

Article

Optimization-Based Wi-Fi Radio Map Construction for Indoor Positioning Using Only Smart Phones

Jian Tan ^{1,2,*}, Xiangtao Fan ^{1,2}, Shenghua Wang ³ and Yingchao Ren ²¹ Hainan Key Laboratory of Earth Observation, Sanya 572029, China; fanxiangtao_radi@163.com² Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China; ryc_radi@163.com³ School of Public Administration and Mass Media, Beijing Information Science and Technology University, Beijing 100093, China; wangshenghua_sn@163.com

* Correspondence: tanjian@radi.ac.cn

Received: 16 August 2018; Accepted: 11 September 2018; Published: 14 September 2018



Abstract: Fingerprinting-based Wi-Fi indoor positioning has great potential for positioning in GPS-denied areas. However, establishing a fingerprinting map (also called a radio map) prior to positioning (site survey) is normally a labor-intensive task. This paper proposes a method for easy site survey without need for any extra hardware. The user can conduct the site survey adopting only a smart phone. The collected inertial-based readings are processed using the pedestrian dead-reckoning algorithms to generate a raw trajectory. Then a factor graph optimization method is proposed to re-estimate the trajectory by adding constraints originated from collected Wi-Fi fingerprints and landmark positions. The proposed method is verified through an experiment in a mall. The mean positioning error is 1.10 m and the maximum error is 2.25 m. This level of positioning accuracy is considered sufficient for radio map generation purposes. A classical baseline algorithm, the k-Nearest Neighbor (kNN) algorithm, is adopted to test the positioning performance of the radio map (RM), which also validates the quality of the constructed RM from the proposed method.

Keywords: Wi-Fi indoor positioning; radio map; pedestrian dead reckoning; factor graph optimization

1. Introduction

Positioning in GPS-denied areas has been attracting great attention for decades and currently there are many solutions for that. Fingerprinting-based Wi-Fi positioning is one of these solutions with great potential for the following two reasons: (1) No extra hardware is needed. The users only need to carry Wi-Fi-enabled devices, such as smart phones, to locate themselves in indoor environments with abundant Wi-Fi signals. This feature renders the solution suitable for the consumer indoor positioning market as either Wi-Fi-enabled devices or Wi-Fi signals are prevalent in many indoor environments, such as malls and airports; (2) The positioning error is within limited bounds. Unlike inertial-based indoor positioning methods, such as [1,2], the positioning error will not accumulate with time.

The working flow of fingerprinting-based Wi-Fi indoor positioning has two phases: the offline phase (also denoted as site survey phase) and the online phase (also denoted as positioning phase). A fingerprint herein is defined as the received signal strength indication (RSSI) from different Wi-Fi access points (APs) at a fixed location in the environment. Normally, the fingerprint is a vector because there are RSSI from more than one AP. The site survey phase is essentially to associate the fingerprints to geometric positions in the environments. After the phase, each fingerprint corresponds to a position and thus a fingerprint radio map (RM) is established. As the RM is normally established offline, it is also called offline phase. The positioning phase is essentially positioning the user with a Wi-Fi-enabled device online. Each time a new fingerprint is collected by the device, it is matched

against the fingerprints from the RM and returns a position. The most commonly seen matching method for this is the k-nearest neighbor (kNN) method [3,4].

Despite the aforementioned two advantages (on the positioning phase), the site survey phase is challenging. Normally, it is recognized as slow and labor-intensive tasks because each position in the environment should be associated with a Wi-Fi signal fingerprint. There are many methods for boosting the efficiency of the site survey phase. The authors in [5] propose to establish the RM in a crowd-sourced way. Many users equipped with foot-mounted inertial measurement units (IMUs) walking in the environments while collecting Wi-Fi fingerprints. After an optimization-based offline process, the trajectories are aligned and calibrated. In this method, the goal is to perform more accurate inertial-based positioning and the RM is a byproduct. The authors in [6] put trajectories generated from a smart phone and some fingerprints at known positions (defined as reference points, RPs) into an optimization framework to establish the RM. The authors in [7,8] propose a quick radio fingerprint collection (QRFC) algorithm for sampling the Wi-Fi signals by clicking on the starting and ending points on the map of the indoor environment. Some methods [9–11] propose to sparsely collect fingerprints at some known positions and then reconstruct the RM. Herein many reconstruction methods are adopted, such as compressed sensing (CS) [9] theory, randomized least absolute shrinkage and selection operator (LASSO) [10] and Gaussian process (GP) regression [11]. Interpolation-based methods are also adopted for RM reconstruction purposes using some RPs. In [12], the authors adopt linear interpolation by using the minimum and mean value of the RSS at three RPs, while in [13] a weighted average is adopted. Although interpolation can reduce the number of RPs needed, the accuracy of interpolation-based methods rely on the granularity of existing RPs. The generated RM will not be accurate if the granularity of RPs is very low. Some other methods propose to generate the RM analytically. In [14,15], the authors adopt the existing RSS at the RPs to train the parameters of a pre-defined path loss model, which assumes a logarithmic decay from the distance to the APs, and can greatly reduce the number of RPs needed. However, to acquire the RM using the path loss model, the positions of the APs should be known, which is often not available. Especially in many large scale public sites such as malls and stations, where there can be more than a thousand of APs. Moreover, the trained path loss model may vary in different parts of the indoor environment and in different directions. The radiosity model was originally adopted in physics to describe the heat transfer in bodies with different temperatures [16]. It is adopted in [17] by solving for the Wi-Fi signal propagation problem in the presence of obstacles such as walls and doors. Since the RM is calculated analytically, no domain expert intervention is needed, which can greatly reduce the cost of site survey. However, the radiosity model need detailed indoor maps to begin with, including the position & materials of the walls, doors and so on. Ray tracing [18]-based methods also construct the RM analytically. In [19], ray tracing-based method along with the uniform theory of diffraction (UTD) is adopted to predict the RF propagation. The mentioned methods are effective in generating the RM. However, they either need extra hardware (e.g., foot-mounted IMU), extra information (e.g., detailed indoor maps and building materials) or extra surveying process (fingerprints at known positions).

This paper proposes a method for generating the RM for the site survey phase. As described in Figure 1, the inertial measurements from the smart phones are adopted to generate the trajectories using the pedestrian dead-reckoning (PDR) algorithm. The PDR algorithm mainly solves two problems: (1) how to estimate the phone orientation; (2) how to estimate the step length and detect step occurrence (stepometer). The trajectories from PDR, some landmark positions along with the collected fingerprints are adopted to form a factor graph. The factor graph is generally a graphic representation of an estimation problem for the trajectories. The inertial-generated trajectories can be regarded as an initial guess of the trajectories. By optimizing the graph, a raw RM is established with the estimated positions. After simple linear interpolation, the RM is ready for positioning purposes. Compared to the aforementioned RM generation methods, this method does not need any extra hardware, much additional information (e.g., detailed indoor maps) or extra surveying process.

The only extra information is some foreknown landmarks positions, which is commonly fulfilled because many indoor positioning applications needs some reference points to begin with.

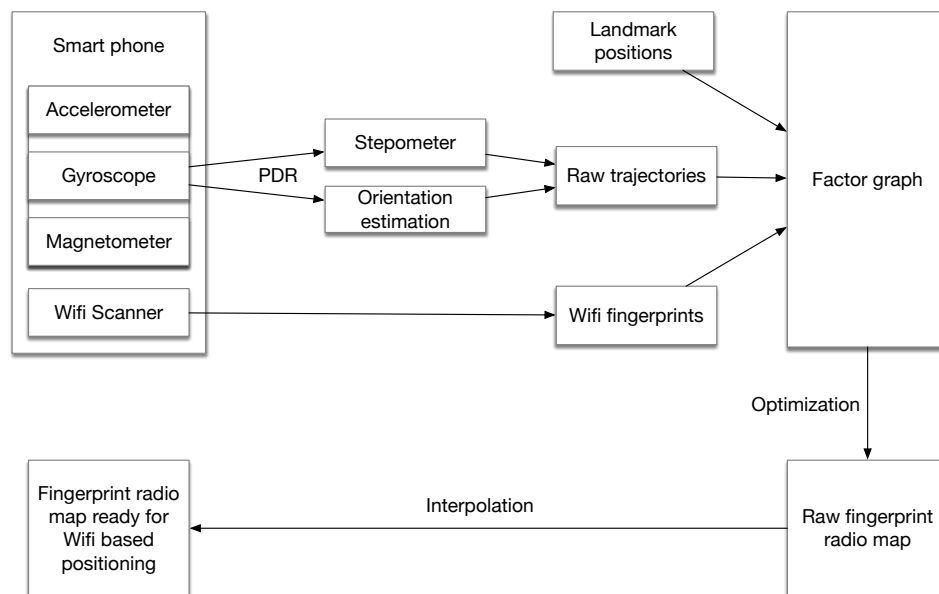


Figure 1. An overall processing flow for the proposed fingerprint radio map establishment.

Some related work including basics of PDR and factor graph estimation is given in Section 2. Section 3 describes the trajectory generation method based on PDR and Section 4 describes the RM generation method based on factor graph optimization. Real-scenario experiments are carried out in Section 5 to validate the proposed method. Section 6 is the conclusion.

2. Related Work

As described in Figure 1, our method relies on trajectories generated from the sensors on the smart phone. This is possible because currently most smart phones have embedded Micro-Electro-Mechanical System (MEMS) accelerometers, gyroscopes and magnetometers. Factor graph-based optimization is widely adopted in simultaneous localization and mapping (SLAM) for finding the poses which best satisfies the measurements. In this section, we give a brief overview of the PDR algorithm and factor graph-based optimization.

2.1. Pedestrian Dead-Reckoning

The PDR algorithm consists of two aspects: orientation estimation and stepometer estimation [20]. Herein stepometer estimation includes step occurrence detection and step length estimation. For orientation estimation, the simplest form of the PDR algorithm is to use a simple Kalman filter for updating the orientation [21]. Classical step occurrence detection methods include peak detection [22] and cross-zero detection [23]. These methods assume that the peaks of the projected acceleration or the “cross-zero” points corresponds to new steps. Classical step length estimation method adopts a parametric model [24] and the parameter differs across persons. Noting that the PDR algorithm here avoids double integration of the accelerations to reduce quicker accumulation of acceleration inertial drifts. However, as both step occurrence detection and step length estimation are based on heuristic assumptions, additional positioning errors are introduced, e.g., inaccurate step counts and inaccurate step length estimation. Therefore, PDR is often a part of hybrid positioning solutions rather than a standalone one.

2.2. Factor Graph-Based Optimization

Factor graph model is in essence a graphic representation for estimation problems and it is widely adopted in SLAM. It is an alternative for sequential Bayesian estimation other than filter-based frameworks, e.g., Kalman filter and particle filter. In filter-based frameworks, the estimation problem is to recursively estimate a posterior and is suitable for online processing [25]. For factor graph optimization, the estimation problem is transformed to minimizing an error energy function. The error energy function is normally in a least square form and can represent the errors between the variables (needs to be estimated) derived “measurements” and the actual measurements. Unlike the filter-based framework, factor graph optimization estimates the variables in a batch and thus is processed offline.

Factor graph optimization generally consists of two steps: construct the factor graph (equivalent to form the error energy function) and optimize the graph. The first step differs across methods and implementations while the second step has general frameworks. It can be regarded as a general least square optimization (LSO) problem, which can be solved via iterative local linearizations using the Gauss-Newton or Levenberg-Marquardt [26] algorithms. By adopting the sparse nature of the graphs, the computational complexity can be greatly lowered using techniques such as Cholesky decompositions [27] and so on. As there already exists some mature and open-sourced methods for solving such LSO problem such as g2o [28] and ceres-solver [29], we focus on constructing the factor graph in this paper. In our implementation, we use the ceres-solver for the LSO problem.

3. Trajectory Generation Based on PDR

As aforementioned, the PDR algorithm consists of orientation estimation and stepometer estimation. In this paper a gradient descent-based orientation estimation algorithm is adopted, which adopts the measured magnetic direction and acceleration to compensate for the orientation error derived from integration over only angular rates. As for the stepometer estimation, we adopt the classical peak detection and heuristic step length estimation algorithms. With the estimated step length L_t and the heading θ_t at the current step at t , we can derive the current position (x_t, y_t) from the previous step (x_{t-1}, y_{t-1})

$$\begin{aligned} x_t &= x_{t-1} + L_t \cos(\theta_t) \\ y_t &= y_{t-1} + L_t \sin(\theta_t) \end{aligned} \quad (1)$$

Noting that as Equation (1) shows, the positioning error has accumulative nature because the next position is dependent on the previous one. The processing flow of the PDR algorithm adopted in our paper in Figure 2.

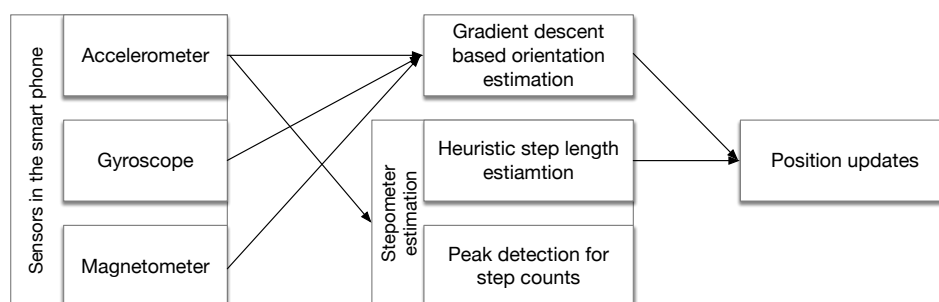


Figure 2. The processing flow for PDR in this paper.

3.1. Orientation Estimation

The gradient descent-based orientation estimation method was proposed by [30], which was originally adopted for an inertial-based human motion tracking system. Here we give a brief overview of how it is adopted for orientation estimation in PDR. The three steps are as follows:

1. Orientation updates based on angular rate measurements.

To avoid singularities, a quaternion representation is adopted here. The attitude of the phone can be updated according to the angular rate measurements in Equation (2)

$$\begin{aligned} {}^S\omega &= [0, \omega_x, \omega_y, \omega_z] \\ {}^S_N\dot{\mathbf{q}} &= 0.5 {}^S_N\bar{\mathbf{q}} \otimes {}^S\omega \end{aligned} \quad (2)$$

here $(\omega_x, \omega_y, \omega_z)$ is the measured angular rate from the gyroscope. The superscript S denotes the sensor frame and the subscript N denotes the navigation frame. ${}^S_N\dot{\mathbf{q}}$ denotes the quaternion derivative describing change rate of the navigation frame relative to the sensor frame. \otimes is the quaternion product operation and the bar of $\bar{\mathbf{q}}$ denotes the normalization operation of a quaternion. For discrete time implementation, it is

$$\begin{aligned} {}^S_N\dot{\mathbf{q}}_t &= 0.5 {}^S_N\bar{\mathbf{q}}_{est,t-1} \otimes {}^S\omega \\ {}^S_N\mathbf{q}_{est,t} &= {}^S_N\bar{\mathbf{q}}_{est,t-1} + {}^S_N\dot{\mathbf{q}}_t \delta t \end{aligned} \quad (3)$$

δt is the sampling interval of the gyroscope and we set it to 0.01 s in our implementation. Using Equation (3), the quaternion of the attitude can be updated. However, the error of the attitude is accumulative due to gyroscope drifts.

2. Forming the error function. To estimate the attitude more accurately, the measurements from the magnetometer and the accelerometer is adopted. An error function $f(\cdot)$ is formed for such purposes in Equation (4).

$$f(\cdot) = {}^S_N\bar{\mathbf{q}}^* \otimes {}^N\bar{\mathbf{M}} \otimes {}^S_N\bar{\mathbf{q}} - {}^S\bar{\mathbf{M}} \quad (4)$$

$*$ is the quaternion adjoint operation. The error function denotes the differences of normalized measurements (accelerometer measurements or magnetometer measurements) between representation in the sensor frame (${}^S\bar{\mathbf{M}}$) and derived representation in the sensor frame (${}^S_N\bar{\mathbf{q}}^* \otimes {}^N\bar{\mathbf{M}} \otimes {}^S_N\bar{\mathbf{q}}$) from the representation in the navigation frame (${}^N\bar{\mathbf{M}}$). Noting that ${}^S\bar{\mathbf{M}}$ and ${}^N\bar{\mathbf{M}}$ has a normalized quaternion representation.

Specifically, for the accelerometer measurements, ${}^S\bar{\mathbf{M}}$ and ${}^N\bar{\mathbf{M}}$ is replaced by ${}^S\bar{\mathbf{a}}$ and ${}^N\bar{\mathbf{g}}$ respectively, where

$$\begin{aligned} {}^S\bar{\mathbf{a}} &= (0, a_x, a_y, a_z) \\ {}^N\bar{\mathbf{g}} &= (0, 0, 0, 1) \end{aligned} \quad (5)$$

here the (a_x, a_y, a_z) is the normalized readings from the accelerometer. Noting that there is only one non-zero element in ${}^N\bar{\mathbf{g}}$, because when the phone is stationary or quasi-stationary, the only accelerometer should be the z-axis gravity.

For the magnetometer measurements, ${}^S\bar{\mathbf{M}}$ and ${}^N\bar{\mathbf{M}}$ is replaced by ${}^S\bar{\mathbf{m}}$ and ${}^N\bar{\mathbf{b}}$ respectively, where

$$\begin{aligned} {}^S\bar{\mathbf{m}} &= (0, m_x, m_y, m_z) \\ {}^N\bar{\mathbf{b}} &= (0, b_x, 0, b_z) \end{aligned} \quad (6)$$

here the (m_x, m_y, m_z) is the normalized reading from the magnetometer. In the navigation frame, the magnetic field can be considered only has horizontal component and vertical component, so ${}^N\bar{\mathbf{b}}$ only has two component. With the magnetometer readings and the accelerometer readings, the combined error function can be written as

$$f_{g,b} = \begin{bmatrix} f_g(\cdot) \\ f_b(\cdot) \end{bmatrix} \quad (7)$$

where $f_g(\cdot)$ is formed by replacing ${}^S\bar{\mathbf{M}}$ and ${}^N\bar{\mathbf{M}}$ in Equations (4) and (5), and $f_b(\cdot)$ is formed by replacing ${}^S\bar{\mathbf{M}}$ and ${}^N\bar{\mathbf{M}}$ in Equations (4)–(6). Then we have

$$f_g(\cdot) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 - q_3q_4) - a_y \\ 2(0.5 - q_2^2 - q_3^2) - a_z \end{bmatrix} \quad (8)$$

and

$$f_b(\cdot) = \begin{bmatrix} 2b_x(0.5 - q_3^2 - q_4^2) - 2b_z(q_2q_4 - q_1q_3) - m_x \\ 2b_x(q_2q_3 - q_1q_4) + 2b_z(q_1q_2 + q_3q_4) - m_y \\ 2b_x(q_1q_3 + q_2q_4) + 2b_z(0.5 - q_2^2 - q_3^2) - m_z \end{bmatrix} \quad (9)$$

q_1, q_2, q_3, q_4 are the components of the quaternion ${}^S_N\bar{\mathbf{q}}_{est,t-1}$.

3. Gradient descent for orientation estimation. To minimize the error function, the gradient descent method is adopted. Noting that here we only update the current estimation per time sample according to

$${}^S_N\mathbf{q}_{\nabla,t} = {}^S_N\bar{\mathbf{q}}_{est,t-1} - \gamma_t \frac{\nabla f_{g,b}(\cdot)}{\|\nabla f_{g,b}(\cdot)\|} \quad (10)$$

$\nabla f_{g,b}(\cdot)$ is the gradient of the error function $f_{g,b}(\cdot)$ and γ_t is a proper scale controlling descending velocity. Same as [30], γ_t is set to

$$\gamma_t = \left\| {}^S_N\dot{\mathbf{q}} \right\| \delta t \quad (11)$$

where δt is the sampling interval.

Combined with Equation (3), the attitude update process can be written as

$$\begin{aligned} {}^S_N\mathbf{q}_{t,est} &= {}^S_N\bar{\mathbf{q}}_{est,t-1} + {}^S_N\dot{\mathbf{q}}_{est,t-1}\delta t \\ {}^S_N\dot{\mathbf{q}}_{est,t-1} &= 0.5 {}^S_N\bar{\mathbf{q}}_{est,t-1} \otimes {}^S\omega + \beta \frac{\nabla f_{g,b}(\cdot)}{\|\nabla f_{g,b}(\cdot)\|} \end{aligned} \quad (12)$$

where the coefficient β is

$$\beta = \sqrt{\frac{3}{4}\omega_{err,max}} \quad (13)$$

$\omega_{err,max}$ is the maximum gyroscope measurement error. The derivation of β is explained in detail in [30] and is directly used in our paper. Then from the attitude quaternion, we can solve for the heading angle as the orientation θ_t .

3.2. Stepometer Estimation

For step occurrence detection we adopt the classical peak detection method. For the step length estimation, we adopt a heuristic parametric model in [24] as

$$L_t = K^4 \sqrt{a_{max} - a_{min}} \quad (14)$$

where a_{max} and a_{min} are the maximum and minimum acceleration during the step. K is a parameter which should be different across user to user. Here we set it as a constant 0.5. Noting that the step length estimation error is one of the error sources of PDR-based position tracking. The trajectories generated from PDR will later be re-estimated using Wi-Fi fingerprints and some pre-known landmark positions.

To sum up, the processing flow of orientation estimation is shown in Figure 3, where z^{-1} denotes one sample interval time back.

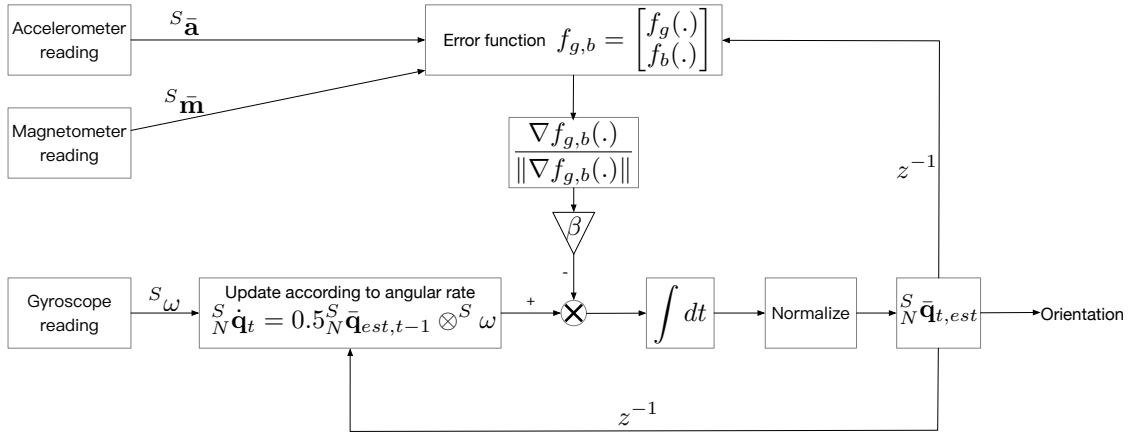


Figure 3. The processing of orientation estimation using readings from the accelerometer, magnetometer and gyroscope.

4. Factor Graph Optimization for RM Generation

The factor graph for RM generation in our method is shown in Figure 4. The variables in the circles denotes the poses at different times, which needs to be estimated. In our implementation, a pose consists of three components

$$\mathbf{s}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \quad (15)$$

where x_t and y_t are the horizontal positions and θ_t is the heading. From the factor graph, both Wi-Fi-based edges and PDR-based edges denote constraints between two poses, while the landmark-based edge denotes a constraints to a single pose. An error energy function can be drawn from the factor graph

$$F(\mathbf{s}_{1:t}) = F_{PDR}(\mathbf{s}_{1:t}) + F_{Wifi}(\mathbf{s}_{1:t}) + F_{landmark}(\mathbf{s}_{1:t}) \quad (16)$$

All the three sources of error energy have a quadratic form

$$F_{(\cdot)}(\cdot) = \mathbf{e}_{(\cdot)}(\cdot)^T \mathbf{\Omega}_{(\cdot)} \mathbf{e}_{(\cdot)}(\cdot) \quad (17)$$

where $\mathbf{e}_{(\cdot)}(\cdot)$ denotes the error between the actual measurements and measurements derived from the poses to be estimated. $\mathbf{\Omega}_{(\cdot)}$ denotes the information matrix of the measurements. By minimizing $(\mathbf{s}_{1:t})$, the maximum likelihood estimation of poses can be acquired. As the minimization process can be regarded as a general LSO problem, we adopt the ceres-solver [29] for solving the problem. Here we focus on how to form the different sources of error energy functions.

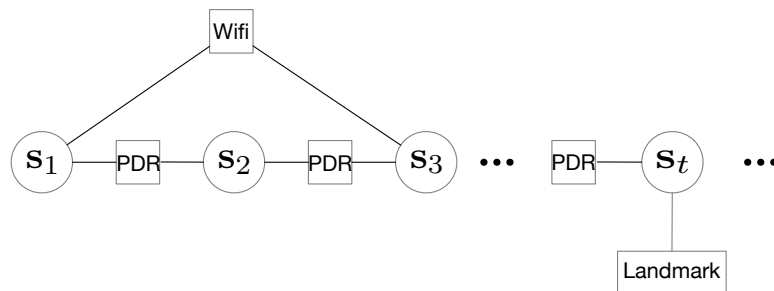


Figure 4. An illustration of the factor graph for RM generation. There are three types of edges: Wi-Fi-based edges, PDR-based edges and landmark-based edges.

After the optimization, the sequences of optimal poses are acquired. With these poses and the collected fingerprints, a raw RM can be established with associated fingerprints and geometric positions. We use an interpolation and extrapolation method in [31] to transform the raw RM into a grid-based one and make it ready for the online positioning phase.

4.1. PDR-Based Error Energy

The PDR algorithm can provide inertial generated trajectories. These trajectories (time sequences of poses) can be regarded as the initial values of the poses to be estimated or optimized. The PDR algorithm can also provide pose changes between adjacent steps. The pose change from the PDR algorithm is \mathbf{u}_t^{PDR} ,

$$\mathbf{u}_t^{PDR} = \begin{bmatrix} L_t^{PDR} \\ \delta\theta_t^{PDR} \end{bmatrix} \quad (18)$$

where L_t^{PDR} is the step length from PDR and $\delta\theta_t^{PDR}$ is the heading change from PDR. The pose change derived from the pose variables \mathbf{s} is

$$\mathbf{u}_t^s = \begin{bmatrix} \sqrt{(x_t^s - x_{t-1}^s)^2 + (y_t^s - y_{t-1}^s)^2} \\ \theta_t^s - \theta_{t-1}^s \end{bmatrix} \quad (19)$$

Then the PDR-based error energy is

$$F_{PDR}(\mathbf{s}_{1:t}) = \sum_t \mathbf{e}_{PDR}(\mathbf{s}_{t-1}, \mathbf{s}_t, \mathbf{u}_t^{PDR})^T \boldsymbol{\Omega}_{t,PDR} \mathbf{e}_{PDR}(\mathbf{s}_{t-1}, \mathbf{s}_t, \mathbf{u}_t^{PDR}) \quad (20)$$

where the error $\mathbf{e}_{PDR}(\cdot)$ is dependent on adjacent pose variable $\mathbf{s}_{t-1}, \mathbf{s}_t$ and the pose change from the PDR process \mathbf{u}_t^{PDR} .

$$\mathbf{e}_{PDR}(\mathbf{s}_{t-1}, \mathbf{s}_t, \mathbf{u}_t^{PDR}) = \mathbf{u}_t^s - \mathbf{u}_t^{PDR} \quad (21)$$

The information matrix for the PDR-based error has the form

$$\boldsymbol{\Omega}_{t,PDR} = \begin{bmatrix} \text{Info}(L_t) & 0 \\ 0 & \text{Info}(\delta\theta_t) \end{bmatrix} \quad (22)$$

where $\text{Info}(L_t)$ and $\text{Info}(\delta\theta_t)$ should be the reciprocal of the variance of PDR-based step length and heading change estimation respectively,

$$\begin{aligned} \text{Info}(L_t) &= \frac{1}{\sigma_{L_t}^2} \\ \text{Info}(\delta\theta_t) &= \frac{1}{\sigma_{\delta\theta_t}^2} \end{aligned} \quad (23)$$

In our implementation, $\text{Info}(L_t)$ and $\text{Info}(\delta\theta_t)$ are set to 20 and 500 respectively. If no other types of error energy are available, the minimum of $\mathbf{e}_{PDR}(\cdot)$ should be zero, when \mathbf{u}_t^s equals \mathbf{u}_t^{PDR} at every step. In this situation, the optimal pose estimations are the poses in the inertial generated trajectories from PDR.

4.2. Wi-Fi-Based Error Energy

Wi-Fi-based error energy is formed using the following two steps:

1. Find the distance between two fingerprints at the k^{th} and q^{th} step.

In Wi-Fi-based fingerprinting methods, a common assumption is often held true: if two fingerprints are with vicinity in signal space, then the positions where the fingerprints are collected are with vicinity in the coordinate space. In our method, we also find the

correspondences of positions by solving for the vicinities in signal space. We compare two arbitrary Wi-Fi fingerprints \mathbf{f}_k and \mathbf{f}_q which are collected at the k^{th} step and the q^{th} step ($k \neq q$). Then we define their distance in signal space like this using a metric similar to [6]

$$d(\mathbf{f}_k, \mathbf{f}_q) = \sqrt{\frac{\sum_i (f_k^i - f_q^i)^2}{N}} \quad (24)$$

where the superscript i on f_k^i denotes the RSS from the i^{th} AP in the vector \mathbf{f}_k , and N is the number of AP in the vector. The correspondences of the fingerprints and the pose variables are shown in Figure 5.

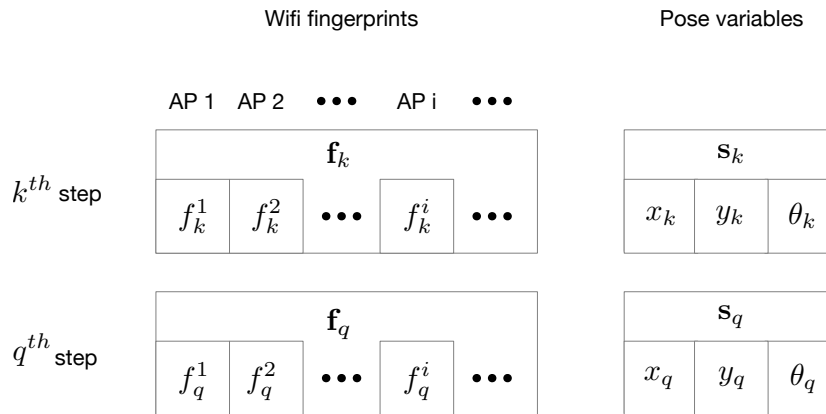


Figure 5. An illustration for the correspondences between the fingerprints and the pose variables.

- Find the error $e_{Wifi}(\mathbf{s}_k, \mathbf{s}_q, \mathbf{f}_k, \mathbf{f}_q)$ according to the distance in the signal space.

In our implementation, if two fingerprints' distance is less than a pre-defined threshold, the distances of the corresponding poses should be within a threshold (with vicinity). Then we define the Wi-Fi-based error like this

$$e_{Wifi}(\mathbf{s}_k, \mathbf{s}_q, \mathbf{f}_k, \mathbf{f}_q) = \begin{cases} d(\mathbf{s}_k, \mathbf{s}_q), & \text{if } d(\mathbf{f}_k, \mathbf{f}_q) < d_{thres, rss} \\ 0, & \text{if } d(\mathbf{f}_k, \mathbf{f}_q) > d_{thres, rss} \end{cases} \quad (25)$$

where $d(\mathbf{s}_k, \mathbf{s}_q)$ means the Euclidian distance between the horizontal positions of (x_k, y_k) and (x_q, y_q) calculated from the pose variables \mathbf{s}_k and \mathbf{s}_q . Figure 6 shows the relationships between the mean positioning errors and different values of $d_{thres, rss}$ in the experiment (details of the experiment will be described in Section 5). We can see that in our implementation, the mean error reaches the lowest around $d_{thres, rss} = 15$. This value is thus adopted for creating Wi-Fi-based constraints.

Then the Wi-Fi-based error energy is

$$F_{Wifi}(\mathbf{s}_{1:t}) = \sum_{k,q} e_{Wifi}^2(\mathbf{s}_k, \mathbf{s}_q, \mathbf{f}_k, \mathbf{f}_q) \Omega_{Wifi} \quad (26)$$

where k and q are two arbitrary step index from $1:t$ and $k \neq q$. Noting that the error is a constant zero when $d(\mathbf{f}_k, \mathbf{f}_q) > d_{thres, rss}$ as shown in Equation (25) and will not contribute to Equation (26). In our implementation, we only include the errors when $d(\mathbf{f}_k, \mathbf{f}_q) < d_{thres, rss}$. This will significantly reduce the number of constraints and is favorable in terms of computational cost for the optimization process later. Here the error $e_{Wifi}(\cdot)$, as well as Ω_{Wifi} are both scalars.

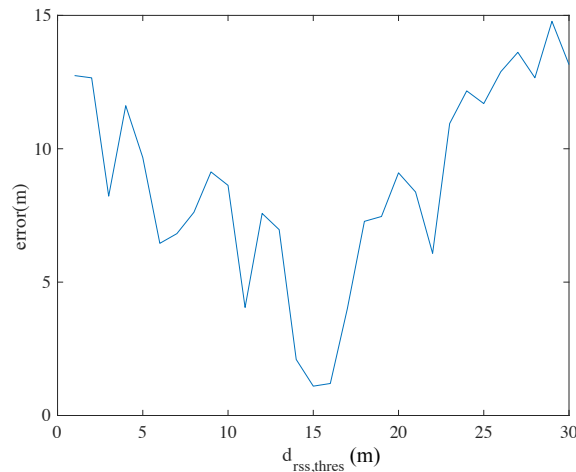


Figure 6. The mean positioning errors of the proposed method according to different values of $d_{thres,rss}$.

4.3. Landmark-Based Error Energy

Assuming we have a pre-defined landmark position $\mathbf{p}_{lm,j} = (x_{lm,j}, y_{lm,j})$ and we have already established the correspondence to the pose variable \mathbf{s}_j at the j^{th} step. This can be easily done by ask the user to press a button on the smart phone when walked to the landmark. Then the landmark-based error is

$$e_{landmark}(\mathbf{u}_j, \mathbf{p}_{lm,j}) = \sqrt{(x_{lm,j} - x_j)^2 + (y_{lm,j} - y_j)^2} \quad (27)$$

where (x_j, y_j) is the horizontal positions taken from the pose variable \mathbf{u}_j and the error $e_{landmark}(\mathbf{u}_j, \mathbf{p}_{lm,j})$ is also a scaler. Then the landmark-based error energy function is

$$F_{landmark}(\mathbf{s}_{1:t}) = \sum_j e_{landmark}^2(\mathbf{u}_j, \mathbf{p}_{lm,j}) \Omega_{landmark} \quad (28)$$

where the sum is over all recorded landmarks by the users and $\Omega_{landmark}$ should be a relatively large number, because it is believed the landmark positions are accurate.

5. Experiment

5.1. Experimental Setup

In our experiment, the Huawei Mate 9 smart phone is adopted. An Android application is developed for collecting data. The application has three main functions.

1. Generate the raw trajectories based on the PDR algorithm. The sampling rate of the accelerometer, gyroscope and magnetometer sensors in the phone is set to 100 Hz. The readings from these sensors can be processed in real time and can generate inertial-based raw trajectories. These poses of the trajectories with timestamps of the phone's system time are saved as file.
2. Collect Wi-Fi-based fingerprints. The Wi-Fi scanner on the phone is set to continuous scan mode with scanning interval of 1 s. However, the actual scanning interval can only reach about 2.5 s (due to system limitations). The fingerprints along with their collecting time are also saved as a file.
3. Record landmarks by pressing the landmark button. When walks to a pre-defined landmark, the user can press the buttons on the phone to record the time and landmark number.

All the types of data are saved as file on the phone with its collecting time. Afterwards, the data can be synchronized and processed offline to generate the RM. We assess the accuracy of the generated RM by assessing the accuracy of the positions where the fingerprints are collected (or the trajectories).

In our implementation, we record as many landmarks as possible. In this way, a part of the landmark recordings can be taken to generate the RM, the left part as ground truth positions to assess the accuracy of the optimized poses (trajectories). In the experiment, the positions of the landmarks are measured with a total station with sufficient accuracy.

5.2. Raw Trajectories Based on PDR

In Figure 7, a user walks in a mall with a hand-held smart phone. The generated trajectories from the PDR algorithm are shown. It is obvious that the estimated trajectory with magnetometer readings is more consistent with the true trajectory than the one without magnetometer readings. This is because the orientation estimation is much more accurate using magnetometer readings. This validates the orientation estimation method adopted in this paper. However, even adopting magnetometer readings, the PDR trajectory still cannot reach sufficient accuracy (the trajectory does not fit the floor plan and the true trajectory well). Other information is needed to re-estimate the trajectory to improve accuracy.

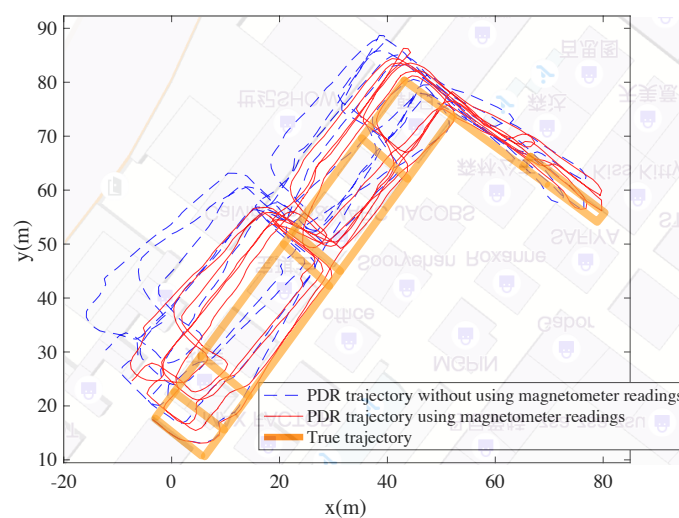


Figure 7. The PDR trajectories with and without magnetometer readings.

5.3. Factor Graph Optimization Results

The PDR trajectory can be re-estimated using factor graph optimization if other types of information are available. Here we present the optimized trajectory using PDR trajectory and

1. Wi-Fi fingerprints
2. landmark positions
3. Wi-Fi fingerprints and landmark positions

Respectively. In this experiment, we have set 18 different landmarks with known positions. The user has walked to these landmarks for 88 times, so that there are 88 landmark recordings in total. We only take 20 of them for factor graph optimization and the left 68 recordings as ground truth positions to verify the accuracy.

5.3.1. Results for Fusing PDR Trajectory and Wi-Fi-Based Constraints

Figure 8a shows the constraints of poses derived from Wi-Fi fingerprints. If two poses are connected with a green line, it means the fingerprints collected at the two poses are with vicinity in the signal space (Wi-Fi distance less than a threshold). In this case, the two poses should be also with vicinity and the term of the error energy should grow with the distances between the two poses. Noting that there are some false alarm constraints and can be regarded as outlier constraints. Figure 8b

shows the optimized trajectory using Wi-Fi fingerprints, which is more accurate than the original PDR trajectory.

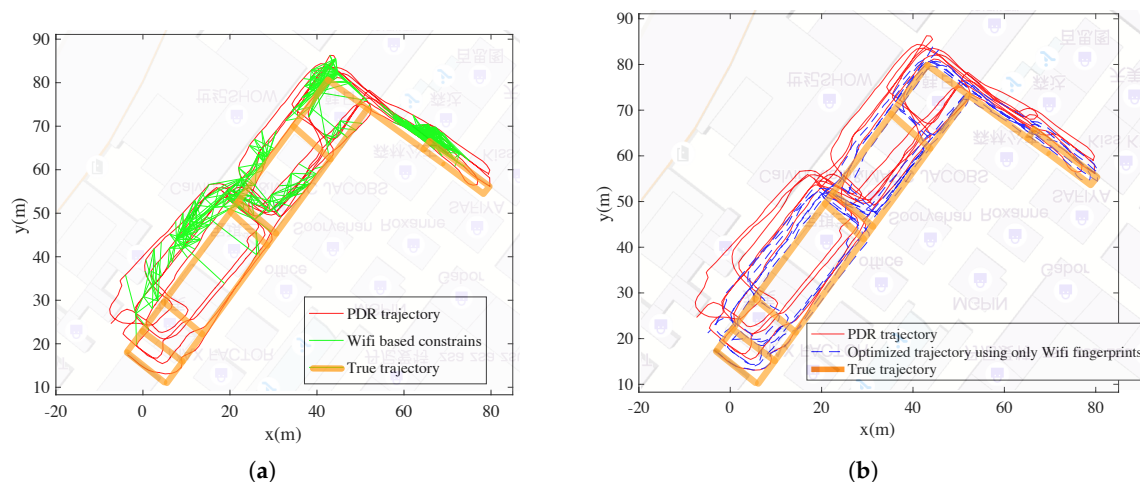


Figure 8. Results for fusing PDR trajectory and Wi-Fi fingerprints. (a) Wi-Fi-based constraints. (b) Optimized trajectory using Wi-Fi-based constraints.

5.3.2. Results for Fusing PDR Trajectory and Landmark Based Constraints

As mentioned before, the user has walked to the landmark for 88 times. However, we only randomly choose 20 of them as landmark constraints for optimization (training data), and the left is adopted to test the positioning accuracy (test data). As shown in Figure 9a, the landmark positions for training are black crosses and the test ones are black triangles. The black lines denote the constraints between the poses in the trajectory and the landmarks. Here there are only 20 constraints. The landmark positions for training can be regarded as direct observation of the poses' positions. After optimization adopting the landmark positions, the trajectory is shown in Figure 9b.

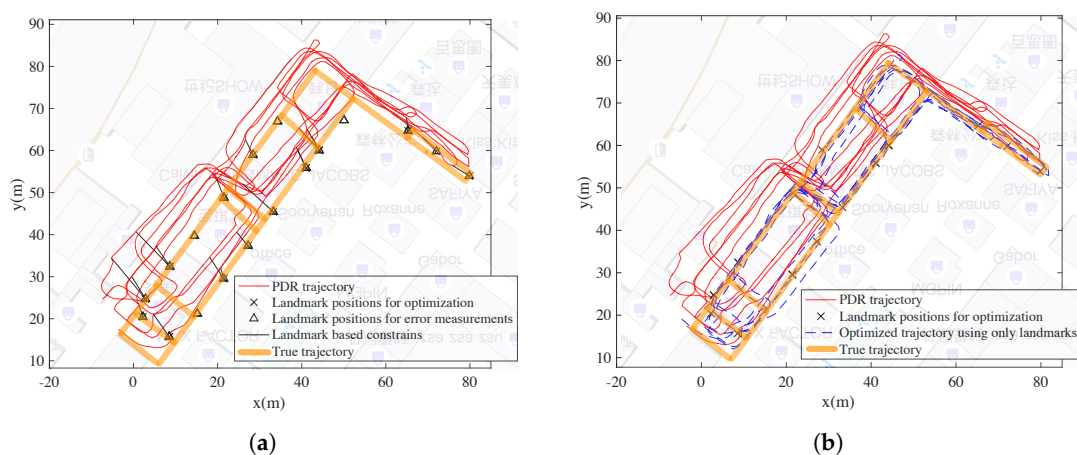


Figure 9. Results for fusing PDR trajectory and landmark positions. (a) Landmark based constraints. (b) Optimized trajectory using landmark constraints.

5.3.3. Results for Fusing PDR Trajectory, Wi-Fi-Based Constraints and Landmark-Based Constraints

The Wi-Fi-based constraints and the landmark-based constraints are shown together in Figure 10a, and the optimized trajectory are shown in Figure 10b. As mentioned, the positioning errors are measured according to the left 68 test landmark positions which are not adopted for optimization.

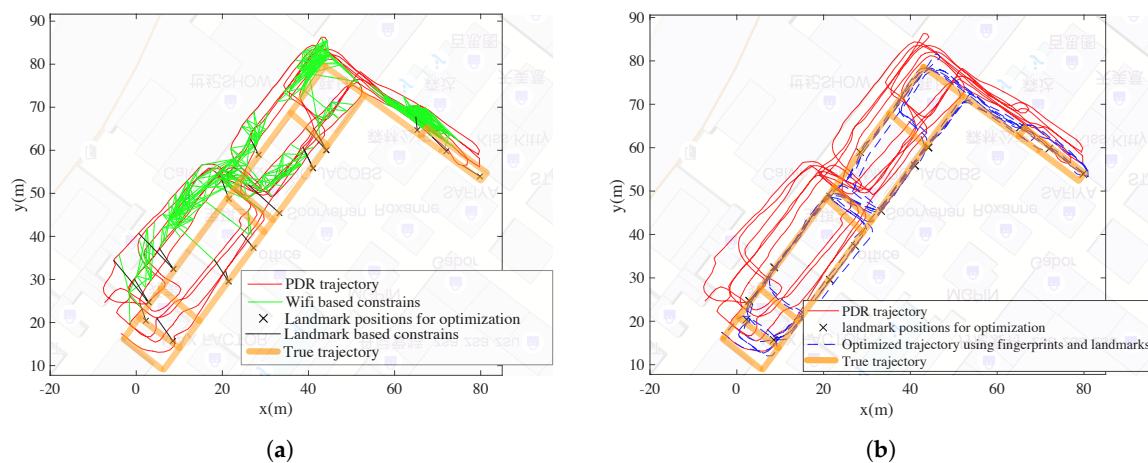


Figure 10. Results for fusing PDR trajectory, Wifi fingerprints and landmark positions. (a) Wifi based and landmark based constraints. (b) Optimized trajectory.

Figure 11 gives the cumulative density functions (CDFs) of the positioning errors with different trajectories. We can see the best performance is the result with all types of information including magnetometer readings, PDR results, Wi-Fi constraints and landmark positions. With all the information, the mean positioning error is about 1.10 m and the maximum error is about 2.25 m, which is considered sufficient for RM generation purposes.

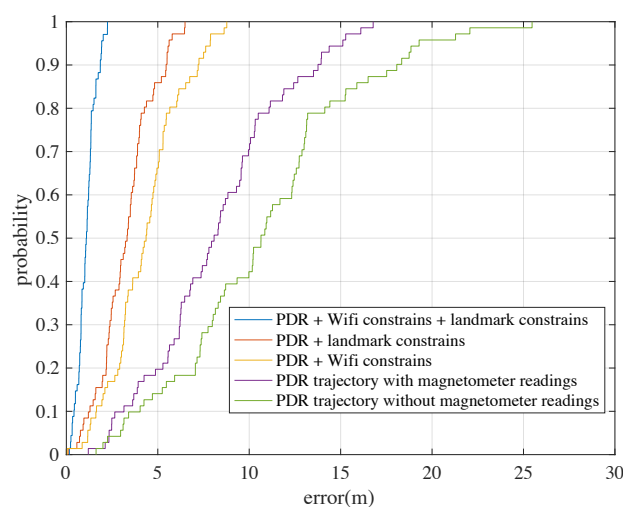


Figure 11. The CDFs of positioning errors at the test landmarks.

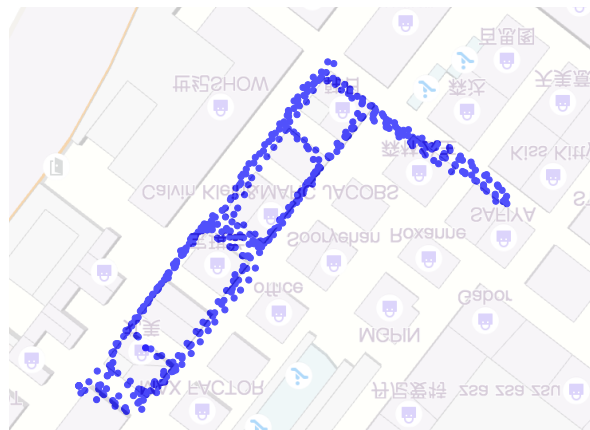
Table 1 gives some statistics of the positioning errors using different types of constraints. For pure PDR results, adopting magnetometer readings as suggested in Section 3.1 can help improve the accuracy of the generated trajectory. Noting that the mean error can decrease from 11.35 m to 8.67 m. We can also see that either adopting Wi-Fi-based constraints or landmark based constraints can contribute to improve the positioning accuracy. The mean error of the optimized trajectory adopting all types of constraints can be decreased to about 1.10 m. In addition, the maximum error for the trajectory is only 2.25 m, which shows robustness for position estimation. The errors of the estimated trajectory are sufficient for RM generation purposes.

Table 1. Error statics at the test landmarks for different trajectories.

Trajectory	Mean Error (m)	Maximum Error (m)	Median Error (m)
PDR trajectory without magnetometer readings	11.35	25.47	10.64
PDR trajectory with magnetometer readings	8.67	15.12	8.08
PDR + Wifi constraints	4.35	7.87	4.36
PDR + landmark constraints	3.12	5.56	3.32
PDR + Wifi constraints + landmark constraints	1.10	2.25	1.09

5.4. Wi-Fi-Based Positioning Results Adopting the Generated RM

After the graph optimization process, a series of positions (a trajectory) can be estimated. According to the timestamps of the positions and the timestamps of the fingerprints, a simple linear interpolation is made to solve for the positions where the RPs are. After interpolation, the positions of the RPs are available are shown in Figure 12. Noting that we only take one fingerprint at each location for factor graph creation purposes. The final RM includes all fingerprints collected.

**Figure 12.** The interpolated positions where fingerprints are collected.

These fingerprints are directly adopted for Wi-Fi-based positioning using the classical kNN algorithm as the baseline approach. Experiments are carried out to show the performance of Wi-Fi-based positioning adopting the constructed RM. According to Figure 13, the CDFs of positioning errors at different times in the day are shown. We can see that about 90% of the errors are less than 5 m for the test in the afternoon (when the RM data are collected). Although in the morning and in the evening the positioning performances are worse, the accuracies are still considered enough for many commercial indoor positioning cases, and can demonstrate the good quality of the constructed RM.

Another experiment is carried out to test how the different types of devices can affect the positioning performance. Here we have shown the positioning errors for 3 different devices adopting the RM (constructed from Huawei Mate 9) in Figure 14. Not surprisingly, the positioning performance of the Xiaomi Mix2 and Samsung S9 is not as good as the Huawei Mate 9, because their measurements of RSS are different. However, this shows that heterogeneous devices do have a large impact on the positioning performance. An easy way for solving the problem is to carry different devices and construct different versions of RMs. The detailed issues of the heterogeneous devices are considered as another topic and will be studied in the future.

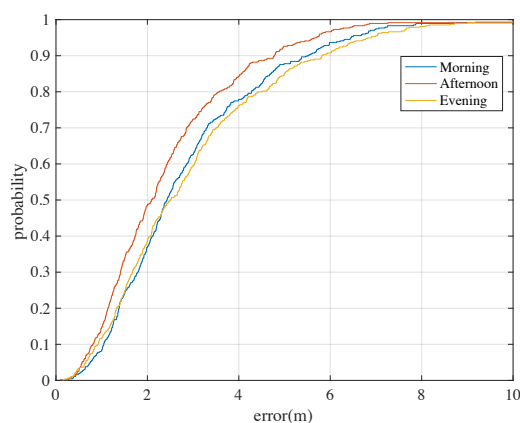


Figure 13. The CDFs of Wi-Fi-based positioning errors adopting the constructed RM at different times of a day using the classical baseline positioning algorithm: the kNN algorithm.

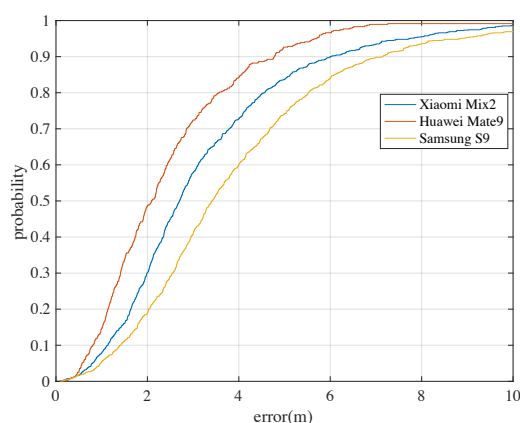


Figure 14. The CDFs of Wi-Fi-based positioning errors adopting the constructed RM with 3 different types of phones. (also using the classical kNN baseline algorithm).

6. Conclusions

A method for generating Wi-Fi-based RM is proposed in this paper. The method can work without any extra hardware such as an IMU, much additional information such as a detailed indoor map, and any extra surveying process such as collecting Wi-Fi fingerprints at known positions. The user only needs a smart phone to collect inertial-based readings (from the inertial sensors in the phone), Wi-Fi fingerprints and landmark positions. The inertial-based readings are adopted to generate a PDR-based raw trajectory. Then the factor graph optimization method is adopted to re-estimate the trajectory using constraints provided by Wi-Fi fingerprints and landmark positions. An experiment is carried out in a mall with a smart phone for data collection. The results have shown that both Wi-Fi-based constraints and landmark-based constraints can contribute to accurate position estimations in the RM. The accuracy of the optimized trajectory using both Wi-Fi fingerprints and landmark positions can reach a mean error of 1.10m, which is considered sufficient for RM generation purposes. Then the constructed RM can be adopted for Wi-Fi-based positioning with satisfactory accuracies.

Author Contributions: J.T., X.F. and S.W. conceived and worked together to achieve this work. Y.R. performed the experiments. J.T. wrote the paper.

Funding: This work was partly supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA19080100, Beijing Municipal Natural Science Foundation (Grant no. Z8162039), the Hainan Provincial Department of Science and Technology (Grant no. ZDKJ2016021), the Natural Science Foundation of Hainan (Grant no. 20154171) and the 135 Plan Project of Chinese Academy of Sciences (Grant No. Y6SG0200CX).

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

Abbreviations

The following abbreviations are used in this manuscript:

RSSI	received signal strength indication
APs	access points
RP	reference points
RM	radio map
kNN	k-nearest neighbor
IMUs	inertial measurement units
CS	compressed sensing
LASSO	randomized least absolute shrinkage and selection operator
GP	gaussian process
PDR	pedestrian dead reckoning
MEMS	Micro-Electro-Mechanical System
SLAM	simultaneous localization and mapping
LSO	least square optimization

References

1. 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, 11–12 March 2010; pp. 135–143. [\[CrossRef\]](#)
2. Yang, S.; Li, Q. Inertial Sensor-Based Methods in Walking Speed Estimation: A Systematic Review. *Sensors* **2012**, *12*, 6102–6116. [\[CrossRef\]](#) [\[PubMed\]](#)
3. 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 2, pp. 775–784. [\[CrossRef\]](#)
4. Youssef, M.; Agrawala, A. The Horus WLAN Location Determination System. In Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, WA, USA, 6–8 June 2005; ACM: New York, NY, USA; pp. 205–218. [\[CrossRef\]](#)
5. 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. [\[CrossRef\]](#)
6. Nowicki, M.; Skrzypczyński, P. *Indoor Navigation with a Smartphone Fusing Inertial and WiFi Data via Factor Graph Optimization*; Sigg, S., Nurmi, P., Salim, F., Eds.; Mobile Computing, Applications, and Services; Springer International Publishing: Cham, Switzerland, 2015; pp. 280–298.
7. Liu, H.H. The Quick Radio Fingerprint Collection Method for a WiFi-Based Indoor Positioning System. *Mob. Netw. Appl.* **2017**, *22*, 61–71. [\[CrossRef\]](#)
8. Liu, H.H.; Liu, C. Implementation of Wi-Fi Signal Sampling on an Android Smartphone for Indoor Positioning Systems. *Sensors* **2018**, *18*, 3. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Feng, C.; Au, W.S.A.; Valaee, S.; Tan, Z. Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing. *IEEE Trans. Mob. Comput.* **2012**, *11*, 1983–1993. [\[CrossRef\]](#)
10. Khalajmehrabadi, A.; Gatsis, N.; Pack, D.J.; Akopian, D. A Joint Indoor WLAN Localization and Outlier Detection Scheme Using LASSO and Elastic-Net Optimization Techniques. *IEEE Trans. Mob. Comput.* **2017**, *16*, 2079–2092. [\[CrossRef\]](#)

11. Atia, M.M.; Noureldin, A.; Korenberg, M.J. Dynamic Online-Calibrated Radio Maps for Indoor Positioning in Wireless Local Area Networks. *IEEE Trans. Mob. Comput.* **2013**, *12*, 1774–1787. [[CrossRef](#)]
12. Rakovic, V.; Angelicinoski, M.; Atanasovski, V.; Gavrilovska, L. Location estimation of radio transmitters based on spatial interpolation of RSS values. In Proceedings of the 2012 7th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), Stockholm, Sweden, 18–20 June 2012; pp. 242–247. [[CrossRef](#)]
13. Kubota, R.; Tagashira, S.; Arakawa, Y.; Kitasuka, T.; Fukuda, A. Efficient Survey Database Construction Using Location Fingerprinting Interpolation. In Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), Barcelona, Spain, 25–28 March 2013; pp. 469–476. [[CrossRef](#)]
14. Zhang, Q.; Zhou, Z.; Xu, W.; Qi, J.; Guo, C.; Yi, P.; Zhu, T.; Xiao, S. Fingerprint-free tracking with dynamic enhanced field division. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, 26 April–1 May 2015; pp. 2785–2793. [[CrossRef](#)]
15. Kim, K.H.; Kim, J.H.; Yoon, Y.J.; Seok, J.H.; Lim, J.W. Propagation model for the WLAN service at the campus environments. In Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, Jeju, Korea, 22–25 April 2003; Volume 1, pp. 196–200. [[CrossRef](#)]
16. Howell, J.; Menguc, M.; Siegel, R. *Thermal Radiation Heat Transfer*; CRC Press: Boca Raton, FL, USA, 2015.
17. Belmonte-Fernández, Ó.; Montoliu, R.; Torres-Sospedra, J.; Sansano-Sansano, E.; Chia-Aguilar, D. A radiosity-based method to avoid calibration for indoor positioning systems. *Expert Syst. Appl.* **2018**, *105*, 89–101. [[CrossRef](#)]
18. Fuschini, F.; Vitucci, E.M.; Barbiroli, M.; Falciasacca, G.; Degli-Esposti, V. Ray tracing propagation modeling for future small-cell and indoor applications: A review of current techniques. *Radio Sci.* **2015**, *50*, 469–485. [[CrossRef](#)]
19. Aly, H.; Youssef, M. New Insights into Wifi-based Device-free Localization. In Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, Zurich, Switzerland, 8–12 September 2013; ACM: New York, NY, USA; pp. 541–548. [[CrossRef](#)]
20. Qian, J.; Pei, L.; Ma, J.; Ying, R.; Liu, P. Vector Graph Assisted Pedestrian Dead Reckoning Using an Unconstrained Smartphone. *Sensors* **2015**, *15*, 5032–5057. [[CrossRef](#)] [[PubMed](#)]
21. Zampella, F.; Khider, M.; Robertson, P.; Jiménez, A. Unscented Kalman filter and Magnetic Angular Rate Update (MARU) for an improved Pedestrian Dead-Reckoning. In Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium, Myrtle Beach, SC, USA, 23–26 April 2012; pp. 129–139. [[CrossRef](#)]
22. Park, J.; Kim, Y.; Lee, J. Waist mounted Pedestrian Dead-Reckoning system. In Proceedings of the 2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Daejeon, Korea, 26–28 November 2012; pp. 335–336. [[CrossRef](#)]
23. Pratama, A.R.; Hidayat, R. Smartphone-based Pedestrian Dead Reckoning as an indoor positioning system. In Proceedings of the 2012 International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 11–12 September 2012; pp. 1–6. [[CrossRef](#)]
24. Alvarez, D.; Gonzalez, R.C.; Lopez, A.; Alvarez, J.C. Comparison of step length estimators from wearable accelerometer devices. In *Encyclopedia of Healthcare Information Systems*; IGI Global: Hershey, PA, USA, 2006; Volume 1, pp. 5964–5967. [[CrossRef](#)]
25. Gu, Y.; Song, Q.; Li, Y.; Ma, M. Foot-mounted pedestrian navigation based on particle filter with an adaptive weight updating strategy. *J. Navig.* **2015**, *68*, 23–38. [[CrossRef](#)]
26. Lourakis, M.I.A.; Argyros, A.A. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Softw.* **2009**, *36*, 2. [[CrossRef](#)]
27. Chen, Y.; Davis, T.A.; Hager, W.W.; Rajamanickam, S. Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. *ACM Trans. Math. Softw.* **2008**, *35*, 22. [[CrossRef](#)]
28. 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. [[CrossRef](#)]
29. Agarwal, S.; Mierle, K. Ceres Solver. Available online: <http://ceres-solver.org> (accessed on 20 February 2018).

30. Madgwick, S.O.H.; Harrison, A.J.L.; Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. In Proceedings of the 2011 IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, 29 June–1 July 2011; pp. 1–7. [[CrossRef](#)]
31. Talvitie, J.; Renfors, M.; Lohan, E.S. Distance-Based Interpolation and Extrapolation Methods for RSS-Based Localization With Indoor Wireless Signals. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1340–1353. [[CrossRef](#)]



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).