Motion Planning for Cooperative Transportation of a Large Object by Multiple Mobile Robots in a 3D Environment

Atsushi YAMASHITA †, Masaki FUKUCHI †, Jun OTA †, Tamio ARAI † and Hajime ASAMA ‡

† Department of Precision Machinery Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, JAPAN
Tel: +81-3-5841-6486 (Direct In) Fax: +81-3-5841-8556 (Department Office)
‡ The Institute of Physical and Chemical Research (RIKEN)
2-1, Hirosawa, Wako-shi, Saitama, 351-0198, JAPAN
E-mail address: *yamasita@prince.pe.u-tokyo.ac.jp*

Abstract

In this paper, we propose a motion planning method for cooperative transportation of a large object by multiple mobile robots in a 3 dimensional environment. This task has various kinds of problems, such as path planning, manipulation, and so on. All of these problems can't be solved at once, since computational time is exploded. Accordingly, we divide a motion planner into a local manipulation planner and a global path (motion) planner, and design these two planners respectively. And we integrate two planners. Namely, we aim at integrating a gross motion planner and a fine motion planner.

As to the local manipulation planner, we build a manipulation technique, which is suitable for mobile robots by position-control. We compute conditions, in which the object becomes unstable during manipulation, and generate the each robot's motion considering the robots' motion errors and indefinite factors from the planning stage. As to the global path planner, we reduce the dimensions of the configuration space (C-space) using the feature of transportation by mobile robots. We can find a solution with searching in this smaller dimensional C-space using the potential field defined in the C-space. And constraints of the object manipulation are considered as the potential function.

We verify the effectiveness of our proposed motion planning method through simulations and experiments.

Keywords: *Multiple Mobile Robots, Motion Planning, Cooperative Transportation, Manipulation, Motion Error, Cell Decomposition, Potential Field*

1. Introduction

It is expected that mobile robots undertake various tasks in manufacturing plants, warehouses, construction sites, and so on. In order to improve flexibility and fault tolerance of tasks, the concept of cooperation by multiple mobile robots is proposed [1]. In the future, mobile robots should work in a real 3 dimensional environment. In this complicated situation, a good motion planning method is very important to accomplish tasks efficiently. There is a big demand for a transport task with robots in particular. Therefore, we propose a motion planning method for cooperative transportation of a large object by multiple mobile robots in a 3 dimensional environment. However, this task has various kinds of problems. For example, we must plan paths of the object to avoid obstacles, construct a stable manipulation method, and decide robot motions. All of these problems can't solve at once, since computational time is exploded. Conventional path planning methods consider only geometrical and topological conditions in particular, such as shapes of obstacles and robots. They don't consider statics and dynamics of the object when robots manipulate it. Accordingly, we divide a motion planner into a local manipulation planner and a global path (motion) planner. The former treats how to manipulate an object (Fig. 1(1)). The latter plans paths of the object and robots (Fig. 1(2)). To integrate two planners, constraints of the object manipulation are considered in the path (motion) planner. In other word, we aim at integrating a gross motion planner and a fine motion planner.

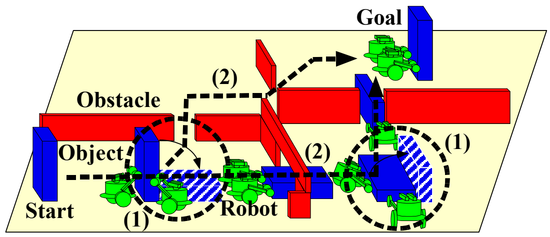


Fig. 1 Transportation Task

The composition of this paper is detailed below. The next chapter reviews the previous works and describes outline of our motion planning method. In chapter 3 and 4, the local manipulation planner and the global motion planner are explained. In chapter 5, we verify our method with simulations and experiments. The conclusions are discussed in chapter 6.

2. Transportation Task by Mobile Robots

2.1 Previous works

Many researches are studied about the motion planning for the classical mover's problem in a 3 dimensional environment. This problem is very difficult because of the high dimensional configuration space (C-space) [2]. A brute force search can solve problems with low dimensional C-space (for example, path planning of one mover in a 2 dimensional environment). However, a simple planner cannot solve high dimensional problems (for example, in a 3 dimensional environment, the dimension of the C-space is at least six). There are many studies about a path planner with high dimensional C-space [3]. Kondo used multi-heuristics by searching [4]. Barraquand et al. constructed a randomized planner [5] and Gupta et al. proposed a backtracking method [6] by solving problems with many degrees of freedom. Hwang et al. proposed resolution-complete and efficient method [7] and Kavraki et al. proposed a probabilistic roadmap method [8]. These methods can find a path of an object or a robot efficiently. These planners purpose finding a path in the C-space to avoid obstacles. In transportation tasks, we need to consider motions of workers or robots that transport and manipulate an object. And these planners don't take account of workers' or robots' motions. Therefore, generated paths (motions) are difficult to realize, for there is no consideration about stability of an object and how to transport and manipulate the object.

As to cooperative transportation [9-10] and manipulation of objects [11-12] with multiple mobile robot system, there are many studies. Rus, et al. [9] built an object pushing and transporting method using the sensor information. The method is based on the knowledge obtained conventionally in the research field of the pushing manipulation [13-16]. Kosuge et al. [10] aimed at transporting an object by lifting, and adopted the feedback control method using the information of robots' force sensors. Khatib et al. [11] controlled inner force that is applied to an object and adopted the stable handling strategy to compensate the motion errors of robots. Sawasaki et al. [12] realized rolling pose change of an object with two mobile manipulators. These methods can create a path of an object and robots or a manipulation way of the object in a 2 dimensional environment by using sensor data. In a 3 dimensional environment, the dimension of the C-space is too large for these methods to plan motion of an object and robots.

2.2 Outline of proposed motion planner

In this method, we divide a motion planner for transportation task into a local manipulation planner phase and a global motion planner phase (Fig. 2). The local planner outputs a way and information of manipulation. The manipulation way indicates motions of an object and robots to change the pose of the object safely. The information of manipulation indicates the manipulation space that is an area necessary to manipulate the object and the manipulation cost that is the necessary time to finish the manipulation. The information of manipulation is utilized to plan the global motions of the objects and the robots. The global motion planner outputs paths of the objects and the robots while avoiding obstacles. The global planner decides when and where to manipulate the object and what kind of manipulation way can be realized based on the result of the local manipulation planner.

2.3 The local manipulation planner

For the local manipulation planner, one of the biggest problems when mobile robots work is the effect of the robots' position errors. In the previous works, the force sensor's information and the image information from the camera are fed back to correct the motion errors. The motion errors of mobile robots are avoided with the on-line method. And excessive inner force that is applied to an object is avoided by force control approach. In other words, feedback control approach has been adopted to manipulate an object by multiple mobile robot system. However, it is difficult for mobile robots to measure force correctly and to manipulate an object with a force-control method like fixed manipulators while they move. Then, in this research, we propose the robust manipulation planning method for the motion errors by taking into consideration of the motion errors in a motion planning stage. We aim at constructing the planning method of manipulation that is suitable for the mobile robots under position control [18].

2.4 The global motion planner

For the global motion planner, the biggest problem is an explosion of the computation time that results from the high dimensional C-space. In this paper, we reconstruct the C-space and reduce the dimension of C-space by considering the feature of transportation task by mobile robots. And we can find a solution with searching in this smaller dimensional C-space using the potential field defined in the C-space. And constraints in manipulating are considered as the work area and the potential function. The work area is computed as the manipulation space, and the potential function is computed as the manipulation cost in the local planner.

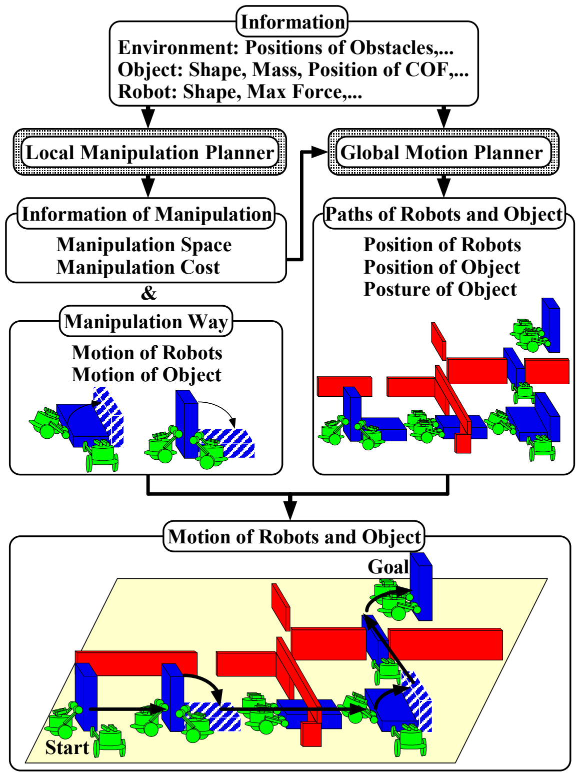


Fig. 2 Outline of Motion Planner

2.5 Manipulation and Transportation Style

We propose that robots manipulate an object by pushing with sticks in multiple mobile robots system (Fig.

3). When the manipulating tasks are carried out, the stability of operation is improved because the contact area to the object becomes larger [17]. The robots tumble the object to change its face that contacts with the floor from one face to the next one. By repeating this tumbling operation, it is possible to change the pose of the object whose arbitrary face touchs the floor.

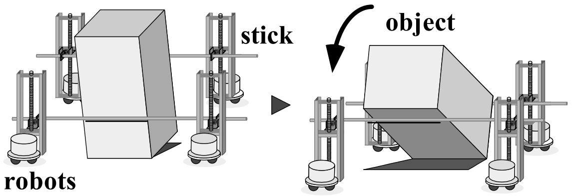


Fig. 3 Tumbling Manipulation

There occur position errors when position controlled mobile robots move. Therefore, there exists a risk that multiple mobile robots apply excessive inner force to an object when they touch the object at the same time. Then, in this paper, more than two sticks cannot touch the object at any time to avoid excessive inner force when the robots move. Since it is difficult for mobile robots to lift up a large or heavy object, they manipulate the object without lifting up operation. Therefore the object always contacts with the environment (a floor). The robots can move the sticks up and down. And pushing and tumbling operations are adopted here.

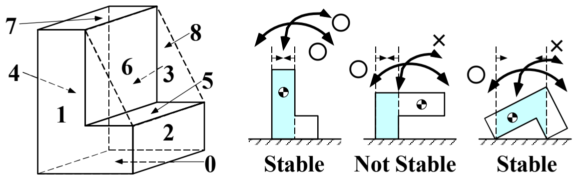
And the robots transport the object around it. While transporting the object, robots don't change their positions to the object. In other word, they don't change their formation. The robots depart from the object when they manipulate it and they keep close to the object when they transport it.

3. Local Manipulation Planning

3.1 Possible Manipulation

The robots tumble the object while one edge (this edge is the center of rotation) always contacts with the floor. We make the following assumptions for analysis:

- All motions are quasi-static.
- The geometry of an object and the location of its center of mass are known.
- Coefficient of friction between the object and the floor and between the object and the stick are known.
- All frictional interactions of the object are described by Coulomb's law of friction.
- Robots can move in omni-directions
- Robots have lift-up mechanisms to control their end-effectors' height (contact points' height).



(a) Object Face **(b) Possible Manipulation**

Fig. 4 Object Representation

In the local planner, we consider all possible manipulation. At first, we compute the state that we can ground an object. A face number represents the grounded state. For example, in the case of an object shown in Fig. 4(a), when Face 0, 1, 2, 3, 4, 7 (ordinary face) and Face 8 (virtual face) touch the floor, the object can be grounded (In this state, robots can transport an object). And Face 5 and 6 cannot touch the floor, the object cannot be grounded. Next, we check if the object can be manipulated from one state to another state by tumbling safely. We use matrix $\boldsymbol{A_{ij}}$ to express whether it is a possible manipulation or not. If the manipulation from Face i to Face j is possible, $\boldsymbol{A_{ij}} = 1$. And if it is impossible, $\boldsymbol{A_{ij}} = 0$ (Fig. 4(b)). In the case of Fig. 4, we can get $\boldsymbol{A_{ij}}$ as follows (Equation (1)).

$$\boldsymbol{A_{ij}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{1}$$

3.2 Formulation of Manipulation

Under the assumptions in section 3.1 and the situation mentioned as in Fig. 5, a 2 dimensional model can express this situation in the local manipulation planner. We can describe these constraints about the object.

$$\mathbf{F_s} + \mathbf{F_e} + \mathbf{G} = \mathbf{0} \tag{2}$$

$$\mathbf{p_s} \times \mathbf{F_s} + \mathbf{p_c} \times \mathbf{G} = \mathbf{0} \tag{3}$$

where

$\mathbf{F_s}$: force at stick point S, $\mathbf{F_e}$: force at contact point O

$\mathbf{G} = (0, -mg)$: force at the center of the mass (G)

$\mathbf{P_s} = (x, y)$: position of the stick

$\mathbf{P_g} = (x_g, y_g)$: center of the mass of the object

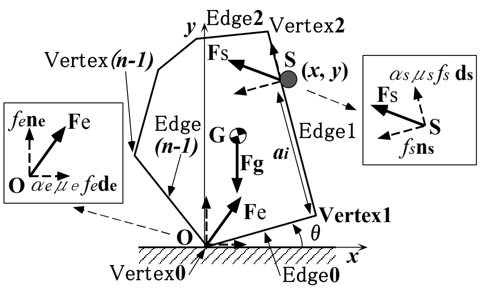


Fig. 5 2D Object Model

Let f_s be the normal contact force at S, and let f_e be the normal contact force at O.

$$\mathbf{F_e} = f_e \mathbf{n_e} + \alpha_e \mu_e f_e \mathbf{d_e} \tag{4}$$

$$\mathbf{F_s} = f_s \mathbf{n_s} + \alpha_s \mu_s f_s \mathbf{d_s} \tag{5}$$

where

$\mathbf{n_s} = (u_n, v_n)$, $\mathbf{d_s} = (u_d, v_d)$

From Equations (2-5), we obtain

$$f_s u_n + \alpha_s \mu_s f_s u_d + \alpha_e \mu_e f_e = 0 \tag{6}$$

$$f_s v_n + \alpha_s \mu_s f_s v_d + f_e - mg = 0 \tag{7}$$

$$\begin{aligned} f_s\left(xv_n - yu_n\right) + \alpha_s \mu_s f_s\left(xv_d - yu_d\right) \\ - mg\left(x_g \cos\theta - y_g \sin\theta\right) = 0 \end{aligned} \tag{8}$$

$$f_e > 0 \quad \text{and} \quad f_s > 0 \tag{9}$$

$$\left|\alpha_e\right| < 1 \quad \text{and} \quad \left|\alpha_s\right| < 1 \tag{10}$$

$$f_s < F_{\max} \tag{11}$$

where

f_s : force at S, f_e : force at O

$F_{\max}$: maximum force of robot

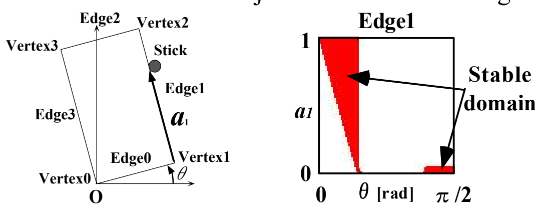
$\mu_{s,e}$: coefficient of friction at S, O

m : mass of the object, θ : angle of the object

We can know whether the object is stable or not by solving Equation (6-8) under constraint of Inequalities (9-11). Inequality (9) means that the object does not detached from the stick and the floor. Inequality (10) means that the object does not slip and keep stable. Inequality (11) means that the robots do not apply the force to the object beyond their ability.

3.3 Stable Domain Graph

We make the stable domain graph that indicates the position of the stick and the angle of the object when the object is stable. Y-axis means the parameter $\boldsymbol{a}_i$ showing the contact position with the object, and X-axis means the parameter θ showing the angle of the object (Fig. 6). The parameter $\boldsymbol{a}_i$ ($0< \boldsymbol{a}_i <1$) expresses the contact point of the stick at Edge i of the object. For example, when $\boldsymbol{a}_1$=0, the stick contacts with the object at Vertex 1. When $\boldsymbol{a}_1$=0.5, the stick contacts with the object at the center of Edge 1.

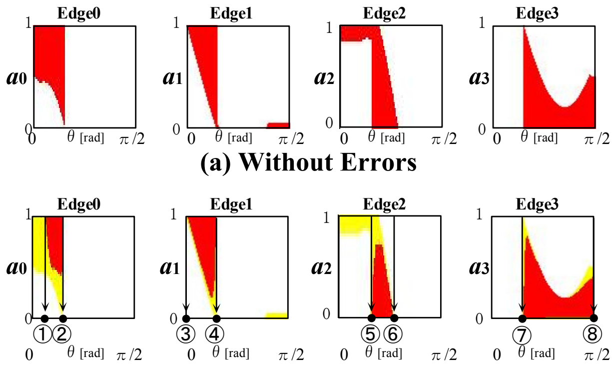


(a) Parameter $\boldsymbol{a}_i$ and θ (b) Stable Domain Graph

Fig. 6 Stable Domain of Edge 1

The object is stable, if the parameter (θ, $\boldsymbol{a}_i$) is in stable domain. Therefore, if the object is operated in this domain, the object manipulation will not fail. Because the number of the edge is four in case of rectangles, four stable domain graphs which show stable domain of each edge are generated (Fig. 7(a)). And the limitation on the movement of robots (for example, the end-effectors' height is limited from $h_{\min}$ to $h_{\max}$) can be considered in the stable domain graph.

The effect of motion errors of the mobile robots and environmental indefiniteness are taken into consideration. As the error factor which influences a stable domain is; (i) the motion errors of mobile robots, and (ii) the accuracy of given data and the change of a coefficient of friction. As to the former problem, we search a domain where the stability of the object is maintained if the position error ($\pm \Delta x$, $\pm \Delta y$) exists on the position of the stick. We can put the bend of the stick on this position error. As to the latter problem, we search a domain where the stability of the object is maintained if coefficient of friction $\mu_{s,e}$ changes within the range of $\pm \Delta \mu_{s,e}$. And the input data errors (position of the center of the mass ($\pm \Delta x_g$, $\pm \Delta y_g$), mass of the object $\pm \Delta m$) can be considered. The stable domain graph considering the motion errors is shown in Fig. 7(b). In Fig. 7(b), the domain reduced as compared with Fig. 7(b) is a domain in which work fails, when there are motion errors or another factors.



(a) Without Errors

(b) With Errors

Fig. 7 Stable Domain Graph of Each Edge

3.4 Operation Graph

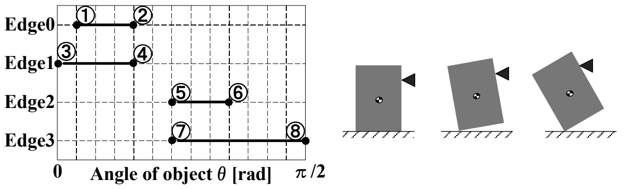
We generate the operation graph and utilize it to plan the motions of the robots efficiently. The procedure of generating the operation graph is as follows. First, we select characteristic points where the operation method may change in the stable domain graph. Concretely, we choose the points whose θ values are the largest and the smallest in stable domain (Fig. 7(b)). And these points are regarded as nodes in the operation graph. Therefore, in the operation graph, all nodes that have physical meanings are collected (Fig. 8(a)). Information that nodes hold in the operation graph is the edge number and the angle of the object θ. In the operation graph, information about $\boldsymbol{a}_i$ is not considered to plan the motions of the sticks efficiently.

In the operation graph, the pose change of the object is planned by moving from one node to other node. In the rectangle case shown in Fig. 6(a), the pose change is performed by moving from node ③ (θ= 0 [rad]) to node ⑧ (θ=π/2 [rad]).

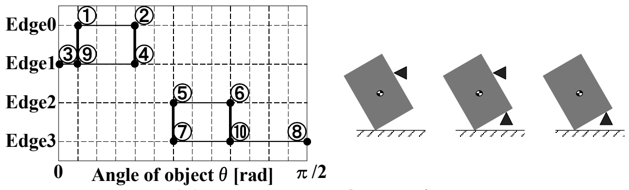
In the operation graph, the arc between two nodes indicates three types of operation methods, (a) the continuous operation, (b) the hand-over operation, and (c) the transfer operation. The arc that is a horizontal line between two nodes indicates the continuous operation (Fig. 8(a)). In this operation, it is possible to change the angle of the object without changing the edge that the stick contacts with. During the continuous operation, the contact point of the stick continuously changes. It means that $\boldsymbol{a}_i$ in the stable domain graph changes continuously. The perpendicular arc indicates the hand-over operation (Fig. 8(b)). In this operation, it is possible to change the edge that the stick contacts with without changing the angle of the object. Node ⑨ and ⑩ are newly generated in the operation graph. The oblique arc indicates the transfer operation (Fig. 8(c)). In this operation, it is possible to change both the angle of the object and the edge that the stick contacts with. This operation is performed when there is no stable domain between two nodes.

To express the difficulties of these operations, the distance of the arc, which means moving cost between two nodes, is introduced. Then, the problem of choosing the manipulation method among three operations comes back to the shortest path problem. Long distance from the start node to the goal node means that the pose changing operation of the object is difficult. Therefore, we choose the shortest path, because the pose-changing task can be performed easily. The distance (cost) of the each arc is defined in Equation (12) and Table 1 ($\Delta\theta$ means the change of the object angle).

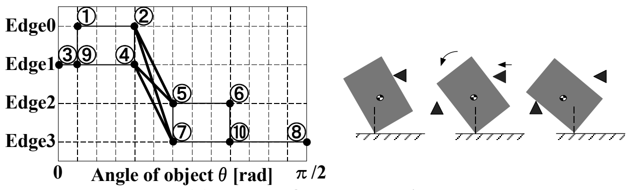
$$\cos t_{a,b,c} = A_{a,b,c}\left(\Delta\theta + B_{a,b,c}\right) \qquad (12)$$



(a) Continuous Operation



(b) Hand-over Operation



(c) Transfer Operation

Fig. 8 Operation Graph and Operation

Accordingly, the costs are determined as follows: cost (a) < cost (b) < cost (c). The cost of the continuous operation is lower, because it is the easiest way to manipulate the object. This operation can be accomplished with one stick and the total time of operation is the shortest compared to other operations. And the cost of the transfer operation is higher, because it is the most difficult. This operation needs two sticks and the object is not stable during this operation. When this operation is performed, a greater effect of the uncertain factor of the environment must be considered compared with other operations.

Table 1 Parameter A and B

A			B		
a	b	C	a	b	c
1	100	1000000	0	100	0.1

After the costs of arcs are determined, Dijkstra's algorithm is used to solve the shortest path planning problem. From the result of the search, the shortest path is gained; ③(initial state)→⑨→④→⑦→⑩→⑧(goal state). Path ③→⑨→④ means the continuous operation on Edge 1. Path ④→⑦ means the transfer operation between the stick on Edge 1 and Edge 3. Path ⑦→⑩→⑧ means the continuous operation on Edge 3.

It is not determined the position where the stick contacts with each edge ($\boldsymbol{a}_i$). The result obtained here is the order of the operation where the stick contacts. The orbit of a stick is not determined at the present.

3.5 Determination of Robot Motion

In this research, the orbit of the stick is determined after a procedure of object operation is determined. If it is the manipulation method for fixed manipulators, the contact position orbit where the force applied to the stick is always minimized may be good. For example, when the stick contacts with Edge 1, the continuous operation where $\boldsymbol{a}_1$ is near 0 leads to the situation where control force is minimum. However, when optimizing the control force, the orbit of a stick becomes a circle-like shape. In this case the perpendicular and the horizontal speed of the stick need to be controlled in a complicated way every time. This has the risk of failing tasks conversely, for it is inconvenient for mobile robots to make complicated operation. Then, the stick orbit that mobile robots can easily control forms. In this research, a straight-line orbit is adopted while avoiding the object. The stick positions when the continuous operation begins and ends are determined to minimize the force at these points, and the meantime is connected linearly. When the orbit is out of the stable domain, the stick position of the beginning and the ending are changed suitably. Then solution that the orbit is always in the stable domain is searched.

The manipulation method of the object shown in Fig. 9 is off-line-planned by the local manipulation planner.

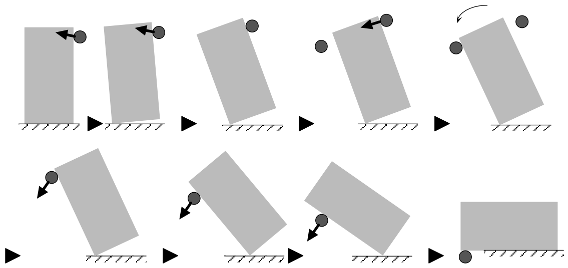


Fig. 9 Planned Manipulation Method

3.6 Information of Manipulation

The manipulation space where robots works with object is represented by the cell decomposition method (the octree method). And the manipulation cost that means the total time to finish manipulation is calculated in the local manipulation planner. The information is used in the global planner.

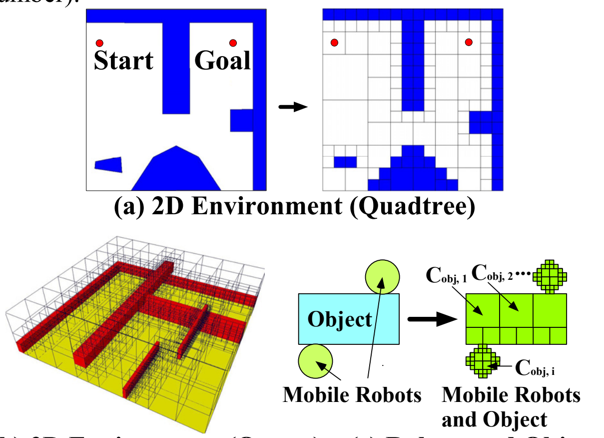
4. Global Motion Planning

4.1 Representation of the Environment

In the global motion planner, paths of the objects and robots while avoiding obstacles can be find. The global planner decides when and where to manipulate the object and what kind of manipulation way can be realized based on the result of the local manipulation planner.

In our method, all things (an object, robots and obstacles) are represented by an octree that is the approximate cell decomposition method for 3 dimensional environment. By using the octree method to represent them, all things can be dealt with in the same way. And the number of cell is small compared with the exact cell decomposition method. When we solve problems with high dimensional C-space, this representation is efficient and practical. In our method, we aim at the resolution-complete and efficient planner.

A 2 dimensional environment represented by a quadtree is shown in Fig. 10(a) and a 3 dimensional environment represented by an octree is shown in Fig. 10(b). Cells are divided into white ones (free space) and black ones (obstacles). The object and robots can be represented as one object, because the formation of the robots is not changed while transporting the object (Fig. 10(c)). As to the object and robots, cells where the object and robots exist are called the object cell $C_{obj,j}$ (j: cell number).



(a) 2D Environment (Quadtree)

(b) 3D Environment (Octree) (c) Robots and Object

Fig. 10 Representation by Cell Decomposition

For representing the environment by an octree, a path free from obstacles can be find efficiently as a sequence of white cells from a start configuration to a goal. And for all things are represented in the same way, collision check between two things can be easily detected by checking the overlaps of cells. Note that, in this method, C-obstacle doesn't compute positively and the planner checks the collisions in the real world (not in C-space).

4.2 Reconstruction of Configuration space

We deal with the problem that multiple mobile robots (the number of robot: r) transport an object around it. The dof of the C-space is (6+r) shown in Fig. 11. (The dof of the object is 6 and the dof of robots is r).

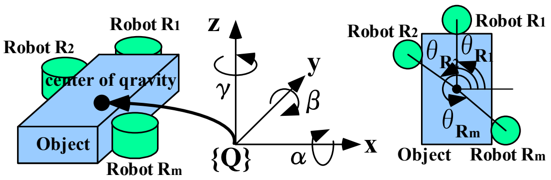


Fig. 11 Dof of C-space

Because of the high dimensional C-space, we reduce the dimensions of the C-space using the feature of transportation with mobile robots. In transportation task with mobile robots, the possible paths and motions are limited and are different from the paths in the problems of free flying objects. Robots cannot transport the object while changing its pose in complicated way. And they change the object pose from one grounded state that one face of the object contacts with the floor to other grounded state. These states are abstracted as characteristic configurations and the planner uses these configurations to find a practical path.

The reconstructed C-space is shown in Fig. 12. In Fig. 12, (1) corresponds R_1, R_2,…,R_m in Fig. 11 and expresses them as the formation of mobile robots (1 dof). (2) corresponds γ in Fig. 11 and expresses it as the orientation of the object and the robots (1 dof). (3) corresponds α and β in Fig. 11 and expresses them as the face of the object contacted with the floor (1 dof). Accordingly, (6+r) dof can be reduced to 5 dof.

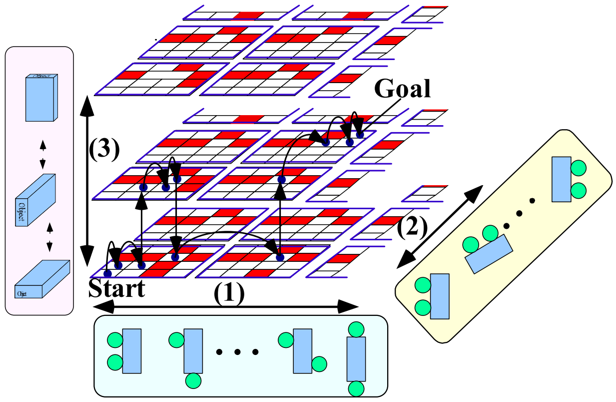


Fig. 12 Reconstructed C-space

4.3 Graph Search Method

A graph search is performed by using the following evaluation function f.

$$f = \sum_{i=0}^{s} U(q_i) + \sum_{i=0}^{s} M(q_{i-1}, q_i) \tag{13}$$

where

q_i : configuration of the object and robots

s : total step from a start to a current configuration

In Equation (13), the first component is the force received from the potential field generated at the configuration q_i. And the second component is the manipulation cost computed in the local planner.

After the cost evaluation function f is defined in Equation (13), A* algorithm is used to find the optimal path in C-space from the start configuration to the goal configuration. By using this function, we can gain a path that is far from obstacles and total time that robots accomplish the transportation task is short.

4.4 Potential Field

A potential field is constructed with a repulsive force from obstacles and attractive force from a goal configuration. The total force from the potential field where the configuration of the object and robots is q, can be calculated as a sum of the force acted on the object cell $C_{obj,j}$ (j = 1 ,…, k). The total force acted on q is defined as follows. (k is the number of the object cell)

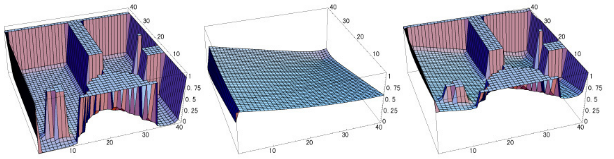
$$U(q)=\sum_{j=0}^{k}P_w(C_{obj,j}) \quad (14)$$

$$P_w(C_{obj,j})=P_{rep}(x_j)+p_{att}(x_j) \quad (15)$$

where

x_j : reference point of object cell $C_{obj,j}$
$P_w(C_{obj,j})$: The force acted on the object and robots
$P_{rep}(x_j)$: repulsive force from obstacles
$P_{att}(x_j)$: attractive force from the goal

In this paper, we adopt the potential field from obstacles mentioned in [19], which is one of the simplest potential functions. The potential field in the 2 dimensional environment of Fig. 10(a) is shown in Fig. 13. And this potential function is normalized with the velocity of robots. Therefore, the dimension of $U(q_i)$ is same as the dimension of a time.



(a) P_{rep} **(b) P_{att}** **(c) P_w**

Fig. 13 Potential Field

4.5Manipulation Cost and Manipulation Space

The manipulation cost that is the total time to finish the manipulation is computed by local manipulation planner. If robots don't manipulate the object and only transport it from configuration q_{i-1} to q_i, the cost $M(q_{I-1},q_i)$ is 0.

The manipulation space computed by the local planner and moving area of robots (Fig. 14) computed by the global planner are both considered as the work area. By making object cell $C_{obj,j}$ based on the work area, the space that is need for manipulation and movement of robots is ensured in the global motion planner.

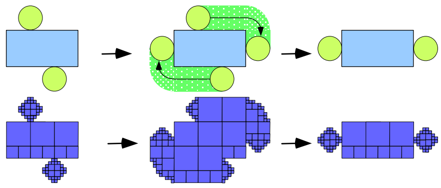


Fig. 14 Moving Area of Robots (Work area)

5. Simulations and Experiments

5.1 Simulations

We verified the proposed planner in this paper through the simulation. The motion planner computes the paths that two robots transport an L-shape large object in a 3 dimensional environment shown in Fig. 10(b).

Simulation results of the global motion planner and the local manipulation planner are shown in Fig. 15 and Fig. 16 (In Fig. 16, obstacles are not displayed because of the simplicity of observation). In Fig. 15, the robots transport the object while avoiding obstacles. And they change the pose of the object at points A and B. The number of times of manipulation can be reduced by using manipulation cost as potential function. In Fig. 16, the manipulation way of the object at point A is computed. In this simulation, objective task has been realized without failure, such as the object slips or falls down under the motion errors of robots and environmental indefiniteness.

It takes about one hour to compute the path of the object and robots with Ultra SPARC-II (334MHz). And this shows that our proposed planner can find appropriate solutions within a practical time efficiently despite of complexity of the problem.

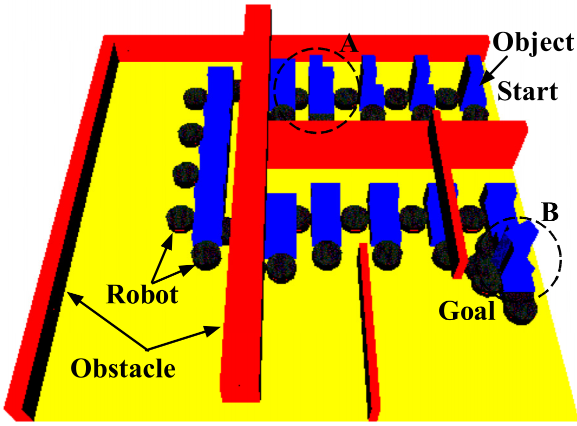
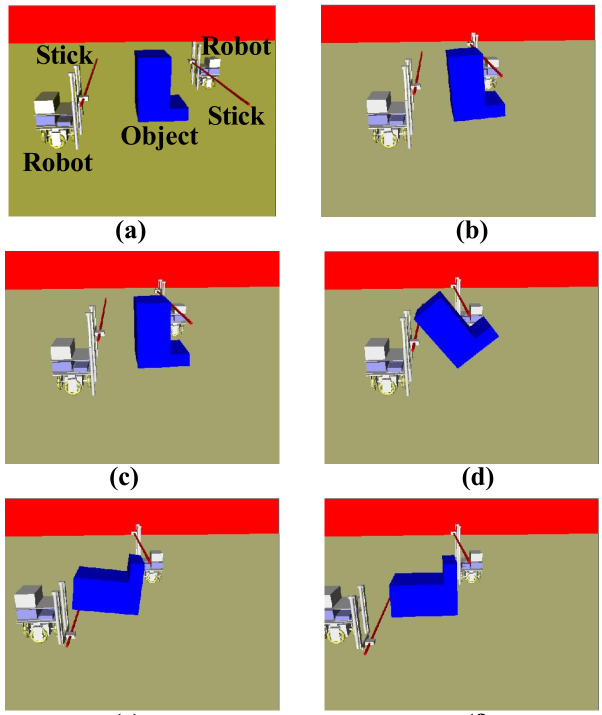


Fig. 15 Simulation Results I

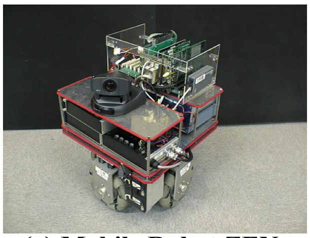


(a) **(b)** **(c)** **(d)** **(e)** **(f)**

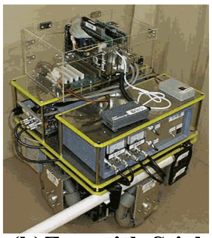
Fig. 16 Simulation Results II

5.2 Experiments

We applied the proposed planning method of manipulating an object to real robot system. The mobile robots can move in all directions [20] and have the sticks.



(a) Mobile Robot ZEN



(b)Zen with Stick

Fig. 17 Omni-Directional Mobile Robot ZEN

The external-world sensor is not carried on the mobile robot. And we set the motion velocity of robots slow enough to keep quasi-static motion.

A result of a simple experiment (tumbling a rectangle with two robots) that adopted our local manipulation planning method is shown in Fig. 18. Here, it can be checked that objective operation can be performed without using sensor information, and the validity of this method was shown.

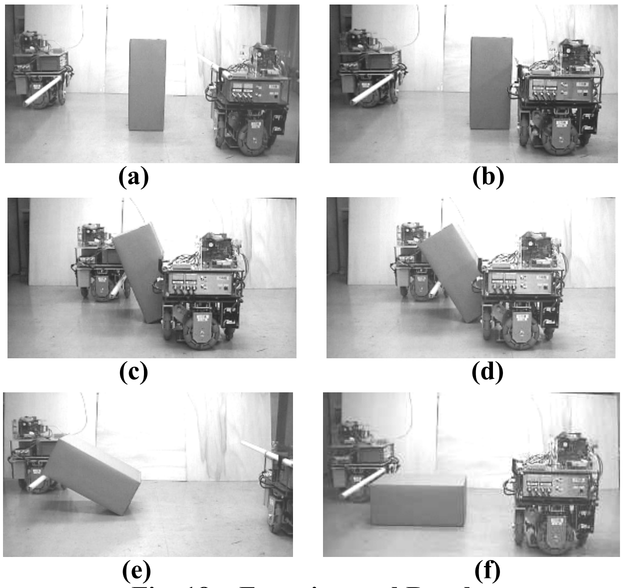


Fig. 18 Experimental Results

6. Conclusions

In this paper, we propose a motion planning method for a cooperative transportation of a large object by multiple mobile robots in a 3 dimensional space. We divide the motion planner into a local manipulation planner and global motion planner.

As to the local manipulation planner, we consider the motion errors in a planning stage beforehand, and build the manipulation technique that is suitable for position-controlled mobile robots. And as to the global motion planner, we reduce the dimensions of the C-space and can find a solution with searching in this smaller dimensional C-space using the potential function. After constructing two planners respectively, we integrate them. Namely, a gross motion planner and a fine motion planner can be integrated by our method.

Simulations and experiments have verified the effectiveness of proposed planning method.

References

[1] Tamio Arai and Jun Ota: "Let us Work Together -Task Planning of Multiple Mobile Robots-", Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.298-303, 1996.

[2] Jean-Claude Latombe: "Robot Motion Planning", Kluwer Academic Publishers, 1991.

[3] Hwang Y. K. and Ahuja N.: "Gross motion planning – a survey", ACM Computing Surveys, Vol.24, No.3, pp.219-292, 1992.

[4] Koichi Kondo: "Motion Planning with Six Degrees of Freedom by Multistrategic Bidirectional Heuristic Free-Space Enumeration", IEEE Transactions on Robotics and Automation, Vol.7, No.3, pp.267-277, 1991.

[5] Jerome Barraquand and Jean-Claude Latombe: "Robot Motion Planning: A Distributed Representation Approach", The International Journal of Robotics Research, Vol.10, No.6, pp.628-649, 1991.

[6] Kamal Kant Gupta and Zhenping Guo: "Motion Plannning for Many Degrees of Freedom: Sequential Search With Backtracking", Proceedings of the 1992 IEEE International Conference on Robotics and Automation, pp.2328-2333, 1992.

[7] Yong K. Hwang and Pang C. Chen: "A Heuristic and Complete Planner for the Classical Mover's Problem", Proceedings of the 1995 IEEE International Conference on Robotics and Automation, pp.729-736, 1995.

[8] Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe and Mark H. Overmars: "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", IEEE Transactions on Robotics and Automation, Vol.12, No.4, pp.566-580, 1996.

[9] Daniela Rus, Bruce Donald and Jim Jennings: "Moving Furniture with Teams of Autonomous Robots", Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.235-242, 1995.

[10] Kazuhiro Kosuge, Tomohiro Oosumi and Kunihiro Chiba: "Load Sharing of Decentralized-Controlled Multiple Mobile Robots Handling a Single Object", Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pp.3373-3378, 1997.

[11] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg and A.Casal: "Vehicle/Arm Coordination and Manipulate Mobile Manipulator Decentralized Cooperation", Proceedings of 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.546-553, 1996.

[12] Naoyuki Sawasaki and Hirochika Inoue: "Cooperative Manipulation by Autonomous Intelligent Robots", *Journal of the Japan Society of Mechanical Engineers*, Series C, Vol.59, No.564, pp.2318-2325, 1993, in Japanese.

[13] Matthew T. Mason: "Mechanics and Planning of Manipulator Pushing Operations", The International Journal of Robotics Research, Vol.5, No.3, pp.53-71, 1986.

[14] Randy C. Brost: "Automatic Grasp Planning in the Presence of Uncertainty", The International Journal of Robotics Research, Vol.7, No.7, pp.3-17, 1988.

[15] Michael A. Peshkin and Arthur C. Sanderson: "The Motion of a Pushed, Sliding Workpiece", IEEE Journal of Robotics and Automation, Vol.4, No.6, pp.569-598, 1988.

[16] Srinivas Akella and Matthew T. Mason: "Posing Polygonal Objects in the Plane by Pushing", The International Journal of Robotics Research, Vol.17, No.1, pp.70-88, 1998.

[17] Atsushi Yamashita, Jun Sasaki, Jun Ota and Tamio Arai: "Cooperative Manipulation of Objects by Multiple Mobile

Robots with Tools", Proceedings of the 4th Japan-France/2nd Asia-Europe Congress on Mechatronics, pp.310-315, 1998.
[18] Atsushi Yamashita, Kou Kawano, Jun Ota, Tamio Arai, Masaki Fukuchi, Jun Sasaki and Yasumichi Aiyama: "Planning Method for Cooperative Manipulation by Multiple Mobile Robots using Tools with Motion Errors", Proceedings of 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.978-983, 1999.
[19] Yong K. Hwang and Narendra Ahuja: "A Potential Field Approach to Path Planning", IEEE Transactions on Robotics and Automation, Vol.8, No.1, pp.23-32, 1992.
[20] Hajime Asama, Masatoshi Sato, Luca Bogoni, Hayato Kaetsu, Akihiro Matsumoto and Isao Endo: "Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism", Proceedings of the 1995 IEEE International Conference on Robotics and Automation, pp.1925-1930, 1995.