# Multimodal Algorithm Based on Particle Filter for Indoor Localization with Smartphones

**Rui Tao[1], Haiyong Luo[2], Fang Zhao[1], Yongzhong Li[2]**

[1]Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2]Institute of Computing Technology Chinese Academy of Sciences, Beijing, 100190, China

**Abstract -** Location-based service is one of the most important demand in ubiquitous computing scenarios. While build-in sensors on smartphones can be used to provide location information for terminals. This paper presents a particle filter, dead reckoning, WIFI based multimodal algorithm for indoor localization with smartphones, and propose a system so called *ICTIPS*. *ICTIPS* make use of WIFI signal strength measurements, accelerometer, magnetometer and gyroscope on smartphones, as well as map information, offering high tracking accuracy and efficiency. In particular, we compare different weight update strategies for particle filter framework and similarity functions for magnetic matching progress to understand the impact on accuracy and efficiency of the positioning system. In our real office environment, *ICTIPS* is evaluated with high quality ground truth, which provide service with an average tracking error around 2 meters.

**Index Terms** – Smartphone, Dead reckoning, Particle filter, DTW, Magnetic

## 1. Introduction

Location-based service (LBS) has been one of the most popular features on mobile phones. As the basic component of LBS, localization and navigation largely determine the quality of service. In the outdoor environment, GPS or cell signals are commonly used, when it comes to the accurate and robust indoor localization case, the problem is still unsolved. Researchers saw the problem, and have put great effort to explore any available signals around or inside devices, but by now on, few of them have achieved the expected meter-level accuracy.

In this paper, we developed a practical indoor localization system which relies on smartphone sensors. This system is able to reliably provide meter-level positioning accuracy for common smartphone users, thanks to its multimodal algorithm. Specifically, the contributions of our work are threefold:

*1)* We designed a reliable multimodal algorithm, which is driven by the well-known particle filter framework [3], and modules like WIFI positioning, pedometer, dead reckoning, turning detection all together contribute to the process.

*2)* We developed a fingerprinting method for magnetic signals [5][6][7], which is used in the weight update procedure of the particle filter process.

*3)* We built a system for indoor positioning using smartphones, and demonstrated that the system can achieve reliable meter-level accuracy.

In the rest of this paper, first we introduce the overall architecture of our indoor positioning system, and then present multimodal algorithms, turning detection method and fingerprinting method. We evaluate the performance of each algorithmic module and the final system, compared results of using different arguments in the magnetic matching step. The paper ends with discussions on the system and future work.

## 2. The ICTIPS Positioning System

### A. Overview

The *ICTIPS* indoor localization system consists of a great amount of modules, the pedometer, dead reckoning module, turning detection module, step length adapter, magnetic module for updating particle weights, and the most important particle filter as in *figure 1*. The system initialize location estimation by our old WiMap[1][2] system, which is a WIFI fingerprinting methods based positioning solution. And the positioning result is shown on a SVG-based map module.

*1) Pedometer*: The responsibility of pedometer is to detect user step, and report to the system main logic. By each step, the system will performs a particle filter process, and then the updated position is given. By default, the step model uses a generic step length.

*2) Dead Reckoning and Turing Detection*: The DR-module uses data from compass, accelerator and gyroscope, put forward an estimated orientation of user movement, which is then being used to select a reliable direction depending on the map information. Similarly, the Turing Detection module also take full advantage of the listed sensors, but to notify sharp rotate actions.
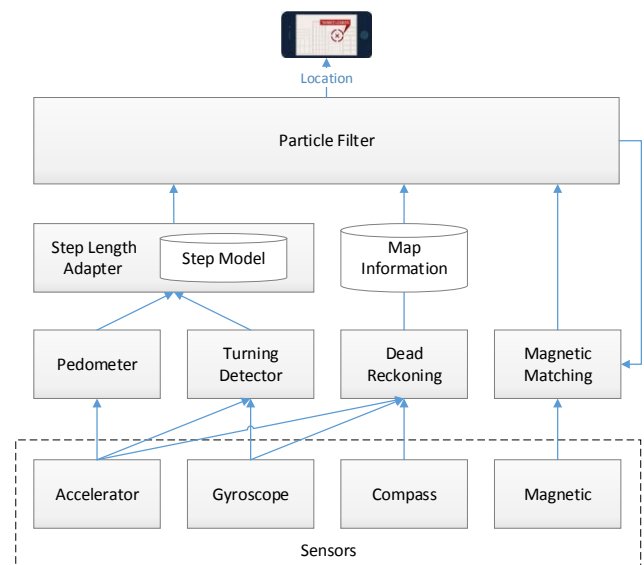


Fig. 1: The overall system architecture

*3) Step Length Adapter*: To fit different user personalities, the system make use of both pedometer and the turning logic to acquire steps and distances as accurate as possible, provide ability of adapting step length.

*4) Magnetic Matching*: Collected magnetic sequence data in the free space of indoor environment is put to use as fingerprint in the offline training phase and as measurements in the online tracing phase. Comparison tasks is driven by the *DTW* or extended *DTW* series algorithm.

*5) Particle Filter*: Particle filter algorithm is selected as the main logic of our system, which as we all known, is a nonparametric form of Bayesian estimation, commonly used in computer vision and tracking. The algorithm integrates information from the dead reckoning module, pedometer, and magnetic matcher, to arrive at the posterior distribution of the position. It is modularized so that different kind of movement methods, constraints, and weight update strategies can be united or replaced as wish.

### B. Particle Filter Framework based Multimodal Algorithm

In the following paragraphs, we describe a particle filter based multimodal method for position tracing. Six different kinds of information are used for our filter: Compass data, accelerometer data, gyroscope data, magnetic data, WIFI signals, and a map of building. The state of each particle consists of two pixel coordinates x and y on the map, accuracy estimated, a direction d, and a step length l, history state of each step, and history of magnetic data, all together contribute to the final result.

The algorithm consists of several phases. Starting with the initialization, where a first set of particles is created, the following phases depend on the incoming data. When a step is detected, the move phase is carried out according to the move model. And weights update relies on the observation model, which will make full use of signals data collected. Then follows a resampling step, where particles with higher weight are split and those who bump the wall are killed. Finally the update phase is proceeded normalizing the weight of all particles according to the measurement model, and the final position result is packed and delivered.

For initialization of particle filter, a fixed number $N$ of particles are generated according to the following initialization scheme:

*1)* Particles located in the certain circle area *(x, y, radius)*, which is defined by the incoming WiMap positioning result. The *x, y* are the coordinates as the center point, and the radius is the accuracy estimated. All particles are randomly generated with different coordinates.

*2)* Particles not on the free space are abandoned and new ones are created instead to match the indicated *N* in demand.

*3)* Each particle holds a different base step length, which fits the following formula: $L_i = L_0 + \delta_i$ , while $\delta_i$ is a random generated increment. This also guarantees diversity of particles.

*4)* Note that all weights of particles are initialized to be *1/N*.

The move phase is triggered when a step is detected by the pedometer module. The compass digital direction is passed in as parameter, and the move direction base is picked from the map information, which matches the given parameter most, and to maintain particle variety, a Gaussian random value is appended. In general, particles move in the free space. The distance that particle shifts forward is assigned by the step length. Note that when a particle is detected to bump the wall, that means it goes out of the free space, it is marked, and its weight will fall to zero in the next phase.

In the reweight phase, we adjust the weight of particles by applying magnetic matching on them. As each particle stores a list of history locations, we can easily retrieve magnetic samples from the database according to the given coordinates. Thus we use *DTW* algorithm to calculate the distance between samples and on-the-fly measurements, convert it into similarity, a value between 0 and 1, and multiply that with the current weight. Note that particles updated by smaller similarity can less influence the result.

After the reweight phase, a resampling algorithm is carried out, including the accumulated weight matrix calculation and the regeneration of particles randomly. The accumulated matrix hashes each interval into a particular particle, so that particles with greater weight, which occupy larger interval, will have more chances to be regenerated, or in other words, split.

Finally, we normalize the weights of particles according to the basic rules:

$$\omega_i = \omega_i / \sum_{i=0}^{N} \omega_i. \tag{1}$$

And the weighted location is updated too.
*Location(x, y, accuracy)*:

$$x = \sum_{i=0}^{N} \omega_i * x_i, \tag{2}$$

$$y = \sum_{i=0}^{N} \omega_i * y_i, \tag{3}$$

$$accuracy = max\big(Distance(x, x_i)\big) \tag{4}$$

### C. Turning detection and Dead Reckoning.

*1)* The Cartesian frame of reference of the phone is represented by the orthogonal *xyz* axes with the *x*-axis pointing to the right side of the phone, the *y*-axis pointing to the top and the *z*-axis leaving the screen, and we intentionally use capitalized *XYZ* to distinguish from the Cartesian frame. We proposed the turning angular calculating formula around *Z*-axis as follows: $\left(\alpha\mu_x + \beta\mu_y + \gamma\mu_z\right)/\sqrt{\mu_x^2 + \mu_y^2 + \mu_z^2}$ [4], in which $\mu_x$, $\mu_y$, $\mu_z$ are the average acceleration readings of *xyz* axes, and $\alpha$, $\beta$, $\gamma$ are the angular displacements around *xyz* axes by integrate gyroscope readings.

*2)* Things may change when the user hold the device in different ways. One worse situation is that the user hold the phone and shake while walking (named SHAKE mode). *Figure 2* shows the movement pattern of the phone in this case.

In this situation, the motor direction of device goes perpendicular with component of gravity in *x*-axis. And device will always reach the highest speed at the bottom.

The *figure 3* shows the differences on speed for different holding styles of device. The green line demonstrate the speed when the device is hold horizontally (the HORIZON mode). The red and blue line shows the speed when swing the arms naturally (the SHAKE mode), but using different hands.

The case of putting device in the pocket is similar to the case of holding device without shaking while keeping the arm drop naturally (named POCKET mode). *Figure 4* and *5* respectively shows the gyroscope readings and accelerator readings of POCKET mode and SHAKE mode. From the first graph we can see only readings of *z*-axis tells obvious distinctions. From the second graph we can reach the conclusions:

*i.* *x*-axis readings of both modes result in close average, but the variance in SHAKE mode is larger.

*ii.* The absolute average of *y*-axis readings of SHAKE mode is bigger, as well as the peek.

*iii.* Readings of *z*-axis in SHAKE mode, that of *yz*-axes in POCKET mode are similar.

Based on the above data, we are able to model each modes, and so it is possible to recognize device state (holding style) in the online phase.

*D. Magnetic Matching*

*1)* In the indoor environment, magnetic signal is various because of the specific metal distribution. Although 3 axis changes when the smartphone is on different posture and placement, the combination of them remains stable. So we choose $d = \sqrt{x^2 + y^2 + z^2}$ as the formula of fingerprinting from raw data.

*2)* The magnetic fingerprints matching is a time series matching problem. As the sampling rate might be different and the speed of device movement might be different as well, time series may be "wraped" non-linearly by stretching or shrinking it along its time axis. *Dynamic Time Warping (DTW)* algorithm can do the favor to find the optimal alignment between two time series. It is often used to determine time series similarity, classification, and to find corresponding regions between two time series.

*3) Distance to Similarity:* What we need to do next to convert distance to similarity, which should between 0 and 1.

When the distance is zero, similarity should be 1, and as distance grows, similarity decreases. The similarity formula is shown as follows: $= e^{-d^2/k}$ , where $k$ is the length of path defined in *DTW* algorithm.

*4) Low-Cost Fast Map Model:* One of the strict problem is how to normalize and store the magnetic data. As magnetic data are collected in several times, even by multiple person, it is important to fuse all these segments a unified whole. One principle is to align segment end point coordinates, and besides, waveform matching method is proposed to calibrate adjacent segments, and finally, multiple samples of same path should be united as one using both calibration algorithm and averaging.

We designed a Matrix-based structure to organize the preprocessed data sets.

*i. Organization of raw fingerprints.* Fingerprints are recorded path by path. Each of the records contains information of coordinates of two end points, and the magnetic on path.

*ii. Matrix and Coordinates Hash Map.* The space on building map can always be covered with grids, what really matters is the total number, which is determined by the size
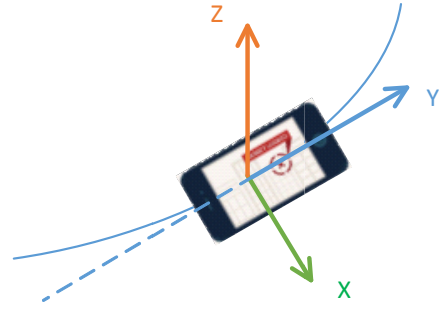


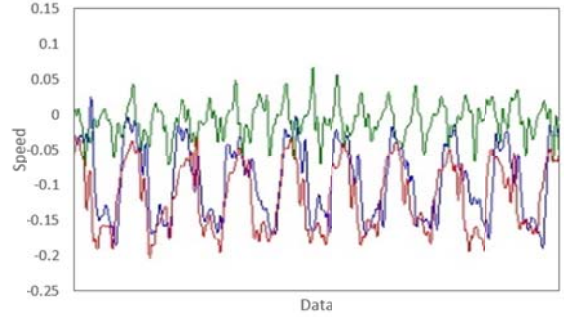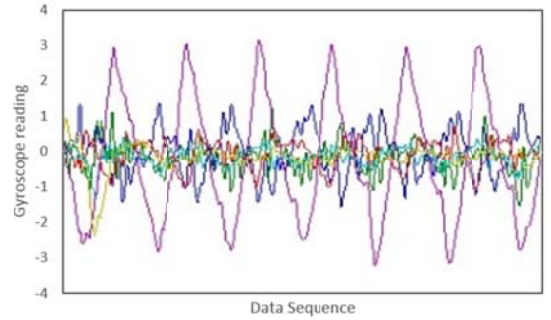Fig. 2: Movement pattern



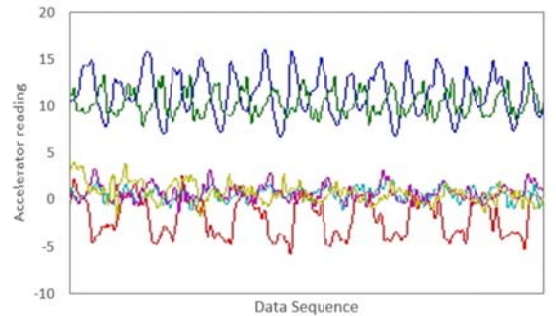Fig. 3: Speed of device



Fig. 4: Gyroscope readings



Fig. 5: Accelerator readings

of each grid. *Linked-List* structure and *Coordinates Hash Map* are two ways to represent the sparse matrix. The two basic functions of the Matrix are designed as follows:

*Load***:** The module reads records in a loop, and for each record, splits path by grid size, and merge fingerprints grid by grid. Add all grid to Matrix (Linked-List or Hash Map).

*Retrieve***:** The track of particle is also split by grid size in the same way, and then search for fingerprints in sequence in the Matrix. Note that in some cases, system may not hit the target, so we need enlarge the search area to find a nearest grid for replace.

## 3. Evaluation

We evaluated the algorithms in our office environment. Information was gathered with different smartphones, which covers HTC, sumsang and xiaomi. In the workspace, 35 grids for the WLAN fingerprint database were selected. 30 samples were collected by scanning the received signal strength (RSS) of each AP for each of four orientation of each grid. Moreover, eight tracks were recorded (see *Figure 6*): While walking along a certain path through the test environment, the magnetic readings were stored in file. For experiment, sensor generated readings at a series of rate from approximately 20Hz to 50Hz.

To evaluate the accuracy of system, we divided each path with specified steps, so that the coordinates of each step is computable. Although the measurement error is inevitable, we tried our best to reduce the influence. The positioning error is calculated as the projected distance between the marked ground truth and the estimated location.

*Figure 7* plots the statistical results of accuracy at different error distance with different magnetic matching length, where number of particles is 100, and sampling rate is 20.

The following table shows the hint rates of Turning Detection module with different mode and turning radius, as a result of 200 tests. Turning radius increases, hint rate falls.

TABLE I   Turning Detection Hint Rate

| MODE  RADIUS(m) | SHAKE | POCKET | HORIZON |
|---|---|---|---|
| 1 | 0.97 | 0.98 | 0.98 |
| 5 | 0.90 | 0.91 | 0.90 |
| 10 | 0.73 | 0.78 | 0.76 |

## 4. Conclusions and Future Work

In this paper we presented a multimodal algorithm based on particle filter and the system built on it. The algorithm take almost all sensor signals into account and provide precise positioning and tracking services. What we do is to focus more on the weights update process in the particle filter framework, core of which is magnetic matching problem and user turning detection.

In the future, we plan to improve the system by adding a confidence evaluation module, user activity recognition module and mark-points module, which can provide additional correction possibilities on characteristic positions. A better magnetic data fingerprinting technique, a more efficient magnetic matching algorithm, a more convenient magnetic data structuring method is needed to achieve better performance. Long-term magnetic matching techniques is also

a meaningful research point. And research on device compatibility is also on our TODO list.
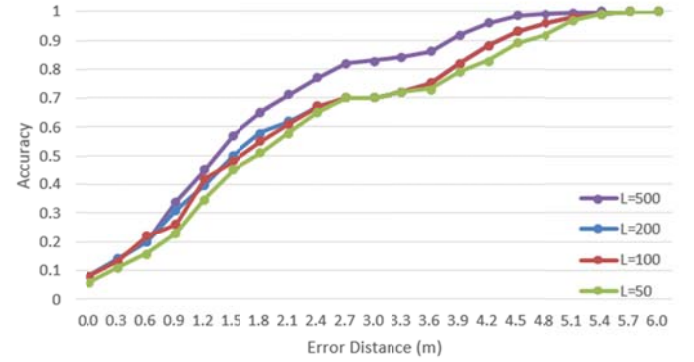


Fig. 6: Evaluation Environment



Fig. 7: CDF of Different Magnetic Matching Length

## References

[1] Junjun Xu, Haiyong Luo, Fang Zhao, Rui Tao, Yiming Lin. Dynamic indoor localization techniques based on Rssi in WLAN environment. Pervasive Computing and Applications (ICPCA), 2011 6th International Conference.

[2] Junjun Xu, Haiyong Luo, Fang Zhao, Rui Tao, Yiming Lin, Hui Li. The WiMap: A Dynamic Indoor WLAN Localization System. IJAPUC (2011)29-38.

[3] Moritz Kessel, Martin Werner. Automated WLAN Calibration with a Backtracking Particle Filter. International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012.

[4] Xiaojun Zhu, Qun Li, Guihai Chen. APT: Accurate Outdoor Pedestrian Tracking with Smartphones. The 32th IEEE International Conference on Computer Communications (IEEE INFOCOM 2013), Turin, Italy, April 14-19, 2013.

[5] W. F. Storms and J. F. Raquet, "Magnetic field aided indoor navigation," in Proc. 13th Eur. Navig. Conf. GNSS, 2009, pp. 1–9.

[6] Brandon Gozick, Kalyan Pathapati Subbu, Ram Dantu, and Tomyo Maeshiro. Magnetic Maps for Indoor Navigation. IEEE

[7] William Storms, Jeremiah Shockley, John Raquet. Magnetic Field Navigation in an Indoor Environment. Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010.