sensors

MDPI

Article

Design of a Hybrid Indoor Location System Based on Multi-Sensor Fusion for Robot Navigation

Yongliang Shi [1,2,3], Weimin Zhang [1,2,3,*], Zhuo Yao [1,2,3], Mingzhu Li [1,2,3], Zhenshuo Liang [1,2,3], Zhongzhong Cao [1], Hua Zhang [3] and Qiang Huang [1,2,3]

[1] School of Mechatronics, Beijing Institute of Technology, Beijing 100080, China; ylshi@bit.edu.cn (Y.S.); 2220170106@bit.edu.cn (Z.Y.); limingzhu@bit.edu.cn (M.L.); 2220170089@bit.edu.cn (Z.L.); czz1410407667@163.com (Z.C.); qhuang@bit.edu.cn (Q.H.)

[2] Key Laboratory of Biomimetic Robots and Systems, Beijing Institute of Technology, Ministry of Education, Beijing 100080, China

[3] Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing 100080, China; zzhanghua@163.com

* Correspondence: zhwm@bit.edu.cn; Tel.: +86-135-0115-3539

Received: 2 September 2018; Accepted: 18 October 2018; Published: 22 October 2018

check for updates

Abstract: In the case of a single scene feature, the positioning of an indoor service robot takes a long time, and localization errors are likely to occur. A new method for a hybrid indoor localization system according to multi-sensor fusion is proposed to solve these problems. The localization process is divided in two stages: rough positioning and precise positioning. By virtue of the K nearest neighbors based on possibility (KNNBP) algorithm first created in the present study, the rough position of a robot is determined according to the received signal strength indicator (RSSI) of Wi-Fi. Then, the hybrid particle filter localization (HPFL) algorithm improved on the basis of adaptive Monte Carlo localization (AMCL) is adopted to get the precise localization, which integrates various information, including the rough position and information from Lidar, a compass, an occupancy grid map, and encoders. The experiments indicated that the positioning error was 0.05 m; the success rate of localization was 96% with even 3000 particles, and the global positioning time was 1.9 s. However, under the same conditions, the success rate of AMCL was approximately 40%, the required time was approximately 25.6 s, and the positioning accuracy was the same. This indicates that the hybrid indoor location system is efficient and accurate.

Keywords: indoor localization; multi-sensor fusion; rough localization; KNNBP; precise localization; HPFL

1. Introduction

It is required that an indoor service robot always accurately estimates its own position in order to complete its task efficiently. Generally, an indoor wheel-based mobile robot estimates its own position by integrating wheel encoders and Lidar [1]. However, when in an environment with a low degree of distinction (e.g., a long corridor or hall), robot positioning is time-consuming and has low accuracy.

To solve these problems, numerous approaches that use inertial sensors, ultrasonic, radio-frequency (RF) ID tags, Wi-Fi, Lidar, vision sensors, and so on have been proposed [2–8]. Grid localization and Monte Carlo localization (MCL) are two common approaches to dealing with the global localization problem [9–11]. The crucial disadvantage of these two approaches is that they bear a heavy online computational burden [12], and the bigger the working space is, the more serious the problem. To solve this problem, Dellaert and Fox [13] introduced the adaptive Monte Carlo localization (AMCL) algorithm. Kullback–Leibler divergence (KLD) sampling makes the localization more efficient, so AMCL is the most widely used approach in robot localization [14]. However, if the global pose happens to be incorrect, AMCL is unable to recover from this global localization failure. Moreover, optimal filtering for non-parametric observation models that optimize the particle distribution was proposed [15], and the efficiency of global positioning was improved, but the failure rate of global localization was still very high. To enhance the performance of global positioning, Thrun [16] proposed the mixture MCL algorithm, but the two steps of sampling and calculating the importance weight increase the computational difficulty, and are very difficult to realize. Therefore, the improved MCL cannot absolutely meet the requirements of a robot in terms of the accuracy and real-time performance of global positioning.

For mobile robots, the upper limit has been reached by solely relying on Lidar and encoders for accurate positioning, and sensor-fusion-based methods are the main trends for indoor localization [17]. Multi-observation sensor resetting (MOSR) localization converges more quickly and accurately by relaxing the assumption that observations from separate frames of the camera are independent [18], but the camera is easily influenced by illumination conditions, and the positioning accuracy of robot drops when the illumination changes drastically [19]. To overcome the positioning error caused by the change of light, Choi [20] introduced an RFID system based on a tag floor using passive RFID tags for mobile robot localization. However, the average error is seriously dependent on the deployment of RFID tags because of its high revamping requirements for the environment where the service robot works. Therefore, this method is not suitable for application in service robots. Depth camera-based indoor mobile robot localization was introduced by Biswas [21], and 3D depth data were used, making the calculation load large. Although the positioning accuracy was less than 0.1 m, the global positioning efficiency was low because the positioning time reached 20 s. Luca [22] proposed a method of combining a fingerprinting algorithm based on K nearest neighbors (KNN) and a particle filter. However, the average error of this method was 1.28 m, making it unable to adapt to the precise positioning of the robot. Particle filter robot localization through robust fusion was proposed [23], but the installation of external equipment is complicated and is not conducive to the popularization of robots. Biswas [24] combined a Wi-Fi signal and Lidar data to realize the localization of a mobile robot by the particle filter algorithm, but the average convergence time was as high as 11 s, and the average positioning error reached 0.7 m.

In this paper, a new indoor localization system is proposed that fuses Wi-Fi, magnetic compass, encoders, and Lidar data and uses a particle filter and a Wi-Fi fingerprint localization algorithm, which effectively solves the problems of accuracy and efficiency in robot global positioning. The Wi-Fi fingerprint localization algorithm adopts the K nearest neighbors based on possibility (KNNBP) algorithm first created here. The probability-based similarity is used instead of the traditional Euclidean distance, eliminating the negative effects of invalid RSSI. So, it is suitable for diverse Wi-Fi environments. The Wi-Fi positioning and the compass provide more effective constraints for the particle distribution, making the robot global positioning less iterative with less resource consumption. When global positioning for the robot is required, the KNNBP Wi-Fi fingerprint localization algorithm first uses the Wi-Fi RSSI to get the global rough position of the robot, and the rough position, Lidar, compass, and grid map are used to get the precise localization of the robot according to the hybrid particle filter localization (HPFL) algorithm. In this process, the approach of uniformly distributing particles in the whole map is abandoned. Instead, only a small number of particles is maintained, which effectively improves the convergence efficiency of the particle filtering. In addition, Gaussian noise is added to the particles after converging to improve the diversity of particles, so the robustness of pose tracking is enhanced. Through the method proposed in this paper, not only the global positioning accuracy and efficiency, but also the local pose tracking of the robot is greatly improved.

The key contributions of the introduced solution can be summarized as follows:

1. An indoor localization system in a fusion of multiple sensors is designed, improving the accuracy rate and efficiency of global positioning for the indoor service robot.
2. A novel Wi-Fi fingerprint localization algorithm called KNNBP is created. Compared with the traditional fingerprinting algorithm, this algorithm adopts the similarity degree based on

conditional probability instead of Euclidean distance, thus improving positioning accuracy and adaptability to diverse indoor environments.

3. HPFL is proposed, and saves computing resources compared with the traditional Monte Carlo localization. Thus, the real-time performance and accuracy are both achieved. Moreover, due to the amelioration of particle diversity, the robustness of local pose estimation is also improved.

The rest of the paper is organized as follows: In Section 2, a system overview is presented, and the methodologies proposed in this paper are presented. In Section 3, experiments are carried out in a long corridor to prove the accuracy and efficiency of the proposed system. In Section 4, discussions about the performance of the system are provided and future work is outlined. Finally, some conclusions are given in Section 5.

2. System Overview and Methodology

In this section, an overview of the system is presented. Then, key modules and important algorithms are described, which include Wi-Fi information preprocessing to build a fingerprint database, rough localization achieved by KNNBP, and precise localization based on HPFL.

2.1. System Overview

The multi-sensor localization system is composed of a Wi-Fi module, a Lidar unit, a magnetic compass, and encoders, as illustrated in Figure 1. We utilized a mini computer with an Intel Core i3 CPU at 2.0 GHz, 2 GB of RAM, and a solid state drive of 64 GB for processing the robot positioning algorithm, including rough localization based on KNNBP and precise localization according to HPFL. Additionally, sensor data for positioning and navigation were also collected through it. The Wi-Fi module embedded in the computer was used to collect the corresponding RSSI and MAC address of each Access Point (AP), and this information was used to establish a fingerprint database and to match the current fingerprint information with the fingerprint database during rough positioning. The magnetic compass was used to provide orientation information for the robot that is used in HPFL for precise localization, and the accuracy of this compass was 2°. The encoder provided mileage information to be used in the prediction process during which the sample motion model odometry is adopted [11], and it was also used in building a grid map. The Lidar was a SICK Tim561 laser detector with 10 m working range and 270° aperture angle. In the update stage of precise positioning based on HPFL, the detection information is mainly used to match the grid map corresponding to the environment to estimate the position and posture information of the robot, for obstacle detection and building the grid map.

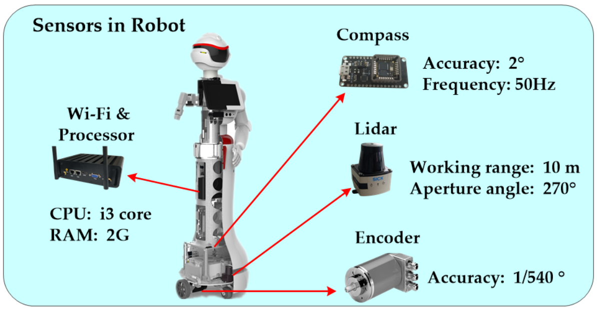


Figure 1. Sensors in the robot.

The multi-sensor fusion localization system is shown in Figure 2 as the following:

Step 1: Cartographer [25] is used to establish a global 2D occupancy grid map. In the process of building a map, the center processor records a set of Wi-Fi fingerprint data every 2 m, and a Wi-Fi fingerprint database matching with the grid map is generated after processing the recorded RSSI.

Step 2: The grid map and the fingerprint database are loaded for global positioning, and the current RSSI of the robot matches with the information in the fingerprint database through the KNNBP algorithm to obtain the rough position of the robot.

Step 3: According to the rough position provided by the Wi-Fi localization and the attitude information provided by the magnetic compass, the HPFL is used to get the precise global position.

Step 4: In the process of indoor navigation, the sample motion model is used to predict the position and posture of the robot, and according to the Likelihood range finder module to update the position and posture, the accurate position of the robot is updated in real time to ensure the position tracking of the robot. To ensure the global localization, the robot is forced to repeat Step 2 every 30 min.

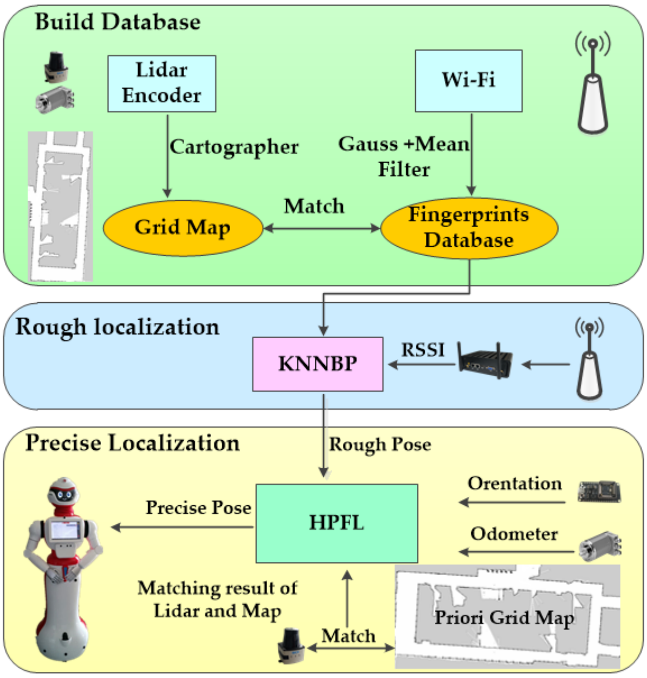


Figure 2. Multi-sensor fusion localization system flow. HPFL: hybrid particle filter localization; KNNBP: K nearest neighbors based on possibility; RSSI: received signal strength indicator.

2.2. Building the Fingerprint Database

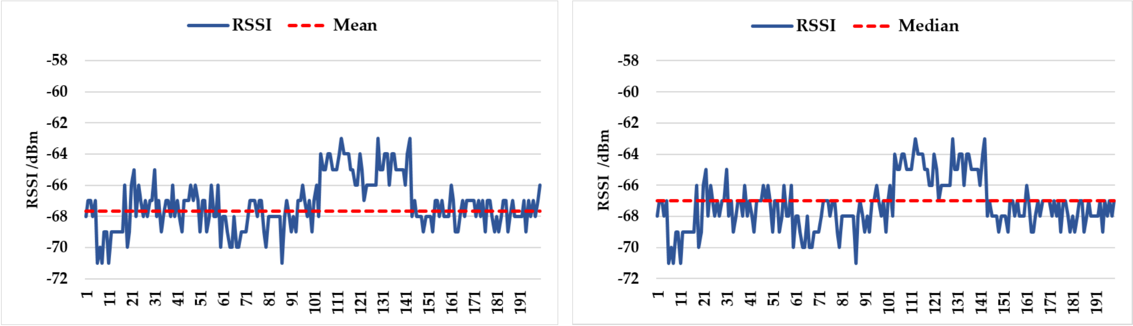
The core problem of establishing the fingerprint database is the pretreatment of the Wi-Fi signal. Although the RSSI is the most commonly used reference information, it is often prone to severe multi-path influence and measurement errors. Therefore, in practical applications, the use of preprocessing techniques on raw collected data helps the RSSI to be a suitable estimator of distance [26]. Rough localization is performed using RSSI, which is the parameter used to estimate distance between mobile and anchor nodes or APs [27]. The position information of the robot and the MAC address and RSSI of different APs are recorded at the recording positions. The RSSI of an AP is collected once every 1 s, and 200 sets of data are obtained in a position. Median filtering and mean filtering are the two most commonly used data preprocessing methods. Therefore, the effects of the two algorithms are compared, and the mean filter is

$$\mathrm{RSSI} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{RSSI}_i. \tag{1}$$

The median filter is

$$\mathrm{RSSI} = \text{Median of } \{\mathrm{RSSI}_1, \mathrm{RSSI}_2, \mathrm{RSSI}_3, \ldots, \mathrm{RSSI}_N\}. \tag{2}$$

As illustrated in Figure 3a, the signal always fluctuates around the mean value, so the data after the mean filtering can represent the signal intensity information of an AP. In Figure 3b, the median of the data set is significantly different from the characteristics of the data set.



(**a**) (**b**)

Figure 3. (**a**) The relationship between the RSSI set and its mean, and (**b**) the relationship between the RSSI set and its median.

However, there are also some RSSI values that have a small occurrence probability and a great difference from the wave center of the signal intensity, because mean filtering is an extremely susceptible value of RSSI. So, using only a mean filter cannot meet the requirement of finding a value that can describe the performance of an RSSI set. Therefore, it is necessary for us to continue to do data processing for Wi-Fi fingerprint information using other methods. In Figure 4, we analyze the RSSI according to a signal intensity probability distribution histogram.

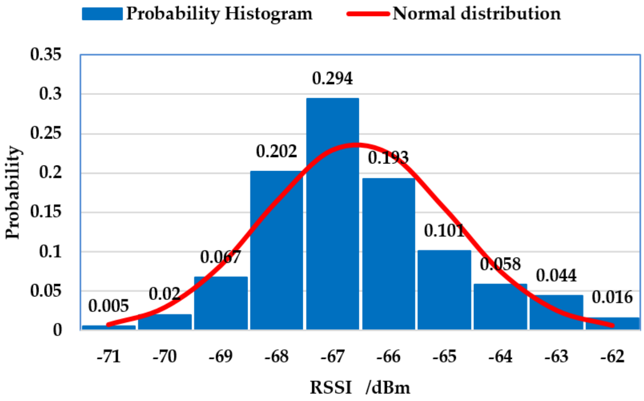


Figure 4. This histogram indicates the RSSI probability distribution of measured values. The *x*-axis is the intensity value of the received signal, and the *y*-axis represents the probability of the received signal intensity values in 1000 measurements. The curve is the normal distribution of the measured RSSI.

From the probability distribution histogram, we can see that the signal intensity was characterized by a normal distribution. The intensity of the signal between −68 dBm and −66 dBm appeared most frequently, and then decreased on both sides, and the frequency of signal intensity on the left and right sides of the histogram were small-probability events. To improve the reliability of the mean filtering, the intensity values of these signals were eliminated. Therefore, a Gaussian filter was introduced to eliminate the noise before performing the mean filtering [28], thus reducing the intensity error of these signals.

According to the histogram of the signal distribution, RSSI signal intensity obeys a normal distribution

$$\frac{\mathrm{RSSI}-\mu}{\sigma} \sim N(\mu,\sigma^2), \tag{3}$$

where μ is the mean, and σ is the standard deviation:

$$\mu = \frac{1}{n}\sum\nolimits_{i=1}^{n} \mathrm{RSSI}_i, \tag{4}$$

$$\sigma = \sqrt{\left(\frac{1}{n}\sum\nolimits_{i=1}^{n} \left(\mathrm{RSSI}_i - \frac{1}{n}\sum\nolimits_{i=1}^{n} \mathrm{RSSI}_i\right)^2\right)}. \tag{5}$$

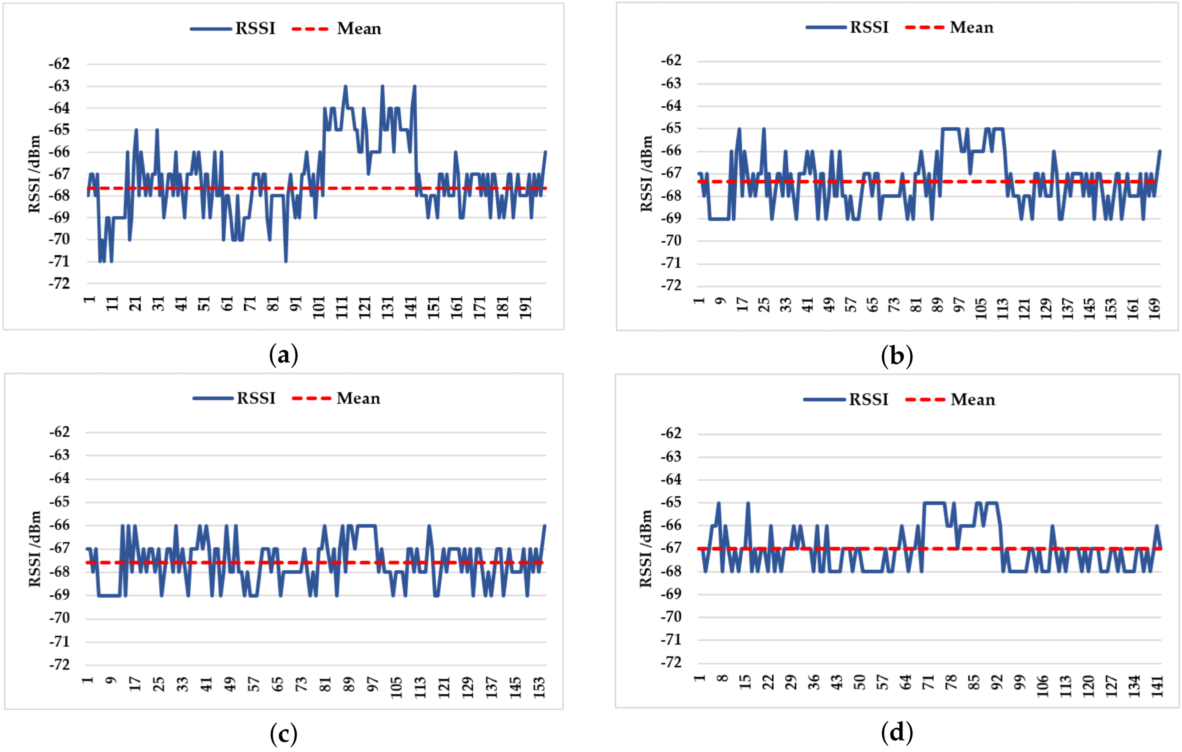
To avoid over- or under-filtering, we selected the data within 90%, 80%, and 68.3% of the probability:

$$\mathrm{RSSI} \in (\mu - 1.65\sigma, \mu + 1.65\sigma), \tag{6}$$

$$\mathrm{RSSI} \in (\mu - 1.28\sigma, \mu + 1.28\sigma), \tag{7}$$

$$\mathrm{RSSI} \in (\mu - \sigma, \mu + \sigma). \tag{8}$$

In the present study, the 200 sets of data collected were processed by a Gaussian filter. According to the collected data, the mean μ was −67.25 dBm, and the variance σ was 1.66, so the final ranges of RSSI were (−70.0 dBm, −64.5 dBm), (−69.3 dBm, −65.1 dBm), and (−68.9 dBm, −65.6 dBm) when the data were within 90%, 80%, and 68.3% of the probability. By means of the mean filter after the Gaussian filter, we obtained the signal intensity change line graph as illustrated in Figure 5. It can be seen from Figure 5c that the mean of RSSI data within 80% probability could mostly depict the characteristic of the set, and after the mixed filtering method of Gaussian and mean filtering, the signal intensity fluctuated less, and the mean value could represent the whole performance of all tested RSSI values at the same position. The mean of the processed data set by Gaussian filter within 80% probability approximated the characteristics of the set. Therefore, the data set was preprocessed by a Gaussian filter within 80% probability and mean filter.



(**a**) (**b**)

(**c**) (**d**)

Figure 5. (**a**) The raw RSSI data and their mean values before Gaussian filtering. (**b**) The relationship between RSSI data within 90% of the probability and their mean. (**c**) The RSSI data within 80% of the probability and their data. (**d**) The relationship between the RSSI data within 68.3% of the probability and their mean.

In practical applications, 200 sampling times per location leads to an inefficient process in establishing a fingerprint database. Therefore, this paper verifies the optimal sampling times. First, 200 RSSI data were collected and processed as mentioned above at a certain position in the laboratory, and the obtained mean was regarded as a reference. Then, at the same location, the RSSIs were collected for different times and processed with the same methods as before to get the final mean, as illustrated in Figure 6. The figure illustrates that the mean value basically fluctuated near the reference value when the sampling times was greater than or equal to 30. Therefore, the optimal number of samplings was 30.

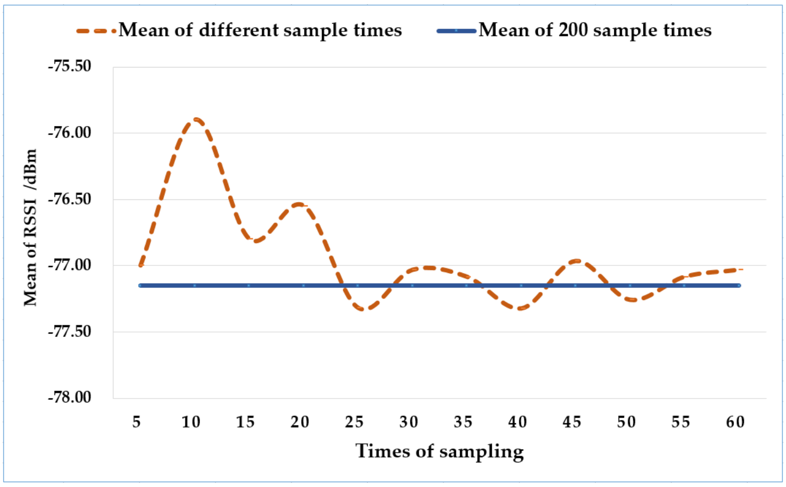


Figure 6. Verification of optimum sampling times.

Finally, the fingerprint database has to be built. When the robot uses cartographer to construct the occupancy grid map, it optimizes the pose based on the branch-and-bound approach for computing scan-to-submap matches [25]. The robot performs a set of Wi-Fi information collection every 2 m according to the optimized position, and RSSI values are recorded every 1 s for 30 consecutive times at each recording location. Then, Gaussian and mean filtering are performed on 30 RSSI values corresponding to each AP, and the mean value is used as Wi-Fi fingerprint information for that location, such as RSSI_{m2} in Table 1, which represents the RSSI value of MAC_m (every AP has a MAC address) in position (x_2, y_2). A Wi-Fi fingerprint database matching with the grid map is ultimately generated. The format of the database is illustrated in Table 1, where m is the number of APs, n is the number of recorded positions, MAC is the MAC address corresponding to an AP, and (x, y) is the recorded position of the grid map matching with the environment.

Table 1. Format of the Wi-Fi fingerprint database.

	(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	...	(x_n, y_n)
MAC_1	RSSI_{11}	RSSI_{12}	RSSI_{13}	...	RSSI_{1n}
MAC_2	RSSI_{21}	RSSI_{22}	RSSI_{23}	...	RSSI_{2n}
...	...	...	...	...	...
MAC_m	RSSI_{m1}	RSSI_{m2}	RSSI_{m3}	...	RSSI_{mn}

2.3. *Rough Localization*

In the rough positioning stage, we denote $F_c = \{(ma_1, r_1), (ma_2, r_2), (ma_3, r_3), \ldots, (ma_m, r_m)\}$ as the Wi-Fi fingerprint information obtained from location points currently, where ma represents the MAC address of the detected AP, and r represents the RSSI value of the AP. Denote $FD = \{F_1, F_2, F_3, \ldots, F_n\}$ as the fingerprint database. $F_j = \{(MA_{j1}, R_{j1}), (MA_{j2}, R_{j2}), (MA_{j3}, R_{j3}), \ldots, (MA_{jm}, R_{jm})\}$ $(F_j \in FD)$ represents the fingerprint information corresponding to different location points in the fingerprint database, in which *MA* represents the MAC address of the recorded AP in *FD*, and *R* represents the RSSI value of the AP.

At the establishment stage of the fingerprint database, *n* positions are sampled, and Wi-Fi fingerprints are generated. The Wi-Fi fingerprint database is a sequence of F_j ($j \in \{1, 2, \ldots, n\}$). At the online positioning stage, the robot first collects the Wi-Fi fingerprint information F_c of the current position, then the information is matched with the fingerprint database *FD* to find the position of the pending point F_j that is mostly similar to the Wi-Fi fingerprint information of the current position.

K-nearest neighbors (KNN) is the most basic and popular discriminative technique matching algorithm for Wi-Fi fingerprint location [29,30]. Based on a similarity measurement such as a distance function, the KNN algorithm determines the *K* closest matches in the signal space to the target. Then, the location of the target can be estimated as the average of the coordinates of the *K* neighbors. The KNN algorithm uses the Euclidean distance as the similarity as expressed by Equation (9) [31]. That is, it selects the corresponding position of the minimum *D* as the coordinates to be obtained:

$$D = \sqrt{\sum_{i=1}^{m} (r_i - R_i)^2}. \tag{9}$$

The RSSI is a function of the distance between the transmitter and the receiving device, which varies due to kinds of in-path interference [32]. According to [33,34], the log-distance path loss (LDPL) model is typically adopted to map RSSI to propagation distance *d*:

$$\text{RSSI} = P_t - (PL(d_0) + 10 \cdot \xi \cdot \lg(\frac{d}{d_0}) + X_\theta), \tag{10}$$

where P_t is the transmission power in dBm, $PL(d_0)$ is the path loss at reference distance d_0, and ξ is the path loss exponent. X_θ is a zero-mean normal random variable reflecting shadowing attenuation in dB. Simulation according to Equation (10) and actual measurements about the relationship between distance and RSSI are illustrated in Figure 7.

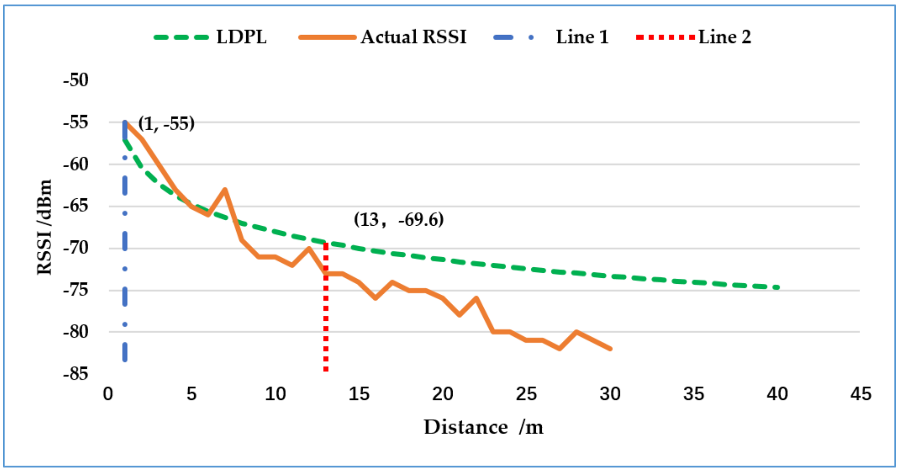


Figure 7. The relationship between distance and RSSI of theoretical simulation and actual measurements. LDPL: log-distance path loss.

The attenuation of RSSI is affected by distance and other interference, among which the distance is the main reason when it is not too large. In the present study, since the RSSI value is regarded as the reference information of the distance, we are looking for a critical point. When the attenuation value of RSSI reaches this critical point, the attenuation of RSSI is more affected by the unexpected interference factor beyond the distance. Here, we define this attenuation value as *V* (which will be used in Algorithm 1). In Figure 7, the absolute gradient of RSSI descending between Line 1 and Line 2 is larger than that after Line 2, and both the actual RSSI and the LDPL follow the rule. So, the critical point is (13, −69.6). This means that when it is about 13 m away from the measured position (i.e., when the V = 15 dBm (it should have been 14.96, and approximation is taken for ease of calculation)), the influence of distance is prone to be small, and it is increased sharply by other interference. So, the RSSI value that is 13 m away from the measured position is unreasonable as the reference information of positioning.

The Wi-Fi fingerprint matching algorithm based on KNN uses all RSSI information including invalid points (e.g., the AP outside 13 m), and its RSSI value cannot be used as the information to judge the distance between the measured position and the target point, so as to introduce an error. Therefore, the accuracy and stability of the matching result are limited. In this paper, we propose a new fingerprint matching algorithm called KNNBP. Compared with the KNN algorithm, KNNBP uses a similarity degree based on conditional probability instead of Euclidean distance, which eliminates the interference of invalid APs. Finally, the normalized probability values are used as judgment information, and the reliability and accuracy are improved. The algorithm defines p as the similarity of each AP, and P represents the similarity between the current location and the location in the fingerprint database. The algorithm is presented in Algorithm 1.

Algorithm 1 KNNBP.

```
Input: $FD=\{F_1, F_2, F_3, \ldots, F_n\}$
       $F_j=\{(MA_{j1}, R_{j1}), (MA_{j2}, R_{j2}), \ldots, (MA_{jm}, R_{jm})\}$ ($F_j \in$ FD)
       $F_c=\{(ma_1, r_1), (ma_2, r_2), \ldots, (ma_m, r_m)\}$
for $j = 1$ to $n$ do
    for $i = 1$ to $m$ do
        if $ma_i = MA_{ji}$ then
            if $|r_i - R_{ji}| < V$ then
                $p_i = 1 - \frac{|r_i - R_{ji}|}{V}$
            end if
        end if
    end for
    $P_j = \frac{1}{a}\sum_{i=1}^{a} p_i$
end for
    sort $P_j$ from largest to smallest
    select the first $K$ positions $(x_k, y_k)$ corresponding to $P_j$
    calculate position $(x, y) = \frac{1}{K}\sum_{j=1}^{K}(x_j, y_j)$
    return $(x, y)$
```

The process of online localization defines the Wi-Fi fingerprint information at the current location as F_c, which sequentially matches with F_j in the fingerprint database FD. The maximum difference is V = 15 dBm. As the algorithm indicates, when the MAC addresses of F_c and F_j are the same, if the RSSI value corresponding to the MAC address satisfies $|r_i - R_{ji}| \geq V$, then the RSSI value is filtered out; and if $|r_i - R_{ji}| < V$, the similarity degree p_i of the AP is calculated as

$$p_i = 1 - \frac{|r_i - R_{ji}|}{V}. \tag{11}$$

The normalized similarity of the measured points in this position is as follows:

$$P_j = \frac{1}{a}\sum\nolimits_{i=1}^{a} p_i, \tag{12}$$

where a ($a \leq m$) is the number of the same MAC addresses for F_c and F_j. The largest K ($K \geq 2$) fingerprint data of P are obtained, and the ultimate position is the mean position of the K corresponding reference points, as expressed in Formula (13), where (x, y) represents the estimated position, (x_j, y_j) is the position corresponding to P_j (j = 1, 2, . . . , K):

$$(x,y) = \frac{1}{k}\sum\nolimits_{j=1}^{K} (x_j,y_j)(j \in K). \tag{13}$$

2.4. Precise Localization Based on HPFL

The average error of the Wi-Fi fingerprint localization algorithm based on KNNBP was 2.6 m. So, single Wi-Fi fingerprint localization cannot meet the navigation requirements for a robot. On the basis of the Wi-Fi fingerprint localization, the present study used the grid map, odometer (encoder data), compass information, and Lidar information to locate the robot accurately. In nonlinear dynamic models, particle filtering methods are commonly applied to develop target tracking algorithms [35,36], so the particle filter is used in HPFL to perform global positioning and pose tracking. The algorithm is presented as Algorithm 2.

The algorithm defines χ_t as the particle set at time t, M as the number of current particle set, M_x is the maximum number of particles determined by the KLD sampling algorithm, and m represents the grid map by cartographer. The *global_flag* is the flag of global positioning, which is a Boolean value that is triggered when the robot starts and, like Step 4 of Section 2.1, fires every 30 min after starting work. Define x_t as the robot's pose, μ_t is sensor observation, and z_t is the measurement model at time t. $wifi(x,y)$ represents the position output of rough localization, $c(\theta)$ is the orientation provided by magnetic compass, ω_{max} is the maximum weight of a particle in the particle set, $\omega_{threshold}$ is the weight threshold of a particle, and m is the list attribute of the object in the map environment. The precise positioning process is as follows:

1. Prediction. The robot predicts its position in the grid map. In line 1, when the maximum weight of the particle is below the threshold or the number of particles is less than the limited number M_x, the particle filter starts working. In lines 4 and 5, if *global_flag* is triggered, the rough pose state of the robot is predicted based on Wi-Fi and compass information. The compass provides attitude information $c(\theta)$, and Wi-Fi fingerprint information provides the position information $wifi(x,y)$, because the angular accuracy of the compass is 2°, and the $wifi(x,y)$ has noise itself compared with the precise position. Thus, Gaussian noise is added on the $wifi(x,y)$ and $c(\theta)$ to improve the robustness of particles, because the accuracy $c(\theta)$ is higher than the $wifi(x,y)$, so $\sigma_1 < \sigma_2$.

$$\begin{array}{l} x_t^{[\mathrm{M}]}(\theta) = c(\theta) + N(0,\sigma_1) \\ x_t^{[M]}(x,y) = wifi(x,y) + N(0,\sigma_2) \end{array} \tag{14}$$

Then, a particle filter is created according to the Gaussian distribution near the rough position instead of the method of globally uniformly generating particles. When the global position is successfully obtained, the robot performs position tracking to maintain the pose to navigate. In line 7 of the algorithm, state prediction is performed according to the sample motion model odometry [11] in the position tracking phase, during which the odometry is obtained according to the encoder, and the current predicted pose of the robot to the particle x_t is assigned. The classical AMCL does not perform pose tracking until the robot begins to move, which is not good for local positioning of the robot. In HPFL, Gaussian noise $N(0,\sigma_3)$ is added even if the particles are converged, as expressed in

Equation (15). This guarantees the diversity of particles, so accurate pose tracking is achieved even if the robot stops moving.

$$x_t^{[M]} = sample\_motion(u_t, x_{t-1}^{[i]}) + N(0, \sigma_3) \tag{15}$$

In the sequential importance re-sampling particle filter (SIR-PF), a duplicating particle is the main reason for loss of the particle diversity [37]. Compared with the algorithm program 5.6 proposed by Thrun [11], HPFL adds Gaussian noise to the predicted robot pose to enhance the local diversity of particles, making the robot relocate, even if it is not moving, when the error between maximum likelihood estimate (MLE) and posterior estimate is large. Thus, the robustness of position tracking is enhanced.

Algorithm 2 HPFL.

```
Input:  particle set of last time $\chi_{t-1}$, orientation information $c(\theta)$, rough position $wifi(x,y)$,
        sensor observation $u_t$, measurement module $z_t$, grid map $m$
Initilization:  current particle set $\chi_t = \phi$ , number of current particles $M = 0$,
                max number of particles $M_x = 0$, max weight of M particles $\omega_{\max}^{[M]} = 0$
                the threshold of particle weight $\omega_{threshold}$
1:  while  $M < M_x$  ||  $\omega_{\max}^{[M]} < \omega_{threshold}$  do
2:    draw $i$ with probability $\propto \omega_{t-1}^{[i]}$     # Particles are extracted from the particle set at time $t-1$
3:    if (global_flag) then     # If global localization is required
4:      $x_t^{[M]}(\theta) = c(\theta) + N(0, \sigma_1)$     # Predict orientation of particles according to compass
5:      $x_t^{[M]}(x,y) = wifi(x,y) + N(0, \sigma_2)$     # Predict position of particles according to Wi-Fi
6:    else
7:      $x_t^{[M]} = sample\_motion(u_t, x_{t-1}^{[i]}) + N(0, \sigma_3)$
        # Predict position of particles according to encoder
8:    end if
9:    $\omega_t^{[M]} = measurement(z_t, x_t^{[M]}, m)$     # Update weights of particles
10:   $\chi_t = \chi_t + \left\langle x_t^{[M]}, \omega_t^{[M]} \right\rangle$     # Update particle set based on Importance Re-sampling
11:   calculate $M_x$ according to KLD_sampling
12:   mark the particles located in obstacles and filter them
13:   $M = M + 1$
14: end while
15: select the $K$ positions $(x_i, y_i)$ in $\chi_t$ according to the largest $K$ weight $\omega^{[M]}(i)$
16: return  $(x,y) = \frac{1}{K} \sum_{i=1}^{K} (x_i, y_i)$
```

2. Update. In line 9, the sensor measurement z_t is used at this stage to correct the belief distribution previously predicted $\omega(x_{t-1})$, and the weight of each particle is calculated. Finally, the posterior confidence of the robot is updated. A lkelihood range finder module [11] is adopted in this process to update the particle state based on the current Lidar data and the prior map. The principle of updating weights is expressed by Formula (16):

$$\begin{aligned} &x_t^{[\mathrm{M}]} \sim p(z_t|x_t, m), \\ &\omega(x_t) = \int p(x_{\mathrm{t}}^M|u_t, x_{t-1}, m)\omega(x_{t-1})\mathrm{d}x_{t-1}. \end{aligned} \tag{16}$$

3. Sampling. The sampling stage (line 11) still uses the classical KLD sampling to determine the size and distribution of the particle set. That is, the more dispersed the particles are, the more vacancies in grid map are filled, and the higher the M_x is [14]. The calculation of M_x is expressed by Formula (17):

$$M_x = \frac{k-1}{2\varepsilon}\{1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}}z_{1-\delta}\}^3, \tag{17}$$

where k is the number of non-empty particles, and $z_{1-\delta}$ is a constant of the $1-\delta$ quantile on the standard positive distribution. Using KLD sampling not only ensures the accuracy of location, but also avoids wasting computational resources.

In line 12, the particles are further optimized according to the actual experimental conditions. During the experiment, we found that the robot often localizes its position in the occupied grid and has a very large weight, indicating that it retains certain reliable information. Here, we will mark similar particles, and "filter" is not a true deletion—the corresponding pose information is just saved but not used.The particle's information will still be used in the next iteration updates.

4. Pose estimation. The particle with the largest weight is selected in each iteration. When the pose change of the particle with the largest weight is lower than the threshold set in this paper, and more than 10 consecutive position changes are within 0.05 m, it indicates that the particle representing the pose of the robot has converged. In this iteration, the K particles with the largest weight are selected, and the mean value of the poses is calculated. That is, the final estimated pose of the robot is (x, y), as in Formula (18):

$$(x, y) = \frac{1}{K}\sum\nolimits_{i=1}^{K}(x_i, y_i). \tag{18}$$

3. Experiments and Results

In this section, experiments are presented to evaluate the performance of the hybrid indoor localization system compared with other methods. The experiments were mainly designed for positioning efficiency and accuracy, where efficiency means less consumption of time under the same conditions, and accuracy means smaller error and higher success rate. First, the Wi-Fi fingerprinting algorithm KNNBP for rough localization was tested and then applied to precise localization. The precise localization was improved on the basis of AMCL that is widely used in robot localization. So, experiments comparing HPFL and AMCL were performed. These experiments were carried out in a long corridor in our laboratory with an area of 702 m^2. In addition, the performance of some localization methods obtained from references was compared with the methods presented in this paper.

3.1. Rough Localization

The Wi-Fi fingerprinting algorithm KNNBP was tested compared with KNN [38]. In this experiment, in order to evaluate the accuracy and efficiency of a location, 12 anchor points were selected to test the location error and time required. The time complexity of both KNNBP and KNN is $O(mn)$, where m is the number of APs, and n is the number of recorded positions. Because KNNBP eliminates invalid data, the computation time was slightly reduced, as illustrated in Figure 8b, and the positioning accuracy was greatly improved, as illustrated in Figure 8a.

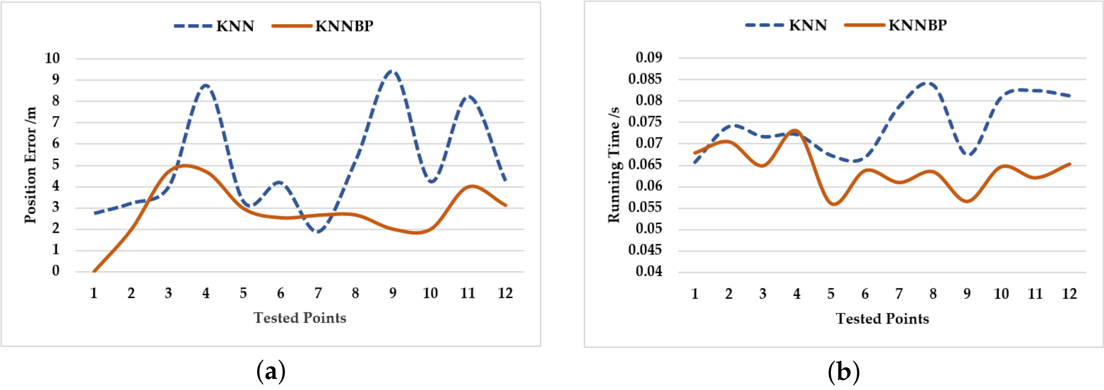


Figure 8. (**a**) Comparison of the positioning test between KNN and KNNBP. (**a**) Position error of rough localization according to KNN and KNNBP. (**b**) Running time for rough localization by KNN and KNNBP.

Average error (AE) and maximum error (ME) were also used to evaluate the performance of the two algorithms. The results are illustrated in Table 2. The AE and ME obtained by the KNNBP were approximately 50% of the AE and ME obtained by KNN. It can be seen that the accuracy of the robot's rough localization was greatly improved by KNNBP.

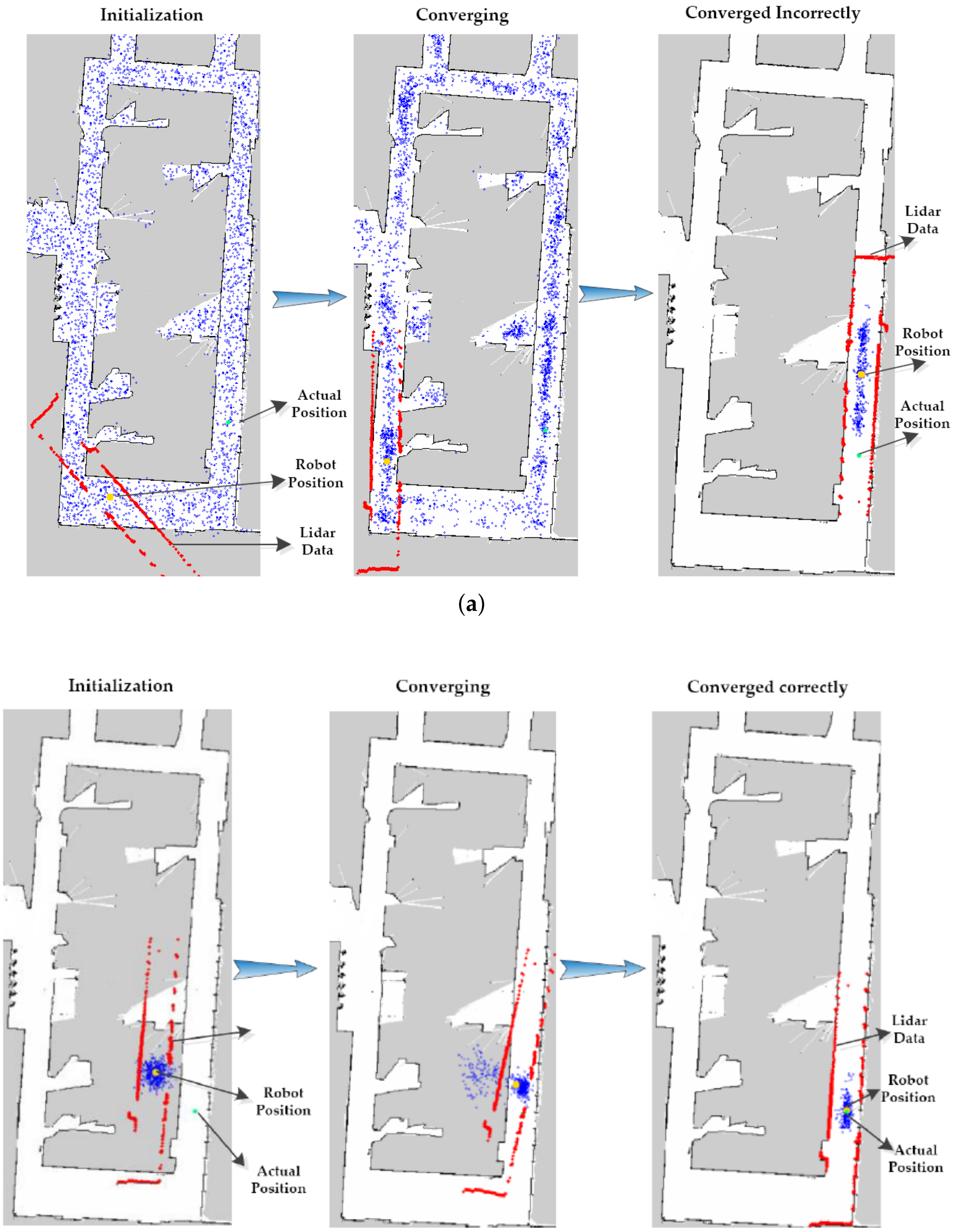
Table 2. Comparison of error evaluation. AE: average error; ME: maximum error.

Evaluation (m)	KNN	KNNBP
AE	4.95	2.6
ME	9.43	4.73

3.2. Precise Localization

Precise localization is mainly composed of two parts: global pose estimation and local pose tracking. The performance of HPFL was compared with the traditional localization algorithm AMCL [11]. When global positioning is triggered, the AMCL positioning adopts the global uniform particle scattering method in the global positioning as illustrated in Figure 9a, and then iteratively updates through particle filtering. The particle filter iterates continuously until the maximum weight of the particle is greater than the threshold. Finally, the particle converges to the proper location considered by the algorithm. The false convergence occurs even when the number of particles is 10,000. As is illustrated in Figure 9a, the Lidar data were not successfully matched with the map, so the pose estimation of the robot was wrong.

To prove the superiority of HPFL, a smaller number of particles was used in the experiments, as illustrated in Figure 9b. When the global positioning was performed by HPFL, the KNNBP-based Wi-Fi fingerprint location information and magnetic compass information were first integrated to provide the robot with a rough global pose, and the particles were scattered in the form of a Gaussian distribution near the pose. When the particles began to iterate, KLD sampling was used to limit the number of effective particles to improve the efficiency and ensure real-time localization. It can be seen that HPFL could ensure the accuracy of positioning even with a smaller number of particles. Finally, the Lidar data and map matched perfectly, so the robot acquired the correct pose estimation.



(**a**)

(**b**)

Figure 9. (**a**) The process of adaptive Monte Carlo localization (AMCL) with 10,000 particles was performed in a grid map with an area of 702 m^2. Particles were initialized by a global uniform distribution method, and they then converged incorrectly. (**b**) The process of HPFL with 3000 particles in the same grid map of the actual scene. Particles finally converged successfully, and the robot got the right position.

Furthermore, experiments were designed to verify the efficiency of AMCL and HPFL with different numbers of particles. From Figure 10a,b for AMCL, it can be seen that the more particles that were adopted, the greater the number of iterations and time consumed, but the success rate was too low when the number of particles was less than 6000, as illustrated in Figure 10c, and even though the number of particles was up to 10,000, incorrect localization still occurred at low probability. When

HPFL was used, the performance was better than the AMCL under the same conditions. Additionally, the success rate of localization by HPFL was 96%, even when the number of particles was 3000.

In addition, HPFL prevented the particles from stopping iterations when false convergence took place, which strengthened the robustness of the local position and enabled the robot to self-correct. During the experiment, due to the low accuracy of the encoder data, the positional deviation occurred after the robot reached the target. As illustrated in Figure 11a, the robot using the AMCL could not perform self-correction if it did not move, whereas the robot using HPFL could correct itself without moving. The calibration result is illustrated in Figure 11b.

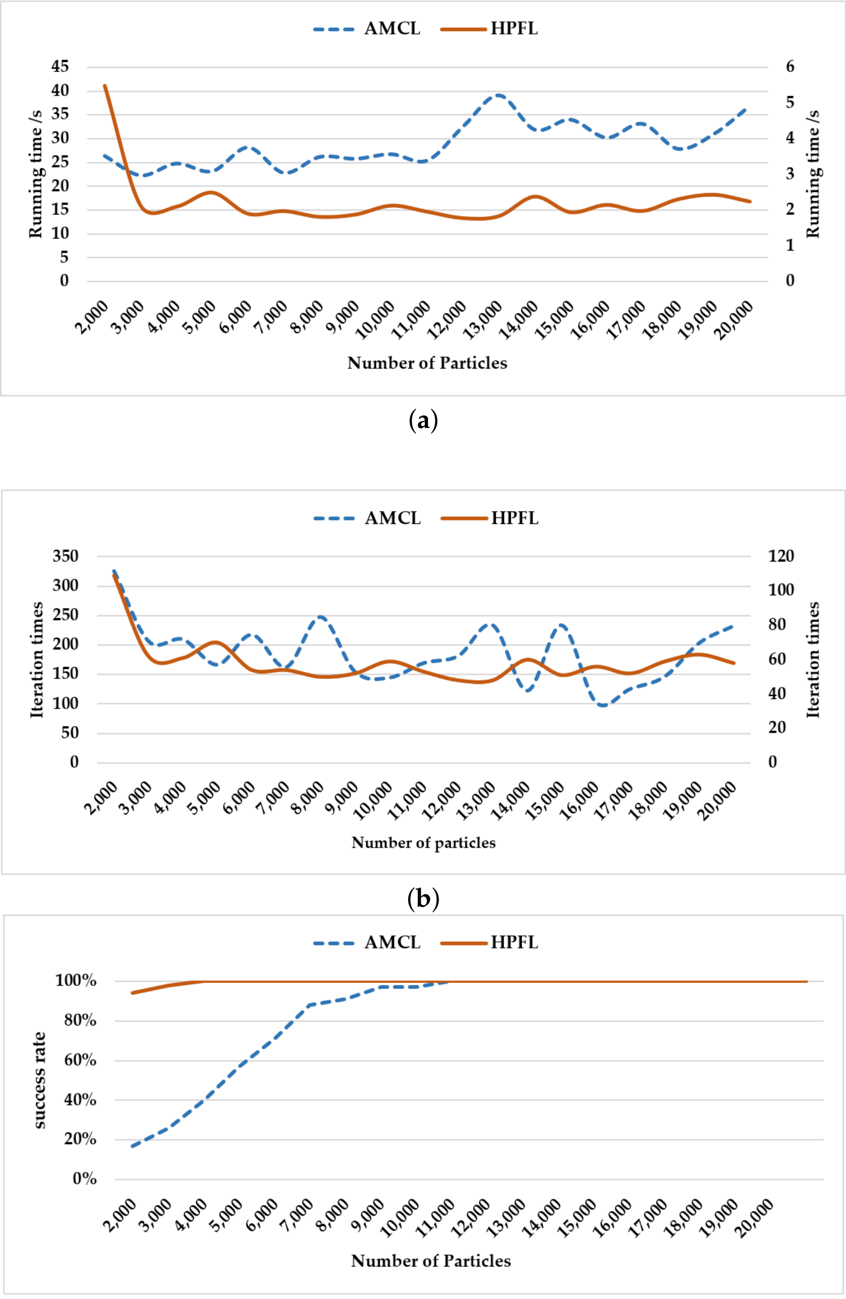


Figure 10. (**a**) Running time for AMCL and HPFL with different numbers of particles. The left *y*-axis is the running time for AMCL, and the right *y*-axis is the running time for HPFL. (**b**) Iteration times of AMCL and HPFL with different numbers of particles. The left *y*-axis is the number of iterations of AMCL, and the right *y*-axis is the number of iterations of HPFL. (**c**) Success rate of AMCL and HPFL with different numbers of particles.

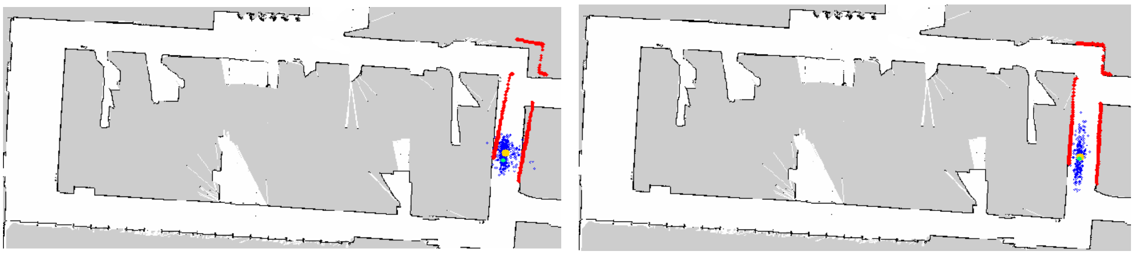


Figure 11. Self-correction of local pose tracking. (**a**) The position deviated when the robot rotated. (**b**) The robot corrected the pose automatically with HPFL.

Finally, the hybrid indoor localization system that adopts KNNBP and HPFL was compared with other indoor positioning methods in terms of positioning time and error. The results indicated that the comprehensive performance in global positioning efficiency and accuracy of the method presented in this study was better than those of other methods, as presented in Table 3.

Table 3. Performance of indoor localization methods.

Localization Methods	Time (s)	Error (m)	Processor
AMCL	25.6	0.05	Core i3
MOSR [18]	31	0.22	Core i5
KNNBP+HPFL	1.9	0.05	Core i3
WVT-bootstrap [33]	0.12	2	Core i5
Method in [24]	11	0.7	A laptop
FSPF [21]	20	0.1	Not public

4. Discussion

In this paper, we presented a hybrid indoor localization system that fuses multiple sensors in order for a robot to achieve autonomous navigation. We first used a KNNBP-based Wi-Fi fingerprinting algorithm to get a rough position, and then developed an HPFL algorithm to perform precise global positioning and local pose tracking for the robot. Experiments in a real-world situation indicated that the positioning error of our solution was 0.05 m at 96% with 3000 particles, the time required for global localization was down to 1.9 s, and the robustness of local pose tracking was improved. Therefore, the hybrid indoor location system in the present study performed better than the most widely used localization algorithm.

In the future, RGBD sensors will be added to the multi-sensor localization system to achieve semantic localization. If the robot is lost or kidnapped, Wi-Fi is not the only reference to provide rough position. Trained landmarks through deep learning by RGB images and point clouds can also help the robot to globally localize accurately, so that the robot can ensure the accuracy and stability of the positioning in highly crowded places.

5. Conclusions

The accuracy and efficiency of localization are very important for indoor robots, and our method helps to solve these problems. The hybrid indoor localization system proposed in this paper combines Wi-Fi fingerprinting with a particle filter to achieve remarkable results. To the best of our knowledge, the KNNBP-based Wi-Fi fingerprinting algorithm was first created in the present study, and the HPFL-based particle filter is an improvement on the basis of AMCL. Our localization system that fuses various sensors not only improves the global positioning but also improves the robustness of local position tracking. The experiments indicated that our positioning method was superior to the traditional location systems in both accuracy and efficiency.

Author Contributions: Y.S. and W.Z. designed the system; Q.H. conceived and designed the experiments; H.Z. and Z.C. performed the experiments; Z.Y. analyzed the data; M.L. performed supplementary experiments in Section 2.2; Z.L. analyzed the data of supplementary experiments in Section 2.2.

Funding: This work was financially supported by National Key R&D Program of China (Grant No. 2017YFB1302200).

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

References

1. Yi, D.H.; Lee, T.J.; Cho, D.D. A New Localization System for Indoor Service Robots in Low Luminance and Slippery Indoor Environment Using Afocal Optical Flow Sensor Based Sensor Fusion. *Sensors* **2018**, *18*, 171. [CrossRef] [PubMed]
2. Yassin, A.; Nasser, Y.; Awad, M.; Al-Dubai, A.; Liu, R.; Yuen, C.; Raulefs, R.; Aboutanios, E. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1327–1346. [CrossRef]
3. Choi, K.H.; Ra, W.-S.; Park, S.-Y.; Park, J.B. Robust Least Squares Approach to Passive Target Localization Using Ultrasonic Receiver Array. *IEEE Trans. Ind. Electron.* **2013**, *61*, 1993–2002. [CrossRef]
4. Yang, P.; Wu, W. Efficient Particle Filter Localization Algorithm in Dense Passive RFID Tag Environment. *IEEE Trans. Ind. Electron.* **2014**, *61*, 5641–5651. [CrossRef]
5. Li, Q.; Li, W.; Sun, W.; Li, J.; Liu, Z. Fingerprint and Assistant Nodes Based Wi-Fi Localization in Complex Indoor Environment. *IEEE Access* **2017**, *4*, 2993–3004. [CrossRef]
6. Blanco, J.L.; Galindo, C.; Ortiz-De-Galisteo, A.; Moreno, F.A. Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach. *Robot. Auton. Syst.* **2009**, *57*, 496–507.
7. Wu, T.; Liu, J.; Li, Z.; Liu, K.; Xu, B. Accurate Smartphone Indoor Visual Positioning Based on a High-Precision 3D Photorealistic Map. *Sensors* **2018**, *18*, 1974. [CrossRef] [PubMed]
8. Wang, Y.-T.; Peng, C.-C.; Ravankar, A.A.; Ravankar, A. A Single LiDAR-Based Feature Fusion Indoor Localization Algorithm. *Sensors* **2018**, *18*, 1294. [CrossRef] [PubMed]
9. Borenstein, J.; Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **2002**, *7*, 278–288. [CrossRef]
10. Burgard, W.; Fox, D.; Hennig, D.; Schmidt, T. Estimating the absolute position of a mobile robot using position probability grids. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, Portland, OR, USA, 4–8 August 1996; pp. 896–901.
11. Thrun, S.; Dieter, F. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005; pp. 250–263.
12. Lei, Z. Self-adaptive Monte Carlo localization for mobile robots using range finders. *Robotica* **2012**, *30*, 229–244.
13. Dellaert, F.; Fox, D.; Burgard, W.; Thrun, S. Monte Carlo localization for mobile robots. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 2, pp. 1322–1328.
14. Fox, D. KLD-Sampling: Adaptive particle filters. *Adv. Neural Inf. Process. Syst.* **2002**, *14*, 713–720.
15. Blanco, J.L.; González, J.; Fernandez-Madrigal, J.A. Optimal Filtering for Non-parametric Observation Models: Applications to Localization and SLAM. *Int. J. Robot. Res.* **2012**, *29*, 1726–1742. [CrossRef]
16. Thrun, S.; Fox, D.; Burgard, W. Monte Carlo Localization with Mixture Proposal Distribution. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, Pittsburgh, PA, USA, 30 July–3 August 2000; pp. 859–865.
17. He, X.; Aloi, D.N.; Li, J. Probabilistic Multi-Sensor Fusion Based Indoor Positioning System on a Mobile Device. *Sensors* **2015**, *15*, 31464–31481. [CrossRef] [PubMed]
18. Coltin, B.; Veloso, M. Multi-observation sensor resetting localization with ambiguous landmarks. *Auton. Robots* **2013**, *35*, 221–237. [CrossRef]
19. Pérez, L.; Íñigo, R.; Rodríguez, N.; Usamentiaga, R.; García, D.F. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. *Sensors* **2016**, *16*, 335. [CrossRef] [PubMed]
20. Choi, B.S.; Lee, J.W.; Lee, J.J.; Park, K.T. A Hierarchical Algorithm for Indoor Mobile Robot Localization Using RFID Sensor Fusion. *IEEE Trans. Ind. Electron.* **2011**, *58*, 2226–2235. [CrossRef]

21. Biswas, J.; Veloso, M. Depth camera based indoor mobile robot localization and navigation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; Volume 44, pp. 1697–1702.
22. Luca, D.G.; Alberto, M. Towards accurate indoor localization using ibeacons, fingerprinting and particle filtering. In Proceedings of the 2016 International Conference on IndoorPositioning and Indoor Navigation (IPIN), Madrid, Spain, 4–7 October 2016.
23. Canedo-Rodríguez, A.; Álvarez-Santos, V.; Regueiro, C.V.; Iglesias, R.; Barro, S.; Presedo, J. Particle filter robot localisation through robust fusion of laser, WiFi, compass, and a network of external cameras. *Inf. Fusion* **2016**, *27*, 170–188. [CrossRef]
24. Biswas, J.; Veloso, M. WiFi localization and navigation for autonomous indoor mobile robots. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 4379–4384.
25. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278
26. Cabrera-Mora, F.; Xiao, J. Preprocessing technique to signal strength data of wireless sensor network for real-time distance estimation. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1537–1542.
27. Subhan, F.; Ahmed, S.; Ashraf, K. Extended Gradient Predictor and Filter for smoothing RSSI. *Sensors* **2014**, *14*, 1198–1202.
28. Zhu, H.; Alsharari, T. An Improved RSSI-Based Positioning Method Using Sector Transmission Model and Distance Optimization Technique. *Sensors* **2018**, *18*, 171. [CrossRef]
29. Li, D.; Zhang, B.; Li, C. A Feature-Scaling-Based *k*-Nearest Neighbor Algorithm for Indoor Positioning Systems. *IEEE Internet Things J.* **2016**, *4*, 590–597. [CrossRef]
30. Alshami, I.H.; Ahmad, N.A.; Sahibuddin, S.; Firdaus, F. Adaptive Indoor Positioning Model Based on WLAN-Fingerprinting for Dynamic and Multi-Floor Environments. *Sensors* **2017**, *17*, 1789. [CrossRef] [PubMed]
31. Xie, Y.; Wang, Y.; Nallanathan, A.; Wang, L. An Improved K-Nearest-Neighbor Indoor Localization Method Based on Spearman Distance. *IEEE Signal Process. Lett.* **2016**, *23*, 351–355. [CrossRef]
32. Chapre, Y.; Mohapatra, P.; Jha, S.; Seneviratne, A. Received signal strength indicator and its analysis in a typical WLAN system (short paper). In Proceedings of the 2013 IEEE Conference on Local Computer Networks, Sydney, Australia, 21–24 October 2013; pp. 304–307.
33. Li, Z.; Braun, T. Passively track wifi users with an enhanced particle filter using power-based ranging. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 7305–7318. [CrossRef]
34. Chang, Q.; Li, Q.; Shi, Z.; Chen, W.; Wang, W. Scalable indoor localization via mobile crowdsourcing and gaussian process. *Sensors* **2016**, *16*, 381. [CrossRef] [PubMed]
35. Ghirmai, T. Distributed particle filter for target tracking: With reduced sensor communications. *Sensors* **2016**, *16*, 1454. [CrossRef] [PubMed]
36. Doucet, A.; Johansen, A.M. A tutorial on particle filtering and smoothing: Fifteen years later. *Handb. Nonlinear Filter.* **2011**, *12*, 656–704.
37. Malarvezhi, P.; Kumar, R. A diversity enhanced particle filter for carrier frequency offset estimation in nonlinear OFDM system. *Wirel. Pers. Commun.* **2016**, *89*, 15–26. [CrossRef]
38. Honkavirta, V. Location Fingerprinting Methods in Wireless Local Area Networks. Master's Thesis, Tampere University of Technology, Tampere, Finland, 2008; pp. 22–23.