

Self-Estimation of Neighborhood Density for Mobile Wireless Nodes

Junji Hamada[†], Akira Uchiyama^{†,‡}, Hirozumi Yamaguchi^{†,‡},
Shinji Kusumoto[†], and Teruo Higashino^{†,‡}

[†]Graduate School of Information Science and Technology, Osaka University

[‡]Japan Science Technology and Agency, CREST

Abstract. In this paper, we propose a method to estimate the density of nodes for pedestrians and/or vehicles with information terminals. The method enables us to provide intelligent services which are environment-aware with highly dynamic movement of nodes such as intellectual navigation that tells the user the best route to detour 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. Also by estimating the change of the density, taking into account the movement characteristics of nodes, the density map is updated in a timely fashion. The simulation experiments have been conducted and the results have shown the accuracy of the estimated density maps.

1 Introduction

Recent innovation of wireless communication technology has brought us possibilities to deploy infrastructure-less wireless applications. For example, in Intelligent Transportation Systems (ITS), collision avoidance systems using Inter-Vehicle Communication (IVC) have been developed and are now being put on practical use [1]. IVCs have been investigated for other applications like traffic and environment information acquisition [2–5].

A number of such studies commonly indicate that ad-hoc communication using short-range wireless devices improves the cost and efficiency of data fusion and diffusion, which have been done by infrastructure. In particular, if moving vehicles and 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 limitations due to deployment of infrastructures. For example, a human navigation system for emergency evacuation will be more intelligent if information on the density of people in its surroundings can be fed into the route decision engine of the system. However, real-time density estimation of mobile nodes by collaboration through ad-hoc networks has not been investigated yet.

In this paper, we propose a method for mobile wireless nodes, which may be pedestrians or vehicles with information terminals, to estimate the density

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. Also by estimating the change of the density, taking into account the movement characteristics of nodes, it is updated in a timely fashion.

The goal of our study is to propose an autonomous protocol to let mobile nodes have accurate density maps with reasonable amount of wireless ad-hoc communication traffic. 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 that the received cones density information seems more fresh. 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 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 value in the current map is spread over $V_{max} \cdot \Delta t$ region. Another function can be designed in such a way that the value is spread only to the directions toward which other nodes exist if mobile nodes are vehicles. This is based on the property that vehicles follow others. 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 correlation between the real and estimated density maps has been measured. The results in four different scenarios have shown that the proposed method could attain high accuracy of the estimated density maps.

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 [2–5]. Similar approaches have been considered in the field of Wireless Sensor Networks (WSNs) [6–9]. Some of them consider aggregating 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 density. Therefore, the data is time-varying in the scale of minutes while VANETs and WSNs target aggregation of data such as road surface condition and wide-area traffic condition information which are relatively stable in long-term. Hence,

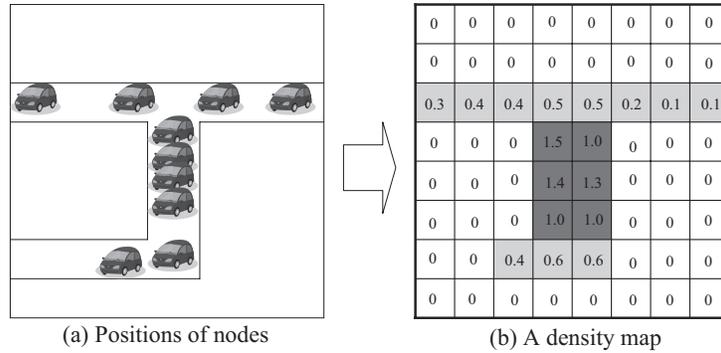


Fig. 1. Example of Density Map

we have to consider the trade-off between timeliness of data of mobile nodes' locations and 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 density map, 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 density for mobile nodes. In this sense, our approach is original.

From the perspective of geographical information, our goal relates to localization algorithms [10–13], 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. Hence, our goal is different from localization algorithms.

3 Self-Estimation of Neighborhood Density

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 geography, node i maintains a density map D_i , which represents locations of other nodes in its surroundings. Concretely, D_i has $X_i \times Y_i$ elements and each element $d_{x,y}$ ($1 \leq x \leq X_i, 1 \leq y \leq Y_i$) represents the node density in the cell (x, y) . 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. For example, this can be estimated based on the speed limits in the case of vehicles.

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. According to this model, $f(D_i)$ diffuses node density 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 density information, and is therefore not useful any longer.
2. Node i adds its location information to D_i . To do this, firstly, node i obtains its area of presence (denoted by R_i) from 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;

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

where n is the number of cells included in the area of presence. Thus the expected density 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 value $1/n$ 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

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.

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 the value of density in each cell to its neighboring cells which have a shared edge with the cell. An weight $\alpha(0 < \alpha < 1)$ is considered when a value of density 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

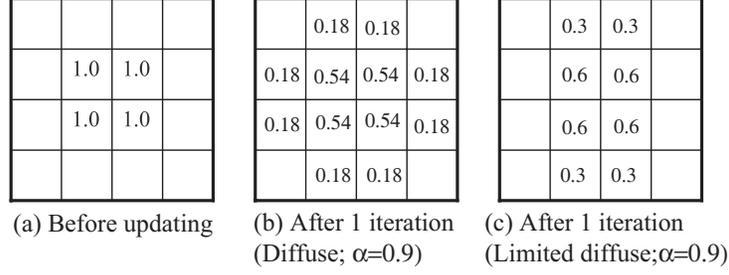


Fig. 2. Update by Estimation Function

```

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

this procedure $\lfloor t * V_{max}/s \rfloor$ times. Fig. 2(b) and Fig. 3 show an example of the update by the diffuse estimation function and its pseudo-code, respectively.

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

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

Limited Diffuse Estimation Function. There are movable areas and unmovable areas if the target nodes are pedestrians or vehicles. Here, we consider an estimation function which distributes density 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.

Fig. 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 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 .

Hybrid Estimation Function. Because a density map is propagated among nodes step by step, the freshness of information in further areas is lower. Hence, it is sometimes hard to estimate movable areas in further regions based on the limited diffuse estimation function as we described before. We combine both the diffuse estimation 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 in the same way that the diffuse estimation function does for the simplicity.

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

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

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.

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 information which indicates freshness of density information in each cell, we regard higher density as more fresh (*i.e.* newer) information. This policy is based on the observation that density in each cell is diffused as time passes and hence higher density is likely to be fresh information. In merging of density maps, for each cell (x,y) , the value $d'_{x,y}$ after the merging is computed as below.

$$d'_{x,y} = \max\{d_{x,y}^i, d_{x,y}^j\} \quad (4)$$

```

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)=d_(x,y)+1/expand_num*d_(x,y)*alpha;
  }
  D_i=D'_i;
}
return D_i;

```

Fig. 4. Limited Diffuse Estimation Function

3.3 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 communication overhead and 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 density information to

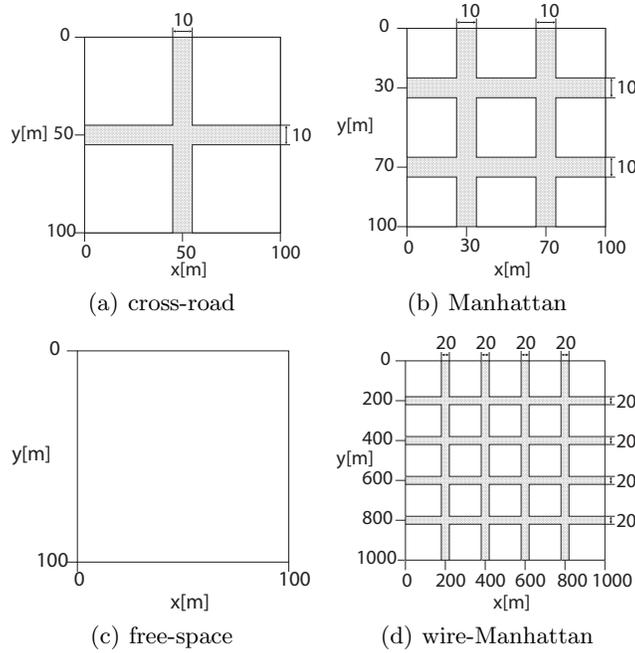


Fig. 5. Simulation Maps

distant areas for 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 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 S_i is defined as below:

$$S_i = \frac{\beta}{N_i} \quad (5)$$

where N_i is the number of neighbors for node i and β is a certain constant. Every t seconds, each node i sends either its density map D_i with the probability of $1/N_i$ or its sub-density map \hat{D}_i otherwise. 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,.

4 Experimental Results

4.1 Settings

We have evaluated the performance of the proposed method using a network simulator MobiREAL [14].

Table 1. Simulation Settings

Parameters	cross-road	Manhattan	free-space	wire-Manhattan
area size (m×m)	100 × 100	100 × 100	100 × 100	1000 × 1000
road width (m)	10	10	10	20
length s of cell (m)	2	2	2	20
radio range R (m)	10	10	10	150
velocity of nodes (m/s)	[0.1, 1.0]	[0.1, 1.0]	[0.1, 1.0]	[36, 54](km/h)
bandwidth (Mbps)	1	1	1	1
number of nodes	200	200	200	453
estimation function $f(D)$; $m=10$ for hybrid	hybrid	hybrid	diffuse	hybrid
threshold TH_d of effective density in density map (node/cell)	0.015	0.01	0.015	0.055
threshold ε of effective density in area of presence (node/cell)	0.002	0.002	0.002	0.002
size of R_i (m)	49 × 49	49 × 49	49 × 49	60 × 60
transmission interval t of par- tial density map (s)	2	2	2	2
maximum transmission inter- val T of density map (s)	10	10	10	10

We have used two simulation areas of which the sizes are 100m × 100m and 1,000m × 1,000m. These areas have several roads of 10m or 20m width. For the 100m × 100m area, we have used three maps; *cross-road* in Fig. 5(a) which has only one intersection, *Manhattan* in Fig. 5(b) which has 4 intersections, and *free-space* in Fig. 5(c). For the 1,000m × 1,000m area, we have used the map called *wire-Manhattan* in Fig. 5(d) which has 8 roads and 16 intersections. In these maps except the free-space map, nodes can only exist on roads, and in every map nodes were deployed uniformly before simulations. Each node moves along a road with a constant velocity which is randomly chosen from [0.1, 1.0](m/s) (in the cases of cross-road, Manhattan and free-space assuming pedestrians) or [36, 54](km/h) (in the case of wire-Manhattan assuming vehicles) at the beginning of simulations. 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. Simulation time is 600 seconds. The simulation settings are summarized in Table 1.

Through the analysis of simulation results, we have confirmed that the accuracy of density maps was very similar among the nodes of different initial locations and moving speeds. Therefore, in the following, we focus on the density map of a particular node (this node is denoted as p) if no explicit explanation is given.

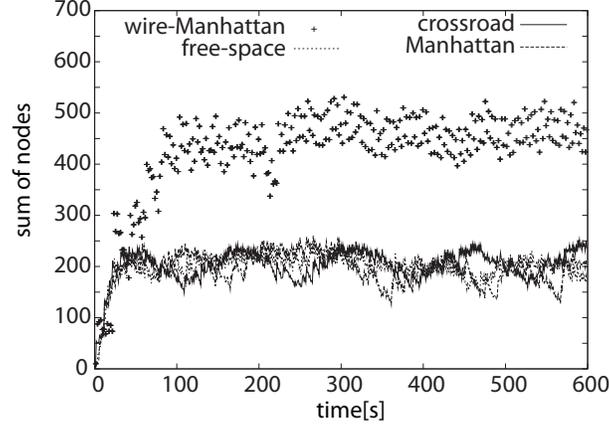


Fig. 6. Estimated Number of Nodes in Density Map of Node p

Table 2. Average Number of Nodes in Estimated Density Map

	Estimated # of nodes
cross-road (21s~600s)	206.098
Manhattan (21s~600s)	199.788
free-space (21s~600s)	202.720
wire-Manhattan (101s~600s)	455.595

4.2 Results

Accuracy of Number of Nodes Fig. 6 shows the estimated number of nodes in the four maps, along the progress of simulation time. Also, Table 2 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 except the wire-Manhattan, the large errors between the estimated and real node densities were measured before 30sec. because it is the initial phase of simulation where each node had started to collect information about the others and the density maps had not been constructed yet. Therefore, we focus on the state after 30sec., where the estimated number of nodes was stable with small errors from the real density. Since the size is quite larger than the others in the wire-Manhattan map, it took about 100sec. to obtain the density information in this case. Nevertheless, it also has the stable state after 100sec. where the estimated number of nodes was stable as well.

Fig. 7(a) and Fig. 7(b) show the real node distribution and its corresponding estimated density map of node p at time 450sec. in the case of the cross-road map. At this time, node p was at the point (50,50) (near the intersection). By comparing the estimated density map with the real node distribution, we can see some errors in the places away from the intersection. However, we can also observe that the estimated densities in the regions except roads were almost zero

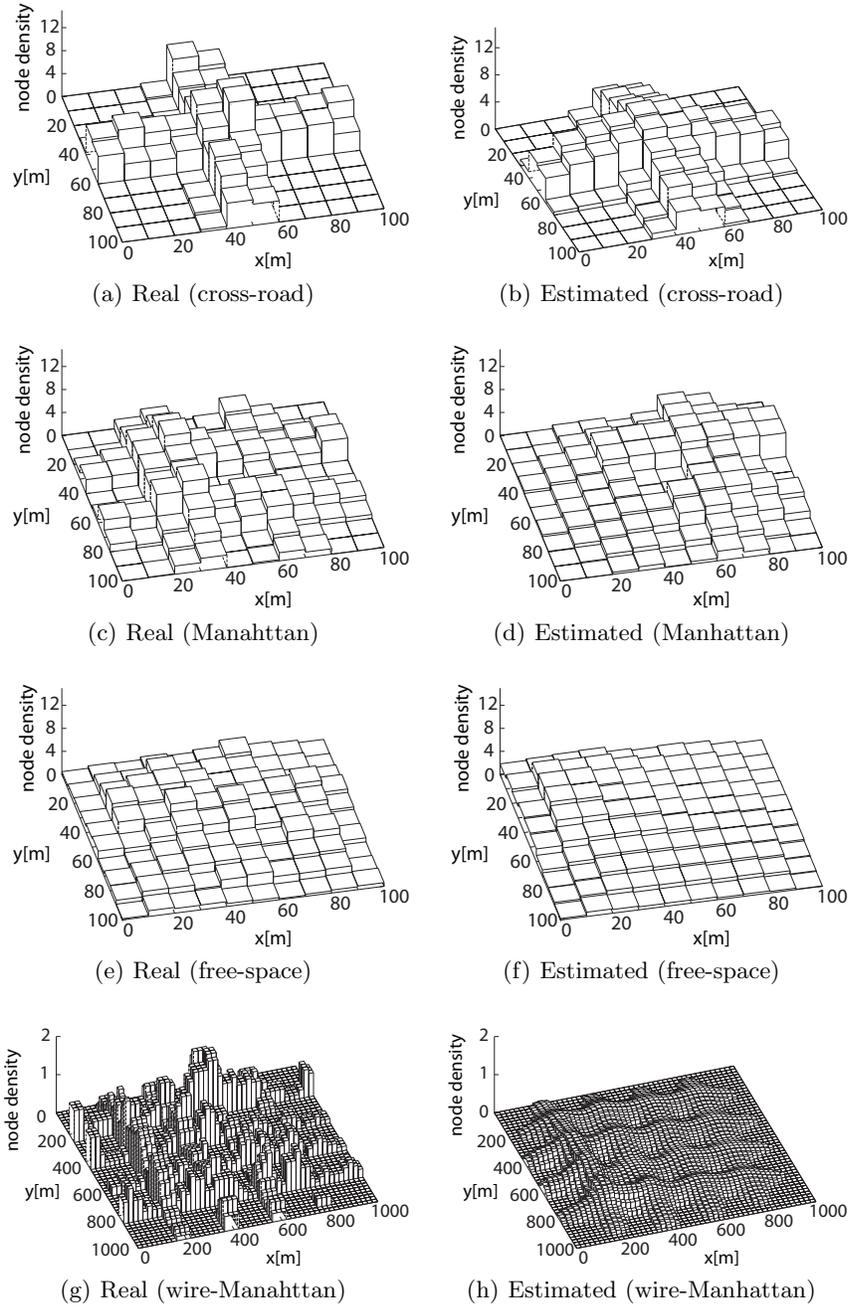
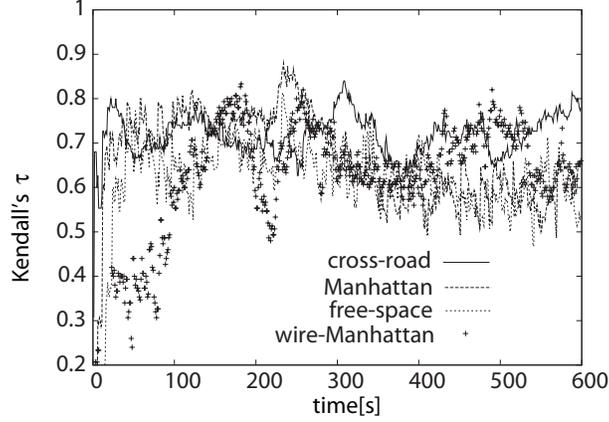


Fig. 7. Real Node Distribution and Estimated Density Map of Node p (at 450sec.)

Table 3. Average Kendall's τ

	Granularity			
	1×1	2×2	5×5	10×10
cross-road (31s~600s)	0.751	0.761	0.755	0.718
Manhattan (31s~600s)	0.545	0.568	0.595	0.674
free-space (31s~600s)	0.538	0.571	0.607	0.635
wire-Manhattan (101s~600s)	0.463	0.523	0.646	0.670

**Fig. 8.** Time vs. Kendall's τ ($g \times g=10 \times 10$)

and those around the intersection were high. This result indicates the estimated density well captures the real node distribution. Similarly, Fig. 7(c) and Fig. 7(d) show the result in the case of the Manhattan map. Node p was at the point (80,30). Through comparison with the real distribution, we can see that there are some errors around the points (30,30) and (30,80), while the density is well-represented in the area around node p . In the freespace map (Fig. 7(e) and Fig. 7(f)), node p was at the point (20,20). We can see the same characteristics with the case of the Manhattan map. Finally, Fig. 7(g) and Fig. 7(h) show the result in the wire-Manhattan map. The position of node p was (600,400). We can see that the estimated density is expanded out of roads though the shape of roads can be recognized from the density map. This is because the size of the region is larger compared to the other maps and hence the diffuse estimation function is applied to most of the region.

From these results, we confirmed that the accuracy of density estimation was higher around the node's location and the estimated density maps could represent the shape of the real density distribution.

Similarity of Density Distribution To see the similarity between the estimated density distribution and the real density distribution, we used Kendall's

Table 4. Comparison of Communication Overhead

	cross-road	Manhattan	free-space	wire-Manhattan
Avg. bandwidth per node (kbps)	40	40	40	40
Avg. bandwidth with reduction per node (kbps)	11.28	16.16	20	10

Table 5. Effect of Communication Overhead Reduction on Kendall’s τ

	reduction	no reduction
cross-road	0.718	0.722
Manhattan	0.674	0.738
free-space	0.635	0.689
wire-Manhattan	0.670	0.673

τ [15]. Here, we introduce the concept of *granularity* to compare the two distributions. The granularity is represented by $g \times g$, which means that $g \times g$ cells are considered as one larger cell in computing the Kendall’s τ . We have changed this granularity from 1×1 to 10×10 . The results are shown in Table 3. The average Kendall’s τ is increasing as the granularity becomes larger in most cases. This is natural because the values of density are often expanded to wider regions (*i.e.* the outside) by the diffuse estimation function.

Also, Fig. 8 shows changes of Kendall’s τ over time. At the beginning, Kendall’s τ increases drastically because each node receives new density information from others. The Kendall’s τ becomes stable in the steady phase after that. For the maps except for the free-space, the averages of Kendall’s τ ranged from 0.67 to 0.718 and we can see the strong similarity. Even in the case of the free-space where it is difficult to predict the movements of nodes, the average Kendall’s τ was 0.635. Therefore, the proposed technique represents the real node distribution well.

4.3 Reduction in Communication Overhead

Effect of Reduction in Communication Overhead In our technique, the target region is divided into cells. The number of cells is 2500 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 sub-density map as we mentioned in Sec.3.3.

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.

Reduction in Communication Overhead vs. Accuracy From the results shown in Table 5, we see Kendall’s τ is lower when the communication overhead

is reduced. Obviously, there is a trade-off between communication overhead and the accuracy of density maps. Therefore, it is important to determine parameters on communication appropriately.

5 Discussions

The proposed method uses a cell matrix to represent a density map. The cell matrix facilitates computation like merging and 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 [6, 16]. Some other possibilities are using 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 data size.

We also discuss another important issue on 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 GPS or other measurements such as position estimation methods like Sextant [17] and UPL [12] due to their likelihood estimation in range-free localization, and (ii) privacy protection in which intentionally randomized positions obscures the true position.

6 Conclusion

In this paper, we have proposed a method for mobile nodes to self-estimate density in its proximity in real-time using ad-hoc wireless communications among these nodes. We have conducted simulation experiments in which correlation between the estimated map and the real density map had been measured.

The road traffic information can be collected and distributed through infrastructures like VICS, by which the covered regions are restricted to major highways and streets which are measured by base stations. On the other hand, the proposed method can utilize each vehicle's density map to build the map of wider areas, which cannot be covered by infrastructures only. This idea can also be applied to probe cars which collect information like traffic and weather. Also such density maps can be utilized by vehicles themselves for intelligent car navigation or other purposes. Consequently, the proposed method fit into many ITS applications.

Another 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 safe navigation toward exits.

Assuming these potential application examples, we are planning to conduct simulations in more realistic environments and to determine appropriate parameter settings and to validate usefulness of the method. Furthermore, autonomy of the protocol is our important goal where protocol parameters like message

transmission intervals can be autonomously converged into appropriate values depending on its neighborhood density for zero-configuration.

References

1. Ministry of Land, Infrastructure, T., of Japan, T.: Advanced safety vehicles (ASV) project <http://www.mlit.go.jp/jidosha/anzen/01asv/>.
2. Korkmaz, G., Ekici, E., Özgüner, F., Özgüner, U.: Urban multi-hop broadcast protocol for inter-vehicle communication systems. In: Proc. of ACM VANET. (2004) 76–85
3. Lochert, C., Scheuermann, B., Mauve, M.: Probabilistic aggregation for data dissemination in VANETs. In: Proc. of ACM VANET. (2007) 1–8
4. Yu, B., Gong, J., Xu, C.: Catch-up: a data aggregation scheme for VANETs. In: Proc. of ACM VANET. (2008) 49–57
5. Zhao, J., Cao, G.: Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology **57**(3) (2008) 1910–1922
6. Gupta, I., Renesse, R., Birman, K.: Scalable fault-tolerant aggregation in large process groups. In: Proc. of IEEE Int. Conf. on Dependable Systems and Networks (DSN). (2001) 433–442
7. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: Tag: a tiny aggregation service for ad-hoc sensor networks. SIGOPS Oper. Syst. Rev. **36**(SI) (2002) 131–146
8. Boulis, A., Ganeriwal, S., Srivastava, M.: Aggregation in sensor networks: an energy accuracy trade-off. Ad Hoc Networks **1**(2-3) (2003) 317–331
9. Papadopouli, M., Schulzrinne, H.: Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In: Proc. of ACM MobiHoc. (2001) 117–127
10. Goldenberg, D., Bihler, P., Cao, M., Fang, J., Anderson, B., Morse, A., Yang, Y.: Localization in sparse networks using sweeps. In: Proc. of MobiCom. (2006) 110–121
11. Li, M., Liu, Y.: Rendered path: range-free localization in anisotropic sensor networks with holes. In: Proc. of MobiCom. (2007) 51–62
12. Uchiyama, A., Fujii, S., Maeda, K., Umedu, T., Yamaguchi, H., Higashino, T.: Ad-hoc localization in urban district. In: Proc. of IEEE INFOCOM. (2007) 2306–2310
13. He, T., Huang, C., Blum, B., Stankovic, J., Abdelzaher, T.: Range-free localization schemes for large scale sensor networks. In: Proc. of MobiCom. (2003) 81–95
14. : Mobireal simulator <http://www.mobireal.net/>.
15. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical recipes in C: the art of scientific computing. second edn. Cambridge Univ. Press (1992)
16. Xu, Y., Lee, W.C., Mitchell, G.: CME: a contour mapping engine in wireless sensor networks. In: Proc. of IEEE ICDCS. (2008) 133–140
17. Guha, S., Murty, R., Sifer, E.: Sextant: a unified node and event localization framework using non-convex constraints. In: Proc. of ACM MobiHoc. (2005) 205–216