Indoor Positioning using BLE iBeacon, Smartphone Sensors and Distance-based Position Correction Algorithm

Anh Vu-Tuan Trinh and Thai-Mai Thi Dinh

Trinh Vu Tuan Anh is with University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam (e-mail: tuananhtv97@gmail.com).
Dinh Thi Thai Mai is with University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam (corresponding author to provide e-mail: dttmai@vnu.edu.vn).

ABSTRACT

In this paper, we propose a Bluetooth Low Energy (BLE) iBeacon based localization system, in which we combine two popular positioning methods: Pedestrian Dead Reckoning (PDR) and fingerprinting. As we build the system as an application running on an iPhone, we choose Kalman filter as the fusion algorithm to avoid complex computation. In fingerprinting, a multi-direction-database approach is applied. Finally, in order to reduce the cumulative error of PDR due to smartphone sensors, we propose an algorithm that we name "Distance-based position correction". The aim of this algorithm is to occasionally correct the tracked position by using the iBeacon nearest to the user. In real experiments, our system can run smoothly on an iPhone, with the average positioning error of only 0.63 m.

Keywords:
Bluetooth Low Energy; Fingerprinting; iBeacon; Indoor positioning; iOS; Kalman filter; Pedestrian Dead Reckoning; Position fusion; Smartphone sensors

1. INTRODUCTION

Indoor positioning is the process of obtaining a device or a user's location in an indoor setting or environment [1]. In recent years, with the rapid development of Internet of Things applications, indoor positioning has been widely studied.

Researchers around the world have applied a number of technologies in their solutions for indoor localization. These include Wi-Fi, Bluetooth Low Energy (BLE), Radio Frequency Identification Device, or Ultra Wideband [1,2]. Out of these techniques, BLE seems to be a better solution, especially with the introduction of BLE iBeacon by Apple Inc. in 2013. iBeacon is a small, wireless device that can send its advertisements to compatible smartphones in its proximity via BLE [3]. A great number of recent research have focused on the use of beacons, since they are simpler to deploy, more energy efficient and low-cost compared to other technologies. Also, as most of the smartphones on the market now support BLE, an iBeacon based indoor positioning system can be built and utilized as a localization app running on smartphones.

Taking algorithms into consideration, the most popular method in iBeacon based indoor positioning is based on Received Signal Strength (RSS). This method can be divided into two main approaches: trilateration and fingerprinting. Trilateration requires the computation of distances between the user and at least 3 beacons, by applying RSS of those beacons in a log-distance path loss model. Meanwhile, fingerprinting requires building an offline RSS map and database of the interested indoor area [1,2]. We then rely on this database to predict the user's position in the online phase. The main problem of RSS based methods is the instability of the beacons' RSS due to noises, multipath fading, non-light-of-sight (NLOS), and other factors caused by the indoor environment [1,2].

Another popular algorithm is Pedestrian Dead Reckoning (PDR), which is based on the data from sensors, such as accelerators and magnetometers, embedded in smartphones. The sensors can provide information about the detection of the user's new steps, the user's step length and the moving direction. The current position can then be computed using the information. Knowing the user's initial position, PDR provides quite high positioning accuracy. However, in long tracking path, the smartphone's sensors can drift overtime and lead to high cumulative error [4,5].

In order to achieve a more accurate indoor positioning system, recent studies tend to fuse BLE beacon's RSS based methods with PDR. One of the first research that combines PDR and iBeacon is the work of Chen et al. [5]. In this work, they applied a particle filter as the fusion algorithm, with each particle representing a position. In the prediction phase, the particles' positions are updated using PDR. Then, in the update phase, the authors use an iBeacon based calibration process, which only starts when the user's device moves into the 4-meter-range of an iBeacon. When the process starts, the iBeacon – user distance is computed using the iBeacon's RSS and the log-distance path loss model. This distance is then used to compute the weight of each particle, before the final user's position is estimated based on the particles and their weight. Hence, the authors' aim when using iBeacon measurements is to reduce the drift of PDR. Another work that also apply particle filter based fusion can be found in [6], where the authors combine trilateration and PDR.

Instead of particle filter, other studies [7-9] use Kalman filter or extender Kalman filter. One of them is [7]. In this research, the state of the Kalman filter is a 2-dimensional vector representing the coordinates of the user's position. In the filter's time update, PDR estimates the current position. Then, in the measurement update, that PDR based position is corrected using trilateration based position. Similar work can be found in [8], in which the difference is that the authors choose fingerprinting instead of trilateration.

In addition, a number of authors [10-12] fuse PDR, iBeacon and Wi-Fi fingerprinting. In the work of Zou et al. [11] – where the authors use particle filter based fusion, iBeacon measurements are only used to compute the particles' weight when the user is in poor Wi-Fi coverage area. Otherwise, if the user is in good Wi-Fi coverage area, the Wi-Fi based positions are used to compute the weight instead.

Hence, there has been a lot of work that chose iBeacon – PDR fusion as the main approach for indoor positioning. Most of them resulted in quite low positioning errors. However, the algorithms in those work require complex and heavy computation. This is not suitable especially if we want to implement the system as an app running on a smartphone, as the app's response time can be delayed due to those complex algorithms. Therefore, the main aim of this paper is to design a fusion based indoor positioning system that not only provide fast, accurate real-time positioning services on smartphones, but also can overcome the ever-present problems of iBeacon and PDR based techniques. To avoid heavy computation, we use a Kalman filter instead of a particle filter, as the fusion algorithm to combine fingerprinting and PDR. In fingerprinting, we build a multi-direction-database for its online phase, in order to reduce the effect of NLOS. Also, we proposed an effective and lightweight algorithm that we call "Distance-based position correction" to occasionally fix the user's position based on the beacon nearest to the user. This helps reducing the cumulative error due to PDR. In experiments, the proposed system runs smoothly as an app on an iPhone. It results in a low average positioning error of only 0.63 m. The details for each part of the proposed system will be introduced in subsequent sections. The rest of the paper is structured as follows: Section II presents the overview of the proposed system. Then, section III describes the system in details. Finally, we show the experimental results in section IV; section V concludes the paper.

2. PROPOSED SYSTEM MODEL

2.1 Proposed system overview

The block diagram for the proposed indoor positioning system is shown in Figure 1. The RSS values from the beacons are first filtered by a moving average filter, before being used

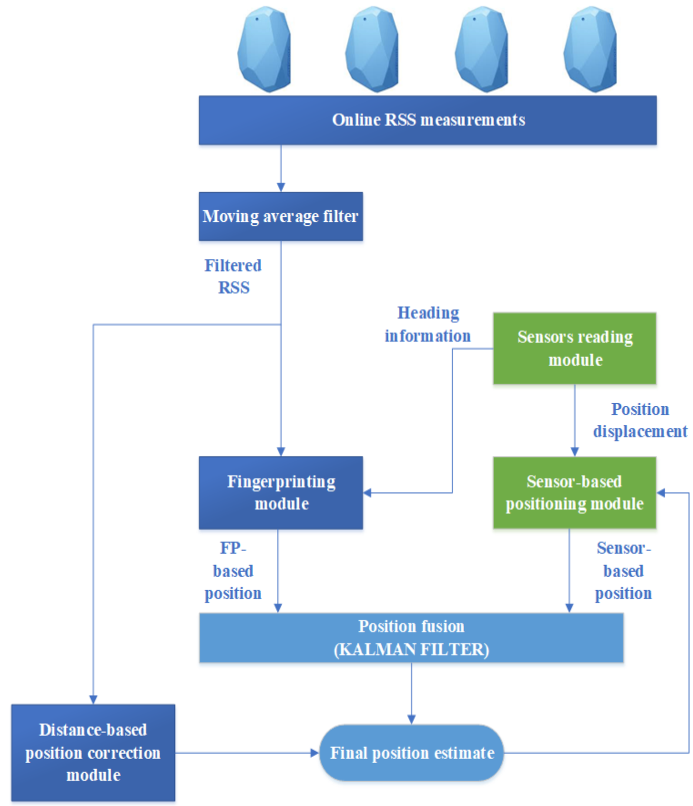


Figure 1: Proposed indoor positioning system

in fingerprinting module and distance-based correction module. The sensor reading module, which is responsible for reading data from the sensors embedded in the user's smartphone, computes the position displacement. This displacement includes step length and heading direction of the user. The sensor-based positioning module then uses that information to estimate the current position. At the same time, having the heading direction from the sensors' data, the fingerprinting module chooses the database corresponding to that heading. Then, based on the chosen database, this module estimate fingerprinting-based position of the user, which is then fused with the sensors-based position by the Kalman filter. Finally, the output of this fusion is occasionally fixed by the Distance-based position correction module, using the filtered RSS from the beacon nearest to the user's smartphone. The correction module is only triggered when the user stands still and near a beacon for an amount of time. The corrected position is the final estimation the user's position.

2.2 iBeacon and iOS development frameworks

In order to build the system as an iOS app, we use two development frameworks provided by Apple Inc., which are called CoreLocation and CoreMotion. CoreLocation allows us to read data from the beacons [13-15]. This data can be identification information of a specific beacon and its RSS value. With CoreMotion, we are able to get access to data from an iPhone's embedded sensors [16]. From that, as the user moves, the user's acceleration and the device's heading direction can be achieved to compute the positon displacement.

3. PROPOSED SYSTEM EXPLANATION

3.1 Moving average filter

The beacon's RSS value is heavily influenced by the indoor environment, thus filtering the RSS of each beacon is necessary. There are a number of methods to filter a beacon's RSS, such as average filter, median filter and Kalman filter [7]. In our work, we use a simple moving average filter to avoid heavy computation. By using a moving window of n RSS values from a beacon, the filtered RSS value of that beacon is calculated as below. This filter is applied for RSS values of all the beacons.

$$RSS_{filtered} = \frac{\sum_{i=0}^{n} RSS_i}{n} \quad (1)$$

3.2 PDR Module

3.2.1 Sensors reading module

This module is responsible for providing the fingerprinting module with the smartphone's heading direction information, and providing the sensor-based positioning module with both that heading direction and the user's acceleration. As mentioned above, we use CoreMotion from the iOS development frameworks to read the sensors' data. As this framework provides data that is already filtered, extra filtering methods are not necessary, hence again we can avoid extra computation.

3.2.2 Sensor-based positioning module

Let $I_t = [x_t, y_t]^T$ be be the 2-dimensional position of the user at time step t. In sensor-based positioning module, I_t can be computed from the previous position I_{t-1} by adding the position displacement u_t.

$$\tilde{I}_t = \hat{I}_{t-1} + u_t \quad (2)$$

The position displacement has the form as follows:

$$u_t = \begin{bmatrix} \Delta_t cos\theta_t \\ \Delta_t sin\theta_t \end{bmatrix} \quad (3)$$

where Δ_t is the user's step length and θ_t is the heading direction at time step t. Thus, in order to detect and calculate the user's position displacement, we need the following information:

- Step detection: detect whether the user makes a move.
- Step length Δ_t.
- Heading direction θ_t.

3.2.2.1 Step detection

CoreMotion framework provides acceleration data according to a three-axis accelerometer [16]. This accelerometer delivers acceleration measurements in each of the three axes as shown in Figure 2. In the scenario of our study, the user holds the smartphone on his/her hands so that the back of the phone is opposite and parallel to the ground. Therefore, only the vertical acceleration a_y, i.e., the acceleration measurement in the y-axis, is sufficient to detect the user's step. A double-threshold is then applied for the vertical acceleration as follows:

$$Step\ detected\ when\ a_{threshold\_1} \leq a_y \leq a_{threshold\_2}.$$

Figure 2: Three-axis accelerometer of a smartphone.

3.2.2.2 Step length

There are a number of methods for calculating a person's step length, including computing based on the height of the person [17], or updating the step length during the walk using walking speed, walking frequency and acceleration [18]. However, for simplicity, we fix the user's step length to a constant value of around 0.6 m.

3.2.2.3 Heading direction

The embedded magnetometer provides information about the phone's magnetic heading, which is the angle of the phone's heading direction relative to the magnetic North. From this, by adding an amount of offset to that value, we compute the heading direction of the smartphone/user in our own coordinates system. In our coordinates system, the range of the heading value can be seen as in Figure 3.

3.3 Fingerprinting

Fingerprinting is a prior scene analysis based technique which includes 2 phases: offline phase and online phase [1,2,19].

3.3.1 Offline phase

In fingerprinting's offline phase in our study, we made a grid map for the area where the indoor positioning system to be used, as demonstrated in Figure 4. The area of 1 grid is 0.6 m x 0.6 m. Then, the RSS values from all the beacons, which are noted by yellow and pink dots, are collected at intersection points of the map. At each point, data is collected at 4 directions of the coordinates systems: 0 degree, 90 degree, 180 degree, and 270 degree. Therefore, there are 4 offline databases in total, each one corresponds to each of those 4 directions. This will help reduce the effect of NLOS to beacons' RSS values, as the user's body can block the signals from beacons.

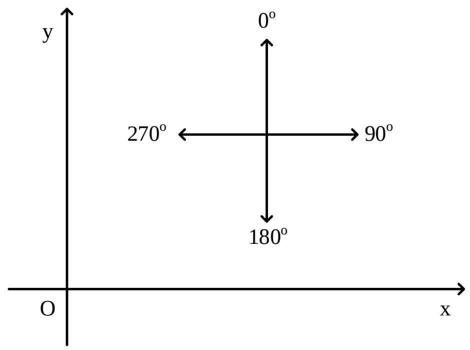


Figure 3: Heading direction in Oxy coordinates system

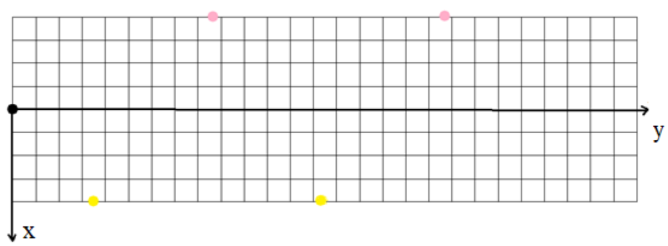


Figure 4: Fingerprinting grid map

In each direction, the RSS vector for an intersection point with position (x, y) has the form as follows: (x, y): $[RSS_1, RSS_2, \dots, RSS_n]$, in which n is the number of beacons. Thus, an intersection point of the grid map will have 4 RSS vectors corresponding to 4 databases. For example, the data for a point with coordinates of (8, 9) is shown below.

$(8, 9)$: $[-66, -85, -76, -79]$ /0 degree
$(8, 9)$: $[-79, -84, -76, -81]$ /90 degree
$(8, 9)$: $[-72, -85, -79, -77]$ /180 degree
$(8, 9)$: $[-71, -79, -74, -69]$ /270 degree

3.3.2 Online phase

In the online phase, based on the heading information from the sensors reading module, the fingerprinting module will choose the database corresponding to that heading direction. Then, we use k-Nearest Neighbor (*kNN*) – a machine learning algorithm that has been applied widely in indoor positioning [19]. The idea of *kNN* is to compute the distances between the online RSS vector observed by the user and every offline RSS vectors in the chosen database. Then, *kNN* returns *k* positions that have the corresponding offline RSS vectors with smallest distances [20]. In our study, *k* equals to 1. Assuming that the online data vector of the user's position is $V_a = [RSS_1, RSS_2, \dots, RSS_n]$ (n is the number of beacons), and the i^{th} position's offline data vector is $V_i = [RSS_{i1}, RSS_{i2}, \dots, RSS_{in}]$. The Euclidean distance between 2 vectors is computed as in Equation (4).

$$d(V_a, V_i) = \sqrt{\sum_{j=1}^{n} |RSS_j - RSS_{ij}|^2} \quad (4)$$

After that, the position corresponding to the offline vector that has the smallest value of d is chosen. In the system's diagram shown in Figure 1, this position is called fingerprinting-based position.

3.4 Kalman filter based position fusion

The sensor-based position and the fingerprinting-based position are fused using a Kalman filter.

3.4.1 Dynamic and measurement models

In the dynamic model of the Kalman filter in the proposed system, let $I_t = [x_t, y_t]^T$ be the user's position at time step *t*, we have:

$$I_t = I_{t-1} + u_t + w_t \quad (5)$$

where I_{t-1} is the user's position at time step *t – 1*, $w_t \sim N(0, Q)$ is the process noise, u_t is the position displacement provided by the sensors reading module ($u_t = \begin{bmatrix} \Delta_t cos\theta_t \\ \Delta_t sin\theta_t \end{bmatrix}$).

In the Kalman filter's measurement update, let $z_t = [x_t^{FP}, y_t^{FP}]$ be the fingerprinting-based position at time step *t*. We have:

$$z_t = I_t + v_t \quad (6)$$

where $v_t \sim N(0, R)$ is the measurement noise.

3.4.2 Time update and measurement update

There are 2 stages in the Kalman filter: time update (prediction) stage and measurement update (correction) stage, which can be seen in Table 1.

Table 1: Two-stage process of Kalman filter

Time update	Measurement update
$\tilde{I}_t = \hat{I}_{t-1} + \begin{bmatrix} \Delta_t cos\theta_t \\ \Delta_t sin\theta_t \end{bmatrix}$ $\tilde{P}_t = P_{t-1} + Q$	$K_t = \tilde{P}_t(\tilde{P}_t + R)^{-1}$ $\hat{I}_t = \tilde{I}_t + K_t(z_t - \tilde{I}_t)$ $P_t = (1 - K_t)\tilde{P}_t$

In the time update, the prior estimate of the user's position $\tilde{I}_t$, which is also the PDR-based position, is computed by adding the position displacement to the previous position $\hat{I}_{t-1}$. Then, the prior covariance $\tilde{P}_t$ is calculated. In the measurement update, after computing the Kalman gain K_t, the posterior user's position $\hat{I}_t$ is estimated using the Kalman gain, the PDR-based position $\tilde{I}_t$ and the measurement z_t, which is the fingerprinting-based position. Finally, the posterior covariance P_t is computed before starting the next loop.

3.5 Distance-based position correction

The aim of this proposed algorithm is to occasionally correct the user's position and prevent the high error of PDR due to drifting. To ensure that the RSS values of the beacons are stable, the module is only triggered when the user stands still for *b* seconds (in our experiments (*b* is set to 6 seconds). The algorithm is based on the distance between the user and the beacon with the strongest RSS at that moment, which is most likely to be the beacon nearest to the user. This distance, denoted by d, is computed by using the popular log-distance path loss model.

$$d = 10^{\frac{RSS_{1\,m} - RSS_d}{10n}} \tag{7}$$

in which $RSS_{1\,m}$ is the RSS of the beacon with the strongest RSS at a reference distance of 1 m, RSS_d is the RSS of that beacon at distance d, and n is the path loss exponent, which is varied in different indoor areas.

According to our experiments, the RSS of the beacon is most reliable if the user stands within the range of 3 m around the beacon. Hence, the algorithm will only continue if $d \leq 3\,m$. In the next step, we compute the Euclidean distance between the user's position (estimated by the Kalman filter) and the beacon. Assuming that the current position is denoted by $P(x_p, y_p)$ and the beacon's position is denoted by $B(x_b, y_b)$. The distance d_p between P and B is computed by:

$$d_p = \sqrt{(x_p - x_b)^2 + (y_p - y_b)^2} \tag{8}$$

If $d_p > d$, the user's position predicted by the Kalman filter is too far from the nearest beacon. The correction module will then correct the user's position $P(x_p, y_p)$ to a new position $C(x_c, y_c)$. The distance between C and the nearest beacon is d. Figure 5 provides a more visualized understanding. In this figure, B is the beacon's position, P is the user's position (estimated by the Kalman filter), and C is the correct position. As the correction position C is the intersection of BP and the circle whose center is B, we find C using basic geometry. C is then the final estimation of the user's position. A summary for the proposed Distance-based position correction algorithm is shown as a flowchart in Figure 6.

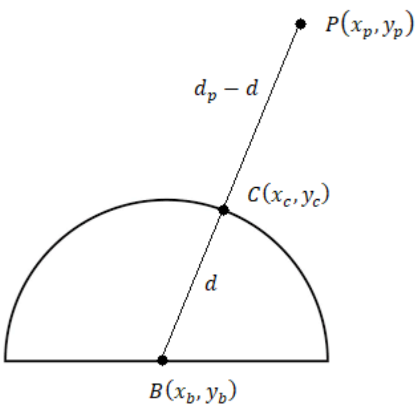


Figure 5: Visualized view of the user's position, the correct position and the beacon

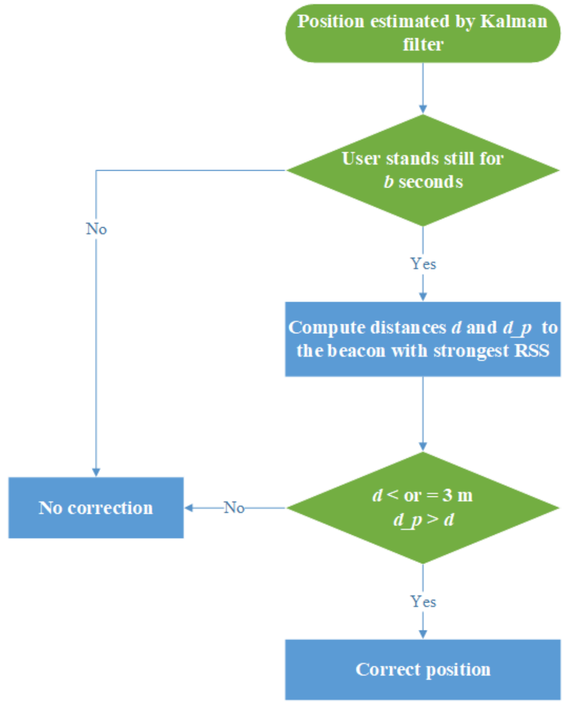


Figure 6: Flowchart of Distance-based position correction algorithm

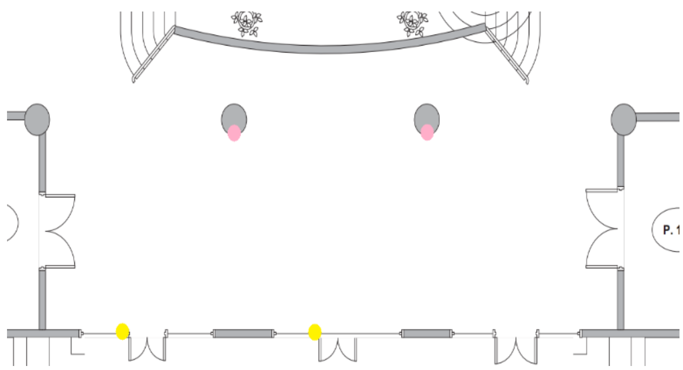


Figure 7: Experiment set-up

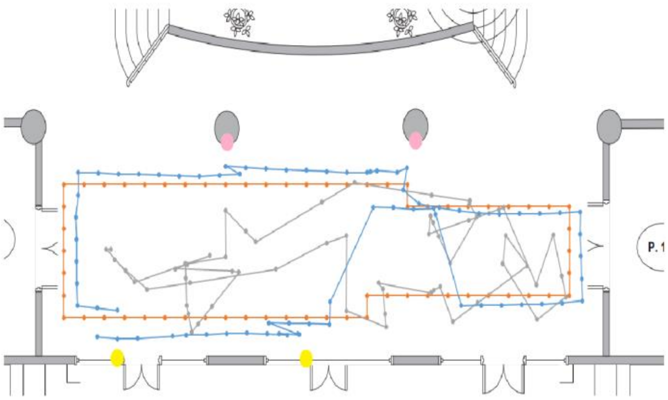
Table 2: Summary of devices' parameters

User Devide	iPhone 5C
Wireless Interface	BLE v4.2/ 2.4 GHz
Operating System	iOS 10.3.3
Beacons	4 Estimote iBeacons
Broadcasting range	50 m
Advertising Interval	100 ms
Broadcasting Power	0 dBm

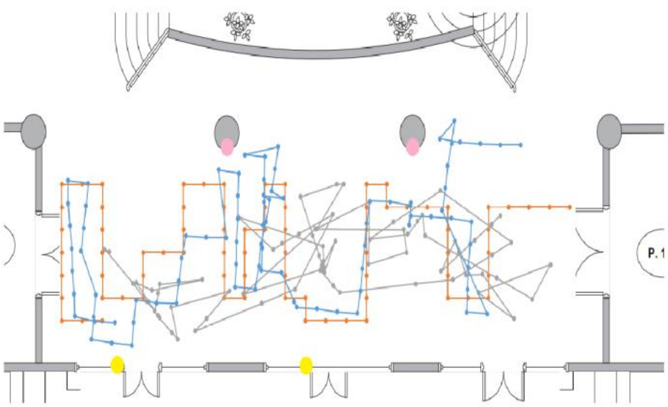
4. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed system, we build an indoor positioning app running on an iPhone 5C. The experiments are conducted on an indoor area of 16.2 m x 4.8 m. Table 2 summarizes the equipment related information used in the experiments. The beacons' positions in the area are shown in Figure 7, in which the pink and yellow dots indicate the beacons. The distance between 2 pink beacons and the distance between 2 yellow beacons are all 6 m. The fingerprinting grid map for this area is in Figure 4.

Then, as the user walks around the area, the app tracks and records the user's position. We did the experiment in 2 cases, one with the proposed system that has the Distance-based position correction module, and the other with the system that does not have it. The results collected from 2 different walking paths are shown in Figure 8.



(a)



(b)

Figure 8: Experimental results: (a) Simple walking path, (b) Complex walking path

In Figure 8, the orange line indicates the true path, the blue line is the tracked path with correction algorithm, and the grey line is the tracked path without it. In the case of a simple walking pattern (Figure 8(a)), without the correction algorithm, the system results in a very high error of up to 5.49 m, with an average error of 1.99 m. This is due to the drift of PDR and mostly due to the instability of beacons' signals. Although we have applied RSS filtering and fingerprinting databases for multiple directions, the fingerprinting-based position is still very unreliable. However, with the correction algorithm, the performance is significantly improved. The maximum error is down to 2.49 m, and the average error is only 0.63 m. In addition, the system also runs and responses well on the iPhone 5C.

With a more complex walking path (Figure 8(b)), the results are quite similar. Without the correction module, the maximum error is 5.03 m and the average error is 2.25 m. The performance is again improved with the proposed algorithm, with the maximum and average errors of 3.05 m and 0.90 m, respectively. A summary of our experimental results including maximum error, average error, variance and mean squared error is included in Table 3.

Table 3: Summary of experimental results

		Max. error (m)	Avg. error (m)	VAR	MSE
Simple path (Figure 8(a))	Without correction module	5.49	1.99	2.02	5.99
	With correction module	2.49	0.63	0.26	0.66
Complex path (Figure 8(b))	Without correction module	5.03	2.25	1.32	6.41
	With correction module	3.05	0.90	0.47	1.28

5. CONCLUSION

In this paper, we have introduced an iBeacon based indoor positioning system that fuses PDR and fingerprinting. In order to avoid complex and heavy computation, we use a Kalman filter as the fusion algorithm and make use of the data provided by the iOS development frameworks. In addition, we proposed a lightweight algorithm called Distance-based position correction, which has shown its high efficiency in the experiments. We also make a positioning app to test the system performance. The app runs well on an iPhone with a low average error of 0.63 m.

ACKNOWLEDGEMENT

This work was supported by a research grant from QG 20.22 Project of Vietnam National University Hanoi.

REFERENCES

1. F. Zafari, A. Gkelias and K. Leung, "A survey of indoor localization systems and technologies". Available: http://arxiv.org/abs/1709.01015v2.
2. M. A. Al-Ammar *et al*., "Comparative survey of indoor positioning technologies, techniques, and algorithms," *2014 International Conference on Cyberworlds*, Santander, 2014, pp. 245-252.
3. Silicon Labs, "Developing beacons with Bluetooth Low Energy (BLE) technology". Available: https://www.silabs.com/products/wireless/bluetooth/developingbeacons-with-bluetooth-low-energy-ble-technology.
4. Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y.C. Soh, and L. Xie, "Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization," *Sensors*, 15, 715-732, Jan. 2015.
5. Z. Chen, Q. Zhu, H. Jiang, and Y. C. Soh, "Indoor localization using smartphone sensors and iBeacons," *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, Auckland, 2015, pp. 1723-1728.
6. V. Chandel, N. Ahmed, S. Arora, and A. Ghose, "InLoc: An end-to-end robust indoor localization and routing solution using mobile phones and BLE beacons," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Alcala de Henares, 2016, pp. 1-8.
7. J. Robesaat, P. Zhang, M. Abdelaal, and O. Theel, "An improved BLE indoor localization with Kalman-based fusion: An experimental study," *Sensors*, 17(5), Apr. 2017.
8. S. Lee, B. Cho, B. Koo, S. Ryu, J. Choi, and S.Kim, "Kalman filter-based indoor position tracking with self-calibration for RSS variation mitigation," *International Journal of Distributed Sensor Networks - Special issue on Location-Related Challenges and Strategies in Wireless Sensor Networks*, vol. 2015, Jan. 2015.
9. Z. Chen, Q. Zhu, and Y. C. Soh, "Smartphone inertial sensor-based indoor localization and tracking with iBeacon corrections," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 4, pp. 1540-1549, Aug. 2016.
10. K. Sung, D.K. Lee, and H. Kim, "Indoor pedestrian localization using iBeacon and improved Kalman filter," *Sensors*, 18(6), May 2018.
11. H. Zou, Z. Chen, H. Jiang, L. Xie, and C. Spanos, "Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon," *2017 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, Kauai, HI, 2017, pp. 1-4.
12. J. Chen, Y. Zhang, and W. Xue, "Unsupervised indoor localization based on smartphone sensors, iBeacon and Wi-Fi," *Sensors*, 18(5), May 2018.
13. Apple, "Getting started with iBeacon". Available: https://developer.apple.com/ ibeacon/Getting-Started-with-iBeacon.pdf.
14. F. Zafari and I. Papapanagiotou, "Enhancing iBeacon based microLocation with particle filtering," *2015 IEEE Global Communication Conference (GLOBECOM)*, San Diego, CA, 2015, pp. 1-7.
15. CoreLocation. Retrieved from https://developer.apple.com/documentation/corelocation.
16. CoreMotion. Retrieved from https://developer.apple.com/documentation/coremotion.
17. P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech House, 2013.
18. J. Kim, H. Jang, D. Hwang, and C. Park, "A Step, Stride and Heading Determination for the Pedestrian Navigation System," *Positioning*, vol. 1, no. 8, 2004.
19. S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor fingerprint positioning based on Wi-Fi: an overview," *International Journal of Geo-Information*, vol. 6, pp. 135, Apr. 2017.
20. L. Zhang, X. Liu, J. Song, C. Gurrin, and Z. Zhu, "A comprehensive study of Bluetooth fingerprinting-based algorithms for localization," *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, Barcelona, 2013, pp. 300-305.

Anh Vu-Tuan Trinh is a bachelor's degree senior student in Electronics and Communications Engineering at University of Engineering and Technology (Hanoi, Vietnam). His current main research direction is indoor localization using BLE iBeacons.

Thai-Mai Thi Dinh is a Lecturer of Faculty of Electronics and Telecommunications, VNU University of Engineering and Technology, Hanoi, Vietnam. She graduated from Post and Telecommunication Institute of Technology, Vietnam in 2006. Then, she received the Master and PhD degrees from Paris Sud 11, France in 2008 and VNU University of Engineering and Technology, Hanoi, Vietnam in 2016, respectively. Her research interests focus on 5G Mobile Networks, Wireless Communications and Indoor Positioning System as well.