Improving the Vertical Accuracy of Indoor Positioning for Emergency Communication

Wonsang Song[*], Jae Woo Lee[*], Byung Suk Lee[†], and Henning Schulzrinne[*]
[*]Department of Computer Science, Columbia University
{wonsang, jae, hgs}@cs.columbia.edu
[†]Department of Electrical Engineering, Columbia University
bsl@ee.columbia.edu

Abstract—The emergency communication systems are undergoing a transition from the PSTN-based legacy system to an IP-based next generation system. In the next generation system, GPS accurately provides a user's location when the user makes an emergency call outdoors using a mobile phone. Indoor positioning, however, presents a challenge because GPS does not generally work indoors. Moreover, unlike outdoors, vertical accuracy is critical indoors because an error of few meters will send emergency responders to a different floor in a building.

This paper presents an indoor positioning system which focuses on improving the accuracy of vertical location. We aim to provide floor-level accuracy with minimal infrastructure support. Our approach is to use multiple sensors available in today's smartphones to trace users' vertical movements inside buildings.

We make three contributions. First, we present the elevator module for tracking a user's movement in elevators. The elevator module addresses three core challenges that make it difficult to accurately derive displacement from acceleration. Second, we present the stairway module which determines the number of floors a user has traveled on foot. Unlike previous systems that track users' foot steps, our stairway module uses a novel landing counting technique. Third, we present a hybrid architecture that combines the sensor-based components with minimal and practical infrastructure. The infrastructure provides initial anchor and periodic corrections of a user's vertical location indoors. The architecture strikes the right balance between the accuracy of location and the feasibility of deployment for the purpose of emergency communication.

I. INTRODUCTION

The emergency communication systems in the United States and elsewhere are undergoing a transition from the PSTN-based legacy system to a new IP-based system. The new system is referred to as the Next Generation 9-1-1 (NG9-1-1) system [1] in the US. We have previously built a prototype NG9-1-1 system [2] based on the Session Initiation Protocol (SIP) [3].

The most important piece of information in the NG9-1-1 system is the caller's location. The location is first used for routing the call to a proper call center. The emergency responders then use the caller's location to pinpoint the caller on site. Therefore, it is essential to determine the caller's location as precisely as possible to minimize delays in emergency response. Delays in response may result in loss of lives.

In the NG9-1-1 system, GPS can provide a user's location accurately when the user makes an emergency call outdoors using a mobile phone. Indoor positioning, however, presents

a challenge because GPS does not generally work indoors. Moreover, unlike outdoors, vertical accuracy is very important in indoor positioning because an error of few meters will send emergency responders to a different floor in a building, which may cause a significant delay in reaching the caller. The importance of vertical positioning makes GPS not a good solution even if GPS signals can somehow reach indoors, since the altitudes reported by GPS are usually inaccurate [4], [5].

Ladetto and Merminod [6] proposed a barometer-based solution for vertical positioning. Barometers, however, have a critical limitation when they are used in a vertical positioning system intended for emergency situations. Firefighters use a technique called positive pressure ventilation (PPV) [7], which means blowing air into a burning building in order to clear out smoke. PPV will result in pressure changes in the building, which will in turn cause large fluctuations in barometer readings. In addition, parts of some buildings are intentionally pressurized for various reasons [8], which will also affect barometer readings.

This paper presents a proposal to augment our previous NG9-1-1 prototype system with a new indoor positioning system. The indoor positioning system focuses on improving the accuracy of vertical positioning. We aim to provide floor-level accuracy with minimal infrastructure support. Our approach is to use multiple sensors, all available in today's smartphones, to trace users' vertical movements inside buildings.

We make three contributions for improving vertical accuracy of indoor positioning. First, we present the elevator module for tracking a user's movement in an elevator. The elevator module calculates the elevator's displacement from linear acceleration obtained from the accelerometer in the user's smartphone. Our solution addresses three core challenges that make it difficult to accurately derive displacement from acceleration.

Second, we present the stairway module which determines the number of floors a user has traveled on foot. Unlike previous systems that track users' foot steps, our stairway module uses a novel landing counting technique. Landings are the level areas either at the top of a staircase or in between flights of stairs.

Third, we present a hybrid architecture that combines the sensor-based components with minimal and practical infrastructure. The infrastructure, consisting of sparsely deployed beacons and central building database, provides initial anchor
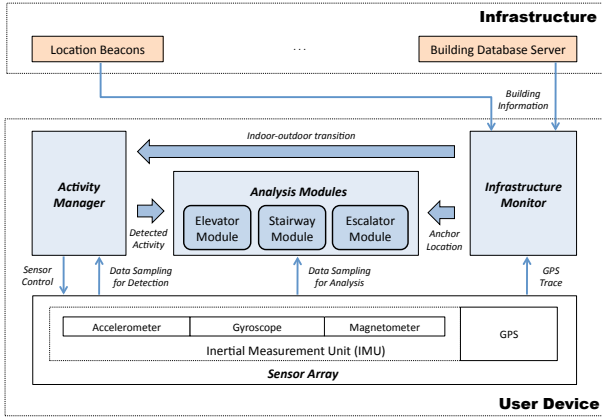
Fig. 1. Architecture overview.

and periodic corrections of a user's indoor vertical location. We believe that the architecture strikes the right balance between accurate location information and the ease of deployment for the purpose of emergency communication.

This paper is organized as follows. Section II presents our overall architecture. Section III describes the design and algorithms of the elevator and stairway modules. Section IV describes implementation details. Section V provides our evaluation results. Section VI discusses related work. Lastly, we conclude and discuss future work in Section VII.

II. ARCHITECTURE OVERVIEW

Figure 1 shows the overall architecture of our vertical positioning system. We describe each component in detail in the following subsections.

A. Sensor array

The sensor array includes different kinds of sensors available in smartphones. The Inertial Measurement Unit (IMU) integrates a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer. Thus, the IMU provides motion sensing with a total of nine degrees of freedom. The accelerometer measures linear accelerations along the three spatial axes. The measured accelerations can be used to detect whether a user is moving, and if so, the user's velocity or traveled distance can be derived from them. The gyroscope measures the angular velocities of rotations around the three spatial axes. The orientation of the device can be derived from the gyroscope measurement. The magnetometer is a digital compass that measures the strength of the Earth's magnetic field. The compass provides the heading of the device. Heading refers to the angle which the device forms with the magnetic north on a level plane.

GPS provides the device's location in the geographic coordinates using satellite signals. GPS cannot be used indoors but it can help detect when a user moves from outdoor to indoor.

B. Analysis modules

The analysis modules collect data from the sensor array and compute a user's location. There are three analysis modules

in our architecture: the elevator module, the stairway module, and the escalator module.

The elevator module calculates the vertical displacement of an elevator by measuring its linear acceleration. The linear acceleration is measured using the device's accelerometer. Integrating the linear acceleration twice with respect to time yields the distance that the elevator has traveled.

The stairway module determines the number of floors a user has traveled by counting the number of landings in stairways. Our landing detection algorithm is based on an intuitive fact that there is less vertical movement on landings than on steps. The stairway module utilizes the accelerometer, the gyroscope, and the magnetometer. We describe the details in Section III-B.

The escalator module is left for future work. We envision that the escalator module will incorporate elements of both elevator and stairway modules.

C. Activity manager

The activity manager coordinates the interactions between the sensor array and the analysis modules. The activity manager monitors the sensors to detect changes in a user's activity, such as indoor-outdoor transitions, riding an elevator, or walking on a stairway. Once the user's activity is identified, the activity manager will select the proper analysis module to process the data from the sensor array. The activity manager can also reduce the sampling rates of the sensors that are not used for the current activity in order to conserve energy.

While the role of activity manager is important in our architecture, our work does not focus on it because we can use existing activity recognition systems [9], [10]. Integrating an existing activity recognition system into our architecture remains as future work.

D. Infrastructure

As we will show in Section V, the elevator and stairway modules perform well within limited ranges, but the modules cannot reliably capture the user's movement over longer vertical distances. Moreover, the sensor-based components can only report *relative location*, i.e., the number of floors that the user has traveled. Therefore, the initial anchor location must be provided in order to obtain the *absolute location*.

Those problems can be solved by deploying an infrastructure for indoor positioning. Densely deployed infrastructure, such as beacons installed every floor and every entrance, can provide accurate location, but the high cost of such installation is a hindrance to wide deployment. Ubiquitous deployment is a requirement for the emergency communication scenario, which is the motivation for our work. On the contrary, sparsely deployed infrastructure will not be able to provide the required level of accuracy.

Our architecture combines the sensor-based components with minimal and practical infrastructure. First, the infrastructure includes location beacons deployed at each entrance of a building. The beacons provide the location of a user's entry to the building. The floor of entry becomes the anchor for all subsequent calculations of the user's vertical location. In

addition to the floor of entry, the beacons also provide other building information which is needed by the analysis modules. The additional building information includes the floor-to-floor height and the number of landings between each pair of floors. User devices include the infrastructure monitor, which interacts with the location beacons.

Second, for the buildings that are not equipped with these beacons, we propose that central authorities such as local governments maintain well-known building database servers. When a user enters a building not equipped with the beacons, the infrastructure monitor sends the last known GPS location to the building database server to retrieve the same building information that the location beacons would have provided. The central authorities responsible for the building databases can reduce the burden of keeping the databases up to date using crowd sourcing.

Lastly, the limited range of the sensor-based components can be overcome by sparsely deploying location beacons at the edge of the range. For example, if the location tracked by the elevator module is reliable up to 20 floors, beacons can be placed at elevator entrances every 20 floors.

III. SYSTEM DESIGN AND ALGORITHMS

A. Elevator module

There are three challenges in accurately measuring the vertical distance that a user has traveled in an elevator. The three challenges are how to extract the vertical component in the accelerometer measurement, how to subtract Earth's gravitational acceleration, and how to eliminate noise and errors.

The accelerometer returns linear accelerations along the three axes. Those three axes are not aligned with the world coordinate system. Instead, they are aligned with the frame of the device. Thus, the axes in the device coordinate system keep changing as the orientation of the device changes. One way to extract vertical acceleration is to combine the accelerometer measurement with the gyroscope measurement. In fact, we do this in the stairway module. In the elevator module, however, we take advantage of the fact that, in the elevator, the dominant movement of the device is in the vertical direction. We simply assume that the measured acceleration is close to vertical, and approximate the vertical projection with the vector itself. Thus, the vertical acceleration is calculated as follows:

$$a_{vertical} \approx |\overrightarrow{a}| = \sqrt{x^2 + y^2 + z^2} \qquad (1)$$

where x, y, and z are three-axis accelerometer measurements. We do not need a gyroscope in this calculation. We justify our approach by making the following two observations. First, a user's sudden movements in the elevator will be filtered out by the low-pass filter, which we will describe shortly. Second, users typically stand still in the elevator, and when they move, the accelerations of the movements are small compared to the vertical acceleration of the elevator. The consequence of this approximation is that whenever there is non-vertical acceleration, we overestimate the vertical acceleration by $\frac{1}{\cos\theta}$, where θ is the angle that the measured acceleration
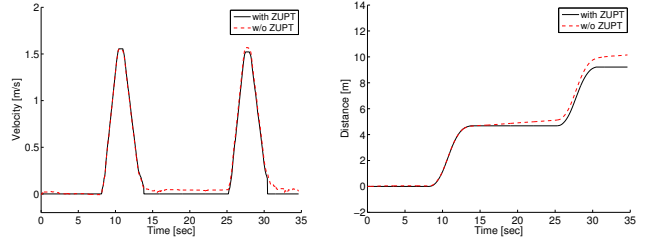
(a) Velocity with and without ZUPT. (b) Distance with and without ZUPT.

Fig. 2.   Comparison of the distance calculations with and without ZUPT.

vector makes with the vertical axis. This overestimation is small, and we compensate it by applying zero velocity update (ZUPT), which we describe later. Our measurement shows that the approximation does not affect the resulting distance calculation.

The vertical acceleration calculated above includes the gravitational acceleration (g), which we need to subtract before computing the traveled distance. In theory, g should be constant at $9.8\,\text{m/s}^2$, but we found slight variations in our experiments. We measured g by sampling the accelerations of smartphones sitting still on a desk. The measured values deviated slightly from g, and moreover, the variations were different on different devices. Smartphone SDKs provide APIs returning g-free acceleration, but they exhibited the same deviation. We eliminate the effect of the deviation in g as follows. We take advantage of the fact that, if we take g out of the acceleration, the integral of the acceleration taken over the duration of the trip must be zero because the elevator is not moving at the end of the trip. Thus we can deduce that the value of g measured by the device is the mean of the acceleration samples taken over the trip.

The accelerometer output contains a significant amount of noise. We apply two existing techniques to tackle this problem. First, we apply a low-pass filter to the accelerometer output. This filters out the user's sudden movements and the accelerometer's inherent noise which we refer to as drift. Second, we apply a technique called zero velocity update (ZUPT) [11] to eliminate accumulated errors. Integrating the acceleration with respect to time produces the velocity of the elevator. We reset the velocity to zero during the period when the acceleration is zero and the velocity is within a predefined threshold. The threshold value we choose is small compared to the speed of the elevator, so that we do not mistakenly zero out the velocity of an elevator moving at a constant speed. The accuracy of the distance calculation is improved in that, at each stop, ZUPT has an effect of wiping out the accumulated errors due to the drift and the user's non-vertical movements.

Figure 2 demonstrates the effectiveness of ZUPT. We compare the computed velocities and distances when an elevator traveled from the first, to the second, and then to the third floor. Without ZUPT, the accumulated acceleration errors result in non-zero velocities when the elevator is at the second and the third floor. This in turn results in an error of approximately one meter in the distance calculation at the end.
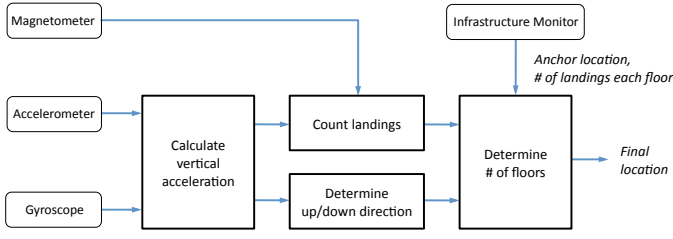
Fig. 3. Overview of the stairway module.

Fig. 4. (a) Vertical acceleration measurement; (b) Spectrogram of vertical acceleration; (c) $p_{walk}(t)$ extracted from (b)

B. Stairway module

The stairway module determines the number of floors a user has traveled using our landing counting algorithm. To the best of our knowledge, landing detection has not been used for vertical positioning systems.

Figure 3 illustrates how the stairway module works. First, the stairway module calculates vertical acceleration from the accelerometer and gyroscope measurements. Unlike an elevator's movement, a user's movement on a stairway is more complex. A gyroscope is needed to transform the acceleration in the device coordinate system to the world coordinate system. We convert the accelerometer measurements in the device coordinate system to the world coordinate system using a rotation matrix as shown below:

$$\overrightarrow{a}' = R\,\overrightarrow{a} \qquad (2)$$

where $\overrightarrow{a}'$ is the acceleration in the world coordinate system, $\overrightarrow{a}$ is the acceleration in the device coordinate system, and R is the rotation matrix. Most smartphone platforms provide an API to obtain R. We then take the resulting z-axis acceleration in the world coordinate system and subtract g from it. We calculate g in the same way as in the elevator module.

The landing counting algorithm compares the amplitude of vertical acceleration between steps and landings. The algorithm is based on the intuitive fact that the amplitude of the vertical acceleration is much smaller on landings than on steps because there are less vertical movements on landings.

Figure 4(a) shows a measurement of a user's vertical acceleration when she walks down four floors passing eight landings. The amplitude difference between steps and landings is clearly observed. Figure 4(b) is the magnitude spectrogram $|X(t, f)|$ in dB scale, transformed from Figure 4(a)'s acceleration data. The regions of small amplitude in Figure 4(a) manifest as reduced magnitude in the frequency range between 0.5 to 2 Hz, which corresponds to human walking.

We define $p_{walk}(t)$ to extract human walking activity from the magnitude spectrogram:

$$p_{walk}(t) = \sum_{0.5\,\mathrm{Hz} < f < 2\,\mathrm{Hz}} 10 \log_{10} |X(t, f)| \qquad (3)$$

where t is time and f is frequency. Figure 4(c) shows $p_{walk}(t)$, where we can clearly observe the dips at landings.

Our landing counting algorithm traces the p_{walk} level shown in Figure 4(c) to count the number of landings. Figure 5(a) illustrates this p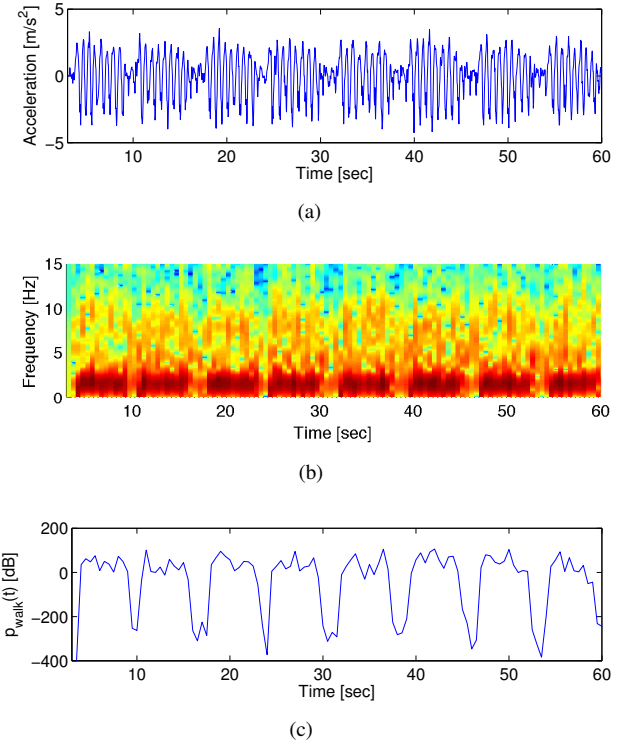rocess. Each landing is characterized by a dip below its mean value. The fall and rise of the level crossing the mean value indicate the beginning and end of a landing, respectively. The beginning and end of a landing are shown as the bumps of the "Landing detection" line in Figure 5(a).

In addition to vertical acceleration, the stairway module uses heading information from the magnetometer to improve the accuracy of landing detection. Most of the time, users turn around 180 degrees on landings. We use such heading changes to correct errors in landing detection, specifically to remove incorrectly identified landings. Since we are only interested in 180 degree turns, our magnetometer reading does not require calibration.

Figure 5(b) shows a case where our algorithm removes two incorrectly identified landings using the heading information from the magnetometer. The dotted line labeled "Heading" shows the heading changes reported by the magnetometer. The heading largely stays the same from 15 sec to 25 sec, and changes from $220°$ to $40°$ in the next two seconds. This $180°$ turn, combined with the bumps on the landing detection line confirms a landing. Note that the seeming discontinuity in the heading from $20°$ to $330°$ at 37 sec is in fact a steady change from $20°$ to $-30°$, wrapping around. The two rectangles in the figure highlight two incorrectly identified landings being removed because the heading did not change during the period.

This heading-based verification of landings makes it unlikely that our algorithm produces false positives. When the acceleration-based landing detection misses a landing to begin with, however, the heading information does not help recover

(a) Case 1: no error.

(b) Case 2: false positives get fixed by magnetometer.

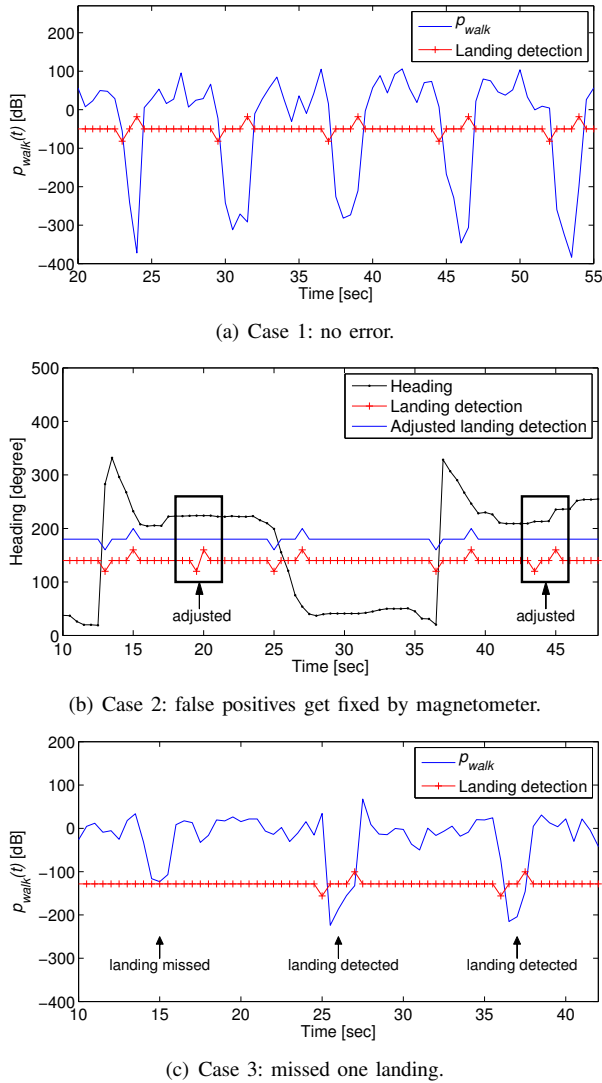

(c) Case 3: missed one landing.

Fig. 5.   Three landing detection cases.

it. Figure 5(c) shows this case. Therefore, our algorithm produces a conservative estimate of the number of landings.

Up/down direction of the user's movement is determined by comparing the average vertical velocity on steps and landings. We determine that the user is ascending if the velocity on steps is higher than the velocity on landings, and vice versa. In theory, the average vertical velocity should be zero on landings, positive when the user walking up steps, and negative when walking down. But in practice, the velocity values can shift due to the noise and errors that have been introduced while extracting vertical acceleration and subtracting g.

The stairway module returns a relative location which is the number of floors the user has traveled from the initial floor. Like the elevator module, the stairway module relies on the information from the infrastructure monitor to get the initial anchor location. The infrastructure monitor also provides the number of landings between each pair of floors. There are typically two landings per floor but the number can vary

depending on the design of a building. In some buildings, for example, there are more landings between the lobby and the second floor.

IV. IMPLEMENTATION

A. Hardware platform

We used the Apple iPhone 4 and 4S for implementation and evaluation. The iPhone 4 contains an accelerometer, a gyroscope, and a magnetometer. The accelerometer in iPhone 4 can measure acceleration from $-2\,g$ to $+2\,g$, where $1\,g$ is $9.8\,\mathrm{m/s^2}$ [12]. The sampling rate can be adjusted from $0.5\,\mathrm{Hz}$ to $1\,\mathrm{kHz}$. We used $30\,\mathrm{Hz}$ for our measurements. The gyroscope measures angular velocity from -250 degree/sec to +250 degree/sec [12]. We also read the gyroscope at $30\,\mathrm{Hz}$. The magnetometer is a three-axis electronic compass manufactured by Asahi Kasei [13]. According to the specification from the manufacturer, the measurement range is $\pm1{,}200\,\mu\mathrm{T}$.

B. Data collection from sensor array

In our current prototype, an application running on iPhone collects data from the sensor array. The measurements from the accelerometer and gyroscope in iPhone can be accessed using the Core Motion framework in iOS. The Core Motion framework provides APIs to retrieve the raw data such as the timestamp and three-axis accelerations. The framework also provides processed motion data, such as *attitude*, which is derived from both the accelerometer and gyroscope. Attitude refers to the spatial orientation of the device with respect to the world coordinates, and can be obtained either as a rotation matrix or as a quaternion. We use the rotation matrix in our implementation of the stairway module.

The heading information from the magnetometer can be accessed using the Core Location framework in iOS. The framework provides two headings: magnetic heading and true heading. Magnetic heading points to the magnetic north pole, and true heading points to the geographic north pole. We use the magnetic heading in our implementation. Both types of heading will satisfy our need to detect a user turning around on landings, but using magnetic heading avoids additional processing to calculate the true heading from the current location, which may consume more energy.

We collected GPS traces outdoors. The Core Location framework provides an API to obtain the device's location. Normally the framework determines the locations using various sources including GPS, Wi-Fi, and cellular network, but a flag can be passed to indicate that we are only interested in GPS locations.

C. Data collection from infrastructure

We chose Bluetooth technology for location beacons because Bluetooth is available on most smartphones. The infrastructure monitor and the beacon communicate using Bluetooth service discovery protocol (SDP) [14]. SDP allows Bluetooth devices to discover available services and their characteristics without initiating a pairing process.

Currently, iOS does not provide APIs for Bluetooth communication. We implemented the Bluetooth client using BTstack [15], an open source Bluetooth stack for iOS. Installing BTstack requires jailbreaking iPhone. We prototyped location beacons as a Java application using BlueCove [16], an open source Java library for Bluetooth.

We used a Mac mini computer wrapped in aluminum foil to prototype a Bluetooth beacon. The foil wrapper decreases the Bluetooth signal strength so that it would not reach the adjacent floors.

The beacon interacts with the infrastructure monitor in the following sequence. First, the infrastructure monitor scans for nearby Bluetooth devices by sending periodic inquiry messages. Second, the infrastructure monitor sends an SDP request to all the discovered Bluetooth devices. The request includes a unique identifier defined for location beacon service, so the request is ignored by all devices that are not location beacons. Lastly, the infrastructure monitor receives an SDP response from a location beacon. An SDP response contains the building's address, the floor where the beacon is located, and for each pair of floors, the height and the number of landings.

The infrastructure monitor falls back on a central building database server when a building is not equipped with location beacons. While a user stays outdoors, the infrastructure monitor tracks the user's location using GPS. When GPS signal is lost, the infrastructure monitor assumes that the user has entered a building, and sends the last known GPS coordinates to the building database server. The building database server finds the nearest entrance from the user's last GPS location, and returns the same information that the location beacon returns.

D. Analysis modules

The current version of our iPhone application does not include the analysis modules. The collected sensor data is sent to a central repository. Using this data, we have tested our algorithms for the analysis modules prototyped in MATLAB.

We are currently developing the analysis modules running on iPhone. It is desirable to run all analysis locally on the user's device whenever possible, so that the user's privacy is preserved as much as possible.

V. EVALUATION

We evaluate the algorithms of our elevator and stairway modules to show that our positioning system can provide floor-level accuracy for the user's vertical location. All evaluation scenarios assume that the activity manager correctly identifies the user's activity and selects the proper analysis module.

A. Elevator module

We evaluated the elevator module in three different research and classroom buildings at Columbia University: CEPSR, Mudd, and Pupin. They have 10, 15, and 13 floors, respectively. Table I shows the reference floor-to-floor height of each building, which we measured using a tape measure, followed

Building name	Floor height (by tape measure)	Average error	Error-to-height ratio
CEPSR	4.65 m	0.08 m	1.6%
Mudd	3.67 m	0.06 m	1.7%
Pupin	3.48 m	0.09 m	2.7%

TABLE I
ERRORS IN ONE FLOOR DISTANCE CALCULATED BY ELEVATOR MODULE.
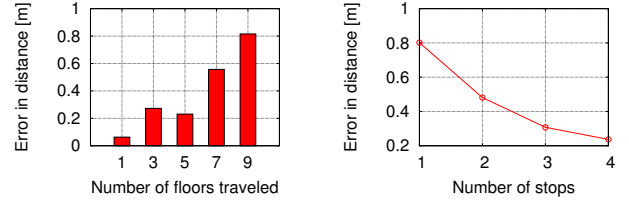
(a) Traveling 1, 3, 5, 7, 9 floors. (b) Traveling 9 floor with 1 - 4 stops.

Fig. 6. Distance errors of elevator module measured in Mudd building.

by the error of the result from the elevator module. The error is the difference between the reference height and the distance calculated by the elevator module when a user moves one floor in an elevator in each building. The error is an average of ten trials, five moving up and five moving down.

Errors are small in all three buildings, indicating that the elevator module can provide accurate vertical location up to a reasonable number of floors. We can extend the range by strategically deploying location beacons. For example, in the Pupin case in Table I, the error is under 3%, so the elevator module will be accurate up to about 15 floors. Thus, location beacons can be deployed conservatively in every 10 floors to cover the entire building.

Figure 6(a) shows distance errors from the elevator module as we increase the number of floors traveled in an elevator without stopping. The graph shows that the errors accumulate as the elevator travels farther. The error of $0.82\,\text{m}$ when the user traveled nine floors is about 22% of the floor-to-floor height, which is well within the margin of error for accurately determining the destination floor.

Figure 6(b) plots the distance errors of traveling nine floors in an elevator as we vary the number of stops that the user has made during the travel. The graph shows that the error decreases as the user makes more stops. This shows the effectiveness of applying ZUPT in the distance calculation. At each stop, ZUPT eliminates accumulated errors by removing residual velocity. Therefore, if the elevator makes stops during the trip, the elevator module's distance estimation becomes much more accurate, extending the upper bound of the elevator module's distance limitation.

B. Stairway module

We evaluated the stairway module in two buildings. One was an office building and the other was a residential building. Both buildings have two landings between each pair of floors. The stairways in the two buildings have different properties: the office building has 27 steps between a pair of floors and the residential building has 16.

Building type	# of floors walked	# of trials	# of trials with 1-landing error
Office	1	10	3
	2	10	1
	3	10	1
	4	10	0
Residence	1	10	0
	2	10	0
	3	10	0
	4	10	0

TABLE II
THE STAIRWAY MODULE MEASUREMENTS.

Table II shows our results. In each building, we walked 1, 2, 3, and 4 floors. We repeated each travel ten times, five walking up and five walking down. There were several *recoverable* errors in the office building. We consider an error recoverable when the stairway module missed a single landing. For example, in one of the trials walking 3 floors in the office building, the stairway module reported 5 landings (2.5 floors) instead of 6 landings (3 floors). In these cases, we assume that the nature of error is an omission of a landing, so we simply round up the value. Our landing detection algorithm described in Section III-B makes false positives unlikely. Indeed, we observed no case of such false positives in this experiment.

Errors of two or more landings are deemed *unrecoverable* because they would result in miscalculations of the number of floors traveled. In this experiment, we have not encountered any unrecoverable error.

We note that, in all trials in Table II, the user moved at a normal walking speed. If the user walks very fast or very slowly, the amplitude difference of the accelerometer reading between steps and landings is much less pronounced. We can address this issue by giving more weight to the heading information from the magnetometer. In the extreme case, we can reverse the roles of the accelerometer and the magnetometer, i.e., instead of using the magnetometer to make adjustments to the landings identified by the accelerometer, we can use the magnetometer first to identify landings. The relative weights of the two sensors can be dynamically determined depending on how pronounced the amplitude difference is.

The iPhone's magnetometer readings, however, often showed large fluctuations in our experiments even when the user did not change direction. For this reason, we chose to use the magnetometer conservatively, i.e., only for correcting false positives. In order to see the effectiveness of the magnetometer-first approach, we conducted the same experiment with the user walking very fast and very slowly, and selected the measurements that did not contain incorrect magnetometer readings. We confirmed that the magnetometer-first approach, when the magnetometer readings are reliable, can cover wider range of human walking speed.

VI. RELATED WORK

Indoor positioning systems can be put into two categories: infrastructure-based and self-contained systems. Infrastructure-based systems rely on infrastructure support such as sensors or beacons deployed in buildings. Sensors detect signals that are emitted by user devices, and beacons transmit signals that are received by user devices. Self-contained systems do not rely on any external entity. Instead, inertial sensors in user devices are used to keep track of users' movements indoors. There are hybrid systems that combine elements from both categories. Our solution is an example of a hybrid system.

A. Infrastructure-based systems

1) Proximity detection: Proximity detection based systems locate users by detecting signals emitted by user devices. The signals usually carry unique IDs for the devices.

Active Badge [17] uses infrared (IR) signals. Since IR signals cannot penetrate walls, there is no interference among sensors in different rooms or floors. Active Badge can determine the room that a user is located with high precision, but requires the user to wear the badge.

Many systems use Bluetooth technology for location detection [18], [19] because Bluetooth is inexpensive and ubiquitous. Bluetooth's longer range and its ability to penetrate walls, however, result in lower accuracy and precision.

Systems relying solely on proximity detection would require a large number of sensors if they were to provide room-level indoor location. Floor-level indoor location might require fewer sensors since only access points such as stairway entrances and elevators need to be covered, but it still presents a significant infrastructure challenge.

2) Triangulation: Triangulation measures the distances from multiple known reference points to determine a user's location.

Cricket [20] and WALRUS [21] transmit RF and ultrasonic signal simultaneously. Because the two signals travel at different speeds, a receiver can derive the distance to the transmitter from the difference in the arrival times. This eliminates the need to synchronize the clocks of all transmitters and receivers, as is the case for other systems based on time-of-arrival.

Ubisense [22] provides a commercial solution for indoor positioning using ultra-wideband (UWB). UWB has an advantage for indoor positioning in that it does not suffer from multipath effect. UWB signal has a short pulse timing, thus the path signal can easily be distinguished from the reflected ones. Ubisense system provides very accurate indoor locations, with errors less than 15 cm.

Compared to proximity detection, triangulation requires fewer sensors, but the number is still high. For example, a Ubisense system installed for a 1,800 m assembly line consists of 470 sensors [23]. In addition, multipath and shading effect make it hard to use triangulation for vertical positioning. Floors and ceilings of a building degrade accuracy. Installing sensors on every floor will solve the problem, but such an installation will lose the triangulation's advantage over proximity detection.

3) Fingerprinting: At each location, fingerprinting identifies signals that have long-term stability. During the *offline* phase, the signal strengths at different location coordinates are recorded to build a fingerprinting database. At the *online*

phase, the real-time signal measurement is looked up in the fingerprinting database to find a matching location.

Many kinds of signals have been used for fingerprinting systems. RADAR [24], Place Lab [25], and Ekahau [26] use ubiquitous Wi-Fi signals. Otsason *et al.* [27] developed a cellular-based fingerprinting system. Patel *et al.* [28] inject RF signals on the power line of a building, and construct a fingerprinting database from the signals emanating from the power line. There are a number of systems that use the distortions of Earth's magnetic field caused by the steel structure of a building [29]–[31].

The main disadvantage of fingerprinting is the effort required to conduct offline surveys. To achieve an acceptable accuracy, signals should be sampled at every meter, and on top of that, at least toward four different directions at each location [24], which generates an enormous amount of data. In addition to the initial deployment, the fingerprinting database needs to be updated whenever there are changes in the environment, such as moving furniture or equipment.

B. Self-contained systems

1) Step-based systems: Step-based systems detect steps in human movements and measure the displacement vector of each step. The displacement vector is composed of the stride length and direction. The user's location is then calculated by adding all displacement vectors to the initial location.

Yeh *et al.* [32] and Vildjiounaite *et al.* [33] use sensors which are mounted on a user's shoes or ankles for measurement. On the one hand, foot-mounted sensors can directly measure human steps, so errors in step detection and stride estimation can be reduced. On the other hand, those systems need customized hardware, which can be an obstacle to wide deployment.

Step detection in these systems is performed by identifying local extrema of amplitude in vertical acceleration. One step contains exactly one maximum and one minimum in a short time interval. Our stairway module similarly monitors the amplitude of vertical acceleration. The difference is that, instead of trying to identify each and every step by scrutinizing vertical acceleration, we detect landings by focusing on large amplitude changes in acceleration, which are easier to identify.

2) Inertial navigation systems: Inertial navigation systems measure a user's acceleration and calculate the distance by double integration. The main challenge is to eliminate the effect of drift, which degrades the accuracy of the distance estimation over time.

NavShoe system [34] uses foot-mounted inertial sensors and applies ZUPT to achieve a significant reduction of errors. While a person is walking, one foot is in "stationary stance phase", while the other is in "moving stride phase". At every stationary stance phase, the velocity of the foot is zero. We have also used ZUPT in our elevator module. As the foot's velocity becomes zero at the stance phase, the elevator's velocity becomes zero when it stops on a floor.

Ojeda and Borenstein [35] have successfully traced a user's movement on a stairway using a foot-mounted IMU. Their system provides the elevation changes in meters, while our stairway module returns the number of floors a user has traveled. The direct measurement of a vertical displacement is possible because their IMU is mounted on the user's foot. It is hard to measure vertical acceleration accurately with a smartphone at an arbitrary position.

Xuan *et al.* [36] and Shanklin *et al.* [12] use smartphones to develop indoor positioning systems. Both systems do not reach the accuracy of the foot-mounted systems because of the lack of adequate mechanisms to handle the accelerometer drift. Moreover, inertial sensors in smartphones are more prone to errors. Our elevator module only considers movements in one direction, thus we can easily filter out noise in other directions. Our use of ZUPT in the elevator module also increases the overall accuracy.

3) Activity classification: Parnandi *et al.* [10] developed a smartphone-based system that focused on floor-level vertical location. From the real-time accelerometer data, the system classifies a user's current activity into one of four classes: elevator up, down, stairs up, and down. The system then estimates the number of floors that the user has traveled simply by dividing the total travel time by the time it takes to travel one floor. This system requires a training period to build a classifier for each activity and to calculate the average times needed to travel one floor.

However, this approach cannot take account of the speed variation of different elevators. Our elevator module can handle different speeds of elevators because we use real-time measurements to derive the user's vertical movement. Parnandi *et al.*'s work can be useful for our activity manager. We can employ their technique to determine whether a user is riding an elevator or walking on a stairway.

C. Hybrid approach

Infrastructure-based systems require a large number of sensors, and self-contained systems suffer from accumulated errors over time. Hybrid approaches are proposed to overcome the shortcomings of the two approaches. In hybrid systems, users' locations are primarily determined using IMU. The location estimation is then adjusted by information from infrastructure, such as RFID beacons [32] or Wi-Fi fingerprinting [37], [38]. Beacons in this case can be deployed in much coarser granularity compared to the systems purely based on infrastructure.

Our system can be viewed as a hybrid system because we primarily rely on IMU, but we anchor a user's location using the entrance information from Bluetooth beacons. Our system can be further extended by placing more beacons, one in every ten floors for example. Wi-Fi fingerprinting at such anchor points can also improve accuracy.

VII. CONCLUSION AND FUTURE WORK

This paper makes three contributions toward improving vertical accuracy of indoor positioning. First, we present the elevator module for tracking a user's movement in elevators. Second, we present the stairway module which determines

the number of floors a user has traveled on foot. Unlike previous systems that track users' foot steps, our stairway module uses a novel landing counting technique. Third, we present a hybrid architecture that combines the sensor-based components with minimal and practical infrastructure. The architecture strikes the right balance between accurate location information and the ease of deployment for the purpose of emergency communication.

We recognize that there are many hurdles to overcome before our system can be deployed in the real world. For instance, our elevator module assumes that the acceleration inside an elevator is mostly vertical. This will not be the case if a user happens to pace back and forth during the ride. Similar shortcomings also exist in the stairway module. The stairway module can produce false positives in some unusual cases. For example, a user can stop in the middle of a stairway, slowly turn around 180 degrees, and walk the rest of the stairway backward. This is highly unlikely, but it illustrates the general limitation of our approach that relies on behavioral norms. As future work, we plan to study the effects of various unusual behaviors, and explore possible solutions to address them.

ACKNOWLEDGMENT

REFERENCES

[1] U.S. Department of Transportation, "Next Generation 9-1-1," http://www.its.dot.gov/ng911/.

[2] J. Kim, W. Song, H. Schulzrinne, A. Zacchi, A. Jain, H. Chenji, C. Magnussen, C. Norton, W. Magnussen, I. Schworer, and K. Trinh, "The Next Generation 9-1-1 Proof-Of-Concept System," in *Proc. of ACM SIGCOMM 2008*, Aug. 2008.

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, 2002.

[4] G. Lammel, J. Gutmann, L. Marti, and M. Dobler, "Indoor Navigation with MEMS sensors," *Procedia Chemistry*, vol. 1, no. 1, pp. 532–535, 2009.

[5] J. Parviainen, J. Kantola, and J. Collin, "Differential Barometry in Personal Navigation," in *Proc. of IEEE/ION Position Location and Navigation System (PLANS)*, May 2008.

[6] Q. Ladetto and B. Merminod, "In Step with INS: Navigation for the Blind, Tracking Emergency Crews," *GPS World*, vol. 13, no. 10, pp. 30–38, 2002.

[7] S. Kerber and W. Walton, "Effect of Positive Pressure Ventilation on a Room Fire," National Institute of Standards and Technology, Tech. Rep. NISTIR-7213, Mar. 2005.

[8] "Building Pressure Diagnostics," http://contractingbusiness.com/enewsletters/cb_imp_9596/.

[9] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach," in *Proc. of the Sixteenth Annual International Symposium on Wearable Computers (ISWC)*, Jun. 2012.

[10] A. Parnandi, K. Le, P. Vaghela, A. Kolli, K. Dantu, S. Poduri, and G. Sukhatme, "Coarse In-building Localization with Smartphones," in *Proc. of the First Annual International Conference on Mobile Computing, Applications, and Services (MobiCASE)*, Oct. 2009.

[11] D. Grejner-Brzezinska, Y. Yi, and C. Toth, "Bridging GPS Gaps in Urban Canyons: The Benefits of ZUPTs," *Navigation*, vol. 48, no. 4, pp. 217–225, 2001.

[12] T. Shanklin, B. Loulier, and E. Matson, "Embedded Sensors for Indoor Positioning," in *Proc. of IEEE Sensors Applications Symposium (SAS)*, Feb. 2011.

[13] S. Dixon-Warren, "Motion Sensing in the iPhone 4: Electronic Compass," http://www.memsjournal.com/2011/02/motion-sensing-in-the-iphone-4-electronic-compass.html.

[14] P. Bhagwat, "Bluetooth: Technology for Short-Range Wireless-Apps," *IEEE Internet Computing*, vol. 5, no. 3, pp. 96–103, 2001.

[15] "BTstack," http://code.google.com/p/btstack/.

[16] "BlueCove," http://bluecove.org/.

[17] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[18] "TOPAZ," http://www.tadlys.co.il/.

[19] R. Agrawal and A. Vasalya, "Bluetooth Navigation System using Wi-Fi Access Points," Cornell University Library, Tech. Rep. arXiv:1204.1748, Apr. 2012.

[20] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Aug. 2000.

[21] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp, "WALRUS: Wireless Acoustic Location with Room-Level Resolution using Ultrasound," in *Proc. of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Jun. 2005.

[22] "Ubisense," http://www.ubisense.net/.

[23] T. Phebey, "The Ubisense Assembly Control Solution for BMW," http://www.rfidjournal.net/masterPresentations/rfid_europe2010/np/chandler_phebey_nov2_500_rtlsManu.pdf, Nov. 2010.

[24] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," in *Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Mar. 2000.

[25] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place Lab: Device Positioning Using Radio Beacons in the Wild," *Pervasive Computing, LNCS*, vol. 3468, pp. 116–133, 2005.

[26] "Ekahau," http://www.ekahau.com/.

[27] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, "Accurate GSM Indoor Localization," in *Proc. of the Seventh International Conference on Ubiquitous Computing (Ubicomp)*, Sep. 2005.

[28] S. Patel, K. Truong, and G. Abowd, "PowerLine Positioning: A Practical Sub-Room-Level Indoor Location System for Domestic Use," in *Proc. of the 8th International Conference on Ubiquitous Computing (Ubicomp)*, Sep. 2006.

[29] J. Chung, M. Donahoe, C. Schmandt, I. Kim, P. Razavai, and M. Wiseman, "Indoor Location Sensing Using Geo-Magnetism," in *Proc. of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Jun. 2011.

[30] B. Gozick, K. Subbu, R. Dantu, and T. Maeshiro, "Magnetic Maps for Indoor Navigation," *IEEE Transaction on Instrumentation and Measurement*, vol. 60, no. 12, pp. 3883–3891, 2011.

[31] "IndoorAtlas," http://www.indooratlas.com/.

[32] S. Yeh, K. Chang, C. Wu, H. Chu, and J. Hsu, "GETA sandals: a footstep location tracking system," *Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 451–463, 2006.

[33] E. Vildjiounaite, E. Malm, J. Kaartinen, and P. Alahuhta, "Location Estimation Indoors by Means of Small Computing Power Devices, Accelerometers, Magnetic Sensors, and Map Knowledge," *Pervasive Computing, LNCS*, vol. 2414, pp. 5–12, 2002.

[34] E. Foxlin, "Pedestrian Tracking with Shoe-Mounted Inertial Sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, 2005.

[35] L. Ojeda and J. Borenstein, "Personal Dead-reckoning System for GPS-denied Environments," in *Proc. of the IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*, Sep. 2007.

[36] Y. Xuan, R. Sengupta, and Y. Fallah, "Making Indoor Maps with Portable Accelerometer and Magnetometer," in *Proc. of the International Conference on Ubiquitous Positioning, Indoor Navigation and Location-Based Service (UPINLBS)*, Oct. 2010.

[37] H. Shin, Y. Chon, K. Park, and H. Cha, "FindingMiMo: Tracing a Missing Mobile Phone using Daily Observations," in *Proc. of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Jun. 2011.

[38] O. Woodman and R. Harle, "Pedestrian Localisation for Indoor Environments," in *Proc. of the 10th International Conference on Ubiquitous Computing (Ubicomp)*, Sep. 2008.