Wi-Go: Accurate and Scalable Vehicle Positioning using WiFi Fine Timing Measurement

Mohamed Ibrahim
WINLAB, Rutgers
mibrahim@winlab.rutgers.edu

Ali Rostami
WINLAB, Rutgers
rostami@winlab.rutgers.edu

Bo Yu
General Motors Research
bo.3.yu@gm.com

Hansi Liu
WINLAB, Rutgers
hansiiii@winlab.rutgers.edu

Minitha Jawahar
WINLAB, Rutgers
minitha.jawahar@rutgers.edu

Viet Nguyen
WINLAB, Rutgers
vietnh@winlab.rutgers.edu

Marco Gruteser
WINLAB, Rutgers
gruteser@winlab.rutgers.edu

Fan Bai
General Motors Research
fan.bai@gm.com

Richard Howard
WINLAB, Rutgers
reh@winlab.rutgers.edu

ABSTRACT

Driver assistance and vehicular automation would greatly benefit from uninterrupted lane-level vehicle positioning, especially in challenging environments like metropolitan cities. In this paper, we explore whether the WiFi Fine Time Measurement (FTM) protocol, with its robust, accurate ranging capability, can complement current GPS and odometry systems to achieve lane-level positioning in urban canyons. We introduce Wi-Go, a system that simultaneously tracks vehicles and maps WiFi access point positions by coherently fusing WiFi FTMs, GPS, and vehicle odometry information together. Wi-Go also adaptively controls the FTM messaging rate from clients to prevent high bandwidth usage and congestion, while maximizing the tracking accuracy. Wi-Go achieves lane-level vehicle positioning (1.3 m median and 2.9 m 90-percentile error), an order of magnitude improvement over vehicle built-in GPS, through vehicle experiments in the urban canyons of Manhattan, New York City, as well as in suburban areas (0.8 m median and 3.2 m 90-percentile error).

CCS CONCEPTS

• **Networks → Location based services**.

ACM Reference Format:
Mohamed Ibrahim, Ali Rostami, Bo Yu, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Fan Bai, and Richard Howard. 2020. Wi-Go: Accurate and Scalable Vehicle Positioning using WiFi Fine Timing Measurement. In *The 18th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '20), June 15–19, 2020, Toronto, ON, Canada.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3386901.3388944

MobiSys '20, June 15–19, 2020, Toronto, ON, Canada

ACM ISBN 978-1-4503-7954-0/20/06…$15.00
https://doi.org/10.1145/3386901.3388944

1 INTRODUCTION

The rapid evolution of advanced driver assistance and vehicle automation systems, along with their growing market [1, 2], have led to increased demand for lane-level vehicle positioning that is accurate even in urban canyon environments. Example applications for such solutions include lane-level navigation and vehicle safety communications.

Today's vehicles primarily use the Global Positioning System (GPS), often in conjunction with vehicle odometry for correcting short term GPS biases. However, in many challenging environments, such as urban canyons, bridges and tunnels, multi-path fading or shadowing considerably degrades satellite positioning accuracy. While research has shown how position accuracy can be further improved to a few meters with motion sensors and map matching [9, 17, 20, 34] in some urban environments, these still face challenges in more extreme urban canyons. To achieve lane-level positioning, highly instrumented automated vehicle prototypes use cameras or LiDAR sensors to reference their measurements against available detailed models and imagery of the roadway. Creating, maintaining, and making available such detailed image models for all roadways is laborious and resource demanding, since it can undergo frequent changes due to reasons such as snow or falling leaves.

To reduce reliance on such resource intensive image registration, we investigate whether WiFi time-of-flight ranging as specified in the WiFi Fine Time Measurements (FTM) standard [8] are sufficiently uncorrelated with GPS measurements to achieve lane-level positioning in urban canyons. While WiFi [19, 58] positioning or is frequently used in smartphones, these have been based on received signal strength (RSS) positioning which is limited to an accuracy of tens of meters in urban canyons. In contrast, FTM ranging can achieve meter-level accuracy in open-space environments [3] [31]. Given this promise of improved accuracy and its wide availability, our study focuses on WiFi FTM and explores its utility in a vehicular context.

To address the lane-level urban canyon positioning challenge, this paper introduces Wi-Go[1], a scalable and accurate vehicle tracking technology that complements GPS and odometry with time-of-flight measurements and multilateration to surrounding access

[1]Wi-Go stands for WiFi, GPS and odometry based tracking

points through the recently standardized WiFi Fine Time Measurement (FTMs) protocol. It can opportunistically use WiFi access points deployed in buildings or in cars parked along the roadway. Since the access point location is usually not known a priori, we design FTMSLAM, a collaborative simultaneous localization and mapping approach to simultaneously track landmark (APs and parked vehicles) positions as well as moving vehicle positions. Realizing this requires addressing several key issues.

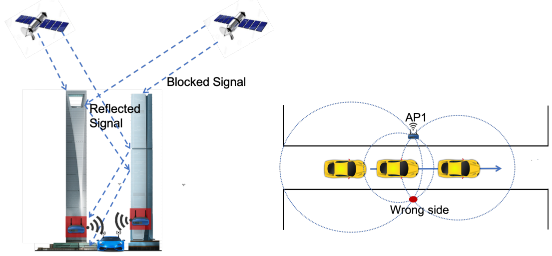
First, moving from a passive-client approach as in conventional RSSI-based WiFi positioning, to the actively-signaling-client approach used in Time-of-Flight based tracking, can introduce contention and channel congestion, particularly in densely populated areas such as urban canyons. This creates challenges when scaling to larger numbers of vehicles and access points. Consider a scenario in which hundreds of nearby vehicles send periodic FTM requests on the same WiFi channels. Since every FTM request triggers a sequence of packet transmissions specific to one client, this can easily exhaust the available channel capacity, introduce latency and therefore degrade the positioning accuracy for these clients. Note that this was not a concern for RSSI-based positioning since many clients can passively overhear the same access point message but due to the lack of synchronized clocks FTM requires round trip packet exchanges. To tackle this issue, Wi-Go mitigates channel congestion by optimizing the FTM message rates to each AP to maximize the tracking accuracy, while remaining below a cumulative allowed message rate in a given region.

Second, the FTM ranging process involving a series of packet exchanges introduces time offsets between the range measurements to individual access points, while standard multilateration method in wireless localization assume quasi-simultaneous ranging to multiple landmarks. This is particularly important in the vehicular context where a fast moving vehicle can travel a significant distance between individual measurements. To address this challenge, we propose *Mobile Multilateration*, a novel tracking algorithm, to track a moving vehicle with individual ranges obtained at different positions, while still leveraging odometery information to relate each position of the vehicle to its previous position.

Third, to minimize receiver complexity, it requires simultaneous localization and mapping (SLAM) techniques that only use range information while existing SLAM work heavily relies on bearing measurements as well (e.g.,[30, 51]). Wi-Go presents a range-only SLAM framework suitable for multilateration with Fine Time Measurements; it explicitly represents the uncertainty of both the position estimates via particle filters, and can use FTM measurements from a second antenna to resolve AP position ambiguity on linear roadways.

Fourth, in environments such as urban canyons, WiFi communication can also be heavily affected by multipath fading and shadowing. Simply using WiFi measurements that are potentially low quality will not yield the desired accuracy improvements over GPS. Therefore we design an uncertainty weighted multilateration technique that estimates and explicitly considers the quality of the current GPS and FTM measurements when updating the vehicle and access point location.

We prototyped Wi-Go with Intel 8260 wireless cards on a small form factor PC installed with roof-mounted antennas in a vehicle and implemented backend algorithms in a cloud server. We



(a) WiFi Vs GPS (b) Vehicle linear movement

Figure 1: (a) WiFi edge over GPS. (b) Effect of linear movement of a vehicle on AP localization error.

evaluated Wi-Go in two deep urban canyon environments of upper Manhattan as well as in a suburban residential setting. Wi-Go achieves 1.3 m median error for vehicle tracking, with relatively low AP density (4 APs), in our urban canyon dataset in which a baseline of a built-in GPS only achieves 9.04 m median localization error. Meanwhile, Wi-Go's adaptive algorithm can efficiently adapt the FTM message rate while still showing 41.7% improvement in terms of vehicle localization error.

Summary of Contributions. Wi-Go and its FTMSLAM algorithm make the following contributions compared to earlier positioning and SLAM algorithms:

- Designing a novel FTM-SLAM algorithm to intelligently complement GPS and odomentry with range-only WiFi FTM data, while addressing the resulting latency and multi-path challenges.
- Mitigating channel congestion due to active time-of-flight ranging messages, by intelligently distributing the cumulative allowed message rate over the nearby APs.
- Introducing the opportunistic use of parked vehicles for positioning by incorporating pairwise-distance estimates between APs in parked vehicles.
- Demonstrating and validating that the Wi-Go system with its use of WiFi FTMs meets lane-level positioning requirements through extensive experiments in two challenging urban canyon environments and a suburban setting, with a small fleet of 5 research vehicles.

2 BACKGROUND

In this section, we will review the WiFi Fine Timing Measurement, SLAM, and vehicle sensing concepts that this work builds on. WiFi positioning has evolved from initial fingerprinting [10] approaches to using Angle-of-Arrival [59], dead reckoning [57] and Time-of-Flight (ToF) [25]. The ToF-based WiFi distance measurements have recently been incorporated in WiFi standards, and promise widespread AP deployment support.

WiFi Fine Timing Measurement (FTM). IEEE 802.11-2016 Standards [8] has included the Fine Timing Measurement protocol, 802.11REVmc, to perform wireless ranging by measuring the round trip time (RTT) between an AP and a WiFi station (STA). The protocol subtracts processing times from the round trip time, converts it into a one-way time-of-flight estimate, and uses this to estimate range using typical propagation speed. As an interactive protocol, it conducts multiple message exchanges to achieve a higher ranging accuracy. Recent research [31] has confirmed that the FTM protocol can achieve meter-level accuracy in open space environments but the accuracy degrades in high multipath environments.

Android [4] and major vendors of WiFi chipsets have started to support the FTM protocol. However, their accessibility to PHY layer information like Channel State Information is quite limited [28] since the station has to associate with the AP, which takes several seconds [46]. This is not suitable for fast moving vehicles.

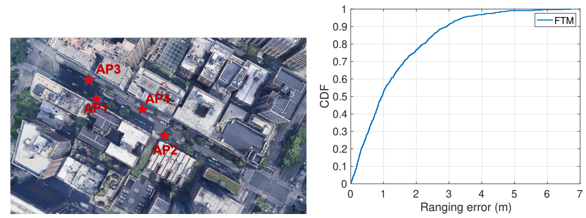
SLAM. Classic SLAM frameworks like FastSLAM [43, 51, 53] require ranging technology (e.g. LiDAR, or stereo cameras) that estimates the distance and bearing to a landmark. Range-only SLAM with RF or sonar beacons[21, 22, 27, 44, 45], approximates the vector connecting current pose with current location estimate of landmark, requiring the initial landmark location estimate to be accurate. Previous range-only SLAM frameworks used in robotics [16, 22, 45] has initialized landmarks through taking majority votes over multiple solutions. Each of the solutions either fits a pair of ranges (assuming the robot does not move in a straight line) or draws probabilistic samples from a circle around the robot location with radius equal to the range. These two initialization approaches may converge to the actual location and achieve high accuracy if the robot is not moving in a straight line, which is not the case in our application (as shown in Fig. 1(b)).

On the other hand, current WiFi SLAM [15, 23, 24, 30] either treats RSSI fingerprints as landmarks and tracks them simultaneously with user's location, or augments WiFi with cameras to estimate bearing to the RSSI-based fingerprints. None of these prior works have estimated the APs' locations simultaneously while tracking a user, because current propagation models that map RSSI to distance are not accurate enough and cannot generalize to different environments. WiGo introduces a new SLAM approach called FTMSLAM, to track APs and vehicles with range-only FTM range measurements. FTMSLAM uses novel opportunistic sensing algorithms including vehicle to AP bearing estimation, adaptive FTM range calibration learning, and vehicle location correction through APs street maps.

On-board vehicle sensors. Modern vehicles are equipped with sensors to measure vehicle speed, steering wheel angle, heading, yaw rate, etc. While this information is usually communicated on the CAN bus (and often readable from the On-Board Diagnostics (OBD-II) port), its encoding is specific to certain vehicles and proprietary [33]. In this paper, we leverage available on-board sensors to feed our FTMSLAM framework with precise odometry. Moreover, we leverage this odometry information as an independent, orthogonal source for correcting FTMs in over- and under-estimate cases.

3 CAN WIFI AUGMENT GPS?

Our Wi-Go system design is motivated by the observations that urban canyons, a key environment with degraded GPS, usually enjoy a dense deployment of WiFi Access Points (APs) (as shown in Fig. 1(a)). Current WiFi outdoor localization[32, 36] mainly counts on RSSI fingerprinting, which is shown to be less accurate due to several innate limitations, like multipath effects [10], variation of



(a) Urban Experiment (b) CDF

Figure 2: WiFi FTM ranging accuracy.

Approach	Vehicle GPS	GPS	Android Loc.
Error (m)	9.04	18.2	19.6

Table 1: Median tracking error of related technologies in Manhattan, NY.

performance between different devices of the same vendor [40], and low granularity [29].The recently available WiFi Fine Time Measurements, instead, promise a more robust, accurate, and fine-grained ranging measurement to enable mobile multilateration techniques, suggesting a possibility that WiFi FTM ranging could further complement GPS.

To examine our hypothesis, we conduct a preliminary experiment in an urban canyon environment (Manhattan, NYC). In this experiment, we place 4 access points on top of 4 parked vehicles as 'virtual' WiFi APs, and evaluate the ranging and location error, in the rest of the paper, using the following four different technology options:

(1) **WiFi FTM baseline**. Our WiFi AP is equipped with an Intel Dual Band Wireless-AC 8260 chipset that supports FTM capability; (2) **Standalone GPS**. Our vehicle is equipped with an after-market U-blox EVK-7P GPS receiver (< 1m precision in an open sky setup); (3) **Vehicle GPS**. Meanwhile, our vehicle retrieves vehicle GPS readings from OBD-II port. This built-in GPS has been internally corrected using the vehicle IMU and odometry sensors; (4) **Android Fusion Location API**. Finally, we collect location measurements through Android Fused Location Provider API from a Google Pixel 3a smartphone that is placed on the vehicle's dashboard. The Android Fused Location Provider API fuses multiple sensors including GPS, WiFi RSSI, and cell tower positioning together to provide localization information. These technologies summarize the state-of-the-art outdoor localization approaches which are either GPS, motion sensors dead-reckoning, or based on WiFi RSSI, Cellular RSSI, or a fusion of some of them.

In order to acquire ground truth locations of the vehicle, two Intel RealSense Depth Cameras are mounted on both sides of the vehicle's roof that log depth information of both sides of the road. We examine the recorded frames looking for landmarks such as light poles and trees and use these landmarks' positions to infer the vehicle's position. Specifically, when the landmark appears in the middle of the field of view, we use the landmark's location along with its depth to compute the current vehicle location.

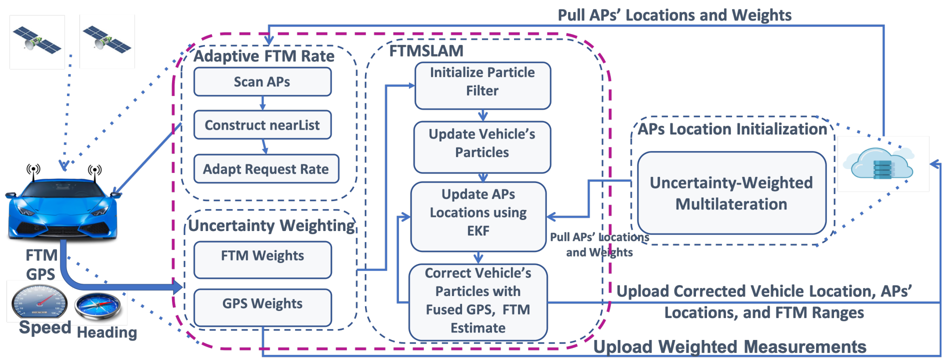


Figure 3: Wi-Go System Architecture Overview.

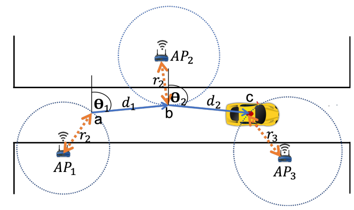


Figure 4: Illustration of our Mobile Multilateration. A vehicle moves from point a, to b, then from b to c.

Fig. 2(b) shows the ranging error of only WiFi FTM as other competing technology options do not report range measurements. The ranging error of WiFi FTM is 0.95 m median error (and 2.9 m at 90-percentile CDF curve). On the other hand, we also analyzed the localization errors of vehicle GPS, Android Fused Location, and standalone GPS as shown in Table 1. This preliminary result indicates that WiFi FTM seems to hold great promises of complementing and augmenting existing outdoor localization techniques.

In the next sections, we describe in detail the design of Wi-Go, which achieves this promise of lane-level positioning.

4 THE Wi-Go SYSTEM

The objective of Wi-Go is to provide a practical positioning system that scales in real world scenarios where tens of APs and hundreds of vehicles compete in the shared WiFi channels through active WiFi FTM ranging. Besides scalability, we aim for uninterrupted meter-level positioning by augmenting WiFi FTMs with GPS and odometry readings.

4.1 System Overview

Incorporating WiFi FTMs for outdoor positioning is challenging due to the following reasons: First, with extended FTM ranging latency, a fast vehicle could register multiple locations at different nearby APs, imposing a vehicle tracking challenge; second, FTMs are affected by multipath and are a range-only measurement that does not offer bearing information, which makes jointly locating APs and vehicles challenging; third, the injection of FTM packets from many vehicles onto the same WiFi channels causes network congestion;

To address the above challenges, we design Wi-Go as shown in Fig. 3. In each participating vehicle, Wi-Go collects WiFi FTMs from the surrounding APs, along with the timestamp, GPS reading, and the vehicle's speed and heading through on-board sensors. We assume that vehicles will at least occasionally enter open-sky GPS conditions, and that this could be used to establish position in the world coordinate frame. The Wi-Go system then tracks a vehicle by starting with an initial location estimate, gradually refining based on successive GPS readings, vehicle odometry, and WiFi FTM range measurements to access points positions. It uses simultaneous localization and mapping techniques to jointly estimate vehicle position and access point positions. As in current WiFi positioning systems, access point position estimates can be shared across vehicles through a server.

WiGo addresses the fast moving vehicle challenge through mobile multilateration, compensates for multipath through uncertainty-aware fusion, and controls FTM request through a congestion-aware optimization. We describe these techniques in the following subsections.

4.2 Mobile Multilateration

Tracking a rapid-moving vehicle is not an easy task, since it requires quick ranging due to vehicle speed. If a vehicle moves at 20 m/s and the standard FTM latency is 0.2 s, then the next ranging measurement will be at least 4 meters away. A new form of multilateration is thus needed for a moving vehicle to collect sequential ranging measurements at different precise locations from multiple nearby APs.

Fig. 4 illustrates this issue: A vehicle starts ranging at location A and obtains a ranging measurement r_1 from AP_1, then it gets r_2 from AP_2 at location B, and finally it collects r_3 from AP_3 at location C. Our goal is to find the current location of the vehicle $\mathbf{v}$, given APs locations $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$ and corresponding ranges $[r_1, r_2, ..., r_n]$. To tackle this issue and relate the measurements of location A and B to location C, we leverage vehicle odometry readings to estimate displacement and heading between these different locations. Formally, we formulate this problem of Mobile Multilateration as a non-linear least squares formulation:

$$\begin{aligned} \underset{v_n}{\textbf{argmin}} \quad & \sum_{i}^{n} (||\mathbf{x}_i - \mathbf{v}_i|| - r_i)^2 \\ \textbf{subject to} \quad & v_i = reckon(v_{i-1}, d_{i-1}, \theta_{i-1} + 180), \\ & (\text{where} \quad i = 1, ..., n) \end{aligned} \tag{1}$$

Through this formulation, we could identify the current location of the vehicle by minimizing the least square error while backwardly holding the constraint functions. For instance, with the last displacement and heading, we can derive the current location v_n from the preceding location v_{n-1}.

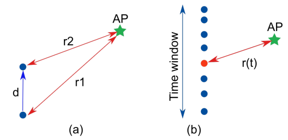


Figure 5: FTM Weighting. (a) Triangular inequality. (b) Assigning measurement confidence based on a time window.

4.3 Uncertainty Weighting

During the collection of measurements, weights are assigned to FTMs and GPS readings, representing the confidence of these readings, which will be used later in the fusion as described in Section 6.

FTM Weighting. We use vehicle wheel encoders and inertial sensors to assign weights to FTM measurements. In this step, we do not correct collected range measurements; instead, we simply assign weights to these measurements and feed them along with their weights to the FTMSLAM algorithm (Section 6). Triangle inequalities are used to evaluate measurement confidence: in Fig. 5(a), let d be the estimated displacement distance reported by the inertial sensors between two FTM measurement locations. Also, let r_1 and r_2 be the measured FTM ranges at these two locations. Then the following triangular inequality must be satisfied: $|r_1 - r_2| \leq d \leq r_1 + r_2$. For each FTM measurement r(t), we evaluate the triangle inequality of r(t) with another FTM measurement within a small time window (Fig. 5(b)). The weight $w^{FTM}(t)$ for measurement $r(t)$ is thus the ratio of measurement pairs that satisfy the triangle inequality over the total number of pairs.

GPS Weighting. GPS NMEA (National Marine Electronics Association) sentences are used to obtain data which is leveraged to assign a weight for the GPS location. For a given time t, this data includes: signal-to-noise ratios $SNR(t)$ for each observed satellite, number of observed satellites $n(t)$, GPS quality $q^{GPS}(t)$, and horizontal dilution of precision $HDoP(t)$. GPS quality indicates 3 different states: no fix ($q^{GPS}(t) = 0$), GPS fix ($q^{GPS}(t) = 1$), or differential GPS fix ($q^{GPS}(t) = 2$). *HDoP* measures the geometric quality of GPS satellites configuration in the sky [37]. The smaller the HDoP number, the better the geometry in terms of being spread in the sky, which has been reflected into the GPS location precision. The GPS weight for the system is calculated as $w^{GPS}(t) = \frac{q^{GPS}(t) * n(t) * mean(SNR(t))}{HDoP(t)}$.

4.4 Congestion-Aware Adaptation of FTM Request Transmission

When many vehicles try estimating their positions using requests and responses from multiple APs nearby, WiFi channel congestion may occur. Since FTM is an active measurement, it is initiated from the STA and consumes bandwidth for every burst it sends out. With hundreds of vehicles and tens of APs sharing limited WiFi channels, it is critical to limit how many FTM requests a vehicle can send out under fixed bandwidth restriction. This challenge is addressed in this section by adapting the samples per burst parameter (spb).

Problem Formulation. Thanks to pioneering studies on congestion control in vehicular networks [12–14], we assume that the vehicle estimates the maximum message rate limit that it is allowed to send without causing congestion. In this paper, the main task is thus to distribute and provide message rates for each vehicle to different APs nearby, while still honoring the cumulative message rate limit and minimizing the vehicle location error. Note that the messaging between vehicles and APs consumes channel bandwidth and that potentially affects APs as well. To address this, specific channels or partial bandwidth could be reserved for FTM messaging or a bandwidth limit can be set for such messaging. This will in turn determine the maximum message rate per vehicle.

To achieve this goal, as shown in the objective function (Eq. 3), we aim to minimize the key factors that contribute to vehicle localization error collectively: the geometry of the chosen subset of APs ($1 - \delta(\mathbf{spb})$), error model in AP location (e_{AP}), and FTM ranging error e_r (reflected in the error covariance matrix $\mathbf{C}_{\Delta r}$).

Approach. A naive solution would be to distribute rate limit equally over surrounding APs; however, not all APs in range can precisely estimate a vehicle's location. Therefore, it is important to determine the subset of nearby APs the FTM requests should be sent to. Collinearly distributed APs, for instance, fail to provide an accurate position estimation; similarly, APs that are in distant locations provide range measurements with larger errors.

Motivated by the above observations, we instead use a weighted round robin schedule for ranging to each nearby AP; in particular, we intentionally adapt the samples per burst parameter for each AP on one hand, while maximizing the vehicle localization accuracy on the other hand. For an AP, the spb can be set to zero if this AP is unlikely to be useful, and the sample rate adapts based on the location of a vehicle and its AP counterpart, respectively.

Given that $\mathbf{x}$ is the AP's location estimates, $\mathbf{v}(t-1)$ is the last location estimate for the vehicle, and $\mathbf{r}(t)$ is the vector of range measurements to these APs, the ranging model would be $\mathbf{r}(t) = distance(\mathbf{x}, \mathbf{v}(t)) + \mathbf{e}$. $\mathbf{e}$ represents the total error originated from FTM ranging errors $\mathbf{e}_r$ plus errors in AP's locations ($\mathbf{e}_{AP}$). As distance is not a linear function of locations of APs and a vehicle, we can linearize these equations using initial estimates of $\mathbf{x}$ and $\mathbf{v}(t)$. Through this linearization, we can determine corrections to the estimates and obtain current location of the vehicle: $\Delta\mathbf{r} = \mathbf{A}\Delta\mathbf{v} + \mathbf{e}$, where $\mathbf{A}$ is an nx3 matrix of the partial derivatives (Jacobian) of the distance function with respect to the unknown vehicle locations. Through a standard least squares solution ($\Delta\mathbf{v}$), the covariance matrix of the solution is:

$$\mathbf{C}_{\Delta\mathbf{v}} = (\mathbf{A}^T \mathbf{C}_{\Delta\mathbf{r}}^{-1} \mathbf{A})^{-1} \tag{2}$$

where $\mathbf{C}_{\Delta\mathbf{r}}$ represents the error contributed by each AP. As this error is dependant on the location of AP (e.g., some APs could be more affected by multipath than others), these errors cannot be assumed to be the same across different APs. We therefore only assume the errors of APs are independent, and the standard deviation of the total error for each AP can be calculated as $\sigma = \sqrt{\sigma_{e_{AP}}^2 + \sigma_{e_r}^2}$.

To optimize the vehicle localization error, we derive our objective function from the covariance matrix of the least squares solution (Eq. 2) as follows:

$$\begin{aligned} \underset{\mathbf{spb}}{\mathbf{argmin}} \quad & ((1 - \delta(\mathbf{spb}))\mathbf{A}^T \mathbf{C}_{\Delta\mathbf{r}}^{-1} \mathbf{A})^{-1} \\ \mathbf{subject\ to} \quad & \sum spb_i = rate_{limit} \\ & spb_i \geq 0, \qquad 1 \leq i \leq |APs| \end{aligned} \tag{3}$$

where $\delta(\mathbf{spb})$ has the same form as the Delta function and $\mathbf{e_{AP}}$, $\mathbf{e_r}$ represents the error of APs' location estimations and range measurements respectively. In this model, prior AP location estimates are assumed to exist already and each estimate has its error derived from the covariance of that estimate; at the same time, newly observed APs are be ranged occasionally through a separate process in order to establish an initial position estimate.

The error model used for FTM range estimates can be estimated empirically by fitting the linear regression function ($e_{r_i} = a \times dist_i + b$) using some real world data [31]. The ranging error is approximated using the Central Limit Theorem when spb samples are taken and averaged. For large spb, the variance of that error can be approximated as $\frac{\sigma^2}{spb_i}$: $\sigma_{e_r} = \sqrt{\frac{(a \times distance(\mathbf{x}, \mathbf{v}(t-1))+b)^2+\sigma_{emp}^2}{spb_i}}$. So the ranging error to a certain AP is estimated based on its last estimated distance and the empirically obtained standard deviation σ_{emp} of these measurements.

5 THE FTMSLAM ALGORITHM

All of these ideas are put together in the FTMSLAM algorithmic framework, the cornerstone of the Wi-Go system. In FTMSLAM, the vehicle and the surrounding WiFi APs are localized and tracked, by incorporating WiFi FTMs, GPS, and on-board sensor measurements.

Novelty. FTMSLAM utilizes range-only measurements (FTM), in direct contrast to conventional SLAM approaches which require both range and bearing measurements. To conquer this challenge, we intentionally take advantage of a suite of novel methods, *vehicular opportunistic sensing*, to update and correct the locations of vehicle and APs respectively. The opportunistic sensing methods include (1) vehicle to AP bearing estimation (Section 6.2), (2) resampling particles from FTM multilateration to refine estimated location (Section 6.3), and (3) adaptive FTM range correction (Section 6.3).

5.1 Location Modeling and Initialization

Location Modeling. Inspired by classic FastSLAM [53], we model the vehicle location using particle filters, which provides a non-parametric probabilistic position representation. Through this model, we update a vehicle's particles through dead reckoning, and then further correct the distribution of particles using the fused location estimate of GPS and WiFi FTM measurements. At the same time, we model the AP location as a Gaussian distribution. In our study, it is found that the lack of angular information in FTM measurements introduces several new challenges: the requirement for accurate initialization, the need for fusing GPS and FTM measurements for the correction step, and the demand to estimate a range calibration factor for different APs. These issues are addressed in Sec 6.2.

AP Location Initialization. This is done through pre-collected, crowd-sourced GPS traces along with FTMs. We estimate the locations of APs using uncertainty-weighted mobile multilateration (Section 4). To be specific, the optimization problem is formulated as follows: given the location $v_i(t)$ of vehicle i at timestamp t, the collected FTM range $r_{ij}(t)$, and their weights $w_{ij}^{FTM}(t)$, the goal is to identify the location of unknown AP x_j. To derive the accurate location information, we compensate the length of wired cables used to extend WiFi antennas by subtracting c from all the FTM ranges $r_{ij}(t)$. We thus solve this mobile multilateration problem through the following weighted non-linear least squares formula:

$$\underset{x_j}{\text{argmin}} \sum_{i}^{N_v} \sum_{t}^{T} w_{ij}^{FTM}(t)(||\mathbf{x}_j - \mathbf{v}(t)|| - (r_{ij}(t) - c))^2 \quad (4)$$

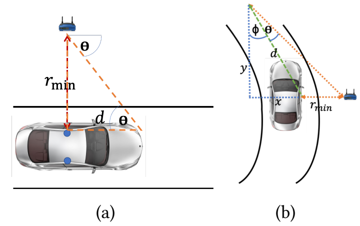


Figure 6: Angle estimation illustration.

5.2 Opportunistic Bearing Estimation

Using dual antennas connected to a WiFi transceiver placed on the vehicle, the bearing is estimated from the vehicle to an AP. As illustrated in Fig. 6, two antennas are fixed on the right and left edges of a vehicle to obtain differential measurements across the direction perpendicular to the vehicle bearing. In practice, nonetheless, the noise in FTM readings could exceed the actual difference between two measurements (left antenna and right antenna), which is highly depending on the width of the vehicle. Even worse, FTM protocols currently implemented on commercial WiFi transceivers do not grant access to any PHY-layer information that could help estimate the angle of arrival (e.g. signal phase).

To overcome this practical challenge, we consider the fact that we can estimate which side the APs are placed as well as the minimum range r_{min} to that AP over the first round of FTM measurements. Later, when we have another estimated minimum range through a new visit to this AP, we take the weighted average between the new estimate and the historical estimates to improve estimation accuracy. In the subsequent rounds, we can estimate the bearing information to that AP, whenever we reach the nearest point to the AP (at this point, the bearing should be either 90 deg or 270 deg from the vehicle's heading). As the vehicle moves forward, we can estimate the distance d from the nearest point to the AP by integrating the current vehicle's speed. With r_{min} and d, we can estimate θ as illustrated in Fig. 6(a): $\theta_i = tan^{-1}\frac{r_{min}}{d_i}$. This equation can be generalized for any road shape as illustrated in Fig. 6(b). Given r_{min}, d, and ϕ, we can estimate θ: $\theta_i = tan^{-1}\frac{x+r_{min}}{y} - \phi$, $x = sin(\phi) * d$, $y = cos(\phi) * d$.

In this algorithm, we heavily rely on the minimum range to the AP, which is attributed to the fact that FTMs are more accurate over short distances. As a result, the estimated bearing through the algorithm is relatively accurate; however, the accuracy could decrease as d increases, since the accumulation of speed sensor error negatively affects the estimated d. To estimate the distance from the vehicle to the AP more accurately, we combine the two FTM measurements from the right and left antennas respectively through a weighted average using the FTM weights.

5.3 FTMSLAM Tracking Framework

FTMSLAM Initialization Step. FTMSLAM requires initializing both the vehicle's location estimate and the APs' location estimates first. To do so, we use a GPS measurement with a sufficiently large GPS weight, or alternatively, through WiFi FTM multilateration with sufficiently accurate estimated locations of surrounding APs.

Update/Prediction Step. In this step, we update the locations of the vehicle and the currently discovered APs. To update a vehicle's location, we estimate its displacement from its previous location, together with the vehicle's heading using its in-vehicle sensors (these sensors provide vehicle kinematic information such as speed, steering wheel angle, heading, and yaw rate). Here, the displacement can be estimated as follows: $d = \frac{1}{2}(s_t - s_{t-1})\Delta t$. Through estimating both displacement and azimuth, which is the heading with respect to the north, we can update the vehicle's location using Vincenty's formula[55] which takes the Earth curvature into account.

In the meantime we also update the locations of surrounding APs, through Extended Kalman Filter (EKF) [53]. Given that the bearing to APs is not always available, we follow a range-only SLAM approach for opportunistically updating APs if the bearing estimation becomes unavailable. Here the location of the AP is updated, by using the vector linking the current estimate of vehicle's location with previous estimate of this AP location. Since this vector depends on the AP location initialization, an accurate initialization is crucial for such a range-only approach. To further improve the accuracy, the bearing estimate is improved when the bearing estimation algorithm is invoked.

Correction Step. In this step, we update the weight of each particle. By correcting the particle's distribution of the vehicle using both GPS and FTMs, we update the location of the vehicle and avoid the accumulation of motion sensors error. There are a variety of different places: in rural or suburban areas where the GPS is accurate but WiFi is not available, and the reverse could be true in urban canyons where GPS reception is poor, but dense APs have already been deployed. It is likely that we can correct the distribution of particles based on fused correction weight between GPS and FTM estimate.

With dense deployment of APs, we can estimate the current vehicle's location using weighted multilateration by using FTMs, their weights ($w_j^{FTM}(t)$) and current location estimates of these APs. Based on the distance between each particle and the current FTM estimate for the vehicle's location, the FTM weight of that particle is updated, $w_{part_i}^{FTM}(t)$. However, this dense enough deployment of APs is not always available. In this case, the weight of the particle is updated based on the average of the error between range measurements and estimated range, which is the distance between the particle and the current location estimate of each AP. Similarly, the GPS weight of that particle $w_{part_i}^{GPS}(t)$ is updated based on the distance between current GPS estimate and the particle.

To fuse these two weights, we take the weighted average of these two particle weights: $w_{part_i}(t) = w^{FTM}(t) * w_{part_i}^{FTM}(t) + w^{GPS}(t) * w_{part_i}^{GPS}(t)$. As the FTM weight of that particle is estimated by leveraging FTM measurements to multiple APs, we estimate this as the average of these FTM weights, so $w^{FTM}(t) = \sum_j^N w_j^{FTM}(t)/N$, where N is the number of currently seen APs.

FTM Resampling. In urban canyons, a vehicle exhibits degraded GPS readings, leading to the particles being tens of meters away from its actual location. This requires an aggressive and fast way of re-weighting the particles when the vehicle starts to observe a sufficient number of APs for multilateration. In such cases, when the current estimated vehicle location through FTM is far away from the last estimated location, all particles will end up with zero weights. To resolve this particle deprivation problem, we resample a small portion of the particles from the FTM multilateration location.

Adaptive FTM Range Correction. FTM weighting cannot mitigate multipath effects by itself, as multipath error in certain areas can be consistent and ubiquitous. An adaptive mechanism for correcting FTM range is thus needed to mitigate multipath effects.

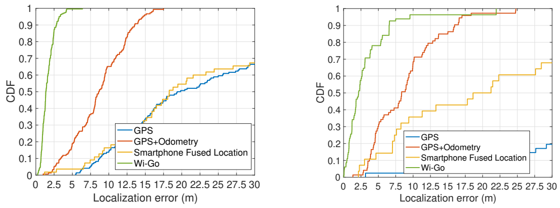
In each update step, FTMSLAM uses the error between the estimated range (FTM) and the corresponding distance between an AP and a vehicle to update AP location. After multiple iterations, the accuracy of AP location estimate is seen to improve, leading to a better estimation of vehicle location. Inspired by this observation, we could calculate a range calibration factor to compensate multipath error, since the distance error to an AP would eventually converge to its actual range error. In practice, a 2D spatial map of the range calibration factor for each AP is built by clustering locations with similar patterns of multipath effects. Hence similar correction values and estimated calibration factors could be used to compensate similar multipath effects.

6 PARKED VEHICLES AS PSEUDO APS

In addition to opportunistically using stationary WiFi access points in buildings or roadside infrastructure that support the WiFi FTM standard, Wi-Go can be extended to also take advantage of WiFi devices in parked vehicles as Pseudo WiFi APs. As a vehicle parks, it can switch from WiFi station (STA) mode to AP mode through relatively straightforward WiFi firmware changes, if vehicle battery management system permits. Using parked vehicles as pseudo APs could effectively increase the density of WiFi APs, and also add high-quality pseudo APs since parked vehicles at the street curb are likely to have a line-of-sight propagation to moving vehicles. This innovative idea, nonetheless, will change several design choices we outlined in Section 4-6. In this section, we intend to briefly discuss these issues and also shed light on their high-level solutions.

Pseudo AP Location Initialization. The last estimated location of a parked vehicle (just before the ignition switches off) could serve as an initial location for that pseudo AP. More importantly, these parked APs can estimate distance to surrounding APs, before switching to a AP mode. This method can effectively estimate the pairwise distances between a subset of APs, which helps jointly initializing APs locations. We designed a different optimization method for parked, stationary vehicles because parked vehicles can obtain pair-wise rang measurements between them. Specifically, we can solve the multilateration problem jointly for all N_{AP} APs, instead of solving the problem independently for each AP:

$$\begin{aligned} \underset{x}{\textbf{argmin}} \quad & \sum_{j}^{N_{AP}} \sum_{i}^{N_v} \sum_{t}^{T} w_{ij}^{FTM}(t)(||\mathbf{x}_j - \mathbf{v}(t)|| - (r_{ij}(t) - c))^2 \\ \textbf{subject to} \quad & ||\mathbf{x}_j - \mathbf{x}_k|| = r_{jk}, \qquad k = 1, 2, \ldots, N_{AP} \end{aligned} \tag{5}$$



(a) Parked Cars (114th st.)   (b) Indoor APs (49th st.)

Figure 7: Vehicle tracking error of Wi-Go Vs current localization systems in Manhattan experiments.

Approach	Parked Cars	Indoor APs
Wi-Go	1.9 m	3.6 m
GPS+Odometry	14.3 m	16.4 m
GPS	18.9 m	27.4 m
Smartphone Fused Loc.	17.8 m	16 m

Table 2: AP localization mean error in Manhattan experiments.

Pseudo AP Location Modeling. Parked vehicles bring another challenge regarding modeling of pseudo AP location: Regular APs inside of buildings rarely change locations (maybe once a year); now, with these parked vehicles, pseudo APs could move more frequently (multiple times per day). To tackle it, we model pseudo AP location probabilistically using a particle filter approach that takes vehicle mobility into account.

Vehicle Battery Duty Cycle. Powering an AP while a vehicle is parked, may raise a battery issue if the pseudo APs are kept on for a long duration. To tackle that, we could either develop an intelligent power management algorithm that relies on Bluetooth Low Energy to wake up WiFi AP unit, or adopt a simple solution to stop vehicles from switching to Pseudo APs if its battery is less than a pre-set threshold (e.g. 80%).

The design and implementation of this idea merit an independent study, and we leave it for future work.

7 EVALUATION

7.1 Experimental Setup

WiFi FTMs. We setup our vehicle with a small form factor computer that contains two Intel Dual Band Wireless-AC 8260. Each WiFi transceiver connects to two WiFi external antennas: 6dBi RP-SMA Dual Band 2.4GHz 5GHz with 1.637 m cable to attach the antennas on the roof (we subtract that length, c from FTM ranges). We leverage an open Linux FTM tool [31] to initiate and extract FTMs from these WiFi cards. On the AP side, we use ASUS Wireless-AC1300 RT-ACRH13 APs which are configured to respond to FTM requests as a built-in capability.

Vehicle Odometry. We connect OBDLink MX to the vehicle OBD-II interface, and retrieve odometry readings to the small form factor computer through Bluetooth interface (Plugable USB adapter).

7.2 Urban Canyon Evaluation

In this section, we evaluate Wi-Go in terms of localization error through an urban canyon experiment in which GPS is significantly

Source	Single FTM	Processing
Median Latency (ms)	20	2.2

Table 3: Wi-Go Latency.

degraded. Moreover, we show how Wi-Go will react in terms of latency, as many APs and vehicles are actively contending over WiFi channels through WiFi FTM packets and usual WiFi data traffic in urban canyon.

Experiment Summary. Fig.2(a) illustrates one of our experiments in upper Manhattan, New York City. We park four vehicles in a single street (182.6 m long) as shown in the figure as red stars, and we place an AP on each vehicle. In the fifth vehicle, we use our form factor PC configured as a WiFi station (STA), which continuously ranges surrounding APs. We also collect odometry, vehicle GPS, and standalone GPS readings from this vehicle. The fifth vehicle drives down the street multiple times to gain statistically meaningful results. We repeated the same experiment in Midtown Manhattan, where we placed six APs inside local shops.

Ground Truth. In obstructed environments like urban canyons, even high precision GPS will experience large errors. It is also infeasible to manually measure a vehicle's location while driving. Due to these reasons, getting accurate ground truth to compare our results to was challenging. We obtain ground truth by leveraging depth cameras as mentioned earlier in Sec. 3. To validate our ground truth methodology, we compare depth readings of a light pole from depth camera and laser range finder (or measuring tape) over a range of 3 to 12 m in a parking lot environment. The resulting error is within 0.73 m. To obtain the ground truth location of the APs, we take pictures of the APs' surrounding environments (buildings and other landmarks), cross reference with Google Street View, and drop pins accordingly on Google Maps to obtain the coordinates of the APs. Prior studies show that Google Earth has an accuracy close to 1 m in metropolitan cities like Montreal [26] and Rome [47].

For these experiments, we compare the performance of Wi-Go to other technologies mentioned earlier. In Fig. 7(a), we show the cumulative distribution function (CDF) of the localization error of the vehicle using Wi-Go. Our Wi-Go achieves 1.3 m median error, and 2.8 m 90-percentile. In contrast, vehicle GPS (GPS+Odometry) achieves 9.04 m median error, and 13.5 m 90-percentile, compared to standalone GPS achieving 19.6 m median error, and 42.1 m 90-percentile. Finally, Android Fused Location Provider API achieves 18.2 m median error, and 58.09 m 90-percentile. In another experiment (Fig. 7(b)), where APs were inside of local shops, the median tracking error of Wi-Go is 2.1 m, while 90-percentile error increases to 6.5 m due to extra signal degradation caused by concrete walls of these shops.

Table 2 shows the AP localization error of Wi-Go system when compared to baseline technology options. As the vehicle moves across the street with more rounds of driving, Wi-Go improves the AP localization accuracy, reaching 1.9 m mean localization error across all the APs. This result demonstrates a significant performance improvement of Wi-Go system over baseline solution. For the baseline achieving 14.3 m mean error, we utilize the traces via vehicle GPS to estimate the locations of APs using multilateration and FTM ranges.

Median latency is reported in Table 3. The median latency to acquire a single FTM reading, i.e. the time difference between the moment that we make a system call to initiate FTM process to the moment that call returns with range measurements from a single AP. We measure this median latency, in our current setup, to be 20 ms. On the other hand, the median processing latency of FTMSLAM is 2.2 ms. The total latency of Wi-Go system is controlled by the latency of extracting the sensing information such as WiFi FTM and OBD readings. With a normal vehicle speed, this measured latency may lead to error of a few meters. We believe that this latency could be improved in the actual production system by optimizing the FTM extraction tool.

Simulating Dense Environment. We use ns-3.30.1 to simulate a scenario of dense environment including 900 vehicles and 200 APs. The mobility trace is generated using SUMO mobility simulator [5] for an approximately 500 m radius around Times Square, NYC, imported from OpenStreetMap [6]. We imported a map of the selected neighborhood in Manhattan to create both the mobility traces of the vehicles and determine the location of buildings to increase simulation realism.

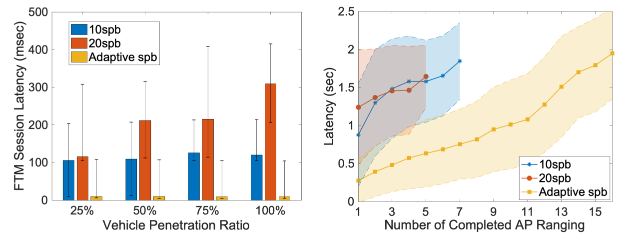
In this simulation, we implemented the WiFi FTM protocol following the technical specification of IEEE 802.11 standard [8] (e.g., packet format and packets size). For system parameters that are not directly regulated in the standard (e.g., expiration timers and other supporting parameters), we directly infer them from our empirical experiments. We log the latency for getting a single FTM ranging in our simulation.

Simulation Parameter	Value
Simulation time	30 sec
Transmission power	16.5 dBm
Channel bandwidth	80 MHz
Channel number	155
Line-of-sight reference pathloss	21.87 dB
Line-of-sight pathloss exponent	3.39

Table 4: Simulation configuration.

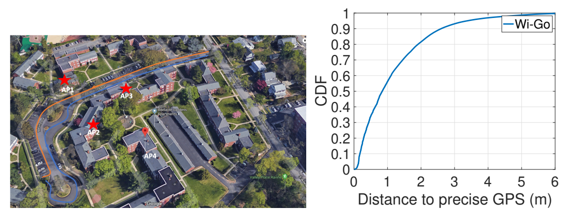
Table 4 shows the configuration used for the simulation. To better capture the impact of the multipath effects in an urban environment, the simulator distinguishes obstructed, none-line-of-sight (NLOS) communications from line-of-sight (LOS) communication as follows: when there is a packet transmission, the simulator identifies the communication category and then chooses an appropriate propagation model to calculate the received signal at the receiver. The obstructed building signal propagation model is an implementation of Mangel et al. [41]. For the line-of-sight model we used pathloss components from a recent industry consortium (CAMP) [7] empirical measurement study, which considers the impact of vehicular traffic condition on the pathloss.

Fig. 8(a) illustrates the median latency of different approaches, grouped together for specific vehicle densities. The error bars indicate 25^{th}% and 75^{th}% of the latency for each bars. Vehicle density is captured by the percentage of cars equipped with Wi-Fi, which we call vehicle penetration ratio, and simulation results are presented for four penetration ratios. It is clearly shown that Wi-Go, labeled as *Adaptive spb*, significantly outperforms the other competing approaches.



(a) (b)

Figure 8: (a) 25^{th}%, median, and 75^{th}% latency for Wi-Go (adaptive spb) compared to baselines using ns-3 simulations, and (b) mean and standard deviation of the latency versus number of completed AP rangings in one AP list scan for 75% penetration ratio.



(a) Residential Experiment (b) CDF

Figure 9: Evaluating vehicle tracking accuracy in the residential apartment complex experiment.

Approach	Mean	Min	Max
Wi-Go	2.6 m	1.3 m	4 m
GPS+Odometry	3.6 m	2.9 m	4.3 m

Table 5: AP localization error in the residential apartment complex experiment.

Fig. 8(b) illustrates the end-to-end latency from the moment that a vehicle starts with a list of APs until it successfully finishes its entire FTM session with the n^{th} AP, averaged across entire simulation. Note that the vehicle starts over with a fresh list of scanned APs either after finishing the current list of APs or after 2.5 s, whichever comes first. The shaded area around each curve shows the standard deviation. It is shown that on average the vehicle can start triangulation by having the third FTM session completed after 500 msec with our proposed adaptive spb approach; in contrast, using 10spb and 20spb approaches, the vehicle has to wait approximately 1500 msec and cannot complete ranging with more than seven APs.

7.3 Residential Apartments Evaluation

Summary. In our second experiment, we evaluate Wi-Go in a residential apartment complex area. We place three APs inside student apartments, in the location where residents usually keep their access point, as shown in Fig. 9(a). The choice of apartment units was limited to a set of volunteers who provided access to their units. Our data collection has been scheduled over a week with two trips each day (morning and afternoon). One 'trip' refers to each marked colored path in Fig. 9(a) (e.g. blue path is one trip).

Ground truth. Ground truth locations of APs are estimated by finding the latitude and longitude of the nearest window on Google Maps and correcting with manually measured AP distance to the window. Regarding vehicle localization error, we use high precision GPS as ground truth to evaluate vehicle localization error.

Fig. 9(b) shows the CDF of the vehicle localization error for this residential area. Wi-Go, with vehicle initialization through accurate GPS measurement, achieves 0.8 m median localization error, and 3.2 m 90-percentile. Table 5 summarizes the AP localization error for Wi-Go compared to the baseline solution (GPS corrected with odometry readings). Our Wi-Go can determine the locations of the APs with 2.6 m mean error over all APs compared to 3.6 m mean error for the baseline. For this baseline result, we filtered noisy GPS readings using our uncertainty weights for GPS, and then used FTM measurements and multilateration to determine the AP's position.

7.4 Micro Benchmarks

Effect of varying number of APs. The density of APs is an important factor affecting vehicle localization error. In Fig. 10, we study this effect in the Manhattan experiment, in which WiFi FTM dominantly affect localization error compared to suburban areas. This figure shows that, as expected, the median of vehicle localization error decreases with increasing number of APs, as we increase more reference points.

Effect of different FTMSLAM algorithms. We compare here the impact of our proposed algorithms over standard multilateration on AP localization error. Fig. 11 shows that our algorithms can gradually improve the average localization error of APs. In our urban canyon dataset with low-quality GPS measurements, adding our GPS and FTM weights to the standard multilateration does not improve the accuracy significantly. In contrast, as we apply FTMSLAM, our tracking framework with our crafted weights, the average localization error drops to 5.8m. When we apply our complete FTMSLAM, by correcting vehicle's particles with FTM multilateration estimate and bearing estimation, the AP localization error is further reduced to 1.9 m. In the residential experiment, Fig. 11 shows that our GPS and FTM weights could significantly improve the average localization error of APs, since a number of outliers exist in these measurements. After we apply complete FTMSLAM, the AP localization error decreases from 4.2 m to 2.6 m.

Effect of varying FTM correction factor. FTM ranging is affected by multipath as shown in previous work [31]. We study how the correction factor, which we subtract from the FTM ranges, could affect FTMSLAM in terms of AP localization error. Fig. 12 indicates that there is an optimization value that minimizes the AP localization error. According to our evaluation, this value is different between our two datasets; as expected, it is higher in the residential dataset compared to the urban canyon dataset. This is because, in the residential scenarios, the APs are placed inside the building and thus more susceptible to multipath, in direct contrast to urban scenarios where APs are on the top of the vehicles characterized with Line-of-Sight signal propagation. Our FTMSLAM derives this FTM correction factor **automatically** through our adaptive FTM range correction algorithm.

AP Location Convergence. Wi-Go improves APs location estimations over time. Wi-Go can converge to meter-level accuracy of an AP location estimation with relatively small number of visits to this AP. Specifically, Wi-Go can converge in Manhattan to 1.9 m, 3.6 m average AP localization error after 10, 3 visits to parked cars, indoor APs (114th st., 49th st.) with 488, 3273 average FTM readings, respectively. The number of readings is a function of the number of visits and the traffic condition (average vehicle speed). Our system can also converge to 2.6 m AP localization error after 14 visits to a residential area with 150 average FTM readings. Note that as more vehicles visit an AP, AP location accuracy improves and, as a result, vehicle localization accuracy improves.

7.5 Evaluating Adaptive FTM Rate

In this section, we evaluate our solution to adapting the FTM message rate to maximize vehicle localization accuracy, while being constrained to a cumulative message rate.

Experimental Results. We conduct an experiment in a suburban area, in which we place 12 APs (first seven are indoors and the rest are outdoor) as shown in Fig. 13(a). In this experiment we aim to show the importance of wisely ranging to a subset of the nearby APs, as well as to study how this mechanism affects the vehicle localization accuracy. To do so, 12 APs are deployed in an area of 117m X 36.5m, and we assume that the locations of these APs are already estimated through previous steps (wardriving multilateration, or FTMSLAM with minimal sample per burst). To avoid any complication, we do not use any preprocessing algorithms (e.g., FTM uncertainty weighting or FTMSLAM), but we utilize nearby APs to track the vehicle using standard multilateration and count only on WiFi FTMs. We use a high precision GPS receiver as ground truth to calculate the final localization error.

Fig. 13(b) compares our adaptive approach to the default approach using fixed spb per AP (2 and 10). As expected, the 2*spb* approach is susceptible to FTM noise, resulting in a high localization error of 9.2 m median error, and more importantly, 50 m 90-percentile. On the other hand, by increasing the parameter to 10*spb*, it is expected to improve the location accuracy, as there are more samples to average out the noise. However, to our surprise, this higher spb approach generates 11.5 m median error, and 52.1 m 90-percentile. This counter-intuitive observation could be explained as follows: Though we increased the sampling set for statistical accuracy, we also significantly increase the opportunity to collect more inaccurate, noisy data from faraway APs.

Our proposed adaptive approach, instead, takes all these factors into account. Our adaptive algorithm achieves 6.7 m median error, and 24.6 m 90-percentile, respectively. These results validate that our approach improves the localization error, since it intelligently selects the subset of APs which are nearby, less affected by FTM noise, and with a lower horizontal dilution of precision.

7.6 Discussion

In this section we discuss the major lessons learnt from our study and the results obtained from different scenarios are related and scrutinized.

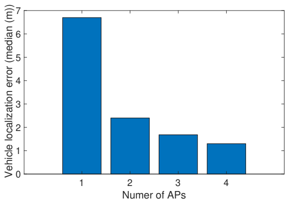


Figure 10: APs density effect on vehicle's localization error in Manhattan.

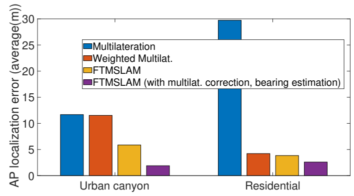


Figure 11: FTMSLAM algorithms effect on AP localization error.

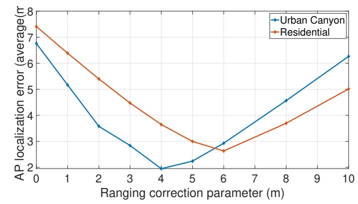
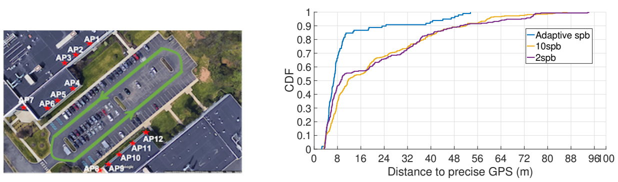


Figure 12: Ranging correction effect on AP localization error.



(a) Scalability Exper. (b) CDF

Figure 13: Evaluating our adaptive FTM message rate approach leveraging FTM ranges only.

Vehicle location error vs. AP localization error. As shown in Fig. 7(a) and Table 2, we notice that the vehicle localization error is slightly lower than the AP localization error, which is attributed to two factors: first, the multilateration reference points (i.e., vehicle trace) used to locate the AP are collinear, while the reference points used to locate the vehicle (i.e., widely spread AP locations) are mostly not, leading to lower dilution of multilateration precision. Second, AP localization is only limited by FTM measurements; the constraints on vehicle location, on the other hand, come down to GPS, odometry, and FTM measurements. This results in more accurate location estimation.

Urban canyon results vs. residential results. The results using the urban canyon and residential datasets consistently show that FTMSLAM alleviates both GPS blockage in obstructed environments and occasional GPS noise in unobstructed environments. Interestingly, the average AP localization error is slightly higher in residential areas, due to signal multipath. We observe that it is imperative to initialize our system with sufficiently accurate location as initial conditions, which is possible using GPS in the residential area, or using Maps/FTMs in urban canyons. Wi-Go is designed to filter noisy measurements when it happens, thus only benefiting from reasonably accurate measurements. As shown in Fig. 11, the different components of our system have different impacts between the two datasets.

8 RELATED WORK

Outdoor Localization. In urban canyons, GPS can achieve on average 24.3m error [34], which can be reduced by 6-8m as shown recently [38]. ParkLoc [20] localizes cars with an accuracy of 4.8m in underground parking garages using inertial sensors in the smartphones of the people inside the car, and semi-supervised learning algorithms. Carloc [34] can track vehicles outdoors (2.7 m mean error) by matching digital maps and leveraging proprietary vehicle sensors to detect roadway landmarks (e.g. speed bumps, stop signs). The LTE standard [18] indicates that a localization accuracy of 20-30m can be achieved with trilateration methods and using neural networks in tandem allows for a minimum error of 2.3m to be possible [50].

WiFi Localization. Indoor localization either assumes the knowledge of APs locations [42, 54] or leverage fingerprinting techniques [36, 48, 56]. Except for fingerprinting, the position of the AP is usually assumed to be known. Locating APs while driving by through ware-driving could lead to up to 32m error [35]. This can be improved by estimating angle of arrival reaching 10-30m localization error [52]. Fingerprinting strategies, however, involve heavy overheads, relatively low accuracy compared to ToA methods similar to GPS. CUPID [49] leverages CSI for indoor tracking, achieving 2.7m median error. WOLoc [58] offers WiFi RSSI-based outdoor tracking through semi-supervised manifold learning technique.

Scalability. Banin et al. [11] proposes a solution using multiple access points which are custom made for using WiFi FTM and communicating with each other to form a geosynchronous system, where the clients track their positions passively, with only the APs transmitting. Llombart et al. in [39] simulate a technique that can be used for mobile devices to select three nearby APs for trilateration. The mobile device to be positioned finds three nearby APs, associates with each of the three APs, calculates the distance and then dissociates. Dedicated hardware is required for both the AP and the mobile device in this method.

9 CONCLUSION

This paper introduces Wi-Go, a vehicle localization system that uses WiFi FTM measurements to achieve lane-level accuracy in challenging urban canyons where GPS accuracy is degraded. It fuses WiFi FTMs with GPS and odometry in the FTMSLAM framework, to simultaneously estimate vehicle positions and the positions surrounding WiFi APs. We further design Wi-Go to adapt the rate of FTM packets in order to mitigate network congestion and latency arising from WiFi FTM's active ranging approach, while maximizing vehicle localization accuracy. We evaluate the system in the urban canyons of Manhattan as well as suburban residential areas. Wi-Go achieves 1.3 m median error in an urban canyon setting with access points on parked vehicles, 2.1m median error in a crowded area with access points in buildings, and 0.8 m median localization error in a suburban environment with access points inside apartment buildings. This shows promise for using WiFi FTM measurements for vehicle positioning even in multi-path rich urban canyons.

REFERENCES

[1] https://goo.gl/AW4Pto. Self-Driving Cars Market Growth.
[2] https://waymo.com/. Google WAYMO Self-Driving Car.
[3] https://goo.gl/BSUCdG. Wi-Fi CERTIFIED Location.
[4] https://goo.gl/1z1LhE. Android Pie Indoor Positioning.
[5] https://sumo.dlr.de/. Simulation of Urban MObility (SUMO).
[6] https://www.openstreetmap.org/. OpenStreetMap.
[7] https://www.campllc.org/. Crash Avoidance Metrics Partners LLC (CAMP).
[8] "IEEE 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 Std 802.11-2016 (Revision of IEEE Std 802.11-2012)"*, pages 1–3534, Dec 2016.
[9] H. Aly and M. Youssef. Dejavu: an accurate energy-efficient outdoor localization system. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 154–163. ACM, 2013.
[10] P. Bahl, V. N. Padmanabhan, et al. Radar: An in-building rf-based user location and tracking system.
[11] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur. Scalable wi-fi client self-positioning using cooperative ftm-sensors. *IEEE Transactions on Instrumentation and Measurement*, 2018.
[12] G. Bansal, B. Cheng, A. Rostami, K. Sjoberg, J. B. Kenney, and M. Gruteser. Comparing limeric and dcc approaches for vanet channel congestion control. In *2014 IEEE 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014)*, pages 1–7. IEEE, 2014.
[13] G. Bansal, J. B. Kenney, and C. E. Rohrs. Limeric: A linear adaptive message rate algorithm for dsrc congestion control. *IEEE Transactions on Vehicular Technology*, 62(9):4182–4197, 2013.
[14] G. Bansal, H. Lu, J. B. Kenney, and C. Poellabauer. Embarc: Error model based adaptive rate control for vehicle-to-vehicle communications. In *Proceeding of the tenth ACM international workshop on Vehicular inter-networking, systems, and applications*, pages 41–50. ACM, 2013.
[15] R. Berkvens, A. Jacobson, M. Milford, H. Peremans, and M. Weyn. Biologically inspired slam using wi-fi. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1804–1811. IEEE, 2014.
[16] J.-L. Blanco, J. González, and J.-A. Fernández-Madrigal. A pure probabilistic approach to range-only slam. In *2008 IEEE International Conference on Robotics and Automation*, pages 1436–1441. IEEE, 2008.
[17] C. Bo, X.-Y. Li, T. Jung, X. Mao, Y. Tao, and L. Yao. Smartloc: Push the limit of the inertial sensor based metropolitan localization using smartphone. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 195–198. ACM, 2013.
[18] S. Chairman. Summary of ad-hoc session on study on indoor positioning enhancements for utra and lte. In *3GPP TSG RAN WG1 Meeting*, pages 20–24.
[19] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 233–245. ACM, 2005.
[20] J. Cherian, J. Luo, and S.-S. Ho. Parkloc: Light-weight graph-based vehicular localization in parking garages. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):99, 2018.
[21] J. Djugash, S. Singh, and P. Corke. Further results with localization and mapping using range from radio. In *Field and Service Robotics*, pages 231–242. Springer, 2006.
[22] J. Djugash, S. Singh, G. Kantor, and W. Zhang. Range-only slam for robots operating cooperatively with sensor networks. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2078–2084. IEEE, 2006.
[23] R. Faragher, C. Sarno, and M. Newman. Opportunistic radio slam for indoor navigation using smartphone sensors. In *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 120–128. IEEE, 2012.
[24] B. Ferris, D. Fox, and N. D. Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007.
[25] D. Giustiniano and S. Mangold. Caesar: carrier sense-based ranging in off-the-shelf 802.11 wireless lan. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, page 10. ACM, 2011.
[26] M. A. Goudarzi and R. J. Landry. Assessing horizontal positional accuracy of google earth imagery in the city of montreal, canada. *Geodesy and Cartography*, 43(2):56–65, 2017.
[27] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with rfid technology. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 1, pages 1015–1020. IEEE, 2004.
[28] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR*, 41(1):53, Jan. 2011.
[29] S. He and S. . G. Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys Tutorials*, 18(1):466–490, 2016.
[30] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. Efficient, generalized indoor wifi graphslam. In *2011 IEEE International Conference on Robotics and Automation*, pages 1038–1043. IEEE, 2011.
[31] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai. Verification: Accuracy evaluation of wifi fine time measurements on an open platform. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 417–427. ACM, 2018.
[32] M. Ibrahim and M. Youssef. Cellsense: An accurate energy-efficient gsm positioning system. *IEEE Transactions on Vehicular Technology*, 61(1):286–296, 2012.
[33] Y. Jiang, H. Qiu, M. McCartney, W. G. Halfond, F. Bai, D. Grimm, and R. Govindan. Carlog: A platform for flexible and efficient automotive sensing. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 221–235. ACM, 2014.
[34] Y. Jiang, H. Qiu, M. McCartney, G. Sukhatme, M. Gruteser, F. Bai, D. Grimm, and R. Govindan. Carloc: Precise positioning of automobiles. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 253–265. ACM, 2015.
[35] M. Kim, J. J. Fielding, and D. Kotz. Risks of using ap locations discovered through war driving. In *International Conference on Pervasive Computing*, pages 67–82. Springer, 2006.
[36] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place lab: Device positioning using radio beacons in the wild. In *International Conference on Pervasive Computing*, pages 116–133. Springer, 2005.
[37] R. B. Langley et al. Dilution of precision. *GPS world*, 10(5):52–59, 1999.
[38] X. Liu, S. Nath, and R. Govindan. Gnome: A practical approach to nlos mitigation for gps positioning in smartphones. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 163–177. ACM, 2018.
[39] M. Llombart, M. Ciurana, and F. Barcelo-Arroyo. On the scalability of a novel wlan positioning system based on time of arrival measurements. In *2008 5th Workshop on Positioning, Navigation and Communication*, pages 15–21. IEEE, 2008.
[40] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos. Differences in rssi readings made by different wi-fi chipsets: A limitation of wlan localization. In *2011 International conference on localization and GNSS (ICL-GNSS)*, pages 53–57. IEEE, 2011.
[41] T. Mangel, O. Klemp, and H. Hartenstein. 5.9 ghz inter-vehicle communication at intersections: a validated non-line-of-sight path-loss and fading model. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):182, 2011.
[42] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim. Sail: Single access point-based indoor localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 315–328. ACM, 2014.
[43] P. Moutarlier and R. Chatila. An experimental system for incremental environment modelling by an autonomous mobile robot. In *Experimental Robotics I*, pages 327–346. Springer, 1990.
[44] P. Newman and J. Leonard. Pure range-only sub-sea slam. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1921–1926. Ieee, 2003.
[45] E. Olson, J. J. Leonard, and S. Teller. Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, 31(4):949–958, 2006.
[46] C. Pei, Z. Wang, Y. Zhao, Z. Wang, Y. Meng, D. Pei, Y. Peng, W. Tang, and X. Qu. Why it takes so long to connect to a wifi access point. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.
[47] G. Pulighe, V. Baiocchi, and F. Lupia. Horizontal accuracy assessment of very high resolution google earth images in the city of rome, italy. *International Journal of Digital Earth*, 9(4):342–362, 2016.
[48] J. Rekimoto. Placeengine: A wifi location platform based on realworld-folksonomy. In *Internet Conference 2006*, pages 95–104, 2006.
[49] S. Sen, J. Lee, K.-H. Kim, and P. Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 249–262. ACM, 2013.
[50] Z. Shi and Y. Wang. Neural network based localization using outdoor lte measurements. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2018.
[51] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68, 1986.
[52] A. P. Subramanian, P. Deshpande, J. Gao, and S. R. Das. Drive-by localization of roadside wifi networks. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 718–725. IEEE, 2008.
[53] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 4(3):380–407, 2004.

[54] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016.

[55] T. Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey review*, 23(176):88–93, 1975.

[56] Q. D. Vo and P. De. A survey of fingerprint-based outdoor localization. *IEEE Communications Surveys & Tutorials*, 18(1):491–506, 2016.

[57] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 197–210. ACM, 2012.

[58] J. Wang, N. Tan, J. Luo, and S. J. Pan. Woloc: Wifi-only outdoor localization using crowdsensed hotspot labels. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[59] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, pages 71–84, 2013.