博士論文（要約）

Efficient Caging Planning Under Uncertainty Based on Configuration Spaces of Target Object and Finger Formation

（対象物体と指配置のコンフィグレーション空間を用いた不確かさを扱える効率的なケージング計画）

万　偉偉

Abstract

This thesis discusses the planning algorithms of a novel type of grasping closure – namely caging. In the author's view, this is the first literature which systematically discusses caging planning algorithms and their applications in robotics. Although the idea of caging is not proposed by the author himself, he further develops the geometric definition of caging and makes it pragmatic.

The thesis contributes in three aspects. Firstly, in caging theory, the thesis initially explains the relationship between caging and traditional research in grasping. Namely, caging is the extension of immobilization. Secondly, in caging planning algorithms, the thesis initially employs caging to deal with uncertainty. It on the one hand proposes efficient algorithms to deal with caging test while on the other hand further proposes efficient algorithms to deal with caging optimization. Both caging test algorithms and caging optimization algorithms are explored in the configuration space of target object and the configuration space of finger formation. Thirdly, in the aspect of applications, the thesis applies the proposed caging planning algorithms to robotic hands and multi-robot cooperative transportation. It discusses how to select proper algorithms according to requirements of real-world applications. Results show that the algorithms are not only robust to various uncertainty but also helpful to reduce the number of fingers or mobile robots.

Main texts of the thesis are divided into four parts. The first three parts corresponds to the three contributions. The first part is the basic concepts of caging. It reviews time-of-the-art progresses in robotic manipulation and presents the contribution in theoretical aspect, namely relationship between caging and traditional research topics in grasping. The remaining two parts discuss caging by using two different spaces. One is the configuration space of target object and the other is the configuration space of finger formation. Details on how to solve the caging problems, namely the caging test problem and the caging optimization problem, in those two spaces are discussed respectively in these two parts. These two parts also involve applications of the caging algorithms like a distributed end-effector, a gripping manipulator and multi-robot cooperation.

The configuration space of target object and the configuration space of finger formation essentially equal with each other by a linear transformation. They actually provide different metrics to measure the robustness of caging. The fourth part of the thesis discusses the relationship of the algorithms in the two spaces and how to choose a proper algorithm according to mechanical structures and specific tasks. It also summarizes the thesis and proposes potential future directions.

The concepts and algorithms proposed in this thesis can efficiently solve 2D manipulation problems in the presence of uncertainty. The author believes that caging is a promising tool to deal with perception and control uncertainty. He would like to spread this tool and explore more about this tool in both theory and application aspects in the future.

Contents

List of Figures

Chapter 1

Introduction

1.1 Robotic Manipulation and Its Difficulties

Robotic manipulation involves two aspects. One is the working subject, namely a robot. The other one is the working motion, namely manipulation or rearranging the world by hands. Prof. Matthew T. Mason discussed a lot about how should a robot rearrange the world with robotic hands in his book "Mechanics of Robotic Manipulation"[Mason, 2001]. Mechanics is one essential problem in robotic manipulation. However, it is never the unique one. Generally speaking, a pragmatic robotic manipulation suffers from either difficulties from mechanism, sensing and mechanics. We can summarize them as following.

- **Mechanism:** Structures and organizations of a robotic hand
- **Sensing:** Perception and understanding of target objects
- **Mechanics:** Forces exerted by fingers and balances of those forces

Mechanism relates to structures of robotic hands. It concentrates on kinematics and actuation, especially how to select and organize mechanical components to obtain better control performance. Prof. Nancy Pollard in his course "Hands: Design and Control for Dexterous Manipulation"[Pollard, 2010] gave a good review of popular hands and related design issues. In most cases, robotic hands are designed according empirical requirements of specific tasks or designed by mimicking certain biological creatures. These hands are in front of difficulties like complicated control and expensive actuators. Low-cost, high-robustness and general-purpose robotic hands remains a popular research topic. One representative work in this topic is [Rodriguez and Mason, 2013](fundamentally based on [Rodriguez and Mason, 2012a]). It won the IEEE ICRA2013 best student paper award.

Sensing relates to perceiving and understanding target objects. Most manipulation systems install two kinds of sensors. One kind is global sensors which play the role of human eyes. The other kind is local sensors which play the role of tactile sensation. Take the WillowGarage PR2 robot[WillowGarage, 2012] for example. Before performing manipulation

tasks, the PR2 robot firstly perceive positions and geometric information of target objects with global sensors on its head. Then, when stretching out its hand for grasping, the PR2 robot exerts forces by using tactile sensors inside its gripping hand. The most important problems that relate to these sensors are their precisions and costs. Low-cost perception devices like KINECT[Microsoft, 2012][1] could have as much as 50mm errors while precise perception devices like a laser ranger could cost thousands of dollars or even overtake the cost of a robotic hand itself. Researchers are still devoting themselves to struggling with the difficulties caused by sensing devices. Two popular solutions, namely reducing the number of necessary sensors and improving the performance of manipulation under various uncertainties take up most of nowadays researches.

Mechanics relates to mathematical and physical analysis of manipulation. It includes but is not limited to form or force closure (see Chapter 7 of [Mason, 2001]). Given the geometric information of target objects, mechanics study calculates the formations that should be shaped to manipulate the target objects and calculates the forces that should be exerted to operate the target objects. The difficulty from mechanics is not independent. It is deeply coupled with mechanism and sensing. An under-actuated mechanism suffers from kinematics and thus changes analysis of mechanics. A noisy perception device offers uncertain surface normals and thus changes conditions of force or form closures. Lots of state-of-the-art works discuss how to optimize mechanics against mechanism and sensing.

None of the three difficulties are independent. They interplay each other and further improve the difficulties of robotic manipulation. Our group design a distributed end-effector to challenge these three difficulties. This distributed end-effector is a background work of my caging topic and I will briefly introduce this it in the next section.

1.2 The Distributed End-effector

Our group proposes a distributed end-effector to challenge the difficulties of mechanism, sensing and mechanics in robotic manipulation. During the design of this end-effector we try to avoid explicit contact between fingers and target object as well as try to reduce the number of motors as much as possible. Fig.1.1 illustrates our concept.

Novelty of this end-effector lies in the following four aspects. (1) Generality: As can be seen from Fig.1.1, our distributed end-effector is installed to a base robotic arm to transport target objects from place to place. The target objects could have any rigid shapes, from commodity packages to cups and plates. In this way, the distributed end-effector is more general than traditional robotic palletizers. (2) Conciseness: We install only one x-y-θ actuator to the end-effector to lower its cost. This actuator will attach, actuate and detach each of those four "hanging" fingers sequentially and each finger can be actuated "distributedly". (3) Fully distributed control: Each finger has some permanent magnets installed to its top and each finger is connected to the "palm" of the end-effector by the magnetic forces exerted

[1]News of KINECT2 has been released recently. KINECT2 is Time-of-Flight based depth sensor. It is more precise as well as low cost.

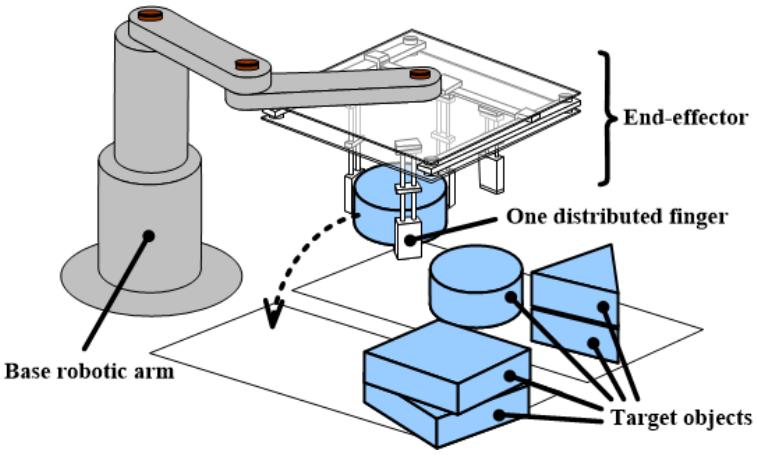


Figure 1.1: The concept of our distributed end-effector.

by those permanent magnets. Connection by magnets offer the freedom that when a finger is attached by the x-y-θ, it can be dragged and rotated freely in the "palm" plane. Each finger is controlled fully distributedly. (4) Prismatic finger body and nails: Each finger has a prismatic body and an inserting nail. The prismatic bodies help to enlength fingers without taking too much space while the inserting nails help to support objects during picking-up and transporting procedure. After perceiving the shape of a target object, the fingers stretch out their bodies to the bottom of the object and insert their nails underneath the object to avoid direct contact. In this way, we no longer need to consider about object materials or analyze contact frictions as long as the links of fingers constrain target objects and the nails have enough power to support object mass. Fig.1.2 shows in detail the mechanism of our x-y-θ actuator and a transporting procedure of the end-effector.

Along with these novelties, we encounter several problems like how to configure motors of the x-y-θ actuator, how to set permanent magnets and how to set the shape and strength of finger nails, etc. These problems relate heavily to mechanical design. However, besides these mechanical problems, we have an essential issue which relates to grasp synthesis. Say, given the shape of an object, how many fingers do we need and what kind of finger formation should we choose to constrain target objects? This problem looks like a traditional grasping problem, or more exactly looks like a form-closure problem since we do not consider frictions explicitly. Unfortunately, it is different. Form closure requires the equilibrium of wrenches. In contrast, the fingers in our case do not explicitly exert forces and we cannot build equations base on the convexity of form or force closure hulls. This problem is pure geometric. It is actually a different kind of closure which relates to caging.

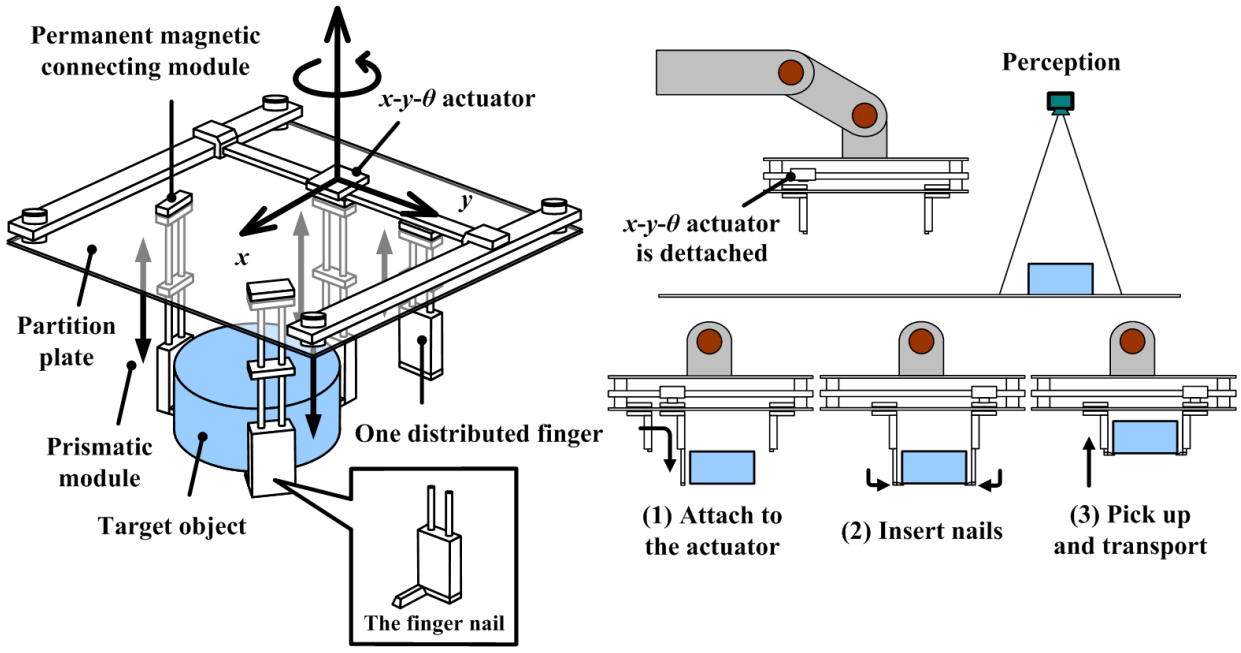


Figure 1.2: Mechanism of the x-y-θ actuator and a transporting procedure.

1.3 Controlling the End-effector by Caging

The first formal proposal of a caging problem is from Kuperberg's paper[Kuperberg, 1990] in 1990. In this paper, Kuperberg proposed the following question.

> Let P be a polygon in the plane, and let C be a set of n points 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.

Kuperberg is a mathematician and he is mathematically strict in this proposal. Assume that any planar objects can be approximated by polygons, we can have a more general form. Inputs to the general form is the geometric shape or boundary clouds of a planar object. Outputs of it is a formation of capturing points that cages the geometric shape so that it can never go to infinity. The grasp closure issue we encountered in the end-effector is a variation of this general form. Actually, our issue is more difficult due to the limitation of hardware and uncertainty from engineering noises. Fig.1.3 illustrates the similarities and differences between our issue and Kuperberg's caging problem. The upper part of Fig.1.3 expresses Kuperberg's caging problem. It shows a caging formation which cages a circular object. In this case, the object is caged and cannot escape from the cage formed by the point finger formation. The lower part of Fig.1.3 expresses the caging issue of our end-effector. In this case, we need to consider extra problems like how many fingers should we employ to reduce

mechanical costs and how to deal with various uncertainty due to perception and control noises.

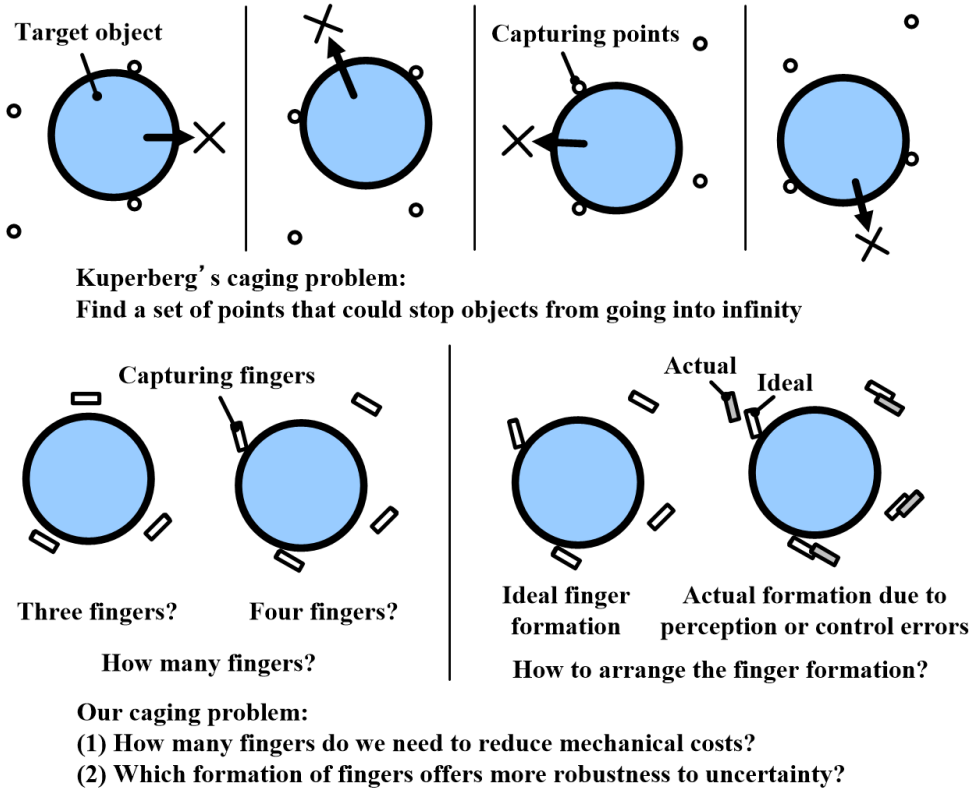


Figure 1.3: Similarities and differences between our issue and Kuperberg's caging problem.

Our issue has the same input as Kuperberg's proposal, namely the geometric shape of a planar object. It also has the same output, namely a formation of capturing finger positions that cages the target object[2]. Despite the similarities, our issue inherently suffers from engineering uncertainty. We not only need to consider whether the finger formation can capture or cage a target object, but also need to consider uncertainty caused by noises from the perception device and noises from control. The uncertainty makes our issue more difficult. Fig.1.4 shows in detail how Kuperberg's caging problem changes in the presence of perception and control noises.

When noises appear during perception procedure, the perceived object boundary could be dramatically different from its groundtruth shape. The difference may become even more dramatic after certain post-processing procedures. The upper part of Fig.1.4(a) illustrates

[2]Readers may notice that in our case finger have shapes. Surely finger shapes can simplify caging. For instance, our end-effector has rectangular finger shape and it can cage target objects more easily than point fingers. Nevertheless, let us temporarily take all fingers as point fingers. This is a sound assumption. Shaped fingers are super sets of point fingers. If an object can be caged by point fingers, it will sure to be caged by shaped ones.

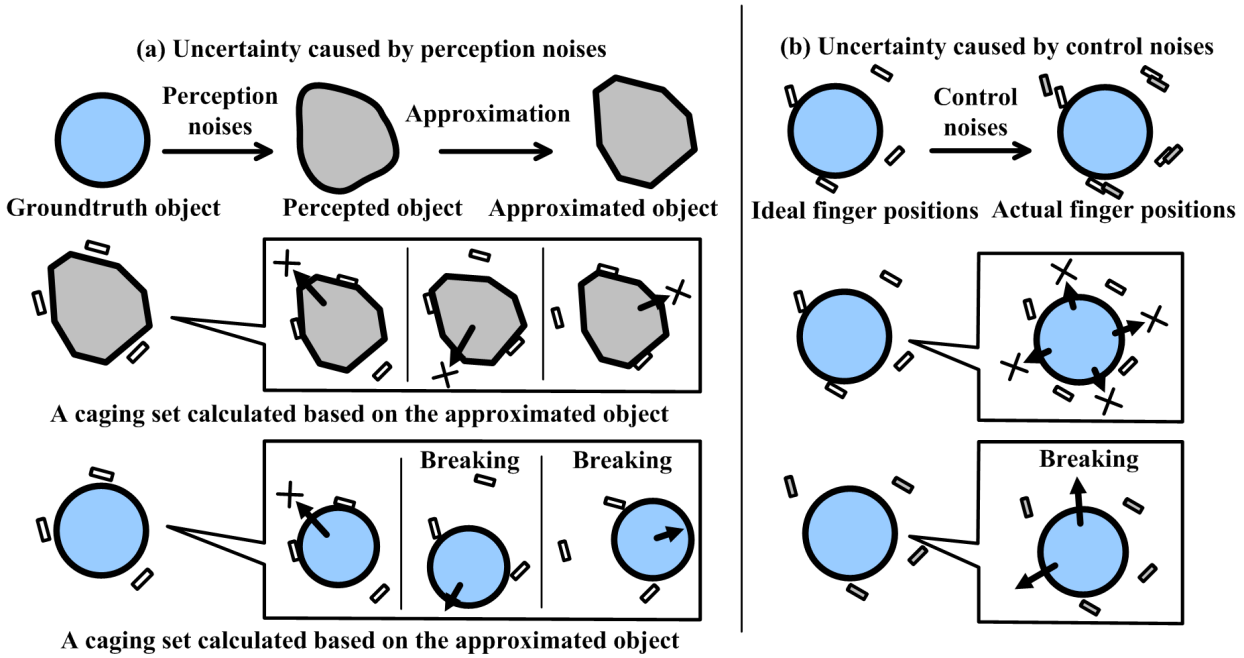


Figure 1.4: Our issue inherently suffers from uncertainty caused by engineering noises.

the perception noises and distortions in the shape. In this illustration, a groundtruth circular object becomes an irregular polygon after noisy perception and approximation. These noisy perception and approximation cause a fatal failure that the caging set calculated based the approximated polygon cannot cage the groundtruth circular object. The lower part of Fig.1.4(a) illustrates the fatal failures and caging breaking.

The same failure happens when control noises appears. Accumulation of noises from motor encoders, belt gears and permanent magnetic connecting modules could degenerate control and drive fingers to unexpected positions. Even though we can find a formation of finger positions to cage a target object, the caging may break due to noises during finger control and actuation. The upper part of Fig.1.4(b) illustrates the degenerated control and wrongly actuated fingers. These wrongly actuated fingers may either squash the target object or result into caging breakings. The lower part of Fig.1.4(b) illustrates a failure case (caging breaking).

Moreover, besides the noisy uncertainty we also need to take into account some other factors like length of the finger nails. In one word, our issue is a similar but complicated version of Kuperberg's caging problem. However, **there is no perfect solution to Kuperberg's caging problem, far from ours**. Consequently, I dive into the following research topic.

> How can we deal with the caging problem and apply it to our end-effector? Or more generally, how can we deal with the caging problem and apply it to robotic manipulation?

This thesis originates from this topic. The caging problem can indeed be divided into two sub-problems. The first one corresponds to Kuperberg's proposal, namely finding a set of caging formations or simply finding a caging set by solving the **caging test problem**. The second one corresponds to our newly encountered issue, namely finding a caging formation that is most robust to uncertainties or simply the **caging optimization** problem. I will generally call the **caging test problem** and the **caging optimization problem** by using **the caging problems** in the context.

In the problem of **caging test**, we need to develop a **caging test** algorithm that can test whether a specific finger formation cages the target object. We will need to find all the caging formations that can pass the caging test algorithm. In the problem of **caging optimization**, we need to develop an algorithm by using robust caging, namely find a measurement, evaluate the robustness of a caging formation with the measurement and picking out a caging formation that has satisfying robustness. Fig.1.5 illustrates the relationship of those caging problems.

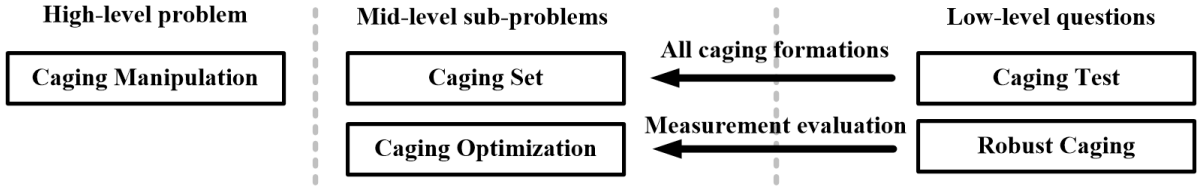


Figure 1.5: The caging problems of this thesis.

This thesis explores the caging problems. It summarizes my study in caging algorithms and presents some applications based on those study. The title of the thesis is named "caging planning" following "grasp planning". It borrows some ideas from path planning literature to solve the caging problems. I may not say the caging problems has been perfectly solved, but I believe my work can contribute to researches in related fields.

1.4 Organization of the Thesis

This thesis is composed of eight chapters. Besides the introduction chapter and the conclusion chapter, the remaining six chapters can be divided into four parts. They are,

Part I, Basic Concepts of Caging. This part introduces the basic concepts of caging and its recent development. It is not only a literature review but also includes some of my proposals that fill up the gap between caging and traditional robotic grasping research. My contribution in this part is the demonstration of both traditional grasping concepts and caging in the **configuration space of target object,** $\mathcal{C}^{\mathsf{obj}}$. I visualize the relationship between caging and traditional research in grasping with $\mathcal{C}^{\mathsf{obj}}$.

This Part I includes Chapter 2.

Part II, Caging in $\mathcal{C}^{\mathsf{obj}}$ and Its Applications. This part discusses in detail of how to deal with the caging problem in $\mathcal{C}^{\mathsf{obj}}$ and presents some applications based on my discussion. $\mathcal{C}^{\mathsf{obj}}$ can be seen as a tool to make caging analysis easier. By employing $\mathcal{C}^{\mathsf{obj}}$ we can review the caging problems from a different viewpoint and solve them intuitively.

Specifically, in this part, I firstly revisit, improve and implement a work by Prof. Jeff Erickson[Erickson et al., 2003][Erickson et al., 2007]. Erickson's idea is quite smart and my implementation improves his idea. However, both Erickson's idea and my implementation are limited to three-finger hands and known finger positions. How to push though those limitations and extend my implementation to general cases become a key issue. I propose an algorithm to solve this issue by fixing fingers alternatively and further reduce the computational complexity of that algorithm by decomposing caging into **translational caging** and **rotational constraints**. The performance of my algorithm are demonstrated and evaluated by the robustness of finger formations with WEBOT simulation software.

Applications in this part includes the distributed end-effector and a multi-robot cooperative transportation system. The applications work well with 2D convex objects. However, convexity is an inherent limitation and the algorithms in $\mathcal{C}^{\mathsf{obj}}$ can only work with 2D convex objects. Objects with concave boundaries invalidate the algorithms easily. This drawback motivates me to explore into another tool, say, **configuration space of finger formation, $\mathcal{C}^{\mathsf{frm}}$**.

This Part II includes Chapter 3 and Chapter 4.

Part III, Caging in $\mathcal{C}^{\mathsf{frm}}$ and Its Applications. This part analysizes the caging problems in $\mathcal{C}^{\mathsf{frm}}$. Like $\mathcal{C}^{\mathsf{obj}}$, $\mathcal{C}^{\mathsf{frm}}$ can also be seen as a tool to make easier caging analysis. The motivation that drive me to this tool is from two aspects. For one thing, I hope to make the caging algorithm work with any 2D shape, not only objects with either convex boundaries, but also concave boundaries, 1-order or high-order boundaries. For the other, I hope to make the caging algorithm complete as well as rapid. This is difficult in $\mathcal{C}^{\mathsf{obj}}$ since the complete algorithm in $\mathcal{C}^{\mathsf{obj}}$ may have a time cost as much as order nine. Therefore, approximation of the complete algorithm is employed in $\mathcal{C}^{\mathsf{obj}}$ to by a combination of **translational caging** and **rotational constraints**. However, the combination is not from strict mathematically analysis and lacks completeness. In $\mathcal{C}^{\mathsf{obj}}$, completeness and rapidness are reciprocal. High completeness implies low rapidness while high rapidness implies low completeness. Therefore, I choose to change to $\mathcal{C}^{\mathsf{frm}}$. Making both complete and rapid algorithms to the caging problems is the second motivation that drives me to $\mathcal{C}^{\mathsf{frm}}$.

In this part III, I will firstly discuss in detail why to change from $\mathcal{C}^{\mathsf{obj}}$ to $\mathcal{C}^{\mathsf{frm}}$. Employing $\mathcal{C}^{\mathsf{obj}}$ instead of $\mathcal{C}^{\mathsf{frm}}$ changes the center of my algorithm from target objects to fingers. We no longer need to bother with specific object shape features like concavity and convexity. This change exactly caters the expectation in the first aspect of my motivation. Then, I introduce the **space mapping** idea which caters the expectation in the second aspect of my motivation. *Raw space mapping* and especially its faster version, the *improved space mapping*, make it possible to update the whole space of $\mathcal{C}^{\mathsf{frm}}$. In other words, the algorithm is complete. At the same time, we can quickly find the **caging sets** in the updated $\mathcal{C}^{\mathsf{frm}}$ and locate an optimized caging formation. In other words, the algorithm is rapid.

My complete and rapid algorithm in this part is applied to the design and implementation of a gripping hand. The algorithm plays important roles in both design and implementation procedures. During design, the $\mathcal{C}^{\mathsf{frm}}$ algorithm is employed to simplify and evaluate deign models. During implementation, the $\mathcal{C}^{\mathsf{frm}}$ algorithm is employed to control the hand to cage and grasp objects. The design and implementation procedure could demonstrate advantages of caging.

This Part III includes Chapter 5 and Chapter 6.

Part IV, In-depth the Relationship Between $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$. $\mathcal{C}^{\mathsf{frm}}$ is sometimes a more powerful tool comparing with $\mathcal{C}^{\mathsf{obj}}$. However, it is unwise to discuss which is better. Both algorithms in $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$ have their advantages and disadvantages. The fourth part of the thesis proves that at different orientations $\mathcal{C}^{\mathsf{frm}}$ is the linear transformation of $\mathcal{C}^{\mathsf{obj}}$. Consequently, the metrics used in $\mathcal{C}^{\mathsf{frm}}$ and $\mathcal{C}^{\mathsf{obj}}$ are different. Both the two tools and their correspondent algorithms have reasons to exist. They therefore should be treated equally.

Actually, the two tools and their algorithms correspond to different solutions of **geometric modeling**. The algorithm in $\mathcal{C}^{\mathsf{obj}}$ uses **wireframe modeling** while the algorithm in $\mathcal{C}^{\mathsf{frm}}$ uses **solid modeling**. Both modeling technology plays important roles in **geometric modeling** and either algorithms in $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$ should exist. I treat them equally and compile them into Part II and Part III of this thesis.

In real world, the algorithms in $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$ should be chosen according to mechanical structure of robots and tasks. If all capture points are distributed and target objects are convex (like the distributed end-effector and multi-robot cooperative transportation), it is wise to do caging planning with the algorithms in $\mathcal{C}^{\mathsf{obj}}$. If capture points can be represented by certain formations or target objects have various shapes (like the gripping hand), it is wise to do caging planning with the algorithms in $\mathcal{C}^{\mathsf{frm}}$.

This Part IV discusses these in-depth relationships. It includes Chapter 7.

The last chapter, Chapter 8, concludes the thesis. It firstly summarizes the whole thesis, especially the algorithms in $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$. Then, this chapter makes clear the contributions. In the third sub-section of this chapter, I discuss about some future directions in algorithms and applications aspects respectively. In the aspect of algorithms, future works could be the discussion of 2.5D/3D objects and the discussion of how to pre-define representative finger formations. In the aspect of applications, future works could be the deployment onto macro/nano manipulation and in-hand re-grasping systems.

Finally, Fig.1.6 shows all the works and their relationships. The sub-figures in Fig.1.6 are representatives of those works. We will see their details throughout the remaining contents.

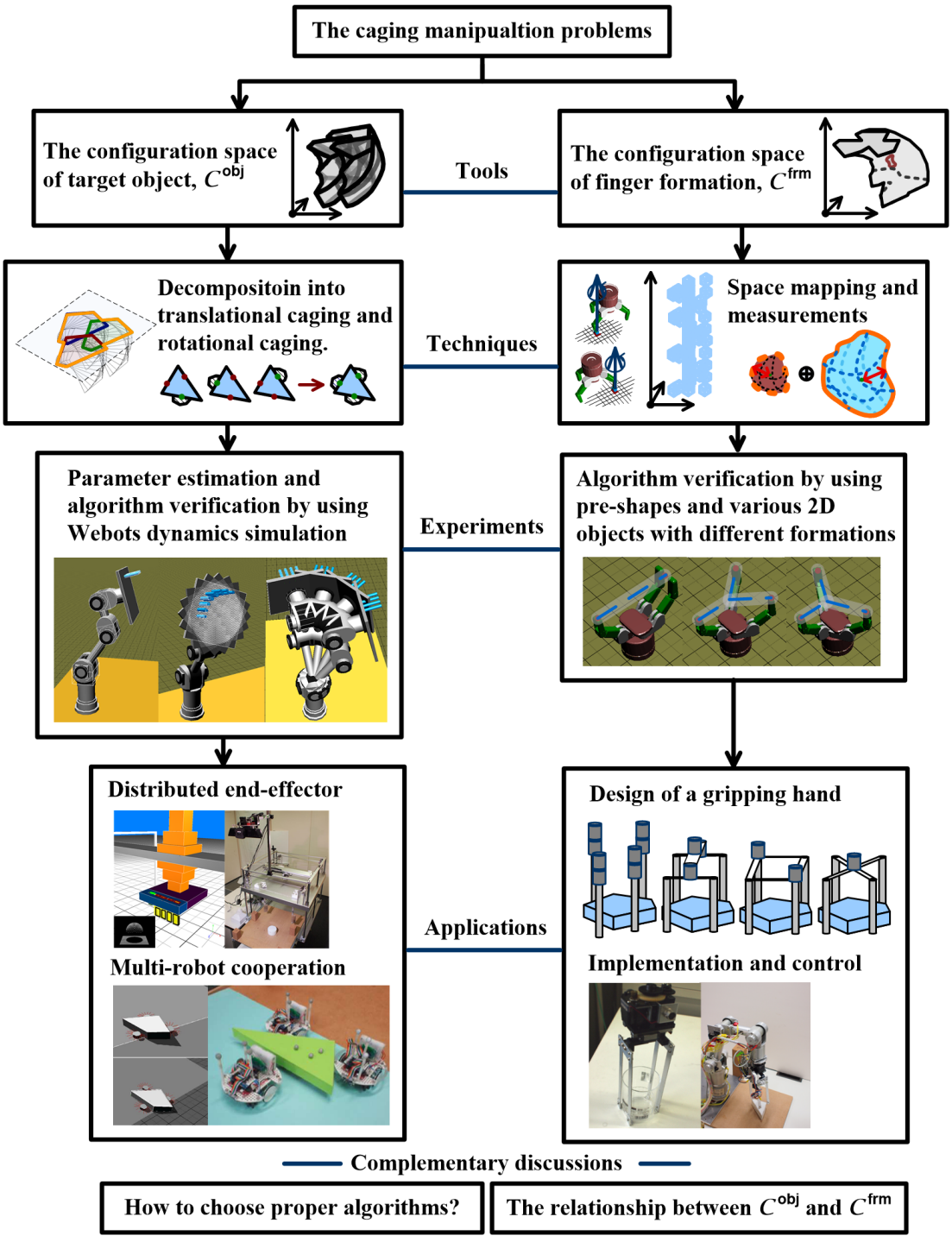


Figure 1.6: Organization of the works included in this thesis.

Part I

Basic Concepts of Caging

Chapter 2

Caging is the General Form of Grasping

2.1 Related Research Topics in Grasping

Before going deeper into caging, let us review some related researches and concepts in grasping. They are (1) force and form closure (2) immobilization and (3) grasping optimization.

2.1.1 Force closure and form closure

The force closure problem is one of the most fundamental problems in grasping. Basically, force closure describes a state in **wrench space**. Detailed reviews and discussions of the force closure problem can be found in Nguyen's publications[Nguyen, 1986a][Nguyen, 1986b]. I will not repeat those mathematically deductions here but would like to visualize the concept with figures. Like its name, **force closure means the wrench vectors exerted by fingers enclose the origin point of wrench space**. Or namely, the origin point of **wrench space** is enclosed by a convex hull which is spanned by wrenches exerted by fingers. Fig.2.1 visualizes this concept with two examples.

There are two points to explain about this figure. The first one is the forces exerted by those point fingers. Each point finger can exert not only normal forces along surface normals of target objects, but also friction forces along tangential directions. The synthesis of normal forces and friction forces is in a region. The areas in the middle of those orange, green and purple segments in the center part of Fig.2.1 illustrate this kind of regions. Since one finger can exert forces in a region, it is possible to ensure force closure with only two fingers. The first example of Fig.2.1 demonstrates this case.

The second point is **wrench space**. A wrench is a force plus a torque. Therefore, a wrench is a six dimensional vector shown in expression (2.1).

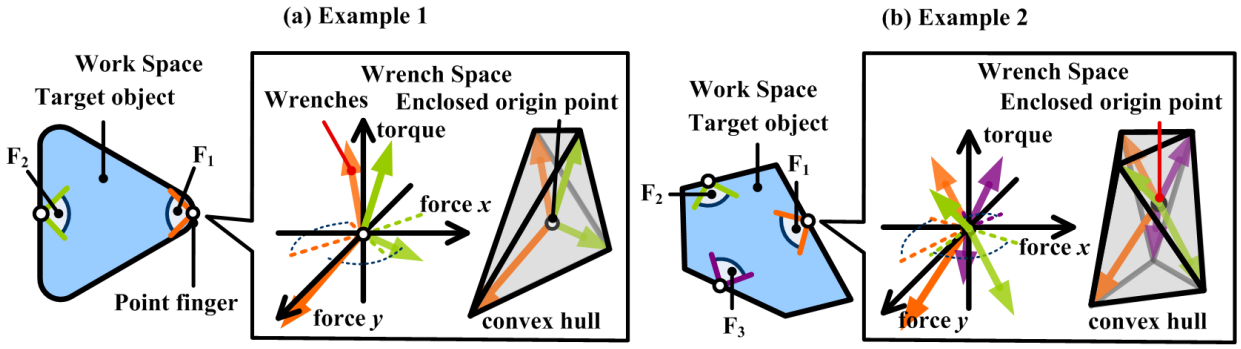


Figure 2.1: Two examples of force closure.

$$\boldsymbol{w_i} = \begin{pmatrix} \boldsymbol{F_i} \\ \boldsymbol{\tau_i} \end{pmatrix} = \begin{pmatrix} (F_{i_x}, F_{i_y}, F_{i_z})' \\ (\tau_{i_x}, \tau_{i_y}, \tau_{i_z})' \end{pmatrix} = \begin{pmatrix} \boldsymbol{F_i} \\ \boldsymbol{r_i} \times \boldsymbol{F_i} \end{pmatrix} = \begin{pmatrix} (F_{i_x}, F_{i_y}, F_{i_z})' \\ (r_{i_x}, r_{i_y}, r_{i_z}) \times (F_{i_x}, F_{i_y}, F_{i_z})' \end{pmatrix} \tag{2.1}$$

Here $\boldsymbol{F_i}$ is the force exerted by point fingers. According to foregoing explanation, this force is the synthesis of normal force and friction force. $\boldsymbol{r_i}$ is a position vector which indicates the relative position between the force and rotational center. All the six dimensional wrenches $\boldsymbol{w_i}$ constitute a six dimensional vector space named **wrench space**. In 2D case, F_{i_z} is always zero and $\boldsymbol{w_i}$ is always a three dimensional vector $\boldsymbol{w_i} = (F_{i_x}, F_{i_y}, \tau_{i_z})$. Therefore, the **wrench space** of a 2D object has three dimensions. The contents in the frames of Fig.2.1 illustrate the 3D **wrench spaces**.

The wrenches exerted by those point fingers in Fig.2.1 are rendered with the same colors as their work space correspondence. **These wrenches span convex hulls which enclose their origin points. Enclosing the origin points indicates that the fingers in both examples of Fig.2.1 form force closures.**

Force closure ensures grasping. However, it is impractical since materials and frictions of target objects are difficult to be perceived and modelled. Therefore, researchers usually discusses force closure without considering frictions. That is the concept of form closure.

Form closure was first proposed by Reuleaux in the year 1875(see [Bicchi, 1995] for a detailed review of those historical work). When referring to form closure, researchers have a common assumption that the point fingers cannot exert friction force. This assumption means that the $\boldsymbol{w_i} = (F_{i_x}, F_{i_y}, \tau_{i_z})$ in the **wrench space** of a 2D object becomes $\boldsymbol{w_i} = (N_{i_x}, N_{i_y}, \tau_{i_z})$. Namely the synthesized force $\boldsymbol{F_i}$ is reduced into the simple normal force $\boldsymbol{N_i} = (N_{i_x}, N_{i_y}, 0)$. The assumption is quite practical since researchers no longer need to consider about frictions to ensure form-closure grasp.

However, form closure causes new problems. It requires a lot more fingers [Mishra et al., 1987] and it is not applicable to circular objects. Reader may compare Fig.2.1 and Fig.2.2 for example. The same point fingers and the same target object ensure force closures in Fig.2.1(a), however, they cannot ensure form closures in Fig.2.2(a). Without

friction, the wrenches exerted by the point fingers cannot span convex hulls to enclose the origin points. [Mishra et al., 1987] proved that (1) if the DoF (Degree of Freedom) of an object is n_{dof}, we need at least $n_{\text{dof}} + 1$ point fingers to ensure form closure and (2) if the object is a circle, we can never ensure form closure with point fingers. For a 2D object whose $n_{\text{dof}} = 3$, we need at least 4 point fingers to ensure form closure. Therefore, the point fingers in Fig.2.2 are insufficient and they cannot ensure form closures.

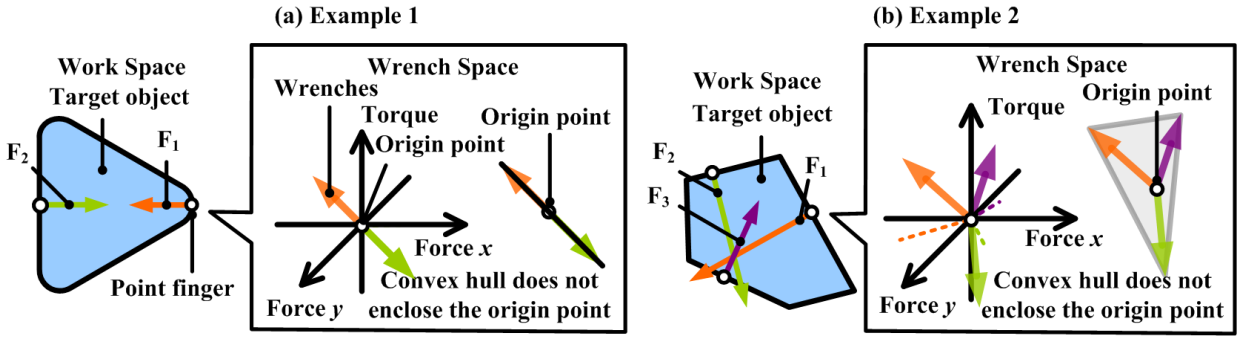


Figure 2.2: Two examples of non-form closure.

Mishra's theory seems to be contradictory to our common sense. Intuitively, people hold an feeling that the second example of Fig.2.2 can be successfully grasped by those point fingers. However, Mishra's theory shows that it would fail. Is there anything wrong with form closure? Elon Rimon tries to explain the contradiction by considering surface curvature of objects. His theory starts from [Ponce et al., 1995], [Rimon and Burdick, 1996] and [Rimon and Blake, 1996]. It becomes mature in [Rimon and Burdick, 1998a] and [Rimon and Burdick, 1998b]. The work is extended to 3D objects in [Rimon, 2001]. According to Rimon's theory, the contradiction between form closure and people's common sense was caused by surface curvatures of target objects. Take Fig.2.3 for example. In traditional definition of form closure, none of the cases in Fig.2.3 are form closure. Rimon improves the traditional definition by considering surface curvatures and introduces the concept of **2nd order form closure**. In Rimon's theory, the traditional form closure is named 1st order form closure since it only considers surface normals while his new closure is named 2nd order form closure since it further takes the derivative of surface normals, namely the curvatures of object surfaces into account. 2nd order form closure considers surface curvatures of target objects and therefore dissolves the contradiction. The surface curvatures at contact points in Fig.2.3(a) go outwards the contact tangent lines. They neither ensure 1st order nor ensure 2nd order form closure. The surface curvatures of the other cases in Fig.2.3(b) either run parallel to or go inwards contact tangent lines. Although they are not 1st order form closures. They fulfills Rimon's 2nd order definition and therefore can ensure successful grasp of target objects.

Besides **2nd order form closure**, another explanation of the contradiction is **infinitesimal motions**[Czyzowicz et al., 1999][van der Stappen, 2005][Cheong et al., 2006]. In force

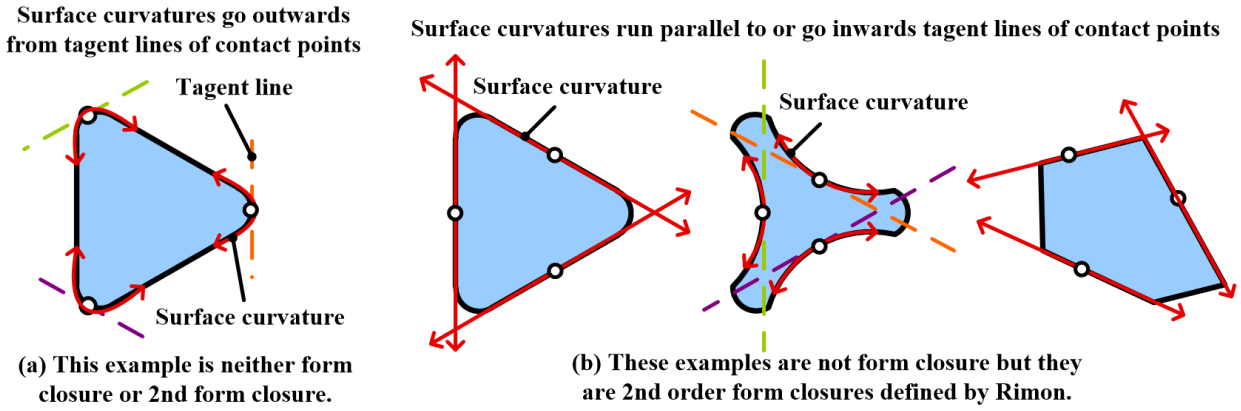


Figure 2.3: 2nd order form closure and surface curvatures.

closure, three fingers are enough to block both **finite motions** and **infinitesimal motions** since friction forces exist at every contacts. However, in form closure, three fingers can only block **finite motions** and target objects may oscillate infinitesimally due to the lack of friction. We may need at least 4 fingers to block both **finite motions** and **infinitesimal motions** and to ensure form closure. This conclusion corresponds to Mishra's $n_{\mathsf{dof}}+1$ theory.

Fig.2.4 illustrates the meanings of **finite motions** and **infinitesimal motions**. It is too strong for fingers to block both finite and **infinitesimal motions** in form closure and that's the reason why the contradiction between form closure and our common sense exists.

Rimon's 2nd order form closure can be recognized as an attempt to deprive **infinitesimal motions** from the definition of 1st order form closure. That is, even if there exists **infinitesimal motions**, a target object could be grasped. However, Rimon's analysis is quite complicated. A more concise way to deprive **infinitesimal motions** is to define closures from another viewpoint, namely immobilization.

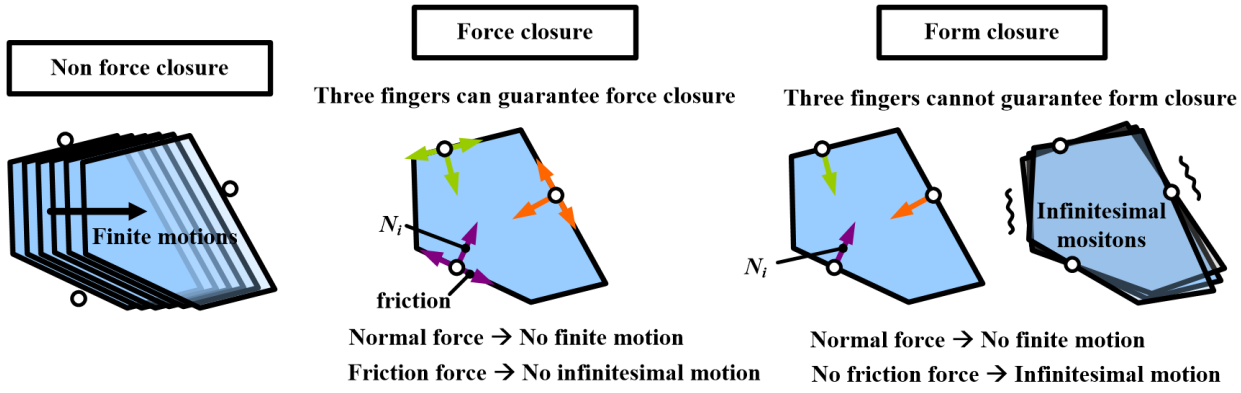


Figure 2.4: Form closure requires more fingers to constrain infinitesimal motions.

2.1.2 Immobilization

Either the definition of force or form closures relate heavily to forces and wrench space. In contrast, immobilization do not analyze forces. Jurek Czyzowicz[Czyzowicz et al., 1999] defines immobilization as following.

> The set of points I is said to immobilize a planar shape P if any rigid motion of P in the plane forces at least one point of I to penetrate the interior of P.

Comparing with force and form closure which require enclosing the origin point of wrench space, the definition of immobilization is pure geometric. It does not result into problems of surface curvatures or infinitesimal motions. **All we need to ensure is that the target object, in its configuration space, is at a fixed single configuration.** Instead of Jurek's definition, let us view the immobilization problem in **configuration space**. More exactly, we should call it **the configuration space of target object** and use symbol $\mathcal{C}^{\text{obj}}$ to indicate it. $\mathcal{C}^{\text{obj}}$ was originally a $\mathbb{R}^2 \times \mathcal{S}$ topology space. This $\mathbb{R}^2 \times \mathcal{S}$ topology space is homeomorphic to the three-dimension Euclidean space $\mathbb{R}^3$ (see Chapter 3 of reference [Choset et al., 2005] for more formal definition). We can therefore use $\mathbb{R}^3$ rather than $\mathbb{R}^2 \times \mathcal{S}$ to represent $\mathcal{C}^{\text{obj}}$ and to simplify the deductions. The first two dimension of $\mathcal{C}^{\text{obj}}$ denote the position of target objects and the third dimension of $\mathcal{C}^{\text{obj}}$ denotes the orientation of target objects. A point in $\mathcal{C}^{\text{obj}}$ is called a configuration and it corresponds the position and orientation of the target object in work space. Fig.2.5 shows the correspondences between a target object in work space and a configuration in $\mathcal{C}^{\text{obj}}$.

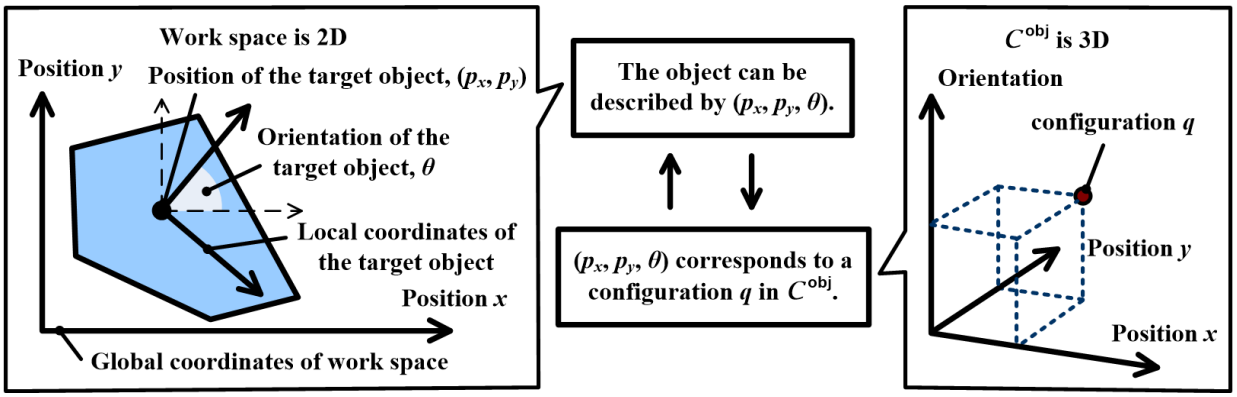


Figure 2.5: Correspondence between an object in work space and a configuration in $\mathcal{C}^{\text{obj}}$.

A point finger in work space corresponds to an **obstacle** in $\mathcal{C}^{\text{obj}}$. Take Fig.2.6 for instance. In the upper-left figure of Fig.2.6, the target object is at a configuration $\boldsymbol{q}$ and it does not collide with the point finger. In this case, the $\boldsymbol{q}$ is free. In the upper-right figure of Fig.2.6, as the target object moves along the red arrow, the target object and the point finger would collide with each other. In that case, the position and orientation of the target object, or namely the configuration $\boldsymbol{q}$ of the target object, becomes obstructed. A target object

could collide with a point finger at many different $\boldsymbol{q}$s. All the $\boldsymbol{q}$s constitute a compact set (sub-space) in $\mathcal{C}^{\text{obj}}$ and we name this set the configuration obstacle. This is shown in the middle row of Fig.2.6. One configuration obstacle is decided by the position of a point finger. Therefore, we say one configuration obstacle corresponds to one point finger. If there are three point fingers like the lower part of Fig.2.6, there will be three correspondent obstacles in $\mathcal{C}^{\text{obj}}$.

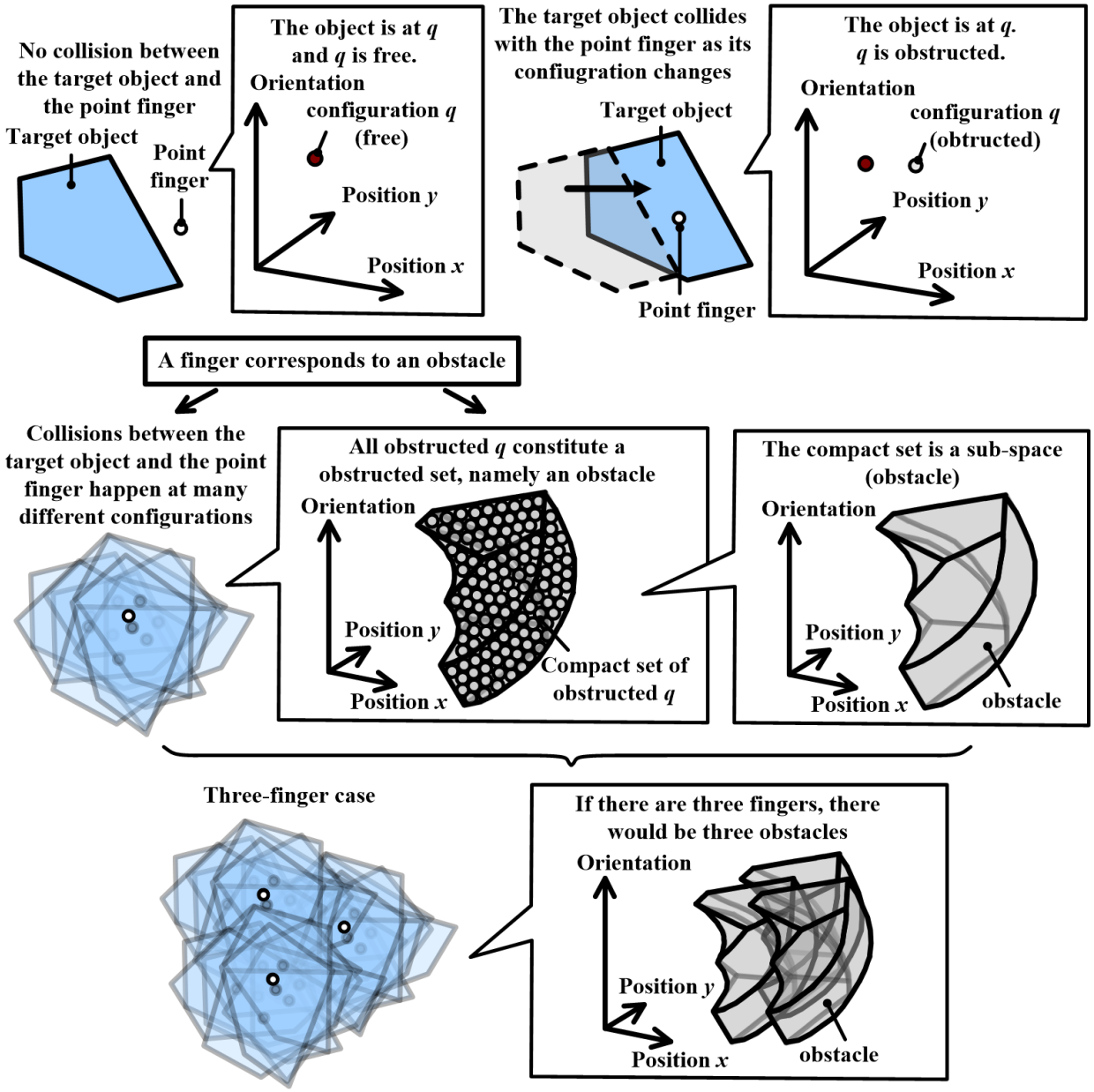


Figure 2.6: Correspondence between a finger in work space and an obstacle in $\mathcal{C}^{\text{obj}}$.

When a target object's configuration is fixed to a single point by the obstacles, we can say the object is immobilized. This is the same as Jurek's definition. When the configuration of a target object is fixed to a single point, any changes in object configuration, or namely any rigid motion of the target object, would cause the configuration to be obstructed by obstacles or namely would force at least one point finger to penetrate into the interior of the target object. An example of immobilization is the fingers and target object in the right part of Fig.2.1. Although those fingers cannot ensure form closure, they immobilize the target object. Fig.2.7 shows the $\mathcal{C}^{\text{obj}}$ of this immobilization example. In this figure, the obstacles are rendered with wire-frames to better illustrate the fixed single configuration. The three wire-framed obstacles compactly enclose an fixed single configuration and therefore the three fingers immobilize the target object. Readers may refer to the two images in the center part of this figure for better comprehension. In the center part, both a whole view which shows all $\mathcal{C}^{\text{obj}}$ and a sliced view which shows only the $\mathcal{C}^{\text{obj}}$ at an orientation θ are rendered. The target object cannot change its position or rotation as any changes in configuration would be obstructed by obstacles. The right part of Fig.2.7 demonstrates two obstructions.

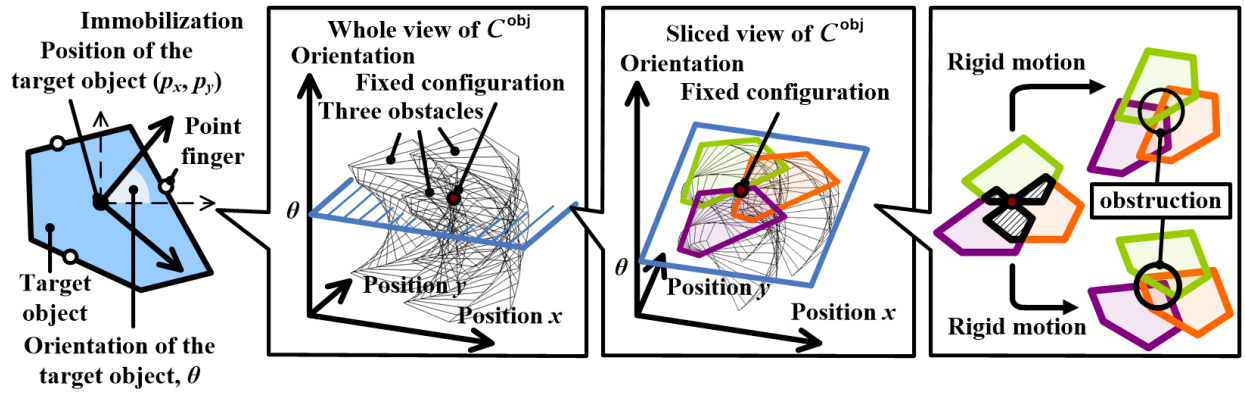


Figure 2.7: Immobilization means a fixed single object configuration.

Immobilization unifies 1st order and 2nd order form closure. 2nd order form closure belongs to immobilization while 1st order form closure is not immobilization. It makes the theory concise. However, we need to pay attention to that immobilization is different from fixture design. In fixture design, researchers expect to fix objects with force closure rather than immobilization. This is because immobilization inherits the problem of infinitesimal motions from form closure, it cannot "firmly" fix objects.

Jurek proved that four points are always sufficient to immobilize any shape. Comparing with force and form closures, immobilization is quite pragmatic. For one thing, it requires few fingers. For another, it involves no wrench analysis. However, that's not the ultimate theory. Caging, the major topic of this thesis, is more general comparing with immobilization. We will see their relationship later in this chapter. Before that, let us review another related topic, namely grasping optimization.

2.1.3 Grasping optimization

The third research related the topic of this thesis is grasping optimization. In accordance with the foregoing introduction, we divide grasping optimization into two aspects. The first one is optimization of force and form closures and the second one is optimization of immobilization.

2.1.3.1 Optimization of force closures and form closures

Basically, following the definition of force closure and form closure, traditional works tend to define a measurement and perform optimization of grasping in the **wrench space**. This **basic measurement** is usually **the radius of Largest Inscribed Sphere (LIS) of the convex hull spanned by wrenches**. Fig.2.8(a) illustrates this basic **basic measurement**.

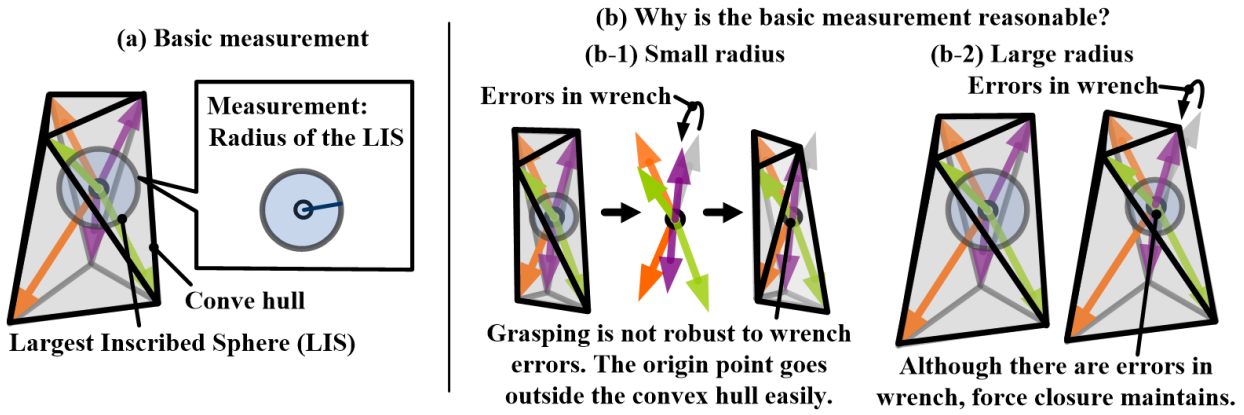


Figure 2.8: Grasping optimization is to find a convex hull whose LIS has largest radius.

The **basic measurement** is reasonable since the larger the radius of LIS is, the more robust a force closure or form closure would be. Fig.2.8(b) illustrates this reason. When the object is unknown, engineering errors, or namely noises from perception devices and control would easily cause errors in normal forces and friction forces. These errors in forces, in the **wrench space**, cause changes in wrench vectors. The middle part of Fig.2.8(b-1) demonstrates this kind of errors and changes. If the radius of LIS is too small, like the case in Fig.2.8(b-1), the origin point of **wrench space** may go outside the convex hull easily and force closure breaks. Therefore, Fig.2.8(b-1) is not robust to errors and changes. Optimization is needed. Comparing with Fig.2.8(b-1), Fig.2.8(b-2) has a larger radius of LIS and it is more robust. Interested readers may refer to the following three references for more details. The first one is reference [Mirtich and Canny, 1994] which gives some intuitive examples that demonstrate the efficacy of **basic measurement** in work space. The second and third one are references [Liu et al., 2004] and [Niparnan et al., 2009] which give formal expressions to calculate **basic measurement** and discuss how to compute force closure and form closures efficiently. Fig.2.8 is based on force closure. Nevertheless, it is not limited to

force closure. It can be applied to form closures in the same way. The **basic measurement** is a fundamental evaluation criteria in **wrench space** optimization.

Of course, there are lots of variations which make the **basic measurement** more pragmatic. For instance, sometimes we need to take into account the given external forces exerted on the target objects. Gravity force is one example of the given external forces. Some other times we need to consider some unavailable contact areas on the surface of target objects. The cutting edge of a knife is one example of those unavailable areas. Grasping the cutting edge does harm to fingers and hands. Fig.2.9 illustrates these pragmatic variations. When there are given external forces, as shown in Fig.2.9(b), the convex hull deforms with respect to the given forces. It results into different radius of LIS and different optimization results. Readers may see [Watanabe and Yoshikawa, 2007] for examples of optimization on given external forces. When there are certain unavailable areas, as shown in Fig.2.9(c), the convex hull deforms with respect to the changes of contacts. It also results into different radius of LIS and different optimization result. Readers may see [Li and Sastry, 1988] for examples of unavailable areas.

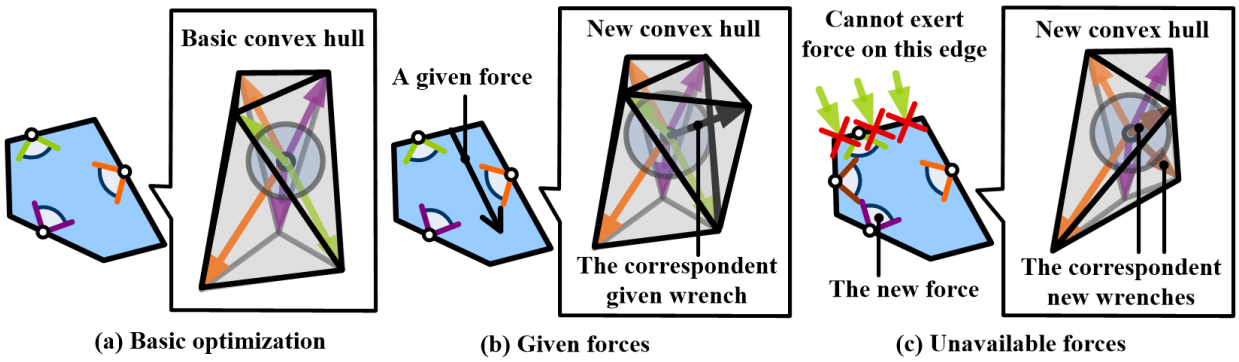


Figure 2.9: Some variations of grasping optimization.

Pure optimization of force closures and form closures is becoming less popular and there are few new publications on this topic in recent two or three years. What's more, it is not directly related to our topic in this thesis. Therefore we won't repeatedly review the historic works of this area. I refer readers to Watanabe's paper [Watanabe and Yoshikawa, 2007] for a good review and classification of literature. Interested readers may also refer to Eris Chinellato's work[Chinellato, 2002][Chinellato et al., 2003][Chinellato, 2008] to better understand different measurements and their performance. They may also refer to fixture design[Wallack and Canny, 1996][Wallack, 1996][Ponce, 1996] to see some practical applications. Readers may find that some figures in [Wallack and Canny, 1996][Wallack, 1996] [Ponce, 1996] are like translational and rotational decomposition in Chapter 3. The similarity, from another view, demonstrates the relationship between grasping closure and caging. We will see the details in Section 2.2.

2.1.3.2 Optimization of immobilization

Like optimization of force closures and form closures, we can also perform Immobilization optimization. Optimization of force closures and form closures are performed in **wrench space** and likewise optimization of immobilization could be performed in $\mathcal{C}^{\text{obj}}$. However, discussions on **immobilization optimization in** $\mathcal{C}^{\text{obj}}$ cannot be found in popular researches. I therefore develop a new measurement for immobilization optimization. I will briefly introduce the new measurement for immobilization in this sub-section. It is an essential connection between grasping and caging.

In previous sections we have seen that immobilization means **the target object, in** $\mathcal{C}^{\text{obj}}$**, is at a fixed single configuration**. In another word, when a target object is immobilized, its correspondent $\mathcal{C}^{\text{obj}}$ can be divided into three components. The first one is the fixed single configuration, which represents the position and orientation of the target object. When the target object is at this configuration, it is immobilized. The second one is the collection of obstacles, which compactly surrounds the single configuration. When the target object is at a configuration of this component, it collides with fingers. The third one is the other "free" space. When the target object is at a configuration of the third component, it can move freely. Considering the three components, we can **evaluate the quality of an immobilization grasp by measuring the minimum distance between the first component, namely the fixed single configuration, and the third component, namely the "free" space.** This measurement is quite intuitive and it is demonstrated in Fig.2.10.

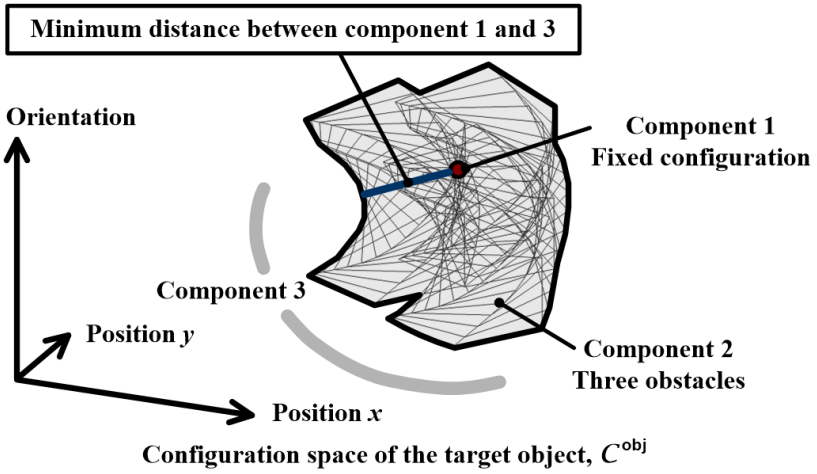


Figure 2.10: An intuitive measurement of immobilization optimization in $\mathcal{C}^{\text{obj}}$.

The same target object and the same fingers as Fig.2.7 is used in this figure. Since the target object is **convex** and the fingers are point fingers, component 2 is composed of three obstacles and it separates component 1 and component 2 compactly. There's no inner-holes in the obstacles. Note that some other objects, such as concave objects or objects with inner

holes, may break the compactness of these components and their measurement won't be as rigid as Fig.2.10.

The measurement in Fig.2.10 can indicate the quality of immobilization. However, it is a rough measurement since it did not take into account that objects are rigid. For rigid objects, it is unnecessary to measure a single obstacle and only the intersections between obstacles plays essential roles. The intersections represents the "distance" between fingers in work space. **The larger an intersection is, the smaller the distance between its correspondent fingers would be.** Fig.2.11 shows the correspondence between the intersections in $\mathcal{C}^{\mathsf{obj}}$ and the inter-finger "distance"s in work space.

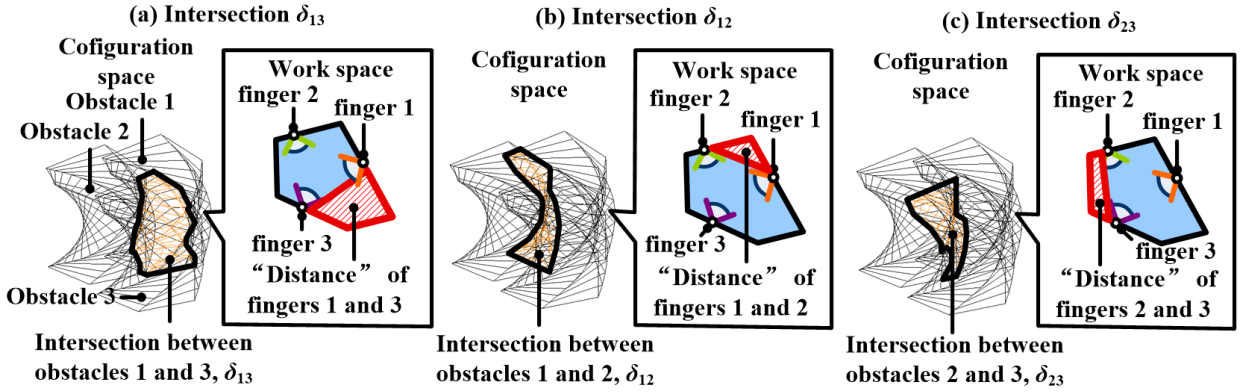


Figure 2.11: Correspondence between intersections of obstacles in $\mathcal{C}^{\mathsf{obj}}$ and inter-finger "distance"s in work space.

Consequently, we can have another measurement, namely **the minimum of intersections**. It is more advisable to use it for rigid objects. Nevertheless, this measurement is quite ambiguous. **The minimum of intersections** requires comparison between the "size"s of intersections. Unfortunately, it is difficult to define a measurement of the "size"s which is difficult to be defined. I rendered the "distance"s between fingers in Fig.2.11 with shadowed areas. Readers may refer to them to retrospect on this difficulty. How can we define a measurement to measure those shadow areas so that it can indicate the quality of immobilization? It remains an open question to immobilization optimization and relates intensively to **caging optimization**.

Caging optimization and immobilization optimization share lots of common backgrounds. They both need the measurement of distances in $\mathcal{C}^{\mathsf{frm}}$. However, I would like to emphasize the differences between them. In grasping optimization, fingers are always in contact with objects. This is not the case in caging. In caging, we would like to deal with engineering errors without considering contacts and forces. The robustness of caging is not to endure force errors, but to endure collisions between fingers and target objects or failures of constraining target objects caused by positioning errors. The difference makes caging optimization more complicated. I will revisit this problem and propose my solutions in relevant chapters later.

Comparing with grasping, caging is force-less and geometric. I am going to put them in the same context and show their relationship in the following texts.

2.2 The Relationship Between Grasping and Caging

Lots of related researches and concepts have been discussed in Section 2.1 and they can be summarized in the following words and figures.

Our discussion begins with force closure. Force closure requires considering both normal forces and friction forces. It is not practical since friction is difficult to model. Therefore, researchers come to the concept of form closure. Theoretical study showed that form closure requires as many as $n_{\text{dof}} + 1$ fingers which conflicts with our common sense. Researchers therefore propose two ways to conquer the contradiction. One is 2nd order form closure which takes surface curvatures of target objects into account. It is a bit complicated. The other is immobilization which divides object motions into finite motions and infinitesimal motions. Immobilization is concise and agrees with our common sense. Besides the basic concepts, researchers performed various attempts to optimize grasping. In force closure and form closure the quality measurement is mainly the radius of LIS. In immobilization, the quality measurement is mainly the minimum distance between the fixed single configuration and configurations in the "free" space.

Caging is the extension of immobilization. If we insert caging into the discussion, it should take an ensuing position after immobilization. Fig.2.12 summarizes our discussion in Section 2.1 with diagrams and shows the position of caging.

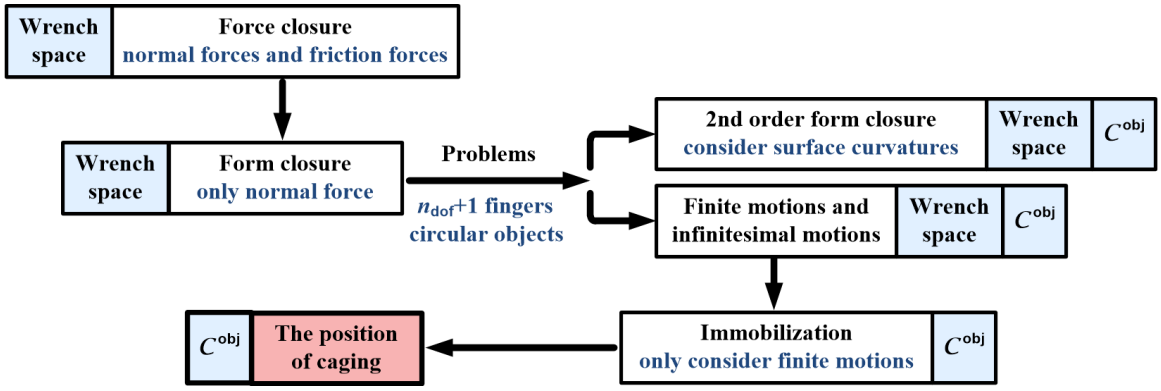


Figure 2.12: The position of caging and its relationship to grasping.

Note that in Fig.2.12, each concept has an accompanying blue frame besides it. The blue frame denotes in which space is the concept developed. Force closure and form closure are developed in **wrench space**. In contrast, immobilization and caging are developed in $\mathcal{C}^{\text{obj}}$. Force closure and form closure are essentially the analysis of forces. In contrast, immobilization and caging are essentially the analysis of geometry. In Fig.2.7 we have seen

the $\mathcal{C}^{\text{obj}}$ of an immobilization grasp. Here I will go on to show the $\mathcal{C}^{\text{obj}}$ of caged objects in Fig.2.13.

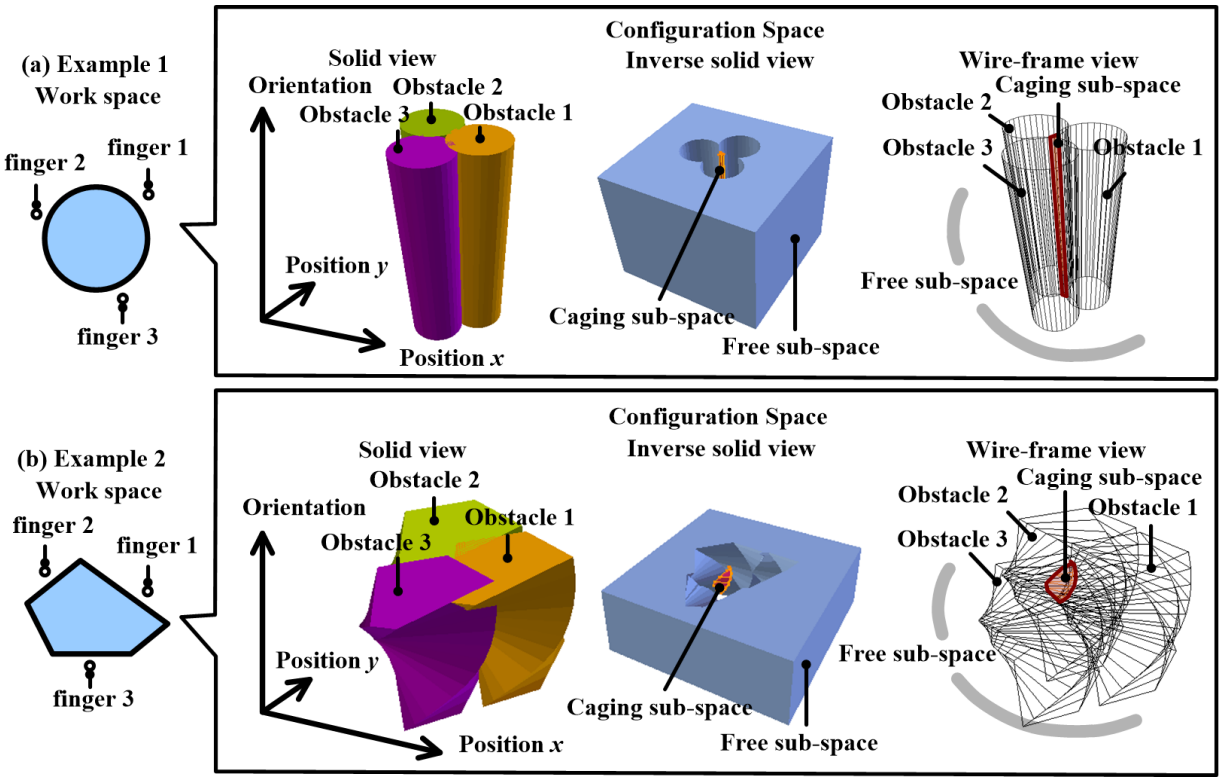


Figure 2.13: The $\mathcal{C}^{\text{obj}}$ of two caged objects.

Two examples of caging are given in Fig.2.13. The first one is a circular object which was used in Fig.1.3. The second one is a polygon which was used in Fig.2.7. According to Kuperberg's definition (see Section 1.3), caging, in work space, means the target object is constrained by fingers. The objects in Fig.2.13 are constrained by fingers in work space and they are caged. In $\mathcal{C}^{\text{obj}}$, caging means the configuration of a target object is constrained in a **caging sub-space**. In side this **caging sub-space**, the target object may move freely. However, it cannot go outside without being penetrated by fingers. The right part of Fig.2.13 illustrates the constraints in $\mathcal{C}^{\text{obj}}$. I made three different drawings to better present it. The first drawing is a solid view of obstacles. This solid view show each obstacle in with solid colors. Like the definition in immobilization (see Fig.2.6), each obstacle corresponds to a finger and there are totally three obstacles in each example. The second drawing is the inverse of the first drawing. In the second drawing, the complementary space of obstacles are rendered. When a target object is caged, the complementary space of obstacles can be divided into two sub-spaces. The first sub-space is a **free sub-space**, it is rendered with light blue in the second drawing of Fig.2.13. The configurations in this **free sub-space** can freely move into infinity without any penetration. The second sub-space is the caging sub-space.

If a target object is at a configuration inside this **caging sub-space**, the object is caged. The caging sub-space is emphasized with orange color in the second drawing. The third drawing is a wire-frame view of obstacles. The wire-frame view gives a more clear view that the caging sub-space is enclosed by obstacles and it is separated from **free sub-space**.

Now we can have the following conclusion that **caging in $\mathcal{C}^{obj}$ means the configuration of the target object is inside a caging sub-space. This sub-space is separated from the free sub-space. It could be either open or closed.** Recall our definition of immobilization. Immobilization means that **the target object, in its configuration space, is at a fixed single configuration.** The difference between caging and immobilization is caging implies a free sub-space while immobilization implies a single configuration. Caging is an extension of immobilization. Fig.2.14 demonstrates the relationship between caging and immobilization with the second example of Fig.2.13.

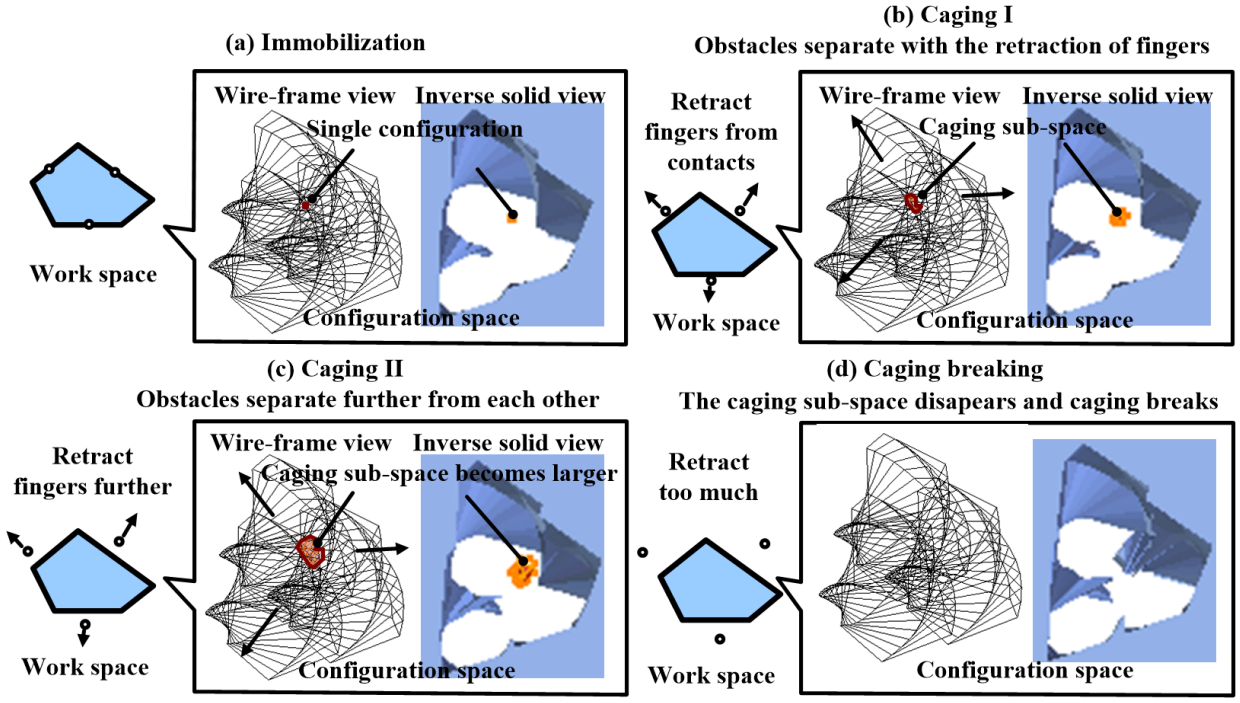


Figure 2.14: The relationship between immobilization, caging and caging breaking.

In figure Fig.2.14(a), the object is immobilized and its $\mathcal{C}^{obj}$ is the same as Fig.2.7. A fixed single configuration exists and it is separated from free spaces by obstacles. If we retract fingers back from the contact points, the single configuration will expand to a compact set, namely a caging sub-space. The immobilization grasping becomes caging. The further we retract fingers, the larger this caging sub-space would be. Fig.2.14(b) and Fig.2.14(c) shows two different cagings. The object in Fig.2.14(c) has higher freedom comparing with Fig.2.14(b) as its caging sub-space is larger. Fig.2.14 shows that if we retract fingers too far from the surface of objects, the caging sub-space may disappear. In that case, the

point fingers will never cage the target object and caging breaks. We can also see Fig.2.14 in inverse order. As we squeeze the fingers, the caging in Fig.2.14(c) and Fig.2.14(b) will finally degenerate into immobilization. The retracting and squeezing directions are important to ensure the continuity between caging and immobilization. We will discuss about that later.

Fig.2.14 demonstrates that caging is the extension of immobilization. However, the object in Fig.2.14 doesn't cover all cases. For example, when the target object is the concave object shown in Fig.2.15 and it is caged by a two-finger formation, no matter how we squeeze the two fingers, they will never immobilize the object. The final shape of the caging sub-space in this case will be an caging surface. That means there must be something else between caging and immobilization.

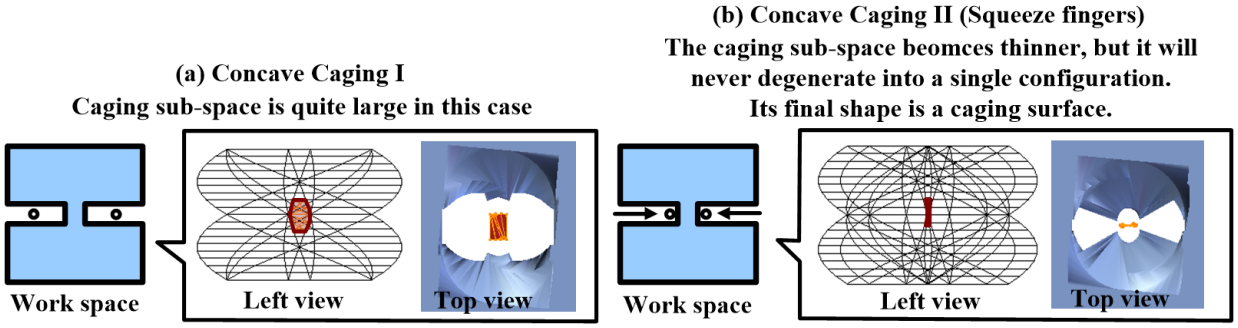


Figure 2.15: Gaps exist between traditional immobilization and caging.

I define this "something else" as **contacting caging**. Readers may compare the following lists for better comprehension.

immobilization	The target object, in its configuration sub-space, is at a fixed single configuration.
contacting caging	The target object, in its configuration sub-space, is on a caging curve and or a caging surface.
caging	The target object, in its configuration sub-space, is in a caging sub-space.

It is easy to get two conclusions from these concepts. (1) We can start from immobilization and move fingers to caging. But we cannot start from caging and move fingers to immobilization. Caging may end up at **contacting caging**. (2) When the target object is convex polygon, there is no **contact caging** and caging directly connects to immobilization. We can either move from immobilization to caging or move from caging to immobilization. That is the most ideal case.

Based on these conclusions, we can further solve a problem. That is, how many fingers are sufficient to cage any objects? In the first conclusion, I claimed that we can start from immobilization and move fingers to caging. That means, **the number of fingers that are**

sufficient to immobilize an object is the number of finger that are sufficient to cage an object. According to Jurek , it requires at least 3 to 4 fingers to immobilize an object in 2D space. When three edges of the object form a triangle, we need at least 3 fingers. Or else, we need at least 4 fingers. More generally, it requires $n_{\mathsf{dim}} + 1$ to $2n_{\mathsf{dim}}$ fingers to immobilize an object in n_{dim} space. (For example, in 3D space, we need at least 4 to 6 fingers to immobilize a 3D object.) Therefore, **the number of fingers that are sufficient to cage an object in n_{dim} space is the same as immobilization. It is $n_{\mathsf{dim}} + 1$ to $2n_{\mathsf{dim}}$.**

Comparing with form closure, the number of fingers that is sufficient to cage an object is much smaller. We need at least $n_{\mathsf{dof}} + 1$ fingers to grasp an object with form closure. A 2D object has 3 DoF and therefore 4 fingers are needed. A 3D object has 6 DoF and therefore 7 fingers are needed. In contrast, we need $n_{\mathsf{dim}} + 1 = 2 + 1 = 3$ to $2n_{\mathsf{dim}} = 2 \times 2 = 4$ fingers to sufficiently cage an object in 2D space and $n_{\mathsf{dim}} + 1 = 3 + 1 = 4$ to $2n_{\mathsf{dim}} = 2 \times 3 = 6$ fingers to sufficiently cage an object in 3D space. Note that this is the **sufficient** number. It is neither the least number nor the maximum number since there are lots of possibilities. When the target object is concave, 2 could be the least number of fingers that are required to cage an object. When the target object has inner holes, 1 could be the least number of fingers. When target objects are convex, 3 or 4 could be the least number of fingers. We will revisit this conclusion in later chapters.

2.3 State-of-the-art Works in Caging

We have connected caging to grasping and seen that caging is an extension of grasping. Connecting caging to traditional grasping research is one contribution of this thesis. To my best knowledge, no other researchers had ever discussed the connection of these concepts in $\mathcal{C}^{\mathsf{obj}}$. Actually, the development of caging is relatively slow due to its abstractness and high cost of computational resources. I will summarize state-of-the-art works in caging in this section. Note that I am not going to repeatedly trace back to those very old publications, but would like to discuss the recent development. Interested readers may refer to [Rodriguez, 2013] for a complete review of old publications.

Major contemporary researchers of caging in the robotics community include Elon Rimon, Andrew Blake, Attawith Sudsang, Vijay Kumar, Zhidong Wang, A. Frank van der Stappen, Yusuke Maeda and Alberto Rodriguez. They all have a series of publications on this topic. Some other researchers, like Jeff Erickson and David J. Cappelleri, also have some publications directly related to caging. Besides them, there are many other researchers who indirectly made great contributions. Let us review their works here.

The first paper that makes caging an independent research topic of robotics is written by Elon Rimon, [Rimon and Blake, 1996]. It is later extended into a journal paper [Rimon and Blake, 1999]. This work is limited to 2D objects and two-finger grippers. Some ensuing improvements of this work involve extension to three-finger grippers and application with visual sensors [Davidson and Blake, 1998a] [Davidson and Blake, 1998b]. The major idea used in these three works is to build a **contact space graph**. This idea is still in use

in the most recent publications of Rimon and his students [Allen et al., 2012]. The name of contact space comes from contacts. Each axis of the contact space corresponds to all the contacts between a finger and the surface of an object. The number of axes of a contact space is the number of fingers that contact with object surfaces. Therefore, the dimension of the contact space is the same as the number of fingers. This space suffers from the curse of dimensionality as finger number increases. That is one reason why the idea of **contact space graph** can only be applied to grippers with limited number of fingers.

The other researcher, Attawith Sudsang, begins his research of caging a little later than Rimon. His first debut is [Sudsang and Ponce, 1998]. In this work, he carries out a simulation on a 2D triangular object to demonstrate his contact-less grasping and manipulation idea. This work is based on a concept named Inescapable Configuration Space (ICS) proposed in [Sudsang et al., 1997] and [Sudsang et al., 2000]. Although this work is not explicitly named "caging", it is essentially the same idea. ICS can be recognized as another description of the **caging sub-space**. However, it is not as intuitive and complete as **caging sub-space**. This is because ICS is defined in work space, not configuration space. Its definition limited its further development. We can find some other similar concepts that are defined in work space and aim to describe **caging sub-space**. The attractive region proposed in [Qiao, 2001] and [Qiao, 2002] is such an concept. These concepts, from another view point, show the interests of robotics community in caging and the importance of choosing a proper mathematical tool. In [Sudsang et al., 1999], Sudsang implements his simulation in [Sudsang and Ponce, 1998] on real mobile robots. These two papers, [Sudsang and Ponce, 1998] and [Sudsang et al., 1999], start the research of performing caging transportation with multiple point robots. Sudsang claims that point fingers and point mobile robots are the same thing. Caging with point fingers shares the same principle as caging with point robots or multi-robot cooperative caging. Later in [Sudsang and Ponce, 2000], Sudsang proposes a basic solution which connects multi-robot cooperative caging and motion planning/obstacle avoidance in the presence of obstacles. Connecting caging to motion planning is quite practical and interesting. Unfortunately, I fail to find their ensuing implementation of this framework. In [Vongmasa and Sudsang, 2006], Sudsang and Vongmasa propose the concept of coverage diameters. This work gives a redundant solution to multi-robot cooperative caging. Although redundant, it is quite practical since caging can be ensured as long as the inter-robot distance is smaller than the coverage diameter. Their solution of coverage diameter is applicable to both convex and concave 2D objects. However, the coverage diameter results into redundant robots. Probably Sudsang realized the drawback of coverage diameter, he published another work [Suarod et al., 2007] which discusses about looser caging. The work [Suarod et al., 2007] is based on Zhidong Wang's proposals. We will review it later when discussing about Wang's works. These works of Sudsang concentrate on caging with point mobile robots. Besides multi-robot cooperative caging, Sudsang also published some works on caging with point fingers. In [Pipattanasomporn et al., 2008] and [Pipattanasomporn and Sudsang, 2011], Sudsang and Pipattanasomporn discusses two-finger squeezing caging. This work is extended to (1) a given formation of fingers and (2) both squeezing and stretching in [Pipattanasomporn et al., 2008].

This work decomposes objects into convex components. Its result is interesting. I am looking forward to their improvement in completeness and limitation of formations. Sudsang and his group are quite active in caging research and make lots of contributions. They also had the idea of using caging to deal with uncertainties [Pipattanasomporn and Sudsang, 2010]. The major difference between their work and mine is their concepts are built in work space. We can find that some of their publications implicitly imply the idea of **contact space graph** and **configuration space**. However, they fail to make explicit expression. Implicitly working in work space obscure their presentation and limits their development.

Zhidong Wang is the first researcher who explicitly expressed caging in $\mathcal{C}^{\mathsf{obj}}$. His work in caging concentrates on multi-robot cooperation. At the very beginning, Wang solves the multi-robot cooperative transportation by task allocation [Wang et al., 1999]. Task allocation requires to specify specific tasks to each robot during cooperative transportation. An impressive task allocation work can be found from [Cheng et al., 2008]. The requirements of task allocation limits its extensibility. I guess that's why Wang change to the idea of caging. Wang's early caging publications were [Wang and Kumar, 2002] and [Wang et al., 2003a]. In these two papers Wang together with Vijay Kumar proposes the concept of object closure and rendered it in $\mathcal{C}^{\mathsf{obj}}$. Object closure is exactly an alternative name of caging. Wang further implements his proposal with three mobile manipulators in [Wang et al., 2003b]. Each mobile manipulator in this work is simplified into a rectangular finger and consequently the implementation is the same as caging with three rectangular fingers. The implementation in [Wang et al., 2003b] controls a precomputed formation of the three mobile manipulators by maintaining certain offset margins from target objects. This formation control strategy is rough and encounters some problems. Therefore, Wang proposes a new control strategy in [Wang et al., 2004]. The control strategy in [Wang et al., 2004] mixes maximum margins and minimum margins of object closure and leader-follower formation control. The new control strategy is validated by using three circular robots and a concave object in [Wang et al., 2005]. These early works of Wang built up a solid basis of caging test. Given a formation of mobile robots and a target object, Wang's algorithm can tell whether the formation of robots could cage the target object[1]. He in [Wang et al., 2006] extends the caging test to a set of robot formations where he could not only test the caging of one formation but many formations in a set. The extended algorithm is named dynamic object closure. Dynamic object closure enables testing many formations of robots at different time intervals and enables robots to cage and transport target objects in real time. It can take the place of formation control. Wang proposes the real-time caging and transportation in [Wang et al., 2009]. The algorithm in this paper plans a path in configuration-time space to connect caging formations calculated by dynamic object closure at different time intervals. I am looking forward to Wang's implementation of the dynamic object closure on real robots. Wang's work is based on his discussion in $\mathcal{C}^{\mathsf{obj}}$. Some of my work borrow and improve his idea. I will refer to them when necessary.

In the same year as Wang and Kumar's publication [Wang and Kumar, 2002], some

[1]His test is not complete. But it is powerful to solve practical problems.

other researchers from Kumar's group published another work based on the idea of object closure [Pereira et al., 2002b][Pereira et al., 2002a]. This work is implemented with three car-like mobile robots and a triangle object with the same control strategy used in [Wang et al., 2003b]. It was later extended to a journal paper [Pereira et al., 2004]. The works of Pereira are not independent, they are more like a complementary branch of Wang. There are some other researches in Kumar's group that work caging. For example, [Fink et al., 2008]. Nevertheless, those works bias towards multi-robot control rather than geometric basis of caging. I am not going to discuss them here. Wang/Kumar's work explicitly discuss caging in $\mathcal{C}^{\mathsf{obj}}$. However, they concentrate too much on the application aspect of multi-robot cooperative transportation and fail to go further into the basic theory. The major difference between Wang/Kumar's work and mine can be concluded into the following three points. (1) They discuss the caging problems only in $\mathcal{C}^{\mathsf{obj}}$ while I explore different tools like $\mathcal{C}^{\mathsf{frm}}$. (2) They do not discuss optimization while I treat caging test and caging optimization as two parallel problems of caging. (3) They assume perfect object information and use redundant number of robots while I assume noisy perception and least number of capture points.

When Wang was trying to figure out a solution to caging test, he proposed the concept of CC-closure object. CC-closure object is **the Configuration obstacle of a Configuration obstacle**. This name is a little obscure but it do fully exploit the the configuration of configuration. It was interesting to find that Jeff Erickson independently developed the same concept in [Erickson et al., 2003] and [Erickson et al., 2007]. Erickon and Wang do not have any interaction with each other but they do proposed the same idea.

Erickson's idea was further developed by Vahedi and Stappen. Stappen's group have excellent background in theoretical grasping and computational geometry and consequently Vahedi's early work analyzes the caging problems in work space by geometric computation. He successfully described the relationship between immobilization and caging (This description is in work space. It is different from my contribution which is in $\mathcal{C}^{\mathsf{obj}}$). In [Vahedi and van der Stappen, 2006] and [Vahedi and van der Stappen, 2008c], Vahedi discusses the problem of caging with two fingers. His algorithm could both perform two-finger caging tests and report two-finger caging sets rapidly. We can find some relationship between these works and [Rimon and Blake, 1996], [Pipattanasomporn et al., 2008] and [Allen et al., 2012]. This is because the vertex-graspings are commonly recognized as the bounds where caging breaks. Vahedi's major contribution is the application of the caging breaking bounds to three-finger cases. In references [Vahedi and van der Stappen, 2007], [Vahedi and van der Stappen, 2008a], [Vahedi and van der Stappen, 2008b] and
[Vahedi and van der Stappen, 2009], Vahedi concentrates on three-finger caging and concentrates on the problem proposed by Erickson's, namely reporting the caging set of a third finger with two given ones or reporting the caging set of a third finger with the given distance between the other two fingers. Especially in [Vahedi and van der Stappen, 2009], Vahedi not only uses work space but also makes a program to show the critical patches in configuration space. Although Vahedi makes certain improvements in computational efficiency, I maintain that his work is essentially the same as Erickson. Vahedi's concept, for instance,

"vertex-grasping", "equilibrium grasping" and "critical patches" actually describe same thing as Erickson's "critical orientation". Nevertheless, Vahedi is the first researcher who purely works on caging theory and describes the relationship between caging and immobilization. He shows the difficulty of caging tests and finding caging sets and partially solves those problems. In his Ph.D thesis [Vahedi, 2009], Vahedi draws lots of important conclusions. For example, the Lemma 6.2.10 and Theorem 6.2.11 on page 68 and 69 show how to maintain caging when shrinking fingers. Many of my ideas in this thesis, like translational caging and accumulation, are borrowed from Erickson and Vahedi's work.

The early works of Maeda and Makita belong to the research field of non-prehensile manipulation [Maeda et al., 2004][Maeda and Makita, 2006]. Try comparing their works in grasping to those works in task allocation of multi-robot cooperative transportation, we can find that the non-prehensile manipulation works of Maeda and Makita are actually the same as multi-robot cooperation by using task allocation. The difference is they use multi-fingers instead of robots. It is very interesting that so many researchers who work with task allocation move to caging. The first caging work of Maeda and Makita is [Makita and Maeda, 2008]. In this paper, they measure the distances between fingers and evaluate with a target may go through those distances. Their solutions are intuitive and require redundant number of fingers. Later, a student from Maeda's group, Yokoi, published a multi-robot cooperation work by using not only mobile robots but also obstacles in the environment [Yokoi et al., 2009]. They take into account walls and transport objects along the walls. Maeda summarizes the work in both [Makita and Maeda, 2008] and [Yokoi et al., 2009] and implements with real robots in [Maeda et al., 2012]. He further discusses and installs some soft parts to rigid mobile robots and fingers in this paper. More recently, Maeda and Makita employ AR (Augmented Reality) markers to recognize shapes of 3D objects. AR markers are quite popular to retrieve shapes of target objects from pre-built databases. Employing AR markers make their caging work practical. However, the number of modeled shapes in the pre-built databases is limited. It is difficult to cover many objects. I in this thesis prefer real-time perception and modeling but I agree that with the help of modern database and machine learning, modeled shapes would bring bright future to robots. Maeda and Makita's work are much more practical comparing with previous works. They are, nevertheless, weak in the caging theories and all their implementations are based on [Makita and Maeda, 2008]. Details of Maeda and Makita's caging research are summarized in Makita's Ph.D thesis [Makita, 2010] (It is written in Japanese.).

Rodriguez is a graduate student of Prof. Matthew T. Mason. He proposes the idea of "from caging to grasping" [Rodriguez et al., 2011]. This idea is nearly the same as my "grasping by caging" proposal. It was interesting that we come to the same idea without any communication. This paper of Rodriguez is awarded the best student paper of Robotics, Science and Systems 2011 and is invited to be published in International Journal of Robotics Research [Rodriguez et al., 2012]. Rodriguez's work in caging is as theoretical as Vahedi. It contrasts significantly with Maeda and Makita's practical implementations. Rodriguez started his research in caging by studying the case of two fingers [Rodriguez and Mason, 2008], which is a well discussed topic of Rimon and Sudsang. The difference of Rodriguez's two-finger

research is he not only considers squeezing but also gives intensive discussions on stretching. He demonstrates that complete caging is a difficult problem as there are too many special cases and he maintains that "from caging to grasping" is essentially to find a F-caging function which could ensure continuous caging of target objects. Rodriguez's study in [Rodriguez and Mason, 2008] and [Rodriguez et al., 2011] is done in topology space of fingers. Using this space as the analyzing tool is quite abstract. He diverts the analyzing tool from topology space into $\mathcal{C}^{\mathsf{obj}}$ in [Rodriguez and Mason, 2012b]. I will discuss later in related chapters the difference between the topology space and $\mathcal{C}^{\mathsf{obj}}$. Rodriguez's most recent work is a review of caging research [Rodriguez, 2013]. His research on caging is theoretical and his review biases towards theoretical researches of caging too.

All the researches in discussed until now work on either fingers or mobile robots. The work of David J. Cappelleri brings something new to our vision. He employs caging in micro-manipulation [Cappelleri et al., 2011] to transport and assemble micro-parts. This paper is later extended to a journal version in [Cappelleri et al., 2012]. Although Cappelleri does not explicitly claim the "grasping by caging" concept, he actually makes use of it. In his publications, Cappelleri firstly cages a micro-object and then shrink the cage into force closure to transport or assemble it. Cappelleri's implementation is based a the redundant calculation like [Makita and Maeda, 2008]. From Makita and Cappelleri's work, we can find that there is a big gap between caging theories and pragmatic applications. Comparing with complicated caging theories, pragmatic applications prefer using simple and redundant algorithms. How to practically use the well developed theories and reduce the redundancy caused by simple algorithms is one challenge in front of us. Rodriguez tells us that complete caging theory involves too many unexpected cases and it is difficult to develop a complete algorithm that takes into account every aspect. Even if the target objects are 2D, the difficulty remains. Therefore, we should seek the balance between complete caging algorithms and pragmatic applications. I am going to discuss my solutions on caging in the next few chapters. In a certain degree, my solutions are complete and ready to be employed by practitioners.

In order to better illustrate the related works, I compile them chronically according to the relationship of the researchers and their ideas. Fig.2.16 shows the chronicle categorization. Each categorized box includes four items, namely the time period, the main researchers, the methodologies and the problems solved. Beside the related works, I attach my work in this thesis into this chronically categorization. Readers may compare my work and the other researchers to better understand the its position.

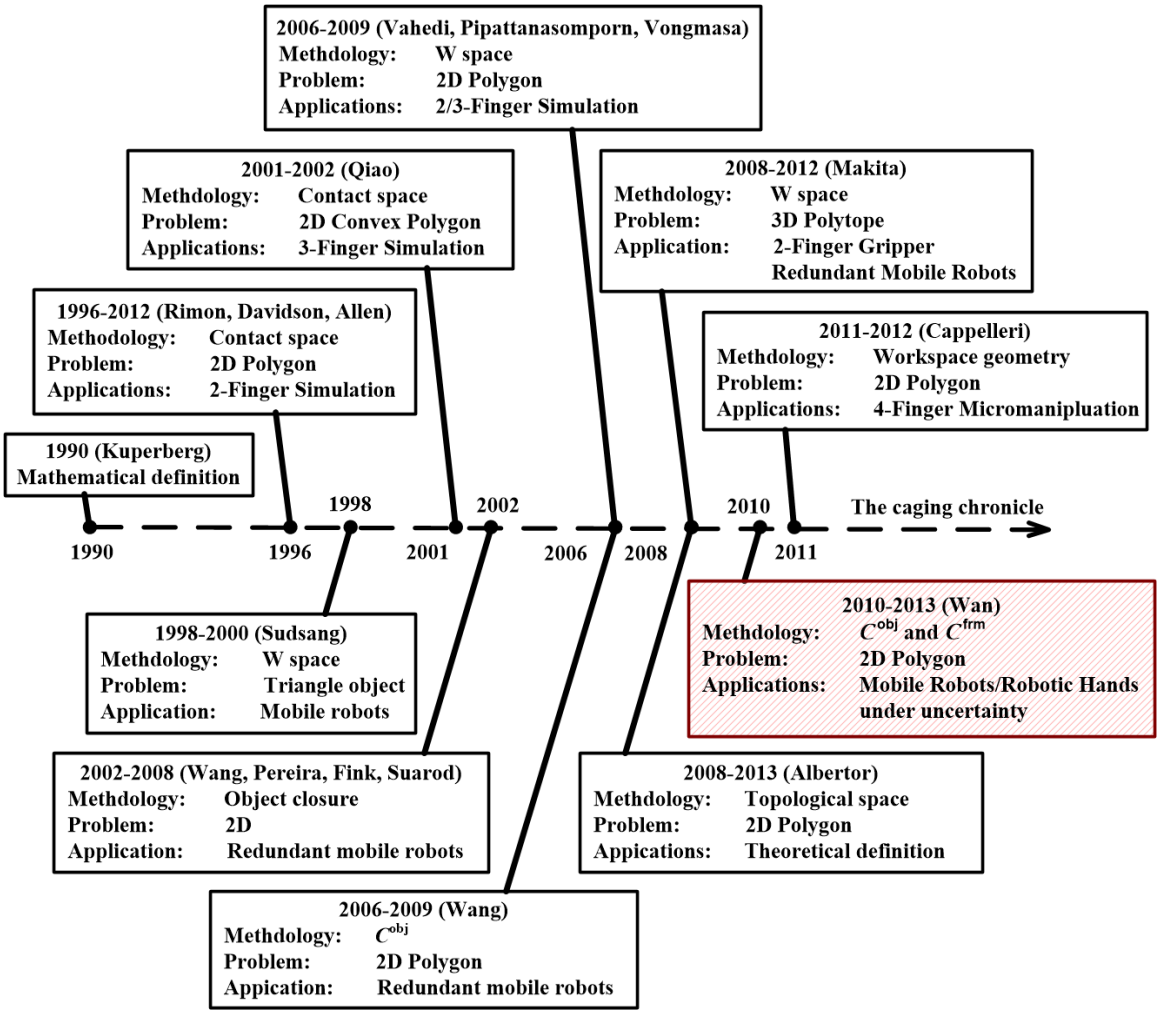


Figure 2.16: A summary of the related works and my contributions.

Part II

Caging in $\mathcal{C}^{\mathsf{obj}}$ and Its Applications

Chapter 3

Caging in The Configuration Space of Target Object

3.1 Caging Test in $\mathcal{C}^{\mathsf{obj}}$

Starting from this chapter, I am going to discuss in detail the tools, techniques, experiments and applications of caging.

Firstly, I list the symbols that will be used in the following part. Readers are recommended to revisit these symbols frequently during their reading process.

$\mathcal{W}$	Work space.
$\boldsymbol{p_i}$	A point in $\mathcal{W}$ space. Since we deal with 2D objects in 2D $\mathcal{W}$ space, the $\boldsymbol{p_i}$ has two coordinate elements $\{p_{i_\mathsf{x}}, p_{i_\mathsf{y}}\}$.
$\mathcal{C}^{\mathsf{obj}}$	The configuration space of target object.
O	The target object in $\mathcal{W}$ space.
$\partial\mathsf{O}$	Boundary of the target object in $\mathcal{W}$ space.
$\boldsymbol{f_i}$	A point finger in $\mathcal{W}$ space. Since we deal with 2D objects in 2D $\mathcal{W}$ space, the $\boldsymbol{f_i}$ has two coordinate elements $\{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}\}$. Mathematically, $\boldsymbol{f_i} = \boldsymbol{p_i}$. I employ this extra symbol $\boldsymbol{f_i}$ rather than using an unified one with $\boldsymbol{p_i}$ to make clear the texts.
$\boldsymbol{q}^{\mathsf{obj}}/\boldsymbol{q_i}^{\mathsf{obj}}$	A configuration of $\mathcal{C}^{\mathsf{obj}}$. Since the configuration space of a 2D object is 3D, a configuration $\boldsymbol{q_i}^{\mathsf{obj}}$ has three coordinate elements $\{q_{i_\mathsf{x}}^{\mathsf{obj}}, q_{i_\mathsf{y}}^{\mathsf{obj}}, q_{i_\theta}^{\mathsf{obj}}\}$. The first two elements $q_{i_\mathsf{x}}^{\mathsf{obj}}$ and $q_{i_\mathsf{y}}^{\mathsf{obj}}$ are actually the same as a position in 2D $\mathcal{W}$ space. That is to say, $\{q_{i_\mathsf{x}}^{\mathsf{obj}}, q_{i_\mathsf{y}}^{\mathsf{obj}}\}$ and $\{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}\}$ both indicate coordinates in 2D plane. The are only different in symbols.
$\mathsf{O}[\boldsymbol{q_i}^{\mathsf{obj}}]$	A target object O at configuration $\boldsymbol{q_i}^{\mathsf{obj}}$. Reader may assume that $\mathsf{O}[\boldsymbol{q_i}^{\mathsf{obj}}]$ represents a region in 2D workspace that is occupied by a target object. It

mathematically equals a set of 2D points. This expression can also be written in the following two forms, (1) $\mathsf{O}[\{q_{i_\mathsf{x}}^{\mathsf{obj}}, q_{i_\mathsf{y}}^{\mathsf{obj}}\}]$ and (2) $\mathsf{O}[q_{i_\theta}^{\mathsf{obj}}]$. The first form, $\mathsf{O}[\{q_{i_\mathsf{x}}^{\mathsf{obj}}, q_{i_\mathsf{y}}^{\mathsf{obj}}\}]$ indicates a region in 2D workspace that is occupied by a target object which could rotate arbitrarily through 0 to 2π. However, its position is fixed at the point $\mathsf{O}[\{q_{i_\mathsf{x}}^{\mathsf{obj}}, q_{i_\mathsf{y}}^{\mathsf{obj}}\}]$. The second form, $\mathsf{O}[q_{i_\theta}^{\mathsf{obj}}]$, indicates a region in 2D workspace that is occupied by a target object which could translate arbitrarily on the 2D plane. However, its orientation is fixed to $q_{i_\theta}^{\mathsf{obj}}$. Note that the superfix are sometimes omitted for conciseness.

$\partial\mathsf{O}[\boldsymbol{q_i}^{\mathsf{obj}}]$ The boundary of the 2D region occupied by a target object at configuration $\boldsymbol{q_i}^{\mathsf{obj}}$. Note that the symbols $\partial\mathsf{O}[\{q_{i_\mathsf{x}}^{\mathsf{obj}}, q_{i_\mathsf{y}}^{\mathsf{obj}}\}]$ and $\partial\mathsf{O}[q_{i_\theta}^{\mathsf{obj}}]$ do not make much sense as the first one is usually the boundary of a circle while the second one is not applicable ($\mathsf{O}[q_{i_\theta}^{\mathsf{obj}}]$ spans all 2D plane).

$\mathcal{F}_i$ One configuration obstacle in $\mathcal{C}^{\mathsf{obj}}$. Please go back to Fig.2.11 for details. Mathematically, it is a set of compact 3D points and a sub-space of $\mathcal{C}^{\mathsf{obj}}$. The subscript indicates the correspondence between $\mathcal{W}$ space fingers and $\mathcal{C}^{\mathsf{obj}}$ space obstacles. For example, $\mathcal{F}_i$ is the $\mathcal{C}^{\mathsf{obj}}$ obstacle of finger $\boldsymbol{f_i}$. Note that the obstacles of different fingers indeed have the same shape in $\mathcal{C}^{\mathsf{obj}}$, they only differ in positions.

$\mathcal{F}_i[q_{i_\theta}^{\mathsf{obj}}]$ Generally, a configuration obstacle $\mathcal{F}_i$ spans the whole *orientation* axis of $\mathcal{C}^{\mathsf{obj}}$. I use $\mathcal{F}_i[q_{i_\theta}^{\mathsf{obj}}]$ to denote a sliced layer of the whole $\mathcal{F}_i$ at *orientation* $q_{i_\theta}^{\mathsf{obj}}$. This combination is not applicable to $q_{i_\mathsf{x}}^{\mathsf{obj}}$ and $q_{i_\mathsf{y}}^{\mathsf{obj}}$.

$\mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}}$ All obstruction-free obstacles in $\mathcal{C}^{\mathsf{obj}}$. It is used to indicate both the caging sub-space and the free sub-space in Fig.2.13. Mathematically, $\mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}}$ is the union of all $\mathcal{F}_i$. Assume there are n fingers, then $\mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}} = \bigcup_{i=1}^{n} \mathcal{F}_i$.

$\mathcal{C}_{\mathsf{free}}^{\mathsf{obj}}$ All free sub-spaces in $\mathcal{C}^{\mathsf{obj}}$. Mathematically, it is complementary to $\mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}}$. Assume there are n fingers, then $\mathcal{C}_{\mathsf{free}}^{\mathsf{obj}} = \mathcal{C}^{\mathsf{obj}} \setminus \mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}} = \{\boldsymbol{q}^{\mathsf{obj}} | (\boldsymbol{q}^{\mathsf{obj}} \in \mathcal{C}^{\mathsf{obj}}) \wedge (\boldsymbol{q}^{\mathsf{obj}} \notin \bigcup_{i=1}^{n} \mathcal{F}_i)\}$

$R^{(\theta)}$ The rotation matrix with respect to an angle θ. For example, if we would like to rotate π, $R^{(\pi)} = \begin{bmatrix} cos(\pi) & sin(\pi) & 0 \\ -sin(\pi) & cos(\pi) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

A caging test problem offers the following conditions. (1) The target object and its initial configuration, say, $\mathsf{O}[\boldsymbol{q_0}^{\mathsf{obj}}] = \mathsf{O}[\{q_{0_\mathsf{x}}^{\mathsf{obj}}, q_{0_\mathsf{y}}^{\mathsf{obj}}, q_{0_\theta}^{\mathsf{obj}}\}]$. (2) The positions of fingers, say, $\boldsymbol{f_1} = \{f_{1_\mathsf{y}}, f_{1_\mathsf{y}}\}$, $\boldsymbol{f_2} = \{f_{2_\mathsf{y}}, f_{2_\mathsf{y}}\}$, ..., $\boldsymbol{f_n} = \{f_{n_\mathsf{y}}, f_{n_\mathsf{y}}\}$ when there are n fingers.

When caging is achieved, we have the following two necessary and sufficient conditions. (1) The $\mathcal{C}_{\mathsf{free}}^{\mathsf{obj}}$ is divided into several disconnected components. Most of the components are enclosed by obstacles, let us denote them with $\mathcal{C}_{\mathsf{fc}}^{\mathsf{obj}} = \bigcup_{i=1}^{u} \mathcal{C}_{\mathsf{fc_i}}^{\mathsf{obj}}$. Here $u = 1$ when the target

object is convex. A special component is the complementary of $\mathcal{C}_{\mathsf{fc}}^{\mathsf{obj}}$, let us denote it with $\mathcal{C}_{\mathsf{ff}}^{\mathsf{obj}}$. This condition follows the following expression, $\mathcal{C}_{\mathsf{free}}^{\mathsf{obj}} = \mathcal{C}_{\mathsf{fc}}^{\mathsf{obj}} \cup \mathcal{C}_{\mathsf{ff}}^{\mathsf{obj}} = (\bigcup_{i=1}^{u} \mathcal{C}_{\mathsf{fc_i}}^{\mathsf{obj}}) \cup \mathcal{C}_{\mathsf{ff}}^{\mathsf{obj}}$, $\mathcal{C}_{\mathsf{fc}}^{\mathsf{obj}} \cap \mathcal{C}_{\mathsf{ff}}^{\mathsf{obj}} = \emptyset$ (2) The configuration of the target object is inside one component of $\mathcal{C}_{\mathsf{fc}}^{\mathsf{obj}}$. Let us denote the configuration of the target object, when performing caging test, is $\boldsymbol{q_0}^{\mathsf{obj}}$. Then, caging requires $\boldsymbol{q_0}^{\mathsf{obj}} \in \mathcal{C}_{\mathsf{fc}}^{\mathsf{obj}}$ or more exactly, $\boldsymbol{q_0}^{\mathsf{obj}} \in \mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}$, $1 \leq k \leq u$. Note that the caging sub-space and the free sub-space of Fig.2.13 are examples of $\mathcal{C}_{\mathsf{fc}}^{\mathsf{obj}}$ and $\mathcal{C}_{\mathsf{ff}}^{\mathsf{obj}}$ respectively.

Recall our discussion of caging in Chapter 2.5: "The target object, in its configuration space, is in a caging sub-space." Following this discussion, caging test can be performed by checking the following expression.

$$(\mathcal{C}_{\mathsf{free}}^{\mathsf{obj}} = (\bigcup_{i=1}^{u} \mathcal{C}_{\mathsf{fc_i}}^{\mathsf{obj}}) \cup \mathcal{C}_{\mathsf{ff}}^{\mathsf{obj}}) \wedge (\boldsymbol{q_0}^{\mathsf{obj}} \in \mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}) \wedge (|\mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}| > 1), 1 \leq k \leq u \tag{3.1}$$

Here $|\mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}|$ means the cardinality, namely the number of elements, of $\mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}$. When $|\mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}| > 1$, the target object is either in the state of caging or in the state of contact caging. When $|\mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}| = 1$, the target object is in the state of immobilization.

Now the caging test problem becomes modeling and intersecting several 3D objects. We have discussed in Section 1.3 that there are two ways of modeling a 3D object. One is **wireframe modeling** while the other one is **solid modeling**. Here I choose the **wireframe modeling** technology to model $\mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}}$. This is because in $\mathcal{C}^{\mathsf{obj}}$, we can easily know the vertices of $\mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}}$. These vertices make **wireframe modeling** easier comparing with **solid modelling**. This is because **Solid modeling** models 3D objects with a set of voxels. If a user want to render an object modeled by **solid modeling**, he has to firstly convert the voxels into a wireframe model [Wikipedia, 2013a]. The conversion process makes **solid modeling** complicated.

Fig.3.1 shows the details of **wireframe modeling**. We have seen in Section 2.1.2 that the $\mathcal{C}^{\mathsf{obj}}$ has three axes, namely *position x*, *position y* and *orientation*. The symbol definition of a configuration $\boldsymbol{q_i}^{\mathsf{obj}} = \{q_{i_\mathsf{x}}^{\mathsf{obj}}, q_{i_\mathsf{y}}^{\mathsf{obj}}, q_{i_\theta}^{\mathsf{obj}}\}$ respectively denote coordinate values along *position x*, *position y* and *orientation*. In order to model the whole wireframe of a $\mathcal{F}_i$. We first discretize rotation, namely the *orientation* axis. With a granularity of $2m+1$, we can divide the rotation of a target object into $2m+1$ angles. In correspondence, the $[-\pi, \pi)$ domain of the *orientation* axis is divided into $2m+1$ coordinate values and the $\mathcal{C}^{\mathsf{obj}}$ between this domain is divided into $2m+1$ layers. Note that we do not need to consider the domains since 2π is the period of rotation and the $\mathcal{C}^{\mathsf{obj}}$ between the other domains are the same as the one between $[-\pi, \pi)$. The $2m+1$ layers have coordinate values along the *orientation* axis ranging from $q_{-m_\theta}^{\mathsf{obj}}$ to $q_{m_\theta}^{\mathsf{obj}}$. The whole model of $\mathcal{F}_i$ is accordingly discretized into $2m+1$ slices. Note that the whole model of $\mathcal{F}_i$ is periodical at every 2π rotation. We only discuss the part between $[-\pi, \pi)$. Fig.3.1(a) illustrates the granularity and divided layers.

When performing caging test, the configuration of the target object is known. It is $\boldsymbol{q_0}^{\mathsf{obj}} = \{q_{0_\mathsf{x}}^{\mathsf{obj}}, q_{0_\mathsf{y}}^{\mathsf{obj}}, q_{0_\theta}^{\mathsf{obj}}\}$. Here the position $\{q_{0_\mathsf{x}}^{\mathsf{obj}}, q_{0_\mathsf{y}}^{\mathsf{obj}}\}$ is equal to the position of a pivot

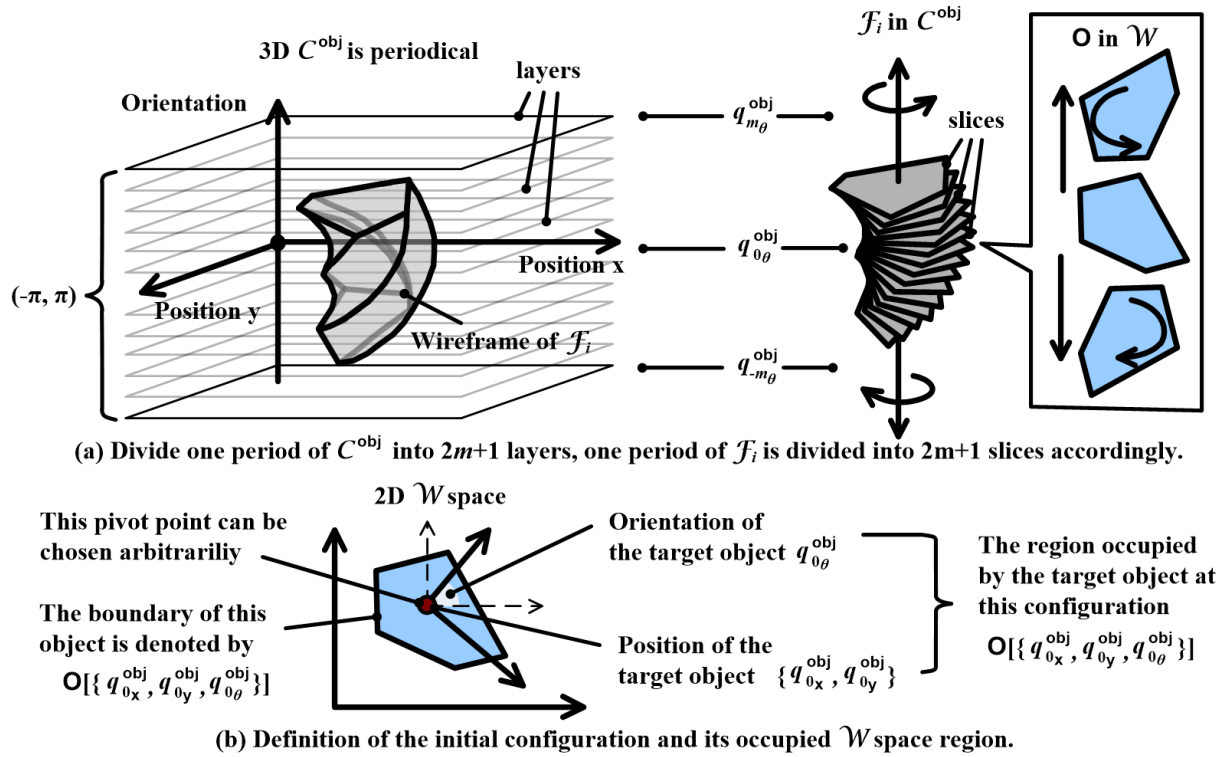


Figure 3.1: Wireframe modeling of a $\mathcal{F}_i$ obstacle and discretization.

point on the target object. This pivot point could be chosen arbitrarily. For instance, [Wang and Kumar, 2002] chooses the geometric center of the target object while [Pereira et al., 2002b] chooses a vertex of the target object. The pivot point is coordinate-invariant and there is no difference between different choices. The $\mathcal{W}$ space region occupied by the target object at its initial configuration therefore can be expressed as $\partial \mathsf{O}[\{q_{0_x}^{obj}, q_{0_y}^{obj}, q_{0_\theta}^{obj}\}]$. Fig.3.1(b) illustrates the initial configuration and the occupied workspace region.

Without any changes in orientation, the $\mathcal{F}_i$ is always a slice at layer $q_{0_\theta}^{obj}$, namely $\mathcal{F}_i[q_{0_\theta}^{obj}]$. If the target object rotates to $q_{j_\theta}^{obj}$, $-m \leq j \leq m$, then the the $\mathcal{F}_i$ becomes a slice at layer $q_{j_\theta}^{obj}$, namely $\mathcal{F}_i[q_{j_\theta}^{obj}]$. Since the target object may rotate to any orientation between $q_{-m_\theta}^{obj}$ and $q_{m_\theta}^{obj}$, $\mathcal{F}_i$ is composed of $2m+1$ layers naming from $\mathcal{F}_i[q_{-m_\theta}^{obj}]$ to $\mathcal{F}_i[q_{m_\theta}^{obj}]$.

Modeling the wireframe of a discretized $\mathcal{F}_i$ essentially equals to modeling $2m+1$ layers of polygon slices. Fig.3.2 shows the detail of how to model the $2m+1$ slices. Given a configuration of the target object, for example $\{q_{j_x}^{obj}, q_{j_y}^{obj}, q_{j_\theta}^{obj}\}$, $-m \leq j \leq m$ and a finger $\boldsymbol{f_1}$, we can generate $\mathcal{F}_1[q_{j_\theta}^{obj}]$ by $R^{(\pi)} \cdot \mathsf{O}[\{f_{1_x}, f_{1_y}, q_{j_\theta}^{obj}\}]$. The following part proves this conclusion.

According to the definition of $\mathcal{F}_0$,

$$\mathcal{F}_1[q_{j_\theta}^{obj}] = \Big\{ \boldsymbol{p_i} | \boldsymbol{f_1} \cap \mathsf{O}[\{p_{i_x}^{obj}, p_{i_y}^{obj}, q_{j_\theta}^{obj}\}] \neq \emptyset \Big\}. \tag{3.2}$$

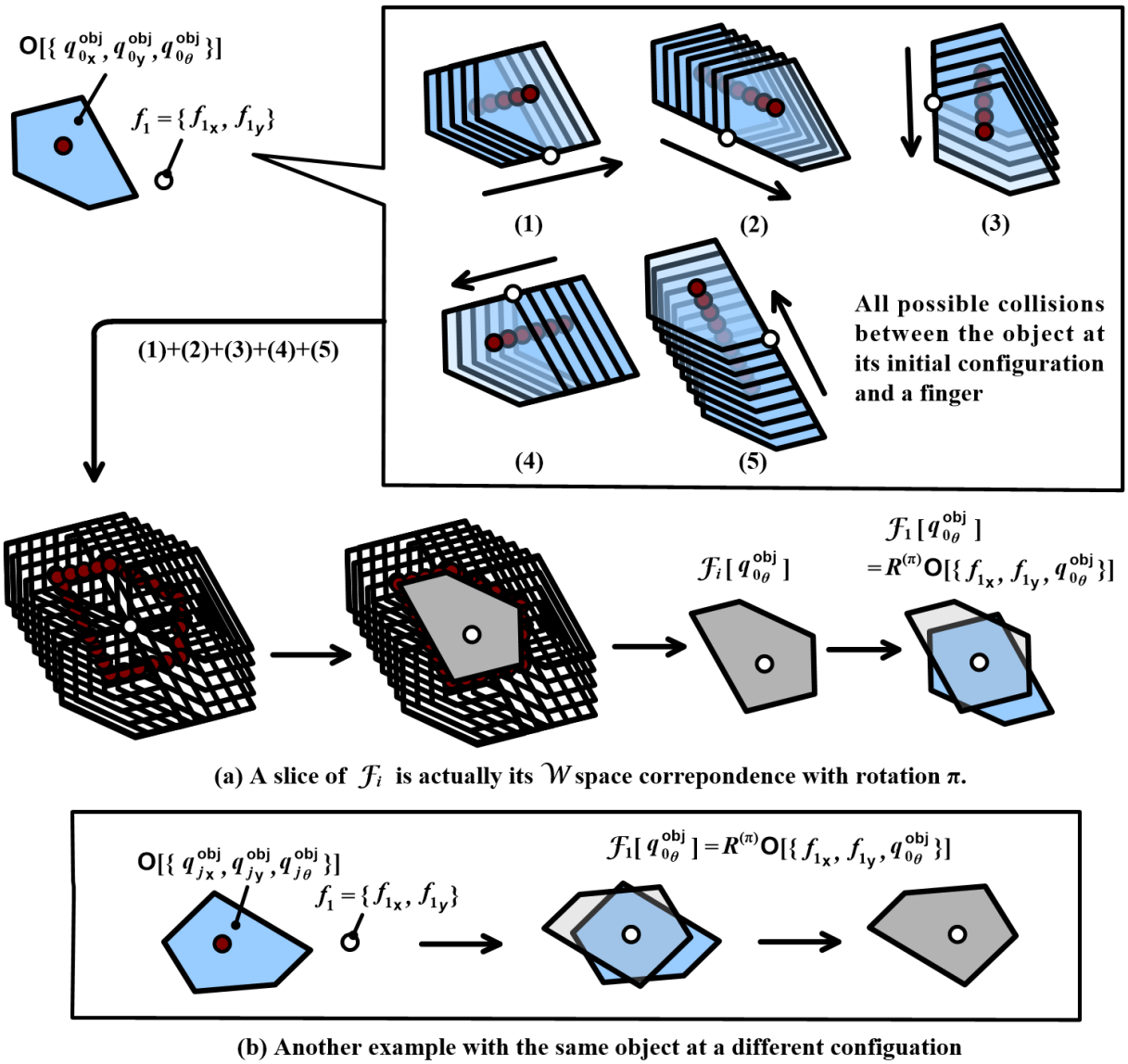


Figure 3.2: Modeling the discretized slices of $\mathcal{F}_i$.

This is equal to

$$\mathcal{F}_1[q_{j_\theta}^{\mathsf{obj}}] = \left\{ \boldsymbol{p_i} | \boldsymbol{p_i} \cap \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathsf{O}[\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, q_{j_\theta}^{\mathsf{obj}}\}] \neq \emptyset \right\} \tag{3.3}$$

since for each point,

$$\boldsymbol{p_i} - \boldsymbol{f_1} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot (\boldsymbol{f_1} - \boldsymbol{p_i}). \tag{3.4}$$

In 2D plane,

$$R^{(\pi)} = \begin{bmatrix} cos(\pi) & sin(\pi) & 0 \\ -sin(\pi) & cos(\pi) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

Therefore,

$$\mathcal{F}_1[q_{j_\theta}^{\mathsf{obj}}] = \left\{ R^{(\pi)} \cdot \partial\mathsf{O}[\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, q_{j_\theta}^{\mathsf{obj}}\}] \right\} \tag{3.6}$$

Fig.3.2(a) graphically illustrates the geometric meaning of these expressions. All the key translations that result into collision between the target object and the finger are shown in Fig.3.2(a)(1-5). By summing up these key translations and connecting the pivot points, we can generate $\mathcal{F}_1[q_{0_\theta}^{\mathsf{obj}}]$. It is easy to find that this $\mathcal{F}_1[q_{0_\theta}^{\mathsf{obj}}]$ equals its $\mathcal{W}$ space correspondence $\mathsf{O}[\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, q_{0_\theta}^{\mathsf{obj}}\}]$ with a π rotation. This is the same as our deduction. In Fig.3.2(b), I give another example with the same object as Fig.3.2(a). The difference is, in this case, the target object is at another configuration, say, $\boldsymbol{q_j}^{\mathsf{obj}}$.

Now we can model the whole $\mathcal{F}_i$ in the following way. Given a finger $\boldsymbol{f_i}$ and a target object at its initial configuration O, we model the $\mathcal{F}_i$ that corresponds to this object by using a set of polygon slices, namely,

$$\left\{ R^{(\pi)} \cdot \partial\mathsf{O}[\{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}, q_{-m_\theta}^{\mathsf{obj}}\}], R^{(\pi)} \cdot \partial\mathsf{O}[\{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}, q_{-m+1_\theta}^{\mathsf{obj}}\}], \ldots, R^{(\pi)} \cdot \partial\mathsf{O}[\{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}, q_{m_\theta}^{\mathsf{obj}}\}] \right\}. \tag{3.7}$$

The caging test in expression (3.1) becomes performing the following procedures. (1) Check whether $\{q_{0_\mathsf{x}}^{\mathsf{obj}}, q_{0_\mathsf{y}}^{\mathsf{obj}}, q_{0_\theta}^{\mathsf{obj}}\}$ is enclosed by $\left\{ \mathcal{F}_1[q_{0_\theta}^{\mathsf{obj}}], \mathcal{F}_2[q_{0_\theta}^{\mathsf{obj}}], \ldots, \mathcal{F}_{n_f}[q_{0_\theta}^{\mathsf{obj}}] \right\}$ when there are n_f fingers. If $\boldsymbol{q_0}$ is enclosed, we can calculate the enclosed region that $\boldsymbol{q_0}$ exists. We denote this region by $\mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}[q_{0_\theta}^{\mathsf{obj}}]$. It is a set of 2D points. (2) For any $\boldsymbol{p_i} \in \mathcal{C}_{\mathsf{fc_k}}^{\mathsf{obj}}[q_{0_\theta}^{\mathsf{obj}}]$, check whether it is enclosed or obstructed by obstacles in neighbour layers. Since the layers adjacent to $q_{0_\theta}^{\mathsf{obj}}$ are $q_{1_\theta}^{\mathsf{obj}}$ and $q_{-1_\theta}^{\mathsf{obj}}$, We should check whether any $\boldsymbol{p_i}$ fulfills that $\{p_{i_\mathsf{x}}, p_{i_\mathsf{y}}, q_{1_\theta}^{\mathsf{obj}}\}$ is enclosed or obstructed by $\left\{ \mathcal{F}_1[q_{1_\theta}^{\mathsf{obj}}], \mathcal{F}_2[q_{1_\theta}^{\mathsf{obj}}], \ldots, \mathcal{F}_{n_f}[q_{1_\theta}^{\mathsf{obj}}] \right\}$ and enclosed or obstructed by $\left\{ \mathcal{F}_1[q_{-1_\theta}^{\mathsf{obj}}], \mathcal{F}_2[q_{-1_\theta}^{\mathsf{obj}}], \ldots, \mathcal{F}_{n_f}[q_{-1_\theta}^{\mathsf{obj}}] \right\}$. If all points in the 2D set are obstructed, then caging is true. If

any point is enclosed, we further calculate the new enclosed region $\mathcal{C}^{\mathsf{obj}}_{\mathsf{fc_k}}[q^{\mathsf{obj}}_{1_\theta}]$ or $\mathcal{C}^{\mathsf{obj}}_{\mathsf{fc_k}}[q^{\mathsf{obj}}_{-1_\theta}]$ that $\{p_{i_\mathsf{x}}, p_{i_\mathsf{y}}, q^{\mathsf{obj}}_{1_\theta}\}$ or $\{p_{i_\mathsf{x}}, p_{i_\mathsf{y}}, q^{\mathsf{obj}}_{-1_\theta}\}$ belongs to and replace the 2D point set with the points in the new enclosed region. For any point in the new enclosed regions, we repeat the procedure done in $q^{\mathsf{obj}}_{1_\theta}$ or $q^{\mathsf{obj}}_{-1_\theta}$ with $\left\{\mathcal{F}_1[q^{\mathsf{obj}}_{2_\theta}],\ \mathcal{F}_2[q^{\mathsf{obj}}_{2_\theta}],\ ...,\ \mathcal{F}_{n_f}[q^{\mathsf{obj}}_{2_\theta}]\right\}$, $\left\{\mathcal{F}_1[q^{\mathsf{obj}}_{3_\theta}],\ \mathcal{F}_2[q^{\mathsf{obj}}_{3_\theta}],\ ...,\ \mathcal{F}_{n_f}[q^{\mathsf{obj}}_{3_\theta}]\right\}$, ...until we reach $\left\{\mathcal{F}_1[q^{\mathsf{obj}}_{m_\theta}],\ \mathcal{F}_2[q^{\mathsf{obj}}_{m_\theta}],\ ...,\ \mathcal{F}_{n_f}[q^{\mathsf{obj}}_{m_\theta}]\right\}$ or with $\left\{\mathcal{F}_1[q^{\mathsf{obj}}_{-2_\theta}],\ \mathcal{F}_2[q^{\mathsf{obj}}_{-2_\theta}],\ ...,\ \mathcal{F}_{n_f}[q^{\mathsf{obj}}_{-2_\theta}]\right\}$, $\left\{\mathcal{F}_1[q^{\mathsf{obj}}_{-3_\theta}],\ \mathcal{F}_2[q^{\mathsf{obj}}_{-3_\theta}],\ ...,\ \mathcal{F}_{n_f}[q^{\mathsf{obj}}_{-3_\theta}]\right\}$, ...until we reach $\left\{\mathcal{F}_1[q^{\mathsf{obj}}_{-m_\theta}],\ \mathcal{F}_2[q^{\mathsf{obj}}_{-m_\theta}],\ ...,\ \mathcal{F}_{n_f}[q^{\mathsf{obj}}_{-m_\theta}]\right\}$. During this repetition, if a point is neither enclosed nor obstructed, caging is considered to be breaking. Or else, caging succeeds.

The narrative of this caging test algorithm seems complicated. Fortunately, it can be implemented with computer programs concisely. Fig.3.3 illustrates the basic ideas of this algorithms with the convex polygon we used in previous figures. The (ccw) part of Fig.3.3(a) shows the continuous caging test along counter-clockwise rotation, namely from $q^{\mathsf{obj}}_{1_\theta}$ to $q^{\mathsf{obj}}_{m_\theta}$. The (cw) part of Fig.3.3(b) shows the continuous caging test along clockwise rotation, namely from $q^{\mathsf{obj}}_{-1_\theta}$ to $q^{\mathsf{obj}}_{-m_\theta}$. At (ccw)-3 and (cw)-2, some points from the continuous refreshing 2D point set are neither enclosed nor obstructed, the caging breaks. The given formation of fingers is considered not able to cage the given target object. Fig.3.3(b) separately illustrates the correspondent failures at (ccw)-3 and (cw)-2 in $\mathcal{W}$ space.

Our figures in Fig.2.13, Fig.2.14 and Fig.2.15 are rendered by using this algorithm. They are programmed with Python and the figures are rendered with the Blender rendering engine [Blender, 2013].

With modern computers, this caging test algorithm can test whether a given formation of fingers could cage a given target object in a few seconds. Given a polygon of n_v boundary points, the computational complexity of this algorithm would be $O(n_v^{n_f} \cdot s \cdot m)$. Here, $O(n_v^{n_f})$ is the complexity of calculating an enclosed region while s is the average size of an enclosed 2D point set. The algorithm is complete with discretization and works with both convex and concave objects.

3.2 Robust Caging in $\mathcal{C}^{\mathsf{obj}}$

We have seen in last part how to discretize $\mathcal{C}^{\mathsf{obj}}$, how to model the wireframes of $\mathcal{F}_i$ at each layer and how to perform **caging test** with the discretization. Beyond **caging test**, we need to (1) find a set of finger formations that could cage the target object and (2) develop a **robust caging** algorithm to find an optimized formation of fingers that could be most robust to endure uncertainty. I refer readers to Fig.1.4 if they need a refresher about uncertainty. Let us firstly consider the item (1), namely how to find a set of finger formations that could cage the target object.

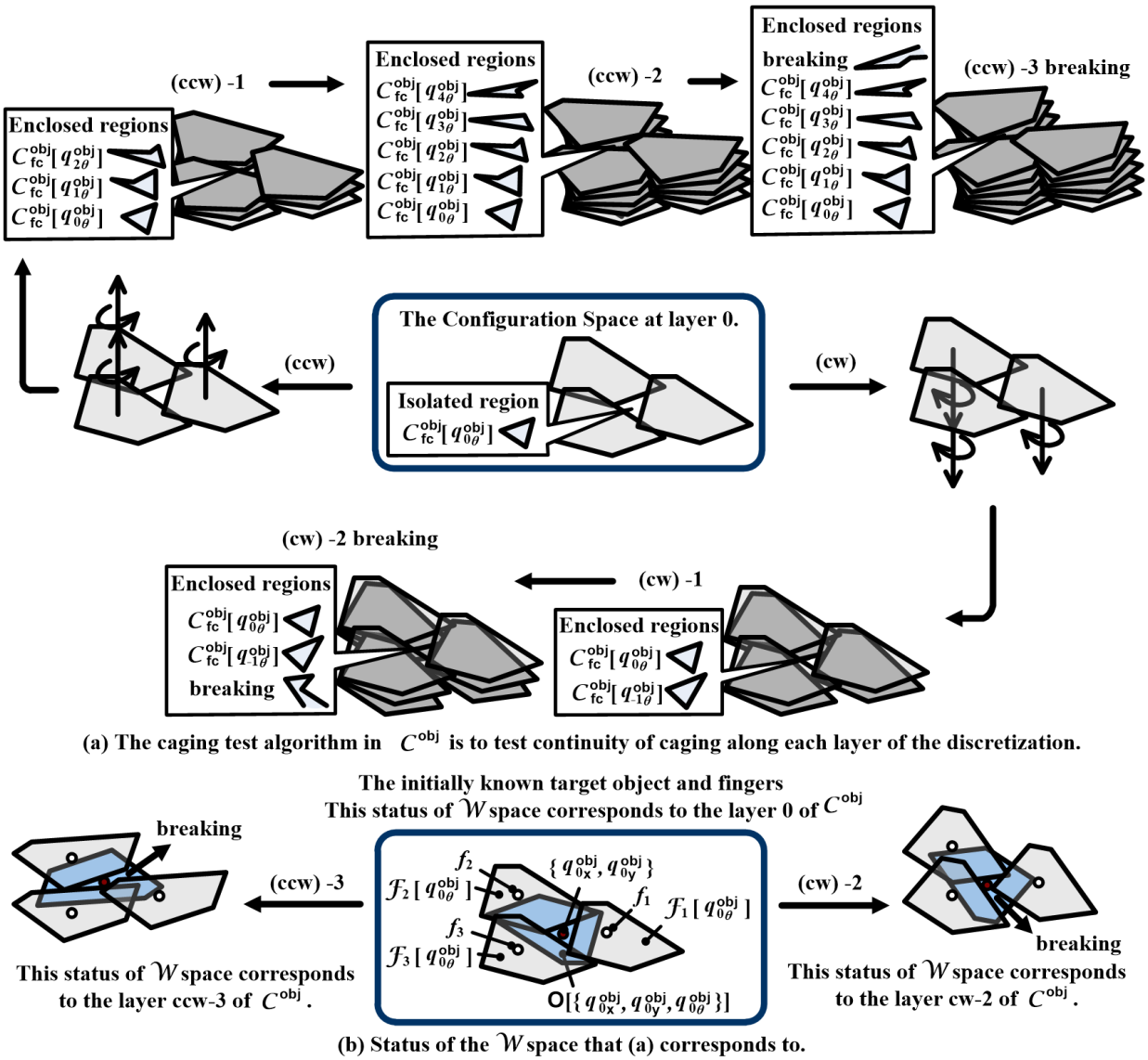


Figure 3.3: The caging test algorithm after discretization becomes testing the continuity of enclosure at each layer.

3.2.1 Finding all possible caging formations is costly

The most intuitive solution to find all possible caging formations is to perform caging test with all finger formations in a certain band that surrounds the target object. Fig.3.4 illustrates this intuitive solution. I proved in Section 2.2 that $n_{\text{dim}} + 1 = 2 + 1 = 3$ to $2n_{\text{dim}} = 2 \times 2 = 4$ fingers are sufficient to cage an object. It was emphasized there that **this is neither the least number nor the maximum number**. It is the sufficient number. That means it would be sufficient if we perform caging test with formations ranging from 2-finger formations to $2 \times 2 = 4$-finger formations. Of course more than 4 fingers would beyond sufficient and far enough for caging 2D objects. We can freely test 5-finger, 6-finger, ..., n_f-finger formations as we like. As examples, I am going to limit my analysis to the sufficient number, namely the 2-finger, 3-finger and 4-finger formations.

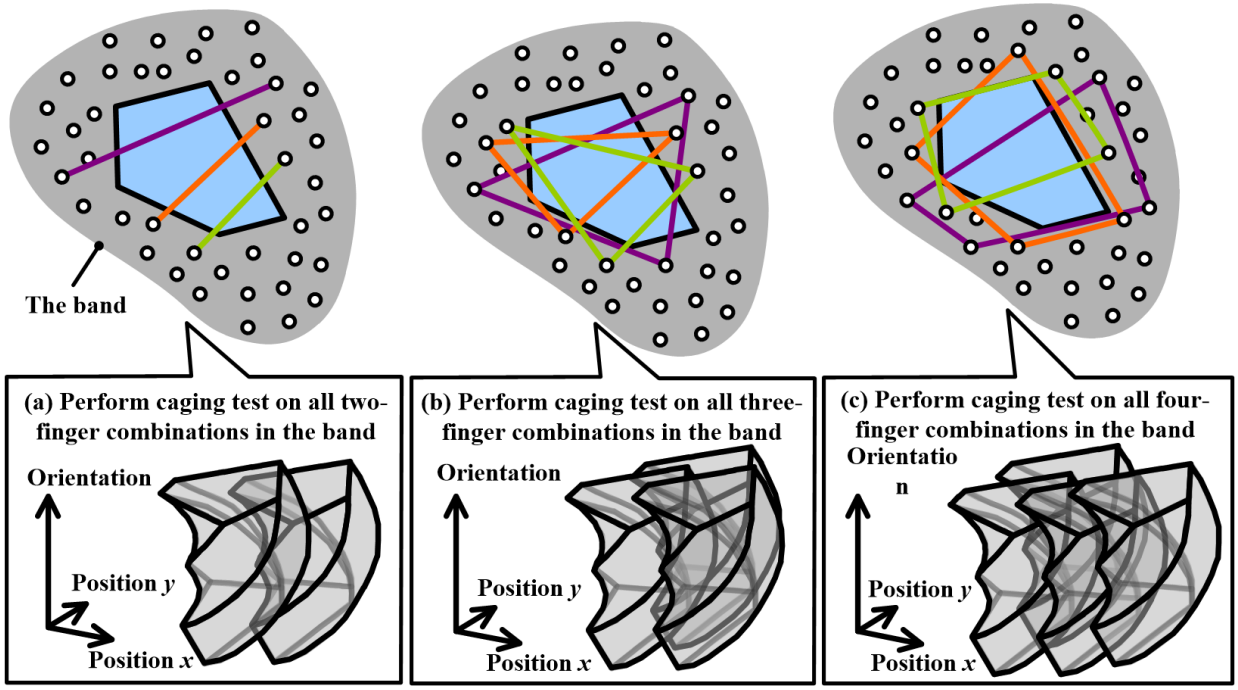


Figure 3.4: An intuitive way to find all caging formations.

Let us see again the intuitive solution in Fig.3.4. In the intuitive solution shown in Fig.3.4, finding all possible 2-finger, 3-finger and 4-finger caging formations means performing caging tests on all 2-finger, 3-finger and 4-finger combinations in the gray band. This is intuitive ans simple. However, it is computationally impossible. Performing one **caging test** with one given formation of fingers would cost $O(n_v^{n_f} \cdot s \cdot m)$. It is to the $(n_f + 2)$th order of variants. Assume that the band includes n_p points, then roughly there would be n_p^2, n_p^3 and n_p^4 finger formations for 2-finger, 3-finger and 4-finger cases. Totally, the cost of finding all caging formations may run as high as $O(n_p^2 \cdot n_v^2 \cdot s \cdot m + n_p^3 \cdot n_v^3 \cdot s \cdot m + n_p^4 \cdot n_v^4 \cdot s \cdot m) = O(n_p^4 \cdot n_v^4 \cdot s \cdot m)$. That's to the 10th order! Or more generally, it would be to the order of $2n_f + 2$ where n_f is

the maximum number of fingers that are employed for caging. Of course it is computational infeasible and we must choose another way to consider this "finding all caging formations" problem.

3.2.2 The caging region of a third finger – Concepts

This part is in publishing.

3.2.3 The caging region of a third finger – Algorithms

This part is in publishing.

3.2.3.1 Tracking the canonical motion

3.2.3.2 Termination conditions

3.2.4 The caging region of a third finger – Demonstrations

This part is in publishing.

3.2.5 The caging region of a third finger – Implementations

This part is in publishing.

3.2.6 Robust three-finger caging and its complexity

This part is in publishing.

3.2.6.1 The measurement of a three-finger immobilization

3.2.6.2 Retracting to a robust three-finger caging

3.3 Faster Robust Caging

This part is in publishing.

3.3.1 Translational constraints and rotational constraints

3.3.2 Collaboration of $Q^{\mathsf{R}}_{\{f_1,f_2,f_3\}}$ and $Q^{\mathsf{T}}_{\{f_1,f_2,f_3\}}$

3.3.3 Some extensions

3.3.3.1 Multiple fingers

3.3.3.2 Grasping by caging

3.3.4 Implementation with Webots

3.3.4.1 Group I – Choosing parameters for the faster robust caging

3.3.4.2 Group II – Evaluating performance of faster robust caging

Chapter 4

Applications I – Distributed Agents

The chapter includes two applications of the faster robust caging algorithm. The first one is our distributed end-effector which has been conceptually illustrated in Fig.1.1 and Fig.1.2. This application uses both KINECT and Swiss Ranger to perceive target objects. We will see the comparison of their performance and the feasibility of my algorithm. The second one is multi-robot co-operation. This application was widely studied by Sudsang, Wang and Pereira, etc. My work here is different from them and it uses the faster robust caging algorithm to calculate caging positions of each robots. The second application uses a V100:R2 OptiTrack [NaturalPoint, 2013] to track and control the motion of each mobile robot. We will see how the algorithm work with it. Note that the "graping by caging" part of this algorithm is not used in the applications. That is because the "grasping by caging" part requires local sensors to perceive the contacts between agents and target objects. Local sensors are not available to the applications of this thesis. However, readers may see the "grasping by caging" procedure in Fig.**??**. In that case, "grasping" is a simple demonstration, it is blind and suffers from the danger of squashing target objects.

4.1 Caging on the Distributed End-effector

4.1.1 Details of the end-effector

4.1.1.1 Hardware implementation

The same as the conceptual illustration in Fig.1.1 and Fig.1.2, the Distributed End-effector is composed of an $x-y-\theta$ actuator and several distributed fingers. The $x-y-\theta$ actuator works as the palm of a robotic hand while the distributed fingers work as the fingers of a robotic hand. Fig.4.1 shows the implemented $x-y-\theta$ actuator. It has two motors to control x and y translation, one motor to control θ rotation and an extra motor to control the insertion of pins. Readers may refer to the connecting module of Fig.**??** and related texts to better understand the θ motor and pin-insertion motor.

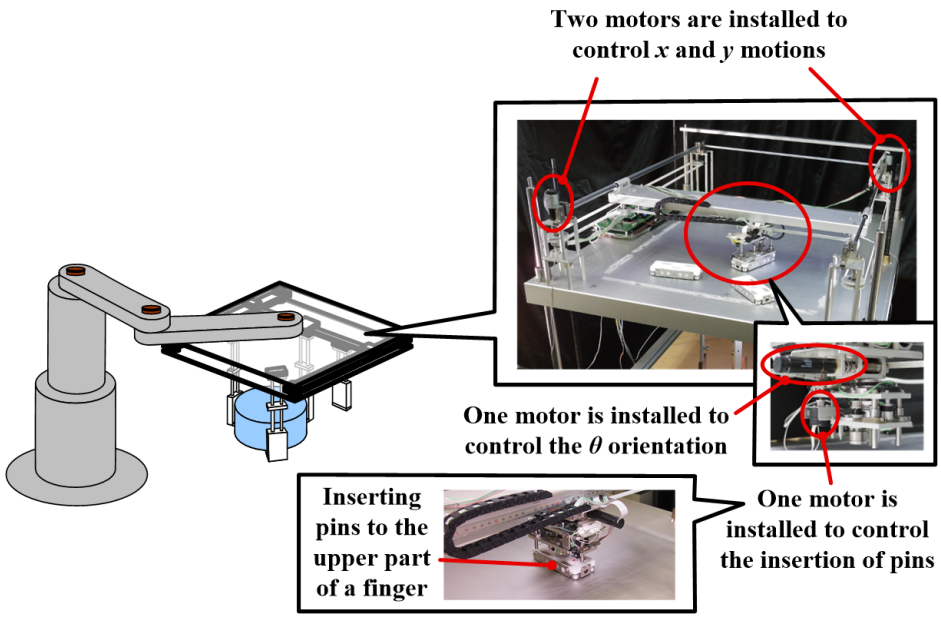


Figure 4.1: The implemented $x - y - \theta$ actuator and the roles of four motors.

The $x-y-\theta$ actuator actuates a finger with three steps. In the first step, it attaches itself to a finger by inserting pins to the fingers. Then, it translates and rotates the attached finger to the position calculated by the faster robust caging algorithm. After that, the $x - y - \theta$ actuator detach itself from the upper part of the finger and finish the actuation. Readers may review Fig.1.2 to recall this procedure. Fig.4.2 shows the details of one distributed finger. It involves a prismatic module which stretches fingers down and a nail module which will be inserted into the bottom of target objects.

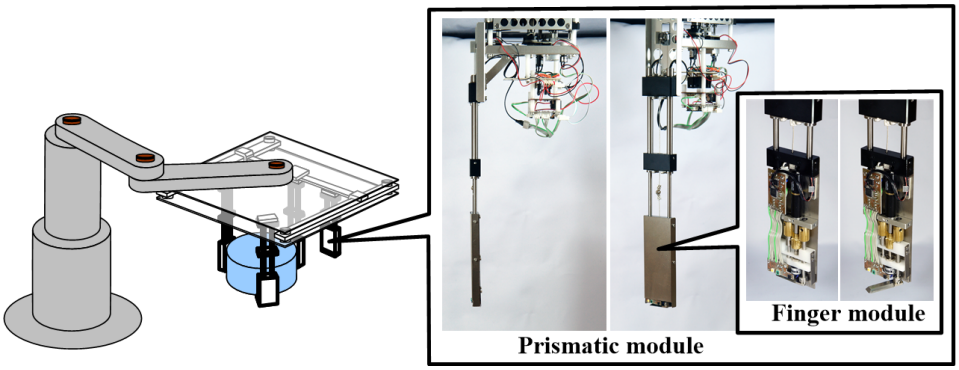


Figure 4.2: The implemented one distributed finger.

Since there are not enough room to install a manipulator shown in the left part of Fig.4.1 and Fig.4.2 to move this end-effector, I use some tricks to test the performance of

my algorithm. Rather than moving the end-effector, I install a sliding plate under the end-effector and move the sliding plate instead. The lower-left dialog box of Fig.4.3 shows the sliding plate. At the corners of this sliding plate, four markers are installed to help calibrate perception devices which are installed on the top. The perception devices involve a KINECT and a Swiss Ranger, they are emphasized in the upper-left dialog box of Fig.4.3.

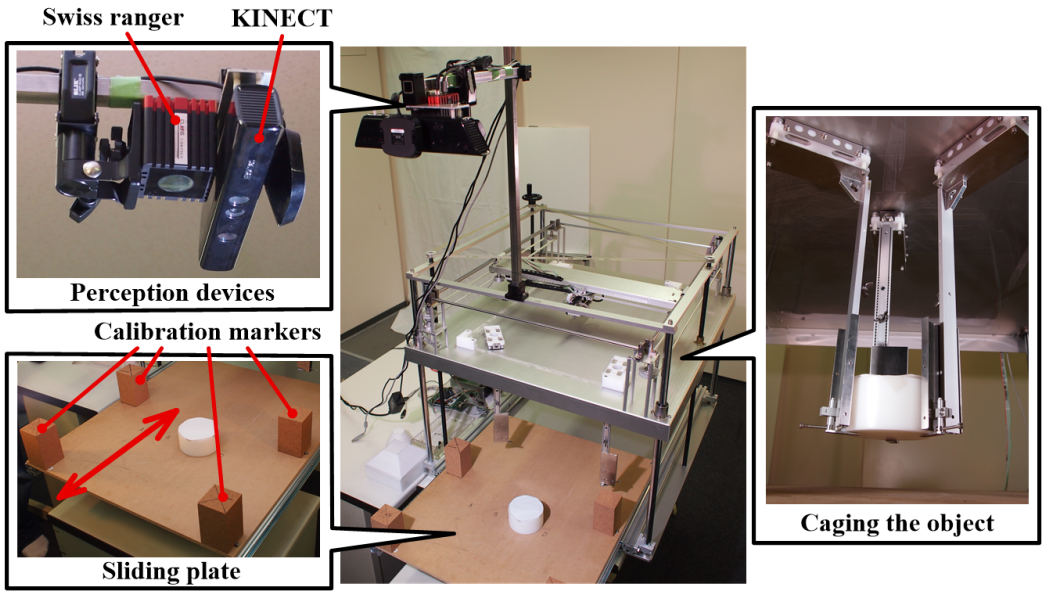


Figure 4.3: The sliding plate and the perception devices.

The right dialog box of Fig.4.3 shows an action where the distributed end-effector cages a cylindrical target object. Note that in this application I use some dummy fingers instead of the complete version shown in Fig.4.2. These dummy fingers have the same mechanical structure as Fig.4.2 except that they are not equipped with motors. These dummy fingers save costs of our implementation as well as being enough to demonstrate the faster caging algorithm.

4.1.1.2 Software integration

In the lower level, the end-effector is controlled by a AVR Atmega2560 micro-controller which can perform basic commands from computers. On the one hand, the micro-controller encapsulates low-level controls like proportional-integral-derivative algorithms and deals with the low-level protocols like formats of encoder data. On the other hand, it receives commands from computers through serial communications and interrupts and controls motors according to those commands.

In the higher level, the end-effector is controlled by MATLAB2011b running on a MacBook. Fig.4.4 shows the MATLAB interface of the high-level controller and its relationship with the low-level micro-controller.

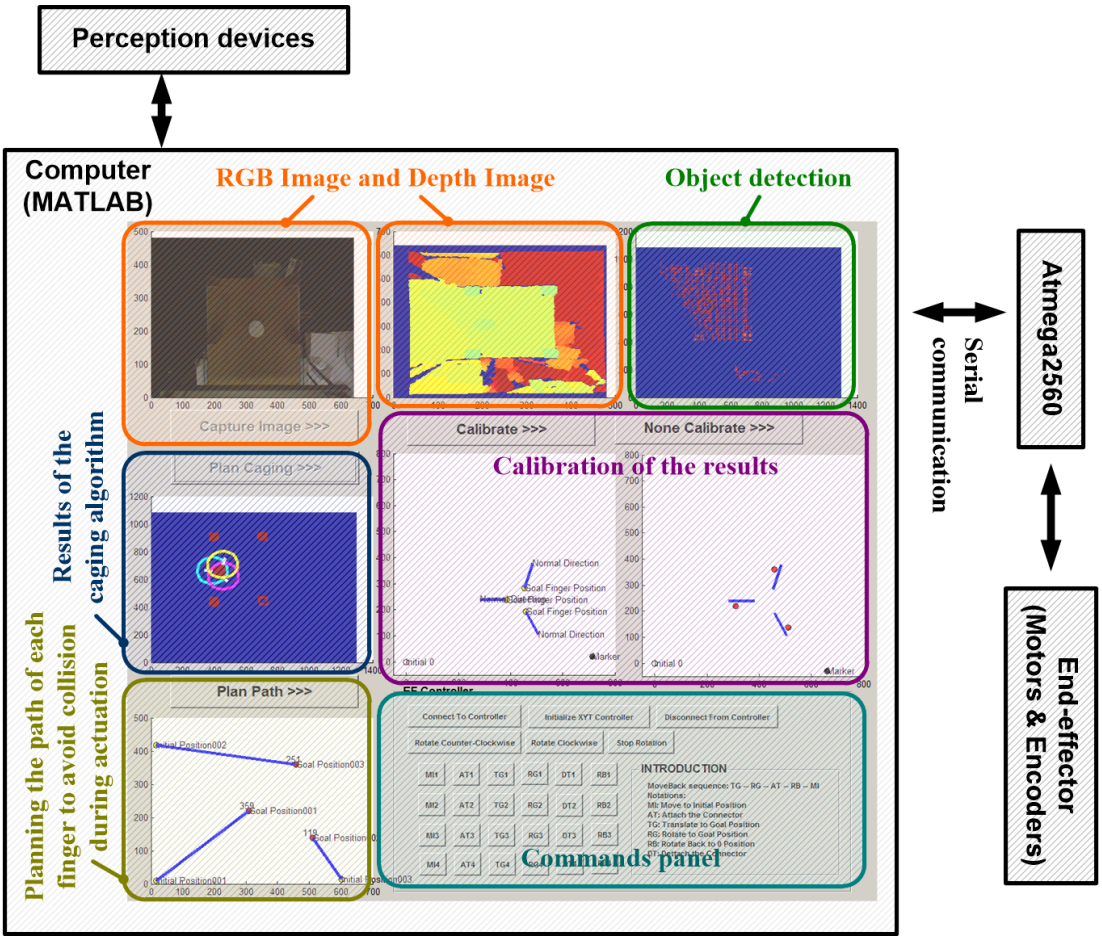


Figure 4.4: Connections between the high-level computer and the low-level micro-controller.

It can be seen from Fig.4.4 that the high-level computer performs the following tasks consequently. (1) Processing the information collected from perception devices. This task is shown by the orange boxes and the green box in Fig.4.4. (2) Calculating an optimized caging formation. This task is shown by the blue box in Fig.4.4. (3) Calibrating the perception devices. This task is shown by the purple box in Fig.4.4. (4) Planning actuation paths of each finger. This task is shown by the yellow box in Fig.4.4. After that, the high-level computer encode the results of those tasks into commands and send the commands the micro-controller through serial communication. This is shown by the cyan box (command panel) in Fig.4.4. The micro-controller actuates the motors on the end-effector according to the received command to perform caging.

4.1.2 Demonstration and analysis

We have reviewed the hardware implementation and software integration in foregoing texts. Next, it is time to see the results of the faster robust caging algorithm. We will see two topics in this sub-section. One is comparison between KINECT and Swiss Ranger and the other one is the demonstration with different convex objects.

4.1.2.1 Comparison between KINECT and Swiss Ranger

I tested two different perception devices, namely the KINECT and the Swiss Ranger. The two perception devices have different precisions so that we can better see the robustness of the faster robust caging algorithm.

KINECT and Swiss Ranger represent two different types of depth cameras. The KINECT calculates the offset of the dot patterns from their factory calibrated ideal positions. Depth of a certain point is evaluated by the offset. The KINECT camera is composed of an IR source, an IR camera and a RGB camera. Its resolution is 640×480. The Swiss Ranger measures changes in the light phase at each image sensor signal. The phase difference divided by period length indicates a portion of maximum measurement, namely the depth value. The Swiss Ranger is composed of a light source and an image sensor. Its resolution is 176×144. Although the Swiss Ranger is lower in resolution, it is higher in precision and of course higher in price.

Precision of the depth camera, especially the KINECT camera, depends on its distance to the scenarios. Therefore, I install the two cameras to four different positions to compare their precisions. The four different positions are shown in the upper-left part of Fig.4.5. I measure the distance between two markers shown in the upper-right part of Fig.4.5 and compare the measured results with ground truth value. Groundtruth value of the distance between markers is $510mm$. However, the measured results differ conspicuously. They are shown in the lower bar-graph of Fig.4.5.

The Swiss Ranger has relative higher precision. Its error is less than $10mm$. The KINECT, however, is much worse and introduces an error of nearly $50mm$. The precision of devices introduces limitation to their caging applications. By comparing the caging regions of objects in Fig.**??**, we can roughly have a reference value that the robustness of caging is about $\frac{1}{10}$ of object diameter [1]. Note that this is a rough value. It cannot be used to evaluate robustness but it can be used to check whether the perception devices are suitable to some caging applications. The Swiss Ranger is potentially fine to work with objects with target objects whose diameters are no less than $100mm$. The KINECT is much worse. It may fail to work as a suitable perception device unless the diameter of target objects are larger than $500mm$.

Fig.4.6 shows the results with the cylinder object shown in Fig.**??**. The diameter of this cylinder object is $100mm$ and the height of this cylinder object is $50mm$. The caging results at those four different heights in the upper-left part of Fig.4.5 are shown in Fig.4.6(a), (b),

[1] Object diameter can be considered as the diameter of largest outer circle that covers the object

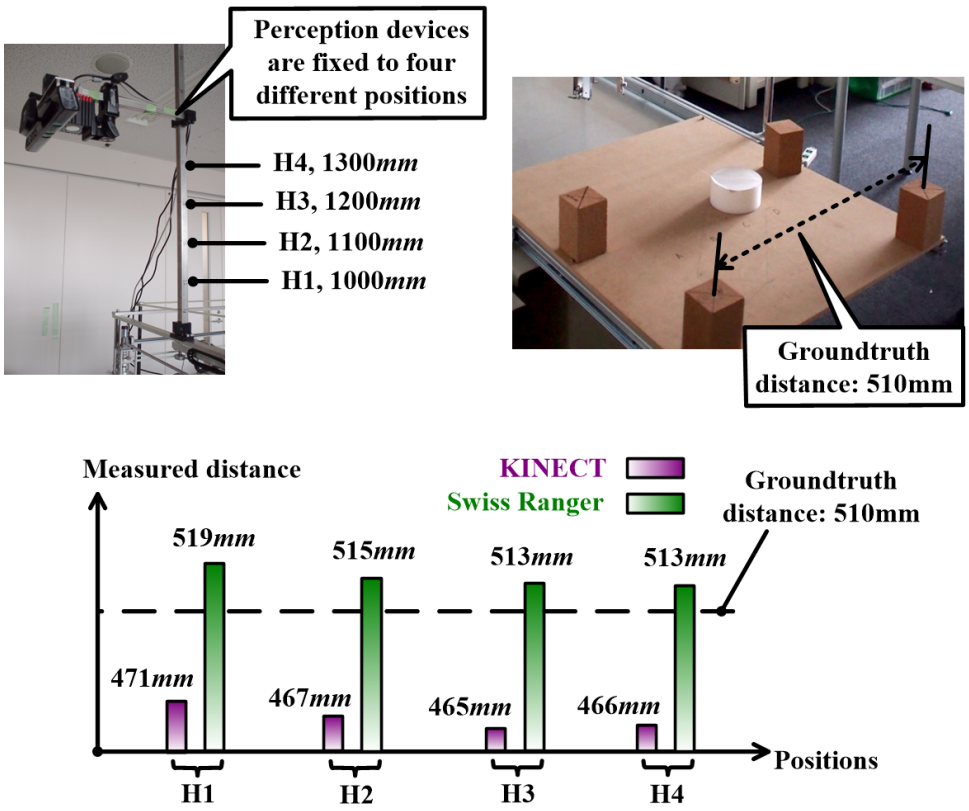


Figure 4.5: Perception errors of KINECT and Swiss Ranger.

(c) and (d) respectively. Both Swiss Ranger and KINECT are involved at each height. The ideal caging results, namely the caging results calculated with perfect shape and position information, are rendered in red while the caging results calculated with cloud points from the perception devices are rendered in green. The Swiss Ranger, especially when its height is higher than H2=1100mm, is robust enough to our faster robust caging algorithm. On the contrary, the KINECT fails due to dramatic offset.

4.1.2.2 Demonstration with various target objects

This part is in publishing.

4.2 Caging on Multi-robot Co-operative Transporation

This part is in publishing.

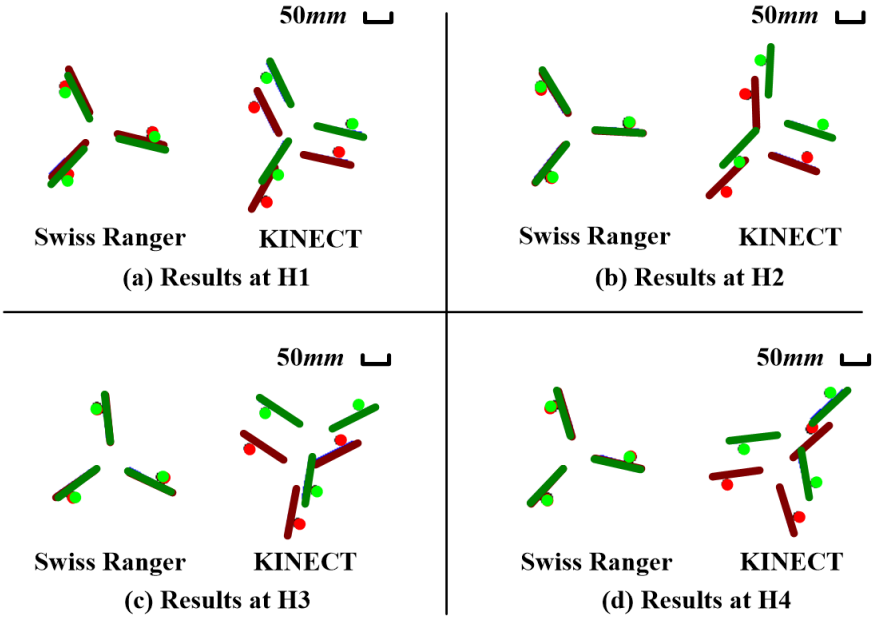


Figure 4.6: Comparing the caging results of different devices.

4.2.1 Simulation

This part is in publishing.

4.2.1.1 Transportation by formations control

4.2.1.2 Choosing proper robot number

4.2.1.3 Software integration

4.2.1.4 Formation control

4.2.1.5 Robustness to perception uncertainty

Part III

Caging in $\mathcal{C}^{\mathsf{frm}}$ and Its Applications

Chapter 5

Caging in The Configuration Space of Fingers

5.1 Changing From $\mathcal{C}^{\mathsf{obj}}$ to $\mathcal{C}^{\mathsf{frm}}$

The simulations and real-world applications in second part of this thesis shows that performance of the faster robust caging algorithm in $\mathcal{C}^{\mathsf{obj}}$ is satisfying. However, it suffers from a fatal drawback. Say, it can only be applicable to convex objects. How can we extend it to many other objects? In order to solve this problem, we need to recall how our faster robust caging algorithm in $\mathcal{C}^{\mathsf{obj}}$ became limited to convex objects. The answer involves many fundamental techniques of the faster robust caging algorithm, for instance, $\mathcal{A}_{\mathsf{c}}$, immobilization optimization and translational caging. That means if we would like to extend the algorithm to various other objects, the whole algorithm should be redesigned. We should no longer employ immobilization optimization and no longer employ translational caging. That is difficult in $\mathcal{C}^{\mathsf{obj}}$ and therefore in the third part of this thesis, I re-consider the algorithm in **the configuration space of finger formation**. This **configuration space of finger formation** will be denoted by $\mathcal{C}^{\mathsf{frm}}$. I will show the details of it and the new algorithm in following contexts.

5.1.1 Consider a different center

I introduced $\mathcal{C}^{\mathsf{obj}}$ in section 2.1.2 when discussing about immobilization. In $\mathcal{C}^{\mathsf{obj}}$, the target object becomes a 3D point while the fingers becomes 3D obstacles. The caging test in $\mathcal{C}^{\mathsf{obj}}$ is actually testing whether a point is enclosed by obstacles. During this procedure, the target object is **the center of planning** and the configuration space encodes the position and orientation of the target object.

We can consider the **center** in a reverse way. Recall that there are two conditions for a caging test problem. I introduced it in the beginning of section 3.1. One condition is the center of $\mathcal{C}^{\mathsf{obj}}$, namely the target object. The other condition is the positions of fingers. Can

we take the position of fingers as the center and build a space that encodes these positions? The answer is positive. Fig.5.1 illustrates this reverse consideration.

When target object is the center, caging test means to see if the target object can escape from the formation of fingers. Fig.5.1(a) illustrates this idea. The correspondent configuration space $\mathcal{C}^{\mathsf{obj}}$ is a three-dimensional $\mathbb{R}^2 \times \mathcal{S}$ space. When the positions of fingers are the center, caging test means to see if the formation of fingers can escape, or go through, the target object. Fig.5.1(b) illustrates this idea. The correspondent configuration space is a $2n_f$-dimensional $\mathbb{R}^2 \times \mathbb{R}^2 \times \ldots \times \mathbb{R}^2 = \mathbb{R}^{2n_f}$ space where n_f is number of fingers. I name this space $\mathcal{C}^{\mathsf{fgr}}$ to emphasize its difference from $\mathcal{C}^{\mathsf{frm}}$.

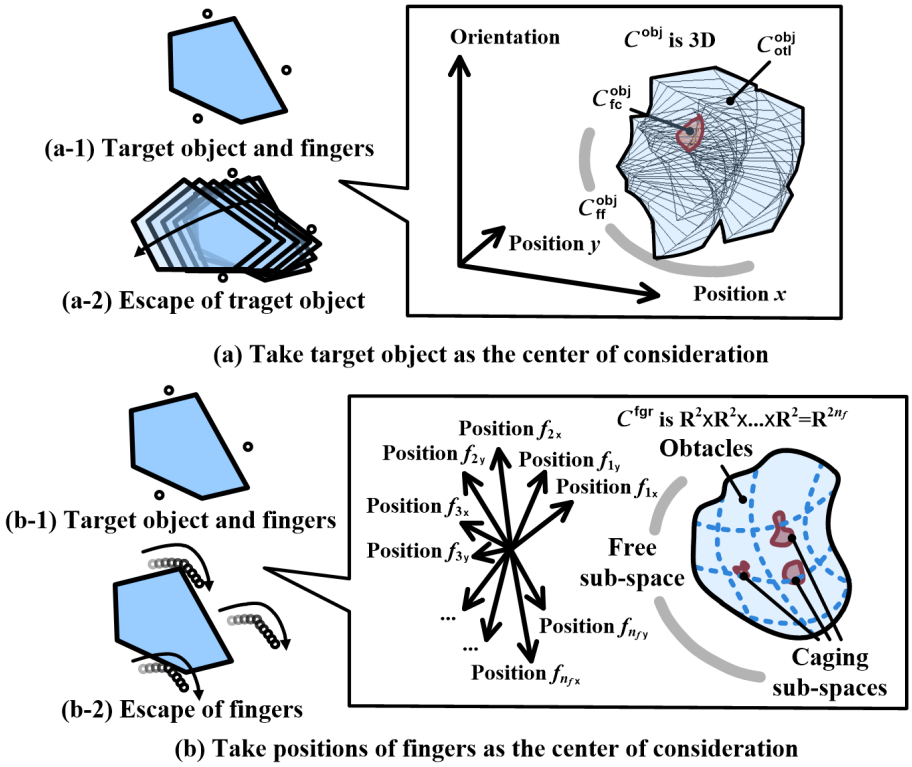


Figure 5.1: Comparison of different centers.

The bad news of taking positions of fingers as the center of consideration is it suffers from curse of dimensionality. As we can see from Fig.5.1, the dimension could be as high as $\mathbb{R}^{2n_f}$ with $2n_f$ fingers. This makes it difficult to model the obstacles in this space. Works like Rodriguez [Rodriguez et al., 2011] and Pippattanasomporn [Pipattanasomporn and Sudsang, 2011] are based on exploration of this space. However, they either take the space as a topological space which cannot be modeled or reduce the number of fingers n_f into 2 for easier analysis. That is not satisfying. There are two strategies to deal with the high dimensionality of $\mathcal{C}^{\mathsf{fgr}}$. One is to consider some techniques in other research fields that can work in high-dimension spaces. Probabilistic approaches in the research field of robotic motion planning

provides such kinds of techniques. These techniques involve but are not limited to Probabilistic Roadmap Method (PRM) [Kavraki et al., 1996] and Rapidly-Exploring Random Trees (RRT) [Lavalle and Kuffner, 2000]. Unfortunately, these techniques only guarantee weak probabilistic completeness ([Choset et al., 2005], pp.242 -246). Caging requires an object to be completely constrained by fingers. It should be depend on a certain probability. Therefore, this is not the best strategy. Another strategy is to reduce the dimension of $\mathbb{R}^{2n_f}$. I will take the second strategy and show the details in the next subsection.

5.1.2 The configuration space of finger formation

Let us review caging in Fig.5.1(b). When positions of fingers are the center of consideration, caging means the formation formed by the fingers cannot escape from target objects. That is to say, we do not need to consider the positions of every fingers. The positions of the formation and its orientation is enough for caging test. This conclusion reduces $\mathbb{R}^{2n_f}$ into a three-dimensional space. This space is different from both $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{fgr}}$. I name it $\mathcal{C}^{\mathsf{frm}}$ to indicate the difference. Fig.5.2 illustrates this $\mathcal{C}^{\mathsf{frm}}$ space. The planar two dimension of $\mathcal{C}^{\mathsf{frm}}$ denote the position of f_1 and the third dimension of $\mathcal{C}^{\mathsf{frm}}$ denotes the orientation of the whole formation. Consequently, a point in $\mathcal{C}^{\mathsf{frm}}$ represents the position and orientation of a fixed finger formation in work space.

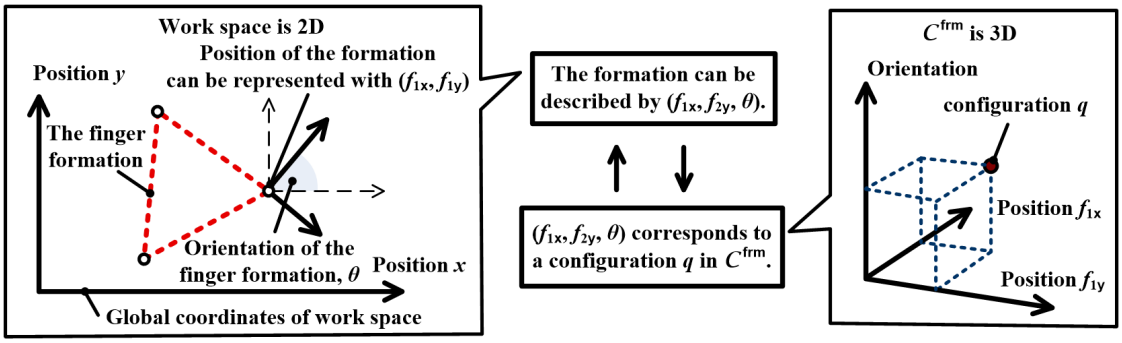
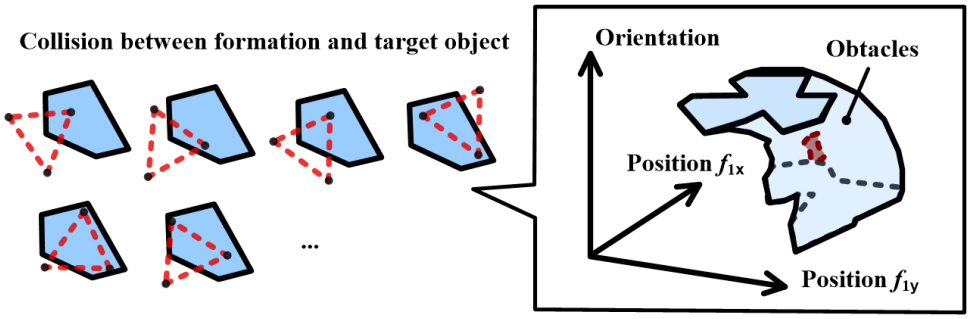


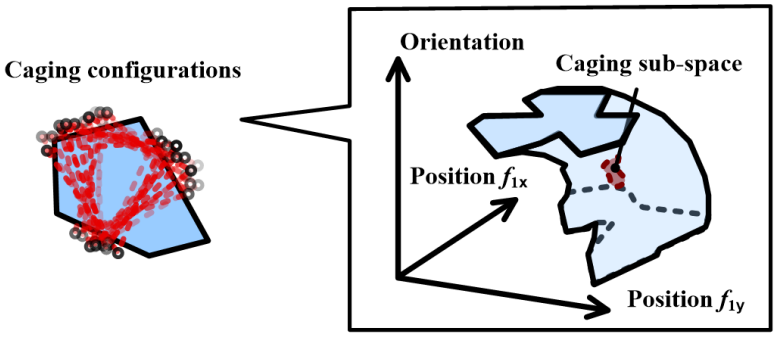
Figure 5.2: The configuration space of finger formation.

Unlike $\mathcal{C}^{\mathsf{obj}}$, the obstacle in $\mathcal{C}^{\mathsf{frm}}$ has no correspondence in $\mathcal{W}$ space. Take Fig.5.3(a) for instance. The obstacle in $\mathcal{C}^{\mathsf{frm}}$ is a compact set of configurations at which the formation collide with target objects. It does not correspond to a single finger like $\mathcal{F}_i$ but correspond to a whole formation. Caging means that (1) the formation does not collide with the target object and (2) the formation cannot go to an infinite configuration without colliding with the target object. In the $\mathcal{C}^{\mathsf{frm}}$, caging correspond to some caging sub-spaces. If a formation is at a configuration inside these caging sub-spaces, it fulfills the two conditions since (1) the caging sub-space is free and it is not obstructed and (2) current configuration cannot be connected to an infinite configuration unless it collides with the obstacles. Fig.5.3(b) illustrates a caging status and an caging sub-space. The caging sub-space in $\mathcal{C}^{\mathsf{frm}}$ is like the

caging sub-space in $\mathcal{C}^{\text{obj}}$. They do have certain relations. I will show their relations later after finishing solving the caging problems.



(a) The obstacles in C^{frm} denote a compact set of configuration points at which the finger formation collides with the target object



(b) All caging configurations belong to one or two compact regions in C^{frm}

Figure 5.3: Caging in the configuration space of finger formation.

$\mathcal{C}^{\text{frm}}$ has low dimensionality which makes it possible to be modeled. However, since the $\mathcal{C}^{\text{frm}}$ only encodes position and orientation of the whole formation, one modeled $\mathcal{C}^{\text{frm}}$ only corresponds to one finger formation. If the relative positions of fingers in a formation changes, the $\mathcal{C}^{\text{frm}}$ becomes totally different and we have remodel those caging sub-spaces and obstacles following Fig.5.3. That is a boring task. Moreover, when the target object changes, the obstacles should also be remodeled because checking whether formation and target objects collides depends on both finger formation and the shape of target objects. How to decouple the connection between $\mathcal{C}^{\text{frm}}$ and finger formation and decouple the connection between $\mathcal{C}^{\text{frm}}$ and obstacles become the bottleneck. I will start discussing these details from section 5.2. Before that, let us define the symbols that will be used in $\mathcal{C}^{\text{frm}}$ and compare $\mathcal{C}^{\text{obj}}$ and $\mathcal{C}^{\text{frm}}$.

The symbols that will be used in $\mathcal{C}^{\text{frm}}$ are as following.

$\mathcal{W}$	Work space. This is the same as $\mathcal{C}^{\text{obj}}$.
$\boldsymbol{\omega_i}$	A discretized grid of $\mathcal{W}$ space. Since $\mathcal{W}$ space is 2D, this grid can be represented by a coordinate $(\omega_{i_x}, \omega_{i_y})$ when granularity of discretization is small enough.

O The target object in $\mathcal{W}$ space. This is the same as $\mathcal{C}^{\mathsf{obj}}$.

$\boldsymbol{f_i}$ A point finger in $\mathcal{W}$ space. This is the same as $\mathcal{C}^{\mathsf{obj}}$. The $\boldsymbol{f_i}$ has two coordinate elements $\{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}\}$. Mathematically, $\boldsymbol{f_i} = \boldsymbol{\omega_i}$ when granularity of discretization is small enough. They both denote a position in 2D plane.

F A formation of fingers in $\mathcal{W}$ space. It is a set of $\mathcal{W}$ space point finger $\mathsf{F} = \{f_i = \{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}\} | i = 1, 2, \ldots, n_f\}$.

$\mathcal{C}^{\mathsf{frm}}$ The configuration space of finger formation.

$\boldsymbol{q}^{\mathbf{frm}}$ $\boldsymbol{q}^{\mathbf{frm}}$ is a configuration in $\mathcal{C}_{\mathsf{frm}}$. Since we have reduced the high dimensional $\mathcal{C}^{\mathsf{fgr}}$ into three-dimensional $\mathcal{C}^{\mathsf{frm}}$, $\boldsymbol{q}^{\mathbf{frm}} = \{q_\mathsf{x}^{\mathsf{frm}}, q_\mathsf{y}^{\mathsf{frm}}, q_\theta^{\mathsf{frm}}\}$ where the first two items denote the position of $\boldsymbol{f_1}$, namely $q_\mathsf{x}^{\mathsf{frm}} = f_{1_\mathsf{x}}$ and $q_\mathsf{y}^{\mathsf{frm}} = f_{1_\mathsf{y}}$. and the last item denotes the orientation of the formation.

$\mathsf{F}[\boldsymbol{q}^{\mathbf{frm}}]$ $\mathsf{F}[\boldsymbol{q}^{\mathbf{frm}}]$ denotes a finger formation F at configuration $\boldsymbol{q}^{\mathbf{frm}}$. Mathematically, it represents a set of 2D positions in $\mathcal{W}$ space occupied by F. For example, when $\mathsf{F} = \{\boldsymbol{f_i} = \{f_{i_\mathsf{x}}, f_{i_\mathsf{y}}\} | i = 1, 2, \ldots, n_f\}$, $\mathsf{F}[\boldsymbol{q}^{\mathbf{frm}}] = \{R^{(q_\theta^{\mathsf{frm}})} \cdot \{f_{j_\mathsf{x}} - f_{1_\mathsf{x}} + q_\mathsf{x}^{\mathsf{frm}}, f_{j_\mathsf{y}} - f_{1_\mathsf{y}} + q_\mathsf{y}^{\mathsf{frm}}, 1\} | 1 < j \leq n_f\}$. Note that this expression should be performed in with homogeneous coordinates with an augmented "1" at the end of $\{f_{j_\mathsf{x}} - f_{1_\mathsf{x}} + q_\mathsf{x}^{\mathsf{frm}}, f_{j_\mathsf{y}} - f_{1_\mathsf{y}} + q_\mathsf{y}^{\mathsf{frm}}\}$, namely $\{f_{j_\mathsf{x}} - f_{1_\mathsf{x}} + q_\mathsf{x}^{\mathsf{frm}}, f_{j_\mathsf{y}} - f_{1_\mathsf{y}} + q_\mathsf{y}^{\mathsf{frm}}, 1\}$ The rotation matrix $R^{(\theta)}$ follows the same definition as section 3.1.

$\mathcal{C}_{\mathsf{otl}}^{\mathsf{frm}}$ The obstacles in $\mathcal{C}^{\mathsf{frm}}$. Mathematically, it is some sub-spaces/compact sets of configurations $\boldsymbol{q}^{\mathbf{frm}}$ where $\mathsf{F}[\boldsymbol{q}^{\mathbf{frm}}] \cap \mathsf{O} \neq \emptyset$.

$\mathcal{C}_{\mathsf{free}}^{\mathsf{frm}}$ All free sub-spaces in $\mathcal{C}^{\mathsf{frm}}$. Mathematically, it is the complementary space of $\mathcal{C}_{\mathsf{otl}}^{\mathsf{frm}}$. Namely $\mathcal{C}_{\mathsf{free}}^{\mathsf{frm}} = \mathcal{C}^{\mathsf{frm}} \setminus \mathcal{C}_{\mathsf{otl}}^{\mathsf{frm}}$.

Readers may compare these notations with those defined in section 3.1 for comparison of $\mathcal{C}^{\mathsf{frm}}$ and $\mathcal{C}^{\mathsf{obj}}$. Recall that caging in $\mathcal{C}^{\mathsf{frm}}$ means two conditions and we can now express the two conditions formally with the defined notations alike the expression in $\mathcal{C}^{\mathsf{obj}}$. When caging is achieved, $\mathcal{C}_{\mathsf{free}}^{\mathsf{frm}}$ is divided into several disconnected sub-spaces. Most of the sub-spaces are enclosed by obstacles. The caging sub-space in Fig.5.3 is one enclosed example. However, it is a special case since it is the only caging sub-space. Generally, there would be always more than one caging sub-space. I denote these caging sub-spaces by $\mathcal{C}_{\mathsf{fc}}^{\mathsf{frm}} = \bigcup_{i=1}^{u} \mathcal{C}_{\mathsf{fc_i}}^{\mathsf{frm}}$ in accordance with the caging sub-spaces in $\mathcal{C}^{\mathsf{obj}}$. Note that $\mathcal{C}^{\mathsf{frm}}$ is inherently different from $\mathcal{C}^{\mathsf{obj}}$. object is convex, u may be still larger than 1. One other disconnected sub-spaces is the complementary of $\mathcal{C}_{\mathsf{fc}}^{\mathsf{frm}}$ which can be denoted by $\mathcal{C}_{\mathsf{ff}}^{\mathsf{frm}}$. In summary, $\mathcal{C}_{\mathsf{free}}^{\mathsf{frm}} = \mathcal{C}_{\mathsf{fc}}^{\mathsf{frm}} \cup \mathcal{C}_{\mathsf{ff}}^{\mathsf{frm}} = (\bigcup_{i=1}^{u} \mathcal{C}_{\mathsf{fc_i}}^{\mathsf{frm}}) \cup \mathcal{C}_{\mathsf{ff}}^{\mathsf{frm}}$, $\mathcal{C}_{\mathsf{fc}}^{\mathsf{frm}} \cap \mathcal{C}_{\mathsf{ff}}^{\mathsf{frm}} = \emptyset$. Whether O can be caged by a finger formation F can be validated by the following expression.

$$(\mathcal{C}_{\mathsf{free}}^{\mathsf{frm}} = (\bigcup_{i=1}^{u} \mathcal{C}_{\mathsf{fc_i}}^{\mathsf{frm}}) \cup \mathcal{C}_{\mathsf{ff}}^{\mathsf{frm}}) \wedge (\boldsymbol{q_0}^{\mathbf{frm}} \in \mathcal{C}_{\mathsf{fc_k}}^{\mathsf{frm}}) \wedge (|\mathcal{C}_{\mathsf{fc_k}}^{\mathsf{frm}}| > 1), 1 \leq k \leq u \tag{5.1}$$

This expression is nearly the same as expression (3.1). $|\mathcal{C}^{\mathsf{frm}}_{\mathsf{fc_k}}|$ means the cardinality of $\mathcal{C}^{\mathsf{frm}}_{\mathsf{fc_k}}$. When $|\mathcal{C}^{\mathsf{frm}}_{\mathsf{fc_k}}| > 1$, the target object is either in the state of caging or in the state of contact caging. When $|\mathcal{C}^{\mathsf{frm}}_{\mathsf{fc_k}}| = 1$, the target object is in the state of immobilization. Note that $\boldsymbol{q}^{\mathsf{frm}}_{\boldsymbol{0}}$ in expression (5.1) is the initial configuration of F. It is not the initial configuration of O.

Everything seems to be following the same rule as $\mathcal{C}^{\mathsf{obj}}$. Unfortunately, we encounter the bottleneck when trying to model and calculate $\mathcal{C}^{\mathsf{frm}}_{\mathsf{fc}}$. Recall Fig.3.1 and Fig.3.2. We can model $\mathcal{F}_i$ in $\mathcal{C}^{\mathsf{obj}}$ with wireframe modeling since there are correspondence between each layer of $\mathcal{F}_i$ and orientation of the target object. We only need to consider one position, name the position $\boldsymbol{f_i}$. In contrast, there is no correspondence between each layer of $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$ and orientation of the formation. That is because not only the orientation but also the position of the formation may change during collision with a target object. Fig.5.4 shows the bottleneck. I propose to solve this bottleneck with a space mapping technique widely used in motion planning. It belongs to **solid modeling** and it has the advantage of decoupling from specific shapes of targe objects. Let us view its details in the next few sections.

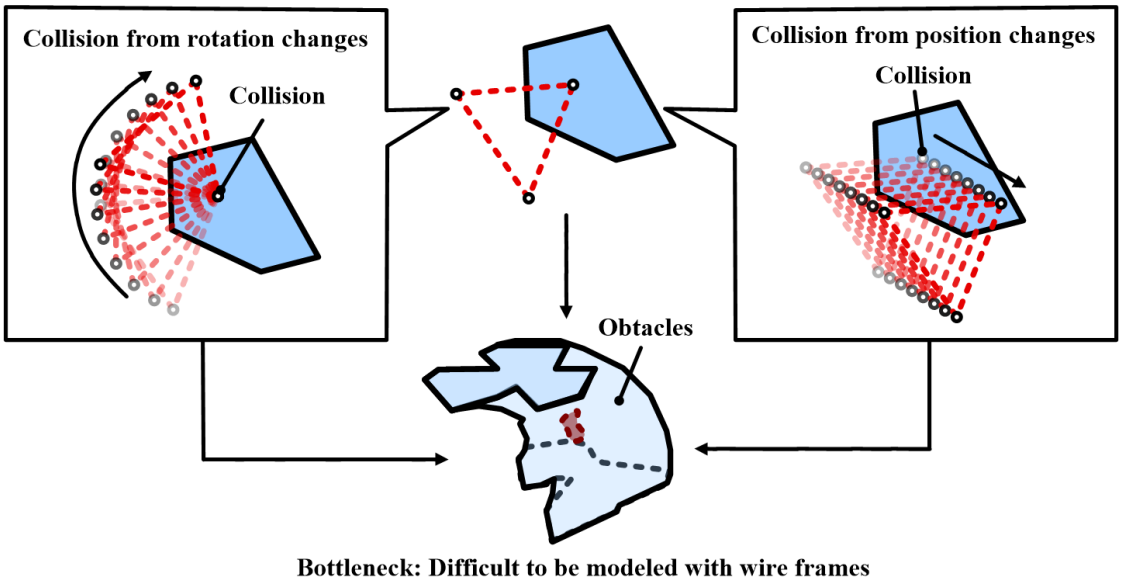


Figure 5.4: $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ is difficult to be modeled with wireframe modeling.

5.2 Space Mapping

This part is in publishing.

Chapter 6

Applications II – Hand Design

Caging can offer robustness to uncertainties in grasping. If a robotic hand is designed based on the idea of caging, it would probably work well with noisy perception devices and low-quality control. In this chapter we will see an application which design ands implements a gripping hand based on the caging algorithm in $\mathcal{C}^{\mathsf{frm}}$. The gripping hand is concise and offers a low-cost alternative to co-operate with noisy data and low-quality control. This chapter includes two sections. In the design section, I will show how I successfully simplify the number of actuators into one by quantitatively analyzing finger formations with caging tests conducted on both random objects and objects from MPEG-7 shape database. Following the simplified one-actuator design I will in the implementation section show the implementation and demonstration of a gripping hand by modifying a SCHUNK RH707 gripper and carried out experiments with a manipulator built on the Neuronics Katana arm. The one-actuator gripping hand could work with the Swiss Ranger and both convex and concave objects. It demonstrates the merits of caging, especially the advantages of caging in $\mathcal{C}^{\mathsf{frm}}$ over $\mathcal{C}^{\mathsf{obj}}$.

6.1 Designing a Gripping Hand by Using Caging

Although lots of theories have been developed in the research field of manipulation and grasping. These theories involve not only some of the previously discussed topics like form/force closure, caging and their optimization but also some of undisclosed topics like enveloping [Trinkle et al., 1988]. However, a large gap exists between these theories and real-world designs and applications. For example, robotic hand dimensions and finger numbers are neither designed according to mathematical formulae of form/force closure nor designed according to perception devices. They are, in most cases, decided by (1) purpose of usage, (2) biomimetic study or (3) mechanical constraints and empirical experiences. In this part, I propose the design a gripping hand according to the theory of caging. The hand has only one actuator. It is concise, low-cost and owns all merits from caging (like robustness to uncertainties).

6.1.1 Retrospecting the hand design

Firstly, let us retrospect the contemporary works of hand design. There are two problems regarding the design of a robotic hand. The first problem is its complexity. Let us compare the following three representative examples — (a) The Schunk JGZ industrial gripper [SCHUNK, 2013], (b) the Barrett Hand [Barrett Technology, 2013] and (c) the Robonaut Hand [Bridgwater et al., 2012]. Note that there are many alike candidates whereas I take these three for instance. The three hands differ significantly in DoFs (Degree of Freedoms), actuation types and purpose of usage. The JGZ gripper has one DoF. It is fully actuated and designed for industrial usage. The Barrett Hand has four DoFs. It is under-actuated and designed to manipulate versatile objects. The Robonaut Hand has twelve DoFs which mimics a human hand. It is dexterous and designed for tele-operation. These hands are designed either according to their usage, biomimetic study or empirical experiences. Comparing with the design strategies of these hands, I would like to take into account of caging and design a hand that is both high in generality and low in DoFs.

Design belongs to the **mechanism** aspect discussed in the beginning of this thesis. But we do need to take into account both **sensing** and **mechanics** as well. Reference [Pollard, 2010] offers a good summary of hand design and the most related works to my case are [Zhang and Goldberg, 2001], [Dollar and Howe, 2010] and [Hammond et al., 2012]. Robotic hands in these works are designed according to "constraining" models. Specially, the SDM hand presented in [Dollar and Howe, 2010] follows principles of enveloping [Trinkle et al., 1988][Dollar and Howe, 2005] and won great success in grasping in unstructured environments. [Hammond et al., 2012] discusses in detail how to reduce motor number and designs an under-actuated robotic hand. Like these works, I also simplify and design our gripping hand based on a "constraining" model. The "constraining" model is caging. By performing caging tests on random objects and objects from MPEG-7 shape data base libraries, I find an optimized actuator and finger setting that have highest successful caging rate. The optimized actuator and finger setting help to reduce the number of actuators into one. At the same time, it owns all merits from caging and endows us the potential to perform safe and robust grasping.

The second problem is integration with perception devices. We have reviewed the popular works that detect objects and synthesize grasping in section 3.3.3.2. In all the techniques and devices employed by those works, database matching is effective in grasping known objects but it is not as satisfying with unmodeled targets. RGB camera is affordable and applied to many industrial systems. However it suffers a lot from unstructured environments. Depth sensors can be summarized into two categories, namely scanners and rangers. Scanners have high precision as well as high costs. Rangers are much cheaper. Examples of rangers involve the ToF-based (Time of Flight) Swiss Ranger or structure light-based KINECT. These two devices were discussed in detail in section 4.1.2.1. With the help of caging, the hand is expected to work with the Swiss Ranger. Noises of the Swiss Ranger are +/-10mm in depth and +/-7mm in horizontal plane. I believe if the hand could work with the Swiss Ranger, it is suitable to most applications.

6.1.2 A basic design based on qualitative analysis

There are some design candidates that are in well accordance with the caging theory. In section 2.2 we have known that **the number of fingers that are sufficient to cage an object in n_{dim} space is $n_{\mathsf{dim}}+1$ to $2n_{\mathsf{dim}}$**. That is to say, we should install at least 2×2 = 4 fingers to cage any 2D shape. It is true that the performance becomes better if there are more fingers. However, I prefer the least number 4 since I would like to reduce the complexity as much as possible.

After deciding the number of fingers, the remaining problem is how to install the four fingers and how to actuate them. One example is the distributed end-effector discussed in the first subsection of Chapter 4. In that design, fingers are attached, actuated and detached sequentially by the single x-y-θ actuator and only three motors are required. Although the design lowers system cost, it introduces a time-consuming attaching-actuating-detaching procedure which slows down operation. Unlike that design, I will in this part consider the installation of actuators by quantitative evaluation with caging algorithms.

Fig.6.1 shows the candidate installations of actuators. Note that I do not consider the shapes of fingers here and they are therefore rendered as simply poles. The first candidate, Fig.6.1(a), is the most intuitive installation. It endows distributed control to each finger and requires as many as eight actuators. The distributed end-effector is actually a variation of it. The last candidate, Fig.6.1(d), drives four fingers simultaneously. It requires only one actuator and the SDM hand [Dollar and Howe, 2010] follows its principle. The last candidate fully ensure equal inter-finger distances which adds strong bias to **rotational constraints** (recall section 3.3.1). However, it has no flexibility for **translational caging**. The SDM hand solved this inflexibility problem by fabricating delicate under-actuated fingers (say, delicate shapes of fingers). In our case which aims at a concise "gripping" hand, or namely a hand with pole-like fingers, Fig.6.1(b) and Fig.6.1(c) are better choices.

The difference between Fig.6.1(b) and Fig.6.1(c) are their levels of biases towards equal inter-finger distances. Fig.6.1(b) has higher flexibility in position control and it holds more bias towards **translational caging**. Nevertheless, three actuators complicate the gripping system. I prefer choosing the two-actuator candidate Fig.6.1(c) as **the basic design**. Fig.6.2 shows in detail of how this basic design works. Each actuator in this installation drives two pairs of fingers and either caging or grasping by caging can be performed by this design.

The basic design is based on qualitative analysis. We would like find some quantitative supports to demonstrate its advantages. In the next section, we will quantitatively analyze this basic design with the caging algorithm in $\mathcal{C}^{\mathsf{frm}}$ and various objects and see if it has high successful rates in caging. If the basic design has high successful caging rates, it is considered to be a satisfying candidate to manipulation with caging and inherits the merits of caging with the help of caging or grasping by caging algorithms.

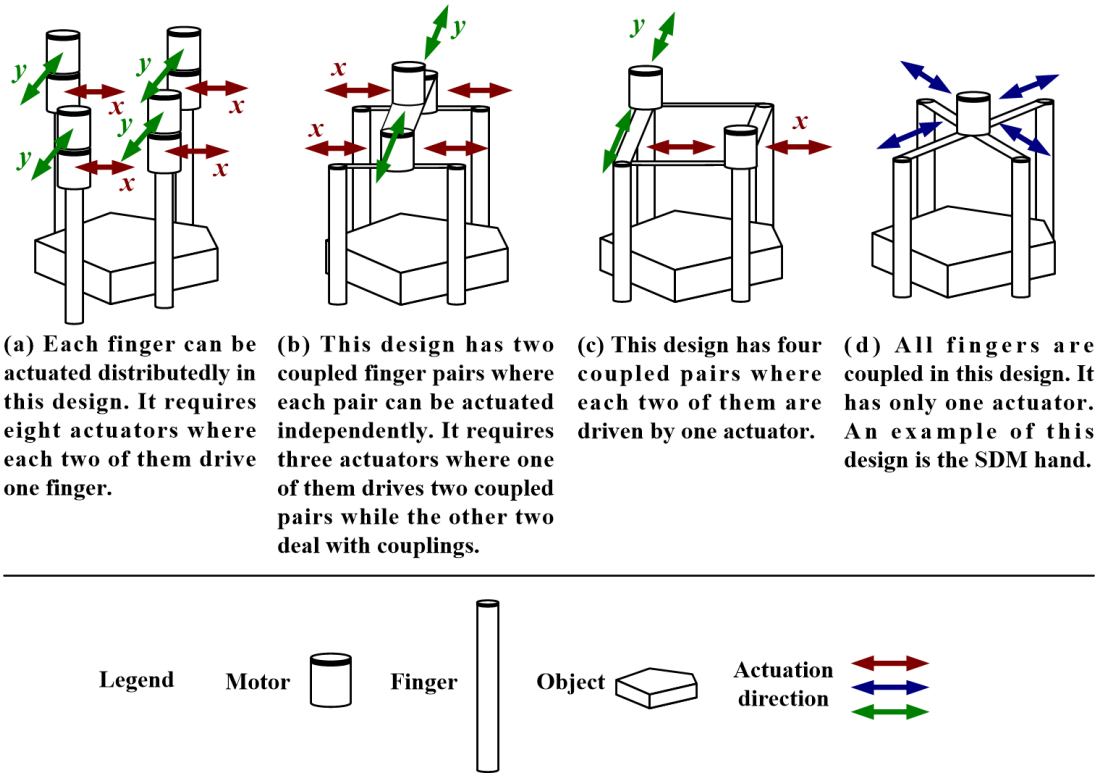


Figure 6.1: Four candidate installations of actuators.

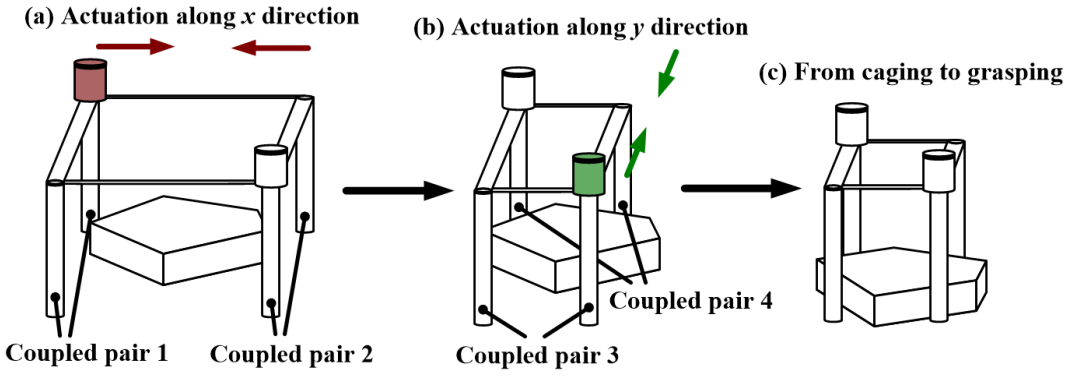


Figure 6.2: The basic design and one of its caging or grasping by caging procedure.

6.1.3 Quantitative analysis of the basic design

6.1.3.1 Objects for quantitative analysis

In order to find some quantitative supports for the basic design, I employ an object generator to randomly generate some shapes and quantitatively evaluate the performance of the basic design with these shapes and caging tests.

It is difficult for a random generator to cover any 2D shapes but we try to enlarge its coverage as much as possible. This is done by setting a parameter n_s, namely the number of sectors. See bold segments in Fig.6.3(a) for details. The object generator generates random shapes inside a background circle decided by different n_s parameters. The figures in in Fig.6.3(b) exemplify some randomized objects. Three groups of randomized objects are generated according to different parameter settings. Readers may refer to Alg.1 to better understand the roles of n_s. The algorithm randomize a position along radial direction at each sector. This position is saved as one vertex of the target object. Final list of target object vertice indices are returned as $\mathsf{P}_{\mathsf{bdry}}$.

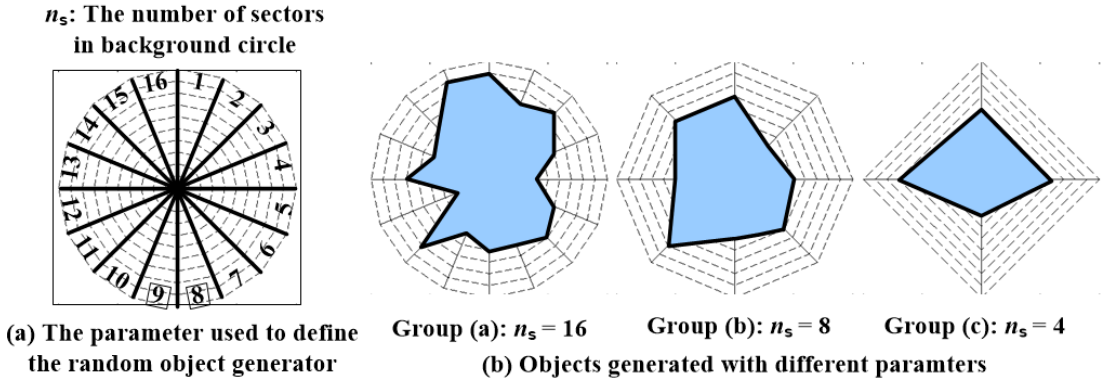


Figure 6.3: The random object generator.

This object generator is subject to the following limitations. (1) It cannot generate shapes with inner holes. This limitation is acceptable since I would like to constrain the caging into squeezing caging. (2) It may require thousands of randomization before reaching a convincing conclusion. In order to conquer the second limitation, we generate objects by three groups. Each group is randomized according to different n_s. Their details are as following. **Group (a): n_s=16.** We expect the random shapes generated in this group may be either smooth (small probability) or with sharp prostrusion (high probability). Shapes in this group should be, in most cases, easy to be caged owing to their protrusions. **Group (b): n_s=8.** The random shapes generated in this group has higher bias towards smooth objects and general polytopes while have less bias towards protrusion. We expect that shapes in this group become more difficult to be caged comparing with Group (a). **Group (c): n_s=4.** The random shapes in this group help to fill up the loss of Group(a) and Group (b). For instance, it has high probability of generating quadrilaterals and trilaterals which

Algorithm 1: The random object generator

Data: n_s
Result: P_{bdry}
begin
  $\mathsf{P}_{bdry} \leftarrow \emptyset$
  for $i \in \{0 : n_a\}$ **do**
    /*Randomly select a position along radial direction*/
    $p_i \leftarrow$randomize a number between 0 and 9
    $\mathsf{P}_{bdry} \leftarrow \mathsf{P}_{bdry} \bigcap p_i$
  end
  return P_{bdry}
end

are hardly generated in Group(a) and Group(b). Shapes in Group (c) should be easier to be caged comparing with Group (b) as their inner angles become sharper. We expect that comparing with a single-group generator, generating shapes by these three groups with different parameter settings could offer convincing conclusions with fewer randomizations.

Besides the random object generator, I further evaluate the performance of the basic design with objects extracted from the MPEG-7 shape library (see Fig.6.4). Shapes in the MPEG-7 library are based on real-world objects, they are more realistic comparing with our random generator. These objects can further confirm the performance of our basic design.

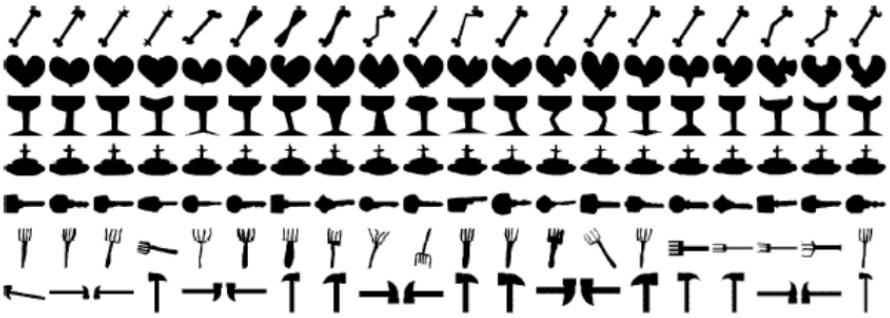


Figure 6.4: Some objects from the MPEG-7 shape library.

In total, we perform caging tests on 1000 shapes from Group (a), 1000 shapes from Group (b), 1000 shapes from Group (c) and 1100 shapes from the MPEG-7 shape library.

6.1.3.2 Results and analysis

The caging test algorithm in $\mathcal{C}^{\mathsf{frm}}$ is limited to one finger formation and one object. It is not applicable to testing the performance of a hand which could form into infinite number of

formations. I propose to solve this problem by defining 20 candidate formations and assume these 20 formations could be representative and cover the infinite number of cases. Fig.6.5 shows the 20 formations. It involves five columns of pair 1 and pair 2 along x axis and four rows of pair 3 and pair 4 along y axis. Readers may refer to the texts in Fig.6.2 to recall the pairs 1, 2, 3 and 4. Note that since x and y axes of the basic design are symmetric, 5 *columns* $\times$ 4 *rows* complements 4 *rows* $\times$ 5 *columns*. In this way, using 5×4=20 formations is more efficient than using symmetric multiplications like 5×5 or 4×4. Note that the background of the object generator is illustrated together with the finger positions in Fig.6.5(b). Readers may compare the dimensions of the twenty formations and the size of randomly generated objects by referring to it.

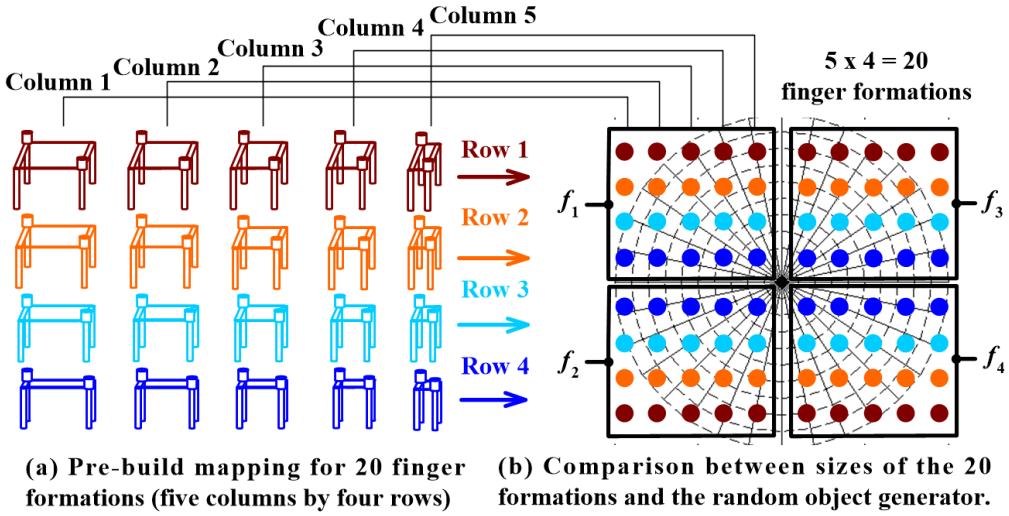


Figure 6.5: The 20 representative finger formations of the basic design.

I quantitatively evaluate the 20 formations of **the basic design** with the 3000 random objects and 1100 MPEG-7 shapes and the caging test algorithm in $\mathcal{C}^{\mathsf{frm}}$. That is to say, we carry out (1000+1000+1000+1100)*20 = 82000 caging tests. This is possible owning to the rapidness and completeness of improved space mapping. Readers may review section 5.3.2.1 to recall the details of caging test in $\mathcal{C}^{\mathsf{frm}}$.

Fig.6.6(a) shows the total successful caging rate of the 20 formations. In this result, if an object can be caged by any of the 20 formation, then the basic design is assumed to be able to cage that object. The result here is obtained by setting n_g=150×150, m=72, τ_{rng_r}=25 and τ_{rng_r}=$+\infty$.

The result indicates that the basic design can cage objects with more than 90% successful rate and we can draw a conclusion that most "normal" objects, either they have convex, concave or smooth boundaries, can be caged by the basic design. Exceptions are those tiny or thin cases shown in Fig.6.6(b). The basic design is not suitable to cage those objects. Actually, those objects are not suitable for general caging. An intuitive example is caging

an eel in fishing. Eels cannot be captured by general fishing net and fish men use special net to cage them.

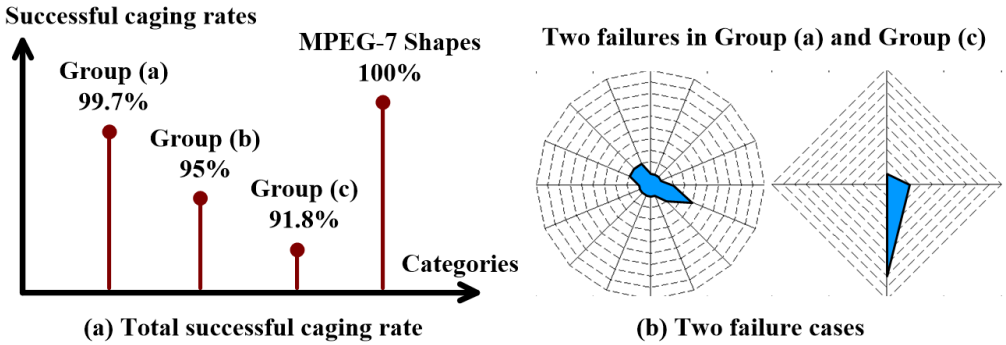


Figure 6.6: The total successful caging rate and two examples of failure.

Note that the objects from MPEG-7 shape library is resized to fit into the background of random object generator. That means the objects from MPEG-7 library cannot be too small. That could be the reason why they can always be caged with 100% successful rate. Objects from our random generator are harder to be caged comparing with MPEG-7 shapes.

More than 90% is quantitatively good performance. However, I wonder if it can be further simplified. Dollar's SDM hand has only one motor, whereas the basic design requires two. In the next section I will further analyze the total successful rate and check if the basic design can be further simplified.

6.1.3.3 Further simplification

The results in Fig.Fig.6.6(a) are the total rates of 20 formations. In another word, the basic design is considered to be able to cage an object as long as a single one from the 20 formation can cage it. This is reasonable as the basic design has two actuators and can be actuated into any of the 20 formations.

If we would like to further simplify the basic design, the most intuitive way is to to delete one actuator. However, deleting one actuator changes the 20 formations. For example, when the x-actuator, namely the red one in Fig.6.2, is deleted, the basic design can no longer be actuated from one column to another. That means the $5\times4 = 20$ formations become a single column of 4 formations. When the y-actuator, namely the green one in Fig.6.2, is deleted, the basic design can no longer be actuated from one row to another. That means the $5\times4 = 20$ formations become a single row of 5 formations.

Suppose we delete the y-actuator for simplification. Note that deleting the x-actuator works in the same way as x axis and y axis are symmetric. Here I delete y because we discretized the formations into 5×4. Deleting the actuator along y direction leaves 5 formations along x direction. The resolution is larger. If it were 4×5, it would be a better choice to delete the actuator along x direction. After deleting the y-actuator, the basic design can

no longer be actuated from one row to another and we can only keep a single row. In that case, the designing problem becomes which row should we retain to ensure high successful caging rates. This problem could be solved by further analyzing the total successful rate in Fig.6.6(a). Fig.6.7 shows the further analysis of Fig.6.6(a). In this figure, the decomposed rates of total results are illustrated. Successful caging rates of each row are shown respectively in the left part. They indicate the successful caging rates of each row of the 20 formations on the three random groups and each row is rendered with different color bars. The right figure is a copy of the left part of Fig.6.6(a) for easy comparison.

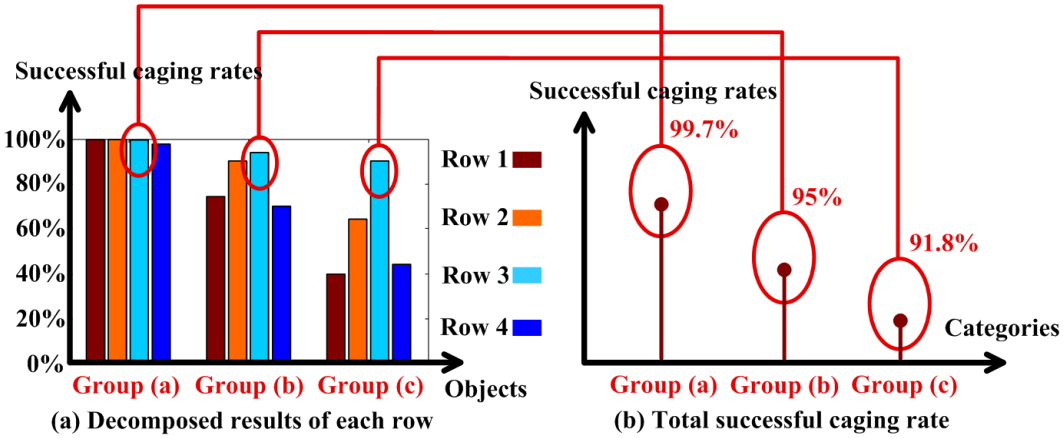


Figure 6.7: The successful caging rates of each row.

As is denoted in Fig.6.7, the third row of formations, namely the row with cyan color, has highest successful caging rates. It is also the key row of the 20 formations. Readers can compare successful caging rates of the "cyan" bars with total successful caging rates in the left part of Fig.6.7 for better comprehension. Successful rates of the "cyan" row on the three groups of random objects, namely the "cyan" bar in the left part of Fig.6.7, are nearly the same as the total successful rates of all 20 formations. That is to say, **the randomized shapes are mainly caged by finger formations in the "cyan" row and we can delete the other rows without much loss of successful rates**.

Consequently, we can get the following simplification rule. **The basic design can be further simplified by fixing one actuator without much loss of successful caging rates. The inter-finger distance of the fixed actuator should be around the "cyan" row.** Fig.6.8 shows the idea. After simplification, only the actuator along x axis remains.

Let us retrospect this simplified design and compare it with Fig.6.1(d). They both have only one actuator. But is the simplified design really better? A confirming conclusion can be drawn by deeper review of Fig.6.8. Fig.6.9 shows in depth the deeper details of the analysis in Fig.6.7. In this figure, not only the successful caging rates of each row but also the successful caging rates of each formation are illustrated. The finger formation with higher successful caging rates have larger circle sizes. Successful caging rates of the simplified design are roughly the sum of "cyan" circles. I use the word "roughly" because there is redundancy

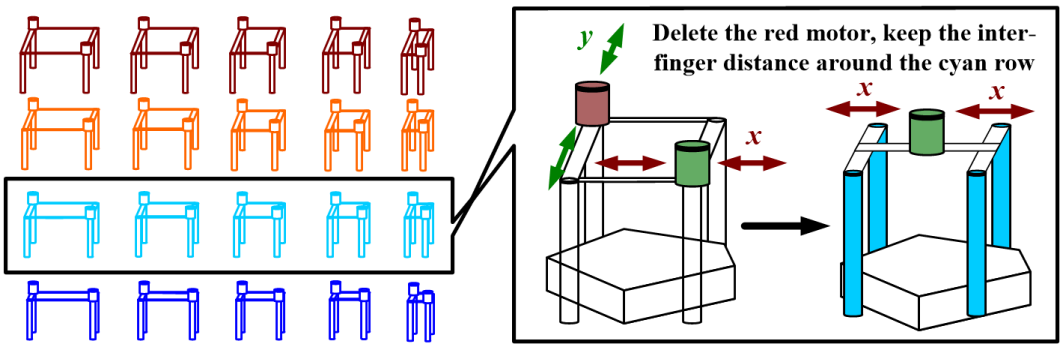


Figure 6.8: Further simplification of the basic design.

in addition. This is a rough analysis. In contrast, successful caging rates of the design in Fig.6.1(d) is roughly the sum of diagonal circles. Note that like the simplified design, there is redundancy in addition. The simplified design has larger sum of circle sizes and higher successful caging rate. Therefore, it is indeed a better design comparing with Fig.6.1(d).

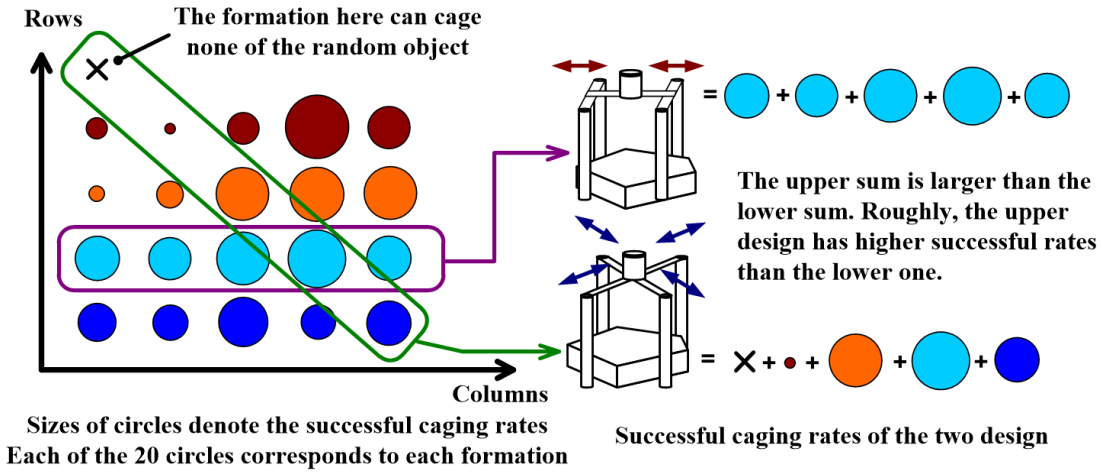


Figure 6.9: Comparison of the simplified design and the design in Fig.6.1(d).

6.2 Implementation of the Design

This part is in publishing.

Part IV

In-depth $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$

Chapter 7

In-depth analysis of $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$

7.1 The Relationship Between $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$

We have explored the caging problem in two spaces. The first one is the configuration space of target object, namely $\mathcal{C}^{\mathsf{obj}}$. The second one is the configuration space of finger formation $\mathcal{C}^{\mathsf{frm}}$. These two spaces are actually equal to each other with a linear transformation. Let us analyze their relationship in this section. The symbols used here are the same as those used in previous contexts of this thesis.

7.1.1 The expressions of $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$

In order to compare these two spaces, we discretize $\mathcal{C}^{\mathsf{obj}}$ into voxels like $\mathcal{C}^{\mathsf{frm}}$ and express $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$ and $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ as two sets of voxels and compare the relationship between them.

Firstly, I am going to deduce the expressions of $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$. According to Fig.3.1, we need to choose a pivot point to build the correspondence between a $\mathcal{W}$ space object to a $\mathcal{C}^{\mathsf{obj}}$ configuration. This pivot point can be chosen arbitrarily so that I use the first coordinate of target object $v_{1_\mathsf{x}}, v_{1_\mathsf{y}}$ as the pivot point[1]. Consequently, the initial configuration of a target object is $\boldsymbol{q}^{\mathsf{obj}}_0 = \{q^{\mathsf{obj}}_{0_\mathsf{x}}, q^{\mathsf{obj}}_{0_\mathsf{y}}, q^{\mathsf{obj}}_{0_\theta}\} = \{v_{1_\mathsf{x}}, v_{1_\mathsf{y}}, 0\theta\}$. After one counter-clockwise rotation (CCW) layer, the configuration becomes $\boldsymbol{q}^{\mathsf{obj}}_1 = \{v_{1_\mathsf{x}}, v_{1_\mathsf{y}}, 1\theta\}$. Likewise, after one clockwise rotation layer (CW) the configuration becomes $\boldsymbol{q}^{\mathsf{obj}}_{-1} = \{v_{1_\mathsf{x}}, v_{1_\mathsf{y}}, -1\theta\}$.

If we denote all vertices of the target object with $\mathsf{V} = \{\{v_{1_\mathsf{x}}, v_{1_\mathsf{y}}\}, \{v_{2_\mathsf{x}}, v_{2_\mathsf{y}}\}, \ldots, \{v_{n_{v\mathsf{x}}}, v_{n_{v\mathsf{y}}}\}\}$, then the $\partial\mathsf{O}[\{v_{1\mathsf{x}}, v_{1\mathsf{y}}, 0\theta\}]$ (see Fig.3.1) can be expressed as following.

$$\begin{aligned} T^{(-v_{1_\mathsf{x}}, -v_{1_\mathsf{y}})}\mathsf{V} =& \{T^{(-v_{1_\mathsf{x}}, -v_{1_\mathsf{y}})}v_1, T^{(-v_{1_\mathsf{x}}, -v_{1_\mathsf{y}})}v_2, \ldots, T^{(-v_{1_\mathsf{x}}, -v_{1_\mathsf{y}})}v_m\} \\ =& \{v_1 - v_1, v_2 - v_1, \ldots, v_m - v_1\} \end{aligned} \tag{7.1}$$

[1]This is different from the contexts in Chapter 3. That is because in Chapter 3, I chose a pivot point different from fingers to make clear the illustrations. In contrast, I choose $v_{1_\mathsf{x}}, v_{1_\mathsf{y}}$ in this section to make it coherent with our deduction of $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$.

$$\begin{aligned}\partial\mathsf{O}[\{v_{1\mathsf{x}}, v_{1\mathsf{y}}, 0\theta\}] &= R^{(0\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V} \\ &=\{R^{(0\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\boldsymbol{v_1}, R^{(0\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\boldsymbol{v_2}, \ldots, R^{(0\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\boldsymbol{v_{n_v}}\}\end{aligned} \tag{7.2}$$

Here, $T^{(\delta_x,\delta_y)}$ means the translation matrix with respect to a shifting pair (δ_x, δ_y). The shifting pair indicates the shift between two grids of $\mathcal{W}$ space, namely $(\delta_x, \delta_y) = \omega_i - \omega_j = (\omega_{i_\mathsf{x}} - \omega_{j_\mathsf{x}}, \omega_{i_\mathsf{y}} - \omega_{j_\mathsf{y}})$. $T^{(\delta_x,\delta_y)}$ can be written as a matrix in the following form.

$$T^{(\delta_x,\delta_y)} = T^{(\omega_i-\omega_j)} = \begin{bmatrix} 1 & 0 & \omega_{i_\mathsf{x}} - \omega_{j_\mathsf{x}} \\ 0 & 1 & \omega_{i_\mathsf{y}} - \omega_{j_\mathsf{y}} \\ 0 & 0 & 1 \end{bmatrix} \tag{7.3}$$

The expression (7.1) moves rotation center to the pivot point $\boldsymbol{v}_1$. Then, expression (7.2) rotates the target object vertices according to its orientation. Since expression (7.2) aims to express $\mathsf{O}[\{v_{1\mathsf{x}}, v_{1\mathsf{y}}, 0\theta\}]$, the rotation matrix is set to $R^{(0\theta)}$. At another layer, say layer 1, expression (7.2) would become $\partial\mathsf{O}[\{v_{1\mathsf{x}}, v_{1\mathsf{y}}, 0\theta\}] = R^{(1\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}$.

Based on these expressions and the expressions (3.2), (3.3), (3.4) and (3.6) in Chapter 3, we can express $\mathcal{F}_1[q^{\mathsf{obj}}_{i_\theta}]$ as expression (7.4). Note that there is nothing new here. Expression (7.4) is actually exactly the same as expression (3.6) except that I change $\partial\mathsf{O}[\{f_{1\mathsf{x}}, f_{1\mathsf{y}}, q^{\mathsf{obj}}_{j_\theta}\}]$ into the form of vertices.

$$\mathcal{F}_1[q^{\mathsf{obj}}_{i_\theta}] = T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}R^{\pi}R^{(i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V} \tag{7.4}$$

Consequently, the whole $\mathcal{F}_1$ can be expressed as the union of all layers, namely $\bigcup_{i=-m}^{m}\mathcal{F}_1[q^{\mathsf{obj}}_{i_\theta}]$. Expression (7.5) shows it. Like the relationship between expression (7.4) and expression (3.6), the expression (7.5) is exactly the same as expression (3.7).

$$\bigcup_{i=-m}^{m} T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}R^{\pi}R^{(i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V} \tag{7.5}$$

Likewise, the $\mathcal{F}^{\mathsf{obj}}_j$ of any other finger f_j is composed of

$$\bigcup_{i=-m}^{m} T^{(f_{j\mathsf{x}},f_{j\mathsf{y}})}R^{\pi}R^{(i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V},\ j = 2, 3, \ldots, m \tag{7.6}$$

In summary, $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$ is the set of voxels enclosed by expression (7.5) and (7.6).

7.1.2 The expressions of $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$

Then, let us deduce the expressions of $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$. According to Fig.5.2, we denote the configuration of finger formation by the coordinates of its first finger and the orientation of the whole finger formation. Therefore, the initial configuration of a finger formation is $\boldsymbol{q}^{\mathsf{frm}}_0 = \{f_{1\mathsf{x}}, f_{1\mathsf{y}}, 0\theta\}$. In accordance with the rules in rotating a target object, after one CCW rotation layer, the

configuration of finger formation rotates reversely and becomes $\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, 1\theta\}$. After one CW rotation step, it becomes $\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, -1\theta\}$. Like $\partial\mathsf{O}[\{v_{1_\mathsf{x}}, v_{1_\mathsf{y}}, 0\theta\}]$, the $\mathsf{F}[\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, 0\theta\}]$ can be expressed by the following two expressions where the first one is to move rotation center to the first finger position while the second one is to rotate the finger formation according to its orientation.

$$\begin{aligned} T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})}\mathsf{F} =& \{T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})} f_1, T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})} f_2, \ldots, T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})} f_{n_f}\} \\ =& \{\boldsymbol{f}_1 - \boldsymbol{f}_1, \boldsymbol{f}_2 - \boldsymbol{f}_1, \ldots, \boldsymbol{f}_{n_f} - \boldsymbol{f}_1\} \end{aligned} \tag{7.7}$$

$$\begin{aligned} &\mathsf{F}[\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, 0\theta\}] = R^{(0\theta)} T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})}\mathsf{F} \\ =& \{R^{(0\theta)} T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})} \boldsymbol{f}_1, R^{(0\theta)} T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})} \boldsymbol{f}_2, \ldots, R^{(0\theta)} T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})} \boldsymbol{f}_{n_f}\} \end{aligned} \tag{7.8}$$

At another layer, say layer 1, expression (7.8) would become $\mathsf{F}[\{f_{1_\mathsf{x}}, f_{1_\mathsf{y}}, 0\theta\}] = R^{(1\theta)} T^{(-f_{1_\mathsf{x}}, -f_{1_\mathsf{y}})}\mathsf{F}$.

We can also express $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ according to specific fingers. This procedure has been discussed when we were building the helical pattern in Fig.**??** and Fig.**??**. The first component of $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ involves the configurations (namely voxels) that fulfills the following two conditions. (1) The formation rotate around finger $\boldsymbol{f}_1$. (2) When the formation is at those configurations, $\boldsymbol{f}_1$ in inside the target object, namely it is inside V. Here the first condition ensures that this component is correspondent with $\boldsymbol{f}_1$ while the second condition ensures that the configurations belong to $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$. This first component can therefore be expressed as following.

$$\bigcup_{i=-m}^{m} \mathsf{V} \tag{7.9}$$

The other components which correspond to the other fingers besides $\boldsymbol{f}_1$ are a bit complicated. Take a finger $\boldsymbol{f}_j$ for example. Like $\boldsymbol{f}_1$, its correspondent component of $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ involves the configurations that fulfills the following two conditions. (1) The formation rotate around finger $\boldsymbol{f}_j$. (2) When the formation is at those configurations, $\boldsymbol{f}_j$ in inside the target object, namely it is inside V. The second condition, according to Fig.**??**, can be converted to the rotation around $\boldsymbol{f}_j$ plus the translation $\boldsymbol{f}_1 - \boldsymbol{f}_j$ (It was $\boldsymbol{f}_1 - \boldsymbol{f}_j + \boldsymbol{f}_1$ in Fig.**??** because we only consider the grid that was occupied by $\boldsymbol{f}_1$.). Therefore, the component that correspond to a finger $\boldsymbol{f}_j$ can be expressed by the following expression.

$$\bigcup_{i=-m}^{m} \begin{bmatrix} 1 & 0 & \\ 0 & 1 & R^{(-i\theta)} \\ 0 & 0 & \end{bmatrix} \begin{bmatrix} f_{1_\mathsf{x}} - f_{j_\mathsf{x}} \\ f_{1_\mathsf{y}} - f_{j_\mathsf{y}} \\ 1 \end{bmatrix} \mathsf{V},\ j = 2, 3, \ldots, n_f \tag{7.10}$$

Consequently, $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ is the set of voxels enclosed by expression (7.9) and (7.10).

7.1.3 The relationship between $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$ and $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$

Now let us compare the relationship between $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$ and $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ and make clear their relationship.

Consider the expression between a layer in $\mathcal{F}_i^{\mathsf{obj}}$ and a layer in the first component of $\mathcal{F}^{\mathsf{frm}}$,

$$\bigcup_{i=-m}^{m} T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})} \mathsf{V} \quad \text{and} \quad \bigcup_{i=-m}^{m} \mathsf{V} \tag{7.11}$$

They essentially differ with a linear transformation $T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}$. If the expressions

$$\begin{gathered} \bigcup_{i=-m}^{m} T^{(f_{j\mathsf{x}},f_{j\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})} \mathsf{V},\ j = 2,3,\ldots,m \\ \text{and} \\ \bigcup_{i=-m}^{m} \begin{bmatrix} 1 & 0 & \\ 0 & 1 & R^{(-i\theta)} \\ 0 & 0 & \end{bmatrix} \begin{bmatrix} f_{1\mathsf{x}} - f_{j\mathsf{x}} \\ f_{1\mathsf{y}} - f_{j\mathsf{y}} \\ 1 \end{bmatrix} \mathsf{V},\ j = 2,3,\ldots,n_f \end{gathered} \tag{7.12}$$

share the same difference, we can draw a conclusion that $\mathcal{C}_{\mathsf{otl}}^{\mathsf{obj}}$ and $\mathcal{C}_{\mathsf{otl}}^{\mathsf{frm}}$ can be converted in layer level by a linear transformation $T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}$.

By left-multiplying $\begin{bmatrix} 1 & 0 & \\ 0 & 1 & R^{(-i\theta)} \\ 0 & 0 & \end{bmatrix} \begin{bmatrix} f_{1\mathsf{x}} - f_{j\mathsf{x}} \\ f_{1\mathsf{y}} - f_{j\mathsf{y}} \\ 1 \end{bmatrix} \mathsf{V}$ with the difference $T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}$, we can get

$$\begin{aligned} & T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})} \begin{bmatrix} 1 & 0 & \\ 0 & 1 & R^{(-i\theta)} \\ 0 & 0 & \end{bmatrix} \begin{bmatrix} f_{1\mathsf{x}} - f_{j\mathsf{x}} \\ f_{1\mathsf{y}} - f_{j\mathsf{y}} \\ 1 \end{bmatrix} \mathsf{V} \\ =& T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} \begin{bmatrix} 1 & 0 & \\ 0 & 1 & R^{(-i\theta)} \\ 0 & 0 & \end{bmatrix} \begin{bmatrix} f_{1\mathsf{x}} - f_{j\mathsf{x}} \\ f_{1\mathsf{y}} - f_{j\mathsf{y}} \\ 1 \end{bmatrix} T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})} \mathsf{V} \end{aligned} \tag{7.13}$$

since $\begin{bmatrix} 1 & 0 & \\ 0 & 1 & R^{(-i\theta)} \\ 0 & 0 & \end{bmatrix} \begin{bmatrix} f_{1\mathsf{x}} - f_{j\mathsf{x}} \\ f_{1\mathsf{y}} - f_{j\mathsf{y}} \\ 1 \end{bmatrix}$ is a translation matrix in the form of $T^(\delta x, \delta y)$ and it is inter-changeable with $T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}$.

Considering that $\forall\phi,\ \forall\delta x$ and $\forall\delta y$

$$R^{\phi} T^{(\delta x,\delta y)} = \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} R^{\phi} \begin{bmatrix} \delta x \\ \delta y \\ 1 \end{bmatrix} \end{bmatrix} R^{\phi} \tag{7.14}$$

expression (7.13) becomes

$$
\begin{aligned}
&T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}R^{\pi}R^{(i\theta)}\begin{bmatrix}1&0&\\0&1&R^{(-i\theta)}\\0&0&\end{bmatrix}\begin{bmatrix}f_{1_\mathsf{x}}-f_{j_\mathsf{x}}\\f_{1_\mathsf{y}}-f_{j_\mathsf{y}}\\1\end{bmatrix}\Bigg]T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}\\
=&T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}R^{(\pi+i\theta)}\begin{bmatrix}1&0&\\0&1&R^{(-i\theta)}\\0&0&\end{bmatrix}\begin{bmatrix}f_{1_\mathsf{x}}-f_{j_\mathsf{x}}\\f_{1_\mathsf{y}}-f_{j_\mathsf{y}}\\1\end{bmatrix}\Bigg]T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}\\
=&T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}\begin{bmatrix}1&0&\\0&1&R^{(\pi+i\theta)}R^{(-i\theta)}\\0&0&\end{bmatrix}\begin{bmatrix}f_{1_\mathsf{x}}-f_{j_\mathsf{x}}\\f_{1_\mathsf{y}}-f_{j_\mathsf{y}}\\1\end{bmatrix}\Bigg]R^{(\pi+i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}\\
=&T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}\begin{bmatrix}1&1&\\0&0&T^{(f_{j\mathsf{x}}-f_{1\mathsf{x}},f_{j\mathsf{y}}-f_{1\mathsf{y}})}\\0&0&\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}\Bigg]R^{(\pi+i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}
\end{aligned}
\tag{7.15}
$$

Since the first and second column of the translation matrix $T^(\delta x, \delta y)$ are $[1\ 0\ 0]'$ and $[0\ 1\ 0]'$ respectively,

$$
\begin{aligned}
&T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}\begin{bmatrix}1&1&\\0&0&T^{(f_{j\mathsf{x}}-f_{1\mathsf{x}},f_{j\mathsf{y}}-f_{1\mathsf{y}})}\\0&0&\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}\Bigg]R^{(\pi+i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}\\
=&T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}\left[T^{(f_{j\mathsf{x}}-f_{1\mathsf{x}},f_{j\mathsf{y}}-f_{1\mathsf{y}})}\begin{bmatrix}1\\0\\0\end{bmatrix}T^{(f_{j\mathsf{x}}-f_{1\mathsf{x}},f_{j\mathsf{y}}-f_{1\mathsf{y}})}\begin{bmatrix}1\\0\\0\end{bmatrix}T^{(f_{j\mathsf{x}}-f_{1\mathsf{x}},f_{j\mathsf{y}}-f_{1\mathsf{y}})}\begin{bmatrix}0\\0\\1\end{bmatrix}\right]\\
&R^{(\pi+i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}\\
=&T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}T^{(f_{j\mathsf{x}}-f_{1\mathsf{x}},f_{j\mathsf{y}}-f_{1\mathsf{y}})}R^{(\pi+i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}\\
=&T^{(f_{j\mathsf{x}},f_{j\mathsf{y}})}R^{(\pi+i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}\\
=&T^{(f_{j\mathsf{x}},f_{j\mathsf{y}})}R^{\pi}R^{(i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}\mathsf{V}
\end{aligned}
\tag{7.16}
$$

The result is exactly the same as expression (7.6) so that we are confirmed that at each layer $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$ and $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ differ with a linear transformation $T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}R^{\pi}R^{(i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}$. Fig.7.1 graphically illustrates the relationship between layers of the two spaces. The dashed polygon shape (one slice of $\mathcal{C}^{frm}$) in the right part of Fig.7.1 can be converted to the dashed polygon shape (one slice of $\mathcal{C}^{obj}$) in the left part of Fig.7.1 by multiplying a linear transformation matrix $T^{(f_{1\mathsf{x}},f_{1\mathsf{y}})}R^{\pi}R^{(i\theta)}T^{(-v_{1\mathsf{x}},-v_{1\mathsf{y}})}$.

This conclusion indicates that the difference of the two spaces are their metrics which are essential to caging robustness. In $\mathcal{C}^{\mathsf{obj}}$, our metrics aim to measure the distance from caging sub-space to free sub-space (see the left part of Fig.7.2). However, due to high computational cost, the metrics were difficult to be employed explicitly so that I decomposed the constraints into translational caging together with a rotational component for approximation. In $\mathcal{C}^{\mathsf{frm}}$, the metrics become clear and we combine the distances to critical surfaces as the quality

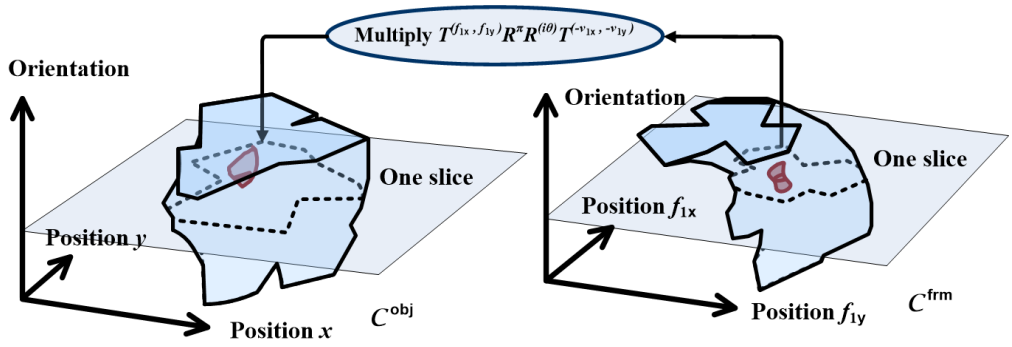


Figure 7.1: The slices in $\mathcal{C}^{frm}$ and $\mathcal{C}^{obj}$ can be converted to each other by a linear transformation.

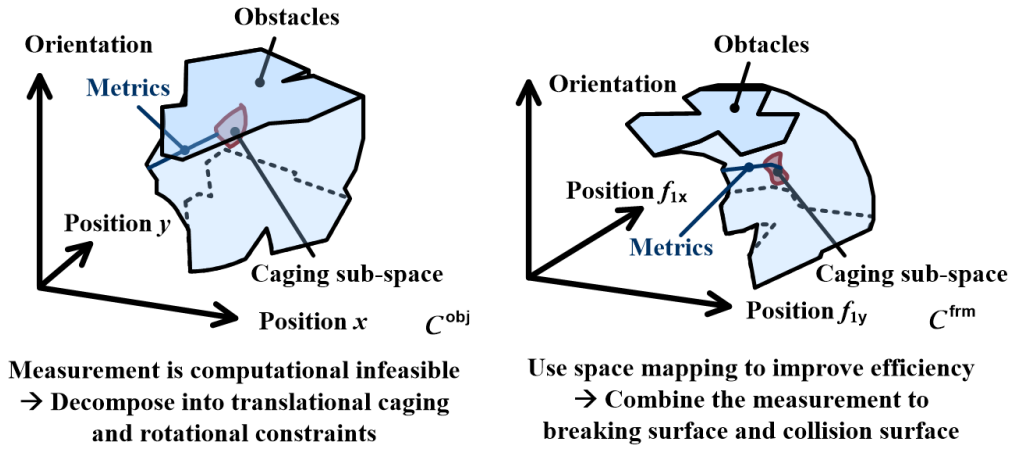


Figure 7.2: The metrics of the two spaces are essentially different.

function (see the right part Fig.7.2). If we analyze the relationship between the metrics in these two spaces in detail, it could be in the following form. Note that there is no transformation along rotational axes, namely the transformation between $\mathcal{C}^{\text{frm}}$ and $\mathcal{C}^{\text{obj}}$ are independent of the rotation. Take a 3-D vector $\overrightarrow{u_{ij}} = \begin{bmatrix} u_{i_x} - u_{j_x} \\ u_{i_y} - u_{j_y} \\ \theta_i - \theta_j \end{bmatrix}$ in $\mathcal{C}^{\text{frm}}$ for example. A metric with weighting matrix W in $\mathcal{C}^{\text{frm}}$ can be expressed by

$$\overrightarrow{u_{ij}}' W \overrightarrow{u_{ij}} \tag{7.17}$$

In $\mathcal{C}^{\text{obj}}$, it becomes the following expression.

$$\begin{array}{c} \left[\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \theta_i - \theta_j \end{array} \right] T^{(f_{1\mathsf{x}}, f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}}, -v_{1\mathsf{y}})} \left[\begin{array}{c} u_{i\mathsf{x}} - u_{j\mathsf{x}} \\ u_{i\mathsf{y}} - u_{j\mathsf{y}} \\ 1 \end{array} \right] \right]' \\ W \left[\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \theta_i - \theta_j \end{array} \right] T^{(f_{1\mathsf{x}}, f_{1\mathsf{y}})} R^{\pi} R^{(i\theta)} T^{(-v_{1\mathsf{x}}, -v_{1\mathsf{y}})} \left[\begin{array}{c} u_{i\mathsf{x}} - u_{j\mathsf{x}} \\ u_{i\mathsf{y}} - u_{j\mathsf{y}} \\ 1 \end{array} \right] \right] \end{array} \tag{7.18}$$

With the same weighting matrix W, the metrics in these two spaces are quite different from each other. **The metrics heavily relate to the quality functions and the robustness of caging. From this viewpoint, $\mathcal{C}^{\mathsf{frm}}$, comparing with $\mathcal{C}^{\mathsf{obj}}$, offers a more intuitive way to choose satisfying metrics.**

7.2 Choosing Proper Algorithms

Now we have explored the caging algorithms to solve the caging test problem and the caging optimization problem in two spaces. One is the configuration space of target object, namely $\mathcal{C}^{\mathsf{obj}}$. The other one is the configuration space of finger formation, namely $\mathcal{C}^{\mathsf{obj}}$. Both the two spaces have some advantages and disadvantages. They can be summarized as Fig.7.3.

Figure 7.3: Comparing the advantages and disadvantages of the algorithms in $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$.

There are two limitations to the algorithms in $\mathcal{C}^{\mathsf{obj}}$. Firstly, they are limited to convex objects. Then, some of their parameters like combination of Q^{T} and Q^{R} and the retraction of fingers are chosen empirically. However, the algorithms in $\mathcal{C}^{\mathsf{obj}}$ are suitable to be used in fully distributed applications like our distributed end-effector or the multi-robot co-operative transportation. That is because in these applications, each finger or mobile robot can be distributedly actuated to any position and there is no apparent eigen-shapes. Their potential finger formations or mobile robot formations are infinite. It is difficult to recover and update every $\mathcal{C}^{\mathsf{frm}}$ of those infinite formations. $\mathcal{C}^{\mathsf{obj}}$ is therefore more suitable than $\mathcal{C}^{\mathsf{frm}}$.

Comparing with $\mathcal{C}^{\mathsf{obj}}$, $\mathcal{C}^{\mathsf{frm}}$ is not limited to convex objects. It can be applied to arbitrary 2D objects, including either convex, concave objects or even objects with hollow holes. Moreover, we do not need the empirical parameters like $\mathcal{C}^{\mathsf{obj}}$. The algorithms in $\mathcal{C}^{\mathsf{frm}}$ is complete to discretization resolution. That is to say, the algorithms can always solve the caging problems as long as the resolution of grids is high enough. In contrast, the algorithms in $\mathcal{C}^{\mathsf{obj}}$ is not complete since it may fail and reject to cage certain objects due to empirical parameter settings. However, the algorithms in $\mathcal{C}^{\mathsf{frm}}$ requires us to maintain, say recover and update, a $\mathcal{C}^{\mathsf{frm}}$ for each finger formation. That means the number of formations is limited. Although the maintenance time of one $\mathcal{C}^{\mathsf{frm}}$ has been improved greatly by shifting pre-built mapping structures, the algorithms in $\mathcal{C}^{\mathsf{frm}}$ are most suitable to robotic hands with eigen-shapes or

robotic hands that can be represented by some pre-defined formations. It cannot cover as many formations as the algorithms in $\mathcal{C}^{\mathsf{obj}}$.

I proved in the last section that when the orientation is fixed, $\mathcal{C}^{\mathsf{obj}}_{\mathsf{otl}}$ and $\mathcal{C}^{\mathsf{frm}}_{\mathsf{otl}}$ can be converted to each other by a linear transformation. Namely, using $\mathcal{C}^{\mathsf{frm}}$ instead of $\mathcal{C}^{\mathsf{obj}}$ is essentially using different metrics. The metrics heavily relate to the quality functions and the robustness of caging. From that viewpoint, $\mathcal{C}^{\mathsf{frm}}$, comparing with $\mathcal{C}^{\mathsf{obj}}$, offers a more intuitive way to choose satisfying metrics.

At this point, we may have a question like this. How can we choose between them or can we combine them, preserving the merits and making up the weakness? I maintain that one may not smoothly combine the algorithms in these two spaces since they are essentially different in metrics. However, both $\mathcal{C}^{\mathsf{frm}}$ and $\mathcal{C}^{\mathsf{obj}}$ have disadvantages and advantages. I recommend using $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$ separately according to mechanical structure of robots and tasks. If all capture points are distributed and target objects are convex, I recommend perform caging planning with the algorithms in $\mathcal{C}^{\mathsf{obj}}$. If capture points are kinematically constrained or target objects have various shapes, I recommend performing caging planning with the algorithms in $\mathcal{C}^{\mathsf{frm}}$. Fig.7.4 shows this idea with concrete illustration.

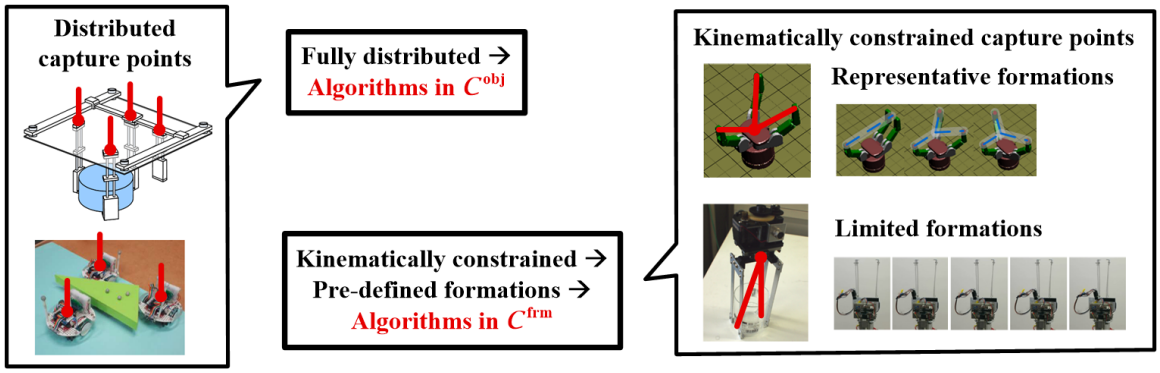


Figure 7.4: Choosing the proper algorithms according to real applications.

Chapter 8

Conclusions and Future Works

8.1 Summary of the Contents

This thesis proposes caging planning algorithms by using the configuration space of target object ($\mathcal{C}^{\mathsf{obj}}$) and the configuration space of finger formation ($\mathcal{C}^{\mathsf{frm}}$). It concludes that caging is the loose form of grasping so that it can be used to deal with uncertainty caused by perception and control. The effectiveness of the algorithms are demonstrated with simulation and real-world platforms.

The thesis is composed of eight chapters. Besides this 8th chapter, the other 7 chapters are as following.

Chapter 1 introduces the basic concepts of robotic manipulation and the basic structure of an end-effector collaboratively developed by me and some students in my group. Along with the introduction, this chapter explains how I started this research in caging and why I am confident that caging is promising. This chapter proposes that caging can be used to deal with uncertainty and can be a good supporting tool for the end-effector. It further defines the caging test and caging optimization problems and gives an overview of thesis organization. Specifically, caging test examines whether caging is attained or not while caging optimization finds a robust finger formation against uncertainty.

Chapter 2 reviews the traditional research topics in grasping closure. It makes clear that caging is the general form of grasping by connecting caging to such traditional grasping research topics as force closure, form closure and immobilization based on an intensive review of contemporary works in grasping closure. In detail, Fig.2.14 visualizes the shift between caging and traditional concepts in grasping in $\mathcal{C}^{\mathsf{obj}}$.

Chapters 3$\sim$6 propose several algorithms to the caging problems in two different configuration spaces $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$ (Chapter 3 and 5), and apply the algorithms to real robotic platforms (Chapter 4 and 6). Specifically, Chapter 3 proposes the caging algorithms by evaluating the regions in which fingers guarantee caging (caging region) based on the re-implementation of a previous work which aims to find the caging region of a third finger given a 2D convex target object and two boundary contacts. Then, it explores how to

push though the limitations of given fingers and extend the algorithms to this proposal to general cases. The solution is fixing the two boundary contacts alternatively and reducing the computational complexity of that algorithm by decomposing caging into translational constraints and rotational constraints. The proposed algorithms solve the caging problems of 2D convex objects. Their ability to endure uncertainty is demonstrated with simulation and real-world platforms like the distributed end-effector and multi-robot cooperative transportation in Chapter 4. Chapter 5 explores algorithms in $\mathcal{C}^{\mathsf{frm}}$ instead of $\mathcal{C}^{\mathsf{obj}}$. Since the center of $\mathcal{C}^{\mathsf{obj}}$ is the target object, it is inherently affected by object shapes. Unlike $\mathcal{C}^{\mathsf{obj}}$, $\mathcal{C}^{\mathsf{frm}}$ is the configuration space of finger formation and is free from the affection of object shapes. It therefore can be more flexible to various objects, including convex, concave and even hollow ones. In detail, this chapter firstly compares $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$. It discusses both the advantages of $\mathcal{C}^{\mathsf{frm}}$ and the difficulties of implementing caging algorithms in $\mathcal{C}^{\mathsf{frm}}$ and proposes to overcome the difficulties by introducing the space mapping idea. Raw space mapping and especially its faster version, the improved space mapping, make it possible to update the whole space of $\mathcal{C}^{\mathsf{frm}}$ completely and efficiently. Based on the updated $\mathcal{C}^{\mathsf{frm}}$, the algorithm can quickly find the caging candidates in the updated $\mathcal{C}^{\mathsf{frm}}$, locate the optimal caging configuration and solve the caging problems. The caging algorithms in $\mathcal{C}^{\mathsf{frm}}$ are applied to the design and implementation of a gripping hand in Chapter 6. The algorithms play role in both design and implementation procedures. During design, the algorithms are employed to simplify and evaluate deign models. During implementation, the algorithms are employed to control the hand to cage and grasp objects. The design and implementation demonstrate advantages of the algorithms.

Chapter 7 explains the relationships of the algorithms and discusses how to select them. It proves that at different orientations $\mathcal{C}^{\mathsf{frm}}$ is the linear transformation of $\mathcal{C}^{\mathsf{obj}}$. Consequently, the metrics used in $\mathcal{C}^{\mathsf{frm}}$ and $\mathcal{C}^{\mathsf{obj}}$ are different. Both the two tools and their correspondent algorithms have reasons to exist. They therefore should be treated in parallel and selected according to requirements of specific applications. The algorithms in $\mathcal{C}^{\mathsf{obj}}$ are suitable to fully distributed capture points but they are limited by object shapes since they are inherently affected by target objects. In contrast, the algorithms in $\mathcal{C}^{\mathsf{frm}}$ are suitable to any 2D object shape but they are subject to limited number of finger formations since they are inherently affected by formations.

8.2 Contributions

The contributions of the thesis can be summarized in Fig.8.1.

As is shown in Fig.8.1, this thesis contributes in theoretical, algorithmic and application aspects. In theoretical contributions, it initially explains the relationship between caging and traditional research in grasping closure. Namely, caging is the extension of immobilization. In algorithmic contributions, it initially uses caging to deal with uncertainty. The thesis on the one hand proposes some rapid algorithms to deal with caging test while on the other hand proposes the caging optimization problem and a series of solutions in both $\mathcal{C}^{\mathsf{obj}}$ and

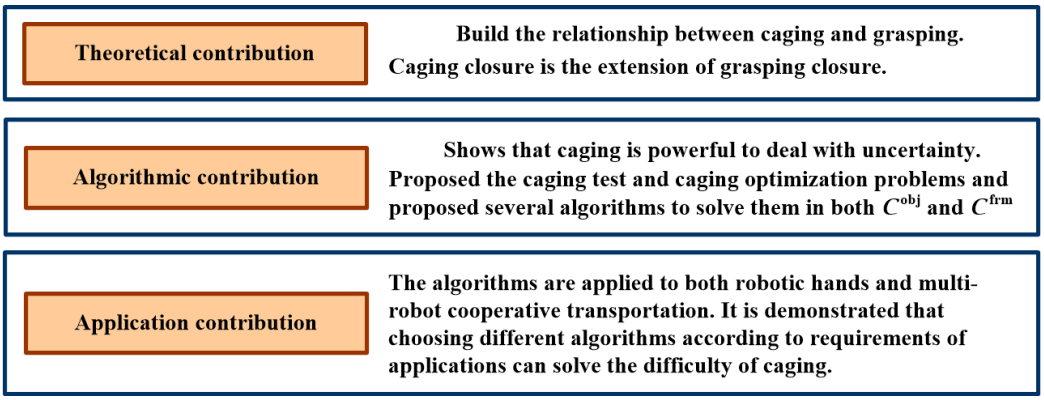


Figure 8.1: Contributions of the thesis.

$\mathcal{C}^{\mathsf{frm}}$. In application contributions, it applies the algorithms to robotic hands and multi-robot cooperative transportation. The applications based on the caging algorithms can not only robustly deal with various uncertainty but also help choose least or proper number of fingers or mobile robots.

The algorithms in $\mathcal{C}^{\mathsf{obj}}$ are limited to convex objects. However, they are suitable to be used in fully distributed applications like our distributed end-effector or the multi-robot co-operative transportation. That is because in these applications, each finger or mobile robot can be distributedly actuated to any position and there is no apparent eigen-shapes. Their potential finger formations or mobile robot formations are infinite. It is difficult to recover and update every $\mathcal{C}^{\mathsf{frm}}$ of those infinite formations.

Comparing with $\mathcal{C}^{\mathsf{obj}}$, $\mathcal{C}^{\mathsf{frm}}$ is not limited to convex objects. It can be applied to convex, concave objects and even objects with hollow holes. However, we have to maintain, say recover and update, a $\mathcal{C}^{\mathsf{frm}}$ for each finger formation. That means the number of formations is limited. Although the maintenance time of one $\mathcal{C}^{\mathsf{frm}}$ has been improved greatly by shifting pre-built mapping structures, it is most suitable to robotic hands with eigen-shapes and it cannot cover as many formations as the algorithms in $\mathcal{C}^{\mathsf{obj}}$.

Both the algorithms in $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$ have their advantages and disadvantages. They actually correspond to different solutions of **geometric modeling**. The algorithm in $\mathcal{C}^{\mathsf{obj}}$ uses **wireframe modeling** while the algorithm in $\mathcal{C}^{\mathsf{frm}}$ uses **solid modeling**. Both modeling technology plays important roles in **geometric modeling** and either algorithms in $\mathcal{C}^{\mathsf{obj}}$ and $\mathcal{C}^{\mathsf{frm}}$ should exist. I demonstrate their relationship and explain how to choose properly between them in Chapter 7. Nevertheless, reader may have found that we can also use the solid modeling technology of $\mathcal{C}^{\mathsf{frm}}$ in $\mathcal{C}^{\mathsf{obj}}$. That is possible. However, that would increase the cost of evaluating one formation in $\mathcal{C}^{\mathsf{obj}}$, making it lose its advantages. The solid modeling technology is working efficiently in $\mathcal{C}^{\mathsf{frm}}$ because the formations and grids imply an unique mapping structure. Simple shifting and addition computation could help recover and update $\mathcal{C}^{\mathsf{frm}}$. The center of $\mathcal{C}^{\mathsf{obj}}$ is the target object and there is no underlying unique mapping structure.

The concepts and algorithms proposed in this thesis can efficiently solve 2D caging problems in the presence of uncertainty. I believe that caging is a promising tool to deal with perception and control uncertainty and would like to explore more about this tool in both theory and application aspects in the future. Some potential future directions can be found in the following texts.

8.3 Further Development of Caging Algorithms and Prospective Application Fields

The future work could involve two aspects. One is further development of the caging algorithms. The other is using it to challenge the difficulties caused by uncertainties in robotics.

8.3.1 Further development of caging algorithms

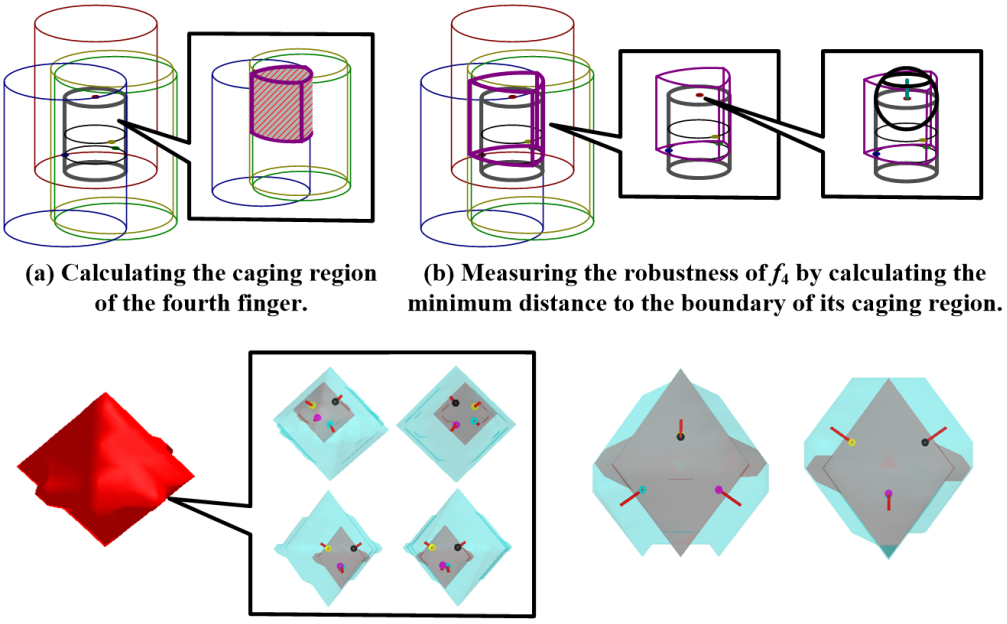
I have repeated lots of times that neither the algorithms in $\mathcal{C}^{\mathsf{frm}}$ nor the algorithms $\mathcal{C}^{\mathsf{obj}}$ are perfect. If one can develop an algorithm that can quickly find an optimized caging with no limitation in the number of finger formations and the shape of target objects, that would be an impressive contribution to caging. However, I wonder whether that impressive algorithm exists with current computers.

Actually, like many researchers in robotics, I believe in data warehouse and cloud computing rather than exact and perfect algorithms. The intelligence of machines should not be limited to on-board computing, but rely to (1) grid computing and (2) large-scale database. Of course, the robots based on data warehouse and cloud computing are limited to those supporting resources. Nevertheless, I believe this solution would succeed. Resources on the cloud shall one day make robots more intelligent than human beings and of course, solves caging problems. In the future, giant corporations would run their own supporting data warehouse and cloud computing resources and offer service to their terminal robots. Robots would be no more than thin hardware clients. Actually, we can already see some publications that are working into this direction. In this year's top conference on robotics, namely ICRA2013, Ben from Prof. Goldberg's group at the University of California, Berkeley published a robot grasping work by using google object recognition engine [Kehoe et al., 2013]. It is motivating and I will continue tracking their publications to see their ensuing steps.

Another future development of the algorithms is how to deal with 2.5D objects. In this thesis, all discussion concentrates on 2D objects. We did not discuss about 2.5D or 3D objects. Actually, the algorithms in $\mathcal{C}^{\mathsf{obj}}$ can be extended to 3D objects intuitively. Fig.8.2 illustrates the extension.

Fig.8.2(a) shows the extension of translational caging on 3D object with a simple cylinder object. I use this simple cylinder because it is more readable comparing with other complicated objects. Given three fingers f_1, f_2 and f_3, the translational caging region of f_4 could be rendered as the intersection of the $\mathcal{F}_{\mathcal{F}}$ of f_1, f_2 and f_3. This is exactly the same as the case of calculating the translational caging region of a third finger with 2D objects. Readers

may compare the shadowed region in Fig.8.2(a) and Fig.?? to understand their similarity. After calculating the caging region of f_4, we can measure the robustness of f_4 by referring to its distance to the boundary of its caging region. The distance of the cylinder object is shown in Fig.8.2(b). Then, like Fig.??, we can evaluate the robustness of a finger formation by alternatively fixing adjacent fingers and by recording the smallest robustness of a single finger. The remaining steps are exactly the same as 2D cases shown in Fig.??. Namely adding rotational constraints and retract fingers. This extension, of course, is still limited to convex shapes. I implemented the extension with a regular octahedron. Fig.8.2(c) shows the result of my implementation. Since regular octahedron is regular, the optimized fingers locate exactly at the center of each surface. Readers may refer the points and segments in Fig.8.2(c) for details. Here the points denote point fingers while the segments denotes surface normals.



(a) Calculating the caging region of the fourth finger.

(b) Measuring the robustness of f_4 by calculating the minimum distance to the boundary of its caging region.

(c) Results of the extension on a 3D regular octahedron.

Figure 8.2: The algorithms in $\mathcal{C}^{\text{obj}}$ can be extended to 3D objects intuitively.

This extension and implementation were neither further explored nor included in the major contents of this thesis because I think caging 3D objects with point fingers is impractical. General robotic hands, even if they are dexterous, can hardly fulfill the kinematic requirements of an optimized caging formations. However, discussing about 2.5D objects deserves efforts. Dealing with 2.5D objects means not only considering about top-view 2D shapes, but also consider the convexity and concavity of side views. Considering side views requires installing local sensors onto capture points. Some students in our group are working on that aspect. They installed infrared sensor arrays to the inner finger surfaces of the distributed end-effector and measure the shape of target object side views. Considering about side views

and dealing with 2.5D objects would not only stop target objects from escaping the finger formation but also stop target objects from "falling out of" the finger formation. Developing those algorithms would be a promising direction.

8.3.2 Perspective application fields

The second aspect of the future work is to improve the robotic applications which are hindered by the problems caused by uncertainties. An optimized caging offers robustness to both caging breaking and collision with target objects, and consequently offers robustness to uncertainties caused by perception and control. I believe that using the caging test and caging optimization algorithms developed in this thesis would greatly improve those hindered applications.

One example, which came into my view when I was exploring the problems caused by uncertainty, is the filed of micro/nano manipulation. When the scale is between $1\mu m$-$100\mu m$, people will name it micro-manipulation. When the scale is lower than $1\mu m$, people name it nano-manipulation. The manipulation in micro/nano world suffers from lots of uncertainties. On the one hand, perception in the micro/nano world is quite difficult. There is no existing method to simultaneously image and manipulate atoms. On the other hand, surface forces (e.g. electrostatic force, surface tension, van der Waals, casimir, etc) in the micro/nano world play dominating roles over gravity or inertia forces. It is difficult to model these forces mathematically. They are uncertain and deteriorate traditional control. Therefore, pure geometric solution like caging would probably play a promising role.

Like the macro-world applications in this thesis, some of the micro/nano manipulation systems are fully distributed while others are not. For example, manipulating objects with optical tweezers or micro-robots usually fall in the first category while manipulating objects with probe-tips or micro-grippers usually fall into the second category. Here are some works that relate to these categories. For instance, the bubble robots in [Hu et al., 2011] are examples of micro-robots. David's work [Cappelleri et al., 2012] and some earlier work of our group [Sato, 1996] are examples of probe-tips. The electro-thermally activated cell manipulator [Chronis and Lee, 2005] is an example of micro-manipulation. In the following part, I would like to discuss in detail about optical tweezers.

Optical tweezers [Grier, 2003][Onda and Arai, 2011] use two highly focused laser beams to trap very small crystal beads. That is possible owning to the attractive and repulsive forces caused by optic photons. Each bead can be viewed as a distributed capture point and researchers use many beads controlled by optical tweezers to co-operatively manipulate target objects in the micro-world. Like the multi-robot co-operative application in chapter 4, the optical tweezer-based secondary micro-manipulation suffers from uncertainty caused by formation control. Moreover, since controlling multiple beams requires more laser beams or more complicated control of spatial real-time modulator to modulate a single beam, researchers prefer less number of beads. Therefore, I strongly believe caging algorithms would play important roles in this field.

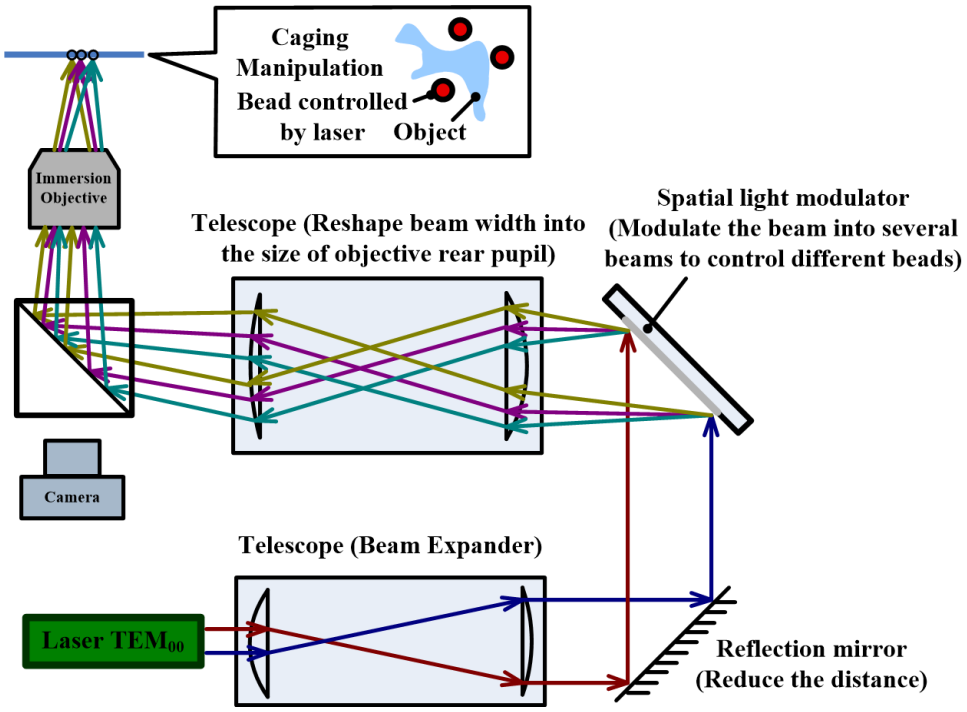


Figure 8.3: A typical micro-manipulation system (optical tweezers).

Fig.8.3 illustrates a typical micro-manipulation system. When spatial real-time modulators are employed, the number of formations of beads is usually limited to pre-programmed modulating patterns. In that case, the caging algorithms in $\mathcal{C}^{\mathsf{frm}}$ is preferable. When spatial real-time modulators are not employed, independent beads are distributedly controlled by independent laser beams. In that case, the caging algorithms in $\mathcal{C}^{\mathsf{obj}}$ is preferable. I am going to organize a workshop on "caging and its application in grasping/multi-agent cooperation" in November, 2013 during the International Conference on Intelligent Robots and Systems and discuss the possibility of these ideas with some researchers in micro/nano manipulation fields.

Another example is using caging to deal with in-hand manipulation. In-hand manipulation means to hold and move an object with one hand. It requires that the object should be constrained into the hand while being manipulated. The handle project [Handle Project, 2013] gives an excellent collection of state-of-the-art researches in this field.

However, state-of-the-art researches in in-hand manipulation aims at the manipulation of dexterous robotic hands which offer the possibility of rigid analysis of forces and force closures. In another word, the robots hold and move objects in hand by using forces. Using forces lead to the major difficulty of in-hand manipulation. That is, the state-of-the-art in-hand manipulation is hindered by the uncertainty from perception and control. Caging can be used to deal with uncertainty. Therefore, we have a perspective future research direction that can we do in-hand manipulation without forces by caging? If that is possible, we can

not only get merits from avoiding explicit force and control analysis but also do in-hand manipulation with simple hands [Mason et al., 2012] and reduce platform costs. Caging-based in-hand manipulation would greatly decrease the cost of deployment and real-world applications. This is a passionating topic and I am going to study further into this aspect in my postdoctoral research.

Caging is not only from the idea of bird cage but also from lots of nature creative (see the Venus flytrap plant[Wikipedia, 2013b][1]). I like caging, I believe in caging and I would like to spread caging to related research and applications. As the last sentence of my thesis, I would like to re-emphasize the merits of caging – Caging offers robustness to uncertainty.

[1]Plant suffers from perception/control uncertainty comparing with animals, they employ caging to complement the drawbacks caused by the uncertainty.

Bibliography

[Allen et al., 2012] Allen, T., Burdick, J., and Rimon, E. (2012). Two-fingered Caging of Polygons Via Contact-Space Graph Search. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4183–4189.

[Barrett Technology, 2013] Barrett Technology (Last updated: 2013). The Barrett Hand. Last accessed: 2013. http://web.barrett.com/support/\{B}arrett{H}and_{D}ocumentation/{BH}8-280_{D}atasheet.pdf.

[Berenson, 2011] Berenson, D. (2011). *Constrained Manipulation Planning*. PhD thesis, Carnegie Mellon University.

[Berenson et al., 2007] Berenson, D., Diankov, R., Nishiwaki, K., Kagami, S., and Kuffner, J. (2007). Grasp Planning in Complex Scenes. In *Proceedings of IEEE International Conference on Humanoid Robots*.

[Berenson and Srinivasa, 2008] Berenson, D. and Srinivasa, S. (2008). Grasp Synthesis in Cluttered Environments for Dexterous Hands. In *Proceedings of IEEE International Conference on Humanoid Robots*.

[Berenson et al., 2009a] Berenson, D., Srinivasa, S., Ferguson, D., Collet Romea, A., and Kuffner, J. (2009a). Manipulation Planning with Workspace Goal Regions. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1397–1403.

[Berenson et al., 2009b] Berenson, D., Srinivasa, S., and Kuffner, J. (2009b). Addressing Pose Uncertainty in Manipulation Planning Using Task Space Regions. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1419–1425.

[Berenson et al., 2011] Berenson, D., Srinivasa, S., and Kuffner, J. (2011). Task Space Regions: A Framework for Pose-Constrained Manipulation Planning. *The International Journal of Robotics Research*.

[Bicchi, 1995] Bicchi, A. (1995). On the Closure Properties of Grasping. *The International Journal of Robotics Research*, 14:319–334.

[Blender, 2013] Blender (Last updated: 2013). Blender. Last accessed: 2013. http://www.blender.org/.

Blender is a open source 3D modeling software which offer a python interface and api for programming and visualization.

[Bohg et al., 2011] Bohg, J., Johnson-Roberson, M., Leon, B., Felip, J., Gratal, X., Bergstrom, N., Kragic, D., and Morales, A. (2011). Mind the Gap – Robotic Grasping under Incomplete Observation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 686–693.

[Bohlin and Kavraki, 2000] Bohlin, R. and Kavraki, L. E. (2000). Path Planning Using Lazy PRM. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 521–528.

[Bridgwater et al., 2012] Bridgwater, L. B., Ihrke, C., Diftler, M. A., Abdallah, M. E., Radford, N. A., Rogers, J. M., Yayathi, S., Askew, R. S., and Linn, D. M. (2012). The Robonaut 2 Hand Designed to Do Work with Tools. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3425–3430.

[Cappelleri et al., 2011] Cappelleri, D. J., Fatovic, M., and Shah, U. (2011). Caging Micromanipulation for Automated Microassembly. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3145–3150.

[Cappelleri et al., 2012] Cappelleri, D. J., Fu, Z., and Fatovic, M. (2012). Caging for 2D and 3D Micromanipulation. *Journal of Micro-Nano Mechatronics*, 7:115–129.

[Cheng et al., 2008] Cheng, P., Fink, J., and Kumar, V. (2008). Abstractions and Algorithms for Cooperative Multiple Robot Planar Manipulation. In *Proceedings of Robotics, Science and Systems*, pages 143–150.
This is a typical work of multi-robot cooperative manipulation by using task allocation. A video accompanying this work can be found at http://www.youtube.com/watch?v=fNBqys2Bqhs.

[Cheong et al., 2006] Cheong, J.-S., Haverkort, H. J., and van der Stappen, A. F. (2006). Computing All Immobilizing Grasps of a Simple Polygon with Few Contacts. *Algorithmica*, 44:117–136.
This work is done by one of Prof. Stappen's student. Comparing to Prof. Stappen review work in 2005, this work involves more details on algorithms.

[Chinellato, 2002] Chinellato, E. (2002). *Robust Strategies for Selecting Vision-Based Planar Grasps of Unknown Objects with a Three-Finger Hand*. PhD thesis, University of Edinburgh.
Eris Chinellato is now a research associate at Imperial College London. In this master thesis, Chinellato discussed and ranked various quality functions of grasping optimization.

[Chinellato, 2008] Chinellato, E. (2008). *Visual neuroscience of robotic grasping*. PhD thesis, Jaume I University.

In his PhD thesis, Chinellato tries to connect grasping optimization with perception and fit it into the frame of neuroscience. This thesis won award for the best European Ph.D thesis in robotics in 2008.

[Chinellato et al., 2003] Chinellato, E., Fisher, R. B., Morales, A., and del Pobil, A. P. (2003). Ranking Planar Grasp Configurations For A Three-Finger Hand. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1133–1138.
A refined version of Chinellato's Master thesis.

[Choset et al., 2005] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

[Chronis and Lee, 2005] Chronis, N. and Lee, L. P. (2005). Electrothermally Activated SU-8 Microgripper for Single Cell Manipulation in Solution. *Journal of Microelectromechanical Systems*, 14:857–863.

[Ciocarlie and Allen, 2009] Ciocarlie, M. and Allen, P. (2009). Hand Posture Subspaces for Dexterous Robotic Grasping. *International Journal of Robotics Research*, 28:851–867.

[Ciocarlie et al., 2007] Ciocarlie, M., Goldfeder, C., and Allen, P. (2007). Dimensionality Reduction for Hand-independent Dexterous Robotic Grasping. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3270–3275.

[Czyzowicz et al., 1999] Czyzowicz, J., Stojmenovic, I., and Urrutia, J. (1999). Immobilizing A Shape. *International Journal of Computational Geometry and Applications*, 9:181–206.
This work gives a formal definition of immobilization discusses immobilization in work space. The algorithms proposed in this work is intuitive and efficient.

[Davidson and Blake, 1998a] Davidson, C. and Blake, A. (1998a). Caging Planar Objects with a Three-Finger One-Parameter Gripper. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2722–2727.
This work extends Rimon and Blake's work on one-parameter two-finger caging in 1996.

[Davidson and Blake, 1998b] Davidson, C. and Blake, A. (1998b). Error-Tolerant Visual Planning of Planar Grasp. In *Proceedings of International Conference on Computer Vision*, pages 911–916.
This is the most early work which uses caging to deal with uncertainty. Although not formally proposed, it implies the idea of "grasping by caging".

[Dollar and Howe, 2005] Dollar, A. M. and Howe, R. D. (2005). Towards Grasping in Unstructured Environments: Grasper Compliance and Configuration Optimization. *Advanced Robotics*, 19:523–543.

[Dollar and Howe, 2010] Dollar, A. M. and Howe, R. D. (2010). The SDM Hand : A Highly Adaptive Compliant Grasper for Unstructured Environments. *International Journal of Robotics Research*, 29:1–11.

[Ekvall and Kragic, 2007] Ekvall, S. and Kragic, D. (2007). Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4715–4720.

[Erickson et al., 2007] Erickson, J., Thite, S., Rothganger, F., and Fellow, J. P. (2007). Capturing a Convex Object with Three Discs. *IEEE Transactions on Robotics*, 23:1133–1140.
This is an extended verion of Prof. Erickson's work in 2003. The only difference is that he added some extra explanations.

[Erickson et al., 2003] Erickson, J., Thite, S., Rothganger, F., and Ponce, J. (2003). Capturing a Convex Object with Three Discs. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2242–2247.
Jeff Erickson is a Professor at University of Illinois at Urbana-Champaign, IL, US. He is in the Department of Computer Science and his work concentrates on Algorithms in Topology. In this paper, Erickson proposed the following problem. Given the positions of two fingers and shape information of a target object, how to find the positions for a third finger that can cage the object. He answered this problem by proposing both an exact algorithm and a z-buffering based approximation.

[Fink et al., 2008] Fink, J., Hsieh, M. A., and Kumar, V. (2008). Multi-Robot Manipulation via Caging in Environments with Obstacles. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1471–1476.

[Gerkey and Mataric, 2002] Gerkey, B. P. and Mataric, M. J. (2002). Pusher-watcher: An Approach to Fault-tolerant Tightly-coupled Robot Coordination. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 464–469.

[Glover et al., 2009] Glover, J., Rus, D., and Roy, N. (2009). Probabilistic Models of Object Geometry with Application to Grasping. *The International Journal of Robotics Research*, 28:999–1019.

[Goldfeder et al., 2009] Goldfeder, C., Ciocarlie, M., Peretzman, P., Dang, H., and Allen, P. K. (2009). Data-driven Grasping with Partial Sensor Data. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1278–1283.

[Grier, 2003] Grier, D. (2003). A Revolution in Optical Manipulation. *Nature*, 424:810–816.

[Hammond et al., 2012] Hammond, F. L., Weisz, J., de la Llera Kurth, A. A., Allen, P. K., and Howe, R. D. (2012). Towards a Design Optimization Method for Reducing the Mechanical Complexity of Underactuated Robotic Hands. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2843–2850.

[Handle Project, 2013] Handle Project (Last updated: 2013). The Handle Project. Last accessed: 2013. http://www.handle-project.eu/.

[Harada et al., 2013] Harada, K., Nagata, K., Tsuji, T., Yamanobe, N., Nakamura, A., and Kawai, Y. (2013). Probabilistic Approach for Object Bin Picking Approximated by Cylinders. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3727–3732.

[Hsiao et al., 2010] Hsiao, K., Chitta, S., Ciocarlie, M., and Jones, E. G. (2010). Contact-Reactive Grasping of Objects with Partial Shape Information. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228– 123.

[Hu et al., 2011] Hu, W., Ishii, K. S., and Ohta, A. T. (2011). Micro-assembly using Optically Controlled Bubble Microrobots. *Applied Physics Letters*, 99:094103–094103–3.
Although the bubble micro-robots is named "optically controlled" bubble microrobots, they are actually thermally controlled. Different from optical tweezers which trap beads with the energy of photons, this paper pulls or drags bubbles by thermal effects and by inducing heat in near-by regions with focused projector light.

[Jiang et al., 2011] Jiang, Y., Moseson, S., and Saxena, A. (2011). Efficient Grasping from RGBD Images: Learning using a New Rectangle Representation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3304–3311.

[Jonathan Weisz, 2012] Jonathan Weisz, De la Llera Kurth, A. A. P. K. A. R. D. H. (2012). Towards a Design Optimization Method for Reducing the Mechanical Complexity of Underactuated Robotic Hands. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2843–2850.

[Kallman and Mataric, 2004] Kallman, M. and Mataric, M. (2004). Motion Planning Using Dynamic Roadmaps. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4399–4404.

[Katana, 2013] Katana (Last updated: 2013). Katana Arm Japan Provider. Last accessed: 2013. http://www.revast.co.jp/service/arm/type01.html.
This website is not the official site of Katana Arm. It is the Japanese proxy. Katana arm is a small robotic manipulator produced by Neuronics Intelligent and Personal Robotics. Official site of Katana Arm is http://www.neuronics.ch/cms$_d$e/web/index.php?id = 385s = hardware$_a$rchitecture.

[Kavraki et al., 1996] Kavraki, L., Svestka, P., Latombe, J. C., and Overmars, M. (1996). Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580.

[Kehoe et al., 2013] Kehoe, B., Matsukawa, A., Candido, S., Kuffner, J., and Goldberg, K. (2013). Cloud-Based Robot Grasping with the Google Object Recognition Engine.

In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4248–4255.

[Kuperberg, 1990] Kuperberg, W. (1990). Problems on Polytopes and Convex Sets. In *Proceedings of DIMACS: Workshop on polytopes*, pages 584–589.
Woldzimierz Kuperberg is a mathematician at Auburn University, AL, US. Before his formal expression of caging in this paper, lots of relevent problems were dicussed in mathematical fields. However, it is Kuperberg who first summarizes those discussions and formally proposes the caging problem.

[Lavalle and Kuffner, 2000] Lavalle, S. M. and Kuffner, J. J. (2000). Rapidly-exploring random trees: Progress and prospects. In *Proceedings of International Workshop on the Algorithmic Foundations of Robotics*, pages 293–308.

[LaVelle, 2006] LaVelle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.

[Leeper et al., 2010] Leeper, A., Hsiao, K., Chu, E., and Salisbury, K. (2010). Using Near-Field Stereo Vision for Robotic Grasping in Cluttered Environments. In *Proceedings of International Symposium on Experimental Robotics*.

[Leven and Hutchinson, 2002] Leven, P. and Hutchinson, S. (2002). A Framework for Real-time Path Planning in Changing Environments. *The International Journal of Robotics Research*, 21:999–1030.

[Li and Sastry, 1988] Li, Z. and Sastry, S. S. (1988). Task-oriented Optimal Grasping by Multifingered Robot Hands. *IEEE Journal of Robotics and Automation*, 4:32–44.
Zexiang Li uis a professor of electrical and electronic engineering at University of Science and Technology, Hong Kong. In this work, he discusses in details the basic optimal criteria of grasping and proposes the new task-oriented optimization.

[Liu et al., 2010] Liu, H., Wan, W., and Zha, H. (2010). A Dynamic Subgoal Path Planner for Unpredictable Environments. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 994–1001.

[Liu et al., 2004] Liu, Y.-H., Lam, M.-L., and Ding, D. (2004). A Complete and Efficient Algorithm for Searching 3-D Form-closure Grasps in the Discrete Domain. *IEEE Transactions on Robotics*, 20:805–816.
Yun-Hui Liu is a professor of the department of mechanical and automation engineering at the Chinese University of Hong Kong. Although the title claims about form closure, this work actually discusses both frictional and frictionless cases. Equations in this work are quite clear for beginners.

[Maeda et al., 2012] Maeda, Y., Kodera, N., and Egawa, T. (2012). Caging-Based Grasping by a Robot Hand with Rigid and Soft Parts. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 5150–5155.

[Maeda and Makita, 2006] Maeda, Y. and Makita, S. (2006). A Quantitative Test for the Robustness of Graspless Manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1743–1748.

[Maeda et al., 2004] Maeda, Y., Nakamura, and Arai (2004). Motion Planning of Robot Fingertips for Graspless Manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2951–2956.

[Makita, 2010] Makita, S. (2010). *Robotic manipulation by incomplete grasping: graspless manipulation and 3D multifingered caging*. PhD thesis, Yokohama National University.

[Makita and Maeda, 2008] Makita, S. and Maeda, Y. (2008). 3D Multifingered Caging: Basic Formulation and Planning. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2697–2702.

[Mason, 2001] Mason, M. T. (2001). *Mechanics of Robotic Manipulation*. The MIT Press.
Matthew T. Mason is a professor at the Robotic Institute of Carnegie Mellon University, PA, US. In this book, Mason introduces all basic concepts and physical/mathematical tools, ranging from statics to dynamics. He also teaches a course based on this book. Some materials of the course can be found from his homepage.

[Mason et al., 2012] Mason, M. T., Rodriguez, A., Srinivasa, S. S., and Vazquez, A. S. (2012). Autonomous Manipulation with a General-Purpose Simple Hand. *The International Journal of Robotics Research*, 31:688–703.

[Michel, 2004] Michel, O. (2004). Webots: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, 1:39–42.

[Microsoft, 2012] Microsoft (Last updated: 2012). Introduction to KINECT. Last accessed: 2013. http://www.microsoft.com/en-us/kinectforwindows/.
Although KINECT is a trademark of Microsoft, it is based on a depth camera technology by a Israeli developer PrimeSense. The technology is named structure light. Like its name, the technology calculates depth according to changes of infrared light structures. It is a low-cost solution to perceive depth information. However, it cannot be as precise as other technologies such as Time-of-Flight.

[Miller et al., 2003] Miller, A. T., Knoop, S., Christensen, H. I., and Allen, P. K. (2003). Automatic Grasp Planning Using Shape Primitives. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1824–1829.

[Mirtich and Canny, 1994] Mirtich, B. and Canny, J. (1994). Easily Computable Optimum Grasps in 2D and 3D. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 739–747.

[Mishra et al., 1987] Mishra, B., , Schwartz, J. T., and Sharir, M. (1987). On the Existence and Synthesis of Multifinger Positive Grips. *Algorithmica*, 2:541–558.
Bud Mishra is a professor of computer science and mathematics at New York University. This paper demonstrates lots of key problems in grasping and form closure. It is one of the most frequently cited paper in basic grasping theory.

[Montemayor and Wen, 2005] Montemayor, G. and Wen, J. T. (2005). Decentralized Collaborative Load Transport by Multiple Robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 372–377.

[NaturalPoint, 2013] NaturalPoint (Last updated: 2013). OptiTrack V100:R2. Last accessed: 2013. http://www.naturalpoint.com/optitrack/products/v100-r2/.

[Nguyen, 1986a] Nguyen, V.-D. (1986a). Constructing Force-Closure Grasps. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1368–1373.
This paper is a simplified version of Nguyen's technical report. It discusses how to synthesize force closures of 2D objects.

[Nguyen, 1986b] Nguyen, V.-D. (1986b). The Synthesis of Stable Force-Closure. Technical Report AI-TR 905, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
This report discussed in detail (1) the conecepts of force closures and (2) how to do force-closure grasp synthesis based on shape of target objects. Both 2D and 3D cases are included in this report.

[Niparnan et al., 2009] Niparnan, N., Phoka, T., and Sudsang, A. (2009). Heuristic Approach for Multiple Queries of 3D n-finger Frictional Force Closure Grasp. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1817–1822.

[Onda and Arai, 2011] Onda, K. and Arai, F. (2011). Robotic Approach to Multi-beam Optical Tweezers with Computer Generated Hologram. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1825–1830.

[Parker, 1998] Parker, L. E. (1998). ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation. *IEEE Transactions on Robotics and Automation*, 14:220–240.

[Pereira et al., 2002a] Pereira, G. A. S., Kumar, V., and Campos, M. F. M. (2002a). Decentralized Algorithms for Multirobot Manipulation via Caging. In *Proceedings of International Workshop on Algorithmic Foundations of Robotics*, pages 242–258.

[Pereira et al., 2004] Pereira, G. A. S., Kumar, V., and Campos, M. F. M. (2004). Decentralized Algorithms for Multirobot Manipulation Via Caging. *International Journal of Robotics Research*, 23:783–795.

[Pereira et al., 2002b] Pereira, G. A. S., Kumar, V., Spletzer, J. R., Taylor, C. J., and Campos, M. F. M. (2002b). Cooperative Transport of Planar Objects by Multiple Mobile Robots Using Object Closure. In *Experimental Robotics VIII*, pages 275–285. Springer.

[Pipattanasomporn and Sudsang, 2010] Pipattanasomporn, P. and Sudsang, A. (2010). Object Caging under Imperfect Shape Knowledge. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2683–2688.

[Pipattanasomporn and Sudsang, 2011] Pipattanasomporn, P. and Sudsang, A. (2011). Two-Finger Caging of Nonconvex Polytopes. *IEEE Transactions on Robotics*, 27:324–333.

[Pipattanasomporn et al., 2008] Pipattanasomporn, P., Vongmasa, P., and Sudsang, A. (2008). Caging Rigid Polytopes via Finger Dispersion Control. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1181–1186.

[Pollard, 2010] Pollard, N. (Course time: Spring, 2010). Hands: Design and Control for Dexterous Manipulation. Last accessed: 2013. http://graphics.cs.cmu.edu/nsp/course/16-899/.
This is a course opened by Prof. Nancy Pollard. Nancy Pollard is a professor at the Robotic Institute of Carnegie Mellon University, PA, US. In this course, Nancy surveys lots of robotic hands and discusses how to design robotic hands. Many popular and interesting publications can be found in this course website.

[Ponce, 1996] Ponce, J. (1996). On Planning Immobilizing Fixtures for Three-Dimensional Polyhedral Parts. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 509–514.

[Ponce et al., 1995] Ponce, J., Burdick, J., and Rimon, E. (1995). Computing the Immobilizing Three-Finger Grasps of Planar Objects. *IEEE Transactions on Robotics and Automation*, 11:868–881.
This paper is a preliminary work of 2nd order mobility theory. We can find how those researchers began to consider about surface curvertures of target objects.

[Qiao, 2001] Qiao, H. (2001). Attractive Regions Formed by Constraints in Configuration Space – Attractive Regions in Motion Region of a Polygonal or a Polyhedral Part with a Flat Environment. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1071–1078.
This paper includes a complete introduction of attrative regions. Attractive regions share the same idea with ICS proposed by Attawith Sudsang. Both of them try to define themselves in configuration space and both of them fail to explicitly express configuration space.

[Qiao, 2002] Qiao, H. (2002). Strategy Investigation with Generalized Attractive Regions. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3315–3320.

[Rao et al., 2010] Rao, D., Le, Q. V., Phoka, T., Quigley, M., Sudsang, A., and Ng, A. Y. (2010). Grasping Novel Objects with Depth Segmentation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2578–2585.

[Rimon, 2001] Rimon, E. (2001). A Curvature-Based Bound on the Number of Frictinless Fingers Required to Immobilize Three-Dimensional Objects. *IEEE Transactions on Robotics and Automation*, 17:679–697.
In this work, Prof. Rimon extends his 2nd order theory to 3D objects.

[Rimon and Blake, 1996] Rimon, E. and Blake, A. (1996). Caging 2D Bodies by 1-Parameter Two-Fingered Gripping Systems. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1458–1464.

[Rimon and Blake, 1999] Rimon, E. and Blake, A. (1999). Caging Planar Bodies by One-Parameter Two-Fingered Gripping Systems. *International Journal of Robotics Research*, 18:299–318.
This is the first paper that makes caging an independent topic in robotics.

[Rimon and Burdick, 1996] Rimon, E. and Burdick, J. (1996). On Force and Form Closure For Multiple Finger Grasps. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1795–1800.
This work is a draft version of 2nd order theory.

[Rimon and Burdick, 1998a] Rimon, E. and Burdick, J. W. (1998a). Mobility of Bodies in Contact – I: A New 2nd Order Mobility Index for Multiple-Finger Grasps. *IEEE Transactions on Robotics and Automation*, 14:696–708.
In this work, Prof. Rimon formally proposes the idea of 2nd order form closure. It tries to make up the gap between Prof. Mishra's theory and our common sense.

[Rimon and Burdick, 1998b] Rimon, E. and Burdick, J. W. (1998b). Mobility of Bodies in Contact – II: How Forces are Generated by Curvature Effects. *IEEE Transactions on Robotics and Automation*, 14:709–717.
This work involves some supporting materials to the proposal in Part I.

[Rodriguez, 2013] Rodriguez, A. (2013). A Review of Caging. In *Proceedings of IEEE International Conference on Robotics and Automation*.
In this work, Rodriguez give a review of caging, including some old discussions outside the research field of robotics.

[Rodriguez and Mason, 2008] Rodriguez, A. and Mason, M. T. (2008). Two Finger Caging: Squeezing and Stretching. In *Proceedings of International Workshop on the Algorithmic Foundations of Robotics*.

[Rodriguez and Mason, 2012a] Rodriguez, A. and Mason, M. T. (2012a). Grasp Invariance. In *International Journal of Robotics Research*.

This is a pre-work of "Effector Form Design for 1DOF Planar Actuation". It is the theoretical part of the design.

[Rodriguez and Mason, 2012b] Rodriguez, A. and Mason, M. T. (2012b). Path-Connectivity of the Free Space. *Transaction on Robotics*, 28:1177–1180.

[Rodriguez and Mason, 2013] Rodriguez, A. and Mason, M. T. (2013). Effector Form Design for 1DOF Planar Actuation. In *Proceedings of IEEE International Conference on Robotics and Automation*.
In this work, Rodriguez discussed how to design a general end-effector for 1DOF balls. Despite the design, this paper gives an excellent review of end-effector design.

[Rodriguez et al., 2011] Rodriguez, A., Mason, M. T., and Ferry, S. (2011). From Caging to Grasping. In *Proceedings of Robotics, Science and Systems*.

[Rodriguez et al., 2012] Rodriguez, A., Mason, M. T., and Ferry, S. (2012). From Caging to Grasping. In *The International Journal of Robotics Research*, volume 31, pages 886–900.

[Rus, 1997] Rus, D. (1997). Coordinated Manipulation of Objects in a Plane. *Algorithmica*, 19:129–147.

[Sato, 1996] Sato, T. (1996). Micro/nano Manipulation World. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 834–841.

[Saxena et al., 2006] Saxena, A., Driemeyer, J., Kearns, J., Osondu, C., and Ng, A. Y. (2006). Learning to Grasp Novel Objects using Vision. In *Proceedings of International Symposium of Experimental Robotics*.

[SCHUNK, 2013] SCHUNK (Last updated: 2013). The SCHUNK JGZ industrial gripper. Last accessed: 2013. http://www.schunk.com/schunk_files/attachments/{JGZ}_160_{EN}.pdf.

[Smith, 2013] Smith, R. (Last updated: 2013). Open Dynamics Engine. Last accessed: 2013. http://www.ode.org/.
This website is not the homepage of ODE. Detailed documents and examples can be found in its Wiki page http://ode-wiki.org/wiki/index.php?title=Main$_{P}age$.

[Suarod et al., 2007] Suarod, N., Boonpion, N., and Sudsang, A. (2007). A Heuristic Method for Computing Caging Formation of Polygonal Object. In *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pages 823–828.

[Sudsang and Ponce, 1998] Sudsang, A. and Ponce, J. (1998). On Grasping and Manipulating Polygonal Objects with Disc-Shaped Robots in the Plane. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2740–2746.
In this work, Sudsang propose to manipulate a triangle object without contacts. It is esentially the idea of caging manipulation.

[Sudsang and Ponce, 2000] Sudsang, A. and Ponce, J. (2000). A New Approach to Motion Planning for Disc-shaped Robots Manipulating a Polygonal Object in the Plane. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1068–1075.
In this work, Sudsang propose a framework to perform motion planning and obstacle avoidance with ICS.

[Sudsang et al., 1999] Sudsang, A., Ponce, J., Hyman, M., and Kriegman, D. J. (1999). Manipulating Polygonal Objects with Three 2-DOF Robots in the Plane. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2227–2234.
In this work, Sudsang implemented his contactless manipulation algorithm with real robots.

[Sudsang et al., 1997] Sudsang, A., Ponce, J., and Srinivasa, N. (1997). Algorithms for Constructing Immobilizing Fixtures and Grasps of Three-Dimensional Objects. In *Algorithmic Foundations of Robotics II*, pages 363–380. A K Peters, Ltd.

[Sudsang et al., 2000] Sudsang, A., Ponce, J., and Srinivasa, N. (2000). Grasping and In-Hand Manipulation: Geometry and Algorithms. *Algorithmica*, 26:466–493.
This paper has a detailed introduction of the Inescapable Configuration Space (ICS). Although ICS was represented differently from our isolated space, they can be recognized as the same concept.

[Sudsang et al., 2002] Sudsang, A., Rothganger, F., and Ponce, J. (2002). Motion Planning for Disc-shaped Robots Pushing a Polygonal Object in the Plane. *IEEE Transactions on Robotics and Automation*, 18:550–562.

[Trinkle et al., 1988] Trinkle, J. M., Abel, J. M., and Paul, R. P. (1988). An Investigation of Frictionless Enveloping Grasping in the Plane. *The International Journal of Robotics Research*, 7:33–51.

[Vahedi, 2009] Vahedi, M. (2009). *Caging Polygons with Two and Three Fingers*. PhD thesis, Utrecht University.

[Vahedi and van der Stappen, 2006] Vahedi, M. and van der Stappen, A. F. (2006). Caging Polygons with Two and Three Fingers. In *Proceedings of International Workshop on Algorithmic Foundations of Robotics*, pages 71–86.

[Vahedi and van der Stappen, 2007] Vahedi, M. and van der Stappen, A. F. (2007). Geometric Properties and Computation of Three-Finger Caging Grasps of Convex Polygons. In *Proceedings of IEEE Conference on Automation Science and Engineering*, pages 404–411.

[Vahedi and van der Stappen, 2008a] Vahedi, M. and van der Stappen, A. F. (2008a). Caging Convex Polygons with Three Fingers. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1777–1783.

[Vahedi and van der Stappen, 2008b] Vahedi, M. and van der Stappen, A. F. (2008b). Caging Polygons with Two and Three Fingers. *The International Journal of Robotics Research*, 27:1308–1324.

[Vahedi and van der Stappen, 2008c] Vahedi, M. and van der Stappen, A. F. (2008c). Towards Output-Sensitive Computation of Two-Finger Caging grasps. In *Proceedings of IEEE Conference on Automation Science and Engineering*, pages 73–78.

[Vahedi and van der Stappen, 2009] Vahedi, M. and van der Stappen, A. F. (2009). On the Complexity of the Set of Three-Finger Caging Grasps of Convex Polygons. In *Proceedings of Robotics, Science and Systems*.

[van der Stappen, 2005] van der Stappen, A. F. (2005). Immobilization: Analysis, Existance, and Output-sensitive synthesis. In *Computer-Aided Design and Manufacturing*, pages 162–187. AMS-DIMACS.
Prof. Stappen is a professor of computer science at Utrecht University. This paper is a review article which involves not only the history of immobilization but also some efficient algorithms.

[Vongmasa and Sudsang, 2006] Vongmasa, P. and Sudsang, A. (2006). Coverage Diameters of Polygons. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4036–4041.

[Vstone, 2013] Vstone (Last updated: 2013). Beauto Rover ARM. Last accessed: 2013. http://vstone.co.jp/products/beauto_rover/index.html//.

[Wallack, 1996] Wallack, A. S. (1996). Generic Fixture Design Algorithms for Minimal Modular Fixture Toolkits. In *Proceedings of IEEE International Conference on Robotics and Automation*.

[Wallack and Canny, 1996] Wallack, A. S. and Canny, J. F. (1996). Modular Fixture Design for Generalized Polyhedra. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 830–837.

[Wang et al., 1999] Wang, Z., Ahmadabadi, M. N., Nakano, E., and Takahashi, T. (1999). A Multiple Robot System for Cooperative Object Transportation with Various Requirements on Task Performing. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1226–1233.

[Wang et al., 2003a] Wang, Z., Hirata, Y., and Kosuge, K. (2003a). CC-Closure Object and Object Closure Margin of Object Caging by Using Multiple Robots. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechantronics*, pages 344–349.

[Wang et al., 2004] Wang, Z., Hirata, Y., and Kosuge, K. (2004). Control a Rigid Caging Formation for Cooperative Object Transportation by Multiple Mobile Robots. In *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, pages 1580–1585.

[Wang et al., 2005] Wang, Z., Hirata, Y., and Kosuge, K. (2005). An Algorithm for Testing Object Caging Condition by Multiple Mobile Robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3022–3027.

[Wang et al., 2006] Wang, Z., Hirata, Y., and Kosuge, K. (2006). Dynamic Object Closure by Multiple Mobile Robots and Random Caging Formation Testing. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3675–3681.

[Wang and Kumar, 2002] Wang, Z. and Kumar, V. (2002). Object Closure and Manipulation by Multiple Cooperating Mobile Robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 394–399.

[Wang et al., 2003b] Wang, Z., Kumar, V., Hirata, Y., and Kosuge, K. (2003b). A Strategy and a Fast Testing Algorithm for Object Caging by Multiple Cooperative Robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2275–2280.

[Wang et al., 2009] Wang, Z., Matsumoto, H., Hirata, Y., and Kosuge, K. (2009). A Path Planning Method for Dynamic Object Closure by using Random Caging Formation Testing. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5923–5929.

[Watanabe and Yoshikawa, 2007] Watanabe, T. and Yoshikawa, T. (2007). Grasping Optimization Using a Required External Force Set. *IEEE Transactions on Automation Science and Engineering*, 4:52–66.
Tetsuyou Watanabe is a professor of mechanical engineering at Kanazawa University. He in this work proposes a fast solution to find optimal grasping positions based on the branch-and-bound algorithm.

[Webots, 2013] Webots (Last updated: 2013). Webots. Last accessed: 2013. http://www.cyberbotics.com//.
Webots is a commerical robot simulation software. It uses Open Dynamic Engine for simulation of dynamics. Comparing with those freewares, Webots maintainers tend to fix bugs as soon as possible and tend to make the components as robust as possible.

[Wikipedia, 2013a] Wikipedia (Last updated: 2013a). Marching Cubes. Last accessed: 2013. http://en.wikipedia.org/wiki/Marching_cubes.
The text from this wikipedia link briefly and clearly shows the principles of marching cube. It also includes the essential research papers that relate to this topic.

[Wikipedia, 2013b] Wikipedia (Last updated: 2013b). Venus Flytrap. Last accessed: 2013. http://en.wikipedia.org/wiki/Venus_flytrap.

The text from this wikipedia link briefly and clearly shows the venus flytrap. It is an example of the natural creative which employ caging.

[WillowGarage, 2012] WillowGarage (Last updated: 2012). Overview of the Willow Garage PR2 robot. Last accessed: 2013. http://www.willowgarage.com/pages/pr2/overview. WillowGarage is a leading corporation who develops hardware and opensource software for personal robotics applications. It is a major supporter of international robotic conferences and a major developer of ROS. PR2 and TurtleBot are two representative robots of Willow Garage.

[Yokoi et al., 2009] Yokoi, R., Kobayashi, T., and Maeda, Y. (2009). 2D Caging Manipulation by Robots and Walls. In *Proceedings of IEEE International Symposium on Assembly and Manufacturing*, pages 16–21.

[Zhang and Goldberg, 2001] Zhang, T. and Goldberg, K. (2001). Design of Robot Gripper Jaws Based on Trapezoidal. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1065–1070.

Acknowledgments

"You will fail to have a great career, unless you fail to have a great career"
by Michael Litt at TEDxUW

Once upon a time, at two o'clock in mid-night, I jumped out of my bed, opened my computer and typed lines of codes to realize an idea that suddenly came into my head. Unfortunately, the idea failed. Once upon a time, at three o'clock on a winter afternoon, I downloaded a newly compiled program to my micro-controller and tried to actuate the robot. Disappointingly, the robot committed suicide. Once upon a time, at seven o'clock on a rainy morning. I checked the review status of my papers on https://ras.papercept.net, expecting at least an accepted one. Unexpectedly, none of them were accepted. Once…

After many "once upon a time" failures like that, I managed to type the texts in this dissertation. The failures caused intense suffering. I am a normal guy and I cannot bear that many failures. It were the instructions and encouragements from my three advisers Prof. Tomomasa Sato, Prof. Yasuo Kuniyoshi and Prof. Rui Fukui that helped me get over them. I sincerely appreciate their advising. I got a lot from Prof. Sato who taught me to keep a journal of research and to be active with colleagues. Following Prof. Sato's instruction, I maintained my passion in robotic research. I got a lot from Prof. Kuniyoshi who helped me to find some sparkling points of my poor work, approved me positively and encouraged me to go deeper. Prof. Kuniyoshi's instruction gives me confidence in my topics. I got a lot from Prof. Fukui who covered every detail of my research. Prof. Fukui is an excellent engineer as well as an excellent brother. He successfully changed my computer-science head to a mechano-informatics head.

Besides my three advisers, I would like to thank the other members of my thesis committee, Prof. Yoshihiko Nakamura, Prof. Masayuki Inaba and Prof. Kei Okada. Their passionating comments helped to improve the draft version of the thesis greatly. Prof. Nakamura advised choosing proper terminologies and reviewing extensively about grasp-closure works which increased the strictness of the thesis. Prof. Inaba and Prof. Okada advised testing more objects and discussing about 3D cases which increased the soundness of the thesis.

Moreover, I would like to thank Prof. Taketoshi Mori, Prof. Masamichi Shimosaka and Prof. Hiroshi Noguchi for their discussions and revision suggestions of my research topics and paper publications. I would like to thank Mr. Keita Kadowaki, Mr. Yamato Niwa and Mr. Kentairo Nishi for their help in implementing some of the mechanical systems. I would like to thank Mr. Shuhei Kousaka, Mr. Masahiko Watanabe and Mr. Takuya Sunagawa for their suggestions on circuits diagrams and control boards. I would like to thank Prof. Mamoru Nakamura, Mr. Masayuki Tanaka, Ms. Keiko Hirosue, Ms. Yuki Naruke, Ms. Etsuko Izumiya and Mr. Yuichi Moriya for their assistance in experimental equipments, academic travelling and other school or social affairs.

Furthermore, I would like to thank Dr. Mosatafa Vahedi from Utrecht University and Dr. Alberto Rodriguez from Carnegie Mellon University for their discussions on theoretical fundamentals. I would like to thank Prof. Pham Quang Cuong from Nanyang Technological University and Dr. Carlos Felipe Santacruz from Intouch Health for their smart suggestions in the implementation of certain algorithms. I would like to thank Prof. Matthew Mason from Carnegie Mellon University, Prof. Elon Rimon from Technion Israel Institute of Technology, Prof. A Frank van der Stappen from Utrecht University, Prof. Zhidong Wang from Chiba Institute of Technology, Prof. Yusuke Maeda from Yokohama National University, Prof. Attawith Sudsang from Chulalongkorn University, Prof. David J. Cappelleri from Purdue University, Prof. Bereson Dimitry from Worcester Polytechnic Institute, Prof. Satoshi Makita from Sasebo National College of Technology and Prof. Jianhua Su from Chinese Academy of Science for their suggestions, discussions or encouragements on my caging research. Without them I may have lost my directions. I would like to thank my master thesis adviser, Prof. Hong Liu from Peking University, for his continuing concern of my oversea study. I would like to thank my sister Prof. Zhenzhen Wan, my friends Dr. Yaozhong Zhang, Dr. Tao Luo, Dr. Min Lu, Dr. Jia Pan, Dr. Boxin Shi, Dr. Feng Lu, Dr. Shuqing Han, Dr. Chunhua Geng, Dr. Lu Lu, Dr. Yangfeng Ji, Ms. Ying Shi, Mr. Pengfei Sun, Ms. Xiaoxia Hu and Mr. Huijun He for their role-model effects.

Finally, without the support and care of my parents, I could have never finished my studies. I would like not only to thank but to dedicate my thesis to them.