Caging Polygonal Objects Using Formationally Similar Three-Finger Hands

Hallel A. Bunis, Elon D. Rimon, Yoav Golan, and Amir Shapiro

***Abstract*—Caging offers a robust strategy for grasping objects with robot hands. This letter describes an efficient caging-to-grasping algorithm for polygonal objects using minimalistic three-finger robot hands. This letter describes how to cage and then grasp polygonal objects, using single actuator triangular three-finger formations, whose shape is determined by any desired immobilizing grasp of the polygonal object. While the hand's *configuration space* is four-dimensional, the algorithm uses the hand's two-dimensional *contact space*, which represents all two- and three-finger contacts along the grasped object boundary. This letter describes how the problem of computing the critical cage formation that allows the object to escape the hand is reduced to a search along a *caging graph* constructed in the hand's contact space. Starting from a desired immobilizing grasp, the graph is searched for the critical cage formation, which is used to determine the *caging regions* surrounding the immobilizing grasp. Any three-finger placement within these regions guarantees robust object grasping. The technique is demonstrated with a detailed computational example and a video clip, which shows caging experiments with a single actuator three-finger robot hand.**

***Index Terms*—Caging, robot grasping, robot grasp planning.**

I. Introduction

CAGING offers a robust approach to grasping objects with multi-finger robot hands under huge uncertainty of the finger positions relative to the grasped object. To securely grasp an object, the fingers are first placed in *caging regions* surrounding a desired immobilizing grasp. This prevents the object from escaping the hand. The hand is then closed while maintaining a cage formation, until the desired immobilizing grasp is reached. This letter focuses on caging polygonal objects using three-finger robot hands. Our focus on three-finger hands is motivated by practical hand mechanism designs.

As discussed by Dollar *et al.* [1], [2], roboticists seek general purpose *minimalistic hand designs* that will posses the smallest number of fingers as well as the smallest number of actuators. Three-finger hands form an attractive minimalistic design, since three point or disc fingers can immobilize every polygonal object which does not possess opposing parallel edges [3]. Moreover, by limiting the three-finger hand to a particular family of formations such as similar triangles, a *single actuator* can simultaneously close the three fingers, thus allowing extremely simple hand designs (Fig. 1).

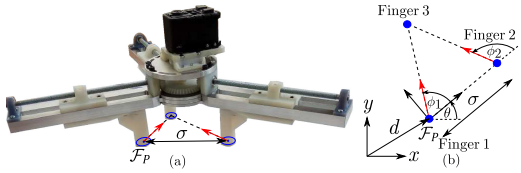

Fig. 1. (a) A specially designed three-finger robot hand. (b) The hand's configuration is specified by the palm frame $\mathcal{F}_P$, the formation scale σ, and the hand shape defined by $\phi=(\phi_1, \phi_2)$.

Since robotic caging was introduced by Rimon and Blake [4], caging theory has advanced considerably. Work on caging includes an examination of the relationship between caging and grasping [5], the inclusion of gravity in partial cage formations [6], and the application of computer vision tools for caging based on topological characteristics of the caged objects [7]. Caging has also been considered for robust object manipulation [8], [9], as well as robust transfer of warehouse objects by mobile robots [10]–[12]. Caging holds much potential in other tasks, such as space debris removal [13] and microscopic object manipulation [14].

A number of caging algorithms for polygonal objects have been proposed in the literature. In the case of two-finger hands, the prevailing caging algorithms involve computing all cage formations directly in the hand's four-dimensional configuration space [15], [16]. However, two-finger cage formations can be alternatively computed in the hand's two-dimensional *contact space* [17], which offers implementation simplicity and geometric verifiability along the grasped object boundary.

In the case of three-finger hands, the existing caging algorithms impose some constraint on the finger formations. Erickson *et al.* [18] attempted to generate small caging regions localized around three particular edges of the grasped polygon. Subsequent work by Vahedi and van der Stappen [15] was able to compute all three-finger caging grasps, provided that two fingers are held fixed as base fingers. Wan [19], [20] considered efficient numerical computation of three-finger cage formations with fixed base fingers. Wan [21] also proposed the topological enumeration of the hand's configuration-space boundary components, in order to efficiently report when a candidate finger formation forms a cage about a given object. Having three as well as higher number of fingers in mind, Pipattanasom-

Manuscript received February 23, 2018; accepted June 21, 2018. Date of publication June 29, 2018; date of current version July 19, 2018. This letter was recommended for publication by Associate Editor M. A. Roa and Editor H. Ding upon evaluation of the reviewers' comments. This work was supported by the Israel Science Foundation under Grant 1253/14. *(Corresponding author: Hallel A. Bunis.)*

H. A. Bunis and E. D. Rimon are with the Department of Mechanical Engineering, Technion—Israel Institute of Technology, Haifa 3200003, Israel (e-mail: hallelb@technion.ac.il; rimon@technion.ac.il).

Y. Golan and A. Shapiro are with the Department of Mechanical Engineering, Ben-Gurion University of the Negev, Beersheba 8410501, Israel (e-mail: yoavgolan1@gmail.com; ashapiro@bgu.ac.il).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The Supplementary Materials contain a video file demonstrating how the I-CAGE Algorithm presented in the paper was used to cage and then grasp a saw. The caging experiment was performed with a specially designed three-finger robot hand to validate the algorithm. This material is 9.6 MB in size.

Digital Object Identifier 10.1109/LRA.2018.2851754

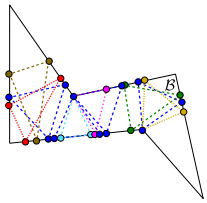

Fig. 2. An equilateral triangle formation allows only four discrete immobilizing grasps of the object $\mathcal{B}$ (blue triangles). This letter considers arbitrary triangle finger formations, which allow a vast choice of immobilizing grasps.

porn *et al.* [22] recently proposed to use *dispersion functions* to compute cage formations as a convex optimization problem. While their method is highly useful for the general caging problem, it is less suited for reaching a specific target immobilizing grasp, since it unnecessarily computes all possible caging regions surrounding a given object.

Motivated by the need to formulate caging algorithms that will lead to pre-specified grasps [5], we recently showed how the *contact space* approach can be extended to three-finger hands that maintain equilateral triangle finger formations. In [23] we laid the theoretical groundwork, while in [24] we validated our work with a real robot hand and added significant computational improvements. However, equilateral formations require finger placement at *very specific grasp points,* which greatly restricts the usefulness of such three-finger formations (see Fig. 2). To overcome this limitation, this letter extends the contact space approach to three-finger hands which maintain completely general triangular formations during the caging-to-grasping process. This introduces great flexibility by allowing selection of the finger formation according to *any desired immobilizing grasp,* which forms an arbitrary triangle, while enforcing a structured hand closing process well suited for minimalistic robot hands. Based on convex decomposition techniques introduced by [15] and [22], we describe an $O(n^3 \log n)$ caging algorithm, where n is the number of edges of the object to be grasped by the robot hand.

The letter introduces in Section II the three-finger caging problem. Section III provides background on the representation of cage formations in the hand's 4D configuration space. Section IV describes the three-finger hand's *contact space*. Section V describes the *caging graph* which is constructed in the hand's contact space. Section VI presents the caging graph search algorithm. Section VII provides a detailed example of how the caging graph can be used for grasping real objects with three disc fingers. The conclusion discusses extension of the caging algorithm to four-finger hands for grasping objects with parallel edges, as well as future extension to 3D grasps. The letter is accompanied by a *video clip* demonstrating caging experiments with the hand of Fig. 1, as well as a MATLAB implementation of the algorithm [25].

II. The Three-Finger Caging Problem

Consider the caging of a polygonal object $\mathcal{B}$ using a robot hand comprised of a rigid *palm*, and three finger mechanisms attached to the palm (Fig. 1(a)). The hand is modeled by a *palm frame*, $\mathcal{F}_P$, that can freely translate and rotate in $\mathbb{R}^2$, whose configuration is specified by $q = (d, \theta) \in \mathbb{R}^2 \times \mathbb{S}$ (Fig. 1(b)). The three point fingers are modeled as the vertices of a variable size triangle which is fixed to the palm frame. The hand's configuration is thus specified by the palm's position and orientation, the hand's triangular shape, and the hand's size, as next described.

The fingers are labeled in counterclockwise order. Finger 1 coincides with the origin of $\mathcal{F}_P$, while Finger 2 is located along the positive x-axis of $\mathcal{F}_P$ (Fig. 1(b)). The distance between Fingers 1 and 2 is specified by a scalar parameter, $\sigma \geq 0$, which will serve as the scale or *size* of the three-finger formation. The position of Finger 3 relative to Fingers 1 and 2 determines the hand's *shape*. In order to specify the hand's shape regardless of the hand's size, the following shape parameters are used.

Definition 1: The inter-finger *shape parameters* are the angles, $\phi = (\phi_1, \phi_2)$, of the palm frame unit vectors that start at Fingers 1 and 2 and point toward Finger 3 (Fig. 1(b)).

The hand's shape parameters, $\phi = (\phi_1, \phi_2)$, are initially determined from a user specified target immobilizing grasp, and remain fixed throughout the grasping process. Hence, the ensuing description of contact space will assume that $\phi = (\phi_1, \phi_2)$ is *fixed*. To ensure that the fingers are always arranged in counterclockwise order, the inter-finger shape parameters are defined for $0 < \phi_1, \phi_2 < \pi$. The pair $(q, \sigma) \in \mathbb{R}^4$ is allowed to change during the grasping process, thus enabling the hand to open and close monotonically according to the scalar parameter σ. Note that σ was chosen as the distance between Fingers 1 and 2 for convenience. Any other convex function of the finger placements could have been chosen instead.

The object $\mathcal{B}$ is assumed to be a rigid polygon that can freely translate and rotate in $\mathbb{R}^2$. The fingers are assumed to interact with $\mathcal{B}$ through *frictionless contacts.* When $\mathcal{B}$ is held by either two or three fingers such that it cannot move relative to the fingers, it is said to be *immobilized* by the fingers. This letter focuses on *squeezing grasps*, where the fingers press toward each other in order to immobilize the object $\mathcal{B}$. When $\mathcal{B}$ has some bounded mobility between the stationary fingers but cannot escape to infinity, it is said to be *caged* by the fingers.

Every immobilizing grasp of $\mathcal{B}$ is surrounded by *caging regions*. When the fingers are placed in these regions such that the formation scale is kept below some critical value, $\sigma \leq \sigma_{\max}$, the object will be caged by the fingers. Starting at such a cage and monotonically decreasing σ will lead to an immobilizing grasp of $\mathcal{B}$, while the object remains caged throughout this process. The *three-finger caging problem* considered here is to compute the critical caging grasp which allows the object to escape the hand, while maintaining the specified hand shape. The critical finger opening at this grasp, $\sigma_{\max}$, is then used to determine the *caging regions* which form the output of the three-finger caging problem.

III. Representation of Cages in the Hand's Configuration Space

A. The Free C-Space Boundary

The configuration space or *c-space* of a three-finger hand is the tuple $(q, \sigma, \phi) \in \mathbb{R}^6$. A fixed-$\phi$ finger formation corresponds to the four-dimensional slice in the hand's c-space, $\mathcal{C}_\phi = (q, \sigma) \in \mathbb{R}^4$. When the object $\mathcal{B}$ lies stationary in $\mathbb{R}^2$, it forms an *obstacle* from the hand's perspective. The object induces a *c-space obstacle* in $\mathcal{C}_\phi$ for each of the three fingers. The union of the c-space obstacles is denoted $\mathcal{CB}_\phi$. The hand's *free c-space*, $\mathcal{F}_\phi$, is the complement of $\mathcal{CB}_\phi$'s interior. The boundary of $\mathcal{F}_\phi$, denoted $\mathsf{bdy}(\mathcal{CB}_\phi)$, consists of all hand configurations at which one or more fingers contact $\mathcal{B}$'s boundary. It consists of single-finger, two-finger and three-finger contact submanifolds. Of particular interest are the two-finger contact submanifolds, together with their intersection curves along the three-finger contact submanifolds, defined next.

Definition 2: The three-finger hand's *contact submanifold*, denoted $\mathcal{S}_\phi$, is the union of the submanifolds of $\mathsf{bdy}(\mathcal{CB}_\phi)$ associated with all hand configurations, (q,σ), at which at least two fingers contact the stationary object $\mathcal{B}$.

B. C-Space Representation of Cage Formations

At an *immobilizing grasp* of $\mathcal{B}$, $(q_0,\sigma_0)\in\mathcal{S}_\phi$, the point q_0 is completely surrounded by the c-obstacles in the σ_0-slice of $\mathcal{C}_\phi$. A small increase of σ above σ_0 causes the c-obstacles to move away from each other, forming a bounded free c-space *cavity* in the corresponding fixed-σ slice in $\mathcal{C}_\phi$. This cavity allows the fixed-σ hand to locally move in a bounded neighborhood of the immobilizing grasp, while $\mathcal{B}$ is kept stationary.

Further increasing σ causes the cavity to expand until eventually a *puncture point* appears on its boundary. At this instant, the puncture point might connect the cavity to an adjacent cavity associated with a different immobilizing grasp of $\mathcal{B}$, thus being an *intermediate puncture point*, or can connect the cavity to infinity, in which case it represents the *escape puncture point* with critical value $\sigma_{\max}$. The latter puncture point will be denoted $(q_1,\sigma_1)\in\mathcal{S}_\phi$, where $\sigma_1=\sigma_{\max}$. Note that both immobilizing and puncture grasps are feasible equilibrium grasps of $\mathcal{B}$. At these grasps the finger forces directed along $\mathcal{B}$'s inward contact normals apply zero net wrench on $\mathcal{B}$.

Starting at an immobilizing grasp (q_0,σ_0) and increasing σ up to σ_1 allows the fixed shape hand to move in the largest bounded area in $\mathbb{R}^2$, while maintaining a cage around $\mathcal{B}$. The three fingers move accordingly in three bounded regions which form the *caging regions* in $\mathbb{R}^2$. To complete the picture in terms of caging theory, the union of the fixed-σ cavities in $\mathcal{F}_\phi$ for $\sigma_0\le\sigma\le\sigma_1$ is termed the *caging set* in $\mathcal{F}_\phi$. Given a user specified immobilizing grasp (q_0,σ_0,ϕ_0) of the object $\mathcal{B}$, our objective is to compute the critical hand size $\sigma_1=\sigma_{\max}$ for the fixed hand shape $\phi=\phi_0$, then use it to determine the caging regions that surround the immobilizing grasp of $\mathcal{B}$.

IV. The Three-Finger Hand Contact Space

The hand's *contact space* parametrizes the contact submanifold, $\mathcal{S}_\phi$, in terms of the finger contacts along the object boundary. Contact space will be used to find an *escape path* along which $\sigma\le\sigma_{\max}$, that starts at (q_0,σ_0), passes through (q_1,σ_1), and ends at a three-finger *pinching-grasp* at which $\sigma=0$, thus reducing the caging problem from a search in $\mathbb{R}^4$ to a search in contact space.

A. The Two-Finger Contact Spaces

The object boundary is parameterized by arclength in counterclockwise direction using the scalar parameter $s\in[0,L]$ (see Fig. 3(a)). Let fingers i and $i+1$ contact the object boundary at points $p(s_i)$ and $p(s_{i+1})$, where $i=1,2,3 \bmod 3$ (Fig. 3(a)). A two-finger contact space is defined as follows.

Definition 3: Let a polygonal object $\mathcal{B}$ be contacted by fingers i and $i+1$ at $p(s_i)$ and $p(s_{i+1})$. A *two-finger contact space* is the parameterization of all two-finger contacts along the object's outer boundary, given by the set $\mathcal{U}_{i,i+1}=[0,L]\times[0,L]$ in the (s_i,s_{i+1}) plane, where L is the perimeter of $\mathcal{B}$.

A three-finger hand has *three* two-finger contact spaces: $\mathcal{U}_{12}$, $\mathcal{U}_{23}$, and $\mathcal{U}_{31}$. Each two-finger contact space $\mathcal{U}_{i,i+1}$ is partitioned into *rectangles*, denoted $\mathcal{R}_{jk}$, each representing all pairs of points along edges j and k of $\mathcal{B}$, contacted by fingers i and $i+1$ respectively. The *diagonal* $\Delta_{i,i+1}=\{(s_i,s_{i+1})\in\mathcal{U}_{i,i+1}: s_i=s_{i+1}\}$ represents all two-finger *pinching configurations* along the object boundary.

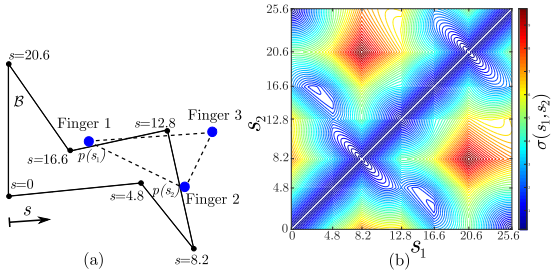

Fig. 3. (a) The s-parameterization of the object boundary. When Fingers 1 and 2 contact the object at $p(s_1)$ and $p(s_2)$, the position of Finger 3 is uniquely determined by the hand shape. (b) The two-finger contact space $\mathcal{U}_{12}$, overlaid with the contours of $\sigma(s_1,s_2)$ induced by the object $\mathcal{B}$. Note the partition of $\mathcal{U}_{12}$ into rectangles according to the object edge pairs.

The inter-finger distance function is defined in $\mathcal{U}_{i,i+1}$ as $d_{i,i+1}=\|p(s_i)-p(s_{i+1})\|$. The three inter-finger distances are related to each other by the law of sines:

$$\frac{d_{12}}{\sin\phi_3}=\frac{d_{23}}{\sin\phi_1}=\frac{d_{31}}{\sin\phi_2} \tag{1}$$

where $\phi_3=\phi_2-\phi_1$. Since the formation scale σ is equal to d_{12} (see Fig. 1), it can be computed as a function of the contact points of any two fingers along the edges of $\mathcal{B}$ using Eq. (1) and the inter-finger distance function, as defined next in the individual two-finger contact spaces.

Definition 4: Let $\mathcal{U}_{i,i+1}$ be the two-finger contact space associated with fingers i and $i+1$. The *formation scale function* is the scalar valued function $\sigma:\mathcal{U}_{i,i+1}\to\mathbb{R}$ given by

$$\sigma(s_i,s_{i+1})=\frac{\sin\phi_3}{\sin\phi_{i+2}}\|p(s_i)-p(s_{i+1})\|$$

The contours of the formation scale function in $\mathcal{U}_{i,i+1}$ are depicted in Fig. 3(b). Note that $\sigma(s_i,s_{i+1})$ is non-negative and continuous on $\mathcal{U}_{i,i+1}$, and attains a global minimum of zero along the diagonal $\Delta_{i,i+1}$. Moreover, $\sigma(s_i,s_{i+1})$ forms a *convex function* in the individual contact space rectangles (Fig. 3(b)). Also note that the contours of $\sigma(s_i,s_{i+1})$ have identical layouts in $\mathcal{U}_{12}$, $\mathcal{U}_{23}$ and $\mathcal{U}_{31}$.

B. The Third-Finger Contact Space Obstacles

When fingers i and $i+1$ move along the stationary object boundary, the third finger moves accordingly so that the chosen hand shape, $\phi=\phi_0$, is maintained. Since the third finger may *not* penetrate the stationary object $\mathcal{B}$ during this motion, *contact space obstacles* are induced in $\mathcal{U}_{i,i+1}$. A graphical procedure for constructing these obstacles is next described. The position of finger i along an edge of $\mathcal{B}$ is given by:

$$p(s_i)=p_i^0+(s_i-s_i^0)\,t_i\;s_i\in[s_i^0,s_i^0+L_i] \tag{2}$$

where $p_i^0=p(s_i^0)$ is the edge's initial vertex, t_i is the edge unit tangent, and L_i is the edge length. The position of finger $i+1$ along some other edge of $\mathcal{B}$ is described by $p(s_{i+1})$. The third finger position, denoted X_{i+2}, is computed by rotating the vector $p(s_{i+1})-p(s_i)$ about $p(s_i)$, and using Eq. (1):

$$X_{i+2}(s_i,s_{i+1})=p(s_i)+R_{i,i+1}\big(p(s_{i+1})-p(s_i)\big)$$

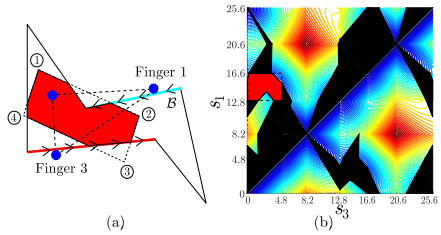

Fig. 4. (a) The third-finger parallelogram, with non-feasible third finger positions marked by a red region inside $\mathcal{B}$. (b) Applying T^{-1} on the red region gives the contact space obstacle in the rectangle $\mathcal{R}_{1,4}$. The contact space $\mathcal{U}_{31}$ is overlaid with all contact space obstacles induced by Finger 2.

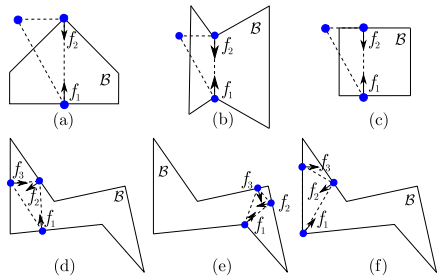

Fig. 5. Equilibrium grasps of $\mathcal{B}$: (a) A grasp at a vertex and an opposing edge. (b) A grasp at two opposing vertices. (c) A grasp at parallel edges. (d) A 3-finger immobilizing grasp at interior edge points. (e) A 3-finger immobilizing grasp at a vertex and two edge points. (f) A 3-finger puncture grasp.

where $R_{i,i+1} = \frac{\sin\phi_{i+1}}{\sin\phi_{i+2}} R_{\phi_i}$, and R_{ϕ_i} is a counterclockwise 2×2 rotation matrix by ϕ_i radians for $i = 1, 3$, and by $\pi - \phi_2$ radians for $i = 2$. Substituting for $p(s_i)$ and $p(s_{i+1})$ and defining $\vec{u} = p_i^0 + R_{i,i+1}(p_{i+1}^0 - p_i^0)$, $\vec{v} = (I - R_{i,i+1})t_i$, $\vec{w} = R_{i,i+1}t_{i+1}$ gives:

$$X_{i+2}(s_i, s_{i+1}) = \vec{u} + \left(s_i - s_i^0\right)\vec{v} + \left(s_{i+1} - s_{i+1}^0\right)\vec{w} \qquad (3)$$

which is a parametrization of a *parallelogram,* with vertices at $\vec{u}$, $\vec{u} + L_i\vec{v}$, $\vec{u} + L_{i+1}\vec{w}$ and $\vec{u} + L_i\vec{v} + L_{i+1}\vec{w}$. Each parallelogram is associated with a particular contact space rectangle, $\mathcal{R}_{jk}$, and a particular pair of fingers. The parallelograms are next used to determine the following contact space obstacles.

Definition 5: When fingers i and $i+1$ contact the stationary object $\mathcal{B}$, the *contact space obstacle,* denoted $\mathcal{CB}_{i,i+1}$, consists of all points $(s_i, s_{i+1}) \in \mathcal{U}_{i,i+1}$ at which the third finger, $X_{i+2}(s_i, s_{i+1})$, lies within the object $\mathcal{B}$.

The 2×2 matrix $T = [\vec{v}\ \vec{w}]$ maps the rectangle $\mathcal{R}_{jk}$ to a parallelogram in $\mathbb{R}^2$, according to Eq. (3). The contact space obstacles can be computed by mapping the intersection of $\mathcal{B}$ and the area enclosed by the parallelogram back to $\mathcal{R}_{jk}$ (Fig. 4):

$$\begin{pmatrix} s_i \\ s_{i+1} \end{pmatrix} = \begin{pmatrix} s_i^0 \\ s_{i+1}^0 \end{pmatrix} + T^{-1}(X_{i+2} - \vec{u}) \in \mathcal{R}_{jk} \qquad (4)$$

where the point X_{i+2} varies in the parallelogram in $\mathbb{R}^2$. Fig. 4(b) shows the entire collection of contact space obstacles, depicted as black regions in $\mathcal{U}_{31}$. Note that the obstacle layout is *different* in each of the three two-finger contact spaces, and depends on the hand shape, $\phi = \phi_0$.

Since T^{-1} maps lines to lines, we obtain the following property which is the basis for the contact space approach to formationally similar three-finger hands.

Lemma 4.1: The contact space obstacles of a fixed-ϕ three finger hand form **polygonal regions** in the two-finger contact spaces $\mathcal{U}_{i,i+1}$, where $i = 1, 2, 3 \bmod 3$.

Finally consider how the two-finger contact spaces relate to each other. The boundary curves of the contact space obstacles consist of configurations at which all three fingers contact the object $\mathcal{B}$. These curves can be regarded as *gluing seams,* which connect the two-finger contact spaces to each other. A formal definition of the hand's contact space follows.

Definition 6: The *three-finger hand contact space*, denoted $\mathcal{U}$, is the union of the individual two-finger contact spaces $\mathcal{U}_{12}$, $\mathcal{U}_{23}$ and $\mathcal{U}_{31}$, glued together along the contact space obstacle boundaries:

$$\mathcal{U} = \mathcal{U}_{12} \cup \mathcal{U}_{23} \cup \mathcal{U}_{31} / \mathsf{bdy}\,(\mathcal{CB}_{i,i+1}) \sim \mathsf{bdy}\,(\mathcal{CB}_{i+1,i+2})$$

where $\mathsf{bdy}\,(\mathcal{CB}_{i,i+1})$ is the contact space obstacle boundary, and the quotient identifies the corresponding point triplets on $\mathsf{bdy}\,(\mathcal{CB}_{12})$, $\mathsf{bdy}\,(\mathcal{CB}_{23})$ and $\mathsf{bdy}(\mathcal{CB}_{31})$.

Contact space $\mathcal{U}$ thus consists of three two-finger contact spaces, such that each point on a contact space obstacle boundary is common to the three two-finger contact spaces. It can be verified that contact space $\mathcal{U}$ is *topologically equivalent* (homeomorphic) to the contact submanifold, $\mathcal{S}_\phi$, in the hand's free c-space $\mathcal{F}_\phi$. Hence, a search for an escape path in $\mathcal{S}_\phi$ can be performed in the hand's contact space $\mathcal{U}$. Note that rendering the full contact space as in Fig. 4(b) is done only for visualization purposes, and is *not* required for constructing the ensuing caging graph.

C. Contact Space Representation of Immobilizing, Puncture and Pinching Grasps

The points in $\mathcal{U}$ that correspond to immobilizing, puncture, and pinching grasps will become nodes of the caging graph. The following proposition characterizes the immobilizing and puncture grasps in contact space.

Proposition 4.2 ([4]): The immobilizing and puncture grasps of $\mathcal{B}$ appear respectively as *local minima* and *saddle points* of the formation scale function $\sigma(s_i, s_{i+1})$ in contact space $\mathcal{U}$.

Recall from Section II that immobilizing and puncture grasps are equilibrium grasps of $\mathcal{B}$. The two-finger equilibrium grasps of a polygon $\mathcal{B}$ are of three types: (1) One finger contacts a vertex and the other contacts an opposing edge of $\mathcal{B}$ (Fig. 5(a)). (2) The two fingers contact opposing vertices of $\mathcal{B}$ (Fig. 5(b)). (3) The two fingers contact opposing parallel edges of $\mathcal{B}$ (Fig. 5(c)). This leads to the following lemma.

Lemma 4.3 ([17]): The two-finger equilibrium grasps of $\mathcal{B}$ are extremum points of $\sigma(s_i, s_{i+1})$, which lie at a *corner point* of $\mathcal{R}_{jk}$, or at an *interior point* of a bounding line of $\mathcal{R}_{jk}$ in $\mathcal{U}$.

The three-finger equilibrium grasps correspond to extremum points of $\sigma(s_i, s_{i+1})$, located on the boundary of the contact space obstacles. Since $\sigma(s_i, s_{i+1})$ is convex in each contact space rectangle, $\mathcal{R}_{jk}$, it can have at most one extremum point in the interior of each line segment along a contact space obstacle boundary, leading to the following lemma.

Lemma 4.4: The three-finger equilibrium grasps of $\mathcal{B}$ are extremum points of $\sigma(s_i, s_{i+1})$, which lie at a *corner point* of a contact space obstacle, or at a single *interior point* of a bounding line segment of a contact space obstacle in $\mathcal{U}$.

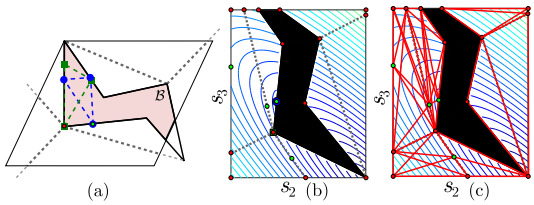

Fig. 6. (a) Part of the convex partitioning of the area surrounding the polygonal object $\mathcal{B}$, which lies inside a third-finger parallelogram. (b) All polygonal shapes inside the parallelogram are mapped under T^{-1} to polygonal shapes in the corresponding contact space rectangle $\mathcal{R}_{5,6}$. (c) The caging graph nodes in each convex region are connected with graph edges.

The three types of three-finger equilibrium grasps are illustrated in Fig. 5(d)–(f). The local minimum and saddle point of $\sigma(s_i, s_{i+1})$ in contact space corresponding to the grasps in Fig. 5(d) and (f) are respectively marked by a blue circle and a green square in Fig. 6(b). All of these extremum points will become nodes of the caging graph.

Finally consider the pinching configurations in $\mathcal{U}$. When the hand maintains a fixed triangular shape, $\phi = \phi_0$, and the distance between any two fingers approaches zero, the hand forms a three-finger pinching configuration. The caging graph *escape nodes* will be located on the two-finger diagonals, $\Delta_{i,i+1}$, which form part of the boundary of the c-obstacle $\mathcal{CB}_{i,i+1}$ (Fig. 4(b)). At these nodes the three fingers are pinched together, and the hand can escape to infinity.

V. The Caging Graph of a Three-Finger Hand

This section describes the *caging graph*, used for finding an escape path in the hand's contact space $\mathcal{U}$. The construction of the graph will be described in two stages: (1) the nodes and edges which are embedded in $\mathcal{U}$, and (2) the *tunnel edges* which ensure equivalence between the graph and the hand's free c-space $\mathcal{F}_\phi$.

A. The Contact Space Caging Graph

The caging graph will be computed *incrementally* according to the contact space rectangles explored during the search for the escape path. For computational efficiency, the complement of the contact space obstacles in each rectangle $\mathcal{R}_{jk}$ will be partitioned into *convex regions*. To this end, the workspace surrounding the object $\mathcal{B}$ within a large enough rectangle is first partitioned into convex regions (Fig. 6(a)). We achieve this with an $O(n \log n)$ constrained Delaunay triangulation [26], which gives $n+4$ triangles, where n is the number of object edges. The inverse mapping T^{-1} is then applied to the induced convex partition of the area enclosed by the parallelogram corresponding to each $\mathcal{R}_{jk}$, as was done for the contact space obstacles, to obtain the convex regions in $\mathcal{R}_{jk}$ (Fig. 6(a) and (b)). Note that if a contact space rectangle contains no obstacle, there is no need to partition it into convex regions, since the rectangle already forms a convex region. The *caging graph*, denoted $G(V, E)$, is defined as follows.

Definition 7: The **nodes** of $G(V, E)$ are (1) the vertices of each convex region in an explored contact space rectangle $\mathcal{R}_{jk}$, and (2) the extremum points of $\sigma(s_i, s_{i+1})$ along the edges of these convex regions. The **edges** of G are linear segments that connect all node pairs in each of the convex regions of $\mathcal{R}_{jk}$.

By construction, all the extremum points of the formation scale function $\sigma(s_i, s_{i+1})$ in each $\mathcal{R}_{jk}$ are nodes of G. Such extremum points represent equilibrium grasps of $\mathcal{B}$. The nodes of G also include *escape nodes* located at corners of $\mathcal{R}_{jk}$ when $j = k$, as well as single nodes along the common edges of neighboring convex regions in $\mathcal{R}_{jk}$ (Fig. 6(b)). **Computation of the caging graph nodes:** Consider a contact space rectangle $\mathcal{R}_{jk}$ in a two-finger contact space $\mathcal{U}_{i,i+1}$. To compute the nodes at the *vertices* of the convex regions in $\mathcal{R}_{jk}$, the intersection of the Delaunay triangulation with the *parallelogram* corresponding to $\mathcal{R}_{jk}$ in $\mathbb{R}^2$ is first computed. The area of intersection of a triangle and a parallelogram can be computed in constant time, and has at most seven vertices and edges [27]. Hence, the intersection of all $O(n)$ triangles with the parallelogram requires $O(n)$ steps. The vertices of the areas of intersection are then mapped to $\mathcal{R}_{jk}$ using Eq. (4), to serve as nodes marked as red dots in Fig. 6(b).

Next consider the computation of the single extremum point of $\sigma(s_i, s_{i+1})$ in the interior of each edge of a convex region in $\mathcal{R}_{jk}$. Let $\vec{s}_1$ and $\vec{s}_2$ be the endpoints of a convex region edge in $\mathcal{R}_{jk}$. The edge is parameterized as $l(\lambda) = \vec{s}_1 + \lambda(\vec{s}_2 - \vec{s}_1) = \vec{s}_1 + \lambda\Delta\vec{s}$ for $\lambda \in [0, 1]$. Let $l(\lambda_{crit})$ denote the extremum point of $\sigma(s_i, s_{i+1})$ along l. Since $\sigma(s_i, s_{i+1})$ is a convex function in each $\mathcal{R}_{jk}$, its extremum points are the same as the extremum points of $\sigma^2(s_i, s_{i+1})$. Using Eq. (2), we can write the formation scale function as: $\sigma(s_i, s_{i+1}) = \rho\|A\vec{s} + \vec{b}\|$, where $\rho = \frac{\sin\phi_3}{\sin\phi_{i+2}}$, $A = [t_i \; -t_{i+1}]$, and $\vec{b} = p^0_i - p^0_{i+1} - A\vec{s}_0$, where $\vec{s}_0 = (s^0_i, s^0_{i+1})$. The gradient $\nabla\sigma^2(s_i, s_{i+1}) = 2\rho^2 A^T(A\vec{s} + \vec{b})$ at $l(\lambda_{crit})$ must be perpendicular to the edge. Hence $\nabla\sigma^2(s_i, s_{i+1})|_{\vec{s}=l(\lambda_{crit})} \cdot \Delta\vec{s} = 0$, or equivalently:

$$\lambda_{crit} = -\frac{\Delta\vec{s}^T A^T \left(A\vec{s}_1 + \vec{b}\right)}{\Delta\vec{s}^T A^T A \Delta\vec{s}} \tag{5}$$

If $\lambda_{crit} \in (0, 1)$, then $l(\lambda_{crit})$ is an extremum point within the edge, and it serves as a node (see green dots in Fig. 6(b)). Note that each node of the caging graph belongs to at least two neighboring convex regions in $\mathcal{U}$. Note too that nodes on the boundary of the contact space obstacles are three-finger nodes. The value of s_{i+2} at these nodes can be computed using Eqs. (2) and (3) to obtain: $s_{i+2} = s^0_{i+2} + t_{i+2} \cdot (\vec{u} - p^0_{i+2} + T(\vec{s} - \vec{s}_0))$. Once the nodes of a convex region are computed, all node pairs are connected with graph edges (Fig. 6(c)). Each convex region contains up to 14 nodes, and therefore a constant number of edges. Since each $\mathcal{R}_{jk}$ contains $O(n)$ convex regions, it contains $O(n)$ graph nodes as well as $O(n)$ graph edges. Hence, the caging graph might contain $O(n^3)$ nodes and $O(n^3)$ edges.

The relation of the caging graph to the sublevel structure of the formation scale function $\sigma(s_i, s_{i+1})$ in $\mathcal{U}$ is next examined. The notation $\sigma(\boldsymbol{v})$ will denote the formation scale value at the point in $\mathcal{U}$ represented by the node $\boldsymbol{v}$.

Definition 8: Let $G(V, E)$ be the caging graph over contact space $\mathcal{U}$. For each $c \geq 0$, a *c-sublevel subgraph* of G, denoted $G_c(V_c, E_c)$, is the subgraph of G whose nodes are given by $V_c = \{\boldsymbol{v} \in V : \sigma(\boldsymbol{v}) \leq c\}$.

The following theorem asserts that the caging graph captures the $\sigma(s_i, s_{i+1})$ sublevel structure of contact space $\mathcal{U}$.

Theorem 1: The caging graph $G(V, E)$ *captures the sublevel structure* of $\mathcal{U}$: there exists a contact space path in $\mathcal{U}$ between two nodes of G, $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, lying entirely in a c-sublevel set of $\sigma(s_i, s_{i+1})$ in $\mathcal{U}$ iff there exists a caging graph path between $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ which lies entirely in a c-sublevel subgraph of $G(V, E)$.

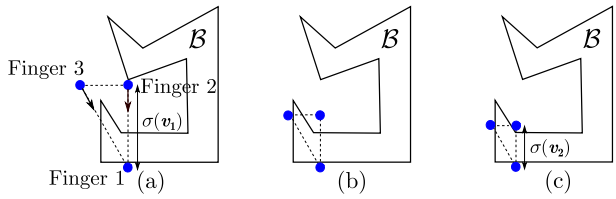

Fig. 7. Caging graph augmentation procedure. (a) The hand configuration is represented by the non-feasible local minimum node $\boldsymbol{v_1}$ in $\mathcal{U}$, with $\sigma(\boldsymbol{v_1})$. (b) While Finger 1 maintains contact and the hand orientation remains fixed, σ is decreased until Finger 3 contacts the object. This hand configuration corresponds to a point u in a convex region of a contact space rectangle. (c) A node $\boldsymbol{v_2}$ that satisfies $\sigma(\boldsymbol{v_2}) \leq \sigma(\boldsymbol{v_1})$ and is in the same convex region is found, and the tunnel edge $(\boldsymbol{v_1}, \boldsymbol{v_2})$ is added to $\overline{E}$.

Proof: Given a *contact space path* between two nodes of G, the path can be partitioned into segments according to the convex regions it passes through in each contact space rectangle $\mathcal{R}_{jk}$. Both endpoints of each path segment are first extended along the convex region's bounding lines they intersect, in the directions along which $\sigma(s_i, s_{i+1})$ *decreases*, until graph nodes are reached. The extended path segment is then replaced by a graph edge, which by convexity of $\sigma(s_i, s_{i+1})$ in each $\mathcal{R}_{jk}$ is not higher than its two endpoints. The contact space path can thus be replaced by a piecewise linear path, which passes between nodes of G and lies in the same sublevel set of $\sigma(s_i, s_{i+1})$ in $\mathcal{U}$. Conversely, any *caging graph path* between two nodes of G is also a contact space path, which lies in the same sublevel set of $\sigma(s_i, s_{i+1})$ in $\mathcal{U}$.

B. The Augmented Caging Graph

A search for an escape path in G is valid only if it yields the same result as a search in the hand's full free c-space $\mathcal{F}_\phi$. The caging graph G is sublevel equivalent to $\mathcal{U}$, and $\mathcal{U}$ is homeomorphic to the contact submanifold $\mathcal{S}_\phi$. Hence, to ensure the same result, *sublevel equivalence* between $\mathcal{S}_\phi$ and the ambient space $\mathcal{F}_\phi$ in terms of connectivity must be established. For each $c \geq 0$, let $\mathcal{F}_c = \{(q, \sigma) \in \mathcal{F}_\phi : \sigma \leq c\}$ be the sublevel set of free configurations in the hand's free c-space $\mathcal{F}_\phi$. Sublevel equivalence between $\mathcal{S}_\phi$ and $\mathcal{F}_\phi$ is defined next.

Definition 9: The contact submanifold $\mathcal{S}_\phi$ and the ambient free c-space $\mathcal{F}_\phi$ are said to be *sublevel equivalent* in terms of connectivity if for each $c \geq 0$, in each connected component of $\mathcal{F}_c$ the subset $\mathcal{F}_c \cap \mathcal{S}_\phi$ is also connected.

Consider the projection function $\pi(q, \sigma) = \sigma$, which maps the fixed-ϕ hand's configuration (q, σ) to the formation scale σ. Sublevel equivalence between $\mathcal{S}_\phi$ and the ambient space $\mathcal{F}_\phi$ breaks when *local minima* of the restriction of $\pi(q, \sigma)$ to $\mathcal{S}_\phi$ are *not* local minima of $\pi(q, \sigma)$ in the ambient space $\mathcal{F}_\phi$. These local minima correspond to grasps at which the hand can close without penetrating the object, termed *non-feasible* equilibrium grasps (Fig. 7(a)). Since $\mathcal{S}_\phi$ and $\mathcal{U}$ are homeomorphic, such configurations correspond to *non-feasible* minima of $\sigma(s_i, s_{i+1})$ in $\mathcal{U}$. To ensure sublevel equivalence of $\mathcal{S}_\phi$ with $\mathcal{F}_\phi$, a graph node at a non-feasible local minimum of $\sigma(s_i, s_{i+1})$ in $\mathcal{U}$ must be connected to a node in a lower c-sublevel subgraph of G by a *tunnel edge*, which represents a σ-decreasing path in the ambient free c-space $\mathcal{F}_\phi$. The caging graph G *augmented* with tunnel edges is denoted $G(V, \overline{E})$. The augmentation procedure is illustrated in Fig. 7.

Each contact space rectangle contains at most one non-feasible minimum of $\sigma(s_i, s_{i+1})$, which generically occurs at a two-finger grasp. Identifying a node at such a minimum and augmenting G with the corresponding tunnel edge involves: checking which of the two contacting fingers can be detached from the object, performing two ray shootings equivalent to closing the hand to find the first contact point of the contracted fingers with the object, then locally expanding the caging graph, and finally connecting the non-feasible minimum of $\sigma(s_i, s_{i+1})$ with a node located in the same convex region as the point that is reached in contact space when the hand closes. This requires $O(n)$ steps per contact space rectangle, and might require a total of $O(n^3)$ steps for all tunnel edges in $\mathcal{U}$.

VI. The Caging Graph Search Algorithm

The augmented caging graph $G(V, \overline{E})$ is sublevel equivalent to the free c-space $\mathcal{F}_\phi$. Hence, a search for an escape path along the edges of $G(V, \overline{E})$ will give the same result as a search in $\mathcal{F}_\phi$. The caging problem, i.e., the search for an escape path, is thus reduced to a simple graph search. The search is based on the following intuitive notion. The formation scale function $\sigma(s_i, s_{i+1})$ can be considered as a potential energy function. Opening the hand while at least two fingers maintain contact with the object requires an increase of energy. The caging algorithm can therefore be considered as a search for a path which connects one of the potential energy function's local minima (the initial immobilizing grasp) to its global minimum set Δ (comprised of all pinching configurations), while passing through the saddle point which requires the least increase in energy (the escape puncture grasp).

The search procedure, summarized below as the I-CAGE algorithm, requires as input the n vertices of the polygonal object $\mathcal{B}$ and the target immobilizing grasp (q_0, σ_0, ϕ_0), which serves as the initial hand configuration in the search. The algorithm uses two node lists: the *open list* $\mathcal{O}$ which holds the explored nodes, and the *closed list* $\mathcal{C}$ which holds the escape path nodes. The procedure starts by initializing the data structures with the target immobilizin grasp. Then, each search iteration is comprised of two stages: (1) Locally expanding the caging graph. This stage is performed only if the current node lies in a contact space rectangle which was not explored, and requires $O(n)$ computation steps. (2) Searching the caging graph for the next escape path node. This stage is performed at each search iteration, and involves adding the unexplored neighbor nodes of the current node to the open list, and finding the node with minimal σ-value in the updated list. Since contact space is partitioned into convex regions, each graph node can have at most $V_{\max}$ neighboring nodes, where $V_{\max}$ is a *constant.* If contact space is fully explored, the caging algorithm requires $O(n^3)$ search iterations for the $O(n^3)$ graph nodes, while sorting the open list requires $O(n^3 \log n)$ steps. The I-CAGE algorithm thus requires $O(n^3 \log n)$ computation steps.

Once the search ends, the local maxima of the plot of $\sigma(\boldsymbol{v})$ for $\boldsymbol{v} \in \mathcal{C}$ versus the search iteration number correspond to all intermediate and escape puncture grasps of $\mathcal{B}$, where the escape grasp has the maximal value of $\sigma_{\max}$. The local minima correspond to neighboring immobilizing grasps along the object boundary. Beyond $\sigma_{\max}$, the cage surrounding the object is broken and the object can escape the hand to infinity. Once the escape puncture grasp has been computed, the *caging regions* can be rendered as 2D regions [23], as next illustrated with an execution example.

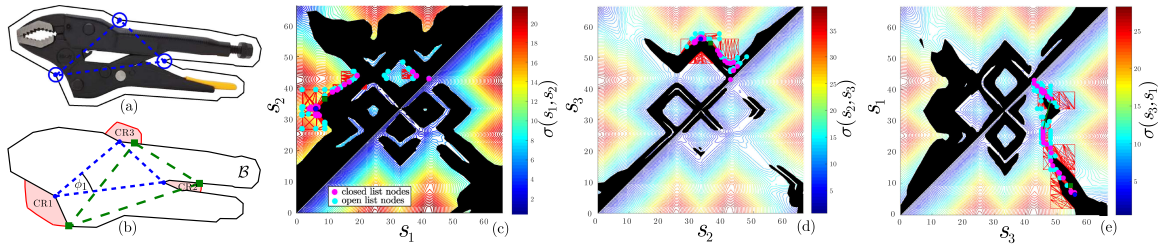

Fig. 8. Execution example of the I-CAGE algorithm on the three-finger caging example (see text). Magnification by zooming is recommended for viewing.

I-CAGE: Incremental Caging Graph Search Algorithm.

Input: n vertices of $\mathcal{B}$, target immobilizing grasp (q_0, σ_0, ϕ_0).
Data Structures: open list $\mathcal{O}$, closed list $\mathcal{C}$, node set V, edge set $\overline{E}$, boolean matrix of explored contact space rectangles $M_{n\times n\times 3}$.
Initialization: $\mathcal{O}=\emptyset, \mathcal{C}=\emptyset, V=\emptyset, \overline{E}=\emptyset, M=\mathbf{0}$.
Partition the area surrounding $\mathcal{B}$ in $\mathbb{R}^2$ into *convex regions*.
Identify a contact space rectangle $\mathcal{R}_{jk}$ in a two-finger contact space $\mathcal{U}_{i,i+1}$ that contains the point p corresponding to (q_0, σ_0).
Function: **ExploreRectangle**$([j,k],\mathcal{U}_{i,i+1})$ {
  Compute all graph nodes and edges in $\mathcal{R}_{jk}$, add them to V and $\overline{E}$, and set $M(j,k,\mathcal{U}_{i,i+1})=1$ }
Set the node at p as the current node $\boldsymbol{v}$, and mark $\boldsymbol{v}$ as explored.
Search procedure:
while $\sigma(\boldsymbol{v})>0$ **do** /* *while no pinching grasp was reached* */
  Stage I – Contact space exploration:
  Find the unexplored rectangles $\mathcal{R}_{jk}$ containing $\boldsymbol{v}$.
  for each $\mathcal{R}_{jk}$ **do**
    ExploreRectangle $([j,k],\mathcal{U}_{i,i+1})$
  end for
  if $\boldsymbol{v}$ is a non-feasible local minimum of $\sigma(s_i, s_{i+1})$ in $\mathcal{U}$ **then:**
    Compute the point u in $\mathcal{U}$ that the fingers will reach when the hand closes.
    Identify a contact space rectangle $\mathcal{R}_{jk}$ in a two-finger contact space $\mathcal{U}_{i,i+1}$ which contains u.
    ExploreRectangle$([j,k],\mathcal{U}_{i,i+1})$
    Identify a node $\boldsymbol{w}$ with $\sigma(\boldsymbol{w})\le\sigma(\boldsymbol{v})$ in the convex region containing u, and add the tunnel edge $(\boldsymbol{v},\boldsymbol{w})$ to $\overline{E}$.
  end if
  Stage II - Local caging graph search:
  Add the unexplored neighboring nodes of $\boldsymbol{v}$ to $\mathcal{O}$, sorted by σ values, and mark these nodes as explored.
  Transfer $\boldsymbol{v}$ to the end of $\mathcal{C}$.
  Select the node with minimal σ-value in $\mathcal{O}$ as the current node $\boldsymbol{v}$.
end while /* $\boldsymbol{v}$ *is now an escape node* */
Transfer $\boldsymbol{v}$ to the end of $\mathcal{C}$.
Return: closed list $\mathcal{C}$ containing the escape path nodes.

VII. Three-Finger Caging-To-Grasping Example

Consider grasping the pliers depicted in Fig. 8(a), using three identical disc fingers. We demonstrate how the I-CAGE algorithm, which is available as a MATLAB implementation [25], is used for this purpose.

A. Target Immobilizing Grasp Selection

The I-CAGE algorithm requires as input the vertices of a polygonal object and a target immobilizing grasp. The vertices of $\mathcal{B}$ are obtained by computing the Minkowski sum of the pliers' outline with a disc whose diameter is equal to the fingers' radius. The resulting object is then approximated by a polygon, $\mathcal{B}$, and the center of each disc finger serves as a point finger (Fig. 8(a)). Increasing the number of edges used for the polygonal approximation will yield a more accurate result of the algorithm, with increased computational cost.

We next describe a graphical method to determine all three-finger immobilizing grasps of $\mathcal{B}$. Recall that any immobilizing grasp must form a feasible equilibrium grasp of $\mathcal{B}$: the *inward pointing* contact normals must positively span $\mathbb{R}^2$, and the lines underlying the normals must intersect at a common point. Moreover, if only edges and *concave* vertices of $\mathcal{B}$ are contacted, such equilibrium grasps are immobilizing grasps [3]. Let each polygonal edge or vertex and its corresponding (generalized) contact normal define a semi-infinite sector. Feasible three-finger equilibrium grasps exist when three semi-infinite sectors have a common area of intersection, and the three contact normals positively span $\mathbb{R}^2$. If no two of the three contact normals are anti-parallel (the generic case), the orthogonal projection of *any point* in the area of intersection onto the three edges or vertices automatically defines the three contact points of a feasible three-finger equilibrium grasp of $\mathcal{B}$. Otherwise, the lines underlying the contact normals that support a two-finger equilibrium grasp partition the area of intersection into subsets. Each subset can be sampled at a single point to check whether the contact normals positively span $\mathbb{R}^2$. If they do, any point within the subset defines a feasible three-finger equilibrium grasp of $\mathcal{B}$. Hence, there exists a huge selection of three-finger immobilizing grasps of the polygon $\mathcal{B}$.

B. Search Algorithm Execution

The immobilizing grasp in Fig. 8(a), with $\phi_0\approx(33^\circ, 130^\circ)$, was selected as the target grasp. Fig. 8(c)–(e) show the three two-finger contact spaces associated with $\mathcal{B}$, and the hand shape $\phi_0\approx(33^\circ, 130^\circ)$. The initial node corresponding to the target three-finger immobilizing grasp is marked as a blue point in each of the three contact spaces. The I-CAGE algorithm was used to compute the corresponding escape puncture grasp. After the I-CAGE algorithm was executed, the caging graph edges in the explored contact space rectangles were marked in red, the nodes that remained in the open list $\mathcal{O}$ were marked in cyan, and the escape path nodes in the closed list $\mathcal{C}$ were marked in magenta. The node corresponding to the escape puncture grasp, which forms a three-finger equilibrium grasp, is marked as a green box in each of the three contact spaces (please zoom). Note that the caging graph includes one tunnel edge, marked

in green in Fig. 8(e). The resulting escape puncture grasp, and the corresponding three *caging regions,* are shown in Fig. 8(b). The pliers are caged when each finger remains in its caging region, such that $\sigma < \sigma_{\max}$. Starting from any caging configuration and decreasing σ to σ_0 while $\phi = \phi_0$ will lead to the target immobilizing grasp, under huge uncertainty in the initial finger placements.

VIII. Conclusion

The letter considered caging polygonal objects using three-finger hands which maintain similar triangle formations. While the hand's configuration space is four dimensional for a given hand shape (i.e., fixed $\phi = (\phi_1, \phi_2)$), the letter presented a caging algorithm that can be implemented in the hand's two-dimensional *contact space* $\mathcal{U}$. Contact space contains obstacles which represent forbidden third-finger positions when two fingers contact the grasped object's boundary. After partitioning the obstacles' complement into convex regions in contact space, the letter defined a *caging graph*, G, whose nodes include all extremum points of the formation scale function $\sigma(s_i, s_{i+1})$ along the bounding lines of the convex regions, as well as all corner points of these regions. The caging graph edges consist of line segments that connect all node pairs in each convex region, as well as tunnel edges which connect local minima of $\sigma(s_i, s_{i+1})$ that represent non-feasible equilibrium grasps. The caging graph is sublevel equivalent to the hand's free c-space $\mathcal{F}_\phi$. Hence, starting from any immobilizing grasp (which forms a triangle), the caging graph is incrementally searched for the intermediate puncture grasps as well as the final puncture grasp through which the object can escape to infinity. Moreover, the search process can be intuitively observed as motion of the fingers along the grasped object boundary. Finally, the escape grasp computed by the search algorithm determines the caging regions surrounding the desired immobilizing grasp.

The following extensions are currently under investigation. While three disc fingers can immobilize all polygonal objects without parallel edges, they *cannot* safely immobilize objects such as rectangles. To immobilize such common objects, a fourth finger must be used, or one of the fingers must be *concave* at the contact point. The contact space approach can be easily extended to four-finger hands which maintain similar quadrilateral formations, by constructing the caging graph in *six* two-finger contact spaces of all finger pairs. Alternatively, we are considering variable curvature fingertip designs, that will be able to safely immobilize objects with parallel edges using three-finger hands. A more important challenge concerns 3D caging-to-grasping. We are currently adapting the contact space approach to *four-finger hands,* that will be able to cage and then grasp 3D objects using tetrahedral finger formations.

References

[1] S. B. Backus and A. M. Dollar, "An adaptive three-fingered prismatic gripper with passive rotational joints," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 668–675, Jul. 2016.

[2] R. D. Howe, A. M. Dollar, and M. Claffee, "Robots get a grip," *IEEE Spectr.*, vol. 51, no. 12, pp. 42–47, Dec. 2014.

[3] E. Rimon and J. W. Burdick, "New bounds on the number of frictionless fingers required to immobilize planar objects," *J. Robot. Syst.*, vol. 12, no. 6, pp. 433–451, 1995.

[4] E. Rimon and A. Blake, "Caging planar bodies by 1-parameter two-fingered gripping systems," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 299–318, 1999.

[5] A. Rodriguez, M. T. Mason, and S. Ferry, "From caging to grasping," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 886–900, 2012.

[6] J. Mahler, F. T. Pokorny, Z. McCarthy, A. F. van der Stappen, and K. Goldberg, "Energy-bounded caging: Formal definition and 2-D energy lower bound algorithm based on weighted alpha shapes," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 508–515, Jan. 2016.

[7] A. Varava, D. Kragic, and F. T. Pokorny, "Caging grasps of rigid and partially deformable 3-D objects with double fork and neck features," *IEEE Trans. Robot*, vol. 32, no. 6, pp. 1479–1497, Dec. 2016.

[8] R. R. Ma, W. G. Bircher, and A. M. Dollar, "Toward robust, whole-hand caging manipulation with underactuated hands," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1336–1342.

[9] Z. Wang and V. Kumar, "Object closure and manipulation by multiple cooperating mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 394–399.

[10] G. Pereira, M. Campos, and V. Kumar, "Decentralized algorithms for multi-robot manipulation via caging," *Int. J. Robot. Res.*, vol. 23, no. 7-8, pp. 783–795, 2004.

[11] A. Sudsang, F. Rothganger, and J. Ponce, "Motion planning for disc-shaped robots pushing a polygonal object in the plane," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 550–562, Aug. 2002.

[12] W. Wan, B. Shi, Z. Wang, and R. Fukui, "Multirobot object transport via robust caging," *IEEE Trans. Syst., Man, Cybern., Syst.*, 2017, to be published.

[13] D. Hirano, H. Kato, and N. Tanishima, "Caging-based grasp with flexible manipulation for robust capture of a free-floating target," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5480–5486.

[14] Q. M. Ta and C. C. Cheah, "Coordinative optical manipulation of multiple microscopic objects using micro-hands with multiple fingertips," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5870–5875.

[15] M. Vahedi and A. F. van der Stappen, "Caging polygons with two and three fingers," *Int. J. Robot. Res.*, vol. 27, nos. 11/12, pp. 1308–1324, 2008.

[16] P. Pipattanasomporn and A. Sudsang, "Two-finger caging of nonconvex polytopes," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 324–333, Apr. 2011.

[17] T. F. Allen, E. Rimon, and J. W. Burdick, "Two-finger caging of polygonal objects using contact space search," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1164–1179, Oct. 2015.

[18] J. Erickson, S. Thite, F. Rothganger, and J. Ponce, "Capturing a convex object with three discs," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1133–1140, Dec. 2007.

[19] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "Grasping by caging: A promising tool to deal with uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 5142–5149.

[20] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "A new grasping by caging solution using eigen-shapes and space mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1566–1573.

[21] W. Wan and R. Fukui, "Efficient planar caging test using space mapping," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 278–289, Jan. 2018.

[22] P. Pipattanasomporn, T. Makapunyo, and A. Sudsang, "Multifinger caging using dispersion constraints," *IEEE Trans. Robot*, vol. 32, no. 4, pp. 1033–1041, Aug. 2016.

[23] H. A. Bunis, E. D. Rimon, T. F. Allen, and J. W. Burdick, "Equilateral three-finger caging of polygonal objects using contact space search," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 919–931, 2018.

[24] H. A. Bunis, E. D. Rimon, Y. Golan, and A. Shapiro, "Caging polygonal objects using equilateral three-finger hands," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1672–1679, Jul. 2017.

[25] H. A. Bunis and E. D. Rimon, "MATLAB implementation of the caging graph search algorithm," 2018. [Online]. Available: https://doi.org/10.24433/CO.7de1108c-81bf-456e-ac6c-9aba7ab652eb

[26] L. P. Chew, "Constrained delaunay triangulations," *Algorithmica*, vol. 4, no. 4, pp. 97–108, 1989.

[27] M. I. Shamos, "Computational geometry," Ph.D. dissertation, Dept. Comput. Sci., Yale Univ., New Haven, CT, USA, 1978.