

Article

Graph Optimization Model Fusing BLE Ranging with Wi-Fi Fingerprint for Indoor Positioning

Rong Zhou, Puchun Chen, Jing Teng *  and Fengying Meng

School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China; zhourong@ncepu.edu.cn (R.Z.); 120202227078@ncepu.edu.cn (P.C.); 120202227098@ncepu.edu.cn (F.M.)

* Correspondence: jing.teng@ncepu.edu.cn

Abstract: To improve the user's positioning accuracy of a Wi-Fi fingerprint-based positioning algorithm, this study proposes a graph optimization model based on the framework of g^2o that fuses a Wi-Fi fingerprint and Bluetooth Low Energy (BLE) ranging technologies. In our model, the improvement in positioning can be formulated as a nonlinear least-squares optimization problem that a graph can represent. The graph regards users as nodes and our self-designed error functions between users as edges. In the graph, the nodes obtain the initial coordinates through Wi-Fi fingerprint positioning, and all error functions aggregate to a total error function to be solved. To improve the solution effect of the total error function and weaken the influence of measurement error, an information matrix, an edge selection principle, and a Huber kernel function are introduced. The Levenberg–Marquardt (LM) algorithm is used to solve the total error function and the affine transformation estimation is used for the drifting solution. Through experiments, the influence of the threshold in the Huber kernel function is explored, the relationship between the number of nodes in the graph and the optimization effect is analyzed, and the impact of the distribution of nodes is researched. The experimental results show improvements in the positioning accuracy of four common Wi-Fi fingerprint-matching algorithms: KNN, WKNN, GK, and Stg.



Citation: Zhou, R.; Chen, P.; Teng, J.; Meng, F. Graph Optimization Model Fusing BLE Ranging with Wi-Fi Fingerprint for Indoor Positioning. *Sensors* **2022**, *22*, 4045. <https://doi.org/10.3390/s22114045>

Academic Editor: David Plets

Received: 11 April 2022

Accepted: 24 May 2022

Published: 26 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: indoor positioning; Wi-Fi fingerprint; BLE ranging; graph optimization

1. Introduction

With increasing demand for location-based services (LBSs), indoor positioning has become a research topic of much interest [1]. In recent years, some RF-based methods for precise indoor positioning have emerged, mainly in two categories: trilateration [2] and fingerprint [3]. The trilateration method requires the physical distance between the target point and at least three anchor points. The measurements used to calculate physical distance in trilateration include received signal strength indication (RSSI) [4], time difference of arrival (TDOA) [5], and time of arrival (TOA) [6]. Compared to TDOA and TOA, RSSI-based ranging avoids the need for expensive clock synchronization systems and additional hardware, and has the advantages of low cost, easy deployment, and simple calculation. The fingerprint method is regarded as superior to trilateration in more complex indoor environments, where the interference, reflection, and refraction of signal forces are superimposed onto the specific points [7].

Wi-Fi technology is widely used in indoor target intrusion detection and localization [8,9]. The Wi-Fi wireless access points (APs) have been widely deployed in indoor environments because of the improvement of wireless sensor technology and the growing demand for network communication, and they provide a physical device foundation for the application of Wi-Fi fingerprint positioning. The Wi-Fi fingerprint method is regarded as low-cost and simple because it requires no additional hardware and performs satisfactorily in complex indoor environments [10]. However, room exists for improvement in its positioning accuracy in complex indoor environments such as underground parking lots.

Multi-sensor fusion positioning is a common method to improve positioning accuracy. Fusing BLE and Wi-Fi for positioning inherits their advantages in low costs and the widespread deployment of basic equipment. Bluetooth Low Energy (BLE) products can run for months on a battery with low-power communication. Its transmission distance of up to 50–100 m meets the requirements of indoor positioning for wireless signals [11,12]. BLE technology has been embedded in billions of products, such as mobile phones, earphones, computers, and cars, providing the basis for its application in indoor positioning. Subsequent studies using BLE beacons for indoor positioning [13] provide a theoretical basis for our work.

Multiple users can use smart devices to obtain initial positioning coordinates by the Wi-Fi fingerprint method. Because BLE signals from other devices are available everywhere, the physical distance between users can be measured by BLE ranging technology and can act as additional constraints to adjust user coordinates.

We propose a graph optimization model combining a Wi-Fi fingerprint and BLE ranging technology to improve indoor positioning accuracy. The general process is as follows. The problem can be considered a nonlinear least-squares optimization problem that a graph can represent, where users are regarded as nodes, and our designed error functions between users are edges. We obtain the initial coordinates of nodes based on the Wi-Fi fingerprint method and aggregate all edges to a total error function. The information matrix, edge selection principle, and Huber kernel function are used to control the measurement error of wireless sensors. The Levenberg–Marquardt (LM) algorithm can solve the total error function to obtain the optimized coordinates of the nodes, and the affine transformation estimation can solve the drifting problem of optimized nodes.

The remainder of this paper is organized as follows. Section 2 introduces the related theories of the Wi-Fi fingerprint method, signal propagation of BLE, and graph optimization based on g^2o . Section 3 provides an overview of the proposed model, the error function and total error function, the drifting solution, BLE signal processing, and the BLE ranging model. Section 4 describes our experimental process and results from multiple perspectives and verifies the feasibility and generalization of the proposed graph optimization model. Section 5 summarizes our study, draws conclusions, and suggests further work.

2. Related Work

2.1. Wi-Fi Fingerprint Positioning

Many well-known indoor positioning systems have been developed based on the Wi-Fi fingerprint method, such as RADAR [14], Nibble [15], and Horus [16]. Proposed fingerprint-matching algorithms include the classic k-nearest neighbors (KNN) [17] and WKNN [18], which assigns weights to each reference point based on KNN. The other two matching algorithms (Stg and GK) used in the experimental part of this paper are introduced as follows.

The Stg algorithm [19] selects the strongest AP data in the initial sample filtering, which can not only reduce the size of the fingerprint database, but can also reduce the calculation speed of the fingerprint-matching process. The fingerprint-matching process of Stg algorithm is based on the KNN algorithm. The GK algorithm [20] is based on parametric modeling of a logarithmic RSS stochastic process, in which each RSS is considered to be the mean, and the standard deviation is set as a constant for all observations. The probability value of the test fingerprint at each RP is calculated by the Gaussian kernel density estimator, and the position of the test point is estimated according to the average value of the coordinates of several RPs with the highest probability value.

Neural network and deep-learning technologies are also applied in Wi-Fi fingerprint positioning [21,22], but regardless of the system or algorithm, the Wi-Fi fingerprint method always follows a general process consisting of offline and online phases, as shown in Figure 1.

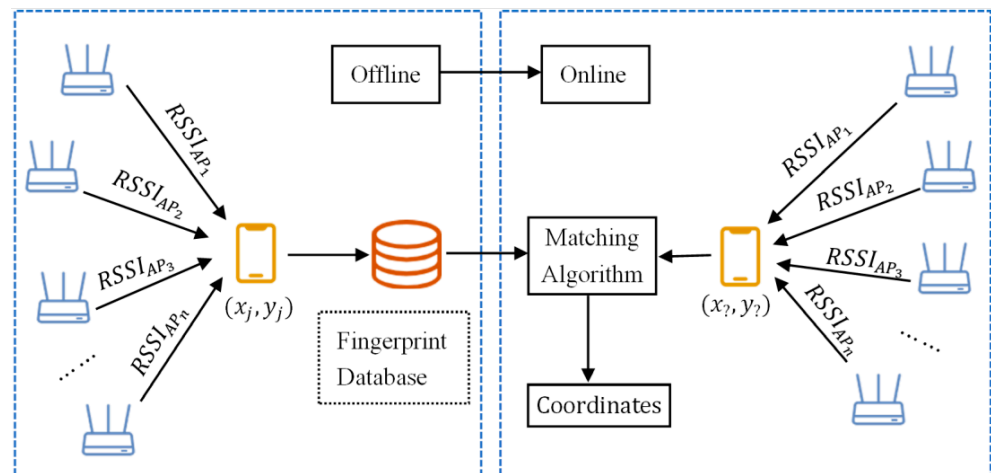


Figure 1. General process of Wi-Fi fingerprint positioning.

In the offline phase, a certain number of APs are deployed, several reference points (RPs) are set up evenly in the indoor environment, and the RSSIs from APs at the RPs are collected to establish a fingerprint database. A fingerprint consists of the physical coordinates and Wi-Fi signal fingerprints of the RPs. Suppose the number of APs involved in positioning is n , the number of RPs is m , and $RSSI_{(i,j)}$ represents the RSSI from AP_i received at RP_j . Since the RSSI is easily affected by environmental changes, multiple RSSI collections are carried out at each RP to build a fingerprint database; so, the k th fingerprint at RP_j can be expressed as

$$FP_j^{(k)} = [RSSI_{(1,j)}^{(k)}, RSSI_{(2,j)}^{(k)}, \dots, RSSI_{(n,j)}^{(k)}, x_j, y_j] \quad (1)$$

If we collect fingerprints k times at RP_j , the Wi-Fi fingerprint database (WFPDB) can be saved as a matrix:

$$WFPDB = \begin{pmatrix} FP_1^{(1)}, FP_1^{(2)}, \dots, FP_1^{(k)} \\ FP_2^{(1)}, FP_2^{(2)}, \dots, FP_2^{(k)} \\ \dots \\ FP_m^{(1)}, FP_m^{(2)}, \dots, FP_m^{(k)} \end{pmatrix} \quad (2)$$

In the online phase, we estimate the positions of users by collecting the RSSIs of different APs in real-time and applying the fingerprint-matching algorithm.

2.2. Signal Propagation of BLE

Common propagation path-loss models include the free space propagation and logarithmic distance path-loss models [23]. The latter is generally used in BLE ranging technology [24,25] and is expressed as

$$P_t(d) = P_t(d_0) - 10\lambda \lg\left(\frac{d}{d_0}\right) + \varepsilon, \quad (3)$$

where $P_t(d)$ is the RSSI when the distance between the Bluetooth transmitting point and receiving point is d , d_0 is the reference distance, $P_t(d_0)$ is the RSSI when the distance between the transmitting and receiving point is d_0 , ε is Gaussian random noise, and λ is the path loss index. For convenience of calculation, usually, $d_0 = 1$ m. To fit the signal propagation model based on the collected RSSI, the simplified model can be expressed as

$$RSSI = A - 10\lambda \lg(d), \quad (4)$$

where $RSSI = P_t(d)$ and $A = P_t(1)$. Both A and λ are empirical values that are related to the indoor environment. By collecting the RSSI of BLE beacons at different distances and fitting the parameters by least-squares, Equation (4) can be used for BLE ranging.

2.3. Graph Optimization Based on g^2o

Many robotics and computer vision studies involve the minimization of nonlinear error functions, a typical example being simultaneous localization and mapping (SLAM) [26–28]. These studies aim to find parameters or state variables that best explain a set of measured values affected by Gaussian noise, where the LM algorithm is used to find the optimal solution [29].

g^2o is a general back-end optimization framework suitable for nonlinear least-squares problems [30], which can be represented graphically in the framework of g^2o , as shown in Figure 2, where each node represents a state variable to be optimized, and an edge between nodes represents a paired observation of the two nodes.

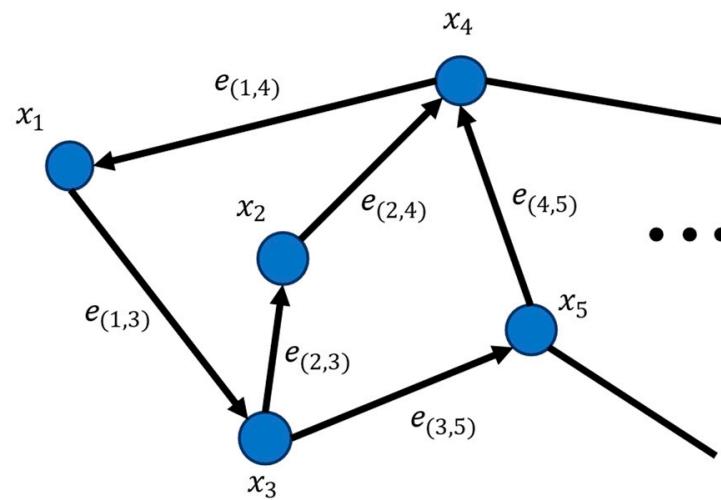


Figure 2. Graphical representation of nonlinear least-squares problems.

Graph optimization based on g^2o represents a nonlinear least-squares problem as a graph and visually displays the structural relationship between the variables. The graph can be directed or undirected, and the corresponding optimization problem can be described as

$$F(x) = e_{(1,3)}^T \Omega_{(1,3)} e_{(1,3)} + e_{(2,3)}^T \Omega_{(2,3)} e_{(2,3)} + e_{(2,4)}^T \Omega_{(2,4)} e_{(2,4)} + e_{(3,5)}^T \Omega_{(3,5)} e_{(3,5)} + e_{(4,5)}^T \Omega_{(4,5)} e_{(4,5)} + \dots, \quad (5)$$

where $F(x)$ is the final total error function to be solved, $e_{(i,j)}$ is the error function between nodes x_i and x_j , and $\Omega_{(i,j)}$ is the information matrix of edge $e_{(i,j)}$. Using BLE ranging data to optimize Wi-Fi fingerprint location results can be abstracted as a graph optimization problem. Users can be defined as nodes in the graph, and the definition of edges between nodes is introduced in Section 3.2.

3. Proposed Graph Optimization Model

3.1. Model Overview

Wi-Fi fingerprint positioning can provide position results at the current moment that are not dependent on the previous one, so there is no cumulative error as the user moves. However, coordinates obtained in this way are not accurate enough due to noise. To optimize the positioning results of a Wi-Fi fingerprint, physical distances between users measured by BLE ranging technology are added to establish redundant constraints. Then, graph optimization based on g^2o can be adopted to adjust the coordinates of users in the graph and improve positioning accuracy. The construction of the model is shown in Figure 3.

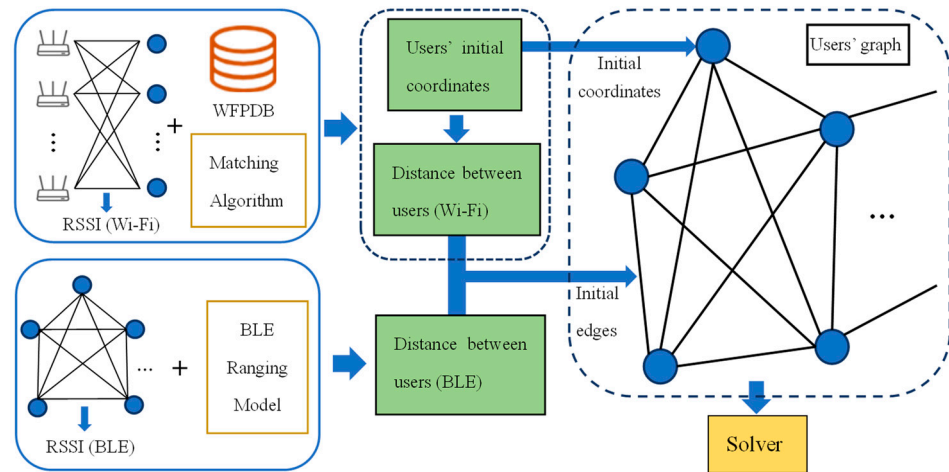


Figure 3. Construction of the proposed model.

We define the users as nodes of the graph and obtain their initial positioning coordinates by the construction of an offline Wi-Fi fingerprint database and online fingerprint matching. Users receive BLE RSSIs from other users' devices and input them to the BLE ranging model to obtain physical distances to other users. We define the error function and regard it as the edges between pairs of nodes. The total error function is constructed by all edges and grows in size with the number of users. The optimal solution of the total error function is obtained by the solver, i.e., the LM algorithm and the drifting solution introduced in Section 3.3.

There will inevitably be several edges with large relative errors in the graph that affect optimization. The robust kernel function, the method of reconstructing the information matrix introduced in Section 3.2, and the filtering conditions of the edges determined by the experiments in Section 4.2 are introduced to weaken the effects of these edges.

When we experimented with the proposed model, we used a laptop (Lenovo, Beijing, China) running Windows 10 as the central server for user positioning calculation. The users need to provide the RSSIs received from APs to the central server, and the initial coordinates of the users can be obtained based on the Wi-Fi fingerprint positioning. The RSSIs received from other users' devices with BLE function also need to be input into server to run the graph optimization model, and users can obtain the optimized position from the server.

3.2. Construction and Solution of Total Error Function

Constructing a reasonable total error function is the key to solving the graph optimization problem. As the number of user nodes in the graph increases, the number of variables (x_i and y_i) in the total error function expand twice as fast as nodes. We find that the number of nodes grows linearly, while the number of edges grows exponentially in the graph as the number of users joining the positioning increases. Because the total error function can easily fall into a locally optimal solution during iteration, assigning each node a good initial value can effectively solve the above problems, that is, obtaining the user's initial coordinates through the Wi-Fi fingerprint positioning. The graph, nodes, and edges are defined as follows.

Suppose there are n user nodes in the graph, user node i is defined as $p_i = (x_i, y_i)$, ($1 \leq i \leq n$), and its initial value as obtained by Wi-Fi fingerprint positioning is denoted as $p_i^w = (x_n^w, y_n^w)$, ($1 \leq i \leq n$). The distance obtained by the BLE ranging model between nodes i and j is defined as $d_{(i,j)}^b$, ($1 \leq i, j \leq n$, $i \neq j$), and the distance between nodes i and j in the graph is defined as

$$d_{(i,j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, (i \neq j). \quad (6)$$

Since the distance between users measured by the BLE ranging model adds constraints on nodes, in our proposed model, $d_{(i,j)}$ will be close to $d_{(i,j)}^b$ by adjusting the locations of nodes. So, the edge between p_i and p_j is defined as

$$e(p_i, p_j) = d_{(i,j)}^b - d_{(i,j)}, (1 \leq i, j \leq n), (i \neq j). \quad (7)$$

The total error function of the entire graph is composed of all of the error functions (called edges) between pairs of nodes. It can be defined as

$$F(p) = \sum_{\langle p_i, p_j \rangle \in p} e^T(p_i, p_j) \Omega_{(i,j)} e(p_i, p_j), \quad (8)$$

$$p^* = \operatorname{argmin} F(p), \quad (9)$$

where $\Omega_{(i,j)}$ is the information matrix of $e(p_i, p_j)$, and $p = \{p_1, p_2, \dots, p_n\}$ and $p^* = \{p_1^*, p_2^*, \dots, p_n^*\}$ are respectively a set of user nodes before and after optimization. $p_i^* = (x_i^*, y_i^*)$, $(1 \leq i \leq n)$ is the final position of p_i after optimization, because p_i keeps changing during the iterative solution of the total error function.

After defining the nodes and edges, we can define the size of the information matrix as 1×1 , which can be regarded as a weighting coefficient. The distance error between nodes based on the Wi-Fi fingerprint positioning cannot be predicted accurately; so, we do not consider adding the statistical indicators of Wi-Fi fingerprint positioning to calculate the information matrix.

Previous experiments with the BLE ranging model have shown that the RSSI of BLE can be approximated to a Gaussian distribution for simplicity [31]. As the distance increases, the BLE RSSI fluctuation increases, which directly leads to an increase in the fluctuation of the ranging error. The relationship between true distance and BLE ranging distance can be defined as

$$d_{(i,j)}^b = d_{(i,j)}^r + l_{(i,j)}, \quad (10)$$

where $d_{(i,j)}^r$ is the real distance between p_i and p_j , and $l_{(i,j)}$ is the error between $d_{(i,j)}^b$ and $d_{(i,j)}^r$. We use $d_{(i,j)}^b$ as a substitute for $d_{(i,j)}^r$ because $d_{(i,j)}^r$ cannot be predicted.

The experiments of Section 4.2 show that the distance error of BLE ranging is not only related to the physical distance, but is also closely related to the standard deviation (SD) of the ranging results. Thus, the SD of BLE ranging is added to calculate the information matrix:

$$\Omega_{(i,j)} = \frac{1}{\sigma_{(i,j)}^2 + 1}. \quad (11)$$

For simplicity of notation, in the rest of this paper, we will express the error function in a new form:

$$e(p_i, p_j) \stackrel{\text{def}}{=} e_{(i,j)}(p). \quad (12)$$

The error function can be approximated by its first-order Taylor expansion around the current initial guess $\hat{p}$:

$$\begin{aligned} e(\hat{p}_i + \Delta p_i, \hat{p}_j + \Delta p_j) &= e_{(i,j)}(\hat{p} + \Delta p) \\ &\cong e_{(i,j)}(\hat{p}) + J_{(i,j)} \Delta p, \end{aligned} \quad (13)$$

where $J_{(i,j)}$ is the Jacobian of $e_{(i,j)}(p)$, computed from the current estimate $\hat{p}$. Since there are errors in both BLE ranging and Wi-Fi fingerprint positioning, the graph will contain wrong edges in many cases. Nonlinear least-squares optimization is greatly affected by errors; thus, it is difficult to guarantee the convergence and performance of the total error function in the iterative process, even if the information matrix has been introduced to weaken the effect of errors. To reduce the occurrence of the above situation, the Huber kernel function [32] is introduced to adjust the edges.

When the error is small, the error function is approximated as a quadratic function with faster growth. When the error is large, it is approximated as a linear function with slower growth, which still ensures that the function is continuous and differentiable. The Huber kernel function is

$$\rho_H(x) = \begin{cases} \frac{1}{2}x^2, & |x| < \delta \\ \delta|x| - \frac{1}{2}\delta^2, & |x| \geq \delta \end{cases} \quad (14)$$

and the total error function with the Huber kernel function can be rewritten as

$$F(p) = \sum_{1 \leq i,j \leq n} \rho_H(e_{(i,j)}(p)) \Omega_{(i,j)}. \quad (15)$$

To preserve the characteristics of the original total error function and the calculated parameters, we can simplify the processing of the Huber kernel function by adjusting the information matrix,

$$\rho_H(e_{(i,j)}(p)) \Omega_{(i,j)} = e_{(i,j)}^T(p) \Omega_{(i,j)}^U e_{(i,j)}(p), \quad (16)$$

where $\Omega_{(i,j)}^U$ is the updated information matrix. The dimension of $e_{(i,j)}(p)$ is 1×1 ; thus,

$$\rho_H(e_{(i,j)}(p)) \Omega_{(i,j)} = (e_{(i,j)}(p))^2 \Omega_{(i,j)}^U, \quad (17)$$

$$\Omega_{(i,j)}^U = \frac{\rho_H(e_{(i,j)}(p)) \Omega_{(i,j)}}{(e_{(i,j)}(p))^2}. \quad (18)$$

Substituting the updated information matrix in the original total error function, we obtain a new total error function with the original form:

$$F_H(p) = \sum_{1 \leq i,j \leq n} e_{(i,j)}^T(p) \Omega_{(i,j)}^U e_{(i,j)}(p). \quad (19)$$

Substituting Equation (13) in Equation (19), $F_H(\hat{p} + \Delta p)$ can be approximated as

$$\begin{aligned} F_H(\hat{p} + \Delta p) &\cong \sum_{1 \leq i,j \leq n} (e_{(i,j)}(\hat{p}) + J_{(i,j)} \Delta p)^T \Omega_{(i,j)}^U (e_{(i,j)}(\hat{p}) + J_{(i,j)} \Delta p) \\ &= \sum_{1 \leq i,j \leq n} e_{(i,j)}^T(\hat{p}) \Omega_{(i,j)}^U e_{(i,j)}(\hat{p}) + 2e_{(i,j)}^T(\hat{p}) \Omega_{(i,j)}^U J_{(i,j)} \Delta p \\ &\quad + \Delta p^T J_{(i,j)}^T \Omega_{(i,j)}^U J_{(i,j)} \Delta p \\ &= \sum_{1 \leq i,j \leq n} c_{(i,j)} + 2b_{(i,j)} \Delta p + \Delta p^T H_{(i,j)} \Delta p \\ &= c + 2b \Delta p + \Delta p^T H \Delta p. \end{aligned} \quad (20)$$

For simplicity of notation, we set $c_{(i,j)} = e_{(i,j)}^T(\hat{p}) \Omega_{(i,j)}^U e_{(i,j)}(\hat{p})$, $b_{(i,j)} = e_{(i,j)}^T(\hat{p}) \Omega_{(i,j)}^U J_{(i,j)}$, $H_{(i,j)} = J_{(i,j)}^T \Omega_{(i,j)}^U J_{(i,j)}$, $c = \sum c_{(i,j)}$, $b = \sum b_{(i,j)}$, and $H = \sum H_{(i,j)}$. $F_H(\hat{p} + \Delta p)$ can be minimized in Δp by solving the linear system based on the LM algorithm,

$$(H + \alpha I) \Delta p^* = -b, \quad (21)$$

where H is the information matrix of this system, I is an identity matrix, and α is a damping factor. This is useful to control the step size in case of nonlinear surfaces and to control the convergence.

The solution is obtained by adding the increments Δp^* to the initial guess:

$$p^* = \hat{p} + \Delta p^*. \quad (22)$$

The popular LM algorithm iterates the linearization in Equation (20), the solution in Equation (21), and the update step in Equation (22). In every iteration, the previous solution is used as the linearization point and the initial guess until a given termination criterion is met. The derivation of the LM algorithm is not the work of this paper and is not repeated here. The implementation process of the LM algorithm is shown in Algorithm 1.

Algorithm 1: Pseudo code of LM algorithm

Input: A total error function $F_H(p)$ and an initial vector p^w

Output: A vector p^* minimizing $F_H(p)$

Begin

$p_{old} = p^w; v = 2;$

$\alpha = 10^{-5} * \max_i \{H_{ii}^{(0)}\};$ // $H^{(0)}$ is the first H in the iteration

While (! Stop Conditions)

Solve $(H + \alpha I)\Delta p^* = -b;$

$p_{new} = p_{old} + \Delta p^*;$

$\psi = \frac{F_H(p_{old}) - F_H(p_{new})}{\frac{1}{2}(\Delta p^*)^T (\alpha \Delta p^* - b)};$

If ($\psi > 0$)

Update H and $b;$

$\alpha = \alpha * \max\{0.5, 1 - (2\psi - 1)^3\}; v = 2;$

Else

$\alpha = \alpha * v; v = 2 * v;$

End

3.3. Drifting Solution Based on Affine Transformation Estimation

The optimized graph has a better structure, but the nodes may drift together because of the missing absolute coordinate constraints. Since the affine transformation estimation can relatively better preserve the optimized structure, we use it to solve the drifting problem. We define the nodes calculated by the LM algorithm as $p^l = (p_1^l, p_2^l, \dots, p_n^l)$ and the nodes estimated by affine transformation as $p^t = (p_1^t, p_2^t, \dots, p_n^t)$. We redefine the coordinates of each node as $p_i^l = (x_i^l, y_i^l, 1)$, ($1 \leq i \leq n$), $p_i^t = (x_i^t, y_i^t, 1)$, ($1 \leq i \leq n$) and $p_i^w = (x_i^w, y_i^w, 1)$, ($1 \leq i \leq n$). The transformation matrix is calculated by

$$\begin{cases} p^l M = p^w \\ M = \begin{pmatrix} a, d, 0 \\ b, e, 0 \\ c, f, 1 \end{pmatrix} \end{cases}, \quad (23)$$

where M represents the transformation matrix. The optimal estimates of parameters a, b, c, d, e , and f in M are calculated by the least-squares method. Our final optimization results of the nodes are

$$p^t = p^l M. \quad (24)$$

3.4. BLE Signal Filter and Ranging Model

During actual signal collection, the received RSSI from the BLE beacon fluctuates in the indoor environment. BLE ranging can affect the accuracy of the edges in the graph; thus, BLE ranging errors should be controlled within a certain range to build the network structure of nodes in the graph. In our experiments, the transmit power of the BLE beacon was +4 dBm, and its broadcast period was set to 80 ms. Figure 4a shows the RSSI data

collected at 3, 6, and 9 m between the BLE beacon and the signal receiving point, and it can be observed that the original RSSI fluctuates greatly.

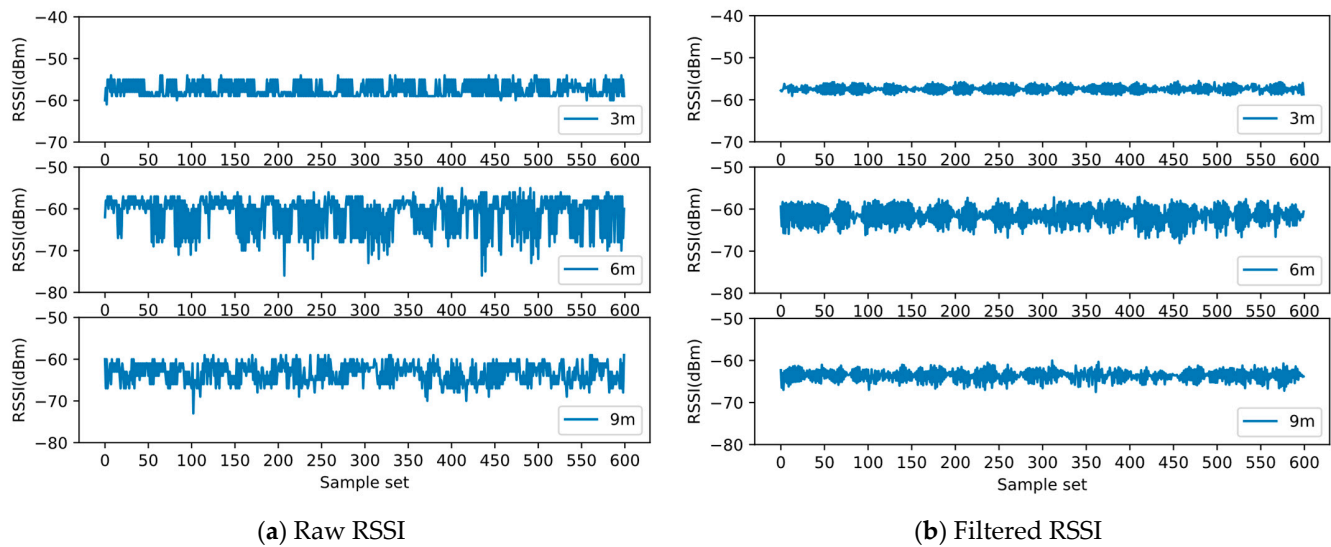


Figure 4. Raw and filtered received signal strength indication (RSSI) of Bluetooth Low Energy (BLE) at 3, 6, and 9 m.

Since the broadcast period of the BLE beacon was set to 80 ms, the average number of RSSIs that could be collected every second was 12.5. To balance the positioning time and a reasonable filter window size, we adopted mean filtering and set the filter window to 10, that is, 10 raw BLE RSSIs were processed each time before input to the BLE ranging model. Obviously, this strategy can effectively reduce the signal fluctuation, as shown in Figure 4b.

Table 1 presents the comparison of the SD of RSSIs collected at 3, 6, and 9 m before and after filtering. The SD can be reduced by nearly 50% after filtering.

Table 1. Standard deviation (SD) of received signal strength indication (RSSI) at 3, 6, and 9 m before and after filtering.

Distance (m)	SD (Original RSSI) (dBm)	SD (Filtered RSSI) (dBm)
3	1.802	0.935
6	4.553	2.594
9	2.542	1.498

This paper uses the simplified BLE ranging model (Equation (4)) and estimates the values of A and n by the filtered RSSI at different distances (0.25 m interval) and least-square curve fitting, as shown in Figure 5.

Then, we can calculate the distance between user nodes based on this mode,

$$\text{RSSI} = -45.688 - 20.835 \lg(d). \quad (25)$$

Through the linear fitting to the SD of the RSSI at different distances, we can obtain the signal propagation model applicable to our experiments,

$$\begin{cases} \text{RSSI} = -45.688 - 20.835 \lg(d) + \eta_{(i,j)} \\ \eta_{(i,j)} \sim N(0.24d_{(i,j)}^r + 1.51) \end{cases}. \quad (26)$$

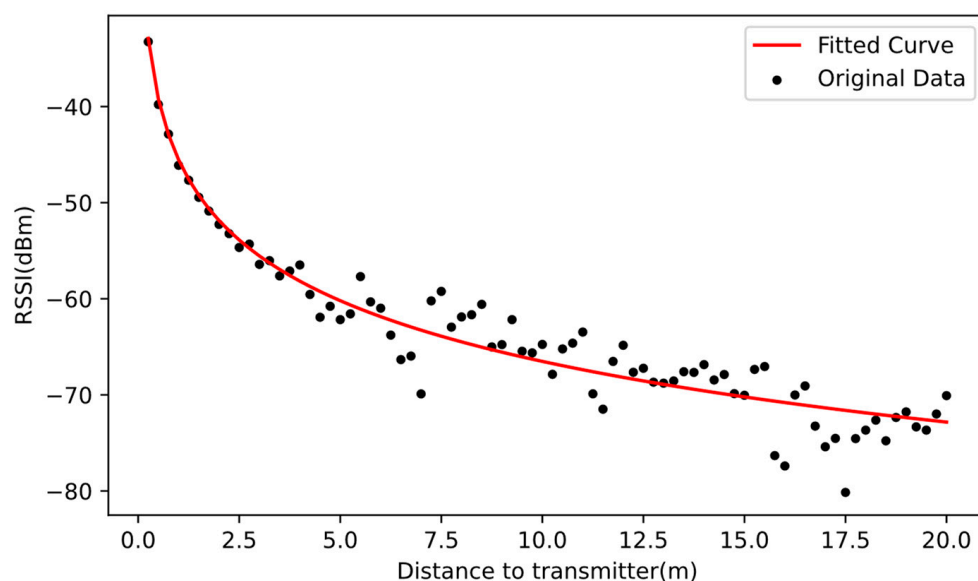


Figure 5. Fitted curve of BLE ranging model.

4. Experiments

4.1. Experimental Environment

Our experiments used the underground parking lot of North China Electric Power University as a testbed, where the Wi-Fi fingerprint database was constructed. The testbed was 58 m long and 42 m wide, and Wi-Fi APs were installed on the wall 2 m high relative to the parking lot surface. Figure 6a is a plan view of the testbed, where black dots mark the 10 AP locations. Test points (TPs) represent users, and RPs and TPs were set in the middle of the passage. The distance between two adjacent RPs was 2 m, and adjacent TPs were 1 m apart. We deployed 86 RPs and 84 TPs, where an “x” represents an RP and a four-color “o” represents a TP. Figure 6b shows the coordinates of RPs and TPs.

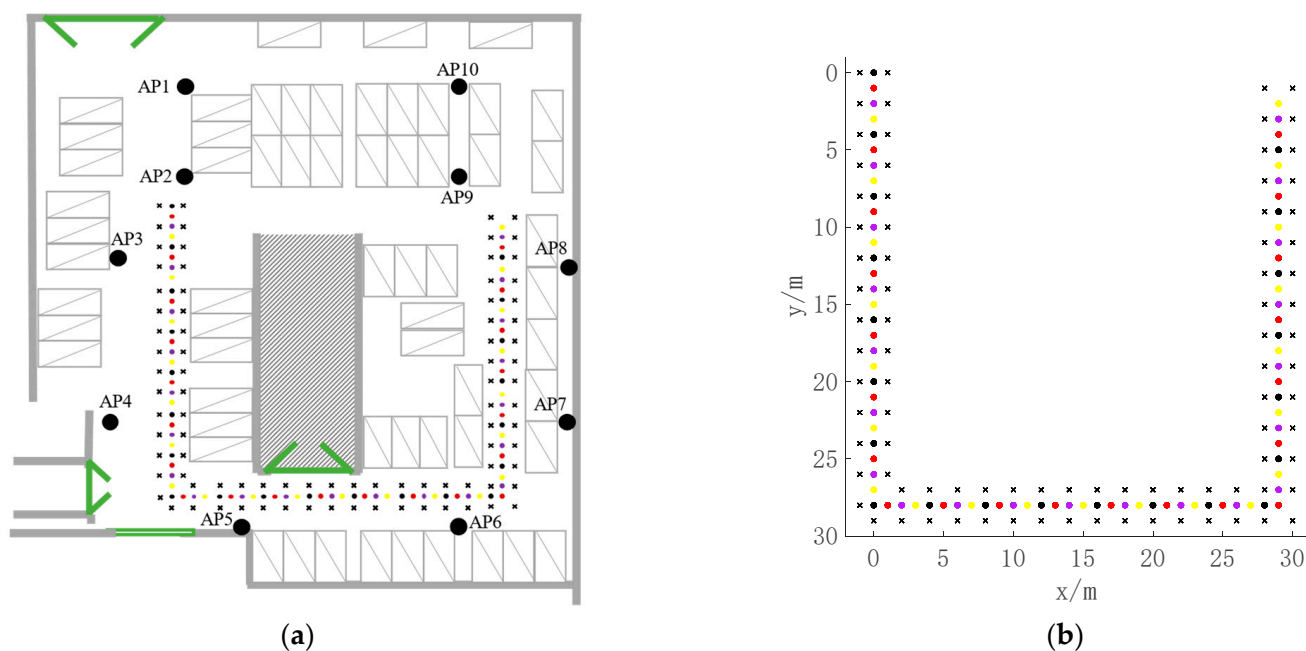


Figure 6. (a) Testbed plan; (b) coordinates of TPs and RPs.

Our device was set 1 m high relative to the parking lot surface, collecting Wi-Fi data at each RP. We collected 10 sets of Wi-Fi data of 10 APs continuously at each RP to construct

the Wi-Fi fingerprint database, and we used the Wi-Fi fingerprint-matching algorithm to obtain the initial location of each TP. To verify the validity of the proposed model and control the cost, we conducted experiments with simulated BLE ranging data and randomly selected points from 84 TPs to perform the graph optimization.

4.2. Edge Selection

Edges with larger errors in the graph are often formed by two user nodes that are far apart. Giving these edges a reasonable information matrix can weaken the influence of errors. However, when we calculate the information matrix $\Omega_{(i,j)}$ (Equation (11)), we need to determine the value of $\sigma_{(i,j)}$. We collected the BLE RSSIs every 0.5 m within a range of 20 m from the BLE beacon and input them to the BLE ranging model (Equation (25)) to calculate the error and SD of ranging in each collection point, as shown in Figure 7.

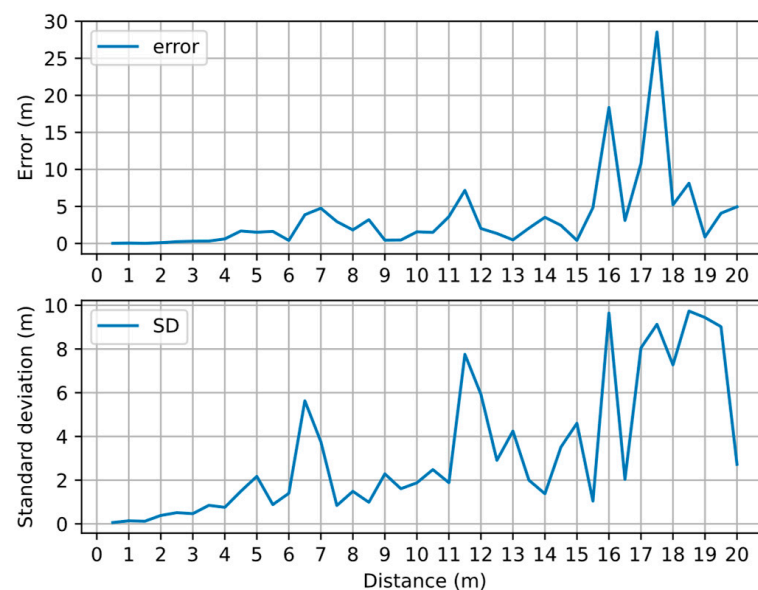


Figure 7. Error and standard deviation (SD) of BLE ranging at different distances.

As the distance increases, the error and SD of ranging generally show a linear upward trend. The greater the distance, the more frequent the fluctuations of the error and SD. Experiments show that most BLE ranging errors within 15 m can be stably concentrated within 5 m, and more are concentrated within 2 m. Table 2 shows the ranging error in different distances within 15 m at 1 m intervals. In the experiment, we collected RSSIs at different distances and calculated the physical distance between the transmitting point and the receiving point, but the results show the discontinuity of the error. This is mainly due to the fact that the influence of the multipath effect on the ranging error is more significant than that of signal attenuation in the range of 20 m. Considering the relationship between ranging error and distance, we decided to only add edges within 15 m to the graph.

Table 2. Ranging error within 15 m at 1 m intervals.

Distance (m)	Error (m)	Distance (m)	Error (m)
1	0.058	9	0.448
2	0.104	10	1.573
3	0.308	11	3.623
4	0.615	12	2.02
5	1.518	13	0.487
6	0.413	14	3.538
7	4.761	15	0.419
8	1.829		

4.3. Influence of Threshold δ in Huber Kernel Function

The Huber kernel function can reduce the effect of large error edges in the graph. If the value of δ is set too small, the weight of most edges will be reduced, which will weaken the role of precise edges. If it is set too large, the weight of large error edges will not be reduced, which will cause the Huber kernel function to lose its effect. So, the δ value should be chosen appropriately.

The experimental design of this part was as follows. We obtained the initial coordinates of 84 TPs by KNN matching algorithm and set the Huber kernel function threshold δ from 0.5 to 3.5. Seven control groups were set up with 7, 9, 11, 13, 15, 17, and 19 nodes per group. Each group randomly selected a corresponding number of points from 84 TPs for optimization and repeated the experiment 1000 times to obtain sufficient data. We also added a comparison experiment without the Huber kernel function in each group. The results in Figure 8 show that the mean positioning error and 75th percentile error were reduced by adjusting the value of threshold δ from 0.5 to 2. Adjusting the value of threshold δ from 2 to 3.5, the mean positioning error and 75th percentile error increased. Obviously, setting δ to 2 is reasonable.

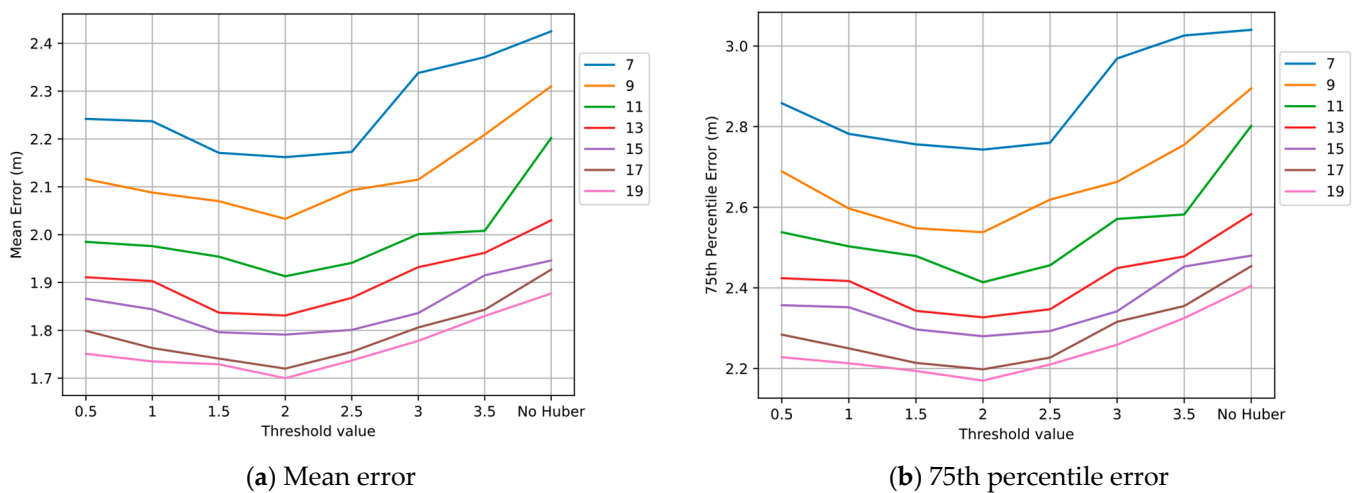


Figure 8. Positioning errors under different thresholds δ .

4.4. Impact of Number of Nodes in Graph

In the experiment, we obtained the initial coordinates of 84 TPs based on the KNN algorithm and set the Huber kernel function threshold δ to 2. Nine control groups were set up with 7, 9, 11, 13, 15, 17, 19, 21, and 23 nodes per group. Each group randomly selected a corresponding number of points from 84 TPs for optimization and repeated the experiment 1000 times to obtain sufficient positioning coordinates. Figure 9 shows that the optimization effect improves as the number of nodes added to the graph for optimization increases from 7 to 23, and the magnitude of the improvement gradually decreases and the error curve tends to be steady.

When the number of nodes reaches 19, the reduction of the mean error and 75th percentile error is less than 2 cm for every two additional nodes. When the number of nodes is 7, the average distance between nodes is often greater than the threshold of 15 m, and too few edges can be added to the graph. Therefore, insufficient constraints and large errors will cause the failure of the total error function to converge and an improvement of positioning accuracy that is not obvious. When the number of nodes reaches 9, the total error function can converge in most cases, further improving the positioning accuracy. The experimental results show that under the KNN algorithm, we can obtain relatively stable positioning results when the number of nodes reaches 19 and δ is set to 2. Figure 10 shows the error cumulative distribution function (CDF) curves of different numbers of nodes.

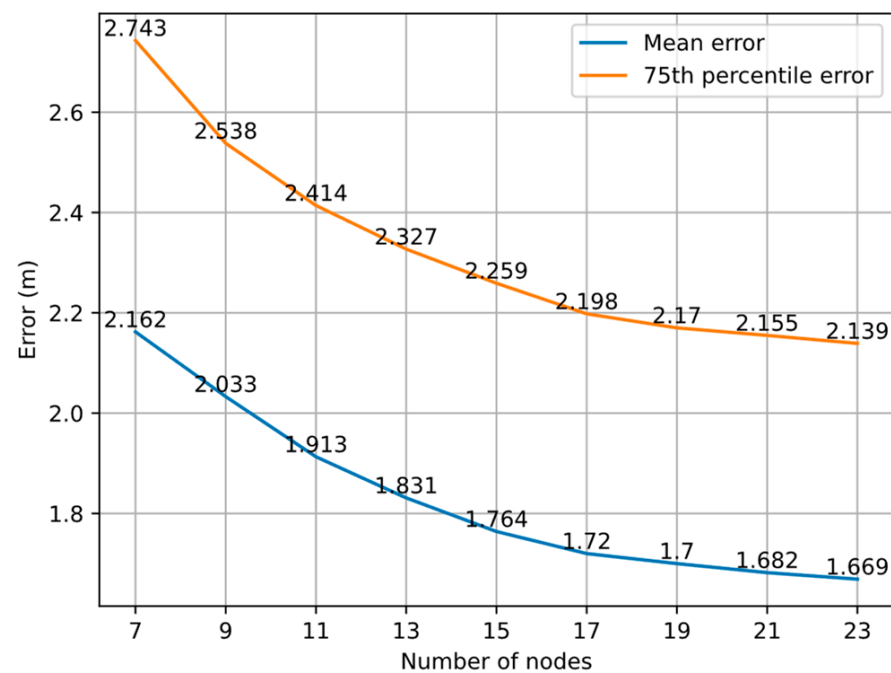


Figure 9. Positioning errors under different numbers of nodes.

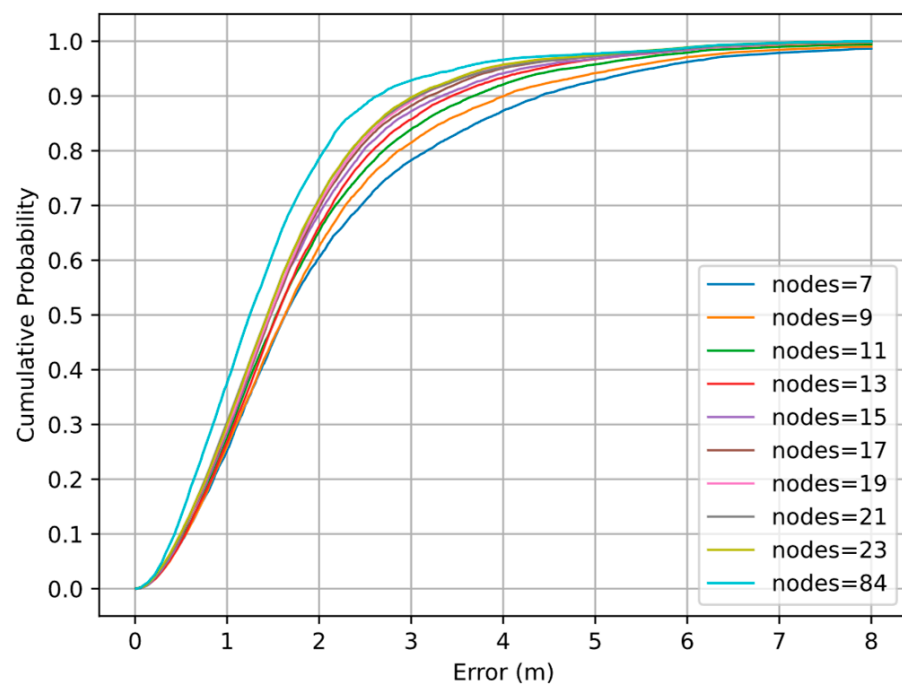


Figure 10. Positioning error cumulative distribution function (CDF) curves of different numbers of nodes.

It can be observed that the spacing between the curves decreases as the number of nodes increases, and the curves with 19, 21, and 23 nodes are tightly packed together. The gap between the curves is especially obvious at 2–5 m, indicating that our graph optimization model can significantly improve the positioning accuracy at 2–5 m as the number of nodes increases.

4.5. Impact of Distribution of Nodes in Graph

The 84 TPs were labeled in U-shaped order with markers ranging from 1 to 84, which were sequentially divided into three groups. Each group contained 28 TPs, representing

one of the three passages in the experimental area. We defined a good distribution as a random selection of nodes from only one of the three groups for graph optimization, and a bad distribution as a random and even selection of nodes from each of the three groups for graph optimization. In the experiment, we set the Huber kernel function threshold δ to 2 and separately conducted group experiments for two distributions. The number of nodes in each group was different and the experiment was repeated 1000 times per group.

Figure 11a shows the positioning errors under different numbers of nodes in good distribution. It can be found that the mean errors and 75th percentile errors of the good distribution are lower than the random distribution (as shown in Figure 9) under the same number of nodes. The reduction of the mean error is less than 2 cm for every two additional nodes in 15 nodes, and the reduction of 75th percentile error is less than 2 cm for every two additional nodes in 17 nodes. Figure 11b shows the positioning errors under different numbers of nodes in bad distribution. In addition to the fact that the errors are higher than the random distribution, the reduction of the mean error and 75th percentile error is less than 2 cm for every two additional nodes in 23 nodes. In the bad distribution, the positioning accuracy decreases compared with the KNN algorithm when the number of nodes is 7, and the positioning error is only slightly lower than the KNN algorithm when the number of nodes reaches 9, which indicates that the number of nodes in our proposed model needs to be greater than or equal to 9 for better work in our experimental area.

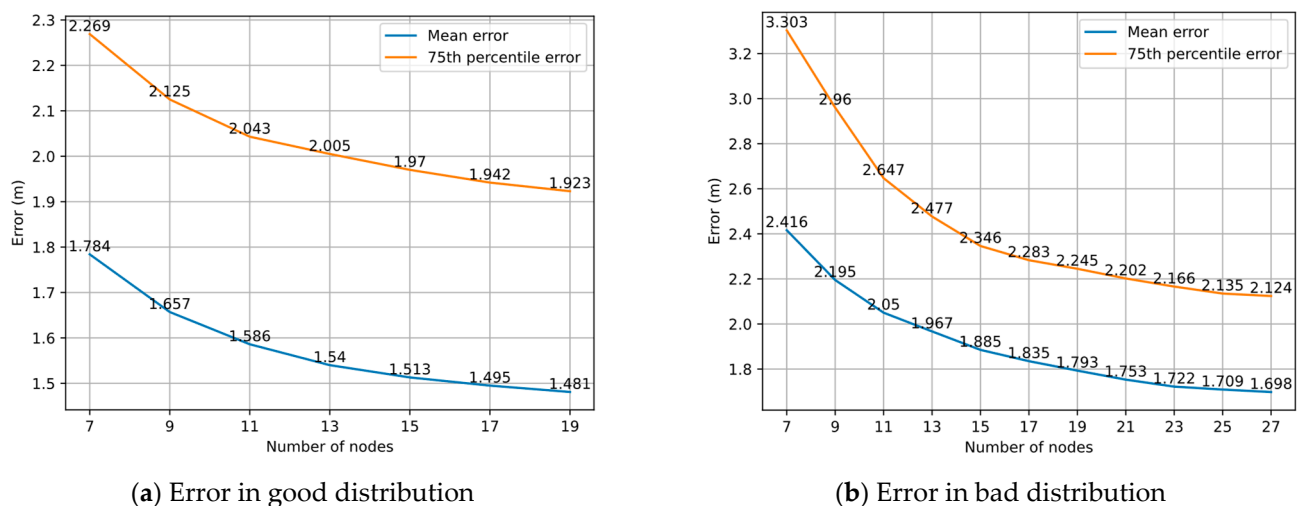


Figure 11. Positioning errors under different numbers of nodes in two types of distribution.

The experimental results show that the error tends to level off with the node increase under both good and bad distributions, and the errors of a good distribution tend to level off faster than those of a bad distribution. This shows that our graph optimization model is influenced by the node distribution. We find that the better the node distribution, the higher the positioning accuracy for the same number of nodes. The positioning accuracy of a good distribution is more likely to achieve relatively stable optimization results.

4.6. Comparison of Wi-Fi Fingerprint-Matching Algorithms after Optimization

The proposed graph optimization model can be used to optimize Wi-Fi fingerprint-matching algorithms other than the KNN algorithm. In the experiments, we constructed the user graph with 19 random nodes and set the Huber kernel function threshold δ to 2 to optimize four Wi-Fi fingerprint-matching algorithms. The error CDF curves of the KNN, W-KNN, GK, and Stg algorithms before and after optimization are shown in Figure 12. Obviously, the proposed graph optimization model can improve the positioning accuracy. We find that the GK algorithm has the highest positioning accuracy, and the ranking of positioning effects before and after optimization has not changed.

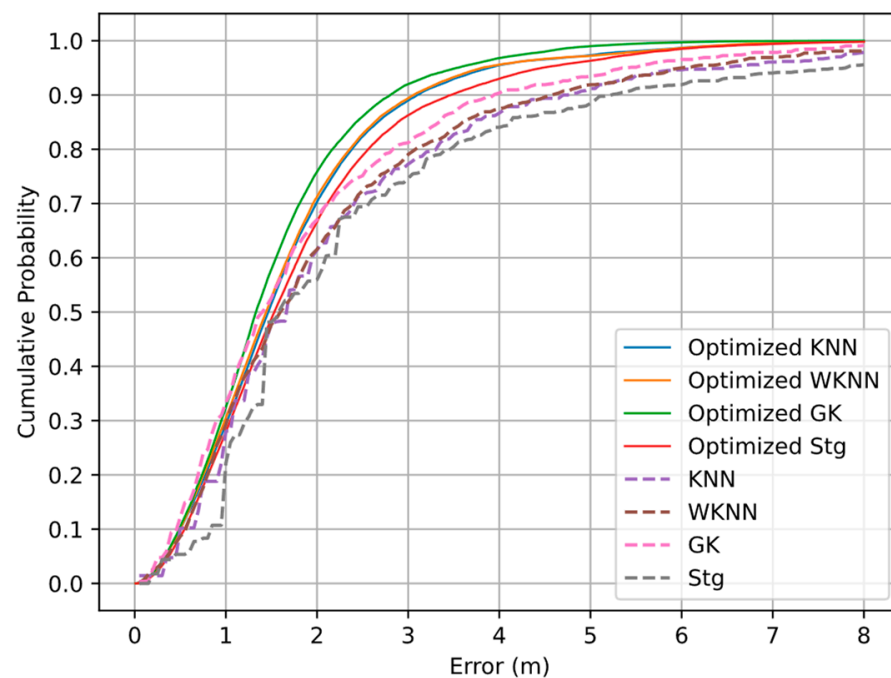


Figure 12. Positioning error CDF curves of Wi-Fi fingerprint-matching algorithms before and after optimization.

The cumulative error probability of different algorithms under a fixed accuracy limit is shown in Table 3. It can be seen that the optimized algorithm (KNN, WKNN, and GK) does not dominate within 1 m, but the positioning effect is gradually improved from 1 m to 3 m, being most obvious at 3 m. This shows that our proposed algorithm can effectively correct node coordinates with large errors. The optimized Stg algorithm is slightly different at 1 m and 1.5 m, which is caused by the uneven error distribution of the Stg algorithm in the range of 0–2 m, as shown in Figure 12. The obvious optimized results at 2 m, 2.5 m and 3 m of the Stg algorithm again validated our conclusion.

Table 3. Cumulative error probability of different algorithms under fixed accuracy limit.

Algorithm	1 m	1.5 m	2 m	2.5 m	3 m
Optimized KNN	29.34%	51.73%	70.24%	82.35%	89.02%
Optimized WKNN	30.47%	52.72%	71.27%	82.79%	89.44%
Optimized GK	33.41%	57.88%	75.9%	86.04%	92.1%
Optimized Stg	27.93%	48.86%	66.59%	78.79%	86.26%
KNN	28.1%	47.86%	61.43%	71.55%	77.26%
WKNN	29.29%	47.74%	61.55%	72.38%	79.05%
GK	33.33%	52.62%	67.02%	75.12%	81.19%
Stg	22.14%	48.21%	55.95%	69.4%	74.64%

Table 4 shows the optimization results of four Wi-Fi fingerprint-matching algorithms. In terms of the mean positioning error, the KNN algorithm improves the positioning accuracy by 23.1%, WKNN by 21.5%, GK by 19.4%, and Stg by 27.2%. The results show that the 75th percentile error and SD are also reduced after optimization, which shows the effectiveness of mixed BLE ranging and Wi-Fi fingerprint positioning based on the graph optimization model at improving the positioning accuracy. We find that the lower the initial localization accuracy, the more obvious the improvement of localization accuracy, which indicates that the proposed model works better for Wi-Fi matching algorithms with larger errors.

Table 4. Error of optimized Wi-Fi fingerprint-matching algorithms.

Indicator	Algorithm	Optimized KNN	Optimized WKNN	Optimized GK	Optimized Stg	KNN	WKNN	GK	Stg
Mean Error (m)		1.7	1.68	1.54	1.82	2.21	2.14	1.91	2.5
75th Percentile Error (m)		2.17	2.14	1.97	2.32	2.75	2.69	2.50	3.01
Error Std (m)		1.19	1.18	1.01	1.3	1.92	1.85	1.67	2.33

From the experimental results, it can be seen that the more accurate the initial positioning results output by Wi-Fi positioning, the more accurate the optimized positioning results, which indicates that the positioning results of our graph optimization model are sensitive to the initial positioning.

5. Conclusions and Further Work

We integrated Wi-Fi fingerprint positioning and BLE ranging technology to locate users in indoor environments based on a graph optimization model. After deriving the relationship between the physical distance and the RSSI of BLE, a BLE ranging simulation was carried out based on the actual RSSI standard deviation. We obtained initial positioning results through Wi-Fi fingerprint positioning, built a graph of users, added edges between user nodes based on BLE ranging data, and summarized them in a large-scale total error function, which we solved to obtain a better structure of nodes. A drifting solution based on the affine transformation was applied to obtain the final optimized position of the users. The experiments showed that the proposed graph optimization model has a relatively good positioning result when the kernel function threshold δ was set to 2. The reduction of the average positioning error and 75th percentile error was less than 2 cm for every two additional nodes when the number of nodes in the graph reached 19. Our graph optimization model can be influenced by the node distribution. The positioning accuracy of a good distribution is more likely to achieve relatively stable and accurate optimization results. The application of the proposed model to KNN, WKNN, GK, and Stg resulted in improvements, which shows that the proposed model is versatile. The positioning results of the four optimized Wi-Fi fingerprint-matching algorithms showed that the initial positioning results can affect the optimized positioning results.

Although the proposed model can improve the positioning accuracy of traditional Wi-Fi fingerprint positioning, there is room for improvement. The experiments showed that the positioning accuracy rankings of other Wi-Fi algorithms before and after optimization are basically the same, and they verified the effect of the initial positioning results on the optimization results. In addition, we hypothesize that the accuracy of ranging between the user nodes in the graph can affect the final positioning results. Therefore, our next step will be to go beyond the use of BLE beacons for ranging. For example, wireless sensors with higher accuracy can be used to construct more reliable edges of the user graph.

Author Contributions: Conceptualization, R.Z. and J.T.; methodology, R.Z.; software, P.C. and F.M.; validation, P.C. and F.M.; formal analysis, J.T.; investigation, R.Z.; resources, R.Z.; data curation, P.C.; writing—original draft preparation, R.Z.; writing—review and editing, J.T. and P.C.; visualization, P.C.; supervision, R.Z.; project administration, R.Z.; funding acquisition, R.Z. All of the authors contributed to the interpretation and discussion of the results, as well as playing a role in drafting the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The experiment uses an internal data set. The data presented in this study are available on request from the corresponding author.

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

References

- Wei, X.; Wei, Z.; Radu, V. Sensor-Fusion for Smartphone Location Tracking Using Hybrid Multimodal Deep Neural Networks. *Sensors* **2021**, *21*, 7488. [\[CrossRef\]](#) [\[PubMed\]](#)
- Jo, H.J.; Kim, S. Indoor Smartphone Localization Based on LOS and NLOS Identification. *Sensors* **2018**, *18*, 3987. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zhou, R.; Yang, Y.; Chen, P. An RSS Transform-Based WKNN for Indoor Positioning. *Sensors* **2021**, *21*, 5685. [\[CrossRef\]](#)
- Ramirez, R.; Huang, C.-Y.; Liao, C.-A.; Lin, P.-T.; Lin, H.-W.; Liang, S.-H. A Practice of BLE RSSI Measurement for Indoor Positioning. *Sensors* **2021**, *21*, 5181. [\[CrossRef\]](#) [\[PubMed\]](#)
- Bottigliero, S.; Milanese, D.; Saccani, M.; Maggiora, R. A Low-Cost Indoor Real-Time Locating System Based on TDOA Estimation of UWB Pulse Sequences. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [\[CrossRef\]](#)
- Makki, A.; Siddig, A.; Bleakley, C.J. Robust High Resolution Time of Arrival Estimation for Indoor WLAN Ranging. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2703–2710. [\[CrossRef\]](#)
- Yang, T.; Cabani, A.; Chafouk, H. A Survey of Recent Indoor Localization Scenarios and Methodologies. *Sensors* **2021**, *21*, 8086. [\[CrossRef\]](#)
- Zhou, M.; Lin, Y.; Zhao, N.; Jiang, Q.; Yang, X.; Tian, Z. Indoor WLAN Intelligent Target Intrusion Sensing Using Ray-Aided Generative Adversarial Network. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *4*, 61–73. [\[CrossRef\]](#)
- Sugasaki, M.; Shimosaka, M. Robustifying Wi-Fi Localization by Between-Location Data Augmentation. *IEEE Sens. J.* **2022**, *22*, 5407–5416. [\[CrossRef\]](#)
- He, S.; Chan, S.G. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 466–490. [\[CrossRef\]](#)
- Zafari, F.; Papapanagiotou, I.; Christidis, K. Microlocation for Internet-of-Things-Equipped Smart Buildings. *IEEE Internet Things J.* **2016**, *3*, 96–112. [\[CrossRef\]](#)
- Takahashi, C.; Kondo, K. Indoor positioning method for augmented audio reality navigation systems using iBeacons. In Proceedings of the 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 27–30 October 2015; pp. 451–452.
- Liu, W.; Guo, W.; Zhu, X. Map-Aided Indoor Positioning Algorithm with Complex Deployed BLE Beacons. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 526. [\[CrossRef\]](#)
- Bahl, P.; Padmanabhan, V.N. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the IEEE INFOCOM 2000, Conference on Computer Communications, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Tel Aviv, Israel, 26–30 March 2000; Volume 772, pp. 775–784.
- Castro, P.; Chiu, P.; Kremenek, T.; Muntz, R. A Probabilistic Room Location Service for Wireless Networked Environments. In Proceedings of the Ubicomp 2001 Conference, Atlanta, GA, USA, 30 September–2 October 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 18–34.
- Youssef, M.; Agrawala, A. The Horus location determination system. *Wirel. Netw.* **2008**, *14*, 357–374. [\[CrossRef\]](#)
- Ferreira, D.; Souza, R.; Carvalho, C. QA-kNN: Indoor Localization Based on Quartile Analysis and the kNN Classifier for Wireless Networks. *Sensors* **2020**, *20*, 4714. [\[CrossRef\]](#) [\[PubMed\]](#)
- Siyang, L.I.U.; De Lacerda, R.; Fiorina, J. WKNN indoor Wi-Fi localization method using k-means clustering based radio mapping. In Proceedings of the 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 25–28 April 2021; pp. 1–5.
- Marques, N.; Meneses, F.; Moreira, A. Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning. In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia, 13–15 November 2012; pp. 1–9.
- Roos, T.; Myllymäki, P.; Tirri, H.; Misikangas, P.; Sievänen, J. A Probabilistic Approach to WLAN User Location Estimation. *Int. J. Wirel. Inf. Netw.* **2002**, *9*, 155–164. [\[CrossRef\]](#)
- Fang, S.; Lin, T. Indoor Location System Based on Discriminant-Adaptive Neural Network in IEEE 802.11 Environments. *IEEE Trans. Neural Netw.* **2008**, *19*, 1973–1978. [\[CrossRef\]](#)
- Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 763–776. [\[CrossRef\]](#)
- Golestani, A.; Petreska, N.; Wilfert, D.; Zimmer, C. Improving the precision of RSSI-based low-energy localization using path loss exponent estimation. In Proceedings of the 2014 11th Workshop on Positioning, Navigation and Communication (WPNC), Dresden, Germany, 12–13 March 2014; pp. 1–6.
- Nowak, T.; Hartmann, M.; Zech, T.; Thielecke, J. A path loss and fading model for RSSI-based localization in forested areas. In Proceedings of the 2016 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC), Cairns, Australia, 19–23 September 2016; pp. 110–113.
- Nguyen, H.A.; Guo, H.; Low, K. Real-Time Estimation of Sensor Node's Position Using Particle Swarm Optimization With Log-Barrier Constraint. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 3619–3628. [\[CrossRef\]](#)
- Lee, G.; Moon, B.-C.; Lee, S.; Han, D. Fusion of the SLAM with Wi-Fi-Based Positioning Methods for Mobile Robot-Based Learning Data Collection, Localization, and Tracking in Indoor Spaces. *Sensors* **2020**, *20*, 5182. [\[CrossRef\]](#)

27. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
28. Sun, C.Z.; Zhang, B.; Wang, J.K.; Zhang, C.S. A Review of Visual SLAM Based on Unmanned Systems. In Proceedings of the 2021 2nd International Conference on Artificial Intelligence and Education (ICAIE), Dali, China, 18–20 June 2021; pp. 226–234.
29. Lourakis, M.L.A.; Argyros, A.A. Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005; Volume 1, pp. 1526–1531.
30. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. G²o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3607–3613.
31. Jianyong, Z.; Haiyong, L.; Zili, C.; Zhaohui, L. RSSI based Bluetooth low energy indoor positioning. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 526–533.
32. Balasundaram, S.; Meena, Y. Robust Support Vector Regression in Primal with Asymmetric Huber Loss. *Neural Process. Lett.* **2019**, *49*, 1399–1431. [[CrossRef](#)]