A Reliability-Augmented Particle Filter for Magnetic Fingerprinting Based Indoor Localization on Smartphone

Hongwei Xie, Student Member, IEEE, Tao Gu, Senior Member, IEEE, Xianping Tao, Member, IEEE, Haibo Ye, Student Member, IEEE, and Jian Lu, Member, IEEE

Abstract—Using magnetic field data as fingerprints for smartphone indoor positioning has become popular in recent years. Particle filter is often used to improve accuracy. However, most of existing particle filter based approaches either are heavily affected by motion estimation errors, which result in unreliable systems, or impose strong restrictions on smartphone such as fixed phone orientation, which are not practical for real-life use. In this paper, we present a novel indoor positioning system for smartphones, which is built on our proposed reliability-augmented particle filter. We create several innovations on the motion model, the measurement model, and the resampling model to enhance the basic particle filter. To minimize errors in motion estimation and improve the robustness of the basic particle filter, we propose a dynamic step length estimation algorithm and a heuristic particle resampling algorithm. We use a hybrid measurement model, combining a new magnetic fingerprinting model and the existing magnitude fingerprinting model, to improve system performance, and importantly avoid calibrating magnetometers for different smartphones. In addition, we propose an adaptive sampling algorithm to reduce computation overhead, which in turn improves overall usability tremendously. Finally, we also analyze the "Kidnapped Robot Problem" and present a practical solution. We conduct comprehensive experimental studies, and the results show that our system achieves an accuracy of 1~2 m on average in a large building.

Index Terms-Indoor localization, magnetic, particle filter, smartphone

1 INTRODUCTION

INDOOR positioning using smartphones has attracted tremendous interests in recent years due to an increasing number of location-based applications that require accurate positioning or continuous tracking in buildings. The Wi-Fi fingerprinting based approach has become popular due to its simplicity and leveraging on the widely available Wi-Fi infrastructure. This approach does not require any specialized hardware or additional infrastructure support. However, tracking the location of a mobile user requires frequent Wi-Fi scanning which is known to be power-hungry for smartphones. In addition, the accuracy of Wi-Fi fingerprinting depends on a number of key factors including the number of Wi-Fi access points deployed, spatial differentiability, and temporal stability of the radio environment [1].

With the availability of more embedded sensors on smartphones, a new fingerprinting approach based on magnetometer has been proposed [2], [3], [4]. This approach is based on the hypothesis that the magnetic field in an indoor setting is non-uniform, and their fluctuations arise from both natural and man-made sources such as steel and

Manuscript received 8 Jan. 2015; revised 10 Aug. 2015; accepted 10 Sept. 2015. Date of publication 18 Sept. 2015; date of current version 29 June 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2015.2480064

concrete structures and electric systems. The abnormalities of magnetic field can be used as fingerprints for indoor localization. While this approach shares the same principle as Wi-Fi fingerprinting, it has several advantages [4], [5], [6], [7]: 1) independent from Wi-Fi infrastructure; 2) more reliable than Wi-Fi; 3) the change of magnetic field with location is quite significant, allowing for precise positioning; 4) more power efficient.

Different from Wi-Fi fingerprints that combine the radio signal strength from multiple access points, a magnetic fingerprint is basically a three-axis vector, consisting of the magnetic field readings along a smartphone's three axes (X, Y and Z). Studies [5], [6], [8] show that in a large indoor space the magnetic fingerprints may not be unique. As a result, large localization error may occur. Most of the existing work [2], [3], [9] leverage on Particle Filter (i.e., Monte Carlo Localization or MCL) to solve this problem and improve accuracy. The basic principle is to use random samples, also referred to as particles, i.e., hypothesis for the user's state such as position and heading direction, to represent the posterior distribution of a user's state. Particle filter recursively re-samples a set of particles according to a series of magnetic fingerprints and their spatial correlations to converge to the true state. The spatial correlations can be obtained by estimating user motion. Therefore, the performance of particle filter depends on: 1) motion estimation; 2) fingerprint measurement and the magnetic fingerprinting model; and 3) resampling.

We discover several limitations among existing particle filter based approaches. First, magnetometer readings are associated with phone orientation. When a phone changes

[•] H. Xie, X. Tao, H. Ye, and J. Lu are with the Department of Computer Science and Technology, the State Key Laboratory for Novel Software Technology, Nanjing University, China.

E-mail: {xhw, yhb}@smail.nju.edu.cn, {txp, lj}@nju.edu.cn.

T. Gu is with the School of Computer Science and Information Technology, RMIT University, Australia. E-mail: tao.gu@rmit.edu.au.

its orientation, we get different vectors. One may collect and store the magnetic readings of all directions at any location [6], [9], which obviously incurs high training cost, or use coordinate transformation [3], which can be error-prone in reality. But both measurement models used in existing particle filter based solutions [2], [3], [9] typically require a smartphone to head the same way as the user. This is not practical as we cannot expect smartphone users keep their smartphone direction the same all the time when moving and navigating in buildings. Moreover, different smartphones may have different types of magnetometers that require carefully sensor calibration before use. Magnetometer calibration is difficult considering a large number of smartphone users. Therefore, both better measurement and magnetic fingerprinting models are needed.

Second, existing motion estimation methods are errorprone. Particle filter is mostly used for tracking mobile robots [10], [11]. When applied to smartphone tracking, motion estimation usually incurs much more noise than robots, such as step miscounting, step length estimation error, or change of heading offset (i.e., the difference between user heading and phone heading). These errors may easily lead to localization failure.

Third, the computation overhead of particle filter is proportional to the number of particles we use, and it is believed that it should be larger than Wi-Fi fingerprinting. When we have a large number of smartphone users, we may see considerable computation demands on server. How to reduce computation cost is not a trivial task while keeping system accuracy.

Finally, the existing particle filter methods suffer from the well known "Kidnapped Robot Problem" [12], in which all particles may lost tracking of the target and hence recovery is not possible. In highly noisy and complex environments, these systems may easily run into such problem.

In this paper, we present a systematic approach to indoor positioning based on magnetic field fingerprinting, named MaLoc (**Ma**gnetic fingerprinting based indoor **Loc**alization). We propose a reliability-augmented particle filter, aiming to address the aforementioned issues and design a practical and accuracy indoor positioning system.

Firstly, we propose a hybrid measurement model to improve localization accuracy by combining the HV magnetic fingerprinting model with the magnetic density fingerprinting model. The HV model is obtained by extracting both the horizontal and vertical components of magnetic vector, which has the key features of being independent from phone orientation. We also explore the gradient of magnetic fingerprint to avoid calibrating different smartphone magnetometers. Secondly, we propose a novel motion estimation method covering step counting, step length estimation and user heading change estimation. This method is also independent from phone orientation. To minimize estimation errors and improve system robustness, we propose a dynamic step length estimation algorithm and a heuristic resampling algorithm. Specially, the heuristic resampling algorithm allows regular use of smartphone during localization such as answering a call or sending a message, whereas in basic particle filter it may arise heading offset change and introduce large errors into heading change estimation, and hence result in localization failure. Thirdly, to reduce computation overhead, we present an adaptive sampling algorithm, named Bundle Iterative Sampling (BIS), which is able to reduce sample size greatly while maintaining high localization accuracy and precision. Finally, we analyze different scenarios that may easily lead to localization failure and propose a practical solution.

In addition, we develop an efficient crowdsourcing tool to collect magnetic fingerprints, aiming to improve the existing time-consuming and labor-intensive training process [7].

MaLoc has many practical implications for real deployment such as no restriction on smartphone's orientation, being able to operate without intervening regular phone applications, and being scalable to a large number of users. In summary, this paper makes the following contributions.

- (1) We propose a reliability-augmented particle filter to improve system performance and robustness, which includes a dynamic step length estimation method to compensate the uncertainty of user's true step length, a heuristic particle resampling mechanism to solve the heading offset change problem.
- (2) We propose a novel hybrid measurement model to improve localization performance, which has the advantage of being independent from phone orientation. Moreover, it avoids calibrating different smartphone magnetometers.
- (3) We propose an adaptive sampling algorithm to reduce computation overhead, making the system be more scalable to a large number of users.
- (4) MaLoc is a systematic solution, which takes many practical problems (i.e., restrictions on smartphone's orientation and kidnapped robot problem) into consideration. Moreover, we conduct comprehensive field studies in a large building. The results show that MaLoc achieves an accuracy of 1~2 meters for various indoor settings and it is more power-efficient than Wi-Fi fingerprinting.

The paper is organized as follows. Section 2 describes the related work. Section 3 reviews the background and our empirical studies of magnetometer and indoor magnetic field. Section 4 presents the architecture of MaLoc. The motion model and the magnetic measurement model are introduced in Sections 5 and 6, respectively. Section 7 shows our bundle iterative sampling algorithm. Section 8 presents the analysis and solution for kidnapped robot problem. Section 9 describes the method for building the fingerprint database. Section 10 describes the experiments, and finally Section 11 concludes the paper.

2 RELATED WORK

Much work has been done in indoor localization over the past decade. Many systems are built upon special infrastructures such as infrared [13], acoustic [14] and bluetooth [15]. It is usually very costly deploying these infrastructures. Other systems leverage on existing, widely available infrastructures, such as GSM, Wi-Fi, and FM. Earlier Wi-Fi based approaches use a propagation model of receiving signal strength (RSS) from Wi-Fi access points (APs) to compute the receiver's location [16]. However, building such a model often requires the prior knowledge of AP positions and the accuracy much depends on the multi-path fading effect. Existing work such as [17] do not need to war-drive the entire space, but they require all the AP positions to be known. EZ [18] does not require any explicit pre-deployment effort of Wi-Fi APs, but it comes at the cost of loss of accuracy, in which the median localization error may be up to 7 m in a large building environment. The Wi-Fi fingerprinting based approach is becoming popular in recent years as it does not require any prior knowledge about APs, as well as it does not leverage on the propagation model. Using Wi-Fi fingerprinting, RADAR [19] and Horus [20] achieve 3-5 and 2 m localization accuracy, respectively. However, the Wi-Fi RSS is sensitive to human presence, and it may vary from time to time, limiting localization accuracy. PinLoc [1] improves the accuracy within 1 m by leveraging the physical layer but extensive profiling is required. FM fingerprinting is also feasible for indoor localization. Chen et al. [21] used FM signals, which is more energy efficient than Wi-Fi fingerprinting, but localization can only be done in room level. GSM based indoor localization methods [22] have also been proposed to work with a coarse precision.

Recent years, magnetic field has been explored as fingerprints for robot localization [23]. Haverinen and Kemppainen [2] proposed an indoor localization system using particle filter for both pedestrian and robot in the corridor of a building. Their system requires the user's heading must keep consistent with the corridor while walking. Chung and Donahoe [6] proposed a system which requires measuring the magnetic readings of all directions at any position and localization is done with an array of magnetometers. Their system achieves an accuracy of 4.7 meters. However, the training cost is quite high. Some work has been proposed using magnetometer on smartphones. LocateMe [4] investigated a magnetic model using smartphones, and localization can be achieved in room level. However, it is restricted in a 1D environment, such as corridor. Bilke [9] proposed a localization system which works in an 2D environment with the mean localization error of 4 meters; but its training cost is comparable to the work done in [6]. The system proposed in [3] also works in 2D environment and is able to avoid the measurements for all directions by coordinate transformation. However, it is error-prone because orientation estimation contains errors, which will be amplified by matrix transformation. No evidence in that paper was found that the system works in practise.

Particle filter has been used for mobile robot localization many year [10], [11]. The systems proposed in [2], [3], [9] are all based on particle filter, however they typically require the smartphone's heading to be the same as the user all the time. Some existing work such as [24], [25], [26] also integrate Wi-Fi fingerprinting with particle filter to improve accuracy, or they combine both Wi-Fi fingerprints and magnetic fingerprints [27]. The main problem of using inertial sensors (pedestrian dead-reckoning) for localization [28], [29] is that small errors in sensing may be magnified by integration [30]. By introducing particle filter or other methods (i.e., kalman filter) for sensor fusion, the errors in measuring inertial sensors can be filtered out or minimized.



Fig. 1. Locus of magnetic readings with the optimum ellipsoidal fit superimposed.

While we leverage on particle filter in this work, we have no restriction on phone orientation during localization. Mobile users are able to use their smartphones as usual. To achieve this, we propose a reliability-augmented particle filter that requires both the motion estimation and the magnetic fingerprinting models must be independent from phone orientation. To this extent, MaLoc is more practical than existing magnetometer based localization systems.

3 MAGNETIC FIELD AND MAGNETOMETER

In this section, we review the background of the geomagnetic field and magnetometer on smartphones, and conduct preliminary empirical studies.

3.1 Characteristics of Magnetometer on Smartphone

The magnetic field value B_p is measured by a smartphone's magnetometer. B_e is the magnetic field vector at the same location in earth coordinate system, which is combined with the geomagnetic field and the magnetic field from the environment. B_p is obtained by the phone rotated in yaw ψ , pitch θ and roll ϕ from B_e , respectively. Its relationship with B_e is defined as Equation (2) in absence of noise [31]

$$\boldsymbol{B}_{p} = \boldsymbol{R}_{x}(\boldsymbol{\phi})\boldsymbol{R}_{y}(\boldsymbol{\theta})\boldsymbol{R}_{z}(\boldsymbol{\psi})\boldsymbol{B}_{e}, \qquad (1)$$

$$B_e = B_{geomagnetic} + B_{environment}, \tag{2}$$

where $R_z(\psi)$, $R_y(\theta)$, $R_x(\phi)$ are the corresponding rotation matrices. When we include the noise: hard-iron effect *V* and soft-iron effect *W*, B_p is defined as Equation (3) [31]:

$$B_p = WR_x(\phi)R_y(\theta)R_z(\psi)B_e + V.$$
(3)

The hard-iron effect V is an offset vector and the soft-iron effect W is a matrix. Because of noise, when the smartphone is rotated at a fixed point, the locus of the magnetic reading will be an ellipsoid, as shown in Fig. 1. If the magnetometer is perfectly calibrated, the locus will be closed to the sphere. Calibrating the magnetometer is mainly to find the hard-iron effect V and the soft-iron effect W. Besides the basic calibration method in [31], there are also various of calibration techniques have been proposed, such as [32], [33], [34].

Authorized licensed use limited to: RMIT University Library. Downloaded on January 09,2021 at 15:32:44 UTC from IEEE Xplore. Restrictions apply.



Fig. 2. Magnetic readings captured by two smartphones along a 50-meter corridor.

In addition, some smartphones may have been calibrated in hardware or system level.

Different smartphones may have different types of magnetometers that vary in their sensitivities. Even the same type of smartphone may read different magnetic values at the same location. Fig. 2 shows the magnetic readings captured by two Galaxy Nexus smartphones along a 50-meters long corridor. Their readings fluctuate at the same location, but the gradients of these two curves remain quite constant. This suggests that we can use the gradients of fingerprints instead of raw fingerprint values to avoid calibrating different magnetometers.

3.2 Characteristics of Magnetic Field

The indoor magnetic field combines the geomagnetic field and the fields from ferromagnetic objects. We are interested to study the factors which may have influence on the magnetic field readings. Fig. 3 shows the magnetic density map captured by a smartphone in a 16.3 m \times 5.4 m indoor area over one month. As we can see, magnetic field readings are quite stable over time, but changed significantly with locations. We also study the magnetic field readings when a smartphone is placed in different height, as shown in Fig. 4. The magnetic readings are collected in a height of 0.5, 1 and 1.5 m from the floor, respectively, along a corridor. Considering errors in the location mapping, the differences are not as significant as the horizontal differences, and their gradients are still very similar. In addition, previous studies [4], [5], [6], [7] also demonstrate that the effects caused by objects carried by the user, other pedestrians and the rearrangement of furniture are very limited. From these studies, we confirm the feasibility of using magnetometer on smartphone for indoor localization.

4 MALOC ARCHITECTURE

From system view, MaLoc consists of a client and a server. The client runs on smartphones to collect sensor data (i.e., velocity acceleration, angular velocity, and magnetic field reading). It then performs step counting, computes heading change $\Delta\theta$ between two contiguous steps, and accesses magnetic values in each step. The preprocessed data will be sent to a server where an augmented particle filter, which operates dedicatedly for each user, performs computation and sends updated locations to each user.

The key component of MaLoc is an augmented particle filter we propose. The particle filter uses a set of particles to estimate the posterior distribution of a system-state



Fig. 3. Magnetic field map captured by smartphones over one month.

conditioned on measurements. The system-state or userstate here is the user's position and heading:

$$\boldsymbol{s} = (x, y, \theta), \tag{4}$$

where x, y represents the user's position and θ is the user's heading direction. A particle is a hypothesis for the user's state with a weight:

$$\chi_i = < \boldsymbol{s}_i, w_i >, \tag{5}$$

where w_i is the weight of the particle and its value is often evaluated by a probability model P(z|s). P(z|s) is the probability of observing z on state s, where z is the observation value. We use magnetic fingerprints as the main observation values z. A higher weight means it is more close to the true state, and the posterior distribution of user's state can be represented by a set of particles.

The basic particle filter contains three essential components: the motion model, the measurement model and the resampling model, as presented in Algorithm 1. The motion model updates each particle's state by estimating user's motion leveraged on the inertial sensors. The measurement model then re-evaluates the particles' weights. Finally, the remaining particles refined by resampling will be more close to the true system state. As a rule, with the recursive operations of the three processes, the prediction for true state will become more and more accurate. Essentially, the performance of a particle filter based method completely depends on how these three models construct.

In MaLoc, shown in Algorithm 2, *N* particles are randomly sampled from the initial area given by a coarsegrained localization method. The loop is controlled by step



Fig. 4. Magnetic readings captured by smartphone with different heights.

counting. For each step a user moves, we will update the particles and prediction.

Algorithm 1. Procedure of Basic Particle Filter

Input: $S_t = \{\chi_i | i = 1, 2, ..., N\}$ representing the belief at time *t*. **Output:** S_{t+1} representing the belief at time t + 1.

- for each particle do 1:
- Update state s_i^{t+1} using control information 2: (motion model).
- Compute the weight w_i^{t+1} based on likelihood of 3: observation value z_{t+1} (measurement model).
- 4: $w_i^{t+1} = P(z^{t+1}|s_i^{t+1})$
- 5: end for
- 6: Resample N particles from old particles according to the discrete distribution given by their weights.

Algorithm 2. Procedure of MaLoc

1: state=LOCALIZING

- 2: Generate N random particles from an initial area.
- 3: for each step do
- Estimate *l* and $\Delta \theta$. 4:
- 5: for each particle do
- 6: Update position and heading by motion model as Equations (6) and (7).
- 7: Evaluate the weight of particles by measurement model as Equation (9).
- 8: end for
- 9: Decide the amount of particles for each resampling method N_d and N_h . ($N = N_d + N_h$)
- 10: Resample N_h particles by our heuristic resampling algorithm.
- 11: Resample $N_{BIS}(<=N_d)$ particles form old particles according to the distribution given by their weights. The amount N_{BIS} is decided by our Bundle Iterative Sampling Algorithm.
- 12: Normalize the weights.
- 13: Predict the user's state by Equation (10).
- 14: if Kidnapped-prone scenarios detected then

15:	state=ABORT_LOCALIZATION
16:	while state==ABORT_LOCALIZATION do
17:	if Get out of kidnapped-prone scenarios then
18:	Restart particle filter.
19:	end if
20:	end while

- end if 21: 22:
- end for

We construct the motion model as Equation (6) and (7):

$$\theta_i^{t+1} = \theta_i^t + \Delta\theta + G_\theta, \tag{6}$$

$$\begin{bmatrix} x_i^{t+1} \\ y_i^{t+1} \end{bmatrix} = \begin{bmatrix} x_i^t \\ y_i^t \end{bmatrix} + \begin{bmatrix} \cos(\theta_i^{t+1}) \\ \sin(\theta_i^{t+1}) \end{bmatrix} \times (l+G_l), \tag{7}$$

where *l* is the step length, $\Delta \theta$ is the user's heading changes between two consecutive steps, G_l and G_{θ} are Gaussian noise. Different from existing work [2], [3], [9], [35], the step length *l* is not constant in MaLoc, and it is estimated dynamically during localization. The step length is different from one user to another, and even one person's step length may be changed. Inaccurate step length estimation will lead to large localization errors, resulting in localization failure, which has been shown in the experiment in Section 10.2. We propose a dynamic step length estimation algorithm to effectively solve these uncertainties. The algorithm also makes MaLoc error tolerant in step counting. $\Delta \theta$ is obtained by fusing the compass and the gyroscope sensor on smartphones. Generally, $G_l \sim N(0, \sigma_l)$ and $G_{\theta} \sim N(0, \sigma_{\theta})$ are used to enlarge the diversity of particles.

Since we use magnetic fingerprint as observation value, then P(z|s) will depend on the intensity of changes of magnetic field at that location and on the age of the map, which difficult to obtain in reality. So the Gaussian pseudodistribution [2], [3], [9] are often used, as presented in Equation (8):

$$P(z|s) = \frac{1}{(2\pi)^{n/2} |V|^{1/2}} \\ exp\left\{-\frac{1}{2}[z - obv(s)]^T V^{-1}[z - obv(s)]\right\},$$
(8)

where *n* is the dimension of *z*, *V* is the covariance, obv(s) is a function to get the observation value of state *s* in the fingerprint database.

In practice, obv(s) will return the fingerprint which is nearest to s geographically. For fast obtaining the nearest fingerprint, a hierarchical data structure has been designed. We split the fingerprint database into grid according to its location. We then build a hash function to map the grid with the set of fingerprints belongs to this grid. We also have a function grid(s) which finds the right grid state sbelongs to in O(1) time. Therefore, when we search for the nearest fingerprint of s in database, we first find its corresponding grid, and then search the nearest fingerprint in this grid. The total search cost is O(m) + O(1), where m is the size of a grid. Compared with the K-D tree, this approach is easier to maintain.

As different smartphone's magnetic reading have bias, we evaluate each particle by Equation (9) instead of using a Bayes filter model $w_i^{t+1} = w_i^t P(z^{t+1} | s_i^{t+1})$ or directly use $w_i^{t+1} =$ $P(z^{t+1}|s_i^{t+1})$ like others. In this way, MaLoc is more stable, and importantly it avoids calibrating different magnetometers

$$\begin{split} w_i^{t+1} &= P\left(z^{t+1} - z^t | s_i^t, s_i^{t+1}\right) \\ &= \frac{1}{(2\pi)^{n/2} |V|^{1/2}} exp\left\{-\frac{1}{2}\left[\left(z^{t+1} - z^t\right) \right. \\ &\left. - \left(obv(s_i^{t+1}) - obv(s_i^t)\right)\right]^T V^{-1}\left[\left(z^{t+1} - z^t\right) \right. \\ &\left. - \left(obv(s_i^{t+1}) - obv(s_i^t)\right)\right]^2\right\}. \end{split}$$
(9)

Resampling is to get rid of the particles with low weights, which contribute almost nothing to prediction, and put resources on particles that are close to the true state with higher probability. In this work, we resample new particles from last iteration particles according to the discrete probability distribution generated by their weights (i.e., a traditional resampling method), and propose a Bundle Iterative Sampling algorithm to adapt the sample size. However, this may result in loss of particle diversity and localization failure



Fig. 5. The magnitude of acceleration readings captured by a smartphone carried in different positions.

when noise (e.g., change of heading offset) is introduced. To address this, we propose a heuristic resampling method to work together with the traditional resampling method. Finally, we estimate the true state with the weighted average of the current particles, as shown in Equation (10):

$$\hat{\boldsymbol{s}} = \sum_{i=1}^{N} \boldsymbol{s}_i \cdot \boldsymbol{w}_i, \tag{10}$$

where the weight w_i has been normalized.

As we can see, the current particles are generated from the last round of localization. If an error results in most of the particles close to the true state being wiped out, it may end up with wrong prediction in the following rounds of localization. To address this issue, we detect the scenarios (named kidnapped-prone scenarios) which are easily lead to localization failure. If we find one, we will then abort localization, and restart the particle filter.

5 MOTION MODEL

In this section, we introduce the methods for step counting, dynamic step length estimation, and heading change estimation. We also present the heuristic resampling algorithm along with heading changes estimation. The proposed augmented particle filter with dynamic step estimation and heuristic resampling will enhance the robustness of the existing particle filter.

5.1 Steps Counting

We use accelerometer on smartphone for step counting. Since a user may carry smartphone in any orientation, we only extract the magnitude of the accelerometer reading. The mean filter is used to smooth the raw acceleration data. As shown in Fig. 5, the acceleration values are captured by a smartphone carried in different ways, e.g., in hand, in the shirt pocket, and in the front pant pocket. The result shows that counting the number of peaks or valleys may not work for all cases. There are also many step counting algorithms in the literature [30], [36], which achieved hight accuracy. In this work, we use a simple one for lower computation overhead. We set four thresholds, T_{acc}^{u} , T_{acc}^{d} and T_{time}^{u} , T_{time}^{d} , to filter false cases. T_{acc}^{u} and T_{acc}^{d} are the upper bound and lower bound of the acceleration variance between two adjacent peak and valley. T_{time}^{u} and T_{time}^d are the upper bound and lower bound of the time interval between them. In this way, step counting achieves better accuracy.

5.2 Dynamic Step Length Estimation

Simply, one can set a constant step length for each user. However, this is not practical as people's step lengths may vary widely. Even the step length of the same user may be changed from time to time. Since the precision of estimating a user's step length has a large impact on localization accuracy, we propose a dynamic step length estimation algorithm. It is based on this idea that when the particles are updated by the motion model, every particle evolve with different step lengths. The distribution of all particle's step lengths is $D_l \sim N(l, \sigma_l)$. Therefore, the average $l_{ave} =$ $\frac{1}{N}\sum_{i=1}^{N} l_i$ will be very close to *l*. However, after resampling, all the particles will be re-evaluated and refined. A particle with a higher weight is more close to the true state. On the other hand, this particle's step length is also closer to the true step length. The weighted average step length after resampling is defined as Equation (11):

$$l_{weighted} = \sum_{i=1}^{N_d} l_i \cdot w_i, \tag{11}$$

where N_d is the number of particles sampled from old particles (See Algorithm 2), because only these parts of particles evolve from the old particles by taking a step. We then conclude that if the particle filter converges in the right way, $l_{weighted}$ will tend to be the true step length. Certainly, if the particle filter cannot converge correctly, this theory may not be guaranteed. Based on this theory, we design a dynamic step length estimation algorithm. Instead of just using $l_{weighted}$ as step length, we design a queue Q size of Q_{size} to cache the most recent l_{weight} . Their average value is used as the estimation of step length for the next round of localization. In practise, the user's initial step length l_0 is set according to his weight and height, and we consider merging it with the average weighted step length

$$l_e = \alpha \cdot \frac{1}{Q_{size}} \sum_{i=1}^{Q_{size}} l^i_{weighted} + (1-\alpha)l_0.$$
(12)

In this way, we can prevent the accidental bad converge result from influencing the estimation. Considering when the user's heading changes, his step length may have some changes, resulting in uncertain influence on the step length estimation. In our algorithm, we will not put this part into the queue. Compared with the step length estimation methods in [28], [30] who update step length model only at the turning, our algorithm can update step length in real time and does not depend on the accuracy of turning detection.

5.3 Heading Change Estimation and Change of Heading Offset

Determining the heading of a user is difficult, even the compass sensor is precise in indoor environments since the user's heading may be different from phone heading (a.k.a. heading offset). In this work, we assume the initial orientation of the user is unknown, which may not be the same with the phone, and eventually we use the augmented particle filter to obtain the user's heading. The heading change between two consecutive steps $\Delta\theta$ is required by the motion model. It is reasonable that we assume the phone heading is



Fig. 6. The angle changes captured by compass and gyroscope when walking along a straight line and turning at a 90-degree corner.

relatively stable with the user when walking. Certainly, the assumption cannot hold all the time as the user may use the mobile phone for other tasks during localization, such as receiving a phone call and sending a message. These interference motions may change the heading offset and bring errors into heading change estimation. Therefore, there are two issues here: (1) how to estimate the phone's heading change between two consecutive steps regardless of its orientation and position; (2) how to eliminate the error introduced by the change of heading offset.

Compass and gyroscope on smartphone can be used to estimate phone's heading change between two adjacent steps. Since compass may be affected by some ferromagnetic objects in indoor environments, we also investigate the performance of gyroscope on smartphone. The gyroscope on smartphone measures the angular velocity around the phone's x-, y- and z-axis, respectively. Usually, gyroscope readings are typically integrated over time to calculate phone's rotation, i.e., rotation angle change. However, we cannot get the heading changes directly as the phone's orientation is unknown. A feasible method is to make use of the gravity sensor on smartphone to obtain angle change around the gravity direction, which is the same as heading change. Fig. 6 compares the performance of heading change estimation using both compass and gyroscope. It depicts the angle changes from the start point when walking along a straight line and a 90-degree corner. As we can see from the figure, gyroscope measurements are more stable than that of compass when walking along a straight line or turning normally. However, when turning fast, errors in the gravity sensor will result in large accumulated errors of gyroscope. Therefore, a wise method is to use a Kalman filter to fuse it with compass when the gyroscope shows the current heading change is little or relatively slow. In this way, we can avoid the interference of ferromagnetic objects on compass measurements. However, when gyroscope shows a fast or heavily heading change, we can only count on compass.

The change of heading offset has certainly negative impact on traditional particle filter based methods. For example, in the case that the user answers a phone call while walking, the user's heading keeps the same but the phone's heading may be changed 180 degrees. As a result, all particles reverse their headings and then get lost in the trap. The localization error will become bigger and bigger, and eventually result in localization failure. Therefore, detecting large heading change of the phone is risky (it may be done by the user's turning or the change of heading offset). We propose a heuristic algorithm to solve this problem, inspired by the random particle filter [12] which randomly samples new particles when the total non-normalized weight decreases. Our algorithm heuristically samples parts of new particles in the resampling phase instead of only sampling from the old particles (See Algorithm 2). How to generate new particles and how many particles are related to the phone's heading changes of this step. We have the following rules.

(1) The number of new particles N_h is proportional to $\Delta \theta$. To implement it in MaLoc, we present Equation (13):

$$N_h = \frac{\Delta \theta \cdot p}{\pi} N, N_h \le N, \tag{13}$$

where p is the parameter to control the sensitivity (usually chosen from [0.5, 0.85] in our experiments), and N is the total number of particles.

- (2) The position of new particles are randomly sampled from the *r*-meter range of the previous predicted location (*x̂*, *ŷ*). *r* can be a constant value or a variable related to Δθ. For simplicity, we set it as a constant value in our experiments.
- (3) The initial heading of new particles are also randomly sampled from the range [θ̂, θ̂ + 2Δθ] or [θ̂+ 2Δθ, θ̂]. θ̂ is the previous predicted heading.

The heuristic resampling algorithm can also increase the diversity of particles, which will make the particle filter more robust and tolerate more motion estimation errors, with the cost of decreasing the localization precision.

6 MAGNETIC MEASUREMENT MODEL

The magnetic measurement model evaluates particles based on the magnetic field observations captured from smartphones. As mentioned before, a magnetometer reading consists of a three-dimensional vector $B_p = (B_x, B_y, B_z)$, representing the magnetic value along the phone's *x*-, *y*-, and *z*-axis, respectively. Generally, there are three ways of using B_p as observation values.

Using B_p directly as observation z [2], [3], [6], [9]. One method is to collect the magnetic field readings of all directions at any location, which not only increases the training cost rapidly, but also reduces the accuracy as the sample space becomes larger. Alternatively, we can estimate phone orientation and transform B_p to the earth coordinate system B_e . However, this is error-prone because orientation estimation usually contains errors and these errors will be amplified on B_e . Both of these two methods have a serious problem that they require the phone's heading to be the same as the user's heading during localization. Because in the measurement phase, each particle obtains the reference fingerprint $B_{x,y,\theta}$ at a location from the database according to its position (x, y) and heading θ (not phone's heading θ_p due to inaccurate compass readings). As θ represents user's heading, if it is not the same as the phone's heading θ_p , the magnetic field reading on smartphone cannot map with any particle's observation $B_{x,y,\theta}$ as it should be mapped to B_{x,y,θ_p} . This may result in localization failure. Therefore, using this type of observation on smartphone is not practical.



Fig. 7. Magnetic field map of the horizontal component (left) and vertical component (right) in a 16.3 m x 5.4 m indoor area.

Using the magnitude *B* of B_p as observation [2], [4]. *B* is a rotation invariant scalar quantity and quite stable. However, the elements in each fingerprint will drop from three to one, reducing the uniqueness of each fingerprint. In large indoor environments, the particle filter may need more time to converge to the right location.

Extracting the horizontal component B_h and vertical component B_v of B_p as observation [7]. The gravity sensor on smartphone provides us the direction of gravity (i.e., the vertical direction). We can extract both the vertical and horizontal components of B_p and construct a new observation value (B_h, B_v) (named HV fingerprint). Fig. 7 shows the magnetic field map of these two components in an indoor area. Obviously, it is more unique than the magnitude fingerprint, which will make the particle filter converge to the right location faster. This fingerprint concept was first mentioned in [7], but there is no real-world system using this model. That is because when a user stands still, the gravity sensor reading is very precise. However, when the user moves, noise will be introduced, resulting in decreasing in accuracy or even ending up with failure.

In MaLoc, we combine both *B* and (B_h, B_v) using a hybrid measurement model. The HV fingerprint has the advantage of more uniqueness, and it makes the particle filter converge to the right location faster. However, it is not as stable as the magnetic magnitude fingerprint. Therefore, we use the HV fingerprint to accelerate the convergence of MaLoc when localization starts, and then we switch it to the magnetic magnitude fingerprint for "tracking".

Since calibrating the offset of different phones' magnetometers is very costly, we just use the variance of the fingerprint value between two consecutive steps (the gradient) instead of using the fingerprint itself to avoid calibrating different magnetometers. As we mentioned before, it is also less sensitive with the changes of magnetic field in the vertical direction and the changes along with time. This is one of the reasons that we use Equation (9) to evaluate the particles. Another one is that although using the Bayes filter model to evaluate the particles will make particle filter converge more quickly and good for exposing localization failure, it also makes the particle filter become very sensitive to noise and may fail easily.

In a very large indoor environment, these fingerprint models or measurement models may not be suitable for localization because the particle filter will need too many particles and a long period of time to converge, incurring high computational overhead and reducing the usability. To address this issue, we leverage on a simply coarse-grained localization method [3], [6] (with $10\sim20$ m or room level accuracy) which narrows the searching space in the initial phase only.

7 BUNDLE ITERATIVE SAMPLING ALGORITHM

As the computation overhead of particle filter is proportional to the number of particles, reducing the number of particles will obviously improve the computation efficiency of particle filter. However, when we decrease the number of particles, the stability of particle filter may degrade too. Clearly, there exists a tradeoff between the computation overhead and the stability or robustness of particle filter. As we discussed in Section 4, a promising solution to reduce computation overhead is to dynamically adjust the number of particles. This is so called adaptive particle filter, and two main adaptive sampling algorithms exist in the literature. The first one is likelihood-based adaptation [10]. The idea is when the non-normalized sum of likelihood (i.e., importance weights) less than a pre-specified threshold, the sampling will stop. However, this approach may not work well in symmetric environments or environments with high likelihood (i.e., different positions have a similar observation value). Hence, this algorithm cannot be applied to magnetic fingerprinting based localization. The other algorithm is KLD-Sampling [37], [38] which uses the Kullback-Leibler distance (KL-Distance) to determine the sample size. The key idea of KLD-sampling is to bound the error introduced by the sample-based belief representation. At each iteration, this approach generates samples until their number is large enough to guarantee that the KL-distance between the maximum likelihood estimate and the underlying posterior does not exceed a pre-specified bound ϵ [39]. Nevertheless, the KLD-Sampling algorithm needs to compute the quantity of the chi-square distribution to check if it is the time to stop sampling in each iteration, hence it involves extra computation overhead. In addition, the KL-distance bound ϵ is not a readily comprehensible variable that is easy to choose. In MaLoc, we propose a more efficient and comprehensible adaptive sampling algorithm—Bundle Iterative Sampling.

The basic principle of BIS is described as follows. The reason why we need so many particles is that we intend to cover as many as possible state spaces. When re-sampling particles one by one, if we find that sampling more particles will not cover new state space, it is not necessary to continue sampling. The procedure of BIS is listed in Algorithm 3. First, we reset all bins to "unoccupied". Here, bins are implemented as a three dimensional grid (i.e., 0.5 m* $0.5 \text{ m} * 45^{\circ}$)[40]. We will then sample particles bundle by bundle. One bundle has BUNDLE_SIZE number of particles. We can sample no more than N_d particles, i.e., at most N_d /BUNDLE_SIZE bundles. In each bundle, we sample particles from old particles according to their weights. For each new sampled particle, we check if it occupies a new bin. We define that a new bin is occupied when it contains at least OCCUPY_THRESHOLD (i.e., 2) particles. In addition, the coordinates of these occupied bins will be



 $\texttt{BUNDLE_SIZE=50, WINDOW_SIZE=4, BINS_INC_THRESHOLD=4, bin=0.5m\times0.5m\times45^\circ}$

Fig. 8. An example of Bundle Iterative Sampling algorithm.

recorded. Then, we can only reset the occupied bins to unoccupied instead of iterating all bins and setting them to unoccupied in the first step. This can reduce some computation overhead as the number of occupied bins is often much less than the number of bins, i.e., often two order of magnitude less. After resampling a bundle of particles, we count how many new bins are occupied. We can now compute the number of new occupied bins BINS INC in a sliding window with a length of WINDOW_SIZE bundles. If it is less than a certain threshold BINS INC THRESHOLD, that means sampling more particles will not bring more bins and we should stop sampling. Fig. 8 shows an example of how the BIS algorithm works. At the initial phase, we can see sampling new particles will always create new bins. However, after particles converge, several bundles of particles may only occupy one new bin, which indicates sampling can stop.

Algorithm 3. Bundle Iterative Sampling Algorithm			
1:	Reset bins to unoccupied.		
2:	$MAX_BN = N_d / BUNDLE_SIZE$		
3:	k = 0		
4:	for $BN = 1:MAX_BN do$		
5:	currentBinNum = k		
6:	for $i = 1$:BUNDLE_SIZE do		
7:	Sample a particle P_j from the discrete distribution		
	given by their weights		
8:	if P_j occupy a new bin then		
9:	k = k + 1		
10:	end if		
11:	end for		
12:	bundle[BN] = k - currentBinNum		
13:	if $BN > = WINDOW_SIZE$ then		
14:	for $i = 0$:WINDOW_SIZE-1 do		
15:	$BINS_INC + = bundle[BN-i]$		
16:	end for		
17:	if BINS_INC <= BINS_INC_THRESHOLD then		
18:	break;		
19:	end if		
20:	end if		
21:	end for		

Now, we show that BIS is able to achieve comparable precision and accuracy to a fixed-size particle filter. That means we have to show particles share similar distribution in both BIS and fixed-size sampling algorithm. Generally, the accuracy of particle filter depends on the state space size that particles can cover. This is obviously guaranteed in BIS because sampling stops when particles cannot occupied more bins. And the precision is related to the space distribution of particles (i.e., the convergence degree of particles). Assume when BIS stops, it has sampled $N_{BIS}(<=N_d)$

particles. At this time, sampling more particles will not occupy new bins. Note that the space distribution of these N_{BIS} particles is p_0 . After sampling N_{BIS} particles (S_0) , we continue sampling another N_{BIS} samples (S_1) (note that its distribution is p_1). S_0 and S_1 are sampled from the same discrete distribution, and they all fall in the same bins (saturated), implying that their distributions are quite similar, denoted as $Distance(p_0, p_1) < \epsilon$, where Distance(p, q) measures the difference between the two probability distributions p and q as follows:

$$Distance(p,q) = \sum_{x} |p(x) - q(x)|.$$
(14)

If we sample N_d particles (the distribution is p_d), we need to prove $Distance(p_0, p_d) < \epsilon$, which is shown as follows. We sample $k = \lceil \frac{N_d}{N_{BIS}} \rceil$ particle sets with a size of N_{BIS} and merge them to get a new distribution $p_{0,1,\dots,k-1}$. Then we have $Distance(p_0, p_d) \approx Distance(p_0, p_{0,1,\dots,k-1})$, as $k \cdot N_{BIS} \ge N_d$.

$$Distance(p_{0}, p_{0,1,\dots,k-1}) = \sum_{x} |p_{0}(x) - p_{0,1,\dots,k-1}(x)|$$

$$= \sum_{x} \left| p_{0}(x) - \frac{\sum_{i=0}^{k-1} p_{i}(x)}{k} \right|$$

$$= \sum_{x} \left| \frac{\sum_{i=1}^{k-1} (p_{0}(x) - p_{i}(x))}{k} \right|$$

$$< = \frac{1}{k} \sum_{x} \sum_{i=1}^{k-1} |p_{0}(x) - p_{i}(x))|$$

$$= \frac{k-1}{k} \epsilon.$$
(15)

This implies that sampling N_{BIS} particles shares a similar distribution with sampling N_d particles. Therefore, BIS achieves comparable precision with fixed-size particle filter.

8 KIDNAPPED ROBOT PROBLEM

When applying particle filter for localization in reality, we may have the following problem. If there exist much noise or many errors in sensor data, the particles which are close to the true state may be wiped out during the resampling phase and the particle filter turns "stranded". This is often called "Kidnapped Robot Problem", i.e., particle filter is difficult to recover, especially in a complex environment. When all particles are put to wrong locations, their nonnormalized weights may not be down to zero immediately as some wrong particles' observations may be very similar to observed magnetic readings. Even at last, the particle filter realizes that none of the particle's observations is within a reasonable range, the time it takes for that may be unacceptably long.

Many approaches have been proposed to address this problem in robot self-localization. 1) A simple but effective method is to add a small number of uniformly distributed, random samples after each update [10]. An improved method is named sensor resetting [12], [41], i.e., the number of samples to add depends on how badly the current set of samples seems to represent the location. However, the



Fig. 9. Magnetic readings when putting phone in pant pocket and holding in hand.

observation environment of magnetic based indoor localization is much more complex and with high likelihood, we will need many random samples, which is not practical. 2) In mixture MCL [11], a reverse particle filter is proposed. It introduces new samples according to observation first and then weights them according to the existing probability distribution over pose. This does not fit MaLoc for the same reason—too many samples are needed. 3) In [42], the particle filter is restarted when the particle cloud diverges over a threshold from the observations. This method has shown an improvement of 36 percent in localization accuracy. This method fits MaLoc well. However, the thresholds in this method are hard to decide, especially in a complex environment.

In MaLoc, we propose an approach to "solve" the kidnapped robot problem in magnetic based indoor localization scenarios from a difference perspective. From our experiments, we first find the scenarios in which localization often failed when using a traditional particle filter: 1) In the beginning process of localization using a small sample set; 2) Suddenly shaking or moving the phone results in large heading offset change; 3) Putting phone in pant pocket. 4) Shaking phone in hand heavily while walking or running.

The first case can be handled by our adaptive sampling method. We can use more particles at the initial phase and then our bundle iterative sampling algorithm dynamically decides how many particles it require in the following updates. The second one has been solved by our heuristic sampling algorithm. The more challenging problem lies on the last two scenarios (i.e., kidnapped-prone scenarios). After comprehensively analyzing the sensor data collected in these two scenarios, we have the following obversations. Shaking phone in hand or in pant pocket will not only result in continuously heading offset change, but also will introduce much noise in magnetic readings because magnetometer on smartphone has not been calibrated perfectly and many smartphones come with poor quality magnetometer. Fig. 9 shows the magnetic readings collected on a 12-meter corridor using a Nexus 3 and Nexus 5, respectively. For each phone, we collect the data in two scenarios – 1) putting in the front pant pocket and 2) holding in hand steadily at the same height of pant pocket. As we can see, when putting Nexus 3 in pant pocket, the magnetic reading will be affected too. At this time, the heading information and magnetic readings both contain too much noise, which is out of the tolerance range of particle filter. Therefore, our strategy is to abort localization when we discover the kidnapped-prone scenarios and restart the particle filter. Detecting kidnapped-prone



Fig. 10. Heading changes measured by gyroscope.

scenarios is easy. In both of these scenarios, the heading measured by gyroscope will present quickly and periodically changes, as shown in Fig. 10. Then we can use an algorithm similar to step counting to detect this pattern.

9 MAPPING THE MAGNETIC FIELD

There are many of magnetic mapping methods [43] or SLAM methods [44] proposed for robot localization based on magnetic field. In this section, we use crowdsourcing to enable fast and low-cost collection of the magnetic field fingerprints using smartphones. This method is inspired by IndoorAtlas [45]. As shown in Fig. 11, the data collector requires walking along these lines to cover an area. On each line, we set the starting point, the heading direction and the path length. We then walk along this path with a slow and constant speed. In this process, the tool application will record data captured by the magnetic sensor on smartphone. A magnetic fingerprint collected will be stored as $\langle B_h, B_v \rangle$ and the magnitude can be computed from it. We simply assume the data collector walking at a constant speed. Each magnetic fingerprint can then be mapped with its location easily. To avoid large mapping errors, each line should not be too long. On each line, we collect intensive magnetic fingerprints, e.g., one fingerprint every 0.1 meter. The distance between the two lines is about 0.6 meter in our experiment. We then fill the fingerprints into this area by interpolation. Eventually, we obtain a magnetic fingerprint in every 0.1 m \times 0.1 m square.

10 EVALUATION

We conducted extensive experimental studies in a large building. Fig. 12 shows the floor plan of the building. Our



Fig. 11. Fingerprints collection approach (left) and Fingerprint collection APP GUI (right).



Fig. 12. Performance of two particle filters with different initial step lengths.

study involves seven volunteers who are mainly students in a local university. Several smartphone models are used, including Google Nexus, Samsung Galaxy, HTC, and Moto. The MaLoc software is pre-installed into these smartphones for collecting user traces. We modify and attach an ink stamper on each volunteer's shoes. While a user walks each step, it will leave a stamp (e.g., an ink mark) on the floor surface. In this way, we obtain all the traces and the ground truth. To avoid more tedious ground truth measurement, in our trace collection, all the volunteers are required to follow these routines step by step. We collect traces over a period of several months. For our experiments, we first focus on evaluating individual techniques we proposed: 1) How much step counting errors can MaLoc tolerate; 2) How well the dynamic step length estimation performs; 3) How well the heuristic resampling algorithm performs; 4) Performance comparison of particle filters with different types of magnetic fingerprinting models; 5) How well the adaptive sampling algorithm performs. We then evaluate the overall performance of MaLoc, including the overall localization accuracy in different scenes or different conditions, the localization accuracy with different walking paths, and the energy consumption in comparison to the Wi-Fi fingerprinting based localization approach.

10.1 Effects of Errors in Step Counting

In the first experiment, we evaluate the influence of step counting error. Fig. 13 shows both localization errors of the traditional particle filter and our augmented particle filter (without adaptive sampling algorithm) in presence of random miscounting steps. The user walks about 300 steps. The traditional particle filter functions correctly until miscounting reaches 50. However, our augmented particle filter



Fig. 13. Localization results of two particle filters when miscounting occurs.

tolerates more miscounting steps, i.e., 80 steps. It is not only because of the heuristic resampling method increases the diversity of particles, but also dynamic step length estimation can compensate miscounting steps. In this experiment, the true step length of the user is about $0.6 \sim 0.65$ m. The average estimated step length measured by our augmented particle filter is 0.68 m when miscounting is 10 steps, but it increases to 0.77 m when miscounting goes up to 50 steps.

10.2 Performance of Dynamic Step Length Estimation

Next, we test our proposed dynamic step length estimation. In this experiment, we set σ_l to 0.2 and Q_{size} to 5. The user's true step length is about 0.6~0.65 m. Fig. 14 compares our augmented particle filter (without heuristic resampling and adaptive sampling algorithm) with the traditional one for localization errors when we set the step length to different values (i.e., 0.5, 0.6, 0.7 and 0.8 m, respectively). As shown in the figure, when the step length deviates much from the true step length in the traditional particle filter, we observe poor accuracy. The traditional particle filter fails to localize when the step length is set to 0.8m. However, our augmented particle filter with dynamic step length estimation is not so sensitive to the initial step length. The experiment implies that although errors exist in the initial step length, we still achieve good accuracy. In addition, we find that the average step length of the localization process is around 0.67 m, no matter what the initial step length is. This implies, after a period, we can infer the user's average step length. However, our dynamic step length estimation algorithm may fail when the initial step length deviates from the true step length too much. We hence set the average step length as the initial value in MaLoc, which works well in practice.



Fig. 14. Performance of two particle filters with different initial step lengths.



Fig. 15. Localization results of two particle filters with heading offset change.

10.3 Performance of Heuristic Resampling Algorithm

We evaluate the heuristic resampling algorithm from two aspects: processing the error of heading change estimation and processing the occasional change of the heading offset. First, we run the particle filter 100 times with precise step counting and a constant step length, respectively. The traditional particle filter fails to localize 4 times as there are errors in heading change estimation causing localization failure. In contrast, our particle filter (only integrate with the heuristic resampling algorithm) localizes all correctly (i.e., 100 times), demonstrating its advantage in robustness compared to the traditional particle filter. Second, we run an experiment to verify the ability of processing the occasional change of heading offset. We use a trace in which the user walks along a 24-meter corridor and answers a phone call while walking. We compare the traditional particle filter and our augmented particle filter for localizing his position. The results are presented in Fig. 15. We observe from the result that the heading offset change only temporarily influence the localization accuracy of MaLoc, but it causes the traditional particle filter converge to wrong locations.

10.4 Effects of Different Types of Observation Values

To compare the performance of different types of magnetic observation values, we conduct an experiment in a hall, in which the traditional particle filter will need a longer period to converge. The user held his smartphone in hand and walked 69 steps in total. He started from a hall, walked about 27 steps and then turned left into a corridor. Fig. 16 depicts the localization results of MaLoc using three different fingerprinting models as observation. As we observe from the figure, the HV fingerprint is sensitive to the user's



Fig. 16. The performance of MaLoc using different types of observations.



Fig. 17. The changes of particle numbers during localization.

turning motion but it has a faster convergence rate. Using the magnetic magnitude fingerprint is more robust, but it has a lower convergence rate than the HV fingerprint does. This shows clearly that the hybrid model certainly takes advantages of both models.

10.5 Evaluation of Bundle Iterative Sampling

We now evaluate the influence of different parameters on the performance of BIS. The key parameter in BIS is BINS_ INC_THRESHOLD. In this experiment, we set BINS_INC_ THRESHOLD to 4 (dash line), 6 (concrete line) and 8 (dotdash line), respectively, and other parameters are set as follows. Max Particles = 2,000, BIN_SIZE = $0.5 \text{ m} \times 0.5 \text{ m} \times 45^\circ$, BUNDLE_SIZE = 50, WINDOW_SIZE = 4, OCCUPY_ THRESHOLD = 2. Fig. 17 shows the sample size (include heuristic sampled particles) during a typical localization using BIS. The result shows that the average sample size of each running is 822, 683, and 562, respectively. We observe that a larger BINS_INC_THRESHOLD results in a smaller sample size.

To test the efficiency of BIS, we conduct the following experiment. A user walks about 300 steps, and we then observe the execution time while varying the maximum number of particles from 1,000 to 3,000. This experiment runs on a ThinkPad notebook (X1 Carbon), with an i7-2 GHz CPU and 8 G memory. The result is shown in Fig. 18. As we can see, the execution time of BIS(BINS_INC_THRESHOLD=4) is almost not affected by the maximum particle number, and it always maintains high efficiency. This is because BIS strategically samples particles that are just enough to cover the current possible state space. In most of the cases, it does not need so many particles like the traditional particle filter. BIS is only sensitive to parameters such as BINS_INC_THRESHOLD, and BUNDLE_SIZE.

While BIS works efficiently, we are also interested in knowing if BIS has any negative effect on localization accuracy. Fig. 19 shows the localization accuracy comparison between fixed-size particle filter and BIS particle filter set with different BINS_INC_THRESHOLD (i.e., 4 and 6, respectively). We observe that BIS particle filter achieves similar accuracy to the fixed-size version when the maximum particle number is large enough. However, when the maximum particle number is too small, both of them may decrease in localization accuracy. In addition, if BIN-S_INC_THRESHOLD is set too large, it may easily run into localization failure. Since the efficiency of BIS particle filter is almost not affected by the maximum particle number, in practice, we set a slightly higher maximum particle number



Fig. 18. The execution time of fixed-size particle filter and BIS-particle filter with different maximum particle numbers.



Fig. 19. The average localization accuracy of BIS and fixed-size sampling with different maximum particle number.

TABLE 1 Different Dividing Method

Dividing Method	Average Sample Size
10×20	828
20×10	837
50×4	887
100×2	931
200×1	1,070

to make it being able to survive in extremely noisy environments with a high probability.

We now discuss the effect of different combination of BUNDLE_SIZE and WINDOW_SIZE. Basically, we only need to count how many new bins are occupied when sampling BUNDLE SIZE×WINDOW SIZE (denoted as WIN-DOW_SAMPLE_SIZE) particles. The reason why we split a sample set into bundles and count new bins in a sliding window instead of counting WINDOW_SAMPLE_SIZE particles directly is to ensure reducing sample size both effectively and efficiently. In this experiment, we set BUND-LE_SIZE×WINDOW_SIZE to 200, and evaluate different dividing methods, i.e., $10 \times 20, 20 \times 10, 50 \times 4, 100 \times 2$, and 200×1 . The maximum sample size is set to 5,000 and BIN-S_INC_THRESHOLD is set to 4. From the result shown in Table 1, we observe that a smaller BUNDLE_SIZE makes sampling stop in good time. But if BUNDLE_SIZE is too small (e.g., 1), it results in a bigger WINDOW_SIZE, which may bring more computation overhead as indicated in Algorithm 3 Lines 13-19. According to our experience, setting BUNDLE SIZE to a value between 20 and 100 yields good performance.

Fig. 20 shows how the average sample size changes with the bin size. BINS_INC_THRESHOLD changes from 1 to 8. It



Fig. 20. Average sample size changes with bin size.



Fig. 21. Localization accuracy with corresponding bin size and BINS_INC_THRESHOLD.

is not surprising to observe that for fixed BINS_INC_THRES-HOLD, the sample size increases with smaller bins. This is because the same set of particles can easily occupy more small bins. But a bigger bin with a big BINS_INC_THRES-HOLD may result in localization failure. Fig. 21 illustrates the average localization accuracy during a typical localization case. For fixed BINS_INC_THRESHOLD, localization algorithm runs 10 times. It shows that with a bin size of $1 \text{ m} \times 1 \text{ m} \times 90^\circ$, MaLoc may be vulnerable to failure when BINS_INC_THRESHOLD goes larger than 1. Therefore, in practice, we first choose a reasonable bin size (i.e., $0.5 \text{ m} \times 45^\circ$), then adjust other parameters to achieve the best performance.

10.6 Overall Performance of MaLoc

We first evaluate MaLoc in different scenes: the hall, the conference room, the office area (mainly corridors) and the library, as shown in Fig. 22. The localization results are shown in Fig. 23. From the figure, we observe that while MaLoc performs well in general, the performance in the conference room, the office area, and the library is much better than that in the hall. One observation is that the floor plan in each of these three scenes is relatively complex than that in the hall. This bring many physical constrains, which will limits the possible behaviors of the user and improve localization accuracy. In addition, more complex building construction may lead to serious magnetic distortion which implies better magnetic fingerprints produced. For further investigation, we collect two different straight trajectories in the hall, one is near to the wall and the pillars (Path A), the other one is in the middle of the hall (Path B), as shown in Fig. 12. In Fig. 24, we can see that the magnetic reading on Path A fluctuations much more than Path B. As a result, MaLoc performs more accurately in Path A than Path B.



Fig. 22. Different scenes. (a) The hall in floor 1. (b) A conference room. (c) The whole corridor of floor 2. (d) A book shelf area in Library.



Fig. 23. Performance of MaLoc in different scenes.

We now move to evaluate how noise in motion estimation and magnetic measurement affect localization accuracy and precision. We conduct six experiments with a user holding his smartphone with four different postures, including: 1) holding the phone at a height of 0.6 m by hand while keeping the same direction as his heading; 2) holding the phone at a height of 1 m by hand; 3) holding the phone at a height of 1.4 m by hand; 4) putting it in his coat pocket; 5) shaking it slightly left and right; 6) holding by hand and swinging along with his arm slightly while walking. We then compare MaLoc with the Wi-Fi fingerprinting based method, as shown in Fig. 25. The results show that holding by hand yields the most accurate result because it introduces less noise in both motion estimation and magnetic measurement (magnetic fingerprints are collect at the height of 1 m). And the height of the phone has little impact on localization accuracy. As mentioned in Section 3, magnetic reading at a fix point's different height shows little difference and the phone is hold on hand in the first three experiments, the result can illustrate little magnetic measurement noise just have limited influence on localization accuracy. Shaking the phone while walking or putting the phone in pocket will result in decreasing of accuracy and precision. It implies that the more noise introduced during magnetic measurement (phone' orientation change will also have some influence on magnetic measurement) and motion estimation, the greater decreased in localization accuracy.

We also build a simple Wi-Fi fingerprinting based localization system. When user walking along a line to collect magnetic fingerprints, it will collect Wi-Fi fingerprints



Fig. 24. The effect of magnetic fluctuations.



Fig. 25. Performance of MaLoc in different conditions and performance comparison with Wi-Fi fingerprinting.



Fig. 26. Performance of MaLoc in different trajectories.

simultaneously (scanning period is 3 seconds). Then we use KNN algorithm to localize. In our Wi-Fi fingerprint database, the average number of AP in each Wi-Fi fingerprint is 6.46, which is quite dense. The 50 and 80 percent errors of the Wi-Fi fingerprinting based approach is 2.5 and 4.5 m, respectively, and the average localization accuracy is 3.5 m. However, when holding the phone in hand at the height of 1m, the average accuracy of MaLoc with 50 percent errors and 80 percent errors is 0.8, 1.6, and 1.0 m, respectively. Even in the worst case (i.e., coat pocket), the average accuracy of MaLoc is 2.0 m which is also much better than Wi-Fi fingerprinting.

To evaluate the performance of MaLoc in different trajectories, we select three typical trajectories (as shown in Fig. 12): a straight line in corridor (Path 1), a trajectory with a corner in an open area (Path 2), and a rectangle trajectory in an open area (Path 3). Fig. 26 depicts the magnetic readings and the localization results when a user walks along these trajectories. The magnetic readings on these three trajectories fluctuates comparably, but they have different the convergence rate and localization precision. It shows that converging in a corridor (i.e., Path 1) seems faster than the other two paths because of the physical limits of the map and any turning actions of the user temporally decreases localization precision as new random particles are sampled.



Fig. 27. Energy efficiency of MaLoc, compared with Wi-Fi fingerprinting.

In the last experiment, we evaluate the energy consumption of MaLoc. To save energy, MaLoc only performs limited computation on smartphone such as collecting sensor data and preprocessing, step counting, and heading change estimation, and leaves heavy computation to server. We conduct this experiment using a Samsung Galaxy Nexus smartphone, and compare the energy consumption in three scenarios: running nothing, running trace collection application of MaLoc only, and running the Wi-Fi scanning application only. All of the sensors' sampling frequency in MaLoc is set to SENSOR DELAY NORMAL and the frequency of Wi-Fi scanning is set to 2 seconds. To get rid of the influence of other applications, as some applications may have background data, we close the network communication module and don't let the two localization application send requests to Server for localization in both experiments. So the energy consuming only compare the sensing and preprocessing part of MaLoc with Wi-Fi scanning. We think it is fare as both of MaLoc and Wi-Fi fingerprinting technology require requesting to Server for localization. Moreover, their request frequency and request data size are comparable (MaLoc only send a request for localization when detect at least 2-3 steps). To prevent smartphone from entering the hibernating mode, we keep the smartphone's screen active during our experiment. Fig. 27 shows the energy consumption in these three scenarios. From the result, we observe that running MaLoc trace collection application saves about 1 hour battery life than running the Wi-Fi scanning application. The average current consumed when running MaLoc is about 220 mA. Thus, we conclude that MaLoc saves about 220 mA·h energy than Wi-Fi scanning within about 6 hours.

11 CONCLUSION

In conclusion, this paper presents a novel indoor localization system named MaLoc. It utilizes magnetic sensor data and inertial sensor data on smartphones by a reliability-augmented particle filter. The most important feature of MaLoc is that it does not impose any restriction on smartphone's orientation, and users are free to use their phones in whatever ways they like during localization. We propose a hybrid magnetic measurement model, which improves the performance of MaLoc and avoids calibrating different magnetometers. Moreover, we propose a set of novel techniques to improve MaLoc's robustness and usability. Through comprehensive experiments based on the traces collected in a large building, we demonstrate that MaLoc is more accurate and energy efficient than the Wi-Fi fingerprinting based technique. The experimental study in this paper has a limited scope since we focus on a university building only. We plan to further develop MaLoc and make it available for both Google Play and Apple Store, and collect public user traces in a variety of indoor spaces such as shopping malls, hotels, and commercial office buildings for more extensive evaluations. We also plan to crowd-source magnetic fingerprints leveraging on the public, and further develop the crowdsourcing method to ensure the quality of magnetic fingerprints since magnetic fingerprints should be collected more tenuously than Wi-Fi fingerprints.

ACKNOWLEDGMENTS

This work was supported by the NSFC under Grants 61373011, 91318301, 61272456, 61472312, and the Foundation for Innovative Research Groups under Grant 61321491.

REFERENCES

- S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the Mona Lisa: Spot localization using PHY layer information," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 183–196.
- pp. 183–196.
 [2] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robot. Auton. Syst.*, vol. 57, no. 10, pp. 1028–1035, 2009.
- [3] E. Le Grand and S. Thrun, "3-Axis magnetic field mapping and fusion for indoor localization," in *Proc. IEEE Conf. Multisensor Fusion Integr. Intell. Syst.*, 2012, pp. 358–364.
- [4] K. P. Subbu, B. Gozick, and R. Dantu, "Locateme: Magnetic-fieldsbased indoor localization using smartphones," ACM Trans. Intell. Syst. Technol., vol. 4, no. 4, p. 73, 2013.
- [5] M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson, "Characterization of the indoor magnetic field for applications in localization and mapping," in *Proc. Int. Conf. Indoor Positioning Indoor Navigation*, 2012, pp. 1–9.
- [6] J. Chung and Donahoe, "Indoor location sensing using geo-magnetism," in Proc. 9th Annu. Int. Conf. Mobile Syst., Appl. Serv., 2011, pp. 141–154.
- [7] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *Proc. Int. Conf. Indoor Positioning Indoor Navigation*, 2012, pp. 1–9.
- Int. Conf. Indoor Positioning Indoor Navigation, 2012, pp. 1–9.
 [8] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in Proc. Ubiquitous Positioning Indoor Navigation Location Based Service, 2010, pp. 1–10.
- [9] A. Bilke and J. Sieck, "Using the magnetic field for indoor localisation on a mobile phone," in *Location-Based Services*. New York, NY, USA: Springer, 2013, pp. 195–208.
- [10] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in Proc. 16th Nat. Conf. Artif. Intell. 11th Innovative Appl. Artif. Intell. Conf. Innovative Appl. Artif. Intell., 1999, pp. 343–349.
- [11] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artif Intell.*, vol. 128, no. 1, pp. 99–141, 2001.
- pp. 99–141, 2001.[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
- [13] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91–102, 1992.
 [14] A. Ward, A. Jones, and A. Hopper, "A new location technique for
- [14] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *Pers. Commun.*, vol. 4, no. 5, pp. 42–47, 1997.
- [15] J. Hallberg, M. Nilsson, and K. Synnes, "Positioning with bluetooth," in Proc. 10th Int. Conf. Telecommun., 2003, vol. 2, pp. 954–958.
- [16] D. Madigan, E. Einahrawy, R. P. Martin, W.-H. Ju, P. Krishnan, and A. Krishnakumar, "Bayesian indoor positioning systems," in *Proc. IEEE INFOCOM*, 2005, vol. 2, pp. 1217–1227.
- [17] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 483–494.
- *Comput. Netw.*, 2014, pp. 483–494.
 [18] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

- [19] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, 2000, vol. 2, pp. 775–784.
- [20] M. Youssef and A. Agrawala, "The horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst. Appl. Serv.*, 2005, pp. 205–218.
- [21] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in Proc. 10th Int. Conf. Mobile Syst. Appl. Serv., 2012, pp. 169–182.
- [22] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, and V. Otsason, "GSM indoor localization," *Pervasive Mobile Comput.*, vol. 3, no. 6, pp. 698–720, 2007.
- [23] S. Suksakulchai, S. Thongchai, D. Wilkes, and K. Kawamura, "Mobile robot localization using an electronic compass for corridor environment," in *Proc. IEEE Int. Conf. Syst. Man, Cybern.*, 2000, vol. 5, pp. 3354–3359.
- [24] F. Evennou and F. Marx, "Advanced integration of wifi and inertial navigation systems for indoor mobile positioning," *Eur. J. Appl. Signal Process.*, vol. 2006, pp. 164–164, 2006.
- [25] H. Nurminen, A. Ristimaki, S. Ali-Loytty, and R. Piché, "Particle filter and smoother for indoor localization," in Proc. Int. Conf. Indoor Positioning Indoor Navigation, 2013, pp. 1–10.
- [26] F. Hong, Y. Zhang, Z. Zhang, M. Wei, Y. Feng, and Z. Guo, "WaP: Indoor localization and tracking using wifi-assisted particle filter," in *Proc. IEEE 39th Conf. Local Comput. Netw.*, 2014, pp. 210– 217.
- [27] C. Zhang, J. Luo, and J. Wu, "MaWi: A hybrid magnetic and wi-fi system for scalable indoor localization," in *Proc. 13th Int. Symp. Inf. Process. Sens. Net.*, 2014, pp. 275–276.
- [28] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 421– 430.
- [29] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, 2008, pp. 114–123.
- [30] A. Rai, K. K. Chintalapudi, and Padmanabhan, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.
- [31] T. Ózyagcilar, "Calibrating an ecompass in the presence of hard and soft-iron interference," *Freescale Semiconductor Ltd.*, 2012.
- [32] J. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, "Geometric approach to strapdown magnetometer calibration in sensor frame," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1293–1306, Apr. 2011.
- [33] V. Renaudin, M. H. Afzal, and G. Lachapelle, "Complete triaxis magnetometer calibration in the magnetic domain," J. Sens, vol. 2010, 2010.
- [34] D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson, "Calibration of strapdown magnetometers in magnetic field domain," *J. Aerosp. Eng.*, vol. 19, no. 2, pp. 87–102, 2006.
 [35] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo locali-
- [35] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, vol. 2, pp. 1322–1328.
- [36] N. Z. Naqvi, A. Kumar, A. Chauhan, and K. Sahni, "Step counting using smartphone-based accelerometer," Int. J. Comput. Sci. Eng., vol. 4, no. 5, pp. 675–681, 2012.
- [37] D. Fox, "Kld-sampling: Adaptive particle filters," in Proc. Adv. Neural Inf. Process. Syst., 2001, pp. 713–720.
- [38] D. Fox, "Adapting the sample size in particle filters through kld-sampling," *Int. J. Robot. Res.*, vol. 22, no. 12, pp. 985–1003, 2003.
 [39] G. Teng, K. Zheng, and W. Dong, "Adapting mobile beacon-
- [39] G. Teng, K. Zheng, and W. Dong, "Adapting mobile beaconassisted localization in wireless sensor networks," *Sensors*, vol. 9, pp. 2760–2779, 2009.
- [40] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," J. Artif. Intell. Res., vol. 11, pp. 391–427, 1999.
- [41] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Sensor resetting localization for poorly modeled mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, vol. 1.1, pp. 454–459.
 [42] B. Turgut and R. P. Martin, "Restarting particle filters: An
- approach to improve the performance of dynamic indoor localization," in *Proc. IEEE Global Telecommun. Conf.*, 2009, pp. 1–7.
 D. Natarro and C. Paret "Magnetic and a configuration of the second seco
- [43] D. Navarro and G. Benet, "Magnetic map building for mobile robot localization purpose," in *Proc. IEEE Conf. Emerging Technol.* & Factory Autom., 2009, pp. 1–4.

- [44] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning, "Simultaneous localization and mapping using ambient magnetic field," in *Proc. IEEE Conf. Multisensor Fusion Integr. Intell. Syst.*, 2010, pp. 14–19.
 [45] I. College J. A. M. Market, M. M. Ma
- [45] [Online]. Available: https://www.indooratlas.com/, 2014.



Hongwei Xie received the BS degree from the Computer and Information Department, Hehai University, China, in 2012. He is currently working toward the master's degree from the Computer Science and Technology Department, Nanjing University, China. His research interests include mobile computing, machine learning, and indoor localization system. He is a student member of the IEEE.



Tao Gu received the bachelor's degree from Huazhong University of Science and Technology, the MSc degree from Nanyang Technological University, Singapore, and the PhD degree in computer science from National University of Singapore. He is an associate professor in the School of Computer Science and Information Technology, RMIT University. His current research interests include mobile and pervasive computing, wireless sensor networks, distributed network systems, sensor data analytics, cyber

physical system, Internet of Things, and online social networks. He is a senior member of the IEEE and a member of the ACM.



Xianping Tao received the PhD degree in computer science from Nanjing University in 2001. He is currently a professor in the Department of Computer Science, Nanjing University. His research interests include software agents, middleware systems, Internetware methodology, and pervasive computing. He is a member of the IEEE.



Haibo Ye received the MSc degree in computer science from Nanjing University in 2012. He is currently working toward the PhD degree in the Department of Computer Science at Nanjing University. His research interests include indoor localization, mobile, and pervasive computing. He is a student member of the IEEE.



Jian Lu received the PhD degree in computer science from Nanjing University in 1988. He is currently a professor in the Department of Computer Science, Nanjing University. He is also the director in the State Key Laboratory for Novel Software Technology. His research interests include programming methodology, pervasive computing, software agent, and middleware. He is a member of the ACM and IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.