Autonomous Management of Everyday Places for a Personalized Location Provider

Yohan Chon, Elmurod Talipov, and Hojung Cha, *Member, IEEE*

***Abstract*—Currently available location technologies such as the global positioning system (GPS) or Wi-Fi fingerprinting are limited, respectively, to outdoor applications or require offline signal learning. In this paper, we present a smartphone-based autonomous construction and management of a personalized location provider in indoor and outdoor environments. Our system makes use of electronic compass and accelerometer, specifically for indoor user tracking. We mainly focus on providing point of interest (POI) locations with room-level accuracy in everyday life. We present a practical tracking model to handle noisy sensors and complicated human movements with unconstrained placement. We also employ a room-level fingerprint-based place-learning technique to generate logical location from the properties of pervasive Wi-Fi radio signals. The key concept is to track the physical location of a user by employing inertial sensors in the smartphone and to aggregate identical POIs by matching logical location. The proposed system does not require *a priori* signal training since each user incrementally constructs his/her own radio map into their daily lives. We implemented the system on Android phones and validated its practical usage in everyday life through real deployment. The extensive experimental results show that our system is indeed acceptable as a fundamental system for various mobile services on a smartphone.**

***Index Terms*—Indoor tracking, inertial sensor, mobile sensing, place learning, smartphone.**

I. Introduction

LOCATION-BASED services are increasingly important for modern mobile devices such as the smartphone. Navigation, social network services, and sharing photos are common applications that utilize user location [1], [2]. These services make use of a temporary user location that is obtained at a certain period of time by manual request. However, emerging mobile services require an advanced localization scheme that would provide everyday location monitoring instead of temporarily locating the user. Many advanced services are available with information on everyday location monitoring. For example, health monitoring system utilizes a user's location to estimate the physical state of elderly person or patients [3]. The system reports daily momentum to improve their health. Another example is an environment-related application that estimates the user's environmental impact and exposure [4]. People learn how their lives affect the environment and subsequently alter their behavior to protect environment. Smart-home system determines the location of residents and recognizes living patterns to provide appropriate services [5], [6]. Life-logging has been proposed to visualize life patterns or to provide an automatically generated life diary [7]. Works in [8] and [9] also propose useful services that enable users to search for their lost mobile devices. The service uses daily location information of a user to track down lost phones or items. To support these kinds of emerging applications, efficient and accurate location monitoring is essential. However, currently available location technologies cannot fully provide user locations in everyday lives.

A mobile device is capable of locating itself based on various approaches. The global positioning system (GPS) is a common solution in outdoor environments. GPS is known to operate poorly in an indoor environment and consumes a relatively large amount of battery power. Cell tower localization with a Wi-Fi positioning system (WPS) is an effective alternative to GPS [10], [11]. The main idea is to preconstruct a radio map of Wi-Fi/Global System for Mobile Communications (GSM) access points (AP) through offline training. Upon a user request for location information, the mobile device detects surrounding Wi-Fi/GSM APs and searches for a matching element in a stored map. These techniques are sufficient to provide temporary location information. However, the following issues should be addressed for a practical solution in everyday location monitoring.

Location in indoor environments. Average users normally spend much of their time in indoor environments where a GPS signal is not available. WPS produces poor localization accuracy since the radio map is constructed using war-drive. Offline training in indoor environments is impractical because the cost is excessive and the coverage area is generally wide and sparse. Users are not likely to acquire precise location information at their point of interests (POIs) that are typically located indoors in modern society. Hence, a practical solution is necessary to cover the location of POIs in indoor environment at a low cost.

Logical location. The ability to determine the logical places is a necessary component to support advanced mobile services. The current locating service for mobile devices is based on physical location information, which is a series of raw coordinates or an address. Many mobile services, however, can be greatly improved by employing information on logical places that are semantically meaningful to the user. Physical location is not sufficient to distinguish places: Physical location of POI is not necessarily generated exactly at the same point, and two-dimensional (2-D) geocode is useless to identify places located

Manuscript received September 2, 2010; revised March 8, 2011; accepted March 14, 2011. Date of publication April 21, 2011; date of current version June 13, 2012. This work was supported by the National Research Foundation of Korea, funded by the Korean Government, Ministry of Education, Science, and Technology, under Grant 2010-0000405. This paper was recommended by Associate Editor B. Chaib-draa.

The authors are with the Department of Computer Science, Yonsei University, Seoul 120-749, Korea (e-mail: yohan@cs.yonsei.ac.kr; elmurod@cs.yonsei.ac.kr; hjcha@cs.yonsei.ac.kr).

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/TSMCC.2011.2131129

closely on different floors. The logical place information includes the name of a store or a personalized meaning, for example, "workplace," "study place," or "attractions." The role of logical information is to recognize visited places with room-level accuracy.

Privacy issues. The centralized location provider is exposed to privacy invasion since the server easily knows a user's location. In particular, everyday location monitoring could be privacy invasive since the system may collect sensitive information that concerns individual location [12]. Accordingly, a personalized solution is necessary.

Energy consumption. Previous research shows that GPS and Wi-Fi operations severely affect a smartphone's battery life [13]. In order to employ location provider in everyday location monitoring, this energy issue should be considered as a primary factor.

In this paper, we propose an autonomous construction of a personalized POI map, named LifeMap, which provides location information for advanced mobile services. Based on our preliminary work [14], we have expanded LifeMap and extensively validated the system with real deployments. Our objective is to discover POIs and to provide locations of POIs in everyday life without a centralized server. The key concept is to exploit an accelerometer and electronic compass to track user location, and to aggregate identical POIs by matching logical location that is generated from the properties of pervasive Wi-Fi APs. The system recognizes user context to estimate accuracy of location, and the aggregation process utilizes measured accuracy to refine location information. The system incrementally constructs user's POIs with a personalized radio map. The proposed system is realized, with off-the-shelf technologies, to be a practical location provider for everyday location monitoring.

The contribution of our system is as follows.

1) Our study is an early attempt to track indoor locations of users by using an accelerometer and electronic compass in smartphones. Tracking human movements only with an inertial sensor is technically difficult when considering the unconstrained placement of mobile phone, noisy and inaccurate sensors, and low computation power in mobile devices. We present the possibility and limitations to use inertial sensors in smartphones through extensive experimental analysis.
2) LifeMap automatically constructs a personalized radio map without offline training. We employ dead reckoning to track a mobile user during the transition from outdoor to indoor, and compensate cumulative errors of dead reckoning by using logical location. The accuracy of physical/logical location in a radio map increases with the number of visits to places. A preconstructed radio map, such as SkyHook [11], is optional in our scheme.
3) LifeMap is implemented on a commercial mobile phone platform. We combine existing localization methods to implement a practical location provider. In particular, the system considers energy consumption and privacy issues for practical usage. We validate LifeMap through real deployment and experiments for the duration of 8 weeks.

The structure of this paper is as follows: In Section II, we briefly describe the system overview of LifeMap. The smartphone-based dead reckoning is presented in Section III. Section IV describes room-level fingerprint-based place learning. We discuss detailed design considerations for privacy and energy consumption in Section V, and the implementation of LifeMap is presented in Section VI. In Section VII, we evaluated the system in three aspects: dead reckoning, place learning, and overall systems. We discuss relevant works and observations/limitations of our system in Section VIII. Finally, we conclude this paper in Section IX.

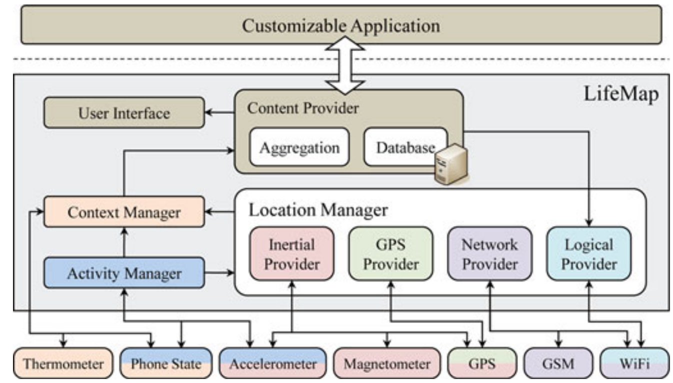

Fig. 1. LifeMap system overview. Each component generates high-level information using several sensors.

II. System Overview

LifeMap is a personalized POI map that offers physical and logical location information. The system is designed to provide locations to users with low latency, minimum energy consumption, and privacy protection. We implement LifeMap on Android phones that are equipped with several built-in sensors such as an accelerometer, electronic compass, GPS, Wi-Fi, and GSM, all of which are common components in the latest smartphones. Fig. 1 shows the overview of the LifeMap system. The major components of LifeMap are the activity manager, location manager, context manager, and content provider.

The activity manager plays a key role as the scheduler of the system. This component recognizes the activity of a user using the accelerometer and the phone state. We employ the variance of an accelerometer signal to monitor change of user activity that includes "moving" and "stationary." Based on this classified activity, the activity manager controls other components to generate user context, including location information. When a user starts to move, the location manager provides classification data to approximate the user's location. When a user stays at a location for a certain period of time, the context manager generates user context from all components, such as location manager, activity manager, and the phone state. The system immediately deactivates each sensor after obtaining the necessary context information. The relationship between a user's activity and a set of components is predefined to minimize energy consumption. Fig. 2 illustrates the activity-based decision rules used to dynamically provide location information.

The location manager provides physical and logical location information. The component comprises several location

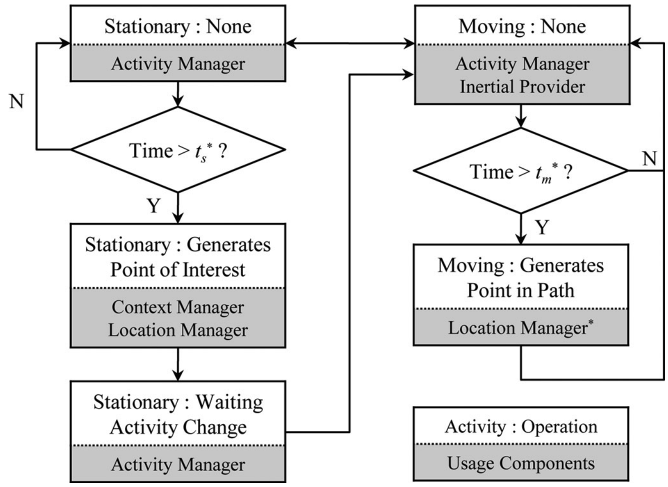

Fig. 2. Decision rules of an activity-based, event-driven approach. The default thresholds are t_s =10 min and t_m =3 min. Users can modify these thresholds by using UI. The location manager does not use logical provider in moving (* t_s : threshold time in stationary, t_m : threshold time in moving state).

Fig. 3. Categorized phone usage and related positions. Described cases are hand holding, watching a video, talking on the phone, in pants pocket, in hip pocket, in jacket pocket, in jacket inside pocket, and inside bag, respectively, from the left.

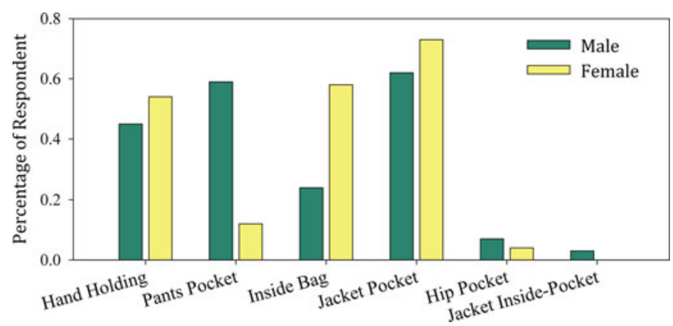

Fig. 4. Favorite position of mobile phones when people move. Participants chose their two most favorite positions, while they were moving around with mobile phones.

providers, as shown in Fig. 1. We obtain physical location information from the inertial provider, the GPS provider, and the network provider. Each provider generates the location information along with an error bound that reflects the accuracy. The inertial provider utilizes an accelerometer and an electronic compass to implement smartphone-based dead reckoning. We used the GPS provider and the network provider that were already implemented in the Android protocol stack. The logical provider determines the logical identification of a place and decides if the current location is a place the user has previously visited.

The context manager produces user context by composing the location information, the activity, and the environmental context. This component generates context nodes and edges to construct a context map, which is represented as a graph. The context map is stored in a database to aggregate and represent user context.

A content provider aggregates generated data into refined data. This component manages the database to provide data to the internal user interface (UI) and other applications. Instead of inferring the personalized logical meaning of places automatically, we provide a place categorization service to induce spontaneous place labeling by a user.

III. Smartphone-Based Dead Rckoning

Dead reckoning is a process that is designed to estimate current location based on a previously determined one. A representative dead reckoning is an inertial measurement unit (IMU)-based method that uses integrations of acceleration measurements. Since the method estimates current location without communication with other components, this technique can supplement the weakness of currently available location providers that are dependent on external infrastructures, such as satellites or Wi-Fi APs. One inherent problem of the IMU-based method, however, is accumulated error, because the process continually adds relative locations to its previous location.

Previous work on IMU-based systems usually assumes a strapdown usage in which the user should fasten inertial sensors around the waist or shoes. Foot-mounted IMU is widely used to revise accumulated errors through a zero-velocity correction method [15]. This method corrects the cumulative error using the stationary moment during each step. Obviously, the foot-mounted method is not a practical approach for smartphone usage, and the scheme demands nontrivial computation to filter noise signals. A gyroscope could be used to enhance tracking accuracy, but even the latest smartphones do not employ a gyroscope due to its relatively high cost. To apply an IMU-based dead reckoning to a smartphone, a new approach should, therefore, be developed that specifically considers unconstrained placement, low processor power, and low efficiency of inertial sensors in the device.

A. Tracking Model

The position of a smartphone on the human body significantly affects the performance of dead reckoning. Although the user freely moves with the smartphone, we studied a limited case of smartphone positions, while a user moves. We categorized eight types of smartphone usage and related positions, as illustrated in Fig. 3. We then conducted surveys on 55 undergraduate students (29 males and 26 females) to find out their favorite carrying positions, while they move, and asked them to list their two most common positions when they move with mobile phones. The survey results, as shown in Fig. 4, indicate that people rarely carry their phones in hip pockets or jacket inside pockets. Based on the survey result, we only focused on four cases to track users' movements, instead of covering all possible positions.

The tracking model should generally deal with three issues: when a user moves, how far a user goes, and where a user goes. Our tracking model assumes that a user only makes forward movement, since the other movement types, such as backward and sidesteps, are rarely monitored in the everyday life of a typical user.

To determine when a user moves, we used the standard deviation of acceleration, which has been a widely used technique for motion detection [16], [17]. Given an accelerometer input vector $\vec{v}_t$ at time t, movement is detected if $\sigma(\vec{v}_{t-W}, \ldots, \vec{v}_t)$ is greater than threshold (e.g., 0.49 m/s^2) for the duration W, which is defined as "moving" state; otherwise, state is "stationary." A peak-detection algorithm [18] is then applied to mark the candidate points of time that have the possibility of user stepping. We utilize the means and variance of acceleration to generate worthy candidates that digress significantly from the average. If $\vec{v}_t \geq \mu(\vec{v}_{t-W}, \ldots, \vec{v}_t) + \sigma(\vec{v}_{t-W}, \ldots, \vec{v}_t)$, then $\vec{v}_t$ is detected as a peak. However, the peak-detection algorithm is prone to miscount steps during irregular behavior, such as swaying, vibration, impact, etc. To address this drawback, we considered the fact that user steps are periodically repeated, and device orientation is usually unchanged, while a user is stepping.

In order to recognize the periodicity of detected peaks, we compare the *max axis* and the *forward axis* of the current peak with those of the previous peak. The max axis is the axis perpendicular to the ground among the three axes of the accelerometer, computed as $\arg\max\{\acute{v}_{x,t}, \acute{v}_{y,t}, \acute{v}_{z,t}\}$, where $\acute{v}_{x,t}, \acute{v}_{y,t}$, and $\acute{v}_{z,t}$ are the absolute average values of accelerometer readings for the x-, y-, and z-axes, respectively, at time t with window W. The acceleration of the max axis is the cardinal value, because the movement of the user does not generate a larger acceleration than the acceleration of gravity. The forward axis is defined as the most similar axis to the direction of user movement, and determined by its variance since walking motion influences the direction of movement more than the orthogonal direction. The forward axis is computed as $\arg\max\{\hat{v}_1, \hat{v}_2\}$, where $\hat{v}_1 = \sigma(v_{1,t-W}, \ldots, v_{1,t})$ and $\hat{v}_2 = \sigma(v_{2,t-W}, \ldots, v_{2,t})$ are the accelerometer variance for other two axes after the max axis is excluded. Then, *abstract orientation* is defined as $\{$*max axis, forward axis*$\}$, which is a combination of max axis and forward axis. In contrast with the original orientation (i.e., roll, pitch, and yaw), which requires matrix transformation, abstract orientation is computed in $O(1)$ and stable over the noisy signal. In summary, a peak is considered as a step if the abstract orientation of the current peak is equal to that of the previous peak. Our scheme would still misstep counts that occurred when a user changes his/her motion, e.g., switches from calling to messaging. We can, however, obtain the overall trend of the user's movement, which is in fact our main interest.

Now, we discuss issues with regard to obtaining information on where the user is heading. Since transforming local acceleration into a global one is a nontrivial operation for smartphones, we restrict the direction of user movement within the line of the sensor's axis. Consequently, vector transformation in three-dimension (3-D) is reduced to a deterministic problem among six directions; i.e., three axes are multiplied by two directions, as illustrated in Fig. 5. Based on this idea, we first find abstract orientation. Then, the issue of whether the movement is forward or backward should be resolved. The velocity, which is obtained from integrating the acceleration, is not useful to solve this problem because gravity cannot be eliminated thoroughly due to the noisy sensors in smartphones. The low frequency of the sensors also causes inaccurate values of the integration. Therefore, we employed a user context, which is derived from the phone usage and the position of the smartphone, to ascertain the direction of movement.

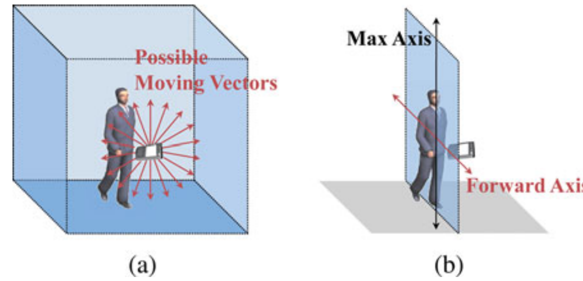

Fig. 5. Effects of determining a max axis and a forward axis. (a) Determining the direction of user movement in a 3-D space is transformed into (b) a deterministic problem that is stable against the noisy signal.

We categorized the user context into three types: hand-held position of smartphone, a strapdown position, and swing phase in pockets. The key factors to categorize user context are the variance of acceleration, device orientation, and phone state. The variance tends to become larger when the position of the smartphone is closer to the lower part of the body. The pants-pocket case has the maximum amount of variance because this position is directly affected by a stepping impact and a gait cycle. On the other hand, the hand-held position of a smartphone has a minimum variance. Abstract orientation also indicates user context. For example, a calling case has a unique orientation, and the horizontal positioning of a smartphone may not be possible in the pocket case. In addition, mobile phones usually provide a status of communication module, such as cellular and network state, which is useful to infer user context. User context being detected, we apply different factors to each type to determine the direction of movement.

1) *Using a smartphone with hand holding.* We can easily determine the direction of movement, while a user is messaging, watching a video on the phone, or talking over the phone, because the direction of movement directly relates the abstract orientation.
2) *In swing phase.* According to the body-segment analysis of gait cycles, we used skewness as a basic factor to decide the direction of movement, since the time for acceleration is generally shorter than the stepping time while a human walks [19].
3) *In strapdown position.* Deciding the direction of movement in strapdown position, including inside a bag and a jacket pocket, is difficult since the smartphone is randomly positioned and hardly affected by gait cycle. We use skewness as a basic standard, but an additional method is required to supplement inaccurate results of the direction of movement.

Since the acquired direction of movement is unstable, the direction should be further enhanced. We revise the generated direction when the GPS signal gives the heading information. Since we restrict movement direction within the sensor's axis, four directions on horizontal planes are candidates as the direction of movement of a user. The amendment method selects the most similar direction to heading information of the GPS signal. The method maintains the revised direction until the abstract orientation is changed. The GPS amendment is used during the transition from outdoor to indoor.

The moving distance is derived by multiplying the step count by the step length, which is estimated with the user's height or manually input by the user. We initiate the error bound with the accuracy of the GPS signal. When a GPS signal is not available, the system uses the summation of the previous accuracy and the measured moving distance to infer the error bound of the estimated location. We ordered the cases of categorized context by the amount of uncertainty involved; i.e., in decreasing order of uncertainty, the strapdown position, swing phases, hand-holding position, and the availability of the heading information of the GPS signal. Depending on users' contexts, we increase the error bound, at most to twice the step length and at least the step length, which are loosely estimated to include the actual location within the error bound.

In summary, we simplified the tracking problem of 3-D vector transformation into a deterministic problem that models human movement as step counting and heading determination. One shortcoming of our tracking model is an error caused by discordance between the user's direction of movement and the axis of the inertial sensors. If the device is on a slant to the actual direction of movement, a 45° maximum error is expected, although the direction of movement is determined appropriately. To comprise this issue, we increase the error bound of tracking results, in accordance with categorized context ordered by the uncertainty involved. Consequently, the aggregation process in our system incrementally refines the location information and eventually reduces the error as small as possible. We presented this issue and evaluated our tracking model in Section VII-A.

IV. Place Learning in Logical Provider

Place learning is to discover semantically meaningful POIs, and to determine logical identification of POIs. Most of the current location technologies simply provide geographical location. This information alone is insufficient for place learning, because the physical location is not generated exactly at the same place, despite the fact that a user generally has a similar life pattern every day. In addition, physical information is not enough to distinguish places that have similar geocode, in different floors, for example. Logical identification is an effective alternative to physical location. With logical information, the system can determine if two places are logically identical, although geographical coordination is unclear.

Place learning considers discovering and identifying POIs. In order to discover POIs, we exploit the accelerometer signal and stable scan of pervasive Wi-Fi APs. The system uses room-level fingerprint to generate/identify logical location. We define POI as a location that a user has stayed for a certain period of time. The "stationary" activity is easily monitored by using the variance of the accelerometer signal. When the "stationary" activity is continuously maintained, the place is considered as POI. We identify three types of places: a visited place, a new place, and not a place. Deciding a place is done by monitoring the stable basic service set identifier (BSSID) scan for a fixed window time [20]. However, determining a visited place is a relatively difficult task due to signal interference and environmental changes in indoor environments [21]. A system can generate a significantly different received signal strength (RSS) vector if the user visits the same place at different times of day. In order to overcome these challenge, we used a room-level fingerprint scheme for place learning and optimized an RSS model that reflects the nature of pervasive Wi-Fi APs in indoor environments.

When a stationary state is continuously maintained, the place is considered as a POI $l_{\rm new}$. Then, Wi-Fi APs are scanned every 10 s for the fixed time window (e.g., 1 min) to generate a logical location $\mathcal{A}_{\rm new}$ that is defined as

$$\mathcal{A}_{\rm new} = \{a_i, a_{i+1}, \ldots, a_n\}.$$

Here, n is the number of AP, and a_i is the $i_{\rm th}$ AP that contains BSSID, service set identifier, number of scan count, mean of signal strength, standard deviation of signal strength, and last scanned time. We used a scan window to perform multiple scans to tolerate noisy signals [20], [22]. Given the observed POI set $\mathbb{L} = \{l_0, l_1, l_2, \ldots, l_N\}$, room-level identification is classified as

$$\text{classify}(l_{\rm new}) = \begin{cases} \text{visited}, & \text{if}(\bigvee_{k=0}^{N} \mathcal{S}(l_k, l_{\rm new}) = \text{true}) \\ \text{new}, & \text{else} \end{cases}$$

where $\bigvee$ is the logical "or" operation and $\mathcal{S}$ is a similarity decision function. $\mathcal{S}$ is computed by combining three methods: a weighted BSSID matching, a distribution comparison method, and the rms of signal difference.

Suppose that a similarity decision should be made for two locations, l_k and $l_{\rm new}$; we first calculate the weighted BSSID match rate, which is similar to Sørensen's similarity index [23]. Let $\mathcal{A}_k$ be the set of APs scanned at location l_k and $\mathcal{A}_{\rm new}$ the AP set scanned at $l_{\rm new}$; the BSSID match rate is computed as $\frac{\mathcal{A}_k \bigcap \mathcal{A}_{\rm new}}{\mathcal{A}_k + \mathcal{A}_{\rm new}}$. In the equation, the scanning count of APs is used. If the match rate between two places is low (e.g., below 0.5), we consider these places different ones. When the match rate exceeds a certain threshold (e.g., above 0.5), the signal strengths of each AP are compared.

We defined an AP as a "large-set AP" if the scanning count is sufficient to make a distribution (e.g., more than 20 times); otherwise, it is called a "small-set AP." We use the rms of signal differences to compare small-set APs, and the large-set APs are compared by their signal distributions. We model the noise variance of Wi-Fi signal as a Gaussian distribution, which is an efficient localization technique for use on resource-constrained devices, because of low computation overhead and modest storage requirements [24], [25]. We then surmise that the AP X's signal distribution, for example, contains a signal of AP X in other location within a range of ± 3 standard deviation to deal

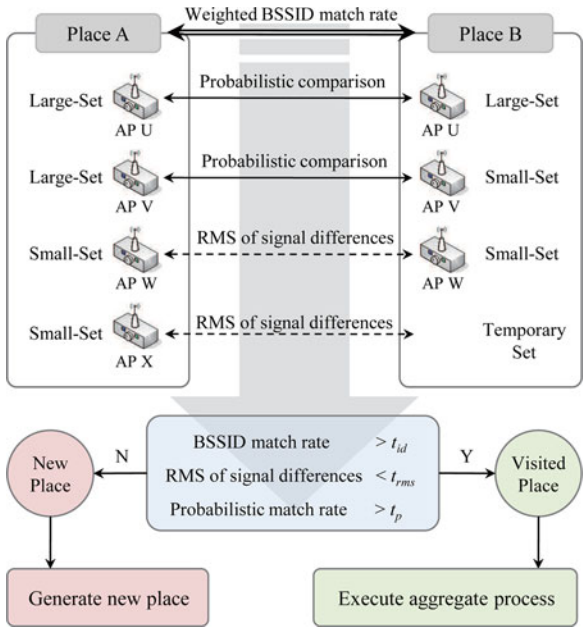

Fig. 6. RSS model and matching process. When one place does not have a specific AP, we compare it with the minimum signal strength (−100 dB) of the Android stack.

with the variation of the interference within 99.7% of the Gaussian distribution.

Fig. 6 summarizes the metrics used in the RSS model and matching process. The function for similarity decision is computed for all places within the error bound of the generated place. If a candidate place is classified as new, a new POI is generated and added to the POI set. Otherwise, physical location information that has a larger error bound is removed, and RSS values are aggregated. The metric is further evaluated in Section VII-B.

V. System Issues

A location provider in mobile devices should seriously consider issues of both energy consumption and privacy protection. The battery lifetime of a mobile device is a critical attribute especially in smartphone. Meanwhile, mobile phones are closely tied to daily life in modern society. People carry mobile phones wherever they go and store substantial amount of personal information in the device. Thus, sharing the information stored in mobile phones may cause a privacy problem. A location provider is particularly exposed to privacy concerns because the system easily knows a user's location.

A. Energy Consumption

Minimization of energy consumption of mobile devices should be carefully considered in continuous sensing such as everyday location monitoring. To design an energy-efficient mechanism, we conducted experiments to understand the power consumption of each component in a smartphone. We used PXI and LabVIEW from National Instruments to measure the current consumption of several components, such as the CPU, accelerometer, electronic compass, GPS, and Wi-Fi in the HTC G1 smartphone. PXI and the battery connector of the G1 were serially connected to measure current consumption. The measurement frequency was set to 5 kHz. Instead of using the battery, we used a dc power supply to deliver constant voltage (4.2 V) to maintain the measurement consistency. All background services in G1 were stopped to obtain an accurate measurement. Table I summarizes the energy consumption of each component. Since the accelerometer running shows low energy consumption, we used the accelerometer with the slowest frequency to recognize a user's activity (i.e., stay and move), which is used as a trigger to activate the other component. Fig. 7 illustrates the current consumption scenario that generates POI under our decision rules. The accelerometer is always ON because of its low current consumption, and other components are activated according to the decision rules. Although the GPS and Wi-Fi are useful components in locating mobile phones, the experiment revealed that GPS consumes a large amount of battery power, and scanning Wi-Fi APs is also a costly process. We present the smartphone's battery lifetime with LifeMap in Section VII-C.

TABLE I
Energy Consumption of Each Component In an HTC G1

Operation		Energy (J)	Note
CPU	Idle	1.60	
	Busy	27.73	Infinite loop
Accel	Level 1	4.36	2~4 Hz
	Level 2	8.04	8~10 Hz
	Level 3	19.24	16~18 Hz
	Level 4	21.93	28~32 Hz
GPS	ON	30.70	
Wi-Fi	Scan	4.03	Every 10 seconds

We averaged several measurements for one-minute durations.

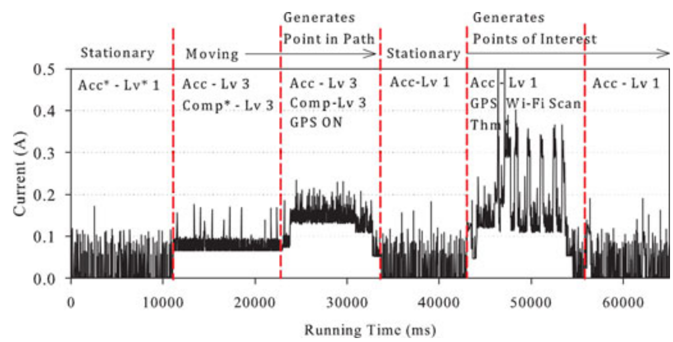

Fig. 7. Current consumption in an HTC G1. Each component is activated according to our decision rules. We set the time threshold to several seconds to present the overall trend (*Acc: accelerometer, Comp: electronic compass, Thm: thermometer, Lv: level).

B. Privacy Protection and Application

The location information of mobile users is critical to their privacy. A centralized system would not be acceptable to users who want to hide their location for privacy reasons. We designed a decentralized location provider for privacy protection. A preconstructed radio map in a centralized server is optional in our scheme, and the user entirely has the right to determine the level of sharing information. Users select one of four sharing levels: none, life-point sharing, logical sharing, and physical sharing. With the life-point sharing level, a user shares a category-related life point, which is described in Section VI. With logical

Fig. 8. UI of LifeMap. A user can make or select interest categories (a) and confirm the history to decide a goal point for each specific category in (c). The blue bar in (b) shows stay time, and the green bar shows by how much time a user's goal has been exceeded. The red bar shows where a user has not spent the desired amount of time. UI presents (d) usage information and (g) location information with generated context. A user chooses each place (e) on the map (e) or (f) on the list to confirm and modify the detailed information, such as place category and place name. (h) Level of information shared is chosen by a user. (a) Category select. (b) Life-point view. (c) Life-point history. (d) Daily information. (e) Map and menu. (f) List of places. (g) Detailed information for a specific place. (h) Sharing information level select.

sharing, users reveal the name of a user-defined category of place, such as a workplace, food place, etc. At the physical sharing level, a user shares his/her complete location information set, including latitude and longitude.

Application availability depends on the sharing level that the user would choose. Users can compare their life points in social networking services at the "life-point sharing" level; this way, we know who is a workaholic and who spends most of his/her time studying. At the "logical sharing" level, intelligent advertisement services are possible. This service automatically provides sales information or discount coupons in a timely manner when user visits the store. Similarly, logical location information is useful for location-based phone profiling. If a user decides to choose the "physical sharing" level on his or her children's devices, security companies can ensure the safety of the users' children. In addition, users may share useful geographical information, such as locations of open APs or ride-sharing information. Even if users decide not to share their information, they can still use LifeMap to check their life pattern. Battery management is also a possible application, because the system analyzes the generated data that reflect a user's life pattern. LifeMap contributes to improving the quality of human life via the implementation of various applications.

VI. Implementation

We implemented LifeMap on the Android protocol stack to evaluate the proposed scheme on commercial mobile phones. Android is an open-source software stack for mobile devices. We tested LifeMap on several Android phones equipped with various sensors, such as an accelerometer, electronic compass, GPS, Wi-Fi, and GSM. Our implementation particularly focused on intuitive UI design and place categorization to induce frequent user interaction and participation.

Although LifeMap determines the identification of generated places, the scheme cannot automatically infer the logical meaning of places for a specific user. Instead of inferring the logical meaning of places, we provide a place categorization that supports the user's lifestyle. First, the user chooses interesting categories from a category set, which may include home, workplace, study place, food place, gym, etc. The user connects his/her POI to the selected categories. Once the place is mapped to the categories, its logical meaning is applied to other places that have the same identification. The system records one life point every minute a user stays at a certain place that is mapped in specific categories. LifeMap visualizes the history of user's life points daily, weekly, and monthly. For example, a user sees how much time he/she spends studying during the current week, and the user can subsequently modify his/her time spending pattern in the next week. Fig. 8(a)–(c) illustrates the UI of the place categorization service.

LifeMap generates location information, as well as other user context information such as activity, connectivity, and environment information. We designed the UI to present the generated user context intuitively, as shown in Fig. 8(d)–(h). The system provides various information and makes a connection to external services such as weather information. A user confirms the number of visit days, the average stay time at a location, connectivity, weather, and a prediction time that will be spent to reach a certain destination based on his/her history. A user can check single-day information, such as sensor usage time, moving distance, and number of steps.

The minor role of these services and UI is to visualize user's life pattern, while the major goal is to understand personal meaning of places and to induce spontaneous user participation for place labeling.

VII. Evaluation

We evaluated our system in three aspects: dead reckoning, place learning, and overall systems. We conducted short-term experiments to evaluate the dead-reckoning scheme with 55 students. The place-learning scheme and the overall system performance are then evaluated with three students for 8 weeks.

A. Dead Reckoning

We conducted experiments to understand the signal characteristics of the accelerometer and electronic compass generated in different positions and walking styles of smartphone users. Participants were given an HTC Hero or HTC G1 to log the raw data of inertial sensors and GPS signals. Each participant walked along a 400-m athletic track of Yonsei University four times and changed the position of the device freely every 100 m. When we asked the participants to change the holding position of the device, we did not suggest which direction to grab the

TABLE II
ACTUAL STEPS OF CATEGORIZED USER CONTEXT

	Messaging	Calling	Hand Holding	Pants Pocket	Hip Pocket	Jacket Pocket	Jacket Inside-Pocket	Inside Bag
Actual Step	15,882	25,528	25,966	22,942	3,764	27,830	3,260	10,932

Participants mainly positioned mobile phones in their favorite positions, such as hand holding, in pants pocket, and in jacket pocket.

phone. We observed the motion of participants and recorded the orientation of the devices.

The position of a smartphone on the human body significantly affects the performance of dead reckoning. We now present the performance of the proposed dead reckoning in various smartphone usage and related positions. The accelerometer signature of human footstep is characterized by rhythmic oscillations. Right and left steps produce slightly different oscillations since a phone is normally placed on one side of the human body. Thus, we define two peaks as one unit to distinguish steps.

We first justify the parameters of peak detection in step counting. We define three parameters, i.e., W, $\Delta t_{\min}$, and $\Delta t_{\max}$, that are required to determine meaningful peaks in oscillations. W is the windows size which is used to determine the mean and standard deviation of an accelerometer signal, $\Delta t_{\min}$ is the minimum periodicity, and $\Delta t_{\max}$ is the maximum periodicity. The role of $\Delta t_{\min}$ is to eliminate the continuous peaks derived from shaking or vibrating motions in a short period of time. We set $\Delta t_{\min}$ as 0.2 s because the average user does not walk five steps in 1 s. $\Delta t_{\max}$ is a time interval between steps to determine periodically repeated oscillations. Based on experimental data, we set $\Delta t_{\max}$ as 1.5 s, although dynamic control of parameters is necessary to reflect the variation of walking frequency. Here, we validate that W is more sensitive to step counting accuracy, which is computed as

$$\text{step counting accuracy} = \frac{\text{\# of measured steps}}{\text{\# of actual steps}}.$$

Table II shows the actual step counts of 55 participants and their favorite positions. Fig. 9 shows the results for $W = 1$ s, 3 s, and 5 s. We found that a short-period window size (i.e., $W = 1$ s) induces inaccurate results due to large variations of standard deviation. The results show that more than 3 s of window size proved robust with less than 10% error. LifeMap employs $W = 5$ s for step counting.

To determine the direction of movement, we defined four factors as deterministic methodologies: max axis, forward axis, categorized context, and direction of movement. Fig. 10 shows the *factor accuracy*, which is computed as

$$\text{factor accuracy} = \frac{\text{\# of corrected measured factors}}{\text{\# of measured steps}}.$$

The accuracy of max axis is meaningless because the movement of a person does not generate a larger acceleration than the acceleration of gravity. The result indicates that determining the forward axis is acceptable except in hip-pocket cases. Direction of movement was measured correctly using a smartphone in stable position such as messaging and calling cases, but the other cases should be revised using heading information from the GPS signal. Heading information is not costly to obtain, since dead reckoning is used to track users when entering into indoor in practical scenarios. The revised method improves accuracy into larger than 90% in every case as illustrated in Fig. 10. Fig. 11 shows the traces of experiments. The messaging case produces the most accurate result, as shown in Fig. 11(a), because the device is stable and screen orientation is obvious to decide direction of movement. In the hand-holding case of Fig. 11(b), the incorrect detection of the direction of movement can be revised using the available GPS signal. Fig. 11(c) shows the inherent shortcoming of the proposed algorithm. The trace was twisted since the device is on a slant to the actual direction. The heading difference directly affects the accuracy of the tracking result. Fig. 12 illustrates the difference between the actual direction of movement and the measured direction. A smartphone in the hand-holding case shows relatively small difference, while approximately 20–40° errors were shown in the pocket case.

We now present the performance of the proposed tracking model in comparison with GPS. We found that a GPS signal is

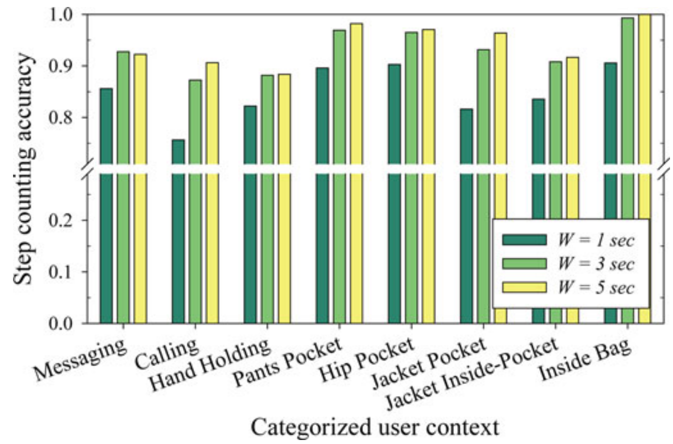

Fig. 9. Result of step counting with different window sizes.

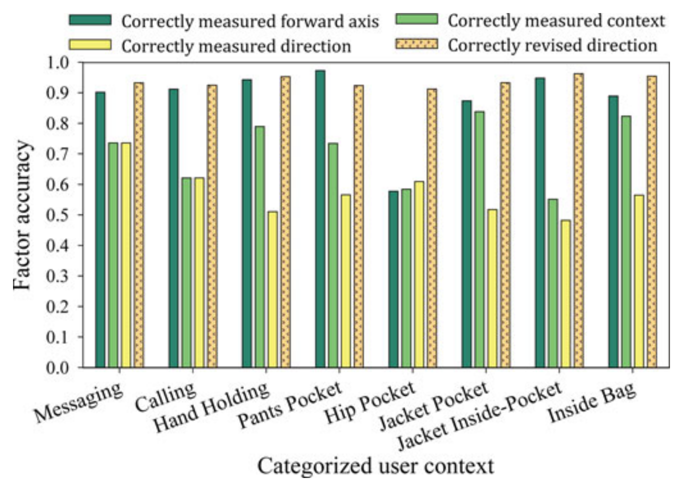

Fig. 10. Corrected measured factors in each user context ($W = 5$ s). The accuracy of the direction of movement without a revise method shows about 60%, which should be revised for accurate tracking.

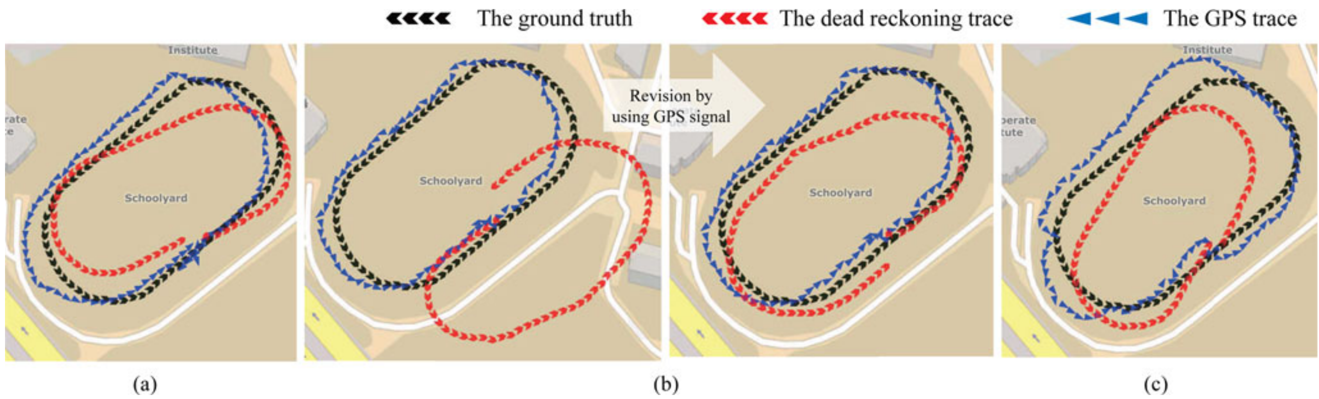

Fig. 11. Trace of smartphone-based dead reckoning. (a) Messaging case. (b) Hand-holding case. The direction of movement is misdetermined. The heading information of a GPS signal is used to revise the direction. (c) Pants-pocket case. The direction of movement is twisted from the actual direction.

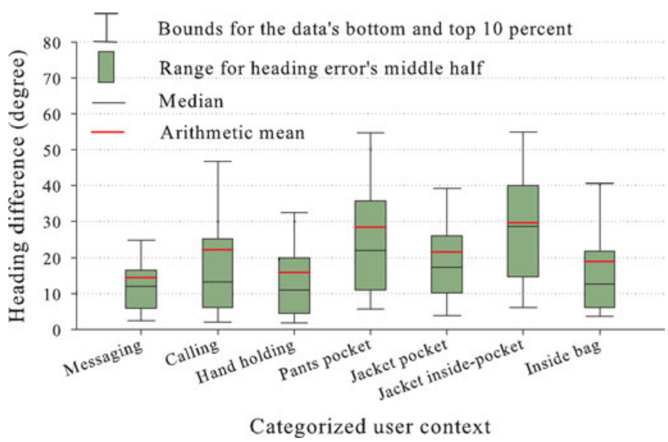

Fig. 12. Difference between actual movement direction and measured direction.

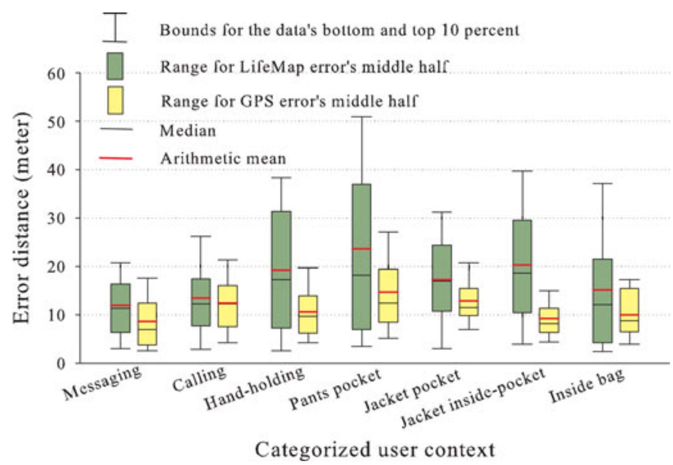

Fig. 13. Error distance of dead reckoning. The performance of dead reckoning is dependent on categorized user context.

insufficient as the ground truth, because of the error shown in Fig. 11. Thus, the ground truth was manually calculated through Google Map. The error distance was evaluated by using the following metrics: GPS error (GE) and average error (AE):

$$\mathrm{GE} = \sum_{i=1}^{K} \left(\frac{e_i^{\mathrm{GPS}}}{K} \right), \text{ where } e_i^{\mathrm{GPS}} = \mathrm{distance}(\mathrm{GPS}_i, R_i)$$

$$\mathrm{AE} = \sum_{i=1}^{N} \left(\frac{e_i^{\mathrm{LM}}}{N} \right), \text{ where } e_i^{\mathrm{LM}} = \mathrm{distance}(\mathrm{LM}_i, R_i)$$

where GPS_i and LM_i represent the user locations reported by GPS and LifeMap, respectively. R_i is the manually calculated coordination for reference. K is the number of coordination from GPS, and N is the number of steps. We synchronized the tracking result to the reference coordination based on the reported time. Fig. 13 shows the performance of dead reckoning. Messaging and calling cases showed an approximately 15-m error, while pocket cases showed about 20–35 m because of the discordance between actual direction and measured direction. The place learning compensates cumulative error in finding POIs. The system refines the location information of POIs based on historical data, as described in Section VII-C.

B. Place Learning

We evaluated the place-learning scheme with a long-term experiment. We collected user traces from three graduate students for 8 weeks. Each student was given an HTC Hero that enables LifeMap as background service to automatically generate POIs in their daily lives. The participant explicitly labeled place name of his or her generated POI. The collected traces contained regular work and home routines with class participation. All participants visited several classrooms located at a walking distance in a campus during class hours. Each participant used different transportation: One drives to school for 20 min; the second participant took a bus to a school for one hour; the other took a subway for about 30 min.

Fig. 14 shows the collected user traces. There are 226 POIs with 2108 visits, 1154 moving nodes, and 1658 APs. Among 226 places, 12 places remained unnamed because these places may not be meaningful to the users. We used 214 named places for evaluation. Logical location comprises automatically constructed information (i.e., Wi-Fi fingerprints) and manually labeled information (i.e., place name). We performed the place-learning scheme without manually labeled information, which is only used as the ground truth.

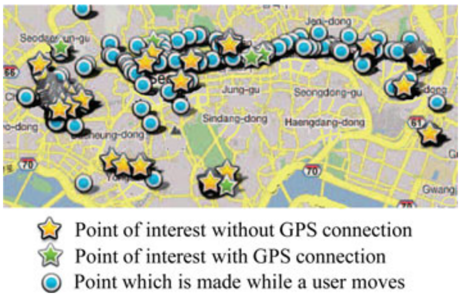

Fig. 14. Collected user traces.

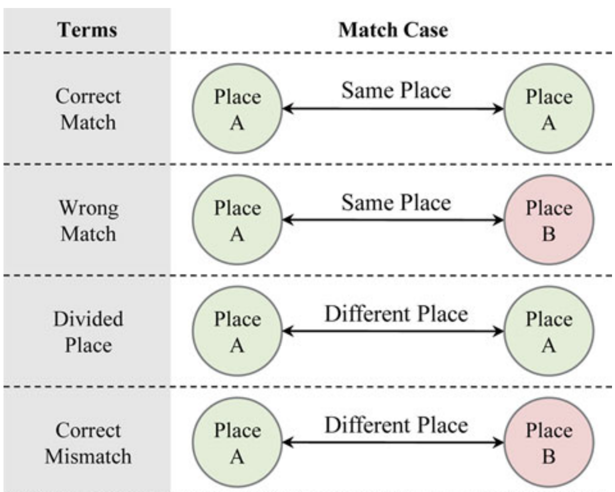

Fig. 15. Defined terms for place recognition.

We evaluated the performance of matching using the brute-force method. We defined terminologies to distinguish various matching results: correct match, wrong match, divided place, and correct mismatch as shown in Fig. 15. We used accuracy and precision as follows:

$$\text{accuracy} = \frac{\#\ \text{correct match} + \#\ \text{correct mismatch}}{\#\ \text{total matched cases}}$$

$$\text{precision} = \frac{\#\ \text{correct match}}{\#\ \text{total matched cases} - \#\ \text{correct mismatch}}.$$

Our RSS model uses three thresholds to ascertain the identification of places: BSSID match rate t_{id}, signal difference threshold t_{rms}, and distribution match rate t_d. Among these thresholds, the BSSID match rate is most insensitive to interference problems because it does not use signal strength. We first evaluated the weighted BSSID matching to confirm the trend of RSS values. High threshold forces decreased aggregations (i.e., correct mach and wrong match), but increased nonaggregations (i.e., divided place and correct mismatch). Precision informs that a low threshold of t_d is more efficient than a high threshold as shown in Table III. Fig. 16 illustrates the trend of RSS values. The result indicates that rms of a signal difference serves as a factor to prevent wrong match when $t_{\text{id}} \leq 0.6$, and distribution comparison is superior in correct match cases. We set the threshold of BSSID match rate to 0.4 and 0.5, and evaluated the performance of the combined method under various thresholds ($t_{\text{rms}} = 12, 14, 16$ and $t_d = 0.5, 0.6, 0.7$) to understand its influence. Fig. 17 shows the results for only four cases. The influence on strict threshold ($t_{\text{rms}} = 12$ and $t_d = 0.7$) is clearly shown in the divided place, but the effect of reducing wrong match cases is relatively small. We decided to use a loose threshold ($t_{\text{id}} = 0.4$, $t_{\text{rms}} = 16$, and $t_d = 0.5$) to maximize reduced wrong match with minimal increased divided place.

To investigate the improvement of LifeMap in place recognition, we compared LifeMap with BeaconPrint [20], which recognizes two places as the same when the overlapping set of beacons contains more than 0.68 to accept within ±1 standard deviation of a standard normal. One of the LifeMap's strengths

TABLE III
NUMBER OF POI RECOGNIZED BY THE BSSID MATCH METHOD

t_{id}	Correct Match	Wrong Match	Divided Place	Correct Mismatch	Accuracy	Precision
0.2	1453	700	75	20485	0.96	0.65
0.3	1420	472	108	20713	0.97	0.71
0.4	1353	345	175	20840	0.97	0.72
0.5	1215	184	313	21001	0.98	0.71
0.6	1021	113	507	21072	0.97	0.62
0.7	740	47	788	21138	0.96	0.47
0.8	482	24	1046	21161	0.95	0.31
0.9	160	4	1368	21181	0.94	0.10

Precision is more effective factor than accuracy since the correct mismatch is significantly larger than other cases.

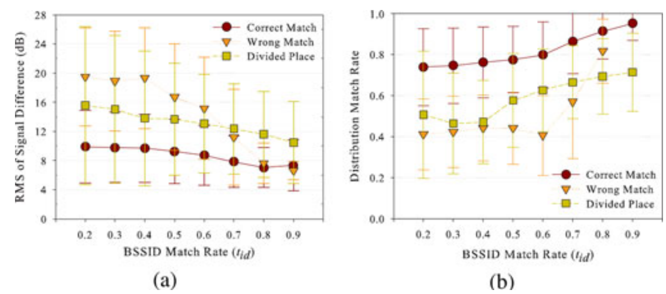

Fig. 16. Trend of (a) rms of signal differences and (b) distribution match rate derived by BSSID matches. An ideal standard is one that includes correct match and excludes wrong match. Strict threshold ($t_{\text{id}} \geq 0.7$) narrows the difference between correct match and wrong match cases.

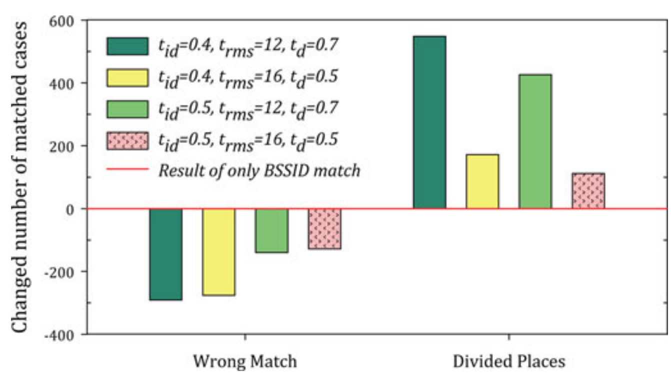

Fig. 17. Comparison between BSSID match results and combined method results.

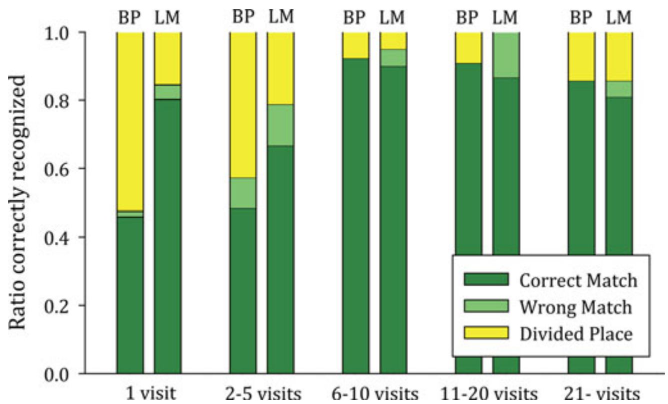

Fig. 18. Percent correctly recognized by the number of visits to the place (BP: BeaconPrint, LM: LifeMap).

is recognizing short visits, as illustrated in Fig. 18. LifeMap is also superior to reduce divided places because of the loose threshold of BSSID match rate. The wrong matched places are noticed in adjacent rooms (e.g., 358 to 353C) or similar locations on different floors (e.g., A542 to A646). Divided places are noticed in places that are visited less frequently (i.e., average 1.3 visits). Nonstatic AP, such as the Wi-Fi gateway in a vehicle and mobile device, is a major factor in the deterioration of the performance of place recognition in a divided place. The proposed scheme is acceptable to recognize visited places, although it has inherent limitations in terms of distinguishing closely packed places when the number of visits is small.

C. Overall Performance

In this section, we evaluated the performance of LifeMap in comparison to GPS and SkyHook [11]. During the experiment period, LifeMap automatically generated a physical location of POI through three methods: GPS, GPS with SkyHook, and LifeMap (i.e., GPS with dead reckoning).

Dead reckoning is a stand-alone method that always decides the location of POIs. Thus, if the dead-reckoning scheme generates the reliability of tracking results and the place-learning method correctly matches identical places, the system can incrementally refine location information with an increasing number of visits. The required condition to this scenario is that error bound should include actual location. The proposed system uses error bound to measure the reliability. Fig. 19 shows an overall scenario of LifeMap. The system generates user's POI with error bound in daily routines, but the result of dead reckoning varies, depending on daily situation, as shown in Fig. 19(a). Logical location is employed to aggregate identical POIs into one POI that has minimum error bound as illustrated in Fig. 19(b) and (c). Consequently, LifeMap provides most accurate physical location based on historical information. On the other hand, SkyHook provides a similar location because of the preconstructed radio map stored in the server. SkyHook generates relatively inaccurate indoor locations near the roadway, since offline training was typically performed by war-driving in the outdoors. In LifeMap, however, the physical location becomes close to the actual location as the number of visits increases.

Fig. 20 shows the performance of GPS, SkyHook, and LifeMap. GPS has poor coverage for POIs, and SkyHook is also inferior to LifeMap in coverage aspects as shown in Fig. 20(a). LifeMap always estimates a user's location, whereas GPS and SkyHook fail to provide location information with the ratio of 79% and 37%, respectively, for visited POIs. Even though GPS and SkyHook are available at POI in indoor environments, they produce locations with poor accuracy and large variations. On the other hand, LifeMap generates accurate location information as the number of visits increases. This means that our system automatically refines location information in daily life. The result indicates that users can obtain location information within about 30 m if the visit number is more than 10 and within about 15 m if it is more than 20. The empirical result shows that the proposed scheme outperforms both GPS and SkyHook in location accuracy, when the visit count is more than 6, as illustrated in Fig. 20(b). Although the accuracy in our scheme is dependent on the number of visits, the proposed scheme has clear advantages of being a decentralized and personalized scheme with which a user gradually constructs his/her own radio map in his/her daily life without offline training.

We report energy consumption of sensing and localization mechanisms through the battery lifetime of the HTC Hero. GPS and SkyHook were experimented to sample the location every 10 s. In addition, we measured the PlaceSense [22] mechanism that scans Wi-Fi APs every 10 s without server communication. This scheme is a representative method of a Wi-Fi-based place-learning system. For LifeMap, we measured lifetime in two states: "stationary" and "moving." We continuously use an accelerometer with the lowest frequency in the "stationary" state. In the "moving" state, we employ an accelerometer with electric compass continuously and turn on the GPS every 3 min. Table IV shows the individual battery lifetime for each sensor/localization scheme. During the experiment, the backlight of screen is always ON with minimum brightness to prevent sleep state. Thus, the measured lifetime is underestimated over the actual lifetime. We estimated that battery lifetime is less than 28 h if "moving" state is about 20%, considering that the average movement ratio of a typical user is under 20% in one day [26]. We believe that smart duty cycling based on user's mobility is a future part of our work to minimize energy consumption.

VIII. Related Work and Discussion

We first compare our system with prior works in three aspects: inertial sensor usage, place learning, and a location provider for everyday tracking. We then discuss a number of issues that would improve the proposed system.

Robertson *et al.* [27] suggested "FootSLAM," which is a simultaneous mapping and localization system for pedestrian navigation in indoor environments with a foot-mounted IMU. FootSLAM provides a positioning accuracy of 1–2 m without map information. However, a foot-mounted approach is impractical for smartphone applications, and the particle filter demands nontrivial computation time, which is not acceptable on a smartphone. In detail, FootSLAM requires 200 MB storage and 30 min computation time on a 2.4-GHz Pentium core to achieve

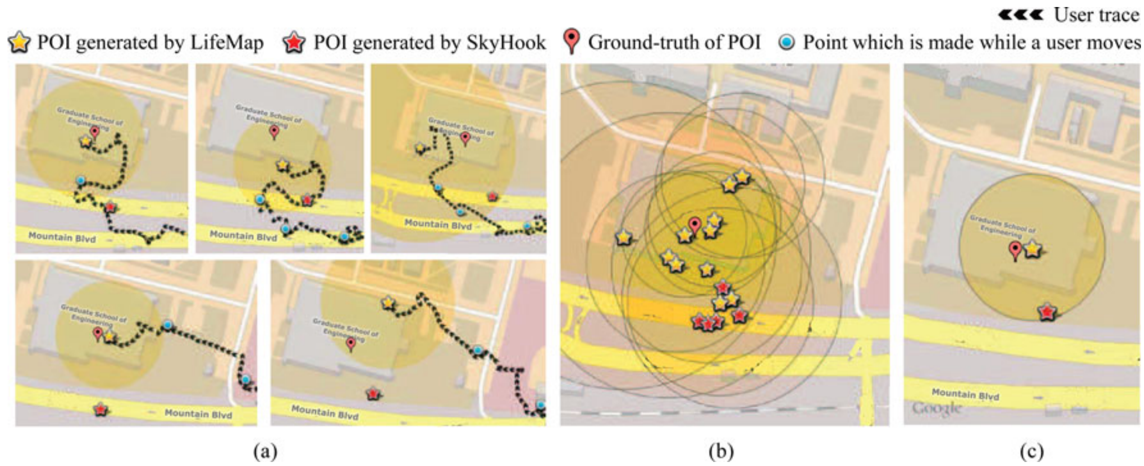

Fig. 19. LifeMap scenario generating accurate POI via an aggregation process. The yellow circle is an estimated error bound of LifeMap. The ground truth was manually calculated through web-based Google map. (a) Daily trace of way to work. The error boundary was estimated based on user context and available heading information of a GPS signal. (b) LifeMap generates POI through smartphone-based dead reckoning. (c) Identical places are aggregated into one place that has minimum error bound.

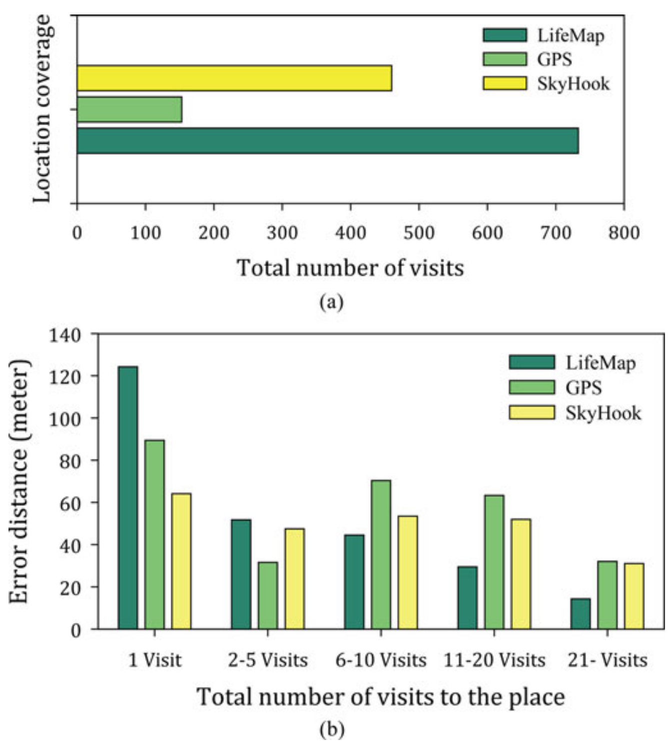

Fig. 20. (a) Location coverage and (b) average error distance of each method from real-life traces. LifeMap refines location information as the number of visits increases.

TABLE IV
BATTERY LIFETIME OF GPS, SKYHOOK, PLACESENSE, AND LIFEMAP

	Idle	GPS	SkyHook	PlaceSense	LifeMap	
					Stationary	Moving
Using sensor	-	GPS	WiFi Scan WiFi Comm.	WiFi Scan	Accel	Accel Comp. GPS
Lifetime (hour)	45.3	8.5	20.1	27.5	32.2	10.2

The backlight of screen is always on to prevent sleep state.

accurate mapping of 700 s of data. Closely related to our work are CompAcc [28] and SparseTrack [29], which employ an electronic compass and accelerometer in mobile phones to track human movements. CompAcc estimates a person's walking pattern and matches it against possible path signature from a map that is not always available in indoor environments. SparseTrack compensates accumulative error of dead reckoning by using additional infrastructures (i.e., ultrasonic sensors). Conversely, both systems do not consider free movement of humans. Users should, therefore, manually adjust their direction of movement to the direction faced by the phone's back when the phone is held statically. In contrast, our system automatically finds movement direction, based on various user contexts, to manage free movement of humans in everyday lives. In addition, LifeMap utilizes Wi-Fi fingerprints to revise drift error of dead reckoning, which is a practical approach in pervasive Wi-Fi environments.

Place learning consists of discovering and recognizing POIs. BeaconPrint [20] discovers POIs by determining stable scans continuously for a defined time period, since the scanned APs are unchanged if a user stays at a certain place. The system uses a number of beacons to recognize POIs. PlaceSense [22] improves the discovering algorithm of BeaconPrint to detect the arrival and departure from a place by exploiting pervasive RF-beacons. The system uses radio beacon's response rates to achieve a robust beacon inference. This technique enables detecting POI, while a user moves. Although an accelerometer-based discovery approach in LifeMap may miss such places, LifeMap is an infrastructure-independent approach and is superior in energy consumption. In addition, we added a distribution comparison to the recognizing method to handle signal interference in indoor environments. SurroundSense [30] focuses on recognizing logical localization in indoor environments. The system utilizes Wi-Fi, accelerometers, microphones, and cameras to generate an ambience fingerprint of places. Extracting features from the microphone and the camera is unique, and the coverage of logical location is expanded to regions that do not have Wi-Fi coverage. LifeMap focuses on managing Wi-Fi signal because the performance of SurroundSense is largely dependent on the properties of the scanned Wi-Fi.

EnTracked [16] is a robust position tracking system in GPS-enabled devices. The system is configurable to realize different tradeoffs between energy consumption and robustness. EnLoc [13] combines GPS, GSM, and Wi-Fi schemes for energy-efficient localization. The system exploits human mobility patterns to predict users' locations to minimize sampling counts. Both EnTracked and EnLoc have a similar goal to our system, i.e., obtaining localization in everyday lives with efficient energy consumption. Each system has different characteristics, such as a robust delay model and mobility pattern-based location prediction. Compared with previous systems, LifeMap is distinct in its ability to provide location information in indoor environments. We also combine logical localization to refine geographical information.

A. Discussion

Although LifeMap is presented as a practical location provider in our work, a number of issues still need to be resolved efficiently.

1) Enhancing Tracking Model: Our tracking model has an inherent problem dealing with heading differences and calibration of walking frequency. Currently, we restricted the direction of movement within the line of the sensor's axis. If the device is on a slant to the actual direction, $45°$of maximum error is expected, although the direction of movement is determined appropriately. Although the system incrementally refines the location information with the aggregation process, the direction error deteriorates the accuracy of a tracking result in each case. Thus, more work is needed to compromise this limitation.

The constant parameters, such as window size, maximum periodicity, and stride length, may generate incorrect results if a user changes walking frequency dynamically. Cho *et al.* proposed "AutoGait," which adaptively estimates a user's stride length using accelerometer and GPS [31]. AutoGait provides an autocalibration method by investigating the relationship between step frequency and stride length. This method can be integrated into our system to eliminate the error derived from the use of constant stride length.

2) Error Bound Estimation: The error bound is important for the validation of generated information. Although the system generates inaccurate location information, the error bound should include actual location and be measured as tightly as possible. LifeMap initializes the error bound from the accuracy of GPS, and increases the error bound by the measured moving distance in a dead-reckoning scheme. Two cases are possible for incorrect estimation on error bound: error in initialization phase and movement without walking. For example, if the GPS estimates error bound as 4 m, even though it is 100 m from an actual location, our system generates error bound that does not include the actual location of the POI. LifeMap may also produce an incorrect location when a user moves with a vehicle into an underground parking-lot of a building, for example. The dead-reckoning scheme estimates the current location near the previous location that was generated in transmit, since a user normally moves vertically from a parking-lot to a POI in a building, for instance. An intelligent hands-off scheme in SkyHook is an alternative solution in this case.

3) Smart Duty Cycling: The current activity-based decision rule uses a naïve approach to turn ON sensors, while a user moves. A sophisticated mechanism is necessary to minimize energy consumption based on collected life patterns. We plan to, therefore, automatically generate a mobility tree based on life patterns. The system may find repeated daily routines, which does not require a big effort, for locating. The system can then focus on providing location when it has a larger degree of uncertainty due to the great variation of life patterns.

IX. Conclusion

In this paper, we presented the design, implementation, and evaluation of LifeMap, which provides an autonomous construction of a personalized POI map for the development of advanced mobile services. The core component of LifeMap is a location management scheme that provides physical and logical location information in everyday lives. Each user constructs his/her own POI map incrementally without a centralized server. We minimized the energy consumption of a device by using a minimum set of sensors based on user activity. User privacy was also considered by designing a decentralized system.

Our belief is that the proposed approach complements current localization technology, taking an important step to expand the domain of mobile services to indoor environments in daily lives. Although LifeMap focuses on the major source of user context (i.e., location), we believe that our approach is a building block toward a sophisticated system that provides various user context, including both location and situation.

References

[1] E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, and A.T. Campbell, "Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, 2008, pp. 337–350.

[2] S. Gaonkar, J. Li, R.R. Choudhury, L. Cox, and A. Schmidt, "Micro-blog: Sharing and querying content through mobile phones and social participation," in *Proc. 6th Int. Conf. Mobile Syst., Appl., Serv.*, 2008, pp. 174–186.

[3] H. Lee, J. S. Choi, and R. Elmasri, "A static evidential network for context reasoning in home-based care," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 6, pp. 1232–1243, Nov. 2010.

[4] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proc. 7th Int. Conf. Mobile Syst., Appl., Serv.*, 2009, pp. 55–68.

[5] H. H. Kim, K. N. Ha, S. Lee, and K. C. Lee, "Resident location-recognition algorithm using a Bayesian classifier in the PIR sensor-based indoor location-aware system," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 2, pp. 240–245, Mar. 2009.

[6] C.-H. Lu, C.-L. Wu, and L.-C. Fu, "A reciprocal and extensible architecture for multiple-target tracking in a smart home," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 1, pp. 120–129, Jan. 2011.

[7] M. Al Masum Shaikh, M. Molla, and K. Hirose, "Automatic life-logging: A novel approach to sense real-world activities by environmental sound cues and common sense," in *Proc. 11th Int. Conf. Comput. Inf. Technol.*, Dec. 2008, pp. 294–299.

[8] H. Shin, Y. Chon, K. Park, and H. Cha, "FindingMiMo: Tracing a missing mobile phone using daily observations," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Serv.*, 2011.

[9] C. Frank, P. Bolliger, F. Mattern, and W. Kellerer, "The sensor internet at work: Locating everyday items using mobile phones," *Pervas. Mobile Comput.*, vol. 4, no. 3, pp. 421–447, 2008.

[10] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Lecture Notes in Computer Science*, vol. 3468, Munich, Germany: Springer, 2005, pp. 116–133.

[11] F. Alizadeh Shabdiz and E. J. Morgan, "System and method for estimating positioning error within a WLAN-based positioning system," U.S. Patent 7 856 234, 2008.

[12] M. N. Gasson, E. Kosta, D. Royer, M. Meints, and K. Warwick, "Normality mining: Privacy implications of behavioral profiles drawn from GPS enabled mobile phones," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 2, pp. 251–261, Mar. 2011.

[13] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, and L. Cox, "EnLoc: Energy-efficient localization for mobile phones," in *Proc. IEEE 28th Conf. Comput. Commun.*, 2009, pp. 2716–2720.

[14] Y. Chon and H. Cha, "LifeMap: Smartphone-based context provider for location-based services," in *Proc. Conf. IEEE Pervas. Comput.*, 2011.

[15] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. Graph. Appl.*, vol. 25, no. 6, pp. 38–46, Nov./Dec. 2005.

[16] M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer, "Entracked: energy-efficient robust position tracking for mobile devices," in *Proc. 7th Int. Conf. Mobile Syst., Appl., Serv.*, 2009, pp. 221–234.

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

[18] M. Mladenov and M. Mock, "A step counter service for Java-enabled devices using a built-in accelerometer," in *Proc. 1st Int. Workshop Context-Aware Middleware Serv.*, 2009, pp. 1–5.

[19] C. Vaughan, B. Davis, and C. Jeremy, *Dynamics of Human Gait*. Champaign, IL: Human Kinetics, 1992.

[20] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes, "Learning and recognizing the places we go," in *Lecture Notes in Computer Science*, Berlin/Heidelberg, Germany: Springer, 2005, vol. 3660, pp. 159–176.

[21] H.-S. Seok, K.-B. Hwang, and B.-T. Zhang, "Feature relevance network-based transfer learning for indoor location estimation," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, doi: 10.1109/TSMCC.2010.2076277.

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

[23] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons," *R. Danish Acad. Sci. Lett.*, vol. 5, no. 4, pp. 1–34, 1948.

[24] A. Haeberlen, E. Flanner, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 70–84.

[25] J. Letchner, D. Fox, and A. LaMarca, "Large-scale localization from wireless signal strength," in *Proc. 20th Nat. Conf. Artif. Intell.*, 2005, vol. 1, pp. 15–20.

[26] Y. Ma, R. Hankins, and D. Racz, "iLoc: A framework for incremental location-state acquisition and prediction based on mobile sensors," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1367–1376.

[27] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Proc. 11th Int. Conf. Ubiquitous Comput.*, 2009, pp. 93–96.

[28] I. Constandache, R. Choudhury, and I. Rhee, "Towards mobile phone localization without war-driving," in *Proc. 29th IEEE Conf. Comput. Commun.*, 2010, pp. 1–9.

[29] Y. Jin, M. Motani, W.-S. Soh, and J. Zhang, "SparseTrack: Enhancing indoor pedestrian tracking with sparse infrastructure support," in *Proc. 29th IEEE Conf. on Computer Communications*, 2010, pp. 1–9.

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

[31] D.-K. Cho, M. Mun, U. Lee, W. Kaiser, and M. Gerla, "AutoGait: A mobile platform that accurately estimates the distance walked," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun.*, 2010, pp. 116–124.

Yohan Chon received the B.S. and M.S. degrees in 2009 and 2011, respectively, from Yonsei University, Seoul, Korea, where he is currently working toward the Ph.D. degree in computer science.

His research interests include mobile sensing systems, human mobility prediction, and context-aware pervasive computing.

Elmurod Talipov received the M.S. degree in computer engineering from Yeungnam University, Seoul, Korea, in 2007. He is currently working toward the Ph.D. degree in computer science from Yonsei University, Seoul, Korea.

His research interests include ad hoc and wireless sensor networks, delay tolerant networks, mobile computing, and wireless communication systems.

Hojung Cha (M'95) received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Korea, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from the University of Manchester, Manchester, U.K., in 1991.

He is currently a Professor of computer science with Yonsei University, Seoul. His research interests include pervasive and mobile systems, embedded operating systems, and sensor network systems.