. As the δ-max connectivity graph $G_{\leq\delta}$ represents $\mathcal{F}_{\leq\delta}$ as a union of constant-complexity subcells, we compute $G_{\leq\delta}$ for all values of δ to report the lower-bounded components of $G_{\leq\delta}$ instead. As we increase δ, $G_{\leq\delta}$ changes at a sequence Δ of certain distances each of which is induced by either a single subcell or two adjacent subcells of $\mathcal{T}$. (A single distance may appear several times in Δ. The sequence Δ is a superset of the set of critical distances of $\mathcal{F}_{\leq\delta}$.) We compute Δ and then we consider the distances of Δ in increasing order. At each distance δ we compute the connectivity graph $G_{\leq\delta}$ by modifying $G_{\leq\delta'}$, where δ' is the distance just before δ in Δ. We report a lower-bounded component of $G_{\leq\delta}$ when the component merges with a component that is not lower-bounded (at which δ becomes equal to the critical maximum distance of that lower-bounded component); then the four-dimensional cells corresponding to the nodes of the connected component of $G_{\leq\delta}$ are reported as a set of δ-squeezing caging arrangements.

The algorithm consists of three steps as follows:

1. Compute the sequence Δ of distances induced by all subcells of $\mathcal{T}$,
2. Consider the distances of Δ in increasing order. For each distance δ, compute the connectivity graph $G_{\leq\delta}$ by modifying $G_{\leq\delta'}$, where δ' is the distance just before δ in Δ, and then report any nodes in components of $G_{\leq\delta}$ that are not lower-bounded and were in lower-bounded components of $G_{\leq\delta'}$.
3. Report the remaining squeezing caging arrangements for which the critical maximum distance is ∞.

Recall that as we increase δ, $\mathcal{F}_{\leq\delta}$ grows monotonically with δ (i.e. the cells can only merge or appear at the critical distances). Therefore, as we increase δ from zero, we distinguish two types of distances induced by a single subcell $\tau \in \mathcal{T}$, at which:

1. $\mathcal{F}_{\leq\delta} \cap \tau$ changes topologically (i.e. a cell appears or two cells merge in $\mathcal{F}_{\leq\delta} \cap \tau$),
2. considering a subcell $\tau' \in \mathcal{T}$ adjacent to τ, a cell of $\mathcal{F}_{\leq\delta} \cap \tau$ becomes adjacent to a cell of $\mathcal{F}_{\leq\delta} \cap \tau'$ inside $\mathcal{F}_{\leq\delta} \cap (\tau \cup \tau')$.

The sequence Δ is the sequence of distances induced by all subcells of $\mathcal{T}$. Since the first type of distances depends only on τ, the number of such distances is bounded by a constant for a given τ. From the fact that every subcell of $\mathcal{T}$ is adjacent to a constant

number of subcells and also $\mathcal{F}_{\leq\delta} \cap \tau$ has a constant number of cells, it follows that the number of distances of the second type is bounded by a constant as well. As a result, we can accomplish the computation of Δ in $O(n^2)$ time and sort its items in $O(n^2 \log(n))$ time.

In the second step we keep track of the changes in $G_{\leq\delta}$ while increasing δ by using a graph-based data structure which we call *squeezing caging graph* and display with $\mathcal{G}$. Consider a single subcell τ and its associated distances of the first type. Recall that at each such distance either a new cell appears or two cells merge together in $\mathcal{F}_{\leq\delta} \cap \tau$. As we increase δ from 0 to ∞, we incorporate a separate node in $\mathcal{G}$ for each appearing cell and each cell that results from merging of two cells in $\mathcal{F}_{\leq\delta} \cap \tau$. Since the number of distances induced by τ is bounded by a constant, the number of nodes in $\mathcal{G}$ associated with τ is bounded by a constant too. Moreover, given a δ arrangement in τ we can find its corresponding node in $\mathcal{G}$ in constant time by keeping the list of such distances and the list of associated nodes for τ. Therefore, for each distance δ, the graph $\mathcal{G}$ includes a node for every node that ever existed in $G_{\leq\delta}$. We label each node in $\mathcal{G}$ either lower-bounded or not lower-bounded. We will also associate a critical maximum distance with each node.

As we consider the distances of Δ, at each distance we take some actions to update $G_{\leq\delta}$ from $G_{\leq\delta'}$ within $\mathcal{G}$, where δ and δ' are the current and previous distances in Δ respectively. The actions taken to update the graph depend on the type of the distance and they follow in order:

1. We compute the edges of the new node (which has appeared or is the result of a merging). If there is no edge or the set of edges only connect to lower-bounded nodes, we label the new node as lower-bounded. Otherwise we label it as not lower-bounded. If the new node is connected to both a lower-bounded node and a node that is not lower-bounded, we perform a maximal report (see below). If the new node is the result of a merging, we add edges to connect the new node to the old nodes defining the new node.

2. We add an edge between the corresponding nodes. If the nodes have different labels, we perform a maximal report.

We do not remove the old nodes and edges from the graph, because they have no effect on the correctness and running time of the algorithm. More importantly, keeping the old nodes allows us to use the final resulted $\mathcal{G}$ as a data structure to efficiently answer whether a given arrangement is squeezing caging.

We perform a *maximal report*, when we label a previously lower-bounded node as not lower-bounded. This happens for squeezing caging arrangements for which δ is their critical maximum distance. Look at Figure 3.6 for some of the critical maximum distances that lead to *maximal report*. This operation consists of three parts: (1) we report the corresponding four-dimensional cells (excluding the two-finger arrangements that the distance between the fingers is less than $r + s$, i.e. intersecting each other) of all the nodes (excluding the old nodes) in the graph that are in the same connected component of the changing node; and (2) we set (including the old nodes) their associated critical maximum distances to the current value of δ; and (3) we label (including

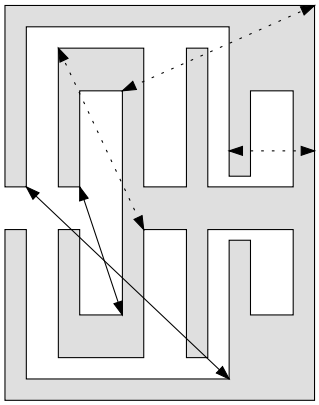


Figure 3.6: Three critical maximum distances displayed with dotted arrows and two critical minimum distances displayed with solid arrows.

the old nodes) them as not lower-bounded. We report every node at most once since a node that is not lower-bounded can never become lower-bounded again. Therefore, the total time devoted to reporting these cells and relabeling the nodes is linear in the number of nodes and therefore is $O(n^2)$.

To accomplish the third step of our algorithm, we consider all nodes of $\mathcal{G}$ for which the critical maximum distance equals zero while they are labeled as lower-bounded. These nodes correspond to two-finger arrangements for which exactly one finger placement is inside a bounded component of F_r or F_s respectively. For all these nodes, we set their critical maximum distance to ∞ and report their corresponding four-dimensional cells. The total time devoted to reporting these cells and adjusting the critical maximum distance of these nodes is also linear in the number of nodes and therefore $O(n^2)$.

If we exclude the time devoted to relabeling of nodes and also the time devoted to performing maximal report (which we have already discussed above), every update operation takes constant time. Because every change is local to a node and its neighbors, and the number of adjacent nodes and the number of edges for each node is constant. Therefore, the changes induced by a single distance of Δ take constant time in total. The following theorem follows from the preceding discussion.

Theorem 3.3.1. *Given a polygon with n edges and two disk fingers, it is possible to report all squeezing caging arrangements in $O(n^2 \log n)$ time.*

After reporting all squeezing caging arrangements, the final graph $\mathcal{G}$ forms a data structure that we can use to answer whether a given two-finger δ-arrangement is a δ-squeezing caging arrangement (where δ is induced from the placement of the given two fingers).

Theorem 3.3.2. *It is possible to compute a data structure requiring $O(n^2)$ space in $O(n^2 \log n)$ time, that can answer in $O(\log n)$ whether a given two-finger arrangement is a squeezing*

caging arrangement.

Proof. Consider a given two-finger δ-arrangement. We would like to know whether or not it is a δ-squeezing caging arrangement. After reporting all squeezing caging arrangements, we consider the final graph $\mathcal{G}$, and T_r and T_s. We find the node of $\mathcal{G}$ associated with the two-finger arrangement in $O(\log n)$ time. We compare the critical maximum distance m assigned to the node with δ. We report that the two-finger arrangement is a squeezing caging arrangement, if m is larger than δ; otherwise we report that the two-finger arrangement is not a squeezing caging arrangement. Therefore, the total time required to answer a query is $O(\log n)$. Clearly the space needed to store the data structure is $O(n^2)$. □

Similar results as Theorems 3.3.2 and 3.3.1 can be obtained for stretching caging arrangements. The two results together lead to the following main results of this section.

Theorem 3.3.3. *Given a polygon with n edges and two disk fingers, it is possible to report all caging arrangements in $O(n^2 \log n)$ time.*

Proof. By Theorem 3.2.2 every caging arrangement is squeezing caging or stretching caging. By Theorem 3.3.1 we can report all squeezing caging arrangements in $O(n^2 \log n)$ time. We can report all stretching caging arrangements similarly in $O(n^2 \log n)$ time. By reporting all squeezing caging arrangements and all stretching caging arrangements separately, we can report all caging arrangements in $O(n^2 \log n)$ time. □

Theorem 3.3.4. *It is possible to compute a data structure requiring $O(n^2)$ space in $O(n^2 \log n)$ time, that can answer in $O(\log n)$ whether a given two-finger arrangement is a caging arrangement.*

Proof. Consider a given two-finger δ-arrangement. If the δ-arrangement is neither a δ-squeezing caging arrangement nor a δ-stretching caging arrangement then according to Theorem 3.2.2 it is not a caging arrangement. According to Theorem 3.3.2, it is possible to check in $O(\log n)$ time whether a given δ-arrangement is a δ-squeezing or a δ-stretching caging arrangement by using a data structure that requires $O(n^2)$ space and can be computed in $O(n^2 \log n)$ time. □

We recall that the set of caging arrangements equals the union of the set of squeezing caging arrangements and the set of stretching caging arrangements. As we know that the complexity of a single subcell of $\mathcal{T}$ as well as the set of squeezing caging arrangements and the set of stretching caging arrangements contained in that subcell are constant, we can compute this union per subcell to obtain the set of all caging arrangements for each subcell. Therefore, it is clear that the total complexity of the set of all caging arrangements is $O(n^2)$.

3.4 Conclusion

We have presented complete algorithms for computing two disk-finger caging arrangements. The running time of the presented algorithm is not proportional to the

complexity of the output, but only to the complexity of the polygon P. We will present an algorithm which is more sensitive to the complexity of the output in chapter 4.

The squeezing and stretching fact has been extended to higher dimensions by Rodriguez and Mason [2008]. They have proven that also in higher dimensions every two-finger caging arrangement is a two-finger squeezing caging arrangement or a two-finger stretching caging arrangement (or both). However, extending the algorithm of this chapter that computes the whole set of two-finger caging arrangements of three-dimensional objects seems challenging, because of the problem of decomposing the admissible space into few simple cells. The convex decomposition technique proposed by Pipattanasomporn et al. [2007] provides a better way to handle the three-dimensional polyhedra. Unfortunately this technique is only applicable to point fingers, and thus the technique is not generalizable to disk fingers. Alternative ways should be looked for to tackle three-dimensional caging problems.

Chapter 4

Towards Output Sensitive Computation of All Two-Finger Caging Arrangements

In the problem of caging a polygon with two disk fingers, it is required to compute all arrangements of two disk fingers that cage a polygonal object with n edges in the plane. This problem was first tackled by Sudsang and Luewirawong [2003]. Their idea is to consider the immobilizing grasps at pairs of two concave vertices, or at a concave vertex and an edge. Taking into account the incident edges only, a local distance was computed for every immobilizing grasp that kept the fingers caging (neglecting the rest of the body of the polygon). As a result, the algorithm is incomplete as it reports only a subset of all caging arrangements of two disk fingers. We in chapter 3 and Pipattanasomporn and Sudsang [2006] independently have solved the problem for two fingers in $O(n^2 \log n)$ time, and also constructed a data structure capable of answering queries in $O(\log n)$ time. The running time of both solutions is independent of the complexity of the reported caging arrangements. Moreover the former solution [Pipattanasomporn and Sudsang, 2006] can be applied only to point fingers.

Several recent papers, including our work in Chapter 3 and also this chapter, distinguish between squeezing and stretching caging arrangements. Squeezing caging arrangements and stretching caging arrangement jointly constitute the set of all caging arrangements. A caging arrangement is a squeezing caging arrangement when the fingers cannot be moved to an arrangement in which the fingers coincide while keeping the distance between the fingers at most equal to the distance between the two finger placements of the given caging arrangement. A caging arrangement is a stretching caging arrangement when the fingers cannot be moved to an arrangement in which the fingers are far from each other with respect to the polygon while keeping the distance between the fingers at least equal to the distance between the two finger placements of the given caging arrangement. Recent work done by Pipattanasomporn et al. [2007]

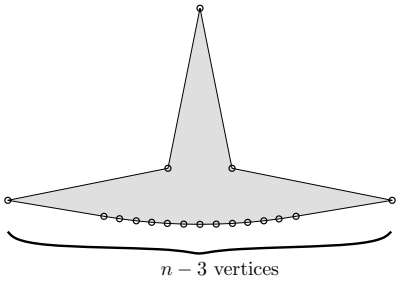


Figure 4.1: An example of a non-convex polygon that cannot be caged with two fingers

deals with computing the set of all squeezing caging arrangements of polygons and polyhedra with two point fingers using convex decomposition techniques. Although the running time is improved from $O(n^2 \log n)$ to $O(n^2 + mn + m^2 \log m)$ in which m is the number of convex partitions of the complement of the polygon, this solution is neither easily generalizable from point fingers to disk fingers, nor from squeezing caging arrangements to stretching caging arrangements.

The combinatorial complexity of the output—the set of caging arrangements—can vary greatly. On the one hand, there are non-convex polygons that cannot be caged with two fingers and therefore the output is empty (see Figure 4.1), while on the other hand there exist polygons for which the complexity of the set of caging arrangements is $\Omega(n^2)$. Certainly, we expect that our algorithm notifies us quickly in the former case, while we definitely find it acceptable if it runs longer in the latter case. The desirable property of achieving performance that is largely determined by the complexity of the output is referred to as *output-sensitivity* in algorithms research. In this chapter we take a first-step towards output-sensitive caging arrangement computation by proposing an algorithm for two disk fingers that runs in time $O(n^{4/3} \log^{2+\epsilon} n + M \log n + c \log(c + M))$, in which M is the number of so-called minimum or maximum arrangements, c is the complexity of the set of reported caging arrangements, and ϵ is an arbitrarily small positive constant. If the set of caging arrangements turns out to be empty, our algorithm will report so in $O(n^{4/3} \log^{2+\epsilon} n)$ time. Compared to existing algorithms, which all have a quadratic dependency on the complexity of the part, we have traded considerable sensitivity to the input for desirable output-sensitivity. Moreover, the output of the algorithm can also be efficiently queried to check whether a given *arbitrary* two-finger arrangement is caging in $O(\log n)$ time.

Having introduced some definitions and assumptions used in the paper in Section 4.1, then we outline our approach to solving the caging problem in Section 4.2. In Section 4.3 we define a new set of grasps called *minimum* and *maximum* grasps and we explain how to compute them efficiently. In Section 4.4 we present our algorithm and analyse the running time of the algorithm. We conclude the paper with a discussion of future work.

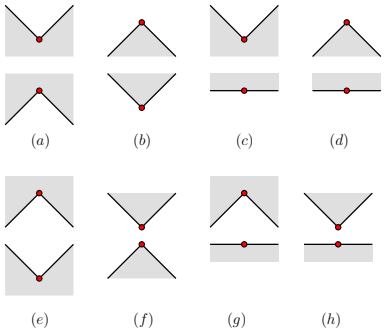


Figure 4.2: The figure displays a number of two-finger equilibrium grasps: the grasps (a), (b), (c), and (d) are squeezing equilibrium grasps while (e), (f), (g), and (h) are stretching equilibrium grasps; the grasps (a) and (c) are squeezing immobilizing grasps while (e) is a stretching immobilizing grasp.

4.1 Definitions and assumptions

This chapter addresses the problem of caging a simple polygon P with two disk fingers. We assume that the fingers have the same radius r. Formally, an object is caged with a number of fingers when its placement lies in a compact region of its free configuration space. Informally, an object is caged with a number of fingers when the placement of the fingers makes it impossible to take the object to infinity without penetrating any finger. Sometimes it is easier for the explanation to consider the polygon fixed and to move the fingers instead while keeping their mutual distances fixed. In this chapter we have used most of the notions we introduced in chapter 3 and thus we do not repeat them here.

Recall from Chapter 3 that every caging arrangement is a squeezing caging arrangement or a stretching caging arrangement (or both). There are also the same two types for equilibrium grasps. An equilibrium grasp however can be either an squeezing equilibrium grasp or an stretching equilibrium grasp. Figure 4.2 displays a number of two-finger equilibrium grasps.

4.2 Squeezing and stretching caging

Here we explain the main idea used to compute all squeezing and stretching caging arrangements. We focus on squeezing caging arrangements only, because the computation of stretching caging arrangements is analogous. The idea is as follows. First we efficiently compute all (squeezing and stretching) immobilizing grasps. Then we consider a parameter d as the distance between the fingers and we continuously increase it starting from the smallest squeezing immobilizing grasp. When d becomes equal to the distance between the fingers of a squeezing immobilizing grasp, we start a new lower-bounded component of squeezing caging arrangements. At any future

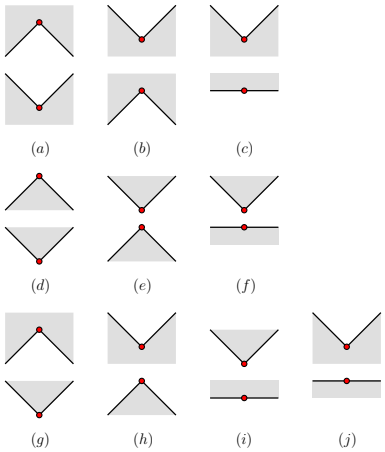


Figure 4.3: The figure displays all different types of minimum and maximum grasps.

stage in our algorithm, this component will be guaranteed to consist of all two-finger arrangements on the boundary of P that are d-max-reachable from the aforementioned squeezing immobilizing grasp. Increasing d the lower-bounded components grow bigger. When two lower-bounded components merge together they remain caging. We stop growing a lower-bounded component when it merges with a component that is not lower-bounded, which happens when the distance between the fingers is equal to the critical maximum distance of that lower-bounded component; alternatively, we strop growing a lower-bounded component when it cannot grow any more, that happens only when both fingers are inside disjoint bounded components of F_r, and the whole corresponding component of $\mathcal{F}$ has been explored. However, we continue to increase d until all lower-bounded components have reached their associated critical maximum distances.

As we have mentioned earlier the lower-bounded component of a squeezing immobilizing grasp for a distance d (which is smaller than the associated critical maximum distance) is the set of all arrangements with both fingers on the boundary of P that are d-max-reachable from the immobilizing grasp. To compute the lower-bounded components completely we compute another group of grasps called minimum grasps. Every squeezing immobilizing grasp is a minimum grasp but not vice versa. In Section 4.3 we define these grasps and then we explain how to compute them efficiently.

4.3 Minimum and maximum grasps

In this section first we formally define the minimum and maximum grasps and then we present an algorithm to compute them efficiently.

Consider the complete function $\lambda : (\partial P_r)^2 \rightarrow (\mathbb{R}^+ \cup \{0\})$ that maps any two-finger placement along the boundary of P_r onto the distance between the fingers in that place-

ment. In the three-dimensional space $(\partial P_r)^2 \times (\mathbb{R}^+ \cup \{0\})$ the continuous function λ can be regarded to consist of a number of (differentiable) patches, each corresponding to all placements of the fingers along a fixed pair of features (edges or circular arcs). Every patch is adjacent to a constant number of patches. The *minimum* and *maximum grasps* correspond to extremal points of the set of patches with respect to the distance between the fingers as the vertical axis. Every immobilizing grasp is a minimum or a maximum grasp but not vice versa.

Figure 4.3 displays all possible minimum and maximum grasps. The grasps of the first row are all immobilizing grasps and the grasps of the second row are all equilibrium grasps while the grasps of the last row are neither immobilizing grasps nor equilibrium grasps. The grasps (b), (c), (e), (f), (h), (i) and (j) are minimum grasps while (a), (d) and (g) are maximum grasps. As can be seen in all cases, every minimum or maximum grasp can be turned into an immobilizing grasp by appropriately exchanging interiors and exteriors at convex vertices and/or along edges. As a result, we can compute all these grasps efficiently by running an algorithm that outputs all immobilizing grasps on all $O(1)$ interior/exterior combinations of edges and concave vertices.

Let us first consider disk fingers with zero radii. We use Cheong et al.'s algorithms [Cheong et al., 2006] to compute all minimum and maximum grasps efficiently. Their algorithms take a set of edges and vertices as the input and report all two-finger immobilizing grasps as the output. To compute the minimum and maximum grasps, in addition to the set of concave vertices and edges we also consider the concave part of every convex vertex and also both sides of every edge as input to the algorithm that computes the immobilizing grasps [Cheong et al., 2006].

Now consider arbitrary disk fingers. Clearly in P_r every convex vertex appears as a circular arc and vice versa. In other words there is no concave circular arc in P_r. Therefore every non-immobilizing minimum or maximum grasp that involves an internal point of a circular arc of P_r corresponds uniquely to a minimum or maximum grasp respectively that involves the corresponding convex vertex of P instead of the circular arc. Hence to compute the minimum and maximum grasps with disk fingers, in addition to the set of concave vertices and both sides of every edge of P_r we also consider the concave part of every convex vertex of P as input to the algorithm that computes the immobilizing grasps [Cheong et al., 2006]. After the computation, we find the corresponding minimum and maximum grasps with disk fingers, by considering the corresponding circular arc of each involved convex vertex.

Let M be the number of non-immobilizing minimum and maximum grasps and I be the number of immobilizing grasps of a given polygon.

Lemma 4.3.1. *All minimum and maximum grasps can be computed in* $O(n^{4/3}\log^{2+\epsilon} n + I + M)$ *time in which* ϵ *is an arbitrarily-small positive constant.*

After computing all minimum and maximum grasps we classify them into four groups: squeezing immobilizing grasps, non-immobilizing minimum grasps, stretching immobilizing grasps, non-immobilizing maximum grasps. The first two types are used to compute the squeezing caging arrangements and the last two types are used

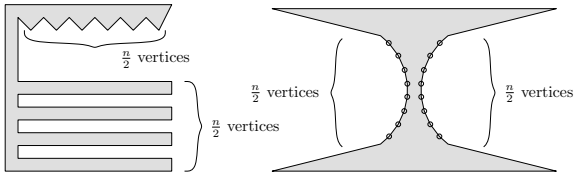


Figure 4.4: The figure (a) displays a polygon in which the number of minimum and maximum grasps is quadratic while the number of squeezing immobilizing grasps is linear. The figure (b) displays a polygon in which the number of minimum and maximum grasps is constant while the complexity of the set of caging arrangements is quadratic.

to compute the stretching caging arrangements.

Let c be the combinatorial complexity (i.e., the number of involved feature pairs) of the set of caging arrangements. The quantities M, I, and c can vary with respect to each other: Figure 4.4.(a) displays a polygon for which M is $\Omega(n^2)$ while both I and c are $O(n)$; Figure 4.4.(b) shows a polygon for which c is $\Omega(n^2)$ while I and M are only $O(1)$.

4.4 Algorithm

In this section we explain the algorithm. The algorithm maintains the set $\mathcal{B}$ of bounded components while increasing a distance parameter d from zero until each lower-bounded component reaches its critical maximum distance. There is a list $\mathcal{L}$ of grasps sorted in increasing order with respect to their distances between the fingers each of which is also associated with its inducing lower-bounded component (if it exists). At each grasp an event happens: a new lower-bounded component may appear, two lower-bounded components may merge, or a lower-bounded component and a component that is not lower-bounded may merge. Appearing happens only at an squeezing immobilizing grasp and in this case there is no inducing lower-bounded component. The two types of merging happen only at non-immobilizing squeezing equilibrium grasps. Though we maintain the lower-bounded components in $\mathcal{B}$ we do not keep an explicit representation of the other components, that are not lower-bounded.

The algorithm is as follows. Initially add all squeezing immobilizing grasps to $\mathcal{L}$, set the current distance parameter d to zero, and let $\mathcal{B}$ be empty. Repeat the following steps until $\mathcal{B}$ is empty and the current distance parameter d is larger than the largest squeezing immobilizing grasp:

- consider (and remove) the grasp $(a, b) \in \mathcal{F}$ with the smallest distance in $\mathcal{L}$, and consider its associated lower-bounded component κ (if it exists):
- if (a, b) is a squeezing immobilizing grasp:
  - form a new lower-bounded component κ for it and add it to $\mathcal{B}$

  - perform the local exploration of κ (explained in Subsection 4.4.1)
  - if the local exploration returned an squeezing equilibrium grasp, add the grasp as the next reachable squeezing equilibrium grasp of κ to $\mathcal{L}$; otherwise, if the local exploration returned *no-grasp*, output κ and ∞ as the critical maximum distance of κ, and remove κ from $\mathcal{B}$

- else if (a, b) is a squeezing equilibrium grasp
  - consider the next grasp $(a', b') \in \mathcal{F}$ with the smallest distance in $\mathcal{L}$, and consider its associated bounded component κ' (if it exists):
  - if (a, b) is the same as (a', b'),
    * merge κ and κ' into a new bounded component κ'' and replace both κ and κ' with κ'' in $\mathcal{B}$
    * continue with local exploration of κ'' (explained in Subsection 4.4.1)
    * if the local exploration returned an squeezing equilibrium grasp, add the grasp as the next reachable squeezing equilibrium grasp of κ'' to $\mathcal{L}$; otherwise, if the local exploration returned *no-grasp*, output κ'' and ∞ as the critical maximum distance of κ'', and remove κ'' from $\mathcal{B}$
  - else
    * output κ and the current value of d as the critical maximum distance of κ, and remove κ from $\mathcal{B}$

4.4.1 Lower-bounded components and local exploration

In this subsection first we explain how we represent a lower-bounded component, and then we explain how we form a new lower-bounded component and also how we perform a local exploration on a lower-bounded component. Every lower-bounded component κ contains a list $\mathbb{F}_\kappa$ of pairs of features (edges or circular arcs), a squeezing equilibrium grasp d_κ which we will explain later, and a list $\mathbb{D}_\kappa$ of grasps sorted in increasing order according to their distances between the fingers (each of which is larger than d_κ). Every pair of features is associated with a number of grasps. When a new pair of features is added to $\mathbb{F}_\kappa$ we add the associated grasps to $\mathbb{D}_\kappa$. The grasps associated with a pair of features are as follows:

1. the four grasps in which the fingers are placed at the endpoints of the features,
2. the grasps with the minimum distance between the fingers such that one of the fingers is at the endpoints of the features,
3. the grasps with the maximum and minimum distance between the fingers,
4. the non-immobilizing minimum grasps associated with this pair of features (explained below).

We associate every non-immobilizing minimum grasp to the grasp that results from freely (i.e. non-intersecting the polygon) squeezing the fingers along the line connecting the two fingers. To compute this associated grasp efficiently we use an algorithm from the computational geometry field referred to as ray shooting [Hershberger and Suri, 1995].

As we have seen in Section 4.4, for each squeezing immobilizing grasp a new lower-bounded component is formed when the distance parameter d becomes equal to the distance between the fingers of that grasp. When a new lower-bounded component is formed we add all adjacent pairs of features of the immobilizing grasp to $\mathbb{F}_\kappa$ and for each newly added pair of features we add its associated grasps to $\mathbb{D}_\kappa$. Initially we set d_κ to the immobilizing grasp.

We perform the local exploration on the lower-bounded component κ to find a reachable squeezing equilibrium grasp with the smallest distance between the fingers larger than that of d_κ. The local exploration is as follows:

- repeat the following steps until $\mathbb{D}_\kappa$ is empty
  - consider (and remove) the grasp with the smallest distance from $\mathbb{D}_\kappa$
  - if the grasp belongs to a squeezing equilibrium grasp such that at least one pair of its adjacent pair of features are not part of $\mathbb{F}_\kappa$ then set d_κ to the equilibrium grasp, return d_κ (and *terminate* the local exploration)
  - otherwise add the newly explored pairs of features to $\mathbb{F}_\kappa$ and for each newly added pair of features add its associated grasps to $\mathbb{D}_\kappa$
- if $\mathbb{D}_\kappa$ has become empty, set d_κ to *no-grasp*, return d_κ (and *terminate* the local exploration)

The newly explored features are either the result of local exploration of the adjacent pairs of features of $\mathbb{F}_\kappa$ or the result of a jump that happens when we consider the local non-immobilizing minimum grasps. The procedure returns *no-grasp* only when both fingers are inside disjoint bounded components of F_r and the whole corresponding component of $\mathcal{F}$ has been explored by the local exploration procedure, and so $\mathbb{D}_\kappa$ has become empty.

4.4.2 Algorithm analysis

In this subsection we analyze the running time of the algorithm explained in Subsection 4.4.1 that computes all squeezing caging grasps and its counterpart algorithm that computes all stretching caging grasps with both fingers on the boundary of P.

Lemma 4.4.1. *It is possible to report all caging grasps with both fingers on the boundary of P in $O(n^{4/3}\log^{2+\epsilon} n + M\log n + c\log(c+M))$ time in which c is the complexity of the output, M is the number of non-immobilizing minimum and maximum grasps, and ϵ is an arbitrarily-small positive constant. The output can be queried in $O(\log n)$ time to see whether or not a given grasp is caging.*

Proof. Using Cheong et al.'s algorithms [Cheong et al., 2006] we can check whether P can be immobilized with two fingers and also report all immobilizing grasps in $O(n^{4/3}\log^{2+\epsilon} n + I)$ time in which I is the number of immobilizing grasps. If there is no two-finger immobilizing grasp there will be no two-finger caging grasp. If I is not zero, then we change the input in a way such that their algorithms output all minimum and maximum grasps in $O(n^{4/3}\log^{2+\epsilon} n + I + M)$ time in which M is the number of non-immobilizing minimum and maximum grasps. The local exploration part of the algorithm spends $O(M \log n)$ time to compute the associated grasps of all non-immobilizing minimum and maximum grasps using the ray-shooting data structure and algorithm from the computational geometry field. The required data structure to perform ray shooting queries can be computed in $O(n)$ time and then each query can be performed in $O(\log n)$ time [Hershberger and Suri, 1995]. The total number of pairs of features added to $\mathbb{F}_\kappa$ for all lower-bounded components κ is proportional to the complexity of the output c. The total number of grasps added to $\mathbb{D}_\kappa$ for all lower-bounded components κ is proportional to the complexity of the output plus the complexity of the non-immobilizing maximum and minimum grasps. Therefore the total running time is $O(n^{4/3}\log^{2+\epsilon} n + I + n\log n + M\log n + (c+M)\log(c+M))$. Since $O(M\log(c+M))$ is in the worst case of the same order of $O(M\log n)$ we can simplify the running time to $O(n^{4/3}\log^{2+\epsilon} n + M\log n + c\log(c+M))$.

To check whether a given two-finger arrangement as a query is squeezing caging we freely squeeze the fingers along the line connecting the two fingers similar to what we do for the minimum grasps to find an associated grasp with both fingers on the boundary of P. As we have stored the critical maximum distance for each reported pair of features, we can first check whether the features of the associated grasp is among the reported pairs of features and if so then we compare the distance between the fingers of the given query two-finger arrangement with the critical maximum distance of the associated grasp. If the former distance is less than the latter distance then the given query two-finger arrangement is caging. □

4.5 Conclusion

We have presented a complete algorithm to compute all caging arrangements in which two equally-size disk fingers are placed on the boundary of a given polygon with n edges. The algorithm runs in $O(n^{4/3}\log^{2+\epsilon} n + M\log n + c\log(c+M))$ time, where M is the number of so-called minimum or maximum grasps, and c is the complexity of the reported output.

Algorithms with a running time that depends largely on the output are preferable over algorithms that solely depend on the input: informally speaking such algorithms tend to charge a user (running time) for what he gets. Compared to existing algorithms we have reduced the dependency on the input and replaced it by a certain dependency on the output complexity.

We believe that there are ways of reducing or even removing the dependency on M. We will discuss this issue in more detail here. One interesting way to extend our results

is to allow for queries with arbitrarily-sized fingers. Finally, we believe that many of the ideas in this chapter will extend to 3D and hence be useful for the computation of two-finger caging arrangements of polyhedra.

Here we describe our idea to compute the set of all two-finger caging grasps of a polygon output sensitively. We follow two steps to compute the set of all two-finger caging grasps, which we explain here. We assume that the set of all immobilizing grasps has been already computed.

1. In the first step, we consider a distance parameter and continuously increase it. When the distance parameter is equal to the distance between the finger placements of an immobilizing grasp we form a new bounded component. As we increase the distance parameter the existing bounded components grow by exploring the set of boundary grasps local (i.e. adjacent) to each component and whose distance is limited by the distance parameter. Two bounded components merge together when they explore the same grasp. The grasp at which merging occurs is necessarily a squeezing equilibrium grasp. A bounded component stops growing temporarily when it has locally explored a squeezing equilibrium grasp that has not been reached by any other bounded component so far. Every bounded component eventually stops at some distance to which we refer as the stopping distance of that component. We will prove that the largest stopping distance is the escaping distance of its associated bounded component.

2. After that all bounded components stopped growing, we follow the second step to report all caging grasps. Consider the list of bounded components. We perform a *non-local exploration* (explained below) on the bounded component b with largest stopping distance to find the set of bounded components B that are subsets of b. We report b and the set B and remove them from the list and we repeat the process with the bounded component that has the largest stopping distance in the remaining bounded components. The second step finishes when the list of bounded components becomes empty.

Here we explain how we perform the non-local exploration. We consider every edge on the boundary of the bounded component. We compute its Minkowski sum with a disk of radius equal to the escaping distance, and find its intersection with the polygon within an acceptable area. For every caging grasp involving this edge the other finger placement should be inside this Minkowski sum.

In the aforementioned two steps there is no need compute the minimum and maximum grasps. Certainly there are a number of lemmas that should be proven. For example, we should prove that the largest stopping distance is the escaping distance of its associated bounded component.

Chapter 5

Caging Polygons with Three Fingers

In the problem of caging a polygon with three fingers, the placements of two fingers, to which we refer as the base fingers, are given. It is required to find all placements of the third finger, such that the resulting fingers cage the polygon. Such placements of the third finger form a number of connected components in the plane, to which we will jointly refer as the *caging set*. Before Erickson et al.'s work [Erickson et al., 2007] the previous complete algorithms had been limited to robotic systems with a single degree of freedom [Davidson and Blake, 1998b, Rimon and Blake, 1996] whereas efforts to tackle robotic systems with multiple degrees of freedom had been limited to approximate algorithms that assume each finger can only interact with a single object edge [Sudsang, 2000, Sudsang and Ponce, 2000, Sudsang et al., 1999]. Given the placements of the base fingers which are required to be on the boundary of a convex polygon, Erickson et al. [Erickson et al., 2007] provided the first complete algorithm for computing the caging set for such polygons in $O(n^6)$ time. However, the problem of computing the caging set when the base fingers are not necessarily on the boundary of the polygon, or when the polygon is not convex, remained open. In this chapter we present a solution for computing the caging set for convex and non-convex polygons for a given placements of the base fingers. The main fact we have used to solve the problem is that the sections of the boundary of these sets that do not belong to the polygon boundary correspond to equilibrium grasps. Our work on three-finger caging uses this fact to extend the results by Erickson et al. [Erickson et al., 2007] from convex polygons to non-convex polygons, and also from the placement of the base fingers on the polygon boundary to arbitrary placements. The running time of our proposed three point-finger caging algorithm is $O(n^6 \log^2 n)$.

In Section 5.1 we introduce some notations and provide some definitions and explain some concepts we have used in this chapter. In Section 5.2 we give an overview of the algorithm that computes the caging set. In Subsection 5.2.1 it is shown that the sections of the boundary of the caging set that do not belong to the polygon bound-

ary correspond to equilibrium grasps. In Section 5.3 we explain our algorithm that computes the set of all three-finger caging arrangements.

5.1 Preliminaries

In this section we introduce some notations and provide some definitions and explain some concepts we have used in this chapter.

5.1.1 Notation

This chapter addresses the problem of caging a polygon P with three point fingers. Formally, P is caged with a number of fingers when its placement lies in a compact valid region of its free configuration space of P regarding the fingers as obstacles. Informally, P is caged with a number of fingers when the fingers make it impossible to take P to infinity without penetrating any finger. In general it is easier for the explanation to consider the polygon fixed and to move the fingers instead while keeping their mutual distances fixed. Therefore, P is caged when it is impossible to rigidly move the fingers to infinity without penetrating P.

The given simple polygon P is bounded by n edges. Define $F = \mathbb{R}^2 \setminus \text{int}(P)$. The set F is the set of all possible placements of a point finger not intersecting P. The set F is closed; in particular, it includes the boundary of P.

A *three-finger* arrangement is a triple (a, b, c) of points in the plane where the points a, b, and c are oriented counterclockwise. In this chapter, we omit "three-finger" and simply say caging arrangement instead of three-finger caging arrangement. We define the *admissible space* for three fingers as $\mathcal{F} = F \times F \times F$.

5.1.2 Definitions

Let T be the vertical decomposition of F into a set of trapezoids. Define $\mathcal{T} = T \times T \times T$. $\mathcal{T}$ is a decomposition of the six-dimensional admissible space $\mathcal{F}$ into cells of constant complexity.

If one three-finger arrangement can be transformed into another by a rigid transformation, we say that those two arrangements have the same *shape*. Our goal is to build a description of all caging arrangements with a given fixed shape. Let $\measuredangle\overrightarrow{ab}$ be the counterclockwise angle between the directed line $\overrightarrow{ab}$ and the positive x-axis. The three-finger arrangement (a, b, c) can also be described by six different parameters: the placement a of the first finger in the plane (requiring two parameters), $\measuredangle\overrightarrow{ab}$, $||a - b||$, $||a - c||$, and $||c - b||$, which we display as a tuple $(x, y, \theta, d, d', d'') \in \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^3$. The three-finger arrangement (a, b, c) of a triangular hand can be regarded to consist of a shape (d, d', d'') which specifies the placements of the fingers with respect to each other and is displayed with $\sigma(a, b, c)$, and a placement of the resulting rigid hand with parameters (x, y, θ). Using this representation of a three-finger arrangement, we separate shape from placement when we investigate the sets we define later.

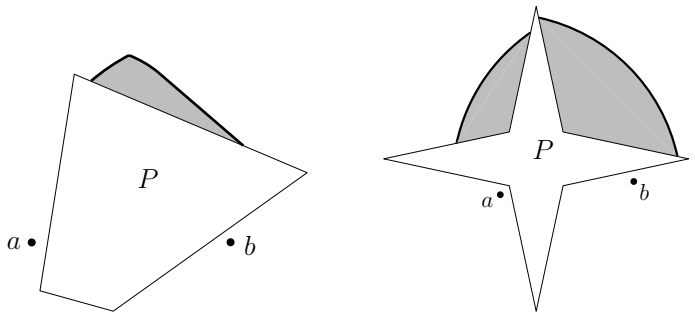


Figure 5.1: Two examples of caging set of a convex and a non-convex polygon.

Let

$$\mathcal{F}_\delta = \{\rho \in \mathcal{F} | \sigma(\rho) = \delta\}$$

where δ is the shape of a planar hand (with a fixed shape). Therefore, $\mathcal{F}_\delta$ is the set of all admissible placements of the hand that has the shape δ. A *δ-arrangement* is a member of $\mathcal{F}_\delta$. Two δ-arrangements are *δ-reachable* if both of them lie in the same connected component of $\mathcal{F}_\delta$. When two three-finger arrangements are *δ-reachable* it is possible to move the hand between the arrangements keeping the shape of the hand fixed (during which the polygon is also fixed).

Consider a given placement $(a, b) \in F \times F$ of the base fingers. Let $C(a, b) \subset F$ denote the set of all placements of the third finger that together with a and b form a three-finger arrangement that cages P. (Recall that the placements of the base fingers a and b, and the placement of the third finger should be oriented counterclockwise.) The set $C(a, b)$ consists of one or more connected components in F and is referred to as the *caging set* of (a, b) and its boundary, i.e. $\partial C(a, b)$, is referred to as the *caging boundary* of (a, b). Given P and a placement $(a, b) \in F \times F$ of the base fingers this chapter presents an algorithm to report $C(a, b)$. Let $K(a, b)$, to which we refer as the *caging curve,* denote the part of the boundary of the caging set that does not belong to the boundary of P; or formally $K(a, b) = \partial C(a, b) \setminus \partial P$. In Figure 5.1 two examples of caging set of a convex polygon and a non-convex polygon is dispalyed. In each case the light gray area is the caging set $C(a, b)$ and the caging curve $K(a, b)$ is displayed with bold curves.

5.2 Three-finger caging

In this section we solve the problem of computing the caging set for a given placement $(a, b) \in F \times F$ of the two base fingers. The relation between the caging curve and equilibrium grasps of P is the main fact used to solve the problem. In Subsection 5.2.1 it is shown that the third finger placed at a point on the caging curve jointly with the given placements of the base fingers corresponds to an equilibrium grasp of P. Therefore, the caging curve consists of a set of three-finger hand shapes of equilibrium grasps such that they involve two fingers with distance equal to d. Consider the

triangles induced by the shapes of all such equilibrium grasps with base fingers placed at (a, b). The placements of the point associated with the third finger induce a set of two-dimensional curves in the plane each of which has a constant complexity and is referred to as an *equilibrium curve*. Each equilibrium curve is induced by (equilibrium contact positions with) a single set of two or three features (edges or vertices) of P.

The fact that the equilibrium curves and the polygon boundary together form the boundary of the caging set of the base fingers implies that all points inside a single cell of the arrangement of the equilibrium curves and the polygon boundary are either caging or non-caging placements of the third finger. (Please note the difference between arrangement of curves and a three-finger arrangement.) We report the caging set by placing the third finger in every cell of the mentioned arrangement of equilibrium curves to find the cells consisting of caging arrangements.

To find out the caging status of a cell we consider the connected components of $\mathcal{F}_\delta$. Since F has exactly one component, there is exactly one unbounded component in $\mathcal{F}_\delta$ that corresponds to non-caging δ-arrangements. A cell is caging if and only if its corresponding component in $\mathcal{F}_\delta$ is bounded. To compute the connected components of $\mathcal{F}_\delta$ we use $\mathcal{T}$. (Recall that $\mathcal{T}$ is a decomposition of the $\mathcal{F}$ into cells of constant complexity.) In Subsection 5.2.2, based on $\mathcal{T}$ we define and construct a graph, called *connectivity graph*, to represent $\mathcal{F}_\delta$ as a union of constant-complexity subcells. Therefore, every component of the connectivity graph corresponds to exactly one component of $\mathcal{F}_\delta$. This way, the computation of the caging arrangements boils down to identifying the connected components of the connectivity graph. The complete algorithm and the running time analysis is explained in Section 5.3.

5.2.1 Equilibrium curves

In this subsection we define the set $\mathcal{E}_P(a, b)$ of so-called equilibrium curves; we prove that the third finger placed on a point on the caging curve jointly with the given placements (a, b) of the base fingers correspond to an equilibrium grasp. In Subsection 5.2.1 we enumerate all possible two- and three-finger equilibrium grasps involving two fingers with distance equal to d.

Let $\mathbb{T}_{(p,q)}^{(a,b)}[X]$ be the rigid transformation needed to map $(a, b) \in \mathbb{R}^2 \times \mathbb{R}^2$ to $(p, q) \in \mathbb{R}^2 \times \mathbb{R}^2$ applied to a set X. Let $E_P : \mathcal{F} \mapsto \{\text{False}, \text{True}\}$ be a predicate that determines whether a given grasp is an equilibrium grasp of P. Let

$$E_P(d) = \{(p, q, r) \in \mathcal{F} | E_P(p, q, r), ||p - q|| = d\}.$$

The set $E_P(d)$ is the set of all possible equilibrium grasps involving two and three fingers, such that the base fingers have a fixed distance equal to d. Draw the triangles defined by the fingers for every such grasp such that the base fingers be placed at two fixed points a and b. Let $\mathcal{E}_P(a, b)$ be the locus of the point associated with the third finger. Formally

$$\mathcal{E}_P(a, b) = \{c \in \mathbb{R}^2 | \exists (p, q, r) \in E_P(d) : \mathbb{T}_{(a,b)}^{(p,q)}[\{r\}] = \{c\}\}.$$

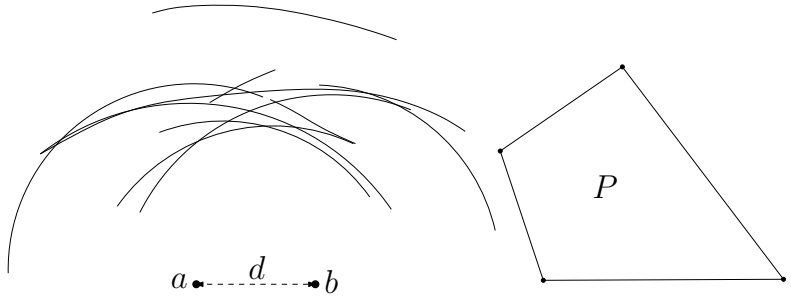


Figure 5.2: A number of equilibrium curves of the polygon P.

$\mathcal{E}_P(a,b)$ is the set of equilibrium curves at a reference location specified by placing the base fingers at a and b respectively. Let β be the number of pairs of edges of P that have two points with distance equal to d. The complexity of β is $O(n^2)$ (and this bound is tight in the worst case [Erickson et al., 2007]). The set $\mathcal{E}_P(a,b)$ contains $O(n\beta) = O(n^3)$ curves of constant degree. See Figure 5.2 for an example.

The following theorem establishes a relation between $K(a,b)$ (caging curve) and $\mathcal{E}_P(a,b)$. It shows that the points on $\partial C(a,b)$ (caging boundary) corresponds to ∂P or to $\mathcal{E}_P(a,b)$.

Theorem 5.2.1. $K(a,b) \subset \mathcal{E}_P(a,b)$.

Proof. Rimon and Blake proved that in a multi-finger one-parameter gripping system, the hand's configuration at which the cage is broken corresponds to an equilibrium grasp [Rimon and Blake, 1996, Proposition 3.3]. That means that as we start from a caging arrangement and continuously change the parameter (either increasing or decreasing the single parameter) at certain value of the parameter the corresponding hand's configuration becomes non-caging. That specific hand's configuration, at which the caging property changes from caging to non-caging, corresponds to an equilibrium grasp. It does not mean that the fingers necessarily form an equilibrium grasp at that placement, rather there exists a placement, reachable from that placement (therefore with the same shape), at which the fingers form an equilibrium grasp. To prove that $K(a,b) \subset \mathcal{E}_P(a,b)$, consider the intersection point c of an arbitrary line l and $K(a,b)$. As we slide the third finger along the line l the caging status changes at c. The base fingers at (a,b) and the third finger sliding on l form a three-finger one-parameter gripping system, and (a,b,c) is the hand's configuration at which the cage is broken. Therefore, the three-finger arrangement (a,b,c) corresponds to an equilibrium grasp (involving two fingers with distance equal to d). □

Types of equilibrium grasps and curves

In this subsection we enumerate all possible equilibrium grasps involving the third finger such that the distance between the base fingers is d. There will be two general cases depending on the number of fingers involved in the equilibrium.

1. *Two-finger equilibrium grasps that involve the third finger*. Since the base fingers should stay at distance d from each other, the base finger not involved in the equilibrium grasp can be at any place on a circular arc with radius d around the involved base finger not intersecting P. Depending on the features on which the two involved fingers are placed, there will be three cases. In all cases the distance between the two involved fingers, of which one is the third finger, is fixed. Therefore, since the distance between the third finger and one of the base fingers (the involved one) is fixed, the locus of the points associated with the third finger describes a circular arc in $\mathcal{E}_P(a, b)$ centered at a or b with radius equal to the fixed distance.

   (a) *along an edge and at a vertex*. In this case just one point on the edge gives an equilibrium grasp which is the intersection point of the altitude line drawn from the vertex to the edge. Since the length of the altitude line is fixed, the distance between the third finger and the involved base finger is fixed too.

   (b) *two vertices*. Since the distance between the two vertices is fixed, the distance between the third finger and the involved base finger is fixed too.

   (c) *along two edges*. In this case the two edges should be parallel and the line passing through the placements of the involved fingers should be perpendicular to both edges. Since the distance between the two parallel edges is fixed, the distance between the third finger and the involved base finger is fixed too.

2. *Three-finger equilibrium grasps of one of the following four subtypes.*

   (a) *a base finger at a vertex*. Since the distance between the base fingers is d, the other base finger should be placed at one of the intersection points of the polygon with the circle of radius d centered at the placement of the base finger placed at the vertex. The third finger can slide on an edge. Therefore, the locus of the points associated with the third finger describes a line segment in $\mathcal{E}_P(a, b)$.

   (b) *the third finger at a vertex*. In this case the base fingers can slide on two edges or a single edge at distance d. If the edges incident to the base fingers are not parallel and are different the locus of the points associated with the third finger describes a limaçon of Pascal in $\mathcal{E}_P(a, b)$, which is proven in Subsection 5.2.1; otherwise the locus is a line segment in $\mathcal{E}_P(a, b)$.

   (c) *all fingers on edges*. If the edges are not parallel, the locus of the points associated with the third finger describes a circular arc in $\mathcal{E}_P(a, b)$, which is proven in Subsection 5.2.1; otherwise the locus is a line segment in $\mathcal{E}_P(a, b)$.

   (d) *a base finger and the third finger at vertices*. Since the distance between the base fingers is d, the other base finger should be placed at one of the intersection points of the polygon with the circle of radius d centered at the placement of the base finger placed at a vertex. Therefore, the locus of all placements of the third finger describes a finite number of isolated points in $\mathcal{E}_P(a, b)$.

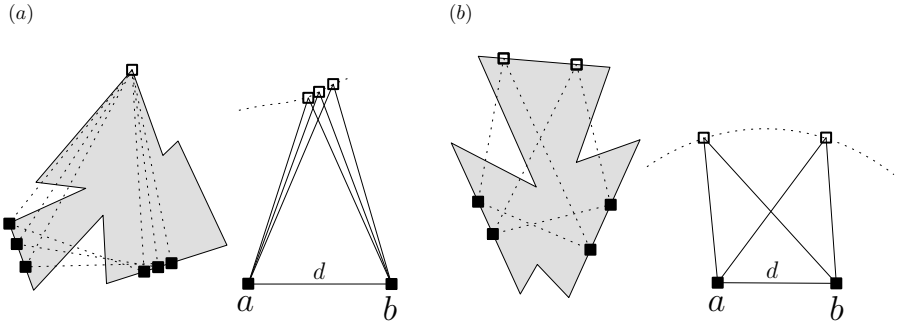


Figure 5.3: Two loci are displayed with dotted curves at the right side of two shaded polygons in which the filled boxes represent the base fingers and dotted triangles represent equilibrium grasps (see the text for the explanation).

> We can check the caging status of each of the isolated points separately in $O(n^3)$ time. Since there are $O(n^3)$ of them, we can check the caging status of all of them in $O(n^6)$ time in a brute force way. Therefore, we discard these points from $\mathcal{E}_P(a, b)$.

In Figure 5.3 two loci are displayed for two polygons. The filled boxes represent the base fingers and the empty boxes represent the third finger. Each dotted triangle represents an equilibrium grasp and is rigidly transformed to the reference locations a and b at the right side of the corresponding polygon. For the polygon (a), the third finger is at a vertex and the base fingers can slide at fixed distance d on two edges for which the locus of all placements of the third finger describes a limaçon of Pascal in $\mathcal{E}_P(a, b)$. For the polygon (b), all three fingers can slide on edges for which the locus of all placements of the third finger describes a circular arc in $\mathcal{E}_P(a, b)$.

Theorem 5.2.2. *The two- and three-finger equilibrium curves involving two fingers with distance equal to d are line segments, circular arcs, and limaçons of Pascal, hence they are two-dimensional curves of constant degree.*

Three fingers on three edges

Consider the triangles induced by all equilibrium grasps of three fingers on three edges such that the distance between the base fingers is d. Rigidly transform the base fingers such that they end up at the fixed points a and b. In this subsection we show that the locus of the point associated with the third finger describes a circular arc.

Consider the triangle $\triangle pqr$ in Figure 5.4 and an equilibrium grasp (u, t, s) formed by three point contacts on qr, rp, and pq respectively, such that $||s - t|| = d$. Since (u, t, s) is an equilibrium grasp the normal lines at the contact points meet at a common point. Let v be the point at which the normal lines meet. Let w be the intersection point of the normal line at u and a line passing through p parallel to qr. Without loss of generality assume that uw intersects pr, and let x be the intersection point.

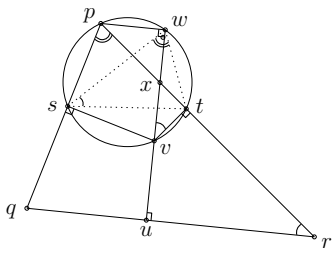


Figure 5.4: Proof for the locus of the third finger when the three fingers slide on edges

Since the triangles $\triangle psv$ and $\triangle vtp$ are right triangles, there is a circumscribed circle passing through the vertices of the polygon $psvt$ of which pv is the diameter. Since the length of st (equal to d) and the angle $\angle tps$ are fixed, the locus of p will be a circular arc with respect to a and b. Since $\angle uwp$ is a right angle, the point w lies on that circle as well.

Since the lines qr and pw are parallel and wu is perpendicular to qr, the length of wu is equal to the altitude line drawn from p to qr. Therefore, the length of wu is fixed. Hence if we show that w is a fixed point on the circle (with respect to the placements of s and t) then it follows that the locus of u is a circular arc with respect to a and b.

To prove that w is a fixed point we show that $\triangle wst$ and $\triangle pqr$ are similar triangles. On one hand, two angles $\angle wst$ and $\angle wvt$ are equal, because they face the same arc of the circle. Since the polygon $urtv$ is an inscribed polygon, two angles $\angle prq$ and $\angle wvt$ are equal. Therefore, $\angle wst$ is equal to $\angle prq$. On the other hand $\angle rpq$ and $\angle tws$ are equal, because they face the same arc of the circle. Therefore, $\triangle wst$ and $\triangle pqr$ are similar triangles.

Since $\triangle wst$ and $\triangle pqr$ are similar triangles and the length of st is fixed, the lengths of ws and wt are fixed too. Therefore, w is a fixed point with respect to a and b. The length of wu is also fixed, because it is equal to the length of the altitude line drawn from p to qr. Therefore, the locus of the point u describes a circular arc centered at w with respect to a and b.

Base fingers on two edges and the third finger at a vertex

Assume that the third finger is at the vertex r and the base fingers are on the two edges e_1 and e_2 at p and q respectively in Figure 5.6. Consider all possible triangles $\triangle pqr$ induced by sliding the points p and q on their corresponding edges such that the length of pq remains fixed. In this subsection we show that, as we draw all these triangles with the base fingers on a pair of fixed points at a and b (with distance equal to d) on the plane, the locus of the point associated with the third finger describes a limaçon of Pascal curve.

First we describe the limaçon of Pascal curve [Wells, 1991]. Consider a circle with radius l centered at the origin in Figure 5.5. Let s be a fixed point on the circle. Consider a line passing through s and t where t is an arbitrary point on the circle. On this line mark points u_1 and u_2 such that $||u_1 - t|| = ||t - u_2|| = k$ where k is a constant length.

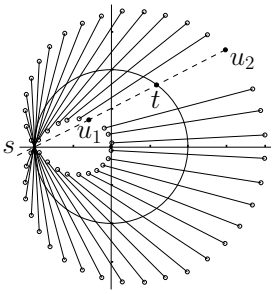


Figure 5.5: limaçon of Pascal curve

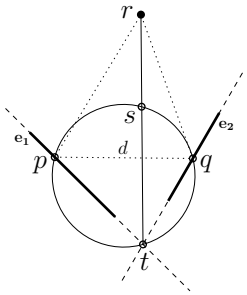


Figure 5.6: Proof for the locus of the third finger when the third finger is at a vertex and the base fingers slide on two edges

The locus of C is the limaçon of Pascal which is a constant-complexity curve of degree four. In this case the Cartesian formula of this curve is

$$(x^2 + y^2 - 2lx)^2 = k^2(x^2 + y^2).$$

Now we prove the claim. Look at Figure 5.6. Assume that the two edges incident to the base fingers intersect each other at t. Consider a circle passing through the points p, q and t. Let s be the intersection point of tr and the mentioned circle. On one hand since the point r is fixed with respect to the edges e_1 and e_2, the angle $\angle ptr$ is a fixed angle. On the other hand since the length of pq and the angle $\angle ptq$ are fixed, then the circle is fixed with respect to the points a and b, and the locus of t describes a circular arc of the circle with respect to a and b. Therefore, the point s is a fixed point with respect to a and b. Taking into account that the length of tr is also fixed, the locus of r with respect to a and b is a limaçon of Pascal curve.

5.2.2 Connectivity graph

Let δ be the shape of a three-finger hand. In this subsection we define a graph called *δ-connectivity graph* to represent $\mathcal{F}_\delta$ as a union of constant-complexity subcells.

Recall from Subsection 5.1.2 that $\mathcal{T}$ is now a decomposition of the six-dimensional admissible space $\mathcal{F}$ into cells of constant complexity. Intersecting each subcell of $\mathcal{T}$ with $\mathcal{F}_\delta$ gives us a cell decomposition of $\mathcal{F}_\delta$, which we denote by $\mathcal{T}_\delta$. The intersection of $\mathcal{F}_\delta$ with a single subcell $\tau \in \mathcal{T}$ can be disconnected; we consider each connected component of $\mathcal{F}_\delta \cap \tau$ to be a distinct subcell in $\mathcal{T}_\delta$. We easily observe that the cells of $\mathcal{T}_\delta$ also have constant complexity.

The δ-connectivity graph G_δ is defined as follows. The vertices of G_δ are the six-dimensional subcells in $\mathcal{T}_\delta$. Two vertices are joined by an edge in G_δ if and only if the interior of the union of the corresponding (closed) subcells in $\mathcal{T}_\delta$ is connected.

We label a connected component of G_δ as bounded if and only if it corresponds to a bounded component of $\mathcal{F}_\delta$. We label the connected component of G_δ that corresponds to the unbounded component of $\mathcal{F}_\delta$ as unbounded. We associate a component status, *'bounded'* or *'unbounded'*, with each node of G_δ depending on whether its connected component differs from the unbounded component.

All three-finger arrangements that are represented by nodes in the unbounded component of G_δ are non-caging arrangements. More importantly, all three-finger arrangements that are represented by nodes in other components of G_δ *are* caging arrangements. To compute the unbounded component of G_δ we consider a non-caging δ-arrangement and its corresponding node in G_δ. (We can easily find such a node by considering a δ-arrangement at a location remote from P with respect to P.) The connected component of G_δ that contains that node is the unbounded component.

Lemma 5.2.3. *Given a polygon P and a shape δ of a three-finger hand, it is possible to compute G_δ and the component status of all nodes in $O(n^3)$ time.*

Proof. Since every trapezoid is adjacent to a constant number of trapezoids in T, the total number of edges is linear in the total number of nodes. Therefore, if there are $O(n)$ trapezoids in T, there will be $O(n^3)$ nodes and edges in G_δ. □

We can find the trapezoid of T that contains a given point in $O(\log n)$ time. (See Subsection 2.2.4 on point location.) Therefore we can find the corresponding node of a given δ-arrangement in G_δ each time in $O(\log n)$ time.

Lemma 5.2.4. *After $O(n^3)$ preprocessing time, we can determine in $O(\log n)$ time whether a given three-finger δ-arrangement is a caging arrangement.*

5.3 Three-finger caging algorithm

In this section we present an algorithm that reports all placements of a point finger such that it cages P together with the given placement $(a, b) \in F \times F$ of the base fingers. The output of the algorithm is a set of disjoint and constant-complexity two-dimensional cells whose union forms the caging set of the base fingers. Each point inside every cell corresponds to a placement of the third finger that cages the polygon jointly with a and b. We assume that the given placements of the base fingers do not cage P alone without the third finger; otherwise the caging set is the entire space F. Using the result of two-finger caging, we can check this case in $O(n^2)$ time.

Recall that we display the arrangement of a set X of curves by $\mathcal{A}(X)$. The arrangement $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$ of curves is the arrangement of the equilibrium curves outside the interior of the polygon P. Clearly all points inside a cell of $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$, together with a and b correspond to three-finger arrangements that are either all caging or all non-caging; in other words $\mathcal{F}_\delta$ does not topologically change for all shapes δ specified by the placements of the base fingers a and b and the placement of the third finger inside a cell of $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$. A cell of $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$ is a *caging cell* if the points inside it together with a and b correspond to caging arrangements; otherwise it is called a *non-caging cell*.

We report the caging set by placing the third finger in every cell of $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$ to find the caging cells. To find out the caging status of a cell we consider the connected components of $\mathcal{F}_\delta$. A cell is caging if its corresponding component in $\mathcal{F}_\delta$ is bounded. We represent $\mathcal{F}_\delta$ with the δ-connectivigy graph G_δ as a union of constant-complexity subcells. Therefore, a cell is caging if its corresponding component in G_δ is bounded.

As we continuously change the shape δ by moving the third finger from one cell of $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$ to one of its adjacent cells, we cross a curve of $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$. (We will consider possible coinciding curves one by one.) Crossing a curve of $\mathcal{A}(\mathcal{E}_P(a,b) \setminus P)$ induces a single topological change to $\mathcal{F}_\delta$. However, as the graph G_δ also depends on the trapezoids of T, changes to G_δ are also implied by changes in the reachability of new triples τ of trapezoids as the shape δ changes. These changes are marked by additional curves in the plane. These curves turn out to be equilibrium grasps implied by triples $\tau \in \mathcal{T}$ of trapezoids.

By considering a triple of trapezoids $\tau = (t_1, t_2, t_3) \in \mathcal{T}$ as a rigid body, we can associate a number of equilibrium curves with τ which we display by $\mathcal{E}_\tau(a,b)$. Since τ has constant complexity, $\mathcal{E}_\tau(a,b)$ consists of a constant number of curves with constant complexity. Clearly $\mathcal{F}_\delta(\tau)$ does not topologically change for all shapes specified by the placements of the base fingers a and b and the placement of the third finger inside a cell of $\mathcal{A}(\mathcal{E}_\tau(a,b))$. For each cell of $\mathcal{A}(\mathcal{E}_\tau(a,b))$ there are a constant number of distinct cells in $\mathcal{F}_\delta(\tau)$, and so a bounded number of nodes is associated with τ in G_δ for that cell. We include all the nodes associated with τ in G_δ from the beginning. Since the number of cells in $\mathcal{A}(\mathcal{E}_\tau(a,b))$ is constant and for each cell the number of nodes associated with G_δ is constant too, the total number of nodes associated with τ in G_δ is constant. Moreover, given a three-finger arrangement in τ we can find its corresponding node in G_δ in constant time by keeping the list of associated nodes for each cell of $\mathcal{A}(\mathcal{E}_\tau(a,b))$. Hence the total number of nodes of G_δ is $O(n^3)$.

Let

$$\mathcal{E}_\mathcal{T}(a,b) = \bigcup_{\tau \in \mathcal{T}} \mathcal{E}_\tau(a,b)$$

which is the union of all equilibrium curves associated with all triples of trapezoids. As the edges of P are edges of T as well we have $\mathcal{E}_P(a,b) \subset \mathcal{E}_\mathcal{T}(a,b)$.

Clearly for an arbitrary triple of trapezoids τ, $\mathcal{F}_\delta(\tau)$ does not topologically change for all shapes specified by the placements of the base fingers a and b and the placement of the third finger inside a cell of $\mathcal{A}(\mathcal{E}_\mathcal{T}(a,b) \setminus P)$. Each boundary curve of a cell in $\mathcal{A}(\mathcal{E}_\mathcal{T}(a,b) \setminus P)$ corresponds to a topological change associated with a triple of

trapezoids (and its adjacent triples of trapezoids); hence it involves a local change in G_δ. Therefore, as we continuously change the shape of δ by moving the third finger from one cell of $\mathcal{A}(\mathcal{E}_\mathcal{T}(a,b) \setminus P)$ to one of its adjacent cells and thus crossing a curve of $\mathcal{E}_\mathcal{T}(a,b)$, we can update G_δ via addition or removal of constant number of edges. Recall that we include all nodes in the graph from the start, and so the only operation we perform is the addition or removal of edges.

The algorithm is as follows; we traverse the cells of $\mathcal{A}(\mathcal{E}_\mathcal{T}(a,b) \setminus P)$ one by one; by crossing every curve we update G_δ by performing a constant a number of changes. To maintain the connected components of the connectivity graph efficiently we use a graph-based data structure called fully dynamic graph [Holm et al., 1998, Henzinger and King, 1999]. Using the fully dynamic graph, it is possible to query for the connectivity of two nodes in the graph in $O(\log n / \log\log n)$ time and to update the mentioned data structure in $O(\log^2 n)$ time. We compute v_δ, a non-caging node, by choosing a grasp remote from the polygon, and finding the corresponding node in the graph in $O(\log n)$ time. Then we use the fully dynamic graph to query for the connectivity of v_δ to the nodes of the current cell, and report the current cell as a caging cell if there is no connectivity.

Theorem 5.3.1. *Given a polygon with n edges and the two placements of the base fingers, it is possible to report in $O(n^6 \log^2 n)$ time all placements of the third finger such that the three fingers jointly cage the polygon.*

Proof. We can compute T in $O(n \log n)$ time. Since the total number of equilibrium curves is $O(n^3)$ computing $\mathcal{A}(\mathcal{E}_\mathcal{T}(a,b) \setminus P)$ takes $O(n^6 \log n)$ time and its complexity is $O(n^6)$. To update the graph, adding and removing edges takes constant time for each crossing of a curve of $\mathcal{A}(\mathcal{E}_\mathcal{T}(a,b) \setminus P)$. However, maintaining the fully dynamic graph and querying the connectivity of the nodes take $O(\log^2 n)$ time for each addition or removal of edges. Therefore, the total running time of the algorithm is $O(n^6 \log^2 n)$. □

5.4 Conclusion

We have presented an algorithm for computing three point-finger caging arrangements. The running time of the presented algorithm is not proportional to the complexity of the output, but only to the complexity of the polygon P.

Extending the results to disk-shaped fingers seems straightforward. Although the curves of equilibrium grasps become more complicated, their algebraic degrees remain constant. Another interesting problem is to consider the three-finger caging queries which ask whether or not a given query of a three-finger arrangement is caging. We will consider three-finger caging queries for convex polygons in Chapters 8 and 9.

Studying special types of the polygons such as convex polygons or star-shaped polygons with the purpose of improving the worst-case running time for these types of polygons is interesting too. We will consider the complexity of the caging set for for convex polygons in Chapter 7.

Finally, extending our results to 3D seems challenging, because of the problem of decomposing the admissible space into few simple cells. Alternative ways should be looked for to tackle 3D caging problems.

Chapter 6

Caging Convex Polygons with Three Fingers: Properties

In the existing work on caging a polygon with three fingers the placements of two fingers, called the *base fingers*, are given. It is required to find all placements of the third finger such that the three fingers together cage the polygon. The set of placements of the third finger forms a number of two-dimensional regions to which we refer as the *caging set* of the third finger. In this chapter we provide several geometric facts about caging convex polygons with three fingers. The notations, concepts, and facts introduced in this chapter are extensively used in the next three chapters 7, 8, and 9.

As we introduce some definitions and assumptions used in the chapter in Section 6.1, we provide several geometric facts about caging convex polygons with three fingers in Section 6.2. In Subsection 6.2.1 we provide a number of geometric properties of three-finger caging arrangements of convex polygons and we show that the upper-boundary of the caging set is x-monotone. In Subsection 6.2.2 we establish a relation between immobilizing grasps and caging arrangements of convex polygons. In Section 6.3 we define a so called *canonical arrangement space* which is a three-dimensional space and is the set of all three-finger arrangements in which the base-fingers are on the boundary of the convex polygon. In the next subsections we present several properties of this space. In Subsection 6.3.1 we define a canonical placement q' for a placement q of a convex polygon P (inspired by the work done by Erickson et al. [2003]) and we establish a relation between the caging set of P at q and the caging set of P at q'. In Subsection 6.3.2 we define a three-dimensional object in canonical arrangement space, which is the result of sliding the convex polygon on both base fingers while keeping the contact with the base fingers in canonical arrangement space. Subsection 6.3.2 and the next two Subsections 6.3.3 and 6.3.4 investigate the properties of canonical arrangement space and the mentioned object defined in that space in order to model the caging problem in that space.

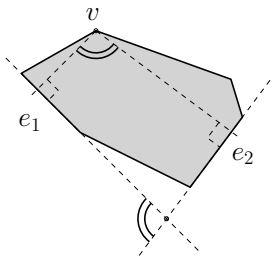


Figure 6.1: The angle between the altitude lines drawn from vertex v to edges e_1 and e_2 is equal to the supplement of the angle between e_1 and e_2.

6.1 Definitions and assumptions

This chapter addresses the problem of caging a convex polygon P with three point fingers. Formally, an object P is caged with a number of fingers when its placement lies in a compact region of its free configuration space. Informally, P is caged with a number of fingers when fixing the fingers it is not possible to take P to infinity without penetrating a finger.

A *convex polygon* is an intersection of a number of half planes; the intersection can be either unbounded (in which case it contains points at infinity) or bounded. Throughout the chapter P is a bounded convex polygon that has a fixed reference frame and n edges. We assume that P is an *open set*.

A *cone* is defined as the intersection of a pair of half-planes whose defining lines are not parallel. The intersection point of the defining lines of a cone is referred to as the *apex* of the cone. Thus, a cone is an unbounded convex polygon and its boundary consists of an apex and two half-lines emanating from the apex. The *angle* of a cone is the angle (of at most π) at the apex between the incident half-lines.

Recall that we have two base fingers at a fixed distance d. We assume that vertices and edges of P are in general position, i.e. no two vertices are at distance exactly d from each other, no vertex has (shortest) distance exactly d to an edge. In addition, we assume that the angle between the altitude lines drawn from any vertex to any pair of edges is not equal to the supplement of the angle between the corresponding pair of edges (i.e. they do not addup to π). (We use the last assumption in Subsection 6.2.2.) See Figure 6.1 to see an example that does not satisfy the last assumption.

Instead of considering rigid placements of the base fingers around the fixed polygon P, we equivalently fix the base fingers at $b_1 = (0, 0)$ and $b_2 = (d, 0)$ and consider possible placements $q \in \mathbb{R}^2 \times [0, 2\pi)$ of P (with respect to these fingers). Let $P[q]$ denote the set of points covered by P when placed at q. We define Q to be the set of all feasible placements of P with respect to the base fingers, so

$$Q = \{q \in \mathbb{R}^2 \times [0, 2\pi) | b_1 \notin P[q] \text{ and } b_2 \notin P[q]\}.$$

We define the caging set $C(q)$ of $q \in Q$ as

$$C(q) = \{c \in \mathbb{R}^2 | c \text{ cages } P[q] \text{ with } b_1 \text{ and } b_2\}.$$

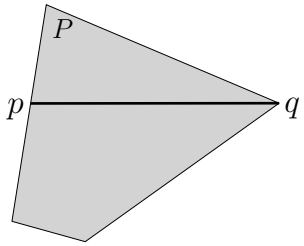


Figure 6.2: The length of the line segment pq equals the base-diameter of P.

Define $K(q) = \partial C(q) \setminus \partial P[q]$, which is a set of curves and referred to as the set of *caging curves* of q (see Figure 6.5). (Here ∂S denotes the boundary of a two-dimensional set $S \subset \mathbb{R}^2$.) Given $K(q)$, we can report $C(q)$ easily by considering the boundary of P as well.

Consider an orientation α. Let $P[\alpha]$ be the polygon P rotated by the angle α around its reference point for which both base fingers are in contact with it and the extensions of the edges touched by the base fingers intersect each other below the x-axis. We refer to $P[\alpha]$ as the *canonical placement* of any other placement of P with orientation α. The polygon $P[\alpha]$ may not be defined for all orientations α, which we explain about when we define the base-diameter.

Consider the length of the intersection of $P[q]$ with a horizontal line. Let the *base diameter* of $P[q]$ be the maximum-length intersection over all horizontal lines. A horizontal line that induces the base diameter is called a *base line*. When a base line intersects the interiors of two edges of $P[q]$, there are infinitely many (obviously equally-long) base lines. Otherwise, there is a unique base line that intersects P either at a vertex and an edge or at two vertices. The *lower base line* of $P[q]$ is the base line whose y-coordinate is the smallest among all base lines. Similarly, the *upper base line* of $P[q]$ is the base line whose y-coordinate is the largest among all base lines. All placements q of P with the same orientation have the same base diameter. When the base line is unique, the lower base line, the upper base line, and the base line itself are all the same. In Figure 6.2, $||\overline{pq}||$ is the base-diameter of the polygon P.

When the base-diameter of P with orientation α is less than d, $P[\alpha]$ is not defined, as it is not possible to place a translated copy of P with orientation α such that it touches both b_1 and b_2. A *critical angle* is an angle α for which d is equal to the base-diameter of $P[\alpha]$.

Every three-finger arrangement in which the polygon is at $P[\alpha]$ can be specified by $(p, \alpha) \in \mathbb{R}^2 \times [0, 2\pi)$, to which we refer as a canonical arrangement. The parameter α specifies the orientation of the polygon and $p = (x, y)$ specifies the location of the third finger. We refer to the space $\mathbb{R}^2 \times [0, 2\pi)$ of all such three-finger arrangements as the *canonical arrangement space*. The canonical arrangement (p, α) is the canonical arrangement of every other three-finger arrangement for which P is at a placement q and has orientation α.

Let the *vertically-adjacent feature* of the finger placement b_1 be the edge of $P[q]$ first

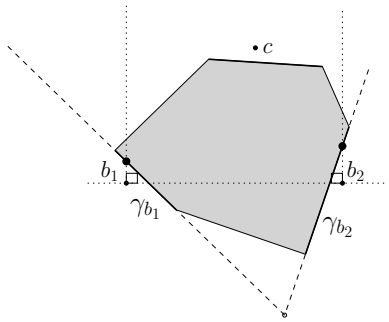


Figure 6.3: γ_{b_1} and γ_{b_2} are the vertically adjacent edges of b_1 and b_2 respectively and the extensions of the edges intersect each other below the x-axis.

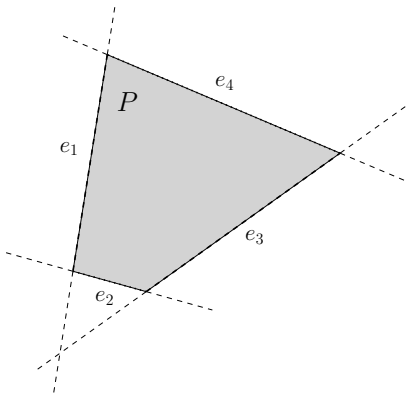


Figure 6.4: The triple of edges (e_1, e_2, e_3) is non-triangular, while the triple of edges (e_3, e_4, e_1) is triangular.

intersected by the vertical line drawn at b_1, which we denote by γ_{b_1}. If the vertical line intersects no edge then consider the vertical line itself as its vertically-adjacent feature. If the intersected feature is a vertex then γ_{b_1} is any of the adjacent edges of the vertex. Define γ_{b_2} similarly for b_2. (See Figure 6.3 to see the vertically adjacent edges of the given base fingers.) We will show later that if the extensions of γ_{b_1} and γ_{b_2} intersect each other on the x-axis, the caging set is empty. As we will see later we can check in $O(\log n)$ time whether the extensions of γ_{b_1} and γ_{b_2} intersect each other below the x-axis. If they intersect each other above the x-axis, we transform the coordinate system by placing the origin at the second base finger, and the first base finger at $(d, 0)$, and then renaming the base fingers. As a result, we can assume that the extensions of γ_{b_1} and γ_{b_2} intersect each other below the x-axis and that the lower base line of $P[q]$ is above the x-axis.

A triple of edges of P is called *triangular* if the supporting lines of the edges form a

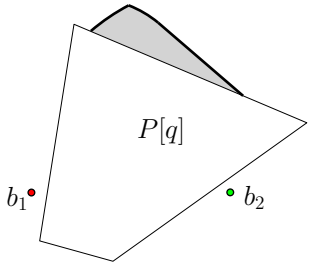


Figure 6.5: The light gray area is $C(q)$, and the set of bold curves is $K(q)$.

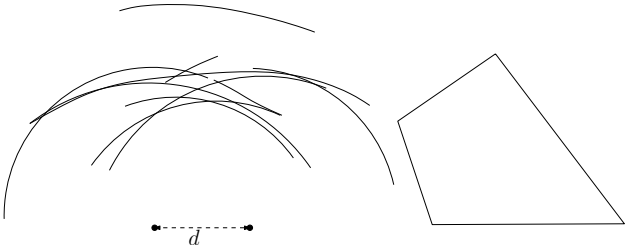


Figure 6.6: Some of the equilibrium curves of the polygon is displayed in which the distance between the base fingers is d.

triangle that encloses P, and is called *non-triangular* otherwise. In Figure 6.4, the triple of edges (e_1, e_2, e_3) is non-triangular, while (e_3, e_4, e_1) is triangular.

A valid three-finger arrangement is a three-finger arrangement in which none of the three fingers intersects the polygon. The set G of all valid three-finger arrangements is given by

$$G = \{(q, c) \in Q \times \mathbb{R}^2 | c \notin P[q]\}.$$

6.1.1 Caging and equilibrium grasps

Since we consider three-finger arrangements, the equilibrium can be established by all three, or just two of the three fingers. As a result, there can be either two-finger or three-finger equilibrium grasps. Since we have assumed general position of the vertices and edges, the two-finger equilibria must necessarily involve the third finger (i.e. non-base).

We define

$$E(q) = \{c \in \mathbb{R}^2 | c \text{ defines an equilibrium grasp of } P[q] \text{ (with } b_1 \text{ and } b_2)\}.$$

Let

$$\mathcal{E} = \bigcup_{q \in Q} E(q),$$

which is the set of all placements of the third finger that ever, i.e., for some placement q of P, form an equilibrium grasp of $P[q]$ (together with b_1 and b_2). Clearly these placements only exist for $q \in Q$ that satisfy at least $b_1 \in \partial P[q]$ or $b_2 \in \partial P[q]$ (or both, obviously, for three-finger equilibrium grasps). The set $\mathcal{E}$ contains a number of two-dimensional curves. (See Figure 6.6 for an example.) Every curve in $\mathcal{E}$ is called an *equilibrium curve* and corresponds to a maximally connected set of equilibrium grasps induced by the same triple or pair of features (edges or vertices) of P. An equilibrium curve is also called an *immobilizing curve* if the corresponding equilibrium grasps are immobilizing.

There are three types of curves constituting $\mathcal{E}$, namely line segments, circular arcs, and limaçons of Pascal by Theorem 5.2.2. The type of an equilibrium curve depends on the inducing features and the number of fingers invovled. Therefore depending on the number of fingers that induce the equilibrium grasp there are two general cases, that both consist of several subtypes;

1. Two-finger equilibria involving a base finger and the third finger

   (a) at two vertices;

   (b) at a vertex and along an edge;

   (c) along two parallel edges.

   In all three cases, since the distance between one of the base fingers and the third finger is fixed, the resulting equilibrium curves are circular arcs. The non-involved base finger is not necessarily on the boundary of the object.

2. Three-finger equilibria with

   (a) two fingers at vertices and one finger along an edge; in this case the third finger is necessarily at a vertex due to our general-position assumption. The resulting set of equilibria consists of a number of isolated points;

   (b) one finger at a vertex and two fingers along edges. There are two cases:

   - one of the base fingers is at a vertex and the other two fingers are along edges, in which case the resulting equilibrium curve is a line segment;
   - the third finger is at a vertex and the other two fingers are along edges, in which case the resulting equilibrium curve is a limaçon of Pascal if the base fingers are along two different edges, and the resulting equilibrium curve is a line segment parallel to the edge if the base fingers are along the same edge;

   (c) all three fingers along edges, in which case the resulting equilibrium curve is a circular arc and the equilibrium grasps are immobilizing if the edges are mutually different; the resulting equilibrium curve is a line segment in the other case that the three fingers are along two parallel edges.

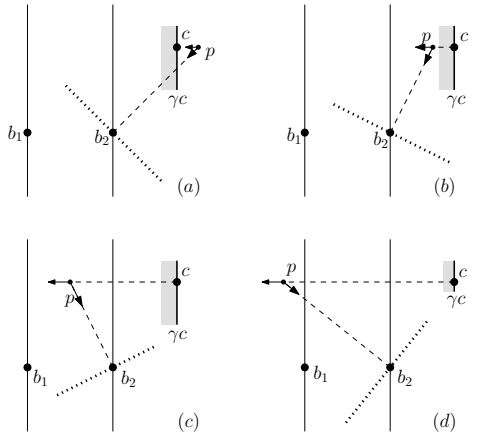


Figure 6.7: Illustration of Lemma 6.1.1.

Note that three-finger equilibria with all fingers at vertices do not occur because of our general-position assumption, which says that no two vertices are at distance exactly d.

In the following lemma we prove that the set of equilibrium curves for a convex polygon P contains no vertical segment.

Lemma 6.1.1. *There are no vertical segments in $\mathcal{E}$.*

Proof. An equilibrium curve of $\mathcal{E}$ can be a vertical line segment only when one of the base fingers is at a vertex and the third finger varies on an edge perpendicular to the x-axis. From the convexity of P and the fact that b_1 and b_2 lie on the boundary of P it follows that c cannot be between the vertical lines through b_1 and b_2, in order to be able to lie on a vertical edge. Now assume without loss of generality that it lies to the right of the vertical line through b_2. Clearly, the horizontal orthogonal at c must point leftward. From the fact that the orthogonals at b_1 and b_2 must intersect on the orthogonal of c and the condition that the directions of the orthogonals have to span $\mathbb{R}^2$ positively it immediately follows that the orthogonal at b_1 has to be directed upward and the orthogonal at b_2 has to be directed downward. (See Subsection 2.1.2 to read more about the notion positive spanning.)

In Figure 6.7 based on the point p where the orthogonals meet each other with respect to the vertical lines through b_1 and b_2 four cases are displayed. The perpendicular line to the line of action at b_2 is tangent to P at b_2, and thus P is on one side of this perpendicular line. In all four cases the perpendicular line is displayed with bold dotted line. In the first two cases (a) and (b) if the orthogonal at b_2 is directed downward then c cannot be on the boundary of P. In the next two cases (c) and (d), if the orthognal at b_2 is directed downward then b_1 cannot be on the boundary of P. □

Let β be the number of pairs of edges that contain a pair of points at distance d: β is $O(n^2)$ and this bound is tight even for convex polygons [Erickson et al., 2003]. The set

$\mathcal{E}$ contains $O(n\beta)$ constant-degree curves. The complexity of the arrangement $\mathcal{A}(\mathcal{E})$ of $\mathcal{E}$ is $O(n^2\beta^2)$.

A three-finger arrangement $(q, c) \in G$ is called *critical* if $c \in K(q)$. Using stratified Morse theory, Rimon and Blake [1996, Proposition 3.3] showed that in a multi-finger one-parameter gripping system, the hand's configuration at which the cage is broken corresponds to an equilibrium grasp. Using this fact In Subsection 5.2.1 we derived the following fact on a critical three-finger arrangement (q, c) which we state here as a Theorem.

Theorem 6.1.2. *If $c \in K(q)$, then there exists a placement $q' \in Q$ such that $c \in E(q')$ and $c \in K(q')$.*

We rephrase Theorem 5.2.1 as the following Theorem. (Recall that in contrast to what we do here, in Chapter 5 the polygon was fixed and we moved the fingers instead.)

Theorem 6.1.3. *$K(q) \subset \mathcal{E}$.*

By Theorem 6.1.3 we have the following Lemma.

Lemma 6.1.4. *The complexity of $K(q)$ is $O(n^2\beta^2)$.*

We will prove in Lemma 6.2.10 that the caging set does not contain isolated points. Therefore, we can safely remove the isolated points from $\mathcal{E}$ to compute the caging set.

6.1.2 Sliding a triangle on two lines

In this subsection we show that when we slide a triangle $\triangle abc$ with its two vertices a and b along two intersecting lines, the locus of the third vertex c follows an elliptical curve. Figure 6.8 illustrates the situation. The circle C_1 passing through a and b and the intersection point i of the two lines has a fixed radius. The length of the line connecting o to c is fixed too. When the triangle $\triangle abc$ slides along the two lines the center o of the circle C_1 follows a circle around i. The angle of the line segment connecting o to c changes at the same speed that the angle of the line segment connecting o and i changes, but at different directions. The locus of c is a special case of hypotrochoid in which the locus is an ellipse [Wells, 1991]. Let the radius of the circle C_1 be r_1 and the distance between o and c be r_2. The semi-major axis of the ellipse is $r_1 + r_2$ and the semi-minor axis is $|r_1 - r_2|$. The location of c follows a line segment in the degenerate case that $\measuredangle acb$ is equal to the complement of the angle between the two lines.

Consider a polygon P and a three-finger arrangement (a, b, c) of which two fingers a and b are along two non-parallel edges e_1 and e_2 of P and the third finger c is outside P. As we slide the three fingers (as a rigid body) along the two edges e_1 and e_2 of P (i.e. P remains in contact with a and b) the locus of the third finger c follows an elliptical curve. We use this fact in Subsection 6.2.2 to prove that a three-finger arrangement is not an immobilizing grasp if one of the finger placements is at a vertex.

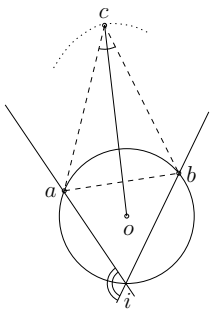


Figure 6.8: Considering rigid motion of a triangle, the location of the third vertex follows an elliptical curve when the two other vertices slide continuously on two different lines.

6.2 Properties of caging arrangements of convex polygons

In this section we provide several geometric facts about caging convex polygons with three fingers. In Subsection 6.2.1 we provide a number of geometric properties of three-finger caging arrangements of P and we show that $K(q)$ is x-monotone. In Subsection 6.2.2 we establish a relation between immobilizing grasps and caging arrangements of convex polygons.

From any point c outside $P[q]$ it is possible to draw two lines tangent to $P[q]$. These two tangent lines define two half-planes containing $P[q]$. (The two half-planes are identical if $c \in \partial(P[q])$.) We denote by $O(q, c)$ the intersection of the two half-planes defined by the point c, and we call it the *occluded cone* of the point c for the placement q of P.

We use the following abbreviations: $O_1(q) = O(q, b_1)$, $O_2(q) = O(q, b_2)$. When the finger placements b_1, b_2, and c cage $P[q]$ we simply say that c cages $P[q]$. Similarly, when the finger placements b_1, b_2, and c does not cage $P[q]$ we simply say that c do not cage $P[q]$.

6.2.1 Shape of the caging curve

In this subsection we prove that $K(q)$ is x-monotone. We start with a selection of intuitive properties of three-finger caging arrangements of convex polygons.

Property 6.2.1. *An unbounded convex polygon cannot be caged.*

Property 6.2.2. *Let $P \subset P'$. If c does not cage P' (with b_1 and b_2) then it does not cage P either.*

Corollary 6.2.3. *If there exists an unbounded convex polygon that contains $P[q]$ but none of the three finger-placements b_1, b_2, and c, then c does not cage $P[q]$.*

All results in the remainder of this subsection apply to any placement $q \in Q$.

Lemma 6.2.4. *If $O_1(q) \cap O_2(q) \cap O(q, c)$ is unbounded then c does not cage $P[q]$.*

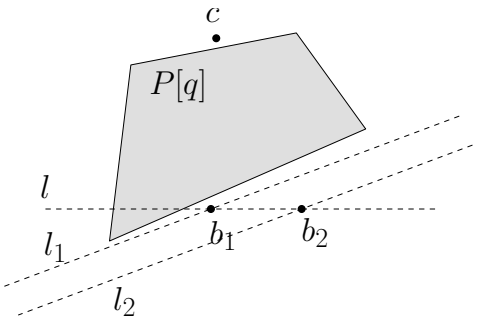


Figure 6.9: Illustration of Lemma 6.2.7.

Proof. Since $O_1(q)$, $O_2(q)$, and $O(q, c)$ are convex polygons, their intersection is convex too. Therefore, if their intersection is unbounded, then according to Corollary 6.2.3, c does not cage $P[q]$. □

If the finger placements b_1, b_2, and c are all along edges of the polygon $P[q]$, the occluded cones turn into half-planes. Corollary 6.2.5 reformulates Lemma 6.2.4 for this specific case.

Corollary 6.2.5. *If the finger placements b_1, b_2, and c are along edges of $P[q]$ whose extensions do not form a triangle enclosing $P[q]$ then c does not cage $P[q]$.*

Lemma 6.2.6. *If there are two parallel lines through the base finger placements b_1 and b_2 that do not intersect $P[q]$ then $C(q) = \emptyset$.*

Proof. $P[q]$ can escape by translation along at least one of the two directions along the parallel lines regardless of the placement of the third finger. Therefore, no placement of the third finger cages $P[q]$, which means $C(q) = \emptyset$. □

Consider a three-finger arrangement and the triangle defined by the finger placements as its three vertices. In the following lemma we prove that if at least one of the edges of the triangle is not intersected by the polygon, then the arrangement is not caging.

Lemma 6.2.7. *If c cages $P[q]$ then $P[q]$ intersects all three edges of $\triangle b_1 b_2 c$.*

Proof. Without loss of generality assume that $P[q]$ does not intersect the edge b_1b_2. Consider the line l that passes through b_1 and b_2. The points b_1 and b_2 divide the line l into three parts. Since $P[q]$ is convex, it intersects at most one of the parts of l. Since $P[q]$ does not intersect the edge b_1b_2, either it does not intersect the line l or it intersects the left-most or the right-most part of l. As a special case of Lemma 6.2.6 (in which case the two parallel lines are the same), if $P[q]$ does no intersect l then c does not cage $P[q]$. Without loss of generality and due to the symmetry assume now that $P[q]$ intersects the left-most part of l. Consider Figure 6.9. Since the line l intersects $P[q]$ and b_1 is closer to $P[q]$ than b_2, there is a line l_1 that passes through b_1 such that b_2 and $P[q]$

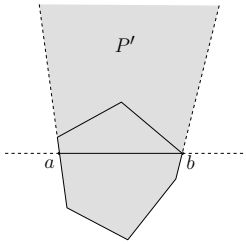


Figure 6.10: Illustration of Lemma 6.2.8.

are on opposite sides of l_1. Since b_2 and $P[q]$ are on opposite sides of l_1, the line l_2 that passes through b_2 and is parallel to l_1 does not intersect $P[q]$ either. According to Lemma 6.2.6, since the two parallel lines l_1 and l_2 that pass through b_1 and b_2 do not intersect $P[q]$, c does not cage $P[q]$. □

Consider a placement q of the polygon and let us assume that $P[q]$ intersects the line segment b_1b_2. Let b be the point at which the extensions of the vertically-adjacent features γ_{b_1} and γ_{b_2} intersect each other. (If γ_{b_1} and γ_{b_2} do not intersect each other, $C(q)$ is empty according to Lemma 6.2.6.) Since $P[q]$ intersects the line segment b_1b_2, the point b cannot be on the x-axis.

In the following two lemmas, we prove that $C(q)$ is restricted to be above the upper base line of $P[q]$ and also vertically above $P[q]$.

Lemma 6.2.8. *$C(q)$ is above the upper base line of $P[q]$.*

Proof. Consider Figure 6.10. Consider the upper base line ab of $P[q]$ and the two (parts of) edges immediately below its intersection points with the boundary of $P[q]$. If we extend these edges in upward direction and add the area between the resulting half-lines and above the lower boundary of $P[q]$ to P', then we obtain an unbounded convex polygon $P' \supset P[q]$. Any placement c of the third finger below the upper base line along with b_1 and b_2 will be outside P'. Corollary 6.2.3 says that such a placement c does not cage $P[q]$, and therefore $c \notin C(q)$. □

Lemma 6.2.9. *Every point in $C(q)$, is vertically above $P[q]$.*

Proof. Consider a point $c \in C(q)$. See Figure 6.11.*b*. Consider two vertical lines at b_1 and b_2 forming a strip Z. Since the lower base line of $P[q]$ is above the x-axis, at least one of the base fingers is vertically below $P[q]$. Therefore, there are two cases to consider: either both b_1 and b_2 are vertically below $P[q]$ or exactly one of them is vertically below $P[q]$. In the former case, if c is vertically below $P[q]$, at the right side of the right vertical line tangent to $P[q]$, or at the left side of the left vertical line tangle to $P[q]$, then $P[q]$ is free to escape by translating upward.

Now consider the latter case in which case exactly one of the base fingers is vertically below $P[q]$. Without loss of generality assume that b_1 is vertically above $P[q]$ and b_2 is vertically below $P[q]$. Since c cages $P[q]$, c cannot be on the left side of the left

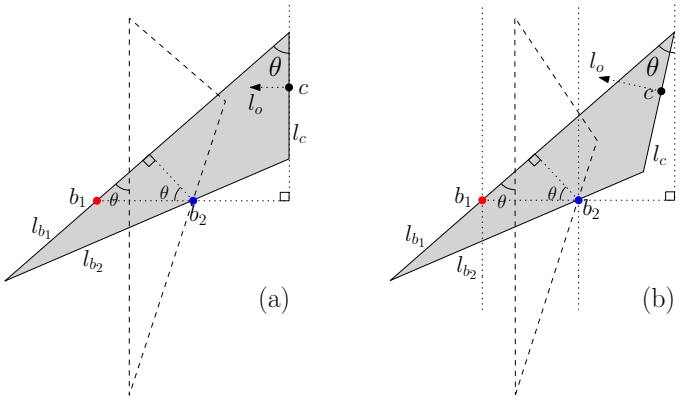


Figure 6.11: If c cages $P[q]$, then c should be vertically above $P[q]$. In both cases $P[q]$ is not displayed but it is inside the gray area. The pictures display the situations when (a) c is at the right side of $P[q]$ or (b) below $P[q]$.

vertical line tangent to $P[q]$. If c is at the right side of the right vertical line tangent to $P[q]$, then let l be a vertical half-plane passing through c that contains $P[q]$; otherwise if c is vertically below c, then let l be an arbitrary half-plane passing through c that contains $P[q]$. Let l_1 be a half-plane passing through b_1 containing $P[q]$ such that its defining line is parallel to the edge of $P[q]$ vertically below b_1. Similarly, let l_2 be a half-plane passing through b_2 containing $P[q]$ such that its defining line is parallel to the edge of $P[q]$ vertically above b_2. By Corollary 6.2.5, since c cages $P[q]$, l, l_1 and l_2 form a bounded triangle T containing $P[q]$. Since c cages $P[q]$, by Lemma 6.2.7 both line segments cb_1 and b_1b_2 are intersected by $P[q]$. Therefore, both b_1 and b_2 have to be on the left side of l.

We show that T can escape by first rotating around b_2 and then translating upward. Consider the orthogonal line l_o at c directed toward T. Since b_2 is on the left side of l_o, it is possible to rotate T around b_2 counterclockwisely and doing so the first obstacle that T encounters is b_1. When T encounters b_1 it has rotated for 2θ in which θ is the angle between the vertical line at c and the defining line of l_1. However, if we rotate T for θ then it is possible to escape T by translating it upward. Since T contains $P[q]$, $P[q]$ can escape as well that contradicts the assumption that c cages $P[q]$. Hence, the lemma follows. □

In the following lemma, we prove that if c cages $P[q]$, then all placements of the third finger that result from moving the third finger vertically downward also cage $P[q]$ (with b_1 and b_2).

Lemma 6.2.10. *Let c be a placement of the third finger that cages $P[q]$. Then any point c' vertically above $P[q]$ and below c also cages $P[q]$.*

Proof. For a contradiction, let c' be a point vertically above $P[q]$ and below c such that it does not cage $P[q]$. Let l^+ be the upward vertical half-line that emanates from c'

and similarly let l^- be the downward vertical half-line that emanates from c'. Since c' does not cage $P[q]$, there is a path for $P[q]$ to escape. Applying the same path for c, since c cages $P[q]$, the polygon has to intersect c and therefore l^+ at some placement of P. On the other hand since P is convex and c' does not cage $P[q]$, the polygon cannot intersect both l^+ and l^- in disjoint segments. Therefore, at some placement q' during the motion before intersecting c, the polygon $P[q']$ has to be completely on one side of the line $c'c$, either intersecting none of l^+ and l^-, or touching the line $c'c$ on a point or an edge.

Without loss of generality we can assume that $P[q']$ is on the left side of the line $c'c$. Then we distinguish two cases based on the positions of b_1 and b_2 with respect to the line $c'c$. (a) If at least one of b_1 or b_2 is on the right side of the line $c'c$, $P[q']$ and at least one of the base fingers are on opposite sides of the line $c'c$. (The line $c'c$ does not intersect $P[q]$). Therefore, c does not cage $P[q']$. (b) If both b_1 and b_2 are on the left side of the line $c'c$, then c does cage $P[q']$ by Lemma 6.2.9. □

The caging set does not contain isolated points by Lemma 6.2.10. Therefore, we can safely remove these points from $\mathcal{E}$ to compute the caging set.

Lemmas 6.2.9 and 6.2.10 lead to the following theorem, which is the main result of this subsection. Recall that the x-axis coincides with the line through the base fingers b_1 and b_2.

Theorem 6.2.11. *$K(q)$ is x-monotone.*

6.2.2 Immobilizing grasps and caging arrangements

In this subsection we establish a relation between immobilizing grasps and caging arrangements. (Recall that G is the set of all valid arrangements.) Consider the three-finger arrangement $(q,(c_x,c_y))$. Let $J((c_x,c_y))$ denote the set of all three-finger arrangements in which the third finger is confined to the fixed x-coordinate c_x and y-coordinate below c_y, i.e.

$$J((c_x,c_y)) = \{(q,(c_x,y')) \in G | y' \leq c_y\}.$$

$J((c_x,c_y))$ consists of several connected components. Let $j(q,(c_x,c_y))$ be the component that contains $(q,(c_x,c_y))$. In other words, $j(q,(c_x,c_y))$ is the set of all three-finger arrangements that are reachable from the three-finger arrangement $(q,(c_x,c_y))$ by moving $P[q]$ and keeping the third finger vertically below its initial placement at (c_x,c_y). $J((c_x,c_y))$ does not depend on a specific placement of P, while in $j(q,(c_x,c_y))$ the placement q is given. For any three-finger arrangement in $g \in j(q,(c_x,c_y))$, we say that the three-finger arrangement g is *sq-reachable* from the three-finger arrangement $(q,(c_x,c_y))$.

Lemma 6.2.12. *If c cages $P[q]$, then for any placement of the third finger c' and placement of the polygon q' such that $(q',c') \in j(q,c)$, c' cages $P[q']$.*

Proof. By definition, all three-finger arrangements reachable from a caging arrangement by a rigid motion are also caging. Moreover by Lemma 6.2.10 decreasing the distance between the third finger and the x-axis keeps the three-finger arrangement caging. □

Let $j^h(q,(c_x,c_y))$ be the slice of $j(q,(c_x,c_y))$ at $y=h$. Let $c=(c_x,c_y)$ be a placement of the third finger that cages $P[q]$. Start from $h=c_y$ and decrease the value of h. When $h=0$, the corresponding three-finger arrangements do not cage P, because all three fingers are collinear. As a result, every component in $j^h(q,(c_x,c_y))$ vanishes at a *minimal caging arrangement* $(q',(c_x,h))$ and a *minimal value* of h for a placement q' of P. In every vanishing component and a corresponding minimal caging arrangement $(q',(c_x,h))$ inside the component, the value of h is a local minimum in a sufficiently small neighborhood of $(q',(c_x,h))$ in $j(q,(c_x,c_y))$.

The following theorem establishes a relation between caging arrangements and immobilizing grasps by proving that the minimal caging arrangements are immobilizing grasps. It is shown that in a minimal caging arrangement (q',c'), $P[q']$ cannot move and is therefore immobilized.

Theorem 6.2.13. *If c' cages $P[q]$ then the local minima of $j(q,c')$ with respect to the y-coordinate are immobilizing grasps.*

Proof. Consider a local minimum (q',c) of $j(q,c')$ with respect to the y-coordinate and one of the corresponding placements. We show that the placement is not a local minimum if it is possible to move the fingers while keeping their mutual distances fixed. First we prove that all fingers have to be on $\partial P[q']$. Then we prove that all fingers are in the interior of edges of P.

If both base fingers are not on $\partial P[q']$ then (q',c) cannot be a local minimum, because it is possible to translate both the third finger and the polygon at q' together vertically downward until the polygon hits one of the base fingers. Now, assume for a contradiction that one of the fingers is not on $\partial P[q']$, which is not the third finger (because then the placement is trivially not a local minimum). Consider the orthogonals to the incident edges at the two fingers on $\partial P[q']$. According to Reuleaux's analysis it is possible to rotate the polygon clockwisely not intersecting the fingers about all points to the left side of both of the orthogonals, or counterclockwisely about all points to the right of both orthogonals; which moves the third finger (as well as the base fingers) off the boundary of P, after which the third finger can be squeezed. This contradicts the local minimum assumption. Moreover the two contacted edges are not parallel otherwise the three-finger arrangement would be non-caging by Lemma 6.2.6. Therefore all three fingers are on $\partial P[q']$. Moreover any possible motion of the polygon that causes at least one of the fingers to lose contact with P will lead to a contradiction.

Because of the general position assumption not all three fingers can be at vertices. Now that we know that all three fingers are at $\partial P[q']$, there are three cases to consider: exactly one of the fingers is at a vertex, two of the fingers are at vertices, or all fingers are on edges.

Consider the case that exactly one of the fingers is at a vertex. Assume that c is at a vertex v_c and e_1 and e_2 are the incident edges of b_1 and b_2 respectively. Rigidly

sliding the triangle $\triangle b_1 b_2 c$ on e_1 and e_2, the location of c follows an elliptical curve, or a line segment in the degenerate case that $\measuredangle b_1 c b_2$ is equal to the complement of the angle between e_1 and e_2. (Consider Subsection 6.1.2 to see why the locus is an elliptical curve.) The degenerate case is excluded by the assumption we made in Subsection 6.1. Therefore, we can assume that the locus of c is an elleptical curve. As a result, the polygon can slide at least in one direction along the base fingers and not collide with c. Effectively, c loses contact with P. When either b_1 or b_2 is at a vertex, we can reason similarly.

Consider the case in which exactly two of the fingers are on vertices. Without loss of generality assume the fingers b_2 and c are on vertices and e_1 is the incident edge of b_1. Then it is possible to find two lines l_{b_2} and l_c not intersecting $P[q']$ passing through b_2 and c respectively such that the three orthogonals at b_1, b_2, and c to e_1, l_{b_2}, and l_c respectively do not meet at a common point. By Lemma 6.2.12 the triangle induced by l_{b_2}, l_c, and e_1 encloses P. Therefore the mentioned orthogonals form a bounded triangle. According to Reuleaux's analysis it is possible to rotate the polygon around any point inside the bounded triangle at least in one direction causing c to lose contact with P.

Consider the case that all fingers are in the interior of the edges. By Lemma 6.2.12 these edges should induce a triangle enclosing P. The orthogonals at the three finger placements to the incident edges have to meet at a common point, otherwise, these orthogonals form a triangle such that from any point inside the triangle it is possible to rotate the polygon at least in one direction making a little space for c to be squeezed more. Since the extensions of the edges incident to the fingers form a triangle that contains P, and the orthogonals meet at a common point, by Theorem 2.1.1, the grasp is immobilizing. □

Corollary 6.2.14. *P can be caged if and only if it can be immobilized.*

By Lemma 6.2.13, a non-empty set of immobilizing grasps are sq-reachable from each caging arrangement. Increasing the y-coordinate of the third finger, at a certain distance from the x-axis, the three-finger arrangement becomes non-caging. This distance is called the *escaping distance* of that caging arrangement.

Property 6.2.15. *The escaping distance of a caging arrangement is equal to the escaping distance(s) of its sq-reachable immobilizing grasp(s).*

6.3 Canonical arrangement space

In this Section we define a so called *canonical arrangement space* which is a 3D space and is the set of all three-finger arrangements in which the base-fingers are on the boundary of the convex polygon. In Subsection 6.3.1 we define a canonical placement q' for a placement q of a convex polygon P and we establish a relation between the caging set of P at q and the caging set of P at q'. In Subsection 6.3.2 we define a three-dimensional object in canonical arrangement space, which is the result of sliding the convex polygon on both base fingers while keeping the contact with the base fingers

in canonical arrangement space. Subsection 6.3.2 and the next two Subsections 6.3.3 and 6.3.4 investigate the properties of canonical arrangement space and the mentioned object defined in that space in order to model the caging problem in that space. In Subsection 6.3.3 we consider a set of three-finger grasps (i.e. all three-finger placements are on the boundary of the convex polygon) in canonical arrangement space that are non-caging. In Subsection 6.3.4 we explain about two-finger equilibrium grasps of convex polygons and their representation in canonical arrangement space.

6.3.1 Canonical arrangements

Recall that a placement q' of P is a *canonical placement* of q if $P[q]$ and $P[q']$ are rigid translates, both b_1 and b_2 are on $\partial P[q']$, and the lower base line of $P[q']$ is above the x-axis. In this subsection, we prove that if c cages $P[q]$ and q' is the canonical placement of q, then it is possible to move the polygon from q to q' by a continuous translation of the polygon not intersecting the three fingers c, b_1 and b_2. Then, we show that $C(q) \subset C(q')$.

The vertically adjacent feature γ_c of c is the edge of the polygon vertically below c. The supporting line of the edge γ_{b_1} induces two half-planes of which only one contains $P[q]$ and is called the *vertically adjacent half-plane* of b_1 and is displayed with h_{b_1}. Define h_{b_2} and h_c similarly. A three-finger arrangement (q, c) *vertically encloses* $P[q]$ if the intersection of h_{b_1}, h_{b_2}, and h_c forms a triangle containing $P[q]$, c is vertically above $P[q]$, and $P[q]$ intersects all three edges of the triangle $\triangle b_1 b_2 c$. When a canonical arrangement (q, α) vertically encloses $P[q]$ then we simply say that the three-finger arrangement (q, α) is an *enclosing* arrangement. (In other words we use the term enclosing arrangements only for canonical arrangements, but we use the term vertically enclosing arrangements for all three-finger arrangements.)

Lemma 6.3.1. *If (q, c) is a caging arrangement, then it vertically encloses $P[q]$.*

Proof. Follows from Corollary 6.2.5 and Lemmas 6.2.7 and 6.2.9. □

Lemma 6.3.2. *Let (q, c) vertically enclose P, and q' be the canonical placement of q. Then $(q', c) \in G$ and it is possible to move P from q to q' by a finite number of collision-free translations.*

Proof. The lower base lines of both $P[q]$ and $P[q']$ are above the x-axis. At least one of the two finger placements b_1 and b_2 is vertically below $P[q]$. Therefore, when we translate P at placement q downward, it hits b_1 or b_2; let q'' be the resulting placement of P and without loss of generality assume b_2 to be the finger on the boundary of $P[q'']$. The three-finger arrangement (q'', c) vertically encloses $P[q'']$ and its vertically adjacent-features are the same as the vertically adjacent-features of (q, c). The lower base line of $P[q'']$ is above the x-axis.

In the rest of the proof, we will identify a cone such that if we translate the polygon along a vector inside the cone the polygon does not intersect c. (We say that a vector v is within a cone, if v can be expressed as convex combination of the two vectors

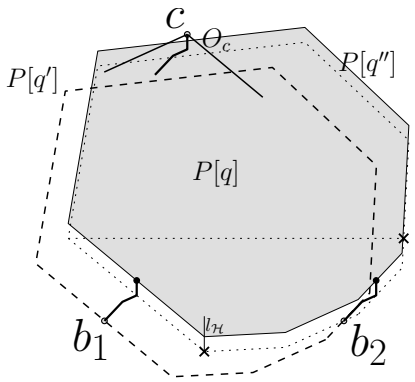


Figure 6.12: Illustration of Lemma 6.3.2.

along the boundary of the cone.) Then we show that there is a set of finite vectors that translates the polygon P at placement q'' to $P[q']$ collision freely.

Consider Figure 6.12. Consider a half-plane passing through c and parallel to the vertically adjacent-feature of b_2 such that it contains b_2. Similarly, consider a half-plane passing through c and parallel to the vertically adjacent-feature of b_1 such that it contains b_1. These two half-planes define a cone O_c whose apex is c. When the polygon is translated along a vector inside the cone O_c the point c does not penetrate the polygon.

Consider the lower horizontal line $l_{\mathcal{D}}$ tangent to $P[q]$ and a vertical line $l_{\mathcal{H}}$ passing through a tangency point on $l_{\mathcal{D}}$, both of which are attached to the polygon (i.e. when we translate the polygon the defined lines are translated with the polygon.) Since $P[q'']$ intersects the line segment b_1b_2, then the base fingers b_1 and b_2 are on opposite sides of $l_{\mathcal{H}}$, and they are between the lower base line of $P[q'']$ and $l_{\mathcal{D}}$.

Here we specify the set of vectors that translate the polygon P at placement q'' to $P[q']$. Let ∂P_s be the part of ∂P between the placement of b_2 on $\partial P[q'']$ and the placement of b_2 on $\partial P[q']$. Both of these placements are on ∂P, between the lower base line and $l_{\mathcal{D}}$, and on the same side of $l_{\mathcal{H}}$. The downward vectors along the edges in ∂P_s are all inside the cone O_c. Because, the first vector induced by the first edge is parallel to the vertically adjacent feature of b_2 and the edge parallel to the vertically adjacent feature of b_1 is above the lower base line. However all edges of ∂P_s are below the lower base line.

We translate the polygon by sliding the polygon on b_2 along the vectors determined by ∂P_s until b_1 resides on the boundary of the polygon. Since the set of translation vectors applied to the polygon are all inside the cone O_c, c does not intersect the polygon during the translation.

In Figure 6.12 the cone O_c is displayed and the set of translation vectors applied to the polygon is displayed for the points on the polygon boundary that will eventually rest on b_1 and b_2. As you can see the set of vectors are all inside the cone O_c. □

The following Theorem is a main result of this subsection. It proves that two verti-

cally enclosing arrangements that are rigid translates are reachable from each other by translation.

Theorem 6.3.3. *If both (q, c) and (q'', c) vertically enclose the polygon and q and q'' are rigid translates, then they are reachable from each other by a finite number of collision-free translations.*

Proof. The theorem follows, because by Lemma 6.3.2 both (q, c) and (q'', c) are reachable from the same canonical arrangement (q', c) by a finite number of collision-free translations. □

The following theorem is also a main result of this subsection. It proves that the caging set of a placement is a subset of the caging set of its canonical placement.

Theorem 6.3.4. *Let q be a placement of P and q' be its canonical placement. Then $C(q) \subset C(q')$.*

Proof. Consider a caging arrangement (q, c), so $c \in C(q)$. By Lemma 6.3.1, (q, c) vertically encloses $P[q]$. Since (q, c) vertically encloses $P[q]$, by Lemma 6.3.2, (q', c) is a valid three-finger arrangement and is reachable from (q, c) by a finite number of collision-free translations. Therefore, (q', c) and (q, c) are either both caging or both non-caging, so $c \in C(q')$. Hence, the theorem follows. □

Theorems 6.3.3 and 6.3.4 prove that if c cages $P[q]$ and q' is the canonical placement of q, then it is possible to move the polygon from q to q' by a continuous translation of the polygon not intersecting the three fingers c, b_1 and b_2.

6.3.2 Visibility in canonical arrangement space

In this subsection we define $\mathcal{P}$ in canonical arrangement space as a three-dimensional object defined by sliding the polygon P on both base fingers (i.e. keeping the contact with both base fingers). The surface patches of $\mathcal{P}$ play an important role in Chapter 7 in establishing a bound on the complexity of the caging set. Recall that every canonical arrangement corresponds uniquely to a point in canonical arrangement space. As we consider the surface patches of $\mathcal{P}$ as obstacles, we define visibility between two points in canonical arrangement space (with the same x and y coordinates) as a sufficient condition that the corresponding canonical arrangements have similar caging properties. Then we explain a number of properties of the surface patches of $\mathcal{P}$.

Consider an edge e of $P[\alpha]$. The edge e together with the edges e_1 and e_2 touched by the base fingers forms a triple of edges. Consider a motion of P at orientation α in which P is rotated and translated while keeping the contact with the base fingers. We refer to this motion as the sliding of P at orientation α. Clearly it is possible to slide P at orientation α in either clockwise or counterclockwise directions. As we slide P at orientation α in each direction one of the base fingers will eventually reach a vertex of the polygon, and the pair of edges touched by the base fingers change. Meanwhile, the trace of the edge e forms a surface patch, $s(e_1, e_2, e) \subset \mathbb{R}^2 \times [0, 2\pi)$, in canonical arrangement space, that corresponds to the triple of edges. Clearly, $s(e_1, e_2, e)$ has a

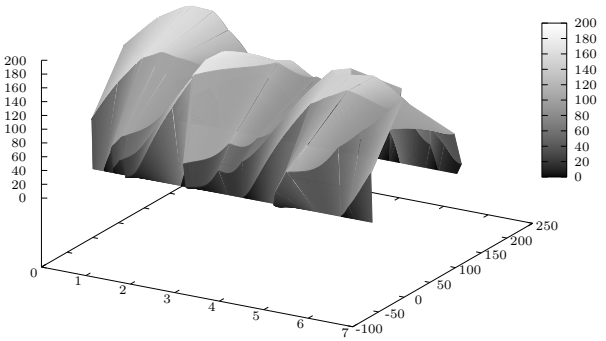


Figure 6.13: The surface patches of $\mathcal{P}$.

constant complexity. Let $\overline{s}(e_1, e_2, e)$ be part of $s(e_1, e_2, e)$ that are induced by all angles α for which the polygon $P[\alpha]$ is below the edge e along the y-axis. The surface patch $\overline{s}(e_1, e_2, e)$ has a constant complexity as well. If the triple (e_1, e_2, e) of edges is triangular then we say that $\overline{s}(e_1, e_2, e)$ is a triangular surface patch.

We define $\mathcal{P}$ in canonical arrangement space as the set of patches $\overline{s}(e_1, e_2, e)$ for all edges e of the polygon P and all pairs e_1 and e_2 of edges touched by the base fingers. In other words, $\mathcal{P}$ is the set of all surface patches that are formed by considering the upper part of $P[\alpha]$ for all angles α for which d is less than the base diameter of $P[\alpha]$. The intersection of the plane $\theta = \alpha$ with $\mathcal{P}$ is the upper part of $P[\alpha]$ where α is an angle for which d is less than the base diameter of $P[\alpha]$. Every surface patch in $\mathcal{P}$ corresponds to a set of three-finger arrangements whose finger placements are on a unique triple of edges of P two of which are touched by the base fingers and the other one is touched by the third finger. Figure 6.13 displays $\mathcal{P}$ for the convex polygon shown in Figure 6.14. In Figure 6.14 the surface patches of $\mathcal{P}$ are projected to (x, θ)-plane. The horizontal gray lines display the orientations in which the pair of edges touched by the base fingers change. The surface patches of $\mathcal{P}$ form a number of connected surfaces that are bounded by critical angles along the θ-axis. Since there are $O(n\beta)$ triples of edges, there are $O(n\beta)$ surface patches in $\mathcal{P}$.

Consider an angle α for which $P[\alpha]$ is defined. The point (p, α) in canonical arrangement space corresponds to a valid canonical arrangement provided that p is not inside $P[\alpha]$.

Consider two points (p, α) and (p, β) in canonical arrangement space that correspond to valid canonical arrangements. Here we define *visibility* between two such three-finger arrangements. In this Chapter and Chapter 7 the visibility is defined only along the θ-axis. In other words, we define visibility between two points in the canon-

ical arrangement space only when the line connecting the two points is parallel to the θ-axis (and thus the same x- and y-coordinates). There are two different line segments that connect (p, α) and (p, β), which correspond to positive and negative directions along the θ-axis. We define (p, β) to be visible from (p, α) if and only if at least one of the two line segments that connect the two points intersect no surface patches of $\mathcal{P}$. According to the definition, when (p, β) and (p, α) are visible from each other, they are reachable from each other by rotation and translation; thus, they are either both caging or both non-caging. In fact, the visibility condition is a sufficient condition for the two three-finger arrangements to be both caging or both non-caging but it is not a necessary condition.

Lemma 6.3.5. *No surface patch of $\mathcal{P}$ has local maxima along the θ-axis in the interior for a fixed x-coordinate.*

Proof. For the sake of contradiction, assume that a surface patch of $\mathcal{P}$ has a local maximum in the interior at orientation α along the θ-axis for a fixed x-coordinate. Considering the polygon fixed and sliding the three fingers as a rigid body along the edges touched by the base fingers, the locus of the third finger follows an elliptical curve. (Consider Subsection 6.1.2 to see why the locus is an elliptical curve.) The elliptical curve intersects $P[\alpha]$ in the interior of an edge at the placement of the third finger. Thus, the third finger does not remain outside the polygon in both directions, and thus at least in one direction, the third finger penetrates $P[\alpha]$. However, if a surface patch of $\mathcal{P}$ has a local maximum in the interior at orientation α along the θ-axis for a fixed x-coordinate, then it is possible to slide the canonical placement of P at orientation α in both directions for small enough orientations. Since the third finger is already on the boundary of the polygon, when we slide the polygon along the base fingers, in both directions the polygon loses the contact with the third finger. □

6.3.3 Triangular borders

The set $\mathcal{P}$ of surface patches consists of a number of triangular and non-triangular surface patches. The set of triangular surface patches forms a number of connected components. *Triangular borders* are the outer boundary of these connected components. In other words, triangular borders are the common boundary between the triangular surface patches and non-triangular surface patches, and also the boundary between triangular surface patches and critical angles. Every canonical arrangement that corresponds to a point on the triangular borders is a non-caging arrangement.

In Figure 6.14 the surface patches of $\mathcal{P}$ are projected to (x, θ)-plane, in which the θ-axis is the vertical axis and the x-axis is the horizontal axis. The triangular borders are displayed in bold and the loci of vertices are displayed in gray; the distance between the two dotted vertical lines equals the distance between the base fingers. As it is displayed there are two connected components enclosed by triangular borders. (Recall that the θ-axis is circular.)

Each plane perpendicular to θ-axis either intersects the triangular borders in two disjoint sets. Each set of intersected features is either a single feature or a connected set

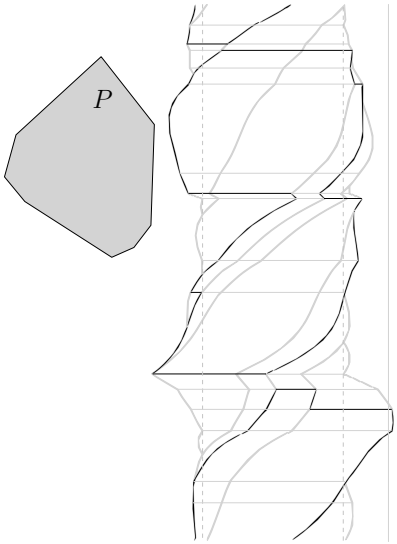


Figure 6.14: See the text explanation in Subsection 6.3.3.

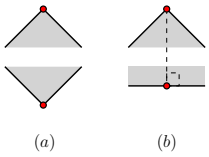


Figure 6.15: Two-finger equilibrium grasps of convex polygons.

of features of P. When we consider x-axis, the features of one of the sets is at the left side of the other set (i.e. their corresponding x-coordinates are smaller than the other set). By considering all planes perpendicular to θ-axis, we decompose the triangular borders into two sets: left borders and right borders. The set of triangular borders is the union of the two sets. The set left borders (right borders) is a collection of sets of connected curves.

6.3.4 Two-finger equilibrium grasps

In this subsection we explain about two-finger equilibrium grasps of convex polygons and their representation in canonical arrangement space. For convex polygons there are only two types of two-finger equilibrium grasps, which are displayed in Figure 6.15. Due to the general position assumption, two-finger equilibrium grasps involve the third finger and one of the base fingers.

Consider a two-finger equilibrium grasp of P at placement q with orientation α.

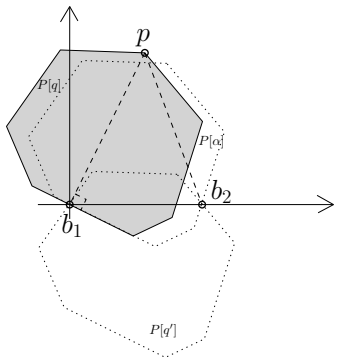


Figure 6.16: Illustration of Lemma 6.3.6.

We show that if $P[\alpha]$ is defined, the equilibrium grasp and its canonical arrangement are reachable from each other by translation, and thus they are non-caging because two-finger equilibrium grasp is non-caging. We state this fact in form of a lemma.

Lemma 6.3.6. *Every two-finger equilibrium grasp and its canonical arrangement are reachable from each other, and thus they are non-caging.*

Proof. Consider a two-finger equilibrium grasp of P in which the third finger and one of the base fingers are involved. Without loss of generality assume the involved base finger is b_1. See Figure 6.16, in which the third finger at p and b_1 form a two-finger equilibrium grasp of $P[q]$. The other base finger b_2 is outside the polygon. As we translate P at q while keeping its contact with b_1, the third finger never intersects or penetrates the translating polygon, until b_2 is placed on the boundary of the translating polygon. There are two possible translated copies of $P[q]$ such that both b_1 and b_2 are on their boundaries, namely $P[q']$ and $P[\alpha]$. Although only $P[\alpha]$ is the canonical arrangement of the two-finger equilibrium grasp, both $P[q']$ and $P[\alpha]$ are reachable from the two-finger equilibrium grasp $P[q]$ and thus both are non-caging. □

According to Lemma 6.3.6, since every two-finger equilibrium grasp is reachable from its canonical arrangement, we can represent the two-finger equilibrium grasps in canonical arrangement space by their canonical arrangements. Since for every edge there is at most one vertex with which a two-finger equilibrium grasp can be formed, there are at most n pairs consisting of a vertex and an edge that define a two-finger equilibrium grasp. Each pair of an edge and a vertex that defines a two-finger equilibrium grasp induces four curves in canonical arrangement space based on the finger that is at the vertex and the base finger that is involved. We refer to these curves as the two-finger equilibrium curves each of which has a constant complexity. (In contrast to Chapter 5 in which the equilibrium curves were defined in a two-dimensional space, here we redefine the two-finger equilibrium curves to live in canonical arrangement space.) Similarly there are at most n pairs consisting of two vertices that define a two-

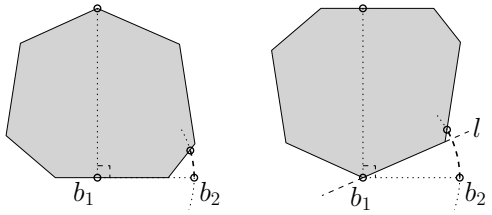


Figure 6.17: Illustration of Lemma 6.3.7.

finger equilibrium grasp. Two vertices that define a two-finger equilibrium grasp are necessarily antipodal.

The two-finger equilibrium curves in canonical arrangement space can be classified into two groups, such that in each group the curves are circular arcs centered around one of the base fingers in the projection to (x, y)-plane. (See Subsection 5.2.1 for more information about the types of equilibrium curves when they are projected to (x, y)-plane.) The two-finger equilibrium curves of each group involve the same base finger either b_1 or b_2.

Two-finger equilibrium grasps that are part of the caging boundary are enclosing arrangements. For this reason, we ignore two-finger equilibrium grasps that are not enclosing arrangements. Consider the canonical arrangements of enclosing arrangements. The third finger for all such canonical arrangements are vertically above triangular surface patches. Therefore, we only consider the canonical representation of two-finger equilibrium grasps that are above the triangular surface patches. For this reason, one endpoint of a two-finger equilibrium curve segment that is on the caging boundary is a subset of a two-finger equilibrium curve that starts/ends at a point vertically above or exactly on triangular borders. According to the explanation we have the following lemma.

Lemma 6.3.7. *A two-finger equilibrium curve segment that is on the caging boundary is a subset of a two-finger equilibrium curve that starts/ends at a point on the planes $x = 0$ or $x = d$ in canonical arrangement space, and a point vertically above or exactly on triangular borders.*

Proof. The important two-finger equilibrium grasps are the grasps whose canonical arrangements are enclosing, otherwise they cannot be on the boundary of the caging set. (Recall that an enclosing arrangement (q, c) is a canonical arrangement that vertically encloses $P[q]$.) For two-finger equilibrium curves in which b_1 is involved we show that one end-point is at the plane $x = 0$. For two-finger equilibrium curves in which b_2 is involved we show that one end-point is at the plane $x = d$ and because of similarity we consider only the two-finger equilibrium grasps in which b_1 is involved. Based on whether one of the base fingers or the third finger is at a vertex, there are two cases to consider. Consider Figure 6.17. For both cases, when the x-coordinate of the third finger is $x = 0$ then the line connecting the third finger and b_1 is perpendicular to the line connecting b_1 and b_2. The line connecting b_1 to b_2 in both cases does not intersect

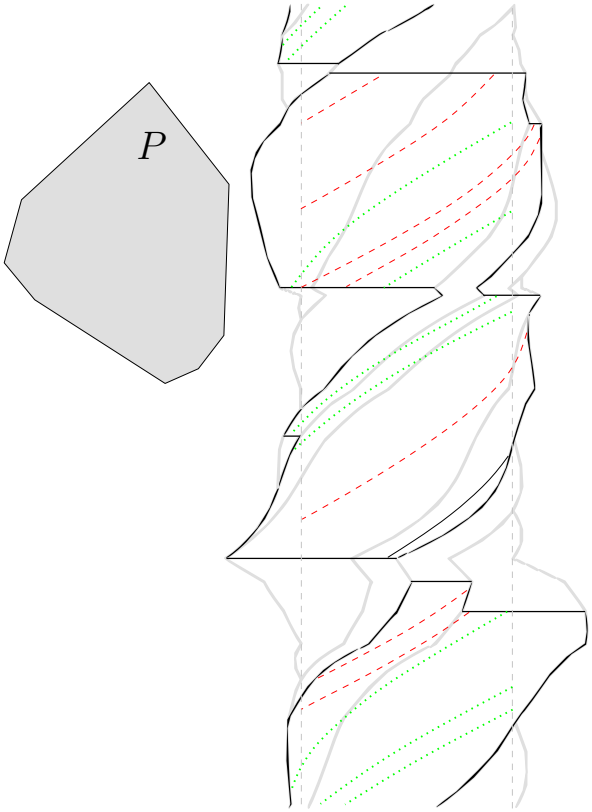


Figure 6.18: See the text explanation in Subsection 6.3.4.

the polygon. (Recall that when a canonical arrangement is an enclosing arrangement the polygon intersects the line connecting the base fingers.) Thus, the three-finger arrangements of the two-finger equilibrium grasps of this group whose x-coordinate is less than zero are non-enclosing arrangements. □

In Figure 6.18 the two-finger equilibrium curves are projected to (θ, x)-plane and are displayed with dashed and dotted curves. The dashed curves correspond to two-finger equilibrium curves that involve b_1. The dotted curves correspond to two-finger equilibrium curves that involve b_2.

6.4 Conclusion

In this chapter we have provided several geometric properties of three-finger caging arrangements of convex polygons. These results will be used in the next three chapters

7, 8, and 9. Generalizing the results to non-convex polygons is an interesting next step. In particular, we have proven in this chapter that for a three-finger caging arrangement of a convex polygon if we squeeze one of the fingers along a perpendicular line toward the line connecting the two other fingers, the resulting three-finger arrangements remain caging. It will be interesting to know whether three-finger caging arrangements of non-convex polygons have similar properties. In particular, we would like to know for a caging arrangement if we squeeze or stretch one of the fingers the resulting three-finger arrangements remain caging. Using the aforementioned squeezing fact for three-finger caging of convex polygons we proved that three-finger caging arrangements are related to three-finger immobilizing grasps of convex polygons. It is interesting to see whether three-finger caging arrangements are related to three-finger immobilizing grasps of non-convex polygons.

Chapter 7

Caging Convex Polygons with Three Fingers: Complexity

7.1 Introduction

In this chapter we consider the worst-case complexity of the caging set for convex polygons with three fingers. A convex polygon with n edges and a placement of two fingers—referred to as the *base fingers*—are given. The caging set is the two-dimensional set of all possible placements of the third finger that together with the base fingers cage the convex polygon.

We in Chapter 5 and Erickson et al. [2007] have proposed algorithms for robotic systems with two degrees of freedom that report the entire solution set for the caging set. Erickson et al. [2007] provided the first algorithm for the exact computation of the caging set of convex polygons, running in $O(n^6)$ time. In their paper the base fingers were assumed to be placed along the boundary of the polygon. They also established an upper bound of $O(n^6)$ on the worst-case complexity of the caging set of convex polygons, where the caging set was shown to be the visible scene of $O(n^3)$ constant-complexity surfaces in a three-dimensional space. We in Chapter 5 have proposed another algorithm generalizing the previous results to compute the caging set of arbitrary polygons for *any* given placement of the base fingers, that runs in $O(n^6 \log^2 n)$ time. We established the same $O(n^6)$ upper bound on the worst-case complexity of the caging set of convex and non-convex polygons, where the caging set was shown to be a subset of the arrangement of $O(n^3)$ constant-complexity 2D curves defined by equilibrium grasps. However, in both cases the mentioned upper bound on the worst-case complexity of the caging set was due to the proposed algorithms, and it remained an open problem to establish a better upper bound for convex or non-convex polygons. In this chapter, we tackle the problem for convex polygons. We prove that the worst-case complexity of the caging set of convex polygons is $O(\lambda_{10}(n^3))$, which significantly improves the already known upper bound of $O(n^6)$, as $\lambda_{10}(n^3)$ is known [Sharir and

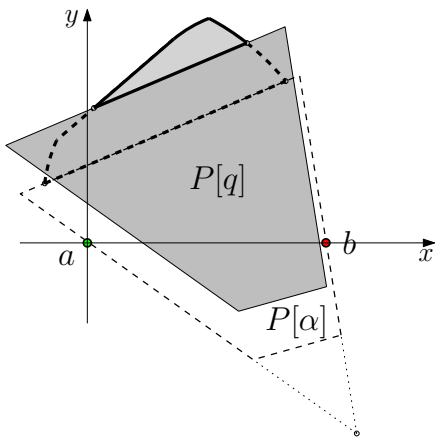


Figure 7.1: $P[q]$ and $P[\alpha]$ are rigid translates. The caging set of $P[\alpha]$ is a superset of the caging set of $P[q]$.

Agarwal, 1996] to be $O(n^3 \log^* n)$[1] and thus very close to $O(n^3)$. To establish the upper bound, firstly we have narrowed down the types of surfaces introduced by Erickson et al. [2007] that can play a role in the caging set complexity. Secondly, we have formulated a new way to compute the caging set using those surfaces. In addition, we develop an efficient algorithm to compute the caging set in $O(\lambda_s(n^3) \log n)$ time using a divide-and-conquer technique.

We have used several notions, concepts, and results of Chapter 6. This chapter consists of three main parts. We consider escaping by translation and find a necessary and sufficient condition for which a three-finger arrangement can escape by translation in Section 7.3. Then we study the relationship between three-finger arrangements from which it is impossible for the given polygon to escape by translation and caging arrangements. In Section 7.4 we define a number of vertical walls using the canonical arrangements of two-finger equilibrium grasps and non-triangular borders. Then we prove that a three-finger arrangement is caging if and only if none of those walls are visible in canonical arrangement space. In Section 7.5 we prove the complexity bound of the caging set and give an efficient algorithm to compute it based on the visibility concepts in canonical arrangement space.

7.2 Definitions and assumptions

In this section we repeat some of the definitions and assumptions we introduced in Chapter 6.

[1] $\log^* n = \min\{i \geq 0 : \log^{(i)} n \leq 1\}$, where i is a nonnegative integer, and $\log^{(i)} n$ is the logarithm function applied i times in succession, starting with argument n.

In the whole chapter, we consider placements of a given convex polygon P that touch both base fingers. The reason is that, the caging set of any other placement q of P is a subset of the caging set of a unique associated placement q' of P that touches both base fingers and are rigid translates. (See Chapter 6 Theorem 6.3.4.) See Figure 7.1, in which the caging set of $P[q]$ is the area surrounded by bold solid curves and the caging set of $P[\alpha]$ is the union of both the area surrounded by bold solid curves and the area surrounded by bold dashed curves. Besides, given the caging set of P at the placement q', the caging set of P at the placement q can be computed easily, which is explained in the next Chapter 8.

The polygon $P[\alpha]$ is the polygon P rotated by the angle α around its reference point for which both base fingers are in contact with it and the extensions of the edges touched by the base fingers intersect each other below the x-axis. We refer to $P[\alpha]$ as the *canonical placement* of any other placement of P with orientation α. In Figure 7.1, $P[\alpha]$ is the canonical placement of $P[q]$. The polygon $P[\alpha]$ is not defined for orientations α which the base-diameter of a translated copy of P with orientation α is more than the distance between the base fingers. (See Section 6.1 for the definition of base-diameter.)

Every three-finger arrangement in which the base fingers are at b_1 and b_2 and the polygon is at $P[\alpha]$ can be specified by the three parameters $((x, y), \alpha) \in \mathbb{R}^2 \times [0, 2\pi)$. The parameter α specifies the orientation of the polygon and the parameters x and y specify the location of the third finger in the plane $\theta = \alpha$. We refer to the space $\mathbb{R}^2 \times [0, 2\pi)$ of all such three-finger arrangements as the *canonical arrangement space*.

A triple of edges of P is called *triangular* if the supporting lines of the edges form a triangle that encloses P, and is called *non-triangular* otherwise. If a canonical arrangement (q, α) vertically encloses $P[\alpha]$ then we say that (q, α) is an enclosing arrangement.

Let $C(\alpha)$ be the set of all placements of the third finger that together with the base fingers form a caging arrangement of $P[\alpha]$. The boundary of $C(\alpha)$ consists of two x-monotone chain of curves of which the lower one is a subset of the boundary of $P[\alpha]$. Let $K(\alpha)$ be the upper part of the curves on the boundary of $C(\alpha)$.

7.3 Escaping by translation

In this section we first present some definitions and a result by Erickson et al. [2007] to identify all placements of the third finger that prevent $P[\alpha]$ from escaping by pure translation; these placements form a two-dimensional region. Then we investigate the relationship between the boundary of this region and the boundary of the caging set.

Let the convex polygon $Q[\alpha, b_1]$ be the union of the set of all translated copies of $P[\alpha]$ touching the base finger b_1. The polygon $Q[\alpha, b_1]$ has twice the number of edges of $P[\alpha]$. Every edge of $P[\alpha]$ is parallel to exactly two edges of $Q[\alpha, b_1]$. Similarly let $Q[\alpha, b_2]$ be the union of the set of all translated copies of $P[\alpha]$ touching b_2.

Let $X[\alpha]$ be the points inside the intersection of $Q[\alpha, b_1]$ and $Q[\alpha, b_2]$ that are above $P[\alpha]$. Figure 7.2 shows $Q[\alpha, b_1]$ and $Q[\alpha, b_2]$; the polygon is displayed in dark-gray and $X[\alpha]$ is displayed in light-gray.

Erickson et al. [2007] have proven that the set $X[\alpha]$ is the set of all placements of the third finger that prevents $P[\alpha]$ from escaping by pure translation.

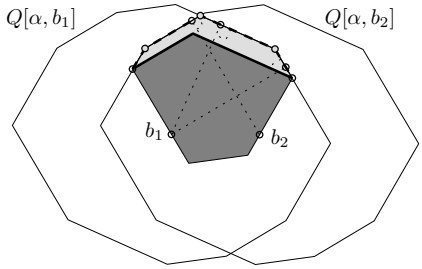


Figure 7.2: Illustration of Lemma 7.3.2.

Lemma 7.3.1. *[Erickson et al., 2007] The polygon $P[\alpha]$ can escape by pure translation if and only if the third finger placement is not within $X[\alpha]$.*

By Lemma 7.3.1, each edge of $P[\alpha]$ on the boundary of $X[\alpha]$ forms a triangular triple of edges together with the edges touched by the base fingers. Let $\partial X^u[\alpha]$ be the upper-boundary of $X[\alpha]$. The set of edges of $\partial X^u[\alpha]$ consists of two continuous sets of edges: one set of edges belonging to the boundary of $Q[\alpha, b_1]$ and one set of edges belonging to the boundary of $Q[\alpha, b_2]$.

Clearly, $X[\alpha]$ is a superset of $C(\alpha)$. The following lemma proves that if a point on $\partial X^u[\alpha]$ is on the caging boundary then either the corresponding three-finger arrangement corresponds to a two-finger equilibrium grasp or the corresponding three-finger arrangement is on a triangular border. The result of this lemma narrows down the surfaces that should be considered to compute the caging set as a visibility problem.

Lemma 7.3.2. *A point on $\partial X^u[\alpha]$ that belongs to $K(\alpha)$ either corresponds to a two-finger equilibrium grasp or it is on a triangular border.*

Proof. Consider orientation α' which is initially equal to orientation α. Consider $Q[\alpha', b_1]$ and $Q[\alpha', b_2]$ and $P[\alpha']$ as functions of α'. As we vary α', $Q[\alpha', b_1]$ rotates around b_1 and $Q[\alpha', b_2]$ rotates around b_2 at the same speed. Consider a fixed point p in the *interior* of an edge e on the boundary of $Q[\alpha, b_1]$. Consider another point p' at the same place of p but on $Q[\alpha', b_1]$. As we vary α', the trace of the point p' follows a circular arc around b_1 until b_1 or b_2 hits a vertex. (The circular arc passes at p.) Therefore, if the line segment that connects b_1 to p is not perpendicular to e, the mentioned circular arc penetrates $Q[\alpha', b_1]$ exactly in one direction. That means, if we slide P along b_1 and b_2 in the other direction, the point p loses contact with $Q[\alpha', b_1]$ and remains outside $Q[\alpha', b_1]$.

However, if the point p is in the interior of $K(\alpha)$ (thus not on its endpoints) it remains so for sufficiently small changes of the orientation α. That means, for sufficiently small changes of the orientation α, the point p should either remain inside $X[\alpha]$ or remain on its boundary. Therefore, the only points p in the interior of an edge of $\partial X^u[\alpha]$ that can be also on $K(\alpha)$ are the points for which the line segment connecting them to b_1 or b_2 is orthogonal to their supporting edge on $Q[\alpha, b_1]$ or $Q[\alpha, b_2]$. Such placements of the third finger correspond to two-finger equilibrium grasps in which one finger is

at a vertex and the other one at an edge such that the line connecting the two fingers is perpendicular to the edge.

There are two groups of points on the boundary of $Q[\alpha, b_1]$ or $Q[\alpha, b_2]$ that are not part of the aforementioned category. One group consists of the placements of the third finger at vertices of $\partial X^u[\alpha]$ that are also vertices of $Q[\alpha, b_1]$ or $Q[\alpha, b_2]$. The other group consists of the two end points of $\partial X^u[\alpha]$.

The placements of the third finger at vertices of $\partial X^u[\alpha]$ that are also vertices of $Q[\alpha, b_1]$ or $Q[\alpha, b_2]$ can also be on $K(\alpha)$. Such placements of the third finger also correspond to two-finger equilibrium grasps in which both fingers are at two antipodal vertices.

The two end points of $\partial X^u[\alpha]$ can also be on $K(\alpha)$, which are on triangular borders. The reason is that, all edges of P on the boundary of $X[\alpha]$ form triangular triple of edges together with the two edges touched by the base fingers. □

In Figure 7.2, the points on $\partial X^u[\alpha]$ that can possibly be on $K(\alpha)$ are marked with small gray circles.

7.4 Caging and non-caging in canonical arrangement space

In this section we prove that it is possible to compute $C(\alpha)$ for a given orientation α by using the surface patches of $\mathcal{P}$, and two types of non-caging arrangements in canonical arrangement space: three-finger arrangements on triangular borders and canonical arrangements of two-finger equilibrium grasps.

If $p \in C(\alpha)$ then the three-finger arrangement (p, α) in canonical arrangement space is only visible from other caging arrangements. In other words, if (p, β) is a non-caging arrangement and it is visible from (p, α) then $p \notin C(\alpha)$. In this section, we formulate a way to identify all points in the plane $\theta = \alpha$ that are visible from non-caging arrangements. In Chapter 6, we have introduced two groups of non-caging arrangements: three-finger arrangements on triangular-borders and two-finger equilibrium grasps. In this section, we define vertical walls on three-finger arrangements that are on triangular borders, and also on canonical arrangements of two-finger equilibrium grasps. These walls represent a set of non-caging arrangements, such that if a point on the plane $\theta = \alpha$ is visible from a point on one of these walls then that point represents a non-caging arrangement. In this section, we prove the important fact that if an enclosing arrangement is not visible from any of the mentioned vertical walls, then it is a caging arrangement. Therefore, these walls and the surface patches of $\mathcal{P}$ provide enough information to compute $C(\alpha)$.

Consider a non-caging arrangement $((x, y'), \beta)$ in canonical arrangement space. Consider another three-finger grasp $((x, y), \beta)$ where $y > y'$; thus the point $((x, y), \beta)$ is vertically above $((x, y'), \beta)$ in canonical arrangement space, and $((x, y), \beta)$ is a non-caging arrangement as well. Consider another enclosing arrangement $((x, y), \alpha)$ in canonical arrangement space. If $((x, y), \alpha)$ is visible from $((x, y), \beta)$, then since $((x, y), \beta)$ is non-caging, $((x, y), \alpha)$ is non-caging as well. To put it in words, if a point vertically above a non-caging arrangement is visible from a given three-finger arrangement

(p, α) then (p, α) is non-caging. Let us consider two specific classes of non-caging arrangements and identify all points vertically above those, whose visibility is important to determine non-caging arrangements.

The set of all three-finger arrangements $((x, y), \beta)$ in canonical arrangement space in which $y > y'$ and $((x, y'), \beta)$ is on a triangular border, defines a number of vertical walls, to which we refer as the *triangular-border walls*. No point in $C(\alpha)$ in the plane $\theta = \alpha$ can be visible from a point on a triangular-border wall. Therefore, the set of points in the plane $\theta = \alpha$ that are visible from no point on the triangular-border walls is a superset of $C(\alpha)$.

Recall that the canonical arrangements of the two-finger equilibrium grasps form a number of curves in canonical arrangement space. We define the upward vertical walls on all points on these curves, to which we refer as the *two-finger equilibrium walls*. Similar to the triangular-border walls, if (p, α) is visible from a point on a two-finger equilibrium wall then (p, α) is non-caging. The two-finger equilibrium walls intersect no surface patches of $\mathcal{P}$ by Lemma 6.3.6.

We define the *non-caging walls* as the union of the set of triangular-border walls and two-finger equilibrium walls. Let $V(\alpha)$ be the set of points in the plane $\theta = \alpha$ that correspond to enclosing arrangements of $P[\alpha]$ and are visible from no point on the non-caging walls. We already know that $C(\alpha)$ is a subset of $V(\alpha)$. We prove that $C(\alpha)$ is equal to $V(\alpha)$.

First we provide a lemma which we use to prove the main result of this section. Consider the intersection of a surface patch of $\mathcal{P}$ with a plane $\boldsymbol{x} = x$ perpendicular to x-axis. Their intersection forms a two-dimensional curve in the space (θ, y), which is monotone along θ-axis. Every point $((x, y), \theta)$ at some orientation θ on that curve (in canonical arrangement space) corresponds to a placement (x, y) of the third finger on the boundary of $P[\theta]$, which with the two base fingers forms a three-finger arrangement. The following lemma states that the local minima of y-coordinates of that curve correspond to immobilizing grasps. Recall that every immobilizing grasp is a caging arrangement.

Lemma 7.4.1. *The local minima of y-coordinates in the interior of the intersection of a triangular surface patch of $\mathcal{P}$ with a plane $\boldsymbol{x} = x$ are immobilizing grasps.*

Proof. Consider such a local minimum $((x, y), \alpha)$. Since the third finger is on the boundary of $P[\alpha]$ and the triple of edges are triangular, then it is not possible to escape by translation. On the other hand, it is not possible to rotate the polygon around any point. Otherwise, it would be possible to rotate the polygon and therefore make some space to squeeze the third finger, that contradicts the local minimum assumption. The rest of reasoning is the same as the reasoning provided in Theorem 6.2.13. Therefore, the grasp is immobilizing. □

The following lemma states that $C(\alpha)$ is equal to $V(\alpha)$. This result will be the foundation of our approach to establish the complexity bound. We already know that no point in $C(\alpha)$ is visible from a two-finger equilibrium wall or a triangular-border wall. We prove that every non-caging enclosing arrangement $((x, y), \alpha)$ is visible from a non-caging wall. If $((x, y), \alpha)$ is visible from a non-caging wall then $((x, y'), \alpha)$ with $y' > y$

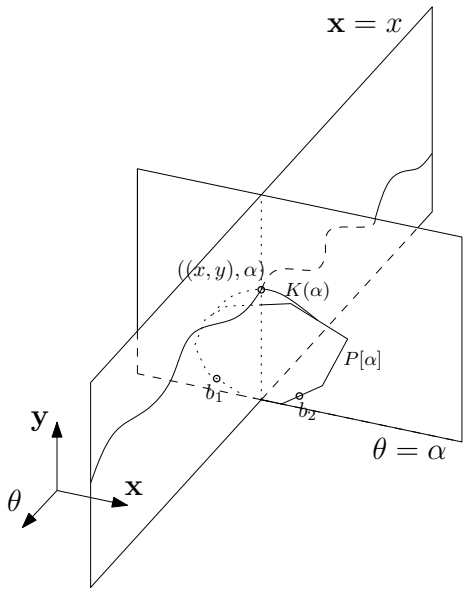


Figure 7.3: Illustration of Lemma 7.4.2.

is also visible from that wall. Therefore, it suffices to consider the upper boundary of $C(\alpha)$, and so to prove the claim for the points on $K(\alpha)$.

Lemma 7.4.2. *For every point (x, y) on $K(\alpha)$, the point $((x, y), \alpha)$ is visible from a point on a non-caging wall.*

Proof. If the point $((x, y), \alpha)$ corresponds to a two-finger equilibrium grasp then $((x, y), \alpha)$ is visible from a two-finger equilibrium wall by Lemma 6.3.6, and the lemma follows. Assume that $((x, y), \alpha)$ does not correspond to a two-finger equilibrium grasp. As a result, the point $((x, y), \alpha)$ must correspond to a three-finger equilibrium grasp by Theorem 5.2.1. Consider the intersection of the plane $\boldsymbol{x} = x$ with the surface patches of $\mathcal{P}$, which is two-dimensional curve in space (θ, y). The number of points on this curve that correspond to three-finger equilibrium grasps is finite. Therefore, assume that $((x, y), \alpha)$ is a point in canonical arrangement space such that it corresponds to a three-finger equilibrium grasp, is not visible from a two-finger equilibrium wall or a triangular-border wall, and has a y-coordinate that is minimal among all such three-finger equilibrium grasps. Consider Figure 7.3 which demonstrates the axes, the polygon, and the point $((x, y), \alpha)$ in canonical arrangement space.

Since (x, y) is on $K(\alpha)$ and thus non-caging, the canonical placement of P at orientation α can escape by first sliding along the base fingers and then by translating, according to Erickson et al. [2007]. Let $((x, y), \beta)$ be a canonical arrangement at which it is possible for the canonical placement of P at orientation β to escape by translation. Similar to $((x, y), \alpha)$ no non-caging wall is visible from $((x, y), \beta)$. Since the polygon

can escape by translation through the three-finger arrangement $((x,y),\beta)$, (x,y) is neither a point inside $C(\beta)$ nor a point inside $X[\beta]$. If (x,y) is on $K(\beta)$, then it corresponds to a two-finger equilibrium grasp or it is on a triangular border by Lemma 7.3.2. Therefore, assume that (x,y) is outside $C(\beta)$. Let Q be the set of all points in canonical arrangement space whose corresponding canonical arrangements are reachable from $((x,y),\beta)$ by sliding the polygon on the base fingers while allowing the third finger to be squeezed. Every non-caging wall visible from a point in Q is also visible from the points $((x,y),\beta)$ and $((x,y),\alpha)$. The set Q contains a number of local minima along the θ-axis with respect to the y-coordinate. Since $((x,y),\beta)$ is not visible from a triangular-border wall the local minima of Q are immobilizing grasps by Lemma 7.4.1. Let $((x,y_m),\alpha_m)$ be one of those immobilizing grasps and consider $(x,y') \in K(\alpha_m)$. We have $((x,y'),\alpha_m) \in Q$ and thus all non-caging walls that are visible from $((x,y'),\alpha_m)$ are also visible from $((x,y),\alpha)$. If $((x,y'),\alpha_m)$ corresponds to a two-finger equilibrium grasp the lemma follows. Otherwise, the three-finger arrangement $((x,y'),\alpha_m)$ corresponds to a three-finger equilibrium grasp by Theorem 5.2.1, for which $y' < y$. The existence of the three-finger arrangement $((x,y'),\alpha_m)$ contradicts the assumption. □

Corollary 7.4.3. *$K(\alpha)$ equals the lower boundary of the part of the non-caging walls visible (i.e. not hidden behind patches of $\mathcal{P}$) from the plane $\boldsymbol{\theta} = \alpha$, projected onto $\boldsymbol{\theta} = \alpha$.*

7.5 Complexity and computation of the caging set

In this section we prove that the complexity of the caging set is close to $O(n^3)$ in the worst case. We also propose an algorithm that efficiently computes the boundary of the caging set in a time that is close to $O(n^3 \log(n))$ in the worst case. The main fact we prove is that the complexity of the visible part of a surface patch of $\mathcal{P}$ not hindered by the non-caging walls is constant. This fact allows us to establish an upper bound on the complexity of the caging set and also to obtain a solution to compute the caging set efficiently.

Project the visible parts of non-caging walls not obstructed by the surface patches of $\mathcal{P}$ in the clockwise viewing direction onto the the plane $\boldsymbol{\theta} = \alpha$. Define $N^+(\alpha)$ to be the lower boundary of that projection. Define $N^-(\alpha)$ similarly as the lower boundary of the projection onto the the plane $\boldsymbol{\theta} = \alpha$ in the counterclockwise viewing direction. According to Corollary 7.4.3, $K(\alpha)$ is the lower boundary of the non-caging walls not obstructed by the patches of $\mathcal{P}$ projected onto the plane $\boldsymbol{\theta} = \alpha$. Clearly $K(\alpha)$ is the minimum of $N^+(\alpha)$ and $N^-(\alpha)$.

Here, however, we formulate a slightly different (but equivalent) way to compute $K(\alpha)$. For each direction we perform two projections.

- We project the visible parts of surface patches of $\mathcal{P}$ not obstructed by the non-caging walls to the plane $\boldsymbol{\theta} = \alpha$, and we compute their upper-boundary.
- Without considering the surface patches of $\mathcal{P}$, we project the non-caging walls to the plane $\boldsymbol{\theta} = \alpha$ and compute their lower boundary.

Let $V^+(\alpha)$ be the maximum of the two resulting boundaries in the clockwise viewing direction. Define $V^-(\alpha)$ similarly for the counterclockwise viewing direction. In Lemma 7.5.1 we prove that $V^+(\alpha)$ is equal to $N^+(\alpha)$ and $N^-(\alpha)$ is equal to $N^-(\alpha)$. Therefore, $K(\alpha)$ is the minimum of $V^+(\alpha)$ and $V^-(\alpha)$.

Lemma 7.5.1. *$V^+(\alpha)$ is equal to $N^+(\alpha)$ and $V^-(\alpha)$ is equal to $N^-(\alpha)$*

Proof. Consider the plane $\mathbf{x} = x$ intersecting the polygon $P[\alpha]$. There are two cases; either there is a surface patch and a non-caging wall such that the intersection of their projections to the plane $\boldsymbol{\theta} = \alpha$ is not empty and the plane $\mathbf{x} = x$ intersects it; or there is no such a surface patch and a non-caging wall. Consider the first case. Because of the way we have defined the non-caging walls there is only one point $p = (x, y)$ on the boundary of visible non-caging walls and visible surface patches at this x-coordinate on the plane $\boldsymbol{\theta} = \alpha$. The point p is both on the upper-boundary of visible surface patches not obstructed by non-caging walls, and also on the lower-boundary of visible non-caging walls not obstructed by surface patches of $\mathcal{P}$. Moreover, the lower boundary of the non-caging walls projected to the plane $\boldsymbol{\theta} = \alpha$ is lower than y at x.

Consider the second case. In this case there is a gap between the lower-boundary of the non-caging walls and the upper-boundary of surface patched projected to the plane $\boldsymbol{\theta} = \alpha$ in which the former is above the latter. The projection of no surface patch and no non-caging wall intersects this gap. Therefore, in this case the lower boundary of visible non-caging walls not obstructed by surface patches is exactly the same as the lower boundary of non-caging walls both projected to the plane $\boldsymbol{\theta} = \alpha$.

As we consider these two cases, it is clear that $V^+(\alpha)$ is equal to $N^+(\alpha)$. We can similarly prove that $V^-(\alpha)$ is equal to $N^-(\alpha)$. □

The main reason we favor the second formulation over the first one is that we can establish an upper bound on the complexity of the caging set as we will see later. Moreover, we suggest an efficient algorithm to compute the caging set by using this formulation.

We prove in Lemmas 7.5.5 and 7.5.6 that the complexity of the visible part of each surface patch not obstructed by the non-caging walls is constant. Before that, we mention three results that can be easily verified.

Lemma 7.5.2. *The two-finger equilibrium walls involving the same base finger do not intersect each other.*

Proof. Assume that the polygon is fixed and we translate the fingers. Consider a two-finger equilibrium grasp. As we slide the involved base finger on the boundary of the polygon the third finger (as well as each of the base fingers) follows a translated copy of the polygon. Consider Figure 7.4. The polygon itself and its translated copy (which is followed by the third finger) intersect each other only at one point. This point corresponds to a placement of the third finger that together with the involved base finger forms the two-finger equilibrium grasp. There is no other translation of the three-finger arrangement (of the equilibrium grasp) in which both the involved base finger and the third finger are on the boundary of the polygon to form another two-finger

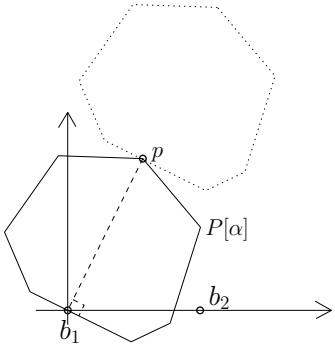


Figure 7.4: Illustration of Lemma 7.5.2.

equilibrium grasp. Clearly if we stretch the third finger (along a line perpendicular to the supporting line of the base fingers) the polygon and its translated copy do not intersect each other any more. □

Observation 7.5.3. *The triangular-border walls do not intersect each other.*

Lemma 7.5.4. *The non-caging walls do not intersect the surface patches of $\mathcal{P}$.*

Proof. The lower boundaries of triangular-border walls lie on the surface patches of $\mathcal{P}$. Thus, the triangular-border walls do not intersect the surface patches. Two-finger equilibrium walls also do not intersect the surface patches of $\mathcal{P}$ by Lemma 6.3.7. Therefore, none of the non-caging walls intersect the surface patches of $\mathcal{P}$. □

First we prove that the visible part of a surface patch not obstructed by triangular-border walls has constant complexity. In Lemma 7.5.6 we consider the two-finger equilibrium walls as well.

Lemma 7.5.5. *The complexity of the visible part of a surface patch of $\mathcal{P}$ not obstructed by triangular-border walls is constant.*

Proof. Since the triangular-border walls are built on the surface patches of $\mathcal{P}$ we can regard them as unbounded in upward and downward directions. The first is true by definition, and the latter can be seen by considering a point which is behind and below a triangular-border wall. Then that point is hindered by the surface patches upon which the triangular-border wall is built.

Recall the sets of left borders and right borders from Subsection 6.3.3. Consider the left borders of triangular borders along x-axis. The left borders are a collection of sets of connected curves. Consider the set of connected curves that is intersected by the plane $\boldsymbol{\theta} = \alpha$. From now on in this lemma we misuse the term left borders to specify this set. (The right borders can be treated similarly.) As we consider the distance from the plane $\boldsymbol{\theta} = \alpha$, the walls built upon left borders can be ordered increasingly. (Two wall with the same θ-coordinates are ordered based on their x-coordinates decreasingly.)

The side of a surface patch that does not face the plane $\theta = \alpha$ is not visible. We consider the set of local maxima of the triangular borders along x-axis and then we consider the sub-sequence of local maxima increasingly. The reason is that a local maximum is completely invisible behind another local maximum whose x-coordinate is larger.

Consider an arbitrary surface patch s. Since the complexity of s is constant it has a constant number of local minima along x-axis. Consider a local minimum of s that is hindered by a triangular-border wall (from the increasing sub-sequence). This hindering wall is the wall that is the closest to the local minimum along θ-axis and is located between the plane $\theta = \alpha$ and the local minimum. Every point of s that its x-coordinate is larger than the local maximum of the hindering triangular-border wall and it is connected by a x-monotone curve (on the surface patch) to the hindered point (i.e. the local minimum point), is not hindered by any other triangular-border wall built upon left borders. Every other point of the surface patch that its x-coordinate is smaller than the local maximum of the triangular-border wall and it is connected by a x-monotone curve to the hindered point (i.e. the local minimum point), is hindered by that triangular-border wall. Every point on s is connected to at least one local minimum with a x-monotone curve. Therefore the effect of a local minimum of a surface patch hindered by a triangular-border wall on the visible part of that surface patch is constant. □

The following lemma is the main lemma we use to provide an upper bound on the complexity of the caging set.

Lemma 7.5.6. *The complexity of the visible part of a surface patch of $\mathcal{P}$ not obstructed by non-caging walls is constant.*

Proof. Consider one side of the plane $\theta = \alpha$ and an arbitrary surface patch of $\mathcal{P}$. Consider the visible part of the surface patch not obstructed by the triangular-border walls. By Lemma 7.5.5 the complexity of this visible part is constant. Therefore, we compute the visible part of the surface patch not obstructed by the triangular-border walls and then remove (or ignore) the triangular-border walls.

Consider the two-finger equilibrium walls that involve the base finger b_1. We show that there is no need to consider part of the surface patch on the right or left side of a two-finger equilibrium wall. The right side of each of the two-finger equilibrium walls resides on triangular-border walls by Lemma 6.3.7. Thus, if part of a surface patch is behind a two-finger equilibrium wall and on its right side, then that part is already hindered by a triangular-border wall. Thus, there is no need to consider the visibility of surface patches on the right side of the two-finger equilibrium walls, as we first consider only parts of surface patches that are not hindered by the triangular-border walls. The other end of the two-finger equilibrium walls all lie on the plane $x = 0$ again by Lemma 6.3.7. This plane divides each surface patch into a constant number of smaller surface patches each completely on one side of the plane.

We increasingly order the two-finger equilibrium walls along θ-axis according to their distance from the plane $\theta = \alpha$. We traverse the two-finger equilibrium walls according to that order and we compute a sub-sequence with decreasing radii. The rea-

son is that, by Lemma 7.5.2 no two-finger equilibrium wall that involve the same base finger intersect each other. Thus, a two-finger equilibrium wall with a larger radius is completely invisible behind a two-finger equilibrium wall with a smaller radius.

As we consider the distance to the line $(0, 0, \theta)$ each surface patch contains a constant number of local maxima. If a local maximum is visible then all points that are monotonically connected to the local maximum and have less distance to the line $(0, 0, \theta)$ are visible too. If a local maximum is not visible, then it is hindered by a number of two-finger equilibrium walls, from which consider the wall (from the decreasing sub-sequence) that is the furthest away from the plane $\boldsymbol{\theta} = \alpha$. Here we say that a time parametrized-path on the surface patch is monotone, if the distances between the points on the path and the line $(0, 0, \theta)$ is either non-decreasing or non-increasing. All points of the surface patch that are monotonically connected to the local maximum (by a path on the surface patch) and have distance less to the line $(0, 0, \theta)$ than the radius of this wall are not hindered by any other two-finger equilibrium wall involving b_1. All points of the surface patch that are monotonically connected (by a path on the surface patch) to the local maximum and have distance larger than the radius of this wall, are hindered. Therefore the effect of a local maximum hindered by a two-finger equilibrium wall on the visible part of the surface patch is constant.

We can similarly argue about the two-finger equilibrium walls that involve the base finger b_2. □

In the following lemma and theorem we provide an upper bound on the worst-case complexity of $K(\alpha)$.

Lemma 7.5.7. *The complexity of $K(\alpha)$ is at most the sum of complexities of $V^+(\alpha)$ and $V^-(\alpha)$.*

Proof. The set $K(\alpha)$ is the minimum of $V^+(\alpha)$ and $V^-(\alpha)$ by Lemma 7.5.1. Let the complexity of $V^+(\alpha)$ be of order $O(f(n))$. Consider the sequence of the breaking points of both $V^+(\alpha)$ and $V^-(\alpha)$ in increasing order with respect to their x-coordinates. Between every two consecutive breaking point, exactly one sub-curve of $V^+(\alpha)$ and exactly one sub-curve of $V^-(\alpha)$ lie within the interval. These two sub-curves intersect each other a constant number of times. Since, the total number of breaking points of both $V^+(\alpha)$ and $V^-(\alpha)$ is of order $O(f(n))$, the total number of breaking points of $K(\alpha)$ is also of order $O(f(n))$. □

A Davenport-Schinzel sequence, $DS(m, s)$-sequence, is a sequence of m symbols in which no two symbols alternate more than s times. The lower boundary of m two-dimensional x-monotone curve segments in which no two curve segments intersect each other more than $s - 2$ times is a $DS(m, s)$-sequence. The maximum length of a $DS(m, s)$-sequence is $\lambda_s(m)$ [Sharir and Agarwal, 1996].

Theorem 7.5.8. *The complexities of both $V^+(\alpha)$ and $V^-(\alpha)$ are of order $O(\lambda_{10}(n^3))$.*

Proof. The degree of the silhouette curves of the visible part of each surface patch is at most four [Erickson et al., 2007]. Therefore each two curves intersect each other at most eight times. The complexity of the upper-boundary of the visible part of each surface

patch projected to the plane $\boldsymbol{\theta} = \alpha$ is $O(\lambda_{10}(n^3))$ by Lemma 7.5.6. The complexity of the lower-boundary of the non-caging walls projected to the plane $\boldsymbol{\theta} = \alpha$ is $O(\lambda_{10}(n^3))$ too. The complexity of the maximum of the two resulting chain of arcs is $O(\lambda_{10}(n^3))$ too. The proof for the last part is the same as the proof explained in Lemma 7.5.7. □

Here we explain a way to compute $K(\alpha)$ in $O(\lambda_{10}(n^3)\log n)$ time. The visible parts of all surface patches can be computed in $O(n^3 \log n)$ time according to Lemmas 7.5.5 and 7.5.6. To compute the upper-boundary of the projected visible parts we use a divide-and-conquer technique. We divide the projected visible parts into two groups and compute the upper-boundary for each group separately. Then we merge the results to compute the final upper-boundary. A simple recursive analysis yealds the upper bound of $O(\lambda_{10}(n^3)\log n)$ on the running time. We use the same technique to compute the lower-boundary for the projected patches of non-caging walls.

Theorem 7.5.9. *$K(\alpha)$ can be computed in $O(\lambda_{10}(n^3)\log n)$ time.*

7.6 Conclusion

We have provided a worst-case bound of almost $O(n^3)$ on the combinatorial complexity of the caging set of a convex polygon with n vertices, and an algorithm with a running time close to $O(n^3 \log n)$ to compute the caging set. Both results present a major improvement over previous results. Our results have been obtained by exploiting a novel formulation of caging in terms of visibility in canonical arrangement space.

The first question that comes to mind is whether the bounds reported here are tight. It is interesting to find examples of convex polygons for which the complexity of the caging set is close to the provided upper bound. Another challenge is to extend the bounds obtained in this chapter to the caging set of non-convex polygons.

Chapter 8

Caging Convex Polygons with Three Fingers: Fixed distance between the base fingers

8.1 Introduction

In the existing works on caging a polygon with three fingers the placements of two fingers, called the base fingers, are given. It is required to find all placements of the third finger such that the three fingers together cage the polygon. The set of placements of the third finger forms a number of two-dimensional regions to which we refer as the *caging set* of the third finger. Erickson et al. [2003] provided the first algorithm for computing all three-finger caging arrangements of convex polygons. In their paper the base fingers were assumed to be placed along the boundary of the polygon and the caging set was computed for the third finger in $O(n^6)$ time. We in Chapter 5 provided the first algorithm for computing all three-finger caging arrangements of arbitrary polygons for *any* given placement of the base fingers with $O(n^6 \log^2 n)$ running time. Careful study of our approach there reveals that given the polygon and the distance of the base fingers it may be possible to compute a data structure within nearly the same running time, that can be queried to report the caging set for *any* given placement of the base fingers more efficiently. In this chapter we show this fact for convex polygons. In particular, when the base fingers are on the boundary of the polygon the query time turns out to be largely proportional to the combinatorial complexity of the reported caging set. Moreover using the same idea we consider another query; for a given placement of the base fingers and a placement of the third finger, we give an algorithm that determines in $O(\log n)$ time whether the resulting three-finger arrangement cages the polygon.

Considering the terms we introduced in Chapter 6 we rephrase the two main questions as follows.

- For a given placement q of P (with respect to the base fingers b_1 and b_2) and a given placement c of the third finger, determine whether $c \in C(q)$.
- For a given placement q of P, determine $K(q)$ (and hence $C(q)$).

In this chapter we have extensively used the notions, concepts, and results of Chapter 6. Therefore, it is crucial to read Chapter 6 before this chapter.

The set $\mathcal{K}$ is defined to be the set of all placements of the third finger that ever, i.e., for some placement q of P, appear on $K(q)$. In Section 8.2 we provide an upper bound on the complexity of $\mathcal{K}$ and an algorithm to compute it. We outline our approach to solving both caging problems in Section 8.3. In Section 8.4 we present algorithms to answer the two different queries. We conclude the chapter with a discussion of future work.

8.2 Complexity and computability of $\mathcal{K}$

In this section we provide an upper bound on the complexity of $\mathcal{K}$ and an algorithm to compute it. We compute $\mathcal{K}$ to develop a data structure we use to answer caging queries in Section 8.4

First we prove that it is possible to decompose the set $\mathcal{K}$ into a set of curve segments each of which is also a curve segment of $\mathcal{A}(\mathcal{E})$. As a result, the complexity of $\mathcal{A}(\mathcal{E})$ is an upper bound on the complexity of $\mathcal{K}$. Then we explain how to determine whether a curve segment of $\mathcal{A}(\mathcal{E})$ is in $\mathcal{K}$. At the end, we provide an algorithm to compute $\mathcal{K}$. In the following lemma we prove that if a curve segment of $\mathcal{A}(\mathcal{E})$ partially belongs to $\mathcal{K}$ then it must entirely belong to $\mathcal{K}$.

Lemma 8.2.1. *Consider a curve segment α in $\mathcal{A}(\mathcal{E})$. If $\alpha \cap \mathcal{K} \neq \emptyset$ then $\alpha \subset \mathcal{K}$.*

Proof. Consider a point $c \in \alpha$ such that $c \in K(q)$. By Theorem 6.2.13, at least one immobilizing grasp such as (q', c') is sq-reachable from (q, c). According to Property 6.2.15, $c \in K(q')$. Since c' is on $\partial P[q']$, c cannot be on $\partial P[q']$. Therefore there is a connected portion α_c of the curve segment α, such that $\alpha_c \subset K(q')$ and c is in the interior of α_c. According to definition the of $\mathcal{K}$ we have that $\alpha_c \subset \mathcal{K}$. If we repeat the same argument for the points on the left and right side of α_c, we can conclude that $\alpha \subset \mathcal{K}$. □

Consider an equilibrium grasp $(q, c) \in G$. Lemma 8.2.2 shows that to check whether $c \in \mathcal{K}$ a constant number of specific three-finger arrangements close to the equilibrium grasp can be checked to see if they are caging or non-caging. Moreover, if we check whether $c \in \mathcal{K}$ (for some $c \in E(q)$) we know that the entire arc belongs to $\mathcal{K}$ by Lemma 8.2.1. We use a triangulation of the exterior of P. In the proof of Lemma 8.2.2 we consider the triangles neighbor to the features (edges and/or vertices) involved in the equilibrium grasp and use the arrangement $\mathcal{A}(\mathcal{E})$ of equilibrium curves to choose a number of three-finger arrangements sufficiently close to the equilibrium grasp.

Lemma 8.2.2. *Assume that $c \in E(q)$. To check $c \in \mathcal{K}$ a constant number of specific three-finger arrangements close to (q, c) can be checked to see if they are caging or non-caging.*

Proof. Let $c = (c_x, c_y)$. When $c \in K(q)$ and $\epsilon \in \mathbb{R}^+$ is sufficiently small there is a set of disconnected components in $G((c_x, c_y - \epsilon))$ that join together in $G((c_x, c_y))$ at (q, c) and all but one components consist of caging placements [Rimon and Blake, 1996]. We use this fact to compute a suitable value for ϵ, and a constant number of specific three-finger arrangements based on the computed ϵ. The ϵ is computed such that if some of these three-finger arrangements are caging and some are non-caging, then it is guaranteed that $c \in \mathcal{K}$.

First consider the following notions. Let

$$\mathcal{H}((c_x, c_y), \delta) = \{(q, (c_x, {c'}_y)) \in G | 0 < c_y - {c'}_y \leq \delta\},$$

in which $\delta \in \mathbb{R}^+$. $\mathcal{H}((c_x, c_y), \delta)$ is the set of all three-finger arrangements whose x-coordinates are fixed and whose y-coordinates are between c_y and $c_y - \delta$. Consider the set of triples of triangles $\mathcal{T}$ that are adjacent to the incident features of the grasp (q, c). (The exterior of P can be triangulated in a way that the cardinality of $\mathcal{T}$ be constant.) Let $\mathcal{H}_{\mathcal{T}}((c_x, c_y), \delta)$ be a subset of $\mathcal{H}((c_x, c_y), \delta)$ in which the placements of the fingers are restricted to be in $\mathcal{T}$.

To verify that $c \in \mathcal{K}$ we determine a value for ϵ such that the components join together or become disconnected in $\mathcal{H}((q, c), \epsilon)$ only at (q, c). This condition on ϵ guarantees that all points inside a component in $\mathcal{H}((q, c), \epsilon)$ have the same caging status. On the other hand all the components that become connected at (q, c) in $\mathcal{H}((q, c), \epsilon)$ are also present in $\mathcal{H}_{\mathcal{T}}((q, c), \epsilon)$. Therefore we can check whether there is a caging component in $\mathcal{H}((q, c), \epsilon)$ that merges with a non-caging component by considering the merging components at (q, c) in $\mathcal{H}_{\mathcal{T}}((q, c), \epsilon)$.

To determine the value of ϵ we need to recall the following notion. Consider a cell $\mathcal{L} \subset \mathbb{R}^2$ in $\mathcal{A}(\mathcal{E})$ that is not covered by $P[q]$, and a connected component $m \subset ((\mathbb{R}^2 \setminus P[q]) \cap \mathcal{L})$. Then for all placements c_1 and c_2 such that $c_1, c_2 \in m$ we have $c_1 \in C(q) \Leftrightarrow c_2 \in C(q)$. Since by Lemma 6.1.1 there is no vertical equilibrium curve, every point on an equilibrium curve is adjacent to two cells of the arrangement: one above it and one below it.

By considering the arrangement $\mathcal{A}(\mathcal{E})$ and the cell below c in the arrangement, determine a point $c' = (c_x, {c'}_y)$ vertically below c in the cell. Let $\epsilon' = c_y - {c'}_y$. Recall that in a one parameter gripper when two components join together or become disconnected from each other the gripper's configuration corresponds to an equilibrium grasp. Therefore, there is no other equilibrium curve that intersects the line segment $c'c$.

The set $\mathcal{H}_{\mathcal{T}}((q, c), \epsilon')$ contains a constant number of connected components each of which has a constant complexity too. Let ρ be the merging components at (q, c) in $\mathcal{H}_{\mathcal{T}}((q, c), \epsilon')$, and consider one of its components α. Since α is a connected component, projecting α to the y-axis results in a connected real interval $\mathfrak{I}(\alpha)$ such that $\sup(\mathfrak{I}(\alpha)) = c_y$[1]. Let $\xi = c_y - \max_{\alpha \in \rho}[\inf(\mathfrak{I}(\alpha))]$ in which $\xi \in \mathbb{R}^+$. Clearly $0 < \xi \leq \epsilon'$. The way we have determined ξ ensures that all components that merge together at

[1]For a set $X \subset \mathbb{R}$, $\sup(X)$ and $\inf(X)$ are the least upper bound and greatest lower bound of X respectively.

(q, c) are present in $\mathcal{H}_{\mathcal{T}}((q, c), \xi)$. Because of the constant complexity of $\mathcal{H}_{\mathcal{T}}((q, c), \epsilon')$, it is possible to determine the value of $\xi \in \mathbb{R}^+$ in constant time. Finally, let $\epsilon = \xi/2$.

For a given placement c of the third finger, define

$$W(c) = \{q \in Q | (q, c) \in G\},$$

which is the set of all possible placements of the polygon P not intersecting c and the base fingers b_1 and b_2. Redefine ρ to be the merging components at (q, c) in $\mathcal{H}_{\mathcal{T}}((q, c), \epsilon)$. Since all three-finger arrangements in each component are either all caging or all non-caging, to check the caging status of each component α in ρ, an arbitrary placement $(q', (c_x, c_y - \epsilon))$ in $\alpha \cap W((c_x, c_y - \epsilon))$ is selected. The caging status of the selected placement $(q', (c_x, c_y - \epsilon))$ is determined by the caging status of its corresponding node in the connectivity graph. The connectivity graph is computed once for the shape formed by the third-finger placement $(c_x, c_y - \epsilon)$ and the base fingers b_1 and b_2, and it can be used to determine the caging status of all components in ρ. Since the complexity of ρ is constant, a constant number of nodes are checked for the caging property in the graph. □

We use the result of Lemma 8.2.2 to compute an orientation interval which contains the orientations of the caging arrangements sufficiently close to the given equilibrium grasp. We use this orientation interval in Section 8.4 to identify the clockwise or counterclockwise orientations of the polygon for which c is not on the caging boundary. To compute this interval, when $c \in \mathcal{K}$, we consider the components of ρ (see the proof of Lemma 8.2.2) that correspond to caging placements and map them to the θ-axis, which results in the orientation interval.

By Lemma 8.2.1 we can conclude that if for an arbitrary equilibrium grasp $(q, c) \in G$ on a curve segment α of $\mathcal{E}$ we have that $c \in \mathcal{K}$ then $\alpha \subset \mathcal{K}$. By using a similar approach we used in Section 5.3, we present an algorithm to compute $\mathcal{K}$ completely. First we briefly describe the approach we followed in Section 5.3 and then we explain our new approach to compute $\mathcal{K}$. In Section 5.3 the placements of the base fingers and the polygon were given, and we computed the caging set for the third finger. We considered the decomposition $\mathcal{A}(\mathcal{E})$. Each three-finger arrangement specified the mutual distances between the fingers, to which we referred as the shape of the three-finger arrangement. Given the shape, a graph was computed to which we referred as the connectivity graph. We used the graph to see whether a three-finger arrangement with a similar shape was caging. Briefly the algorithm of Section 5.3 is as follows. We placed the third finger in each cell of the decomposition which together with the base fingers specified a three-finger arrangement. We used the graph to see whether the three-finger arrangement was caging and we reported the complete cell as a caging cell if the three-finger arrangement was caging. Here however, a placement of the third finger in each cell of the decomposition does not specify a three-finger arrangement but only the shape of a three-finger arrangement; because, the placement of P is not given. We can use the shape to compute the connectivity graph, and then we should check whether there is any caging arrangement with the same shape whose cell is on the boundary of caging set. To check the existence of those caging arrangements for

an equilibrium grasp we use the result of Lemma 8.2.2 to compute a shape (used to compute the connectivity graph) and then to check a constant number of three-finger arrangements close enough to the equilibrium grasp to see if they are caging.

Theorem 8.2.3. *$\mathcal{K}$ can be computed in $O(n^6 \log^2 n)$ time.*

Proof. Consider an equilibrium curve e and one of its curve segments α in $\mathcal{A}(\mathcal{E})$. By Theorem 6.1.2 and Lemma 8.2.1 to check whether α is ever part of the caging boundary, an arbitrary equilibrium grasp $(q, c) \in G$ on α can be checked to see if $c \in \mathcal{K}$. By Lemma 8.2.2, only a constant number of placements close to that grasp needs to be checked for the property.

We use the same approach we used in Section 5.3 to compute all possible connectivity graphs. Here, however, for each curve segment of $\mathcal{A}(\mathcal{E})$ the caging status of a constant number of placements in a sufficiently small neighborhood (see Lemma 8.2.2) are queried each time. The set $\mathcal{K}$ can be computed in $O(n^6 \log^2 n)$ time. □

8.3 Approach and data structure

In this section, we develop a data structure that can be used to compute all curve segments of $K(q)$ when q is a canonical placement. We explain in the next section how to determine $K(q')$ from $K(q)$ when q' is an arbitrary placement of P and q is the canonical placement of q'. From now on, we focus only on the canonical placements of P.

The data structure associates the curve segments of $\mathcal{K}$ with orientation θ if and only if they appear in $K(\theta)$. The data structure is the set of curve segments of $\mathcal{K}$ associated with two-dimensional regions in (x, θ)-space. Every curve segment of $\mathcal{K}$ is associated with a region in (x, θ)-space, and no two regions intersect. As we compute the data structure, we can report $K(\alpha)$ by finding the regions that intersect the line $\theta = \alpha$ and reporting their associated curve segments of $\mathcal{K}$. In the rest of this section we explain the data structure in more detail and specify how to compute it and also the asymptotic time required to compute it.

Since P is a convex polygon, every canonical placement can be specified by a single parameter, which is the angle θ. Recall that, the union of $K(\theta)$ for all possible orientations θ is $\mathcal{K}$.

Lemma 8.3.1. *The set of all orientations such as θ for which a certain point (x_p, y_p) of $\mathcal{K}$ is on $K(\theta)$ forms an orientation-interval.*

Proof. Let g be the equilibrium grasp that corresponds to the point (x_p, y_p) of $\mathcal{K}$. Consider the set of all caging arrangements whose escaping equilibrium grasp is g. This set is a connected component in configuration space. The caging arrangements can be arbitrarily close to g in the configuration space. Therefore, the projection of this set to the subspace θ forms an interval. Consider an arbitrary caging arrangement in this set with orientation α. This three-finger arrangement corresponds to a canonical arrangement with the same orientation. Starting from the canonical arrangement, the third

finger can be stretched without intersecting the polygon until the third finger be placed on $K(\alpha)$. Clearly, the orientation of the resulting three-finger arrangement is α and the third finger is at (x_p, y_p). Therefore, the set of all orientations such as θ of all caging arrangements whose escaping equilibrium grasp is g is the same as the set of all orientations such as θ of all canonical caging arrangements whose escaping equilibrium grasp is g. The set of all orientations such as θ of all canonical caging arrangements whose escaping equilibrium grasp is g is the same as the set of all orientations such as θ for which a certain point (x_p, y_p) of $\mathcal{K}$ is on $K(\theta)$. Therefore, the set of all orientations such as θ for which a certain point (x_p, y_p) of $\mathcal{K}$ is on $K(\theta)$ forms an interval and we can represent this interval by a line segment in canonical arrangement space parallel to θ-axis that contains the canonical arrangement of the equilibrium grasp. The line segment does not cross the surface patches of $\mathcal{P}$ and either it has no endpoints or both of them are on $\mathcal{P}$. □

To associate $(x_p, y_p) \in \mathcal{K}$ with the mentioned orientation-interval I_θ we consider (x, θ)-space and associate x_p, i.e. the x-coordinate of the point, with I_θ. Consider all points in (x, θ)-space that are associated with the points of a specific curve segment of $\mathcal{K}$. This set of points forms a connected two-dimensional region in that space. The data structure is the set of curve segments of $\mathcal{K}$ associated with their corresponding two-dimensional regions in (x, θ)-space. Every curve segment of $\mathcal{K}$ is associated with a region in (x, θ)-space, and no two regions intersect. Since all curve segments of $\mathcal{K}$ are x-monotone, the caging regions of all curve segments of $\mathcal{K}$ are disjoint.

As we compute the data structure, we can report $K(\alpha)$ by finding the regions that intersect the line $\theta = \alpha$ and reporting their associated curve segments of $\mathcal{K}$. A canonical placement of the polygon with orientation α is caged with the base fingers and the third finger if and only if there is a region that contains the point whose x-coordinate and θ-coordinate are the same as the x-coordinate of the third finger and the orientation of the polygon respectively.

Let $\mathcal{P}$ be a three-dimensional shape in canonical arrangement space $\mathbb{R}^2 \times \mathbb{S}^1$ that represents all possible canonical placements of P. The set of surface patches of $\mathcal{P}$ may consist of several connected components. Recall that β is the number of pairs of edges that contain a pair of points at distance d. The surface patches of $\mathcal{P}$ contain $O(n\beta)$ piecewise-smooth algebraic surface patches of constant degree with no self-intersections. The canonical grasps of three-finger equilibrium grasps that are not immobilizing are located on the common boundary of surface patches of $\mathcal{P}$. The immobilizing grasps play no role on the boundary of caging region and therefore we do not consider them.

Now we explain how we compute the associated orientation-interval of a point $c = (c_x, c_y)$ on a curve segment α of $\mathcal{K}$. Let (q, c) be the corresponding equilibrium grasp of c, in which $q = (q_x, q_y, q_\theta)$. Let (q', c) be the canonical arrangement of (q, c), in which $q' = (q'_x, q'_y, q_\theta)$. The three-finger arrangement (q', c) corresponds to the three-dimensional point (c, q_θ) in canonical arrangement space. Walking on two half-lines parallel to θ-axis emanating from (c, q_θ) corresponds to *all* canonical arrangements reachable from (q', c) by sliding the polygon on the two base fingers while keeping the polygon in contact with both of them. Both of the half-lines either do not intersect

$\mathcal{P}$ or both intersect $\mathcal{P}$. The half-lines (until their first intersections with $\mathcal{P}$ each specify an orientation) identify the orientation-interval $[\theta_1, \theta_2]$ of c.

Let θ be the vertical axis in canonical arrangement space and α a curve segment of $\mathcal{K}$. To compute the caging region of α, we need to identify the surfaces of $\mathcal{P}$ above and below the three-dimensional points (in canonical arrangement space) corresponding to each point of α. Here we explain how to find the surface patches above and below. We build a *vertical decomposition* of the surface patches of $\mathcal{P}$. Since the surface patches of $\mathcal{P}$ have no self intersection, the required vertical decomposition of $\mathcal{P}$ can be constructed in $O((n\beta)^{2+\epsilon})$ time in which ϵ is an arbitrary small positive constant [Chazelle et al., 1991, Schwarzkopf and Sharir, 1996]. We compute a vertical decomposition for each connected component of the set of surface patches of $\mathcal{P}$ separately. All points inside each cell of the computed vertical decomposition have the same surface patch above and the same surface patch below. We find the surface patches above and below each curve segment of $\mathcal{K}$ by traversing all cells of the vertical decomposition and finding the intersecting equilibrium curves with each cell. To find the intersecting equilibrium curves we do differently for two-finger and three-finger equilibrium curves. A three-finger equilibrium grasp is a canonical arrangement and its corresponding pint in canonical arrangement space is located on the common boundary of two surface patches of $\mathcal{P}$. We traverse all cells of the vertical decomposition; for each cell s, we consider the upper and lower surfaces of s and their boundaries that belong to three-finger equilibrium grasps; we associate s with parts of three-finger equilibrium curves that correspond to the curve segments on the boundaries of upper and lower surfaces of s. We use this information for each three-finger equilibrium curve to associate curve segments of $\mathcal{K}$ (or possibly part of them) that belong to three-finger equilibrium grasps with cells of vertical decomposition. The canonical representation of all two-finger equilibrium grasps in canonical arrangement space form a number of curves T_2; there are $O(n)$ such curves in canonical arrangement space; we traverse all cells of the vertical decomposition again; for each cell s, we find all curves in T_2 that intersect s, and associate s with the intersected part of each two-finger equilibrium curve. We use this information for each two-finger equilibrium curve to associate curve segments of $\mathcal{K}$ (or possibly part of them) that belong to two-finger equilibrium grasps with cells of vertical decomposition that contain them.

Now we explain how to compute the associated caging region for a curve segment α of $\mathcal{K}$. For each curve segment of $\mathcal{K}$ we have already associated the cells of the vertical decomposition that contain the curve. We use the associated cells of the vertical decomposition to find the upper and lower surface patches and then to compute the associated caging region for each part of α in constant time. The caging regions of all parts of α form its caging region. (The total number of parts of all curve segments of $\mathcal{K}$ is bounded by $O(n^6)$.)

Lemma 8.3.2. *The caging regions of all curve segments of $\mathcal{K}$ can be computed in $O(n^7)$ time.*

After that we compute the caging regions for all curve segments of $\mathcal{K}$, we take the union of all caging regions associated with a single equilibrium curve e, and we associate the inducing curve segments of $\mathcal{K}$ with each curve segment on the boundary

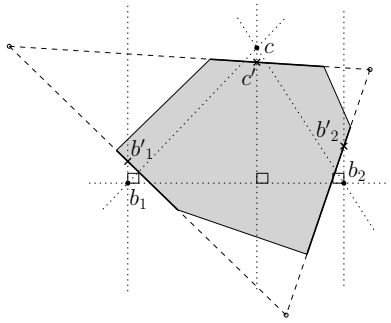


Figure 8.1: When (q', c) is a caging arrangement the caging status of (q, c) can be checked by answering six number of ray shooting queries at the fingers where q' is the canonical placement of q.

of the union. To compute the union, we remove the common boundary between every two adjacent critical-regions of two adjacent curve-segments of $\mathcal{K}$ that belongs to e.

Theorem 8.3.3. *The caging regions of all equilibrium curves can be computed in $O(n^7)$ time.*

8.4 Query processing

Given a convex polygon P and the distance d between the base fingers, we provide algorithms to answer two different queries in this section. In both queries an arbitrary placement q of the polygon is given. In the first query the placement c of the third finger is also given and it is required to answer whether $c \in C(q)$, i.e. whether c cages $P[q]$ with b_1 and b_2. In the second query it is required to report $K(q)$ (from which $C(q)$ can be reported easily).

8.4.1 Three-finger caging query

In this subsection an arbitrary placement $q = (q_x, q_y, q_\theta)$ of the polygon and a placement c of the third finger are given. We would like to determine whether (q, c) is a caging arrangement, which is called as the *three-finger caging query*.

Consider Figure 8.1. If (q, c) does not *vertically enclose* $P[q]$ then (q, c) is non-caging by Lemma 6.3.1. In a ray shooting query it is required to find the first point of contact between a query ray and a given polygon. As we compute the required data structure in $O(n)$ time, every ray shooting query can be answered in $O(\log n)$ time [Hershberger and Suri, 1995]. By considering the definition of the *vertical enclosing*, the following lemma easily follows.

Lemma 8.4.1. *It is possible to check whether (q, c) vertically encloses $P[q]$ by answering a constant number of ray shooting queries in $O(\log n)$ time.*

Note that we can check whether the extensions of γ_{b_1} and γ_{b_2} intersect each other below the x-axis similarly.

Assume that (q, c) vertically encloses $P[q]$ (otherwise it is non-caging). Let q' be the canonical placement of q. By Theorem 6.3.3, (q', c) is a caging arrangement if and only if (q, c) is a caging arrangement. Therefore to answer the three-finger caging query a data structure is required with which it is possible to determine whether (q', c) is caging. To answer the question we build a point-location data structure for the planar caging regions of all curve segments of $\mathcal{K}$ with θ as the vertical axis. The point-location data structure returns a vertical decomposition as well. (See Subsection 2.2.4 for more information on point-location data structure.) All points in each cell of the vertical decomposition are associated with the same curve segment of $\mathcal{K}$. The data structure can be computed in $O(n^2\beta^2 \log n)$ time and a point-location query can be answered in $O(\log n)$ time [de Berg et al., 2008]. Given the two parameters q_θ and c_x as the input, the point-location data structure returns the curve segment α whose caging region contains the pair (c_x, q_θ) as a point if it exists. If there is no such region then there is no such caging arrangement. Otherwise we consider α to obtain the maximum y-coordinate m_y of the third finger over all caging arrangements that are represented by the same two parameters (c_x, q_θ). The three-finger arrangement (q', c) is caging if and only if $c_y < m_y$.

Theorem 8.4.2. *The required data structure to answer three-finger caging queries can be computed in $O(n^7)$ time and can be queried in $O(\log n)$ time.*

8.4.2 Caging curves query

In this subsection we discuss an approach to report $K(q)$ for any query placement $q = (q_x, q_y, q_\theta)$ of the polygon. We refer to such a query as *caging curves query*. The main idea is that q_θ can be used to determine the related curve segments of $K(q)$ in $\mathcal{K}$.

Assume that b_1 and b_2 are on the boundary of $P[q]$. Let $\mathcal{U}$ be the set of equilibrium sub-curves whose caging regions are intersected by the line $\theta = q_\theta$. Each member of $\mathcal{U}$ constitutes a part of $K(q)$. This line intersection query can be performed easily using a simple one-dimensional interval-tree constructed on the projections of the curve segments on the boundary of caging regions of the equilibrium curves on θ-axis. Recall that we associate the inducing curve segment of $\mathcal{K}$ with each curve segment on the boundary of the caging region of an equilibrium curve. We use the association to find the parts of each equilibrium curve that are on the caging curves. When the base fingers are not on the boundary of $P[q]$, one extra step is required to report $K(q)$. Each curve of $\mathcal{U}$ may be (partially or totally) covered by $P[q]$. If all sub-curves of $\mathcal{U}$ are covered totally by $P[q]$, $C(q)$ is empty. The sub-curves of $\mathcal{U}$ are intersected with $P[q]$ to find the uncovered parts of the sub-curves. Each set of connected sub-curve parts with two endpoints on $\partial P[q]$ should be checked to see if they are part of the caging curves. To do that, we consider one of the two endpoints c and its corresponding three-finger arrangement (q, c) as a three-finger caging query, and check to see if (q, c) is caging by using the results of Subsection 8.4.1. If (q, c) is a caging arrangement, then the set of sub-curve are reported as part of $K(q)$.

Theorem 8.4.3. *$K(q)$ can be reported in $O(n+k)$ time, where k is the complexity of the caging curves when the base fingers are on $\partial P[q]$, and k is the maximum complexity of the caging curves over all placements of the polygon, otherwise.*

Proof. The interval tree can be queried in $O(\log(n^2\beta^2)+k) = O(\log n + k)$ time in which k is the number of sub-curves in $\mathcal{U}$. The intersection of $\mathcal{U}$ and $\partial P[q]$ can be computed in $O(n+k)$ time. When the base fingers are on $\partial P[q]$ the non-intersecting sub-curve parts are reported as the caging curves. When the base fingers are not on $\partial P[q]$, to find the part of $\mathcal{U}$ on $K(q)$, the required vertical ray shooting queries at the base fingers are performed once in $O(\log n)$ time. Then the caging property of the corresponding three-finger arrangement of each intersection point of $\mathcal{U}$ and $\partial P[q]$ can be checked in constant time. Here, we already know that the corresponding three-finger arrangement of each intersection point is a rigid translation of a caging arrangement. We only need to know whether the three-finger arrangement vertically encloses $P[q]$ to know whether it is caging. □

The algorithm is output sensitive when the base fingers are on $\partial P[q]$. However, when it is not the case, k may be far more than the complexity of the output.

Theorem 8.4.4. *The required data structure to answer caging curves queries can be computed in $O(n^7)$ time.*

8.5 Conclusion

We have used the collection of geometric facts of caging arrangements of convex polygons presented in Chapter 6 to derive an algorithm that computes a data structure in $O(n^7)$ time for a given convex polygon with n edges and a specified distance between two so-called base fingers. The data structure allows us to solve three-finger caging queries in $O(\log n)$ time and caging curves queries in $O(n+k)$ time. Clearly the running time of the algorithm for the latter problem is not necessarily proportional to the complexity of the caging curves when the base fingers are not on the polygon boundary. Therefore one direction to continue the work is to enhance the results by developing an output sensitive algorithm for that case. Using the same idea we will generalize the results to all distances of the base fingers in Chapter 9. Generalizing the results to non-convex polygons is also interesting but with such polygons the concept of canonical arrangements is not well-defined yet.

Chapter 9

Caging Convex Polygons with Three Fingers: Variable distance between the base fingers

9.1 Introduction

In this chapter we consider three-finger caging arrangements of convex polygons. Given a convex polygon P and placements of two fingers —referred to as the base fingers— the set of all placements of the third finger that together with the base fingers cage P form a number of two dimensional regions in the plane, to which we refer as the *caging set* of the third finger.

Erickson et al. [2003] provided the first complete algorithm for computing all three-finger caging arrangements of convex polygons. In their paper the base fingers were assumed to be placed along the boundary of the polygon and the caging set was computed for the third finger in $O(n^6)$ time. In chapter 5 we provided the first complete algorithm for computing all caging arrangements of arbitrary polygons for *any* given placement of the base fingers with $O(n^6 \log^2 n)$ running time. Given a convex polygon and the distance between the base fingers, in chapter 8 we reported a data structure that is computed within nearly the same running time. The data structure can be queried to report the caging set for any given placement of the base fingers. The data structure can also be queried to determine whether a given three-finger arrangement is caging. The solution is restricted to the case in which the distance between the base fingers equals some priorly given value d.

In this chapter we abandon the restriction on the distance between the base fingers by generalizing the data structure so that it can be queried for *arbitrary* distances between the base fingers. To do that we compute the set of shapes of all so called *critical arrangements,* which are three-finger arrangements on the boundary of caging sets. To answer the queries we introduce a new set of three-finger grasps called *vertex*

grasp. A vertex grasp is a three-finger grasp such that one of the fingers is at a vertex and the other two fingers are at the interior of two edges. A vertex grasp can be caging or non-caging. Every caging arrangement is associated with a unique critical arrangement, obtained by stretching the third finger vertically away (i.e. along a line perpendicular to the line connecting the base fingers) from the base fingers until the resulting three-finger arrangement becomes non-caging. We compute the critical arrangements associated with caging vertex-grasps and immobilizing grasps. Given a query arrangement we compute its associated vertex grasp or immobilizing grasp (if it exists, otherwise the query arrangement is not caging). Then we look for the critical arrangement associated with the vertex grasp (once again if it exists, otherwise the query arrangement is not caging). We show that the query arrangement is caging if and only if the height of the critical arrangement is more than the height of the query arrangement, where the *height* of a three-finger arrangement is the distance between the third finger and the line through the base fingers.

We prove in Subsection 9.3.1 that the set of shapes of all critical arrangements form a number of surface patches in the three-dimensional space in which we tackle the problem. Using these facts we outline our approach to solving both caging problems in Section 9.4. In Section 9.5 we present algorithms to answer the two different queries. We conclude the chapter with a discussion of future work.

9.2 Preliminaries

In Subsection 9.2.1 we introduce some notations and in Subsection 9.2.2 we study equilibrium grasps and how to represent them to solve the caging problem.

9.2.1 Definitions and assumptions

Unlike the previous three chapters, in this chapter we find it easier to fix the polygon and consider all possible placements of three fingers around the polygon. For this reason we redefine some of the concepts and notations according to this new assumption. Consider a planar three-finger hand. A placement of the hand, referred to as a *three-finger placement*, is denoted with the placement of each finger, e.g. $(a, b, c) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2$, such that $\triangle abc$ is a counter-clockwise triangle. Let $\overrightarrow{ab}$ be the directed half line that emanates from a and passes through b. Therefore in a three-finger placement (a, b, c) the placement c of the third finger is always on the left side of the directed half line $\overrightarrow{ab}$. A *three-finger arrangement* is a placement of the hand that does not intersect P. Define $F = \mathbb{R}^2 \setminus \text{int}(P)$. The set F^3 $(= F \times F \times F)$ is the set of all three-finger arrangements. Throughout the paper the first two fingers are referred to as the *base fingers.*

Consider the three-finger placement (a, b, c). Let $\measuredangle\overrightarrow{ab}$ be the angle between the directed half line $\overrightarrow{ab}$ and the positive x-axis half line in counter-clockwise direction. Define $D(c, ab)$ as the distance of the point c to the line passing through a and b. Alternatively, the placement (a, b, c) can be described by six parameters: the placement a

of the first finger on the plane (requiring two parameters), $\measuredangle\overrightarrow{ab}$, the algebraic distance of the feet of the altitude line drawn from c to ab to a, $D(c, ab)$, and $||\overline{ab}||$, which we display as a tuple $(x, y, \theta, f, h, d) \in \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^3$. We refer to this representation as the *triangular representation* of (a, b, c). The three-finger placement (a, b, c) of a triangular hand can be regarded to consist of a *shape* (f, h, d) which specifies the placements of the fingers with respect to each other and is displayed with $\sigma(a, b, c)$, and a placement (x, y, θ) of the hand.

Consider P and a given placement $(a, b) \in F^2$ of the base fingers. Let $C(a, b) \subset F$ be the set of all placements of the third finger that *together* with a and b form a three-finger arrangement that cages P. (Note that the term "together" refers to placements $c \in F$ of the third finger such that c is on the left side of the directed half line $\overrightarrow{ab}$). The set $C(a, b)$ consists of two dimensional regions in F and is referred to as the *caging set* of (a, b). The boundary of $C(a, b)$, i.e. $\partial C(a, b)$, is referred to as the *caging boundary* of (a, b). Let $K(a, b) = \partial C(a, b) \setminus \partial P$, which is the upper boundary of the caging set of (a, b).

9.2.2 Equilibrium grasps

Let $E : F^3 \rightarrow \{True, False\}$ be a predicate that determines whether a given three-finger arrangement is an equilibrium grasp. Let

$$E = \{(a, b, c) \in F^3 | E(a, b, c)\},$$

which is the set of all three-finger arrangements such that either two of the fingers or all three fingers form an equilibrium grasps of P. Let

$$\mathcal{E} = \{\lambda \in \mathbb{R}^3 | \exists (a, b, c) \in E : \sigma(a, b, c) = \lambda\},$$

which is the set of shapes that define at least one equilibrium grasp of P. Let us now consider the three-dimensional space (f, h, d) of all shapes, to which we refer as the *shape space*. The set $\mathcal{E}$ defines a number of 3D surfaces in shape space each of which is called an equilibrium surface. The cross section of an equilibrium surface with the plane $\boldsymbol{d} = d$, for some $d > 0$ defines a 2D curve, which is called an equilibrium curve. An immobilizing surface is a subset of an equilibrium surface whose shapes define at least one immobilizing grasp. (Recall that every immobilizing grasp is an equilibrium grasp.) Each equilibrium surface has a constant degree and the total number of equilibrium surfaces is $O(n^3)$. Therefore the arrangement $\mathcal{A}(\mathcal{E})$ contains $O(n^9)$ surface patches. As we will see later, $\mathcal{A}(\mathcal{E})$ can be used to compute the set of shapes of all critical three-finger arrangements.

Let

$$V = \{(a, b, c) \in F^3 | c \in K(a, b)\}$$

which is the set of all three-finger critical arrangements, and let

$$\mathcal{V} = \{\lambda \in \mathbb{R}^3 | \exists (a, b, c) \in V : \sigma(a, b, c) = \lambda\}$$

which is the set of shapes that define at least one critical arrangement. The following lemma holds by Theorem 5.2.1.

Lemma 9.2.1. $\mathcal{V} \subset \mathcal{E}$.

Although $\mathcal{V} \subset \mathcal{E}$, It is not immediately clear how to compute $\mathcal{V}$, and what its complexity will be. In Subsection 9.3.1 we prove that every surface patch of $\mathcal{A}(\mathcal{V})$ is the union of a set of surface patches of $\mathcal{A}(\mathcal{E})$. Therefore the complexity of $\mathcal{A}(\mathcal{E})$ is an upper bound on the complexity of $\mathcal{V}$.

The shape space is a three dimensional space that obviously encodes only the shapes of caging arrangements. To answer the queries we also need some information about the placements of the three-finger arrangements. (Recall the triangular representation of a three-finger arrangement.) As we will see in Section 9.5 we will encode the placements of caging arrangements by representing the set of all caging arrangements by the set of all caging vertex-grasps and immobilizing grasps. We will define a set of transformations that takes a query arrangement and outputs either a vertex grasp or an immobilizing grasp. The transformations are such that the resulting grasp is sq-reachable from the original query arrangement. (See Subsection 6.2.2 for the definition of sq-reachable.)

9.3 Critical arrangements

In this chapter we need the notions *vertically-enclosing arrangements* and *canonical arrangements* whose definitions and properties are explained in subsection 6.3.1. Since in this chapter we have assumed that the polygon is fixed (in contrast to the previous chapters that we have assumed the base fingers are fixed), the direction of the line connecting the base fingers must be assumed as the horizontal direction, and that the direction perpendicular to that line is therefore the vertical direction.

9.3.1 Computability of all critical arrangements

In this subsection we prove that that every surface patch of $\mathcal{A}(\mathcal{V})$ is the union of a set of surface patches of $\mathcal{A}(\mathcal{E})$. We begin with a simple lemma used in Lemma 9.3.2.

Lemma 9.3.1. *If $(a, b, c) \in F^3$ is a non-immobilizing caging arrangement, then there exists a reachable caging arrangement with the same shape such that none of the fingers is on ∂P.*

Proof. Let $(a', b', c') \in F^3$ be an immobilizing grasp that is sq-reachable from (a, b, c) and also assume that $(a', b', c'') \in F^3$ has the same shape as (a, b, c). Therefore (a', b', c'') is a caging arrangement reachable from (a, b, c) while a' and b' are on ∂P and c'' is not on ∂P. Consider a vertical line $l_{c''}$ drawn from c'' to the line $\overline{a'b'}$ passing through a' and b'. Clearly $l_{c''}$ intersects P at c'. Now if we rigidly translate the three-finger arrangement at (a', b', c'') along the vector $\overrightarrow{c''c'}$ for $||\overrightarrow{c''c'}||/2$, during the translation P is not intersected and the resulting three-finger arrangement has no finger on ∂P. □

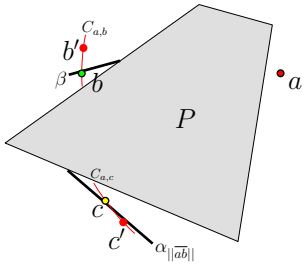


Figure 9.1: Illustration of Lemma 9.3.2

Lemma 9.3.2 is the main result of this subsection. It says that if a surface patch of $\mathcal{A}(\mathcal{E})$ partially belongs to $\mathcal{V}$ then it must entirely belong to $\mathcal{V}$.

Lemma 9.3.2. *If a surface patch α of $\mathcal{A}(\mathcal{E})$ satisfies $\alpha \cap \mathcal{V} \neq \emptyset$ then $\alpha \subset \mathcal{V}$.*

Proof. Let the 2D curve segment α_d be the cross section of α with the plane $\boldsymbol{d} = d$, for some $d > 0$. Consider an arbitrary shape $(f, h, d) \in (\alpha \cap \mathcal{V})$ such that it is in the interior of α and therefore the interior of α_d. We show that there is a closed interval $[d_1, d_2]$ such that $d_1, d_2 \in \mathbb{R}^+$ and $d_1 < d < d_2$ for which for any distance $d' \in [d_1, d_2]$ we have $\alpha_{d'} \subset \mathcal{V}$. Repeating the same argument for d_1 and d_2 we can eventually cover all the surface of α and therefore prove that $\alpha \subset \mathcal{V}$.

Consider Figure 9.1. Consider a critical three-finger arrangement $(a, b, c) \in F^3$ whose shape is (f, h, d) (note that $d = ||\overline{ab}||$) and none of its fingers is on ∂P. The three-finger arrangement (a, b, c) exists by Lemma 9.3.1. By Lemma 8.2.1 we have $\alpha_{||\overline{ab}||} \subset \mathcal{V}$. There are three types of equilibrium curves in $K(a, b)$: line segments, limaçon of Pascal curves, and circular arcs centered at the base fingers placements by Theorem 5.2.2. Without loss of generality assume that $\alpha_{||\overline{ab}||}$ is either a circular arc centered at b, a line segment, or a limaçon of Pascal curve.

Consider the circle $C_{a,c}$ centered at a passing through c. Since the complexity of $\alpha_{||\overline{ab}||}$ is constant and also since $\alpha_{||\overline{ab}||}$ is not a circular arc centered at a, there are a constant number of points on $\alpha_{||\overline{ab}||}$ for c at which $C_{a,c}$ is tangent to $\alpha_{||\overline{ab}||}$. Therefore assume that c is not one of those points on $\alpha_{||\overline{ab}||}$. We can assume so, because we are free to choose an arbitrary point on $\alpha_{||\overline{ab}||}$. Since $C_{a,c}$ is not tangent to $\alpha_{||\overline{ab}||}$, it intersects the curve segment $\alpha_{||\overline{ab}||}$ at c.

Consider the three-finger arrangement (c, a, b) with the third finger at b and the base fingers at c and a. We show that the third finger placement b is on the caging boundary $K(c, a)$. Consider the circle $C_{a,b}$ centered at a passing through b. Every placement b' (arbitrarily close to b) on $C_{a,b}$ for the third finger together with a and c define a three-finger arrangement (c, a, b'), that corresponds to a three-finger arrangement (a, b, c') in which the third finger is placed at a point c' on the circle $C_{a,c}$. These two three-finger arrangements (a, b, c') and (c, a, b') are either both caging or both non-caging, because they are reachable from each other by a single rotation around a. Therefore in (c, a, b) the finger placement b is on the caging boundary $K(c, a)$.

The point b is part of an equilibrium curve segment $\beta \in K(c,a)$ on the caging boundary. The equilibrium curve segment β is necessarily induced by the same features of P that induce the equilibrium curve segment $\alpha_{||\overline{ab}||}$. Therefore if $\alpha_{||\overline{ab}||}$ is a circular arc centered at b, segment β will be a circular arc centered at c, and if $\alpha_{||\overline{ab}||}$ is a line segment or a limaçon of Pascal curve, β is either a limaçon of Pascal curve or a line segment. Moreover if c is in the interior of $\alpha_{||\overline{ab}||}$ then necessarily b is in the interior of β, because if b is on the intersection of two equilibrium curves, then c should also be on the intersection of their corresponding equilibrium curves. However we have assumed that c is in the interior of a curve segment on the caging boundary.

Since sliding the third finger of the latter three-finger arrangement on $C_{a,b}$ the caging status changes at b, the circle $C_{a,b}$ is not tangent to β at b. Therefore if $C_{a,c}$ is not tangent to $\alpha_{||\overline{ab}||}$ at c then $C_{a,b}$ is not tangent to β at b and vice versa. Therefore there exists a closed interval on β (including b) such that when one of the base fingers is placed on any of its points and the other base finger on a, c is on the caging boundary of the third finger. Therefore in one hand c is placed on an equilibrium curve induced by a single set of features of P. On the other hand this interval corresponds to a closed interval $[d_1, d_2]$ of distances between the base fingers such that $||\overline{ab}|| \in (d_1, d_2)$. Therefore for all placements b' of the second finger (of the former three-finger arrangement) on the mentioned interval on β, c is located on $\alpha_{||\overline{ab'}||}$ on the caging boundary (i.e. $\alpha_{||\overline{ab'}||} \subset K(a,b')$) in which $||\overline{ab'}|| \in [d_1, d_2]$. In other words $\alpha_{d'} \subset \mathcal{V}$ in which $d' \in [d_1, d_2]$. □

A surface patch of $\mathcal{A}(\mathcal{E})$ is called critical if it contains the shape of a critical three-finger arrangement in its interior. For every point of a critical surface patch in $\mathcal{A}(\mathcal{E})$, which represents a shape, there exists a critical three-finger arrangement with the same shape by Lemma 9.3.2. Lemma 9.3.2 allows us to determine the criticality of a patch of $\mathcal{A}(\mathcal{E})$ by just checking the criticality of a single point on that patch. Recall that every point of a surface patch of $\mathcal{A}(\mathcal{E})$ corresponds to an equilibrium grasp formed by three fingers, that can be either a two-finger or a three-finger equilibrium grasp.

9.3.2 Computing all critical surface patches

In this subsection we compute the set of shapes of all critical arrangements. The set of shapes of all critical arrangements corresponds to a set of surface patches of $\mathcal{A}(\mathcal{E})$ by Lemma 9.3.2. We refer to these patches as the critical surface patches of $\mathcal{A}(\mathcal{E})$. We have seen in Subsection 9.3.1 that it suffices to check a single point inside a patch for criticality to determine whether the entire patch is critical. To check whether a surface patch α of $\mathcal{A}(\mathcal{E})$ is critical we consider an arbitrary point in the interior of α and check whether the corresponding equilibrium grasp is critical.

We follow exactly the same approach explained in Section 8.2 to compute all critical surface patches. The only difference is that here we deal with an arrangement of three-dimensional surfaces.

Lemma 9.3.3. *All critical surface patches in the shape space of a convex polygon can be computed in $O(n^9 \log^2(n))$ time.*

Proof. The updating of the dynamic graph takes $O(\log^2 n)$ time each time a surface patch is crossed. We can check whether a surface patch is critical in $O(\log n)$ time by using the connectivity graph. Since there are $O(n^9)$ surface patches, the total running time is $O(n^9 \log^2 n)$. □

9.3.3 Computing the lower rigid translate and the canonical arrangement

Consider a given placement (a, b) of base fingers such that $||\overline{ab}||$ is less than the base diameter of P at orientation $\measuredangle\overrightarrow{ab}$. (See subsection 6.3.1 for the definition of base diameter.) Consider the direction perpendicular to the line ab as the vertical direction. There are two rigid translates of (a, b) on the boundary of P. Let (p, q) be the rigid translate that is lower than the other one. Given the convex polygon P, in this subsection we explain a data structure we compute to report the lower rigid translate (p, q) of a given base-finger placement (a, b) as a query. By definition, if (p, q, r) is the canonical arrangement of (a, b, c) then (p, q) is the lower rigid translate of (a, b). To compute the canonical arrangement of (a, b, c), first we compute the lower rigid translate (p, q) of (a, b) and then we check whether or not r is inside P.

Consider a line parallel to and below the base line of $\measuredangle\overrightarrow{ab}$ that is tangent to P. The length of the intersection of a parallel line increases monotonically from the tangent line to the base line. We can sort the in-between vertices according to their distances from the tangent line. Considering a parallel line passing at an in-between vertex, the line intersects an edge of P. Therefore each in-between vertex is associated with an edge. Continuously changing the orientation, at certain orientations the order of the in-between vertices changes when the orientation of a line passing at two vertices equals the current orientation. Therefore considering all orientations, there are a number of orientation intervals such that for each interval the sorted list of the in-between vertices and also their associated edges are fixed. Since there are $O(n^2)$ pairs of vertices, there are also $O(n^2)$ orientation intervals. For each interval we store the sorted list of the in-between vertices along with their associated edges. Using an orientation-interval tree it is possible to find in $O(\log n)$ time the interval containing the orientation $\measuredangle\overrightarrow{ab}$. Then using a binary search on the sorted list of the in-between vertices (and their associated edges) of the containing interval, we can compute the edges of P on which p and q must reside.

We use an interval tree of orientations [de Berg et al., 2008] as a data structure to compute the lower rigid translate of a given base-finger placement. There are $O(n^2)$ different intervals such that for each interval the list of vertices (below the base-line) sorted according to their distance from the base line is fixed. For each interval we store this sorted list of $O(n)$ vertices. The following lemma states the running time and the storage space of the algorithm.

Lemma 9.3.4. *Given a placement of base fingers it is possible to compute its lower rigid translate in $O(\log n)$ time. The necessary data structure requires $O(n^3)$ storage space and can be computed in $O(n^3)$ time.*

9.4 Approach and data structure

In Subsection 9.4.1 we provide an overview of the two algorithms to answer the two caging queries. Recall that every caging arrangement is associated with a critical arrangement that is obtained by stretching the third finger until the cage is broken. The critical arrangement corresponds to a point on a critical surface patch, and the critical surface patch is a 2D cell in the arrangement of equilibrium surfaces. The critical arrangement of a caging arrangement can be computed in a constant time if the associated critical surface patch is known. In Subsection 9.4.2 we present a data structure that takes a so called vertex grasp as the input and returns whether or not the grasp is caging, and if the grasp is caging the data structure reports the associated critical surface patch. We present a similar data structure that takes an immobilizing grasp as the input and reports the associated critical surface patch. These data structures are used in Section 9.5 to answer the caging queries.

9.4.1 Overview of the algorithms

Here we explain briefly how we determine whether a three-finger arrangement is caging (which is explained in more detail in Subsection 9.5.1) and also how we compute the caging boundary for a given placement of two fingers (which is explained in more detail in Subsection 9.5.2).

Three-finger arrangement caging query: Consider a convex polygon P with n edges. We should compute a collection of data structures to answer whether a three-finger arrangement is caging. Consider the three-finger arrangement $(a, b, c) \in F^3$ as a query which can be described uniquely with the six parameters $(x, y, \theta, f, h, d) \in \mathbb{R}^2 \times \mathbb{S} \times \mathbb{R}^3$. First we check whether (a, b, c) is vertically enclosing. (See Subsection 6.3.1.) This property is necessary for (a, b, c) to be a caging arrangement. Then we determine the canonical arrangement $(a', b', c') \in (\partial P)^2 \times F$ (a, b, c). (See Subsection 6.3.1.) If the canonical arrangement does not exist the three-finger arrangement is again not caging. Moreover since (a, b, c) vertically encloses P then (a, b, c) is reachable from (a', b', c'). Again if the canonical arrangement is non-caging, the query arrangement is non-caging. We determine whether the canonical arrangement is caging by applying a number of transformations to the canonical arrangement. All transformations we apply are such that the resulting arrangements are sq-reachable from the query arrangement. (See Subsection 6.2.2 for more details on the notion sq-reachable.) (Stretching the third finger of a caging arrangement vertically away along a line perpendicular to the line connecting the base fingers, at a certain distance from the line connecting the base fingers, the three-finger arrangement becomes non-caging. Recall that this distance is called the *escaping distance* of that caging arrangement.) Note that the escaping distances of all caging arrangements sq-reachable from the query

arrangement are the same. Every canonical arrangement can be described uniquely by the four parameters $(\theta, f, h, d) \in \mathbb{S} \times \mathbb{R}^3$. We compute another related three-finger grasp $(a', b', c'') \in (\partial P)^3$ from (a', b', c') by squeezing the third finger along a line perpendicular to the line connecting the base fingers. Every such three-finger grasp with all fingers on ∂P can be described uniquely by the three parameters $(\theta, f, d) \in \mathbb{S} \times \mathbb{R}^2$. Every finger of the three-finger grasp (a', b', c'') resides on a different edge of P.

From the three-finger grasp (a', b', c''), we allow the base fingers to slide along their adjacent edges of the polygon and the third finger to vertically squeeze (thus decreasing the height of the grasp). We stop this sliding and/or squeezing as soon as either the resulting three-finger grasp is an immobilizing grasp, or one of the fingers resides at a vertex. Let $(p, q, r) \in (\partial P)^3$ be the resulting grasp. In the case that one of the fingers is at a vertex we refer to (p, q, r) as a three-finger vertex-grasp. If the third finger placement r is at a vertex or (p, q, r) is an immobilizing grasp, (p, q, r) can be described uniquely by the two parameters $(\theta, d) \in \mathbb{S} \times \mathbb{R}$ and the triple of features (edges and/or vertices). The reason is that (θ, d) can uniquely describe the placements of the base fingers on their corresponding edges. Then, for a vertex-grasp which the third finger is at a vertex the placement of the third finger is at a specific vertex. In an immobilizing grasp the three orthogonal lines at the three finger-placements to the adjacent edges meet at a common point. Therefore, the placement of the third finger is uniquely defined when the placements of the base fingers are specified. For a vertex-grasp which one of the base fingers resides at a vertex, consider the feet of the altitude line drawn from the vertex to the edge on which the other base finger resides. Without loss of generality assume that p is at a vertex and q is on an edge e. If the feet of the altitude line on e is outside e then (p, q, r) can be described uniquely by the two parameters (f, d) and the triple of features. If the feet of the altitude line is inside e, then (p, q, r) can be described uniquely by the two parameters (f, d), the triple of features, and whether q is at the left-side or at the right-side of the altitude line. The reason is that for each d there are at most two possible placements of q on e on the opposite sides of the altitude line. Then the parameter f uniquely specifies where the third finger should be placed on its corresponding edge.

Three-finger vertex-grasps induced by a single triple of features including exactly one vertex defines a number of so called vertex-grasp surfaces in the shape space. (Immobilizing surfaces are defined in subsection 9.2.2.) For every vertex-grasp and immobilizing surface a data structure is computed in Subsection 9.4.2. The data structure returns a critical three-finger arrangement (p', q', r') associated with the given three-finger vertex-grasp or immobilizing grasp (p, q, r) if and only if (p, q, r) is a caging arrangement or a critical arrangement. If (p, q, r) is a non-caging arrangement, (a, b, c) is a non-caging arrangement as well. Note that when (p, q, r) is a caging arrangement, the height of (p', q', r') is equal to the escaping distance of (p, q, r). (Recall that the *height* of a three-finger arrangement is the distance between the third finger and the line through the base fingers.) If (p, q, r) is a caging arrangement, then to find out whether (a, b, c) is a caging arrangement or not, the height of (p', q', r') is compared to the height of the original three-finger arrangement (a, b, c). If the former is larger, (a, b, c) is a caging arrangement. Otherwise (a, b, c) does not cage P.

Caging boundary query: Given the placement $(a, b) \in F^2$ of two fingers, $\partial C(a, b)$ should be computed. Recall from Chapter 6 that $K(a, b)$ is the upper boundary of $\partial C(a, b)$. We compute the lower rigid translate (a', b') of (a, b). First we compute $K(a', b')$, and then we compute $K(a, b)$ from $K(a', b')$.

9.4.2 Data structure

In this subsection we present a data structure that takes a vertex grasp as the input and returns whether or not the grasp is caging, and if the grasp is caging the data structure reports the associated critical surface patch. We present a similar data structure that takes an immobilizing grasp as the input and reports the associated critical surface patch. (Note that an immobilizing grasp is both a caging arrangement and an equilibrium grasp by definition.) Recall that every caging arrangement is associated with a critical arrangement that is obtained by stretching the third finger until the cage is broken. The critical arrangement corresponds to a point on a critical surface patch, and the critical surface patch is a 2D cell in the arrangement of equilibrium surfaces. The critical arrangement of a caging arrangement can be computed in a constant time if the associated critical surface patch is known. First we introduce vertex grasps and the so called vertex-grasp surfaces in the shape space. Then we specify the data structure and its complexity in more detail. Finally, we explain how to compute the data structure.

A vertex grasp is a three-finger grasp in which one of the fingers is at a vertex and the other two fingers are at the interiors of two edges. A vertex-grasp surface is a surface in the shape space associated with a vertex and a pair of edges consisting of all shapes defined by grasps in which the three fingers contact the given vertex and two edges. There are $O(n^3)$ vertex-grasp surfaces. Depending on which finger is at the vertex, there will be three different vertex-grasp surfaces associated with a vertex and a pair of edges. Let $\mathcal{E}'$ be the union of equilibrium surfaces and vertex-grasp surfaces. The three-dimensional arrangement $\mathcal{A}(\mathcal{E}')$ of surfaces may divide a critical surface patch into more pieces. However, the total complexity of the arrangement of surfaces is still $O(n^9)$. In the rest of this subsection whenever we mention a critical surface patch we mean a piece of a critical surface patch in the three-dimensional arrangement $\mathcal{A}(\mathcal{E}')$ of surfaces. This arrangement of surfaces has a certain property that we use it later.

Here we describe the data structure that enables us to query whether a given vertex grasp is caging. If the corresponding grasp is caging the data structure returns the critical surface patch associated with the caging grasp as well. Consider a vertex-grasp surface $\mathcal{S}$. Every caging vertex grasp that is a point on $\mathcal{S}$, can be associated with at most one critical surface patch. The associated critical surface patches of all caging vertex grasps on $\mathcal{S}$, divide the surface of $\mathcal{S}$ into a number of two-dimensional disjoint regions; each region therefore corresponds exactly to one critical surface patch. We call these disjoint regions as the caging regions on $\mathcal{S}$. The total complexity of these caging regions is $O(n^9)$. If the sequence of polygon features involved in $\mathcal{S}$ is a triple (edge, edge, vertex), i.e. the third finger is at a vertex, then we map the surface of $\mathcal{S}$ (along with the caging regions each of which is associated with a critical surface patch) to the two-dimensional space (θ, d). Otherwise, if the sequence of polygon features involved

in $\mathcal{S}$ is a triple (vertex, edge, edge) or a triple (edge, vertex, edge), i.e. one of the base fingers is at a vertex, then we divide the surface of $\mathcal{S}$ into two surfaces (explained below) and we map each of them to the two-dimensional space (f, d). Assume in a vertex-grasp (p, q, r) that one of the base fingers p is at a vertex v and the other base finger q is on the edge e. We divide the surface of $\mathcal{S}$ into two surfaces based on whether q is at the left-side or the right-side of the altitude line drawn from v to e. The same facts also apply to the immobilizing surfaces. Similarly every immobilizing grasp, which is a point on an immobilizing surface, can be associated with exactly one critical surface patch. We map the surface of an immobilizing surface (along with the caging regions each of which is associated with a critical surface patch) to the two-dimensional space (θ, d).

For each vertex-grasp surface and each immobilizing surface we need to determine which region contains a given point. This problem is known as point location query in computational geometry. To answer a point location query efficiently on each surface we compute a planar point-location data structure of the caging regions [de Berg et al., 2008]. (See Subsection 2.2.4 for more information on point location.) The data structure is these planar point-location data structures in spaces (f, d) and (θ, d), each of which corresponds to exactly one vertex-grasp surface or one immobilizing surface. Since there are $O(n^3)$ vertex-grasp surfaces and $O(n^3)$ immobilizing surfaces the total complexity of the data structure is $O(n^9 \times n^3) = O(n^{12})$. For each such surface, since the complexity of the caging regions is $O(n^9)$, we can build the data structure in $O(n^9 \log n)$ time and then a point location query can be answered in $O(\log n)$ time. Therefore the total time needed to compute the data structure for all vertex-grasp surfaces and immobilizing surfaces is $O(n^3 \times n^9 \log n) = O(n^{12} \log n)$.

Here we explain a way to compute the caging regions on a vertex-grasp surface each of which is associated with a critical surface patch. We use the same algorithm to compute the caging regions on an immobilizing surface each of which is associated with a critical surface patch. First we describe a planar arrangement of curves on the surface of each equilibrium surface, that we use inside the algorithm. Recall that every critical surface patch is a 2D cell in the three-dimensional arrangement $\mathcal{A}(\mathcal{E}')$ of surfaces.

Consider an equilibrium surface $\mathcal{S}$. Consider the intersection of $\mathcal{S}$ with other equilibrium surfaces and vertex-grasp surfaces, that forms a set of 2D curves on $\mathcal{S}$. Let e_1 be this set of curves, that contains $O(n^3)$ curves. Project the vertex-grasp surfaces and immobilizing surfaces to $\mathcal{S}$ along the axis h. The boundaries of the projected surfaces form another set of 2D curves on $\mathcal{S}$. Let e_2 be this set of curves, that contains $O(n^3)$ curves. Consider a cell α in the planar arrangement $\mathcal{A}(e_1 \cup e_2)$ of curves on the surface of $\mathcal{S}$. Since α is a cell in the mentioned arrangement of curves, either α is a subset of a critical surface patch or it intersects no critical surface patch on the surface of $\mathcal{S}$. Again since α is a cell in the mentioned arrangement of curves, as we project α to a vertex-grasp surface $\mathcal{S}'$, either the projection is empty or the projection does not cross the boundaries of $\mathcal{S}'$. Similarly, as we project α to an immobilizing surface $\mathcal{S}'$, either the projection is empty or the projection does not cross the boundaries of $\mathcal{S}'$. The arrangement $\mathcal{A}(e_1 \cup e_2)$ of curves can be computed in $O(n^6 \log n)$ time. Since there are $O(n^3)$

equilibrium surfaces, the total time needed to compute these planar arrangements for all equilibrium surfaces is $O(n^3 \times n^6 \log n)$=$O(n^9 \log n)$.

The algorithm to compute the caging regions for each vertex-grasp surface and immobilizing surface is as follows. We compute the aforementioned planar arrangement for each equilibrium surface $\mathcal{S}$. We consider each cell α in the planar arrangement that is a subset of a critical surface patch inside the equilibrium surface $\mathcal{S}$, and compute the sq-reachable vertex-grasp surfaces and the sq-reachable immobilizing surfaces. To compute the sq-reachable surfaces we consider an arbitrary point (that specifies a shape) inside the cell α and compute a variation of the connectivity graph for the corresponding shape. We can choose an arbitrary point inside the cell α because α is a two dimensional sub-cell in the three-dimensional arrangement $\mathcal{A}(\mathcal{E}')$ of surfaces. This is the reason that we have considered $\mathcal{E}'$ instead of $\mathcal{E}$. (In the next paragraph we will explain about the connectivity graph.) After that we have computed the sq-reachable surfaces for all equilibrium surfaces, then for each vertex-grasp surface $\mathcal{S}$ we know the critical surface patches that are associated with some points inside $\mathcal{S}$. To compute the caging regions on $\mathcal{S}$ we can simply project those associated critical surface patches to the surface of $\mathcal{S}$ along the axis h. We follow the same procedure to compute the caging regions for each immobilizing surface.

We use a variation of the connectivity graph in which the hand has one-parameter freedom of squeezing the third finger vertically to the supporting line of the base fingers. (See Subsection 5.2.2 for more information about the connectivity graph.) This one-parameter connectivity graph contains $O(n^3)$ nodes and edges and can be built in $O(n^3)$ time. For every critical surface patch α we consider an arbitrary critical arrangement whose shape belongs to a point in the interior of α and build the corresponding one-parameter connectivity graph. The critical arrangement belongs to a node which is located inside a bounded component in the connectivity graph. By traversing all nodes of the corresponding bounded component of the connectivity graph we find all sq-reachable vertex-grasp and immobilizing surfaces.

From what we have explained so far, the following lemma follows that establishes the space complexity of the data structure and the running time of the algorithm that computes the data structure.

Lemma 9.4.1. *The complexity of caging regions and their associated critical surface patches of all vertex-grasp surfaces and immobilizing surfaces of a convex polygon is $O(n^{12})$ and they can be computed in $O(n^{12} \log n)$ time.*

9.5 Query processing

Given the convex polygon P, we build a number of data structures with which we can solve two problems: (1) for a given three-finger arrangement as a query, we give an algorithm in Subsection 9.5.1 that determines in $O(\log n)$ time whether the three-finger arrangement cages P, and (2) for a given placement of two fingers as a query, we give an algorithm in Subsection 9.5.2 that outputs in $O(n \log n + k)$ time all placements of the third finger such that the three fingers together constitute a caging arrangement

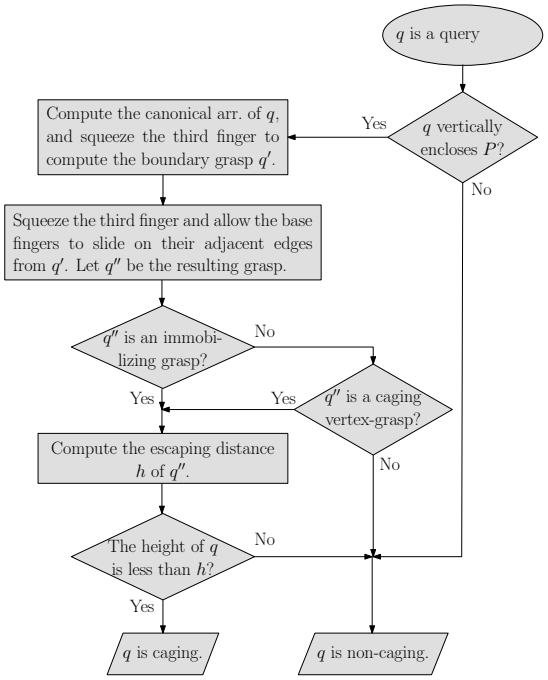


Figure 9.2: The graphical representation of our algorithm to determine whether a given query arrangement is caging.

of P. In the latter case k is proportional to the complexity of the output when the base fingers are on the boundary. When the base fingers are not on the boundary, k is proportional to the complexity of the caging set with the highest complexity.

9.5.1 Three-finger caging query

A three-finger arrangement is given as a query. Using the computed data structures, we present an algorithm to check whether the three-finger arrangement is caging. The algorithm checks whether or not a three-finger arrangement cages P in three steps. (See the graphical representation in Figure 9.2.)

1. Check whether the given query arrangement vertically encloses P. If it does not vertically enclose P it is not a caging arrangement by Lemma 6.3.1.
2. Find the canonical arrangement for the given three-finger arrangement. If the canonical arrangement does not exist the given three-finger arrangement is not caging by Lemma 6.3.2. Since the query arrangement vertically encloses P then it is reachable from the canonical arrangement by the same Lemma.
3. Determine the escaping distance of the canonical arrangement, which we will explain below, and compare it to the height of the original query arrangement. If

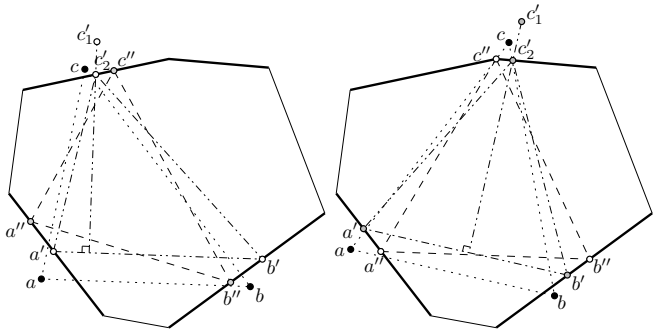


Figure 9.3: In both figures, (a, b, c) is the query arrangement, (a', b', c'_1) is the canonical arrangement, (a', b', c'_2) is the boundary grasp, and (a'', b'', c'') is the resulting grasp when we squeeze the third finger and allow the base fingers to slide on the edges.

the height of the given three-finger arrangement is less than the escaping distance then the given three-finger arrangement cages P. Otherwise it does not cage P.

It remains to explain how to compute the escaping distance of a given canonical arrangement. Consider the first vertex grasp or the immobilizing grasp that results from squeezing the third finger so that it resides on ∂P while sliding the base fingers on ∂P. If the third finger can be squeezed such that it becomes collinear with the base fingers then the given three-finger arrangement is not caging. Otherwise either one of the fingers resides at a vertex and in this case the resulting vertex grasp is part of a vertex-grasp surface (see the right example in Figure 9.3), or the resulting three-finger grasp forms an immobilizing grasp which is part of an immobilizing surface (see the left example in Figure 9.3). Using the corresponding 2D data structure in either (θ, d)-space or (f, d)-space we determine in $O(\log n)$ time which region contains the vertex grasp or the immobilizing grasp. If there is no such region, the given canonical arrangement is a non-caging arrangement. Otherwise we use the critical surface patch associated with the region to find the escaping distance in constant time. The height of the given three-finger arrangement is less than the height of the escaping distance if and only if the three-finger arrangement cages P.

Theorem 9.5.1. *The three-finger arrangement caging query can be answered in $O(\log n)$ time. The necessary data structure requires $O(n^{12})$ storage space and can be computed in $O(n^{12} \log n)$ time.*

9.5.2 Caging-boundary query

In this subsection we discuss an approach to report $\partial C(a, b)$ for any query placement $(a, b) \in F^2$ of the base fingers. The approach consists of two main steps. First we compute the rigid translate $(a', b') \in (\partial P)^2$ of (a, b). We report the curve segments on $\partial C(a, b)$ by first computing the curve segments on $\partial C(a', b')$ and then translating

them back to (a, b). We check the caging property of the resulting set using the same technique from Chapter 8.

We can compute the lower rigid translate $(a', b') \in (\partial P)^2$ of (a, b) in $O(\log n)$ time (see Subsection 9.3.3). First we compute $C(a', b')$. Now we explain how to compute the caging boundary vertically above (with respect to $\overrightarrow{a'b'}$) an interval on ∂P while the base fingers are on ∂P. Consider an edge of ∂P, which is a line segment l. Using the same technique of the Subsection 9.5.1 each point of l is associated with either a fixed vertex-grasp surface or a fixed immobilizing surface. Therefore considering all points on l, l is divided into a constant number of intervals such that each interval is associated with either a fixed vertex-grasp surface or a fixed immobilizing surface. Each interval when mapped to the corresponding 2D structure (of the vertex-grasp or immobilizing surface) in (θ, d)-space or (f, d)-space results in a vertical line segment (with d as the vertical axis), because the distance between the fingers is fixed. Using a vertical decomposition of the caging regions outlined in Subsection 9.4.2, it is possible to find the corresponding regions intersected by the vertical line segment in $O(\log n + k')$ time in which k' is the number of intersected regions. Each intersected region corresponds to a curve segment on $\partial C(a', b')$. We do this computation for $O(n)$ edges. Therefore, we can compute $\partial C(a', b')$ in $O(n \log n + k)$ time in which k is proportional to the complexity of $\partial C(a', b')$.

To compute $C(a, b)$ from $C(a', b')$ we follow the same procedure explained in the previous chapter in Subsection 8.4.2.

According to Lemma 6.3.4 and what we have mentioned the following lemma holds.

Theorem 9.5.2. *It is possible to report the caging boundary of a given placements of the base fingers in $O(n \log n + k)$ time in which k is proportional the complexity of the output description when the base fingers are on the boundary. If the base fingers are not on the boundary k is proportional to the complexity of the caging set with the highest complexity. The necessary data structure requires $O(n^{12})$ storage space and can be computed in $O(n^{12} \log n)$ time.*

9.6 Conclusion

In this chapter we have computed a collection of data structures in polynomial time for a given convex polygon with n edges. The data structures allow us to solve three-finger caging queries in $O(\log n)$ time and caging boundary queries in $O(n \log n + k)$ time in which k is proportional to the complexity of the output description when the base fingers are on the boundary. When the base fingers are not on the boundary, k is proportional to the complexity of the caging set with the highest complexity. However the running time analysis is based on very naive bounds on the complexity of overlays in Subsection 9.4.2. We conjecture that the actual bounds are far better than those that we have given. Therefore in practice we expect our solutions to be much more efficient. Generalizing the results to non-convex polygons is an interesting next step but with such polygons the concept of canonical arrangements still has to be properly defined and studied.

Chapter 10

Conclusion and Future Works

In this chapter we will summarize our results and give an overview on possible ways that our results can be extended. In this thesis we have studied two- and three-finger caging arrangements of polygons. All algorithms we have provided report the set of all possible solutions.

Two-finger caging

For the case of two-finger caging we have presented an algorithm that reports the set of all two-finger caging arrangements for a polygon. We have also shown how to compute a data structure for a given polygon that can be queried whether a given two-finger arrangement is caging. The results on two-finger caging arrangements of polygons can be mainly extended by providing an algorithm whose running time is sensitive to the complexity of the reported solution set, i.e. an output-sensitive algorithm. To improve the results of Chapter 4, we believe that there are ways of reducing or even removing the dependency on M, i.e. the number of maximum and minimum grasps, which we have explained in detail in the conclusion of that Chapter. Extending the results so that the query includes the radii of the disk fingers is interesting too. We believe that many of the ideas can be extended to three dimensions and hence be useful for the computation of two-finger caging arrangements of polyhedra. There are two reasons for this claim. Firstly we can locally explore the set of two-finger grasps on a pair of surfaces similar to what we did for a pair of edges. Secondly, the minimum and maximum grasps can also be defined in three-dimensional spaces.

Three-finger caging

For the case of three-finger caging we have presented an algorithm that takes a (convex or non-convex) polygon and the placements of two fingers as input and reports the set of all possible placements of the third finger that cage the polygon together with the

two given fingers. However the computation of three-finger caging arrangements of non-convex polygons is still a major open area to be discovered. The complexity of the caging set for non-convex polygons is the main open problem. We believe that the upper bound of $O(n^6)$ on the worst case complexity of the caging set is far from tight. We have proven in Chapter 3 that every two-finger caging arrangement is a two-finger squeezing caging or a two-finger stretching caging arrangement (or both). For a three-finger caging arrangement of a convex polygon we have proven in Chapter 6 that if we squeeze one of the fingers along a perpendicular line toward the line connecting the two other fingers, the resulting three-finger arrangement remains caging. It is interesting to see whether three-finger caging arrangements of non-convex polygons have similar properties. In particular, we would like to know for a caging arrangement if we squeeze or stretch one of the fingers the resulting three-finger arrangements remain caging. Even if this property does not hold for all types of non-convex polygons it would be interesting to find classes of non-convex polygons for which this property holds. Using the squeezing and stretching fact for two-finger caging we proved that two-finger caging arrangements are related to two-finger immobilizing grasps. Using the squeezing fact for three-finger caging of convex polygons we proved that three-finger caging arrangements are related to three-finger immobilizing grasps of convex polygons. It would be interesting to see whether three-finger caging arrangements are related to three-finger immobilizing grasps of non-convex polygons. Another interesting problem is to consider three-finger caging queries which asks whether or not a given query arrangement is caging. Studying special types of polygons such as star-shaped polygons with the purpose of improving the worst-case running time for these types of polygons is interesting too. Extending our results to three-dimensional spaces seems challenging, because of the problem of decomposing the free space into few simple cells. Hence we should look for alternative ways to tackle three-dimensional caging problems.

Three-finger caging arrangements of convex polygons

In this thesis we have studied three-finger caging arrangements of convex polygons in more detail. We have established a better upper bound on the worst case complexity of the caging set for convex polygons than for non-convex polygons. We have presented an algorithm that computes the caging set with a running time that is close to the provided worst-case complexity of the caging set. However we have not proven that this achieved upper bound is tight. It is interesting to find examples of convex polygons for which the complexity of the caging set is close to the provided upper bound.

For the case of three-finger caging problem and a given convex polygon we have presented a number of data structures that enable us to query whether a given three-finger arrangement is caging. Moreover given the placements of two fingers the data structures can be queried to report the set of all caging placements of the third finger. Clearly the provided upper bounds on the running time and also on the complexity of data structures are not tight. The running time analysis is based on very naive bounds on the complexity of overlays. We conjecture that the actual bounds are better than

those that we have given. In addition, our initial experiments on the complexity of the arrangement of equilibrium curves suggest that the complexity of the arrangement is less than quadratic in number of equilibrium curves.

Uncertainty

Although, the caging arrangements themselves have been used to deal with uncertainty in finger placements to achieve good firm grasps, we can consider uncertainty for the caging arrangements themselves. In practice, there are always errors both in finger placements and also in object shapes.

Caging arrangements can tolerate the error in finger placement by specifying the region in which the fingers can be placed and the range of acceptable distances between every two fingers. A related interesting problem is to identify cases where the regions for each finger are independent of each other. In other words, we would like to compute regions for which no matter where we place fingers inside the regions the resulting multi-finger arrangements cage the given object. One related work on independent regions, is the work done by Rimon and Blake [1996]. They studied such independent regions for two-finger caging arrangements.

There are a number of ways to specify shape variation from which we mention two of them here. One possible approach to specify shape variations is to consider a band around the original polygon boundary. The problem then changes into finding caging arrangements that cage any polygon defined within that band. One other possible way to specify shape variations is to specify the acceptable region for each vertex of a polygon. We could for example limit all acceptable places of a vertex to be within a circle of certain radius. The problem then changes into finding caging arrangements that cage any polygon whose vertices are within the acceptable specified regions.

Other versions of caging

We can also consider other versions of the caging problem. One such variant is to consider caging with some restrictions. For example to open a door with a hand we do not need to completely cage the handle of the door with the handle; instead we need a multi-finger arrangement that prevents the handle from escaping in a certain direction. One other example is to consider the quality of caging arrangement as a restriction. Informally, sometimes we do not need all caging arrangements, but a few good ones. One other variant is caging three-dimensional objects with a rope. For example to compute where we can place a loop of rope around a three dimensional object to cage it, and what are the acceptable lengths of the rope.

Applications

Applying caging to certain tasks is interesting too. Caging has already been used to do manipulations [Sudsang and Ponce, 2000, Diankov et al., 2008, Sudsang, 2000, Pereira et al., 2004]. It would be interesting to apply caging to perform manipulations that needs more precision; for example to place a peg in the hole, which is an assembly operation for which the friction effects such as jamming and wedging should be also considered.

References

Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington DC, USA, 2002. IEEE. ISBN 0-7803-7273-5.

J. M. Abel, W. Holzmann, and J. M. McCarthy. On grasping planar objects with two articulated fingers. In *Proeedings IEEE International Conference on Robotics and Automation*, March 1985.

Pankaj K. Agarwal, Alon Efrat, and Micha Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput*, 29:39–50, 1996.

I. J. Balaban. An optimal algorithm for finding segments intersections. In *SCG '95: Proceedings of the eleventh annual symposium on Computational geometry*, pages 211–219, New York, NY, USA, 1995. ACM Press. ISBN 0-89791-724-3. doi: http://doi.acm.org/10.1145/220279.220302.

A. S. Besicovitch. A net to hold a sphere. *The Mathematical Gazette*, 41:106–107, 1957.

A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *ICRA*, pages 348–353. IEEE, 2000.

B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir. A singly exponential stratification scheme for real semi-algebraic varieties and its applications. *Theor. Comput. Sci.*, 84(1):77–105, 1991. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/0304-3975(91)90261-Y.

Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6:485–524, 1991.

J. S. Cheong. *Immobilizing Grasps for Two- and Three-Dimensional Objects*. PhD thesis, Dept. of Computing and Information Sciences, Utrecht University, Nov. 2006.

J. S. Cheong and A. F. van der Stappen. Output-sensitive computation of all form-closure grasps of a semi-algebraic set. In *ICRA*, pages 772–778. IEEE, 2005.

J. S. Cheong, H. J. Haverkort, and A. F. van der Stappen. On computing all immobilizing grasps of a simple polygon with few contacts. In T. Ibaraki, N. Katoh, and H. Ono, editors, *ISAAC*, volume 2906 of *Lecture Notes in Computer Science*, pages 260–269. Springer, 2003. ISBN 3-540-20695-7.

J. S. Cheong, H. J. Haverkort, and A. F. van der Stappen. Computing all immobilizing grasps of a simple polygon with few contacts. *Algorithmica*, 44:117–136, 2006.

M. R. Cutkosky. Mechanical properties for the grasp of a robotic hand. Technical Report CMT-RI-TR-84-24, Carnegie Mellon Robotics Institute, 1984.

J. Czyzowicz, I. Stojmenovic, and J. Urrutia. Immobilizing a shape. *Int. J. Comput. Geometry Appl.*, 9(2):181–206, 1999.

C. Davidson and A. Blake. Error-tolerant visual planning of planar grasp. In *Sixth International Conference on Computer Vision (ICCV)*, pages 911–916, 1998a.

C. Davidson and A. Blake. Caging planar objects with a three-finger one-parameter gripper. In *ICRA*, pages 2722–2727. IEEE, 1998b.

M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 3rd ed. edition, 2008.

R. Diankov, S. Srinivasa, D. Ferguson, and J. Kuffner. Manipulation planning with caging grasps. In *IEEE International Conference on Humanoid Robots*, December 2008.

D. Ding, Y. H. Liu, Y. T. Shen, and G. Xiang. An efficient algorithm for computing a 3d form-closure grasp. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1223–1228, 2000. ISBN 0-7803-6348-5.

D. Ding, G. Xiang, Y. H. Liu, and M. Y. Wang. Fixture layout design for curved workpieces. In *ICRA* DBL [2002], pages 2906–2911. ISBN 0-7803-7273-5.

J. Erickson, S. Thite, F. Rothganger, and J. Ponce. Capturing a convex object with three discs. In *ICRA*, pages 2242–2247. IEEE, 2003.

J. Erickson, S. Thite, F. Rothganger, and J. Ponce. Capturing a convex object with three discs. *IEEE Tr. on Robotics*, 23(6):1133–1140, 2007.

J. Fink, N. Michael, and V. Kumar. Composition of vector fields for multi-robot manipulation via caging. In *Robotics Science and Systems*, Atlanta, GA, jun 2007.

J. Fink, M. A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *ICRA*, Pasedena, CA, May 2008.

K. Gopalakrishnan and K. Goldberg. Gripping parts at concave vertices. In *ICRA* DBL [2002], pages 1590–1596. ISBN 0-7803-7273-5.

D. Halperin. Arrangements. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 24, pages 529–562. CRC Press LLC, Boca Raton, FL, 2004.

M. R. Henzinger and V. King. Randomized fully dynamic graph algorithms with polylogarithmic time per operation. *J. ACM*, 46(4):502–516, 1999. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/320211.320215.

J. Hershberger and S. Suri. A pedestrian approach to ray shooting: shoot a ray, take a walk. *J. Algorithms*, 18(3):403–431, 1995. ISSN 0196-6774. doi: http://dx.doi.org/10.1006/jagm.1995.1017.

J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. In *STOC '98: Proceedings of the 30th annual ACM symposium on Theory of computing*, pages 79–89, New York, NY, USA, 1998. ACM Press. ISBN 0-89791-962-9. doi: http://doi.acm.org/10.1145/276698.276715.

J. W. Jameson. *Analytic Techniques for Automated Grasps*. PhD thesis, Dept. of Mechanical Engineering, Stanford University, June 1985.

J. R. Kerr. *An Analysis of Multi-Fingered Hands*. PhD thesis, Dept. of Mechanical Engineering, Stanford University, Jan. 1985.

W. Kuperberg. Problems on polytopes and convex sets. *DIMACS Workshop on Polytopes*, pages 584–589, 1990.

K. Lakshminarayana. Mechanics of form closure. *ASME, Paper No. 78-DET-32*, 1978.

J. W. Li, M. H. Jin, and H. Liu. A new algorithm for three-finger force-closure grasp of polygonal objects. In *ICRA*, pages 1800–1804. IEEE, 2003.

X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *Int. J. Robotics Res.*, 9(1):61–74, 1990.

M. Mason. *Mechanics of Robotic Manipulation*. MIT Press, August 2001. Intelligent Robotics and Autonomous Agents Series, ISBN 0-262-13396-2.

W. Meyer. Seven fingers allow force-torque closure grasps on any convex polyhedron. *Algorithmica*, 9(3):278–292, 1993.

B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2:541–558, 1987.

Ketan Mulmuley. A fast planar partition algorithm, i. *Journal of Symbolic Computation*, 10(3/4): 253–280, 1990.

J. R. Munkres. *Elementes of Algebraic Topology*. Addison-Wesley, 1984.

V. Nguyen. The synthesis of stable force-closure grasps. Technical Report AITR-905, Massachusetts Institute of Technology, Cambridge, MA, USA, 1986.

G. A. S. Pereira, V. Kumar, and M. F. M. Campos. Decentralized algorithms for multirobot manipulation via caging. *Int. J. Robotics Res.*, 2004.

P. Pipattanasomporn and A. Sudsang. Two-finger caging of concave polygon. In *ICRA*, pages 2137–2142. IEEE, 2006.

P. Pipattanasomporn, P. Vongmasa, and A. Sudsang. Two-finger squeezing caging of polygonal and polyhedral object. In *ICRA*, pages 205–210. IEEE, 2007.

J. Ponce, J. Burdick, and E. Rimon. Computing the immobilizing three-finger grasps of planar objects. In *Proceedings of the 1995 Workshop on Computational Kinematics*, pages 281–300, 1995.

J. Ponce, S. Sullivan, A. Sudsang, J. Boissonnat, and J. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research*, 16:11–35, 1997.

F. Reuleaux. *The Kinematics of Machinery*. Macmilly and Company, 1876. Republished by Dover in 1963.

E. Rimon. A curvature-based bound on the number of frictionless fingersrequired to immobilize three-dimensional objects. *IEEE Transactions on Robotics and Automation*, 17(5):679–697, Oct. 2001.

E. Rimon and A. Blake. Caging 2d bodies by 1-parameter two-fingered gripping systems. In *ICRA*, volume 2, pages 1458–1464. IEEE, 1996.

E. Rimon and J. Burdick. New bounds on the number of frictionless fingers required to immobilize planar objects. *Journal of Robotic Systems*, 12(6), June 1995.

E. Rimon and J. W. Burdick. Mobility of bodies in contact—part i: A 2nd–order mobility index for multiple–finger grasps. *IEEE Tr. on Robotics and Automation*, 14(5):696–717, 1998.

A. Rodriguez and M. Mason. Two finger caging: squeezing and stretching. In *Eighth Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2008.

J. K. Salisbury. *Kinematics and Force Analysis of Articulated Hands*. PhD thesis, Dept. of Mechanical Engineering, Stanford University, May 1982.

Otfried Schwarzkopf and Micha Sharir. Vertical decomposition of a single cell in a three-dimensional arrangement of surfaces and its applications. In *SCG '96: Proceedings of the twelfth annual symposium on Computational geometry*, pages 20–29, New York, NY, USA, 1996. ACM. ISBN 0-89791-804-5. doi: http://doi.acm.org/10.1145/237218.237230.

M. Sharir and P. K. Agarwal. *Davenport-Schinzel sequences and their geometric applications*. Cambridge University Press, New York, NY, USA, 1996. ISBN 0-521-47025-0.

G. Shephard. A sphere in a crate. *Journal of London Mathematical Society*, 40:433–434, 1965.

R. Somoff. Uber schraubengeschwindigkeiten eines festen korpers bei verschiedener zahl von stutzflachen. *Zietschrift fur Mathematik und Physik*, 42:133–153, 1897.

A. Sudsang. Grasping and in-hand manipulation: Geometry and algorithms. *Algorithmica*, 26 (3-4):466–493, 2000.

A. Sudsang and T. Luewirawong. Capturing a concave polygon with two disc-shaped fingers. In *ICRA*, pages 1121–1126. IEEE, 2003.

A. Sudsang and J. Ponce. A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane. In *ICRA*, pages 1068–1075. IEEE, 2000. ISBN 0-7803-5889-9.

A. Sudsang, J. Ponce, and N. Srinivasa. Grasping and in-hand manipulation: Experiments with a reconfigurable gripper. *Advanced Robotics*, 12(5):509–533, 1998.

A. Sudsang, J. Ponce, M. Hyman, and D. J. Kriegman. On manipulating polygonal objects with three 2-dof robots in the plane. In *ICRA*, pages 2227–2234, 1999.

A. F. van der Stappen, C. Wentink, and M. H. Overmars. Computing immobilizing grasps of polygonal parts. *International Journal of Robotics Research*, 19(5):467–479, 2000.

D. Wells. *The Penguin Dictionary of Curious and Interesting Geometry*. London: Penguin, 1991.

Samenvatting

Industriële robots worden ontworpen om een grote verscheidenheid aan taken in productielijnen van diverse industriële sectoren uit te voeren, zoals assemblage of het uitlijnen van onderdelen. Om deze taken uit te voeren, moet de robotarm het object op de juiste manier vastpakken.

Bij een voldoende stevige greep op het object is het gegarandeerd dat het object niet kan bewegen ten opzichte van de robot. Een dergelijk stevige greep is niet altijd nodig voor bepaalde operaties. Tot nu toe hebben onderzoekers en ontwerpers deze stevige greep gebruikt om met industriële robots objecten op te pakken en te verplaatsen. De mens, die vaak als voorbeeld dient voor onderzoekers, gebruikt echter vaak een losse greep om bepaalde operaties uit te voeren. Bijvoorbeeld, om een onderdeel te transporteren, is het niet nodig om dat onderdeel zodanig stevig vast te pakken dat het niet kan bewegen tussen de grijpende vingers. In plaats daarvan is het voldoende dat het onderdeel niet tussen de vingers door kan glippen. In dit proefschrift formaliseren we het idee van een 'losse' greep door het begrip 'caging' (lett. kooien). De hoeveelheid literatuur alsmede de korte historie van ongeveer 10 jaar, laten zien dat er nog niet veel bekend is over 'caging'. De initiële resultaten duiden er ook op dat 'caging' een ingewikkeld probleem is.

In dit proefschrift gaan we ervan uit dat de vingers zijn schijven met een straal van nul of meer. Een object is door een aantal vingers ge-'caged' als het onmogelijk is om het object naar een willekeurige positie te brengen, ver van zijn oorspronkelijke positie zonder met die beweging door een vinger heen te ewegen. We noemen een plaatsing van een aantal vingers een 'arrangement' van diezelfde vingers. Een caging-arrangement voor een object is een plaatsing van vingers zodanig dat het object niet kan ontsnappen. Een caging-arrangement garandeert dat, ongeacht de beweging (translatie of rotatie) die het object maakt tussen de vingers, het binnen het bereik van de vingers zal blijven en daardoor niet zal kunnen ontsnappen. Evenzo, een niet- caging-arrangement is een plaatsing van vingers, zodanig dat het object willekeurig ver kan worden geplaatst van zijn begin positie, zonder door een vinger heen te bewegen. Figuur 10.1 toont een caging-arrangement van drie vingers. Vervolgens toont Figuur 10.2 een niet-caging-arrangement van drie vingers en hoe het object tussen deze vingers door kan ontsnappen.

Er zijn verschillende algoritmes ontwikkeld die alle 'caging' oplossingen of een deelverzameling daarvan berekenen. In dit proefschrift berekenen we de complete verzameling aan oplossingen voor de problemen die we hebben bestudeerd. Er zijn vele scenario's waarbij men de hele oplossing wil weten. Een van die scenario's is dat we een vastgepakt object willen kunnen bewegen tussen een aantal obstakels door. We willen graag een mogelijke greep voor het object waarbij de vingers botsingen met de obstakels proberen te vermijden. Een ander scenario is dat we een vastgepakt object willen kunnen loslaten op een manier dat het in een specifieke oriëntatie stil komt te liggen.

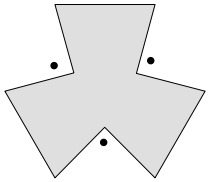


Figure 10.1: Een caging-arrangement van drie vingers.

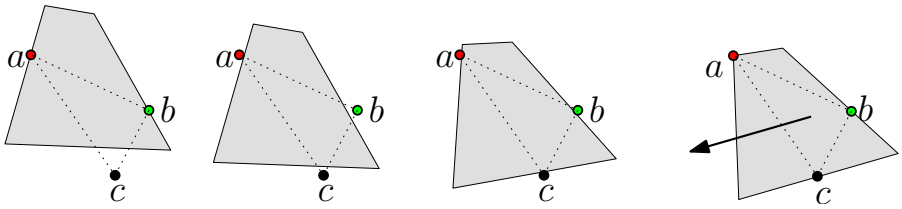


Figure 10.2: Een niet-caging arrangement (uiterst links) van drie vingers en een manier waarop het object aan de vingers kan ontsnappen.

Het is duidelijk dat caging met twee vingers niet altijd mogelijk is. Convexe polygonen bijvoorbeeld, kunnen altijd tussen twee vingers ontsnappen ongeacht hun positie. Figuur 10.4 illustreert een niet-convex polygon dat ook niet ge-caged kan worden met twee vingers. Daarom bekijken we ook drie-vingerige 'caging' arrangements.

In dit proefschrift bestuderen we het probleem om polygonen met twee of drie vingers te 'cagen'. In hoofdstuk 3 presenteren we een algoritme dat alle 'caging' arrangements berekent voor caging met twee vingers. We bewijzen dat elke caging arrangement samenknijpend 'squeezing' of rekkend 'stretching' is. Een caging arrangement is samenknijpend, wanneer de vingers niet naar een positie kunnen bewegen waarin de vingers samenvallen terwijl de afstand tussen de vingers ten hoogste gelijk is aan de aanvangsafstand tussen de vingers in het gegeven caging arrangement. Een rekkend arrangement is een arrangement waarin de vingers niet willekeurig ver van elkaar vandaan kunnen worden bewogen ten opzichte van het object, zonder dat de afstand kleiner wordt dan de afstand van de gegeven aanvangsposities in het arrangement. Als toevoeging daarop construeren we een datastructuur die kan vertellen of een gegeven twee-vinger arrangement caging is. Figuur 10.3 toont voorbeelden van een samenknijpend arrangement en een rekkend arrangement van twee vingers.

De combinatorische complexiteit van de set van alle twee-vingerige caging arrangements kan erg variëren. Aan de ene kant zijn er niet-convexe polygonen waarbij caging met twee vingers niet mogelijk is (Figuur 10.4 toont een dergelijk niet-convex polygon). Terwijl aan de andere kant er polygonen bestaan voor welke de set van caging arrangements groot is. Natuurlijk verwachten we van ons algoritme dat deze ons snel een antwoord geeft in het eerste geval, maar we vinden het acceptabel dat de rekentijd wat langer is in het tweede geval. De gewenste eigenschap om een hoge performance te halen wordt grotendeels bepaald door de complexiteit van de uitkomsten, welke in algoritmisch onderzoek 'output-sensitivity' wordt genoemd. In hoofdstuk 4 nemen we een eerste stap in de richting van een algoritme waarvan de rekentijd grotendeels proportioneel is aan de grootte van de verzameling van gerapporteerde

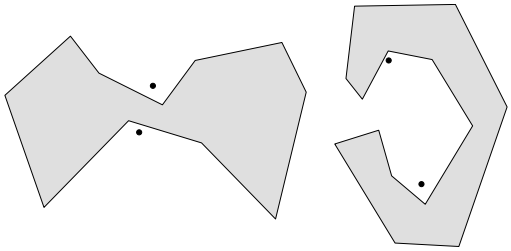


Figure 10.3: Een samenknijpend arrangement (links) en een rekkend arrangement van twee vingers (rechts).

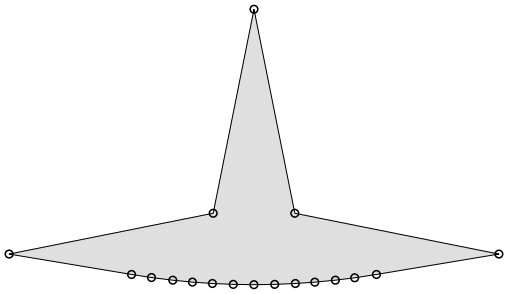


Figure 10.4: Een niet-convex polygon waarbij caging met twee vingers niet mogelijk is.

twee-vingerige caging arrangements. Vergeleken met bestaande algoritmen, welke een kwadratische afhankelijkheid hebben van de complexiteit van het onderdeel, hebben we een aanzienlijke gevoeligheid voor de grootte van de input ingeruild voor de gewenste 'output-sensitivity'. De output van het algoritme kan zelfs efficiënt antwoord geven op de vraag of een gegeven willekeurige twee-vinger plaatsing caging is.

Voor drie-vinger caging onderscheiden we drie problemen:

- Allereerst, gegeven een polygoon en plaatsing van twee vingers, de basisvingers, berekenen we alle mogelijke plaatsingen van de derde vinger, zodanig dat deze samen met de basisvingers een caging arrangement vormen.
- Ten tweede, gegeven een polygoon berekenen we een datastructuur om te beantwoorden of een gegeven arrangement van drie vingers caging is.
- Ten derde, voor een gegeven polygoon construeren we een datastructuur. Met behulp van deze datastructuur en een gegeven plaatsing van de basisvingers, kunnen we sneller alle mogelijke plaatsingen berekenen van de derde vinger zodanig dat de drie vingers een caging arrangement vormen.

In hoofdstuk 5 presenteren we een oplossing om de caging set voor convexe en niet-convexe polygonen te berekenen voor een gegeven plaatsing van de basisvingers. Aangetoond wordt dat de secties aan de rand van deze regio's die niet behoren tot de rand van de polygoon corresponderen met 'equilibrium grasps' . Wanneer een hand een object vast heeft in rust,

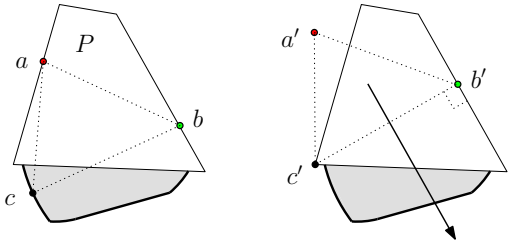


Figure 10.5: Een drie-vinger arrangement (links) waarbij de derde vinger op de rand van de caging set ligt, en de bijbehorende equilibriumgreep (rechts), welke een twee-vinger equilibrium greep is.

oefenen de vingers krachten en momenten uit die elkaar uit zouden moeten balanceren om de plaatsing van het object niet te verstoren. Van een dergelijke greep wordt gezegd dat hij een equilibrium bereikt. Figuur 10.5 toont een drie-vinger arrangement waarbij de derde vinger op de rand van de caging set ligt. De bijbehorende equilibrium greep, welke een twee-vinger equilibrium greep is, wordt getoond aan de rechterkant.

In hoofdstuk 7 tonen we aan dat de worst-case complexiteit van de caging set van convexe polygonen een stuk kleiner is dan die van niet-convexe polygonen. Om een bovengrens voor de complexiteit van de caging set van convexe polygonen te kunnen geven, hebben we eerst het aantal verschillende typen oppervlakken die een rol spelen in de complexiteit van de caging set verlaagd. Ten tweede hebben we een nieuwe manier geformuleerd om de caging set te berekenen aan de hand van deze oppervlakken. Daarnaast hebben we een efficiënt algoritme ontwikkeld om de caging set te berekenen door middel van een verdeel-en-heers-aanpak.

In hoofdstuk 8 nemen we aan dat het convexe polygon en de afstand tussen de basisvingers zijn gegeven. We berekenen een data structuur die kan worden gebruikt om de caging set van elke gegeven plaatsing van de basisvingers efficiënter te rapporteren. Wanneer de basisvingers op de rand van een polygoon liggen, blijkt de zoektijd grotendeels proportioneel te zijn aan de combinatorische complexiteit van de gerapporteerde caging set. Gebruikmakend van hetzelfde idee, bekijken we een andere zoekvraag: voor een gegeven plaatsing van de basisvingers en een plaatsing van de derde vinger, geven we een algoritme dat bepaalt of het resulterende arrangement de polygoon kooit.

In hoofdstuk 9 verlaten we de restricties over de afstand tussen de basisvingers door de datastructuur van hoofdstuk 8 te generaliseren, zodat die kan worden gebruikt voor willekeurige afstanden tussen de basisvingers.

Acknowledgment

Although doing research is not easy, it can be very fun and exciting. Imagin yourself working on a problem, and just after several weeks of thinking with frustrating results, suddenly a new idea sparkles and then you feel the growing of self confidence, which is exciting. However, the problem one tries to solve should be worthy to work on. For this reason, I would like to give my most special thank to my supervisor Frank van der Stappen, who proposed an interesting research subject to me. Beside being my good supervisor, during the past four years we have talked about many things in a way that I have always felt he is my friend. "Thank you!" loudly.

I would like to thank prof. Mark Overmars and the members of the reviewing committee: prof. Ken Goldberg, prof. Elon Rimon, prof. Mark de Berg, prof. Frans Groen, and prof. Remco Veltkamp for reading my thesis and for their constructive comments.

I would like to thank my coauthors Magdalene G. Borgelt, Marc van Kreveld, Maarten Löffler, Jun Luo, Damian Merrick, and Rodrigo I. Silveira. Together we have worked on a problem and I enjoyed the experience of team-working.

I would like to thank my colleagues who made my work place an enjoyable place. I would like to thank Iris, Jur, Dennis, Onno, Esther, Frank (ter Haar), Reinier, Mohammad Fatih, Roland, and Arno (with no specific order).

My son Ali was born in the Netherlands during my first PhD year. I and my wife have enjoyed taking our son to ING child playgroup. I would like to thank the organizers Wiebke and Tanya.

I would like to thank a number of friends that have helped me and my family to feel home in the Netherlands. I would like to thank Malekinejad, Honarvar, Samadi, Mousavi, Abaam, Ghamarian, Farshi, Fatemi, Khafaji, Moayeri, Hassani, Al-Tamimi, Saiegh, Al-Mashat, and Al-lawati.

I would like to thank my new colleagues in Tree-C Technology, Gerard, Bart, Ward, Rene, Nico, Nienke, Renate, Bastiaan, and Sander. A special thank goes to Bart den Hartog and Nico Kruithof who translated the summary into Dutch.

Without the support and care of my wife Roya I could have never finished my studies. I would like not only to thank but to dedicate my thesis to her.

Mostafa Vahedi
August 2009

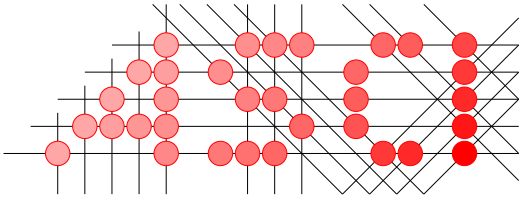
Curriculum Vitae

Mostafa Vahedi

22-09-1979	born in Khorramshahr, Iran
1996-2000	BS in Computer Engineering at Sharif University of Technology, Tehran, Iran
2000-2003	MS in Software Engineering at Sharif University of Technology, Tehran, Iran
2005-2009	Information and Computing Sciences at Utrecht University, the Netherlands

Colofon

This thesis was typeset by the author in LaTeX 2ε. The main body of the text was set using 10 points Palatino font.



Advanced School for Computing and Imaging

This work was carried out in the ASCI graduate school.
ASCI dissertation series number 184.

ISBN-10: 90-393-5135-2
ISBN-13: 978-90-393-5135-2