

Capturing a Convex Object with Three Discs

Jeff Erickson, Shripad Thite, Fred Rothganger, and Jean Ponce *Fellow, IEEE*

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, the so-called *capture region*, that prevent P from escaping to infinity via continuous rigid motion. We show that the computation of the capture region reduces to 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.

Index Terms—Robots; Motion-planning; Capturing; Caging; Capture region; Multiple manipulators

I. INTRODUCTION

In this paper, we study the problem of capturing a convex object in the plane with three congruent disc-shaped robots. By *capturing* an object, we mean preventing it from escaping to infinity by a rigid, continuous motion. The problem of capturing an object with the minimum number of points was posed by Kuperberg [1], [2]. Capturing has also been called *caging* in the literature [3], [4], [5], [6]. A captured object may not be immobilized but it is confined to a bounded subset of the plane. In practice, the 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.

The *capture region* (or *caging set* [3]) is the set of configurations of a robotic system that capture the object. 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 [7], [8] and, as noted in [9], [10] for example, force closure implies

form closure and vice versa. They are the traditional theoretical basis for grasp planning algorithms (see, for example, [11], [12], [10], [13]).

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 [16] 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 [17], [18], [3] for related work): an object is captured when it lies in a compact valid region of its configuration space.

Grasping an object means fixing the object in one pre-defined configuration or one of a finite number of possible configurations. On the other hand, capturing an object means ensuring that the object is constrained to a compact region of space. An immobilizing grasp certainly captures the object. Thus, capturing is a weaker condition than grasping. Capturing is potentially useful in more applications because it permits uncertainty in the position and orientation of the object, whether because of actual uncertainty or imprecision in measurement. Capturing avoids the need to model friction between robots and the object being captured since the robots are not needed to be in contact with the object. Further, we expect that in practice, for well-shaped objects, capturing permits more relaxed coordination between the robots because each robot is allowed to be anywhere in a sufficiently large region of space and the object is still captured for every possible position of each robot anywhere in its region.

Capture regions have been applied to a number of problems in sensorless manipulation, including grasping and in-hand manipulation [3], [19], [20], mobile robot motion planning [21], [22], parts feeding [23], [17], and stable pose computation [18]. The survey by Bicchi and Kumar [24] reviews capturing and related problems. The current paper presents 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 [23],

[17], [18] or to robotic systems with a single dof [3], [25], 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 [21], [19], [20], [22].

We propose 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 (2 dof) with a convex object P in its initial configuration (3 dof), and characterize the set of positions of a third robot c in the plane (2 dof) 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.

Not every convex object P can be captured by three robots and it is therefore interesting to determine whether the capture region is empty. For example, a parallelogram can always escape by translation parallel to one of its two edge directions [2]. The restriction that two robots are fixed means that the third robot may not be able to capture P . For instance, if a and b are in contact with two parallel edges of P , then no placement of c will prevent P from escaping by pure translation in a direction parallel to these two contact edges. If a and b are given in contact with two non-parallel edges of P , there may still be no placement of c that will capture P . For instance, in Figure 1, no placement of c will prevent the small counter-clockwise rolling motion of the triangle that is sufficient to allow it to translate to infinity.

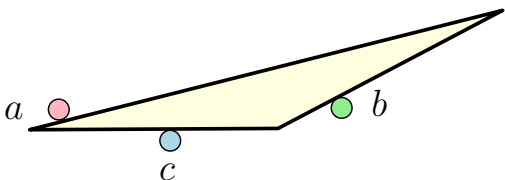


Fig. 1. The triangle cannot be captured by the third robot c with robots a and b fixed, no matter where c makes contact with the third edge.

On the other hand, only two robots may suffice to capture a non-convex object. Vahedi and van der Stappen [6], and also independently Pipattanasompom and

Sudsang [5], gave algorithms for capturing a concave polygon with two robots.

Computing a capture region is the first step towards computing a motion plan for manipulating a polygonal object in the plane, such as by moving one robot at a time while keeping the object captured. Such a plan requires minimal coordination between robots and allows uncertainty in the position of the object. With this purpose, we require to compute the set of all possible positions, if any, of a third robot that would capture the object while keeping the other two robots fixed. Knowing the component of this capture region containing the current position of the third robot suffices to compute how far in any direction the third robot can be moved from its current position. In this paper, we take the first step towards this goal by considering the special case when the initial configuration of the object is in contact with the two fixed robots.

Subsequent to the appearance of our results in a conference paper [26], Vahedi and van der Stappen [6] extended our results to capturing a non-convex polygon without holes with three congruent disc-shaped robots. They compute all positions of the third robot that capture the given convex object given the placement of the other two robots fixed relative to the initial configuration of the object. The fixed robots need not be in contact with the object boundary. Using pseudo-triangulations and dynamic graph data structures, they maintain the connected components of the capture region of the third robot as the distance between the two fixed robots varies continuously.

We choose a coordinate system such that robot a is at the origin; 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.

II. CANONICAL MOTIONS

We assume in this section that the position of robot c is fixed. Polygon P is allowed an arbitrary continuous rigid motion without penetrating a , b , or c . We define a particular kind of motion of P , called a *canonical motion* (Definition 1). We show that if P can escape from its initial configuration by a rigid motion, then it can also escape by a *canonical motion*. This will allow us to characterize capture regions in a simple fashion in the next section by restricting our attention to canonical motions.

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. Henceforth, we will use P to denote the result of this Minkowski sum, which is also convex. 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 applies to arbitrary convex planar objects and robots with nonzero radius. We work in the configuration space $\mathbb{R}^2 \times \mathbb{S}^1$ of possible positions (x, y) and orientations θ of P . We will abuse notation in the sequel and also designate by $a, b, \text{ or } c$ the point in any $\theta = \text{constant}$ plane (or θ -slice) of $\mathbb{R}^2 \times \mathbb{S}^1$ where the vertical line erected at the corresponding robot position intersects that plane.

Each robot defines in $\mathbb{R}^2 \times \mathbb{S}^1$ 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 rigid, continuous motion of P without penetrating any robot describes a curve in valid space. Hence, 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 2). 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 2 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 the corresponding pocket by V_{θ} . 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.

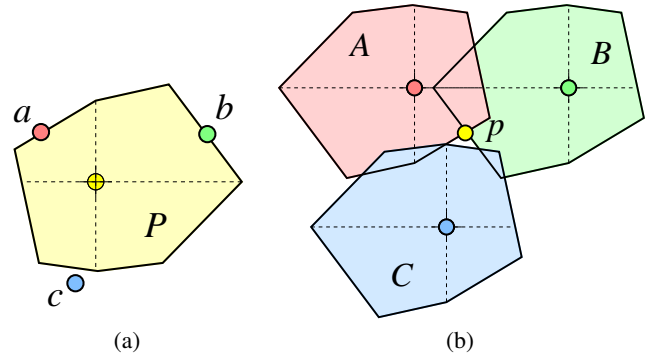


Fig. 2. A convex object P in contact with the robots a and b in (a) workspace and (b) configuration space. Note that p lies in a pocket in this case. The θ subscripts have been omitted for readability.

Let us assume from now on that a pocket exists for $\theta = 0$ and that the initial configuration belongs to this pocket. We define a *canonical motion* as follows (Figure 3).

Definition 1 (Canonical motion): A canonical motion of P is a monotonic increase or decrease in orientation θ while maintaining contact between P and the robots a and b until at least one of the following conditions is satisfied:

- 1) two of the obstacles no longer intersect, allowing the object to escape by a pure translation; or
- 2) P is blocked from further rotation by a contact with c ; or
- 3) P returns to its original orientation.

A *canonical escape motion* is defined as a canonical motion ending with condition (1) being satisfied, the so-called *escape condition*, after which the polygon P can translate to infinity. 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 $0 \leq \theta < \psi$, 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 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

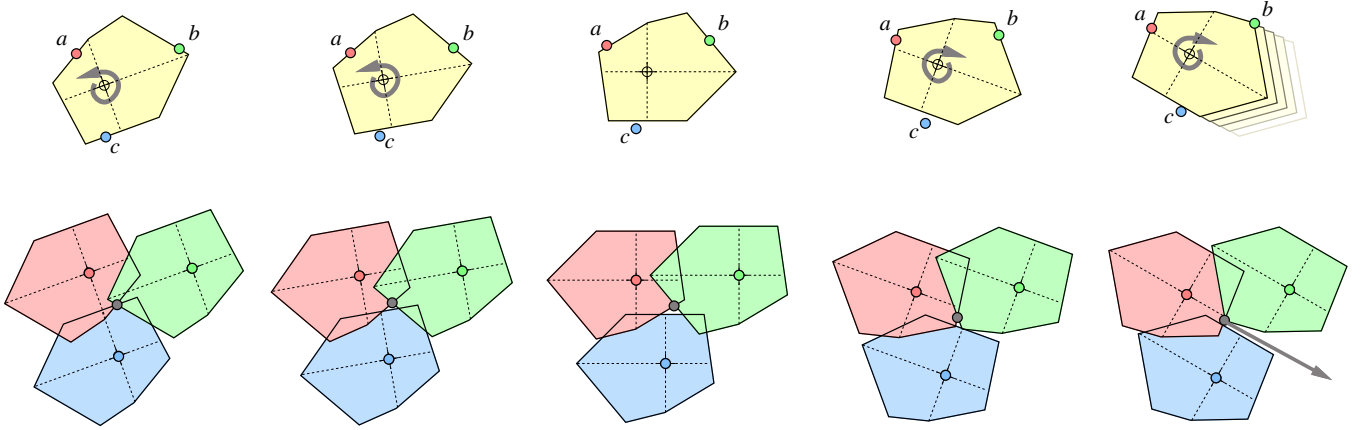


Fig. 3. Canonical motions, with the corresponding changes in the configuration plane. Center to left: counterclockwise turning stopped by a triple contact. Center to right: clockwise turning ending in escape through bc .

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: P can escape if and only if it can escape by some canonical motion.

Proof: If P_0 is not in a pocket of the θ -slice in the initial configuration $\theta = 0$, then there exists a pure translation which allows P to escape while remaining in its initial orientation. Otherwise, if P can escape, then P_0 belongs to an unbounded component of valid space. Therefore, there exists a θ^* such that P_{θ^*} belongs to an unbounded component of the θ^* -slice of valid space; we can choose a minimum such θ^* . We can assume without loss of generality that $\theta^* > 0$. For values of θ just less than θ^* , P_{θ} is in a pocket of the θ -slice, so every pair of obstacle regions intersects. The orientation θ^* is the smallest value for which some pair of obstacle regions have disjoint interiors. The polygon P can escape by executing the following sequence of steps which is a canonical motion: P in orientation $\theta = 0$ lies at the vertex v which is the intersection of the boundaries of A_0 and B_0 . As θ increases or decreases monotonically, the locus of vertex v is an edge e_v of the contact space. P monotonically increases or decreases its orientation θ while remaining on the edge e_v , which means P remains in contact with the robots a and b . When the edge e_v ends, P continues to follow the edge defined by the intersection of the next pair of intersecting surface patches, one from A_{θ} and the other from B_{θ} . Finally, P achieves the orientation θ^* at which some pair of obstacle regions have disjoint interiors. This configuration satisfies the escape condition (1). Now, P can escape by pure translation while remaining in orientation θ^* . ■

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.

III. 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 xy -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 $X = X^0 \cap X^+ \cap X^- = X^+ \cap X^-$.

A. 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 4). It is clear that c is in A_{θ}^* (respectively B_{θ}^*) if and only if A_{θ} and C_{θ} (respectively B_{θ} and C_{θ}) intersect. It is also clear that A_{θ}^* (respectively B_{θ}^*) is the area swept by P_{θ} while maintaining contact with a (respectively b).

Let us assume that A_{θ} and B_{θ} intersect (otherwise the object can escape by a translation through ab),

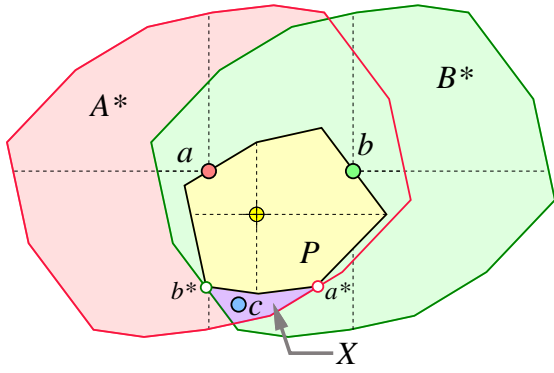


Fig. 4. The A_θ^* , B_θ^* , P_θ^* , and X^θ regions at $\theta = 0$. The subscripts and superscripts have been omitted for readability.

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 4; 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: 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 .

Proof: The polygon P in its initial configuration does not collide with c , and 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 $A_\theta^* \cap B_\theta^* \setminus P_\theta^*$. Therefore, the condition $c \in X^\theta$ is necessary for ensuring that the configuration p is in a pocket, otherwise escaping by translation would be trivial.

The polygon cannot escape the pocket through ab since A_θ and B_θ intersect. It remains to show that it cannot escape through ac or through bc . Because $c \in A_\theta^*$, we know that $A_\theta \cap C_\theta \neq \emptyset$. Therefore, the polygon cannot escape through ac . By a symmetric argument, $B_\theta \cap C_\theta \neq \emptyset$, preventing escape through bc . ■

B. 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 5). 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.

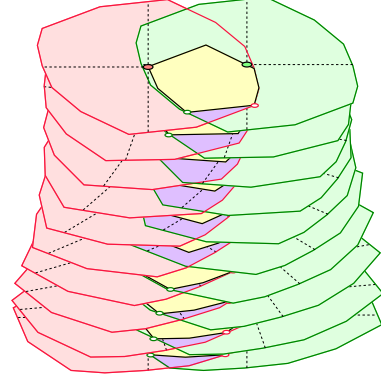


Fig. 5. The volumes bounded by the surfaces \mathcal{A}^* , \mathcal{B}^* , and \mathcal{P}^* . We show only a finite number of slices to reveal some of the internal structure of these volumes.

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 the “vertical” axis. We have the following result.

Lemma 4: 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 that there is no need to check that c remains in

the lower component of $A_\theta^* \cap B_\theta^* \setminus P_\theta^*$ during the canonical motion since Δ_c would have to cross A^* , B^* , or P^* first in order for c 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 $X = X^0 \cap X^+ \cap X^- = X^+ \cap X^-$.

IV. COMPUTING THE CAPTURE REGION

Lemma 4 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.

A. Critical Orientations

The polygon $Q = P \oplus (-P)$ has $2n$ edges and can be constructed in linear time. The two obstacles A^* and 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 lies on an edge of B_θ (or vice versa).

Lemma 5: The critical orientations and the counter-clockwise 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 [27], [28].

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. Consider a polygon with one vertex at the center of a circle and another $n/2$ vertices positioned at a fixed distance beyond the radius of the circle (Figure 6).

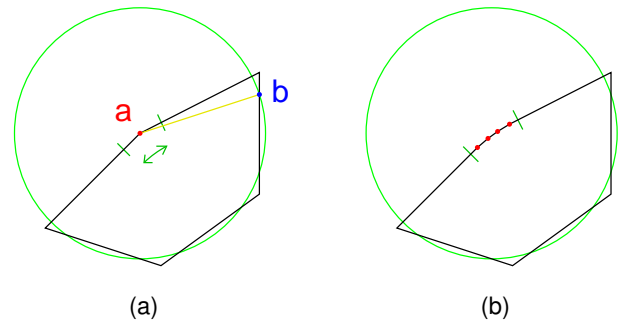


Fig. 6. A convex polygon with $\Omega(n^2)$ events. (a) Point a reverses its motion around the boundary of P roughly $n/2$ times in the neighborhood of a vertex. (b) Replace the vertex with $n/2$ vertices near the center and perturb to make convex.

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 6 forces a to reverse direction $\Omega(n)$ times. Suppose we add $n/2 - 1$ 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 for each edge that b traverses. This results in $\Omega(n^2)$ events.

B. 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. Each surface patch is a ruled surface swept by an edge of A_θ , B_θ , or P_θ as the orientation θ increases or decreases monotonically. 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). These curves induce 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 [29]. 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.

Theorem 6: 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* [30], [31], 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 have constant algebraic degree and because any two patches meet only at their boundaries, each cylindrical decomposition has complexity $O(n^2p^2)$. The cylindrical decomposition can be computed in $O(n^2p^2)$ time using a randomized incremental algorithm, such as the algorithm described in [32]. (For a similar combinatorial analysis, see [33].) 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}^* , 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 .

Theorem 7: 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.

C. 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 xy 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 that 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^+ , and I^- .

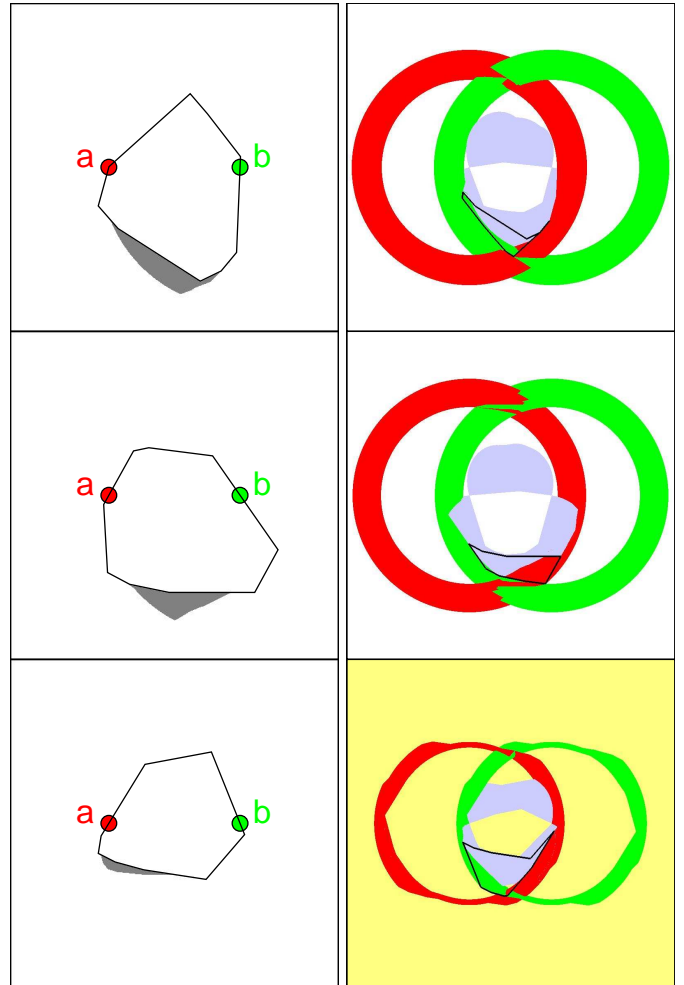


Fig. 7. *Left:* three polygons and their capture regions (shaded). *Right:* the projections of the surfaces \mathcal{A}^* , \mathcal{B}^* , \mathcal{P}^* , Π and Π' . The background is white for the first two polygons, indicating that Π is visible and there is no escape angle. The shaded background for the third polygon indicates that Π' is visible and an escape angle exists. Other colors indicate other surface patches.

V. EXPERIMENTS

We implemented our approximate algorithm. Figure 7 shows three examples. In each case, the left part of

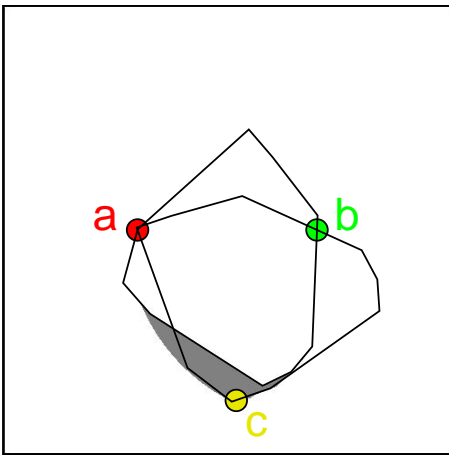


Fig. 8. The initial and final positions of a polygon during the rotational phase of a counterclockwise canonical motion. Here, c is on the boundary of the capture region and the polygon is about to escape by translation through ac .

the figure shows the polygon in its initial configuration and the corresponding capture region is shaded. 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 8 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.

VI. CONCLUSION

We conclude with a brief discussion of extensions and open problems.

Both the exact and approximate algorithms extend to discs with nonzero radius. As noted before, the discussion of Sections II and III 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.

We sketch an extension of our result to the case when the three robots a , b , and c have different radii; the difficulty is when robot c has smaller radius than either a or b . The robot c would have to belong to a subset of X^θ (Figure 4) to prevent escape through either ac or bc . This subset of X^θ is characterized not only by the three

radii but also by the pair of edges where a and b make contact. Specifically, the portion of A_θ^* and B_θ^* below the line a^*b^* would have to be offset by the difference in radii. Lemma 3 can then be proved for the new definition of X^θ . The overall complexity of the algorithm remains unchanged because the upper bound on the number of surface patches is already proportional to the number of critical orientations.

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. It would be interesting to integrate the new results obtained in this paper with previous work on fixturing, grasping, and in-hand manipulation [19], [20], mobile robot motion planning [34], [22], and parts feeding [23].

Capturing an object may be the first step in immobilizing it. It is an interesting open problem to compute the subset of the capture region within which the third robot can move eventually to an immobilizing grasp of the object.

Given a placement of the third robot, our algorithm can determine whether this placement captures the convex polygon together with the other two robots. The recent result of Vahedi and van der Stappen [6] relaxes the requirement that the polygon be convex and that the two fixed robots be in contact with the boundary of the polygon in its initial configuration. It remains an open problem to determine all triples of robot placements, relative to the initial configuration of a convex or non-convex polygon, that prevent the polygon from subsequently escaping to infinity.

ACKNOWLEDGMENT

This research was supported in part by the National Science Foundation under grant IRI-9907009. Research by Shripad Thite and Fred Rothganger was conducted while both were Ph.D. students in Computer Science at the University of Illinois at Urbana-Champaign. Research by Jean Ponce and Fred Rothganger was conducted at the Department of Computer Science and the Beckman Institute of the University of Illinois at Urbana-Champaign. We thank the anonymous referees for useful comments.

REFERENCES

- [1] W. Kuperberg, "Problems on polytopes and convex sets," in *DIMACS Workshop on Polytopes*, 1990, pp. 584–589.
- [2] J. O'Rourke, "Computational geometry column 9," *ACM SIGACT News*, vol. 21, no. 1, pp. 18–20, January 1990.
- [3] E. Rimon and A. Blake, "Caging 2D bodies by one-parameter two-fingered gripping systems," in *IEEE Int'l. Conf. Robotics & Automation*, Minneapolis, MN, April 1996, pp. 1458–1464.

- [4] C. Davidson and A. Blake, "Caging planar objects with a three-finger one-parameter gripper," in *Proc. IEEE Int'l Conf. Robotics & Automation*, Leuven, Belgium, June 1998, pp. 2722–2727.
- [5] P. Pipattanasompom and A. Sudsang, "Two-finger caging of concave polygon," in *Proc. IEEE Int'l Conf. Robotics & Automation*, 2006, pp. 2137–2142.
- [6] M. Vahedi and A. F. van der Stappen, "Caging polygons with two and three fingers," *Algorithmic Foundations of Robotics VII*, (S. Akella, N. Amato, W. Huang, and B. Mishra eds.), Springer Tracts in Advanced Robotics, Springer-Verlag, 2006, proc. 7th Workshop on Algorithmic Foundations of Robotics (WAFR).
- [7] R. Ball, *A treatise on the theory of screws*. Cambridge University Press, 1900.
- [8] B. Roth, "Screws, motors, and wrenches that cannot be bought in a hardware store." MIT Press, 1984, pp. 679–693.
- [9] B. Mishra and N. Silver, "Some discussion of static gripping and its stability," vol. 19, no. 4, pp. 783–796, 1989.
- [10] V.-D. Nguyen, "Constructing force-closure grasps," vol. 7, no. 3, pp. 3–16, June 1988.
- [11] C. Ferrari and J. Canny, "Planning optimal grasps," Nice, France, June 1992, pp. 2290–2295.
- [12] B. Mishra, J. Schwartz, and M. Sharir, "On the existence and synthesis of multifinger positive grips," *Algorithmica, Special Issue: Robotics*, vol. 2, no. 4, pp. 541–558, November 1987.
- [13] 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," vol. 16, no. 1, pp. 11–35, February 1997.
- [14] E. Rimon and J. W. Burdick, "Mobility of bodies in contact—I: A new 2^{nd} order mobility index for multiple-finger grasps," vol. 14, no. 5, pp. 696–708, 1998.
- [15] —, "Mobility of bodies in contact—II: How forces are generated by curvature effects," vol. 14, no. 5, pp. 709–717, 1998.
- [16] A. Sudsang, J. Ponce, and N. Srinivasa, "Algorithms for constructing immobilizing fixtures and grasps of three-dimensional objects," in *Algorithmic Foundations of Robotics II*, J.-P. Laumont and M. Overmars, Eds. AK Peters, Ltd., 1997, pp. 363–380.
- [17] R. Brost, "Dynamic analysis of planar manipulation tasks," Nice, France, June 1992, pp. 2247–2254.
- [18] D. Kriegman, "Let them fall where they may: capture regions of curved objects and polyhedra," vol. 16, no. 4, pp. 448–472, August 1997.
- [19] A. Sudsang, J. Ponce, and N. Srinivasa, "Grasping and in-hand manipulation: Experiments with a reconfigurable gripper," *Advanced Robotics*, vol. 12, no. 5, pp. 509–533, December 1998.
- [20] —, "Grasping and in-hand manipulation: Geometry and algorithms," *Algorithmica*, vol. 26, pp. 466–493, January 2000.
- [21] A. Sudsang, J. Ponce, M. Hyman, and D. Kriegman, "On manipulating polygonal objects with three 2-dof robots in the plane," Detroit, MI, 1999, pp. 2227–2233.
- [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," 2002, conditionally accepted for publication.
- [23] S. Blind, C. McCullough, S. Akella, and J. Ponce, "Manipulating parts with an array of pins: A method and a machine," vol. 20, no. 10, pp. 808–818, 2001.
- [24] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int'l Conf. Robotics & Automation*, 2000, pp. 348–353.
- [25] A. Sudsang and T. Luewirawong, "Capturing a concave polygon with two disc-shaped fingers," in *Proc. IEEE Int'l Conf. Robotics & Automation*, 2003, pp. 1121–1126.
- [26] J. Erickson, S. Thite, F. Rothganger, and J. Ponce, "Capturing a convex object with three discs," in *Proc. IEEE Int'l Conf. Robotics & Automation*, 2003, pp. 2242–2247.
- [27] M. E. Houle and G. T. Toussaint, "Computing the width of a set," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-10, no. 5, pp. 761–765, 1988.
- [28] G. T. Toussaint, "Solving geometric problems with the rotating calipers," in *Proc. IEEE MELECON '83*, 1983, pp. A10.02/1–4.
- [29] K. Clarkson and P. Shor, "Applications of random sampling in computational geometry II," *Discrete Comput. Geometry*, vol. 4, no. 1, pp. 387–421, 1989.
- [30] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006. [Online]. Available: <http://planning.cs.uiuc.edu/web.html>
- [31] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, ser. Algorithms and Computation in Mathematics. Springer, 2003, vol. 10.
- [32] K. Mulmuley, *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice-Hall, 1994.
- [33] P. Agarwal, J. Erickson, and L. Guibas, "Kinetic binary space partitions for intersecting segments and disjoint triangles," in *Proc. 9th ACM-SIAM Symp. Discrete Algorithms (SODA)*, 1998, pp. 107–116.
- [34] A. Sudsang and J. Ponce, "On grasping and manipulating polygonal objects with disc-shaped robots in the plane," Leuven, Belgium, June 1998, pp. 2740–2746.



Jeff Erickson is an Associate Professor and Willett Faculty Scholar in the Department of Computer Science at the University of Illinois at Urbana-Champaign, USA. He has a Ph.D. in Computer Science from the University of California at Berkeley. He is a member of the ACM, including SIGACT, SIGGRAPH, and SIGMOD. He can be reached by email at jeffe@cs.uiuc.edu.



Shripad Thite is a Postdoctoral Fellow in the Center for the Mathematics of Information at the California Institute of Technology in Pasadena, USA. He was a postdoctoral researcher in the Department of Computer Science at the Technische Universiteit Eindhoven, the Netherlands. He has a Ph.D. and M.S. in Computer Science both from the University of Illinois at Urbana-Champaign, and a B.E. in Computer Engineering from the University of Pune, India, while studying at the Government College of Engineering (COEP). He is a member of the ACM, including SIGACT. He can be reached by email at shripad@caltech.edu.



Fred Rothganger works at Sandia National Labs in Albuquerque, New Mexico, having obtained a Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign, USA. He has an M.S. in Computer Science from the University of Massachusetts Boston and a B.A. *cum laude* from Central Bible College in Springfield, Missouri. He can be reached by email at rothgang@uiuc.edu.



Jean Ponce is a Professor in the Département d'Informatique, Ecole Normale Supérieure, Paris, France, and heads a research team, WILLOW, which is a joint effort of ENS, INRIA, and ENPC. He is also Professor in the Department of Computer Science and the Beckman Institute of the University of Illinois at Urbana-Champaign, USA. He has a Doctorat d'Etat in Computer Science from the University of Paris Orsay. He is an IEEE Fellow since 2003. He can be reached by email at Jean.Ponce@ens.fr.