FindingMiMo: Tracing a Missing Mobile Phone using Daily Observations

Hyojeong Shin, Yohan Chon, Kwanghyo Park and Hojung Cha
Department of Computer Science
Yonsei University, Seoul, Korea
{hjshin, john, khyo and hjcha}@cs.yonsei.ac.kr

ABSTRACT

With the widespread use of smartphones, the loss of a device is critical, both in disrupting daily communications, and in losing valuable property. When a mobile device is missing, localization techniques may assist in finding the device. Current techniques, however, hardly provide a complete solution because of inaccurate position estimation, especially in *indoor* environments. In this paper, we describe a software architecture called FindingMiMo, which tracks and locates a missing mobile device in indoor environments. The system consists of a missing mobile which logs diverse environmental features on a daily basis, and a chaser which traces the trail of the device using the observation log. During daily operation, the mobile device does not perform location estimation; it only observes the ambient features such as radio signals to minimize its operation cost. Instead, the chaser determines where the missing device measured the observations. This research implemented the scheme on Android-based smartphones. Real experiments with carefully designed, missing-and-tracking scenarios show that the participants successfully approached their lost phones within four meters distance, on average.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Signal processing systems; J.0 **[Computer Applications]**: GENERAL

General Terms

Algorithms, Design, Experimentation, Measurement

Keywords

Localization, Ambient Monitoring, Place Learning, Indoor Navigation, Lost and Found

1. INTRODUCTION

As the use of handheld devices, such as smartphones, is rapidly growing, the mobile devices are becoming more valuable and missing devices are not tolerable for most users. When a device is missing, current services can help finding the device. For example, Apple Inc. and Skyhook Inc. provide MobileMe services [1, 2] to find a missing device. MobileMe remotely controls the device and displays users' messages on the screen. It provides the current approximate location by using a GPS signal, cell tower ID, and Wi-Fi fingerprints. However, location estimation becomes inaccurate especially in indoor environments. GPS signals do not reach inside a building and a pre-learned database for radio fingerprints is generally not available for most buildings. This is critical because people spend most of their time indoors [3] and the mobile device is often lost in indoor environments.

Indoor localization is a subject of active research in mobile computing area. The popularity of wireless local area networks (LAN) has opened up a new opportunity for indoor localization. The infrastructure of wireless LAN is well established and mobile devices are usually equipped with wireless modules. Fingerprinting methods using radio signals [4-7] provide accurate positions of the mobile devices even in indoor environments. However, the radio map construction is a costly process and preparing the map for buildings in advance is not trivial. Positioning systems, based on place learning, provide indoor locations [8, 9]. The systems analyze the statuses of users and environments, and recognize locations. Although these approaches recognize semantically meaningful places, we may fail to estimate the physical location of a device where the device is lost in an unknown place. Finding a lost phone poses several challenges. First, the mobility-tracking scheme in the mobile device should be highly efficient because mobile devices have limited battery duration. Second, the device searching process must be accomplished even in indoor environments where floor plans, GPS signals, pre-learned radio maps, and non-standard hardware are not generally available. Third, the chasing process should provide room-level accuracy and let the owner easily approach the vicinity of the missing device. Lastly, additional hardware functionality or infrastructure should not be necessary for the chasing process to occur.

This paper describes a scheme, called FindingMiMo (Finding a Missing Mobile), to trace a missing device. The key components of the system are an ambient logger application for a mobile device and a chaser application. The former is called *missing mobile* and the latter is called *chaser*. Recent smartphones are normally equipped with various components such as GPS, Wi-Fi, and various sensors, which enable the observation of environmental features. However, active operation of indoor pedestrian tracking is not feasible because of the limited energy budget. Thus, the mobile device monitors only the ambient features, such as Wi-Fi signals, and save the log with an adaptive sampling schedule. When the device is lost, the chaser would trace the observations trail in the log from the missing mobile. Although the ambient observations do not contain geometric locations, the data would reveal the clues to find the device. Migrating the

MobiSys'11, June 28–July 1, 2011, Bethesda, Maryland, USA.

functionality of pedestrian tracking from mobile to chaser device minimizes the energy cost for its daily operation. We implemented the scheme on an Android platform. The feasibility of the system was validated by playing games of *hide-and-seek*.

FindingMiMo assumes practical deployment conditions. The service should not be terminated by a user on purpose. A network provider or an OS provider can indeed adopt our solution and the service can be implemented as a system thread. In addition, the lost phone should be alive when the user notices the device is missing to allow the transmission of the ambient log. However, this assumption is not critical because the lost phone is able to upload the log with an unusual-status detection algorithm. A user can also assume that a chaser device will not suffer from energy shortage; it can, in fact, recharge itself anytime.

The contributions of the paper are as follows. First, we propose an efficient scheme to locate a missing device where the device is lost in indoor environments. It formulates and solves the problems of retrieving location information from ambient observations. The scheme monitors only the ambient features surrounding the mobile device to reduce unnecessary usage of system resources. Second, we propose a chaser application that guides the user to the vicinity of the lost device. The chaser analyzes the status of the user, such as Wi-Fi measurements and steps, and provides information for indoor navigation. This approach is quite successful since the participants were able to approach the missing mobile with only the information served by the chaser application. Finally, we have actually deployed the scheme in the real world and proved the feasibility of the proposal. We tested the scheme in the Seoul Coex, the largest underground shopping mall in Korea. We also performed hide-and-seek games that had a user search for the hidden device using the information from the scheme without *a priori* information.

The rest of the paper is organized as follows: Section 2 discusses previous work on indoor localization and related approaches. Section 3 formulates the problem. Section 4 describes the main architecture of FindingMiMo, followed by the system evaluation in Section 5. Section 6 discusses issues regarding the scheme and Section 7 concludes the paper.

2. RELATED WORK

Studies into locating objects [10-12] detect and manage the presence of objects. The approaches attach discoverable radio tags such as RFID (Radio Frequency Identification) and Bluetooth-tag [13] to objects and install RFID reader in the environment or in the mobile phone. Static or mobile supervisors manage the location database and search for the objects' present. Unlike these approaches, we are highly motivated to find a mobile device without infrastructure since a mobile device can implicitly be lost in unsupervised place.

Windows Phone by Microsoft [14] and MobileMe by Apple [1] provide "*find a lost phone*" service. The service displays the approximate location of the remote device on a map, displaying a message or playing a sound on the remote device. The service also sets a pass code lock for wiping out the storage remotely for security reasons, and provides all services in both PC and mobile applications. To map the location of a device they employ the positioning service, e.g. MobileMe uses the XPS of Skyhook [2], which combines GPS, cell-tower-based estimation, and Wi-Fi fingerprints. The positioning service provides highly accurate position estimation, even in remote, but the location service is often not sufficient to find the device under certain conditions (e.g. the device is lost indoors and the user does not have any idea where he/she lost the device). Our solution does not substitute these approaches, but provides the indoor navigation when the approaches might fail.

In general, indoor pedestrian tracking is an active research issue. The systems are categorized to place learning and geometric localization. Place learning algorithms are constructed to recognize semantic places with room-level accuracy using radio beacons (e.g., cell towers, Wi-Fi, and Bluetooth) and surrounding factors (e.g., light, color, texture, and sound patterns). The common idea is that two places are considered an identical place if they have relatively similar place signatures. To generate place signatures from environmental features, each system uses different approaches. SensLoc [8] uses the Tanimoto coefficient of the Wi-Fi vector to detect the entrance and departure time of meaningful places. iLoc [9] uses the cosine similarity of GSM and Wi-Fi vectors to detect movement and meaningful places. Place learning studies such as SoundSense, SurroundSense and Jigsaw [15-17] uses Wi-Fi, accelerometers, microphones, and cameras to generate an ambient fingerprint of places. These approaches may be not suitable to estimate the physical location of the lost device where the device is located in an unknown place.

Geometric localization is categorized as either an infrastructure-based system or a ubiquitous system. The infrastructure-based approaches deploy extra antennas to detect the mobile device. Ubisense provides Ultra Wide-Band (UWB)-based commercial products, which support precise real-time position referencing, long lifetime, and customized applications [18]. UbiSense deploys UWB antennas in a building and receives periodically emitted signals from mobile UWB tags. The active tags periodically emit UWB signals, and stationary antennas receive the pulses and estimate the current location of the tag using both TDOA and distance range. Ubisense is successful in terms of accuracy and energy consumption. For the use of finding a lost phone, however, we do not assume that every building has the infrastructure for indoor localization.

Ubiquitous approaches actively observe the features of the environment and estimate the current locations of mobile devices. The popularity of wireless local area networks (WLAN) has opened up a new opportunity for indoor localization. Massive research on the fingerprinting method has proven the accuracy and robustness of localization using WLAN [4-7]. These approaches are based on correlation between location and radio signal pattern. They gather labeled radio vectors during offline training and search the current position according to currently received unlabeled radio vectors. Gathering and managing the labeled data offline is quite a demanding process. Radar and PlaceLab have tried to reduce these efforts by automating and simplifying the training process [4, 19]. The fingerprinting method has proven that a radio signal pattern is tightly connected to a certain position; this connection is mathematically handled and estimated. However, the off-line radio training process still requires non-trivial cost. In addition, the radio map for the building where a mobile device is lost most likely has not been made before the device is lost, and by the time a user realizes he/she needs one, it is too late.

Apart from the radio technique, tracking a person's walking path using strapdown inertial sensors has been studied [20-22]. The foot-mounted MEMS sensors are installed to detect the steps of a person, using a full six degrees of freedom (6DOF) inertial

navigation. The strapdown inertial platform is comprised of triads of accelerometers and gyroscopes. These localizations provide highly accurate location of the mobile device. Greenfield [23] introduced an activity-based navigation, which guides a user to destination with human activities derived from sensor data. The approach proved the feasibility of activity-based navigation by implementing a car finding application. While Greenfield captures the series of action of human and revives the actions, we present a lightweight ambient logging and a method to trace the observations.

In summary, the positioning system for a lost device should consider the realistic constraints of energy consumption, accuracy, and stand-alone operation. In this paper, the FindingMiMo solution provides lightweight daily operation of the mobile device and highly accurate indoor tracking with no additional device.

3. PROBLEM FORMULATION

Finding a missing device is to retrieve physical locations from the ambient observation log. Figure 1 (a) shows the hidden Markov model describing this problem. In the proposed scheme, a mobile device observes ambient features and logs them during its daily operation, y_k □Y. When a mobile device measures feature y_k in location x_k at time k, where k□{…,τ}, the location x_k, including the final location of the mobile device, $x_{k=\tau}$, is unknown, which is a hidden state. The log contains position information, such as GPS readings, $y_{k'}$; the state $x_{k'}$ is known. Eventually, the searching process of the chaser is to retrieve the final location, $x_{k=\tau}$, from the known place, $x_{k'}$.

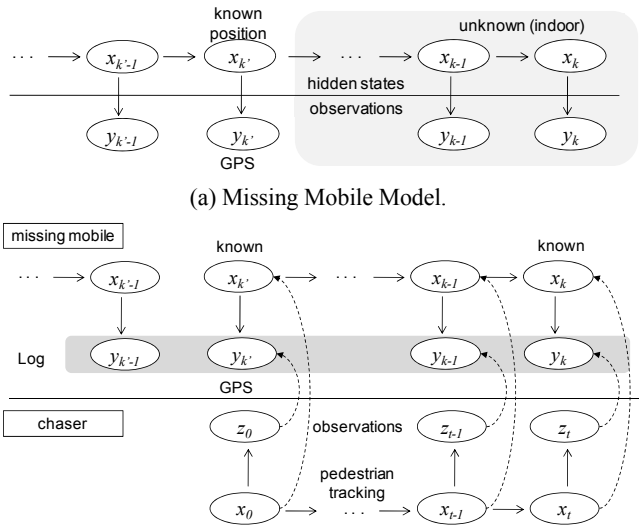
Figure 1(b) illustrates how the chaser device determines the unknown location. At the beginning, a chaser device visits the known place $x_{k'}$ and tests its first observation z_0 with the observation $y_{k'}$. The current observation may be similar to the observation $y_{k'}$. Then the chaser starts the chasing process. The chaser may visit one of nearby places, x_t, and test the current observation z_t. The chaser compares the observation z_t to the set of logs Y and finds the best-matched observations,

$$\arg\max_{y_k} sim(z_t, y_k), y_k \in \mathrm{Y}, 0 \le sim() \le 1,$$

where a function *sim*() is the Tanimoto coefficient function to evaluate the similarity between two input vectors [24]. The function is defined as:

$$T(y_1, y_2) = \frac{y_1 \cdot y_2}{\|y_1\|^2 + \|y_2\|^2 - y_1 \cdot y_2}$$

Radio observations are transformed into vector space, whose attribute vectors are the signal strength vectors of observed access points. The current position x_t is physically close to x_k in proportion to the similarity. Based on the current position x_t, the similarity and the time index k, the chaser decides the next place x_{t+1}. The chaser is able to visit every possible place with the intention of chasing x_k with an increased k index. With repeated visiting of x_t and testing the observation z_t, the chaser finally approaches the final place of the missing mobile $x_{k=\tau}$. Figure 1 (b) illustrates that the chaser's states are linked to the missing mobile's hidden states during the chasing process. The chaser finds the series of place x_k where the observation y_k is taken, and the dotted line represents this matching. This empirical matching process is called *visit-and-test* in this paper. Finally, we obtain a series of x_k, which is the movement path of the missing mobile. The chaser



(a) Missing Mobile Model.

(b) Chaser Model.

Figure 1. A missing mobile model and a chaser model. The missing mobile logs the ambient feature y_k at unknown position x_k. The chaser visits a position x_t and determines the link between y_k and x_k

empirically searches the moving path of the missing mobile and eventually approaches the device.

The Place-matching Problem

Input*: initial position $x_{k'}$, series of observations y_k (from a missing mobile) and live observations z_t (from a chaser device)*

Output*: the moving path of the missing mobile x_k including the final place of the device and the moving path of the chaser x_t with the derived similarity index s_t*

The goal is to match the series of place x_t to the series of observation y_k. From the known place $x_{k'}=x_o$, the chaser should empirically visit and check the next all possible places x_t at each state transition which is a non-trivial task. In real environments, however, the circumstances are such that the structural condition limits the chaser's movement in indoor environments. While a chaser visits a path among some corridor branches, the chaser may encounter a well-matched series of observations to the log and drop other corridors. Even a wide-open space such as a lobby or lounge has a limited number of paths, such as gates, elevators, and rooms. The user can quickly make his/her choice based on his/her memory. The chaser application also analyzes the ambient observations and provides useful information to guide the user to the final place.

4. FindingMiMo

The scenario is as follows: A mobile device collects daily ambient observations (e.g. the Wi-Fi channel condition) and logs them. When the device is lost, the user prepares chaser device and remotely receives the log from the lost phone. Then the chaser traces the device by determining the places where the log is taken. Figure 2 illustrates this scenario. During a daily usage of mobile phone, the device observes the ambient features, Wi-Fi signal conditions. After the device is lost, the chaser tries to retrieve the path which the lost device has traveled. The chasing process is

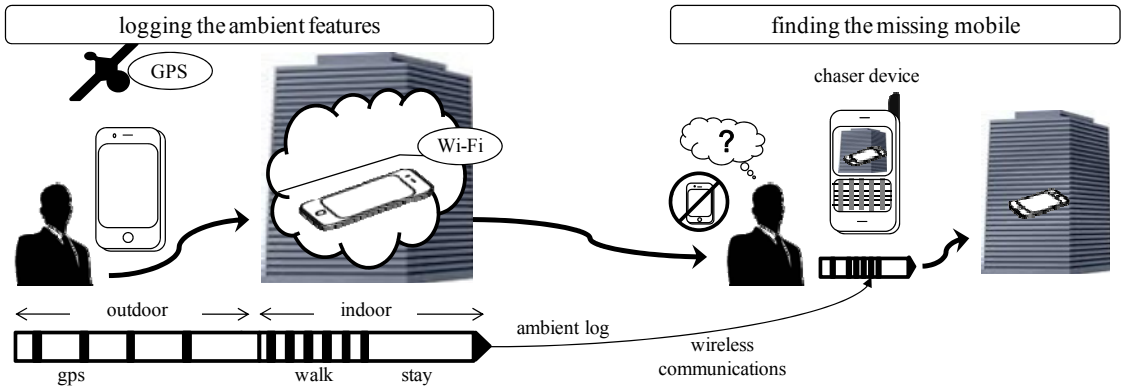


Figure 2. A user carries his/her mobile device which logs daily ambient observations. One day, the device is lost somewhere indoors and the user cannot imagine the location. The user borrows a chaser device and downloads the current ambient log from the lost phone. Then, the user traces the clues in the log. Our solution guides the user to approach the lost phone even where the log does not carry the geometric information of the device.

similar process to the "warm/cold game" [1]. Although the final location of the lost device is unknown, the radio signals observed on the path are known. Comparing the Wi-Fi signals to the observations in the log, the chaser application displays if the user is walking along the right path. When the user stands or walks along the right path, the phone may observe the same observation in the log. When the user goes off the path, the user encounters unfamiliar radio signals and the chaser application notifies the user of the veer event.

The system architecture of a missing mobile and a chaser is illustrated in Figure 3. The missing mobile consists of the LifeMap middleware [25, 26] and a missing mobile application. LifeMap is a place-learning tool, which organizes the places that the user frequently visits and analyzes the movement pattern of the user. The missing mobile application saves the log based on the status monitoring from the middleware. The log contains outdoor positions, POIs, Wi-Fi observations and other information. The log is transmitted via wireless networks such as cell networks to the chaser device. The chaser consists of SmartSLAM [27] and the chaser application. SmartSLAM is a run-time indoor pedestrian tracking system. Unlike LifeMap, SmartSLAM monitors the physical movement of the user. It analyzes the moving path of the user and constructs the indoor floor plan. The estimated indoor position is used by the chaser application for visual representation. The chaser application analyzes the Wi-Fi observation and provides navigation information. The chaser application guides the user to the place of the missing mobile by displaying the useful information, and the user eventually approaches the missing mobile.

The following sections present the information required for device searching, and describe each component in detail.

[1] Warm/Cold Game: This game lets a kid search for a hidden item (or prize). The hider sends the kid out of the room and stashes the item in a safe. The hider helps the kid by yelling out "warm" when the kid is headed in the right direction and "cold" when he is not.

4.1 Chasing Information

The chaser application provides the clue information for tracking the missing mobile. The clue information is as follows.

Clue Report

I. Circumstantial evidence (*clues*)
II. Indoor pedestrian tracking (*navigation*)
III. Observation analysis (*log similarity*)
IV. Virtual distance to the trace (*trace similarity*)
V. Virtual distance to the destination (*target similarity*)
VI. Chasing progress (*progress*)

The FindingMiMo platform provides the clue report that help the user approaches the missing device. The circumstantial evidence (*clues*) is semantic information when the device is lost. The information includes an approximate missing time, the previous place the user visited if it is known, the label of lastly visited place, approximate location from positioning services and so on. This information may remind the user where he/she visited, and the possible places where the device might be.

Indoors, the chaser device performs indoor pedestrian tracking with SmartSLAM (*navigation*). SmartSLAM detects human's walking trajectory and builds up the indoor floor plan. The pedestrian tracking provides the current indoor location, $\widehat{x_t}$. The pedestrian tracking also helps the user to understand the current chasing status.

Wi-Fi is employed in this paper mainly for ambient observation. The observation report helps a user decide the place to search. First, the chaser analyzes the current observation z_t, and compares the similarities between current observations and logged observations (*log similarity*). Based on the result, the chaser determines whether the chaser is standing on the path that the missing mobile passed through. The log similarity matrix is defined using the Tanimoto coefficient:

$$\{s_t \mid s_t = sim(z_t, y_k)\}, y_k \in \mathrm{Y}, 0 \le s_t \le 1.$$

The Tanimoto coefficient which measures the similarity of two vectors provides the comparison.

Within the log similarity matrix, we obtain the best-matched observation and its similarity. The similarity index to the best-

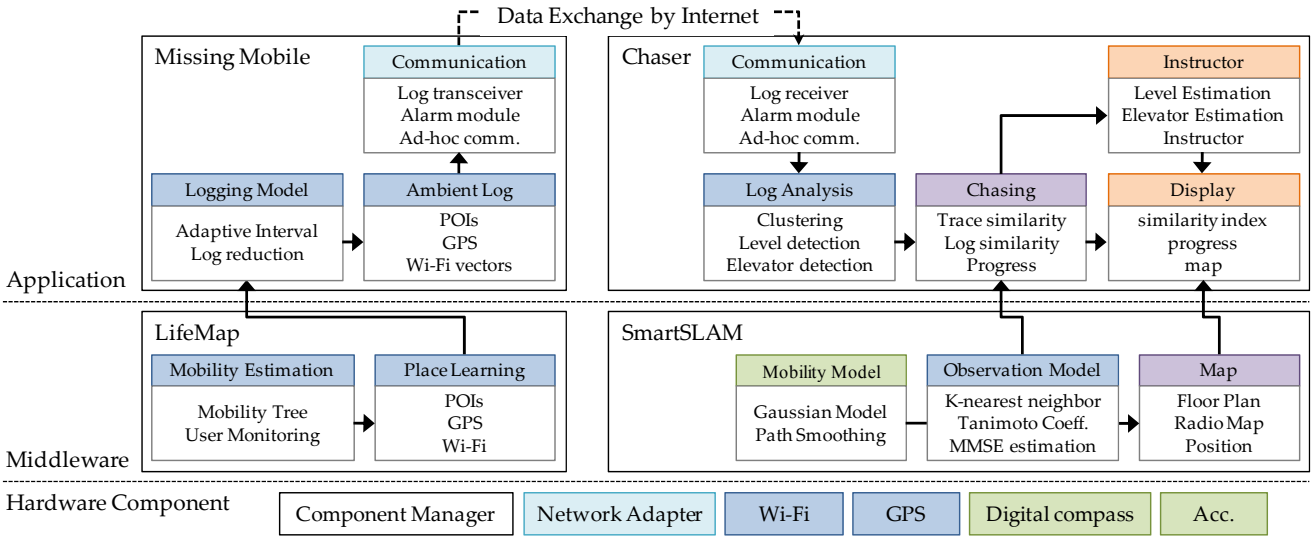


Figure 3. Architectures of a missing mobile and a chaser device. LifeMap provides the ambient observations and known places. SmartSLAM supports run-time indoor pedestrian tracking for a chaser application.

matched observation is semantically the virtual distance to the trail of the missing mobile, called *trace similarity*. The trace similarity indicates whether the user is following the route that the missing mobile passed or not. The trace similarity is represented as

$$max(sim(z_t, y_k)), y_k \in \mathrm{Y}.$$

Additionally, the similarity index to the last observations in the log represents the virtual distance to the destination, called *target similarity*. The target similarity is represented as

$$sim(z_t, y_\tau), y_\tau \in \mathrm{Y}.$$

The target similarity is useful when the user is near the missing mobile. Semantically, if the target similarity is one, the chaser finds the missing mobile.

The *chasing progress* is the time index of the best-matched log. The progress is

$$\arg\max_k sim(z_t, y_k), y_k \in \mathrm{Y}, \text{given } z_t.$$

When the chasing progress becomes the size of the log, the chaser approaches the vicinity of the missing mobile. Within the vicinity, the chaser can access the missing mobile and trigger the alarm-sound. The chaser then searches the surrounding area and finds the device.

4.2 LifeMap: Context-aware Mobile Sensing

LifeMap is a smartphone-based middleware for tracing human mobility: time-resolved place and path. The system continuously monitors a user's physical location (i.e., latitude and longitude) with ambient fingerprint (i.e., radio signal fingerprint). LifeMap automatically constructs a context map comprised of nodes and edges. The node represents a point of interest (POI), and the edge is a path to encompass minimal context on the user movement. With LifeMap, a user always acknowledges his/her current location, and the mobile phone can notify/share the change of places to internal and third-party applications.

LifeMap makes use of various sensors including GPS, GSM, and Wi-Fi, which are commonly found in the latest smartphones. To reduce energy consumption, LifeMap uses activity-based sensor selection, which is widely used technique to activate different sensors in moving and stationary states [8, 18, 28]. The basic idea is that the system uses a power-intensive sensor, such as a GPS, if the user is moving, while the system minimizes sensor usage time in stationary states. To detect a user's movement, LifeMap utilizes the signal fingerprint of surrounding Wi-Fi access points (APs). The movement detector is defined using the Tanimoto Coefficient between previous scanned fingerprints and the current one [8, 24]:

$$\mathcal{M} = \begin{cases} \text{move}, & \text{if } \frac{\vec{f}_1 \cdot \vec{f}_2}{\|\vec{f}_1\|^2 + \|\vec{f}_2\|^2 - \vec{f}_1 \cdot \vec{f}_2} \le \varphi \\ \text{stationary}, & \text{else} \end{cases},$$

where $\vec{f}_1$ is the previous scanned Wi-Fi vector, $\vec{f}_2$ is the current scanned one, and φ is the similarity threshold. The significant difference of similarity indicates the change of place.

In the moving state, the system periodically collects GPS signals and Wi-Fi vectors. When a stationary state is continuously maintained, the place is considered to be a meaningful place and the system generates place signatures including its Wi-Fi vector. A user confirms his visited places and labels them as he/she wants. Any place labeled by a user is considered as a *known place*.

LifeMap monitors the user's mobility and determines the meaningful places. In this paper, the role of LifeMap is to provide a user's movements and known places for finding mobile devices.

4.3 MissingMobile: Daily Ambient Logger

The logger is collecting environmental features to provide clues for the user's searching process. A key issue is to optimize data collection to find a missing mobile device.

The essential information to find a device is an ambient radio log on the path from the last known place (i.e., labeled place or known physical location) to an unknown location. The application collects

Wi-Fi vectors on transition paths by using the information from LifeMap: detection of a user's movement and labeled places. When LifeMap detects the movement state and it fails to obtain location information from GPS module, the logger application starts to collect Wi-Fi vectors with fixed time intervals (i.e., 5 seconds). Otherwise, the application is set to sleep state. To minimize the amount of stored information, the system limits the collection period to one day, and resets the path information when a user re-visits the previously known place or when the GPS provides accurate location information. Consequently, a missing mobile stores Wi-Fi vectors of paths from the known places to unknown locations, which include a considerable amount of uncertainty.

The role of the logger application is to provide clues for finding the missing mobile if the device is lost. The logger first transmits the stored information to the chasing device. When the chasing device is closely approaching the missing mobile, the logger causes the missing mobile to make a loud sound to indicate its final position by manual request from the chaser device. Automatic notification can be achieved by implementing ad-hoc communication.

4.4 SmartSLAM: Indoor Pedestrian Tracking

SmartSLAM is an indoor user-tracking solution using smartphones. SmartSLAM traces pedestrian movement in indoor environments and simultaneously constructs an indoor floor plan. Indoor pedestrian tracking, where the GPS signal is not visible, is a challenge. A pre-learned radio map for a fingerprinting-based method is not realistic for anonymous buildings. Dead reckoning using inertial sensors has an accumulative error drift. SmartSLAM overcomes these limitations by employing the simultaneous localization and mapping (SLAM) technique [29] and sensors equipped in the smartphone. SmartSLAM proposes a mobile model to encode human steps and an ambient observation model to recognize the place. The observation model memorizes the place's features and adjusts the error from the mobile model when the user visits a known place.

An accelerometer and a magnetometer in smartphone device are employed to estimate the user's walking path. The system analyzes the acceleration signal pattern at its peak point and counts steps when the same patterns are periodically found. The heading direction is observed by the digital compass. According to unconstrained placement of the smartphone, the absolute orientation is unknown. Therefore, only the change of heading is used. In fact, absolute orientation is not essential to provide geometric information in an indoor environment because people can understand a rotation of the map to explore a building.

SmartSLAM uses the Wi-Fi observation model to identify locations. The observation model scans Wi-Fi channels and stores the signal strength from each AP. In an urban area, Wi-Fi infrastructure is well established and most off-the-shelf smartphones are equipped with a Wi-Fi module. The set of signal strength readings, measured at a certain position, called a Wi-Fi vector, represents a landmark at that position. While the user revisits the identified place, the estimated walking path from the mobile model is adjusted. The sum of the corrected paths represents the floor plan of the building.

SmartSLAM provides the indoor floor plan as a map and the location of the user. A user's information can be placed on the map. With a control interfaces, one can store the map, read the current

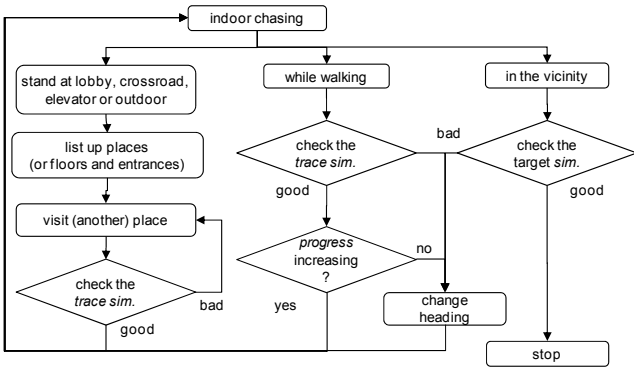


Figure 4. Chasing policy of the user

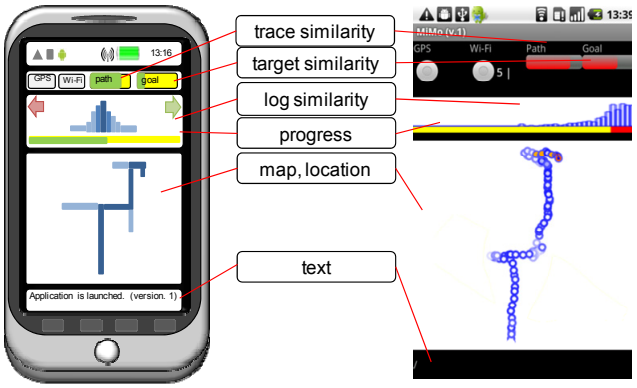


Figure 5. User interface of the chaser application

location of the user, store objects at the current position and retrieve objects from the position. These interfaces are used by the chaser device.

4.5 Chaser: Device Tracking

After a login process, the chaser device is authorized to access the lost phone. The authorized device accesses the missing mobile and receives the log data. If the log data is not aware of the specific location of the missing mobile, the user follows the searching process.

First, the user should select a target building. Because the log from the lost phone contains GPS locations, an approximate location can be obtained. A user visits entrances of close buildings and tests Wi-Fi observations. If the device observes similar Wi-Fi signals to those of the log, the user begins indoor tracking.

Inside the building, SmartSLAM provides indoor location of the mobile user (navigation). While the user is walking, the chaser application measures the Wi-Fi conditions and compares them with the observations in the log. The result is provided to the user (log similarity, trace similarity, target similarity and progress).

With this information, the searching process proceeds as in Figure 4. The user can understand the surrounding environment, such as a lobby, corridor and elevator. At any position, the user can list the places that he/she can access. Then, the user tries to visit one of the places and tests the ambient observation, denoted visit-and-test. If the trace similarity is good, it means the user visited the right place. If the log similarity becomes bad, the user should return to the last

place that shows the high trace similarity. This visit-and-test process is the method used to decide the chasing route when the user encounters an ambiguous situation. The information is provided via GUI, illustrated in Figure 5.

At any time while the chaser walks, a decrease of the trace similarity means that the chaser has visited the wrong place. The chaser should return. This information is useful when the chaser meets crossroads. When the user tests some branches, the chaser application denotes which branch is the right path. If the trace similarity is good, the chaser stands on the trace that the missing mobile passed through. If the progress, which is an index of best-matched observation in the log, goes backward and the trace similarity stays high, it means that the chaser stands on the right trace but the user is going backwards. Therefore, the chaser should turn back. If the trace similarity is good and the progress increases, the chaser comes close to the final place of the missing mobile.

In the vicinity of the lost phone, the user is able to physically search the surrounding places. If the device is not revealed, the chaser device pushes a message via cell-network to trigger sound-playing. This feature is already available in the current commercial services [1, 14].

5. EVALUATION

We implemented the proposed middlewares and applications: LifeMap, SmartSLAM, Missing Mobile, and Chaser. LifeMap, along with the missing mobile, is implemented as background service. SmartSLAM and chaser are implemented as activities, with variation of application for the Android platform. All systems are implemented and tested in Google Android API 2.1.

First, we validated the feasibility of daily logging in terms of energy efficiency and space complexity (Sec. 5.1). For a proof of concept, we performed a simple experiment that searches a hidden device (Sec. 5.2). We proved that the information provided by the chaser application is sufficient to search for the device. The system quickly reports the situation of the user and helps him/her to make correct decisions. Although the searching process succeeded in finding the device, we found that distinguishing different floors of a building is challenging problem. To analyze this effect, we conducted further analysis on vertical localization (Sec. 5.3). During searching, the knowledge and proficiency of a searcher highly affects the results. To evaluate our system in the absence of searcher's knowledge or proficiency, we designed a hide-and-seek game (Sec. 5.4). One participant hides a device and multiple chasers separately try to search for the hidden device. Most of the chasers succeeded in finding the device with a little individual difference. Last, to evaluate the feasibility of the FindingMiMo platform in real environment, we conducted a lost-and-found experiment in a large shopping mall (Sec. 5.5). For all experiments, we employed different phone models for the missing device and chaser device.

5.1 Missing Mobile

We evaluated the performance of Wi-Fi logging in two aspects: energy consumption and space complexity. The analysis on some parameters for Wi-Fi logging is omitted since our system uses a scan window scheme, which is a widely used technique to tolerate noisy radio signals [8, 9]. The sampling interval for GPS sensing is 2 minutes; we activates it for 30 seconds for single positioning. The interval of Wi-Fi scanning is adaptively determined, based on a user's movement. In the move state, the system stores Wi-Fi vectors every 5 seconds. In the stationary state, we activates Wi-Fi

(a) History of places where the user has visited

(b) One path trace near building in campus

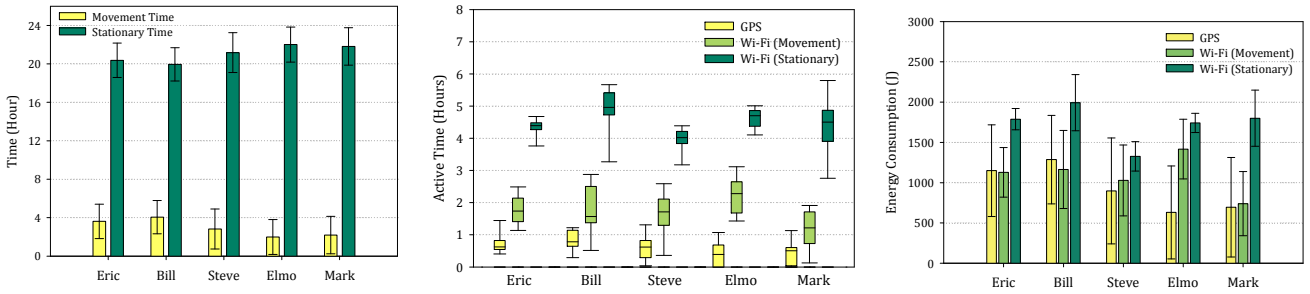
(c) Data trace during collection period

Figure 6. Data traced during a collection period. (b) Circles are locations in transit, and the star is the last known location provided by GPS in a path.

module for 30 seconds to scan the surrounding APs every 10 seconds. The sampling interval is set at 2 minutes. The similarity threshold of the Wi-Fi vector is set at 0.7 [8].

To evaluate the energy consumption of our system in daily life, user traces were collected from five students for two weeks using HTC Hero, HTC Desire, Samsung Galaxy A and Samsung Galaxy S smartphones. The Wi-Fi logger was running as a background service to trace Wi-Fi trails and sensor usage time automatically. Since LifeMap manages user's movement and places, the service provides the history of places where the user has visited. This data includes the lost time of the device and prior place before the device is lost. Figure 6(a) is a screen shot of the history journaling from LifeMap. Figure 6(b) and (c) show data collected during the collection period. The results indicate that the last location provided by GPS in transit can be used as a landmark to infer a building where the mobile device is, as illustrated in Figure 6(b).

We first measured the movement time and stationary time of each user to estimate energy consumption in one day as described in Figure 7(a). Previous works found that typical users spend approximately 13% to 20% of their time outdoors each day [3, 9], and the collected data set shows similar results. We measured the average active time and the energy consumption for each sensor in daily life with LifeMap and Missing Mobile. The proposed system activates GPS, Wi-Fi with movement state, and Wi-Fi with stationary state for 0.6±0.2 hours, 1.7±0.4 hours, and 4.2±0.6 hours, respectively, as illustrated in Figure 7(b). If a user travels a long distance, the active times of GPS and Wi-Fi increase, and vice versa. Based on the collected data traces and phone energy profiles



(a) Average movement time and stationary time in a day

(b) Sensor active time distribution per day

(c) Average energy consumption in a day

Figure 7. Sensor usage pattern of each participant. (a) Users spend approximately 8% to 17% of a day to move. (b) Sensor active time with movement state is widely distributed since the individual behavior has diverse pattern. (c) Average energy consumption in a day, including CPU consumption, is 3.7kJ that may reduce the battery's lifetime by 14% in average.

during sensing [30], we estimate the average energy consumption of a Wi-Fi logger. The proposed system consumed 3.7±0.5 kJ in a day, as shown in Figure 7(c). The results indicate that the Wi-Fi logger may reduce the battery's lifetime by 9% at least, 14% in average, and 27% at most, since 60% of people use their smartphone for 14 to 41 hours with a single battery charge [31]. In other words, a user can use the smartphone with the proposed system for 12.4 to 31.8 hours depending on individual usage patterns. We believe that such consumption for continuous system is a tolerable cost for insurance to find a lost device.

The size of data traces depends on a user's behavior pattern and AP deployments. The number of scanned APs for a day was 367 at least and 1,573 at most, and the number of rows in the database was 50,160 at least and 248,700 at most. In other words, the average amount of data for a day is 4.5MB to 22.3MB, which is a reasonable cost considering that the data storage of recent mobile devices is at least 4GB. Since the goal of the Wi-Fi logger is to leave traces to find the lost device, the system can only store Wi-Fi traces in transition between known and unknown places. Thus, the amount of data traces can be reduced because the system may eliminate Wi-Fi traces on previous paths when a user re-visits known places or GPS provides the accurate location information. Figure 8 illustrates the space complexity of Wi-Fi traces of a few days. The trace indicates that, when a user moves, the space complexity is gradually increased, while the system removes previous data if a user re-visits the known places. Indeed, the typical storage usage is around 5MB, although the space complexity is dependent on individual behavior patterns.

The size of the log can be reduced by removing the redundant information in the log. If a missing mobile moves via places in {A, B, C, B, and D}, the sub-sequence {B, C, and B} is redundant information. The path is simplified into {A, B, and D}, since two observations measured in the place B are similar to each other. It is clear that observations between the two Bs can be removed.

5.2 Chasing a hidden device

For a proof of concept, we performed a simple searching experiment. A user carried a smartphone from the main gate of the engineering building to a random location in the building. The user visited the building with the other device and traced the sample log, called a second visit. The Wi-Fi sample interval was 1 second. The moving scenario was: entering the building, passing a lobby, going

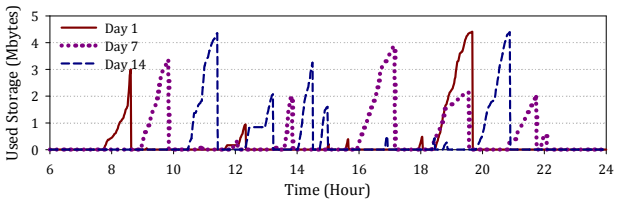
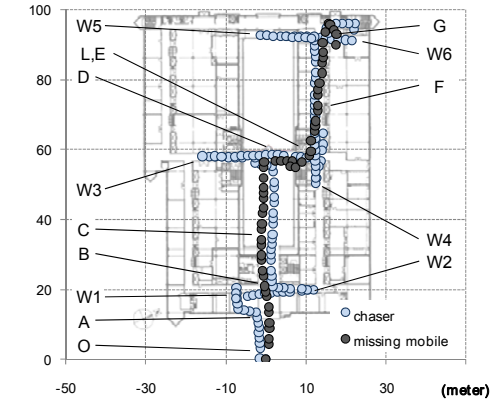


Figure 8. Space complexity for storing Wi-Fi vectors in a day. The used storage is reduced when a user re-visits known places.

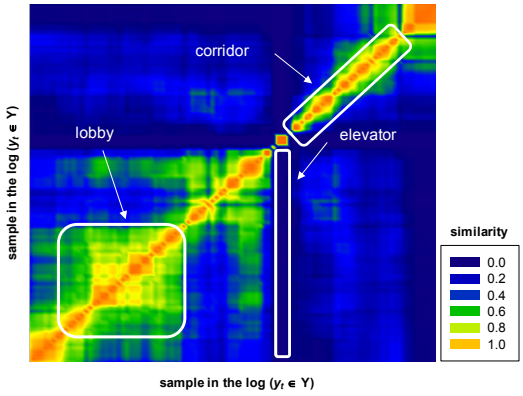
up by elevator, walking through a corridor, entering a room, and staying. The route of the missing mobile is illustrated in Figure 9(a).

The user carried the same device with the Wi-Fi observation log. The log contained the environmental information such as logging time, last GPS position and further derived clues. Figure 9(b) shows the similarity matrix between the observations in the log itself. Light yellow parts mean the observations are similar to each other and dark blue parts mean they are not. Since the observations in the log are self-compared, a slant line is found in the middle of the figure. The figure exhibits some distinctive features of the observations. The slant line is clearly distinguished and the other space are covered with dark blue. This means that the observations in the log are distinguishable from each other and the mobile device has been continuously moving. At the middle of the figure, the dark area represents a closed room, for example, elevator. The observation in this area is clearly distinguished from the nearby observations. At the right-top corner, a square in orange is found and it means that the mobile device stays for a while. The average signal strength and the number of signals also decrease as illustrated in Figure 9(c). Open spaces and corridors also can be determined; at the lobby, the observations are similar to each other and within the corridor, the observations are distinguishable.

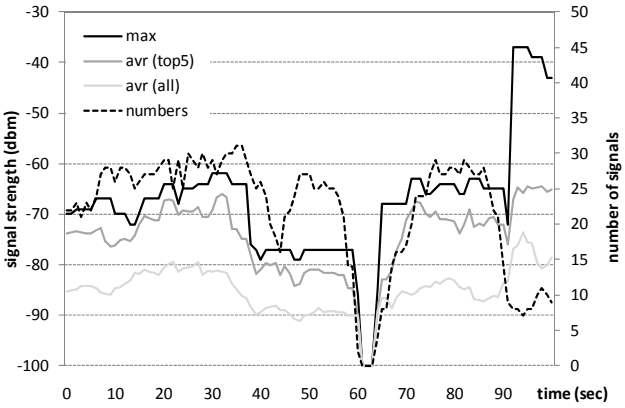
The chaser enters the building and samples the ambient features. At each sampling, the chaser analyzes the observation report; log similarity, trace similarity, and target similarity. Figure 10(a) represents the similarity matrix between the observations in the log and the observations during the chasing process. The matrix is also the series of log similarities. X-axis represents time elapse and Y-axis represents the samples in the log. Top on Y-axis is destination.



(a) Movement path



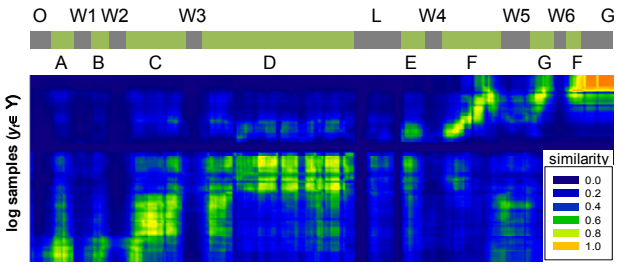
(b) Log analysis



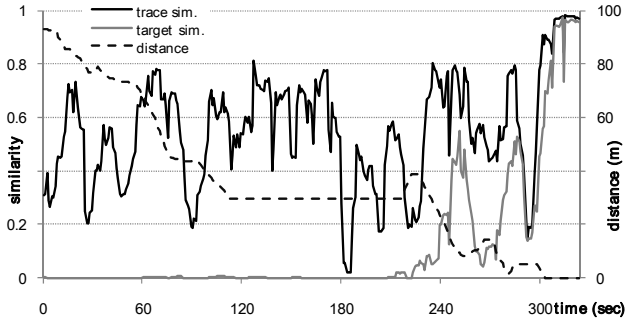
(c) Signal pattern of the log

Figure 9. Experiment setting: (a) movement path of the missing mobile and chasing path of the chaser. The route of the missing mobile is measured by inertial sensors. (b) the features of the observations in the log. (c) signal strength and the number of signals

In the figure 10(b), the trace similarity dynamically changes according to the location of the chaser. In this example, the user enters the building from outdoors {O}. The pedestrian trace is shown in Figure 9(a). At the beginning of the graph, A, the trace similarity in (b) increases and bright-green indexes are found at the lower part of the graph that means the current observations are similar with the former observations in the log. The user visits the



(a) Set of trace similarity



(b) Trace similarity and target similarity

Figure 10. The similarity information during the searching. (a) Color index presents the similarity between radio signal vector observed by the missing mobile and that by the chaser.

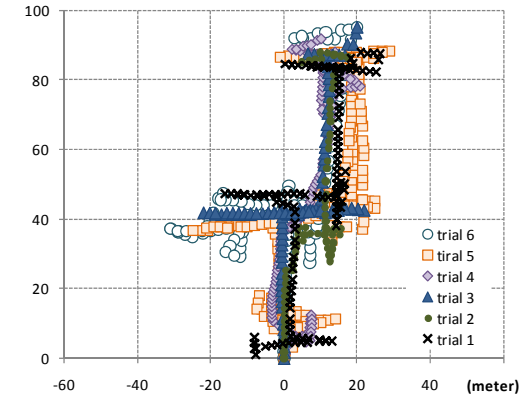
left corner of the building, W1, and the trace similarity decrease. It means that the user visits the wrong place. Then, the user becomes aware of standing on the wrong place and heads to another place. Thus, the user returns back to B (=A) and continues his/her searching. While the user visits some wrong places {W*n*} and he/she finally approaches the destination {G}. The mark {L} represents the observations in the lift.

The searching experiments were conducted multiple times with diverse visiting policies (e.g. right-weight method, left-weight method, and random visit); the result is illustrated in Figure 11(a). Figure 11(b) represents the distance to the destination for each chaser. The figure shows that some trials found the lost device more quickly than others did. Most trials reached the vicinity of the lost phone by approximately 6 meters distance. At 6 meters away from the missing mobile, a user can physically search for the device or make a call to the device. Increase of distance in the figure indicates wrong visits by the users. The users quickly recognized those mistakes and approached the vicinity of the final place. This result is acceptable since it could take a long time to find a small device in the building without help of FindingMiMo.

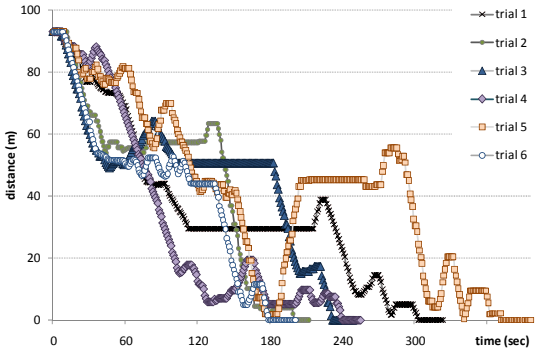
5.3 Vertical localization

Determining the exact level where the missing mobile is located is a challenge. First, neighbor floors have similar radio observations. Second, a user may use a vehicle such as an elevator.

During the previous experiment, one of the chasers visited a wrong floor that received similar Wi-Fi observations in the log. The missing mobile was on the second floor and the chaser visited the first floor. The trace similarity index was fine and the user mistakenly thought that he was on the right path. After searching his vicinity, the user returned to the elevator and continued the



(a) Foot prints of chasers



(b) Distance to the missing mobile

Figure 11. Chasing result of multiple trials

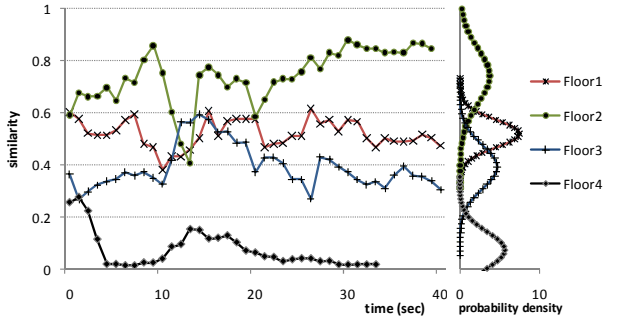


Figure 12. Path similarities from adjacent levels. The missing mobile exists on level 2 and the measurement from level 2 is distinguishable from other levels

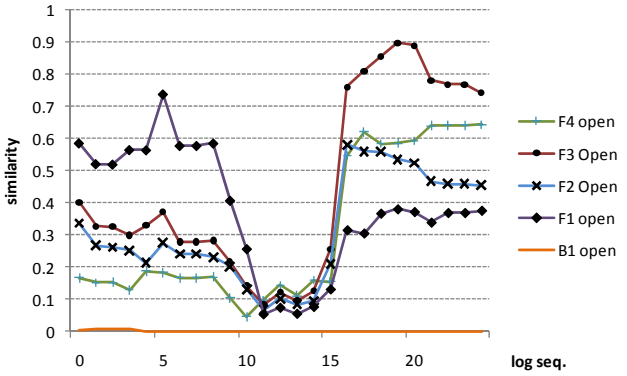


Figure 13. Log similarity in the elevator. The missing mobile gets on at floor 1 and gets off at floor 3. The chaser observes the highest similarity at floor 3.

chasing process. The fact that Wi-Fi observations measured on close levels are similar caused this mistake.

We analyzed the difference of observations from neighbor floors. We measured the trace similarity while walking along the corridors with the same layout but on different levels. The path corresponds to the moving path between about 70 to 90 seconds in the log shown in Figure 9(c). Figure 12 shows four measurements at levels 1 to 4. The missing mobile's moving path is level 2. The measured signals on level 1 and level 3 are similar to the observations in the log. The measurement from level 2 shows higher similarity, average 0.74, and it is distinguishable. However, the user can determine the right floor by visiting multiple floors. The result depends on the environment and the user may consider this situation. The chaser may visit nearby levels when the similarity index is doubtful.

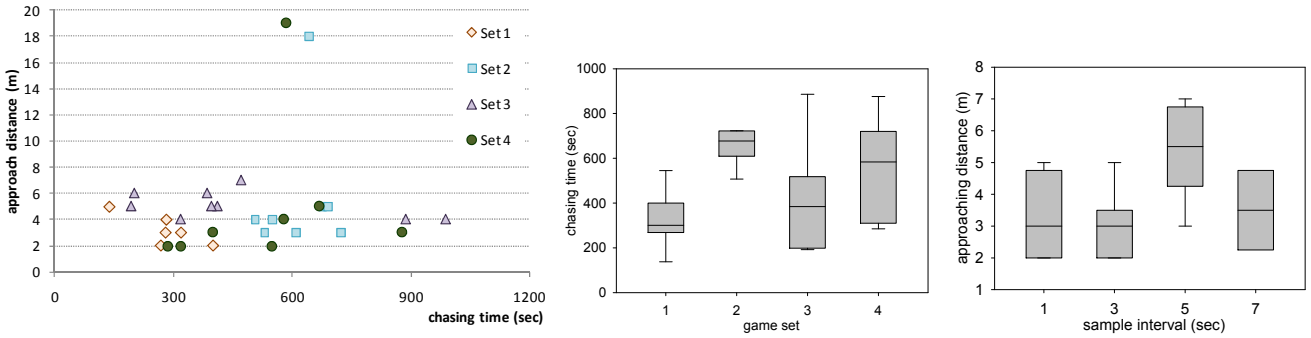
The user may use indoor vehicles such as elevators, escalators and stairs. Elevators are the worst environments in the scheme, since elevators block radio signals and access any floor. The scheme uses radio signals and continuously determines the path. We overcame this limitation by visiting every floor in the building via elevator. Although this approach is not valid in skyscrapers, this limitation is not critical in real environments. The user may visit a limited number of floors in his/her life and the user is implicitly aware of candidate floors.

We designed an experiment using elevators. The position of the elevator is given by log analysis in the previous section. The purpose is to check whether the chaser can determine the right floor where the missing mobile exited the elevator. First, the missing mobile measured the Wi-Fi logs entering the elevator at floor 1 and exiting at floor 3. Then the chaser used the elevator and accessed every floor by pressing all buttons. The elevator door opened at each floor. The chaser visited floors from the basement to level 4. The chaser analyzes the Wi-Fi observations while the doors are open. The measurement was performed inside of the elevator. Figure 13 shows the trace similarity at each floor. The observations from level 3 show the highest similarity to the right side of the logs, which is the recent part, and the difference is distinguishable. The chaser can clearly determine which floor the user should have chosen. Replication of the experiment in the neighboring elevator produced similar results.

5.4 Hide-and-Seek game

The human's intuition and knowledge affect the chasing process. To control the effect, we designed a hide-and-seek game. A participant hides a mobile device somewhere in a building and the chaser group chases the missing mobile. The goal of the game is to show that a chaser can follow the route of the missing mobile with only the information from the chaser application. The missing mobile was a HTC hero, and the chaser group used different devices such as Google Nexus One, HTC desire, Samsung Galaxy series, and Sky Sirius.

We designed four games and each game took place at four different buildings on the university campus. 36 people joined the game. Table 1 shows the environments of the games, and Table 2 shows the average chasing times and approaching distances.



(a) result plot of game sets (b) chasing time of game sets (c) approach accuracy by sample interval

Figure 14. Chasing time and approach accuracy. (a) chasing time and approach accuracy of multiple game sets (b) chasing time distribution of game sets. Boxes represent 50% of result. Outliers represent 5% and 95%. The lines in a boxes are medians. (c) sampling interval effect. The figure shows the approach distance to the missing phone in successful cases. No meaningful difference is found.

Figure 14 (a) and (b) show the chasing times and approach distances from the missing devices. Most of the chasers approached the vicinity of the missing mobiles. Only three trials failed; a case was considered a failure if the chasing time exceeded 30 minutes or if the chaser's approach distance to the device was over 15 meters. In successful cases, the participants approached the location at 4.1 meters distance within 425 seconds, on average. Since the chasing time includes waiting time for elevators, the results of the chasing times do not have meaningful differences.

The sampling interval is one of the key factors for energy consumption. Adjusting the sampling interval is a trade-off between energy consumption and the quality of information. We tested the games with logs measured with various sampling intervals. Figure 14 (c) shows the approaching accuracy of each successful case with various sample intervals. There is no meaningful relationship between performance and the sample interval of the missing mobile under 10-second intervals. In an additional experiment, some participants commented that the log with 10-second intervals had too few radio samples to find the location, although they were able to follow the trail of the hidden mobile. The sample interval is a tradeoff between the density of information and energy consumption. Determining the best sample interval is not trivial, but people found experiments with logs by 10-second sample intervals inconvenient.

Additionally, we performed the additional game at building A where one participant hid the mobile device in the parking garage, which was on basement level 2. No Wi-Fi access point was installed on the floor. On average, four faint signals from adjacent floors were observed. Even with these weak signals, a chaser succeeded in tracing the missing mobile within ten meters distance. This result also proves that the chasing process can be successful, even in a situation where few access points are available. Below basement level 2, however, no signal was observed and chasing was not possible.

The hide-and-seek game proves that the chaser's information guides the user to approach the missing mobile without a *priori* knowledge of the location. In real situations, the user may be aware of his/her walking route and the chasing process would become much easier.

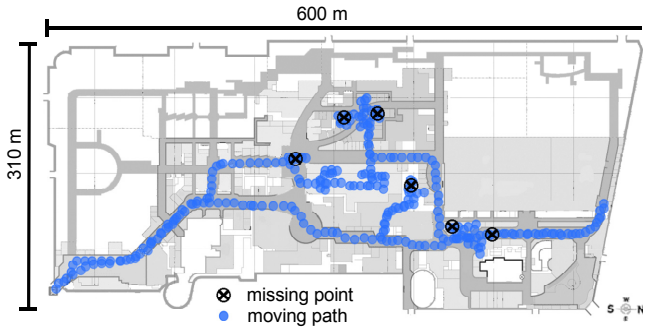


Figure 15. Movement of the device and missing points. Solid circles represent the moving route of the missing mobile and crosses represent the places where the device is missing.

Table 1. Game Environment

Game	1	2	3	4
Building	A	B	C	D
Area (m^2)	6,505	5,366	3,482	3,646
Moving distance (m)[i]	116	183	105	117
Number of floors	9	6	9	8
Number of APs[ii]	70	206	139	122
Number of participants	6	9	9	8

i: Moving distance of a hider. Vertical movement is not considered.
ii: Number of access points observed by a hider

Table 2. Chasing result

Game	1	2	3	4
Approaching distance (m)	3.25	3.5	5.1	6.2
Chasing time (sec)	526	615	411	546
Failure[i]	0	2	0	1
WPS location error (m)[ii]	37	15	39	45

i:The cases that take over 30 minutes or approach no closer than 15 meters
ii: Location service error provided by the network provider (indoor)

5.5 Case study: Shopping mall

For a more realistic scenario, we conducted experiments in a real shopping mall. The participant carried the mobile device and visited a large shopping mall, the Seoul COEX convention center, which is the largest underground shopping mall in Korea (about 195,000 m^2 area). The complex has many amusing places, restaurants and shops. The carrier explored the place for about three hours. We assumed that the device would be lost at a random time during shopping. From the three-hour log, we extracted six possible missing places and generated the logs. The next day, the user randomly selected one of the logs and chased the location of the missing event. Figure 15 shows the map of COEX and the exploration route. The result is shown in Table 3. The chaser succeeded in all scenarios and the search was completed in 10 minutes, on average. This result is indeed impressive because the average chasing route was 459 meters. The user can quickly find the locations because, although the user does not know the exact location of the lost device, he/she is aware of the route he/she explored the previous day. The scenario is realistic; a participant performed the chasing process with a different device the next day.

Table 3. Chasing results for six missing events

Place	Distance (m)	Route (m)	Time (sec.)
Beverage	360	360	386
Shop 1	368	368	374
On a path	395	395	373
Restaurant	365	541	500
Shop 2	345	523	783
Rest Room	315	567	933

6. DISCUSSION

6.1 User Guide

The FindingMiMo chasing process is not an automatic one. A user visits a place and performs the chasing by him/herself. The accuracy of the process depends on the user's searching skill and/or knowledge on the system. This section discusses guideline to use our scheme better.

A mobile device generally has limited resources. A device samples a Wi-Fi observation and analyzes the similarity feature in every second. The similarity report is displayed with the delay, the user may misunderstand. Considering the delay, the user would experience better results walking slowly.

FindingMiMo employs Wi-Fi fingerprinting for landmarks. The Wi-Fi signal can penetrate windows or thin objects. Wi-Fi fingerprints, measured at two places in the vicinity, are similar to each other, even where two places are located on different levels in a building. This feature may make the chasing process confusing. The user should consider other places that observe similar radio signals. If the user receives a similar observation from the log and does not find the missing mobile, he/she should consider visiting another floor. Perhaps, the same observation can be found on an upper or lower floor.

6.2 Errorneous Data from the Chaser

Altitude information is difficult to obtain from GPS reading. If the current GPS reading is measured at a high floor, the reading could confuse the user. GPS can be read via windows or on a bridge between buildings. As we consider the last GPS reading as the start position (i.e., entry point of the building), a major part of the log can be omitted or the user could start from the wrong place.

Another confusing situation is the signals measured from an open space, where the signal variation in the location is insignificant. One of the experiment sites has a bridge over a lobby. Similar Wi-Fi observations were obtained both in the lobby and on the bridge. This situation confuses the chasing process, because the user cannot distinguish the places based on the ambient observation. In addition, the log reducing process may generate a route that the user cannot physically follow.

For energy efficient ambient logging, the logger operates with a sleep schedule. The logger sleeps where the mobile device stays still. The log may have a gap in observations during the sleep interval.

6.3 Efficient Ambient Logging

One of the issues in ambient logging is the policy applied to the sampling schedule. The ideal approach is to sample the ambient feature only when the user is moving. To detect the user's movement, the accelerometer can be employed. Observing the accelerometer, however, requires continuous CPU operation, which consequently consumes non-trivial amounts of energy. Work in [8, 32] proposed the duty cycling of accelerometers to reduce energy consumption for continuous sensing. This method can be integrated into our system to reduce energy consumption for movement detection. The asymmetry multicore processor (AMP) approach has been studied [33, 34], which employs multicores that have different performance, complexity and power consumptions. By employing AMP, a mobile device can continuously observe sensors with low energy consumption. With this technique, the ambient logging becomes much less power hungry since the stationary Wi-Fi sampling is not required.

The periodic GPS sampling is, in fact, an overhead since a GPS device has long bootstrapping delay and heavy energy cost. The missing mobile observes the GPS signal for outdoor location estimation. When a network operator or a location service provider provides accurate outdoor location, the missing mobile can use these platforms for outdoor logging. For example, Skyhook Inc provides a Wi-Fi fingerprint map and its location estimation is highly accurate in outdoor environments. Since the missing mobile keeps monitoring the Wi-Fi signals, Google map, for instance, can serve the location of the mobile device. With this approach, the energy consumption of the GPS module is replaced with that of the localization service.

6.4 Robustness

We assume that the mobile device is on when the user notices the device is missing. In a real situation, the device may be off for some reason; the battery may run out or someone who picks up the device may turn it off temporary. In the former case, recharging of the device is not guaranteed. Since we assume that the missing mobile application operates as a system thread, it can monitor the battery level and upload the current log to a server before the system turns off. In the latter case, the finder may turn on the device for private usage or for returning it to its owner. A server may periodically check the status of the missing mobile. At any time the device wakes up, the log can be transmitted and the chasing process becomes available.

6.5 Application Opportunities

Use of the FindingMiMo system would lead to many interesting applications. For example, the system can be used to find a missing child or a pet. In this example, the mobile device is implemented as a smart tag that embeds an ambient logging module, GPS module and a communication module. In detail, the wristlet-type smart tag is attached to a child's wrist. A pet can wear a neck-tag. The tag observes the Wi-Fi channel conditions and logs them. The chaser application in the parent/owner's smartphone also observes the Wi-Fi channel conditions. At any time Wi-Fi observations from two devices differentiate, the chaser alarms to notify the absence of the child or the pet. When the child or pet are missing, the chaser traces the track of the child or pet via the same process of FindingMiMo. A Wi-Fi module is cheap and Wi-Fi is a well-established infrastructure. It also has communication abilities. If service providers were to open their resources for public service log transmission, this scenario becomes realistic, and valuable.

7. CONCLUSION

This paper proposes a scheme to find a lost mobile phone using daily logging and chasing applications, especially in indoor environments. We believe that a simple localization method is not sufficient to find the phone under certain conditions. The FindingMiMo solution provides intermediate guides to the final location of the missing mobile phone. It provides a visit-and-test method to determine the relationship between the physical locations and radio observations even where the log from the lost phone does not carry its geometric location. The information is provided by a graphical interface on screen in run-time. As a result, a user can navigate inside a building and approach the 'lost phone destination'.

The work reported in this paper is practical and the result is successful. The logging cost is a reasonable insurance fee for the option of being able to locate a missing phone. This application is available to mobile devices equipped with inertial sensors, GPS, and Wi-Fi modules. We hope that our solution allows people to find and keep personal property and contributes to an increasingly mobile society.

8. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No.2011-0000156).

9. REFERENCES

[1] Apple. MobileMe. Web Site. http://www.me.com

[2] Skyhook. XPS. Web Site. http://www.skyhookwireless.com/howitworks/

[3] N. Klepeis, W. Nelson, W. Ott, J. Robinson, A. Tsang, P. Switzer, J. Behar, S. Hern, and W. Engelmann. The national human activity pattern survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Analysis and Environmental Epidemiology*, vol. 11, 2001, pp. 231-252.

[4] P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In *Proccedings of IEEE International Conference on Computer Communications* (INFOCOM 2000). Tel-Aviv, Israel, March 2000..

[5] B. Li and J. Salter, A. Dempster, and C. Rizos. Indoor positioning techniques based on wireless LAN. In *Proccedings of Wireless Broadband and Ultra Wideband Communications* (AusWireless 2006), Sydney, Australia, March, 2006.

[6] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis. On Indoor Position Location with Wireless LANS. In *Proccedings of IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications* (PIMRC 2002). Lisboa, Protugal, September. 2002.

[7] W. M. Yeung, J. Zhou, and J. K. Ng. Emerging Directions in Embedded and Ubiquitous Computing. Springer, 2007, ch. Enhanced Fingerprint-Based Location Estimation System in Wireless LAN Environment, pp. 273–284.

[8] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proccedings of ACM Conference on Embedded Networked Sensor Systems* (SenSys 2010) , Zurich, Switzerland, November, 2010.

[9] Y. Ma, R. Hankins, and D. Racz. iLoc: a framework for incremental location-state acquisition and prediction based on mobile sensors. In *Proccedings of ACM Conference on Information and Knowledge Management* (CIKM 2009), New York, NY, USA, November 2009.

[10] K. Yap, v. Srinivasan, and M. Motani, MAX: Human-centric search of the physical world. In *Proccedings of ACM Conference on Embedded Networked Sensor Systems* (SenSys 2005), San Diego, CA, USA, November 2005.

[11] C. Decker, U. Kubach, and M. Beigl, Revealing the retail black box by interaction sensing. In *Proccedings of IEEE Internaltional Conference on Distributed Computing Systems* (ICDCS 2003), Providence, RI, USA, May 2003.

[12] C. Frank, P. Bolliger, F. Mattern, and W. Kellerer. The sensor internet at work: Locating everyday items using mobile phones. *Pervasive and Mobile Computing*. Elsevier, vol. 4 no. 3, June 2008, pp. 421-447.

[13] Wibree Technology. www.wibree.com

[14] Microsoft. Windows Phone. http://www.microsoft.com/windowsphone/en-gb/howto/wp7/start/find-a-lost-phone.aspx

[15] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell. Soundsense: scalable sound sensing for eoplecentric applications on mobile phones. In *Proccedings of ACM Internation Conference on Mobile Systems, Applications, Services* (MobiSys 2009), New York, NY, USA, June 2009.

[16] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience ingerprinting. In *Proccedings of ACM International Conference on Mobile Computing and Networking* (MobiCom 2009), New York, NY, USA, September 2009.

[17] H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In *Proccedings of ACM Internation Conference on Mobile Systems, Applications, Services* (Sensys 2010), Zurich, Switzerland, November 2010.

[18] UbiSense. Web Site. http://www.ubisense.net

[19] 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 *Proccedings of International Conference on Pervasive Computing* (Pervasive 2005), Munich, Germany, May 2005.

[20] B. Krach and P. Robertson. Cascaded Estimation Architecture for Integration of Foot-Mounted Inertial Sensors. In *IEEE/ION Proccedings* of *Positioning Location and Navigation Symposium* (PLANS 2008), Indian Wells, USA, May 2008.

[21] P. Robertson, M. Angermann, and B. Krach. Simultaneous Localization and Mapping for Pedestrians using only Foot-Mounted Accelerometers. In *Proccedings of ACM*

International Conference on Ubiquitous Computing (UbiComp 2009), Orlando, USA, September 2009.

[22] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications*, IEEE, vol. 25, no. 6, November 2005, pp. 38–46.

[23] A. Bernheim Brush, A. Karlson, J. Scott, R. Sarin, A. Jacobs, B. Bond, O. Murillo, G. Hunt, M. Sinclair, K. Hammil, and S. Levi, User Experiences with Activity-Based Navigation on Mobile Devices. In *Proccedings of ACM International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI 2010), Lisboa, Portugal, September 2010.

[24] D. Rogers and T. Tanimoto. A Computer Program for Classifying Plants. *Science 21*. October 1960, pp. 1115-1118.

[25] Y. Chon and H. Cha. LifeMap: A Smartphone-based Context Provider for Location-based Service. *Pervasive Computing*. IEEE, vol. 10, no. 2, April-June 2011, pp. 58-67.

[26] Y. Chon, E. Talipov, and H. Cha, Autonomous Management of Everyday Places for Personalized Location Provider. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* (SMCC), in press.

[27] H. Shin, Y. Chon, and H. Cha. SmartSLAM: Constructing an Indoor Floor Plan using Smartphone. Yonsei University, Tech. Rep., 2010. MOBED-TR-2010-2.

[28] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *Proccedings of ACM Internation Conference on Mobile Systems, Applications, Services* (MobiSys 2010), San Frasisco, USA, June 2010.

[29] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*. Springer-Verlag New York, Inc., 1990, pp. 167-193.

[30] Y. Chon, E. Talipov, H. Shin, and H. Cha. Mobility Prediction-based Smartphone Energy optimization for Everyday Location Monitoring. Yonsei University, Tech. Rep., 2010. MOBED-TR-2010-3.

[31] H. Falaki, R. Mahajan, and S. Kandula. Diversity in Smartphone Usage. In *Proccedings of ACM Internation Conference on Mobile Systems, Applications, Services* (MobiSys 2010), San Fransisco, USA, June 2010.

[32] J. Paek, J. Kim, and R. Govindan, Energy-efficient rate-adpative GPS-based positioning for smartphones. In *Proccedings of ACM Internation Conference on Mobile Systems, Applications, Services* (MobiSys 2010), San Fransisco, USA, June 2010..

[33] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullesen. Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction. In *Proccedings of International Symposium on Microarchitecture* (MICRO 2003), San Diego, CA, USA, December 2003.

[34] J. Li and J. F. Martinez. Dynamic Power-Performance Adaptation of Parallel Computation on Chip Multiprocessors. In *Proccedings of International Symposium on High-Performance Computer Architecture* (HPCA 2006), Austin, Texas, USA, February 2006.