International Journal of Robotics and Automation, Vol. 16, No. 4, 2001

38

DISTRIBUTED GUIDANCE KNOWLEDGE MANAGEMENT BY INTELLIGENT DATA CARRIERS

D. Kurabayashi,* K. Konishi,** and H. Asama***

Abstract

The authors built a device and algorithm to implement autonomous robots that can enhance efficiency through autonomous knowledge acquisition and sharing. They also propose a quantitative -valuation algorithm of task execution by autonomous mobile rot In the robotic system, the intelligent data carrier (IDC) provides knowledge of guidance for autonomous mobile robots. An IDC summarizes fragments of knowledge from individual robots and reports the best direction to a destination that a robot wants to go. Based on a stochastic model, the authors establish a function to evaluate effectiveness of knowledge management by the IDCs. They also describe an algorithm to arrange layout of IDCs by relaxed problem solution.

Key Words

Autonomous knowledge acquisition and sharing, intelligent data carrier, stochastic model

1. Introduction

Currently, researchers are trying to create a robotic system that can function in any general environment. Realization of autonomous task execution would be especially advantageous in hazardous environments. Most robotic systems require a model environment in order to execute tasks effectiv Autonomous robotic systems should create models of the environment by themselves. However, such tasks are not easy for current autonomous robots because they have only limited ability to sense and thus survey the environment. In such cases, the method of knowledge acquisition and sharing becomes very important.

Some researchers have proposed local communication methods for enhancing communication between robots [1–4]. In previous related studies, only robots kept the knowledge. Let us consider what takes place on the ground: ants forage effectively by pheromone trails, dogs claim territory by smell. Social creatures improve the efficiency of their actions by storing information in the environment. Simulation studies by Drogoul and Feber [5] have suggested that pheromone trails are very effective for completion of iterative transportation tasks to be performed by autonomous agents. This kind of data storage and communication system can be applied by creating model environments for robotic systems for the sharing of knowledge and for cooperation.

We have proposed a device that enables local communication; we refer to the device as an "Intelligent data carrier" (IDC) (Fig. 1). We have developed not only such a device but also algorithms to enhance the efficiency of task execution performed by autonomous robots via knowledge sharing and acquisition through the intelligent data carrier system in particular environments.

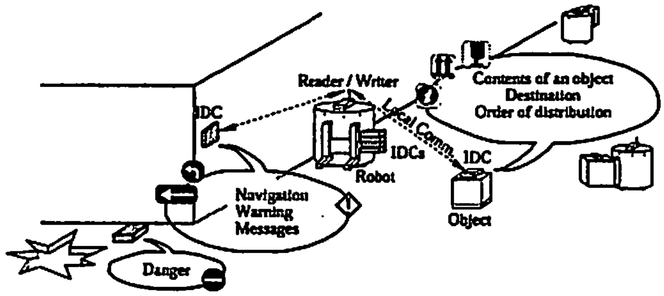


Figure 1. An overview of local communication by the IDC.

This work focuses on multiple robot guidance system using IDCs. Each robot can work without a global map of the work area by sharing the information of the map through IDCs that are arranged in the work area. We denote an algorithm via which robots autonomously acquire and share knowledge for guidance. Simulation results show that the IDC can navigate robots effectively. We introduce the hardware and algorithm as Part 1.

There is, however, no algorithm for arranging IDCs such that each robot works the most effectively. In order to provide the algorithm for placing IDCs, this article proposes a mathematical model of behaviour of multiple robots with IDCs. The aim of this study is to provide an algorithm such that multiple robots arrange IDCs au-

* Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, Japan; e-mail: dkura@ieee.org

** Science University of Tokyo, 1-3 Kagurazaka, Shinjuku-ku, Tokyo, 162-8601, Japan; e-mail: konishi@ms.kagu.sut.ac.jp

*** The Institute of Physical and Chemical Research (RIKEN), 2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan; e-mail: asama@go.jp

(paper no. 206-2510)

tonomous under the unknown environment. It is, however, difficult to provide such an algorithm. Therefore, as a first step in providing such an algorithm, this article deals with the IDCs arrangement problem under the known environment in part 2. The IDCs arrangement problem considered here is to find the optimal arrangement such that each robot can reach the directed destination in the work area. The work area is formulated as a graph and a matrix, and the problem of finding the optimal arrangement of IDCs is formulated as the problem of minimizing the maximum eigenvalue of the matrix. A numerical example shows that the optimal arrangement of IDCs can be obtained quickly.

This article is organized as follows. Section 2 explains IDCs and the robot guidance system. We propose an algorithm for autonomous robots to acquire and share knowledge of guidance. Section 3 provides the mathematical model for the behaviour of robots with IDCs in the work area. We create a mathematical model to analyze the ehaviours of robots and to evaluate the efficiency of the proposed algorithm. In Section 4, the IDCs arrangement problem is formulated. We denote an algorithm and a numerical example of the problem.

Part 1: Distributed Guidance System for Autonomous Robots

2. Autonomous Knowledge Acquisition and Sharing

2.1 Intelligent Data Carrier System

We have developed an intelligent data carrier (IDC) [6–7] to reduce the traffic of global communication by providing local communication links and local information management functions. By reading information and writing it into the IDCs, robots can use IDCs as media for inter-robotic communication.

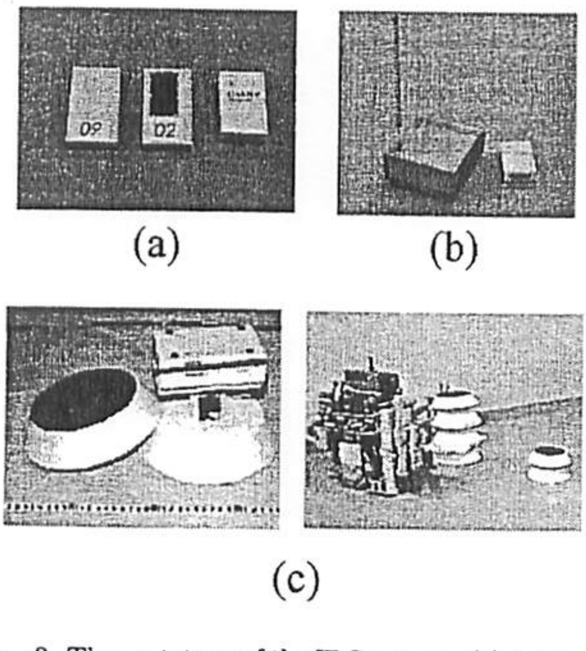


Figure 2. The prototype of the IDC system. (a) A tag. (b) A reader/writer. (c) Cover and portable system.

The IDC system consists of portable information storage (tags) and read-write devices carried by the robots (Fig. 2). Tags are usually referred to as an IDC. A tag has its own CPU, memory, and battery. A user can download and execute original program into the tags. The specifications of the IDC are shown in Table 1. By placing the IDCs in specific locations in a particular environment, robots can allocate functions to act as agents for information storage and management (Fig. 2(c)). The communication range is up to 3.0 [m].

Table 1
Specifications of the IDC

Media	Electromagnetic Wave
Frequency	290, 310 [MHz]
Memory	32 bytes
Modulation	ON/OFF keying
Data Rate	1200 [bps]
Power Source	a Li-ION battery (3.6V)
Size	tag: 110 x 65 x 25 [mm] reader/writer: 195x130x50 [mm]

2.2 Problem Settlement

In this work, we consider iterative transportation tasks (e.g., [8]). A robot has to carry objects to given destinations. We posit an environment in which several destinations are located. When a robot achieves the desired destination, it receives instructions regarding the next destination and then continues with the task at hand.

We assume that robots do not have maps, because a fixed map, made by humans, may decrease the flexibility of autonomous robotic systems. We consider adaptability to unknown environments crucial. We assume that a robot consists of the following characteristics.

- A robot does not have a map and does not estimate its global position.
- A robot can sense walls and distinguish paths, junctions, and destinations. It can also distinguish branches at junctions.
- A robot can remember the last visited destination and count, in steps, the duration of running time.

A robot cannot understand its global position, but it can understand the local situation around it.

In this example, we assume that the transportation task is as follows.

- A robot is given only the ID number of a destination. When a robot arrives at the destination, it receives another destination ID at random.
- The work area is a maze-like environment that consists of square cells and walls (Fig. 3).
- A robot can move to neighbouring cells, but only one step at a time.
- We do not consider cases of collisions among robots.

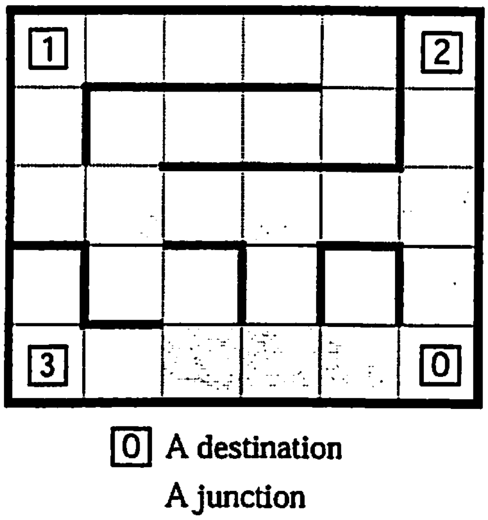


Figure 3. An example of a work area.

2.3 Algorithm to Acquire and Share Knowledge

In this section, we propose an algorithm to build knowledge for guidance autonomously. A robot should select feasible branches at junctions, branches that lead to the shortest path to its current destination, for effective transportation is the goal. We set IDCs at junctions to facilitate autonomous robots' decisions. Robots can store and share their fragments experience by the IDCs. A robot does not need to estimate its global position if a robot can obtain sufficient knowledge from an IDC.

As we assumed, a robot does not know its global position. Thus we propose an algorithm to locate where branches connect at junctions to expected destinations. This proceeds according to the last visited destination and entered branch. For example, when a robot, which has started destination 1, enters a junction through a southern branch, we can expect the southern branch may lead us to destination 1.

We describe the data structure of an IDC in Fig. 4. At the initial state, no valid data are recorded. When a robot enters a communication area of an IDC at a junction, it reports the branch of entry, the ID number of the most recently visited destination, and running step measured from the destination to the IDC.

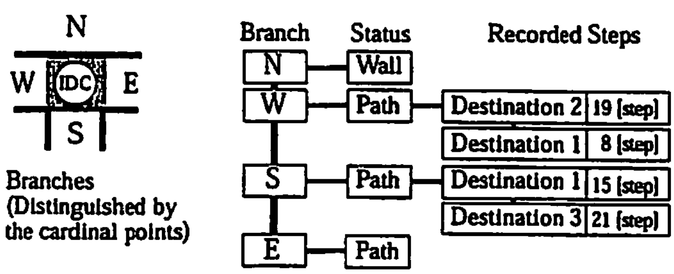


Figure 4. Data structure in an IDC.

When the IDC has already received data about the same destination in the same branch, it compares the current running step with a former one. If the new one is shorter than the former, the record is renewed.

When a robot wants to go to destination 1, it should choose a branch leading to the fewest numbers of steps from the destination. In the example of Fig. 4, the robot should choose the W (western) branch. We implemented an algorithm of probabilistic branch selection. The procedure of the algorithm is as follows.

1. When a robot cannot communicate with an IDC at a junction, it chooses a branch at random.
2. A robot whose destination is j approaches a junction that has m branches. We describe recorded steps from destination j in branch $i \in m$ as t_{ij}. We can find four items of data, $t_{W1} = 8$, $t_{W2} = 19$, $t_{S1} = 15$, $t_{S3} = 21$, in the example shown in Fig. 4.
3. When the IDC has no record of destination j, the robot chooses a branch at random.
4. When the IDC has one or more branches that contain records about destination j, it calculates $s_{ij} = \frac{1}{t_{ij}}$. If branch i has no record about destination j, set $s_{ij} = 0$.
5. Calculate probability p_{ij} to choose branch i as in equation (1). $P_{\min}$ denotes a fixed probability that a robot chooses a branch randomly.

$$p_{ij} = (1 - p_{\min}) \frac{s_{ij}}{\sum_j^m s_{ij}} + \frac{p_{\min}}{m} \qquad (1)$$

2.4 Simulation Results

We verify the effectiveness of the proposed algorithm by simulations. We evaluate the number of achieved destinations in constant steps. We assume the environment shown in Fig. 3 and make simulations with or without IDCs at the junctions. A robot works for 1,000 steps. We set $P_{\min} = 0.10$. We set IDCs in all junctions in the case of applying IDC and the proposed algorithm.

Fig. 5(a) shows results by single robot. We can see that the proposed algorithm with IDCs achieved about 600% more destinations than without IDCs, although we did not give robots any knowledge about the environment in advance.

Fig. 5(b) and 5(c) compare of simulations by a single robot and four robots. The environment of Fig. 5(b) has no IDC, but that of Fig. 5(c) has IDCs at all junctions. The thin line in both figures denotes the number of achieved destinations by single robot, and the thick line illustrates the average number of four robots.

When robots cannot use IDCs, the thin and thick lines are almost the same. When robots can obtain and share knowledge of guidance by IDCs (Fig. 5(c)), the average of the four robots is about 10% higher than the result of the single robot. Four robots can acquire knowledge for guidance much faster than a single robot does. Those results suggest that the proposed algorithm and IDC system realized implicit cooperation among autonomous robots without explicit communication nor a priori knowledge.

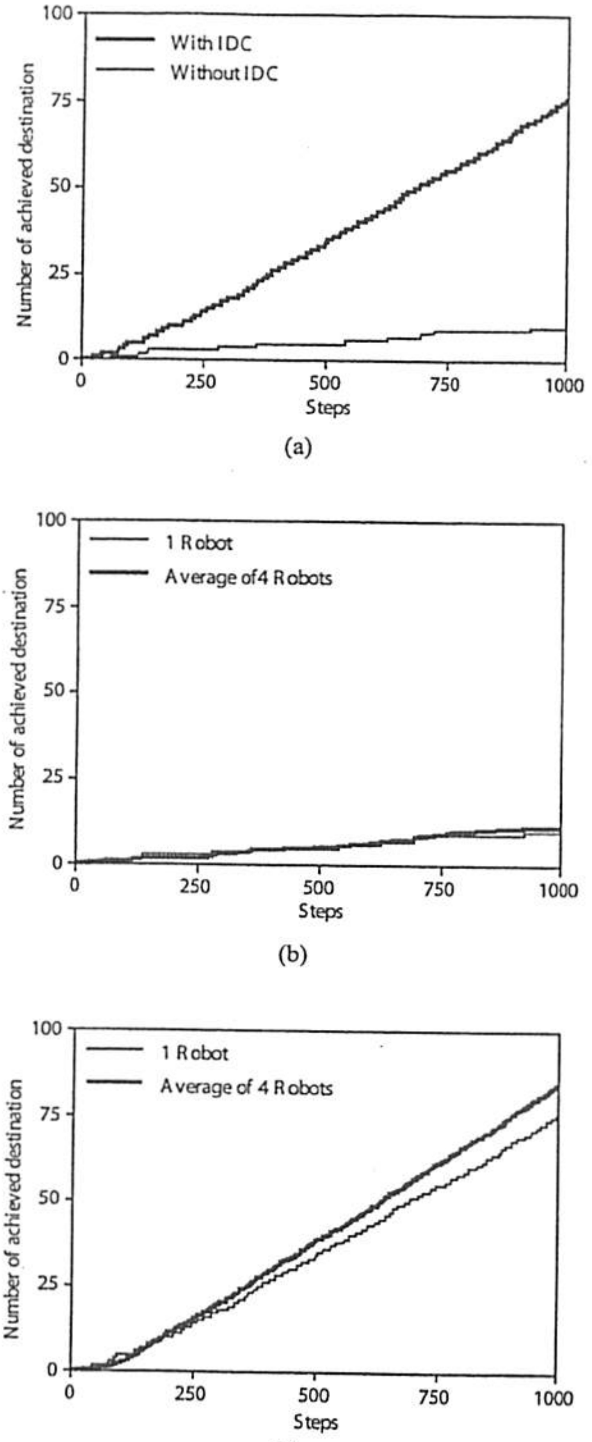


Figure 5. Comparison of task execution. (a) Single robot. (b) Four robots, without IDC. (c) Four robots, with IDC.

2.5 Implementation and Demonstration

We applied the IDC system and the proposed algorithm to an actual robotic system. We attached the IDC system to an omni-directional autonomous mobile robot [9], shown in Fig. 6. The system was developed by the authors. This robot has its own computer with an MMX Pentium (200MHz), 16MB memory, and an ISA bus (Table 2). It is equipped with infrared sensors [10], which are located in the silver circle at the top; a gyro is placed on the robot so that it knows its orientation. It runs at 0.04[m/s] with four Pb batteries.

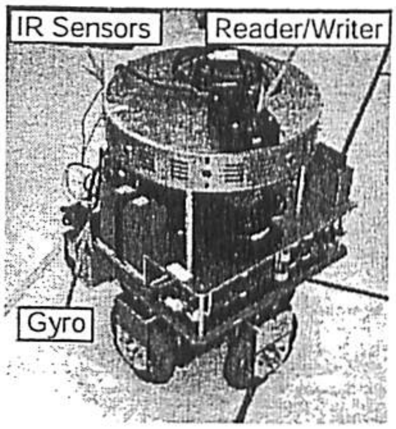


Figure 6. An autonomous omni-directional robot.

Table 2
Specifications of Mobile Robots

Robot Size	0.42 x 0.42 x 0.70 [m]
CPU	MMX Pentium 200 [MHz]
Sensor	IR sensors, a gyro
Running Speed	0.04 [m/s]
Batteries	12V 5Ah x 4

We configured an experimental environment that was surrounded by walls (Fig. 7). There were two junctions in the environment. The robot kept its distance from the walls by reactive motions dictated by the infrared sensors. The infrared sensors were also used to locate deadends and junctions. We placed two IDCs at each junction and four IDCs at each deadend (Fig. 7(b))

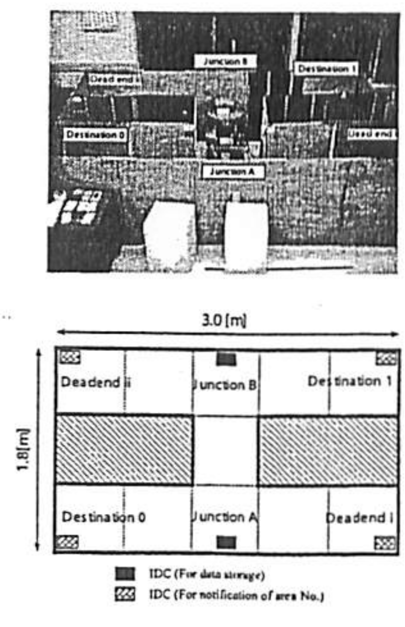


Figure 7. The experimental environment.

The robot ran for 40 steps (about 25 minutes) per experiment. We conducted experiments with two IDCs, one IDC, and without IDC. Fig. 8 indicates trajectories of a robot in a train diagram. The "⊙" in the figure represents the achievement of a destination. We evaluated the achieved number after 40 steps (Fig. 9). The robot driven by the proposed algorithm improved motion by utilizing knowledge from IDCs in the junctions. The experimental robot achieved six destinations. The robot without access to an IDC achieved only three destinations, because it had to choose a branch randomly at each junction. The effectiveness of the proposed algorithm can be increased if robots are allowed to run for more steps, because robots can make mistakes at the beginning, when each IDC has not yet accumulated knowledge.

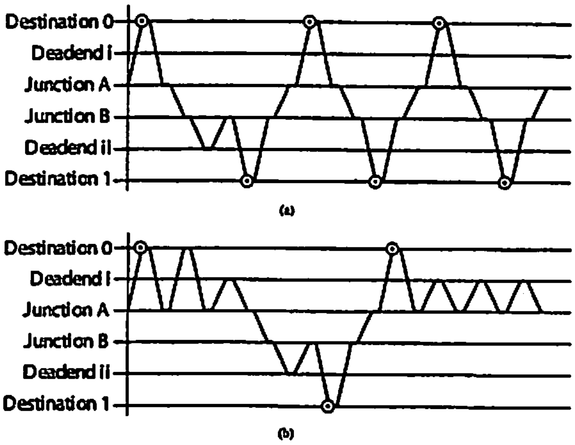


Figure 8. Comparison of trajectories of the robots. (a) With IDC. (b) Without IDC.

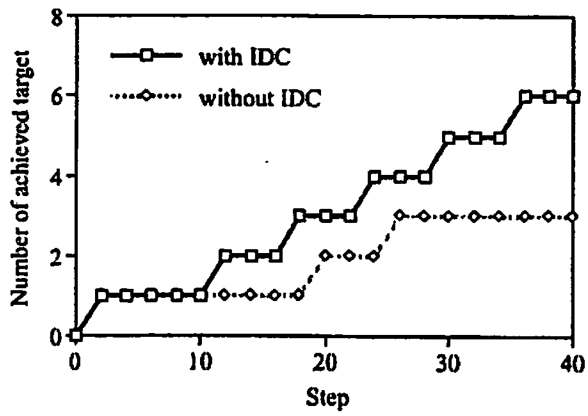


Figure 9. Number of achieved targets.

The experimental results suggest that the proposed algorithm and system can perform in an actual robotic system.

2.6 Discussion

Although we assume only fixed (static) environment, applications in dynamic environments are demanded. Adaptation of the proposed algorithm to dynamic environments is still open to problems, because each independent IDC has to decide the validity of its information. The authors have proposed a heuristic algorithm [11] by which an IDC decides whether it should renew its current data or not. In the algorithm, we apply a logistic function, which derives probability to delete data according to steps reported by robots. In the proposed algorithm [11], an IDC decides whether it deletes current data or not according to the following procedures and equations.

1. A robot r starts from previous destination i and proceeds to a new destination j. After t_r steps from destination i, it meets IDC k and reports the steps. When the IDC k has knowledge about both i and j, we denote them d_i and d_j, respectively. When the IDC k does not have both, skip step 2.
2. According to current data of IDC k, we expect that the robot r can reach IDC k by d_i steps and will arrive at j after d_j steps when there are no changes in the environment. So we compare t_r with d_i and d_j to calculate probability p_{del} to erase the IDC's current knowledge. We apply logistic function (2) to calculate:

$$p_{del} = K\frac{e^{m(t_r-(d_i+d_j+d_j^2))}}{1+e^{m(t_r-(d_i+d_j+d_j^2))}} \tag{2}$$

Equation 3 determines m, which results in $p_{del} = 0.01$ when $t_r = d_i + d_j$:

$$m = \frac{log99}{d_j^2} \tag{3}$$

3. When the IDC does not erase its knowledge, it obtains data from and suggests a direction to the robot as in a former algorithm.

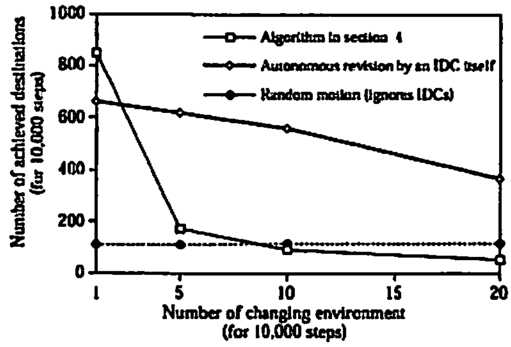


Figure 10. A heuristic algorithm [11] for dynamic environments.

We use simulations to verify the effectiveness of this algorithm. In the simulations, we set $K = 0.5$ and $p_{min} = 0.01$, and applied 10 robots. As a dynamic environment, we exchange locations of destinations in Fig. 3: $1 \to 2$, $2 \to 3$, $3 \to 0$, $0 \to 1$. We execute each simulation for 10,000 steps. We set four conditions, where we exchange destinations 1, 5, 10, and 20 times (during 10,000 steps), respectively. Figure 10 compares of achieved destinations. The algorithm for fixed environment (proposed in Section 4) performs very well when locations of destinations are hardly exchanged. However, in dynamic environments its performance becomes quite low. In the condition where we exchange destinations 20 times, it achieves fewer destinations than robots did by random motion. In contrast to the result, the heuristic algorithm introduced in this section can maintain its performance where locations of destinations are frequently exchanged.

The proposed algorithm also has applicability to an environment in which IDCs are not located in all junctions. In other words, the proposed system can function flexibly if some IDCs are broken.

In the next section, we analyze performance when there are some junctions without IDC tags in a map. Based on the analysis, we calculate optimal layout of IDC tags in Section 4.

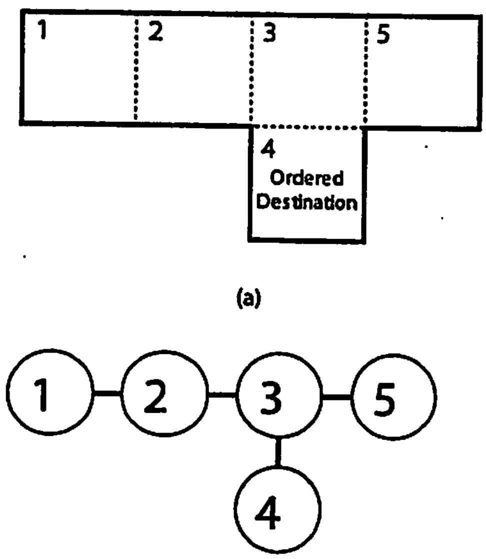
Part 2: Evaluation and Arrangement of the Proposed System

. Performance Evaluation by a Stochastic Model

3.1 Formulation

In this section, we formulate the effects of knowledge sharing and establish a method of evaluating autonomous robotic systems with the IDC system. We estimate ideal performance of knowledge sharing with IDCs as a performance index of the proposed algorithm. We calculated the expected number of steps needed to reach the given destinations. In order to analyze the behaviour of robots with IDCs, a stochastic model was introduced. The positions of robots were described by a state equation.

First, let us consider the cases lacking IDCs. We can assume that the work area of robots is divided by grids, as shown in Fig. 11(a). The work area itself is described by an undirected graph, as shown in Fig. 11(b). The stochastic model is given based on the graph, as follows:



(a)

(b)

Figure 11. Example of evaluation model. (a) Layout of a work area. (b) Undirected graph.

$$x(t+1) = Ax(t) \tag{4}$$

$$x(t) = [x_1(t)x_2(t)\cdots x_n(t)]^T \tag{5}$$

In the above equations, x_i denotes the probability that a robot is in the ith node of the graph at step t. Therefore, x_i must satisfy $0 \leq x \leq 1$. $A = (a_{ij})$ denotes the state transition matrix and a_{ij} describes the probability that the robot in the jth node moves into the ith node. The robot that arrives at the given destination, stops, and stays there. We assume here that these robots disappear from the work area. Hence, if the given destination is lth node:

$$a_{il} = 0 \; i = 0, 1, \cdots, n \tag{6}$$

$$A = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 1 \\ 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{bmatrix} \tag{7}$$

In any case, the maximum eigenvalue of A is smaller than 1, and it is clear that $x \to 0$ as $t \to \infty$.

Next, let us consider the case that employs IDCs. In the node with an IDC, a robot can obtain information about a branch to a desired position; therefore, such robots move exactly into the node that is close to the ordered destination. In other words, such robots will move into the next node with a probability 1.

$$A = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

3.2 Evaluation Function

Here, A_k is a transition matrix that indicates transition of a robot seeking destination k. We calculate expected steps E_k that it will take to reach destination k.

$$\begin{aligned} E_k &= \textstyle\sum_{t=0}^{\infty} c^T x(t) \\ &= c^T \textstyle\sum_{t=0}^{\infty} A_k^t x(0) \\ c &= [11\cdots 1] \end{aligned} \tag{9}$$

where $x(0)$ indicates initial conditions, and $\sum_i^n x_i(0) = 1$.

As a single work area may have several destinations, we evaluated the average of expected steps when a work area has m destinations:

$$E_{index} = \frac{\sum_{i=1}^{m} E_i}{m} \tag{10}$$

We created a model for evaluation by employing a simple Malkov model. A robot could proceed back and forth on the model even when in the middle of a corridor. Actual robots do not perform useless movements. We introduce a directed graph to avoid the movements (Fig. 12).

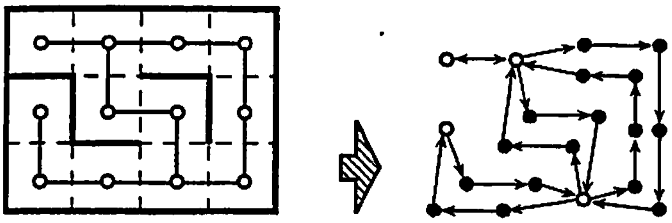


Figure 12. Arrangement of a work area.

3.3 Performance Evaluation

The performance of the proposed system depends on the IDC layout. We considered 2^6 combinations for six junctions in the map (Fig. 3). We evaluate performance of transportation in each case by the use of simulations. We compare the averaged number of steps to reach given destinations. Figure 13 offers a comparison of the ideal number of steps calculated by the proposed method with the actual number of steps obtained by simulation during 10,000 steps. The vertical axis shows averaged steps needed to reach given destinations. The horizontal axis indicates combinations of IDC layout possibilities. We sorted the combinations according to ideal steps.

A robot driven by the proposed algorithm, described in Section 4, had to acquire knowledge for guidance autonomously because IDCs have no data at the beginning of the task. The comparison shows how much the overhead of autonomous knowledge acquisition is.

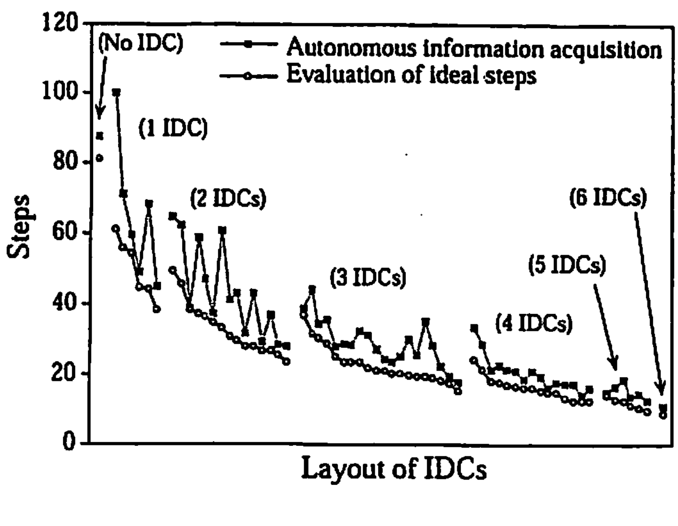


Figure 13. Evaluation of the algorithm to acquire guidance information by autonomous agents.

The robot in simulations payed +28% steps on average to reach a desired destination. The standard deviation, σ, was 0.173. We evaluated that the proposed robotic system with IDCs pays +28% steps for autonomous knowledge acquisition and sharing.

Figure 13 also illustrates the robustness of the proposed system. The proposed system has two aspects of robustness. One is the robustness against failures of robots because the environment itself keeps knowledge. Even if a new robot is applied when some robots are broken or move to other places, it can perform the same as the former robots can. The other aspect concerns the failures of IDCs. IDCs are distributed and work independently; we can see the performance when we remove some IDCs at junctions in the environment shown in Fig. 3. If one of six IDCs is broken, the system still achieves more than 75% of destinations. Even if half of six IDCs are removed, the system retains about 41% achievement of destinations, which is about three times as much as the case without IDCs. Additionally, we only have to replace a broken IDC with a new one to restore the performance of the system because robots autonomously construct guidance knowledge by the proposed algorithm.

4. IDCs Arrangement Problem

This section deals with the IDCs arrangement problem. It is best that we allocate IDCs in all nodes; however, we have to choose only some nodes to put IDCs because there are not enough IDCs. Therefore, IDCs should be arranged so that each robot can reach the directed destination as soon as possible.

The original arrangement problem takes combinatorial order calculation time, which is hard to solve. In this section, we introduce relaxed form so that we obtain results in a polynominal time.

4.1 Problem Formulation

It is important to evaluate the fitness of arrangement of IDCs. In this work, the fitness of arrangement of IDCs is considered as quickness of convergence of $x(t)$ in (4). The quickness of convergence x depends on the maximum eigenvalue of A. Thus, the IDCs arrangement problem considered here is finding the arrangement of IDCs that minimizes the maximum eigenvalue of A.

We assume that there are k IDCs and m nodes on which IDCs can be put. Then, there are ${}_mC_k$ ways to arrange the IDCs. As mentioned in Section 3, the state transition matrix A depends on the arrangement of IDCs, and therefore it is necessary to find the best state transition matrix A out of ${}_mC_k$ candidates, which we describe as a set $\mathcal{A}$. Hence, the IDCs arrangement problem is formulated as follows:

$$\text{Minimize } \bar{\lambda}(A) \quad \text{subject to} \quad A \in \mathcal{A} \tag{11}$$

where $\bar{\lambda}(\cdot)$ denotes the maximum eigenvalue.

4.2 Relaxed Problem

The optimization problem described by (11) is combinatorial problem that is difficult to solve. In order to overcome combinatorial difficulties, we introduce the relaxed problem [12]. The following problem is the relaxed problem of (11):

$$\text{Minimize } \bar{\lambda}(\hat{A}) \quad \text{subject to} \quad \hat{A} \in \hat{\mathcal{A}}, \tag{12}$$

$$\hat{\mathcal{A}} := \left\{ \hat{A} \in \Re^{n \times n} : \begin{array}{c} \hat{A} = (\hat{a}_{ij}), \\ \hat{a}_{ij} = a_{ij} \ {}^{\forall} j \notin \mathcal{N}, \\ \sum_{i=1}^{n} a_{ij} = 1 \ {}^{\forall} j \in \mathcal{N}, \\ 0 \leq \hat{a}_{ij} \leq 1 \end{array} \right\} \tag{13}$$

where $\mathcal{N}$ is an index set of candidates for the node where IDC is put. The relaxed problem described by the above equations can be solved quickly by the method based on the interior point method [13]

It is clear that the following equation is satisfied because $\mathcal{A} \in \hat{\mathcal{A}}$:

$$\inf_{\hat{A} \in \hat{\mathcal{A}}} \bar{\lambda}(\hat{A}) \leq \inf_{A \in \mathcal{A}} \bar{\lambda}(A) \tag{14}$$

Therefore, the lower bound of $\bar{\lambda}(A)$ and approximate optimal solution can be obtained quickly.

After solving the relaxed problem, we have to estimate the optimal solution of (11) from the solution of (12). Let $\hat{A}^*$ denote the optimal solution of the relaxed problem (12). The approximate optimal solution can be obtained by finding $A \in \mathcal{A}$ that is close to $\hat{A}^*$. There is no guarantee that approximate optimal solution obtained in this way is 'ie best.

Step 1. Derive relaxed problem (12) from original problem (11).

Step 2. Obtain $\hat{A}$ by solving relaxed problem (12).

Step 3. Find $A \in \mathcal{A}$ that is close to $\hat{A}$.

The state transition matrix A is usually given with an unknown error. For example, when a robot finds an unknown obstacle, A must contains a model error. Thus, A naturally contains error ΔA as (15):

$$A = A_0 + \Delta A \tag{15}$$

where A_0 denotes the ideal state transition matrix, which is unknown.

Recall equation (11). The state transition matrix A is usually given with an unknown error. Let A_0^* denote the optimal solution derived by ideal state transition matrix A_0; A^* indicates a solution derived by A that contains model error, and $\hat{A}^*$ means the relaxed solution of (12). If the following inequalities are established:

$$|\bar{\lambda}(A^*) - \bar{\lambda}(A_0^*)| \leq \epsilon \tag{16}$$

$$|\bar{\lambda}(\hat{A}_0^*)| - \bar{\lambda}(A_0^*)| \leq \hat{\epsilon} \tag{17}$$

$$\hat{\epsilon} \leq \epsilon \tag{18}$$

Then we can estimate the optimal solution of (11) from the solution (12) with the tolerance ϵ.

Let us consider the IDCs arrangement problem as shown in Fig. 14. There are two IDCs that may put on (a), (b), (c), or (d). Fig. 15 shows the graph of Fig. 14. Then the state transition matrix is:

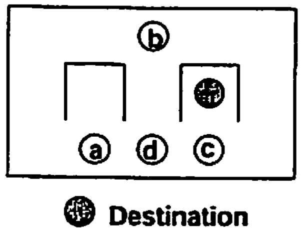


Figure 14. Numerical example: Two IDCs are put on two junctions among (a)–(d).

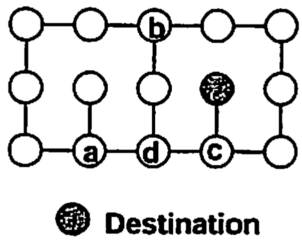


Figure 15. Graph expression of the working field.

$$A = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & b_1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & b_2 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left.\begin{array}{cccccccc}
0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0\\
\frac{1}{2}&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0\\
0&0&\frac{1}{2}&0&0&0&0&0\\
0&0&0&\frac{1}{2}&0&0&0&0\\
0&0&0&0&a_1&0&0&0\\
0&0&0&0&0&d_1&0&0\\
0&0&0&0&0&0&c_1&0\\
0&0&0&0&0&0&0&\frac{1}{2}\\
0&0&0&0&a_2&0&0&0\\
0&0&0&\frac{1}{2}&0&d_2&0&0\\
\frac{1}{2}&0&0&0&a_3&0&c_2&0\\
0&0&0&0&0&d_3&0&\frac{1}{2}\\
0&0&\frac{1}{2}&0&0&0&c_3&0
\end{array}\right] \tag{19}$$

where:

$$a_1 + a_2 + a_3 = 1 \tag{20}$$

$$b_1 + b_2 + b_3 = 1 \tag{21}$$

$$c_1 + c_2 + c_3 = 1 \tag{22}$$

$$d_1 + d_2 + d_3 = 1 \tag{23}$$

$$0 \le a_i \le 1,\ i = 1,2,3 \tag{24}$$

$$0 \le b_i \le 1,\ i = 1,2,3 \tag{25}$$

$$0 \le c_i \le 1,\ i = 1,2,3 \tag{26}$$

$$0 \le d_i \le 1,\ i = 1,2,3 \qquad (26)$$

The relaxed problem (12) can be solved using MATLAB, and the relaxed state transition matrix $\hat{A}$ is obtained:

$$\hat{a}_1 = 0.39 \tag{28}$$

$$\hat{a}_2 = 0.12 \tag{29}$$

$$\hat{a}_3 = 0.49 \tag{30}$$

$$\hat{b}_1 = 0.24 \tag{31}$$

$$\hat{b}_2 = 0.35 \tag{32}$$

$$\hat{b}_3 = 0.41 \tag{33}$$

$$\hat{c}_1 = 0.93 \tag{34}$$

$$\hat{c}_2 = 0.05 \tag{35}$$

$$\hat{c}_3 = 0.02 \tag{36}$$

$$\hat{d}_1 = 0.17 \tag{37}$$

$$\hat{d}_2 = 0.11 \tag{38}$$

$$\hat{d}_3 = 0.72 \tag{39}$$

In the result, $\hat{c}_1$ and $\hat{d}_3$ are large. Therefore, two IDCs are put on the node (c) and the node (d). This arrangement is the same as the best one of all combinations in the map. Let $\bar{\lambda}(A^*)$ denote the optimal solution of the problem (11) and $\bar{\lambda}(\hat{A}^*)$ denote the optimal solution of the relaxed problem (12). Then:

$$\bar{\lambda}(\hat{A}^*) = 0.9263 \tag{40}$$

$$\bar{\lambda}(A^*) = 0.9331 \tag{41}$$

and we can see that the difference between $\bar{\lambda}(\hat{A}^*)$ and $\bar{\lambda}(A^*)$ is quite small. This result shows that the optimal arrangement of IDCs can be obtained quickly by solving the relaxed problem.

5. Conclusion

In this work, we proposed a device and algorithm to implement autonomous robots that can enhance efficiency through autonomous knowledge sharing and acquisition system, referred to as an intelligent data carrier system. In conclusion, we accomplished the following:

- We introduced a device that enables local communication, the IDC.
- We implemented a system of autonomous guidance knowledge acquisition and sharing, carried out by autonomous mobile robots.
- We formulated behaviours of autonomous robots and suggested a method to evaluate the performance of the proposed algorithm.
- We formulated the arrangement problem of IDCs in a particular environment. We denote the method to solve the problem by means of relaxation.

References

[1] E. Yoshida, M. Yamamoto, T. Arai, J. Ota, & D. Kurabayashi, A design method of local communication area in multiple mobile robot system, *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, 1995, 2567–2562.

[2] E. Yoshida, T. Arai, M. Yamamoto, J. Ota, & D. Kurabayashi, Evaluating the efficiency of local and global communication in distributed mobile robotic systems, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Osaka, Japan, 1996, 1661–1666.

[3] T. Balch & R.C. Arkin, Communication in reactive multiagent robotic systems, *Autonomous Robots, 1*, 1994, 27–52.

[4] S. Ichikawa & F. Hara, Experimental characteristics of multiple-robots behaviors in communication network expansion and object fetching, *Distributed Autonomous Robotic Systems, 2*, 1996, 183–194.

[5] A. Drogoul & J. Feber, From Tom Thumb to the Dockers: Some experiments with foraging robots, *From Animals to Animats, 2*, 1992, 451–459.

[6] T. Fujii, H. Asama, T. Numers, T. Fujita, H. Kaetsu, & I. Endo, Co-evolution of a multiple autonomous robot system and its working environment via intelligent local information storage, *Robotics and Autonomous Systems, 19*, 1996, 1–13.

[7] D. Kurabayshi & H. Asama, Knowledge sharing and cooperation of autonomous robots by intelligent data carrier system, *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, 2000, 464–469.

[8] Y. Yoshimura, J. Ota, K. Inoue, D. Kurabayashi, & T. Arai, Iterative transportation planning of multiple objects by cooperative mobile robots, *Distributed Autonomous Robotics Systems, 2*, 1996, 171–182.

[9] H. Asama, M. Sato, L. Bogni, H. Kaetsu, A. Masumoto, & I. Endo, Development of an omni-directional mobile robot with 3 DOF decoupling drive mechanism, *Proc. Int. Conf. on Robotics and Automation*, Nagoya, Japan, 1995, 1925–1930.

[10] Y. Arai, S. Suzuki, S. Kotosaka, H. Asama, H. Kaetsu, & I. Endo, Collision avoidance among multiple autonomous mobile robots using LOCISS (LOcally Communicable Infrared Sensory System), *Proc. of IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, 1996, 2091–2096.

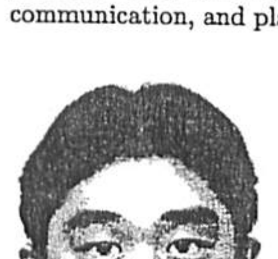
[11] D. Kurabayashi & H. Asama, Autonomous knowledge acquisition and revision by intelligent data carriers in a dynamic environment, *Robotics and Mechatronics, 13*(2), 2001, 154-159.

[12] G.L. Nemhauser, A.H.G. Rinnooy Kan, & M.J. Todd, editors *Optimization*, volume 1 of Handbooks in Operations Research and Management Science. (North-Holland, 1989).

[13] Y.E. Nesterov & A.S. Nemirovskii, *Interior-point polynomial methods in convex programming: Theory and Algorithms* (Philadelphia, USA: SIAM Publications. SIAM, 1993).

Biographies

Daisuke Kurabayashi received the M.E. and Ph.D. degrees from the Faculty of Engineering, the University of Tokyo, in 1995 and 1998, respectively. From 1998 to 2001 he worked at the Institute of Physical and Chemical research (RIKEN) as a postdoctoral researcher. Currently he is a lecturer in the Tokyo Institute of Technology. His research concentrates on cooperative motion, communication, and planning of autonomous agents.



Katsumi Konishi received the B.S.C., M.S.C., and Ph.D. degrees in engineering from the university of Tokyo in 1996, 1998, and 2001, respectively. He is currently a Research Associate at the Faculty of Engineering, Science University of Tokyo. His current research interests include a global optimization approach for nonconvec and its application to control system design.

Hajime Asama received the Be., M.E., and Ph.D. degrees from the University of Tokyo in 1982, 1984, and 1989, respectively. He jointed the Chemical Engineering Laboratory of the Institute of Physical and Chemical Research (RIKEN) in 1986, and is currently a Senior Scientist of the Department Research Fundamentals Technology, and the Leader of the Extreme Conditions Mechatronics Team of RIKEN. He is also a Visiting Associate Professor at Saitama University, Japan. He received the Promotion of Advance Automation Technology Award in 1992, JSME Robomec Award in 1995, JSME Best Poster Award in 1996, RoboCup 97 Engineering Challenge Award, and RoboCup98 Japan Open JSAI Award as a member of UTTORI United Team. He edited the monograph *Distributed Autonomous Robotic Systems* (Vols. 1 and 2) (Springer-Verlag, 1994, 1996). His main interests are distributed autonomous robotics systems, cooperation of multiple autonomous mobile roots, maintenance robots, and intelligent bioprocess systems. He is a member of RSJ, JSME, IEEE and the New York Academy of Science among other organizations.