Pervasive and Mobile Computing 6 (2010) 623–641

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Map estimation using GPS-equipped mobile wireless nodes

Shinichi Minamimoto, Sae Fujii *, Hirozumi Yamaguchi, Teruo Higashino

Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

ARTICLE INFO

Article history:
Received 10 April 2010
Accepted 3 June 2010
Available online 10 June 2010

Keywords:
Mobile wireless nodes
GPS
Ad hoc wireless communication
Map estimation

ABSTRACT

Large accidents and disasters in crowded regions such as business districts and universities may create a large number of patients, and first responders need to recognize the presence and location of buildings for their efficient rescue operations. In this paper, we propose an algorithm to estimate the two-dimensional (2D) shapes and positions of buildings, simultaneously using GPS logs and wireless communication logs of mobile nodes. The algorithm is easy to implement since it only needs general wireless devices such as smartphones. The results from the experiments conducted assuming rescue operation scenarios have shown that the proposed method could quickly generate a map with 85% accuracy.

1. Introduction

Situation awareness is the basis of ubiquitous society. We try to sense or capture physical phenomena such as change of temperature and rain, or try to recognize and analyze the forms, locations and behavior of the real world's objects (such as vehicles and pedestrians) and landscape. We have learned that such situation awareness is also very significant for rescue operations in cases when many people are injured suddenly by a large accident or a disaster in small and condensed regions. For example, in Japan, we experienced a very tragic train accident in 2005 in which over 100 people died and about 460 people were injured. It has been reported that rescue teams need to recognize the positions and condition of patients for efficient rescue operations in such a situation [1]. We have started to design and develop an electronic "triage" system. It continuously senses the vital signs of the patients and estimates their locations by IEEE802.15.4-based wireless sensor networks. We are leading this national project, involving five organizations with several medical doctors and professors in an emergency care department [2].

These doctors say that fast recognition of obstacles such as buildings in the region will be very helpful for rescue operations and treatment actions. It is desirable to generate a *local map* of the site, which tells us building and street structure information in a city section, the presence of warehouses in a factory, or complicatedly connected small buildings on a university campus. However, such a local and thus detailed map cannot often be obtained from a public map, especially if the region is private property, and even pathways (or streets) may be changed after a disaster. Using digital images of the landscape or range information from radar sensors is a possibility to build a map, but dedicated effort (i.e., taking pictures or measuring ranges at specific points toward specific directions) to obtain such information is required. This encumbers efficient rescue operations since doctors and rescue teams always need manpower for treatment actions. Thus *automated* acquisition of a local map *without dedicated hardware* is mandatory in such emergency situation.

In this paper, we propose a local map estimation algorithm for the recognition of an accident site in an emergency situation. We assume that each member in the rescue teams, called a *mobile node*, is equipped with a GPS receiver and a mid-range communication device such as IEEE802.11 or IEEE802.15.4 that can directly communicate with others several

* Corresponding author. Tel.: +81 6 6879 4557; fax: +81 6 6879 4559.
E-mail addresses: s-minmmt@ist.osaka-u.ac.jp (S. Minamimoto), s-fujii@ist.osaka-u.ac.jp (S. Fujii), h-yamagu@ist.osaka-u.ac.jp (H. Yamaguchi), higashino@ist.osaka-u.ac.jp (T. Higashino).

1574-1192/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2010.06.001

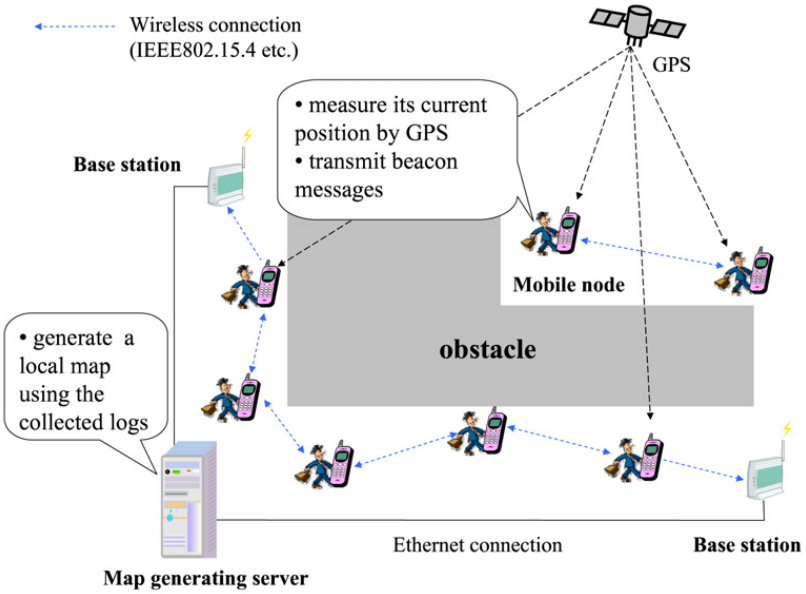

Fig. 1. Environment for proposed algorithm.

tens of meters away. Since such equipment is very general, the algorithm does not require dedicated devices. The algorithm estimates movable areas and obstacles using position information from GPS receivers and communication logs among mobile nodes. In more detail, it identifies the trajectories of mobile nodes from GPS logs in order to estimate pathways, and simultaneously estimates the presence of objects from the communication logs among mobile nodes in order to estimate obstacles. These results are finally merged to output an estimated entire map.

Our aim is to clarify the challenges in this automated generation of local maps and provide an efficient and reasonable approach. We need to take into account that GPS errors and uncertainty of radio propagation with presence of obstacles may have a negative effect on the map accuracy. To cope with this problem, we conducted several preliminary field experiments. Based on the results, we take an approach to using probabilities and counters to determine whether each subregion is occupied by an obstacle or not. After estimating the rough form of the obstacles, image-processing techniques are applied to improve the readability of the map. Two field experiments and several simulation experiments were conducted to validate the effectiveness of the algorithm. In particular, in the field experiments, we estimated the map of a 150 m $\times$ 190 m region on our university campus and that of a 225 m $\times$ 250 m region with many apartment buildings. The results from those experiments have shown that maps with about 85% accuracy were generated within a few hundred seconds.

Compared with our preliminary work that was presented in [3], this paper has considerable extensions along with new experimental results. They are summarized as follows. (i) We have conducted additional simulations and field experiments. In particular, a new field experiment was conducted in a region with many buildings. From the experimental results, we have confirmed that our method could recognize all the buildings and pathways with sufficient accuracy. These results are presented in Sections 4 and 5. (ii) We have designed several extensions to the basic algorithm to enhance its capability. First, we have implemented a function to enable combined use of existing and generated maps that facilitates situation recognition. Second, we have addressed our ideas to deal with plausible and realistic situations that had not been considered in the basic algorithm. The details of the extensions are explained in Section 6. (iii) In order to motivate our work, we introduce a known map estimation technique called SLAM that simultaneously estimates the location and map of mobile robots. Then we explain the difficulty of applying SLAM to our case. This is introduced in Section 7.

The rest of this paper is organized as follows. Section 2 outlines the problem and design of the proposed algorithm, and Section 3 gives the algorithm description. Section 4 explains the simulation results, which are followed by the results from the two field experiments in Section 5. In Section 6, we give discussions on the design of possible algorithm extensions. Section 7 summarizes the related work and addresses the contribution of this paper. Finally we conclude this paper in Section 8.

2. Problem statement and algorithm design

2.1. Problem statement

In Fig. 1, we exemplify the environment in which our proposed algorithm works. A targeted region consists of movable space such as pathways, and obstacles such as buildings. We assume that a *mobile node* (or simply a *node*) is a person who has a wireless terminal and can move only in movable space. Each node has a GPS receiver and measures its current position every T_p seconds. This position information contains some error range, which is unknown in the algorithm. It also has a

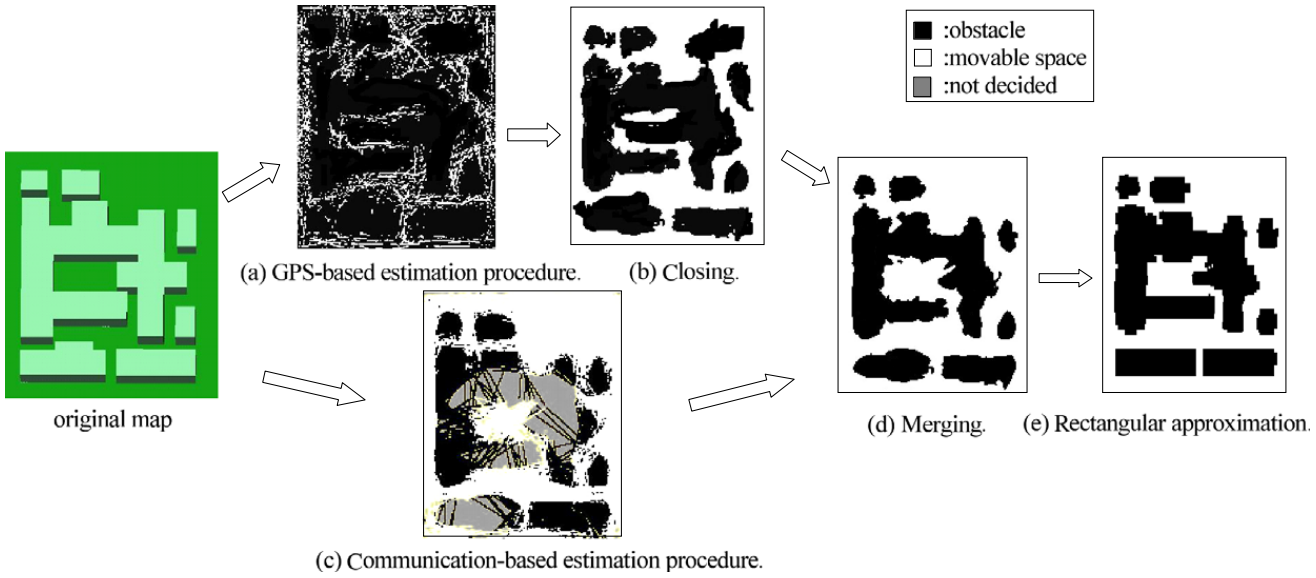

Fig. 2. Algorithm outline.

mid-range communication device such as IEEE802.11 and IEEE802.15.4. It transmits a *beacon message* that contains the measured position every T_c seconds. We assume that each node roughly knows the global time, which is easily obtained through the GPS device.

Every time node i measures its position, it records (i, p_i, t), where p_i and t are the measured position and time, respectively. This is called a *GPS log*. In addition, when node j receives a beacon message from node i that contains p_i, node j records (i, j, p_i, p_j, t), where t denotes the reception time of this message (global time) and p_j is the latest measured position of node j. This is called a *communication log*. Both the GPS logs and communication logs collected by mobile nodes are sent to a single server. We assume that the delivery of those logs is done in various ways; for example, mobile nodes can send them when they meet base stations or by multi-hop transmission over other nodes. On the server, the proposed algorithm estimates the shapes and positions of obstacles.

The problem treated in this paper is to estimate the movable space and obstacles in a targeted area as accurately as possible using all the GPS logs and communication logs.

2.2. Map generation—challenges and approaches

We provide a centralized algorithm. The outline of our algorithm is shown by the produced maps in Fig. 2. The algorithm consists of two independent map estimation procedures: (i) estimation of movable space by GPS logs (called the *GPS-based estimation procedure*; map (a) of Fig. 2) and (ii) estimation of obstacles by communication logs (called the *communication-based estimation procedure*; map (c) of Fig. 2). After applying the GPS-based estimation procedure, an image-processing technique called *closing* is applied (map (b)). Finally, the two maps (b) and (c) are merged into a single map (d), and finally refined by an original image-processing technique called *rectangular approximation* (map (e)).

In the following, we state the design consideration of the two procedures, addressing the challenges we face in the problem.

2.2.1. GPS-based estimation procedure

Mobile nodes move in movable space. Therefore, for each GPS log (i, p_i, t), position p_i is in the movable space if p_i contains no error. In addition, for two time-subsequent logs (i, p_i, t) and $(i, p_i', t + T_p)$, the estimated trajectory between p_i and p_i' is also in the movable space. Here, since the positions may contain errors, a simplistic decision may fail to precisely estimate the movable space. Our approach is a counter-based one in which for each small grid cell in the region we count how many times the cell is marked as "movable space". Since GPS errors can be considered quasi-random ones in terms of time, location and nodes, the cells in real movable space will possibly have larger counts. Therefore, this straightforward idea alleviates GPS measurement errors.

2.2.2. Communication-based estimation procedure

One plausible approach is to use the received signal strength (RSS) information. We may derive the expected RSS (denoted as $\widehat{rx}$) from a known radio propagation model assuming there is no obstacle between p_i and p_j. Then we compare the measured RSS rx with the expected RSS $\widehat{rx}$, and see how much rx deviates from $\widehat{rx}$. Based on this deviation, we may estimate the existence of obstacles between p_i and p_j. However, several factors such as multi-path signals or radio signals from other sources may interfere with radio propagation and may fluctuate the RSS values. For example, Wireless LAN interferes

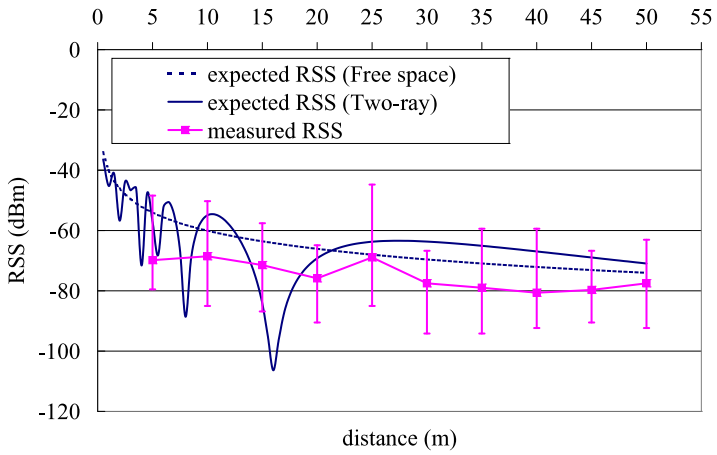

Fig. 3. Expected and measured RSS (versus distance).

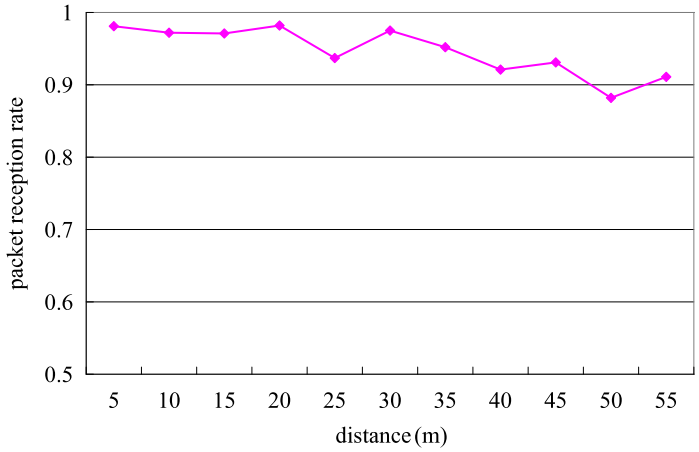

Fig. 4. Measured packet error rate (versus distance).

with 2.4 GHz radio frequency. Human bodies and humidity may also reduce the signal power. In order to observe such phenomena, we conducted a simple field experiment. We used ZigBee modules JN5139 [4] (Jennic Ltd.). The experiment was done in open space without any obstacle, and two ZigBee modules were set 1 m above the ground. One module transmitted a 26 byte packet every 1 s with transmission power 0 dBm, and in total 10 packets were transmitted for each distance. Fig. 3 shows the expected and measured RSS values versus the two nodes' distance. To derive the expected RSS, we assumed 2.4 GHz frequency, and used the two-ray ground model with $\lambda = 0.125$ m and $\gamma = -1$ and the free-space model [5]. This graph shows that the measured RSS values fluctuated even in the same distance case, and do not fit for the models even in this stationary environment. [6] has also shown that the errors of RSS-based ranging measurements are of non-Gaussian distribution in nature. Therefore we can easily conclude that they are not applicable to such a situation that we assume.

Since the RSS is too sensitive, we consider using packet reception ratios. For two nodes that have shorter distance than the expected maximum communicable range (denoted by R), we can estimate the presence of obstacles between two nodes based on the following intuitive rules: (1) if node j could receive a beacon message from node i, there is no obstacle between p_i and p_j, and (2) if the distance between p_i and p_j was less than R and node j could not receive a beacon message from node i, there is an obstacle between p_i and p_j. However, interference also affects the packet reception ratio as in the case of the RSS. For example, a packet is not delivered even if there is no obstacle between p_i and p_j and the distance between them is less than R. Another concern is radio diffraction. A packet is delivered even if there is an obstacle between p_i and p_j.

In order to see to what extent such phenomena happen, we also measured the packet reception ratio in the previous experiment of Fig. 3. The result is shown in Fig. 4. In an ideal environment, the packet reception ratio in this graph should be 100%, but actually around 10% is lost, for various reasons. Also, to see the effect of diffraction, we used Jennic JN5139 (2.4 GHz) modules as in the previous experiment. As shown in Fig. 5, we used two JN5139 modules. The sender was either located at point 1 or 2, and it transmitted 10,000 packets of 26 bytes with −18 dBm. The receiver was located at one of points 3–10, and it counted the received packets. We note that the RSS threshold of a JN5139 module was −96 dBm. The expected RSS was derived using the knife-edge diffraction model [7]. Table 1 shows the expected RSS values and packet reception ratio. From the results, in any case that the line of sight is blocked by the obstacle, the packet loss ratio was large and diffraction merely occurred. The model indicates that in 2.4 GHz RF, the expected RSS was around −95 dBm, which

Table 1
RSS calculated by knife-edge diffraction model.

Sender		Receiver							
		3	4	5	6	7	8	9	10
1	Packet reception ratio (%)	0	0	0	93.3	0	0	0	90.7
	RSS by diffraction model (dBm)	−108.7	−104.9	−99.2	−90.6	−112.7	−109.0	−103.1	−93.1
2	Packet reception ratio (%)	0	13.7	59.8	94.4	0	0	0	91.7
	RSS by diffraction model (dBm)	−103.7	−100.7	−94.7	−87.0	−108.3	−104.6	−99.0	−90.6

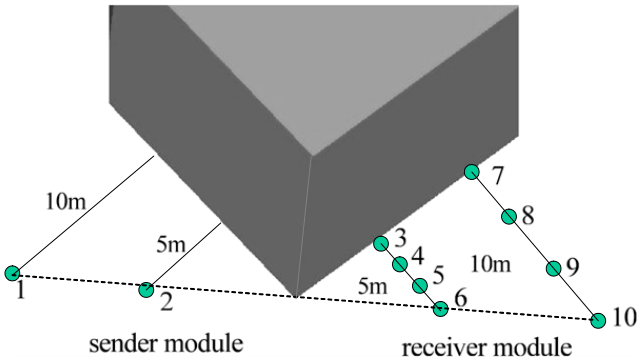

Fig. 5. Diffraction propagation.

is almost the RSS threshold. Since in most cases the measured RSS was smaller than the expected one, packet delivery by diffraction is not likely to occur. Consequently, we can take a simple approach to using the packet reception instead of RSS, but we still need to take into account that GPS errors and unexpected loss of packets may obscure the decision. For this, we introduce a probability to represent the degree of likelihood that the packet delivery is done as expected. Also, to increase the confidence, we introduce counters as in the movable space estimation procedure.

In the following section, we give the details of the algorithm.

3. Algorithm description

The algorithm divides a targeted area into $m \times n$ square cells, and estimates for each cell whether it is occupied by an obstacle or not. Hereafter, a cell occupied by an obstacle is called an *obstacle cell* and one in movable space is called a *non-obstacle cell*. A cell at row a and column b is denoted by $g_{a,b}$ ($1 \leq a \leq m$ and $1 \leq b \leq n$).

3.1. GPS-based estimation procedure

In the GPS-based estimation procedure, for each GPS log (i, p_i, t), we find the cell containing p_i. In addition, for two subsequent GPS logs (i, p_i, t) and $(i, p_i', t + T_p)$, we find the cells on the line segment between p_i and p_i'.

Here, we denote each of such cells by c. c might be likely to be a non-obstacle cell, but this should not be the final decision due to ambiguity from GPS errors. Hence, we determine that c is a non-obstacle cell only if mobile nodes transit over cell c more than h times, where h is the average number of mobile node transits over the cell. Here, we explain how to determine the value of h. We let N, T and V denote the number of mobile nodes, the time length during which logs are collected, and the average speed (m/s) of mobile nodes, respectively. Also, the area is $x \times y$ (m^2) and the side length of a cell is denoted by g (m). Since the expected time for a mobile node to transit from a cell to its neighboring cell is g/V (s), the expected number of cells all the nodes transit during time T is TVN/g. Also, since the number of cells in the targeted area is xy/g^2, the average number of mobile node transits per cell is derived by $h = gTVN/xy$.

Finally, to prevent non-obstacle cells, over which few nodes transit, from being determined as obstacle cells, we apply the *closing* process, which is known as an image-processing technique [8]. Closing is used to reveal thin-line characters and lines in figures, and consists of two steps: dilation and erosion of white pixels. In the proposed algorithm, we regard obstacle cells as black pixels, and non-obstacle cells as white pixels. In the dilation step, each black pixel is changed to a white pixel if it has more than five white pixels as its adjacent cells. In the erosion step, each white pixel is changed to a black pixel if it has fewer than four black pixels as its adjacent cells. We apply the dilation k times, and after that apply the erosion k times (empirically $k = 3$ produces good results). Figs. 6 and 7 show examples of GPS-based estimation and closing, respectively.

3.2. Communication-based estimation procedure

In the communication-based estimation procedure, for each cell $g_{a,b}$, we prepare two integer counters $T_{a,b}$ and $F_{a,b}$ initialized by zero. For each pair (i, p_i, t) and (j, p_j, t) of two GPS logs where the distance between two nodes is less than the

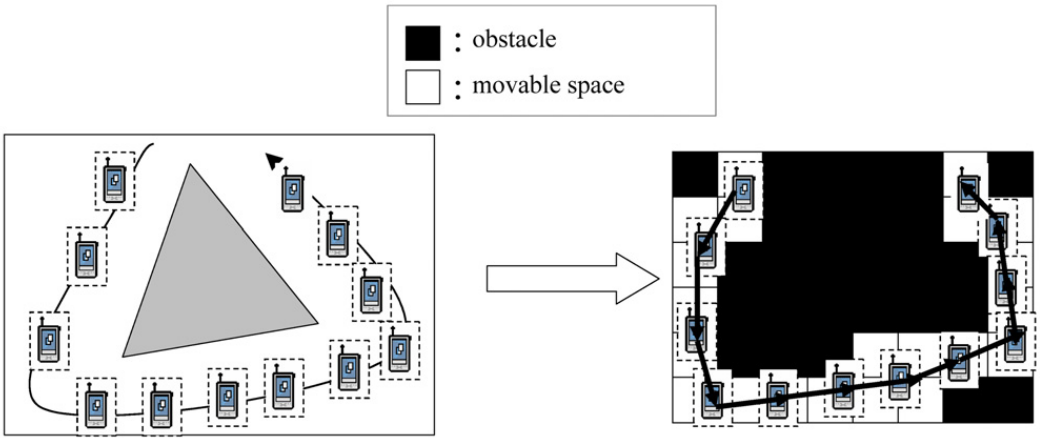

Fig. 6. GPS-based estimation procedure.

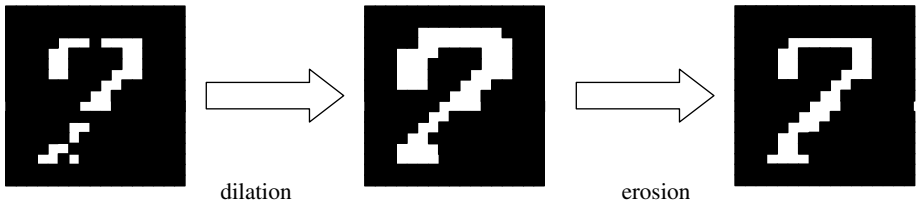

Fig. 7. Closing technique.

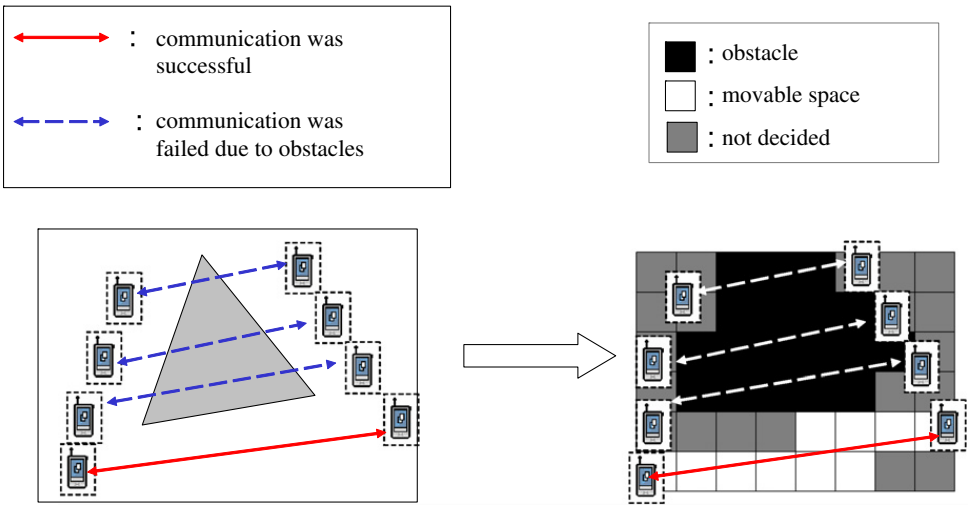

Fig. 8. Communication-based estimation procedure.

expected maximum communicable range R, we check if the corresponding communication log (i, j, p_i, p_j, t) exists or not. If it exists, then for each cell $g_{a,b}$ on the line segment $p_i - p_j$ we increase $F_{a,b}$ by one. Otherwise, we increase $T_{a,b}$ by one. Here, $T_{a,b}$ is the count judging that $g_{a,b}$ is an obstacle cell, and $F_{a,b}$ is the one judging that it is a non-obstacle cell. Fig. 8 shows an example of the communication-based estimation procedure.

Based on $T_{a,b}$ and $F_{a,b}$, we determine if $g_{a,b}$ is an obstacle cell or not. Here, there is a possibility that $T_{a,b}$ is increased but $g_{a,b}$ is actually a non-obstacle cell. In contrast, there is also a possibility that $F_{a,b}$ is increased but $g_{a,b}$ is actually an obstacle cell. Because GPS positions include errors and radio propagation is uncertain, such incorrect decisions may happen. Therefore, we cannot rely only on those counters.

To cope with such ambiguity, the proposed algorithm calculates the score to determine whether or not $g_{a,b}$ is an obstacle cell based on Bayesian estimation. Bayesian estimation is a method to estimate the event of a hypothesis from a given observed event. Here we define A as the event that two nodes communicate with each other over a cell, and B as the event that the cell is actually an obstacle cell. Also, $P(A)$ and $P(B)$ are the probabilities of events A and B, respectively. Therefore, $P(B|A)$ is the posterior probability that the cell is an obstacle cell after we know that two nodes communicate over it. $P(B|A)$

is given by formula (1) according to the Bayesian theorem.

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}. \tag{1}$$

We may assign 0.5 to the prior probability $P(A)$ because we cannot initially know whether or not two nodes communicate over the cell. If we do so, formula (1) is reduced to formula (2).

$$P(B|A) = 2P(A|B)P(B). \tag{2}$$

From formula (2), the probability that a cell is an obstacle cell is the prior probability multiplied by $2P(A|B)$ when two nodes are regarded as communicable over the cell.

In a similar way, we represent $P(B|\bar{A})$, the posterior probability that a cell is an obstacle when nodes are regarded non-communicable over the cell, by formula (3). Also, if we assign 0.5 to the prior probability $P(A)$, the probability that a cell is an obstacle cell is the prior probability multiplied by $2P(\bar{A}|B)$ when two nodes are regarded as non-communicable over the cell.

$$P(B|\bar{A}) = \frac{P(\bar{A}|B)P(B)}{P(\bar{A})} \tag{3}$$

$$P(B|\bar{A}) = 2P(\bar{A}|B)P(B). \tag{4}$$

By the fact that nodes communicate over a cell, the probability that the cell is an obstacle cell is increased by $2P(A|B)$. Similarly, by the fact that nodes do not communicate over a cell, the probability that the cell is a non-obstacle cell is increased by $2P(\bar{A}|B)$. Based on this idea, we define the score $p_{a,b}$ given by formula (5) to determine whether cell $g_{a,b}$ is an obstacle cell or not, using $T_{a,b}$ and $F_{a,b}$ assuming that the initial value of $P(B)$ is 0.5. If $p_{a,b}$ is greater than a certain threshold, we determine that $g_{a,b}$ is an obstacle cell (empirically, 0.8 is appropriate for the threshold).

$$p_{a,b} = \frac{1}{2} \times (2P(A|B))^{T_{a,b}} \times (2P(\bar{A}|B))^{F_{a,b}}. \tag{5}$$

In the experiments described in the following sections, we assign 0.1 to $P(A|B)$ from the preliminary experiment in Section 2.2 because we can see that the probability that two nodes communicate by diffraction is low when there is an obstacle between them. Also, we assign 0.9 to $P(\bar{A}|B)$ based on the results of the packet reception rate in Section 2.2. When $4.5 \times T_{a,b} \leq F_{a,b}$ holds, it is determined that $g_{a,b}$ is an obstacle cell. We note that if both $T_{a,b}$ and $F_{a,b}$ are zero, we do not make a decision for $g_{a,b}$.

3.3. Merging maps and refinement

Finally, we obtain a single map by merging two maps from the above two procedures. The decision for $g_{a,b}$ is done as follows.

- If $g_{a,b}$ is determined as an obstacle cell in both procedures, $g_{a,b}$ is determined as an obstacle cell.
- If $g_{a,b}$ is determined as a non-obstacle cell in both procedures, $g_{a,b}$ is determined as a non-obstacle cell.
- If $g_{a,b}$ is determined as a non-obstacle cell by either one of the two procedures, $g_{a,b}$ is determined as a non-obstacle cell. One reason for this rule is that, in the communication-based estimation procedure, the "non-obstacle" decision is more credible than the "obstacle" decision because all the cells between two nodes that cannot communicate with each other are determined as obstacle cells even though most of them are actually non-obstacle ones. Another reason is that, in the GPS-based estimation procedure, cells over which nodes do not transit are determined as non-obstacle cells even though they are actually obstacle cells.
- If a decision is not made for $g_{a,b}$ in the communication-based estimation procedure, we rely on the decision by the GPS-based estimation procedure.

The obtained map is likely to be distorted, as shown in Fig. 2(d). However, the general forms of buildings are close to polygons. In such cases, we may apply the final refinement procedure called *rectangular approximation*. The procedure recognizes a (small) set of cells that constitute a single obstacle by comparing the decision of each cell with those of neighboring cells, and approximates its boundaries by lines. By applying rectangular approximation, we can derive more readable maps, as shown in Fig. 2(e).

4. Simulation experiments

We evaluated the performance of the proposed algorithm by simulations using the QualNet simulator [9] and Wireless InSite module [10] that can accurately simulate radio propagation based on several models. In order to test the performance in such situations that radio propagation is interrupted by obstacles such as buildings and the mobility of nodes is restricted, we used a map shown in Fig. 9 that models a 150 m × 190 m region on our university campus (a picture of the region is shown in Fig. 10). We assume that each mobile node moves along a pathway and randomly chooses a new direction except backward at each intersection, and the speed follows a normal distribution with mean 1.5 m/s and variance 0.01. In order to simulate the radio propagation accurately, we used the radio propagation model provided by Wireless InSite, assuming

Table 2
Simulation settings.

Maximum radio range (R_{max})	25, **50**, 75 (m)
Number of mobile nodes	**15**, 30, 45
Beacon message transmission frequency (T_c)	**1.0**, 5.0, 10.0 (s)
GPS positioning frequency (T_p)	**1.0**, 5.0, 10.0 (s)
Average position error (μ)	0, **5.0**, 10.0 (m)

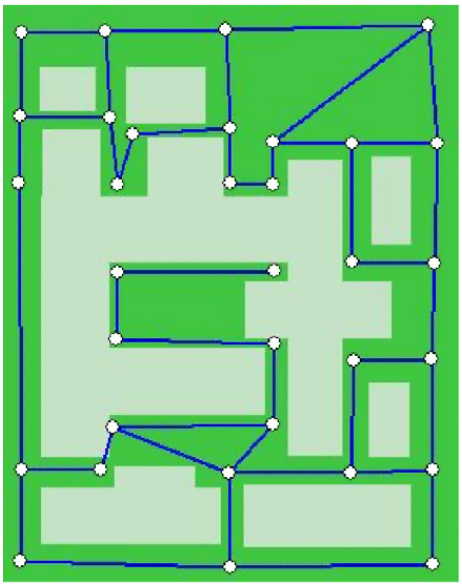

Fig. 9. Target map in simulation.

Fig. 10. Air photograph of target map.

2.4 GHz RF. Also, we set its transmission power to such a value that makes the maximum radio range be R_{max} according to the two-ray ground model [5]. We note that in this simulator with the radio propagation model, interference affected by multi-path signals is simulated. Moreover, we assume that the GPS errors follow a normal distribution with mean μ and variance 1. The size of a cell g was set to 1 m, and the expected maximum communicable range of our algorithm R was equal to R_{max}. The other parameter settings are shown in Table 2 (the default values are emphasized by bold font).

With the above settings, we generated a local map using the GPS logs and communication logs during a period of 600 s, and evaluated the ratio of cells estimated correctly to the entire cells. This ratio is denoted by *Hit* and is defined as follows:

$$Hit = \frac{1}{mn}\sum_{a=1}^{m}\sum_{b=1}^{n} hit(g_{a,b}),$$

(a) $R_{max} = 25$. (b) $R_{max} = 50$. (c) $R_{max} = 75$.

Fig. 11. Generated maps under different R_{max} values.

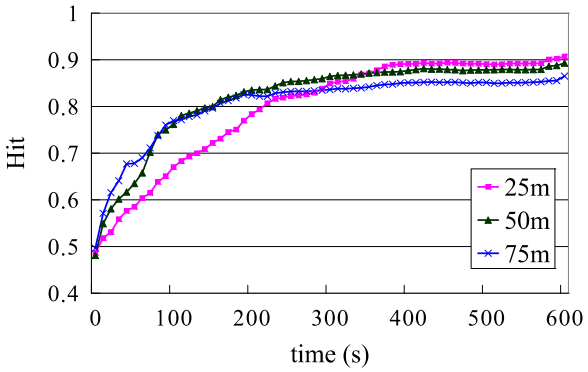

Fig. 12. Impact of R_{max} on *Hit*.

where m is the number of cells in the row and n is the number of cells in the column. $hit(g_{a,b})$ returns 1 if $g_{a,b}$ is estimated correctly, and returns 0 otherwise.

4.1. Impact of parameters on estimation accuracy

We evaluated the impact of several factors on the estimation accuracy. We varied one of the parameters given in Table 2, and set the others to the default values.

4.1.1. Maximum radio range

We observed *Hit* under different transmission powers such that R_{max} in an ideal environment was 25 m, 50 m or 75 m. Fig. 11 shows the generated maps, and Fig. 12 shows *Hit*. We can see that the estimation accuracy is better as the range is shorter. This is because our algorithm regards all the cells over the line segment between two nodes that are closer than R as obstacle cells if they cannot communicate with each other. The number of incorrect cells with short radio range is smaller than that with long radio range.

4.1.2. Number of mobile nodes

We varied the number of nodes. Fig. 13 shows the result when the number of nodes was set to 15, 30 or 45. We can see that, with the larger number of nodes, *Hit* is larger and it converges quickly. This is simply because we can get more GPS logs and communication logs.

4.1.3. Beacon message interval

We varied the beacon message interval, which is denoted by T_c. It was set to 1, 5 or 10 s. The result is shown in Fig. 14. We can see that a longer interval results in a smaller *Hit* value due to fewer communication logs. However, *Hit* is still larger than 0.8 with the longest interval $T_c = 10$.

4.1.4. Average position error

GPS errors are expected to greatly affect both GPS-based and communication-based estimation procedures since these procedures rely on position information from the GPS system. We set the average position error to 0 m, 5 m or 10 m, and

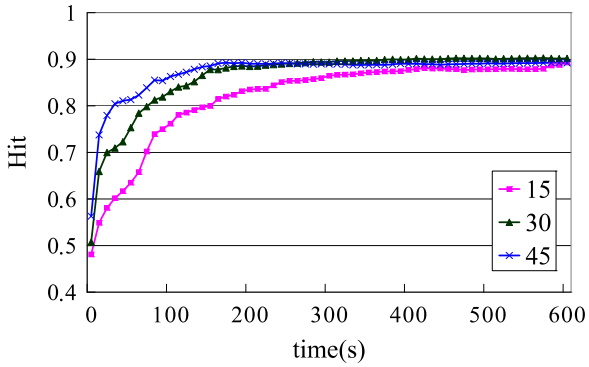

Fig. 13. Impact of the number of nodes on *Hit*.

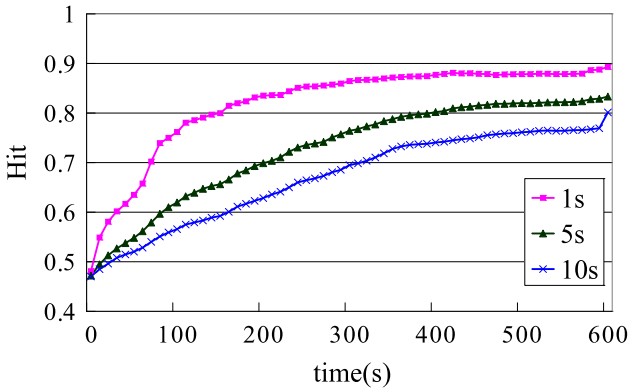

Fig. 14. Impact of beacon message frequency T_c.

(a) $\mu = 0$ m. (b) $\mu = 10$ m.

Fig. 15. Generated maps under different position errors.

measured *Hit*. Fig. 15 shows the generated maps, and Fig. 16 shows the corresponding values of *Hit*. It is natural that the value of *Hit* under a smaller position error is better. However, even in the case of 10 m, we can obtain a readable map, and this can further be improved by additionally applying image-processing techniques (e.g., closing).

4.1.5. GPS positioning frequency

We also evaluated *Hit* on varying the GPS positioning frequency, denoted by T_p. It was set to 1, 5 or 10 s. From Fig. 17, the GPS positioning frequency has little effect on the estimation accuracy. This is because we adopt a linear interpolation technique between subsequent positions, and trajectories can be derived with low frequency.

4.2. Individual evaluation of the two estimation procedures

In this section, we evaluate the GPS-based and communication-based estimation procedures individually. We compared the three maps obtained by the GPS-based estimation procedure, by the communication-based estimation procedure, and by both procedures (the final product of the proposed algorithm). We set the parameters to the default values in Table 2.

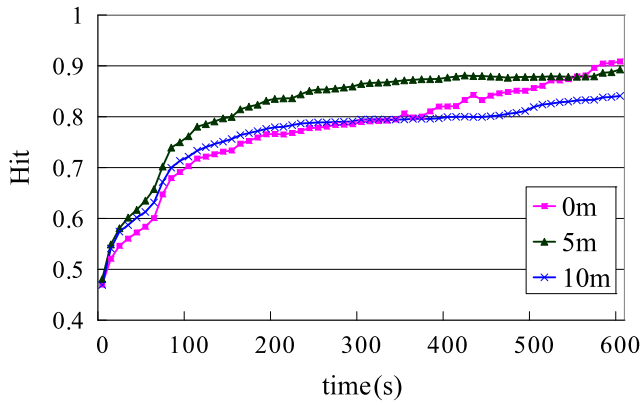

Fig. 16. Impact of average position error μ.

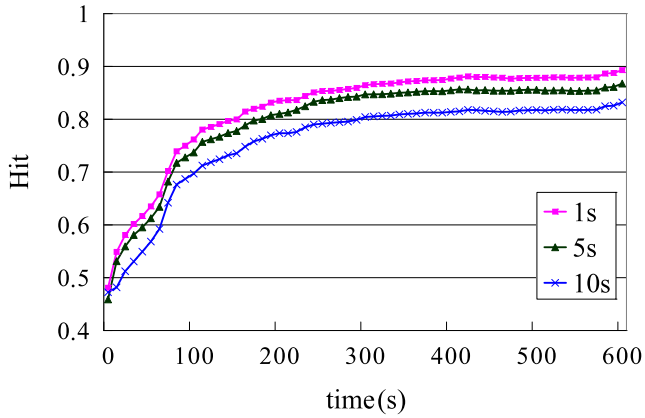

Fig. 17. Impact of GPS positioning frequency T_p.

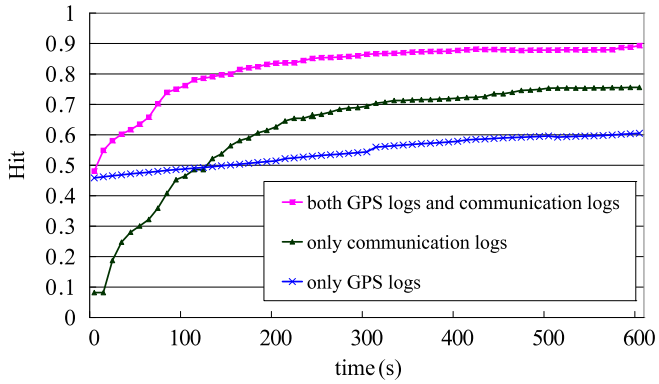

Fig. 18. Performance of each procedure.

From Fig. 18, we can see that the final map is the most accurate. The estimation accuracy of the GPS-based one is monotonically increasing, but it is not sufficient for practical use. Similarly, that of the communication-based one is also increasing, but it does not reach the final result. This clearly shows the necessity and effectiveness of the combined use of the two procedures.

4.3. Stationary nodes

All nodes were mobile in the above simulations. However, considering disaster-relief scenarios, the co-presence of stationary nodes (such as patients) with highly mobile nodes (such as first responders) should be considered. Therefore, we evaluated *Hit* under a scenario with mobile and stationary nodes. From Fig. 19, in the case of "all-mobile" nodes, *Hit* is larger and converges more quickly than in the case of "half-mobile–half-stationary" nodes, because we can obtain more GPS logs in the former case. However, *Hit* is about 0.9 in both cases.

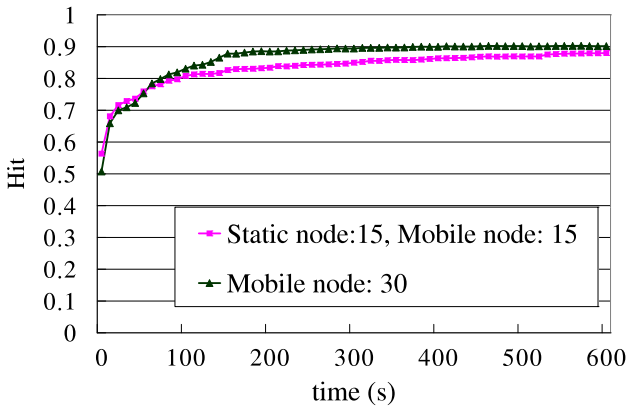

Fig. 19. Impact of static nodes.

Table 3
Analysis of communication logs.

Real experiment	Simulation	
	Success (%)	Failure (%)
Success	91.1	6.4
Failure	8.8	93.6

Table 4
Analysis of GPS logs.

Position error (m)	Native GPS	Corrected GPS
0–5	4742	4853
5–10	1996	2191
10–15	948	1106
15–20	605	627
20–25	361	201
25–30	150	7
30–35	82	0
35–40	49	0
40–45	6	0
45–50	1	0
50	45	0

5. Field experiments

We conducted two field experiments to evaluate the performance of our method in a real environment. One experiment was conducted to observe the performance difference between real and simulated environments. To this goal, we conducted the field experiment in the same situation as the simulation scenario of Section 4, where a part of our university campus was targeted. Another experiment was conducted to verify the algorithm performance in regions with many buildings.

5.1. Experiment on university campus

5.1.1. Log collection and analysis

We collected GPS logs and communication logs of ten persons for 600 s in a 150 m $\times$ 190 m region on Osaka University campus, shown in Fig. 10. We let each person (mobile node) have a Jennic JN5139 module [4] (IEEE802.15.4 module) and an I-O DATA USBGPS22 receiver [11] (GPS receiver). Each person moved along a pathway and randomly chose a direction (except backward) at each intersection. The moving speed was about 1.5 m/s. Every second, the position was measured and a beacon message was sent ($T_c = T_p = 1$ s). The transmission power was -18 dBm, and the maximum radio range $R_{\max}$ was about 50 m according to the two-ray ground model [5].

First, we compared the communication logs in the real environment with those in the simulation. Table 3 shows the success/failure ratio of communications in the simulation and in the real environment: "failure" means that the communication between two nodes failed even though they were closer than distance $R_{\max}$; otherwise the communication is regarded as a "success". From Table 3, we can see that more than 90% of the results match.

Second, we analyzed the GPS logs collected in the real environment. The "native GPS" column of Table 4 shows the distribution of the GPS position errors. We can see that some GPS logs contain large errors (more than 50 m), since the GPS signals were sometimes blocked by buildings. Such large errors may have a serious impact on the estimation accuracy.

(a) Using corrected GPS logs ($Hit = 0.86$).

(b) Using native GPS logs ($Hit = 0.77$).

Fig. 20. Generated maps.

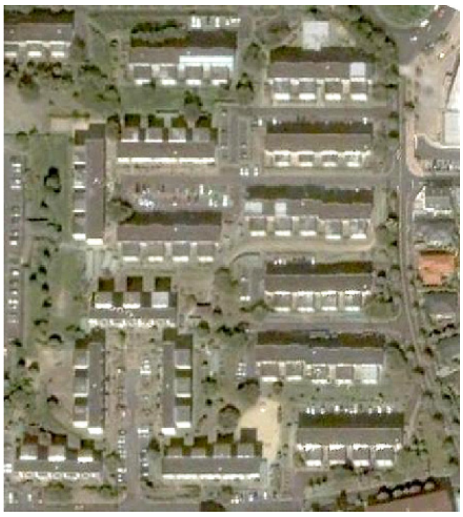

Fig. 21. Target map in the large area.

Hence, we try to eliminate such large position errors by considering their prior and posterior positions. For each measured position, if it is an "outlier" that deviates from the line segment between its prior and posterior positions, we modify the position to the midpoint of the line segment. The error distribution of the corrected GPS logs is shown in the column headed "Corrected GPS" in Table 4. We can see that position errors of the GPS logs are mitigated and the average position error is 6.33 m in this case.

5.1.2. Experimental results

Using the communication logs and the corrected GPS logs of the previous section, we evaluated *Hit*, the ratio of the correctly estimated cells to the entire cells. The generated map using this corrected GPS logs is shown in Fig. 20(a), which has similar *Hit* value (0.86) to that of the simulation experiment (0.89). From these results, we can say that our algorithm can be used in a real environment.

For comparison purposes, we measured *Hit* using the native GPS logs. Fig. 20(b) illustrates the generated map where *Hit* is 0.77. It is much lower than that with the collected GPS logs. Obviously accurate position information is significant, but a GPS system can be used for this purpose by simply filtering outliers.

5.2. Experiment in a region with apartment buildings

Another experiment was conducted in a 225 m $\times$ 250 m region (Fig. 21) with several apartment buildings. We collected GPS logs in this region every second using a Sony nav-u [12]. Here we recall that, from the experimental result of Section 5.1.2 (Table 3), 90% of the collected communication logs have the same successful ratio as those of the simulations. Therefore, using the map shown in Fig. 22 that models the target region, we used Qualnet and Wireless InSite to collect the communication logs. The simulation settings are the same as those in Section 4 and logs were collected for 600 s. We note that the part of the region where no node visited during these 600 s was excluded for the evaluation of *Hit*.

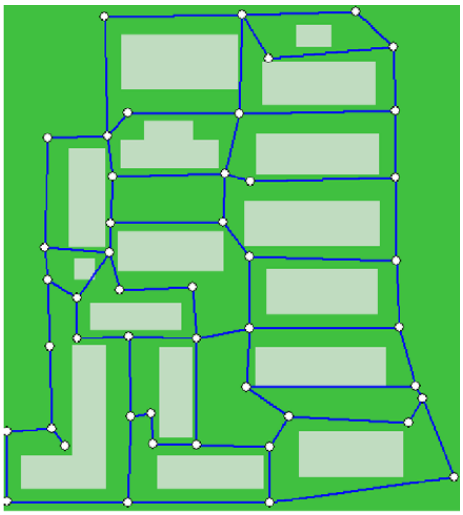

Fig. 22. Air photograph of the target map.

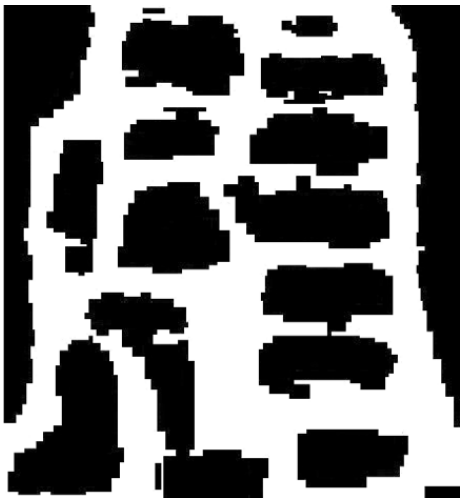

Fig. 23. Generated map in the large area.

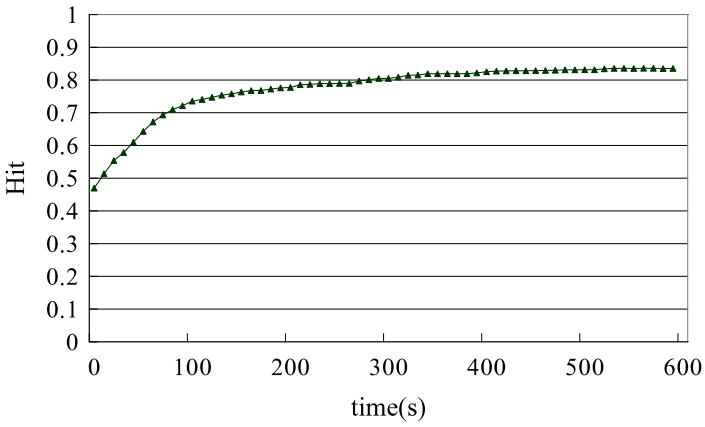

Fig. 24. *Hit* in the large area.

Fig. 23 shows the estimated map in which all the buildings and pathways are reproduced. Fig. 24 shows the values of *Hit* over time; the final value was 0.835. We note that the *Hit* value in the simulation with the normal distribution of position errors with mean 5 m and variance 0.01 is 0.842, which is very similar. Furthermore, we can observe that *Hit* converges at around 300 s, which means a reasonable time for map generation.

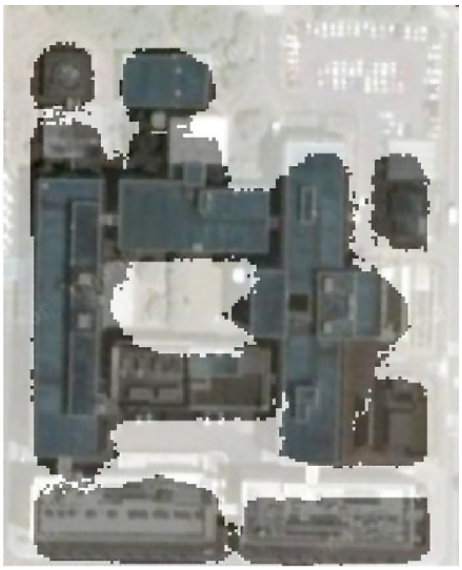

Fig. 25. Example of overlaying two images.

Table 5
RSS calculated by the knife-edge diffraction model (versus height of obstacles).

The distance between two nodes d (m)	The height of an obstacle h (m)	RSS (dBm)
1	9.5	−95.8
2	2.0	−95.1
3	1.0	−95.6
4	0.5	−95.1
5	–	–

6. Extensions and discussions

6.1. Combined use of existing and estimated maps

Although existing maps or satellite images may not present the latest geography nor detailed structure of buildings, they are helpful in enhancing the performance of our algorithm. We have implemented a prototype system to display an estimated map overlaid on an existing image like Fig. 25, which can increase the readability of the estimated map. Furthermore, we may use existing maps in the rectangular approximation phase of the algorithm. More concretely, we can use the information about the existing obstacles to estimate the shapes and positions of estimated obstacles by finding correspondence. This procedure may help our algorithm to estimate the shapes of obstacles more accurately. We may also generate maps by adding new estimated obstacles to existing maps, which is effective in reducing the computation time.

6.2. Increasing detection capability

When the two maps from GPS logs and communication logs are merged into a single map, we determine cell $g_{a,b}$ as an obstacle cell if $g_{a,b}$ is determined as an obstacle in both maps. Hence, our method regards the area over which nodes can communicate with each other but cannot transit as a non-obstacle area. For example, nodes may communicate with each other over low buildings, a river and a cliff but cannot go through these areas. We discuss the possible extensions to recognize such "low obstacles" and "impassable areas".

6.2.1. Low obstacle detection

Nodes may communicate over low obstacles by radio diffraction, and the proposed method may estimate that the cells in this region are non-obstacle cells. Here, we verify the effect of radio diffraction at the top of obstacles on the proposed method using the knife-edge diffraction model.

We assume that the nodes are JN5139 modules with transmission power −18 dBm (the maximum radio range $R_{\max}$ based on the knife-edge diffraction model is about 55 m in this case) and 2.4 GHz frequency. The height of their antenna is set to 1.5 m from the ground. The RSS threshold of the JN5139 module is −96 dBm. We assume a slim obstacle at the center of two nodes, as shown in Fig. 26. Based on the knife-edge diffraction model, we derive the maximum height h of the obstacle for distance d between two nodes changing h by 0.5 m, when RSS is stronger than −96 dBm.

From the results shown in Table 5, we can see that nodes can communicate with each other with 1 m distance even though the obstacle is rather tall. This is because over a shorter distance, attenuation by distance is smaller than that by

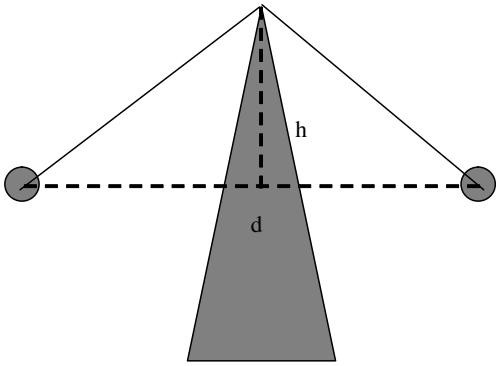

Fig. 26. Knife-edge diffraction model.

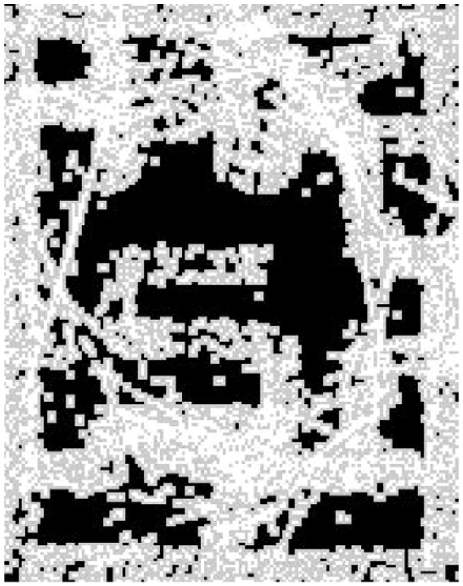

Fig. 27. Generated map with gradation.

diffraction. However, when the distance d is longer than 2 m, attenuation by distance has a great impact on the RSS. In particular, nodes cannot communicate if the distance is longer than 5 m. Furthermore, in the case of thick obstacles, the RSS is decreased because radio diffraction occurs at many points on the obstacles. As a result, it is not likely that nodes communicate with each other over low buildings.

6.2.2. Impassable area detection

The proposed method may regard impassable areas over which nodes can communicate with each other as non-obstacle areas. One plausible approach to recognize such areas is to detect the nodes' detour from GPS logs. However, since the nodes may not always follow the shortest path, we cannot rely on such a way of detection.

Therefore, we take an approach that uses the "likelihood" of obstacles instead of binary decision. For each cell, we calculate the ratio of the transit count over the cell to the expected average transit count. Then the color of each cell is determined as white color's RGB(255, 255, 255) value multiplied by the calculated ratio. As a result, the color of a cell can represent its likelihood of being an obstacle. Another simple approach is to represent the difference of two maps generated by the GPS-based procedure and the communication-based procedure using different colors. Figs. 27 and 28 illustrate the maps by these approaches, respectively.

6.3. Analysis of the amount of communication logs

Finally, we discuss the number of communication logs necessary to achieve sufficient accuracy. We revisit the result in Fig. 12. In all the cases of $R_{\max}$, the estimation accuracy is convergent when we use communication logs collected for more than 300 s, which correspond to 4000 communication logs. In order to evaluate the time to obtain 4000 communication logs, we conducted simulations varying the number of nodes (15, 30 and 45). The other parameters were set to the values in Table 2. It took 352 s, 143 s and 94 s with 15, 30 and 45 nodes, respectively. From these results, the proposed algorithm can generate a local map with 85% accuracy assuming only 15 members' GPS logs and communication logs collected for 6 min.

Fig. 28. Generated map with different colors.

7. Related work and contribution

In this section, we introduce existing techniques for map generation, and explain why it is difficult to use them in our case.

7.1. Object and geography recognition in various application domains

Recognizing the shapes, materials and positions of objects using cameras and sensors has been considered in many application domains for different purposes. In Intelligent Transport Systems (ITSs), many methodologies have been designed for vehicles to recognize obstacles and pedestrians to assist safe driving. For example, [13] proposes a method that identifies pedestrians and obstacles on roads using stereo cameras and range measurement sensors like infra-red sensors and laser sensors. The methods described in [14,15] estimate the positions of neighboring vehicles using sensor fusion techniques and inter-vehicle communication. In [16], the authors propose a method to detect and track road signs from landscape images, whereas [17] proposes a method that updates the existing map by discovering new roads using the GPS positions of cars.

Meanwhile, some methods of localizing nodes in sensor networks try to estimate such topology that involves "holes" where no node exists and no communication occurs over them [18–23], whereas [24] proposes an obstacle localization method that uses the effect of obstacles on wireless communication assuming that there are no sensors in the obstacle field. However, these methods are designed for sensor networks with a large number of stationary nodes. Therefore it is very difficult to apply them to our problem.

7.2. Position and map estimation for mobile robots

7.2.1. SLAM overview

Simultaneous localization and mapping (SLAM) techniques [25,26] have been well investigated. One common goal is to control the movement of autonomous robots in unknown environments such as disaster scenes. These SLAM techniques build maps of the surroundings and simultaneously estimate the positions of mobile robots. The methods described in [25,26] assume that each robot has cameras, range measurement sensors and gyroscopes, and the robot creates local maps using the information from these devices. Then the methods create an entire map of the environment by fusing the local maps based on the positions of mobile robots estimated by dead reckoning. In this way, the SLAM methods require accurate distance to obstacles and considerable computation power to process enormous amounts of data from the devices. In addition, many methods have been proposed using laser range sensors [27], sonic waves [28], ultrasonic waves [29] and cameras [30] to generate maps. However, these methods require special hardware for measurement, which is not ideal for disaster situations.

7.2.2. Basic principle of SLAM

We introduce the basic principle of SLAM approaches. We believe this is helpful in understanding the difference from our approach.

We denote a state to be estimated and measurement data that correspond to sensing from the environment as x and d, respectively. According to the Bayes rule, the probability of state x under d is given as

$$p(x|d) = \eta \cdot p(d|x) \cdot p(x) \tag{6}$$

where $p(d|x)$ denotes the likelihood of measurement data d under state x, and $p(x)$ represents the likelihood of state x before the measurement of d. η normalizes $p(d|x) \cdot p(x)$ to a probability distribution.

Since measurement d is obtained and state d is changed over time in robotic mapping, we introduce the notion of time to Eq. (6). We let x_t and d_t denote the state and measurement data at time t, and let D_t denote the sequence of measurement data till time t, i.e., $D_t = (d_0, \ldots, d_{t-1}, d_t)$. We obtain the following:

$$p(x_t|D_t) = \eta \cdot p(d_t|x_t) \cdot p(x_t|D_{t-1}). \tag{7}$$

We note that $p(x_t|D_{t-1})$ is a probability distribution obtained as the product of the transition probability from state x_{t-1} to state x_t and the likelihood of x_{t-1} under measurement D_{t-1}.

$$p(x_t|D_{t-1}) = \int p(x_t|x_{t-1}) \cdot p(x_{t-1}|D_{t-1}) \mathrm{d}x_{t-1}. \tag{8}$$

Using Eq. (8), Eq. (7) can be transformed into

$$p(x_t|D_t) = \eta \cdot p(d_t|x_t) \int p(x_t|x_{t-1}) \cdot p(x_{t-1}|D_{t-1}) \mathrm{d}x_{t-1}. \tag{9}$$

In robotic mapping, measurement data d_t are usually pairs of z_t and u_t, which are data from the robot's sensors (called sensor data) and robot motion commands in the time interval $[t-1, t)$, respectively. Knowing that robot motion commands are independent of states and that the sensor data do not affect the states, Eq. (9) can be transformed into

$$p(x_t|Z_t, U_t) = \eta \cdot p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) \cdot p(x_{t-1}|Z_{t-1}, U_{t-1}) \mathrm{d}x_{t-1} \tag{10}$$

where Z_t and U_t are the sequences of z and u till time t, respectively. Furthermore, since state x_t is usually a pair of a map m which is time-independent and a state s_t of the robot, we finally obtain

$$p(s_t, m|Z_t, U_t) = \eta \cdot p(z_t|s_t, m) \int p(s_t|u_t, s_{t-1}) \cdot p(s_{t-1}, m|Z_{t-1}, U_{t-1}) \mathrm{d}s_{t-1}. \tag{11}$$

Eq. (11) indicates that, for estimation of a map and a state of the robot, we need prior knowledge of $p(s_t|u_t, s_{t-1})$ and $p(z_t|s_t, m)$ in addition to the data and motion sequences Z_{t-1} and U_{t-1}. $p(s_t|u_t, s_{t-1})$ is a state transition probability from s_{t-1} to s_t with motion commands u_t. Since this probability is time independent, $p(s_t|u_t, s_{t-1})$ can be rewritten as $p(s|u, s')$, and this is generally called the motion model. This can be known beforehand as the robot's motion specification. $p(z_t|s_t, m)$ is also a time-independent probability that represents the likelihood of sensor data under a pair of a state and a map. This can be rewritten as $p(z|s, m)$, and is generally called the perception model. This can also be known by prior learning on how the sensor data are generated from states and maps.

7.2.3. Why not SLAM?

SLAM is designed for mobile robots. It is not suitable for first responders who move around the disaster sites, for the following reasons. First, it is difficult to control and record first responders' precise motions such as movement distance and direction; i.e., there is no motion model of first responders. It should also be noted that, unlike for robots, the "states" of those responders are not stable since they are engaged in rescue activities. Furthermore, it is not realistic to impose additional burdens on those responders to measure the environment. Distance measurement involves dedicated actions and devices. Taking pictures and video are of limited use at night and need additional information on shot directions and positions.

7.3. Our contributions

The contribution of our method is two-fold. First, we only use ad hoc communication devices and GPS receivers of mobile nodes. Since they are very general nowadays and they do not require dedicated actions for measurement, they can be used in rescue operations [2] or many other cases. Secondly, estimating obstacle maps using those devices is a very new and challenging problem. For this problem, we have developed an efficient and practical algorithm using both the position and the communication history of mobile nodes, incorporating image-processing techniques.

8. Conclusion

We have proposed an algorithm to estimate the shapes and positions of obstacles using mobile nodes' ad hoc communication devices and GPS receivers. The proposed algorithm estimates movable space and obstacles using GPS logs and communication logs, and refines the result by applying some image-processing procedures to obtain a readable map. Through several experiments in simulations and real environments, we have shown that our algorithm could generate readable and accurate maps.

As we stated in Section 1, medical doctors and rescue workers say that geographic information is very important in rescue and treatment actions in emergency situations. Therefore, in our ongoing project [2], we are trying to incorporate this algorithm into our "electronic triage system" for instant and automated generation of local maps. We will also conduct more experiments in real environments to assess the scalability and availability of our algorithm. This is part of our ongoing work.

Acknowledgement

This work was supported in part by Japan Science and Technology, CREST.

References

[1] Japanese Association for Disaster Medicine, Japanese Journal of Disaster Medicine 12 (2007) (in Japanese).
[2] T. Higashino, Advanced wireless communication technology for efficient rescue operations, Japan Science and Technology Agency. Available: http://www.jst.go.jp/kisoken/crest/en/area02/1-04.html.
[3] S. Minamimoto, S. Fujii, H. Yamaguchi, T. Higashino, Local map generation using position and communication history of mobile nodes, in: Proc. PerCom. 2010, IEEE Computer Society, USA, 2010, pp. 2–10.
[4] Jennic Ltd., JN5139 IEEE802.15.4/JenNet Evaluation Kit. Available: http://www.jennic.com/products/development_kits/jn5139_ieee802154_jennet_evaluation_kit.
[5] J.D. Parsons, The Mobile Radio Propagation Channel, Wiley, USA, 1992.
[6] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, D. Culler, The effects of ranging noise on multihop localization: an empirical study, in: Proc. IPSN 2005, IEEE Press, USA, 2005, pp. 73–80.
[7] W.C.Y. Lee, Mobile Communications Engineering, McGraw-Hill Professional, USA, 1982.
[8] J.C. Russ, The Image Processing Handbook, CRC press, USA, 2006.
[9] Scalable network technologies, QualNet web page. Available: http://www.scalable-networks.com/products/qualnet/.
[10] Remcom, Wireless insite web page. Available: http://www.remcom.com/wireless-insite.
[11] I-O DATA INC., USBGPS2 web page. Available: http://www.iodata.jp/product/mobile/gps/usbgps2/ (in Japanese).
[12] Sony Corp., nav-u web page. Available: http://www.sony.jp/nav-u/ (in Japanese).
[13] T. Gandhi, M.M. Trivedi, Pedestrian protection systems: issues, survey, and challenges, IEEE Transactions on Intelligent Transportation Systems 8 (3) (2007) 413–430.
[14] A. Polychronopoulos, M. Tsogas, A.J. Amditis, L. Andreone, Sensor fusion for predicting vehicles' path for collision avoidance systems, IEEE Transactions on Intelligent Transportation Systems 8 (3) (2007) 549–562.
[15] M. Rockl, T. Strang, M. Kranz, V2V communications in automotive multi-sensor multi-target tracking, in: Proc. VTC-2008-Fall, IEEE, USA, 2008, pp. 1–5.
[16] C.-Y. Fang, S.-W. Chen, C.-S. Fuh, Road-sign detection and tracking, IEEE Transactions on Vehicular Technology 52 (5) (2003) 1329–1341.
[17] J.J. Davies, A.R. Beresford, A. Hopper, Scalable, distributed, real-time map generation, IEEE Pervasive Computing 5 (4) (2006) 47–54.
[18] Y. Wang, J. Gao, J.S.B. Mitchell, Boundary recognition in sensor networks by topological methods, in: Proc. MobiCom 2006, ACM, USA, 2006, pp. 122–133.
[19] S. Funke, Topological hole detection in wireless sensor networks and its applications, in: Proc. the 2005 Joint Workshop on Foundations of Mobile Computing, ACM, USA, 2005, pp. 44–53.
[20] R. Zheng, A. Pendharkar, Obstacle discovery in distributed active sensor networks, in: Proc. INFOCOM 2009, IEEE, USA, 2009, pp. 909–917.
[21] S. Funke, C. Klein, Hole detection or: how much geometry hides in connectivity? in: Proc. The 22th Annual Symposium on Computational Geometry, ACM, USA, 2006, pp. 377–385.
[22] R. Ghrist, A. Muhammad, Coverage and hole-detection in sensor networks via homology, in: Proc. IPSN 2005, ACM, USA, 2005, p. 34.
[23] Q. Fang, J. Gao, L.J. Guibas, Locating and bypassing holes in sensor networks, Mobile Networks and Applications 11 (2) (2006) 187–200.
[24] F. Reichenbach, R. Salomon, D. Timmermann, Distributed obstacle localization in large wireless sensor networks, in: Proc. IWCMC 2006, IEEE, USA, 2006, pp. 1317–1322.
[25] H. Choset, K. Nagatani, Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization, IEEE Transactions on Robotics and Automation 17 (2) (2001) 125–137.
[26] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, IEEE Robotics & Automation Magazine 13 (2) (2006) 99–110.
[27] Y.D. Kwon, J.S. Lee, A stochastic map building method for mobile robot using 2-d laser range finder, Autonomous Robots 7 (2) (1999) 187–200.
[28] H.J. Jeon, B.K. Kim, Feature-based probabilistic map building using time and amplitude information of sonar in indoor environments, Robotica 19 (4) (2001) 423–437.
[29] O.M. Al-Jarrah, O.Q. Bani-Melhem, Building maps for mobile robot navigation using fuzzy classification of ultrasonic range data, Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology 11 (3–4) (2001) 171–184.
[30] A. Prusak, O. Melnychuk, H. Roth, I. Schiller, R. Koch, Pose estimation and map building with a time-of-flight-camera for robot navigation, International Journal of Intelligent Systems Technologies and Applications 5 (3) (2008) 355–364.