

Article

Indoor Positioning Based on Bluetooth Low-Energy Beacons Adopting Graph Optimization

Zheng Zuo ¹, Liang Liu ^{2,*}, Lei Zhang ² and Yong Fang ²

¹ College of Electronics and Information Engineering, Sichuan University, Chengdu 610064, China; b110434@stu.scu.edu.cn

² College of Cybersecurity, Sichuan University, Chengdu 610064, China; zhanglei2018@scu.edu.cn (L.Z.); yfang@scu.edu.cn (Y.F.)

* Correspondence: liangzhai118@scu.edu.cn

Received: 6 October 2018; Accepted: 30 October 2018; Published: 2 November 2018

Abstract: Bluetooth Low-Energy (BLE) beacons-based indoor positioning is a promising method for indoor positioning, especially in applications of position-based services (PbS). It has low deployment cost and it is suitable for a wide range of mobile devices. Existing BLE beacon-based positioning methods can be categorized as range-based methods and fingerprinting-based methods. For range-based methods, the positions of the beacons should be known before positioning. For fingerprinting-based methods, a pre-requisite is the reference fingerprinting map (RFM). Many existing methods focus on how to perform the positioning assuming the beacon positions or RFM are known. However, in practical applications, determining the beacon positions or RFM in the indoor environment is normally a difficult task. This paper proposed an efficient and graph optimization-based way for estimating the beacon positions and the RFM, which combines the range-based method and the fingerprinting-based method. The method exists without need for any dedicated surveying instruments. A user equipped with a BLE-enabled mobile device walks in the region collecting inertial readings and BLE received signal strength indication (RSSI) readings. The inertial measurements are processed through the pedestrian dead reckoning (PDR) method to generate the constraints at adjacent poses. In addition, the BLE fingerprints are adopted to generate constraints between poses (with similar fingerprints) and the RSSIs are adopted to generate distance constraints between the poses and the beacon positions (according to a pre-defined path-loss model). The constraints are then adopted to form a cost function with a least square structure. By minimizing the cost function, the optimal user poses at different times and the beacon positions are estimated. In addition, the RFM can be generated through the pose estimations. Experiments are carried out, which validates that the proposed method for estimating the pre-requisites (including beacon positions and the RFM). These estimated pre-requisites are of sufficient quality for both range-based and fingerprinting-based positioning.

Keywords: BLE-based indoor positioning; fingerprinting; graph optimization

1. Introduction

Location has become increasingly important for many position-based services (PbS). However, although the global navigation satellite system (GNSS) can satisfy the needs for most outdoor situations, indoor positioning still remains a challenge. The abundant sensors embedded in today's mobile devices have greatly enhanced their ability to sense the indoor environment, thus providing possibilities for indoor positioning. Many indoor positioning methods rely on pre-installed infrastructures and can provide reasonable accuracy, such as UWB ranging anchor-based [1,2], Wi-Fi access points (APs)-based [3,4], Ultrasound [5,6]-based and so on. Among these methods, Bluetooth low-energy (BLE) beacons have great potential due to their advantages:

- The positioning process is independent of extra hardware. After the BLE beacons have been installed, the only equipment needed is a BLE-enabled smart phone, while foot-mounted or waist-mounted inertial pedestrian positioning [7,8] needs special inertial sensors.
- It is suitable for positioning adopting a wide range of mobile devices, including both Android devices and IOS devices. While Wi-Fi-based positioning is only suitable for Android devices, because the API of IOS does not provide the Wi-Fi scanning results.
- The deployment cost for BLE beacons is low. Once the beacons are deployed, it can continuously work for a long duration (e.g., half a year) using their inner batteries, because the beacon nodes have low power consumption.

The BLE beacon-based positioning methods consist of mainly two types: range-based and fingerprinting-based. Range-based methods adopt a pre-defined radio frequency (RF) path-loss model to estimate the distance between the receivers (users) and the beacons. Assuming at least three received signal strength indication (RSSI) measurements are available, the user's position can be solved by trilateration according to the distances estimated accordingly. The authors in [9] proposed three different methods for trilateration.

- The least square estimation (LSE) method [9]. By minimize the square sums of the distance errors, an optimal position can be found.
- The three-border method. By establishing the equations which represent the distance between the user and the beacons, the user's position can be found by solving the equations.
- The centroid method. A polygon is firstly defined according to the vertexes defined by the intersecting points from the distance arcs. Then the centroid is regarded as the user's position.

In [10], a more sophisticated adaptive propagation model is proposed, which models the parameters of the path-loss model with a particle filter. In [11], a combination of channel-separate polynomial regression model is adopted for estimating the distances between the receiver and the beacons, and the results show better accuracy than a simple propagation model.

The fingerprinting-based method consists of two phases: the offline phase and the online phase. In the offline phase, the fingerprints (a vector of RSSIs from different beacon nodes collected at a coordinate point) at different places are collected to create a reference fingerprint map (RFM). In the online phase, a fingerprint collected at an unknown place is compared with the fingerprints in the RFM, to solve for the position of the user. The online phase is rather simple and a commonly seen method for the phase is the k-nearest neighbor (kNN) method. However, as fingerprints needs to be collected at different places at the offline phase, the workload is huge. Moreover, the positions where the fingerprints are collected needs to be known, which is challenging. Normally, the positions need to be measured through dedicated surveying methods. For example, in [11], the fingerprints are collected manually at many fixed positions, then these fingerprints are interpolated according to a Gaussian process regression model to generate an RFM.

As mentioned, the range-based BLE beacon positioning methods need to know the positions of the beacons to start with. Similarly, a pre-requisite for fingerprinting-based methods is the RFM. Existing methods mainly focus on the positioning methods and assumes the beacon positions and the RFM is known before positioning. However, in practical applications, determining the beacon positions or fingerprinting collection positions, especially for the indoor environment with a relatively large area, is an energy intensive and error prone task.

Unlike the mentioned methods, the proposed method in the paper provides an efficient way for estimating the pre-requisites for BLE-based positioning. The proposed method can estimate both the beacon positions and the RFM using graph-based optimization. The priority advantage of the proposed method is that it is free of any dedicated surveying process for the beacon positions and RFM. A user equipped with a BLE-enabled mobile device walks in the region. As shown in Figure 1, through the embedded sensors, the mobile device can collect inertial measurements and RSSI from BLE beacons. The method can fuse three types of information: the pedestrian dead reckoning (PDR)

information, the fingerprint information, and the range information. More specifically, the inertial measurements are processed through the PDR method to generate the constraints at adjacent poses. In addition, the BLE fingerprints are adopted to generate constraints between poses (with similar fingerprints). The RSSIs are adopted to generate distance constraints between the poses and the beacon positions (according to a pre-defined path-loss model). The constraints are then adopted to form a cost function with a least square structure. By minimizing the cost function, the optimal user poses at different times and the beacon positions are estimated. In addition, the RFM can be generated through the pose estimations.

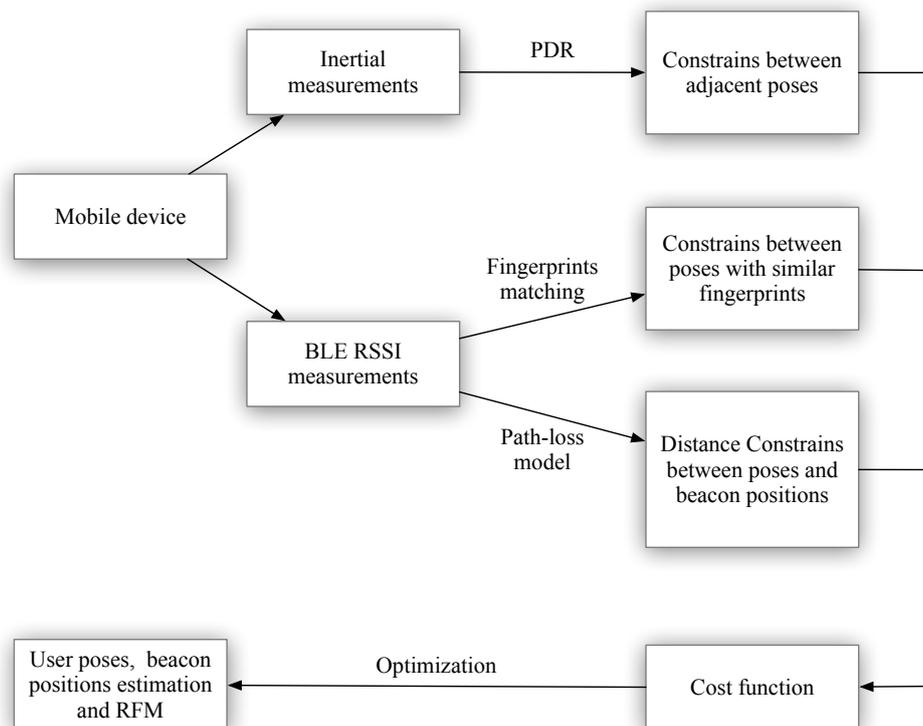


Figure 1. The overall data processing flow of the proposed method.

A few related works are given in Section 2 including the features of the BLE RSSI, PDR method and graph-based optimization. Section 3 is the method for forming the cost function and the optimization method. Experiments are carried out in Section 4 to validate the proposed method. The last section is the conclusion (Section 5).

2. Related Work

2.1. The BLE RSSI Features

BLE beacons operate in the same 2.4 GHz band as the Wi-Fi transceivers. Normally, in positioning, the beacons are in advertising mode, broadcasting short and flexible messages at flexible update rates [12,13]. Apple's iBeacons are one type of such BLE beacons (supported by Android devices too), which also broadcast messages using the BLE standard [14]. The messages contain the ID of the beacons, so that the receiver can distinguish difference beacons. Different from the Wi-Fi APs, which broadcast messages at 20 MHz wide band, the BLE has 40 channels with width 2 MHz. Only three channels are adopted for broadcasting and each broadcasting is repeated for on the three channels. Due to the advertising scheme of the BLE beacons, the RSSI from these beacons have these features:

- The RSSI can change very dramatically with even a very small spatial change. As each of the broadcasting channels are narrow banded, the BLE signals have fast fading nature. Also,

the indoor environment is very complex with many surfaces (e.g., walls, floors and so on), where the BLE signals can reflect. This can further cause multipath fades and add instability for the RSSI. In an experiment performed in [15], the noise and fading effect can even domain the RSSI with longer distance to the beacons.

- The RSSI can be reported for multiple times or not reported at all during a single scan. As the advertisement is repeated on three channels and if a scan is longer than a broadcasting interval, multiple reporting on the same beacon can be seen. Also, if the RSSI from all the three channels is below the environment noise due to fading effect, no reporting of the beacon is available for the scan. This can cause inconsistency for adjacent scans in time.

The mentioned features can cause huge noises in both RSSI (first feature) and beacon availability (second feature). To minimize the unfavorable effects, some smoothing processes should be added for the RSSI. In this paper, we adopt the median value of nearby RSSI (within a time window size of 1 s) as the current RSSI. This has been validated by an experiment in [13], which found out that 10 Hz beaconing and 1 s batch processing of median or mean values produced the best results.

2.2. Pedestrian Dead Reckoning

The PDR algorithm can update the pedestrian's positions adopting the inertial measurements. Normally, the position update rate relies on the step rate of the pedestrian, i.e., the positions are updated each time a new step is taken by the pedestrian (in Equation (1)). (x, y) are the positions, the subscripts are the step indexes, L is the stride length estimation and θ is the orientation estimation.

$$\begin{aligned} x_k &= x_{k-1} + L_k \cos(\theta_k) \\ y_k &= y_{k-1} + L_k \sin(\theta_k) \end{aligned} \quad (1)$$

The details of the PDR algorithm differ with the body part where the inertial measurement unit (IMU) is mounted. If the IMU is mounted on the foot, the zero-velocity update (ZVU) method is adopted. In this method, a filtering process (assuming zero velocity) is added to the normal inertial calculation when the foot of the pedestrian touches the ground (zero-velocity phase) [16,17]. If the IMU is waist-mounted or hand held, no obvious zero-velocity phase can be found. Only the orientation is estimated through inertial calculation, while the stride occurrences and stride length are estimated through heuristic models [18]. In our implementation, as the inertial sensors in the mobile device is hand held, the second processing method of PDR is adopted. However, as the PDR algorithm relies heavily on heuristic assumptions, the accuracy is not satisfactory. Moreover, the insufficient accuracy of the gyroscopes can deteriorate orientation estimation. Even with magnetometer readings, the orientation estimation cannot be improved much, due to the many magnetic disturbances in the indoor environment [19]. Here, the PDR algorithm can only provide a rough constraint between pose estimation at adjacent time (here the PDR constraints denotes the stride length and orientation difference estimation between steps, which will be explained in the next section). Other types of constraints are needed to achieve pose estimations with reasonable accuracies.

2.3. Graph-Based Optimization

Graph-based optimization is currently widely adopted in robotics for solving the simultaneously localization and mapping (SLAM) problem [20]. It aims to estimate the poses of the robotics while generating a map of the environments (e.g., landmark maps, occupancy grid maps, feature maps and so on). The SLAM procedures consist of front-end tracking and back-end optimization.

- In the front-end tracking procedure, sensors on the robots are adopted to find constraints between the poses at different times. For example, features in adjacent image frames are adopted to find position and attitude constraints between poses (visual odometry) [21]. These constraints can generate a sequence of initial poses at different times (trajectory). However, this trajectory has accumulative error. If other types of sensor measurements (other than odometry observations)

are available, more constraints can be formed, and the trajectory can be estimated more accurately using the back-end optimization.

- The back-end optimization procedure is generally establishing a cost function and then minimize it. The previously mentioned odometry observations and all other available observations are adopted to generate constraints between pose variables. Each constraint denotes a term in the cost function, which represents the errors between the sensor observation and the observation derived from pose variables. The cost function has a least square form which is in essence representing how the observations match the estimation. By finding the best match (minimizing the cost function), the optimal poses (trajectory) and maps can be found. The establishment of constraints and optimization process in our case is explained in detail in the next section.

As shown in Figure 2, in our implementation, the front-end tracking procedure corresponds to generate initial and less accurate poses through the results of PDR algorithm. The back-end optimization procedure corresponds to finding the optimal poses and beacon positions from the cost function using three types of constraints: PDR constraints, constraints through fingerprints matching and distance constraints derived from the path-loss model. Noting that the initial poses correspond to the poses estimated solely from the PDR algorithm, and is adopted as initial values for the back-end optimization procedure. The front-end PDR algorithm is already well studied in a wide range of publications and considered trivial. We focus on adopting the constraints to build the back-end optimization procedure.

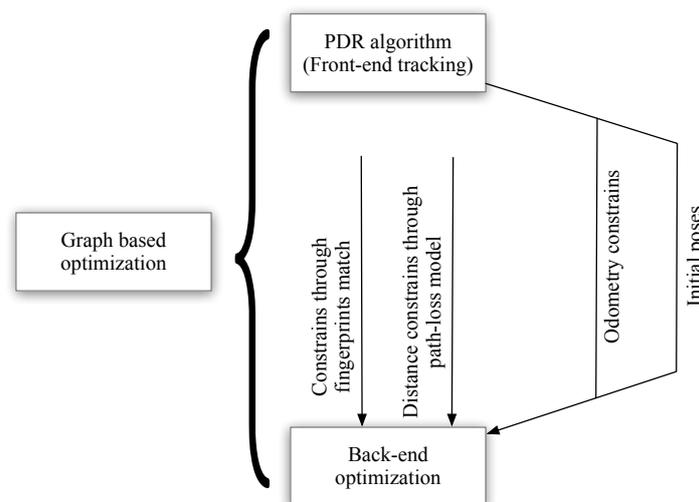


Figure 2. The graph SLAM procedures and its implementation in our method.

3. Method

The problem of user poses and beacon position estimation in our method is transformed to a least square optimization problem. In this section, a general least square optimization framework for user pose estimation is given at first. Then the framework is elaborated by adding least square terms according to the mentioned constraints to the cost function. After optimization using the Levenberg-Marquardt algorithm [22], the optimal user poses and beacon positions can be found.

3.1. Fundamentals of the Least Square Optimization Framework

The model for general pose estimation is formed at first. Assuming the noises are Gaussian, the maximum likelihood estimation of the poses is equivalent to optimize the formed cost function. Some fundamentals for the optimization process are given here.

3.1.1. Formation of Least Squares

In a general pedestrian positioning problem, the poses of the pedestrian and the observations satisfy:

$$\begin{aligned}\mathbf{q}_t &= f(\mathbf{q}_{t-1}, \mathbf{s}_t) + \mathbf{w}_t \\ \mathbf{o}_t &= g(\mathbf{q}_t, (\cdot)) + \mathbf{v}_t\end{aligned}\quad (2)$$

Here, the pose at time t is denoted by \mathbf{q}_t and it is a three-element vector $[x_t, y_t, \theta_t]$. The observation at time t is denoted by \mathbf{o}_t . \mathbf{s}_t is the from the PDR algorithm, which denotes the pose change between the $t-1^{th}$ pose and the t^{th} pose. Here it is a two-element vector $[L_t, \delta\theta_t]$ which is stride length estimation and heading change estimation respectively according to the PDR formation in Equation (1) (also denoted as odometry observation). \mathbf{w}_t and \mathbf{v}_t are Gaussian noise and is assumed to be independent at different times. The $f(\cdot)$ and $g(\cdot)$ are functions correspond to the pose propagation model and the observation model, respectively. In the propagation model, the pose at the next step \mathbf{q}_t is dependent on the previous pose \mathbf{q}_{t-1} and the pose change \mathbf{s}_t derived from the PDR algorithm. In the observation model, the current observation \mathbf{o}_t is dependent on the current pose \mathbf{q}_t and other variables (represented as (\cdot) in Equation (2)). Noting that in real implementation, (\cdot) can denotes another pose or other new variables need to be estimated. For generalization reasons, (\cdot) is adopted instead.

In the formation, as the noises are assumed to be Gaussian, the maximum likelihood estimation of the poses is equivalent to finding the optimal poses which minimize the cost function with a least square form in Equation (3). The detailed proof of the equivalence can be found in [20].

$$\mathbf{F}(\mathbf{q}) = \sum_t \mathbf{e}_{t,f} \mathbf{W}_t^{-1} \mathbf{e}_{t,f}^T + \sum_t \mathbf{e}_{t,g} \mathbf{V}_t^{-1} \mathbf{e}_{t,g}^T \quad (3)$$

where

$$\begin{aligned}\mathbf{e}_{t,f} &= \mathbf{q}_t - f(\mathbf{q}_{t-1}, \mathbf{s}_t) \\ \mathbf{e}_{t,g} &= \mathbf{o}_t - g(\mathbf{q}_t, (\cdot))\end{aligned}\quad (4)$$

The error $\mathbf{e}_{t,f}$ denotes the difference between the actual pose and the pose derived from the previous pose and the odometry observation from the PDR algorithm. The $\mathbf{e}_{t,g}$ denotes the actual observation and the derived observation. The matrix \mathbf{W} and \mathbf{V} denote the noise variance of \mathbf{w} and \mathbf{v} respectively. We present the details of noise variances in the respective error terms later. From Equation (3), we can see that it can represent how the observations (including odometry observations) is consistent with the pose estimations. By minimizing $\mathbf{F}(\mathbf{q})$, the optimal pose estimations can be found which can best satisfy the observations.

A factor graph can be adopted to better represent the cost function and relationships of the variables in it. In Figure 3, the circles denote the variables to be estimated (here the poses \mathbf{q}_t). A square denotes a constraint existing between the variables they connect. Each square can correspond to a square term in Equation (3). Here are two types of constraints (squares). \mathbf{s}_t denotes the PDR derived odometry constraints between adjacent pose variables, which correspond to the term $\mathbf{e}_{t,f} \mathbf{W}_t^{-1} \mathbf{e}_{t,f}^T$ in Equation (3). \mathbf{o}_t denotes the observation constraints between a pose variable and another variable (\cdot) , which correspond to the term $\mathbf{e}_{t,g} \mathbf{V}_t^{-1} \mathbf{e}_{t,g}^T$ in Equation (3). (\cdot) can be a pose variable or other type of variables need to be estimated. For BLE beacon-based graph in our method, (\cdot) denotes other pose variables and the beacon position variables, which will be explained later. The factor graph can give a good visual representation of the cost functions and the variables need to be estimated. Therefore, the term graph optimization is adopted to represent the minimization process of the cost function.

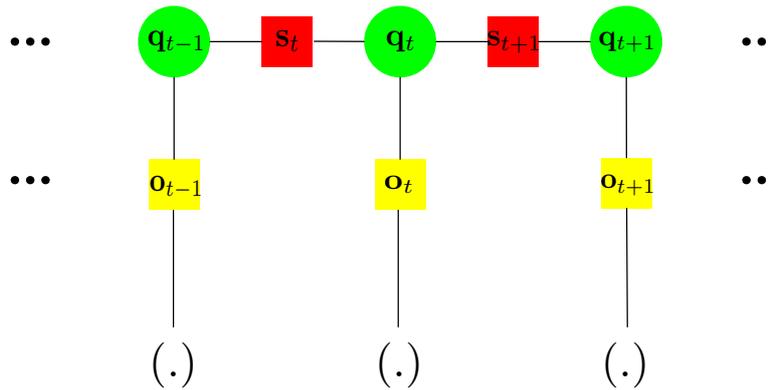


Figure 3. The factor graph representation of the cost function in Equation (3).

3.1.2. Levenberg-Marquardt Based Graph Optimization

The general square optimization (g2o) framework is proposed in [20] and is adopted in our method. Here we give a simple review of how the graph optimization problem is solved. We re-write the cost function in a more general form as:

$$F(\mathbf{x}) = \sum_{k \in \mathcal{C}} \underbrace{\mathbf{e}_k(\cdot) \Omega_k \mathbf{e}_k(\cdot)^T}_{F_k} \tag{5}$$

where

- \mathbf{x} is the set of many variables to be estimated, and it is $\mathbf{x} = (\mathbf{x}_1^T, \dots, \mathbf{x}_n^T)^T$;
- \mathcal{C} is the set for possible k , if we assume there are n variables in the factor graph, then it can be written as $k = (1, \dots, n)$;
- Ω_k denotes the information matrix for the k^{th} constraints (or square error terms). Here the information matrix corresponds to the inverse of the noise variance in Equation (3);
- $\mathbf{e}_k(\cdot)$ represents the errors from constraints. As the error is dependent on one or more variables, it can be written as:

$$\mathbf{e}_k(\cdot) = \mathbf{e}_k(\mathbf{x}_k) = \mathbf{e}_k(\mathbf{x}) \tag{6}$$

Then we adopt the Levenberg-Marquardt algorithm for finding the variable values which minimize the cost function iteratively. Assuming that in the previous iteration, the variable values can be written as $\check{\mathbf{x}}$. Here we consider the initial values of the variables is the value for the first iteration. Then we use the following steps for finding the variable values for the next iteration:

1. Expand the $\mathbf{e}_k(\mathbf{x})$ using first-order Taylor expansion at the variable value $\check{\mathbf{x}}$ (value for the previous iteration):

$$\mathbf{e}_k(\check{\mathbf{x}}_k + \Delta) = \mathbf{e}_k(\check{\mathbf{x}} + \Delta) \simeq \mathbf{e}_k + \mathbf{J}_k \Delta \tag{7}$$

where the \mathbf{J}_k is the Jacobian matrix for the error function $\mathbf{e}_k(\mathbf{x}_k)$ at the value $\check{\mathbf{x}}$. \mathbf{e}_k is the value for the error function $\mathbf{e}_k(\mathbf{x}_k)$ at $\check{\mathbf{x}}$. Δ is a minor increment for the variable \mathbf{x} . Then we substitute the single error function (Equation (7)) to F_k in Equation (5) and we can get:

$$\begin{aligned} F_k(\check{\mathbf{x}} + \Delta) &= \mathbf{e}_k(\check{\mathbf{x}} + \Delta)^T \Omega_k \mathbf{e}_k(\check{\mathbf{x}} + \Delta) \\ &\simeq (\mathbf{e}_k + \mathbf{J}_k \Delta)^T \Omega_k (\mathbf{e}_k + \mathbf{J}_k \Delta) \\ &= \underbrace{\mathbf{e}_k^T \Omega_k \mathbf{e}_k}_{c_k} + 2 \underbrace{\mathbf{e}_k^T \Omega_k \mathbf{J}_k}_{\mathbf{b}_k} \Delta + \Delta^T \underbrace{\mathbf{J}_k^T \Omega_k \mathbf{J}_k}_{\mathbf{H}_k} \Delta \\ &= c_k + 2\mathbf{b}_k \Delta + \Delta^T \mathbf{H}_k \Delta \end{aligned} \tag{8}$$

Then the overall error function (Equation (5)) with a minor increment can be written as:

$$\begin{aligned} \mathbf{F}(\check{\mathbf{x}} + \Delta) &= \sum_{k \in \mathcal{C}} \mathbf{F}_k(\check{\mathbf{x}} + \Delta) \\ &\simeq \sum_{k \in \mathcal{C}} (c_k + 2\mathbf{b}_k \Delta + \Delta^T \mathbf{H}_k \Delta) \\ &= c + 2\mathbf{b} \Delta + \Delta^T \mathbf{H} \Delta \end{aligned} \quad (9)$$

where

$$\begin{aligned} c &= \sum c_k \\ \mathbf{b} &= \sum \mathbf{b}_k \\ \mathbf{H} &= \sum \mathbf{H}_k \end{aligned} \quad (10)$$

- Calculate the derivation of Equation (9) over the minor increment Δ and makes it equal to zero (get the minimized error). We can get

$$\mathbf{H} \Delta = -\mathbf{b} \quad (11)$$

Here \mathbf{H} is the information matrix of the system. By solving the linear system in Equation (11), an increment can be acquired as Δ^* , then we update the variable value for the current iteration:

$$\check{\mathbf{x}} = \check{\mathbf{x}} + \Delta^* \quad (12)$$

The Gauss-Newton algorithm solves for the increment in Equation (11). For the Levenberg-Marquardt algorithm, a damping factor is added to Equation (11) to control the increment step size (more robust than the Gauss-Newton algorithm in general) and we can get

$$(\mathbf{H} + \lambda \mathbf{I}) \Delta = -\mathbf{b} \quad (13)$$

According to the algorithm, if $\mathbf{F}(\check{\mathbf{x}} + \Delta) < \mathbf{F}(\check{\mathbf{x}})$, we decrease the damping factor λ , otherwise we increase it to control the step size. The detailed strategy for choosing the suitable λ is described in [22]. Here we just give a simple principle.

- Continue step 1 and step 2 until the increment norm $\|\Delta\|$ is less than a pre-defined threshold. Then the current iteration of variable value $\check{\mathbf{x}}$ is considered optimal for the variables.

The dimension of \mathbf{H} grows with the dimension of the variables \mathbf{x} . However, the sparsity nature of the matrix \mathbf{H} can be adopted to efficiently solve the linear system in Equation (13).

3.2. BLE Beacon-Based Graph Optimization

The graph optimization method for a general least square cost function is described in the previous subsection. Here we focus on forming the cost function for BLE beacon implementation using the constraints from PDR, beacon-based range constraints and fingerprint-based constraints.

3.2.1. Cost function for BLE Beacon Implementation

Again, a factor graph is adopted to represent the cost function in Figure 4. Noting that the variable (.) in Figure 3 is specified here. There are three types of constraints here.

- The PDR constraints. Similar to Figure 3, the PDR constraints are between adjacent poses in time.
- The beacon position constraints. New variables representing beacon positions are added to the factor graph in Figure 4. For example, the distance $d_{k,t-1}$ between the pose variable \mathbf{q}_{t-1} and the k^{th} beacon position variable \mathbf{p}_k is derived from the RSSI according to the path-loss model.
- The fingerprints matching constraints. These types of constraints are between two pose variables whose collected fingerprints are with vicinity. For example, Figure 4 shows that the fingerprint collected at \mathbf{q}_1 and \mathbf{q}_t are with vicinity.

Based on the factor graph, we elaborate the square error terms of the cost function based on the different types of constraints.

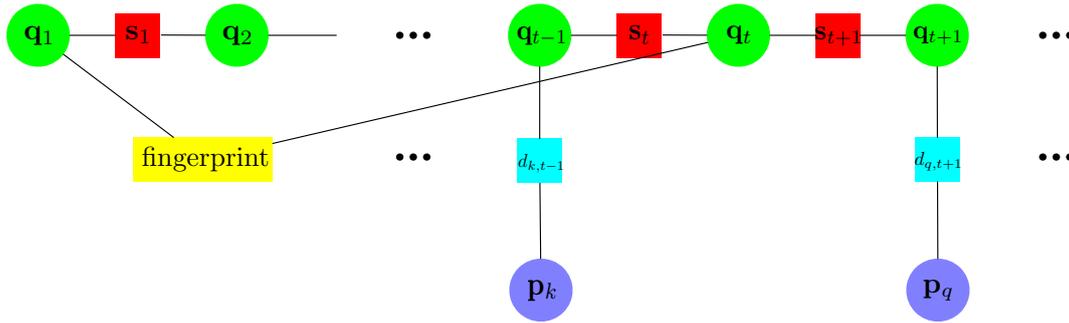


Figure 4. An illustration of the factor graph in BLE beacon implementation in our method.

3.2.2. Error Terms for PDR-Based Constraints

As mentioned before, the pose variable \mathbf{q}_t here has three components $[x_t, y_t, \theta_t]^T$, which are the coordinate positions and the heading. From the PDR algorithm, the pose changes can be “observed” and can be written as

$$\mathbf{s}_t^{pdr} = \begin{bmatrix} L_t^{pdr} \\ \Delta\theta_t^{pdr} \end{bmatrix} \quad (14)$$

where L_t^{pdr} and $\Delta\theta_t^{pdr}$ are the “observed” stride length and heading change, respectively. As mentioned before, the error term for PDR-based constraints is dependent on adjacent pose variables \mathbf{q}_{t-1} , \mathbf{q}_t and the “observed” stride length L_t^{pdr} and heading change $\Delta\theta_t^{pdr}$ (components of \mathbf{s}_t^{pdr}). We write it as

$$\mathbf{e}_{pdr}(\mathbf{q}_{t-1}, \mathbf{q}_t, \mathbf{s}_t^{pdr}) = \begin{bmatrix} \sqrt{(x_t^{\mathbf{q}} - x_{t-1}^{\mathbf{q}})^2 + (y_t^{\mathbf{q}} - y_{t-1}^{\mathbf{q}})^2} \\ \theta_t^{\mathbf{q}} - \theta_{t-1}^{\mathbf{q}} \end{bmatrix} - \begin{bmatrix} L_t^{pdr} \\ \Delta\theta_t^{pdr} \end{bmatrix} \quad (15)$$

where the $x_t^{\mathbf{q}}$, $x_{t-1}^{\mathbf{q}}$ and $\theta_t^{\mathbf{q}}$ are the components taken from the pose variable \mathbf{q}_t . Then the sum of the square error terms derived from PDR constraints can be written as

$$F_{pdr}(\cdot) = \sum_t \mathbf{e}_{pdr}(\mathbf{q}_{t-1}, \mathbf{q}_t, \mathbf{s}_t^{pdr})^T \mathbf{\Omega}_{pdr} \mathbf{e}_{pdr}(\mathbf{q}_{t-1}, \mathbf{q}_t, \mathbf{s}_t^{pdr}) \quad (16)$$

where the error term $\mathbf{e}_{pdr}(\mathbf{q}_{t-1}, \mathbf{q}_t, \mathbf{s}_t^{pdr})$ is from Equation (15). The matrix $\mathbf{\Omega}_{pdr}$ can be considered as a weighting matrix for the square error term. Normally, if the “observation” from the PDR results is accurate, the weighing matrix should be larger and thus making the square error term more significant in the overall cost function. In graph optimization, the weighting matrix is the inverse matrix of the noise variance. This also holds true for other types of error terms. The weighting matrix $\mathbf{\Omega}_{pdr}$ is a 2×2 matrix and here it is:

$$\mathbf{\Omega}_{pdr} = \begin{bmatrix} 1/0.5 & 0 \\ 0 & 1/0.1 \end{bmatrix} \quad (17)$$

here we consider the noise variance of the step length is 0.5 m^2 and the noise variance of heading is 0.1 rad^2 for each step in PDR.

To start the optimization, the initial values of the pose variable is considered the estimated poses from the PDR algorithm. If only one type of constraints (PDR-based constraints) exist in the cost function, the optimal poses should be the initial poses. In such a situation, the cost function is zero.

3.2.3. Error Terms for Beacon Position Constraints

As mentioned before, the RSSI readings from BLE beacons have fast fading nature. Therefore, in our implementation, a pre-processing procedure is added to smooth the RSSI. Here the median value of the RSSI from a beacon within a time window (with duration 1 s) is taken as the current RSSI.

To estimate the distance from a beacon, a path-loss model is needed. Here we adopt the path-loss model in [23],

$$\text{RSSI} = -(10n \log_{10} d + A) \quad (18)$$

where parameter A is the absolute RSSI value represented by dBm at 1 m away from the beacon; n is a parameter related to the signal propagation environment and d is the distance from the beacon. In our implementation, we use the method proposed in [9] to estimate the parameters.

The path-loss model with pre-defined parameter in our implementation is

$$\text{RSSI} = -(10 \times 2.5 \log_{10} d + 30) \quad (19)$$

An issue worth noting is that the model will be inaccurate when readings are higher than 0 dBm. However, we have checked our measurements and we have found out that none of RSSI readings are higher than 0 dBm. In our implementation, the BLE beacons adopted have the largest emitting power of 4 dBm and it is set to 4 dBm here. The reason maybe that, in the experiments, the beacons are installed on the walls and the person does not go too near to the walls.

Here we only use the RSSI if it is larger than -100 dBm. A simple experiment is performed to explain the reasons. In an indoor corridor, the user holding the receiver starts from a place far away from the beacon, walks near the beacon, and then leaves. The RSSI readings are shown in Figure 5. We can see that the RSSI readings at lowest are a bit lower than -100 dBm (in this case, it is -105 dBm). This corresponds to the situation that the user is far away from the beacon, and the signal is buried in noise. In our implementation, we consider -100 dBm as the sensitivity of the receiver and RSSI readings smaller than -100 dBm (though seldomly seen) are discarded.

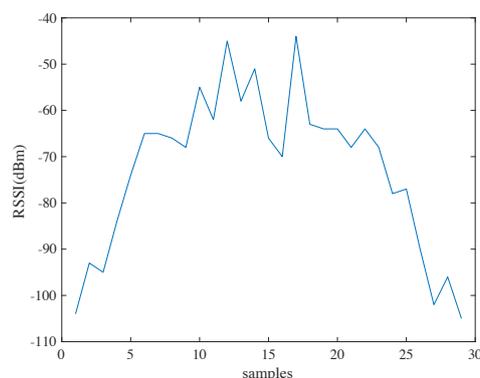


Figure 5. The RSSI readings when a user holding the receiver (phone) starts from a place far away from the beacon, walks near the beacon, and then leaves.

Assuming a RSSI reading for the k^{th} beacon is available at the t^{th} step. From the path-loss model, a distance $d_{k,t}$ can be derived from Equation (19). Then the error term derived from range-based constraint should be dependent on the pose variable \mathbf{q}_t , the beacon position variable \mathbf{p}_k and the derived distance $d_{k,t}$. It can be written as

$$e_{range}(\mathbf{q}_t, \mathbf{p}_k, d_{k,t}) = \sqrt{(x_t^{\mathbf{q}} - x_k^{\mathbf{p}})^2 + (y_t^{\mathbf{q}} - y_k^{\mathbf{p}})^2} - d_{k,t} \quad (20)$$

where $x_t^{\mathbf{q}}$ and $y_t^{\mathbf{q}}$ are taken from the components in the pose variable \mathbf{q}_t ; $x_k^{\mathbf{p}}$ and $y_k^{\mathbf{p}}$ are taken from the beacon position variable \mathbf{p}_k . Then the sum of square error terms derived from path-loss model and RSSI readings can be written as

$$F_{range}(\cdot) = \sum e_{range}(\mathbf{q}_t, \mathbf{p}_k, d_{k,t})^2 \Omega_{range} \quad (21)$$

where the sum is over all available ranges derived from RSSI readings. Again, Ω_{range} is the weight of the range-based square errors. Ω_{range} is a scaler 1/5 and we consider the noise variance in range estimation is 5 m².

Another importance thing worth noting is that the initial values for the beacon position variables is needed to start the optimization. Here we consider no heuristic information about the beacons are available including the number of beacons and their positions. To start the optimization with initial guess of the beacon positions, we assume the initial beacon positions are at the poses from the PDR results when the beacons' RSSI are available for the first time.

3.2.4. Error Terms for Fingerprint Matching Constraints

For fingerprinting-based positioning using BLE beacon, an RFM should be established in the offline phase. The RFM consists of many fingerprints collected at different known positions. In our implementation, the user walks in the area collecting fingerprints. The optimal poses can provide the position estimations in the RFM.

In fingerprinting-based BLE positioning method, the distances between fingerprints can indicate the correlations for the positions. If two fingerprints are similar (with small distance), then the positions where the fingerprints are collected should be also with small distance (high correlation). We also use the mentioned feature for generating constraints through fingerprint matching. For such purposes, the distance metric is defined firstly according to [24],

$$d(\mathbf{RSSI}_j, \mathbf{RSSI}_k) = \sqrt{\frac{\sum_{i=1}^N (\text{RSSI}_j^{(i)} - \text{RSSI}_k^{(i)})^2}{N}} \quad (22)$$

where the subscripts denote two different RSSI readings; $\text{RSSI}_j^{(i)}$ is the i^{th} component in the vector \mathbf{RSSI}_j ; N is the number of the beacons. Two RSSI vectors may have readings from different beacons. In this case, we add default readings to the RSSI vector such that both vectors have readings from the same beacon set. Here the beacon set denotes the union set of the two original beacon sets. An example is shown in Figure 6, where the original \mathbf{RSSI}_j and \mathbf{RSSI}_k have three and two components, respectively. After completion, they both become a 4-component vector with readings from the same beacons. The added default RSSI reading is set to -120 dBm in our implementation.

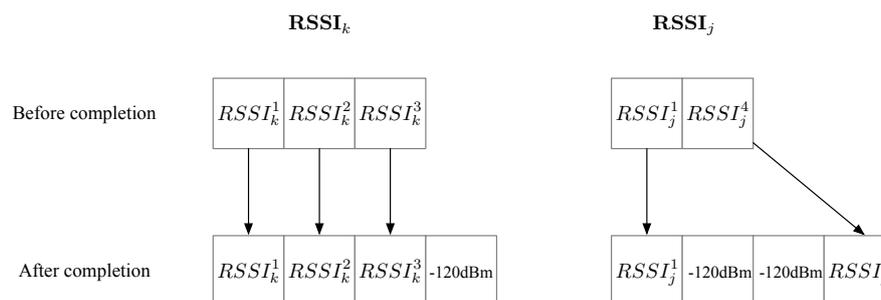


Figure 6. An example for the RSSI completion process before calculation of the RSSI distance.

Assuming the two RSSI readings \mathbf{RSSI}_j and \mathbf{RSSI}_k correspond to the pose variables \mathbf{q}_j and \mathbf{q}_k respectively. The distance $d_{rssi}(j, k)$ between the RSSI readings can be calculated according to Equation (22). If the RSSI distance is less than a threshold $d_{rssi,thres}$ indicating high correlation on the

corresponding positions, the error term of fingerprint matching-based constraint exists. The error term is

$$e_{rssi}(\mathbf{q}_j, \mathbf{q}_k, \mathbf{RSSI}_j, \mathbf{RSSI}_k) = \begin{cases} 0, & d_{pos}(j, k) < d_{pos, min} \\ e_{ubound}, & d_{pos}(j, k) > d_{pos, max} \\ e_{ubound} \frac{d_{pos}(k, q) - d_{pos, min}}{d_{pos, max} - d_{pos, min}}, & \text{otherwise} \end{cases} \quad (23)$$

where

- $d_{pos}(j, k)$ is the position distance calculated from the pose variables \mathbf{q}_j and \mathbf{q}_k .
- $d_{pos, min}$ and $d_{pos, max}$ are the two position distance thresholds. If the position distance is less than $d_{pos, min}$, then the error should be 0, because we allow some position differences for high correlation poses. If the position distance is larger than $d_{pos, max}$, the error is considered to reach an upper bound e_{ubound} .
- If the position distance is between the two position thresholds, the error grows linearly with the position distance $d_{pos}(k, q)$.

The thresholds for fingerprint-based constraints in our implementation are in Table 1. Noting that the choice of these thresholds is similar to the implementation in [25], where the RSSI are from Wi-Fi APs. This set of parameters also works fine in our implementation. We will perform an exhaustive search for the optimal parameters in the future.

Table 1. The thresholds adopted for fingerprint-based constraints in our implementation.

Thresholds	Value
$d_{rssi, thres}$	8 dBm
$d_{pos, min}$	5 m
$d_{pos, max}$	20 m
e_{ubound}	1

Then the sum of square error terms derived from fingerprint matching can be written as

$$F_{finger}(\cdot) = \sum e_{rssi}(\mathbf{q}_j, \mathbf{q}_k, \mathbf{RSSI}_j, \mathbf{RSSI}_k)^2 \Omega_{finger} \quad (24)$$

where the sum is over any two pose indexes whose fingerprint matches. Again, Ω_{finger} is the weight to control the error significance. Ω_{finger} is again a scaler 1/10 and we consider the noise in fingerprinting matching is 10 m².

Then the overall cost function is the sum of the three types of constraints.

$$F(\cdot) = F_{pdr}(\cdot) + F_{range}(\cdot) + F_{finger}(\cdot) \quad (25)$$

4. Experiment

4.1. Settings

In our experiments, we adopt the Google nexus 6p smart phone to collect the data. A custom Android application is designed for such purposes. In this application, the inertial data, including the acceleration, angular rate and the magnetometer readings are collected at a rate of 50 Hz. During the BLE scanning, the scan mode for the system is set to scan with low latency for continuous and fast scanning of the BLE beacons. In the Android system, this is done by setting the ScanSettings class to a constant as SCAN_MODE_LOW_LATENCY. The inertial data are processed in real time to estimate the coarse position changes for the person holding the phone in the hand adopting the PDR algorithm. The position changes along with the BLE scanning results (including the beacons' names, mac addresses

and RSSI values) are stored on the phone for offline processing. The BLE beacons adopted in our implementation are the Smart Beacons from the Bright Beacon company [26]. In our implementation, we set the broadcasting interval of the beacons to 300 ms and the emitting power to 4 dBm.

4.2. Accuracy for Positioning the Beacons

As shown in Figure 7, 48 BLE beacons are installed in an office site on the walls or on the surfaces of furniture. The beacons are installed approximate evenly covering the area. The size of the area is approximately $90\text{ m} \times 37\text{ m}$. Although there are 48 beacons installed, we manually take out half of the beacons (24 beacons) to see how the different number of beacons affect the performance of the proposed approach (in Figure 7). We refer to the different number of beacons as dense beacons and sparse beacons in the remaining of the paper. In this setting, the mean number of beacons over the minimum signal threshold for the dense beacon situation is about 6 and for the sparse case the number is about 3. The beacon positions are measured adopting a total station with sufficient accuracy and is regarded as ground truth positions. However, the ground truth positions of the beacons are only adopted for calculating the positioning errors of the beacons. During the optimization, we assume that the positions of the beacons are unknown and is estimated from the optimization process. We have labeled the beacons with different numbers and made a table of the numbers and the corresponding mac address of the beacons. In this way, we can identify the different beacons. A person holding the smart phone walked in the site for about 30 min with a total walking distance of about 1.6 km. The collected data is adopted for generating the aforementioned cost function. By minimizing the cost function, the beacon positions can be estimated. With the ground truth of the beacon positions, the position estimation errors can be acquired.

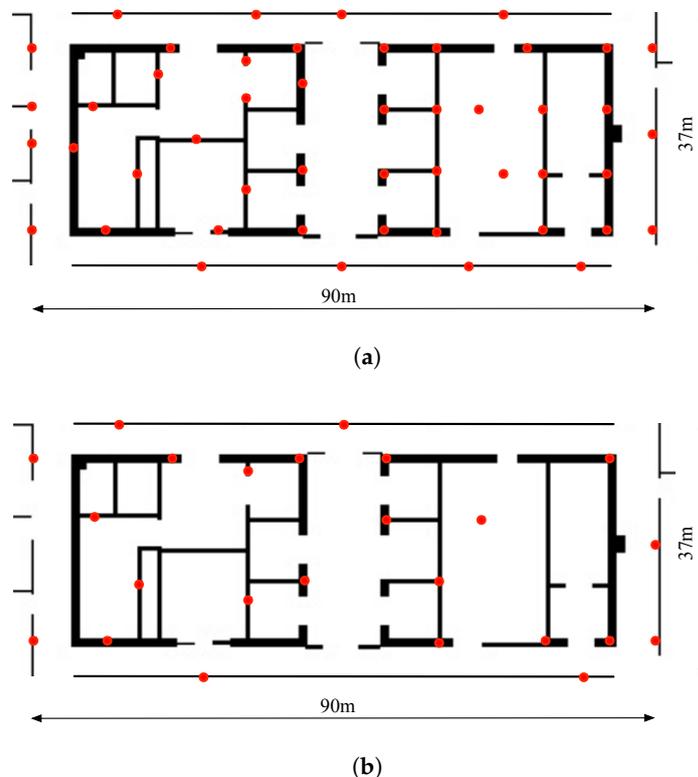


Figure 7. The positions of BLE beacons installed in the site (on the walls or on the furniture surfaces) (a) Densely deployed BLE beacons (b) Sparsely deployed beacons. In our implementation, we just ignore some of the RSSI readings to simulate the situation of sparsely deployed beacons

Figures 8 and 9 show the histograms of the errors for adopting the two types of error terms combination: PDR constraint terms + beacon position constraint terms and PDR constraint terms + beacon position constraint terms + fingerprint matching constraint terms. We can see that the incorporation of fingerprint matching error terms can improve the accuracy for positioning the beacons. Tables 2 and 3 show some of the error statistics of the two combinations. For the proposed method (3 types of error terms combination), the mean positioning error of the beacons is only about 1.27 m in the dense beacon situation and 2.26 m for the sparse beacon situation. We show later that this level of beacon errors, especially for the dense beacon situation is not significant in deteriorating the performance of beacon-based positioning. However, for certain, this approach can estimate the beacon positions which is exempt from the heavy working load for carefully measure the beacon positions. In this experiment, it took 3 persons using nearly the whole day to measure the ground truth positions of the 48 beacons. Noting that here some large errors for beacon position estimations still exist. This may be due to the fact that the large noises in PDR results and the large noise in RSSI readings while the user is walking.

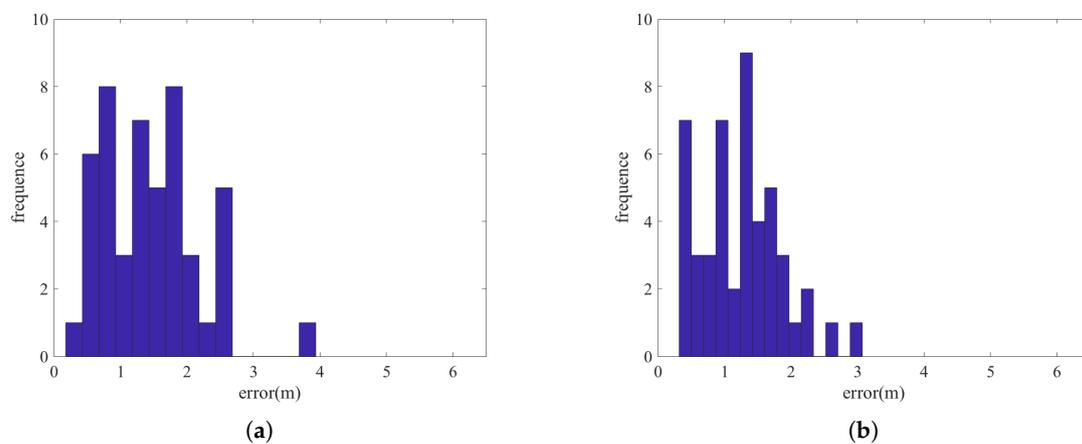


Figure 8. The histogram of beacon positions errors for dense beacons situation (a) Cost function with PDR constraint error terms and beacon position constraint error terms (b) Cost function with PDR constraint error terms, beacon position constraint error terms and fingerprint matching constraint error terms.

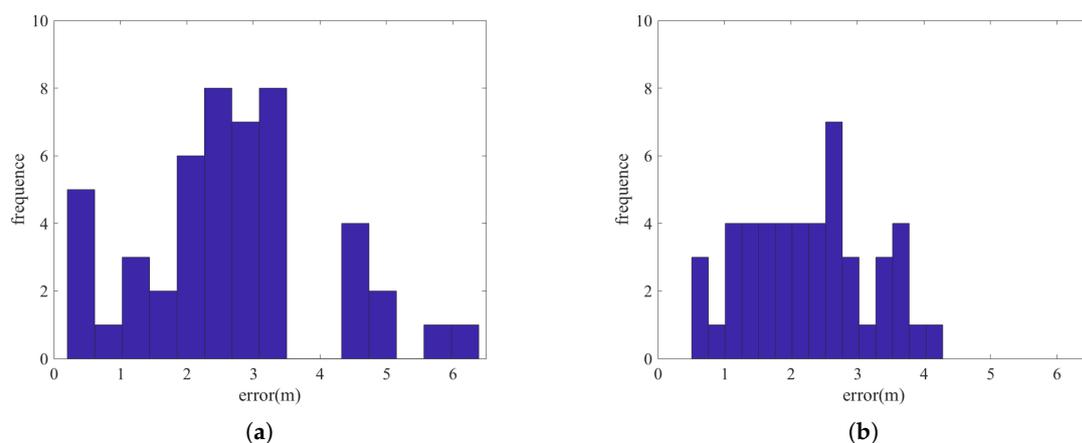


Figure 9. The histogram of beacon positions errors for sparse beacons situation (a) Cost function with PDR constraint error terms and beacon position constraint error terms (b) Cost function with PDR constraint error terms, beacon position constraint error terms and fingerprint matching constraint error terms.

Table 2. Some of the statistics of the beacon position errors for dense beacons situation.

Cost Function Error Terms	Mean Error (m)	Median Error (m)	Maximum Error (m)
PDR + beacon position	1.45	1.42	3.94
PDR + beacon position + fingerprint matching	1.27	1.28	3.07

Table 3. Some of the statistics of the beacon position errors for sparse beacons situation.

Cost Function Error Terms	Mean Error (m)	Median Error (m)	Maximum Error (m)
PDR + beacon position	2.68	2.65	6.39
PDR + beacon position + fingerprint matching	2.26	2.31	4.28

4.3. Accuracy for Beacon Based Positioning

As mentioned before, there are two types of beacon-based positioning method: range-based and fingerprinting-based. For range-based method, the positions of the beacons need to be known for positioning. For fingerprinting-based method, an RFM should be known for positioning. The proposed method in this paper gives an efficient way to estimate the beacon positions and to establish an RFM. To verify that the quality of the estimation for the beacon positions and the RFM is sufficient to support beacon-based positioning, another experiment is carried out. The person walked into the site again collecting BLE scanning RSSI readings. These RSSI readings are regarded as test data for showing the accuracy of beacon-based positioning. Here as the beacon-based positioning normally has many outliers, we adopt a simple extended Kalman filter (EKF)-based positioning method to fuse the beacon positioning results and the position update data from PDR. Noting that all the results including the results presented in Sections 4.3.1 and 4.3.2 are results from the EKF. The adoption of the EKF for fusing PDR results and beacon positioning results is trivial and can be found in publications such as [11]. Therefore, how the EKF is implemented is not described in detail in the manuscript.

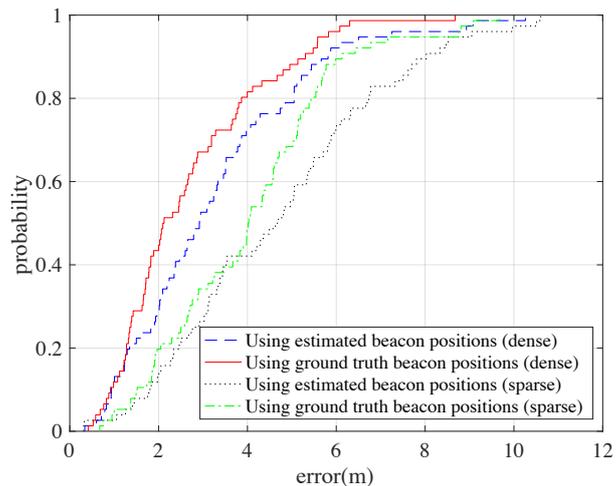
For measuring the ground truth of the persons, we have put some stickers to the ground and labeled the stickers with numbers on it to distinguish them. The stickers can be regarded as landmarks and the positions of landmarks are also measured through a total station prior to the positioning experiments. During the positioning experiments, the person can record the time and the landmark label during walking. This is done through pressing buttons on the APP designed for collecting data. In this way, the ground truth positions of the person when he/she walks to the landmarks are known. The positioning errors can then be determined through the differences between the estimated positions and the ground truth positions. In fact, we have set 50 landmarks roughly even distributed in the area. Since the positions of the landmarks should not affect the final results much, we did not show the positions of the landmarks.

4.3.1. Accuracy for Range-Based Positioning

Here a classical type of range-based positioning is selected to test the positioning performance: the LSE method. By minimizing the square sums of the distance errors to the beacon positions, an optimal position can be found. Noting that here, the beacon positions should be known before positioning. Figure 10 shows the cumulative distribution functions (CDFs) of the positioning errors using ground truth beacon positions and estimated beacon positions (from the results in Section 4.2). Table 4 shows some of the error statistics of the two situations. It is not surprising that the positioning performance using the ground truth beacon positions is better than using the estimated beacon positions. However, the performance drop-down is not very significant with the mean error increase from 2.66 m to 3.25 m in the dense beacon situation and from 4.03 m to 4.69 m in the sparse beacon situation.

Table 4. Error statistics of range-based positioning using ground truth beacon positions and estimated beacon positions.

Beacon Positions	Mean Error (m)	50% Error(m)	70% Error (m)
ground truth positions (dense)	2.66	2.11	3.95
estimated positions (dense)	3.25	2.92	3.18
ground truth positions (sparse)	4.03	4.02	5.09
estimated positions (sparse)	4.69	4.66	5.90

**Figure 10.** The CDFs of positioning errors using the ground truth beacons positions and the estimated beacon positions.

4.3.2. Accuracy for Fingerprinting-Based Positioning

Our method can provide an RFM adopting the estimated positions of the person and the collected BLE RSSI readings. The RFM can be adopted for fingerprinting-based positioning. The positions of the collected fingerprints are firstly estimated adopting the proposed method. Then we partition the area to small grids with the size of $0.5\text{ m} \times 0.5\text{ m}$. The RSSI fingerprints at the centers of the grids are predicted through a simple Gaussian regression process adopting nearby fingerprints. These predicted fingerprints are then adopted to form the RFM in our implementation. Here the weighted kNN method is chosen for beacon-based positioning (in our case $k = 3$). The CDFs of positioning errors is shown in Figure 11. The mean error of fingerprinting-based positioning here is about 2.78 m for dense beacon situation and about 4.11 m for sparse beacon situation, which are considered sufficient for many indoor positioning applications. Noting that here the positioning performance is slightly better than the LSE method. This is consistent with the results in [11].

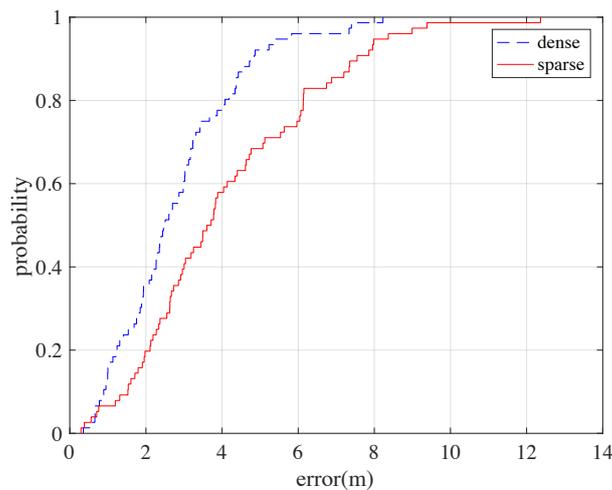


Figure 11. The CDF of positioning errors using the estimated RFM.

4.3.3. Accuracy Comparisons for Fingerprinting Based Positioning and Range-Based Positioning

Fingerprinting-based positioning normally outperforms range-based positioning. This is also true in our experiments. The fingerprinting-based results and range-based results should be compared under the same condition. In fact, the fingerprinting-based results using the estimated Reference Fingerprint Map (RFM) should be compared to the range-based results using estimated beacon positions. We directly take the CDFs from our experiments and compare them in Figure 12a,b. It can be seen that the performance of fingerprinting-based positioning is better than range-based positioning under both dense and sparse beacon situations. Table 5 gives the mean error comparisons. We can see that the mean error of fingerprinting-based is 0.47 m less than range-based under dense beacon situation and 0.58 m less under sparse beacon situation. Noting that these results are directly taken from Sections 4.3.1 and 4.3.2, and is presented differently here. The results have shown that the differences are not huge. This may be due to the reason that range-based and fingerprint-based methods suffer or deteriorate differently on the inaccuracies of estimated beacon positions and RFM, respectively. This leads to another interesting topic which can be studied in the future.

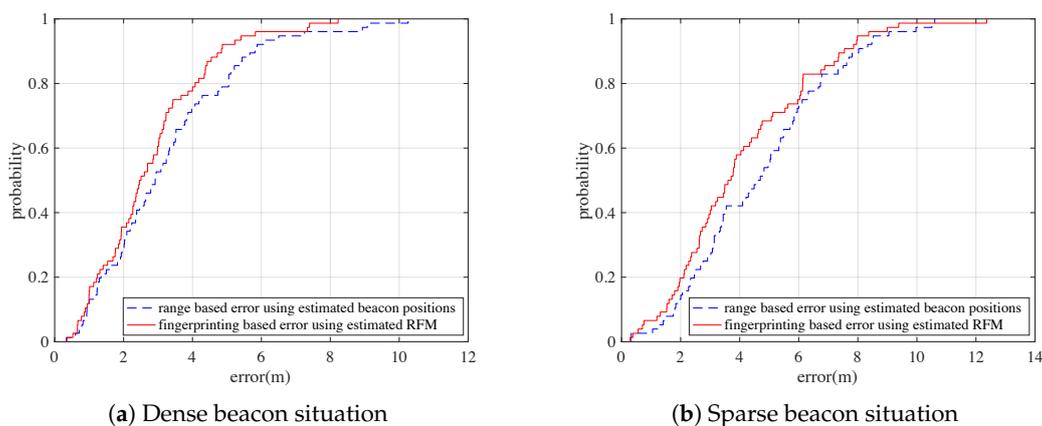


Figure 12. CDF comparison for range-based errors and fingerprinting-based errors under dense and sparse beacon situations using estimated beacon positions and RFM.

Table 5. Mean error comparisons for range-based positioning and fingerprinting-based positioning using estimated beacon positions and RFM.

	Dense Beacon Situation (m)	Sparse Beacon Situation (m)
range-based	3.25	4.69
fingerprinting-based	2.78	4.11

4.3.4. Accuracy Comparisons of the Proposed Method and Another Method

The proposed method provide an efficient way for estimating the beacon positions and the RFM without dedicated and time costly surveying process at the price of loss of positioning accuracy. However, considering the time and energy saved from dedicated surveying (it took three persons a whole day to measure the beacon positions using a total station in our situation), the method is meaningful.

A comparison to the method in [11] is presented here. The results comparisons are shown in Table 6. Noting that the accuracy of our method indeed is worse than that reported in [11]. However, for range-based case, our method is exempt of dedicated measuring of beacon positions, while in [11] the beacons are carefully measured with a range finder. For fingerprinting-based case, our method adopts an estimated RFM, while in [11] the RFM is constructed by exhaustive surveying 150 points in the site.

Table 6. Mean positioning error comparisons of our method and the method reported in [11].

	Dense Beacon Situation (m)	Sparse Beacon Situation (m)
range-based (our method)	3.25	4.69
range-based (method in [11])	2.57	3.93
fingerprinting-based (our method)	2.78	4.11
fingerprinting-based (method in [11])	1.67	2.83

5. Conclusions

Most BLE beacon-based indoor positioning methods need some pre-requisites prior to positioning. For range-based methods, the beacons' positions are needed. For fingerprinting-based methods, the RFM are needed. Surveying the beacon positions or the RFM is normally an energy intensive task. This paper proposes an easy way to estimate the pre-requisites for BLE beacon-based indoor positioning using graph optimization. On the hardware perspective, only the BLE beacons and a BLE-enabled mobile device is needed. The mean errors of the beacon positions are 1.27 m in dense beacon situation and 2.26 m in sparse beacon situation. Using the estimated pre-requisites, the BLE beacon positioning performance is considered enough for many indoor positioning applications, although it is worse than methods adopting accurate beacon positions and RFM.

In the future, we plan a deeper and more complete research into the proposed method on a wide range of affecting factors, including the number of beacons needed, the effect of walking persons in the environment, the effect of more sophisticated path-loss model, the upper bound error, and so on.

Author Contributions: Z.Z., L.L. and L.Z. conceived and worked together to achieve this work. Z.Z. wrote the paper. L.Z performed the experiments. Y.F. gave revisions on the paper.

Funding: This work was supported by National Key Research and Development Program of China Special Fund for Cyberspace Security (No. 2017YFB0802900).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BLE	Bluetooth low-energy
PbS	position-based service
RFM	reference fingerprinting map
RSSI	received signal strength indication
PDR	pedestrian dead reckoning
GNSS	global navigation satellite system
APs	access points
RF	radio frequency
LSE	least square estimation
kNN	k-nearest neighbor
IMUs	inertial measurement units
ZVU	zero-velocity update
RM	radio map
SLAM	simultaneous localization and mapping
EKF	extended Kalman filter
CDFs	cumulative distribution functions

References

1. Ruiz, A.R.J.; Granja, F.S. Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2106–2117. [[CrossRef](#)]
2. Martínez-Sala, A.S.; Losilla, F.; Sánchez-Aarnoutse, J.C.; García-Haro, J. Design, Implementation and Evaluation of an Indoor Navigation System for Visually Impaired People. *Sensors* **2015**, *15*, 32168–32187. [[CrossRef](#)] [[PubMed](#)]
3. Bagosi, T.; Baruch, Z. Indoor localization by WiFi. In Proceedings of the IEEE 7th International Conference on Intelligent Computer Communication and Processing, ICCP 2011, Cluj-Napoca, Romania, 25–27 August 2011; pp. 449–452. [[CrossRef](#)]
4. Park, J.G.; Charrow, B.; Curtis, D.; Battat, J.; Minkov, E.; Hicks, J.; Teller, S.; Ledlie, J. Growing an Organic Indoor Location System. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, Boston, MA, USA, 6–9 June 2004; ACM: New York, NY, USA, 2010; pp. 271–284. [[CrossRef](#)]
5. Alarifi, A.; Al-Salman, A.; Alsaleh, M.; Alnafessah, A.; Al-Hadhrami, S.; Al-Ammar, M.A.; Al-Khalifa, H.S. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors* **2016**, *16*, 707. [[CrossRef](#)] [[PubMed](#)]
6. Hazas, M.; Hopper, A. Broadband ultrasonic location systems for improved indoor positioning. *IEEE Trans. Mob. Comput.* **2006**, *5*, 536–547. [[CrossRef](#)]
7. Godha, S.; Lachapelle, G. Foot mounted inertial system for pedestrian navigation. *Meas. Sci. Technol.* **2008**, *19*, 75202. [[CrossRef](#)]
8. Alvarez, J.C.; Alvarez, D.; López, A.; González, R.C. Pedestrian Navigation Based on a Waist-Worn Inertial Sensor. *Sensors* **2012**, *12*, 10536–10549. [[CrossRef](#)] [[PubMed](#)]
9. Wang, Y.; Yang, X.; Zhao, Y.; Liu, Y.; Cuthbert, L. Bluetooth positioning using RSSI and triangulation methods. In Proceedings of the 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2013; pp. 837–842. [[CrossRef](#)]
10. Cinefra, N. An Adaptive Indoor Positioning System Based on Bluetooth Low Energy RSSI. Ph.D. Thesis, Politecnico di Milano, Milan, Italy, January 2013.
11. Zhuang, Y.; Yang, J.; Li, Y.; Qi, L.; El-Sheimy, N. Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons. *Sensors* **2016**, *16*. [[CrossRef](#)] [[PubMed](#)]
12. Davidson, R.; Akiba, C.; Cufi, C.; Townsend, K. *Getting Started with Bluetooth Low Energy*; O'Reilly Media: Sebastopol, CA, USA, 2014; Chapter 1.
13. Faragher, R.; Harle, R. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In Proceedings of the 27th International Technical, Clearwater, FL, USA, 8–12 September 2014.

14. Čabarkapa, D.; Grujić, I.; Pavlović, P. Comparative analysis of the Bluetooth Low-Energy indoor positioning systems. In Proceedings of the 2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Niš, Serbia, 14–17 October 2015; pp. 76–79. [[CrossRef](#)]
15. Dong, B.; Burgess, T.; Neuner, H. indoo.rs CaLibre—Unsupervised database free RSSI calibration indoor navigation. In Proceedings of the 2017 European Navigation Conference (ENC), Lausanne, Switzerland, 9–12 May 2017; pp. 198–205. [[CrossRef](#)]
16. Nilsson, J.O.; Gupta, A.K.; Handel, P. Foot-mounted inertial navigation made easy. In Proceedings of the IPIN 2014—2014 International Conference on Indoor Positioning and Indoor Navigation, Busan, Korea, 27–30 October 2014; pp. 24–29. [[CrossRef](#)]
17. Jiménez, A.R.; Seco, F.; Prieto, J.C.; Guevara, J. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In Proceedings of the 2010 7th Workshop on Positioning, Navigation and Communication, Dresden, Germany, 3 November 2010; pp. 135–143. [[CrossRef](#)]
18. Lan, K.; Shih, W. Using simple harmonic motion to estimate walking distance for waist-mounted PDR. In Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC), Paris, France, 1–4 April 2012; pp. 2445–2450. [[CrossRef](#)]
19. Li, B.; Gallagher, T.; Dempster, A.G.; Rizos, C. How feasible is the use of magnetic field alone for indoor positioning? In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia, 13–15 November 2012; pp. 1–9. [[CrossRef](#)]
20. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. G2o: A general framework for graph optimization. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3607–3613. [[CrossRef](#)]
21. Scaramuzza, D.; Fraundorfer, F. Visual Odometry [Tutorial]. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92. [[CrossRef](#)]
22. Lourakis, M.I.A.; Argyros, A.A. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Softw.* **2009**, *36*, 1–30. [[CrossRef](#)]
23. Albowicz, J. Receive position estimation in sensor network. In Proceedings of the IEEE International Conference on Network Protocols, Riverside, CA, USA, 11–14 November 2001.
24. Gu, Y.; Zhou, C.; Wieser, A.; Zhou, Z. WiFi-based trajectory alignment, calibration and crowdsourced site survey using smart phones and foot-mounted IMUs. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017; pp. 1–6. [[CrossRef](#)]
25. Nowicki Michał and Skrzypczyński, P. Indoor Navigation with a Smartphone Fusing Inertial and WiFi Data via Factor Graph Optimization. In *Mobile Computing, Applications, and Services*; Sigg, S., Nurmi, P., Salim, F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 280–298.
26. Smart Beacon. Available online: http://www.brtbeacon.com/main/product_mall.shtml (accessed on 10 October 2018).

