Journal of Information Processing Vol.21 No.2 198–205 (Apr. 2013)

[DOI: 10.2197/ipsjjip.21.198]

Regular Paper

Self-Estimation of Neighborhood Distribution for Mobile Wireless Nodes

Yuki Sakai[1,a) Akira Uchiyama[1,2] Hirozumi Yamaguchi[1,2] Teruo Higashino[1,2]

Received: April 17, 2012, Accepted: November 2, 2012

Abstract: In this paper, we propose a method to estimate the node distribution for pedestrians with information terminals. The method enables us to provide situation-aware services such as intellectual navigation that tells the user the best route to go around congested regions. In the proposed method, each node is supposed to know its location roughly (i.e., within some error range) and to maintain a density map covering its surroundings. This map is updated when a node receives a density map from a neighboring node. Each node also updates the density map in a timely fashion by estimating the change of the density due to node mobility. Node distribution is obtained from the density map by choosing cells with the highest density in a greedy fashion. The simulation experiments have been conducted and the results have shown that the proposed method could keep average position errors less than 10 m.

Keywords: portable terminals, location based services, mobile applications, adhoc networks

1. Introduction

A recent innovation of wireless communication technology has brought us possibilities to deploy infrastructure-less wireless applications. In Intelligent Transportation Systems (ITS), many research efforts have been conducted for situation awareness of pedestrians and vehicles based on DSRC for collision avoidance. For example, OKI has developed a DSRC attachment for mobile phones for pedestrian safety [1] by broadcasting positions of pedestrians obtained by GPS. People centric sensing [2] is also an emerging technology using sensing information such as traffic information from smartphones for urban sensing.

These studies indicate that ad-hoc wireless communication is a cost-efficient way of data fusion and diffusion among local agents. In particular, if pedestrians can estimate and obtain the information on their surroundings in real-time through ad-hoc communication, many services and applications can be provisioned without infrastructures. For example, it would be more beneficial to a human navigation system for emergency evacuation and to stranded commuters in disasters if information on the distribution of people in its surroundings is available. Some literature have proposed methods for people density estimation in urban areas. For example, in Ref. [3], Bluetooth scan is used for estimating the number of nearby nodes. Reference [4] has investigated people density estimation using locations of mobile phones obtained via base stations for large scale urban monitoring. However, real-time estimation of mobile node distribution by collaboration through ad-hoc networks is still challenging.

In this paper, we propose a method for mobile wireless nodes, which are pedestrians, to estimate the distribution of mobile nodes in their surroundings. In the proposed method, each node is assumed to know its location roughly (i.e., within some error range) and to maintain a density map covering its surroundings. This map is updated when a node receives a density map from a neighboring node. Each node also updates its density map in a timely fashion by estimating the change of node distribution over time due to node mobility.

The goal of our study is to propose an autonomous protocol to let mobile nodes have accurate node distribution with reasonable amount of wireless ad-hoc communication traffic. For estimating the node distribution, a mobile node may independently maintain and share positions of each node. However, the amount of data exchanged among mobile nodes may be large since a large number of nodes are expected in urban areas. To achieve a data size which is independent of the number of nodes, we use a density map where we divide a target region into grid cells and the expected numbers of nodes (i.e., density) in each cell are maintained. Nodes can estimate the current distribution of nodes from their own density maps by finding cells with a high density in a greedy fashion. To build a density map, with a certain interval, each node broadcasts its own density map where its *area of presence* (the area in which a true location is included) is merged. On receiving a density map from neighboring nodes, the node updates such a part of its own density map by the received density information which seems to be fresher.

We note that there is a clear trade-off between the freshness of density information and the required amount of wireless capacity to exchange density information. To pursue this trade-off, we have two key ideas. First, we provide an *estimation function* that

1 Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan
2 Japan Science Technology and Agency, CREST, Chiyoda, Tokyo 102–0076 Japan
a) y-sakai@ist.osaka-u.ac.jp

estimates the future density map based on its time-varying characteristics. As a simple example, if we know the maximum speed $V_{\max}$ of mobile nodes, an estimation function that estimates the density map after Δt time can be designed in such a way that each density in the current map is spread over $V_{\max} \cdot \Delta t$ region. Another function can be designed in such a way that the density is spread only to the directions toward which other nodes exist. This is based on the property that pedestrians walk on roads. Second, we design an adaptive protocol that controls the transmission interval of messages depending on the density of surroundings, in order to avoid similar density maps to be emitted to the wireless channel.

The simulation experiments have been conducted and the similarity between the real and estimated distributions has been measured. The results in three different scenarios have shown that the proposed method could attain average localization errors less than 10 m.

2. Related Work

In Vehicular Ad-hoc NETworks (VANETs), there have been various approaches to aggregate and disseminate several types of contexts like road surface condition, temperature, traffic jam information [5], [6], [7], [8]. Similar approaches have been considered in Wireless Sensor Networks (WSNs) [9], [10], [11], [12]. Some of them consider reducing the amount of data based on its similarity (i.e., elimination of data redundancy) and others consider in-network computing of given queries.

Our proposed method falls into these categories in the sense that it is aimed at aggregating (sensed) data with less amount of traffic. However, the proposed method is designed for mobile nodes to self-estimate their neighborhood distribution. Therefore, the data is time-varying in the scale of minutes while VANETs and WSNs target the aggregation of data such as load surface condition and wide-area traffic condition information which are relatively stable in long-term. Hence, we have to consider the trade-off between timeliness of data of mobile nodes' locations and the traffic overhead. We note that object detection and tracking in WSNs have to deal with real-time motion of objects (thus the data must be time-varying in very short term). However, these applications are not aimed at aggregating data but detecting objects.

As we stated in the introduction, each node has estimation functions to estimate the dynamic change of the node distribution, and exchanges the estimated result with others to help increase the accuracy of density maps. Also depending on the neighborhood density, each node controls the transmission interval. Based on these two ideas, we have designed a protocol that deals with a unique problem, that is, self-estimation of mobile node distribution. In this sense, our approach is original.

Our goal relates to localization algorithms [13], [14], [15], [16], which aim to estimate positions of nodes. However, the goal of localization algorithms is to estimate each node's position *by itself* and does not much care about positions of other nodes. Also their main concern is accuracy, while our challenge is to design a protocol that pursues the trade-off between the accuracy and the traffic.

There are several methods for estimating the density of people in urban areas [17], [18], [19]. Mobile space statistics [19] presented by NTT DOCOMO tracks populations of each area by counting mobile phone users observed at each base station. However, this approach aims at large-scale statistics such as the population in a city, which is different from our target. Reference [17] proposes a method to reconstruct the people flow from existing person-trip survey data. Reference [18] proposes a method for density estimation using coarse location information obtained from mobile phone call data. To the best of our knowledge, there is no research to provide a real-time estimation of node density in urban areas using the cooperation among mobile users. A straightforward approach is to upload position data obtained by GPS from all the nodes by using 3G networks. However, the 3G network traffic is overloaded in such an approach particularly in urban areas where a large number of people exist. In contrast, we use an ad hoc communication between mobile nodes to share the density information while avoiding 3G network overloading.

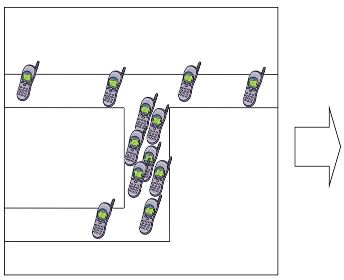
3. Self-Estimation of Neighborhood Distribution

3.1 Overview

We assume that each node i is equipped with a wireless device and knows its (rough) location through GPS or other technologies. We also assume that the region is divided into square cells with s(m) edge. Based on this cell representation of the geography, node i maintains a density map D_i, which represents the node density (i.e., the expected node numbers) in its surroundings. Concretely, D_i has $X_i \times Y_i$ elements and each element $d_{x,y}(1 \le x \le X_i, 1 \le y \le Y_i)$ represents the node density in the cell (x, y). We define the size X_i, Y_i and the location of the density map as node dependent values since each node may require its local view of the density map depending on applications. An example of a density map is shown in **Fig. 1**. We assume each node knows the maximum speed $V_{\max}$ of all the nodes to estimate the change of the node density in each cell by predicting movement of nodes. This is because the maximum speed may be estimated easier than the average speed, which largely depends on the time, the locations, the density and geometrical attributes such as the path's width.

Each node i executes the following procedures every t seconds.

(1) Node i updates its density map D_i by using a given estimation function f. We assume a typical moving pattern in the target environment is modeled into the estimation function.



(a) Positions of nodes

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0.3	0.4	0.4	0.5	0.5	0.2	0.1	0.1
0	0	0	1.5	1.0	0	0	0
0	0	0	1.4	1.3	0	0	0
0	0	0	1.0	1.0	0	0	0
0	0	0.4	0.6	0.6	0	0	0
0	0	0	0	0	0	0	0

(b) A density map

Fig. 1 An example of a density mapx.

According to this model, $f(D_i)$ diffuses the density values in each cell toward its surrounding cells that are supposed to be reachable within a message exchange interval denoted by t. This represents the estimated movement of other nodes. We note that in $f(D_i)$, if $d_{x,y}$ is less than a certain threshold denoted by TH_d after updating, $d_{x,y}$ is set to zero. For TH_d, we set the value which is too small or too old as the density information and which is therefore not useful any longer.

(2) Node i adds its presence information to D_i. To do this, firstly, node i obtains its area of presence (denoted by R_i) from the GPS or other measurement devices where R_i is the area which includes node i's true position. We represent R_i as a set of cells as follows;

$$R_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{in}, y_{in})\}$$

where n is the number of cells included in the area of presence. Thus the density value in each cell of R_i is $1/n$. Secondly, this value is added to the density value of each cell in the density map D_i. This procedure is executed only when the elapsed time since node i records R_i becomes longer than a certain Δt_i seconds. For Δt_i, we set the expected time for the density $1/n$ added to each cell to be less than a certain threshold (denoted by ε) due to the estimation function. Hence, Δt_i should be set according to the estimation function.

(3) Node i sends D_i to its neighbors.

(4) Node i updates D_i when i receives D_j from neighboring node j.

We explain the details of these procedures in the following section.

3.2 Algorithm

3.2.1 Estimation Function

Density maps are updated by the estimation function f, which is given beforehand. Typical movement patterns in the target region and/or the target nodes are modeled in the estimation function. Here, we describe (i) the diffuse estimation function, (ii) the limited diffuse estimation function, and (iii) the hybrid estimation function as examples of typical movement patterns and their estimation functions.

3.2.1.1 Diffuse Estimation Function

When the maximum speed of nodes is the only known fact, there is a possibility that each node moves toward any directions in the region. Thus, the diffuse estimation function divides density values in each cell to its neighboring cells which have a shared edge with the cell. A weight $\alpha (0 < \alpha < 1)$ is considered when a density value is divided so that aging of information can be regarded. Because the edge size of a cell is s(m) and updates are repeated every t seconds, the diffuse estimation function iterates this procedure $\lfloor t * V_{\max}/s \rfloor$ times. **Figure 2** (b) and **Fig. 3** show an example of the update by the diffuse estimation function and its pseudo-code, respectively.

	1.0	1.0	
	1.0	1.0	

(a) Before updating

	0.18	0.18	
0.18	0.54	0.54	0.18
0.18	0.54	0.54	0.18
	0.18	0.18	

(b) After 1 iteration (Diffuse; α=0.9)

	0.3	0.3	
	0.6	0.6	
	0.6	0.6	
	0.3	0.3	

(c) After 1 iteration (Limited diffuse;α=0.9)

Fig. 2 Update by estimation function.

```
for(step=0; step<floor(t*Vmax/s); step++){
 D'_i=D_i;
 foreach (d_(x,y) in D_i){
  d'_(x,y)=0.2*d_(x,y)*alpha;
  d'_(x-1,y)=d_(x-1,y)+0.2*d_(x,y)*alpha;
  d'_(x,y-1)=d_(x,y-1)+0.2*d_(x,y)*alpha;
  d'_(x+1,y)=d_(x+1,y)+0.2*d_(x,y)*alpha;
  d'_(x,y+1)=d_(x,y+1)+0.2*d_(x,y)*alpha;
  }
 D_i=D'_i;
}
return D_i;
```

Fig. 3 Diffuse estimation function.

In this function, Δt_i is determined based on k which satisfies the following condition:

$$\frac{\alpha^k}{2k^2 + 2k + 1} \leq \varepsilon \quad (1)$$

Here, k is the number of iterations by the diffuse estimation function. The left part in the above condition approximately denotes a density value in one cell after k steps, starting from a single cell of which a density value is 1. The denominator is the number of cells and the numerator means the freshness of the latest recorded area of presence. Each iteration is executed once in $s/V_{\max}$ seconds. Therefore,

$$\Delta t_i = \frac{k * s}{V_{\max}}. \quad (2)$$

3.2.1.2 Limited Diffuse Estimation Function

There are actually movable areas and unmovable areas if pedestrians walk on roads. Here, we consider an estimation function which distributes density values in each cell to only movable areas in its neighboring cells. We do not assume any maps but exploit a density map to estimate movable areas in this function.

Figure 2 (c) and **Fig. 4** show an example of an update by this limited diffuse estimation function and its pseudo-code, respectively. In this function, for each direction (i.e., up, bottom, left and right), we calculate the average density of cells to which the distance from the diffused cell $d_{x,y}$ is less than m cells. Then, if the result is more than TH_{move}, $d_{x,y}$ is divided by the number of directions which satisfy the condition and diffused to them. In the same way as the diffuse estimation function, α is regarded for aging. This procedure is iterated $\lfloor t * V_{\max}/s \rfloor$ times.

In the case of the limited diffuse estimation function, the number of cells which satisfy the condition varies every time it updates a density map. Thus, it is complicated to derive Δt_i precisely. For this reason, we use the same rule with the diffuse estimation function to determine Δt_i.

3.2.1.3 Hybrid Estimation Function

Because a density map is propagated among nodes hop by hop, the freshness of information in farther areas is lower. This means that it is sometimes hard to estimate movable areas in farther regions based on the limited diffuse estimation function as we described before. Hence, we combine both the diffuse estimation

```
for(step=0; step<floor(t*Vmax/s); step++){
 D'_i=D_i;
 foreach(d_(x,y) in D_i){
  expand_num=1;
  sum=0;
  for(j=1; j<=m; j++) sum+=d_(x+j,y);
  avg=sum/m; right=false;
  if(avg >= TH_move){
   right=true; expand_num++;
  }
  sum=0;
  for(j=1; j<=m; j++) sum+=d_(x-j,y);
  avg=sum/m; left=false;
  if(avg >= TH_move){
   left=true; expand_num++;
  }
  sum=0;
  for(j=1; j<=m; j++) sum+=d_(x,y+j);
  avg=sum/m; down=false;
  if(avg >= TH_move){
   down=true; expand_num++;
  }
  sum=0;
  for(j=1; j<=m; j++) sum+=d_(x,y-j);
  avg=sum/m; up=false;
  if(avg >= TH_move){
    up=true; expand_num++;
  }
  if(right) d'_(x+1,y)=d_(x+1,y)+1/expand_num*d_(x,y)*alpha;
  if(left) d'_(x-1,y)=d_(x-1,y)+1/expand_num*d_(x,y)*alpha;
  if(down) d'_(x,y+1)=d_(x,y+1)+1/expand_num*d_(x,y)*alpha;
  if(up) d'_(x,y-1)=d_(x,y-1)+1/expand_num*d_(x,y)*alpha;
  d'_(x,y)=1/expand_num*d_(x,y)*alpha;
 }
 D_i=D'_i;
}
return D_i;
```

Fig. 4 Limited diffuse estimation function.

function and the limited diffuse estimation function and propose the hybrid estimation function. In the hybrid estimation function, for the cells in the proximity of the current position, the limited diffuse estimation function is used and the diffuse estimation function is applied to distant areas.

We define the areas around the current position as the cells included in R_i, and use the limited diffuse estimation function for cells included in R_i and the diffuse estimation function for other cells. Δt_i is determined according to the same manner as the diffuse estimation function for the simplicity.

3.2.2 Recording Area of Presence

Each element $d'_{x,y}$ after recording node i's area of presence is calculated as defined below.

$$d'_{x,y} = \begin{cases} d_{x,y} + \dfrac{1}{n}, & \text{if } (x,y) \in R_i; \\ d_{x,y}, & \text{otherwise.} \end{cases} \tag{3}$$

where n denotes the number of elements in R_i. In this formula, the larger the size of R_i, the smaller the value added to each cell in R_i becomes.

3.2.3 Merging Density Maps

When a node i receives a density map D_j from another node j, node i merges D_i with D_j. Because each density map does not include any information which indicates the freshness of the density information in each cell, we regard a higher density as more fresh (i.e., newer) information. This policy is based on the observation that the density in each cell is diffused as time passes and hence a higher density is likely to be a fresher information. In merging of density maps, for each cell (x,y), the value $d'_{x,y}$ after the merging is computed as below.

$$d'_{x,y} = \max\{d^i_{x,y}, d^j_{x,y}\} \tag{4}$$

3.3 Getting Node Distribution from a Density Map

The node distribution is obtained from a density map D by finding cells with the highest density in a greedy fashion. The algorithm is described below in details.

(1) Find a cell $c = (x_c, y_c)$ with the highest density d_{x_c,y_c}. If the densities of all cells are zero, terminate the distribution estimation.

(2) Set an estimated position of a node to a point in c and initialize a set C of cells to c.

(3) If $d_{x_c,y_c} \geq 1.0$, subtract 1.0 from d_{x_c,y_c} and return to the first step.

(4) Repeat adding neighboring cells of all cells in C to C until $\sum_{e \in C} d_{x_e,y_e} \geq 1.0$. Here, a neighboring cell of a cell c is a cell that shares any border or corner of c. If the algorithm cannot find any neighboring cells, terminate the distribution estimation.

(5) Subtract 1.0 from C. For this purpose, we sort cells in C

into descending order $c(1), c(2), \ldots, c(n)$ in terms of the density. For $c(1), c(2), \ldots, c(m-1), c(m), c(m+1), \ldots, c(n)$, set $d_{x_{c(1)},y_{c(1)}}, \ldots, d_{x_{c(m-1)},y_{c(m-1)}}$ to 0 where $\sum_{i=1}^{m-1} d_{x_{c(i)},y_{c(i)}} < 1.0$ and $\sum_{i=1}^{m} d_{x_{c(i)},y_{c(i)}} \geq 1.0$. Then, subtract $1.0 - \sum_{i=1}^{m-1} d_{x_{c(i)},y_{c(i)}}$ from $d_{x_{c(m)},y_{c(m)}}$, and go to the first step.

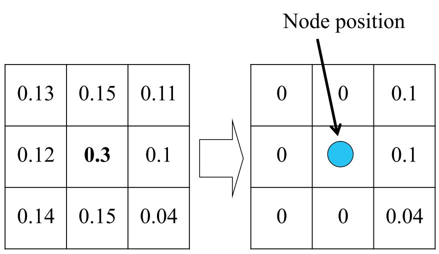
Intuitively, the algorithm sorts densities in descending order, and iterates the estimation of node positions from the cell with the highest density. At that time, we have to subtract 1.0 in total from the density map since a density of 1.0 corresponds to the presence of one node. **Figure 5** shows an example of this algorithm. In Fig. 5 (a), the highest density is 0.3 in the center cell, which is less than 1.0. Therefore, the densities of the neighboring cells and the center cell are sorted in descending order $\{0.3, 0.15, 0.15, 0.14, 0.13, 0.12, 0.11, 0.1, 0.04\}$. Then, all values larger than 0.11 are set to 0 since $(0.3+0.15+0.15+0.14+0.13+0.12) = 0.99$ and $(0.3+0.15+0.15+0.14+0.13+0.12+0.11) = 1.1$. Finally, $1 - 0.99 = 0.01$ is subtracted from 0.11, and a node position is estimated by choosing a point in the center cell as shown in Fig. 5 (b).

3.4 Reduction of Communication Overhead

Each node i sends its density map D_i every t seconds. The data size of D_i is inversely proportional to the size s^2 of a cell and proportional to the size of the target region. We introduce a technique which adjusts the view of a density map sent to neighbors, depending on the number of neighbors, in order to pursue the trade-off between the communication overhead and the accuracy.

We denote a sub-density map of D_i as $\hat{D}_i$ hereafter. Ideally, it is better to send a density map D_i every t seconds in order to propagate the density information to distant areas for a higher accuracy. However, if the density around a node is high, it seems enough to send density maps from a few nodes in the surroundings because the information in distant areas is likely to be very similar among those density maps.

Based on this idea, our technique uses a sub-density map $\hat{D}_i$ of which the size is fixed and smaller than the density map. Every t seconds, each node i selects either its density map D_i or its sub-density map $\hat{D}_i$ to broadcast. The density map is selected with the probability of $1/N_i$ where N_i is the number of i's neighbor nodes. In addition, node i broadcasts D_i only if it has not sent D_i in the last T seconds in order to guarantee that a density map is sent in a certain period of time.



(a) Density map (b) After estimation

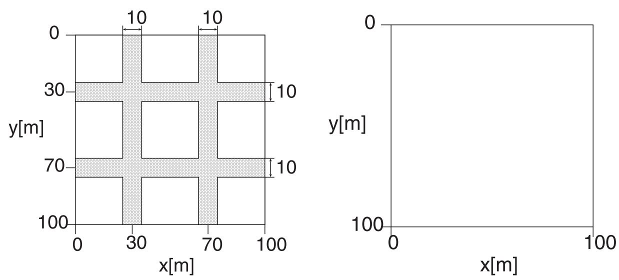
Fig. 5 Estimation of node distribution from a density map.

4. Experimental Results

4.1 Settings

We have evaluated the performance of the proposed method using a network simulator MobiREAL [20]. For simulation, we have used two maps of which the sizes are 100 m × 100 m. *Manhattan* in **Fig. 6** (a) which has 4 intersections and roads of 10 m width, and *free-space* in Fig. 6 (b). In the Manhattan map, nodes can only exist on roads, and in every map nodes were deployed uniformly before simulations. Nodes move along a road with a constant velocity which is randomly chosen from [0.1, 1.0] (m/s). Each node changes its direction to the opposite if it encounters a border, and randomly chooses one of the three directions except the backward direction if it enters an intersection. In the free-space map, the random waypoint mobility model [21] with pause time 0 and the moving speed range of [0.1, 1.0] (m/s) was used. 200 nodes moved according to the above mobility models in each scenario. The length s of grid cells was set to 2 m. We used the radio range of 10 m and the network bandwidth of 1 Mbps. We have assumed the location information R_i is obtained by GPS and given as a square region of size 49 m × 49 m of which the center is the real node position. The simulation settings are summarized in **Table 1**. We have empirically decided threshold values shown in Table 1.

Through the analysis of simulation results, we confirmed that the accuracy of the estimated node distribution is very similar among the nodes of different initial locations and moving speeds. Therefore, in the following results, we focus on the density map of a particular node (this node is denoted as p hereafter) if no explicit explanation is given. We have measured the positioning error to assess our method since it is quite intuitive and understandable not only for applications but also for people. By measuring the positioning error, we can compare our approach with the case where GPS positions of all the nodes are collected.



(a) Manhattan (b) free-space

Fig. 6 Simulation maps.

Table 1 Simulation settings.

Parameters	Manhattan	free-space
estimation function $f(D)$	hybrid (m=10)	diffuse
available density threshold TH_d (node/cell)	0.01	0.015
density threshold ε for location registration interval (node/cell)	0.002	0.002
sub-density map Tx interval t(s)	2	2
density map Tx interval T(s)	10	10

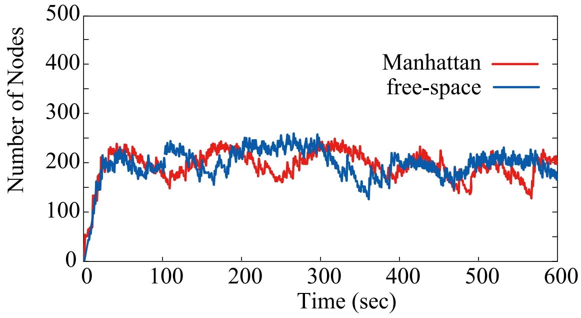


Fig. 7 Time vs. estimated number of nodes in a density map.

Table 2 Average number of nodes in an estimated density map.

	Estimated # of nodes
Manhattan (21 s∼600 s)	199.788
free-space (21 s∼600 s)	202.720

4.2 Results

4.2.1 Accuracy of a Number of Nodes

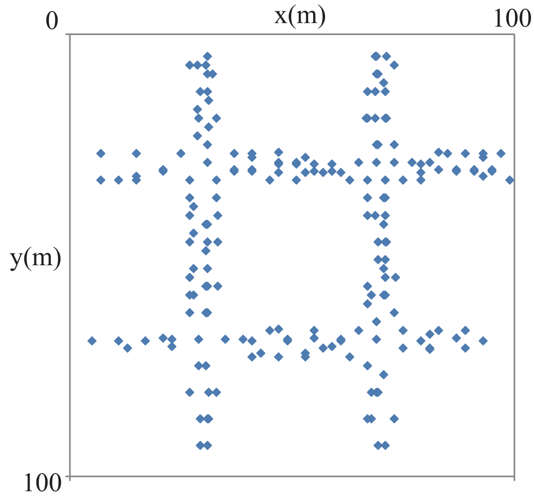
Figure 7 shows the estimated number of nodes in the two maps, along the progress of the simulation time. **Table 2** also shows the average number of nodes in each case. We can see that these averaged values are very close to the original values. In all the cases, large errors between the estimated and real node densities were measured before 30 sec. because it is the initial phase of the simulation where each node had started to collect information about the others and the density maps covering whole areas had not been constructed yet. Therefore, we focus on the state after 30 sec., where the estimated number of nodes was stable with small errors from the real density.

4.2.2 Similarity of the Estimated Distribution

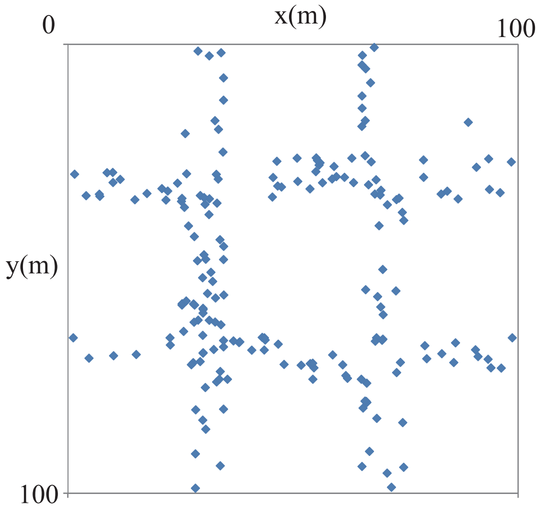
Figure 8 (a) and Fig. 8 (b) show the estimated node distribution of a node p at time 450 sec. and its corresponding real node distribution in the case of the Manhattan map. By comparing the real node distribution with the estimated one, we can see some errors in each node position. However, we can also observe that the estimated node distribution has dense and sparse areas quite similar to the real ones. Lines of nodes imply the roads in the real world and this indicates the estimated node distribution well captures the real node distribution.

To evaluate the accuracy of the estimated node distribution, we focus on positioning errors between the estimated positions and the real positions. Positioning errors can be defined as distances between nodes in the estimated node distribution and its corresponding nodes in the real node distribution. Note that we never know the node identification in the estimated distribution. Therefore, for each estimated position, the nearest node in the real distribution is regarded as the corresponding node, and the distance between the estimated position and the real position of the corresponding node is used for calculating positioning errors. Here, each node in the real distribution is selected as a corresponding node of an estimated position only once so as to evaluate positioning errors properly.

Figure 9 (a) and Fig. 9 (b) show averages of positioning errors (denoted as a solid line) in each simulation map. The averages of positioning errors in the both maps fluctuate due to the mobility. When the node p is in the proximity of the intersections or the center of the map, averages of positioning errors will be smaller because the node p can receive fresher information from different directions in such areas efficiently, and the accuracy of the density map of node p can be improved. **Table 3** shows the average of positioning errors throughout the whole simulation. We can see that the average positioning error in the Manhattan map is about 10 m and that in the free-space map is less than 10 m approximately. These results mean that the proposed method can correctly estimate the node distribution in terms of the street level considering the road width of 10 m and the road segment length of 40 m in the Manhattan map. From the above results and observations of the average positioning error as a quantitative criterion, we can confirm the effectiveness of the proposed method.



(a) Estimated (Manhattan)

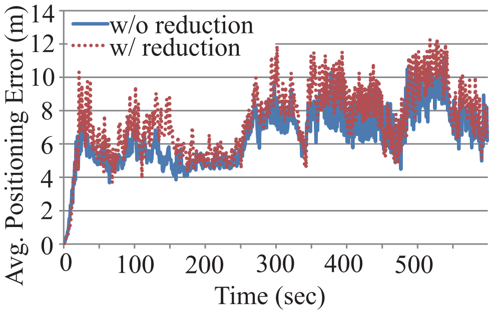


(b) Real (Manhattan)

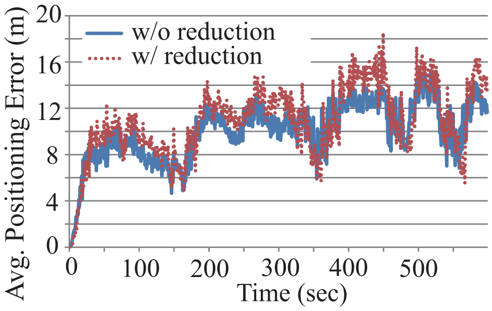
Fig. 8 Real node distribution and estimated node distribution of node p (at 450 sec.).

4.3 Reduction in Communication Overhead

In our technique, the target region is divided into cells. The number of cells is 2,500 in the default simulation setting, and we assume that each cell requires 4 bytes. Then, the data size of a density map is 10 Kbytes. Each node sends its density map periodically and hence the communication overhead may be large. To reduce this communication overhead, we use a sub-density map



(a) free-space



(b) Manhattan

Fig. 9 Time vs. average positioning errors in estimated distribution of node *p*.

Table 3 Avg. positioning errors in estimated node distribution of node *p*.

	Manhattan	free-space
w/o reduction	10.03	6.50
w/ reduction	10.89	7.39

Table 4 Comparison of average bandwidth per node.

	Manhattan	free-space
w/o reduction (kbps)	40	40
w/ reduction (kbps)	16.16	20

as we mentioned in Section 3.4.

In order to see the effect of this scheme, we evaluated the amount of traffic. The result is shown in **Table 4**. We could confirm that our scheme could reduce approximately 50–75% of the original traffic.

From the results shown in Fig. 9 (a) and Fig. 9 (b), we can see the average of positioning errors increases as time elapses in most cases with message reduction (denoted as dotted line), compared with the cases without message reduction. The second row of Table 3 also shows the averages of positioning errors with message reduction are 109% and 114% as much as those without message reduction in the Manhattan map and in the free-space map respectively. Obviously, there is a trade-off between the communication overhead and the accuracy of an estimated node distribution. Therefore, it is important to determine the parameters on the communication appropriately.

5. Discussion

The proposed method uses a cell matrix to represent a density map. The cell matrix facilitates the computation like merging and the mobility estimation, while the data size may be large, depending on both the region and cell sizes. In WSNs, there is a method to build a contour map of the data sensed by wireless sensor nodes [9], [22]. Some other possibilities use some encoding technique to compress the map. We are trying to clarify their advantages and disadvantages in terms of the trade-off between the computation overhead and the data size.

We also discuss another important issue on the position information. In the proposed method, each node may provide its position information with some error range. This has the following two advantages, (i) robustness to position errors caused by the GPS or other measurements such as position estimation methods like Sextant [23] and UPL [15] due to their likelihood estimation in range-free localization, and (ii) privacy protection in which intentionally randomized positions obscure the true position.

The maximum speed V_{max} affects the estimation accuracy. If we use overestimated maximum speeds, the accuracy degrades because the estimated density spreads faster than the real speeds. In this sense, the maximum speed used in the simulation is overestimated since we have used 1.0 m/s as the maximum speed although the real speeds uniformly distributed within [0.1, 1.0] (m/s). Nevertheless, our approach has achieved a reasonable performance.

6. Conclusion

In this paper, we have proposed a method for pedestrians to self-estimate the node distribution in their proximity in real-time using ad-hoc wireless communications among these nodes. We have conducted simulation experiments to see the accuracy and the communication overhead of the proposed method. Through quantitative evaluation by measuring positioning errors, we have confirmed the average position error is less than 10 m, which is comparable with GPS errors. This result indicates our method estimates the node distribution accurately.

One of our potential application domain is personal navigation. In huge shopping centers and fireworks festivals (in the case of Japan) in which many people get around, observing their locations through their mobile terminals will be helpful not only for commercial use but also for safe navigation toward exits.

Assuming these potential application examples, we are planning to conduct simulations in more realistic environments, to determine appropriate parameter settings and to validate the usefulness of the method. Furthermore, the autonomy of the protocol is our important goal where protocol parameters like message transmission intervals can be autonomously converged into appropriate values in each cell depending on its neighborhood densities for zero-configuration.

References

[1] OKI: OKI Press Release, available from ⟨http://www.oki.com/en/press/2009/01/z08113e.html⟩.

[2] Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musolesi, M., Fodor, K. and Ahn, G.-S.: The Rise of People-Centric Sensing, *IEEE Internet Computing*, Vol.12, No.4, pp.12–21 (2008).

[3] Wepper, J. and Lukowicz, P.: Collaborative Crowd Density Estimation with Mobile Phones, *Proc. International Workshop on Sensing Applications on Mobile Phones* (*PhoneSense*) (2011).

[4] Horanont, T. and Shibasaki, R.: An Implementation of Mobile Sensing for Large-Scale Urban Monitoring, *Proc. International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems* (*UrbanSense*) (2008).

[5] Korkmaz, G., Ekici, E., Özgüner, F. and Özgüner, U.: Urban Multi-hop Broadcast Protocol for Inter-vehicle Communication Systems,

Proc. ACM International Workshop on VehiculAr Inter-NETworking, Systems, and Applications (*VANET*), pp.76–85 (2004).
[6] Lochert, C., Scheuermann, B. and Mauve, M.: Probabilistic Aggregation for Data Dissemination in VANETs, *Proc. ACM International Workshop on VehiculAr Inter-NETworking, Systems, and Applications* (*VANET*), pp.1–8 (2007).
[7] Yu, B., Gong, J. and Xu, C.: Catch-up: A Data Aggregation Scheme for VANETs, *Proc. ACM International Workshop on VehiculAr Inter-NETworking, Systems, and Applications* (*VANET*), pp.49–57 (2008).
[8] Zhao, J. and Cao, G.: VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks, *IEEE Trans. Vehicular Technology*, Vol.57, No.3, pp.1910–1922 (2008).
[9] Gupta, I., Renesse, R. and Birman, K.: Scalable Fault-tolerant Aggregation in Large Process Groups, *Proc. IEEE International Conference on Dependable Systems and Networks* (*DSN*), pp.433–442 (2001).
[10] Madden, S., Franklin, M., Hellerstein, J. and Hong, W.: TAG: A Tiny AGgregation service for ad-hoc sensor networks, *SIGOPS Operating Systems Review*, Vol.36, No.SI, pp.131–146 (2002).
[11] Boulis, A., Ganeriwal, S. and Srivastava, M.: Aggregation in Sensor Networks: An Energy Accuracy Trade-off, *Ad Hoc Networks*, Vol.1, No.2-3, pp.317–331 (2003).
[12] Papadopouli, M. and Schulzrinne, H.: Effects of Power Conservation, Wireless Coverage and Cooperation on Data Dissemination among Mobile Devices, *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing* (*MobiHoc*), pp.117–127 (2001).
[13] Goldenberg, D., Bihler, P., Cao, M., Fang, J., Anderson, B., Morse, A. and Yang, Y.: Localization in Sparse Networks Using Sweeps, *Proc. ACM International Conference on Mobile Computing and Networking* (*MobiCom*), pp.110–121 (2006).
[14] Li, M. and Liu, Y.: Rendered Path: Range-free Localization in Anisotropic Sensor Networks with Holes, *Proc. ACM International Conference on Mobile Computing and Networking* (*MobiCom*), pp.51–62 (2007).
[15] Uchiyama, A., Fujii, S., Maeda, K., Umedu, T., Yamaguchi, H. and Higashino, T.: Ad-hoc Localization in Urban District, *Proc. IEEE International Conference on Computer Communications* (*Infocom*), pp.2306–2310 (2007).
[16] He, T., Huang, C., Blum, B., Stankovic, J. and Abdelzaher, T.: Range-free Localization Schemes for Large Scale Sensor Networks, *Proc. ACM International Conference on Mobile Computing and Networking* (*MobiCom*), pp.81–95 (2003).
[17] Sekimoto, Y., Shibasaki, R., Kanasugi, H., Usui, T. and Shimazaki, Y.: PFlow: Reconstructing People Flow Recycling Large-Scale Social Survey Data, *IEEE Pervasive Computing*, Vol.10, No.4, pp.27–35 (2011).
[18] Horanont, T. and Shibasaki, R.: Nowcast of Urban Population Distribution using Mobile Phone Call Detail Records and Person Trip Data, *Proc. International Conference on Computers in Urban Planning and Urban Management* (*CUPUM*) (2011).
[19] NTT DOCOMO Technical Journal Editorial Office: Measures for Recovery from the Great East Japan Earthquake Using NTT DOCOMO R&D Technology, *NTT DOCOMO Technical Journal*, Vol.13, No.4, pp.96–106 (2012).
[20] MobiREAL: MobiREAL Simulator, available from ⟨http://www.mobireal.net/⟩.
[21] Broch, J., Maltz, D., Johnson, D., Hu, Y.-C. and Jetcheva, J.: A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols, *Proc. ACM International Conference on Mobile Computing and Networking* (*MobiCom*), pp.85–97 (1998).
[22] Xu, Y., Lee, W.-C. and Mitchell, G.: CME: A Contour Mapping Engine in wireless sensor networks, *Proc. International Conference on Distributed Computing Systems* (*ICDCS*), pp.133–140 (2008).
[23] Guha, S., Murty, R. and Sirer, E.: Sextant: A Unified Node and Event Localization Framework Using Non-convex Constraints, *Proc. International Symposium on Mobile Ad Hoc Networking and Computing* (*MobiHoc*), pp.205–216 (2005).

Yuki Sakai received his M.E. degrees in Information Science and Technology from Osaka University, Japan in 2009. His current research interests include applications and services on mobile and overlay networks.

Akira Uchiyama received his M.E. and Ph.D. degrees in Information and Computer Science from Osaka University in 2005 and 2008, respectively, and is currently an Assistant Professor at Osaka University. He was a visiting scholar in University of Illinois at Urbana-Champaign in 2008 and a research fellow of the Japan Society for the Promotion of Science from 2007 to 2009. His current research interests include localization and applications in wireless networks and pervasive computing. He is a member of IEEE.

Hirozumi Yamaguchi received his B.E., M.E. and Ph.D. degrees in Information and Computer Sciences from Osaka University, Japan in 1994, 1996 and 1998, respectively. He is currently an Associate Professor at Osaka University. His current research interests include design, development, modeling and simulation of mobile and wireless networks and applications. He is a member of IEEE.

Teruo Higashino received his B.S., M.S. and Ph.D. degrees in Information and Computer Sciences from Osaka University, Japan in 1979, 1981 and 1984, respectively. He joined the faculty of Osaka University in 1984. Since 2002, he has been a Professor in Graduate School of Information Science and Technology at Osaka University. His current research interests include design and analysis of distributed systems, communication protocol and mobile computing. He is a senior member of IEEE, a fellow of IPSJ, and a member of ACM and IEICE of Japan.