Parasitic Location Logging: Estimating Users' Location from Context of Passersby

Kota Tsubouchi
Yahoo Japan Corporation
Tokyo, JAPAN

Teruhiko Teraoka
Yahoo Japan Corporation
Tokyo, JAPAN

Hidehito Gomi
Yahoo Japan Corporation
Tokyo, JAPAN

Masamichi Shimosaka
Tokyo Institute of Technology
Tokyo, JAPAN

***Abstract*—People often turn off location logging when the batteries of their smartphones get low, to reduce the phone's power consumption and prolong its operation. Here, we propose an innovative data sharing scheme called as the Parasitic Location Logging (PLL). PLL can acquire location of such users, what we call parasitic users, without invoking any location functionalities by the GPS and Bluetooth low energy (BLE) sensors of their smartphones. PLL estimates parasitic users' location and trajectory by relying on other users who pass by the parasitic user, what we call host users, as evidence that they are located in close proximity. The results of field experiments showed that PLL dramatically decreases battery consumption of parasitic users' smartphones and that the position of parasitic users can be identified accurately. Moreover, the battery consumption of PLL was rigorously evaluated in a laboratory setting to demonstrate its benefit. An agent simulation evaluating the proposed calculation algorithm under various conditions in realistic environments validated the robustness of PLL.**

***Index Terms*—Parasitic Location Logging, location data completion, heuristic search**

I. Introduction

With the widespread use of smartphones, continuously acquired location information can provide users with benefits in various applications in ubiquitous computing applications. For instance, government organizations, urban designers, and area marketers can statistically analyze location information collected from many users to estimate the number of people staying or moving in a certain area (i.e., urban dynamics) [1], or model user movement data to identify their preferences or patterns in their daily activities [2]. Such analyses can then be used for city planning, marketing, etc. [3].

Moreover, users can post their location histories on their blogs and social media sites to maintain their life logs as alternatives to their diaries. Users can browse their location information that has been uploaded to a server and review the history of their activities, such as where they visited on a specific day (e.g., Google Maps TimeLine Service[1]). The collected location information can also be used for security and safety purposes, e.g., to recommend evacuation routes for people in a natural disaster or to discover a child or old person who has wandered away from home.

However, when the battery of the smartphone is low, some operating systems and applications automatically reduce the frequency of location logging, and consequently, some users may choose to turn off the location logging itself in order to save battery energy. This causes their location dataset to have many missing locations. The missing data from such an incomplete dataset are hard to recover, and this leads to serious problems for certain applications.

This paper proposes Parasitic Location Logging (PLL), which is an innovative approach that continually collects users' location data with little battery consumption even after their location acquisition function has stopped. Despite its preliminary nature, we believe that it will lead to a future standard for efficiently constructing user location datasets in a large-scale in a distributed manner.

PLL estimates the locations of target users (we call them parasitic users) from location information generated by different users who pass by them (we call the passersby host users) as evidence that they are located in close proximity. The parasitic users are those whose smartphones have little remaining battery power, while the host users are those whose phones have enough battery power. The information on the locations at which they pass by is estimated from the host users' location information, in particular, the time at which they pass by each other. The advantage of PLL is that the parasitic user's location can be recovered without using any of the GPS or WiFi sensing capabilities of their phone even if the host users did not sense their location information just when they passed by parasitic users.

Opportunistic data sharing [4]–[6] is related to PLL. In ordinal opportunistic data sharing framework, a user acquires and shares location information acquired by the GPS sensor on the spot with other users. Each 'host' user acquires position information every time he or she passes by another user, and as the number of people passing each other increases, the number of times of their own position information is acquired also increases. The ordinary opportunistic location sharing method; however, drastically increases the battery consumption of the host users; thus, it is likely that even users with enough battery power would tend to avoid the host role. Moreover, both parasitic and host users have to turn on Bluetooth low energy (BLE) scanning in order to join opportunistic location data sharing, and the system detects users passing by from the BLE signal. Here, even BLE scanning poses a critical problem because the parasitic users would have little battery remaining.

Bluewave [6] is another opportunistic data sharing technique that allows devices to opportunistically share contexts when

[1]https://www.google.com/maps/timeline?pb

they are nearby. In Bluewave's scheme, a user can obtain his/her approximate location by scanning for and detecting a BLE signal from a neighbor user who uploads their updated location as a context broker. However, users of Bluewave face difficulty in estimating their location accurately, because Bluewave is a framework for sharing general contextual information, not specifically locations. It imposes a contextual restriction in which the context broker acquires and delivers the location information to users in close proximity and within a short time. That is, differences between the time of updating the location of a context broker and the time of passing by cause the location estimates to be inaccurate. PLL can provide accurate pass locations to parasitic users.

The contributions of this paper are as follows: 1) Parasitic Location Logging (PLL), which is able to acquire user location information while restricting the user's battery consumption, is proposed. 2) Demonstration experiments implementing the PLL scheme prove the feasibility of the location logging method. Moreover, the battery consumption of PLL is evaluated rigorously in laboratory free of ambient noise. 3) The calculation algorithm is evaluated under various conditions in an agent simulator. The evaluation indicates that the PLL calculation algorithm is robust to various environmental conditions and that the PLL calculation method is much more accurate than existing methods.

The rest of this paper is as follows. Section 2 summarizes the idea and implementation of PLL. Section 3 describes three experiments for demonstrating the feasibility of PLL and discusses their results in Section 4. Section 5 reviews related work. We conclude this paper in Section 6.

II. Parasitic Location Logging

A. Basic idea of Parasitic Location Logging

In PLL-based systems, users can play two roles: *parasitic* and *host* users who are respectively takers and givers of location information. A parasitic user is one whose smartphone has little battery power remaining and who obtains location information from the location logs of host users who pass by, instead of sensing any GPS, WiFi or Bluetooth signals with their own device. The users' roles are assumed to be set automatically and change dynamically depending on the amount of battery power remaining of their smartphones. Parasitic users can acquire their location without consuming much battery power.

B. Requirements of Parasitic Location Logging

The requirements for PLL system are considered from the viewpoints of the parasitic role and host role. From the viewpoint of the parasitic role, the PLL scheme has to greatly reduce the battery consumption of the parasitic user, even though locations should be continuously acquired. In an ordinary opportunistic data sharing scheme, although parasitic users do not have to acquire their location from GPS by themselves, they must scan the host users' smartphone and exchange data. Thus, the ordinary scheme still requires

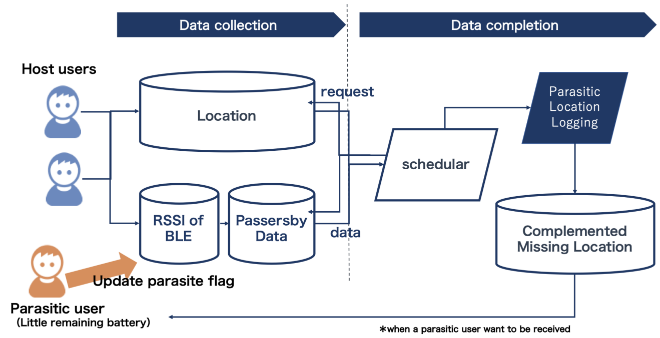


Fig. 1: Overview of Parasitic Location Logging

parasitic users to use their phones' battery to scan the host users' smartphone. The accuracy of the acquired location is also important: the more accurate the location acquisition is, the more benefit there is for the parasitic users. The locations acquired from Bluewave scheme are expected to have low accuracy, since the parasitic users acquire recently updated location of host users (not the location that they passed by one another).

The requirements imposed on the host users are that the battery consumption of their smartphones should not increase too much in playing the host role. In ordinary opportunistic data sharing scheme, host users collect their locations when passing by parasitic users and send the data to parasitic users. This taxes the battery of the host's smartphones and makes it unlikely that someone would want to play that role. On the other hand, in the Bluewave scheme, host users just scan for the BLE of the parasitic users' smartphone and send them their recently updated location later. The Bluewave scheme does not drastically increase the battery consumption of the host users, and thus, people would be more inclined to become host users in that scheme. The requirements stated above are challenging but can be met by the PLL scheme described here.

C. Data Processing of Parasitic Location Logging

As shown in Figure 1, PLL consists of two phases: data collection and data completion. When a parasitic user's smartphone has little battery power left (for example, less than 30%), the location logging application sends a signal to the server, which flips the "parasite flag" of the parasitic user in the database. GPS and BLE active-scanning functions of the parasitic user's smartphone are assumed to be turned off. After that, the parasitic user relies on information gathered by host users who pass by to complement his or her missing location data at the server. The parasitic user's application receives the complemented location data when he/she wants to receive them, such as when the smartphone is charged.

In the data collection phase, the location information of the host user as well as the passing time are gathered at a defined frequency. Then, the data accumulated by the host users are transmitted over the Internet to the server, where they are stored, at a low frequency, e.g., twice a day.

In the data completion phase, PLL estimates where users pass each other from information on who the host user and parasitic user are and when they passed each other. Note that only the host users scan the BLE signals of the parasitic users and share their GPS information, the parasitic user keeps their GPS and BLE active-scanning functions turned off.

PLL does not complement the parasitic user's location information with data from only one passerby. It finds a number of passing candidates from one user's passing information and narrows down the candidate points by using several different pieces of information collected when the user passed by. The candidate passing locations are determined from the difference between the most recent time that position information was acquired by the host user and time of the passing. Multiple candidate passing points are calculated from one piece of passing information, and realistic candidate passing points are selected using several pieces of passing information. The selected point is one that is reasonable in consideration of the above time difference. Repeating this process creates the position information of many users. Choosing one candidate is a challenging task. The parasitic user's origin, destination and even part ground-truth data are unknown, so we have little information for narrowing down the candidate points.

The position information of the parasitic users calculated by PLL is stored in the server and analyzed in the same way as ordinary position information. The parasitic users can download their missing location and confirm them in a location logging application, for instance.

D. Parasitic Location Logging Algorithm

In this section, our parasitic location logging algorithm is presented from the definition of variables used in the algorithm. Then we formulate how to infer parasitic location.

1) Problem setting: Prior to the formulation how to infer parasitic location from the location traces from the host users, the variables used in the algorithm are defined so as to clarify the problem setting of parasitic location logging. Let $\boldsymbol{G}^{(i)} = \{\boldsymbol{r}_j^{(i)}, t_j^{(i)}\}_{j=1}^{n_i}$ denote a location trace of i-th host user, where $\boldsymbol{r}_j^{(i)} \in \mathbb{R}^2$ depicts j-th location, i.e. longitude and latitude, in the trace, and $t_j^{(i)}$ represents its corresponding time stamp, respectively. In the system, we assume that whole collection of host user traces over u users are aggregated i.e. $\{\boldsymbol{G}^{(i)}\}_{i=1}^{u}$. Assuming that we could also obtain the parasitic logs without location information as $\boldsymbol{P}^{(k)} = \{h_l^{(k)}, \tau_l^{(k)}\}_{l=1}^{m_k}$, where $h_l^{(k)}$, and $\tau_l^{(k)}$ represent ID of the host user at l-th passing of the k-th parasitic user, and its time stamp, respectively. It should be noted that location information itself is not aggregated from the parasitic user smartphones, neither from the host user's smartphones. That implies that the system does not invoke functionality of GPS sensing i.e. the power consumption is not increased from the usual location logging.

2) Parasitic location logging as range based optimization: From the dataset of host users $\{\boldsymbol{G}^{(i)}\}_i^u$, and of parasitic users $\{\boldsymbol{P}^{(k)}\}_{k=1}^{u'}$, the system recovers the location information of k-th parasitic user $\hat{\boldsymbol{q}}_l^{(k)}$, then stores them into $\tilde{\boldsymbol{P}}^{(k)} =$

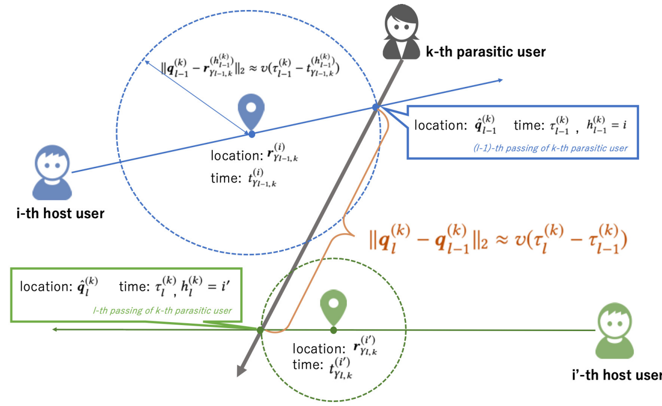


Fig. 2: Situation of Parasitic Location Logging Calculation

$\{\hat{\boldsymbol{q}}_l^{(k)}, h_l^{(k)}, \tau_l^{(k)}\}_{l=1}^{m_k}$. In other words, the location of parasitic users could be inferred per the passing-by usual users.

For simplicity, we employ range based algorithm where distance between two types of anchor information are used, and assume that walking velocity of whole users including host and parasitic users is constant as v. From (2) with constant velocity information while walking, the following two equations related to distance could be derived as follows:

$$\|\boldsymbol{q}_l^{(k)} - \boldsymbol{q}_{l-1}^{(k)}\|_2 \approx v(\tau_l^{(k)} - \tau_{l-1}^{(k)}) \tag{1}$$

$$\|\boldsymbol{q}_l^{(k)} - \boldsymbol{r}_{\gamma_{l,k}}^{(h_l^{(k)})}\|_2 \approx v(\tau_l^{(k)} - t_{\gamma_{l,k}}^{(h_l^{(k)})}), \tag{2}$$

where $\gamma_{l,k}$ depicts the latest time stamp ID of $h_l^{(k)}$-th host user at l-th passing of k-th parasitic user. In other words, $\gamma_{l,k}$ is defined as

$$\gamma_{l,k} = \max_j\{j | t_j^{(h_l^{(k)})} \leq \tau_l^{(k)}\}. \tag{3}$$

From these range information, we employ the following constraint least square optimization in the system:

$$\begin{aligned} &\min_{\boldsymbol{q}_{1:m_k}^{(k)}} \frac{1}{2} \sum_{l=2}^{m_k} \left(\|\boldsymbol{q}_l^{(k)} - \boldsymbol{q}_{l-1}^{(k)}\|_2 - v(\tau_l^{(k)} - \tau_{l-1}^{(k)}) \right)^2, \\ &\text{s.t. } \|\boldsymbol{q}_l^{(k)} - \boldsymbol{r}_{\gamma_{l,k}}^{(h_l^{(k)})}\|_2 = v(\tau_l^{(k)} - t_{\gamma_{l,k}}^{(h_l^{(k)})}),\ l = 1, \ldots, m_k \end{aligned} \tag{4}$$

It should be noted that the range information of host users at l-th passing by k-th parasitic user is employed as constraint in the optimization while the range of parasitic user between l-th and $l-1$-th passing information is used as objective.

Due to the nature of the range constraint, the above optimization problem could be much simplified. Specifically, we also employ alternative representation towards bound constraint minimization as follows. From the geometric constraint, we could represent the location of parasitic user $\boldsymbol{q}_l^{(k)}$ as

$$\boldsymbol{q}_l^{(k)} = \boldsymbol{r}^{(h_l^{(k)})} + v(\tau_l^{(k)} - t_{\gamma_{l,k}}^{(h_l^{(k)})}) \begin{pmatrix} \cos\theta_l^{(k)} \\ \sin\theta_l^{(k)} \end{pmatrix}. \tag{5}$$

With this representation, the above constraint linear least square problem could be projected into a bound constraint nonlinear least square problem $\theta^{(l)}_{1:m_k}$, where each angle $\theta^{(k)}_l$ ranges from 0 to 2π.

For further improvement of location accuracy, we employ a heuristic approach to randomly choose the passing by information of k-th parasitic user instead of using all the passes of this user from our experience. Specifically, the system repeatedly executes the optimization process through random sampling and accumulates each result to the priority queues, which is akin to beam search method. Finally, the system picks up the best from the candidates.

E. Parasitic Location Logging System

Figure 1 illustrates PLL implemented as a system or an application in which passes are detected using Bluetooth (BLE). Note that there is no restriction on the means of detection, and Bluetooth is only an example. Also, although the position information is updated at a timing decided by the scheduler, it is possible for PLL to operate in real time by performing the calculation as soon as the position information is collected; however, the parasitic user will not be able to download the complemented location data in real time since the battery of his or her smartphone is assumed to be low. Parasitic users are assumed to download their complemented location as they want, such as when they charge their smartphone. Establishing an Internet connection consumes a lot of battery power. For this reason, the position information of each host user is acquired at a low frequency, for example, once every 30 minutes or longer.

The host user's Bluetooth information is collected as a list. In particular, we assume that the list of service set identifiers (SSID) is sent to the server at certain intervals from the host users' smartphones. It is possible to send the information to the server every time the user passes by someone, but this would consume a large amount of battery energy. Instead, the SSID of each passerby is temporarily stored in the smartphone and sent to the server together with the location information acquired at a low frequency. In the server, the SSID and the ID of the position information are linked, so that it is possible to know what the persons passed by and the latest position information of the passersby.

The scheduler on the server runs the PLL calculation at regular intervals on the collected location information of host users and passing information between host users and parasitic users.

In the PLL system, a host device uploads sensed data and a parasitic device downloads the estimated location information when it has enough battery power.

III. Experiments on Parasitic Location Logging

We evaluated PLL in three experiments. The first was a field experiment aimed at determining whether the overall system works well in a realistic situation (See Section III-A). The second experiment was an evaluation in a laboratory environment (See Section III-B). We measured the battery consumption accurately in a laboratory setting. The results helped us to understand the benefits of PLL. The third experiment was a simulation (See Section III-C). We implemented an agent simulator to validate PLL in various situations.

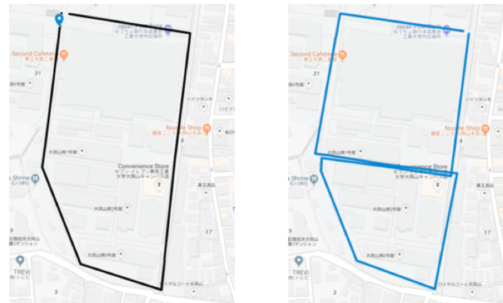


Fig. 3: Two routes of the experiment

A. Experiment 1: Total Performance in Field Experiment

1) Experimental Overview: To demonstrate the feasibility of PLL, experiments were conducted at a university campus. Eleven participants, including four parasitic users and seven host users, walked around carrying smartphones of the same type (Google Nexus 5X) and GPS loggers. A PLL app was installed on the smartphone of each host participant to periodically sense nearby Bluetooth devices. The GPS logger recorded GPS coordinates and timestamps every 5 seconds in order to collect the ground-truth location information about the participants.

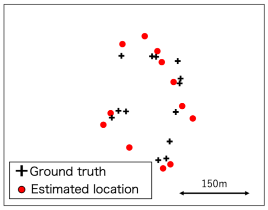
Each participant followed one of a given set of routes starting from different positions and directions on the campus. The lengths of the routes were about 800 m and 1100 m (see Figure 3). Two parasitic participants (user A and D) walked a simple route around buildings, as shown on the left side of Figure 3. The others (user B and C) walked a route that included a pathway between two buildings, as shown on the right side of the figure. Five twenty-minute trials of walking a given route were conducted. Each parasitic participant encountered the other participants several times on the route in each trial. In order to compare battery consumptions, three users had one more smartphone which did not use PLL (ordinary users).

2) Parameter Settings of Application in Field Experiment: Table I lists the parameters for the smartphones of each type of user in the field experiment. Parasitic users turned on only BLE sensors and turned off BLE active-scanning. They were discoverable by host users' BLE active-scanning. Host users and ordinary users turned on their all sensors (BLE, GPS, and WiFi). The smartphones of the host users scanned for GPS signals about every 300 seconds and BLE signals about every 60 seconds. The ordinary users only scanned for GPS signals at the same frequency as that of the host users. Baseline statuses were prepared for the comparison, and all sensors were turned off in the baseline setting.

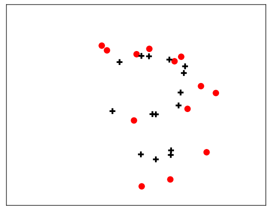
Nearby users were able to be identified from the sensed Bluetooth device information, and the time of each encounter with a passerby was determined from the timestamp when

TABLE I: Definition of smartphone setting

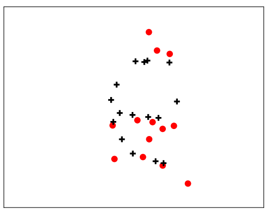
	BLE	GPS	WiFi	LTE
baseline	Off	Off	Off	On
parasitic users	discoverable only	Off	Off	On
host users	scan (60 sec.)	scan (300 sec.)	On	On
ordinary users	discoverable only	scan (300 sec.)	On	On



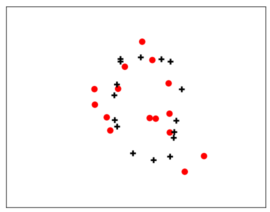
(a) User A: AE = 13.8 (m)



(b) User B: AE = 16.6 (m)



(c) User C: AE = 18.3 (m)



(d) User D: AE = 28.1 (m)

Fig. 4: Result of estimation of four parasitic users' location from seven host users. Average error(AE) is shown in each caption. The path of user A and D are in the left, that of user B and C are in the right in Figure 3).

the strongest received signal strength indication (RSSI) of the user's BLE was detected. The RSSI threshold to detect a passerby was set at -90 dBm or greater.

In the PLL calculation for estimating the parasitic users' locations, we conducted 5 million heuristic searches in the PLL calculation for this experiment. Repeated calculations lead to a more accurate route estimation, but it also increases the calculation time.

3) Ground Truth Data Collection: In addition to the above sensor data, ground truth data were recorded for the performance evaluation. When a subject passed any of the other subjects, he or she recorded the latitude and longitude of passing point and the passing time in seconds of the passing point manually. The recorded passing position was taken to be the correct position (i.e. ground truth data).

4) Experimental Results: Figures 4a, 4b, 4c, and 4d show the results of PLL for four of the parasitic participants. In each case, the position information was completed using PLL, i.e., without using any the position information that the parasitic users acquired from their GPS sensors. The black points (marked as plus) indicate the ground-truth passing positions, and the red points (marked as circle) indicate the passing points estimated by PLL. The results show that almost all of the users' movements were accurately reproduced.

The average difference (error) between the estimated passing position (red) and the actual ground truth (black) are shown in the captions of each figure. Although the error varies somewhat among users, the average value is about 20 meters. Assuming that the user walks at 1.3 m/s and that the GPS data are collected at 5-minute intervals, one point of position information was acquired approximately every 400 m. The average error of 20m yields an estimation error (error between the estimated passing point and the correct passing point in the ground truth data) of about 5.0%, meaning that PLL accurately completed the user's position information. Note that PLL could complete these passing locations accurately **without** using any of the parasitic user's location information at all.

5) Battery Consumption in the Field Experiment: We checked the battery consumption of PLL users with those of ordinary users, who were not PLL users and who scanned their locations by themselves.

The battery consumption of the parasitic users' smartphones was **0.67 times** that of the ordinary users on average, while the battery consumption of the host users was **1.03 times** that of the ordinary users. Compared with the ordinary users, the increase in the battery consumption of the host users was very small (about +3%), whereas the decrease in the battery consumption of the parasitic users was rather large (about -33%). In terms of the whole system, when the number of host users per parasitic user was less than 11, their total battery consumption was reduced.

B. Experiment 2: Battery Consumption Test in Laboratory Free of Ambient Noise

1) Experiment Overview: The battery consumption is affected by noise due to congestion situation and by setting the scanning sensors frequency. To get an accurate estimate of the battery power consumed, we conducted a rigorous experiment in an ambient-noise-free laboratory environment.

2) Experimental Conditions: Uncontrolled environments (like the university campus) are often filled with noise from the BLE and Wifi signals of numerous other smartphones. Such signals should be reduced if the battery power consumption is to be accurately evaluated. Hence, we performed an experiment in a laboratory free from such ambient noise.

Figure 5 shows a photo of the laboratory environment. The laboratory environment could receive LTE radio waves and GPS signals and had no devices other than the tested ones that could send BLE and Wifi signals. Previous research has indicated typical living and working situations have 5 BLE signals on average in (20 BLE signals in crowded situations). We installed beacons (EddyStone [2]) that represented a crowd of people carrying smartphones.

We installed only the test application on new Android smartphones (Pixel 3). We used developer mode to turn off the smart battery-saving modules, which automatically controls the data acquisition frequency and reduced battery consumption. The test application changed the acquisition frequency of the BLE scan and GPS signal reception and recorded the battery level every 60 seconds.

[2]https://github.com/google/eddystone

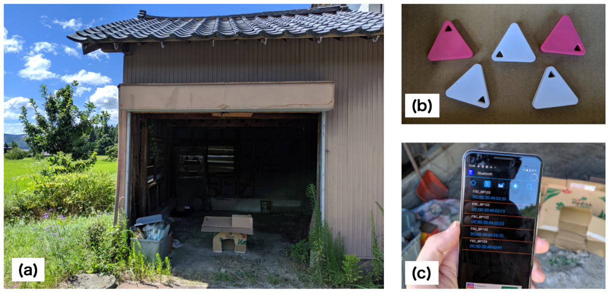


Fig. 5: Laboratory environment: (a) scenery of the experiment's location; (b) beacons representing a crowd carrying smartphones; (c) there were no BLE devices other than the tested ones.

The test application could change the parameter settings of the smartphones to simulate four roles: parasitic, host, ordinary, and baseline. The parameter settings of the parasitic, host, and ordinary roles are listed in Table I: baseline means all sensors were turned off. The *parasitic role* represented the parasitic user; in this case, BLE and LTE were turned on and GPS and Wifi were turned off. The *host role* simulated the host user; all of the sensors were turned on. The *ordinary role* simulated users who do not play a PLL role; GPS scan was turned on for acquiring their own positions, while BLE scan was turned off (in discoverable mode). The *baseline role* only recorded battery consumption every 60 seconds; all sensors were turned off. We do not consider the battery consumption for recording the status, so we compared all of the other roles with the baseline.

We put the smartphones on a desk in the laboratory and measured their batteries over the course of 10 hours.

3) Battery Consumption Measurement: We used the battery discharge ratio, which is the decrease in battery voltage per unit time, as a metric of battery consumption. Figure 6 shows an example of the data. The x-axis shows the time passed, and the y-axis shows the voltage. We calculated a regression function of the data and took the angle of the regression line with the x-axis as the metric of the battery function. The Random Sample Consensus (RANSAC) algorithm was used to calculate the regression function. We chose ten points from the data at random and drew the line with the method of least squares. We repeated the RANSAC method 10000 times to get the angle of the regression function. Thus, the acquired battery discharge ratio means how much voltage decreased during a unit time.

4) Experimental Results: Table II shows the battery consumption for the same parameter settings as in the field experiment. The figure Table II denotes the discharge amount per 60 seconds. The smartphones of the host users in the field experiment scanned for the GPS signal every 300 seconds and scanned for BLE signals every 60 seconds. The ordinary users acquired their locations by GPS every 300 seconds.

The results did not show any statistically significant difference between the battery consumptions of the host role and ordinary role. On the other hand, the battery consumption of the parasitic role was about 1.6 times higher than that of the baseline, while that of the host role was about 2.4 times higher, and that of the ordinary role was 2.5 times higher. These results match those of the field experiment.

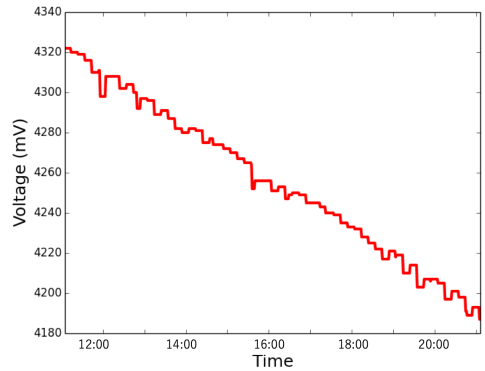


Fig. 6: Sample of battery discharge data

TABLE II: Battery discharge rate for each role in field experiment (ratio shows that the ratio against baseline)

	mean	std. error	ratio
baseline	0.0648	0.0175	1.0
ordinary (GPS:300 sec.)	0.164	0.0295	2.53
host (BLE: 60sec., GPS: 300sec.)	0.156	0.0286	2.407
parasitic	0.106	0.0159	1.63

Additionally, Figure 7 shows results for when the host users changed their data acquisition frequencies. The x-axis shows the BLE sensing frequency, and the y-axis shows the battery discharge ratio. The different colors denote the results from GPS scans at different frequencies. As for ordinary users, the battery discharge ratio was 0.19 in acquiring of GPS every 60 seconds, and it reduce by 0.10 in every 600 second acquisition. From these investigation, the battery consumption was affected mainly by the GPS scanning frequency; the BLE scanning frequency had little impact on battery consumption when it was longer than 60 seconds.

PLL assumes that host users scan for BLE signals about every 60 seconds. This assumption proved to have little impact on the total battery consumption. The battery consumption of the host role was almost equal to that of the ordinary role.

C. Experiment 3: Agent Simulation in a Realistic Setting

1) Overview of Agent Simulation: PLL should be evaluated under various conditions, but this would entail much labor if we did so manually. Here, we built an agent simulator and examined different situations with it by changing the simulation conditions. We also used the simulator to evaluate the performance of the PLL calculation methodology.

2) PLL Agent Simulator: In the PLL agent simulator, agents independently move around in virtual cities and their movement histories are recorded every unit time. Each city consists a grid of roads (50 * 50 grids). To simplify the evaluation, we made the grid of each city square.

Initially, a number of agents were distributed in the city, and their initial locations were decided at random. The initial

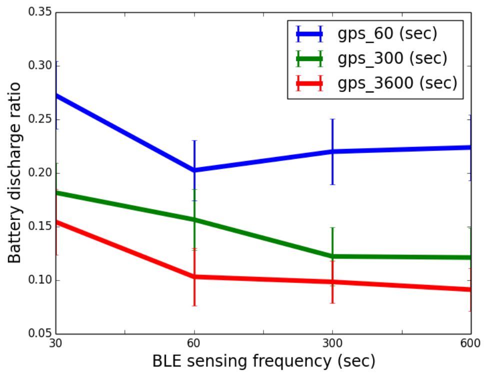


Fig. 7: Battery discharge ratio of host users

number of host users (p1) and parasitic users (p2) are defined in the simulator parameter, where p1 and p2 denotes ID of parameter in Table III. The agents move in the same way regardless of their role, and we check whether the location information of the parasitic users can be restored with the information of the host users. Each agent had goal directions (decided randomly), and travelled along the road grid. Each one made turns in accordance with a parameter called the "straight ratio (p3)". For example, an agent with a straight ratio of 1.0 moved only in its goal direction, whereas an agent with a straight ratio of 0.5 had a 50% chance of changing its direction after each time unit. The agents who arrived at their goal point disappeared from the simulator. New agents were created every unit time. The number of new agents per unit time was defined as a simulation parameter. The status of each agent was recorded during each unit time, and it was used to evaluate the performance of PLL.

The locations of the parasitic agents were estimated from the passing times of the host users and the locations of the host users acquired every 5 unit times by default. Location acquisition frequency (p4) is one of parameter in the simulation. Note that the locations of the host agents were recorded every 5 unit times in the field experiment described above, then one unit time was 60 seconds in the field experiment.

3) Experimental Conditions: Table III shows the parameters of the simulation. We evaluated the PLL performance by changing four parameters: number of host users (p1), number of parasitic users (p2), straight ratio of each agent (p3), and location acquisition frequency (p4). In each simulation, one parameter was changed and the other parameters were kept at their default values. In all tests, the velocity of each user was 1 grid per unit time. In order to reduce the effect of random decisions, we repeated the simulation 100 times under the same conditions.

4) Simulation Results: First, we evaluated the number of passes in the simulation for each combination of parameters. The orange line in Figure 8 indicates the total number of passes in each simulation. If there were few passes, we cannot estimate the locations of parasitic users completely. Thus, the

TABLE III: Selected parameter set in simulation

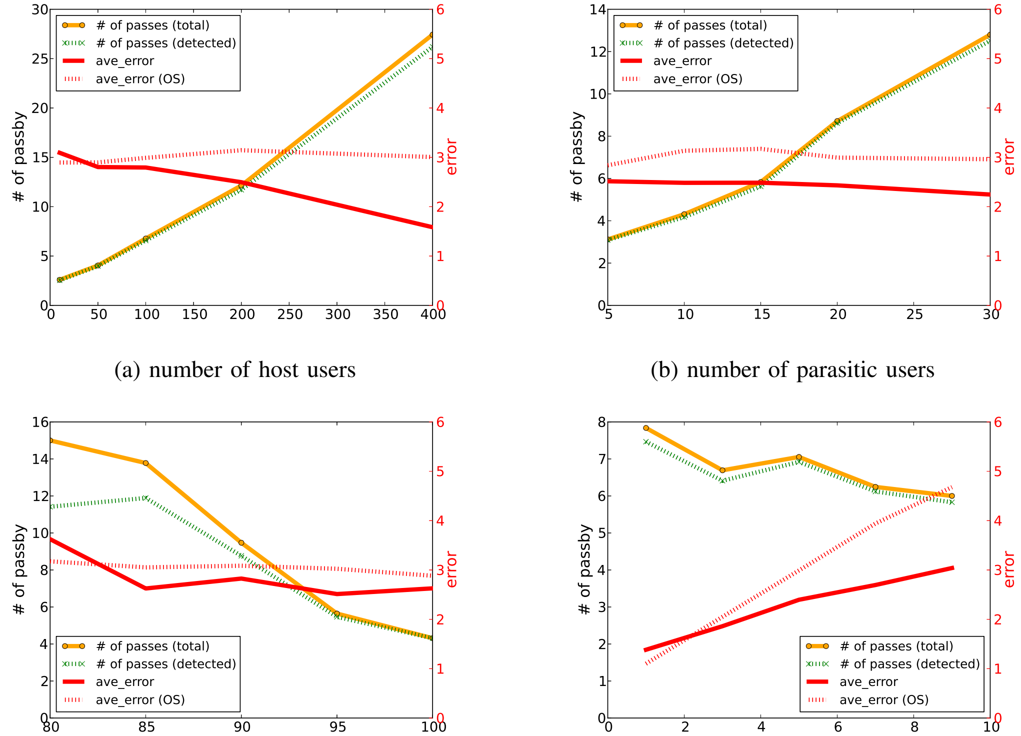
ID	parameter	changing value (**bold : default**)
p1	number of host users	[10, 50, **100**, 200, 400]
p2	number of parasitic users	[5, 10, **15**, 20, 30]
p3	straight ratio of each user (%)	[80, 85, 90, **95**, 100]
p4	location acquisition interval (every n unit times)	[1, 3, **5**, 7, 9]

frequency of passes is useful for determining the total benefit of PLL. Figure 8 specifies a natural but essential result that the number of passes is almost in proportion to the population density of the area adopted in simulation. As more parasitic and more host users are simulated with a lower straight ratio, the number of passes increases.

The green line in Figure 8 indicates the detected number of passes. The PLL calculation method cannot estimate all passes. For instance, discovering a candidate user for a passing point in PLL becomes difficult if the interval between two passings is long or the user moves in a complicated manner. The difference between the orange and green lines denotes the number of failed estimations. From this simulation result, estimations often failed when the agents' straight ratio was small. The results also revealed that the proposed calculation method is sensitive to the agents' straight ratio, but robust to other conditions. The result that the estimation is more difficult when users frequently turn around their route is intuitive.

We examined the distance error between the estimated passing location and the ground truth. The average error in each figure (the red bold line in Figure 8) depicts the total distance error divided by the total number of estimated passings in the simulation. From this simulation, the most important finding is that the error average is constant, and the average value itself is at most 3.0, which means that the error has little effect, regardless of the conditions in the simulation. Moreover, the average error decrease when the number of host users increases. This result shows that the more number of host users can improve PLL system efficiency.

Next, we evaluated the advantage of PLL compared to Bluewave (i.e., opportunistic data sharing which host users give their recent updated location to parasitic users without PLL calculation). In Figure 8, the dotted red line is the average error when users share locations by using the Bluewave methodology. It clearly indicates that the average error of PLL is less than that of Bluewave (about 30% of average error in default value conditions). Interestingly, as the number of host users and the GPS acquisition interval of host users increase, the difference between the PLL error and the Bluewave error becomes larger. PLL can be a more effective solution when the number of host users and the GPS acquisition interval large. For instance, PLL reduces the average error by about half when the host user acquires their location at 10 unit times interval. As described in the Introduction and Table IV, PLL shares location data whereas Bluewave shares various user contexts. The proposed algorithm for complementing missing locations can be built on top of the Bluewave framework for collecting location data.



(a) number of host users

(b) number of parasitic users

(c) agents' straight ratio (%)

(d) GPS acquisition interval (every n unit times)

Fig. 8: Number of detected passes and average error. The left y-axis is the number of passes, and the right y-axis is the average error. OS denotes Opportunistic Sharing and "ave_error (OS)" shows the average error when host users give their recent updated location to parasitic users without PLL calculation.

IV. Discussion

A. Comparison with Existing Opportunistic Sharing methods

The biggest advantage of PLL compared with existing opportunistic sensed data sharing methods such as CoMon is that it does not use much battery power of the host users. In the laboratory experiments, the battery consumption of the host users is strongly affected by the GPS sensing frequency, and the existing methods increase the number of GPS scans as the number of passes increases. On the other hand, the battery consumption of the host users is almost unaffected by the PLL scheme because the number of GPS scans made by them does not increase even when the number of passes increases.

B. Service Design for Keeping PLL Fair

By using PLL, users can store their location histories without worrying about their battery consumption. This means that everyone would want to be a parasitic user. However, were all users to become parasitic, no data would be accumulated for the data completion process. In order to control the number of such "free riders", we need to design the PLL service carefully. PLL should work only when the battery of a parasitic user is low, whereas users with charged batteries should be encouraged to act as host users who contribute their location data in a collaborative manner.

The authors suppose that the following constraints will work for fair PLL service: 1) Users can be parasitic only when their remaining battery energy is less than a threshold. 2) Users have to act as host users in order to get the privilege to be a parasitic user. In this case, the parasitic role can be incentive for playing the host role.

Additionally, PLL is also valid in situations where the user cannot receive GPS signals. Users cannot receive GPS signal when the GPS signal receiver is out of order or when the GPS signal or Internet connection is bad.

C. Other limitations

Issues remain to be overcome before PLL can be considered practical. We should solve estimation inaccuracy arising from actual environment like different walking speeds of users, error of GPS sensor, and so on. Moreover, little number of host users cases can be also issue.

V. Related Work

A. Motivation for Continuously Location Logging

Many researchers have studied how location information can be utilized. For instance, it can be used for making location-information histories in a life-logging service [7], [8], for modeling the behavior of individual users and for analyzing urban dynamics [9]–[16]. Though the frequency of collecting information also varies with the purpose as does the motivation for acquiring it in the first place, be it individual, corporate, or governmental, continuous logging (with no missed locations) is important for improving the performance in each case.

In particular, position information collected by smartphones has been used to analyze check-in histories in Foursquare [14], and position information such as geo tags attached to tweets has been used to analyze the behaviors of Twitter users [17]. However, in these cases, location information is not collected unless the user is checking in or posting a tweet. Such data are too sparse to use for keeping track of a user's lifestyle.

As far as the authors know, there is no research on using position information acquired from user smartphones continuously without any sensor data such as GPS or BLE. In contrast, PLL collects location information even for users with low batteries; it can acquire location information continuously and universally without consuming much power (GPS and BLE active scanning can be completely turned off).

B. Constructing Dense Datasets by Collaborating with Others

Sharing data with other users enables dense datasets to be obtained. In particular, there has been a lot of research on data sharing including crowd sensing and peer-to-peer data sharing.

In crowd sensing systems, many unspecified users upload sparse sensor data. As a result, large-scale dense datasets can be collected on, for example, traffic conditions, ambient noise, pollution, and local contexts [18]–[20]. Techu [21] and SecureFind [22] are crowdsourcing systems for finding lost objects. These systems are not intended for logging the locations of objects.

Sensor data collected via crowd sensing are often affected by sparse or missing values [23], [24]. Kurasawa et al. [23] proposed to estimate a missing value in sensed data of crowd sensing by using correlations among multiple types of sensor data. This idea forces the user to always carry multiple sensors.

There have been many studies on peer-to-peer data sharing by nearby users through opportunistic communication networks [4], [5], [25]–[28]. CoMon [25] is a platform to cooperatively monitor ambient data by sharing it with nearby users. CoMon limits sharing to devices that are within direct communication range of each other and targets cooperation with long stayers only. CHICHAT [4] was proposed to reduce the size of context representations designed to exchange data via device-to-device direct communication. Encore [27] is a context-based private communication platform. It detects nearby users within Bluetooth radio range and enables participants at events to communicate and share information.

The methods used in the above studies depend on direct data exchange with nearby users via an opportunistic communication network. They are not suited to keeping track of daily activities, because they require time and effort to establish communication networks and exchange data on-the-fly.

Bluewave [6] is a Bluetooth-based technique that allows devices to opportunistically share contexts when they are nearby. Users can share context information including location information without directly communicating with nearby devices as in PLL. However, Bluewave lacks a mechanism for estimating or complementing context information and instead simply shares the raw information from the sensors. Thus, unless a host user who happens to be nearby has sensed and advertised the location information of the current place, parasitic users cannot obtain accurate location information especially while both users are walking.

TABLE IV: Comparison of PLL and other techniques

	Direct?	Scope	Shared data
Ordinary ODS(†)	YES	Contexts data	Raw
Bluewave [6]	NO	Contexts data	Raw
PLL	NO	location only	Estimated

†Opportunistic sharing (e.g., CHICHAT [4], CoMon [25], Encore [27])

Teraoka et al. [29] proposed symbiosis location logging (SLL), a method that is close in concept to PLL. SLL aims to enrich the location logs of users, by their sharing location data with each other. Different from opportunistic sensing methods, SLL does not exchange data, but rather estimates locations with location history data acquired at the usual frequency and with information from passersby. However, SLL requires the parasitic user's location, as well as the host users' location and host users' direction.

Table IV compares the various opportunistic data sharing techniques with the PLL technique. In PLL, the user devices do not have to be directly connected in order to share location data. The host user provides raw sensed location data to the server, but the parasitic user can get estimated accurate location data even if the host user did not sense the location data just when they passed by each other.

VI. Conclusion

Parasitic Location Logging (PLL), which is an innovative approach that continually collects users' location data with little battery consumption even after their location acquisition function has stopped, is proposed. With PLL, a parasitic user can acquire their position information by analyzing only the locations of host users who pass by the parasitic user; thus, the parasitic user does not have to turn on their GPS, WiFi, or BLE active scanning. To realize this functionality, we formulate a simple recovering method of parasitic user locations as range based quadratic optimization using passing information of parasitic and host users.

Our experiments showed that PLL dramatically decreased the battery consumption (by about 40%) of parasitic users. Moreover, we found that the battery consumption of a host user was almost equal to that of an ordinary user in our PLL experiments. Our agent simulation revealed that the proposed algorithm can estimate the passing location accurately, especially when large number of host users with less frequent of GPS acquisition are employed, a realistic scenario of widespread IT services. Thus, PLL is introduced as a complementary methodology to existing opportunistic location data sharing scheme. Some issues and limitations that impact practical applications of PLL were discussed for future work. We believe that PLL will become a prominent method for complementing the location logs of smartphones when they run low on battery power.

REFERENCES

[1] Z. Fan, X. Song, and R. Shibasaki, "CitySpectrum: A non-negative tensor factorization approach," in *Proc. of UbiComp*, 2014.

[2] M. Shimosaka, K. Maeda, T. Tsukiji, and K. Tsubouchi, "Forecasting urban dynamics with mobility logs by bilinear poisson regression," in *Proc. of UbiComp*, 2015.

[3] K. Nishi, K. Tsubouchi, and M. Shimosaka, "Hourly pedestrian population trends estimation using location data from smartphones dealing with temporal and spatial sparsity," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014.

[4] S. Cho and C. Julien, "Chitchat: Navigating tradeoffs in device-to-device context sharing," in *Proc. PerCom 2016*, ser. PerCom '16, 2016, pp. 1–10.

[5] M. Kadadha, H. Al-Ali, M. A. Mufti, A. Al-Aamri, and R. Mizouni, "Opportunistic Mobile Social Networks: Challenges Survey and Application in Smart Campus," in *Proc. WiMob 2016*, 2016, pp. 1–8.

[6] A. A. de Freitas, M. Nebeling, A. S. K. K. Ranithangam, J. Yang, and A. K. Dey, "Bluewave:Enabling Opportunistic Context Sharing via Bluetooth Device Names," in *Proc. EICS 2016*, 2016, pp. 38–49.

[7] A. Mehrotra, S. R. Müller, G. M. Harari, S. D. Gosling, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Understanding the role of places and activities on mobile phone interaction and usage patterns," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 84:1–84:22, Sep. 2017.

[8] V. K. Singh and R. R. Agarwal, "Cooperative phoneotypes: Exploring phone-based behavioral markers of cooperation," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16, 2016, pp. 646–657.

[9] T. Yabe, K. Tsubouchi, and Y. Sekimoto, "Cityflowfragility: Measuring the fragility of people flow in cities to disasters using gps data collected from smartphones," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 117:1–117:17, Sep. 2017.

[10] S. Yang, M. Wang, W. Wang, Y. Sun, J. Gao, W. Zhang, and J. Zhang, "Predicting commercial activeness over urban big data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 119:1–119:20, Sep. 2017.

[11] J. Lee, K. Lee, E. Jeong, J. Jo, and N. B. Shroff, "Context-aware application scheduling in mobile systems: What will users do and not do next?" in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16, 2016, pp. 1235–1246.

[12] Z. Fan, X. Song, R. Shibasaki, T. Li, and H. Kaneda, "Citycoupling: Bridging intercity human mobility," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16, 2016, pp. 718–728.

[13] M. Xu, D. Wang, and J. Li, "Destpre: A data-driven approach to destination prediction for taxi rides," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16, 2016, pp. 729–739.

[14] M. Lichman, D. Kotzias, and P. Smyth, "Personalized location models with adaptive mixtures," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '16, 2016, pp. 67:1–67:4.

[15] Y. Li, J. Bao, Y. Li, Y. Wu, Z. Gong, and Y. Zheng, "Mining the most influential k-location set from massive trajectories," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '16, 2016, pp. 51:1–51:4.

[16] J. D. Mazimpaka and S. Timpf, "A visual and computational analysis approach for exploring significant locations and time periods along a bus route," in *Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, ser. IWCTS '16, 2016, pp. 43–48.

[17] Z. Liu, Y. Huang, and J. R. Trampier, "Spatiotemporal topic association detection on tweets," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '16, 2016, pp. 28:1–28:10.

[18] G. Bin, W. Zhu, Y. Zhiwen, W. Yu, Y. Neil Y., H. Runhe, and Z. Xingshe, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Computing Surveys*, vol. 48, no. 1, 2015.

[19] Z. Pengfei, Z. Yuanqing, and L. Mo, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proceedings of the MobiSys 2012*, ser. MobiSys '12, 2012, pp. 379–392.

[20] C. Yohan, F. L. Nicolas D., C. Hojung, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proceedings of the UbiComp '12*, ser. UbiComp '12, 2012, pp. 481–490.

[21] I. Agadakos, J. Polakis, and G. Portokalidis, "Techu: Open and Privacy-Preserving Crowdsourced GPS for the Masses," in *Proc. MobiSys 2017*, 2017, pp. 475–487.

[22] S. Jingchao, Z. Rui, J. Xiaocong, and Z. Yanchao, "Securefind: Secure and privacy-preserving object finding via mobile crowdsourcing," *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*, vol. 15, no. 3, pp. 1716–1728, 2016.

[23] H. Kurasawa, H. Sato, A. Yamamoto, H. Kawasaki, M. Nakamura, Y. Fujii, and H. Matsumura, "Missing Sensor Value Estimation Method for Participatory Sensing Environment," in *Proc. 2014 PerCom*, 2014, pp. 103–111.

[24] Z. Jiangchuan, L. Siyuan, and N. Lionel M, "Effective routine behavior pattern discovery from sparse mobile phone data via collaborative filtering," in *Proceedings of the PerCom 2013*, ser. PerCom 2013, 2013, pp. 29–37.

[25] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song, "Comon: Cooperative ambience monitoring platform with continuity and benefit awareness," in *Proc. MobiSys 2012*, 2012, pp. 43–56.

[26] O. Helgason, S. T. Kouyoumdjieva, and G. Karlsson, "Opportunistic Communication and Human Mobility," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1597–1610, 2014.

[27] A. Paarijaat, E. Viktor, L. Matthew, S. Elaine, B. Bobby, and D. Peter, "Encore: Private, context-based communication for mobile social apps," in *Proceedings of the MobiSys '14*, ser. MobiSys '14, 2014, pp. 135–148.

[28] A. A. D. Freitas and A. K. Dey, "Using Multiple Contexts to Detect and Form Opportunistic Groups," in *18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2015, pp. 1612–1621.

[29] T. Teraoka, K. Tsubouchi, and H. Gomi, "Symbiotic Construction of Individual's Rich Location Dataset," in *Proc. UbiComp 2018*, 2018, pp. 271–274.