How to manipulate an object robustly with only one actuator (An application of Caging)

Weiwei Wan, Rui Fukui, Masamichi Shimosaka, Tomomasa Sato and Yasuo Kuniyoshi

***Abstract*— 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. This paper takes into account these merits and designs and implements a gripping hand based on the idea of caging. The gripping hand is concise and offers a low-cost alternative to co-operate with noisy data and low-quality control. According to previous work, we need four fingers to cage any 2D objects. That is to say, if each finger has one, two or three degree of freedoms, we will totally need four, eight or twelve actuators. The large number of actuators would be costly. This paper 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. It successfully lowers costs while maintains high performance. Following the simplified one-actuator design we implement a gripping hand by modifying a SCHUNK RH707 hand and carried out experiments with a manipulator built on the Neuronics Katana arm. The one-actuator gripping hand could work well with common depth cameras (Swiss Ranger) and pick up various objects. It bridges the gap between caging theories and applications and demonstrates the merits of caging.**

I. INTRODUCTION

Lots of theories have been developed in the research field of manipulation and grasping. These theories involve but are not limited to form/force closure [1], enveloping [2], caging [3] and many other optimization works [4]. 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 paper, we propose the design and implementation of 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). It could work with popular depth cameras (Swiss Ranger) and pick up various 2D objects. It bridges the gap between caging theories and applications and demonstrates the merits of caging.

There are two problems in design and implementation of a robotic hand. The first problem is its complexity. Let us compare the following three representative examples — (a) the Schunk JGZ industrial gripper [5], (b) the Barrett Hand [6] and (c) the Robonaut Hand [7]. Note that there are many alike candidates whereas we 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 underactuated 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, we hope to take into account of caging and design a hand that is both high in generality and low in DoFs.

The most related works to the aspect of design complexity are [8], [9] and [10]. Robotic hands in these works are designed according to "constraining" models. Specially, the compliant SDM hand of [9] follows principles of enveloping [11] and won great success in grasping in unstructured environments. [10] discusses in detail how to reduce motor number and designs an under-actuated robotic hand. Like these works, we 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, we find an optimal 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 [12][13].

The second problem is integration with perception devices. Popular works in grasping employ (a) database matching [14][15], (b) RGB cameras [16] or (c) Depth sensors [17][18] to detect objects and synthesize grasping. 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 rangers 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. We hope that with the help of caging, our hand could work with depth sensors and especially the Swiss Ranger. Noises of the Swiss Ranger are +/-10*mm* in depth and +/-7*mm* in horizontal plane. We believe if the hand could work with the Swiss Ranger, it is suitable to most applications.

The works that interest us most in the aspect of collaboration with perception devices are [19], [20] and [21]. These

Weiwei Wan (Research Fellow of Japan Society for the Promotion of Science), Rui Fukui, Masamichi Shimosaka, Tomomasa Sato and Yasuo Kuniyoshi are with Graduate School of Information Science and Technology, The University of Tokyo, JAPAN. wan@ics.t.u-tokyo.ac.jp

works deal with uncertainty in grasping. [19] deals with uncertainty by matching target objects to similar shapes in a database and perform grasping according to the strategies stored in the database. [20] deals with uncertainty by introducing a pre-defined geometric Task Space Region (TSR) for safe pre-grasp. [21] deals with uncertainty by recovering object shapes according to symmetric properties. Similarly but more formally, we deal with uncertainty by using the merits from caging [12].

In summary, we on the one hand employ caging to design. On the other hand, we employ caging to deal with uncertainty. That is to say, we are designing against uncertainty. It not only makes our design concise but also offers robustness to noisy perception devices.

The contribution of this paper is as following. (1) We propose the special ratio of a one-motor caging gripper by performing caging tests on both randomized objects and objects from MPEG-7 shape database. (2) We propose a concise, low-cost and robust gripping hand which inherits the merits of caging.

II. THE BASIC DESIGN

Caging is a pure geometric problem which aims to constrain an object like a bird cage so that the object may not move into infinity. Following [12] and [22][23], we know that (1) **the minimum caging is immobilization** and (2) **at least $2n_{\text{dim}}$ fingers are required to cage any target in n_{dim} dimensional space**. In 2D applications, at least $2 \times 2 = 4$ fingers are required to cage any shape. Therefore, we need to install four fingers to a caging-based robotic hand.

An important problem in installing the four fingers is how to actuate them. One example is the distributed caging manipulator in our previous work [24]. In this design, fingers are attached, actuated and detached sequentially by a single x-y-θ actuator and it requires only three actuators. Although the design lowers system cost, it introduces a time-consuming attaching-actuating-detaching procedure which slows down operation. Unlike the design in [24], we in this paper consider the installation of actuators by quantitative evaluation with caging algorithms.

In [25] and [26], we know that caging is actually a combination of "translational caging" and "rotational caging"[1]. Consequently, we propose to firstly design several candidates of actuator installations by considering "rotational caging". Then, we actuate the "rotational caging" based candidate installations with caging algorithms. In this way, we may have a satisfying combination of "translational caging" and "rotational caging". Fig.1 shows the candidate installation designs[2]. These candidates are based on 'rotational caging". Essentially, they are proposed by biasing towards **equal inter-finger distances** which is required by "rotational caging" before introducing certain apriori information.

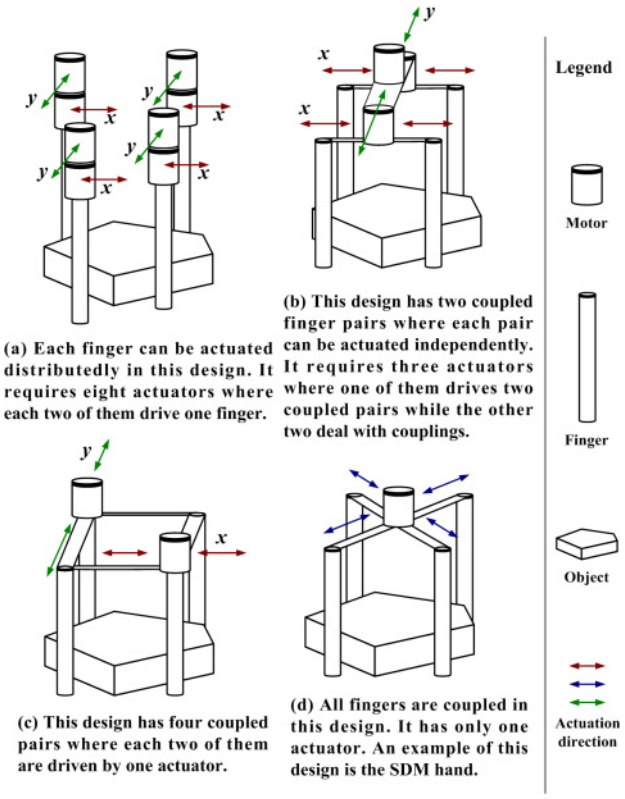


Fig. 1. The candidate actuator installations to cage 2D objects with four fingers

The first candidate, Fig.1(a), is the most intuitive installation. It endows distributed control to each finger and requires as many as eight actuators. Our previous work in [24] is actually a variation of it. The last candidate, Fig.1(d), is fully biased to "rotational caging" and drives four fingers simultaneously. It requires only one actuator and the SDM hand [9] follows its principle. The last candidate fully ensure **equal inter-finger distances**. However, it has **no flexibility for "translational caging"**. The SDM hand solved this inflexibility problem by installing 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.1(b) and Fig.1(c) are better choices.

The difference between Fig.1(b) and Fig.1(c) are their levels of biases towards **equal inter-finger distances**. Fig.1(b) has higher flexibility in position control and it holds more bias towards "translational caging". Nevertheless, three actuators complicate the "gripping" system. We prefer the two-actuator candidate Fig.1(c) as **the basic design**. Fig.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 a caging algorithm and see

[1]If an object cannot escape by x and y motion (translational motion), it is in the state of "translational caging". If an object cannot escape by rotating motion, it is in the state of "rotational caging". If an object cannot escape by any motion (either x, y, rotating motion or the combination of them), it is in the full state of "caging".

[2]In this paper we concentrate on concise "gripping" hands and actuators. We did not consider the shapes of fingers and they are therefore rendered as simply poles.

if it has high successful rates in caging objects. If the basic design has high successful caging rates, it is considered to be a satisfying combination of "rotational caging" and "translational caging". It could cage objects easily and inherits the merits of caging with the help of caging or grasping by caging algorithms.

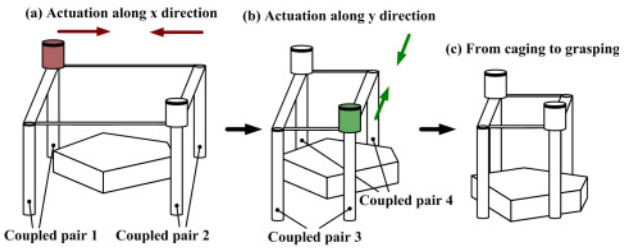


Fig. 2. The basic design in a caging or grasping by caging procedure

III. PERFORMANCE OF THE BASIC DESIGN

A. Random object generator

In order to find some **quantitative supports** for the basic design, we 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 two parameters shown in Fig.3. Readers may refer to Alg.1 and its following paragraphs to better understand the roles of these two parameters.

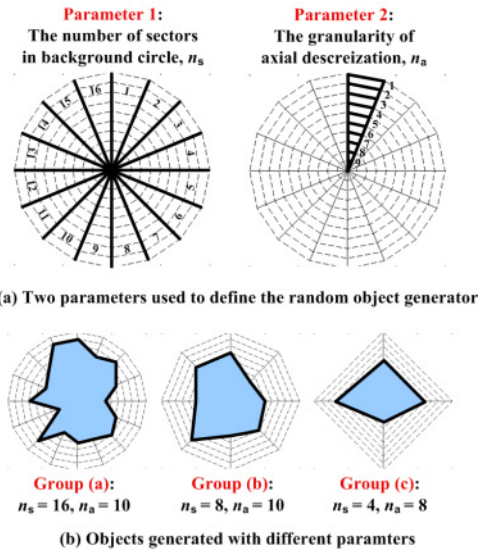


Fig. 3. Our object generator generates random shapes inside a background circle which involves two parameters, namely n_s (the number of sectors, see bold segments in upper-left figure) and n_a (the granularity of axial discretization, see bold segments in upper-right figure). These two parameters are used to enlarge the coverage of the random object generator. The figures in (b) exemplifies some randomized objects. We generate three groups of randomized objects according to different parameter settings.

Our object generator is subject to the following limitations. (1) It cannot generate shapes with inner holes. This limitation is acceptable since we would like to constrain our caging into squeezing caging [27]. (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 and n_a. **Group (a)**: n_s=16, n_a=10. We expect the random shapes generated in this group may be either smooth (small probability) or with sharp prostrution (high probability). Shapes in this group should be, in most cases, easy to be caged owing to their protrusions. **Group (b)**: n_s=8, n_a=10. The random shapes generated in this group has higher bias towards smooth objects and general polytopes while has less bias towards protrusion. We expect that shapes in this group become difficult to be caged comparing with Group (a). **Group (c)**: n_s=4, n_a=10. 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 are hardly seen 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.

Algorithm 1: The random object generator

Data: The number of sectors in background circle, n_s;
The granularity of axial discretization, n_a
Result: A vector of vertices on object boundary, O_{bdry}

```
begin
    O_bdry ← ∅
    for i ∈ {0 : n_a} do
        p_i ← randomize a number between 0 and n_s
        O_bdry ← O_bdry ⋂ p_i
    end
    return O_bdry
end
```

Besides the random object generator, we further evaluate the performance of our basic design with objects extracted from the MPEG-7 shape library (see Fig.4). Shapes in the MPEG-7 library are based on real-world objects, they are more "real" comparing with our random generator[3]). These objects can further confirm the performance of our basic design.

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.

B. Caging test algorithm

Besides the random generator, we need a rapid caging test algorithm to check whether the randomized objects can be caged successfully by the basic design. State-of-the-art

[3]It is easy to find that shapes in the library have lots of protrusions which make them easier to be caged comparing with our randomly generated ones.

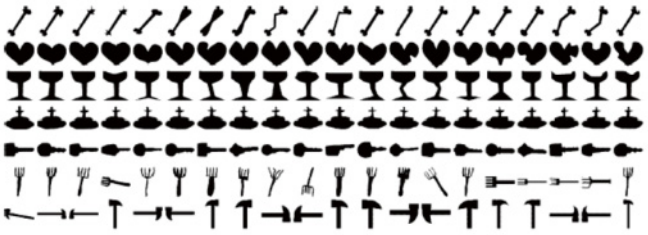


Fig. 4. Shapes from the MPEG-7 library. Each shape has eleven variations and we randomly choose 100 shapes, namely 100*11 = 1100, for experiments.

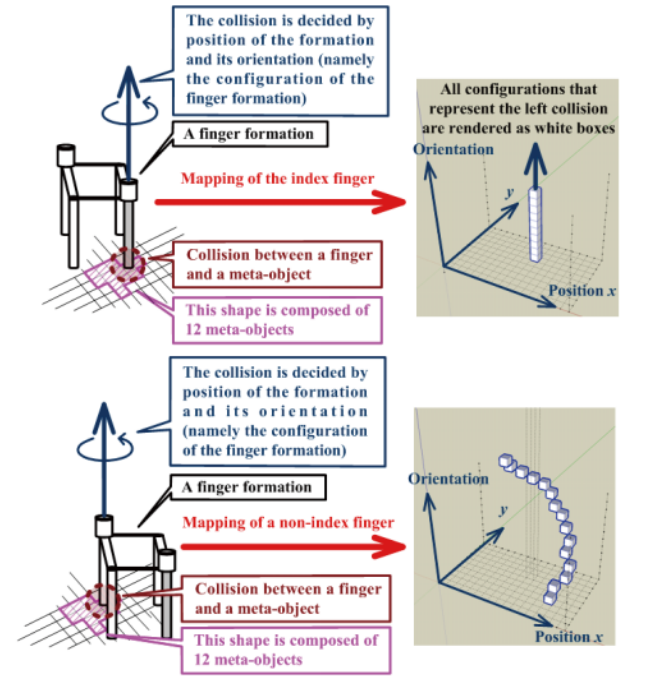
research can only solve caging test problems with certain limintations. For example, [26] is limited to fixed fingers, [28] suffers from incompleteness, [29]'s finger number is limited to two and [30][31]'s finger number is limited to three. There is no suitable caging test algorithms to our evaluation problem. We therefore independently developed a space mapping-based algorithm [32] and use it to perform complete and rapid caging tests. We will briefly revisit the algorithm in this subsection.

Testing whether an object is caged by a finger formation can be viewed as a path planning problem. Here we take the finger formation as the subject robot while take the object as the obstacles. If a path planner cannot plan a path for the subject robot (namely the finger formation) to move through the obstacles (namely the object) without collision, the object is caged. Therefore, we can follow certain complete and fast path planners, for instance [33], to perform complete and rapid dynamic caging tests.

The details of our caging test algorithm involves two phases. **(1) Off-line mapping:** In this phase, our algorithm remembers the relationship between a finger formation and a meta-object. Fig.5 illustrates the idea. This phase prepares a mapping structure for complete and dynamic tests. **(2) On-line testing:** In the second phase, our algorithm decompose the target shape into a set of meta-objects and rebuild the Configuration space (C space) of the finger formation based on the meta-object set. If the finger formation, in its C space, is enclosed by obstacles. The object is caged. Fig.6 illustrates this idea. We cannot show too much details due to page limitation but we encourage readers refer to our illustrations in Fig.5 and Fig.6 or the reference [32] to better understand the two phases.

The off-line mapping saves lots of resources and make dynamic caging tests rapid and complete. We pre-build off-line mappings for 20 finger formations as shown in Fig.7. The 20 formations actually aim to cover the discretization of all accessible finger formations by the basic design. We suppose that the 20 formations can offer enough granularity for evaluating the caging performance of the basic design[4]. See Fig.7 to compare the 20 formations and the random generator.

[4]Since x and y axes of the basic design are symmetric, 4×5 complements 5×4. In this way, using 4×5 discretization is nearly the same as using 9×9.



The white boxes denote obstacles in the C space. Given an object, we can find all the collision, find all the white boxes (see the colored lines in lower-right figure) and rebuild the C space.

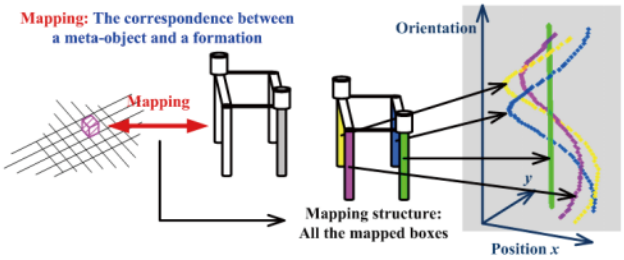
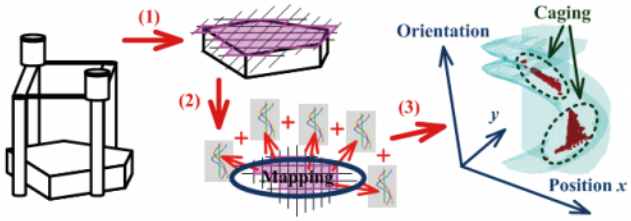


Fig. 5. The off-line mapping between a finger-formation and a meta-object. In C space, the mapping is a structure shown in lower-right figure. Note that a meta-object can be viewed as the smallest box division of a target shape. In this illustration, the target shape is composed 12 meta-objects.



(1) Divide the object into meta-objects (purple area).
(2) Find all the maping structures that correspond to the meta-objects from (1).
(3) Rebuild the C space buy summing up the mapping structures in (2). Here we render the sum (obstacles) with cyan color and render the areas enclosed by the sum with red color for better visual effect. If the finger formation is at a configuration (position and orientation) in the red area, it can cage the object.

Fig. 6. On-line rebuilding of C space. Existence of red areas indicates existence of caging.

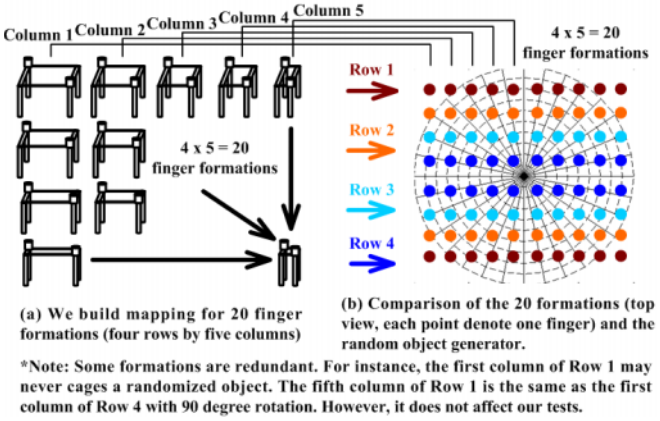


Fig. 7. We pre-build off-line mappings for 20 finger formations to test the caging performance of the basic design.

C. Evaluation with caging tests

With the 3000 random objects and 1100 MPEG-7 shapes, we carry out (1000+1000+1000+1100)*20 = 82000 caging tests to evaluate the performance of our basic design. Fig.8 shows the results of these tests. Here caging is considered successful when the number of boxes in a enclosed red area is larger than 25 (see the red areas in the right part of Fig.5, we use n_b to denote this number).

The number of boxes in a enclosed red area, namely n_b, is an important threshold as it indicates the ability of caging against uncertainty. A larger n_b means a formation could be very "far" from obstacles (or very "far" from colliding the object) and hence offers higher robustness towards uncertainty. In our tests, we divide the C space into 150×150×144 boxes and we set n_b to 25 as the caging threshold. By the way, when we use caging as a pre-grasp step of grasping by caging, we should choose an optimal caging configuration from the enclosed red area that has largest distance from area surface. We will see more details about it in the implementation part of this paper.

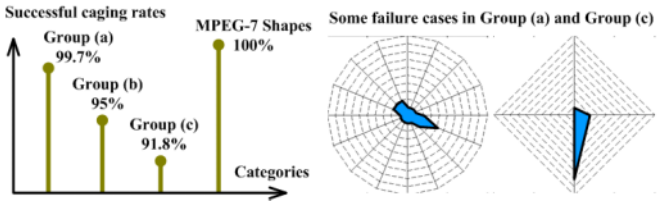


Fig. 8. Performance of the basic design and two examples of failures.

As expected, objects from our random generator are harder to be caged comparing with MPEG-7 shapes. This is because the generator sometimes generates very tiny or very thin objects (see right part of Fig.8) which are difficult to be caged. However, these tiny or thin cases are very rare and we claim that they are not suitable for general caging (take eels for example, they cannot be captured by general fishing net and fish men use special net to "cage" them). In most cases, the performance of our basic design is satisfying. It can cage objects with more than 90% probability. We can draw a conclusion that **most "normal" objects, either they have convex, concave or smooth boundaries, can be caged by the basic design**.

The basic design is quantitatively good. However, it still requires two actuators. In the next section we will dive into the result of the tests and check if the basic design can be further simplified.

IV. FURTHER SIMPLIFICATION

The results in Fig.8 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 to delete one actuator. However, deleting one actuator changes the 20 formations. For example, when the x-actuator is deleted (the red one in Fig.2), the basic design can no longer be actuated from one column to another. That means the 4×5 = 20 formations become a single column of 4 formations. When the y-actuator is deleted (the green one in Fig.2), the basic design can no longer be actuated from one row to another. That means the $4 \times 5 = 20$ 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 4×5. If it were 5×4, it would be a better choice to delete x actuator. 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. Decomposed view of the results in Fig.8 could help solve this problem. Fig.9 shows the decomposed view of Fig.8. In this figure, successful caging rates of each row are illustrated respectively.

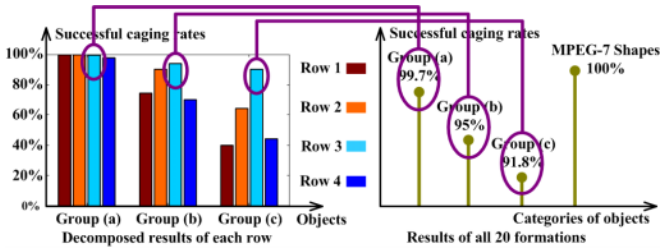


Fig. 9. Decomposed view of the successful caging rates in Fig.8. The left figure shows the successful caging rates of each row of the 20 formations on the three random groups with correspondent color bars. The right figure is a copy of the left part of Fig.8.

Different color bars in the left part of Fig.9 corresponds the successful caging rates of different rows in Fig.7. It is easy to find that the third row of formations, namely the row with "cyan" color, has highest successful caging rates. Actually, the "cyan" row not only 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

successful in the left part of Fig.8 for better comprehension. Here we copy the left part of Fig.8 to the right of Fig.9 for convenience. Successful rates of the "cyan" row on the three groups of random objects (left figure) are nearly the same as successful rates of all the 20 formations (right figure). 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 and the inter-finger distance of the fixed actuator should be around the "cyan" row.** Fig.10 shows the idea. After simplification, only the actuator along x axis remains (see the right part of Fig.10).

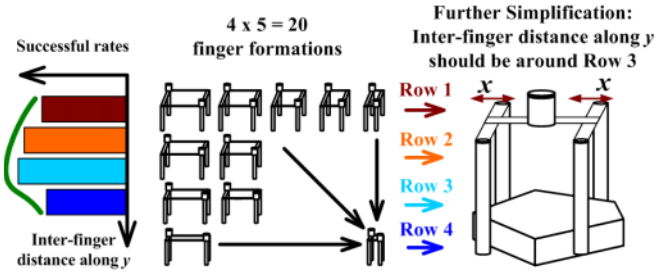


Fig. 10. Further simplification of the basic design. The left part of this figure shows correspondent successful caging rates of each row of the middle formations. Inter-finger distance should be neither too large (the red bar) nor too small (the blue bar) to ensure high successful caging rates. The inter-finger distance around the "cyan" row is the best choice for simplification (see the installation design in right part).

Let us retrospect this simplified design and compare it with Fig.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.9. Fig.11 shows in depth the specification of Fig.9. In this figure, caging rates of each of the 20 finger formations in Fig.7 are illustrated respectively. The finger formation with larger caging rate has a larger circle size. Successful caging rates of the simplified design should be roughly the sum of "cyan" circles (there is redundancy in addition) while successful caging rates of the design in Fig.1(d) is roughly the sum of diagonal circles (there is redundancy in addition). The simplified design has higher sum of circle sizes (higher successful caging rate) and it is a better design comparing with Fig.1(d).

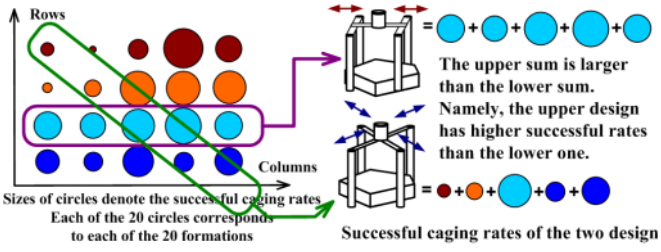


Fig. 11. Further decomposition of Fig.9. Different of circles in this figure correspond to successful caging rates of each finger formation in Fig.7. This figure could roughly confirm that the simplified design is better comparing candidate Fig.1(d).

Now, we can implement a one-actuator gripping hand by using the **simplification rule**. The implementation would be concise as well as effective. It could benefit from all merits of caging, like dealing with noisy perception devices and low-quality control.

V. IMPLEMENTATION

Before implementation, we should first digitalize the "cyan" row. According to ratios of the formation rows in Fig.7, we set fingers as following. For an object in a background circle of diameter 8, we choose 3 as its fixed inter-finger distance. That is to say, the "cyan" row is digitalized into a 8:3 ratio[5]. Then, we can implement a gripping hand based on this ratio. The left part of Fig.12(a) and (b) illustrates settings of fingers and their installation on a RH707 hand. Here, the range of x axis actuation is set between $10mm$ and $80mm$ while the fixed inter-finger distance along y axis is fixed to $30mm$ to maintain the 8:3 ratio. **There is no special mechanisms in this implementation. It is nothing more than a "gripping" hand. However, the 8:3 ratio, which is based on lots of caging tests, changes the essence of "gripping". According to the analysis in Section IV, this implementation should be able to cage or grasp by caging objects inside a $80mm$-diameter background circle with more than 90% successful caging rate. Moreover, it would be robust to co-operate with noisy devices and low-quality control.**

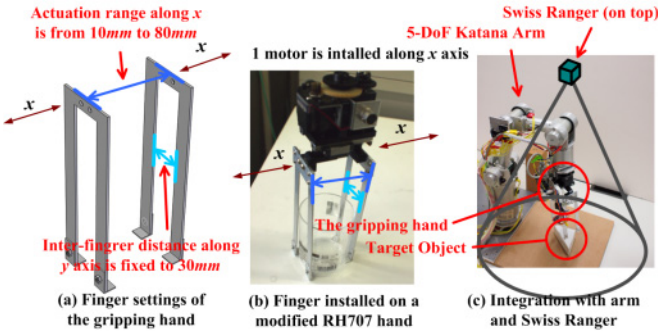


Fig. 12. Implementation of our gripping hand the integrated system.

We integrate the gripping hand with a 5-DoF Katana Arm and a SR4000 Swiss Ranger to perform various caging and grasping by caging tasks. Fig.12(c) shows an overview of the integration. Perception device, namely the Swiss Ranger, is installed on top of the arm. As discussed, the Swiss Ranger suffers from large noises and perceives top-view 2D shapes of target objects with +/- $7mm$ precision. Fig.13 demonstrates the noises. It compares a real object and its correspondent polytope from the Swiss Ranger. The perceived polytope differs a lot from the real triangular object. Explicitly calculating force/form closures based on this noisy polytope is quite tough and causes danger in "picking up" tasks.

[5]This is a rough ratio. Actually, a value between 4:1 and 2:1 will always do. Readers may question why is this ratio a good candidate. We suppose this ratio has some relationship with the principles of caging and would like to explore more about it in the future.

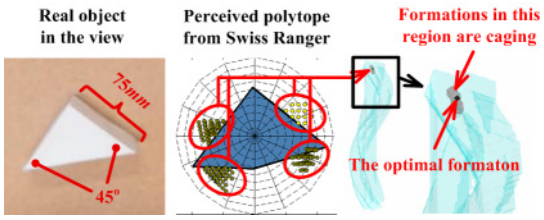


Fig. 13. Perception devices introduce noises in perception. Positions of the first finger of a formation is denoted by light yellow, we introduce this color to better indicate orientations of the finger formations.

Thanks to caging, our hand design can be robust to the noises. The yellow points in Fig.13 show the caging finger formations of the gripping hand. They correspond to configurations (position and orientation) in all caging areas of C space (recall the red areas in Fig.6 and see the red area in right part of Fig.13). Any one of the caging formations could cage the object and we can choose an optimal one from the them as the pre-grasp caging formation. The optimal formation corresponds to a formation that has largest *minimum distance* from the surface of the caging area. It therefore has maximum robustness to uncertainties, avoiding both collision and loss of caging caused by either perception noises or low-quality control. Details of the optimal formation was discussed in our previous work [32]. We can calculate the optimal caging formation for our hand online and pick up objects robustly in a grasping-by-caging way.

The following steps summarize the whole procedure of a grasping-by-caging task performed by our gripping hand.

Step 1: Get the cloud points from the Swiss Ranger and detect the target objects. Fig.14 illustrates this step.

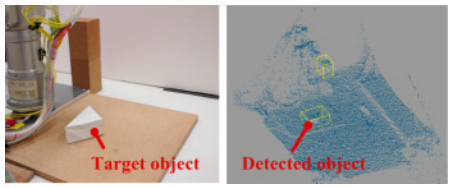


Fig. 14. In the first step, we extract the basic shape of target object by using the cloud points collected from Swiss Ranger.

Step 2: Calibrate the object position and simplify the extracted shape into a polytope. Fig.15 illustrates this step.

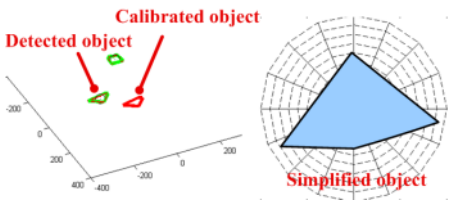


Fig. 15. In the second step, we simplify the extracted shape into a polytope and calibrate its position.

Step 3: Update the C space of the polytope and find all the caging configurations. The caging configurations may separate into several disconnected components. We employ the component that has largest number of configurations and check whether the number of configurations in this component are larger than n_b. If it is smaller, the algorithm rejects caging the target object and caging is not safe. Fig.16 illustrates this step.

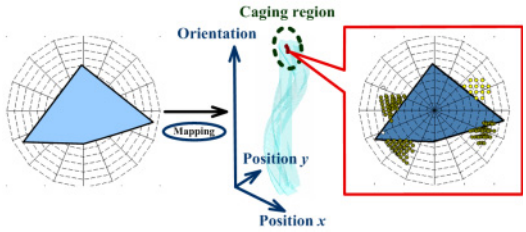


Fig. 16. In the third step, C space of the polytope is updated. The red region in this figure denotes all those caging configurations and it is the single component. In $\mathcal{W}$ space, the configurations in the red region corresponds to a series of formations (see right part). We compare it with n_b.

Step 4: Find the optimal caging and actuate the manipulator. Fig.17 illustrates this step.

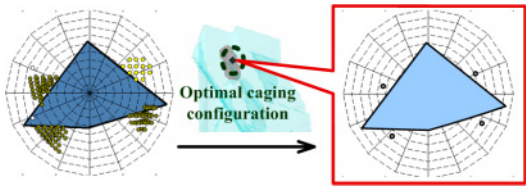


Fig. 17. In the fourth step, we calculate the optimal caging configuration from all caging configurations in the largest component. This optimization procedure is done by measuring the distance between a configuration in the component and the surface.

Fig.18 shows an actuation based on these four steps. In this figure, the designed hand cage and "grasp by caging" the target object and move it to another place. Besides Fig.18, we make demonstrations with several other objects to show the superiority of our design. Fig.19 shows the details of the other objects. **Their demonstrations have been compiled into a video attachment accompanying this paper. The video not only includes caging and "grasping by caging" tasks of various objects like boxes, octagons and concave polytopes but also includes lots of comparisons with other designs and failures.** We strongly recommend readers refer to the video to better comprehend our design, especially the fingers settings and caging.

VI. CONCLUSIONS

In this paper we design and implement a "gripping" hand based on the idea of caging. After comparing the caging performance of $4 \times 5 = 20$ finger formations, we found a 8:3 ratio which offers highest successful rates in one-motor hand design. The ratio makes our design concise as well as effective. We implement our design based on a RH707 hand and demonstrate its performance with Katana Arm and Swiss Ranger. The design inherits all merits from caging and works robustly with noisy perception devices. To our best knowledge, this work is the first attempt to design a

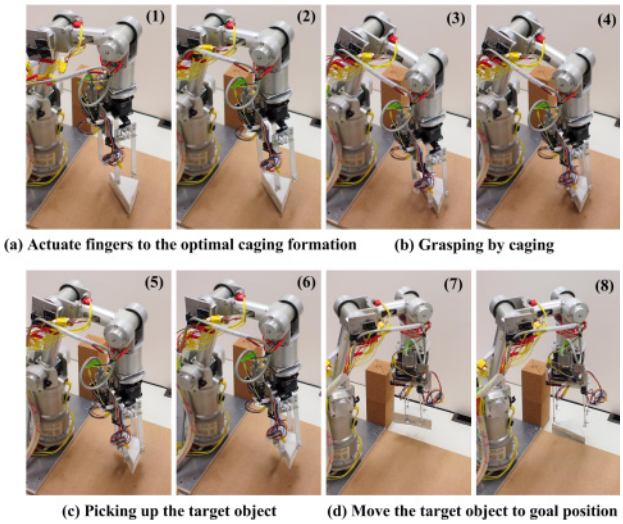


Fig. 18. Actuation of with the optimal caging configuration. The orientation and positions of those fingers in this figure is calculated according to step 1 to 4. The manipulator here cages, "grasp by caging" and moves target objects from initial position on the plate to goal position outside.

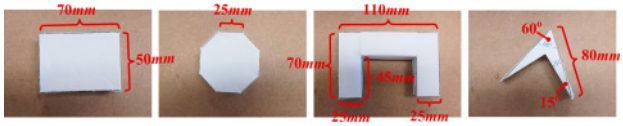


Fig. 19. The other four objects demonstrated in the video attachment.

gripping hand against uncertainty and we would like explore more in this direction in the future.

References

[1] M. T. Mason, *Mechanics of robotic manipulation*. The MIT Press, 2001.

[2] J. M. Trinkle, J. M. Abel, and R. P. Paul, "An investigation of frictionless enveloping grasping in the plane," *The International Journal of Robotics Research*, 1988.

[3] E. Rimon and A. Blake, "Caging planar bodies by one-parameter two-fingered gripping systems," *International Journal of Robotics Research*, 1999.

[4] T. Watanabe and T. Yoshikawa, "Grasping optimization using a required external force set," *IEEE Transactions on Automation Science and Engineering*, 2007.

[5] The SCHUNK JGZ industrial gripper. [Online]. Available: http://www.schunk.com/schunk_files/attachments/JGZ_160_EN.pdf

[6] The Barrett Hand. [Online]. Available: http://web.barrett.com/support/BarrettHand_Documentation/BH8-280_Datasheet.pdf

[7] L. B. Bridgwater, C. Ihrke, M. A. Diftler, M. E. Abdallah, N. A. Radford, J. M. Rogers, S. Yayathi, R. S. Askew, and D. M. Linn, "The robonaut 2 hand designed to do work with tools," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2012.

[8] T. Zhang and K. Goldberg, "Design of robot gripper jaws based on trapezoidal," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2001.

[9] A. M. Dollar and R. D. Howe, "The sdm hand : A highly adaptive compliant grasper for unstructured environments," *International Journal of Robotics Research*, 2010.

[10] F. L. Hammond, J. Weisz, A. A. de la Llera Kurth, P. K. Allen, and R. D. Howe, "Towards a design optimization method for reducing the mechanical complexity of underactuated robotic hands," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2012.

[11] A. M. Dollar and R. D. Howe, "Towards grasping in unstructured environments: Grasper compliance and configuration optimization," *Advanced Robotics*, 2005.

[12] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "Grasping by caging: A promising tool to deal with uncertainty," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2012.

[13] A. Rodriguez, M. T. Mason, and S. Ferry, "From caging to grasping," *The International Journal of Robotics Research*, 2012.

[14] J. Glover, D. Rus, and N. Roy, "Probabilistic models of object geometry with application to grasping," *The International Journal of Robotics Research*, 2009.

[15] M. Johnson-Roberson, J. Bohg, M. Bjorkman, and D. Kragic, "Scene representation and object grasping using active vision," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[16] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A. Y. Ng, "Learning to grasp novel objects using vision," in *Proceedings of International Symposium of Experimental Robotics*, 2006.

[17] B. Wang, L. Jiang, J. Li, H. Cai, and H. Liu, "Grasping unknown objects based on 3D model reconstruction," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005.

[18] A. Maldonado, U. Klank, and M. Beetz, "Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[19] C. Goldfeder, M. Ciocarlie, P. Peretzman, H. Dang, and P. K. Allen, "Data-driven grasping with partial sensor data," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[20] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, 2011.

[21] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2011.

[22] E. Rimon and J. W. Burdick, "Mobility of bodies in contact – I: A new 2nd order mobility index for multiple-finger grasps," *IEEE Transactions on Robotics and Automation*, 1998.

[23] A. F. van der Stappen, "Immobilization: Analysis, existance, and output-sensitive synthesis," in *Computer-Aided Design and Manufacturing*. AMS-DIMACS, 2005.

[24] R. Fukui, K. Kadowaki, Y. Niwa, W. Wan, M. Shimosaka, and T. Sato, "Design of distributed end-effectors for caging-specialized manipulator (Design concept and development of finger component)," in *Proceedings of International Symposium on Experimental Robotics*, 2012.

[25] G. A. S. Pereira, V. Kumar, and M. F. M. Campos, "Decentralized algorithms for multirobot manipulation via caging," *International Journal of Robotics Research*, 2004.

[26] Z. Wang, Y. Hirata, and K. Kosuge, "An algorithm for testing object caging condition by multiple mobile robots," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[27] A. Rodriguez and M. T. Mason, "Two finger caging: Squeezing and stretching," in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics*, 2008.

[28] Z. Wang, H. Matsumoto, Y. Hirata, and K. Kosuge, "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*, 2009.

[29] P. Pipattanasomporn and A. Sudsang, "Two-finger caging of nonconvex polytopes," *IEEE Transactions on Robotics*, 2011.

[30] M. Vahedi and A. F. van der Stappen, "On the complexity of the set of three-finger caging grasps of convex polygons," in *Proceedings of Robotics, Science and Systems*, 2009.

[31] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "On the caging region of a third finger with object boundary clouds and two given contact positions," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2012.

[32] ——, "A new "grasping by caging" solution by using eigen-shapes and space mapping," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2013.

[33] P. Leven and S. Hutchinso, "A framework for real-time path planning in changing environments," *The International Journal of Robotics Research*, 2002.