Computer Communications 73 (2016) 108–117

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Smartphone positioning in sparse Wi-Fi environments

Wasiq Waqar*, Yuanzhu Chen, Andrew Vardy

Memorial University of Newfoundland, 838 Pinewood Crescent, 702 Ottawa, Canada

ARTICLE INFO

Article history:
Available online 25 September 2015

Keywords:
Localization
Indoor positioning
Context-aware applications

ABSTRACT

Indoor localization using mobile devices such as smartphones remains a challenging problem as GPS (Global Positioning System) does not work inside buildings and the accuracy of other localization techniques typically comes at the expense of additional infrastructure or cumbersome war-driving. For such environments, we propose a localization scheme which uses motion information from the smartphone's accelerometer, magnetometer, and gyroscope sensors to detect steps and estimate direction changes. At the same time, we use a Wi-Fi based fingerprinting technique for independent position estimation. These measurements along with an internal representation of the environment are combined using a Bayesian filter. This system will allow us to reduce the amount of training required and work in sparse Wi-Fi environments. We test our approach in two real-world environments to show the benefits of incorporating user motion for indoor localization.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In the past, most of the attention was given to Location Based Services (LBS) in outdoor environments as GPS played the dominant role in localization. Recently, we are seeing a paradigm shift in the mobile applications market, where indoor LBS is being considered the new frontier. Due to the increasing number of mega size multi-level constructions like airports, shopping malls, universities and other facilities, people tend to spend more time indoors. Research shows that people only spend 10–20% of their time outdoors [1] and more than 70% calls originate from indoors which indicates great potential for indoor LBS.

The proliferation of smartphones is motivating researchers to look at other ways for more reliable and energy efficient indoor positioning of users which have a reasonable tradeoff between accuracy, reliability, cost, and scalability. To minimize deployment and infrastructure costs, different technologies are being explored. Indoor positioning is challenging as GPS may not work inside buildings so most common solutions take advantages of existing RF (Radio Frequency) infrastructures like Wi-Fi access pointa (AP) and cellular base stations. There are several ways in which RF signals can be used for positioning. It is not easy to model the radio propagation in indoor environments because of diffraction, scattering, shading, severe multipath, low probability for availability of line-of-sight (LOS) path, and specific site parameters such as floor layout, moving objects, and numerous reflecting surfaces. There is no single good model for an indoor radio multipath characteristic so far. Different techniques have different advantages and disadvantages. Hence, using more than one type of positioning algorithm at the same time may yield better performance. There are different triangulation, proximity or fingerprinting based algorithms available which deal with the indoor positioning problem in various ways.

On the other hand in robotics, inertial sensors, laser range-finders, and computer vision are used to provide accurate localization without the requirement of fixed infrastructure. Mobile devices, such as smartphones and music players, have recently begun to incorporate a powerful yet diverse set of sensors. These sensors include GPS receivers, microphones, cameras, proximity sensors, magnetometers, temperature sensors, accelerometers, and gyroscopes. In the future, other sensors like altimeters, barometers, etc., may be included into these devices. Inertial measurement units (IMUs) like accelerometers and gyroscopes are being embedded in most of the latest smartphones. Accelerometers measure 3D linear accelerations of the device whereas gyroscopes give angular velocities. Most modern smartphones also include a magnetometer for raw magnetic readings and heading information. Using these sensors one can estimate the user's motion and characterize their activity as, for example, walking, standing, jumping, running etc. User motion can then also be used to keep track of position via dead reckoning.

Problems arise when using RF based positioning schemes in environments where RF signals are sporadic or sparsely deployed. Due to the placement of APs (Access Points) and cell towers, there might be areas where RF signals are not available. Similarly there may be disruption in the RF signals due to limits on radio range, energy resources, and other sources of noise. In such environments, it is better to incorporate additional information from IMUs for localization with opportunistic RF based position correction.

* Corresponding author. Tel.: +1 6137162997.
E-mail address: wasiq.waqar@mun.ca (W. Waqar).

http://dx.doi.org/10.1016/j.comcom.2015.09.002
0140-3664/© 2015 Elsevier B.V. All rights reserved.

Our main contributions to address the above challenges are summarized as follows:

- **We identify an opportunity to use sensor-based dead-reckoning and opportunistic Wi-Fi positioning for localization using smartphones in areas where there is sparse Wi-Fi coverage.** Our approach does not require the installation of additional infrastructure.
- **We developed and used an iOS app on the Apple iPhone 4 to evaluate our technique. This app was tested in the tunnels of Memorial University of Newfoundland which have very limited Wi-Fi coverage.**

The subsequent sections expand on each of these contributions, beginning with a short related research overview followed by our proposed idea, evaluation, and conclusion.

2. Related work

Smartphone accelerometers have been used in some mobile localization schemes in an assistive or collaborative manner. In Surround-sense [9], they are used as one of the parameters for the fingerprint, whereas CompAcc [2] uses them to count the number of steps taken to estimate the distance travelled by a pedestrian.

In [11] the authors gave a novel particle filtering based scheme for indoor positioning which does not rely on any infrastruction and uses only the sensors from the smartphones. But their system is not stand alone as their design requires a centralized system. In [6] the authors dont rely on any Wi-Fi but depend on a more accurate step counter and turn detections for position accuracy. However in buildings where multiple floors have the same layout, this scheme might fail and some kinds of auto correction measure has to be taken. In other work [3,5,10,12], researchers have used accelerometer data to detect human activities such as walking, standing, climbing stairs, jogging, etc. A short overview of related work is covered in [14].

3. System architecture

In probabilistic robotics, a *belief* is the internal knowledge of the robot or a system about the state of the world. In our case state means the location of the subject in our environment. States cannot be measured directly, but we can represent and estimate the probability that the system lies in each possible state. We use the term *belief* to refer to the conditional probability distribution over all possible states. This distribution assigns a probability to each possible hypothesis with regards to the true state. State x_t is generated stochastically from state x_{t-1} meaning that the belief at time t is calculated from its past belief at time $t-1$. The most general algorithm for calculating beliefs is given by the *Bayes Filter* algorithm. Algorithm 1 depicts Bayes Filter which is a recursive Bayesian state estimation technique utilized in mobile robotics and other applications [13].

Algorithm 1: The general algorithm for Bayes Filtering.

Input: $u_t, z_t, bel(x_{t-1})$

1: **for all** x_t **do**
2: $\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$
3: $bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t)$
4: **end for**

Output: $bel(x_t)$

This algorithm is recursively applied at every iteration when belief $bel(x_t)$ needs to be calculated from $bel(x_{t-1})$. Bayes filter possesses two essential steps. In Line 2, it processes the control u_t. It does so by calculating a belief over the state x_t based on the prior belief over state x_t and the control u_t. u_t in our case is the motion captured from the motion model. This step of the algorithm is also called *prediction* [13].

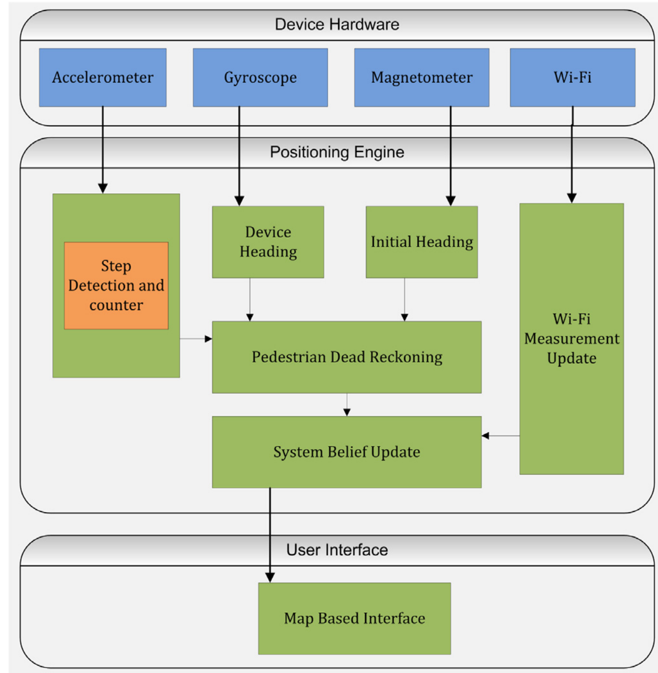

Fig. 1. Block diagram of the proposed system.

The second step of Bayes filter is called the measurement update. In line 3, the Bayes filter algorithm multiplies the belief $\overline{bel}(x_t)$ by the probability that measurement z_t may have been observed. It does so for each hypothetical posterior state x_t. To compute the posterior belief recursively, the algorithm requires an initial belief $bel(x_0)$ at time $t = 0$. If we are ignorant about the initial condition we can initialize using the uniform distribution.

3.1. Design overview

Fig. 1 shows the block diagram of our proposed system. In our localization scheme we divided our map into a grid. The center of these grid cells are referred to as anchor points which have known physical coordinates (x, y). The grid space between two anchor positions determines the resolution or granularity of the positioning system. The system state variable x_t indicates the anchor point that is closest to the current position. The initial belief of the system is assumed to be uniform as the system will not know where the user is positioned. The on-board magnetometer is noisier compared to gyroscope when giving heading estimation [15]. Therefore, we only use the magnetometer to initialize the orientation of the user and calibrate the gyroscope. This is one of the assumptions of our system that we ask the user to face one of the corridors (potential path where the user can walk). After this initialization/calibration process we keep track of the heading using the gyroscope. We use the step counter [15] to estimate the distance travelled and the gyroscope to estimate the direction in which this distance is travelled. As shown in Fig. 1, accelerometers are used to detect the steps taken.

3.2. Motion model

Using a step counter and gyroscope one can estimate the user's recent trajectory and then predict $\overline{bel}(x_t)$. Step detection is the automatic determination of the moments in time at which footsteps occur. If one wants to use accelerometer data to detect just the instant motion of the device, one needs to be able to isolate sudden changes in the movement from the constant effect of gravity.

Peak detection is a method which calculates the steps from the 3-axis accelerometer readings. A threshold value can be used to detect a peak. If changes in acceleration are too small, the step counter will discard them. The step counter can work well using this algorithm, but sometimes it can be over sensitive. The algorithm that we chose for our step counter is inspired by an analog pedometer [17]. The algorithm used for our step counter using mobile phone accelerometer is available in [15]. There are several other algorithms available for step counters but most of them are primarily for accelerometers attached to the foot, hip or other body part.

The iPhone 4 has a 3-axis gyroscope which can measure angular velocities about the axes. The Core Motion Framework of the Apple iOS SDK also provides us access to built in functions which manage and keep track of the device's attitude after the application starts. Rotation around the z-axis is called yaw and at the start of the application it is calibrated with the initial stable magnetic heading. The comparison and performance of estimating direction with gyroscope compared with magnetometer is discussed in [15].

3.2.1. Belief update strategies

To study our motion model, a set of anchor points is maintained and the probability distribution over this set is represented by $bel(x_t)$. Figs. 11 and 12 show the test environments and the positions of all anchor points.

In the time interval $[t-1, t]$ the user advances from position x_{t-1} to position x_t. The step counter and gyroscope report back the relative change in position (x_{rel}, y_{rel}). As we know the initial heading and current heading of the user, we can determine the user's direction of travel. So from the last position and the new position we can determine x_{rel} and y_{rel} which are distances travelled in the x-direction and the $y-$direction with respect to our map.

$$x_{rel} = \alpha \cos(\theta + \beta) \tag{1}$$

$$y_{rel} = \alpha \sin(\theta + \beta) \tag{2}$$

where θ is the initial orientation of the device during initialization, β is the yaw of the device and α is the step length.

The corresponding relative motion parameters (x^*, y^*) for the given poses x_{t-1} and x_t are calculated in lines 1 and 2. These basically come from the known positions in the map. The function $norm(a, b)$ implements an error distribution over a with zero mean and standard deviation of b which was empirically chosen as 4m. The motion model is used as step 2 in our Bayes filter implementation.

3.3. Wi-Fi fingerprinting

In classic fingerprinting algorithms, vectors of Received Signal Strength (RSS) measured in online phase and offline phase are directly compared to each other. The nearest neighbor method simply calculates the Euclidean distance in signal space between the live RSS reading and the fingerprint. A major drawback of using this technique is that different devices, because of their hardware and software (sometimes devices of the same make and model), report different RSS values which may differ from the RSS stored in the database. This will degrade the performance of the positioning system. In contrast, rank based localization [8] uses only ranks of the RSS values because the rank information is less sensitive to any bias and scale.

Fig. 2 shows the block diagram of the rank based fingerprinting algorithm. In this algorithm, first the RSS values measured in the online phase from different APs are first sorted from strongest to weakest. Ranks $(1, 2, 3, \ldots)$ are assigned to APs based on the position in the sorted vector. Rank 1 is given to the strongest AP, meaning with the highest RSS value. Rank vectors are created from the fingerprints

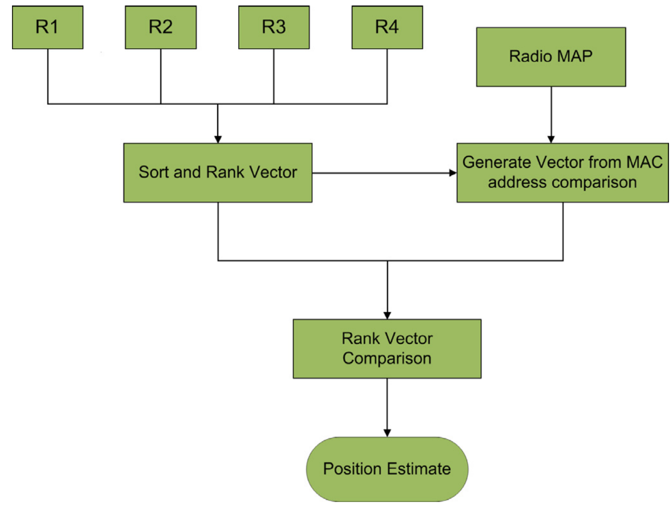

Fig. 2. Block Diagram of the Wi-Fi Rank based fingerprinting.

stored in the database. Ranks are assigned based on the MAC address and rank of AP in the online phase. Then this vector is also sorted strongest to weakest keeping the rank assigned to them. In ideal cases, the sorted ranked vector from online phase and sorted ranked vector from offline phase will be identical hence showing perfect similarity.

In case an AP which was in the online phase was not found in the database, the rank vector created from the database is padded with 0, to achieve the same length as the rank vector from the online. Other techniques, including via Gaussian kernel [4] which calculates the likelihood of an anchor point using the RSS value similarity between two vectors, also face the dimension mismatch problem. In real indoor environments the dimension of the fingerprints of different anchor points vary considerably. If simple likelihood calculation mechanism (e.g., Euclidean distance or cosine similarity) is used, mismatching could lead to large positioning errors.

Spearman's footrule distance measures total elementwise displacement between two vectors. It is similar to the Manhattan distance for quantitative variables. According to [7] Spearman's footrule perform the best amongst other similarity measures. Assuming u_k is the rank of the kth element in vector U, v_k is the rank of the kth element in vector V and n is the number of elements in vectors U and V, Spearman's footrule distance can be computed as follows:

$$D_s = \sum_{k=1}^{n} |u_k - v_k|.$$

The similarity measure above return the scores for every anchor point. The anchor point with the lowest score is considered the best match. Ideally using k smallest reference points to calculate the estimated position yields a better result. The author In [7] proposed a p-center problem to estimate the final position estimate. In the rank based technique the distribution of scores will differ because of several reasons. The number of APs visible in the querying scan and position where the scan was done affects the distribution of the scores. For instance if the scan is done at a corner where 20 APs are visible compared to another location where only 5 APs are visible, the distribution of scores will differ a lot. The random test on 13 anchor points in the Engineering Building was done. It was noted that the accuracy of the position estimate was independent from the score distribution. Fig. 3 shows the maximum and minimum score distribution and Fig. 4 shows the normalized entropy of the score distribution. As the user initiates the application, the belief is uniformly distributed. Entropy is a measure of the uncertainty associated with a random variable and is also referred to as the expected value of the information contained

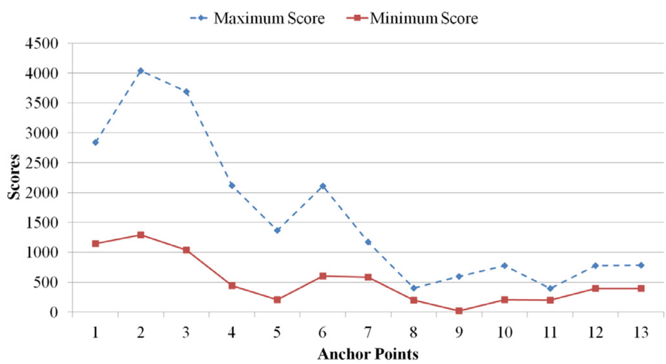

Fig. 3. The minimum and maximum scores at different anchor points.

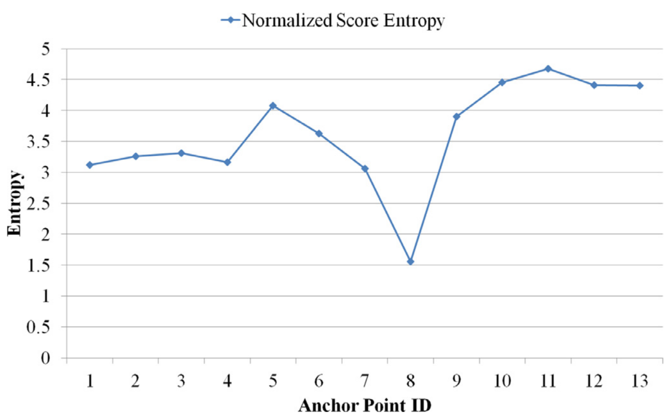

Fig. 4. After normalizing the scores, entropy is calculated.

in a message, which in our case is the belief. At position 5 to 9 the accuracy was under 8 m whereas 1–4 and 10–13 the error was greater then 8 m. The best match at positions 6 and 8 were estimated the correct position but both the entropy and min-max distribution does not infer a trend. The distribution of scores tells us that our certainty of our position estimate is not dependent on the score distribution. Hence we used a different approach to use Wi-Fi for position correction. We assign weights $w1$, $w2$, and $w3$ to the best 3 matched anchor points only if they are all within 2 hop neighbors to each other. Otherwise we ignore the Wi-Fi scan. It means that when the Wi-Fi localization module estimates the best 3 matches, the weights are assigned only if each anchor point is at least 2-anchor point distance to any of the other two. For our experiments we assign 0.4, 0.3 and 0.2 weight to the three best matched positions.

4. Performance evaluation

We will explain our experimental methodology, settings, scenarios, and results in this section. Our main experimental goal is to measure the benefit of using motion information to track and position the user in an indoor environment.

4.1. Methodology

The system evaluation contains multiple phases. The first phase is to test the performance of our step counter which is a major part of our motion model. After checking the accuracy we can determine if it is good enough to be used in our motion model. The accuracy and precision of our motion model is then tested in two different indoor environments.

The second phase is the evaluation of our measurement model. By analyzing the performance metrics, we can determine if it can be used for opportunistic measurement update. Furthermore, it is important to test our system in an environment which has sparse Wi-Fi coverage. Next, we explore the benefit of using motion for localization and tracking and analyse the advantages of using rank based Wi-Fi in sparsely distributed Wi-Fi environment. We measure the benefit in the following aspects:

- *System performance*
  Hypothesis 1: *The system accuracy and precision of motion assisted indoor positioning is better than other localization systems in sparse Wi-Fi environment*. Most of the current indoor technologies used are essentially Wi-Fi only. Their performance is related to very laborious training of the environment. Our system's motion model should be able to accurately position and track a user walking in an indoor environment. The turns in the environment are helpful in shortlisting the user's possible positions. Although the error while walking in the same direction accumulates, turning into another corridor should reduce this error. We argue that using the motion model alone is sufficient for short-term user tracking. Wi-Fi based corrections are beneficial, especially in sparse Wi-Fi environments where there are only a few APs. Our system will require only few Wi-Fi training points in these environments and would perform much better than other Wi-Fi dependent indoor localization schemes.
- *Cost*
  Hypothesis 2: *The system training and maintenance cost can be reduced*. The system training effort is reduced in a sparse Wi-Fi environment as fewer survey points are needed for data collection. The motion model does not need any training. More importantly, if the environment has unique features in terms of corridor layout and number of turns, the system will require fewer Wi-Fi landmarks and can be more dependent on the motion model alone. When the indoor environment changes (e.g., Wi-Fi infrastructure or environment layout alteration), the RSSI fingerprints database has to be updated or even re-generated from scratch in order to adapt to such changes. If the number of such survey points are fewer the cost to update will be lower compared to other Wi-Fi based systems.
- *Scalability*
  Hypothesis 3: *The system can work in different indoor environments*. The system is scalable as it can be quickly adapted to any environment, both with dense Wi-Fi and with limited Wi-Fi coverage. Only environment maps are needed with internal representation of possible user position points. Moreover the resolution of the grids can also vary and the accuracy would not directly depend on the grid resolution. As accuracy depends more on the stepcounter rather than how dense the grid is.
- *Robustness*
  Hypothesis 4: *The system can recover from false position estimates*. Unusual movement of the user may confuse the system. For example, if the user is walking in a circle, it is possible the system might become more uncertain about its position. We argue that our system over time can recover from this uncertainty.

4.2. Experimental settings

Experiments and evaluations of our motion model, measurement model and hybrid localization scheme were carried out in two contrasting environments at Memorial University. The first was part of the 2nd floor of the Engineering Building. The space was divided into a grid using a 3×3 m cell size. 42 positions were selected within the hallways for the anchor points. 33 of these anchor points were surveyed for Wi-Fi data and a fingerprint was created for each anchor points. The survey points are those anchor points where Wi-Fi training was done and we have a Wi-Fi fingerprint available. The anchor points are possible locations the user can be in the environment.

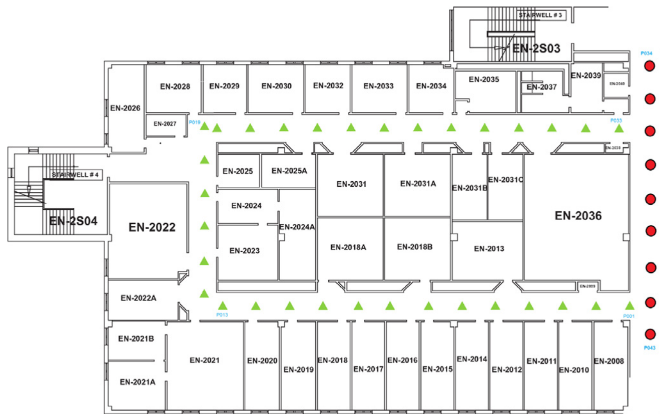

Fig. 5. Map of the Engineering Building. Green triangles are the anchor points where data has been collected and the system has fingerprints for those locations. Red circles are untrained areas.

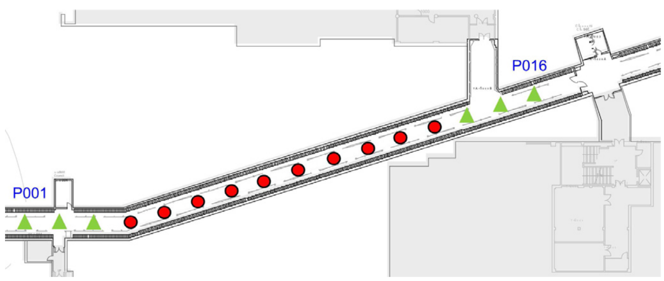

Fig. 6. Map of part of the university tunnel. Green triangles are the points where Wi-Fi is sporadically available and red discs are positions where no Wi-Fi is available. Fingerprints for locations with green triangles are available.

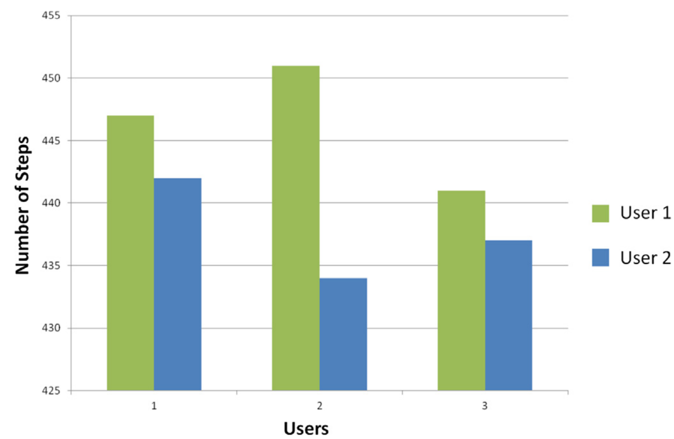

Fig. 7. Number of steps detected when walked 500 steps .

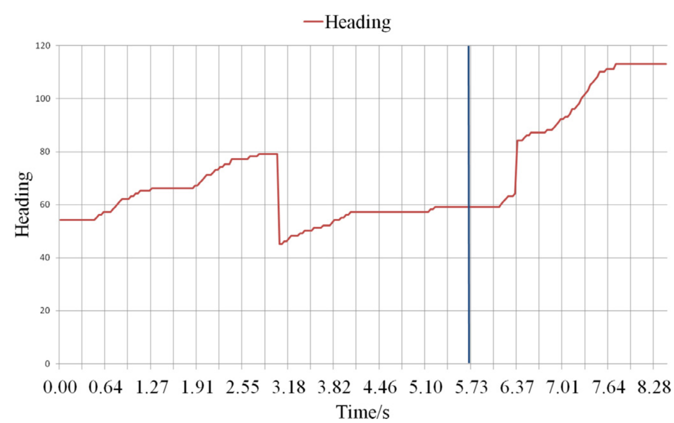

Fig. 8. Magnetic heading readings when walking from a center of a corridor to the intersection of corridors in the Engineering Building. .

The distance between two anchor points is nearly 6 steps (3.5 m), so belief is chosen to be updated after every 6 steps in this environment. Fig. 5 shows the map of the Engineering Building field test environment.

The second environment is the Tunnel system which connects different buildings of the university. There is no Wi-Fi coverage provided for the tunnels. Fig. 6 shows the map of the tunnel system. The only Wi-Fi signals available are at entrance positions. Hence the areas of Wi-Fi AP visibility is very limited and also sporadic in nature. The Engineering Building has more sharp turns, whereas the tunnel has smaller turns. The distance between two anchor points here is 5.5 m. Therefore the belief update happens after every 9 steps. Most of the commercial pedometers choose step length as $0.413 \times h$, where h is the height of the user. In our experiments step length is kept at 0.69 m.

The major assumptions for our experiments are as follows

- The user is always located in the areas for which the anchor points are defined in the system.
- The device is always pointing in the direction of the user's motion.
- The user walks close to the corridor's center.

4.3. Motion model evaluation

The step counter was evaluated by two different users by walking 500 steps holding the device in the hand. The experiment was repeated 3 times by walking the same path. Fig. 7 shows the accuracy of the step counter. Intuitively it can be seen that the step detection depends a lot on human gait. Apart from this it also depends on how a user is holding the device. Some users tend to hold the device in a more stable manner while others sway their hands while walking. But this problem can be solved by multiplying a user specific scaling factor to the threshold of step detection. The accuracy of the step counter was comparable to other commercial step counters available on Apple's app store. Therefore it was considered reliable enough to use in our motion model.

Fig. 9 shows the magnetic map of the environment to show more deviations near the corners compared to the middle of the corridors. When the application starts, the gyroscope has to be initialized to the orientation of the user in the environment using magnetometer. The magnetometer is noisy, a small experiment was done to see the stability of the magnetic heading readings in the environment. It has been noted that there is greater magnetic instability and interference in the corners and intersections. The standard deviation of magnetic readings in the major parts of the corridors is 9 degrees whereas it is 21 degrees near or at the corners. In order to correctly identify the initial orientation, we set a check that in the initialization phase if the magnetic readings have a standard deviation more than 12 degrees. If so the initialization process is repeated. Fig. 8 shows the heading readings when approaching an intersection. The horizontal axis describes the time in seconds.

In order to test the motion model the user walked in the corridors of the Engineering Building. Although in this experiment the Wi-Fi integration was disabled but only those anchor points were considered in which we had Wi-Fi fingerprints available. To denote the true position of the user in the map a small human figure marker is used to show the true location and also the direction of walking. As the application starts the algorithm first calibrates for the heading of the device using the magnetometer. Once the calibration is done, the gyroscope keeps track of the orientation of the user while walking. The circles in the screenshots in Fig. 10 show the belief distribution of the system. The anchor point with the highest probability will

Fig. 9. Magnetic map of Engineering Building.

Fig. 10. Screenshots of motion model in Engineering Building.

show the biggest circle and all the remaining anchor points will have circle sizes relative to it as the probabilities are normalized before belief distribution is shown to the screen. This way it is easier to visualize how the belief distribution is shifting and converging. It can be observed from Fig. 10a that all circles are of equal size as in the beginning the belief is uniformly distributed. From Fig. 10.b it can be observed that during the application start-up the initial orientation has been detected as towards the right (East) with respect to the map, hence the probability distribution shifts towards those corridors which have a pathway towards East. Fig. 10(c–f) shows how the

Fig. 11. Screenshots of motion model in Engineering Building continued from Fig. 10.

probability distribution shifts along the direction where the user is walking. Although at this point the algorithm is uncertain where the user is positioned. However, it can keep track if the user turns back and starts moving in the opposite direction.

The user keeps walking towards the end of the corridor and turns right. Fig. 11a shows that the probability suddenly converges to one of the anchor points near the corner. This happens because the algorithm detects that the user has taken a right turn. So that anchor point will have a higher probability to be the true position which will have the same relative motion from a neighboring anchor point. Fig. 11b shows that user is tracked as the probability shifts in the same way as the movement of the user. In Fig. 11c two corner anchor points have almost equal probability as the belief was updated during the turn. The belief is updated every 6 steps taken by the user. This update frequency was chosen to correspond with the distance between two anchor points. The user then turns back start walking the same path the user came from. Fig. 10(d–f) shows that the belief of the system shifts correctly with the motion of the user.

In a similar experiment, we also considered other anchor points in the area which were depicted as red circles in Fig. 5. These anchor points do not have fingerprints as no Wi-Fi data was collected at these points. Other Wi-Fi only based solutions would not work very well in these conditions. Luo et al. [7] did experiments under same conditions. Their error increased from 2 to 9 m when they moved from trained area to untrained area. Fig. 12(a–f) and Fig. 13(a–f) depicts the screenshots of the positioning application when it walks in the untrained area.

Fig. 12. Screenshots of motion model in Engineering Building in unmapped regions.

Fig. 13. Screenshots of motion model in Engineering Building in unmapped regions continued from Fig. 12.

4.4. Entropy of belief

In another experiment the user was asked to walk in the corridor with our localization app in the trained areas of Engineering Building. Fig. 14a shows the heat map of the probability distribution over time. The *x*-axis describe the *i*th update of belief. The position IDs are listed on *y*-axis where the color intensity shows the probability of being at each location. The belief at x36, x64 and x88 are examples where

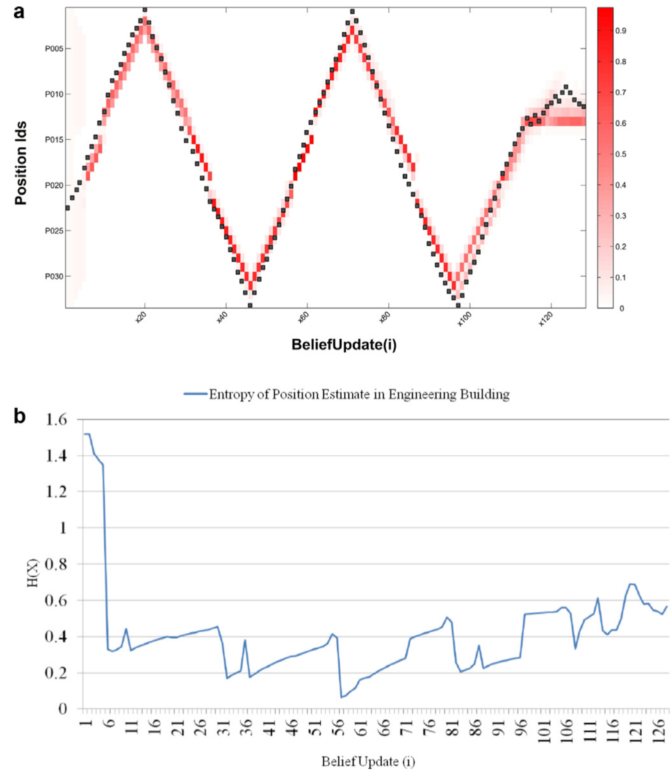

Fig. 14. (a) Motion model heat map at Engineering Building with dense Wi-Fi coverage. Black annotations describing actual user position. (b) Entropy in the Engineering Building .

the position correction happens due to turning. Overall it can be seen that the position is tracked pretty well along the path of the user. From belief update x112 to x128 the user changed his direction of walking after a few steps a couple of times creating a to-and-fro user trail. It can be observed in the heat map that the uncertainty starts to increase as the probability distribution spreads out. Thus, a malicious behavior by the user in terms of walking in circles and moving to-and-fro in the corridor over short distances might confuse the belief system.

Fig. 14b shows the entropy of the same heat map. At x5 the entropy falls greatly due to a turn. Initially the probability was uniform so the entropy was maximum but as soon as the user turned the belief became more certain due to the recognition of a corner. Every time the user turns a corner, the uncertainty decreases and we can see a drop in entropy. After x112 the entropy increases, showing the confusion caused by user motion.

4.5. Rank based Wi-Fi measurement model

Our Wi-Fi localization scheme returns similarity scores between the current measurement and every anchor point which has been surveyed for stored Wi-Fi data. The lowest score is considered the best match. To test the rank based fingerprinting technique we assumed that the best match anchor point is the estimated position. We tested this in our Engineering Building at each anchor point. The error was recorded by logging the distance between the ground truth and the estimated output position. Fig. 15 shows the cumulative error distribution. The mean error was about 4.1 m. We compared our system with the Wi-Fi based localization scheme by Luo et al. [7] which uses a different fingerprinting approach for localization. They employ the Gaussian kernel, which is commonly used to calculate the likelihood between an RSSI fingerprint in system anchors and the live RSSI measurement to generate likelihood candidates. The top-*k* candidates

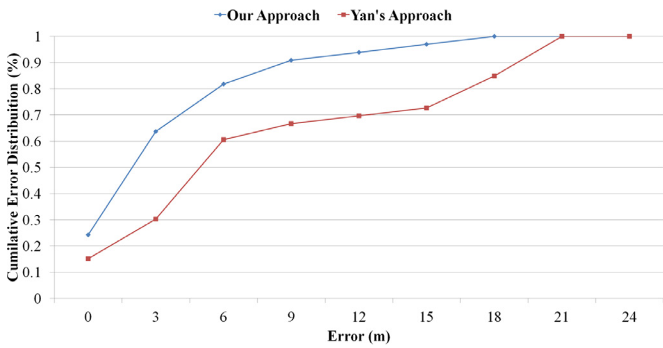

Fig. 15. Cumulative error distribution of the rank based fingerprinting in Engineering building.

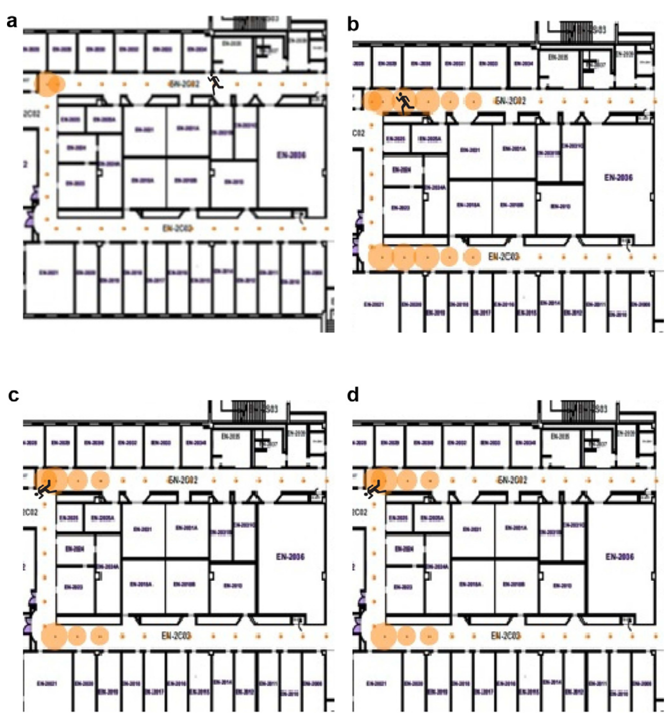

Fig. 16. Recovering from an erroneous position estimate due to motion model.

are then used to determine a final position using the vertex p-centers problem.

Fig. 16 describes a situation in which the Wi-Fi measurement was updated to a wrong location. This test was done in the Engineering Building, where the Wi-Fi APs are denser and the Wi-Fi environment is not sparse, meaning that at most of the locations, similar APs are visible. As in our Wi-Fi positioning module we create a rank of the APs visible to compare it with a fingerprint, due to fluctuations of the radio signals it is possible that it updates and positions the user at a wrong location. Similarly, there can be a scenario in which the error accumulates over time due to the motion of the user. In Fig. 16a, it can be seen that, the user is present near the middle of the North corridor but the position 16b and Fig. 16c. But after the turn, it again converges. Fig. 16d shows that the motion model would be able to recover in this situation. Although in a sparse Wi-Fi environment, where the APs at one area are distinct compared to other areas, the error due to Wi-Fi will be smaller.

4.6. Performance in a sporadic Wi-Fi environment

To test our system in an environment which has sparse Wi-Fi coverage, we chose the university tunnel system which has no Wi-Fi

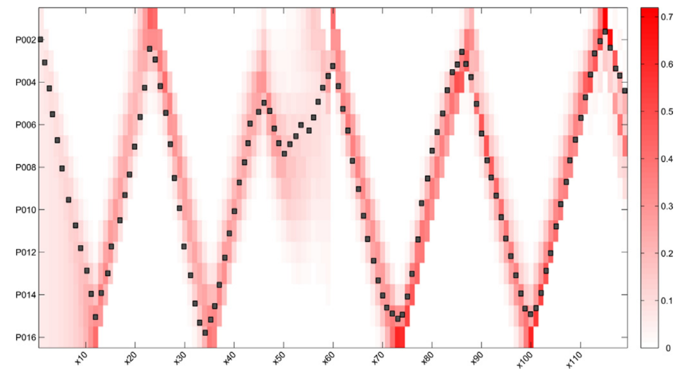

Fig. 17. Heatmap of motion model in the tunnel.

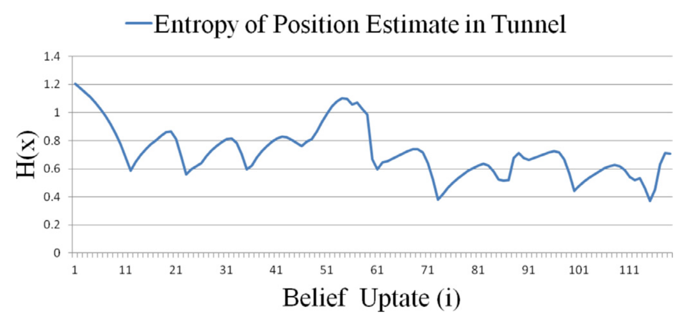

Fig. 18. Entropy in the tunnel.

available but sporadic signals are available at the different entrances of the tunnels from different buildings. Fig. 6 shows the map of one such section of the tunnel. This figure shows 16 anchor points from one entrance to another. All neighboring anchor points are equally distant from each other. It is assumed that initially the system does not know the user's true position. Initializing with a Wi-Fi scan can initialize user position if the user is in one of the entrance areas.

Fig. 17 shows the heat map of the user's walk in the tunnel. On horizontal-axis we have the belief updates and on vertical-axis we have the 16 anchor points. We annotated the map with approximate actual position of the user to compare the belief distribution with the movement of the user. At x0 the belief is uniformly distributed but from x0 to x12 we can see that the belief slowly converges. From x12 to x45 the probability distribution is not that scattered and position estimates are more confident. From x45 to x60 the probability distribution becomes less reliable as the user changes his direction more frequently similar to the test done in Engineering Building. At x60 the Wi-Fi measurement update is triggered. At this point it detects P001 as the most likely position. The probability distribution shifts heavily towards that position as we give higher weight to the anchor points with higher Wi-Fi similarity. In the tunnels the Wi-Fi is sporadically available in only P001–P004 and then P015-P016 as described before. No Wi-Fi is detected in any anchor points between them. Hence when the Wi-Fi update step is triggered, due to the diversity of visible AP's between these two regions, the position correction has smaller error.

Fig. 8 shows the entropy of the belief in the tunnel. If we compare the entropy plotting of Engineering Building and tunnel it can be observed that the entropy in the tunnel does not drop as much as compared to the entropy in the Engineering Building. This is because the tunnel lacks sharp turns as compared to the Engineering Building. Although the accuracy from the most probable position estimate is comparable in both locations the certainty is less because of the absence of sharp turns. At x51 to x59 it can be observed that due to the to-and-fro motion in the same corridor the entropy increases. It sharply decreases again at x60 when Wi-Fi measurement update is triggered Fig. 18.

Next, we will consider the hypotheses mentioned in Section 4.1 in light of our experimental results.

- *System performance*
  Hypothesis 1: *The system accuracy and precision of motion assisted indoor positioning is better than other Wi-Fi only localization systems in sparse Wi-Fi environment*. As it can be seen from the heatmaps of both environments that the system tracks and positions the user with fairly good accuracy regardless of the density of Wi-Fi coverage. For our experiments the accuracy in the Engineering Building was under 4 m whereas in the tunnels it was around 6 m on average. The best-performing but intensively trained Horus system [16] has a 0.7 to 4 m average positioning error using 100 Wi-Fi scans and much smaller grid space (1.52 m and 2.13 m). Generally for our system a single accuracy figure can not be given as it depends upon the shape and size of the environment. Sharp turns help reduce positioning error estimates and long corridors accumulate errors. The second factor is the amount of Wi-Fi landmarks available for position correction.
- *Cost*
  Hypothesis 2: *The system training and maintenance cost can be reduced*. We tested our system in two different environments. One had dense Wi-Fi coverage and had training data available for all the anchor points. On the other hand in the tunnel environment, the Wi-Fi was sporadically available at only 6 locations. No survey was done for those anchor points which had no Wi-Fi coverage so they were treated as untrained anchor points. As different areas in such environments have distinct Wi-Fi visibility, this can be exploited to our advantage to correct the position only and rely more on human motion for positioning. In our motion model evaluation, we observed that in the environment where there are more turns, the position estimate is better than the environment with less turns. Turns help the motion model to detect change in orientation and inherent map matching in the motion model help to converge the belief. Due to less reliance on Wi-Fi, minor changes in Wi-Fi infrastructure will have less impact on the system performance.
- *Scalability*
  Hypothesis 3: *The system can work in different indoor environments*. We tested our system in two completely contrasting environments. One had sharper turns with denser Wi-Fi coverage and the other had less turns but sparse Wi-Fi environment. The grid size in both the environment was also different as it was 3 m in the Engineering Building and 5.5 m in tunnels. This system is more scalable than other indoor positioning systems as it would require less training and would even work in sporadic Wi-Fi environments where Wi-Fi only systems would fail.
- *Robustness*
  Hypothesis 4: *The system can recover from false position estimates*. In both the environments during our field test we confused the system by walking in to-and-from (Fig. 14b and Fig. 17) fashion to create more uncertainty in the belief. When triggered Wi-Fi updates remove this ambiguity. If Wi-Fi is updated in the wrong location, it can be recovered in two different ways. The first one is due to the motion model the belief starts to become more uncertain. It starts to converge again if there is a turn which can uniquely position it. The second way it can be recovered is when another Wi-Fi update is triggered. Although Wi-Fi update can be erroneous too, but there is a chance that the error is reduced.

5. Future work

Wi-Fi based localization technologies are relatively robust and accurate compared to other indoor localization technologies. One of the main factors for these technologies to be popular is that the infrastructure often already exist. The RSSI fingerprinting based schemes perform better than triangulation based schemes because they do not depend on specific signal propagation models. However, the system performance greatly depends upon the rigorous training process and regular system maintenance in the form of regular fingerprint updates. These regular fingerprint updates are required if there has been any changes in the environment in terms of replacing a access points or moving furniture etc. In addition to that, these systems do not work in areas where Wi-Fi coverage is sparsely distributed.

These shortcomings can disable above mentioned localization systems. Moreover, because of high system overhead in terms of training data and cost of war-driving, we believe there is a need for more efficient and cost effective techniques. We believe that reducing training and maintenance cost and increasing the system robustness are very promising research directions.

In addition, we see that the current generation of smartphones have various embedded sensors including motion sensors like accelerometers and gyroscope. Although GPS receivers are present in most smartphones, they are of no help indoors. But magnetometers can be used to detect direction and heading. We recognize the opportunities presented by these sensors to detect human motion and the possibility to incorporate this knowledge to help position users in an indoor environment. Hence, we would also not rely on any external infrastructure except Wi-Fi coverage which is likely to exist in many environments.

In this work, the primary contributions are evaluation of a motion assisted indoor positioning system for an indoor environment especially focused on sparse Wi-Fi coverage. We can use ideas from robotics in which a belief is maintained about the possible position estimate rather than relying on dead reckoning to output one final pose estimate. The distance moved by the user is calculated by the number of steps taken and then estimating the user trail by calculating the direction of each step. The user trail is matched with possible path signatures from the environment map using the motion model. The best match yields a higher likelihood for position estimate. Hence more distinct features in terms of turns and direction of corridors will give us higher accuracy. But in environments with similar corridors in terms of length and orientation, we will get multiple hypotheses for the user's position. In this situation we use Wi-Fi based position correction. Our Wi-Fi position estimation techniques uses rank on the visible APs based on their strengths rather than the actual RSSI values. This technique has an additional benefit of being device independent as different manufacturers of networks cards have different standards for RSSI values but rank information is invariant to any monotonic increasing transformation (bias and scale) [8]. Wi-Fi AP's is used as landmarks to update the position belief when it is required by the system to update its position. This can happen after a fixed number of steps to avoid error accumulation due to the motion model.

One of the major benefits of this system is cost effectiveness. The initial training required by doing war-driving and collecting Wi-Fi data decreases significantly. Although the tradeoffs between accuracy and cost of training will depend on the environment, we can see the real benefit in such a system in sparse Wi-Fi coverage area.

Based on these principles we built a prototype mobile application for the iPhone and conducted experiments to evaluate it. Our experiments showed encouraging results and indicate motion assisted positioning as a viable option for indoor environments. The system is scalable and more cost effective than Wi-Fi only schemes because it requires less training.

References

[1] Indoor LBS And Hyper - Local Content Is the Next Gold Rush for Mobile Commerce, 2010, (http://www.indoorlbs.com/2010/06/indoor-lbs-and-hyper-local-content-is.html). (accessed Oct. 2012).
[2] I. Constandache, X. Bao, M. Azizyan, R.R. Choudhury, Towards mobile phone localization without war-driving, in: Proceedings of the 29th Conference on Information Communications (Infocom), 2010, pp. 2321–2329.

[3] A.M. Khan, Y.-K. Lee, S. Lee, T.-S. Kim, Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis, in: Proceedings of the 5th International Conference on Future Information Technology, 2010.
[4] A. Kushki, K.N. Plataniotis, A.N. Venetsanopoulos, Kernel-based positioning in wireless local area networks, IEEE Trans. Mob. Comput. 6 (6) (2007) 689–705, doi:10.1109/TMC.2007.1017.
[5] J.R. Kwapisz, G.M.W.S.A. Moor, Activity recognition using cell phone accelerometers, in: Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data, 2010.
[6] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, F. Zhao, A reliable and accurate indoor localization method using phone inertial sensors, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, in: UbiComp '12, ACM, New York, NY, USA, 2012, pp. 421–430, doi:10.1145/2370216.2370280.
[7] Y. Luo, Y. Chen, O. Hoeber, Wi-fi-based indoor positioning using human-centric collaborative feedback, in: Communications (ICC), 2011 IEEE International Conference on, 2011, doi:10.1109/icc.2011.5963278.
[8] J. Machaj, P. Brida, R. Piche, Rank based fingerprinting algorithm for indoor positioning, in: Indoor Positioning and Indoor Navigation (IPIN), 2011, doi:10.1109/IPIN.2011.6071929.
[9] A. Martin, C. Ionut, R.C. Romit, SurroundSense: Mobile phone localization via ambience fingerprinting, in: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom), 2009, pp. 261–272.
[10] A. Offstad, E. Nicholas, R. Szcodronski, R.R. Choudhury, AAMPL: accelerometer augmented mobile phone localization, in: Proceedings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less environments, ACM, New York, NY, USA, 2008, pp. 13–18. http://doi.acm.org.qe2a-proxy.mun.ca/10.1145/1410012.1410016.
[11] A. Rai, K.K. Chintalapudi, V.N. Padmanabhan, R. Sen, Zee: zero-effort crowdsourcing for indoor localization, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, in: Mobicom '12, ACM, New York, NY, USA, 2012, pp. 293–304, doi:10.1145/2348543.2348580.
[12] N. Ravi, N. Dandekar, P. Mysore, M.L. Littman, Activity recognition from accelerometer data. In: Proceedings of the 17th conference on Innovative applications of artificial intelligence (2005).
[13] S. Thrun, Probabilistic robotics, Commun. ACM 45 (3) (2002) 52–57, doi:10.1145/504729.504754.
[14] W. Waqar, Y. Chen, A. Vardy, Exploiting smartphone sensors for indoor positioning: A survey., in: Proceedings of the Newfoundland Conference on Electrical and Computer Engineering, 2011.
[15] W. Waqar, A. Vardy, Y. Chen, Motion modelling using Smartphones for indoor mobilephone positioning, in: Proceedings of the Newfoundland Conference on Electrical and Computer Engineering, 2011.
[16] M. Youssef, A. Agrawala, The Horus WLAN location determination system, in: Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, 2005, pp. 205–218.
[17] N. Zhao, Full-Featured Pedometer design realized with 3-axis digital accelerometer, 2011, (http://www.analog.com/library/analogdialogue/archives/44-06/pedometer.html). (accessed Oct. 2012).