UrbanMobilitySense: A User-Centric Participatory Sensing System for Transportation Activity Surveys

Fang-Jing Wu and Hock Beng Lim

***Abstract*— Transportation activity surveys collect the travel behavior of people, including when, where, and how they travel for urban planning purposes. Traditionally, transportation activity surveys are carried out using conventional questionaires, which are labor intensive and error prone. In this paper, we have developed a smartphone-based mobility sensing system, called UrbanMobilitySense, which captures human mobility information automatically to conduct transportation activity surveys. The UrbanMobilitySense system was designed to address two critical issues: 1) energy conservation and 2) privacy preservation. To optimize the energy utilization of smartphone, we avoid using the GPS sensor when the user is at long-stay places and filter out redundant data before data uploading. To preserve personal privacy, each smartphone maintains the user's long-stay places by two separate profiles: 1) private place profile and 2) public place profile. The former maintains the privacy-preserved places (e.g., home), whereas the latter maintains the public places (e.g., parks). We implement the UrbanMobilitySense system to conduct real-world transportation activity surveys, study the performance of our system through extensive experiments, and analyze the computational complexity of the proposed algorithms. The outcome of our work has been deployed in Singapore to support the Land Transport Authority's transportation activity surveys.**

***Index Terms*— Crowdsourcing, intelligent transportation systems, participatory sensing, pervasive computing, smart cities.**

I. Introduction

RECENTLY, smartphones equipped with various types of sensors have enriched the confluence of ubiquitous computing and wireless sensor networks (WSNs) and also boosted many promising cyber-physical systems [1] such as urban monitoring [2] and intelligent transportation systems [3].

This paper is motivated by a real-world problem that analyzes transportation activities in urban areas. In Singapore, the Land Transport Authority conducts transportation activity surveys every four years. Typically, the surveys are conducted via complicated questionnaires which are problematic and error-prone. We exploit the sensing capability of smartphones to conduct transportation activity surveys automatically, where users carry smartphones to capture mobility seamlessly [4], [5]. The collected data is uploaded to the backend servers which recover transportation behavior including the visited locations and transportation modes such as Mass Rapid Transit (MRT), buses, and walking.

However, using smartphones to collect human mobility raises two challenges, *energy conservation* and *privacy preservation*. The former goal is to optimize the energy usage of smartphones for capturing real moving traces, while the latter goal is to avoid exposing personal places in the course of data collection. To achieve the both goals, we design a mobility sensing system for smartphones, called *UrbanMobilitySense*, which aims at reducing energy consumption for sensing and data uploading in a privacy-preservation way. To reduce energy consumption for sensing, the main idea is to avoid using GPS in users' long-stay places (such as home, offices, parks, and the zoo). To reduce energy consumption for uploading, we design data filters to upload low-quantity and high-quality data. However, exposing users' long-stay places may compromise personal privacy. Thus, we separate each user's long-stay places into two profiles, the *private place profile* ('private places' for short) and the *public place profile* ('public places' for short). The private places maintain private-preserved places which are learned by each user's smartphone based on the ambient network signatures and are only kept by the smartphone, while the public places maintain frequent places which are identified by the backend servers based on the density of all users' data and are shared among them. Note that privacy preservation is place-centric because users usually are more willing to expose their location information in public places for accessing location-based services. Finally, we implement our system to conduct transportation activity surveys in the real world and study the performance of our system through real-world experiments. Fig. 1 shows the application scenario and the data flow in our system, where personal data is collected to the backend servers and then a confluence of public knowledge (i.e., public places) goes back to each user's smartphone for facilitating mobility sensing.

As our work is motivated by a real-world problem, the UrbanMobilitySense has the following unique features and contributions. First, compared to the duty-cycle-based GPS control schemes [6], [7] which cannot capture accurate traces all the time in real-world deployment due to uncertain sensitivity of GPS sensors, the UrbanMobilitySense can

Manuscript received April 17, 2014; revised June 22, 2014 and September 3, 2014; accepted September 3, 2014. Date of publication September 23, 2014; date of current version October 21, 2014. This work was supported by the National Research Foundation of Singapore through the Future Urban Mobility Programme of the Singapore-MIT Alliance for Research and Technology. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ashish Pandharipande.

F.-J. Wu is with the Intelligent Systems Centre, Nanyang Technological University, Singapore 639798, and also with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 119613 (e-mail: fangjing@cs.nctu.edu.tw).

H. B. Lim is with the Intelligent Systems Center, Nanyang Technological University, Singapore 639798 (e-mail: limhb@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2014.2359876

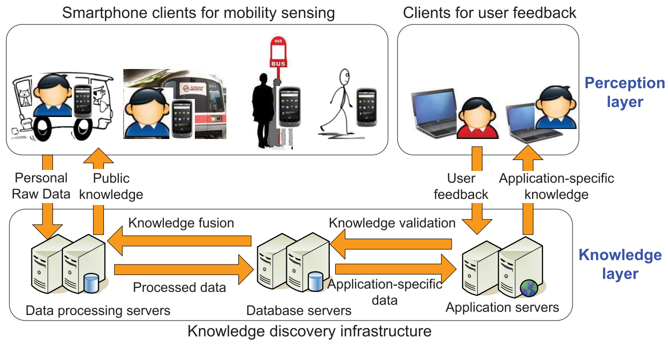


Fig. 1. System architecture and data flow of the UrbanMobilitySense.

collect large-scale and highly dynamic mobility data in the real world. Second, our system works with uncertain localization technologies instead of a GPS-only localization system [6], [8]. Third, as existing work conducts place matching on a remote server [9], our system performs place matching locally on each user's smartphone for the privacy-preservation purpose. Finally, the UrbanMobilitySense has been deployed in Singapore to support real-world transportation activity surveys.

The rest of this paper organization is as follows. We discuss the existing work in Section II. Section III explains our system design. Section IV shows implementation and experimental results. Performance evaluation and complexity analysis of our system are explained in Section V. Section VI discusses open issues from practical deployment perspectives to highlight potential future work. Section VII concludes this paper.

II. Related Work

Using vehicles equipped GPS sensors or smartphones to collect daily mobility information has attracted a lot of attention. Work [10]–[13] infers users' transportation modes. By collecting real-time GPS locations of cars, [8] estimates the current traffic conditions of road segments such as waiting time at intersections. Based on GPS and WiFi data collected by vehicles, [14] designs a traffic monitoring system to estimate travel delays along road segments, where coexisting GPS and WiFi data is collected periodically. Reference [15] determines whether users are in a bus, where the the bus schedules are considered to match users' traces. In [6], adaptive duty cycle is designed to control GPS usage. The work in [16] and [17] conducts transit tracking and predicts arrival time of buses. Based on the colors of traffic signals detected by smartphones, [18] advises on driving speed to reduce the vehicle fuel consumption. However, [9] and [19]–[23] focuses on discovering meaningful places instead of absolute GPS coordinates. In [9], a remote server identifies ambient fingerprints of places and performs place matching. Based on GPS data, a density-based clustering algorithm is proposed to recognize meaningful places [19]. Using a set of radio beacons to define a physical space is proposed in [20]. In [21], the backend server discovers the places visited by users based on the patterns of changes in ambient network signals. In [22], users manually label places to avoid using GPS in the labeled places. In [23], users are allowed to verify the places learned by smartphones and define new places manually.

Comparing with the above existing work, our system has the following technical breakthroughs. First, while the existing systems highly rely on GPS, our system allows lower-accurate WiFi and GSM information to compensate for GPS missing. Second, as the existing efforts focus on providing particular transportation-related services or tracking passengers in buses, our system more emphasizes how to develop intelligence of smartphones to capture high-quality and low-quantity mobility data. Third, our system addresses the privacy issue from the sensing perspective. Fourth, our system allows a knowledge loop which incorporates private and public knowledge to improve mobility sensing. Fifth, our system conducts place learning automatically, where users will not engage in labeling their long-stay places. Finally, instead of off-line place matching in remote servers, our system conducts real-time place matching locally without incurring too much computation cost for smartphones.

III. System Design

Fig. 1 shows our system architecture including two layers, *perception layer* and *knowledge layer*. The perception layer collects raw sensing data and feedback from users, while the knowledge layer analyzes the collected mobility data to represent as application-specific knowledge. In the perception layer, smartphone clients are responsible for collecting mobility data including GPS/WiFi/GSM data and 3-axes accelerometer data, while the clients for collecting user feedback and validation could be either mobile phones, laptops, or desktops. The knowledge layer consists of *data processing servers*, *database servers*, and *application servers*. The data processing servers transform personal raw sensing data into application-specific data which is high-level mobility information (e.g., stops and transportation modes). The database servers maintain processed data. The application servers provide an interactive interface to represent application-specific knowledge and receive feedback from users. Finally, the data processing servers conduct knowledge fusion (e.g. data clustering) to find out the public knowledge (e.g., public places) for improving the energy usage at clients. Next, we make observations from experiments to explain the challenges of real deployment that motivates us to design a place-aware mobility sensing scheme and then explain our system design in detail.

A. Challenges of Real Deployment

Since our system must support real-world transportation activity surveys, we discuss the essential requirements and challenges for real deployment as follows.

1) Uncertainty of Sensor Sensitivity: Since our system must accommodate data from various types of smartphones, a duty-cycle-based GPS control scheme [6], [7] cannot capture accurate traces all the time for such real deployment. This is because GPS sensors on different smartphones may have different sensitivity even though these smartphones are the same models. Below, we conduct an experiment to justify the observation. We implement a speed-based GPS control scheme on smartphones, where GPS switches on for 30 seconds and switches off for a short period of time. The GPS-off interval

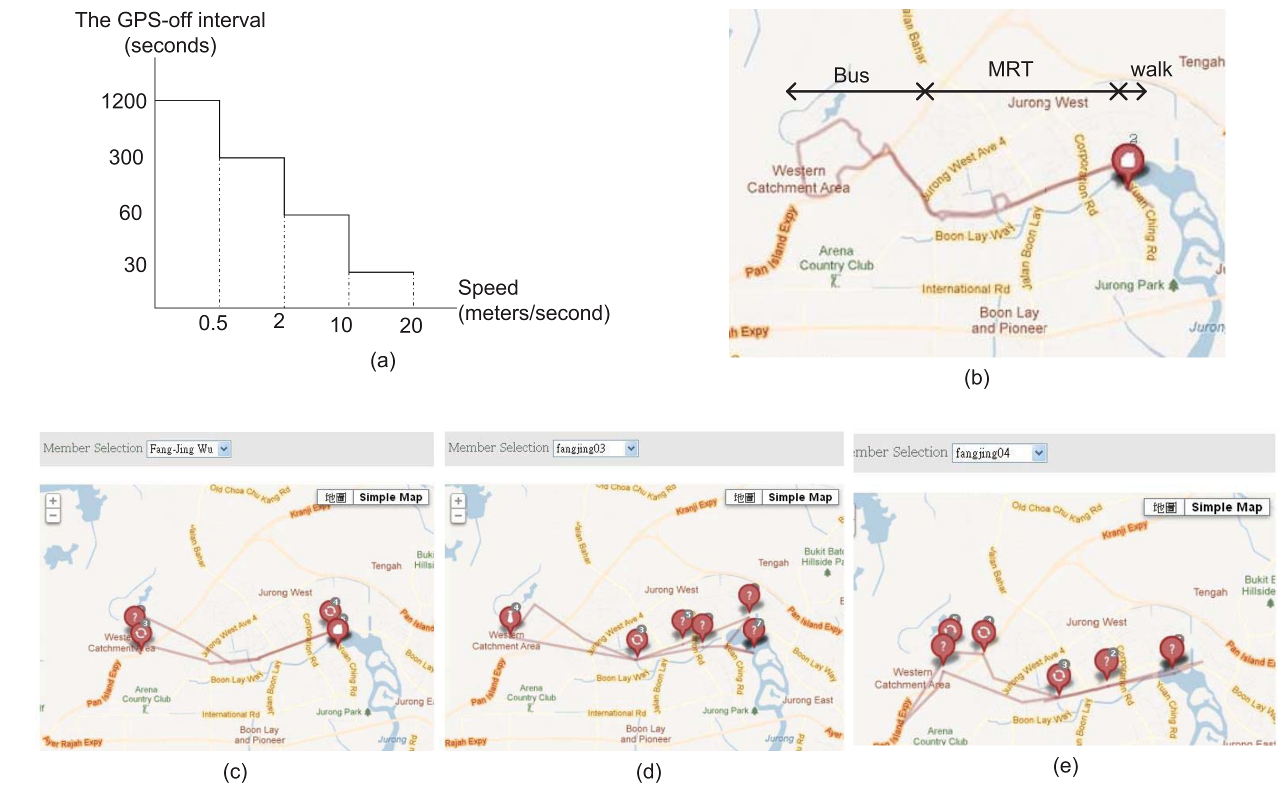


Fig. 2. Traces collected by duty-cycle-based approaches. (a) A speed-based mapping for GPS-off intervals. (b) Traces collected by Nexus One with an always-on GPS. (c) Traces collected by Nexus One with duty-cycled GPS. (d) Traces collected by HTC Sensation with duty-cycled GPS. (e) Traces collected by Samsung Galaxy s2 with duty-cycled GPS.

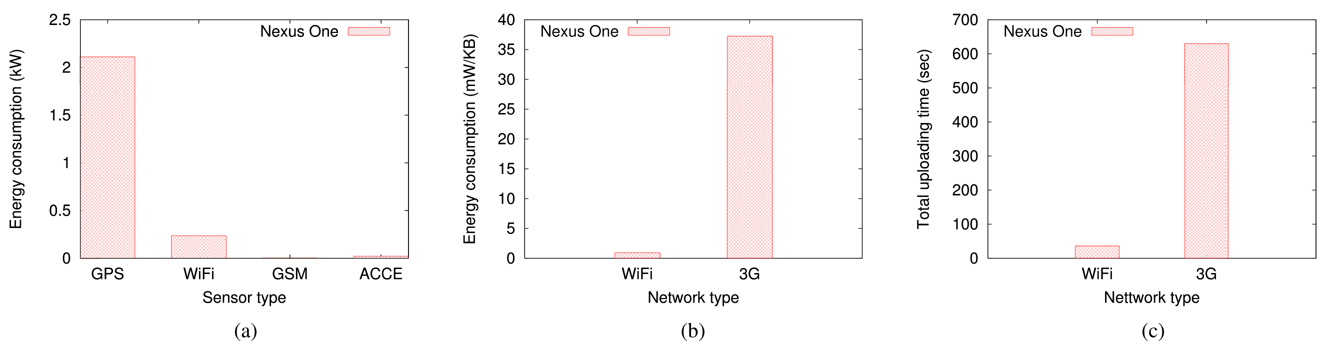


Fig. 3. Energy consumption of sensing and uploading data. (a) Energy consumption for sensing. (b) Energy consumption for uploading. (c) Total uploading time.

is based on the real-time speed, as shown in Fig. 2(a). In this experiment, a single user carries 4 smartphones (2 Nexus one, one HTC Sensation, and one Samsung Galaxy S2) to move along a round-trip trace over a single day. One of them collects traces using an always-on GPS data collection scheme, and the other three are using the speed-based GPS control scheme. The collected traces are shown in Fig. 2(b)-(e). As it can be seen in Fig. 2(c)-(e), the traces collected by the three smartphones are different from each other even though they control GPS sensors in the same way.

2) *Energy Conservation:* To avoid interrupting normal usage of smartphones, energy conservation for smartphones is an essential requirement for real-world deployment. We conduct two experiments to observe the energy consumption for sensing and uploading, respectively. We consider PowerTutor [24] which is a power monitor for android devices to measure energy consumption of different hardware components. In the first experiment, we study the energy consumption for sensing GPS, WiFi, GSM, and accelerometer data, respectively. Each type of sensors is turned on individually to collect data for two hours. Fig. 3(a) shows the experimental results. As it can be seen, the GPS sensor consumes much more energy than other types of sensors. This motivates us to design a place-aware mobility sensing scheme that infers mobility intention of users to control the GPS adaptively. In the second experiment, we study the energy consumption for uploading via 3G and WiFi network interfaces, respectively. For each type of network interfaces, 200 files are uploaded to

the server, where each file has a size of 15.4 MB. Fig. 3(b) shows that data uploading via the 3G network consumes much more energy than via the WiFi network. Also, the total uploading time via the 3G network is significantly longer than via the WiFi network because the bandwidth of the 3G network is much less than the WiFi network, as shown in Fig. 3(c). This motivates us to design data filters to reduce the amount of uploaded data.

3) Privacy Preservation: Since participatory sensing is a technical means to conduct transportation activity surveys, user privacy is a critical issue in such real-world deployment. Moreover, the ownership of personal place information must be kept by each individual.

4) Human Behavior Inference: Inference of moving intention for such real-world deployment is a challenge. Typically, localization relies on a massive radio map collected by a prior training phase. However, collecting and training such a massive radio map for urban-scale applications will incur extremely high cost. Moreover, fine-grained localization is not necessarily for collecting urban-scale mobility. For example, there is no need to distinguish the user in the bathroom from the user in the bedroom for transportation activity surveys.

To meet the above requirements, we consider the following principles to design our UrbanMobilitySense.

1) Design Place-Aware Sensing Instead of Duty-Cycled Sensing: For energy conservation, we will exploit place information to avoid switching on/off GPS frequently. In this way, sensor sensitivity will not affect the quality of collected traces.

2) Maintain Privacy-Preserved Place Profiles Locally: For privacy preservation, each user's smartphone locally caches a private place profile self-learned by the smartphone. On the other hand, a public place profile identified by the backend servers can be downloaded by smartphones. In this way, users can keep their own private place information, and public knowledge also can be shared in the public place profile.

3) Exploit Place Information to Detect Space Transition Instead of Localization: We model each place as a set of ambient network signatures. Thus, we can infer if the user moves from one space towards the other one without localization support.

B. Place-Aware Mobility Sensing

The main idea is to avoid using GPS in each user's long-stay places. We assume all users trust the software on their smartphones. Fig. 4 shows the software components for smartphones. The *sensing control middleware* coordinates all components adaptively. The component of *place learning* consists of the private place learner and the public place monitor. The former one learns the private place profile automatically, while the latter one downloads the freshest public place profile from the backend servers. The component of *place matching* detects whether the user appears in private/public places. The *activity time predictor* estimates how long the user may stay in a public place. The *status detection* detects whether the smartphone is moving or stationary so as to trigger place

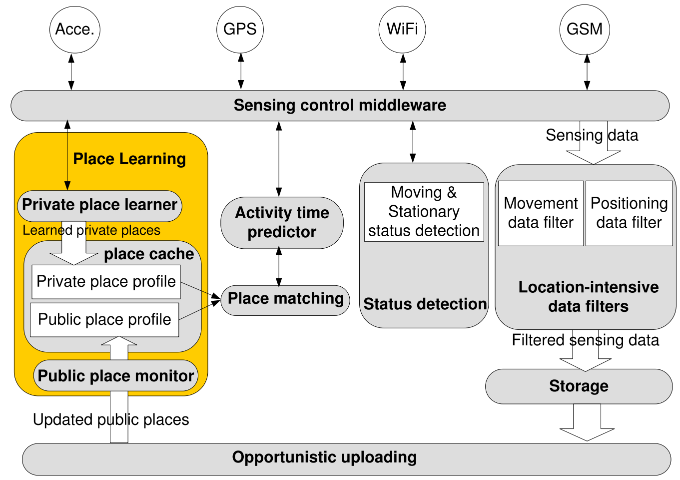


Fig. 4. The software design for smartphones.

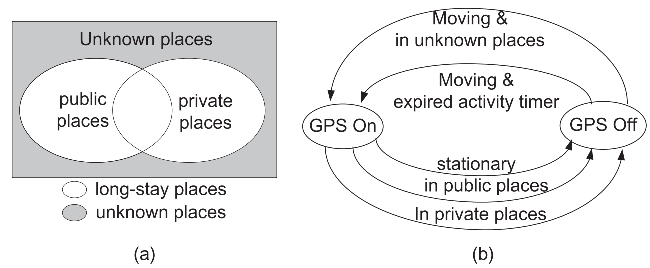


Fig. 5. The GPS control incorporated with a place model. (a) The diagram of place model. (b) The state transition of GPS sensor.

matching for GPS control. The *location-intensive data filters* picks up higher-accurate location data[1] and accelerometer data with significant changes instead of uploading all of raw data. The *opportunistic uploading* component uploads filtered data to the backend servers once the Internet is available. Next, we explain our design in detail.

1) Sensing Control Middleware: This component coordinates the sensing timing of sensors. The 3-axes accelerometer, WiFi, and GSM data is collected periodically. The GPS control is based on the place model in Fig. 5(a) incorporated with the place learning and matching algorithm (explained in Section III-B2 and Section III-B3). The set of private places includes the user's personal long-stay places such as home, offices, and public parks. The set of public places includes human activity hotspots which are contributed by all of users such as bus stations, the airport, and public parks. Note that there may exist overlap between private and public places since a human activity hotspot (e.g., a public park) may be also one of a particular user's long-stay places. A place is defined as an unknown place if it belongs neither private places nor public places. Based on the place model, the GPS sensor follows the state transition in Fig. 5(b). When the smartphone detects moving status in an unknown place, it will turn on GPS. The smartphone will turn off GPS immediately when it detects either one of the three situations: (i) stationary status, (ii) user

[1] In addition to the GPS localization technology, we also consider network-based localization technologies (i.e., WiFi and GSM networks) which transform WiFi BSSID/GSM Cell IDs into a pair of latitude and longitude.

in a public place, or (iii) user in a private place. If the user is in a public place, the sensing control middleware requests the activity time predictor (explained in Section III-B4) to estimate how long the user may move away from the current place and starts an activity timer. After the activity timer is expired, the smartphone turns on the GPS again if it stays in moving status.

2) Place Learning: We design place learning algorithms to identify the long-stay places of users based on the patterns of network changes and density of data.

a) Private place learner: This component maintains the WiFi signatures of the user's private place profile $\mathcal{F}$. Initially, $\mathcal{F} = \emptyset$. When the user's smartphone enters an unknown place at time t, it starts learning the place's signatures $P_i = P(t)$, where $P(t) = \{W_1, W_2, \ldots, W_k\}$ which contains all visible WiFi BSSIDs $W_1, W_2, \ldots, W_k$ at time t. The P_i will be updated by $P_i = P_i \cup P(t+1)$ at time $t+1$ if $P_i \cap P(t+1) \neq \emptyset$ which means the user staying in the current place. Otherwise, it gives up learning P_i because the user leaves from the current place. The above learning process will be repeated incrementally for Δ_L duration to learn P_i. Once the Δ_L is expired, $\mathcal{F}$ is updated by $\mathcal{F} = \mathcal{F} \cup \{P_i\}$. In this way, $\mathcal{F}$ will be updated from time to time if the user enters other unknown place P_j, where $P_j \cap P_k = \emptyset$, for each $P_k \in \mathcal{F}$.

b) Public place monitor: This component checks whether the samrtphone has cached the newest public place profile. If not, the smartphone downloads the newest one from the backend servers once there is Internet connection. Based on the density of GPS data in the backend servers, the public place profile $\mathcal{H} = \{Q_1, Q_2, \ldots, Q_n\}$ maintains the hotspots of users' mobility, where $Q_i = (x_i, y_i)$ is the pair of latitude and longitude coordinates of public place Q_i. The Density-based Spatial Clustering of Applications with Noise (DBSCAN) is applied to figure out the public place profile $\mathcal{H}$.

3) Place Matching: Our place matching consists of two parts: *private place matching* and *public place matching*. The former detects whether the user in a private place, while the latter detects whether the user in a public place.

a) Private place matching: When a smartphone perceives the real-time WiFi signature at time t, denoted by $R(t) = \{W_1, W_2, \ldots, W_n\}$, it will compute the *'similarity'* between $R(t)$ and each $P_i \in \mathcal{F}$ which is defined as $|R(t) \bigcap P_i|$, where $W_1, W_2, \ldots, W_n$ are the visible WiFi BSSIDs at time t. If $|R(t) \bigcap P_i| \geq \tau_w$, for $P_i \in \mathcal{F}$, the smartphone concludes the user in private place P_i. Here, τ_w is a predefined threshold which indicates the tolerance to WiFi signal variation. For example, $\tau_w = 1$ is a loose threshold.

b) Public place matching: Let Ω denote a real-time GPS fix. If there exists $Q_i \in \mathcal{H}$ such that $d(\Omega, Q_i) < R_a$ and $d(\Omega, Q_i) < d(\Omega, Q_j)$, $Q_i, Q_j \in \mathcal{H}$, the smartphone infers user in public place Q_i, where $d(\Omega, Q_i)$ is the distance between Ω and Q_i, and R_a is a predefined threshold of activity range.

4) Activity Time Predictor: The component creates a *feedback loop* between smartphones and the backend servers to exploit the public places contributed by all of participants for reducing energy consumption. The main idea is to delay GPS usage when the user appears in a public place. Specifically, when the smartphone detects the user in public place Q_i,

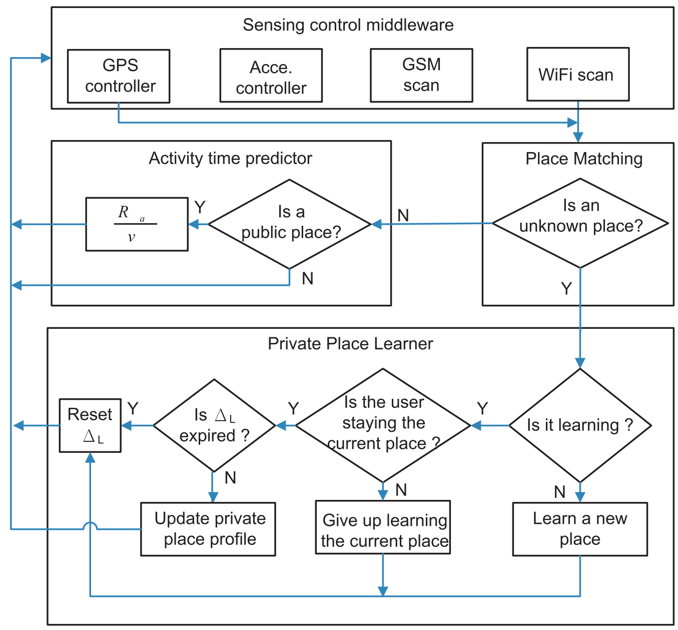


Fig. 6. The workflow of place learning and matching on a smartphone.

it predict the activity duration by R_a/v which is how long the user will move out of the activity range of Q_i, where v is the real-time speed. Then, the smartphone must turn off the GPS immediately for R_a/v duration.

5) Status Detection: This component determines the user status. Let $a(x)$, $a(y)$, and $a(z)$ denote the accelerometer's reading in x, y, and z axis, respectively. If either $a(x)$, $a(y)$, or $a(z)$ is greater than a predefined threshold τ_a, the smartphone detects a shaking event. When the smartphone detects consecutive shaking events for a period of T_m, it infers the user in moving status. Similarly, if there is no shaking event for T_m, the user is in stationary status.

Fig. 6 shows the cooperation between the above components. When the smartphone collects either a GPS fix or a set of WiFi signals, it conducts place matching. When the smartphone detects an unknown place, it starts to learn the unknown place if it is not learning the unknown place. Otherwise, the smartphone checks whether the user is staying in the current place. If so, the place's signatures are updated incrementally until Δ_L is expired. Otherwise, it gives up learning the current place because it's not a long-stay place of the user. On the other hand, when the smartphone detects either a public place or a private place, it invokes the activity time predictor to estimate the activity time. If the user is in a private place, it simply considers an infinite activity time until an unknown place is detected.

6) Location-Intensive Data Filters: This component reduces the amount of uploaded data so as to reduce energy consumption for uploading. Thus, we design two data filters to pick up 'high-quality' and 'low-quantity' data instead of uploading all of raw data.

a) Positioning data filter: The main idea is to collect the higher-accurate location information if multiple types of locations coexist. Specifically, when a piece of sensing data s_i arrives at time t_i, the filtering principles are as follows.

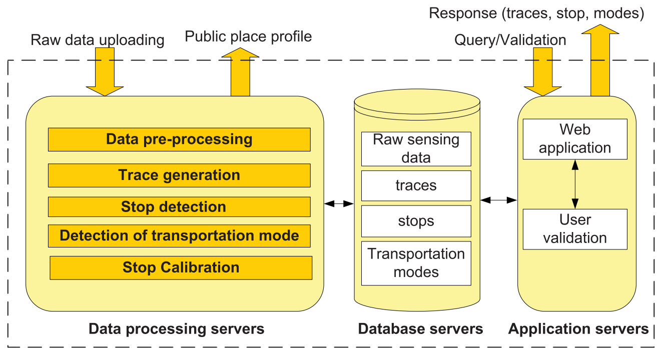


Fig. 7. The software design of the backend infrastructure.

- If s_i is GPS data, s_i must be uploaded.
- If s_i is WiFi data, find out the freshest GPS data in the storage of the smartphone, say g_k with timestamp t_k. If the time gap $(t_i - t_k) > I_g$, then s_i must be uploaded. Here, I_g is the sampling interval of GPS data. In this case, s_i must compensate for GPS data missing. Otherwise, s_i is simply dropped.
- If s_i is GSM data, find out the freshest GPS data and the freshest WiFi data, say g_k with timestamp t_k and w_j with timestamp t_j. If $(t_i - t_k) > I_g$ and $(t_i - t_j) > I_w$, then s_i must be uploaded. Here, I_w is the sampling interval of WiFi data. In this case, s_i must compensate for GPS and WiFi data missing. Otherwise, s_i is dropped.

b) Movement data filter: This component will select the accelerometer data with significant variation. Note that the system simply considers that the accelerometer data is the same as before if there is no accelerometer data is uploaded. Specifically, when a piece of accelerometer data s_i arrives at time t_i, s_i must be uploaded if s_i indicates moving status mentioned in Section III-B5. Otherwise, s_i is dropped.

C. Transportation Activity Inference

We design an algorithm to transform the collected sensing data into transportation behavior. Fig. 7 shows the software design of our backend infrastructure. The data processing servers infer each user's transportation behavior including moving traces, stops, stop duration, and transportation modes. The database servers maintain raw data and high-level transportation behavior inferred by our algorithm. The application servers provide an interactive interface for user validation. Below, we explain the proposed algorithm in detail.

Step 1 (Data Pre-Processing): This step transforms raw data into meaningful information. Each uploaded file is a mixture of accelerometer, GSM, WiFi, and GPS data. Each piece of accelerometer data contains a timestamp and sensing readings in x-axis, y-axis, and z-axis. Each piece of GSM data contains a timestamp, the Mobile Country Code, the Mobile Network Code, the Location Area Code, the cell ID, and the received signal strength of GSM network. Each piece of WiFi data contains a timestamp and information of visible WiFi networks including BSSIDs, SSIDs, and received signal strengths. Each piece of GPS data contains a timestamp, a pair of latitude and longitude, and the speed. For accelerometer data, we compute the average, standard deviation, maximum, and minimum for each given time window. For WiFi and GSM data, the triangulation positioning is conducted to determine user locations.

Step 2 (Trace Generation): We consider GPS and network-based locations together to recover daily traces. The lower-accurate GSM and WiFi locations will compensate for GPS missing. Specifically, each user's locations are sorted by the timestamps to generate the user's daily traces. The network-based WiFi and GSM locations will fill the intervals of no GPS fixes. Thus, a trace is composed of several locations, each of which has a format of $\rho(t_k) = \langle t_k, \alpha(t_k), \beta(t_k), \phi(t_k)\rangle$. Here, t_k is the timestamp, $\alpha(t_k)$ is the latitude, $\beta(t_k)$ is the longitude, and $\phi(t_k)$ is the positioning source.

Step 3 (Stop Detection): This step analyzes where and when a user made stops. We cluster user locations into stops based on the spatial-and-temporal density and positioning sources. Let $\xi = \{\rho(t_k) = \langle t_k, \alpha(t_k), \beta(t_k), \phi(t_k)\rangle | t_k \in [t_0, t_{N-1}]\}$ denote a trace with N locations during the time interval of t_0 to t_{N-1}. we consider the following two principles for clustering.

- Two consecutive locations $\rho(t_i)$ and $\rho(t_{i+1})$ are regarded as two different stops if $\phi(t_i) \neq \phi(t_{i+1})$. In this case, the user may enter into or leave from an indoor place.
- Any two locations $\rho(t_i)$ and $\rho(t_{i+1})$ are clustered into a single stop if $|t_i - t_{i+1}| \leq T_{stop}$ and $D(\rho(t_i), \rho(t_{i+1})) \leq \Omega$, where T_{stop} is a predefined time window in the temporal domain and Ω is a predefined diameter in the spatial domain. Here, $D(\rho(t_i), \rho(t_{i+1}))$ is the distance between $\rho(t_i)$ and $\rho(t_{i+1})$.

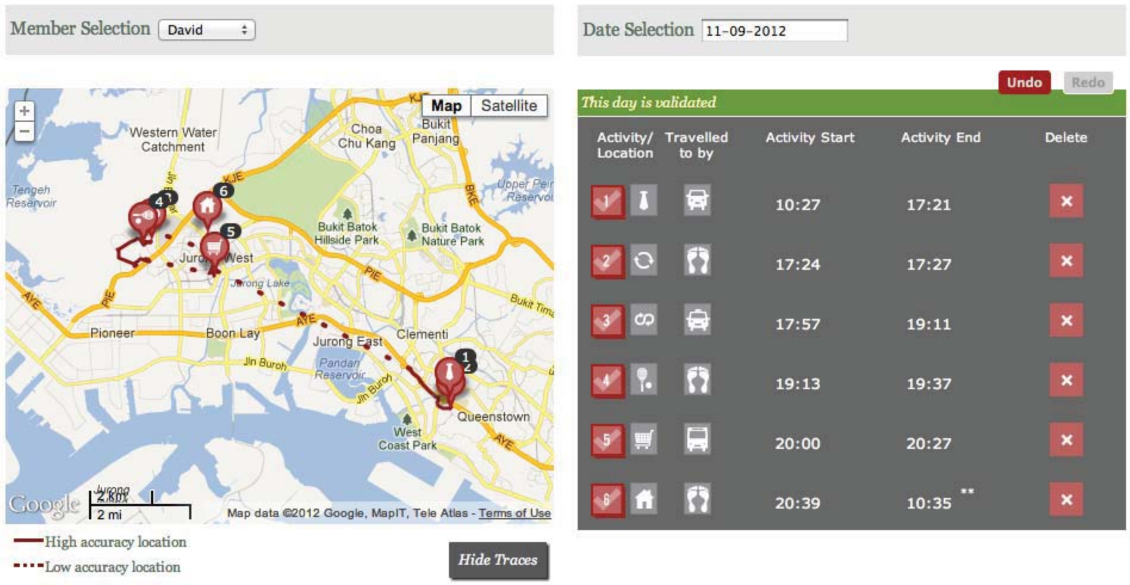
Step 4 (Detection of Transportation Modes): We construct a decision tree [25] to classify coarse-grained transportation modes including walking, MRT, buses, and cars. In the offline phase, we extract four features, (1) the maximal speed, (2) the average speed between stops, (3) the force variance of accelerometer data, and (4) the distance to the closest bus and MRT stops, to contract the decision tree. In the real-time phase, the transportation mode between two consecutive stops will be determined based on the decision tree. Note that the decision tree can be updated periodically (e.g., monthly).

Step 5 (Stop Calibration): The above steps may identify more than actual stops since the GSM cell ID changes even though the user is staying in the same location. We then consider the patterns of GSM changes and detected transportation modes to calibrate the detected stops.

- We merge two consecutive stops if the set of visible GSM cell IDs at the two stops are the same.
- For any three successive stops, denoted by S_{i-1}, S_i, S_{i+1}, we remove S_i if $M(S_{i-1}, S_i)$ and $M(S_i, S_{i+1})$ are the same mode, where $M(\cdot, \cdot)$ denotes the transportation mode between two consecutive stops. In this case, a vehicle may be waiting for traffic light or for passengers to alight at S_i which are of no interest.

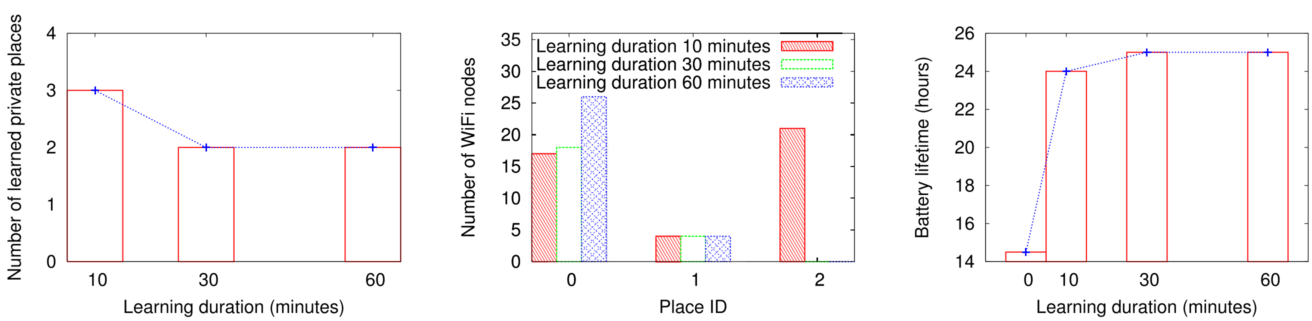
IV. Implementation and Experimental Results

Next, we explain the implementation details for real-world deployment and conduct extensive experiments to study smartphones' battery lifetime, the quantity and quality of collected data, and the accuracy of the transportation activity inference.



(a) (b)

Fig. 8. User interfaces of our web service. (a) The Web interface of mobility diary. (b) The Web interface for user validation.



(a) (b) (c)

Fig. 9. Experimental results of place learning and matching. (a) Number of learned private places. (b) Number of WiFi nodes in each private place. (c) Battery lifetime for different learning durations.

A. Mobility Sensing Platforms and Web Services

We implement the place-aware mobility sensing scheme as a mobile application which is always running in the background to collect mobility data seamlessly. Each smartphone collects accelerometer, GPS, WiFi, and GSM data. We collect accelerometer's data every 500 milliseconds, scan ambient WiFi signals every 30 seconds, and collect GSM data every 5 minutes. The learning duration Δ_L is 60 minutes for collecting WiFi signatures of private places. The similarity threshold for private place matching is $\tau_w = 1$, and the activity range for public places is $R_a = 300$ meters. For status detection, we consider $\tau_a = 0.25$ and $T_m = 5$ seconds. The parameters of data filters are $I_g = 1$ seconds and $I_w = 30$ seconds. The above parameters are the default values to conduct the following experiments. As our system must be able to scale up to analyze massive concurrent data arrival from a huge number of users, we implement the backend infrastructure using the MapReduce technology, where a master server allocates the data analysis tasks mentioned in Section III-C to several slave servers and itself in a load-balancing way. We adopt MySQL to implement the database. The data analysis tasks are implemented in Ruby programs. Fig. 8 shows our web service, where users are allowed to validate the detected stops, traces, and transportation modes via the web interface.

B. Experimental Results and Case Studies

We conduct extensive experiments to study how our sensing scheme affects place information, battery lifetime, and the quality and quantity of collected data. We also study the accuracy of transportation activity inference. We compare our system against a naive sensing scheme which always uses GPS when user is in moving status and uploads all of raw data.

1) Experimental Results of Place Learning and Matching: First, we change the learning duration Δ_L by 0 (i.e., without place learning), 10, 30, and 60 minutes to study the number of learned private places, the number of WiFi nodes included in each private place's signature, and the battery lifetime. In this experiment, we consider a single user's mobility data. As it can be seen in Fig. 9(a), the number of learned private places is bounded by the diversity of the user mobility, and a too short

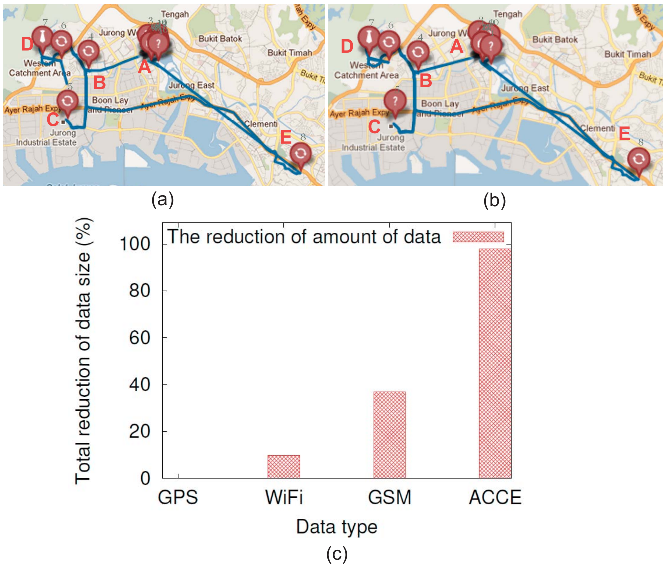


Fig. 10. Experimental results of the location-intensive data filters. (a) Data collected by the naive scheme. (b) Data collected by our scheme. (c) The reduction of data size.

learning duration may define too many infrequent and short-stay private places (e.g., a shopping mall). Fig. 9(b) shows the number of WiFi nodes included in each private place's signature. When the learning duration is 10 minutes, the smartphone can learn three different private places including home (with place ID 0), the office (with place ID 1), and the shopping mall (with place ID 2). Note that the smartphone only can know the place IDs and the corresponding WiFi signatures, while semantic place information (i.e., 'home', 'offices', and 'shopping malls') is the ground truth from the user. We can see that the longer learning duration may include more WiFi nodes into a private place's signature (e.g., place ID 0) due to WiFi signal variation. Because the visible WiFi networks are changed over time, the smartphone with the longer learning duration has more opportunities to scan more visible networks especially for those networks with weak signals. Finally, Fig. 9(c) shows that our scheme prolongs the battery lifetime significantly. The battery lifetime cannot be improved anymore when a larger learning duration is considered. This is because the smartphone only spends its energy for capturing real moving traces.

2) Experimental Results of Location-Intensive Data Filters: Next, we study the quality and quantity of collected data. As the sensor sensitivity is uncertain, in the following experiments, we collect data by the naive scheme and filter the collected data by our data filters to study the quality and quantity of data before and after filtering.

First, we study a single user's traces in a day to discuss the quality of collected data. Fig. 10 shows the collected traces, where the user travels from A to B by taking the MRT, along a round-trip trace between B and C and then to D by taking a bus, from D to B and then to E by taking a cab, and finally from E to A by taking the MRT. We can see that the less amount of data collected by our scheme can recover similar traces and stops generated by the the naive scheme. In Fig. 10(a), there is a jumping trace nearby D because the naive scheme still collects the lower-accurate GSM data between successive GPS and WiFi data to generate traces. Our scheme relieves such situations, as shown in Fig. 10(b).

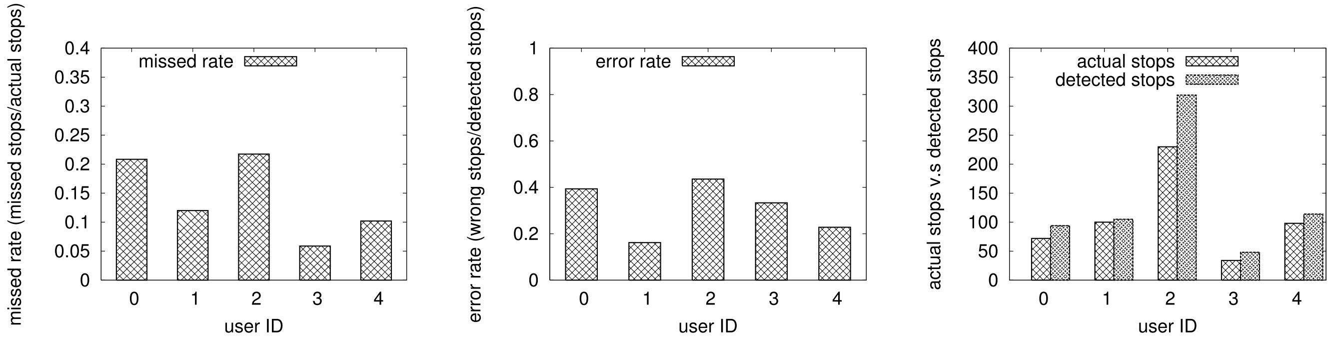
Then, we consider 177 users' traces in Singapore for 6 months to study the quantity of collected data. Fig. 10(c) shows the experimental results. The reduction of GSM data is more significant than the reduction of WiFi data because of dense WiFi signals around users. The reduction for accelerometer data is extremely significant because users are stationary most of the time in their daily life.

3) Accuracy of Transportation Data Analysis: Based on the validation provided by users, we study the accuracy of transportation data analysis. We consider 5 users' mobility data to study the accuracy of our stop detection algorithm. We consider two performance metrics, (a) the missed rate: the ratio of the number of undetected stops to the number of actual stops that the user has taken and (b) the error rate: the ratio of the number of wrong stops that have not taken by users but are detected by our algorithm to the number of stops detected by our algorithm. Fig. 11 (a) shows that the average missed rate is less than 0.25. Fig. 11 (b) shows that the error rate is less than 0.4. Fig. 11 (c) shows that the number of detected stops is slightly more than the number of actual stops.

V. Performance Evaluation

A. Accuracy Evaluation of Place Detection

We evaluate *sensitivity* and *specificity* to study the accuracy of place detection. Let Φ^+ and Φ^- denote positive and negative detection results, respectively. A positive detection result indicates that a user is detected in a particular place, while a negative detection result indicates that a user is not detected in a particular place. Here, we use $\Phi^+_{P_i}$ and $\Phi^-_{P_i}$ for each private place P_i to distinguish from $\Phi^+_{Q_j}$ and $\Phi^-_{Q_j}$ for each public place Q_j. Let Ψ^+ and Ψ^- denote positive and negative actual statuses, respectively. A positive actual status indicates that a user is actually in a particular place, while a negative actual status indicates that a user actually is not in a particular place. Similarly, $\Psi^+_{P_i}$ and $\Psi^-_{P_i}$ are for each private place P_i to distinguish from $\Psi^+_{Q_j}$ and $\Psi^-_{Q_j}$ for each public place Q_j. Then, we define the *sensitivity* of our place detection algorithm to a private place P_i as the probability, denoted by $\mathbb{P}(\Phi^+_{P_i}|\Psi^+_{P_i})$, that the detection result says a user in P_i when the user is actually in P_i. Here, $\mathbb{P}(\Phi^+_{P_i}|\Psi^+_{P_i}) = \frac{\Theta(\Phi^+_{P_i}|\Psi^+_{P_i})}{\Theta(\Phi^+_{P_i}|\Psi^+_{P_i})+\Theta(\Phi^-_{P_i}|\Psi^+_{P_i})}$, where $\Theta(\Phi^+_{P_i}|\Psi^+_{P_i})$ is the number of true positive detection and $\Theta(\Phi^-_{P_i}|\Psi^+_{P_i})$ is the number of false negative detection. Similarly, we define the sensitivity of our place detection algorithm to a public place Q_j as $\mathbb{P}(\Phi^+_{Q_j}|\Psi^+_{Q_j}) = \frac{\Theta(\Phi^+_{Q_j}|\Psi^+_{Q_j})}{\Theta(\Phi^+_{Q_j}|\Psi^+_{Q_j})+\Theta(\Phi^-_{Q_j}|\Psi^+_{Q_j})}$. Thus, we define the sensitivity of our place detection algorithm to private places and public places as $\mathbb{P}(\Phi^+_{\mathcal{F}}|\Psi^+_{\mathcal{F}}) = \frac{\Sigma_{P_i \in \mathcal{F}} \Theta(\Phi^+_{P_i}|\Psi^+_{P_i})}{\Sigma_{P_i \in \mathcal{F}} (\Theta(\Phi^+_{P_i}|\Psi^+_{P_i})+\Theta(\Phi^-_{P_i}|\Psi^+_{P_i}))}$ and $\mathbb{P}(\Phi^+_{\mathcal{H}}|\Psi^+_{\mathcal{H}}) = \frac{\Sigma_{Q_i \in \mathcal{H}} \Theta(\Phi^+_{Q_i}|\Psi^+_{Q_i})}{\Sigma_{Q_i \in \mathcal{H}} (\Theta(\Phi^+_{Q_i}|\Psi^+_{Q_i})+\Theta(\Phi^-_{Q_i}|\Psi^+_{Q_i}))}$, respectively.



(a) (b) (c)

Fig. 11. (a) The missed rate of stop detection. (b) The error rate of stop detection. (c) The number of detected stops versus the number of actual stops.

So, the overall sensitivity of our algorithm is defined as

$$\mathbb{P}(\Phi^+|\Psi^+) = \frac{\Sigma_{U_i\in\mathcal{F}\bigcup\mathcal{H}}\Theta(\Phi^+_{U_i}|\Psi^+_{U_i})}{\Sigma_{U_i\in\mathcal{F}\bigcup\mathcal{H}}(\Theta(\Phi^+_{U_i}|\Psi^+_{U_i})+\Theta(\Phi^-_{U_i}|\Psi^+_{U_i}))}.$$

Next, we define the *specificity* of our place detection algorithm. The specificity of our place detection algorithm to a private place P_i is defined as the probability $\mathbb{P}(\Phi^-_{P_i}|\Psi^-_{P_i}) = \frac{\Theta(\Phi^-_{P_i}|\Psi^-_{P_i})}{\Theta(\Phi^+_{P_i}|\Psi^-_{P_i})+\Theta(\Phi^-_{P_i}|\Psi^-_{P_i})}$ that the detection result says a user not in P_i when the user actually is not in P_i, where $\Theta(\Phi^+_{P_i}|\Psi^-_{P_i})$ is the number of false positive detection and $\Theta(\Phi^-_{P_i}|\Psi^-_{P_i})$ is the number of true negative detection. Similarly, the specificity of our place detection algorithm to a particular public place Q_j is defined as $\mathbb{P}(\Phi^-_{Q_j}|\Psi^-_{Q_j}) = \frac{\Theta(\Phi^-_{Q_j}|\Psi^-_{Q_j})}{\Theta(\Phi^+_{Q_j}|\Psi^-_{Q_j})+\Theta(\Phi^-_{Q_j}|\Psi^-_{Q_j})}$. Thus, the specificity of our place detection algorithm to private places and public places are defined as $\mathbb{P}(\Phi^-_{\mathcal{F}}|\Psi^-_{\mathcal{F}}) = \frac{\Sigma_{P_i\in\mathcal{F}}\Theta(\Phi^-_{P_i}|\Psi^-_{P_i})}{\Sigma_{P_i\in\mathcal{F}}(\Theta(\Phi^+_{P_i}|\Psi^-_{P_i})+\Theta(\Phi^-_{P_i}|\Psi^-_{P_i}))}$ and $\mathbb{P}(\Phi^-_{\mathcal{H}}|\Psi^-_{\mathcal{H}}) = \frac{\Sigma_{Q_i\in\mathcal{H}}\Theta(\Phi^-_{Q_i}|\Psi^-_{Q_i})}{\Sigma_{Q_i\in\mathcal{H}}(\Theta(\Phi^+_{Q_i}|\Psi^-_{Q_i})+\Theta(\Phi^-_{Q_i}|\Psi^-_{Q_i}))}$, respectively. So, the overall specificity of our algorithm is defined as

$$\mathbb{P}(\Phi^-|\Psi^-) = \frac{\Sigma_{U_i\in\mathcal{F}\bigcup\mathcal{H}}\Theta(\Phi^-_{U_i}|\Psi^-_{U_i})}{\Sigma_{U_i\in\mathcal{F}\bigcup\mathcal{H}}(\Theta(\Phi^+_{U_i}|\Psi^-_{U_i})+\Theta(\Phi^-_{U_i}|\Psi^-_{U_i}))}.$$

Below, we conduct experiments to study the accuracy of place detection. We consider 4 indoor places and 4 outdoor places. The 4 indoor places include 2 private places learned by the smartphone and 2 non-private places selected randomly. The 4 outdoor places include 2 MRT stations in the public place profile and 2 non-public places selected randomly. Since our sensing design keeps the ground truth of personal places in the smartphones of users who are involved in the real-world campaign, this experiment considers a limited dataset. We conduct the experiment for 20 times in each place, and the experimental results are shown in Fig. 12. The sensitivity of our algorithm to private places is $\mathbb{P}(\Phi^+_{\mathcal{F}}|\Psi^+_{\mathcal{F}}) = \frac{40}{40+0} = 100\%$, while the specificity of our algorithm to private places is $\mathbb{P}(\Phi^-_{\mathcal{F}}|\Psi^-_{\mathcal{F}}) = \frac{40}{0+40} = 100\%$. The sensitivity of our algorithm to public places is $\mathbb{P}(\Phi^+_{\mathcal{H}}|\Psi^+_{\mathcal{H}}) = \frac{39}{39+1} = 97.5\%$, while the specificity of our algorithm to public places is $\mathbb{P}(\Phi^-_{\mathcal{H}}|\Psi^-_{\mathcal{H}}) = \frac{38}{2+38} = 95\%$. So, the overall sensitivity is $\mathbb{P}(\Phi^+|\Psi^+) = \frac{79}{80} = 98.75\%$, and the overall specificity is

Actual statuses	Detection results	
	Φ^+_F	Φ^-_F
Ψ^+_F	40	0
Ψ^-_F	0	40

(a)

Actual statuses	Detection results	
	Φ^+_H	Φ^-_H
Ψ^+_H	39	1
Ψ^-_H	2	38

(b)

Fig. 12. Experimental results of place detection. (a) Experimental results of private place detection. (b) Experimental results of public place detection.

$\mathbb{P}(\Phi^-|\Psi^-) = \frac{78}{80} = 97.5\%$. Apparently, GPS localization errors affect accuracy of public place detection.

B. Complexity Analysis of Place Matching

Next, we analyze the computation cost of the place matching. First, we analyze the computation cost of the private place matching. Let denote P_{max} denote the private place with the maximal number of WiFi nodes and $|P_{max}|$ denote the number of WiFi nodes in P_{max}. When the smartphone receives a real-time network signals at time t_k, denoted by $R(t_k)$, for each $P_i \in \mathcal{F}$, the computation cost is $O(|P_{max}| \cdot |R(t_k)|)$ to check if $|R(t_k) \bigcap P_i| \geq \tau_w$. Since the smartphone must consider all of private places in $\mathcal{F}$ to perform the private place matching, the total computation cost for the private place matching is $O(|P_{max}| \cdot |R(t_k)| \cdot |\mathcal{F}|)$. Then, we analyze the computation cost of the public place matching. When a GPS fix Ω arrives, the smartphone must consider all of public places in $\mathcal{H}$ to check whether the user is within the activity range of a public place. This takes $O(|\mathcal{H}|)$ computation cost. Finally, the total computation cost of the place matching is $O(|P_{max}| \cdot |R(t_k)| \cdot |\mathcal{F}| + |\mathcal{H}|)$ when the smartphone receives $R(t_k)$ and Ω at the same time. Note that the number of visible networks in a particular place is usually limited (i.e., $|P_{max}|$ is limited). Then, we can consider a predefined threshold to limit the number of learned WiFi nodes for a private place so as to reduce the computation cost.

VI. DISCUSSION

Our system bridges mobile sensing technologies and a real-world application to understand human mobility better. Below, we discuss the challenges and open issues for our future work.

A. Real-Time Versus Opportunistic Perception Technologies

Although a 'real-time' sensing and uploading approach can provide the more accurate urban mobility information, it will impose high energy cost on smartphones. On the other hand, the mobility survey is a 'soft real-time application' in the sense that data collection is delay-tolerant. For sensing technologies, we then suggest an intermittent and opportunistic sensing in this work which collects mobility data only when a user makes transportation-related movements from a place to another place. For uploading technologies, we consider an opportunistic data uploading instead of a real-time uploading because the connection is not always available. Moreover, a user may prefer to upload data only when the smartphone is charged (e.g. in long-stay places) for the energy-saving purpose. In the future, a user may maintain an '*action profile*' to launch tasks when the smartphone detects a long-stay place. For example, a user defines an uploading task in the action profile only when the place has good connection and the smartphone is charged that can be detected by the smartphone itself automatically. With the action profiles in the long-stay places, the system will not spend energy for uploading.

B. Single-Modal Versus Multi-Modal Positioning Technologies

Our system considers the positioning technologies using smartphones, where the accuracy provided by the fine-grained GPS positioning system is around 5-10 meters, the WiFi positioning system provides an accuracy of 30-50 meters, and the coarse-grained cell-based positioning system has an accuracy of several hundreds of meters [26]. However, since a single positioning source is not able to fully support our transportation activity surveys because of the availability of positioning sources. For example, GPS cannot work in an underground environment. So, multi-modal positioning sources cooperates together to capture user mobility in our current system. Moreover, a combined positioning technology with multi-modal positioning sources could be considered [27] to improve the positioning accuracy in the future.

VII. CONCLUSION

Motivated by a real-world application, we have designed the UrbanMobilitySense system which seamlessly captures human mobility in urban areas. We have addressed privacy and energy issues in this work, where personal place information is protected by two separate profiles for optimizing energy usage. The system has been deployed in Singapore to support the Land Transport Authority's transportation activity surveys.

REFERENCES

[1] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems," *Pervasive Mobile Comput.*, vol. 7, no. 4, pp. 397–413, 2011.

[2] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The BikeNet mobile sensing system for cyclist experience mapping," in *Proc. ACM 5th Int. Conf. Embedded Netw. Sensor Syst.*, 2007, pp. 87–101.

[3] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, "Wireless sensor networks in intelligent transportation systems," *Wireless Commun. Mobile Comput.*, vol. 9, no. 3, pp. 287–302, 2009.

[4] F.-J. Wu, H. B. Lim, F. Pereira, C. Zegras, and M. Ben-Akiva, "A user-centric mobility sensing system for transportation activity surveys," in *Proc. 11th ACM Int. Conf. Embedded Netw. Sensor Syst.*, 2013, pp. 1–2, Art. ID 74.

[5] Y. Xiao *et al.*, "Transportation activity analysis using smartphones," in *Proc. IEEE Consum. Commun. Netw. Conf.*, Jan. 2012, pp. 60–61.

[6] Y. Chon, E. Talipov, H. Shin, and H. Cha, "Mobility prediction-based smartphone energy optimization for everyday location monitoring," in *Proc. 9th ACM Int. Conf. Embedded Netw. Sensor Syst.*, 2011, pp. 82–95.

[7] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *Proc. 8th ACM Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 299–314.

[8] C.-H. Lo, W.-C. Peng, C.-W. Chen, T.-Y. Lin, and C.-S. Lin, "CarWeb: A traffic data collection platform," in *Proc. 9th Int. Conf. Mobile Data Manage.*, Apr. 2008, pp. 221–222.

[9] M. Azizyan, I. Constandache, and R. R. Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. ACM 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 261–272.

[10] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," *Artif. Intell.*, vol. 171, nos. 5–6, pp. 311–331, 2007.

[11] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on GPS data for web applications," *ACM Trans. Web*, vol. 4, no. 1, pp. 1–36, 2010, Art. ID 1.

[12] E. Miluzzo *et al.*, "Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application," in *Proc. ACM 6th Int. Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 337–350.

[13] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, pp. 1–27, 2010, Art. ID 13.

[14] A. Thiagarajan *et al.*, "VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. ACM 7th Int. Conf. Embedded Netw. Sensor Syst.*, 2009, pp. 85–98.

[15] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones," in *Proc. ACM 8th Int. Conf. Embedded Netw. Sensor Syst.*, 2010, pp. 85–98.

[16] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proc. ACM 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 379–392.

[17] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, "EasyTracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones," in *Proc. ACM 9th Int. Conf. Embedded Netw. Sensor Syst.*, 2011, pp. 68–81.

[18] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "SignalGuru: Leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proc. ACM 9th Int. Conf. Mobile Syst., Appl., Services*, 2011, pp. 127–140.

[19] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personally meaningful places: An interactive clustering approach," *ACM Trans. Inf. Syst.*, vol. 25, no. 3, 2007, Art. ID 12.

[20] U. Ahmad, B. J. d'Auriol, Y.-K. Lee, and S. Lee, "The election algorithm for semantically meaningful location-awareness," in *Proc. 6th Int. Conf. Mobile Ubiquitous Multimedia*, 2007, pp. 55–63.

[21] D. H. Kim, J. Hightower, R. Govindan, and D. Estrin, "Discovering semantically meaningful places from pervasive RF-beacons," in *Proc. 11th Int. Conf. Ubiquitous Comput.*, 2009, pp. 21–30.

[22] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "SensLoc: Sensing everyday places and paths using less energy," in *Proc. ACM 8th Int. Conf. Embedded Netw. Sensor Syst.*, 2010, pp. 43–56.

[23] D. H. Kim, K. Han, and D. Estrin, "Employing user feedback for semantic location services," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 217–226.

[24] *PowerTutor: A Power Monitor for Android-Based Mobile Platforms*. [Online]. Available: http://ziyang.eecs.umich.edu/projects/powertutor/index.html, accessed Oct. 9, 2011.

[25] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Reading, MA, USA: Addison-Wesley, 2005.

[26] S. von Watzdorf and F. Michahelles, "Accuracy of positioning data on smartphones," in *Proc. 3rd Int. Workshop Location Web*, 2010, pp. 1–4, Art. ID 2.

[27] S. Zirari, P. Canalda, and F. Spies, "WiFi GPS based combined positioning algorithm," in *Proc. IEEE Int. Conf. Wireless Commun., Netw., Inf. Secur.*, Jun. 2010, pp. 684–688.

Fang-Jing Wu, photograph and biography not available at the time of publication.

Hock Beng Lim, photograph and biography not available at the time of publication.