Caging Polygons with Two and Three Fingers

Mostafa Vahedi and A. Frank van der Stappen

The International Journal of Robotics Research 2008; 27; 1308
DOI: 10.1177/0278364908098485

Mostafa Vahedi
A. Frank van der Stappen
Department of Information and Computing Sciences,
Utrecht University,
3508 TB Utrecht, The Netherlands
{vahedi, frankst}@cs.uu.nl

Caging Polygons with Two and Three Fingers

Abstract

We study two- and three-finger caging grasps of a given polygonal object with n edges. A grasp is said to cage an object when it is impossible to take the object to a distant location without penetrating a finger. Using a classification into squeezing and stretching cagings, we provide an algorithm that reports all *caging grasps of two disk fingers in $O(n^2 \log n)$ time. Our result extends and improves a recent solution for point fingers (Pipattanasomporn and Sudsang 2006). In addition, we construct a data structure in $O(n^2 \log n)$ time requiring $O(n^2)$ space that can be queried in $O(\log n)$ time whether a given two-finger grasp cages the polygon. We also establish a relation between two-finger caging grasps and two-finger immobilizing grasps of polygons without parallel edges. We also study caging grasps with three point fingers. Given the placements of two so-called base fingers, the caging region is the set of* all *placements of the third finger that jointly with the base fingers forms a caging grasp of a polygonal object. Using the relation between equilibrium grasps and the boundary of the caging region, we present an algorithm that reports the entire caging region in $O(n^6 \log^2 n)$ time. Our result extends a previous solution that only applies to convex polygons (Erickson et al. 2007).*

KEY WORDS—Robotic manipulation, caging grasps, planer parts, geometric data structures

1. Introduction

The caging problem (or capturing problem) was posed by 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. In other words, a polygon is caged with a number of fixed fingers when it is impossible to take it to infinity without penetrating any finger.

Caging is related to the notions of form (and force) closure grasps (see, e.g., the text book of Mason (2001)), and equilibrium and immobilizing grasps (Rimon and Burdick 1998). Form closure, introduced by Reuleaux more than a century ago, is a way of defining the notion of "firm grip" of the object when friction is not taken into account (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. (When friction is taken into account, such grasps are often called force closure grasps.) Four point fingers are sufficient and often necessary to hold a polygonal object in form closure (Markenscoff et al. 1990). An equilibrium grasp is a grasp whose grasping fingers can exert wrenches (not all of them zero) through the grasping points to balance the object (Sudsang and Luewirawong 2003). Rimon and Burdick (1998) showed that there are equilibrium grasps that are not form closure grasps, but they nevertheless completely immobilize the object by preventing any *finite* motion of the object through curvature effects in configuration space (forming an immobilizing grasp). Therefore, every form closure grasp is an immobilizing grasp, and every immobilizing grasp is an equilibrium grasp but not necessarily vice versa. Czyzowicz et al. (1999) provided a necessary and sufficient geometric condition for a simple polygon to be immobilized by three frictionless point contacts. A review on grasping and related problems can be found in the paper by Bicchi and Kumar (2000).

Rimon and Blake (1996) introduced the notion of the *caging set* (also known as the *inescapable configuration space* (Sudsang et al. 1998; Sudsang 2000), and recently regularly referred to as the *capture region* (Pipattanasomporn and Sudsang 2006; Erickson et al. 2007; Pipattanasomporn et al. 2007)) of a hand as all hand configurations which maintain the object caged between the fingers. Moreover, using stratified Morse theory they showed that in a multi-finger one-parameter gripping system, the hand's configuration at which the cage is broken corresponds to a frictionless equilibrium grasp. Caging sets have been applied to a number of problems in manipula-

The International Journal of Robotics Research
Vol. 27, No. 11–12, November/December 2008, pp. 1308–1324
DOI: 10.1177/0278364908098485

tion such as grasping and in-hand manipulation (Sudsang et al. 1998; Sudsang 2000), mobile robot motion planning (Sudsang et al. 1999; Sudsang and Ponce 2000), and error-tolerant immobilizing grasp on a planar object (Rimon and Blake 1996; Davidson and Blake 1998a,b). To briefly elaborate, in manipulation caging grasps can be used when an object should be picked up and moved to some destination, and there is no need to really prevent all motions of the object as is done in the immobilizing grasps and fixturing; instead the fingers should just hold the object such that when moving the fingers as a rigid body the object moves with the fingers, although the object may have some freedom among the fingers. Despite the possible imprecision in finger placement with respect to the object, an appropriate caging grasp 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. Clearly convex[1] polygons and certain non-convex polygons cannot be caged with two fingers. Therefore, it is important to also consider caging grasps that involve more than two fingers.

1.1. Caging with Two Fingers

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 paper, however, we are interested in *all* possible two-finger caging grasps of simple polygonal objects. 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 grasps of two disk fingers. Pipattanasomporn and Sudsang (2006), independently of 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 grasp is a so-called squeezing caging grasp or a so-called stretching caging grasp without proving this non-obvious claim (see below). Moreover, generalizing the results to disk fingers was not discussed in their paper. There is very recent work performed by Pipattanasomporn et al. (2007) on computing only the set of all squeezing caging grasps 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 grasps to stretching caging grasps.

In Section 3 we present an algorithm that computes all caging grasps 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 grasp is caging. Our work on two-finger caging extends and improves the result of Pipattanasomporn and Sudsang (2006) by providing an algorithm for computing all caging grasps 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 proved that every caging placement is a squeezing or a stretching caging, and therefore established a relation between two-finger caging grasps and immobilizing grasps of polygons without parallel edges. A caging grasp is a squeezing caging grasp when the fingers cannot be moved to a grasp in which the fingers coincide while keeping the distance between the fingers at most equal to the distance of the given caging grasp. A caging grasp is a stretching caging grasp when the fingers cannot be moved to a grasp 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 of the given caging grasp.

1.2. 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 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 region* of the base fingers. Figure 1 shows two examples of caging region of a convex polygon and a nonconvex polygon. Before the work of Erickson et al. (2007) the previous complete algorithms had been limited to robotic systems with a single degree of freedom (Rimon and Blake 1996; Davidson and Blake 1998a) 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 et al. 1999; Sudsang 2000; Sudsang and Ponce 2000). Given the placements of the base fingers which are required to be on the boundary of a convex polygon, Erickson et al. (2007) provided the first complete algorithm for computing the caging region of the base fingers for such polygons in $O(n^6)$ time. However, the problem of computing the caging region when the base fingers are not necessarily on the boundary of the polygon, or when the

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

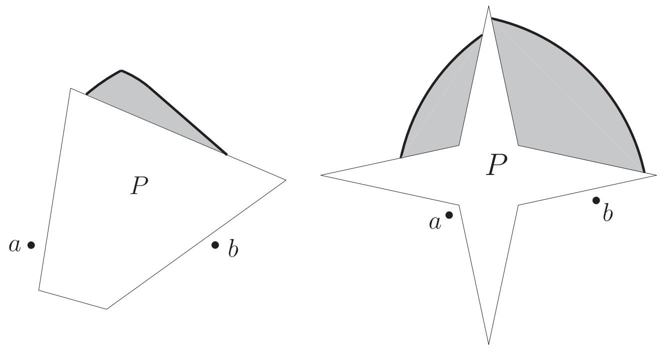

Fig. 1. Two examples of caging region of a convex and a non-convex polygon.

polygon is not convex, remained open and will be tackled in this paper.

In Section 4 we present a solution for computing the caging region of the base fingers for non-convex polygons for a given placements of the base fingers. It is shown that the sections of the boundary of these regions 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. (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 $O(n^6 \log^2 n)$.

2. Preliminaries

In this paper we address the problem of caging a polygon P with two disk fingers and also 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.

2.1. Notation

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 $\{a + b \mid a \in A, b \in B\}$.) 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 *grasp* is a tuple of points in the plane, where each point is the center of a finger. A *two-finger* grasp is a pair (a, b) of points in the plane. A *three-finger* grasp is a triple (a, b, c) of points in the plane where the points a, b and c are oriented counterclockwise.

In the first part of the paper, 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 $\mathcal{F} = F_r \times F_s$. Similarly, we define the *admissible space* for three fingers (which are disk fingers of radius zero) as $\mathcal{F} = F_0 \times F_0 \times F_0$.

2.2. Pseudo-trapezoidation

We use pseudo-trapezoidation to decompose the admissible space into constant-complexity cells such that each cell is adjacent to a constant number of cells. A pseudo-trapezoidation of F_r is a decomposition T_r of F_r into pseudo-trapezoids. 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. Pseudo-trapezoids clearly have constant complexity. We present an algorithm to decompose F_r in $O(n \log n)$ time into $O(n)$ pseudo-trapezoids such that every pseudo-trapezoid is adjacent to a constant number of pseudo-trapezoids. The set F_r is bounded by line segments and circular arcs of radius r.

We assume that the polygon P is a simple polygon in general position, which means that no two vertical lines through a vertex or tangent to a circular arc coincide. This is easily established by appropriately rotating the entire scene.

To compute the pseudo-trapezoidation, from every (segment and arc) endpoint and from every vertical tangent of an arc walls are extended in upward and/or downward direction until they hit another arc or edge. Every such point is called a *source point*. In every pseudo-trapezoid there are at most two vertical line segments, and hence at most two unique source points. Every source point can be in at most three pseudo-trapezoids. Therefore, every pseudo-trapezoid is adjacent to a constant number of pseudo-trapezoids. Moreover, the total number of pseudo-trapezoids is linear in the complexity of F_r. We can compute the pseudo-trapezoidation in $O(n \log n)$ time by using a sweep-line approach (de Berg et al. 1997).

Lemma 2.1. *It is possible to decompose F_r in $O(n \log n)$ time into $O(n)$ pseudo-trapezoids such that every pseudo-trapezoid is adjacent to a constant number of pseudo-trapezoids.*

The the pseudo-trapezoidation of F_r can also be used as a point-location data structure to find the pseudo-trapezoid that contains a given point in F_r in $O(\log n)$ time (de Berg et al. 1997).

3. Two-finger Caging

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 part we only consider two-finger grasps we generally omit the use of "two-finger". Recall that now we have $\mathcal{F} = F_r \times F_s$.

3.1. Definitions

Define $\mathcal{T} = T_r \times T_s$. $\mathcal{T}$ is a decomposition of the four-dimensional admissible space $\mathcal{F}$ into cells of constant complexity. A *δ-grasp* is a grasp for which the distance between the fingers is $\delta \in \mathbb{R}^+$. Let

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

which is the set of all δ-grasps. A *δ-min-grasp* is a grasp for which the distance between the fingers is at least $\delta \in \mathbb{R}^+$, and similarly a *δ-max-grasp* is a grasp 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-grasps. Similarly, let

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

which is the set of all δ-max-grasps. 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 grasp whose finger placements coincide. A grasp is called a *δ-squeezing caging grasp* 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 grasp whose finger placements are far from each other with respect to P. A grasp is called a *δ-stretching caging grasp* if and only if it is a point of an upper-bounded component of $\mathcal{F}_{\geq\delta}$.

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

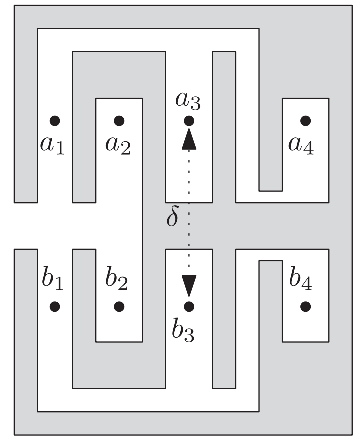

Fig. 2. Reachability notions and caging types.

In Figure 2 a shaded polygon and four δ-grasps (a_1, b_1), (a_2, b_2), (a_3, b_3) and (a_4, b_4) are displayed. The grasp (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 grasps are *δ-max-reachable* if they are δ-max-grasps and they lie in the same component of $\mathcal{F}_{\leq\delta}$, and *δ-min-reachable* if they are δ-min-grasps and they lie in the same component of $\mathcal{F}_{\geq\delta}$. When two grasps are δ-max-reachable, it is possible to move the two-finger hand between the grasps keeping the distance between the fingers at most δ. Similarly, when two grasps are δ-min-reachable it is possible to move the two-finger hand between the grasps keeping the distance between the fingers at least δ. In Figure 2 no two displayed grasps 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 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 grasp 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 grasp. We define the critical maximum distance of a grasp g to be ∞ when the finger placements of g lie in disjoint components of F_r and F_s. Every grasp 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.

The set $\mathcal{F}_{\geq\delta}$ also grows monotonically when δ is decreased from ∞ to 0. Similarly, the *critical minimum distance* of a grasp 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 grasp. 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 grasp 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.

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. Overview of the Approach

Let δ be the distance between the fingers of a two-finger hand. We prove in Section 3.3 that every caging δ-grasp is a δ-squeezing caging grasp or a δ-stretching caging grasp (or both); using this fact we establish a relation between caging grasps and immobilizing grasps in Section 3.4. In addition, we use this fact to report all caging grasps by reporting all squeezing caging grasps and stretching caging grasps 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 grasps can be computed similarly, we focus only on the computation of squeezing caging grasps.

In Section 3.5, 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 grasps. 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 grasps, 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 grasps in that component. We present the algorithm and its running time in Section 3.6. We also obtain a data structure, based on the connectivity graph, that can be used to determine whether a given grasp is caging.

3.3. Squeezing and Stretching Caging Grasps

In this section we prove that any caging δ-grasp is a δ-squeezing or δ-stretching caging grasp (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-grasp (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 grasps that are in the same connected component of $\mathcal{F}_{\geq\delta}$ as (a, b) or, equivalently, all grasps that are δ-min reachable from (a, b). Similarly, the set $\mathcal{F}_{\leq\delta}(a, b)$ contains all δ-max grasps that are in the same connected component of $\mathcal{F}_{\leq\delta}$ as (a, b) or, equivalently, all grasps 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., the text book of 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-grasp is δ-squeezing or δ-stretching caging grasps (or both). In brief, if this is not the case then there is a caging δ-grasp (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-grasp (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-grasp 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 $\beta_1 \simeq \beta_2$.

Lemma 3.1. *Let (a, b) and (a', b') be grasps 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. Consider the function $\Delta : [0,1]^2 \mapsto \mathbb{R}^+$ where $\Delta(u,v) = \|h_1(u,v) - h_2(u,v)\|$. Since both h_1 and h_2 are continuous, Δ is continuous as well. Moreover, we have $\Delta(0,v) \le \delta$ and $\Delta(1,v) \ge \delta$ for all $v \in [0,1]$, and $\Delta(u,0) = \Delta(u,1) = \delta$ for all $u \in [0,1]$.

To simplify the proof, we 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.

Let X be the set of points $(u,v) \in \square$ such that $\Delta(u,v) \ge \delta$, and let X_0 be the component of X containing point $(0,0)$. Now let Y denote the closure of $\square \setminus X_0$, and let Y_0 be the component of Y that contains the point $(0,0)$. The set Y_0 is homeomorphic to a disk; 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 $[-\epsilon, 1] \times [0,1]$. Thus, the set $Y_0 \cap [0,1]^2$ is a simple 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 $t \in [0,1]$. 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 3.1 to prove that any caging δ-grasp is a δ-squeezing or δ-stretching caging grasp.

Theorem 3.2. *Given a polygon and a caging δ-grasp, the grasp is a δ-squeezing caging grasp or a δ-stretching caging grasp.*

Proof. We prove this by contradiction. Assume that there is a caging grasp $(a,b) \in \mathcal{F}$ such that the distance between the fingers is δ, and it is neither a δ-squeezing nor a δ-stretching caging grasp. Consider a δ-grasp (a',b') at a remote location from P, which is neither a δ-squeezing caging grasp, nor a δ-stretching caging grasp. We show that both $\mathcal{F}_{\le\delta}(a,b)$ and $\mathcal{F}_{\ge\delta}(a,b)$ 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 grasp, and if exactly one of a and b is in a bounded component, then (a,b) is a δ-squeezing caging grasp, contradicting the assumption that (a,b) is neither a δ-squeezing nor a δ-stretching caging grasp.

The set $\mathcal{F}_{\le\delta}(a,b)$ contains (a',b') because of the following two reasons. First, $\mathcal{F}_{\le\delta}(a,b)$ contains a grasp where the finger placements coincide. Second, 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 $\mathcal{F}_{\ge\delta}(a,b)$ contains (a',b'). First, $\mathcal{F}_{\ge\delta}(a,b)$ contains a grasp whose fingers are far from each other with respect to P. Second, 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 $\mathcal{F}_{\le\delta}(a,b)$ 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 $\mathcal{F}_{\ge\delta}(a,b)$ 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'.

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 3.1 to prove the claim.

Let H_r be the convex hull[2] 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:

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

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

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

We easily verify that the pair of paths α_2' and β_2' are δ-min such that $\alpha_1 \simeq \alpha_2'$ and $\beta_1 \simeq \beta_2'$. □

3.4. Caging and Immobilization

We establish a relationship between caging grasps 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, 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 3.3. *Every squeezing minimal grasp of a polygon without parallel edges is an immobilizing grasp.*

Proof. Consider a squeezing minimal grasp (a, b). As 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 3.4. *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) proved that any grasp satisfying these conditions is an immobilizing grasp. □

Corollary 3.5. *Let P be a simple polygon without parallel edges. The grasp in a lower-bounded component of $\mathcal{F}_{\le\delta}$ that minimizes the distance between the fingers exists and it immobilizes P. Similarly, the grasp in an upper-bounded component of $\mathcal{F}_{\ge\delta}$ that maximizes the distance between the fingers exists and immobilizes P.*

Lemma 3.6. *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 every caging grasp is a squeezing caging grasp or a stretching caging grasp. Using Corollary 3.5 the claim follows. □

3.5. Connectivity Graph

In this section we define a graph called *δ-max connectivity graph* to represent $\mathcal{F}_{\le\delta}$ as a union of constant-complexity subcells. Recall from Section 3.1 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 $\mathcal{F}_{\le\delta}$ gives us a cell decomposition of $\mathcal{F}_{\le\delta}$, which we denote by $\mathcal{T}_{\le\delta}$. The intersection of $\mathcal{F}_{\le\delta}$ with a single subcell $\tau \in \mathcal{T}$ can be disconnected; we consider each connected component of $\mathcal{F}_{\le\delta} \cap \tau$ to be a distinct subcell in $\mathcal{T}_{\le\delta}$. We easily observe that the subcells of $\mathcal{T}_{\le\delta}$ also have constant complexity.

The *δ-max connectivity graph* $G_{\le\delta}$ is defined as follows. The vertices of $G_{\le\delta}$ are the four-dimensional subcells in $\mathcal{T}_{\le\delta}$. Two vertices are joined by an edge in $G_{\le\delta}$ 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 grasp whose finger placements coincide. The resulting set of representative grasps 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-grasps of the non-lower-bounded nodes are not δ-squeezing caging grasps, some of them may still be caging grasps. Recall that one δ-grasp 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. Similarly, we can find the pseudo-trapezoid of T_s that contains a given point in $O(\log n)$ time. Therefore, we can use the pseudo-trapezoidations T_r and T_s to find the corresponding node of a given δ-max-grasp in $G_{\leq\delta}$ each time in $O(\log n)$ time.

Lemma 3.7. *Given a polygon P and a distance δ, it is possible to compute $G_{\leq\delta}$ and find out whether each node is contained in a lower-bounded component in $O(n^2)$ time.*

Lemma 3.8. *After $O(n^2)$ preprocessing time, we can determine in $O(\log n)$ time whether a given two-finger δ-max-grasp is a δ-squeezing caging grasp.*

3.6. Two Disk-finger Caging Algorithm

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

Briefly, we consider all values of δ, and report the lower-bounded components of $\mathcal{F}_{\leq\delta}$ as the squeezing caging grasps. 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 grasps.

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 grasps 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 between 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 distance 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 a *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 grasp 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 all nodes that ever existed in $G_{\leq\delta}$. We label each node in $\mathcal{G}$ either lower-bounded or not lower-bounded. We 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 grasp 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 grasps for which δ is their critical maximum distance. Look at Figure 3 for some of the critical maximum distances that lead to a *maximal report*. This operation consists of three parts: (1) we report the corresponding four-dimensional cells (excluding the grasps that the distance between the fingers is less than $r+s$, i.e. intersecting each other) of all of 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 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 $O(n^2)$.

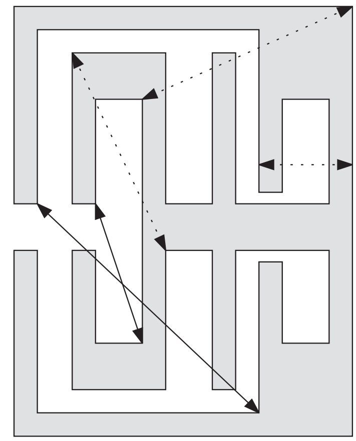

Fig. 3. Three critical maximum distances displayed with dotted arrows and two critical minimum distances displayed with solid arrows.

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 grasps for which exactly one finger placement is inside a bounded component of F_r or F_s, respectively. For all of 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. As 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 3.9. *Given a polygon with n edges and two disk fingers, it is possible to report all squeezing caging grasps in* $O(n^2 \log n)$ *time.*

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

Theorem 3.10. *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 δ-grasp is a δ-squeezing caging grasp.*

Proof. After reporting all squeezing caging grasps, we consider the final graph $\mathcal{G}$, and T_r and T_s. We use T_r and T_s to find the pair of pseudo-trapezoids that contain the finger placements, and then to find the node of $\mathcal{G}$ associated with the grasp in $O(\log n)$ time. We compare the critical maximum distance m assigned to the node with δ. We report that the grasp is a squeezing caging grasp if m is larger than δ; otherwise we report that the grasp is not a squeezing caging grasp. 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 to Theorems 3.10 and 3.9 can be obtained for stretching caging grasps. The two results together lead to the following main results of this section.

Theorem 3.11. *Given a polygon with n edges and two disk fingers, it is possible to report all caging grasps in $O(n^2 \log n)$ time.*

Proof. By Theorem 3.2 every caging grasp is squeezing caging or stretching caging. By Theorem 3.9 we can report all squeezing caging grasps in $O(n^2 \log n)$ time. We can report all stretching caging grasps similarly in $O(n^2 \log n)$ time. By reporting all squeezing caging grasps and all stretching caging grasps separately, we can report all caging grasps in $O(n^2 \log n)$ time.

Theorem 3.12. *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 δ-grasp is a caging grasp.*

Proof. If the δ-grasp is neither a δ-squeezing caging grasp nor a δ-stretching caging grasp, then according to Theorem 3.2 it is not a caging grasp. According to Theorem 3.10, it is possible to check in $O(\log n)$ time whether a given δ-grasp is a δ-squeezing or a δ-stretching caging grasp 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 grasps equals the union of the set of squeezing caging grasps and the set of stretching caging grasps. As we know that the complexity of a single subcell of $\mathcal{T}$ as well as the set of squeezing caging grasps and the set of stretching caging grasps contained in that subcell are constant, we can compute this union per subcell to obtain the set of all caging grasps for each subcell. Therefore, it is clear that the total complexity of the set of all caging grasps is $O(n^2)$.

4. Three-finger Caging

This part of the paper addresses the problem of caging a polygon P with three point fingers. In this section we assume that a grasp is a three-finger grasp.

4.1. Definitions

Recall from Section 2 that $\mathcal{F} = F_0 \times F_0 \times F_0$ is the admissible space of three point fingers. Define $\mathcal{T} = T_0 \times T_0 \times T_0$. $\mathcal{T}$ is a decomposition of the six-dimensional admissible space $\mathcal{F}$ into cells of constant complexity.

If one grasp can be transformed into another by a rigid transformation, we say that those two grasps have the same *shape*. Our goal is to build a description of all caging grasps 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 grasp (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 grasp (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 grasp, we separate shape from placement when we investigate the sets we define later.

Let

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

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-grasp* is a member of $\mathcal{F}_\delta$. Two δ-grasps are *δ-reachable* if both of them lie in the same connected component of $\mathcal{F}_\delta$. When two grasps are *δ-reachable* it is possible to move the hand between the grasps keeping the shape of the hand fixed (during which the polygon is also fixed).

Let $C : \mathcal{F} \mapsto \{\text{False}, \text{True}\}$ be a predicate that determines whether a given grasp cages P. Consider a given placement $(a, b) \in F_0 \times F_0$ of the base fingers. Let $\mathcal{C}(a, b) \subset F_0$ denote the set of all placements of the third finger that together with a and b form a grasp 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 $\mathcal{C}(a, b)$ consists of one or more connected components in F_0 and is referred to as the *caging region* of (a, b) and its boundary, that is, $\partial\mathcal{C}(a,b)$, is referred to as the *caging boundary* of (a,b). Formally,

$$\mathcal{C}(a,b) = \{c \in F_0 \mid C(a,b,c)\}.$$

Given P and a placement $(a,b) \in F_0 \times F_0$ of the base fingers, we present an algorithm to report $\mathcal{C}(a,b)$. Let $\mathcal{K}(a,b)$, which we refer to as the *caging curve*, denote the part of the boundary of the caging region that does not belong to the boundary of P; or formally $\mathcal{K}(a,b) = \partial\mathcal{C}(a,b) \setminus \partial P$. In Figure 1 two examples of the caging region of a convex polygon and a non-convex polygon are displayed. In each case the light gray area is the caging region $\mathcal{C}(a,b)$ and the caging curve $\mathcal{K}(a,b)$ is displayed with bold curves.

4.2. Overview of the Approach

In this section we solve the problem of computing the caging region for a given placement $(a,b) \in F_0 \times F_0$ 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 Section 4.3 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 arrangement of a set X of two-dimensional curves is the set of maximally connected zero-dimensional, one-dimensional and two-dimensional subsets induced by the curves of X not intersecting any of the subsets. The fact that the equilibrium curves and the polygon boundary together form the boundary of the caging region of the base fingers implies that all points inside a single cell of the arrangement (i.e. a maximally connected two-dimensional subset of the arrangement not containing any point on a curve) of the equilibrium curves and the polygon boundary are either caging or non-caging placements of the third finger. We report the caging region by placing the third finger in every cell of the mentioned arrangement to find the cells consisting of caging grasps.

To find the caging status of a cell we consider the connected components of $\mathcal{F}_\delta$. Since F_0 has exactly one component, there is exactly one unbounded component in $\mathcal{F}_\delta$ that corresponds to non-caging δ-grasps. 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 $\mathcal{F}$ into cells of constant complexity.) In Section 4.4, based on $\mathcal{T}$ we define and construct a graph, called a *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 $\mathcal{F}_\delta$. In this way, the computation of the caging grasps boils down to identifying the connected components of the connectivity graph. The complete algorithm and the running time analysis is explained in Section 4.5.

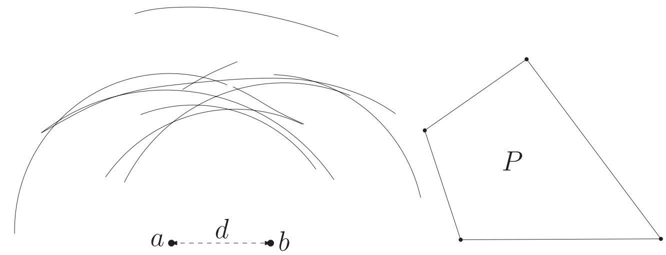

Fig. 4. A number of equilibrium curves of the polygon P.

4.3. Equilibrium Curves

In this section 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 Section 4.3.1 we enumerate all possible two- and three-finger equilibrium grasps involving two fingers with distance equal to d.

Let $\mathbb{T}^{(a,b)}_{(p,q)}[X]$ 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

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

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,

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

Here $\mathcal{E}_P(a,b)$ 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 4 for an example.

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

Theorem 4.1. *We have $\mathcal{K}(a, b) \subset \mathcal{E}_P(a, b)$.*

Proof. Rimon and Blake (1996, Proposition 3.3) 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. 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 $\mathcal{K}(a, b) \subset \mathcal{E}_P(a, b)$, consider the intersection point c of an arbitrary line l and $\mathcal{K}(a, b)$. 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 grasp (a, b, c) corresponds to an equilibrium grasp (involving two fingers with distance equal to d). □

4.3.1. Types of Equilibrium Grasps and Curves

In this section 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 a 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 also fixed.

   b. *Two vertices*. Since the distance between the two vertices is fixed, the distance between the third finger and the involved base finger is also fixed.

   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 also fixed.

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 Section 4.3.3; 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 Section 4.3.2; 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 $\mathcal{E}_P(a, b)$. 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 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-hand 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 $\mathcal{E}_P(a, b)$.

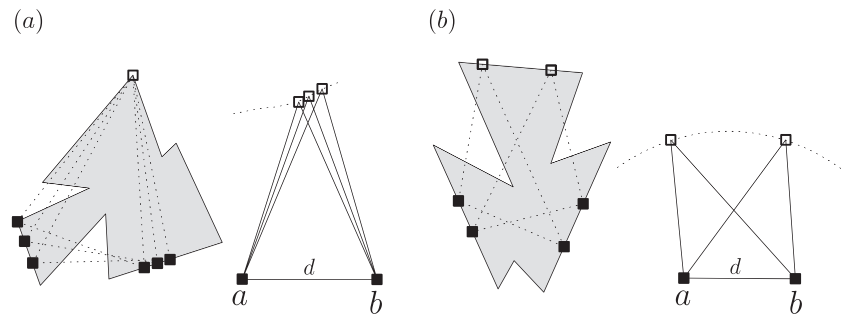

Fig. 5. Two loci are displayed with dotted curves at the right-hand side of two shaded polygons in which the filled boxes represent the base fingers and dotted triangles represent equilibrium grasps (see the text for the explanation).

Theorem 4.2. *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.*

4.3.2. 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 section we show that the locus of the point associated with the third finger describes a circular arc.

Consider the triangle $\triangle pqr$ in Figure 6 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.

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 the 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

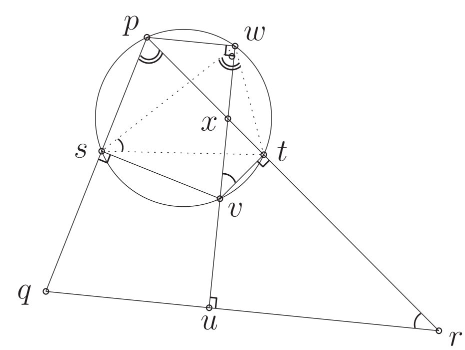

Fig. 6. Proof for the locus of the third finger when the three fingers slide on edges.

$\angle prq$ 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 also fixed. 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.

4.3.3. 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 8. 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 section we show that, as we draw all of these triangles with the base fingers on a pair of fixed points at a and b (with distance equal to d) on

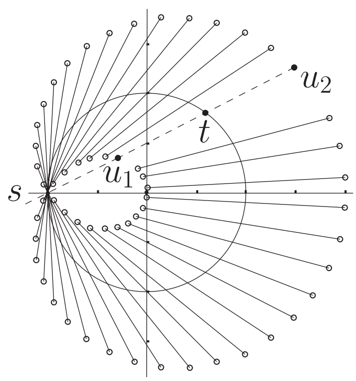

Fig. 7. Limaçon of Pascal curve.

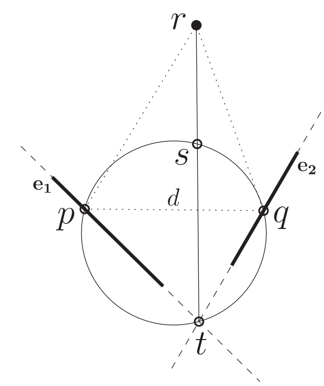

Fig. 8. Proof for the locus of the third finger when the third finger is at a vertex and the base fingers slide on two edges.

the 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 7. 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. 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 8. 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 the 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.

4.4. Connectivity Graph

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

Recall from Section 4.1 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 grasps that are represented by nodes in the unbounded component of G_δ are non-caging grasps. More importantly, all grasps that are represented by nodes in other components of G_δ *are* caging grasps. To compute the unbounded component of G_δ we consider a non-caging δ-grasp and its corresponding node in G_δ. (We can easily find such a node by considering a δ-grasp 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 4.3. *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_0, the total number of edges is linear in the total number of nodes. Therefore, if there are $O(n)$ trapezoids in T_0, there will be $O(n^3)$ nodes and edges in G_δ. □

We can find the trapezoid of T_0 that contains a given point in $O(\log n)$ time. Therefore, we can use the trapezoidation T_0

to find the corresponding node of a given δ-grasp in G_δ each time in $O(\log n)$ time.

Lemma 4.4. *After $O(n^3)$ preprocessing time, we can determine, in $O(\log n)$ time, whether a given three-finger δ-grasp is a caging grasp.*

4.5. 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 $(a, b) \in F_0 \times F_0$ 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 region 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 region is the entire space F_0. Using the result of two-finger caging, we can check this case in $O(n^2)$ time.

Recall that the arrangement of a set X of two-dimensional curves is the set of maximally connected zero-dimensional, one-dimensional and two-dimensional subsets induced by the curves of X, which we will display by $\mathcal{A}(X)$. The arrangement $\mathcal{A}(\mathcal{E}_P(a, b) \setminus P)$ 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 grasps 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 three-finger caging grasps; otherwise it is called a *non-caging cell*.

We report the caging region 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-connectivity 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 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_0, 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 of 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 also constant, the total number of nodes associated with τ in G_δ is constant. Moreover, given a grasp 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_0 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 the addition or removal of a 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 number of changes. To maintain the connected components of the connectivity graph efficiently we use a graph-based data structure called a 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 using the trapezoidation T_0. 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 4.5. *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_0 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 a 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. Conclusion

We have presented complete algorithms for computing two disk-finger and three point-finger caging grasps. In both cases 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 so that the query includes the radii of the disk fingers is interesting too. These two issues are the main interesting problems that we would like to pursue as a future work on two-finger caging.

For the three-finger case, extending the results to disk-shaped fingers seems straightforward. Although the curves of equilibrium grasps become more complicated, their algebraic degrees remain constant. We intend to implement the algorithms to gain more insight into the shapes of caging regions and their combinatorial complexities with the purpose of improving the worst-case running time of our algorithm. In addition, we would like to consider the three-finger caging query as well which asks whether or not a given query grasp is caging. 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 also interesting.

Finally, extending our results to three dimensions seems challenging, because of the problem of decomposing the admissible space into a few simple cells. For the two-finger caging, 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. Hence, we will look for alternative ways to tackle three-dimensional caging problems.

Acknowledgment

We like to thank the anonymous reviewers for their constructive comments.

M. Vahedi is supported by the Netherlands Organisation for Scientific Research (NWO). A.F. van der Stappen is partially supported by the Dutch BSIK/BRICKS-project.

References

Bicchi, A. and Kumar, V. (2000). Robotic grasping and contact: a review. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 348–353.

Czyzowicz, J., Stojmenovic, I. and Urrutia, J. (1999). Immobilizing a shape. *International Journal of Computational Geometry and Applications*, **9**(2): 181–206.

Davidson, C. and Blake, A. (1998a). Caging planar objects with a three-finger one-parameter gripper. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2722–2727.

Davidson, C. and Blake, A. (1998b). Error-tolerant visual planning of planar grasp. *Proceedings of ICCV*, pp. 911–916.

de Berg, M., van Kreveld, M., Overmars, M. and Schwarzkopf, O. (1997). *Computational Geometry: Algorithms and Applications*. Berlin, Springer.

Erickson, J., Thite, S., Rothganger, F. and Ponce, J. (2007). Capturing a convex object with three discs. *IEEE Transactions on Robotics*, **23**(6): 1133–1140.

Gopalakrishnan, K. and Goldberg, K. (2002). Gripping parts at concave vertices. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1590–1596.

Henzinger, M. R. and King, V. (1999). Randomized fully dynamic graph algorithms with polylogarithmic time per operation. *Journal of the ACM*, **46**(4): 502–516.

Holm, J., de Lichtenberg, K. and Thorup, M. (1998). Polylogarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC'98)*. New York, ACM Press, pp. 79–89.

Kuperberg, W. (1990). Problems on polytopes and convex sets. *Proceedings of the DIMACS Workshop on Polytopes*, pp. 584–589.

Markenscoff, X., Ni, L. and Papadimitriou, C. H. (1990). The geometry of grasping. *The International Journal of Robotics Research*, **9**(1): 61–74.

Mason, M. (2001). *Mechanics of Robotic Manipulation* (*Intelligent Robotics and Autonomous Agents Series*). Cambridge, MA, MIT Press.

Munkres, J. R. (1984). *Elementes of Algebraic Topology*. Readin, MA, Addison-Wesley.

Pipattanasomporn, P. and Sudsang, A. (2006). Two-finger caging of concave polygon. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2137–2142.

Pipattanasomporn, P., Vongmasa, P. and Sudsang, A. (2007). Two-finger squeezing caging of polygonal and polyhedral object. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 205–210.

Rimon, E. and Blake, A. (1996). Caging 2D bodies by 1-parameter two-fingered gripping systems. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1458–1464.

Rimon, E. and Burdick, J. W. (1998). Mobility of bodies in contact—part I: a 2nd-order mobility index for multiple-finger grasps. *IEEE Transactions on Robotics and Automation*, **14**(5): 696–717.

Sudsang, A. (2000). Grasping and in-hand manipulation: geometry and algorithms. *Algorithmica*, **26**(3–4): 466–493.

Sudsang, A. and Luewirawong, T. (2003). Capturing a concave polygon with two disc-shaped fingers. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1121–1126.

Sudsang, A. and Ponce, J. (2000). A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1068–1075.

Sudsang, A., Ponce, J., Hyman, M. and Kriegman, D. J. (1999). On manipulating polygonal objects with three 2-DOF robots in the plane. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2227–2234.

Sudsang, A., Ponce, J. and Srinivasa, N. (1998). Grasping and in-hand manipulation: experiments with a reconfigurable gripper. *Advanced Robotics*, **12**(5): 509–533.

Wells, D. (1991). *The Penguin Dictionary of Curious and Interesting Geometry*. London, Penguin.