Caging of Rigid Polytopes via Dispersion Control of Point Fingers

Dissertation Proposal

Peam Pipattanasomporn
Advisor: Attawith Sudsang

May 21, 2009

Part I
Introduction

1 Literature Review

The problem of object caging was originally posed by Kuperberg [1] as a problem of designing a minimum formation of points to prevent an object from escaping to infinity by any continuous rigid motion. A caged object does not need to be immobilized but must be confined within a bounded region. Studies in caging generally involve attempts to loosely envelope the object by means of simple and robust strategies that tolerate uncertainty and imprecision in measurements and controls. In the past few decades, the concept has been applied to various tasks such as grasping and in-hand manipulation [2], [3], [4], [5], [6] motion planning [7], [8] part feeding [9], stable stance computation [10], [11] where manipulators that work altogether as a cage are possibly mobile robots, fingers of grippers, arrays of pins, cylindrical rods, for example.

An interesting class of caging is object grasping, a means to employ fingers to fix the object at a certain configuration. Most grasping works are

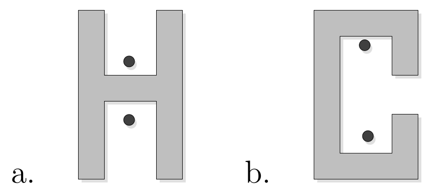


Figure 1: The dark dots and the shaded regions represent fingers and objects, respectively. a. caging by squeezing fingers. b. caging by stretching fingers.

motivated by high-precision applications that require firmly grasped object. Computation of grasps involves finding a configuration that allows the fingers to exert force that can counter balance external forces and torques. This usually relies on sufficient conditions such as force closure and form closure [12]. Caging, on the other hand, relies only on geometrical obstruction in order to prevent object from eluding the grasp. This offers a relatively large connected set of solutions, namely the *caging set* [2] (or *capture region* [13]). A caging set contains finger configurations that once the fingers enter a caging set, the object cannot escape as long as the fingers are maintained to stay inside the caging set. While the object remains caged, it is possible to tighten the cage by controlling the fingers to move in such a way that the range of possible object's motion is reduced. If the cage continues to be tightened over time, the object will eventually be immobilized. This process is called *error-tolerant grasping* [4]. Unlike typical grasping that relies only on the force closure condition, error-tolerant grasping is much more robust and provides the required manipulation precision that cannot be achieved by loosely caging the object. Similar concept resembles to caging and error-tolerant grasping also appeared in stable stance computation [11].

Given a sufficient number of fingers, object caging on a plane can be achieved by evenly placing fingers in a circle formation to surround the object. As long as the distance between any pair of adjacent fingers is kept under an upper bound such as the object's *diameter* [14], or *coverage diameter* [15], the object cannot escape. However, this often leads to inefficient utilization of fingers as two fingers are sufficient to cage most concave objects. Rimon and Blake works [2], have laid fundamental concepts in caging and proposed a numerical solution to determine a caging set that belongs to a given immobilizing grasp of two fingers. Caging with two fingers can be classified by the way their separation distance is maintained. One is caging by squeezing fingers. Moving the fingers closer together will tighten the cage (see Figure 1.a). The other is caging by stretching fingers. The cage is tighten as the finger separation distance increases (see Figure 1.b). It has been proven by Rodriguez and Mason [16] that a caging set of two finger caging can only be either squeezing or stretching or both, for any compact connected contractible object. Regarding to the cage types, the object could escape if the distance is either below or above a certain value called the *critical distance*. A critical distance is either *maximal distance* or *minimal distance*. The maximal distance is the greatest distance such that the object cannot escape as long as the fingers' separation distance does not exceed such distance, Conversely, the minimal distance is the smallest distance that the object cannot escape as long as the fingers' separation distance does not fall below the value. The maximal and the minimal distance serve as an upper or a lower bound separate distance to maintain the caging by squeezing and stretching, respectively. Vahedi and van der Stappen [17], Sudsang and Pipattanasomporn [18] independently proposed algorithms that report all two-fingered caging sets for a given polygonal object. As the number of fingers increases, the problem of reporting all caging sets becomes more complex. One reason is that the caging set associated with a caging configuration can no longer be parameterized by just a critical distance. Erickson et al. [13] studied caging convex object with three fingers and proposed both exact and practical algorithm to render capture region assuming that two robots are fixed on the boundary of the convex object. Their work was extended to non-convex polygon by Vahedi and van der Stappen [19].

To the best of our knowledge, most caging works are restricted in two dimensional workspace and a few number of fingers. Caging with more than three fingers mostly remains unexplored. Even for the three finger caging problem, none of the published works have proposed an efficient method to create a complete catalog of all caging sets. Though caging with diameter or coverage diameter is an easy and intuitive way to cage objects when many fingers are available, it does not provide significant associations with possible error-tolerant grasps the way caging set based approaches can.

Roughly speaking, our goal is to study the problem of polyhedral object caging given any number of fingers. We assume that each finger is a point and the object is a bounded rigid polytope embedded in any finite dimensional Euclidean space ($\mathbb{R}^m$ with Euclidean metric, for some integer m). We aim to identify all the caging sets and recognize their associations with immobilizing grasps. We will precisely define our objectives and scopes in the following section.

2 Problem Overview

This section serves as an in-depth review intended to clarify the problem yet to be solved: object caging and error-tolerant grasping. The first and the second subsection focus on object caging and error tolerant grasping, respectively. The scope and the objectives of our study will be concluded in the last subsection.

2.1 Object Caging

Essentially, object caging is all about mutual geometrical obstruction created by an object and a cage. Both the object and the cage are assumed to be rigid bodies. Each of them can be a single or multiply connected components. Typically, the point fingers are commanded to form the cage to surround the object which is a single rigid body. Ideally, the fingers may follow any continuous trajectories or remain fixed in place as commanded to prevent the object from escaping. In the classical definition [1], when the fingers are fixed and the object cannot travel arbitrarily far from the fingers, the object is said to be caged, see Figure 2.a. If the object can travel far from the fingers, it is said to escape from the cage, see Figure 2.b. On the other hand, if the same phenomenon is observed from the object frame of reference instead of a static one, the formation of fingers (the cage) will be observed as moving together towards infinity, escaping from the obstructing object, see Figure 2.c. During the escape, the shape of finger formation is also preserved i.e. any two formations in the trajectory can be rotated and translated (rigidly transformed) to one another. Whether the object is caged by a finger formation does not depend on the choice of the reference frame so we will stick with the object frame of reference otherwise stated. Doing so permits us to describe configuration of the system with the positions of the fingers alone. Given that the workspace is $\mathbb{R}^m$, the configuration space is $\mathbb{R}^{m \times n}$ where n is the number of fingers. In practice, we usually do not want any finger to move too far from the object since the workspace is limited and fingers located too far from the object obviously do not help in caging the object. Instead of preventing all the fingers from going arbitrarily far from

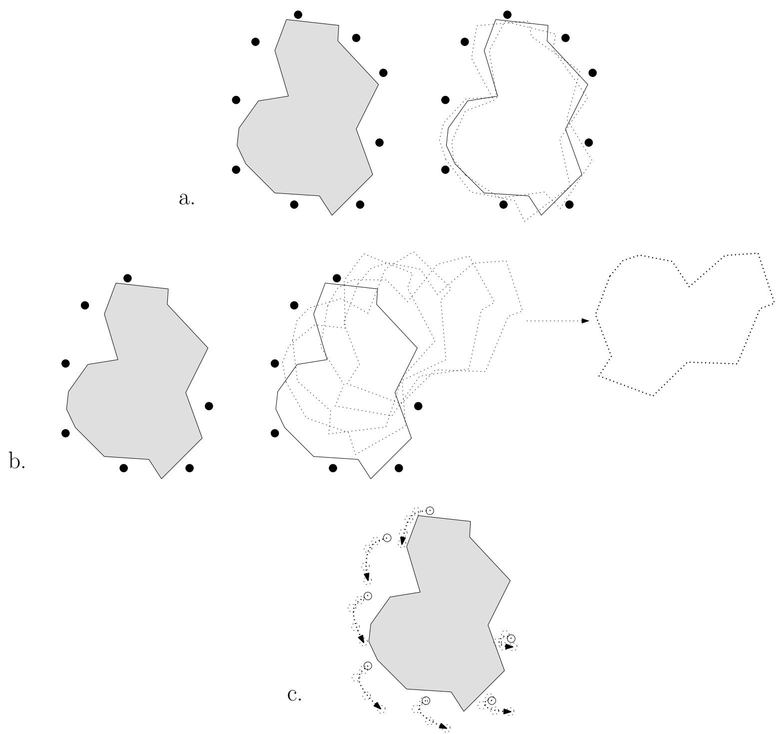


Figure 2: a. the object cannot escape without penetrating the obstructing fingers. b. the object can escape by rotation and translation without penetrating the fingers. c. consider the same situation as b) but the object is seen as a static obstacle, the finger formation can then be rotated and translated to escape from the object.

the object, sometimes preventing *any* of the fingers to go outside a compact workspace is more desirable. Our study of caging will focus on preventing any of fingers from moving outside a compact workspace.

Given an object and a sufficient number of fingers, we can have a large volume of solutions for the caging problem. A solution here refers to an initial finger formation that will guarantee to prevent the object from escaping, i.e. cage the object, if the finger are to form such formation. Let us refer to those formations as *caging formations*. The others are referred as *non-caging formations*.

Though the fingers are commanded to fix in place, their positions of the fingers relative to the object may not remain the same since the object motion cannot be controlled. We may observe, from the object frame, the fingers rigidly move as a whole from place to place. The set of all configurations reachable by movement of the object after the fingers are fixed is exactly the configurations that are reachable by rigidly moving the formation of the fingers. From our point of view, configurations in such set are all equivalent. All of these configurations are either caging formations or non-caging formation, depending whether the set is bounded or not. We therefore categorize the configurations (finger formations) into classes, each class contain configurations that are pairwise reachable by a collision-free rigid motion. For example, configurations in Figure 3.a are grouped together into a class and those in Figure 3.b are grouped together into another class since squeezing the fingers cannot be achieved by rigid transformation of the finger formation. Observing that by slightly varying the separation distance between fingers in the example gives us a new class. Therefore, collection of all these classes is still not discrete. In practice, the fingers cannot be perfectly controlled. The formation shape may deform when the fingers move without perfect synchronization or collide with the object. In [2], any caging formations that are connected in the configuration space neighborhood are grouped together to form a caging set. For example, configurations in Figure 3.a, 3.b are in the same caging set since it is obvious that the fingers can stretch from distance d to d' while the object remain caged (observe that during stretching, the configurations with separation distance in between d and d' are all in the caging set). This grouping yields a finite collection of caging sets, in general. Each caging set represents a collection of cage designs that once the fingers form one such cage, they can transform to another in the collection while the object remain caged. If the goal is to cage the object, we therefore need to keep the fingers within a caging set. A typical strategy to achieve this is composed of the three following steps:

1. **Determine a caging set that suits for the task.** Characteristics of a desirable caging set depend on the objective of the caging task. For example, the *size* of a caging set is crucial for the robustness of caging as it is easier to maintain the fingers within a larger caging set.

2. **Enter the caging set.** In this process, each finger has to be aware of its position relative to the object so that the fingers can be controlled to a caging formation in the caging set.

3. **Remain within the caging set.** Once the fingers are within a caging

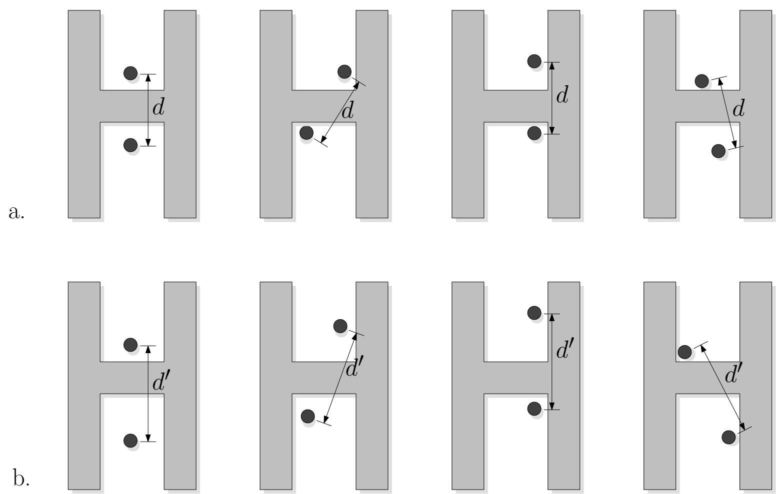


Figure 3: Caging formations with separation distance between the fingers equal to a. d, and b. d'.

set. We can just maintain the fingers' formation shape to remain inside the caging set.

However, describing a caging set is not an easy task since each caging set is embedded in an mn-dimensional configuration space. To do this directly, we need a way to characterize for each caging set its mn-dimensional volume. Alternatively, it is possible to rely on an implicit representation. Recall the object caging problem with two fingers in two dimensional workspace, where an object can be caged by squeezing or stretching fingers. (squeezing: Figure 1.a, 3.a, 3.b; and stretching: 1.b). In this setting, the fingers' separation distance, a one dimensional measure, is necessary and sufficient to describe the formation shape of the the two fingers. Decreasing or increasing the separation distance are therefore the only two approaches to enter or leave a caging set. This implies that a caging formation along with the critical distance to be maintained (above or below) implicitly represents a caging set. However, this only applies to such a simple setting.

With more fingers, we have a higher degree of formation shape control, and a higher dimensional set of all possible formation shapes. Let us consider the case of three fingers in a two dimensional workspace. Observe that we need two additional parameters (for example, the position of the third finger relative to the previous two fingers) to identify the formation's shape instead of the separation distance alone. Varying one of these parameters results in the change of the formation's shape. Entering or exiting a caging set can be done in numerous ways by increasing/decreasing only one, two or all of the three parameters at different speeds. To be precise, the set of all possible approaches has the size equal to that of S^2, a two-dimensional sphere, which is an uncountable set. Recall that previously we have only two possible operations: increasing or decreasing the separation distance; which is equal to the size of S^0. Each approach possibly exits/enters a caging set at a different critical value. As a result, a complete implicit description of a caging set requires a caging formation and its *critical boundary* which is now a 2-dimensional surface instead of a critical distance. Keeping fingers within a caging set is no longer as simple as keeping their separation distance below a value, but to keep the fingers' formation shape parameters within a region wrapped by a 2-dimensional surface. This 2-dimensional surface will depend on the geometry of the object and the parametrization of the configuration space.

Higher number of fingers lead to more control parameters and more complex critical boundaries. Though this will enlarge caging sets and permits us to cage object that we cannot with fewer fingers, it is much more difficult and resource consuming to recognize the critical boundaries of such caging sets. In practice, we may also face problems of controlling multiple parameters inside some caging sets especially when their critical boundaries have multiple curves and turns in the higher dimensional space. Moreover, fingers of some jaws and grippers cannot be controlled freely. Their degree of freedom may be less than mn, the number of fingers times the workspace dimension. An alternative solution is to trade off some caging formations for less computational resources and more simple cage maintenance mechanism supported by most gripper systems. We will choose a portion of a caging set that can be efficiently recognized, and caged instead of the whole caging set. How such a portion is defined, and recognized will be the main study of this work.

2.2 Error-tolerant Grasping

Apart from caging the object, we are also interested in error-tolerant grasping. An error tolerant strategy to grasp the object is to first cage the object and then tighten the cage, reducing the range of all possible fingers/object motions, until the fingers are immobilized with respect to the object. Such state is called *immobilizing grasp*. The fingers and the object can be treated as a single rigid body while being in this state. Both error-tolerant and typical grasping attempt to perform an immobilizing grasp. The difference is that error-tolerant grasping focuses more on caging the object prior to an immobilizing grasp. The fingers will be control to enter a caging set and move to a specific portion of the caging set, then appropriately tighten the cage. This is to ensure the desired immobilizing grasp since it is possible that there are more than one immobilizing grasp in a single caging set. In case of two-finger caging, see an example in Figure 4. Observe the emerging structure induced by the process of tightening the cage. To obtain this structure, we have grouped configurations that: if the fingers begin at one of such configurations and continue to tighten, they will eventually end up in the same set of possible destinations (configurations). Each group is shown as a node in the figure. Each branch from a node represents a possibility to traverse to another node via tightening the cage. Let us call this structure *caging graph*. Caging graph provides a complete map for object caging and error-tolerant grasping. It informs how caging sets, or subsets of caging sets and immobilizing grasps are linked together. We are interested in recovering caging graph for any workspace dimensions and any number of fingers. Again, the boundaries defining the nodes are hyper-surfaces causing the same problem as when determining critical boundaries of caging sets. In addition, the number of possible methods to tighten a cage explodes with increasing degrees of control, which is resemble to how the number of ways to enter/exit a caging set explodes. This greatly increases the complexity of nodes and branches in the caging graph. The reason is that destinations after a cage tightening depend on both the tighten method used and the configuration prior to tightening. It is better to group a combination of configurations and tightening methods that lead to the same set of possible destinations.

The objective of our study in error-tolerant grasping will follow that of caging. We will attempt to simplify the structure and the shape of nodes for computational efficiency by discarding some immobilizing grasps that require sophisticated control to achieve and/or not compatible with a given gripper system.

A caging graph structure from a free configuration space shares much resemblance with that produced by a morse decomposition of the same free configuration space [20]. This is not a surprise since a solution to a caging problem provided by [2] also relies on the stratified morse theory. However, the current practical implementation of general morse decomposition is very

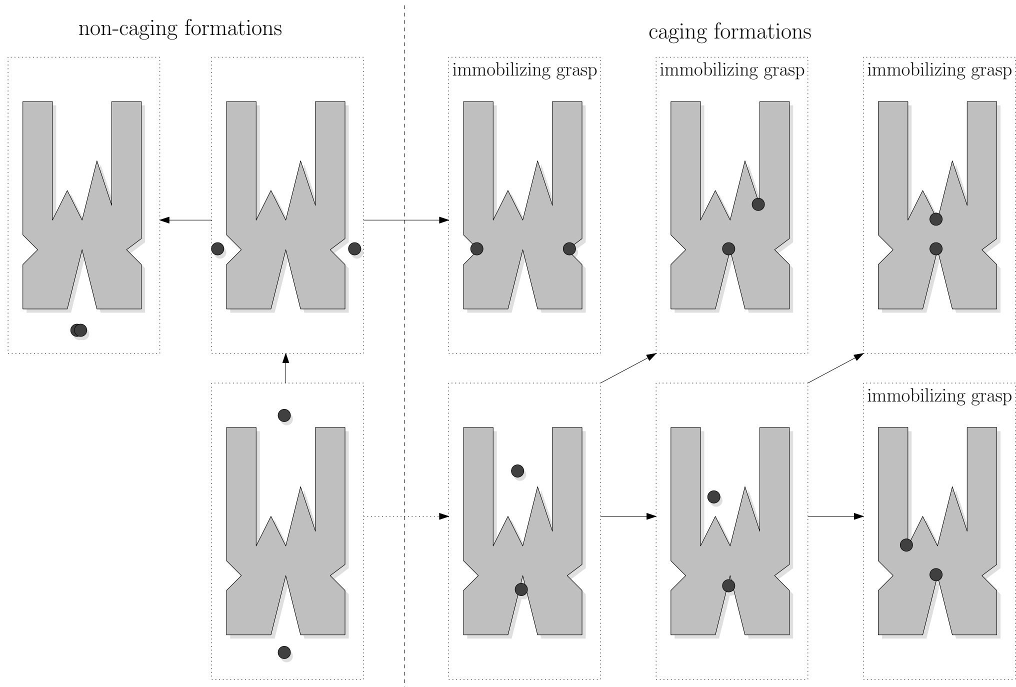


Figure 4: A structure that functions as a map for caging and immobilizing grasp.

limited to small dimensional space. Though the configuration space in our problem is of mn dimensional, it possesses many regular structures for example, the space to be decomposed is constructed from cartesian products of multiple copies of the workspace. We will create an algorithm that will take advantages of assumptions stated earlier to generate the caging graph efficiently.

We will focus on the efficiency of the algorithm in the aspect of the time required for generating the caging graph, which should be polynomial with respect to the number of features of the rigid object. Consider the object that resembles a section of a saw in Figure 10. Suppose that the number of the saw's teeth is k on each side, l_0, l_1, l_2 are much longer than l_3 so that placing two fingers between any two teeth, one on the left and the other on the right side, always result in caging the saw. This way, each finger can be placed any where in $2k-2$ gaps between the teeth but not all the fingers are

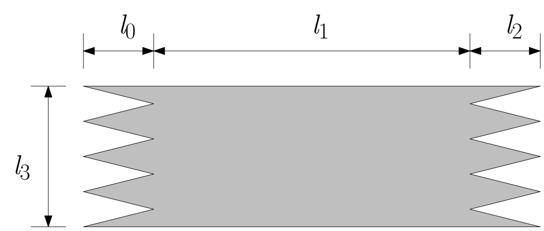


Figure 5: An object which obviously has the number of all possible distinct immobilizing grasps exponentially proportional to the number of fingers, see text.

on the same side. Therefore, the number of all possible immobilizing grasps is $(2k-2)^n - 2(k-1)^n$. Given that v is the number of vertices representing the saw, it can be observed that $2k-2 = v$. This means that the worst case running time of the algorithm must be $\Omega(v^n)$. The running time will be exponential with respect to the number of fingers but will be polynomial to the number of features for a constant number of fingers.

2.3 Scope and Objectives

Our goals are to study the problem of caging and error-tolerant grasping of multiple fingers via a simplified control of finger formation. We assume that each finger is a point, and the object is a bounded rigid polytope embedded in a compact and contractible subspace of $\mathbb{R}^m$. We will design an algorithm capable of constructing a caging graph to assist planning and controlling tasks related to caging and error-tolerant grasping. The caging graph will be constructed from a given rigid polytope and a simplified control capability. The caging graph will contain the information of all the caging sets up to the simplified control capability in a simplified representation, including caging sets' associations with immobilizing grasps. This graph will allow the caging sets and immobilizing grasps to be efficiently enumerated. It will be able to efficiently handle a class of queries: whether a configuration of fingers lies in a caging set, or a node in the caging graph. A caging graph can serve as a simplified map of the configuration space. A manipulation planner may choose a caging set in the caging graph and use its critical distance as a measure to determine error-tolerance if the caging formation is to be deployed there. Alternatively, the planner may choose a node that is guaranteed to lead to an immobilizing grasp or another node that leads to all "acceptable" immobilizing grasps. Feedback-controllers can make use of the query to check whether the current formation is already the desired caging set/node.

Our algorithm will be distinguished from [2] in that the algorithm provided in [2] just numerically computes a critical distance of a single caging set given an immobilizing grasp. Unlike [17] which applies only to two dimensional object, our algorithm will work with any polytopes of finite dimensions.

The following is a checklist for the summarized objectives.

1. Study and simplify the problem of caging and error-tolerant grasping of polyhedral objects with multiple fingers.

2. Find a practical control strategy for the fingers to cage and grasp the object such that the strategy will lead to an efficient computation of caging set and provide a robust approach to cage and grasp the object.

3. Design an efficient algorithm to compute all caging sets with respect to the simplified control.

4. Design an efficient algorithm to compute the caging graph for error-tolerant grasping with respect to the simplified control.

5. Provide an efficient algorithm to query whether a configuration is inside a caging set/subset.

Note that all the algorithms running time will be polynomial to the number of features describing the polytope, the number of fingers are given as considered as a constant.

3 Preliminary Study

In this section, we will present our preliminary study and the direction that we are engaging in solving the problem. The readers are referred to our published works: [18], [21] and [22]; for more detail.

One of the most natural way to cage an object is via finger squeezing. Caging and grasping object by squeezing fingers can be seen as an attempt to envelope object with fingers (maybe more than two). The cage is tightened by reducing the "size" of fingers' formation. Caging by squeezing fingers is undoubtedly one of the most common strategy used by humans and, for this reason, it is supported by most gripper systems. So far, squeezing caging with two fingers is said to be the act of keeping the two fingers' separation distance below a value. We can extend the concept of squeezing fingers to arbitrary number of fingers by defining it to be the act of maintaining the size of the fingers' formation smaller than a value. Note that "the size of fingers' formation" is ambiguous and can be defined in many ways, for example:

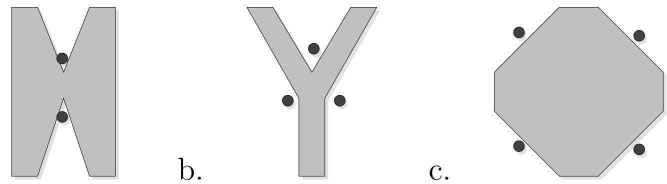
1. **Ring circumference**. The circumference of ring formation is the sum of distance between pairwise *adjacent* fingers: $\delta^1_{\text{ring}}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \equiv \|\mathbf{x}_1 - \mathbf{x}_2\|_2 + \|\mathbf{x}_2 - \mathbf{x}_3\|_2 + ... + \|\mathbf{x}_{n-1} - \mathbf{x}_n\|_2 + \|\mathbf{x}_n - \mathbf{x}_1\|_2$, each $\mathbf{x}_i$ represents a finger position in $\mathbb{R}^m$. Caging by maintaining this below a value is resemble to surrounding the object with fingers. The object cannot move outside unless some fingers move away from the other, increasing the circumference. Figure 6.a-c display examples of caging formations with respect to δ^1_{ring}. Notice that when $n = 2$ (two fingers), $\frac{1}{2}\delta^1_{\text{ring}}$ is exactly the separation distance between the fingers.

2. **Maximum separation distance among each pair of adjacent fingers in a ring**: $\delta^\infty_{\text{ring}}(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \equiv \max\{\|\mathbf{x}_1 - \mathbf{x}_2\|_2, \|\mathbf{x}_2 - \mathbf{x}_3\|_2, ..., \|\mathbf{x}_{n-1} - \mathbf{x}_n\|_2, \|\mathbf{x}_n - \mathbf{x}_1\|_2\}$. This induces caging by maintaining the maximum gap between each pair of adjacent fingers in the ring.

3. **Weighted sum of lattice edge p-lengths**. Let w_{ij} be a non-negative weight for the lattice edge linking between x_i and x_j, $L(i)$ be the set of finger indices such that, for any $j \in L(i)$, $\mathbf{x}_i$ and $\mathbf{x}_j$ form the lattice's edge, we can define the formation's size by: $\delta^p_L(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \equiv \frac{1}{2}\Sigma^n_{i=1}\Sigma_{j \in L(i)} w_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|^p_2$; when $p \in \{1, 2, ..., \infty\}$. Since $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \in \mathbb{R}_+$ (the set of all nonnegative real values) and $x^p : \mathbb{R}_+ \to \mathbb{R}$ is convex and nondecreasing [23], $\|\mathbf{x}_i - \mathbf{x}_j\|^p_2$ is convex for any i, j and so does its non-negative weighted sum: $\delta^p_L(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$. This δ^p_L generalizes the previously exemplified formation's size. For example, observe that $\delta^1_L = \delta^1_{\text{ring}}$, and $\delta^\infty_L = \delta^\infty_{\text{ring}}$ when:

$$L(i) = \left\{ \begin{cases} 1, & i = n; \\ i + 1, & \text{otherwise.} \end{cases} \right\}.$$

The previously stated functions: δ^1_{ring}, $\delta^\infty_{\text{ring}}$, δ^∞_L, including the separation distance between two fingers; shares one common property in that they are all convex. These functions are non-negative weighted summation or maximum



a. b. c.

Figure 6: The fingers form caging formations up to controlling δ^1_{ring} below a value in a., c., e..

among affine compositions of norms [23]. All of them intuitively represents the size of the fingers' formation as they attain their minimal values when all the fingers are at the same point. We have proven that every convex function that is rigid transform invariant possess this property.

Each definition of the formation's size depends on preferences of users and may be limited by the gripper system's mechanics. However, we may not be able to guarantee to cage the object starting at some caging formations in the previously defined notion if we can only control the formation's size. What we aim to study is therefore *caging formations up to a given control capability* i.e. initial configurations where can we guarantee to cage the object using the control capability. It follows automatically that each caging set will also be defined up to the control capability i.e. classes of maximally connected caging formations up to the control capability. In the same manner, non-caging formations also depends on the control capability.

Our task is to compute the caging graph with respect to a given such definition. We will take advantage of convexity to reduce the complexity of the problem and design an efficient algorithm that produce caging graph as proposed.

4 Acknowledgements

This work was support by Chulalongkorn University graduate scholarship to commemorate the 72^{nd} anniversary of his majesty King Bhumibol Adulyadej and The Thailand Research Fund.

References

[1] W. Kuperberg, "Problems on polytopes and convex sets," *DIMACS Workshop on Polytopes*, pp. 584–589, January 1990.

[2] E. Rimon and A. Blake, "Caging 2d bodies by 1-parameter two-fingered gripping systems," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, April 1996, pp. 1459–1464.

[3] C. Davidson and A. Blake, "Caging planar objects with a three-finger one-parameter gripper," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, May 1998, pp. 2722–2727.

[4] ——, "Error-tolerant visual planning of planar grasp," in *Proceedings of IEEE International Conference on Conference on Computer Vision*, January 1998, pp. 911–916.

[5] K. G. Gopalakrishnan and K. Goldberg, "Gripping parts at concave vertices," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, 2002, pp. 1590–1596.

[6] A. Sudsang, J. Ponce, and N. Srinivasa, "Grasping and in-hand manipulation: Experiments with a reconfigurable gripper," *Advanced Robotics*, vol. 12, no. 5, pp. 509–533, 1997.

[7] A. Sudsang, J. Ponce, M. Hyman, and D. J. Kriegman, "On manipulating polygonal objects with three 2-dof robots in the plane," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, 1999, pp. 2227–2234.

[8] A. Sudsang, F. Rothganger, and J. Ponce, "Motion planning for disc-shaped robots and pushing a polygonal object in the plane," *IEEE Transactions on Robotics and Automation*, vol. 18, August 2002.

[9] S. J. Blind, C. C. McCullough, S. Akella, and J. Ponce, "Manipulating parts with an array of pins: A method and a machine," *The International Journal of Robotics Research*, vol. 20, no. 10, pp. 808–818, 2001.

[10] D. J. Kriegman, "Let them fall where they may: capture regions of curved objects and polyhedra," *International Journal of Robotics Research*, vol. 16, pp. 448–472, 1997.

[11] E. Rimon, R. Mason, J. W. Burdick, and Y. Or, "A general stance stability test based on stratified morse theory with application to quasi-static locomotion planning," *IEEE Transactions on Robotics*, vol. 24, pp. 626–641, June 2008.

[12] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Preecedings of IEEE International Conference on Robotics and Automation*, vol. 1, April 2000, pp. 348–353.

[13] J. Erickson, S. Thite, F. Rothganger, and J. Ponce, "Capturing a convex object with three discs," in *Proceedings of IEEE International Conference on Robotics and Automation*, September 2003, pp. 2242–2247.

[14] A. Sudsang, "A sufficient condition for capturing an object in the plane with disc-shaped robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, 2002, pp. 682–687.

[15] P. Vongmasa and A. Sudsang, "Coverage diameters of polygons," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 4036.

[16] A. Rodriguez and M. Mason, "Two finger caging: squeezing and stretching," in *Workshop on the Algorithmic Foundations of Robotics (WAFR) 2008*, December 2008.

[17] M. Vahedi and A. F. van der Stappen, "Caging polygons with two and three fingers," in *Workshop on the Algorithmic Foundations of Robotics (WAFR) 2006*, July 2006.

[18] P. Pipattanasomporn and A. Sudsang, "Two-finger caging of concave polygon," in *Proceeding of IEEE International Conference on Robotics and Automation*, May 2006, pp. 2137–2142.

[19] M. Vahedi and A. F. van der Stappen, "Caging polygons with two and three fingers," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1308–1324, 2008.

[20] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion, Theory, Algorithms and Implementations.* MIT Press, 2005, ch. 6. Cell Decompositions, pp. 170–178.

[21] P. Pipattanasomporn, P. Vongmasa, and A. Sudsang, "Two-finger squeezing caging of polygonal and polyhedral object," in *Proceeding of IEEE International Conference on Robotics and Automation*, April 2007, pp. 205–210.

[22] ——, "Caging rigid polytopes via finger dispersion control," in *Proceeding of IEEE International Conference on Robotics and Automation*, May 2008, pp. 1181–1186.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004, ch. Convex functions, pp. 67–126.