Wi-Fi RTT Ranging Performance Characterization and Positioning System Design

Chengqi Ma, Bang Wu, Stefan Poslad, and David R. Selviah, *Member, IEEE*

Abstract—The aim of this research is to implement a precise Wi-Fi indoor positioning system (IPS) or localization system based upon the IEEE 802.11mc fine-timing measurement (FTM) scheme also known as the Wi-Fi round trip time (RTT) ranging technique, where ranging refers to a sub-process of positioning that determines the distance between a transmitter and receiver. Our system and its algorithms were implemented using a COTS (Commercial-Off-The-Shelf) smartphone and Wi-Fi access points. Experiments were conducted in several real-life indoor environments. This paper presents the detailed Wi-Fi RTT ranging performance of these devices in different system configurations and characterizes the systematic biases and noise model to improve the ranging accuracy. A novel three-step-positioning method is proposed to overcome the issues of no or multiple intersect points in trilateration due to ranging errors to improve positioning accuracy. This consists of the following: 1) systematic bias determination and removal; 2) clustering-based trilateration (CbT) supported by weighted concentric circle generation (WCCG), namely CbT & WCCG; 3) positioning result and trajectory optimization using a Kalman filter. As a result, the evaluation experiments gave a position accuracy of ±1.2 m in 2D static positioning and ±1.3 m for dynamic motion tracking. Also, our CbT & WCCG method demonstrate good tolerance against ranging errors. Moreover, the computational cost and positioning accuracy of CbT & WCCG methods are compared with least square (LS) and recursive least square (RLS) methods and the accuracy standard deviation of our algorithm is the closest to the Cramer–Rao bound (CRB).

Index Terms—Indoor positioning system (IPS), Indoor localization, Wi-Fi-based positioning system (WPS), trilateration, tracking, Wi-Fi fine-timing measurements (FTM), Wi-Fi round-trip time (RTT)

1 Introduction

Indoor Positioning Systems (IPS) are increasingly needed as we spend more time indoors (at least in urban areas) and as indoor spaces become far more complex. IPS applications include navigating to find a physical asset in unfamiliar spaces such as a book, a shopping item, a person or exit.

There is no standard positioning system for indoors unlike for outdoors (Global Positioning System, GPS). There are further challenges for IPS, compared to outdoors, as we need a better accuracy, e.g., 1.7 m (a typical adult arm span) to enable us to find and reach to get things in indoor spaces that are more densely cluttered than outdoor spaces. There are no global maps, indoor spaces also often tend to be more 3D rather than 2D and furniture and objects are often moved around changing the free path for navigation.

A common infrastructure found in most indoor environments, IEEE 802.11, that is used for local area network communication can also be used for positioning – a Wi-Fi-based positioning system (WPS), which is the focus of this paper. Classic WPSs are based on three methods. The first is the traditional path loss model-based method. This uses the Received Signal Strength Indicator (RSSI) to calculate distances between an Access Point (AP) and a Mobile Terminal (MT). Then, trilateration is applied to estimate the terminal's location [1]. However, this method suffers from the issue of instability of RSSI, caused by path interference, small scale fading and shadowing, which are easily affected by different surroundings [2]. The second method is known as a fingerprinting-based technique [3], [4] which is generally considered to be more accurate than the path loss model based method [5]. This method typically includes two phases. The first phase, called the offline survey phase, is to build a fingerprint database or a radio map consisting of RSSI information from multiple APs collected at known locations. The second online localization phase determines user positions via matching radio knowledge of their current position with the pre-established radio map. The third method makes use of Wi-Fi Channel State Information (CSI) (e.g., amplitude and phase) to realize localization [6]. Compared to RSSI which only provides signal strength level, the CSI of each channel is a multi-dimensional vector containing the information from all subcarriers (usually over 30 subcarriers in each channel), which has many more features [7]. CSI-based positioning methods include CSI fingerprinting, Angle of Arrival (AOA) and Time of Flight (ToF) or Roundtrip Time of Flight (RToF). They are all sensitive to environmental changes and surrounding movements, which influence the positioning accuracy. Also, ToF requires highly accurate synchronization between all parts of the system while RToF is free of such a requirement. CSI fingerprinting also faces the

- *Chengqi Ma and David R. Selviah are with the Electronic and Electrical Engineering Department, University College London, Gower Street, WC1E 6BT London, U.K.*
  E-mail: {chengqi.ma.16, d.selviah}@ucl.ac.uk.
- *Bang Wu and Stefan Poslad are with the Electronic Engineering and Computer Science Department, Queen Mary University of LondonMile End Road, E1 4NS London, U.K.*
  E-mail: {bang.wu, stefan.poslad}@qmul.ac.uk.

Manuscript received 21 Oct. 2019; revised 15 June 2020; accepted 24 July 2020. Date of publication 28 July 2020; date of current version 7 Jan. 2022.
(Corresponding author: Bang Wu.)
Digital Object Identifier no. 10.1109/TMC.2020.3012563

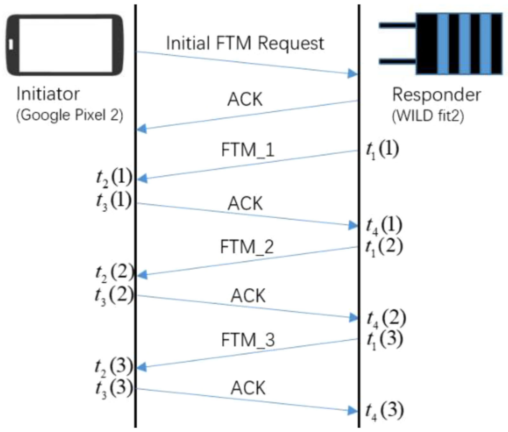


Fig. 1. RTT measurement illustration [13].

problem of intensive computational workload for building and maintaining the fingerprint database as well as the challenge of managing substantial amounts of storage securely. These three WPS methods have been researched and developed extensively [8], [9], [10], [11], [12]. The second of these three methods based upon fingerprinting seems to be the most accurate and widely used. However, fingerprinting is not scalable because it requires a radio-map to be built in each new scenario and even its location accuracy is affected by dynamic object occlusion such as people moving in that space.

In this paper, we focus on one of the latest techniques called the Fine-Timing Measurement (FTM) [13] scheme based ranging and localization. The FTM scheme has been supported by the IEEE 802.11mc standard since 2016 [13], [14]. FTM can be understood as a RToF measurement approach, also called Wi-Fi Round-Trip Time (RTT), which allows a MT to determine its distance from an AP by measuring the duration of a radio wave transmission time frame traveling back and forth between the transmitter and receiver, which are generally called the initiator and responder. This method can give even more accurate measurements by including the timestamp recorded at both the initiator and responder with nanosecond resolution [14].

Fig. 1 shows the RTT measurement approach overview (Detailed descriptions of the initiator, Google Pixel 2 [15], and responder, WILD fit2 [16], are given in Section 3.1). The process begins when an Initial FTM Request is broadcast from the initiator. The system records the timepoint,t_1, when the Fine-Timing Measurement, FTM signal is emitted from the responder and the timepoint, t_2, when the signal arrives at the initiator. After the internal processing at the initiator, an acknowledgement (ACK) is sent at timepoint, t_3, from the initiator which reaches the responder at timepoint, t_4. To improve the measurement accuracy, the round-trip time, RTT is determined by an average of, n, round trips [14].

$$RTT = \frac{1}{n}\left(\sum_{K=1}^{n} t_4(K) - \sum_{K=1}^{n} t_1(K)\right) - \frac{1}{n}\left(\sum_{K=1}^{n} t_3(K) - \sum_{K=1}^{n} t_2(K)\right). \tag{1}$$

Then, the distance between the initiator and the responder, d, can be calculated using $d = RTT \cdot c/2$, where, c, is the speed of electromagnetic wave propagation. The initiator and responder clocks do not need to be synchronized as time differences between readings taken by the same clock are calculated. However, if the clocks are not stable and are subject to drift, reference [13] describes a correction including clock offset, eq. (2), to further reduce the ranging errors.

$$v\ (t_i) = v(t_{i-1}) + \dot{v}\cdot(t_i - \ t_{i-1}), \tag{2}$$

where $v(t_i)$ is the clock offset relative to the true signal arrival time at current timepoint, t_i. $\dot{v}$ is the clock skew or the rate of change of the clock offset. In this paper, we do not apply this time offset cancelation as we analyzed the raw data characteristics from the initiator (mobile phone). We will consider this in future work.

FTM is considered to be the next generation of Wi-Fi based IPS [17]. The FTM protocol is supported by state-of-the-art Wi-Fi chipsets [18] and all Android OSs later than Android Pie (version 9.0) also support Wi-Fi RTT signal processing functions. It means that commercial off-the-shelf (COTS) MTs can be used in a Wi-Fi RTT based positioning system without any extra hardware (on the receiver side). Although the fundamental principle of Wi-Fi RTT measurement has been deeply investigated [19], as far as we are aware, there has been little research focusing on applying this technology to realize a positioning system.

Investigation of the performance of Wi-Fi RTT devices in real-life environments is extremely helpful to gain a better understanding of this technology and to motivate a variety of potential applications. Hence, in this research, our main contributions are:

1. Quantifying the detailed Wi-Fi RTT ranging performance in a variety of working modes and environments, which includes determination of Systematic Bias, the Minimum Ranging Interval Limitation and Maximum Stable Working Range. Note such RTT performance data is not currently part of (AP and smartphone) device manufacturer's data sheets; we recommend that it will be useful to include these in future. (see Section 3.3)
2. Proposing, demonstrating and proving the benefits of a new systematic bias removal process by applying it with existing least-square positioning methods (see Section 3.4).
3. Proposing a novel Weighted Concentric Circle Generation (WCCG) method to overcome issues of no or multiple intersect points in trilateration when using Wi-Fi RTT to implement an IPS. (see Section 4.2)
4. Developing a new Clustering-based Trilateration (CbT) method for the Wi-Fi RTT based IPS to improve its accuracy and reproducibility. This method can be applied for static positioning for the case when enough ranging measurements exist. It can also be applied along with WCCG for dynamic motion tracking. (see Section 4.3)
5. Implementing and demonstrating an IPS based on Wi-Fi RTT using commercial devices (APs and smartphones) with the above new techniques. (See Section 5)

This paper is structured as follows: Section 2 surveys ranging technologies, positioning methods and issues in traditional IPSs. Section 3 presents our first main contribution which is the detailed Wi-Fi RTT ranging performance in an indoor office-type room and the systematic bias removal process. Section 4 introduces our new WCCG & CbT positioning algorithms. Section 5 presents the results of static positioning and dynamic motion tracking to demonstrate the performance of our design of Wi-Fi RTT-based IPS and compares these with the best existing positioning methods and with Cramer-Rao Bound (CRB). Finally, the last Section 6 gives the conclusions and future work.

2 Related Work

Position determination or localization involves several subprocesses. Ranging is the process that determines the distance between two things, e.g., a transmitter and receiver. There are different ranging methods but two of the most common are RSSI (e.g., traditional Wi-Fi and Bluetooth Low Energy, BLE) and Time of Flight (ToF) or Roundtrip Time of Flight (RToF) (e.g., Global Positioning System, GPS, Wi-Fi Round-Trip Time, RTT). To determine the 2D or 3D position in space usually requires comparing distance measurements from 1 receiver with respect to 3 or more transmitters. Trilateration often uses 3 transmitters to calculate a position in a 2D space (the focus of this paper). Multilateration can be regarded as a more general case of trilateration. A further distinction can be made between these terms, when the ToF is multiplied by the propagation speed, it is termed a pseudo-range [20] (as used by multilateration) in contrast to the true range (as used in trilateration).

2.1 Types of IPS

Many indoor positioning systems have been developed based on different technologies such as Wi-Fi [21], Bluetooth Low Energy (BLE) [22], Magnetic Field (MF) [23], Pedestrian Dead Reckoning (PDR) [3], Radio Frequency Identification (RFID), Ultra-wideband (UWB), Light Detection and Ranging (Lidar) [24], [25], [26], [27], [28], visible light [29], computer vision [30], acoustic base systems [31] and Li-Fi [32], etc. Visible light and Lidar technology have a particularly high ranging accuracy (millimeter level). However, they are not suitable for an IPS when the surroundings are changing often or with crowds of moving people. The main reason is that the ranging process requires a Line-of-sight (LOS) between transmitter and receiver. MF and PDR based IPS, are both based on micro-electromechanical systems (MEMSs). The MF IPS faces the same issue as fingerprinting system requiring the establishment of a radio map, while PDR has the problem of error accumulation over time due to a drift in the accuracy of the sensors. The acoustic IPS, gives accurate positioning, but the short range of acoustic signals and the effect of background sound noise make it unsuitable for a ubiquitous localization system. Thus, we will discuss other radio frequency (RF) based IPSs (Wi-Fi, RFID, BLE and UWB) in detail, which are more practical.

When implementing an IPS, positioning accuracy, applicable environments, and hardware costs are three principal factors to be considered. So, we present in Table 1, a review and comparison of the most popular IPSs based on different technologies.

TABLE 1
IPSs Based on Different Technologies

	IPS accuracy (plus and minus)	Signal range (Indoor, LOS)	Hardware cost (minimum requirement)
Wi-Fi [33], [31]	1.8 - 6 m	100 m	£30 - 50
RFID [31]	< 10 cm	1 - 2 m	£10 - 30
BLE4.0 [31]	1 m -2.5 m	70-100 m	£20 - 40
UWB [31]	< 10 cm	10 m	£40 - 60

In Table 1, each of these technologies has different strengths and weaknesses. An important point is that for almost all of these technologies additional hardware has to be installed in the environment with additional costs, except for Wi-Fi for which access points are already widely installed in most buildings for local wireless network services. As for the user side, BLE and Wi-Fi are integrated into smartphones whereas RFID and UWB are currently not but will need to be if they are to be used for positioning. Due to the small signal ranges indoors, RFID and UWB are not suitable for large indoor spaces although they have much higher accuracy. Consequently, Wi-Fi RTT as the latest technology in Wi-Fi devices seems to be a promising choice for more accurate IPS without additional hardware installation costs.

2.2 Position Estimation

2.2.1 Trilateration

Trilateration positioning [34] uses three distance measurements to transponders whose positions are known, to determine the two-dimensional $(x,\ y)$ coordinates of an unknown position. It is necessary to obtain the distances between the Access Points, APs and the Mobile Terminal, MT (carried by a user) to determine the location of users.

Fig. 2, shows the two-dimensional (x, y) plane. The positions, O_i at $(x_i,\ y_i)$ are the locations of the AP$_i s$, where i is the AP index. P is the position of the MT at the intersection of three ranging circles with radii, r_1, r_2, r_3 being the respective distances to the three APs calculated according to eq. (3). The position of $P(x, y)$ can be calculated by eq. (4).

$$\begin{aligned} r_1 &= \sqrt{(x-x_1)^2+(y-y_1)^2} \\ r_2 &= \sqrt{(x-x_2)^2+(y-y_2)^2} \\ r_3 &= \sqrt{(x-x_3)^2+(y-y_3)^2} \end{aligned} \tag{3}$$

$$\begin{bmatrix} r_1^2-r_2^2+x_2^2-x_1^2+y_2^2-y_1^2 \\ r_1^2-r_3^2+x_3^2-x_1^2+y_3^2-y_1^2 \end{bmatrix} = 2\begin{bmatrix} x_2-x_1 & y_2-y_1 \\ x_3-x_1 & y_3-y_1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}. \tag{4}$$

Trilateration [35] can be based on different ranging principles, e.g., RSSI, ToF and RToF.

The free-space path-loss model is widely used to estimate the distance between a receiver and a transmitter,

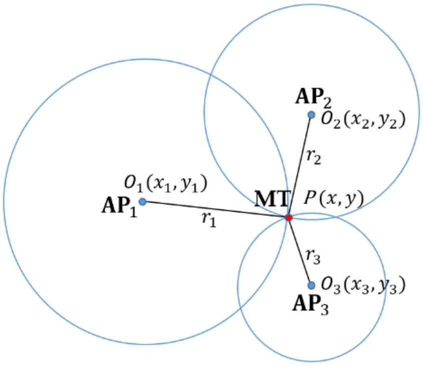


Fig. 2. Trilateration principle.

which is known as the log-distance path loss model [12]. The RSSI-based positioning system is popular in wireless sensor network (WSN) node localization research [36]. However, in Wi-Fi RSSI based applications, due to the complexity of 3D surroundings, the value of RSSI can be affected easily by nearby static or moving objects made from different materials, shadowing effects and multipath effects [2]. All these lead to inaccurate ranging results and poor positioning accuracy. Consequently, RSSI based WPS tends to be fingerprinting-based positioning systems.

Time of Flight, ToF means the time taken for a signal to travel from a transmitter to a receiver. In the ToF and RToF methods r_1, r_2 and r_3 are measured and eq. (4) is used to find the (x, y) coordinates. The distance is calculated by multiplying the speed of light by the time difference between the time when a signal transmitted arrives at a receiver. Wi-Fi device based ToF [19] requires special hardware calibration and synchronization to give accurate range measurements, which makes it costly. Wi-Fi RTT introduced in Section 1, directly obtains ranges, and can be easily implemented in standard commercial Wi-Fi APs.

2.2.2 Trilateration Issues

In ideal trilateration, all ranging circles should intersect at a single point. However, because of ranging errors, they can either have more than one intersection point or no intersection point. In practical use, there can be more than three APs to obtain multiple ranging circles which will result in more complicated intersection cases. Many methods [10], [37], [38], [39], [40], [41] have been proposed to calculate the optimal estimate of position from three ranging circles. In the case when multiple intersection points are calculated, statistical models can be used to quantify the uncertainty of the position estimation [37]. If an area intersected by multiple circles can be calculated, the most common way is to calculate the centroid as an estimation of the position [10].

For some special situations with multiple intersecting points, [38], [39], [40] proposed specific solutions. [38] proposed a method called a Line Intersection Algorithm. It deals with a situation where three lines determined by three pairs of circles can intersect at a single point and this point is considered to be the estimated result. [39] directly took the middle point of the shortest line formed by each pair of intersection points as the estimation result. The Closest Point Algorithm proposed in [40] chose a point from the smallest circle as the estimation result. [41] recommended to use a circular annulus ($\mathrm{radius} = \mathrm{r}_{raw} \pm error$) based on a raw ranging measurement and its positive and negative error. Three circular annuli are more likely to have an intersection area. Then, the centroid of all points intersected by six circles (each circular ring has two circles) is viewed as the estimation value.

Although all these methods proposed solutions to improve the trilateration performance, they can only deal with specific intersection situations. A more general intersection method is required. Thus, we propose a novel method called Clustering-based Trilateration (CbT) to deal with the trilateration intersection issues to improve the positioning accuracy.

2.2.3 Least Square Estimation

The traditional positioning estimation method for trilateration is a Least Squares (LS) [42] estimation, which minimizes the sum of the squares of the residuals of each calculation. The aim of the LS method is to estimate, Θ, depending on $\Psi = \Phi\Theta$. The matrix form can be expressed as,

$$\begin{bmatrix} \psi_1 \\ \vdots \\ \psi_k \end{bmatrix} = \begin{bmatrix} \phi_1^T \\ \vdots \\ \phi_k^T \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \tag{5}$$

where Θ is the parameter vector to be estimated, representing coordinates in the positioning (can be 2D or 3D). $\phi_i^T = [\alpha_1^i \quad \ldots \quad \alpha_n^i]$ is the i-th coefficient of Θ, ψ_i is the i-th corresponding result. Eq. (4) is an example of eq. (5), e.g., $\theta_1 = x$, $\theta_2 = y$ and Ψ is the left side of eq. (4). According to LS, the estimation of Θ (noted by $\hat{\Theta}$) has a solution:

$$\hat{\Theta} = \left(\Phi^T\Phi\right)^{-1} \Phi^T\Psi. \tag{6}$$

The above solution is applied when $\Phi^T\Phi$ is not singular. If not, singular value decomposition (SVD) should be used to calculate $\hat{\Theta}$ [43].

2.2.4 Recursive Least Square Estimation

Other than research focused on finding the intersections of ranging circles, an efficient and well-known method to estimate the MT location is called Recursive Least Square (RLS) estimation [44]. RLS is a popular and practical algorithm used extensively in signal processing, communications, and automatic control systems [45], [46]. Compared to the LS method, the RLS algorithm minimizes the sum of the squares of the residuals in an online and efficient manner without repeating the least squares estimation at each step.

Based on eqs. (5) and (6), let, $\hat{\Theta}_k$ be the position estimation at current step, k. As the RLS method recursively estimates the parameter, $\hat{\Theta}_k$ is deduced from the previous step, $k-1$, $\hat{\Theta}_{k-1}$, namely, $\hat{\Theta}_k = f(\hat{\Theta}_{k-1})$. The recursive equations for RLS are:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + K_k\epsilon_k \tag{7}$$

$$\epsilon_k = \psi_k - \phi_k^T\hat{\Theta}_{k-1} \tag{8}$$

$$K_k = P_k\, \phi_k \tag{9}$$

$$P_k = P_{k-1} - \frac{P_{k-1}\phi_k\phi_k^T P_{k-1}}{1+\phi_k^T P_{k-1}\phi_k} \tag{10}$$

$$P_{k-1} = \left(\Phi_{k-1}^T\Phi_{k-1}\right)^{-1}. \tag{11}$$

Here, $\hat{\Theta}_0$, Φ_0 should be given.

Numerous publications on the solution of the trilateration range equations have been published. [45] presented a good example of RLS for trilateration positioning with low computational complexity. The algorithm is applied to real measurement data of a UWB indoor positioning system in 3D space.

2.3 Geometry Impact on Positioning Accuracy

Geometry impact is the deployment of AP nodes at known locations impacts the localization performance [47]. The positions of AP nodes are also known as landmarks. Two main factors influence the positioning performance namely, the deployment geometry and the number of landmarks. An optimized placement design can efficiently minimize the localization error. It is better to avoid the error caused by geometry impact, rather than try to compensate for this later.

According to [47], it claims that for landmark deployment, a square shape (4 landmarks) plus one node at the center of the mass has been found to be an optimal deployment. As regards the quantity of nodes, a small number of nodes can achieve a good positioning result when there is enough coverage of the environment. In [48], optimal 3D landmark placement research is presented using the Fisher information matrix to find the best positions to place sensors to give the best localization result.

2.4 System Performance Evaluation

We assessed the performance of our new algorithm in terms of average error, standard deviation, statistical analysis and compared our performance with the best existing well-known methods and with the Cramer-Rao Bound (CRB) [36] also known as Cramer-Rao lower Bound (CRLB).

To calculate the CRB, let A be the estimation parameter; x, σ be the estimation and standard deviation of unbiased methods. As multiple unbiased estimators are employed, the likelihood function is the product of individual probability distribution functions, pdfs as shown in equation (12)

$$\mathrm{p}\ (x;\ \mathrm{A}) = \prod_{i=0}^{n} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{1}{2\sigma_i^2}(x-A)^2\right], \tag{12}$$

Where, n, is the number of unbiased estimators. σ_i is the standard deviation of the *i*-th estimator. According to the CRB definition, for any unbiased estimators, the variance of A should satisfy the condition,

$$\mathrm{var}(\mathrm{A}) \geq \frac{1}{I(\mathrm{A})} = \frac{1}{-E\left[\frac{\partial^2 \ln \mathrm{p}(x;\ \mathrm{A})}{\partial A^2}\right]} = \frac{1}{\sum_{i=0}^{n}\frac{1}{\sigma_i^2}}, \tag{13}$$

where $I(\mathrm{A})$ is the Fisher information criterion [49]. When the variance of an unbiased estimator equals $\mathrm{var}(\mathrm{A})$, it has the lowest mean squared error among all unbiased methods [50].

CRB is widely used in wireless sensor network (WSN) node localization performance evaluation [51], [52]. In an application of a RSSI based ranging measurement localization system [36], the authors proposed a novel iterative tree search algorithm (I-TSA) in comparison with the maximum likelihood estimator (MLE) and multidimensional scaling (MDS) and proposed CRB as a performance reference to show the advantages and limitations of proposed new algorithms and systems. Similarly, [52] introduced a Quantized Cramer Rao Bound (Q-CRB) method to adapt the CRB, to characterize the behavior of location errors of the LS position estimation for various system parameters, e.g., granularity levels, measurement accuracies, and localization boundaries.

3 Wi-Fi RTT Ranging Performance Analysis and Ranging Correction

3.1 Wi-Fi RTT Initiators and Responders

RTT ranging technology requires an initiator, also called a sender, to initiate RTT ranging callbacks and a responder to act as ranging target (Fig. 1). A MT with a RTT supported wireless card was programmed as an initiator and a Wi-Fi AP that supports the IEEE 802.11mc protocol was configured as a responder. In our experiments, the Google Pixel 2 [15] smartphone was chosen as the testbed for the initiator and the Wi-Fi Indoor Location Device (WILD) fit2 [16] was chosen as the responder, because they fully support the RTT technique.

If an Android smartphone is programmed as an initiator, the Android Operating System (OS) has currently a minimum requirement of version 9.0 (Android Pie) of the OS together with a SDK version greater than 28. Note that for all OSs later than Android Pie, Google has currently limited the scanning rate for Wi-Fi APs broadcasting probe requests to 4 times every 2 minutes in active scanning mode and only 1 time every 30 minutes in passive scanning mode, which is called ‘scan throttling’ [53], [54]. Due to this limitation on scanning frequency, it is difficult to realize a precise IPS using Wi-Fi RSSI based IPSs. Fortunately, the RTT ranging approach is an independent callback between the MT and AP, which has no such limitation.

WILD fit2 [16] is a ready-to-use Wi-Fi RTT gateway released by Compulab,[1] supporting the IEEE 802.11mc protocol. A WILD fit2 AP can be configured as a responder to work in both 2.4 GHz and 5 GHz bands with three different channel bandwidths (20 MHz, 40 MHz, and 80 MHz). Our experiments show that the RTT ranging result and signal characteristics can be quite different for different channel bandwidths. If multiple responders set up ranging callbacks with the same initiator simultaneously, the initiator can only process ranging callbacks in a serial manner. It means that it will take at least *m* times the callback processing interval to obtain all ranges, where *m* is the number of connected responders. Also, it is important to assign responders different channel numbers to avoid RTT processing conflicts. In this paper, we used four WILD fit2 APs (with knowledge of their position coordinates) to implement our positioning system which could be extended to higher numbers of APs

[1] https://www.compulab.com/

MAC Address	Ranging Result	Ranging Standard Deviation	RSSI	Number of Range Calculation Trials	Number of Successful Range Calculations	Time Stamp

Fig. 3. RTT data package structure.

to investigate if the system performance can be improved in future work.

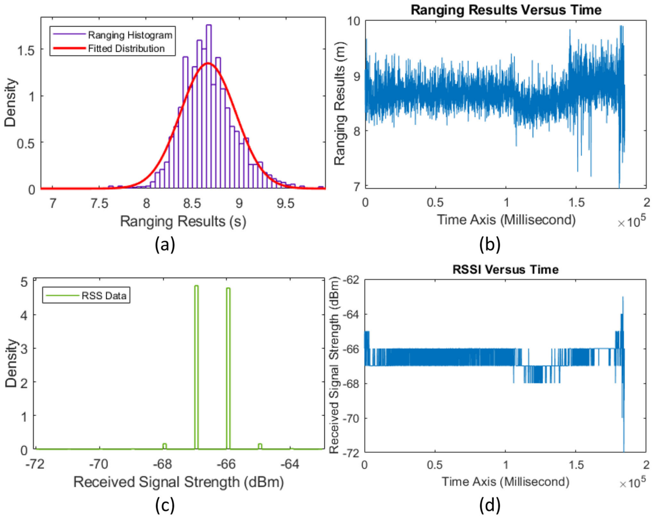
3.2 Wi-Fi RTT Ranging Data

The RTT Ranging data package is used at the initiator and each package includes 7 parts in the Android OS. The data package structure is shown in Fig. 3.

The MAC address is the identity of the responder. The ranging result is the average distance between the initiator and responder in units of millimeters calculated by the MT OS. It is calculated from 7 successful ranging attempts and the standard deviation is calculated from these results as well. The RSSI in units of dBm is the signal level of the ranging callbacks. The number of range calculation trials and the number of range calculations are recorded as well. Finally, the time stamp (generated from the Android OS) records the time at which the data package was generated, which is recorded with an accuracy of milliseconds.

We conducted tests to find the basic RTT ranging accuracy and characteristics. Both initiator and responder were placed at fixed positions and there was a clear line-of-sight between them. The test environment was a normal office-type indoor environment, where both MT and AP were placed at a 1.4 m height supported by plastic tripods.

Fig. 4 shows typical examples of the basic characteristics of the RTT ranging results and RSSI obtained. Fig. 4a is a histogram of the RTT ranging results, where 2700 data package were recorded over 180 seconds giving a normal distribution. Fig. 4b shows the RTT ranging results chronologically during the 180 seconds of data collection, showing that the ranging results did not drift with time. A glitch appears after a few seconds which is caused by manual operation of the MT. The corresponding RSSI results are shown in Figs. 4c and 4d showing that the signal level is very stable and with no interference or environmental change. The noisy head and tail in Fig. 4d are caused by manual operation of the initiator. It shows very stable RSSI values during the test, but it is sensitive to interference for example slightly touching the screen of the MT.



(a)

(b)

(c)

(d)

Fig. 4. Example of basic characteristics of RTT ranging results and RSSI. (a) Ranging results histogram, (b) Chronological ranging results within 180 seconds testing time, (c) RSSI histogram, (d) Chronological RSSI within 180 seconds testing time.

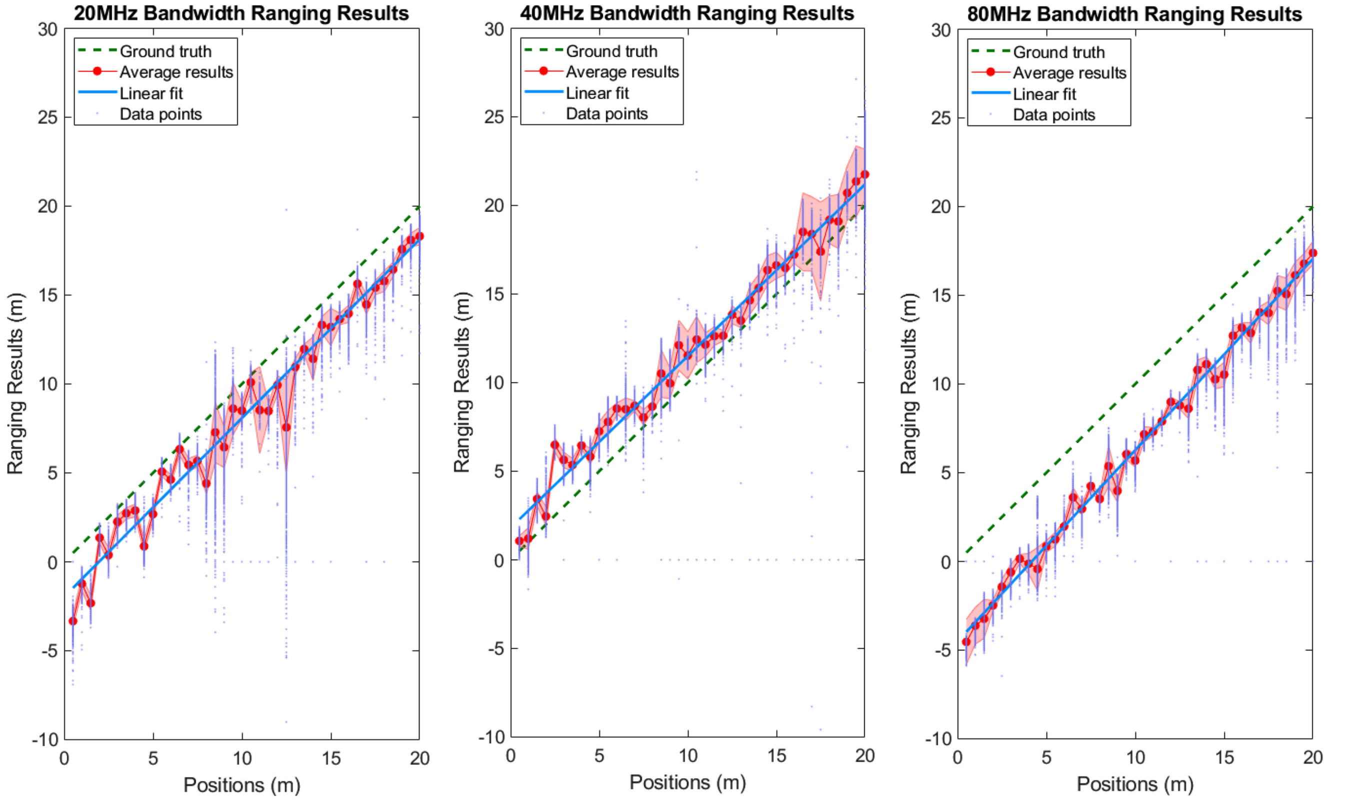
TABLE 2
RTT Ranging Performance Indicator in Different Bandwidths

Bandwidths		20 MHz	40 MHz	80 MHz
Minimum Ranging Interval Limitation (Maximum Scanning Frequency)		41 ms (24 Hz)	38 ms (26 Hz)	35 ms (29 Hz)
Maximum Stable Working Range		25 m	20 m	14 m
Linear Fitted Slope, s		1.007	0.976	1.079
Systematic Bias, e_{sys}		-1.949 m	1.776 m	-4.501 m
RMS Error of Ranging Results (without systematic bias), e_{RMS}		1.018 m	0.821 m	0.517 m
Error Probability Distribution Function (Ranging Noise Model) n_m	Mean, μ	-4.2e-16 m	1.7e-15 m	8.3e-17 m
	Standard Deviation, σ	1.073 m	0.651 m	0.273 m

3.3 Wi-Fi RTT Ranging Performance for Different Channel Bandwidths

In tests of ranging performance using different channel bandwidths, we defined and tested two important characteristics of Wi-Fi RTT ranging: The Minimum Ranging Interval Limitation and the Maximum Stable Working Range. First, the minimum ranging interval limitation is the shortest interval between RTT ranging requests that the initiator can achieve in our testbed. ‘Ranging failed’ reports are automatically generated by the Android OS if the ranging request is sent sooner than this limitation. Second, the maximum stable working range means that, within a certain distance, the callback link can be established and the rate of ‘ranging failed’ reports is less than 20 percent (experimental based empirical value). If the initiator and responder are placed apart further than the stable working range, it is hard for the responder to be detected by the initiator. When the callback link is established successfully and then the initiator and responder are moved further apart than the stable working range, the RTT responder is still capable of providing ranging results but ‘ranging failed’ is constantly reported. Both characteristics depend on the bandwidths used. Detailed test results are listed in Table 2.

A Wi-Fi AP can be configured to work in one of three channel bandwidths: 20, 40, 80 MHz. This is the same for RTT supported APs. The official document from Android gives a statement [55] about the ranging accuracy in different channel bandwidths as *“a range estimate is expected to have the following tolerances: 80 MHz: 2 meters, 40 MHz: 4 meters, 20 MHz: 8 meters”*. To investigate the ranging performance, we designed experiments to place the initiator at fixed spacings every 0.5 meter from 0 to 20 meters along a straight line between the initiator and responder. Three groups of tests were conducted for 20, 40 and 80 MHz channel bandwidths, respectively. At each test position, the initiator collected data for 5 minutes with the maximum scanning frequency listed in Table 2.



(a) (b) (c)

Fig. 5. RTT ranging test results in different bandwidths, (a) 20 MHz, (b) 40 MHz, and (c) 80 MHz.

Figs. 5a, 5b and 5c shows the ranging performance of the 20, 40 and 80 MHz channel bandwidths, respectively. The x-axis is the true distance and the y-axis shows the measurement results. In each of these figures, the red points are the average of ranging results at one test position while the red shaded regions show the standard deviations. The small purple dots are the raw ranging data. The dashed green line is the ground truth and the blue line is the linear fit to the average ranging results. All the lines of linear fit are almost parallel to the dashed lines of ground truth, which means that the differences between the blue line and dashed green line can be identified as a systematic bias. The equation of the linear fit is

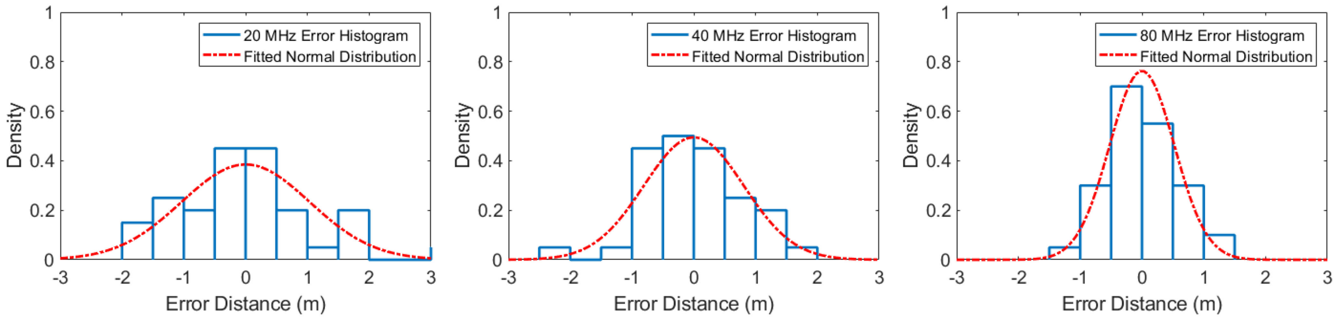
$$r_m = \ s \cdot r_t + e_{sys}, \tag{14}$$

where r_m is the measured ranging result, r_t is the ground truth, s is the slope of the linear fit line and e_{sys} is the systematic bias. The calculated parameters in eq. (14): slope, s, and systematic bias, e_{sys}, are listed in Table 2. The systematic bias can be removed using eq. (15) derived from eq. (14)

$$r = r_t = \frac{r_m - \ e_{sys}}{s}, \tag{15}$$

where r_m is the ranging result, and r is the optimal estimate of range without systematic bias. After removal of the systematic bias, the Root Mean Square (RMS) error of ranging results,e_{RMS}, in different bandwidths has been calculated (as shown in Table 2), which indicates the accuracy of the ranging without systematic bias.

We also plotted histograms of the errors at all test points after removal of the systematic biases, then fit normal distributions to them as shown in Figs. 6a, 6b and 6c for 20, 40 and 80 MHz bandwidths, respectively. All histogram bars



(a) (b) (c)

Fig. 6. Error probability distribution function after removal of systematic biases in (a) 20 MHz, (b) 40 MHz, and (c) 80 MHz bandwidths.

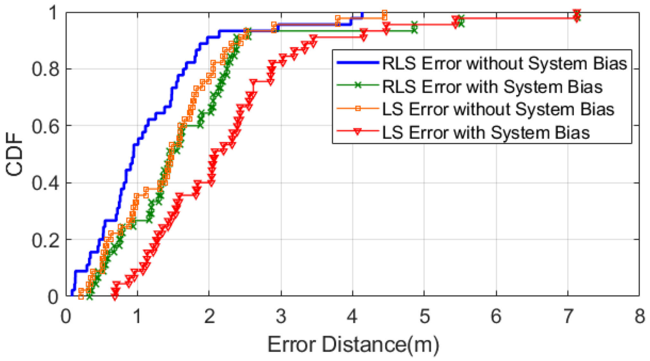


Fig. 7. Error CDF comparison between with and without systematic bias.

TABLE 3
Effect of Systematic Bias Removal

	Average Error	Standard Deviation	90% CDF	60% CDF
Recursive Least Square	1.165 m	0.881 m	1.975 m	1.113 m
Recursive Least Square with bias	1.744 m	1.302 m	2.389 m	1.622 m
Least Square	1.494 m	0.881 m	2.436 m	1.600 m
Least Square with bias	2.284 m	1.238 m	3.451 m	2.395 m

are set to have the same width in meters. The fitted result, red dashed curve, gives a distribution of $n_m \sim N(\mu, \sigma^2)$, where n_m is the error, μ is the mean,σ is the standard deviation. Detailed data for these three distributions is listed in Table 2. In Fig. 6, the fitted normal distribution is the Gaussian noise model of ranging. So, eq. (14) can be rewritten as

$$r_m = s \cdot r_t + e_{sys} + n_m. \tag{16}$$

It can be concluded that the ranging performance for different channel bandwidths is quite different from each other. For different types of environment and conditions, the responders should be configured to have the most suitable channel bandwidth. For example, 20 MHz is best for large indoor open spaces because it has the largest stable working range; In the case of large numbers of responders in small environments, 80 MHz is more suitable because the small ranging interval limitation can give high ranging data collection rate. 40 MHz is best as a default setting. Therefore, the authors selected the 40 MHz channel bandwidth as the configuration for all the following experiments.

3.4 Systematic Bias Removal Test

The systematic bias removal process was demonstrated and validated by applying it to a positioning system based on the trilateration principle of eqs. (3) and (4). The positioning algorithms were chosen to be two well-known methods Least Square (LS) and the Recursive Least Square (RLS) as described in Sections 2.2.3 and 2.2.4. We compared the positioning results estimated from the raw ranging data and the results estimated from ranging results after the systematic bias removal process.

In this validation test, 4 APs were placed in a square pattern in a 12 $m \times 8$ m room and the test was repeated 48 times at each of 12 different test positions. At each test position, the MT was oriented horizontally in each of four directions for each of four individual tests. Fig. 7 shows the Cumulative Distribution Function, CDF of errors to compare the effect of systematic bias removal with the LS and RLS methods, respectively. The blue thick curve and green '×' marked curve are the RLS positioning result without and with systematic bias, respectively. The orange '□' marked curve and red 'Δ' marked curve represent the LS result without and with systematic bias, respectively. The CDF curves directly show that our bias removal process is highly effective in reducing positioning error.

We also calculated the average error, standard deviation, 90 percent CDF and 60 percent CDF for each test, as shown in Table 3. With systematic bias removal, the average error is reduced from 1.744 m to 1.165 m for RLS; and is reduced from 2.284 m to 1.494 m for LS. The standard deviation for both methods decreases to 0.881 m. Also, the accuracy is improved for both algorithms at 60 percent CDF (RLS: by 0.509 m, LS: by 0.795 m) and at 90 percent CDF (RLS: by 0.414 m, LS: by 1.015 m).

Consequently, we conclude that our new systematic bias removal process is highly effective in improving the positioning accuracy when it is applied with two existing well-known positioning methods. In the following Section 4, the ranging performance indicators, e_{sys}, e_{RMS}, μ and σ in Table 2 will be used to compare with our new positioning WCCG and CbT algorithms.

4 Wi-Fi RTT Positioning System and Core Algorithms

4.1 System Framework

Many characteristics of Wi-Fi RTT ranging were measured in real-life experiments in the last section. In this section, the authors present a design for a new positioning system.

We designed a three-step positioning approach as shown in Fig. 8. Step one is a data collection and pre-processing step, where raw data is recorded, and the systematic bias is removed. According to the data in Table 2, the systematic bias can be removed if the working bandwidth of the APs are known to the initiator (the working bandwidth information can be obtained from the normal broadcasting Wi-Fi probe request package). Then, step two is the positioning approach. This step combines a process of Weighted Concentric Circle Generation (WCCG) to deal with the issue of

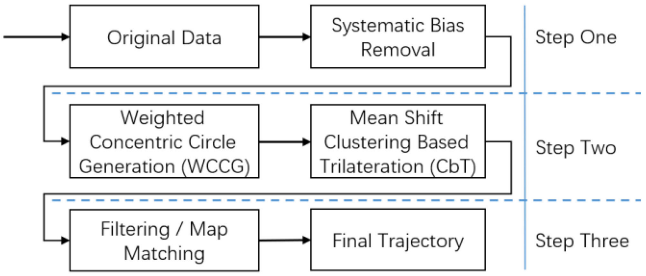


Fig. 8. New positioning system flow chart.

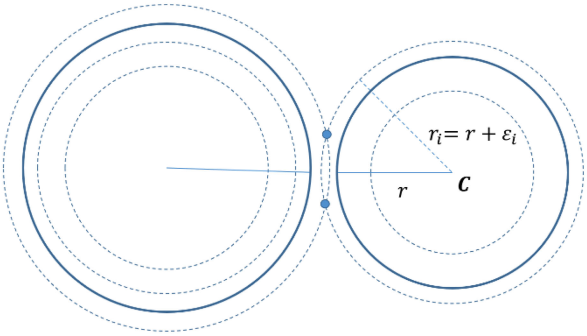


Fig. 9. Illustration of Weighted Concentric Circle Generation (WCCG) to solve the no intersection issue.

no or insufficient intersection points with the following process of mean shift CbT. Step three is the process of positioning result optimization. A digital filter, for example, a Kalman filter, or another algorithm such as map matching is applied to improve the positioning results, especially for dynamic motion trajectory tracking.

4.2 Weighted Concentric Circle Generation (WCCG)

In a two-dimensional Euclidian geometry, the location can be found by using trilateration to calculate the intersection points of circles using the known positions of the circle centers with the corresponding measured ranges as radii. Two circles have three cases of intersection, which are: no actual intersection, one single degenerate point and two distinct points. Assuming we have M number of ranging results from different responders to one initiator, then the number of intersection points should be $2\ ^M C_2 = M(M-1)$ which includes both actual and imaginary points. In our case, as we have four responders, $M = 4$, so twelve (six pairs of) intersection points can be determined from a single ranging data package.

In an ideal situation, when the range measurements are stable and sufficiently accurate, each pair of ranging circles give at least one actual intersection point. However, due to ranging errors, the intersection points can be calculated either at wrong positions or as no actual intersection point. In our case, the no intersection case can often happen after removal of the systematic bias (1.776 m in the 40 MHz bandwidth case), which leads to lack of intersection points for the following positioning process. Furthermore, in the 20 and 80 MHz cases, the negative systematic biases (if there is no bias removal process) can make the ranging result always shorter than the true distance. This may lead to a problem that even with systematic bias removal, there is still no intersection that can be found. To deal with this issue, we proposed the WCCG method to generate a group of concentric circles to increase the chance of finding intersection points.

Assuming the ranging result without systematic bias is r, and the ranging error is ε_i, then the radius of a concentric circle should be $r_i = r + \varepsilon_i$. In Fig. 6 and Table 2, we have a known error probability distribution $n_m \sim N(\mu, \sigma^2)$ which to determine the set of $\varepsilon = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_i, \cdots, \varepsilon_k]$.

Fig. 9 shows how the concentric circle method can solve the problem of no intersection. Both thick circles are two ranging results and all dashed thin circles are concentric circles to the two blue circles. The two thick circles have no intersection, but the dashed thin circles can supply intersection points. In this way, the issue of no intersection points can be fixed.

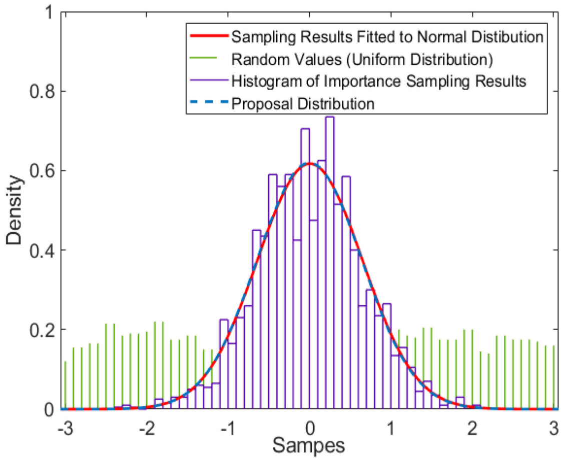


Fig. 10. Example of importance sampling.

Hence, we need to generate a concentric circle group $C = [C_1, C_2, \ldots, C_i, \ldots, C_k]$ to create intersections for the situation in Fig. 9. All circles in C have the same center which is the coordinate of the responder and each of C_i corresponds to a radius, r_i. Then we use importance sampling [56] to generate the set ε. According to the basic importance sampling principle,

$$E_p\ (f) = \frac{1}{m} \sum_{i=1}^{m} \frac{f(X_i)p(X_i)}{q(X_i)},\ X_i \sim q, \tag{17}$$

where $E_p(\cdot)$ denotes the expectation value for $X_i \sim q$, m is the number of samples in ε. $f(\cdot)$ is a function of X_i. $p(\cdot)$ is a probability density function which is the desired distribution; $q(\cdot)$ is the probability density function of the proposal distribution, $n_m \sim N(\mu, \sigma^2)$, which we found in Section 3.3. Here, $w = p/q$ is a weight of samples from q. This converts a uniformly distributed random sample distribution to the normal distribution fitting the error histogram.

Fig. 10 is an example of an importance sampling result, where the (blue) dashed curve is the proposal distribution, the (purple) bars are the sampling result histogram. The (green) impulses are randomly generated uniformly distributed values. After the process of importance sampling, the red curve, which is the sampling result fitted normal distribution, is matched with the distribution we wanted. With the methods of importance sampling, there should be no limitation to the shapes of proposal distributions that can be achieved. In future development of ε generation, the proposal distribution can be a more complicated model or manually adjusted to get a better positioning result.

In this way, each single ranging result can generate a group of concentric circles, which can be considered as a simulation of multiple noisy measurements recorded at a fixed position. After the WCCG process, all intersection positions will be calculated for the next process of Mean shift clustering.

4.3 Clustering Based Trilateration (CbT)

Trilateration requires at least three range measurements corresponding to different circles to determine a location.

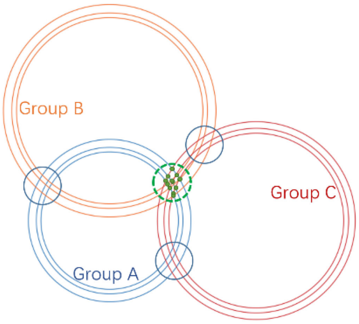


Fig. 11. Schematic diagram intersections of concentric circle groups.

Theoretically, four intersection points can be calculated from two pairs of circles and two of them should appear at or congregate at the same location which should be the estimated result. Fig. 11 shows the intersections between three concentric circle groups (each group has 3 circles), where group A and B have two groups of intersection points and the same for groups B and C and; groups A and C. There is a total of 27 intersection points assembled in the dashed circled area where the optimal estimate of the position of the initiator should be.

In position estimation, the mobile terminal can be estimated to be within the area with the highest point density. So, we propose a method to automatically cluster all calculated intersection points and find the highest density cluster area, as a practical solution to estimate the position of the initiator.

We used the clustering-based algorithm in two different types of positioning scenario. In a static positioning scenario, both MT and APs are placed at fixed positions. In this case, all ranging results during a certain time are processed together to calculate all intersection points and then mean shift clustering is used to determine the MT's location. In a dynamic positioning scenario, this method is used along with WCCG we proposed in Section 4.2. The WCCG can supply enough ranging circles based on a single range reading. It makes this process more like a simulation of multiple measurements at fixed positions.

4.3.1 The Principle of Mean Shift Clustering

The mean shift clustering algorithm is a non-parametric iterative algorithm that is widely used in mode recognition, clustering, etc. [57]. Both the shape and the scale of a cluster can be identified based on different kernels, neighborhood functions and mean shift clustering bandwidths. It also has an especially important feature that there is no requirement for the number of clusters and cluster centers to be known a priori, which suits our case as we do not know how many clusters we want and where they are.

Several parameters and functions should be known or pre-set before applying the mean shift clustering algorithm. First, a kernel $K(\cdot)$ function which is normally set to be a Gaussian Kernel [58].

$$K\ (d) = \frac{1}{\sqrt{2\pi}B}e^{-0.5(d/B)^2}, \tag{18}$$

where d is the distance determined by the neighborhood function, $N_{neighbor}(\cdot)$. B is called the mean shift clustering bandwidth which is a pre-set parameter to determine the scale of the cluster. Second, sample set $\boldsymbol{\Pi_i}$ includes all intersection points, $\boldsymbol{p}$, calculated from the ranging results. We have $N_{neighbor}(\cdot)$ which is the neighborhood function to determine the neighbor points around a sample position $\boldsymbol{p}$, where $\boldsymbol{p} \in \boldsymbol{\Pi_i}$. Here, we define $N_{neighbor}(\cdot)$ to be the Euclidean distance threshold to select nearby points, so that for $\boldsymbol{p} = (x, y)$ we have

$$N_{neighbor}(\boldsymbol{p}) = (\boldsymbol{p} - \boldsymbol{p}_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \tag{19}$$

For the first time of clustering, a sample point is randomly selected. Then the mean $m(\boldsymbol{p})$, which is also called the mean shift, of the intersection points near the randomly selected intersection point, is calculated using [57]

$$m(\boldsymbol{p}^l) = \frac{\sum_{\boldsymbol{p}_i^l \in \boldsymbol{P}_i^l} K(\boldsymbol{p}^l - \boldsymbol{p}_i^l)\boldsymbol{p}_i^l}{\sum_{\boldsymbol{p}_i^l \in \boldsymbol{P}_i^l} K(\boldsymbol{p}^l - \boldsymbol{p}_i^l)} = \hat{\boldsymbol{p}} = \boldsymbol{p}^{l+1}\ . \tag{20}$$

The cluster center, $\boldsymbol{p}^l$, at iteration, l, is moved to the newly calculated position, $\hat{\boldsymbol{p}}$, and is used as the input position for the next iteration, $m(\boldsymbol{p}^{l+1})$. Iterations of, $\boldsymbol{p}$, and updating continue until $\boldsymbol{p} = \hat{\boldsymbol{p}}$ when the position no longer moves, to finally find the cluster center, which is the end of the iterations for the first cluster.

All points in found clusters are excluded from the following iterations to find the next cluster and a new first sample point is randomly selected from amongst the remaining points. This is repeated several times until all samples are classified to different clusters.

4.3.2 Mean Shift Clustering-Based Positioning

After the clustering process, the cluster with the highest density of points is identified as the main cluster with the cluster center $C_m = (x_m, y_m)$. However, due to the ranging errors, the intersection area we want can be broken into a few different clusters close to each other. Hence, we identified sub-clusters with cluster centers $C_{sub}^i = (x_{sub}^i, y_{sub}^i)$, where i is the index number of the sub-cluster, which must also be considered as including points contributing to the optimal estimate. The distance between all clusters and the main cluster (center to center) is calculated and if the distance is shorter than a threshold, T_{select}, the cluster will be named as a sub-cluster.

The execution of sub-clusters selection requires two parameters to be specified, which are the mean shift bandwidth, B, and the threshold, T_{select}. First, we set the mean shift bandwidth, B, to be the RMS Error of Ranging Results (without systematic bias), e_{RMS}, where e_{RMS} is listed in Table 2. It is reasonable to believe that intersection points representing the position of the initiator should be assembled within a circle cluster with a radius of e_{RMS}. Another option for B is to use the standard deviation of the Error Probability Distribution function, σ in Table 2. However, in our case, the tests for e_{RMS} and σ showed that e_{RMS} gives a better clustering result. Second, the threshold for sub-cluster, choice T_{select} is set to be twice the maximum error in the Error Probability Distribution function histogram from Fig. 6 (respectively for different working bandwidths). It

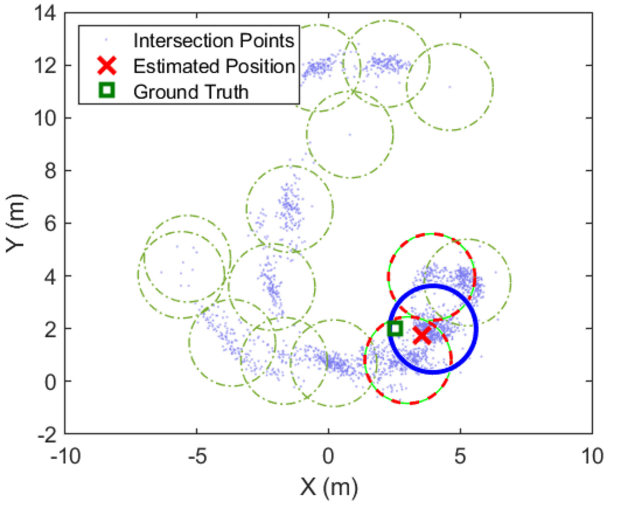


Fig. 12. Example of mean shift clustering.

means that if the detected cluster center is further than the maximum of the ranging error from the main cluster, then the points in these clusters are unlikely to represent the initiator position.

Fig. 12 is an example of a mean shift clustering and positioning result. In this figure, the (green) square is the ground truth of the MT and the (red) '×' symbol is the estimated position by our method. The (blue) thick circle is the main cluster, the (red - green) dashed-line circles are the sub-clusters and the (green) thin dotted-line circles are the rest of the detected clusters which are all identified as false intersection point clusters.

The final estimated position is the average coordinate of all points included in main cluster and all sub-clusters. The reason we do not calculate the average position of cluster centers as the estimation is that the sub-cluster has fewer points than the main cluster. The inclusion of all selected points in the calculation means that the greater the number of points a cluster has, the greater contribution it has to the final estimation.

4.3.3 *Algorithm Pseudo-Code*

In Step Two of our system shown in Fig. 8, CbT and WCCG are combined for a complete positioning approach. We called this combination CbT & WCCG. Fig. 13 shows the pseudo-code that implements the entire approach. Note that when there is enough ranging data for a static position, the WCCG part can be skipped and we can directly calculate the intersection point group $\boldsymbol{P}$.

4.4 CbT and WCCG Configuration Test and Computational Cost Evaluation and Comparison

Our Clustering-based Trilateration (CbT) method requires enough data packages (ranging measurements) to find the main cluster clearly and accurately. On the other hand, the quantity of concentric circles is strongly associated with the computational cost in the combination of algorithms CbT & WCCG. Thus, it is important to find an optimized quantity of concentric circles to achieve a reliable result at a minimal computational cost.

```
input  ranging data (continually received or imported as an entire
dataset); APs coordinates; Systematic bias, $e_{sys}$, $s$
output  MT_position coordinates

[initialize]  Parameter initialization for WCCG and CbT.
WCCG: number of circles $k$; proposal distribution $n_m \sim N(\mu, \sigma^2)$;
CbT: Mean shift clustering bandwidth $B(e_{RMS})$, for sub-cluster selection $T_{select}$; kernel identification (eq. (18))
loop until no import ranging data
  [weighted concentric circles generation]
  for each concentric circle
    $w = p/q$ calculate importance weight refers to eq. (17)
  end for
  resampling according to weight set $\boldsymbol{w}$
  get concentric circle group $\boldsymbol{C}$
  calculate all intersection points set $\boldsymbol{P}$
  [clustering]
  loop until no points remain in $\boldsymbol{P}$
    randomly select a cluster center $\boldsymbol{p}$
    loop until $\boldsymbol{p} = \hat{\boldsymbol{p}}$ convergence
      $N(\boldsymbol{p}^l) = (\boldsymbol{p}^l - \boldsymbol{p}^{l-1})$
      $\boldsymbol{p}^{l+1} = m(\boldsymbol{p}^l) = \hat{\boldsymbol{p}}$ mean shift refers to eq. (20)
    loop end
    saving $C_j$ and count number of points in it
    exclude all points in this cluster from $\boldsymbol{P}$
  loop end
  [main and sub-cluster selection]
  sort clusters, the cluster with the largest number of points is the
      main cluster $C_m$, saving $C_m = (x_m, y_m)$.
  for all $j$ clusters
    If $|C_j - C_m| < T_{select}$
      saving $C_{sub}^i = C_j$
  end for
  calculate MT_position = mean $(C_m + \sum C_{sub}^i)$
  return MT_position
end loop
```

Fig. 13. Pseudo-code for the CbT and WCCG algorithm.

We ran evaluation tests in this section based on datasets collected from the real experiments. Each dataset has 300 readings recorded in a 50 s collection time and each reading has ranging results from 4 APs. All tests are offline running in MATLAB (CORE i7, 6 processors; 16 GB RAM), to find the relationship between critical parameters and calculation time in our algorithms. Also, we ran LS and RLS algorithms for the same dataset to compare the computational cost.

Fig. 14 is a typical example of the positioning error results as a function of data collection time. Each red average error distance data point is calculated by repeatedly applying our CbT method 50 times while the shaded region shows the standard deviation of the results. At first, due to the brief time for data collection (a small number of data packages), the positioning results are very unstable, but both the results and uncertainty reduce to a stable low level after a certain number of data collection packets. Through multiple tests, we found that it takes about 7 seconds, around 40 recordings, on average, to keep the standard deviation at a stable low level to give a stable positioning result. This fact shows that the positioning at a fixed

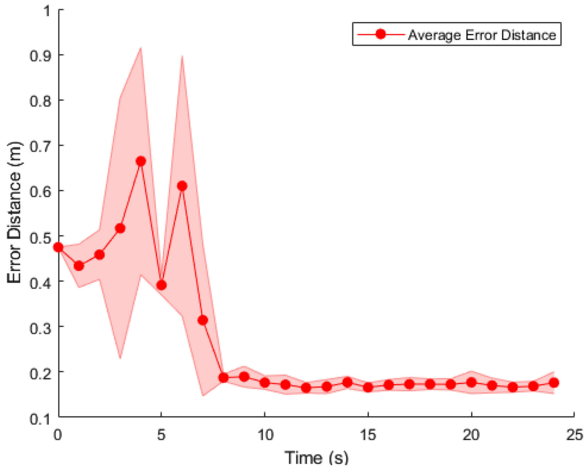


Fig. 14. Example of positioning error results versus data collection time.

position using mean shift clustering requires about 40 data readings to give an accurate and robust positioning result and a longer data collection duration contributes little to further improving the estimation result.

On the other hand, the computational cost of CbT & WCCG is strongly affected by the parameter settings in WCCG and the most important parameter affecting the computational cost is the number of concentric circles, k, in the initialization step of the pseudo-code in Fig. 13. Fig. 15 shows the test results of the relationship between the number of concentric circles and the calculation time. The calculation time shows a linear increase with a very shallow slope when we increase the number of circles by 5 circles for each test. It means that even if k was set as a large quantity, e.g., 100, circles for WCCG, the overall computational cost is still extremely low.

We also compared the computational cost between our algorithm and existing well-known algorithms. We did not use the computational complexity (e.g., the number of multiplications) as a way to compare the computational costs because the clustering calculation is based on randomly selecting an initial sample for finding each cluster, so that it is hard to know how many calculations will be executed and how many clusters will be found. Thus, we ran a test to compare the CbT, CbT & WCCG, LS, and RLS algorithms on the same computational device to compare the time cost, and the results are shown in Table 4. We programmed the LS trilateration method in MATLAB to solve eq. (4) and used publicly available MATLAB code for the RLS method [45]. We performed 48 different tests with each test using 300-readings and calculated the average and standard deviations of the ranges. The ranging time in this test was 0.16 s per reading. Also, the time required to calculate a single reading data point was recorded to find the minimum report interval which shows how long our algorithm should take to report the positioning result. The results are shown in Table 4. Note that this calculation time does not include the systematic bias removal process which can be used with any of these algorithms.

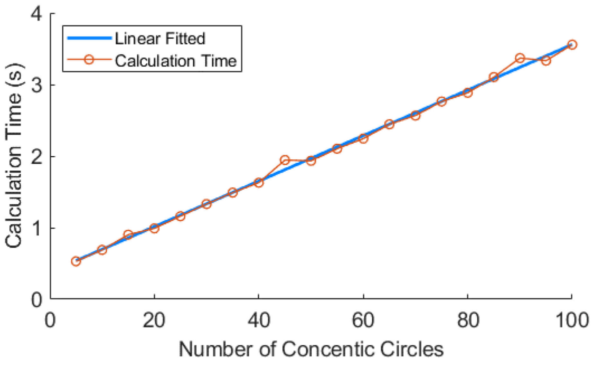


Fig. 15 Relationship between calculation time and number of concentric circles.

TABLE 4
Calculation Time Using Different Algorithms

	CbT	CbT & WCCG	LS	RLS
Average Time for 300-reading Datasets	0.107 s	1.503 s	102.335 s	0.273 s
Standard Deviation	0.125 s	0.217 s	0.594 s	0.007 s
Minimum Report Interval	-	0.074 s	1.396 s	0.033 s

In Table 4, CbT takes the shortest time (0.107 s) to finish the estimation for the entire dataset being more than twice as fast as the second fastest algorithm RLS. However, a single reading is not available to calculate a result for CbT as it requires multiple readings for clustering. The minimum time required for CbT should be determined by the ranging frequency, which means that it requires time to collect around 40 ranging data to give a reliable result. To note, both CbT and CbT & WCCG have larger standard deviations than RLS because the clustering process is based on randomly selecting the initial sample for cluster detection. It means that for each random choice of first sample, the calculation time will be different. An important point is that CbT & WCCG, LS and RLS have minimum report intervals which are less than the ranging frequency sampling time of 0.16s for 4 APs, so all are complete before the next reading is taken and, therefore, none incur a penalty in speed which is limited by the ranging frequency sampling time.

4.5 Kalman Filter Implementation

In step three of our designed system flow chart in Fig. 8, a digital filter is used to increase our position accuracy for dynamic motion estimation. In this work, we uses a Kalman filter [59]. The Kalman filter is recursive, using the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is needed. First, we define a state vector at time t to be:

$$\boldsymbol{state}\ (t) = \begin{bmatrix} \boldsymbol{p}(t) \\ \boldsymbol{ve}(t) \end{bmatrix}, \tag{21}$$

Where the location $\boldsymbol{p}\ (t) = [\, x(t) \quad y(t)\,]^T$ and the velocity $\boldsymbol{ve}\ (t) = [\, v_x(t) \quad v_y(t)\,]^T$ in our 2D scenario which assumes that the object is moving at a constant speed. Thus,

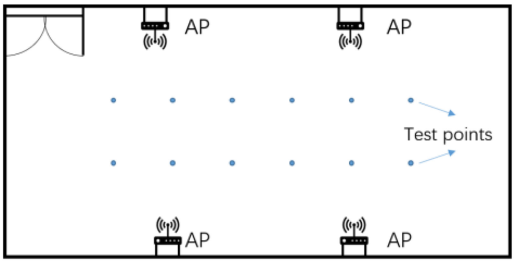


Fig. 16 Illustration of the experimental environment.

according to [59], to predict the position and velocity, we must solve the five equations:

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \\ K_k &= P_k^- H^T\left(HP_k^- H^T + R\right)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k\left(z_k - H\hat{x}_k^-\right) \\ P_k &= (1 - K_k H)P_k^-, \end{aligned} \tag{22}$$

where, $\hat{x}_k^-$ is the state estimation, and $\hat{x}_k$ is the updated state estimation. A is the transfer matrix between states at consecutive time steps, Δt, which can be expressed as

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{23}$$

B is the control-input model which is applied to the control vector u_k, since we do not have a control-input, $u_k = 0$. P_k^- is the estimated covariance matrix of the state vector at time step k based on the output at time step $k-1$. Q is the covariance matrix of the Gaussian noise in finding the priori estimation from the transfer matrix A. R is the covariance matrix of the Gaussian noise in the measurements of the user's states. K_k is the Kalman gain which describes the influence of the measurement on the final posteriori estimation. z_k is the measurement of the user's position from the Wi-Fi RTT positioning system. H is the observation matrix which can be set as

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{24}$$

5 Experimental Results and Analysis

5.1 Experimental Configuration

In our experiments for both static and dynamic positioning accuracy tests, we used APs (WILD fit2) [48] at four known positions which all use the 40 MHz bandwidths and one MT (Google Pixel 2) [15]. The test environment is a 12 m × 8 m office room with 12 test positions as shown in Fig. 16 to test the static positioning accuracy. Wi-Fi APs (responders) are placed at fixed positions whose locations in x and y were measured using a laser rangefinder. To minimize any error contributions due to geometry impact, our APs were deployed in a square shape [41].

All experiments were conducted without any people being present (only two experimenters were in the room during the experiment). Line-of-sight is guaranteed between all APs and the MT, and the MT is held horizontally on a plastic tripod for all static point positioning tests and carried by a remotely controlled robot for the dynamic tests. Also, because we only present the positioning performance in a 2D plane, the MT and APs are placed at the same altitude, 1.4 m from the floor. As the MT may have different performance in different directions, at each test point we placed the MT oriented horizontally in four directions and carried out four individual tests.

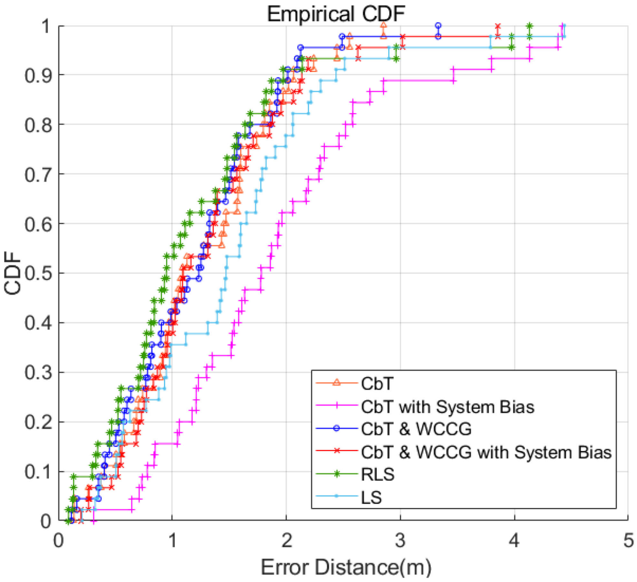


Fig. 17 CDF of static positioning error distance (system bias has been removed from all curves except where indicated.)

5.2 Static Positioning Experimental Results

The static positioning tests were carried out with both Wi-Fi APs (responders) and MT (initiator) placed at fixed positions. The overall test accuracy calculated by different processing approaches is shown in Fig. 17 as a Cumulative Distribution Function (CDF). In addition, statistical analysis results e.g., average errors, Root Mean Square (RMS) [60] errors, standard deviations, 90 percent CDF and 60 percent CDF are listed in Table 5. We also included LS and RLS for comparison with our algorithms. Note that the LS and RLS results in Fig. 17 and Table 5 have the systematic bias removed which was already shown to be beneficial in Fig. 7 and Table 3. The positioning result of our new clustering-based method, CbT, varies from one run to another using the same input data because the first point to initiate mean shift clustering is randomly selected as explained in Section 4.3.1. This results in the accuracy calculation having a slight variation so that the accuracy we used for statistical analysis was calculated from the average of three independent runs of the CbT algorithm.

In Fig. 17, the (orange) curve with 'Δ' symbol shows the CDF of the positioning results estimated by CbT and the result with systematic bias is shown by the (pink) curve with '+' symbols. The (blue) curve with 'o' symbols shows the CbT & WCCG positioning results and the (red) curve with '×' symbol shows the results with systematic bias. The (green) curve with '$*$' symbol shows the positioning results

TABLE 5
Static Positioning Accuracy Experimental Results (Using a 40-MHz Working Bandwidth)

	CbT	CbT with Bias	CbT & WCCG	CbT & WCCG with Bias	RLS (with bias removal)	LS (with bias removal)	CRB
Average Error	1.246 m	1.916 m	1.200 m	1.299 m	1.165 m	1.494 m	-
RMS Error	1.405 m	2.149 m	1.372 m	1.491 m	1.455 m	1.730 m	-
Standard Deviation	0.657 m	0.983 m	0.673 m	0.741 m	0.881 m	0.881 m	0.236 m
90% CDF	2.063 m	3.467 m	2.012 m	2.136 m	1.975 m	2.436 m	-
60% CDF	1.452 m	1.93 m	1.326 m	1.373 m	1.113 m	1.600 m	-

using the RLS method while the (light blue) curve with '.' symbol shows the results using LS.

From Table 5, in terms of average error, the RLS algorithm using our method for systematic bias removal has the best result giving a 1.165 m accuracy while CBT & WCCG with our method of bias removal gives a close second best accuracy with 1.200 m accuracy differing by only 35 mm. These results are better than all previous Wi-Fi based systems reported in [33], [61].

A particularly important advantage of our CbT & WCCG method we discovered is that when the systematic bias is not removed, we still achieve a 1.299 m accuracy which is only 99 mm worse than the accuracy with bias removal. These results show that the CbT & WCCG method has a better tolerance for systematic error than all other methods. Our new systematic bias removal procedure improves the CBT average error in Table 5 by 0.67 m and the CbT & WCCG average error by 99 mm. Looking back to Table 3 our new systematic bias removal procedure improves the LS average error by 0.790 m and the RLS average error by 0.579 m.

In terms of the robustness of positioning methods, the CbT & WCCG method with systematic bias removal shows advantages compared to all other methods in terms of RMS error (1.372 m) and CbT with systematic bias removal shows advantages compared to all other methods for standard deviation (0.657m). The standard deviation values of CbT with systematic bias removal (0.657m) and CbT & WCCG with systematic bias removal (0.673m) are closest to that of the Cramer-Rao Bound, CRB (0.236 m), which proves that these two methods are the most robust amongst the six methods listed in Table 5. Nevertheless, the standard deviation of all six methods is worse than that of CRB, showing that these methods still can be further improved in terms of robustness.

In the analysis of Fig. 17 CDF plot, RLS with our systematic bias removal process performs the best for both the 90 and 60 percent CDF errors. For the 90 percent CDF, the CbT with systematic bias removed gives 2.063 m and CbT & WCCG with systematic bias removed gives 2.012 m compared to RLS with systematic bias removed is 1.975 m, although all results are close to each other. For the 60 percent CDF, the CbT with systematic bias removed is 1.452 m and the CbT & WCCG with systematic bias removed is 1.326 m while RLS with systematic bias removed is 1.113 m, which is the best.

We conclude that if the systematic bias is stable over time and repeatable for particular models of smartphones and Wi-Fi APs, then they can be calibrated and stored as lookup tables for use in our systematic bias removal procedure. In this case our method and the RLS method with our systematic bias removal both can achieve satisfactory positioning accuracy. However, if the systematic bias vary over time or with temperature and from model to model and manufacturer to manufacturer of smartphones and Wi-Fi APs then the best method to use is CbT & WCCG without bias removal as this method is very tolerant to systematic bias and the additional loss of accuracy compared to the RLS with our systematic bias removal is only 0.139 m.

5.2 Dynamic Motion Tracking Experimental Results

In the dynamic motion test, a remotely controlled robot with a plastic shelf, shown in Fig. 18, is used to carry the initiator smartphone. Also, another camera phone was used to record all details such as the start/end time, positions and a second view of the trajectory to record the ground truth of the movement.

When the smartphone is dynamically moved by the robot, the data recording position is different at each time. So, our new algorithms CbT & WCCG are used for each data recording to find the ranging result and so give a positioning result.

Since the dynamic motion tests are designed to be performed in straight-line trajectories, a basic Kalman filter was used to improve the dynamic motion tracking result. Advanced filters such as Extended Kalman Filter (EKF) or particle filters can be applied when dealing with more complicated motion or motion in a three-dimensional space. The test results are presented in Table 6.

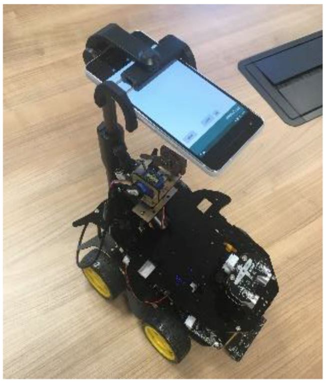


Fig. 18. Picture of an initiator carried by a robot.

TABLE 6
Dynamic Motion Test Results with CbT & WCCG
(Using a 40-MHz Working Bandwidth)

	Average Error	RMS Error	Standard Deviation
Dynamic tracking result	1.45 m	1.82 m	1.09 m
Dynamic result with Kalman filter	1.31 m	1.59 m	0.90 m

Comparing Table 6, with the static positioning results of Table 5, both RMS and average errors do not reach the accuracy level of the static situation but still give an average error of 1.45 m and a RMS error of 1.82 m. The Kalman filter improves the accuracy from 1.45 m to 1.31 m by 0.14 m and gives an improvement both in RMS error and standard deviation showing that it can improve the results.

6 Conclusion and Future Work

This paper gave a detailed description of how to characterize, design and implement the next generation of Wi-Fi based positioning supporting infrastructure – Fine Time Measurement (FTM), also known as Wi-Fi RTT ranging and positioning technology. We proposed a ranging measurement system using commercial-off-the-shelf (COTS) Wi-Fi RTT devices and smartphones. We also proposed a new systematic bias removal process and proved that it significantly improves the ranging accuracy and hence, the positioning accuracy when Wi-Fi RTT is implemented as an indoor positioning system. We also proposed a new positioning method of Clustering-based Trilateration (CbT) with Weighted Concentric Circles Generation (WCCG) to deal with the issues of no intersection points in trilateration. The accuracy of our method achieves the same level as the well-known RLS method if it is also combined with our new systematic bias removal process. Our new method of CbT & WCCG has a far better standard deviation and more closely approaching the CRB limit than any method we investigated. Also, it is an important discovery that CbT & WCCG performs well when there is an unknown or even time varying systematic bias so there is no need for prior calibration and removal of systematic bias. Finally, our Wi-Fi RTT ranging based positioning system has an average accuracy of around 1.20 m in an office environment for static positioning and 1.31 m for dynamic motion using a Kalman filter. As a trilateration positioning method, compared to a fingerprint-based system, it completely avoids the labor and time intensive initial survey (offline calibration) step. With knowledge of the position of the router's positions, the entire system can be simply implemented and easily brought into successful operation. Furthermore, because all experiments and tests are based on COTS devices, it means that it can be directly used in practical applications in our daily lives today.

In this paper, we used four WILD fit2 APs (with knowledge of their position coordinates in a 2D plane) to implement our positioning system which can easily be extended to higher numbers of APs to investigate the effect of this on system performance. Considering the three plots in Fig. 5, this performance was obtained with a clear line-of-sight so an investigation of how obstructions will affect the RTT ranging should also be carried out. With the methods of importance sampling, there should be no limitation to the shape of desired noise probability distributions. In future development, the desired noise probability distribution can be a more complex model to obtain a better positioning result.

Acknowledgments

Chengqi Ma and Bang Wu contributed equally to this work.

References

[1] P. P. OnkarPathak, R. Palkar, and M. Tawari, “Wi-Fi indoor positioning system based on RSSI measurements from Wi-Fi access points–a trilateration approach,” *Int. J. Sci. Eng. Res.*, vol. 5, no. 4, pp. 1234–1238, 2014.

[2] B. H. Liu, B. P. Otis, S. Challa, P. Axon, C. T. Chou, and S. Jha, “The impact of fading and shadowing on the network performance of wireless sensor networks,” *Int. J. Sens. Netw.*, vol. 3, pp. 211–223, 2008, doi: 10.1504/ijsnet.2008.019006.

[3] C. Ma, C. Wan, Y. W. Chau, S. M. Kang, and D. R. Selviah, “Subway station real-time indoor positioning system for cell phones,” in *Proc. Int. Conf. Indoor Positioning Indoor Navigation*, 2017, pp. 1–7, doi: 10.1109/IPIN.2017.8115912.

[4] C. Wu, Z. Yang, and C. Xiao, “Automatic radio map adaptation for indoor localization using smartphones,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 517–528, Mar. 2018.

[5] C. Basri and A. El Khadimi, “Survey on indoor localization system and recent advances of WiFi fingerprinting technique,” in *Proc. 5th Int. Conf. Multimedia Comput. Syst.*, 2016, pp. 253–259.

[6] X. Tian, S. Zhu, S. Xiong, B. Jiang, Y. Yang, and X. Wang, “Performance analysis of Wi-Fi indoor localization with channel state information,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1870–1884, Aug. 2019.

[7] Z. Zhou, C. Wu, Z. Yang, and Y. Liu, “Sensorless sensing with WiFi,” *Tsinghua Sci. Technol.*, vol. 20, no. 1, pp. 1–6, Feb. 2015, doi: 10.1109/TST.2015.7040509.

[8] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in *Proc. IEEE Int. Conf. Comput. Commun., 19th Annu. Joint Conf. IEEE Comput. Commun. Societies*, 2000, vol. 2, pp. 775–784.

[9] M. Brunato and R. Battiti, “Statistical learning theory for location fingerprinting in wireless LANs,” *Comput. Netw.*, vol. 47, no. 6, pp. 825–845, Apr. 2005, doi: 10.1016/j.comnet.2004.09.004.

[10] M. Youssef and A. Agrawala, “The horus location determination system,” *Wireless Netw.*, vol. 14, no. 3, pp. 357–374, Jun. 2008, doi: 10.1007/s11276-006-0725-7.

[11] W. Zhang *et al.*, “Radius based domain clustering for WiFi indoor positioning,” *Sensor Rev.*, vol. 37, no. 1, pp. 54–60, 2017.

[12] B. Wu, Z. Ma, S. Poslad, and W. Zhang, “An efficient wireless access point selection algorithm for location determination based on RSSI interval overlap degree determination,” in *Proc. Wireless Telecommun. Symp.*, 2018, pp. 1–8.

[13] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, “Scalable Wi-Fi client self-positioning using cooperative FTM-sensors,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 3686–3698, Oct. 2019.

[14] M. Ibrahim *et al.*, “Verification: Accuracy evaluation of WiFi fine time measurements on an open platform,” in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 417–427, doi: 10.1145/3241539.3241555.

[15] “These are the google pixel 2 and pixel 2 XL,” *VentureBeat*, Accessed: Oct. 16, 2019. [Online]. Available: https://venturebeat.com/2017/10/02/these-are-the-google-pixel-2-and-pixel-2-xl/

[16] “WILD, ” *Fit IoT*, Accessed: Oct. 07, 2019. [Online]. Available: https://fit-iot.com/web/product/wild/

[17] R. Want, W. Wang, and S. Chesnutt, “Accurate indoor location for the IoT,” *Computer*, vol. 51, no. 8, pp. 66–70, Aug. 2018, doi: 10.1109/MC.2018.3191259.

[18] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, “High-accuracy indoor geolocation using collaborative time of arrival (CToA)-whitepaper,” *IEEE*, vol. 802, p. 17, 2017.

[19] A. Makki, A. Siddig, M. Saad, and C. Bleakley, "Survey of WiFi positioning using time-based techniques," *Comput. Netw.*, vol. 88, pp. 218–233, Sep. 2015, doi: 10.1016/j.comnet.2015.06.015.

[20] W. Zhuang and J. Tranquilla, "Modeling and analysis for the GPS pseudo-range observable," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, no. 2, pp. 739–751, Apr. 1995.

[21] Z. Ma, B. Wu, and S. Poslad, "A WiFi RSSI ranking fingerprint positioning system and its application to indoor activities of daily living recognition," *Int. J. Distrib. Sens. Netw.*, vol. 15, no. 4, pp. 1–14, Apr. 2019, doi: 10.1177/1550147719837916.

[22] Z. Ma, S. Poslad, J. Bigham, X. Zhang, and L. Men, "A BLE RSSI ranking based indoor positioning system for generic smartphones," in *Proc. Wireless Telecommun.*, 2017, pp. 1–8.

[23] Z. Ma, S. Poslad, S. Hu, and X. Zhang, "A fast path matching algorithm for indoor positioning systems using magnetic field measurements," in *Proc. IEEE 28th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun.*, 2017, pp. 1–5.

[24] J. Duan, K. Lachhani, H. Baghsiahi, E. Willman, and D. R. Selviah, "Indoor rigid sphere recognition based on 3D point cloud data," in *Proc. Irish Mach. Vis. Image Process.*, 2014, pp. 28–33.

[25] Z. Ma, J. Bigham, S. Poslad, B. Wu, X. Zhang, and E. Bodanese, "Device-free, activity during daily life, recognition using a low-cost lidar," in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–6.

[26] D. R. Selviah, "Robust Automatic 3D point cloud registration and object detection," *Geomat. World*, pp. 1–8, 2018.

[27] D. R. Selviah, "Automated 3D point cloud data processing using AI," *Geomat. World*, vol. 28, no. 1, pp. 18–21, 2020.

[28] K. Lachhani, J. Duan, H. Baghsiahi, E. Willman, and D. R. Selviah, "Correspondence rejection by trilateration for 3D point cloud registration," in *Proc. 14th IAPR Int. Conf. Mach. Vis. Appl.*, 2015, pp. 337–340.

[29] Z. Yang, Z. WANG, J. Zhang, C. Huang, and Q. Zhang, "Polarization-based visible light positioning," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 715–727, Mar. 2019.

[30] B. Li *et al.*, "Vision-based mobile indoor assistive navigation aid for blind people," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 702–714, Mar. 2019.

[31] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surv. Tuts.*, vol. 21, no. 3, pp. 2568–2599, Thirdquarter 2019.

[32] H. Haas, L. Yin, Y. Wang, and C. Chen, "What is LiFi?," *J. Lightw. Technol.*, vol. 34, no. 6, pp. 1533–1544, Mar. 2016, doi: 10.1109/JLT.2015.2510021

[33] S. He and S.-G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surv. Tuts.*, vol. 18, no. 1, pp. 466–490, Firstquarter 2016.

[34] B. T. Fang, "Trilateration and extension to Global Positioning System navigation," *J. Guid. Control Dyn.*, vol. 9, no. 6, pp. 715–717, 1986, doi: 10.2514/3.20169.

[35] D. Niculescu and Badri Nath, "Ad hoc positioning system (APS) using AOA," in *Proc. 22nt Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2003, vol. 3, pp. 1734–1743, doi: 10.1109/INFCOM.2003.1209196.

[36] A. Alhasanat, B. Sharif, C. Tsimenidis, and J. Neasham, "Efficient RSS-based collaborative localisation in wireless sensor networks," *Int. J. Sensor Netw.*, vol. 22, no. 1, pp. 27–36, 2016.

[37] W. Navidi, W. S. Murphy, and W. Hereman, "Statistical methods in surveying by trilateration," *Comput. Statist. Data Anal.*, vol. 27, no. 2, pp. 209–227, Apr. 1998, doi: 10.1016/S0167-9473(97)00053-4.

[38] S. Pradhan, S.-S. Hwang, H.-R. Cha, and Y.-C. Bae, "Line intersection algorithm for the enhanced TOA trilateration technique," *Int. J. Humanoid Robot.*, vol. 11, no. 4, Oct. 2014, Art. no. 1442003, doi: 10.1142/S0219843614420031.

[39] S.-S. Hwang and S. Pradhan, "Comparison approach of intersection distances for advanced TOA trilateration," *Int. J. Humanoid Robot.*, vol. 14, no. 3, Sep. 2017, Art. no. 1750019, doi: 10.1142/S0219843617500190.

[40] S. Hwang and S. Shin, "Advanced TOA trilateration algorithm for mobile localization," in *Proc. IEEE Asia-Pacific Conf. Antennas Propag.*, 2018, pp. 543–544, doi: 10.1109/APCAP.2018.8538295.

[41] G. Retscher, J. Kleine, and L. Whitemore, "Trilateration approaches for indoor Wi-Fi positioning," in *Proc. E3S Web Conf.*, 2019, vol. 94, Art. no. 02002, doi: 10.1051/e3sconf/20199402002

[42] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Philadelphia, PA, USA: Siam, vol. 15, 1995.

[43] W. Murphy and W. Hereman, "Determination of a position in three dimensions using trilateration and approximate distances," Colorado School Mines, Golden, CO, USA, Tech. Rep., MCS-95, vol. 7, p. 19, 1995.

[44] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[45] A. Norrdine, "An algebraic solution to the multilateration problem," in *Proc. 15th Int. Conf. Indoor Positioning Indoor Navigation*, 2012, vol. 1315, pp. 1–4.

[46] K. H. Choi, W.-S. Ra, S.-Y. Park, and J. B. Park, "Robust least squares approach to passive target localization using ultrasonic receiver array," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 1993–2002, Apr. 2013.

[47] Y. Chen, J.-A. Francisco, W. Trappe, and R. P. Martin, "A practical approach to landmark deployment for indoor localization," in *Proc. 3rd Annu. IEEE Commun. Soc. Sensor Ad Hoc Commun. Netw.*, 2006, vol. 1, pp. 365–373.

[48] J. Perez-Ramirez, D. K. Borah, and D. G. Voelz, "Optimal 3-D landmark placement for vehicle localization using heterogeneous sensors," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 2987–2999, Sep. 2013.

[49] J. J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 40–47, Jan. 1996.

[50] Z. Ben-Haim and Y. C. Eldar, "The Cramér-Rao bound for estimating a sparse parameter vector," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3384–3389, Jun. 2010.

[51] J. Werner, J. Wang, A. Hakkarainen, D. Cabric, and M. Valkama, "Performance and Cramer–Rao bounds for DoA/RSS estimation and transmitter localization using sectorized antennas," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3255–3270, May 2016.

[52] A. Al-Qaisi, A. I. Alhasanat, A. Mesleh, B. S. Sharif, C. C. Tsimenidis, and J. A. Neasham, "Quantized lower bounds on grid-based localization algorithm for wireless sensor networks," *Ann. Telecommun.*, vol. 71, no. 5–6, pp. 239–249, 2016.

[53] "WifiManager startScan throttled in P Beta (Developer Preview 2) [79906367]- Visible to Public - Issue Tracker," Accessed: Oct. 07, 2019. [Online]. Available: https://issuetracker.google.com/issues/79906367

[54] "WifiManager, " *Android Developers*, Accessed: Oct. 07, 2019. [Online]. Available: https://developer.android.com/reference/android/net/wifi/WifiManager

[55] "Wi-Fi RTT (IEEE 802.11mc)," *Android Open Source Project*, Accessed Aug. 05, 2019. [Online]. Available: https://source.android.com/devices/tech/connect/wifi-rtt

[56] M. Mazumdar, "Importance sampling in reliability estimation," in *Proc. Rel. Fault Tree Anal.*, 1975, pp. 153–163.

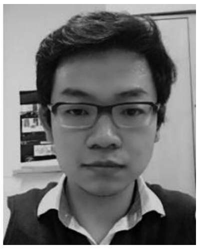
[57] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[58] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, Jan. 1975.

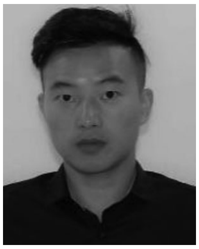
[59] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. 1995. Chapel Hill, NC, USA: Univ. of North Carolina Press.

[60] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Clim. Res.*, vol. 30, no. 1, pp. 79–82, 2005.

[61] P. Davidson and R. Piché, "A Survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 2, pp. 1347–1370, Secondquarter 2017.



Chengqi Ma received the BS degree in communication engineering from the Harbin Institute of Technology, Harbin, China, in 2012, and the MS degree in wireless communication from Lund University, Lund, Sweden, in 2015. He is currently working toward the PhD degree in Electrical Engineering Department at University College London, London, U.K. His research interests include the development of indoor positioning, navigation and tracking system, human activity recognition and Internet of Things.



Bang Wu received the BS and MS degrees from the School of Geodesy and Geomatics from Wuhan University, Wuhan, China, in 2014 and 2016, respectively. He is currently working toward the PhD degree at the School of Electronic Engineering and Computer Science at Queen Mary, University of London, London, U.K. His research interests include indoor positioning and indoor navigation, human activity recognition, Internet of Things and artificial intelligence.

Stefan Poslad received the PhD degree from Newcastle University. He is currently an associate professor with the School of Electronic Engineering and Computer Science at Queen Mary, University of London, London, U.K. He heads the IoT Lab and his research interests are indoor positioning and indoor navigation, human activity recognition, Internet of Things, ubiquitous computing, Semantic Web, distributed system management, and artificial intelligence.

David R Selviah (Member, IEEE) received the PhD degree in photonic engineering from Trinity College, Cambridge University, Cambridge, UK. He is currently a reader of optical devices, interconnects, algorithms and systems with the Electronic and Electrical Engineering Department at University College London, UCL, London, U.K. His current research interests include machine learning, feature recognition, 3D point cloud processing, indoor positioning and navigation, and quantum dot material characterization.