Push the Limit of WiFi based Localization for Smartphones

Hongbo Liu†, Yu Gan†, Jie Yang†, Simon Sidhom†, Yan Wang†,
Yingying Chen†, Fan Ye*

†Stevens Institute of Technology, Hoboken, NJ 07030, USA
†{hliu3, ygan, jyang, ssidhom, ywang10, yingying.chen}@stevens.edu

* IBM Thomas J. Watson Research Center, Hawthorne, NY 10532, USA
*fanye@us.ibm.com

ABSTRACT

Highly accurate indoor localization of smartphones is critical to enable novel location based features for users and businesses. In this paper, we first conduct an empirical investigation of the suitability of WiFi localization for this purpose. We find that although reasonable accuracy can be achieved, significant errors (e.g., $6 \sim 8m$) always exist. The root cause is the existence of distinct locations with similar signatures, which is a fundamental limit of pure WiFi-based methods. Inspired by high densities of smartphones in public spaces, we propose a peer assisted localization approach to eliminate such large errors. It obtains accurate acoustic ranging estimates among peer phones, then maps their locations jointly against WiFi signature map subjecting to ranging constraints. We devise techniques for fast acoustic ranging among multiple phones and build a prototype. Experiments show that it can reduce the maximum and 80-percentile errors to as small as $2m$ and $1m$, in time no longer than the original WiFi scanning, with negligible impact on battery lifetime.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed Applications*; C.3 [**Special-Purpose and Application based Systems**]: *Real-time and embedded systems*; H.5 [**Information Interfaces and Presentation(e.g., HCI)**]: Sound and Music Computing

General Terms

Design, Experimentation, Measurement, Algorithms, Performance

Keywords

Smartphone, Peer Assisted Localization, WiFi fingerprint localization

1. INTRODUCTION

Indoor localization is a critical enabler for location based smartphone applications. In many environments (e.g., airport terminals, railway stations and shopping malls), the location helps users access navigation, merchandise and promotion information; businesses need it to understand the patterns of customer visit and stay, such as the popularity of different sections in the store, or the spatial-temporal distribution of passenger flows.

Accurate indoor localization on smartphones, however, remains elusive. Although there have been some recent commercial offerings such as Google Maps 6.0 and Shopkick [1], they either have errors up to 10 meters [9], or only locate at the granularity of stores. There has been a plethora of academic work on indoor localization. Those achieving high accuracy usually require special hardware not readily available on smartphones [20, 21], or infrastructure expensive to deploy [6, 17]. WiFi-based localization leverages prevalent wireless access points, thus avoiding such drawbacks. But most studies [3, 27] have been largely based on laptops with quite different antenna forms and possibly radio characteristics, whereas recent work on smartphone indoor localization [2, 15, 24] achieves room or floor level accuracies. The feasibility of leveraging the most prevalent WiFi infrastructure for high accuracy localization on smartphones is still an open question.

In this paper, we first conduct a set of experiments to empirically study the impact of various factors on the accuracy of WiFi localization on smartphones. We find that although reasonable accuracy (e.g., $3 \sim 4$m) can be achieved, there always exist large errors (e.g., $6 \sim 8$m) unacceptable for many scenarios. Similar or much larger errors (e.g., $>$15m) have been reported in previous studies [3, 27]. One work [7] found that high accuracy (e.g., sub-meter median and 2m maximum) is possible but only under hundreds of APs, infeasible in practical settings. Such errors may cause a passenger make a wrong turn leading to a different train platform, or a store erroneously stock up for a section with much less real customer interests. Our investigation on these large errors reveals the insight that they are caused by possibly faraway locations with similar WiFi signatures, an intrinsic phenomenon of the radio signal propagation and fundamental limit of WiFi methods.

On the other hand, we observe that smartphones are gradually woven into our social life and usually a high density of them exist in public spaces. The relative positions of nearby peer devices could be used as physical constraints on the possible location of a smartphone. Inspired by this observation, we set out to study how to exploit the unique physical constraints among smartphones to reduce large errors and push the limit of WiFi based approaches. We propose a peer-phone assisted localization approach that leverages the acoustic ranging between peers, without requiring special hardware yet producing highly accurate location estimates.

In particular, the peer-phone assisted localization can be carried out concurrently with WiFi localization or when a smartphone has

MobiCom'12, August 22–26, 2012, Istanbul, Turkey.

obtained a rough location from WiFi but needs further improvements. The targeting smartphone exchanges sound signals with nearby peer devices. A server collects such ranging estimates and constructs a graph of the relative positions among peers. It applies a localization algorithm that maps the vertices of the graph against the WiFi signature database to locate all peers jointly. Experiments using data from various environments, including the airport, train station, and shopping mall, have shown that our approach can reduce 80 percentile error to about 1m, and limit the maximum error to about 2m, demonstrating the feasibility of WiFi for high accuracy localization.

Specifically, we make the following contributions:

- We discover the root cause of large errors as the existence of faraway locations sharing similar radio signatures, which is due to the intrinsic dynamic propagation of the radio signal, thus presenting a fundamental limit of WiFi methods.
- We propose a peer-phone assisted localization approach utilizing minimum auxiliary COTS sound hardware for reducing large errors and push the limit of WiFi approaches. We devise a peer-assisted localization algorithm that leverages acoustic ranging and locates peer phones jointly for greatly improved accuracy. We identify the frequencies, sound signal design, detection and emission scheduling methods appropriate for fast ranging among multiple peers that are unobtrusive, robust to noise and have minimum impact on users' regular activities.
- We prototype our system and carry out real world experiments. The results demonstrate that our approach greatly reduces the maximum error from $6-8$m to 2m, and limit 80 percentile error to 1m, which were shown empirically possible but only under hundreds of APs [7]. The assistance finishes in time no longer than a few seconds of WiFi scanning, and poses negligible impact on battery lifetime.

The rest of this paper is organized as follows. In Section 2, we perform a systematic evaluation on WiFi localization on smartphones and report our findings. Section 3 introduces our system design and present the detailed peer assisted localization algorithm. We study the frequencies, sound detection methods for fast concurrent ranging among multiple peer phones in Section 4. We describe the system implementation and report evaluation results in Section 5. We discuss related issues and survey related work in Section 6 and 7. Finally, Section 8 concludes the paper.

2. PERFORMANCE OF USING WIFI ALONE FOR SMARTPHONE LOCALIZATION

To understand the practical performance of smartphone localization using WiFi, we first conduct a systematic study on the impact of various factors (e.g., orientation, holding position, time of the day and number of samples). We find that reasonable accuracy can be achieved in many cases (e.g., $\sim 4m$). However, large errors (e.g., beyond $6 \sim 8m$) always exist. Further investigation reveals two root causes: static environmental effects, and dynamic obstacles or interferences, both of which pose fundamental limits on WiFi localization accuracy.

2.1 Methodology

Fingerprint Based Localization. Fingerprint based method was pioneered by Bahl et. al [4] and is the most popular WiFi localization approach. It first measures the "fingerprint", the WiFi signal strengths from various access points (APs) at a number of known locations and stores them as *training data*. A device samples the signal strengths from various APs to obtain *testing data*. Then an algorithm finds the "closest fingerprints" in the training data to the sample, using Euclidean distance in the signal space where each dimension is for a different AP. A location estimation is given based on the locations of the closest fingerprints (e.g., the centroid of a few "closest" fingerprint locations). Recent work [8, 24] showed that the training data could be constructed without extensive site survey, making this approach even more attractive. In the test, we build a fine-grained signature map as training data, using interpolation between locations with actual measurements.

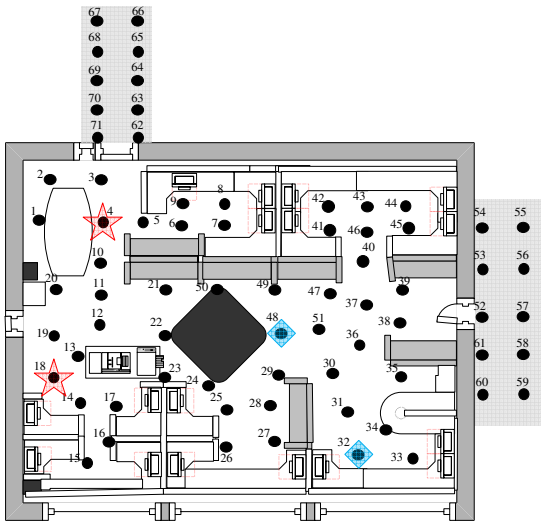


Figure 1: Each dot in the floor map represents a location where the RSS fingerprint is measured.

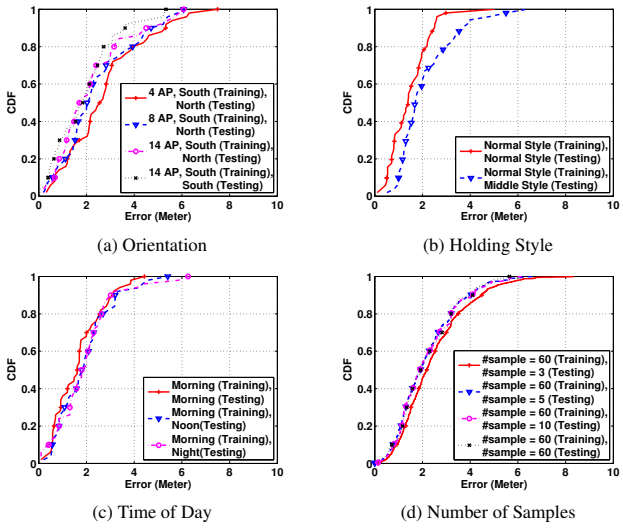
Experimental Setup We conduct the study in an office environment as shown in Figure 1 with a set of Android phones. The experimental area is 12 m $\times$ 11 m with hallways, office wall dividers and furnitures, such as desks, shelves and chairs. The phone takes 60 Received Signal Strength (RSS) samples at each of the 71 known locations from 14 APs. Each location can observe signals from 8-9 APs on average. We repeat the above process for each of the 4 factors, orientation, holding position, time of the day and number of samples (shown in Table 1, to understand how they impact the localization performance. For each test, the default parameters are south for orientation, normal style for holding position, 60 samples and morning time.

2.2 Impact of Various Factors

Figure 2 presents the cumulative distribution function (CDF) of the localization error under various factors. We first examine how the orientation affects the RSS readings due to the blocking and reflection of radio signals by the human body. In outdoor cases,

Factors	Description	Value
Orientation	The direction to which the user is facing	West, North, East, South
Holding Position	The way the user holds the phone	Holding bottom, Holding middle
Number of Samples	The number of samples taken at each location	3,5,10,60
Time of the Day	Different times that we conduct the measurements	Morning, Noon, Night

Table 1: Factors under study



(a) Orientation (b) Holding Style

(c) Time of Day (d) Number of Samples

Figure 2: WiFi localization error under various factors.

10dB RSS difference for certain APs was observed on smartphones [29]. In our indoor environment, we recorded an average of 4dB difference. We believe this reduced variation is due to stronger multipath effects of indoor environments, thus the lack of direct line of sight does not attenuate the signal as much. In particular, as shown in Figure 2 (a), when a mismatched training data set is used (e.g., south-facing training set for north-facing user), long tails of CDF curves exhibit large errors ranging from 6m to 8m. Even when the matching training set is applied, errors beyond 5m still exist.

Due to the small size, how the user's hand holds the phone can affect the received radio signal as well. We tried two holding positions: bottom and middle. Figure 2 (b) shows that using mismatched training and testing data (e.g., bottom-holding as training to localize middle-holding phones) can lead to large error beyond $6m$. In addition, the localization results are evaluated under three different times of the day representing morning, noon and evening in Figure 2 (c). We again observe long tails beyond 5 meters when mismatched training and testing data are used.

Finally, more samples lead to more reliable measurements as the input for localization, but at the cost of higher energy and latency overhead. To understand the accuracy-overhead tradeoff, we depict the localization error when varying the number of samples from 3 to 60 in Figure 2 (d) under a training set of 60 samples per location. We find that using more than 5 samples does not significantly improve 90-percentile accuracy. We thus choose 5 samples in our WiFi localization study throughout the paper.

2.3 Root Cause of Large Errors

We make one critical observation from the above investigation: although reasonable accuracy can be achieved in many cases, large errors $6m$ or more always exist. In many applications these large errors can cause problems, such as giving the user incorrect navigation instructions, or wrong statistics about the visits of customers to different sections inside a store.

After a close examination of those large errors, we find the root cause. In essence, two physically distant locations happen to share similar WiFi signal strength measurements, thus a testing sample is erroneously localized to a physically faraway location with short Euclidean distance in the signal space. This can be classified into two cases: (1) permanent environmental settings such as walls, furniture placement, which affect radio signal propagation and create persistent similar signal reception; and (2) transient factors or measurement mismatch between training and testing data. Such transient variation in RSS reception is due to dynamic changes in the environment, such as a nearby moving object or wireless interference from other electronic devices, while the mismatch can be in orientation, holding style, time of the day or number of samples.

We illustrate case 1 by three locations in Figure 1: 18 and 13 are close to each other whereas 4 is farther at the other side of the room (marked as red stars). However, during testing we find that locations 18 and 4 share similar WiFi fingerprints. The distance between their fingerprints in the signal space is only 1.98 dB/AP, whereas the fingerprint at closeby location 13 has a distance of 2.44 dB/AP to that of location 18. In this case, the office wall dividers cause the large localization error beyond 6m at location 18.

The example for case 2 are location 32, 34 and 48 (marked as blue squares in Figure 1). Locations 32 and 34 are close to each other. However, we find that the WiFi fingerprint at location 32 becomes similar to that at location 48 at night when less people are around. Thus when testing at location 32 using the training data collected at night, location 32 will be matched to 48, instead of 34, resulting in large errors of over 6m. Through our study, we find that the percentage of large errors resulted from Case 1 is 60% to 70% while that from Case 2 accounts for the rest. Both cases are caused by irregular multipath reflections, an intrinsic character of radio signals. They present fundamental limits for WiFi localization to achieve high accuracy.

3. PEER ASSISTED LOCALIZATION

From the previous investigation, WiFi as-is is not a suitable candidate for high accuracy localization due to large errors. However, is it possible to address this fundamental limit without the need for additional hardware or infrastructure? Our answer is yes: by exploiting acoustic ranging, a phone can use nearby peer phones as reference points and obtain its relative positions to them. This imposes unique physical constraints on the possible location of the phone, thus reducing the uncertainty and improving the accuracy.

Such an idea is motivated by two observations. First, in many public indoor environments (e.g., airport terminals, railway stations, shopping malls and museums), there are usually a high density of users, thus smartphones. Each neighboring peer has a unique physical location for restraining the location uncertainty of a smartphone. Second, a number of research work [19, 26] has shown that highly accurate relative ranging can be achieved within a car (passenger vs. driver side) or between a pair of mobile devices (at centimeter accuracy) by using sound signals.

3.1 Design Goals and Challenges

The above concept may sound quite simple. However, building such a peer-assisted localization system involves a number of great challenges in both the design and implementation:

- **Peer assisted localization algorithm.** How to utilize the physical constrains imposed by the neighborhood peers to reduce the large errors incurred from WiFi localization? Given only the relative distances among peers and their location estimates are available in real scenarios, exactly what is the algorithm?

- **Concurrent acoustic ranging of multiple phones.** Previous work on acoustic position estimation was for one or two devices only. When there are multiple devices and they all do acoustic ranging, how can we tell which distance measure is

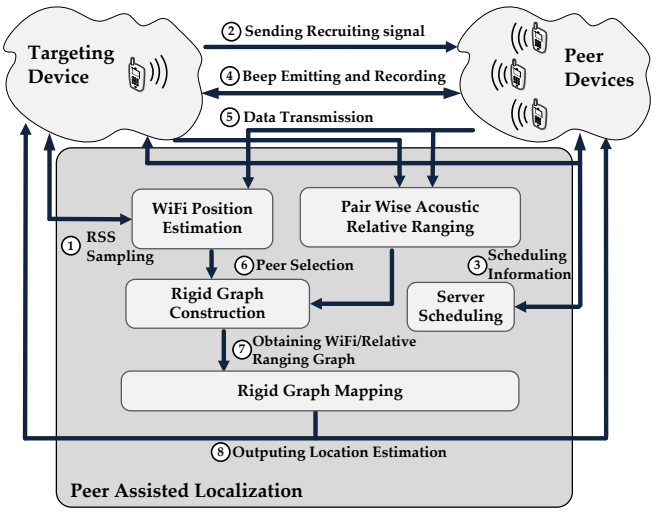


Figure 3: The workflow of our system.

for which pair? How to design and detect the sound signal, so that the system is robust to noises in different environments?

- **Ease of use.** The peer assistance process should complete in short time; otherwise users may have moved to different locations. The sound emitted should not annoy or distract users from their regular activities.

In the rest of this section, we will present an overview of the system, then present the details of the peer assisted localization algorithm. Section 4 addresses challenges in concurrent acoustic ranging and ease of use.

3.2 System and Algorithm Overview

Our system works as follows (shown in Figure 3): when a target phone needs to further improve accuracy or desires high accuracy upfront, it broadcasts a special audio signal to "recruit" nearby peers. Those receiving the recruiting signal send their identifiers to a server. The server comes up with a schedule about which phone should emit a beep signal at which time slot. The phones involved, both the target and peers, emit the beeps accordingly. They also record the beeps from others and send the files back to the server. All peers also conduct WiFi sampling and send the measurements to the server. The server determines the locations of peers from WiFi samples, and distances among them from acoustic ranging based on the recorded sound files. Finally the server computes the new location estimate of the target using our peer assisted localization algorithm, and sends back the result to the target.

The intuition underlying the peer assisted localization algorithm is to construct a graph based on the relative distances among devices, then "superimpose" the graph onto the signature map based on the initial WiFi location estimates. The algorithm "rotates" and translationally [1] "moves" the graph against the signature map, such that the vertices are placed "closest" to the true locations, as measured by certain metric. Where the vertices are placed become the new location estimates.

In the algorithm, we use as the metric the sum of RSS Euclidean distances between the WiFi samples of each device and the WiFi signature of where its graph vertex is placed. Such a metric quantifies the aggregate "closeness" between WiFi samples and new estimated locations of all devices. We use the example in Figure 4 to illustrate the intuition. Initially, each phone has a location estimate (vertices in the dashed-line graph), e.g., from WiFi localization. Because the acoustic ranging is highly accurate to identify the relative distances between peers (in terms of centimeters), the shape of the solid-line graph is quite close to that of the ground truth (the dotted-line graph). Such additional relative distance constraints "force" the new location estimate of the target to move closer to its real location, thus reducing large localization errors and achieving higher accuracy.

[1] "Translationally" means moving the graph without rotation.

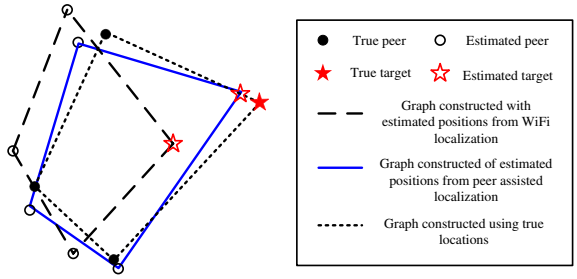


Figure 4: Illustration of using physical constraints to perform peer assisted localization.

3.3 Peer Assisted Localization Algorithm

Our peer assisted algorithm comprises of two main parts: (a) graph orientation estimation: estimating the range of graph orientation angle ϕ [2] by combining the acoustic ranging information and initial WiFi localization results. This is important because we find that its performance is sensitive to the orientation (shown in Section 5.5). (b) new location estimation: search for the optimal combination of graph orientation angle and translational position, such that the RSS distance summation is minimized.

The algorithm takes two inputs: 1) $\{p_i\}, i = 0, \cdots, M$: initial WiFi location estimations of the target phone (p_0) and its peers ($p_1, \cdots, p_M$); 2) $\{l_{i,j}\}, i, j = 0, \cdots, M$: pairwise acoustic ranging measurements between device i and j. The algorithm produces $\{q_i\}$, the new location estimate of each involved device i. In the following, the first 3 steps estimate the orientation range, and the last two find the new locations.

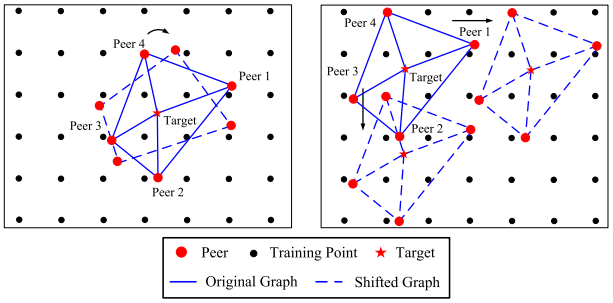
Step 1: Compute edge directions from acoustic ranging. The server constructs a graph G' with the pairwise ranging measurements, $l_{i,j}, i, j \in \{0, \cdots, M\}$. It rotates G' such that the direction of its longest edge is parallel to the X axis. Then it calculates the direction vector $v'_l = \frac{p'_i - p'_j}{|p'_i - p'_j|}, (i < j)$ for the L (e.g., 3)[3] longest edges, where p'_i, p'_j are the locations of the two vertices of an edge.

Step 2: Compute edge directions from initial WiFi localization. The server also computes the direction vectors for the same L edges using initial WiFi locations: $v_l = \frac{p_i - p_j}{|p_i - p_j|}, (i < j)$.

Step 3: Graph Orientation Estimation. The server rotates G' to find the optimal graph orientation Φ that maximizes the inner

[2] We define ϕ as the angle between the X axis and the direction of the graph's longest edge, from the vertex of the smaller ID to the other.

[3] When choosing L longest edges, we experimented various values (e.g., 2, 3 and 4) and found different values of L have little impact on the final localization results. We used the value of 3 as an example in our evaluation.



(a) Rotation (b) Translational movements

Figure 5: The translational movements or rotation of a graph preserve the distance constraints between vertices.

product summation between $\{v_l\}$ and $\{v'_l\}$:

$$\Phi = argmax \sum_{l=1}^{L} v'_l v_l^T \quad (1)$$

The inner product is a monotonically decreasing function of the angle difference between two vectors. Thus Equation 1 minimizes the angle differences of L edges obtained from acoustic ranging and initial WiFi localization.

Due to ranging errors, the real orientation may differ from Φ. To ensure that the true orientation is covered, the search of $\{q_i\}$ is conducted in an orientation range of $[\Phi - \Delta\Phi, \Phi + \Delta\Phi]$, where $\Delta\Phi = 20^\circ$ from our empirical study in Section 5.5. We choose the L longest edges because the longer an edge, the less the impact on its direction caused by the same ranging error.

Step 4: Set the Search Scopes. Based on the initial WiFi position estimation p_i, the server tries to superimpose G' onto the WiFi signature map, such that each vertex i except the target is restricted inside a small circle A_i centered at p_i with radius r_i during the searching process in following step. We set r_i at $2m$ based on empirical study (see Section 5.1).

Step 5: Joint location estimation. Finally, with a small movement step of α meters and rotation step of β degrees (set to $0.1m$ and 2° based on empirical study), the server searches for the optimal location and orientation combination to superimpose the graph against the fingerprint map using the following objective function, where the moving and rotating operations are illustrated in Figure 5 (a) and (b) respectively:

$$\underset{\{q_i\},\ q_i \in S}{\arg\min} \sum_{i=0}^{M} [f(q_i) - f(p_i)]\,[f(q_i) - f(p_i)]^T \quad (2)$$

where $f(x) = [RSS_x^1\ RSS_x^2 \cdots\ RSS_x^k]$ is the WiFi fingerprint or measurement at location x (i.e., the RSS measurements of k detected access points).

Equation 2 produces the final results by finding a set of locations $\{q_i\}, i = 0, 1, ..., M$ from the WiFi fingerprint database S, so as to minimize the summation of RSS distances, each of which is between device i's WiFi measurement $f(p_i)$ and the signature $f(q_i)$ at its new location estimation q_i.

One comment we want to make is that the distances between some pairs of vertices may not determine the shape of a graph uniquely. For example, a square is *flexible* since its vertices can rotate against each other and form a family of rhombi while preserving the edge lengths, whereas the shape of a triangle is "rigid" (i.e., uniquely determined) given the lengths of the three edges. The rigid graph theory [14] describes under what conditions a graph is *rigid*: A complete graph with a distance between any two vertices is rigid. The concurrent ranging among multiple devices and the resulting pairwise distances give a complete, thus rigid graph. [4]

Extensive evaluation shows that our algorithm greatly improves the localization accuracy of the target phone (details in Section 5). The improvement is limited only under some rare cases (e.g., when peers are aligned almost on a straight line, or clustered together and located far away from the target), which we discuss in Section 5.4.1.

4. ACOUSTIC RELATIVE RANGING

Our peer-assisted approach requires acoustic ranging among multiple phones. This raises two new issues: First, how to ensure that the signals from different peer phones do not interfere. Sound signals do not carry a MAC address so there is no way to tell which phone emitted which sound. Second, the whole process has to complete in short time. Otherwise users may have moved, causing inaccurate ranging results. In this section, we describe our signal design, detection, and scheduling techniques that satisfy the requirements of concurrent multi-peer ranging.

4.1 Beep Signal Design and Detection

4.1.1 Acoustic Ranging Principle

In principle, ranging can be done by Time-of-Arrival (TOA) method that estimates the sound travel time from one device to another. The difficulty is in uncertainties: both the emitting and detecting have variable delays difficult to measure. The lack of clock synchronization between devices further adds to the problem. To address these issues, we have each involved peer device emit signals and the uncertainties will cancel out each other in calculation, similar to Beepbeep [19]. By doing so, only delay measurements from the same device are needed, thus circumventing the above issues.

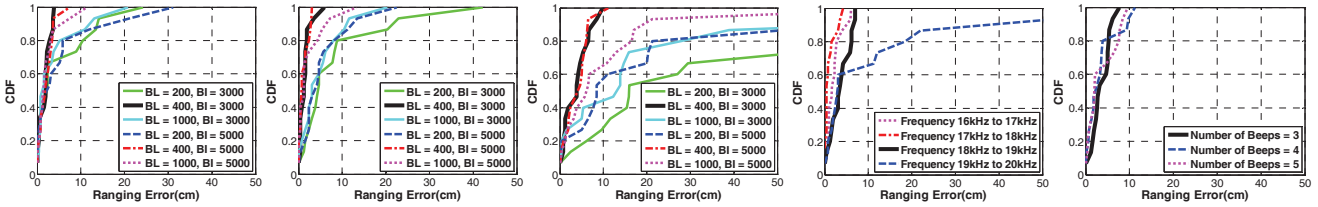
4.1.2 Beep Design

We need to design the acoustic ranging signal carefully. The signal should be robust to various background noise (e.g., human conversations, PA announcements and music) that make it hard to detect. It should have minimal disturbance to people's normal activities. Based on several considerations, we choose the frequency band between 16kHz and 20kHz for the signal.

The majority of the background noise are located at the lower frequency band (e.g., conversation between 300Hz to 3400Hz, music from 50Hz to 15kHz, which covers almost all naturally occurring sounds). 16-20kHz is still audible to human ears [11], but much less noticeable and thus present less disturbances. We also found that the current cell phone microphones are more sensitive to high frequency sounds than human ears [26]. A high frequency beep at the edge of the microphone's frequency response curve make it both easier to filter out noise and renders the signal unnoticeable to most people.

Our sound signal consists of several evenly paced beeps of equal lengths. We study how to set the number, length and frequency of the beeps, and the length of the intervals in Section 4.2.1. They all impact ranging accuracy and time. For example, too short a beep may not be picked up by the microphone, while too long a beep will add more delay to the ranging.

[4]Due to outliers in acoustic ranging, we may miss some pairwise distances with a small probability. The small search scopes set in step 4 of the algorithm help ensure the rigidity.



(a) BL and BI design, 100cm (b) BL and BI design, 200cm (c) BL and BI design, 300cm (d) Frequency design, 300cm (e) Number of beeps design, 300cm

Figure 6: Beep Design: results obtained from HTC EVO.

4.1.3 Beep Detection

Beep detection determines exactly when the beep signal arrives and it is critical to the accuracy of relative ranging. We investigate the behaviors of two most common signal detection methods, *change-point detection* and *correlation-based* under high frequency band.

Change-Point Detection Method: This method requires the beep have distinct energy uniformly distributed over a short frequency band (e.g., 16-18kHz). Given that band, this method first filters out the background noise using a Short Time Fourier Transform (SIFT). It then identifies the first strong signal that deviates from the noise in the targeted frequency band. The observed energy distributions of the recorded signal differ significantly before and after the arrival of the beep signal. Sequential change-point detection technique is adopted to identify the arrival of the signal, i.e., the exact time point of changed distribution [26].

Correlation-based Method: This method uses a chirp signal of a much shorter length. It correlates the emitted chirp sound with the recorded signal using L2-norm cross-correlation, and picks as the signal detection time when the correlation value reaches the maximum. To be robust to multipath, the earliest sharp peak in the correlation values is used as the signal detection time [19].

4.2 Detailed Design

Through extensive experiments we find that change point has much higher accuracy than correlation. We will present the signal design for change point first, then compare the two methods under different environments.

4.2.1 Sound Signal Parameters

Experimental Set Up: We used HTC EVO and Android Developer Phone 2 (ADP2) phones and 44.1 kHz sampling rate in recording. During each test, two smartphones of the same model are used. One phone emits the sound first. Upon receiving the signal, the other phone emits the sound and both phones record. The relative ranging is computed by obtaining the time difference based on the signal detection in the recorded sounds by the two phones as described in Section 4.1.1. We spaced the two phones apart at 100cm, 200cm, and 300cm respectively. Each curve in our results is obtained with 30 runs.

Beep Length (BL) and Beep Interval (BI) analysis: Figure 6 (a) - (c) presents the cumulative distribution function (CDF) of the ranging error when the beep length is set to 200, 400 and 1000 samples and the beep interval is set to 3000 and 5000 samples, respectively. The frequency band is 18kHz to 19kHz. It is clear that beep length of 400 samples achieves the best performance: the ranging error is around 10cm even when the two phones are placed 300cm away, and the error is similar with beep interval of 3000 and 5000 samples. We thus choose beep length of 400 samples and beep interval of 3000 samples in our system implementation.

Beep Frequency Band: With the above settings, we examine the performance of different frequency bands beyond 15kHz. Figure 6 (d) shows the results when varying the frequency band from 16kHz to 20kHz with two HTC EVO phones 300cm apart (better results are obtained in 100cm and 200cm cases). The ranging errors are less than 10cm when the beep frequency is below 19kHz. Similar experiments are conducted for ADP2 and we find that the highest frequency band producing comparable ranging results is 16-17kHz. We choose 16-17kHz range for ADP2 and 18-19kHz for HTC EVO in our system.

Number of Beeps (NB): Outliers can occur in ranging results due to dynamic factors in the environment. The sound signal should contain multiple beeps, so as to obtain multiple ranging estimates and filter out outliers. However, too many beep in the sound signal may make it more susceptible to multi-path distortions. Figure 6 (e) shows the ranging error when different numbers of beeps are employed under frequency band 18-19kHz with BL = 400 samples and BI = 3000 samples. Basically, comparable performance (less than 10cm) is observed across three to five beeps. In our implementation, we used three beeps, which is robust up to two outliers on opposite side of the true distance. We observe that the probability of getting reliable results is more than 95%.

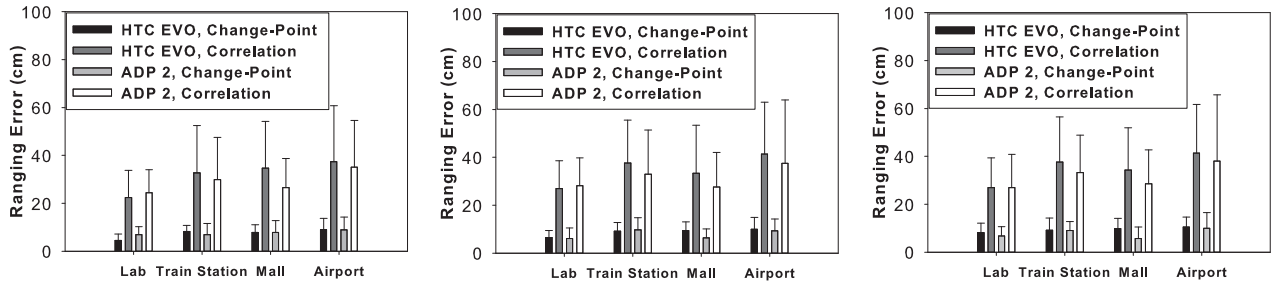
4.2.2 Robustness to Various Noises

To evaluate the robustness of the design, we conduct a comprehensive study under various environments including lab, shopping mall, train station, and airport. There are all kinds of noises during our test such as human talking, radio broadcasting, dog barking, and trolley rolling. In each scenario, two phones are placed 100cm, 200cm, and 300cm apart.

Figure 7 shows the median and 90th percentile ranging error using change-point detection method and correlation-based method respectively under high frequency band. For change-point, we use 16-17kHz for ADP2 and 18-19kHz for HTC EVO. The sound signal follows the previous parameters in Section 4.2.1. For correlation method, we use 16kHz-20kHz for both phone types and the chirp signal length is 50ms [19]. We find that change-point detection significantly outperforms correlation in all cases: it has consistent low median errors around 10cm, while that of correlation ranges over 20cm to 40cm; its 90th percentile error is always below 20cm, while that of correlation can go up to 65cm. We thus choose change-point detection method for acoustic ranging in our system.

4.3 Server Based Emission Schedule

Identify Nearby Peers. A target phone needs to find which peers are nearby to get their help. Since large errors may exist in the WiFi location estimates for the target and peers, using WiFi



(a) 100cm (b) 200cm (c) 300cm

Figure 7: Ranging errors under different environments

localization alone is not reliable. We let the target phone emit a customized sound signal containing a single beep to identify truly nearby peers. Only phones really close enough can detect the signal. Those willing to help can send their IDs to the server. The server will then instruct these peers when they should emit the ranging signal.

Beep Emission Strategy: To speed up the ranging process, multiple phones need to emit sounds in a short time. Due to the lack of synchronized clocks, this may lead to interference among beeps from different phones, thus prohibiting accurate signal detection. There are two options to avoid such interference: time-division multiplexing or frequency-division multiplexing. We have tested concurrent emission of signals of different frequency bands, but find that interference still exists due to energy leakage among nearby frequency bands.

We employ time-division multiplexing based on server scheduling. [5] We divide the time into slots of length t_s long enough for the beep emission of one phone (e.g., 14000 samples or about $0.3s$, for the previous signal parameters and a beginning padding of 3000 samples). After receiving the IDs of all phones, the server comes up with a back-to-back schedule for them. It sends the schedule to them, with the ith phone starting beeping after a delay of $T_p + i \times t_s$. T_p is some extra buffer to accommodate small variations in the reception of the schedule at different phones. It ensures that all phones have received the schedule before its execution.

Each phone records sound from others when its time slot has not come and plays the sound during its time slot. After finishing the whole schedule, all phones send their IDs and recorded files to the server, which performs signal detection and estimates distances.

5. IMPLEMENTATION AND EVALUATION

In this section, we first describe how to detect the presence of large errors in the initial WiFi localization results in Section 5.1. We present the experimental methodology, testing scenarios and evaluation metrics in Section 5.2. We implement a prototype of our proposed system and report its performance in Section 5.3, including the accuracy, localization latency, and incurred energy consumption using Android phones. To capture the statistical performance of our scheme, we apply a trace-driven approach to conduct an extensive study of the impact of the quality and quantity of peers in Section 5.4. Finally, we study the sensitivity of our algorithm to orientation estimation in Section 5.5 and show the algorithm's performance under different real-world environments, such as train station, shopping mall and airport, in Section 5.6.

5.1 Detecting the Presence of Large Errors

The system needs to detect the presence of large errors in the initial WiFi position estimation for two purposes. First, target devices need this to help decide whether they should trigger peer assistance. Second, when selecting peers, we find that only those with small initial errors can serve as reliable reference points. However, the system does not know the true location, how can it tell whether large errors exist in a position estimation? Based on the insights on large errors described in Section 2, we use 3 simple rules for reliable detection.

Rule I: Examine the probability of large errors at the initial estimated location. [6] If a location has a large probability (e.g. > 0.8) for an error beyond a threshold (e.g., 4.5m), declare the presence. This rule deals with consistent large errors at specific locations caused by permanent environmental settings (i.e., Case 1 in Section 2.3). For each location in the training data, we randomly pick 5 from the 60 samples to compute a location estimate, and repeat 400 times to compute the probability of large error occurrence.

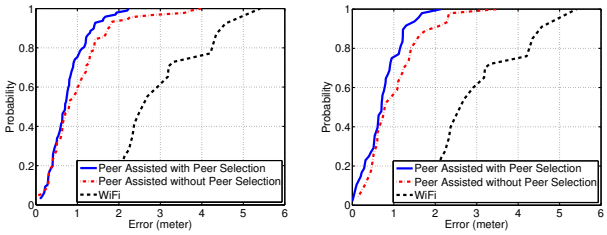
Rule II and III: Examine the two sets of APs detected in the samples and in the training data at the estimated location. If the ratio of the number of common APs to the number of all detected APs is below a certain threshold, declare the presence; Compare the standard deviations of each AP in the samples and the training data at the initial estimated position. Among all common APs, if the fraction of APs having larger standard deviations in the samples exceeds a threshold, declare the presence. These two rules deal with dynamic changes caused by transient factors (i.e., Case 2 in Section 2.3). They measure the statistical difference between the samples and the training data at the estimated location.

Intuitively, the system should be aggressive in declaring the presence for large errors, i.e., upon the detection from any of the 3 rules. Thus most targets having large errors are detected, and a peer with slight suspicion of having large errors is excluded. The cost is a few targets with smaller errors unnecessarily triggering peer assistance (but further improving accuracy), or peers with small errors excluded from assistance.

The threshold values for the three rules are set such that they minimize false classification on the training data. For example, for detecting bad targets, we can use 4.5m, 62%, 82%; for excluding bad peers, 1.5m, 90% and 42%. We find that collectively, 87% of bad targets are correctly identified, and among peers selected, 90% indeed have errors less than 1.5m.

[5] We investigate an alternative where each phone randomly choose a timeslot to emit sound, but find that much more time slots are needed and it is difficult to distinguish the signals from different phones.

[6] The nearest location in the interpolated training data set is actually used. Similar for the other two rules.



(a) 5 Peers HTC results (b) 5 peers ADP2 results

Figure 8: Localization performance for peer assisted method with and without peer selection rule

5.2 Evaluation Methodology

We conduct experiments using Android Developer Phone 2 and HTC EVO. Both phones support 44.1kHz audio sampling. HTC EVO uses 512 MB RAM and 1 GHz Qualcomm QSD8650 processor, while ADP2 192 MB RAM and slower 528MHz MSM7200A processor. [7]

We generate the sound file based on the design in Section 4.1.2, which consists of three beeps, each created by uniformly distributing white noise and then bandpass filtered to 16-17kHz for ADP2 and 18-19kHz for HTC EVO.

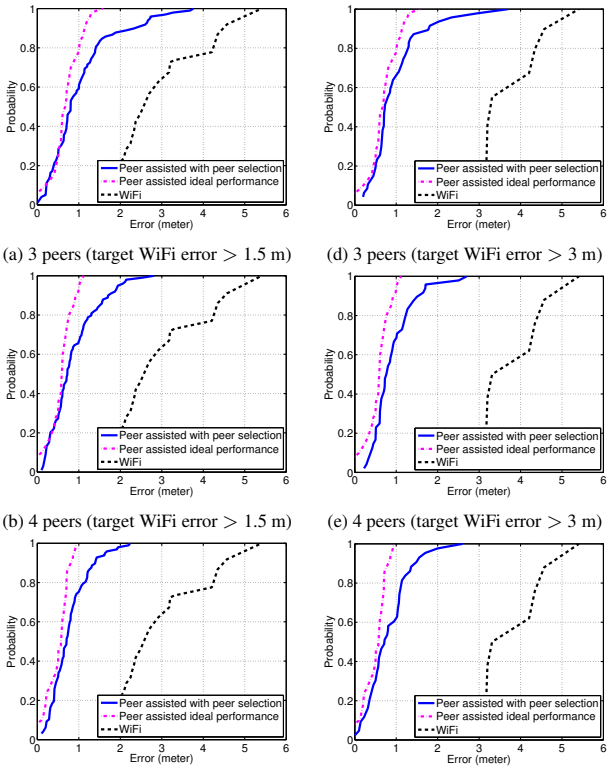
We implement a system prototype including an Android smartphone app and a backend server, both written in Java. The server runs on Lenovo Thinkpad X201 with Intel Core i5 2.53GHz processor and 4GB DDR3 RAM. The app does WiFi RSS sampling, acoustic emitting and recording. The measurements are sent to the server for processing. The app has been tested for both ADP2 and HTC EVO. Both the server and app are multi-threaded so that they can perform multiple tasks in parallel to speed up the whole process.

Experimental Scenarios. We use the prototype to validate our algorithm design and measure practical performance. In particular, we select 10 target phone locations with large errors, each with different combinations of peer locations from the office setup in Figure 1. Then we repeat the peer assisted localization for each target/peer location combination 10 times to measure the localization accuracy, total latency and energy consumption. In total we have 100 combinations of target/peer locations for the prototype experiments.

Trace-driven Statistical Performance. To capture the overall statistical performance under factors that we do not have enough manpower/device (e.g., more numbers of peers) or control (e.g., noise in different environments), we conduct trace-driven experiments. We use 200 combinations of target/peer locations in our office setup, feed the training data as WiFi samples, and distances perturbed with errors following the same distribution as results obtained in real environments (e.g., train station, shopping mall, and airport), as input to the peer-assisted algorithm. We study its the effectiveness of peer selection rules (Section 5.1), and the impact of various factors such as the number of peers, quality vs. quantity of peers, orientation estimations.

Metrics: We use the localization error to quantify the accuracy. We also measure the total time needed to complete the peer assistance process, and decompose the total time to understand the dominant factors. We measure the energy consumption using tools in [28] that is shown to achieve accuracy within 5%.

[7]We encountered some problem doing concurrent sound playing and recording on iPhone and are still investigating the issue.



(a) 3 peers (target WiFi error $>$ 1.5 m) (d) 3 peers (target WiFi error $>$ 3 m)

(b) 4 peers (target WiFi error $>$ 1.5 m) (e) 4 peers (target WiFi error $>$ 3 m)

(c) 5 peers (target WiFi error $>$ 1.5 m) (f) 5 peers (target WiFi error $>$ 3 m)

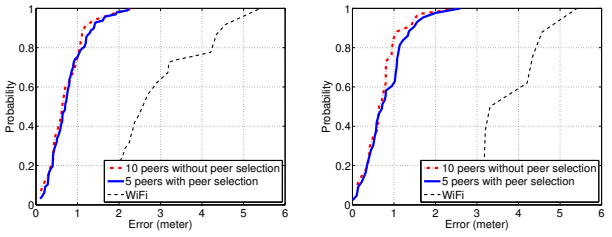
Figure 9: Peer-Assisted localization performance for target phone under different number of peers for HTC EVO. The first colomn is results when the target WiFi error is greater than 1.5m and the second column with target WiFi error greater than 3m.

5.3 Prototype performance

Localization Accuracy We evaluate the accuracy in the lab for 3 and 4 phones (2 and 3 peers) due to limited manpower and device availability. Peers have WiFi localization errors under 1.5m, and the target over 3 meters. As shown in Table 2, we can observe that the peer assisted method can reduce the maximum error from 6.7m to 4.3m and 3 m with 2 peers and 3 peers respectively. For median error, there is at least 1.5m accuracy improvement. Furthermore, more peers benefit the localization performance: 3 peers further reduces 1.3m on the maximum error compared with 2 peers.

Overall Latency and Decomposition We report the latency measurements of major components in the peer assistance process. The recruiting takes about 1s, which is independent of the number of peers involved. Then the scanning of 5 WiFi samples takes about 4.8s on phones (HTC EVO), while in parallel the beep emitting takes 1.5s (4 phones, 0.3s time slot, 1 extra buffer slot), the file uploading 1s, the TOA on server 1.2s. After WiFi and TOA results, the algorithm takes another 2s on server. In total that is 7.8s (i.e., $1 + max(4.8, 3.7) + 2$). We want to point out that we have not optimized the system yet. E.g., the algorithm is mostly matrix operation, which can speed up at least one order of magnitude using proper libraries [8]. That would reduce the algorithm time to 0.2s, making the total time down to 6s. Since most other components

[8]http://stackoverflow.com/questions/529457/performance-of-java-matrix-math-libraries shows almost 40 times speedup.



(a) target WiFi error $>$ 1.5 m (b) target WiFi error $>$ 3 m

Figure 10: Impact of peer quality vs. peer quantity.

can finish in parallel and faster than WiFi scan, our system does not pose much more latency than required in the original WiFi localization.

Energy consumption We also estimate the energy overhead using tools and methodology in [28]. The WiFi scan, sound emitting/recording and file uploading are the three major components. WiFi scan of 4.8s takes about 0.12 Joule. The beeps occupy about 0.18s in the 0.3s timeslot while the recording during the 1.5s schedule (4 phones) results in a file of 100KB. These audio parts take about 0.35 Joule. File uploading is about 2.1 Joules. In total it is 2.57 Joules, with file uploading dominating at 80%. All these happen in about 8s, translating into about 320mW additional power consumption. This is smaller than the average power of a phone (e.g., HTC Evo lasts 12.7 hours with average power of 450mW). We believe such overhead does not pose a burden for the battery life.

5.4 Overall Statistical Performance

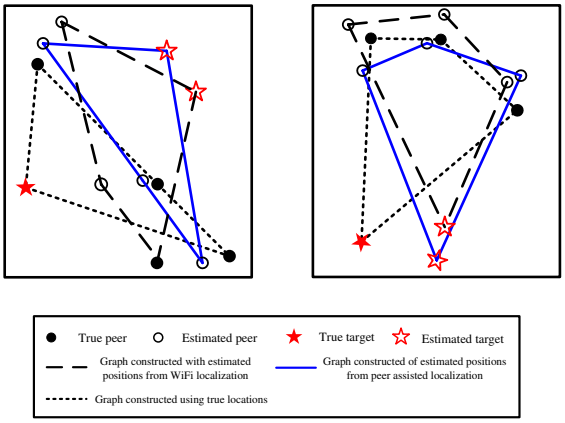
We next use trace-driven experiments by feeding the results of initial WiFi estimates and error-perturbed acoustic ranging results as inputs to capture the statistical behavior of our peer-assisted method.

Effect of Peer Selection. We first study the effect of the 3 peer selection rules (Section 5.1) on localization accuracy. Figure 8 shows the localization error CDFs of those testing points with initial WiFi localization errors exceeding 1.5m for both HTC EVO and ADP2. We observe great performance improvements with and without peer selection with CDF curves shifting to the left significantly for both phones. In particular, the median error exhibits a 75% improvement under peer assistance with and without peer selection. We find that peer selection is essential for improving localization accuracy of points with large errors: the long CDF tail has been reduced 60% from 5.5m to 2.2m. This shows that peers filtered by the rules are more reliable reference points and constitute more accurate constraints on the target location. For the rest of the paper, we only present results with peer selection.

Impact of the number of peers. Figure 9 shows the localization accuracy for the target phone when varying the number of peers at 3, 4 and 5. In our experiments, the median error of initial WiFi localization is 1.5m, which indicates that about 50% of the testing

Localization Error (meter)	Mean	Median	90%	Maximum
Peer assisted (3 peers)	1.6	1.5	2.8	3
Peer assisted (2 peers)	1.9	1.8	3.1	4.3
WiFi	3.7	3.3	5.6	6.7

Table 2: Localization accuracy of the prototype: 2 and 3 peers for a target phone with large initial errors ($>$ 3 m)



(a) Mirroring outliers (b) Rotation outliers

Figure 11: Illustration of larger errors for peer assisted localization method.

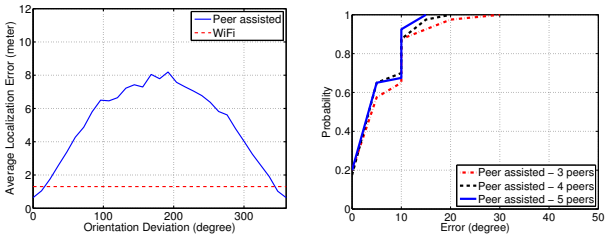
target points have small errors below 1.5m which are acceptable in most applications, while the other 50% of the target phones experiences errors beyond 1.5m. Based on this observation, we perform two sets of tests to evaluate the effectiveness of our peer-assisted approach: the first test applies to the 50% of the target phones with errors beyond 1.5m as shown in the first column of Figure 9, and the second test applies to those suffering from large errors exceeding 3m (about 30% of target phones) as shown in the second column of Figure 9.

The most encouraging result is that our approach show consistently great performance improvement for both test cases, suggesting that our algorithm is highly effective to reduce large errors. In general, more peers lead to more improvements in accuracy: When the number increases from 3 to 5 in Figure 9, the maximum error in the peer-assisted curve reduces from about 3.8m to 2.4m for both test cases, while the median error stays more or less the same around 0.7m. This is because more peers pose more constraints and less uncertainty on the location of the target. We also show the ideal performance where the true graph orientation and distances between phones are known for the algorithm. The maximum and median error are 1.5m and 0.5m. For about 80% of the case, the accuracy is very close (within 0.4m) to the ideal performance. This indicates that our orientation angle estimation is quite accurate.

Quality vs. Quantity of Peers One interesting question is how important is the quality versus the quantity of peers. Figure 10 presents the localization error CDF when using peer assisted localization with 10 peers without selection and 5 peers with selection (based on the rules designed in Section 5.1) respectively. We find that the performance of using 5 peers with selection is comparable to that of using 10 peers without selection when applying to scenarios of target phones with WiFi error greater than 1.5m, and only slightly worse (only for the bottom 40% cases and differ by about 0.2m) when the target WiFi error is greater than 3m. Thus, by utilizing peer selection only half of the peers is needed to achieve a similar performance as when adopting peers blindly. This observation strongly suggests that peer quality is more important than peer quantity and peer selection is critical.

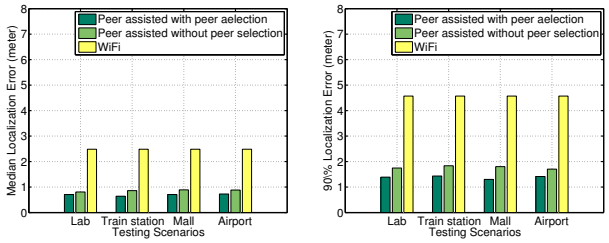
5.4.1 Insights on Remaining Errors

During our initial experiments we analyzed what happened for the points towards the tail of the CDF curve under peer assistance. We found two reasons limiting the accuracy improvement. The first



(a) (b)

Figure 12: (a) Average localization error under different orientation deviations; (b) Orientation estimation performance with different number of peers



(a) Median Error (b) 90-Percentile Error

Figure 13: Localization performance under different environments with 5 peers.

is called "mirroring" (shown in Figure 11 (a)). When the peers are almost aligned on a straight line, the constructed rigid graph could be a flipped "mirror" of the ground truth. If the mirrored graph happens to have a smaller RSS distance summation, the target would be located to the opposite side of the line. By comparing the ranging measurements, we can tell whether such alignment happens and avoid using such peer combinations. We have addressed this problem in all previous results.

The second is due to errors in orientation when the target is relatively faraway from peers (shown in Figure 11 (b)). When the peers are close to each other, a small orientation deviation can move the target far away from its true location while the peers are still estimated close to their true locations. We tried some heuristics by comparing the relative distances among peers and to the target and found encouraging preliminary improvements (maximum error reduction from 3.6m to within 2m). We are still investigating for a complete solution to this problem.

5.5 Orientation Estimation and Sensitivity

The peer-assisted localization accuracy is affected by the orientation estimation. we take a rigid graph of 4 phones constructed using their true distances to showcase the sensitivity of localization results to orientation errors. We rotate the graph so it deviates from its true orientation at a fixed degree, then we move the graph translationally to find the optimal locations. The results are shown in Figure 12 (a). We find that when the orientation deviation is larger than 24°, the average localization error becomes larger than that of initial WiFi localizations ($1.3m$ in this case); it increases sharply to 8 meters when the orientation deviation approaches 180°. We examine cases with other numbers of peers, and find that generally the orientation deviation shall not exceed 20°, otherwise the improvement will be very limited. Figure 12 (b) shows the accuracy of orientation estimation under different number of peers. We find that our orientation estimation method is indeed very effective, which can restrict the error within $\pm 20^\circ$ and $\pm 10^\circ$ around the true angle for over 98% and 90% of the testing scenarios.

5.6 Impact of Various Environments

We further study the peer-assisted localization performance for the bottom 50% target points (i.e., those with $>$ 1.5m initial errors) with 5 peers, using ranging results perturbed with errors under various environments. In Figure 13, we find that our localization performance only varies slightly when comparing real-world environmens (e.g., train station, shopping mall and airport) to the lab environment. The median errors are all around 0.7m whereas the 90% errors are about 1.4m; even without peer selection, the median and 90% errors are around 0.8m and 1.7m respectively. Compared to the target's WiFi results of median error 2.5m and 90% error 4.5m, our proposed method is quite robust to noises in different environments.

6. DISCUSSION

Peer Involvement. Since peers need to spend energy and bandwidth for helping the target, we envision that whichever application that leverages our localization solution may have certain rewards (e.g., points, virtual currency) exchanged to compensate peers that help out. Such incentives are already present in many mobile applications. In our system, peers need to run a recruiting signal detection thread to know whether they are needed. To avoid running this thread continuously, the server may send push notifications to peers in a large area around a target upon its request for help. The notification triggers the thread; if no recruiting signal is detected after a timeout threshold, a peer will stop the thread. We note that the improvement in accuracy depends on the number and quality of peers. In general, more high quality peers lead to greater improvements. But even with just a few (e.g., 3) quality peers, there is still significant reduction in error.

Movements of users. Peers might be moving during the acoustic ranging. Movements affect the accuracy only when they occur during the sound-emitting period, which happens concurrently with and takes less time than WiFi scanning (1.5s vs 4.8s in our prototype). Thus we do not pose more constraints on movements than existing WiFi methods. If the peers are static or do not move too much in this 4.8s interval, both the ranging and WiFi measurements are still accurate. In many public spaces it is common for many users to remain still for a short while (e.g., resting in airports, window-shopping in stores). Movements before and after the WiFi scanning (and beeping) do not affect the accuracy. We recognize that less peers may be suitable due to movements and plan to examine additional input such as acceleration to filter out those that have moved too much. The main latency bottleneck is the WiFi scanning time, which depends on both hardware and OS that we do not have control. Our experiments show that the acoustic ranging design is robust to noises under different environments. One issue we plan to further investigate is the effects of clothes and human body. When phones are placed in pockets, the clothes or human body may attenuate the sound signal.

Triggering peer assistance. Our work provides the technology for peer assistance. However, it is eventually up to the users to decide when they desire such help. The rules provide some hints to users about the likelihood of large errors to help them make the decision. Users may also set up certain policies about under what kinds of conditions assistance is needed. One that always requires help gets high accuracy, at the cost of paying more points or vir-

tual currency to others. In our system, more than one target phone may request assistance at the same time and the acoustic signals may collide if these targets are close to each other. Since users do not need assistance continuously, they request help only once in a while, e.g., when wondering which hallway to take to a train platform. So the likelihood of two nearby users requesting help simultaneously is small. When this happens, the server can distinguish them and provide assistance one at a time. This may delay the localization to some targets, but it still provides accurate results.

7. RELATED WORK

Smartphone indoor localization has attracted tremendous interests recently. Methods in prior work mainly focus on employing more sensing modalities, less infrastructure support, and reducing the efforts of building signal maps [2, 8, 15, 24]. SurroundSense [2] utilizes multiple sensing modalities (e.g., cameras, microphones and accelerometers) available on smartphones to perform logical localization (e.g., different stores) via ambiance fingerprinting by combining optical, acoustic, and motion attributes. WiGEM [13] proposes a learning based approach that uses the Gaussian Mixture Model (GMM) and employs Expectation Maximization (EM) to estimate the model parameters without relying on labor-intensive "training." It is robust to multiple factors (including device and power level variability, mobility, and changes of indoor spaces) that many training-based systems are susceptible to. WILL [24] aims to perform indoor logical localization without the need of building radio signature maps ahead of time. It exploits abrupt signal changes through walls and accelerometers to infer user movements and achieve room level accuracy. EZ [8] is another work that targets configuration-free indoor localization by utilizing genetic-based algorithms. Large errors still exist with 50 and 80 percentile errors at 7m and 10m, respectively. WiGEM, WILL and EZ are complementary to our system for reducing the efforts of building the radio signal map during training.

Few studies have leveraged the unique peer constraints to assist smartphone localization. Virtual Compass [5] utilizes both WiFi and Bluetooth and results in a median error of 3-4m. Other studies require special hardware or infrastructure not readily available on smartphones [17, 18]. Our work leverages abundant peers in public spaces to reduce large errors. It aims at the most prevalent WiFi infrastructure and do not require any special hardware, which is essential for easy deployment.

There have been quite some work for acoustic positioning techniques such as Cricket [20], Bat system [16], ENSBox [12], and WALRUS [6]. Recently, several proposals have studied ranging between cell phones using low frequency bands (e.g., 2-6kHz). Beepbeep [19] proposes a ToA based acoustic-ranging method between two phones. It can achieve 1 or 2cm accuracy in a range of 10m. Whistle [25] leverages multiple receivers with well-known locations to receive two sounds signals from the target. It avoids tight synchronization usually required in Time Difference of Arrival (TDoA) methods. Qiu *et al.* [22] utilizes the acoustic signal to estimate the relative position between two phones to support phone-to-phone games and apps.

Compared to the above work, we face a different challenge of fast and accurate ranging among multiple peer phones. We use server scheduling to coordinate multiple peer phones to avoid signal interference and collision for almost concurrent pairwise ranging. We propose signal design and detection methods that are lightweight in computation, robust to noisy environments, and much less perceptible to the human ear due to the adoption of high frequency band (e.g., 16-19kHz). Our system utilizes minimum auxiliary COTS sound hardware to reduce large errors incurred from general WiFi-based approaches.

For smartphone based localization using acoustic signals, Tarzia *et al.* [23] introduces a technique based on ambient sound fingerprint called Acoustic Background Spectrum. They exploit acoustic signals as fingerprints instead of measuring the ranging information between phones, and their localization granularity is at room level. Constandache *et al.* [10] deploys inaudible sound beacons randomly placed in the building, as a reference frame for correcting users' movement traces captured by the accelerometer and compass. However, this system requires an extra acoustic infrastructure and has localization errors around 8m on average. We exploit the constraints from nearby peer phones to achieve much higher localization accuracy, without the need for extra acoustic infrastructure.

8. CONCLUSION

Indoor localization on smartphones is critical to enable novel features for location based applications. However, existing approaches have yet to prove that they can satisfy what is desired in many business scenarios. Due to the prevalence of WiFi infrastructure, we set out to study the accuracy that WiFi localization can practically achieve on smartphones. We find that despite reasonable accuracies in many cases, the dynamic radio propagation poses fundamental limits and causes large errors. Inspired by the idea of relative positions of nearby peer devices as unique physical constraints on the possible location of a smartphone, we propose a peer assisted localization approach that leverages much more accurate distance estimate through acoustic ranging. Extensive experiments have demonstrated our approach successfully pushes further the limit of WiFi localization accuracy to what is empirically possible only under hundreds of APs, making WiFi a candidate for high accuracy localization. Our system does not pose much more latency than required in the original WiFi localization and has negligible impact on the battery lifetime.

9. ACKNOWLEDGMENTS

We sincerely thank our shepherd Samir R. Das and the anonymous reviewers for their invaluable feedback which helped improve this paper. This work is supported in part by the National Science Foundation Grants CNS0954020, CNS1016303, and CCF1018270.

10. REFERENCES

[1] Shopkick application. http://www.shopkick.com/.

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

[3] P. Bahl, V. Padmanabhan, and A. Balachandran. Enhancements to the RADAR User Location and Tracking System. Technical Report Technical Report MSR-TR-2000-12, Microsoft Research, February 2000.

[4] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'00)*, pages 775–784, March 2000.

[5] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. D. Corner. Virtual compass: Relative positioning to sense mobile social interactions. In *Proceedings of the 8th International Conference on Pervasive Computing (Pervasive'10)*, pages 1–21, 2010.

[6] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp. WALRUS: wireless acoustic location with room-level resolution using ultrasound. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services (MobiSys'05)*, pages 191–203, 2005.

[7] G. Chandrasekaran, M. A. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. P. Martin. Empirical evaluation of the limits on localization using signal strength. In *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*.

[8] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking (MobiCom '10)*, pages 173–184, 2010.

[9] CNN. New version of google maps brings indoor floor plans to your phone. http://www.cnn.com/2011/11/30/tech/mobile/new-version-google-maps-indoors-wired/.

[10] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see bob?: human localization using mobile phones. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking (MobiCom'10)*, pages 149–160, 2010.

[11] S. Gelfand and H. Levitt. *Hearing: An Introduction to Psychological and Physiological Acoustics*. Marcel Dekker New York, 2004.

[12] L. Girod, M. Lukac, V. Trifa, and D. Estrin. The design and implementation of a self-calibrating distributed acoustic sensing platform. In *Proceedings of the 4th international conference on Embedded networked sensor systems (Sensys '06)*, pages 71–84. ACM, 2006.

[13] A. Goswami, L. E. Ortiz, and S. R. Das. WiGEM: a learning-based approach for indoor localization. In *Proceedings of the 7th International Conference on emerging Networking EXperiments and Technologies*, 2011.

[14] J. Graver, B. Servatius, and H. Servatius. *Combinatorial rigidity*. Graduate studies in mathematics. American Mathematical Society, 1993.

[15] Y. Haibo, G. Tao, Z. Xiaorui, X. Jingwei, T. Xianping, L. Jian, and J. Ning. Ftrack: Infrastructure-free floor localization via mobile phone sensing. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'12)*, 2012.

[16] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (Mobicom '99)*, pages 59–68, 1999.

[17] M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for co-located mobile devices. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services (MobiSys'05)*, pages 177–190, 2005.

[18] M. Minami, Y. Fukuju, K. Hirasawa, and S. Yokoyama. DOLPHIN: a practical approach for implementing a fully distributed indoor ultrasonic positioning system. *Proceedings of 6th international conference on Ubiquitous Computing (UbiComp'04)*, 2004.

[19] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: A high accuracy acoustic ranging system using cots mobile devices. In *In Proceedings of the 5th international conference on Embedded networked sensor systems (Sensys'07)*, pages 1–14, 2007.

[20] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom'00)*, pages 32–43, Aug 2000.

[21] A. Prorok, P. Tome, and M. Alcherio. Accommodation of NLOS for Ultra-Wideband TDOA Localization in Single- and Multi-Robot Systems. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[22] J. Qiu, D. Chu, X. Meng, and T. Moscibroda. On the feasibility of real-time phone-to-phone 3d localization. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys'11)*, pages 190–203, 2011.

[23] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th international conference on Mobile systems, applications, and services (MobiSys'11)*, pages 155–168, 2011.

[24] C. Wu, Z. Yang, Y. Liu, and W. Xi. WILL: Wireless Indoor Localization Without Site Survey. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'12)*, pages 64–72, 2012.

[25] B. Xu, R. Yu, G. Sun, and Z. Yang. Whistle: Synchronization-free tdoa for localization. In *Proceedings of the 2011 11st International Conference on Distributed Computing Systems (ICDCS'11)*, pages 760–769, 2011.

[26] J. Yang, S. Sidhom, G. Chandrasekaran, V. Tam, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin. Detecting driver phone use leveraging car speakers. In *Proceedings of The 17th Annual International Conference on Mobile Computing and Networking (MobiCom'11)*, pages 97–108, 2011.

[27] M. Youssef and A. Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services (MobiSys'05)*, pages 205–218, 2005.

[28] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*.

[29] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng. I am the antenna: Accurate outdoor ap location using smartphones. In *Proceedings of The 17th Annual International Conference on Mobile Computing and Networking (MobiCom'11)*, pages 109–120, 2011.