Two and Three Finger Caging of Polygons and Polyhedra

Thesis by

Thomas F. Allen

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2016
(Defended October 27, 2015)

For Serena

Acknowledgements

I would like to express my heartfelt gratitude to my adviser, Joel Burdick, for his support and guidance during my stay at Caltech. He is an exceptional adviser, who cares about his students at least as much as he cares about their work. His advice and direction have substantially shaped my academic path, and this work, for the better. He provided me with excellent opportunities to learn robotics both as an engineer and as an academic, while insulating me from worries about where my funding will come from.

I would like to thank my co-author, Elon Rimon, for his substantial contribution each of the papers on which this thesis is based, and for not tolerating my hand-waving and challenging me to be mathematically rigorous.

My first year at Caltech would have been impossibly difficult without the support of my friends and classmates, who spent long hours and late nights with me in SFL: Gerry, Aaron, Trevor, and Jeff.

Thanks to all of my labmates, especially Jeffrey, Melissa, Matt, Rangoli, and Krishna, for many interesting discussions about food and Emacs, and for listening to long-winded explanations of my work, even when it was essentially unrelated to theirs. Special thanks goes to Krishna for his friendship, insightful discussions about robotics, coffee, and weight training, and for being my point man while finishing this thesis from northern California.

Thanks to my rock climbing friends, Gerry, Erick, Kevin, and Ernie, as well as my wife, Serena. They helped me unwind from Caltech, climbing with me at Holcomb Valley Pinnacles, the Palm Springs Tramway, and the gyms of the Inland Empire.

Special thanks to Paul Backes and NASA's Jet Propulsion Laboratory for supporting my research and giving me the opportunity to work on an amazing robotics project, DARPA's Autonomous Robotic Manipulation competition.

My time at JPL was exceptional due to the patient mentorship of Nick Hudson, Jeremy Ma, Paul Hebert, and Max Bajracharya. There I learned invaluable lessons about real robotic systems, and their camaraderie and technical excellence serves as a model of the engineer I want to be. I don't know of a better way to learn robotics than working in their lab.

Much thanks to Maria Koeper and Sonya Lincoln for helping me keep tabs on my adviser and

keeping everything running smoothly in the Burdick group, and to Cheryl Geer for tolerating and fixing my missed deadlines and bureaucratic blunders.

Much thanks to my understanding boss, Kevin Albert. In addition to creating and managing an excellent team, he has never pressured me to prioritize work over my academic obligations.

I want to acknowledge the support of my loving family, especially my parents, for providing me with the opportunities I needed to be here today. In addition, my grandfather, Tom, has served as a role model, both academically and otherwise, and it is his path through UC Berkeley and Caltech that I have followed.

Finally, I want to acknowledge my lovely wife, Serena. From visiting me during late night homework sessions and proofreading my academic papers, to providing love, companionship, and unconditional support throughout, I couldn't have done this without her tireless support.

Abstract

Multi-finger caging offers a rigorous and robust approach to robot grasping. This thesis provides several novel algorithms for caging polygons and polyhedra in two and three dimensions. Caging refers to a robotic grasp that does not necessarily immobilize an object, but prevents it from escaping to infinity. The first algorithm considers caging a polygon in two dimensions using two point fingers. The second algorithm extends the first to three dimensions. The third algorithm considers caging a convex polygon in two dimensions using three point fingers, and considers robustness of this cage to variations in the relative positions of the fingers.

This thesis describes an algorithm for finding all two-finger cage formations of planar polygonal objects based on a contact-space formulation. It shows that two-finger cages have several useful properties in contact space. First, the critical points of the cage representation in the hands configuration space appear as critical points of the inter-finger distance function in contact space. Second, these critical points can be graphically characterized directly on the objects boundary. Third, contact space admits a natural rectangular decomposition such that all critical points lie on the rectangle boundaries, and the sublevel sets of contact space and free space are topologically equivalent. These properties lead to a caging graph that can be readily constructed in contact space. Starting from a desired immobilizing grasp of a polygonal object, the caging graph is searched for the minimal, intermediate, and maximal caging regions surrounding the immobilizing grasp. An example constructed from real-world data illustrates and validates the method.

A second algorithm is developed for finding caging formations of a 3D polyhedron for two point fingers using a lower dimensional contact-space formulation. Results from the two-dimensional algorithm are extended to three dimension. Critical points of the inter-finger distance function are shown to be identical to the critical points of the cage. A decomposition of contact space into 4D regions having useful properties is demonstrated. A geometric analysis of the critical points of the inter-finger distance function results in a catalog of grasps in which the cages change topology, leading to a simple test to classify critical points. With these properties established, the search algorithm from the two-dimensional case may be applied to the three-dimensional problem. An implemented example demonstrates the method.

This thesis also presents a study of cages of convex polygonal objects using three point fingers.

It considers a three-parameter model of the relative position of the fingers, which gives complete generality for three point fingers in the plane. It analyzes robustness of caging grasps to variations in the relative position of the fingers without breaking the cage. Using a simple decomposition of free space around the polygon, we present an algorithm which gives all caging placements of the fingers and a characterization of the robustness of these cages.

Contents

Acknowledgements **iv**

Abstract **vi**

Contents **x**

List of Figures **xiv**

List of Algorithms **xiv**

Notation **xv**

1 Introduction **1**

1.1 Motivation 1

1.2 Review of Existing Literature 2

1.2.1 Grasping 2

1.2.2 Caging 3

1.3 Caging Basics 5

1.4 Current Uses and Limitations of Caging 6

1.5 Thesis Organization and Contributions 7

2 Two Fingers in Two Dimensions **9**

2.1 Introduction 9

2.2 Preliminaries and Problem Definition 10

2.3 Contact Space Formulation of Caging 14

2.4 Contact Space Representation of the Caging Sets 15

2.4.1 Two-finger Equilibrium Grasps in Contact Space 16

2.4.2 The Inter-finger Distance Function in Contact Space 18

2.5 Rectangular Decomposition of Contact Space 21

2.6 The Contact Space Caging Graph 22

2.6.1 The Caging Graph 23

2.6.2 The Augmented Caging Graph . . . 26
2.7 The Contact Space Algorithm . . . 30
2.8 Algorithm Walk Through . . . 32
2.9 Graphical Depiction of Caging Set as Two Capture Regions . . . 34
2.10 Caging Set Computational Example . . . 35
2.11 Extensions of the Caging Algorithm . . . 36
2.12 Summary and Extensions . . . 39

3 Two Fingers in Three Dimensions **41**
3.1 Introduction . . . 41
3.2 Preliminaries and Problem Statement . . . 42
3.3 Contact-Space Reformulation of Caging . . . 43
3.4 The Caging Graph . . . 44
3.4.1 Location of Important Grasps . . . 45
3.5 The Caging Algorithm . . . 46
3.5.1 Analysis of the Closed List . . . 46
3.6 An Example . . . 47
3.7 Characterization of Nodes of G . . . 47
3.8 A Catalog of Immobilizing and Puncture Grasps . . . 52
3.9 Sublevel Equivalence . . . 54
3.9.1 Sublevel Equivalence of U and G . . . 54
3.9.2 The Sublevel Equivalence of U and F . . . 56
3.9.3 Sublevel Equivalence at False Immobilizing Grasps . . . 57
3.9.4 Sublevel Equivalence at False Puncture Grasps . . . 57
3.9.5 Tunnel Curve Construction . . . 57
3.10 Summary . . . 58

4 Three Finger Caging **59**
4.1 Preliminaries and Robust Caging . . . 60
4.1.1 Motivational Example . . . 61
4.1.2 Robust Caging Definition . . . 64
4.2 Single Triad Caging . . . 65
4.3 Multi-Triad Caging . . . 66
4.4 Divisions of Shape Space . . . 67
4.4.1 Puncture Manifolds . . . 69
4.4.2 Immobilizing Manifolds . . . 71
4.4.3 Other Grasps . . . 71

4.5 A Test of Caging Status 74
4.5.1 Test for Feasibility 75
4.5.2 Test for Escape 75
4.6 Example 75
4.7 Computational Complexity 78
4.8 Conclusion 78

5 Conclusion **80**
5.1 Summary of Thesis Contributions 80
5.2 Opportunities for Future Work 81

Appendices **83**

A Stratified Morse Theory Proofs **84**

Bibliography **88**

List of Figures

1.1 A robot attempts to open a door. With an immobilizing grasp, the robots gripper must accurately follow the trajectory of the door handle, which may prove difficult. Caging the door handle gives the robot significantly more freedom in gripper trajectory while still guaranteeing that the door may be opened. 6

1.2 A robot arm attempts to grasp a pipe during the DARPA Autonomous Robotic Manipulation program. If the robot bumps the pipe it may fall over and roll away. Achieving a caging grasp prior to closing the gripper can guarantee that the object does not escape during the grasping motion. 7

2.1 The immobilizing grasp along with the initial, intermediate, and maximal caging sets of a polygonal object. Note that each of these sets is also subject to a constraint on the maximum inter-finger distance, which cannot be represented in two dimensions. . 13

2.2 Critical points of $d(s_1, s_2)$ which are local minima in $\mathcal{U}$. 19

2.3 Critical points of $d(s_1, s_2)$ which are saddles in $\mathcal{U}$. 19

2.4 Critical points of $d(s_1, s_2)$ which are local maxima in $\mathcal{U}$. 19

2.5 A subset of the immobilizing grasps, puncture grasps, and escape points of a polygonal object. Values of the boundary parameter s are shown at selected vertices. 20

2.6 Contours of $d(s_1, s_2)$ in contact space $\mathcal{U}$ for the object shown in Figure 2.5. Note that $\mathcal{U}$ represents a topological 2-torus. 20

2.7 Contours of $d(s_1, s_2)$ for the object in Figure 2.5. Contact space rectangles are delineated with dashed lines. The nodes of G are marked with circles. 24

2.8 The caging graph G for the object in Figure 2.5. Circles represent the nodes of G, shown in Figure 2.7. Lines represent the edges of G. Note that the graph has the topology of a 2-torus: nodes on the left edge correspond to nodes on the right edge, and similarly for nodes on the top and bottom edges. 24

2.9 A polygonal object $\mathcal{B}$ with a handle-like feature. Selected grasps are shown. 28

2.10 Contact space contours of $d(s_1, s_2)$ and selected grasps for the object of Figure 2.9. Portions of $\mathcal{U}_{\leqslant d*}$ are shown shaded; the two disjoint regions (which are connected in the free c-space) are problematic. The dotted line represents a *tunnel curve* which does not lie in contact space $\mathcal{U}$. 28
2.11 A portion of the augmented caging graph for the object of Figure 2.9, corresponding to the contact space region shown in Figure 2.10. The tunnel curve edge is depicted as a thick red line. 29
2.12 Important grasps discovered during the exploration of the polygon shown in physical space. 33
2.13 Important grasps discovered during the exploration of the polygon shown in contact. . 33
2.14 The capture regions surrounding an immobilizing grasp of $\mathcal{B}$. 34
2.15 Input data to the caging algorithm: (a) 3D triangular mesh data, and (b) a 2D polygonal projection of the point cloud data, with arc-length value of selected vertices. . . . 35
2.16 Critical grasps and escape point for the flashlight in physical space. 36
2.17 A portion of contact space $\mathcal{U}$ for the flashlight, showing nodes of the open list $\mathcal{O}$ and the closed list $\mathcal{X}$. Critical grasps in the closed list $\mathcal{X}$ are shown with the same symbols as in Figure 2.16. 37
2.18 A graph of $d(s_1, s_2)$ at the nodes of the closed list $\mathcal{X}$, showing the exploration of the object's boundary along with critical grasps and the escape point. The graph's local maxima indicate puncture grasps (see text). 37
2.19 Two-finger immobilization and its puncture grasp. (a) Both fingers are placed within a hole of $\mathcal{B}$. (b) One finger is placed in a hole, the other finger is placed outside $\mathcal{B}$. . . 38

3.1 Exploration of a polyhedron showing finger positions for explored nodes. Note that nodes of G lie in $\mathbb{R}^6$ and cannot be fully visualized in $\mathbb{R}^2$, so pairs of finger positions are shown. 48
3.2 A second view of the polyhedron. 48
3.3 Inter-finger distance of nodes in the closed list, $\mathcal{X}$, vs. exploration step of the algorithm. 49
3.4 Geometry used to characterize point $q_0 = (p_1^0, p_2^0)$. Note that no surfaces of $\mathcal{B}$ are shown. 49
3.5 Construction of a path showing that if $\mathcal{M}_2$ is connected, then D^- is connected. 49
3.6 A catalog of possible immobilizing grasps. Each contact represents a local minimum in the inter-finger distance when the opposing finger is fixed. 53
3.7 A catalog of possible puncture grasps. Each contact represents a saddle in the inter-finger distance when the opposing finger is fixed. 53
3.8 A parameterization of two infinite planes, which demonstrates the convexity of $d(\mathbf{s})$. . 55

4.1 Parameterization of the hand shape for a three point-fingered hand. The pairwise inter-finger distances are given by σ_1, σ_2, and σ_3. 61

4.2 A polygon, $\mathcal{B}$, and two grasps: ρ_a cages $\mathcal{B}$; ρ_b does not. Also shown is a decomposition of free space around a convex polygon, $\mathcal{B}$, dividing it into regions, $\mathcal{R}_i$. 62

4.3 A physical caging region for a polygon, shown in yellow. The red, green, and blue regions shown in Fig. 4.3 are possible placements of f_1, f_2, and f_3, respectively, which satisfy both $\rho \in \mathcal{T}_{2,3,5}$ and $\sigma \in \mathcal{H}$ (see Fig. 4.4). Starting from one of these positions, the restriction $\sigma \in \mathcal{H}$ guarantees that the hand will not leave $\mathcal{T}_{2,3,5}$. 63

4.4 A shape space caging region, $\mathcal{H}$ for the polygon shown in Fig. 4.3. Restriction of the hand shape to this region of $\mathcal{S}$ will prevent the hand from leaving the initial triad, $\mathcal{T}_{2,3,5}$. 63

4.5 A physical caging region for polygon $\mathcal{B}$, considering the union of two triads $\mathcal{U} = \mathcal{T}_{2,3,5} \cup \mathcal{T}_{2,3,6}$. If the fingers are initially placed in the red, green, and blue regions, restricting the shape parameter to lie the region $\mathcal{H} \in \mathcal{S}$ (shown in Fig. 4.6) will prevent the fingers from leaving $\mathcal{U}$. 68

4.6 A shape space caging regions, $\mathcal{H}$, for a set of triads, $\mathcal{U}$, of the polygon shown in Fig. 4.5. Restriction of the shape parameters to this region, $\mathcal{H}$ will prevent fingers starting in $\mathcal{U}$ from leaving $\mathcal{U}$. 68

4.7 Puncture manifolds which divide $\mathcal{S}$ into caging and escape regions. These manifolds are generated by grasps shown in Fig. 4.8. 70

4.8 Grasps in physical shape which result in puncture manifolds. Corresponding points in $\mathcal{S}$ are shown as colored circles in Fig. 4.7. 70

4.9 A grasp in which one finger lies at a vertex of $\mathcal{B}$ while another lies at the perpendicular projection of that vertex onto an edge of $\mathcal{B}$, which allows the hand to transition from one triad to another. 72

4.10 A grasp in which one finger lies at a vertex of $\mathcal{B}$ while two other fingers lie on edges of $\mathcal{B}$, which allows the hand to transition from one triad to another. 72

4.11 An immobilizing grasp, whose hand shape lies on an immobilizing manifold. 73

4.12 Two grasps whose hand shape of these grasps does *not* lie on a boundary of the caging region. 73

4.13 A parameterization of grasps with two finger contacts. This provides tests of whether a point is caging or escape, and feasible or non-feasible. 76

4.14 Physical caging region for an initial caging region composed of eight triads. The caging region associated with these triads is quite complicated, and is shown in Fig. 4.15. . . 76

4.15 Shape space caging region, $\mathcal{H}$ for the polygon shown in Fig. 4.14. Colored portions are puncture manifolds; black portions are immobilizing manifolds. The region is quite complex. 77

List of Algorithms

2.1 Contact Space Caging Algorithm 30
2.2 Contact Space Tunnel Curve Construction 31

Notation

Used Throughout	
$\mathcal{B}$	Object to be caged
f_i	Finger i
p	Position of a finger in space
σ	Finger opening parameter
q	Configuration of $\mathcal{B}$
$\mathcal{F}$	Free space
$d(\cdot)$	Inter-finger distance function

Two Finger Caging	
$\mathcal{U}$	Contact space
$\mathcal{C}$	Configuration space
$\mathcal{CB}$	Configuration space obstacle generated by $\mathcal{B}$
$\mathcal{S}$	Double contact manifold
$\pi(\cdot)$	Projection function
$\mathcal{U}_{\leqslant c}$	A c-sublevel set of $\mathcal{U}$
Δ	The contact space diagonal
$\mathcal{R}_{i,j}$	Contact space rectangle
$\mathcal{P}_{i,j}$	Contact space polychoron (4-D polytope)
G	The caging graph
G_T	The augmented caging graph
v	A node of the caging graph
$\mathcal{T}$	Tunnel curves
$\mathcal{O}$	The open list
$\mathcal{X}$	The closed list
$\mathcal{E}$	Boundary planes
$\mathcal{M}$	The medial region
$\mathcal{L}$	The lateral region

Three Finger Caging	
$\mathcal{S}$	Shape space
$\mathcal{R}_i$	Region i
ρ	Configuration and hand shape
$\mathcal{T}_{i,j,k}$	Triad associated with $\mathcal{R}_i$, $\mathcal{R}_j$, and $\mathcal{R}_k$
$\mathcal{U}$	Union of triads
$\mathcal{H}$	Shape space caging region
$\mathcal{I}$	Initial caging grasp region
$\mathcal{E}_t$	Single transition escape region
$\mathcal{C}_t$	Single transition caging region

Chapter 1

Introduction

1.1 Motivation

Robots are increasingly being used to complete complicated tasks in unstructured environments such as home environments, disaster areas, and war zones. Until recently, most robots worked in highly structured environments, such as assembly lines, and interacted with the world using end-effectors which are designed for a specific task—for example a robot on an auto assembly line might have a welder or spray painting device permanently attached to it. Now robots are being asked to interact with a large variety of objects, and require general purpose grippers to do so.

Robotic grasping and manipulation in unknown environments is made difficult by several factors. The objects to be grasped may not be previously known to the robot, and models of these objects generated using onboard visual sensing often have significant error. Industrial robots are typically very precise, but to ensure this precision they are also typically expensive, heavy, and dangerous for humans to work near. To make robots cheaper, lighter, and safer, precision in the accuracy of end-effector position is often sacrificed. This introduces uncertainty in the position of the gripper relative to the object. Robotic grasping theory must account for such uncertainty in order to provide robust and reliable grasping in unstructured environments.

One method for dealing with such uncertainty is robotic caging. Unlike an immobilizing grasp, in which the robotic hand grasps an object so as to prevent all motion of the object relative to the hand, a caging grasp only prevents the object from escaping through the fingers, while allowing some motion of the object relative to the hand. Caging grasps can be sufficient for completion of many tasks. For example, if a robot grasps a door handle so as to immobilize the handle relative to the hand, then the robot must follow the exact trajectory of that handle as it opens the door. If instead the robot cages the handle, then the robot can simply follow the approximate path of the handle, while the nature of a cage guarantees that the door will indeed be opened. Caging may also be used as a waypoint on a path to an immobilizing grasp. Instead of moving straight to an immobilizing grasp, a process in which uncertainty in the objects shape or position may cause the grasp to fail,

the robot instead moves first to a caging configuration, which guarantees that the object cannot escape from the hand. The robot then closes its fingers into an immobilizing grasp.

This thesis presents several algorithms for calculating caging regions of various objects. Both 2D polygons and 3D polytopes are considered, using two or three fingers. While developing these algorithms, several new insights into the geometry of caging on contact space are developed.

1.2 Review of Existing Literature

This section starts with a limited review of the theory of immobilizing grasps, followed by a more thorough review of the literature of caging. Each chapter also includes discussion of the works most related to that chapter.

1.2.1 Grasping

Grasping theory dates back at least to Reuleaux [31], who analyzed part fixturing. Some of the earliest work in robotic grasping comes from Asada [5], whose PhD thesis studied prehensile grasps, and Mason and Salisbury [21], who worked to build a three fingered hand.

An important aspect of grasping is how to model the contact between a finger and the object being grasped. Salisbury [39] considered different friction models, including point contact without friction, point contact with friction, and a soft contact model in which frictional force in the finger can support torque around the contact normal. Cutkosky [12] considered rolling contact and different finger shapes, such as pointed or round fingers.

The concept of an equilibrium grasp was first defined by Salisbury [39]. An equilibrium grasp is one in which all finger contacts obey their respective contact constraints and the object is in static equilibrium (the finger forces exactly balance a particular external wrench applied to the object). Two stricter conditions on having a secure grasp are *form closure* and *force closure*.

Form closure is the condition that any finite or infinitesimal movement of the object will cause it to penetrate a finger, meaning that the object is immobilized in the hand in a purely geometric sense, which is a stronger condition than force closure [20]. An algorithm to find all frictionless point finger placements that produce a form closure grasp of a polygon, and all placements which achieve second-order immobility was given by van der Stappen et al [44].

Force closure is the condition that all external wrenches can be balanced by finger forces which obey contact constraints. Nguyen [23] and Ponce et al [30] gave algorithms for constructing force closure grasps based on the shape of the object. Rimon and Burdick [34] considered the relationship between form and force closure, and defined first and second order form closure, as well as second order force closure. Additionally, Bicchi [6] investigated form and force closure as well as discussed partial form and force closure.

For detailed reviews of grasping based on form and force closure concepts, see Bicchi and Kumar [7] and Okamura et al [24].

1.2.2 Caging

Robotic caging refers to hand configurations in which the object is unable to move arbitrarily far from the hand without penetrating one of the fingers. Most cages do not immobilize the object relative to the hand. Caging was first defined by Kuperberg [19] who posed the following geometry problem:

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

If the polygon is the object to be caged and the points are the finger bodies, then this describes a robotic hand caging a polygonal object. Extension of the problem statement to three dimensions objects, non-polygonal objects, and non-point fingers are straightforward. (Extending solutions of the problem to such cases is not straightforward.)

Caging was introduced to the field of robotics by Rimon and Blake in 1996 [32, 33]. They considered a smooth object caged by two point or disc fingers. They used stratified Morse theory (SMT) [18] to show that changes in the topology of free configuration space only occur at particular equilibrium configurations, and referred to these configurations as *puncture grasps*.

Perhaps the majority of work in caging has related to two finger cages. Vahedi and van der Stappen [43] showed that all two finger cages using two point or disc fingers are either squeezing or stretching cages. A squeezing cage is a set of configurations in which the object will not escape as long as the Euclidean distance between the fingers remains below a certain threshold; in a stretching cage the inter-finger distance must remain above a certain threshold. Rodriguez and Mason [36] formalized the work of Vahedi and extended it to all compact, connected, contractible objects.

Several algorithms exist for finding all caging grasps of a polygon using two point or disc fingers. Both Pipattanasomporn and Sudsang [28] and Vahedi and van der Stappen [43] presented algorithms based on a convex decomposition of free space around the object. These algorithms allow for computation of all caging regions in $\mathcal{O}(n^2 \log n)$ and querying of a particular configuration in $\mathcal{O}(\log n)$ time. While these algorithms are exact, complete, and computationally efficient, they are algorithmically complex.

A simple algorithm was developed by Pipattanasomporn and Sudsang [25] in which all finger positions are projected onto the surface of the polygon. From these possible motions along the surface a *crawling graph* is generated, which can be searched to find caging configurations. The

work reported in Chapter 2 [1] complements this earlier work with several significant contributions. First, it fully analyzes the underlying low-dimensional *contact space* and characterizes the topological properties which define caging formations in the hand's configuration space. Second, it analyzes the inter-finger distance function in configuration space. These properties allow the construction of a *caging graph* which has significantly simpler structure than Pipattanasomporn's *crawling graph*, and has a simple mapping to geometric features on the object boundary.

I extend the contact space analysis to 3D [3], and provide a purely geometric analysis of the critical points where caging topology changes, which is valid in both two and three dimensions.

Most of the rest of the literature of caging deals with robotic hands utilizing more than two fingers. Many of these relate the configuration of the hand to a single scalar parameter. Davidson and Blake [14] considered a three-fingered hand whose relative finger positions are determined by a scalar parameter, and found critical value of this parameter for which the topology of the caging sets change.

Several works consider allowing the fingers more than one degree of freedom, but mapping their position to a single parameter. The object then remains caged as long as certain conditions on this parameter are maintained. Both Wang and Kumar [47] and Fink et al [17] created conservative sufficient conditions for caging when given a scalar function on the relative positions of fingers. They considered this problem in the context of cooperating mobile robots, with the robot bodies standing in for fingers.

Two works consider more general mappings of finger positions to a scalar parameter. Pipattanasomporn and Sudsang [29] used a convex function such as a norm on the pair-wise inter-finger distances and referred to this as *dispersion control*. The object will remain caged as long as the finger positions are maintained such that the scalar mapping remains below a certain threshold. Rodriguez [38] extended and formalized these ideas, referring to is as f-caging.

A few works consider caging regions generated by restricting the relative position of the fingers to higher dimensional regions. Erickson et al [16] considered three point or disc fingers caging a convex polygon. Given two initial finger placements on the boundary of the polygon, they presented an algorithm which finds the 2D regions in which the third finger may be placed. Vahedi and van der Stappen [43] extended this algorithm to non-convex polygons and initial finger placements of the fingers which do not lie on the boundary of the polygon.

The relative position of three fingers can be fully described by three parameters. Only a few authors have considered restrictions on the hand shape in the three dimensional parameter space. Sudsang [40] [41] [42] considered caging a polygon with three disc-shaped mobile robots. He found three circular regions near the boundaries of the object such that placement of a robot into each of these regions guaranteed that the object remained caged, and dubbed these regions MICaDs - maximally independent capture discs. While this does provide a three-dimensional restriction on the

relative positions of the robots/fingers, the algorithm is considerably conservative: First, it considers only contact between an individual robot and one particular edge of the object. Second, disc shaped regions are generally only a subset of the actual 3D caging regions. Finally, even given the first two restrictions, the algorithm returns a conservative disc for some cases.

The methodology presented in Chapter 4 [4] considers three point fingers caging a convex polygon in two dimensions. It allows for three parameters (which is the general case for three point fingers) and finds the full 3D caging regions within which the hand shape may vary while preventing the object from escaping.

There are also several methods which numerically calculate cages for more complicated hands. Diankov et al [15] consider an object and hand described by 3D polytopes and assume that a given caging grasp is provided. They present an algorithm which uses rapidly exploring random trees to explore the 6-DOF region around the initial grasp which represents allowable transformation between the hand and object. The algorithm does not test for the existence of a cage, but requires an initial hand configuration which is known to be caging. Wan et al [46] [45] use 2D slices of higher dimensional spaces to calculate cages for three fingered hands. With these methods, it is difficult to guarantee that the sampling methods will accurately represent the underlying geometry of the problem.

More recent work includes Zarubin et al [48], who used heuristics based on geodesic balls to find grasps which mostly enclose the object in either circle or sphere, and Cappelleri et all [9] who used motion primitives to find sufficient conditions for caging during open loop micro-manipulation assembly tasks.

1.3 Caging Basics

This thesis considers a robotic hand consisting of two or three point or disc fingers caging a polygonal or polyhedral object, denoted $\mathcal{B}$. The i fingers will be denoted f_i, and their positions in space denoted p_i.

For a given fixed set of finger positions relative to an object, the object is said to be *caged* if it is impossible to move the object arbitrarily far from the hand without one or more of the fingers penetrating the object.

Whether a given configuration is caging depends only on the relative position of the fingers with respect to the object. Thus, it is equivalent to consider fixing the position of the fingers relative to some fictitious hand base, and then considering motion of the fingers relative to to the body. As is customary in caging literature, we will adopt this approach throughout this work.

Caging considers two different types of motion: motion of the object relative to the fingers and motion of the fingers relative to each other. Typically the motion of the fingers relative to each

other can be controlled to some precision, while a robot has little control of the motion of the object relative to the fingers. This motivates the definition of a *hand shape*, which is the position of the fingers relative to each other.

Most caging literature, including this work, assume that the object geometry is perfectly known. For online grasping based on sensor information, this may not be a good assumption. Discussion of extensions of caging algorithms to uncertainty in object geometry is found in the conclusion.

1.4 Current Uses and Limitations of Caging

Caging has two main uses in object grasping and manipulation. The first is manipulation of constrained object, in which a caging grasp can allow the robot additional degrees of freedom while allowing being sufficient for task completion. For example, consider a robot opening a door. If the robot grasps the door handle firmly, then the motion of opening the door will dictate a full 6-DOF trajectory of the robots gripper. This requires the robot to have at least six degrees of freedom, and requires either compliance in the robot arm, or very precise control. By caging the door handle, the robot gains significant leeway in its gripper trajectory, while still guaranteeing that the door will be opened, see Fig. 1.1.

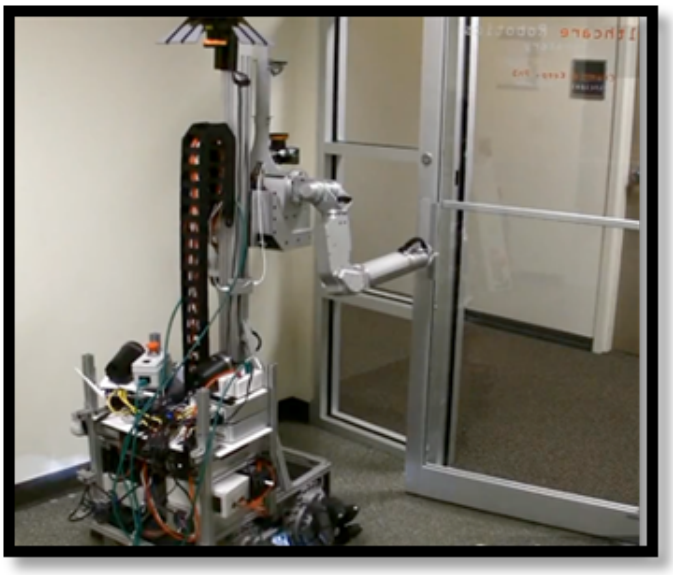


Figure 1.1: A robot attempts to open a door. With an immobilizing grasp, the robots gripper must accurately follow the trajectory of the door handle, which may prove difficult. Caging the door handle gives the robot significantly more freedom in gripper trajectory while still guaranteeing that the door may be opened.

A second use for caging is as a step towards an immobilizing or force closure grasp. Due to uncertainty in the position of the gripper relative to the object being grasped, there is a risk that the gripper might bump the object before a secure grasp is achieved, preventing the grasp from

being completed. If the robot can achieve a caging grasp before contacting the object, and find finger motions that move to a secure grasp (i.e., closing the gripper) while maintaining a cage, then the grasp will be achieved even if there is unexpected contact between the gripper and object. An example of this situation can be seen in Fig. 1.2.

Figure 1.2: A robot arm attempts to grasp a pipe during the DARPA Autonomous Robotic Manipulation program. If the robot bumps the pipe it may fall over and roll away. Achieving a caging grasp prior to closing the gripper can guarantee that the object does not escape during the grasping motion.

While caging is a promising approach for object manipulation and robust grasping, state-of-the-art algorithms, including those presented here, have significant limitations.

Algorithms involving two point or disc fingers require that the object have at least one concavity, as convex objects cannot be caged by two point or disc fingers. Robust three-finger caging, discussed in Chapter 4, begins to address this problem. However, most robotic grippers would require more than three points to approximate reasonably.

The majority of caging algorithms involve two-dimensional objects and/or two point (or disk) fingers. While two-dimensional algorithms can be applied to three dimensional problems (as is done in Chapter 2), the usefulness of this approach is limited to object that can reasonably approximated as two dimensional. Automatic fixturing of quasi-two-dimensional objects (such as those created by laser cutter or water jet machines) might be a useful application of two-dimensional algorithms.

1.5 Thesis Organization and Contributions

The remainder of this thesis is organized as follows. Chapter 2 develops a simple algorithm for finding caging formations of a 2D polygon using two point fingers. Section 2.3 describes a reformulation of caging in terms of *contact space*, the set of hand configurations in which both fingers touch the object.

This formulation provides an intuitive and easy-to-implement algorithm for finding the caging sets of a given polygon. This method considers topology of the configuration space of the hand in terms of level sets of an inter-finger distance function. A simple method for identifying those critical points and searching a graph consisting of those points is presented. This chapter also develops several new insights about the geometry, topology, and analysis of caging on contact configuration space. A real world example demonstrates and validates the algorithm. The analysis of this section is carried out in terms of stratified Morse theory, details of which are provided in Appendix A.

Chapter 3 extends the methods and analysis of Chapter 2 to caging of 3D polytopes with two point fingers. A similar inter-finger distance function and graph are presented, along with the necessary analysis to extend the ideas to three dimensions. Additionally, it develops a straightforward geometric method for classifying the important points of the hand's configuration space which does not rely on stratified Morse theory.

Chapter 4 develops an algorithm for caging convex polygons in two dimensions using three point fingers. It allows complete freedom for the fingers to move, resulting in a three parameter description of the hand shape. It provides the exact bounds on how far the fingers may move relative to each other while still guaranteeing a cage. The methods and analysis of this algorithm are substantially independent of the previously described two finger algorithms.

Chapter 5 summarized the contributions of the thesis and considers future research opportunities.

Chapter 2

Two Fingers in Two Dimensions

2.1 Introduction

This chapter describes an algorithm for finding all two-finger cage formations of planar polygonal objects using a contact space formulation. Recognizing the advantages of contact space, Pipattanasomporn and Sudsang [25] were the first to propose a contact-space scheme which computes two-finger cage formations surrounding a polygonal object. This work complements this early work with several significant contributions. First, it provides a full analysis which explains how the low-dimension contact space captures the essential topological properties that define cage formations in the hand's full configuration space. Second, it provides a detailed analysis of the properties of the inter-finger distance function in contact space. In particular, it considers sublevel sets of the inter-finger distance function in both free configuration space and contact space, and determines when these two level sets are topologically equivalent. These properties are used to construct the *caging graph,* which has a significantly simpler structure and direct interpretation in terms of geometric features along the object's boundary. Third, this chapter describes a fully implementable caging algorithm which is validated on a real-world example.

The caging algorithm described here accepts as input a geometric description of a polygonal object, together with a desired *immobilizing grasp* of the object (see [10, 13, 35] for a discussion and examples of immobilizing grasps). Starting with the immobilizing grasp, the caging algorithm systematically explores a discrete set of vertices and edge segments along the object's boundary, while efficiently reporting a series of *caging sets*[1] surrounding the desired immobilizing grasp—the initial caging set, every intermediate caging set, and the largest caging set beyond which the object can escape to infinity. This novel aspect of the algorithm has practical benefits. It provides grasping systems with a choice of initial finger openings and finger placements that retract to the target grasp, or to one of nearby secure grasps as most suitable for the given application.

This chapter is organized as follows. Section 2.2 introduces basic terminology and defines the

[1]The notion of *caging set* is defined later in this section.

two-finger caging problem. Section 2.3 formulates the caging problem as a contact space search. Section 2.4 associates the critical points of the caging sets with critical points of the inter-finger distance function in contact space. Section 2.5 introduces a rectangular decomposition of contact space and summarizes useful properties of this decomposition. Section 2.6 introduces the *caging graph* in contact space, and considers its topological relationship with the hand's full configuration space. Section 2.7 describes an algorithm that searches the caging graph and reports the initial, intermediate, and maximal cages associated with a desired immobilizing grasp. Section 2.9 describes how these sets can be graphically depicted as two capture regions surrounding the physical object. Section 2.10 applies the caging algorithm to a real-world example, while Section 2.11 considers several extensions of the basic caging algorithm. This chapter concludes with a short summary and discussion of various extensions to this algorithm. The work in this thesis is primarily based on Allen et al [1–4].

2.2 Preliminaries and Problem Definition

The notion of *configuration space*, usually abbreviated as *c-space*, is critical to caging analysis. Throughout this work we consider a moving rigid body $\mathcal{B}$, surrounded by stationary rigid bodies (fingers), denoted $\mathcal{O}_1, ..., \mathcal{O}_k$, which serve as obstacles that prevent $\mathcal{B}$ from moving to certain poses (combinations of position and orientation). Configuration space allows us to determine whether and when $\mathcal{B}$ collides with an obstacle, allowing the detailed interaction between (possibly complex) rigid bodies to be precomputed. Determining if $\mathcal{B}$ is in collision with one or more of the obstacles reduces to determining if the configuration of $\mathcal{B}$, which is a point, lies within a configuration space obstacle.

First, the configuration of $\mathcal{B}$ is specified by a vector $v \in \mathbb{R}^n$ and a rotation $\mathcal{R} \in SO(n)$, where n is 2 or 3, depending on the whether the problem is two or three dimensional. The pair (v, R) thus make up configuration space of $\mathcal{B}$, denoted $\mathcal{C}$. Any position of $\mathcal{B}$ is thus a point in $\mathcal{C}$, and any continuous motion of $\mathcal{B}$ is a curve in $\mathcal{C}$. Second, two rigid bodies cannot occupy the same physical space. Thus, the obstacles $\mathcal{O}_1, ..., \mathcal{O}_k$ introduce forbidden regions in the configuration space of $\mathcal{B}$, referred to as *c-space obstacles*. Specifically, a c-space obstacle corresponding to obstacle $\mathcal{O}_i$, denoted $\mathcal{CO}_i$, is the set of configurations of $\mathcal{B}$ where $\mathcal{B}$ and $\mathcal{O}_i$ intersect.

Once the c-space obstacles are pre-computed, then checking whether $\mathcal{B}$ is in collision with an obstacles becomes the test of whether that point in configuration space lies inside of a c-space obstacle. The area of c-space not occupied by c-space obstacles is referred to as free c-space (or just free space), denoted $\mathcal{F}$.

For caging applications, it is common to consider the fingers movable, and the object $\mathcal{B}$ fixed. In this case $\mathcal{B}$ generates a c-space obstacle for each finger, denoted $\mathcal{CB}$.

This chapter considers the caging of a polygonal object, $\mathcal{B}$, by two point fingers. Let the two

fingers initially hold the object at an immobilizing grasp along its outer boundary. As the fingers move apart, the object remains caged for a finite range of finger openings. Eventually the inter-finger distance reaches a critical value beyond which the object can escape the cage formed by the two fingers. This critical finger opening may only allow an intermediate escape into a larger cage associated with other object features, or an ultimate escape to infinity. Rimon and Blake [33] have shown that each of these critical events corresponds to a *frictionless equilibrium grasp* of the object $\mathcal{B}$. These critical events provide full information on the cage formations surrounding the object $\mathcal{B}$.

This chapter describes an algorithm that computes the caging sets using *contact space.* Let $x_1, \ldots, x_n$ denote the object's vertices, arranged in counterclockwise order along its outer boundary. The i^{th} finger contact along the object's outer boundary is parametrized in counterclockwise order by $s_i \in [0, L]$ for $i = 1, 2$. The parametrization starts with $s_i = 0$ at x_1 and ends with $s_i = L$ again at x_1. The value of L is usually the length of the object's perimeter. The finger positions along the object's boundary in a reference frame fixed to $\mathcal{B}$ are denoted $p(s_i)$ for $i = 1, 2$. The two-finger *contact space* is defined as follows.

Definition 2.2.1 *Let a polygonal object, $\mathcal{B}$, be contacted by two point fingers at $p(s_1)$ and $p(s_2)$.* **Contact space** *is the set $\mathcal{U} = [0, L] \times [0, L]$ in the (s_1, s_2) plane parameterizing all two-finger contacts along the object's boundary.*

Note that the (s_1, s_2) plane is periodic modulo L, and hence contact space $\mathcal{U}$ represents a topological 2-torus. The notion of contact space can be extended to disc fingers, with some complication incurred by the rotation of such fingers about vertices of $\mathcal{B}$. This topic is discussed in Section 2.11. Another consideration concerns finger placements within a hole of $\mathcal{B}$. While the object cannot escape to infinity in this case, one may still specify an immobilizing grasp and ask for the local caging sets surrounding this grasp. Section 2.11 will discuss an extension of the caging algorithm to objects with holes.

Two fingers that move freely in $\mathbb{R}^2$ have a total of four degrees of freedom. One can interpret the two fingers as the following two-fingered hand. Designate the line segment connecting the fingers as the hand's virtual base, which can freely move and rotate in $\mathbb{R}^2$; then define the inter-finger distance as the fingers' *opening parameter*, σ. The resulting hand configuration space is defined as follows.

Definition 2.2.2 *The* **configuration space** *of a two-fingered hand, denoted $\mathcal{C}$, is the four-dimensional space $(q, \sigma) \in \mathbb{R}^3 \times \mathbb{R}$, where $q \in \mathbb{R}^3$ is the hand's base configuration, and $\sigma \in \mathbb{R}$ is the fingers' opening parameter.*

The fingers' opening parameter, σ, can be thought of as a shape parameter of the two-fingered hand [36]. From the hand's point of view, the object $\mathcal{B}$ forms an obstacle. The *c-space obstacle* corresponding to $\mathcal{B}$, denoted $\mathcal{CB}$, is the set of hand configurations at which one or both fingers

intersect the stationary object $\mathcal{B}$.[2] The hand's *free c-space,* $\mathcal{F}$, is the complement of $\mathcal{CB}$'s interior:

$$\mathcal{F} = \mathcal{C} - \mathsf{int}(\mathcal{CB}),$$

where int denotes set interior. The boundary of $\mathcal{F}$ (which is the same as the boundary of $\mathcal{CB}$) consists of all hand configurations at which one or both fingers touch the object's boundary. The notion of a *caging set* is formulated in $\mathcal{F}$ as follows.

Definition 2.2.3 *The* **caging set** *of $\mathcal{B}$ is the set of all two-finger placements in $\mathcal{F}$ such that $\mathcal{B}$ cannot be moved arbitrarily far from the fingers.*

In general, two-finger cages can be of two possible types: *squeezing* cages where the fingers close inward in order to grasp the object, and *stretching* cages where the fingers open outward in order to grasp the object [38, 43]. While this work focuses on *squeezing cages,* contact space can also be used to compute stretching cages as discussed in Section 2.11.

Definition 2.2.4 *A* **squeezing caging set** *of $\mathcal{B}$ is any subset of caging points in $\mathcal{F}$ from which the object $\mathcal{B}$ is immobilized when the fingers are monotonically squeezed to reduce the fingers' opening parameter σ.*

In this work the term *caging set* will automatically imply a squeezing caging set. The caging algorithm will identify the following caging sets.

Problem Definition: Given an initial two-finger immobilizing grasp of $\mathcal{B}$, compute the following three types of caging sets:

1. The *initial caging set*—the largest caging set from which the fingers are guaranteed to return to the initial immobilizing grasp while keeping the object caged during the finger squeezing process.

2. The *intermediate caging set*—any caging set which contains the initial caging set, such that all fingers end at a finite number of possible immobilizing grasps while keeping the object caged during the finger squeezing process.

3. The *maximal caging set*—the largest caging set which contains the initial caging set, such that the fingers can be squeezed into a finite number of possible immobilizing grasps while keeping the object caged during the finger squeezing process.

Example: Consider the polygonal object $\mathcal{B}$ depicted in Figure 2.1, which is initially immobilized by two point fingers. The initial, intermediate, and maximal caging sets associated with this grasp

[2] The c-space obstacle $\mathcal{CB}$ corresponds to $\mathcal{B}$'s initial location in $\mathbb{R}^2$. Any disturbance of $\mathcal{B}$ by the closing fingers will not break the cage or affect the eventual arrival of the 2-finger hand to the desired immobilizing grasp.

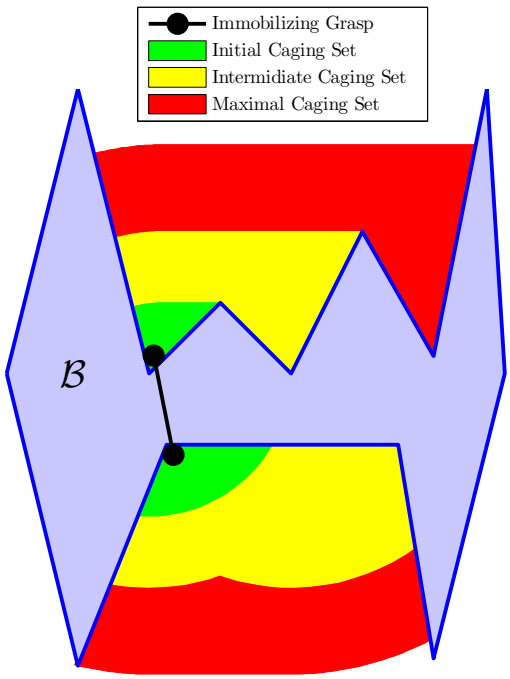


Figure 2.1: The immobilizing grasp along with the initial, intermediate, and maximal caging sets of a polygonal object. Note that each of these sets is also subject to a constraint on the maximum inter-finger distance, which cannot be represented in two dimensions.

are depicted in Figure 2.1. Each shaded region represents allowable finger placements that maintain the respective caging set, provided that the inter-finger distance is kept below the critical value associated with this caging set. Note that smaller caging sets are subsets of larger caging sets. ○

The caging sets will be computed in contact space using the following inter-finger distance function.

Definition 2.2.5 *The* **inter-finger distance** *function is the real valued function* $d : \mathcal{U} \to \mathbb{R}$ *given by*

$$d(s_1, s_2) = |p(s_1) - p(s_2)|, \tag{2.1}$$

where $p(s_1)$ *and* $p(s_2)$ *are the finger positions along the object's boundary*

The function $d(s_1, s_2)$ measures the inter-finger distance across all two-finger contacts along the boundary of $\mathcal{B}$. To properly relate the function d with the fingers' opening parameter, σ, consider the following configuration space submanifold.

Definition 2.2.6 *The* **double-contact submanifold** *in* $\mathcal{C}$*, denoted* $\mathcal{S}$*, consists of all hand configurations at which both fingers touch the object's boundary.*

Note that $\mathcal{S}$ forms a two-dimensional manifold in $\mathbb{R}^4$, which is globally parameterized by contact space $\mathcal{U}$. Also note that $\mathcal{S}$ lies on the boundary of the free c-space $\mathcal{F}$. Let $\pi : \mathcal{F} \to \mathbb{R}$ denote the projection function $\pi(q, \sigma) = \sigma$. Then $d(s_1, s_2)$ is the restriction of π to $\mathcal{S}$, such that $\mathcal{S}$ is parametrized by $(s_1, s_2) \in \mathcal{U}$.

2.3 Contact Space Formulation of Caging

Instead of searching the hand's full configuration space for the caging sets, this chapter proposes to search contact space, $\mathcal{U}$, thereby reducing the dimensionality of the problem from $\mathbb{R}^4$ to $\mathbb{R}^2$. To show that a search in $\mathcal{U}$ can find the caging sets, we formulate the caging problem as the existence of *test paths* which lie entirely in sublevel sets $d(s_1, s_2) \leqslant c$ in $\mathcal{U}$. Sublevel sets of both $\mathcal{F}$ and $\mathcal{U}$ are defined below.

Definition 2.3.1 *The* **c-sublevel set** *of $d(s_1, s_2)$ in contact space is the set $\mathcal{U}_{\leqslant c} = \{(s_1, s_2) \in \mathbb{R}^2 : d(s_1, s_2) \leqslant c\}$. The* c-sublevel set *of π in the free c-space, $\mathcal{F}$, is the set $\mathcal{F}_{\leqslant c} = \{(q, \sigma) \in \mathcal{F} : \pi(q, \sigma) \leqslant c\} = \{(q, \sigma) \in \mathcal{F} : \sigma \leqslant c\}$.*

The idea of an *escape* plays a central role in caging theory. The standard definition of an escape is the ability to move the two-fingered hand with a fixed finger opening arbitrarily far from the object, implying that the object can escape from the given grasp when the fingers are held fixed. The corresponding notion of an *escape* in contact space is defined as follows.

Definition 2.3.2 *An* **escape point** *in $\mathcal{U}$ is any point on the diagonal $\Delta = \{(s_1, s_2) \in \mathcal{U} : s_1 = s_2\}$, where the two fingers touch the same point on the object's boundary, $p(s_1) = p(s_2)$.*

The notion of escape points in $\mathcal{U}$ is equivalent to the standard definition of an escape as follows. When an escape point is reached in $\mathcal{U}$, the two fingers become coincident and can move arbitrarily far from the object.[3] Conversely, suppose the hand can move arbitrarily far from $\mathcal{B}$ according to the standard definition of an escape. The two fingers can be pinched together when located away from $\mathcal{B}$, then moved back as a single finger to the object's boundary (which is an escape point in $\mathcal{U}$). The standard notion of escape is thus equivalent to the notion of escape points in $\mathcal{U}$. Similarly to Definition 2.3.2, the *escape points* in $\mathcal{F}$ are defined as points $(q, \sigma) \in \mathcal{S}$ which correspond to the diagonal Δ in $\mathcal{U}$. At these hand configurations both fingers touch the same boundary point of the object $\mathcal{B}$.

Consider the meaning of a *cage formation* that can be closed into an immobilizing grasp in terms of the level sets of π in the hand's free configuration space.

Definition 2.3.3 *A* **c-sublevel path** *in the free c-space, $\mathcal{F}$, is a path along which the maximum value of the fingers' opening parameter does not exceed c.*

Let the object $\mathcal{B}$ be initially immobilized at a hand configuration (q_0, σ_0). For a certain interval of finger openings, $[\sigma_0, \sigma_1)$, the object remains caged by the two fingers. For each σ in this interval, the connected component of the sublevel set $\mathcal{F}_{\leqslant \sigma}$ containing the point (q_0, σ_0) does not contain any

[3] Recall that the fingers contact the object's outer boundary.

escape points. The cage is broken at a critical hand configuration, (q_1, σ_1), which corresponds to a frictionless equilibrium grasp of $\mathcal{B}$ [32, 33]. A definition of equilibrium grasps follows.

Definition 2.3.4 *A rigid object $\mathcal{B}$ is held at a frictionless two-finger* **equilibrium grasp** *when the net force and torque applied on $\mathcal{B}$ is zero:*

$$\begin{pmatrix} f_1 \\ p(s_1) \times f_1 \end{pmatrix} + \begin{pmatrix} f_2 \\ p(s_2) \times f_2 \end{pmatrix} = \vec{0}, \tag{2.2}$$

where the finger forces f_1 and f_2 are applied along the contact normals at $p(s_1)$ and $p(s_2)$; $p \times f = p^T J f$ for $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

A critical equilibrium grasp where the cage is broken, (q_1, σ_1), will be termed a *puncture point* or *puncture grasp*. Starting at the immobilizing grasp, (q_0, σ_0), the value of σ at the *first* puncture point can be viewed as the minimum value, σ_1, such that there exists a σ_1-sublevel path which starts at (q_0, σ_0) and either ends at an escape point or reaches a neighboring immobilizing grasp. The value of σ at the *last* puncture point, denoted σ_{esc}, can be viewed as the minimum value of σ such that there exists a σ_{esc}-sublevel path between (q_0, σ_0) and an escape point.

To correctly identify the caging sets in contact space, a search for escape paths in $\mathcal{U}$ must yield the same answer as a search for escape paths in $\mathcal{F}$. Similarly to Definition 2.3.3, a *c-sublevel path* in $\mathcal{U}$ is a path along which the maximum value of the inter-finger distance does not exceed c. Consider the following notion of *sublevel equivalence* between $\mathcal{U}$ and $\mathcal{F}$.

Definition 2.3.5 *Contact space, $\mathcal{U}$, is* **sublevel equivalent** *with the free c-space, $\mathcal{F}$, if for any two points $(q, \sigma), (q', \sigma') \in \mathcal{S}$ there exists a c-sublevel path between (q, σ) and (q', σ') in $\mathcal{F}$ if and only if there exists a c-sublevel path between their equivalent points in $\mathcal{U}$.*

To correctly identify the caging sets in contact space, one must ensure that $\mathcal{U}$ is *sublevel equivalent* with $\mathcal{F}$. Many objects, including the one depicted in Figure 2.1, possess this property. However, complex objects having handle-like features do not possess this property. Section 2.6 will deal with such objects, restoring sublevel equivalence of $\mathcal{U}$ and $\mathcal{F}$.

2.4 Contact Space Representation of the Caging Sets

This section establishes that immobilizing and puncture grasps are critical points of $d(s_1, s_2)$ in contact space $\mathcal{U}$. Then it characterizes the remaining critical points of $d(s_1, s_2)$ and illustrates these points with examples.

2.4.1 Two-finger Equilibrium Grasps in Contact Space

We start with a characterization of the critical points of $d(s_1, s_2)$ in contact space $\mathcal{U}$.

Lemma 2.4.1 *The* **critical points** *of the inter-finger distance function $d(s_1, s_2)$ in $\mathcal{U}$ are:*

1. *Edge-edge critical points, where the fingers touch opposing points on two parallel edges of $\mathcal{B}$.*
2. *Vertex-edge critical points, where one finger touches a vertex while the other finger touches an opposing edge of $\mathcal{B}$.*
3. *Vertex-vertex critical points, where the fingers touch opposing vertices of $\mathcal{B}$.*

Proof By the chain rule, the gradient of $d(s_1, s_2) = ||p(s_1) - p(s_2)||$ is given by

$$\nabla d(s_1, s_2) = \frac{1}{||p(s_1) - p(s_2)||} \begin{pmatrix} (p(s_1) - p(s_2)) \cdot p'(s_1) \\ -(p(s_1) - p(s_2)) \cdot p'(s_2) \end{pmatrix}$$

At the critical points $\nabla d(s_1, s_2) = \vec{0}$. In order to evaluate the gradient, there are three cases to consider. In case (i) both fingers contact edge interior points. In this case $p'(s_1)$ and $p'(s_2)$ are the edge tangents at the contacts, and the condition $\nabla d(s_1, s_2) = \vec{0}$ implies that the two contacts must be located on parallel edges of $\mathcal{B}$, such that the line passing through the contacts is collinear with the edge normals. In case (ii) one finger touches a vertex of $\mathcal{B}$, and we need the notion of the *generalized contact normal* [11]. At an edge interior point it is the edge's unit normal at this point, while at a vertex it is the convex combination of the unit normals to the edges meeting at the vertex.[4] Suppose one finger, $p(s_1)$, touches a vertex while the other finger, $p(s_2)$, touches an edge interior point of $\mathcal{B}$. The tangent $p'(s_1)$ at the vertex can be any vector orthogonal to the vectors of the generalized contact normal at the vertex. Using t_{11} and t_{12} to denote the tangents to the edges meeting at the vertex, the critical points satisfy the non-smooth condition:

$$\frac{1}{\|p(s_1) - p(s_2)\|} \begin{pmatrix} (p(s_1) - p(s_2)) \cdot (\lambda_1 t_{11} + \lambda_2 t_{12}) \\ -(p(s_1) - p(s_2)) \cdot p'(s_2) \end{pmatrix} = \vec{0},$$

for some $\lambda_1, \lambda_2 \geqslant 0$. It follows from the gradient expression that a vertex-edge critical point occurs when the generalized contact normal at $p(s_1)$ contains a vector collinear with the edge normal at $p(s_2)$. This occurs at the perpendicular projection of the vertex onto the opposing edge. Case (iii) can be similarly treated using the generalized contact normal at the two vertices contacted by the fingers. □

The next lemma asserts that the frictionless two-finger equilibrium grasps of $\mathcal{B}$ are critical points of $d(s_1, s_2)$.

[4] When a point finger touches a convex vertex of $\mathcal{B}$, one should model the finger as a small disc touching the vertex. Any vector from the generalized contact normal at the vertex can be realized as a physical finger force direction.

Lemma 2.4.2 *The frictionless two-finger equilibrium grasps of the object $\mathcal{B}$ are* **critical points** *of $d(s_1, s_2)$ in $\mathcal{U}$.*

Proof Assume the vectors of the generalized contact normal point into $\mathcal{B}$'s interior. Under this interpretation, any vector from the generalized contact normal can be realized as a physical finger force direction. The two-finger equilibrium grasps of $\mathcal{B}$ are finger placements having opposing generalized contact normals according to (2.2). This condition matches the critical point condition, $\nabla d(s_1, s_2) = \vec{0}$, in all three cases listed in Lemma 2.4.1. □

The critical points of $d(s_1, s_2)$ in $\mathcal{U}$ can be local minima, saddles, or local maxima. The following proposition asserts that the immobilizing grasps of $\mathcal{B}$ are local minima of $d(s_1, s_2)$.

Proposition 2.4.3 *The two-finger immobilizing grasps of $\mathcal{B}$ are* **local minima** *of $d(s_1, s_2)$ in $\mathcal{U}$.*

Proof The c-space obstacle induced by $\mathcal{B}$ is the union $\mathcal{CB} = \mathcal{CB}_1 \cup \mathcal{CB}_2$, where $\mathcal{CB}_i$ is the set of hand configurations at which the i^{th} finger overlaps the object $\mathcal{B}$ for $i = 1, 2$. Let $(q_0, \sigma_0) \in \mathcal{F}$ represent an immobilizing grasp of $\mathcal{B}$, corresponding to a point (s_1^0, s_2^0) in $\mathcal{U}$. Since $\mathcal{B}$ is immobilized by the fingers, the point (q_0, σ_0) is completely surrounded by the finger c-obstacles in the fixed-σ slice of $\mathcal{C}$ passing through σ_0. As the fingers' opening parameter, σ, increases in the interval $[\sigma_0, \sigma_0 + \epsilon]$, the isolated point expands to a 3D cavity in each fixed-σ slice of $\mathcal{C}$. This cavity corresponds to the cage. The boundary of each cavity consists of the finger c-obstacle surfaces, as well as the intersection curve of these surfaces. For sufficiently small ϵ, the intersection curve is a topological closed loop, and it represents one contour of $d(s_1, s_2)$ in $\mathcal{U}$. As these contours surround the point (s_1^0, s_2^0) and their value increases in the interval $\sigma \in [\sigma_0, \sigma_0 + \epsilon]$, the function $d(s_1, s_2)$ must have a local minimum at (s_1^0, s_2^0) in $\mathcal{U}$. □

The next proposition asserts that the puncture grasps of $\mathcal{B}$ are saddle points of $d(s_1, s_2)$ in $\mathcal{U}$.

Proposition 2.4.4 *The two-finger puncture grasps of $\mathcal{B}$ are* **saddle points** *of $d(s_1, s_2)$ in $\mathcal{U}$.*

A proof of the proposition appears in the appendix. Intuitively, let (q_1, σ_1) be a puncture point in $\mathcal{F}$, and let (s_1^1, s_2^1) be the corresponding point in $\mathcal{U}$, with $d_1 = d(s_1^1, s_2^1)$. As σ increases in the interval $[\sigma_1 - \epsilon, \sigma_1 + \epsilon]$, two locally distinct connected components of the sublevel set $\mathcal{F}_{\leqslant \sigma}$ meet at the puncture point and become a single component for $\sigma \geqslant \sigma_1$. This topological change also occurs in the double-contact submanifold $\mathcal{S}$, which contains the puncture point. Since $\mathcal{S}$ is parametrized by $\mathcal{U}$, two locally distinct connected components of the sublevel set $\mathcal{U}_{\leqslant d_1}$ meet at the point (s_1^1, s_2^1) and become a single component for $d \geqslant d_1$, which is exactly the behavior one expects at a saddle point of $d(s_1, s_2)$ in $\mathcal{U}$.

2.4.2 The Inter-finger Distance Function in Contact Space

The immobilizing and puncture point grasps are only a *subset* of the critical points of $d(s_1, s_2)$ in $\mathcal{U}$. The full set of critical points can be divided into feasible and non-feasible equilibrium grasps. At the *feasible* equilibrium grasps, the generalized contact normals at the finger contacts oppose each other, thus ensuring the existence of opposing finger forces at the contacts. At the *non-feasible* equilibrium grasps, the generalized contact normals contain only parallel force directions which cannot support an equilibrium grasp. The feasible equilibrium grasps can be further divided into *squeezing* and *stretching* grasps defined in Section 2.2. The following example illustrates some of these grasps.

Example: Consider the two-finger equilibrium grasps depicted in Figures 2.2, 2.3 and 2.4. All of these grasps correspond to critical points of $d(s_1, s_2)$ in $\mathcal{U}$. Feasible grasps are shown in blue; non-feasible grasps are shown in red. Note that feasible only indicates that the contact normals are in opposing directions and that grasp forces could be exerted on the object. Not all feasible grasps are relevant to caging. The grasps shown in Fig. 2.2(a),(e), Fig. 2.3(a),(e), and Figure 2.4(a) are feasible squeezing equilibrium grasps, while the grasps shown in Fig. 2.2(d),(h), Fig. 2.3(d),(h), and Fig. 2.4(d) are feasible stretching equilibrium grasps. The remaining contact arrangements are also critical points of $d(s_1, s_2)$, but these are non-feasible equilibrium grasps. The grasps that are most relevant to caging are as follows: Squeezing immobilizing grasps are only those shown in Fig. 2.2(a),(e). Squeezing puncture grasps are only those shown in Fig. 2.3(a),(e). The only form of stretching immobilizing grasp is shown in Fig. 2.4(d). Stretching puncture grasps are shown in Fig. 2.3(d),(h). ○

Among the critical points of $d(s_1, s_2)$ in $\mathcal{U}$, the local minima occur either at a pair of opposing concave vertices, or with one finger at a concave vertex and the other on an opposing edge of $\mathcal{B}$ (Figure 2.2). The saddles of $d(s_1, s_2)$ occur either at a pair of opposing vertices such that one vertex is convex while the other vertex is concave, or with one finger at a convex vertex and the other on an opposing edge of $\mathcal{B}$ (Figure 2.3). The local maxima of $d(s_1, s_2)$ are important for stretching cages as discussed in Section 2.11. Focusing on the local minima and saddles of $d(s_1, s_2)$, consider the following example.

Example: Consider the polygonal object $\mathcal{B}$ depicted in Figure 2.5. The numbers at the vertices indicate arc-length measured counterclockwise from the left-most vertex x_1. Two immobilizing grasps, two puncture grasps, and one escape point are shown. The corresponding contours of $d(s_1, s_2)$ in contact space $\mathcal{U}$ are shown in Figure 2.6. The immobilizing grasps (marked with ○) lie at local minima, while the puncture grasps (△) lie at saddle points of $d(s_1, s_2)$. The escape point (▫) lies along the diagonal, Δ, where $d(s_1, s_2) = 0$. ○

When the object $\mathcal{B}$ has parallel edges, opposing finger placements along these edges correspond to degenerate critical points of $d(s_1, s_2)$. Such critical points form line segments rather than isolated

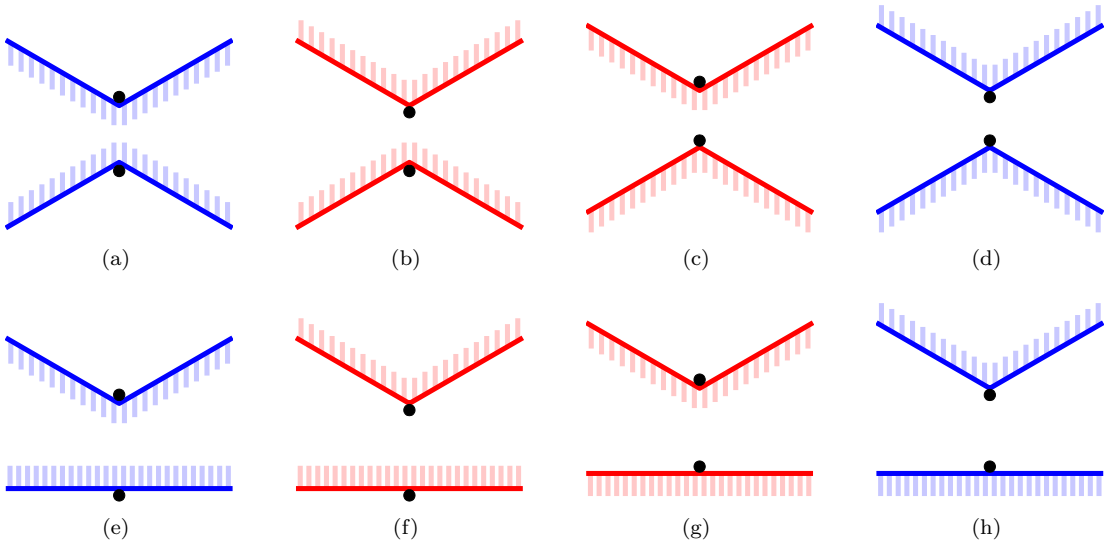


Figure 2.2: Critical points of $d(s_1, s_2)$ which are local minima in $\mathcal{U}$.

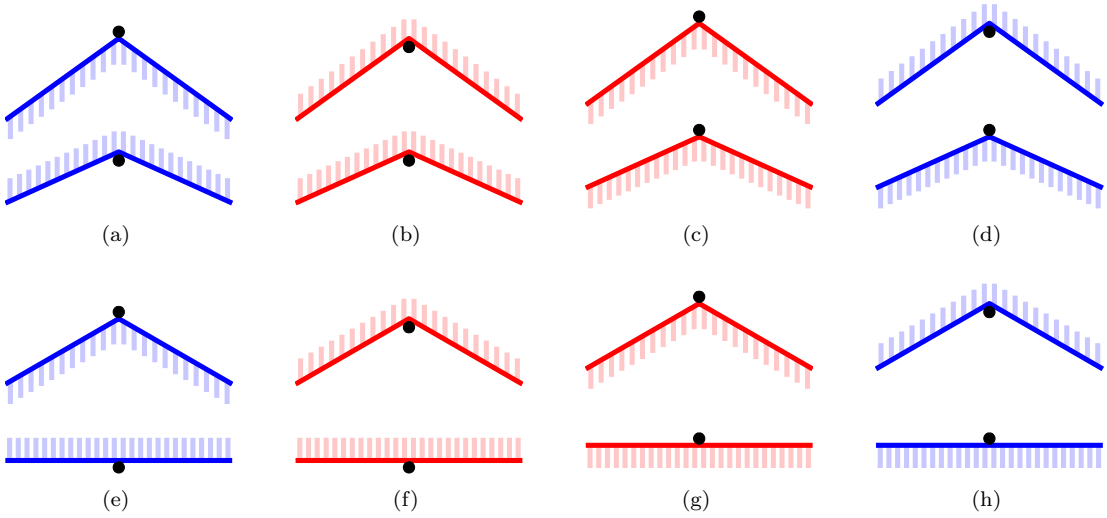


Figure 2.3: Critical points of $d(s_1, s_2)$ which are saddles in $\mathcal{U}$.

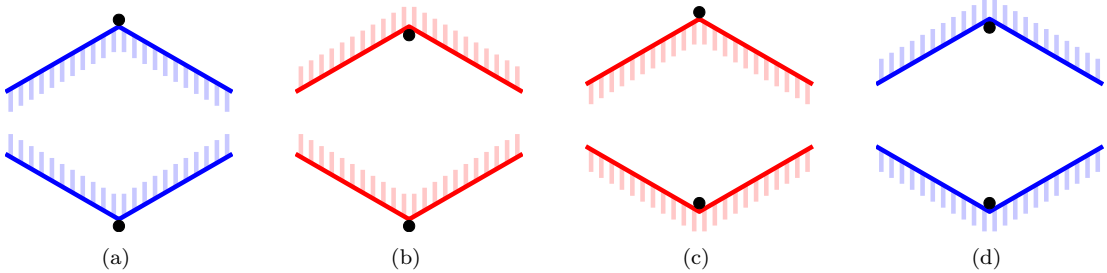


Figure 2.4: Critical points of $d(s_1, s_2)$ which are local maxima in $\mathcal{U}$.

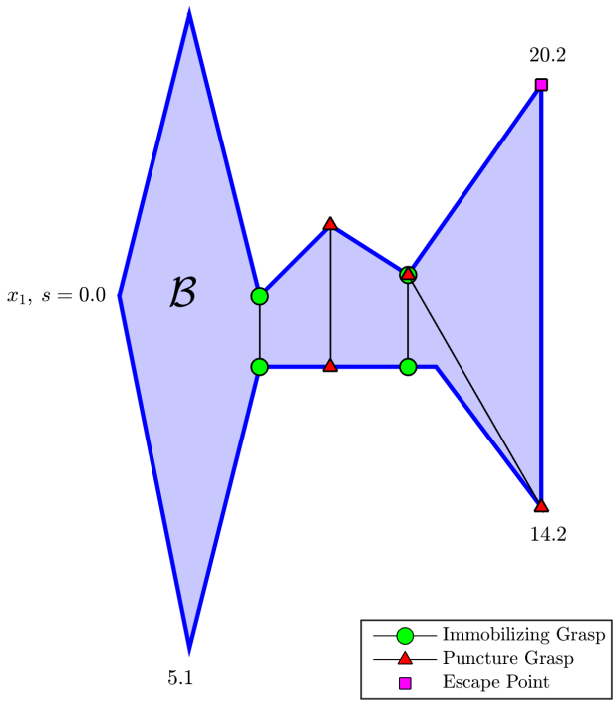


Figure 2.5: A subset of the immobilizing grasps, puncture grasps, and escape points of a polygonal object. Values of the boundary parameter s are shown at selected vertices.

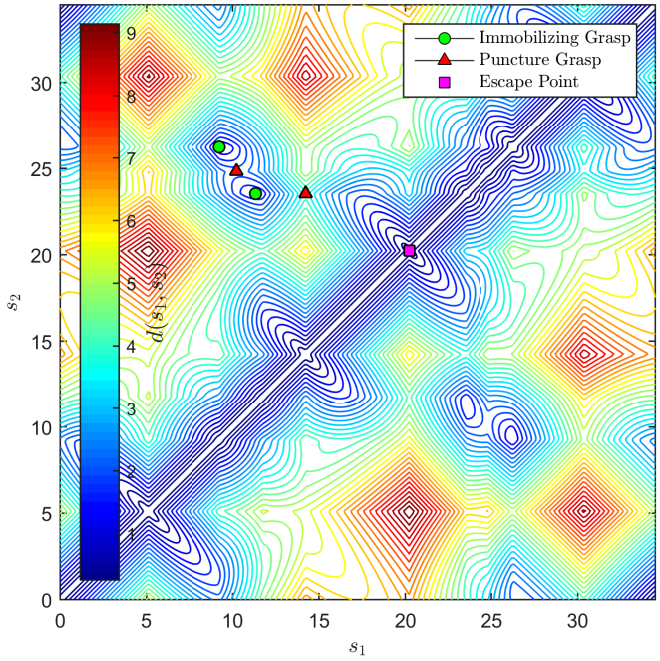


Figure 2.6: Contours of $d(s_1, s_2)$ in contact space $\mathcal{U}$ for the object shown in Figure 2.5. Note that $\mathcal{U}$ represents a topological 2-torus.

points as discussed below.

2.5 Rectangular Decomposition of Contact Space

Contact space possesses a natural decomposition into rectangles which have a special relationship to the critical points of $d(s_1, s_2)$. Let the edges of $\mathcal{B}$ be labeled as $e_1, \ldots, e_n$ in counterclockwise order.

Definition 2.5.1 *The* **contact-space rectangles,** *denoted $\mathcal{R}_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$, correspond to all possible placements of finger $p(s_1)$ along the edge e_i and finger $p(s_2)$ along the edge e_j of $\mathcal{B}$.*

The partition of contact space into rectangles is illustrated in Figure 2.7. Each rectangle $\mathcal{R}_{ij}$ includes its bounding lines. These lines correspond to a placement of one finger at a vertex of $\mathcal{B}$, while the other finger moves freely along one edge of $\mathcal{B}$. The corners of $\mathcal{R}_{ij}$ correspond to finger placements at two vertices of $\mathcal{B}$. The following properties of the contact-space rectangles will prove useful in later sections.

Lemma 2.5.2 *The function $d(s_1, s_2)$ is a smooth* **convex function** *in each contact-space rectangle $\mathcal{R}_{ij}$.*

Proof Each contact-space rectangle, $\mathcal{R}_{ij}$, is associated with a placement of the fingers on two particular edges of $\mathcal{B}$. Let t_1 and t_2 denote the *unit* tangents to these edges, and let p_0 be the intersection point of the lines underlying these two edges in $\mathbb{R}^2$. Along these edges, the boundary parametrization takes the form $p(s_1) = p_0 + s_1 t_1$ and $p(s_2) = p_0 + s_2 t_2$, such that (s_1, s_2) vary in $\mathcal{R}_{ij}$. Using the notation $\mathbf{s} = (s_1, s_2)$ and substituting for $p(s_1)$ and $p(s_2)$ in $d(s_1, s_2) = \|p(s_1) - p(s_2)\|$ gives

$$d(s_1, s_2) = \left(\mathbf{s}^T K \mathbf{s}\right)^{\frac{1}{2}}, \tag{2.3}$$

where

$$K = \begin{bmatrix} 1 & -t_1 \cdot t_2 \\ -t_1 \cdot t_2 & 1 \end{bmatrix}.$$

The eigenvalues of K are $1 \pm t_1 \cdot t_2$. Since $\|t_1 \cdot t_2\| \leqslant 1$, the two eigenvalues are non-negative. Hence K is positive semi-definite, and $d(s_1, s_2) = \|K^{1/2}\mathbf{s}\|$. The function $d(s_1, s_2)$ is thus a composition of the Euclidean norm (a convex function) with the linear function $K^{1/2}\mathbf{s}$. Such a composition preserves convexity, hence $d(s_1, s_2)$ is convex in each $\mathcal{R}_{ij}$. □

The next two corollaries follow directly from Lemma 2.5.2 and the definition of convexity.

Corollary 2.5.3 *Every bounding line of a contact-space rectangle, $\mathcal{R}_{ij}$, contains at most one critical point of $d(s_1, s_2)$ in its interior. If such a point exists, it is a local minimum of the restriction of $d(s_1, s_2)$ to this line.*

When $d(s_1, s_2)$ has a critical point in the interior of a bounding line of $\mathcal{R}_{ij}$, the fingers contact a vertex and the perpendicular projection of the vertex on an opposing edge of $\mathcal{B}$. Note that such critical points can be local minima or saddles of $d(s_1, s_2)$.

Corollary 2.5.4 *Let (s_1, s_2) and (s'_1, s'_2) be two points on the boundary of a single contact-space rectangle, $\mathcal{R}_{ij}$. The straight line path between these two points lies in a single connected component of the c-sublevel set $\mathcal{U}_{\leqslant c}$ where $c = \max\{d(s_1, s_2), d(s'_1, s'_2)\}$.*

The critical points of $d(s_1, s_2)$ in $\mathcal{U}$ will form the caging graph nodes. To ensure that all critical points are isolated points (and hence well defined nodes), a segment of degenerate critical points associated with parallel edges of $\mathcal{B}$ will be converted into two isolated points as follows. Each segment of degenerate critical points lies in a particular rectangle, $\mathcal{R}_{ij}$, with its endpoints located on bounding lines of $\mathcal{R}_{ij}$. Hence, without loss of generality, we retain only the *endpoints* of such a line segment. The following proposition will form a basis for the caging graph construction.

Proposition 2.5.5 *Let $\mathcal{U}$ be the contact space of a polygonal object $\mathcal{B}$. The* **critical points** *of $d(s_1, s_2)$ in $\mathcal{U}$, excluding the diagonal Δ and using the endpoints of degenerate critical points associated with parallel edges, are isolated points on the* **bounding lines** *of the contact space rectangles.*

Proof The critical points of $d(s_1, s_2)$ can be edge-edge, vertex-edge, or vertex-vertex critical points according to Lemma 2.4.1. The edge-edge critical points occur on parallel edges of $\mathcal{B}$. Such points have been converted to isolated vertex-edge critical points. Thus, excluding the diagonal Δ, all critical points of $d(s_1, s_2)$ involve at least one vertex of $\mathcal{B}$, and hence must lie on bounding lines of the contact space rectangles. A vertex-edge critical point is necessarily an isolated point in $\mathcal{U}$, since any local shifting of the finger contact along the edge opposing the vertex will give an edge normal which no longer passes through the vertex. A vertex-vertex critical point is, by construction, an isolated point in $\mathcal{U}$. □

Adapting the algorithm to handle polygonal objects having parallel edges is relatively straightforward. Consider the set formed by the points on one edge whose perpendicular projection onto the other edge lies on the interior of that other edge. If this set is non-empty, and not a single point, then it forms a non-isolated critical point which is a straight line (of unity slope) in contact space. Its endpoints lie on the boundary of that contact space rectangle, and are each endpoint would be a node in the *caging graph* described in the following section. The algorithm presented will handle parallel edges by treating this pair of nodes as a single node of that graph.

2.6 The Contact Space Caging Graph

This section describes the *caging graph,* which captures the topology of the sublevel sets of $d(s_1, s_2)$ in $\mathcal{U}$. The graph will be the basis for the caging algorithm described in Section 2.7.

2.6.1 The Caging Graph

Definition 2.6.1 *The* **caging graph**, *G, is an undirected graph with the following nodes and edges:*

Nodes: *The nodes of G correspond to finger placements either on pairs of vertices of $\mathcal{B}$, called* **vertex-vertex nodes**, *or at a vertex of $\mathcal{B}$ and its perpendicular projection onto an edge of $\mathcal{B}$, called* **vertex-edge nodes.**

Edges: *All nodes lying in the* same *contact-space rectangle, $\mathcal{R}_{ij}$, are connected to each other by edges in G.*

The nodes of G are located at the corners of each contact-space rectangle, $\mathcal{R}_{ij}$, and possibly at a single interior point on each bounding line of $\mathcal{R}_{ij}$. The edges of G connect each node on the boundary of $\mathcal{R}_{ij}$ to all other nodes on the boundary of $\mathcal{R}_{ij}$. Note that all such nodes are critical points of the distance function in contact space because the distance function is non-smooth at all vertices, and thus along all edges of contact space rectangles. Thus, vertex-vertex nodes, which lie at corners of contact space rectangles are non-smooth in two dimensions, while vertex-edge nodes lie on the edge of a contact-space rectangle, and are at a minimum along that edge. However, minima, saddles, and maxima of the distance function will only occur at the grasps shown in Figs. 2.2-2.4.

Remark: The contact-space scheme of Pipattanasomporn and Sudsang [25] proposed a *crawling graph* analogous to the caging graph. The crawling graph's nodes represent all edge pairs of $\mathcal{B}$, while its edges represent transitions between neighboring edges along the object's boundary. The crawling graph cannot be easily embedded in contact space $\mathcal{U}$, but it offers an alternative graph that can be used to search for the caging sets surrounding an immobilizing grasp of $\mathcal{B}$. ○

Example: Consider the polygonal object shown in Figure 2.5. The contours of $d(s_1, s_2)$ in $\mathcal{U}$ for this object are shown in Figure 2.7, with the axes labeled by arc-length along the object's perimeter. The nodes of G are marked by circles, and they are all located on bounding lines of the contact space rectangles. The full caging graph G is shown in Figure 2.8. Note that this graph, like contact space, has the topology of a 2-torus. For instance, the graph nodes at the four corner of $\mathcal{U}$ represent the same two-finger placement at the object vertex x_1. ○

The following lemma asserts that all critical points of $d(s_1, s_2)$ are located at nodes of G.

Lemma 2.6.2 *All the critical points of $d(s_1, s_2)$ in $\mathcal{U}$, except for those on the diagonal Δ, are located at nodes of the caging graph G.*

Proof The isolated critical points of $d(s_1, s_2)$ in $\mathcal{U}$ lie on bounding lines of the contact space rectangles according to Proposition 2.5.5. Critical points at the rectangles' corners (vertex-vertex grasps) are all nodes of G. Critical points in the interior of the rectangles' bounding lines (vertex-edge grasps) can only occur at a single point according to Corollary 2.5.3. Every such critical point corresponds to a vertex of $\mathcal{B}$ having a perpendicular projection onto an opposing edge of $\mathcal{B}$, and

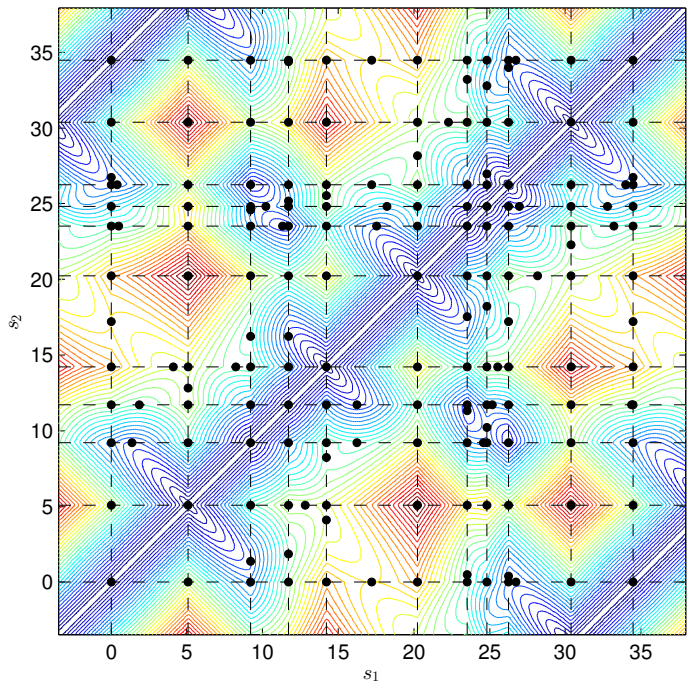


Figure 2.7: Contours of $d(s_1, s_2)$ for the object in Figure 2.5. Contact space rectangles are delineated with dashed lines. The nodes of G are marked with circles.

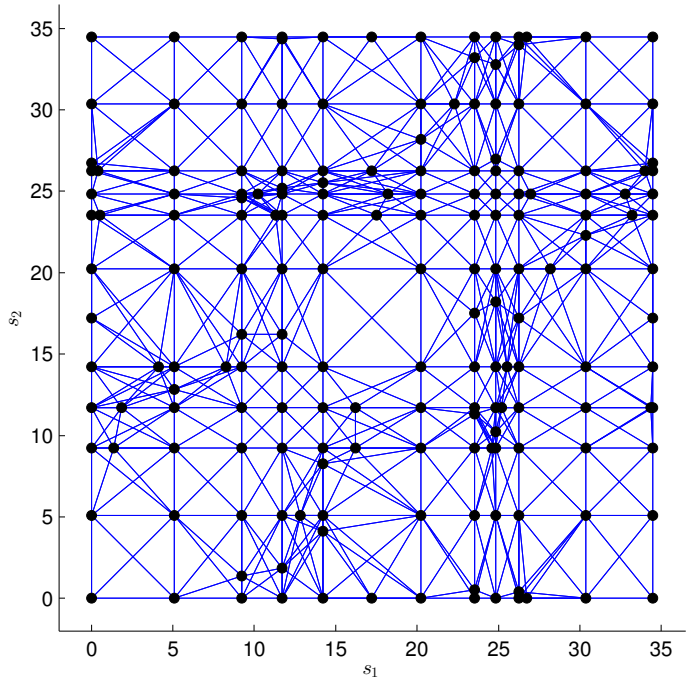


Figure 2.8: The caging graph G for the object in Figure 2.5. Circles represent the nodes of G, shown in Figure 2.7. Lines represent the edges of G. Note that the graph has the topology of a 2-torus: nodes on the left edge correspond to nodes on the right edge, and similarly for nodes on the top and bottom edges.

hence is also a node of G. The non-isolated critical points of $d(s_1, s_2)$ are associated with finger placements on two parallel edges of $\mathcal{B}$. These critical points span line segments with endpoints on bounding lines of the contact space rectangles. At each endpoint, one finger touches a vertex of $\mathcal{B}$ while the other finger touches an interior point of an opposing edge of $\mathcal{B}$. These endpoints are therefore vertex-edge nodes of G. □

It is important to note that the caging graph, G, contains many escape nodes. These nodes lie at the two corner points of each contact-space rectangle $\mathcal{R}_{ii}$ for $i = 1 \ldots n$ on the diagonal Δ. To ensure that the caging graph can be used to search for the caging sets, we introduce the notion of *discrete sublevel sets* in G, then show that G preserves the sublevel connectivity of $\mathcal{U}$.

Definition 2.6.3 *A discrete* **c-sublevel set** *of the caging graph G is the set of nodes, $v(s_1, s_2)$, given by $G_{\leqslant c} = \{v(s_1, s_2) \in G : d(s_1, s_2) \leqslant c\}$.*

The following theorem asserts that G preserves the sublevel connectivity of contact space $\mathcal{U}$.

Theorem 2.6.4 *The caging graph G is* **sublevel equivalent** *to contact space $\mathcal{U}$. That is, there exists a path in $\mathcal{U}$ between two nodes of G, v_i and v_j, lying entirely in $\mathcal{U}_{\leqslant c}$ if and only if there exists a path along the edges of G between v_i and v_j lying entirely in $G_{\leqslant c}$.*

Proof First consider a contact space path, α, which connects two nodes, v_i and v_j, while lying entirely in $\mathcal{U}_{\leqslant c}$. Divide this path into segments, such that each segment lies in a single contact space rectangle. Next, replace each path segment with a straight line path between the segment's endpoints. By convexity of $d(s_1, s_2)$ on the individual contact space rectangles, the maximum value of $d(s_1, s_2)$ on each linear segment is upper bounded by the maximum value of $d(s_1, s_2)$ on each original path segment. We now have a piecewise linear path connecting v_i and v_j. On each linear segment of this path, the value of $d(s_1, s_2)$ is upper bounded by c according to Corollary 2.5.4. The two endpoints of each linear segment lie on the boundary of some rectangle $\mathcal{R}_{ij}$. Finally, shift each endpoint, if it does not lie exactly at a node of G, to the minimum point of $d(s_1, s_2)$ along the same bounding line of $\mathcal{R}_{ij}$, which is always a node of G according to Corollary 2.5.3. This local shifting can only *decrease* the value of $d(s_1, s_2)$ for that endpoint. The equivalent path in G thus lies entirely in the discrete sublevel set $G_{\leqslant c}$.

Next consider a caging graph path, β, which connects two nodes v_i and v_j while lying entirely in $G_{\leqslant c}$. For each pair of adjacent nodes along the path, replace the caging graph edge connecting these nodes by a straight line segment embedded in $\mathcal{U}$. Each of these line segments lies in one rectangle R_{ij}. The maximum value of $d(s_1, s_2)$ along each of these path segments cannot be greater than the value of $d(s_1, s_2)$ at either endpoint according to Corollary 2.5.4. The entire piecewise linear path thus lies in $\mathcal{U}_{\leqslant c}$. □

The following corollary follows directly from Theorem 2.6.4.

Corollary 2.6.5 *The caging graph G has two properties:*

1. *Each connected component of $\mathcal{U}_{\leqslant c}$ discretizes into a connected subgraph of G containing all critical points of $d(s_1, s_2)$ lying in the connected component of $\mathcal{U}_{\leqslant c}$ and no others.*

2. *Every pair of sublevel sets of $d(s_1, s_2)$ in $\mathcal{U}$ that meet at a saddle point, v, discretizes into two subgraphs of G that meet at the corresponding node, v, of G.*

2.6.2 The Augmented Caging Graph

Theorem 2.6.4 ensures the sublevel equivalence of contact space, $\mathcal{U}$, and the caging graph, G. However, it is critical that $\mathcal{U}$ be sublevel equivalent with the free c-space, $\mathcal{F}$, as well. For many objects this holds true, but for certain objects having handle-like features, sublevel equivalence of $\mathcal{U}$ and $\mathcal{F}$ can fail.

Example: This possibility is illustrated in Figure 2.9. Starting at the immobilizing grasp (marked with $\circ$, green), the maximal puncture grasp (marked with $\rhd$, red) occurs at $\sigma = d^*$. It is clear that there exists a d^*-sublevel path in $\mathcal{F}$ from the immobilizing grasp to both the non-feasible local minimum ($\circ$, blue) and the local puncture ($\triangledown$, yellow). However, no such path exists in $\mathcal{U}$. Hence $\mathcal{U}$ is *not* sublevel equivalent with $\mathcal{F}$ for this object. $\circ$

In order to restore sublevel equivalence of $\mathcal{U}$ with $\mathcal{F}$, this section introduces *tunnel curves* whose attachment to $\mathcal{U}$ will ensure its sublevel equivalence with $\mathcal{F}$. The tunnel curves will become additional edges of the caging graph G, thus ensuring that a search of the augmented caging graph will always find the correct caging sets for any given object.

To clarify where tunnel curves are needed, consider the double-contact submanifold $\mathcal{S}$ embedded in the free c-space $\mathcal{F}$. Let us examine the effect of small changes in the fingers' opening parameter, σ, on the connectivity of the σ-sublevel sets in $\mathcal{F}$, and compare it to the connectivity of the σ-sublevel sets in the submanifold $\mathcal{S}$. Starting at an immobilizing hand configuration, $(q_0, \sigma_0) \in \mathcal{S}$, the connected component of this point in $\mathcal{S}_{\leqslant \sigma_0}$ as well as in $\mathcal{F}_{\leqslant \sigma_0}$ is the isolated point (q_0, σ_0). As σ increases by a small amount to $\sigma_0 + \epsilon$, the sublevel set $\mathcal{S}_{\leqslant \sigma_0 + \epsilon}$ expands locally around (q_0, σ_0), forming a connected set in $\mathcal{S}$. Since this set is connected, a σ-sublevel path connecting any two points in this set can be constructed entirely within $\mathcal{S}$. Sublevel equivalence of $\mathcal{F}$ and $\mathcal{S}$ (and hence $\mathcal{U}$) fails when, for some $\delta > 0$, the sublevel set $\mathcal{S}_{\leqslant \sigma_0 + \delta}$ ceases to be path-connected in $\mathcal{S}$, although it still lies in a *single* connected component of the ambient sublevel set $\mathcal{F}_{\leqslant \sigma_0 + \delta}$. This event always occurs at a local minimum of the function $\pi(q, \sigma) = \sigma$ in $\mathcal{S}$, such that the local minimum is *not* an immobilizing grasp of $\mathcal{B}$. At such a local minimum, a new connected component of the sublevel set $\mathcal{S}_{\leqslant \sigma_0 + \delta}$ appears, and the sublevel equivalence of $\mathcal{S}$ (and hence $\mathcal{U}$) with $\mathcal{F}$ breaks down.

The *tunnel curves* lie in the free c-space, $\mathcal{F}$, with their endpoints located on the submanifold $\mathcal{S}$. Each tunnel curve starts at a non-immobilizing local minimum of π in $\mathcal{S}$, follows a σ-decreasing path in $\mathcal{F}$ until reaching $\mathcal{S}$ at a lower σ value, and then continues within $\mathcal{S}$ until reaching a point corresponding to a node of G. The tunnel curves are constructed as follows.

Tunnel curves construction: Let (q_0', σ_0') be a local minimum of the function $\pi(q, \sigma) = \sigma$ in $\mathcal{S}$, which is *not* an immobilizing grasp of $\mathcal{B}$. For instance, Figure 2.9 depicts such a non-feasible local minimum, where one finger is located at a vertex while the other finger is located at an interior point of an opposing edge of $\mathcal{B}$. Starting at (q_0', σ_0'), at least one finger will be able to locally move away from $\mathcal{B}$ in a straight line toward the other finger (see proof of Theorem 2.6.7). Retract this finger while holding the other finger fixed on the object's boundary, until the finger hits a new edge of $\mathcal{B}$ (if the retracting finger hits the stationary finger, this gives an escape point discussed below). At this stage both fingers contact the object. Slide both fingers simultaneously along their respective object edges while minimizing σ (i.e. squeeze both fingers), until reaching the unique minimum of the inter-finger distance along the current object edges. This defines the tunnel-curve's other endpoint. Figure 2.10 shows the tunnel curve, which starts with a single finger retracting and continues with a squeezing of both fingers to a local minimum. If the fingers meet during the closing process, their contact space point is located on the diagonal, Δ, where $\sigma = 0$. In this case the current contact-space rectangle contains two escape nodes at its corners. Set the tunnel curve's other endpoint at the closest escape node along the diagonal Δ.

Based on the construction procedure, the tunnel curves are defined as follows.

Definition 2.6.6 *Let $\mathcal{S}$ be the double-contact submanifold parametrized by contact space $\mathcal{U}$. A* **tunnel curve** *starts at a non-immobilizing local minimum of $\pi(q, \sigma) = \sigma$ in $\mathcal{S}$, moves with decreasing σ in the free c-space, $\mathcal{F}$, then continues within $\mathcal{S}$ to the endpoint located at a node of G as described above.*

Remark: The contact space scheme [25] also augments the *crawling graph* with special transition edges analogous to the tunnel curves. These special transition edges are added when the two fingers reach along the object's boundary a corner point of $\mathcal{B}$, where one finger can break away from the boundary towards the opposing finger. While the number of such transition edges is significantly larger than the number of tunnel curves, these edges offer a means to augment the crawling graph with curves that ensure sub-level equivalence with the ambient free c-space $\mathcal{F}$. ○

Let $\mathcal{T}$ be set of all tunnel curves in $\mathcal{F}$. The following theorem asserts that the union $\mathcal{S} \cup \mathcal{T}$ (and hence the union $\mathcal{U} \cup \mathcal{T}$) is sublevel equivalent to the free c-space $\mathcal{F}$.

Theorem 2.6.7 *Let (q_0, σ_0) be an immobilizing grasp of $\mathcal{B}$, and let (q_{esc}, σ_{esc}) be the maximal puncture point associated with (q_0, σ_0). The union of the double-contact submanifold $\mathcal{S}$ with the*

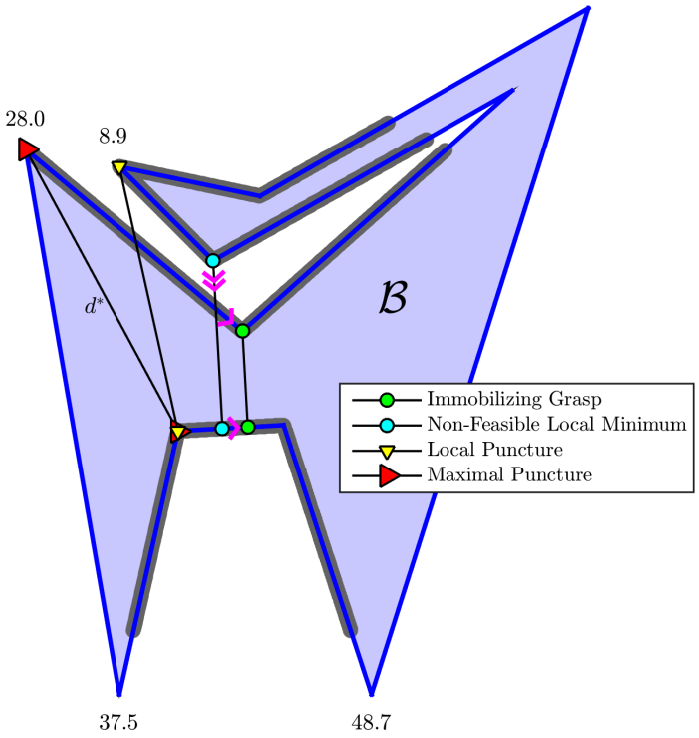


Figure 2.9: A polygonal object $\mathcal{B}$ with a handle-like feature. Selected grasps are shown.

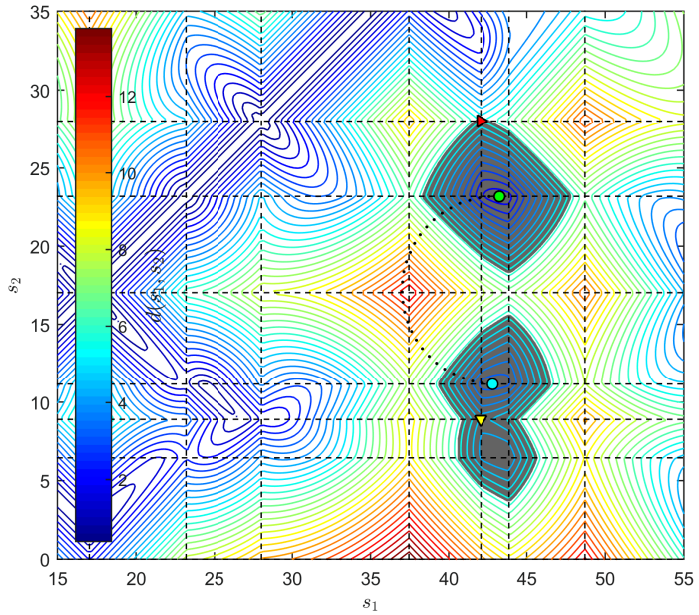


Figure 2.10: Contact space contours of $d(s_1, s_2)$ and selected grasps for the object of Figure 2.9. Portions of $\mathcal{U}_{\leqslant d^*}$ are shown shaded; the two disjoint regions (which are connected in the free c-space) are problematic. The dotted line represents a *tunnel curve* which does not lie in contact space $\mathcal{U}$.

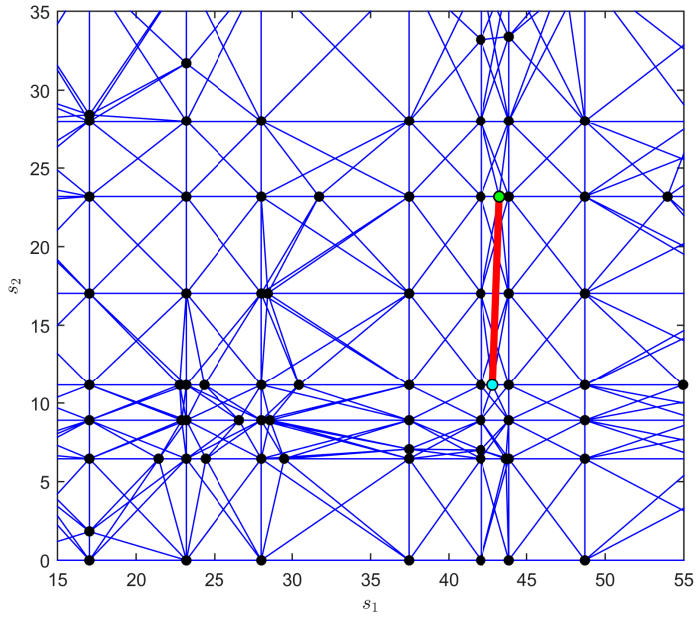


Figure 2.11: A portion of the augmented caging graph for the object of Figure 2.9, corresponding to the contact space region shown in Figure 2.10. The tunnel curve edge is depicted as a thick red line.

tunnel curves, $\mathcal{S} \cup \mathcal{T}$*, is* **sublevel equivalent** *to the connected component of* $\mathcal{F}_{\leqslant \sigma}$ *containing* (q_0, σ_0) *for* $\sigma \in [\sigma_0, \sigma_{esc}]$.

A proof of Theorem 2.6.7 appears in the appendix. Each tunnel curve starts at a local minimum of $d(s_1, s_2)$ in $\mathcal{U}$ which is a non-feasible equilibrium grasp of $\mathcal{B}$. This start point is a node of G. The tunnel curve then moves in the free c-space $\mathcal{F}$ while monotonically decreasing the inter-finger distance, until establishing a new two-finger grasp at a lower finger opening. The tunnel curve continues with a squeezing motion along the object's current edges, until reaching the unique local minimum of $d(s_1, s_2)$ along the two object edges (Figure 2.10). The resulting endpoint is also a node of G. Each tunnel curve can therefore be thought of as a *handle* attached to contact space $\mathcal{U}$ at two points which are nodes of the caging graph G. This interpretation leads to the following definition of the *augmented caging graph.*

Definition 2.6.8 *Let* G *be the caging graph of a polygonal object* $\mathcal{B}$*. The* **augmented caging graph,** *denoted* G_T*, is the graph* G *augmented with edges corresponding to the tunnel curves in* $\mathcal{F}$*.*

Figure 2.11 shows a portion of the augmented caging graph, G_T, for the object of Figure 2.9, corresponding to the contact space region shown in Figure 2.10. The tunnel curve edge depicted in Figure 2.11 connects a non-feasible local minimum of $d(s_1, s_2)$ with another node of G, located at an immobilizing grasp of $\mathcal{B}$. The original caging graph, G, is sublevel equivalent to contact space $\mathcal{U}$ (and hence to the double-contact submanifold $\mathcal{S}$) according to Theorem 2.6.4. The union $\mathcal{S} \cup \mathcal{T}$

is sublevel equivalent to the free c-space $\mathcal{F}$ according to Theorem 2.6.7. Since $G_T = G \cup \mathcal{T}$, the augmented caging graph G_T is sublevel equivalent to the free c-space $\mathcal{F}$.

2.7 The Contact Space Algorithm

Starting at an immobilizing grasp of $\mathcal{B}$, the caging algorithm searches the augmented caging graph, G_T, using two lists of nodes: a list of open nodes, $\mathcal{O}$, and a list of closed nodes, $\mathcal{X}$. The *open list* is kept sorted in ascending order of the associated value of $d(s_1, s_2)$. The *closed list* is kept in the order by which each node was added to this list. To start, the node of G_T associated with the initial immobilizing grasp is marked as open and inserted into $\mathcal{O}$, while $\mathcal{X}$ is initially empty. At each step, the node in $\mathcal{O}$ with the smallest value of $d(s_1, s_2)$ is marked as *current*, removed from $\mathcal{O}$, and added to the *end* of the closed list $\mathcal{X}$. If the $d(s_1, s_2)$ value of the current node is zero, the algorithm halts, as an escape point has been found. Otherwise, each node adjacent to the current node in G, if not in $\mathcal{X}$ and not already in $\mathcal{O}$, is added to $\mathcal{O}$.

The algorithm halts after a finite number of steps, since there is always a maximal puncture point grasp (corresponding to a node of G_T), beyond which an escape point (corresponding to an escape node of G_T) can be reached. The algorithm's output is extracted from the closed list $\mathcal{X}$ as discussed below. In particular, the node in $\mathcal{X}$ having the maximal value of $d(s_1, s_2)$ is the *maximal* puncture point grasp associated with the initial immobilizing grasp. A pseudo-code of the caging algorithm follows.

Algorithm 2.1 Contact Space Caging Algorithm

Data structures: open list $\mathcal{O}$, closed list $\mathcal{X}$.
Initialize: $\mathcal{O}$ = initial immobilizing grasp, $\mathcal{X} = \varnothing$.
while $d(CurrentNode) > 0$ **do**
  Set $CurrentNode$ = lowest $d(s_1, s_2)$ value node in $\mathcal{O}$.
  Mark $CurrentNode$ as **explored.**
  Expand $CurrentNode$ (add all non-explored adjacent nodes in G_T to $\mathcal{O}$).
  Remove $CurrentNode$ from $\mathcal{O}$.
  Add $CurrentNode$ to end of $\mathcal{X}$.
end while
return list of closed nodes $\mathcal{X}$.

The caging algorithm runs on the augmented caging graph, G_T, which requires that tunnel curve edges be added to the caging graph G. These edges are added to G as follows. Consider a non-feasible local minimum of $d(s_1, s_2)$, at which finger 1 lies at p_1 and finger 2 lies at p_2. The node of G corresponding to this grasp is the start point of the tunnel curve. Assume that finger 1 is able to move towards p_2 without penetrating $\mathcal{B}$ (at every non-feasible local minima one or both fingers may be moved towards the other finger without penetrating $\mathcal{B}$). Find the intersection of the line segment joining p_1 to p_2 with all edges of $\mathcal{B}$. If no such intersection exists, the tunnel curve's end point is

located at an escape node of G, where both fingers lie at p_2. Otherwise, find the intersection which is closest to p_1. Generically, the grasp in which finger 1 lies at this intersection while finger 2 lies at p_2 is located in the *interior* of a contact space rectangle. The tunnel curve's end point is located at the node of G on the boundary of this contact space rectangle, having the minimum $d(s_1, s_2)$ value in this rectangle. The procedure is summarized in Algorithm 2.2.

Algorithm 2.2 Contact Space Tunnel Curve Construction

```
Input: caging graph, G.
Initialize: G_T = G.
for each non-feasible local minimum, v_n = (p_1, p_2) ∈ G, do
   Set tunnel curve start node, v_s = v_n.
   Define line segment l running from p_1 to p_2.
   Find all intersections of l with edges of B.
   if no such intersection point exists then
      set tunnel curve's end node, v_e = (p_2, p_2).
   else
      p_int = intersection point closest to p_1,
      R_ij = contact space rectangle containing (p_int, p_2),
      v_e = minimum inter-finger distance node in R_ij.
      Set tunnel curve's end node at v_e.
   end if
   Add edge from v_s to v_e to G_T.
end for
return G_T, the augmented caging graph.
```

Interpretation of the closed list: Consider the nodes $v \in \mathcal{X}$ of the closed list, indexed in the order they were added to $\mathcal{X}$. The first node added (the initial immobilizing grasp), is denoted v_0, and so on.

Definition 2.7.1 *The i^{th} node, v_i, in the closed list $\mathcal{X}$ is a* **local minimum** *in $\mathcal{X}$ if $d(v_i) < \min\{d(v_{i-1}), d(v_{i+1})\}$.*

Definition 2.7.2 *The i^{th} node, v_i, in the closed list $\mathcal{X}$ is a* **local maximum** *in $\mathcal{X}$ if $d(v_i) > \max\{d(v_{i-1}), d(v_{i+1})\}$.*

Definition 2.7.3 *The i^{th} node, v_i, in the closed list $\mathcal{X}$ is a* **puncture-related local maximum** *if:*

1. *It is a local maximum in $\mathcal{X}$.*
2. *There does not exist a lower indexed puncture-related local maximum in $\mathcal{X}$, v_j for $j < i$, such that $d(v_j) > d(v_i)$*
   (i.e. it is the highest value node so far in $\mathcal{X}$).

The interpretation of the closed list $\mathcal{X}$ is simple. The first element of $\mathcal{X}$, by construction, and all local minima are immobilizing grasps of $\mathcal{B}$. All puncture-related local maxima in $\mathcal{X}$ are puncture-point grasps of $\mathcal{B}$. While the node representing each of these punctures is a saddle of $d(s_1, s_2)$ in

$\mathcal{U}$, it appears as a local maximum in $\mathcal{X}$. This occurs because once the search algorithm discovers a saddle, it continues searching *lower valued* nodes in the unexplored portion of the previously disjoint sublevel set of the augmented caging graph, G_T. The lowest indexed puncture-related local maximum in $\mathcal{X}$ represents the puncture-point grasp of the local caging set surrounding the immobilizing grasp. The highest indexed puncture-related local maximum in $\mathcal{X}$ represents the maximal puncture-point grasp of the maximal caging set, beyond which the object $\mathcal{B}$ can escape to infinity. Finally, all other puncture-related local maxima represent puncture-point grasps associated with all intermediate caging sets surrounding the immobilizing grasp.

Computational complexity: Let the object $\mathcal{B}$ have n vertices and edges. Each vertex of $\mathcal{B}$ induces at most $2n$ nodes (n vertex-vertex pairs and n vertex-edge pairs) in the caging graph G. Hence there are $\mathcal{O}(n^2)$ nodes in G. The open list, $\mathcal{O}$, contains at most $\mathcal{O}(n^2)$ nodes, which may be kept sorted in $\mathcal{O}(n^2 \log n)$ time. Additionally, the object $\mathcal{B}$ may have up to $\mathcal{O}(n^2)$ non-feasible equilibrium grasps that may induce tunnel curves. The construction of $\mathcal{O}(n^2)$ tunnel curves may be done in $\mathcal{O}(n^2\ log\ n)$ time using ray shooting algorithms. Based on these considerations, the caging algorithm runs in $\mathcal{O}(n^2 \log n)$ time. Additionally, while the algorithm searches for the initial, intermediate, and maximal caging sets surrounding an initial immobilizing grasp, it can be adapted to find the maximal caging sets surrounding *all* immobilizing grasps of $\mathcal{B}$, also in $\mathcal{O}(n^2\ log\ n)$ time, which is comparable to [27, 43].

2.8 Algorithm Walk Through

Returning to the example show in Fig. 2.5, we will step through the algorithm. The grasps explored are shown in physical space in Fig. 2.12 and in contact space in Fig. 2.13. The algorithm starts at an initial immobilizing grasp, marked with green circles. It next discovers a local puncture, marked with orange triangles, and then finds a second immobilizing grasp, marked with blue circles. It explores several grasps which are not immobilizing or puncture, show in contact space as black circles. (For clarity, these grasps are not shown in physical space.) I then discovers the maximal puncture grasp, marked with red triangles, and moves directly to the escape point, marked with a magenta square, thus completing the search.

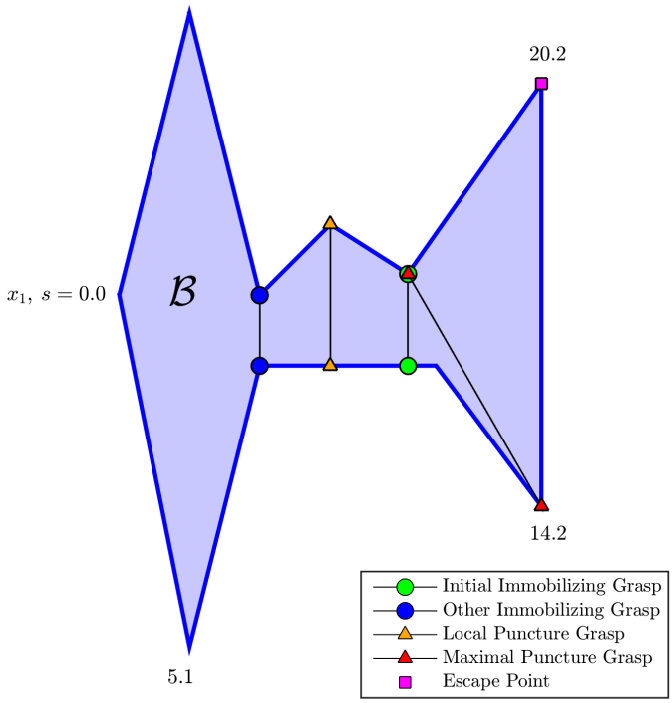


Figure 2.12: Important grasps discovered during the exploration of the polygon shown in physical space.

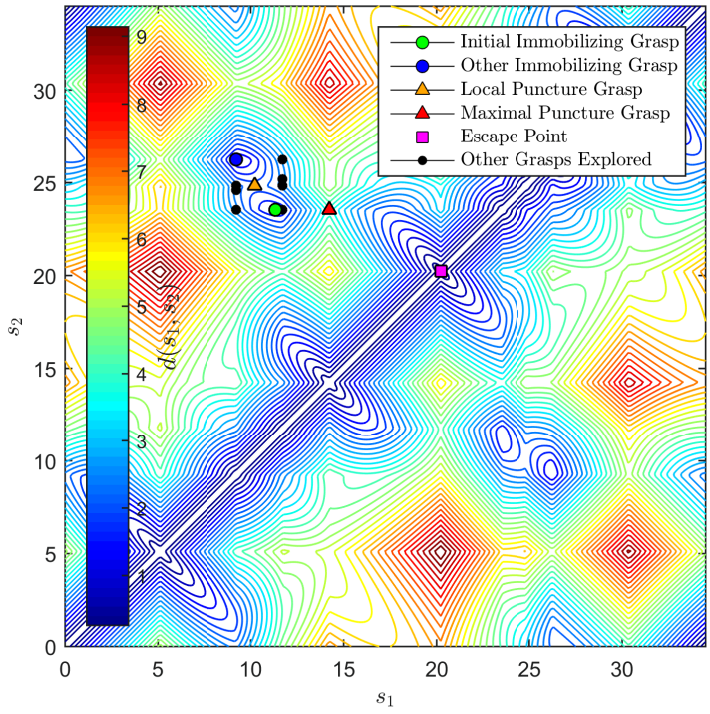


Figure 2.13: Important grasps discovered during the exploration of the polygon shown in contact.

2.9 Graphical Depiction of Caging Set as Two Capture Regions

This section briefly describes the graphical rendering of the caging sets as *capture regions* in $\mathbb{R}^2$.[5] Let (q_0, σ_0) be the initial immobilizing grasp, and let (q_1, σ_1) be the puncture point of the caging set associated with the immobilizing grasp. As a preliminary step, determine the object's boundary segments within the capture regions as follows. Place the two fingers at the immobilizing grasp, then slide both fingers in contact with the object's boundary while increasing the fingers' opening parameter from σ_0 to σ_1. Mark the object's boundary curves touched by the fingers during this process as γ_1 and γ_2 (Figure 2.14). Next, slide one finger along γ_1 while marking the curve located at a perpendicular distance of σ_1 on the object's opposite side. Denote by γ_1' the portion of this curve bounded by the endpoints of γ_2 (Figure 2.14). Repeat this process with the curve γ_2, this time generating a curve γ_2' with endpoints at those of γ_1. One capture region is bounded by γ_1 and γ_2', while the other capture region is bounded by γ_2 and γ_1'. The technique is illustrated in the following example.

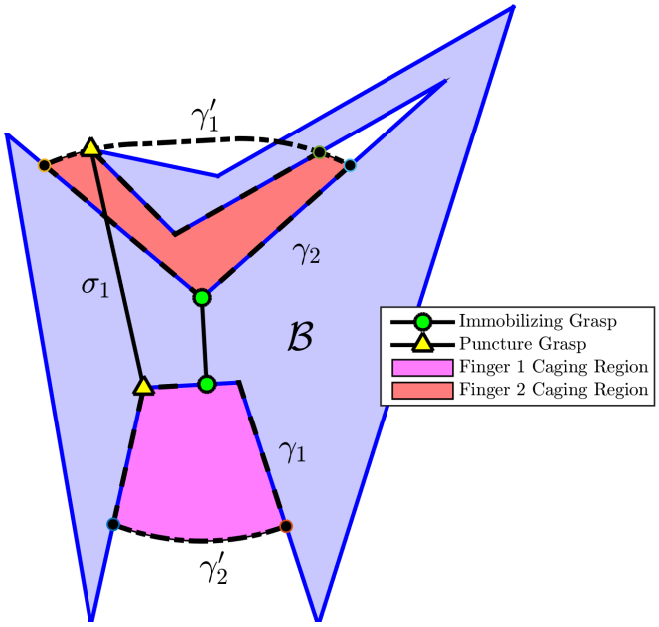


Figure 2.14: The capture regions surrounding an immobilizing grasp of $\mathcal{B}$.

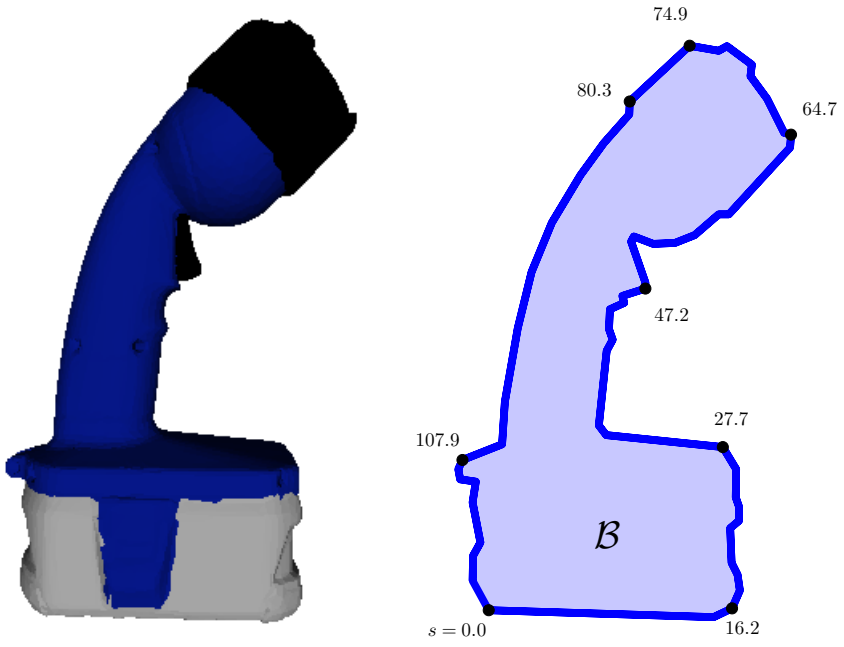
Example: Consider the polygonal object of the previous example, depicted again in Figure 2.14. Starting at the immobilizing grasp, the boundary segments touched by the fingers while increasing the fingers' opening parameter from σ_0 to σ_1, γ_1 and γ_2, are depicted in Figure 2.14. The tracing of γ_1 at a perpendicular distance σ_1 yields the curve γ_1' whose endpoints coincide with those of

[5] The graphical rendering of the capture regions requires a technical condition which holds for all reasonable objects. See [33] for this condition.

γ_2. A similar tracing of γ_2 gives the curve γ_2' whose endpoints coincide with those of γ_1. Note that both curve tracings involve rotation of the perpendicular segment at the object's vertices, with the rotation range determined by the edge normals at the vertex. The lower finger capture region is bounded by γ_1 and γ_2'. The upper finger capture region is bounded by γ_2 and γ_1', as well as by a portion of the object's boundary which penetrates this region. Every two-finger placement in the shaded capture regions with finger opening $\sigma \leqslant \sigma_1$ will retract to the immobilizing grasp at the object's center. The two capture regions together with the condition $\sigma \leqslant \sigma_1$ thus provide a 2D representation of the four-dimensional caging set. ○

2.10 Caging Set Computational Example

Here we present a practical implementation of the caging algorithm on a flashlight, provided as part of DARPA's Autonomous Robotic Manipulation (ARM) program [22]. The flashlight geometric data was provided off-line using a laser scanner, but could be attained in real time from stereo vision or LIDAR. First, the point cloud data was projected onto a plane, and its outer envelope was approximated as a 2D polygon, $\mathcal{B}$. The original point cloud data and its polygonal approximation are shown in Figure 2.15.



(a) htbp (b) htbp

Figure 2.15: Input data to the caging algorithm: (a) 3D triangular mesh data, and (b) a 2D polygonal projection of the point cloud data, with arc-length value of selected vertices.

Next the polygon $\mathcal{B}$, along with an initial immobilizing grasp specified near the flashlight's trigger,

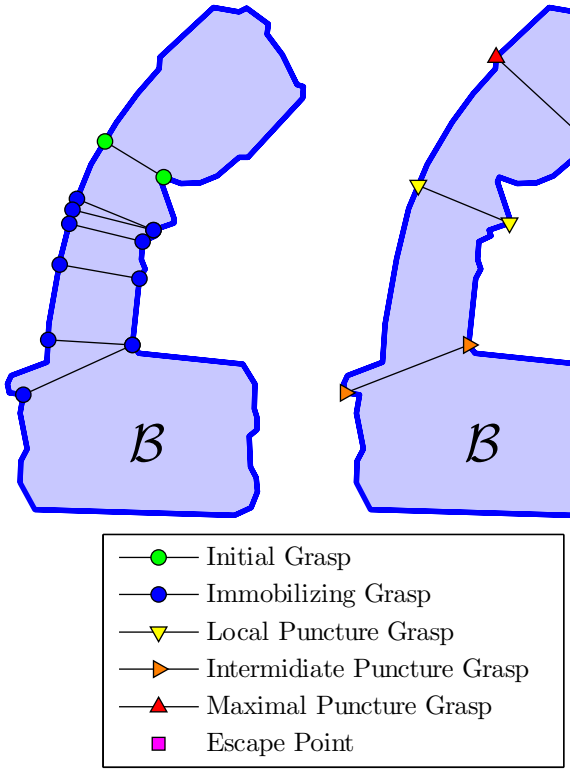


Figure 2.16: Critical grasps and escape point for the flashlight in physical space.

were provided as input to the caging algorithm (Figure 2.16). Starting from the immobilizing grasp, the algorithm explored the caging graph with increasing finger opening parameter. A contour plot of $d(s_1, s_2)$ is shown in Figure 2.17, along with the critical grasps and nodes in the open list $\mathcal{O}$ (gray circles) and the closed list $\mathcal{X}$ (dark circles), developed during algorithm execution. Note that less than 6% of the total caging graph nodes were actually explored during the search.

A graph of $d(s_1, s_2)$ at the nodes of $\mathcal{X}$ developed by the algorithm is shown in Figure 2.18. The local minima in the graph indicate immobilizing grasps, while the local maxima which are highest so far in $\mathcal{X}$ indicate puncture grasps. Based on this graph, the caging algorithm returned the initial caging set, several intermediate caging sets, and the maximal caging set surrounding the immobilizing grasp. The neighboring immobilizing grasps and the series of puncture grasps computed by the algorithm are shown in physical space in Figure 2.16.

2.11 Extensions of the Caging Algorithm

This section discusses three extensions of the basic caging algorithm. The first extension concerns caging objects with holes. Although an object cannot escape to infinity when a finger is placed in its hole, the local caging sets can still be useful for robust finger grasping. Let us therefore consider

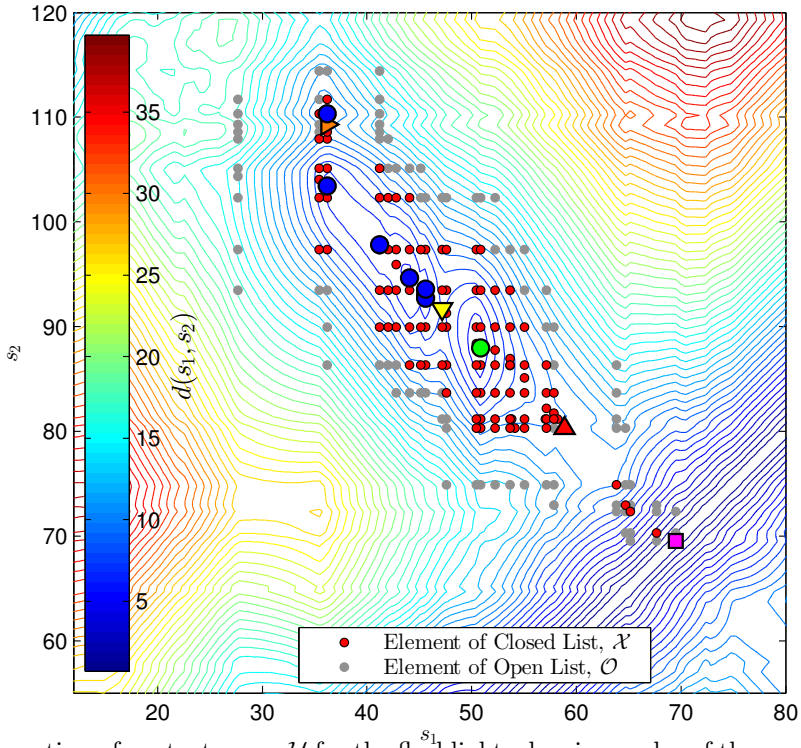


Figure 2.17: A portion of contact space $\mathcal{U}$ for the flashlight, showing nodes of the open list $\mathcal{O}$ and the closed list $\mathcal{X}$. Critical grasps in the closed list $\mathcal{X}$ are shown with the same symbols as in Figure 2.16.

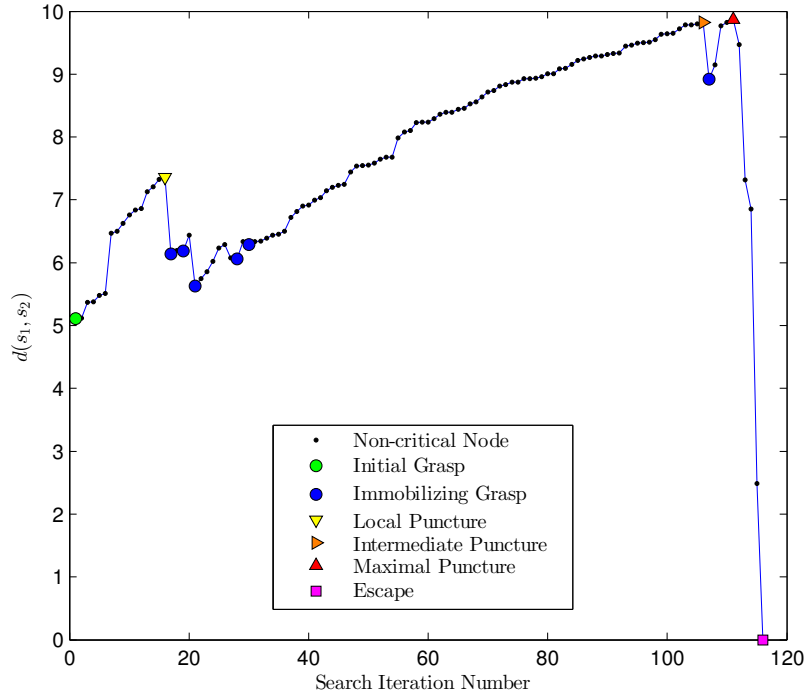


Figure 2.18: A graph of $d(s_1, s_2)$ at the nodes of the closed list $\mathcal{X}$, showing the exploration of the object's boundary along with critical grasps and the escape point. The graph's local maxima indicate puncture grasps (see text).

two cases. In the first case both fingers are placed inside a hole of $\mathcal{B}$ (Figure 2.19(a)). In this case contact space parametrizes the two-finger placements along the hole's boundary. Starting at an immobilizing grasp within the object's hole, an increasing finger opening will eventually reach a puncture grasp beyond which the two fingers can pinch together inside the hole (Figure 2.19(a)). This defines *escape points* for the caging algorithm, which can now identify the initial, intermediate, and maximal caging sets within the object's hole.

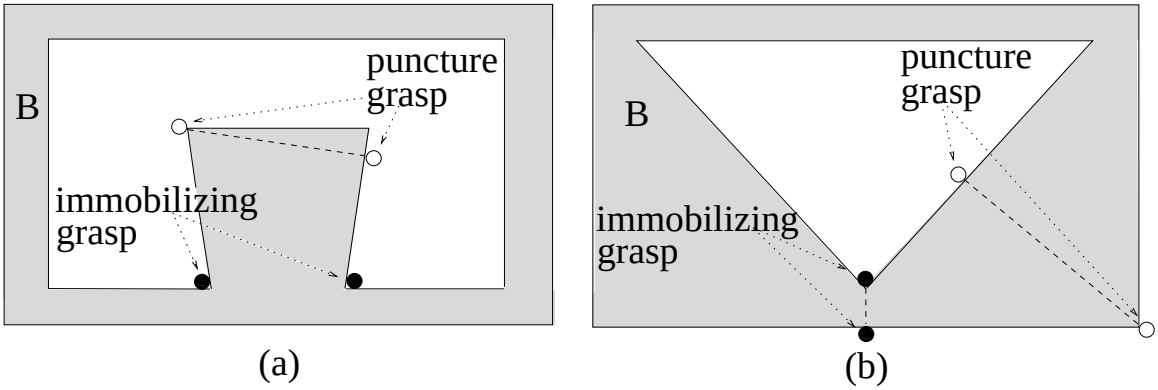


Figure 2.19: Two-finger immobilization and its puncture grasp. (a) Both fingers are placed within a hole of $\mathcal{B}$. (b) One finger is placed in a hole, the other finger is placed outside $\mathcal{B}$.

A more interesting case occurs when one finger is placed inside a hole of $\mathcal{B}$, while the other finger is placed outside the object (Figure 2.19(b)). Contact space parametrizes the two-finger placements along the object's outer and inner boundaries. Starting at an immobilizing grasp, an increasing finger opening will eventually reach a critical puncture grasp beyond which the two-fingered hand can complete a full 360° rotation while maintaining a fixed finger opening (Figure 2.19(b)). Equivalently, the object will be able to complete a full 360° rotation in $\mathbb{R}^2$ while the fingers are kept stationary. The basic caging algorithm will have to be slightly modified in order to identify full rotation nodes in the caging graph. Such nodes occur at the first puncture grasp whose sublevel set in $\mathcal{U}$ covers the entire outer and inner boundaries of $\mathcal{B}$. At this puncture grasp the sublevel set of $d(s_1, s_2)$ in $\mathcal{U}$ becomes periodic in both s_1 and s_2. By monitoring arrival to such nodes during the caging graph exploration, the algorithm will be able to report the caging sets surrounding the immobilizing grasp, until a full 360° rotation of the object $\mathcal{B}$ becomes feasible.

A second extension of the algorithm concerns *stretching cages,* where the fingers open outward in order to grasp the object [38, 43]. The object $\mathcal{B}$ is immobilized by a stretching grasp when the two fingers are located at opposing concave vertices of $\mathcal{B}$ (Figure 2.4(d)). Such grasps are local maxima of $d(s_1, s_2)$. As the two fingers move closer together, they will eventually reach a puncture grasp which allows the two-fingered hand to escape to infinity, or re-open and reach a neighboring immobilizing grasp. In order to extend the algorithm to stretching cages, one has to ensure that $\mathcal{U}$

preserves the *superlevel set* connectivity of the free c-space $\mathcal{F}$,[6] then construct a caging graph, G, that preserves the superlevel set connectivity of $\mathcal{U}$.

To ensure superlevel equivalence, contact space and the caging graph must be augmented with tunnel curves, this time associated with local maxima of $d(s_1, s_2)$ representing non-feasible equilibrium grasps. The caging graph G will have to be modified as follows. The nodes of Gremain the same, while two edge types will have to be removed from G. If a bounding line of a contact space rectangle, $\mathcal{R}_{ij}$, contains an interior critical point of $d(s_1, s_2)$, the caging graph edge connecting the vertex-vertex nodes at the line's endpoints is removed from G. Similarly, edges that connect diagonally opposite corners of $\mathcal{R}_{ij}$ are removed from G (all other edges of G remain intact). A preliminary analysis indicates that the modified caging graph satisfies the superlevel equivalence required for computing stretching cages. However, these ideas must be carefully evaluated and verified on real-world examples.

A third extension concerns caging with two disc fingers. As a preliminary step, the object $\mathcal{B}$ must be converted into a c-space object, $\mathcal{CB}$, by expanding its boundary outward by the fingers' radius. However, $\mathcal{CB}$ contains straight line edges as well as circular arcs incurred by the object's convex vertices. When contact space is used to parametrize such circular arcs, the critical points of the inter-finger distance function, $d(s_1, s_2)$, no longer lie along the bounding lines of the contact space rectangle associated with such arcs. Rather than construct *generalized* caging graphs which account for such object boundaries, every circular arc on the boundary of $\mathcal{CB}$ can be approximated by a regular k-segment polygonal arc. The caging algorithm can then be executed on the polygonal approximation of $\mathcal{CB}$. The resulting caging sets will closely approximate the exact caging sets (with the quality of approximation increasing with k), and will usually identify the exact object vertex-edge pairs associated with puncture grasps of $\mathcal{B}$.

2.12 Summary and Extensions

This chapter described a technique for computing the two-finger caging sets surrounding an immobilizing grasp of a polygon, $\mathcal{B}$, using contact space. Rather than compute a cellular decomposition of the hand's four-dimensional configuration space, it suffices to consider the two-dimensional contact space, $\mathcal{U}$, parameterizing the finger positions along the object's boundary. Within contact space, the critical points of the inter-finger distance function, $d(s_1, s_2)$, can be used to compute the caging sets. The immobilizing grasps are local minima of $d(s_1, s_2)$, while the puncture grasps which determine the caging sets are saddles of $d(s_1, s_2)$. This chapter established several properties of the inter-finger distance function. It is convex on each contact space rectangle, and its critical points (except escape points on the diagonal) are located on the rectangles' bounding lines. Critical points on parallel

[6] Contact space $\mathcal{U}$ is *superlevel equivalent* to $\mathcal{F}$ if all points $p_1, p_2 \in \mathcal{S}$ that can be connected by a path in $\mathcal{F}$ satisfying $\sigma \geqslant c$ can also be connected by a path in $\mathcal{U}$ (equivalently, a path in $\mathcal{S}$) satisfying $d(s_1, s_2) \geqslant c$.

edges of $\mathcal{B}$ form segments of degenerate local minima, the endpoints of such segments also lie on the rectangles' bounding lines.

These properties lead to the construction of a *caging graph,* G, in contact space $\mathcal{U}$. The nodes of G contain all the critical points of $d(s_1, s_2)$, while the edges of G are constructed so as to preserve the sublevel set connectivity of the function $d(s_1, s_2)$ in $\mathcal{U}$. However, proper computation of the caging sets requires augmentation of $\mathcal{U}$ with *tunnel curves,* which form additional edges of the caging graph G. Every tunnel curve starts at a local minimum of $d(s_1, s_2)$ of a special type (a non-feasible equilibrium grasp of $\mathcal{B}$), and ends at at a node of G having a lower value of $d(s_1, s_2)$. This chapter established sublevel equivalence of the augmented caging graph with the hand's free c-space, $\mathcal{F}$, and thus laid the groundwork for a simple caging algorithm. Starting from the node of G representing a desired immobilizing grasp, the caging algorithm searches G for saddles where $d(s_1, s_2)$ attains successively increasing values. The puncture grasps corresponding to these saddles determine the initial, intermediate, and maximal caging sets in contact space. The entire computation takes $\mathcal{O}(n^2 \log n)$ time, where n is the number of vertices and edges of $\mathcal{B}$. The caging algorithm has been fully implemented under DARPA's Autonomous Robotic Manipulation program, and its output has been demonstrated on a real-world example.

The basic caging algorithm can be extended in several ways. First, objects with holes can be treated under the framework described in this chapter. Second, contact space seems to be equally useful for computing *stretching cages.* This chapter suggested a modified caging graph that preserves the superlevel connectivity of $\mathcal{U}$, such that its augmentation with tunnel curves satisfies superlevel equivalence with $\mathcal{F}$. However, these ideas have to be carefully evaluated and validated on real-world examples. A third extension concerns caging with two disc fingers. In this case $\mathcal{B}$ is first transformed into a c-space object, then a polygonal approximation of the c-space object can be processed by the caging algorithm described here. However, contact space can possibly be used to define a generalized caging graph which will allow computation of the caging sets associated with two disc fingers.

Chapter 3

Two Fingers in Three Dimensions

3.1 Introduction

This chapter extends the work from Chapter 2 to consider caging of 3D polyhedra, still using two point fingers. Many of the concepts and definitions are similar or identical to Chapter 2, but is written to be independent of that chapter. Analysis in the 2D case was done using Stratified Morse Theory. In contrast, this chapter utilizes simple geometric descriptions to perform similar functions.

Several works by Pipattanasomporn et al [27, 29] have examined caging of 3D polyhedra. The contact-space formulation presented in this chapter allows most of the computation to be done in the 4D contact space, rather than the object's 6D configuration space. This approach leads to an algorithm which is simple to implement and has sound computation efficiency.

The algorithm presented in this chapter takes as starting input a 3D polyhedron and an initial immobilizing grasp. Starting with that grasp, it searches a graph composed of critical points of the inter-finger distance function. It reports the local, intermediate, and maximal caging sets related to this immobilizing grasp.

Note that two point fingers cannot immobilize a polyhedron, as it will always be able to rotate about a line passing through both fingers. However, the two-fingered hand is immobilizing relative to the object. In this chapter, immobilizing refers to this situation.

This chapter is organized as follows. Section 3.2 defines the problem and introduces the inter-finger distance function, $d(\mathbf{s})$. Section 3.3 reformulates the 6D caging problem into 4D contact space. Section 3.4 introduces a discrete caging graph, G, whose nodes will be searched using the algorithm presented in Section 3.5. An example in Section 3.6 demonstrates the algorithm. Section 3.7 presents a geometric method to characterize all nodes of G, which results in a catalog of all squeezing, immobilizing, and puncture grasps, shown in Section 3.8. In certain cases, important information is lost when moving from configuration space to contact space. Section 3.9 shows when information is lost and how to restore it, such that the search algorithm is correct.

3.2 Preliminaries and Problem Statement

A 3D polyhedral object, $\mathcal{B}$, is to be caged by two point fingers, denoted f_1 and f_2. The polyhedron may not have cavities in its interior, but may have holes passing through it. Two-finger cages come in two types: *squeezing* cages, where the fingers move together to cage the object, and *stretching* cages, where the fingers move apart to cage the object [37, 43]. This work considers only squeezing cages. To simplify the explanation, we exclude polyhedra containing parallel opposing faces, parallel opposing edges, and edges and faces which are parallel and not coplanar, though the algorithm can handle these cases. Non-convex faces of $\mathcal{B}$ must be divided into a disjoint collection of convex faces.

Definition 3.2.1 *The* **configuration space** *(c-space) of a two-fingered hand, denoted $\mathcal{C}$, is the 6D space consisting of configurations $q = (p_1, p_2) \in \mathbb{R}^6$, where $p_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ represents the position of the i^{th} finger, for $i = 1, 2$.*

From the hand's point of view, the object $\mathcal{B}$ forms an obstacle. The *c-space obstacle* corresponding to $\mathcal{B}$, denoted $\mathcal{CB}$, is the set of hand configurations at which one or both fingers intersect the stationary object $\mathcal{B}$. The hand's *free c-space, $\mathcal{F}$*, is the complement of $\mathcal{CB}$'s interior. The boundary of $\mathcal{F}$ consists of all hand configurations at which one or both fingers touch the object's boundary.

Our approach is formulated in contact space.

Definition 3.2.2 ***Contact-space**, denoted $\mathcal{U}$, consists of all hand configurations at which both fingers touch the object boundary. Contact space is a 4-dimensional submanifold of $\mathcal{C}$, lying on the boundary of $\mathcal{F}$.*

Since the position of one finger on the surface of $\mathcal{B}$ may be parameterized using a subset of $\mathbb{R}^2$, $\mathcal{U}$ may be parameterized as a subset of $\mathbb{R}^4$. Denote a point in contact space with $\mathbf{s} = (s_1, s_2, s_3, s_4) \in \mathbb{R}^4$. For any point $\mathbf{s} \in \mathcal{U}$, the corresponding finger positions in $\mathcal{C}$ may be given by $q(\mathbf{s}) = (p_1(s_1, s_2), p_2(s_3, s_4))$. Note that implementation of this parameterization is not required for our caging algorithm.

Contact space can be naturally decomposed into a set of polychora (4-polytopes), which provide a useful structure to calculate caging sets.

Definition 3.2.3 *A **contact-space polychoron**, $\mathcal{P}_{ij}$ corresponds to all possible placements of f_1 at $p_1(s_1, s_2)$ on face i and f_2 at $p_2(s_3, s_4)$ on face j of $\mathcal{B}$.*

Each polychoron, $\mathcal{P}_{ij}$, is a convex set. The set of polychora completely covers contact space and their interiors are disjoint. The following definition will be critical to the consideration of caging.

Definition 3.2.4 *The **inter-finger distance function** is a scalar valued function giving the distance between the fingers. In $\mathcal{C}$, $d : \mathcal{C} \to \mathbb{R}$ and in $\mathcal{U}$, $d : \mathcal{U} \to \mathbb{R}$.*

$$d(q) = d(p_1, p_2) = ||p_1 - p_2|| \tag{3.1}$$

$$d(\boldsymbol{s}) = ||p_1(s_1, s_2) - p_2(s_1, s_2)|| \tag{3.2}$$

The notion of a *caging set* is formulated in $\mathcal{F}$ as follows.

Definition 3.2.5 *The* **caging set** *of $\mathcal{B}$ consists of all pairs of finger placements in $\mathcal{F}$ such that $\mathcal{B}$ cannot be moved arbitrarily far from the fingers.*

Definition 3.2.6 *A* **squeezing caging set** *of $\mathcal{B}$ is any pair of finger placements, $q_0 \in \mathcal{F}$, such that $\mathcal{B}$ cannot be moved arbitrarily far from the fingers while the inter-finger distance is constrained such that $d(q) \geqslant d(q_0)$.*

When two fingers initially holding $\mathcal{B}$ at an immobilizing grasp are moved apart, the object remains caged for a finite range of the finger opening distance. Eventually the inter-finger distance reaches a critical value beyond which the object can escape the cage formed by the two fingers. This critical finger opening may only allow an intermediate escape into a larger cage formed by other object features, or an ultimate escape to infinity. We propose to find those critical events by searching contact space, $\mathcal{U}$.

Problem Definition: Given an initial two-finger immobilizing grasp of $\mathcal{B}$, compute the following types of caging set:

1. The *initial caging set*—the largest caging set from which the fingers are guaranteed to return to the initial immobilizing grasp while keeping the object caged during the squeezing process.

2. The *intermediate caging set*—any caging set which contains the initial caging set, such that all fingers end at a finite number of possible immobilizing grasps while keeping the object caged during a squeeze.

3. The *maximal caging set*—the largest squeezing caging set which contains the initial caging set.

3.3 Contact-Space Reformulation of Caging

Rather than searching configuration space, $\mathcal{C} \in \mathbb{R}^6$, this chapter reduces the dimensionality of the problem by searching contact space, $\mathcal{U} \subset \mathbb{R}^4$. To show that a search in $\mathcal{U}$ can find the caging sets, we formulate the caging problem as the existence of a test path between two points lying entirely in a sublevel set of the function $d(\mathbf{s})$ in $\mathcal{U}$.

The idea of an *escape* plays a central role in caging theory. The standard definition of an escape is the ability to move the two-fingered hand with a fixed finger opening arbitrarily far from the object. The corresponding notion of an *escape* in contact space is defined as follows.

Definition 3.3.1 *A* **contact-space escape point** *in $\mathcal{U}$ is any point where the fingers are coincident on the boundary of a contact-space polychoron. We denote this as $\Delta = \{\boldsymbol{s} \in \mathcal{U} \mid p_1(s_1, s_2) = p_2(s_3, s_4)\}$.*

The notion of a contact-space escape point in $\mathcal{U}$ is equivalent to the standard definition of an escape as follows. Upon reaching a contact-space escape point, the two fingers are coincident and can move arbitrarily far from $\mathcal{B}$. Conversely, suppose the hand can move arbitrarily far from $\mathcal{B}$ according to the standard definition of escape. The two fingers can be moved to be coincident, then moved back to the object's boundary, which is an escape point in $\mathcal{U}$. Every contact-space escape point in $\mathcal{U}$ has an equivalent point in $\mathcal{F}$.

A *c-sublevel set* of $d(\mathbf{s})$ in contact space is the set $\mathcal{U}_{\leqslant c} = \{\mathbf{s} \in \mathbb{R}^4 : d(\mathbf{s}) \leqslant c\}$. Similarly, a c-sublevel set of $d(q)$ in the free c-space is the set $\mathcal{F}_{\leqslant c} = \{q \in \mathbb{R}^6 : d(q) \leqslant c\}$.

Definition 3.3.2 *A* **c-sublevel path** *in $\mathcal{F}$ or $\mathcal{U}$ is a path along which the maximum value of the inter-finger distance does not exceed c.*

Thus, an object is caged for inter-finger distance σ^* if and only if there does not exist a σ^*-sublevel path in $\mathcal{F}$ from its current configuration to an escape point.

Let the object $\mathcal{B}$ be initially immobilized at a hand configuration q_0. For a certain interval of finger openings, $[\sigma_0, \sigma_1)$, where $\sigma_0 = d(q_0)$, the object remains caged by the two fingers. The cage is broken at a hand configuration, q_1, where $\sigma_1 = d(q_1) > \sigma_0$. This grasp is referred to as a *join point*, *puncture point*, or *puncture grasp.* Starting at the immobilizing grasp q_0, the value of σ at the *first* puncture point can be viewed as the minimum value, σ_1, such that a σ_1-sublevel path exists between q_0 and either an escape point or another immobilizing grasp. The value of σ at the *last* puncture point can be viewed as the minimum value, σ_{max}, such that a σ_{max}-sublevel path exists between q_0 and an escape point.

3.4 The Caging Graph

Rather than searching the continuous spaces $\mathcal{F}$ or $\mathcal{U}$, we show below that only a discrete set of points must be searched. This set of points form nodes in the *caging graph*, denoted G, which captures the topology of the sublevel sets of $d(\mathbf{s})$ in $\mathcal{U}$. The graph will be the basis for the algorithm described in Section 3.5.

3.4.1 Location of Important Grasps

Rimon and Blake [33] showed that immobilizing and puncture grasps occur only at frictionless equilibrium grasps. For a polyhedron, excluding non-generic parallel geometry, frictionless equilibrium grasps only occur at a finite, discrete set of points, corresponding to finger pairs touching vertices, edges, or faces. The combinations of contact pairs which may result in frictionless equilibrium grasps are:

1. a *vertex-vertex* pair
2. a *vertex-edge* pair, with one finger touching an edge at the point where the perpendicular projection of the vertex onto that edge lies
3. a *vertex-face* pair, with one finger touching a face at the point where the perpendicular projection of the vertex onto that face lies
4. an *edge-edge* pair, in which the fingers lie at points corresponding to the unique minimum distance between those two lines, such that a line connecting the two points is perpendicular to both edges

For cases 2-4, the point may not exist for a specific pair, if the location of either finger does not lie on the interior of its associated edge or face. These points will form the nodes of a *caging graph*, defined below. They are exactly the critical points of the inter-finger distance function, $d(\mathbf{s})$, in $\mathcal{U}$. Note that because contacts both lying on faces of the polyhedron do not form frictionless equilibrium grasps (as parallel faces are excluded), all these points lie on the *boundaries* of contact-space polychora.

Definition 3.4.1 *The* **caging graph**, *G, is an undirected graph with the following nodes and edges:*
Nodes: *The nodes of G correspond to all critical points of the inter-finger distance function in $\mathcal{U}$. In particular, they are all vertex-vertex pairs, and the unique local minimum associated with each vertex-edge, vertex-face, and edge-edge pair, if that minimum exists.*
Edges: *All nodes lying on the boundary of the* same *contact-space polychoron, $\mathcal{P}_{ij}$, are connected to each other by edges in G.*

Because we propose to search the caging grasps, G, rather than free c-space, $\mathcal{F}$, we must justify that the two searches will produce the same result. To that end we introduce the notion of *sublevel set equivalence.*

Definition 3.4.2 *Contact space, $\mathcal{U}$, is* **sublevel equivalent** *with the free c-space, $\mathcal{F}$, if the following holds: for any two points $q, q' \in \mathcal{U}$ there exists a c-sublevel path between q and q' in $\mathcal{U}$ if and only if there exists a c-sublevel path between their equivalent points in $\mathcal{F}$.*

Similarly, G and $\mathcal{U}$ are sublevel equivalent if, for any two nodes, $q, q' \in G$ there exists a c-sublevel path between q and q' in G if and only if there exists a c-sublevel path between their equivalent points in $\mathcal{U}$.

The caging graph, G, and contact space, $\mathcal{U}$ are always sublevel equivalent. The proof is delayed until Sec. 3.9. For many objects, $\mathcal{U}$ and $\mathcal{F}$ are sublevel equivalent. However, for some objects, they are not. For such objects we must perform a rigorous characterization of the critical points of $d(\mathbf{s}) \in \mathcal{U}$, and augment G with a (usually small) number of additional edges. We will proceed assuming that $\mathcal{F}$ and $\mathcal{U}$ are sublevel equivalent, and analyze the case when they are not in Sec. 3.9.

3.5 The Caging Algorithm

The caging algorithm is structured as follows. Starting at an immobilizing grasp of $\mathcal{B}$, the algorithm searches the caging graph, G, using two lists of nodes: a list of open nodes, $\mathcal{O}$, and a list of closed nodes, $\mathcal{X}$. The *open list*, $\mathcal{O}$ is kept sorted in ascending order of the associated value of $d(\mathbf{s})$. To start, the node of G associated with the initial immobilizing grasp is marked as open and inserted into $\mathcal{O}$, while $\mathcal{X}$ is initially empty. At each step, the node in $\mathcal{O}$ with the smallest value of $d(\mathbf{s})$ is marked as *current*, removed from $\mathcal{O}$, and added to the *end* of the closed list $\mathcal{X}$. If the $d(\mathbf{s})$ value of the current node is zero, the algorithm halts, as an escape point has been found. Otherwise, each node adjacent to the current node, if it is not already in $\mathcal{X}$ or $\mathcal{O}$, is added to $\mathcal{O}$.

The algorithm's output is extracted from the list $\mathcal{X}$ as discussed below. In particular, the node in $\mathcal{X}$ having the maximal value of $d(\mathbf{s})$ is the *maximal* puncture grasp associated with the initial immobilizing grasp.

3.5.1 Analysis of the Closed List

Consider the closed list, $\mathcal{X}$, and index each node, $v \in \mathcal{X}$ in the order it was added. Thus, the first node added (the initial immobilizing grasp), is denoted v_0, and so on. Inter-finger distance associated with each node in $\mathcal{X}$, denoted $d(v_i)$, can be seen in Fig. 3.3.

Definition 3.5.1 *The i^{th} node, v_i, in the closed list $\mathcal{X}$ is a* **local minimum** *in $\mathcal{X}$ if $d(v_i) < \min\{d(v_{i-1}), d(v_{i+1})\}$.*

Definition 3.5.2 *The i^{th} node, v_i, in the closed list $\mathcal{X}$ is a* **local maximum** *in $\mathcal{X}$ if $d(v_i) > \max\{d(v_{i-1}), d(v_{i+1})\}$.*

Definition 3.5.3 *The i^{th} node, v_i, in the closed list $\mathcal{X}$ is a* **puncture-related local maximum** *if: 1) it is a local maximum of $\mathcal{X}$, and 2) there does not exist a lower-indexed puncture-related local maximum in $\mathcal{X}$, v_j for $j<i$, such that $d(v_j) > d(v_i)$.*

The interpretation of the closed list is simple. The first element of $\mathcal{X}$, by construction, and all local minima are immobilizing grasps of $\mathcal{B}$. All puncture-related local maxima are puncture grasps of $\mathcal{B}$. While the node representing this puncture is a saddle of the function $d(\mathbf{s})$ in contact space, $\mathcal{U}$, it appears as a local maximum in $\mathcal{X}$. This occurs because once the search algorithm discovers a saddle, it continues searching lower-valued nodes in the unexplored portion of the previously disconnected sublevel set of the caging graph, G. The lowest indexed puncture-related local maximum in $\mathcal{X}$ represents the puncture point associated with the local caging set. The highest indexed puncture-related local maximum in $\mathcal{X}$ represents the maximal puncture point associated with the maximal caging set. Finally, all other puncture-related local maxima represent puncture points associated with all intermediate caging sets.

3.6 An Example

An example demonstrates the algorithm, searching a polyhedron (modified from [8]) with 326 vertices, 648 faces, and 972 edges. Starting at an immobilizing grasp, the algorithm explores contact space, generating an open list and closed list. Finger positions on $\mathcal{B}$ visited during exploration are shown in Figs. 3.1 and 3.2. Inter-finger distances associated with nodes of the closed list, $\mathcal{X}$, including the immobilizing and puncture grasps, and the escape point are shown in Fig. 3.3, plotted against the exploration step of the algorithm. The exploration took 80 steps. Note that only 729 of the 136632 nodes (0.53%) needed to be checked.

3.7 Characterization of Nodes of G

In order to establish the sublevel equivalence of free c-space, $\mathcal{F}$, and contact space, $\mathcal{U}$, we must first characterize all points at which the topology of either space may change, which will only occur at nodes of G. Consider a node of G, $q_0 = (p_1^0, p_2^0)$, and its corresponding physical geometry. The following definitions, illustrated by Fig. 3.4, will help characterize such a node.

Definition 3.7.1 *The **grasp line** for a grasp at $q_0 = (p_1^0, p_2^0)$ is the line segment from p_1^0 to p_2^0 .*

Definition 3.7.2 *The **boundary planes**, denoted $\mathcal{E}_1$, $\mathcal{E}_2$, associated with grasp $q_0 = (p_1^0, p_2^0)$ are two planes perpendicular to the grasp line, passing through p_1^0 and p_2^0, respectively.*

Definition 3.7.3 *The **medial region**, denoted $\mathcal{M}$, associated with grasp $q_0 = (p_1^0, p_2^0)$ is the region lying between $\mathcal{E}_1$ and $\mathcal{E}_2$, not including those planes.*

Definition 3.7.4 *The **lateral region**, denoted $\mathcal{L}$, associated with grasp $q_0 = (p_1^0, p_2^0)$ is the complement of $\mathcal{M}$.*

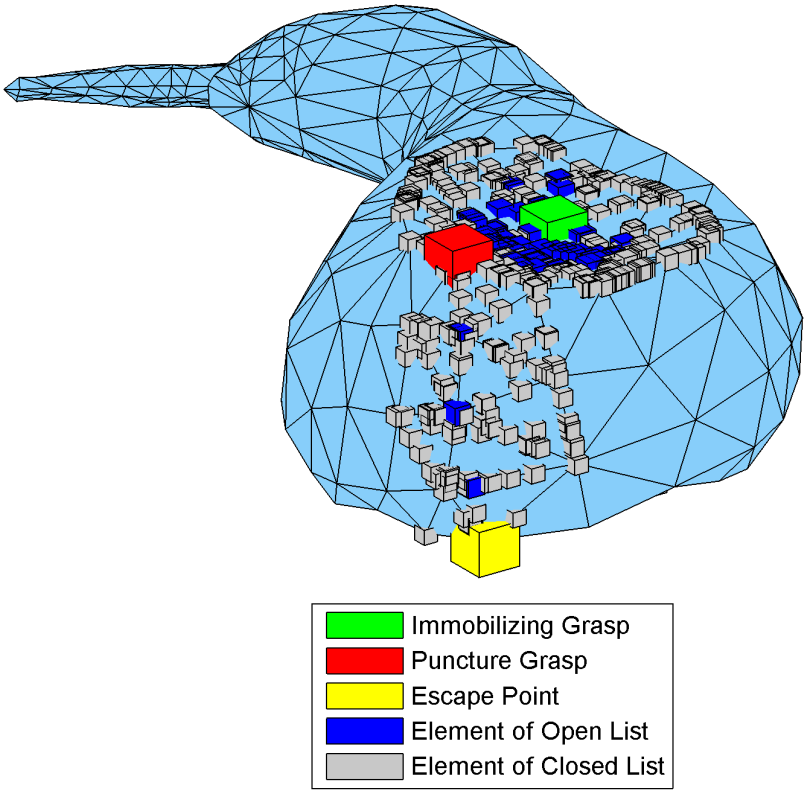


Figure 3.1: Exploration of a polyhedron showing finger positions for explored nodes. Note that nodes of G lie in $\mathbb{R}^6$ and cannot be fully visualized in $\mathbb{R}^2$, so pairs of finger positions are shown.

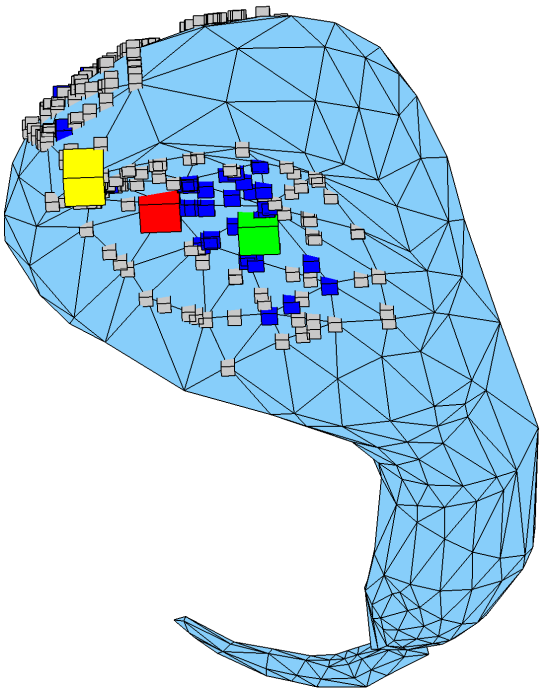


Figure 3.2: A second view of the polyhedron.

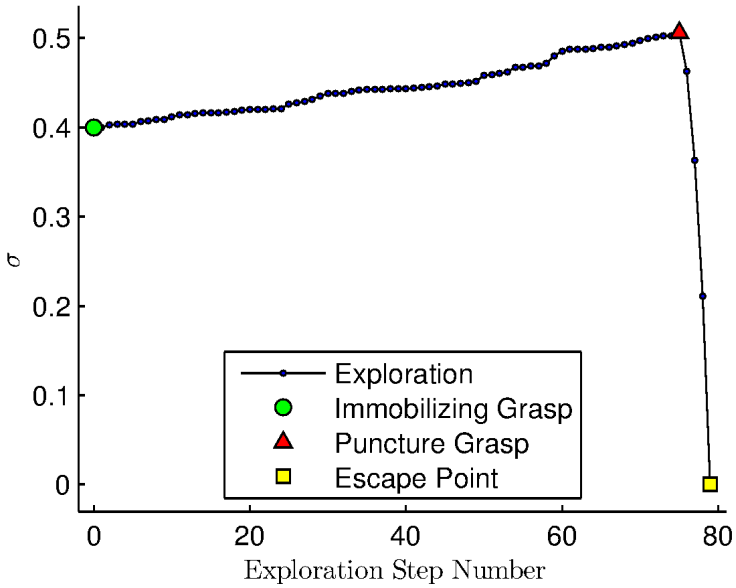


Figure 3.3: Inter-finger distance of nodes in the closed list, $\mathcal{X}$, vs. exploration step of the algorithm.

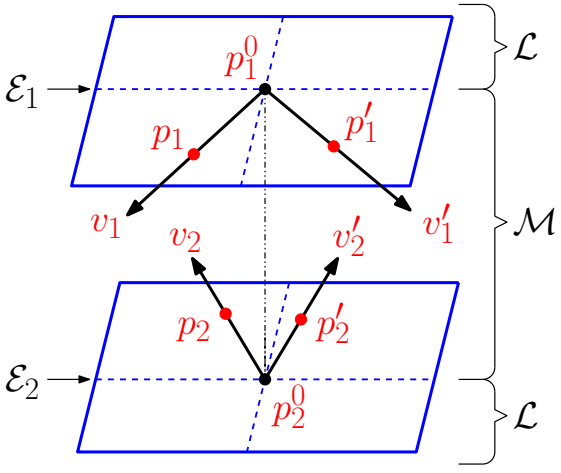


Figure 3.4: Geometry used to characterize point $q_0 = (p_1^0, p_2^0)$. Note that no surfaces of $\mathcal{B}$ are shown.

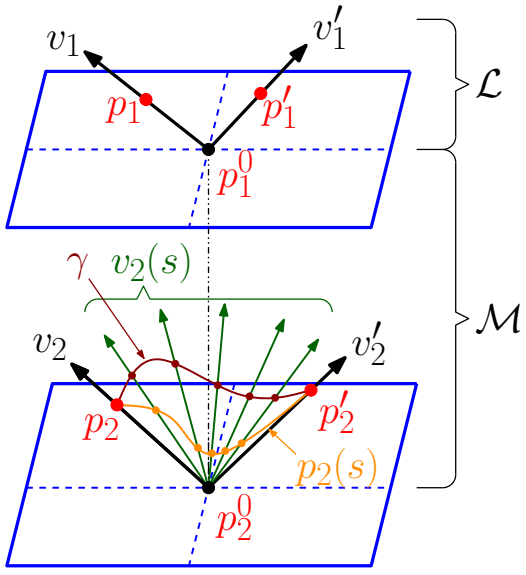


Figure 3.5: Construction of a path showing that if $\mathcal{M}_2$ is connected, then D^- is connected.

Definition 3.7.5 *Let $\mathcal{M}_i$ denote the intersection of the $\mathcal{M}$, free physical space, and a ball of radius δ around point p_i^0.*

Definition 3.7.6 *Let $\mathcal{L}_i$ denote the intersection of $\mathcal{L}$, free physical space, and a ball of radius δ around point p_i^0.*

The following test characterizes any double contact as an immobilizing grasp, puncture grasp, or neither.

Theorem 3.7.7 *At a double-contact point, $q_0 = (p_1^0, p_2^0)$, if $\mathcal{M}_1$ and $\mathcal{M}_2$ are both empty, then q_0 is an immobilizing grasp. If $\mathcal{M}_1$ is empty and $\mathcal{M}_2$ is a path-disconnected set (or $\mathcal{M}_2$ empty and $\mathcal{M}_1$ disconnected), then q_0 is a puncture point. Otherwise, q_0 is neither.*

Theorem 3.7.7 states that double-contact points may be characterized only by the topology of the regions $\mathcal{M}_1$ and $\mathcal{M}_2$, specifically if each region is the empty set, a path-disconnected set, or a non-empty and path-connected set. Hereafter, disconnected will refer to a path-disconnected set, and connected will refer to a non-empty and path-connected set. The following four cases cover all possible combinations:

1. If $\mathcal{M}_1$ and $\mathcal{M}_2$ are both non-empty (either connected or disconnected), then the topology of sublevel sets of $\mathcal{F}$ do not change at q_0.
2. If $\mathcal{M}_1$ and $\mathcal{M}_2$ are both empty, then q_0 is an immobilizing grasp.
3. If $\mathcal{M}_1$ is empty and $\mathcal{M}_2$ is disconnected (or vice versa), then q_0 is a puncture grasp.
4. If $\mathcal{M}_1$ is empty and $\mathcal{M}_2$ is connected (or vice versa), then the topology of sublevel sets of $\mathcal{F}$ does not change at q_0.

The following characterizations of immobilizing and puncture grasps will be used to analyze the above four cases. Consider a point $q_0 = (p_1^0, p_2^0)$, with $d(q_0) = d_0$. Let D^0 be the slice of $\mathcal{F}$ restricted to $d(q) = d_0$, and let D^- be the slice of $\mathcal{F}$ restricted to $d(q) = d_0 - \epsilon$.

An **immobilizing grasp** at q_0 is characterized locally around q_0 by two criteria:

1. D^- is empty
2. D^0 is an isolated point

Similarly, a **puncture grasp** is characterized by two criteria:

1. D^- is disconnected
2. D^0 is connected

To analyze D^- and D^0, consider the following parameterization of a region *in physical space*, in a neighborhood of $q_0 = (p_1^0, p_2^0)$. Let v_i be the vector originating from p_i^0 passing through point p_i, and r_i be the distance from p_i^0 to p_i. Thus, $q = (p_1^0 + r_1 v_1, p_2^0 + r_2 v_2)$. The inter-finger distance associated with this point is given by

$$d(q)^2 = d_0^2 + d_0(r_1 v_1^z - r_2 v_2^z) + \mathcal{O}(\mathbf{r}^2), \tag{3.3}$$

where $d_0 = d(q_0)$ and v_i^z is the z-component of v_i. Note that r_i must be non-negative.

The following geometric insight will also aid in analyzing the above cases. Consider a point p_1^0 on the boundary of $\mathcal{B}$ (which may lie on an edge or at a vertex) and a neighborhood of p_1^0 small enough that no *other* edges or vertices lie in this neighborhood. In this neighborhood, the boundary of $\mathcal{B}$ is formed by one or more planes meeting at p_1^0. Any intersections of those planes also meet at p_1^0. Consider a point, p_1 near p_1^0, and a vector v_1 originating from p_1^0 and passing through p_1, as in Fig. 3.4. If p_1 lies on the boundary of $\mathcal{B}$, then all points $p_1^0 + \alpha v_1$ for $\alpha \in (0, \delta)$ lie on the boundary $\mathcal{B}$ in some δ-neighborhood around p_1^0. Similarly, if p_1 lies outside of $\mathcal{B}$, then all points $p_1^0 + \alpha v_1$ also lie outside of $\mathcal{B}$.

For **case 1**, if both $\mathcal{M}_1$ and $\mathcal{M}_2$ are non-empty, then D^- is non-empty and connected. This will be shown by constructing a path between any two points, $q = (p_1, p_2) \in D^-$ and $q' = (p_1', p_2') \in D^-$. Let each point lie along a correspondingly named vector as in Fig. 3.4.

Points in D^- satisfy the following equation:

$$d(q)^2 = d_0^2 + d_0(r_1 v_1^z - r_2 v_2^z) + \mathcal{O}(\mathbf{r}^2) = d_0^2 - \epsilon + \epsilon^2. \tag{3.4}$$

This equation has a solution (near q_0 and for small ϵ) if either v_1^z is negative (i.e., $v_1^z \in \mathcal{M}_1$) or v_2^z is positive (i.e., $v_2^z \in \mathcal{M}_2$).

Start by assuming that both p_1 and p_1' lie in $\mathcal{M}_1$ and p_2 and p_2' lie in $\mathcal{M}_2$. To construct a path, denoted Γ, we start at (p_1, p_2) and move finger one, f_1, along v_1 to p_1^0. Eq. (3.4) can be solved for r_2, which is approximately linear in r_1, so f_2 may be moved along v_2 to maintain an inter-finger distance of $d_0 - \epsilon$ during this motion. Move f_2 along v_2 to p_2^0, while moving f_1 along v_1' to maintain the inter-finger distance. Finally, move f_2 along v_2' to p_2', while moving f_1 along v_1'. Finger one will arrive at p_1'.

Next, consider a point $q = (p_1, p_2)$ with $p_1 \in \mathcal{M}_1$ and $p_2 \in \mathcal{L}_2$. Move f_2 along v_2 to p_2^0 while moving f_1 along v_1. Denote this path β; its endpoint lies on path Γ. Any two points in D^- may be connected with combinations of paths β and Γ. Thus, if both $\mathcal{M}_1$ and $\mathcal{M}_2$ are non-empty, then D^- is connected and the sublevel sets of $\mathcal{F}$ do not change topologically at q_0.

For the remaining cases, either $\mathcal{M}_1$ or $\mathcal{M}_2$ is non-empty. Without loss of generality, assume that $\mathcal{M}_1$ is empty. Note that if $\mathcal{M}_1$ is empty, then $\mathcal{L}_1$ must be non-empty.

For **case 2**, if both $\mathcal{M}_1$ and $\mathcal{M}_2$ are empty, then D^- is empty, as (3.4) has no solutions. The fact that q_0 is an isolated point can be seen as follows: the only configurations containing $q = (p_1, p_2)$ that give $d(q) = d_0$ involve p_1 lying on $\mathcal{E}_1$, and p_2 lying on $\mathcal{E}_2$, exactly at the perpendicular projection of p_1 onto $\mathcal{E}_2$. Since non-generic parallel geometry has been excluded, this can only happen when $q = q_0$. Thus, q_0 is an isolated point.

For **case 3**, if $\mathcal{M}_2$ is disconnected, then D^- is disconnected and q_0 is a puncture grasp. This may be shown by contradiction. Assume D^- is connected; then there exists a path in D^- from $q = (p_1, p_2)$ to $q' = (p_1', p_2')$. Since $\mathcal{M}_2$ is disconnected, the path of f_2 must pass through $\mathcal{L}_2$. But $p_1 \in \mathcal{L}_1$ and $p_2 \in \mathcal{L}_2$ always results in $d(q) \geqslant d_0$, and cannot lie in D^-.

For **case 4**, if $\mathcal{M}_2$ is connected, then D^- is connected and q_0 is not a puncture point or immobilizing grasp. To show this we will construct a path in D^- between arbitrary points $q = (p_1, p_2) \in D^-$ and $q' = (p_1', p_2') \in D^-$. Choose r_1 and r_1' such that $p_1 = p_1^0 + r_1 v_1$ and $p_1' = p_1^0 + r_1' v_1'$. The path of finger one follows $p_1(s)$, as defined below:

$$p_1(s) = \begin{cases} p_1^0 + r_1(s) v_1 & : s \in [0, \frac{1}{2}] \\ p_1^0 + r_1(s) v_1' & : s \in (\frac{1}{2}, 1] \end{cases}, \tag{3.5}$$

where $r_1(s)$ is chosen such that it varies from r_1 to zero over $s \in [0, \frac{1}{2}]$ and from zero to r_1' over $s \in [\frac{1}{2}, 1]$. Since $\mathcal{M}_2$ is connected, then there exists a path γ from p_2 to p_2', lying in $\mathcal{M}_2$, as shown in Fig. 3.5. Parameterize this path such that $\gamma(0) = p_2$ and $\gamma(1) = p_2'$. Consider a vector, $v_2(s)$, originating from p_2^0 that, for any s, passes through $\gamma(s)$. Since $\gamma(s)$ lies outside of $\mathcal{B}$, then, by the geometric arguments above, all points $p_2^0 + r_2(s) v_2(s)$ lie outside of $\mathcal{B}$ for $r_2(s) \in (0, \delta)$. Solving (3.4) gives

$$r_2(s) = \frac{1}{v_2^z(s)} (r_1(s) v_1^z + 2\epsilon - \frac{1}{d_0} (\epsilon^2 - \mathcal{O}(\mathbf{r}^2)), \tag{3.6}$$

where v_1^z is the z-component of either v_1 or v_1'. Since both $r_1(s)$ and $v_2(s)$ are continuous, and $v_1^z, v_2^z > 0$, there exists a continuous positive solution for $r_2(s)$, for $s \in [0, 1]$. Let $p_2(s) = p_2^0 + r_2(s) v_2(s)$. The path $q(s) = (p_1(s), p_2(s))$ connects q to q' and lies entirely in D^-. Thus, if $\mathcal{M}_2$ is connected then q_0 is not a puncture point or immobilizing grasp.

3.8 A Catalog of Immobilizing and Puncture Grasps

The above characterization gives a relatively simple way to categorize a double-contact point as an immobilizing grasp, puncture grasp, or neither. This section translates the above conditions into simple geometric tests that apply to most grasps. Section 3.9 will analyze problem cases.

Definition 3.8.1 *A double-contact configuration, $q_0 = (p_1^0, p_2^0)$, is* **regular** *if the following condition is met. Consider a point $q = (p_1, p_2)$ near q_0, such that a line segment joining p_1 and p_2 lies parallel*

to the grasp line. All such line segments must lie entirely within the object, $\mathcal{B}$, locally near p_1 and p_2.

Intuitively, *regular* grasps are those containing material between the finger contacts. The following geometric test may be used to evaluate a *regular* double contact $q_0 = (p_1^0, p_2^0)$. Consider the surface of $\mathcal{B}$ near p_1^0. Consider the inter-finger distance of points near p_1^0 while keeping f_2 fixed at p_2^0, giving $d(p_1) = ||p_1 - p_2^0||$. Using this distance function, p_1 may be characterized as a local minima, a saddle, a local maxima, or none of these.

The contact pair $q_0 = (p_1^0, p_2^0)$ is an immobilizing grasp when $\mathcal{M}_1$ and $\mathcal{M}_2$ are both empty. This occurs when both contact points, evaluated with the other contact point fixed, are local minima. Thus, possible *regular* immobilizing grasps are shown in Fig. 3.6.

The contact pair $q_0 = (p_1^0, p_2^0)$ is a puncture grasp when $\mathcal{M}_1$ is empty and $\mathcal{M}_2$ is disconnected (or vice versa). This occurs when one contact point is a local minima and the other is a saddle, each characterized with the other point fixed. Possible *regular* puncture grasps are show in Fig. 3.7. Note that the saddle nature of contacts are drawn such that they divide a sublevel set into two disconnected regions: generically, the surface of a polyhedron may form saddles which divide sublevel sets into more than two connected regions, as in a monkey saddle.

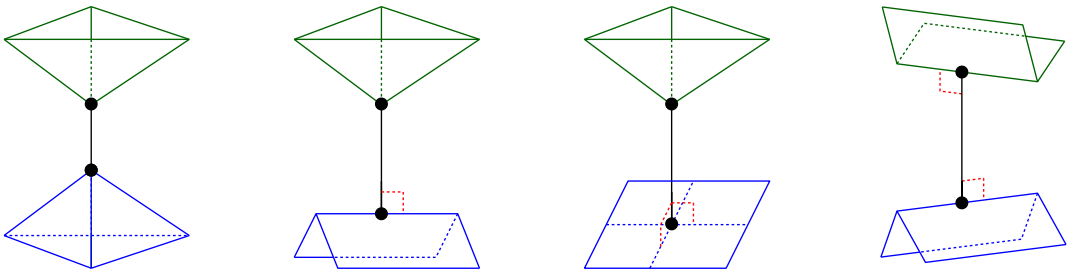


Figure 3.6: A catalog of possible immobilizing grasps. Each contact represents a local minimum in the inter-finger distance when the opposing finger is fixed.

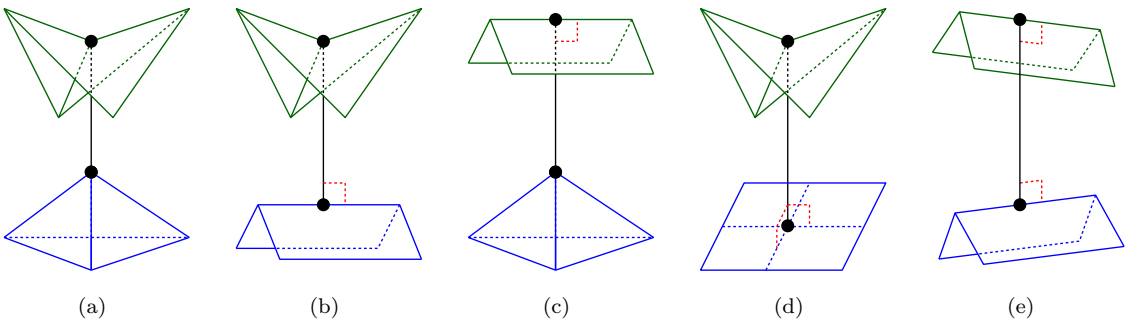


Figure 3.7: A catalog of possible puncture grasps. Each contact represents a saddle in the inter-finger distance when the opposing finger is fixed.

3.9 Sublevel Equivalence

The above algorithm assumed sublevel equivalence of free c-space, $\mathcal{F}$, and contact space, $\mathcal{U}$, as well as that of $\mathcal{U}$ and the caging graph, G. This section first shows the sublevel equivalence of $\mathcal{U}$ and G, using the convexity of the inter-finger distance function. It then considers the case when sublevel equivalence of $\mathcal{U}$ and $\mathcal{F}$ fails, and shows that simple changes can restore sublevel equivalence of $\mathcal{U}$ and $\mathcal{F}$.

3.9.1 Sublevel Equivalence of $\mathcal{U}$ and G

The caging graph depends on the following decomposition of $\mathcal{U}$ having some useful properties.

Lemma 3.9.1 *The function $d(\boldsymbol{s}) = ||p(s_1, s_2) - p(s_3, s_4)||$ is a smooth **convex function** in each contact-space polychoron.*

Proof Each contact-space polychoron, $\mathcal{P}_{ij}$, represents the placement of the fingers on two particular faces of $\mathcal{B}$. Consider the infinite planes underlying these faces. All non-parallel planes meet along a common line. Parameterize the position of a point on each plane as in Fig. 3.8, yielding $p_1 = (s_1, s_2, 0)$ and $p_2 = (s_3 \cos\theta, s_4, s_3 \sin\theta)$. The inter-finger distance function can be written in these coordinates as $d(\mathbf{s}) = \left(\mathbf{s}^T K \mathbf{s}\right)^{\frac{1}{2}}$, where

$$K = \begin{bmatrix} 1 & 0 & -\cos\theta & 0 \\ 0 & 1 & 0 & -1 \\ -\cos\theta & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}.$$

The eigenvalues of K are $\{0, 2, 1 \pm \cos\theta\}$, which are all non-negative. Thus K is positive semi-definite and $d(\mathbf{s}) = ||K^{1/2}\mathbf{s}||$. The function $d(\mathbf{s})$ is thus a composition of the Euclidean norm (a convex function) with the linear function $K^{1/2}\mathbf{s}$. Such a composition preserves convexity, and $d(\mathbf{s})$ is therefore convex in each polychoron, $\mathcal{P}_{ij}$. □

The next corollary follows from Lemma 3.9.1 and the definition of convexity.

Corollary 3.9.2 *Let $\boldsymbol{s}$ and $\boldsymbol{s}'$ be two points on the boundary of a single contact-space polychoron, $\mathcal{P}_{ij}$. The straight line path between these two points lies in a single connected component of the c-sublevel set $\mathcal{U}_{\leqslant c}$ where $c = \max\{d(\boldsymbol{s}), d(\boldsymbol{s}')\}$.*

Definition 3.9.3 *A* **c-sublevel set** *of the caging graph G is the set of nodes, v, given by $G_{\leqslant c} = \{v(\boldsymbol{s}) \in G : d(\boldsymbol{s})) \leqslant c\}$.*

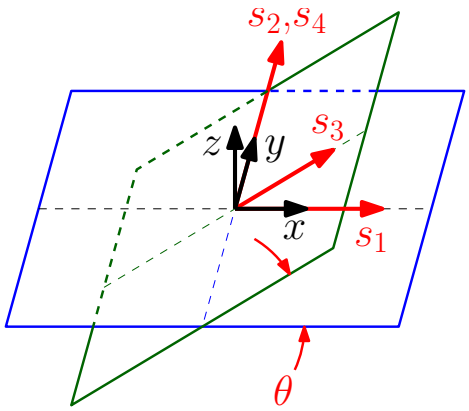


Figure 3.8: A parameterization of two infinite planes, which demonstrates the convexity of $d(\mathbf{s})$.

The following theorem asserts that G preserves the sublevel connectivity of contact space, $\mathcal{U}$.

Theorem 3.9.4 *The caging graph G is* **sublevel equivalent** *to contact space, $\mathcal{U}$. That is, there exists a path in $\mathcal{U}$ between two nodes $v_i, v_j \in G$, lying entirely in $\mathcal{U}_{\leqslant c}$, if and only if there exists a path along the edges of G between v_i and v_j lying entirely in $G_{\leqslant c}$.*

Proof For the forward direction, start with a path between two nodes, $v_i, v_j \in \mathcal{U}$, lying entirely in $\mathcal{U}_{\leqslant c}$. Divide this path into segments such that each segment lies in one contact-space polychoron. Next, replace each path segment with a straight line path between the segment's endpoints. By the convexity of $d(\mathbf{s})$ on the individual contact-space polychoron, the maximum value of $d(\mathbf{s})$ on each straight line segment is upper bounded by the value of $d(\mathbf{s})$ at its endpoints. The path connecting v_i and v_j is now piecewise linear. On each linear segment of this path, the value of $d(\mathbf{s})$ is still upper bounded by c according to Corollary 3.9.2. The two endpoints of each linear segment lie on the boundary of some polychoron $\mathcal{P}_{ij}$. Shift each endpoint to the minimum point of $d(\mathbf{s})$ along the same boundary edge of $\mathcal{P}_{ij}$, which is always a node of G. This local shifting can only *decrease* the value of $d(\mathbf{s})$ for that endpoint. The equivalent path in G thus lies entirely in the discrete sublevel set $G_{\leqslant c}$.

For the backwards direction, start with a discrete path between the nodes v_i and v_j lying entirely in $G_{\leqslant c}$. For each pair of adjacent nodes along the path, connect the two nodes by a straight line segment in $\mathcal{U}$. Each straight line segment lies in one polychoron, $\mathcal{P}_{ij}$. The maximum value of $d(\mathbf{s})$ along each segment is upper bounded by the value of $d(\mathbf{s})$ at either endpoint, per Corollary 3.9.2. The entire piecewise linear path thus lies in $\mathcal{U}_{\leqslant c}$. □

The following corollary is a direct result of Theorem 3.9.4.

Corollary 3.9.5 *The caging graph G has the two properties:*

1. *The critical points of $d(\boldsymbol{s})$ in each connected component of $\mathcal{U}_{\leqslant c}$ correspond to a connected subgraph of G containing only the critical points lying in that component.*

2. *Every pair of sublevel sets of $d(\boldsymbol{s})$ in $\mathcal{U}$ that meet at a puncture point, $v \in \mathcal{U}$, correspond to two subgraphs of G that meet at the corresponding node, $v \in G$.*

3.9.2 The Sublevel Equivalence of $\mathcal{U}$ and $\mathcal{F}$

To ensure sublevel set equivalence of $\mathcal{U}$ and $\mathcal{F}$, we must consider the topological changes which occur in each. Above we analyzed the topological changes at point $q_0 = (p_1^0, p_2^0)$ in free c-space, $\mathcal{F}$, based on the topology of two regions of $\mathcal{F}$, $\mathcal{M}_1$ and $\mathcal{M}_2$. Consider similar regions in $\mathcal{U}$, namely the intersection of the *boundary* of the polyhedron and the medial region, $\mathcal{M}$, around each of the finger placements. Denote these regions $\mathcal{M}_1'$ and $\mathcal{M}_2'$. Note that $\mathcal{M}_i'$ is just the restriction of $\mathcal{M}_i$ to the boundary of $\mathcal{B}$.

The previous determination of the topology of $\mathcal{M}_i$ around contact points only assumes that if $p_1 = p_1^0 + r_1 v_1$ lies outside (or on the boundary) of $\mathcal{B}$, then $p_1^0 + \alpha v_1$ also lies outside (or on the boundary) of $\mathcal{B}$, for $\alpha \in (0, \delta)$. This assumption is true when restricting the fingers to the boundary of $\mathcal{B}$, so the resulting tests may be used to characterize $\mathcal{U}$ as well. Thus, if $\mathcal{M}_1'$ and $\mathcal{M}_2'$ are both empty, then at grasp q_0, an isolated point of $\mathcal{U}$ appears, which did not exist in $\mathcal{U}_{\leqslant d_0 - \epsilon}$, where $d_0 = d(q_0)$. This point would be an immobilizing grasp if finger motions were restricted to $\mathcal{U}$. If $\mathcal{M}_1'$ is empty and $\mathcal{M}_2'$ is disconnected, then at q_0, two (or more) regions of $\mathcal{U}$ which were disconnected in $\mathcal{U}_{\leqslant d_0 - \epsilon}$ join to form a connected component in $\mathcal{U}_{\leqslant d_0}$. This point would be a puncture grasp if finger motions were restricted to $\mathcal{U}$. If neither condition holds, then the topology of sublevel sets of $\mathcal{U}$ do not change at q_0.

We must now consider when the topology of sublevel sets of $\mathcal{U}$ and $\mathcal{F}$ change, and when those changes differ. Each region, $\mathcal{M}_i$ and $\mathcal{M}_i'$ may be either empty, connected, or disconnected. It can be shown that if $\mathcal{M}_i$ is empty, then $\mathcal{M}_i'$ is also empty, and that if $\mathcal{M}_i$ is disconnected, then $\mathcal{M}_i'$ is also disconnected. Thus, there are only two cases in which the topology of $\mathcal{M}_i$ and $\mathcal{M}_i'$ may differ: 1) $\mathcal{M}_i$ may be connected while $\mathcal{M}_i'$ is disconnected, or 2) $\mathcal{M}_i$ may be connected while $\mathcal{M}_i'$ is empty.

Adding the three cases where the topology of $\mathcal{M}_i$ and $\mathcal{M}_i'$ are the same (both empty, both connected, or both disconnected), there are five possibilities for the topology of the $\mathcal{M}_i$-$\mathcal{M}_i'$ pairs, resulting in 25 combinations when considering both contacts. Out of those 25 combinations, there are nine cases in which the topological changes of $\mathcal{F}$ and $\mathcal{U}$ differ, but these nine cases only take two forms, which we refer to as *false immobilizing grasps* and *false puncture grasps*. Both of these cases, and their effect on the sublevel equivalence of $\mathcal{U}$ and $\mathcal{F}$ will be considered in the following subsections.

Note that for a *regular* grasp at $q_0 = (p_1^0, p_2^0)$, each connected component $\mathcal{M}_i'$ is associated with exactly one connected component of $\mathcal{M}_i$, specifically the portion of free space located between that component of $\mathcal{M}_i$ and the boundary plane, $\mathcal{E}_i$. Thus, for a *regular* grasp, the topology change in $\mathcal{U}$ is the same as for $\mathcal{F}$ at every point.

3.9.3 Sublevel Equivalence at False Immobilizing Grasps

In a false immobilizing grasp, $\mathcal{M}_1'$ and $\mathcal{M}_2'$ are both empty, while either $\mathcal{M}_1$ or $\mathcal{M}_2$ is connected (or both). Thus, at q_0, an isolated point appears in $\mathcal{U}$, while no topological change occurs in $\mathcal{F}$. In $\mathcal{U}$, this appears to be an immobilizing grasp. The examples in Fig. 3.6 are false immobilizing grasps if the material lies above and below the contact points, rather than between them.

Consider the changes of topology in both $\mathcal{U}$ and $\mathcal{F}$ around a false immobilizing grasp at q_0. In $\mathcal{U}$, the sublevel set $\mathcal{U}_{\leqslant d_0-\epsilon}$ is empty in a neighborhood of q_0. The sublevel set $\mathcal{U}_{\leqslant d_0}$ is the single point q_0, and $\mathcal{U}_{\leqslant d_0+\epsilon}$ is a small region around q_0. This cavity is isolated from other points in $\mathcal{U}$. There is no change in the topology of sublevel sets of $\mathcal{F}$ near q_0, and $\mathcal{U}$ and $\mathcal{F}$ are *not* sublevel equivalent. An isolated point appears in $\mathcal{U}$, but this point must lie in a connected component of $\mathcal{F}$ containing other double-contact points. The tunnel curve construction found below is essentially a constructive proof of this fact, and restores sublevel equivalence.

3.9.4 Sublevel Equivalence at False Puncture Grasps

In a false puncture grasp, $\mathcal{M}_1'$ is empty and $\mathcal{M}_2'$ is disconnected (or vice versa), while either $\mathcal{M}_1$ or $\mathcal{M}_2$ is connected. Thus, at q_0, two disconnected regions in $\mathcal{U}$ join, while no topological change occurs in $\mathcal{F}$. In $\mathcal{U}$, this appears to be a puncture grasp. The examples in Fig. 3.7 are false puncture grasps if the material lies above and below the contact points, rather than between them.

At a false puncture grasp, the changes in topology of $\mathcal{F}_{\leqslant c}$ and $\mathcal{U}_{\leqslant c}$ are qualitatively different, but sublevel equivalence still holds. At these types of grasp, two disconnected regions in $\mathcal{U}$ join at q_0, while the same regions are locally connected in $\mathcal{F}_{\leqslant d_0-\epsilon}$. However, if $\mathcal{U}$ and $\mathcal{F}$ are sublevel equivalent for $d(\mathbf{s}) = d_0 - \epsilon$, then the two regions must already be connected in $\mathcal{U}_{\leqslant d_0-\epsilon}$. As long as sublevel equivalence is not violated for inter-finger distances smaller than $d_0 - \epsilon$, the changes in topology at q_0 will not violate sublevel equivalence.

3.9.5 Tunnel Curve Construction

To maintain sublevel equivalence between $\mathcal{U}$ and $\mathcal{F}$ at a false immobilizing grasp, we will add a *tunnel curve*, which is a path lying in $\mathcal{F}$ with endpoints in $\mathcal{U}$, to both $\mathcal{U}$ and G. This additional connectivity will guarantee that $\mathcal{F}$, $\mathcal{U}$, and G remain sublevel equivalent.

At a false immobilizing grasp, at least one finger can move away from object $\mathcal{B}$ in a straight line towards the other finger. Retract this finger while holding the other finger fixed on $\mathcal{B}$'s boundary, until the finger hits a new surface of $\mathcal{B}$. Both fingers now contact the object along its boundary. Slide both fingers simultaneously along the body while minimizing σ (i.e., squeeze the fingers), until reaching the unique minimum of the inter-finger distance on these surfaces. This defines the tunnel-curve's endpoint. If, during the closing process, the fingers meet, the tunnel curve's endpoint is any

escape point on the current contact-space polychoron. The start and end point of the tunnel curve are nodes of the caging graph, G. For each tunnel curve, add an edge to G connecting the nodes representing its endpoints.

The addition of these curves will restore the sublevel equivalence of $\mathcal{F}$, $\mathcal{U}$, and G, which may be seen as follows. The only points at which the topology of sublevel sets in $\mathcal{F}$ and $\mathcal{U}$ become different is at a false immobilizing grasp, when an isolated point appears in $\mathcal{U}_{\leqslant d_0}$, which is not isolated in $\mathcal{F}_{\leqslant d_0}$. Once a tunnel has been added to this point in $\mathcal{U}$ (and G), it is not isolated in $\mathcal{U}_{\leqslant d_0}$ (or G) when it appears.

3.10 Summary

This chapter presented a new method to calculate the caging sets for a polyhedron by searching contact space rather than free space. The algorithm is based on two innovations. First, all critical points of the inter-finger distance function were identified and characterized, resulting in a catalog of all immobilizing and puncture grasps. The critical points are nodes in a caging graph. Second, the relationships between 6D configuration space, 4D contact space, and the caging graph were analyzed, allowing a search of the caging graph. It was shown that contact space can be completely decomposed into 4D polychora, on which the inter-finger distance is convex, allowing the relationship between contact space and the caging graph to be fully characterized. Additionally, the points at which the topology of sublevel sets of contact space and free configuration space differed were identified. The addition of edges to the caging graph reconciles these differences. An algorithm which searches the caging graph for puncture points of successively increasing inter-finger distance was presented, and an example demonstrated the implementation of the algorithm.

Chapter 4

Three Finger Caging

This chapter considers caging a convex planar polygonal object using three point fingers. In particular, it quantifies exactly the allowable motions of the fingers relative to each other, such that at no point during those motions the object can escape. This is in contrast to most other three finger caging algorithms, which make no guarantees of caging if the fingers are allowed to vary their relative positions while the object is allowed to move. This algorithm is developed mostly independently from the concepts used in the preceding two finger algorithms.

Pipattanasomporn et al [29] gave an algorithm which maps the relative positions of n fingers to a scalar parameter, which they refer to as *dispersion*. A polytope is caged by maintaining the value of this function below (or above) some critical value. They give an algorithm to compute such cages (which are a subset of all caging formations) for some specific scalar functions. Rodriguez et al [38] formalized these ideas and extended them to a more general class of objects, but do not provide an explicit algorithm. They also considered the relationship between such cages and immobilizing grasps.

For two disc fingers, Vahedi and van der Stappen [43] showed that such robustness comes "for free" – if an object is caged for a particular inter-finger distance of $\sigma = \sigma^*$, then it is caged for either $\sigma \in (0, \sigma^*]$ or $\sigma \in [\sigma*, \infty)$, which they refer to as *squeezing* or *stretching* cages, respectively. They also showed that this robustness property is not guaranteed for more than two fingers.

The work of Pipattanasomporn et al [29] and Rodriguez et al [38] described above provide a subset of all n-fingered cages. They provide an analogy to *squeezing* (or *stretching*) caging for more than two fingers: if a scalar mapping on the finger formation is maintained below (or above) some critical value, then the object will remain caged.

The three finger algorithms of Erickson, et al [16] and Vahedi and van der Stappen [43] divide free space into cells. Placement of the third finger into a subset of these cells provides a cage. While neither author explicitly considers robustness, it appears that these cages will be robust to movement of the third finger within a single cell (relative to the other two fingers), but not necessarily to movement of the third finger into other cells of the caging region, nor to changes in the distance

between the first two fingers.

Perhaps the work most similar to this chapter is Sudsang et al [40–42], which considered caging a polygon with three disc fingers, and found the largest discs of equal radius in which free movement of the fingers guarantee that the object remains caged. The triple of discs is a useful subset of the entire caging region. However, these works considered only the subset of cages in which each finger touches only a single edge of the polygon, resulting in very small caging regions for objects with short sides. Additionally, the regions given are conservative for some cases, resulting in discs which are not of maximal size.

This work considers all three parameters which describe the relative position of three point fingers in the plane. It provides an exact characterization of the robust caging regions (defined later) produced by three-fingered placements, but at the cost of significant computational complexity.

4.1 Preliminaries and Robust Caging

A convex planar polygonal object, $\mathcal{B}$, is to be caged by three point fingers. The n edges of $\mathcal{B}$ are labeled e_i, and the n vertices are denoted by v_i, which also describes their position in the plane. Its convexity guarantees that $\mathcal{B}$ has no holes in its interior. We refer to the three fingers as f_1, f_2, and f_3, and also use $f_i \in \mathbb{R}^2$ to refer to the position of the i^{th} finger in the plane. We define a *grasp* as the placement of the triple of finger placements, f_1, f_2, and f_3, in the plane. Note that this definition of a grasp does not require any of the fingers to touch $\mathcal{B}$. Following Vahedi [43] we refer to any two grasps related by a rigid transformation as having the same hand *shape*. Consequently, the position of the fingers relative to $\mathcal{B}$ may be decomposed into the hand shape, denoted σ, and the position of that shape relative to $\mathcal{B}$. The hand shape is parameterized by the pairwise inter-finger distances, $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, where $\sigma_1 = ||f_2 - f_3||$ and so on. See Fig. 4.1.

Definition 4.1.1 *The **shape space** (c-space) of the three-fingered hand, denoted $\mathcal{S} = \mathbb{R}^3_{\geqslant 0}$, is the pairwise inter-finger distances, σ_1, σ_2, and σ_3. We require the fingers are numbered in counter-clockwise order to guarantee uniqueness of configuration given the three inter-finger distances.*

The position of a fixed hand shape relative to $\mathcal{B}$ are given by $q = (x, y, \theta) \in \mathbb{R}^2 \times \mathbb{S}$ where x and y are the position of f_1, while θ represents the angle between the x-axis and a line drawn between f_1 and f_2.[1]

Definition 4.1.2 *The **configuration space** (c-space) of the three-fingered hand is the six dimensional space $\boldsymbol{\rho} = (\boldsymbol{q}, \boldsymbol{\sigma}) \in \mathbb{R}^2 \times \mathbb{S} \times \mathcal{S}$, where $\boldsymbol{q} \in \mathbb{R}^2 \times \mathbb{S}$ is the hand's base configuration, and $\boldsymbol{\sigma} \in \mathcal{S}$ is the hand shape. See Fig. 4.1.*

[1] Since a convex object cannot be caged with two of the three fingers coincident, the singularity that occurs when f_1 and f_2 are coincident may be ignored.

For a given configuration, $\boldsymbol{\rho} = (\mathbf{q}, \boldsymbol{\sigma})$, the object is caged if it cannot be moved arbitrarily far from the fingers without penetrating any of the fingers. Equivalently, the object is caged if the fingers cannot be moved arbitrarily far from it while holding the shape parameters constant.

This can be formalized as follows, using notation from Vahedi [43]. Let $F = \mathbb{R}^2 \backslash \mathbf{int}(\mathcal{B})$ be the set of all *admissible* placements for one finger in the plane, and let $\mathcal{F} = F \times F \times F$ be the free configuration space (c-space) of the three-fingered hand. Let $\sigma(\boldsymbol{\rho}) = \delta = (\sigma_1, \sigma_2, \sigma_3)$ be the shape parameter of a given grasp, $\boldsymbol{\rho}$. Then $\mathcal{F}_\delta = \{\rho \in \mathcal{F} : \sigma(\rho) = \delta\}$ is the restriction of free space to a particular hand shape, δ. A grasp, $\boldsymbol{\rho}$, is a cage if it belongs to a bounded component of $\mathcal{F}_\delta$, where $\delta = \sigma(\rho)$.

4.1.1 Motivational Example

This chapter focuses on cages which are *robust* to variations in the hand shape, σ, which motivates the following example. Consider a specific grasp $\rho_a = (q_a, \sigma_a)$, shown in red in Fig. 4.2; this is a cage of $\mathcal{B}$. Now consider starting at grasp ρ_a and allowing σ to vary. If σ varies in some small neighborhood of σ_a in shape space (while q is allowed to vary freely), the object will remain caged.

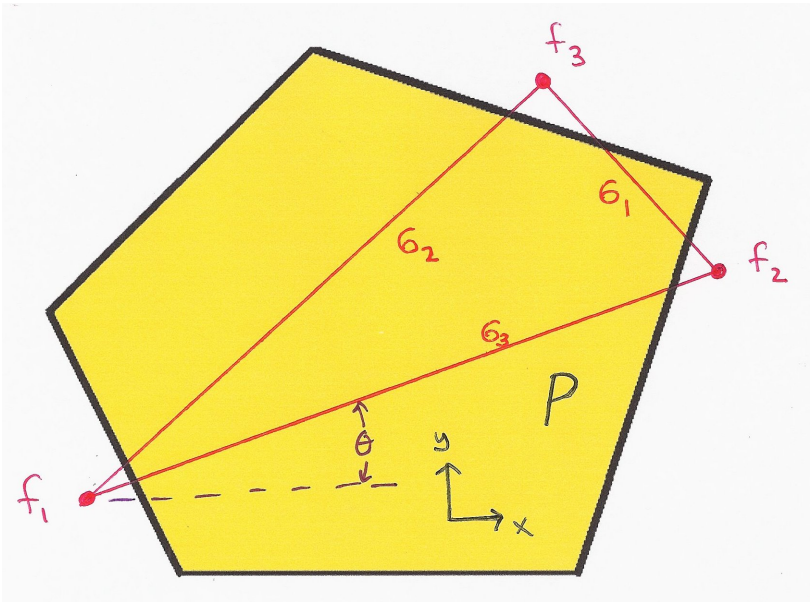


Figure 4.1: Parameterization of the hand shape for a three point-fingered hand. The pairwise inter-finger distances are given by σ_1, σ_2, and σ_3.

However, consider grasp $\rho_b = (q_b, \sigma_b)$, shown in green. Clearly the fingers may be moved rightward until they are arbitrarily far from $\mathcal{B}$, and grasp ρ_b is not a cage of $\mathcal{B}$. Now consider starting at ρ_a, but allow the three hand-shape parameters to vary from σ_a to σ_b. When σ reaches σ_b (or before), the hand will be able to escape. Thus, allowing the hand shape to vary from σ_a to σ_b does not provide a cage of $\mathcal{B}$.

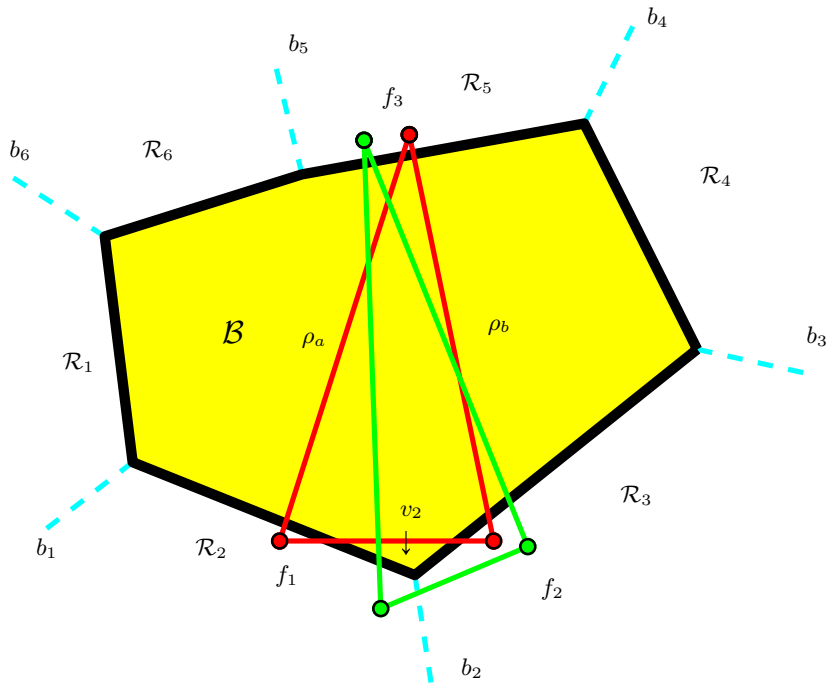


Figure 4.2: A polygon, $\mathcal{B}$, and two grasps: ρ_a cages $\mathcal{B}$; ρ_b does not. Also shown is a decomposition of free space around a convex polygon, $\mathcal{B}$, dividing it into regions, $\mathcal{R}_i$.

Intuitively, grasp ρ_b is able to escape rightward because f_1 is able to move "past" vertex v_2. This motivates the following decomposition of free space around $\mathcal{B}$. From each vertex, v_i, draw a *bounding line*, denoted b_i from that vertex to infinity. The bounding line bisects the angle between the two edges of $\mathcal{B}$ which meet at vertex v_i. These n bounding lines divide free space into n regions, denoted $\mathcal{R}_i$, which are unbounded convex quadrilaterals; see Fig. 4.2.

Any grasp lies in a triple of these regions, motivating the following definition. A *triad* is a triple of free space regions, denoted $\mathcal{T}_{i,j,k} = \mathcal{R}_i \times \mathcal{R}_j \times \mathcal{R}_k$. For all grasps in $\mathcal{T}_{i,j,k}$, f_1 lies in $\mathcal{R}_i$, f_2 lies in $\mathcal{R}_j$, and f_3 lies in $\mathcal{R}_k$,. Thus, grasp ρ_a lies in $\mathcal{T}_{2,3,5}$ in Fig. 4.2.

As a step towards finding cages of $\mathcal{B}$ and characterizing their robustness, we consider a simplified problem: what restriction of the hand shape will prevent the hand from leaving a single triad? That is, what hand shapes form a cage for a given triad? (Note that this is usually a sufficient condition for caging $\mathcal{B}$.)

Here we present an example of a *robust cage* (defined later) of polygon $\mathcal{B}$, shown in Fig. 4.3. The red, green, and blue regions of Fig. 3 represent possible placements of f_1, f_2, and f_3, respectively. Fig. 4.4 shows a region of shape space, denoted $\mathcal{H}$, associated with triad $\mathcal{T}_{2,3,5}$ of $\mathcal{B}$. Given the starting positions from Fig. 4.3, the cage will be maintained while allowing the hand shape to vary anywhere within $\mathcal{H}$. More formally, consider a grasp, $\rho = (q, \sigma)$ that lies in $\mathcal{T}_{2,3,5}$ and satisfies $\sigma \in \mathcal{H}$. $\mathcal{H}$ is carefully chosen such starting at ρ, $\mathcal{B}$ will remain caged, even while allowing σ to vary freely in $\mathcal{H}$.

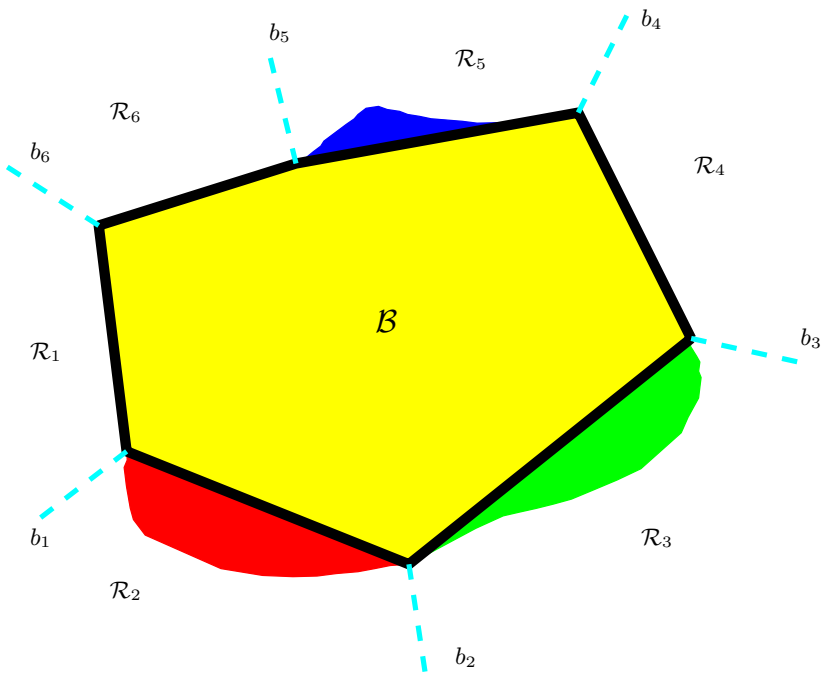


Figure 4.3: A physical caging region for a polygon, shown in yellow. The red, green, and blue regions shown in Fig. 4.3 are possible placements of f_1, f_2, and f_3, respectively, which satisfy both $\rho \in \mathcal{T}_{2,3,5}$ and $\sigma \in \mathcal{H}$ (see Fig. 4.4). Starting from one of these positions, the restriction $\sigma \in \mathcal{H}$ guarantees that the hand will not leave $\mathcal{T}_{2,3,5}$.

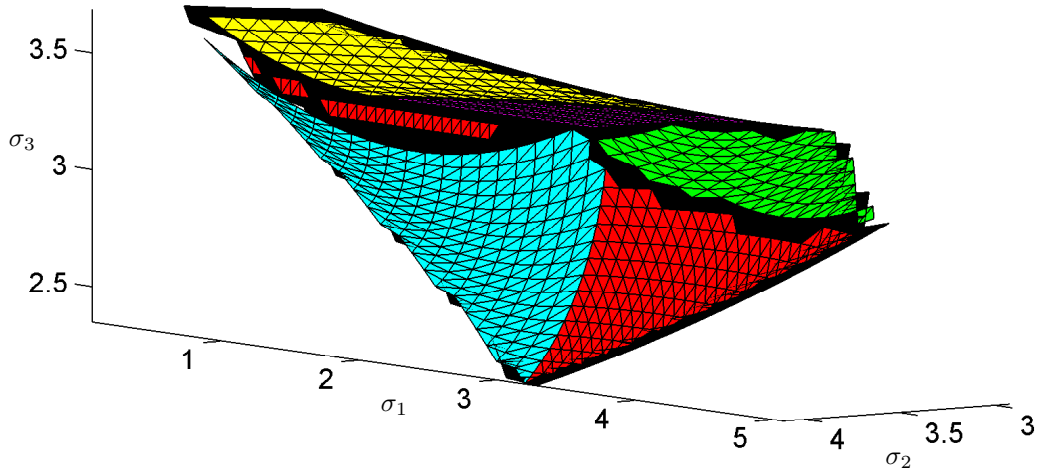


Figure 4.4: A shape space caging region, $\mathcal{H}$ for the polygon shown in Fig. 4.3. Restriction of the hand shape to this region of $\mathcal{S}$ will prevent the hand from leaving the initial triad, $\mathcal{T}_{2,3,5}$.

$\mathcal{H}$ is bounded by two-dimensional manifolds, each resulting from an equilibrium grasp of $\mathcal{B}$. These manifolds will be discussed in detail in Section 4.4. The colored manifolds shown in Fig. 4.4 result from grasps in which one finger lies at a vertex of $\mathcal{B}$. Examples of such grasps, and the manifolds they generate are shown in Figs. 4.7 and 4.8. The black manifold results from immobilizing grasps in which all three fingers lie on edges of $\mathcal{B}$.

4.1.2 Robust Caging Definition

Definition 4.1.3 *Given a grasp, $\rho_0 = (q_0, \sigma_0) \in \mathcal{F}$ which is a cage of $\mathcal{B}$, the set of hand shapes $\mathcal{H} \subset \mathcal{S}$ forms a **robust cage** of $\mathcal{B}$ (associated with q) if $\sigma_0 \in \mathcal{H}$, and $\mathcal{B}$ remains caged while allowing q to vary freely and σ to vary within $\mathcal{H}$.*

More formally, $(\rho_0 \in \mathcal{F}, \mathcal{H} \subset \mathcal{S})$ form a **robust cage** of $\mathcal{B}$, if and only if ρ_0 lies in a bounded component of $\mathcal{F}_{\mathcal{H}}$, where $\mathcal{F}_{\mathcal{H}} = \{\rho \in \mathcal{F} : \sigma(\rho) \in \mathcal{H}\}$. Note that the restriction $\sigma \in \mathcal{H}$ is never a sufficient condition to cage an object - placements far from the object will never cage it, regardless of the shape of the hand. Thus, if we place our hand in configuration p_0, we can allow the hand shape to vary within $\mathcal{H}$ and the object is guaranteed to remain caged.

Other starting configurations near ρ likely form robust cages of $\mathcal{B}$, motivating the following definition.

Definition 4.1.4 *The pair $(\mathcal{I} \subset \mathcal{F}, \mathcal{H} \subset \mathcal{S})$ form a **robust caging set** of $\mathcal{B}$ if, for all $\{\rho \in \mathcal{I} : \sigma(\rho) \in \mathcal{H}\}$, ρ and $\mathcal{H}$ form a robust cage of $\mathcal{B}$.*

Thus, if $\mathcal{I} \subset \mathcal{F}$ and $\mathcal{H} \subset \mathcal{S}$ form a robust caging set, then starting from any grasp $\rho \in \mathcal{I}$, $\mathcal{B}$ will remain caged while allowing σ to vary within $\mathcal{H}$. More formally, the pair $(\mathcal{I} \in \mathcal{F}, \mathcal{H} \subset \mathcal{S})$ form a robust caging set of $\mathcal{B}$ if and only if all $\rho \in \mathcal{I}$ lie in a bounded component of $\mathcal{F}_{\mathcal{H}}$.

We are thus interested in two regions. First, $\mathcal{I}$ is a portion of free c-space in which we want to initially place our fingers. Second, $\mathcal{H}$ is a region of hand shape space, $\mathcal{S}$. If we start at a grasp in $\mathcal{I}$ whose shape lies in $\mathcal{H}$, then we can allow the fingers to vary within $\mathcal{H}$ while guaranteeing a cage. This robustness is one of the main contributions of this chapter. We are also interested in the boundaries of these regions (particularly $\mathcal{H}$) as this is where our guarantee of a cage fails. The following sections will analyze these regions. Section 4.2 will discuss how to find robust caging regions which prevent the hand from leaving a single triad of $\mathcal{B}$. Section 4.3 will extend this analysis to unions of triads. Section 4.4 will describe the manifolds which bound robust caging regions of $\mathcal{S}$, and Section 4.5 will describe simple geometric test to determine if a given point in $\mathcal{S}$ is a cage associated with a set of triads.

4.2 Single Triad Caging

This section will analyze a single triad of $\mathcal{B}$, and consider what restriction on the hand shape will prevent the hand from leaving that triad. For the hand to leave a particular triad, one finger must cross a bounding line and thus move from one region to an adjacent region. The movement of the hand from one triad to another is referred to as a *transition*, and each transition is associated with a particular finger crossing a particular bounding line. These transitions may occur due to variation in either q or σ. (Cases in which two fingers cross two bounding lines simultaneously will be shown to be unimportant.) Thus, in the example in Fig. 4.2, grasp ρ_b moving "past" v_2 can be viewed as f_1 crossing b_2, which is a transition from $T_{2,3,5}$ to $T_{3,3,5}$. Each triad has six adjacent triads and thus six transitions which can potentially leave that triad.

Consider a single triad, $\mathcal{T}_{p,q,r}$, and a single transition into an adjacent triad $\mathcal{T}_{s,t,u}$. Let this transition represent a particular finger, f_i, crossing b_j, one of the bounding lines associated with $\mathcal{T}_{p,q,r}$. For each point, σ^* in shape space, $\mathcal{S}$, we would like to know if fixing the hand shape to $\sigma = \sigma^*$ and allowing q to vary will prevent f_i from crossing b_j, and thus prevent the hand from moving from $\mathcal{T}_{p,q,r}$ to $\mathcal{T}_{s,t,u}$. This motivates the following definition.

Definition 4.2.1 *A **single transition caging region**, denoted $\mathcal{C}_t$, associated with adjacent triads $\mathcal{T}_{p,q,r}$ and $\mathcal{T}_{s,t,u}$, is the set of all points $\sigma^* \in \mathcal{S}$ such that a hand with fixed shape σ^** cannot *move directly from $\mathcal{T}_{p,q,r}$ into $\mathcal{T}_{s,t,u}$.*

Definition 4.2.2 *A **single transition escape region**, denoted $\mathcal{E}_t$, associated with adjacent triads $\mathcal{T}_{p,q,r}$ and $\mathcal{T}_{s,t,u}$, is the set of all points $\sigma^* \in \mathcal{S}$ such that a hand with fixed shape σ^** may *move directly from $\mathcal{T}_{p,q,r}$ into $\mathcal{T}_{s,t,u}$.*

Consider the triad $\mathcal{T}_{p,q,r}$ and the transition to $\mathcal{T}_{s,t,u}$ in which f_i crosses b_j, and a given grasp, $\rho = (q, \sigma) \in \mathcal{T}_{p,q,r}$. If we restrict σ to lie in $\mathcal{C}_t$ (associated with that transition), then the hand will not be able to move from $\mathcal{T}_{p,q,r}$ into $\mathcal{T}_{s,t,u}$ while we allow q to vary freely. Conversely, if we allow σ to take values in $\mathcal{E}_t$, then the hand will be able to move into $\mathcal{T}_{s,t,u}$ and thus escape from $\mathcal{T}_{p,q,r}$.

The boundary between $\mathcal{E}_t$ and $\mathcal{C}_t$ is termed a *puncture manifold*. These can be seen as the colored surfaces in Fig. 4.4. For hand shapes in the interior of $\mathcal{C}_t$ the hand is unable to leave a triad via a particular transition. However, if variations in hand shape which lie on that puncture manifolds are allowed, then the hand will be able to leave the triad by that transition. In Section 4.4 we will consider a transition, t, in which f_i crosses bounding line b_j. We will show that, for t, the associated puncture manifold is a two-dimensional surface, which divides $\mathcal{S}$ is divided into two regions, $\mathcal{E}_t$ and $\mathcal{C}_t$. Each puncture manifolds is the union of four smooth manifolds, each of which results from an equilibrium grasp in which f_i lies at vertex v_j (where b_j contacts $\mathcal{B}$). These grasps are shown in Fig. 4.8; the resulting manifolds in Fig. 4.7. Section 4.4 will provide a catalog of all hand shapes

for which the topology of c-space changes.

We now return to consider all six transitions which allow the hand to leave triad $\mathcal{T}_{p,q,r}$. If a particular restriction on σ allows the hand to reach any of the six adjacent triads, then it is able to escape from $\mathcal{T}_{p,q,r}$. Thus, the *escape region*, denoted $\mathcal{E}$, associated with $\mathcal{T}_{p,q,r}$ is the union of the six single transition escape regions of $\mathcal{T}_{p,q,r}$. Similarly, the *caging region*, denoted $\mathcal{C}$, associated with $\mathcal{T}_{p,q,r}$ is the intersection of the six single transition caging regions of $\mathcal{T}_{p,q,r}$. For triad $\mathcal{T}_{2,3,5}$ of the polygon shown in Fig. 4.3, the caging region in shape space is shown in Fig. 4.4. Valid placements of the fingers in $\mathcal{T}_{2,3,5}$ are shown as the red, green, and blue regions in Fig. 4.3.

Thus, for configurations in $\mathcal{T}_{p,q,r}$, allowing σ to vary within $\mathcal{C}$ (while q varies freely) will guarantee that the hand does not leave $\mathcal{T}_{p,q,r}$. More formally, for $\mathcal{C}$ associated with $\mathcal{T}_{p,q,r}$, the set $\mathcal{F}_{\mathcal{C}}$ contains a bounded component lying entirely within $\mathcal{T}_{p,q,r}$. Returning to our previous discussion of robust cages, $\mathcal{T}_{p,q,r}$ forms an *initial caging region*, and the pair $(\mathcal{T}_{p,q,r}, \mathcal{C})$ forms a *robust caging set* of $\mathcal{B}$.

For some values of $\sigma \in \mathcal{S}$, there may be no points in $\mathcal{T}_{p,q,r}$, having that value of σ. This motivates the following definition.

Definition 4.2.3 *A point $\sigma^* \in \mathcal{S}$ is **feasible** relative to a particular triad $\mathcal{T}_{p,q,r}$ if there exists a grasp $\rho \in \mathcal{T}_{p,q,r}$ satisfying $\sigma(\rho) = \sigma^*$.*

Similarly, a point is *infeasible* if no such grasp exists in $\mathcal{T}_{p,q,r}$. Practically, this means that if a point in $\mathcal{C}$ is infeasible, then grasps having that hand shape cannot be used to cage a particular triad, as no placement in that triad have that hand shape. Section 4.4 will discuss the manifolds which divide the caging region into feasible and infeasible regions, which are generated by the set of immobilizing grasps associated with a particular triad.

4.3 Multi-Triad Caging

We now consider regions of $\mathcal{F}$ other than single triads. In particular, we consider the union of more than one triad.

Definition 4.3.1 *A **union**, denoted $\mathcal{U} \subset \mathcal{F}$ formed by the union of n triads of $\mathcal{B}$, given by $\mathcal{U} = \bigcup_{i \in I} \mathcal{T}_i$, where I is an index set of n triads of $\mathcal{B}$.*

Next we shall determine what restriction on shape space will prevent an initial configuration lying in $\mathcal{U}$ from leaving $\mathcal{U}$.

Remark 4.3.2 *It is important to note that the caging region associated with a set of more than one triad is neither superset or subset of the caging regions of the individual triads.*

Consider a particular triad, $\mathcal{T}_{p,q,r} \in \mathcal{U}$. Assume $\mathcal{T}_{p,q,r}$ is adjacent to another triad $\mathcal{T}_{s,t,u} \notin \mathcal{U}$. Since the transition from $\mathcal{T}_{p,q,r}$ to $\mathcal{T}_{s,t,u}$ would cause the hand to leave $\mathcal{U}$, the escape region

associated with this transition must be an escape region for $\mathcal{U}$, just as it was when considering $\mathcal{T}_{p,q,r}$ as an individual triad. Next, assume that $\mathcal{T}_{p,q,r}$ is adjacent to a third triad, $\mathcal{T}_{x,y,z} \in \mathcal{U}$. Since movement from $\mathcal{T}_{p,q,r}$ to $\mathcal{T}_{x,y,z}$ does *not* cause the hand to leave $\mathcal{U}$ (as both $\mathcal{T}_{p,q,r}$ and $\mathcal{T}_{x,y,z}$ lie in $\mathcal{U}$), the escape region associated with this transition is *not* an escape region for $\mathcal{U}$. This motivates classifying transitions as follows.

Definition 4.3.3 *Given a union, $\mathcal{U}$, an **internal transition** is any transition between two triads,* both *of which lie in $\mathcal{U}$.*

Definition 4.3.4 *Given a union, $\mathcal{U}$, an **external transition** is any transition between two triads, one of which lies in $\mathcal{U}$, the other of which does not.*

Consider the same polygon, shown again in Fig. 4.5, but consider what restrictions on the hand shape will prevent the hand from leaving $\mathcal{U} = \mathcal{T}_{2,3,5} \cup \mathcal{T}_{2,3,6}$. In this case, the transition from $\mathcal{T}_{2,3,5}$ to $\mathcal{T}_{2,3,6}$ is an *internal* transition. Thus, hand shapes in the escape region associated with this transition do not allow the hand to escape from $\mathcal{U}$.

We refer to the single transition escape regions as *internal escape regions* if they are associated with an internal transition, and *external escape regions* if they are associated with an external transition. The escape region associated with $\mathcal{U}$ is thus the union of the external escape regions associated with all of the external transitions of $\mathcal{U}$. Internal escape regions are ignored.

Consider $\mathcal{U} = \mathcal{T}_{2,3,5} \cup \mathcal{T}_{2,3,6}$ for the polygon in Fig. 4.5. Placing f_1, f_2, and f_3 in the red, green, and blue regions will result in a cage which does not leave $\mathcal{U}$, as long as the hand shape is restricted to lie in $\mathcal{H} \subset \mathcal{S}$, shown in Fig. 4.6.

Unfortunately, the caging region for a union of triads is neither the union nor intersection of the caging region for the triads composing the union. Consider the union of two triads $\mathcal{U} = \{\mathcal{T}_{pqr}, \mathcal{T}_{stu}\}$. The caging region associated with just $\mathcal{T}_{pqr}$ is composed of hand shape region $\mathcal{H}_{pqr}$ along with the restriction that the initial placement lies in $\mathcal{I}_{pqr}$. Similarly, the caging region for just $\mathcal{T}_{stu}$ is $\mathcal{H}_{stu}$ with initial initial placement lies in $\mathcal{I}_{stu}$. However, considering $\mathcal{H}_{pqr} \cup \mathcal{H}_{stu}$. Requiring that the initial placement lie in $\mathcal{I}_{pqr} \cup \mathcal{I}_{stu}$ will not guarantee a cage, as a hand shape lying in $\mathcal{H}_{pqr}$ with an initial placement in $\mathcal{I}_{stu}$ will not guarantee a cage. Additionally, restricting the initial placement to $\mathcal{I}_{pqr} \cap \mathcal{I}_{stu}$ does not produce a useful cage as this intersection is the empty set.

4.4 Divisions of Shape Space

This section provides a catalog of grasps which are relevant to three finger cages of convex polygons. It presents the three types of equilibrium grasp which bound robust caging regions, and considers all other types of grasps, showing that they do not form boundaries of the robust caging region.

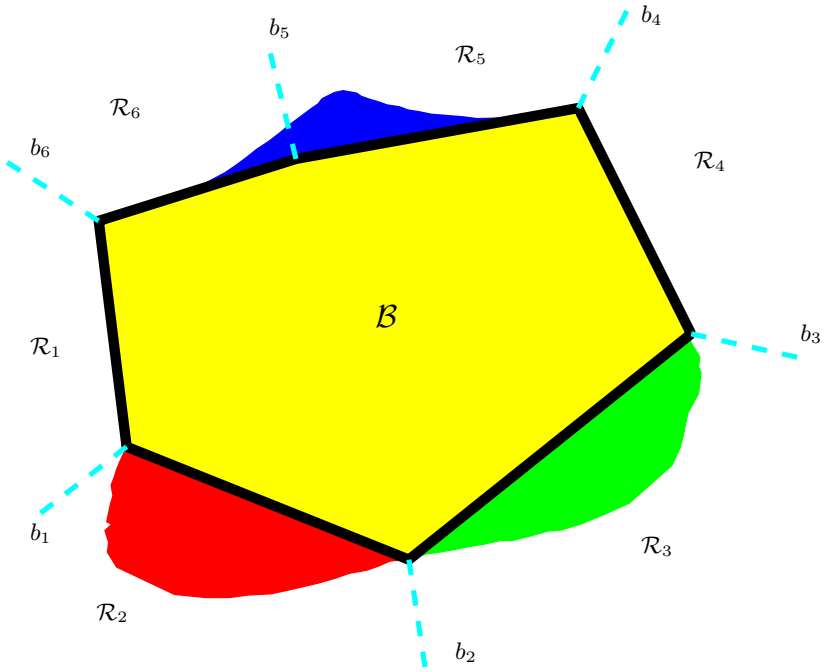


Figure 4.5: A physical caging region for polygon $\mathcal{B}$, considering the union of two triads $\mathcal{U} = \mathcal{T}_{2,3,5} \cup \mathcal{T}_{2,3,6}$. If the fingers are initially placed in the red, green, and blue regions, restricting the shape parameter to lie the region $\mathcal{H} \in \mathcal{S}$ (shown in Fig. 4.6) will prevent the fingers from leaving $\mathcal{U}$.

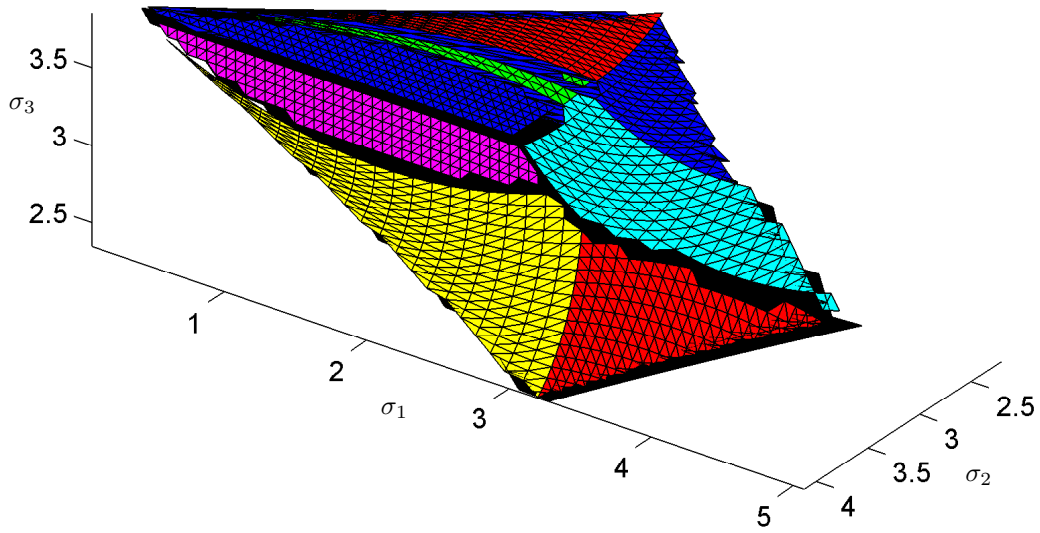


Figure 4.6: A shape space caging regions, $\mathcal{H}$, for a set of triads, $\mathcal{U}$, of the polygon shown in Fig. 4.5. Restriction of the shape parameters to this region, $\mathcal{H}$ will prevent fingers starting in $\mathcal{U}$ from leaving $\mathcal{U}$.

Each of the three fingers may lie at a vertex of $\mathcal{B}$, on and edge of $\mathcal{B}$, or in free space, resulting in 27 possible combinations. Simple geometric arguments show that two of these cases result in puncture manifolds, which divide S into caging and escape regions, and that one of these cases give rise to immobilizing manifolds, which divide S into feasible and infeasible regions. We discuss other grasps and show that they do not give rise boundaries of the caging region.

4.4.1 Puncture Manifolds

Here we describe the two dimensional *puncture manifolds* which divide shape space, $\mathcal{S}$, into escape regions and caging regions. These are the manifolds in shape space which bound the caging region, as seen in Fig. 4.6.

For each transition between two adjacent triads of $\mathcal{B}$, a 2D manifold divides $\mathcal{S}$ into an escape region and a caging region. Consider the transition between $\mathcal{T}_{p,q,r}$ and $\mathcal{T}_{s,t,u}$, in which finger f_i crosses bounding line b_j. The surface associated with this transition is the union of four two-dimensional manifolds. Each one is results from an equilibrium grasp in which f_i lies at the vertex from which b_j emanates. Those four manifolds are shown in Fig. 4.7. Four of the equilibrium grasps which generate these manifolds are shown with dashed lines in Fig. 4.8. The corresponding points in $\mathcal{S}$ are shown as circles which lie on the puncture manifold.

These four manifolds come in two types. One is formed by the set of equilibrium grasps in which one finger lies at vertex v_j (from which b_j emanates), while the other lies on an edge, at the perpendicular projection of v_j onto that edge. The third finger lies in free space, producing two degrees of freedom. One equilibrium grasps associated with each of these manifolds is shown in blue and yellow in Fig. 4.8. The puncture manifolds associated with these grasps are shown in corresponding colors in Fig. 4.7. These manifolds correspond to variations in hand shape in which one σ_i is constant ($i = 1, 2, \text{or } 3$).

The other two manifolds are formed by equilibrium grasps in which one finger lies at vertex v_j while the other two fingers lie on edges of $\mathcal{B}$. Two of these grasps are shown in red and green in Fig. 4.8, and the corresponding manifolds in $\mathcal{S}$ are shown as corresponding colors in Fig. 4.7. The particular grasps are shown as colored circles in S. Other grasps which generate these manifolds occur when f_1 and f_2 both lie in the red regions (or both in the green regions) of their respective edges.

We will now show that the manifolds described above do lie on the boundary of the caging and escape regions associated with a single transition between adjacent triads, and that other types of contact do not lie on that boundary. Consider a particular transition, t, such that f_i crosses b_j. For t, shape space is divided into an escape region (hand shapes for which f_i is able to cross b_j) and a caging region (hand shapes for which f_i is able to cross b_j). Any hand shape, σ^* that lies on the boundary between these regions has the following property: for $\sigma = \sigma^*$, f_i can cross b_j (for some

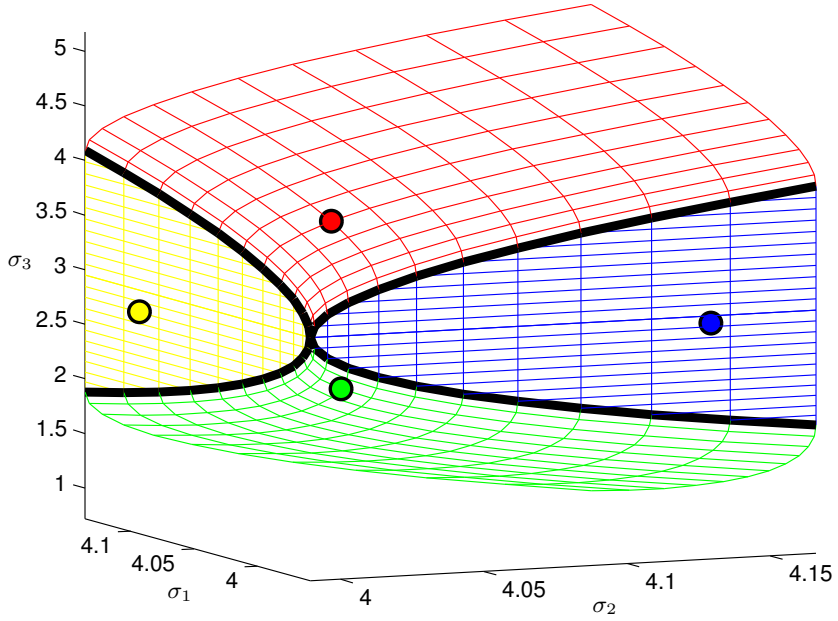


Figure 4.7: Puncture manifolds which divide $\mathcal{S}$ into caging and escape regions. These manifolds are generated by grasps shown in Fig. 4.8.

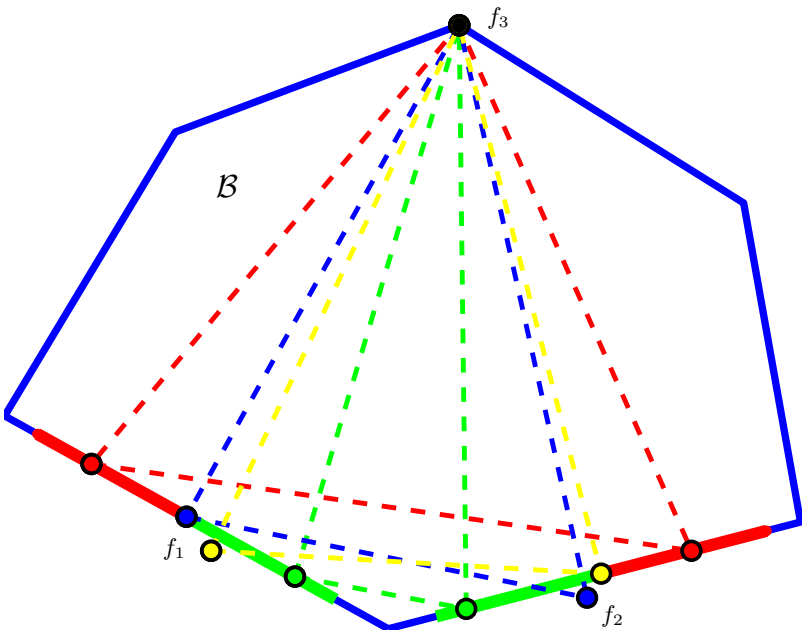


Figure 4.8: Grasps in physical shape which result in puncture manifolds. Corresponding points in $\mathcal{S}$ are shown as colored circles in Fig. 4.7.

variations of q), but for some of the values of σ near σ^*, f_i is unable to cross b_j for any values of q.

Consider the grasp shown in Fig. 4.9, in which f_3 lies at v_j, and f_2 lies on edge e_k at the perpendicular projection of v_j onto e_k. For this hand shape, $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$, f_3 can reach b_j. However, for perturbations of this shape, specifically ones in which $\sigma_1 < \sigma_1^*$, f_3 will be unable to reach b_j while allowing q to vary. Thus, this configuration lies on the boundary of this single transition escape region. Thus configurations in which one finger lies at a vertex of $\mathcal{B}$ while another finger lies at the perpendicular projection of that vertex onto an edge form boundaries between single transition escape and caging regions in $\mathcal{S}$.

Consider the grasp shown in Fig. 4.10, in which f_3 lies at v_j, while f_1 and f_2 lie on edges of $\mathcal{B}$. Possible movements of the fingers (while keeping σ constant) are shown as orange arrows. For $\sigma = \sigma^*$, f_3 is able to cross b_j. However, for perturbations which reduce any (or all) of the inter-finger distances, f_3 will be unable to cross b_j. Thus this configuration lies on the boundary between single transition escape and caging regions in $\mathcal{S}$.

4.4.2 Immobilizing Manifolds

Another set of manifolds divide shape space for a single triad into feasible and infeasible regions. Recall that feasible regions of $\mathcal{S}$ for a single triad $\mathcal{T}_{i,j,k}$ contain points σ^* for which there exits $\rho \in \mathcal{T}_{i,j,k}$ satisfying $\sigma(\rho) = \sigma^*$. Consider the frictionless immobilizing grasp shown in Fig. 4.11. A necessary and sufficient condition for a three-fingered grasp to be (second order) immobilizing is that the three contact normals meet at a common point. The set of such grasps for a triple of edges forms a two-dimensional surface in contact space. The value of σ^* for the grasp in Fig. 4.11 is feasible as the fingers all lie on the boundary of free space. However, reducing any (or all) of the inter-finger distances will produce a point which is in the infeasible region of $\mathcal{S}$, as a hand with that shape will be impossible to place in $\mathcal{T}_{i,j,k}$.

4.4.3 Other Grasps

Now consider a hand shape which does not lie on the boundary between single transition escape and caging regions in $\mathcal{S}$. Consider the grasp where f_3 lies at a vertex of $\mathcal{B}$, but f_1 and f_2 lie in free space, shown as a purple grasp in Fig. 4.12. For this hand shape, f_3 can reach v_j. Also, for any small perturbation in hand shape, f_1 and f_2 may move in free space while leaving f_3 able to reach b_j. Thus, this configuration lies in the interior of the single transition escape region, and not on the boundary of this region. Similar arguments may be used to show that grasps in which one finger lies on an edge of $\mathcal{B}$ while the others lie in free space, or where all three fingers lie in free space also do not lie on the boundary of a caging region. Thus, no grasp in which zero or one finger contact the object form a boundary of a caging region.

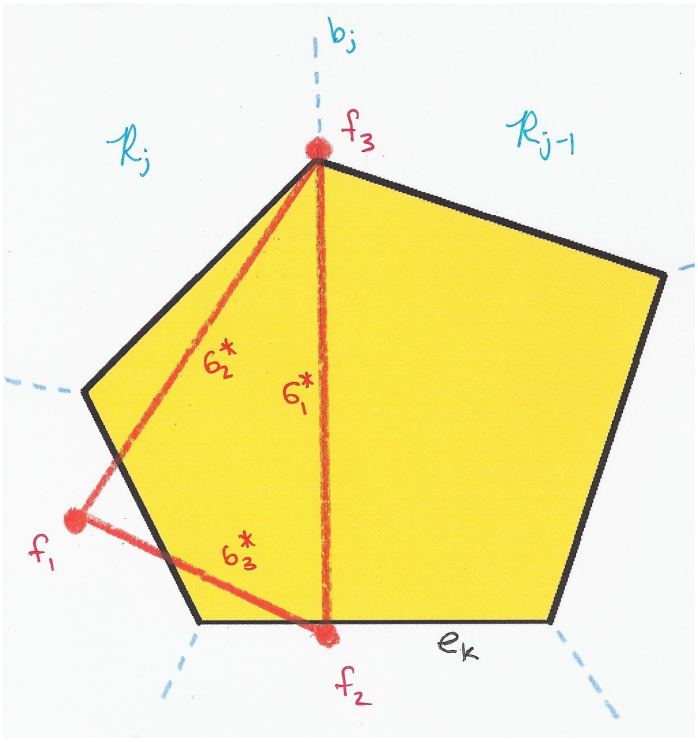


Figure 4.9: A grasp in which one finger lies at a vertex of $\mathcal{B}$ while another lies at the perpendicular projection of that vertex onto an edge of $\mathcal{B}$, which allows the hand to transition from one triad to another.

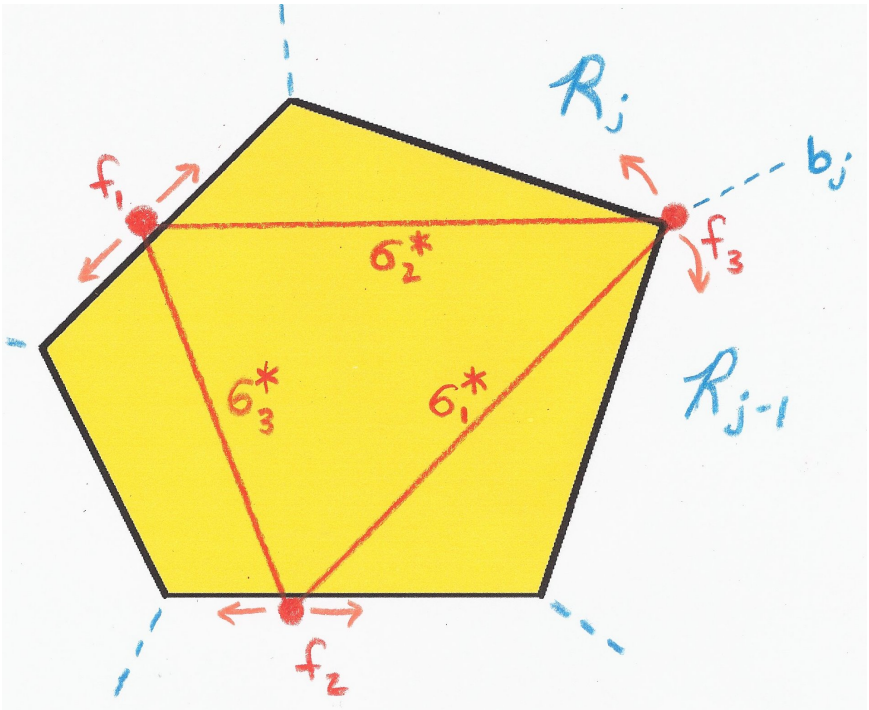


Figure 4.10: A grasp in which one finger lies at a vertex of $\mathcal{B}$ while two other fingers lie on edges of $\mathcal{B}$, which allows the hand to transition from one triad to another.

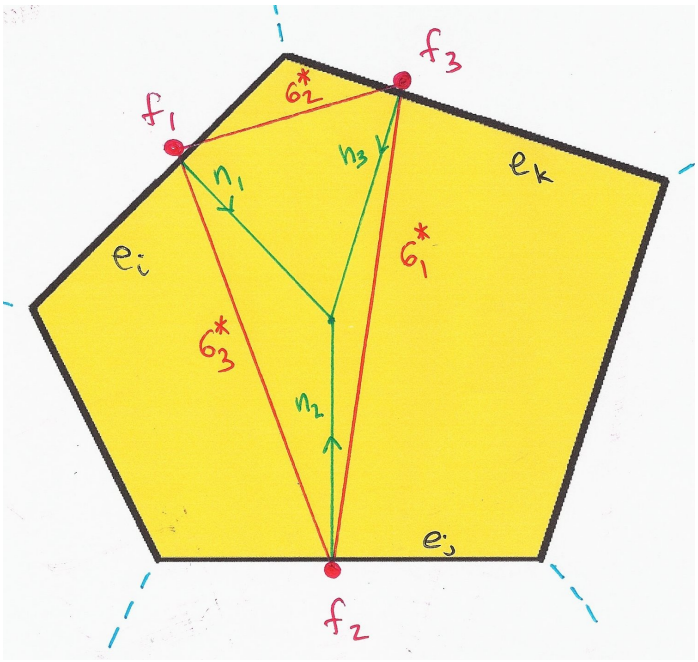


Figure 4.11: An immobilizing grasp, whose hand shape lies on an immobilizing manifold.

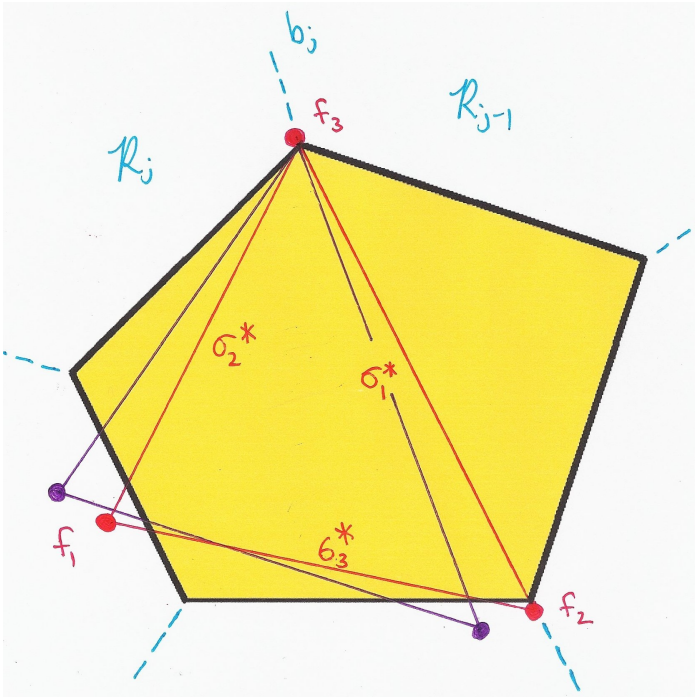


Figure 4.12: Two grasps whose hand shape of these grasps does *not* lie on a boundary of the caging region.

Finally, consider the grasp shown in Fig. 4.12, with hand shape $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$, where f_3 lies at v_j (where b_j contacts $\mathcal{B}$), and f_2 lies at another vertex of $\mathcal{B}$. This hand shape also lies in the interior of the single transition escape region, $\mathcal{E}_t$, associated with f_3 crossing b_j. To see this, consider rotating the hand (keeping the shape fixed at σ^*), as shown in purple. This is a configuration discussed above, and lies in the interior of $\mathcal{E}_t$. Similar arguments (not presented here) may be made about other types of grasp in which two fingers touch the object, such as grasps where both fingers touch edges of $\mathcal{B}$, or grasps where one finger lies at a vertex of $\mathcal{B}$, while the other lies on an edge but not at the perpendicular projection of the vertex onto that edge. None of these configurations form boundaries of the caging region.

There are four cases in which all three fingers touch $\mathcal{B}$. The cases in which all three fingers lie on edges of $\mathcal{B}$ or where two fingers lie on edges while one lies at a vertex have been considered above. The final two possible combination (two fingers at vertices with the third on an edge, or all three fingers at vertices) may be shown to be limiting cases of the three-finger puncture manifolds described above, and are thus lower dimensional portions of those manifolds.

Thus, after considering all possible type of three-finger contact, only three result in boundaries of the caging region: two equilibrium grasps which result in puncture manifolds and one equilibrium grasp which results in an immobilizing manifold.

4.5 A Test of Caging Status

This section describes tests which determine whether a given point, σ, lies in the local escape region or local caging region, and if it lies in the feasible or non-feasible region. This test may be used in grasp planning to find particular grasps which are cages of $\mathcal{B}$ as well as identify the regions in which the hand shape may vary without losing that cage. We start by describing a parameterization of all grasps for which $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$ in which two of the fingers touch edges of $\mathcal{B}$.

Let finger f_1 contact edge e_i and f_2 contact e_j, as shown in Fig. 4.13. Let the origin be the intersections of the lines underlying e_i and e_j, and let the x-axis lie along e_j and the angle between those lines be ϕ. Let the distance from the origin to f_1 be r_1, and the distance from the origin to f_2 be r_2. Finally, let θ represent the angle between the x-axis and a line running from f_1 to f_2. The position of f_1 is:

$$f_1 = \begin{bmatrix} r_1 \cos\phi \\ r_1 \sin\phi \end{bmatrix} = \begin{bmatrix} r_2 + \sigma_3 \cos\theta \\ \sigma_3 \sin\theta \end{bmatrix} \tag{4.1}$$

Solving for r_1 and r_2 allows f_3 can be written in terms of θ. This parameterization is the basis for the following tests.

4.5.1 Test for Feasibility

Because the test for feasibility is simpler, we will describe it first. For a given value of $\boldsymbol{\sigma}$, we can write $f_3(\theta)$, as described above. Using the 2D cross product, we can get the signed distance from f_3 to e_k. This distance takes the form $a\cos\theta + b\sin\theta + c$. If this distance is non-negative, then there exist finger placements f_1 and f_2 such that f_3 lies in free space, and thus a feasible placement exists for hand shape σ. This is true when $a^2 + b^2 - c^2 \geqslant 0$. Since a, b, and c are all functions of the geometry of e_i, e_j, e_k, and $\boldsymbol{\sigma}$, this can be easily evaluated for any value of $\boldsymbol{\sigma}$. It also gives a measure of how far from the feasible/non-feasible boundary a given configuration is.

4.5.2 Test for Escape

The test of escape is similar to the test for feasibility. Since the caging region is bounded by puncture manifolds resulting from both two and three finger equilibrium grasps, we must check both. We start with a check related to three finger equilibrium grasps, as follows. As above, we can write the position of f_3 as a function of θ and known geometry. We are interested in whether f_3 can reach a particular boundary line, b_m. If the finger is able to reach b_m (under variations in q), then this hand shape will be in the single transition escape region, while if it is unable to reach b_m, then the hand shape lies in the single transition caging region for this particular transition. To determine this, consider the signed distance from f_3 to b_m, which may be written $d\cos\theta + e\sin\theta + f$; see Fig. 4.13. Solving for the zeros of this equation gives us the two values θ_a and θ_b, for which f_3 touches b_m. We then simply check the signed distance from $f_3(\theta_a)$ and $f_3(\theta_b)$ to $b_{m\perp}$, where $b_{m\perp}$ is a line emanating from v_m, perpendicular to b_m.

The checks related to two-fingered equilibrium grasps are simple. For the particular bounding line shown in Fig. 4.8, the checks are simply $\sigma_1 \geqslant \sigma_1^*$ and $\sigma_2 \geqslant \sigma_2^*$ where σ_1^* and σ_2^* are the minimum distance from v_m to e_i and e_j, respectively.

4.6 Example

Here we present the cages for a union of triads, $\mathcal{U}$, of the polygon shown in Fig. 4.14. In particular, we are interested in the union of the eight triads in which f_1 lies in $\mathcal{R}_1$ or $\mathcal{R}_2$, f_3 lies in $\mathcal{R}_4$ or $\mathcal{R}_5$, and f_3 lies in $\mathcal{R}_7$ or $\mathcal{R}_8$. The shape space caging region, $\mathcal{H}$, is shown in Fig. 4.15. Possible finger placements in $\mathcal{U}$ which satisfy $\sigma \in \mathcal{H}$ are shown as the red, green, and blue regions in Fig. 4.14. Restricting the hand shape to lie in $\mathcal{H}$ will prevent the fingers from leaving the initial caging region.

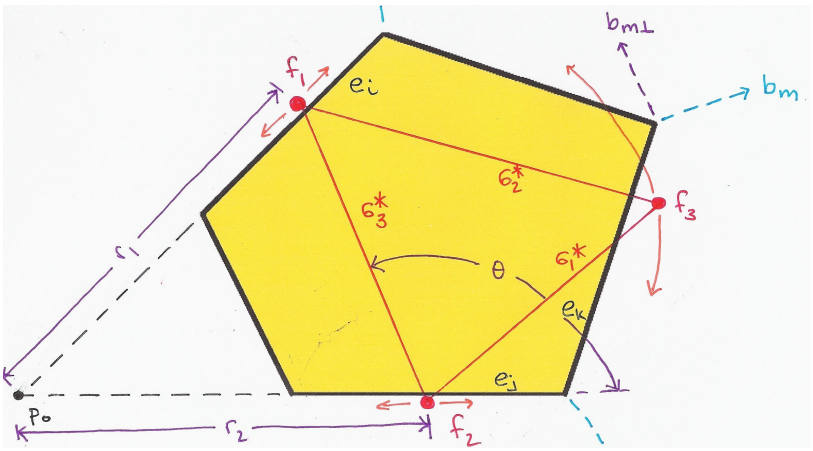


Figure 4.13: A parameterization of grasps with two finger contacts. This provides tests of whether a point is caging or escape, and feasible or non-feasible.

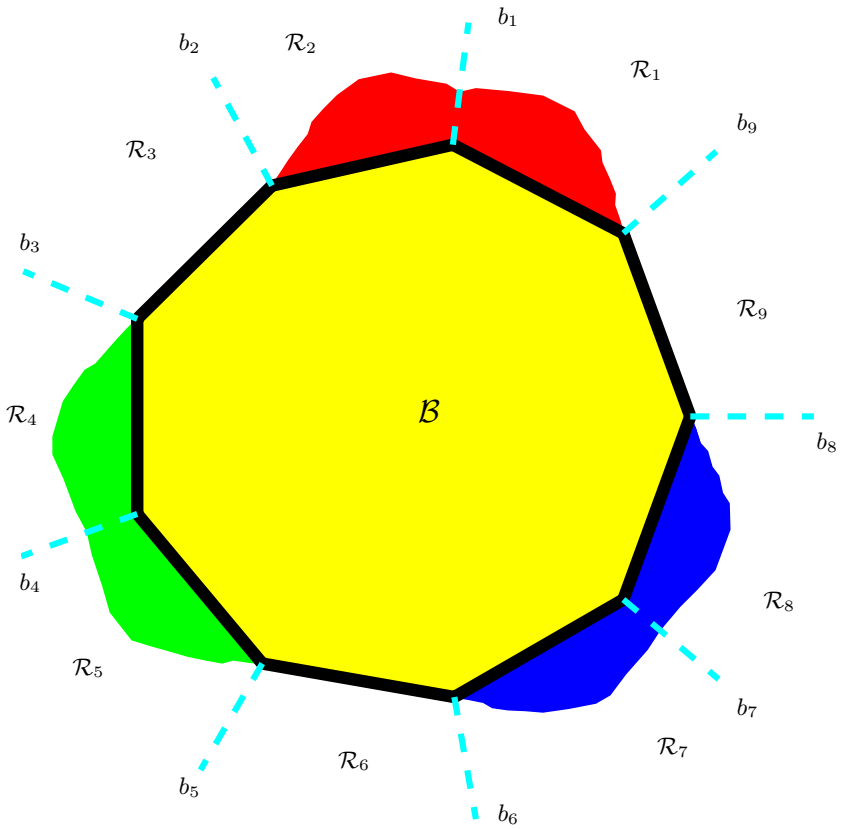


Figure 4.14: Physical caging region for an initial caging region composed of eight triads. The caging region associated with these triads is quite complicated, and is shown in Fig. 4.15.

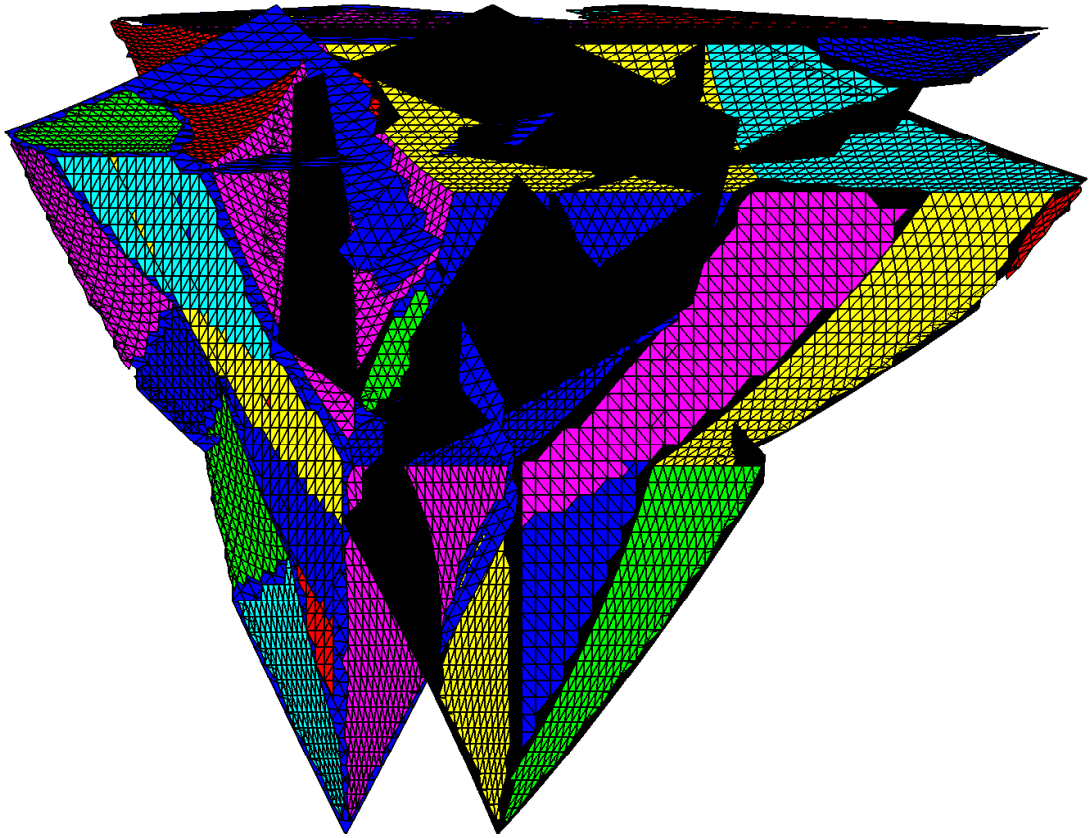


Figure 4.15: Shape space caging region, $\mathcal{H}$ for the polygon shown in Fig. 4.14. Colored portions are puncture manifolds; black portions are immobilizing manifolds. The region is quite complex.

4.7 Computational Complexity

Each triad is associated with three three-finger puncture manifold and three two-finger puncture manifolds. These deal only with the geometry local to the triad, and thus can be computed in constant time. For a set of k triads, there will be at most $6k$ external edges. Thus, any point associated with a given union of k triads $\mathcal{U}$ can be checked in $\mathcal{O}(k)$ time.

Unfortunately, the number of possible unions of triads is combinatorial in n^3, where n is the number of edges of $\mathcal{B}$. Thus, computing all cages for $\mathcal{B}$ results in impractically large computational cost, even for modest n. In practice the single union of triads shown in Fig. 4.14 can be calculated in seconds, but the number of possible unions makes computing them all completely infeasible.

Several things may be done to improve this. First, any triad for which the edge normals do not span $\mathbb{R}^2$ can be trivially escaped. Thus, any union of triads containing such a triad may be ignored. Additionally, if a union of triads does not form a connected set, the disconnected components may be considered separately.

Second, not all unions must be considered to find sufficient conditions for caging. The example in Fig. 4.14 shows a robust caging region which would be useful for grasp planning even if other unions of triads have not been checked.

Comparisons of computational complexity to existing algorithms is difficult. Vahedi and van der Stappen [43] gives an algorithm to find whether a three finger grasp cages a polygon in $log(n)$ time, which is significantly better than this algorithm, but does not consider robustness.

This algorithm may find robust caging regions associated with a single union of triads, $\mathcal{U}$, by sampling shape space, as a cost of $\mathcal{O}(k)$ per sample, where k is proportional to the number of triads in $\mathcal{U}$. Neither the work of Erickson, et al [16] or Vahedi and van der Stappen [43] nor are intended to answer this question. Both Pipattanasomporn et al [27] and Rodriguez et al [38] could give a useful subset of the robust caging region, but comparing computational complexity is not straightforward.

4.8 Conclusion

This chapter considers caging a convex polygon with three point fingers. It discusses a notion of *robustness* of a cage to variations in the hand parameter. It gives a simple decomposition free space around the polygon into *triads*, which forms the basis for analysis. It considers robust cages which prevent the hand from leaving either a single triad, or a union of triads, and shows that the region of hand space within which the hand shape may vary depends on which triads are considered. It gives a simple geometric test for whether a given hand shape cages a given union of triads. An implemented example shows that the algorithm can produce useful caging regions.

The algorithm comes at considerable cost in computational complexity. Based on the shapes

of the caging regions for relatively simple cases, we believe that this complexity is inherent to the problem. Clearly there is much work yet to be done in the area of robust caging.

Several ways this algorithm could be improved is by finding a larger set of unions of triads which do not need to be searched, and finding unions of triads that are likely yield large robust caging regions. It is possible than analyzing a given union of triads may give insight into which triads might be fruitfully added to that union. Additionally, methods for finding simple geometric regions within the robust caging region (e.g. the largest inscribed sphere) would be useful. Extension to non-convex polygons is desirable but appears difficult. Extension to 3D objects requires extension to non-convex objects, as convex 3D objects cannot be caged with three point fingers.

Chapter 5

Conclusion

5.1 Summary of Thesis Contributions

The main contributions of this thesis are caging algorithms for two and three fingers, as well as deep analysis of caging on contact space, new insights about the topological relationship between configuration space and contact space for caging, and a catalog of important grasps that underlie these algorithms. The first two algorithms calculate two finger cages of polygonal and polyhedral objects in two and three dimensions. The third algorithm calculates robust three finger cages of convex polygons in two dimensions. All algorithms are supported with novel analysis of the topology of configuration space, as well as straightforward geometric interpretations of the results, and implemented examples.

The two-finger, two-dimensional algorithm calculates the initial, intermediate, and maximal cages associated with an initial immobilizing grasp of an object. This algorithm is based on constructing a graph whose nodes are include all puncture grasps, which are the points at which the topology of configuration space changes. Topology of configuration space is analyzed using stratified Morse theory to show that there are only a few special points where the topology of a simple graph differs from the topology of configuration space, and a simple procedure for modifying the graph to ensure equivalent sublevel set topology is given. A catalog of all possible two finger grasps is presented, producing an easy test as to which grasps may be immobilizing or puncture grasps. The resulting algorithm is both conceptually simple and easy to implement. An implementation based on real-world data supports and validates the practical use of the algorithm.

The two-finger, three-dimensional algorithm is an extension of the two-dimensional algorithm to three dimensions. In addition to handling the complexities of three dimensions, this chapter uses straightforward geometric arguments to produce a catalog of all possible type of two finger grasps in three dimensions. This catalog is significantly more complicated than for two dimensions, but a simple topological test is given which allows any grasp to be easily checked. Additionally, analysis is made using geometry, making the algorithm understandable without the use of stratified Morse

theory. The algorithm is simple to implement, and an implemented example is presented.

The three-finger algorithm is the first algorithm which produces exact caging regions for a three-fingered hand which are robust to variation in the position of the fingers relative to each other while the object is free to move. Caging regions are given as a region of c-space into which the hand must be place, and regions of the hand shape are defined which prevent the object from escaping. These regions are three dimensional, and are bounded by two dimensional manifolds referred to as puncture manifolds, which are analogous to puncture points in two dimensions. Straightforward geometric descriptions of these puncture manifolds are given, along with simple tests to determine whether a given hand shape lies within a particular caging region.

5.2 Opportunities for Future Work

There are several direct extensions to the work presented here. Computational complexity of the three finger algorithm make it impractical to find all caging sets for polygons with even moderate number of edges and sides. This is because different unions of triads produce different caging regions, and the number of unions of triads is combinatorial in the number of edges of the polygon. However, many unions of triads do not form useful cages. For example, unions containing a triad whose edge normals do not span $\mathbb{R}^2$ can be trivially escaped, and unions which are not connected can be treated as independent unions. Work to reduce the number of triads could improve the computational complexity of this approach substantially.

The three fingers work could be extended to non-convex polygons. The connectivity of different triads is substantially more complicated for non-convex polygons. This complexity could be treated using a graph structure similar to the two-finger algorithms presented here, or using a convex decomposition of free space. Analyzing the puncture grasps, which are manifolds for three fingers (instead of points, for two fingers) would be the predominant challenge.

Other than a work by Pipattanasomporn [26], very little of the caging literature has considered uncertainty in object shape knowledge. Much of the usefulness of caging will relate to grasping objects whose shape is determined online from vision sensors. The object models generated in this way will have uncertainty in the position of vertices. In addition, information about portions of the object will be missing, either due to occlusion by other object, or, for three dimensional objects, occlusion of the back side of the object by itself.

The two finger algorithms presented here could be extended to deal with uncertainty in vertex position as follows. Uncertainty in vertex position creates uncertainty in the inter-finger distance associated with different nodes of the caging graph. A scalar metric on this uncertain distance could be developed. For a bounded uncertainty, then worst-case distance could be used. For unbounded uncertainty (e.g., a Gaussian distribution) then some tradeoff between mean and variance could

be used. This metric would replace the inter-finger distance function in the two finger algorithms, and possibly the three finger algorithm. For the two finger case, one complication is that some vertex-edge nodes might only exist statistically.

The two finger algorithms could also be extended to consider updated information as the vision system refines its estimate of the object. For the two finger algorithms this would probably involve graph search replanning.

More generally, future work in caging should be aimed at making caging useful for grasping using real robotic hands. To do so, progress is needed in two main areas. First, most caging needs to be extended to cases involving more than two fingers in three dimensions. Currently algorithms that consider more than two fingers are restricted to two dimensions, and algorithms in three dimensions are restricted to only two fingers. An algorithm that analyzed four or more fingers caging non-convex polyhedrons in three dimensions, even if it provided approximate answers, or subsets of the entire caging region would have applications on real robotic system. Second, errors in finger placement or object models are a major limiter in grasping, whether involving caging or not. Caging algorithms which provide robustness to these forms of uncertainty would improve grasping performance on real world systems. The robustness provided by methods in Chapter 4 is a first step in this direction.

Extensions to four or more fingers in three dimensions might follow from several different approaches. Algorithms which project a set of finger positions to a single parameter [29, 38] could applied in three dimensions. These algorithms would be particularly useful if projections functions can be found which map nicely to single parameter gripper. Contact space algorithms, such as those presented here, could be extended to three dimensions. For more than two fingers, important grasps exist in which some of the fingers do not touch the object - these cases would have to be incorporated to contact space, possibly with a concept similar to the contact space tunnel curves presented here. In two dimensions, all puncture grasps occur (generically) with only two or three fingers touching the object. Thus, extensions to more fingers (still in two dimensions) could involve considering them as sets of three fingers, then synthesizing the resulting analysis. There are likely similar bound on the number of fingers that generically result in puncture grasps in three dimensions. Extensions to the three finger algorithm presented here to non-convex and three dimensional objects, and to more than three fingers, appear possible, but computational complexity will be a significant challenge. Because caging essentially involves proving lack of existence of paths, progress in caging might come from planning algorithms, particularly algorithms that could be efficiently updated as configuration space changes due to changes in the relative position of fingers.

This work has made a small step towards defining a robust cage for three point fingers, as have Pipattanasomporn and Sudsang [26] for two point fingers. Extensions of robustness to the position of the fingers relative to each other, position of the fingers relative to the object, and geometry of the object would improve the applicability of caging to real-world system.

Appendices

Appendix A

Stratified Morse Theory Proofs

This appendix contains proofs of Proposition 2.4.4 and Theorem 2.6.7.

Proposition 2.4.4. *The two-finger puncture grasps of $\mathcal{B}$ are* **saddle points** *of $d(s_1, s_2)$ in $\mathcal{U}$.*

Proof Let (q_1, σ_1) be a puncture point in $\mathcal{C}$, and let (s_1^1, s_2^1) be the corresponding point in $\mathcal{U}$. When the fingers' opening parameter, σ, varies in the interval $[\sigma_1 - \epsilon, \sigma_1]$, each fixed-$\sigma$ slice of $\mathcal{C}$ contains a 3D cavity surrounded by the finger c-obstacles. In a local neighborhood of (q_1, σ_1), the fixed-σ slices of the two c-obstacles overlap and their surfaces intersect along two curves—one from the inside of the cavity and one from the outside of the cavity. Call this set of curves $\alpha(\sigma)$. The curves of $\alpha(\sigma)$ appear as pairs of locally distinct contour segments of $d(s_1, s_2)$ in $\mathcal{U}$. These contours approach the point (s_1^1, s_2^1) as σ increases in the interval $[\sigma_1 - \epsilon, \sigma_1]$. Next consider the interval $[\sigma_1, \sigma_1 + \epsilon]$. When σ varies in this interval, the fixed-σ slices of $\mathcal{C}$ contain an expanding puncture in a local neighborhood of (q_1, σ_1). The c-obstacle surfaces in each fixed-σ slice intersect along two distinct curves in this local neighborhood. Call this set of curves $\beta(\sigma)$. The curves of $\beta(\sigma)$ also appear as pairs of locally distinct contour segments of $d(s_1, s_2)$ in $\mathcal{U}$. These contours move away from (s_1^1, s_2^1) as σ increases in the interval $[\sigma_1, \sigma_1 + \epsilon]$. The point (s_1^1, s_2^1) is thus surrounded by two families of contour segment pairs. The family $\alpha(\sigma)$ approaches the point (s_1^1, s_2^1) with $\sigma \in [\sigma_1 - \epsilon, \sigma_1]$, while the family $\beta(\sigma)$ moves away from (s_1^1, s_2^1) with $\sigma \in [\sigma_1, \sigma_1 + \epsilon]$. The point (s_1^1, s_2^1) is thus a saddle point of $d(s_1, s_2)$ in $\mathcal{U}$. □

Theorem 2.6.7 requires some background in stratified Morse theory [18]. Stratified Morse theory, or SMT, captures the changes in the topology of the fixed-σ slices of the free c-space $\mathcal{F}$ in terms of the critical points of the function $\pi : \mathcal{F} \to \mathbb{R}$, given by $\pi(q, \sigma) = \sigma$. When σ varies in the open interval between adjacent critical values of π, the sublevel sets $\mathcal{F}_{\leqslant c} = \{(q, \sigma) \in \mathcal{F} : \pi(q, \sigma) \leqslant c\}$ are topologically equivalent (homeomorphic) to each other. In particular, the path connectivity of the sublevel sets is preserved between critical values of π. Any path connectivity change in the fixed-σ slices of $\mathcal{F}$ must occur locally at a critical point of π in $\mathcal{F}$.

Let p be a critical point of π in $\mathcal{F}$.[1] The type of a critical point is characterized by the behavior

[1] At a critical point of π the gradient, $\nabla\pi = (0, 0, 0, 1)$, is orthogonal to the submanifold of $\mathcal{F}$ containing this point.

of π on two complementary subsets of $\mathcal{F}$. The first set is the submanifold of $\mathcal{F}$ containing the point p, denoted $\mathcal{M}$. The other set, the *normal slice* at p, is constructed as follows. Let $D(p)$ be a small disc centered at p, which intersects the submanifold $\mathcal{M}$ only at p and is transversal to $\mathcal{M}$. The *normal slice*, denoted $E(p)$, is the set $E(p) = D(p) \cap \mathcal{F}$. The behavior of π on $\mathcal{M}$ is characterized by the *Morse index,* denoted κ, which is the number of negative eigenvalues of $D^2\pi(p)$ evaluated on $\mathcal{M}$. The behavior of π on $E(p)$ is determined by the *lower half link* set, denoted l^-. It is defined as the intersection of $E(p)$ with the level set $\mathcal{F}_{c-\epsilon} = \{(q,\sigma) \in \mathcal{F} : \pi(q,\sigma) = c - \epsilon\}$. Thus $l^- = E(p) \cap \mathcal{F}_{c-\epsilon}$, where $c = \pi(p)$ and $\epsilon > 0$ is a small parameter.

According to SMT, a critical point p is a *local minimum* of π in $\mathcal{F}$ if and only if $\kappa = 0$ and $l^- = \varnothing$. The condition $\kappa = 0$ ensures that π has a local minimum along the submanifold $\mathcal{M}$. The condition $l^- = \varnothing$ ensures that π has a local minimum along the transversal directions in a local neighborhood of $\mathcal{F}$ centered at p. The following lemma specifies under what condition l^- is empty at p. Each finger c-obstacle, $\mathcal{CB}_i$, is a four-dimensional set in $\mathcal{C}$. Let $\boldsymbol{\eta}_i(p)$ denote the outward unit normal to $\mathcal{CB}_i$ at a point $p \in \mathbf{bdy}(\mathcal{CB}_i)$. In the lemma, $\tilde{f}$ is a smooth function defined on $\mathcal{C}$ (which can be thought of as $\mathbb{R}^4$), and f is the restriction of $\tilde{f}$ to the free c-space $\mathcal{F}$.

Lemma A.0.1 ([33]) *Let $f : \mathcal{F} \to \mathbb{R}$ be a Morse function on the free c-space $\mathcal{F}$. Let p be a critical point of f on a submanifold $\mathcal{S}$ of $\mathcal{F}$, such that $\mathcal{S}$ is the intersection of the finger c-obstacle boundaries. Then a necessary and sufficient condition for the* **lower half link** *of f at p, l^-, to be* **empty** *is:*

$$\nabla \tilde{f}(p) = \lambda_1 \boldsymbol{\eta}_1(p) + \lambda_2 \boldsymbol{\eta}_2(p), \tag{A.1}$$

for some strictly positive scalars λ_1 and λ_2.

The lemma will be used in the proof of the following theorem.

Theorem 2.6.7. *Let (q_0, σ_0) be an immobilizing grasp of $\mathcal{B}$, and let (q_{esc}, σ_{esc}) be the maximal puncture point associated with (q_0, σ_0). The union of the double-contact submanifold $\mathcal{S}$ with the tunnel curves, $\mathcal{S} \cup \mathcal{T}$, is* **sublevel equivalent** *to the connected component of $\mathcal{F}_{\leqslant \sigma}$ containing (q_0, σ_0) for $\sigma \in [\sigma_0, \sigma_{esc}]$.*

Proof Let $p = (q, \sigma)$ denote points in $\mathcal{C}$. The connected component of $\mathcal{F}_{\leqslant \sigma}$ containing the point $p_0 = (q_0, \sigma_0)$ may contain other immobilizing grasps. Suppose it contains only one additional immobilizing grasp located at $p_0' = (q_0', \sigma_0')$ (the proof extends to multiple immobilizing grasps). Note that p_0 and p_0' lie on the double-contact submanifold $\mathcal{S}$. Since these are immobilizing grasps, they are *local minima* of the function $\pi(q, \sigma) = \sigma$ in the free c-space $\mathcal{F}$. The points p_0 and p_0' initially lie in their individual caging sets. As σ increases beyond σ_0, the two caging sets eventually join at a puncture point, denoted $p_1 = (q_1, \sigma_1)$, which also lies in $\mathcal{S}$.

Let $z, z' \in \mathcal{S}$ be two arbitrary points in the connected component of $\mathcal{F}_{\leqslant \sigma}$ containing p_0. Let us

show that z and z' can be connected by a σ-sublevel path which lies in $\mathcal{S} \cup \mathcal{T}$. If z and z' have σ-decreasing paths in $\mathcal{S} \cup \mathcal{T}$ leading to the *same* immobilizing grasp, the union of these two paths connects z and z' in the sublevel set $\mathcal{F}_{\leqslant\sigma}$. Suppose z has a σ-decreasing path in $\mathcal{S} \cup \mathcal{T}$ leading to p_0, while z' has a σ-decreasing path in $\mathcal{S} \cup \mathcal{T}$ leading to p_0'. In this case p_0 and p_0' lie in the same connected component of $\mathcal{F}_{\leqslant\sigma}$, hence $\sigma \geqslant \sigma_1$. The puncture point p_1 can be connected by σ-decreasing paths in $\mathcal{S} \cup \mathcal{T}$ to both p_0 and p_0'. Hence p_0 and p_0' can be connected via p_1 by a σ_1-sublevel path which lies in $\mathcal{S} \cup \mathcal{T}$. Since $\sigma \geqslant \sigma_1$, the points z, z', and p_1 all lie in $\mathcal{F}_{\leqslant\sigma}$. It follows that z and z' can be first connected to p_0 and p_0', which can be connected via p_1 by a σ-sublevel path in $\mathcal{S} \cup \mathcal{T}$. This establishes the sublevel equivalence of $\mathcal{S} \cup \mathcal{T}$ with $\mathcal{F}_{\leqslant\sigma}$.

Starting at a point $p \in \mathcal{S}$ in $\mathcal{F}_{\leqslant\sigma}$, let (s_1, s_2) be the corresponding point in contact space $\mathcal{U}$. If (s_1, s_2) is not a local minimum of $d(s_1, s_2)$ in $\mathcal{U}$, follow a d-decreasing path in $\mathcal{U}$ until reaching a local minimum. This path corresponds to a path in $\mathcal{S}$ which starts at p and ends at a local minimum of the function π in $\mathcal{S}$, denoted p^*. There are now two cases to consider. In the first case p^* is a local minimum of π in a local neighborhood of $\mathcal{F}$ centered at p^*. In this case p^* is necessarily an immobilizing grasp of $\mathcal{B}$ and the proof is complete. In the second case p^* is a local minimum of π along the submanifold $\mathcal{S}$, but *not* a local minimum of π in a local neighborhood of $\mathcal{F}$ centered at p^*. According to SMT, the lower half link of π at p^*, l^-, must be *non-empty* in this case.

We now show that the condition $l^- \neq \varnothing$ implies the existence of a *tunnel curve,* which starts at p^* and moves away from $\mathcal{S}$ in the free c-space $\mathcal{F}$, along a σ-decreasing path that eventually meets $\mathcal{S}$ at a lower σ point. The disc $D(p^*)$ mentioned in the definition of l^- lies in the plane based at p^* and spanned by the finger c-obstacle normals $\boldsymbol{\eta}_1(p^*)$ and $\boldsymbol{\eta}_2(p^*)$. To a first-order approximation, the normal slice at p^*, $E(p^*) = D(p^*) \cap \mathcal{F}$, is the set of tangent vectors:

$$E(p^*) = \big\{u \in T_{p^*}\mathcal{C} : \boldsymbol{\eta}_1(p^*) \cdot u \geqslant 0 \text{ and } \boldsymbol{\eta}_2(p^*) \cdot u \geqslant 0\big\},$$

where $T_{p^*}\mathcal{C} \cong \mathbb{R}^4$ is the tangent space of $\mathcal{C}$ at p^*. Note that $E(p^*)$ forms a 4D cone pointing into the free c-space $\mathcal{F}$ at p^*. Let σ^* be the σ-value of p^*. Since l^- is non-empty, all level sets of π just below σ^* have a non-empty intersection with $E(p^*)$. Since $\pi(q, \sigma) = \sigma$, this means that $E(p^*)$ has a non-empty intersection with the half-space $H = \{(q, \sigma) \in \mathcal{C} : \sigma \leqslant \sigma^*\}$. The non-empty intersection $E(p^*) \cap H$ consists of tangent vectors pointing into $\mathcal{F}$ along σ decreasing directions. Any local motion along these directions moves away from the submanifold $\mathcal{S}$ into $\mathcal{F}$ along a σ decreasing path. Moreover, one of these tangent directions must be tangent to the boundary of either $\mathcal{CB}_1$ or $\mathcal{CB}_2$ at p^* (otherwise $E(p^*) \cap H$ must be empty). The σ decreasing path starting along this direction and staying on the respective finger c-obstacle boundary has the property that its corresponding finger contact remains fixed on the object's boundary. This σ decreasing path moves away from $\mathcal{S}$, and is a *tunnel curve* described in Section 2.6.

Finally, we show that the tunnel curve which started at p^* eventually meets the submanifold $\mathcal{S}$ at a lower σ value. Along this tunnel curve, one finger retracts toward the other finger which remains fixed on the object's boundary. If the retracting finger never hits a new edge of $\mathcal{B}$, it must eventually hit the stationary finger. At this instant the pinched fingers can escape to infinity. But the initial point, p^*, lies in the connected component of p_0 within the sublevel set $\mathcal{F}_{\leqslant\sigma}$ for $\sigma \leqslant \sigma_{esc}$. An escape to infinity can therefore occur only for finger openings $\sigma \geqslant \sigma_{esc}$. Since the σ-decreasing tunnel curve started at p^* with $\sigma^* < \sigma_{esc}$, an escape to infinity is *not* possible. The tunnel curve must therefore meet the submanifold $\mathcal{S}$ at a lower σ value. □

Bibliography

[1] Thomas F. Allen, Joel W. Burdick, and Elon Rimon. Two-fingered caging of polygons via contact-space graph search. In *IEEE Int. Conf. Robotics and Automation*, pages 4183–4189, 2012.

[2] Thomas F. Allen, Joel W. Burdick, and Elon Rimon. Two-finger caging of polygonal objects using contact space search. *Robotics, IEEE Transactions on*, 31(5):1164–1179, Oct 2015.

[3] Thomas F. Allen, Elon Rimon, and Joel W. Burdick. Two-fingered caging of 3D polyhedra using contact space search. In *IEEE Int. Conf. Robotics and Automation*, pages 2005–2012, 2014.

[4] Thomas F. Allen, Elon Rimon, and Joel W. Burdick. Robust three-finger three-parameter caging of convex polygons. In *IEEE Int. Conf. Robotics and Automation*, pages 4318–4325, 2015.

[5] Haruhiko Asada. Studies on prehension and handling by robot hands with elastic fingers. *PhD dissertation, Kyoto University*, 1979.

[6] Antonio Bicchi. On the closure properties of robotic grasping. *The International Journal of Robotics Research*, 14(4):319–334, 1995.

[7] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: a review. In *IEEE Int. Conf. Robotics and Automation*, volume 1, pages 348–353, 2000.

[8] John Burkardt. gourd.obj, 2013.

[9] D.J. Cappelleri, M. Fatovic, and U. Shah. Caging micromanipulation for automated microassembly. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3145–3150, May 2011.

[10] J. S. Cheong, H. J. Haverkort, and A. F. van der Stappen. Computing all immobilizing grasps of a simple polygon with few contacts. *Algorithmica*, 44(2):117–136, 2006.

[11] F. H. Clarke. *Optimization and Nonsmooth Analysis*. SIAM Publication, 1990.

[12] Mark R Cutkosky. Mechanical properties for the grasp of a robotic hand. Technical report, DTIC Document, 1984.

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

[14] Colin Davidson and Andrew Blake. Caging planar objects with a three-finger one-parameter gripper. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2722–2727. IEEE, 1998.

[15] Rosen Diankov, Siddhartha S Srinivasa, Dave Ferguson, and James Kuffner. Manipulation planning with caging grasps. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 285–292. IEEE, 2008.

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

[17] J. Fink, M. A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *IEEE Int. Conf. Robotics and Automation*, pages 1471–1476, 2008.

[18] Mark Goresky and Robert MacPherson. *Stratified morse theory*. Springer, 1988.

[19] Wlodzimierz Kuperberg. Problems on polytopes and convex sets. In *DIMACS Workshop on polytopes*, pages 584–589, 1990.

[20] K Lakshminarayana. *Mechanics of form closure. Technical Report 78-DET-32*. ASME, 1978.

[21] Matthew T Mason and J Kenneth Salisbury Jr. *Robot hands and the mechanics of manipulation*. MIT press, 1985.

[22] Defense Advanced Research Projects Agency (DARPA). Autonomous robotic manipulation (ARM), August 2014.

[23] Van-Duc Nguyen. Constructing force-closure grasps. *The International Journal of Robotics Research*, 7(3):3–16, 1988.

[24] Allison M Okamura, Niels Smaby, and Mark R Cutkosky. An overview of dexterous manipulation. In *IEEE Int. Conf. Robotics and Automation*, volume 1, pages 255–262. IEEE, 2000.

[25] Peam Pipattanasomporn and Attawith Sudsang. Two-finger caging of concave polygon. In *IEEE Int. Conf. Robotics and Automation*, pages 2137–2142, 2006.

[26] Peam Pipattanasomporn and Attawith Sudsang. Object caging under imperfect shape knowledge. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2683–2688, May 2010.

[27] Peam Pipattanasomporn and Attawith Sudsang. Two-finger caging of nonconvex polytopes. *IEEE Transactions on Robotics*, 27(2):324–333, 2011.

[28] Peam Pipattanasomporn, Pawin Vongmasa, and Attawith Sudsang. Two-finger squeezing caging of polygonal and polyhedral object. In *IEEE Int. Conf. Robotics and Automation*, pages 205–210, 2007.

[29] Peam. Pipattanasomporn, Pawin Vongmasa, and Attawith Sudsang. Caging rigid polytopes via finger dispersion control. In *IEEE Int. Conf. Robotics and Automation*, pages 1181–1186, May 2008.

[30] Jean Ponce, Steve Sullivan, Attawith Sudsang, Jean-Daniel Boissonnat, and Jean-Pierre Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *The International Journal of Robotics Research*, 16(1):11–35, 1997.

[31] Franz Reuleaux. *Theoretische Kinematic*. Ripol Classic Publishing House, 1875. Translated as *Kinematics of Machinery*, New York: Dover, 1963.

[32] E. Rimon and A. Blake. Caging 2D bodies by 1-parameter two-fingered gripping systems. In *IEEE Int. Conf. Robotics and Automation*, pages 1458–1464, 1996.

[33] E. Rimon and A. Blake. Caging planar bodies by one-parameter two-fingered gripping systems. *Int. J. Robot. Res.*, 18(3):299–318, 1999.

[34] E. Rimon and J. Burdick. On force and form closure for multiple finger grasps. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 2, pages 1795–1800 vol.2, Apr 1996.

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

[36] A. Rodriguez and M. T. Mason. Two finger caging: squeezing and stretching. *Algorithmic Foundation of Robotics VIII*, pages 119–133, 2009.

[37] A. Rodriguez, M.T. Mason, and S. Ferry. From caging to grasping. *Proc. Robotics: Science and Systems*, 2011.

[38] Alberto Rodriguez, Matthew T Mason, and Steve Ferry. From caging to grasping. *Int. J. Robot. Res.*, 31(7):886–900, 2012.

[39] J Kenneth Salisbury and B Roth. Kinematic and force analysis of articulated mechanical hands. *Journal of Mechanical Design*, 105(1):35–41, 1983.

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

[41] A. Sudsang, J. Ponce, M. Hyman, and D. J. Kriegman. On manipulating polygonal objects with three 2-DOF robots in the plane. In *IEEE Int. Conf. Robotics and Automation*, pages 2227–2234, 1999.

[42] A. Sudsang, F. Rothganger, and J. Ponce. Motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Transactions on Robotics and Automation*, 18(4):550–562, 2002.

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

[44] A Frank van der Stappen, Chantal Wentink, and Mark H Overmars. Computing immobilizing grasps of polygonal parts. *The International Journal of Robotics Research*, 19(5):467–479, 2000.

[45] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi. A new grasping by caging solution using eigen-shapes and space mapping. In *IEEE Int. Conf. Robotics and Automation*, pages 1566–1573, 2013.

[46] Weiwei Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi. Grasping by caging: A promising tool to deal with uncertainty. In *IEEE Int. Conf. Robotics and Automation*, pages 5142–5149, 2012.

[47] Zhidong Wang and V. Kumar. Object closure and manipulation by multiple cooperating mobile robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 394–399, 2002.

[48] Dmitry Zarubin, Florian T Pokorny, Marc Toussaint, and Danica Kragic. Caging complex objects with geodesic balls. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2999–3006. IEEE, 2013.