Proc. of the 2nd International Symposium on Wireless Pervasive Computing 2007 (ISWPC2007), DVD-ROM (Feb. 2007).

A Route Planning Method Using *Cost Map* for Mobile Sensor Nodes

Masaki Nakamiya, Yasue Kishino, Tsutomu Terada, and Shojiro Nishio
Graduate School of Information Science and Technology, Osaka University, Japan.
Email:{nakamiya.masaki, yasue, tsutomu, nishio}@ist.osaka-u.ac.jp

***Abstract*— In this paper, we propose a route planning method for mobile sensor nodes using *cost map*. The proposed method achieves novel route planning that can solve several practical problems from previous works: the limitations of sensing ranges, obstacles on the node's routes, and restriction of their movements. This method can determine the route that has the lowest power consumption. We also propose a sensing area defining method for dealing with many kinds of sensors. Furthermore, we compared the proposed method to A*algorithm, a well-known route planning algorithm. We implemented a prototype of the sensor node to verify our algorithm in a real environment.**

I. Introduction

In recent years, the development of radio communication technologies and the miniaturization of electrical devices has made possible the development of sensor networks. Sensor networks consist of sensor nodes that have radio communication ability [7]. Currently, sensor networks using mobile sensor nodes that can migrate freely with actuators are receiving a lot of attention [2], [3]. In such networks, mobile sensor nodes can acquire sensing data according to dynamically changing demands, and they can travel in danger zones where people cannot, such as a subway station with a gas leak.

However, most previous works do not address the problems encountered in practical use. For example, though sensors have various shapes of sensing ranges, in most cases, the sensing range is defined as a simple circular shape. In addition, since each node has its own movement characteristic based on its driving method, the best route to a destination should be determined taking those characteristics into account. Furthermore, most conventional work on mobile sensor networks does not consider obstacles on the way to a destination and the influences of the ground condition. Although, in robotics research, some methods have been proposed to account for these factors, each method can perform only for a particular type of node. Thus, it is difficult to acquire suitable routes for various types of nodes in practical environments.

In this paper, we propose a route planning method for mobile sensor nodes by using *cost map*. Moreover, the proposed method defines the sensing area using four parameters. We clarified the effectiveness of the proposed method in performance evaluation by comparing it with A*algorithm. We also implemented prototypes of sensor nodes to verify our proposed method in actual environments.

II. Proposed Method

In this study, we assume a static environment where the ground condition does not change dynamically. The field condition, which means obstacles and influences of the ground condition, is allocated to a field map represented as a two-dimensional grid. We call one grid *cell*. *Ground cost*, which is the extra cost to pass the cell, is set for each cell on the field map. Furthermore, users request various kinds of sensing data at various places continuously, and several nodes cooperate and move for sensing according to these requests.

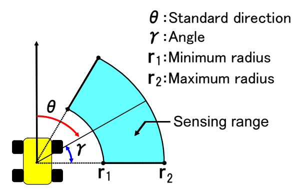


Fig. 1. Sensing model

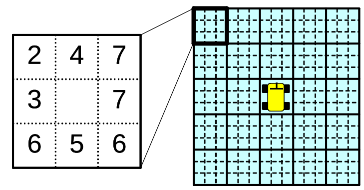


Fig. 2. Example of base cost

We assume various kinds of nodes, such as a car-type node, a caterpillar-type node, and a node with two legs. Since each type of node has its own movement characteristics and the suitability of a node varies depending on geographical features and users' requirements, we assume various types of nodes simultaneously for mobile sensor networks. In addition, in order to satisfy users' various requests, a sensor node may have various sensors. In light of this, we assume a mobile sensor networks using various types of mobile nodes mounted with various kinds of sensors.

A. Sensing model

Some sensors, such as thermometers, hygrometers, noise meters, and barometers, can sense all in directions. Other sensors, such as infrared, distance, light, and touch sensors, can only sense in limited directions. In this paper, the sensing model is defined by four parameters as shown in Fig. 1. This definition of the sensing model enables our algorithm to use many kinds of sensors. Our method allocates a route for the sensor node, considering both sensing ranges and the directions of sensors.

B. Route planning method

The proposed method is based on A*algorithm, which is widely used for route planning in the field of artificial intelligence. A*algorithm can determine the route with the lowest cost. It reduces the amount of calculation necessary by selecting the next cell, which is assumed to be the closest to the destination. In addition, it is has been proven that A*algorithm can always find the lowest cost route if there is a solution[1].

(a) Base cost A (b) Base cost B

Fig. 3. Base cost that direction of node is not considered

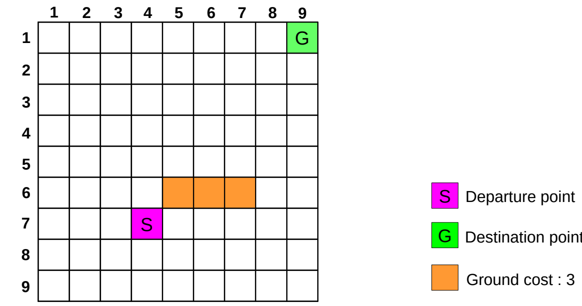


Fig. 4. Example of field map

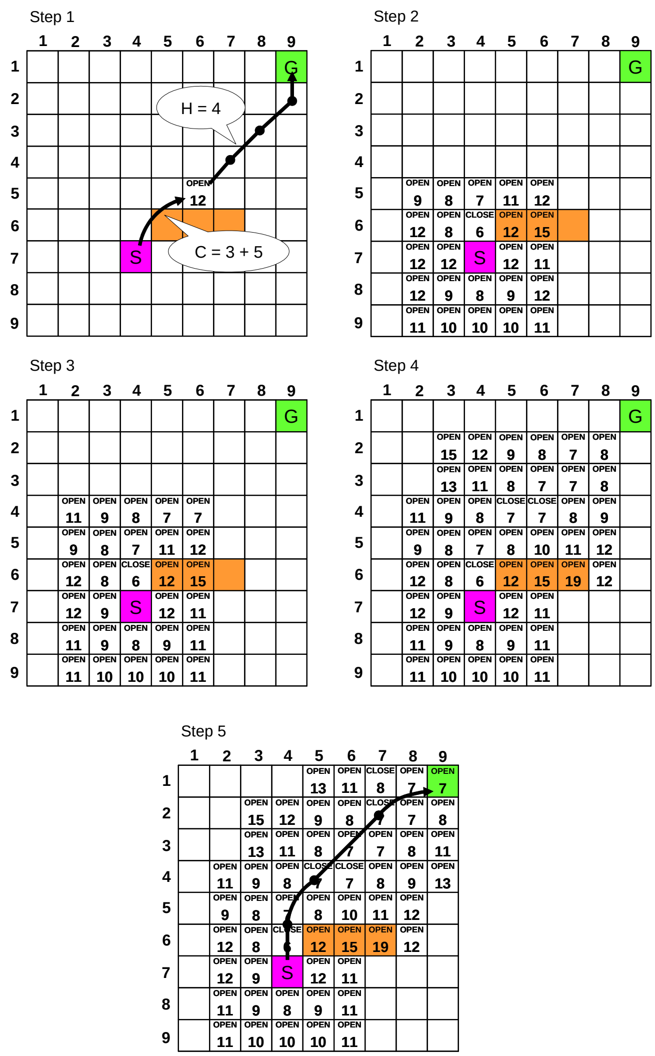


Fig. 5. Example of route planning

In the following section, we explain the proposed method through a comparison with the A*algorithm. The proposed method determines a route from a departure point to a destination point on which power consumption is lowest in consideration of geographical features, obstacles, and movement characteristics of the node.

Base cost

Base cost represents actual power consumption of a node for moving to surrounding cells without ground influences. In our method, we measure it with an actual node in advance. As shown in Fig. 2, the costs are measured in each of eight directions for each cell. The figure represents an example of a 5×5 base cost of a car-type node, and it means that the node needs six unit costs when it moves to the upper-left corner and turns to the lower-left direction at the cell. Figure 3 shows an example of the measured base cost (the direction is abbreviated for simplicity). As shown in Fig. 3(b), we also measure a base cost in which the beginning direction of the node is upper-right.

Cost map

The proposed method determines a route with the lowest power consumption by constructing a *cost map* from the base cost and the ground cost. In this work, *cost* means electric power consumption of the sensor node. A cost map is a map that shows the cost to reach the cell and the distance to the destination on each cell. We call the sum of these two values $Score$. There are three differences between the proposed algorithm and A*algorithm.

- The proposed method measures the $Score$ for eight directions of each cell.
- While A*algorithm calculates the $Score$ in neighboring cells of a selected cell, the proposed method overlays the center of the base cost on a selected cell and calculates the $Scores$ in the overlaid area.
- The proposed method uses the two base costs properly according to the beginning directions of the node.

Ground cost

The ground cost represents the extra cost needed to pass over a cell due to geographical influences such as roughness and the slant of fields. When the proposed method calculates the cost to reach a certain cell, it adds the ground cost in all cells over which a node passes until it reaches the target cell. By this operation, our method realizes route planning considering geographical influence. The ground cost is arranged on a field map. An infinite value of the ground cost means that no node can enter the place.

Calculation steps

The proposed method constructs a cost map using a field map such as that shown in Fig. 4 and base costs such as those shown in Fig. 3. Figure 5 shows an example of route planning.

In the figure, the direction of the node is not considered for the sake of simplicity.

For route planning, our method calculates the $Score$ of each cell using the following equation, the same as the one for A*algorithm.

$$Score = C + H$$

C shows the total cost required for the node to move from a departure point to the cell, and H shows the expected cost to the destination. H is calculated as the cost required for the node to move from the cell to the destination without ground cost. For calculation, we divide the cells into three states.

NO_CALCULATE: Cells whose $Score$ is not calculated
OPEN: Cells whose $Score$ has already been calculated
CLOSED: Cell for which the shortest route to the cell has already been calculated

In the following, we describe the details of the proposed method using the example in Fig. 5. In this, the cost is described in cost units. One cost unit represents the cost required for a node to move forward the distance of a cell's width.

Step 1: The proposed method overlays the departure point of the cost map with the center of the base cost, calculates the $Scores$ in the overlaid area, and sets the states of the cells in the overlaid area as OPEN. At the departure point, the node faces in the upper direction, and it uses base cost A at first. For example, on cell (6, 5), C is eight unit costs, which is the sum of three unit costs by base cost A and five unit costs by the ground cost, and H is four unit costs, which is the number of cells to the destination. The $Score$ of cell (6, 5) becomes 12 unit costs, which is the total of C and H.

Step 2: The method selects cell (4, 6), which has the lowest $Score$ of all OPEN cells, and it changes the state of cell (4, 6) to CLOSED.

Step 3: The method overlays the center of the base cost to the cell (4, 6), calculates the $Scores$ in the overlaid area, changes the state of the cells to OPEN, and records the selected cell (4, 6) as the previous cell. When the node moves to the cell from the departure point, it faces in the upper direction. Thus, it uses base cost A. On cells (2, 8), (3, 7), and (6, 8), the new $Score$ is lower than the previous $Score$. In this case, the $Score$ and information of the last cell passed are updated.

Step 4: The method selects cell (6, 4), which has the lowest $Score$ of all OPEN cells, and sets the state of cell (6, 4) to CLOSED. Then, it calculates the $Score$ of the cells overlaid by base cost, and changes the states of these cells to OPEN. When the node moves to the cell from the last cell passed, it faces in the upper-right direction. Thus, it uses base cost B.

Step 5: This procedure is repeated until the destination cell is selected and its state is set to CLOSED. Since each cell stores the coordinate of the last passed cell, we can trace the best route.

(a) Caterpillar-type node (b) Car-type node

Fig. 6. Prototypes of sensor nodes

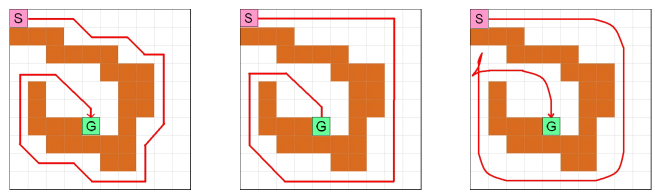
2.2	5.9	6.9	4.9	1.2	4.9	6.9	5.9	2.2
4.3		6.0	5.1		5.1	6.0		4.3
7.2	8.6	7.3	6.1	7.0	6.1	7.3	8.6	7.2
5.5	7.5	3.4	3.7	0.0	3.7	3.4	7.5	5.5
5.5		7.4	3.5		3.5	7.4		5.5
9.6	9.5	10.0	7.7	7.9	7.7	10.0	9.5	9.6
6.5	5.9	4.9	1.8	1.2	1.8	4.9	5.9	6.5
3.9		4.3	5.1		5.1	4.3		3.9
11.6	10.2	8.0	5.3	8.7	5.3	8.0	10.2	11.6

(a) Base cost A

5.0	4.2	7.8	4.7	1.2	5.1	5.0	4.9	1.5
6.9		6.9	5.4		5.7	6.1		4.9
8.6	4.5	7.8	8.6	6.4	5.8	5.4	6.1	5.0
5.9	3.4	5.1	3.5	3.4	0.0	5.8	5.7	5.1
5.3		4.2	7.7		3.4	6.4		1.2
6.2	6.9	5.1	8.6	7.7	3.5	8.6	5.4	4.7
5.1	1.8	1.5	5.1	4.2	5.1	7.8	6.9	7.8
6.9		1.8	6.9		3.4	4.5		4.2
8.5	6.9	5.1	6.2	5.3	5.9	8.6	6.9	5.0

(b) Base cost B

Fig. 7. Example of measured base cost of car-type node



(a) A*algorithm (b) Caterpillar-type node (c) Car-type node

Fig. 8. Results of route planning
(S: departure point, G: destination point)

Actually, each cell has the costs of eight directions, its status, and the previously passed cell. In Step 4, the method selects the lowest cost cell, whose direction cost is the lowest in all cells. At the destination point, the sensing range of the node must cover the requested sensing area.

III. Performance Evaluation

We evaluated the proposed route planning method by comparing it with A*algorithm. We used several prototypes of sensor nodes to verify the proposed method in actual environments. We assumed an environment where there are cells representing obstacles, such as a wall, whose ground costs are infinite. We assumed two types of nodes. One was a caterpillar-type node that can rotate and move forward and back. The other was a car-type node that can turn at an angle adjusted by the front wheels. We developed both nodes with MindStorm (LEGO company)[8], as shown in Fig. 6, and measured the base costs in 5×5 and 3×3 cells. An example of the base cost of the car-type node is shown in Fig. 7.

A. Comparison of route planning methods

Figure 8 shows an example of route planning results using A*algorithm (Fig. 8(a)), the proposed method using a

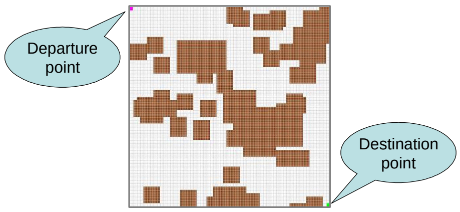


Fig. 9. Example of geographical features arranged at random

caterpillar node (Fig. 8(b)), and the proposed method using a car node (Fig. 8(c)). Although A*algorithm can determine one of the shortest routes, this may include extra rotation of the node as shown in Fig. 8. When a caterpillar-type node changes direction, it must stop, rotate, stop and restart. That extra rotation results in position error of the nodes and extra costs.

In Fig. 8(b), the node rotates fewer times than in the case of A*algorithm. This means the node suffers extra cost with A*algorithm though it plans the shortest route. The proposed method can consider the turning cost by using the base cost. In Fig. 8(c), the node can rotate at 90 degrees smoothly at a wide corner. However, in negotiating a narrow corner, it needs to move backward and forward several times. The proposed method can determine a route that avoids narrow corners as much as possible and reduces the extra cost. Since the movement characteristics of the nodes are not considered in A*algorithm, many narrow corners will be set on the route.

B. Energy consumption in different size of field map

We evaluated our method and A*algorithm in a simulation experiment. We used square field maps, and the sides of the field maps were changed from 60 cells to 100 cells in 10-cell increments. In the field map, square-shaped ground costs with widths of 5 cells and 10 cells were placed randomly as shown in Fig. 9. The departure point is the upper-left corner of the field map where the node's direction is lower-right, and the destination point is the most lower-right cell in the field map where node's direction is lower-right.

Figure 10 shows the simulation results using the caterpillar-type node. The results show that the proposed method can determine routes with lower costs than A*algorithm in all sizes of field maps. The reason is that our method can determine routes requiring fewer rotations than those of A*algorithm.

Figure 11 shows the results using a car-type node. In this case, the proposed method can also determine routes with lower costs than A*algorithm.

In addition, the difference between the two methods using the car-type node is larger than the difference with the caterpillar-type node. When the car-type node turns at 90 degrees around a narrow corner, the node needs to move backward and forward several times, incurring extra costs. Thus, a car-type node has a high possibility of being assigned as ineffective route in A*algorithm.

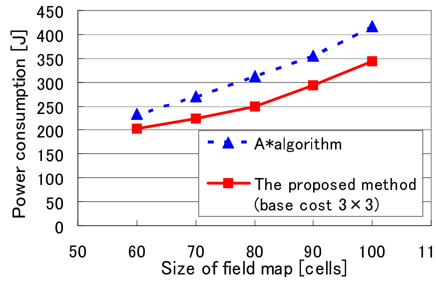


Fig. 10. Power consumption vs. size of field maps (caterpillar-type node)

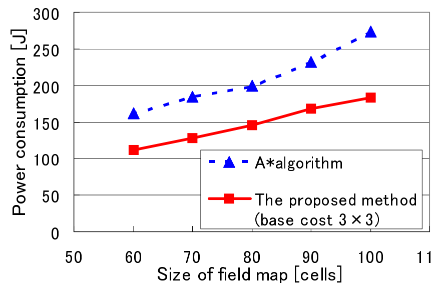


Fig. 11. Power consumption vs. size of field maps (car-type node)

TABLE I
POWER CONSUMPTION VS. OBSTACLE ARRANGEMENTS

		Random	Maze	Laboratory
Caterpillar	A*	416.8[J]	493.9[J]	250.6[J]
	3×3	343.7[J] (18%)	416.4[J] (16%)	228.4[J] (8.9%)
	5×5	334.1[J] (20%)	412.6[J] (17%)	227.4[J] (9.3%)
Car	A*	251.1[J]	366.0[J]	148.1[J]
	3×3	164.1[J] (35%)	225.9[J] (38%)	117.0[J] (21%)
	5×5	151.1[J] (40%)	194.8[J] (47%)	108.1[J] (27%)

C. Various environments

In this phase, we assume use of a caterpillar-type node and a car-type node and simulate route planning in the following three environments.

Random: Square ground costs are arranged at random
Maze: Ground costs are arranged like a maze (Fig. 12)
Laboratory: Tables, shelves, and walls are obstacles as in our laboratory (Fig. 13)

TABLE I shows the simulation results. The values in the case-arc show the ratio of improvement from the results obtained with A*algorithm. The results show that the proposed method performs effectively and reduces power consumption for both nodes in all environments. In particular, the improvement ratio is high in the case of the car-type node.

D. Size of base cost

TABLE I also shows the ratios of improvement in power consumption by using a 5×5 base cost compared with using a 3×3 base cost.

In the results of the car-type node, power consumption is reduced by using a wider base cost in all environments. This

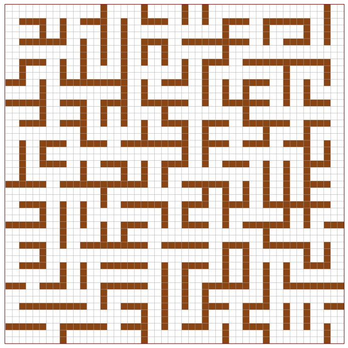


Fig. 12. Example of field map with maze configuration

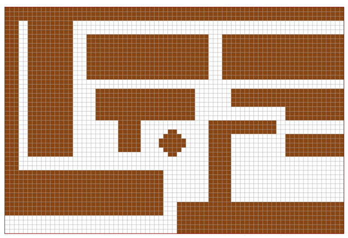


Fig. 13. Example of field map with our laboratory's configuration

is because in the case of the 5×5 base cost, more smooth and effective routes are determined than with the 3×3 base cost. In addition, the ratio of improvement is especially large in the maze-type field, since the aisles are narrow and the nodes must turn many times.

With the caterpillar-type node, the ratio of improvement is smaller than that of the car-type node. Since the caterpillar-type node can turn more easily than the car-type node, the route change from spreading the base cost is less, and the ratio of improvement becomes smaller.

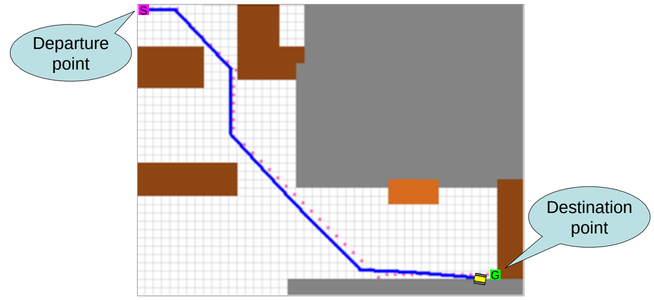
IV. Implementation

A. Mobile sensor nodes

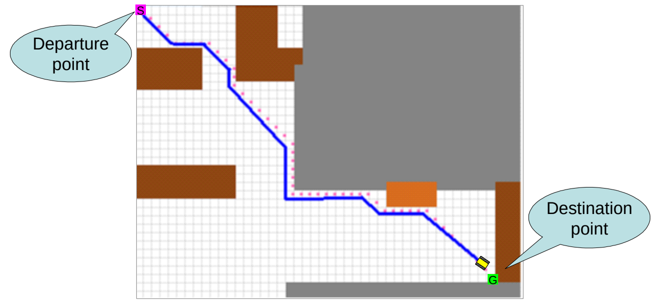
We developed prototypes of sensor nodes with Mindstorm RCX2 (LEGO Company). RCX2 received its program from a PC via IR communication and controlled its motors and sensors with a built-in 8 bit CPU. The caterpillar-type node we developed can control the right and left wheels with two motors. We can use three commands: go forward, go backward, and rotate and can control the duration of the actions as well.

B. Experiment

We verified the proposed method using the caterpillar-type node. Beforehand, we charted a field map on which desks, chairs, and walls, are obstacles and their ground costs are infinity. The node was in the upper direction at both the departure and destination points. We guided the node along the routes determined using A*algorithm and the proposed method.



(a) Proposed algorithm



(b) A*algorithm

Fig. 14. Mounting results

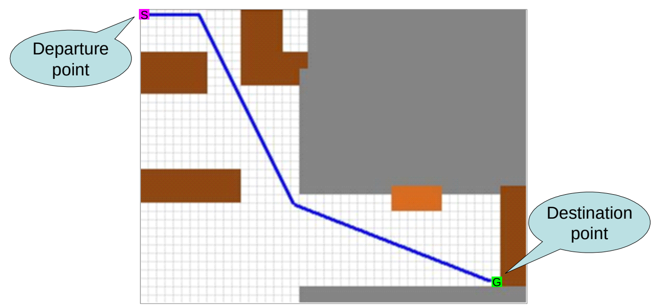


Fig. 15. Shortest route

TABLE II

Comparison of power consumption

	Experiment	Simulation	Error
Proposed method	115.5[J]	114.9[J]	0.6[J]
A*algorithm	133.4[J]	133.1[J]	0.3[J]
Shortest route	108.7[J]	102.7[J]	6.0[J]

In Figs. 14(a) and 14(b), the dotted line represents the results of the route planning, and the solid line represents the actual route that the node moved.

Both A*algorithm and the proposed method discovered the shortest routes. However, in the proposed method the node turned only four times in reaching the destination, whereas in A*algorithm the node had to turn nine times.

Figure 15 shows the shortest route. However, it is difficult to determine the shortest route by calculation.

TABLE II shows the varying energy consumption of the node as it moved along the routes determined by A*algorithm, our method, and the shortest route extracted by humans. As shown in Figs. 14(a) and 14(b), there is little difference between the commanded route and the actual route that the node moves considering moving error. However, the rate of the differences in costs between the simulation and the experiment is only about 0.5% in our method, and only about 0.2% in A*algorithm. This means that the proposed method can be implemented correctly in an actual environment.

V. Considerations

A. Ground costs

In this research, obstacles that nodes cannot enter and geographical features are collectively called the ground cost, and the ground cost that the node incurs to pass over the cell is represented by a constant number. If we use many types of nodes, the ground cost is different according to the type. For example, a caterpillar-type node can move over rough ground with little ground cost compared to other types of nodes. Therefore, the value of the ground cost will change for different types of nodes.

The current method simply adds the ground cost to the base cost. In actuality, if the node passes over rough ground, the node not only incurs extra costs, but its direction may also be changed. Accordingly, we need to consider other approaches for calculating such kinds of errors. In addition, if the ground slopes slightly, we must set different ground costs for each direction.

B. Measuring base cost

In this paper, we measured the base cost manually. Therefore, we need a more convenient way; for example, programming nodes to measure their own base cost automatically.

In mobile sensor networks, many types of nodes are used. Since the optimum size of cells is determined from the size and shape of the nodes and wheel position, we need to propose a cost map using variable cell size.

C. Size of base cost

Since the proposed method needs to measure base cost with an actual node in advance, the larger the size of base cost we use, the more operation of the node we need. Moreover, the cost of calculating the $Score$ becomes larger with a larger base cost. On the other hand, although generally a wider base cost contributes to acquiring a better route, there is a case in which the acquired route does not change with different sizes of base costs, as is the case with using a caterpillar-type node. Therefore, we need to consider the optimum base cost size given the characteristics of a node.

D. Moving error of actual nodes

From our experiments, since the actual node did not move over rough ground, there is little difference between the routes that our method calculated and on which the node actually moved. Normally, the more times a node turns, the more moving errors accumulate. Therefore, it is significant that our method can determine a route that requires a node to turn fewer times than that of A*algorithm.

E. Related work

There are numerous and wide-ranging works in the field of mobile sensor networks. For example, RAMOS[4] has aimed to achieve cooperative routing for sensor nodes, changing some modes of the nodes according to the situation. Tilak[6] has studied localization for mobile sensor nodes. In the Wang's study[5], mobile sensor nodes move to enlarge total sensing coverage. However, these works do not address the problems in practical use that we have considered.

In Parasitic Mobility[2], nodes move by being parasitic for people or animals. However, it was proposed for an environment where there are many people or animals. One of the merits of mobile sensor networks is that nodes can enter dangerous places that people and animals cannot. However, this feature is not currently being exploited.

VI. Conclusion

In this paper, we proposed a route planning method for mobile sensor nodes by using cost map to solve several practical problems. The proposed method determines a route to a required sensing area with the lowest power consumption considering a sensor node's characteristics and obstacle arrangement. We also verified the effectiveness of the proposed method by comparing it with A*algorithm.

Our future work includes collaborative sensing among several nodes taking into consideration the kinds of sensors, sensing timing, and collisions of nodes. Moreover, we plan to propose a query language for mobile sensor networks.

Acknowledgment

This research was supported in part by " The 21st Century Center of Excellence Program " and Grantin-Aid for Scientific Research (A) (17200006) of the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

[1] D. M. Bourg and G. Seemann, "AI for Game Developers," Oreilly, Jul. 2004.
[2] M. Laibowitz and J. A. Paradiso, "Parasitic mobility for pervasive sensor networks," Proc. of Third International Conf. on Pervasive Computing (PERVASIVE 2005), pp. 255–278, Munich, May 2005.
[3] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks," Proc. of the IEEE International Conf. on Robotics and Automation (ICRA 2002), pp. 1143–1148, Washington, May 2002.
[4] R. Suzuki, K. Makimura, H. Saito, and Y. Tobe, "Prototype of a sensor network with moving nodes," Proc. of 1st International Workshop on Networked Sensing Systems, Tokyo, Jun. 2004.
[5] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," Proc. of the Conf. on Computer and Communications Societies (INFOCOM 2004), Vol. 23, No. 1, pp. 2469–2479, Hong Kong, Mar. 2004.
[6] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K. D. Kang, "Localization for mobile sensor networks," Proc. of International Conf. on Mobile Computing and Networking, pp. 45–57, Philadelphia, Sep. 2004.
[7] Crossbow, http://www.xbow.jp/.
[8] MindStorms, http://mindstorms.lego.com/.