Effective Map Building Using a Wave Algorithm in a Multi-Robot System

Dilshat Saitov[1], **Ulugbek Umirov**[2], **Jung Il Park**[2], **Jung Won Choi**[3] and **Suk Gyu Lee**[1,#]
1 Department of Electrical Engineering, Yeungnam University, 214-1, Daedong, Gyongsan-si, Gyongbuk, Korea, 712-749
2 School of Electrical Engineering and Computer Sciences, Yeungnam University, 214-1, Daedong, Gyongsan-si, Gyongbuk, Korea, 712-749
3 Department of Electronic Engineering, Kumoh National University, Sinpung-dong, Gumi-si, Korea, 730-701
Corresponding Author / E-mail: sglee@ynu.ac.kr, TEL: +82-53-810-2487, FAX: +82-53-810-4767

KEYWORDS : Wave algorithm, Mapping, Simulation, Mobile robot

Robotics and artificial intelligence are components of IT that involve networks, electrical and electronic engineering, and wireless communication. We consider an algorithm for efficient navigation by building a precise map in a multi-robot system under conditions of limited and unlimited communications. The basis of the navigation algorithm described in this paper is a wave algorithm, which is effective in obtaining an accurate map. Each robot in a multi-robot system has its own task such as building a map for its local position. By combining their data into a shared map, the robots can actively seek to verify their relative locations. Using shared maps, they coordinate their exploration strategies to maximize exploration efficiency. To prove the efficiency of the proposed technique, we compared the final results with the results in Burgard[8] *and Stachniss.*[9-10] *All of the simulation comparisons, which are shown as graphs, were made in four different environments.*

Manuscript received: / Accepted: October 23, 2007 / Accepted: January 25, 2008

1. Introduction

A great deal research has been conducted on the subject of map-building by autonomous mobile robots. An autonomous mobile robot works well if it is equipped with various kinds of expensive sensors, but it may prove to be too expensive and have less fault tolerance. Several trade-offs and issues exist concerning the acquisition of accurate information, the enhancement of the system's fault tolerance, and the reduction of the system cost. To cope with dynamic environments and multiple tasks, research has recently focused on multi-agent and multiple distributed autonomous robotic systems. Since multi-robot systems are characterized by distributed control, autonomy, enhanced fault tolerance, and communication, we decided to use a team of mobile robots for building a map in our simulation. Each robot is assigned a task such as building a map of its local position, and the robots combine their data into shared maps. The robots coordinate their exploration strategies to maximize the efficiency of their exploration using these shared maps. In our multiple simulations, we tried to create a more flexible, robust, and efficient system. A multi-robot system is better able to accomplish the map-building task because the robots cooperate and assemble a more accurate map by combining all their data.

Two main approaches are used in building a map. The first one is simultaneous localization and mapping (SLAM)[1,4] based on a Kalman filter. The map it produces presents the posterior probability of the location of some features or landmarks that can be detected by a robot while exploring the environment.

The second approach is the so-called expectation–maximization (EM) technique.[3] In this method, the mapping task is solved using an algorithm based on the EM principle. This method is quite different from the previously described Kalman filter algorithm approach since EM-based mapping does not produce a full posterior, but rather the most likely map. It also has some drawbacks, such as the inability to generate maps incrementally because of its iterative nature. However, it could be useful for mapping large areas due to its insensitivity to the data association problem.

The algorithms of navigation and mapping algorithms that were considered in Stachniss[10] attracted our attention because the basis of the authors' algorithm in Burgard,[8] Stachniss[9-10] and our algorithm in Saitov[11] are the same. What the authors in Burgard[8] and Stachniss[9-10] called "frontier cells," we called the "wave algorithm." After a careful examination, we determined that no difference exists between these two algorithms. It is no coincidence that they were developed independently because in our opinion, they are logically the most advantageous algorithms in navigation. We improved our algorithm and called it the "wave algorithm with directional priority," while the authors of Burgard[8] and Stachniss[9-10] also enhanced their algorithm and called it the "optimized method." We will describe the obvious differences of these methods in detail in the following sections.

A second reason why the navigation and mapping algorithms in Stachniss[10] attracted our attention is that they are easier to implement in real life than those in Dissanayake,[1] Pulford[3] and Williams.[4]

We take a completely different approach to the problem. Our method completely ignores the issue of relative individual robot positions. Our robots operate independently of each other for some time to generate individual local maps. After they have explored their areas, the small maps they have created can be merged together to form a global map. The key strength of this method is based not on features, but rather on occupancy grids, i.e., metric arrays in which the value in each cell indicates whether the related location is free space or part of an obstacle.

2. Wave Algorithm

We will describe the basic algorithm, which we call the wave algorithm.[4,5] The main idea of this exploration strategy is to move to the closest location where the robot can gather information about a cell that has not been sufficiently explored. We explain the strategy in the following figure. It provides short trajectories for single robot exploration tasks. First, a robot explores the area around itself and then seeks a minimum distance to the next point.

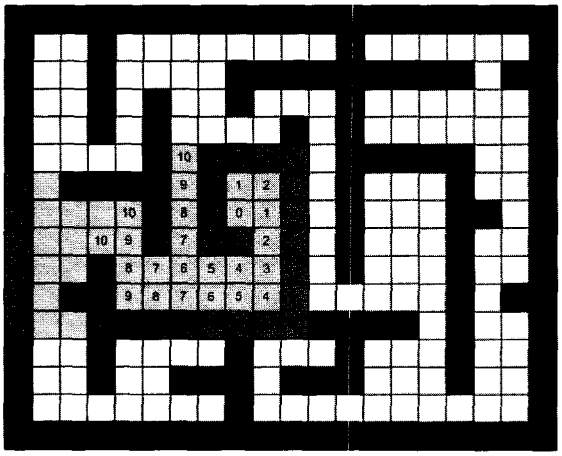


Fig. 1 Simulation of the wave algorithm

In Fig. 1, from its location in cell 0, a robot checks all the available cells around it and numbers each one from 1 to n. Then robot goes to the lowest numbered cell and checks the available cells again.

Table 1 Pseudocode of the wave algorithm

```
wM [ cX, cY ] = 0;
i = 0;
exist = false
do
{
  exist = false;
  for (x = 0; x < width; x ++)
    for (y = 0; y < width; y ++)
      if (dM [x, y] is open and free) and
          wM [x, y] is unassigned and
        neighbor of wM [x, y] is i)
      then (wM [x, y] = i +1; exist = true;)
  i = i + 1;
}
  while (exist);
```

3. Previous Research and Results

In our previous research,[11] we used the wave algorithm with directional priority and we compared our results to those in Stachniss.[10] We will describe the algorithm used in Stachniss[10] and then ours in more detail to show the efficiencies and inefficiencies of each. We considered three methods for map building according to Stachniss.[10]: the uncoordinated, the coordinated, and the optimized methods.

3.1 Uncoordinated method

In Stachniss,[10] the authors relied on value iteration, which is a type of dynamic programming algorithm used to determine the cost of reaching the current frontier cells. In their approach, the cost for traversing a grid cell (x, y) is proportional to its occupancy value $P(occ_{xy})$.The minimum-cost path is computed using the following two steps.

3.1.1 Initialization

The grid cell that contains the robot location is initialized to 0, and all others to ∞:[10]

$$V_{x,y} \leftarrow \begin{cases} 0, & if\langle x, y\rangle \, is \, the \, robot \, position \\ \infty, & otherwise \end{cases} \tag{1}$$

3.1.2 Update loop

For grid cells (x, y), the update loop has the form of Eq. (2):

$$V_{x,y} \leftarrow \min_{\substack{\Delta x=-1,0,1 \\ \Delta y=-1,0,1}} \left\{ V_{x+\Delta x, y+\Delta y} + \sqrt{\Delta x^2 + \Delta y^2} \times P(occ_{x+\Delta x, y+\Delta y}) \right\} \tag{2}$$

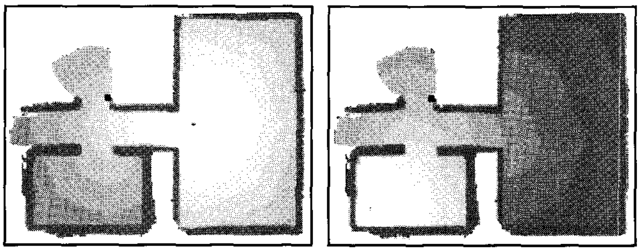


Fig. 2 Typical value functions obtained for two different robot positions. The black rectangle indicates the target points in the unknown area with minimum cost

3.2 Coordinated method

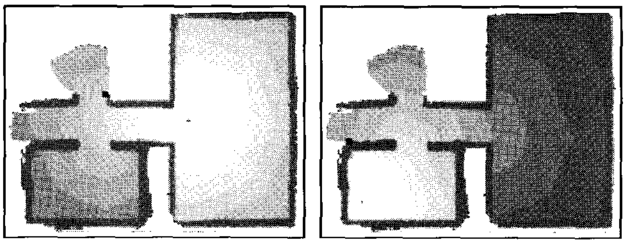


Fig. 3 Target positions obtained using the coordinated approach. In this case, the target point for the second robot is to the left in the corridor

Figure 3 shows the efficiency of the coordinated technique.[8] In this method, two robots with the same cost must choose different target cells, while in the uncoordinated technique, robots would choose the same target position (see Fig. 2). The robot on the left in Fig. 3 chooses to investigate the nearest point. Since robots constantly exchange data, robot 2 in the right-hand figure, knows the chosen destination of robot 1 and chooses a different nearest point.

3.3 Optimized method

The third approach considered in Stachniss[10] was the optimized method. Consider the situation shown in Fig. 4 in which two robots are investigating a corridor connecting two adjacent rooms. The shaded area has already been investigated. The assignment resulting from an application of their algorithm is depicted in the left image of this figure. Suppose both target points a and b have the same utility. In the first round the algorithm assigns robot 2 to a since this assignment has the least cost of all possible assignments. Therefore, in the second round, robot 1 is assigned to b. If we assume that both robots require the same amount of time to explore a room, this assignment is clearly suboptimal. A better assignment is shown in the right side of Fig. 4.

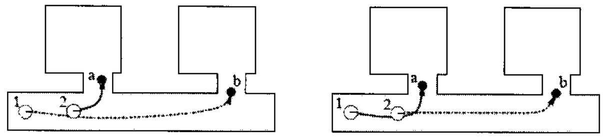


Fig. 4 Exploring robots with directional priority

The trajectories depicted in the left-hand image that result from algorithm 1 are suboptimal. If robot 1 moves to point *a* and robot 2 moves to the location *b* as illustrated in the right figure, the time needed to finish the exploration task decreases because the maximum time needed to reach the rooms is less.

By directing robot 1 to the left room and robot 2 to room on the right, the two-robot team can finish the job more quickly because the time required to reach the rooms is reduced. As noted above, one way to overcome this problem is to consider all possible combinations of target points and robots. The authors in Stachniss[10] wanted to minimize the trade-off between the utility of frontier cells and the distance to be traveled. However, just adding the distances to be traveled by the two robots does not make a difference in situations like that shown in Fig. 4. To minimize the completion time, we therefore modify the evaluation function so that it considers squared distances in choosing target locations:

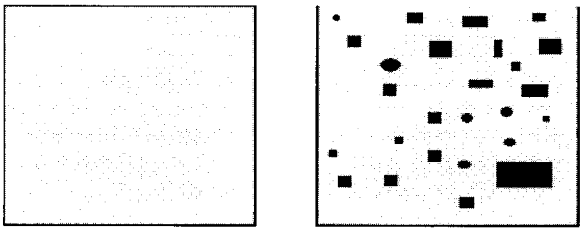
$$\arg\max_{(t_1,\ldots\ldots,t_n)} \sum_{i=1}^{n} [U(t_i | t_1,\ldots.,t_{i-1},t_{i+1},\ldots..,t_n) - \beta \times (V_{t_i}^{i})^2] \quad . \quad (3)$$

Here, $t_{1 \ldots\ldots} t_n$ are cells that have already been investigated, U is the utility of frontier cell t_n, V is the cost for obtaining frontier cell, i is a specific robot, and β is the importance of U versus V.

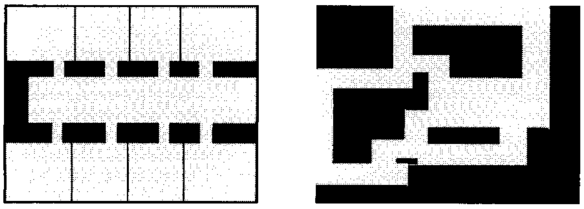
4. Wave Algorithm with Directional Priority

In this section, we describe the algorithm that we presented in Saitov.[11] In general, the wave algorithm uses costs and frontier cells when choosing the best target point, just like the uncoordinated algorithm.[10] However, instead of the optimized method described in 4.3, we assume that each robot has its own direction on the plane. This means that the probability of them choosing the same point is very low right from the start. Notice that in Saitov[11] we assumed that the communicational range was unlimited.

Figure 5 depicts four different environments for simulations.



A) Empty environment B) Unstructured environment



C) Office environment D) Corridor environment

Fig. 5 Maps used for the simulation experiments

The following screenshots of simulations are based on the algorithm described in this paper. All nine robots have the same range for their sensors. The size of the map is 200 × 200 pixels.

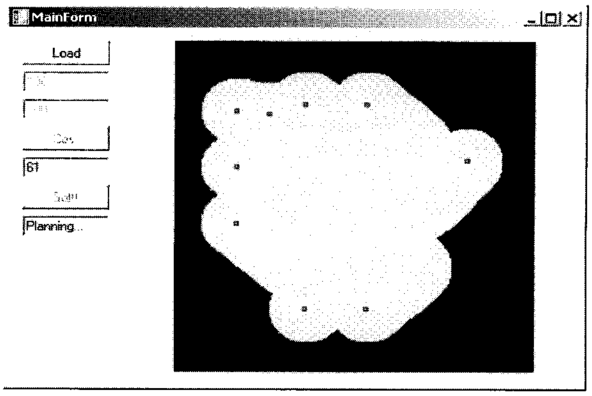


Fig. 6 Exploration of an empty environment

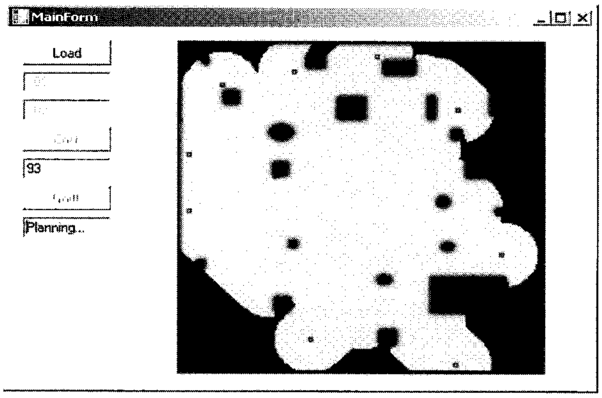


Fig. 7 Exploration of an unstructured environment

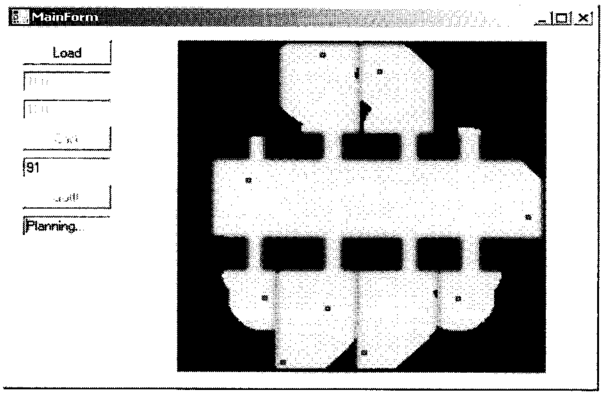


Fig. 8 Exploration of an office environment

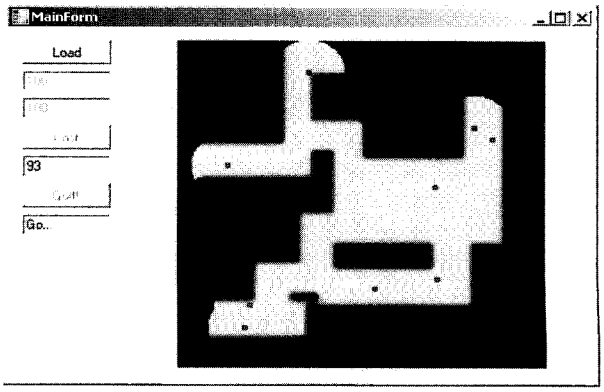


Fig. 9 Exploration of a corridor environment

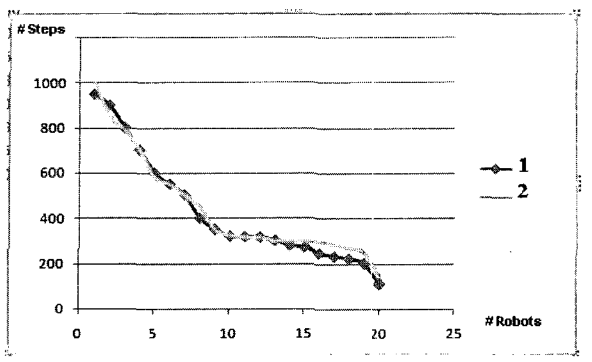


Fig. 10 Comparison of the efficiency of the algorithms (empty environment)

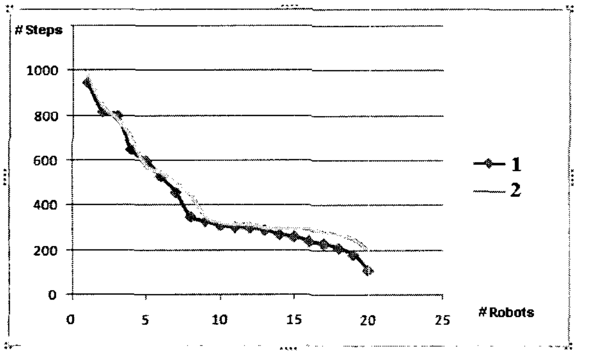


Fig. 11 Comparison of the efficiency of the algorithms (unstructured environment)

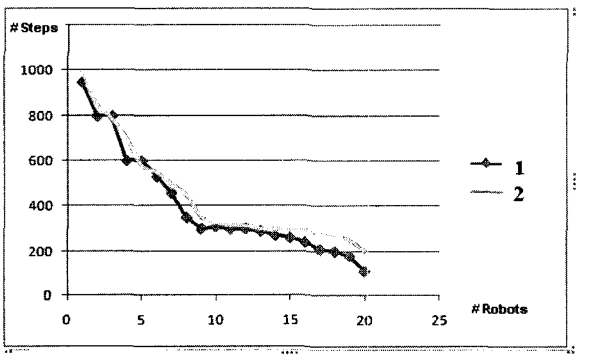


Fig. 12 Comparison of the efficiency of the algorithms (office environment)

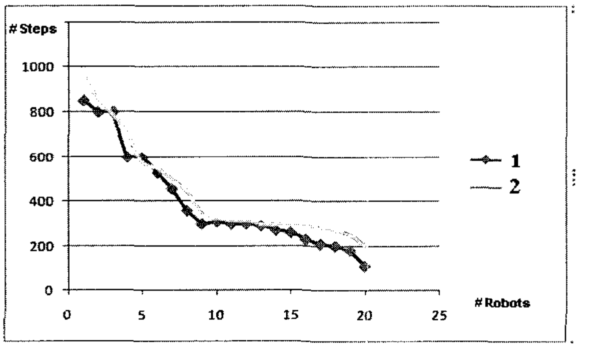


Fig. 13 Comparison of the efficiency of the algorithms (corridor environment)

Figures 10–13 show comparisons of the optimized method Stachniss[10] and the wave algorithm with directional priority Saitov.[11] These figures show the number of steps as a function of the number of robots. The line with squares marked "1" represents the improved algorithm based on the wave algorithm Saitov,[11] while the plain line marked "2" is for the algorithm used in Stachniss[9-10] In Fig. 13, the values of the simulation results are approximately the same. Figures 11–13, however, are different. If we compare the number of steps for any number of robots, we can see that our algorithm is more efficient.

5. Wave Algorithm with Directional Priority and Limited Communication

Supporting a permanent static or dynamic wireless network in the real world can be problematic, particularly for mobile robots in confined areas. We extended our research to consider our proposed algorithm with limited communication. For this, we used map merging, in which we consider each explored part of a map collected by a robot to be a separated matrix.[6] Let us assume that A and B are two positive real numbers. A * B is a function such that

$$\text{m: } [0, \text{A}] * [0, \text{B}] \rightarrow \text{R}. \tag{4}$$

In Eq. (4), A and B represent rows and columns of the matrix, respectively. The next step is the definition of a transformation used to try different relative placements of two maps to find a good merge. We assume that the location of a point in the plane is expressed in homogeneous coordinates; that is, the point (x, y) is represented by the vector $[x\ y\ 1]^{\text{T}}$, where the trailing superscript T indicates the transpose operation.

Let t_x, t_y, and φ be three real numbers. The transformation associated with t_x, t_y and φ is the function

$$T\ t_x, t_y \varphi(x, y) : R^2 \rightarrow R^2 \tag{5}$$

defined as follows:

$$\text{T}\,\text{t}_\text{x}, \text{t}_\text{y}\varphi(\text{x}, \text{y}) : \begin{bmatrix} \cos\varphi & -\sin\varphi & \text{t}_\text{x} \\ \sin\varphi & \cos\varphi & \text{t}_\text{y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \text{x} \\ \text{y} \\ 1 \end{bmatrix} \tag{6}$$

Here, the transformation given in (5) corresponds to a rotation about the origin of the point (x, y) of φ, followed by a translation by $(\text{t}_\text{x}, \text{t}_\text{y})$. In the following sections, we see in the results that the mapping algorithms produce occupancy grids in which the rotation and translation transformations are apparently sufficient for merging real-world data.

5.1 Pairwise map merging

Building a shared map using two small maps requires the best possible overlap between these two maps. Let m_1 and m_2 be two maps in I_{A*B}. The overlap between the maps can be described in the following equation:

$$\omega(m_1, m_2) = \sum_{i=0}^{A-1} \sum_{j=0}^{B-1} Eq(m_1[i, j]; m_2[i, j]) \tag{7}$$

where Eq $(a;b)$ is 1 when $a = b$, and 0 otherwise. The overlapping function ω measures how much the two maps agree.

In the following figures, we see screenshots of the simulation results for exploring the map. Note that in these simulations, the robots have limited communication. Under such conditions, we used the method of merging local maps in Birk.[6]

Figures 18–20 show the exploration time as a function of the communication range for different numbers of robots. In case of two, three, and four robots, we observe that the communication range and the exploration time are closely related to the number of robots. Note, however, that the effectiveness of the algorithm with limited communication directly depends on the size of the area and the number of robots.

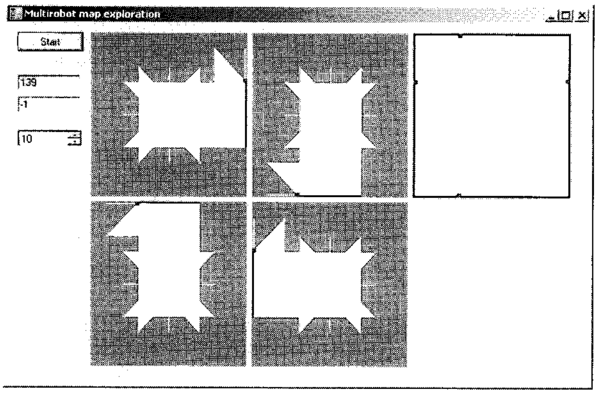


Fig. 14 Directional priority algorithm with limited communication applied to the empty environment

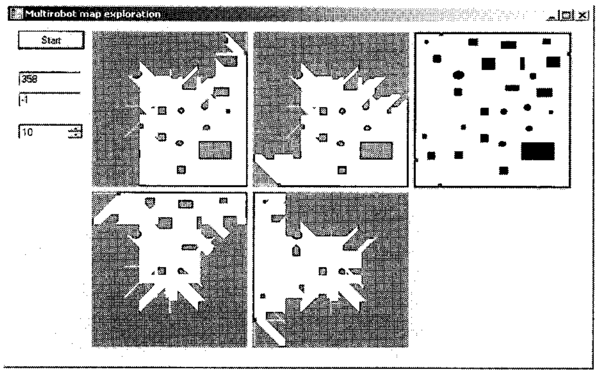


Fig. 15 Directional priority algorithm with limited communication applied to an unstructured environment

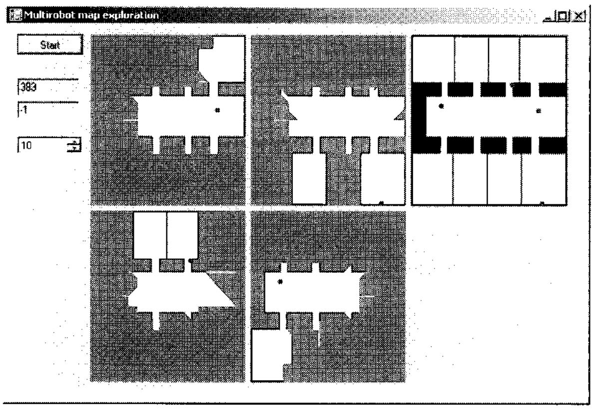


Fig. 16 Directional priority algorithm with limited communication applied to the office environment

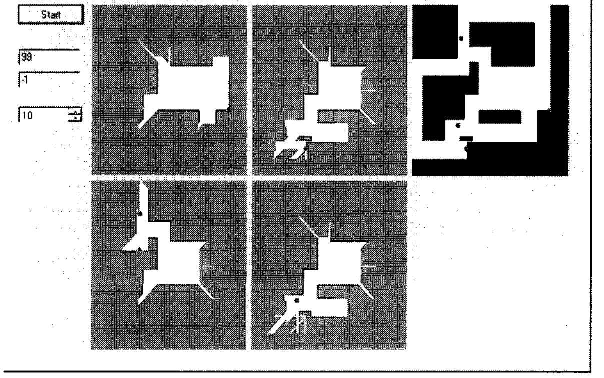


Fig. 17 Directional priority algorithm with limited communication applied to the corridor environment

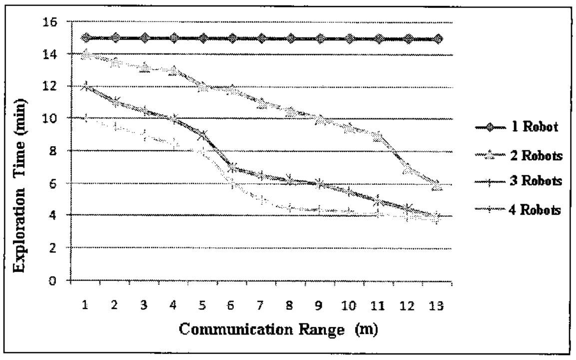


Fig. 18 Exploration time as a function of communication range for different numbers of robots in an unstructured environment

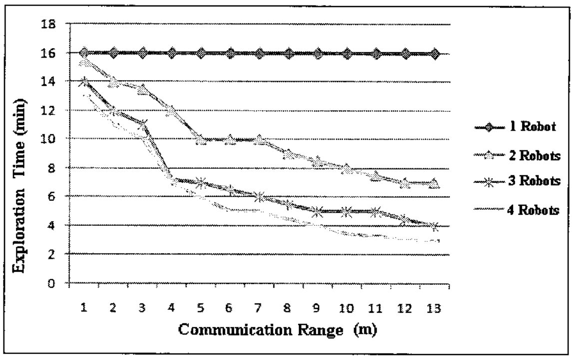


Fig. 19 Exploration time as a function of communication range for different numbers of robots in an office environment

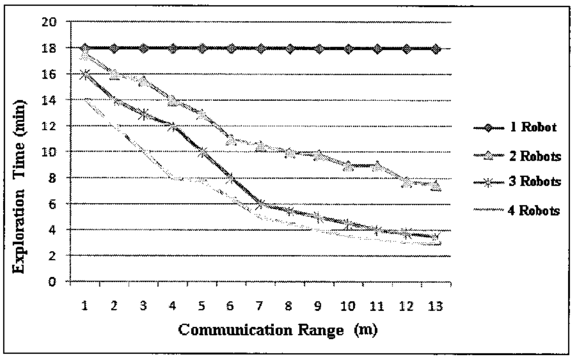


Fig. 20 Exploration time as a function of communication range for different numbers of robots in the corridor environment

6. Conclusions

We presented a distributed approach to mobile robot mapping and exploration. This system enables teams of robots to explore the environment more efficiently from either known or unknown locations. The robots explore independently until they can communicate with each other. When they meet, they exchange sensor information with each other, share maps of the areas they have explored, and build one shared map. According to our simulation results, this approach might be much more useful for map building in real-world multi-robot systems than the approach in Stachniss.[10]

In further research, we plan to improve the proposed algorithm by adding neural networks and fuzzy logic system methods of map building and navigation in uninvestigated areas.

ACKNOWLEGEMENT

This research was supported by Yeungnam University research grants in 2007.

REFERENCES

1. Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H. and Csorba, M., “BA solution to the simultaneous localization and map building (SLAM) problem,” IEEE Transactions on Robotics and Automation, Vol. 17, No. 3, pp. 229-241, 2001.
2. Ko, J., Stewart, B., Fox, D., Konolige, K. and Limketkai, B., “BA practical, decision-theoretic approach to multi-robot mapping and exploration,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3232-3238, 2003.
3. Pulford, G. W. and La Scala, B. F., “Map estimation of target maneuver sequence with expectation–maximization algorithm,” IEEE Transactions on Aerospace and Electronic Systems, Vol. 38, No. 2, pp. 367-377, 2002.
4. Williams, S., Dissanayake, G. and Durrant-Whyte, H., “Towards multi-vehicle simultaneous localization and mapping,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2743-2748, 2002.
5. Yamauchi, B., “Frontier-based exploration using multiple robots,” in Proceedings of the Second International Conference on Autonomous Agents, pp. 47-53, 1998.
6. Birk, A. and Carpin, S., “Merging occupancy grid maps from multiple robots,” IEEE Transactions on Robotics and Automation, Vol. 94, No. 7, pp. 1384-1397, 2006.
7. Roumeliotis, S. and Bekey, G., “Distributed multirobot localization,” IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, pp. 781-795, 2002.
8. Burgard, W., Moors, M. and Schneider, F., “Collaborative exploration of unknown environments with teams of mobile robots,” IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, pp. 781-795, 2002.
9. Stachniss, C., Mozos, O. M. and Burgard, W., “Speeding-up multi-robot exploration by considering semantic place information,” IEEE Transactions on Robotics and Automation, Vol. 18, No. 5, pp. 581-587, 2006.
10. Stachniss, C., Mozos, O. M. and Burgard, W., “Coordinated multi-robot exploration,” IEEE Transactions on Robotics and Automation, Vol. 21, No. 3, pp. 376-386, 2005.
11. Saitov, D., Umirov, U., Lee, S. G. and Park, J. I., “Enhanced map building using wave algorithm in multi-robot systems,” International Symposium on Mechatronics and Automatic Control Section (ISMA), pp. 116-122, 2007.
12. Sebastian, B., Ulrich, W., Michael, B. and Thorsten, S., “Multi robot path planning for dynamic environments; a case study,” International Conference on Intelligent Robots, pp. 1245-1250, 2001.
13. Peter, C., Ron, P. and Daniela, R., “Localization and Navigation Assisted by Networked Cooperating Sensors and Robots,” International Journal of Robotics Research, Vol. 24, Issue 9, pp. 771-786, 2005.
14. Raj, M., Kingsley, F. and Lynne, E. P., “Distributed cooperative outdoor multirobot localization and mapping,” Autonomous Robots, Vol. 17, Issue 1, pp. 23-39, 2004.
15. Ioannis, R., Gregory, D. and Evangelos, M., “Multi-robot collaboration for robust exploration,” Annals of Mathematics and Artificial Intelligence, Vol. 31, Issue 1-4, pp. 7-40, 2001.
16. Weihua, S., Qingyan, Y., Jingdong, T. and Ning, X., “Distributed multi-robot coordination in area exploration,” Robotics and Autonomous Systems, Vol. 54, Issue 12, pp. 945-955, 2006.