RescueMe: An Indoor Mobile Augmented-Reality Evacuation System by Personalized Pedometry

Junho Ahn and Richard Han
Department of Computer Science, University of Colorado, Boulder, Colorado 80309
Email: junho.ahn@colorado.edu

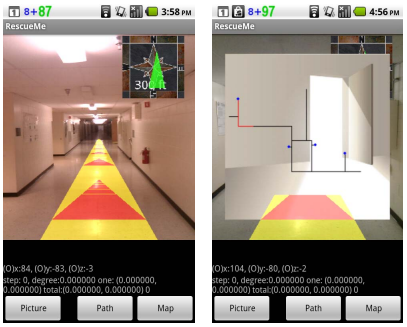
***Abstract*—Emergency applications have recently become widely available on modern smartphones. Nearly all of these commercial applications have focused on providing simple accident information in outdoor settings. AR in indoor environments poses unique challenges, due to the unavailability of GPS indoors and WiFi-based positioning limitations. In this paper, we propose the use of RescueMe, a novel system based on indoor mobile AR applications using personalized pedometry and one that recommends the most optimal, uncrowded exit path to users. We have developed the RescueMe application for use within large-scale buildings, with complex paths. We show how RescueMe leverages the sensors on a smartphone, in conjunction with emergency information and daily-based user behavior, to deliver evacuation information in emergency situations.**

I. Introduction

In 2009, the annual losses attributable to fire in non-residential buildings included 90 civilian lives and 1,500 serious injuries that occurred in 89,200 buildings, according to the U.S. FEMA (Federal Emergency Management Agency) [1]. Injury or death from fire within a building can be the result of asphyxiation caused by the inhalation of smoke, people being crushed by the walls collapsing, the spread of the fire that burns them before they can escape, or by crowds of people trampling each other in an effort to escape. The common factor is the time it takes for people to evacuate. If they can escape from the fire within the building in a sufficient amount of time, they can survive.

Our key motivation is to develop a mobile application that can help people evacuate quickly from buildings in emergency situations, such as fire. Our application, RescueMe, is based on existing built-in sensors on a smart phone and cloud servers via mobile social networking infrastructure. Our application helps users evacuate from dangerous places as quickly as possible, without the use of any additional installed infrastructure in the building. It also provides information about crowded areas (hallways, exit doors) within a building, so that users can be diverted from these areas to the fastest exits. Some existing emergency systems such as RFID or Wi-Fi help to locate users to provide a simple shortest path for evacuation. However, such systems provide no Augmented Reality (AR) guidance and require installing a costly infrastructure of hundreds if not thousands of devices. Our application leverages existing smartphones and cloud servers to provide information for locating users and identifying optimal evacuation paths.

A prototype of RescueMe is shown in Figure 1. As the user walks up and down the hallway, the AR tags indicates the exit path and the remaining distance. For example, suppose a person is visiting a hotel or unfamiliar office building, and the fire alarm sounds. A user would turn on RescueMe and follow the Augmented Reality indicator arrows to safety. RescueMe localizes the individual by taking image snapshots of landmarks, and matching them with well-known locations, using a service such as IQEngine. A user could take a photo of the room number or name on the hotel or office door that they're standing next to, which would localize them. After that, RescueMe would recommend the best individualized exit path with the shortest exit time for the user by using AR on the user's smart phone. While the user is following that path, if the server recognizes that the evacuating path is very crowded through other users' walking speeds, it notifies the user of a new evacuation path to take.



(a) Viewing 3D map (b) Viewing 2D map

Fig. 1. A screenshot of the mobile front-end component of the RescueMe mobile augmented reality system. The mobile component is running on an Android smart phone.

The contributions of this paper are as follows: an infrastructure-less emergency evacuation system that works indoors using image-based localization; a mobile guidance system using AR on smartphones to direct users to safety; a personalized pedometry system that estimates individual stride length to more accurately determine how far a user has walked; and a recommendation algorithm for the shortest delay path that reroutes users around crowded exits to uncrowded exits.

II. Related Work

RFID tags are explored [11] as a means for non-GPS localization. RFID tags, statically placed in a building beforehand with precise location knowledge can be used as navigational waypoints. The requirement to install such an infrastructure

978-0-7695-4624-7 2011

DOI 10.1109/APSCC.2011.26

within the environment is a major limitation of this approach. For the mobile phone localization, received signal strength (RSS) of Wi-Fi signals is the current preferred method [8], [4], [5]. Methods used are multilateration and fingerprinting. Multi-lateration requires at least 3 access points and precise knowledge of their locations to triangulate a user's location through RSS measurements. Fingerprinting requires the user to map the Wi-Fi signal propagation characteristics of the environment beforehand, creating a probabilistic heat map that may be consulted to compare RSS readings obtained by the user. All of the proposed techniques require a prior knowledge about the environment and so their application is less generalized.

Pedometer dead-reckoning (PDR) techniques provide a more general solution by not requiring any prior modification or knowledge of the navigational environment. Given a known starting location (such as that provided by GPS at the entry point to a building), users can probabilistically determine their navigational pathway through footstep detection and heading estimation. Such techniques are limited to the sensor placement and the sensor quality. The most accurate results require external sensor placement in the users shoes [8], [13], [3], e.g. yielding 0.5 m - 0.75 m accuracies in 8725 m2 of 3-story building space [13]. Using only the mobile smart phone device offers a more generalized solution, but suffers from limited sensor accessing, poor MEMs sensor quality, and looser coupling between the user's movement and the sensing capability. CompAcc [6] can also localize a mobile user by using a map-matching technique along with GPS, an accelerometer, and compass–but only in outdoor areas. This author's method cannot be used to localize a user within a building initially, because it requires a GPS connection to initially localize the user. Even if the user was initially localized within the building, this author's method could not be used, because it requires the periodic use of GPS to correct for localizing error. Stride estimation is done statically beforehand, contributing to the errors, depending on the variability of a user's gait. Dynamic stride estimation has been explored, but sensors are normally mounted on the foot in order to capture leg swing. Work to compensate for these errors employ the Kalman filter [8], [12], [11], Weinberg expression [10], or zero-velocity updates (ZUPT) [8], [3]. These three techniques minimize inertial drift and can predict actions based on prior event knowledge. In particular, the Kalman filter has shown wide usage, and the ZUPT has shown excellent correction accuracies.

III. System Design and Architecture

In this work, our goal is to design and develop an indoor augmented reality system for evacuation, by leveraging the sensing capabilities of smartphones and user behavior. In this section, we first highlight the system design requirements and challenges, then describe the overall system architecture and key components.

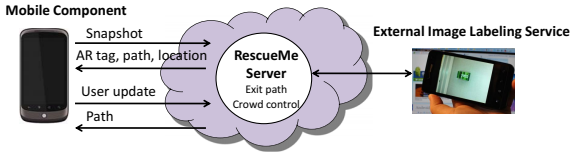


Fig. 2. Cloud based RescueMe architecture.

A. System Design Requirements and Assumptions

Supporting indoor augmented reality for evacuation calls for advances in a number of research areas, including accurate and efficient indoor localization, efficient indoor AR rendering and user-friendly interfacing, and effective evacuation functionalities.

Indoor localization is a key design requirement for indoor AR systems. Due to the diversity and dynamics of indoor environments and user activities, we need to identify a user's indoor location precisely (with fine granularity and robustness), with low latency, and without incurring too much overhead on the mobile device.

Efficient exit path recommendation is also important for evacuating a dangerous place in the building. The system needs to effectively recommend the best exit path, avoiding crowded places.

Our solution should not make unrealistic assumptions about the existence of extensive infrastructure to assist with any of the above tasks. Our system does not assume the existence of elaborate indoor localization systems. Even WiFi connectivity and WiFi localization are not necessary, so long as there exists an external wireless connection, such as 3G/4G. We assume only the capabilities and sensors common to most standard smartphones, e.g. today's iPhones and Android phones, that commonly have a camera capable of capturing continuous video, and accelerometers capable of measuring motion. We assume that the digital compass works indoors, which we've verified to be normally true in typical building settings. We do not assume the existence of gyroscopes on the phone, since not all smartphones support them. We also assume that RescueMe already has the map image of their building, the geographical orientation and actual size of their building, room doors' location with the room number and the walking paths, marked on the map. RescueMe supports evacuation services in all buildings without any additional cost, if the information is provided.

Our task is then to show how, under these assumed conditions, we can construct a system that successfully supports mobile augmented reality in typical indoor settings.

B. System Architecture

The architecture of RescueMe shown in Figure 2 consists of three logical components:

1) *Mobile component*: This component runs on the user's smartphone. This component implements four important functions. First, it provides inputs for image-based positioning by transmitting appropriate snapshots of nearby

room number. Second, it implements a pedometry-based localization algorithm to accurately determine the current position and orientation of the user as the user walks up and down the hallway. Third, it implements 3-D rendering of the building from the current perspective (location, orientation, etc.) using AR tags. Finally, it records user's walking patterns such as walking impact and a stride length in an outdoor area to predict user's walking distance in the building.

2) *External Image labeling Service*: RescueMe uses an external image labeling service to identify a room number from a room number snapshot taken by the mobile component. This service may use any of the well known techniques, such as computer vision and crowd sourcing, for label identification. In our current prototype, we use a commercial image labeling Web service called IQ Engines [2].
3) *RescueMe Server*: This component implements our recommendation algorithm for the best exit path for each user. RescueMe selects the path with the shortest time to evacuate. This is not always the shortest distance path. For example, if the shortest distance path is too crowded, RescueMe observes the delays of other users in the system, and finds another path that is faster, rerouting and notifying each affected user in real time.

IV. System Components

In this section, we describe the design and implementation of each component of the RescueMe system. The RescueMe Server coordinates all interactions among the system components, and is responsible for maintaining a spatial database that contains the building's map and metadata that will be rendered by the AR component on the mobile client. The RescueMe Server is implemented as a Web server that exposes its functionality externally.

A. Pedometry-based Localization

A pedometry-based dead-reckoning (PDR) system for indoor localization has been implemented in the Java language for an Android Nexus One smart phone, running on the Android 2.3 (Froyo) operating system. The NexusOne smart phone employs two tri-axis motion sensors, which we leverage for PDR localization: an accelerometer and a digital compass. The accelerometer we use for both step detection and stride estimation. The digital compass is used to determine the user's heading-direction from which the direction of motion is estimated.

1) Footstep Detection: To detect a step, we follow a multistep signal processing method. First, the *x-axis*, *y-axis*, and *z-axis* accelerometer values are normalized by removing the effect of gravity through a mean removal operation. Secondly, we calculate a moving average of the normalized accelerometer signal. A moving average serves to both minimize errors induced by varying the user orientation of the phone in 3-axes, as well as remove unwanted high frequencies from the data. Thirdly, we examine both positive and negative peaks in the processed accelerometer signal trace. A genuine footstep will generate both a bound and a rebound phase corresponding to the foot striking and pushing against the ground. A footstep is therefore characterized by a positive peak closely followed by a negative peak in the accelerometer data. If the amplitude difference between a positive and a negative peak is greater than a set threshold, a step is recorded. Because footstep frequency is roughly 2-3 Hz, we require the temporal distance between a positive and successive negative peak to be ≤ 300 ms. Peak amplitude difference is required to exceed a threshold of 1.0 g. Both values were experimentally determined and verified as well-performing choices. The addition of a dynamic threshold adaptation scheme was tested and found to perform worse than the static scheme, which we possibly attest to the low maximum android accelerometer sample rate, and so the static threshold method is reported.

2) Learned Gait Pattern Between Outdoor Places: RescueMe measures a daily-based stride length for the user, when he is walking in outdoor locations in normal situations, in order to build a personalized stride length which is used in the RescueMe application. For this measurement, the application needs to recognize when the user is walking down a street, hovering around one area, or remaining in the same place. We found a study which determined that people usually visit the same locations at similar times. The study which investigated the areas visited by 100,000 subjects found that they exhibit habitual space-time movements, with reasonably small variation [14]. We used this finding about human behavior, based on similar space-time patterns to predict the location of users. Based on the study, we stored the regularly visited places on the phone. When the user stays at the same place or hovers around the place for a certain length of time, we do not measure the user's stride length. However, when the user walks between outdoor places located a certain distance from each other, RescueMe measures the gait pattern of the user by using the accelerometer and GPS. The walking patterns are adjusted according to a user's age, sex, height, weight, health, etc [15], [16], [17]. These characteristics of a user affect the amplitude of the accelerometer. RescueMe estimates a stride length of users by using the adjusted amplitude measurement of the accelerometer. This outdoor measurement increases the accuracy of our estimation of the user's walking distance within the hallway of a large-scale building, without requiring the use of GPS indoors.

3) Daily-based Personalized Stride Estimation: Our system provides a distance estimation between each detected footstep. In order to estimate stride lengths, we have developed a stride estimation function that is based upon the historical measurements of the impact of a user's footstep, which corresponds to the detected amplitude measured between the positive and negative peak in an accelerometer signal trace.

To further refine our stride estimation function, we however need to first correct for a periodic measurement error found in the calculation for the length of a footstep. We examined the *x-axis*, *y-axis*, and *z-axis* of the accelerometer that measures the number of footsteps. We found that the orientation of the

phone is strongly related to the measurements of the three axes, and that one of the axes will best determine the actual number of footsteps we are measuring in our algorithm. To determine the correct number, we take the maximum measurement of the three axes. The equation for measuring footsteps is shown in Equation 1. Actually, we calculated the sum of the footsteps by running the equation periodically, every few seconds, to collect an adequate number of footsteps from the user and to capture the variable footstep measurements based on the orientation of the phone.

$$Footsteps = Max(Footsteps_x, Footsteps_y, Footsteps_z) \quad (1)$$

We also measured the walking impact of the user's footsteps as he walks from one outdoor location to the next. The impact strength is calculated by measuring the amplitude between the positive peak point and the negative peak point on an accelerometer. We calculated the average impact of the user's footsteps from the data from the three axes, regardless of the orientation of the phone, by using the following Equation 2.

$$\begin{aligned} Strength(t) = Avg(&Length_{(HighPeak_x, LowPeak_x, t)}, \\ &Length_{(HighPeak_y, LowPeak_y, t)}, \\ &Length_{(HighPeak_z, LowPeak_z, t)}) \end{aligned} \quad (2)$$

We calculated the user's stride, using the number of footsteps and the total walking distance taken during one measurement time period. The walking distance is computed between two locations from GPS connections. We divide the walking distance by the number of footsteps to measure the user's stride as in Equation 3. We pair this stride measurement with the user's corresponding walking impact measurement to predict a user's normal stride (e.g., whether walking, running, slow, fast, etc.).

$$\begin{aligned} Stride(t) &= WalkingDistance_{(GPS1_{t-1}, GPS_t)} \\ &/ \quad footsteps_{(t-1,t)} \end{aligned} \quad (3)$$

4) Map-Matching-Based Orientation and User's Direction Estimation: Upon successful detection of a user's footstep, the simultaneously polled digital compass sensor data is examined in order to determine the direction a user is heading based upon the footstep detected. For each footstep we read the angle of the compass on the building map to locate the user. The true angle direction the user is heading is not measured correctly by the mobile phone as he walks, because the orientation of the phone varies as it is shaken by the user, holding it in his hand. For example, if the angle direction the user is walking is 90 degrees, the phone normally detects the angle's measurement within a range between 60 and 110. If we use the raw angle data read from the phone, the walking path's measurement data will vary from the user's true path. The map-matching technique reduces this problem. This technique is also used to detect the hallway turning points on the building map. When a user turns left or right, RescueMe changes the angle of the compass to the angle of the map. To recognize the turning points, we created a buffer with limited window size to save previous measured angles stored on the phone as the user walks. If the measurements of more than half of the angles are different from the user's previously determined direction, we store the user's current direction as this new angle measurement. Part of the map-matching algorithm rounds phone angle measurements that are close to the actual map angle measurements to the map numbers observed in the buffer. Also, if a new angle measurement is not a possible angle measurement of the map, or if it is not within a plausible range of acceptable map measurements, RescueMe does not change the current angle's measurement to the new one. We discuss the actual experiment for developing this estimation algorithm in the Experimental Results section.

5) Solution for incremental measurement problem: RescueMe provides two methods for solving an incremental problem generated when using pedometry. Pedometric measurements are generated by the accelerometer of the phone. The accelerometer scans for noise, caused by the shaking or jiggling of the phone as the user walks. Such noise affects the measurements used to calculate the walking distance of the user. The longer the walking time, the more the error may increase. We use two different methods to correct for such errors: map-matching combined with user's direction-path change, and manual re-selection of a room number or re-taking a picture of a room number. With the first method, the RescueMe client recognizes the user's location when the user changes the angle path he is walking. RescueMe compares this change in location with the building map, and if it matches, resets the user's previously recorded location to this new correct one–the actual hallway turning point on the map. In the second method, the user decides to reset his own location manually by selecting a room number close to where he is standing. Using this method, he can reset his current location by either re-taking a picture of the room number now closest to him, or by selecting from a list of room numbers already stored on the server. In the second instance, the user selects from a list of room numbers on his current floor that were sent to him originally by RescueMe when he first accessed the server.

B. Evacuation path recommendation

The RescueMe application provides a recommended exit path to a user, based on the shortest path [7] from the user to the exit location. For example, the user requests an evacuation path, using his smartphone, when he finds himself in a dangerous place or situation in a building. The user takes a picture of the room number nearest to him, and the phone client then sends the server this information. The server calculates exit times for possible exit locations, based on the walking speed of the user and the distance between the user and the exit doors. The formula for the exit time calculation is shown below in equation 4. After comparing the various exit times, the server sends back an exit path with the lowest time to the user's phone. That path will be the shortest possible path the user could take to exit the building.

$$ExitTime = WalkingDistance / WalkingSpeed \quad (4)$$

The RescueMe application helps users to avoid crowded hallways in a building. For example, when there is a crowd of people around an exit door, trying to evacuate quickly, most of them will not be able to exit in a timely and safe manner. If the user begins to approach such an exit location, based upon RescueMe's original recommendation of the shortest path, the user's walking speed will slow down significantly. Once the application detects the user's reduction in walking speed, it will do two things. The RescueMe application will recalculate the next shortest exit path for the user, and then it will share this information with other users via the RescueMe server. Thus, we have enhanced the previous formula to accommodate this situation. The application measures the delay time of the crowd trying to exit by the door. This newly calculated exit time uses the original exit time from the starting point to the crowd point (the location at which the user's speed is reduced) plus the delay time, as shown in the formula 5 below.

$$\begin{aligned} ExitTime &= WalkingDistance_{Start,CrowdPoint} \\ &/ \; WalkingSpeed + DelayTime \end{aligned} \quad (5)$$

The application estimates a delay time for the user that is calculated using the detected reduced speed and the distance from the crowd point to the recommended exit door. Specifically, this delay time is calculated by dividing an "exiting" distance from the crowd point (the user's location) by the reduced speed as in equation 6.

$$\begin{aligned} DelayTime &= WalkingDistance_{CrowdPoint,Exit} \\ &/ \; ReducedWalkingSpeed \end{aligned} \quad (6)$$

C. Augmented Reality

We have implemented AR on the smart phone to support the overlaying of path-based tags on the hallway. The tags not only give a 3D depth perspective, but are also 3D themselves and can be rotated to face the user regardless of the viewing angle. We accomplish this by exploiting the OpenGL library as explained below. The application differently renders the 3D information depending on the behavior of users. We describe how to render a view of the textured 3D model using the OpenGL library.

1) 3D navigation: Our AR emergency application provides the user with information about the path to take to evacuate a building, using a 3D presentation of evacuation tags for viable exit paths as in Figure 1(a). The tags are displayed in succession on a recommended exit path within a hallway, thus indicating the direction and distance the user needs to follow. We use a depth-based presentation of arrow tags to help users easily recognize the distances they must walk to follow the designated exit paths. When looking down the hallway, a user will know the walking distance he needs to go, because the application displays multiple arrow tags in succession up to the point in the hallway in which the user needs to turn or exit. Meanwhile, as the user walks down the hallway, the application displays a different tag in a corner of the phone that includes directional symbols, conveying information on which way to turn, along with text showing the remaining walking distance. Thus, the user can identify whether he is following the correct direction while he is walking. The directional arrow tags and the additional symbol/distance tag will enable the user to find a viable exit path and proceed in the right direction on that path until he gets to the exit door.

2) 2D Map: The RescueMe application also provides a 2D map, which users can access when they press a button, so that they can see their current location within the building and all of the recommended exit paths on this map as in Figure 1(b). Thus users can know where they are headed and where they are located in the building, while walking. This 2D map, which was implemented with this application, is based on OpenGL. The users can use this map anytime and even evacuate the building by using it. Users can also change the size or the position of the map by manipulating the touch screen on the smart phone. When walking in a building that has intricate or complicated paths, users can see their current location and a portion of the map at any time.

D. Image-based Positioning

RescueMe requires accurate determination of the user's location in an indoor environment. The Global Positioning System (GPS) cannot be used in indoor environments, since line-of-sight communication between GPS receivers and satellites is not possible in an indoor environment. Radio frequency (RF) positioning systems that use WiFi and Bluetooth radios on smartphones provide limited accuracy (1 - 3 m) due to the complexities associated with indoor environments, including a variety of obstacles (people, furniture, equipment, etc.) and sources of interference and noise from other devices [9].Therefore, we investigated the use of other positioning technology in RescueMe. RescueMe uses a commercial image labeling Web service, called IQ Engines [2], to determine the user's initial starting position whenever the user takes a picture of a room number close to him. IQ Engines uses a combination of computer vision and crowdsourcing to tag a photo with a label describing the content of the image. When an image is submitted to IQ Engines, the image is first processed by a computer vision system in an effort to provide an accurate label. If the computer vision system cannot identify the image, then IQ Engines passes the image to its crowdsourcing network for analysis and tagging. According to IQ Engines, the time to return a label for an image varies from a few seconds for the computer vision system, to a few minutes for the crowdsourcing system. In the RescueMe application, a user's location is determined in the following way. First the user takes a picture of a room number above a door to provide the RescueMe server with his location. The picture is sent to the IQEngines server, which then identifies all text within the picture. The IQEngines server sends back the text of the room number to the RescueMe client, the user's phone. If the server finds the exact same number in its database, it sends the room number, location, and other associated metadata for that room back to the user. In the RescueMe application, buildings' door locations are expressed using the following dimensions: floor level, and x-axis and y-axis positions of the doors in

Fig. 3. Orientations of Nexus One

the building. If the text that is returned to the phone client contains errors, such as the omission of a letter, the addition of an incorrect letter, or the substitution of a correct letter with an incorrect one, the RescueMe client will use an edit distance algorithm [18] to determine the correct room number. The client then queries the user to see if the room number is correct. When the user gives an affirmative answer, this information is sent to the RescueMe server, and the correct room number, and thus the location of the user, is determined.

V. Experimental Results

A. Training Personalized Pedometry

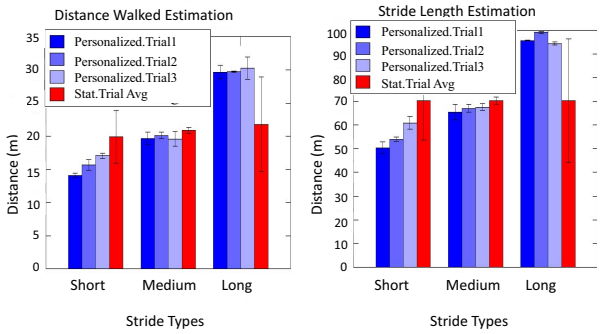
TABLE I
COUNTING FOOTSTEPS DEPENDING ON THE ORIENTATION

Orientation	footsteps	x-axis	y-axis	z-axis	maximum steps
A	470	380	192	**460**	z-axis
B	472	**464**	252	452	x-axis
C	468	72	332	**461**	z-axis
D	476	160	**452**	380	y-axis
E	452	416	**432**	232	y-axis
F	460	**456**	412	372	x-axis

1) Historical pedometry measurement: We conducted an experiment for correctly counting footsteps, depending on the orientation of the phone using Equation 1, in order to use this information in the user's personalized stride length measurement. We collected this data by measuring the impact of each footstep of the user, while he was walking to different locations in outdoor areas, using GPS and an accelerometer. The average stride length was calculated by using the number of footsteps as in Equation 3. The phone was carried in a pocket or in a bag, and positioned in different orientations as in Figure 3. We measured the number of footsteps by using the three axes of the accelerometer as in Table I. In the case A and case C, the z-axis of the accelerometer strongly affects the detection of the number of footsteps. The x-axis strongly affects the detection in cases B and F, while the y-axis strongly affects the detection in cases D and E. Through this experiment, we evaluated the accuracy of the proposed algorithm as to whether it was detecting the number of footsteps correctly, regardless of the orientation of the phone.

2) GPS duty-cycling experiment for personalized pedometry: The user's personalized stride length is continually measured only when the user is in outdoor areas, by using GPS whenever the user is walking from one outdoor area to another. The GPS sensor obtains a user's location from satellites, but the antenna for GPS on the phone is not robust enough to connect instantly to a satellite, with one try. Thus, we conducted an experiment to determine the maximum delay time for the phone's GPS connection in various situations. In the delay experiment, the average delay time for connecting was about 7.1 seconds and the standard deviation was about 5.9 seconds. The maximum delay time was recorded to be 15 seconds in this experiment. We also performed a test of setting the duty-cycle time for accessing GPS to every 5 minutes and found that the application consumed less than 15 percent of the battery power in 10 hours time on the Nexus One.

B. Personalized Pedometry-based Localization



(a) Overall distance walked. (b) Stride length estimation.

Fig. 4. Personalized and static-based estimation methods are compared for individual stride and overall distance. Overall distance includes both step detection and stride estimation errors.

TABLE II
STEP DETECTION ACCURACY

Stride Length	Measured Steps	Actual Steps	Error (%)
Short 1	28	30	-6.67
Short 2	29	30	-3.33
Short 3	28	30	-6.67
Regular 1	30	30	0.00
Regular 2	30	30	0.00
Regular 3	29	30	-3.33
Long 1	31	30	3.33
Long 2	30	30	0.00
Long 3	32	30	6.67
$\|Avg.\|$			3.33
Std. Dev.			4.41

1) Accuracy using Personalized Pedometry: The pedestrian localization via pedometer and heading estimation systems were implemented and tested in Java on a Nexus One smart phone running the Android 2.3 (Froyo) operating system. User tests to evaluate pedometry step detection, stride estimation, and the combination of step detection and personalized stride length measurements into an overall distance walked estimation were carried out. Additionally, differing types of users were simulated, varying from an "engaged" user who wishes to learn how to use the system to obtain the best performance, to the "casual" user who is not interested in performance and so uses the system in a careless manner.

To evaluate pedometry system step, stride, and distance accuracy, a user was tasked to walk three trials of 30 paces in testing each of three different types of user strides. The

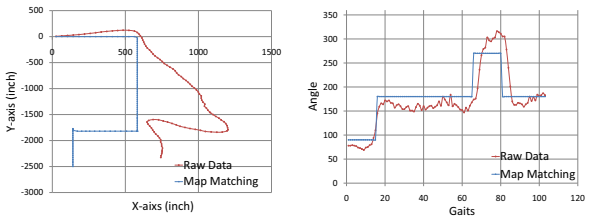
first stride type is a "short stride," which is a deliberately short stride of about 50-55 cm. The second stride type is a "medium stride," which is a comfortable stride length of about 65-70 cm, which is natural for most users. Lastly, the "long stride" is one that is the largest the user can manage without jogging or running; a length of about 95-100 cm. The results of these nine trials can collectively be seen through Table II and Figures 4(b) and Figure 4(a). Table II shows our system to have an overall step detection error rate of 3.33 percent. In fact, for longer tests that we omit here, step detection accuracy was shown to improve with the number of strides taken.

In Table II, short strides have a tendency for under detection, while long strides are prone for over detection. This is due to the static threshold used for detection, which is tuned for the normal stride length scenario. An adaptive step threshold detection scheme was implemented and tested, but suffered a poorer performance than the static method. We theorize this counterintuitive result to be due to the accelerometer's 10 Hz maximum sampling rate on the Nexus One smart phone not providing a smooth enough data curve for the adaptive algorithm to leverage effectively.

Figure 4(b) compares the static and personalized stride length estimation techniques. The resulting stride lengths represent the average stride length of each of the 9 user trials completed, calculated by the overall distance measured divided by number of steps detected, but not actually taken. This removes any additional step detection errors that might be present and allows a pure comparison of stride length estimation. The personalized stride estimation generated 2.33 percent error, while the static stride estimation suffers 17.06 percent error. Interestingly, because the static method was tuned for the medium stride length, its average error actually outperforms that of the personalized method on the same data set. A point of note is the extreme accuracy of the long stride under the personalized estimation scheme. The error bars are almost too small to be seen, averaging to 99.6 percent stride length accuracy for this stride type. This excellent accuracy is most likely due to the flatness of the alpha correction function for large positive peak amplitudes.

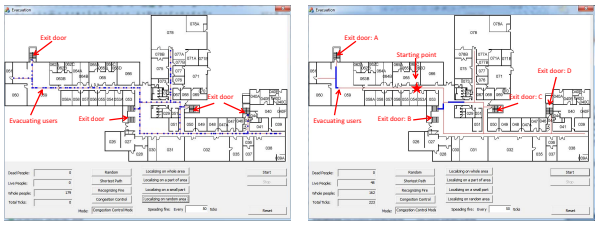
Figure 4(a) addresses the combination of error from step detection as well as stride estimation techniques. An overall walk distance is measured by our system and is compared against the ground truth walked distances. Figure 4(b) shows that in some cases, e.g. personalized trial 1 for a short stride, an error in step works to reduce the error stride. However, in most cases, if both kinds of error are present they combine with one another, which is evident by the increase in overall error from stride (2.33 percent) and step (3.33 percent) to distance walked (3.43 percent).

2) Map-matching experiment for RescueMe: RescueMe uses a map-matching technique to locate users by using pedometry. If the hand holding the phone is shaking, the orientation sensor on the phone reads an erratically changing angle, so that the direction of the walking path drifts from the correct path. We performed an experiment for recognizing a user's walking path direction, using a map-matching technique combined with personalized stride length measurement. Figure 5(a) shows that the walking path drifted a considerable distance from the actual path, without the use of the map-matching technique. When the map-matching technique with the personalized stride length measurements was applied, RescueMe fixed the drifting walking path to the correct one, i.e. the corrected path produced by map-matching exhibited right angles correctly corresponding to the hallways corridors where we walked, as shown in this diagram.



(a) Drifted path (b) Drifted angle.

Fig. 5. Experiment for Orientation and Heading Estimation



(a) Randomly-distributed people (b) People in a crowd

Fig. 6. Simulation for RescueMe

We also conducted an experiment for predicting the walking path angle, when a user changes his direction. In this experiment, we created a constantly replenished five window buffer to determine the heading-direction angle, by storing the angle measurement in each window for each step the user takes. Each time three of the five measurements are similar to each other within the buffer, RescueMe stored the current direction as the new angle measurement of the heading direction of the user as shown in Figure 5(b). The combined use of these two algorithms successfully utilized a match-mapping technique with pedometry to localize the user on the map.

C. RescueMe simulation for the exit path recommendation

TABLE III
SIMULATION: RANDOMLY DISTRIBUTED PEOPLE

Simulation	Random	Shortest Path	RescueMe
Simulation 1: time (tick)	947	240	240
Simulation 2: time (tick)	673	563	368

We conducted two types of simulations to evaluate the length of time it takes for people to evacuate an emergency situation in a building. We contrasted three scenarios–one with randomization (no algorithm), one using the shortest-exit path algorithm, and the third, using the RescueMe algorithm.

The first simulation involved 179 people who were deployed randomly within the building at the start of the simulation. For the second simulation, 162 people were deployed in one specific area of the building. The simulated people were programmed to move one step every tick at the same speed, as they moved towards the various exit door choices. If more than one person arrives at the same place, the exit speed of each person is delayed by the others adjacent to them.

In the first simulation for evacuating the building, we initially investigated the case where people were distributed randomly. Figure 6(a) shows that 179 people were randomly deployed in the hallways of the building. We then measured the evacuation time of these people, using three different scenarios. First, the simulated people were randomly evacuated to an exit door of "their choice." Second, the people were evacuated to the shortest-path exit door. Third, the RescueMe application evacuated the randomly distributed people to exit doors that were calculated to take the shortest amount of time. Simulation 2 in Table III shows the total evacuation time for all the people for each scenario. These findings show that in the first case, the random-choice (non-algorithm) method resulted in it taking 947 ticks for all the people to exit the building. However, it took 240 ticks for all the people to evacuate in both the second and third scenarios. Thus, both algorithmic methods worked equally better than the first method. For people evenly or uniformly dispersed throughout a building, knowing the shortest path or shortest amount of time (RescueMe) to an exit door is best.

The second simulation was conducted to show how best to improve the exit time when people are unevenly distributed throughout a building, such as when they are gathered as a crowd for a presentation in one location in a building. Simulation 2 in Table III shows that RescueMe provides the best result for evacuating a crowd of people from within a building. All of the people could evacuate the building within 368 ticks using this method, whereas the shortest path method took longer (563 ticks), and the random method even longer. People using the shortest path method often ended up at the same exit door, since it was the shortest path from the shared crowded area. Each individual person's exit time was delayed by the people between them and the exit door. However, for the people in the RescueMe scenario, most of them were able to exit a door uncrowded by other people, as shown in Figure 6(b). In this case, as shown in the diagram, RescueMe recommended one of three exit doors: A, B, and C; thus the group of people were dispersed equally to allow them to evacuate quicker through a less crowded exit door.

VI. Conclusions

This paper has presented RescueMe, a novel system that uses pedometry and indoor mobile augmented reality, to recommend the best exit path with the shortest time to users needing to evacuate a building in emergency situations. Through an evaluation of this application, we have shown how our system leverages the sensors on a smartphone, in conjunction with personalized daily walking stride length estimation and emergency information, to support a timely evacuation. Our practical pedometry algorithm with personalized stride estimation provides high positioning accuracy.

Acknowledgment

This work was supported by a grant from the National Science Foundation, CCF 1048298. The authors wish to thank Youjin Seo, Luvi, Jamie Williamson, and Cathy Kerry for helping with some experiments or sharing ideas.

References

[1] Federal Emergency Management Agency (FEMA). http://www.fema.gov/.

[2] Iq engines: Image recognition and visual search. http://www.iqengines.com.

[3] S. Beauregard. Omnidirectional pedestrian navigation for first responders. In Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on, pages 33 36, 2007.

[4] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In Proceedings of the 6th ACM conference on Embedded network sensor systems, SenSys '08, pages 337 350, New York, NY, USA, 2008. ACM.

[5] M. Emiliano, C. T. Cory, R. Ashwin, C. Tanzeem, L. Zhigang, and C. T. Andrew. Darwin phones: the evolution of sensing and inference on mobile phones. MobiSys 2010, 2010.

[6] I. Constandache, R. Roy Choudhury, and I. Rhee. CompAcc: Using Mobile Phone Compasses and Accelerometers for Localization. In INFOCOM, 2010.

[7] Dijkstra, E. W. "A note on two problems in connexion with graphs". Numerische Mathematik, 1959, 1: 269?271. doi:10.1007/BF01386390.

[8] C. Fuchs, N. Aschenbruck, P. Martini, and M. Wieneke. A survey on indoor tracking for mission critical scenarios. Pervasive and Mobile Computing, In Press, Corrected Proof:-, 2010.

[9] Y. Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. IEEE Communications Surveys & Tutorials, 11(1):13 32, 2009.

[10] S. Ladstaetter, P. Luley, A. Almer, and L. Paletta. Multisensor data fusion for high accuracy positioning on mobile phones. In Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, MobileHCI '10, pages 395 396, New York, NY, USA, 2010. ACM.

[11] L. E. Miller, P. F. Wilson, N. P. Bryner, M. H. Francis, J. R. Guerrieri, D. W. Stroup, and L. Klein-berndt. Rfid-assisted indoor localization and communication for first responders. 2006.

[12] F. Seco, A. Jimenez, C. Prieto, J. Roa, and K. Koutsou. A survey of mathematical methods for indoor localization. In Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on, pages 9 14, 2009.

[13] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In Proceedings of the 10th international conference on Ubiquitous computing, UbiComp '08, pages 114 123, New York, NY, USA, 2008. ACM.

[14] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, Understanding individual human mobility patterns, Nature, vol. 453, 2008.

[15] Schmitz A, Silder A, Heiderscheit B, Mahoney J, Thelen DG. Differences in lower-extremity muscular activation during walking between healthy older and young adults. J Electromyogr Kinesiol 2009; 19: 1085.91.

[16] Callisaya ML, Blizzard L, Schmidt MD, McGinley JL, Srikanth VK. Sex modifies the relationship between age and gait: a population-based study of older adults. J Gerontol A Biol Sci Med Sci 2008; 63: 165-170.

[17] Brown LA, Gage WH, Polych MA, Sleik RJ, Winder TR (2002a) Central set influences on gait: age-dependent effects of postural threat. Exp Brain Res 145:286-296

[18] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physice-Doklady, 1966. 10:707-710.