2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 979-8-3503-2433-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/MASS58611.2023.00080

SWiDir: Enhancing Smartphone-based Walking Direction Estimation with Passive WiFi Sensing

Khairul Mottakin, Kiran Davuluri, Zheng Song
Dept. of Computer and Information Science, University of Michigan-Dearborn, MI, USA
{khairulm, kirandav, zhesong}@umich.edu

***Abstract*—Dead reckoning is a promising yet overlooked smartphone based indoor localization technology. It relies on phone-mounted sensors for counting steps and estimating walking directions, requiring no massive deployment of additional sensors or landmarks. However, it suffers from the misalignment between phone's direction and human's movement direction, which makes the estimated walking direction unreliable and eventually leads to inaccurate location estimation. To solve this problem, this paper introduces SWiDir, an approach that calibrates walking direction by integrating active smartphone sensing with passive WiFi sensing. SWiDir deploys a few WiFi devices to form a correction zone, and use their WiFi Channel State Information (CSI) to infer human's movement inside the zone. We adopt the training-free WiFi Fresnel Zone model, and introduce an accurate and robust direction estimation model by exploring the geometrical relationship between the user's movement and its impact on the Fresnel zones. We built our testbed with 4 Raspberry Pis, forming a large correction zone and evaluated SWiDir across 5 participants in 2 different real environments. Our extensive experiments show that SWiDir achieves 8.89 degrees of average 75 percentile error in walking direction estimation, which is 64% lower than the state-of-the-art existing approaches.**

***Index Terms*—Walking Direction Estimation, Dead Reckoning, Smartphone Sensor Calibration, Channel State Information (CSI), WiFi Sensing**

I. Introduction

Knowing the accurate location of the users can enable a wide range of applications, including navigation, social media, and location-based marketing. GPS has long been the gold standard for outdoor localization. However, due to the signal's susceptibility to weakening or obstruction by physical barriers, its applicability in indoor environments is limited. Other than GPS, there are two categories of approaches for indoor localization, device-free and smartphone-based. However, these approaches limit their applicability to pervasive indoor environments due to their inherent limitations [1].

Smartphones have become an essential part of our daily lives, and their widespread use is likely to continue for the foreseeable future. Smartphone based indoor localization technologies include WiFi RSSI [2], dead reckoning [3], BLE-beacon [4], and camera-based [5]. Among these technologies, Pedestrian Dead Reckoning (PDR) is relatively suitable for being used pervasively, as it only relies on the IMU sensors already integrated in smartphones. Although estimating the walking distance by IMU is accurate, the walking direction estimation is inaccurate because of the misalignment between phone's direction and human's movement direction, which eventually leads to inaccurate location estimation [6].

Many recent studies focus on enhancing smartphone-based direction estimation. These approaches either rely on external landmarks [7]–[9], or fuse the IMU sensor readings by applying various statistical techniques [10]–[14]. However, the landmark based approaches require pedestrians to pass specific landmarks consecutively, and the sensor fusion approaches are usually user-specific and are sensitive to how the user walks/carries the phone.

In this paper, we enhance PDR-based direction estimation by gathering additional information from passive WiFi sensing. Passive WiFi sensing is a device-free indoor localization approach that has been studied widely due to the pervasive existence of WiFi. To combine the advantages of smartphone-based motion sensing and WiFi sensing, we deploy WiFi devices to form WiFi sensing regions (correction zones), and use accurate walking distance collected by smartphone to improve the accuracy of pure WiFi sensing. Our approach provides better accuracy and reliability compared with WiFi-based direction estimation, and eliminates the aforementioned constraints posed by smartphone-based direction estimation.

In summary our key contributions are as follows:

- We introduce SWiDir, a direction correction system that corrects smartphone-based user direction estimation by forming Correction zone with passive WiFi sensing. Our approach accurately measures user's walking directions when she passes a correction zone, and utilizes the measured directions to correct the direction of the compass.
- We develop a Fresnel zone-based approach that measures walking directions accurately. The novelty of our approach lies in its unique geometrical model for capturing user's movement by both smartphone and WiFi receivers, making it more accurate and robust.
- We analyze and evaluate our system in 2 real environments and compare its performance with state-of-the-art approaches. We extensively measure the impact of different environments, walking directions and distances on the accuracy of our approach. Our results show that SWiDir consistently outperforms existing approaches that is based either on WiFi sensing or smartphone sensor fusion.

The remainder of this paper is organized as follows. Section II provides a review of existing solutions and their associated problems. In Section III, we present our system overview and design considerations. Section IV outlines the main methodol-

2155-6814/23/$31.00 ©2023 IEEE
DOI 10.1109/MASS58611.2023.00080

ogy behind our system. A detailed description of our system implementation is provided in Section V, while Section VI presents our evaluation results. Finally, we conclude this paper with Section VII.

II. Related Works

There have been numerous efforts to accurately estimate the walking direction and location of a subject by utilizing either smartphone IMU sensor or by manipulating WiFi CSI (Channel State Information) [15], [16]. Both of the approaches suffer from their own limitations. Here, we briefly describe the methodologies of few recent and popular related works, their strengths and weaknesses.

A. Pedestrian Dead Reckoning

In pedestrian dead reckoning (PDR), step count and step length are used to estimate the distance traveled by a pedestrian, while direction estimation is used to determine the orientation of the pedestrian. Knowing the orientation and length of each step will allow pedestrian to predict her real-time locations given the starting location.

The major categories of step counting techniques using the smartphone's built-in accelerometer are threshold setting, peak detection, correlation analysis, and spectral analysis [17]. Experiments show existing step counting approaches already achieve very good accuracy: the results in [17] demonstrate an error rate of lower than 5% for most pedestrians.

Many algorithms for estimating step length require user-specific training [18]. In recent years, few training-free algorithms [19] have also been developed for accurately measuring the step length, independent of the pedestrian. This paper only concentrates on orientation estimation, leaving the step count and step length estimation methods out of our purview because their accuracy is already high and steadily improving.

Compass is the default direction estimation method built-in for Android and iOS. However, due to the misalignment between the direction of the phone and the user's walking direction, using a smartphone compass for dead reckoning only gets moderate precision [6]. Next, we review and summarize the state-of-the-art walking direction estimation approaches.

B. Calibration-based Direction Estimation

Calibration-based direction estimation requires the pedestrian to pass two landmarks, whose locations are known in prior. The pedestrian's trajectory between these two landmarks are recorded, with the phone compass providing inaccurate walking directions. After passing the second landmark, the actual walking direction is calculated, which is further used to infer the projection between the phone's compass reading to pedestrian's walking direction. The landmarks can be either LED light [8], acoustic [9], or Bluetooth beacons [20], [21].

However, the calibration-based direction estimation approaches suffer from the following drawbacks: 1) they usually require to deploy additional hardware devices; 2) their applicability is restricted by the environment (e.g., SoundMark fails in noisy environment and LiDR needs the LED lights to be in a specific shape); 3) they require the pedestrians to pass two consecutive locations, which adds extra hurdle.

C. WiFi-based Direction Estimation

Device-free direction estimation is a technique used to estimate the direction of a moving pedestrian without relying on any devices attached to her. Literatures such as WiDir [22] and WiDar [23] estimate user's direction in a device-free manner by employing one or multiple theoretical models (e.g., Fresnel zone, Doppler Frequency Shift (DFS), and Angle-of-Arrival (AoA)). Among them, the Fresnel zone model stands out for its simple geometrical characteristics and ability to attain precise direction estimations.

Fresnel zone is formed when a radio wave propagates from transmitter (Tx) to receiver (Rx) and takes multiple paths, resulting constructive and destructive interference as the lengths of the reflected paths alternate [22]. Given a radio wavelength λ, Fresnel zones having n number of ellipses are formed by adhering the following properties [22]:

$$|RxP| + |PTx| - |TxRx| = n\lambda/2 \tag{1}$$

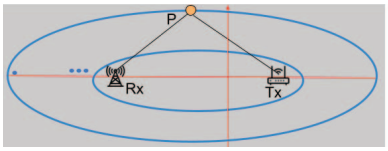


Fig. 1: Geometry of Fresnel Zones.

where P is a point on the nth ellipse as shown in Fig. 1. The path length ($|RxP|$ + $|PTx|$) of the signal reflected or diffracted through the nth Fresnel zone boundary is $n\lambda$/2 longer than that of the Line-of-Sight (LoS) path length ($|TxRx|$). There are infinite number of Fresnel zones, but Raspberry Pis can only measure 256 sets of Fresnel zones as they use 256 OFDM subcarriers. The innermost ellipsoid is called the First Fresnel Zone (FFZ). Traditional FZ approaches suffer from high errors when users are walking within certain regions, which will be detailed in Sec. III-B.

To conclude, the phone compass is not suitable for determining direction in PDR, and each of the two methods for direction estimation has its own drawbacks. Next we present the design overview of SWiDir.

III. SWiDir System

In this section, we first present the overview of SWiDir system, which combines the advantages of passive WiFi sensing and smartphone motion detection for walking direction estimation. Next, we will delve into the various design considerations that were taken into account while designing the SWiDir system.

A. System Overview

To provide walking directions for pedestrians in an indoor environment, SWiDir consists of three major components: 1) **phones** held by pedestrians in an arbitrary manner; 2) **correction zones** which are square in shape and size of around 3*3 meters, with one WiFi router at the center and

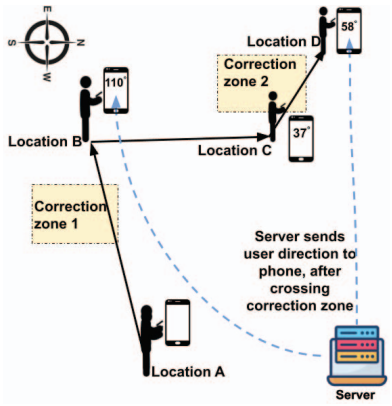


Fig. 2: A Simple Use-case to Illustrate the SWiDir System

four WiFi receivers at the borders; 3) a **server** that collects and processes sensing information from users' phones and receivers via wireless communication links (WiFi or cellular networks), and sends calculated directions back to the phones. Fig. 2 demonstrates our system components, where a user walks from location A to location D via B and C, passing correction zone 1 and 2, with a phone attached to her arm.

When the user is crossing the correction zone, we use the phone's step count results and the passive WiFi CSI information collected by WiFi transceivers to estimate the users walking direction, and corrects the phone's compass direction with the estimated user's movement direction.

To demonstrate how SWiDir system works, consider the simple use-case portrayed in Fig. 2. The user starts walking from point A and stops at point D, along the paths AB, BC and CD. When crossing correction zone 1, SWiDir estimates her walking direction as $110°$ w.r.t. to global north, and sends the direction to her phone. At point B, the user turns $98°$ towards global east, continues walking until she reaches point C. As the compass direction is different from the user's walking direction, the compass direction incorrectly shows $37°$ instead of $12°$($110°$-$98°$=$12°$). As soon as she crosses correction zone 2, phone corrects compass direction as $58°$ at location D.

B. SWiDir Design Considerations

The main challenge of SWiDir is to provide precise walking directions when the user crosses a correction zone. To address this challenge, SWiDir leverages the benefits of motion sensing on smartphones and passive WiFi sensing, thereby enhancing accuracy in navigation.

We have chosen the Fresnel zone-based (FZ) approach for passive WiFi sensing due to its training-free nature. However, traditional FZ approaches calculate direction estimation based on the ratio of the number of Fresnel zones that the user walks across in the x and y axes. This method is an approximation and contributes to errors in the estimation. To address this problem, SWiDir introduces innovative solution. Rather than relying on ratios, SWiDir uses the geometrical relationship between the user's movement and its effect on the Fresnel zones as depicted in Fig 4a. Therefore, this approach enables SWiDir to accurately calculate direction without approximation. In addition to that SWiDir leverages the precise walking distance data collected by smartphones to enhance the accuracy and reliability of WiFi sensing. The details will be given in the next sections.

IV. Methodology

This section initially outlines SWiDir's workflow, followed by a detailed discussion on movement direction calculation.

A. Workflow of SWiDir

Figure 3 demonstrates the workflow of SWiDir. To reiterate, SWiDir employs WiFi-based direction estimation when the user crosses the correction zone. Subsequently, upon exiting the zone, direction correction is conducted on the phone's compass, using the estimated WiFi-based direction as a reference. The steps and details are described as follows.

1) A Network Time Protocol (NTP) is incorporated into the central server to synchronize time between the smartphone and all 4 receivers.
2) A user carrying a smartphone moves into a correction zone. When she crosses the Line-of-Sight between any pair of WiFi transceivers at time T_0, the crossing event is detected by analyzing CSI, which will be described in detail in our system implementation.
3) After analyzing CSI, the server sends T_0 to the smartphone, which keeps collecting the motion sensor's readings while the user is walking. By processing the sensor's readings, it keeps a record of the movement steps of its user, the phone's compass reading, and the timestamp of each step.
4) The phone sends a time T_1 and a distance d to the server, with T_1 being the time the user has walked k more steps after crossing LoS and d being the movement distance between T_0 and T_1. The k value is given as a prior and we calculate d based on each user's individual step length.
5) The server processes the CSI data, counts the fluctuation during time T_0 and T_1 and calculates the user's walking direction θ_p using Eq. 2 and Eq. 3.
6) The server sends back the direction to the phone.
7) The phone then corrects its compass direction received from the server.

B. Movement Direction Calculation in Correction Zone

Fig. 4a shows how the user's direction is calculated from motion sensor readings and WiFi sensing. We use transmitter Tx and four receivers (Rx1 - Rx4) to form a coordination system, with Tx being (0,0) and Rx4 being the positive x axis. When the user crosses the LoS formed by Tx-Rx4 at T_0, we use (ls, 0) to denote the point of LoS crossing. The LoS crossing time can be detected by analyzing the CSI on receiver, but we cannot tell the LoS crossing point (i.e., ls is unknown). The user then continues walking k steps and reaches point P at time T_1. As shown in Fig. 4b, we can express the coordinates of P (user's location at T_1) as:

$$(x = ls + d\cos\theta_p, y = d\sin\theta_p) \tag{2}$$

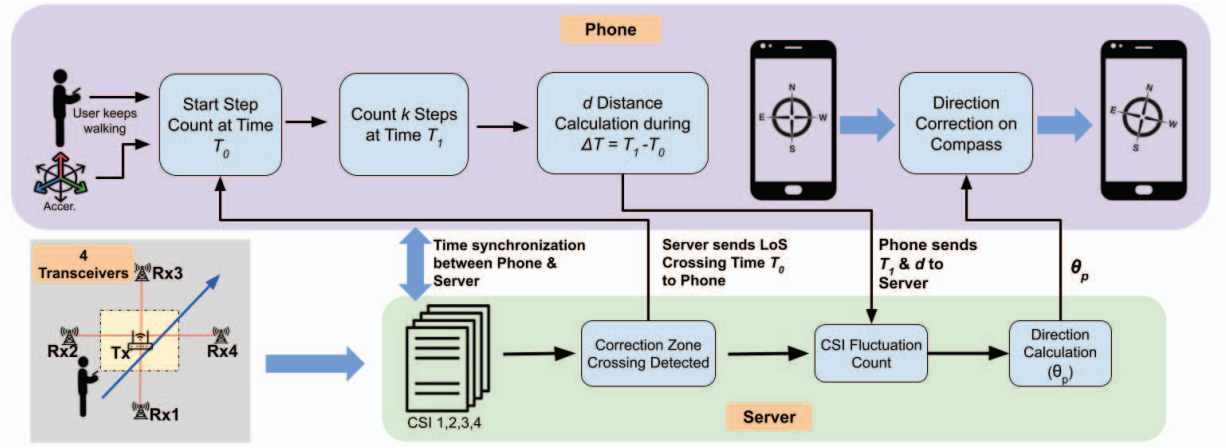
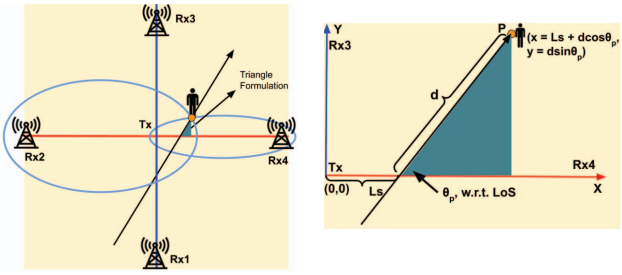


Fig. 3: System Workflow of SWiDir



(a) User's Trajectory (b) User's Location at T_1

Fig. 4: Geometrical Derivation of SWiDir using Triangle

where ls denotes the distance between Tx and crossing point of user on LoS at time T_0, and θ_p denotes the user's walking direction at location P towards the WiFi transceiver system's coordination. In the equation, the distance d between LoS crossing and P is given by the smartphone, while the two unknown parameters are ls and θ_p.

When the user walks between T_0 and T_1, she crosses multiple Fresnel zones formed between two transceiver pairs, Tx and Rx2, and Tx and Rx4. Such Fresnel zone crossing events can be detected by analyzing the CSI captured by corresponding receivers, as they cause fluctuations on the receiving amplitude levels. For simplicity, we approximate the sequence of the Fresnel zone at location P by using the number of Fresnel zone the user crossed [24]. Here we use m and n to denote the sequence of the Fresnel zone of Rx2 and Rx4 at location P, which can be calculated by fluctuation counts. According to the Fresnel zone model, we have:

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

where a_1^n, b_1^n, a_2^m, b_2^m are the major and minor axes of the nth and mth ellipses respectively (see Fig. 5) and can be calculated from the Fresnel zone theory given m and n; c is half of the distance between the transmitter and receiver, which is given during system setup. Finally, replacing the x, y values obtained from Eq. 2 into Eq. 3, we can calculate the user direction θ_p and the LoS crossing point $(ls, 0)$.

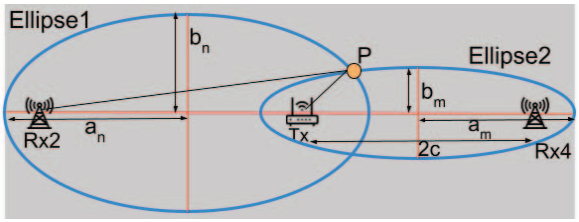


Fig. 5: The mth and nth Fresnel Zone for Rx2 and Rx4

V. System Implementation

In this section, we provide a description of hardware setup and steps involved in the implementation of SWiDir software.

A. Hardware Setup

Each correction zone involves 4 receivers and 1 transmitter. We use Raspberry Pi 4B as receiver to collect CSI data. The reason for selecting the Raspberry Pi 4B was due to its wide availability and low cost. However, the default firmware of Broadcomm WiFi chip does not allow to capture CSI data. Therefore, a modified firmware [16] developed by Nexmon was used to capture CSI data. The firmware passes the CSI data from the link layer to the host system by creating frames with CSI embedded as transport layer payloads. A TP-Link Archer A7 router was used as a transmitter working in 5GHz band. We ping the router every 5ms to trigger it's pong packet, which will be received by the four receivers.

A desktop with Intel core i7 and 8GB RAM is used as server, which is connected to all receivers by Ethernet. The server is set up to process the CSI data and all the calculations for direction computation. Additionally, it runs a Network Time Protocol (NTP) server, which handles the time synchronization among all system components.

Lastly, we use a Huawei Nova 3i smartphone as the user's device. It communicates with the server through TCP socket and synchronizes its time automatically from the NTP server.

B. Software Implementation

We implement the following steps on the server along with the android App used in the phone:

1) LoS Crossing Detection: LoS crossing detection step involves searching for a pattern on streams of CSI reading from 4 receivers and returning the time T_0 as LoS crossing time as well as the receiver number (i.e. Rx1, 2, 3 or 4) through which the user crosses the LoS. We observe a specific pattern on the CSI reading when the user is crossing the LoS. Fig. 6 shows the pattern (marked by red rectangle box) when user crosses LoS between the time $t_0 = 3.7sec$ and $t_0' = 4.4sec$. It is noticeable from the pattern that the energy amplitude during the time window t_0 and t_1 falls below -55dBm. The reason is when user is crossing the LoS she blocks the radio signal between the transmitter and receiver, resulting in low energy observed in CSI reading. Since the time interval while crossing LoS is small (less than 1 sec), we take $T_0 = (t_0 + t_0')/2$, which is the average of the interval.

We write a python script to implement a simple search technique to find the pattern. The script searches for consecutive amplitude values equal or less than the threshold in time domain. The window size of the pattern depends on the body shape of the user, her walking speed, and the hardware specifications. We repeat the experiments and observe that for our hardware and correction zone setup, a normal sized human walking at the speed between 0.5 to 1.5 m/s can be correctly detected by applying a threshold amplitude value of -55dBm, which is the parameter we use in our evaluation.

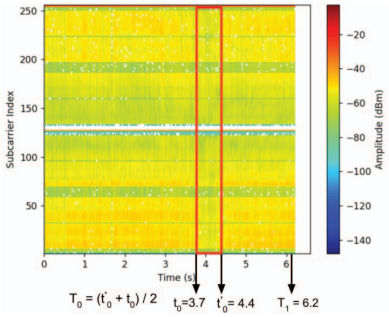


Fig. 6: LoS Crossing Detection

2) CSI Fluctuation Count: This step gets m and n by counting fluctuations in the CSI data captured by $Rx2$ and $Rx4$ between T_0 and T_1, where T_0 is given by the LoS crossing detection and T_1 is given by the phone. While walking along a radio propagation path, human causes peaks and valleys in the measured CSI data in the Fresnel zone. Counting the number of peaks and valleys tells which Fresnel zone the user is in after walking k steps.

CSI data collected by Raspberry Pi is noisy. To remove the noise, we apply the Least-square smoothing filter [25] to smooth the CSI without warping the waveform too much. The filter determines a polynomial fit for a certain number of input samples, i.e., a sample window. From extensive experiments conducted on different scenarios(environments, walking directions and distances), we empirically choose 51 as the window size. Figure 7 demonstrates the results before and after smoothing the CSI. We then use *find_peaks* function from *SciPy* package to count the fluctuation. We experimentally choose the minimum height as 750 (raw amplitude value, not in dBm scale) to be counted as a fluctuation.

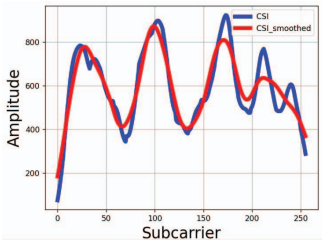


Fig. 7: Before and After CSI Smoothing

3) Direction Calculation: Knowing m and n, the next step is to solve the 2 ellipses equations derived in Eq. 3. Given n, the boundary of n-th Fresnel zone b_1^n formed by Rx2 is defined by Fresnel zone model as follows:

$$b_1^n = \{|Rx_2P| + |PTx| - |TxRx_2| = n\lambda/2\} \tag{4}$$

where λ is a parameter determined by the radio wavelength. Similarly, we can obtain the boundary for m-th Fresnel zone b_2^m formed by Rx4. Additionally, for the $ellipse1$ and $ellipse2$, we can get 3 more properties of ellipses according to Fresnel zone theory. They are required to calculate the values of a_1^n, b_1^n, a_2^m and b_2^m for Eq. 3. For simplicity, 3 properties are given below for $ellipse1$:

$$\begin{aligned} |Rx_2P| + |PTx| &= 2a_1^n \\ |TxRx_2| &= 2c \\ (a_n^1)^2 - c^2 &= (b_n^1)^2 \end{aligned} \tag{5}$$

Solving Equations 4 and 5, b_1^n is obtained as follows:

$$b_1^n = \frac{\sqrt{n\lambda^2 + 8n\lambda c}}{4} \tag{6}$$

Putting b_1^n in Eq. 5, a_1^n is derived, where c is a known variable (from Eq. 3). Similarly, we can get the values of a_2^m and b_2^m for $ellipse2$. Therefore, we can solve Eq. 3 to obtain 2 unknown variables θ_p (direction) and ls as depicted by Fig. 4b. Once direction θ_p and ls distance are derived, user location x and y at point P can easily be obtained using Eq. 2.

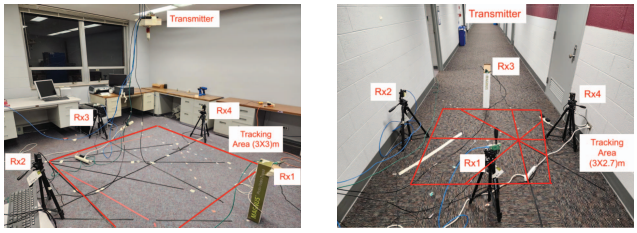
4) SWiDir App Implementation: We base our App implementation on an open-source compass app [26]. In particular, our App (1) communicates with the server to receive T_0 and θ_p, and sends T_1 and d; (2) detects user's steps and records time; (3) corrects compass direction received from the server.

VI. Evaluation

This section describes the procedure and results of our evaluation.

A. Testbed Setup

We conducted experiments in two indoor environments to properly evaluate the performance of SWiDir, which are: a lab office of 7.5m×6m and a large empty corridor of 46m×3m. In both environments, we placed the four receivers forming a 3m×3m correction zone. The lab office has four tables, four chairs, and a few desktops and monitors. The lab office represents an environment with rich multi-path reflections, and the corridor represents an environment with static reflections.



(a) Office Room (b) Corridor

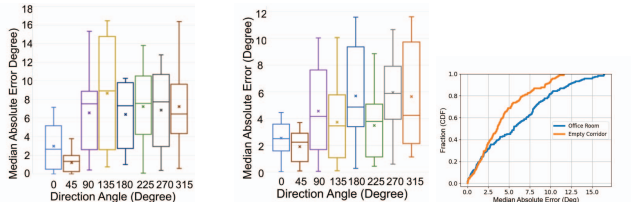
Fig. 8: Testbed Setup

The receivers were placed 50cm above the ground and mounted on tripods as shown in Fig. 8. The distance between the WiFi router and each Raspberry Pi was 1.5m. The router was attached to the ceiling, at the perpendicular bisector of the 4 receivers. Maintaining a certain distance from the ground is important to reduce radio waves' reflection from the ground.

B. Performance Evaluation

Firstly, we assessed our WiFi sensing approach by analyzing the direction estimation accuracy for 8 basic paths. These basic paths crossed the WiFi router from 8 directions separated 45 degrees apart, i.e., 0, 45, 90, 135, 180, 225, 270 and 315 degrees towards the line from Rx2 to Rx4. As the smartphone-based distance estimation is known to be accurate and is not the focus of our research, we assume that the distance d measured by smartphone is accurate, and our experiment only measured the errors caused by our WiFi sensing approach. To establish the actual directions for each path, we used a digital protractor and markers to demarcate the path on the ground. We enlisted three volunteers to walk from the designated starting point to the end point while maintaining their torso alignment with the straight line. We repeated each data collection eight times and captured 384 sets of WiFi CSI data (2 environments×3 volunteers×8 paths×8 repeats).

Figure 9 depicts the Median Absolute Error(MAE) for the eight basic paths in the office room and empty corridor respectively. Figure 9.(a) shows that the overall MAE for the office room is 6 degrees and deviation is 5.41 degrees. Figure 9.(b) shows that the overall MAE for empty corridor is around 5 degrees and deviation is 4.04 degrees. Figure 9.(c) shows that compared with the office room with rich multi-path reflections, empty corridor produced less error: 75th percentile error in corridor is around 6 degrees while in office room the error is around 8 degrees. Among all the angles, only 45 degree has less than 4 degrees of error for upper quantile in both of the rooms.



(a) Office Room (b) Empty Corridor (c) CDF Comparison

Fig. 9: Accuracy of SWiDir in 2 Different Environments

We then assessed how different LoS crossing locations impact accuracy. As shown by Fig. 10, the testers passed the LoS between Rx2 and Rx4 with a fixed walking direction but different LoS crossing locations. These paths (P_1, P_2, P_3, P_4 and P_5) each has the same length of 3m, and are 0.5m apart. Fig. 11a shows that direction estimation results for all the paths has less than 7 degrees of median error, indicating that our WiFi sensing approach can achieve good results with different LoS crossing points.

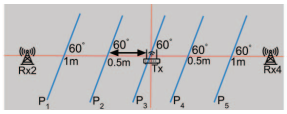
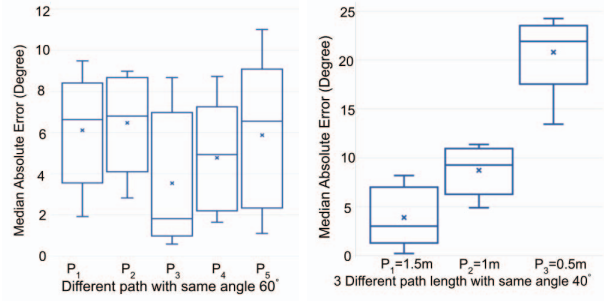


Fig. 10: Paths with Different LoS Crossing Locations



(a) Different LoS Crossing Pt. (b) Different Distance d

Fig. 11: Impact of LoS Crossing Locations and Distances

We also observed that the human body's impact on the accuracy cannot be ignored. Although we model human body as a point in our equations, the width of human body actually causes an impact on the CSI reflection and the Fresnel zone fluctuation counts. Such impact is less significant if the testers walk a longer distance after crossing LoS. Hence, we conducted another experiment to measure how distance d (the length of the path after LoS crossing) impacts accuracy. The testers passed the LoS with a fixed walking direction but different walking distances d (i.e., 0.5m, 1m, 1.5m). Figure. 11b exhibits that lower d distance incurs more error. Based on this observation, when deploying the correction zones, the developers need to choose the locations where the users can continue walking for at least 0.5m after LoS crossing.

Lastly, we calculated the median and 75th percentile accuracy achieved by SWiDir in all testing scenarios, and compared it with three additional state-of-the-art works namely

WiDir [22], Widar [23] and WalkCompass [27]. Of the three works, WiDir and Widar rely on WiFi passive sensing, while WalkCompass is smartphone-based. The accuracy of these approaches was obtained from publications since we lacked access to their source codes. Table- I shows that SWiDir outperforms all the three works for 75th percentile and median error except Widar achieved median error of 5 degrees, where SWiDir attain 6 degrees. Compared with the closest (WalkCompass), SWiDir reduces the 75th percentile error by 64%. The reason for achieving outperforming result is the geometrical relationship which does not cause cumulative error unlike other models.

Approaches	75th Percentile Error (in degree)	Median Error (in degree)
WiDir	23	10
WalkCompass	14.2	8
WiDar	18	5
SWiDir (Our Approach)	**8.89**	**6**

TABLE I: Comparing SWiDir's Accuracy with Other Methods

C. Discussion

As far as we know, SWiDir is the first attempt to combine passive WiFi sensing with phone-based direction estimation. As indicated by our evaluation, SWiDir achieved good accuracy comparing with other state-of-the-art models. It is also training-free, so it can be deployed in a plug-and-play fashion. However, likewise the other RF-based human sensing applications such as localization and gesture recognition, SWiDir is also impacted by rich multi-path and its accuracy drops greatly when two persons enter the same correction zone simultaneously. Besides, pedestrian should also walk for 0.5 meters after crossing LoS. Hence, developers need to pay attention to satisfy these requirements when using SWiDir.

VII. Conclusion

In this paper, we present SWiDir, a system for accurately estimating movement direction by combining passive WiFi sensing with phone based direction estimation. The novelty of our approach lies in introducing additional receivers for better WiFi sensing and utilizing a geometrical approach for accurate direction estimation. We implement SWiDir using Raspberry Pi and our extensive experiments show that SWiDir achieves 8.89 degrees of average 75 percentile error in walking direction estimation, which is 64% lower than the state-of-the-art approaches.

Acknowledgment

This research is supported by NSF through grant 2104337.

References

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[2] W. Xue, W. Qiu, X. Hua, and K. Yu, "Improved wi-fi rssi measurement for indoor localization," *IEEE Sensors Journal*, vol. 17, no. 7, 2017.

[3] K. Sung and H. Kim, "Bayesian navigation system with particle filtering and dead reckoning in urban canyon environments," in *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 2012, pp. 73–75.

[4] P. Spachos and K. N. Plataniotis, "Ble beacons for indoor positioning at an interactive iot-based smart museum," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3483–3493, 2020.

[5] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg, "Indoor positioning and navigation with camera phones," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 22–31, 2009.

[6] N. Mohssen, R. Momtaz, H. Aly, and M. Youssef, "It's the human that matters: Accurate user orientation estimation for mobile computing applications," *arXiv preprint arXiv:1411.2156*, 2014.

[7] N. Yu, X. Zhan, S. Zhao, Y. Wu, and R. Feng, "A precise dead reckoning algorithm based on bluetooth and multiple sensors," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 336–351, 2017.

[8] B. Hussain, Y. Wang, R. Chen, H. C. Cheng, and C. P. Yue, "Lidr: Visible-light-communication-assisted dead reckoning for accurate indoor localization," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 742–15 755, 2022.

[9] H. Chen, F. Li, and Y. Wang, "Soundmark: Accurate indoor localization via peer-assisted dead reckoning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4803–4815, 2018.

[10] D. Yan, C. Shi, T. Li, and Y. Li, "Flexpdr: Fully flexible pedestrian dead reckoning using online multimode recognition and time-series decomposition," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 15 240–15 254, 2022.

[11] J. Perul and V. Renaudin, "Learning individual models to estimate the walking direction of mobile phone users," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12 306–12 315, 2019.

[12] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Robust indoor positioning with lifelong learning," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2287–2301, 2015.

[13] J.-S. Lee and S.-M. Huang, "An experimental heuristic approach to multi-pose pedestrian dead reckoning without using magnetometers for indoor localization," *IEEE Sensors Journal*, vol. 19, no. 20, 2019.

[14] Q. Wang, H. Luo, H. Xiong, A. Men, F. Zhao, M. Xia, and C. Ou, "Pedestrian dead reckoning based on walking pattern recognition and online magnetic fingerprint trajectory calibration," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 2011–2026, 2020.

[15] Y. Lin, W. Dong, B. Li, and Y. Gao, "Tinycsi: A rapid development framework for csi-based sensing applications," in *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020, pp. 549–557.

[16] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free your csi: A channel state information extraction platform for modern wi-fi chipsets," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 2019, pp. 21–28.

[17] F. Gu, K. Khoshelham, J. Shang, F. Yu, and Z. Wei, "Robust and accurate smartphone-based step counting for indoor localization," *IEEE Sensors Journal*, vol. 17, no. 11, pp. 3453–3460, 2017.

[18] S. Vandermeeren, H. Bruneel, and H. Steendam, "Feature selection for machine learning based step length estimation algorithms," *Sensors*, vol. 20, no. 3, p. 778, 2020.

[19] F. Bo, J. Li, and W. Wang, "Mode-independent stride length estimation with imus in smartphones," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5824–5833, 2022.

[20] Y. Hu, F. Qian, Z. Yin, Z. Li, Z. Ji, Y. Han, Q. Xu, and W. Jiang, "Experience: Practical indoor localization for malls," in *MobiCom'22*, 2022, pp. 82–93.

[21] Z. Chen, Q. Zhu, and Y. C. Soh, "Smartphone inertial sensor-based indoor localization and tracking with ibeacon corrections," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 4, pp. 1540–1549, 2016.

[22] D. Wu, D. Zhang, C. Xu, Y. Wang, and H. Wang, "Widir: walking direction estimation using wireless signals," in *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, 2016, pp. 351–362.

[23] K. Qian, C. Wu, Z. Yang, Y. Liu, and K. Jamieson, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity wi-fi," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2017, pp. 1–10.

[24] M. A. R. Ahad, U. Mahbub, and T. Rahman, *Contactless human activity analysis*. Springer, 2021, vol. 200.

[25] R. W. Schafer, "What is a savitzky-golay filter?[lecture notes]," *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[26] iutinvg, "Simple android compass implementation," 2018. [Online]. Available: https://github.com/iutinvg/compass

[27] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 329–342.