SensLoc: Sensing Everyday Places and Paths using Less Energy

Donnie H. Kim, Younghun Kim†, Deborah Estrin, Mani B. Srivastava†
UCLA CSD CENS, UCLA EED NESL†
{dhjkim, destrin}@cs.ucla.edu, {kimyh, mbs}@ucla.edu†

Abstract

Continuously understanding a user's location context in colloquial terms and the paths that connect the locations unlocks many opportunities for emerging applications. While extensive research effort has been made on efficiently tracking a user's raw coordinates, few attempts have been made to efficiently provide everyday contextual information about these locations as places and paths. We introduce SensLoc, a practical location service to provide such contextual information, abstracting location as place visits and path travels from sensor signals. SensLoc comprises of a robust place detection algorithm, a sensitive movement detector, and an on-demand path tracker. Based on a user's mobility, SensLoc proactively controls active cycle of a GPS receiver, a Wi-Fi scanner, and an accelerometer. Pilot studies show that SensLoc can correctly detect 94% of the place visits, track 95% of the total travel distance, and still only consume 13% of energy than algorithms that periodically collect coordinates to provide the same information.

Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Real-time and embedded systems

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Semantic Location Context, Energy-efficient Tracking.

1 Introduction

As mobile devices have become capable of locating themselves almost all the time, a variety of mobile applications have emerged that seek to continuously track a user's location context. For instance, geo-reminders allow us to set and receive a to-do list whenever we enter or leave a particular place [22, 27]. Social applications plan to provide services for seamlessly sharing whereabouts, querying users that are presently located at an art gallery, and inferring hotspots by the frequency of physical visits by users [8, 7]. Tracks generated by humans also provide useful information for map building, traffic estimation, and ride sharing [30, 1]. Moreover, automatically detected visit and travel behaviors can help studies of human spatial and temporal behavior, and support research for urban planning, sustainability, epidemics, and health care [9, 28]. Interestingly, all these applications can benefit from continuously understanding and keeping track of location as people normally do: places and paths. By automatically learning the places that one visits throughout one's daily life, noticing when one enters and leaves these places, and remembering paths one travels between them, we can unleash many interesting applications.

An obvious choice for tracking a user's location context today is to periodically collect coordinates from available positioning systems (*e.g.*, GPS) and directly provide them to applications. Places of interest are manually defined by drawing a circle or a polygon ahead of time, and paths are parsed from day-long traces by post-processing algorithms, if not done manually. However, we argue that such schemes fail in discovering many interesting indoor places, struggle to scale, and consume unnecessary energy. Most of the places we go and stay are indoors, and even a single building (or adjacent ones) can contain multiple places especially in dense urban environments. Unfortunately, this is where current positioning systems suffer in providing accurate position fixes. Manually delineating and labeling places from scratch one by one also does not scale and can omit interesting places that we are less conscious of having visited. Moreover, continuously tracking a person's location comes with a significant energy cost, discouraging potential users. Current track-based applications either cope with a reduced sampling rate with lower fidelity or depend on users to manually start and stop tracking. Recently, many research efforts have focused on efficiently tracking user's coordinates while retaining the distance-error bound specified by applications [20, 25, 6, 15]. While these algorithms can provide location traces in an energy efficient way, they do not provide everyday location context in a semantically meaningful way.

In this paper, we present SensLoc, a system that provides user's location context as places and paths while reducing

This work was supported in part by NSF Grant CNS-0529235.

SenSys'10, November 3–5, 2010, Zurich, Switzerland.

its impact on the device's battery life. We show that interpreting location closer to our semantics not only provides a richer set of context information, but also plays a key role in designing an energy-efficient location tracking mechanism. Studies have shown that people spend approximately 89% of the time indoors and 5% in a vehicle with the remaining 6% spent at outdoors [16]. Thus, GPS, which provides accurate position fixes when it has a clear view of sky, may only be needed for about 10% of the time while other adaptive and more energy-efficient mechanisms should be used to detect semantic indoor places for the majority of the time. The key challenges we face to provide such service are 1) accurately detecting places closer to our semantics, 2) automatically parsing travel paths from day-long location traces, and 3) minimizing energy consumption.

We overcome these challenges by designing a robust place detection algorithm, a sensitive movement detector, and an on-demand path tracker. A place detection algorithm attempts to automatically find places (colloquial representations of locations such as "my office' or "5th floor cafe") that carries a semantic meaning to an individual user. Semantic places are directly inferred from pervasive radio signals by periodically scanning neighboring beacons. To reduce energy consumed during a stay at a place, scans are suspended while a movement detector detects no movement from a more energy-efficient inertial sensor. A path is defined as a set of time series coordinates that interconnects places. Paths are tracked by acquiring periodic position fixes from position systems only when traveling between places.

Our main contributions are as follows: We 1) propose a new abstraction of continuous location: places and paths, 2) present a framework that provides location context as places and paths using less energy, and 3) provide quantitative studies illustrating expected performance and energy cost when used everyday.

To evaluate our framework, we gathered three different data sets from both real-life and scripted-tours. Five individuals collected data for a week and two people for four weeks as they went about their normal lives. A scripted-tour data set comprised of 50 visits to 25 different places people go often near a campus. Each volunteer also kept a written diary of places they visited with enter and exit times. Using these data sets, we evaluate SensLoc's effectiveness in detecting place visits, tracking travel paths, and its overall energy consumption during a daily operation. While the performance and cost indeed depends on a user's surrounding and travel patterns, we show that SensLoc consistently outperforms previous place learning techniques, promptly tracks paths, and saves significant energy.

2 System Overview

We first describe a high level usage scenario of the system, and then present the internal details. As SensLoc runs in the background of the mobile device, places are gradually learned as a user visits them and spends a substantial amount of time. A new place is learned by saving its place signature whenever a visit to an unknown place is detected, and sometime later in the day asking the user to confirm and tag a name, such as "home", "Fred's office", or "Organic foods @westwood". A user can recall the place by looking at the visit time, presented as enter and leave time, and the associated geographic coordinate, plotted on a map, provided as a hint. Revisited places are recognized using previously saved place signatures. Entrance to and departure from selected places are notified to applications requesting the *place detection* service. When a user leaves a place, *path tracking* (if enabled) is initiated until the user arrives at another place. Any positioning system available on the device can be used including GPS or systems supported by energy-efficient mechanisms [20, 25, 6, 15] to track paths. If *path recording* is requested, paths are saved, and provided to various applications requesting the service. Unrecorded path tracking can also provide real-time current positions to navigation and location-based search applications with minimum delay by periodically updating the user's current position. This is also when real-time positions are most likely used (*e.g.*, when I'm mobile), and quick responses are most appreciated (*e.g.*, when I'm lost).

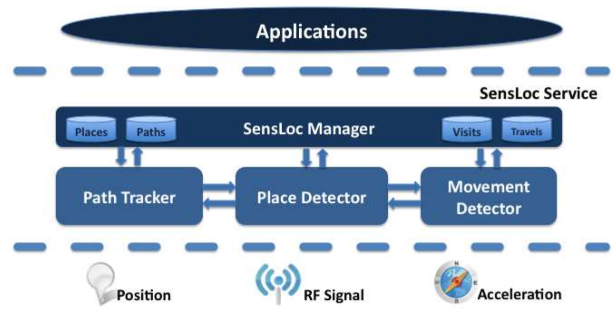
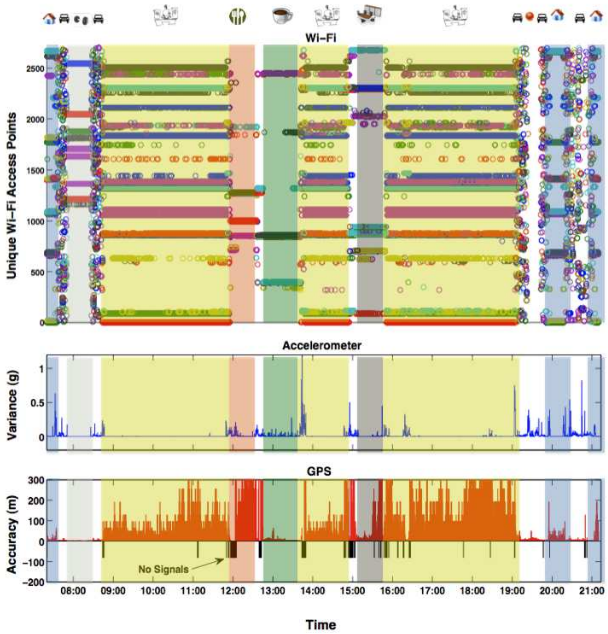
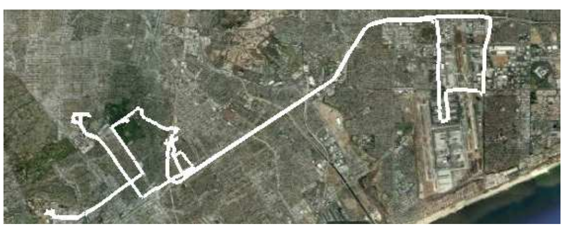


Figure 1. System Architecture

Figure 1 presents the overall architecture of SensLoc. The system consists of three main building blocks to provide its service while reducing its energy requirements: place detector, movement detector, and path tracker. In the next section, we describe a particular set of algorithms using GPS, Wi-Fi, and accelerometer that implement these architectural elements, but other algorithms can be deployed. The place detector regularly scans neighboring radio beacons to detect place visits when the radio environment stabilizes indicating an entrance. Once an entrance is determined, the place detector consults with the place database to recognize the place and triggers the movement detector to find opportunities to sleep. If no movement is detected, the movement detector signals the place detector to sleep, and awakens it when a movement is detected again. When the place detector senses that the surrounding radio environment is changing, it declares a place departure, saves the visit history, turns off the movement detector, and powers on the path tracker. Path tracking is initiated and records the path to the path database (if enabled) until the next place visit. Path tracker can also hint the place detector to sleep when the user is traveling at high speeds, and unlikely to approach a place anytime soon. We use Wi-Fi access points (APs) to sense places, accelerometer to detect movements, and GPS to track paths. We describe the details next.



(a) Sensor data



(b) Corresponding Location Trace

Figure 2. Location and sensor traces from a single day following normal routines. The icons on the top row illustrate the ground truth. Each dot in the Wi-Fi scatter plot is a beacon found from a scan. Acceleration magnitude variance is computed over 10 second window with 50% duty-cycling. Accuracy value is reported from the GPS module.

3 SensLoc Algorithms

The biggest challenge facing SensLoc is accurately identifying place visits and path travels while minimizing energy usage. We use a novel place visit inference technique, take a hybrid approach to save energy, and track paths only when traveling between places.

3.1 Place Detection

Detecting place visits involves two steps: sensing a stable radio environment that indicates an entrance to a place and detecting significant changes signaling a departure. Intrinsic noises in the signals caused by multi-path, signal fading, and interference make such task challenging. Even when staying at a place, beacons may be seen intermittently, particularly when transiently traversing edge of certain APs. Placement of the device near a human body also causes interference and irregular beacon losses. Beacons are also typically not confined to a single place. Yet, humans are creatures of habit constraining their movement in certain areas (even at a place). As depicted in Figure 2(a), surrounding radio signals can well approximate human location interpretation,

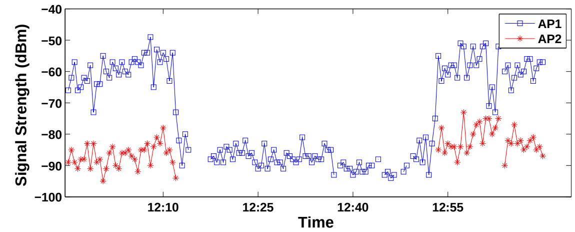


Figure 3. Changes in signal strength when a user visited two places in different floors. Relying on the absence of every representative beacons to detect departure fails in detecting direct visits to adjacent places.

and others have shown that a well defined set of beacons can overcome the noisy radio environment [11, 14].

Our place detection algorithm is built upon previously proposed ideas. However, we improve its ability to discriminate close places governed by common beacons and reduce false place detections by exploiting signal strength changes and adapting to diverse beacon densities. Like other approaches, a scan window, as opposed to a single scan, is used to tolerate noisy radio signals and beacon losses. A window size w defines the number of scans used for each operation. Sliding windows are used to provide more accurate detection as opposed to non-overlapping windows. Locally administered APs (*e.g.*, local networks created by laptops or mobile phones) are pre-filtered to rely on static APs. These can be distinguished by examining the second least significant bit of the most significant byte of the mac address. This simple filtering helps significantly in reducing false place detections.

Similar radio environment is determined using the Tanimoto coefficient [12], which is widely used for measuring similarity between two fingerprints F_1 and F_2. The Tanimoto similarity penalizes a small number of shared entries (beacons) more than the cosine similarity and is defined as:

$$T(F_1, F_2) = \frac{F_1 \cdot F_2}{||F_1||^2 + ||F_2||^2 - F_1 \cdot F_2}$$

Input Wi-Fi scans are transformed into vector space so that the Tanimoto coefficient can be used. The attribute vectors are the signal strength vectors of the fingerprints. Our algorithm uses a group of scans (determined by a window size w) to infer fingerprints, defined by the list of beacons, combined with their signal strength and response rate. Response rate is the ratio of the detection count and the total number of scans for each beacon, and has been found to be more robust in predicting distance than signal strength [4]. The mean of the signal strength is calculated ignoring zero values and over the selected group of scans; Zeros are assigned when the beacons are not detected. Both entrance and departure detection use this similarity measure, but the scope of the included scans and beacons in each figerprint differs slightly.

Entrance Detection. Continuously seen similar scan windows imply potential entrance to a place. Similar to other prior works, we determine an entrance when c_{max} consecutive scan windows pass the similarity test. We differ from these works by using the Tanimoto coefficient to measure similarity. To start an examination of a potential stay at a place, the current scan window is saved and compared against the following scan window. A certainty value c is increased when the Tanimoto coefficient of the previous and current scan window is above threshold t_{sim}. Otherwise, we reset c to 0, and clear the fingerprint. Scan windows determined to be similar are merged for comparison against the next scan window. A place entry is declared when c reaches c_{max}. As yet not enough statistics of a place are gathered for more insightful decisions; therefore, we take a conservative approach by including all beacons detected in two fingerprints when computing the coefficient.

Departure Detection. Scan windows that are dissimilar from the current place's fingerprint indicate that the RF environment is changing and implies leaving a place. Accumulated beacon statistics during the stay are used to make more careful examinations in detecting departures. To suppress influences of infrequent beacons, only a subset of beacons are used to measure similarity. After entering a place, we select representative beacons with a response rate higher than the threshold r_{rep}. The vector space is reduced to representative beacons, and the similarity score is evaluated over this subspace. The certainty value c (ranging from 0 to c_{max}), which reached c_{max} during entrance, decrements when the similarity score is below t_{sim}, and increments otherwise. A place departure is declared when c reaches 0. The Tanimoto coefficient robustly detects adjacent places than previous techniques [11, 14] by exploiting signal strength changes and penalizing the disappearance of a subset of representative beacons. Figure 3 depicts a simplified real-life example of signal changes in subsequently visited adjacent places sharing a subset of strong beacons.

The scheme above robustly detects place visits when a place has at least one beacon consistently detected during a stay. However, detecting places with weak beacon signals still remains as a challenge. To cope with these places and improve our place coverage, we adjust our two parameters, representative beacon threshold r_{rep} and window size w, to the radio environment. Threshold r_{rep} is adaptively defined by observing the highest response rate from the detected beacons and subtracting r_{step}. If the subtracted value is below minimum threshold r_{min}, r_{min} is used as r_{rep}, and window size w is doubled to include more scans and deal with the sparse radio environment. We show the effects of these techniques in Section 4.3.3.

Place Recognition. We apply the same Tanimoto coefficient used in detecting place visits to recognize revisited places. Once the entrance is determined, the coefficient is evaluated over the representative beacons of two fingerprints in comparison. During the recognition phase, if the two fingerprints have a coefficient higher than a threshold t_{sim}, then fingerprints are deemed similar.

3.2 Movement Detection

An accelerometer monitors movements and finds opportunities to save energy when the device is stationary. Design goals of our movement detector are 1) low-power usage, 2) robust detection regardless of random orientations, and 3) low tolerance to movement. We duty cycle the accelerometer 50% by setting the duration to 5 seconds and the period to 10 seconds to reduce the energy consumed by the sensor. Acceleration magnitude is computed over all three axis to tolerate random orientations of the device. To detect movement, we compute the variance of the magnitude over a sliding window with window size w_{acc}.

We conservatively find sleep opportunities to preserve accuracy, as a couple of more Wi-Fi scans are not extremely expensive. Our main targets are long-duration sleep opportunities when the device is left alone. The accelerometer is turned on when a user enters a place and stays for more than five minutes. We avoid immediately checking sleep opportunities to provide enough time to accumulate beacon statistics of the place. When the c value used for detecting place visits is at c_{max}, indicating a steady stay at a place, the variance of the magnitude is compared against a conservative threshold v_{mov}. If the variance is below the threshold, beacon scans are postponed until the variance is over the threshold again. Whenever a movement is detected, we reactivate the scans for at least 5 minutes to prevent missing place departures when the user, for example, walks with the device in hand causing low variance in acceleration.

3.3 Path Tracking

Physical location of the device is periodically traced using available positioning systems while traveling between places. We collect GPS fixes periodically for path tracking but other positioning systems or energy saving mechanisms may be used [20, 25, 6, 15]. Sampling interval of position fixes is a tunable parameter depending on the application needs. As shown in Figure 2(a), in general, we spend a non-negligible amount of time indoors where GPS has low accuracy and travel long distances in comparatively small amount of time. Well-timed path tracking allows us to save energy when the marginal utility of additional position fixes is low, and aggressively localize when the user is heading elsewhere.

SensLoc enables path tracking when its place detection algorithm declares place departure. SensLoc additionally saves unnecessary beacon scans when traveling at high speeds. Wi-Fi scans are turned off, especially during long drives when the average of the speed estimation (provided by the GPS module) over a sliding window (with size w_{gps}) exceeds threshold value 2 m/s (average human walking speed is 1.3-1.5 m/s). The assumption here is that the user will slow down when approaching a place. Tracking is powered off when place entrance is determined. We do not apply blacklisting Wi-Fi signatures based on past-experience [20] to predict when GPS positions are not available and is left as future work. However, the transition time between indoor places are typically short and does not significantly impact the overall battery life.

4 Evaluation

We evaluate SensLoc using three different data sets collected from both real-life and scripted tours. First, we use a data set collected by five persons following their normal lives for a week to examine how the performance and energy cost are affected by different mobility patterns and environment. We illustrate that while the performance and cost indeed depend on a user's surroundings and travel patterns, SensLoc robustly detects place visits better than previous approaches and requires less energy overall. We also show that well-timed path tracking allows us to collect GPS coordinates only when they matter the most while at the same time saving energy. Next, a data set collected by two persons over four weeks is used to evaluate how well SensLoc learns the places they visit and recognizes revisits over a month of usage. We also discuss the paths SensLoc found and the fluctuation in energy consumption over the four weeks. Finally, we use a data set carefully designed to illustrate the strength of our new place detection algorithm in discriminating closely located places and detecting places with weak beacon signals. A step-by-step experiment is presented to show the effects of our improvements and parameters.

We start with explaining how we collected the above data sets, and the metrics we used to evaluate our place discovery performance. As a place typically does not have a universal shape or size, we depend on diaries kept by each user to act as the ground truth as others previously suggested. Then, we explain two previously proposed place learning algorithms we implemented for comparison and delve into our results. We emphasize that none of the previous place learning techniques experimentally studied their energy requirements and we are the first to benchmark them.

4.1 Data Collection

We collected sensor data traces using HTC G1 mobile phones, equipped with an integrated GPS, Wi-Fi, and accelerometer. The phones were loaded with custom software configured to collect GPS, Wi-Fi, and accelerometer traces every second. They also came with a voice/data plan and data collectors were encouraged to use them as they normally use their mobile phones.

Two individuals collected data for four weeks and five persons for a week as they went about their normal lives. Data was collected mainly within two city limits in different continents, while a couple of traces were also collected in other cities during short trips. To collect ground truth, we asked each data collector to keep a diary of the places they stayed during the data collection period along with the entry and exit times. After each data collection, we plotted the GPS coordinates on a map and reviewed the results with the data collectors to help them make sure their log entries were as complete as possible. These diaries and maps provided the ground truth information about the actual places the data collector visited, and the times they entered and left those places. However, as GPS data was not available (or accurate) in many indoor locations where people spent most of their time, we had limitations on achieving perfect ground truth. At times, participants forgot to log short visits or the time they actually entered or left a place.

We additionally arranged a set of scripted tours of 50 visits to 25 different places. Each data collector individually selected five places they often go to near campus ahead of time and visited them twice. Data collectors were inclined to select shortest paths, and the travel sequence included direct visits to closely located places such as neighboring stores, and rooms separated by a single floor. Places included various snack bars, stores, cafeterias, and lab rooms in 8 different buildings on campus, two outdoor plazas, and 5 different

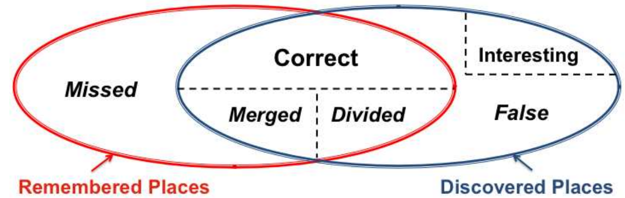


Figure 4. *Remembered place* (by users) and *discovered places* (by place detection algorithms). More *correct* and *interesting* places indicate better performance, while the distribution of erroneous places allows us to understand the strength and weakness of each technique. *Interesting* places are usually brief visits that were mistakenly unrecorded.

markets and stores near campus. Distance between places varied from 1 to 10 minutes by a normal walk. Data collectors were asked to stay at a place for at least 8 minutes and carefully record the entrance and departure times.

4.2 Evaluation Methods

Place. To quantitatively evaluate the effectiveness of detecting place visits closer to how people normally perceive places, we use a set of meaningful and erroneous places defined in [14] (Figure 4). We ask human participants to log any place they visited and stayed for more than 5 minutes and use this as our ground truth[1], rather than attempting to define a place geographically. Places recorded by users are called *remembered places* and places discovered by place detection algorithms are called *discovered places*. Places that are both recorded and discovered are further categorized as *correct*, *merged*, and *divided*. If two different places are discovered as a single place, the place is labeled as *merged*, and if a single place is divided into two or more, it is labeled as *divided*. Others that were both remembered and discovered are classified as *correct*. Recorded but not discovered places are called *missed*. Places that are not recorded but discovered are further classified *interesting* if the user claims it was mistakenly unrecorded, and *false* otherwise. We further define *precision* and *recall* as follows:

$$Precision = \frac{\#\,Correct + \#\,Interesting}{\#\,Discovered},\ Recall = \frac{\#\,Correct}{\#\,Remembered}$$

Finally, the accuracy of detected entrance and departure times are measured by the difference between the time determined by a place discovery technique and the time manually recorded by data collectors.

Path. Obtaining ground truth of the travel distance is fundamentally challenging. Recording every sidewalk, crossroad, and turn is accurate but very costly, especially when collecting real-life data for multiple days. Instead, we aggressively clean the GPS samples to estimate travel distance and use this data as our ground truth. Filtering GPS data is necessary as it is subject to errors up to several km (a.k.a. jumps) and outages indoors. We use three criteria to filter noisy GPS samples: 1) accuracy value above 30 m, 2) visible satellite number less than four (a GPS receiver requires at

[1]Home is regarded as a single place and commercial places are associated with stores. In office areas, small rooms within a few seconds of walking is considered as one place, while others that fall outside this determination are considered distinct places.

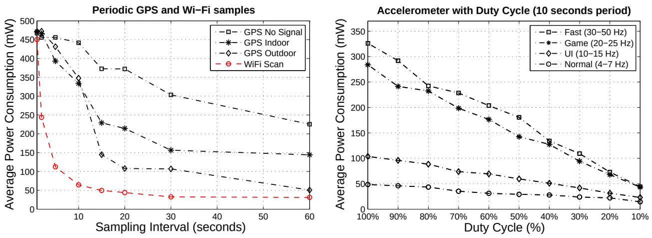


Figure 5. HTC G1 Energy Profile. Two Agilent 34410A digital multimeters are used to measure power consumption (through external power supply) while running our software. The power data is downloadable from here: http://urban.cens.ucla.edu/resources/g1power/.

least four satellites for positioning [25]), and 3) speed value equal to 0. We empirically determined the threshold values. This process allows us to filter inaccurate GPS fixes, that otherwise would have led to an overestimate of travel distance.

Energy consumption. To evaluate energy consumption, we use the average power to estimate the impact on battery life. The latest mobile phones are typically loaded with a 1500mAh lithium-ion battery which has a 3.7V nominal voltage. Thus, a phone powered by a 1500mAh battery can last 24 hours if it uses 231.25mW (1500mAh · 3.7V / 24h) on average. We estimate the average power requirement for each algorithm by using the energy profile of HTC G1, which we experimentally obtained (Figure 5), and the active-time duration (combined with sampling rate) of each sensor.

4.3 Experiment Results

We start with briefly explaining our parameter settings used for the experiments, and the two state-of-the-art place learning algorithms, PlaceSense [14] and Kang *et al.* [13], we implemented to evaluate our place detection algorithm. Sensitivity analysis of our parameters is presented at the end of this section.

SensLoc uses a similarity threshold t_{sim} to determine entrance and departure. In our experiment, t_{sim}, which can range from 0 to 1, was set to 0.7 as it was empirically found to be most effective. Step size (r_{step}), which defines the range of response rate used to select representative beacons, was set to 0.2. Minimum threshold (r_{min}), which sets the lower bound of the threshold, was set to 0.5. We experimented with two different Wi-Fi scanning intervals, 10 and 30 seconds, and a window size of 30 and 60 seconds were used, respectively. The certainty value, which determines the number of scan windows that are used to detect entrance and departure, was set to three (and two when the scanning interval is 30 seconds), as suggested by others using similar windowing mechanisms. For movement detection, we duty-cycled accelerometer 50%, and a conservative variance threshold value (v_{mov}) 0.2 was used to find sleep opportunities. Paths were tracked with a 1/10 Hz GPS sampling rate.

PlaceSense, similar to ours, relies on radio beacons to find places. Absence of newly-seen beacons triggers entrance detection and disappearance of every representative beacon signals departure. For fair comparison, we matched the parameter settings with SensLoc when they have same interpretations. We used a Wi-Fi sampling interval of 10 seconds, a window size w of 30 seconds, and sliding windows

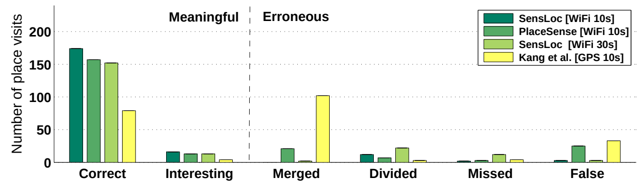


Figure 6. Number of place visits found from five people over a week following their normal life. SensLoc outperforms PlaceSense with more *correct* places, and fewer *merged* and *false* places.

were applied. Stable depth s and tolerance depth t is equivalent to our certainty value c so we also used three. A fixed representative beacon threshold was set to 0.9 as suggested.

Kang *et al.* designed a time and distance based clustering algorithm to find significant locations and is used widely by others using GPS trajectories to find significant locations [18, 26]. We used parameters for time t=5 minutes and d=200 meters. GPS coordinates collected every 10 seconds were used as input and GPS failures were regarded as a stay within the distance threshold. To improve its performance, GPS samples with low accuracy value (above 30m) were filtered.

4.3.1 5-person short-duration study

The goal of this study is to evaluate SensLoc over varied demography. Participants included an office worker, a housewife, a pharmacist, and two graduate students in different departments. Their age varied from mid-20s to mid-50s. The participants collected their traces over a week following their normal routines, and kept a diary of the actual places they visited with entrance and departure times. To provide anonymity, we assign the pseudonyms Adam, Beka, Charlie, Dana, and Evan.

Typical traces contained ordinary work and home routines during weekdays and visits to shopping areas for weekends. Adam drives to work and visits various buildings located in a walking distance during work hours. His office is in the 7th floor of a 8-story building. He has the longest commute. Beka also regularly drives to a private institute and spends 5 hours a day during weekdays, but does not have many places near it other than a classroom which is located on the 4th floor of a 4-story building. Her traces instead includes frequent visits to several grocery stores and a couple of restaurants. Charlie takes a bus to school and visits various lab rooms and class rooms during the week. On the weekend, he visits several stores. Dana walks to work in a dense urban area and has the shortest commute distance. She works at the first floor of a 2-story building located near tall buildings. She visits various lunch places near work and lives on the 9th floor of a 14-story building. Evan commutes by a bus and visits about three different places at work, and sometimes stops by stores before arriving at home. The weekend includes visits to superstores.

Place Detection. First, to evaluate our effectiveness in detecting place visits, we compare SensLoc against previous techniques using the metrics presented in Section 4.2. Figure 6 illustrates the place detection performance of different techniques on the 5-person 1-week data set. Overall, SensLoc outperformed other place detection algorithms with

	Adam				Beka				Charlie				Dana				Evan				All			
	SL10	SL30	PS	KA	SL10	SL30	PS	KA	SL10	SL30	PS	KA	SL10	SL30	PS	KA	SL10	SL30	PS	KA	SL10	SL30	PS	KA
Cor.	47	45	40	12	25	24	25	19	40	36	41	18	30	23	24	11	34	33	31	19	176	157	159	79
Int.	3	2	3	1	4	2	4	1	6	6	2	0	0	0	0	0	3	3	4	3	16	13	13	4
Mer.	0	0	6	37	0	0	0	5	0	0	2	24	0	2	6	20	0	0	5	16	0	2	19	102
Div.	2	1	2	0	1	2	1	2	2	4	0	1	3	4	1	0	2	3	1	0	10	17	7	3
Mis.	0	4	1	0	0	0	0	0	1	3	0	0	0	4	2	2	1	1	0	2	2	12	3	4
Fla.	1	2	5	10	0	0	3	0	1	1	9	7	1	0	3	2	0	0	5	14	3	3	25	33
Recall	0.96	0.90	0.82	0.24	0.96	0.92	0.96	0.73	0.93	0.84	0.95	0.42	0.91	0.70	0.73	0.33	0.92	0.89	0.84	0.51	0.94	0.84	0.85	0.42
Precision	0.94	0.94	0.77	0.22	0.97	0.93	0.88	0.74	0.94	0.89	0.80	0.36	0.88	0.79	0.71	0.33	0.95	0.92	0.76	0.42	0.94	0.89	0.77	0.38

Table 1. The distribution of discovered places by different users (SL10: SensLoc [Wi-Fi 10s], SL30: SensLoc [Wi-Fi 30s], PS: PlaceSense [Wi-Fi 10s], KA: Kang *et al.* [GPS 10s]). Overall, SensLoc outperformed others with the largest number of *correct* places and the smallest number of *false* places.

the largest number of *correct* places. PlaceSense merged places where more than one beacon is found strongly in both places, a known limitation [14]. For example, short trips (typically within a few minutes of walking), such as moving from one floor to another in the same building often shared strong beacons in both floors, and consequently were merged as a single place. In contrast, SensLoc's detection algorithm exploits changes in signal strengths, and was able to correctly distinguish many adjacent indoor places. SensLoc also significantly reduced false detections using its robust similarity measurement and improved filtering method. However, PlaceSense had relatively more *false* places. It recognized a slow walk through an open area or a hallway as a place when at least one strong beacon was found consistently during the walk as it only depends on the visibility of the beacon.

Kang *et al.*, based on GPS, resulted in significantly more *merged* places as many proximate places located in nearby buildings were identified as a single place. Intuitively, as GPS coordinates were not available indoors, adjacent places in closely located buildings could not be identified. Kang *et al.* also resulted in the largest number of *false* places. Similar to PlaceSense, slow walks in an open area or slow drives on a congested road were often identified as a place. Note, however, that it *divided* fewer places, by correctly finding more superstores or outdoor places where Wi-Fi APs were sparse. Large stores were sometimes not covered by strong Wi-Fi APs and Wi-Fi beacon-based techniques *divided* them into several different places (or *missed*). However, as they typically were located in a single-story building, high-quality GPS coordinates were available in these places. Thus, a hybrid approach adequately combining both techniques may improve the overall performance of the system.

It is interesting to note that decreasing SensLoc's Wi-Fi sampling rate did not significantly degrade the performance, although it resulted in more *missed* or *divided* places when beacon signals were sparse or weak. Increasing the sampling rate naturally improves performance, but comes with a cost of energy consumption. Our result suggests that SensLoc can provide accurate place detection with a 1/10 Hz Wi-Fi scanning rate, and can reduce its sampling rate to 1/30 Hz when the remaining energy-level of the device is low but still can provide a reasonable level of performance. However, even with a Wi-Fi sampling rate of 1/10 Hz, the overall energy consumption of SensLoc is significantly lower than others as we show in the next section.

Table 1 shows the distribution of erroneously discovered places by users. On average, SensLoc correctly recalled 94% of the visited places and 94% of the found places were actually visited. The improvement of SensLoc over PlaceSense was noticeable when users had many nearby places in their routine. For example, Adam and Evan frequently visited two office rooms in different floors at work which PlaceSense failed to detect. Dana's routine included frequent visits to nearby stores which shared strong beacons. However, when a user's daily routine did not include many nearby places, SensLoc performed similar to or slightly better than PlaceSense. For instance, Beka had a simple routine visiting only a single place in each building, and the performance difference was less significant. Overall, SensLoc detected fewer false places than PlaceSense, resulting in a consistently higher recall number for every participant.

Daily Avg.	Adam	Beka	Charlie	Dana	Evan
Distance (m)	28602.87	16679.40	18046.97	8951.18	13614.47
Coverage (%)	98.5	97.3	98.2	86.1	94.1

Table 2. The total distance traveled by each participant and percent of the travel distance covered by SensLoc. On average, it only used GPS for about two hours a day and yet covered above 95% of the distance.

Path Tracking. To evaluate how effectively SensLoc tracks user's travel paths, we consider two aspects of tracking: the percentage of the total travel distance covered by an on-demand path tracking and the quality of the collected position estimates.

First we investigate the travel distance recorded by SensLoc compared to when the device continuously tracked a user's location. We define coverage as the percentage of the distance tracked by our path tracker and the total travel distance tracked by continuous GPS tracking. Table 2 presents the daily average travel distance of each data collector and the coverage of SensLoc. The average travel distance was mostly influenced by a participant's commute distance. On average, SensLoc covered nearly 95% of the travel distance while the coverage slightly differed between data collectors.

The coverage varied from 86.1% to 98.5% depending on a user's daily travel pattern. A small delay before starting path tracking when a user departs from a place was the main cause of missing partial travel distances. As the absolute length of the lost distances were similar between users, participants with a longer average travel distance had a better coverage than others with a shorter average travel distance. For instance, Dana particularly traveled a shorter distance as she lives in a dense urban area and walks to work and had the lowest coverage. On the other hand, Adam with the longest commute distance received the highest coverage. Nonethe-

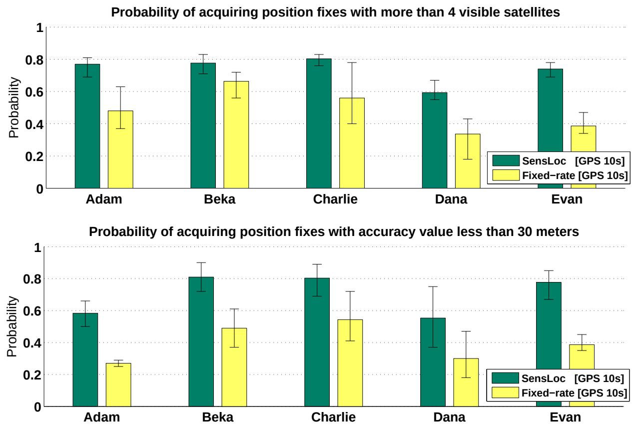


Figure 7. The overall quality of the collected GPS coordinates measured by the reported accuracy value and the number of visible satellites used.

less, the coverage was not significantly compromised for all data collectors (more than 95% for 4 participants, and 86.1% for one participant). Note that SensLoc, on average, only had to activate GPS for two hours a day (presented next), and yet could cover above 95% of the travel distance.

The overall quality of the collected position estimates are measured by the reported accuracy value and the number of visible satellites used for estimating the position. As long as a GPS receiver has stable signals from at least four satellites, the estimation error of the position fix is known to be within 15 meters [25]. First we investigated the average probability of high quality GPS position fixes among the collected ones. We compared the positions collected from SensLoc against the ones from periodic GPS sampling. Both used 1/10 Hz as their GPS sampling rate. Figure 7 demonstrates that we can carefully acquire GPS coordinates only when it is most likely to provide estimates with high accuracy by tracking GPS only when traveling between places.

Naturally, the places that participants spend most of their time largely affected these results. Adam and Evan's work place had no signals or severely inaccurate position estimates. Charlie's work place was located at the top floor and had positions with relatively better accuracy values. During the experiment, GPS usually had more visible satellites and provided fixes with good accuracy values when the floor is near the rooftop, but degraded significantly when multiple floors are above. Dana showed lower quality as she mostly traveled between high-rise buildings in a dense urban area. Results from Beka's data illustrate that the accuracy value can be worse than 30 meters even when the GPS used more than four visible satellites to estimate its position. Her classroom, which she frequented, had about four visible satellites on average, but the accuracy value of the position fixes was often worse than 30 meters and had a lot of jumps. Figure 8 illustrates the cumulative distribution of the quality of GPS position fixes from a single day, showing that we can suspend GPS tracking when it is unlikely to provide good estimates.

Energy Consumption. Finally, we analyze SensLoc's daily energy consumption by different users to understand the energy cost of continuously detecting place visits and tracking travel paths. To estimate the power consumption,

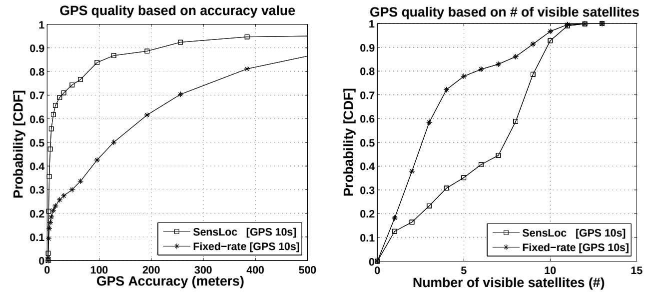


Figure 8. GPS Quality CDF from a single day. The fixed-rate scheme collected more position fixes from indoors, reducing the overall quality of the collected GPS position estimates. In contrast, SensLoc acquired position fixes mostly when a user travelled outdoors.

we logged the time each sensor was activated and used the average energy consumption of each sensor we attained from power measurements of the HTC G1 phone (Figure 5). Here, we present the case when SensLoc used a sampling interval of 10 seconds for both GPS and Wi-Fi. We also experimented with higher Wi-Fi sampling rates to see if it improves the place detection performance. However, more samples did not necessarily improve the performance by introducing more noises, while consuming significantly more energy (Figure 5). On the other hand, higher GPS sampling rates could improve path information (especially when traveling at high speeds). but does not directly affect SensLoc's performance and is a tunable parameter that responds to application needs and energy budgets. The duty cycle of the accelerometer was 50% (over a 10 seconds period) to detect movements (as explained in Section 4.3), and provided sufficient information for detecting movements to trigger Wi-Fi scans while saving substantial amount of energy.

Figure 9 presents the average time each sensor was active during a day's operation by each participant. The error bars show the maximum and minimum time each sensor was active for a day during the one week experiment. As sensors are powered on and off depending on the user's location and movement, on-times were largely dependent on the level of activity during a particular day. For instance, when a user visited many places or traveled long distances, the active time of Wi-Fi and GPS increased, while the active time of the accelerometer decreased. On the other hand, when a user mostly stayed at home, the accelerometer was on almost all the time, while others were barely used. The bars in Figure 9 also implicitly reveal the relative number of places each

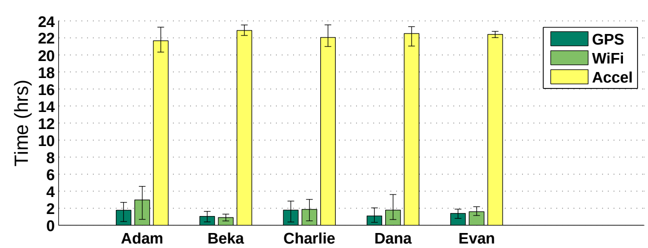


Figure 9. The average active time of each sensor used by SensLoc. For most the time, the accelerometer monitors the user's movement, while GPS and Wi-Fi are used less frequently.

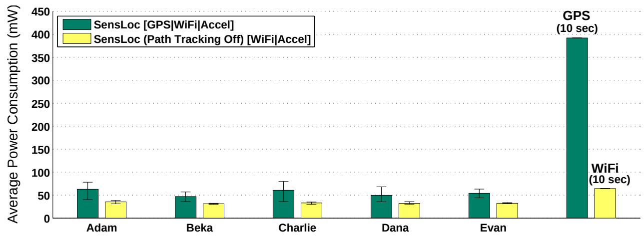


Figure 10. Average daily power consumption across five different users. The level of power consumption depended on the level of mobility.

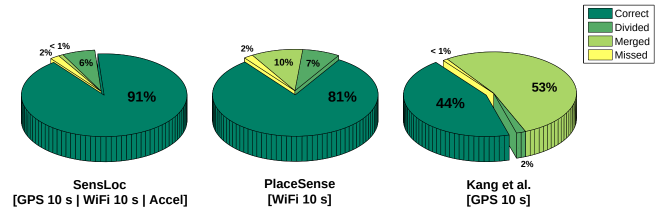


Figure 11. Percent correctly discovered new places from the four weeks real-life data set.

participant visited over the week experiment. Adam had the largest number of place visits, which resulted in longer Wi-Fi active times and shorter accelerometer active times, while Beka exhibited the opposite by having the smallest number of place visits. The average on-times of each sensor over the five participants were 1.4 hours for GPS, 1.8 hours for Wi-Fi, and 22.3 hours for accelerometer. This also agrees with the studies [16, 9] reporting that people spend nearly 90% of their time indoors.

Using the results from above, we further estimated the overall energy used by SensLoc (Figure 10). Additionally, we provide the estimated energy cost when path tracking is disabled and only place and movement detection is enabled. This eliminated the cost of tracking GPS positions as well as the savings achieved by turning off Wi-Fi when GPS reported that the user is traveling at high speeds. We added this to compare against PlaceSense which does not track paths nor uses an accelerometer to save Wi-Fi scans. PlaceSense detects places with periodic Wi-Fi scans and its energy cost can be derived from the average energy consumed by sampling Wi-Fi scans every 10 seconds. Savings were drastic. SensLoc without path tracking used 32.8mW on average which is nearly 50% of what PlaceSense used (64.38mW). The savings were from using a more energy-efficient duty cycled accelerometer instead of Wi-Fi scans when the device was immobile. When SensLoc also tracked paths by sampling GPS coordinates every 10 seconds only while traveling between places, it still used only 54.8mW on average which is about 87% less than the energy consumed by collecting GPS every 10 seconds (392.2mW).

4.3.2 2-person long-duration study

To better understand SensLoc's expected performance and energy cost over a long-term use, we use a data set collected by two people over four weeks. We first investigate how well SensLoc discovers new places and recognizes them when they are revisited. Then we discuss the paths that were found and the fluctuation in energy consumption over the 28 days of experiments. Before we start, we briefly explain our two data collectors who provided the data set.

Our data collectors are assigned the pseudonyms George and Harry. George is a parent of one child. Many of his places involve driving his kid to school, restaurants, and extracurricular activities. He has two different work places where he goes for half of a week each. His wife usually drops him off at work and he takes buses home. One of the two offices is located in a multiplex building where he frequently visits several office rooms separated by more than a minute's walking distance. Weekends involve visits to various stores and markets. He also drove to friends living in different cities a couple of times during the data collection. He visited 61 unique places and traveled 1003.08 km over 4 weeks.

Harry is single and drives to work but occasionally chooses to take buses. When he drives, he takes three different routes to commute depending on the time of the day to beat traffic. He visits several nearby buildings and outdoor patios during work hours for meetings and lunch. He also takes lessons during lunch time twice a week near work and frequently visits a coffee shop. Many of his places are near work or home. For weekends, he regularly visits outdoor courts for sports, restaurants, gas stations, and a couple of stores for groceries. He visited 49 unique places and traveled 568.28 km for 4 weeks.

Learning New Places. First we evaluate how well SensLoc discovered new places that our data collectors visited over a month as they went about their normal routines. Here, we compare SensLoc with previous place learning algorithms: PlaceSense and Kang *et al.* We labeled each visit with a unique place name which we learned from the diaries provided by our data collectors, and then found every first visit to a unique place. A total of 110 first visits to a unique place was used for this evaluation.

Figure 11 illustrates the percentage of correctly detected first visits for each algorithm. Both PlaceSense and SensLoc found new places well when the place was covered by at least one steady Wi-Fi AP. SensLoc correctly found 10% more first visits than PlaceSense by significantly reducing *merged* detections using its new place detection algorithm. SensLoc particularly performed better than PlaceSense in separating two adjacent places sharing similar beacons. Kang *et al.* logically failed to find many indoor places in closely located buildings, and correctly found only 44% of the places. While the two beacon-based techniques outperformed the GPS-based technique overall, in some cases, GPS-based approach worked better. For example, a couple of places including middle school gyms, outdoor parks, and particular superstores had sparse and weak Wi-Fi APs signals, and were challenging for beacon-based techniques to detect well, resulting in *divided* places. On the other hand, a GPS-based algorithm worked better in correctly detecting these places, having only 2% of *divided* places (compared to 5% of SensLoc and 7% of PlaceSense). Thus, to further improve the place coverage, a hybrid approach using both radio signals and positions to detect places may be considered.

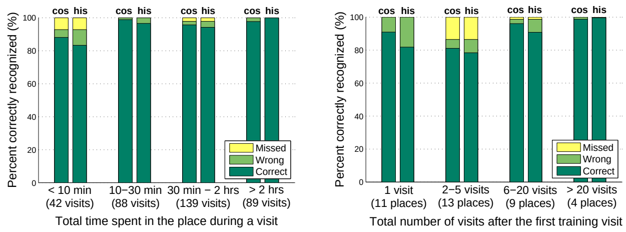


Figure 12. Percent correctly discovered and recognized from the four weeks real-life data set using two different recognition algorithms (cos: cosine-based recognition, his: histogram-based recognition).

Recognizing Revisited Places. To investigate the overall performance of recognizing revisits, we used the first correctly discovered visit for learning, and used every other visits for testing the recognition performance. We also compared our new cosine-based place recognition scheme against the previously proposed histogram-based scheme [11]. For this evaluation, we used 358 revisits to 37 unique places and excluded 73 places which only had one visit over the four weeks (43 places from George and 30 places from Harry). The percentage of correctly recognizing revisits is the ratio of the algorithm correctly recognizes the place to the total number of visits the data collector actually made. Errors are further broken down to *wrong* (recognized as a different place) and *missed*.

Figure 12 illustrates the recognition performance of SensLoc by the total time spent in the place during a visit and by the total number of visits after the first visit. Overall, 96% of the revisits were correctly recognized by using the cosine-based recognition scheme, while 94% was correctly recognized by the histogram-based scheme. The improvement was noticeable in correctly differentiating closely located places. Places that resulted in *wrong* recognition were recognized as another adjacent place sharing similar beacons for both algorithms. *Missed* places included 2-3 short visits to a room covered with no Wi-Fi APs and a couple of outdoor places which were visited 4-6 times during the data collection. The radio environment of a particular place affected our recognition performance more than the visit duration or the visit frequency.

Paths Connecting Places. SensLoc automatically traced and parsed 293 paths over the four weeks (excluding 150 indoor travels) from the two data collectors' daily routines. George traveled 148 times between 102 unique pairs of places, while Harry traveled 145 times between 90 unique pairs of places, creating 293 paths in total. Figure 13 shows the distribution of the paths by their travel distance, travel

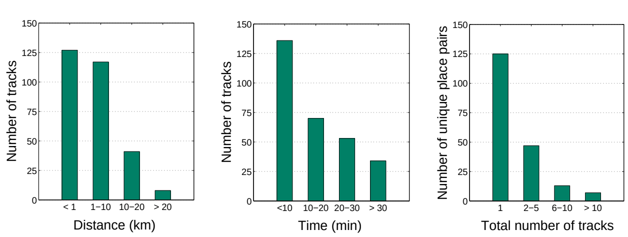


Figure 13. The distribution of recovered paths by their distance length, time duration, and unique pairs of places the path connected.

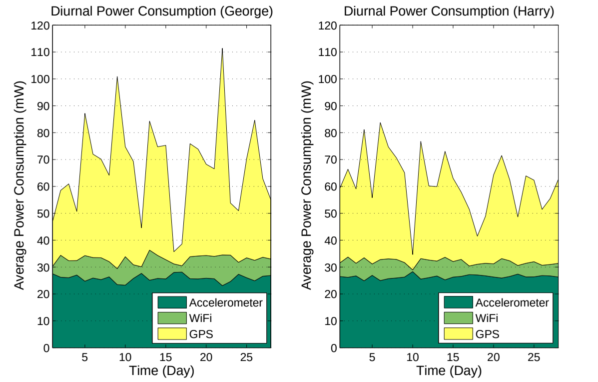


Figure 14. Stack plots of diurnal power consumption over the four weeks. The power consumption of GPS fluctuates substantially depending on the level of mobility. High-peaks are due to long-distance drives to different cities, and dips are from stay-at-home days.

time duration, and the number of tracks between a unique pair of places it connected. Most of the paths shorter than 1 km were travelled by walks and the longer ones were by driving cars or taking other transportation such as buses.

Unlike other tracking schemes, SensLoc additionally provided new context information for these paths. By attaching information about which places the path connects, SensLoc allows us to filter and query paths based on where the path started and ended. For example, if an application wants to find the shortest path or the fastest route between "home" and "my office", it can simply query every recorded path connecting the two places. As an experiment, we queried every path between "home" and "my office" from Harry's data. We retrieved 30 paths[2] which we could distribute to three different ranges of distances. There were six paths 6-7 km long; four paths 8-9 km long; 20 paths 10-11 km long. By simply looking at the travel distance, we could infer that Harry has at least three different routes he takes to beat the traffic depending on the time of the day.

Diurnal Energy Consumption Patterns. Finally, we investigate the daily power consumption of SensLoc over the four weeks. By looking at the changes in the energy consumption over multiple days (as illustrated in Figure 14), we can notice how the user's mobility over the day can significantly affect SensLoc's average power consumption for that particular day. While the average power consumption of Wi-Fi and accelerometer stayed around 6mW (2-4 hours a day) and 26mW (20-22 hours a day), the average power consumption of GPS fluctuated significantly from a minimum of 4.5mW (15 minutes a day) to a maximum of 76.8mW (4.7 hours a day). Note that Figure 14 is a stack plot that accumulates the average power consumption of the accelerometer, Wi-Fi, and GPS. The two peaks in George's data illustrates the two long-distance travels he made to the friends living in different cities. On the other hand, the days with noticeably low GPS usage illustrates that the data collector mostly stayed at home. Consequently, the accelerometer was used more than other days. However, even for the worst day,

[2]This also implied that he often stopped by other places in between as he went to work more than 15 times.

SensLoc averaged about 110mW[3]. The battery life can be significantly improved if we lower the GPS sampling rate as well (note that we collected GPS every 10 seconds).

4.3.3 Controlled Experiments

Our place detection algorithm dynamically adjusts the representative threshold (r_{rep}) and the window size (w) to adapt to the changing beacon environment. However, the algorithm still depends on a fixed similarity score threshold (t_{sim}) and a response rate step size (r_{step}). In this section, we investigate the sensitivity of the performance results to the choice of these fixed parameters to arrive at sweet spots for these parameter values. We use a scripted-tours data set as it provides more accurate ground truth over the real-life data set, but the results are consistent in both data sets. The scripted-tours included visits to many indoor places adjacent to each other, such as rooms in different floors and stores cluttered in a commercial area near campus. We first examine the similarity score threshold (t_{sim}) and the response rate step size (r_{step}) by varying these parameters while fixing the rest of the parameters to values which resulted in the best performance. Then, we illustrate how changing the window size (w) impacts the performance. Finally, the performance of our final version is compared against other algorithms including PlaceSense and Kang *et al.* with different GPS and Wi-Fi sampling rates. We also illustrate the visit-time boundary accuracy of different techniques and sampling rates.

t_{sim}	Cor.	Mis.	Fal.	Mer.	Div.	Precision	Recall
0.9	43	7	0	0	0	0.86	0.86
0.8	46	4	0	0	0	0.92	0.92
0.7	48	2	0	0	0	0.96	0.96
0.6	43	2	1	4	1	0.84	0.86
0.5	43	1	2	6	0	0.83	0.86

Table 3. The distribution of errors with different similarity threshold. Larger threshold value t_{sim} results in fewer *merged* and *false* places, but more *missed* places.

As shown in Table 3, a smaller Tanimoto similarity threshold t_{sim} resulted in fewer *missed* places and more *merged* places as it becomes conservative in declaring a departure. A larger t_{sim} increases the level of similarity it expects to determine if the signal fingerprints are from a single place. It also increases the chances of missing places by falsely inferring stays as travels. All of the *missed* places included commercial stores where data collectors had more movement within a place or received weak beacon signals. A smaller t_{sim} allows more fingerprints with lower similarity to be regarded as from a single place, and increases the number of *merged* places. Thus, a moderate value such as 0.7 is preferable for reducing *missed* and *merged* places.

When measuring the similarity of two fingerprints for detecting an entrance, every beacon found in both fingerprints is used. However, to avoid intermittent beacons falsely triggering a departure, a subset of strong beacons are used to evaluate the similarity between two fingerprints when detecting a departure. SensLoc selects strong beacons based on their response rate, and the threshold (r_{rep}) is adaptively defined by observing the maximum response rate and subtracting a step size r_{step}. Thus, a lager r_{step} includes more beacons in the representative beacon set and becomes more sensitive to changes. Table 4 shows that a larger r_{step} results in more *false* places. In contrast, a smaller r_{step} uses only the strong beacons to evaluate the similarity and increases the number of *merged* places. Again, a middle ground value such as 0.2 led to an overall better performance.

r_{step}	Cor.	Mis.	Fal.	Mer.	Div.	Precision	Recall
0	39	1	1	10	0	0.76	0.78
0.1	46	2	0	2	0	0.92	0.92
0.2	48	2	0	0	0	0.96	0.96
0.3	48	2	0	0	0	0.96	0.96
0.4	47	3	1	0	0	0.92	0.94

Table 4. The distribution of errors with different representative beacon threshold. Larger r_{step} includes more beacons in representative set.

w	Cor.	Mis.	Fal.	Mer.	Div.	Precision	Recall
20	45	5	0	0	0	0.90	0.90
30	48	2	0	0	0	0.96	0.96
40	49	1	0	0	0	0.98	0.98
50	46	2	1	2	0	0.80	0.92
60	46	0	2	4	0	0.88	0.92
adaptive	50	0	0	0	0	1.00	1.00

Table 5. The distribution of errors with different window size. Larger w results in more *merged* places, but fewer *missed* places.

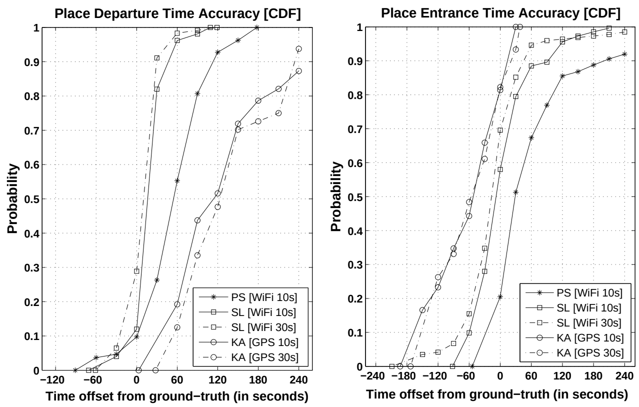
A window size w of a scan window combined with the sampling rate defines the number of samples included in each scan window. More samples in a scan window allows us to detect places where beacons are sparse and weak, but it becomes less sensitive to changes. Table 5 shows that using a larger w reduces *missed* places by additionally detecting places with low-response-rate beacons, but also increases *merged* and *false* places. Thus, we take an adaptive approach by increasing w when the neighboring beacons are sparse. We infer this when the adapted representative threshold is below r_{min}, which is set to 0.5 for this experiment. This adaptive approach eliminated the *missed* places and also suppressed merging places where beacons are dense.

Algorithm	Cor.	Mis.	Fal.	Mer.	Div.	Precision	Recall
SL (WiFi 10s)	50	0	0	0	0	1.00	1.00
SL (WiFi 30s)	46	4	0	0	0	0.92	0.92
PS (WiFi 10s)	34	5	3	11	0	0.64	0.68
PS (WiFi 30s)	34	0	5	16	0	0.61	0.68
KA (GPS 10s)	10	4	2	38	0	0.19	0.20
KA (GPS 30s)	9	7	3	35	0	0.18	0.18

Table 6. The distribution of errors by different algorithms with varying sample rate (SL: SensLoc, PS: PlaceSense, KA: Kang *et al.*)

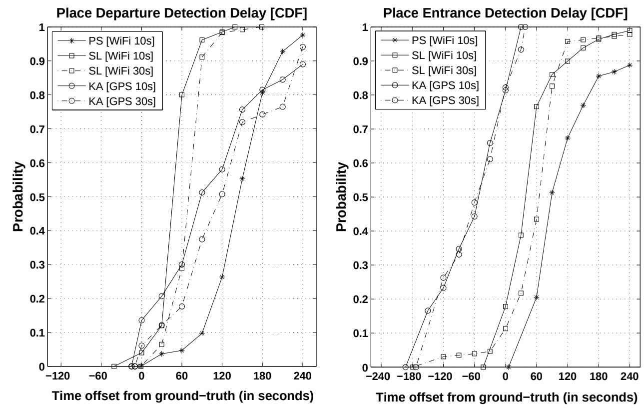
To summarize, we use the settings that lead to the best place detection performance and compare each algorithm. For SensLoc, our results suggest that a moderate value works best for the two fixed parameters, and adjusting other parameters based on the changing radio environment further improves performance results. Table 6 compares the performance of different place detection algorithms on traces from scripted tours. The parameter settings were same as described in Section 4.3. PlaceSense *missed* places where

[3] A 1500mAh battery can last 24 hours when, on average, 231.25 mW is used.



(a) Departure (b) Entrance

Figure 15. Time boundary accuracy. For 80% of the visits detected by SensLoc, time offset of departure times are within 0 to 30 seconds, and entrance times are within -60 to 60 seconds.



(a) Departure (b) Entrance

Figure 16. Detection Delay. For 80% of the visits detected by SensLoc, detection delay of departure times are within 30 to 60 seconds, and entrance times are within -30 to 120 seconds.

beacons were weak and *merged* places where places shared similar beacons. In contrast, SensLoc was better in detecting adjacent places sharing more than one strong beacon. Naturally, Kang *et al.* based on GPS resulted in more *merged* places, but worked reasonably well in finding building-level significant locations, unless the buildings are closely located. Reducing the sampling rate to 1/30 Hz from 1/10 Hz generally degraded the performance for all algorithms, but not significantly. The certainty value (c_{max}) combined with the sampling rate determines the minimum time delay before declaring entries and exits. We empirically found that $c_{max} = 3$ for 1/10 Hz and $c_{max} = 2$ for 1/30 Hz work best.

Finally, we evaluate the time-boundary accuracy of the detected places by their entrance and departure time. We first compare the accuracy of different algorithms, and then discuss how the sampling rate affects the accuracy. To measure the time-boundary accuracy of places found by algorithms, we measure the time offset of entrance and departure times from the ground truth (logged in diaries). We exclude *missed* places and use only the beginnings and ends that matched with the ground truth for *divided* and *merged* places. We discuss the departure time first as it may affect a subsequently visited place's entrance time. Many of the places visited during the scripted tours were within a couple of minutes walking distance and could affect each other's time boundaries.

As shown in Figure 15(a), the time offset of departure time for SensLoc was within 0-30 seconds for about 80% of the visits. For 10% of the visits, the inferred departure time was 0-60 seconds earlier than the actual departure time, and 10% had 30-120 seconds offset. PlaceSense generally had more delays in the found departure times as it has to lose every representative beacon before it declares a departure. About 80% of the visits had 0-120 seconds offset in their detected departure time. Kang *et al.* had even larger offsets as the places it found were generally coarser than places found by beacon-based algorithms. Figure 15(b) illustrates the accuracy of the found entrance times. The time offset of entrance times for SensLoc was within -60 to 60 seconds for about 80% of the visits, while PlaceSense's entrance times were relatively delayed, possibly due to the delayed departure time of the previously visited place. Decreasing the sampling rate had insignificant impact on the accuracy of the inferred entrance and departure time for all algorithms.

Lastly, we investigate the actual time it takes before the algorithm can declare an entrance or a departure. As place learning algorithms use several scan windows before declaring an entrance or a departure, the detection time and the inferred time-boundary differs. We define the time it takes to detect events as detection delay (offset from ground truth) and illustrate each algorithm's detection delay in Figure 16. SensLoc resulted in lower detection delay for both entrance and departure compared to others. The departure detection delay for SensLoc was 30-60 seconds for 80% of the visits when using a sampling rate of 10 seconds, while PlaceSense exhibited 90-210 seconds delays. Similarly, SensLoc detected entrance within 120 seconds for 90% of the visits while PlaceSense took the same amount of time for about 70% of the visits. By using a robust similarity measurement method to detect places, SensLoc improves the time-boundary accuracy and detection delay as well.

5 Related Work

Semantic Place Learning. Place learning algorithms attempt to find meaningful places from raw sensor data. We can broadly classify them into two categories: geometry-based and fingerprint-based approaches.

Geometry-based algorithms identify places as a set of coordinates within circles or polygons. These algorithms use periodically collected position estimates to detect an individual's stay in a certain region and infer significant places. Essentially, the achievable granularity (*e.g.*, room-level or building-level) depends on the underneath positioning system they rely on. For example, Marmasse *et al.* defines a place as a Euclidian ball with a fixed radius where GPS is unavailable [23]. Ashbrook *et al.* and Toyama *et al.* proposed using a variant of the k-means clustering algorithm to tune the clusters found by GPS signal losses [2, 31]. Liao *et al.* iteratively infer activities and significant places from

GPS traces using a hierarchical CRF model, and the model is trained by fitting parameters using a labeled trace [19]. Nurmi *et al.* proposed a Dirichlet process clustering algorithm that does not require parameter fitting, but may take hundreds of thousands of iterations to converge depending on the trace [24]. Kang *et al.* [13] improved an approach proposed by Hariharan *et al.* [10] that finds places by defining temporal and spatial stay thresholds without depending on GPS disappearances. Unlike others requiring the entire trace for offline segmentation and inference, their heuristic algorithm based on distance and time can be used in real-time and is computationally less expensive. These approaches are found to be fairly effective at discovering building level places or outdoor places, but suffers from differentiating indoor places at dense urban area besides the energy consumed by continuously estimating positions.

Ambient fingerprints have been successfully used for detecting semantic places with finer granularity than that of the geometry-based approaches. This includes RF fingerprints (*e.g.*, Wi-Fi, Bluetooth, and cell tower), surrounding color, texture, and sound pattern. Among these, RF fingerprints have been most popular in discerning subtle differences between semantically different places. The key benefit over other ambient signatures is that the RF beacons can be monitored regardless of placement of a mobile device. For example, currently connected cell towers were used to learn and recognize places [17, 7, 33] albeit with coarse granularity, and at the cost of complex implementation [29]. BeaconPrint rely on new beacons to infer place visits [11]. PlaceSense further improved its place discovery accuracy by using separate mechanisms to detect entrances and departures [14]. These algorithms are compatible with cell towers, but Wi-Fi APs provided more robust and finer grained information about semantic places. Other ambient signatures such as sound, light, color, and texture information can be used to further discern subtle differences between adjacent locations. SurroundSense uses ambient sound, light, color, user motion in a place in addition to RF signals [3]. Besides its ability to discriminate adjacent places that share similar radio bacons, it can also cluster semantically closer places (*e.g.*, remotely located franchised stores sharing similar looks). SoundSense uses acoustic signatures to recognize activities and places [21]. However, these approaches are not appropriate for detecting visit boundaries and require careful placement of a mobile phone such that the sensors can measure these signatures unobstructed.

Energy-efficient Path Tracking. To make continuous localization practical, several research efforts have dealt with energy-efficient location tracking focused on preserving the distance-error bounds requested by applications. Three recurring methods are 1) intermixing a set of positioning systems with varying accuracy and energy requirements, 2) predicting mobility to schedule the next location estimate, and 3) using low power sensors to find sleep opportunities.

EnLoc switches between localization techniques by finding the optimal localization accuracy for a given energy budget using dynamic programing [6]. It uses human mobility patterns to further improve its performance by predicting user mobility rather than using the last known location between consecutive location readings. A-Loc is based on a selection algorithm that determines the most energy-efficient localization technique to meet the accuracy requirement (which is also assumed to change as a user moves to different areas) [20]. It predicts future user location using a model based on HMM, updates the location and sensor error models, and selects the sensor with minimum energy use. EnTracked focuses on outdoor pedestrian tracking and assumes that applications specify their distance-error limits [15]. It detects movements using an accelerometer to turn off GPS, and uses speed estimates provided by GPS to predict movement and schedule the next location sample. RAPS uses a collection of techniques to adaptively sample GPS coordinates [25]. It duty cycles an accelerometer to detect movement, uses space-time history of user movements to predict mobility, and checks a GPS-available probability table based on the surrounding cell towers. It also allows users to share positions with neighboring users through Bluetooth. Zhuang *et al.* also uses an accelerometer to detect movements, schedules two different localization techniques, and adjusts sampling rates based on the battery-level [34]. EEMSS employs low power sensors to detect user states and context, and triggers high power sensors only when necessary [32]. While doing this, they duty cycle each sensor to further save energy. More recently, Constandache *et al.* combine map information and dead reckoning based on low power sensors to reduce GPS samplings [5]. These techniques can be used to track paths for SensLoc, although they may be needed for about 10% of the time on average.

6 Conclusion and Future Work

Our results show that SensLoc can both semantically and energy-efficiently provide location context to applications by using a combination of acceleration, Wi-Fi, and GPS sensors to find semantic places, detect user movements, and track travel paths. Place visits and path travels are inferred from raw sensor data, which is energy-efficiently achieved by leveraging our tendency to spend about 90% of the time indoors and 10% in a vehicle or at outdoors. Precision and recall of detecting semantic places are both improved compared to the previous state-of-the-art PlaceSense approach by additionally exploiting signal strength changes of the surrounding beacons and adapting parameters to the neighboring beacon density. The accuracy gains are particularly noticeable when a user's routine includes back-to-back visits to nearby indoor places (*e.g.*, rooms on different floors) that shares even a single strong beacon. SensLoc's enhanced place detection algorithm also improves the detected place entrance and departure times by over 2.3 times the precision of previous approaches. However, at some places where beacon signals are weak and unstable, PlaceSense, which only considers the presence of beacons, detects places more robustly. Path tracking is only initiated when a user is traveling between places, which allows us to achieve highly efficient duty cycling of positioning systems (*e.g.*, GPS 8.3% active time), and still covers 95% of the travel distance. This not only saves energy but also boosts the overall quality of the collected position estimates. Lastly, the average power consumption of SensLoc is about 54.8 mW, which is 6.2 times less than that of collecting GPS periodically. On average, accelerometer, Wi-Fi, and GPS are activated for about 20-22, 2-4, and 1-2 hours everyday, respectively.

We believe SensLoc has solved some of the major practicality issues with continuous location tracking, and illustrated that an approach with a holistic and semantic point of view may provide a realistic solution for many applications. Our results also suggest that there is still more room for improvement to push the place detection performance even further. Adaptive approaches intermixing several place learning techniques based on the radio environment and the application needs may allow us to cover the remaining 5% places that are challenging. Using more energy-efficient sensors may also reduce the energy cost. For example, cell tower information, which almost comes for free, can replace Wi-Fi scans, if mobile service providers become less reluctant in disclosing cell tower information and more platforms provide common APIs to scan every neighboring cell towers. However, we think most research should focus on developing an application stack with a well-defined set of APIs, and create a feedback loop with the users that could tell us what is really important to address. The outcome of these field studies will expose application demands and provide nuances to tune the system for particular uses or situations.

7 References

[1] G. Ananthanarayanan, M. Haridasan, I. Mohomed, D. Terry, and C. A. Thekkath. Startrack: a framework for enabling track-based applications. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 207–220, New York, NY, USA, 2009. ACM.

[2] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Comput.*, 7(5):275–286, 2003.

[3] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272, New York, NY, USA, 2009. ACM.

[4] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *MobiSys '05*, pages 233–245, New York, NY, USA, 2005. ACM.

[5] I. Constandache, R. R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *INFOCOM'10: Proceedings of the 29th conference on Information communications*, pages 2321–2329, Piscataway, NJ, USA, 2010. IEEE Press.

[6] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. P. Cox. Enloc: Energy-efficient localization for mobile phones. In *INFOCOM*, pages 2716–2720. IEEE, 2009.

[7] J. Froehlich, M. Y. Chen, I. E. Smith, and F. Potter. Voting with your feet: An investigative study of the relationship between place visit behavior and preference. In *Ubicomp '06*, pages 333–350, 2006.

[8] 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 *MobiSys '08*, pages 174–186. ACM, 2008.

[9] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[10] R. Hariharan and K. Toyama. Project lachesis: Parsing and modeling location histories. In M. J. Egenhofer, C. Freksa, and H. J. Miller, editors, *GIScience*, volume 3234, pages 106–124. Springer, 2004.

[11] J. Hightower, S. Consolvo, A. LaMarca, I. E. Smith, and J. Hughes. Learning and recognizing the places we go. In *Ubicomp '05*, pages 159–176, 2005.

[12] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[13] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. In *WMASH '04*, pages 110–118, New York, NY, USA, 2004. ACM.

[14] D. H. Kim, J. Hightower, R. Govindan, and D. Estrin. Discovering semantically meaningful places from pervasive rf-beacons. In *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*, pages 21–30, New York, NY, USA, 2009. ACM.

[15] M. B. Kjaergaard, J. Langdal, T. Godskand, and T. Toftkjaer. Entracked: energy-efficient robust position tracking for mobile devices. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 221–234, New York, NY, USA, 2009. ACM.

[16] N. E. Klepeis, W. C. N. WC, W. R. O. WR, and et al. The national human activity pattern survey (nhaps): A resource for assessing exposure to environmental pollutants. *Journal of Exposure Analysis and Environmental Epidemiology*, 11(3):231–252, 2001.

[17] K. Laasonen, M. Raento, and H. Toivonen. Adaptive on-device location recognition. In *Pervasive '04*, pages 287–304, 2004.

[18] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *GIS*, page 34, 2008.

[19] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *Int. J. Rob. Res.*, 26(1):119–134, 2007.

[20] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy aware localization for mobile devices. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, 2010.

[21] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 165–178, New York, NY, USA, 2009. ACM.

[22] P. J. Ludford, D. Frankowski, K. Reily, K. Wilms, and L. Terveen. Because i carry my cell phone anyway: functional location-based reminder applications. In *CHI '06*, pages 889–898. ACM, 2006.

[23] N. Marmasse and C. Schmandt. Location-aware information delivery with commotion. In *HUC '00*, pages 157–171. Springer-Verlag, 2000.

[24] P. Nurmi and S. Bhattacharya. Identifying meaningful places: The non-parametric way. In J. Indulska, D. J. Patterson, T. Rodden, and M. Ott, editors, *Pervasive*, volume 5013 of *Lecture Notes in Computer Science*, pages 111–127. Springer, 2008.

[25] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, June 2010.

[26] S. Reddy, K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using context annotated mobility profiles to recruit data collectors in participatory sensing. In *LoCA '09: Proceedings of the 4th International Symposium on Location and Context Awareness*, pages 52–69, Berlin, Heidelberg, 2009. Springer-Verlag.

[27] T. Sohn, K. A. Li, G. Lee, I. E. Smith, J. Scott, and W. G. Griswold. Place-its: A study of location-based reminders on mobile phones. In M. Beigl, S. S. Intille, J. Rekimoto, and H. Tokuda, editors, *Ubicomp '05*, volume 3660 of *Lecture Notes in Computer Science*, pages 232–250. Springer, 2005.

[28] C. Song, Z. Qu, N. Blumm, and A.-L. Barabsi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, Feb. 2010.

[29] N. Song. Discovering user context with mobile devices: location and time. Thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science., 2006.

[30] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 85–98, New York, NY, USA, 2009. ACM.

[31] N. Toyama, T. Ota, F. Kato, Y. Toyota, T. Hattori, and T. Hagino. Exploiting multiple radii to learn significant locations. In *LoCA '05*, pages 157–168, 2005.

[32] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 179–192, New York, NY, USA, 2009. ACM.

[33] G. Yang. Discovering significant places from mobile phones - a mass market solution. In R. Fuller and X. D. Koutsoukos, editors, *MELT*, volume 5801 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2009.

[34] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. In *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 315–330, New York, NY, USA, 2010. ACM.