UPL: Opportunistic Localization in Urban Districts

Akira Uchiyama, *Member*, *IEEE*, Sae Fujii, Kumiko Maeda, Takaaki Umedu, *Member*, *IEEE*, Hirozumi Yamaguchi, *Member*, *IEEE*, and Teruo Higashino, *Senior Member*, *IEEE*

Abstract—We propose an opportunistic ad hoc localization algorithm called Urban Pedestrians Localization (UPL), for estimating locations of mobile nodes in urban districts. The design principles of UPL are twofold. First, we assume that location landmarks are deployed sparsely due to deployment-cost constraints. Thus, most mobile nodes cannot expect to meet these location landmarks frequently. Each mobile node in UPL relies on location information received from its neighboring mobile nodes instead in order to estimate its area of presence in which the node is expected to exist. Although the area of presence of each mobile node becomes inexact as it moves, it can be used to reduce the areas of presence of the others. Second, we employ information about obstacles such as walls, and present an algorithm to calculate the movable areas of mobile nodes considering obstacles for predicting the area of presence of mobile nodes accurately under mobility. This also helps to reduce each node's area of presence. The experimental results have shown that UPL could be limited to $0.7r$ positioning error in average, where r denotes the radio range by the above two ideas.

Index Terms—Mobile ad hoc networks, range-free localization, urban pedestrians

1 Introduction

Location information provides people in cities with highly personalized and reliable services such as route navigation, location-dependent advertisements and localized communication. Since GPS does not often work in urban areas, WiFi and/or cellular base stations are also used for localization [1], [2], [3], [4], [5], [6], [7], [8], of which transmission range is from 100 m to kilometers. However, receiving signals from landmarks over a distance requires clear Line-Of-Sight (LOS), which is sometimes hard to obtain in urban areas with tall buildings and underground cities. On the other hand, some other techniques assume a large number of landmarks with short transmission range in order to cover the target fields by these landmarks. Another alternative is to introduce mobile landmarks that enhance the coverage of regions [9]. This works fine if these mobile landmarks well cover the target field. However, it is also hard for ordinary (nonlandmark) nodes to find those mobile landmarks.

Some techniques exploit indirect information from landmarks. Techniques categorized into "collaborative multi-lateration" such as [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] assume that landmarks' position information is delivered in a multihop manner, and the distance to the landmarks is approximated by information on wireless ad hoc networks such as the number of hops. The other techniques categorized into "iterative multi-lateration" such as [21], [22], [23], [24], [25] use the estimated position of each node to estimate positions of its neighbors and run this estimation process iteratively. However, these techniques may not work well if nodes are mobile and networks are frequently partitioned, which is true in urban areas.

To overcome this problem, in this paper, we present an opportunistic ad hoc localization algorithm called Urban Pedestrians Localization (UPL) designed for positioning mobile terminals in urban areas. The key idea of UPL is the following. In urban areas, we do *not* assume that mobile nodes hear signals from location landmarks frequently and that networks are fully-connected. Thus, each mobile node in UPL maintains its own area of presence, and updates the area whenever it encounters other nodes and receives information about the areas of presence of those nodes. Localization is performed by intersecting its own area of presence with the received ones considering the radio range. This "opportunistic" localization does not require well-connected ad hoc networks. On the other hand, due to movement of mobile nodes, the area of presence expands as time passes. Let us consider an example in free space. At time t, we assume that a node's area of presence is represented as a circle of radius u centered at point c. If the maximum speed of the node is v_{max}, then the area of presence of the node at time $t + \Delta t$ can be estimated as the circle of radius $u + v_{max} \cdot \Delta t$ centered at c. This means that the area of presence expands quadratically with respect to v_{max} at each time unit. However, actually in urban areas, mobile nodes do not move in free space but in space restricted by walls and streets. Considering this fact, expanding the area of presence in such a way is overcautious. Thus, our algorithm utilizes obstacle information, and precisely determines the movable areas of mobile nodes considering the obstacles. To do so, each node only needs to know an obstacle map of its neighborhood. Unlike car navigation systems, such a map is simple and lightweight to distribute.

• *The authors are with the Mobile Computing Laboratory, Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 5650871, Japan. E-mail: {utiyama, s-fujii, umedu, h-yamagu, higashino}@ist.osaka-u.ac.jp, kumaeda@jp.ibm.com.*

Manuscript received 18 Oct. 2011; revised 12 Feb. 2012; accepted 24 Feb. 2012; published online 28 Mar. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2011-10-0577. Digital Object Identifier no. 10.1109/TMC.2012.86.

The main contributions of this work are as follows: Considering urban environments where pedestrians pass each other on pathways between buildings and in underground cities, we have designed an "opportunistic" localization algorithm called UPL. As far as we know, this is the first range-free ad hoc localization that is tailored to such environments. We note that the algorithm should be compact so as to be implemented for small devices such as smartphones. To this goal, we provide an efficient way of estimating mobility on a given map. To examine its performance, extensive simulations, real experiments using sensor nodes, and comparison with the tracking result of iPhone 3GS have been conducted. From the simulation results assuming a real region of Osaka downtown, UPL could achieve smaller position errors (7 m in average with 10 m of radio range) than the two existing localization methods by using simple devices only. Theoretical analyses are also provided to show its property.

In our previous work [26], we have presented a new concept of localization which well fits for positioning mobile nodes in urban districts, and proposed an efficient localization algorithm. In this paper, we have newly investigated the characteristics of UPL by additional simulations and real experiments, and have given theoretical analyses on the computation cost and estimation accuracy.

2 Related Work

2.1 Sensor Localization

For sensor localization, many efforts have been dedicated so far to investigate localization algorithms [27]. *Range-based* positioning techniques rely on physical information between nodes [28], [29], [30], [31], [32]. These range-based techniques are generally very accurate, e.g., positioning accuracy in the order of centimeters, although they incur cost for the dedicated hardware to measure distance.

On the other hand, range-based techniques using RSSI do not generally need additional hardware for localization. The study in [33] and [34] investigates to improve the quality of RSSI-based measurement. MDS-MAP is proposed in [16], which uses shortest path information between nodes to estimate the distance, and uses "multidimensional scaling" to determine the positions. Zero-configuration localization for indoor environments using RSSI has been proposed in [35], which automatically adjusts the distance function of RSSI in order to achieve robustness against fluctuations in the radio signal. Goldenberg et al. [36] propose the Sweeps algorithm for localization in bilateration networks where the average degree is relatively low. Sequence-based localization is presented in [37], which utilizes the ranking of signal strength from multiple landmarks. In urban environments, there are many buildings and walls that often cause multipath effects. In addition, different types of wireless devices are used in different types of mobile terminals such as smartphones and laptops. For this reason, RSSI measurements are unreliable in urban districts.

In contrast to range-based techniques, *range-free* techniques are cost effective and reliable alternatives, since they only require communication function between nodes while the accuracy is generally lower than range-based techniques. This is because they only rely on information included in received messages such as connectivity. Thus, they are well-fit for urban ad hoc wireless networks. Savvides et al. [38] have presented the following categories of localization algorithms; hyperbolic trilateration (positioning by distances from three landmarks), triangulation (if angles are obtained via AOA), and maximum likelihood multilateration (taking the region where the target node most likely exists). Multilateration is further classified into the following three simple categories, called *atomic*, *collaborative*, and *iterative* multilateration. The atomic multilateration uses more than two landmarks to determine a position. In the collaborative multilateration, nodes exchange landmark information cooperatively to be used for estimation. The iterative multilateration uses estimated positions as new reference points and estimates the other positions iteratively. Since our proposed UPL exploits range-free multilateration, we survey similar techniques in more detail.

First, as the atomic multilateration techniques, Centroid [39] assumes landmarks with long transmission radius or a large number of landmarks so as to cover the target fields. Each node estimates its position by calculating the centroid of all the landmarks it hears. APIT [40] uses a set of triangles formed by landmarks which have long transmission radius, and determines the location by checking that a node is inside or outside of each triangle. MCL [9] takes a different approach where each node updates the set of its likelihood points whenever it hears a signal from a landmark. IMCL [41] applied prediction of moving directions based on MCL.

Second, as the cooperative multilateration techniques, Ad hoc Positioning System (APS) [10], [11] approximates the distance from an indirect landmark by several ways, by estimation of hop distance or the number of hops. Savarese and Rabaey [12] present a similar hop-based approach. Amorphous [13] is also range-free localization that utilizes hop counts from landmarks as well, where each node estimates its coordinates by finding coordinates that minimize the total squared error between the calculated and estimated distances. Yang et al. [14] propose HCRL which exploits hop count ratios to landmarks from each node and transmission power control to increase accuracy. Self-configurable positioning [15] takes a similar approach where a distance is approximated via hop-based analysis, but then landmarks exchange the information to determine their coordinates followed by local coordination determination performed by each node. Lim and Hou [17] present a method to map "proximity" information, e.g., hop count information, between nodes into geographic distance considering the characteristics of anisotropic networks. Wang and Xiao [18] present an algorithm for sensor positioning in concave areas based on distance bounds. Li and Liu [19] use a boundary detection protocol for sensor localization in concave areas. Another sensor localization technique in concave areas is presented in [20], which estimates reliable landmarks and utilizes them for position estimation. For anisotropic sensor networks, Xiao et al. [42] propose sensor localization which mitigates the effect of detour paths by relying on landmarks within a few hops. Highly scalable sensor localization is proposed in [43],

based on the Ricci flow method for distorting a curved Riemannian metric to a flat one. Wang et al. [44] identify turning nodes, that are on the corners of the shortest paths between nodes, for accurate distance estimation.

Finally, as the iterative multilateration, recursive position estimation in [21] presents a fundamental technique for iterative multilateration. Bischoff and Wattenhofer [22] present a theoretical limit of connectivity-based multihop positioning and provide an algorithm that outperforms hop-based algorithms. Guha et al. [23] present a range-free algorithm called Sextant, which uses connectivity information. Sextant represents a region of presence by Bézier curves to pursue the accuracy. Then, the region is updated by iterative multilateration. Furthermore, study in [24] improved MCL [9] by using estimated locations of neighbors in addition to location information from landmarks. A bounding-box method is used to reduce the computation time in order for finding valid samples of Monte Carlo-based algorithms in [25].

These methods will work fine in open space and/or in connected (i.e., not partitioned) networks that we cannot expect in urban districts.

2.2 Localization in Urban Districts

Localization based on Wi-Fi and/or cellular networks has been investigated for mobile nodes in urban districts. In Place Engine [1], signal strength from multiple Wi-Fi access points at each location (i.e., fingerprints) is used for localization of mobile terminals. The accuracy is on the order of 5 to 100 meters depending on the density of Wi-Fi access points. Similar services such as Google Latitude [3] using Skyhook technology [2] and Virtual Earth [4] by Microsoft have recently been launched. The performance of metropolitan-scale war walking and war driving is analyzed in [45].

Some others use signals from Global System for Mobile Communications (GSM) cellular towers. The study in [5] measures AOA from cellular towers. TOA is also used to measure geographic relationships (i.e., distance/angle) between a mobile node and cellular towers [6]. Laitinen et al. [7] present a GSM-based localization technique using fingerprinting. In order to achieve better accuracy, Place Lab [8] uses Wi-Fi and GSM signals, and reduces the median of the position errors to 15-20 meters in urban areas. Chiu and Chen [46] use a mixed-norm approach to consider obstruction of signals from cellular towers measured by TOA for mobile location estimation in urban areas. LOS and Non-Line-Of-Sight (NLOS) states of signals from base stations are switched based on a Markov process, and used for accurate distance estimation in [47]. LAH [48] exploits ad hoc communication between nodes and employs a linear regression model for predicting the node mobility from history of encountered landmarks. In the existing localization techniques for urban areas above, the error ranges from tens of meters to kilometers, which are not sufficient to provide various services for pedestrians in urban districts.

On the other hand, Dead Reckoning (DR) [49], [50] estimates a position based on the speed and heading of a mobile node. Many recent approaches use accelerometers, gyro sensors, and digital compasses that are available on off-the-shelf smartphones. SparseTrack [51] combines DR with sparse beacon nodes to overcome cumulative errors of DR for indoor pedestrian tracking. However, there still remain challenges to accurate DR. For example, the accuracy of DR changes depending on device placements such as on a hip, in a pocket, and in a bag. Steinhoff and Schiele [52] study the accuracy of DR for different types of trouser pockets.

TABLE 1
Notations

Symbol	Notation
r_{max}	maximum communication range
v_{max}	maximum speed of nodes
R_i^t	area of presence of node i at time t
$R_i^t \oplus r_{max}$	area of presence of node i at time t expanded by maximum communication range r_{max}

3 UPL Algorithm

3.1 Overview and Assumption

We assume that each node is equipped with a PAN device such as ZigBee and Bluetooth. Our algorithm does not depend on specific hardware; however, it is not realistic to assume that each portable device continues seeking neighbors which are several tens of meters away, especially in a dense crowd like in a downtown area, due to limitations of batteries and bandwidth. Thus, as a realistic hardware environment, we assume these PAN technologies. Note that it is also possible to set a high threshold on received signal strength from existing WiFi access points to detect that a node is in the proximity of those access points. We use the notation shown in Table 1. For simplicity of discussion, we assume the same maximum communication range r_{max} for all the nodes. We also assume the same maximum velocity v_{max} for all the nodes. Hereafter, we let R_i^t denote an estimated area of presence (or simply called an *area of presence*) of node i at time t. The area of presence of node i denotes the region in which node i is expected to exist.

Each node broadcasts hello messages with regular intervals to its neighbors. A hello message transmitted from node i at time t includes the area of presence $R_i^{t_i}$, where t_i is the time when the last localization was executed at node i, and time duration $t - t_i$ (denoted by Δt_i). At the beginning of localization, R_i is unknown and initialized by receiving known areas of presence of others (e.g., landmarks). Thus, $R_i^{t_i}$ denotes the most recently updated area of presence of node i, and time duration Δt_i indicates the elapsed time from t_i. Clock synchronization among mobile nodes is not required because Δt_i is calculated by a local timer of node i.

We also assume that each node has a common obstacle map M of a target region, which can be preinstalled or downloaded from base stations such as WiFi hotspots and landmark nodes. The data structures of areas of presence and M are given later in Section 3.2. In UPL, obstacles are defined as regions where pedestrians cannot always exist and the assumed radio signal is obstructed. Thus, thick and high walls of metal or concrete are obstacles in most cases but fences may be not. Note that the obstacle map does not need to represent all obstacles, that is, the more obstacles

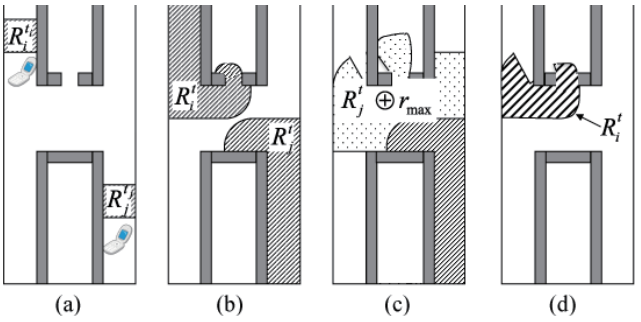


Fig. 1. Localization process of UPL.

are represented by the obstacle map, the more accurate the localization results are. This means UPL still works even if there is no information about obstacles. In order for automatically constructing the obstacle map, we can simply use image processing techniques if a digital map of the region is available. In the real experiments in Section 5, we extracted obstacle regions from Google Maps and Yahoo! Maps by simple image processing. Alternatively, we may use a map generation technique using both location information and ad hoc communication among nodes presented in [53].

When node i receives a hello message from node j at time t, node i immediately runs the UPL algorithm to update its area of presence. UPL is executed as follows: Node i calculates R_i^t and R_j^t from given $R_i^{t_i}$ and $R_j^{t_j}$ where t_i and t_j denote the time when the last UPL was executed at nodes i and j, respectively (Figs. 1a and 1b). Moreover, based on the calculated R_j^t where node j is expected to exist at time t, node i calculates the region in which node j's signal can be heard at time t (Fig. 1c). We calculate this region by expanding R_j^t by adding the maximum communication range r_{max} and it is denoted by $R_j^t \oplus r_{max}$. Finally, we obtain the new area of presence of node i at time t, by intersecting R_i^t and $R_j^t \oplus r_{max}$ as shown in Fig. 1d because node i must be located in both R_i^t and $R_j^t \oplus r_{max}$.

We say that R_i^t is *complete* if and only if R_i^t contains the position of node i at time t. Moreover, for two complete areas of presence R_i^t and $\hat{R}_i^t$ of node i at time t, if $|R_i^t| < |\hat{R}_i^t|$ then R_i^t is said to be more *accurate* than $\hat{R}_i^t$ where $|R_i^t|$ denotes the area of R_i^t. Our goal is to design an algorithm that can determine complete areas of presence of all mobile nodes which are as accurate as possible.

To take benefit from ad hoc localization by mobile nodes and obstacle information, we need to calculate the movement of nodes among obstacles. Thus, we carefully design the data structures and the algorithms which do not overload small hand-held devices with low computational capability. In the following sections, we present the data structure and UPL algorithm details.

The information about areas of presence can be used for many services. For example, if service providers receive areas of presence of shopping customers, they may send the information about shops and restaurants in that area. In such a case, to avoid sending useless information for users, precise identification of the area of presence is an important issue. For services which need to identify the location of nodes such as navigation systems, we also present how to determine a certain point from a given area of presence in Section 3.4.

3.2 Data Structure

Several data structures have been introduced and used to represent an area of presence in the existing localization algorithms. In [23], several data structures are given and discussed in detail. In a simple way, we may use circles or simple polygons such as rectangles to approximate areas of presence, but obviously it lacks the accuracy of approximation. Monte Carlo-based localization such as MCL [9] uses a set of randomly selected points to represent an area of presence, where simplicity is a merit for small handheld devices. Sextant [23] utilizes a list of representative points and the area is approximated as a set of Bézier curves between those points.

Unlike these methods, we divide a target region into small cells and represent areas of presence and obstacles by sets of cells. Here, the form of each area of presence in UPL is more complex than those in the existing methods because we consider regions restricted by obstacles and the movement of nodes in such complex regions. There are some methods that deal with obstacles. For example, Sextant [23] takes into account an area restricted by obstacles. However, it only deals with stationary nodes. Considering the fact that in our case the operations on areas of presence are more complex, we use grid cells to represent areas of presence and obstacle maps.

3.3 UPL Algorithm Details

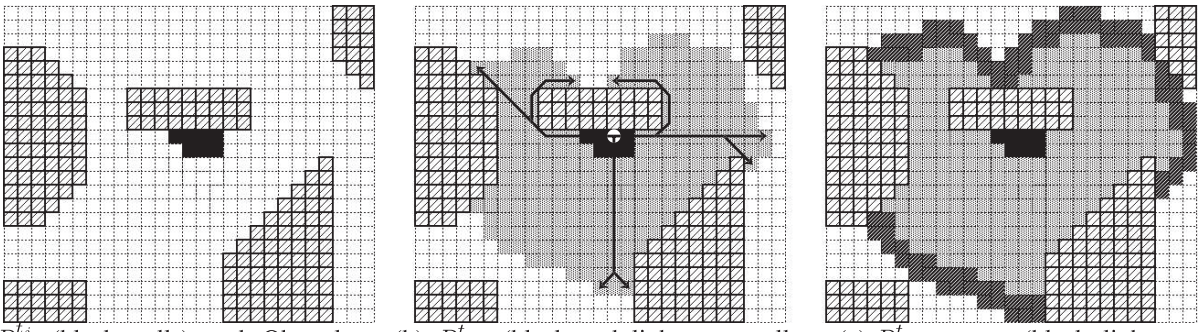
3.3.1 Computing Area of Presence

We present an algorithm to compute R_j^t, an area of presence of node j at time t, from given $R_j^{t_j}$ ($t_j < t$), $\Delta t_j = t - t_j$ and an obstacle map M. R_j^t and M are represented by sets of cells. This algorithm is referred to as the Area of Presence Computation (APC) algorithm and does not require complex functions, for example, like computing intersections of polygons. Instead, we only use simple and lightweight operations.

First, we represent the map M by set FS, the set of all the cells in the movable space (free space) of M. That is, FS is the set of all the cells which are not included by the obstacles. Basically, to calculate R_j^t, the APC algorithm adds the cells in FS within $v_{max} \cdot \Delta t_j$ distance from $R_j^{t_j}$. However, it may be expensive under the existence of obstacles to calculate the shortest distance from $R_j^{t_j}$ for each cell in FS. We therefore design an algorithm that can obtain R_j^t by expanding $R_j^{t_j}$ by simple computation. This is accomplished by approximating distance calculation. The detailed process is described below.

We introduce some terminologies and notations. For cell g, a cell which shares a side with g is called a *side cell*, and a cell which is not a side cell of g and shares a single vertex with g is called a *diagonal cell*. The side cells and diagonal cells are called *neighboring cells*. Also, for a set G of cells, a cell in G which has at least one neighboring cell outside G is called a *border cell*. For each cell g, we let $d(g)$ denote the (approximated) shortest distance from $R_j^{t_j}$ ($d(g) = 0$ if g is in $R_j^{t_j}$). Here, the distance of two neighboring cells is the Euclid distance between the centers of the cells.

The APC algorithm is described as follows: We start with $R_j^t \leftarrow R_j^{t_j}$ and iterate the following procedure. For each border cell g of R_j^t with the shortest distance $d(g)$, we add



(a) $R_j^{t_j}$ (black cells) and Obstacles (lightly shaded cells with solid lines) (cell size $w = 1$)

(b) R_j^t (black and light gray cells; $v_{max}\Delta t_j = 10$)); A white circle shows an estimated position.

(c) $R_j^t \oplus r_{max}$ (black, light gray and gray cells; $r_{max} = 2$)

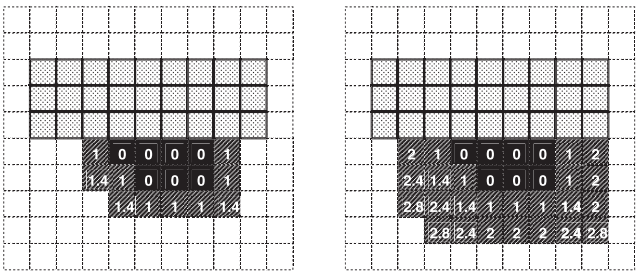
Fig. 2. Computing an area of presence and its expansion by maximum communication range.

each side cell g' of g in FS to R_j^t if $d(g) + w \leq v_{max} \cdot \Delta t_j$ where w is the side length of a cell. If g' is added, we set the shortest distance of g' as $d(g') = d(g) + w$. Similarly, we add each diagonal cell g'' of g in FS to R_j^t if $d(g'') + \sqrt{2}w \leq v_{max} \cdot \Delta t_j$. If g'' is added, we set the shortest distance of g'' as $d(g'') = d(g) + \sqrt{2}w$. Then, we remove all g' and g'' which have been added to R_j^t from FS. This procedure is repeated until there are no new cells in FS which can be added to R_j^t. Figs. 2a and 2b show examples of $R_j^{t_j}$ and R_j^t, respectively. Also in Fig. 3, we illustrate how $R_j^{t_j}$ expands at each step of the algorithm where $\sqrt{2} \approx 1.4$. The formal description of this APC algorithm is given in Fig. 4.

3.3.2 Expanding Area of Presence by Communication Range

Then, we compute $R_j^t \oplus r_{max}$, the area of presence of node j at time t expanded by the maximum communication range r_{max}. We call this area a *communicable area* of j at time t hereafter. There is a possibility to receive a hello message from j at time t in this communicable area under the assumptions that the maximum velocity of nodes is v_{max} and the maximum communication range is r_{max}.

For computing communicable areas, we may apply the APC algorithm. In actual fact, radio propagation may be affected by multipath effects in urban districts where many buildings exist. Ideally, we should design a propagation prediction algorithm considering the multipath effects by using precise radio propagation models such as Rayleigh fading and Rician fading [54]. However, in our environment, it is not reasonable to implement such a complex algorithm considering the computation cost. Furthermore, even though we apply such an NLOS propagation model, its impact on the performance is expected to be small because UPL expands an area of presence by the communication range from every point in the area of presence. For example, Fig. 5a shows a real position in an area of presence and LOS and possible NLOS areas from the position. Note that the effect of reflection is negligible in such places like pathways in underground cities and streets among buildings because the radio range is too short (e.g., 10 m) for reflection. To see the effect of diffraction, we calculated RSS at five positions shown in the figure by the knife-edge diffraction model [55]. The transmission power was set to -18 dBm, and the RSS threshold of a receiver was set to -78 dBm, which was the same value as the expected RSS at 10 m distance using the free space path loss model [56]. The calculated RSS values are shown in Table 2. We can see all the RSS values are lower than the threshold. Furthermore, as shown in Fig. 5b, LOS from every point in the area of presence includes most of possible NLOS points. For this reason, the difference of LOS

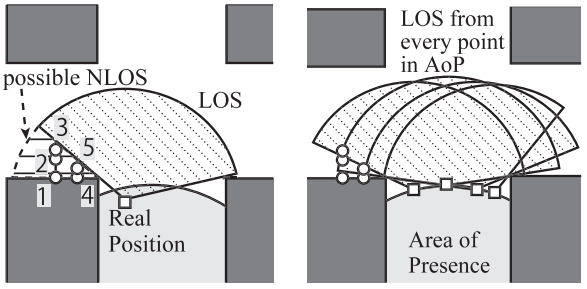


(a) After the 1st Step (b) After the 2nd Step

Fig. 3. First and second steps of the APC algorithm (value inside a cell represents distance from $R_j^{t_j}$).

```
1  R_j^t ← R_j^{t_j}; B_0 ←[set of border cells of R_j^{t_j}];
2  w ←[side length of a cell];
3  FS ←[set of cells in free space except R_j^{t_j}]; h ← 0;
4  do {
5     B_{h+1} ← ∅;
6     for each g ∈ B_h {
7        for each [side cell g' of g in FS] {
8           if (d(g) + w ≤ v_max · Δt_j) then {
9              B_{h+1} ← B_{h+1} ∪ {g'}; d(g') ← d(g) + w;
10             FS ← FS − g';
11          }
12       }
13       for each [diagonal cell g'' of g in FS] {
14          if (d(g) + √2 w ≤ v_max · Δt_j) then {
15             B_{h+1} ← B_{h+1} ∪ {g''}; d(g'') ← d(g) + √2 w;
16             FS ← FS − g'';
17          }
18       }
19    }
20    R_j^t ← R_j^t ∪ B_{h+1}; h ← h + 1;
21 } while (B_h ≠ ∅)
```

Fig. 4. Area of presence computation algorithm.



(a) Possible NLOS (b) LOS from AoP

Fig. 5. LOS from every point in AoP covers NLOS.

and NLOS models does not have so much impact on the performance. Hence, we adopt a LOS propagation model and simply use the APC algorithm to expand R_j^t (see Fig. 2c).

3.3.3 Intersecting Two Areas

Intersecting R_i^t and $R_j^t \oplus r_{max}$ is simple. We seek cells which are included in the both areas and obtain the new area of presence, $R_i^t \cap \{R_j^t \oplus r_{max}\}$.

3.4 Selecting a Point from Estimated Area of Presence

From the definition of an area of presence, a node exists at a point in its area of presence. The probability that a point in node's area of presence is its actual position is uniform and equal to the inverse of the size of the area of presence.

Many applications require a point as an estimation result. Here, in case that we need to identify the location of a node from its area of presence, we give a position estimation function that determines the point in the area of presence, which minimizes the upper bound of distance errors. We have used the following estimation function to determine the point p from an area of presence R and an obstacle map M

$$\text{select } p \in R \text{ that minimizes } \max_{p' \in R} dist(p, p'), \qquad (1)$$

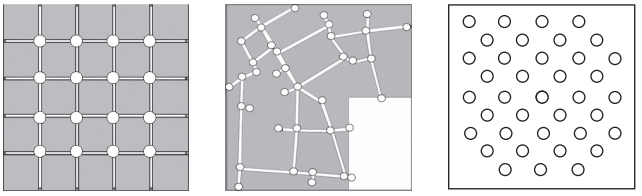
where $dist(p, p')$ is the shortest distance between p and p' on M (that is, the shortest distance among obstacles). An example of such a point is illustrated as a white circle in Fig. 2b. Considering the fact that the actual point should exist within R, selecting such p that minimizes the maximum distance between p and another point p' in R minimizes the maximum distance error of positioning.

3.5 Other Optimization

The existing techniques have utilized some other information to assist more accurate localization. Some techniques maintain and use the history of signal receptions from a landmark. Concretely, by knowing the timing when the node entered or left the communication range of a landmark, we may be able to identify the position of the node exactly on the disc edge of the communication range of the landmark. Also some try to extend the communication range of a landmark by multihop propagation of the landmark information. In our experiments, we have borrowed a sophisticated technique from MCL [9], which uses a two-hops advertisement of landmark positions to extend the coverage of landmarks.

TABLE 2
RSS Calculated by a Knife-Edge Diffraction Model

Position No. in Fig. 5	1	2	3	4	5
RSS (dBm)	-97.43	-91.32	-86.28	-93	-87



(a) Manhattan (b) Osaka (c) Free space

Fig. 6. Simulation maps (snapshot from MobiREAL).

4 Performance Evaluation

4.1 Simulation Settings

To confirm the effectiveness of UPL, we have conducted simulations using our open source mobile network simulator MobiREAL [57], [58]. We have used three maps: 1) *Manhattan region* [59] of $500\ \mathrm{m} \times 500\ \mathrm{m}$ with eight streets (Fig. 6a); 2) the real map of $500\ \mathrm{m} \times 500\ \mathrm{m}$ region in front of the Osaka train station (Fig. 6b) called *Osaka downtown;* and 3) *Free space* of $500\ \mathrm{m} \times 500\ \mathrm{m}$ without obstacles (Fig. 6c). In the figures, the radio range and deployment of landmarks are represented as circles. In Manhattan and Osaka downtown regions, we have specified areas except roads as obstacles, and LOS radio propagation was considered.

UPL requires the maximum velocity of nodes v_{max} as an input parameter and assumes moving speeds of nodes are no more than v_{max}. However, in practice, the speeds may be faster than v_{max}. Hence, it is important to determine v_{max} so that we can change its correctness, i.e., the number of nodes of which speeds are no more than v_{max}. For this purpose, the moving speeds v of nodes are determined so that they follow the normal distribution of (v_μ, v_σ^2) where $v > 0$. Based on these parameters, v_{max} is defined as

$$v_{max} = v_\mu + k v_\sigma,$$

where $k > 0$ and $k = 5$ in the default setting. From the characteristic of the normal distribution, approximately 99.999942 percent of nodes do not exceed v_{max} when $k = 5$. We used the above normal distribution for moving speeds in Manhattan and Osaka downtown regions. In Manhattan region, nodes are moved according to "random-street-decision" mobility where at each intersection each node decides to which direction it goes, except the backward direction. In Osaka downtown region, nodes' mobility is "random-street-decision" mobility except in the free space of $170\ \mathrm{m} \times 250\ \mathrm{m}$ at the bottom right corner where nodes move based on Random Waypoint mobility [60]. In Free space, nodes are moved according to Random Waypoint mobility where pause time is zero. As for Random Waypoint mobility, the minimum and maximum moving speeds are 1.0 and 2.0, respectively, and v_{max} is set to 2.0. In all the simulations, the default values described in Table 3

TABLE 3
Simulation Parameters

Region size	500m×500m
Cell size (w)	2 (m)
Radio range (r_{max})	10 (m)
# of Nodes	2,000
Speed distribution	Normal dist. of v_σ = 0.1, v_μ = 1.5
Hello message interval	2 (sec.)

are used except when we explicitly mention it. Also, we assume each node has an obstacle map from the beginning.

4.2 Impact of Parameters and Environments

We have conducted the simulations of the first type in Manhattan region. For the areas of presence, we have measured their accuracy, i.e., their sizes. The average sizes are derived from 10 simulation cases. The results are summarized in Table 4.

4.2.1 Cell Size

The cell size w may affect accuracy. We have evaluated the average sizes of areas of presence under different values of cell size w (1 m, 2 m, and 4 m). From the result, we can see explicit difference of the sizes. This is because smaller cells can represent the borders of areas of presence more accurately. In the case of $w = 4$, the average size is about $400\ \mathrm{m}^2$, which is rather large compared with the cases of $w = 2$ and $w = 1$. Also we do not see big difference between the cases of $w = 1$ and $w = 2$. Considering the fact that the cell size quadratically affects memory space, we may select $w = 2$.

4.2.2 Hello Message Interval

The frequency of hello message transmissions will affect the accuracy because receiving a "fresh" area of presence information is much more useful to reduce the area of presence for the receiver. In order for a node (say node i) to receive a hello message from a node (say node j) which is moving toward node i, the interval must not be greater than r_{max}/v_{max}. In this experiment, $r_{max} = 10$ m and $v_{max} = 2$ m/s; thus, the interval of 5 seconds is the maximum. We see the accuracy in different intervals (1, 2, 5, and 10 sec). From the result, the interval of 2 seconds seems the most reasonable interval because it achieved the similar size as the case of 1 second interval with the less number of messages.

4.2.3 Moving Speed

As moving speeds become large, there is a merit for each node that it may meet more nodes. However, quick expansion of areas of presence is a demerit. To see what happens if we increase the speeds of nodes, we have varied the average moving speed v_μ. The size of the area of presence increases linearly as the speed increases. Thus, slow speed is better for the accuracy. Since increasing speeds has similar effect with increasing hello message intervals, this is the result as expected. This result proves our localization is well fit for slow pedestrians.

TABLE 4
Impact of Parameters and Environments on Performance

Cell Size	1m×1m	2m×2m	4m×4m	
Area of Presence (m^2)	206.9	209.1	372.1	
Hello Interval (sec.)	1	2	5	10
Area of Presence (m^2)	170.0	209.1	358.2	514.4
Avg. Speed (m/s)	1.5	3.5	5.5	7.5
Area of Presence (m^2)	209.1	240.1	286.2	339.5
# of Landmarks	8	12	16	
Area of Presence (m^2)	464.8	326.3	209.1	

4.2.4 Number of Landmarks

One of the advantages of UPL is that it does not require many landmarks. To see what happens if we decrease the number of landmarks, we have varied the number of landmarks (8, 12, and 16). This is the natural result where the accuracy increases as the number of landmarks increases. Here, 16 landmarks with 10 m transmission range only cover 12.8 percent of the movable areas ($32 \times 10^3\ \mathrm{m}^2$). Even in such a region with sparsely deployed landmarks, UPL could achieve reasonable accuracy where $200\ \mathrm{m}^2$ indicates that we can identify a node on a street with 8 m width within 25 m length.

4.3 Effect of Opportunistic Localization and Precise Calculation of Movement

To see the effectiveness of our two key ideas: 1) opportunistic localization between mobile nodes, and 2) precise calculation of nodes' movement among obstacles, we also examine the performance of two simplified versions of UPL; UPL_{no_adhoc} which uses information from landmarks only and UPL_{no_obs} which does not utilize obstacle information to predict the movement of nodes and to calculate communication range. We have measured the average sizes of areas of presence in UPL_{no_adhoc}, UPL_{no_obs} and UPL, and calculated each ratio to original UPL in Manhattan region. We call this the *degeneracy ratio*.

Table 5 shows the averages of the degeneracy ratios. We can see that ad hoc localization and obstacles information effectively reduce 89 and 19 percent of the sizes of the areas of presence, respectively. Fig. 7 shows the size distributions of areas of presence. As we can see, obstacle information

TABLE 5
Degeneracy Ratio

	Degeneracy ratio
UPL_{no_adhoc}	8.961
UPL_{no_obs}	1.228
UPL	1.000

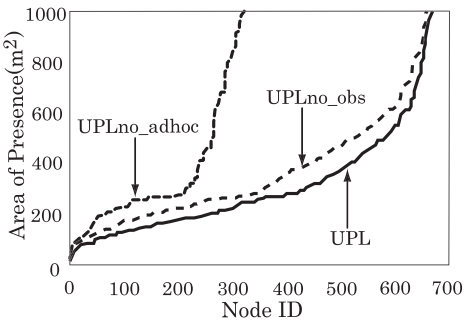
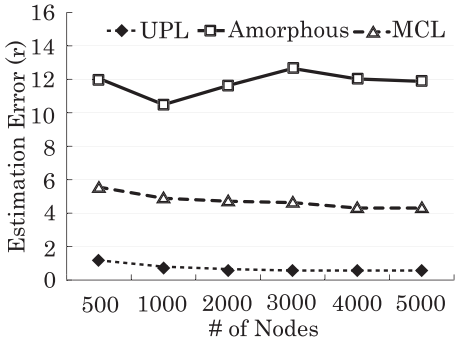
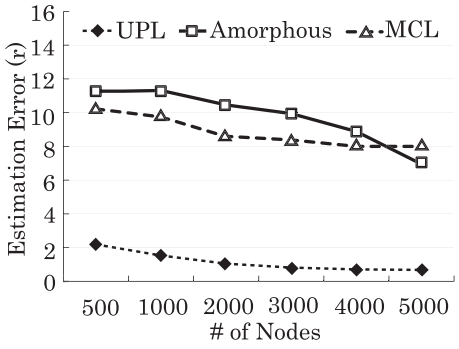


Fig. 7. Size distribution of areas of presence.



(a) Osaka downtown



(b) Free space

Fig. 8. Estimated position errors.

gives a certain amount of impact on accuracy of many nodes. On the other hand, opportunistic localization dramatically improves the accuracy of some nodes. This fact indicates that in some places which are away from landmarks, opportunistic localization is very helpful. From these results, we could confirm the effectiveness of two key ideas of UPL.

4.4 Comparison with Other Approaches

For comparison, we have measured the estimated position errors of UPL, MCL [9], and Amorphous [13] using Osaka downtown region and Free space. MCL only uses short-range landmarks in addition to the maximum velocity of nodes, and Amorphous performs multihop cooperative multilateration. We chose these two typical methods for comparison although many mobile localization algorithms have been presented. This is because we can see the difference between UPL and other localization algorithms using location information of other mobile nodes in addition to landmarks from the result of UPL_{no_obs} shown in Section 4.3. In UPL, we have used the position estimation function (1) of Section 3.4. Figs. 8a and 8b show the position errors regarding the communication range ($r = 10$ m). We see the accuracy of MCL and UPL in Osaka downtown is higher than that in Free space. This is because areas where nodes can move in Osaka downtown are smaller than in Free space, which means nodes receive more areas of presence in Osaka downtown.

First, the accuracy of Amorphous is expected to be higher as the node density increases, which is true in Free space (see Fig. 8b). However, it is not true in Osaka downtown where the shape of the network is restricted by obstacles. Surprisingly, Amorphous shows less accuracy compared with MCL and UPL, that limit the hello message propagation to two hops at most (see Fig. 8a). Second, in MCL, the more the node density, the higher the accuracy in both of two maps. MCL propagates landmarks' information to two hops, and the increase of the node means the wider spread of accurate information. In order to achieve the comparable accuracy to UPL, MCL requires more landmarks because MCL relies on direct (or two hops) information from landmarks. On the other hand, UPL outperforms these methods and the error is $0.7r$, that is, 7 m in Osaka downtown. Moreover, UPL achieves the higher accuracy than Amorphous even in Free space because UPL is opportunistic localization and sustainable against the low node density.

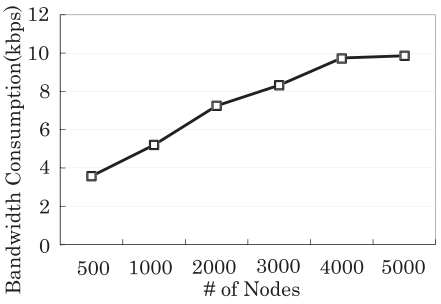


Fig. 9. Bandwidth consumption.

4.5 Communication Overhead

The number of messages sent in UPL is much less than flooding-based techniques such as Amorphous since each node in UPL periodically broadcasts its area of presence to neighbors. However, the size of each message is large because it contains grid cells' information. Hence, in order to see communication overhead, we changed the number of nodes and measured the bandwidth consumption by UPL in Osaka downtown. From the results shown in Fig. 9, we see that the bandwidth consumption becomes larger as the number of nodes increases. However, at the same time, the size of each hello message becomes smaller because the sizes of areas of presence decrease. From this result, the bandwidth consumption by UPL is small enough for the implementation based on Bluetooth, ZigBee, and so on.

4.6 Effect of v_{max}

UPL assumes the speeds of nodes are no more than v_{max}. In practice, the speeds of nodes may be faster than v_{max} however. Here, we evaluate UPL in such situations. We set v_μ and v_σ to 1.5 and 1.0, respectively. Fig. 10 shows the

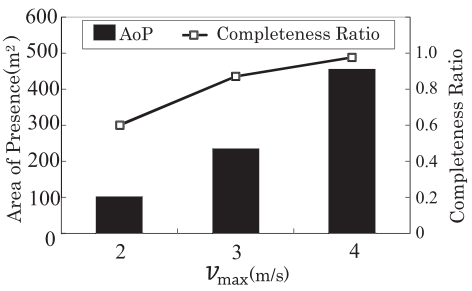
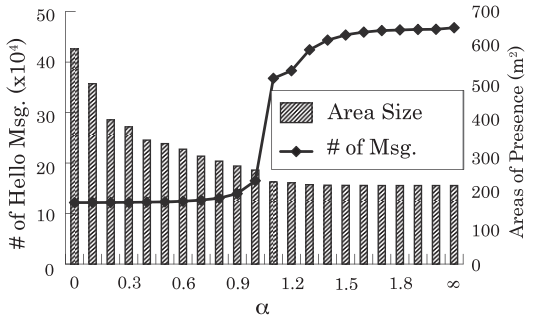
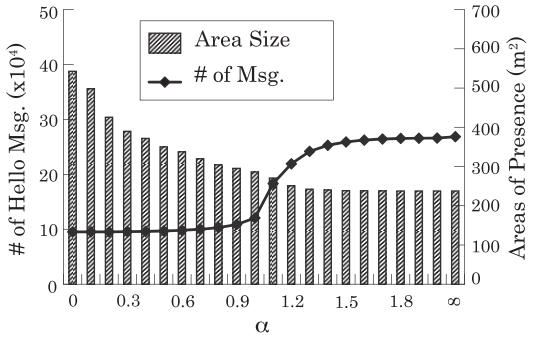


Fig. 10. Impact of v_{max} on performance.



(a) Hello Msg. Interval=1sec.



(b) Hello Msg. Interval=2sec.

Fig. 11. Effect of selective broadcast.

completeness and the accuracy where the parameter k for v_{max} is set to 0.5, 1.5, or 2.5. Here, the completeness ratio is defined as

$$\frac{\#\text{ of complete areas of presence}}{\#\text{ of calculated areas of presence}}.$$

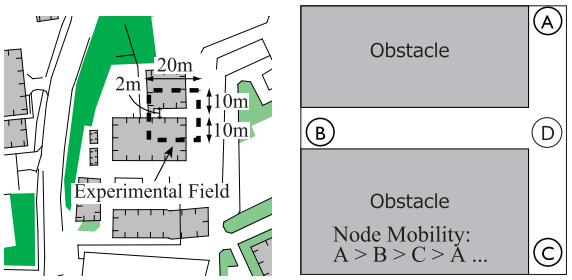
Theoretically, all areas of presence are complete. However, the approximation caused by APC algorithm and the data structure of grid cells make errors between theoretical areas of presence and computed ones. The computed areas of presence are therefore not always complete. In addition, areas of presence may become incomplete when the moving speeds of nodes exceed v_{max}.

In Fig. 10, the completeness and the size decrease as the number of nodes whose speeds are more than v_{max} increases. This is because the degeneracy of completeness leads to the occurrence of an "empty region" as a result of computing an intersection of two regions.[1] On the other hand, both the completeness and the size increased when v_{max} was set to 4 since most of nodes moved slower than v_{max}. This is a tradeoff. In order to increase the accuracy, we may set different v_{max} for each node and broadcast it with hello messages if moving speeds widely vary depending on nodes.

4.7 Selective Broadcast for Message Reduction

In urban districts, many areas of presence broadcast from neighboring nodes, may be similar in size and location due to high density. Such similar areas of presence cannot contribute to improve location accuracy of other nodes.

1. In all the simulations, we expand areas of presence gradually when an intersection of two regions is empty.



(a) Real Map (b) Obstacle Map

Fig. 12. Region used in real experiment.

Here, a method named selective broadcast is introduced for message reduction of similar areas of presence. Selective broadcast decides whether a node executes broadcasting of its area of presence or not by comparing the size $|R_i|$ of its own area of presence and the average size $|\overline{R_{recv}}|$ of areas of presence received in the last hello message interval. A node broadcasts a hello message if the following condition meets:

$$|R_i| < \alpha \times |\overline{R_{recv}}|, \tag{2}$$

where $\alpha \geq 0$ is a constant. Note that if the node has not received any hello message in the last interval, it broadcasts a hello message.

We simulated the effect of selective broadcast with different values of α in Manhattan map. We also changed the hello message interval to 1 or 2 seconds. From the result in Fig. 11, it is obvious that there is a tradeoff between the number of broadcasting and accuracy. In this result, $\alpha = 1.0$ achieves a good tradeoff, and reduces the number of hello messages by 50 to 80 percent approximately keeping accuracy compared to the case of $\alpha = \infty$ where selective broadcast is not employed. The effect of selective broadcast is large especially when the hello message interval is 1 second because areas of presence of neighbors are similar in most cases.

5 Real Experiment

5.1 Simulation Using Real Communication Log

5.1.1 Setting

To see the effect of radio propagation that is observed in the real world, we have conducted a real experiment using a small sensor called MICAz Mote [61], which supports ZigBee communication. In a $20\ \mathrm{m} \times 20\ \mathrm{m}$ square region with T-shaped roads, four Motes were deployed at intersections A, B, C, and D, and regarded as landmarks (see Fig. 12). The obstacle map was defined as shown in Fig. 12b based on the image from GoogleMap. In this region, three persons holding Motes started walking from the landmark A, B, and C to the adjacent landmarks, respectively, as shown in Fig. 12. Each Mote broadcast a hello message at the rate of 5 Kbps during the experiment. Specifically, the packet payload size is limited to 64 Bytes and 10 packets were broadcast every second. In the experiment, we regarded node i successfully received an area of presence from node j only when i received all the 10 packets from j. The transmission power was adjusted so as to the transmission range was approximately 2 m, and r_{max} was set to 4 m. The

TABLE 6
Real Experiment Result

	Real	Ideal
Completeness ratio	0.95	1.0
Avg. Area of Presence	$18.9m^2$	$14.7m^2$

moving speeds of nodes were 1.25 m/s and v_{max} was set to 2.0 m/s. In the above environment, three persons walked for 192 seconds and recorded the number of packets successfully received from other nodes. Based on the recorded logs, areas of presence were calculated by UPL. The cell size was set to $1\ \mathrm{m} \times 1\ \mathrm{m}$, and the cells covering the buildings were specified as obstacle cells in UPL. We have evaluated sizes of areas of presence and completeness ratios after updates of areas of presence. The results were compared with those in ideal radio propagation.

5.1.2 Result

The result is shown in Table 6. The completeness ratio in the ideal case is 1.0, and the average size of areas of presence is $14.7\ \mathrm{m}^2$, which is 22 percent smaller than that of the case of radio propagation that is observed in the real world. This is natural because nodes can transmit areas of presence information immediately when they enter the neighbors' radio range in the ideal radio propagation case. Such perfect sensing capability of transmission opportunity can keep the areas smaller, and accordingly those in any other cases will be larger than the ideal case, leading to less accuracy. Nevertheless, by the experiment in this section, we can prove that the accuracy of UPL with realistic propagation is still sufficient for street-level pedestrian tracking since an area of presence covers the road segment where the node exists in most cases.

5.2 Comparison with Smartphone Tracks and Other Methods in a Real Urban Underground City

5.2.1 Setting

To see the localization accuracy of commercial off-the-shelf devices and the effectiveness of UPL, we collected pedestrian tracks recorded by iPhone 3GS in a real urban underground city in Osaka (see Fig. 13a), and compared it with simulation results of UPL, Amorphous, and MCL. The area size is $246\ \mathrm{m} \times 358\ \mathrm{m}$, and GPS does not work in the area. Even without GPS signals, iPhone can find its location by using RSS database from WiFi base stations and cellular towers. We used the obstacle map shown in Fig. 13b where the obstacles were represented as shaded areas, and assumed five landmarks were deployed at the points of white circles in Fig. 13b. The obstacle map was easily obtained by applying image processing to the original map. The circles also represent their radio range. Note that we compensated the iPhone tracks so that the tracks did not cross over obstacles. For this purpose, if a line connecting two adjacent locations crosses over obstacle(s), we have replaced the line with the shortest path between the locations. To get stable WiFi signals for the iPhone tracks, we also paused at locations represented by small circles in Fig. 13a for about 60 seconds. Assuming UPL and MCL can detect whether the node is moving or not with an accelerometer, UPL and MCL set the maximum speed used in localization to zero during the pauses. The mobility was reproduced by Urban Pedestrian Flow (UPF) [62] based on the number of pedestrians counted at several locations. In the reproduced mobility, the average number of nodes was about 243. The other parameters were set to the values shown in Table 3.

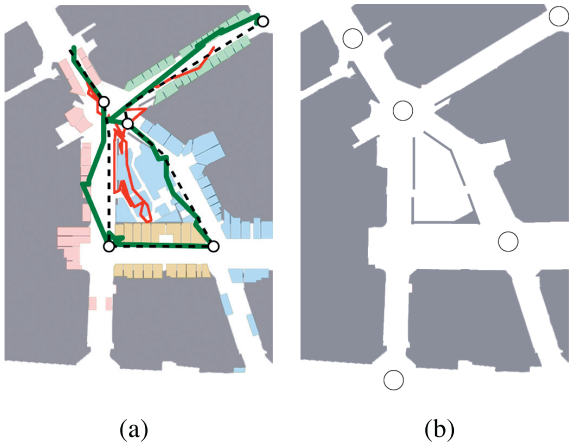


Fig. 13. Real underground city map and tracking results. (a) Underground city map and tracking results: ground truth (dashed); iphone (red thin); UPL (green bold). (b) Obstacle map and landmark coverage.

5.2.2 Result

The tracking results of UPL and iPhone are shown in Fig. 13a. We have drawn a line between a location and its adjacent location in the results. Overall, the average localization error of iPhone was about 50 m, and that of UPL was about 14 m. We can see the tracks of UPL are more similar to the ground truth than those of iPhone, and UPL correctly estimates the tracks in terms of streets in most cases. On the other hand, iPhone tracks sometimes jump to distant locations since the location estimation of iPhone relies on WiFi and cellular towers with low granularity. Moreover, the average error of Amorphous was 89 m because multihop networks were often disconnected due to mobility. Hence, distance estimation from more than two landmarks could not be processed in many cases. On the other hand, the average error of MCL was 72 m due to the small number of landmarks. From these results, we have confirmed UPL works well for pedestrians in urban areas.

6 Analysis

6.1 Space Complexity

MCL represents areas by a set of points; hence, the data size is proportional to the number of points. In Sextant, the data size is proportional to the number of control points since areas are represented by a set of Bézier curves. Thus, the space complexities of above two techniques are on the order of bytes.

On the other hand, UPL represents areas by grid cells. Here, we estimate the required memory space based on the parameter settings used in Section 4. Assuming the region is $500\ \mathrm{m} \times 500\ \mathrm{m}$ and the size of cells is $2\ \mathrm{m} \times 2\ \mathrm{m}$, the total

number of cells is 250×250. The data size for an obstacle map or an area of presence is approximately 8 Kbytes because 1 bit is enough for each cell to represent the cell is included in obstacles or areas of presence. Each node executes UPL whenever it receives an area of presence from others. The required memory space in UPL is three times of grid cells needed for representation of the whole region, that is, 24 kbytes. Additionally, APC algorithm needs to calculate the shortest distances between an area of presence and each grid cell. When the settings are $v_{max} = 2$ m/s, $r_{max} = 10$ m, and $\Delta t_{max} = 2$ sec, the expanded distance is 14 m at most. Thus, we need 1 byte per cell for the computation of the expansions. This means APC algorithm requires 62.5 Kbytes. Based on the above discussion, the memory space required in each node is about 90 Kbytes, which is large compared with MCL and Sextant. However, it is still acceptable on recent handheld terminals that have a memory on the order of megabytes.

6.2 Time Complexity

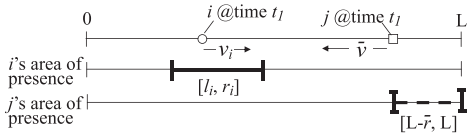
Calculating the shortest distance between any two points in a region with obstacles requires a visibility graph, of which the time complexity is $O(n^2 \log n)$ where n is the number of obstacle edges [63]. In a straightforward approach, expansion of an area of presence represented by a polygon with M vertices is computed as follows:

1. Compute the bounding box of the polygon.
2. Expand the bounding box by $v_{max} \cdot \Delta t$ without considering obstacles. The bounding box always includes the expanded area of presence.
3. Construct grid cells in the expanded bounding box.
4. For each cell in free space, check its reachability from the original area of presence, i.e., check if the distance from the cell to any one of the cells in the original area of presence is equal to $v_{max} \cdot \Delta t$ or less. For this purpose, we need calculate distance between two points in a region with obstacles. Time complexity of this calculation is $O(n^2 \log n)$ where n is the number of obstacle edges.
5. Build the expanded area of presence represented by a polygon from the cells that are reachable from the original area of presence.

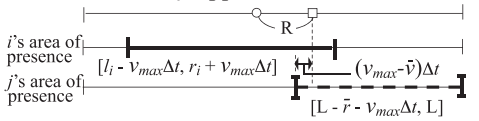
On the other hand, our APC algorithm does not require calculation of distance in a region with obstacles. Instead, the algorithm iterates calculation of distances to neighboring cells. The expansion of an area of presence can be computed in $O(v_{max} \cdot \Delta t / w)$ time in any environment, where w denotes the cell size. Hence, it is reasonable to choose grid cells as data structure, considering energy consumption for computation.

6.3 Accuracy Analysis

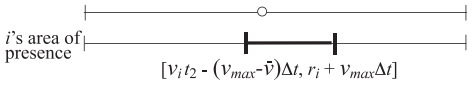
In order to simplify the analysis, we consider areas of presence on a 1D segment of which the length is L m as shown in Fig. 14a. We assume that two landmarks are placed at the locations 0 and L, and N nodes $(1, 2, \ldots, N)$ move on this segment $[0, L]$. In this environment, we focus on a certain node i and consider the change of i's area of presence represented as a segment in $[0, L]$. For simplicity of discussion, the moving speed and the radio range of all



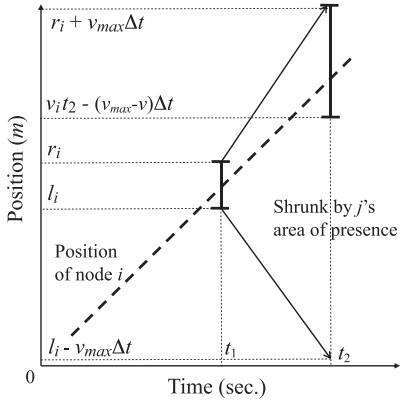
(a) i and j approach each other



(b) i receives j's area of presence at time t_2 $(\Delta t = t_2 - t_1)$



(c) i's area of presence is shrunk by j's area of presence



(d) Time vs. i's area of presence

Fig. 14. One-dimensional model for accuracy analysis.

nodes except node i are $\bar{v} (\leq v_{max})$ and $\bar{r} (\leq r_{max})$, respectively. Note that each node knows only v_{max} and r_{max}. Moreover, we assume the following. First, all nodes except for the focused node i only use information sent directly from landmarks. Hence, we do not consider exchange of areas of presence among nodes except i. Second, the focused node i receives an area of presence of another node j immediately when i comes into the radio range of j. We can ignore the hello message interval based on this assumption.

We consider that node i starts moving at time 0 with speed v_i and node j which moves from location L toward 0 at time t_0 with speed $\bar{v}$. We let $t_1 (> t_0)$ denote the time when j goes out of the radio range of the landmark. Then, the area of presence of j at time t_1 is represented by a segment $[L - \bar{r}, L]$. We denote i's area of presence at that time as a segment $[l_i, r_i]$. Hereafter, areas of presence are supposed to be within the segment $[0, L]$ (see Fig. 14a).

We define $t_2 (> t_1)$ as the time when i comes into the radio range of j. Here, we assume that i has not encountered any nodes since t_1. The area of presence of i at t_2 is represented by $[l_i - v_{max}\Delta t, r_i + v_{max}\Delta t]$ $(\Delta t = t_2 - t_1)$. The area of presence of j is represented by $[L - \bar{r} - v_{max}\Delta t, L]$ likewise (see Fig. 14b).

Then, i shrinks its area of presence by using j's area of presence. However, the "tail" of the j's segment (i.e., the segment end in the opposite direction of j's movement) is hardly useful because it is larger than the "head" of i's segment (i.e., the segment end in the direction of i's

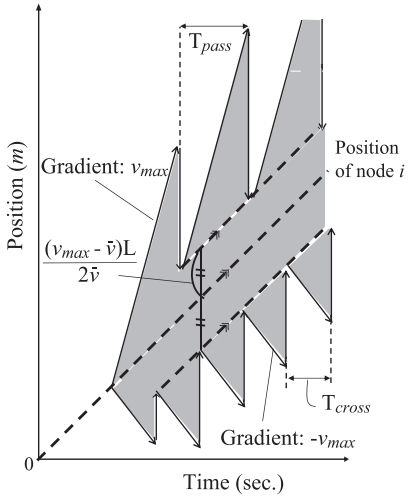


Fig. 15. Time versus node i's area of presence.

movement). On the other hand, j's head is effective for shrinking. j's head is expanded to $(v_{max} - \bar{v})\Delta t$ ahead from the current position of j, and the distance between i and j is $\bar{r}$. i's tail is cut off to $v_i t_2 - (v_{max} - \bar{v})\Delta t$ (see Fig. 14c). This example is shown in Fig. 14d as a graph. Similarly, i's head is cut off to $(v_{max} - \bar{v})t_2$ ahead from i's current position when j moves toward the same direction as i.

Next, we focus on the interval between encountering two adjacent nodes in each case of passing and crossing. We consider two adjacent nodes j and k that are moving toward the same direction, and k is following j. The interval is defined as the length of time it takes for i to enter the radio range of k after entering that of j. Assuming the uniform distribution of nodes, the distance between two adjacent nodes moving toward the same direction as i is

$$\frac{2L}{N+1}(m).$$

Thus, the interval of passing T_{pass} is

$$T_{pass} = \frac{2L}{(\bar{v} - v_i)(N+1)}(sec.) \quad (\text{if } \bar{v} > v_i).$$

Similarly, the interval of crossing T_{cross} is

$$T_{cross} = \frac{2L}{(\bar{v} + v_i)(N+1)}(sec.).$$

The extent of shrinking becomes smaller proportionally to the product of $(v_{max} - \bar{v})$ and the elapsed time since the encountered node received information from the last landmark. In this analysis, we approximate the extent of shrinking to that obtained at the location $L/2$. Therefore, when j passes/crosses i, the head/tail of i's area of presence is cut off

$$(v_{max} - \bar{v})\frac{L}{2\bar{v}} \tag{3}$$

away from i's current position. Consequently, i iterates expanding and shrinking its area of presence at the interval of T_{cross} and T_{pass} as shown in Fig. 15.

The average length $|M_i|$ of i's area of presence is equal to the average height of the shaded area in Fig. 15. We

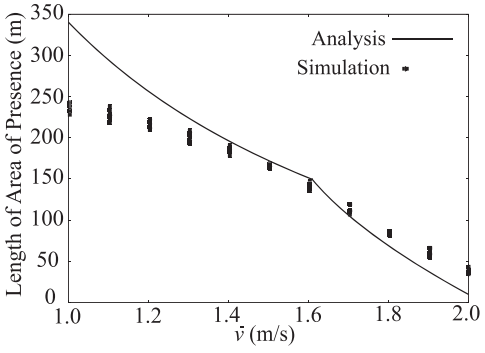


Fig. 16. Moving speed $\bar{v}$ versus length of areas of presence.

can calculate $|M_{upper}|$, the average length of i's head, as follows:

$$|M_{upper}| = \begin{cases} \dfrac{(v_{max} - \bar{v})L}{2\bar{v}} + \dfrac{(v_{max} - v_i)L}{(\bar{v} - v_i)(N+1)} \\ \quad \left(\text{if } \bar{v} > v_i \text{ and } |M_{upper}| \le \frac{(v_{max} - v_i)L}{2\bar{v}}\right) \\ \dfrac{(v_{max} - v_i)L}{2v_i} \qquad (\text{otherwise}). \end{cases}$$

Because of the approximation in the expression (3), $|M_{upper}|$ may become longer than that without passing. Hence, we limit the maximum of $|M_{upper}|$ in addition to the condition $\bar{v} > v_i$. The following denotes the average length $|M_{lower}|$ of i's tail:

$$|M_{lower}| = \frac{(v_{max} - \bar{v})L}{2\bar{v}} + \frac{(v_{max} + \bar{v})L}{(\bar{v} + v_i)(N+1)}. \tag{4}$$

We then obtain

$$|M_i| = |M_{upper}| + |M_{lower}|. \tag{5}$$

In order to confirm the correctness of the analysis above, we have compared $|M_i|$ in the analytic model and simulations. We set $L = 500\,\text{m}$, $v_{max} = 2\ \text{m/s}$, $v_i = 1.5\ \text{m/s}$, $r_{max} = 10\,\text{m}$, and $N = 100$, and measured $|M_i|$ by simulations. Both of the results in Fig. 16 show the similar values and trend, and we could confirm the correctness of the analysis. When $\bar{v}$ is less than v_i, the simulation result is smaller than the analytic result. The reason is the effect of ad hoc localization among other mobile nodes. In the simulation, an area of presence is shrunk by others' areas of presence although we assume that nodes except i rely only on the directly received information from landmarks in the analysis. This difference affects the size of the area of presence in the analysis remarkably as $\bar{v}$ is low because the analytic model expands the areas of presence considering the worst case, which results in the larger areas of presence.

The concept of the 1D analysis can be extended to a 2D case. Different from the 1D analysis, the 2D analysis requires considering encounter directions in 360 degree. For this purpose, we may be able to use the expected encounter time between different nodes provided in [64]. If the encounter intervals for each direction are given, then we can calculate the average lengths for each direction similarly to (4) and (5). Approximation of shapes of areas of presence (e.g., circles) may be necessary for the simplicity of the analysis.

7 Conclusion

In this paper, we have proposed an opportunistic localization algorithm called UPL for positioning mobile users in urban districts. UPL exploits obstacle information and location information from neighboring nodes. For this purpose, each mobile node maintains its area of presence where the node exists. To show the effectiveness of UPL, extensive simulations and real experiments have been conducted. From the results, we have shown that UPL could achieve reasonable accuracy for positioning of mobile users in urban cities, compared with typical existing methods.

Acknowledgments

This work was supported in part by Core Research for Evolutional Science and Technology (CREST, JST), the KDDI foundation, and the CPS-IIP Project (FY2012-FY2016) in the research promotion program for national level challenges by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

References

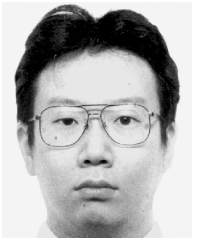
[1] *Place Engine,* http://www.placeengine.com/en, 2012.
[2] *Skyhook Wireless,* http://www.skyhookwireless.com/, 2012.
[3] *Google Latitude,* http://www.google.com/latitude, 2012.
[4] *Virtual Earth,* http://virtualearth.msn.com/, 2012.
[5] S.C. Swales, J.E. Maloney, and J.O. Stevenson, "Locating Mobile Phones and the US Wireless E-911 Mandate," *Proc. IEE Colloquium Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications,* pp. 2/1-2/6, 1999.
[6] L. Lopes, E. Villier, and B. Ludden, "GSM Standards Activity on Location," *Proc. IEE Colloquium Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications,* pp. 7/1-7/7, 1999.
[7] H. Laitinen, J. Lahteenmaki, and T. Nordstrom, "Database Correlation Method for GSM Location," *Proc. IEEE Vehicular Technology Conf.,* pp. 2504-2508, 2001.
[8] J. Hightower, A. LaMarca, and I.E. Smith, "Practical Lessons from Place Lab," *IEEE Pervasive Computing,* vol. 5, no. 3, pp. 32-39, July-Sept. 2006.
[9] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *Proc. ACM MobiCom,* pp. 45-57, 2004.
[10] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *Proc. IEEE GlobeCom,* pp. 2926-2931, 2001.
[11] D. Niculescu and B. Nath, "DV-Based Positioning in Ad Hoc Networks," *J. Telecomm. Systems,* vol. 22, nos. 1-4, pp. 267-280, 2003.
[12] C. Savarese and J. Rabaey, "Robust Positioning Algorithms for Distributed Ad Hoc Wireless Sensor Networks," *Proc. USENIX Ann. Technical Conf.,* pp. 317-328, 2002.
[13] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network," *Proc. Int'l Workshop Information Processing in Sensor Networks,* pp. 333-348, 2003.
[14] S. Yang, J. Yi, and H. Cha, "HCRL: A Hop-Count-Ratio Based Localization in Wireless Sensor Networks," *Proc. Ann. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* pp. 31-40, 2007.
[15] H. Wu, C. Wang, and N.-F. Tzeng, "Novel Self-Configurable Positioning Technique for Hop Wireless Networks," *IEEE/ACM Trans. Networking,* vol. 13, no. 3, pp. 609-621, June 2005.
[16] Y. Shang, W. Ruml, Y. Zhang, and M.P.J. Fromherz, "Localization from Mere Connectivity," *Proc. ACM MobiHoc,* pp. 201-212, 2003.
[17] H. Lim and J.C. Hou, "Localization for Anisotropic Sensor Networks," *Proc. IEEE INFOCOM,* 2005.
[18] C. Wang and L. Xiao, "Locating Sensors in Concave Areas," *Proc. IEEE INFOCOM,* 2006.
[19] M. Li and Y. Liu, "Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes," *Proc. ACM MobiCom,* pp. 51-62, 2007.
[20] B. Xiao, L. Chen, Q. Xiao, and M. Li, "Reliable Anchor-Based Sensor Localization in Irregular Areas," *IEEE Trans. Mobile Computing,* vol. 9, no. 1, pp. 60-72, Jan. 2010.
[21] J. Albowicz, A. Chen, and L. Zhang, "Recursive Position Estimation in Sensor Networks," *Proc. IEEE Int'l Conf. Network Protocols (ICNP),* pp. 35-41, 2001.
[22] R. Bischoff and R. Wattenhofer, "Analyzing Connectivity-Based Multihop Ad-Hoc Positioning," *Proc. Second IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom),* 2004.
[23] S. Guha, R. Murty, and E.G. Sirer, "Sextant: A Unified Node and Event Localization Framework Using Non-Convex Constraints," *Proc. ACM MobiHoc,* pp. 205-216, 2005.
[24] M. Rudafshani and S. Datta, "Localization in Wireless Sensor Networks," *Proc. Int'l Workshop Information Processing in Sensor Networks,* pp. 51-60, 2007.
[25] S. Zhang, J. Cao, L. Chen, and D. Chen, "Accurate and Energy-Efficient Range-Free Localization for Mobile Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 9, no. 6, pp. 897-910, June 2010.
[26] A. Uchiyama, S. Fujii, K. Maeda, T. Umedu, H. Yamaguchi, and T. Higashino, "Ad-Hoc Localization in Urban District," *Proc. IEEE INFOCOM,* pp. 2306-2310, 2007.
[27] C. Wang and L. Xiao, "Sensor Localization under Limited Measurement Capabilities," *IEEE Network,* vol. 21, no. 3, pp. 16-23, May/June 2007.
[28] D. Niculescu and B. Nath, "VOR Base Stations for Indoor 802.11 Positioning," *Proc. ACM MobiCom,* pp. 58-69, 2004.
[29] C.K. Seow and S.Y. Tan, "Non-Line-of-Sight Localization in Multipath Environments," *IEEE Trans. Mobile Computing,* vol. 7, no. 5, pp. 647-660, May 2008.
[30] O. Hernandez, F. Bouchereau, and D. Munoz, "Maximum Likelihood Position Estimation in Ad-Hoc Networks Using a Dead Reckoning Approach," *IEEE Trans. Wireless Comm.,* vol. 7, no. 5, pp. 1572-1584, May 2008.
[31] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. ACM MobiCom,* pp. 32-43, 2000.
[32] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha, "Tracking Moving Devices with the Cricket Location System," *Proc. ACM MobiSys,* pp. 190-202, 2004.
[33] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE INFOCOM,* pp. 775-784, 2000.
[34] N. Patwari and A.O. Hero, "Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks," *Proc. ACM Int'l Conf. Wireless Sensor Networks and Applications,* pp. 20-29, 2003.
[35] H. Lim, L.-C. Kung, J.C. Hou, and H. Luo, "Zero-Configuration, Robust Indoor Localization: Theory and Experimentation," *Proc. IEEE INFOCOM,* 2006.
[36] D.K. Goldenberg, P. Bihler, M. Cao, J. Fang, B.D.O. Anderson, A.S. Morse, and Y.R. Yang, "Localization in Sparse Networks Using Sweeps," *Proc. ACM MobiCom,* pp. 110-121, 2006.
[37] K. Yedavalli and B. Krishnamachari, "Sequence-Based Localization in Wireless Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 7, no. 1, pp. 81-94, Jan. 2008.
[38] A. Savvides, C.-C. Han, and M.B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. ACM MobiCom,* pp. 166-179, 2001.
[39] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm. Magazine,* vol. 7, no. 5, pp. 28-34, Oct. 2000.
[40] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc. ACM MobiCom,* pp. 81-95, 2003.
[41] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Distributed Localization Scheme for Mobile Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 9, no. 4, pp. 516-526, Apr. 2010.
[42] Q. Xiao, B. Xiao, J. Cao, and J. Wang, "Multihop Range-Free Localization in Anisotropic Wireless Sensor Networks: A Pattern-Driven Scheme," *IEEE Trans. Mobile Computing,* vol. 9, no. 11, pp. 1592-1607, Nov. 2010.
[43] M. Jin, S. Xia, H. Wu, and X. Gu, "Scalable and Fully Distributed Localization with Mere Connectivity," *Proc. IEEE INFOCOM,* pp. 3164-3172, 2011.
[44] Y. Wang, K. Li, and J. Wu, "Distance Estimation by Constructing the Virtual Ruler in Anisotropic Sensor Networks," *Proc. IEEE INFOCOM,* pp. 1-9, 2010.

[45] A.W. Tsui, W.-C. Lin, W.-J. Chen, P. Huang, and H.-H. Chu, "Accuracy Performance Analysis Between War Driving and War Walking in Metropolitan Wi-Fi Localization," *IEEE Trans. Mobile Computing,* vol. 9, no. 11, pp. 1551-1562, Nov. 2010.
[46] W.-Y. Chiu and B.-S. Chen, "A Mixed-Norm Approach Using Simulated Annealing with Changeable Neighborhood for Mobile Location Estimation," *IEEE Trans. Mobile Computing,* vol. 9, no. 5, pp. 633-642, May 2010.
[47] C.-Y. Yang, B.-S. Chen, and F.-K. Liao, "Mobile Location Estimation Using Fuzzy-Based IMM and Data Fusion," *IEEE Trans. Mobile Computing,* vol. 9, no. 10, pp. 1424-1436, Oct. 2010.
[48] J. Yi, J. Koo, and H. Cha, "A Localization Technique for Mobile Sensor Networks Using Archived Anchor Information," *Proc. Ann. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* pp. 64-72, 2008.
[49] A. Agarwal and S.R. Das, "Dead Reckoning in Mobile Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf.,* pp. 1838-1843, 2003.
[50] J. Koo, J. Yi, and H. Cha, "Localization in Mobile Ad Hoc Networks Using Cumulative Route Information," *Proc. 10th Int'l Conf. Ubiquitous Computing (UbiComp),* pp. 124-133, 2008.
[51] Y. Jin, M. Motani, W.-S. Soh, and J. Zhang, "SparseTrack: Enhancing Indoor Pedestrian Tracking with Sparse Infrastructure Support," *Proc. IEEE INFOCOM,* pp. 1-9, 2010.
[52] U. Steinhoff and B. Schiele, "Dead Reckoning from the Pocket - An Experimental Study," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom),* pp. 162-170, 2010.
[53] S. Minamimoto, S. Fujii, H. Yamaguchi, and T. Higashino, "Local Map Generation Using Position and Communication History of Mobile Nodes," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom),* pp. 2-10, 2010.
[54] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems. I. Characterization," *IEEE Comm. Magazine,* vol. 35, no. 7, pp. 90-100, July 1997.
[55] W.C. Lee, *Mobile Communication Engineering.* McGraw-Hill, 1982.
[56] J.D. Parsons, *The Mobile Radio Propagation Channel.* Wiley, 1992.
[57] *MobiREAL,* http://www.mobireal.net, 2012.
[58] K. Maeda, A. Uchiyama, T. Umedu, H. Yamaguchi, K. Yasumoto, and T. Higashino, "Urban Pedestrian Mobility for Mobile Wireless Network Simulation," *Ad Hoc Networks,* vol. 7, no. 1, pp. 153-170, 2009.
[59] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Comm. and Mobile Computing,* vol. 2, pp. 483-502, 2002.
[60] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. ACM MobiCom,* pp. 85-97, 1998.
[61] *MEMSIC Corporation,* http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html, 2012.
[62] K. Maeda, K. Sato, K. Konishi, A. Yamasaki, A. Uchiyama, H. Yamaguchi, K. Yasumoto, and T. Higashino, "Getting Urban Pedestrian Flow from Simple Observation: Realistic Mobility Generation in Wireless Network Simulation," *Proc. ACM/IEEE Int'l Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM),* pp. 151-158, 2005.
[63] M.D. Berg, O. Schwarzkopf, M.V. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications,* second ed. Springer-Verlag, 2000.
[64] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Performance Analysis of Mobility-Assisted Routing," *Proc. ACM MobiHoc,* pp. 49-60, 2006.

Akira Uchiyama received the ME and PhD degrees in information science and technology from Osaka University in 2005 and 2008, respectively, and is currently an assistant professor at Osaka University. He was a visiting scholar in the 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 ubiquitous computing. He is a member of the IEEE and Information Processing Society of Japan (IPSJ).

Sae Fujii received the ME and PhD degrees in information science and technology from Osaka University, Japan, in 2008 and 2011, respectively. Her current research interests include applications and services on mobile ad hoc networks. She is a member of Information Processing Society of Japan (IPSJ).

Kumiko Maeda received the ME and PhD degrees in information science and technology from Osaka University, Japan, in 2009 and 2006, respectively. She is currently with IBM Japan, Ltd. Her current research interests include wireless network simulation and high performance computing for heterogeneous systems.



Takaaki Umedu received the ME degree in informatics and mathematical sciences and the PhD degree in information science from Osaka University, Japan, in 2001 and 2004, respectively. He is currently an assistant professor at Osaka University. His current research interests include design and implementation of distributed systems and mobile systems. He is a member of the IEEE and the Information Processing Society of Japan (IPSJ).

Hirozumi Yamaguchi received the BE, ME, and PhD 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 the IEEE.

Teruo Higashino received the BS, MS, and PhD 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 the Graduate School of Information Science and Technology at Osaka University. His current research interests include the design and analysis of distributed systems, communication protocols, and mobile computing. He is a senior member of the IEEE and the IEEE Computer Society, a fellow of Information Processing Society of Japan (IPSJ), and a member of ACM and IEICE of Japan.