โครงร่างวิทยานิพนธ์

(THESIS PROPOSAL)

ชื่อเรื่อง (ภาษาไทย)	แผนการจับวัตถุที่เสถียรและมีประสิทธิภาพ โดยใช้บริเวณสัมผัสอิสระและการกักขังวัตถุ
ชื่อเรื่อง (ภาษาอังกฤษ)	EFFICIENT AND ROBUST GRASP PLANNING BASED ON INDEPENDENT CONTACT REGION AND CAGING

เสนอโดย	นายธีศิษฏ์ มคะปุญโญ
รหัสนิสิต	5471411721
หลักสูตร	วิศวกรรมศาสตรดุษฎีบัณฑิต
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
สถานที่ติดต่อ	ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพ 10330
โทรศัพท์	+66-(0)2-218-6956, +66-(0)2-218-6957
อีเมล์	teesit.m@student.chula.ac.th
อาจารย์ที่ปรึกษา	ผศ. ดร. อรรถวิทย์ สุดแสง

คำสำคัญ (ภาษาไทย)	การกักขังวัตถุ, การจับวัตถุ, แผนการจับวัตถุ, บริเวณสัมผัสอิสระ
คำสำคัญ (ภาษาอังกฤษ)	CAGING, GRASPING, GRASP PLANNING, INDEPENDENT CONTACT REGION

Dissertation Proposal

Efficient and Robust Grasp Planning based on Independent Contact Region and Caging

Teesit Makapunyo
Advisor: Asst. Prof. Attawith Sudsang

April 21, 2014

1 Introduction

One of the most common but also most mysterious processes in our body is the hand activity, especially grasping. Grasping is a task every human can perform with minimal effort, but nobody knows its underlying mechanism that makes it such a natural ability. Researchers have been trying to unlock this secret for decades and slowly obtain better understanding of human grasping. A noteworthy article recently published by Castiello [1] provides an overview of behavioral neuroscience in grasping. Several studies of brain activities mentioned in the article reveal the potential location of the grasping process in our brain and its strong connection to our visual and tactile sensory. This kind of brain mapping studies has not only answered several pending questions, they also open up even more issues and confirm that the human grasping process is extremely complex. We are still very far from fully understanding the mystery of human grasping.

For ages, human grasping has been a constant source of inspiration for roboticists. An ultimate goal is to build a robot capable of grasping objects the way humans normally do. This objective is desired by applications such as humanoid and service robots, which recently has gained a wide public attention. There are annual competitions such as DARPA Robotics Challenge and Robocup@Home that require grasping ability as a critical part of their missions. Practical autonomous grasping system that works in everyday environment is, however, still undergoing experimental and trial-and-error stages. Nevertheless, there are robots claimed to perform well in separate and specific tasks such as opening a door and drawer [2], making pancakes [3], and baking cookies [4].

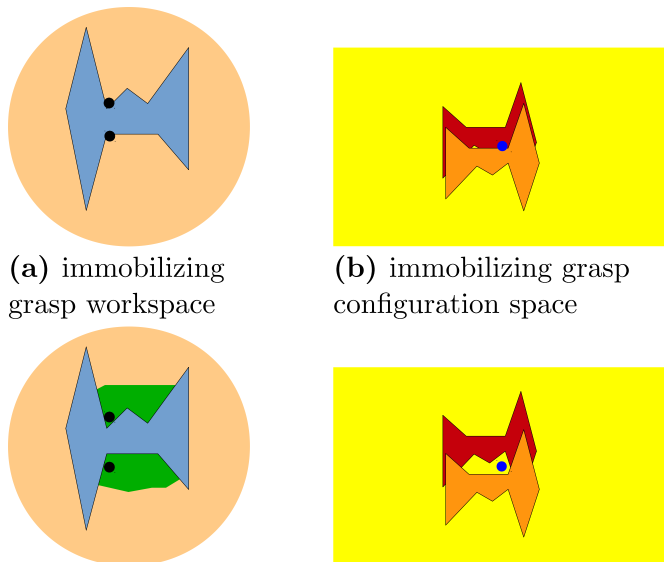
Since the very beginning of research in robotics, grasping has always been one of the main topics of study. Despite decades of effort, practical autonomous grasping by a multi-fingered hand, also known as dexterous grasping, is still in general an open problem. Only limited success can be found in the area of industrial application where structured environment is required. Early grasping research in the late 80s and early 90s focused mainly on establishing the conditions for stable grasps, such as the concept of force closure and form closure. Such conditions have served as the foundation for grasp computation, i.e., finding contact points on the object that guarantees a stable grasp. Studies in this stage toward the late 90s were mostly compartmentalized such that grasp computation is done in isolation from grasp execution, assuming that the robot grasping problem can be divided into many mutually exclusive sub-problems. Most early works assumed a known, perfect model of object, typically polyhedron, that is hardly achievable in practice due to limitation of computing power and reliability of sensors at the time. This limitation had been drastically lifted in the 00s with the advent of cheaper and more accurate sensors, so different approaches heavily based on using sensor data has become very attractive. The arrival of data driven strategy has drawn more researchers from the computer science community particularly those in the field of machine learning. Although results from this new approach appear promising, connection with theoretical foundation of grasping is still weak making generalization and extension difficult.

In this thesis, we propose to tackle the problem of grasp computation by taking into account the availability of real sensor data and real-time requirement. Our approach is to use the solid concept of caging and independent contact regions as the main scheme. Although we plan to make heavily use of data, the scheme is based on well understood theories, as opposed to heuristics derived from machine learning methods. Extending the aforementioned concepts to handle real sensor data under real-time requirement is still an open problem. The main challenge is the real-time performance which requires efficient algorithms and implementation to be developed. Due to the increased performance and more availability of high-performance computing platform (such as GPUs cluster), we believe that this goal is now achievable.

In the next section, we briefly explain some definition that commonly used in robotics grasping. In section 3, we review grasping articles and related topics such as grasp stability, grasp quality index, data-driven grasping, caging and motion planning. Several sections at the end of this proposal state our motivation, problem challenge, methodology and, finally, our work schedule.

2 Basic definition

In grasp analysis, we usually focus on stability of a grasp pose that is a set of contact points where fingers fixate on object boundary. A grasp is represented in *workspace* which an object and contact points are substituted by polytopes and points respectively. To analyze grasp stability, we often uses other interpretation like *wrench space* that represent force and torque exerted on an object and *configuration space* that describe target kinematics relative to other object in workspace.



(a) immobilizing grasp workspace

(b) immobilizing grasp configuration space

(c) caging workspace

(d) caging configuration space

Figure 1. Example of workspace and configuration space showing relation between them. In workspace (figure 1a and 1c), the blue polygon represents a object to be manipulated and black dots represent finger tips that act as obstacle and gain control over target. A green region in figure 1c illustrates caging set, an independent region of each finger placement that can cage the object. In configuration space (figure 1b and 1d), the red and orange polygon are area that the object can not translate into without colliding one or more fingers and the blue dot is the origin of configuration space that represent current position of the object in workspace. It also exhibits a major different between immobilizing grasp and caging, an area of isolated free space.

2.1 Workspace

Workspace usually refers to a 2D or 3D real coordinate space that robot can operate in the scene. For 2D workspace, an object to be grasped is usually represented by polygon or a curved function. While in 3D, it can be in many forms depending on different types of input such as points cloud, polyhedra, triangle mesh and composition of primitive shapes. Manipulator that often refers as fingers is a part of robot that interact and manipulate the target. For simplicity, it usually represents by a set of points in workspace because, in theory, we only interest interaction between object and manipulator that occurs at the contact point where finger's tip touches an object. This way, grasping computation can be invariant to manipulator mechanics.

2.2 Wrench space

Wrench space represent force and torque that exert on an object in 6D space (3D for 2D workspace). It can efficiently resolve resultant force and moment that act on rigid body and evaluate its linear and angular velocity. A wrench that represents forces and torque exerted by manipulator on contact point is called *primitive wrench*. In grasping, the arrangement of the primitive wrenches, especially their convex hull, indicates grasp stability, i.e., the capability of resistance against any perturbation. In some articles, the wrench subspace that formulates a specific task objective is called *task wrench space*. It is denoted by ellipsoid or convex hull of wrenches that manipulator required to exert in order to complete the task.

2.3 Configuration space

Configuration space describes all possible motion states of finger position in workspace having object as obstacle or vice versa. Figure 1 shows an example mapping between workspace and configuration space. A subspace that an object does not overlap fingers is called free configuration space, in short, *free space*. The application of configuration space mostly notable in kinematics such as finding a collision-free path that connected between two configuration, determine degree of freedom and formulate a cage that bound an object within limited region. When all fingers is fixed in places with respect to each other, they are called *preshape* or *finger formation*. Sometimes, finger formation is described by shape function. The parameter of shape function controls distribution and internal structure of finger formation. When fingers surround an object and completely isolate object's free space as depict in figure 1d, a cage is formed. In other words, fingers cage an object when there is no path connecting an initial pose to arbitrary far configuration without colliding an obstacle. Such path, if existed, called an *escaped path*. It is sufficient to assume that arbitrary far position is a state that convex hull of both object and fingers are not overlap. Sometimes, it is sufficient to evaluate grasp pose and caging set only on object boundary. *contact space* can express a finger position on a planar object's boundary with only single parameter, object perimeter. Some literature compute the optimal grasp pose and caging set in this space because their algorithms only compute on object boundary not free space.

3 Literature reviews

3.1 Grasping

As mentioned before, contact points play crucial part in grasping. In grasp theory, grasp stability is a important factor that implies good grasp. A stable grasp is a set of contact points that exert force on object to resist external force and keep the object in equilibrium state. The conditions that define grasp stability from contact points: force closure and form closure grasp are explained in following section. Many grasp planning measure the goodness of grasp from its stability. For those grasp planning, the optimal grasp is a grasp that has highest quality according to their definition of grasp quality. In section 3.1.2, we will review several grasp quality measurements based on grasp stability. Stability may be an indication of good grasp in theory, but, in real robot, it is not sufficient as explained in section 3.1.3. Next, we will discuss independent contact region as one of interesting strategy to grasp an object robustly. Many recent articles investigate a new approach to synthesize a good grasp from sensor and control data using machine learning and heuristics instead of stability analysis. This new kind of grasp planning is often called data-driven grasp synthesis that is briefly discussed in the last section of this topic. The comprehensive reviews in following topics can be found in [5–7].

3.1.1 Closure property

when holding an object in hand at rest position, the total forces and moments that exert on object are balanced and maintain at the equilibrium state. In theory, a grasp that can exert arbitrary force and moment from the contact points to resist arbitrary perturbation is called force closure grasp. Force closure grasp is computed under simple physics model with several assumptions such as frictionless contact, contact with Coulomb friction, rigid body and hard contact model. With those assumptions, the grasp stability can be measured via force and torque exerted at the contact points in wrench space. The minimal requirement for force closure grasp is the origin of wrench space lies strictly within convex hull formed by primitive wrenches [8]. Force closure grasp can be determined by solving linear optimization problem [9] or collision detection [10] in wrench space. Many grasp quality measurement based on force and torque model are also calculated in that space. They are discussed latter in the next section.

Another condition that defines stable grasp is called form closure grasp. Form closure grasp can be accomplished by fixing fingers along object boundary in specific formation that will prevent any infinitesimal object movement. In other words, an object can not move without colliding one or more fingers. This stability criteria utilizes fingers as geometrical constraints that obstruct all object motion as illustrated in figure 1a and 1b. Form closure is closely related to force closure property. Rimon and Burdick [11] showed that form closure grasp is equivalent to force closure grasp with frictionless contact. The different between form closure and force closure grasp had been clearly pointed out by Bicchi [12]. He stated that the different between the two is the perspective in which the problem is analyzed.

Form closure validation is usually formulated as linear programming problem. Each contact point imposes one or more geometrical constraints in the system that forbid object movement along specific direction. The form closure grasp is valid when all possible object motion violate at least one constraints, that is, the existence of feasible solution is necessary and sufficient for negating the form closure condition. Moreover, the minimum number of fingers required for form closure grasp can be solved in this form. It had been studied in [8, 13, 14]. They proved that at least four and seven friction-less contact points are needed to achieve form closure grasp for 2D and 3D cases, respectively. When Coulomb friction is considered, two and three contact points are sufficient in 2D and 3D cases.

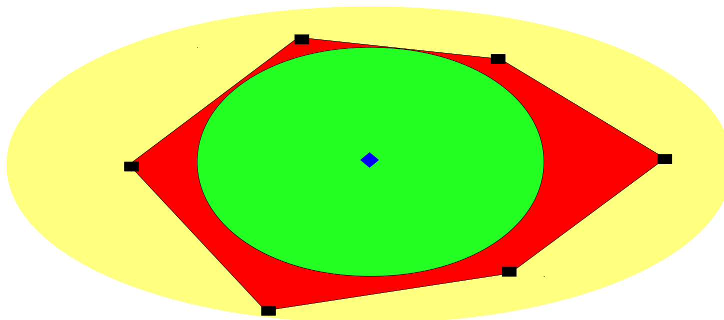


Figure 2. A simplified illustation of wrench space and ϵ-metric. The blue diamond at the center is the origin of wrench space. The black rectangles define boundary of force and torque that can be exerted from manipulator, primitive wrenches, and red polygon is their convex hull that represent capacity to resist external perturbation of a grasp. The ϵ-metric [15] is the largest sphere that fit in the convex hull of primitive wrenches. It is represented by green circle centered at the origin.

3.1.2 Grasp quality index

The existence of closure properties are necessary but insufficient conditions to evaluate grasp quality. A single scalar value representing grasp quality is preferred since it can be compared and optimized to find the optimal grasp. Over several decades, a wide range of analytical procedures to evaluate this value had been introduced. Surez et al. [16] categorized grasp quality measurements into several domains. We will briefly review several noteworthy ones to serve as general idea regarding grasp quality evaluation. The classical one called ϵ-metric proposed by Ferrari and Canny [15] assesses grasp quality in term of stability against external force. It uses radius of largest sphere centered at origin of wrench space that fit into convex hull of unitary primitive wrenches (see figure 2). Zhu and Wang introduced a similar metric called Q-distance [17]. It generalize grasp quality in wrench space as distance from origin to the convex hull of primitive wrenches. Unlike ϵ-metric that treats external force as spherical task wrench space, Q-distance can measures grasp quality when expected perturbation is known by consider task wrench space as convex polytopes. Then, The grasp quality is measured as the largest scale of task wrench space that fit into convex hull of primitive wrenches. Some quality metric derive from algebraic property of grasp matrix that defines the relation between finger force and the net wrench applied to an object in the form of linear optimization problem. Li and Sastry [18] suggested that optimal grasp can be obtained by maximizing the smallest singular value of the grasp matrix. Another similar index called grasp isotropy index was introduced in [19]. Its value is based on ratio between maximum and minimum singular value of grasp matrix. It prefers uniform distribution of force and torque from each contact points, i.e., maximum and minimum singular value are equal.

Beside physics model analysis, geometric relation of contact points can be served as estimation of grasp quality. This concept is easier to implement because it skips difficult computation of physical behavior between an object and fingers. In [19,20], they suggested that uniform distribution of contact points is a sign of good grasp pose. The size of convex hull formed by contact points is also related to grasp stability [21, 22]. Both idea are supported by logical relation between their geometrical condition and grasp quality measured in ϵ-metric. Mirtich and Canny [21] proved that the optimal three-fingered grasp is the largest equilateral grasp because convex hull of primitive wrenches is maximized when all finger spread evenly and the bigger grasp size can resist more external torques.

When a task objective is known, it can be substituted by a set of wrenches that must be applied/resist on an object while being grasped. Those wrenches usually combine into wrench polytopes or ellipsoid called task wrench space. Task-oriented quality measurements [23,24] use the relationship between task wrench space and convex hull of primitive wrenches to classify grasp quality. This idea informs that grasp quality should depend on additional information other than the ability to resist arbitrary force and torque.

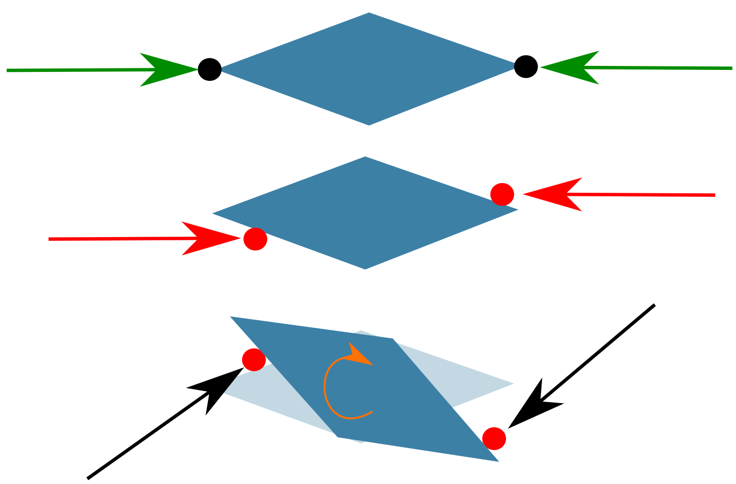


Figure 3. An example of inaccurate robot fail to grasp an object using ϵ-metric. The top image shows approaching direction (green arrow) and position (black dots) of the optimal grasp according to ϵ-metric. The middle image shows the actual path and finger placements of inaccurate manipulator that slightly miss the target. In final image, robot squeezes fingers together and try to grasp an object. The resultant force (black arrows) cause an object to rotate and slip out of the hand, thus the actual grasp is unstable.

3.1.3 Grasping failure

The recent work in [25–27] suggested that closure property is necessary but insufficient condition for practical application. Grasp planning based on analytical methods such as ϵ-metric and Q-distance are very sensitive to errors in position and normal of contact points. The errors are often caused by inaccurate sensors that effect both finger placements and localization process between robot and object. Another factors are accessibility and difficulty for manipulator to reach and grasp at desired position, for example, the optimal grasp, according to ϵ-metric, for rectangle in figure 3 is the furthest pair of its vertices which can not be easily grasped and may end up in unstable pose. Therefore, the optimal grasp computed from inaccurate data might actually be a bad choice in practical robot. There are several ways to mitigate those problems directly and indirectly in following section.

3.1.4 Independent Contact Region

Nguyen [30] introduced a strategy to make force closure grasp more robust called independent contact region (ICR). He suggested that finger placement for force closure grasping should be a region on object boundary rather than a contact point. The contact regions in ICR must satisfy one condition: any grasp that formed by placing all finger within contact regions is force closure grasp. The optimal ICR is a set of the largest contact regions that satisfy force closure condition. The main advantage of ICR is its robustness against errors in finger placement. When grasping is performed in real robot, the actual contact points often differ slightly from target location due to inaccuracy in the system. By placing fingers at the center of ICR, it can ensure that force closure condition still valid under some error tolerance.

Phoka et al. [28, 29] proposed an efficient method to compute optimal ICR in planar case. The process is shortly explained and illustrated in figure 4. For 3D object, Roa and Surez [31, 32] computed an ICR that conform to specified task objective and controlled minimum quality. They first randomly samples a set of force closure contact points that satisfy a task wrench space. From initial grasp pose, a locally optimum one is acquired via oriented search that explore the grasp that can resist the largest external force. Then, the algorithm grows ICR around the optimal one by sampling nearby contact points that can replace the old ones and still fulfill task objective without losing grasp quality. Krug et al. [33] proposed a more efficient algorithm that extend from [32]. They use geometrical analysis on wrench space to derive a better search zone for ICR expansion. More ICR implementation in various scenarios can be found in [34–37].

3.1.5 Data-driven grasp synthesis

Since modern physics simulators are faster and more realistic. Roboticists often use them to simulate grasping result instead of using real robot. Several simulators that focus

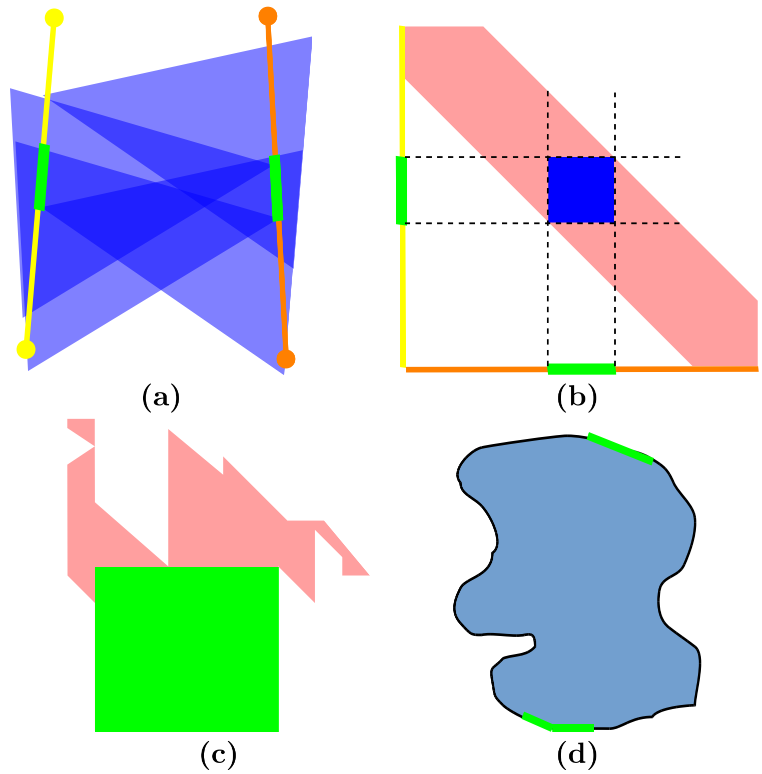


Figure 4. An overview of ICR synthesis proposed in [28, 29]. Figure 4a shows the Independent Contact Region between a pair of edges from geometrical analysis. It is represented by overlapped segments (bold green line) from friction cone (light blue triangle) intersecting opposite edge. In contact space, figure 4b illustrates graspable region between the pair of edges that has force closure property (pink strip) and ICR is defined by blue axis-aligned rectangle and bold green axis. The optimal ICR, a largest graspable region, can be evaluated from the composition of graspable region in every pair of edge as seen in figure 4c. The last figure 4d shows an example of the optimal ICR (a pair of green segments) from their experimental results.

on grasping especially are available, such as GraspIt! [38] and OpenRAVE [39]. Since then, a new approach to synthesize a good grasp from simulated data called data-driven grasping is rapidly developed. Oppose to analytical one, It classifies a good grasp using machine learning and heuristic. The typical strategy of data-driven grasping is building up a knowledge base that maps between object's features to a potential grasp. This knowledge base uses machine learning methods such as support vector machine, neural network and Bayesian network to learn the pattern or model of good grasp from training data. The training data may come from grasp simulator, existing grasp database [40, 41], human demonstration [42], manually labeled example and online trial-and-error. Bohg et al. [7] provided comprehensive reviews and summarize recent studies in data-driven grasp synthesis. The empirical approach like this usually surpluses analytical grasp planning in practical experiment because good grasps learned from carefully chosen set of training data are not only stable but they usually have several properties that increase success rate of grasping, e.g., accessibility and reliability. Those trait inherently come from the power of learning algorithm but it has several drawbacks. The performance of this kind of grasp planning mostly depend on the correspondence of training data and the test subjects that reflect capability of learning process. Its accuracy and performance are significantly decreased when the set of test subjects is large and vary in shape. Another problem is the lack of solid grasp theory to support its good results. This issue makes it hard to understand and analyze its performance: what factors that truly lead to a successful grasp and why.

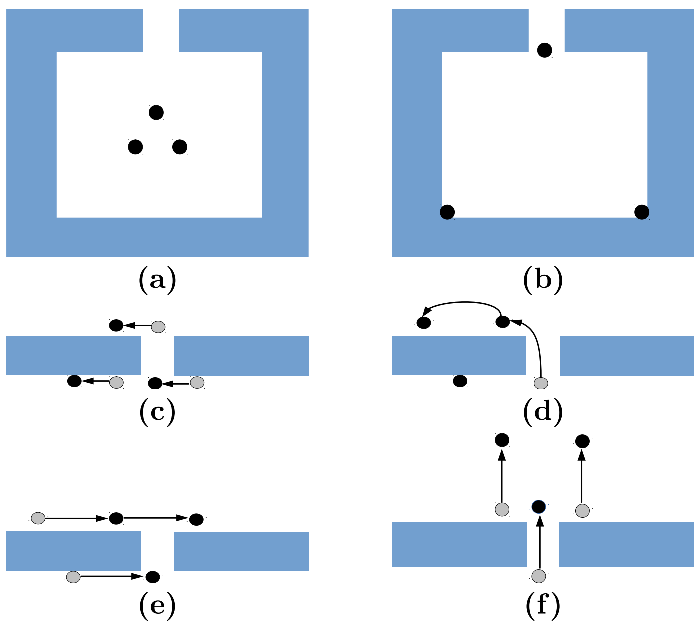


Figure 5. An example scenario shows that three fingers can not maintain caging while stretch the initial finger formation (figure 5a) to reach an immobilizing grasp (figure 5b) from [43]. Figure 5c, 5d, 5e and 5f show an escaped sequence at small gap of the object that may occur during stretching.

3.2 Caging

The caging problem was first defined in [44]. It stated that a polygon, P, in the plane is captured by a set of n points, C if P can not be moved arbitrary far from its original position without overlapping at least one point of C. In configuration space, an object is caged when there is no collision-free path connecting between its current state and state at infinity. Caging can be used as a method to move object from one place to another [45,46] because when the manipulator, a set of points, move as rigid body, an object in the cage will follow. A naive way to construct a cage with sufficient number of fingers is to evenly place fingers around object in a circle formation such that an object can not pass a gap between any two adjacent fingers. The maximal width of the gap that an object can not pass had been discuss in [47, 48].

In grasping, a cage might represent a useful waypoint to grasp an object. A set of cages called pregrasping cages [43] have a trivial strategy that can lead to immobilizing grasp. Pregrasping cage with two-fingered manipulator is the most trivial and studied case. Its caging property remain valid as long as we keep separation distance, a distance between two fingers, below (squeezing) or above (stretching) certain threshold. Under this condition, all two fingers cages are either squeezing cage, stretching cage or both [49]. This characteristic provides a trivial method to grasp an object: depending on the type of cage, either close or open the fingers until immobilizing grasp is formed. The squeezing and stretching condition guarantee that an object can not escape while manipulator grasping it, thus, accurate positioning or closed-loop control are not necessary.

Rimon and Blake [50, 51] were the first that studied planar squeezing cage. Using stratified morse theory, they realized that squeezing cages with locally minimal separation distance are immobilizing grasp and a puncture grasp, the critical contact points before a cage is break, is a configuration with locally maximal separation distance along object boundary. Based on similar concept, Allen et al. [52] proposed a discrete caging graph in contact space and efficient way to compute caging sets by searching through critical points in that graph. On the contrary, Pipattanasomporn and Sudsang [53] computed all planar squeezing and stretching cage via decomposition of configuration space at object boundary. They created a graph which each node represent the optimal cage in a configuration subspace and each edge represent their connectivity that defined by the optimal path connecting between subspace. Then, the caging set is created by traversing the graph with modified shortest path algorithm that use separation distance as a cost. This alogrithm was later enhanced by using convex decomposition of configuration space and cover 3D case in [54, 55].

Finding a pregrasping cage with three or more fingers is not as easy as two-fingered case. Rodriguez et al. [43] showed that the previous strategy, squeezing or stretching fingers, did not maintain a cage under some circumstances (see figure 5). They also generalized the concept of squeezing and stretching to multi-fingers pregrasping cage via the definition of shape function and grasping function but a concrete method to find such cages was first studied by Pipattanasomporn, Vongmasa and Sudsang [56]. They proposed strategy to formulate pregrasping cages called dispersion and concentration control that can find cages for polytopes in arbitrary dimension with multiple fingers. Dispersion control is a convex function that maps from scalar value to a set of finger formation that invariant with respects to rigid transformations, e.g., the maximum distance between any fingers. Like their previous works [53, 54], they built a graph called caging roadmap by decompose an object configuration space to convex subspace and, using the convex function, identify all caging set in the grasp like their two-fingered case. Wan et al. [] used voxel grid in 3D configuration space to illustrate free space and identify holes enclosed by obstacles that is a cage for some predefined finger formations.

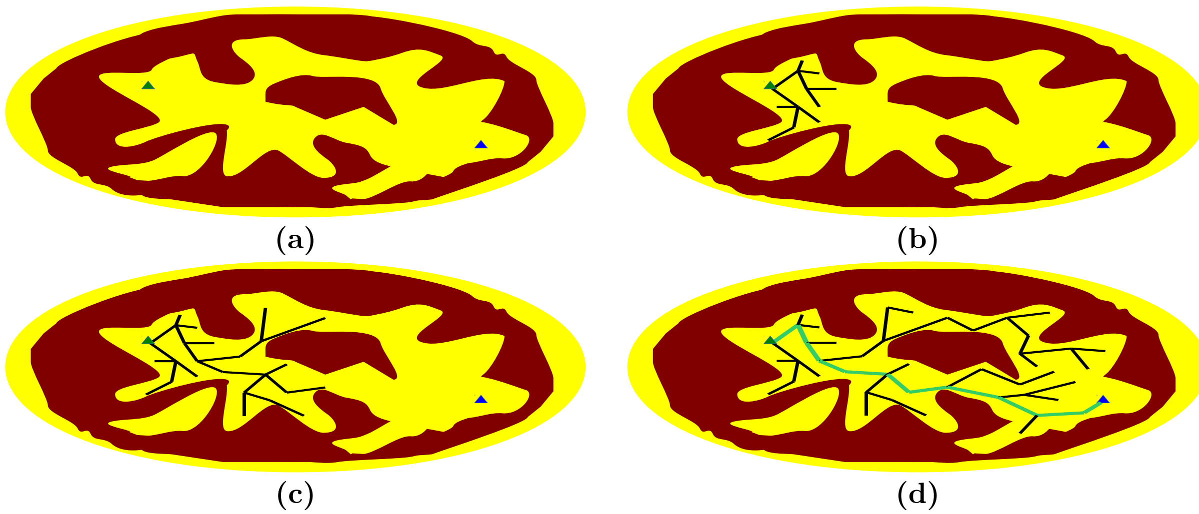


Figure 6. A sequence tree expansion showing probabilistic motion planning process in configuration space. The yellow area is free space and the red one is obstacle. Given a initial position (green triangle) and goal position (blue triangle), motion planning rapidly generate collision-free path (black trees) trying to connect between the two locations. After the solution is found (cyan path), it can be enhanced with path optimization and path smoothing process to obtain a better one.

3.3 Motion planning

The most common problem in many robotics task is transportation. It is often in a form of finding a path to moving something from one location to another with clutter environment and complex control system. A process that used to find those path is called motion planning. A path-finding problem can be think as a dual problem of caging and immobilizing grasp that trying to prevent an object to move away from fingers. Our previous works [57, 58] introduced a framework that measure caging ability of non-caged configuration by sampling an escape path that guide an object away for manipulator. In manipulation task, motion planning is also necessary for finding a collision-free path connecting initial hand position to a predicted grasp pose. Zito et al. [59] proposed a motion planning for push manipulation, the most basic form of manipulation task. The motion planning explores a series of push action that able to move target object to a desire goal pose. It predict the outcome of push action by simulator and rapidly find the results that suit its objective in probabilistic manner.

The state of the art in motion planning lied in probabilistic motion planning. Its main process is construction of a trajectory graph or tree in configuration space where node is a collision-free configuration and edge is a collision-free path connecting between them. Probabilistic motion planning randomly samples a set of node and connect them together until the initial pose is connected to goal configuration (see figure 6). The random distribution may bias toward unexplored area and those two endpoints. Examples of this type of motion planning includes randomized path planner [60], probabilistic roadmap planning [61], rapidly-exploring random tree [62] and single-query bi-directional lazy collision checking planner [63]. Those methods are probabilistically complete, i.e., they are guarantee to find a solution if one existed given no time limit. However they cannot determine if the problem has no solution. The comprehensive reviews of sampling-based motion planning had been discussed in [64].

4 Motivation

As mentioned in section 3.1.3, a fail grasp in real robot is often caused by inaccuracy in sensors and manipulator. Grasp plannings based on stability analysis such as ϵ-metric and Q-distance fail to capture those errors in their framework. The analysis of force and torque equilibrium require an ideal robot that make no mistake but it is still impractical using today technology. In short, the grasp stability computed from a set of contact points alone is not enough to execute a successful grasp. There are many more factors that influence the success rate of grasp which are not clearly formulated in theory. The data-driven grasp planning indirectly take those factors into account by learning a good grasp from training data which provides relatively good results for a small group of test data but the performance decreases as the problem scale up (e.g., larger data-sets) and scale-out (i.e., object shape and properties are more variant). Another drawback of data-driven approach is the absence of solid theory that could support and justify its outcome. It is hard to analyze what and why chosen features are really relevant to a successful grasp. The good result in the experiment from very complex grasp planning seem to be convincing but a grasp planning that only uses simple heuristics [65] is able to give a relatively good performance, too.

We believe that a successful grasp could not be analyzed from a set of contact points alone. Every step of grasping process starting from how to bring manipulator closer to the target, the arrangement of fingers before contact and during grasp and the strategy to maintain object in stable position is the important factors that define a good grasp. For example, forming a pregrasping cage before grasp an object results in a better grasp. It ensures that an object is geometrically constrained and reduce the chance that the object slipped out of hand during grasp. So, in this work, we will investigate the importance of pregrasping cage that effect grasp quality and utilize it in grasp planning. Robustness, stability and accessibility are also necessary traits of a good grasp. In our opinion, an analytical method that best captures those properties is independent contact region. So we intend to implement a grasp planning that features both caging and independent contact region that yields a better performance for practical grasping experiment.

5 Objective and Scope

In this work, we will propose a new grasp planning that combines the strong points of both caging and independent contact region. It must robust to inaccuracy and uncertainty as well as fast and efficient when deploy in real robot system. The new grasp planning should be able to find a good and stable grasp simply from geometrical information of an object. In analytical approach, both independent contact region and pregrapsing cage are promising tools to solve those problem. ICR can maintain a stable grasp when manipulator is imprecise by choosing the best finger placements based on local grasp stability in neighborhood area. On the other hands, pregrasping cage does not contact nor control an object directly but it ensure that an object can not slipped away from manipulator while the fingers closing in to grasp an object, thus, greatly increase success rate of the operation. We also believe that a grasp planning that consider both approaching trajectory from pregrasping cage and stability and robustness from independent contact region should yield better performance compare to conventional methods.

Our primary goal is implementation of practical grasp planning that utilize grasp quality based on integration of ICR and caging. The scope of our work limit to grasping with two-fingered manipulator in 3D physics simulator with Gaussian noise to simulate inaccuracy and uncertainty during grasp. Primitive shapes and daily objects are the test subject for our experiment. The proposed method should have following features:

- The algorithm must be deterministic and efficient in real-time application, i.e., runtime should be less than 30 seconds for most case.
- The outcome is stable grasp that has safe approaching trajectory even in the present of imprecision and uncertainty in the process.
- It should yield better or equal performance (e.g., success rate) when grasp the same test subjects compare to conventional grasp planning.

6 Challenge

The grasp planning that integrate both ICR and caging arises a couple of implementation issues: their compatibility and performance. First, their tasks and solution are fundamentally different. ICR computes an optimal grasp on object boundary from contact normal while caging finds a set of finger formation, that may not directly contact an object, based on kinematics theory. There is still no explicit way to unify and compute them in the same space thus combine them together is not a trivial task. Both methods are also

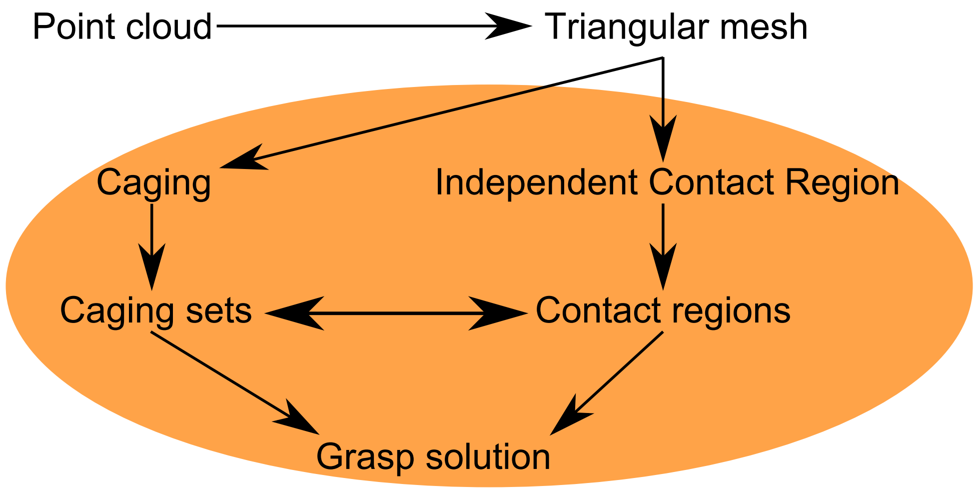


Figure 7. An overview of proposed grasp planning. The initial step is input processing, conversion from point cloud data to appreciate representation like 3D mesh. Then, we compute the most preferable grasp from both ICR and caging. One of the possible integration approach is matching the set of caging and ICR solution that are caging sets and contact regions. The final solution is a pair of faces represented finger placements and its associated cages that manipulator should maintain during grasp. This work flow is one of possible implementation drafts and procedure in orange region may subject to change.

computationally expansive. For average cases, it takes 150 seconds and 500 seconds to compute all the solution of ICR [66] and robust cage [67] respectively. Without efficient optimization, the proposed grasp planning may take more than 15 minutes which is not acceptable in practice. On the bright side, ICR and caging problem can be divided into a lot of smaller, independent tasks that can be processed simultaneously. The practical run-time can be achieved by parallel computing scheme that was recently introduced known as general-purpose computing on graphics processing unit (GPGPU) but parallel programming requires dedicated implementation that has high complexity and prone to coding errors. Thus, implementing grasp planning to be efficient and robust is challenging and worthy researched topic.

7 Methodology

The initial design of our new grasp planning is summarized in figure 7. The input object is an inaccurate 3D point cloud that synthesized from polyhedron mesh. In preprocessing step, the point cloud convert back into a high quality triangular mesh that mostly consist of equilateral triangle with similar size using surface reconstruction methods such as Poisson surface reconstruction, quadric edge collapse decimation and iso-parameterization that available in MeshLab [68]. Then, the algorithm computes caging sets and size of independent contact region between every faces of mesh simultaneously using parallel computing, e.g., OpenMP, CUDA or OpenCL. The best way to grasp an object is then evaluated from those two solutions. The process in final step, how to efficiently combining both solutions and produce a good result, is the main topics of this work.

In previous works [57, 58], we investigated the ability to cage an object of non-caging configuration called partial cage. Contrast to a cage that completely block all escape path of an object, partial cage focuses on configuration that partly obstructs object movement in some direction but an object is still able to escape arbitrary far from fingers. Partial cage measurement estimates partial cage quality in term of the effort an object required to escape away from partial cage. It is indirectly measured by the characteristics of an escaped path that sampling via motion planner. We believe that high quality partial cage is a useful waypoint that lead to a good and stable grasp like pregrasping cage.

As for current progress of this work, we implement an efficient algorithm to find optimal ICR of 3D mesh using parallel computing especially GPU power. The collision detection between point and cone can easily determine force closure condition between two pair of faces (see algorithm 1). Then, ICR is expanded by recursively union and intersect the sets of force closure data between neighboring faces (see [66] for more detail). The preliminary result shows promising performance, i.e., less than 15 seconds for triangular mesh with 20,000 faces. The future work plan is briefly described in the table below.

Task	Duration (month)
Study and implement ICR, caging, and related algorithm	5
Learn an existing physics simulator	3
Develop and analyze possible combination of above methods	7
Summarize the result	3

Algorithm 1 A pseudocode that check force closure property between two triangle

$triA\,[1\ldots 3] \leftarrow$ a list of triangle A's vertices
$triB\,[1\ldots 3] \leftarrow$ a list of triangle B's vertices
$normA \leftarrow$ a normal vector of triangle A
$normB \leftarrow$ a normal vector of triangle B
$fr \leftarrow$ a radius of frictional cone
$isFc \leftarrow True$
for $i \leftarrow 1$ to 3 **do**
  for $j \leftarrow 1$ to 3 **do**
    $isFc \leftarrow isFc \wedge pointInCone\,(triA\,[i]\,, triB\,[j]\,, fr)$
    $isFc \leftarrow isFc \wedge pointInCone\,(triB\,[i]\,, triA\,[j]\,, fr)$
  end for
end for
return $isFc$

References

[1] U. Castiello, "The neuroscience of grasping," *Nature Reviews Neuroscience*, vol. 6, no. 9, pp. 726–736, 2005.

[2] A. Jain and C. C. Kemp, "Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, p. 18071814.

[3] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth, "Robotic roommates making pancakes," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, p. 529536.

[4] M. Bollini, J. Barry, and D. Rus, "Bakebot: Baking cookies with the pr2," in *The PR2 Workshop: Results, Challenges and Lessons Learned in Advancing Robots with a Common Platform, IROS*, 2011.

[5] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1. IEEE, 2000, p. 348353.

[6] A. Okamura, N. Smaby, and M. Cutkosky, "An overview of dexterous manipulation," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 255–262 vol.1.

[7] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp SynthesisA survey," *Robotics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–21, 2013.

[8] B. Mishra, J. T. Schwartz, and M. Sharir, "On the existence and synthesis of multi-finger positive grips," *Algorithmica*, vol. 2, no. 1, p. 541558, 1987.

[9] Y.-H. Liu, "Qualitative test and force optimization of 3-d frictional form-closure grasps using linear programming," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 1, p. 163173, 1999.

[10] X. Zhu, H. Ding, and S. Tso, "A pseudodistance function and its applications," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 344–352, 2004.

[11] E. Rimon and J. Burdick, "On force and form closure for multiple finger grasps," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, 1996, p. 17951800.

[12] A. Bicchi, “On the closure properties of robotic grasping,” *The International Journal of Robotics Research*, vol. 14, no. 4, p. 319334, 1995.

[13] F. Reuleaux, *The Kinematics of Machinery: Outlines of a Theory of Machines. German original (1875). Translated by A. Kennedy.* MacMillan and Co., London, 1876.

[14] X. Markenscoff, L. Ni, and C. H. Papadimitriou, “The geometry of grasping,” *The International Journal of Robotics Research*, vol. 9, no. 1, p. 6174, 1990.

[15] C. Ferrari and J. Canny, “Planning optimal grasps,” in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on.* IEEE, 1992, p. 22902295.

[16] R. Surez, J. Cornell, and M. R. Garzn, *Grasp quality measures.* Institut d’Organitzaci i Control de Sistemes Industrials, 2006.

[17] X. Zhu and J. Wang, “Synthesis of force-closure grasps on 3-d objects based on the q distance,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 669–679, 2003.

[18] Z. Li and S. S. Sastry, “Task-oriented optimal grasping by multifingered robot hands,” *Robotics and Automation, IEEE Journal of*, vol. 4, no. 1, p. 3244, 1988.

[19] B.-H. Kim, B.-J. Yi, S.-R. Oh, and I. H. Suh, “Non-dimensionalized performance indices based optimal grasping for multi-fingered hands,” *Mechatronics*, vol. 14, no. 3, pp. 255–280, 2004.

[20] Y. C. Park and G. P. Starr, “Grasp synthesis of polygonal objects using a three-fingered robot hand,” *The International journal of robotics research*, vol. 11, no. 3, p. 163184, 1992.

[21] B. Mirtich and J. Canny, “Easily computable optimum grasps in 2-d and 3-d,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on.* IEEE, 1994, p. 739747.

[22] E. Chinellato, R. B. Fisher, A. Morales, and A. P. Del Pobil, “Ranking planar grasp configurations for a three-finger hand,” in *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, vol. 1. IEEE, 2003, p. 11331138.

[23] C. Borst, M. Fischer, and G. Hirzinger, “Grasp planning: How to choose a suitable task wrench space,” in *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 1. IEEE, 2004, p. 319325.

[24] R. Haschke, J. J. Steil, I. Steuwer, and H. Ritter, “Task-oriented quality measures for dextrous grasping,” in *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on.* IEEE, 2005, p. 689694.

[25] J. Felip and A. Morales, “Robust sensor-based grasp primitive for a three-finger robot hand,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on.* IEEE, 2009, p. 18111816.

[26] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka, “Physical human interactive guidance: Identifying grasping principles from human-planned grasps,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 899–910, 2012.

[27] J. Weisz and P. K. Allen, “Pose error robust grasping from contact wrench space metrics,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on.* IEEE, 2012, p. 557562.

[28] T. Phoka, P. Vongmasa, C. Nilwatchararang, P. Pipattanasomporn, and A. Sudsang, “Planning optimal independent contact regions for two-fingered force-closure grasp of a polygon,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, p. 11751180.

[29] ——, “Optimal independent contact regions for two-fingered grasping of polygon,” *Robotica*, vol. 30, no. 6, pp. 879–889, 2011.

[30] V.-D. Nguyen, “Constructing force-closure grasps,” *International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988, cited By (since 1996)305.

[31] M. A. Roa and R. Surez, “Independent contact regions for frictional grasps on 3D objects,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on.* IEEE, 2008, p. 16221627.

[32] M. Roa and R. Suarez, “Computation of independent contact regions for grasping 3-d objects,” *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 839–850, 2009.

[33] R. Krug, D. Dimitrov, K. Charusta, and B. Iliev, “On the efficient computation of independent contact regions for force closure grasps,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on.* IEEE, 2010, p. 586591.

[34] J. Ponce, D. Stam, and B. Faverjon, “On computing two-finger force-closure grasps of curved 2D objects,” *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 263–273, 1993.

[35] J. Ponce and B. Faverjon, “On computing three-finger force-closure grasps of polygonal objects,” *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 6, pp. 868–881, 1995.

[36] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet, “On computing four-finger equilibrium and force-closure grasps of polyhedral objects,” *The International Journal of Robotics Research*, vol. 16, no. 1, pp. 11–35, 1997.

[37] J. Cornella and R. Surez, “Fast and flexible determination of force-closure independent regions to grasp polygonal objects,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, p. 766771.

[38] A. Miller and P. Allen, “Graspit! a versatile simulator for robotic grasping,” *Robotics Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110–122, 2004.

[39] R. Diankov and J. Kuffner, “Openrave: A planning architecture for autonomous robotics,” *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, p. 79, 2008.

[40] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, “The columbia grasp database,” in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, p. 17101716.

[41] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, “The princeton shape benchmark,” in *Shape Modeling Applications, 2004. Proceedings*, 2004, pp. 167–178.

[42] A. M. Schmidts, D. Lee, and A. Peer, “Imitation learning of human grasping skills from motion and force data,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, p. 10021007.

[43] A. Rodriguez, M. T. Mason, and S. Ferry, “From caging to grasping,” *The International Journal of Robotics Research*, vol. 31, no. 7, p. 886900, 2012.

[44] W. Kuperberg, “Problems on polytopes and convex sets,” in *DIMACS Workshop on polytopes*, 1990, p. 584589.

[45] G. A. Pereira, M. F. Campos, and V. Kumar, “Decentralized algorithms for multi-robot manipulation via caging,” *The International Journal of Robotics Research*, vol. 23, no. 7, p. 783795, 2004.

[46] A. Sudsang and J. Ponce, “A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane,” in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, p. 10681075.

[47] A. Sudsang, "A sufficient condition for capturing an object in the plane with disc-shaped robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, p. 682687.

[48] P. Vongmasa and A. Sudsang, "Coverage diameters of polygons," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, p. 40364041.

[49] A. Rodriguez and M. T. Mason, *Two finger caging: squeezing and stretching.* Springer, 2009.

[50] E. Rimon and A. Blake, "Caging 2d bodies by 1-parameter two-fingered gripping systems," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, 1996, p. 14581464.

[51] E. Rimon, "Caging planar bodies by one-parameter two-fingered gripping systems," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 299–318, 1999.

[52] T. Allen, J. Burdick, and E. Rimon, "Two-fingered caging of polygons via contact-space graph search," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, p. 41834189.

[53] P. Pipattanasomporn and A. Sudsang, "Two-finger caging of concave polygon," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on.* IEEE, 2006, p. 21372142.

[54] P. Pipattanasomporn, P. Vongmasa, and A. Sudsang, "Two-finger squeezing caging of polygonal and polyhedral object," in *Robotics and Automation, 2007 IEEE International Conference on.* IEEE, 2007, p. 205210.

[55] P. Pipattanasomporn and A. Sudsang, "Two-finger caging of nonconvex polytopes," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 324–333, 2011.

[56] P. Pipattanasomporn, P. Vongmasa, and A. Sudsang, "Caging rigid polytopes via finger dispersion control," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, p. 11811186.

[57] T. Makapunyo, T. Phoka, P. Pipattanasomporn, N. Niparnan, and A. Sudsang, "Measurement framework of partial cage quality," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on.* IEEE, 2012, p. 18121816.

[58] ——, "Measurement framework of partial cage quality based on probabilistic motion planning," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, p. 15741579.

[59] C. Zito, R. Stolkin, M. Kopicki, and J. L. Wyatt, "Two-level RRT planning for robotic push manipulation," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, p. 678685.

[60] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, p. 628649, 1991.

[61] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, p. 566580, 1996.

[62] S. M. LaValle, "Rapidly-exploring random trees a new tool for path planning," 1998.

[63] G. Snchez and J.-C. Latombe, "A single-query bi-directional probabilistic roadmap planner with lazy collision checking," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, R. Jarvis and A. Zelinsky, Eds. Springer Berlin Heidelberg, 2003, vol. 6, pp. 403–417.

[64] P. d. M. B. en Muestreo and U. Compendio, "Sampling-based motion planning: a survey," *Computacin y Sistemas*, vol. 12, no. 1, p. 524, 2008.

[65] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, 2003, p. 36923697.

[66] C. Nilwatchararang, "Computing optimal 2-finger independent contact region for 2D and 3D polygonal object grasping," 2011.

[67] P. Pipattanasomporn, "Caging of rigid polytope via dispersion control of point fingers," 2011.

[68] P. Cignoni, M. Corsini, and G. Ranzuglia, "Meshlab: an open-source 3d mesh processing system," *Ercim news*, vol. 73, p. 4546, 2008.