

*Full paper*

## **Realizing the exploration and rearrangement of multiple unknown objects by an actual mobile robot**

YUSUKE FUKAZAWA, TREVAI CHOMCHANA, JUN OTA \*, HIDEO YUASA,  
TAMIO ARAI, HAJIME ASAMA and KUNIAKI KAWABATA

*Department of Precision Engineering, School of Engineering, The University of Tokyo,  
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan*

Received 13 November 2003; accepted 24 March 2004

**Abstract**—This paper offers a proposal for realizing the exploration and rearrangement of multiple unknown objects that lay scattered in working environments. The objective of the exploration task is to find all the objects in the environments. On the other hand, the objective of the rearrangement task is to carry all the objects to their goal position. Many applications are possible if the exploration and rearrangement tasks are combined. Some of them are cleaning, mine detecting and housework. An algorithm that integrates two tasks is presented with respect to the effectiveness of the path length and computational cost. In addition, an exploration algorithm is proposed that can work well in an environment that has many objects. In order to verify the algorithm, experiments are conducted with an actual robot. In the experimentals, an environmental recognition method is developed by attaching a mark to the objects. The robot recognizes the objects by finding the mark. It then obtains information from the mark. The mark is also used to modify the odometry error of the robot by computing its configuration relative to a mark attached to a wall. The success rate of this experiment was almost 80% in 20 trials.

**Keywords:** Exploration task; rearrangement task; mobile robot; multiple objects.

### **1. INTRODUCTION**

Planning motion so that a mobile robot can explore and rearrange multiple unknown objects is challenging. Both exploration and rearrangement tasks have multiple applications. An exploration task is useful for watching a building for security or cleaning tasks. On the other hand, a rearrangement task has other applications, such as the conveyance of objects in warehouses or factories as well as homes. If the exploration and rearrangement tasks are integrated, more applications are expected, such as cleaning, mine detecting and housework. The concept of exploration and

---

\*To whom correspondence should be addressed. E-mail: [ota@prince.pe.u-tokyo.ac.jp](mailto:ota@prince.pe.u-tokyo.ac.jp)

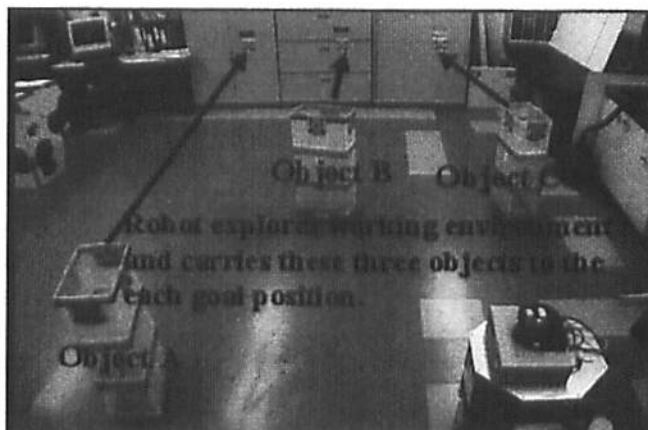


Figure 1. Working environment.

rearrangement tasks is shown in Fig. 1. Three objects, that a robot does not know in advance, will be identified and carried to their own goal configurations as soon as possible. Our objective is to realize exploration and rearrangement tasks of multiple unknown objects.

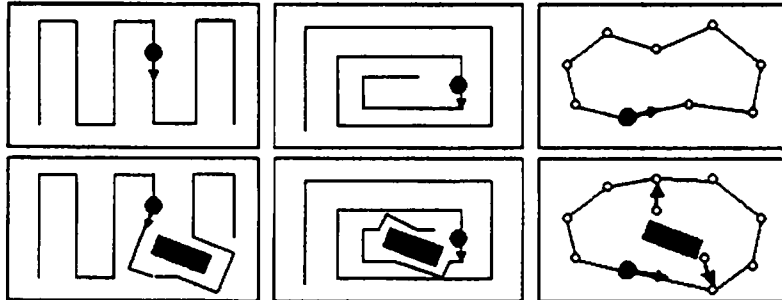
### 1.1. Past research

Past approaches to exploration and rearrangement tasks may be roughly divided into three parts.

*1.1.1. Exploration task with multiple unknown objects.* Generally, exploration tasks have been based on two types of path generation [1]: zigzag [2–4] and spiral [5]. The concepts of both the zigzag and spiral paths are shown in Fig. 2. In both algorithms, the exploration path is generated by combining line segments arranged in a working environment.

Both the zigzag path and the spiral path are widely used for room cleaning or floor coating because they are easy to plan. However, in an environment with many unanticipated objects, we often re-plan the new path because of a newly found object. As a result, new paths must be created that detour around the objects in order to re-plan the path, as shown in Fig. 2. These changes require an excess of computational cost or an excess of additional path length.

*1.1.2. Rearrangement task with multiple known objects.* Ben-Shahar *et al.* [6] have used an actual robot to accomplish rearrangement tasks with multiple objects. Andreas [7] works out the problem (SOKOBAN-problem) that robot pushes an object to the goal area under the grid environment. In order to minimize the number of times that a robot must push an object to a goal area, Andreas uses ‘Iterative Deepening A\*’ (IDA\*) [8], which is a kind of heuristics search methodology. However, in both cases cited above, the researchers assume that the locations and



**Figure 2.** Overview of each exploration path algorithm: zigzag (left), spiral (center) and point-distributing (right).

shapes of all objects encountered are known. In addition, the encounters often end in a deadlock because the only option a robot has is to push.

*1.1.3. Rearrangement task with multiple unknown objects.* Ishiguro *et al.* [9] try to control a garbage-collecting robot by referring to an immune system. The robot executes simple actions such as ‘avoidance of obstacle’, ‘holding object’ and ‘exploration of object’. In this research, the sensor input works as the antigen and the selected action attempts to neutralize the antigen (which works as the antibody). Consequently, an appropriate action is selected for a given situation. Pfeifer *et al.* [10] developed a self-sufficient garbage-collecting robot by constructing a neural network that connects sensory inputs with motor outputs (Extended Braitenberg Architecture (EBA)). However, these approaches are all based on the concept called random walk; therefore, it is not clear whether the selected action is the best one. In addition, the above approaches do not guarantee the accomplishment of the task.

Ota [11] executed a rearrangement task of multiple objects whose positions and shapes were unknown using ‘Learning Real-Time A\*’ (LRTA\*) [12], which is a kind of real-time search methodology. However, there were two problems. One is that some of the objects were not found because the robot was not designed to carry out exploration tasks. The other is that it is not based on experiments, but only simulations.

## 1.2. Challenges and their solutions

As can be seen from the above review, the problem that a robot explores and rearranges multiple objects has not been solved. In this paper, we have following challenges.

### 1.2.1. Challenges.

- (i) *Combining exploration and rearrangement tasks.* Apparently, this challenge is easily solved by joining the rearrangement task under a known environment to an exploration task under an unknown environment, as shown in Fig. 3. In

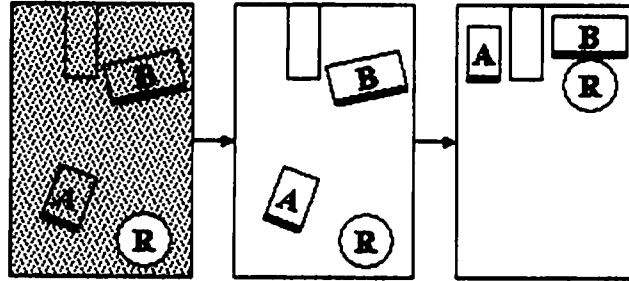


Figure 3. Past approaches.

Fig. 3, at the beginning, a robot executes the exploration task in an unknown environment in order to identify any objects in the operating field; the robot then executes a rearrangement task of the known objects. However, this approach has various short-comings, such as high computational cost and ineffective path generation caused by repeated trips to the same object in each trial during the exploration and rearrangement tasks. Therefore, it is necessary to develop a method that will integrate the tasks so that both the computational cost and path length can be smaller.

- (ii) *Exploration paths that can be repeatedly revised.* When embarking on an exploration task, we need an exploration algorithm that can be repeatedly revised, because we assume that there are many unknown objects in the working environment. In such environments, robots must be equipped to alter their paths, sometimes repeatedly, when they encounter unanticipated objects.
- (iii) *A successful system that would help robots recover from odometry errors and recognize objects in the operational field must be developed.* A method for recognizing objects within the environment must be developed in order to conduct successful experiments with an actual mobile robot. That is because when the robot finds a new object, the robot must be able to obtain the information necessary for the rearrangement task, such as goal configuration of objects, shape of objects and holding configuration of objects. In addition, any odometry errors, which are a universal problem with mobile robots, must be solved.

**1.2.2. Solutions.** We deal with the first challenge (i) by alternatively repeating the exploration and rearrangement tasks. With respect to the second challenge (ii), the exploration task is accomplished by extending the point-distributing exploration algorithm. With respect to the third challenge (iii), a mark that contains information about an object is attached to an object or a wall in order for a robot to find an object or modify an odometry error.

### 1.3. Preconditions

The following assumptions are set.

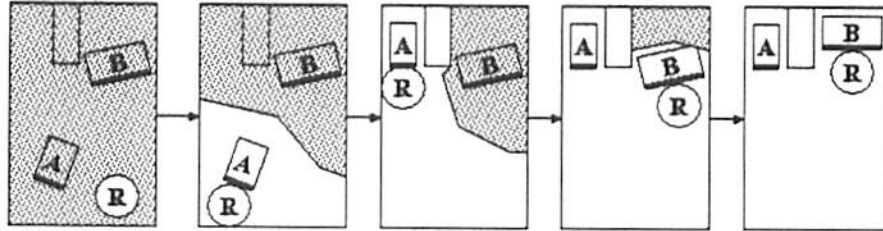


Figure 4. Proposed method.

- The shape of the environment is known.
- The sensing area ( $S_{\text{sen}}$ ) of a mobile robot is described as a circle with a radius  $R_{\text{sen}}$ .
- A mobile robot can obtain the information of an object if a part of it is within the sensing area.
- The robot finalizes the task when both the exploration and rearrangement tasks have been completed.
- The objective of the exploration task is to recognize all the objects as soon as possible.
- The objective of the rearrangement task is to carry all the objects to each goal position as soon as possible.

## 2. METHOD OF INTEGRATION OF THE EXPLORATION AND REARRANGEMENT TASKS

In this section, the exploration and rearrangement tasks are integrated. The concept of integration is shown in Fig. 4. In Fig. 4, the robot executes an exploration task to find an object at the beginning. If object A is found, the robot will execute the rearrangement task of this object. After the robot moves object A, it will return to the exploration task. By repeating this procedure, the robot accomplishes the exploration and rearrangement tasks. The flow chart of the proposed method is shown in Fig. 5. The robot executes the rearrangement task when it identifies objects that have not been rearranged. On the other hand, the robot executes an exploration task when it realizes that no objects are to be rearranged except for those that have already been rearranged. More details are provided about the exploration and rearrangement tasks in Sections 3 and 4, respectively.

## 3. EXPLORATION TASK

To deal with the second challenge (ii), a point-distributing algorithm [13], which we developed, was extended. In this algorithm, first, sensing points are distributed throughout the working environment for the robot to recognize its surround-

ings; then, the exploration path, which connects all of the sensing points, is generated [13]. Figure 2 (right) shows an overview of the generation algorithm for the point-distribution path. By using the point-distributing algorithm, the path can be easily revised by moving a few of the sensing points when the robot encounters a new object. However, if there are many unexpected objects, any convergence will hardly be achieved by re-planning of the path because the movement of the sensing points is based on the distance between the sensing point and the object [13]. To deal with this problem, we propose a method in which sensing points move according to the sensing area to be covered by them.

The method proposed here consists of two types of planning: initial path planning and re-planning of the path. More details about each type are presented in the following section.

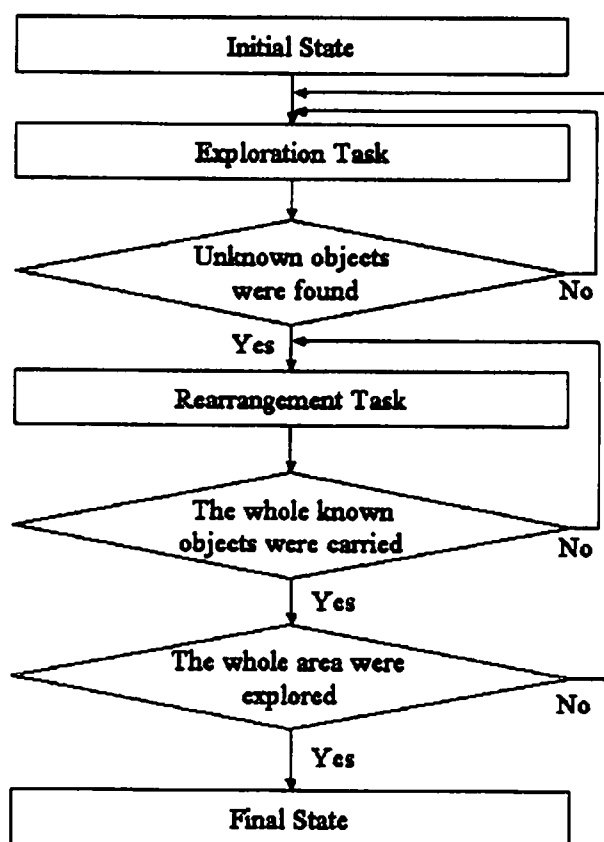


Figure 5. Flow chart of proposed method which integrates the exploration task and rearrangement task.

### 3.1. Initial exploration path planning

A flow chart of the proposed method is shown in Fig. 6. In the following section, more details are provided about moving and adding sensing points. First, a potential function was designed that evaluates the arrangement of the sensing points and describes the movement of the sensing points according to the potential function. In this section, we do not consider positioning and sensing errors of the robot. Measures for these errors in real robot systems will be discussed in Section 6.3.

**3.1.1. Design of the potential function.** In this section, the design of the potential function, which evaluates the arrangement of sensing points, is described. An ideal arrangement, which consists of the smallest number of sensing points, was designed. This kind of arrangement is called hexagonal close packing (HCP) of the sensing points (Fig. 7). Under the arrangement of sensing points with HCP, the area covered by each of the sensing points becomes a regular hexagon inscribed by a circle with a radius of  $r_s$ . Accordingly, the ideal covering area  $S_0$  of a sensing point is described as:

$$S_0 = \frac{3\sqrt{3}}{2} R_{sen}^2 \tag{1}$$

Next, the covering area  $S_u$  of sensing point  $u$  is defined. First of all, the set of grid points  $G$  within  $S_u$  is defined as (2).

$$G = \{g | l(g, u) < R_{sen}, l(g, u) < l(g, v), \forall v \neq u\} \tag{2}$$

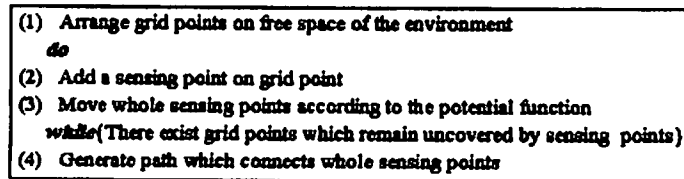


Figure 6. Flow chart of the proposed method.

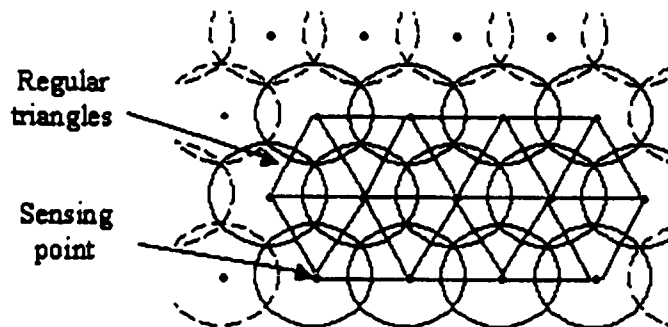


Figure 7. Hexagonal closed packing of sensing points.

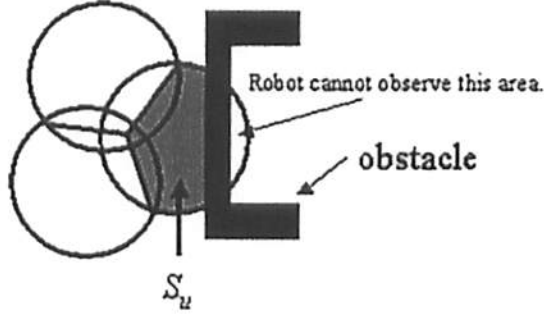


Figure 8. Covering area of the sensing point.

Here,  $l(g, u)$  represents the distance between the grid point  $g$  and the sensing point  $u$ . However, as shown in Fig. 8, there is an area that the robot cannot recognize because of an obstacle, even when the area is within the sensing field. The area is removed by setting  $l(g, u)$ , which intersects with an obstacle that is larger than  $R_{\text{sen}}$ . Sensing point  $v$ , whose distance to sensing point  $u$  is within  $2 \times r_s$ , is defined as a neighboring sensing point of  $u$ . Using the above definition,  $S_u$  is defined as:

$$S_u \approx n(G) \times d^2, \quad (3)$$

where  $n(G)$  and  $d^2$  represent the total number of set  $G$  and the area per grid point, respectively.  $d$  equals the distance between grid points. Finally, the potential function  $f(u)$  is defined as:

$$f(u) = |S_u - S_0|. \quad (4)$$

The smaller the function  $f(u)$  is, the closer to ideal is the arrangement of the sensing points.

Sensing points are moved so that the  $f(u)$  of each sensing point is small and uniform. Equations (5) and (6) are used as an indicator of ‘small’ and ‘uniform’, respectively. In order to evaluate the arrangement of sensing points, we design the potential function  $W(f)$  as:

$$W_0(f) = \sum_{u \in V} f(u)^2, \quad (5)$$

$$W_1(df) = \sum_{u \in V} \sum_{v \in V(u)} |f(v) - f(u)|^2, \quad (6)$$

$$W(f) = W_0(f) + W_1(df). \quad (7)$$

Here,  $V$  and  $V(u)$  represent all the set of sensing points and the set of sensing points  $v$  neighboring  $u$ , respectively.  $df$  represents the gradient of  $f$ .

**3.1.2. Movement of a sensing point.** First, the moving direction of a sensing point is obtained. The moving direction is restricted to the eight surrounding grid points  $u_k (k = 0, 1, \dots, 7)$ . At each  $u_k (k = 0, 1, \dots, 7)$ , the potential value  $f(u_k)$  is



obtained. The sensing point moves to the direction of  $u_k$ , whose  $f(u_k)$  is minimum. The sensing point  $u$  continues to move until no neighboring  $f(u_k)$  are smaller than  $f_u$ , as in:

$$\min\{f(u_0), f(u_1), \dots, f(u_7)\} > f(u). \quad (8)$$

Next, the moving distance  $D(u)$  is obtained.  $D(u)$  is designed so that it will be proportional to the differential of  $f(u)$  against time. The differential of  $f(u)$  is also designed so that it is proportional to a gradient of the potential function  $W(f)$ , as in:

$$\begin{aligned} \frac{\partial f}{\partial t}(u) &= -\frac{\delta W(f)}{\delta f}(u) \\ &= -2f(u) - 4 \sum_{v \in V(u)} (f(u) - f(v)). \end{aligned} \quad (9)$$

Equation (9) is modeled as the Reaction-Diffusion Equation on a graph [14]. Therefore,  $D(u)$  can be described using the magnitude of the moving direction acquired as:

$$D(u) = \frac{\partial f}{\partial t}(u) \times \frac{1}{\|f(u) - \min\{f(u_0), f(u_1), \dots, f(u_7)\}\|}. \quad (10)$$

**3.1.3. Addition of sensing points.** After moving all of the sensing points, according to the previous section, the coverage status of all of the grid points is checked. If there are grid points that have not yet been covered by an area of sensing points, a sensing point is added to the grid points at the greatest distance from the entire arrangement of sensing points. After the sensing point is added, all of sensing points, including the new one, are moved. This procedure, which repeats the movement and addition of sensing points, lasts until the sensing area of sensing points covers the entire group of grid points.

**3.1.4. Path generation.** After the arrangement of the sensing points is completed, the shortest exploration path is generated by connecting all of the sensing points for a mobile robot to go around. The path generation problem can be modeled as the Traveling Salesman Problem (TSP) [15], which has been proven to be NP-hard (non-deterministic polynomial-time-hard) [16]. A suboptimal path is generated using the Simulated Annealing (SA) method [15]. To be able to generate a path that can prevent a robot from detouring, the path length that intersects obstacle is set to be infinite.

## 3.2. Re-planning the exploration path

**3.2.1. Rearrangement of sensing points.** When a robot revises its path, a new working area must be defined. A new working area is defined by excluding the already recognized and overlapped areas with newly found objects. These areas can

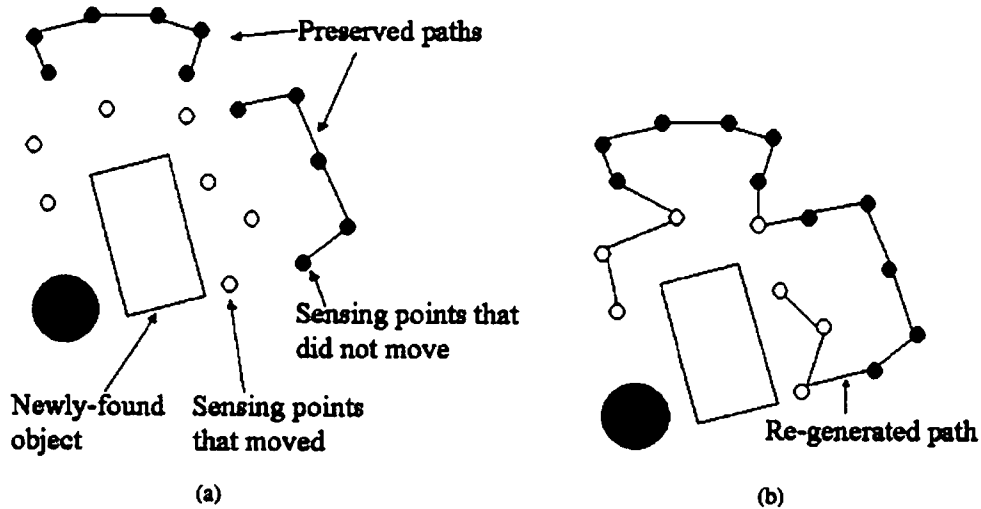


Figure 9. Flow of re-generation of the path. (a) The movement of sensing points. (b) Re-generated path.

be excluded by removing grid points. Sensing points are rearranged on the newly defined working area.

The rearrangement of sensing points can be completed easily by moving a few sensing points that are close to the boundary of an excluded area. That is because the magnitude of a sensing point's movement is diffused from the boundary of an excluded area, according to the Reaction-Diffusion Equation on a graph (Fig. 9a).

**3.2.2. Regeneration of the exploration path.** A previously generated path is used for regenerating the path as follows. First, the path is cut off between a moved sensing point and one that was not, and separated paths are preserved (Fig. 9a). Next, the shortest path is obtained, which connects preserved paths and moved sensing points (Fig. 9a). This kind of problem has been modeled as the Rural Chinese Postman Problem (RPP) [17], which is proven to be NP-hard. The RPP can be transformed to the TSP by transforming all edges into nodes [18]. Hence, TSP algorithms can be used to solve the RPP. In this paper, the SA method is used to solve the TSP.

#### 4. REARRANGEMENT TASK [11]

The objective of the rearrangement task is to move entire objects to each goal position as soon as possible. Ota [11] proposed an algorithm that achieves the rearrangement of multiple objects by using a real-time heuristics search methodology. This algorithm is adopted here for the rearrangement task. The following is an outline of the algorithm.

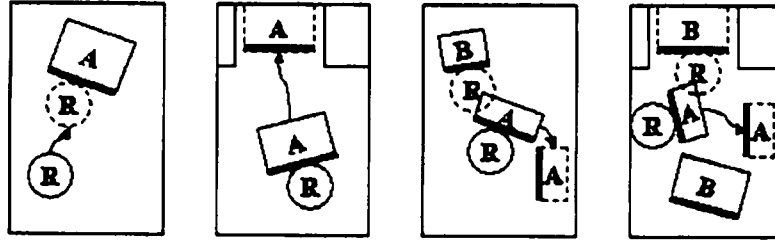


Figure 10. Four primitive motions.

#### 4.1. Four primitive motions

In this algorithm, four primitive motions are defined as follows (Fig. 10):

- (i) Motion to reach the grasping configuration of object A.
- (ii) Motion to transfer object B to its goal position.
- (iii) Motion to carry object A to the intermediate configuration so that the robot can reach the grasping configuration of object B.
- (iv) Motion to carry object A to a temporal intermediate configuration to carry object B to its destination.

#### 4.2. Selection of primitive motions

In this algorithm, the robot repeats the following steps to acquire an appropriate primitive motion:

- (i) List the entire executable primitive motions  $a$ . For each primitive motion, calculate  $f(x) = k(x, a) + h(x')$  at the current state  $x$ , where  $x'$  is the next state of  $x$  when a robot takes action  $a$ ,  $h(x')$  is the current lower bound of the actual cost from  $x'$  to the goal state and  $k(x, a)$  is the cost for action  $a$ .
- (ii) Update the lower bound of the cost for the state  $x$  as follows:

$$h(x) \rightarrow \min_{x'} f(x').$$

- (iii) Select the primitive motion  $a$  moving to neighbor  $x'$ , which has the minimal  $f(x')$  value. Finish the procedure when the robot reaches the goal. Go to Step 1 when the robot reaches at  $x'$  or finds a new object within its sensing area.

#### 4.3. Design of a heuristics function

Ota modeled a heuristics function as the minimum length that connects all the line segments connected between the present configuration and the goal configuration of all the objects. The cost of the line segments corresponds to the path length. Obtaining the minimal length is modeled as Stacker Crane Problem of graph theory [15]. The Stacker Crane Problem is a generalization of the TSP, which connects all the nodes and edges.

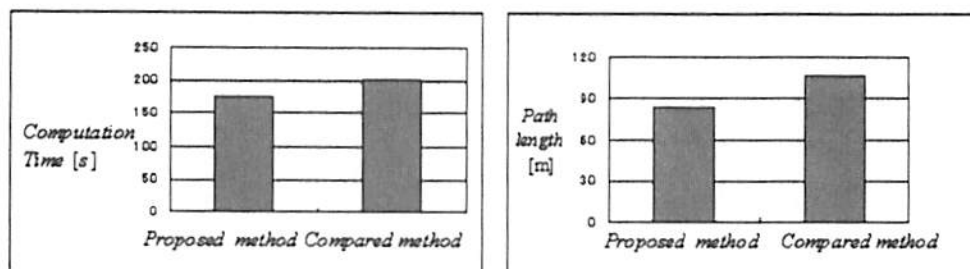


Figure 11. Simulation results: (left) computation time (s) and (right) path length (m).

## 5. SIMULATION OF THE PROPOSED METHOD

In this section, the proposed algorithm is simulated in order to verify its efficacy. As a comparative method, the method explained in Section 1.2.1 is used, i.e. at the beginning, a robot executes the exploration task in an unknown environment in order to identify any objects in the operating field; the robot then executes a rearrangement task of the known objects. The procedures are shown in Fig. 3. The simulations are demonstrated with 10 trials in a  $7 \times 7$  m<sup>2</sup> area in which there are three objects. At every trial, the location of each object is selected at random and the goal position of each object is always the same. The computer specifications are Pentium 4: 1.8 GHz CPU with 512 Mb of memory.

The total computational cost of the proposed method is 14% smaller than that of the comparative method (Fig. 11, left). The total path length of the proposed method is 27% smaller than that of the comparative method (Fig. 11, right). Approaching the same object twice in one exploration and rearrangement trial causes more redundancy than that resulting from using the our proposed method. The effectiveness of the proposed method can be demonstrated from the above.

Calculation cost for the planning is known to be  $O(n_m^4)$ , where  $n_m$  is the number of movable objects [11]. This analysis shows that the proposed method is effective when the number of objects is relatively small.

## 6. EXPERIMENT

### 6.1. Experimental equipment

In this section, experiments with the exploration and rearrangement tasks are reported. The environment is a rectangle  $4 \times 5$  m, and there are three objects whose positions have been selected at random.

The mobile robot used in this experiment has an omni-directional mechanism as a motion system named ZEN (Fig. 12). It was developed by the Institute of Physical and Chemical Research. In order to grasp an object, the robot is equipped with a lift mechanism. As a sensory system, the robot has a CCD camera and eight ultrasonic sensors. In addition, it has a QR decoder that can read a QR code, which is a

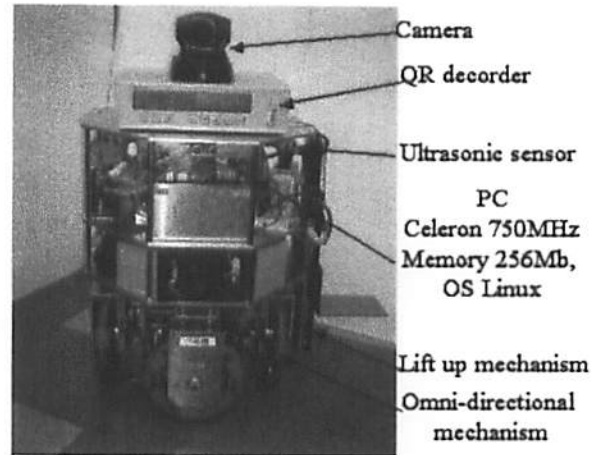


Figure 12. View of the robot 'ZEN'.

kind of two-dimensional bar code. The QR decoder and QR code are products of Keyence Co. and Denso Co., respectively.

### 6.2. Environmental recognition

In order for a robot to recognize objects, a mark is attached [19] to the objects, as shown in Fig. 13. The mark consists of two parts — one for the mark and another for the memory. The mark part is used in order to recognize the object. The color of the mark part is cyan, which is less likely to be misidentified. The memory part is used in order to obtain such information about the objects as shape, grasping configuration and goal position, which are inside the QR code. The flow of environmental recognition is reported in the following.

First, the image of a camera is binarized by the cyan color and the mark is then extracted by contracting the binary image. The image of the mark will be contracted to four points; on the other hand, the other image will be contracted to one or two points. Misidentified images can be eliminated by the above procedure. The relative configuration of an object can be obtained from four contracted points. The mark is zoomed in on the basis of the relative configuration to the mark and the QR code, which has information about the object, is then read by the QR decoder.

Considering the maximal horizontal angle of the camera, which is  $48.8^\circ$ , a robot must rotate a camera in order to cover a sensing area which is defined as a circle. In order to shorten the time needed for rotating the camera, robot observes the surroundings with the use of an ultrasonic sensor at the beginning. It then rotates the camera to the direction in which the ultrasonic sensor reacts.

### 6.3. Recovery from odometry error

Two methods are used to deal with the odometry error of the robot. In the first recovery method, the robot, using the mark attached to the object, modifies its

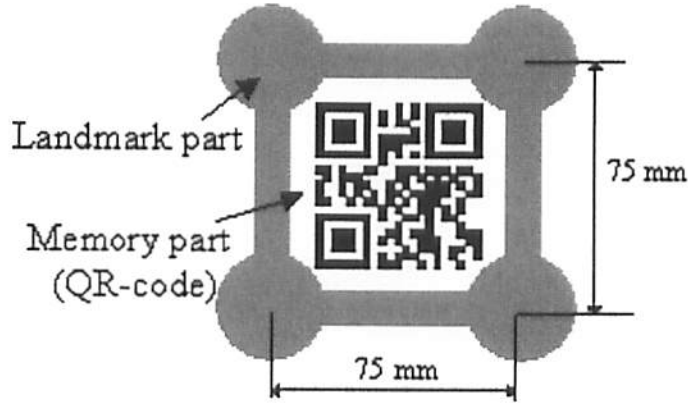


Figure 13. View of the mark.

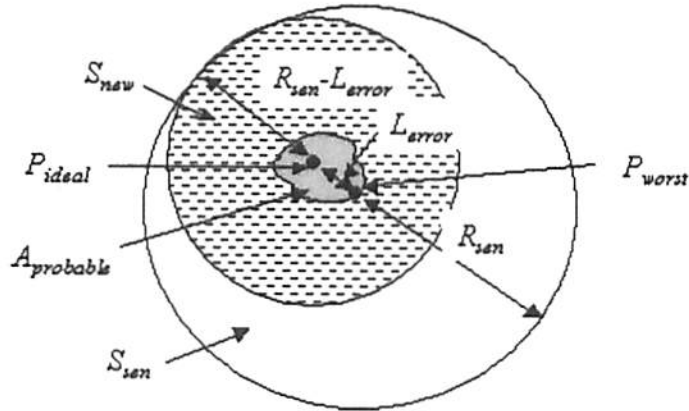


Figure 14. The recognizable area of a robot (newly defined sensing area).

relative position to the object just before grasping it. In addition, immediately before moving the object to its goal position, the robot modifies its position relative to the room by using the mark that is attached to a wall.

In the second recovery method, the new sensing area  $S_{new}$  is designed to be smaller than the actual one; in other words, this designed area is a certainly recognizable area even for a robot that has an odometry error (Fig. 14). First, the probable area  $A_{probable}$  within which the robot can exist is obtained; this area is determined by the motion mechanism, the control mechanism and the sensing mechanism of the mobile robot. The most probable position  $P_{ideal}$  within  $A_{probable}$  is considered to be its ideal position. We also obtain the position  $P_{worst}$ , which is the greatest distant position from  $P_{ideal}$  within  $A_{probable}$ ; we then set the distance between  $P_{ideal}$  and  $P_{worst}$  as  $L_{error}$ .  $S_{new}$  is shown as a circle with a radius  $R_{sen} - L_{error}$ .

The following parameters were obtained from the basic experiment. The actual sensing area of robot was 1.0 m ( $= R_{sen}$ ).  $A_{probable}$  is shown as a circle with a radius of 0.3 m for 6.0 m of movement. Considering the area of the working

environment of this experiment, we can safely say that a robot can localize itself using the mark attached to the wall or the object within 6.0 m of movement. Therefore,  $L_{\text{error}}$  becomes 0.3 m.  $S_{\text{new}}$  is a circle with a radius of  $1.0 - 0.3 = 0.7$  m. In our experiment, this area of the circle is used as the sensing area of the robot.

#### 6.4. Experiment and discussion

The sequence of the experiment is shown in Fig. 15. First, the robot executed an exploration task in order to find an object. Grid points and sensing points were arranged, and the path that connects all of the sensing points was then generated (Fig. 15b). In the way of the exploration path, the robot finds a new object A (Fig. 15c). The robot obtains specific information, such as the relative configuration to object A, the goal configuration of object A the shape of object A, and the grasping configuration of object A from the attached mark. The robot interrupts the exploration task and starts executing the rearrangement task based on the information it acquires. All the executable primitive motions at this state are listed as follows:

- (1) Motion to reach the grasping configuration of object A.
- (2) Motion to transfer object A to its goal position.

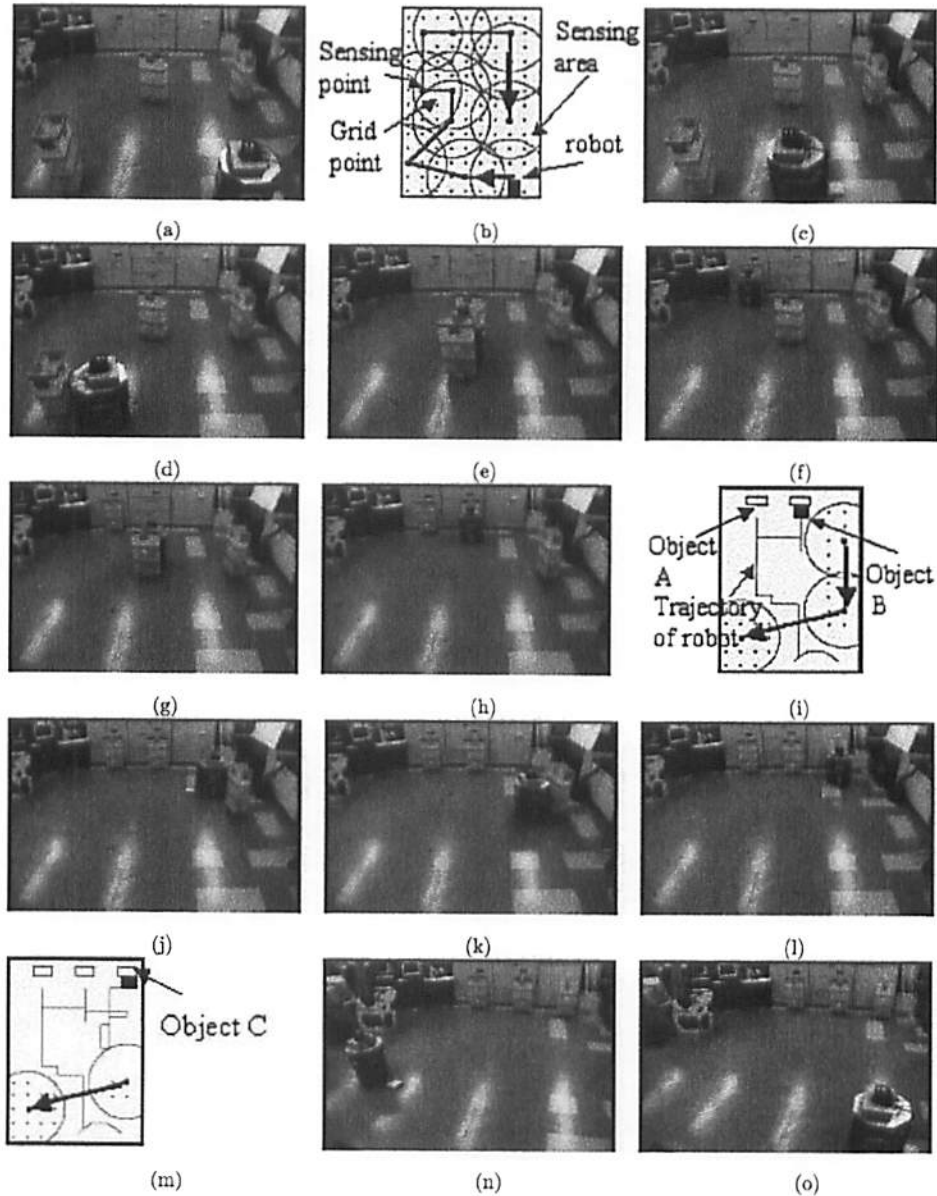
The robot selected motion 1 and tried to carry object A to the goal configuration (Fig. 15d). While carrying object A, the robot encountered object B (Fig. 15e). The motion was interrupted, and all the executable primitive motions at this state were listed as follows:

- (1) Motion to transfer object A to its goal position.
- (2) Motion to carry object A to a temporal intermediate configuration to carry object B to its destination.
- (3) Motion to reach the grasping configuration of object B.

The robot selected motion 1, which was the nearest to the goal state (here, the goal state of the rearrangement task is the state in which both A and B are carried to their own goal positions). The robot carried object A to its goal position (Fig. 15f). After executing the motion, all the executable primitive motions at this state were listed as follows:

- (1) Motion to reach the grasping configuration of object B.
- (2) Motion to carry object A to the intermediate configuration so that the robot can reach the grasping configuration of object B.
- (3) Motion to carry object A to a temporal intermediate configuration to carry object B to its destination.

The robot selected motion 1, which was the nearest to the goal state, and it then grasped object B (Fig. 15g). After executing the motion, all the executable primitive motions at this state were listed as follows:



**Figure 15.** Views of the experiment.

- (1) Motion to reach the grasping configuration of object A.
- (2) Motion to transfer object B to its goal position.
- (3) Motion to carry object B to the intermediate configuration so that the robot can reach the grasping configuration of object A.
- (4) Motion to carry object B to a temporal intermediate configuration to carry object A to its destination.



The robot selected motion 2, which was the nearest to the goal state. The robot carried it to the goal position (Fig. 15h). Since the robot did not know the existence of other objects, the rearrangement task was interrupted, and the exploration task was resumed. The sensing points were rearranged on the working area, and the exploration paths were regenerated (Fig. 15i). At the first sensing point, the robot recognized object C (Fig. 15j). The robot interrupted the exploration task and resumed the rearrangement task. All the executable primitive motions at this state are listed as follows:

- (1) Motion to reach the grasping configuration of object A.
- (2) Motion to reach the grasping configuration of object B.
- (3) Motion to reach the grasping configuration of object C.

The robot selected motion 3, which was the nearest to the goal state. The robot grasped object C (Fig. 15k). After executing the motion, all the executable primitive motions at this state are listed as follows:

- (1) Motion to reach the grasping configuration of object A.
- (2) Motion to reach the grasping configuration of object B.
- (3) Motion to transfer object C to its goal position.
- (4) Motion to carry object C to the intermediate configuration so that the robot can reach the grasping configuration of object A.
- (5) Motion to carry object C to a temporal intermediate configuration to carry object A to its destination.
- (6) Motion to carry object C to the intermediate configuration so that the robot can reach the grasping configuration of object B.
- (7) Motion to carry object C to a temporal intermediate configuration to carry object B to its destination.

The robot selected motion 3, which was the nearest to the goal state. The robot carried object C to its goal position (Fig. 15l). Since the robot did not know any other objects, it interrupted the rearrangement task and resumed the exploration task. The sensing points were rearranged within the working area again (Fig. 15m and n). Since the entire working area had been recognized and no other objects had been found, the robot went back to its initial position (Fig. 15o).

The average time for each experiment was approximately 20 min, which included 8 min for recognizing objects or the environment, 11 min for moving and 20 s for planning. Considering the planning time, the task was finished within a reasonable time. The success rate of this experiment was almost 80% in 20 trials. Most of the failures were caused by odometry errors by the robot. As stated in Section 6.3, a robot can modify its configuration by observing the mark; however, this could be achieved based on the assumption that the robot must exist within the recognizable area of the mark. Therefore, if the odometry error is much larger, the robot will pass over the mark and fall into a situation in which it cannot modify its configuration. Conversely, the task was always completed as long as the robot could

identify the mark. From the above, the control of a mobile robot that was capable of exploring an area and rearranging multiple objects of unknown shapes and positions was achieved.

## 7. CONCLUSIONS

In this paper, a method is proposed for the control of a mobile robot that executes the exploration and rearrangement of multiple unknown objects. We proposed a method that integrates the exploration and rearrangement algorithm. Its effectiveness is shown by simulation with respect to the computational cost and path length. In addition, experiments were conducted with an actual robot. In the future, we will improve the success rate by combining the self-localization algorithm with the proposed method. In addition, we will extend this system to multi-robot systems.

### *Acknowledgements*

This research is partly supported by the Institute of Physical and Chemical Research.

## REFERENCES

1. M. Held, *On the Computational Geometry of Pocket Machinig, Lecture Notes in Computer Science 500*. Springer-Verlag, Berlin (1991).
2. Z. L. Cao, Y. Huang and E. L. Hall, Region filling operations with random obstacle avoidance for mobile robots, *J. Robotic Syst.* **5**, 87–102 (1989).
3. V. Lumelsky, S. Mukhopadhyay and K. Sun, Sensor-based terrain acquisition: a 'seed spreader' strategy, in: *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and System*, Tsukuba, pp. 62–67 (1989).
4. E. U. Acar and H. Choset, Sensor-based coverage of unknown environments: incremental construction of Morse decompositions, *Int. J. Robotics Res.* **21**, 345–366 (2002).
5. D. Kurabayashi, J. Ota, T. Arai and E. Yoshida, An algorithm of dividing a work area to multiple mobile robots, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Pittsburg, PA, Vol. 2, pp. 286–291 (1995).
6. O. Ben-Shahar and E. Rivlin, Practical pushing planning for rearrangement tasks, *IEEE Trans. Robotics Automat.* **14**, 549–565 (1998).
7. J. Andreas, Pushing the limits: new developments in single agent search, PhD Thesis, University of Alberta (1999).
8. R. E. Korf, Depth-first iterative-deeping: an optimal admissible tree search, *Artificial Intelligence* **27**, 97–109 (1985).
9. A. Ishiguro, Y. Shirai, Y. Watanabe and Y. Uchikawa, Emergent construction of immune networks for autonomous mobile robots through the metadynamics function, in: *Proc. 4th Eur. Conf. on Artificial Life*, Brighton, pp. 318–326 (1997).
10. R. Pfeifer and C. Scheier, *Understanding Intelligence*. MIT Press, Cambridge, MA (1999).
11. J. Ota, Rearrangement planning of multiple mobile objects by using real-time search methodology, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington, DC, pp. 947–953 (2002).
12. R. E. Korf, Real-time heuristic search, *Artificial Intelligence* **42**, 189–211 (1990).

13. C. Trevai, K. Ichikawa, Y. Fukazawa, H. Yuasa, J. Ota, T. Arai and H. Asama, Real-time cooperative exploration by reaction-diffusion equation on a graph, in: *Proc. 6th Int. Symp. on Distributed Autonomous Robotic Systems*, Fukuoka, pp. 383–392 (2002).
14. H. Yuasa and M. Ito, Internal observation systems and a theory of reaction-diffusion equation on a graph, in: *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 3669–3673 (1998).
15. E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan and D. B. Shmoys, *The Traveling Salesman Problem — A Guided Tour of Combinatorial Optimization*. Wiley-Interscience, New York (1985).
16. J. K. Lenstra and A. H. G. Rinnooy Kan, On general routing problems, *Networks* **6**, 273–280 (1976).
17. M. Kwan, Graphic programming using ODD or EVEN points, *Chinese Math.* **1**, 273–277 (1962).
18. M. Ringer, G. Reinelt and G. Rinaldi, The traveling salesman problem, in: *Handbooks in Operations Research and Management Science, Vol. 7, Networks*, M. O. Ball, T. L. Magnaniki, G. L. Monma and G. L. Nernhauser (Eds). Elsevier Science, Amsterdam.
19. R. Katsuki, J. Ota, T. Mizuta, T. Kito, T. Arai, T. Ueyama and T. Nishiyama, Design of an artificial mark to determine 3D pose by monocular vision. in: *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 995–1000 (2003).

#### ABOUT THE AUTHORS



**Yusuke Fukazawa** received the BS and MS degrees from the Faculty of Engineering, University of Tokyo in 2002 and in 2004, respectively. From 2004, he joined NTT Docomo Cooperation. His research interests are learning mechanism of mobile robot system and multiple mobile robot system.



**Trevai Chomchana** received the BS degree from the Faculty of Mechanical Control Engineering, University of Electro-Communication in 2001, and the MS degree from the Faculty of Engineering, University of Tokyo in 2003, respectively. He is currently a PhD candidate of the Faculty of Engineering, the University of Tokyo.



**Jun Ota** received the MS and DS degrees from the Faculty of Engineering, University of Tokyo in 1989 and 1994, respectively. From 1989 to 1991, he joined Nippon Steel Corp. In 1991, he was a Research Associate of the University of Tokyo. In 1996, he became an Associate Professor at the Graduate School of Engineering, University of Tokyo. From 1996 to 1997, he was a Visiting Scholar at Stanford University. His research interests are multiple mobile robot system, environmental design for robot systems, human–robot interface and cooperative control of multiple robots.



**Hideo Yuasa** received the BS and MS degrees in Engineering from Nagoya University in 1984 and 1986, respectively, and obtained the PhD from the same university in 1992. From 1992, he was an Assistant Professor at the Department of Electronic-Mechanical Engineering, Nagoya University. From 1993, he held an additional post of Visiting Researcher of the Bio-Mimetic Control Research Center, Institute of Physical and Chemical Research (RIKEN). Since 1999, he was an Associate Professor at the Department of Precision Engineering, The University of Tokyo. He passed away on September 2002. His specialty was distributed autonomous systems, especially mathematical models and its applications.



**Tamio Arai** received the MS and DS degrees in Engineering from the Department of Precision Machinery Engineering, University of Tokyo in 1972 and 1977, respectively. He was a Visiting Researcher in the Department of Artificial Intelligence, Edinburgh University from 1979 to 1981. He has been a Professor in the Department of Precision Machinery Engineering, University of Tokyo since 1987. His specialties are assembly and robotics, especially multiple mobile robots including the legged robot league of RoboCup. He works in IMS programs in holonic manufacturing systems. He has contributed to robot software in ISO activities. He is an active member of CIRP, a member of IEEE, RSJ, JSPE and Honorary President of the Japan Association for Automation Advancement. Since 2000, he has been a Director of Research into Artifacts, Center for Engineering, University of Tokyo, and proposes a new field of engineering — Service Engineering.



**Hajime Asama** received the MS and DS degrees in Engineering from the Department of Precision Machinery Engineering, University of Tokyo in 1984 and 1989, respectively. He worked at RIKEN (Institute of Physical and Chemical Research, Japan) as a Research Associate, Research Scientist, Senior Research Scientist and Senior Scientist from 1986 to 2002, and became the Professor of RACE (Research into Artifacts, Center for Engineering), University of Tokyo in 2002. He received the JSME Robotics and Mechatronics Division Robotics and Mechatronics Award in 1995, the JSME Robotics and Mechatronics Division Robotics and Mechatronics Academic Achievement Award in 2000, the JIDPO Good Design Award in New Frontier Design Category in 2002, etc. He was an Editor of *Distributed Autonomous Robotics Systems*, for its second and fifth volumes which were published by Springer-Verlag, Tokyo. He is a member of the IEEE, RSJ, JSME, SICE, New York Academy of Science, etc. His main interests are distributed autonomous robotic systems, cooperation of multiple autonomous mobile robots, emergent robotic systems, intelligent data carrier systems and service engineering.



**Kuniaki Kawabata** received the BE, ME and PhD degrees in Electrical Engineering from Hosei University in 1992, 1994 and 1997, respectively. He joined the Biochemical Systems Laboratory at RIKEN (Institute of Physical and Chemical Research) as a Special Postdoctoral Researcher from 1997 to 2000. In 2000, he joined the Advanced Engineering Center at RIKEN as a Research Scientist. In 2002, he joined the Distributed Adaptive Robotics Research Unit, RIKEN. His research interests cover distributed autonomous robotic systems (DARS), emergent systems and adaptive mechanisms. He is a member of the RSJ, SICE, IEEJ, JSAI and IEEE.