Capturing a Convex Object with Three Discs

Jeff Erickson Shripad Thite Fred Rothganger Jean Ponce
Department of Computer Science
University of Illinois at Urbana-Champaign

Abstract: This paper addresses the problem of capturing an arbitrary convex object P in the plane with three congruent disc-shaped robots. Given two stationary robots in contact with P, we characterize the set of positions of a third robot that prevent P from escaping to infinity and show that the computation of this so-called capture region reduces to the resolution of a visibility problem. We present two algorithms for solving this problem and computing the capture region when P is a polygon and the robots are points (zero-radius discs). The first algorithm is exact and has polynomial-time complexity. The second one uses simple hidden-surface removal techniques from computer graphics to output an arbitrarily accurate approximation of the capture region; it has been implemented and examples are presented.

1 Introduction

This paper addresses the problem of capturing a convex object in the plane with three congruent disc-shaped robots. In practice, these robots may be mobile platforms, the fingertips of a gripper, the locators of a modular fixturing system, or the pins of a reconfigurable parts feeder. Applications include robotic grasping, sensorless manipulation, and flexible automation.

Capture regions are related to the notions of force/form closure and immobilizing grasps from kinematics and robotics: For a hand to hold an object securely, it should be capable of preventing any motion due to external forces and torques. A grasp that prevents any infinitesimal motion of the object is said to achieve *form closure*, and it is said to achieve *force closure* when it can balance any external force and torque. Force and form closure are dual notions from classical kinematics [2, 16] and, as noted in [10, 11] for example, force closure implies form closure and vice versa. They are the traditional theoretical basis for grasp planning algorithms (see, for example, [6, 9, 11, 12]).

Recently, Rimon, and Burdick have introduced the notion of *second-order immobility* [14, 15] and shown that certain equilibrium grasps of a part which do not achieve form closure effectively prevent any *finite* motion of this part: In effect, an object is immobile when it lies at an isolated collision-free point of its configuration space. Sudsang, Ponce, and Srinivasa [19] introduced the notion of *capture region* of a robotic system as the set of configurations of this system that may not immobilize the object being manipulated but prevent it from escaping to infinity (see [4, 8, 13] for related work): An object is captured when it lies in a compact valid region of its configuration space.

Capture regions have been applied to a number of problems in sensorless manipulation, including grasping and in-hand manipulation [13, 20, 21], mobile robot motion planning [18, 22], parts feeding [3, 4], and stable pose computation [8]. This paper presents (to the best of our knowledge) the first algorithm for computing the exact capture region associated with a robotic system with multiple degrees of freedom (dof): Previous exact algorithms have been limited to static situations [3, 4, 8] or to robotic systems with a single dof [13], whereas efforts to tackle robotic systems with multiple dof have been limited to approximate algorithms that assume that each robot can only interact with a single object edge, and output relatively small capture regions [18, 20, 21, 22].

This paper proposes an approach that takes into account the entire boundary of a convex object and will (in general) output much larger regions. We focus on the case of two fixed robots a and b in contact with a convex object P in its initial configuration, and characterize the set of positions of a third robot c that prevent P from escaping to infinity. We show that the computation of this capture region reduces to the resolution of a visibility problem. We present two algorithms for solving this problem and computing the capture region when P is a polygon and the robots are points (zero-radius discs). The first algorithm is exact and has polynomial-time complexity. The second one uses simple hidden-surface removal techniques from computer graphics to output an arbitrarily accurate approximation of the capture region; it has been implemented and examples are presented.

We assume without loss of generality that robot a is at the origin of the coordinate system; robot b is on the positive x axis; the initial orientation of the object makes a zero angle with the positive x axis; and a, b, and c are labeled in clockwise order. All angles are measured with respect to the positive x axis, with positive counterclockwise angles.

2 Canonical Motions

We assume in this section that the position of c is fixed, and show that when P can escape from its initial configuration by a rigid motion, it can also escape by a *canonical motion*. This will allow us to characterize capture regions in a simple fashion in the next section.

Without loss of generality, we can assume that the robots are points (zero-radius discs) by replacing P by its Minkowski sum with a disc congruent to the three robots. Here onwards we will use P to denote the result of this Minkowski sum which is also convex. We work in the configuration space $\mathbb{R}^2 \times S^1$ of possible positions (x, y) and orientations θ of P. We will abuse notation in the sequel and also designate by a, b, or c the point in any $\theta =$ constant plane (or θ-*slice*) of $\mathbb{R}^2 \times S^1$ where the vertical line erected at the corresponding robot position intersects that plane.

Each robot defines in $\mathbb{R}^2 \times S^1$ an obstacle consisting of a twisted column whose cross-sections are rotated copies of P. For example, robot a defines the obstacle $A = \cup_\theta (a \oplus P_{\theta+\pi}) \times \{\theta\}$, where $\oplus$ denotes the Minkowski sum operator and P_θ denotes P rotated by some angle θ about its fixed reference point. We similarly define obstacles B and C corresponding to the robots b and c. The complement of the union of these three obstacles is called the *free space*. Obstacles are closed sets and free space is open. *Contact space* is defined as the set of configurations that belong to the boundary of one of the obstacles but not to the interior of any of them. Finally, *valid space* is the union of free and contact space. A particular configuration of P is *captured* if and only if the corresponding point p lies in a compact component of valid space.

It is convenient to visualize rigid motions of the object by separating the translation component from the rotation component, viewing just a single θ-slice of configuration space at a time (Figure 1). For any angle θ, the configuration plane contains three obstacles $A_\theta = a \oplus P_{\theta+\pi}$, $B_\theta = b \oplus P_{\theta+\pi}$, and $C_\theta = c \oplus P_{\theta+\pi}$. As we increase θ, the obstacles A_θ, B_θ, and C_θ rotate counterclockwise around the corresponding points a, b, and c.

We define a *pocket* of a θ-slice as any compact component of its valid part. Since A_θ, B_θ, and C_θ are convex, there is at most one pocket (see Figure 1 for an example), and a necessary and sufficient condition for its existence is that all obstacles intersect pairwise, but the interior of $A_\theta \cap B_\theta \cap C_\theta$ be empty. When this condition is satisfied, we denote by V_θ the corresponding pocket. When a pocket does not exist (or when the initial object configuration does not belong to it), the component of valid space that contains the initial configuration is unbounded, and the object can obviously escape by a pure translation.

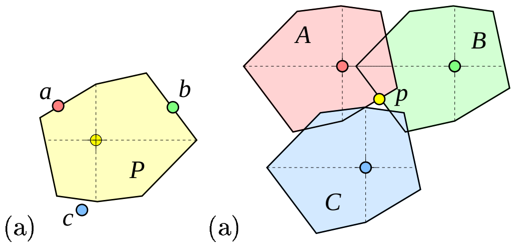

Figure 1. A convex object P in contact with the robots a and b in (a) workspace and (b) configuration space. Note that p lies in a pocket in this case. In all examples shown in this paper, the object is a polygon and the robots have zero radius, but the discussion in this section and the next one applies to arbitrary convex planar objects and robots with nonzero radius. The θ subscripts have been omitted for readability.

Let us assume from now on that a pocket exists for $\theta = 0$ *and* the initial configuration belongs to this pocket. We define a *canonical motion* as follows (Figure 2):

> Monotonically increase or decrease θ while maintaining contact between P and the robots a and b until (1) two of the obstacles no longer intersect, allowing the object to escape by a pure translation, or (2) it is blocked for further rotation by a contact with c, or (3) it returns to its original orientation.

A *canonical escape motion* is defined as a canonical motion ending by an escape to infinity with condition (1) being satisfied. Suppose that such a motion does not exist. By definition, the three obstacles intersect pairwise throughout the two (clockwise and counterclockwise) canonical motions. Suppose condition (2) is satisfied and the object's rotation is blocked by c at orientation ψ during its counterclockwise canonical motion. Since the initial configuration belongs to a pocket and the motion is continuous, it is clear that $A_\theta \cap B_\theta \cap C_\theta$ is empty when $0 \le \theta < \psi$, equal to a single point (the blocking configuration) when $\theta = \psi$, and has a nonempty interior for some interval $[\psi, \psi']$. In particular, a pocket exists for $\theta \in [0, \psi]$, and there is no pocket in the range (ψ, ψ'). It is easy to see that if the

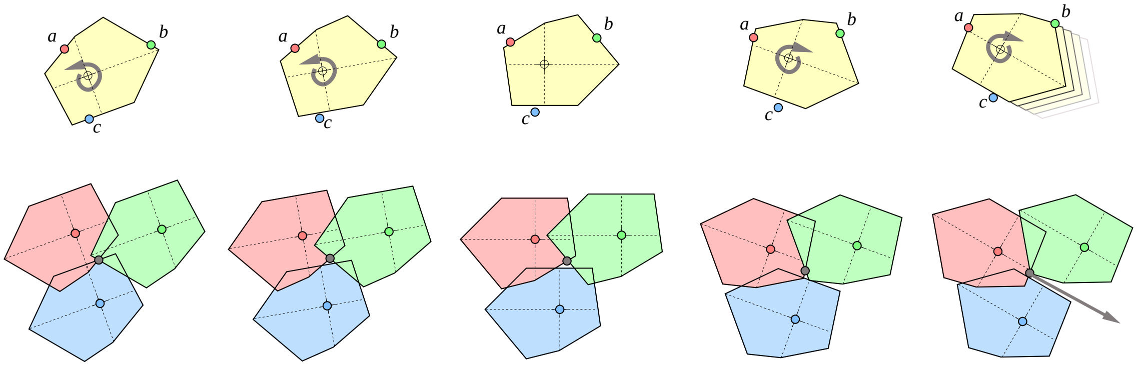

Figure 2. Canonical motions, with the corresponding changes in the configuration plane. Center to left: counterclockwise turning stopped by a triple contact. Center to right: clockwise turning ending in escape through bc.

object is blocked by a counterclockwise rotation at orientation ψ, it must also be blocked by a clockwise one for some orientation ϕ (and vice versa). We can apply the same line of reasoning as before to clockwise rotations in the range $[\phi, 0]$, and it follows that the stack of pockets $V = \cup_{\theta=\phi}^{\psi} V_\theta$ is a compact connected component of the valid configuration space. When there is no blocking motion and the canonical motions end with condition (3) satisfied, contiguous pockets exist at every orientation, and the stack $\cup_{\theta=0}^{2\pi} V_\theta$ defines a compact component of $\mathbb{R}^2 \times S^1$. We have proven the following lemma.

Lemma 2.1. *P can escape if and only if it can escape by some canonical motion.*

Our proof allows us to define three types of escape, depending on which two of the configuration obstacles A_θ, B_θ, and C_θ do not overlap at the object's final orientation θ. If A_θ and B_θ are disjoint, we say that *P escapes through ab*; we define *escape through ac* and *escape through bc* analogously.

3 Characterizing the Capture Region

The robot locations that capture P are those that prevent counterclockwise and clockwise canonical escape motions as well as escape by pure translation at $\theta = 0$. Let X^+, X^- and X^0 denote the corresponding regions of the plane. We characterize below X^0 and X^+ as the projections in the x, y plane of simple configuration space surfaces, and show that $X^+ \subset X^0$. The set X^- can be characterized in a symmetric way, and the capture region is $X = X^0 \cap X^+ \cap X^- = X^+ \cap X^-$.

3.1 Preventing Escape by Translation

Recall from the previous section that the configuration obstacles are $A_\theta = a \oplus P_{\theta+\pi}$, $B_\theta = b \oplus P_{\theta+\pi}$, and $C_\theta = c \oplus P_{\theta+\pi}$. As noted before, a necessary and sufficient for the existence of a pocket is that the three obstacles intersect pairwise but their overall intersection has an empty interior. To operationalize this condition, we introduce the *second-order* configuration obstacles $A^*_\theta = A_\theta \oplus P_\theta = a \oplus P_\theta \oplus P_{\theta+\pi}$ and $B^*_\theta = B_\theta \oplus P_\theta = b \oplus P_\theta \oplus P_{\theta+\pi}$ (Figure 3). It is clear that c is in A^*_θ (resp. B^*_θ) when A_θ and C_θ (resp. B_θ and C_θ) intersect. It is also clear that A^*_θ (resp. B^*_θ) is the area swept by P_θ while maintaining contact with a (resp. b).

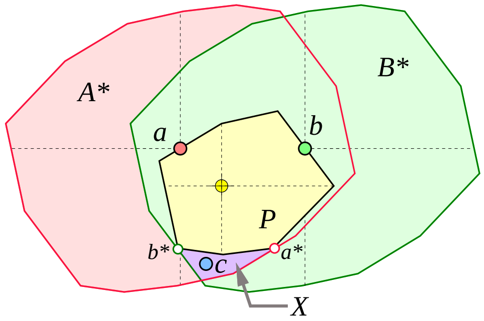

Figure 3. The A^*_θ, B^*_θ, P^*_θ, and X^θ regions at $\theta = 0$. The subscripts and superscripts have been omitted for readability.

Let us assume that A_θ and B_θ intersect (otherwise the object can escape by a translation through ab), and denote by P^*_θ the placement of P_θ that maintains contact with a and b during the rotational part of the canonical motion. The *tangent lines* are the tangents to the boundary of P^*_θ at the points where it touches a and b. By construction, P^*_θ must lie in the intersection of A^*_θ and B^*_θ and be tangent to the boundary of $A^*_\theta \cap B^*_\theta$ in the two points a^*_θ and b^*_θ of ∂P^*_θ that are the furthest away from the contact lines (see Figure 3; this follows directly from the properties of A^*_θ and B^*_θ mentioned earlier).

The set $A^*_\theta \cap B^*_\theta \setminus P^*_\theta$ is divided by the points a^*_θ

and b^*_θ into two connected components, one below the line $a^*_\theta b^*_\theta$, call it X^θ, and one above, call it Y^θ. We have the following result.

Lemma 3.1. *P is unable to escape by translation at orientation θ if and only if A_θ and B_θ intersect, and c is in X^θ.*

A formal proof is omitted for lack of space. Informally, the conditions of Lemma 3.1 guarantee that the three obstacles intersect pairwise, that P in its initial configuration does not collide with c, and that it cannot escape by pure translation along one of the contact edges, which it could do if c were in the upper component Y^θ of $A^*_\theta \cap B^*_\theta \setminus P^*_\theta$.

Since we have assumed that P is in contact with a and b in its initial configuration, A_0 and B_0 must intersect, and the set of positions of c that prevent escape by pure translation at $\theta = 0$ is simply X^0.

3.2 Preventing Canonical Escape Motions

We denote by $\mathcal{A}^*$ and $\mathcal{B}^*$ the surfaces respectively swept by the boundaries of A^*_θ and B^*_θ as θ varies between 0 and 2π (Figure 4). The *escape angle* is defined as the first orientation (when one exists) for which A_θ and B_θ no longer intersect. We denote by $\mathcal{P}^*$ the surface swept by the boundary of P^*_θ as θ varies between 0 and the escape angle σ if it exists (P^*_θ is not defined for $\theta > \sigma$ in this case), and between 0 and 2π otherwise. Finally, we denote by Π and Π' the two planes respectively defined by $\theta = 2\pi$ and $\theta = \sigma$ if an escape angle σ exists and $\theta = 2\pi + 1$ otherwise.

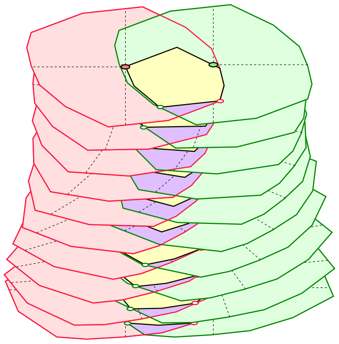

Figure 4. The volumes bounded by the surfaces $\mathcal{A}^*$, $\mathcal{B}^*$, and $\mathcal{P}^*$. We only show a finite number of slices to reveal some of the internal structure of these volumes.

We identify the plane of possible positions for c with the plane Π_0 defined by $\theta = 0$ in the object configuration space, with the θ direction serving as "vertical" axis. We have the following result.

Lemma 3.2. *The counterclockwise capture region X^+ consists of the points c in X^0 such that the vertical half-line erected above c in configuration space intersects either $\mathcal{P}^*$ or Π before it intersects $\mathcal{A}^*$, $\mathcal{B}^*$, or Π'.*

Proof: Let Δ_c denote the vertical half-line erected above c. First note that Δ_c will obviously always intersect one of the five surfaces of interest for some $\theta \leq 2\pi + 1$. Let θ_0 be the value of θ where the first intersection occurs. There are five cases, depending on which surface is intersected first. If this surface is

1. $\mathcal{A}^*$: the obstacles A_θ and C_θ stop intersecting at $\theta = \theta_0$, with the object free to escape by translation through ac;

2. $\mathcal{B}^*$: the obstacles B_θ and C_θ stop intersecting at $\theta = \theta_0$, with the object free to escape by translation through bc;

3. Π': θ_0 is the escape angle; the obstacles A_θ and B_θ stop intersecting at $\theta = \theta_0$, with the object free to escape by translation through ab;

4. $\mathcal{P}^*$: the rotation is blocked in θ_0, preventing further rotation before any escape by translation can occur;

5. Π: the object is back to its original configuration, and a canonical escape motion does not exist.

The lemma immediately follows. □

Note: there is no need to check that c remains in the lower component of $A^*_\theta \cap B^*_\theta \setminus P^*_\theta$ since Δ_c would have to cross $\mathcal{A}^*$, $\mathcal{B}^*$, or $\mathcal{P}^*$ first to move to its upper component.

As mentioned earlier, the set X^- can be characterized in a symmetric way. Since $X^+ \subset X^0$, we finally have $X = X^0 \cap X^+ \cap X^- = X^+ \cap X^-$.

4 Computing the Capture Region

Lemma 3.2 allows us to reduce the computation of the capture region to the resolution of a visibility problem. We now present two algorithms for solving this problem when P is a polygon and the three robots are points (zero-radius discs). The first algorithm is exact and runs in polynomial time. The second one returns an approximation of the capture region computed efficiently with hidden-surface removal techniques from computer graphics. Both algorithms require the computation of a set of *critical orientations* to compute an appropriate description of $\mathcal{P}^*$, as described in the next section.

4.1 Critical Orientations

The polygon $Q = P \oplus (-P)$ has $2n$ edges and can be constructed in linear time. The two obstacles $\mathcal{A}^*$ and $\mathcal{B}^*$ are obtained by translating Q so its reference point coincides with a or b, then sweeping it along a helicoidal trajectory. The case of $\mathcal{P}^*$ is more complicated since the polygon must remain in contact with a and b throughout the rotational part of the canonical motion. The surface of $\mathcal{P}^*$ is continuous and piecewise-smooth, with orientation discontinuities occurring at *critical orientations*, where the contact edges change, and a vertex of A_θ intersect an edge of B_θ (or vice versa).

Lemma 4.1. *The critical orientations and the counterclockwise escape angle σ (if it exists) can be computed in $O(p)$ time, where p is the number of critical orientations, which is itself $O(n^2)$.*

Proof: To begin, we compute either of the two intersection points of the boundaries of the initial configuration obstacles A_0 and B_0, in $O(n)$ time. We then maintain the pair of intersecting edges of A_θ and B_θ as θ increases continuously, using a variant of the *rotating calipers* algorithm of Toussaint [7, 23].

The edge pair changes exactly when an endpoint of one edge crosses the other edge (primary event). Thus, at any orientation, there are only four possible events at which the crossing edge pair can change next. We can predict the orientation of each event in $O(1)$ time, so we can update the edge pair in $O(1)$ time per event. At each event, we can also detect in constant time whether the obstacles still intersect at all. The algorithm halts either when we discover that A_θ and B_θ are disjoint (in which case $\theta = \sigma$), or when we reach $\theta = 2\pi$ (in which case there is no escape angle).

The running time of the algorithm is $O(p)$, where p is the number of critical orientations found by the algorithm. Since the polygons A_θ and B_θ are rotating at the same rate, a single edge pair can be involved in at most a constant number of events during one full rotation. Thus, $p = O(n^2)$. □

Surprisingly, there are convex polygons P and points a and b for which this algorithm must process $\Omega(n^2)$ events (see Appendix), although these polygons are unlikely to occur in practical applications.

4.2 Exact Algorithm

If we use a rational parameterization of the circle S^1, each one of the surfaces $\mathcal{A}^*$, $\mathcal{B}^*$, and $\mathcal{P}^*$ is a piecewise-smooth collection of algebraic surface patches of constant degree with no self-intersection. We project the boundary curves and silhouette curves of each patch to the starting plane Π_0. Since there are $O(np)$ surface patches altogether, we obtain a set of $O(np)$ algebraic curve segments (namely degree-four *limaçon* arcs, circular arcs, and line segments). Together with the boundary of X^0, which is comprised of $O(n)$ line segments, these curves induces a subdivision $\mathcal{C}$ of the plane into cells with total complexity $O(n^2p^2)$, and we can compute this cell decomposition in $O(n^2p^2)$ time using a randomized incremental algorithm [5]. The points in each cell of $\mathcal{C}$ all have the same object above, and the same object below. Thus, the capture region is the union of cells of $\mathcal{C}$. This immediately implies the following upper bound.

Lemma 4.2. *The worst-case complexity of the capture region X is $O(n^2p^2) = O(n^6)$, where $p = O(n^2)$ is the number of critical orientations.*

To finish the computation of the capture region, we need to identify the objects above and below each cell in $\mathcal{C}$. To do this efficiently, we compute two three-dimensional cylindrical decompositions, one for the lower envelope of the surface patches above Π_0, the other for the upper envelope of the surface patches below Π_0. Because the $O(np)$ surface patches meet only at their boundaries, we can show that each cylindrical decomposition has complexity $O(n^2p^2)$ and can be computed in $O(n^2p^2)$ time using a randomized incremental algorithm. (For a similar combinatorial analysis, see [1].) The intersection of any cylindrical cell with Π_0 is the union of several cells in $\mathcal{C}$. For each cylindrical cell that touches $\mathcal{P}^*$ or Π, we mark the corresponding cells in $\mathcal{C}$, each in constant time. The cells that are marked twice, once from above and once from below, comprise the capture region X.

Lemma 4.3. *The capture region X can be computed in time $O(n^2p^2) = O(n^6)$, where $p = O(n^2)$ is the number of critical orientations.*

4.3 Approximate Algorithm

A discrete approximation of the capture region can also be computed using classical hidden-surface removal techniques such as z-buffering to render polyhedral approximations of all surfaces of interest in a rasterized version of the x, y plane, the orientation θ acting as depth for orthographic projection. Given the critical orientations, it is easy to construct polyhedral approximations of $\mathcal{A}^*$, $\mathcal{B}^*$, and $\mathcal{P}^*$ that achieve any desired degree of accuracy.

The algorithm proceeds in three elementary steps:

1. **Render X^0:** Construct the polygon X^0 and a bounding rectangle R^0 for it, and rasterize X^0

into an $N \times N$ image buffer I_0 representing R^0 with background color 0 and foreground color 1.

2. **Render** X^+**:** Initialize a second $N \times N$ image buffer I^+ representing R^0 with color 0. Attach to it a z-buffer with initial depth 0. Assign the color 1 to the (polyhedral approximations) of $\mathcal{P}^*$ and Π, and assign the color 0 to the (polyhedral approximations) of $\mathcal{A}^*$, $\mathcal{B}^*$, and Π'. Render the five surfaces.

3. **Render** X^-**:** Repeat the process of step 2 to render the five surfaces associated with X^- into a new image buffer I^-. Note: the nonnegative value of $-\theta$ has to be used as depth in this case.

4. **Output** X as the binary AND of I^0, I^-, I^+.

We have implemented this algorithm. Figure 5 shows three examples. In each case, the left part of the figure shows the polygon in its initial configuration and the corresponding capture region. The right part of the figure shows the projections of the surfaces $\mathcal{A}^*$, $\mathcal{B}^*$, $\mathcal{P}^*$, Π and Π' after hidden-surface removal, with the outline of the region X^0 overlaid. The three polygons have respectively 9, 9, and 8 edges, with 20, 18 and 7 critical orientations. In the last case, the distance between the two robots is greater than the width of the polygon, and two of the critical orientations are escape angles.

Figure 6 shows the first of the three polygons on the verge of escaping through ac when robot c is on the boundary of the capture region.

5 Discussion

Let us conclude with a brief discussion of future work. First on our list is the implementation of the proposed exact algorithm. Extending both approaches to discs with nonzero radius should not pose conceptual difficulties since the discussion of Sections 2 and 3 is valid for arbitrary convex objects and discs of arbitrary radius. Adapting the two algorithms to this case will essentially require adapting the computation of critical orientations so it handles convex generalized polygons bounded by line segments and circular arcs, and, in the case of the exact algorithm, constructing the arrangement of slightly more complicated curves. It would of course be interesting to extend the approach presented in this paper to non-convex objects, but it is not clear at this point whether appropriate canonical escape motions can be defined in this case.

From a practical point of view, we believe that capture regions as characterized and computed in this paper will prove a useful tool for various problems in robotics and flexible manufacturing. We intend to

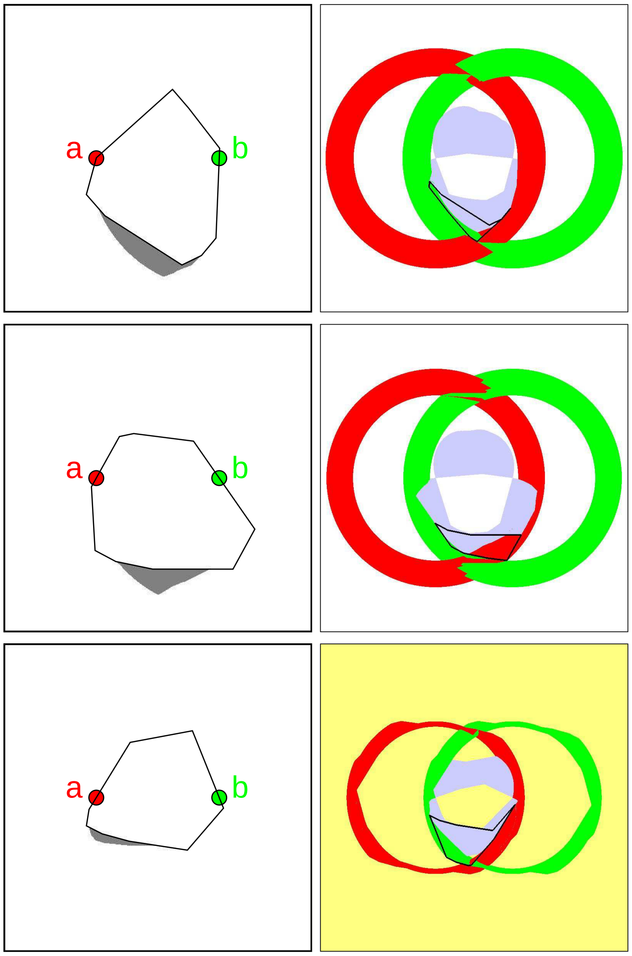

Figure 5. Left: three polygons and their capture regions. Right: the projections of the surfaces $\mathcal{A}^*$, $\mathcal{B}^*$, $\mathcal{P}^*$, Π and Π'. The background is white for the first two polygons, indicating that Π is visible and there is no escape angle. The shaded background for the third polygon indicates that Π' is visible and an escape angle exists.

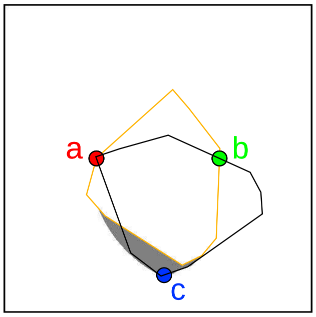

Figure 6. The initial and final positions of a polygon during the rotational phase of a counterclockwise canonical motion. Here c is on the boundary of the capture region and the polygon is about to escape by translation through ac.

demonstrate that this is indeed the case by integrating the new results obtained in this paper with some of our previous work on fixturing, grasping, and in-hand manipulation [20, 21], mobile robot motion planning [17, 22], and parts feeding [3].

Acknowledgments. This research was supported in part by the National Science Foundation under grant IRI-9907009.

Appendix

We demonstrate here with an example that the number of critical orientations associated with a convex polygon of size n is $\Omega(n^2)$. Consider a polygon with one vertex at the center of a circle and the other vertices positioned at a fixed distance beyond the radius of the circle (Figure 7).

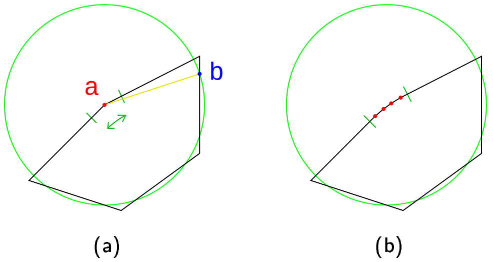

Figure 7. A convex polygon with $\Omega(n^2)$ events. (a) Point a reverses its motion around ∂P $\Omega(n)$ times in the neighborhood of a vertex. (b) Replace the vertex with $\Omega(n)$ vertices near the center and perturb to make convex.

Let the radius of the circle be the distance between a and b. The turning of P is equivalent to a and b traveling around the perimeter of P, so we will think in those terms for a moment. As the angle increases, occasionally a or b must reverse direction to maintain contact. These reversals occur whenever the segment ab is perpendicular to the edge containing either a or b. The construction of P in Figure 7 forces a to reverse direction $O(n)$ times. Suppose we add n more vertices very near the vertex at the center of the circle, and perturb them so that the object remains convex. If the vertices are placed between the points where a reverses, then a will pass $\Omega(n)$ vertices as b traverses one edge. This results in $\Omega(n^2)$ events.

References

[1] Pankaj K. Agarwal, Jeff Erickson, and Leonidas J. Guibas. Kinetic binary space partitions for intersecting segments and disjoint triangles. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 107–116, 1998.

[2] R.S. Ball. *A treatise on the theory of screws.* Cambridge University Press, 1900.

[3] S. Blind, C. McCullough, S. Akella, and J. Ponce. Manipulating parts with an array of pins: A method and a machine. *International Journal of Robotics Research*, 20(10):808–818, 2001.

[4] R.C. Brost. Dynamic analysis of planar manipulation tasks. In *IEEE Int. Conf. on Robotics and Automation*, pages 2247–2254, Nice, France, June 1992.

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

[6] C. Ferrari and J.F. Canny. Planning optimal grasps. In *IEEE Int. Conf. on Robotics and Automation*, pages 2290–2295, Nice, France, June 1992.

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

[8] D.J. Kriegman. Let them fall where they may: capture regions of curved objects and polyhedra. *International Journal of Robotics Research*, 16(4):448–472, August 1997.

[9] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica, Special Issue: Robotics*, 2(4):541–558, November 1987.

[10] B. Mishra and N. Silver. Some discussion of static gripping and its stability. *IEEE Systems, Man, and Cybernetics*, 19(4):783–796, 1989.

[11] V-D. Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7(3):3–16, June 1988.

[12] J. Ponce, S. Sullivan, A. Sudsang, J-D. Boissonnat, and J-P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research*, 16(1):11–35, February 1997.

[13] E. Rimon and A. Blake. Caging 2D bodies by one-parameter two-fingered gripping systems. In *IEEE Int. Conf. on Robotics and Automation*, pages 1458–1464, Minneapolis, MN, April 1996.

[14] E. Rimon and J. W. Burdick. Mobility of bodies in contact—i: A new 2^{nd} order mobility index for multiple-finger grasps. *IEEE Transactions on Robotics and Automation*, 14(5):696–708, 1998.

[15] E. Rimon and J. W. Burdick. Mobility of bodies in contact—ii: How forces are generated by curvature effects. *IEEE Transactions on Robotics and Automation*, 14(5):709–717, 1998.

[16] B. Roth. Screws, motors, and wrenches that cannot be bought in a hardware store. In *Int. Symp. on Robotics Research*, pages 679–693. MIT Press, 1984.

[17] A. Sudsang and J. Ponce. On grasping and manipulating polygonal objects with disc-shaped robots in the plane. In *IEEE Int. Conf. on Robotics and Automation*, pages 2740–2746, Leuven, Belgium, June 1998.

[18] A. Sudsang, J. Ponce, M. Hyman, and D.J. Kriegman. On manipulating polygonal objects with three 2-dof robots in the plane. In *IEEE Int. Conf. on Robotics and Automation*, pages 2227–2233, Detroit, MI, 1999.

[19] A. Sudsang, J. Ponce, and N. Srinivasa. Algorithms for constructing immobilizing fixtures and grasps of three-dimensional objects. In J.-P. Laumont and M. Overmars, editors, *Algorithmic Foundations of Robotics II*, pages 363–380. AK Peters, Ltd., 1997.

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

[21] A. Sudsang, J. Ponce, and N. Srinivasa. Grasping and in-hand manipulation: Geometry and algorithms. *Algorithmica*, 26:466–493, January 2000.

[22] A. Sudsang, F. Rothganger, and J. Ponce. A new approach to motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Transactions on Robotics and Automation*, 2002. In press.

[23] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.