# A Route Planning Method for Multiple Mobile Sensor Nodes

Masaki Nakamiya†, Tsutomu Terada‡ and Shojiro Nishio†
†Graduate School of Information Science and Technology, Osaka University Yamadaoka 1-5, Suita-shi, Osaka 565-0871, Japan Email: {nakamiya.masaki,nishio }@ist.osaka-u.ac.jp
‡Graduate School of Science and Technology, Kobe University Rokkodai-cho 1-1, Nada-ku, Kobe-shi, Hyogo 657-8501, Japan Email: tsutomu@eedept.kobe-u.ac.jp

***Abstract*— We have proposed a route planning method for a mobile sensor node using *cost map*. The proposed method achieves novel path planning that can solve several practical problems in conventional work: limitations of sensing areas, barricades on nodes' paths, and restrictions on nodes' movements. In this research, we propose a route planning method that can avoid collisions. Moreover, we also propose a query language to specify required sensing data. The proposed method effectively allocates multiple sensor nodes, and we show its effectiveness through simulation study.**

***Keywords*— Sensor Networks, Mobile Sensor Nodes, Route Planning, Energy Consumption, Query Language**

## I. Introduction

Currently, sensor networks using mobile sensor nodes that can migrate freely with actuators are receiving a lot of attention. We have proposed a route planning method for mobile sensor nodes using *cost map* [1]. The proposed method achieves novel path planning that can solve several practical problems in conventional work: limitations of sensing areas, barricades on nodes' paths, and restrictions on nodes' movements. However, our previous method can be applied only to one node and cannot avoid collisions from multiple nodes' movements. In this research, we propose a route planning method that can avoid collisions among multiple nodes. By using our method, we can concurrently allocate multiple nodes to multiple sensing areas and decrease electrical power consumption. Moreover, in the previous method, we need to determine the sensing area for each node considering the kinds of sensors mounted on the node and the time to get the data by the node. In order to solve this problem, we also propose a query language for mobile sensor networks. It enables us to specify sensing conditions easily and in detail: place, time, and sensor type. In this research, we show the advantage over other algorithms by comparing the electrical power consumption of nodes. The remainder of this paper is organized as follows. Section 2 describes the environmental assumptions, Section 3 explains our method in detail, and Section 4 evaluates its performance. Finally, we introduce some related works and conclude the paper in Section 5.

## II. Assumptions

In this study, we assume a static environment where the ground condition does not change dynamically. The ground 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. It is important to decrease the electrical power consumption with the nodes moving as much as possible. Therefore, in this research, we define the power consumption as the *cost* and aim at the reduction of cost. We assume various kinds of nodes, such as a car-type node, a tank-type node, and a node with two legs. In addition, in order to satisfy users' various requests, we assume mobile sensor networks using various types of mobile nodes mounted with various kinds of sensors. Moreover, the nodes keep sensing continuously and do not move after reaching the sensing areas. One of the assumed applications is in disaster relief in which buildings have been damaged. We make a rough field map of the accident site from range sensor data. From a field map, the user determines the sensing area where a human may be stranded, a gas leak or fire may happen, and so on. Then, multiple mobile nodes migrate to the sensing area in order to follow the user's sensing requests.

## III. Proposed methods

On requesting sensor data, users set multiple conditions: range, sensor types, time, accuracy, and so on. In order to adjust to the user's purpose, the system needs the flexibility to define multiple sensing conditions. Assuming that we use various types of nodes with various types of sensors for sensing multiple areas, there are a lot of allocation patterns. When a user specifies sensing requests by simple query language, our method automatically allocates multiple nodes to target areas and constructs routes for all nodes considering energy consumption and collisions among these nodes.

### A. Query language

We propose a simple query language for mobile sensor networks. Users set the conditions of sensing area, sensor type, time limits, and required accuracy. Sensing range is specified by a rectangle. Nodes have to arrive at the sensing area within the time limit. Accuracy can be set at two levels: low or high. In high level accuracy, a node must be allocated at the place where the mounted sensor covers the entire sensing

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

Fig. 1. Base cost in which direction of node is not considered

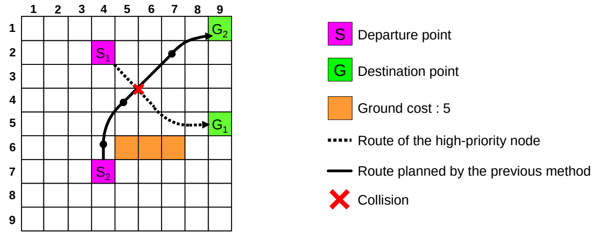



Fig. 2. Example of field map

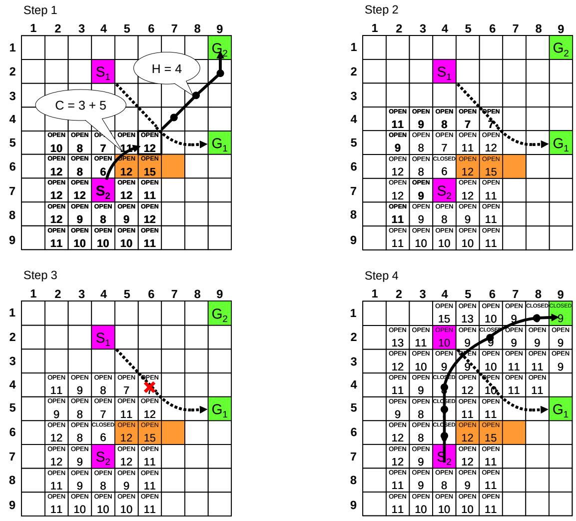


Fig. 3. Example of route planning

area. On the other hand, in low level accuracy, a node is allocated at the place where only part of the sensing area is covered. A query sentence is as follows:

SELECT $type$, $accuracy$
FROM $(x,\ y)$, $(width,\ height)$
LIMIT $time$

.

$type$ and $accuracy$ represent type of sensor and accuracy level of sensor data, respectively. $(x,\ y)$ represents the coordinate of the left-upper cell of the sensing area, $width$ and $height$ represent the number of cells of the sensing area in one row and column, and $time$ represents the time limit until acquiring sensing data.

### B. Route planning method

Our method determines the best allocation pattern between nodes and sensing areas considering collisions and energy consumption, based on a request written in the proposed query language. An overview of the proposed method is as follows. First, the system lists all the allocation patterns between nodes and sensing areas considering the sameness of the sensor types on the nodes and the required sensing areas. For each sensing area and each node, the system discovers the route that has the lowest power consumption using our proposed method, which can avoid collisions between multiple nodes. Finally, the system decides on one allocation pattern between nodes and sensing areas, whose total power consumption is the lowest, and lets the nodes move.

#### Route planning method to avoid collisions among multiple nodes

The proposed method in this paper is based on our previous route planning method using a cost map. The previous method measures the *base cost* (Fig. 1) that represents the actual power consumption of a node for moving to surrounding cells without ground influence with an actual node in advance. In order to consider the difference in the power consumption in the node's direction after it moves, the costs are measured in each of eight directions for each cell. Figure 1 represents an example of a $5\times5$ base cost of a car-type node (the direction is abbreviated for simplicity). The previous method determines a route with the lowest power consumption by constructing a cost map that shows the $Score$ of each cell, and calculates it for each cell using the following equation:

$$Score = C + H$$

. $C$ shows the total cost required for the node to move from a departure point to the cell, and $H$ the cost required for the node to move from the cell to the destination without ground cost. $Score$ is measured for eight directions of each cell. In each calculation step, the proposed method uses the two base costs properly according to the beginning directions of the node on the cell. 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), and CLOSED (Cells for which the shortest route to the cell has already been calculated).

In order to detect whether collisions will occur or not, the proposed method calculates not only the $Score$ but also the $Time$ of each cell, which represents the time elapsed while the node moves from the departure point to the cell. In our research, the situation in which a node approaches another node within the same cell is defined as another collision. To avoid collisions, our method chooses one of two solutions considering power consumption: exploring another route and making the node stop at the adjacent cell.

The proposed method discovers the route without collisions between multiple nodes using a field map (Fig. 2) and base cost (Fig. 1). Figure 3 shows an example of another route planning. In the figure, the direction of the node is not considered for the sake of simplicity. In this example, the route of the first-priority node is already calculated ($S_1$ to $G_1$), and the other node tries to avoid the collision with the first-priority node. **Step 1**: The proposed method overlays the departure point of the cost map with the center of the base cost, calculates the $Scores$ and $Times$ 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 initially. For example, on cell (6, 5),

$C$ is 8 unit costs, which is the sum of 5 unit costs by base cost A and 3 unit costs by the ground cost, and $H$ is 4 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. Then, it overlays cell (4, 6) with the center of the base cost, calculates the $Scores$ and $Times$ in the overlaid area, and sets the states of the cells in the overlaid area as OPEN, and records $Times$ and the selected cell (4, 6) as the last cell. Since the node faces in the upper direction at cell (4, 6), our method uses base cost A. On cells (2, 5), (2, 8), and (3, 7), the new $Scores$ are lower than the previous $Scores$ that have already been calculated. In this case, $Scores$, $Times$ and information of the last cell are updated.

**Step 3**: On repeating the previous steps, when the proposed method selects the cell where the upper-priority nodes also pass, such as cell (6, 4), it compares the recorded $Time$ for the upper-priority nodes with the $Time$ of the focused node at cell (6, 4). We assume nodes' collisions when the span between the recorded $Times$ and the $Time$ of the focused node is shorter than the time length the node takes to move through two cells. If the proposed method detects the collision, it searches for another route or lets the node stop at the adjacent cell. In the latter, nodes take on extra costs to stop, continue staying, and restart. Therefore, $Score$ and $Time$ on cell (6, 4) are updated.

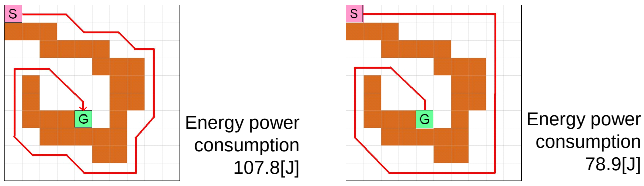
**Step 4**: This procedure is repeated until the destination cell is selected and its state is set to CLOSED. Since cell stores the coordinate of the last passed cell, our method can trace the route without collisions. In this example, although the discovered route intersects the route of the upper-priority node, it avoids collisions because of the time lag.

## IV. Performance Evaluation

First, we developed several prototypes of sensor nodes with MindStorm (LEGO company), and verified the proposed method in actual environments. Then, we measured the base costs in 3×3 cells with the prototypes. Both the width and height of each cell in the field map are 20cm, which is suited to the actual node's size. To evaluate the proposed method in detail, we implemented a simulator study. 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 tank-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.

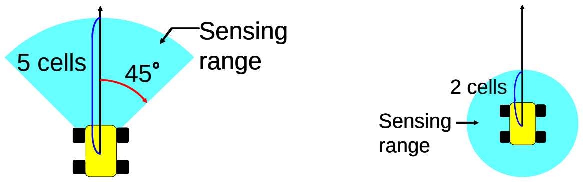
### A. Previous route planning methods

We have proposed a route planning method for mobile sensor nodes using a cost map, and compared it with the A* algorithm, which is widely used for route planning in the field of artificial intelligence. Figure 4 shows an example of route planning results using the A*algorithm (Fig. 4(a)) and the proposed method using a tank-type node (Fig. 4(b)). Although the A*algorithm can determine one of the shortest routes, this may include extra rotation of the node. In Fig. 4(b), the node rotates fewer times than in the case of the A*algorithm. This means that the proposed method considers the turning cost by using the base cost. Furthermore, the proposed method can achieve the reduction of the energy power consumption from the A*algorithm.




(a) A*algorithm (b) Tank-type node

Fig. 4. Results of route planning
(S: departure point, G: destination point)




(a) Infrared sensor (b) Thermometers and hygrometers

Fig. 5. Sensing model

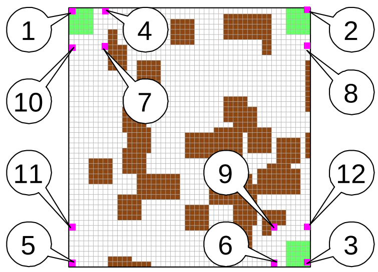



Fig. 6. Examples of geographical features

### B. Performance evaluation of the proposed method

We evaluated our method by comparing it with three methods; one avoids collisions by delaying the departure times of the lower-priority nodes, another avoids collisions by stopping at the adjacent cell, and the other searches for the route that has the second lowest power consumption without any collision. Note that stopping nodes consumes half of the power for moving, which is measured on actual prototype nodes. In the evaluation, we assume two types of nodes: tank-type and car-type nodes, and three kinds of sensors: thermometer, hygrometer and infrared sensor. To consider these sensor's characteristics, we define the sensing model of these sensors as shown in Fig. 5.

**Energy and time consumption in different sizes of field maps**

We evaluated the sum of energy consumption of each node and the time consumption for our method and three comparative methods. Time consumption designates the time for all nodes to reach the sensing areas. We use square field maps with geographical features arranged at random as shown in Fig. 6. The size of the map is changed from 60 cells to 100 cells in 10-cell increments. We use 6 tank-type

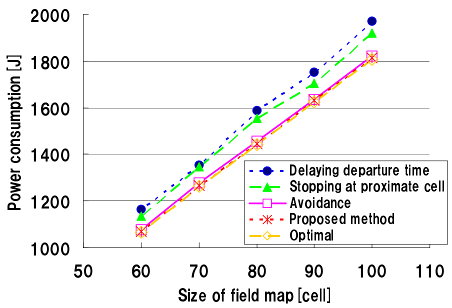



Fig. 7. Power consumption vs. size of field maps (arranged at random)

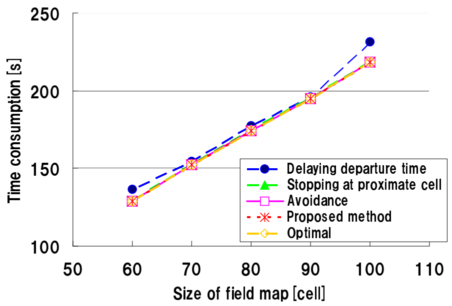



Fig. 8. Time consumption vs. size of field maps (arranged at random)

nodes and 6 car-type nodes, mounted with one or two types of sensor (no node has the same type mounted on it) and set each node's departure position near the corners of the field map. The numbers represent the nodes' numbers. In the simulation, the nodes, which have a smaller node number, have higher priority. There are three sensing areas with 4 cell widths: the upper-left corner (Infrared sensor), upper-right corner (Hygrometer) and lower-right corner (Thermometer) as shown in Fig. 6(b). The example of the query is shown as follows:

SELECT $temp$, 1
FROM (96, 96), (4, 4)

.

Figure 7 shows the simulation result of the power consumption. In these figures, "optimal" means the sum of route planning by our previous method virtually in the environment that results when multiple nodes stay at the same cell. In other words, the optimal method cannot be used in actual environments. Although, the result shows that the proposed method determines routes with more costs than the optimal method, it discovers routes with lower costs than the three comparative methods in all conditions. For the sake of avoiding to collisions, one of the comparative methods delays the departure times of nodes to move to the sensing area without any stops, another method stops the lower-priority nodes at the proximate cell and lets the nodes restart after a brief stay. In these two processes, the nodes need extra power consumption to stop, keep staying, and restart. On the other hand, the proposed method searches for another route or lets the node stop at the proximate cell to avoid collision, and selects appropriate avoidance that takes lower power consumption. Compared with the comparative method that only searches for another route to avoid collisions, our method improves power consumption somewhat. However, this result means that the proposed method mainly selects another route to avoid collisions; in some cases, avoidance by stopping at an adjacent cell demands less power than moving by another route. Figure 8 shows the simulation result of the time consumption. The result shows that, although the proposed method determines the routes with a bit less time consumption than the three comparative methods, the ratio of improvement is smaller than the result of power consumption. Time consumption means the time until all nodes finish being allocated. In other words, the time consumption is the time for the node that has the longest trip to arrive at the sensing area. Therefore, unless that node crashes into another node and stops to avoid collisions, this value does not change.

## V. Related Works and Conclusion

There are numerous and wide-ranging works in the field of mobile sensor networks. For example, RAMOS [3] has aimed to achieve cooperative routing for sensor nodes, changing some modes of the nodes according to the situation. In Wang's study [4], mobile sensor nodes move to enlarge total sensing coverage. However, these works do not address the problems in practical use that we have considered. There are also many previous works, which focus on path planning for mobile robot navigation. Ramirez achieves a local path planning for nonholonomic mobile robots in an environment with obstacles by using a feasible velocities polygon [2], and Wuwei proposes a robot navigation scheme based on searching points on an arc [5]. Although these works need both the mathematic model of moving and the path planning algorithm for each type of node, our proposed method can be utilized for any type of node by only measuring base cost.

In this research, we propose a route planning method that can avoid collisions among multiple nodes and a query language that is adapted to mobile sensor networks. We verified the effectiveness of the proposed method by comparing it with two comparative algorithms. Our future work includes considering data transfer by migrating to the data center or using multi-hop communication.


## Acknowledgment

This research was supported in part by Grant-in-Aid for Scientific Research (A) (17200006) of the Japanese Ministry of Education, Culture, Sports, Science and Technology.


## References


[1] M. Nakamiya, et al.: A route planning method using cost map for mobile sensor nodes, *Int'l Symposium on Wireless Pervasive Computing (ISWPC 2007)* (Feb. 2007).

[2] G. Ramirez and S. Zeghloul: A new local path planner for nonholonomic mobile robot navigation in cluttered environments, in *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA 2000)*, Vol. 3, pp. 2058–2063 (Apr. 2000).

[3] R. Suzuki, et al.: Prototype of a sensor network with moving nodes, *Special Issue on Int'l Workshop on Networked Sensing Sytem 2004 Transaction of the Society of Instrument and Control Engineers*, Vol. E-S-1, No. 1, pp. 52–57 (2006).

[4] G. Wang, et al.: Movement-assisted sensor deployment, in *IEEE Trans. on Mobile Computing*, Vol. 5, No. 6, pp. 640–652 (June 2006).

[5] C. Wuwei, et al.: A new navigation method for an automatic guided vehicle, *Journal of Robotic Systems*, Vol. 21, No. 3, pp. 129–139 (2004).