



WiFi-RTT indoor positioning using Particle, Genetic and Grid filters with RSSI-based outlier detection

Khalil Jibran Raja and Paul D. Groves[✉]

UCL Engineering, University College London, London, UK

Corresponding author: Khalil Jibran Raja; Email: zceskjr@ucl.ac.uk

Received: 26 March 2024; Revised: 16 September 2025; Accepted: 10 October 2025

Keywords: genetic filter; grid filter; Indoor positioning; particle filter; WiFi RTT

Abstract

The paper explores the accuracy of WiFi-Round Trip Timing (RTT) positioning in indoor environments. Filtering techniques are applied to WiFi-RTT positioning in indoor environments, enhanced by Residual Signal Strength Indicator (RSSI)-based outlier detection. A Genetic and Grid filter are compared with a Particle filter and single-epoch least-squares across a range of test scenarios. In static scenarios, 67% of trials had sub-metre accuracy and 90.5% had a root mean square error (RMSE) below 2 m. In Non-Line-of-Sight (NLOS) conditions, 38% of trials had sub-metre accuracy, whereas for environments with full Line-of-Sight (LOS) conditions, 95.2% of trials had sub-metre accuracy. In scenarios with motion, 22.2% of trials had sub-metre accuracy. RSSI-based outlier detection in NLOS conditions, provided an average improvement of 41.3% over no outlier detection across all algorithms in the static and 14% in the dynamic tests. The Genetic filter achieved a mean improvement of 49.2% in the static and 47% in the dynamic tests compared with least squares.

1. Introduction

WiFi is one of the primary tools for mobile indoor positioning due to WiFi access points' (APs) vast infrastructure presence and typically static location (Groves, 2013). WiFi Fine Time Measurement (FTM) is a WiFi protocol enabled in routers and mobile devices that use chips compatible with 802.11 mc (Google, 2022); it enables the Time of Flight (ToF) of a WiFi signal to be determined. The system that applies this protocol is referred to as WiFi Round Trip Timing (RTT). This protocol allows ToF-based positioning algorithms to be applied using WiFi signals, which could lead to more accurate and reliable indoor positioning than WiFi Residual Signal Strength Indicator (RSSI)-based positioning. Non-Line-of-Sight (NLOS) reception and multipath interference are inevitable in complex indoor environments and degrade WiFi RTT accuracy.

The research in this paper explores the accuracy and reliability of WiFi RTT positioning in a variety of indoor environments using filtering techniques. Four positioning algorithms are explored: single-epoch least squares, a Particle filter using Sequential Importance Resampling (SIR), a new Genetic filter that applies a genetic algorithm to the resampling step of a particle filter and a Grid filter which is also new to WiFi positioning. All methods were also augmented with an RSSI-based outlier detection model, which detects NLOS signals and severe multipath interference by finding inconsistencies between the RTT range and RSSI of the measurement. The algorithms were tested in several static positioning settings and in scenarios where the mobile device was in motion.

For the in-motion experiments, the algorithms incorporated a step-lagged Pedestrian Dead Reckoning (PDR) model with dynamic step length estimation, using the inertial sensors on the mobile device for step detection, orientation estimation and to derive step length.

The paper will begin by providing a background of the current literature, then an explanation of all the techniques explored in this paper will be provided. Next, the experimental methodology will be provided and finally, the results of all experiments will be analysed.

2. Background

Currently, companies such as Google, crowd source large amounts of WiFi data to construct models and predict the locations of APs to improve the practicality of indoor positioning by having a larger database. WiFi RTT could be used in this way, but the accuracy would likely be better than RSSI with a given amount of data or the same accuracy with less data due to RTT being specifically designed for providing ranging measurements. It has been found that fingerprints change in changing environments such as underground stations (Ma et al., 2017), limiting their accuracy in certain locations where indoor positioning would be useful.

Google believes WiFi RTT can achieve sub-metre accuracy (Google, 2022; Van Diggelen et al., 2018). Malkos and Hazlett (2014) found that the Line-of-Sight (LOS) ranging error on calibrated devices was approximately 1 m, up to a distance of 35 m. This aligns with research conducted in this paper, where the accuracy up to a 10 m distance for LOS scenarios was sub-metre when the instruments were correctly calibrated. NLOS ranging error is the error caused by a signal's estimated range to be larger than the true range as a result of obstructions between the transmitter and receiver. This can be caused by signal obstruction or signal reflection off of objects in the environment.

WiFi RTT-based positioning has been explored by several different authors. Gentner et al. (2020) reviewed the distribution of the WiFi-RTT distance estimation errors and developed a Gaussian mixture model. This model was then used in a Particle filter. In an experiment involving a person carrying a mobile device being tracked, the average positioning error was found to be 0.93 m, which is an improvement of 0.45 m over the model using a standard Gaussian measurement model. Gentner et al. found there was a -1.3 m mean bias between the measured and true distance, indicating some sort of instrument bias. This bias is also supported by research conducted as part of this paper, where the instrument bias was measured at approximately -1 m.

Guo et al. (2019) opted for a different approach that revolved around a hybrid RSSI and RTT model. The model essentially used the RSSI of each RTT signal to determine whether the signal was from a LOS or NLOS signal. This was done by comparing the predicted distance based on the measured RSSI using an RSSI path loss model against the RTT measured range. If the two ranges were different by a certain margin, then the RTT measurement would not be used. The paper compared RSSI fingerprinting against a Kalman filter using WiFi RTT ranging data. The average accuracy of the fingerprinting was 3.41 m, whilst the Kalman filter had an accuracy of 2.04 m for the same environment. When the RSSI outlier detection method was used, the positioning accuracy of the RTT-based Kalman filter improved to 1.44 m.

Sun et al. (2020) and Yu et al. (2019) integrated PDR with WiFi RTT in an extended Kalman filter. Sun et al. used the Mahony complementary filter (Mahony et al., 2008) for estimating the heading angles from the gyro and magnetometer measurements, and the Weinberg (2002) model for step length estimation. The model in this paper also used RSSI for outlier detection by determining the standard deviation of the RSSI and ranging data in one second. If the standard deviation was above a certain threshold, then the measurement was removed from consideration for the positioning solution. This method achieved an RMSE of 1.1 m. This is better than WiFi FTM multilateration without outlier detection, which achieved an RMSE of 2.74 m. Guo et al. (2019) Hashem et al. (2020) and Sun et al. (2020) clearly showed that the accuracy of WiFi RTT could be improved by developing algorithms that also take into account the RSSI of a WiFi RTT measurement.

Gentner and Avram (2020) developed a WiFi RTT Simultaneous Localisation and Mapping (SLAM) model. This is one of the first papers that does not assume prior knowledge of the environment such as the location of APs or a fingerprint database. This model follows a SLAM approach with a Particle filter

and fuses WiFi RTT ranging measurements with IMU data from the same mobile device; the model estimates the AP locations and the position of the mobile device simultaneously. Gentner and Avram (2020) highlight the importance of using a Particle filter over a low complexity extended Kalman filter due to the nonlinearity of the measurements. The estimated mobile device position using this method had an RMSE below 1 m.

Malkos and Hazlett (2014) and Gentner et al. (2020) explored WiFi RTT in combination with a Particle filter. Aside from a few other papers that also use Particle filters, not much research has been conducted on alternative filtering techniques. Three filtering techniques will be explored in the following sections alongside a new RSSI-based outlier detection model.

3. Measurements

For the experiments in this paper, all data were collected simultaneously by the same custom mobile app, the RTT measurements were received with pings at 100 ms intervals, and the IMU data were collected as fast as the mobile and were capable of delivering IMU updates (most sensors updated faster than 20 ms).

A motion model must be integrated into the prediction step of the filters to properly reflect the expected position of a particle. PDR is an approach that takes advantage of step detection and uses a human step to indicate the movement of the mobile device. The PDR model is being explained first as it is common to all filters used in this paper. The step detection is carried out using the Google Android API (Android [1], 2023) due to its convenience and because this paper does not intend to explore step detection models, and this detects a ‘high variation in acceleration’ from the mobile device’s accelerometer. For step detection, a step length estimation is required to determine the distance travelled by a mobile device when a step occurs. The step length varies for many reasons; it can vary due to different pedestrians having different walking gaits and pedestrians walking at different speeds. The model used in this paper is based on the Mikov step length estimation equation (Mikov et al., 2013),

$$q = \max(\alpha t_{\text{step}} \sqrt[4]{f_{z,\text{max}} - f_{z,\text{min}}}, q_{\text{max}}) \quad (1)$$

where q is the step length, α is a constant parameter that is calibrated per pedestrian, t_{step} is the time taken for a given step, $f_{z,\text{max}}$ is the maximum specific force along the z axis during the step and $f_{z,\text{min}}$ is the minimum specific force along the z axis during the step. This model was selected because it provided an improvement (Mikov et al., 2013) over the commonly used Weinberg (2002) model, but does not require significant calibration such that this model would be difficult to reproduce. The model was calibrated by walking along 2 paths of known length with different step sizes and selecting the value of the constant α that minimised the overall distance measurement error. An upper limit of $q_{\text{max}} = 1100$ mm was imposed to mitigate the impact of gross outliers in the mobile device’s IMU measurements.

The direction of travel was assumed to match the heading output by the Android ‘orientation sensor’ (Android [2], 2023), which calculates 3D orientation using the device’s magnetometers, accelerometers and gyroscopes. This was transformed to the coordinate frame of the indoor map. The initial heading error is smaller than subsequent heading error as these are subject to random hand-orientation changes from the pedestrian during walking.

4. Filtering techniques

4.1 Particle filter

Particle filters are a promising method for WiFi RTT positioning, as they are better suited than standard Kalman filters for highly nonlinear problems (Arulampalam et al., 2002; Gentner et al., 2020). WiFi RTT is a nonlinear problem as the RTT measurements are not linear functions of the mobile

device position and are non-Gaussian. The latter is mainly a result of NLOS reception errors and multipath effects.

The process for the Particle and Genetic filters follows the diagram shown in Figure 1.

An initial position estimate is determined based on a single epoch least squares algorithm.

Initialisation – N_p particles are created using the initial position estimate as the mean. The particles are then randomly distributed around the initial position estimate to account for uncertainty. The standard deviation of the state distribution matches the uncertainty of the initial position estimate. The initial distribution of the particles follows a Gaussian distribution.

Prediction – For positioning in motion, the PDR model described in Section 3 is used. The positions of the particles are adjusted using

$$X_{k+1}^n = X_k^n + \cos(\varphi_{k+1}^n) r_{k+1}^n \quad (3)$$

$$Y_{k+1}^n = Y_k^n + \sin(\varphi_{k+1}^n) r_{k+1}^n \quad (4)$$

where X are the x coordinates of the particles and Y are the y coordinates of the particles, according to the coordinate frame of the environment, which can vary for indoor environments, k represents the epoch of the particles and n represents the particle index. The displacement, r_k , and orientation, φ_k , are retrieved as described below and defined in Equations (5) and (6), respectively.

As steps are discrete events, the movement of the device appears to be discrete from the perspective of the filter as the prediction step will update the device's position at each step. This is an issue as WiFi measurements can be recorded mid-step, which are used to determine the weights of particles. Thus, only updating the particle states per step will not give a precise representation of the device's state. To resolve this, step-lagged smoothing is used. This method allows the filter to account for the movement of the mobile device during a step by using the time taken between a step along with the step length to determine a velocity of the step. When a new epoch of WiFi RTT ranges is received, the distance travelled is determined using the step velocity and the time taken between the WiFi RTT epochs, allowing the particles access to a more precise representation of the mobile device's expected position during a step.

The step length and orientations are retrieved as described in Section 3 and then system noise is incorporated into these values, as described as follows for the displacement and heading respectively:

$$r_{k+1}^n = \frac{N(q_{k+1}, \sigma_q)}{n_e} \quad (5)$$

$$\varphi_{k+1}^n = N(\varphi_{k+1}, \sigma_\varphi) \quad (6)$$

where r_k represents the displacement of the device for an epoch, q represents the step length, σ_q represents the assumed standard deviation of the step length error, n_e represents the number of filter epochs that have occurred between a full step and N indicates that each element is sampled from a Gaussian distribution. In Equation 6, φ_k represents the heading of the mobile device, φ_{k+1}^n represents the heading of the device with Gaussian noise incorporated and σ_φ represents the assumed standard deviation of the heading error.

The full motion model algorithm is shown in Figure 2 and occurs during the prediction step of all filters.

Update – The weights of each particle are updated by calculating how closely the measured ranges to each landmark match the particles' distance to the landmarks. The model used for weighting is described as follows:

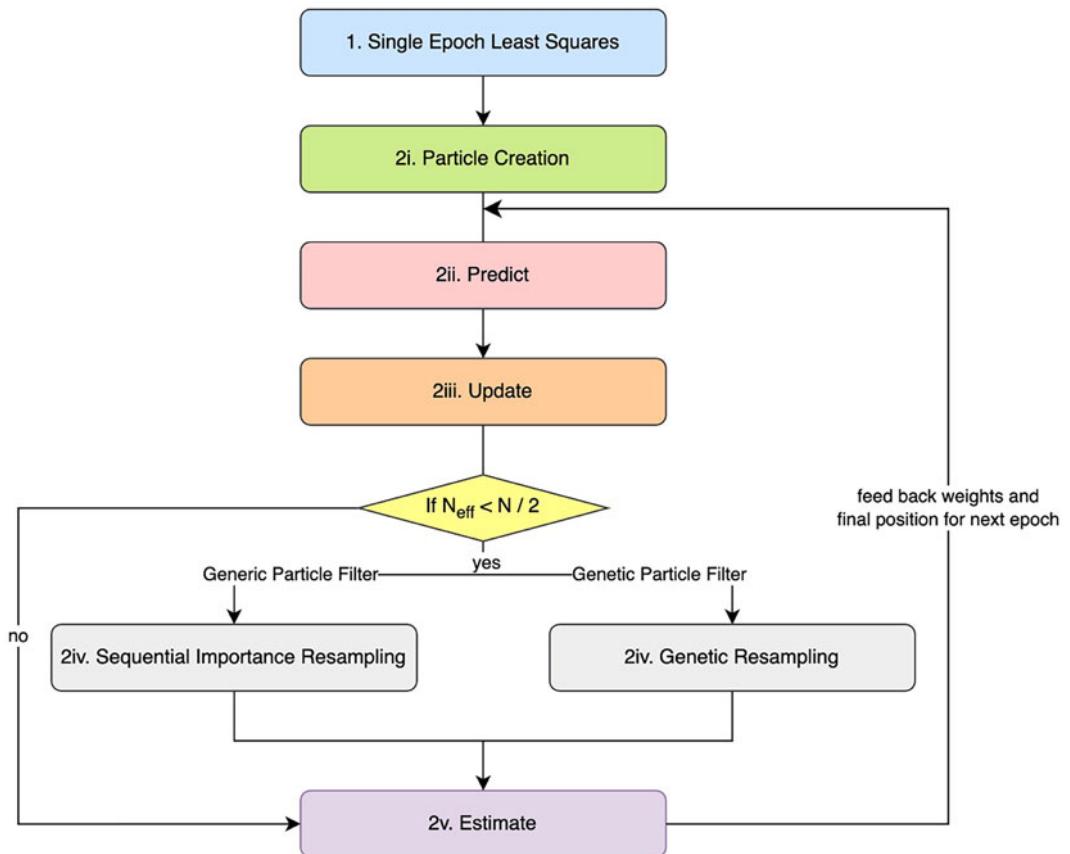


Figure 1. Generic particle and genetic filter process.

$$w_{k+1} = w_k \frac{1}{\sigma_k^i \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(z_k^i - d_k^i)^2}{\sigma_k^i}} \quad (7)$$

The update step for computing weightings is the step for determining how strongly a particle matches the measured state from state estimates. The Euclidean distance, d_k^i , between each particle and the landmarks are computed, where i is the AP landmark being measured from and k represents the epoch which is then treated as the mean in a Normal distribution alongside a standard deviation. The standard deviation is assumed to be equal across all APs at this stage, which is modified using RSSI-based outlier detection which is described in Section 5. Once the normal distribution is determined, the probability distribution function (PDF) of the distribution at z_k^i , the measurement obtained for the distance between the AP, i , and the mobile device is calculated at epoch k , which gives the particle weight for that landmark. The weights for all landmarks for each epoch are then multiplied together to give a final weight for that particle and this process is repeated for all particles. Following this, the weights are normalised.

Resampling – If Equation (8) evaluates to be true, then Sequential Importance Resampling (SIR) is carried out. In this process, the higher weighted particles are retrieved and replace lower weighted particles whilst resetting all weights.

The effective sample size, N_{eff} , is determined as shown in Equation (9) so long as Equation (8) is evaluated as true,

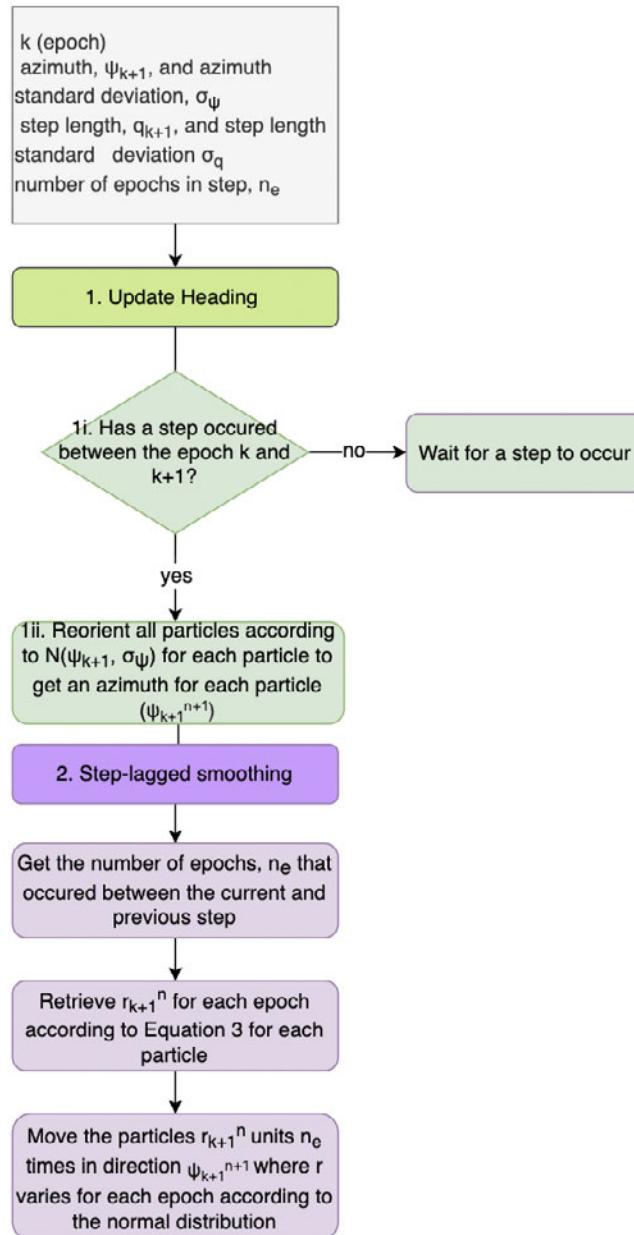


Figure 2. Step-lagged PDR motion model.

$$N_{eff} < \frac{N_p}{2} \quad (8)$$

$$N_{eff} = \frac{1}{\sum_{n=1}^{N_p} (w_k^n)^2} \quad (9)$$

where N_p is the number of particles and w_k^n is the normalised weight of the n th particle at epoch k . In most Particle filters, SIR is used (Arulampalam et al., 2002), where resampling is triggered because the number of effective particles is too low. SIR essentially replicates higher weighted particles and deletes

lower weighted particles. This is done to mitigate particle degeneracy. As the algorithm progresses, there will be particles that have very low weight and contribute nothing to the solution, and are thus wasted computation. The resampling step essentially removes these low weighted particles in favour of the higher weighted particles such that more particles are being used to identify the position of the mobile device.

Estimate – Compute the mean and standard deviation of the states using the particle weights according to the following:

$$\bar{x}_w^k = \frac{\sum_{n=1}^{N_p} w_k^n X_k^n}{\sum_{n=1}^{N_p} w_k^n} \quad (10)$$

$$\bar{y}_w^k = \frac{\sum_{n=1}^{N_p} w_k^n Y_k^n}{\sum_{n=1}^{N_p} w_k^n} \quad (11)$$

where \bar{x}_w^k is the weighted average of the x coordinate of the position estimate at epoch k and \bar{y}_w^k is the weighted average of the y coordinate of the position estimate at epoch k .

The Prediction, Update, Resampling and Estimation steps are repeated for each epoch.

4.2 Genetic filter

The Genetic filter (Higuchi, 1997; Park et al., 2007) offers an alternative to SIR at the resampling step; instead, the resampling follows a genetic algorithm. The intention of this method is to provide an alternative and potentially more effective way of mitigating particle degeneracy and increase the diversity of the particles during resampling than SIR, such that fewer particles could be used for the same level of accuracy. Mitigating particle degeneracy is important as it reduces the chance of the particles of a Particle filter focusing on false positives too quickly. Essentially, the Genetic filter provides the particles with more opportunity to explore the search space to identify regions which could result in a higher weighting. A Genetic filter has been applied to an ultrawideband (UWB) positioning solution by Zhou et al. (2021).

The resampling step of a Genetic filter is composed of three steps: classification, crossover and mutation. At the selection/classification step, the algorithm gets a set of strongly weighted and weakly weighted particles. Classification is the process of sorting the particles into mating pools to be used in the crossover step. The particles are sorted into ascending order based on weight. Then, the particles are split into two particle sets, computed by

$$nint(p) \leq N_{eff} < nint(p + 1) \quad (12)$$

where p represents the integer that is used as the index to split the ordered particle set, where the set with index 0 to p is the higher weighted set and the set with starting index $p+1$ to N_p is the lower weighted set.

For the crossover step, to summarise, each strongly weighted particle undergoes an arithmetic crossover with another strongly weighted particle to produce two offspring particles which will replace the two parent particles. Then, all weakly weighted particles undergo an arithmetic crossover with a strongly weighted parent particle and the new particles replace the original particles.

Crossover is the process of taking two random particles from the mating pools and exchanging information between them to construct a new particle (referred to as offspring). The crossover operation conducted for this algorithm is an arithmetic crossover,

$$\begin{aligned} \omega_k^{off, n} &= \alpha_1 \omega_k^{h,n} + (1 - \alpha_1) \omega_k^{h,n+1} \\ \omega_k^{off, n+1} &= \alpha_2 \omega_k^{h,n+1} + (1 - \alpha_2) \omega_k^{h,n} \end{aligned} \quad \text{where } n \leq p \quad (13)$$

$$\alpha_1 = \frac{w_k^{h,n}}{(w_k^{h,n} + w_k^{h,n+1})} \quad \text{where } n \leq p$$

$$\alpha_2 = \frac{w_k^{h,n+1}}{(w_k^{h,n} + w_k^{h,n+1})} \quad (14)$$

Equations (13) and (14) describe the arithmetic crossover conducted on the higher weighted set. The two offspring particles are then used for the next epoch. The total number of particles remains the same. Here, ω_k^{off} represents the states of the offspring particles and ω_k^h represents the states of the higher weighted particles,

$$\omega_k^{off,n} = \beta \omega_k^{l,n} + (1 - \beta) \omega_k^{h,0} \quad \text{where } p \leq n + 1 \leq N_p \quad (15)$$

Equation 15 describes the arithmetic crossover conducted for the lower weighted particle set. In this version of the crossover, instead of taking two parents from the same set, a particle from the higher weighted set is crossed over with a random particle from the lower weighted particle set. The parent particle used and the new offspring particle are then passed onto the next generation. Here, ω_k^l represents the states of the lower weighted particles, where β is a random number between 0 and $(N_p - N_{eff})/N_p$.

For this algorithm, a mutation step was not implemented, as noise is incorporated in the prediction step which achieves the same objective.

4.3 Grid filter

The Grid filter differs from the Genetic filter and Particle filter in the initialisation, prediction and resampling steps. Instead of using particles, the search space is split into grid squares. Each grid square intersection represents a candidate position of the mobile device and fundamentally, instead of particles moving around and each particle's weight being updated, weights are moved throughout the grid to represent the posterior distribution of the device's state. This algorithm has also been explored by Zhong & Groves (2022). The process follows the following steps, this is also shown in Figure 3.

Initialisation – The search area is initialised as a square grid of a given size κ (in mm) centred around the initial position estimate, this is then split into two-dimensional (2D) grid squares with grid spacing, η (in mm), to form an array of dimensions, κ/η . Following this, the grid squares are weighted according to their distance to the centre of the grid (the initial position estimate) according to a Gaussian distribution with a standard deviation that matches the sensor noise. For this paper, η was set as 50 mm and κ was initialised as 2 m.

Prediction – For positioning in motion, the PDR model described in Section 3 is used to retrieve the displacement and heading values. However, the position displacement differs from that of the Particle filters to ensure that the coordinates remain in a grid formation. The displacement of all grid coordinates is determined as described by

$$X_{k+1}^n = X_k^n + \cos(\bar{\varphi}_k) \bar{r}_k$$

$$Y_{k+1}^n = Y_k^n + \sin(\bar{\varphi}_k) \bar{r}_k \quad (16)$$

where the mean values of the heading, $\bar{\varphi}_k$, and displacement, \bar{r}_k , are used to displace the grid.

However, this does not effectively account for sensor noise. Thus, a separate process must be followed to incorporate sensor noise into the grid coordinates and weights, which is described as follows. First, a new set of grid coordinates 'c grid' is generated that are equal to the grid coordinates of the main grid. A multivariate Gaussian distribution is generated from the heading uncertainty and step length uncertainty. From this distribution, a new set of weights are generated for the 'c grid' by calculating the probability density function of the distribution for the distances between each coordinate and the mean of the coordinates. With the weights of the 'c grid' and the weights of the grid coordinates from the previous epoch, the weights are convolved using direct convolution to form a new set of weights for the grid coordinates that incorporate the sensor noise of the PDR.

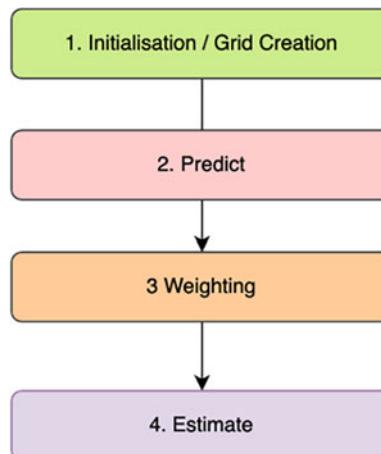


Figure 3. Grid filter process.

Weighting – Each grid point is weighted according to its distance from all of the landmarks. The model used for the weighting is the same as the Particle filter and Genetic filter, described in Equation (7). This weighting represents the likelihood that the device is at that grid point. Following this, the weights are normalised.

Using the weights of each grid point, the weighted average position is determined according to Equations (10) and (11). This represents the estimated position of the mobile device at that epoch.

Repeat the Prediction, Weighting and Estimation steps for each epoch.

5. RSSI-based outlier detection

All positioning algorithms explored in this paper were also augmented with a new environment-agnostic RSSI-based outlier detection model. The algorithm detects NLOS signals and severe multipath interference by finding and accounting for inconsistencies between the RTT range measurement and the RSSI of the measurement. Guo et al. and Sun et al. have also used RSSI for outlier detection; however, these algorithms were optimised for the environments by conducting surveys of the RSSI path loss model and fingerprinting. The environment-agnostic RSSI-based outlier detection model proposed in this paper is as follows.

1. RSSI-threshold determination:

$$\begin{aligned}
 R_{i,k}^{\text{threshold}} &= -(51.4 + 20 \log_{10}(\bar{d})) \quad \text{where} \quad d_{\min} < \bar{d} < 8 \\
 R_{i,k}^{\text{threshold}} &= -(65.5 + 33 \log_{10}(\bar{d}) / 8) \quad \text{where} \quad \bar{d} > 8
 \end{aligned} \tag{17}$$

- Using Equation (17), for each access point, i , the expected RSSI is computed given that distance, giving the RSSI threshold, $R_{i,k}^{\text{threshold}}$, where i represents the landmark, k represents the epoch, \bar{d} represents the median measured RTT distance between an AP and the mobile device across 2 seconds worth of data, and d_{\min} is a minimum distance, below which RSSI-based outlier detection is not used. It was found during experimentation that the relationship between RSSI and RTT measurements below 4 m was misleading; thus, RSSI-based outlier detection is not conducted below a certain RTT range measurement. Equation (18) was derived from Bensky

(2016), but modified slightly to account for its use as threshold detection by increasing the constant value. Optimising these values for each specific environment is a potential research focus, but for the purpose of this paper, this was not done to keep the model environment-agnostic.

3. During the filtering process, following the prediction step but before the update step, each AP is compared against the measured RSSI threshold of that AP. If the measured RSSI is lower than the threshold, then this particle is treated with lower confidence and the absolute difference between the measured RSSI and threshold is taken, δ_{rss_i} . For each epoch, these differences are normalised to produce an RSSI-based weighting factor, ε_k^i , for AP, i , and the epoch, k .
4. ε_k^i are then used during the update step as a multiple to the standard deviation of the measured distances, this increases the measurement noise of the measurements with an RSSI below the threshold with the magnitude of the difference creating greater measurement noise. The standard deviation is computed in the following equation:

$$\sigma_i^n = \sigma \times (1 + \varepsilon_k^n) \quad (18)$$

This can then be used to obtain the standard deviation, σ_i^n , used in Equation (7) for all filters. For this paper, σ , was set at 1 m.

6. Experimental tests

6.1 Methodology

The experimental layout for static positioning is described in detail by Raja and Groves (2023). The experiments to test the algorithms in motion involved moving through different routes in a single environment. The environment was irregular in shape and had many walls, both interior and exterior, to create a diverse range of multipath and NLOS scenarios during the routes walked by the pedestrian. The algorithms were tested across three different routes, varying in complexity and NLOS conditions; these are shown in Figure 4. Each route was repeated in both directions, so in total, six scenarios are collected. In Figure 4, the ‘forward’ route starting point is shown with the purple square and the ‘reverse’ route starting point is shown with the red square. The trials involved a pedestrian holding a smartphone and walking on top of fixed step markers placed on the ground along the route of the trial. The markers were approximately 670 mm apart. The location of each marker was measured against the reference origin to generate the step marker’s ground truth coordinates. To align the ground truth data with the measured data, the trials were filmed. The timestamps of the steps from the video were used to determine the expected position of a device at a given time, allowing the algorithms to have an accurate comparison point at each step. The standard deviation of the step length assuming white noise was calibrated as follows. A pedestrian carrying the mobile device walks 20 steps in a straight line 10 times. The real distance travelled is measured and the estimated distance travelled is calculated using the PDR model. The standard deviation of the difference is computed and this is divided by the number of steps to give the standard deviation of a step.

6.2 Experimental results

6.2.1 Static positioning

The preliminary results of the static data were presented at the ION GNSS+ 2023 conference by Raja and Groves (2023), however, the Grid filter algorithms have since been improved and will be discussed here.

The RMSE of the positioning solution for each environment is shown in Table 1. Each row refers to the algorithm being used with and without outlier detection, whilst each column refers to the trial. Each trial is also marked if it had NLOS scenarios. Here, 27 out of 42 or 64.3% of tests produced sub-metre

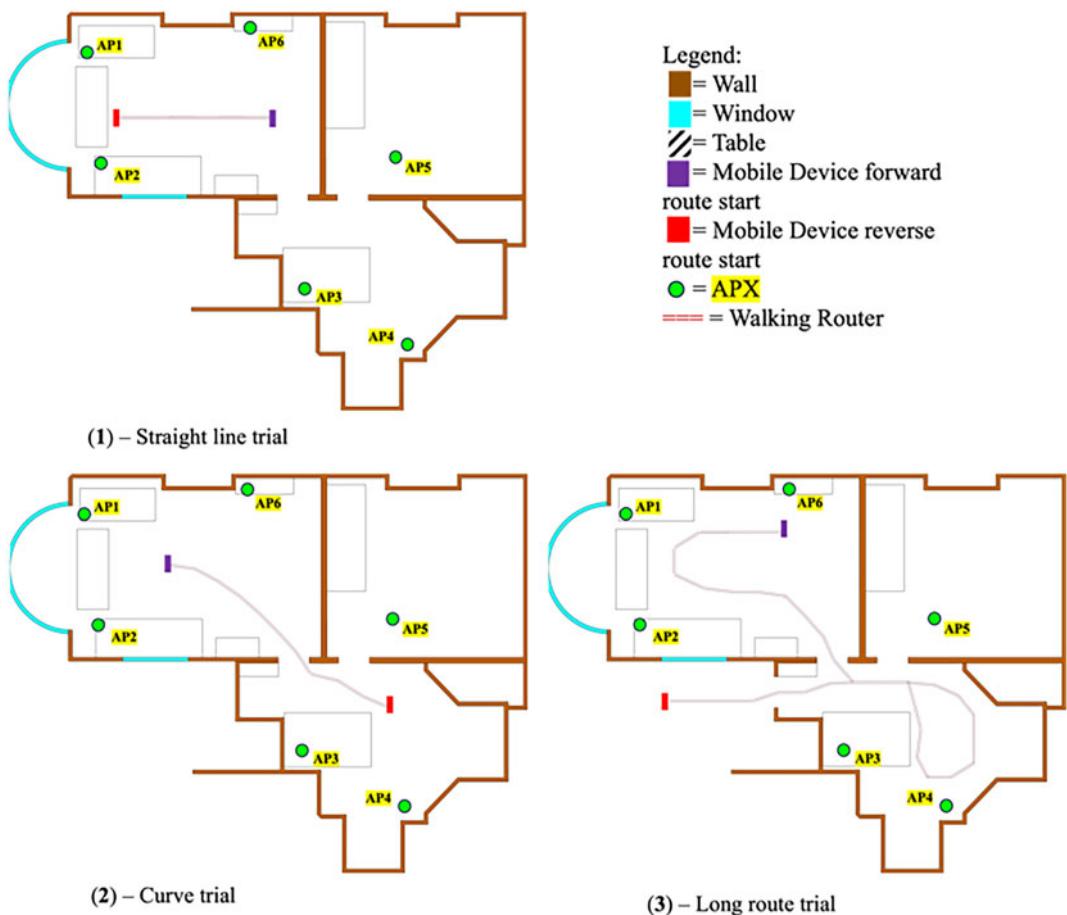


Figure 4. Positioning in motion experimental environments.

Table 1. Positioning solution RMSE for trials and algorithms configuration in static

| Environment | A (NLOS) | B (LOS) | C (LOS) | D (LOS) | E (NLOS) | F (NLOS) |
|-------------------------------------|----------|---------|---------|---------|----------|----------|
| Algorithm | RMSE (m) | | | | | |
| Single epoch least squares | 2.19 | 1.25 | 0.93 | 0.54 | 1.26 | 1.17 |
| Particle filter | 1.09 | 0.41 | 0.40 | 0.46 | 1.24 | 1.20 |
| Particle filter + outlier detection | 0.71 | 0.46 | 0.35 | 0.54 | 1.15 | 0.75 |
| Genetic filter | 1.16 | 0.58 | 0.38 | 0.43 | 1.12 | 0.88 |
| Genetic filter + outlier detection | 0.85 | 0.51 | 0.35 | 0.54 | 0.62 | 0.56 |
| Grid filter | 1.09 | 0.28 | 0.52 | 0.44 | 1.19 | 1.17 |
| Grid filter + outlier detection | 0.45 | 0.34 | 0.48 | 0.46 | 1.10 | 1.07 |

accuracy and 41 out of 42 or 97.6% of trials had an RMSE below 2 metres, indicating there is a strong argument for WiFi RTT being able to produce sub-metre accuracy for positioning, as pitched by Google (2022), as long as the AP biases are calibrated. Environments B, C and D where all APs had a LOS to the mobile device produced sub-metre accuracy for 20 out of 21 trials, indicating that in optimal conditions (with calibration), WiFi RTT could provide a reasonable positioning solution for most pedestrian navigation use cases. In environments A, E and F where there were NLOS signals present,

Table 2. Percentage decrease of RMSE against least squares for each environment in the static case and algorithm configuration

| Environment | A (NLOS) | B (LOS) | C (LOS) | D (LOS) | E (NLOS) | F (NLOS) | Mean percentage improvement (%) |
|-------------------------------------|-------------|------------|------------|------------|-------------|-------------|---------------------------------|
| Algorithm | | | | | | | |
| Particle filter | 50 | 67 | 57 | 16 | 2 | -3 | 32 |
| Particle filter + outlier detection | 67 | 63 | 62 | 1 | 9 | 36 | 40 |
| Genetic filter | 47 | 54 | 59 | 21 | 11 | 25 | 36 |
| Genetic filter + outlier detection | 61 | 59 | 62 | 0 | 51 | 52 | 48 |
| Grid filter | 50 | 78 | 45 | 19 | 6 | 0 | 33 |
| Grid filter + outlier detection | 80 | 73 | 49 | 16 | 13 | 9 | 40 |
| Mean percentage improvement | 59 | 66 | 56 | 12 | 15 | 20 | |

only 7 out of 21 trials achieved sub-metre accuracy. This is to be expected as NLOS and multipath effects vary substantially from environment to environment.

Table 2 shows the percentage improvement of each algorithm combination against the baseline least squares algorithm, the mean percentage improvement of each algorithm across all trials is shown on the rightmost column. In environments C and D, all of the filters improved the positioning accuracy over least squares. This is also true for environments A and B. However, in environment F, the Particle filter and Grid filter without outlier detection performs worse than least squares. The Grid filter with outlier detection produced the highest improvement overall of 80% in environment A. The algorithms provided the greatest mean improvement in environment B; this is the simplest environment with the smallest distances between the APs and mobile devices, and no NLOS signals, and would be expected to have the highest accuracy given there are fewer error sources when compared with more complex environments like environments E and F. The best performing algorithm on average across all environments was the Genetic filter with and without outlier detection with 48% and 36% mean percentage improvement over single epoch least squares, respectively.

6.2.2 Positioning in motion

In this section, the algorithms will be analysed for all scenarios where the mobile device was in motion. Each scenario going forward and in reverse will be analysed separately. Table 3 summarises the results of the trials such as the minimum positioning solution RMSE at any step in the trial, the maximum positioning solution RMSE at any step in the trial, the average positioning solution RMSE of all steps in the trial and the final positioning estimate RMSE of a trial. The average column is of particular note.

As shown in Table 3, the least squares solution positioning accuracy is poorer than the other algorithms. The average positioning RMSE for least squares was 2.79 m when taking an average across all trials. For all trials except Trial 2R, least squares was the worst performing algorithm based on the average RMSE across all steps. Furthermore, for all trials except Trials 2R and 3R, the final position RMSE was highest for least squares. Observing the least squares position estimates in the maps shown in Figures 5, 6, 7, 8, 9 and 10, it is clear that irrespective of the results in Table 3, the position estimates do not correctly follow the pedestrian's path and are visibly inaccurate, placing the pedestrian in an incorrect room on most occasions, which demonstrates the importance of the PDR model in the positioning algorithm. The following analysis will exclude least squares due to the reasons described above.

Table 3. RMSE position error statistics for trials in motion and each algorithm combination

| Environment | Statistic over all steps for trial | Least squares | Particle filter | Particle filter + RSSI | Genetic filter | Genetic filter + RSSI | Grid filter | Grid filter + RSSI |
|--|------------------------------------|---------------|-----------------|------------------------|----------------|-----------------------|-------------|--------------------|
| Trial 1F RMSE (m) | Minimum position | 0.58 | 0.36 | 0.44 | 0.41 | 0.38 | 0.41 | 0.24 |
| | Maximum position | 5.25 | 0.79 | 0.58 | 0.64 | 0.78 | 1.16 | 1.08 |
| | Average position | 3.33 | 0.65 | 0.50 | 0.50 | 0.51 | 0.85 | 0.75 |
| | Final position | 3.99 | 0.47 | 0.48 | 0.57 | 0.69 | 0.41 | 0.25 |
| Trial 1R RMSE (m) | Minimum position | 2.47 | 2.34 | 2.04 | 2.25 | 2.09 | 2.31 | 2.37 |
| | Maximum position | 4.73 | 3.11 | 3.11 | 3.11 | 3.11 | 3.11 | 3.11 |
| | Average position | 3.37 | 2.53 | 2.25 | 2.44 | 2.32 | 2.55 | 2.66 |
| | Final position | 3.37 | 2.25 | 2.55 | 2.68 | 2.46 | 2.66 | 2.76 |
| Trial 2F RMSE (m) | Minimum position | 0.97 | 0.57 | 0.27 | 0.83 | 0.42 | 0.56 | 0.57 |
| | Maximum position | 8.49 | 1.64 | 1.62 | 1.62 | 1.62 | 1.76 | 1.64 |
| | Average position | 3.03 | 1.19 | 0.90 | 1.28 | 0.97 | 1.29 | 1.19 |
| | Final position | 8.49 | 0.78 | 0.30 | 1.15 | 0.67 | 0.56 | 0.55 |
| Trial 2R RMSE (m) | Minimum position | 0.22 | 1.06 | 0.96 | 1.12 | 0.96 | 1.74 | 1.70 |
| | Maximum position | 3.01 | 3.01 | 3.01 | 3.01 | 3.01 | 3.01 | 3.01 |
| | Average position | 1.76 | 1.43 | 1.41 | 1.48 | 1.50 | 2.32 | 2.30 |
| | Final position | 2.41 | 1.78 | 1.86 | 1.89 | 2.10 | 2.42 | 2.41 |
| Trial 3F RMSE (m) | Minimum position | 0.58 | 0.28 | 0.26 | 0.10 | 0.15 | 0.52 | 0.36 |
| | Maximum position | 9.81 | 3.13 | 3.01 | 3.39 | 3.23 | 4.22 | 4.22 |
| | Average position | 2.74 | 1.11 | 1.02 | 1.16 | 1.05 | 1.86 | 1.77 |
| | Final position | 4.00 | 1.00 | 0.78 | 1.14 | 0.87 | 2.51 | 2.37 |
| Trial 3R RMSE (m) | Minimum position | 0.11 | 1.05 | 0.89 | 0.80 | 0.88 | 0.15 | 0.23 |
| | Maximum position | 8.33 | 2.76 | 2.58 | 2.63 | 2.64 | 2.03 | 1.90 |
| | Average position | 2.48 | 2.06 | 1.75 | 1.93 | 1.69 | 1.28 | 1.27 |
| | Final position | 1.43 | 1.51 | 1.15 | 1.77 | 1.30 | 0.77 | 0.76 |
| Average of Average across all trials position RMSE (m) | | 2.79 | 1.49 | 1.31 | 1.47 | 1.34 | 1.69 | 1.66 |

In 8 out of 36 trials with filtered solutions, the average position RMSE over all steps of a given trial was sub-metre and in 29 out of 36 trials, the average position RMSE was below two metres. In virtually all trials, the PF and GrF performed similarly to each other, and the GrF performed worse on average in comparison to the other algorithms when looking at average RMSE. All algorithms in Trial 1R had a positioning RMSE greater than 2 m. For Trial 1R, as can be seen in Figure 6, the paths of all filters are translated approximately 2.5 m to the north of the actual path. This positioning error is worse than Trial 1F for all algorithms. The poor accuracy of the algorithms in comparison could be a result of poor initialisation which could be a result of fewer LOS APs during the earlier part of the trial. This discrepancy is potentially caused by the geometry of the APs, as shown in Figure 6, in combination with the pedestrian's path. Practically, these kinds of poor initialisation errors can be resolved with better PDR algorithms that account for error in the orientation sensor as well as access to more user-specific calibration data for step length. In this trial, the pedestrian blocks AP1 and AP2 with their body. This results in attenuation and, since the range is lower than 4 m, the outlier detection algorithm will not de-weight the range estimate and without outlier detection, the algorithms will treat the ranges as reliable, even though they are NLOS due to the pedestrian's body blocking the signals. This threshold could be calibrated and varied to account for these scenarios, but this has not been explored in this paper.

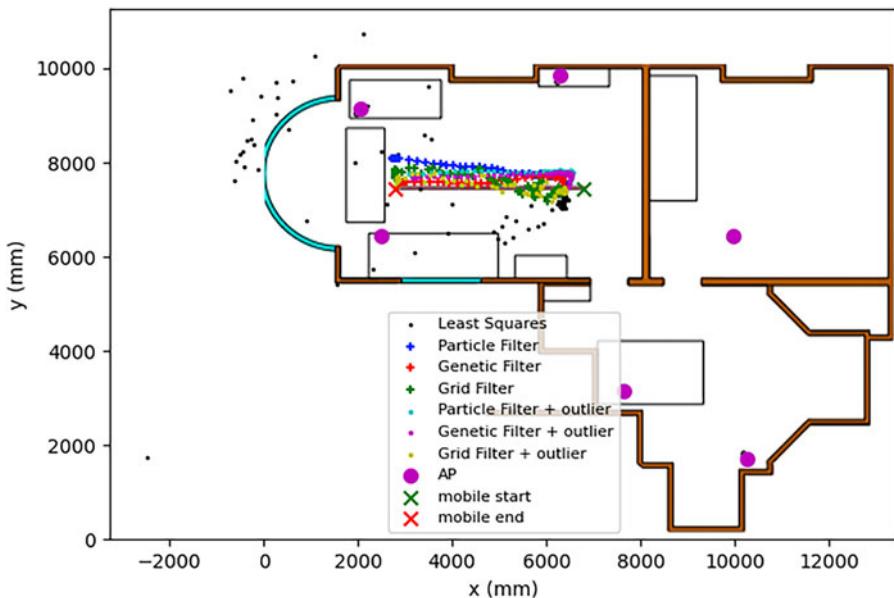


Figure 5. Trial 1F position per epoch.

Table 4. Initial position RMSE for each trial in motion

| Trial ID | Initial position RMSE (m) |
|----------|---------------------------|
| Trial 1F | 0.58 |
| Trial 1R | 3.11 |
| Trial 2F | 1.62 |
| Trial 2R | 3.01 |
| Trial 3F | 1.44 |
| Trial 3R | 1.85 |

As shown in Table 4, the initial position RMSE was 3.11 m, which is approximately 5.4 times the initial position RMSE of Trial 1F. This will drastically reduce the accuracy of the results from all algorithms. The Grid filter in Trial 2R also had an average position RMSE greater than 2 m. In Figure 8, this is evident as, compared with the other algorithms, the path of the pedestrian is estimated at a different location. In Table 4, the initial position RMSE was greater than 3 m, contributing to the overall positioning inaccuracy of the trial. For all algorithms, the larger positioning error at the final steps of Trial 2R is due to a poor orientation value at step 7, resulting in an incorrect trajectory of the pedestrian, demonstrating a shortfall in the orientation feature of the PDR model. Poor orientation readings also impacted the performance of the algorithms in 3F and 3R, demonstrating that improving the orientation sensor or orientation processing model could improve the overall positioning accuracy of the algorithms. These errors do not accumulate linearly with step count. The system is mostly impacted by anomalies in the readings, be this from the step length being incorrect or the orientation being incorrect for a given prediction step.

In 33 out of 36 trials, the final position estimate was more accurate than the initial position estimate. This indicates that not only were the algorithms successful at maintaining the path of the pedestrian, they also improved the accuracy of the position estimate compared with the initial position estimate. With outlier detection included, the greatest absolute improvement was made by the PF in Trial 2F with an improvement of 1.32 m and the greatest improvement without outlier detection was made by the PF

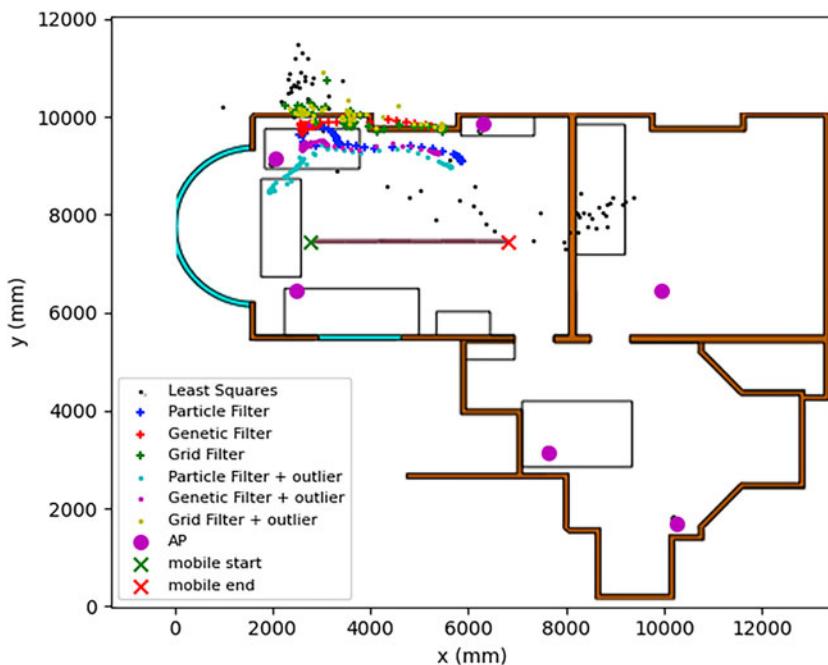


Figure 6. Trial 1R position per epoch.

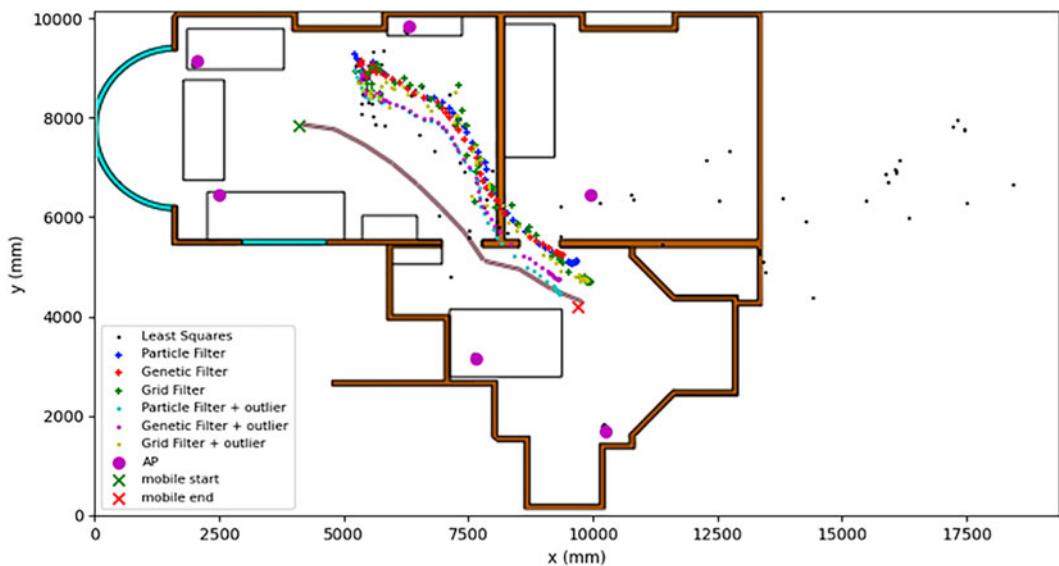


Figure 7. Trial 2F position per epoch.

in Trial 2R with an improvement of 1.23 m. The greatest percentage improvement between the initial and final position estimates without outlier detection was made by the GrF in Trial 2F with an improvement of 287%.

Trials 3F and 3R were the longest trials with a combination of different manoeuvres by the pedestrian. These trials consisted of 35 steps with the pedestrian also walking outside of the building. Figure 9 shows a trial run on the map when the device is moving through the forward scenario

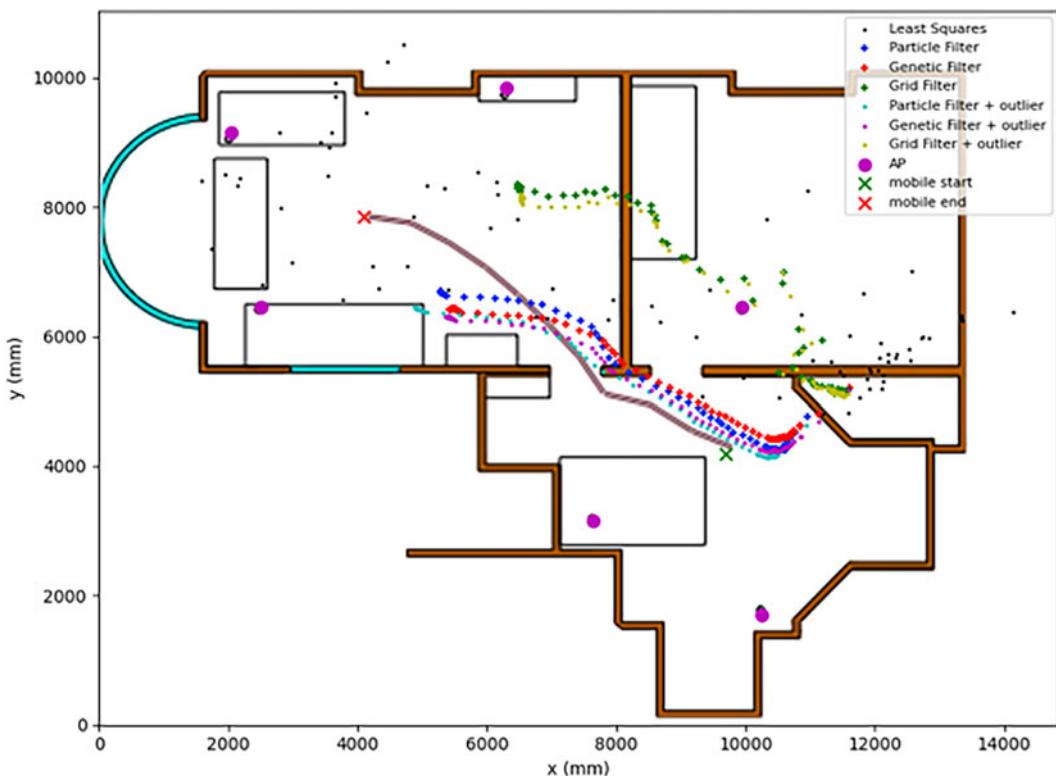


Figure 8. Trial 2R position per epoch.

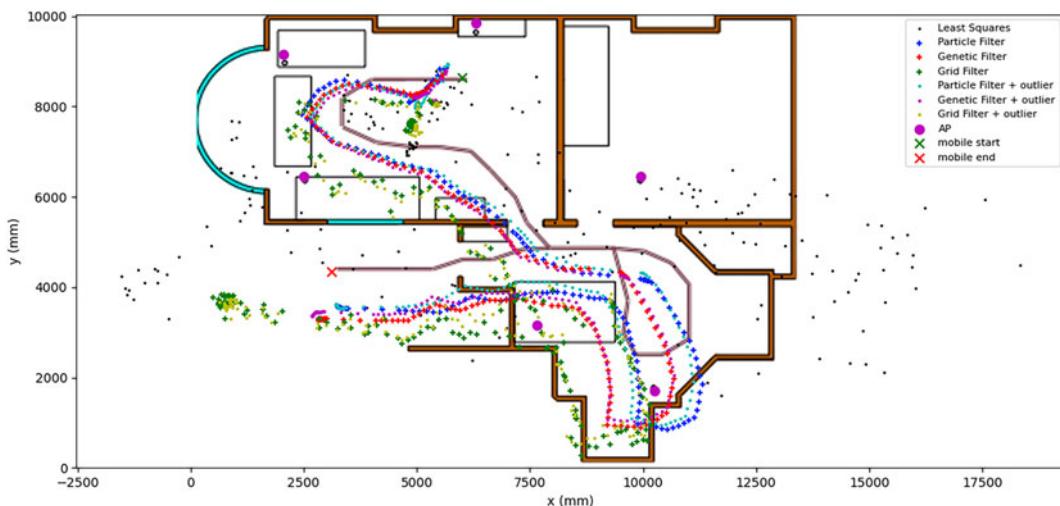


Figure 9. Trial 3F position per epoch.

(Trial 3F). Figure 10 shows a trial run on the map when the device is moving through the reverse scenario (Trial 3R), beginning outside of the building. In the case of Trial 3F, the GrF final position RMSE is almost double that of the other filters with an RMSE of 2.51 and 2.37 m for the GrF without and with RSSI-based outlier detection, respectively, and from Figure 9, the path is also visibly incorrect, unlike the other filters. However, in Trial 3R, it appears that the GrF better guides the PDR model

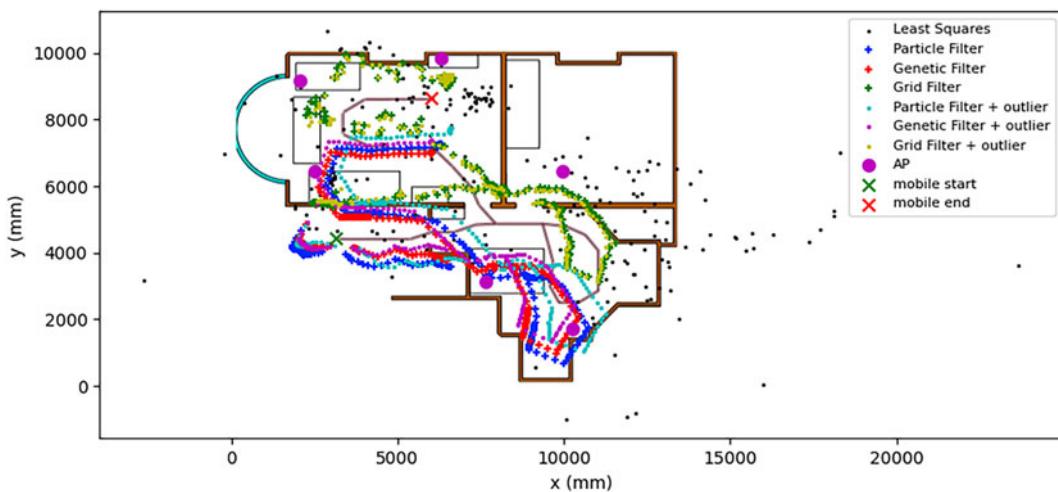


Figure 10. Trial 3R position per epoch.

Table 5. RMSE percentage position error improvement for algorithms and trials in motion

| Trial ID | Particle filter + RSSI (%) | Genetic filter + RSSI (%) | Grid filter + RSSI (%) |
|----------|----------------------------|---------------------------|------------------------|
| Trial 1F | 24 | -1 | 11 |
| Trial 1R | 11 | 5 | -4 |
| Trial 2F | 24 | 25 | 8 |
| Trial 2R | 1 | -1 | 1 |
| Trial 3F | 8 | 9 | 5 |
| Trial 3R | 15 | 13 | 0 |
| Average | 14 | 8 | 3 |

through the path and most accurately represents the path of the user. The final position estimate for the GrF in this trial was sub-metre both with and without outlier detection.

Table 5 shows the RMSE percentage improvement of the average positioning error for each trial for the algorithms with and without RSSI-based outlier detection. As can be seen in 15 out of 18 trials, the RSSI-based outlier detection improved the positioning solution. Any outlier detection will have a false positive rate, so these results are quite promising as the decrease in accuracy from the expected false positive rate does not result in a drastic change in accuracy. The largest improvement was with the GrF in Trial 2F, where the outlier detection provided an improvement of 25%. The outlier detection provided the greatest improvement for the Particle filter with an average improvement of 14%, whilst the outlier detection provided the least improvement to the GrF with an average of 4%. A large limitation of the RSSI-based outlier detection algorithm is caused by an edge case where the increased WiFi-RTT measurement matches with the RSSI path loss propagation model measured range. This results in the reading not being picked up by the RSSI-based outlier detection model. As a result, the erroneous signal will reduce the accuracy and reliability of the overall positioning solution. Furthermore, the material of the blockage (brick wall, body, wooden door etc.) varying also has an impact on the RSSI property of the signal compared with the WiFi RTT measured range. This property will depend on the attenuating properties of the material.

7. Conclusions

Overall, the results demonstrate that for WiFi RTT-based positioning, filtering techniques provide a superior positioning solution to basic least squares for positioning, and the addition of RSSI-based outlier detection provides an improved positioning solution in NLOS environments. The overall average position RMSE for least squares was 1.23 m. The best performing algorithm without outlier detection on average was the Genetic filter with a positioning RMSE of 0.76 m; an improvement of 36% over least squares. The best performing algorithm with outlier detection on average was the Genetic filter, with a positioning RMSE of 0.57 m; an average improvement of 48% over single-epoch least squares for the RMSE compared with the Particle filter and Grid filter with outlier detection, which both provided an average improvement of 40%. This filter provided an improved method for handling particle degeneracy during the filtering process when compared with the Particle filter. Between the filters, the Grid filter seemed to perform inaccurately in more complex environments, but very accurately in less complex environments.

When the algorithms were tested in motion-based scenarios with PDR, the results were less consistent and the best performing algorithm varied per trial. The overall average position RMSE for least squares across all trials was 2.79 m. The best performing algorithm without outlier detection on average was the Genetic filter with a positioning RMSE of 1.47 m; an improvement of 47% over least squares. The best performing algorithm with outlier detection on average was the Particle filter, with a positioning RMSE of 1.31 m; an average improvement of 53% over single-epoch least squares for the RMSE. The RSSI-based outlier detection improved accuracy, but to a lesser degree than in the static experiments. The conclusions from these trials were that WiFi RTT can be fused with sensors available in a mobile device to produce an integrated navigation solution that can achieve sub-metre accuracy for a moving pedestrian. If WiFi RTT were more widely supported across existing access points, and mobile devices had access to a database of their locations and associated biases, the methods explored in this paper could be implemented directly on smartphones. However, poor orientation readings and, to a certain extent, poor step length estimation can result in a degraded path estimation due to over-reliance on the PDR at the prediction step.

Competing interests. The authors declare none.

References

Android [1]. (2023). Sensor. https://developer.android.com/reference/android/hardware/Sensor#TYPE_STEP_DETECTOR

Android [2]. (2023). Position sensors. https://developer.android.com/guide/topics/sensors/sensors_position#sensors-pos-orient

Arulampalam, M., Maskell, S., Gordon, N. and Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, **50**, 174–188.

Bensky, A. (2016). *Wireless Positioning Technologies and Applications*. 2nd ed. Artech House.

Gentner, C. and Avram, D. (2021). WiFi-RTT-SLAM: Simultaneously Estimating the Positions of Mobile Devices and WiFiRTT Access Points. In: *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), St. Louis, Missouri*, 3142–3148. <https://doi.org/10.33012/2021.18081>

Gentner, C., Ulmschneider, M., Kuehner, I. and Dammann, A. (2020). WiFi-RTT Indoor Positioning. In: *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, Oregon*, 1029–1035.

Google. (2022). Wi-Fi Location: Ranging with RTT. <https://developer.android.com/guide/topics/connectivity/wifi-rtt>.

Groves, P.D. (2013). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. 2nd ed. Artech House.

Guo, G., Chen, R., Ye, F., Peng, X., Liu, Z. and Pan, Y. (2019). Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS Ranging Approach. *IEEE Access*, **7**, 176767–176781. <https://doi.org/10.1109/ACCESS.2019.2957753>.

Hashem, O., Youssef, M. and Harras, K. A. (2020). WiNar: RTT-Based Sub-meter Indoor Localization using Commercial Devices. In: *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom), Austin, TX, USA*, 1–10, <https://doi.org/10.1109/PerCom45495.2020.9127363>.

Higuchi, T. (1997). Monte Carlo Filter Using the Genetic Algorithm Operators. *Journal of Statistical Computation and Simulation*, **59**, 1–23.

Ma, C., Wan, C., Chau, Y. W., Kang, S. M. and Selviah, D. R. (2017). Subway Station Real-Time Indoor Positioning System for Cell Phones. In: *Proceedings of International Conference on Indoor Positioning Indoor Navigation*, 1–7. <https://doi.org/10.1109/IPIN.2017.8115912>.

Mahony, R., Hamel, T. and Pflimlin, J.-M. (2008). Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control*, **53**, 1203–1218.

Malkos, S. and Hazlett, A. (2014). Enhanced WiFi Ranging with Round Trip Time (RTT) Measurements. In: *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, Florida*, 108–116.

Mikov, A., Moschevikin, A., Fedorov, A. and Sikora, A. (2013). A Localization System Using Inertial Measurement Units from Wireless Commercial Hand-Held Devices. IN: *International Conference on Indoor Positioning and Indoor Navigation*, Montbeliard, France, 1–7. <https://doi.org/10.1109/IPIN.2013.6817924>.

Park, S., et al. (2007). A New Particle Filter Inspired by Biological Evolution: Genetic Filter. *World Academy of Science, Engineering and Technology*, **33**, 83–87.

Raja, K.J. and Groves, P.D. (2023). WiFi-RTT Indoor Positioning Using Particle, Genetic and Grid Filters with RSSI-Based Outlier Detection. In: *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)*, Denver, Colorado, 1644–1655.

Sun, M., Wang, Y., Xu, S., Qi, H. and Hu, X. (2020). Indoor Positioning Tightly Coupled Wi-Fi FTM Ranging and PDR Based on the Extended Kalman Filter for Smartphones. *IEEE Access*, **8**, 49671–49684.

Van Diggelen, F., Want, R. and Wang, W. (2018). *How to Achieve 1-meter Accuracy in Android*. <https://www.gpsworld.com/how-to-achieve-1-meter-accuracy-in-android/>.

Weinberg, H. (2002). Using the adxl202 in Pedometer and Personal Navigation Applications. Analog Devices AN-602 application note, 2.2.

Yu, Y., Chen, R., Chen, L., Guo, G., Ye, F. and Liu, Z. (2019). A Robust Dead Reckoning Algorithm Based on Wi-Fi FTM and Multiple Sensors. *Remote Sensing*, **11**, 504. <https://doi.org/10.3390/rs11050504>

Zhong, Q. and Groves, P.D. (2022). Multi-Epoch 3D-Mapping-Aided Positioning Using Bayesian Filtering Techniques. *NAVIGATION: Journal of the Institute of Navigation*, **69**, 1.

Zhou, N., Lau, L., Bai, R. and Moore, T. (2021). A Genetic Optimization Resampling Based Particle Filtering Algorithm for Indoor Target Tracking. *Remote Sensing*, **13**, 132. <https://doi.org/10.3390/rs13010132>