

WaP: Indoor Localization and Tracking Using WiFi-Assisted Particle Filter

Feng Hong, Yongtuo Zhang, Zhao Zhang, Meiyu Wei, Yuan Feng and Zhongwen Guo

Department of Computer Science and Technology

Ocean University of China, Qingdao, China

Email: hongfeng@ouc.edu.cn

Abstract—High accurate indoor localization and tracking of smart phones is critical to pervasive applications. Most radio-based solutions either exploit some error prone power-distance models or require some labor-intensive process of site survey to construct RSS fingerprint database. This study offers a new perspective to exploit RSS readings by their contrast relationship rather than absolute values, leading to three observations and functions called turn verifying, room distinguishing and entrance discovering. On this basis, we design WaP (WiFi-Assisted Particle filter), an indoor localization and tracking system exploiting particle filters to combine dead reckoning, RSS-based analyzing and knowledge of floor plan together. All the prerequisites of WaP are the floor plan and the coarse locations on which room the APs reside. WaP prototype is realized on off-the-shelf smartphones with limited particle number typically 400, and validated in a college building covering $1362m^2$. Experiment results show that WaP can achieve average localization error of 0.71m for 100 trajectories by 8 pedestrians.

I. INTRODUCTION

Indoor localization provides crucial services for mobile and pervasive applications from advertisement promotion in shopping mall to navigation during emergency rescue. Taken along by users round-the-clock, mobile phones nowadays have become the most important information interface between users and environments, stimulating extensive research on localization based on smartphones.

Besides exploiting properties observed from real traces [1] [2], previous research on indoor localization systems can generally be divided into two categories based on the nature of measurements: (1) Pedestrian Dead Reckoning (PDR) based on Inertial Measurement Units (IMU) and (2) Received Signal Strength (RSS) as a metric for location determination.

PDR systems exploit the readings of off-the-shelf IMUs embedded on smartphones, such as accelerometers, gyroscopes and magnetometers to detect steps, calculate stride length and determine heading direction. PDR localization precision suffers from that small errors could be magnified by integration [3]. The noise in low cost IMUs in commodity smartphones will cause error in the process of numeric integration during walking [4] [5]. And the unconscious human behavior, such as hand trembling [6], will also create great change on the readings of gyroscope, while the phone is held in hand. This kind of readings introduces errors into indoor localization and tracking, for it has nothing to do with human motion, so some calibrations are needed, e.g, in turning phase [7].

RSS-based approaches can be further classified into two categories: model-based and fingerprint-based. Model-based

techniques are error-prone mainly because the uncertain influences cannot be reflected in power-distance model, like background interference, non-uniform spreading, signal fading and reflections [8]. D. Turner et al. [9] investigated several model-based techniques, reporting average error greater than 5 meters. Meanwhile, the accurate locations of APs are often unable to get for privacy reasons and redeployment of APs.

Fingerprint-based systems [10] [11] include an off-line training phase and an operating phase. Training phase aims to build the RSS fingerprint database through site survey by engineers. In operating phase, users interact with the fingerprint database to get their locations by sending his current RSS fingerprint. To some extent, the accuracy of localization depends on the density of site survey [12] [13]. And it exposes the user's location and trajectory to the database server, causing some privacy issues. Site survey is not only labor intensive but also vulnerable to environmental changes. Several researches have been proposed to replace site survey by crowdsourcing, exploiting PDR to get the location and RSS fingerprint [14] [15].

To address above issues, we bring up this question: *How to combine these two approaches to realize localization and tracking with high accuracy without exposing the trajectory?* Several new challenges are confronted under this question. First, we need a new perspective to interpret RSS readings, for model-based technologies are error-prone and the fingerprint-based approaches suffer from privacy issues. Second, we need new methods on how to apply RSS readings to address the error accumulation problem of PDR. Last but not least, the computation resources of smartphone are still limited, so the components implemented in smartphones should be light-weighted.

In this study, we propose WaP (WiFi-Assisted Particle filter), an indoor localization and tracking approach exploiting dead reckoning with the help of RSS readings. Inspired by our previous study on RSS [16], we find that notable variety will appear in RSS values during the motion of a pedestrian. This offers us a new perspective to exploit RSS readings by their contrast relationships rather than absolute values, concluding in three observations of Turn Verifying, Room Distinguishing and Entrance Discovering (details in Section III). WaP adopts particle filter as its core to represent and control uncertainty in dead reckoning. When applying these observations on RSS, WaP corrects the accumulated error of dead reckoning to achieve high accuracy. Meanwhile, WaP only needs several hundred of particles typically 400 distributed in 1.2m radii around the center location, which can be dealt efficiently with

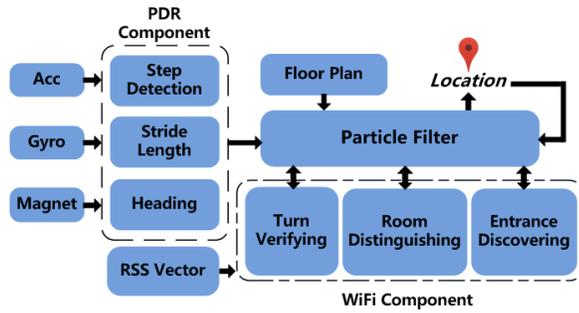


Fig. 1: WaP Architecture

commodity smartphones' restricted resources of computation and storage.

Without arduous site survey and deploying an extra server, WaP works only on the smartphone itself under the knowledge of the floor plan and the coarse knowledge in which room the APs reside. The floor plan can be downloaded while entering the target building. The coarse distributions of APs can be supplied within the floor plan, or can be provided through naming the APs according to their room numbers by users, which is common in practice.

To validate this design, we implement a prototype system on Google Android G3 phones. Extensive experiments are conducted in a middle-sized college building covering $1362m^2$. The average localization error of 100 trajectories by 8 pedestrians is 0.71m.

The major contributions are as follows:

- We identify a new perspective of applying RSS readings by their contrast relationships rather than absolute values, and exploit such observations into three functions, i.e. Turn Verifying, Room Distinguishing and Entrance Discovering.
- We design a practical scheme that requires limited computation and storage resources and no interaction with center servers. The preconditions of WaP are only the floor plan and which room APs reside rather than a RSS fingerprint database or exact locations of APs.
- The prototype of WaP is realized on Android smartphones, and is evaluated upon 100 trajectories by 8 pedestrians. WaP achieves average localization accuracy of 0.71m.

The rest of the paper is organized as follows. We present the system overview of WaP in Section II. Section III illustrates our new perspective on exploiting RSS readings and how these observations can be utilized in WaP. The complete design of WaP is showed in Section IV. Section V exhibits the experiment result of WaP prototype. We conclude the work in Section VI.

II. WAP OVERVIEW

This section presents the system architecture of WaP, shown in Fig. 1. WaP takes the readings of WiFi sensor, inertial sensors and magnetometer embedded on smartphone as input, outputting the user's location upon each step.

WaP contains three major components: PDR, WiFi and Particle Filter (PF). Based on the accelerometer readings, PDR component counts the steps of the user, and calculates the stride length of each step. It also computes the heading direction of each step based on gyroscope and magnetometer readings. Merging these, a $\langle displacement, direction, time \rangle$ tuple forms the human motion vector, which will pass to PF component. The realization details of PDR component are based on our previous work [7], so we eliminate the details of PDR due to page limit though there are slight improvements in step counting and heading estimation.

WiFi component records RSS values periodically from all available APs in the floor as a RSS vector of $\langle rss_1, rss_2, \dots, rss_n, time \rangle$ if n APs can be heard. WiFi component has three functions, all of which are deduced from that user motion will cause notable variety in RSS vectors. Turn Verifying will judge whether the heading direction change comes from user's turning or other transient unconscious behavior like hand trembling, which may mislead PDR due to insufficient scanning of low cost inertial sensors. If the latter happens, it will correct the heading direction. Room Distinguishing focuses on the change of RSS vector. It informs the PF component which room the user is entering by detecting specific variety of elements in RSS vectors. Entrance Discovering tries to discover a possible path when the estimated position remains almost the same while particles keep dying for a number of steps. Such three observations and functions are described in Section III.

Particle filter represents and handles the uncertainty of PDR, leveraging the constraints imposed by the floor plan and the indication of the WiFi component. PF component redistributes every particle according to the motion vector from PF component in propagation phase. Then the correcting phase first correct the weight of each particle according to the floor plan and calculate the weighted center of the particles. Upon the geometric relationship between the new center and last tracking position, PF component invoke Turn Verifying, Room Distinguishing and Entrance Discovering of WiFi component to farther correct the particle weights. The resampling phase follows and outputs the center of weighted particles as the current estimated location of the pedestrian. The details of PF component will be described in Section IV. The sampling rate of WiFi sensors is much lower than that of inertial sensors, so WiFi component employs the RSS vector which is the nearest to a certain motion vector in time series.

III. OBSERVATIONS

Model-based RSS localization techniques are error-prone and require the exact locations of APs. Besides, fingerprint-based RSS localization needs site survey or crowd sourcing to construct fingerprint database, which is not only labor-intensive but time-consuming. Even the AP redeployment in the same room would make the database lose efficacy. So a new perspective to apply RSS readings is nontrivial. We describe our observations and corresponding methods to exploit RSS readings in this section, only needing to know which room the APs reside in. This kind of coarse locations can be directly provided by naming the APs after the room number.

Our perspective is to exploit RSS readings by their contrast relationship rather than absolute values, coming from our

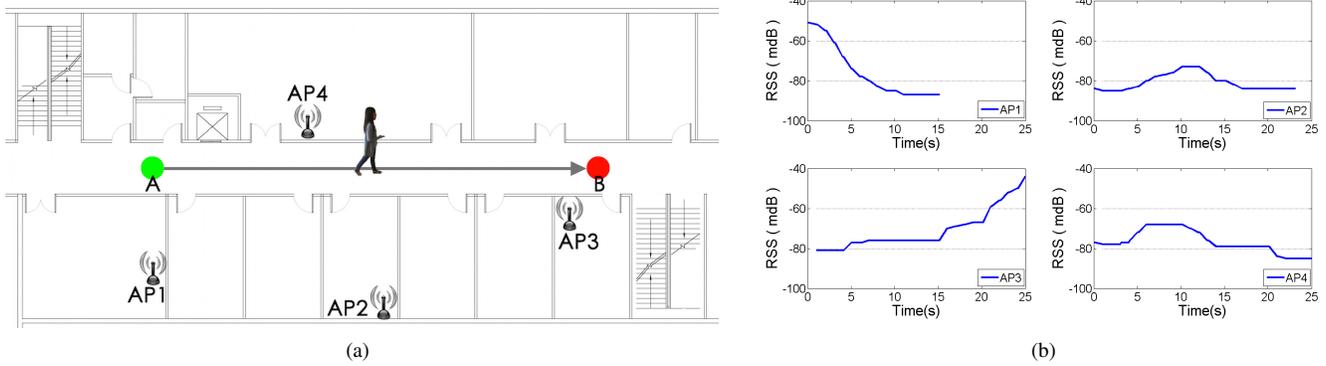


Fig. 2: Observation of RSS variety while motion (a) trajectory (b) RSS values

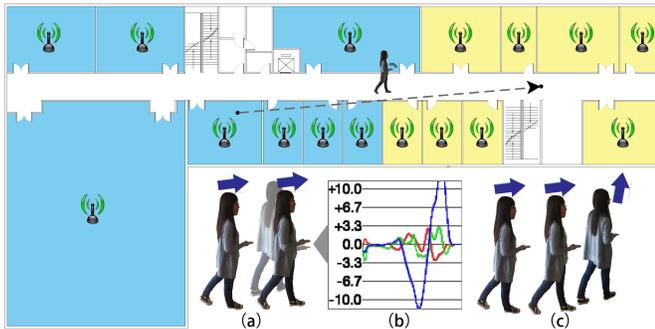


Fig. 3: Trace on Turn Verifying with *RSS Direction*: (a) hand trembling (b) readings of gyroscope (c) intentional turn

previous RSS study [16]. We observe notable varieties in RSS vectors during human motion process, shown in Fig. 2. When walking from point A to B, Fig. 2(b) shows the pattern that the closer a pedestrian walks to an AP, the higher RSS values the mobile phone reads.

This observation is leveraged into three detailed circumstances, to judge whether an intentional turn appears or just some unconscious trembling behavior happens in motion, to distinguish which room the pedestrian enters, and to amend from the dead end to nearby entrance of a certain room. We call them Turn Verifying, Room Distinguishing and Entrance Discovering respectively, elaborating them in the following.

A. Turn Verifying

When a pedestrian walks, unconscious human behaviors, such as hand trembling, will cause great change on the readings of direction sensors. Figure 3 shows a walking scenario that the pedestrian walks along the corridor from left to right. There is an unconscious trembling at the current step. The readings of gyroscope are shown in Fig. 3(b), including a considerable fluctuation. Counting such fluctuation into direction integration, an offset of heading occurs, and PDR component creates the motion vector indicating the pedestrian arrives at the shadow location in Fig. 3(a). If such wrong estimation is not corrected, huge localization errors will appear through accumulation, which will be demonstrated in one real trajectory in evaluation section.

When examining RSS vectors between continuous steps,

Algorithm 1: *verifyTurn*(t_c, t_{p1}, t_{p2})

Input: t_c, t_{p1}, t_{p2}

// t_c, t_{p1}, t_{p2} : current and last two steps' times

Output: *HeadingDirection*

```

1  $v_1 = CalculateRSSDirection(t_c, t_{p1});$ 
2  $v_2 = CalculateRSSDirection(t_c, t_{p2});$ 
3  $\omega = arccos(v_1 \cdot v_2) / (|v_1| \cdot |v_2|);$ 
4 if  $\omega \in (\pi/4, 3\pi/4)$  then
5   //trembling threshold=  $\pi/4$ ;
6   return HeadingDirection on  $t_{p1}$ ;
7 end
8 else
9   return HeadingDirection on  $t_c$ ;
10 end

```

Algorithm 2: *CalculateRSSDirection*(t_c, t_p)

Input: t_c, t_p

// t_p : current and previous steps' times

Output: *RSS Direction*

```

1 for each  $rss$  in RSS vector do
2   if  $rss_{t_c(i)} - rss_{t_p(i)} > 0$  then
3     | add  $AP_i$  to  $Set_I$ ;
4   end
5   else
6     | add  $AP_i$  to  $Set_D$ ;
7   end
8 end
9 if  $Set_I \neq \phi$  &&  $Set_D \neq \phi$  then
10  |  $p_I = CalculateGeometricCenter(Set_I);$ 
11  |  $p_D = CalculateGeometricCenter(Set_D);$ 
12  | return  $p_I - p_D$ ;
13 end
14 else
15  | return null;
16 end

```

we find the key to distinguish pedestrian turning from hand trembling. We classify the APs into two sets according to whether the RSS value increases or decreases between consecutive RSS vectors. So APs are differentiated into the increasing set (Set_I) and the decreasing set (Set_D) shown in yellow (right) and blue (left) separately in Fig. 3. Because WaP does not have exact locations of APs, we paint the rooms they

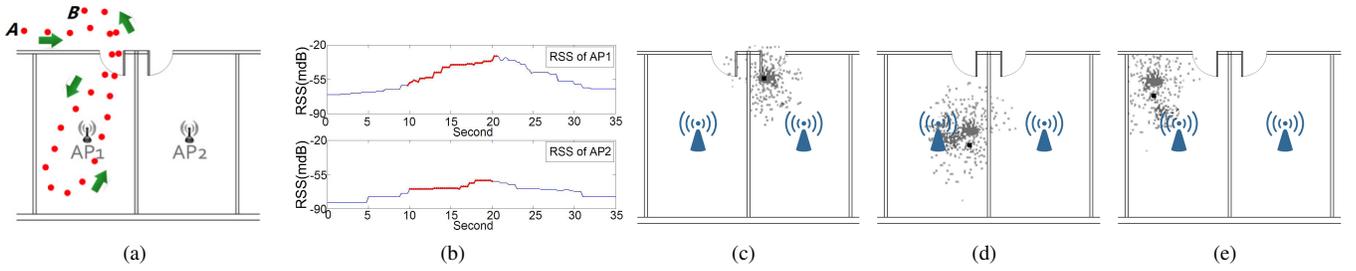


Fig. 4: Trace and process on Room Distinguishing (a) trajectory (b) RSS readings (c) before handling (d) after handling (e) after a few more steps

residing. Assuming that each AP resides in the center of their rooms, the geometric centers of Set_I and Set_D can be deduced as two points in Fig. 3 with an arrow linked each other. We call the direction of this segment as $RSS\ Direction$ of the current step.

$RSS\ Direction$ is used as an index to confirm whether a turn happens. If the angle of two consecutive $RSS\ Directions$ is higher than a threshold defined beforehand, WaP affirms the turn behavior; otherwise, WaP replaces the heading direction with the one from previous step. Because if no turn exists in motion, there will be no obvious change on $RSS\ Direction$, shown as the arrow above the pedestrian's head in Fig. 3(a). Algorithm `verifyTurn` is illustrated as Algorithm 1 and 2, which is invoked by PF component when a turn is found according to inertial sensors' readings. When an intentional turn is conducted, a remarkable change on $RSS\ Direction$ will appear, because elements in both Set_I and Set_D will change due to the change of receiving position and the block of human body. Hence, the new $RSS\ Direction$ from the current geometric center of Set_D to that of Set_I changes considerably, as the arrows shown in Fig. 3(c), calculated from the trace data containing a real turn.

B. Room Distinguishing

When a user is entering a room, we find a clear tendency in the change of RSS vectors. Figure 4(a) demonstrates a walk trace, in which a pedestrian starts from point A after standing still for 5 seconds, enters the room R_1 where AP_1 resides and goes to point B. Figure 4(b) reveals the RSS readings from AP_1 and AP_2 during the walking process with sampling frequency of 4 Hz. The increase of RSS from AP_1 is larger than that from AP_2 , especially after entering R_1 , shown in red color. In such scenario, the RSS readings of AP_1 will be the highest in the RSS vector. And the maximal change in RSS readings among successive steps will be from AP_1 as well. These two conditions give us the hint to distinguish which room is entered, agreed with all the traces. We quantify the conditions of room entering in Algorithm 3.

This observation helps distinguishing which room the pedestrian enters, when two doors are close to each other as the scenario shown in Fig. 4 (c)-(e). If the user is localized in the adjacent room, it will arise problems for context-aware applications and even fail the PDR systems in further steps. We discuss this example here and a typical trace containing such scenario in evaluation section.

As the distance between the doors of R_1 and R_2 is only 0.8m, we find in Fig. 4(c) that plenty of particles enter R_2

Algorithm 3: $distinguishRoom(t_c, t_p)$

Input: t_c, t_p
 // t_c, t_p : current and previous steps' times
Output: $RoomNumber$

- 1 $j_1 = \underset{i}{argmax}(RSS_{t_c}(i));$
- 2 $j_2 = \underset{i}{argmax}(RSS_{t_c(i)} - RSS_{t_p(i)});$
- 3 **if** $j_1 == j_2$ **then**
- 4 | **return** $RoomNumber$ where AP_{j_1} resides;
- 5 **end**
- 6 **else**
- 7 | **return** $null$;
- 8 **end**

Algorithm 4: $handleRoomDistinguishing()$

// recalculate weights

Input: $RSS, Floorplan, CurrentRoom\ r$
Output: $null$

- 1 find the wall of r , splitting P_i into Set_1 and Set_2 ;
- 2 $RSS_i(1) = r_{ss}$ measured from Set_1 's nearest AP;
- 3 $RSS_i(2) = r_{ss}$ measured from Set_2 's nearest AP;
- 4 **for** each particle in Set_1 **do**
- 5 | $w_i = \frac{(90+RSS_i(1))}{180+RSS_i(1)+RSS_i(2)} * w_i$;
- 6 **end**
- 7 **for** each particle in Set_2 **do**
- 8 | $w_i = \frac{(90+RSS_i(2))}{180+RSS_i(1)+RSS_i(2)} * w_i$;
- 9 **end**

because of the accumulated error of PDR. If the location is estimated based on this distribution of particles, the smartphone will believe the user enters R_2 , shown as the dark point. This is why WaP needs Room Distinguishing, which will correct the weights of the particles through Algorithm 4.

Algorithm 3 and 4 will be invoked by PF component in the correcting phase on finding a room-enter motion. After that, PF component will resample all the particles as distributed in Fig. 4(d). After further steps the particles will be distributed like Fig. 4(e), calibrating the weighted average location of particles to the reasonable position. Algorithm 4 uses 90 to compensate for the low bound of RSS measurements, which is $-90db$.

C. Entrance Discovering

In order to save the computation and storage resource on smartphone, the number of particles is limited to several hundred typically 400 in WaP, distributed under Gaussian

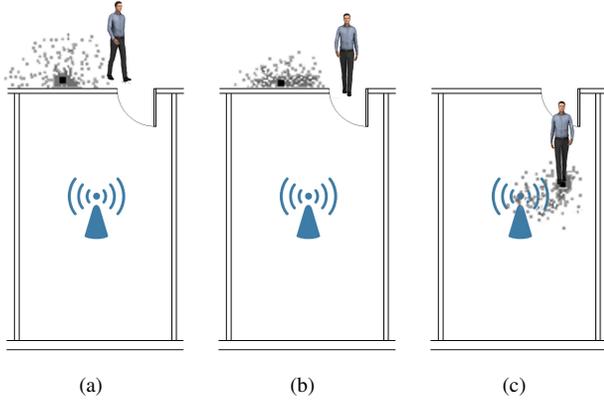


Fig. 5: Process on Entrance Discovering (a) before (b) dead end affirming (c) after

distribution within the radius of 1.2m which is half of a common corridor width. There may exist some odd situation that the steps of a pedestrian continue counting, but the estimated position keeps almost the same while particles dying. Figure 5(a) shows such scenario that a pedestrian enters a room while the particles keep dying with the estimated position against the wall and no particle is resampled in the room.

This kind of scenario may happen because of PDR errors and the limit particle distribution which cannot cover the right path. However, it is hard to tell by inertial sensors whether the pedestrian is colliding against the wall or PDR errors accumulate. Again the key to alleviate such dilemma lies in RSS vectors. If there are continuous changes on readings from some APs, WaP takes it for granted that the pedestrian is not running into a wall but is on a moving trajectory, as shown in Fig. 5(b).

WaP chooses three steps as a reasonable threshold. After finding that the estimated position keeps almost the same, PF component invokes `discoverEntrance` in Algorithm 5. It first find the AP of most RSS variety during these three steps. Then it queries the floor map to locate the nearest room entrance between the founded AP and current estimated position. A new particle will be created with weight 1, and positioned upon this entrance center plus three motion vectors of the last three steps. All other particles will be eliminated. It may somehow find the wrong room due to the uncertain influence on wireless signals. WaP can further correct such wrong choice through Room Distinguishing after further steps. The pedestrian will be localized to the position with higher accuracy after resampling phase of PF component, shown in Fig. 5(c).

IV. WiFi-ASSISTED PARTICLE FILTER

PF component receives the motion vector from PDR component. As the core of WaP, PF component drives particle filter and interacts with WiFi component, and outputs localization result. It starts with a set of weighted particles:

$$P_t = \{ \langle p_t^i, w_t^i \rangle, i = 1, \dots, N \}$$

$$p_t^i = (x_t^i, y_t^i, \theta_t)$$

Algorithm 5: `discoverEntrance()`

Input: $estPos_i, estPos_{i-1}, estPos_{i-2}, estPos_{i-3},$
 $thrd, RSS_t, \{l_i\}, \{\varphi_i\}$

// $\{l_i\}$: step lengths, $\{\varphi_i\}$: step directions, $thrd$: threshold

Output: P_t

```

1 if distance( $\|estPos_i - estPos_{i-1}\|$ ) <  $thrd$  then
2   if distance( $\|estPos_{i-1} - estPos_{i-2}\|$ ) <  $thrd$  &&
   distance( $\|estPos_{i-2} - estPos_{i-3}\|$ ) <  $thrd$  then
3      $j = \operatorname{argmax}(RSS_t(i) - RSS_{t-3}(i));$ 
4      $(x, y) =$  center of the entrance of the Room
      $(AP_j)$  resides;
5      $x_t^1 = x + \sum_{n=0}^2 l_{i-n}(\cos(\varphi_{i-n}));$ 
6      $y_t^1 = y + \sum_{n=0}^2 l_{i-n}(\sin(\varphi_{i-n}));$ 
7      $w_t^1 = 1;$ 
8      $p_t^1 = (x_t^1, y_t^1, \varphi_t);$ 
9      $P_t = \{ \langle p_t^1, w_t^1 \rangle \};$ 
10    return  $P_t$ 
11  end
12 return  $P_t$ 
13 end
```

where p_t^i is the estimated position with weight w_t^i of the i^{th} particle at step t . N is the number of particles. Iteratively, the new set P_t is generated from P_{t-1} through three phases as following:

Particle propagation: This phase updates positions for each particle. Both stride length l_t and heading orientation θ_t are passed by PF component and assumed to interfere by zero-mean Gaussian random noises. Hence the new location and heading orientation of the i^{th} particle at step t are

$$\theta_t = \varphi_t + \varepsilon$$

$$x_t^i = x_{t-1}^i + (l_t + \delta)\cos(\theta_t)$$

$$y_t^i = y_{t-1}^i + (l_t + \delta)\sin(\theta_t)$$

where δ and ε are zero-mean Gaussian noises for stride length and heading orientation respectively.

Particle correction: This phase is responsible to correct the weights of propagated particles. First, under the constraint of the floor plan, if the particle moves across a wall during the propagation, it will be given $w_t = 0$. On the contrary, longer surviving time (number of steps), higher probabilities are the particles besides the true location. So the weights are further updated as:

$$w_t^i = \frac{w_{t-1}^i}{\sum_{i \in P_t} w_{t-1}^i}$$

Then the wighted center of all particles are calculated and compared to the previous estimated positions. PF component will interact with WiFi component on particular conditions. If a turn is found, PF component will invoke Turn Verifying. If Turn Verifying denies the current turn and changes the heading direction back to the one from last step, PF component will cancel current position update and redo the propagation phase. Upon case of room entering, Room Distinguishing will be invoked. When the estimation position stays still for three steps

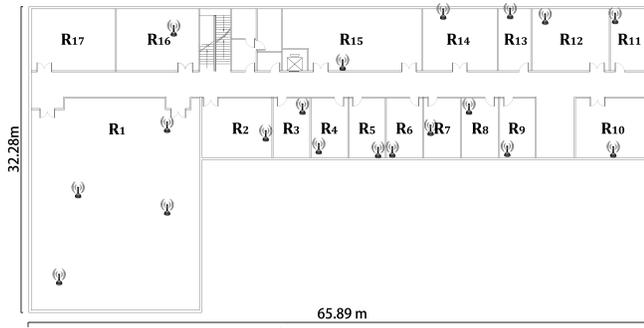


Fig. 6: Floor plan and AP distribution

by some wall, Entrance Discovery will be called. All these functions help in correcting the particle weights.

Resampling: This phase first delete the particles with weight 0. And all surviving particles will generate new ones distributed under Gaussian distributions according to their weights. The weighted center of all particles will be calculated again and outputted as the estimated position of current step.

V. EVALUATION

We develop the prototype on Google G3 phones, which support WiFi and inertial sensors and magnetometer. The sampling frequency of inertial sensors can achieve about 100Hz on this type of smartphone, and we set it at 40Hz in order to save resources. The sampling frequency of WiFi sensor is much lower, which is approximately 4Hz. Experiments are conducted on one floor of the campus building covering $1362m^2$, as shown in Fig. 6. 19 APs are deployed in this floor, one for each office room and 4 in the large laboratory. Room R_{17} is a storage room without any AP. As no site survey is organized to obtain the exact deployment of APs, WaP only has the knowledge that which room APs reside in and thereby assumes the room center as AP location.

Holding smartphones in hand during the experiments, 8 volunteers (4 females and 4 males) collect 100 trajectories. We first illustrate the performance of WaP on a long trajectory, which contains occurrences of Turn Verifying, Room Distinguishing and Entrance Discovering. The result is compared to the scheme without WiFi component i.e. PDR+PF. We then analyze the error distribution of all traces on the impact of AP number, comparing to Horus [11], a fingerprint-based system. Next, an offline analysis on particle number and distribution is conducted to affirm the computation-saving feature of WaP, compared to the system in [5], which exploits PDR and particle filter without WiFi signals.

A. Performance of WaP

To confirm the validity of WaP, a long complex trajectory, shown in Fig. 7, is designed, and the trajectory is labeled in alphabetical order according to the walking sequence, starting from point A and ending at point Q. Note that at point N, a hand trembling is arranged to demonstrate the function of Turn Verifying. Following the route sketched in Fig.7, localization and tracking results with PDR+PF and with WaP are presented in Fig. 8 (a) and Fig. 8 (b) respectively, in which the trajectory starts from the triangle icon and ends at the star icon.

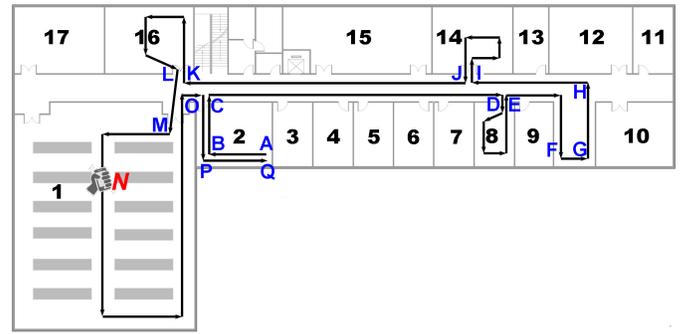


Fig. 7: Ground truth

Figure 8 (a) shows the localization and tracking result only with components of PDR and PF. There are two areas, pointed by red arrows, where critical errors for tracking happen, which makes the final estimated location unacceptably far from that in ground truth. In the area pointed by red arrow 1, the ground truth trajectory is that the pedestrian enters the room labeled R_8 , turns backwards and walks out. While most particles move past the door of R_8 and enter R_9 owing to limited number of particles, restricted distribution radii and some accumulated error on distance calculation in PDR. Hence the pedestrian is located in room R_9 , which needs to be corrected by Room Distinguishing of WiFi component. The area pointed by red arrow 2 is where the hand trembling is arranged. The system determines that a turn is made there due to unreliable readings of the gyroscope. As there is no Turn Verifying, a turn has been made in Fig. 8(a), eventually leading to great returning error.

Figure 8 (b) shows the localization and tracking result of WaP, which is better than that without WiFi component. Room Distinguishing is evoked and finds R_8 pointed by blue arrow 1, where biggest RSS reading changes happen for its corresponding AP. The weights of particles residing in R_8 will increase, then the pedestrian will be located to R_8 after particles resampling. The circle linked by the arrow represents the result of Room Distinguishing in Fig. 8(b). After that, further steps of the pedestrian is tracking correctly in room R_8 .

At the area pointed by blue arrow 2, when the results of PDR indicate that a turn happens, Turn Verify function of WiFi component begins and compares the *RSS Direction* of current step to previous step. As there is no obvious change on *RSS Directions* i.e. the angle between two *RSS Directions* is smaller than the threshold, Turn Verify function will change the heading direction to the one of previous step.

After finding that the estimated position keeps the same against the wall for three times, Entrance Discover function begins at the area pointed by blue arrow 3. It first checks whether the RSS vector changes continuously in these steps. For the scenario here, the RSS readings from the AP in R_2 see a continuously upward trend, and the AP that has the maximum reading variety is also in R_2 . Then the nearest entrance between the estimated location and R_2 , will be looked up in the floor plan. Entrance Discover function thereby set all particles with zero weight, create a new particle with weight 1 at the center of the door of R_2 , and plus previous motion vectors, shown as the three circles. Finally the estimated

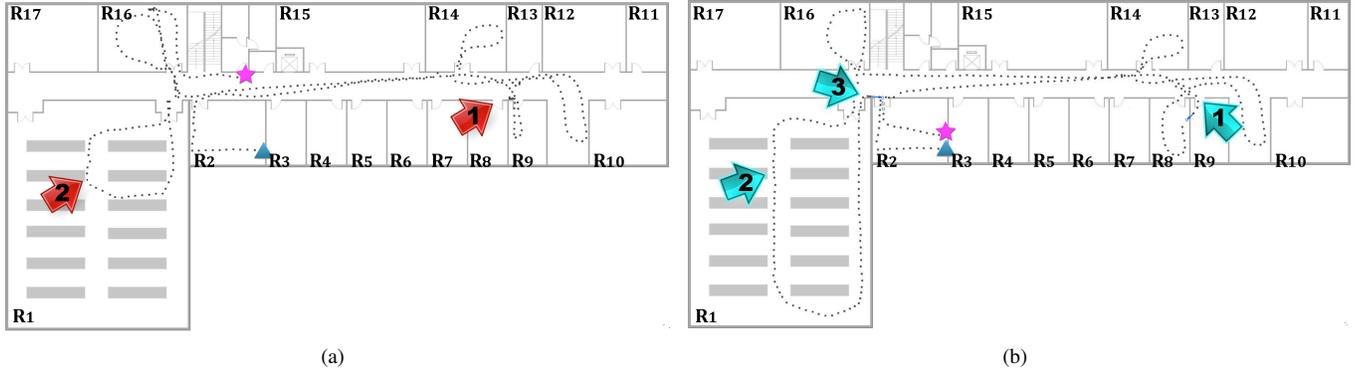


Fig. 8: Performance of WaP on a long trajectory (a) PDR+PF (b) WaP

position arrives at the end point only 0.9m away from the ground truth end.

B. Performance vs. AP Number

In order to further evaluate the performance of WaP, we cut down on the use of APs, setting the number 5, 10 and 15. We compare the localization results to Horus [11], a fingerprint-based approach without exploiting inertial sensor. We carry out site survey for the RSS fingerprint database of Horus with a sampling density of 2m beforehand.

Though we have 19 APs on the whole floor, there are only 15 AP-installed rooms in Fig. 6. WaP does not survey the exact position of each AP, so utilizing all four APs in room R_1 does not help much on performance of WaP. So we only choose one AP from room R_1 and every AP from all the other 14 rooms for the experiment with 15 APs. For 10 APs, we choose the APs residing in room $R_1, R_2, R_3, R_6, R_8, R_9, R_{12}, R_{14}, R_{15}$ and R_{16} . And the APs residing in room R_1, R_5, R_8, R_{12} and R_{16} are employed in the 5-AP experiment.

As it is too toilsome to survey the ground truth of each step, we examine the distance between the tracking position and the room entrance center on the time when pedestrians walking through the entrances. Every pedestrian is asked to walk through the entrances on the center beforehand. The tracking end position is also compared to the end of the ground truth. These differences are regarded as the localization error, used in this and next subsection.

100 trajectories by 8 pedestrians are used to construct localization error CDF in Fig. 9. It reveals that above 86% localization error distribution is below one meter of WaP for 15APs, however Horus needs over 2m to satisfy 86% of localization error distribution. As for 10-AP and 5-AP situations, performance of WaP to a large extent surpasses that of Horus. Besides, Horus is more vulnerable to the decrease of AP number: the errors upon 5 and 10 APs increase dramatically than 15 APs and even worse than WaP with 5 APs.

The average and standard deviation of localization error are described in Table 1. It shows the localization precision of WaP is about 100% better than Horus for all three scenarios. And the performance of WaP is also above 40% more stable than Horus with AP number of 5 and 10, and 120% with 15, respectively, confirming the localization stability. It should be

emphasized that WaP achieves that without any training effort, a.k.a. onerous site survey.

C. Performance vs. Resource Cost

Although location accuracy can be theoretically boosted by using more particles, the computation and storage cost of PF is proportion to the number of particles, leading to a trade-off between localization accuracy and utilization of smartphone resources.

In order to gain insight into such compromise, we conduct offline analysis on the trace data of 100 trajectories. The different choices of particle numbers are 100, 400, 1000, 4000 and 10000, respectively; and the choices of distribution radius are 1.2, 1.8, 2.4 and 3 meters, respectively. Therefore 20 combinations of particle number and distribution radius are adopted. WaP is compared with one adaptive approach proposed in [5], which exploits PDR and particle filter, without applying indoor WiFi signals.

Figure 10(a) shows localization error distribution of the approach in [5] under the above 20 combinations. The location error is among [104cm, 350cm] for all combinations shown in the color bar. The minimum occurs at 10000 particles with distribution radius 3m, while the maximum occurs at 400 with 3m. It also shows that for different particle numbers, the minimum location error exists with different distribution radius. For example, the minimum for 400 particles is 1.9m with radius 1.2m. This reflects that the density and coverage of particles affect localization precision, also existing in WaP.

For WaP, Fig. 10(b) shows the maximum location error is 1.18m under the composition of 100 articles with distribution radius 1.2, which is below the results for most combinations in Fig. 10(a). And the minimum happens at 10000 particles with distribution radius 3.0m, which is 0.31m. But for all the other combinations, the location error is above 0.6m. For 400 particles, the minimum average location error is of 0.7m under the sub-meter level. Actually it is challenging for a smartphone

TABLE I: Localization error comparison: WaP vs. Horus

#AP	WaP		Horus	
	Ave.(m)	S.D.	Ave.(m)	S.D.
15	0.71	0.57	1.56	0.81
10	1.92	1.04	3.94	1.49
5	3.42	1.32	6.63	2.80

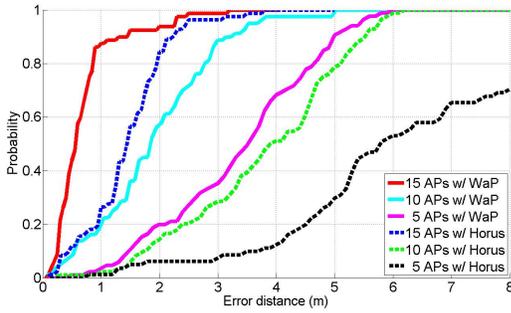


Fig. 9: Localization error CDF on different AP number, comparing WaP to Horus

itself to handle too many particles in real time without the aid of an extra server. Considering the computation and storage resources on mobile phone, and the radius of the human body, WaP chooses such combination for common use as in all above experiments. It indicates that WaP can perform satisfactorily even though restricted resources are used.

VI. CONCLUSION

This work proposes a novel approach, WaP, for indoor localization and tracking. WaP explores a new perspective on how to apply RSS readings by their contrast relationship rather than absolute values. WaP exploits particle filter as the bridge to integrate dead reckoning, RSS-based motion correcting and knowledge of floor map. WaP only needs the prerequisite of the floor plan with the coarse knowledge in which room APs reside rather than the exact locations. Performance of the prototype suggests WaP is a light-weighted and unconventional approach for indoor localization and tracking, motivating us to discover and utilize novel potential indicators under indoor environments.

ACKNOWLEDGMENT

This research is partially supported by National Natural Science Foundation of China under Grant # 61379128 and # 60933011.

REFERENCES

- [1] M. Azizyan, L. Constandache and R. R. Choudhury, *SurroundSense: mobile phone localization via ambience fingerprinting*, in Proceedings of 15th ACM MobiCom, 2009, pp. 261-272.
- [2] S. Rallapalli, L. Qiu, Y. Zhang and Y.-C. Chen, *Exploiting temporal stability and low-rank structure for localization in mobile networks*, in Proceedings of 16th ACM MOBICOM, 2010, pp. 161-172.
- [3] U. Steinhoff and B. Schiele, *Dead Reckoning from the Pocket*, in Proceedings of 8th IEEE PerCom, 2010, pp. 162-170.
- [4] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef and R.R. Choudhury, *No Need to War-Drive: Unsupervised Indoor Localization*, in Proceedings of the 10th MobiSys, 2012, pp. 197-210.
- [5] 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 Proceedings of the 14th ACM UbiComp, 2012, pp. 421-430.
- [6] Y. Jin, H.-S. Toh, W.-S. Soh and W.-C. Wong, *A Robust Dead-Reckoning Pedestrian Tracking System with Low Cost Sensors*, in Proceedings of 9th IEEE PerCom, 2011, pp. 222-230.
- [7] F. Hong, H. Chu, L. Wang, Y. Feng and Z. Guo, *Pocket Mattering: Indoor Pedestrian Tracking with Commercial Smartphone*, in 3rd International Conference on Indoor Positioning and Indoor Navigation, 2012.

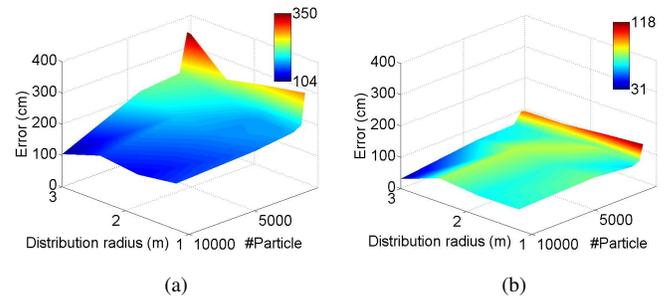


Fig. 10: Localization error vs. sampling radius and number of particles (a) approach in [5] (b) WaP

- [8] K. Chintalapudi, A. P. Iyer and V. N. Padmanabhan, *Indoor Localization without the Pain*, in Proceedings of 16th ACM MOBICOM, 2010, pp. 173-184.
- [9] D. Turner, S. Savage and A. C. Snoeren, *On the empirical performance of self-calibrating WiFi location systems*, in Proceedings of 36th IEEE LCN, 2011, pp. 76-84.
- [10] P. Bahl and V. N. Padmanabhan, *RADAR: An In-building RF-based User Location and Tracking System*, in Proceedings of 9th IEEE INFOCOM, 2000, pp. 775-784.
- [11] M. Youssef and A. Agrawala, *The horus WLAN location determination system*. In Proceedings of 3rd ACM MobiSys, pages 205-218, 2005.
- [12] H. Liu, Y. Gan, J. Yang, S. Sidhom and Y. Wang, *Push the Limit of WiFi based Localization for Smartphones*, in Proceedings of 18th ACM MOBICOM, pp. 305-316, 2012.
- [13] G. Chandrasekaran, M. A. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. P. Martin, *Empirical evaluation of the limits on localization using signal strength*, in Proceedings of the 6th IEEE SECON, 2009, pp. 1-9.
- [14] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan and R. Sen, *Zee: Zero-Effort Crowdsourcing for Indoor Localization*, in Proceedings of 18th ACM MOBICOM, 2012, pp. 293-304.
- [15] Z. Yang, C. Wu and Y. Liu, *Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention*, in Proceedings of 18th ACM MOBICOM, 2012, pp. 269-280.
- [16] Z. Guo, Y. Guo, F. Hong, et al. *Perpendicular Intersection: Locating Wireless Sensors with Mobile Beacon*, in Transactions on Vehicular Technology, IEEE, Vol. 59, no. 7, pp. 93-102, 2010.