EasiTrack: Decimeter-Level Indoor Tracking with Graph-based Particle Filtering

Chenshu Wu, *Member, IEEE,* Feng Zhang, *Member, IEEE,* Beibei Wang, *Member, IEEE,* and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—Despite decades of efforts, existing indoor location systems do not easily scale with low cost while maintaining high accuracy. We present EasiTrack, an indoor tracking system that achieves decimeter accuracy using a single commodity WiFi Access Point (AP) under Non-Line-Of-Sight conditions and can deploy at scale with almost zero cost. EasiTrack makes two key technical contributions: First, it incorporates RF-based inertial measurement algorithms that can accurately infer a target's moving distance purely using the RF signals received by itself. Second, EasiTrack devises a map-augmented tracking algorithm that outputs fine-grained locations by jointly leveraging the distance estimates and an indoor map that is ubiquitously available nowadays. We build a fully functional real-time system centering around a satellite-like architecture, which enables EasiTrack to support an unlimited number of clients. We have deployed EasiTrack in 7 different scenarios (including offices, hotels, museums, and manufacturing facilities) to track both humans and machines. The results reveal that EasiTrack achieves a median 0.25 m and 90%tile 0.69 m accuracy in distance measurement, a median 0.58 m and 90%tile 1.33 m location accuracy for tracking objects and a median 0.70 m and 90%tile 1.97 m accuracy for tracking humans in both LOS and NLOS scenarios and supports a broad coverage of 50 m×60 m using a single AP. It is also verified that EasiTrack can be easily deployed in massive buildings with little cost, promising a practical solution for ubiquitous indoor tracking.**

***Index Terms*—Indoor tracking, wireless localization, indoor maps, RIM, particle filter**

I. Introduction

Indoor positioning systems (IPS) using commodity off-the-shelf (COTS) WiFi are among the most promising solutions for ubiquitous tracking thanks to the wide availability of already deployed WiFi infrastructure. They should ideally satisfy the following four requirements:

- **Low(Zero)-cost**: They should be easy to install and deploy with low or ideally zero efforts. Ideally, they should be able to locate a mobile device using a *single arbitrarily installed* AP (even without knowing any of its information like location and orientation), without requiring any hardware changes at either side.
- **Scalable**: The systems should be scalable in two folds: They should scale to a large number of different buildings and should support large populations of concurrent mobile devices, just as GPS does, both with negligible costs.
- **Large coverage**: The systems should cover a large area, be it close or at a distance, having Line-Of-Sight (LOS) or behind multiple walls to the AP, with consistently high accuracy.
- **Accurate**: They should be able to provide sub-meter accuracy, as demanded by many applications [45]. Such accuracy is needed to enable in-building navigation, for example, directing a customer to a product in the store or a robot to a work station.

If the above requirements are all satisfied, we can imagine an indoor location system becomes a ubiquitous "indoor GPS" that is made available anywhere having WiFi signals and for any device with a commodity WiFi chip.

To the best of our knowledge, however, no existing technology satisfies all these requirements. Recent techniques based on Angle of Arrival (AoA) or Time of Flight (ToF) [19], [41], [58] could yield sub-meter median accuracy. However, they usually require large bandwidth (for ToF) [41], [56] and many phased antennas (for AoA) [57], [58] for good performance, both of which are limited on COTS WiFi. Moreover, they all require the precise installation of multiple APs (to obtain their reference locations and orientations). Fingerprint-based approaches are neither accurate enough nor cost-efficient due to expensive prior training [7], [35], [39], [61]. Other technologies combining with inertial sensors are deployable but have limited accuracy [17], [31], [33], [43], [49]. The latest released 802.11mc [1] supports round-trip time measurements, but does not offer sub-meter resolution either, and still needs multiple APs for multilateration. Pedestrian Dead-Reckoning (PDR) solutions [17], [22] are easy to deploy, yet face significant drawbacks of accuracy. A recent work WiBall [65] takes an important step towards the above goals by estimating moving distances via WiFi signals. However, it only produces accurate locations assuming detailed floorplan with the knowledge of corridors, crossing, doors, etc., rendering it costly and not scalable.

This paper presents EasiTrack[1], an indoor tracking system that meets all the four requirements above. EasiTrack achieves sub-meter accuracy in both LOS and Non-LOS (NLOS) scenarios using a single arbitrarily installed AP, without knowing its location. It can be easily deployed at scale, with no extra inputs but a plain imagery floorplan of the area of interests, which would be needed anyway for most location-based services and is ubiquitously available. EasiTrack's architecture centers around a satellite-like design, which supports massive concurrent clients without affecting the channel capacity, and

C. Wu, F. Zhang, B. Wang, and K. J. R. Liu are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, 20742, and the Origin Wireless Inc., Greenbelt, MD, 20770
E-mail: {cswu, fzhang15, bebewang, kjrliu}@umd.edu

[1] Coined from **E**asy, **a**ccurate, and **s**calable **i**ndoor **track**ing.

preserves privacy since a client never needs to announce its presence but only passively listens to the AP.

As shown in Fig. 1, to track a client, EasiTrack first estimates the moving distance from its measured Channel State Information (CSI) and infers the moving direction from the built-in inertial sensors. It then employs an effective map-augmented tracking algorithm to incorporate distance and orientation estimates, which outputs fine-grained locations. Specifically, EasiTrack advances the state-of-the-art in tracking in two distinct ways:

(1) Distance estimation by leveraging multipaths: EasiTrack first contributes a super-resolution distance estimation algorithm, which accurately estimates the incremental displacement of a moving radio merely by the CSI measured by itself. The idea is derived from the recently proposed RF-based Inertial Measurement (RIM) [52], which estimates multiple motion parameters, including moving distance, heading direction, and rotating angle using RF signals. Original RIM resolves all these parameters using more antennas than what is available on commodity devices. In EasiTrack, we employ RIM for only distance estimation, which requires a minimum of two antennas. The unique insight of moving distance estimation using RIM is that, as shown in Fig. 2, when an array translates, two of its antennas may traverse through the same locations in the space, one preceded by the other with a particular time delay. Upon the moment the latter antenna arrives at the same location as the former one, the array will have traveled for exactly a distance equal to the antenna separation. Noticing this, we determine the time delays by using the multipath profiles at different locations as virtual antennas. The moving speeds can then be calculated by dividing the antenna separation by the time delays, and the moving distance can accordingly be derived by integrating speeds over time.

The algorithm is demonstrated to be highly accurate, yet only suitable for relatively constrained movements, such as shopping carts, robots, and automated guided vehicles (AGVs). To enable EasiTrack for free human tracking as well, we further incorporate the approach developed in [65] that infers moving distances for unconstrained motions from CSI. Then we design a universal architecture that allows switching the two methods for different scenarios. By truly leveraging multipaths, instead of resolving and combating them like previous works [19], [58], EasiTrack's distance estimation is location independent, working under both LOS and NLOS conditions.

(2) Map-augmented probabilistic tracking: While the distance estimation is accurate, it does not ensure sub-meter tracking accuracy when integrated with orientations obtained by inertial sensors, which suffer from significant accumulative errors. EasiTrack employs indoor maps to cope with these uncertainties. Imagine that Google Maps improves GPS accuracy using outdoor road maps to place a car on the correct road. Similarly, indoor maps impose geometric constraints to a target's in-building movements. For example, a user does not penetrate a wall. Existing works have exploited indoor maps. However, some of them face accuracy limitations dictated by smartphone dead-reckoning [22], [33], while others like WiBall [65] apply rule-based enhancement, which relies on detailed structure knowledge of a floorplan.

In EasiTrack, we design a novel map-augmented probabilistic approach for accurate indoor tracking. Our algorithm models an indoor map as a weighted graph and feeds it into a graph-based particle filter (GPF), which jointly handles distance and orientation errors and outputs accurate locations. The graph-based model enables EasiTrack to use a small number of particles (e.g., 50), thereby making it run in real-time, even on resource-limited devices. Also, the proposed GPF requires merely information on accessible and inaccessible areas from the map, which can be directly extracted from a plain floorplan image that is ubiquitously available. It does not need any structured knowledge, such as corridors, crossings, doors, or rooms, etc. Thus, EasiTrack can easily scale to many buildings with little cost.

Summary of results: We build a fully functional real-time system of EasiTrack, consisting of hardware prototype using commodity WiFi chipsets and software sets. To comprehensively evaluate the accuracy and deployment simplicity, we have deployed EasiTrack in 7 different buildings to track both humans and machines, including three office buildings, one museum, one hotel, and two manufacturing facilities, all with complex multipath environments. Few of previous systems have been tested under such stressful and extensive conditions. Our key results reveal the following: 1) EasiTrack achieves a median accuracy of 0.25 m and 90%tile accuracy of 0.69 m in moving distance estimation by using RIM with a single COTS WiFi chipset; 2) EasiTrack achieves a median 0.58 m and 90%tile 1.33 m location error for tracking objects, and a median 0.70 m and 90%tile 1.97 m error for tracking humans, in both LOS and NLOS conditions with a single AP; 3) It is almost zero cost to deploy EasiTrack in a new building. Provided the map, the AP can be set up in minutes at any location that gives coverage; 4) EasiTrack can track in a large area similar to the AP's signal coverage. In our deployment, it can track over a 50 m×60 m area with one AP. Additionally, the results also confirm the robustness and scalability of EasiTrack, outperforming the state-of-the-art approaches [65].

Contributions: The core contribution of this paper is EasiTrack, a ubiquitous indoor tracking system that achieves sub-meter accuracy in both LOS and NLOS scenarios using a single unknown AP and scales to massive buildings and end clients with (almost) zero costs. Despite over three decades of research, enabling such capabilities in one IPS system has been regarded as challenging, and we are unaware of any prior works by which the same would have been achieved. EasiTrack also contributes a CSI-based moving distance estimation algorithm and a map-augmented tracking algorithm, each of which could separately complement existing techniques. It also offers a streamlined reference framework for scalable IPS design and implementation. With this, we believe EasiTrack sets the stage for a ubiquitous IPS solution for world-wide deployment.

II. Overview

Fig. 1 depicts an overview of EasiTrack's architecture. The left part illustrates the satellite-like protocol of EasiTrack. In an example usage scenario, a client, which could be a mobile, wearable, robot, automated guided vehicle (AGV), or any

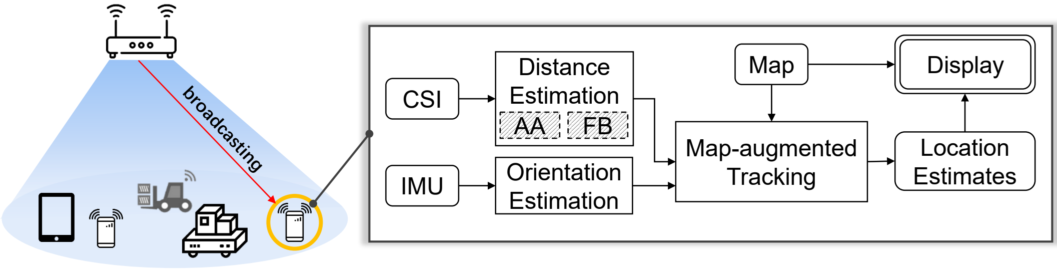


Fig. 1: Overall architecture: EasiTrack passively collects CSI from one AP and records inertial sensors built in mobile clients. It first calculates the moving distance from CSI by leveraging rich multipaths, and orientation from IMU. Then it fuses the two types of estimates to continuously track, in combination with an indoor map. By this architecture, EasiTrack works like an "indoor GPS" (but without the need of multiple "satellites") that supports large populations of clients.

other electronic equipped with a WiFi radio and inertial sensors (IMU), passively listens to an AP and measures CSI from the incoming packets. The client also reads its built-in inertial sensors. EasiTrack's core engine, running on the client, infers the moving distance and orientation from the measurements and incorporates them together to track the client's continuous locations. The AP is simple: It does nothing but periodically sending signals that could be detected by the clients. Such simple-core, complex-edge architecture design provides significant benefits: 1) Scalable: It allows concurrent positioning of a large population of clients, without affecting the channel capacity; 2) Privacy: It preserves user privacy because the client does not need to send out any signal which may expose its presence, thereby preventing others from sniffing its location; 3) Flexible: The clients could perform different tracking algorithms as needed, all under the same environments.

The right part of Fig. 1 shows EasiTrack's work flow on a mobile client. There are two key modules:

Distance estimation: EasiTrack involves two distinct approaches, namely *Antenna Alignment* (AA) [52] and *Focusing Ball* (FB) [65], to estimate the moving distance traversed by the user. Both approaches leverage the rich multipaths to estimate moving distances, yet are applicable in different tracking scenarios: AA is highly accurate and is suitable for tracking objects like shopping carts, robots, and industrial AGVs with relatively constrained motions; while FB is less reliable compared to AA, yet it is more generic and is superior for unconstrained human tracking. In EasiTrack system, the two approaches can switch as needed on the fly.

Map-augmented tracking: EasiTrack tracks a user's location by fusing the moving distance estimated from CSI and the moving direction measured by inertial sensors. Although the logic flow is similar to conventional dead-reckoning, we present an effective, practical design that incorporates indoor maps for precise tracking in a scalable way. Specifically, EasiTrack takes the indoor map as input and transforms it into a weighted graph. The output graph is then fed into a *Graph-based Particle Filter* (GPF), which leverages the geometric constraints imposed by the map and jointly learns the accurate 2D location and orientation of a target when it moves. The location estimates are then displayed to users together with the map. Since the proposed GPF only uses an ordinary indoor map (e.g., an image of the floorplan), it can easily scale to massive buildings with few costs.

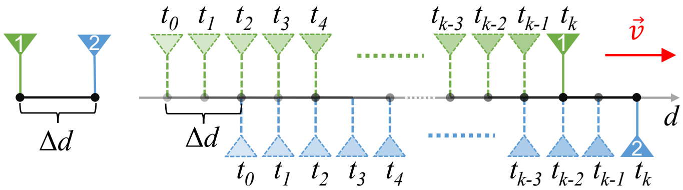

1
2
Δd
t_0 t_1 t_2 t_3 t_4
t_{k-3} t_{k-2} t_{k-1} t_k
$\vec{v}$
Δd
d
t_0 t_1 t_2 t_3 t_4
t_{k-3} t_{k-2} t_{k-1} t_k

Fig. 2: Illustration of virtual antenna alignment. When the array moves, after a displacement of Δd, antenna 1 will arrive at t_2 at a location where antenna 2 was at t_0. Thus the moving speed can be derived as $\Delta d/(t_2 - t_0)$. Similarly, the instantaneous speeds can be estimated for every time point after that, and becomes $\Delta d/(t_k - t_{k-2})$ for the current moment t_k. Note for ease of display, antenna 2 is plotted upside down.

III. Moving Distance Estimation

Precise estimation of the moving distance of a device has been a significant bottleneck in indoor tracking and inertial sensing. Previously, IMUs have been widely exploited in the way of multiplying step count by stride lengths. This category of approaches, however, are well known to yield huge errors since stride lengths are difficult to estimate and vary significantly over time and subjects. In this section, we introduce two different approaches that can estimate the incremental displacement of a moving device by leveraging rich multipaths indoors.

A. RF-based Moving Distance Estimation

(1) Estimating Distance by Virtual Antenna Alignment. Virtual Antenna Alignment (AA) was first introduced in [52] for RF-based Inertial Measurement (RIM), which estimates the moving distance, heading direction, and rotating angle. Original RIM resolves all these parameters using more antennas than what is available on most commodity devices. In EasiTrack, we employ RIM for moving distance estimation only and implement it in a special case with only two or three antennas on one single WiFi chip.

Here we briefly present the core idea of AA-based distance estimation. Consider that a two-antenna array is moving along the line joining themselves (we term the line as the *antenna array line*). As shown in Fig. 2, when the radio moves, one antenna will immediately follow the trajectory of the other. In other words, the two antennas travel through the same spatial locations and thus observe similar (ideally identical) CSI series, one after the other, with certain time delays Δt. Evidently, during Δt, the array has traveled for a distance of Δd, which equals to the antenna separation. Note that Δd is fixed and known for a given array. Thus if we can estimate the precise time delay Δt, we will obtain the accurate speed estimate as $v = \Delta d / \Delta t$. And by continuously estimating Δt, the real-time speeds along the whole trajectory could be derived. Then the moving distance is directly calculated as $d = \int_0^T v_t dt$, where T is the total traveling time.

The above intuition leads to a robust mechanism, named antenna alignment, for moving distance estimation in RIM [52]. The key is to determine the accurate time delay Δt, namely, the time difference when an antenna arrives at a location traversed by the other antenna (i.e., the two antennas are "aligned"). Observe that, due to rich multipaths indoors, the CSI measured at one location could serve as a distinct location profile. Then the task is equivalent to identifying the precise time difference when one antenna observes the most similar channel measurements with the other. Towards this end, RIM employs the Time-Reversal Resonating Strength (TRRS) [8], [54] as the similarity metric for CSI.

The TRRS for two CSI H_1 and H_2 is calculated as follows:

$$\eta(H_1, H_2) = \frac{|H_1^{\mathrm{H}} H_2|^2}{\langle H_1, H_1 \rangle \langle H_2, H_2 \rangle}, \tag{1}$$

where $(\cdot)^{\mathrm{H}}$ denotes the conjugate transpose. The above TRRS has been demonstrated to be robust and sensitive, yielding super-resolution at the sub-centimeter level for both LOS and NLOS conditions.

To make it more discriminative and robust, we exploit spatial diversity from multiple transmit antennas. Since we only focus on moving distance using a linear array, we enhance the TRRS calculation by combining the different pairs of receive antennas, if any. Given our case of a linear array of three antennas (numbered as 1, 2, and 3), we combine the TRRS of the first two antennas (*i.e.*, 1 and 2) and that of the second two antennas (*i.e.*, 2 and 3) since they have the same Δd and thus expect the same delay time. Taking Fig. 3b as an example, the upper matrix will be averaged with the middle matrix to augment the aligned peaks.

To pinpoint the precise time delay when two antennas i and j are spatially aligned, we perform a local mapping between the CSI snapshot of antenna i at time t, denoted as $H_i(t)$, against those of antenna j measured within a time window of $2l$, denoted as $[H_j(t-l), \cdots, H_j(t), \cdots, H_j(t+l)]$. Fig. 3(a) shows an example of the TRRS trend $[\eta(H_i(t), H_j(t+k)), k = -l, \cdots, l]$. As seen, one can identify the time delay by looking for the maximum TRRS peak. Formally, we have

$$\Delta t = |\underset{k \in \{-l, \cdots, l\}}{\arg\max}\ \bar{\eta}(H_i(t), H_j(t+k))|. \tag{2}$$

While dynamic programming is employed in [52] for peak finding, our measurements show that, by combining the TRRS of two pairs of antennas, we can confidently find the peaks by simply looking up the maximum values. In practice, we further apply a regression around the TRRS peak area to find a finer-grained time lag, which will break down the resolution limit by sampling rates, as shown in Fig. 3a.

Although AA involves CSI mapping, it does not need any prior training and is immune to environmental dynamics because the mapping is done in a transient window. It is also demonstrated in [52] to tolerate a deviation of 15°, *i.e.*, the moving direction slightly deviates from the antenna array line. We also experimentally validate the results, as shown in Fig. 3c. This is a crucial feature for handling realistic movements without requiring the target to move strictly along the antenna array line, underpinning practical tracking of targets like robots, AGVs, shopping carts, etc.

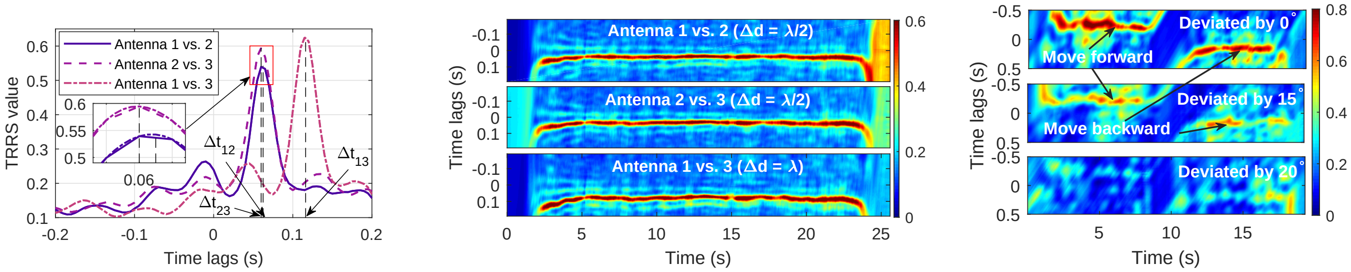
(2) Estimating Distance by Focusing Ball. AA, however, is unsuitable for human tracking since it is impractical to ask a person to carry a device and move it along a line during natural walking. To estimate moving distance for pedestrians, we incorporate a recently developed method based on time-reversal focusing ball effect in [65], where a critical statistical property of indoor RF signal propagation is discovered and modeled: The spatial distribution of multipath profiles, represented by CSI, is subject to a determinate function of spatial separation. It underlies an opportunity to reversely infer the moving distance from the distribution of CSI, whenever the device moves. We briefly review the main results below and refer readers for more details in [65].

Consider the TRRS in (1) for two CSI measured by the same receive antenna i at two locations L_0 and L, denoted as $H_i(t; L_0)$ and $H_i(t; L)$. We have [65]:

$$\eta(H_i(t; L_0), H_i(t; L)) \approx J_0^2(\frac{2\pi}{\lambda} d), \tag{3}$$

where λ is the wavelength, d is the distance between L_0 and L, and $J_0(x)$ is the zeroth-order Bessel function of the first kind. From (3), one can estimate the moving distance by calculating the TRRS distribution for one CSI and the subsequent measurements on the receiver, as detailed in [65].

Comparing with AA, the accuracy of FB for distance estimation will be lower because the TRRS measurements may slightly deviate the theoretical distribution in practice due to the non-uniformity of multipath distribution. Yet FB method is superior in its independence of moving directions and locations since the property arises from the nature of numerous indoor multipaths. As a result, it is favorable for human tracking and employed in EasiTrack for this purpose. A key feature of both approaches is that, different from prior systems that attempt to *resolve and combat multipaths* for tracking [19], [58], EasiTrack distinctively *leverages numerous multipaths* together: The more multipaths and the more uniform they are, the better performance it can generally achieve [52], [65]. As



(a) TRRS

(b) TRRS matrix

(c) TRRS for deviated moving

Fig. 3: Estimating time delays from TRRS matrix. (a) The maximum TRRSs only appear at a single time point when one antenna presents at a location where the other antenna was Δt time ago, with super resolution. (b) The TRRS matrix for a straight moving trajectory of a three-antenna array. (c) The TRRS decreases if the device does not move along the antenna array line, yet evident peaks still exist for deviation angles within 15°, thus allowing speed estimation.

a result, we can estimate distances independently from many factors, including location, orientation, environment, etc.

B. Orientation Estimation

In addition to moving distance, EasiTrack also needs the moving direction to track a target continuously. EasiTrack utilizes the ubiquitous inertial sensors and follows standard operations to derive orientation information [61]. In particular, we mainly employ the gyroscope to measure the turning angles and accumulate the measurements to infer the current heading direction. As we aim at 2D movements, we project the gyroscope readings based on the device's local coordinate system onto the gravity direction, which can be inferred from the accelerometer before the device starts moving. By such, the device can be in any position and does not need to be placed horizontally.

Inertial sensors are also leveraged to detect movements, and further determine if the target is making a turn or not, which will be used later for tracking.

IV. Map-Augmented Probabilistic Tracking

Intuitively, the locations can be directly calculated by integrating the consecutive measurements of moving distance and orientation, in a similar manner as dead-reckoning. This approach, however, suffers from significant errors due to the erroneous orientation measured by inertial sensors.

In EasiTrack, we propose to incorporate indoor maps to achieve precise tracking with coarse-grained orientation and distance observations. Two opportunities inspire our design: First, indoor maps impose useful geometric constraints to the target's movements. Second, digital maps are nowadays ubiquitously available. In particular, industry efforts have been carried out to calibrate indoor maps for public places [2]. Research advances have also promoted the automatic generation of digital maps via mobile crowdsourcing [10], [15].

We are not the first to use maps in tracking. Google Maps rely on road maps to correct GPS errors. Similar ideas have also been explored indoors [22], [33], [43]. However, we leverage indoor maps in a lightweight and scalable way. Specifically, we devise a graph-based particle filter based on a graph-based representation of an indoor map. We first present the graphical representation before diving deep into the proposed GPF in the following.

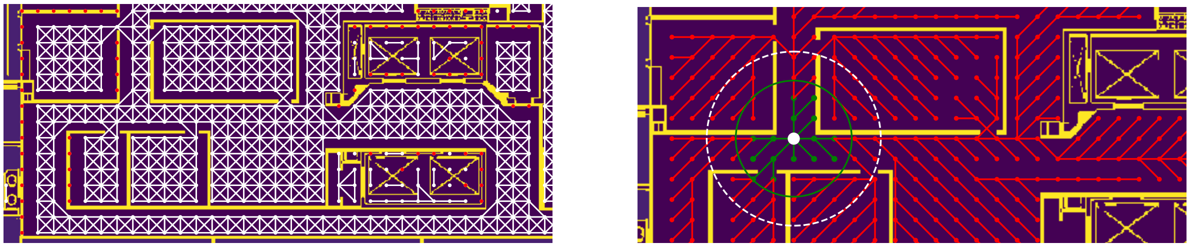
A. Graph Representation of Map

We take a standard indoor map, in the form of a plain image, as input. We do not request structured information, e.g., the knowledge of walls, corridors, and rooms, which is difficult to obtain from the map. Instead, we merely extract access information that tells whether a location (a pixel) is accessible or not. Hence, the map can be, for example, a simple image with a specific color indicating inaccessible pixels (e.g., black walls), the most typical form for indoor maps.

Map discretization: To reduce the computation complexity and achieve real-time tracking, we transform the map into a weighted graph. We first sample the space (i.e., map) by a set of uniformly distributed points with a specific grid size s, each point becoming a vertex. Every vertex could be accessible or inaccessible, depending on its location on the map (i.e., a pixel in the image). Then we add edges between a vertex and each of its adjacent vertexes if the edge does not intersect any obstacles. Each edge is weighted by its physical distance between two vertexes (which is either s or $\sqrt{2}s$ under grid sampling). By doing such, we build a weighted graph $G = (V, E)$, as shown in Fig. 4a. The sampling density s trades off the grid resolution and the number of vertexes. As we target at sub-meter accuracy, s should be below 1 m and adapts to different maps.

Reachability matrix: Once we have transformed a map into a weighted graph, we extract the intrinsic geometric properties by generating a *reachability matrix* M to represent the pair-wise *reachable distances* of $|V|$ vertexes. Each element m_{ij} of the matrix denotes the reachable distance between vertexes V_i and V_j, which is defined as their shortest path distance. m_{ij} will become an infinite value if there does not exist an unobstructed path between V_i and V_j. Fig. 4b illustrates an example of reachability for one vertex, which basically shows all the shortest paths starting from it to all other vertexes.

Note that the reachable distance is different from the Euclidean distance, but rather the walking distance between two locations. In fact, the reachable distance is usually larger than



(a) A graph representation of an indoor map.

(b) Reachability expressed by shortest paths.

Fig. 4: Graph-based map representation. For clarity, a grid size of 0.7 m is used and only partial map is displayed.

the Euclidean distance, because the straight line joining two locations is frequently blocked indoors. To avoid too huge size of M in case of large tracking areas, in practice we ignore too distant vertex pairs and make M a rather sparse matrix. For example, only elements $m_{ij} \leq 10$ m will be stored.

In the following section, we present how to design and implement an augmented particle filter by leveraging the graphical map with the above-pre-processed results.

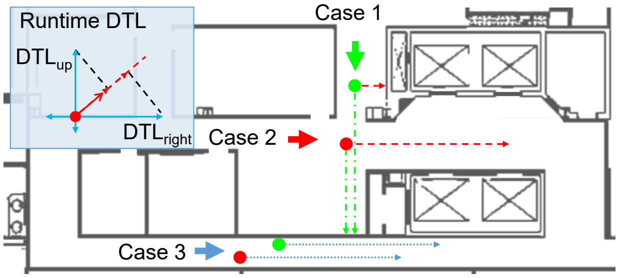


Fig. 5: DTL examples. In case 1, the particle in green has a larger DTL than the one in red. In case 2, when the particles are moving towards the right, the red one gets a larger weight. In case 3, both particles hold very large DTLs, which are thus forced to a fixed max-DTL value. In runtime, the DTL is approximated from the pre-calculated DTLs along four primary directions.

B. Tracking by Graph-based Particle Filter

We employ N particles, $\mathbf{X} = (\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(N)})$, with a three dimensional joint probability distribution. Each particle is represented as $\mathbf{X}^{(i)} = (x^{(i)}, y^{(i)}, \theta^{(i)})$, where $(x^{(i)}, y^{(i)})$ denotes its 2D location and $\theta^{(i)}$ is the orientation of the ith particle.

Particle Movement: For the motion measurement $(\Delta d_t, \theta_t)$ at time t, the ith particle is updated as

$$\begin{aligned} x_t^{(i)} &= x_{t-1}^{(i)} + (\Delta d_t + \delta^{(i)}) \cos(\theta_t + \alpha^{(i)}), \\ y_t^{(i)} &= y_{t-1}^{(i)} + (\Delta d_t + \delta^{(i)}) \sin(\theta_t + \alpha^{(i)}) \end{aligned} \tag{4}$$

where $\delta^{(i)}$ and $\alpha^{(i)}$ are the Gaussian random noises added to the moving distance and orientation measurements, respectively.

Particle Weighting: Initially, each particle gets an equal weight of $w_0^{(i)} = 1/N$. At every step, particles undergo two weighting assignment: 1) Any particle that hits any inaccessible area (typically the walls) during a move will "die" and gets eliminated; 2) All other particles survive and will be weighted based on the *distance-to-live* (DTL) defined as below.

The key idea to determine a particle's weight, without the help of any additional measurements from extra signal sources, is that a particle should get a larger weight if it is more likely to survive for a longer distance before it is going to hit some obstacles, and otherwise a smaller weight. This insight directly leads to the specification of DTL. As shown in Fig. 5, the DTL for a particle is calculated as the maximum accessible distance from the particle position along the particle moving direction. In the case of very long corridors, we set a maximum DTL to avoid overlarge values (Case 3 in Fig. 5).

In principle, the DTL needs to be calculated for every particle at every step, which is, however, too costly. In the proposed GPF, we devise an efficient approximation. During the pre-processing stage (§IV-A), we additionally calculate the DTLs in four basic directions (i.e., left, right, up, and down) in advance. Then the DTL along a certain direction θ can be accordingly calculated as the sum of the projection of the DTLs along the two closest basic directions (e.g., right and up for the example in Fig. 5) on the current moving direction. Albeit the resulted DTLs are not necessarily equal to the exact DTLs by definition, we demonstrate in real system that such approximation is effective and efficient. The main reason behind this is that only the relative ranking of particle weights, rather than the absolute values, matter in the GPF. The above operation retains similar weight ranking as the exact DTLs, yet significantly reduces the runtime complexity.

Resampling: Our GPF implements a novel approach for resampling. It first resamples N_{live} particles from the importance distribution, interpreted by particle weights $\{w_t^{(i)} | i = 1, 2, \cdots, N_{live}\}$, by following classical sequential importance resampling approach [4]. Here N_{live} denotes the number of surviving particles during the current step. However, for each dead (thus eliminated) particle, we choose to regenerate, instead of resampling, a new particle to avoid sample impoverishment problem (particles being over-concentrated) [23].

As shown in Fig. 4b, we consider a certain neighboring area centered at the target's current location for regeneration. Supposing the target is currently at (x_t, y_t) closest to vertex V_t, we first gather all of its reachable vertexes V_j with a reachable distance $m_{tj} \leq r$ and calculate each of their DTLs based on the current moving direction. Then we perform importance sampling to draw $N - N_{live}$ new particles among these candidate locations, using their DTLs as importance distribution. As shown in Fig. 4b, the regeneration radius r dynamically increases (white circle) when the target is detected to be turning and decreases otherwise (green circle).

Target location estimation: Finally, at each step, we estimate the target's location using information of all the particles. Intuitively, the target location can be determined as the weighted centroid of all particles' locations. However, the centroid may be an inaccessible location, or the line joining it to the current location may intersect a wall. Therefore, in EasiTrack, we resort to a similar concept of *medoid*. In general, a medoid is a representative within a data set whose average dissimilarity to all the objects in the set is minimal. Compared with centroid, using the medoid ensures that the resulted target location is always valid. Formally, the target location is estimated as the location of particle $\mathbf{X}_t^{(p^\star)}$, where

$$p^\star = \underset{i \in \{1, \cdots, N_{live}\}}{\arg\min} \sum_{j=1}^{N_{live}} \frac{\phi(\mathbf{X}_t^{(i)}, \mathbf{X}_t^{(j)})}{w_t^{(j)}}, \tag{5}$$

where $\phi(\mathbf{X}_t^{(i)}, \mathbf{X}_t^{(j)})$ denotes the Euclidean distance between the two particles' locations. Then the series of location estimates are smoothed and displayed to users.

C. Combating Accumulative Orientation Errors

Gyroscope is known to suffer from significant accumulative errors. As reported by the latest work [37], the error can accumulate to above 50° after 2 minutes of running. According to our experimental experience, it could produce over 30° errors for a single 90° natural turning. These considerable errors in orientation, if not calibrated, will lead to significant location errors that even the GPF fails to correct because all particles will be guided to move in the wrong directions.

To eliminate the accumulative errors, we devise a technique to reset the orientation opportunistically. The key observation is that, when the target is moving along a roughly straight path but not making a turn, the moving trajectory during this non-turning period offers a useful hint to infer the current heading direction. EasiTrack leverages these hints and performs opportunistic moving direction recalibration. Specifically, once such a non-turning segment is detected, we reset the current moving direction by estimating the centerline direction of the straight segment. For common behaviors, the target is moving straight rather than turning most of the time. Thus we can employ the reset operation quite frequently. And by continuous recalibration, the accumulative direction errors will be effectively mitigated, even over very long running.

D. Obtaining Initial Location

As EasiTrack only measures moving distances and directions, it needs a global location to initiate. The particle filter is capable of determining the initial location by initializing a set of particles uniformly distributed over all possible locations. With a sufficient number of random particles over the area of interests, the algorithm will progressively converge after a certain time. The initial location will be locked backpropagation from the converged location. The traveled path can be retraced, as well. Once the initial location is found, the system will reduce to a small number of particles for efficiency. Considering a real application and system, a practical alternative is to leverage the last-seen GPS signals at building entrances, as well as to request manual inputs from end-users. We mainly experiment with manual inputs in current EasiTrack system since the primary goal is to demonstrate our distance estimation and map enhancement techniques. To prove the concept, we implement an anchor as such by using an AP with a directional antenna. Although these technologies may not offer initial location at high accuracy, the caused errors will later be overcome by the proposed GPF, as will be demonstrated in §VI. Therefore, the initial locations could be gathered by various methods, such as occasional GPS indoors [11], potential indoor distinct landmarks [43], ubiquitous BLE beacons [25], and other conventional localization schemes [61]. Compared to using particle filtering, initial locations provided by external information generally offer a shorter time-to-first-fix.

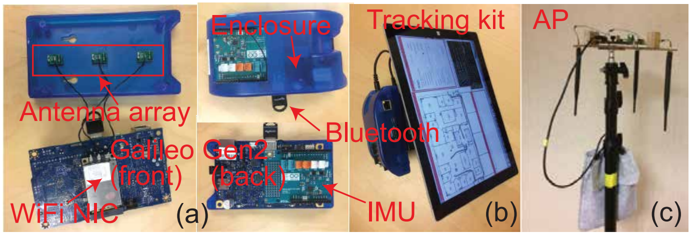


Fig. 6: EasiTrack hardware.

V. Implementation

We build a fully functional real-time system of EasiTrack, which involves many implementation details. We briefly mention some of them here.

Hardware: We implement EasiTrack using commercial WiFi NICs. To build a mobile client (named an *EasiTracker*), we use an Intel Galileo Gen2 microcontroller board, which is attached with a Qualcomm Atheros 9k series chipset and a Bosch Sensortec BNO055 IMU. Fig. 6 illustrates our hardware equipment. The board runs Linux OS, and we modify the driver to report CSI. The WiFi chipset comes with three antennas, which are spaced linearly at a default separation of 4 cm (to fit the enclosure of the Galileo board) for AA method. The antennas can be arbitrarily arranged for FB method. We run another board with the same WiFi card as an AP, which is configured to the broadcast mode on a 5 GHz channel. The clients use chip antennas for consideration of compact form factor, while the AP is equipped with standard line antennas to provide better coverage. Although we use Galileo board for experiments, EasiTrack is ready to run on commodity smartphones once the CSI becomes accessible on them.

Software: We first prototyped all algorithms in MATLAB and Python. The codes are later all translated into C++ to build the real system. We develop a tool in C++ running on Galileo Gen2 to collect CSI and sensor data, which are sent back to a host machine via Bluetooth. Currently, we use a Surface Pro as the host to run the tracking system and display the results with a GUI (Fig. 6(b)), all implemented in C++.

CSI denoising: CSI measured on COTS WiFi contains phase offsets, such as carrier frequency offset (CFO), sampling frequency offset (SFO), and symbol timing offset (STO), in addition to initial phase offset caused by the phase-locked loops. EasiTrack is immune to the annoying initial phase distortions, which can only be manually calibrated previously [58], by taking the absolute value in Eqn. 1. We calibrate the other linear offsets by using the phase cleaning approach in [19].

Integrating multiple receive antennas: The proposed AA method requires a minimum of two antennas as an array. In our implementation, the WiFi chipsets have three antennas. To fully utilize them, we arrange them in a uniformly spaced linear array. Then we integrate the results of two pairs: the 1st and 2nd antennas as one pair and the 2nd and the 3rd as the other. The FB method works with a single antenna, yet we also fuse the TRRS of all receive antennas to facilitate distance estimation.

Movement detection: We employ inertial sensors to determine whether the target is moving. Only when the device is detected to move will EasiTrack estimate the moving distance. To mitigate the interference of irrelevant movement (e.g., hand motions) in human tracking, we additionally apply a mask to detect if a user is walking or not by identifying steps using IMU [43], [61].

Tracking in very large areas: EasiTrack uses a single AP to achieve ubiquitous tracking with remarkable coverage. Technically, it works in any place covered by the AP, regardless of LOS or NLOS conditions. In practice, one single AP may not be sufficient to cover a very large tracking area fully. In such a situation, multiple APs are required to provide full coverage. Accordingly, EasiTrack needs to handle handover between different APs when the client roams about the space, although it always uses only one AP at one time for tracking. We use a straight-forward RSS-based handover scheme in our implementation for this purpose.

Map pre-processing: EasiTrack needs merely information on accessible and inaccessible areas from the ordinary images of indoor floorplans. This information could be automatically recognized from pixel colors. For example, usually, the walls are marked in darker colors while open spaces are blank and white (or vice versa). However, depending on the quality of the obtained map image, there might be some interferences (e.g., texts, dimensions, etc.) that affect automatic recognition. To cope with that, we currently need to manually pre-process the map to highlight all the obstacles (mainly walls) with a specific color. According to our experience, it takes about 10 minutes for one who knows the basic operations of Photoshop to process a normal map.

VI. Evaluation

We evaluate our system in real-world scenarios using COTS WiFi. We then deploy EasiTrack in 5 different buildings to evaluate human and cart tracking and two industry manufacturing facilities to test AGV tracking.

A. Performance on Distance Estimation

Methods: We first evaluate the accuracy and impacting factors of the proposed distance estimation method, AA, and compare it with FB. To do so, we put the EasiTracker on a cart and move it roughly along a straight line. We evaluate different traces of about 10 m long and perform both AA and FB on the collected CSI, respectively. We set up a camera to capture the ground truth moving distances.

Overall accuracy: Fig. 7 compares the overall performance of AA and FB on moving distance estimation. As seen, AA achieves a remarkable median error of about 0.25 m and 90%tile error of 0.69 m, while FB yields 0.75 m and 1.82 m errors, respectively. Although the performance gains of AA come at the cost of relatively constrained movements, we believe AA opens up a wide range of exciting opportunities due to the high accuracy, especially for tracking machines and objects like AGVs, shopping carts, robots, etc. In the following, we study several factors that may impact AA's performance. The impacting factors of the FB method have been extensively studied in [65]. Thus we only evaluate the overall tracking performance in the next section when incorporating it in EasiTrack.

Sampling rate: Certain sampling rates will be needed by EasiTrack to ensure enough resolution in distance estimation; otherwise, the peak resolution (recall Fig. 3a) is limited. Fig. 8 shows the impacts when downsampling the CSI data from 200 Hz to 20 Hz. The results show that a sampling rate of 200 Hz, which we use in EasiTrack, is adequate for normal speeds of about 2 m/s.

Antenna diversity and separation: Fig. 9 shows the diversity of antenna pairs. We test different combinations of the three antennas available on our device as well as fuse two pairs together. As seen, AA produces consistently high performance using different antenna pairs, which is further improved when combining multiple pairs. We also examine different antenna separations, ranging from $\lambda/2, \lambda, 3\lambda/2$ to 2λ. As shown in Fig. 10, different separations only see negligible differences. Yet, as a rule of thumb, the separation should be larger than $\lambda/2$; otherwise, coupling effects will come into play.

B. Benchmark Performance

Methods: Now we study the tracking performance of EasiTrack system. We deploy the system on one floor of a typical office building, as shown in Fig. 11. There are rooms separated by dry walls, concrete pillars, and elevators inside the floor. We place the AP in the middle of the building to provide good coverage. The AP works on channel 153. There are regular WiFi traffics on the same and adjacent channels. During experiments, people are working around as usual.

For AA-based EasiTrack (EasiTrack-AA), we place the device on a cart and push it around. While for FB-based (EasiTrack-FB), we ask a human to hold the device in hand and walk naturally. To study the quantitative location errors, we mark a set of checkpoints at a density of about every two meters. The ground truth is recorded when a user passes by a checkpoint. Then we employ several users to walk (or push a cart) around different areas. Multiple users can be tracked at the same time, no matter whether they are testing EasiTrack-AA or EasiTrack-FB. Due to budget constraints, we build three tracking kits (each consisting of a Galileo Gen2 board and a

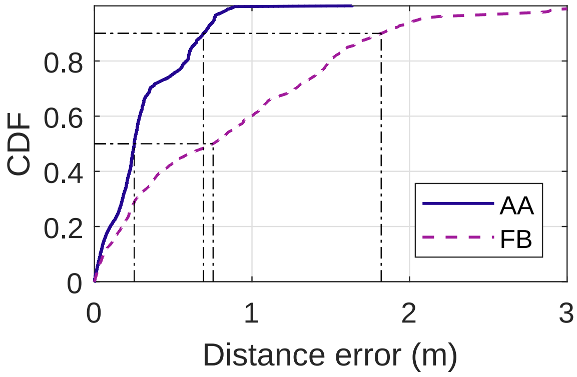


Fig. 7: Comparison of AA and FB

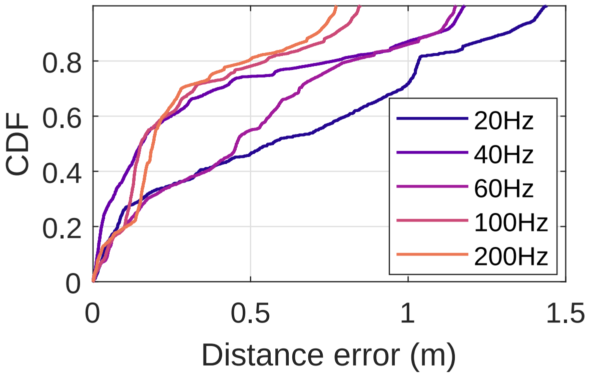


Fig. 8: AA wrt. sampling rate

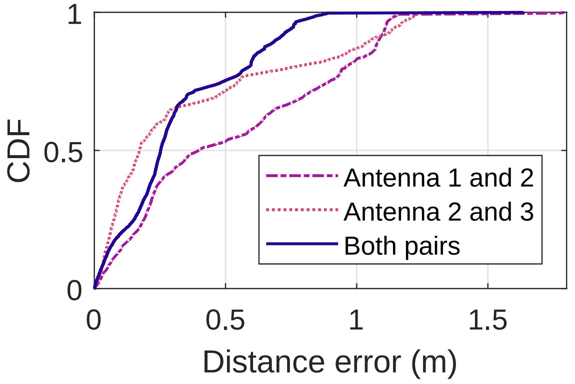


Fig. 9: AA wrt. different pairs

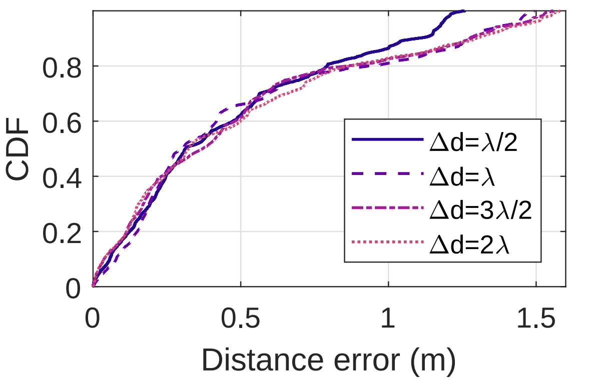


Fig. 10: AA wrt. antenna separations

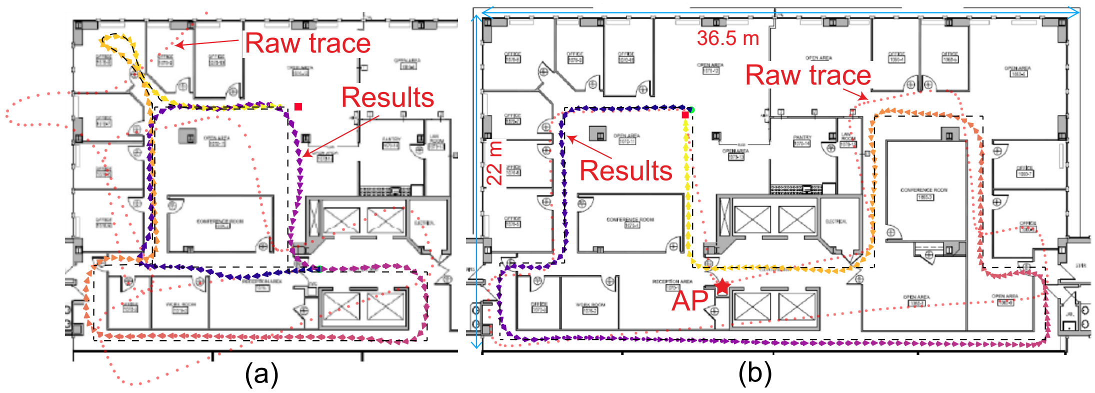


Fig. 11: Benchmark floor and example tracking traces. (a) Human tracking with EasiTrack-FB. (b) Cart tracking with EasiTrack-AA. True traces are marked in dashed black lines.

Surface Pro), although EasiTrack can support any number of tracking kits. The results are then analyzed to evaluate the two methods, respectively.

Overall performance: Fig. 11 shows two examples of the tracking results[2]. Fig. 12 shows that both EasiTrack-AA and EasiTrack-FB achieve sub-meter median accuracy. Specifically, EasiTrack-AA yields a median 0.58 m and 90%tile 1.33 m error, respectively, for cart tracking. Surprisingly, EasiTrack-FB achieves only slightly worse performance for human tracking with a median error of about 0.70 m, and a 90%tile error of 1.97 m. Compared to the distance estimation accuracy in Fig. 7, the overall tracking accuracy of EasiTrack-AA is slightly lower, which is mainly limited by orientation errors arising from sensors. In contrast, the accuracy of EasiTrack-FB improves a lot, contributed by the proposed GPF, which effectively corrects location errors with a map.

Impacts of particle number: A unique feature of EasiTrack is that it achieves excellent performance using a small number of particles thanks to the graph-based model. We investigate how the performance would change concerning the amount of particles, ranging from 20 to 800. As shown in Fig. 13, EasiTrack achieves considerable performance with only 100

[2] Real-time videos and data about more tracking results will be available upon publication.

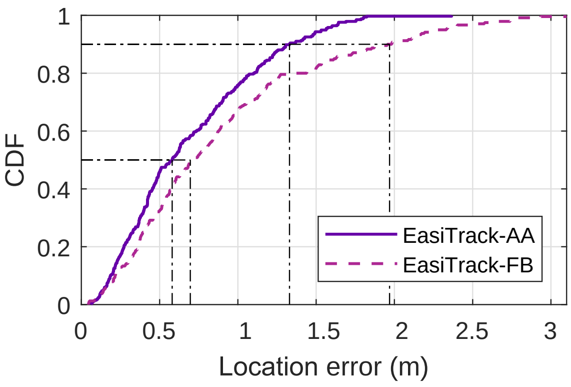


Fig. 12: Overall tracking accuracy

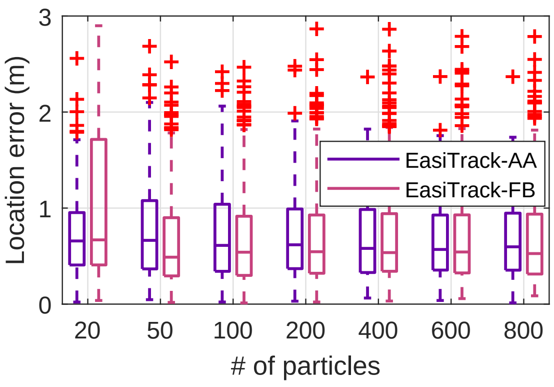


Fig. 13: Impacts of particle amounts

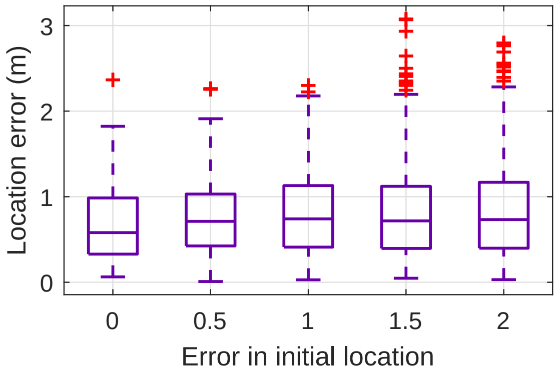


Fig. 14: Impacts of initial location errors

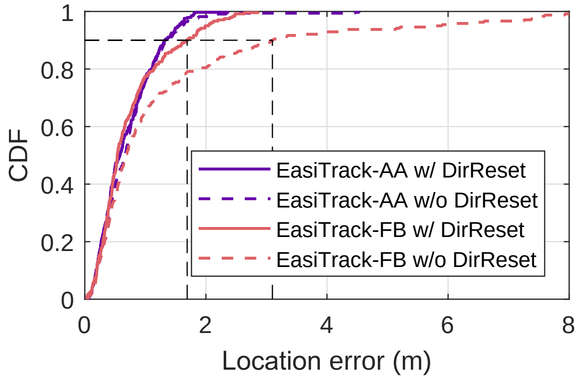


Fig. 15: Benefits of direction reset

particles. The results show that EasiTrack requires orders of magnitude less particles than existing approaches [22], [33]. However, 20 particles are too few and may lead to large tail errors, especially for EasiTrack-FB. In practice, we recommend a minimum of 50 particles.

Impact of initial locations: When using manual inputs for the initial location, it is interesting to examine how sensitive it is to the uncertainties in initial location. We add random errors to the start point, from 0.5 m to 2 m, and evaluate the traces with the erroneous initial locations. As shown Fig. 14, the proposed GPF overcomes these errors and maintains similar performance, which only slightly degrades to a median 0.73 m and 90%tile 1.56 m error for EasiTrack-AA when there are 2 m errors in initial locations. In other words, EasiTrack does not rely on precise initial locations to start. Instead, it only needs a coarse-grained input, which again could be provided by the user, opportunistic GPS, or other available anchors.

In the case of no available initial location, EasiTrack employs random particles to search for it. The system is demonstrated to fix a starting point by using 1000 particles successfully. In our experiments, EasiTrack will converge to the correct location after about 20 to 40 seconds of walking (corresponding to about 25 to 50 m in our case). More practically, the system fixes the first location after about 4 turns. Taking the trace in Fig. 11 as an example, the system converges after passing through the left-bottom corner. In practice, however, the delay to first fix depends on the floor layout and how a user walks. Hence we believe the results promise continuous tracking for applications insensitive to initialization delays.

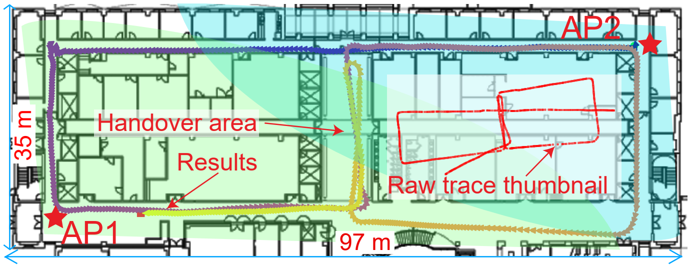
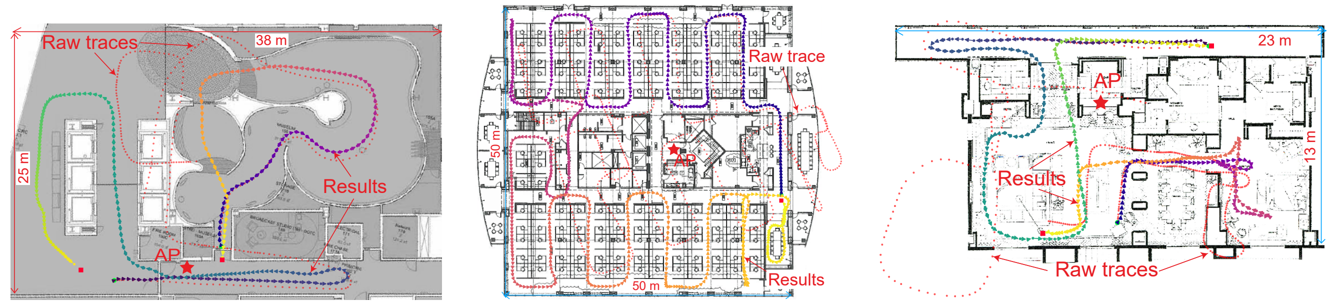


Fig. 16: Tracking with handover in large areas

Benefits of direction reset: We individually study the benefits of the direction reset module. To do so, we rerun all the traces offline by disabling the direction reset feature, and compare the results in Fig. 15. As shown, large errors are remarkably reduced for EasiTrack-FB, with a 90%tile error of 3.1 m to 1.7 m, by automatic direction reset, which eliminates large accumulative direction errors caused by sensors. For EasiTrack-AA, it achieves marginal accuracy gains. The reason is that direction error is less significant than human walking when placing the device on a cart. However, when direction errors accumulate, the direction reset will take effect.

Coverage: Our benchmark evaluation demonstrates EasiTrack's tracking coverage in a typical office of 36 m×22 m. To demonstrate the coverage limit, we test tracking in a large building of about 100 m×35 m with many concrete



(a) Technology museum **(b) Office building** **(c) Hotel suite**

Fig. 17: Human tracking in three different buildings.

walls and pillars, where one AP is not sufficient to cover the entire floor, and thereby, we can test tracking with handover between two APs. As shown in Fig. 16, we place two APs at two diagonal corners of the building, each roughly covers half of the tracking areas. The results validate that EasiTrack roams smoothly between the two APs and achieves similar high accuracy under either AP's coverage, as well as in the overlapping areas (middle corridors), thereby allowing it to track in vast areas by merely setting up additional APs.

C. *Real World Deployments*

To verify the tracking accuracy and deployment simplicity of EasiTrack, we carry out real-world deployment at multiple sites with two different scenarios considered: human tracking and AGV tracking. For these buildings, we are not permitted to set up checkpoints for ground truth. Thus we could only ask the testers to save real-time outputs of EasiTrack system. We then depict the results using gradient colors and compare them with the traveled paths[3]. Our on-site testing lasts for at least 2 hours to several days for different sites and involves at least two clients at each site. In the following, we randomly show one or two tracking traces, among many others, for each scenario, due to space limitations.

Tracking humans in buildings: To test EasiTrack-FB for human tracking, we deploy it in a technology museum, a hotel suite, and an office building. During tracking, there are people working routinely around.

Fig. 17 illustrates the tracking results at the three different sites. As shown, over all scenarios, with varying users of testing and completely different environments, EasiTrack-FB achieves consistently high accuracy and recovers the moving trajectory gracefully. In particular, in the museum (Fig. 17a), which is a non-regular indoor space, EasiTrack still tracks accurately, demonstrating its performance in the area with various layouts. In Fig. 17c, the user's movements around a bed inside a master room are also successfully tracked. It is worth noting that Fig. 17b demonstrates that EasiTrack can track the target accurately over an area of 50 m×50m, with a single AP placed in the center.

Tracking machines in factories: We have tested EasiTrack-AA for cart tracking in the benchmark building. We further test the performance for AGV tracking in two manufacturing facilities with different types of AGVs. Since we do not have access to the facilities, the systems are deployed and tested by our collaborators at their facilities. The results are shown in Fig. 18.

For the first facility (Fig. 18a), the AGV is rather big and moves slowly at around 0.5 m/s. The tester installed EasiTracker inside the AGV with a metal enclosure, which causes severe signal attenuation and makes the tracking very challenging. During testing, the AGV moves along a rectangular area for minutes to hours, and our system tracks its real-time location. Fig. 18a depicts the tracking traces for about 10 minutes. As seen from the raw traces without GPF, under the challenging deployment scenario, the distance estimation is not very accurate, yet still rather consistent over different rounds. Meanwhile, for this AGV, the direction errors accumulate considerably, making the raw traces drift quickly. Nevertheless, since the moving routes are relatively limited, EasiTrack still tracks the trajectories well, and error does not accumulate over time.

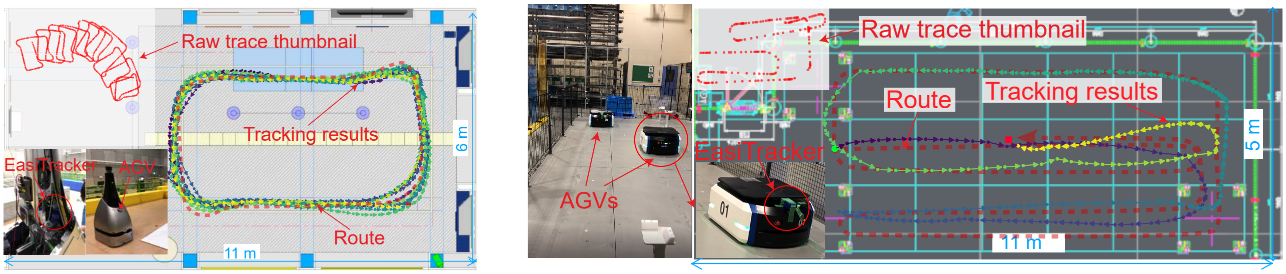
In Fig. 18b, the AGVs are smaller yet move faster at various speeds up to about 2.5 m/s, touching the upper limit that 200Hz sampling rate can support. Two AGVs were being tracked at the same time, with our EasiTracker attached to the front. The AGVs make in-place turns to change direction and move quite freely inside a rectangle area. Fig. 18b illustrates a typical moving trace with our system's tracking results. The results demonstrate that EasiTrack recovers the AGVs' trajectories, although errors occur in distance estimation due to the fast speeds (regarding our current sampling rate) and the harsh industrial environments.

Our current deployment in industrial facilities is limited to relatively small areas to prove the feasibility. Deploying and testing in more extensive areas is future work.

D. *System Latency*

EasiTrack runs in real-time without perceivable delays to end-users. It involves a constant delay due to the window length of l for calculating the TRRS matrix (Recall Fig. 3b). Consider $l = 20$, it introduces a delay by only 0.1s given a sampling rate of 200Hz. EasiTrack outputs a distance estimate

[3] For clarity of presentation, we only show the raw traces without the GPF and the final tracking results. The ground truths of the tracking routes are not displayed, yet can be easily inferred therein.



(a)

(b)

Fig. 18: AGV tracking in two facilities. In both cases, the AP is installed outside the shown areas.

for every CSI sample, *i.e.*, every 5 ms. To reduce computation, we will process a batch of distance estimates (*e.g.*, 10) and report location updates every 50 ms. The user in practice can decide the update rate.

We are interested in the system complexity introduced by different modules, especially distance estimation and GPF. We first measure the resource utilization on Surface Pro with an Intel Core i7 4650U and 8.0GB RAM. The system occupies about 5% of the CPU and about 10M of RAM when running without the GPF. The usage increases to about 8% of CPU and 180M of RAM when the GPF runs with 100 particles, and further to about 12% of CPU with similar RAM usage when using 400 particles in the GPF.

It is also interesting to see if the system can run on embedded devices. To do so, we port the system into Linux and run it on an embedded board equipped with dual-core Cortex-A7 @ up to 1.2 GHz. The results show that EasiTrack runs in real-time with 400 particles by utilizing about 35% of the CPU and about 80MB RAM (The memory usage is smaller than that on Surface Pro because we do not involve a user GUI for this embedded implementation).

VII. Discussions

Aiming at an accurate tracking system that is easy to deploy world wide, there is room for improvements for EasiTrack.

Map availability and quality: EasiTrack leverages digital floorplans by seeing the underlying trend that indoor maps are becoming more and more widely available. Driven by the massive demand in indoor positioning, the industry has been striving to gather and construct indoor maps for world-wide buildings - usually termed as indoor mapping. Examples include giant companies include Google, Apple, Baidu, and start-ups such as IndoorAtlas, Aruba Meridian, Jibestream, Mapwize, etc. Different from some traditional floorplan in plain images, these newly constructed maps contain detailed and comprehensive information, some of which are even interactive. It is promising that, with proper APIs, these maps could be directly inputted into EasiTrack without any pre-processing.

Map understanding: Currently, some manual efforts are required to pre-process a map (i.e., highlighting the inaccessible pixels). Since we only need accessibility properties of pixels, this could be done automatically with the help of sophisticated imaging processing techniques like edge extraction [29]. We also note that the pre-processing will not be needed for modern maps captured in structured formats with high quality [2], [5].

Map learning: Our graph model treats all vertexes (locations) equally. With data gathered from real-world deployment, the model could dynamically learn the probabilistic distributions of different locations. For example, people may tend to make a right turn at a certain location, while most go straight at another. By progressively learning this information, the tracking performance could be improved continuously. Also, the system could be extended to discover obstacle changes (e.g., newly appeared or moved furniture) by learning from the history tracking data.

Large open spaces: As demonstrated by our real-world deployments, EasiTrack generalizes to different buildings. However, it still needs improvement in the case of large open spaces (e.g., a great hall), where the map hardly offers any useful information.

VIII. Related Works

There are numerous research on indoor localization using WiFi [27], [42], [61]. We briefly review approaches based on fingerprinting, triangulation, and combination with inertial sensing. We also review the literature on indoor maps.

Fingerprinting: Fingerprint-based approaches collect signal measurements at different locations as wireless fingerprints (which could be RSSIs of multiple APs [6], [7], [26], [33], [39], [40], [43], [48]–[51], [62] or CSI measurements [8], [35], [54]), and then perform pattern matching to localize. The best systems report a median accuracy of 0.6 m [8], which, however, could significantly degrade to meters due to environmental dynamics [26] and temporal changes [50]. Centimeter accuracy is achieved by using CSI as fingerprints [8], [9], [54], which, however, need extensive training that prohibits its practical applications. Also, these systems are hard to deploy since they need an intensive site survey to bootstrap and potential recalibration due to changes.

Triangulation: Triangulation is widely exploited to localize targets based on range or angle estimates to multiple APs. Early systems attempt to infer ranges from signal strengths using a propagation model [11], [34], [53], which are limited to errors around 2-4 m at best. Recent efforts employing ToF [14], [31], [38], [41], [59] and AoA [13], [19]–[21], [57], [58] achieve state-of-the-art accuracy at the decimeter level.

However, these systems usually require many antennas (e.g., as high as 8 [58] while typical APs have three) or large bandwidths (e.g., [41] uses frequency hopping). Moreover, they are difficult to deploy also because the requirements of extremely precisely installed APs: small errors in AP location or orientation result in significant location errors to them.

Inertial sensing: Inertial sensing has been leveraged as standalone or integrated tracking systems [17], [31], [33], [34], [37], [43], [49], [61]. While being easy-to-deploy, inertial sensors suffer from accumulative errors. Recent innovations have promoted orientation sensing to a median orientation error around 10° [37], [68]. However, distance estimation for pedestrians using inertial sensors is challenging and still open [61]. As a result, prior works using inertial sensing yield errors at meters.

Recent innovation [52] enables RF-based inertial measurement at high precision, which underpins the foundation of accurate indoor tracking. WiBall [65] also derives moving distance from RF signals and yields location estimates by combining inertial sensors for direction and floorplans for correction. Compared with EasiTrack, however, WiBall is less accurate in distance estimation. More importantly, it applies rule-based correction and relies on detailed floorplan with rich structured knowledge about corridors, doors, crossings, walls, etc., which are costly to obtain. In contrast, EasiTrack employs probabilistic tracking using a plain image of the floorplan, rendering it robust and scalable.

Indoor maps and particle filtering: We further review research on indoor maps, which would be indispensable parts of indoor location systems to enable many applications, yet are less explored. Indoor maps are becoming increasingly available by dedicated industrial engineer efforts [2] and crowdsourcing [3], [10], [12], [15], [36], [67]. They can be utilized to ease the burden of fingerprint calibration via crowdsourcing [33], [43], [49] or to reduce tracking errors [16], [17], [22], [47]. Particle filtering has been widely used for this sensor fusion purpose [16], [22], [24], [33]. However, prior works usually implement it in a continuous map space, rely on a secondary measurement system (e.g., WiFi fingerprinting) for weight updating [16], [24], and typically require a few thousands of particles (e.g., 3000 in [22]). EasiTrack differs in its graph-based model that enables using less than 100 particles and particle weighting using map information only, without additional signals or constraints.

Other modalities: For completeness, we refer to modalities other than WiFi that are also exploited for indoor tracking, such as RFID [44], [46], [55], [60], ultrasound [30], [32], [66], visible light [28], [63], [64], [69], Bluetooth [18], [25], etc. However, none of these techniques are as ubiquitous as commercial WiFi infrastructure.

IX. Conclusion

We present EasiTrack, an indoor location system that achieves decimeter-level accuracy with broad coverage of both LOS and NLOS areas and scales to massive buildings with almost zero cost and an unlimited number of users. It achieves these distinct and strong sides by using a single unknown AP, without knowing its location/orientation, making it a promising solution for ubiquitous indoor tracking in practice. We contribute an approach for CSI-based moving distance estimation and a map-augmented tracking algorithm. We implement a real-time system and deploy it in different buildings and facilities under real-world settings to track humans and machines. The results demonstrate that EasiTrack scales over various conditions with consistent sub-meter accuracy and outperforms prior works with many distinct and strong sides.

References

[1] Ieee approved draft standard for information technology– telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE P802.11-REVmc/D8.0, August 2016*, pages 1–3774, Jan 2016.

[2] Google Indoor Maps. https://www.google.com/maps/about/partners/indoormaps/, 2019.

[3] M. Alzantot and M. Youssef. Crowdinside: Automatic construction of indoor floorplans. In *Proceedings of ACM SigSpatial*, pages 99–108. ACM, 2012.

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[5] S. Azhar. Building information modeling (bim): Trends, benefits, risks, and challenges for the aec industry. *Leadership and management in engineering*, 11(3):241–252, 2011.

[6] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of ACM MobiCom*, pages 261–272. ACM, 2009.

[7] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings of IEEE INFOCOM*, 2000.

[8] C. Chen, Y. Chen, Y. Han, H. Lai, F. Zhang, and K. J. R. Liu. Achieving Centimeter-Accuracy Indoor Localization on WiFi Platforms: A Multi-Antenna Approach. *IEEE Internet of Things Journal*, 4(1):122–134, Feb. 2017.

[9] C. Chen, Y. Chen, Y. Han, H.-Q. Lai, and K. R. Liu. Achieving centimeter-accuracy indoor localization on wifi platforms: A frequency hopping approach. *IEEE Internet of Things Journal*, 4(1):111–121, 2016.

[10] S. Chen, M. Li, K. Ren, X. Fu, and C. Qiao. Rise of the indoor crowd: Reconstruction of building interior view via mobile crowdsourcing. In *Proceedings of ACM SenSys*, pages 59–71. ACM, 2015.

[11] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of ACM MobiCom*, pages 173–184. ACM, 2010.

[12] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *Proceedings of ACM MobiCom*, 2014.

[13] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson. Phaser: Enabling phased array signal processing on commodity wifi access points. In *Proceedings of ACM MobiCom*, pages 153–164. ACM, 2014.

[14] W. Gong and J. Liu. Sifi: Pushing the limit of time-based wifi localization using a single commodity access point. *Proceedings of ACM IMWUT*, 2(1):10, 2018.

[15] Y. He, J. Liang, and Y. Liu. Pervasive floorplan generation based on only inertial sensing: Feasibility, design, and implementation. *IEEE Journal on Selected Areas in Communications*, 35(5):1132–1140, 2017.

[16] J. Hightower and G. Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *International conference on ubiquitous computing*, pages 88–106. Springer, 2004.

[17] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach. Graph-based data fusion of pedometer and wifi measurements for mobile indoor positioning. In *Proceedings of ACM UbiComp*, pages 147–158. ACM, 2014.

[18] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui. Rssi based bluetooth low energy indoor positioning. In *Proceedings of IEEE IPIN*, pages 526–533. IEEE, 2014.

[19] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of ACM SIGCOMM*, 2015.

[20] M. Kotaru and S. Katti. Position tracking for virtual reality using commodity wifi. In *Proceedings of IEEE CVPR*, 2017.

[21] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of ACM MobiCom*, pages 483–494. ACM, 2014.

[22] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of ACM MobiCom*, pages 421–430. ACM, 2012.

[23] T. Li, S. Sun, T. P. Sattar, and J. M. Corchado. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with applications*, 41(8):3944–3954, 2014.

[24] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz. Voronoi tracking: Location estimation using sparse and noisy sensor data. In *Proceedings of IEEE/RSJ IROS*, volume 1, pages 723–728. IEEE, 2003.

[25] X.-Y. Lin, T.-W. Ho, C.-C. Fang, Z.-S. Yen, B.-J. Yang, and F. Lai. A mobile indoor positioning system based on ibeacon technology. In *Proceedings of IEEE EMBC*, pages 4970–4973. IEEE, 2015.

[26] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the limit of wifi based localization for smartphones. In *Proceedings of ACM MobiCom*, pages 305–316. ACM, 2012.

[27] K. R. Liu and B. Wang. *Wireless AI: Wireless Sensing, Positioning, IoT, and Communications*. Cambridge University Press, 2019.

[28] S. Liu and T. He. Smartlight: Light-weight 3d indoor localization using a single led lamp. In *Proceedings of ACM SenSys*, page 11. ACM, 2017.

[29] R. Maini and H. Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.

[30] W. Mao, Z. Zhang, L. Qiu, J. He, Y. Cui, and S. Yun. Indoor follow me drone. In *Proceedings of ACM MobiSys*, pages 345–358. ACM, 2017.

[31] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim. Sail: Single access point-based indoor localization. In *Proceedings of ACM MobiSys*, pages 315–328. ACM, 2014.

[32] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of ACM MobiCom*, pages 32–43. ACM, 2000.

[33] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of ACM MobiCom*, pages 293–304. ACM, 2012.

[34] S. Sen, J. Lee, K.-H. Kim, and P. Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of ACM MobiSys*, pages 249–262. ACM, 2013.

[35] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the Mona Lisa: spot localization using PHY layer information. In *Proceedings of ACM MobiSys*, 2012.

[36] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: Indoor pathway mapping made easy. In *Proceedings of {USENIX} NSDI*, pages 85–98, 2013.

[37] S. Shen, M. Gowda, and R. R. Choudhury. Closing the gaps in inertial motion tracking. In *Proceedings of ACM MobiCom*, 2018.

[38] N. Tadayon, M. T. Rahman, S. Han, S. Valaee, and W. Yu. Decimeter ranging with channel state information. *arXiv preprint arXiv:1902.09652*, 2019.

[39] X. Tian, R. Shen, D. Liu, Y. Wen, and X. Wang. Performance analysis of rss fingerprinting based indoor localization. *IEEE Transactions on Mobile Computing*, 16(10):2847–2861, 2017.

[40] X. Tian, M. Wang, W. Li, B. Jiang, D. Xu, X. Wang, and J. Xu. Improve accuracy of fingerprinting localization with temporal correlation of the rss. *IEEE Transactions on Mobile Computing*, 17(1):113–126, 2018.

[41] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *Proceedings of USENIX NSDI*, 2016.

[42] B. Wang, Q. Xu, C. Chen, F. Zhang, and K. R. Liu. The promise of radio analytics: a future paradigm of wireless positioning, tracking, and sensing. *IEEE Signal Processing Magazine*, 35(3):59–80, 2018.

[43] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of ACM MobiSys*, pages 197–210. ACM, 2012.

[44] J. Wang and D. Katabi. Dude, where's my card?: Rfid positioning that works with multipath and non-line of sight. In *Proceedings of ACM SIGCOMM*, volume 43, pages 51–62. ACM, 2013.

[45] R. Want, W. Wang, and S. Chesnutt. Accurate indoor location for the iot. *Computer*, 51(8):66–70, 2018.

[46] T. Wei and X. Zhang. Gyro in the air: tracking 3d orientation of batteryless internet-of-things. In *Proceedings of ACM MobiCom*, pages 55–68. ACM, 2016.

[47] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of ACM UbiComp*, pages 114–123. ACM, 2008.

[48] C. Wu, J. Xu, Z. Yang, N. D. Lane, and Z. Yin. Gain without pain: Accurate wifi-based localization using fingerprint spatial gradient. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):29, 2017.

[49] C. Wu, Z. Yang, and Y. Liu. Smartphones based crowdsourcing for indoor localization. *IEEE Transactions on Mobile Computing*, 14(2):444–457, 2014.

[50] C. Wu, Z. Yang, and C. Xiao. Automatic radio map adaptation for indoor localization using smartphones. *IEEE Transactions on Mobile Computing*, 17(3):517–528, 2018.

[51] C. Wu, Z. Yang, Z. Zhou, Y. Liu, and M. Liu. Mitigating large errors in wifi-based indoor localization for smartphones. *IEEE Transactions on Vehicular Technology*, 66(7):6246–6257, 2016.

[52] C. Wu, F. Zhang, Y. Fan, and K. J. R. Liu. Rf-based inertial measurement. In *Proceedings of ACM SIGCOMM*, August 19-24 2019.

[53] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni. Csi-based indoor localization. *IEEE Transactions on Parallel and Distributed Systems*, 24(7):1300–1309, 2013.

[54] Z.-H. Wu, Y. Han, Y. Chen, and K. R. Liu. A time-reversal paradigm for indoor positioning system. *IEEE Transactions on Vehicular Technology*, 64(4):1331–1339, 2015.

[55] F. Xiao, Z. Wang, N. Ye, R. Wang, and X.-Y. Li. One more tag enables fine-grained rfid localization and tracking. *IEEE/ACM Transactions on Networking*, 26(1):161–174, 2018.

[56] Y. Xie, Z. Li, and M. Li. Precise power delay profiling with commodity wi-fi. *IEEE Transactions on Mobile Computing*, 2018.

[57] Y. Xie, Y. Zhang, J. C. Liando, and M. Li. Swan: Stitched wi-fi antennas. In *Proceedings of ACM MobiCom*, 2018.

[58] J. Xiong and K. Jamieson. ArrayTrack: a fine-grained indoor location system. In *Proceedings of USENIX NSDI*, 2013.

[59] J. Xiong, K. Sundaresan, and K. Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *Proceedings of ACM MobiCom*, pages 537–549. ACM, 2015.

[60] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices. In *Proceedings of ACM MobiCom*, pages 237–248. ACM, 2014.

[61] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu. Mobility increases localizability: A survey on wireless indoor localization using inertial sensors. *ACM Computing Surveys*, 47(3):54, 2015.

[62] M. Youssef and A. Agrawala. The horus wlan location determination system. In *Proceedings of ACM MobiSys*, pages 205–218. ACM, 2005.

[63] C. Zhang and X. Zhang. Litell: robust indoor localization using unmodified light fixtures. In *Proceedings of ACM MobiCom*, pages 230–242. ACM, 2016.

[64] C. Zhang and X. Zhang. Pulsar: Towards ubiquitous visible light localization. In *Proceedings of ACM MobiCom*, pages 208–221. ACM, 2017.

[65] F. Zhang, C. Chen, B. Wang, H.-Q. Lai, Y. Han, and K. R. Liu. Wiball: A time-reversal focusing ball method for decimeter-accuracy indoor tracking. *IEEE Internet of Things Journal*, 5(5):4031–4041, 2018.

[66] L. Zhang, K. Liu, Y. Jiang, X.-Y. Li, Y. Liu, P. Yang, and Z. Li. Montage: Combine frames with movement continuity for realtime multi-user tracking. *IEEE Transactions on Mobile Computing*, 16(4):1019–1031, 2017.

[67] B. Zhou, M. Elbadry, R. Gao, and F. Ye. Batmapper: Acoustic sensing based indoor floor plan construction using smartphones. In *Proceedings of ACM MobiSys*, pages 42–55. ACM, 2017.

[68] P. Zhou, M. Li, and G. Shen. Use it free: Instantly knowing your phone attitude. In *Proceedings of ACM MobiCom*, 2014.

[69] S. Zhu and X. Zhang. Enabling high-precision visible light localization in today's buildings. In *Proceedings of ACM MobiSys*, pages 96–108. ACM, 2017.

Chenshu Wu (M'15) received his B.E. degree in the School of Software in 2010 and Ph.D. degree in the Department of Computer Science in 2015, both from Tsinghua University, Beijing, China. He is now a Postdoc Research Associate in the Department of Electrical & Computer Engineering University of Maryland, College Park, and a Principal Scientist at Origin Wireless Inc. His research interests include Internet of Things, wireless sensing, and mobile computing. He is a member of the IEEE and the ACM.

Feng Zhang (M'18) received his B.S. and M.S. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, in 2011 and 2014, respectively. He received his Ph.D. degree from the Department of Electrical and Computer Engineering, University of Maryland, College Park in 2018. He is currently with Origin Wireless, Inc. His research interests include wireless sensing, statistical signal processing, and wireless indoor localization. He was the recipient of Distinguished TA award from the University of Maryland and the State Scholarship from the University of Science and Technology of China.

Beibei Wang (SM'15) received the B.S. degree in electrical engineering (with the highest honor) from the University of Science and Technology of China, Hefei, in 2004, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park in 2009. She was with the University of Maryland as a research associate in 2009-2010, and with Qualcomm Research and Development in 2010-2014. Since 2015, she has been with Origin Wireless Inc. where she is now a chief scientist. Her research interests include wireless communications and signal processing. Dr. Wang received the Graduate School Fellowship, the Future Faculty Fellowship, and the Dean's Doctoral Research Award from the University of Maryland, and the Overview Paper Award from IEEE Signal Processing Society in 2015. She is a co-author of Cognitive Radio Networking and Security: A Game-Theoretic View (Cambridge University Press, 2010).

K. J. Ray Liu (F'03) is a Distinguished University Professor of University of Maryland, College Park, in 2007, where he is Christine Kim Eminent Professor of Information Technology. He leads the Maryland Signals and Information Group conducting research encompassing broad areas of information and communications technology with recent focus on wireless AI.

Dr. Liu was a recipient of the 2016 IEEE Leon K. Kirchmayer Award on graduate teaching and mentoring, IEEE Signal Processing Society 2014 Society Award, IEEE Signal Processing Society 2009 Technical Achievement Award, and over a dozen of best paper awards. Recognized by Web of Science as a Highly Cited Researcher, he is a Fellow of IEEE and AAAS. His invention of Time-Reversal Machine by Origin Wireless Inc. won the 2017 CEATEC Grand Prix award.

Dr. Liu is IEEE Vice President, Technical Activities – Elect. He was President of IEEE Signal Processing Society, where he has served as Vice President – Publications and Board of Governor, and a member of IEEE Board of Director as Division IX Director. He has also served as the Editor-in-Chief of IEEE Signal Processing Magazine.

He also received teaching and research recognitions from University of Maryland including university-level Invention of the Year Award; and college-level Poole and Kent Senior Faculty Teaching Award, Outstanding Faculty Research Award, and Outstanding Faculty Service Award, all from A. James Clark School of Engineering.