Journal of Intelligent Computing and Networking
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
E-mail: jicn.office@ffspub.com

Article

Visual Assistant: A Vision-Assisted System for Generating Wi-Fi Device Maps

Jinfan LIU, Renrui TAN, Jiajie CAO, Xinyu TONG†
Colleague of Intelligence and Computing, Tianjin University, Tianjin 300072, China
†*E-mail: xytong@tju.edu.cn*

Received: June 3, 2025 / Revised: June 28, 2025 / Accepted: July 15, 2025 / Published online: July 28, 2025

Abstract: Device-free sensing systems have the potential to enhance human-computer interaction applications, but their effectiveness is highly dependent on the precise placement of devices, which typically requires a significant amount of labor input. To alleviate this labor-intensive process, we present the *Visual Assistant*, an inventive system engineered to autonomously generate environmental maps and accurately identify the locations of Wi-Fi infrastructure for sensing. Our proposed approach employs commercially available devices in smart homes including the Wi-Fi router, the camera, and the robotic vacuum cleaner. The *Visual Assistant* consists of the following two stages: i) the camera mapping stage synthesizes the coordinate systems with respect to the camera and the Simultaneous Localization and Mapping (SLAM) robots; ii) the Wi-Fi mapping stage proposes the hyperbolic model to inversely pinpoint the positions of undisclosed Wi-Fi infrastructure based on human movement and the camera. The innovation of this paper lies in the proposed hyperbolic model, which theoretically reveals how to infer Wi-Fi location based on known user trajectories. We conduct comprehensive experiments to verify the performance of the *Visual Assistant*. The results suggest our system can yield centimeter-level and decimeter-level mapping accuracy in camera mapping and Wi-Fi mapping stages, respectively, marking a substantial stride in the development of device-free sensing systems.

Keywords: Mapping; Wi-Fi; vision
https://doi.org/10.64509/jicn.11.23

1 Introduction

1.1 Background and Motivation

Contact-free sensing enables functionalities such as localization, fall detection, and respiration detection, playing a critical role in smart homes and elderly care applications. Among contact-free sensing systems utilizing millimeter waves, cameras, and other technologies, Wi-Fi stands out as the most promising due to its ubiquity. However, smart home applications urgently demand the ***plug and play*** characteristics. For instance, Wi-Fi-based sensing systems can be deployed without additional manual calibration. The advantages are twofold: i) From the consumer's perspective, they can benefit from smart applications without getting into the technicalities; ii) From the manufacturer's perspective, they do not need to train specialized personnel to maintain the sensing system for consumers.

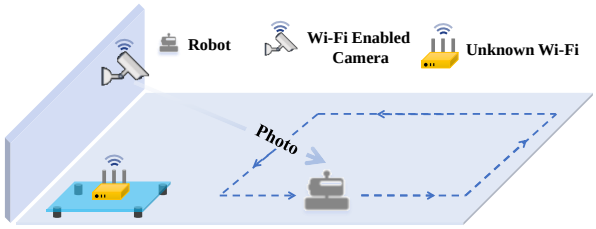
To solve the aforementioned issues, existing studies [1–5] have proposed methods to automatically construct Wi-Fi maps. However, these works suffer from at least one of the following limitations: i) They necessitate an involved process of phase calibration. Specifically, we often need expensive professional equipment, such as power dividers, to connect the antennas of the transceivers [5]. Unfortunately, not all Wi-Fi devices have exposed antennas, which would require us to disassemble the smart devices; ii) They require a non-linear antenna layout to achieve phase cleaning, but such antenna layouts are not always available. Moreover, many smart devices are not even equipped with three or more antennas to form a non-linear antenna layout.

1.2 Basic Idea and Challenges

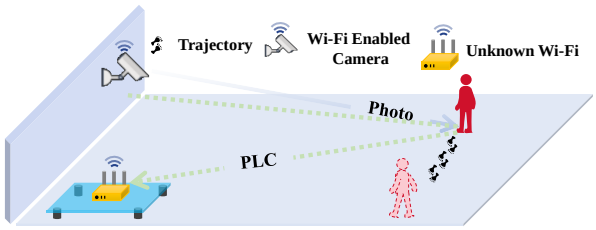
In this paper, we introduce a novel calibration-free, automated Wi-Fi mapping system christened ***Visual Assistant***. This system is engineered not only to establish the relative configuration of Wi-Fi networks but also to accurately pinpoint the locations of these networks on authentic architectural floor plans. As shown in Figure 1, the core methodology

† Corresponding author: Xinyu Tong
*Academic Editor: Chunxiao Jiang

of our *Visual Assistant* system employs everyday household devices, specifically Wi-Fi-equipped digital cameras and robotic vacuum cleaners. These devices collaborate to form a complete prototype system. The robotic vacuum cleaner plays a vital role in creating the indoor floor plan, while the Wi-Fi-enabled digital camera marks the positions of all Wi-Fi infrastructure within this mapped area. Despite the promising technological advancements embodied by the *Visual Assistant* system, its design and implementation still face two significant challenges that require further exploration and mitigation:



(a) Camera mapping



(b) Wi-Fi mapping

Figure 1: (a) a general scenario of camera mappings; (b) the camera and the user are utilized to acquire the location of unknown Wi-Fi devices.

Challenge 1: How to establish accurate correspondence between home scene and floor plan? One primary obstacle we encounter is the difficulty of utilizing a camera to accurately correlate the physical home scene with the constructed floor plan. Although the robotic vacuum cleaner is capable of developing an indoor map, the direct establishment of a mapping relationship between the scene and the floor plan remains elusive. To address this challenge, we propose a solution informed by insights gleaned from Perspective-n-Point (PnP) [6], a fundamental problem presented early in the realm of Computer Vision. Many well-developed algorithms exist for solving PnP [7–9]. In essence, our approach employs the robotic vacuum cleaner and the camera to construct a transformation matrix H that bridges the gap between the ground plane and the camera photo plane.

Challenge 2: How to map Wi-Fi device locations onto the floor plan? A seemingly direct approach to this challenge would be utilizing a camera to identify Wi-Fi devices and subsequently mark their positions on the floor plan. However, this approach proves to be inadequate when a Wi-Fi device falls outside of the camera's field of view, particularly due to Non-Line-of-Sight (NLOS) conditions. Our strategy to address this issue involves leveraging the movements of a user within the monitored space as an intermediary, thus enabling us to infer the location of Wi-Fi devices beyond the field of view. Notably, our theoretical analysis has led us to the development of a hyperbolic model. The crux of this model is the understanding that, given any two points on a user's walking trajectory, the location of the unknown Wi-Fi device should lie along a hyperbola, with these two points serving as its foci.

1.3 Contributions

In this paper, we make the following key contributions:

1. We propose a hyperbolic model, which theoretically reveals how to infer Wi-Fi location based on known user trajectories. In particular, given the location of the transmitter and the user's trajectory, the receiver's position aligns with the hyperbolas whose foci are any two randomly selected points along the trajectory. Based on our theoretical discovery, we develop an innovative method to infer the location of unknown receivers.
2. We create a working prototype of the *Visual Assistant*. As depicted in Figure 1, this system harmoniously integrates SLAM, image analysis, and Wi-Fi multi-modal information. The experimental results of our system provide compelling evidence of its efficacy, with up to 80% of all test scenarios achieving a mapping accuracy within 0.5 meters.

Based on the above technical contributions, we outperform existing systems in the following aspects:

1. We experiment with automatic mapping using commonly available home devices such as a camera, a robot vacuum cleaner, and commercial Wi-Fi devices. Our system does not require additional time-consuming phase calibration, thus promoting the deployment of Wi-Fi sensing in ordinary home environments.
2. We harness the full potential of multimodal data, addressing the challenge of aggregating information in distributed systems and achieving complementarity between various technologies.

The remainder of this paper is organized as follows. We first discuss related works in §Section 2. Then we present an overview of our proposed system in §Section 3 while the design details are illustrated in §Section 4. We implement and evaluate the proposed system in §Section 5. Finally, we conclude this paper in §Section 6.

2 Related Work

2.1 Wi-Fi Localization Technique

Due to the ubiquitous deployment of Wi-Fi devices, Wi-Fi based localization has attracted wide attention [3, 10–16]. These works can be classified into two categories, device-based and device-free methods.

2.1.1 Device-based Localization System

These systems leverage the integrated sensors within these devices, including accelerometers [17], gyroscopes [18], magnetometers [19], and barometers [20], to gather data pertaining to the device's motion and spatial orientation. By utilizing

the inherent resources of the device, device-based methodologies eliminate the need for additional infrastructure, making real-time tracking more feasible. However, the effectiveness of these methods may be influenced by various factors, including the accuracy of sensors. Moreover, the requirement to carry devices for indoor localization purposes may cause inconvenience for certain individuals.

2.1.2 Device-free Localization System

Device-free methodologies eliminate the need for target devices to be equipped with any additional hardware or sensors. Instead, they strategically leverage the existing infrastructure within the environment, such as Wi-Fi access points [13] or Bluetooth beacons [21], to perform indoor localization and tracking tasks. By analyzing signal measurements or noticeable changes in the environmental conditions, these device-free methodologies [22–25] can deduce the approximate location of a target device without requiring any active engagement from the device itself. However, it is noteworthy that these device-free sensing systems lack the capacity to discern user identities. Additionally, these methodologies often encounter difficulties due to environmental variations and may face challenges when attempting to simultaneously localize multiple targets.

2.2 Hyperbolic Models in Wi-Fi Sensing

Hyperbola is an important mathematical concept within the Cartesian coordinate system, which describes a set of points whose distance from two fixed points, noted as the foci, differ a constant value. It can be written as the following equation:

$$abs(|PF_1| - |PF_2|) = c, \tag{1}$$

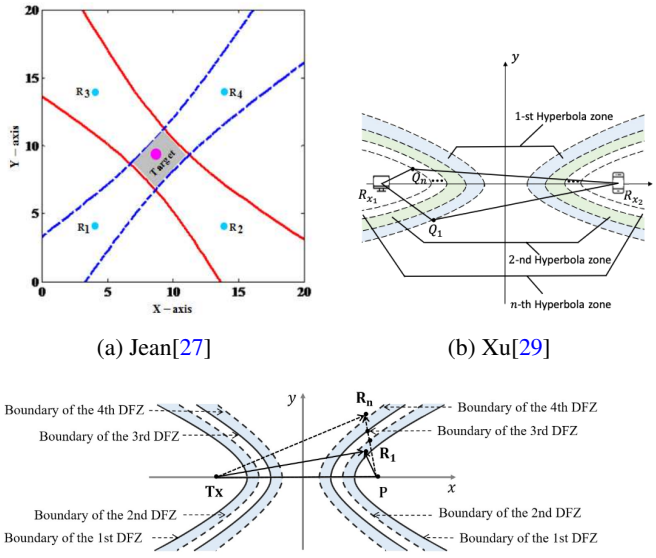
where P is the point on the plane, F_1 and F_2 are two foci and c is a constant number.

Shown in Table 1, in the context of Wi-Fi Sensing, many prior works have discoveries based on the concept of Hyperbola [26–29]. Anvar [26] proposed a hyperbolic model to address the challenge of accurate indoor localisation. It proposed a hyperbolic model based on Received Signal Strength Indication (RSSI) fingerprinting, involving the estimation of Time Difference of Arrival (TDOA) between signals from different transmitters. Specifically, the distance between ith source (BS) and the mobile receiver is given as: $R_i = \sqrt{(X_i - x)^2} = \sqrt{X_i^2 + Y_i^2 - 2X_i x - 2Y_i y + x^2 + y^2}$, where (X_i, Y_i) and (x, y) are coordinates of ith BS and mobile station respectively. The range difference between base stations with respect to the first arriving BS is: $R_{i,1} = v \cdot t_{i,1} = R_i - R_1$. This measurement is transformed into range differences between base stations, creating a set of nonlinear hyperbolic equations. Jean [27] discovered a hyperbolic model to localize a transmitter using Received Signal Strength (RSS). The mathematical model is based on the ratio of the transmitter-receiver distance between two receivers, with the assumption that the power of the received signal decreases by the loss value over the distance, proportional to the distance to the transmitter. Shown in Figure 2a, the estimated location is bounced by two pairs of hyperbolas. Liu [28] proposed a hyperbolic model to localise a *moving* receiver using Channel State Information (CSI). The model is based on the multipath effects, using a stationary transmitter and a known signal reflector. It proposed Zones of Hyperbolas, known as the Dynamic Frenel Zones. As is shown in Figure 2c, all hyperbolas share the same foci, which are the Transmitter and the signal reflector, and each hyperbola represents a certain CSI pattern, indicating a candidate area for the unknown moving receiver. Xu [29] proposed a novel theoretical model, called the Hyperbolic zone, to reveal the fundamental sensing mechanism in Non-Light-of-Sight (NLoS) scenarios. It proposed the Differential Path Length Change Rate (DPLCR) to resolve the problem of PLCR discrepancies in both LoS and NLoS scenarios and consequently derive a hyperbolic zone as:

$$abs(|Q_n R_{x1}| - |Q_n R_{x2} - |R_{x1} R_{x2}|) = n\frac{\lambda}{2}, \tag{2}$$

where R_{x1} and R_{x2} are two receivers, λ is the wavelength, Q_n is a point on the n^{th} hyperbola, $abs(\cdot)$ means to solve the absolute value. The equation can be illustrated in Figure 2b, where the target Q_i can be tracked through the Hyperbola zone.

All these models are proposed using only the source of Wi-Fi signals, either the RSSI, TDOA or the more modern choice of CSI. The Wi-Fi model alone provides rich information though, our model utilise the LiDAR sensor, which provides even richer information, providing even better performance.



(a) Jean[27] (b) Xu[29]

(c) Liu[28]

Figure 2: Related Work: Hyperbolic Models

2.3 Wi-Fi Mapping

Despite promising applications, the effectiveness of Wi-Fi sensing is highly dependent on the precise placement of devices, which typically requires a significant amount of labor input. To address this problem, existing studies [1–5, 30] have proposed methods to automatically construct Wi-Fi maps. However, as illustrated in Table 2, they typically use old models like AoA, failed to enable the rich information of modern CSI. TriLoc [1] proposes a new antenna layout for CSI based Wi-Fi localization systems, which can significantly

Table 1: Comparison of Related Hyperbolic Models

Name	Model Name	Source	Application
Anvar [26]	Pairs of Hyperbolas	RSS	Localize a moving transmitter by multiple receivers.
Jean [27]	Pairs of Hyperbolas	RSSI	Localize multiple mobile users by multiple receivers.
Liu [28]	Dynamic Fresnel Zone	CSI	Localize a moving receiver by a transmitter and a reflector.
Xu [29]	Hyperbolic Zone	DPLCR (CSI)	Tracking in NLoS scenarios by transceivers.
Ours	Hyperbolic Cluster	CSI & LiDAR	Localize multiple receivers by an external moving object.

Table 2: A comparison of related Wi-Fi mapping methods.

System	Model	Modalities	Accuracy
TriLoc 2019 [1]	ADoA	Wi-Fi	0.60m @80%
LocAP 2020 [2]	AoA	Wi-Fi & SLAM	0.90m @80%
MapFi 2023 [3]	AoA+	Wi-Fi	0.74m @80%
VoiceMap 2023 [4]	AoA	Wi-Fi & Acoustic	0.40m @80%
Ours	Hyperbola	Wi-Fi & LiDAR	0.40m @80%

reduce the calibration efforts required for large-scale deployment. The proposed triangular antenna layout can achieve 80% AoA measurement error within 9 degrees for any direction, resulting in promising localization accuracy without a labor-intensive site survey. LocAP[2] is an autonomous and accurate system to estimate access point location attributes, antenna placements, and deployment orientation. It establishes the requirements for reverse localization to ensure accurate access point locations. LocAP utilizes a calibration process to estimate the relative geometry of the access points and the user's location accurately. The calibration process involves measuring the signal strength of Wi-Fi signals at different locations in the environment and using this information to estimate the location attributes of the access points. However, LocAP requires a time-consuming phase calibration process, which is impractical in real deployment. MapFi[3] can autonomously map WiFi infrastructure for indoor localization without the need for labor-intensive site surveys. The system proposes a general method to estimate AoA and generate the WiFi map. VoiceMap[4] presents an autonomous mapping system for voice localization using a microphone array. The system utilizes a sweeping robot to explore the environment and a microphone array to localize the robot, establishing the positional relationship between the two. Although this methods present better accuracy comparing with others, the nature of sounds still prevents it from vast deployment since souds require strict environmental setup.

These works suffer from at least one of the following limitations: i) They necessitate an involved process of phase calibration. Specifically, we often need expensive professional equipment, such as power dividers, to connect the antennas of the transceivers [5]. Unfortunately, not all Wi-Fi devices have exposed antennas, which would require us to disassemble the smart devices; ii) They require a non-linear antenna layout to achieve phase cleaning, but such antenna layouts are not always available. Moreover, many smart devices are not even equipped with three or more antennas to form a non-linear antenna layout; iii) They are not practical in ordinary smart home environment. The process of Wi-Fi mapping proposed in these work requires direct communication between each pair of devices, which is inapplicable in a typical smart home environment where devices communicates via routers. Our work, in contrast, only requires direct communication between a central transmitter and other devices, which is realistic in the deployment of smart home.

3 System Overview

The workflow of our proposed system is illustrated in Figure 3, which consists of the following two stages:

Camera Mapping Stage. The objective of this stage is to align the position of the camera with the map generated by SLAM [31–33] on the robot. Our approach involves using the camera to capture the relative position in relation to the robot, followed by designing a transformation matrix H to translate the 2D locations in the camera's images to the corresponding locations on the SLAM-generated ground plane for the subsequent stage. In particular, the robot employs SLAM to create a comprehensive map of the environment, reporting the robot's locations on the ground plane. Meanwhile, the camera captures images and leverages a YOLOv5-based [34] neural network to identify the robot and extract its 2D locations. Once an adequate number of location pairs are gathered

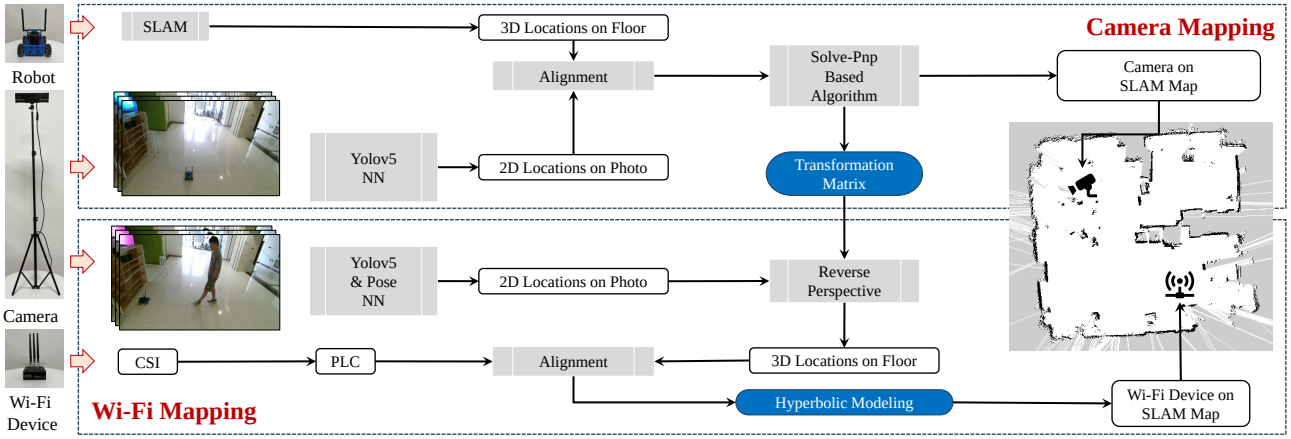


Figure 3: Our proposed system consists of two stages of camera mapping and Wi-Fi mapping.

and time-aligned, a specialized Solve-Pnp-based program is executed using OpenCV [35] to calculate both the camera's location and the transformation matrix.

Wi-Fi Mapping Stage. The aim of this stage is to ascertain the location of the Wi-Fi device on the SLAM map. Our insight is that cameras and Wi-Fi devices can perceive a moving human from distinct perspectives, and by analyzing the intrinsic relationship between different observed features, we can determine the relative positions between the camera and the Wi-Fi device. In particular, the camera can capture the user's location directly, while the Wi-Fi devices can detect changes in the signal caused by the reflections on the human body. In this paper, we conduct a theoretical analysis to demonstrate that the receiver is situated on the hyperbola with any two user locations serving as the foci. By leveraging our novel theory, we are ultimately able to localize the Wi-Fi devices on the SLAM map.

4 System Designs

This section elucidates the architecture of our system. The paramount goal is the autonomous alignment of both the camera and Wi-Fi device with the reference coordinates, extracted from a physical map generated via SLAM. To effectively achieve this, our explanation is structured into two sequential yet interconnected phases: the Camera Mapping and the subsequent Wi-Fi Mapping stages.

4.1 Camera Mapping Stage

The primary focus of the first stage is to ascertain the precise geographical location of the camera and subsequently compute a transformation matrix. This matrix plays a pivotal role in converting a given 2D location from an image, captured by the camera, into its corresponding position on the ground plane.

4.1.1 Camera Localization

The process of acquiring the precise location of the camera utilizes the Solve-PnP algorithm, a method that has been researched and validated by studies such as those by [7, 8] and numerous others. The mechanism of this algorithm is centered around creating a correspondence between object points on the ground plane and their corresponding 2D perspective projection coordinates within an image. Our world coordinate system is facilitated by a SLAM-based autonomous robot, which transmits its location information at each time slot in the form of $[X_i, Y_i, Z_i]^T$. To harmonize this with the 2D location, a neural network, underpinned by the YOLOv5 architecture, is trained to recognize the robot and subsequently extract its 2D coordinates, denoted as $[x_i, y_i]^T$. This process is visually represented in Figure 4. In the concluding steps, we synergistically combine the intrinsic parameters of the camera and the distortion coefficients, which are obtained beforehand through an established calibration process [36], to generate the translation vector and the rotation vector. From these vectors, we can directly calculate the precise location of the camera as

$$\begin{cases} R_{3\times3} = Rodrigues(rvec), \\ Location = -R^{-1} \times tvec, \end{cases} \tag{3}$$

where *rvec* and *tvec* represent the rotation and translation vectors, respectively. The function *Rodrigues*, as proposed by [37], is used to transform a rotation vector from a compact 3×1 form to a more comprehensive 3×3 form.

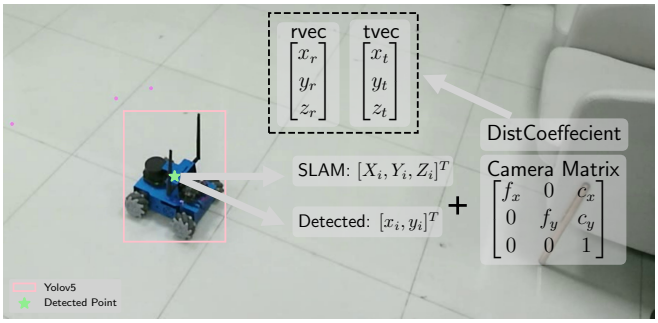


Figure 4: Camera location is determined according to Eq. 3.

4.1.2 Transformation Matrix

In subsequent stages, the system is tasked with generating a trajectory that encapsulates a person's movements within the environment, exclusively reliant on the camera's observational data. Given our scenario, wherein the individual traverses consistently along the same ground level, the z axis

can be disregarded. Consequently, the challenge is transmuted into a task of transforming points from the image plane to the ground plane. To achieve this, a transformation matrix denoted by H is necessitated as follows:

$$H \times [x_i, y_i, 1]^T = [X_i, Y_i, 1]^T. \tag{4}$$

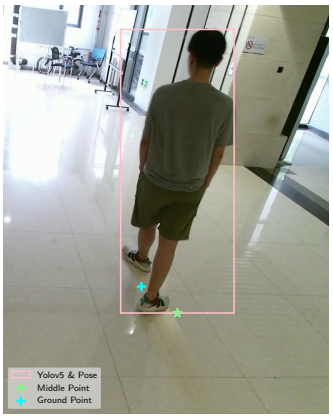
This transformation matrix is more widely recognized as a *homography*, the solutions for which have been previously proposed in scholarly literature [38, 39]. This has been encapsulated within the OpenCV function *findHomography*. As specified in the OpenCV documentation, this function discerns a perspective transformation between two planes given multiple pairs of (*srcPoints*, *dstPoints*). In the context of our study, the *srcPoints* corresponds to points on the image plane, while the *dstPoints* represent points on the ground plane. Therefore, in integrating this process into our system, we disregard the z axis from the reported location data procured from the autonomous robot during the data collection phase, yielding a sequence of 2D-2D pairs. Subsequently, the *findHomography* function is called to derive the transformation matrix H. In later computations, the trajectory can be determined by the application of Eq. 4. In practical experiments, however, this is accomplished using the *perspectiveTransform* function.

4.2 Wi-Fi Mapping Stage

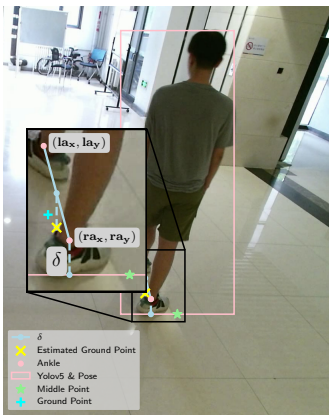
This subsection is centered around two pivotal components within Wi-Fi mapping: pedestrian localization and hyperbolic modeling. The pedestrian localization aspect is designed to estimate an individual's location within an environment based exclusively on camera data, whereas the hyperbolic modeling facet leverages CSI to derive Path Length Change (PLC), subsequently introducing the hyperbolic algorithm for the mapping of the Wi-Fi device. For a better understanding of our system's operations, we will first delve into the details of PLC before discussing hyperbolic modeling.

4.2.1 Pedestrian Localization

At the core of this study, we aim to precisely map a person's trajectory within a predefined environment. To accomplish this, we first focus on pinpointing the exact point where the person makes contact with the ground, a point we denote as the ***ground point***. Our approach involves transitioning this ground point from the image plane, where it is initially captured, onto a ground plane, utilizing a transformation matrix H derived from the camera mapping stage. A fundamental step in this process is the estimation of the ground point for each captured frame. A preliminary approach might involve the application of neural networks, such as YOLOv5, to delineate a bounding box around the person in the image. In this scenario, the midpoint of the bottom edge of the bounding box naturally suggests itself as the ground point. However, careful observations reveal that this proposed ground point may diverge from the actual point of contact with the ground in certain situations, as illustrated in Figure 5a. This discrepancy introduces potential errors in subsequent computations.



(a) YOLOv5 only



(b) YOLOv5 + pose estimation

Figure 5: Ground point detection. (a) illustrates a scenario in which the midpoint deviates significantly from the ground point; (b) demonstrates a combinational method that considerably enhances the accuracy of ground point detection.

In pursuit of accurate location estimation, we augment our system with insights derived from [40]. This approach involves the incorporation of an additional pose estimation modul[41] within our neural network pipeline, which assists in recognizing the position of the person's ankles. This enhancement enables us to estimate the ground point based not solely on the bounding box but also on the relative position of the ankles. In practical terms, we calculate the ground point G as

$$G = [(la_x + ra_x)/2,\ (la_y + ra_y)/2 - \delta]^T, \tag{5}$$

where la and ra denote the positions of the left and right ankles, respectively. An offset value δ is also computed, defined as

$$\delta = bb_{y_{max}} - \max(la_y, ra_y), \tag{6}$$

where $bb_{y_{max}}$ is the y coordinate of the bounding box's bottom edge. Our improved estimation of the ground point, which is more accurate, is displayed in Figure 5b. Once the ground point for each frame has been determined, the person's movement trajectory within the environment can be established using the transformation matrix H described above in Eq. 4.

4.2.2 Path Length Change

Wi-Fi tracks a user's trajectory differently from vision-based methods, observing changes in reflected path length (PLC). As seen in Figure 6a, for a transmitter (T_x) and receiver (R_x),

the reflected path is represented by $\|T_xP\| + \|PR_x\|$. The Fresnel zone forms a cluster of ellipses, denoting locations with a constant reflection path length, with the foci being T_x and R_x. When people move along, they traverse distinct ellipses, each associated with a different reflection path length. The calculation of PLC from raw CSI readings employs the PLCR. Previous studies, such as [30, 42, 43] detail the process of converting CSI to PLCR. From [30], we obtain PLCR using

$$f_D = -\frac{1}{\lambda}\frac{dL(t)}{dt} = -\frac{1}{\lambda}v_r, \tag{7}$$

where f_D, v_r, and $L(t)$ represent the Doppler Frequency Shift (DFS), PLCR, and dynamic path length at time t respectively.

Having established the basic idea of PLCR, we will not delve further into it at this stage. Instead, we will now shift our focus to the process of transitioning from PLCR to PLC. To understand its physical implications and importance in our system, we revisit Figure 6a. As depicted in Figure 6a, a person (P) resides within a zone. The person's trajectory is highlighted in red and its corresponding PLC in ideal situations is visualized in Figure 6b. It is apparent that the person moves from P to A in an anticlockwise direction. When the PLCR is integrated over a certain time span, the result is the PLC, which represents the number of Fresnel zones that the trajectory has crossed within that time frame. More specifically, if N_i is the label for the Fresnel zone at time point i, then plc_{ij} must satisfy:

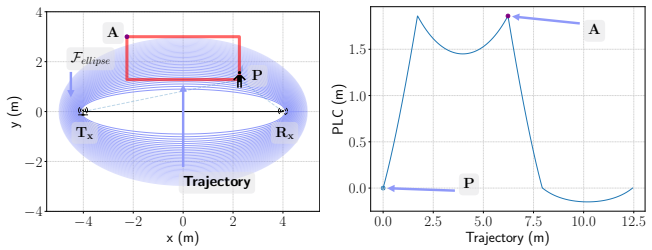
$$plc_{ij} = \int_j^i plcr_x\, dx = N_i - N_j. \tag{8}$$

Consider $N_i = \|P_iT_x\| + \|P_iR_x\|$, where P_i denotes the location point at time instance i. We can rewrite Eq. 8 as

$$plc_i - plc_j = (\|P_iT_x\| + \|P_iR_x\|) - (\|P_jT_x\| + \|P_jR_x\|), \tag{9}$$

where plc_i is plc_{i0} and 0 corresponds to the starting time of the recorded trajectory.

In conclusion, the variation between two PLCs physically signifies the difference in the sum of distances from the trajectory to the two Wi-Fi devices at two different time instances.



(a) Fresnel zone (b) PLC for the trajectory

Figure 6: Fresnel zone model. (a) shows Fresnel zone model for PLCR discussed in [30]; (b) illustrates an ideal PLC for the trajectory.

4.2.3 Hyperbolic Modeling

Once the physical interpretation of PLC is comprehended within our context, the formulation of the hyperbolic model becomes relatively straightforward. Taking into account that the transmitter T_x is securely affixed to the camera, whose location is already known according to the camera mapping stage, Eq. 9 can be enhanced as follows:

$$\|P_iR_x\| - \|P_jR_x\| = (plc_i - plc_j) - (\|P_iT_x\| - \|P_jT_x\|). \tag{10}$$

Consequently, as depicted in Figure 7, for any pair of selected points P_i and P_j, given the corresponding plc_i and plc_j and the location of the transmitter T_x, the unknown receiver R_x is situated on a hyperbola with P_i and P_j as the foci.

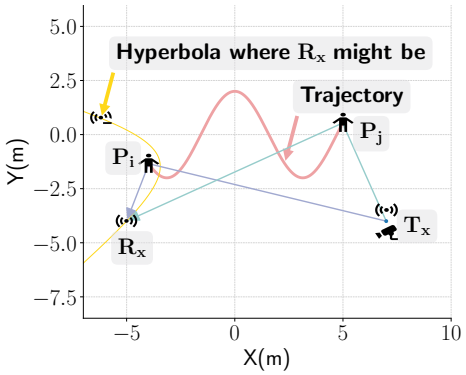


Figure 7: Hyperbolic modeling. When i and j are selected, $\|T_xP_i\|$ and $\|T_xP_j\|$ are immutable. Thus, revealed in Eq. 10, the possible locations of R_x, determined by P_i and P_j, follows a hyperbola with foci at P_i and P_j.

Figure 8a illustrates a scenario in an ideal environment. When P_i, P_j and P_k are selected, the value of $\|P_iT_x\|$, $\|P_jT_x\|$ and $\|P_kT_x\|$ are immutable. According to Eq. 10, three hyperbolas with foci at (P_i, P_j), (P_i, P_k) and (P_j, P_k) respectively are drawn. The intersection of these hyperbolas is considered as the estimation of the receiver R_x. In this ideal situation when errors are eliminated, the estimation perfectly determines the real location of the receiver. Despite our ideal modeling, errors during actual deployment are unavoidable. As shown in Figure 8b, in situations where errors are introduced, the estimated positions can greatly vary, forming a constellation around the receiver's actual location. In response to this challenge, we propose a novel approach that leverages the hyperbolic model in combination with a threshold-based algorithm to mitigate these errors and obtain a robust estimation of the receiver's location. Our proposed solution employs a geometrically intuitive method. As shown in Figure 9, drawing hyperbolas of a larger width increases the probability of encompassing the receiver's correct location, despite potential errors. By delineating multiple hyperbolas of considerable width, we can pinpoint the area of maximum intersection as it exhibits the highest likelihood of being the receiver's actual location. However, resolving multiple intersections of hyperbolas demands substantial computational resources. Our algorithm navigates around this by avoiding the direct solution of intersections. Instead, it determines the likelihood of each candidate location by iteratively applying the algorithm to multiple pairs of foci.

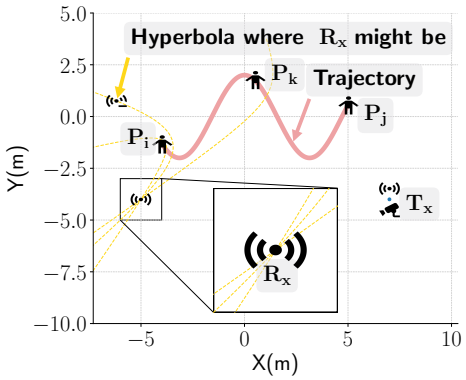
The procedure delineated in Algorithm 1, outlined below, commences by generating a matrix of potential receiver locations within a preset boundary, specifically, a rectangle extending from the bottom-left corner at (x_0, y_0) to the

top-right corner at (x_1, y_1). Next, it calculates the difference between the actual PLC, collected from Wi-Fi, and the estimated PLC. The estimated PLC is calculated according to Eq. 11, upon the candidate receiver's location and the real position of the transmitter gathered from the camera mapping phase. This calculated discrepancy is then assigned to the variable e_t. The error metric e_t is determined via:

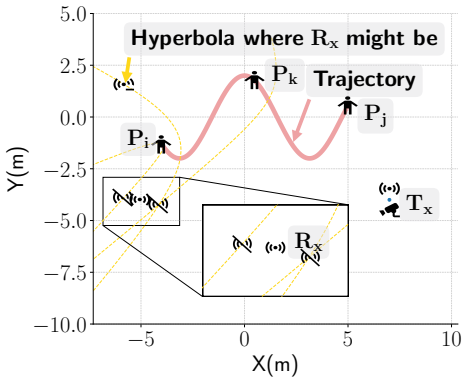
$$e_t = \sum_{i,j} F\Big((plc_i - plc_j) - \big((\|P_iTx\| + \|P_iR_t\|) - (\|P_jT_x\| + \|P_jR_t\|)\big)\Big), \tag{11}$$

where $F(\cdot)$ represents a threshold-based filter, defined as

$$F(x) = \begin{cases} x, \text{if } |x| \geq threshold, \\ 0, \text{if } |x| < threshold. \end{cases} \tag{12}$$



(a) An ideal situation

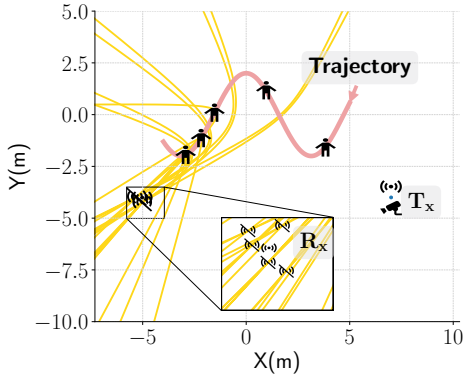


(b) A non-ideal situation

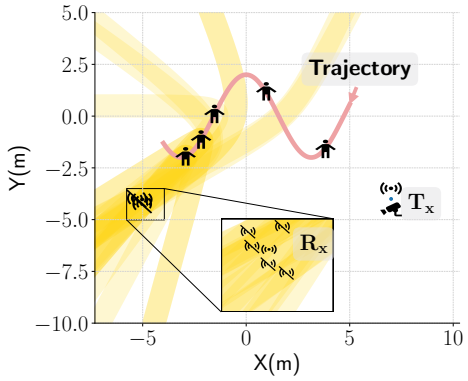
Figure 8: Two situations. (a) In situations devoid of errors, the intersection point of multiple hyperbolas accurately determines the receiver's location; (b) When errors occur, the estimations display significant variations.

In practical terms, the threshold is set manually, based on experience and judgement, and it remains consistent across all tests within the same contexts, namely the Office and the Open-Space scenarios, which will be discussed in detail later in this paper. Specifically, we establish the threshold for the Office scenario at 1.1, and set it at 2.3 for the Open-Space scenario.

Following the computation of errors for each candidate, the candidates are sorted in ascending order based on the error. Ultimately, the receiver's location is estimated by averaging the top 5% of candidates with the smallest errors. In this way, our hyperbolic algorithm robustly maps the Wi-Fi receiver. In the next section, we present implementation and experimental results that demonstrate the effectiveness of our system design.



(a) Thinner width



(b) Larger width

Figure 9: Intuition for our algorithm. (a) The correct location cannot be determined by the intersections; (b) When using a larger width, the correct location can be determined by the area of intersections of multiple hyperbolas.

5 Implementation and evaluation

In this section, we delve into the implementation of our system, followed by an evaluation of its effectiveness. First, we will provide an in-depth view of the system's implementation, including the system configurations and algorithm intricacies. Then, we analyze the evaluation methodology and experimental setup, presenting the resulting data. This will be coupled with a thorough discussion of the results as well.

5.1 Implementation

We first discuss the hardware in the experimental scenarios, followed by a detailed illustration of the software implementation. Second, we present the setup for the experiments.

5.1.1 Hardware

The system, predominantly developed using C++, is executed on a Thinkpad T14 Gen2 laptop, equipped with an Intel Core i7-10510U 4.9 GHz CPU. This laptop serves as a central hub, facilitating connections between all other components. The

(a) KinectV2 (b) Robot (c) Wi-Fi device

Figure 10: (a) KinectV2 camera in our use case provides only the RGB frames while Depth information is ignored; commercial RGB cameras can replace it. (b) Autonomous bot provides 3D locations, depicting the world coordinates, which many commercial robotic vacuum cleaners are capable of. (c) Wi-Fi device is in a commercial off-the-shelf fashion furnished with Intel 5300 NICs.

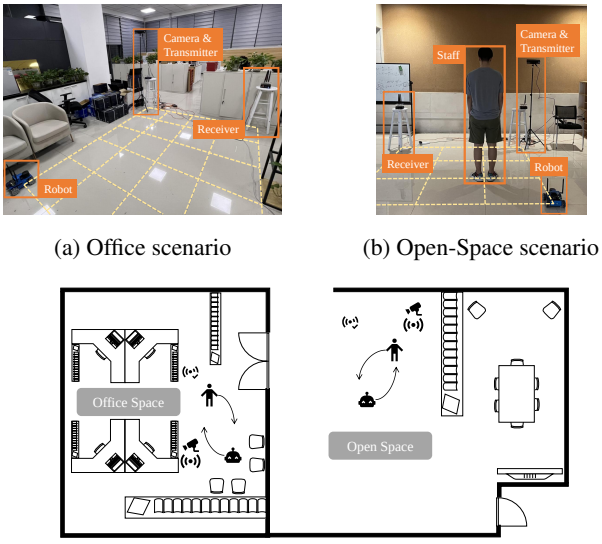
neural networks, implemented with the PyTorch library, run on a Flask-based server hosted on a MacBook Several other key components, as depicted in Figure10, are also employed The KinectV2 camera is used specifically for extracting RGB frames, whose functionality can be effectively replaced by any standard commercial RGB camera after calibration. An autonomous robot equipped with SLAM technology, akin to the ones found in commercial robotic vacuum cleaners, also forms an integral part of this setup. We also utilize a pair of commercial off-the-shelf Wi-Fi devices equipped with Intel 5300 NICs. One device serves as a transmitter and is attached to the camera, while the other operates as a receiver and is situated at an unkown location.

5.1.2 Software

Data processing tasks are primarily handled using std17 version of C++. We integrate various libraries to streamline these processes. Notably, OpenCV is implemented to provide an array of computer vision-related functionalities, such as the *Rodrigues* function referenced in Eq. 3. Additionally, the *freenect2*[44] library enables connections of streamlining frames from the KinectV2 to the terminal.

5.1.3 Experiment Setups

As shown in Figure 11, we establish office and open-space scenarios to evaluate the performance of our algorithm. In both scenarios, the camera is deliberately adjusted to preclude direct visibility of the receiver. The world coordinates in both settings are defined by the autonomous robot, with the center of the coordinates coinciding with the center of the robot, as shown in both Figure11a and Figure11b. In the office scenario, we position both the camera and the transmitter at $[3.0\text{m}, 0]$, with the receiver located at $[3.0\text{m}, 2.4\text{m}]$. In the open-space scenario, the camera and the transmitter are situated at $[2.4\text{m}, 0]$, with the receiver at $[2.4\text{m}, 2.4\text{m}]$. In both scenarios, the program autonomously selects between 15 to as many as 100 individual frames during the camera mapping stage to ensure sufficient accuracy for subsequent stages. During the Wi-Fi mapping stage, we enlist the aid of two volunteers to traverse three predefined and several randomly chosen trajectories in both scenarios to validate the performance of the localization algorithm.



(a) Office scenario (b) Open-Space scenario

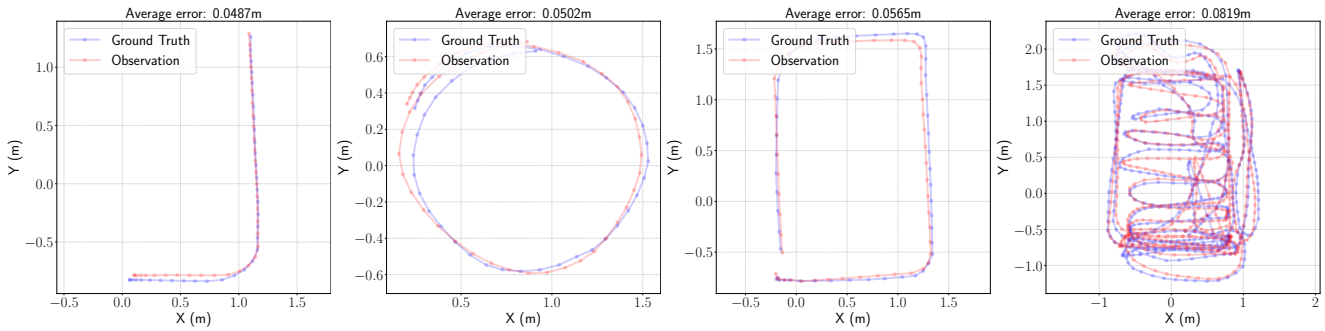
(c) Floor plan

Figure 11: Experiment setup. The office (a) and open-space (b) scenarios have been established for validation. (c) presents the floor plans of both scenarios.

5.2 Evaluation

Firstly, we evaluate the effectiveness of the transformation matrix. Our fundamental approach involves determining the Euclidean distance between the location reported by the autonomous robot and the location perceived by the camera at any given time instance. The robot is programmed to follow a predefined trajectory[1]. Subsequently, we assess the performance of our proposed hyperbolic model during the Wi-Fi mapping phase. In this setting, we define an error as the Euclidean distance between the algorithmically determined and

[1]Given that camera localization is based on the Solve-PnP algorithm—a methodology extensively analyzed and validated for efficiency—we do not present evaluations of its accuracy in this study.



(a) L (b) Circle (c) Rectangle (d) Random

Figure 12: Transformation matrix accuracy. Four trajectories, corresponding to the ones used in later evaluation, have been conducted. The ground truth is provided by the SLAM-bot, while the observation is solely based on the camera's data.

manually measured locations of the unknown Wi-Fi. We display this performance using the experiment setups previously delineated.

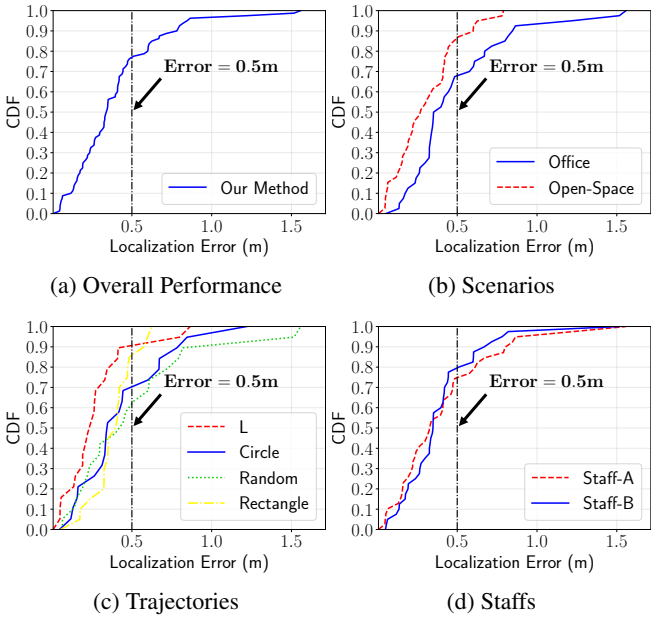
5.2.1 Transformation Matrix

A series of tests is conducted within two distinct scenarios to thoroughly evaluate the holistic performance of the transformation matrix. As visualized in Figure 12, we utilize four diverse trajectories - L-shape, circle, rectangle, and random - to graphically exhibit the precision of the transformation matrix. During the experiment, the terminal concurrently retrieves the observed and reported locations from the camera pipeline and the robot, respectively, at any given moment. Once an ample number of location pairs have been amassed, we compute the Euclidean distance between these two sets of coordinates to ascertain the extent of error. Remarkably, the method demonstrates consistent performance across both scenarios, displaying satisfactory accuracy in all four trajectories. It registers an average deviation of merely 0.05 meters in the three predefined trajectories, with a slightly higher deviation of 0.08 meters in the most unpredictable random trajectory, as shown in Figure 12d. These results are regarded as satisfactory for progression to the subsequent stages of our research.

5.2.2 Wi-Fi Mapping Evaluation

Multiple experiments are conducted across various scenarios to evaluate the robustness of our algorithm. In the course of the actual experiments, an individual ambulates while the Wi-Fi infrastructure initiates the collection of Channel State Information (CSI), concurrently signaling the camera to commence recording the trajectory. Once the CSI data has been gathered, the Wi-Fi infrastructure once again communicates with the camera, and the collected data is subsequently processed and aligned within the terminal to determine the location of the unknown Wi-Fi. It should be noted that the camera mapping is predetermined and verified using the methods previously discussed, which achieved centimeter-level accuracy. As illustrated in Figure 13a, approximately 80% of all test cases attain an accuracy of less than 0.5 meters, with an average accuracy of 0.339 meters. To further underscore the robustness of our algorithm, we delve into three distinct aspects in the following sections.

Impact of Environment. To evaluate the algorithm's effectiveness across varied environments, we conduct experiments in two distinct scenarios as illustrated in Figure 11. These scenarios encompass an office environment and an open-space environment, with their setup detailed earlier. The office environment, compared to the open outdoor space, presents a higher number of obstacles, thereby amplifying the multipath effects on the collected CSI data. Yet, as depicted in Figure 13b, our post-experiment analysis reveals that 70% of the collected data achieve an accuracy of less than 0.5 meters in both scenarios. Notably, almost 90% of the data from the open-space environment attain an accuracy of less than 0.5 meters. Furthermore, the average precision metrics for both scenarios stand at 0.27 meters for the open-space and 0.35 meters for the office setting.

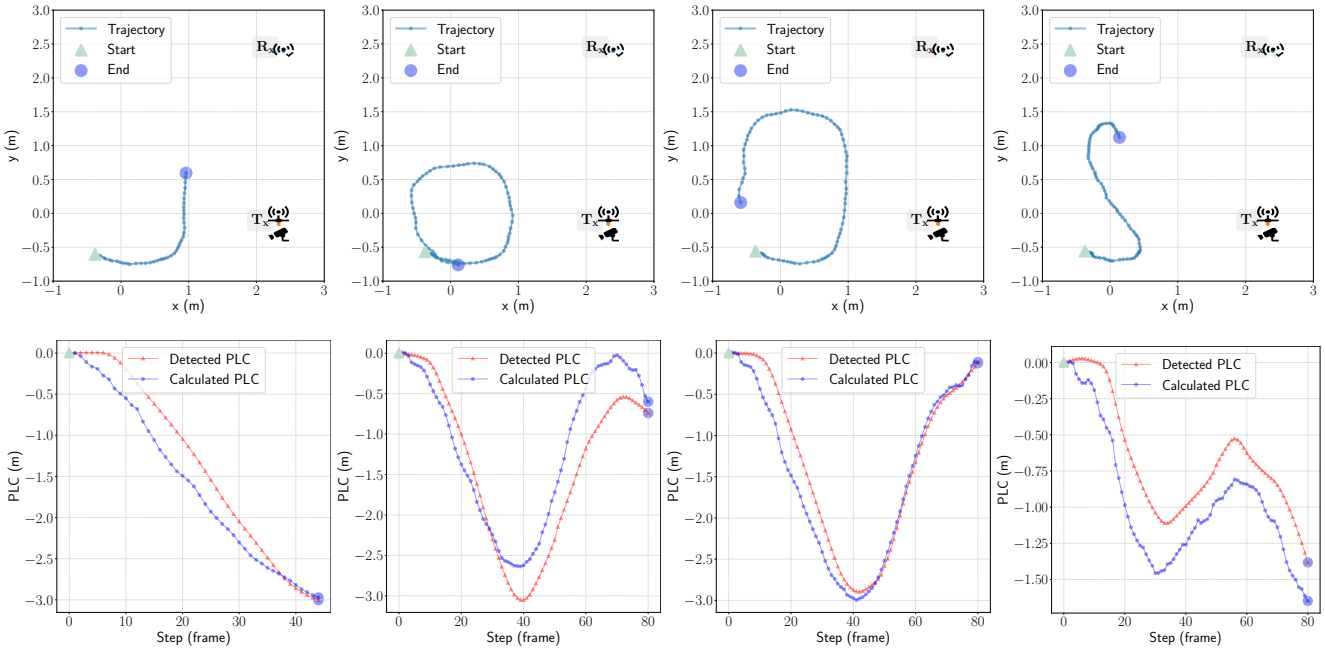


(a) Overall Performance (b) Scenarios

(c) Trajectories (d) Staffs

Figure 13: Wi-Fi localization accuracy. (a) for the overall Localization Error; (b) for localization errors in two different Scenarios; (c) for localization errors in four identical trajectories; (d) for localization errors conducted by two individuals.

Impact of Trajectory. Recognizing that individuals do not follow rigid paths consistently in real-world scenarios, we instruct our staff to traverse three predefined and one randomly selected trajectory for a comprehensive evaluation of



(a) L (b) Circle (c) Rectangle (d) Random

Figure 14: Trajectories and PLCs. Above are the trajectories ascertained exclusively based on Camera Observations. Below are the corresponding plots of PLCs. Detected PLCs are collected via Wi-Fi. Calculated PLCs are derived according to Eq. 9, considering the known locations of the receiver R_x, the transmitter T_x, and assuming $j = 0$.

the algorithm's performance. The results of these four types of trajectories, as recorded by the camera, and their corresponding PLCs are visually represented in Figure 14. Importantly, the *Detected PLCs* are acquired through Wi-Fi devices, while the *Calculated PLCs* are computed based on Eq. 9, factoring in the actual locations of the receiver R_x, the transmitter T_x, and with $j = 0$. Our analyses expose inherent biases between the two sets of PLCs across all trajectory types. However, the resultant localization errors consistently fall within an acceptable range, with up to 60% achieving an accuracy of less than 0.5 meters. As shown in Figure 13c, examining further the accuracy details for the four trajectories, we observe the following: both the L-shaped and rectangle trajectories exhibit high accuracy, with 90% of instances yielding a precision of less than 0.5 meters. The L-shaped trajectory averages a precision of 0.24 meters, while the rectangle trajectory is at 0.39 meters. The circle trajectory sustains a precision rate of 70% and an average error of 0.34 meters. In contrast, the random trajectory hits the 60% precision mark, averaging an error of 0.44 meters.

Impact of User Diversity. Recognizing the diversity of potential users for the proposed localization algorithm, we pursue accuracy validation measures encompassing different individuals within the workforce. Experimental trials are conducted with two staff members, each presenting distinct walking patterns, fluctuating velocities, and diverse physical attributes. Throughout the course of the experiments, variations are observed in the discrepancies between the paths predicted by the PLC and the actual trajectories followed by different staff members. Despite these discrepancies, the mean localization error derived from Wi-Fi signals for both participants yields satisfactory outcomes. As shown in Figure 13d, particularly, the accuracies achieved for staff members A and B are less than 0.5 meters, with success rates of 75% and 80% respectively. Moreover, the average deviations are as low as 0.32 and 0.34 meters for staff members A and B, respectively.

6 Conclusions

In this paper, we unveiled a new system named *Visual Assistant*. The system was specifically designed to autonomously create environmental maps and precisely locate cameras and Wi-Fi infrastructure. The system operates in two key stages: camera mapping and Wi-Fi mapping. During the camera mapping stage, we harnessed modern computer vision and SLAM (Simultaneous Localization and Mapping) technologies to effectively map the environment. In the Wi-Fi mapping stage, we introduced a hyperbolic model that uses camera-recorded trajectories and changes in path length to identify unknown Wi-Fi infrastructure. We tested the system using a commercial Wi-Fi network, a standard camera, and a robot. The results showed centimeter-level accuracy in camera mapping and decimeter-level accuracy in Wi-Fi mapping. In summary, our system marks a significant step forward in the field of indoor device-free tracking systems.

Funding

This work was supported by the National Natural Science Foundation of China under Grant 62472307, 62372324 and 62402336.

Author Contributions

Conceptualization, Jiajie CAO, Jinfan LIU and Xinyu TONG; methodology, Jiajie CAO and Jinfan LIU; software, Jinfan

LIU; validation, Jinfan LIU, Renrui TAN, Jiajie CAO and Xinyu TONG; formal analysis, Jiajie CAO and Jinfan LIU; investigation, Jiajie CAO and JInfan LIU; resources, Xinyu TONG and Renrui TAN; data curation, Jinfan LIU; writing—original draft preparation, Jinfan LIU and Jiajie CAO; writing—review and editing, Renrui TAN and Xinyu TONG; visualization, Jinfan LIU; supervision, Xinyu TONG; project administration, Xinyu TONG; funding acquisition, Xinyu TONG.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data Avaliable

The data and materials used in this study are availableupon request from the corresponding author.

References

[1] Tong, X., Li, H., Tian, X., Wang, X.: Triangular antenna layout facilitates deployability of csi indoor localization systems. In 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–9 (2019). https://doi.org/10.1109/SAHCN.2019.8824894

[2] Ayyalasomayajula, R., Arun, A., Wu, C., Rajagopalan, S., Ganesaraman, S., Seetharaman, A., Jain, I.K., Bharadia, D.: LocAP: Autonomous millimeter accurate mapping of WiFi infrastructure. In 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20), pp. 1115–1129 (2020)

[3] Tong, X., Wang, H., Liu, X., Qu, W.: Mapfi: Autonomous mapping of wi-fi infrastructure for indoor localization. IEEE Transactions on Mobile Computing **22**(3), 1566–1580 (2023) https://doi.org/10.1109/TMC.2021.3108155

[4] Chen, S., Tan, R., Wang, Z., Tong, X., Li, K.: Voicemap: Autonomous mapping of microphone array for voice localization. IEEE Internet of Things Journal **11**(2), 2909–2923 (2024) https://doi.org/10.1109/JIOT.2023.3294937

[5] Xiong, J., Jamieson, K.: Arraytrack: a fine-grained indoor location system. In 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), pp. 71–84 (2013)

[6] Zheng, Y., Kuang, Y., Sugimoto, S., Astrom, K., Okutomi, M.: Revisiting the pnp problem: A fast, general and optimal solution. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2344–2351 (2013)

[7] Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate $O(n)$ solution to the pnp problem. International journal of computer vision **81**, 155–166 (2009)

[8] Terzakis, G., Lourakis, M.: A consistently fast and globally optimal solution to the perspective-n-point problem. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pp. 478–494 (2020). Springer

[9] Hesch, J.A., Roumeliotis, S.I.: A direct least-squares (dls) method for pnp. In 2011 International Conference on Computer Vision, pp. 383–390 (2011). IEEE

[10] Gjengset, J., Xiong, J., McPhillips, G., Jamieson, K.: Phaser: Enabling phased array signal processing on commodity wifi access points. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, pp. 153–164 (2014)

[11] Goswami, A., Ortiz, L.E., Das, S.R.: Wigem: A learning-based approach for indoor localization. In Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies, pp. 1–12 (2011)

[12] Joshi, K., Hong, S., Katti, S.: Pinpoint: Localizing interfering radios. In 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), pp. 241–253 (2013)

[13] Kotaru, M., Joshi, K., Bharadia, D., Katti, S.: Spotfi: Decimeter level localization using wifi. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, pp. 269–282 (2015)

[14] Kumar, S., Gil, S., Katabi, D., Rus, D.: Accurate indoor localization with zero start-up cost. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, pp. 483–494 (2014)

[15] Ladd, A.M., Bekris, K.E., Rudys, A., Marceau, G., Kavraki, L.E., Wallach, D.S.: Robotics-based location sensing using wireless ethernet. In Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, pp. 227–238 (2002)

[16] Sen, S., Radunovic, B., Choudhury, R.R., Minka, T.: You are facing the mona lisa: Spot localization using phy layer information. In Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp. 183–196 (2012)

[17] Gao, C., Harle, R.: Semi-automated signal surveying using smartphones and floorplans. IEEE Transactions on Mobile Computing **17**(8), 1952–1965 (2018) https://doi.org/10.1109/TMC.2017.2776128

[18] Zhang, L., Tan, S., Chen, Y., Yang, J.: A phoneme localization based liveness detection for text-independent speaker verification. IEEE Transactions on Mobile Computing **22**(10), 5611–5624 (2023) https://doi.org/10.1109/TMC.2022.3187432

[19] Li, X.-Y., Liu, H., Zhang, L., Wu, Z., Xie, Y., Chen,

G., Wan, C., Liang, Z.: Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint. IEEE/ACM Transactions on Networking **27**(5), 1945–1958 (2019) https://doi.org/10.1109/TNET.2019.2933269

[20] Lee, G., Jung, S.-H., Han, D.: An adaptive sensor fusion framework for pedestrian indoor navigation in dynamic environments. IEEE Transactions on Mobile Computing **20**(2), 320–336 (2021) https://doi.org/10.1109/TMC.2019.2946809

[21] Tong, X., Liu, K., Tian, X., Fu, L., Wang, X.: Fineloc: A fine-grained self-calibrating wireless indoor localization system. IEEE Transactions on Mobile Computing **18**(9), 2077–2090 (2019) https://doi.org/10.1109/TMC.2018.2871206

[22] Hillyard, P., Patwari, N.: Never use labels: Signal strength-based bayesian device-free localization in changing environments. IEEE Transactions on Mobile Computing **19**(4), 894–906 (2020) https://doi.org/10.1109/TMC.2019.2901782

[23] Kaltiokallio, O., Hostettler, R., Patwari, N.: A novel bayesian filter for rss-based device-free localization and tracking. IEEE Transactions on Mobile Computing **20**(3), 780–795 (2021) https://doi.org/10.1109/TMC.2019.2953474

[24] Chen, M., Liu, K., Ma, J., Gu, Y., Dong, Z., Liu, C.: Swim: Speed-aware wifi-based passive indoor localization for mobile ship environment. IEEE Transactions on Mobile Computing **20**(2), 765–779 (2021) https://doi.org/10.1109/TMC.2019.2947667

[25] Zhang, D., Xie, W., Liao, Z., Zhu, W., Jiang, L., Zou, Y.: Beyond rss: A prr and snr aided localization system for transceiver-free target in sparse wireless networks. IEEE Transactions on Mobile Computing **21**(11), 3866–3879 (2022) https://doi.org/10.1109/TMC.2021.3063629

[26] Narzullaev, A., Selamat, M.H., Sharif, K.Y., Muminov, Z.: Wi-Fi received signal strength-based hyperbolic location estimation for indoor positioning systems. International Journal of Information and Communication Technology **14**(2), 175 (2019)

[27] Essiben, J.-F.D., Ihonock, L.E., Hedin, E.R., Joe, Y.S.: A Novel Hyperbolic Position Location Estimation in Wi-Fi Environments. Yong S. Joe Journal of Engineering Research and Application **9**(3), 13–19 (2019)

[28] Liu, J., Li, W., Gu, T., Gao, R., Chen, B., Zhang, F., Wu, D., Zhang, D.: Towards a Dynamic Fresnel Zone Model to WiFi-based Human Activity Recognition. In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), pp. 1–24 (2023)

[29] Xu, X., Meng, X., Tong, X., Liu, X., Xie, X., Qu, W.: Hypertracking - Exploring the Hyperbolic Model for Non -line-of-sight Device-free Wi-Fi Tracking. In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, pp. 1–26 (2023)

[30] Ge, W., Tian, Y., Liu, X., Tong, X., Qu, W., Zhong, Z., Chen, H.: Crosstrack: Device-free cross-link tracking with commodity wi-fi. IEEE Internet of Things Journal **10**(20), 18028–18041 (2023)

[31] Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based slam. IEEE Intelligent Transportation Systems Magazine **2**(4), 31–43 (2010)

[32] Aulinas, J., Petillot, Y., Salvi, J., Lladó, X.: The slam problem: a survey. In Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, pp. 363–371 (2008)

[33] Xi, W., Ou, Y., Peng, J., Yu, G.: A new method for indoor low-cost mobile robot slam. In 2017 IEEE International Conference on Information and Automation (ICIA), pp. 1012–1017 (2017). https://doi.org/10.1109/ICInfA.2017.8079050

[34] Khanam, R., Hussain, M.: What is yolov5: A deep look into the internal features of the popular object detector. arXiv preprint arXiv:2407.20892 (2024)

[35] Bradski, G.: The OpenCV Library. https://opencv.org/. Accessed: 2025-06-01 (2000)

[36] Zhang, L., Koch, R.: An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. Journal of visual communication and image representation **24**(7), 794–805 (2013)

[37] Gallego, G., Yezzi, A.: A compact formula for the derivative of a 3-D rotation in exponential coordinates. Journal of Mathematical Imaging and Vision **51**(3), 378–384 (2015)

[38] Dubrofsky, E.: Homography estimation. B.Sc., Carleton University (2009)

[39] Agarwal, A., Jawahar, C., Narayanan, P.: A Survey of Planar Homography Estimation Techniques. https://cdn.iiit.ac.in/cdn/cvit.iiit.ac.in/images/JournalPublications/2005/A_survey_of_planar_homography_estimation.pdf

[40] Lima, J.P., Roberto, R., Figueiredo, L., Simões, F., Thomas, D., Uchiyama, H., Teichrieb, V.: 3d pedestrian localization using multiple cameras: A generalizable approach. Machine Vision and Applications **33**(4), 61 (2022)

[41] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M.: Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204 (2020)

[42] Qian, K., Wu, C., Yang, Z., Yang, C., Liu, Y.: Decimeter level passive tracking with wifi. In Proceedings of the 3rd Workshop on Hot Topics in Wireless, pp. 44–48 (2016)

[43] Qian, K., Wu, C., Yang, Z., Liu, Y., Jamieson, K.: 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, pp. 1–10 (2017)

[44] Xiang, L., Echtler, F., Kerl, C., Wiedemeyer, T., Lars, hanyazou, Gordon, R., Facioni, F., laborer2008, Wareham, R., Goldhoorn, M., alberth, gaborpapp, Fuchs, S., jmtatsch, Blake, J., Federico, Jungkurth, H., Mingze, Y., vinouz, Coleman, D., Burns, B., Rawat, R., Mokhov, S., Reynolds, P., Viau, P.E., Fraissinet-Tachet, M., Ludique, Billingham, J., Alistair: Libfreenect2: Release 0.2. https://doi.org/10.5281/zenodo.50641 . https://doi.org/10.5281/zenodo.50641