SpyLoc: A Light Weight Localization System for Smartphones

Mostafa Uddin
Old Dominion University
muddin@cs.odu.edu

Tamer Nadeem
Old Dominion University
nadeem@cs.odu.edu

***Abstract*—In this paper, we design, implement and evaluate the SpyLoc localization system. The design goal of SpyLoc is to develop a light weight and high accuracy localization system for off-the-shelf smartphones. SpyLoc leverages both the acoustic interface (microphone/speaker) and the Wi-Fi interface at the kernel-level of smartphones as well as the inertial sensors in smartphones to achieve high localization accuracy. SpyLoc does not require any central controller unit nor any collaboration with nearby devices. Furthermore, in SpyLoc, each user's smartphone works autonomously to estimate its location. We implement and evaluated the complete SpyLoc using commercial off-the-shelf smartphones. Our result shows that SpyLoc can achieve less then 1 meter accuracy for more than 90% of the time for both indoor and outdoor environments.**

I. Introduction

In many indoor environments (e.g., airport terminal, railway station, shopping mall, and office building), knowing the location of the user would enable several interesting application and services. For example, accurate indoors guidance, efficient network management, generation of safety alerts, access to merchandise and promotion information, analyzing the popularity of different section in the store, movement of the passenger etc. Numerous previous works have been done in the area of indoor localization. Most of the localization works have been based on Radio Frequency (RF)-based techniques that leverages signal strength of RF signal from different nearby RF sources or infrastructures (e.g., Wi-Fi access point, cellular tower). Recent works show that the existence of same signatures or fingerprints of RF signals at different distinct locations prevents from achieving high accuracy localization system [1], [2], [3] based on RF-fingerprint. Researchers have tried to improve the localization accuracy by leveraging advanced PHY layer information (i.e. channel state information) from the Wi-Fi chipset. However, such PHY layer information is not easily accessible in typical devices such as off-the-shelf smartphones.

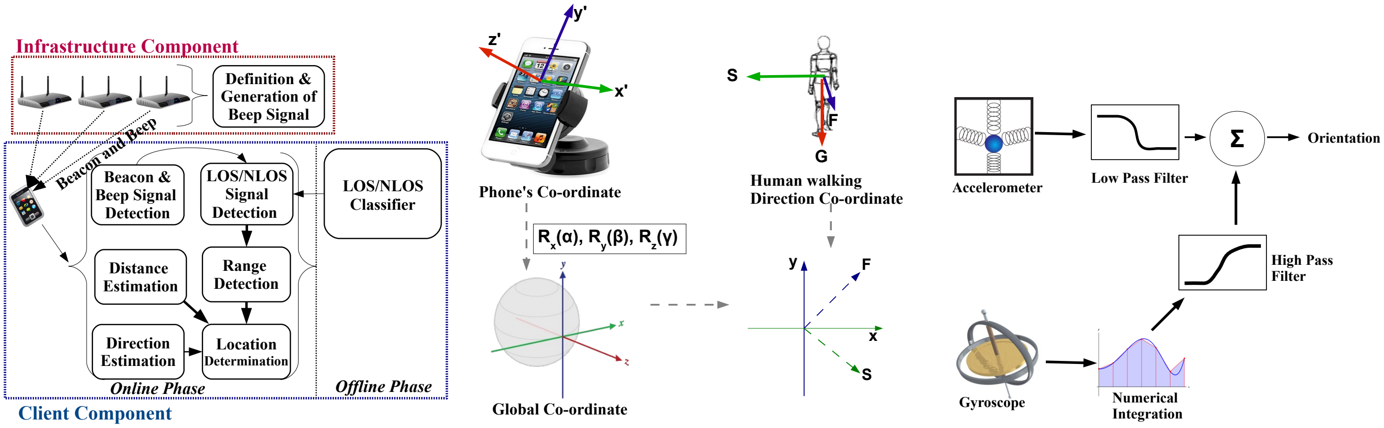
Nowadays, researchers are more inclined on developing indoor localization system for off-the-shelf smartphones because of the enormous popularity of location based applications [1], [2], [4], [5], [3], [6], [7]. Recently, many researchers are trying to combine multiple modalities such as sound and inertia sensors of the smartphones with the Wi-Fi to achieve higher accuracy localization system [1], [2], [8], [9], [4], [5], [3], [6], [10], [11]. For example, localization schemes in[1], [2], [8], [9] utilize the acoustic based ranging scheme and combine it with the RF at the application layer. Other approaches, which utilize existing inertial sensors in off-the-shelf smartphones [12], [13], highly depend on the knowledge of the environment. For example, the proposed system in [12] highly depends on the layout of the building, which is not practical in large open public spaces (e.g. airport, metro station, shopping mall).

978-1-4799-4657-0/14/$31.00 © 2014 IEEE

In this paper, we design, develop and evaluate a light-weight high-accuracy indoor/outdoor localization system (*Spy-Loc*) for off-the-shelf smartphones. SpyLoc leverages both: 1) the integration of the acoustic interface (microphone/speaker) and the Wi-Fi interface at the kernel-level of smartphones, and 2) the inertial sensors in smartphones in order to achieve high localization accuracy. The proposed system uses a combination of both ranging-based and dead reckoning based approaches.

In the ranging-based approach, similar to the well known Cricket [8], SpyLoc utilizes the difference in arrival times of concurrent transmissions of radio and acoustic signals at the target device to infer the distance. Unlike Cricket, which was designed with special hardware that is not applicable to smartphones, SpyLoc uses our developed ranging scheme RF-Beep [14] that is designed and developed for smartphones. Other previous acoustic-based range estimation works require two-way communications between smartphones[15], [2], [1]. This two-way communications are needed to eliminate the time synchronization requirement, and the sound generation uncertainties in smartphones. However, these localization schemes require centralized system for smartphones collaboration that incur high overhead. Unlike these works, our ranging scheme RF-Beep, integrates the audio interface and the Wi-Fi interface at the kernel space to eliminate the uncertainties of sound generation in smartphones. In addition, RF-Beep scheme leverages this kernel level integration to achieve the time synchronization requirement using only one-way communication.

In the dead reckoning approach, we use the inertial sensors to detect steps and moving direction to estimate traveled distance by the user. Most recent step detection techniques used in previous work apply Dynamic Time Wrapping (DTW) algorithm [16] to find out the step pattern from inertial sensors raw data. The advantage of using DTW is that it can detect the step pattern regardless of how fast the user is walking. Note that, user's walking speed is not always the same over time. However, in previous works, the step detection techniques use a fixed time window of samples to estimate the step pattern. In addition, none of these schemes have any analytic explanation about the selection of such fixed window of samples other than empirical observations for individuals. Unlike the previous works, in this paper, we select the window size based on the step model analytics of human walking. In addition, we also propose an adaptable shifting window algorithm that addresses the challenge of different walking speeds of different individuals.



(a) (b) (c)

Fig. 1: a) Architecture overview of the different components of SpyLoc localization system. b) Phone, user's walking and global coordinate systems and relations between them. c) Sensor fusion block diagram for direction estimation.

Detecting the moving direction, regardless of the orientation and the position of the smartphone (i.e. shirt pocket, pant pocket, in bag, in user's hand, attach to belt etc.), is a challenging task. Given the inaccuracy of compass/magnetic sensor in smartphones in indoor environments due to the surrounding ferromagnetic devices [17], and the overhead incurred in the frequent calibration of this sensor, we develop in this paper a technique to find the relative rotation/turn of the user based on the inertial sensors only rather than using the compass/magnetic sensor.

SpyLoc leverages the benefits of both the dead reckoning and the ranging schemes to build a practical localization system. Given the accumulative errors of the inertial sensors to track user's movement over the time, SpyLoc uses the RF-Beep ranging scheme to calibrate this error in order to improve the localization accuracy. Unlike ranging-based or RF-based localization schemes that require multiple reference points (e.g., access points), using the dead reckoning in SpyLoc reduces the number of needed reference points to at least one reference point to locate and track user's movement accurately. This low dependency on ranging scheme and the elimination of any calibration make SpyLoc a light-weight system and practically suitable for mobile users.

From experiments, Non Line-of-Sight (NLoS) scenarios, where sound sources are blocked from the smartphone by obstacles, degrade the accuracy of RF-Beep ranging scheme. However, we show how can we utilize the unique acoustic features to differentiate between the Line-of-Sight (LoS) and the NLoS acoustic signals. To the best of our knowledge, this sound signal classification for both LoS and NLoS scenarios is a new contribution for localization systems.

We summarize our contribution in this paper as follow:

- Implementation of SpyLoc; a light-weight high-accuracy localization system using off-the-shelf smartphones.
- Evaluation of SpyLoc under several real scenarios and different mobility conditions.
- Development of a stride detection algorithm to efficiently detect the users stride using inertial sensors. Given different users have different strides and speeds, the proposed algorithm is adaptable to detect and estimate different stride lengths corresponding to different users and different speeds.
- Development of a robust direction change detection algorithm that infer user's relative direction change by fusing inertial sensors.
- Detailed study of the Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) acoustic signals and development of a classification method to differentiate between LoS and NLoS signals.

The paper is structured as follows. We introduce SpyLoc with brief discussion on each component in the system in Section 2. In Sections 3-6, we describe in detail the different modules that construct client components of SpyLoc. We evaluate the performance of the system under different indoor/outdoor scenarios in Section 7. Then we discuss the related works in Section 8 and conclude with Section 9.

II. SpyLoc Localization System

Figure 1a shows the different modules of SpyLoc system. Similar to many localization systems, SpyLoc consists of two main components: infrastructure component that runs on infrastructure hardware, and client component that runs on the user's device (i.e. smartphone). In this paper, we refer to an infrastructure device running the SpyLoc infrastructure component by a *beacon device*. The beacon device periodically broadcasts a RF message (i.e. Wi-Fi beacon frame), which we refer to as a *beacon*. In addition, the beacon device also generates an acoustic signal, *beep* following each broadcasted beacon message. In a practical scenario, a beacon device could be an Access Point (AP) with additional acoustic interface (i.e. speaker, mic and sound driver). A user's smartphone running SpyLoc client component (i.e., SpyLoc application) will capture the beacon messages and the corresponding beep signals from the surrounding beacon devices. Using the captured beacon messages and the corresponding beep signals in addition to the inertial sensors in smartphone, SpyLoc application will infer the user's location.

In a typical usage scenario, when the user starts the SpyLoc application in the smartphone, it will initially collect the beep signals from the surrounding beacon devices along their corresponding beacon messages. Each captured beep signal at user's smartphone will be classified either as a Line-of-Sight (LoS) signal or as a Non Line-of-Sight (NLoS) signal with respect to its corresponding beacon device. Only for the LoS signals, the application determines the relative ranges between the user's smartphone and the corresponding beacon devices. Then, the application combines three estimated ranges from three different beacon devices to estimate the user initial location. After fixing the user initial location, SpyLoc uses the inertial sensors to detect the distance and the direction to track the next user's locations. On periodic bases, SpyLoc uses an estimated range to a single LoS beacon device to calibrate and calculate the accurate location of the user.

In SpyLoc, user smartphone calculates the location locally and no collaboration with neighboring devices is required. SpyLoc only requires one-way transmission; the transmission of the beacon messages and the beep signals by the beacon devices. Hence, user application does not require sharing any sound signal or Wi-Fi information with any nearby device or access point. This enables to preserve and protect the user security and privacy, and makes SpyLoc energy efficient application for smartphones.

In the following subsection, we provide a brief overview of the infrastructure component and the client component.

A. Infrastructure Component

The beacon device (e.g. Wi-Fi AP), which runs the infrastructure component of SpyLoc, periodically generates a RF beacon message followed by a beep signal. A single frequency sinusoidal acoustic signal defines the basic sound that we refer to as a *tone* in the paper. A mixture of tones (i.e., set of frequencies) defines a beep signal. Given the typical human hearing perception diminishes after 18kHz, we utilize the 18kHz-21kHz audio frequency range that is perceptible to the most of the off-the-shelf smartphones [18]. From experiments, we found that if the frequency space between two adjacent tones is 250Hz, then it is sufficient to avoid the interference and be able to detect the individual tones at the user side. Therefore, we the selection of 10 tones (i.e., frequencies) $(f_1, f_2, ..., f_{10})$ from 18kHz-21kHz audio range, we have up to 2^{10} unique beep signals. One of the major challenge in the infrastructure component is, *How to uniquely and autonomously define the tones of the beep signal for each beacon device?*

In selecting the tones, we utilize the least ten bits of the Wi-Fi MAC address the beacon device. Each bit position within this ten bits sequence is corresponding to a unique tone from the possible tones $\{f_0, f_1, ..., f_9\}$ (i.e. the $0th$ bit map to f_0, $1th$ bit map to f_1, and so on). A value of 1 in a bit position indicates that the corresponding tone exists in the beep signal and vice versa for the value of 0. For example, if the MAC address of a beacon device is $C4 : 2C : 03 : 3A : 2C : A1$ that has least ten bits as 0010100001, then the selected tones of the beep signal for that beacon device would be $\{f_0, f_5, f_7\}$. The beep assignment mechanism guarantees a very low probability of duplicate assignment. For example, with ten beacon devices within the proximity of each other, the probability of having two or more beacon devices with the same beep is about 4% ($\sum_{i=0}^{i=9}(2^{10} - i)/\sum_{i=0}^{i=9} 2^{10}$).

In our implementation, we utilize the Wi-Fi beacon frame as SpyLoc beacon message. We assume that each beacon device could add its location information in the payload of the Wi-Fi beacon frame. Thus, SpyLoc client application could be pre-configured with the locations of the beacon devices, or learn the beacon devices locations from their beacon messages. In section 5, we describe more details about the beep signal detection technique, and how to map the beep signal to its corresponding beacon frame. In our implementation, we use Nokia smartphone (i.e. Nokia N900) with an external speaker (i.e. Nokai MD-11) as a beacon device.

B. Client Component

In SpyLoc, client component consists of five main modules shown in Figure 1a and described in the following:

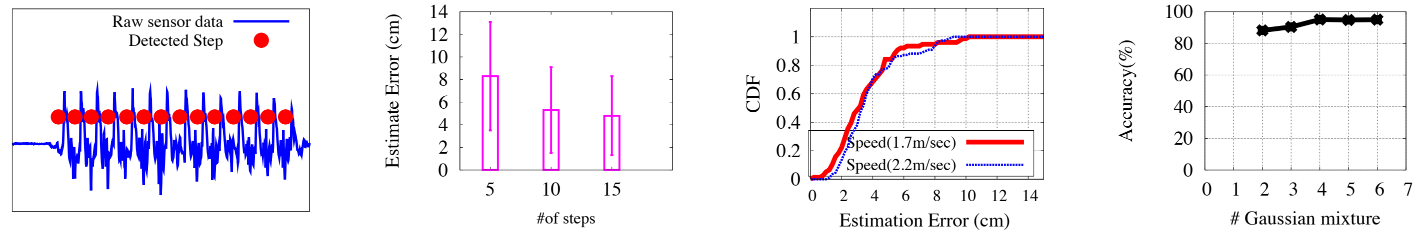
Distance Estimation module: Distance estimation module estimates the distance at each step of the user. In order to estimate the distance, this module applies an adaptable step detection algorithm. Then, it utilizes a personal step model to infer the user's step length. SpyLoc client application utilizes both the ranging scheme and the step detection algorithm to build user's step model online.

Direction Estimation module: This module estimates the user's change of direction with respect to the human's moving direction. First, It fuses the multiple inertial sensors such as the accelerometer sensor and the gyroscope sensor to find out the orientation of the phone. Second, it uses the Principal Component Analysis [19] technique to infer the direction of the user moving direction. In sections 3 and 4 we describe both the distance and the direction estimation modules respectively in more details.

Non Line-of-Sight (NLoS)/ Line-of-Sight (LoS) Detection module: The NLoS/LoS detection module consists of a classifier and feature extraction components. We use a binary classification model through an offline training to classify and detect whether a received beep signal is LoS (or NLoS). Once the SpyLoc application on the user's smartphone receives a beep signal, it uses the feature extraction component to extract the corresponding features from the beep signal and then detects whether the beacon device corresponding to a received beacon is in the LoS (or in the NLoS) of the user's smartphone.

Range Detection module: For a LoS beep signal, range detection module is responsible to estimate the user's distance to the corresponding beacon device. This module uses the Time Difference of Arrival (TDOA) of both the beacon message and the corresponding beep signal to estimate the distance between the user smartphone and the beacon device. This range detection module is built based on our previous work RF-Beep [14].

Location Determination module: The Location determination module calculates user location based on the outcomes of the range detection, distance estimation, and direction estimation modules. Knowing the current location of the user, this module estimates the next possible locations using the distance and direction estimation modules. However, at that time, If the range detection module provides the relative ranging of the user smartphone for at least one beacon device (i.e., when the user is in LoS with at least one beacon device), then this module utilizes the estimated range to calibrate user location. Under certain conditions (e.g., locate user initial position, and building step model) the range detection module utilizes the relative ranging of the user smartphone from at least three beacon devices (i.e., when the user is in LoS with



(a) (b) (c) (d)

Fig. 2: (a) Step detection in raw sensing data. (b) Accuracy of the step model in estimation average step length. (c) Accuracy of step length detect for different speed of walking. (d) Classification Accuracy of detecting LoS/NLoS.

at least three beacon devices) to apply triangulation technique to determine user location.

In the following sections, we describe each of the client component modules in more details.

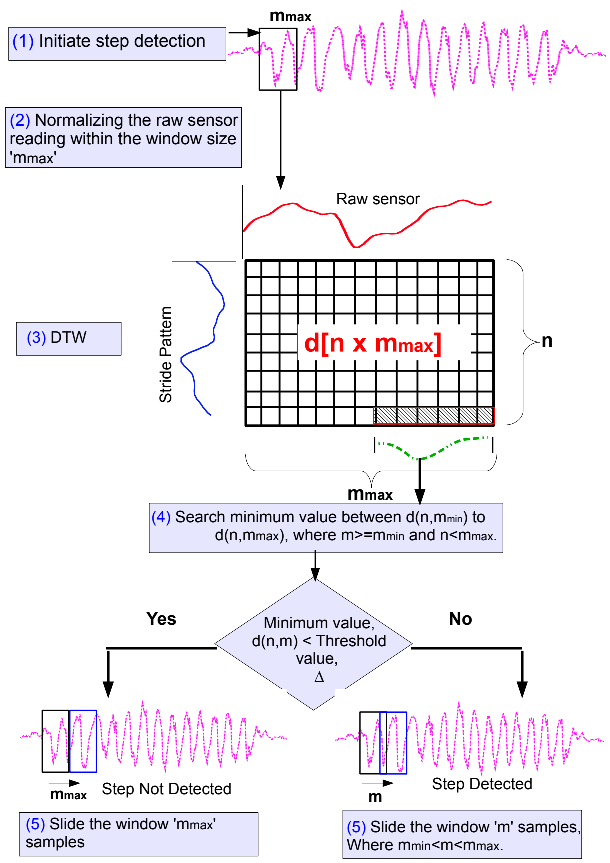


Fig. 3: Adaptable step detection module in SpyLoc localization system.

III. Distance Estimation Module

This module has two submodules i) *Step Detector*, and ii) *Personalized Step Model*

i) Step Detector: In step detection we use the commonly used accelerometer and the gyroscope sensors of the smartphone to detect and track user steps. Figure 3, shows the details of our adaptive step detection algorithm. In SpyLoc, step detection is activated only when the user moves. In doing this, we utilize the change in gyroscope sensor reading above certain threshold as an indication of movement and initiates the step detection algorithm to start capturing the accelerometer data. In our implementation, we set this threshold value to 0.3. Given the stride length (step size) is proportional to the walking speed [20], we use m_{max} parameter to represent the maximum length of a person step in terms of number of samples and defined as follows, $m_{max} = \frac{s_{max}}{v_{max}} \times f_a$, where s_{max} is the maximum length of a person's step, v_{max} is the maximum walking speed of person, and f_a is the collection frequency of accelerometer samples from the smartphone. Similarly, m_{min} parameter is defined as the minimum length of a person step. We use the $s_{max}, v_{max}, s_{min}$, and v_{min} values defined in [20] in our step detector algorithm. Since we use $f_a = 50$ samples/sec in our implementation, the corresponding m_{max} and m_{min} are 65 and 25 respectively.

After collecting m_{max} raw 3-axis accelerometer samples, we calculate the vector magnitude of each 3-axis accelerometer sample in order to make our algorithm independent of the user orientation. Similar to other algorithms [16], [12], [13], we apply the Finite Impulse Response (FIR) low pass filter to reduce the impact of the noise in the m_{max} samples. Then, we normalize the m_{max} samples before feeding it to the Dynamic Time Wrapping (DTW) algorithm. The DTW algorithm compares the similarity between the predefined step pattern (with size of n samples where $n \leq m_{max}$) and the captured m_{max} samples to detect whether a step exists within the m_{max} samples. Unlike correlation and threshold-based methods [12], [13], DTW adaptively detect user step regardless of the different lengths corresponding to different walking speeds. In our implementation, we used the predefined step pattern of size $n = 45$ samples.

The DTW algorithm calculates $d[n \times m_{max}]$ matrix scores with positive values. The lower score of $d[i, j]$ indicates a better matching between predefined stride pattern of size i samples and the captured samples of size j. Unlike common use of the DTW algorithm [16], we search for a cell $d[n, m]$ with the minimum value between $d[n, m_{min}]$ and $d[n, m_{max}]$ cells (the minimum value at the green dotted curve In the Figure 3). If this minimum value is below a certain threshold Δ, then a step length of m samples is detected. Otherwise, there is no step detected within the captured m_{max} sample. By conducting several experiments, we set the threshold Δ to 0.4 in our implementation. If a step is detected, then we shift the searching window for detecting the next step by m samples. Otherwise we shift it by m_{max} samples. Figure 3 shows how accurately the detected step by our scheme matches the actual accelerometers samples corresponding to the actual user step in a walking experiment.

ii) Personalized Step Model: We use the commonly used following step length model [16], [12] as our personalized step model, $s = a \times f + b$, where s is the step length, f is the frequency of steps, and a, b are the person-dependent constants. In order to define the personalized step model, we have to calculate the constant parameters a, b for each user. SpyLoc utilizes the ranging scheme to track the user consecutive locations using at least three ranges from three different beacon devices. Then the system matches those locations with the step counting to build the step model of the user using line-fitting algorithm.

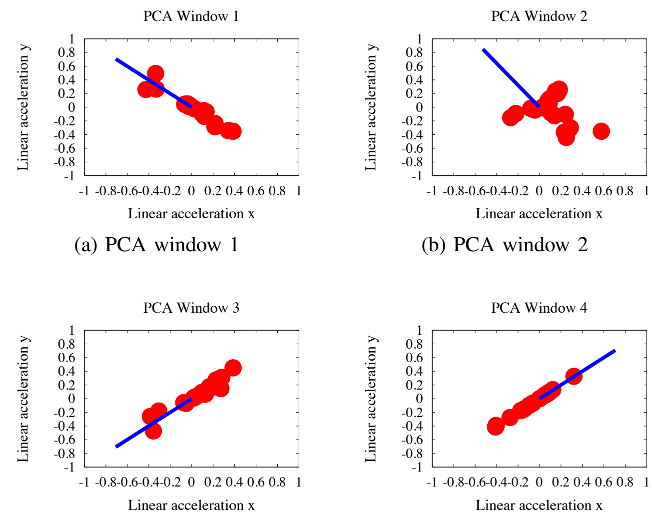
iii) Evaluation: In figure 2b, we plot the estimated step length error by building the step model from 5,10, and 15 detected steps. Increasing the number of steps to build the model reduces the overall error of estimating the step's length. In figure 2c, we also evaluate our adaptive step length estimation technique for two different speeds. We use speeds of 1.7 m/sec and the 2.2 m/sec as the normal and the fast walking speed of a person respectively. In both speeds, we found almost similar distribution of estimation error. In figure 2c, about 90% of the estimation error is less then 6cm for both speeds.

IV. Direction Estimation Module

In a practical environment, the smartphone could have any arbitrary orientation with respect to the user direction of movement. Therefore, It is a challenging task to determine the changes in user direction using the smartphone's sensors reading. In order to address this challenge, we consider three different coordinate systems shown in Figure 1b: phone coordinate system, user's walking coordinate system, and global coordinate system. While the user's walking coordinate system represents the forward direction, side, and gravity, the global coordinate system represents the north pole, the east and the gravity of the earth. In addition, the global coordinate system is a fixed coordinate system, while the other two coordinate systems are not fixed. For example, the phone coordinate varies with the phone orientation, while the user coordinate changes with the change in moving direction. Thus our idea is to map both the phone and the user's walking coordinates to the global coordinate. In phone coordinate, we need to determine the three rotation (orientation) angles $(\alpha_x, \beta_y, \gamma_z)$ around the three axis to transform the phone coordinate to the global coordinate.

In determining the phone orientation, we use the accelerometer and the gyroscope sensors. We avoid the magnetic field/compass sensor due to its high sensitivity to the surrounding magnetic devices. Figure 1c shows the block diagram of the sensor fusion technique we use to determine the phone orientation with respect to the global coordinate system.

In user's walking coordinate, the gravity (G) axis is the same as the -z axis in the global coordinate. Moreover, the other two axises (the forward direction (F), and the side (S) direction) are in the x, y plane of the global coordinate. Note that, the linear accelerometer readings from the smartphone are in respect to the phone' coordinate. Therefore we use the estimated orientation $\alpha_x, \beta_y, \gamma_z$ to transform the linear acceleration readings of the smartphone to global coordinate. Now, if we plot the linear acceleration readings in the x, y plane of the global coordinate, then the highest variation of changes of the projected readings will indicate the user's walking direction (F axis). We apply the Principal Component



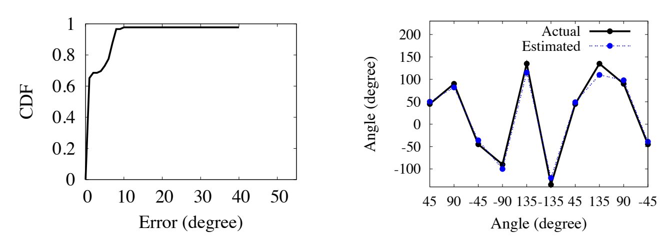
(a) PCA window 1 (b) PCA window 2

(c) PCA window 3 (d) PCA window 4

Fig. 4: Sequence of non-overlapping PCA windows. The blue line represents the PCA axis.

Analysis (PCA) [19] analysis on the transformed samples to find out the direction of the F axis in the x, y plane. In our implementation, we use 25 samples as the PCA window where 33.36 is our sampling rate per second. Figure 4 shows four sequential PCA windows where the user took a 90 degree turn. The dots in the plots are the transformed linear acceleration samples in the x, y plane of the global co-ordinates. The straight line in the plot represents the PCA axis which is the user's walking direction. In Figure 4, the PCA windows 2 and 3 shows the transition of the user's 90 degree turn.

Evaluation & Discussion: In Figure 5a, we evaluate the stability of our direction estimation technique while the user is walking straight. In a straight walk, the *cdf* plot shows that 70% of estimated direction changes are less then couple of degrees, while 98% of the values are less than ten degrees. In Figure 5b, we evaluate our direction estimation technique by plotting the estimated direction changes when the user takes a sequence of direction changes. Note that, we use the right and the left turns as positive and negative direction changes respectively.



(a) (b)

Fig. 5: (a) CDF of direction estimation errors while user is walking straight. (b) Evaluating different degrees of relative direction changes.

V. NLoS/LoS Detection Module

In this section, we start with describing the scheme a SpyLoc client uses to detect the beep signal from the captured

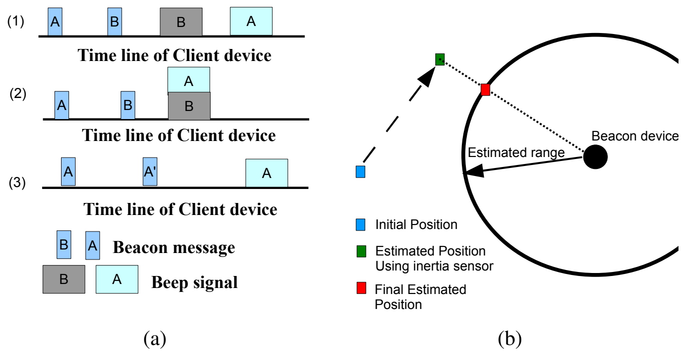


Fig. 6: (a) Timeline of the received beacon messages and beep signals at client device. (b) Calculating the user position using single range estimation.

signal. Then, we analyze some common wavelet patterns of the beep signal and describe some anomalies that we have observed under different LoS/NLoS conditions. Finally, we describe the binary classification model (LoS/NLoS) we apply on beep signals.

A. *Beep Signal Detection*

Precise detection of a beep signal at the receiving side is crucial in estimating the accurate range between the user and a beacon device. We use our developed method described in RF-Beep [14] to detect the different tones in a beep signal. In addition, correlating the beep signal to its corresponding beacon frame from a particular beacon device is also challenging. This challenge has also been addressed in Cricket [8] system, where they fundamentally transmit a customized RF frame for a long duration to overlap the audio tone transmission. However, such solution is not practically adaptable with the existing RF infrastructure (e.g. Enterprise Wi-Fi Network). In our system, SpyLoc exploits the existing RF infrastructure (i.e. WLAN) without disturbing its typical operation. For example, we have described in section II-A how we utilize the periodic Wi-Fi beacon frames from Wi-Fi APs as our beacon messages. Therefore, in SpyLoc we have to the address the correlation challenge in a different way without disturbing the regular communications of the existing RF infrastructure.

Figure 6a shows three challenge scenarios of mapping the received beep signal to its corresponding beacon message. Considering the first scenario, where a client device receives a beacon message from beacon device ’A’ followed by a beacon message from beacon device ’B’. Thereafter, the client device receives a beep signal from device ’B’ before receiving a beep signal from device ’A’. In such scenario, in order to prevent the ambiguity of mapping beep signals to the correct beacon messages, beep signals from specific beacon device need to be uniquely identified. In section II-A, we described how uniquely beep signals could be constructed for each individual beacons device in order to resolve the described scenario. In the second scenario, a client device receives both beep signals from two beacon devices ’A’ and ’B’ at same time. Given the clients knows the tones of the beep signal of each beacon device either be pre-configured or by including the information in the beacon messages, the client will be able to distinguishable between the two beep signals if each beep signal has unique non-overlapping tones. Otherwise, the client will only detect the beep signal with the dominant set of tones. Therefore, in this scenario the device will be able to correctly map at least one (if not both) of the beep signals to its corresponding beacon device. In the third scenario, a client device receives another beacon message before receiving the beep signal of the previous beacon message. Given a typical AP is transmitting Wi-Fi beacon (i.e., beacon message) every 100ms and given the acoustic signal speed in the air, the third scenario will be infeasible if we the distance between a beacon device and a client is within 35 meters. Studies show that the typical range of a Wi-Fi AP at indoor environment is less than 35 meters [3].

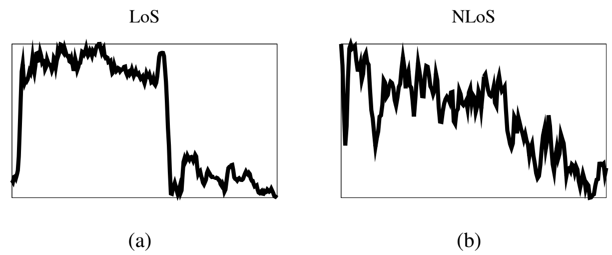


Fig. 7: a) The exponential slope of reverberation decay for LoS scenarios. b) The linear slope of reverberation decay for NLoS scenarios.

B. *Beep Signal under LoS/NLoS Conditions*

In this subsection, we describe the features used to classify and detect whether the captured beep signal is corresponding to a LoS (or a NLoS) scenario. In SpyLoc, we use the reverberation time [21] to classify between the LoS and the NLoS beep signals. Reverberation is the collection of reflected sounds from the surrounding surfaces/walls following the direct sound. The reverberation time is defined as the time taken by the reflected sound energy to decay by 60dB after the direct sound source stops. From experiments, we found the decay in the energy/intensity of the reverberation for LoS scenarios is exponential right after the direct sound stops (Figure 7b). On the other hand, the reverberation decay is almost linear in NLoS scenarios (figure 7b). We leverage this characteristic to build our classification model based on the features extracted from reverberation decay time.

C. *Classification of LoS/NLoS Signals*

This section describes both the training and testing phase of the classification model we developed to detect whether the beep signal is corresponding to a LoS (or a NLoS) scenario. Given the reverberation decay does not always reduce to 60dB due to background noise, we use the set of times needed to decay by 5dB, 10dB, 15dB, 20dB, 25dB, and 30dB as our features (experiments show that up to 30dB decay is enough to differentiate between exponential or linear decays). We labeled those times as $T_5, T_{10}, T_{15}, T_{20}, T_{25}$, and T_{30} respectively. In training the classification model, we apply the following:

Using the training dataset, we extract the feature set $F = \{T_5, T_{10}, T_{15}, T_{20}, T_{25}, T_{30}\}$ for each beep signal in the dataset. We use a naive Bayes Classifier [22] model with equal prior probability to classify between LoS and NLoS beep signals. The following probabilistic model is used as our naive Bayes classifier,

$$P(C|F) = \frac{P(C)P(F|C)}{P(F)}$$
$$where \quad P(C|F) \propto P(F|C) \tag{1}$$

where $C = \{C_1 \equiv LoS, C_2 \equiv NLoS\}$. We use m - Gaussian Mixture Model to build our two likelihood function $P(F|C_1)$ and $P(F|C_2)$. Then we apply the *Expected-Maximization*(EM) [23] technique to estimate the parameters of these two likelihood function from our training data set. Finally we use the classification model in Equation 2 to classify the extracted features F_t from the testing dataset,

$$classify\{C_1, C_2\} = \max_{c} P(F_t|C = c) \quad (2)$$

The naive Bayes model (Equation 2) basically performs binary classification to detect whether the received beep signal is corresponding to LoS (or NLoS) scenario. We experimented with different number of gaussian mixture models to verify the performance of our Bayes model as shown in Figure 2d. We observe that Gaussian Mixture Model with 4 mixtures is enough to achieve 96% accuracy. We collected 200 LoS and 100 NLoS beep signals for our dataset. We used 10-fold cross validation over the dataset to calculate the overall performance (Figure 2d).

VI. Location Determination Module

The location determination module is responsible to determine user location based on both the ranging estimation and the dead reckoning approach using inertial sensors. If the user is in LoS of at least three beacon devices, location determination module applies the triangulation technique to estimate the user actual location. However, in scenarios where the user is moving, (e.g. walking or running), this triangulation technique is not always practically applicable due to the very short duration available for location computations. In addition, the existence of three LoS beacon devices (e.g Wi-Fi AP) in one location is not common. Therefore, our location determination module utilizes the inertia sensors to predict user location. However, such technique is not highly reliable in predicting user location over long periods since localization errors in dead reckoning scheme get accumulated over time. Thus, instead of using three ranges from three beacon devices, our location determination module utilizes only single range estimation to calibrate the estimated location from the inertia sensors. In situations where no LoS beacon device exists, location determination module relies on the inertia sensors to predict the location.

Figure 6b shows the calibration steps for estimating user location. First, we use the inertia sensors to estimate the user next position from the given initial position. Second, we calculate the user range to a nearby LoS beacon device. Finally, we estimate user location by calibrating the user next position and the range estimation as shown in figure 6b.

VII. Performance Evaluation

A. Experiment Scenarios

We evaluate the SpyLoc localization system under the following different scenarios:

- **Scenario 1:** In this scenario, we conduct the experiments in a pre-defined path at an indoor environment. Figure 8a shows the locations of the beacon devices and the actual walking path. Note that, during these experiments we use only single range estimation and the inertial sensors to estimate user location. In addition, we also evaluate the SpyLoc system under different walking speeds.
- **Scenario 2:** The experiments are conducted in a public space (i.e. Web Center) where we follow a casual path as shown in Figure 8b. Figure 8b also shows the positions of the deployed three beacon devices. Similarly, during these experiments, we use only single range estimation and the inertial sensors to estimate user location. Experiments also evaluate the SpyLoc system under different rates of beep signals (e.g. beep/1sec, beep/10sec, beep/20sec and beep/30sec.) in order to mimic the scenarios when SpyLoc client application doesn't receive any LoS beep signal for different durations.
- **Scenario 3:** In this scenario, we conduct our experiments inside the Web Center building. Figure 10a shows the locations of the three *beacon devices* and the positions where we estimate the locations. In these experiments, we estimate the user location using only the range estimation scheme.
- **Scenario 4:**: The experiments are done in an open parking lot space. Figure 10c shows the locations of three *beacon devices* and the different positions where we estimate the location. We use these experiments to evaluate the range estimation scheme for outdoor scenarios.

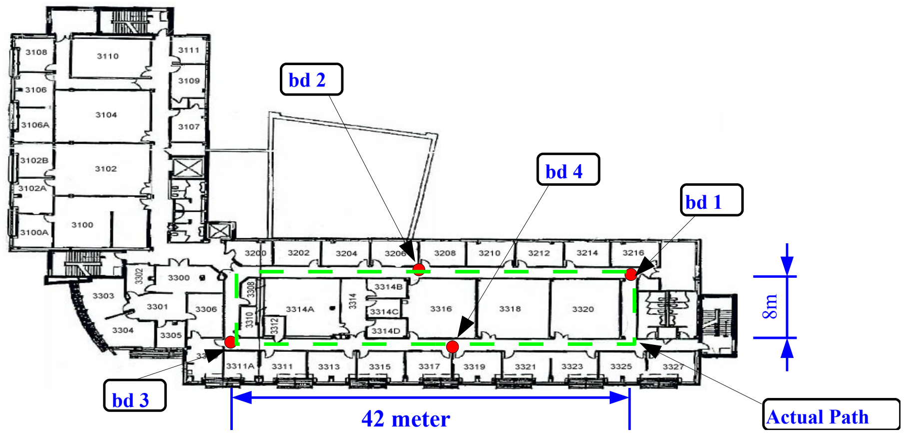
While results of both scenario 1 and scenario 2 are averaged over 5 runs, results of both scenario 3 and scenario 4 are averaged over 10 runs for each of the 50 distinguish locations. Note that, in all experiment scenarios, we track the ground truth location by marking the paths and the positions on the floor.

B. Experiment Results

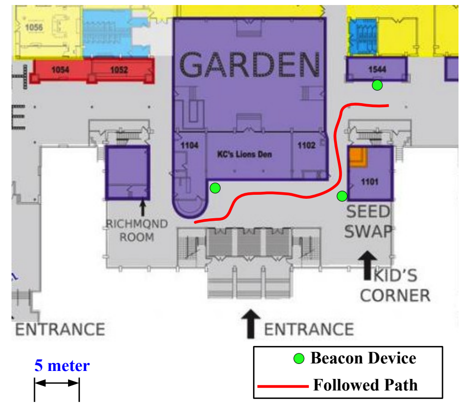
Scenario 1: The plot in Figure 9a shows the CDF of the overall estimation error by the SpyLoc Client for the walking path shown in Figure 8a. As shown, the estimation error is less than 50cm for 90% of the time. Plots in Figure 9b shows the CDF of the estimation error for the walking path shown in Figure 8a for different walking speeds. The figure shows that the estimation error is less than 90cm for 90% of the time for all different speeds. These results verify the feasibility and the efficiency of our SpyLoc system under different mobility conditions.

Scenario 2: The plot in Figure 9c shows the CDF of the estimation error for the walking path at Web Center that is shown in Figure 8b. These results verify the good accuracy of SpyLoc even for normal casual walking patterns. In some of these scenarios, SpyLoc client application might not be able to receive any LoS beep signal from the nearby beacon devices. In such cases, SpyLoc will rely solely on the inertial sensors to estimate the user locations until it becomes in the LoS of a beacon device and starts to receive LoS beep signals again. In order to mimic this situation, we change the transmission rate of beep signals. Plots in Figure 9d show the CDF of the estimation error under different beep rates. The results show that relying on the inertial sensors for long time could can result in high estimation errors. Hence, by distributing the beacons devices such that each location is in the LoS of a beacon device would significantly enhance the location estimation process.

Scenario 3: For the scenarios when the user just launched the SpyLoc application and need to estimate user initial

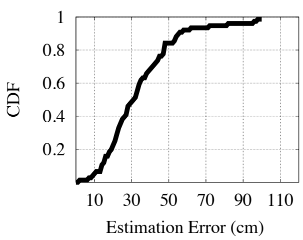


(a)

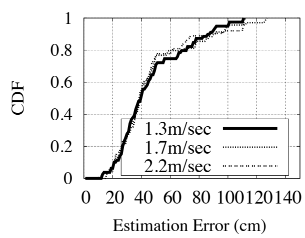


(b)

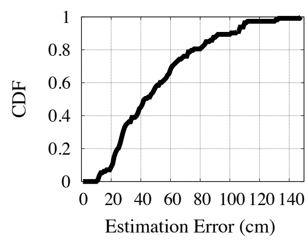
Fig. 8: (a) Indoor experiment setup at the department building where the walking path is marked by the green line. (b) Indoor experiment at Web Center where the walking path is marked by the red line.



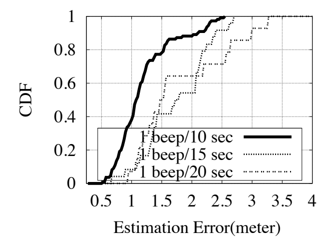
(a)



(b)

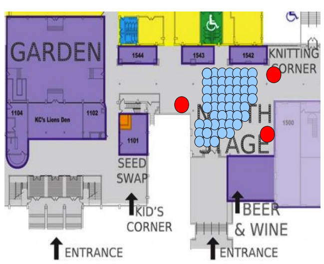


(c)

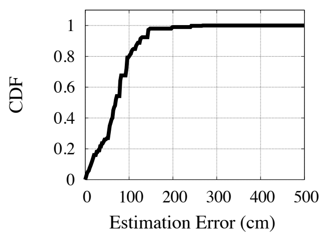


(d)

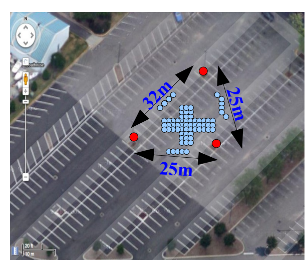
Fig. 9: (a) CDF of the overall estimation error for the walking path shown in Figure 8a. (b) CDF of the localization error for the walking path shown in Figure 8a using different speeds. (c) CDF of the estimation error for the walking path in Web Center (Figure 8b). (d) CDF of the localization error for the walking path in Web Center (Figure 8b) using different beep rates.



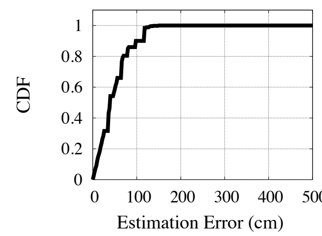
(a)



(b)



(c)



(d)

Fig. 10: (a) Indoor experiments at Web Center with three LoS beacon devices. (b) CDF of the localization error for the indoor experiments shown in Figure 10a. (c) Outdoor experiments at the parking lot with three LoS beacon devices. (d) CDF of the localization error for the outdoor experiments shown in Figure 10c.

position, or when the application does not receive any LoS beep signal for a long duration, SpyLoc will need to calculates three ranges to three LoS beacon devices in order to initiate or calibrate user location. In order to evaluate this situation, we run experiments to estimate the user location using only the range estimation technique. Figure 10a shows the positions of the collected samples (i.e., blue circles). These experiments are done in a public place with a lot of surrounding students and with several furniture such as tables, chairs and desktop PCs. We deployed three beacons devices in which the distance between any two beacon devices is in the range of 20-25m. Figure 10b shows the CDF of the estimation error in which the error is less than 1.5 meter for more than 90% of the time.

Scenario 4: The experiments are conducted at outdoor to validate the feasibility of using SpyLoc application in outdoor environments. In Figure 10c, blue circles show the position of the collected samples. Figure 10d shows the distribution of the estimation error in cm for these experiments. The figure shows that the error in location estimation is less then 1meter for more than 90% of the samples collected in this outdoor experiment.

VIII. Related Work

Most of the localization research works have been based on Radio Frequency (RF)-based techniques that leverages the signal strength of RF signals from different nearby RF sources or infrastructures (e.g., Wi-Fi AP, Cellular Tower) [24], [9], [7], [25]. Recently, researchers are combining multiple modalities such as sound with the Wi-Fi to achieve higher accuracy localization system [1], [2], [26], [8]. For example, localization schemes [1], [2], [27], [28] utilize the acoustic based ranging [15] scheme and combine it with the RF-based schemes at the application layer. In addition, some localization systems use multiple modalities of the smartphone to determine user location at different levels of accuracy[10], [13], [16], [12]. The following table compares between the recent proposed schemes and SpyLoc.

	[1], [2]	[13]	[16], [12]	[8]	SpyLoc
RF or Acoustic Fingerprint	Yes	Yes	No	No	No
Smartphone Usability	Yes	Yes	Yes	No	Yes
Support User Motion	No	Yes	Yes	No	Yes
Accuracy	1-5m	1.7m	1-5m	1m	1m
Floor layout or Landmarks	No	Yes	Yes	No	No
Backend / Centralized system	Yes	Yes	No	No	No

IX. Conclusion

In this paper, we propose a practical location determination system that leverages the balance between our high accuracy ranging scheme and the light weight dead reckoning approach. We show how both the acoustic and the RF interfaces in off-the-shelf smartphones could be utilized to achieve a high accuracy localization system. In addition, we also develop an efficient and light way to utilize the inertial sensors to determine the distance and direction of the user movement. We also show how we leverage the unique characteristics of the acoustic signals to differentiate between LoS and NLoS acoustic signals. Finally, we evaluate our system under different real scenarios where we consider different walking speeds. Results show that SpyLoc in most of the scenarios is able to achieve less then 1 meter accuracy.

References

[1] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, "Centaur: locating devices in an office environment," in *The 18th Int. Conference on Mobile Computing and Networking (MobiCom'12)*, 2012.

[2] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *The 18th International Conference on Mobile Computing and Networking (MobiCom'12)*, 2012.

[3] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *The 10th International Conference on Mobile Systems, Applications and Services (MobiSys12)*, 2012.

[4] A. Matic, A. Popleteev, V. Osmani, and O. Mayora-Ibarra, "FM radio for indoor localization with spontaneous recalibration," *Pervasive Mobile Computing*, vol. 6, December 2010.

[5] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka., "Spot localization using phy layer information." in *The 10th International Conference on Mobile Systems, Applications, and Services (MobiSys12), 2012.*

[6] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. W. I, "Indoor location sensing using geo-magnetism." in *The 9th Int. Conf. on Mobile Systems, Applications, and Services (MobiSys11), 2011.*

[7] M. Youssef and A. Agrawala, "The horus wlan location determination system." in *The 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys'05)*, 2005.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system." in *The 6th International Conference on Mobile Computing and Networking (MobiCom'00), 2000.*

[9] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp, "Walrus: wireless acoustic location with room-level resolution using ultrasound," in *The 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys05)*, 2005.

[10] M. Azizyan, I. Constandache, and R. R. Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting." in *The 15th International Conference on Mobile Computing and Networking (MobiCom'09), 2009.*

[11] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum." in *The 9th International Conference on Mobile Systems, Applications, and Services (MobiSys'11), 2011.*

[12] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *The 18th Int. Conf. on Mobile Computing and Networking (Mobicom '12)*, 2012.

[13] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization." in *The 10th International Conference on Mobile Systems, Applications, and Services (MobiSys'12), 2012.*

[14] M. Uddin and T. Nadeem, "Rf-beep: A light ranging scheme for smart devices." in *PerCom 2013.*

[15] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: A high accuracy acoustic ranging system using cots mobile devices." sydney, Australia, ACM SenSys 2007.

[16] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, ser. UbiComp '12. New York, NY, USA: ACM, 2012, pp. 421–430.

[17] Z. Sun, A. Purohit, S. Pan, F. Mokaya, R. Bose, and P. Zhang, "Polaris: Getting accurate indoor orientations for mobile devices using ubiquitous visual patterns on ceilings," ser. ACM HotMobile '12, 2012.

[18] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, "Detecting driver phone use leveraging car speakers." in *International conference on Mobile computing and networking, 2011.*

[19] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.

[20] S. H. Collins and A. D. kuo, "Two independent contributions to step variability during over-ground human walking." *Journal of Applied Physiology*, 2011.

[21] R. Burkard, J. Eggermont, and M. Don, *Auditory Evoked Potentials: Basic Principles and Clinical Application*, ser. Point (Lippincott Williams and Wilkins) Series. Lippincott Williams & Wilkins, 2007.

[22] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI-01 workshop on "Empirical Methods in AI"*.

[23] S. Borman, "The expectation maximization algorithm: A short tutorial. unpublished paper available at http://www.seanborman.com/publications," Tech. Rep., 2004.

[24] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 00: the 19th Annual IEEE Conference on. Computer Communications*, tel-Aviv, Israel: IEEE Infocom, March 2000.

[25] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "Ariadne: a dynamic indoor signal map construction and localization system," in *The 4th International Conference on Mobile Systems, Applications, and Services (MobiSys06)*, 2006.

[26] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application." in *The 5th Int. Conference on Mobile Computing and Networking (MobiCom'99), 1999.*

[27] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, "On the feasibility of real-time phone-to-phone 3d localization." in *The 9th ACM Conference on Embedded Networked Sensor Systems (SenSys11)*, 2011.

[28] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda., "Swordfight: Enabling a new class of phone-to-phone action games on commodity phones," in *The 10th International Conference on Mobile Systems, Applications, and Services (MobiSys'12)*, 2012.