ARIEL: Automatic Wi-Fi based Room Fingerprinting for Indoor Localization

Yifei Jiang[†], **Xin Pan**[†], **Kun Li**[◇] **Qin Lv**[†], **Robert P. Dick**[§], **Michael Hannigan**[*], **Li Shang**[◇]

[†] Dept. of CS, [◇]Dept. of ECEE, [*]Dept. of ME, University of Colorado Boulder, CO 80309 U.S.A.
[§] EECS Department, University of Michigan, Ann Arbor, MI 48109 U.S.A.
[† ◇ *]{yifei.jiang, xin.pan, qin.lv, li.shang, hannigan}@colorado.edu, [§]dickrp@eecs.umich.edu

ABSTRACT

People spend the majority of their time indoors, and human indoor activities are strongly correlated with the rooms they are in. Room localization, which identifies the room a person or mobile phone is in, provides a powerful tool for characterizing human indoor activities and helping address challenges in public health, productivity, building management, etc. Existing room localization methods, however, require labor-intensive manual annotation of individual rooms.

We present ARIEL, a room localization system that automatically learns room fingerprints based on occupants' indoor movements. ARIEL consists of (1) a zone-based clustering algorithm that accurately identifies in-room occupancy "hotspot(s)" using Wi-Fi signatures; (2) a motion-based clustering algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and (3) an energy-efficient motion detection algorithm to minimize the noise of Wi-Fi signatures. ARIEL has been implemented and deployed for real-world testing with 21 users over a 10-month period. Our studies show that it supports room localization with higher than 95% accuracy without requiring labor-intensive manual annotation.

1. INTRODUCTION

People spend approximately 90% of their time in buildings [21]. Understanding the indoor activities and environments of occupants will offer valuable insights to a wide range of pressing challenges, such as public health, employee productivity, building security and energy management. For instance, the economic consequences of poor indoor environments in U.S. commercial buildings were estimated at $40–160 billion per year in lost wages and productivity, administrative expenses, and health care costs [8].

A key observation of this work is that indoor environments are well structured – Buildings are organized into rooms with distinct functionalities, supporting and also defining the activities, interests, and social roles of their occupants. For instance, an office building is typically organized into offices, conference rooms, and lounges; a shopping center consists of a large number of shops selling different products; and a residential home is comprised of a living room, bedrooms, a kitchen, etc. Therefore, room localization, i.e., identifying which room an occupant is in, will help characterize the indoor activities of occupants, thereby providing valuable insights to address health, social, and economic related issues of indoor environments.

Recently, new applications and services that leverage indoor room information have started to emerge [9, 11, 15, 19]. Room fingerprinting, which measures and assigns unique RFS (radio-frequency signals, e.g., Wi-Fi) to rooms, is the de facto technique for room localization. The time-consuming manual annotation process is a key limitation of existing room fingerprinting techniques. They either rely on an expert surveyor [11, 15], or leverage a collaborative but less accurate effort [17] among volunteers [9, 19], to manually collect Wi-Fi fingerprints for each room and associate them with predefined room IDs.

This article describes ARIEL, an automatic room localization system using Wi-Fi based room fingerprint analysis based on personal mobile phones carried by occupants. Designing an indoor room localization system that learns room fingerprints without manual annotation is challenging. First, due to signal reflection, refraction, diffraction, and absorption, indoor Wi-Fi signals are noisy. Such noise obscures the unique relationship between Wi-Fi fingerprints and individual rooms. Second, occupant-specific indoor activities directly affect the room fingerprinting process. For instance, in-room occupancy "hotspot(s)" are unevenly distributed both spatially and temporally. With such distribution, clustering algorithms may learn multiple fingerprints for a room. Third, as a collaborative voluntary effort, the overhead, e.g., energy consumption, imposed on personal mobile phones must be low. In response to these challenges, we have developed the following algorithms:

UbiComp '12, Sep 5-Sep 8, 2012, Pittsburgh, USA.

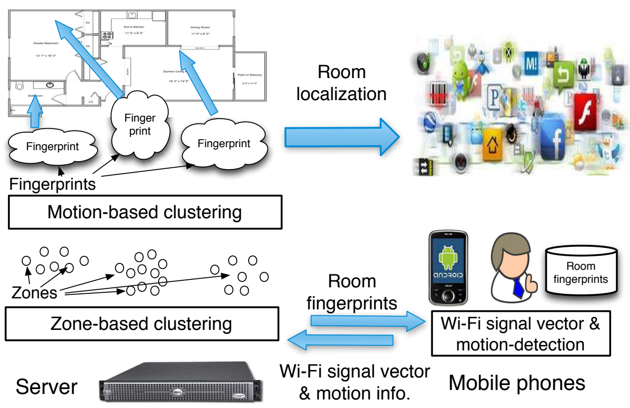

Figure 1. ARIEL system architecture.

- A zone-based clustering algorithm that accurately and automatically identifies in-room occupancy hotspot(s) using Wi-Fi signature;
- A motion-based clustering algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and
- An energy-efficient motion detection algorithm to minimize the noise of Wi-Fi signature.

ARIEL has been implemented and deployed. In a 10-month user study with 21 volunteers, ARIEL demonstrates excellent accuracy for room localization over a wide range of building environments, and offers comparable accuracy (95%) to existing supervised learning techniques requiring time-consuming manual annotation.

The rest of this paper is organized as follows. Section 2 summarizes the ARIEL system architecture. Sections 3 and 4 present the zone-based and motion-based clustering algorithms. Section 5 discusses the energy-efficient motion detection algorithm. Section 6 evaluates the proposed techniques. Section 7 surveys related work. Finally, Section 8 concludes this work.

2. PROBLEM FORMULATION AND SYSTEM OVERVIEW

This section formulates the Wi-Fi based room fingerprinting problem for indoor localization and summarizes ARIEL system architecture.

2.1 Room Fingerprinting and Localization

Room fingerprinting: Given a building with a set of rooms R and K Wi-Fi stationary access points (APs), Wi-Fi based room fingerprinting is to assign, to each room $r \in R$, a unique Wi-Fi signature consisting of m Wi-Fi signal vectors, $\mathbb{V}_r = \{V_{rj} | j \in \mathbb{Z} \ and \ 1 \leq j \leq m\}$, and

$$V_{rj} = \{ap_1 : rss_1, ..., ap_k : rss_k\}, \tag{1}$$

where each access point ap is identified by its MAC address, rss is the received signal strength (RSS) value of ap observed by a mobile phone. Note that k ($k \leq K$), the number of access points observed, could vary for each Wi-Fi scan, and m ($m \geq 1$), the number of Wi-Fi signal vectors, also varies by room. Automatically learning the Wi-Fi signature for each room is the key challenge addressed in this work.

Room localization: Given a run-time h-dimensional ($h \leq K$) Wi-Fi signal vector V_o observed by a user's mobile phone, indoor room localization determines the room that the user is in by measuring the similarity between V_o and each Wi-Fi signal vector $V_{rj} \in \mathbb{V}_r$ in each room r's fingerprint. We adopt the n-gram augmented Bayesian room localization method, which can achieve high accuracy, even when data are gathered using heterogeneous mobile phones [12]. Let Q_o be the sequence of access points sorted in descending RSS values:

$$\begin{aligned} Q_o &= seq(V_o) = (ap_{q_1}, \cdots, ap_{q_h}) \quad \text{with} \\ & rss_{q_i} \geq rss_{q_j} \quad (1 \leq q_i < q_j \leq h). \end{aligned} \tag{2}$$

An n-gram of Q_o is defined as a subsequence of length n extracted from the sequence Q_o at position i ($1 \leq i \leq h - n + 1$:

$$ngram_i(Q_o) = (ap_{q_i}, ap_{q_i+1}, \cdots, ap_{q_i+n-1}). \tag{3}$$

In the set of all rooms R, the most likely room $\hat{r}$ of occupation is determined as follows:

$$\hat{r} = \arg\max_{r \in R} \left[\prod_i P(ngram_i(Q_o)|r)P(r) \right]. \tag{4}$$

In other words, $\hat{r}$ is the room with the highest probability of producing the same ordering of access points as in subsequence $ngram_i(Q_o)$. Note that each V_{rj} ($1 \leq j \leq m$) in room r is also converted to an n-gram subsequence based on Equation 2 and Equation 3. The signature of room r, $\mathbb{V}_r = \{V_{rj} | j \in \mathbb{Z} \ and \ 1 \leq j \leq m\}$, is therefore converted to

$$\begin{aligned} \{ngram_i(seq(V_{rj})) \quad | \quad & i, j \in \mathbb{Z}, 1 \leq j \leq m \\ & and \ 1 \leq i \leq |seq(V_{rj})| - n + 1\}. \end{aligned}$$

$P(ngram_i(Q_o)|r)$ is the probability of $ngram_i(Q_o)$ appearing in the converted signature of room r.

2.2 System Overview

ARIEL supports automatic indoor room fingerprinting and room localization using collaborative Wi-Fi signature analysis based on personal mobile phones carried by occupants. Figure 1 illustrates the overall system architecture, which is comprised of components on the mobile phone side and server side.

On the mobile phone side, ARIEL performs the following operations.

- The run-time Wi-Fi signal vectors observed by each mobile phone are collected and delivered to the server to support room fingerprinting and room localization.
- The Wi-Fi signal vector stream is further annotated with motion data from build-in accelerometer, i.e., either collected when the occupant is in motion or stationary. Such information is also delivered to the server

to improve Wi-Fi fingerprint identification of in-room occupancy hotspot(s) and inter-hotspot correlation.

- Each mobile phone also maintains a local database storing the fingerprints[1] of the rooms that the user has visited before, which serves as a local cache, enabling run-time on-device room localization without engaging the server.
- A system software module provides room localization APIs to support high-level applications & services.

On the server side, ARIEL performs room fingerprinting and localization through an incremental process.

- Given the streams of Wi-Fi signal vectors and the corresponding motion information collected from mobile phones, ARIEL uses the zone-based clustering algorithm to incrementally identify in-room occupancy hotspot(s), or zone(s). Meanwhile, inter-zone correlations are identified by the motion-based clustering algorithm, then zones belonging to the same room are merged into a new cluster. Each cluster is assigned a room ID and the Wi-Fi signal vectors in the cluster form the room fingerprint.
- Using the n-gram augmented Bayesian room localization method, run-time room localization services are then offered to the occupants. A room fingerprint database maintains room IDs, room fingerprints, and the converted room fingerprints (n-gram AP subsequences and corresponding probabilities). The converted room fingerprints are selectively synchronized to each user's mobile phone based on the user's room visit history and predicted room visits in the future.

3. ZONE-BASED CLUSTERING

Our first step leverages Wi-Fi signals collected in stationary sessions (i.e., when a user is stationary in a room). We propose a zone-based clustering algorithm to accurately identify within-room stationary occupancy hotspot(s), i.e., zone(s). Each zone is then identified by a unique Wi-Fi signature, consisting of a set of Wi-Fi signal vectors which are typical for that zone.

Stationary occupancy hotspot (zone) identification is critical for accurate room fingerprinting and localization due to the following reasons. Within a room, human activities are highly nonuniform. One or multiple stationary occupancy hotspots typically exist, such as the couch area in a living room, cashier desk in a store, and desks in an office. Leveraging the mobile phones carried by occupants, more Wi-Fi signal samples are naturally collected from these zones, offering more robust fingerprints for a room. Vice versa, since occupants spend more time around these zones, their fingerprints usually better match the Wi-Fi signal vectors reported by occupants at run-time, improving accuracy.

[1]These are the converted room fingerprints, i.e., n-gram subsequences of APs and corresponding probabilities of seeing each of them in a room.

The zone-based clustering algorithm uses the Wi-Fi signal vectors reported by the mobile phones when the occupants are stationary, e.g., sitting or standing (detected by our motion detection algorithm described in Section 5.1). We further define the set of Wi-Fi signal vectors collected during a stationary period of an occupant as a *Wi-Fi session*. Next, giving a large collection of Wi-Fi sessions reported from multiple occupants, the zone-based clustering algorithm aims to determine the distances between the collected Wi-Fi sessions and partition these Wi-Fi sessions into one or more clusters, each corresponding to one of the stationary occupancy hotspots (zones).

3.1 Distance of Wi-Fi Sessions

The primary challenge of automatically identifying indoor stationary occupancy hotspots comes from the high noise of the collected Wi-Fi signal vectors, which is mainly due to i) high noise of indoor Wi-Fi signals from reflection, diffraction, and absorption; and ii) heterogeneity of the occupants' mobile phones. Therefore, directly measuring inter Wi-Fi session distance is error prone (see Figure 2(a)).

Our zone-based clustering method consists of a novel de-noising procedure that is applied to the collected Wi-Fi sessions. Given a Wi-Fi session S containing m Wi-Fi signal vectors V_{Sj} ($1 \leq j \leq m$), our de-noising procedure works in three steps.

1) Computing the 2-grams of Wi-Fi session signals, which are robust to phone heterogeneity. Specifically, each V_{Sj} is converted to an $ngram_i(seq(V_{Sj}))$, with $n = 2$, $1 \leq i \leq |seq(V_{Sj})| - n + 1$, $1 \leq j \leq m$, using Equations 2 and 3.

2) Averaging Wi-Fi RSS values to minimize signal noise. Specifically, for each $ngram_i(seq(V_{Sj}))$, $n = 2$, we define APS_S as the set of all 2-gram AP sequences obtained from S and RSS_S as the corresponding set of RSS differences between the two APs in each 2-gram. Let APS^*_S be the set of *unique* 2-grams in APS_S (note $APS^*_S \subseteq APS_S$), for each unique 2-gram $APS_{Sk} \in APS^*_S$, we identify all occurrences of APS_{Sk} in APS_S and compute the average of their corresponding RSS difference values. That is, for $1 \leq i \leq |APS_S|$,

$$RSS_{Sk} = avg\{RSS_{Si} | APS_{Si} = APS_{Sk}\}. \quad (5)$$

3) Weighting each unique 2-gram APS_{Sk} based on its occurrence rate, with more frequent 2-grams having heavier weights. The *occurrence rate* of each unique access point sequence APS_{Sk} is equal to the total number of occurrences of that sequence in the session S divided by the number of Wi-Fi signal vectors in that session. For $1 \leq i \leq |APS_S|$,

$$OCC_{Sk} = |\{APS_{Si} | APS_{Si} = APS_{Sk}\}| / m. \quad (6)$$

Then, the weighted RSS vector for session S is defined as follows:

$$WR(S) = \{APS_{Sk} : OCC_{Sk} \times RSS_{Sk}\} \quad (7)$$

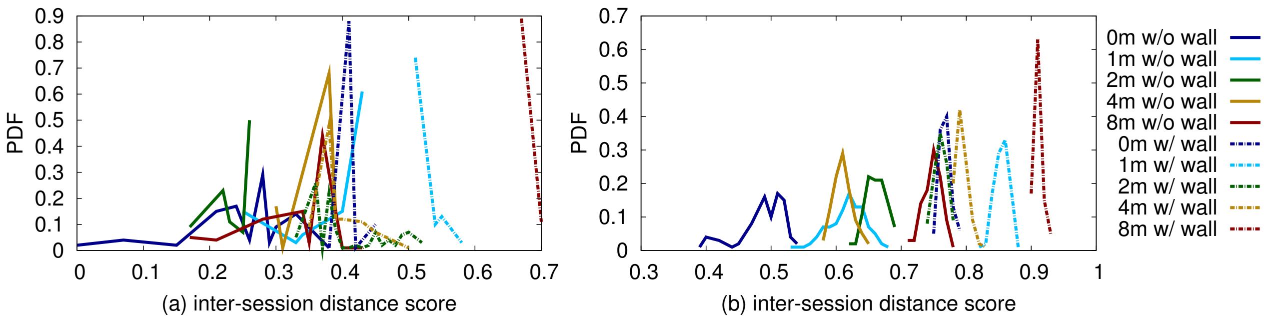

Figure 2. Within room or across wall: Distance function comparison between (a) direct RSS-based distance measure and (b) our method.

for all $APS_{Sk} \in APS^*_S$.

Finally, the distance function, based on Tanimoto distance, for two Wi-Fi sessions S_u and S_v is defined as

$$d(S_u, S_v) = 1 - \frac{WR(S_u) \cdot WR(S_v)}{|WR(S_u)|^2 + |WR(S_v)|^2 - WR(S_u) \cdot WR(S_v)}. \quad (8)$$

For each access point sequence APS_k that appears only in $WR(S_u)$, we set the corresponding weighted RSS value to zero in $WR(S_v)$, and vice versa.

Figure 2 compares the accuracy of inter-session distance measure using RSS directly [14] (Figure 2(a)) with our method (Figure 2(b)). In this study, two Wi-Fi sessions are gathered under different physical distances (0 to 8 meters), either within the same room (w/o wall) or in different rooms (w/ wall). It shows that our method can differentiate Wi-Fi sessions in different physical scenarios.

3.2 Density-based Wi-Fi Clustering

Given the inter Wi-Fi session distance measure described above, ARIEL incorporates a clustering procedure to partition the collected Wi-Fi sessions into clusters. Each cluster corresponds to one of the stationary occupancy hotspots.

We propose to use density-based clustering [7], which features a well-defined cluster model called "density-reachability". It connects points within a specific distance threshold (*Eps*) and a specific density criterion: minimum number of points ($MinPts$) within that distance threshold. Setting parameters *Eps* and $MinPts$ is discussed in Section 6. The rationale behind using density-based clustering is as follows. First, the stationary occupancy hotspots may have arbitrary shapes, which can be properly handled by density-based clustering, but not some other techniques, e.g., k-means clustering. Second, our distance measure may still leave residual noise and density-based clustering can effectively filter out such noise.

We now describe our density-based Wi-Fi session clustering technique. Given a set of Wi-Fi sessions D, we define a Wi-Fi session S_p as *density-reachable* from S_q, if there exists a chain of sessions $S_1, ..., S_n$ ($S_n = S_p$ and $S_1 = S_q$) that for $1 \leq i \leq n-1$,

$$S_{i+1} \in N_{Eps}(S_i) \text{ and } |N_{Eps}(S_i)| \geqslant MinPts, \quad (9)$$

where $N_{Eps}(S_i) = \{S_j | S_j \in D \wedge d(S_i, S_j) \leqslant Eps\}$.

Given the specific *Eps* and $MinPts$, we also define a Wi-Fi session S_p as *density-connected* to S_q if there exists a session S_k such that S_p and S_q are density-reachable from S_k. A *zone* z is defined as a non-empty subset of D satisfying the following conditions:

1. $\forall S_p, S_q \in D$: if $S_p \in z$ and S_q is density-reachable from S_p, then $S_q \in z$; and
2. $\forall S_p, S_q \in z$: S_p is density-connected to S_q.

Each zone, which can be in arbitrary shapes, consists of a set of Wi-Fi sessions and is robust to the outliers/noise. In order to identify zones incrementally as more Wi-Fi data are collected, the clustering process is conducted as follows.

Given a newly collected Wi-Fi session S, we first identify a set of Wi-Fi sessions within distance *Eps* from S, i.e., $D_S = \{S_k | S_k \in D_S \text{ and } d(S, S_k) \leqslant Eps\}$. Then we define a set D_{upd}, which includes Wi-Fi sessions that need to be updated:

$$D_{upd} = \{S_k | S_k \in D_S \cup \{S\} \text{ and } N_{Eps}(S_k)| \geqslant MinPts\}. \quad (10)$$

The new session S is categorized as follows:

- If D_{upd} is empty then there is no new cluster formed and no change to the existing clusters after the insertion of session S. Session S is marked as a noise and will be considered later.
- If none of the sessions in D_{upd} belongs to any zone before the insertion of S, then a new zone z is created that contains D_{upd} and S.
- If all the sessions in D_{upd} are members of the same zone z, then session S and possibly some $S_k \in D_{upd}$ are absorbed into zone z.
- If sessions in D_{upd} belongs to multiple zones, then all those zones and session S are merged into one zone.

4. MOTION-BASED CLUSTERING

In the motion-based zone clustering phase, we aim to identify zones (stationary occupancy hotspots) existing in the same room, then combine them together as the room's fingerprint. This is a difficult task because occupants may stay at any zone in a room and those zones can be far from, or close to, each other. Our goal is to combine zones in the same room, but not across rooms.

We describe a novel motion based-clustering algorithm based on an important observation – user mobility patterns are different in different rooms within a certain time period, because of the room functions or user habits and activities. For example, users may move more frequently in an office than in a meeting room and two users may have different mobilities in their own offices. Therefore, if two nearby zones are in the same room, they are likely to be surrounded by similar *moving Wi-Fi scans*, which are gathered when users are moving (discussed in Section 5). However, that is less likely to happen for two zones in different rooms. Our motion-based clustering algorithm determines whether two zones, z_a and z_b, are in the same room as follows.

First, we describe a process to select moving Wi-Fi scans for each zone. For a given zone z and K moving Wi-Fi signal vectors V_k ($1 \leq k \leq K$), we want to select m ($m \leq K$) moving Wi-Fi scans as motion profile of z, defined as $\mathbb{V}_z$, that can well represent how users move around near z within a time period. The selection is based on the probability of V_k belonging to z as follows:

$$\prod_{1 \leq i \leq |seq(V_k)|+n-1} P(ngram_i(seq(V_k))|z) > \tau_m. \quad (11)$$

We use n-grams of signal vectors because they are robust to noise and produce more accurate probabilities. The calculation of $ngram()$ is based on Equation 3. The parameter τ_m determines the closeness of the moving Wi-Fi signal vector V_k to zone z. If τ_m is close to 1, then the moving Wi-Fi scans in $\mathbb{V}_z$ are very close to z. $\mathbb{V}_z$ can only represent motion features of that zone, not the room, and are not helpful for identifying the inter-zone correlation. On the other hand, if τ_m is close to 0, the moving Wi-Fi scans in $\mathbb{V}_z$ surrounding the zone within a large area (e.g., cross multiple rooms) are selected. In that case, $\mathbb{V}_z$ may present motion features of multiple rooms, which may cause the merging of zones in different rooms. In our system, we set

$$\tau_m = 0.85 \times \max_k \{\prod_i P(ngram_i(seq(V_k))|z\}$$
$$\text{for } 1 \leq k \leq K, 1 \leq i \leq |seq(V_k)| + n - 1, \quad (12)$$

where $\max_k\{\prod_i P(ngram_i(seq(V_k))|z\}$ is the highest probability that a moving Wi-Fi scan V_{max} belongs to z. Namely, V_{max} is gathered at a position that is nearest to z, e.g., when users are leaving or arriving at z. We set τ_m a value slightly lower (0.85 $\times$) than the highest probability in order to get the maximum number of moving scans for z in the same room. Note that higher values of τ_m can result in failure to identify the correlation between two zones far from each other in a large room. In that case, we need to leverage middle-point zones in between or expand the size of the zones to shorten the distance between them.

Second, given two zones z_a and z_b, and corresponding motion profiles $\mathbb{V}_{z_a}$ and $\mathbb{V}_{z_b}$, we calculate the motion profile similarity of z_a and z_b using Jaccard similarity:

$$J_{z_a,z_b} = \frac{\mathbb{V}_{z_a} \cap \mathbb{V}_{z_b}}{\mathbb{V}_{z_a} \cup \mathbb{V}_{z_b}}. \quad (13)$$

If J_{z_a,z_b} is close to 1, its indicate that z_a and z_b are most likely in the same room because they are surrounded by almost the same moving Wi-Fi scans. Note that at least one of the zones has none empty $\mathbb{V}$ in calculation of J_{z_a,z_b}.

Calculating the value of J_{z_a,z_b} based on data gathered during a short time period may result in accuracy because the motion patterns in two adjacent rooms may be briefly similar. In response to that problem, the system calculates the value of J_{z_a,z_b} multiple times over different time periods. It then calculates the average value of J_{z_a,z_b} in order to eliminate above noise. Setting threshold τ_J for J_{z_a,z_b} is discussed in Section 6.

5. WI-FI SENSING ON MOBILE PHONES

In this section, we present an energy-efficient Wi-Fi sensing technique that automatically and intelligently collects representative stationary Wi-Fi sessions and moving Wi-Fi scans from mobile phones. The Wi-Fi collection process is controlled by two modules: 1) motion detection to determine a user's current status (moving or stationary), and 2) duplication checking to determine if Wi-Fi collection is needed at current location.

5.1 User Motion Detection

The key challenge of user motion detection on mobile phones is to achieve high accuracy and energy efficiency. Since a user's motion status can change at any time, the detection process needs to be constantly active. Previous work on motion detection has used either high-frequency (32 Hz) accelerometer readings to accurately detect all types of user activities [18], or medium-frequency (20 Hz) accelerometer readings to detect user's motion status [20]. The average power consumption of accelerometer is 330 mW for 30–50 Hz, 290 mW for 20–25 Hz, and only 50 mW for 4–7 Hz [14]. The first two frequencies result in unacceptably short battery lifespans if used continuously.

We describe a novel motion detection algorithm that can accurately determine if a user is moving or stationary using only 5 Hz acceleration sampling. The algorithm is based on our observation that when a person is moving (mainly walking), the average of *absolute* acceleration changes (m_{abs}) within a short period (e.g., 3 seconds) is large but the average of acceleration changes (m) is small. This is because when a user is walking, the body is in oscillation. As a result, the sign of acceleration readings also oscillate from positive and negative values, canceling each other in the average acceleration change but not the average absolute acceleration change. However, when user is not walking (e.g., no position change), the above property of acceleration data (m is large and m_{abs} is small) rarely occurs. For example, when a user types on phone, both m and m_{abs} will be small. And when a user picks up phone from desktop, both m and m_{abs} will be large.

Given acceleration readings (X, Y, Z) in a 3-second window with sampling frequency of 5Hz, and X, Y, and Z are acceleration readings in three axis, we first calculate the acceleration changes (i.e., difference of adjacent readings) on each axis, $(\Delta X, \Delta Y, \Delta Z)$. We then calculate the average acceleration change m as

$$m = \|(\mathrm{avg}(\Delta X), \mathrm{avg}(\Delta Y), \mathrm{avg}(\Delta Z))\| \tag{14}$$

and the average absolute acceleration change m_{abs} as

$$m_{abs} = \|(\mathrm{avg}(\Delta X_{abs}), \mathrm{avg}(\Delta Y_{abs}), \mathrm{avg}(\Delta Z_{abs}))\| . \tag{15}$$

We conducted a set of experiments on 10 users of different gender and age to collect acceleration data. Those experiments include stationary scenarios: completely stationary, playing with phone, and standing with some body movements; and moving scenarios: walking with phone in pocket (both in jacket and pants) and in a bag (on shoulder and in hand), walking on stairway, and with various walking speed. Figure 3 shows a scatter plot of m_{abs} and m/m_{abs} values we have obtained in different moving and stationary scenarios. We observe that moving instances are concentrated in the upper left region. The outliers are due to users making turning or pausing. Based on this figure, we determine a user is moving when $\tau_{m_{abs}} \geqslant 300$ and $\tau_{m/m_{abs}} \leqslant 0.15$, otherwise the user is stationary.

Once a user is determined to be stationary, the system starts stationary Wi-Fi session sampling till the user changes to the moving status. When the user is determined to be moving, the system starts moving Wi-Fi scans sampling continuously for 5 minutes or till the user changes to the stationary status.

5.2 Wi-Fi Duplication Check

Our automatic Wi-Fi collection mechanism is passive and highly dependent on user motion patterns. Thus, we may collect a lot of redundant Wi-Fi data at places that a user visits often, such as home and office. Such redundant data increase the energy use on phone and computation overhead on server without improving the clustering results. We describe a Wi-Fi duplication check mechanism to stop Wi-Fi collection at a place when sufficient data have been collected.

Our duplication check is performed at the zone level. The server maintains a status for each zone – "full" if the zone contains enough Wi-Fi sessions (we use a threshold of 50) and "not full" otherwise. This status

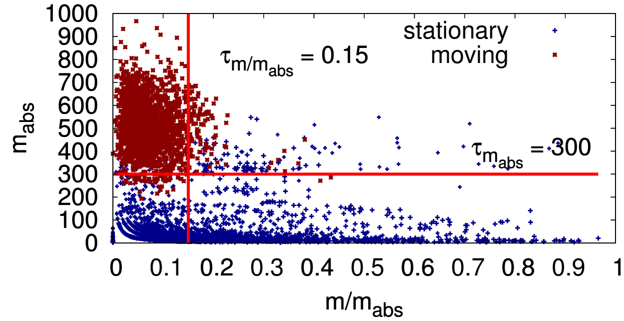

Figure 3. Thresholds for motion detection (moving vs. stationary).

information is distributed to phones periodically. On the phone side, the application first samples one Wi-Fi signal V, and calculates the most likely zone $\hat{z}$ based on Equation 4. If $P(ngram_i(seq(V))|\hat{z})$ is larger than a given threshold (0.8 in our system) and zone $\hat{z}$ is full, then the system skips the stationary scan and following moving scans. Otherwise the Wi-Fi data will be collected.

6 EVALUATIONS

ARIEL has been implemented and deployed for real-world evaluation. This section presents the experimental setup, system accuracy, and efficiency analysis results. We also explore ARIEL's parameter settings and discuss the problems encountered during user study.

6.1 Experimental Setup

ARIEL is implemented on mobile phones and a server. The zone-based Wi-Fi clustering algorithm and motion-based zone clustering algorithm are implemented in python on the server side. The motion detection algorithm is implemented on the Android mobile platform. In addition, the Wi-Fi signal vector and room fingerprint database is built using MySQL. A web server receives Wi-Fi signal vectors from mobile phones, delivers room localization services, and synchronizes room fingerprints to mobile phones.

Over a period of 10 months, a total number of 21 participants, including faculty members and graduate students, have participated in the user study. Overall, we have collected Wi-Fi data for 193 rooms, in which 85 have at least one adjacent room and 61 rooms have been visited by at least two participants. The user study covered a wide range of building environments, including hospitals, supermarkets, restaurants, a university campus, apartments, and houses.

For evaluation purpose, during the user study, each user was asked to provide related room information, e.g., room name, room entry/departure time, and activity in the room, through a user interface integrated in our mobile application. Weekly meetings with the users were held to verify the accuracy of the user input data. The user data was also verified using collected GPS data (for the rooms in which GPS signals were available) and Wi-Fi data in building level. This data set is used to establish ground truth between each Wi-Fi session and the corresponding room. Each cluster automatically identified by ARIEL is mapped (via majority voting) to the room with most Wi-Fi sessions in the cluster.

6.2 Accuracy

This section first evaluates the overall accuracy of the room localization service provided by ARIEL, and then investigates the accuracy of the fingerprinting techniques.

ARIEL automatically builds room fingerprints through an incremental process using the Wi-Fi signal vector samples gathered by personal mobile phones, and then delivers room localization service back to the individual occupants. Therefore, the accuracy of the room localization service depends on the accuracy of room fingerprints, which in turn is a function of the amount of Wi-Fi signal samples collected for each room. A good room localization system should achieve high accuracy with a reasonable number of Wi-Fi signal samples.

Figure 4 shows the accuracy of the ARIEL room localization service as a function of the number of Wi-Fi signal vector samples. Given the user study including 21 participants and 193 rooms, it shows that, first, when the Wi-Fi samples collected per room are less than 200, ARIEL is unable to generate fingerprints. This is due to the fact that, given the default system setting of ARIEL, the clustering algorithm cannot build up meaningful clusters using such limited number of Wi-Fi samples. Next, as the collected Wi-Fi sample count increases, the accuracy of room localization quickly improves, and the variance of service quality decreases. For 400 or more samples per room, ARIEL achieves 95% accuracy, which is comparable to past works that require time-consuming manual annotation [9, 12]. Note that obtaining 400 Wi-Fi samples per room requires collection of 15 Wi-Fi sessions, and each session takes less than two minutes.

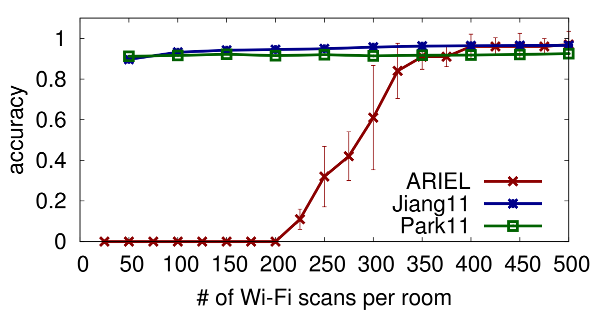

Figure 4. Room localization accuracy vs. number of Wi-Fi sessions.

Figure 5 shows room localization accuracy as a function of numbers of users. It shows that the accuracy of ARIEL improves as the number of users per room increases. When more users visit the same room, a better spatial coverage in terms Wi-Fi signal samples will be obtained during room fingerprinting. On the other hand, even if room fingerprinting is conducted based on data from a single user, ARIE is still able to localize with over 90% accuracy.

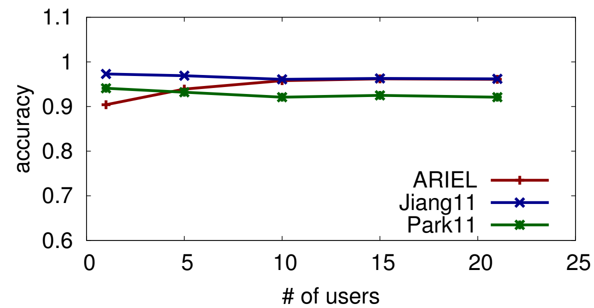

Figure 5. Room localization accuracy vs. number of users.

We next evaluate the room fingerprinting techniques, and focus on evaluating the performance of the clustering algorithms. Specifically, ARIEL uses clustering techniques to construct room-level clusters consisting of Wi-Fi sessions belonging to each room. Two potential errors introduced by the clustering methods include 1) Wi-Fi sessions collected in room A are incorrectly assigned to a cluster belonging to room B and 2) the clustering algorithm is unable to merge all the zones belonging to the same room into one cluster, resulting in multiple clusters per room. To this end, we introduce the following two performance metrics.

1. *Purity* measures the percentage of Wi-Fi sessions with correct room assignment divided by the total number of assigned Wi-Fi sessions.

2. *Unity* measures the quality of room-level zone merging, i.e., it is the reciprocal of the total number of zones assigned to each room, a average over all the rooms. The unity of one is ideal.

Figure 6 shows the accuracy of the clustering methods as a function of number of Wi-Fi sessions. ARIEL starts to produce clusters when the number of collected Wi-Fi sessions is greater than 200. The unity measure improves as the number of Wi-Fi sessions increases, because more Wi-Fi session collections form more connections between in-room stationary hotspots, thus improving clustering quality. The value of unity converges when the total number of Wi-Fi sessions reaches 400. The purity measure slightly degrades as the number of Wi-Fi sessions increases, which introduces more Wi-Fi variations.

Figure 7 shows the clustering accuracy as a function of the number of users. When the system has more users, the purity of the clusters slightly degrades as more variations are introduced by different users and different phones. Meanwhile, the unity of the clusters improves as different motion patterns of users within a room help bridge in-room hotspots, thus improving

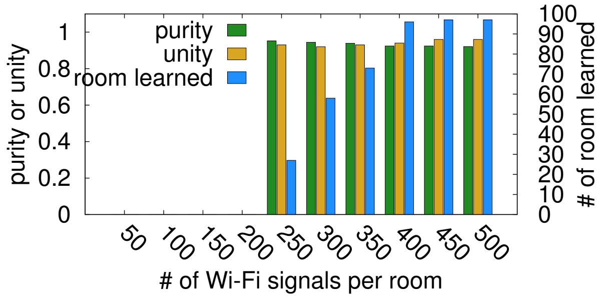

Figure 6. Clustering accuracy vs. number of Wi-Fi sessions.

the clustering quality.

Overall, this study shows that ARIEL offers high clustering quality.

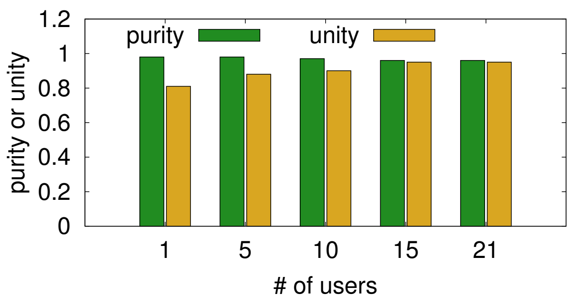

Figure 7. Clustering accuracy vs. number of users.

6.3 Energy Efficiency

The energy overhead imposed by ARIEL on the mobile phone has two main parts: Wi-Fi sensing and motion detection. We calculate the average power consumption based on power measurement data from Android HTC G1 [1]. The average power consumption on reading acceleration at 4–7Hz is 50mW, at 20–25Hz is 285mW, and at 30–50Hz is 325mW. The average power consumption of Wi-Fi scan at 1/5Hz is 108mW.

We first compare our motion detection method with two other approaches: 1) frequent acceleration sampling [18], which targets general user activity recognition including stationary and walking and 2) moderate acceleration rate sampling [20], which focuses on motion detection. As shown in Figure 8, given the same data set, our detection algorithm achieves the highest energy efficiency, nearly 5$\times$ better than that of the other two approaches. In addition, the average motion detection accuracy of our approach is 94%, which is higher than the medium-frequency approach and slightly lower than the high-frequency approach.

We also study the power consumption of Wi-Fi sensing. We found that the active time duration of Wi-Fi sensing varies among users. On average, the Wi-Fi active time over a day ranges from 20 minutes to 2 hours with an average sampling frequency of 1/5Hz. The corresponding average power consumption is 4.5mW.

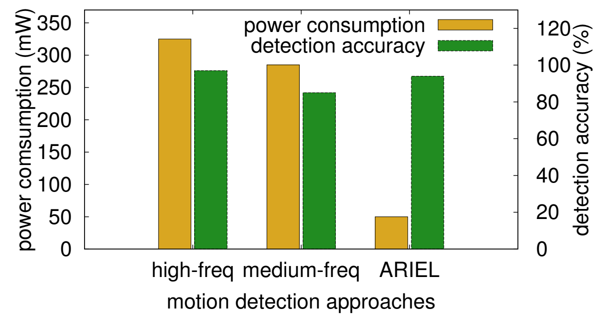

Figure 8. Comparison of power consumption and detection accuracy of different motion detection methods.

6.4 Parameter Settings

Several parameters in the ARIEL system need to be determined: 1) density parameters Eps and $MinPts$, 2) threshold τ_J for motion-based clustering, and 3) size of Wi-Fi stationary session (number of Wi-Fi scans in a session). These three parameters directly impact the purity and unity of the room fingerprints.

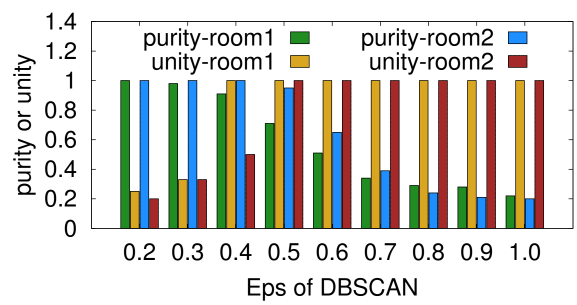

Figure 9. Impact of different Eps values in two different rooms.

In our density-based clustering algorithm, we set $MinPts$ to 5 for all data because 1) in our experiments, we found no difference among clustering results when $MinPts \geq 5$; and 2) larger $MinPts$ requires more computation effort and more Wi-Fi sessions per room. Figure 9 shows the purity and unity of two rooms with different Eps values. The two rooms have different wall types: room 1 shares a thin drywall with adjacent rooms, while room 2 shares a thick concrete wall with adjacent rooms. Generally, larger Eps can increase the unity of cluster but reduce the purity, vice versa. The two rooms with different wall types have different clustering results. Without prior knowledge of the wall type, it is difficult to choose the right value for Eps. As shown in Figure 10, the average purity and unity of all data change with the value of Eps. We set Eps conservatively to 0.32, which ensures high purity for all the clusters but relatively low unity, i.e., a room may have multiple clusters but each cluster has high purity. The low unity of the clusters will be compensated for in the merging phase.

Another important parameter in our system is the zone merging threshold τ_J introduced in Section 4. As shown in Figure 11, if we set the threshold too low, the purity decreases a lot. On the other hand, if we set the threshold too high, clusters in the same room may not

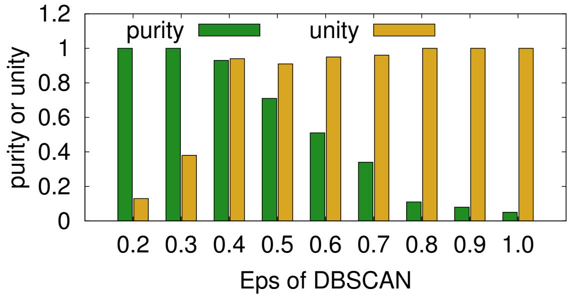

Figure 10. Average clustering accuracy as a function of Eps.

be merged. In the tests, we empirically set τ_J to 0.5.

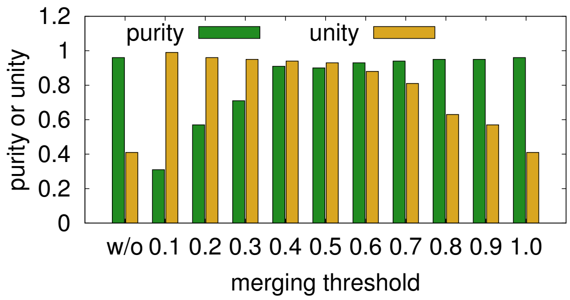

Figure 11. Average clustering accuracy as a function of merging threshold τ_J.

Session size also impacts system performance. A session with more Wi-Fi scans gives more accurate distances, but requires more energy for Wi-Fi scan. Figure 12 shows that, as session size increases, the distance between two sessions become more stable. We evaluated our system with different session sizes; the results are shown in Figure 13. When session size is larger than 30, the room purity and unity do not change much. Although the session distance is not perfect with a size of 30 as shown in Figure 12, our density-based clustering algorithm is robust to outliers and achieves high accuracy even with some noise.

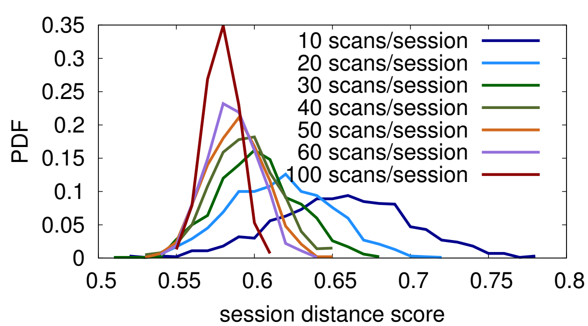

Figure 12. Session distance distribution vs. session size.

7. RELATED WORK

In this section, we survey works that are most related to ours, focusing on indoor localization techniques based on Wi-Fi signals. We also discuss works that aim to reduce the effort of fingerprint collection for indoor localization.

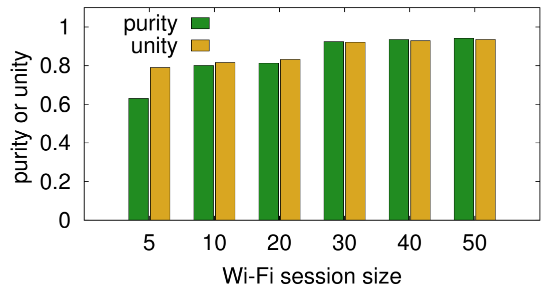

Figure 13. Average room localization accuracy vs. session size.

Indoor localization has been a topic of active research, some focusing on indoor positioning while others (similar to our work) aiming to determine rooms rather than the exact indoor positions. Indoor positioning systems [2, 10] can be easily extended to room-level localization but require extensive effort for fingerprint sampling or the pre-knowledge of AP positions, which are not scalable. Previous room location methods either rely on manual room fingerprint collection, which imposes a high cost [11], or leverage user feedback to reduce deployment and maintenance cost [3]. Issues such as determining when user input is actually required, and discounting erroneous and stale data are addressed by the work of Park et al. [19, 9]. However, their approach requires a floor plan to deal with input errors and the inconsistent room naming preferences of different users. Furthermore, the input effort is non-negligible, given the small screens and keyboards of mobile phones. Most of these systems use Wi-Fi signals as room or position fingerprints.

To reduce fingerprint collection efforts, some works focused on reducing sampling locations and sampling time while maintaining similar localization accuracy [15, 4]. Other works proposed calibration-free approaches to improve scalability but require AP infrastructure information. Gwon and Jain proposed TIX (triangular Interpolation and extrapolation), a calibration-free mechanism that used the three APs with the highest RSSIs to determine the centroid of the triangle [10]. Lim et al. proposed a zero-configuration indoor localization method, which used online calibration and truncated singular value decomposition (SVD) to characterize the relationship between RSSI and geographical distance to anchors [16]. More recently, unsupervised indoor positioning systems [5, 22] have been proposed by leveraging GPS signals near a window or internal landmark that can be easily identified. However, these approaches rely on indoor GPS signasl, indoor landmarks, and/or floor plans. Our solution is fully automated and requires no manual annotation or prior knowledge of APs or floor plans, yet achieves high room localization accuracy.

Density-based clustering [7, 6] is a popular method and attractive for spatial identification. It clusters objects based on neighborhood density, which is defined by a given radius (Eps) and a minimum number of objects

($MinPts$). It can identify clusters of arbitrary shapes and is robust to noise. Density-based clustering is thus appropriate for our problem and achieves high performance as demonstrated by the evaluation results.

8. CONCLUSIONS AND DISCUSSIONS

This paper has presented ARIEL, a fully automated indoor room localization system. To accurately identify rooms without extensive manual annotation, we proposed and developed a number of novel techniques: 1) a zone-based clustering algorithm that accurately identifies in-room occupancy hotspot(s) using Wi-Fi signatures; 2) a motion-based clustering algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and 3) an energy-efficient motion detection algorithm to minimize the noise of Wi-Fi fingerprints. ARIEL has been implemented and deployed for a 10-month study with 21 participants. The evaluation results demonstrate that our automated clustering algorithm generates clusters that are high representative of individual rooms and achieves high accuracy (95%) for room localization. The accuracy is comparable to existing techniques that require labor-intensive manual annotation.

The cluster identities generated by ARIEL can serve as room identification and are sufficient for most applications that do not require semantic name/label for each room. If room names are required, ARIEL can rely on users' feedback [13] to provide this information. The advantage of our system is that there are much fewer clusters than WiFi fingerprints, significantly reduced annotation effort. With unique identities, we can easily identify different rooms with the same name, learn how people name the same room differently, and identify commonly-used names for a room.

ACKNOWLEDGEMENTS. This work was supported in part by the National Science Foundation under awards CNS-0910995, CNS-0910816, CNS-1059372, and TL-0964545.

REFERENCES

1. G1 Android Power Measurement. http://urban.cens.ucla.edu/resources/g1power/.
2. P. Bahl and V. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *INFOCOM '00*, 2000.
3. E. S. Bhasker, S. W. Brown, and W. G. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. In *PerCom'04*, 2004.
4. X. Chai and Q. Yang. Reducing the calibration effort for probabilistic indoor location estimation. *Mobile Computing, IEEE Transactions on*, 2007.
5. K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *MobiCom '10*, 2010.
6. M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental clustering for mining in a data warehousing environment. In *VLDB '98*, 1998.
7. M. Ester, H. peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD '96*, 1996.
8. W. Fisk. Health and productivity gains from better indoor environments and their relationship with building energy efficiency. *Annu. Review Energy Environ*, 25:537–566, 2000.
9. J. geun Park, D. Curtis, S. Teller, and J. Ledlie. Implications of device diversity for organic localization. In *INFOCOM '11*, 2011.
10. Y. Gwon and R. Jain. Error characteristics and calibration-free techniques for wireless LAN-based location estimation. In *MobiWac '04*, 2004.
11. A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom '04*, 2004.
12. Y. Jiang, K. Li, L. Tian, R. Piedrahita, X. Yun, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. MAQS: a personalized mobile sensing system for indoor air quality monitoring. In *UbiComp '11*, 2011.
13. D. H. Kim, K. Han, and D. Estrin. Employing user feedback for semantic location services. In *UbiComp '11*, 2011.
14. D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. SensLoc: Sensing everyday places and paths using less energy. In *SenSys '10*, 2010.
15. J. Krumm and J. C. Platt. Minimizing Calibration Effort for an Indoor 802.11 Device Location Measurement System. In *Microsoft Research, Technical Report MSR-TR-2000-12*, 2003.
16. H. Lim, L.-C. Kung, J. C. Hou, and H. Luo. Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure. *Wireless Netwroks*, 2010.
17. J. Lin, G. Xiang, J. I. Hong, and N. Sadeh. Modeling people's place naming preferences in location sharing. In *Ubicomp '10*, 2010.
18. H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys '10*, 2010.
19. J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *MobiSys '10*, 2010.
20. A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod. Accurate, low-energy trajectory mapping for mobile devices. In *NSDI'11*, 2011.
21. U.S. Environmental Protection Agency Green Building Workgroup. Buildings and their impact on the environment: A statistical summary, 2009.
22. H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *MobiSys '12*, 2012.