Ad-hoc Localization in Urban District

Akira Uchiyama, Sae Fujii, Kumiko Maeda, Takaaki Umedu, Hirozumi Yamaguchi and Teruo Higashino
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565–0871, JAPAN
Email: {utiyama, s-fujii, k-maeda, umedu, h-yamagu, higashino}@ist.osaka-u.ac.jp

***Abstract*— In this paper, we present a range-free ad-hoc localization algorithm called UPL (Urban Pedestrians Localization), for positioning mobile nodes in urban district. The design principle of UPL is two-fold. (1) We assume that location seeds are deployed sparsely due to deployment-cost constraints. Thus most mobile nodes cannot expect to meet these location seeds frequently. Therefore, each mobile node in UPL relies on location information received from its neighboring mobile nodes in order to estimate its area of presence. The area of presence of each mobile node becomes inexact as it moves, but it is helpful to reduce the areas of presence of the other mobile nodes. (2) To predict the area of presence of mobile nodes accurately under mobility, we employ information about obstacles such as walls, and present an algorithm to calculate the movable areas of mobile nodes considering obstacles. This also helps to reduce each node's area of presence. The experimental results have shown that by the above two ideas UPL could achieve 8m positioning error in average with 10m of radio range.**

I. Introduction

Location information is significant for future ubiquitous systems that provide people in cities with highly-personalized and reliable services such as route navigation, location-dependent advertisements and localized communication. For determining positions of devices, many localization algorithms have been presented so far. Usually, localization algorithms assume that some location seeds advertise their accurate positions to nodes in their transmission ranges. Some of them assume a few number of seeds with longer transmission ranges (GPS falls into this category) and some others assume a large number of seeds with short transmission ranges in order to cover the target fields by these seeds. However, receiving signals from seeds over a distance requires clear line-of-sight, which is sometimes hard to obtain between buildings, in underground cities and so on. Additionally, it is highly expensive to widely deploy a number of short-range location seeds. Another alternative is to assume mobile seeds to enhance the coverage of regions [1]. This works fine if these mobile seeds well cover the target field. However, depending on the size of the target field and the speeds and density of mobile seeds (especially, if we assume slow pedestrians in large cities), it is also hard for ordinary (non-seed) nodes to find the mobile seeds.

Some techniques exploit indirect information from seeds. Techniques categorized into "collaborative multi-lateration" such as [2] assume that position information of seeds is delivered in a multi-hop way, and the distance to the seeds is approximated by additional information such as the number of hops on the wireless ad hoc networks. The other techniques categorized into "iterative multi-lateration" such as [3] use the estimated position of a node to estimate positions of its neighbors iteratively. However, these techniques may not work well if *the wireless ad hoc networks are mobile and frequently partitioned*, which is true in urban cities. For example, pedestrians and vehicles in cities are not stationary, and they are not always connected due to obstacles such as buildings and due to nonuniform deployment and density.

To overcome this problem, in this paper, we present a localization algorithm called UPL (Urban Pedestrians Localization) designed for positioning mobile terminals in urban city areas. The key idea of UPL is the following. As discussed earlier, in urban cities we do *not* assume that mobile nodes hear signals from location seeds frequently and also mobile nodes 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 the two (or more) areas considering radio range. On the other hand, due to movement of mobile nodes, the area of presence expands as time passes. However, actually in urban cities mobile nodes do not move in free space but in space restricted by walls and streets. Considering this fact, our algorithm fully 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, we assume obstacle maps are simple and lightweight enough, and distributed with location information by seeds.

To confirm the effectiveness of the algorithm, we have conducted realistic simulations using our open source mobile network simulator MobiREAL [4]–[6]. From the experimental results, UPL could achieve smaller position errors (8m in average with 10m of radio range) in a real region of Osaka downtown than two known localization methods.

II. UPL Overview

Each node is equipped with a personal area communication 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 battery limitations. Thus as a realistic hardware environment we assume these PAN communication technologies. For simplicity of discussion, we assume the same communication range r for all the nodes. We also assume the same *maximum* velocity V_{max} for all the

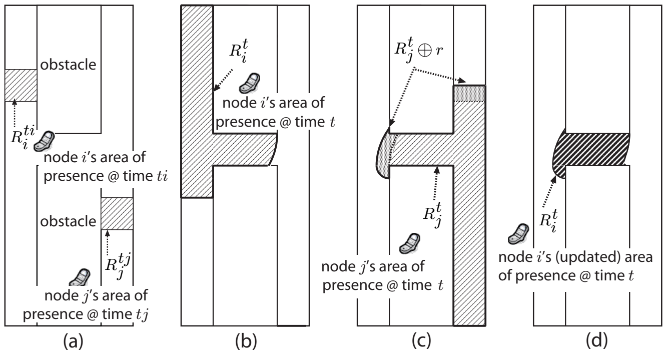


Fig. 1. Localization Process of UPL

nodes (this does not mean that each node moves with the same speed). 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 is the region in which node i expects to exist.

Each node broadcasts hello messages with regular intervals to its neighbors. A hello message transmitted by node i includes the area of presence $R_i^{t_i}$ and time $\Delta t_i = t - t_i$ where t_i is the time when the last localization was executed at node i and t is the time when the message was transmitted. 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 since t_i. We need not assume clock synchronization because Δt_i is calculated by a local timer obviously. The hello message also includes a common obstacle map M of a target region which is lightweight enough. We assume that this map is initially given by seed nodes, and then it is distributed by hello messages among mobile nodes. The data structures of $R_i^{t_i}$ and M are given later in Section III-A.

When node i receives a hello message from node j, node i immediately runs UPL to update its area of presence. UPL is executed as follows. We denote by t the time when node i received the hello message. Node i calculates R_i^t and R_j^t from given $R_i^{t_i}$ and $R_j^{t_j}$, respectively (Figs. 1(a) and 1(b)). 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. 1(c)). We calculate this region by expanding R_j^t by communication range r and it is denoted by $R_j^t \oplus r$. Finally, we obtain the new area of presence of node i at time t, by intersecting R_i^t and $R_j^t \oplus r$ as shown in Fig. 1(d) because node i must be located in both R_i^t and $R_j^t \oplus r$.

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 $\hat{R}_i^t$. Our goal is to design an algorithm that can determine areas of presence of nodes which are as complete and accurate as possible.

Area of presence information itself can be used for many services. For example, if service providers receive an area of presence of a shopping customer, it 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. Also, for services which need to identify the location of nodes such as navigation systems, we present a position estimation function that determines the most likelihood point from a given area of presence. This function is given in Section III-D.

III. UPL Algorithm

A. Area of Presence and Obstacle Map

Several data structures have been introduced and used to represent an area of presence in past localization algorithms. Ref. [3] gives and discusses some data structures in details. In the most simplest way, we may use circles or simple polygons such as rectangles to approximate areas of presence, but obviously it lacks the accuracy of approximation. MCL [1] uses a set of randomly selected points to represent an area of presence, where simplicity is a merit for small hand-held devices. Sextant [3] utilizes a list of representative points and the area is approximated as a set of Bezier curves between those points.

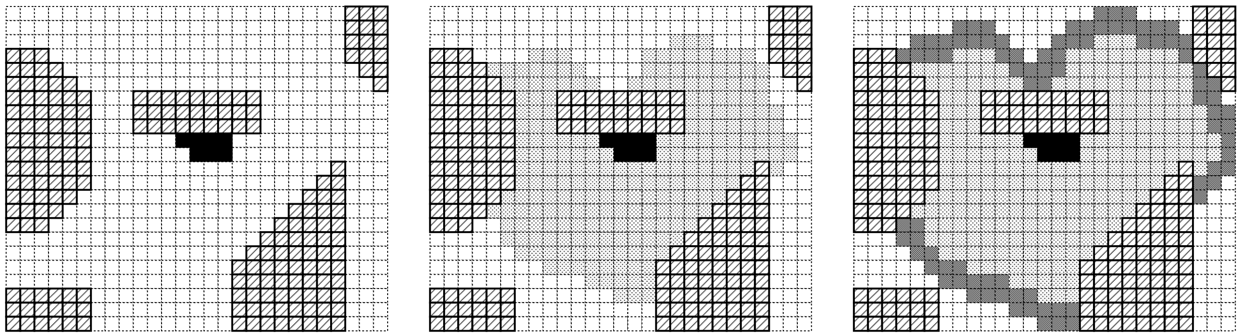
Unlike these methods, we divide a target region into small grids and represent areas of presence and obstacles by sets of grids. Here, the form of each area of presence in UPL is more complex than existing methods because we consider regions restricted by obstacles and the movement of nodes in such complex regions. There are some methods that deal with obstacles. For example, Sextant [3] takes into account an area restricted by obstacles. However, it does not need to consider the computation of movement of nodes because it deals with stationary nodes. Considering the fact that in our case the operations on areas of presence are more complex, we should benefit from simple data structure.

B. UPL Algorithm Description

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 grids. This algorithm is referred to as the APC (Area of Presence Computation) algorithm and does not require complex functions like ones used in graphics libraries, instead we only use simple operations which are lightweight enough.

First, we represent the map M by set FS, the set of all the grids in the movable space (free space) of M. That is, FS is the set of all the grids which are not included by the obstacles. Basically, to calculate R_j^t, the APC algorithm adds to $R_j^{t_j}$ the grids 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 grid in FS. Therefore, we design the algorithm such that we can expand $R_j^{t_j}$ step by step while keeping a certain accuracy.

We introduce some terminologies and notations. For grid g, a grid which shares a side with g is called a *side grid*, and a



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

(b) R_j^t (black and light gray grids; $V_{max}\cdot\Delta t_j{=}10$; $\Delta t_j{=}t{-}t_j$))

(c) $R_j^t \oplus r$ (black, light gray and gray grids; $r{=}2$)

Fig. 2. Computing Area of Presence and Its Expansion by Communication Range

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

The APC algorithm is described as follows. We start with $R_j^t = R_j^{t_j}$ and iterate the following procedure. For each border grid g of R_j^t with the shortest distance $d(g)$, we add to R_j^t each side grid g' of g in FS if $d(g)+w \leq V_{max}\cdot\Delta t_j$ where w is the side length of a grid. If g' is added, we set the shortest distance of g' as $d(g') = d(g) + w$. Similarly, we add to R_j^t each diagonal grid g'' of g in FS 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 grids in FS which can be added to R_j^t. Fig. 2 shows examples of $R_j^{t_j}$, R_j^t and $R_j^t \oplus r$.

2) Expanding Area of Presence by Communication Range: Then we compute $R_j^t \oplus r$, the area of presence of node j at time t expanded by communication range r. For this purpose, we may apply the APC algorithm. Here, radio propagation may be affected by phenomena such as fading and diffraction especially in city sections. Therefore, ideally, we should design a propagation prediction algorithm considering these phenomena. However, in our environment, it is not reasonable to implement such a complex algorithm. Also, as we stated briefly in Section II, we assume PAN communication devices with relatively small communication range. Therefore, we may ignore the impact by those phenomena and simply use the APC algorithm to expand R_j^t.

3) Intersecting Two Areas: Intersecting R_i^t and $R_j^t \oplus r$ is simple. We seek grids which are included in the two areas and obtain the new area of presence, $R_i^t \cap \{R_j^t \oplus r\}$. Finally, computing the intersection of two polygons can be done by the Boolean operation of intersection using a map overlay technique.

C. 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 seed. Concretely, by knowing the timing when the node entered or left the communication range of a seed, we may be able to identify the position of the node exactly on the disc edge of the communication range of the seed. Also some try to extend the communication range of a seed by multi-hop propagation of the seed information. In our experiments, we have borrowed two sophisticated techniques from MCL [1], one of which uses the history of signal receptions from a seed called leaver and arriver, and another uses two-hops advertisement of seed positions to extend the coverage of seeds.

D. Position Estimation

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 most likelihood point in the area of presence. We have used the following estimation function to determine the most likelihood point p from an area of presence R and 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). Considering the fact that the actual point should exists within R, selecting such p that minimizes the maximum distance between p and another point p' in R is helpful to minimize errors of positioning.

IV. Performance Evaluation

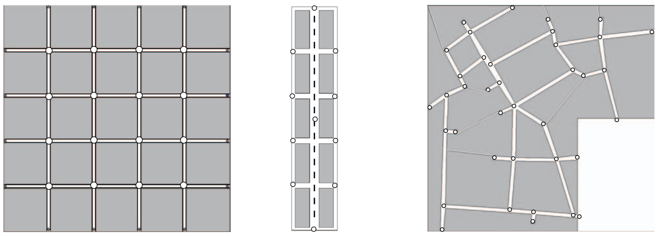
A. Simulation Settings

We have used three maps, (i) *Manhattan region* [7] of $500m\times500m$ with 8 streets (Fig. 3(a)), (ii) *Divided road region* of $100m\times500m$ with a road divided by a median strip (Fig. 3(b)) and (iii) the map of a $500m\times500m$ real region in front of the Osaka train station (Fig. 3(c)) called *Osaka downtown region*. In all the regions, we have specified areas except roads as obstacles and nodes are moved according to

TABLE I

SIMULATION PARAMETERS

Grid length (w)	1, **2** or 4 (m)
Radio range (r)	10 (m)
# of nodes	500, 1,000, **2,000** or 3,000
Speed distribution	Normal distribution of **[1.0, 2.0]**, [3.0, 4.0], [5.0, 6.0], [7.0, 8.0] or [9.0, 10.0] ($m/sec.$)
Hello message interval	2 ($sec.$)
Initial node deployment	Uniform distribution



(a) Manhattan (b) Divided Road (c) Osaka Downtown

Fig. 3. Simulation Maps (Snapshot from MobiREAL Animator where positions of the seeds are plotted and shaded areas are obstacles)

TABLE II

IMPACT OF PARAMETERS AND ENVIRONMENTS ON PERFORMANCE

Parameter/Environment		Accuracy (m^2)	Completeness
Grid length (m)	1	213.3	1.0
	2	216.1	1.0
	4	408.4	1.0
Moving speed ($m/sec.$)	[1.0, 2.0]	216.1	1.0
	[3.0, 4.0]	257.1	1.0
	[5.0, 6.0]	319.1	1.0
	[7.0, 8.0]	397.0	1.0
	[9.0. 10.0]	438.0	0.96
# of nodes	500	435.5	1.0
	1,000	295.7	1.0
	2,000	216.1	1.0
	3,000	188.6	1.0

"random-street-decision" mobility where at each intersection each node decides to which direction it goes, except the backward direction. In the Manhattan region, we deployed 16 seeds and the width of roads was $8m$. The other settings in the Manhattan region are described in Table I where default values are emphasized by bold font. In the Divided road region, we set the hello message interval to $5sec.$, and the number of nodes and the number of seeds were set to 400 and 11, respectively. The other parameters were set to the default values in Table I. We note that nodes can communicate through the median strip which is represented as the dashed line in Fig. 3(b) although they cannot go across it. Finally, in the Osaka downtown region, we used the random-street-decision mobility except that in the free space of $170m\times250m$ at the bottom right corner we used the Random Waypoint mobility [8]. We deployed 38 seeds, and the other parameters were set to the default values described in Table I.

We conducted three types of experiments. The experiments of the first type were conducted in the Manhattan region to see the performance characteristics of UPL varying several parameters. In particular, we focused on grid length, moving speed of nodes and the number of nodes. The experiments of the second type were done to validate the effects of (i) ad hoc localization and (ii) precise calculation of node movement among obstacles. For this purpose, we also evaluated the performance of two simplified versions of UPL; UPL_{no_adhoc} which did not perform ad hoc localization between mobile nodes, and UPL_{no_obs} which did not utilize obstacle information to predict the movement of nodes. In these experiments we have used the Divided road region. The experiments of the third type were conducted to see the performance compared with typical existing methods. We have selected Amorphous [2] which performs multi-hop cooperative multi-lateration, and MCL [1] which uses short-range seeds. These methods estimate the position of each node, therefore, we have used the position estimation function (1) of Section III-D. In these experiments we have used the Osaka downtown region to see the practical performance in a real region.

All results are derived from 10 simulation cases. The experimental results of these three types are presented and analyzed in the following sections, Sections IV-B, IV-C and IV-D, respectively.

B. *Impact of Parameters and Environments on Performance*

For the areas of presence of nodes, we have measured *accuracy* and *completeness*. To see accuracy, we have measured the average sizes of areas of presence. Here, the completeness is the ratio of the localizations which generated complete areas of presence to all the performed localizations. The results are shown in Table II.

a) Grid Size: We have evaluated accuracy and completeness under different values of grid length w ($1m$, $2m$ and $4m$). From the result, we do not see the difference of completeness for different values of w (the completeness was 1.0 in all cases, that is, the perfect completeness), and we can see explicit difference of accuracy. This is because smaller grids can represent the borders of areas of presence more accurately. In the case of $w = 4$, the average size was $408.4m^2$, which is rather large compared with the cases of $w = 2$ and $w = 1$. Also we do not see big difference of accuracy between the cases of $w = 1$ and $w = 2$. Considering the fact that the grid size quadratically affects memory space, we may select $w = 2$.

b) Moving Speed: As movement speeds become large, each node may be able to 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 minimum and maximum speeds. The size of the area of presence increases linearly as the speed increases. In the case of $v = [9.0, 10.0]$, completeness decreases a little bit. This is because in the APC algorithm, a straight moving trajectory is approximated by a zigzag line with the same length, which results in smaller estimation of area of presence than the most accurate, complete one. As the moving speed is higher, the moving distance becomes larger and accordingly this approximation error becomes larger (*i.e.* completeness

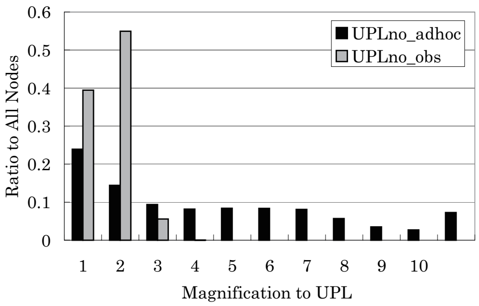


Fig. 4. Distribution of Magnification in Divided Road Region

TABLE III
MAGNIFICATION AND COMPLETENESS

	Magnification	Completeness
UPL$_{no_adhoc}$	4.396	1.0
UPL$_{no_obs}$	1.264	1.0
UPL	1.000	1.0

becomes lower). Thus slow speed is better for accuracy and completeness. This result proves that our localization is well fit for slow pedestrians.

c) Number of Nodes: Obviously, UPL is largely affected by node density. Thus we have varied the number of nodes and have measured accuracy and completeness. From the result, we need a certain density to achieve accurate areas of presence. However, this means that nodes need to *encounter* with a certain number of nodes, and do *not* require a well-connected ad hoc network. Such feature is good in city sections where it is hard to assume that large ad hoc networks are constructed and maintained.

C. Effect of Ad-hoc Localization and Precise Calculation of Movement

We have measured the accuracy in UPL$_{no_adhoc}$ and UPL$_{no_obs}$, and calculated the ratios to original UPL. We call such a ratio *magnification*.

Fig. 4 shows the distributions at points in the whole region. As we can see, obstacle information gives a certain amount of impact on the accuracy of many nodes. On the other hand, ad hoc localization dramatically improves the accuracy of some nodes. This fact indicates that in some places which are away from seeds, ad hoc localization is very helpful.

Table III shows the averages of the magnification and completeness. We can see that all versions of UPL are complete and that ad-hoc localization and obstacles information effectively reduce 77% and 21% of the sizes of the areas of presence, respectively. From this result, we could confirm the effectiveness of two key ideas of UPL.

D. Comparison with Other Approaches

For comparison purpose, we have measured the estimated position errors of UPL, MCL [1] and Amorphous [2] using the Osaka downtown region. Fig. 5 shows the position errors regarding the communication range ($r = 10m$) according to

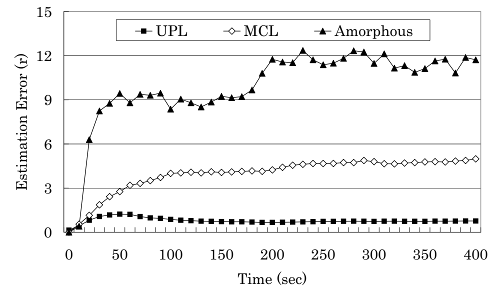


Fig. 5. Estimated Position Errors

the progress of simulation time. Here, the important fact is that Amorphous achieved relatively low accuracy, and MCL performed better than Amorphous even though MCL does not use ad hoc networks and just uses direct information from seeds. This is because accurate estimation by a hop-based technique may be difficult in urban district where most space is restricted by obstacles. On the other hand, UPL outperforms these methods and the error is at most $0.8r$, that is, $8m$.

V. CONCLUSION

In this paper, we have proposed a range-free localization algorithm called UPL (Urban Pedestrians Localization) for positioning mobile users in urban district. From several experimental results, we have shown that UPL could achieve reasonable accuracy for positioning of mobile users in urban city areas, compared with typical existing methods. More precise evaluation of the proposed algorithm using other several realistic urban city regions is part of our future work.

REFERENCES

[1] L. Hu and D. Evans, “Localization for mobile sensor networks,” in *Proc. of 10th Annual International Conference on Mobile Computing and Networking (MobiCom 2004)*, 2004, pp. 45–57.

[2] R. Nagpal, H. Shrobe, and J. Bachrach, “Organizing a global coordinate system from local information on an ad hoc sensor network,” in *Proc. of Information Processing in Sensor Networks*, 2003, pp. 333–348.

[3] S. Guha, R. Murty, and E. G. Sirer, “Sextant: a unified node and event localization framework using non-convex constraints,” in *Proc. of 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005)*, 2005, pp. 205–216.

[4] *MobiREAL web page*, tools are available on http://www.mobireal.net.

[5] K. Konishi, K. Maeda, K. Sato, A. Yamasaki, H. Yamaguchi, K. Yasumoto, and T. Higashino, “MobiREAL simulator – evaluating MANET applications in real environments –,” in *Proc. of 13th IEEE Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2005, pp. 499–502.

[6] 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,” in *Proc. of 8th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2005)*, 2005, pp. 151–158.

[7] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications & Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483–502, 2002.

[8] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *Proc. of 4th Annual International Conference on Mobile Computing and Networking (MobiCom 1998)*, 1998, pp. 85–97.