ROBOMECH Journal
a SpringerOpen Journal

RESEARCH ARTICLE Open Access

Realization method for a rearrangement task by multiple mobile robots in consideration of map errors

Norisuke Fujii, Taiki Ogata, Tatsunori Hara and Jun Ota*

Abstract

In this paper, we propose a realization method for a rearrangement task involving multiple movable objects and mobile robots. All of the objects are transported from their initial positions to goal positions. It is important for mobile robots to cope with map errors in order to execute tasks in a real environment. However, a rearrangement task is a very complicated process involving constraints related to transportation order and scarcity of spatial resources. Therefore, it is difficult to develop an adequate method to deal with map errors and still maintain task performance. We primarily address two questions: what kinds of tactics are needed and when these tactics should be applied. Our method involves adopting a problem-partitioning structure that divides a complicated rearrangement problem into simple subproblems. Furthermore, we design a real mobile robot group. Using the realization method and real robot group, we conduct experiments involving a rearrangement task in an actual environment. The results show that our proposed method can cope with map errors quickly, while maintaining task realization performance.

Keywords: Rearrangement; Multiple robots; Real robots; Re-planning; Movable objects

Background

These days, mobile robots are expected to execute tasks with increasing variety and complexity. Among these tasks, rearrangement tasks are fundamental. Robots transport objects from initial positions to goal positions. Rearrangement tasks are used in various applications, such as product transportation in warehouses [1] and clean-up tasks in indoor situations [2,3]. An example of a problem encountered by a robot is shown in Figure 1. In this figure, each box labeled with an “R” indicates a robot. Other boxes indicate movable objects, and the black squares indicate obstacles. Objects are located in their initial positions, as shown in “Initial state,” and in their goal positions, as shown in “Goal state”.

When robots execute a task in a real environment, it is difficult for them to execute predetermined plans precisely because of localization errors, map errors caused by sensor noise or unforeseen objects, and transformation of working environments, i.e. change in position and orientation, of obstacles caused by events such as a closed door being opened. In this paper, for brevity, we refer to all these above difficulties as “map errors”. It is very important to cope with those map errors to realize the rearrangement task.

Related work

In general, task realization methods for a mobile robot can be divided into two groups: 1) methods to generate motion reactively based on information from sensors (without developing a task plan) and 2) methods for which a task plan is developed before the robot starts to move based on this plan, called off-line plan. A rearrangement task is very complicated involving constraints related to the order of transportation [4] and the scarcity of spatial resources. In most applications, the working environment is small, requiring robots to operate in narrow corridors through which they cannot pass [5]. Therefore, all of the previous studies for rearrangement tasks adopted the second method mentioned above [6-11].

* Correspondence: ota@race.u-tokyo.ac.jp
Research into Artifacts, Center for Engineering (RACE), The University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa, Chiba 277-8568, Japan

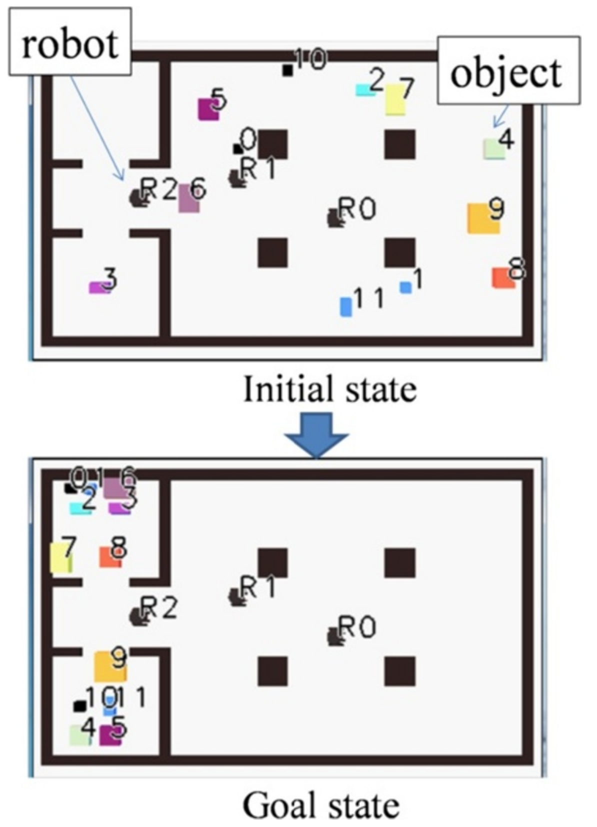


Figure 1 Example of a rearrangement task.

The planning problem for a rearrangement task is complicated. Most previous studies focused only on the problem of planning and did not propose methods for task realization to cope with map errors [6-10]. Only one previous study proposed a method to cope with map errors in a rearrangement task. In that study, the robot's motions were slightly modified by the localization process [11]. However, in a case involving the transportation, this method cannot be used to maintain effective task realization.

For tasks other than rearrangement tasks, several studies coped with map errors. Some examples are a path adaptation and a re-planning method for a sweeping task [12]. Some other papers deal with the problem of error recovery [13,14], but the targets are for specific tasks, such as mobile robot navigation [13], and for handling an object, such as a mobile manipulator [14]. They cannot be applied to rearrangement tasks.

Since a rearrangement task is very complicated, no studies have been conducted on developing appropriate motions to adapt to map errors.

Objective and approaches

The objective of this paper is to propose a task execution method for a rearrangement task that copes with map errors. To achieve this objective, we need to handle these issues:

- The proposed algorithm should adequately cope with various kinds of map errors ranging from small ones, such as sensing errors, to large ones, such as the transformation of obstacles.
- The processing time should be short enough to be applied to dynamic environments.

A conceptual illustration of our approach is shown in Figure 2. We assume the off-line planning result is given in advance before task realization, which is not the scope of this paper. The proposed task realization method solves a complicated rearrangement realization problem with the combination of relatively simple re-plan modules ("Re-plan modules A, B, C" in Figure 2) using a territorial approach [15-17]. This method makes it possible to solve the problem depending on the characteristics of errors. The off-line plan is partially modified to cope with map errors. If the errors are easy to deal with, the problem is solved quickly. On the other hand, if the errors are complicated, the problem is solved in a more deliberate way. Here, cares should be taken that each robot realizes the task distributedly by exchanging information, such as trajectories, with other robots.

We need to determine two issues: (a) the details of the modules and (b) the timings to activate each module. For issue (a), each module solves the corresponding error characteristics (shown as a conditional branch in Figure 2). With respect to issue (b), it is more difficult to design a rule that fulfills the requirement. Therefore, we set up these two basic policies: 1) the basic behavior of the robots are divided into three units, localization, path generation to a target position, and setting of the target position. Three modules (module A, B and C) are prepared corresponding to each part. 2) The modules should be activated when it is necessary to change the off-line plan. Following these policies may decrease calculation cost in most cases.

Problem descriptions

A rearrangement problem involving one robot was formulated by Alami et al. [18]. Fujii and Ota extended this work to a case involving multiple robots [19]. For the detailed formulation, please refer to their papers.

In this paper, the following assumptions are made:

1. Two grasping positions are used for an object. They are directed toward the front side and the back side of the object.
2. All of the robots have a differential drive and have 0 turning radius.
3. All of the robots can self-localize and localize objects but have errors.

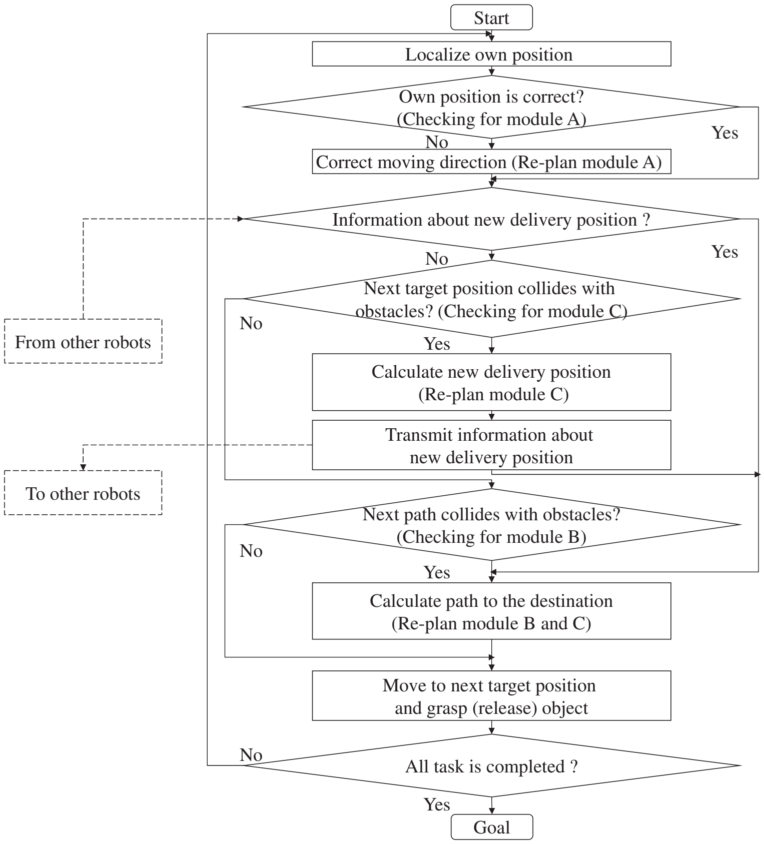

Start
Localize own position
Own position is correct? (Checking for module A)
No
Yes
Correct moving direction (Re-plan module A)
Information about new delivery position ?
No
Yes
Next target position collides with obstacles? (Checking for module C)
No
Yes
From other robots
Calculate new delivery position (Re-plan module C)
Transmit information about new delivery position
To other robots
Next path collides with obstacles? (Checking for module B)
No
Yes
Calculate path to the destination (Re-plan module B and C)
Move to next target position and grasp (release) object
No
All task is completed ?
Yes
Goal

Figure 2 A conceptual diagram of the proposed method.

4. The robots are equipped to communicate with each other. The communication noises can be ignored, assuming that communication is reliable.
5. Each robot can grasp only one object at a time, and each object requires only one robot to be transported.

In addition to the above assumptions, we prescribe constraints on the calculation time for one re-planning. For most applications (e.g., product transportation in a warehouse), the calculation time for off-line planning should be a few minutes [1]. On the basis of this knowledge, the calculation time was limited to five minutes in a previous study that addressed rearrangement planning [6]. Re-planning modules may be required several times. Therefore, the selected calculation time for one re-planning module should be shorter than the calculation time for off-line planning. In this paper, the selected time limit for one re-planning module is one minute.

For the third assumption, we need to consider map errors. For example, the errors are classified into four in [11]: (a) odometry error, (b) sensor noise, (c) error between the model environment and the real working environment, and (d) alteration of the environment, such as appearance of a new object. We need to deal with these map errors.

In a case where the numbers of robots and transported objects are changed, it is necessary to develop an entirely new plan. We do not consider such a case in this paper.

Methods

An outline

In the proposed method, each robot iteratively uses sensors to measure the working environment and execute the rearrangement task. From the sensor data, the robots estimate what kind of map errors exist. If needed, the robots activate the re-planning modules. Following the new, modified task plan, each robot restarts its movement. These procedures continue until all of the objects are transported to their goal positions.

The off-line plan

The off-line task plan needs to be expressed with a manipulation path, which is defined as “a finite sequence of paths in a configuration space for taking an object to somewhere (called transit path) and paths for placing an object somewhere (called transfer paths). The task of moving on a transfer path and placing an object somewhere is called a transfer task. In the rearrangement task, robots iterate these two tasks alternately" [7]. As for setting of the target positions of an object, sometimes it is necessary to put the object to a temporal position, not its goal position, for efficient transportation. The location where an object is temporarily placed is called a delivery position. Then the robot's trajectory can be expressed as shown in Figure 3, without loss of generality. As for the off-line plan methodology in this paper, we apply a previously developed approach [7], in which path development is realized with the planned results of territorial approach and extended PSP (Project Scheduling Problem [20]) solver.

Conceptual design of realization phase

As a method to cope with map errors in the rearrangement task, it is important to design the details of multiple modules and the timings to activate each module to develop the task realization procedure.

Design of re-plan modules

We design the details of the re-plan modules. For each module, it is necessary to modify the task plan in order to maintain the effectiveness and shorten the calculation time as much as possible.

In this paper, we design the following three modules that deal with several kinds of map errors.

Map errors are difference between the real environment and the model environment. The environment includes (a) placement of robots, and (b) placement of obstacles and movable objects. We design the modules from this point of view. As for (a), we think one module is necessary for dealing with the situation when the position in the model (recognized position) of the robot is wrong, i.e. different from the real position. As for (b), the present plan should be changed when we cannot no longer continue using the plan. As shown in Figure 3, the structure of the plan is combination of paths and the target positions. Here we can only change delivery positions among the target positions. From this discussion, we think we need to prepare two more modules: one is the module for path re-planning when the present paths cannot be applied, the other is the module for re-planning delivery positions when the present delivery positions cannot be used, due to positional change of obstacles.

For more detail, we set up the following three modules:

module A) Re-planning of the robot's moving direction. This can be activated with the localization method when necessary. By applying this module, robots are able to come back on the correct path.

module B) Re-planning of paths. This module can cope with a case where robots cannot use the present paths because of map errors.

module C) Re-planning of delivery positions. This module is necessary to the situation when the present delivery positions becomes ineffective with map errors. Following the re-planning of a delivery position, the robots also need to re-plan the transfer path. Therefore, this module requires a relatively large calculation cost compared to modules A and B.

Design of timings to activate re-plan modules

We design the timings to activate each module. If possible, it is desirable for robots to choose the most suitable module whenever map errors occur.

We propose two basic policies to design the timings to re-plan. First, we activate re-plan modules only when we judge from map errors information that we cannot continue the present plan. Second, in order to keep real-time performance, we divide checking routines whether to activate the re-plan modules (with low calculation cost) and main routines of re-planning (with high calculation cost). These policies decrease the total number of applied modules, and in most cases, the calculation time may be shortened. Based on the above policies, we propose the algorithm shown in Figure 2, consisting of three checking routines and three re-plan modules.

How to detect map errors is a very important issue for real-world working experiments. Here we assume that shapes, positions and orientations of objects are measured by using sensors on the robots, and map errors can be calculated by obtaining the difference between the above data and those on the map. Implementation in the experiments will be discussed in Section Details of modules and real robots.

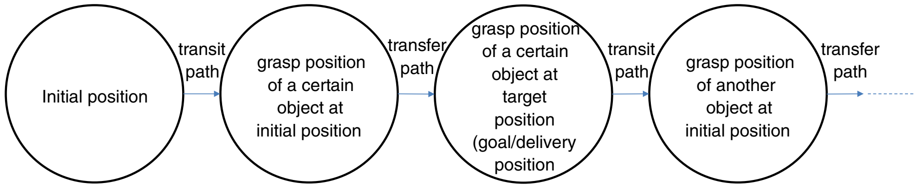

Initial position
transit path
grasp position of a certain object at initial position
transfer path
grasp position of a certain object at target position (goal/delivery position
transit path
grasp position of another object at initial position
transfer path

Figure 3 Sequence of robot off-line planning.

An overview of hardware settings

In order to develop task realization including re-planning, computer architecture is a very important issue. In the case of the mobile robot groups considered, there are two ways that computers can be used to develop a task plan and do re-planning. The first way is decentralized system approach, where each robot does decision-making by utilizing a computer. The second way is centralized system approach, where one off-board computer is utilized for task realization in the system. In this paper, we take the former approach. This is because (a) the former method is easily implemented and decreases communication costs, and (b) the latter method is inherently precarious. The whole system would fail if the computer fails.

Details of modules and real robots

Realization procedure

In the proposed method, three modules are applied: A) modification of the moving direction by the localization method, B) re-planning the paths, and C) re-planning the delivery positions, which are applied in the order of A, C, then B because latter half of the re-plan module C is the same as the re-plan module B. For module B, we utilize one of the existing methods [21]. This method is designed to always find a path and is easy to implement. For module C, we apply the same procedure proposed in ref. [7]: when a robot detects map errors, it determines a new delivery position for the object that the robot is transporting at that time. A new delivery position is found so that the entire task time may be as short as possible. Furthermore, the merit of the territorial approach is maintained. A new delivery position is determined so that the new path and the paths of the other robots do not cross.

The dashed line in Figure 2 indicates the communication step. The end point of the dashed line indicates the verification module executed by the robot receiving the communication.

The details of the procedure for one robot are as follows (Figure 2):

i. Given an off-line plan in advance.
ii. Obtain position and orientation of the robot by a localization method.
iii. If the position of the robot is same as the estimated one, move to step v.
iv. Correct moving direction (Re-plan module A).
v. Confirm the communication from other robots. If there is any information, move to step x.
vi. Check whether there is any overlap between the next target position (the delivery position) and obstacles. If no, move to step ix.
vii. Calculate a new delivery position (First half of Re-plan module C)
viii. Transmit information about new delivery position to other robots.
ix. Check whether there is any overlap between the path to the next destination and obstacles. If no, move to step xi.
x. Calculate the path to the destination (Re-plan module B. Latter half of Re-plan module C)
xi. Move to the next target position and grasp (or release) the object.
xii. If all task is completed, then end. Else go back to step ii.

The contents of the communication between robots consist of delivery positions that have been modified and their new locations. Each robot develops a new transfer path independently based on the location of the new delivery position. To re-plan the new path, we utilize the same method as the planning phase [19]. Similarly, we utilize the same module as the planning phase to re-plan the delivery position.

Development of real multiple mobile robots

To execute a rearrangement task by the above procedure, we need to develop a real multiple mobile robots, along with sensors to localize and locate objects and hardware for communication, including map errors. The task is very complicated, consisting of several transit and transfer path generation and delivery position calculation, and it is necessary to show the proposed algorithm can solve complicated task realization with map errors by minimal and effective change of off-line plan.

We use the Pioneer P3-DX mobile robot made by Adept MobileRobots (Figure 4). This robot is equipped with two wheels and one caster. This type of robot has no inner ring differences. As a CPU of the embedded PC, Pentium III 850 MHz with 256 MB memory is utilized.

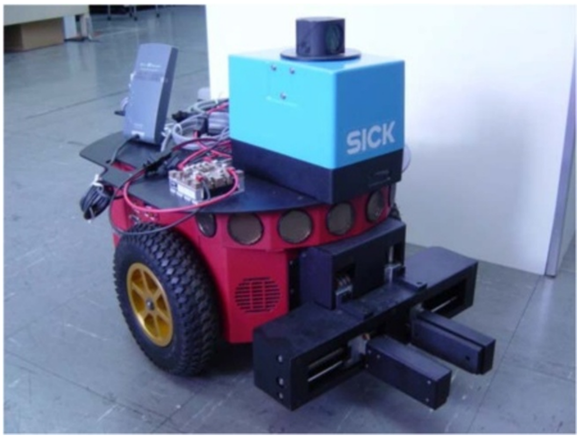


Figure 4 The robot used in the experiments.

There are two ways for robots to obtain information about a working environment. In the first method, multiple sensors are placed in the working environment, and the robot receives data from these sensors. In the second method, the robot is equipped with sensors that measure the working environment. We adopt the latter method from the viewpoint of setup cost and the ability to make measurements over a large area. Numerous sensors can be mounted on a mobile robot, including touch sensors, ultrasonic sensors, etc. Among these, we adopt a laser range finder because this sensor has a high accuracy and is easy to use to collect sensor data. We use the NAV200 laser range finder (the box on the robot in Figure 3) made by SICK. This sensor can know the position of reflectors. The accuracy of the sensor is 5.7×10^{-3} [m] with position and 1.9×10^{-3} [rad] with orientation when the measuring distance is within 2 [m]. The time for sensing is less than 5.0×10^{-1} [s] for one measurement.

Cylindrically-shaped objects are utilized for movable objects, Here, each has a radius of 1.5×10^{-3} m and equipped with a reflector. The robots can grasp the objects from any direction. Self-localization and environmental recognition are realized by using NAV200 and reflectors [11]. Marks are attached on all the walls, which are only obstacles in the working environment. One mark is consisted of two reflectors with constant small distance, as shown in Figure 5. Because the distance between the two reflectors is given to the robots in advance, the robots can distinguish movable objects (cylindrically-shaped objects) and obstacles (walls) if they see reflectors. Because two reflectors consist one mark, relative position and orientation of one mark from the robots can be measured. The process of self-localization and environmental recognition is shown in Figure 6. First, the robot detects surrounding reflectors as shown in Figure 6(a). Position and orientation of all marks (attached on the walls) in the global coordinate are represented in the given map as shown in Figure 6(b). The robot can realize self-localization and position of objects as shown in Figure 6(c). When the robot measure more than one mark at the same time, the measured value is averaged. As for transformation, we assume that at least three walls and movable objects in total can be seen from one robot at the same time, and assume that at most one wall moves at the same time. If the error of self-localization of the robot is larger than the accuracy of the sensor, re-plan module A is activated by assuming there is self-localization error. When position of a certain wall is changed, the robot can recognize it by checking relative position and orientation of the marks attached on the walls, and reflectors on the objects. In the case of Figure 6, if position and orientation of one mark (one wall) is changed in the case of Figure 6(c), we can recognize it by comparing the measured information about other marks or reflectors of objects.

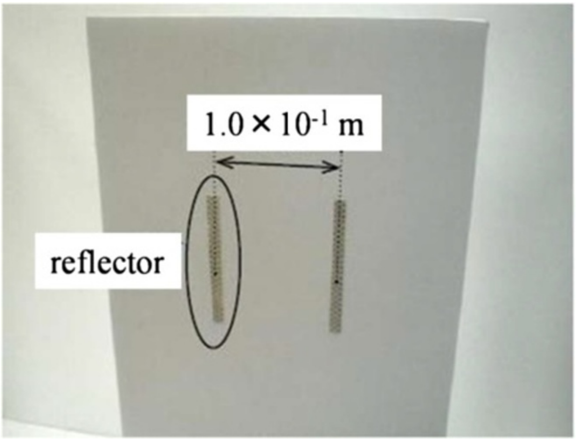


Figure 5 The mark attached on the wall.

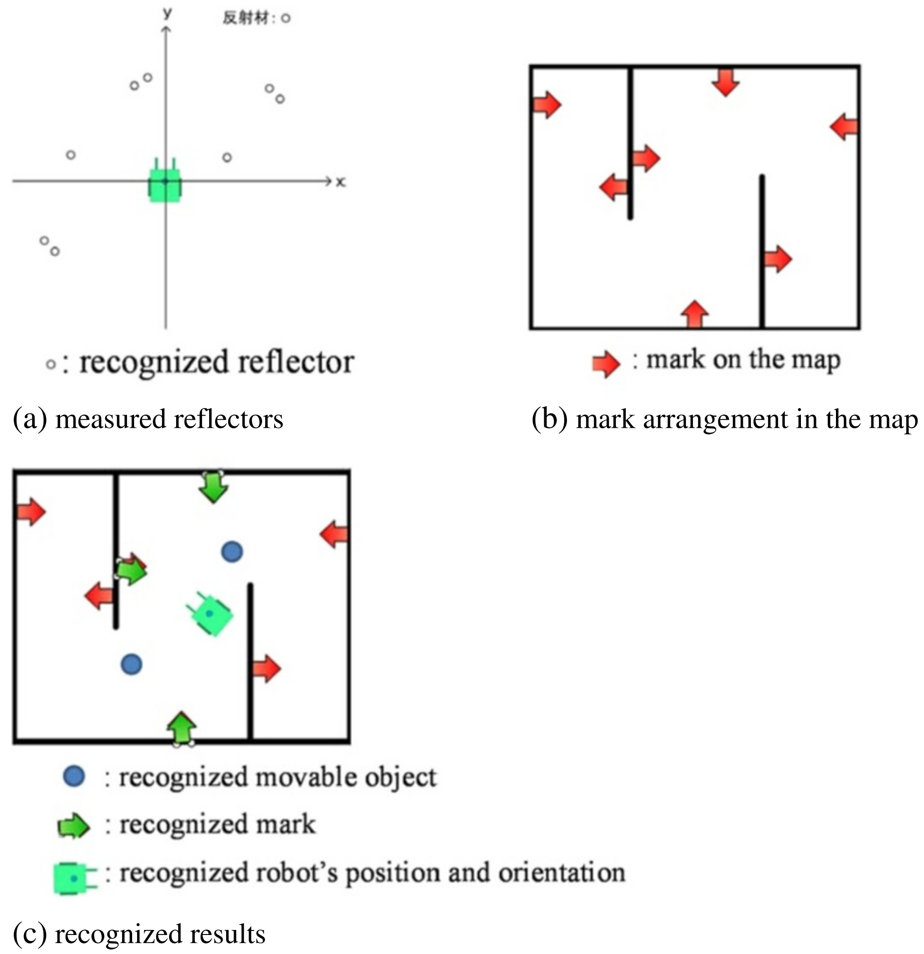
Results and discussion

Analysis of calculation cost

To accurately evaluate our proposed method, we discuss the calculation costs for the re-planning modules.

The calculation cost for checking module for activating re-plan module A depends on the number of reflectors that the robots actually detect, which is influenced by the total number of reflectors that are set in the working environment. Here, the total number of reflectors is indicated by l, and we assume that a robot can detect all of the reflectors in the working environment. The calculation cost to execute one module is $O(l^5)$, with $O(l^2)$ needed to calculate the positions of reflectors from the position data of the reflectors and $O(l^3)$ to calculate a robot's position from the positions of the reflectors (determined from the results of [11]). In the experiment, calculation cost of this module is very small in comparison to those of main re-plan modules B and C. This is because the number of measurable reflectors is not so large (less than 10) in the experiment. The calculation cost of main re-plan module A, checking modules of B and C are also very small.

The calculation cost for the re-plan module B depends on the number of degrees of freedom and the slices of each axis. Here, we define m as the number of slices of each axis. The calculation cost is $O(m^3)$ in the case of the P3-DX mobile robot (determined from [21]). The actual number of operations can increase depending on the method for detecting collisions between robots and obstacles. However, we use a 3[m] * 3[m]-sized environment, and the size of each cell in a robot's configuration space is 5.0×10^{-2}[m] * 5.0×10^{-2} [m] * π/8[rad] in the experiments performed in a real


y
反射材：○
x
○ : recognized reflector
: mark on the map
: recognized movable object
: recognized mark
: recognized robot's position and orientation

(a) measured reflectors

(b) mark arrangement in the map

(c) recognized results

Figure 6 An example of mark recognition. (a) indicates measured position of reflectors, **(b)** indicates mark arrangement in the map of the robot, and **(c)** indicates recognized result of robot's position and orientation.

environment. From these values, module B can also be executed in a short time.

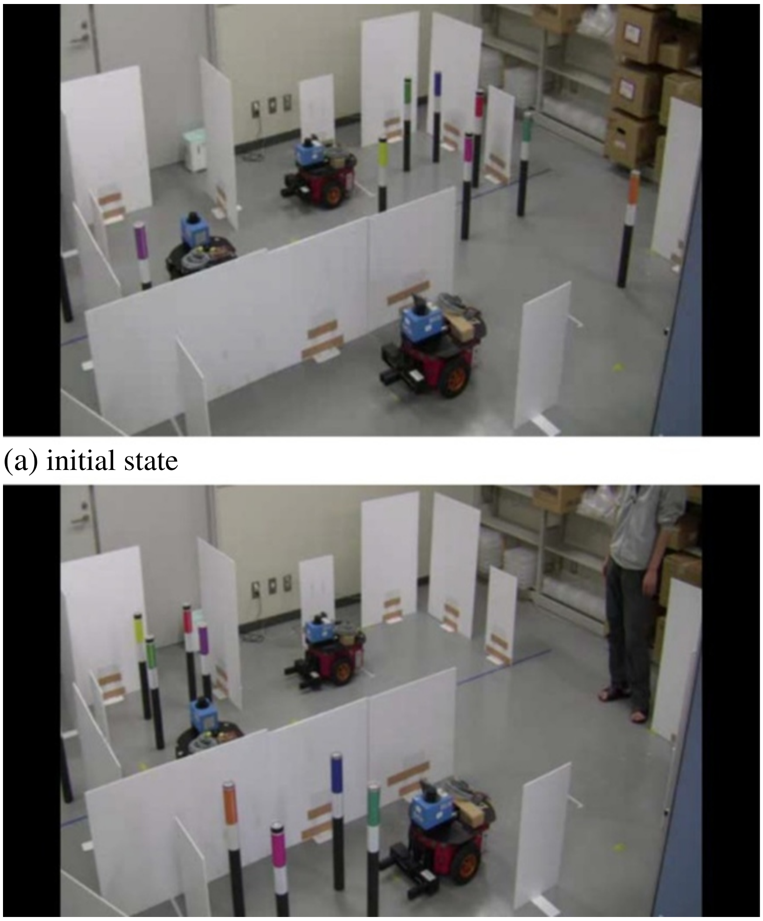
In the first half of the re-plan module C, a robot iterates to estimate the entire task time when a new delivery position is set to each position. We use part of the algorithm for the Critical Path Method [22] to estimate the entire task time for a new delivery position. The calculation cost for one estimation is $O((n\text{-}1)^2p^2)$, where n is the number of robots and p is the number of objects. Therefore, the calculation cost for re-planning one new delivery position is $O(m^3(n\text{-}1)^2p^2)$. If a robot re-plans all of the delivery positions, the calculation cost is $O(m^3(n\text{-}1)^3p^3)$. Comparing these values makes it clear that our re-planning module is fast from the viewpoint of order representation.

Experiments in a real environment

We conduct experiments in a real environment in order to evaluate our method of considering the calculation cost that is not described earlier and the influence of communication time and sensing time. Experiments are conducted in the working environment shown in Figure 7 and involve 3 robots and 8 objects (cylinders). Each experiment is conducted twice. In experiment 1, the shape of the obstacles (position and orientation of walls) does not change. In experiment 2, the objects are transformed so that the predefined delivery positions cannot be used.

The results of experiment 1 are shown in Figure 8. In these results, the robots complete all their transportations. Lines with arrows in Figure 8 show rough trajectories of robots. They modify their moving directions by localization (re-plan module A explained in Section Details of modules and real robots) many times and do not re-plan their paths and delivery positions. It costs about 260 seconds to complete all of the transportations. Position accuracy of the robots is about 5.0×10^{-2} [m] in average, which can be absorbed with localization module (the re-plan module A) of the robots.

The results indicate that our proposed method can cope with map errors. The entire task time increases



(a) initial state

(b) goal state

Figure 7 Setting of the experiments in a real environment. (a) indicates the initial state, **(b)** indicates the goal state.

compared with the off-line task plan. One reason for this difference is that the planner does not consider the costs for grasping objects. Except for this, task realization effectiveness is maintained. From the above discussion, our proposed method may effectively realize a rearrangement task in a case where map errors are insignificant.

We next show the results of experiment 2 in Figure 9. In this experiment, a human worker adds, moves or removes a wall as shown in Figure 9(a), (d), (f) and (j). The robots detect this transformation by measuring the position of the marks attached on walls. The robots can fulfil the overall task. In this case, each robot calculates checking routines for re-plan modules A, B, and C. The robots re-plan the delivery positions in two cases shown in Figure 9(a) and (f). The robots judges that there is no need for activating re-plan module C for the environmental changes shown in Figure 9(d) and (j). As a result of applying the re-planning module C, the robots transport objects to different delivery positions, as shown in Figure 9(c) and (i). Main routine for re-plan module C is conducted twice, with a calculation time for one re-planning of about 40 seconds. Re-plan module A is activated many times. On the other hand, there is no chance to activate re-plan module B in this experiment. This result indicates the proposed algorithm activate re-plan modules only when it is necessary.

The robots avoid collisions by re-planning new delivery positions. This indicates that our proposed method can be applied to a wider variety of situations than a simple method that does not execute re-planning. Cares should be taken that the algorithm in [11] cannot solve this kind of problem. The entire task time increases by about 80 seconds compared to the time estimated by the pre-developed task plan. The total calculation time for re-planning is also 80 seconds (40 × 2). Therefore, our proposed method can maintain the effectiveness of the task plan itself. The calculation time for one re-planning module is within one minute. From these results, our method has sufficient performance to fulfill our objective.

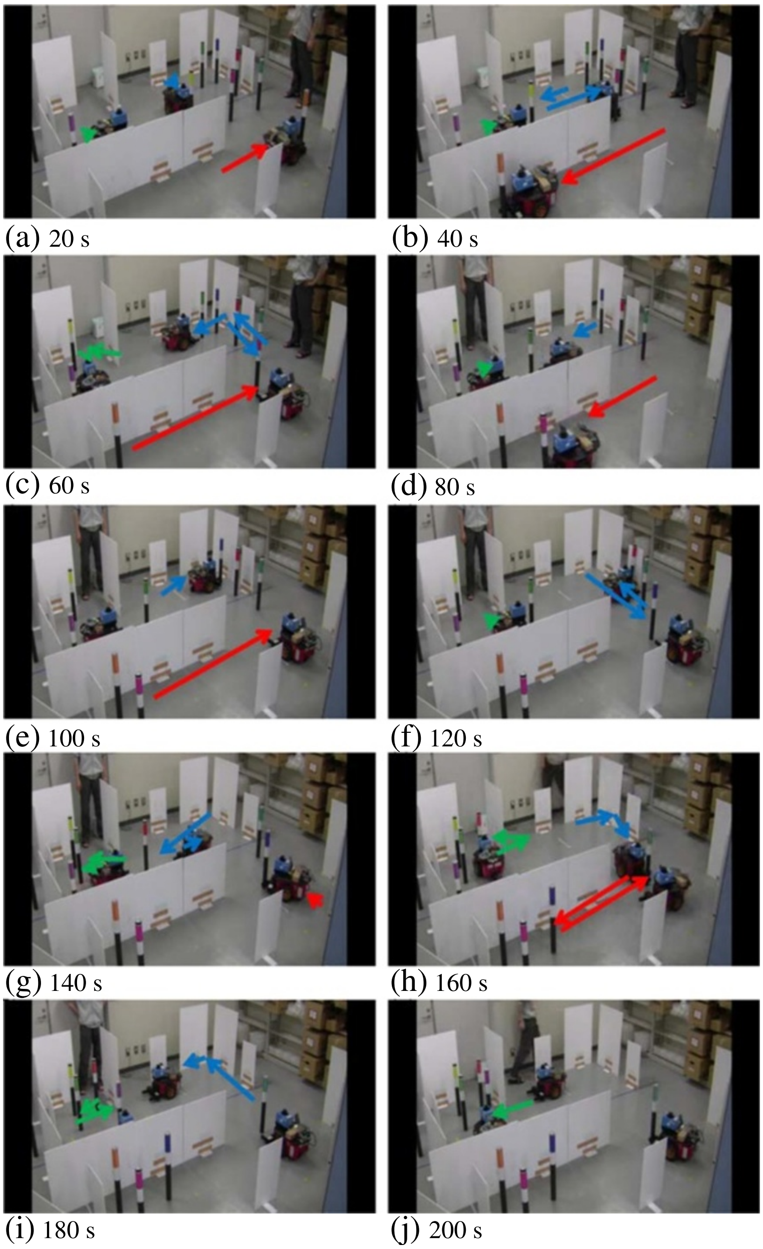


Figure 8 Results of experiment 1 (with static environment). (a), **(b)**, **(c)**, **(d)**, **(e)**, **(f)**, **(g)**, **(h)**, **(i)** and **(j)** show the states at 20, 40, 60, 80, 100, 120, 140, 160, 180 and 200 second, respectively.

Conclusion

A rearrangement task by multiple robots is a fundamental task that is involved in various applications. It is very complicated with constraints involving transportation order and scarcity of spatial resources. Therefore, it is difficult to design modules that cope with map errors and maintain task realization effectiveness.

Based on a problem-partitioning structure, we designed rearrangement task realization that involve (a) design of three modules in consideration of characteristics of map errors and structure of the plan result, and (b) design of when they should be activated with the combination of checking routines and main re-planning routines. In addition to the proposition of a realization procedure, we constructed real robot systems. The proposed method and systems were tested by experiments in actual environments including transformation of obstacles. The results showed the effectiveness of our method.

Contribution of this study is to realize such complicated tasks for multiple mobile robot systems in a real

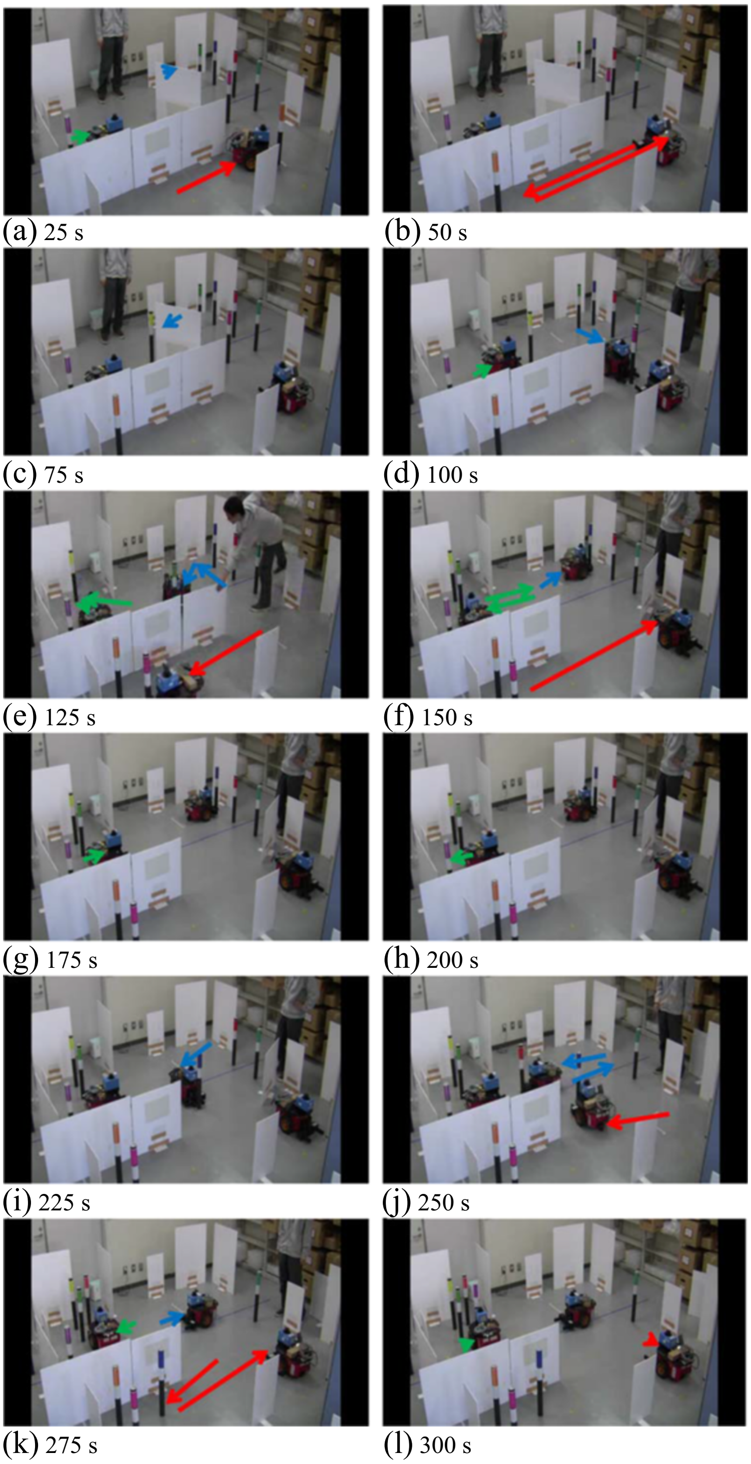


Figure 9 Results of experiment 2 (with transportation of walls). **(a)**, **(b)**, **(c)**, **(d)**, **(e)**, **(f)**, **(g)**, **(h)**, **(i)**, **(j)**, **(k)** and **(l)** show the states at 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275 and 300 second, respectively.

world. The proposed algorithm makes it possible to realize the task by changing the off-line plan as small as possible, with three modules designed from the structure of planning result, activating based on recognized situation.

In future work, we hope to develop a simple index that represents map errors. Such an index can be used to decide what kind of module should be applied to specific map errors.

Competing interests

The authors declare that they have no competing interests.

Authors' contribution

NF carried out the algorithm design and experiments, and drafted the manuscript. TO participated in data analysis of the obtained results. TH participated in discussion about the obtained results. JO helped to draft the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This work was partially supported by the Yazaki Memorial Foundation for Science & Technology and was supported through the Global COE Program, "Global Center of Excellence for Mechanical Systems Innovation", by the Ministry of Education, Culture, Sports, Science, and Technology.

Received: 10 June 2014 Accepted: 7 October 2014
Published online: 25 October 2014

References

1. Nikaido M, Kobayashi T, Ito T, Higashi T, Tamura H, Ota J (2009) Conceptual warehouse-design algorithm using network flow model. Adv Robot 23(6):705–724
2. Winfield AFT (2009) Foraging robots. In: Meyers RA (ed) Encyclopedia of Complexity and Systems Science. Springer, New York
3. Drogoul A, Ferber J (1992) From Tom thumb to the dockers: some experiments with foraging robots. In: Proceedings of the Second International Conference on Simulation of Adaptive Behavior., pp 451–459
4. Russell SJ, Norvig P (2002) Artificial intelligence, A Modern Approach, 2nd edn. Prentice Hall, New Jersey
5. Hopcroft JE, Schwartz JT, Sharir M (1984) On the complexity of motion planning for multiple independent object; PSPACE-hardness of the "Warehouseman's problem,". Int J Robot Res 3(4):76–88
6. Fujii N, Inoue R, Takebe Y, Ota J (2010) Multiple robot rearrangement planning using a territorial approach and an extended Project Scheduling Problem Solver. Adv Robot 24(1–2):103–122
7. Fujii N, Ota J (2011) Territorial and effective task decomposition for rearrangement planning of multiple objects by multiple mobile robots. Adv Robot 25(1):47–74
8. Cherif M, Vidal M (1998) Planning handling operations in changing industrial plants. In: Proc IEEE Int Confe Robotics and Automat., pp 881–886
9. Cambon S, Alami R, Grabot F (2009) A hybrid approach to intricate motion, manipulation and task planning. Int J Robot Res 28(1):104–126
10. Yamashita A, Arai T, Ota J, Asama H (2003) Motion planning of multiple mobile robots for cooperative manipulation and transportation, IEEE Trans. Robot Automation 19(2):223–237
11. Takebe Y, Fujii N, Inoue R, Ota J (2009) Realization of rearrangement task by multiple robots. In: ICROS-SICE Int Joint Conf., pp 2C16–1
12. Abe Y, Shikano M, Fukuda T, Arai F, Tanaka Y, Sakamoto S (2000) Vision based working robot based on architecture of control with error recovery. J Japan Society Mech Eng 66(642):514–521, in Japanese
13. Yamazaki K, Tomono M, Tsubouchi T, Yuta S (2006) Motion planning for a mobile manipulator based on joint motions for error recovery. In: Proc 2006 IEEE/RSJ Int Conf Intell Robots and Syst (IROS 2006)., pp 7–12
14. Kurabayashi D, Koga S, Arai T, Ota J (1999) Local path re-planning for unforeseen obstacle avoidance by autonomous sweeping robots. J Robot Soc Japan 17(7):966–973, in Japanese
15. Fontan MS, Mataric MJ (1998) Territorial multi-robot task division. IEEE Trans Robot Automat 14(5):815–822
16. Tewolde GS, Wu C, Wang Y, Sheng W (2008) Distributed multi-robot work load partition in manufacturing automation. In: IEEE Proc Int Conf Automat Sci and Engg., pp 504–509
17. Richer TJ, Corbett DR (2004) A dynamic territorial robotic system. In: Proc Australian Conf on Robotics and Automat
18. Alami R, Laumond JP, Simeon T (1995) Two manipulation planning algorithms. In: Goldberg K, Halpern D, Latombe J-C, Wolson R (eds) Algorithmic Foundation of Robotics. A. K. Peters, Boston, pp 109–125
19. Fujii N, Ota J (2008) Rearrangement task realization by multiple mobile robots with efficient calculation of task constraints. Adv Robot 22(2):191–213
20. Hartmann S (2001) Project scheduling with multiple modes: a genetic algorithm. Annals Oper Res 102(1–4):111–135
21. Konolige K (2000) A gradient method for realtime robot control. In: Proc. 2000 IEEE/RSJ Int Conf Intel Robotic Syst (IROS2000)., pp 639–646
22. O'Brien J (1965) CPM in Construction Management: Scheduling by the Critical Path Method. McGraw-Hill, New York; London

doi:10.1186/s40648-014-0016-5
Cite this article as: Fujii *et al.*: **Realization method for a rearrangement task by multiple mobile robots in consideration of map errors.** *ROBOMECH Journal* 2014 **1**:16.