From Analysis to Practice: Three-finger Caging of Planar Convex Objects

Weiwei Wan, Rui Fukui, Masamichi Shimosaka and Tomomasa Sato

***Abstract*— This paper presents the development of planar caging manipulation. It involves a preliminary conclusion where the targets are limited to convex objects and the finger number is limited to three. Despite the popularity of form or force closure analysis, we prefer caging as it owns merits like requiring little dynamics, reducing kinematics and affording robust breaking margins to tolerate control errors. The analysis part of this paper theoretically discusses optimization procedures that best exploits the merits from caging. The practice part presents implementation details of our proposal in real work space by employing KINECT, a low-cost depth image capture produced by PrimeSense. Especially, some artifice and strategies are discussed and compared in this part to fulfill application requirements. Experimental results show the efficacy of our analysis and its promising future.**

I. Introduction

Traditionally, grasping and manipulation relate much to force analysis, namely, form closure and force closure [1][2]. Despite its popularity, force analysis requires pre-processing on the raw perception data to evaluate forces. Intrinsically, the pre-processing procedures, e.g. curve or polygon fitting [3], comprise an error source. Further, the problem grows more complicated as materials of target surfaces, target internal properties, or possible external force sets are taken into account [4][5]. For instance, reference [5] shows some results of grasping with respect to a given external force set and different friction coefficients. It costs too much to perform direct iteration that the paper employs a branch-and-bound approach to reduce complexity. The force-based closure theory is powerful, however they are vulnerable to perception or control errors, guarantee no robustness and lack potential in pragmatic applications.

The toughness of force analysis drive researchers to seek passive alternatives. Caging, in this case, is introduced. The aim of caging is to configure fingers properly so that target objects are constrained and may not escape into infinity. Unlike force-based closures, caging is sometimes named "object closure" as it does not take forces into account. The base of force analysis is "immobilization" while the base of caging is "constraints". In Configuration space (C space), force-based closure indicates a configuration point while caging indicates a compact configuration region.

Initially, reference [6] discusses the caging problem of one-parameter gripping system and proves that equilibrium grasps are potentially where caging breaks. References [7], [8] respectively propose regions of finger positions that guarantee caging. These regions lead to "breaking margins" of this paper. References [9][10] introduce caging to coordinate multiple robots in transportation. The forceless property of caging makes multiple robots easy to maneuver transportation targets as long as friction does not cause into jam. Reference [11] discussed our previous work in the application of caging on active systems. Although much attention has been paid to caging, it is far from well exploited since previous works are mainly devoted to caging test, or whether caging is formed and target is constrained. We propose to optimize grasping and manipulation towards the merits of caging robustness. In the best case when perception is exact, our approach generates a single C space point. When certain errors are encountered, our approach may still maintain caging manipulation robustly as targets are constrained in a compact region.

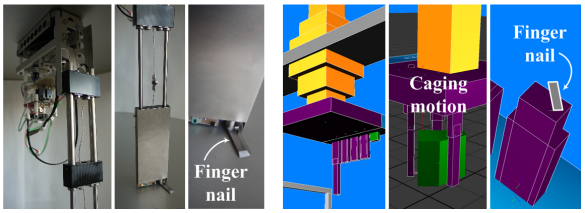

Fig. 1. Mechanical finger structure and its simulation model

We implement our approach by using both virtual simulation scanners and real depth cameras with concisely designed mechanism. Fig.1 demonstrates the mechanical structure of our manipulator. The left figure in Fig.1 is the mechanical hardware and the right figure shows its simulation in WEBOTS. Our manipulator is composed of a palm with several fingers hanging beneath it. On the tip of each finger, a rotational nail is installed to constrain target objects in the vertical direction. The finger tip is a key contribution to caging since it is not only a smart mechanical design but also a solution to the curse of dimensionality. Caging problems in real world 3D space are successfully projected to horizontal plane, say, 2D planar caging problems, owning to the finger tips. In this paper we focus on generating finger positions where robust caging can be obtained by configuring fingers onto them. Actuation of the fingers are not considered. Consequently, major challenges of this procedure are perception. Thanks to the optimization towards caging robustness, a low-cost PrimeSense KINECT can be enough for caging and demonstrates good performance in experiments.

Organization of this paper is as following. The analysis part is presented in Section II. Details such as background concepts and approximate alternatives are also introduced in this Section. Section III proposes practice of our analysis,

Weiwei Wan, Rui Fukui, Masamichi Shimosaka and Tomomasa Sato are with Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, JAPAN. wan@ics.t.u-tokyo.ac.jp

including some implementation artifice, experiments and comparison. Conclusions are drawn in Section IV, accompanied by discussions on future works.

II. Analysis of Optimal Caging

A. CC space and breaking margins

The configuration of fingers can be explicitly expressed as point positions[1]. Then we can intuitively employ $(x_0, y_0, x_1, y_1, x_2, y_2)$, where (x_i, y_i) denotes one finger position, or employ $(x_0, y_0, \theta_0, d_0, \theta_1, d_1)$, where (x_0, y_0) denotes an initial finger position and θ, d denote the relative positions of the other two fingers, to present these configurations. Consequently, all possible three-finger caging configurations comprise a 6-dimensional manifold and the most robust caging is the maximum corresponding to an optimizing function of this manifold. However, despite its intuition, the explicit 6-dimensional presentation costs much to perform searching and it is hard to find proper heuristics for robustness, which are essential to optimization.

Following the traces of popular works in caging [7][12][10], we propose to solve optimization problems in *CC* space. *CC* space is based on the concept of *C* space and means the configuration space of a configuration finger. Fig.2 demonstrates this idea.

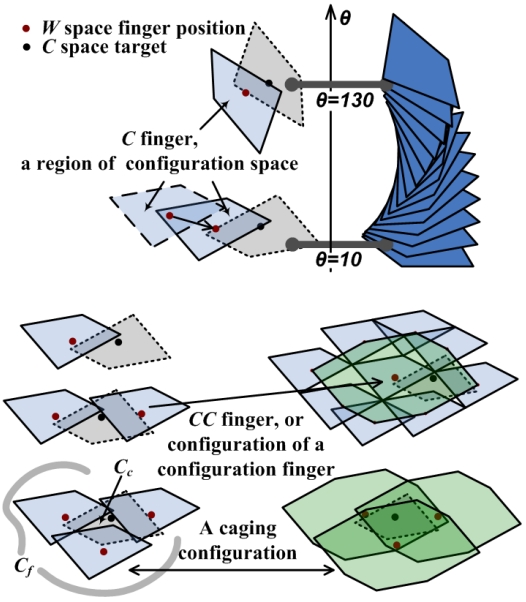

Fig. 2. *C* space, *CC* space and the caging configurations

C space is originally employed in motion planning and it is introduced into caging for easier analysis. In order to decide whether a finger position is obstructed by the target object, we change target into a configuration point, or *C* space target in Fig.2. Correspondingly, the point fingers, or *W* space finger positions in Fig. 2 are changed into *C* fingers. Then it is easier to check whether given point fingers are obstructed by targets or not. Note that the *C* space which corresponds to planar fingers is $\mathbb{R}^2 \times S$ space, meaning that *C* fingers rotate with respect to orientation of the target object.

Given three fingers and a target object whose θ is fixed, the correspondent *C* fingers can be obtained by employing the conversion procedure introduced in last paragraph. Whether the target object is caged can then be validated by checking if its correspondent *C* space target is in a compact free region embraced by *C* fingers. The bottom figure in the middle of Fig. 2 illustrates this idea.

Formally, these concepts can be expressed as following. Here, C_{obj} denotes the *C* space of a given target object, A_{obj} represents a *W* space object, A_i represents a *W* space finger (simplified into a position in this case), and q denotes a configuration of C_{obj}.

$$C_{oi} = \{q | q \in C_{obj} \wedge (A_{obj}(q) \cap A_i \neq \varnothing)\} \tag{1}$$

$$C_{free} = \{q | q \in C_{obj} \wedge q \notin \cup_{i=1}^{N} C_{oi}\} \tag{2}$$

Here C_{oi} is exactly the *C* finger of Fig.2 and C_{free} corresponds to the free space which is not obstructed by C_{oi}. Given a fixed orientation, C_{oi} and C_{free} are projected to a slice along the *S* dimension. The snapshots at θ=10 and θ=10 of Fig.2 corresponds to projected slices. Whether a target object is caged at a given slice, or whether a translational caging can be validated by checking if its correspondent *C* space configuration is in C_c, a subspace of C_{free} where

$$C_{free} = (C_c \cup C_f) \wedge (C_c \neq \emptyset) \wedge (C_f \cap C_c = \emptyset) \tag{3}$$

Nevertheless, the toughness of representing C_c blocks us from deeper exploration. *CC* space, in this case, is employed. Since C_c requires analyzing the relationship between fingers, one finger is considered against another and the configuration of a configuration finger is discussed. The basic idea is to take one *C* finger and generate its "C_{oi}" with respect to another *C* finger, which is taken as an obstacle. Note that fingers are assumed to be of the same geometrical property in this procedure.

Expression (4) formally illustrates *CC* fingers. Here, qq denotes a configuration of CC_{obj}.

$$CC_{oi} = \{qq | qq \in CC_{obj} \wedge (C_{oj}(qq) \cap C_{oi} \neq \varnothing)\} \tag{4}$$

By introducing *CC* space we can calculate C_c implicitly by checking whether each A_i is independent of the other fingers. The following lemma concludes this idea. This lemma corresponds to the bottom figure in the right of Fig.2. Details are shown in Fig.3. Note that $i-1$ or $i+1$ should be modulated with respect to three. Explicit modulation operation is hidden for conciseness.

Lemma 1: In the case of 3 finger convex caging, if each A_i, $i = 1,2,3$ is separated from the others by its adjacent ∂CC_{ok}, $k = i-1, i+1$ and target boundary ∂A_{obj}, the given target object is caged with respect to translational motions.

The solid red, green and blue segments in Fig.3 correspond to effective parts of ∂CC_{oi} while the dashed boundary of target object correspond to ∂A_{obj}. The finger positions in the right of Fig. 3 form a caging configuration as each finger

[1]Some readers may argue the advantages from finger shapes. However, fingers with shapes are easier compared with point fingers and we assume point fingers are ready for further extension.

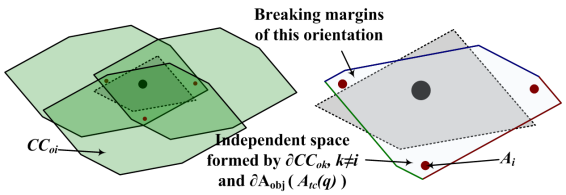

Fig. 3. Demonstration of Lemma1

is separated from the others by intersections of boundaries. Actually, it is both sufficient and necessary to translational caging when none of the fingers are redundant. We can draw this conclusion for the following reasons.

- Each C_{oi} overlays its adjacent $C_{o(i-1)}$ and $C_{o(i+1)}$.
- A_{obj} is inside the chain formed by C_{oi}, guaranteeing the existence of C_c.

By the way, Lemma.1 can be extended and applied to parallelograms where at least four fingers should be used for caging. This paper focuses on three-finger caging of convex objects, leaving four-finger problems to future works.

We define the areas that correspond to separated A_i the breaking margins of a fixed orientation, namely $A_{tc}(q)$. Here the subscript "tc" means translational caging. Please refer to Fig.3 for details. Since objects may rotate arbitrarily, the actual breaking margins that resist to both translational and rotational motions should be smaller than $A_{tc}(q)$. Details of actual breaking margins will be discussed in SectionII-B. Breaking margins are helpful to generating the most robust translational caging configurations. Given a finger position A_i and its correspondent breaking margin A^i_{tc}, robustness can be evaluated by estimating the distance from A^i_{tc} to the outer boundary of B_i. Formally, we would like to find the configuration A_1, A_2 and A_3 where the worst value of robustness evaluation reaches maximum. The following expressions illustrate this idea.

$$\underset{A_1,A_2,A_3}{argmax}\{\min_{i=1,2,3} \|A_i - \partial(CC_{o(i-1)} \cap CC_{o(i+1)})\|\} \tag{5}$$

Expression (5) shows optimization procedure that best resists to translational optimization. Complete optimization is more complicated as it requires the consideration of rotations. We will discuss complete optimization in the next part.

B. Complete Caging Optimization

Complete caging optimization means to perform maximum searching towards the actual breaking margins. Therefore, the major challenge of complete caging optimization is to decide the breaking margin with respect to two given adjacent fingers and arbitrary orientations. As an intuitive conclusion, the most robust caging configurations should be comprised of finger contacts on target boundary. Accordingly, we reduce the searching space into three-point combinations of boundary clouds, drastically improving efficiency.

Given two contacts on the boundary, breaking margin of the third finger could be accumulated by continuously

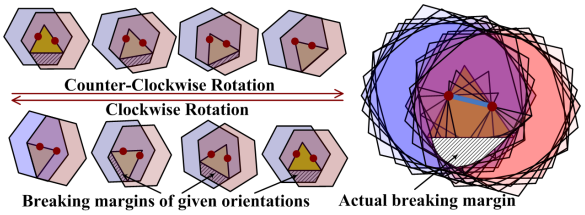

Fig. 4. The actual breaking margin of a triangle target with respect to two given finger contacts

changing target orientation. Fig.4 demonstrates this idea by using a triangle target for better visual rendering.

The two red contacts on triangle edge in Fig.4 denote the two adjacent finger positions. Continuous changing of target orientation is performed by counter-clockwise and clockwise rotation. Upper and lower figures in the left of Fig.4 correspond to this procedure. The shadow areas in each of the left eight figures demonstrate the breaking margins of its correspondent orientation.

Breaking margin at the initial orientation, $A_{tc}(q_{init})$, is a super set of actual breaking margin. Fig.5 illustrates this idea. When the third finger, which is rendered as a green point in Fig.5, is in a certain area of $A_{tc}(q_{init})$, target object may escape by hybrid translating and rotating motions. As has been proved by references [7][8], $A_{tc}(q)$ only guarantees that target object may not escape by translation. It is a larger area than actual breaking margin which provides protection not only against translational motions but also rotating ones. By continuous accumulating breaking margins at different orientations, the actual breaking margin can be obtained. The right figure of Fig.4 demonstrates the procedure of margin accumulation. Actual breaking margin of a third finger with respect to the two given red finger contact positions is rendered with shadow in the right of Fig.4.

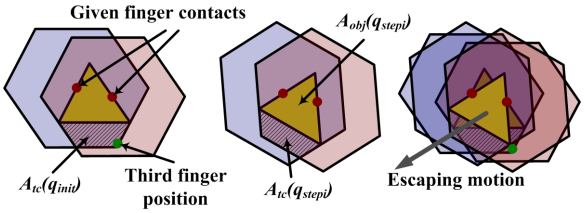

Fig. 5. $A_{tc}(q_{init})$ is a super set of actual breaking margin

Formally, the accumulation procedure is as following.

$$A_{pa}(q_{step_i}) \leftarrow A_{pa}(q_{step_{i-1}}) \cap (A_{tc}(q_{step_i}) \cup A_{obj}(q_{step_i})) \tag{6}$$

$$\begin{aligned} A_{da} &\leftarrow (A^+_{da} \cap A^-_{da}) \cup A^{'}_{da} \\ A^+_{da} &\leftarrow \cup^{n}_{i=1}(A_{obj}(q_{step_i}) \cap A_{pa}(q_{step_{i-1}})) \\ A^-_{da} &\leftarrow \cup^{-n}_{i=-1}(A_{obj}(q_{step_i}) \cap A_{pa}(q_{step_{i+1}})) \\ A^{'}_{da} &\leftarrow (\cap^{n}_{i=1} A_{pa}(q_{step_i})) \cap A_{pa}(q_{step_0}) \cap (\cap^{-n}_{i=-1} A_{pa}(q_{step_i})) \end{aligned} \tag{7}$$

Here, q_{step} denotes the continuously changing orientations, A_{pa} indicates the potential caging area. It is initially the same as $A_{tc}(q_{init}) = A_{tc}(q_0)$ while accumulates following the rule of expression (6) as target rotates. $A^{'}_{da}$ denotes a sufficient caging area that permanently exists. The determinate caging

area, A_{da}, is finally obtained through expression (7). Details of these expression and some experiments are written in a separate paper. Interested readers may refer to it for more information later.

Despite the completeness of this caging algorithm, it could cost at worst $O(n^2)$ where n relates to boundary rasterization, say, number of critical orientations or number of rasterized boundary points. If we perform maximum searching, expression (5), directly based on this complete algorithm, total cost might be as worse as $O(n^5)$. Therefore, it is necessary to seek approximate alternatives.

C. Approximate Caging Optimization

Orientation accumulation brought in dramatic cost of computational resource. We hope to exploit the power of breaking margins at fixed orientations and approximate determinate caging area by introducing certain constraints. We expect that it is possible to obtain satisfying results without orientation accumulation after introducing new constraints.

Reference [10] introduced bounded rotational angle from ρ-θ coordinates for caging test. ρ-θ curve is a function between the distance from rotation anchor to CC_{obj} boundary and CC_{obj} orientations. The major idea of bounded rotational angle is to ensure that the target object collides into fingers before rotating beyond the boundaries. Fig.6 shows the ρ-θ curve of a convex polygon. It is easy to find that largest bounded rotational angles appear at global minimum of the ρ-θ curve. Therefore, the finger positions that are most robust to rotation should coincide in ρ with the positions where global minimum appears and should be as far away from the curve as possible. Note that this constraint only considers resistance to rotation and guarantees no caging. Analytically, a robust caging should (a) form cage and own good translational margins (b) approximate global minimum of ρ value (c) locate at positions far from going beyond the ρ-θ curve. The three red points in the right of Fig.6 illustrate these requirements. ρ values of them approximate global minimum while their distance to the curve, for instance radius of the magenta circle, are relatively large.

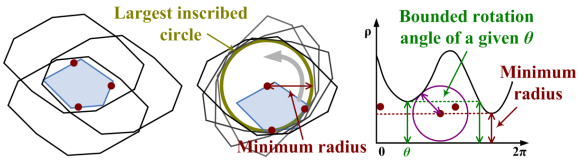

Fig. 6. ρ-θ curve of a convex polygon

As a conclusion, we need to (a) ensure caging is formed and (b) seek balance between translational and rotational constraints. Specifically, (b) involves two aspects as (b.1) largest translational margins and (b.2) global minimum approximation and large clearance from ρ-θ curve. Item (a) can be fulfilled according to Lemma.1. Whereas, item (b) is ambiguous and may bear various strategies. In this paper, we propose to seek the balance by selecting configurations with the least maximum finger distance from results of expression (5)[2]. Fig. 7 demonstrates this idea.

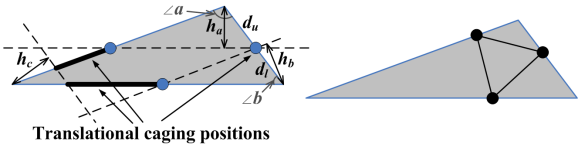

Fig. 7. Seeking the balance of translational and rotational constraints

The thick black region on edges of the triangle target shows all possible A_1, A_2 and A_3 combinations that own the same max value of expression (5). It can be obtained by intersecting CC_{oi}, CC_{oj} and CC_{ok} correspondingly. Since the target object in Fig.7 is a simple triangle, its equivalent max configuration regions can be generated by intersecting attitudes. The dash line in Fig. 7 left shows the intersection of attitudes where $h_a = h_b = h_c$. Further analysis shows that $\frac{d_u}{d_l}$ is indeed the ratio of $\frac{\angle b}{\angle a}$. The max configuration regions on edges of the triangle target are therefore the thick black regions formed by altitude intersections. Although max configuration regions of simple triangular objects own interesting property and relate to altitudes, they become complicated with general polygons and may not be drawn manually.

Our proposed balance-seeking strategy, or the least maximum finger distance, seeks the three-finger combination whose maximum inter-finger distance is smallest compared with the other combinations of the maximum A_{tc} boundary regions. Formally, we have the following expression where A_1, A_2 and A_3 are results of expression (8).

$$\underset{A_1,A_2,A_3}{argmax}\{min(\|A_1-A_2\|,\|A_2-A_3\|,\|A_3-A_1\|)\} \qquad (8)$$

Additionally, some research proposes solution to cover radius [13] as a preliminary result for sufficient caging. Cover radius is widely used in practical cages. For instance, distance between poles of a bird cage should always be smaller than a certain distance, or cover radius, to constrain a bird. The discussion goes complicated as our aim is actuation of fingers which might suffer from perception or control errors. Indeed, cover radius caging is non-optimal to errors. The middle figure of Fig.6 demonstrates this idea. Radius of the largest inscribed circle of CC_{obj} shares the same sense with cover radius. Cover radius caging requires (a) target object is caged and (b) the distance between point fingers should be smaller than cover radius, but there's no evaluation on caging breaking. Consequently, it is vulnerable to errors and caging may break easily with wrong finger positions. Actually, daily cages are powerful because they employs lots of redundant poles. These poles can be viewed

[2]Although this approximation obtains efficiency, we do not claim its completeness in caging. The making up procedure of least inter-finger distance is ill with certain shapes. For example, the three-finger configuration that corresponds to largest translational breaking margin of a semicircle may never be caged. However, our proposal is practical as it guarantees resistance to translation while ensures relative large rotational margin. Objects can always be constrained in actual environment, indicating its robustness.

as necessary fingers with non-point shapes which are beyond our discussion.

III. Practice, Experiments and Analysis

The practice part is based on approximate caging optimization. It presents and discusses implementation details in real applications, including comparison on employment of normals and how to deal with numerical problems.

KINECT, a product of PrimeSense, is employed as the perception device in our implementation. Only depth information is utilized to detect boundaries of targets. Target object boundaries in our implementation are obtained by repetitive perception and caging. The following items list the algorithm. By using this algorithm, we can extract boundaries of overlaying targets and pick them up iteratively.

(1) Pick up the peak pixel p_m from filtered depthmap.
(2) Expand p_m with a region growing technology.
(3) Extract boundary of grown region as target's boundary.
(4) Perform caging and move the target away from view.
(5) Capture a new depth map and go back to (1).

Unlike the other perception settings [14], we do not require exact installation of KINECT. The only limitation is that the perceived depth map should not deviate too much from top view. The robustness of our proposal makes the system possible to endure slight deviation and errors of perception. In experiments, we capture depth map by manually holding the KINECT camera. Despite its roughness, our algorithm can output qualified caging positions. On simple targets, the positions satisfy human expectation.

In reality, the perception devices shall be installed to a fixed position of the manipulator and afford better data. Our experiments with the manually hold perception device can output satisfying results, indicating a better performance on fixed manipulators.

Fig.8 shows some targets during our experiments and their correspondent depth map. Some of the target objects overlay each other while some are independent. Major targets of the sub-figures in Fig.8 in left-right and top-down order are mouse, bottle cap, plate, tape, tape box and paper box respectively.

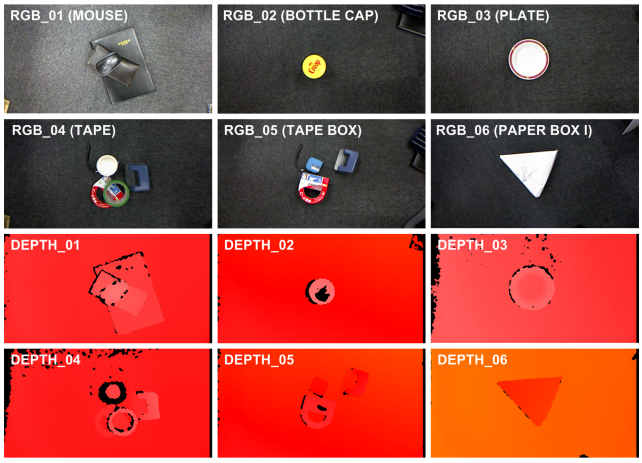

Fig. 8. Some targets and their correspondent depth map

Fig.9 shows the procedures employed in searching the most robust caging positions. Since the positions have been limited to contact boundary points, contact caging positions should form immobilizing grasping [8]. Therefore, we may alter Lemma. 1 into detecting whether surface normals at contact positions balance each other. Note that although the normal detecting alternative outperforms Lemma.1 in efficiency, we maintain the original version as (a) normals depend a lot on generating algorithms, (b) normal errors always exist due to flaws of perception devices and (c) simply using normals is not applicable to non-convex targets. In our implementation, we generate normals by calculating the expectation of two K-curve edges. The average value, $\frac{1}{4}(\hat{n}^1_{K_1} + \hat{n}^1_{K_2} + \hat{n}^2_{K_1} + \hat{n}^2_{K_2})$ is calculated as surface normal of a given boundary point. Here, $\hat{n}^i_{K_j}$ is the normal that corresponds to the ith edge of the jth K-curve. This strategy outperforms the others, such as PCA or general curve fitting, in its flexibility to perception errors. Nevertheless, it is not as effective at sharp corners. Actually, surface errors and surface fineness intrinsically conflict each other, and it is still an active research field of how to balance the trade-off. In our environment, keeping fineness or sharp corners is unnecessary as largest margins cannot appear with contact positions around them, see "critical orientations" in [7]. The criteria, expressions (5) and (8) not only brings robustness into caging, but also relaxes the evaluation of surface normals. The right two figures of Fig.9 respectively demonstrate surface normals generated from expectation of K-curve edges and the result of optimal searching based on contact positions that owns balancing normals.

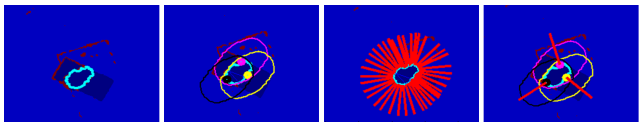

Fig. 9. Procedures and results in processing a mouse target

In the mouse example, there is little difference between our Lemma. 1-based caging detection (2nd figure of Fig.9) and the normal balancing alternative (4th figure of Fig.9), indicating the robustness of the normal alternative. Note that normals in these figures are drawn with red line segments and the normal balancing alternative is applicable to three-finger caging of convex objects only.

Table.I shows experimental results with the target objects listed in Fig.8. Each target owns two rows where the upper one denotes the results from surface normals and the lower one denotes the results from Lemma.1. The item "boundary" shows the number of boundary cloud points detected from our depth sensor. Values under this item are raw numbers without any processing. The "cost" column shows time efficiency of our implementation. Lemma.1-based cage detection acquires more resources than normal-based strategy. However, we do not impose preference on a certain one as normals may introduce new errors. The "closure" item indicates all finger configurations that form caging. It proves our claim that normal-based strategy is not as powerful

TABLE I

EXPERIMENTAL RESULTS WITH THE TARGETS LISTED IN FIG.8

Targets	Boundary	Cost	Closure	Exp.(5)	Exp.(8)
Mouse	373	7.19s	920	125	74.25
		87.72s	13244	125	74.25
Bottle cap	347	5.32s	652	106	68.00
		66.25s	10660	121	68.59
Plate	599	7.53s	765	369	116.77
		95.23s	11480	405	117.55
Tape	403	6.90s	744	125	77.99
		84.21s	12341	146	77.70
Tape box	465	6.76s	631	325	87.97
		75.79s	10660	370	90.52
Paper box	757	8.08s	702	4505	120.20
		112.50s	11480	4825	113.02

since it outputs fewer closure configurations. Columns "Exp. (5)" and "Exp. (8)" are the maximum values obtained in optimizing expression (5) and expression(8). Although the difference is minor, we may get more robustness by using Lemma.1.

Visual positions of the most robust finger configurations are shown in Fig. 10. Only the normal-based caging positions are demonstrated in this figure. Despite its efficiency, our implementation, say, averaging two K-curve edge normals, is specially designed for the application and may fail on some unexpected targets. Practitioners may make choice between normal-based strategy and Lemma. (1) according to their demands. Note that bottle cap is not shown in this figure as its shape and environment settings are similar to plate.

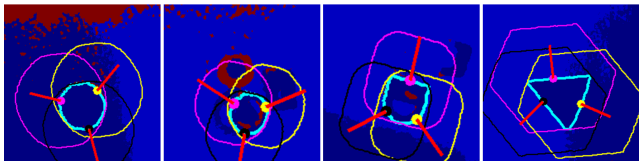

Fig. 10. Visual results of plate, tape, tape box and paper box in left-right order (Note that the last one suffers from numerical problems, see Fig.11 to fix the idea.)

As has been discussed in Section 2.3, the approximate optimization is to seek the balance between translational and rotational constraints. Specifically, expression (8) is optimized based on the results of expression (5). In practice, programs may suffer a lot of numerical problems and deviate from our expectation. Fig.11 demonstrates such an example. The paper box in Fig.11 is made according to Fig.7. The 2nd figure of Fig.11 shows the three-finger configurations by optimizing expression (8) based on the result of expression (5). The results, however, deviate from our analysis in Section 2.3 due to perception errors and numerical problems. One useful artifice in implementation is to select the maximum region of expression (5) but the exact maximum. The 3rd and 4th figures of Fig. 11 show the optimizing results based on maximum region where configurations that own breaking margins larger than $\frac{9}{10}$ and $\frac{4}{5}$ of maximum value, respectively, are included. This artifice depends on implementation. In the worst case, the results may degenerate into searching configurations that form equilateral triangles. Practitioner should choose region thresholds carefully according to their application environments.

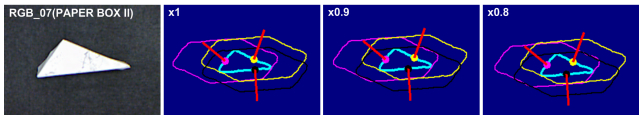

Fig. 11. The approximate robust positions by choosing different maximum region thresholds

By the way, besides selecting the optimum we may loosen the requirements and obtain a sub-optimal configuration set that plays as candidate goals in motion planning and reduce the limitation of kinematics. Interested readers may refer to reference [15] for more information.

IV. CONCLUSIONS AND FUTURE WORKS

In this paper, we first introduce robustness of caging and propose approaches to calculate the best caging configurations. Then, we implemented and evaluated it in real work space with KINECT. Our robust caging can work well even with a low-cost sensor, indicating the efficacy of the proposal and its promising future. In the best case, our approach generates a single C space point. With certain errors, we may still maintain caging manipulation robustly as targets are constrained in a compact region. Future works involve dealing with concave target objects and extending to four fingers. Finally, we would like to actuate the designed mechanism to retrieve daily utilities forcelessly, doing little harm to targets.

REFERENCES

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

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

[3] N. J. Redding, "Implicit polynomials, orthogonal distance regression, and the closest point on a curve," *PAMI*, 2000.

[4] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *Journal of Robotics and Automation*, 1988.

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

[6] E. Rimon and A. Blake, "Caging planar bodies by one-parameter two-fingered gripping systems," *IJRR*, 1999.

[7] J. Erickson, S. Thite, F. Rothganger, and J. P. Fellow, "Capturing a convex object with three discs," *Trans. on Robotics*, 2007.

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

[9] A. Sudsang, F. Rothganger, and J. Ponce, "Motion planning for disc-shaped robots pushing a polygonal object in the plane," *Trans. on Robotics and Automation*, 2002.

[10] Z. Wang, Y. Hirata, and K. Kosuge, "An algorithm for testing object caging condition by multiple mobile robots," in *IROS*, 2005.

[11] R. Fukui, T. Mori, and T. Sato, "Application of caging manipulation and compliant mechanism for a container case hand-over task," in *ICRA*, 2010.

[12] G. A. S. Pereira, V. Kumar, and M. F. M. Campos, "Decentralized algorithms for multirobot manipulation via caging," *IJRR*, 2004.

[13] P. Vongmasa and A. Sudsang, "Coverage diameters of polygons," in *IROS*, 2006.

[14] Z. Xue, A. Kasper, J. M. Zoellner, and R. Dillmann, "An automatic grasp planning system for service robots," in *ICRA*, 2009.

[15] R. Diankov, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation planning with caging grasps," in *International Conference on Humanoid Robots*, 2008.