

Article

Not peer-reviewed version

---

# SWiLoc: Fusing Smartphone Sensors and WiFi CSI for Accurate Indoor Localization

---

[Khairul Mottakin](#)\*, Kiran Davuluri, [Mark Allison](#), [Zheng Song](#)\*

Posted Date: 30 August 2024

doi: 10.20944/preprints202408.2254.v1

Keywords: indoor localization; walking direction estimation; dead reckoning; smartphone sensor fusion; channel state information (CSI);WiFi sensing




Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# SWiLoc: Fusing Smartphone Sensors and WiFi CSI for Accurate Indoor Localization

Khairul Mottakin <sup>1</sup>, Kiran Davuluri <sup>1</sup>, Mark Allison <sup>2</sup> and Zheng Song <sup>1,\*</sup>

<sup>1</sup> Department of Computer and Information Science, University of Michigan-Dearborn, MI, USA; khairulm, kirandav, zhesong@umich.edu

<sup>2</sup> College of Innovation and Technology, University of Michigan-Flint, MI, USA; markalli@umich.edu

\* Correspondence: zhesong@umich.edu

**Abstract:** Dead reckoning is a promising yet often overlooked smartphone-based indoor localization technology that relies on phone-mounted sensors for counting steps and estimating walking directions, without the need for extensive sensor or landmark deployment. However, misalignment between the phone's direction and the user's actual movement direction can lead to unreliable direction estimates and inaccurate location tracking. To address this issue, this paper introduces SWiLoc, an enhanced direction correction system that integrates passive WiFi sensing with smartphone-based sensing to form Correction Zones. Our two-phase approach accurately measures the user's walking directions when passing through a Correction Zone and further refines successive direction estimates outside the zones, enabling continuous and reliable tracking. In addition to direction correction, SWiLoc extends its capabilities by incorporating a localization technique that leverages corrected directions to achieve precise user localization. This extension significantly enhances the system's applicability for high-accuracy localization tasks. Additionally, our innovative Fresnel zone-based approach, which utilizes unique hardware configurations and a fundamental geometric model, ensures accurate and robust direction estimation, even in scenarios with unreliable walking directions. We evaluate SWiLoc across two real-world environments, assessing its performance under varying conditions such as environmental changes, phone orientations, walking directions, and distances. Our comprehensive experiments demonstrate that SWiLoc achieves an average 75th percentile error of 8.89 degrees in walking direction estimation and an 80th percentile error of 1.12 meters in location estimation. These figures represent reductions of 64% and 49%, respectively for direction and location estimation error, over existing state-of-the-art approaches.

**Keywords:** indoor localization; walking direction estimation; dead reckoning; smartphone sensor fusion; channel state information (CSI); WiFi sensing

## 1. Introduction

Accurately determining user location can facilitate numerous applications such as navigation, social media, and location-based marketing [1]. While GPS is the preferred method for outdoor localization, its effectiveness for indoors is restricted due to signal degradation or blockage by physical obstacles. Beyond GPS, indoor localization techniques fall into two categories: device-free and smartphone-based. However, these methods are not widely adopted in pervasive indoor settings due to their inherent constraints [2].

Smartphones have become indispensable in our daily activities and are expected to remain popular in the years to come. Technologies for indoor localization based on smartphones include WiFi RSSI [3], Pedestrian Dead Reckoning (PDR)[4], BLE-beacon[5], and camera-based methods [6]. Among these, PDR is particularly practical for widespread use as it utilizes the IMU sensors built into smartphones. While distance measurement via IMU sensors is precise, estimating walking direction is less reliable due to the misalignment between the smartphone's orientation and the actual direction of human movement. Accurate direction estimation is crucial as it significantly enhances the ability to estimate a user's accurate location within complex indoor environments, thus improving the overall

effectiveness of navigation and location-dependent applications. This directional misalignment can lead to significant errors in location determination [7].

Building on previous discussions regarding smartphone-based localization technologies, considerable research has been directed toward improving the accuracy of direction estimation using smartphones. These methodologies generally fall into two categories: those that depend on external landmarks [8–10] and those that enhance IMU sensor data through various statistical techniques [11–15]. The former requires users to consistently pass by designated landmarks, which may not always be feasible in fluid urban environments. The latter, while effective, tends to be tailored to individual users, reflecting specific gait patterns and how the device is carried, which can limit its general applicability and sensitivity.

In this paper, we improve Pedestrian Dead Reckoning (PDR)-based direction estimation by integrating it with passive WiFi sensing to enhance overall localization accuracy. Passive WiFi sensing, a widely studied device-free indoor localization method, leverages the ubiquitous presence of WiFi. By merging the benefits of smartphone-based motion sensing with WiFi sensing, we establish WiFi sensing regions (correction zones) using WiFi devices. We then utilize the precise walking distances recorded by smartphones to refine the accuracy of WiFi sensing. This integrated approach not only yields superior accuracy and reliability compared to standalone WiFi-based direction estimation but also effectively enhances the precision of location estimation, addressing the limitations inherent in smartphone-based direction estimation methodologies.

This paper is an extended version of our previous work [16], which was presented as a short paper at the MASS'23. In this version, we expanded the methodology by establishing a correlation between the phone's compass direction and the user's walking direction (we call it Phase 1); as well as by using this correlation to estimate the user's walking direction based on the phone's compass direction (Phase 2). We used this accurate walking direction to estimate user's precise location. Additionally, we extended the experimental results by conducting real-world experiments in continuous traces. Finally, we provided a more comprehensive analysis on how different phone holding positions impacts the walking direction accuracy, thereby the localization accuracy. We also compared these findings with state-of-the-art localization approaches.

In summary, our main contributions are as follows:

- In this paper, we introduce SWiLoc, a novel direction correction system that leverages passive WiFi sensing to form Correction Zones for refining smartphone-based user direction estimates. Our **two-phase** approach not only accurately measures the user's walking directions when they pass through a correction zone but also utilizes these measured directions to estimate their successive directions outside correction zones. This is done by first establishing a correlation in Phase 1 and using this correlation in Phase 2.
- Building on the first contribution, we extend SWiLoc's capabilities by implementing an accurate localization technique that uses the corrected directions to achieve precise user localization. This extension enhances the system's utility by enabling continuous and accurate tracking of the user's movements, providing a robust solution for applications requiring high localization accuracy.
- Our third contribution is the resolution of **unreliable walking directions** through our innovative and distinctive hardware configurations. We discuss and resolve the unreliable direction problem in this paper. Our model is based on the Fresnel zone-based approach that not only ensures reliable direction estimations in challenging scenarios but also significantly enhances localization accuracy.
- Our system undergoes rigorous analysis and evaluation across two real-world settings, where its performance is benchmarked against state-of-the-art methods. We thoroughly assess how various factors—such as environmental conditions, ways the phone is held, walking directions, and varying locations and distances—affect the precision of our method. The results demonstrate

that SWiLoc consistently outperforms other existing methods in both direction estimation and localization, regardless of whether they utilize WiFi sensing or smartphone sensor fusion.

The structure of this paper is organized as follows: Section 2 reviews existing solutions and identifies their limitations. Section 3 describes our system's overview and design principles. Section 4 details the methodology of our system. Section 5 offers a comprehensive description of our system's implementation. The evaluation results are presented in Section 6. The paper concludes with Section 7.

## 2. Related Works

Numerous studies have focused on precisely determining the walking direction and location of an individual using either smartphone IMU sensors or WiFi CSI (Channel State Information) [17,18]. Each method has its inherent drawbacks. In this section, we provide a concise overview of several recent and notable studies, highlighting their advantages and disadvantages.

### 2.1. Pedestrian Dead Reckoning (PDR)

Dead reckoning involves continuously calculating the distance or direction from a fixed starting point to determine the current location in indoor navigation settings. In PDR, the number of steps and the length of each step are utilized to calculate the distance a pedestrian has covered, while direction estimation helps to find the pedestrian's current heading. Understanding both the direction and the step length enables pedestrians to find their real-time positions from an initial location.

The primary methods for counting steps through a smartphone's built-in accelerometer include threshold setting, peak detection, correlation analysis, and spectral analysis, as noted in [19]. Research indicates that these existing step counting methods are highly accurate, with [19] reporting error rates below 5% for the majority of pedestrians.

Numerous step length estimation algorithms necessitate customized training for each user, as discussed in [20]. However, recent advancements have led to the development of several training-free methods[21] that accurately determine step length regardless of the individual pedestrian. In this paper, we implemented a simple yet accurate step count and step length estimation method, although our main focus is to accurately identify pedestrian's heading direction and location.

The compass is the standard method for estimating direction embedded in both Android and iOS devices. However, the discrepancy between the orientation of the phone and the actual walking direction of the user results in only moderate accuracy when using a smartphone compass for dead reckoning, as noted in [7]. Following this, we will examine and encapsulate the latest advancements in walking direction and location estimation techniques.

### 2.2. Smartphone Sensor Fusion Based Direction and Location Estimation

By sensor fusion, only using the sensor readings obtained from a smartphone (accelerometer, gyroscope, and magnetometer) can get a precise estimation of users' walking direction [7,11–15, 22–24]. These approaches calculate the cumulative acceleration of pedestrian and project it to the horizontal plane, by associating the 3D angular rotation of the phone (captured by gyroscope) with 3D accelerometer and magnetometer readings. We further divide these approaches into two sub-categories, based on whether they require user-specific training or not.

**Training-based:** The training-based approaches [11–15,22] obtain context information on human motion and device attachment to more accurately estimate their directions. To do so, they employ statistical techniques (Naive Bayes, Kalman Filter or Particle Filter), classifiers or Machine Learning models(CNN, LSTM etc.) However, these methods require labor-intensive data collection. Moreover, one specific threshold value might not work for different pedestrians, causing negative impact on accuracy. Lastly, these approaches typically demand pedestrians to walk for a long distance, thus create lag while inferring direction.

**Training-free:** Some training-free approaches (e.g, SmartPDR [23] and RMPCA [24]) assume that the pedestrian carries the phone in a fixed mode. They use principle component analysis (PCA) to filter out the noises in direction estimation. Some other approaches (e.g, Humaine [7]) can adapt to multiple mode of users holding the phone. However, all these approaches require certain patterns in user's walking and holding the phone, which results in low direction estimation accuracy when the requirements are not met.

### 2.3. Calibration-Based Direction and Location Estimation

Calibration-based direction estimation necessitates that the pedestrian traverse between two predefined landmarks. The trajectory of the pedestrian, captured while moving between these landmarks, is monitored, though the phone's compass may provide unreliable direction. Upon reaching the second landmark, the true walking direction is computed and used to adjust the correlation between the phone's compass output and the pedestrian's actual movement direction. These landmarks can include options such as LED lights [9], acoustic markers [10], or Bluetooth beacons [25,26].

Nevertheless, calibration-based direction estimation methods face several limitations: 1) they often necessitate the installation of additional hardware devices; 2) their effectiveness is limited by environmental conditions (for instance, SoundMark is ineffective in noisy settings and LiDR requires LED lights to be arranged in a specific configuration); 3) they require that pedestrians travel through two sequential locations, which introduces an additional complication.

### 2.4. Direction and Location Estimation using WiFi

Device-free direction and location estimation is a method for determining the movement direction and location of a pedestrian without any devices carried by the individual. This approach utilizes existing wireless signals, such as WiFi, within the environment. These signals reflect off the moving pedestrian, and the resulting reflections are captured by several receivers positioned throughout the area. Research such as WiDir [27], WiDar [28] and WiDar2 [29] determines the user's direction using device-free techniques by implementing various theoretical models, including the Fresnel zone, Doppler Frequency Shift (DFS), Time-of-Flight (ToF), and Angle-of-Arrival (AoA). Notably, the Fresnel zone model is distinguished by its simple geometrical features and its capability to provide accurate direction and subsequent location estimations.

The *Fresnel zone* is created when radio waves travel from a transmitter (Tx) to a receiver (Rx), traversing multiple paths that result in both constructive and destructive interference due to the alternating lengths of these paths [27]. When considering a radio wavelength  $\lambda$ , Fresnel zones consist of  $n$  ellipses, defined by the following properties:

$$|RxP| + |PTx| - |TxRx| = n\lambda/2 \quad (1)$$

where  $P$  represents a point on the  $n$ th ellipse, as depicted in Figure 1. The path length ( $|RxP| + |PTx|$ ) of the signal that is reflected or diffracted through the  $n$ th Fresnel zone boundary is  $n\lambda/2$  longer than the direct Line-of-Sight (LoS) path length ( $|TxRx|$ ). Although an infinite number of Fresnel zones exist, Raspberry Pis are limited to measuring 256 sets of Fresnel zones due to the use of 256 OFDM subcarriers. The innermost ellipsoid is referred to as the First Fresnel Zone (FFZ). Conventional Fresnel Zone approaches exhibit significant errors when users move within certain areas, which will be further explored in Section 3.2.

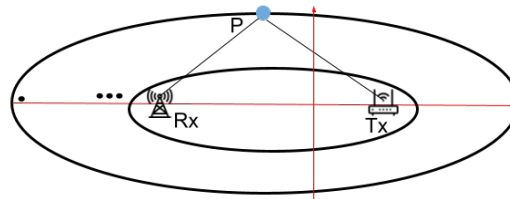


Figure 1. Geometry of Fresnel Zones.

In summary, the phone compass is inadequate for direction determination in PDR, and each direction estimation method we have explored exhibits specific limitations. We will next provide a design overview of SWiLoc.

### 3. SWiLoc System

In this section, we introduce the SWiLoc system, which integrates passive WiFi sensing with smartphone motion detection to estimate walking direction and location. Subsequently, we will explore the design considerations that shaped the development of the SWiLoc system.

#### 3.1. System Overview

SWiLoc, designed to provide walking directions and locations for pedestrians in an indoor space, comprises three primary components: 1) **phones** carried by pedestrians in arbitrary orientation; 2) **correction zones**, each measuring approximately 3x3 meters and containing a WiFi router placed at center with four WiFi receivers positioned at the corners; 3) a **server** that gathers and processes data from the users' phones and receivers through wireless links (WiFi or cellular networks), and then transmits the calculated directions back to the phones. Figure 2 illustrates the setup of our system, depicting a user navigating from location A to D via locations B and C, moving through correction zones 1 and 2, with a phone strapped to their arm.

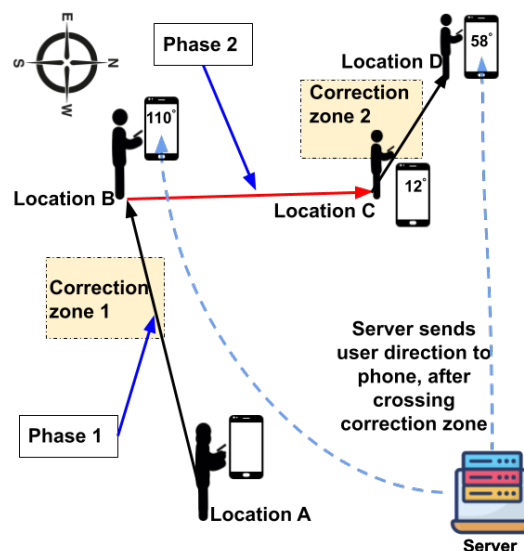


Figure 2. Illustration of SWiLoc System.

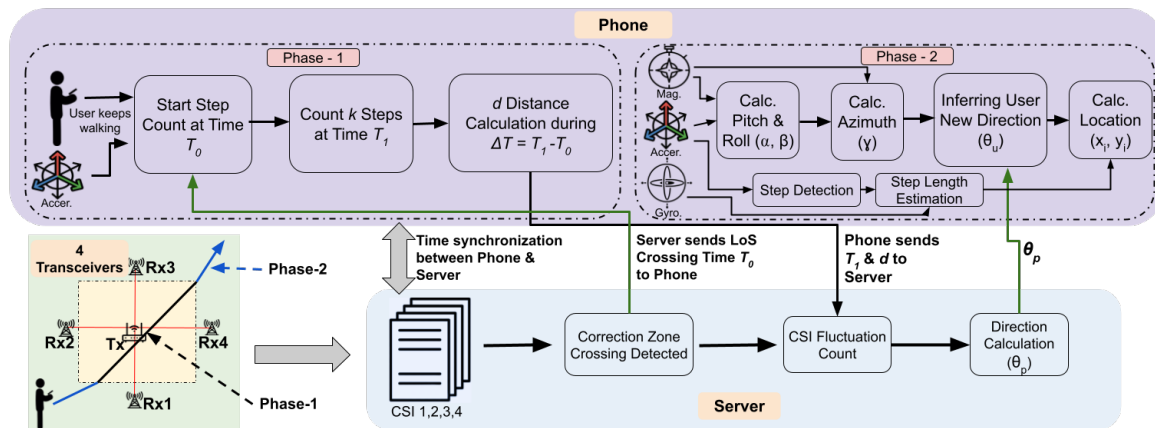


Figure 3. System Workflow of SWiLoc.

SWiLoc works in **two phases**: (Phase 1) when the user is crossing the correction zone; (Phase 2) after the user crosses the zone. In phase 1, we utilize the phone's step count data along with passive WiFi CSI information gathered by WiFi transceivers to determine the user's walking direction, and correlate their movement direction with the phone's direction; In phase 2, we collect the phone's compass direction by which we infer the user's movement direction and thereby estimate the location using the direction inferred.

To illustrate the functionality of the SWiLoc system, let's examine a simple scenario depicted in Figure 2. The user begins their journey at point A and concludes at point D, traversing the segments AB, BC, and CD. Upon entering correction zone 1, SWiLoc *phase 1* assesses their walking direction as  $110^\circ$  relative to global north and communicates this direction to their phone. At point B, the user adjusts their course by  $98^\circ$  toward global east and proceeds until reaching point C. Since the compass direction diverges from their actual walking trajectory, SWiLoc *phase 2* updates their walking direction using their previous direction and the compass adjustment, resulting in a new direction of  $12^\circ$ .

In summary, SWiLoc consists of two phases. During phase 1, we establish a correlation between the phone's compass direction and the user's walking direction. In phase 2, we use this correlation to estimate the user's walking direction based on the phone's compass direction. Regardless of how the user holds her phone, SWiLoc can accurately update the real-time walking direction, as long as the phone's relative position to the user remains unchanged. This allows SWiLoc to function with the phone held in any arbitrary position, such as in the hand palm, shirt pocket, or pant pocket. If the user changes the phone's orientation while walking, a new correlation will be established when the user passes the next correction zone.

### 3.2. SWiLoc Design Considerations

In most times, SWiLoc users stay in phase 2 and rely on the correlation between phone's direction and human's movement direction to infer their directions. The primary challenge for SWiLoc is achieving accurate walking direction and location estimates in phase 1, as phase 2 relies on the results of phase 1. When the user crosses the correction zone, SWiLoc combines the advantages of smartphone-based motion sensing and passive WiFi sensing to improve accuracy.

We opted for the Fresnel zone-based (FZ) method for passive WiFi sensing because it does not require training. Nevertheless, traditional FZ methods are prone to two major issues. Firstly, the FZ model struggles with unreliable directions, which are directions where there is no or very little variation in the CSI data. These directions are parallel or nearly parallel to the Line-of-Sight (LoS) between a transceiver pair as shown in Figure 4a. The direction estimation results become inaccurate when the user walks along these unreliable directions [27]. Secondly, FZ methods estimate direction by calculating the ratio of the number of Fresnel zones traversed by the user along the x and y axes. This approach, being an approximation, introduces errors in the estimation process.

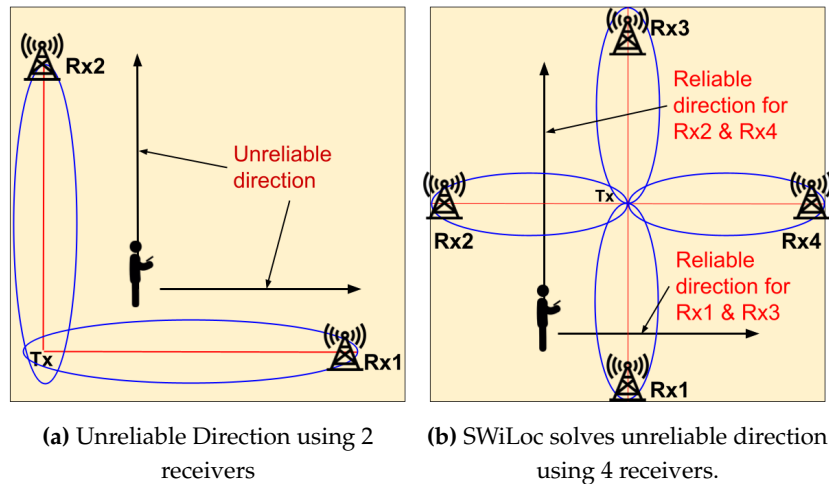


Figure 4. Towards Solving Unreliable Direction Problem.

To address the aforementioned problems, SWiLoc introduces innovative solutions, summarized below. The details will be given in the next sections.

1) To solve the unreliable direction problem, SWiLoc incorporates additional receivers as shown in Figure 4b. As two receivers placed perpendicular to each other may provide inaccurate estimation, we use opposite receivers (either Rx2 and Rx4 or Rx1 and Rx3, depending on the user's direction) for direction estimation.

2) In order to adapt to the changes in system hardware setup, SWiLoc utilizes accurate walking distance data from smartphones to improve the precision and reliability of WiFi sensing. Rather than relying on ratios, SWiLoc uses the geometrical relationship between the user's movement and its effect on the Fresnel zones as depicted in Figure 5a. This method allows SWiLoc to compute direction and location accurately without resorting to approximations.

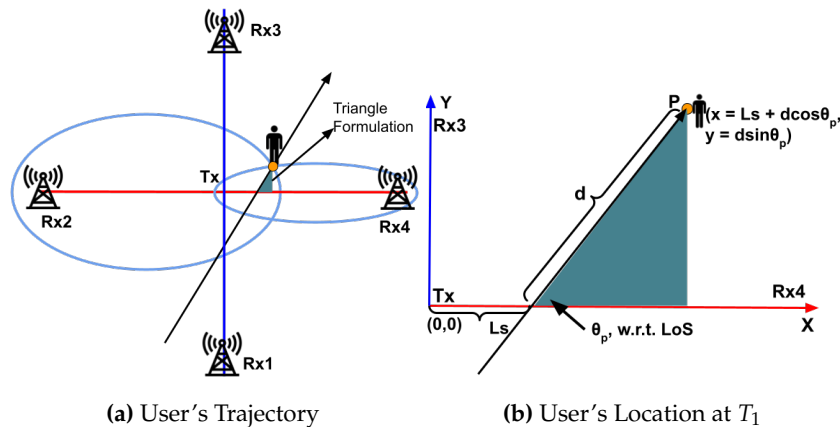


Figure 5. Geometrical Derivation of SWiLoc using Triangle.

## 4. Methodology

This section initially presents the workflow of SWiLoc, followed by comprehensive discussions on the calculation of movement direction (phase 1) and the correction of phone orientation alongside location estimation (phase 2).

### 4.1. Workflow of SWiLoc

Figure 3 illustrates the workflow of SWiLoc. To recap, SWiLoc operates in two phases: WiFi-based direction estimation when the user crosses the correction zone and phone sensor based estimation after the user leaves the zone. The steps and details of both of the phases are described as follows.

**Start:**

1. The central server integrates a Network Time Protocol (NTP) to ensure time synchronization between the smartphone and all four receivers.

#### Phase 1:

2. A user with a smartphone enters a correction zone and crosses the Line-of-Sight between a pair of WiFi transceivers at time  $T_0$ . This crossing event is identified through CSI analysis, details of which are elaborated in our system implementation section.
3. Following CSI analysis, the server transmits  $T_0$  to the smartphone, which continues to gather data from the motion sensor as the user walks. This data includes the user's step count, the phone's orientation (pitch, roll, and azimuth), and the timestamp for each step, all of which are processed and recorded by the smartphone.
4. The smartphone transmits the time  $T_1$  and distance  $d$  to the server, where  $T_1$  represents the time taken for the user to walk  $k$  additional steps after crossing the LoS and  $d$  denotes the distance traveled between  $T_0$  and  $T_1$ . The value of  $k$  is predetermined and  $d$  is calculated using the individual step length of each user.
5. The server analyzes the CSI data, calculates the fluctuation count between times  $T_0$  and  $T_1$  and determines the user's walking direction  $\theta_p$  by applying Equation (2) and Equation (3).
6. The server returns the calculated direction to the smartphone.

#### Phase 2:

7. The phone receives the user's walking direction  $\theta_p$  and maps the phone's orientation to the user's walking direction during  $T_0$  and  $T_1$  by using Equation (4).
8. User continues walking, relying on the mapping formed in the previous step to infer the user's walking direction from phone's orientation.
9. Finally, the phone computes user's location using the corrected walking direction. Phase 1 repeats when the user moves into a next correction zone.

#### 4.2. Computation of Direction in Correction Zone

Figure 5a illustrates the method for determining the user's direction using motion sensor data and WiFi sensing. g. A coordination system is established using a transmitter Tx at (0,0) and four receivers (Rx1 - Rx4), with Rx4 aligning with the positive x-axis. When the user intersects the LoS between Tx and Rx4 at time  $T_0$ , the intersection point is designated as  $(ls,0)$ . Although the CSI analysis at the receiver detects the time of LoS crossing, the exact crossing point,  $ls$ , remains undetermined. The user then walks  $k$  steps to reach point  $P$  at time  $T_1$ . As detailed in Figure 5b, the coordinates of  $P$  (user's location at  $T_1$ ) can be represented as:

$$(x = ls + d \cos \theta_p, y = d \sin \theta_p) \quad (2)$$

Here,  $ls$  represents the distance from Tx to the user's crossing point on the LoS at time  $T_0$ , and  $\theta_p$  indicates the direction the user is walking from location  $P$  towards the WiFi transceiver system's coordinate system. In the formula, the smartphone provides the distance  $d$  from the LoS crossing to point  $P$ , with  $ls$  and  $\theta_p$  remaining as the two unknown parameters.

As the user moves between  $T_0$  and  $T_1$ , they cross several Fresnel zones established between the pairs of transceivers, Tx and Rx2, and Tx and Rx4. These Fresnel zone crossings are identifiable by analyzing the CSI from the corresponding receivers, which display fluctuations in amplitude levels. For ease of understanding and representation, we simplify the sequence of Fresnel zones at location  $P$  by counting the number of zones crossed by the user [30]. We assign  $m$  and  $n$  to represent the sequence

of the Fresnel zones at location  $P$  for Rx2 and Rx4, respectively, which are determined by counting these fluctuations. Based on the Fresnel zone model, we establish the following:

$$\begin{aligned} \text{ellipse1} : \frac{(x-c)^2}{(a_1^n)^2} + \frac{(y)^2}{(b_1^n)^2} &= 1 \\ \text{ellipse2} : \frac{(x+c)^2}{(a_2^m)^2} + \frac{(y)^2}{(b_2^m)^2} &= 1 \end{aligned} \quad (3)$$

here  $a_1^n$ ,  $b_1^n$ ,  $a_2^m$  and  $b_2^m$  represent the major and minor axes of the  $n$ th and  $m$ th ellipses respectively (refer to Figure 6). These terms are derived from Fresnel zone theory using the values of  $m$  and  $n$  and  $c$  is half the distance between the transmitter and receiver, as established during the system setup. By substituting the  $x$ ,  $y$  values from Equation (2) to Equation (3), we can determine the user's direction  $\theta_p$  and the LoS crossing point  $(l_s, 0)$ .

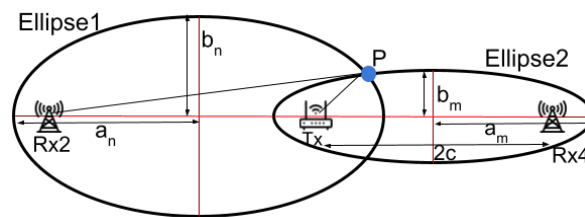


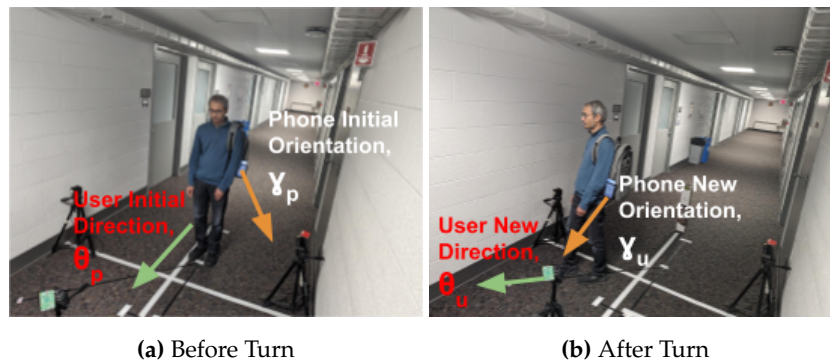
Figure 6. The  $m$ th and  $n$ th Fresnel Zone for Rx2 and Rx4.

#### 4.3. Phone-Based Direction Estimation

Different ways of user carrying the phone cause different misalignments between the phone's orientation and the user's walking direction. As long as the user doesn't change the way they hold the phone, the mapping between the two directions doesn't change. SWiLoc builds this mapping from  $\theta_p$  and the phone's orientation at location  $P$ , as shown by Equation (4).

$$\theta_u = (\theta_p + (\gamma_u - \gamma_p)) \mod 360 \quad (4)$$

where  $\gamma_u$  is the phone's new orientation and  $\theta_u$  is the user's new direction. Figure 7 illustrates this mapping: user's current direction obtained from correction zone is  $\theta_p$  and phone's current orientation is  $\gamma_p$ . After user takes a turn, user's new direction  $\theta_u$  is inferred by correcting phone's new orientation  $\gamma_u$  using the mapping in Equation (4). We calculate the phone's orientation towards the global north (i.e., azimuth, which is the rotational angle about the phone's  $z$  axis) by its 3D accelerator's and magnetometer's readings.



(a) Before Turn (b) After Turn

Figure 7. Mapping Phone Orientation to User Direction.

#### 4.4. Location Estimation

With the accurate direction estimation, we can now accurately estimate the user's location, that involves 3 main steps: step detection, step length estimation, and location calculation. By detecting

each step, estimating the length of each step, and iteratively updating the position, we can continuously track the user's movement and calculate their new location.

#### 4.4.1. Step Detection

To detect steps using smartphone accelerometer, we first compute the magnitude of the acceleration vector to combine the three axes and reduce noise using  $a_m(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}$ . Next, we apply a low-pass filter to the magnitude signal to remove high-frequency noise, which can be done using  $a_f(t) = \alpha \cdot a_m(t) + (1 - \alpha) \cdot a_f(t - 1)$ , where  $\alpha$  is the smoothing factor. Finally, peak detection in the filtered signal is done by identifying local maxima that exceed a predefined threshold  $T$ , using the condition  $a_f(t) > T$  and  $a_f(t) > a_f(t - 1)$  and  $a_f(t) > a_f(t + 1)$  to confirm a peak at time  $t$ . By applying these steps, we can accurately detect steps based on the identified peaks in the filtered acceleration data.

#### 4.4.2. Step Length Estimation

To estimate step length using the step detection algorithm, we collect accelerometer and gyroscope data from the smartphone during walking and identify steps using the filtered acceleration magnitude. For each detected step, we extract features: the mean acceleration magnitude  $Acc_{mean} = \frac{1}{n} \sum_{i=1}^n a_m(i)$ , peak acceleration  $Acc_{peak} = \max(a_m(i))$ , mean gyroscope reading  $Gyro_{mean} = \frac{1}{n} \sum_{i=1}^n \sqrt{\omega_x(i)^2 + \omega_y(i)^2 + \omega_z(i)^2}$ , and step frequency  $S_{freq} = \frac{1}{\Delta t}$ . These features are then combined into a feature vector  $x$  and given as follows:  $x = [Acc_{mean} \quad Acc_{peak} \quad Gyro_{mean} \quad S_{freq}]^T$ . The step length  $L$  is estimated using a dot product with a calibration vector  $w$ , such that  $L = w \cdot x$ . This approach leverages sensor data to estimate step length accurately without requiring the user's height.

#### 4.4.3. Location Calculation

Given that the user, after reaching at point  $P$ , continues walking in the new direction  $\theta_u$ , then the new position  $(x_i, y_i)$  can be calculated iteratively using the following formulas:

$$\begin{aligned} x_i &= x_{i-1} + L_i \cdot \cos(\theta_u \cdot \pi/180) \\ y_i &= y_{i-1} + L_i \cdot \sin(\theta_u \cdot \pi/180) \end{aligned} \quad (5)$$

where  $x_{i-1}$  and  $y_{i-1}$  are the coordinates of the previous position and  $L_i$  is the estimated length of the  $i$ -th step. By updating the position after each detected step, we can continuously track the user's location in the new direction  $\theta_u$ .

## 5. System Implementation

In this section, we outline the hardware configuration and the procedural steps required to implement the SWiLoc software.

### 5.1. Hardware Setup

Each correction zone consists of four receivers and one transmitter. We utilize the Raspberry Pi 4B as the receiver to gather CSI data, chosen for its accessibility and affordability. However, the standard firmware of the Broadcom WiFi chip does not support CSI data capture. Thus, we employed modified firmware [18] from Nexmon that enables CSI data capture. This firmware conveys CSI data to the host system by embedding it within transport layer payloads in frames. For transmission, a TP-Link Archer A7 router operating on the 5GHz band was used. To initiate data transmission, we ping the router every 5ms, prompting it to send a pong packet received by the four receivers.

We use a desktop equipped with an Intel Core i7 processor and 8GB of RAM as the server, which connects to all receivers via Ethernet. This server processes the CSI data and performs all necessary

calculations for direction and location computation. Additionally, it operates a Network Time Protocol (NTP) server to manage time synchronization across all system components.

Finally, a Huawei Nova 3i smartphone serves as the user's device. It communicates with the server via a TCP socket and automatically synchronizes its time with the NTP server.

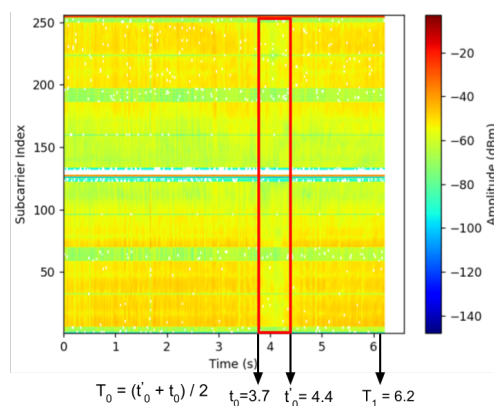
## 5.2. Software Implementation

We implement the following procedures on the server, with the android App installed on the phone:

### 5.2.1. LoS Crossing Detection

The LoS crossing detection step identifies a specific pattern in the CSI readings from four receivers and determines the time  $T_0$  as the LoS crossing moment, along with the specific receiver (e.g., Rx1, 2, 3, or 4) where the user intersects the LoS. A distinct pattern emerges in the CSI data as the user crosses the LoS. Figure 8 displays this pattern (highlighted within a red rectangular box) occurring between  $t_0 = 3.7\text{sec}$  and  $t'_0 = 4.4\text{sec}$ . The pattern reveals that the energy amplitude drops below  $-55\text{dBm}$  during this interval. This reduction occurs because the user obstructs the radio signal between the transmitter and receiver, leading to decreased energy in the CSI readings. Given the brief duration of the LoS crossing (less than 1 sec), we calculate  $T_0$  as the midpoint of the interval,  $T_0 = (t_0 + t'_0)/2$ , representing the average time of crossing.

We developed a Python script that employs a simple search technique to identify the pattern. The script looks for consecutive amplitude values that meet or fall below a set threshold in the time domain. The size of the pattern window varies depending on the user's body shape, walking speed, and the hardware specifications. Through repeated experiments, we observed that in our hardware and correction zone setup, a person of average size walking at a speed between 0.5 to 1 m/s can be accurately detected by setting a threshold amplitude value of  $-55\text{dBm}$ . This parameter has been used for our evaluations.



**Figure 8.** LoS Crossing Detection.

### 5.2.2. CSI Fluctuation Count

This phase determines  $m$  and  $n$  by counting fluctuations in the CSI data collected by Rx2 and Rx4 between  $T_0$  and  $T_1$ , where  $T_0$  is identified through LoS crossing detection and  $T_1$  is provided by the smartphone. As a person moves along a radio propagation path, they create peaks and valleys in the CSI data within the Fresnel zone. By counting these peaks and valleys, we can ascertain which Fresnel zone the user occupies after walking  $k$  steps.

CSI data captured by the Raspberry Pi contains noise. To mitigate this, we employ the Least-square smoothing filter [31], which effectively smooths the CSI data while minimally altering the waveform. This filter creates a polynomial fit based on a predefined number of input samples, known as a sample window. After thorough testing across various environments, walking directions, and distances,

we have empirically set the window size at 51. Figure 9 displays the CSI data before and after the smoothing process. We then utilize the *find\_peaks* function from the *SciPy* package to count the fluctuations. Through experimental evaluation, we set the minimum height as 750 (measured in raw amplitude values, not dBm) to identify a fluctuation.

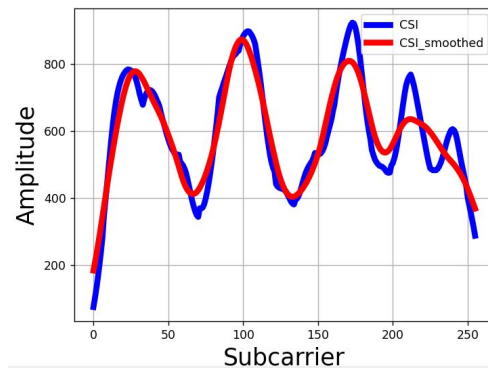


Figure 9. Before and After CSI Smoothing.

### 5.2.3. Direction Calculation

Once  $m$  and  $n$  are determined, the subsequent step involves solving the two ellipse equations presented in Equation (3). For a given  $n$ , the boundary of the  $n$ -th Fresnel zone,  $b_1^n$ , formed by Rx2, is specified by the Fresnel zone model as follows:

$$b_1^n = \{|Rx_2P| + |PTx| - |TxRx_2| = n\lambda/2\} \quad (6)$$

Here,  $\lambda$  represents a parameter set by the radio wavelength. Similarly, the boundary for the  $m$ -th Fresnel zone  $b_2^m$ , associated with Rx4, is also determined. Furthermore, for *ellipse1* and *ellipse2*, three additional properties derived from Fresnel zone theory are essential for calculating  $a_1^n$ ,  $b_1^n$ ,  $a_2^m$  and  $b_2^m$  in Equation (3). For clarity, the three properties for *ellipse1* are outlined below:

$$\begin{aligned} |Rx_2P| + |PTx| &= 2a_1^n \\ |TxRx_2| &= 2c \\ (a_1^n)^2 - c^2 &= (b_1^n)^2 \end{aligned} \quad (7)$$

By solving Equations 6 and 7, we derive  $b_1^n$  as follows:

$$b_1^n = \frac{\sqrt{n\lambda^2 + 8n\lambda c}}{4} \quad (8)$$

Inserting  $b_1^n$  into Equation (7), allows us to calculate  $a_1^n$ , with  $c$  as a predetermined variable (from Equation (3)). Similarly, the values for  $a_2^m$  and  $b_2^m$  for *ellipse2* can be determined. This allows us to solve Equation (3) and determine the two unknown variables  $\theta_p$  (direction) and  $ls$ , as illustrated in Figure 5b. With  $\theta_p$  and  $ls$  distance established, the user's location coordinate  $x$  and  $y$  at point  $P$  can readily be calculated using Equation (2).

### 5.2.4. Location Calculation

Once we have determined the calculated direction ( $\theta_p$ ), we apply Equation (4) to deduce the user's new movement direction ( $\theta_u$ ) based on changes in the phone's orientation, which occur due to the user's changed direction. Finally, we use Equation (5) to compute the user's new location ( $x_i, y_i$ ) by utilizing the updated direction ( $\theta_u$ ).

### 5.2.5. SWiLoc App Implementation

Our app development is based on an open-source compass application [32]. Specifically, our app (1) interacts with the server to obtain  $T_0$  and  $\theta_p$ , and transmits  $T_1$  and  $d$ ; (2) detects the user's steps and logs the time and the phone's orientation at each step; (3) calculates the real-time direction of the user by Equation (4); (4) computes the successive locations of the user using the Equation (5).

## 6. Evaluation

This section outlines the procedure and findings of our assessment. We first measured the accuracy achieved by phase 1 only, and then further measured our overall accuracy, i.e., phases 1 and 2.

### 6.1. Testbed Setup

We carried out experiments in two distinct indoor settings to assess the performance of SWiLoc effectively. These included a lab office measuring 7.5m by 6m and a large empty corridor of 46m by 3m. In each environment, we positioned four receivers to create a 3m by 3m correction zone. The lab office, equipped with four tables, four chairs, and several desktops and monitors, is characterized by rich multi-path reflections. Conversely, the corridor is indicative of an environment with static reflections.

The receivers, mounted on tripods 50cm above the floor, are depicted in Figure 10. Each Raspberry Pi was positioned 1.5m from the WiFi router, which was affixed to the ceiling directly above the midpoint of the four receivers. Keeping the receivers elevated from the ground is crucial to minimize the reflection of radio waves off the floor.

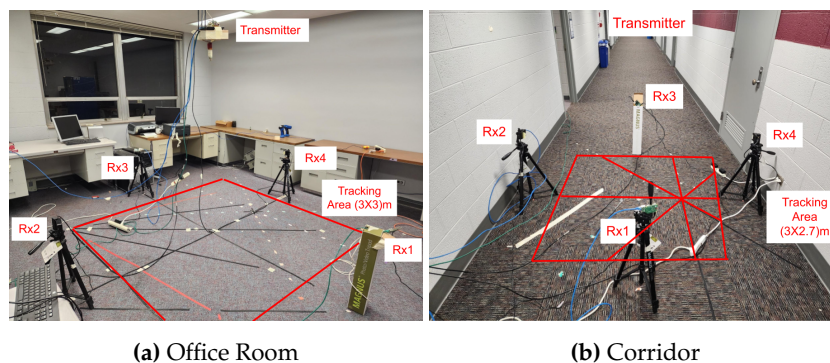


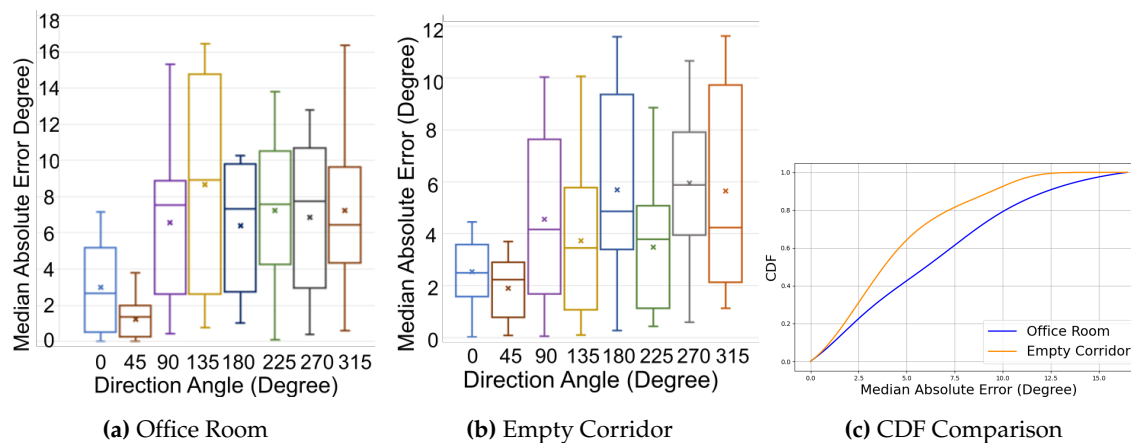
Figure 10. Testbed Setup.

### 6.2. Performance Evaluation for Phase 1 Only

Primarily, we evaluated the WiFi sensing method used in phase 1 by testing the accuracy of direction estimation across eight basic paths. These paths intersected the WiFi router from eight different directions, each separated by 45 degrees: 0, 45, 90, 135, 180, 225, 270, and 315 degrees, relative to the line from Rx2 to Rx4. Since smartphone-based distance estimation is recognized for its accuracy and is not the primary focus of our study, we proceeded under the assumption that the distance  $d$  recorded by the smartphone was precise. Our experiment focused solely on measuring the errors originating from our WiFi sensing method. To accurately set the directions for each path, we employed a digital protractor and placed markers on the floor. Three volunteers were instructed to walk from a specific start point to an endpoint, ensuring their torsos were aligned with the marked line. We conducted eight repetitions for each data collection, collecting a total of 384 sets of WiFi CSI data across two environments, involving three volunteers, eight paths, and eight repeats each.

Figure 11 presents the Median Absolute Error (MAE) for eight fundamental paths within both an office room and an empty corridor. Figure 11a indicates that the overall MAE in the office room is 6 degrees, with a standard deviation of 5.41 degrees. Figure 11b displays that the overall MAE in the empty corridor is approximately 5 degrees, with a standard deviation of 4.04 degrees. Figure 11c illustrates that, in contrast to the office room, which is affected by substantial multi-path reflections,

the empty corridor results in lower errors, with a 75th percentile error of about 6 degrees compared to 8 degrees in the office room. Notably, the 45-degree angle consistently exhibits less than 4 degrees of error in the upper quantile for both environments. The boxplots for both environments demonstrate that our approach accurately estimates unreliable directions (0, 90, 180 and 270 degrees).



**Figure 11.** Phase 1 Accuracy in 2 Different Environments.

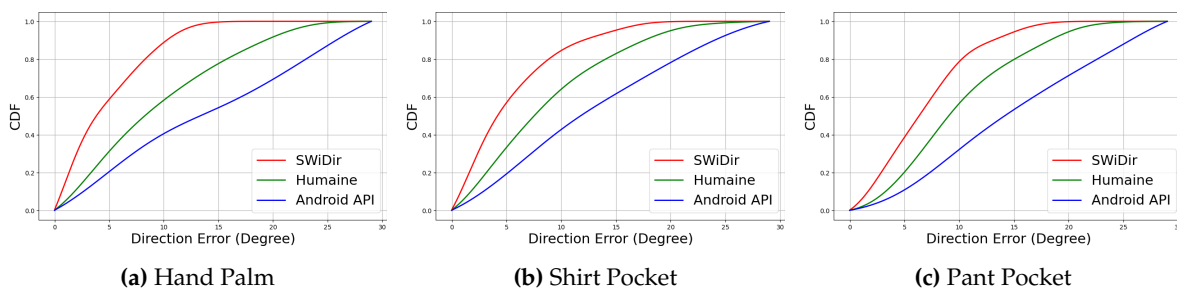
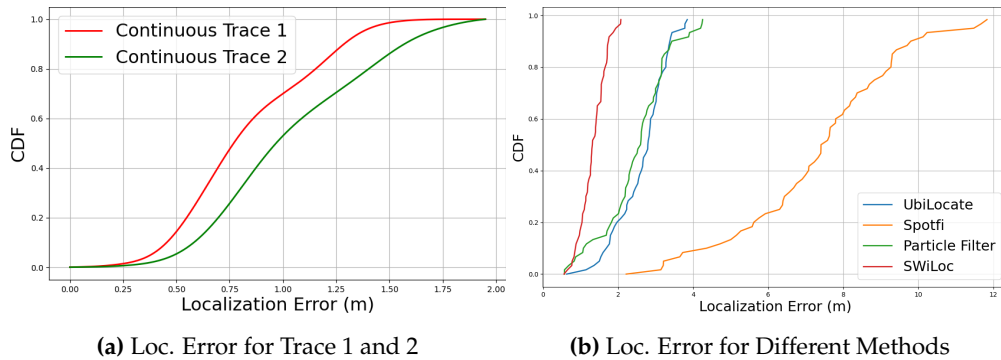
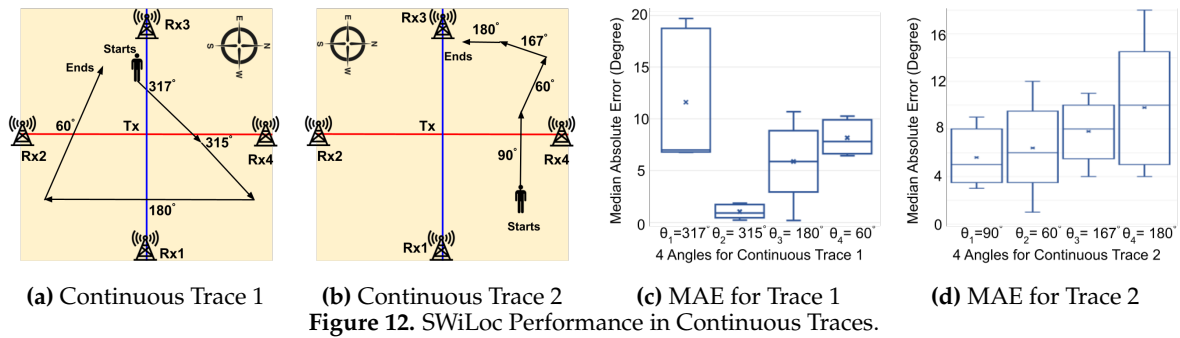
### 6.3. Performance Evaluation for SWiLoc

We further incorporated a phone to evaluate SWiLoc's overall accuracy (including both phase 1 and phase 2). Following our previous testing approach, we pre-defined 2 continuous traces (namely Trace 1 and 2) in corridor and demarcated the path on the ground to guide testers. Trace 1 and 2 have a length of 6.7m and 4.8m respectively. These continuous traces have multiple segments each, with some segments involving LoS crossing and other not. Five volunteers held the phone in 3 different positions, and the phone interacted with the server for direction estimation. To obtain the ground truth direction during walking, one observer used a laptop which was synchronized with the NTP server to record the time when testers started or took a turn. We repeated each data collection 5 times.

Figure 12 illustrates the continuous traces and their corresponding accuracy for direction estimation. Trace 1 and trace 2 have 3 and 4 segments respectively. After each segment of walk, our tester changed their directions and the phone inferred their new directions based on the mapping obtained from phase 1, till the tester crossed another LoS. We observed that segment 2 exhibited higher accuracy than other segments in trace 1.

We then infer the tester's location with every 1 sec interval using the estimated walking direction of the tester. As mentioned previously, 5 volunteers walked through these 2 traces with an average speed of 0.63m/s, with minimum and maximum speed of 0.51m/s and 0.96m/s respectively. Therefore, our system calculated approximately 300 locations (around 12 locations  $\times$  5 volunteers  $\times$  5 repetition) for trace 1 and 200 locations (around 8 locations  $\times$  5 volunteers  $\times$  5 repetition) for trace 2. Figure 13a shows the Cumulative Distribution Function (CDF) for localization error for both of the traces. Trace 1 exhibits less error than trace 2, as trace 1 consists of 4 LoS crossing points. On the other hand, trace 2 has only 1 LoS crossing point.

We also tested how different ways of holding the phone impacted the walking direction accuracy. Figure 14a- 14c shows the CDF graphs for 3 hold positions, comparing SWiLoc with two other systems, Humaine [7] and Android's builtin compass. The CDF results show that SWiLoc achieves the best accuracy regardless of the phone holding position. These results provide evidence to support our claim that combining active smartphone-based PDR with passive WiFi-sensing-based correction zones yields greater accuracy than existing state-of-the-art systems.



In addition to that, we computed the median and 75th percentile accuracy achieved by SWiLoc across all test scenarios and compared these figures with three other state-of-the-art works: WiDir [27], Widar [28] and WalkCompass [22]. WiDir and Widar utilize passive WiFi sensing, whereas WalkCompass relies on smartphone technology. We derived the accuracies of these methods from their respective publications, as we did not have access to their source codes. As detailed in Table 1, SWiLoc surpasses the performance of these models in terms of 75th percentile and median errors, although Widar records a slightly lower median error of 5 degrees compared to SWiLoc's 6 degrees. Against WalkCompass, the closest competitor, SWiLoc achieves a 64% reduction in the 75th percentile error. This superior performance is due to the use of a geometrical relationship that avoids the cumulative errors commonly seen in other models.

**Table 1.** Comparing SWiLoc's Walking Direction Accuracy (Phase 1) with Other Methods.

Approaches	75th Percentile Error (in degree)	Median Error (in degree)
WiDir	23	10
WalkCompass	14.2	8
WiDar	18	5
<b>SWiLoc (Phase 1)</b>	<b>8.89</b>	<b>6</b>

Finally, we compared our localization results with the state-of-the-arts localization approaches, shown in Table 2. Among the approaches, UbiLocate [33], Spotfi [34], Spring [35], Fusic [36] are based on passive sensing, while Kalman-filter [37], SmartPDR [23], Particle-filter based [38], LSTMLoc [39] are active fusion based. SWiLoc outperformed both the active and passive based localization approaches by achieving 1.12m of 80-th percentile localization error. Among all the methods, we implemented UbiLocate, Spotfi and Particle Filter based methods and tested for only Trace 2 holding the phone on hand palm. Figure 13b demonstrated that SWiLoc achieved highest localization accuracy among the 3 other methods. Please note that we employed Intel 5300 NIC for implementing Spotfi and UbiLocate testbed and utilized Linux CSI tool [40] to collect and process CSI data.

**Table 2.** Comparing SWiLoc's Localization Accuracy (Phase 1 & 2) with Other Methods.

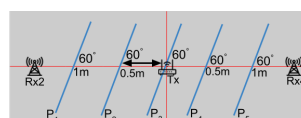
Approaches	80th Percentile Localization Error (m)	Base Sensing Method
UbiLocate	2.2	Passive
Spotfi	6.0	Passive
Spring	3.7	Passive
Fusic	3.4	Passive
Kalman-filter based	4.6	Active Fusion
PDRLoc	6.71	Active Fusion
Particle-filter based	2.21	Active Fusion
LSTMLoc	2.36	Active Fusion (Training-based)
<b>SWiLoc (Phase 1 &amp; 2)</b>	<b>1.12</b>	<b>Active Fusion &amp; Passive, Training free</b>

#### 6.4. Sensitivity Analysis of SWiLoc

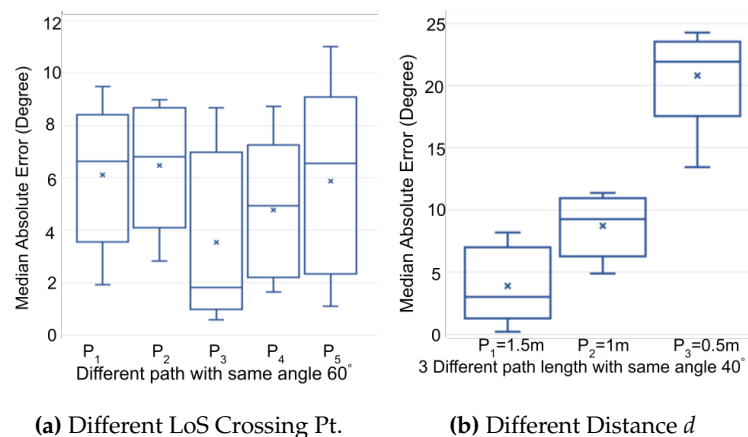
In this subsection, we analyzed the sensitivity of SWiLoc by examining how varying LoS crossing locations, distances and direction accuracy impact overall system performance, particularly focusing on direction and localization accuracy.

##### 6.4.1. Impact of Varying LoS Crossing Locations

We evaluated the impact of varying LoS crossing locations on accuracy. As depicted in Figure 15, the testers traversed the LoS between Rx2 and Rx4 maintaining a consistent walking direction, but at different crossing points. These paths ( $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$  and  $P_5$ ), each 3m in length and spaced 0.5m apart, are further analyzed in Figure 16a. The results show that the direction estimation for all paths exhibited a median error of less than 7 degrees, demonstrating that our WiFi sensing method performs well across various LoS crossing points.



**Figure 15.** Paths with Different LoS Crossing Locations.



(a) Different LoS Crossing Pt.

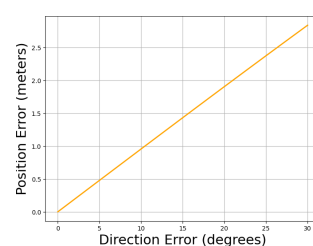
(b) Different Distance  $d$ **Figure 16.** Impact of LoS Crossing Locations and Distances.

#### 6.4.2. Impact of Distance $d$ on Direction Accuracy

We noticed that the influence of the human body on accuracy is significant. While our models treat the human body as a point, the actual width of the body impacts the CSI reflections and Fresnel zone fluctuation counts. This effect diminishes if the testers walk a longer distance post-LoS crossing. Consequently, we conducted further experiments to assess how the distance  $d$  (the path length after crossing the LoS) affects accuracy. The testers crossed the LoS maintaining a constant walking direction but varying walking distances  $d$  (i.e., 0.5m, 1m, 1.5m). Figure 16b shows that shorter distances  $d$  lead to greater errors. From these findings, it is advisable for developers setting up correction zones to ensure that area allows users to walk at least 0.5m after crossing the LoS to minimize errors.

#### 6.4.3. Impact of Direction Accuracy on Localization Accuracy

While it is evident that greater directional error leads to increased localization inaccuracies, quantifying this error propagation is crucial. Therefore, we have systematically measured how errors in direction influence localization accuracy. Figure 17 shows that as direction errors increase, the position error grows non-linearly, indicating that precise direction estimation is crucial. Moderate to large direction errors, like 30 degrees, lead to substantial deviations in the final location, emphasizing the importance of minimizing direction errors to maintain accurate localization.

**Figure 17.** Impact of Direction Error on Loc. Accuracy.

### 6.5. Discussion

To the best of our knowledge, SWiLoc represents the first attempt to merge passive WiFi sensing with smartphone-based direction estimation. Our evaluations suggest that SWiLoc delivers robust accuracy compared to other cutting-edge models. Additionally, it requires no training, enabling it to be deployed in a plug-and-play manner.

We also notice some of its limitations, as stated below:

**1) Requirements on Physical Deployments:** Similar to other RF-based human sensing applications like patient monitoring and gesture recognition, SWiLoc is affected by complex multi-path environments, and its accuracy significantly decreases when two individuals enter the same correction zone at the same time. Additionally, it is necessary for the pedestrian to walk at least 0.5 meters after crossing

the LoS. Therefore, when implementing SWiLoc, developers must ensure these conditions are met to maintain system effectiveness.

**2) Assumption on Phone's Orientation:** Phase 2 of SWiLoc relies on the assumption that the mapping between user's direction and phone's orientation won't change. However, sometimes a pedestrian may change how they carry the phone. Considering this, it is necessary to deploy multiple correction zones to update the mapping. Besides, as a future work direction, we plan to integrate our approach with other sensor fusion-based approaches such as WalkCompass.

## 7. Conclusions

In this paper, we introduce SWiLoc, a system designed to precisely estimate movement direction and location by integrating passive WiFi sensing with smartphone-based direction estimation. Our innovative approach incorporates extra receivers to enhance WiFi sensing and employs a geometric method to improve direction and thereby localization accuracy. SWiLoc, implemented using Raspberry Pi, demonstrates through comprehensive testing an average 80th percentile error of 1.12 meter in location estimation, achieving a 49% reduction over existing state-of-the-art methods.

**Author Contributions:** Conceptualization, K.M. and Z.S.; methodology, K.M. and Z.S.; software, K.D. and K.M.; validation, K.D. and K.M.; formal analysis, K.M.; investigation, K.M. and K.D.; resources, Z.S. and M.A.; data curation, K.M. and K.D.; writing—original draft preparation, K.M. and Z.S.; writing—review and editing, K.M., Z.S., K.D. and M.A.; visualization, K.M. and K.D.; supervision, Z.S. and M.A.; project administration, Z.S. and M.A.; funding acquisition, Z.S. and M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is supported by NSF through grant 2104337.

**Data Availability Statement:** Not applicable.

**Use of Artificial Intelligence:** AI or AI-assisted tools were not used in drafting any aspect of this manuscript.

**Conflicts of Interest:** The authors report no conflicts of interest.

## References

1. Farahsari, P.S.; Farahzadi, A.; Rezazadeh, J.; Bagheri, A. A survey on indoor positioning systems for IoT-based applications. *IEEE Internet of Things Journal* **2022**, *9*, 7680–7699.
2. Zafari, F.; Gkelias, A.; Leung, K.K. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials* **2019**, *21*, 2568–2599.
3. Xue, W.; Qiu, W.; Hua, X.; Yu, K. Improved Wi-Fi RSSI measurement for indoor localization. *IEEE Sensors Journal* **2017**, *17*.
4. Sung, K.; Kim, H. Bayesian navigation system with particle filtering and dead reckoning in urban canyon environments. 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON). IEEE, 2012, pp. 73–75.
5. Spachos, P.; Plataniotis, K.N. BLE beacons for indoor positioning at an interactive IoT-based smart museum. *IEEE Systems Journal* **2020**, *14*, 3483–3493.
6. Mulloni, A.; Wagner, D.; Barakonyi, I.; Schmalstieg, D. Indoor positioning and navigation with camera phones. *IEEE Pervasive Computing* **2009**, *8*, 22–31.
7. Mohssen, N.; Momtaz, R.; Aly, H.; Youssef, M. It's the human that matters: Accurate user orientation estimation for mobile computing applications. *arXiv preprint arXiv:1411.2156* **2014**.
8. Yu, N.; Zhan, X.; Zhao, S.; Wu, Y.; Feng, R. A precise dead reckoning algorithm based on Bluetooth and multiple sensors. *IEEE Internet of Things Journal* **2017**, *5*, 336–351.
9. Hussain, B.; Wang, Y.; Chen, R.; Cheng, H.C.; Yue, C.P. Lidr: Visible-light-communication-assisted dead reckoning for accurate indoor localization. *IEEE Internet of Things Journal* **2022**, *9*, 15742–15755.
10. Chen, H.; Li, F.; Wang, Y. SoundMark: Accurate indoor localization via peer-assisted dead reckoning. *IEEE Internet of Things Journal* **2018**, *5*, 4803–4815.
11. Yan, D.; Shi, C.; Li, T.; Li, Y. FlexPDR: Fully flexible pedestrian dead reckoning using online multimode recognition and time-series decomposition. *IEEE Internet of Things Journal* **2022**, *9*, 15240–15254.

12. Perul, J.; Renaudin, V. Learning individual models to estimate the walking direction of mobile phone users. *IEEE Sensors Journal* **2019**, *19*, 12306–12315.
13. Xiao, Z.; Wen, H.; Markham, A.; Trigoni, N. Robust indoor positioning with lifelong learning. *IEEE Journal on Selected Areas in Communications* **2015**, *33*, 2287–2301.
14. Lee, J.S.; Huang, S.M. An experimental heuristic approach to multi-pose pedestrian dead reckoning without using magnetometers for indoor localization. *IEEE Sensors Journal* **2019**, *19*.
15. Wang, Q.; Luo, H.; Xiong, H.; Men, A.; Zhao, F.; Xia, M.; Ou, C. Pedestrian dead reckoning based on walking pattern recognition and online magnetic fingerprint trajectory calibration. *IEEE Internet of Things Journal* **2020**, *8*, 2011–2026.
16. Mottakin, K.; Davuluri, K.; Song, Z. SWiDir: Enhancing Smartphone-based Walking Direction Estimation with Passive WiFi Sensing. 2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS). IEEE, 2023, pp. 599–605.
17. Lin, Y.; Dong, W.; Li, B.; Gao, Y. TinyCSI: A Rapid Development Framework for CSI-based Sensing Applications. 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2020, pp. 549–557. doi:10.1109/MASS50613.2020.00073.
18. Gringoli, F.; Schulz, M.; Link, J.; Hollick, M. Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets. Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, 2019, pp. 21–28.
19. Gu, F.; Khoshelham, K.; Shang, J.; Yu, F.; Wei, Z. Robust and accurate smartphone-based step counting for indoor localization. *IEEE Sensors Journal* **2017**, *17*, 3453–3460.
20. Vandermeeren, S.; Bruneel, H.; Steendam, H. Feature selection for machine learning based step length estimation algorithms. *Sensors* **2020**, *20*, 778.
21. Bo, F.; Li, J.; Wang, W. Mode-Independent Stride Length Estimation With IMUs in Smartphones. *IEEE Sensors Journal* **2022**, *22*, 5824–5833.
22. Roy, N.; Wang, H.; Roy Choudhury, R. I am a smartphone and i can tell my user's walking direction. Proceedings of the 12th annual international conference on Mobile systems, applications, and services, 2014, pp. 329–342.
23. Kang, W.; Han, Y. SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal* **2014**, *15*, 2906–2916.
24. Deng, Z.A.; Wang, G.; Hu, Y.; Wu, D. Heading estimation for indoor pedestrian navigation using a smartphone in the pocket. *Sensors* **2015**, *15*, 21518–21536.
25. Hu, Y.; Qian, F.; Yin, Z.; Li, Z.; Ji, Z.; Han, Y.; Xu, Q.; Jiang, W. Experience: Practical indoor localization for malls. MobiCom'22, 2022, pp. 82–93.
26. Chen, Z.; Zhu, Q.; Soh, Y.C. Smartphone inertial sensor-based indoor localization and tracking with iBeacon corrections. *IEEE Transactions on Industrial Informatics* **2016**, *12*, 1540–1549.
27. Wu, D.; Zhang, D.; Xu, C.; Wang, Y.; Wang, H. WiDir: walking direction estimation using wireless signals. Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing, 2016, pp. 351–362.
28. Qian, K.; Wu, C.; Yang, Z.; Liu, Y.; Jamieson, K. Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi. Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2017, pp. 1–10.
29. Qian, K.; Wu, C.; Zhang, Y.; Zhang, G.; Yang, Z.; Liu, Y. Widar2. 0: Passive human tracking with a single Wi-Fi link. Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, 2018, pp. 350–361.
30. Ahad, M.A.R.; Mahbub, U.; Rahman, T. *Contactless human activity analysis*; Vol. 200, Springer, 2021.
31. Schafer, R.W. What is a Savitzky-Golay filter?[lecture notes]. *IEEE Signal processing magazine* **2011**, *28*, 111–117.
32. iutinv. Simple android compass implementation, 2018.
33. Pizarro, A.B.; Beltrán, J.P.; Cominelli, M.; Gringoli, F.; Widmer, J. Accurate ubiquitous localization with off-the-shelf IEEE 802.11 ac devices. Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, 2021, pp. 241–254.
34. Kotaru, M.; Joshi, K.; Bharadia, D.; Katti, S. Spotfi: Decimeter level localization using wifi. Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, 2015, pp. 269–282.

35. Rea, M.; Abrudan, T.E.; Giustiniano, D.; Claussen, H.; Kolmonen, V.M. Smartphone positioning with radio measurements from a single wifi access point. *Proceedings of the 15th international conference on emerging networking experiments and technologies*, 2019, pp. 200–206.
36. Jiokeng, K.; Jakllari, G.; Tchana, A.; Beylot, A.L. When FTM discovered MUSIC: Accurate WiFi-based ranging in the presence of multipath. *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1857–1866.
37. Chen, Z.; Zou, H.; Jiang, H.; Zhu, Q.; Soh, Y.C.; Xie, L. Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. *Sensors* **2015**, *15*, 715–732.
38. Sun, M.; Wang, Y.; Xu, S.; Cao, H.; Si, M. Indoor positioning integrating PDR/geomagnetic positioning based on the genetic-particle filter. *Applied Sciences* **2020**, *10*, 668.
39. Zhang, M.; Jia, J.; Chen, J.; Deng, Y.; Wang, X.; Aghvami, A.H. Indoor localization fusing wifi with smartphone inertial sensors using lstm networks. *IEEE Internet of Things Journal* **2021**, *8*, 13608–13623.
40. Halperin, D.; Hu, W.; Sheth, A.; Wetherall, D. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM computer communication review* **2011**, *41*, 53–53.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.