Cooperative Manipulation with Least Number of Robots via Robust Caging

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

***Abstract*— One problem in multi-robot cooperative manipulation is redundancy. Too many robots are waste of hardware and increase control complexity. This paper solves the problem of redundancy by robust caging. Robust caging calculates caging positions from translational immobilization with respect to translational constraints and rotational constraints. On the one hand, robust caging helps to reduce the necessary number of robots in cooperation. On the other hand, the initial positions of necessary robots in robust caging are optimized to offer large robustness to control errors. Our proposal with robust caging is implemented to transport target objects over slopes. The algorithm can choose least number of robots with respect to shape of target objects and requirements of robustness. At the same time, each robot may endure as much as 256*ms* time step and 1*cm* control error, showing the superiority of robust caging.**

I. INTRODUCTION

Multi-robot cooperative manipulation is tightly coupled cooperation which aims to finish a complex manipulation task with several simple robots. Classical works in this realm relates to task allocation, application-specific control analysis and sensor fusion [1][2][3][4]. The delicate analysis and design, like prehensile manipulation in the realm of robotic manipulators, are usually target-specific or robot-specific. This makes it hard to implement general and intelligent systems. Many researchers introduce caging into this field to alleviate the toughness in force and timing design. Caging-based multi-robot manipulation employs several simple robots to transport a target object by enveloping it [5][6][7][8]. With caging, system designer can be saved from intensively designing force and control sequence of each robot. Formation control is enough to take over everything.

Although caging-based multi-robot manipulation is promising, researchers employ redundant robots to ensure successful caging. Take Fig.1(a) for example of redundancy. At each time step, many robots are alternatively wasted. Only part of the robots contribute to manipulation while most of the others are idle. We would like to decrease the number of idle robots as well as maintaining caging robustness as much as possible (see Fig.1(b)).

The problem of too many wasted robots may trace back to the study of caging. Caging aims to capture a target object so that it can be constrained in a certain area and may not escape into infinity. Much theoretical study has been devoted to caging, starting from one-parameter [9] caging to three-finger caging [10][11] and multi-finger caging [12]. Most of these

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

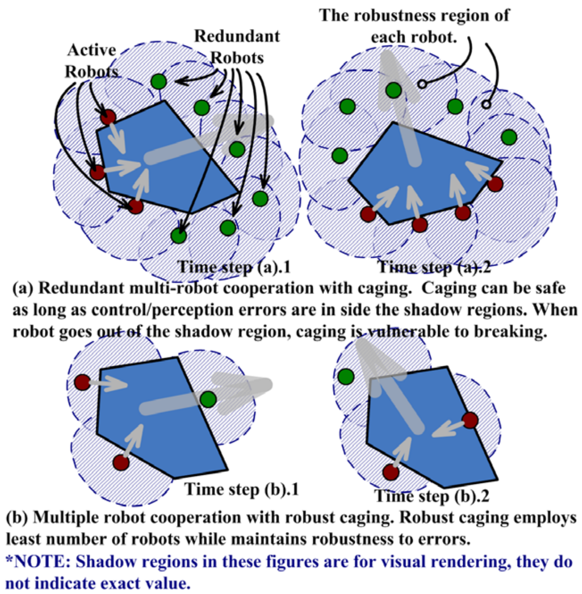

Fig. 1. Caging-based cooperative manipulation. (a) shows the manipulation with traditional caging tests. In this case, many robots may become idle at different time step. The green circles in (a) at time step (a).1 and time step (a).2 illustrate the wasted robots. (b) shows the manipulation with robust caging. Robust caging helps us to reduce the number of redundant robots while maintain caging robustness at different time steps. In this case, at most two robots may be in idle state at each time step.

works concentrate on whether caging is formed. Namely, they try to solve a caging test problem that given several fingers or positions of robots, whether target objects can be constrained in a certain region. Following this originality of caging, caging-based multi-robot cooperation aims to cage target objects rather than reduce robot number.

We in this paper would like to go deeper into decreasing the number of robots as much as possible. It is the same as finding the optimal caging with least robots and large robustness towards breaking. To solve this problem, we start from the minimum state of caging, which is “translational immobilization”, and take the optimal minimum state (optimal translational immobilizing positions) as initial robot positions. The optimal minimum state is calculated with respect to two kinds of constraints, namely the translational constraints and rotational constraints. Since in minimum state of caging the robot positions “translationally immobilize” target objects, they certainly “cage” them. The translational constraints and rotational constraints together optimize the “cage” with large robustness. Minimum state of caging offers us good initialization of “robust caging”. The robustness from robust caging makes our system satisfying to perform cooperative object manipulation by maintaining caging formations.

Tbl.I compares classical and caging-based multi-robot cooperative manipulation. Our proposal with robust caging seeks the advantages from either of them while tries to overcome the disadvantages. Thanks to robust caging, our proposal owns superiority in all of the following aspects

1. Least number of robots
2. Application-independent
3. Robust to locomotion quality
4. Implicit force control

Our proposal helps us to reduce the number of necessary robots as much as possible according to shape of target objects and requirement of user-defined robustness. It saves lots of resources. Readers are welcome to apply this technique to cooperative systems with limited agents.

TABLE I

COMPARISON OF CLASSICAL AND CAGING-BASED MULTI-ROBOT COOPERATION[1]

	Classical solutions	Caging solutions
Advantages	Small robot number	Application independent
	Precise manipulation	Low requirements on locomotion quality
		Implicit force control
Disadvantages	Application-specific	Large robot number
	High requirements on control quality	Low manipulation precision

[1]Our proposal with robust caging owns superiority in all red fields.

Organization of this paper is as following. Related works will be discussed in Section II. Section III discusses the caging problem in Configuration space (C space). Minimum caging and the least number of robots required for cooperative manipulation are presented in Section IV. Section V shows an overview of our proposal. Implementations and experiments are shown in Section VI, followed by conclusions and future works in the last section.

II. RELATED WORKS

The most related works to our proposal are [5][6][7][8][13] and [14]. Sudsang's work [13] shares nearly the same idea with robust caging in this paper. The other works are caging-based cooperative manipulation with redundant robots that falls into Fig.1(a). In this section, we would like to discuss them in detail.

Sudsang [13] performed some initial work in this field. He proposed to plan manipulations when jamming does not occur during task execution. His proposal, Inescapable Configuration Space (ICS), is actually the caging region. Some preliminary idea of this work could be found in [15]. Like Daniel [14], Sudsang tries to model caging with two assumptions. In the first place, each robot finally contacts with target edges as target objects move. In the second place, the contact edges should form a triangle. Note that like [10], [11] and [16] these two assumptions are reasonable when number of robots is three.

We are convincing in [13]'s ability in dealing with many target objects. Some implementations on symmetric objects or simple polygons could be found in Sudsang's publication. However, it could be too strict for certain polygons. For instance, three-robot cooperative manipulation may have larger ICS region along connecting edges in Fig.2 (see (b)) while Sudsang's explicit mathematical expressions may limit the actual result to a subset (see (a)). Moreover, when target object is a circle, his proposal may fail as caging does not fall into full contact with robots. Last but not least, the robots no longer lie along edges as total number increases. Larger number of robots go beyond the paper's discussion.

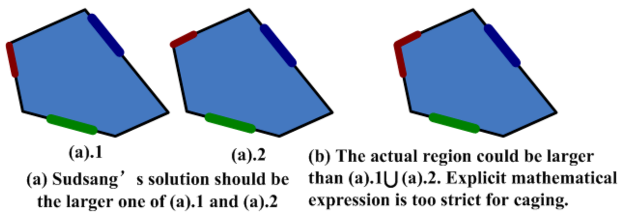

Fig. 2. Sudsang's explicit mathematical expressions could be too strict for the polygon in this figure. By constraining to three edges, the result could be the larger one of (a).1 and (a).2. It is a subset of actual caging robustness shown in (b).

Compared with Sudsang's work, our proposal does not perform explicit mathematical computation. We find the minimum state of caging/optimal translational immobilization with respect to both translational and rotational constraints. What's more, we do not limit robot number to three. When maximum translational breaking robustness (see $\tau_{\max}$ in Fig.7) is smaller than expectation of user, our system may refer to extra robots for help.

Besides the work with three mobile robots and explicit mathematical expressions, Pereira [5] and Wang [6] respectively employed the potential rotational region to check caging. Pereira proposed to consider the problem by exerting rotational constraints. However, the work did not give any rigid definition on the boundary of rotation. Wang smartly employs CC space objects and proposed to check whether the distance between point robots is smaller than the minimum of ρ-θ curve in span $\theta \in [\theta_{-}\text{-}\Delta\theta, \theta_{+}+\Delta\theta]$. Despite its smartness, the algorithm needs either (1) a predefined $\Delta\theta$ to limit calculation or (2) checking the minimum value in $\theta \in [0,2\pi)$ without $\Delta\theta$. In the first case, $\Delta\theta$ breaks the completeness of caging test while in the second case, the algorithm degenerates into a similar work as [12]. Conditions of the second case is too greedy and requires redundant robots.

Reference [7] and the recent work [8] both try to calculate the number of robots by $(2\pi r_{\mathsf{cage}})/(2r + D_{min}(\mathsf{obj}))$. It is actually the same as [5] and [6], and researchers need to tune parameter r to get minimum value. It cannot guarantee the minimum of robot number.

Our proposal is neither as strict as [13] nor as loose as [6]. We maintain that the proposal is promising in cooperative manipulation with least number of robots as well as high robustness towards various noises. Previous caging cooperation and our work are compared in Tbl.II for better comprehension.

TABLE II

COMPARISON OF CAGING-BASED MANIPULATION

	Category I[1]	Category II[1]	Our work
Agent Number	3	Redundancy	Least number
Assumption	Contact/Triangle	Sufficient agents	3-4 agents [2]
Robustness [3]	$\tau_{(1)} < \tau_{(2)}$	$\tau_{(1)} < \tau_{(2)}$ $\tau < \tau_{(2)}$	$\tau_{(1)} \leq \tau_{(3)}$

[1]Category I includes [13] and [14] ([10] and [11] could be loosely classified into this category, too). Category II includes [5], [6], [7], [8] and [12].
[2]Our proposal can cage an object with 3-4 agents according to its shape. See following sections and our previous work [17] for details.
[3]$\tau_{(1)}$ and $\tau_{(2)}$ indicate the robustness of first and second categories while $\tau_{(3)}$ denotes the robustness of our work.

III. PRELIMINARIES

Caging is a pure geometric problem where researchers would like to constrain an object in a certain area so that it may not move into infinity. In C space of target object (C^{obj}) [6][10], whether the target object is caged can be validated by checking if its correspondent C space configuration is in a compact free region enclosed by C fingers. Formally, these concepts can be expressed by

$$\mathcal{R}_i = \{\boldsymbol{q} | \boldsymbol{q} \in C^{\text{obj}} \wedge (\mathcal{W}_{\text{obj}}[\boldsymbol{q}] \cap r_i \neq \varnothing)\} \quad (1)$$

$$C_{\text{free}} = \{\boldsymbol{q} | \boldsymbol{q} \in C^{\text{obj}} \wedge \boldsymbol{q} \notin \bigcup_{i=1}^{n} \mathcal{R}_i\} \quad (2)$$

$$(C_{\text{free}} = (C_{\text{fc}} \cup C_{\text{ff}})) \wedge (C_{\text{fc}} \neq \emptyset) \wedge (C_{\text{ff}} \cap C_{\text{fc}} = \emptyset) \wedge (\boldsymbol{q}_{\text{init}} \in C_{\text{fc}}) \quad (3)$$

Notations in these equations are as following.

C^{obj}: The C space of a given target object.

$\mathcal{W}_{\text{obj}}$: The Work space ($\mathcal{W}$ space) object. It actually indicates the target object.

r_i: A $\mathcal{W}$ space point mobile robot.

$\boldsymbol{q}$: $\boldsymbol{q} = \{q_x,\ q_y,\ q_\theta\}$, a configuration in C_{obj}. $\{q_x,\ q_y\}$ denotes the position of $\mathcal{W}_{\text{obj}}$ while q_θ denotes its orientation. $\mathcal{W}_{\text{obj}}[\boldsymbol{q}]$ represents $\mathcal{W}_{\text{obj}}$ at a configuration $\boldsymbol{q}$.

$\boldsymbol{q}_{\text{init}}$: The initial configuration of target object. $\mathcal{W}_{\text{obj}}[textbfq_{\text{init}}]$ represents the target object before caging.

$\mathcal{R}_i$: The C obstacle which corresponds to r_i.

C_{free}: The C space which is not obstructed by any $\mathcal{R}_i$.

Testing whether a target object is caged can be validated by checking if its correspondent initial C space configuration $\boldsymbol{q}_{\text{init}}$ is in C_{fc}, an isolated subspace of C_{free}. Expression (3) corresponds to this caging test idea in C^{obj}. Caging is attained when expression (3) is *true*.

C^{obj} is a $\mathbb{R}^2 \times \mathbf{S}^1$ space for planar manipulation. In one extreme case, C_{fc} may degenerates into an isolated point (this point is equal to $\boldsymbol{q}_{\text{init}}$) or an isolated periodic segment along $\mathbf{S}^1$ of C^{obj}. We name this extreme case the minimum state of caging (see Fig.3). The minimum state of caging is either pure immobilization or stops target objects from translational movement. Therefore, minimum state of caging can also be viewed as "translational immobilization".

Fig.3 illustrates the three-dimensional C^{obj} of the target object in Fig.1. Changes of C_{fc} as $\mathcal{R}_i$ squeezes are shown in Fig.3(a)→Fig.3(b)→Fig.3(c) order. In this case, the final minimum state of caging is a single point which implies pure immobilization.

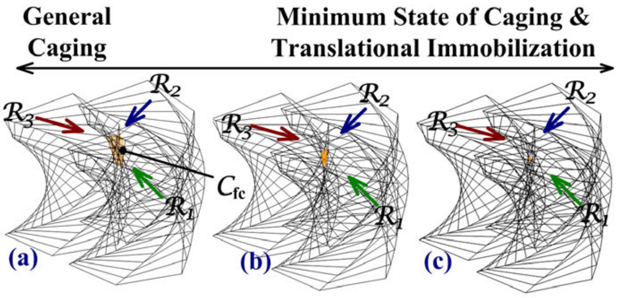

Fig. 3. As $\mathcal{R}_i$ moves along the colored arrows, caging may finally degenerates into "translational immobilization", namely pure immobilization or immobilization that stops target object from translational movement. In the case illustrated in this figure, C_{fc} degenerates into an isolated point in C^{obj}. It is pure immobilization.

IV. ROBUST CAGING

Robust caging calculates caging positions from minimum state of caging with respect to translational constraints and rotational constraints. It is actually the translational immobilization with some constraints to guarantee large robustness against caging breaking.

Firstly, let us discuss about the minimum state of caging and the least number of robots in caging a planar target. Since the minimum state of caging is translational immobilization, the least robot number to cage an object is the least robot number that can immobilize it or can stop it from displacement in position. Without friction, we need at least $n_{\text{dof}} + 1$ robots for form closure while $n_{\text{dim}}+1$ to $2n_{\text{dim}}$ robots for immobilization or caging. Here, we assume point robots, convex polygons and zero friction (see [18] for non-point fingers). n_{dof} denotes the degree of freedom while n_{dim} denotes the dimension of $\mathcal{W}$ space.

Therefore, in applications of cooperative planar manipulation, at least 2+1=3 (e.g. polygons with three edges forming a triangle) to 2×2=4 (e.g. rectangles or semicircles) point mobile robots are required to cage any polygon. More specific provements could be found in our previous work [17].

Then, let us consider about the optimal caging, namely the caging with large robustness to caging breaking. Following the idea that minimum state of caging is translational immobilization, we start from all the translational immobilizing multi-robot combinations and pick out the optimal one with respect to certain constraints.

Intuitively, the constraints should be intersections of adjacent $\mathcal{R}_i$. Fig.4 demonstrates this idea. These intersections imply the robustness of caging breaking. A smaller intersection is more sensitive to breaking as control errors may vanish small intersections easily and connect C_{fc} to C_{ff}.

However, it is hard to rigidly express the robustness of caging breaking through intersections as (1) the intersections are decided by relative positions of $\mathcal{R}_i$ and (2) the robustness of breaking is the smallest movement for the disappearance of these intersections.

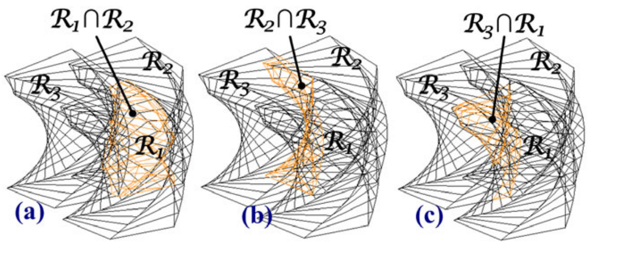

Fig. 4. The orange mesh demonstrates the intersections of adjacent $\mathcal{R}_i$. Minimum caging is the translational immobilization state with respect to these intersections. The intersections, indeed the smallest movement of $\mathcal{R}_i$ for disappearance of any intersection, imply robustness of caging breaking.

Our solution is to decompose the constraints in an engineering way into translational constraints and rotational constraints.

Take a slice at q_θ along $\mathbf{S}^1$ of C^{obj} for example. At a certain slice q_θ, the intersections of adjacent $\mathcal{R}_i$ are shown in Fig.5. The smallest movement for the disappearance of the intersections at q_θ is denoted as a purple segment in Fig.5(b). The purple segment represents translational constraints. Translational constraints denote the robustness to translational caging breaking where motions of target objects are limited to movements along x and y axis without rotation (namely when $\mathbf{S}^1$ is fixed at a certain q_θ).

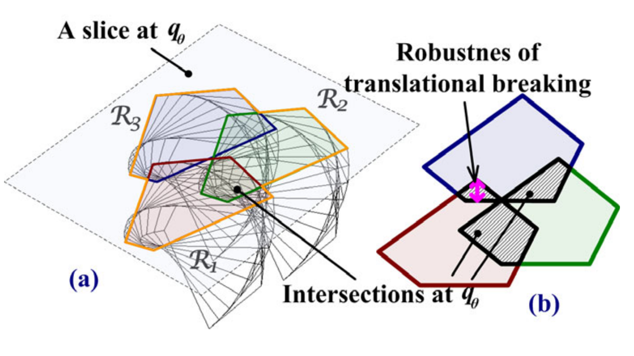

Fig. 5. The intersections of adjacent $\mathcal{R}_i$ at a certain slice q_θ. When motions of target objects are limited to translation, the purple segment in (b) indicates the robustness of breaking, namely the translational breaking.

When target objects rotate, rotational constraints take effect. The rotational constraints are complementary to translational constraints. These two constraints collaborate together to approximate optimal minimum caging state. In the first place, we filter out a set of candidate multi-robot positions by the length of purple segments with translational constraints. Then, the positions with smallest scatter (rotational constraints) are chosen as the optimal result. We choose the smallest scatter as the second screening step since robot positions with smaller scatter tend to add stronger constraints to rotation when there's no a priori information. Formally, this procedure is expressed as following.

$$\mathcal{T}_\tau = \{R | e^t_R > \tau\} \tag{4}$$

$$e^r_R = \min(E(|r_{i_u} - r_{i_v}|)S(|r_{i_u} - r_{i_v}|)),\ \ r_{i_u}, r_{i_v} \in R,\ \ R \in \mathcal{T}_\tau \tag{5}$$

Notations in these two expressions are as following.

- $\mathcal{T}_\tau$: The set of candidate multi-robot formations filtered by value τ. The purple segment length (see Fig.5) of any element in this set is smaller than τ.
- R: A vector denoting the positions of each robots. It represents the multi-robot formation.
- e^t_R: The length of purple segments in Fig.5. It denotes the translational (superfix t indicates translation) breaking robustness of R.
- e^r_R: The smallest scatter of a given R. Similar as t in e^t_R, the superfix r indicates rotation.
- r_{i_u}: The position of one robot.
- $E()S()$: The method to calculate scatter of a given R. It is to multiply the expectation $E()$ and variance $S()$ of the distance between adjacent robot positions.

The parameter τ plays an important role in collaborating translational constraints and rotational constraints.

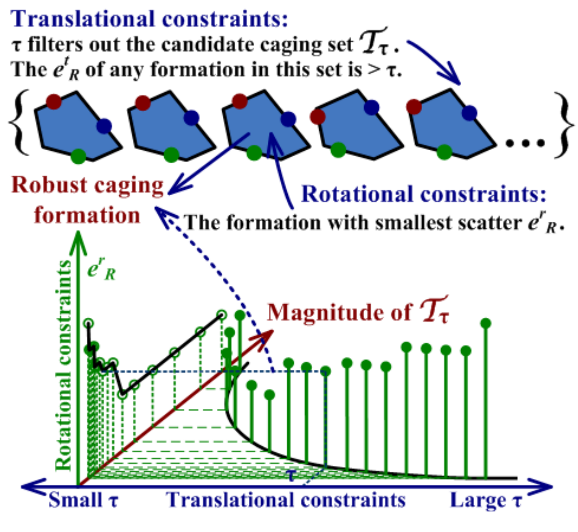

Fig. 6. The roles of translational constraints and rotational constraints in obtaining minimum caging R. τ plays an important role in collaborating the two constraints. In the first step, τ filters out a set $\mathcal{T}_\tau$. The magnitude of this set varies as τ changes (see the horizontal black curve). In the second step, rotational constraints filters out the element with minimum scatter e^r_R as robust formation (see the green segment with blue dash).

Take Fig.6 for example. In one extreme case where $\tau \to 0$, the translational constraints become invalid. All immobilizing combinations pass through the translational constraints. They are directly tested against rotational constraints. The result degenerates into an immobilization grasp with least inter-finger distance and variance. In the other extreme where $\tau \rightarrow \tau_{\max}$, $\mathcal{T}_\tau$ only holds one element so that it makes the rotational constraints invalid. Here, $\tau_{\max}$ denotes the maximum translational breaking robustness of all immobilizing formations.

Either the two extremes produces sub-optimal results and τ should be chosen carefully. The translational constraints filter out some multi-robot formations $\mathcal{T}_\tau$ from immobilizing combinations on target boundary while the rotational constraints choose the formation with smallest e_R^r as optimal result. The optimal result is the R for robust caging.

In the experimental section, we will see that robust caging ensures satisfying tolerance to control errors. They make cooperative manipulation with least number of robots and large robustness possible.

V. OVERVIEW AND FORMATION CONTROL

In this section, we are going to integrate the fragments of previous sections and discuss manipulation with formation control.

A. Overview

Robust caging helps us to find the positions of multiple robots with satisfying robustness to breaking. Besides the core robust caging algorithm, we need the shape of target object as input value and need to pre-compute $\tau_{\max}$ for better choice of τ, see Fig.7 for details.

Core robust caging algorithm in this flow chart is emphasized with a bold dash box while the other external components are denoted by solid boxed named external (a), external (b) and external (c) respectively.

The external (a) component decides how many robots should be employed at least to manipulate a target object taking account of target shape and requirements of applications. On the one hand, as has been analyzed in Section IV, the number could be 3 or 4 according to target shapes (see the "Exist" switch in external (a) of Fig.7). On the other hand, our proposal compares the maximum translational breaking margin $\tau_{\max}$ with a user-defined robustness for flexibility (see the "Is $\tau_{\max}$ large enough" switch in Fig.7). The external (a) component may refer to more robots according to user-defined requirements of robustness.

The external (b) component employs $\tau_{\max}$ to decide the choice of τ. In our implementation, τ is chosen as a proportion of $\tau_{\max}$. The external (c) component perform fine tuning of R to avoid jam. Rigid employment of R may cause robots to collide with target object. External (c) adds some disturbance to R to guarantee the safety of caging initialization. After caging initialization, target objects could be manipulated by mobile robots with a $R^{'}$ formation.

B. Formation control

Maintainence of $R^{'}$ falls into the research field of formation control. Major topic of traditional formation control can be viewed as maintaining the distance between adjacent robots. Nevertheless, formation control in cooperative manipulation requires more stuff. For example, since each robot is controlled independently, there could be errors in relative positions that cause into jam, namely robots may squash target objects at a certain time step. Therefore, researchers usually defined certain rules to avoid jam [6][15].

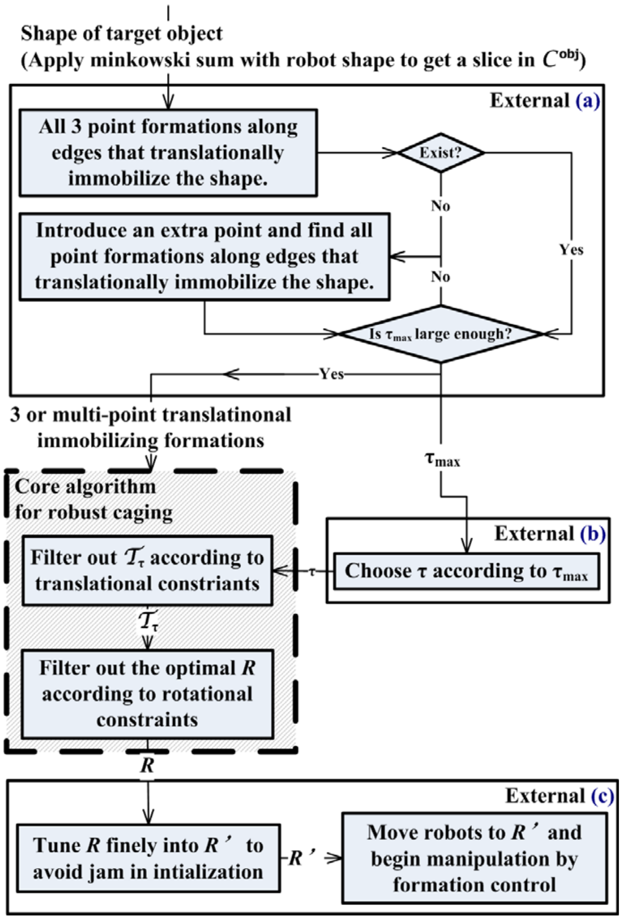

Fig. 7. The overview of our proposal. It is composed of the core robust caging algorithm in a bold dash box with external (a), (b) and (c) algorithms in solid boxes.

Our proposal does not need to follow rules owing to high robustness from robust caging. In the formation control procedure, a leader robot is chosen. The other robots follow the leader as well as maintain $R^{'}$. Note that this implementation may not guarantee maximum safety since we do not want to incorporate explicit specification of motion orders, directions and leaders. However, it could offer satisfying safety to jam owing to optimal caging robustness.

Fig.8 demonstrates the idea of our formation control. Motion of each robot is decided by decentralized planners (such as paths produced by potential field planner between initial positions and goal positions in Fig.8) and they could be along any direction. If the motion of robots at one time step are too drastic, jam may appear. We expect robust caging could avoid jam and endure certain formation deviation appeared in the movement of each single robot since it offers satisfying robustness to caging breaking and enough

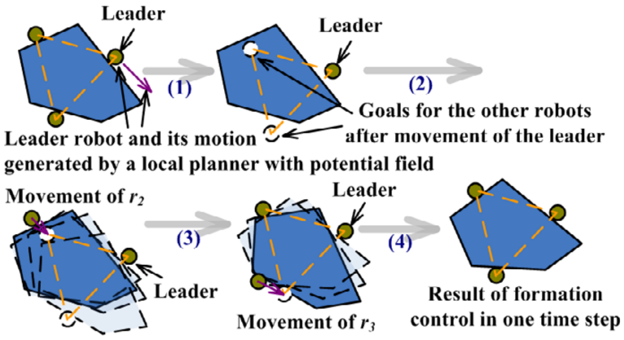

Fig. 8. Formation control strategy of our proposal. This strategy does not employ any pre-defined rules and it is not of ultimate safety towards jam. However, it could endure certain deviation appeared in the movements of each single robot owing to robust caging and fine tuning. Readers may refer to the experimental section for its tolerance to errors.

tolerance to control errors. Tolerance to motion at one time step will be discussed in the experimental section. Of course in real applications, practitioners could attain better performance by defining motion orders, directions and specific leaders.

VI. EXPERIMENTS AND ANALYSIS

Our experiments are performed with WEBOTS simulation environment. The Open Dynamic Engine (ODE) embedded in WEBOTS offers us a powerful tool to test cooperative manipulation with least number of robots and minimum caging.

A. Environment setup

Fig.9 shows the scene of our experiments. The task defined in this scene is to move the target object cooperatively from initial position to goal region at the other side of the slope. The inclination of slope at either side is $0.2rad$. Friction coefficients of target objects are set to 0, indicating that target objects may move freely in the cage formed by robots. The free motion from 0 coefficient brings ultimate challenge and ensures exhaustive tests against caging. Each robot is run as an independent process so that they result into random errors like mutiple robots in real world.

Four different kinds of target objects are employed in our experiments. Their dimensions are shown in Fig.9(a), Fig.9(b), Fig.9(c) and Fig.9(d).

Our core algorithm for robust caging works with the Minkowski sum of target shape and robot dimension. The target shape is its projection on ground. In that case, the ball object suffers more from breaking. Height of mobile robots is 0.28 and it is lower than boundary of ball projection (0.5). Therefore, the fine tuning component in external (c) of Fig.7 should be smaller on the ball object while larger on the other target objects. Empirically, we set the fine tuning of target (a), (b), (c) and (d) with $0.1\tau_{\max}$, $0.1\tau_{\max}$, $0.01\tau_{\max}$ and $0.1\tau_{\max}$ expansion.

In implementation, τ is set to $0.7\tau_{\max}$. We do not discuss the performance of different τ in this paper. Interested readers may refer to our previous work which concentrates on the

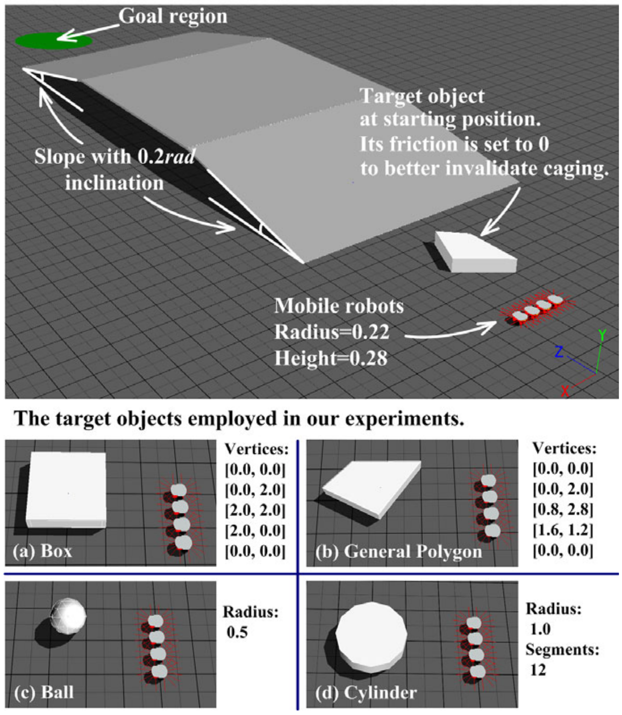

Fig. 9. Parameter settings and various objects employed in our experimental environment. Readers may compare robot size and object size easily by referring to each figure in the lower part.

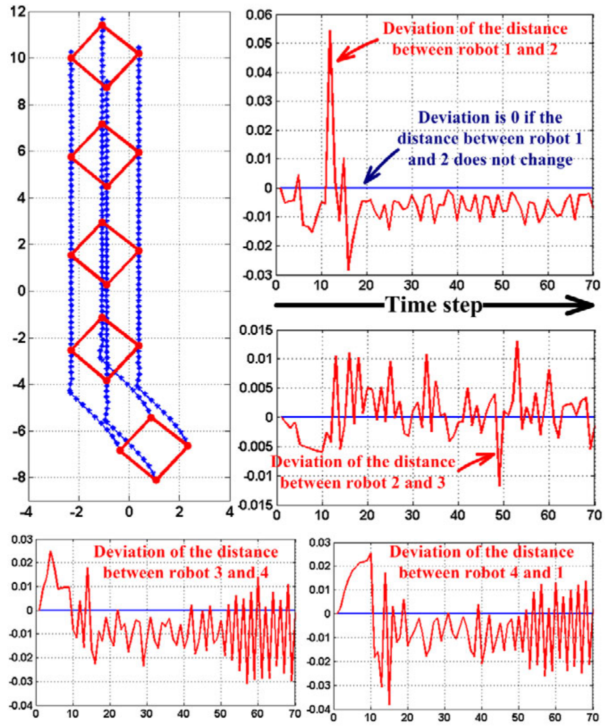

Fig. 10. This figure shows the errors of formation control in manipulating object (a). The curves in the upper-left figure shows the formation during cooperative manipulation and the curves in the other figures shows the variation of of inter-robot distance during manipulation.

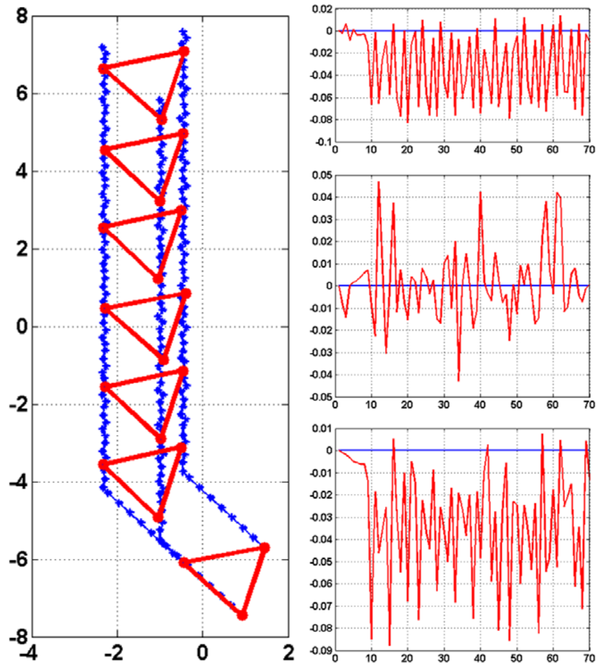

Fig. 11. This figure shows the errors of formation control in manipulating object (b). It follows the same form as Fig.10.

choice of τ [17]. Fig.10 and Fig.11 respectively shows the tolerance of robust caging to errors of formation control in manipulating object (a) and object (b). Similar as our expectation, at least four robots are needed for object (a) while at least three robots are required for object (b).

The left figures in Fig.10 and Fig.11 show the formation formed by robots in manipulation. The other figures show the variation in inter-robot distance during manipulation. Initial inter-robot distance is tagged with blue straight lines in these figures. These figures are taken from the first 70 time steps (see the horizontal axis). Although the distance varies dramatically (see the vertical axis), our robust caging manipulation can offer great tolerance and perform robustly. The length of a single time step is set to $64 milisecond$, namely the strategy shown in Fig.8 are performed every $64 milisecond$.

The curves in Fig.12 show the error in formation control with different time steps. As the length of one time step increases, the error increases accordingly. Deviation of formation depends on the length of one time step. Although larger deviation errors appear as the length of time step increases, our robust caging algorithm can endure them. With objects (a) and objects (b)), our caging algorithm can be robust to any of the three different time steps shown in Fig.12. The caging formation from our proposal can not only reduce robot number as much as possible but also offer satisfying robustness.

Object (c) is a special case as its size is relatively small compared with robots and different from its projection on the ground. In our experiments, only formation control of every $64 miliseconds$ could guarantee successful manipulation. Actually, circular objects with small dimension could be most challenging to formation control and could be vulnerable to caging breaking. After changing the sphere into a cylinder with larger size (Radius = 1, see object (d)), the robots can perform successful manipulation at either $64 miliseconds$ or $128 miliseconds$ intervals. Fig.13 shows the errors of formation control in these two successful cases.

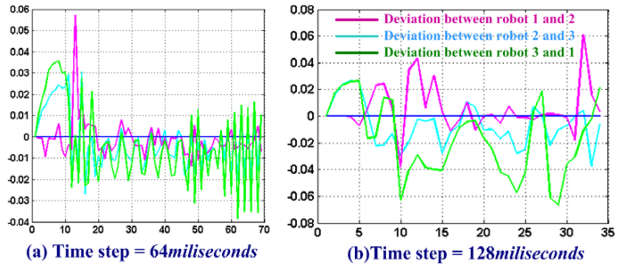

Fig. 13. Cooperative manipulation of the cylinder object (d) (Radius = 1) is successful with both time steps in this figure. Namely, our caging algorithm can endure the formation control error as much as 0.06 for this object.

By the way, our algorithm may suggest more robots to obtain larger tolerance to formation control errors. As has been discussed, the "Is $\tau_{\max}$ large enough?" switch in external (a) of Fig.7 corresponds to this idea. By adding redundant robots like previous works [6][7], both manipulation of the ball and cylinder may endure formation control errors of $256 miliseconds$. Fig.14, which compares the results of 3-robot and 4-robot manipulation of object (c), demonstrates the idea.

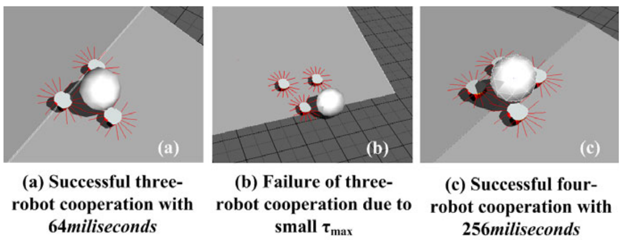

Fig. 14. The algorithm may suggest more robots for larger tolerance to formation control errors. By adding a redundant robot, manipulation of the ball object with $256 miliseconds$ becomes safer (see (c)).

In the end, we show certain frames of 3-robot manipulation with object (b) in Fig.15 for an overview of a whole experimental procedure. Readers may refer to the attached video clip for details.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose to perform multi-robot cooperative manipulation with both (1) least number of robots and (2) large robustness to control errors. The proposal is realized with robust caging and we carried out various experiments to validate its performance. Experiments show that our proposal can satisfactorily fulfill our expectation in multi-robot cooperative manipulation. Robust caging is a promising tool in reducing agent numbers and maximizing error tolerance. In the future, we would like to explore more applications of this tool.

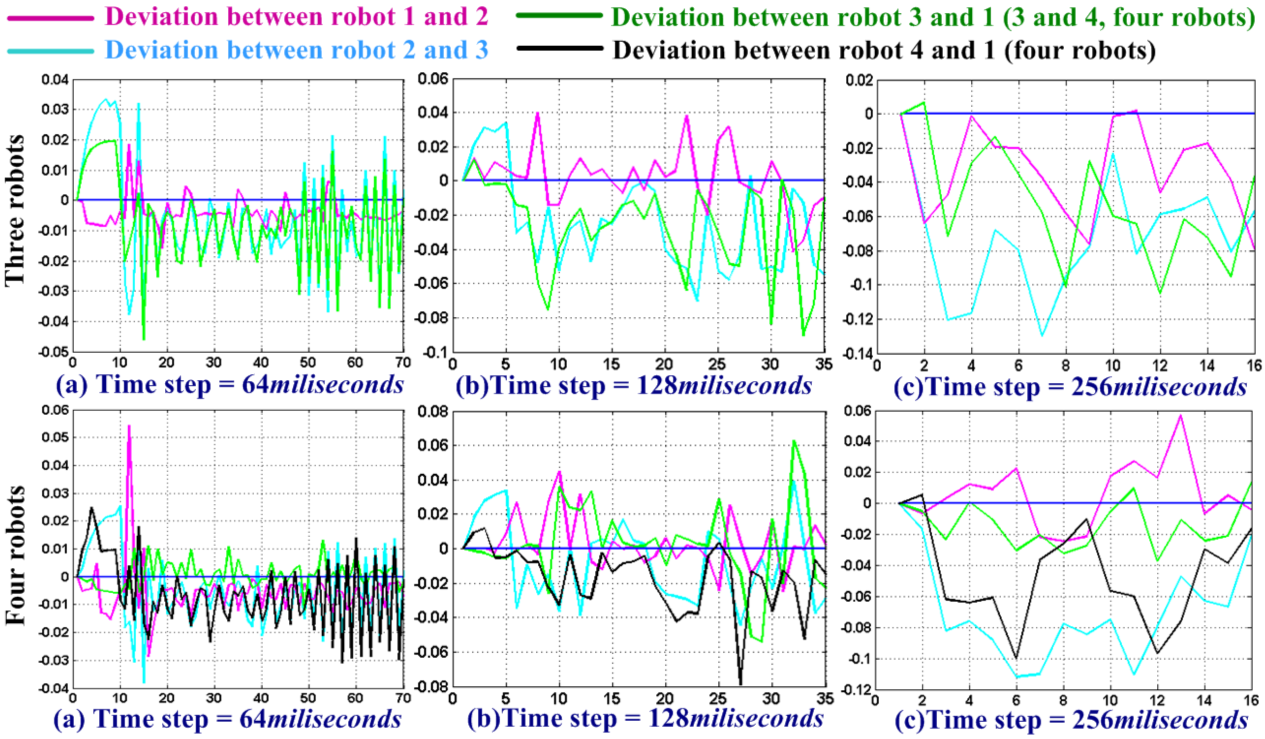

Fig. 12. The curves in these figures illustrate the variation of inter-robot distance during manipulation with respect to different time step (Each figure is the overlay of all employed robots). Our caging proposal succeeds with any of these errors. Even with a loose formation control (reorganize the formation at every 256*miliseconds*) and relative large control error (more than 0.1), the robots can manipulate object (b) and object (c) robustly.

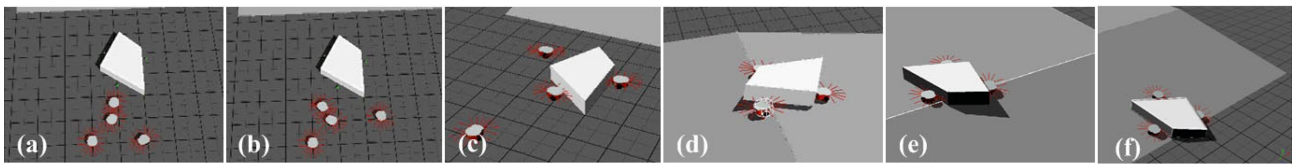

Fig. 15. Some frames of 3-robot manipulation with object (b). It involves (1) motion planning and loose cooperation in figure (a), (b) and (c) and (2) cooperative object manipulation with robust caging in figure (d), (e) and (f).

References

[1] L. E. Parker, “Alliance: An architecture for fault tolerant multi-robot cooperation,” *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 220–240, 1998.

[2] B. P. Gerkey and M. J. Mataric, “Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination,” in *Proc. of ICRA*, pp. 464–469, 2002.

[3] G. Montemayor and J. T. Wen, “Decentralized collaborative load transport by multiple robots,” in *Proc. of ICRA*, pp. 372–377, 2005.

[4] P. Cheng, J. Fink, and V. Kumar, “Abstractions and algorithms for cooperative multiple robot planar manipulation,” in *Proceedings of Robotics, Science and Systems*, pp. 143–150, 2008.

[5] G. A. S. Pereira, V. Kumar, and M. F. M. Campos, “Decentralized algorithms for multirobot manipulation via caging,” *International Journal of Robotics Research*, vol. 23, pp. 783–795, 2004.

[6] Z. Wang, Y. Hirata, and K. Kosuge, “An algorithm for testing object caging condition by multiple mobile robots,” in *Prof. of IROS*, pp. 3022–3027, 2005.

[7] J. Fink, M. A. Hsieh, and V. Kumar, “Multi-robot manipulation via caging in environments with obstacles,” in *Proc. of ICRA*, pp. 1471–1476, 2008.

[8] D. J. Cappelleri, M. Fatovic, and U. Shah, “Caging micromanipulation for automated microassembly,” in *Proc. of ICRA*, pp. 3145–3150, 2011.

[9] E. Rimon and A. Blake, “Caging planar bodies by one-parameter two-fingered gripping systems,” *International Journal of Robotics Research*, vol. 18, pp. 299–318, 1999.

[10] J. Erickson, S. Thite, F. Rothganger, and J. P. Fellow, “Capturing a convex object with three discs,” *IEEE Transactions on Robotics*, vol. 23, pp. 1133–1140, 2007.

[11] M. Vahedi and A. F. van der Stappen, “Caging convex polygons with three fingers,” in *Prof. of IROS*, pp. 1777–1783, 2008.

[12] P. Vongmasa and A. Sudsang, “Coverage diameters of polygons,” in *Prof. of IROS*, pp. 4036–4041, 2006.

[13] A. Sudsang, F. Rothganger, and J. Ponce, “Motion planning for disc-shaped robots pushing a polygonal object in the plane,” *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 550–562, 2002.

[14] D. Rus, “Coordinated manipulation of objects in a plane,” *Algorithmica*, vol. 19, pp. 129–147, 1997.

[15] A. Sudsang and J. Ponce, “On grasping and manipulating polygonal objects with disc-shaped robots in the plane,” in *Proc. of ICRA*, pp. 2740–2746, 1998.

[16] 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 *Proc. of ICRA, to appear*, 2012.

[17] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, “Grasping by caging: A promising tool to deal with uncertainty,” in *Proc. of ICRA, to appear*, 2012.

[18] E. Rimon and J. W. Burdick, “New bounds on the number of frictionless fingers required to immobilize 2D objects,” in *Proc. of ICRA*, pp. 751–757, 1995.