A Dummy-based Anonymization Method Based on User Trajectory with Pauses

Ryo Kato
Osaka University, Japan
kato.ryo@ise.eng.osaka-u.ac.jp

Mayu Iwata
Osaka University, Japan
iwata.mayu@ist.osaka-u.ac.jp

Takahiro Hara
Osaka University, Japan
hara@ist.osaka-u.ac.jp

Akiyoshi Suzuki
Osaka University, Japan
suzuki.akiyoshi@ist.osaka-u.ac.jp

Yuki Arase
Microsoft Research Asia
yukiar@microsoft.com

Xing Xie
Microsoft Research Asia
xingx@microsoft.com

Shojiro Nishio
Osaka University, Japan
nishio@ist.osaka-u.ac.jp

ABSTRACT

A variety of services utilizing users' positions have become available because of rapid advances in Global Positioning System (GPS) technologies. Since location information may reveal private information, preserving location privacy has become a significant issue. We proposed a dummy-based method of anonymizing location to protect this privacy in our previous work that generated dummies based on various restrictions in a real environment. However, the previous work assumed a simplified mobility model in which users kept moving and did not stop. If we assume a more realistic mobility model in which users often pause to visit various attractions, it becomes increasingly more difficult to generate dummies that will move naturally. In this paper, we assumed that the users' movements are known in advance and propose a dummy-based anonymization method based on user movements, where dummies move naturally while stopping at several locations. We simulated user movements on real map information and verified the method we propose was more effective than the previous one.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human information processing

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Location-based services, Location privacy, Mobile computing, GPS

1. INTRODUCTION

Location based services (LBSs) are becoming more common due to the growing popularity of mobile devices equipped with GPS receivers. LBS providers provide a variety of services based on user locations, such as local searches, planning of routes, and location based advertisements. However, location information provides, or enables, a lot of private information to be inferred, e.g., where an LBS user is living, to which school his/her children go, and where his/her friends live. Krumm [13] issued warnings about this problem. His experiments revealed that it is possible to estimate a user's home location within a range of 60 meters by only using the last location they used that day. The situation is more serious when users continuously use LBSs, such as searching near-by attractions while visiting cities, since their accumulated location histories make it easier to detect private locations. Most commercial LBSs require us to update our position every few minutes according to an investigation conducted by Busic and Filjar [5]. Beresford et al. [2] defined location privacy as the ability to prevent other parties from learning one's current or past location. They also warned that a system collecting users' locations potentially invaded their location privacy.

Numerous studies have been conducted to protect users' location privacy. There are two requirements to deploy a system that preserves users' location privacy [17]: 1) it should be a closed system, i.e., able to be executed on users' mobile devices and not leak their location information outside and 2) it should not disrupt benefits to users or LBS providers. The second requirement is important to make the entire ecosystem beneficial; otherwise, no users or LBSs would use a system to protect privacy. Dummy-based methods [12], [14], [18] have satisfied these requirements in previous studies. They have generated dummy users and sent their locations with the user's location to LBSs, so that the LBSs could not distinguish the locations of users or dummies. However, these previous methods did not take into consideration physical constraints in a real environment, and thus, their actual robustness in protecting privacy was questionable. The robustness of dummy-based methods strongly depends on how naturally the dummies behave. If dummies do not behave like humans, such as moving at unreasonable speeds and being in the middle of seas or tops of mountains, it is easy to identify them as dummies.

ACM SIGSPATIAL GIS '12, November 6-9, 2012. Redondo Beach, CA, USA

The traceability of locations is also a critical issue in a real environment. Since there are no perfect systems, it is always possible that user locations will accidentally be exposed. If this happens, the revealed location may further divulge locations where users were and are going to be. For example, if there are no dummies in a reachable area from a revealed location in a subsequent query, it is apparent that whoever is in the area is the user. Users' past/future locations can be traced from the exposed location in this way.

We proposed a dummy-based method of anonymizing location that generated dummies around a user taking into consideration restrictions in the real world to solve these problems and protect location privacy in a real environment in our previous work [17]. We also simulated the movement of dummies to make them more natural during consecutive use of LBS, where we tried to lower the traceability of user locations.

However, the previous work assumed a simplified mobility model in which a user kept moving and did not stop. When we assume a more realistic mobility model in which the user often pauses to visit various attractions, it becomes more difficult to generate dummies that can move naturally. For example, even when a user stops at a particular location, dummies cannot simply pause at the current spot if there are no attractions around them, i.e., pausing there would be an unnatural behavior.

We have assumed that user movements (e.g., trajectory, pause time, and position) are known in advance in this paper, as a first step, and propose a dummy-based method of anonymization based on the user movements with pauses. Our method generates dummies that move naturally while stopping at several locations taking into account geographical information such as the locations of attractions. Note that we have assumed user movements where what he/she is going to do can be precisely predicted in advance in this paper. Our previous method did not make this assumption and only determined subsequent locations of dummies based on the current and past locations of the user and dummies. Thus, it was difficult to react to user pauses as previously described. However, our method can determine dummies' movements in advance based on predicted user movements based on this assumption.

Although this assumption might seem too strong (or unrealistic) in real situations, there are many situations where we can predict user movements in advance. For example, we can predict a user's future movements based on certain additional information, e.g., his/her pre-registered plans on where and when he/she is going to go and the history of his/her trajectories, even though these are not always very accurate. Although there have been a large number of studies on accurately predicting user movements, a review of these is beyond the scope of this paper. It also makes sense to assume that a user has explicitly registered his/her plans on movements before using our method, so that his/her location privacy is well protected. Nonetheless, we have focused on how to generate dummies based on user movements that have been predicted in advance with a great deal of accuracy in this paper.

Our method is used to determine the positions and times that dummies should pause and it generates the dummies' movements based on them. Our method distributes dummies uniformly in the target area to anonymize a user's location, and stops them at positions where the user can pause. It makes dummies cross the user's path and that of other dummies at positions where they have paused to lower traceability. We simulated a user's movements on a real map and verified that our new method was more effective than our previous approach.

The three main contributions of this paper are summarized below.

- We propose a method that takes into account user movements with pauses to generate natural dummies. It lowers the traceability of user locations to quickly recover from accidental disclosures of user locations.

- Our method is practical in that can be directly applied to the current ecosystem of LBSs. A user and dummies specifically share the user's own registration ID on an LBS, which enables the user to take advantage of the membership benefits of the LBS while the LBS providers can obtain the user's approximate location without the risk of invading his/her privacy.

The rest of the paper is organized as follows. Section 2 describes related work and Section 3 presents details on the proposed method. Section 4 describes our evaluation of the new approach and Section 5 concludes the paper with a discussion on future work.

2. RELATED WORK

There have been numerous studies on protecting location privacy. We can categorize them into three approaches that 1) create intermediates between users and LBS servers, 2) transform user locations, and 3) generate dummy locations.

The first approach called spatial cloaking [6], [7], [9], [10], [15], [19] (or the generalization [3] method) ensures a user's location is mixed with at least k candidates, i.e., a user's location cannot be identified over the probability of $1/k$. It collects k users' locations and sends the minimum region including those k users to an LBS server as a query instead of the user's exact location. The hiding method [2] conceals users' locations during a certain period. Users during this hiding period do not request LBSs and exchange their user names with one another to make it difficult for tracers to connect to their previous/subsequent locations. By doing so, adversaries cannot trace a user's location history, even if it could identify the user's location at a certain timing. All of these methods need to pool users' locations, and they thus assume a trusted third-party server to mediate interactions between the users and the LBS server [2], [9], [10], [11], [19], or use peer-to-peer collaboration between mobile users [7]. However, it is difficult in practice to deploy a completely safe third-party server. In addition, mobile peer-to-peer collaboration suffers from the same problem of location privacy, since users have to share their location information with others they do not know. Moreover, these methods fail to anonymize a user's location if there are insufficient numbers of users around him/her.

The second approach is the method of obfuscation [1], [8] that replaces a user's location with a near-by intersection or building to obscure his/her real location. However, if there are no appropriate targets around the user, the substitute location is far from that of the user, which degrades the quality of the LBS. The method of spatial transformation [11] uses Hilbert curves to transform the user's location and sends the transformed location to the LBS server. Since the transformation is one-way, the LBS provider cannot decode the user's location. The disadvantage of this method is that it also needs LBS providers to transform all their location data (such as locations of shops), which is not a trivial effort in maintaining services.

The last approach is dummy-based [12], [14], which generates dummies and sends their locations with the actual user's location to an LBS server, as was described in Section 1. Fig. 1 outlines an example where a user is issuing a query asking for near-by restaurants; this approach sends the user's location with the locations of dummies. The LBS provider then returns lists of restaurants that are close to each of the locations in the query (the user's and dummies' locations). The user can choose a restaurant from the list by ordering results based on the distances from his/her location (or

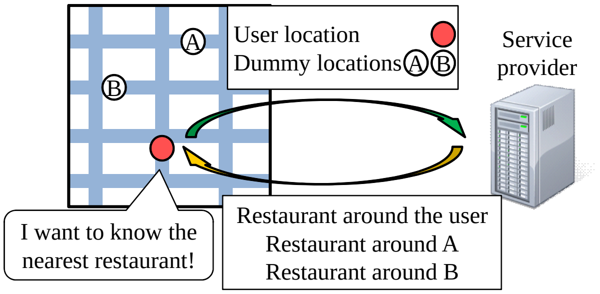


Figure 1: Example of dummy-based approach

by simply filtering out all results corresponding to the dummies' locations).

We should, of course, anonymize a user's location as a practical solution to protecting location privacy. We should simultaneously not disrupt the benefits users and LBS providers are supposed to have; otherwise, the solution is less likely to be used. For example, we can anonymize a user's location by simply using randomly generated pseudonyms. However, most LBSs require user registration, and users also cannot take advantage of premium services that may be provided to active users.

We adopted the dummy-based approach, because it satisfies all the above requirements. The LBS provider can obtain information on customer behavior and directly charge the user service fees, and the user may subscribe to specialized services from the LBS, such as service discounts and personalized services. This is important for the entire ecosystem of LBSs. Moreover, it is reliable since it does not need intermediate servers/devices to collect or process location information. We previously proposed [17] a dummy-based method of anonymizing location that generated dummies around the user in a grid and that took into consideration physical constraints in a real environment. The previous method assumed a simplified mobility model in which users kept moving and did not stop. When we assume a more realistic mobility model in which users often pause to visiting various attractions, it becomes more difficult to generate dummies that move naturally. Our method in this paper generates dummies based on known user movement with pauses. By doing so, generated dummies can move naturally while stopping at several locations like an actual user.

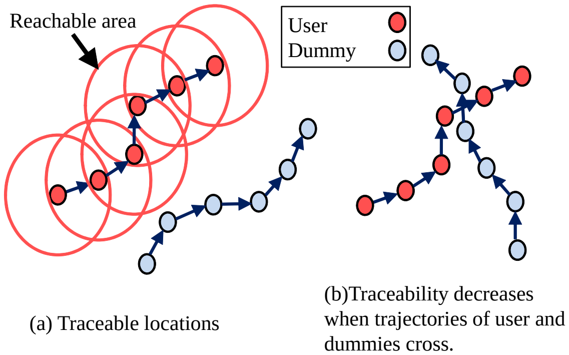
3. PROPOSED METHOD

Our method anonymizes a user's location with dummies based on a known user's movements taking into account restrictions in a real environment. This section first presents an assumption about LBSs, then discusses restrictions in a real environment, and describes our method in detail.

3.1 Assumption

We have assumed LBSs in which a user successively issues service requests (at constant or inconstant intervals) by sending his/her location information to the LBS provider, and the LBS provider sends back information related to the user's current location. The user's device sends his/her location information with that of some dummy locations to protect his/her location privacy.

We have assumed a mobility model of the user in which he/she is moving while stopping at several locations on the way to a final destination, e.g., he/she is stopping by a convenience store on the way to his/her office. More specifically, he/she is walking with some distribution of speed and is stopping at several locations for a certain time. The user is using the shortest route between locations where he/she is stopping. We have assumed that the maximum speed, the minimum and maximum pause times, and the user's movement plans (e.g., trajectory, pause time, and position) are known in advance.



(a) Traceable locations

(b)Traceability decreases when trajectories of user and dummies cross.

Figure 2: Traceability

We have also assumed that the user's mobile device has prefetched the map data of the area around where he/she is located [1] to understand geographical information (e.g., which road he/she can walk on and at which location he/she can stop).

3.2 Restrictions in Real Environment

3.2.1 *Consistency of Movements*

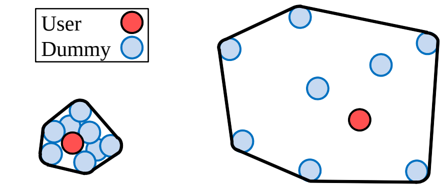
If the current location of a dummy is unreachable (too far) from any previous locations of a user and all dummies, (remember that the LBS provider cannot clearly distinguish them), the LBS provider can easily detect that this location is that of a dummy. For example, when a user has requested an LBS at a certain time and then requests the service again three minutes later, a dummy located more than 10 km away from the previous location is obviously a dummy. Therefore, we should determine locations of dummies to maintain consistency in their movements, i.e., the location of each dummy has to be within a reachable area from its previous location. In addition, we should consider actual road networks when calculating the distance between two locations, rather than a simple Euclidean distance. We also should exclude areas that people normally do not inhabit, such as seas and forests, as locations for dummies. For example, it is easy to detect a dummy if it is moving from a pedestrian sidewalk to the center of a highway, even though the moving distance is acceptable in terms of its moving speed.

Our method determines the locations of dummies by taking into consideration the actual map information to satisfy these conditions. More specifically, our method assumes all dummies are continuously moving at almost the same speed as the user (i.e., they have not jumped to a distant location) only on road networks.

3.2.2 *Traceability*

We should also take into account the traceability of user locations in a real environment. It might be possible to infer the trajectories of a user's movements from the location history that has been accumulated at an LBS provider based on limitations in people's movements. We define traceability as the ability to identify a user's trajectory by combining consecutive locations during a certain period. The traceability problem becomes serious especially when the user's location is accidentally detected by whatever means. If the user's locations are traceable, all previous (and possibly future) locations also become obvious. For example, when a user's locations are traceable, the user's trajectory is easily distinguished from

[1] Mobile applications providing this kind of functionality are actually available, partly due to public map data on OpenStreetMap project (http://www.openstreetmap.org/).



(a) Lower anonymity (b) Higher anonymity

Figure 3: Anonymous Area

dummies, as Fig. 2 (a) shows. A simple but effective approach to lower traceability is to cross the trajectories of the user and dummies. Yanagisawa et al. proposed a method [18] of crossing the user's and dummies' paths by keeping dummies waiting until the user had arrived at the dummies' locations. This meant that crossing was only triggered when the user was heading to the same intersection as the dummies. However, since dummies randomly moved with this method and it resulted in them spreading out over a wide area, there were few chances for the user and dummies to cross.

Our method increases the chances they will cross by sharing the positions where they pause.

3.2.3 Anonymous area

We defined the anonymous area to measure the anonymity of the user's location as a criterion to evaluate how secure a user was in terms of location privacy. The anonymous area was defined following Lu et al. [14] as the size of the minimum convex covering all locations included in a service request to an LBS. Fig. 3 (b) has larger location anonymity than Fig. 3 (a), i.e., the anonymous area in Fig. 3 (b) is larger than that in Fig. 3 (a). It should be noted that the same anonymous area in a real environment does not simply mean the same degree of location anonymity. For example, even though the sizes of two anonymous areas are the same, more buildings are contained in big cities than in rural areas, and thus, location anonymity in big cities is higher than that in rural areas. The appropriate size for an anonymous area depends on the situation. Thus, we have assumed that users (or applications) specify their requirements regarding the size of the anonymous area (and the number of dummies) in this paper.

Lu et al. proposed a method of arranging dummies in a grid so that its size satisfied the anonymous area requirement [14] for the user's location. Although this method focused on the user's anonymous area, it neither took into account the consistency of movements nor real geographic constraints.

Our method does not explicitly construct a grid by using the user and dummies, but it locates dummies at positions (cells) where there are fewer of them (and the user) so that it can roughly form a grid and satisfy the required size for the anonymous area.

3.3 Our Approach

Our method first determines sets of positions where dummies should pause (*pause positions*) and times when they should start to pause at the pause positions (*pause start times*) based on plans of a known user's movements and his/her requirements for the size of the anonymous area and the number of dummies. Then, our method determines the dummies' movements where they move toward the pause positions and arrive there at the pause start times (then start to pause).

Our method specifically carries out the following procedure by repeating Steps (1) to (3) one by one for all the dummies. The first dummy is generated based on plans for the known user's movements, and the subsequent ones are based on both the generated dummies' and user's movements.

Algorithm 1 : Determining the base pause position and base pause start time of kth dummy

```
input: a list of a user and generated dummies' movements $D = \{D_0, \dots, D_{k-1}\}$ (sets of their pause positions, pause start times and pause durations), simulation end time $t_{end}$
output: a pause position and pause start time $PP$ of $k$th dummy

$t \leftarrow 0$
repeat
    //generate a grid every 1000 [s]
    $t \leftarrow t + 1000$
    generate a grid with $3 \times 3$ cells $G = \{G_0, \dots, G_8\}$ around the center of positions of $D$ at $t$
    for $i = 0$ to 8 do
        $G_i.exist_t \leftarrow$ the number of members in $D$ within $G_i$
    end for
until $t \geq t_{end}$
//get a grid cell where the smallest number of the user and dummies exist
$G_{base} \leftarrow G_i$ with min($\sum_{t=0}^{t_{end}} G_0.exist_t, \dots, \sum_{t=0}^{t_{end}} G_8.exist_t$)
//get the earliest time when the smallest number starts and keeps for $T$ [s]
$t_{base} \leftarrow$ time when min($\sum_{t=0}^{T} G_{base}.exist_t, \dots, \sum_{t=t_{end}-T}^{t_{end}} G_{base}.exist_t$) occurred
repeat
    $p_{base} \leftarrow$ GetPausePositon($t_{base}$)
until $p_{base}$ is within $G_{base}$
return $PP \leftarrow <p_{base}, t_{base}>$

//get a pause position where the user and dummies can stop based on the map
function GetPausePosition(time $t$)
    required anonymous area size $anonymousArea$
    possible pause positions based on the map information $P = \{P_0, \dots, P_m\}$
    $center \leftarrow$ the center of positions of the user and dummies in $D$ at $t$
    repeat
        possible pause position $p \leftarrow$ random($P_0, \dots, P_m$)
    until $p$ is within $anonymousArea$ around $center$
    return $p$
```

(1) Determine the base pause position and base pause start time of a new dummy to satisfy the required size of the anonymous area.

(2) Determine sets of shared pause positions and shared pause start times of the dummy to lower traceability.

(3) Determine the dummies' movements where it has moved while stopping at the initial and shared pause positions.

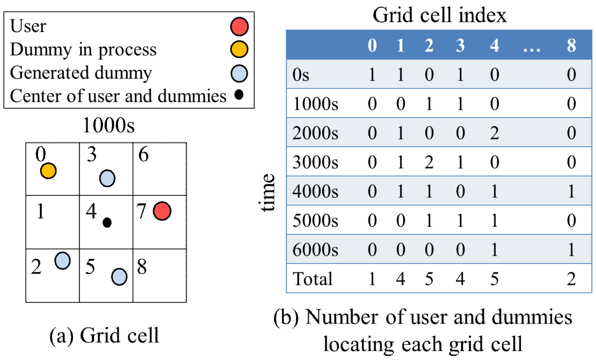
The following subsection provides details on the three steps.

3.3.1 Determining base pause position and base pause start time

First, our method determines the pause position and the pause start time of a new dummy to satisfy the required size of the anonymous area. We define this position as the *base pause position* and the time as the *base pause start time*. The dummy effectively arrives at the base pause position at the base pause start time and stops there for a certain period of time.

Algorithm 1 shows the algorithm for determining the base pause position and base pause start time of a new dummy.

As Fig. 4(a) shows, the base pause start time and base pause position are determined based on the grid containing the user's and dummies' locations that have already been generated at a certain interval of time. The grid is a square having three × three grid cells and each grid cell has an index. The width of the grid is $\sqrt{S}$ to satisfy the required size of the anonymous area, S. The center of the grid is set to the average position of the user's and dummies' locations that have already been generated. The area and time with the smallest number of users and dummies can be identified by counting the number of users and generated dummies located in each grid cell for all time intervals between the start time (0 [s])



(a) Grid cell

Grid cell index

time	0	1	2	3	4	…	8
0s	1	1	0	1	0		0
1000s	0	0	1	1	0		0
2000s	0	1	0	0	2		0
3000s	0	1	2	1	0		0
4000s	0	1	1	0	1		1
5000s	0	0	1	1	1		0
6000s	0	0	0	0	1		1
Total	1	4	5	4	5		2

(b) Number of user and dummies locating each grid cell

Figure 4: Example of determining the base pause position and base pause start time

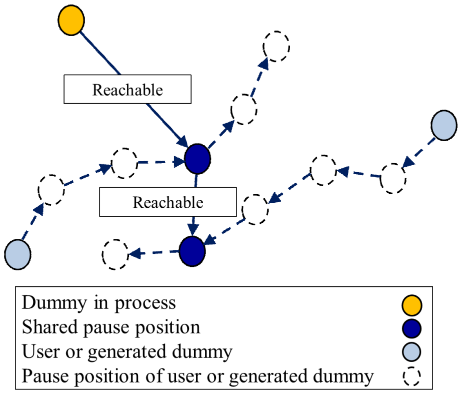

Reachable
Reachable
Dummy in process ●
Shared pause position ●
User or generated dummy ●
Pause position of user or generated dummy ○

Figure 5: Sharing pause positions of user and dummies

of the user's movements and end time. The base pause position is determined as a random position in that area (grid cell). The base pause start time is determined as the time when the smallest number first occurred in that area.

For example, the bottom row in the table in Fig. 4 lists how many users and dummies there were in each grid cell, where counting was carried out every 1,000 [s]. There are no users or dummies between 1,000 [s] and 5,000 [s] in the grid cell with index 0, i.e., the number of users and dummies is the smallest of all grid cells. The base pause position in this case is determined to be a position in that grid cell and the base pause start time is determined to be 1,000 [s].

3.3.2 Determining sets of shared pause positions and shared pause start times

After the base pause position and base pause start time are determined, our method then generates sets of pause positions and their pause start times to lower traceability. To cross dummies, it specifically makes the dummy in the process share the pause positions of the user and already generated dummies when they stop. We define these positions as *shared pause positions* and times as *shared pause start times*.

Algorithm 2 shows the algorithm for determining sets of shared pause positions and shared pause start times.

First, our method finds pause positions for the user or already generated dummies within a reachable area from the base pause position of the dummy in the process, as Fig. 5 shows. One of these pause positions is determined to be a shared pause position, and the time when the dummy will arrive there is determined as a shared pause start time. Then, additional shared pause positions are successively determined based on the areas reachable from the shared pause positions that have already been determined and the base pause position. Our method finds as many shared pause positions as possible in this way until there are no other pause positions that the dummy can share with the user and already generated dummies.

Algorithm 2 : Determining sets of shared pause positions and shared pause start times of kth dummy

```
input: a list of a user and generated dummies' pause movements $D = \{D_0, ..., D_{k-1}\}$ (sets of their pause positions, pause start times, pause durations, and the number of cross $cross$), sets of pause potistions and pause start times $PP$ of $k$th dummy
output: sets of pause potistions and pause start times $PP$ of $k$th dummy

repeat
    $D_{mincross} \leftarrow D_i$ with min($D_0.cross$, ..., $D_{k-1}.cross$)
    if all pause start times in $PP$ of $k$th dummy $\leq t_{shared}$ then
        repeat
            $p_{shared} \leftarrow$ random(pause positions of $D_{mincross}$)
            $t_{shared} \leftarrow$ pause start time with $p_{shared}$ in $D_{mincross}$
        until $p_{shared}$ is within the reachable area from a pause position with the latest pause start time in $PP$ of $k$th dummy
    else if all pause start times in $PP$ of $k$th dummy $\geq t_{shared}$ then
        repeat
            $p_{shared} \leftarrow$ random(pause positions of $D_{mincross}$)
            $t_{shared} \leftarrow$ pause start time with $p_{shared}$ in $D_{mincross}$
        until a pause position with the earliest pause start time in $PP$ of $k$th dummy is within the reachable area from $p_{shared}$
    else
        repeat
            $p_{shared} \leftarrow$ random(pause positions of $D_{mincross}$)
            $t_{shared} \leftarrow$ pause start time with $p_{shared}$ in $D_{mincross}$
        until ($p_{shared}$ is within the reachable area from a pause position with $t_{shared}$'s previous pause start time in $PP$ of $k$th dummy) AND
        (a pause position with $t_{shared}$'s next pause start time in $PP$ of $k$th dummy is within the reachable area from $p_{shared}$)
    end if
    append <$p_{shared}, t_{shared}$> to $PP$
    increase $D_{mincross}.cross$
    increase $k$th dummy.$cross$
until $k$th dummy.$cross \geq$ ave($D_0.cross$, ..., $D_{k-1}.cross$)
return $SP$
```

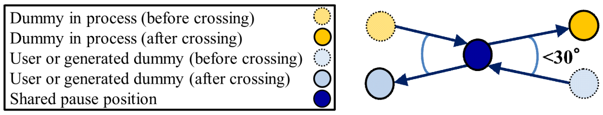

Dummy in process (before crossing)
Dummy in process (after crossing)
User or generated dummy (before crossing)
User or generated dummy (after crossing)
Shared pause position
<30°

Figure 6: Crossing not decreasing traceability

When a shared pause position is chosen from possible candidates, our method takes into account how many times each user and the dummies will cross the paths of others. It specifically and preferentially chooses a pause position for the user or a dummy with the smallest number of crossings as a shared pause position to ensure fairness between the user and dummies in terms of the number of crossings.

Our method does not choose a shared pause position if the direction of movement by the dummy in the process and that of the dummy (or the user) sharing the pause position are almost opposite and do not change much after the pauses (i.e., are almost straight), as Fig. 6 shows. This is because their routes can easily be distinguished from each other in this case since returning to the direction of arrival is unnatural for the user while going straight is natural. Our method specifically does not choose a shared pause position when the angle of the entrance direction of a dummy and the exit direction of another dummy is less than 30°.

3.3.3 Determinig dummy's movements

Even though our method simply connects the base and shared pause positions, the dummy's movements are not always as natural

Algorithm 3 : Determining kth dummy's movement

```
input: sets of pause positions and pause start times $PP_in$ of $k$th dummy, minimum pause duration $T_m$, maximum pause duration $T_M$, simulation end time $t_{end}$
output: sets of pause positions, pause start times, and pause durations $PP_{out} = \{<position_0, start_0, pause_0>, ..., <position_n, start_n, pause_n>\}$ of $k$th dummy

destination position start time $s_{dest} \leftarrow$ min(pause star times in $PP_{in}$)
destination position $p_{dest} \leftarrow$ a pause position with $s_{dest}$ in $PP_{in}$

//determine the first position
$pause_0 \leftarrow 0$
$start_0 \leftarrow 0$
repeat
    $position_0 \leftarrow$ GetPausePosition(0)
until $p_{dest}$ is within the reachable area from $position_0$
append $<position_0, start_0, pause_0>$ to $PP_{out}$

//determine pause positions after the first position
$i \leftarrow 1$
repeat
    repeat
        repeat
            //generate mid-pause position
            $pause_i \leftarrow$ random($T_m, T_M$)
            repeat
                $position_i \leftarrow$ GetPausePosition($start_{i-1} + pause_{i-1}$)
            until $position_i$ is within the rechable area from $position_{i-1}$
            $start_i \leftarrow start_{i-1} + pause_{i-1}$ + time from $position_{i-1}$ to $position_i$
            append $<position_i, start_i, pause_i>$ to $PP_{out}$
            $i \leftarrow i + 1$
        until $p_{dest}$ is unreachable from $position_i$
        //set a destination position as the next pause position
        $position_i \leftarrow p_{dest}$
        $start_i \leftarrow s_{dest}$
        $pause_{i-1} \leftarrow s_{dest} - start_{i-1} -$ time from $position_{i-1}$ to $p_{dest}$
        append $<position_i, start_i, pause_i>$ to $PP_{out}$
        update $pause_{i-1}$ in $PP_{out}$
        //update destination position
        $s_{dest} \leftarrow$ min(pause start times later than $start_i$ in $PP_{in}$)
        $p_{dest} \leftarrow$ a pause position with $s_{dest}$ in $PP_{in}$
    until $start_i >$ max(pause start times in $PP_{in}$)
until $start_i \geq t_{end}$
return $PP_{out}$
```

as those of a real user. For example, when the next pause position is located near the current one and the time interval between their pause start times is very long, the dummy needs to stay at either of the pause positions longer than the maximum pause time for the user, which never happens in a user's movements (i.e., unnatural movements).

Our method makes sets of *mid-pause positions* and *mid-pause start times* as additional pause positions and times on the way to the next pause position (base or shared pause position) based on a *reachable area circle* to solve this problem and ensure consistency of movement, as Fig. 7 shows. The dummy necessarily arrives at each of the mid-pause positions at its mid-pause start time, and pauses for a certain time. By doing so, the dummy's movements become more natural. The reachable area circle is defined as the circle around the dummy's next pause position, whose radius is the maximum reachable distance from its current pause position until the start time for the next pause.

Algorithm 3 shows the algorithm for determining the dummy's movements based on the base pause position and shared pause positions.

More specifically, our method first determines the *first position* of the dummy in the process, where it is initially generated, within the reachable area circle around the pause position with the earliest pause start time (the first *destination position*) of all the determined pause positions (the base and shared pause positions). Then, if the dummy takes too long to reach the destination position, as Fig. 7(a) shows, our method randomly determines a mid-pause position (and mid-pause start time) within the reachable area circle around the destination position, which is reachable and closer from its first position. If the dummy arrives at the mid-pause position and still takes too long, it determines the next one within the reachable area circle around the destination position, which is reachable and closer to the current mid-pause position. By doing so, the dummy gradually gets closer to the destination position while retaining natural movements. Even if there are no possible positions that the dummy can pause at within the reachable area circle, our method does not choose a position where the dummy cannot arrive at the destination position on or before its pause start time after that position (see Fig. 7(b)). The destination position in this case is determined as the next pause position after the current mid-pause position. Based on these determined mid-pause positions, our method determines the dummy's movements from the first position to the pause position with the earliest pause start time. After the dummy arrives at the first destination position, the base or shared pause position with the next earliest pause start time is chosen as the next destination position. By repeating this process, our method determines the dummy's movements that pass all the base and shared pause positions.

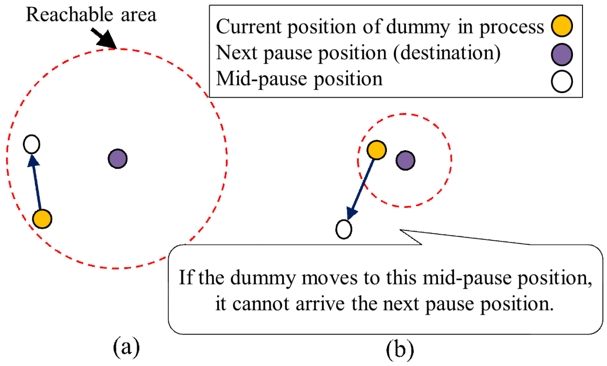
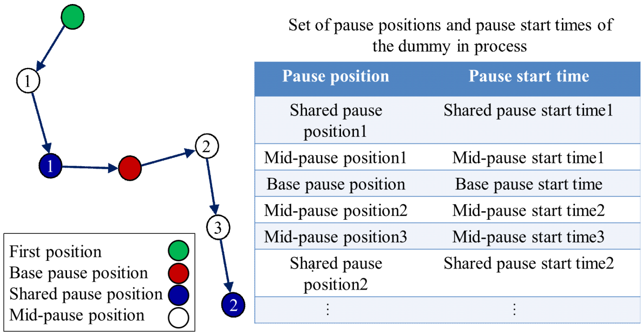

Reachable area
Current position of dummy in process
Next pause position (destination)
Mid-pause position
If the dummy moves to this mid-pause position, it cannot arrive the next pause position.
(a)
(b)

Figure 7: Determining the dummy's movement based on the reachable area circle



Pause position	Pause start time
Shared pause position1	Shared pause start time1
Mid-pause position1	Mid-pause start time1
Base pause position	Base pause start time
Mid-pause position2	Mid-pause start time2
Mid-pause position3	Mid-pause start time3
Shared pause position2	Shared pause start time2
⋮	⋮

Figure 8: Example of determining dummy's movement

Briefly, the dummy moves while stopping at three types of pause positions: the base pause position, shared pause positions, and mid-pause positions. Fig. 8 outlines an example of how a dummy's movements are determined.

4. EVALUATION

We aimed to evaluate the robustness of our proposed method from a quantitative perspective (i.e., statistical analysis) in this experiment. Since there are no existing studies (except for ours) on

Table 1: Parameters used in experiment

Parameter	Value
Service cycle [s]	180
Moving speed [m/s]	1.30
Area size [m^2]	15200^2
Number of dummies	16, 25
Maximum pause time [s]	600
Minimum pause time [s]	60
Anonymous area requirement [m^2]	1000^2, 1100^2,..., 2000^2

location privacy protection that have assumed a real environment, it was difficult to directly compare our method with other existing approaches. Therefore, we compared our method with our previous method [17].

4.1 Setting for Evaluation

We simulated people's movements in Kyoto, Japan using a network simulator MobiREAL [2], which moved while randomly stopping at several positions. We set positions where a user and dummies could stop every 50 [m] along roads. Table 1 summarizes the parameters and the values we used in our evaluation.

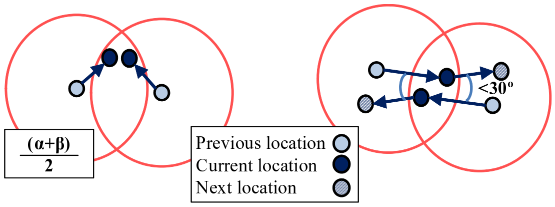
The number of dummies is an important aspect of our method. For example, a larger number of dummies reduces the time for the user and dummies to cross, i.e., it is easier to reduce traceability. However, since mobile devices are basically constrained in terms of resources such as network bandwidth and users generally want to reduce service usage costs (more dummies mean more service requests and greater usage costs), the number of dummies should be reduced as much as possible, i.e., to a minimum number that ensures the user's privacy requirements. The optimal number depends on the situation, which cannot generally be known. Thus, we evaluated two different numbers of dummies, N = 16 and 25 (including the user), in our evaluation.

4.2 Evaluation Metrics

We used three evaluation criteria to measure the performance of each method in terms of satisfaction with anonymity and reduced traceability.

- Anonymous Area Achieving Ratio-Count (AAAR-Count) This metric was aimed at measuring the satisfaction rate for anonymous area requirements. We specifically counted the number of service requests where the size of the anonymous area, which is the minimum convex covering all locations of dummies and the user, satisfied the size of the required anonymous area for all service requests issued during the simulation time. The AAAR-Count is defined as the ratio of the number of satisfied service requests to all service requests. Thus, the AAAR-Count reaches 100% when the sizes of required anonymous areas are satisfied in all service requests.

- Anonymous Area Achieving Ratio-Size (AAAR-Size) This metric was also aimed at measuring the satisfaction rate of anonymous area requirements. We specifically calculated the average size of anonymous areas for all service requests issued during the simulation time. The AAAR-Size is defined as the ratio of the average size of the anonymous area to the size of the required anonymous area. Thus, the AAAR-Size exceeds 100% when the average size of anonymous areas is larger than the requirements.



(a) The trajectories of user and dummies cannot be identified.

(b) The trajectories of user and dummies can be identified.

Figure 9: Transition of probability to being the user

- Mean Time to Confusion (MTC)

  Each of the queried locations had a probability of being the user's location. When the user's location is accidentally identified, e.g., by a report of a sighting in a real environment, the probability of the location being the user's is one. We defined the stochastic transition of the possibility for each location as follows.

  As Fig. 9(a) shows, when a location with the probability of being user's location α and another location with probability β are located in an area that can be reached from both of their previous locations, these two locations cannot be distinguished. The probability of both locations being the user's is $(\alpha + \beta)/2$ in this situation. We used the Mean Time to Confusion (MTC) defined in Shokri et al. [16] to measure traceability in our evaluation. MTC is defined as the mean time that is necessary to anonymize the user's location from an accidental disclosure by the LBS provider. Every time a service request is issued, we calculate the entropy of the probability of it being the user's location by $H = -\sum_{i \in \mathbf{D}} p_i \log p_i$, where p_i is the probability of location i being the user's location and D is the set of all locations corresponding to the user and all dummies. We assumed that the user's location would sometimes be divulged by the LBS provider in our evaluation. We defined MTC as the mean time period from the time when H becomes zero (when the user's location is revealed) to the time when H exceeds one (when we can regard the user's location as being anonymized). Smaller MTC means lower user traceability.

  Here, we did not lower the probability of a location being the user's if dummies (and the user) encountered one another from opposite directions on a road and approximately went straight (i.e., did not change directions much), as Fig 9(b) shows. Their routes in this case can be easily identified because returning after an encounter is unnatural for the user while going straight is more natural.

4.3 Comparison Methods

We compared three methods in this evaluation.

- Previous method:
  The method proposed by Suzuki et al. [17] assumed that the user's movements could not be predicted. This method is used to arrange dummies around the user in a grid to react to user's movements to achieve the size of the required anonymous area. It also makes dummies cross paths with the user to reduce the traceability of the user's location. It should be noted that dummies in this method cannot naturally pause like the user but do pause at unnatural positions where the user does not pause. Thus, it is easy to distinguish the user from dummies by visual observation. However, we ignored this problem in this evaluation.

[2] http://www.mobireal.net

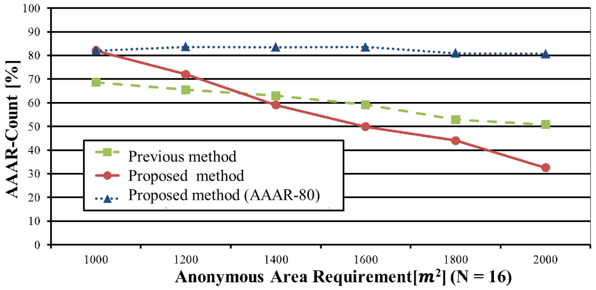


Figure 10: Anonymous Area Achieving Ratio-Count (AAAR-Count) (N = 16)

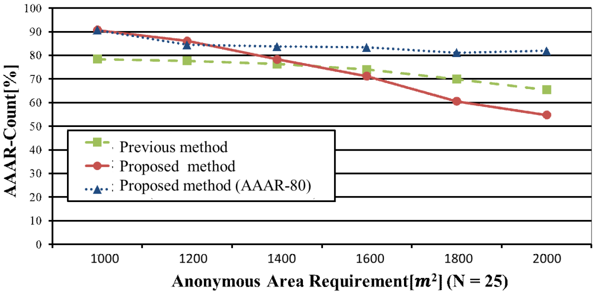


Figure 11: Anonymous Area Achieving Ratio-Count (AAAR-Count) (N = 25)

- Proposed method:
  The method proposed in this paper generates dummies that move naturally while stopping at pause positions based on the plans of a known user's movements.

- Proposed method (AAAR-80):
  Our method had larger size requirements for the anonymous area as input to achieve 80% of the AAAR-Count. We found that our method had unacceptably low AAAR-Counts in some situations through some preliminary experiments. We need to set larger size requirements than necessary for the anonymous area (as input) to avoid this. We found cases in our method where we changed the size requirements for the anonymous area appropriately to achieve AAAR-Counts of not less than 80% in all the simulation settings. We could verify our method was effective while sufficiently satisfying the size required for the anonymous area.

4.4 Evaluation Results

4.4.1 Anonymous Area Achieving Ratio (AAAR)

Figs. 10, 11, 12, and 13 plot the AAAR-Counts and AAAR-Sizes with various anonymous area requirements and two different numbers of dummies (N = 16 and 25). The results indicate that as the anonymous area requirements get larger, both the AAAR-Counts and the AAAR-Sizes of the proposed and previous methods get smaller. This has a negative impact on the process to reduce traceability from the AAAR-Counts and AAAR-Sizes, which occasionally shrinks the anonymous area where the user and dummies cross. Therefore, the size requirement for the anonymous area as input should be set larger than the user's requirement to ensure an anonymous area requirement tht has higher probability.

When the anonymous area requirement is small (smaller than 1200^2 $[m^2]$ for the AAAR-Count and 1400^2 $[m^2]$ for the AAAR-Size where N = 16, and smaller than 1400^2 $[m^2]$ for the AAAR-Count and 1600^2 $[m^2]$ for the AAAR-Size where N = 25), both the AAAR-Count and AAAR-Size obtained with the proposed approach are larger than those with the previous method. When the

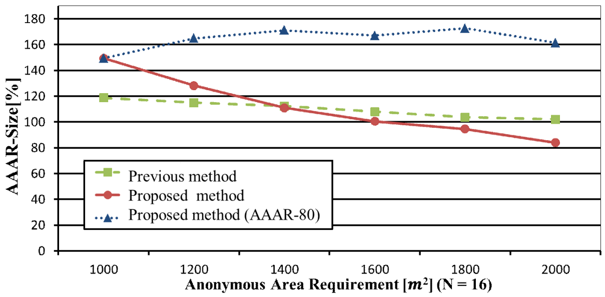


Figure 12: Anonymous Area Achieving Ratio-Size (AAAR-Size) (N = 16)

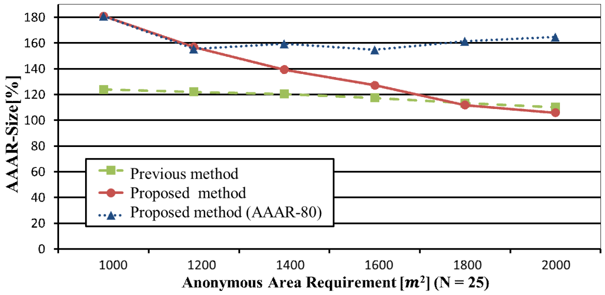


Figure 13: Anonymous Area Achieving Ratio-Size (AAAR-Size) (N = 25)

anonymous area requirement is small for the previous method, the size of the grid around the user also becomes small and dummies are gathered in a small area. When the positions of dummies are changed in this situation from their ideal locations in a grid due to geographical restrictions such as the topology of the road network, the anonymous area requirement often cannot be satisfied. However, since the proposed method determines dummies' movements in advance (not reactively) based on the plans of the known user's movements, it is not affected by such geographical restrictions and can satisfy the anonymous area requirement with a high degree of probability. Therefore, when the anonymous area requirement is small, the proposed method achieves a larger anonymous area than that obtained with the previous method.

When the anonymous area requirement is large (larger than 1400^2 $[m^2]$ for the AAAR-Count and 1400^2 $[m^2]$ for the AAAR-Size where N = 16, and larger than 1600^2 $[m^2]$ for the AAAR-Count and 1800^2 $[m^2]$ for the AAAR-Size where N = 25), both the AAAR-Count and AAAR-Size obtained with the proposed method are smaller than those with the previous method. This demonstrates the effectiveness of arraying dummies around the user in a grid with the previous method. The proposed method determines as many shared positions as possible, on the other hand, to make the user and dummies cross each other's paths to reduce traceability. This shrinks the anonymous area, and it thus becomes difficult to satisfy the anonymous area requirement.

Both the AAAR-Count and AAAR-Size with the proposed and previous methods where N = 25 are larger than those where N = 16 for all cases of anonymous area requirements. The AAAR-Size and AAAR-Count with the proposed method where N = 25 are 16.6% and 24.8% larger than those where N = 16 on average. This is because when there are many dummies, the anonymous area can easily increase.

Here, we will discuss our investigations into the anonymous area size requirement as input to achieve 80% of the AAAR-Count for each of the required anonymous areas with the proposed method. Table 2 summarizes the results. The proposed method (AAAR-80) in Figs. 10, 11, 12, and 13 show the results when using the

Table 2: Anonymous area size as input to achieve 80% of AAAR-Count

required anonymous area $[m^2]$	anonymous area as input $(N = 16)$ $[m^2]$	anonymous area as input $(N = 25)$ $[m^2]$
1000^2	1000^2	1000^2
1200^2	1500^2	1200^2
1400^2	2000^2	1600^2
1600^2	2300^2	1900^2
1800^2	3800^2	2800^2
2000^2	6200^2	4500^2

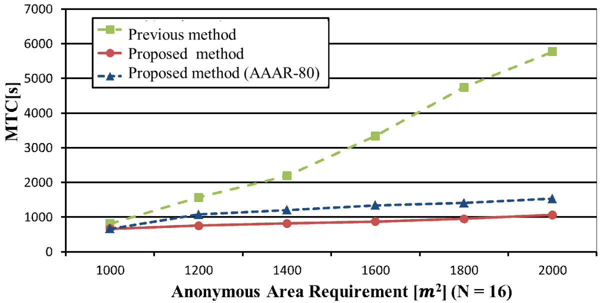


Figure 14: Mean Time to Confusion (MTC) (N = 16)

values in the second and third row of Table 2 as inputs for the required anonymous area. As we can see from the table, the degree (ratio) of increase for the size requirement of the anonymous area as input increases as the required anonymous area increases. For example, when the anonymous area requirements are $1000^2[m^2]$ and $2000^2[m^2]$ for N = 16, those as inputs are $1000^2[m^2]$ for the former and $6200^2[m^2]$ for the latter.

We set the moving speed of the user to $1.30[m/s]$ in this evaluation and the service cycle to $180[s]$. Thus, even when dummies kept walking during intervals between two successive service requests, they could only move $234[m]$. Therefore, it was actually difficult to constantly satisfy large anonymous area requirements such as $2000^2[m^2]$ while reducing MTC. Therefore, we needed to set it much larger than the required values as input.

The anonymous area size requirements as input where N = 25 were smaller than those where N = 16. For example, when the anonymous area requirement was $2000^2[m^2]$ for N = 25, the anonymous area size requirement as input was $4500^2[m^2]$, which is $1700^2[m^2]$ smaller than that where N = 16. This is because when there are many dummies, it is easy to increase the anonymous area, as was previously explained.

As Figs. 10, 11, 12, and 13 indicate, when setting larger anonymous area size requirements as input than the actual requirements (i.e., the proposed method (AAAR-80)), both the AAAR-Count and AAAR-Size with the proposed method (AAAR-80) are much larger than that with the proposed and previous methods. Thus, we could confirm that the proposed method (AAAR-80) could solve the problem of decreasing AAAR when there were large required anonymous areas.

4.4.2 Mean Time to Confusion (MTC)

Figs. 14 and 15 plot MTCs with various anonymous area requirements where N = 16 and 25. We can see from the results that as the anonymous area requirement decreases, the MTCs with the proposed and the previous methods decrease. This is because dummies are located closer to the user and the user's location is within dummies' reachable areas in most cases, i.e., there is basically no need for the user and dummies to intentionally cross one another's paths.

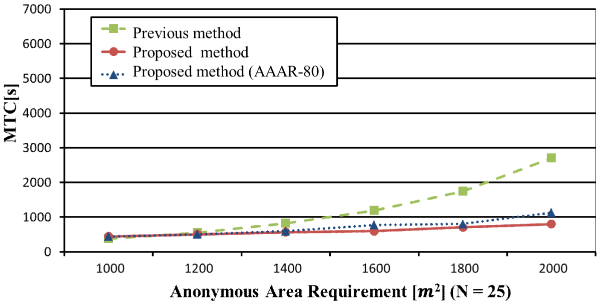


Figure 15: Mean Time to Confusion (MTC) (N = 25)

The MTCs with the proposed approach are much lower than those with the previous method for all cases of anonymous area requirements and numbers of dummies (N = 16 and 25). The differences in MTCs for the proposed and previous methods increases as the anonymous area requirements increase. For example, when the anonymous area requirements are $1000^2[m^2]$ and $2000^2[m^2]$ for N = 16, the MTCs for the proposed method are 148 [s] and 4710 [s] lower than those for the previous method. This demonstrates the effectiveness of aggressively reducing traceability with the proposed method. It is difficult to make dummies naturally and frequently cross the user's path with the previous method, because the grid must be maintained to satisfy the anonymous area requirements. Actually, the previous method only makes the user and dummies cross one another's paths when some dummies have moved ahead of the user in the grid. However, the proposed method determines as many shared positions as possible to make dummies cross paths with the user (with higher priority than the anonymous area requirements). The directions of movement by the dummies also affect the differences in MTCs for the proposed and previous methods. Dummies specifically move in various directions with the proposed method while they tend to move in parallel with the user's direction of movement with the previous method. Therefore, the previous method provides fewer chances for dummies to enter the user's reachable area.

The MTCs obtained with the proposed method and those with the previous method where N = 25 were larger than those where N = 16 for all anonymous area requirements. The MTCs with the proposed method where N = 25 were 251 [s] lower on average than those where N = 16. This is because when there were many dummies, they were located closer to the user and the user's location was within the dummies' reachable areas in more cases, which increased the chances of the user crossing the dummies' paths.

The MTCs obtained with the proposed method (AAAR-80) were much lower than those with the previous method for all anonymous area requirements and numbers of dummies (N = 16 and 25). For example, when the anonymous area requirement was $2000^2[m^2]$, the MTCs with the proposed method (AAAR-80) where N = 16 and N = 25 were 4240 [s] lower for the former and 1905 [s] lower for the latter than that with the previous method. The differences in MTCs increased as the required anonymous areas increased. The MTCs with the proposed method (AAAR-80) were slightly higher than those with the proposed method, and the differences increased as the required anonymous areas increased. For example, when the anonymous area requirement was $2000^2[m^2]$, the MTCs with the proposed method (AAAR-80) where N = 16 and N = 25 were 468 [s] and 329 [s] lower than those with the proposed method, respectively. We could confirm from these results that even when larger anonymous area requirements were set as inputs than the actual requirements, the proposed method achieved much lower traceability (i.e., MTCs) than that with the previous method. We could also confirm that by enlarging the anonymous area requirements as inputs, the proposed method could drastically improve AAAR while slightly sacrificing MTCs.

Finally, when the anonymous area was large, the MTCs with the proposed method where N = 16 were lower than those with the previous method where N = 25. Specifically when the anonymous area requirement was $2000^2[m^2]$, the MTCs with the proposed method were 1644 [s] lower than those with the previous method. This demonstrates that even when there were few dummies, the proposed method achieved lower traceability than that with the previous method. Therefore, the proposed method could decrease the service usage cost for dummies while maintaining few MTCs.

5. CONCLUSIONS

We proposed a method of anonymizing a user's location based on his/her movements with pauses when using LBSs with mobile devices. The proposed approach generated dummies that moved naturally while stopping at several locations like the user; the dummies also took into consideration geographical restrictions. It simulated the dummies' movements naturally so that no LBS provider could distinguish the actual user from dummies. It made the dummies stop at locations where there were fewer dummies to anonymize the user's location. Our method made the user and dummies cross one another's paths at locations where they paused.

We simulated the user's movements on a real map and verified our proposed method was more effective than our previous one [17]. As a result, the proposed approach decreased traceability while it made dummies stop at several more locations than the previous method did. When the anonymous area requirements were larger than $1400^2[m^2]$, the AAAR-Size and AAAR-Count with the proposed method were smaller than those with the previous method. However, by setting larger anonymous area requirements as inputs, the proposed method achieved larger AAAR-Sizes than those with the previous method. Furthermore, increasing the number of dummies also effectively satisfied the required anonymous areas.

We plan to conduct user experiments to evaluate the robustness of our method with humans as part of future work. Additionally, we plan to extend our method to make dummies react naturally to unpredictable user movements to make our approach more realistic. For example, when a user reacts to a push-service such as location-based advertisements or a traffic-jam warning, he/she may go to the advertised locations of interest or may take detours to avoid traffic jams. In such a case, dummies should also change their movement based on the user behavior.

6. ACKNOWLEDGEMENT

This work was partially supported by the Microsoft Research Core Project, and Grant-in-Aid for Exploratory Research (24650038) and Grant-in-Aid for for Scientific Research (B)(24300037) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

7. REFERENCES

[1] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P.Samarati, “Location Privacy Protection through Obfuscation-Based Techniques,” In Proc. DBSec, pp 47–60, 2007.

[2] A. R. Beresford and F. Stajano, “Location Privacy in Pervasive Computing,” In Proc. Pervasive, pp. 46–55, 2003.

[3] C. Bettini, S. Mascetti, X. S. Wang, and S. Jajodia, “Anonymity in Location-Based Services, Towards a General Framework,” In Proc. MDM, pp. 69–76, 2007.

[4] R. W. Bohannon, “Comfortable and Maximal Walking Speed of Adults Aged 20-79 Years: Reference Values and Determinants,” In: Age Ageing, pp. 12–19, 1997.

[5] L. Busic, and R. Filjar, “The Role of Position Reporting Frequency in LBS Qos Establishment,” In Proc. SoftCOM, pp. 209–213, 2006.

[6] R. Cheng, Y. Zhang, E. Bertono, and S. Prabhakar, “Preserving User Location Privacy in Mobile Data Management Infrastructures,” In Proc. PET, pp. 393–412, 2006.

[7] C. Y. Chow, M. F. Mokbel, and X.Liu, “A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-Based Services,” In Proc. GIS, pp. 171–178, 2006.

[8] M. Duckham and L. Kulik, “A Formal Model of Obfuscation and Negotiation for Location Privacy,” In Proc. PERVASIVE, pp. 152–170, 2005.

[9] B.Gedik and L. Liu, “Location Privacy in Mobile Systems: A Personalized Anonymization Model,” In Proc. ICDCS, pp. 620–629, 2005.

[10] M. Gruteser and D. Grunwald, “Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking,” In Proc. ISENIX MobiSys, 2003.

[11] A. Khoshgozaran, and C. Shahabi, “Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy,” In Proc. SSTD, pp. 239–257 2007.

[12] H. Kido, Y. Yanagisawa, and T. Satoh, “An Anonymous Communication Technique using Dummies for Location-Based Service,” In Proc. ICPS, pp. 88–97, 2005.

[13] J. Krumm, “Realistic Driving Trips for Location Privacy,” In Proc. PERVASIVE, pp. 25–41 2009.

[14] H. Lu, C. S. Jensen, and M. L. Yiu, “PAD: Privacy-Area Aware, Dummy-Based Location Privacy in Mobile Services,” In Proc. Int'l Workshop on Data Engineering for Wireless and Mobile Access, pp. 16–23, 2008.

[15] M. Masanori, I. Yoshiharu, “Anonymizing User Location and Profile Information for Privacy-aware Mobile Services,” In Proc. GIS-LBSN, pp. 69–75, 2010.

[16] R. Shokri, J. Freudiger, M. Jadliwala, and J. P. Hubaux, “A Distortion-Based Metric for Location Privacy,” In Proc. WPES, p. 6, 2009.

[17] A. Suzuki, M. Iwata, Y. Arase, T. Hara, X. Xie, S. Nishio, “A User Location Anonymization Method for Location based Services in a Real Environment,” In Proc. GIS, pp. 398–401, 2010.

[18] Y. Yanagisawa, H. Kido, and T. Satoh, “Location Traceability of Users in Location-Based Services,” In Proc. Int'l. Conf. on Mobile and Ubiquitous Computing, 2006.

[19] L. Yao, G. Wu, J. Wang, F. Xia, C. Lin, G. Wang, “A Clustering KAnonymity Scheme for Location Privacy Preservation,” In Journal on Trust, Security and Privacy in Computing and Communication Systems, Vol. 95-D, No. 1, pp. 134–142, 2012.