Personal and Ubiquitous Computing manuscript No.
(will be inserted by the editor)

Smartphone-based real-time indoor positioning using BLE beacons

Robert Riesebos · Andrés Tello · Viktoriya Degeler

Received: date / Accepted: date

Abstract To deal with the degraded performance of Global Navigation Satellite Systems (GNSS) in indoor environments, numerous Indoor Positioning Systems (IPS) have been developed. The rapid proliferation of smartphones has led to many IPSs that utilize positioning technologies that are readily available on modern smartphones; including Bluetooth Low Energy (BLE).

Using radio signals such as BLE in indoor environments comes with a number of challenges that can limit the reliability of the signal. In dealing with these challenges, most existing BLE-based IPSs introduce undesired drawbacks such as an extensive and fragile calibration phase, strict hardware requirements, and increases in the system's complexity. In this paper, an IPS is developed and evaluated that requires minimal setup for indoor environments and has a sufficiently low complexity to be run locally on a modern smartphone.

An extensive exploration of the IPS' parameters was performed. The best performing parameter combinations resulted in a median positioning error of 1.48 ± 0.283 meters, while using the log-distance path loss model for distance estimation and Weighted Centroid Localization with a weight exponent between 2.0 and 3.5 for position estimation. Finally, filtering the position estimates using the proposed confidence indicator resulted in a small, but significant decrease in the positioning error.

Keywords Indoor localization · Indoor positioning system · Bluetooth positioning · Bluetooth Low Energy · Weighted Centroid Localization

Robert Riesebos, Andrés Tello, Viktoriya Degeler
Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence
University of Groningen, The Netherlands
E-mail: rriesebos@hotmail.com, andres.tello@rug.nl, v.degeler@rug.nl

1 Introduction

The Global Positioning System (GPS) alongside other Global Navigation Satellite Systems (GNSS), such as the European Union's Galileo, are a straightforward approach to provide position estimates to users across the globe. However, these systems have a significant limitation: they are not useful when positioning in indoor scenarios is required. This is caused by the fact that the GNSS' signal is not strong enough to penetrate through solid building materials — severely degrading the indoor performance of these systems. Consequently, many Indoor Positioning Systems (IPS) have been developed to fill the gap in the global coverage of satellite-based systems.

The global indoor positioning and navigation market was valued at 6.92 billion dollars in 2020, and is projected to grow to 23.6 billion dollars by 2025 [1]. This evaluation is driven by the many application areas of IPSs, as well as the growing ubiquity of the technologies supporting these systems. In particular, the rapid proliferation of smartphones with support for receiving and transmitting various radio frequency has led to accessible, low-cost solutions. Additionally, these smartphones are directly linked to their users, making positioning of these smart phones synonymous to positioning the corresponding users. This, in turn, enables Internet of Things (IoT) integration and provides opportunities for market research, navigation aid and context-aware assistance.

While there are many different technologies that are used for indoor positioning, only a handful are currently available on modern smartphones. Notable examples include Bluetooth Low Energy (BLE)[1] and Wi-Fi. This paper aims to take full advantage of the ubiquity of modern smartphones and their sensing capabilities. In particular, it focuses on using BLE beacons as reference points to determine the position of a positioning subject that is carrying a smartphone serving as a BLE receiver.

The main challenge of using radio signals such as BLE for indoor positioning is dealing with various effects that compromise the radio signal, decreasing its reliability. Examples of such effects include complicated interference patterns, multipath propagation where the signal reaches the receiver through multiple different paths, and Non-Line-of-Sight (NLOS) conditions in which the signal strength is reduced due to obstacles between the transmitter (beacon) and receiver.

A lot of research has been conducted on dealing with these problems, and many IPSs using BLE have already been developed [2, 3]. However, in dealing with the challenges of using BLE, many solutions introduce undesired drawbacks such as requiring an extensive calibration phase (that is invalidated when the indoor layout changes), strict hardware requirements, and increases in the system's complexity [2]. When the system becomes too complex to be run locally on the smartphone, significant latency between receiving BLE signals

[1] A technology similar to classic Bluetooth but with significantly lower power consumption

and position estimation can be introduced. Additionally, offline positioning becomes infeasible.

The main objective of this paper was to create a BLE-based IPS that avoids these undesired drawbacks. As such an IPS was developed that requires minimal setup for indoor environments, has a sufficiently low complexity that positioning can be done in real-time — locally on the smartphone — and that uses inexpensive, readily-available BLE beacons. To accomplish this, the variance between Received Signal Strength Indicator (RSSI) measurements from BLE beacons was investigated, and methods to deal with this variance were explored. Multiple ways to convert the RSSI measurements into distance estimates were introduced, as well as different methods to approximate the smartphone's position from these distance estimates.

To evaluate the proposed Indoor Positioning System (IPS), the system's parameters were exhaustively explored, and the best performing parameter combinations were further discussed. In short, the main contributions of this paper are:

- An Indoor Positioning System (IPS) with a median positioning error of about 1.5 meters, that runs locally on a smartphone and requires minimal setup;
- A novel confidence indicator for position estimations;
- An exhaustive exploration of the parameters involved in the different stages of the IPS by replaying recorded BLE beacon measurements.

Additionally, this paper also includes explorations of ideas from the literature such as on-device identification of BLE channels and ground truth interpolation.

The remainder of the paper is structured as follows. Section 2 contextualizes this paper, and examines recent related works that also utilize BLE as their primary positioning technology. Next, in Section 3, the implementation of the proposed IPS is discussed. The section starts with a rundown of the architecture of the IPS, and a closer look at BLE. The implementation is broken down in four main steps that are used to structure the section. Section 4 discusses the experiment environment, experiment parameters and how they can be efficiently explored. It also presents the ground truth for the experiments and a method to extend the ground truth when required. Lastly, the error metrics used to evaluate the experiments are introduced. The results of the experiments are presented and discussed in Section 5. Finally, in Section 6, the paper is summarized and opportunities for future work are discussed.

2 Related work

The work by Faragher and Harle [4], written in 2015, is one of the most influential works on BLE-based indoor positioning [2]. It provided the first experimental test of fine-grained BLE positioning using fingerprinting, and it was the first work to show that the use of three advertising channels to transmit

BLE signals leads to severe received signal strength (RSS) variations. In order to mitigate these variations when collecting measurements, they used a time window of RSS measurements. In their experiments, the BLE beacons were initially set to broadcast at a frequency of 50 Hz with a transmit power of 0 dBm. In total, 19 beacons were distributed around an approximately 600 m^2 environment. With this setup, window sizes of around 0.5 to 2.0 seconds — for the measurements window used in the online stage — provided the best performance with a median positioning error of about 1 meter. Further exploration of the parameters revealed that reducing the transmit power to -12 dBm and the broadcast frequency to 10 Hz resulted in a similar median positioning error.

In 2016, Kriz, Maly and Kozel [5] combined BLE fingerprinting with Wi-Fi fingerprinting to improve the overall positioning accuracy. For each technology a separate set of fingerprints was collected and stored. Each fingerprint was taken in a 52 m by 43 m area, in which 17 BLE beacons were deployed. To evaluate the positioning error, a leave-one-out cross-validation technique was applied on the collected fingerprints. From the set of 680 fingerprints, one was chosen in each iteration and its position was estimated based on the distance to the other fingerprints. The results showed a 23% improvement when BLE beacons were used in addition to Wi-Fi access points, and yielded a median positioning error of 0.77 meters. However, the mobile application was only capable of collecting fingerprint measurements, and the system was only evaluated using fingerprints that were constructed from a large number of RSSI samples — the 680 measurements consisted of 115,511 individual RSSI samples. Each measurement took 10 seconds to complete.

In the same year, Zhuang, Yang, Li, Qi and El-Sheimy [6] proposed an algorithm consisting of multiple components in order to provide smartphone-based indoor positioning using BLE beacons. The components include a channel-separate polynomial regression model (PRM), channel-separate fingerprinting (FP), multi-level outlier detection and extended Kalman filtering (EKF). The polynomial regression model was used to provide distance estimates by modelling the relationship between RSS and distance (for the BLE beacons) as an nth-degree polynomial. The polynomial coefficients were estimated by collecting RSS values at several known locations in the experimental setup, and performing least squares fitting. This was done for each advertising channel. For the fingerprinting component, fingerprint databases were constructed for each advertising channel, and for the aggregate of all channels. In the online phase, position estimates were generated for each channel by matching new measurements to the constructed databases. These position estimates were then used in the first outlier detection step, together with the distance estimates from PRM, to obtain an enhanced distance estimate for each observed beacon. Finally, the enhanced distance estimates were supplied to an extended Kalman filter which was used to predict the position of the target. After the EKF, the second outlier detection algorithm, based on statistical testing, is performed to remove remaining outliers from the EKF. The proposed algorithm was evaluated in the corridors of an environment covering 60 by 40

meters, using BLE beacons set to transmit at a frequency of 10 Hz with a transmit power of -16 dBm. Missing ground truth points were generated by interpolating between reference points, using timestamps collected with a stopwatch. The results showed that the mean positioning error averaged over two trajectories was 1.66 meters for a dense distribution of beacons (1 beacon per 9 meter), and 1.98 meters for a sparse distribution of beacons (1 beacon per 18 meter). The median positioning error averaged over both trajectories was 1.59 meters for the dense distribution, and 1.67 meters for the sparse beacon distribution.

Later in 2016, Subedi, Kwon, Shin, Hwang, and Pyun [7] published a paper that focused on using Weighted Centroid Localization (WCL) with BLE beacons. To estimate the distances between the receiver and the BLE beacons from the measured RSSI values, the log-distance path loss model was used. Before the distance was estimated, the RSSI measurements were filtered using a Kalman filter on top of a moving average filter. The developed system was evaluated in a 2.5 meters wide corridor using BLE beacons that broadcast at an interval of 300 ms, or 3.33 Hz, with a transmit power of 4 dBm. In total 14 beacons were placed along the corridor. The placement was done in pairs of two at a height of 2.5 meters, with a distance of 4.5 meters between each pair. The results showed that, out of the considered weight exponents of 0.5, 1.0, and 1.5, a weight exponents of 0.5 performed best for their test environment. The corresponding mean positioning error was about 1.8 meters when all measurement locations were averaged.

Two years later, in 2018, Sadowski and Spachos [8] compared Wi-Fi, BLE, Zigbee, and LoRaWAN for use in an IPS. To estimate the distances to the BLE beacons from the RSSI, the log-distance path loss model was used. After obtaining the distance estimates, trilateration was used to estimate the location of the receiver. For their BLE experiments, three beacons were used. These beacons were placed to form a right-angled triangle with an equal base and height of length d. Three values were considered for the length d: 1 meter, 3 meters, and 5 meters — corresponding to experimental areas of 0.5 m^2, 4.5 m^2, and 12.5 m^2 respectively. For each length, the receiver was placed at three points within the experimental area. The resulting nine configuration were evaluated in two separate environments. Their results showed that the average mean positioning error for all 18 experiments, divided between both environments, was 0.753 meters when using BLE. When only including experiments where d equals 5 meters, the average mean positioning error was 1.151 meters. The authors also compared the average power consumption between the considered technologies, and concluded that Wi-Fi used the highest amount of power utilizing 216.71 mW, while BLE used the least amount of power, consuming only 0.367 mW.

Similar to the work of Zhuang et al. [6] from three years earlier, in 2019, Huang, Liu, Sun, and Yang [9] proposed an indoor positioning method that took advantage of the three separate BLE advertising channels. To separate the advertising channels, BLE beacons were configured to only broadcast on a single channel. For each advertising channel a series of RSS measurements

was performed at distances between 0 and 19.2 meters, with 1.2 meter increments. Using these measurements, three channel-specific distance models were obtained by fitting the data. Before the distance models were used, a data filtering step was performed. In this step, a sliding window was employed to filter the RSS values. Filtering was done by taking the median of the measurements in the sliding window. This data filtering step was performed for each advertising channel. The RSS values obtained from the data filtering step were used to estimate the distances using the distance models. The distances for each separate channel were combined into a single, final distance estimate. Finally, weighted trilateration was used to convert the processed distance estimates into position estimates. The system was evaluated in two environments; a classroom measuring 5 by 10 meters, and an office room of 9 by 12 meters. In each environment four beacons were deployed, one at each corner of the environment. The proposed method achieved median positioning errors between 1.8 and 2.0 meters over three experiments. The two experiments were performed in the classroom, and one in the office room. Furthermore, two different smartphones were used in the classroom experiments.

Finally, in 2020, L. Liu, Li, Yang, and T. Liu [10] published a real-time indoor positioning method that fused positioning estimates obtained by using trilateration and fingerprinting. Additionally, a Pedestrian Dead Reckoning (PDR) approach was explored, and the results from both the BLE-based method and the PDR method were fused using a Kalman filter. The initial position needed in the PDR method was provided by the BLE-based method. The BLE-based method used the log-distance path loss model to convert RSSI measurements to distance estimates. These distance estimates were fed directly into the trilateration positioning method, and the corresponding RSSI values were used in the fingerprinting method. To fuse both methods, a weighted average of both position estimates was taken. To evaluate their system, it was deployed on a 44 by 16 meter university floor. Only a single corridor was utilized. This corridor spanned the entire width of the floor and the majority of the length of this corridor had a width of only 1.7 meters. In total 10 were deployed along the edges of the corridor. These beacons had a broadcast frequency of 1 Hz. In total, two different routes were traversed, and three experiments were performed per route. The results of all six experiments for the BLE-based method were averaged, resulting in a median position error of 2 meters.

Whereas previous works mostly rely on a single positioning method, this paper provides an extensive comparison of multiple different positioning methods; employed in the same environment using the same distance estimates. Similarly, multiple distance estimation models are explored, and the log-distance path loss model is compared to distance estimation models obtained by fitting experimental data. Furthermore, the positioning results are enhanced with a novel confidence indicator. Finally, unlike a substantial portion of previous work, this work does not rely on fingerprinting, and it is evaluated in a non-corridor environment using BLE beacons broadcasting at only 2 Hz.

3 Implementation

Our implementation of the BLE-based IPS consists of four steps, going from detecting the BLE beacons and obtaining RSSI measurements, to determining the position of the smartphone receiver. The steps are illustrated in Figure 1. Each step will be discussed in a separate sub-section. Section 3.3 describes

Fig. 1: Steps of an RSSI-based indoor positioning system

the RSSI measurements and their characteristics, Section 3.4 deals with filtering the RSSI measurements to decrease variance, Section 3.5 explains how the filtered RSSI is converted to a distance estimation, and finally, Section 3.6 describes how the estimated distances are used to obtain a position estimation. Before discussing each step, the architecture of the IPS is explained in Section 3.1, and a closer look at the BLE technology is provided in Section 3.2.

3.1 Architecture

The architecture of the system consists of four main components: the Android application, the Express RESTful API, the Cassandra database and the MongoDB database. These components are shown in Figure 2, and elaborated upon in the following subsections.

3.1.1 Android Application

The Android application implements all the steps listed in Figure 1. It continuously scans for BLE beacons. When beacons are detected they are displayed in a beacon list with the corresponding information. The estimated position of the smartphone is also continuously calculated using the positioning methods discussed in Section 3.6. The results of these calculations are listed separately for each positioning method in the "Positioning" tab.

The app bar contains a "Start Recording" button. When this button is pressed the application starts recording the RSSI values for the detected beacons along with their timestamps, the calculated distances, the estimated position (using the configurable default positioning method), and an advertising channel estimation. These measurements are send to the Express back-end.

Furthermore, when recording is started, a "Checkpoint" button is shown along with a checkpoint counter. Whenever the checkpoint button is pressed

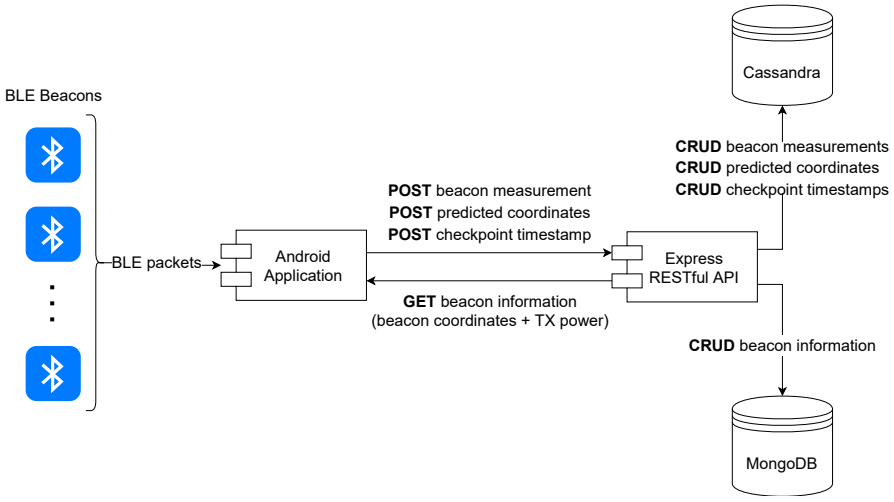


Fig. 2: Indoor Positioning System's architecture

the checkpoint counter is incremented, and the last checkpoint timestamp is send to the back-end[2]. Finally, when the user is done recording they can tap the button in the app bar again to stop recording.

All the parameters related to the measurements window, RSSI filtering, distance estimation, positioning and more can be configured in the settings.

The calculations are done on the device itself, as they are not too resource intensive. This has several advantages, including saving transmission bandwidth, not requiring the continuous connection to the Express back-end server, and allowing for faster positioning results.

3.1.2 Express RESTful API

The Express back-end server serves as a RESTful API that provides a CRUD interface for the two databases. It handles beacon-related calls such as the beacon coordinates and RSSI measurements, as well as calls related to positioning information such as position estimates and checkpoint timestamps.

3.1.3 Databases

The application employs two databases, each serving its own purpose and each based on the most suitable system: one is based on Cassandra, and the other one on MongoDB.

The Cassandra database is used to store time-series data. Specifically, the beacon measurements, predicted coordinates and checkpoint timestamps. These

[2] The purpose of the checkpoints and their timestamps is explained in Section 4.1.3.

are stored in three separate tables. The beacon measurements table stores the beacon identifier, timestamp, RSSI and distance. The predicted coordinates are stored along with a confidence indicator, and the checkpoint timestamps are stored along with the checkpoint identifier.

The MongoDB database has a so-called collection, in which the beacon information is stored. The beacon information consists of the beacon identifier, the TX power (transmission power) and the beacon coordinates (x, y).

3.2 Bluetooth Low Energy technology

Bluetooth Low Energy is a technology that operates in the 2.4 GHz band. Unlike classic Bluetooth technology that uses seventy-nine 1 MHz wide channels, Bluetooth Low Energy uses forty channels that are 2 MHz wide. The different channels are depicted in Figure 3 (adapted from [11]). There are two types

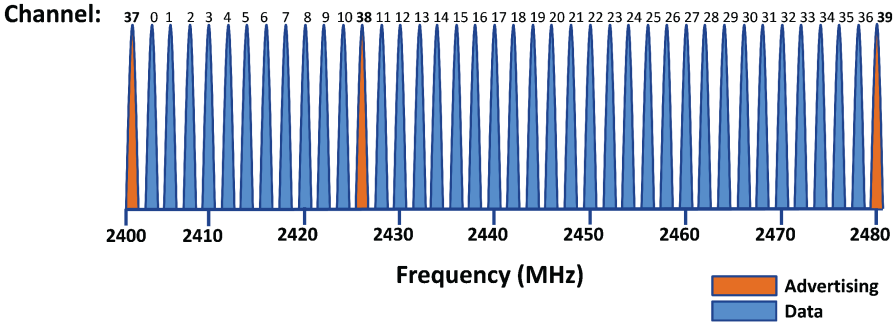


Fig. 3: Bluetooth Low Energy channels [11]

of channels: data channels and advertising channels. The advertising channels are used to broadcast advertising packets. Three advertising channels exist in order to counter-act interference. Specifically, channels 37, 38 and 39 with frequencies of 2402 MHz, 2426 MHz, and 2480 MHz are used. The remaining channels are intended for data transfer.

3.2.1 Bluetooth Low Energy beacons

BLE beacons are hardware transmitters that use the BLE technology to broadcast advertising packets. Consequently, BLE beacons only utilize the advertising channels. Each beacon transmits an advertising packet on all three channels simultaneously[3], with a constant delay between each broadcast called the advertising interval. The receiver listens to only a single channel at once, for a

[3] Some beacons can be configured to only transmit on specific advertising channels.

duration called the scan window. This scan window is periodically repeated every scan interval.

The BLE beacons used in this paper are shown in Figure 4. A total of ten of these beacons were used. These beacons use the iBeacon protocol, introduced by Apple in 2013 [12]. In accordance with this protocol, the beacons transmit a unique identifier and a transmission power (TX power) value indicating the signal power at a reference distance of one meter. The BLE beacons are not

(a) Flat (b) Standing

Fig. 4: BLE beacons

time-synchronized, meaning they do not support Time of Flight-based signal properties. Furthermore, the Angle of Arrival of the signal is not accessible since neither the receiver nor the BLE beacons used for this paper support the latest Bluetooth 5.1 specification that requires an in-built antenna array [13]. Because of these limitations, the only available signal property is the Received Signal Strength Indicator (RSSI).

3.3 RSSI measurements

To receive the advertising packets broadcast by the BLE beacons, a smartphone is used. In particular the OnePlus 6 is used, which supports Bluetooth 5.0 [14]. The scanning interval between measurements is 500 ms or 2 hertz, the maximum frequency supported by the available BLE beacons.

3.3.1 Variance in RSSI measurements

To get an idea of the variance of the RSSI, we took 100 measurements of the RSSI at a static distance of one meter. Then, to better visualize the variance in the RSSI, we took a rolling average and the corresponding rolling standard

deviation with a window size of 30 measurements. The results are shown in Figure 5, where the blue line indicates the rolling average, and the shaded area indicates the standard deviation. For this specific beacon, the average RSSI

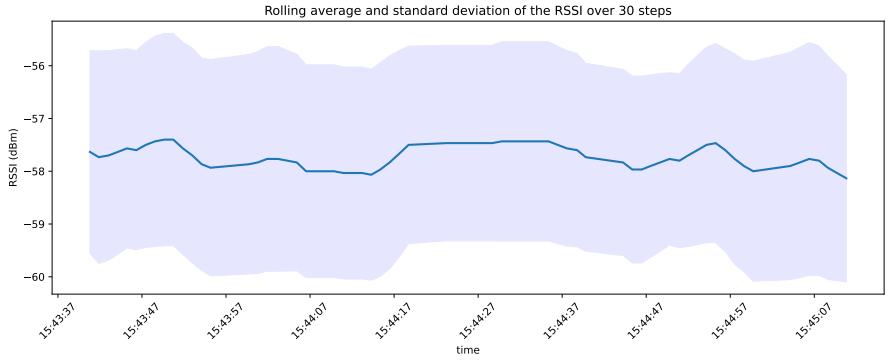


Fig. 5: Rolling average and standard deviation of the RSSI over time

over the 100 measurements was -57.7 dBm with a standard deviation of 1.94 dBm. The raw measurements are shown as a scatter plot in Figure 6. There

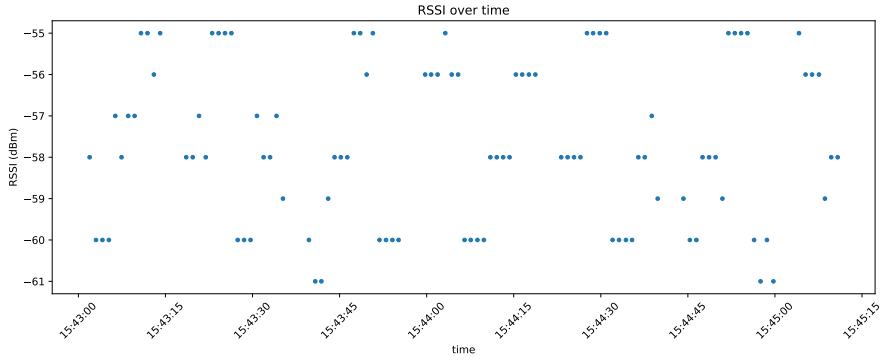


Fig. 6: Scatter plot of the RSSI over time

are three distinct RSSI values that dominate the measurements: -55 dBm, -58 dBm, and -60 dBm. This corresponds to the three different channels on which advertising packets are send. To better illustrate this fact, the probability density function of the RSSI is plotted in Figure 7. In this figure, three distinct peaks are visible at the discrete RSSI levels mentioned before, reaffirming the existence of three distinct RSSI levels related to the different advertising channels.

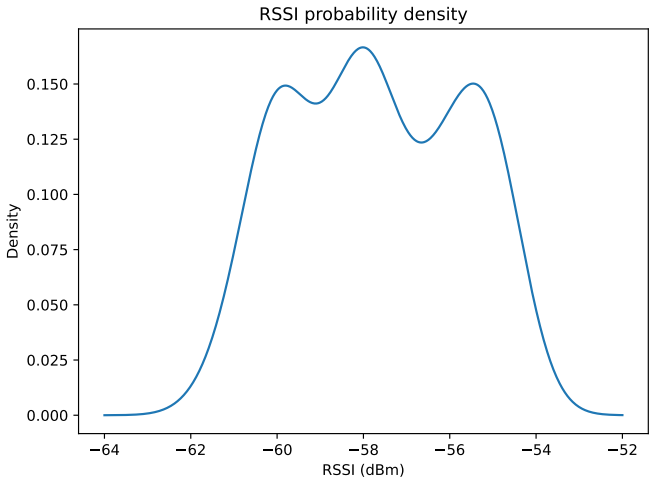


Fig. 7: Probability density function of the RSSI

We repeated the above experiment by measuring the average RSSI at a distance of one meter for all ten beacons. The resulting average RSSI at one meter ranges from -54.0 dBm to -57.7 dBm, with standard deviations between 1.49 dBm and 2.55 dBm.

3.4 RSSI filtering

The next step, after measuring the Received Signal Strength Indicator, is to filter out the variance between the RSSI measurements previously observed. To do so, a window of measurements is implemented, essentially acting as a buffer to store the last n number of measurements, where n is a configurable window size. Each beacon has its own measurements window, as illustrated in Figure 8. Once each BLE packet is received, the radio receiver reports the RSSI value. This value is then added to the measurements window of the corresponding BLE beacon. In the example shown in Figure 8, the measurements window size is five ($n = 5$). When a new measurement is added, and the measurements window is fully filled, the oldest measurement is discarded.

Now, to deal with the variance in the measurements, we implemented three simple methods to find the central tendency of the measurements in the measurement window: the mean (average), the median and the mode. From these methods, the mean is the least suitable filtering method to deal with variance as it can be easily skewed by outliers. An important limitation of the mode is that it is not defined if all measurements occur only once. When this situation occurs, and the mode is selected as RSSI filtering method, the median is used as a fallback method.

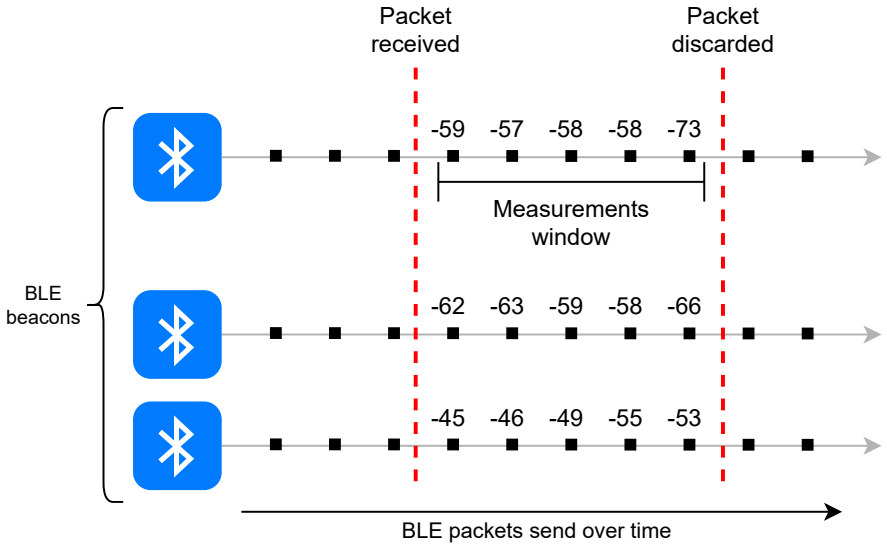


Fig. 8: Measurements windows for multiple BLE beacons

3.4.1 Window size

An important parameter involved in RSSI filtering is the window size (n) — the size/length of the measurements window. A window size of one ($n = 1$) means that no RSSI filtering occurs, and that the variance in the measurements is in no way reduced. On the other hand, changes in the RSSI are directly reflected in the distance estimation. Conversely, if the window size is too large, changes in the RSSI might have a delayed effect on the distance estimation, leading to a position estimation that lags behind on reality.

Furthermore, the appropriate window size is dependent on the movement speed of the positioning subject, as higher velocities may require smaller measurement windows in order to keep up with the changes in position. Also, if the positioning subject is stationary, the measurements window can be expanded to increase the accuracy of the distance estimation and decrease the positioning error.

3.5 Distance estimation

The next step towards indoor positioning is distance estimation. To obtain distance estimates from RSSI measurements, signal propagation models are used [15, 16]. These models rely on the fact that signals incur a loss in signal strength as they propagate through space. This is a consequence of the reduction in power density due to attenuation — a phenomenon called path loss, or path attenuation. The distance estimation step is arguably the most critical one, as accurate distance estimation directly translates to accurate position

estimation. We investigated various distance estimation models, also referred to as signal propagation models. First, the log-distance path loss model is discussed. Then we explore fitting logarithmic models to quantitative data, and finally we look into ways to account for switches between different advertising channels — sometimes referred to as frequency hopping.

3.5.1 The log-distance path loss model

A commonly used path loss model is the so-called log-distance path loss model [17], given in Equation 1,

$$\overline{PL}_{[dB]}(d) = \overline{PL}_{[dB]}(d_0) + 10n \log_{10}\left(\frac{d}{d_0}\right), \tag{1}$$

where $\overline{PL}$ refers to the average path loss, d_0 is a reference distance at which the path loss is known (usually one meter), n is the path loss exponent indicating the rate at which the path loss increases with distance, and d is the distance between the transmitter and receiver.

The path loss can be substituted by the RSSI, and the term $\frac{d}{d_0}$ can be simplified to just d when a reference distance of one meter is used. This results in the following equation:

$$\overline{RSSI} = \overline{RSSI}(d_0) + 10n \log_{10}(d). \tag{2}$$

Finally, rewriting this equation for the distance d gives us:

$$d = 10^{\frac{\overline{RSSI} - \overline{RSSI}(d_0)}{10n}}. \tag{3}$$

The RSSI at a reference distance d_0 of one meter is often included in the signal send by the transmitter.

The log-distance path loss model has two parameters that can be adjusted: the path loss exponent (n) affecting the rate at which the RSSI decreases with distance, and the RSSI at a reference distance d_0.

Typical values for the path loss exponent range from 2.0 to 3.5 [17]. To get a better idea of the effect of changing the path loss exponent, the log-distance path loss model is plotted in Figure 9 for distances between 0.5 and 12 meter, while varying the path loss exponent from 2.0 to 3.5. The reference RSSI at distance $d_0 = 1$ meter is kept constant at a value of -60 dBm. This plot clearly shows that lower path loss exponents lead to flatter curves with an intersect that is also slightly lower. Conversely, higher path loss exponents have a larger intersect, and the RSSI decreases at an increased rate.

The other parameter of interest is the reference RSSI at a distance of one meter, called the transmission power (TX power). To explore the effect of different values of the transmission power, we plot the log-distance path loss model for transmission powers from -70 dBm to -50 dBm in Figure 10. The path loss exponent is kept at a constant value of 2.0. Unsurprisingly, decreasing the transmission power has the effect of translating the model's

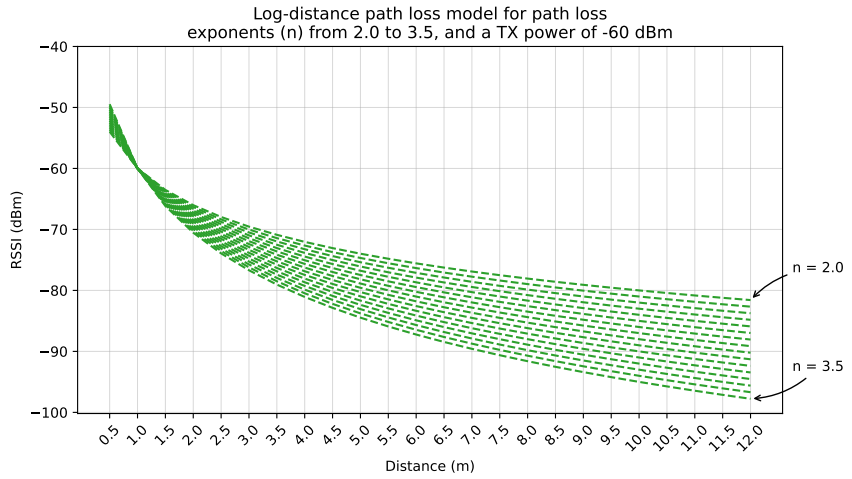


Fig. 9: Log-distance path loss model for path loss exponents (n) from 2.0 to 3.5. The transmission power (TX power) has a constant value of -60 dBm.

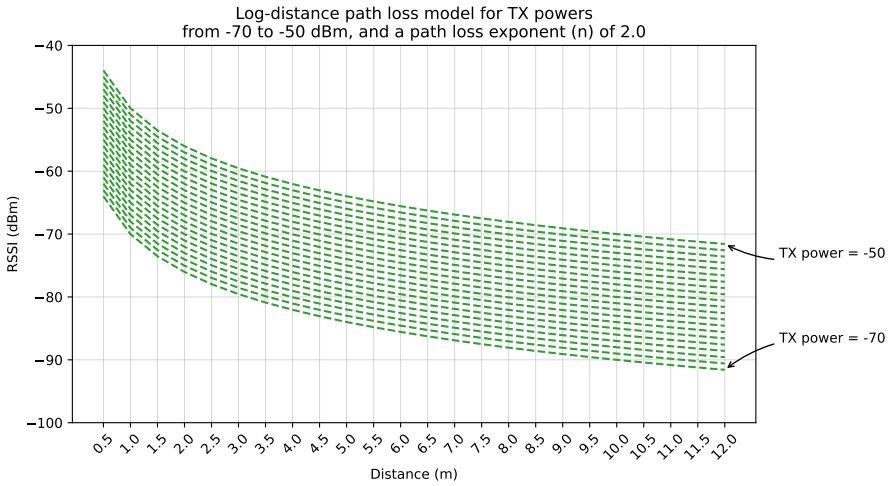


Fig. 10: Log-distance path loss model for transmission powers (TX power) from -70 to -50 dBm. The path loss exponent (n) has a constant value of 2.0.

curve downwards. While the transmission power is included in the packets send by the beacons, we chose to utilize the values obtained by the one meter experiments. The reasoning for this is that all beacons were pre-configured with a TX power value of -59 dBm, which does not account for differences between the individual beacons.

3.5.2 Fitted logarithmic models

Another approach to distance estimation is measuring the Received Signal Strength Indicator for a wide range of distances, and using these measurements to fit the data with a trendline. This trendline can then serve as a distance model specifically tailored to the beacons' characteristics. While such a model does encapsulate characteristics specific to the beacons, it also inadvertently captures qualities of the receiver.

Nonetheless, we measured the RSSI at 24 different distances, from 0.5 meters to 12 meters with increments of 0.5 meters. At each distance we took 100 measurements with Line-of-Sight (LOS) conditions, and 100 measurements with Non-Line-of-Sight (NLOS) conditions simulated by standing in front of the receiver. The results are shown in Figure 11. Each point represents the

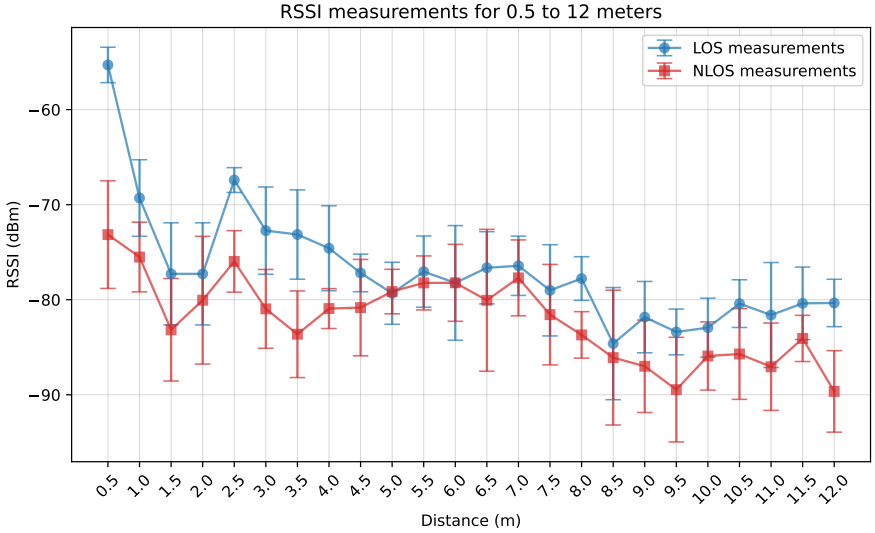


Fig. 11: Average RSSI measurements at distances between 0.5 and 12 meters

average of 100 measurements at the corresponding distance. As expected, the NLOS measurements at each distance lie below their LOS counterpart. Another thing to note is that there does not seem to be a relation between the standard deviations, indicated by the error bars, and the distance between the beacon and the receiver. Furthermore, the RSSI decreases as the distance increases, similar to the log-distance path loss model.

To obtain a distance model from the measurements, we fit the data using a logarithmic least squares method. This involves replacing each distance d with the natural logarithm of the distance $\ln(d)$, and using linear least squares to approximate the line that best fits the measurements. The LOS and NLOS

measurements are fit separately, and the average trendline is obtained by averaging the slope and intercept of both trendlines. The resulting line equations are given in Table 1. The fitted trendlines are plotted in Figure 12, along with

Table 1: Equations of the fitted trendlines

Trendline	Fitted line equation
LOS	$-6.338\ln(d) - 66.765$
NLOS	$-3.851\ln(d) - 75.869$
Average	$-5.094\ln(d) - 71.317$

the RSSI measurements. The error bars are omitted to increase visual clarity. As before, each point represents the average of 100 measurements at the cor-

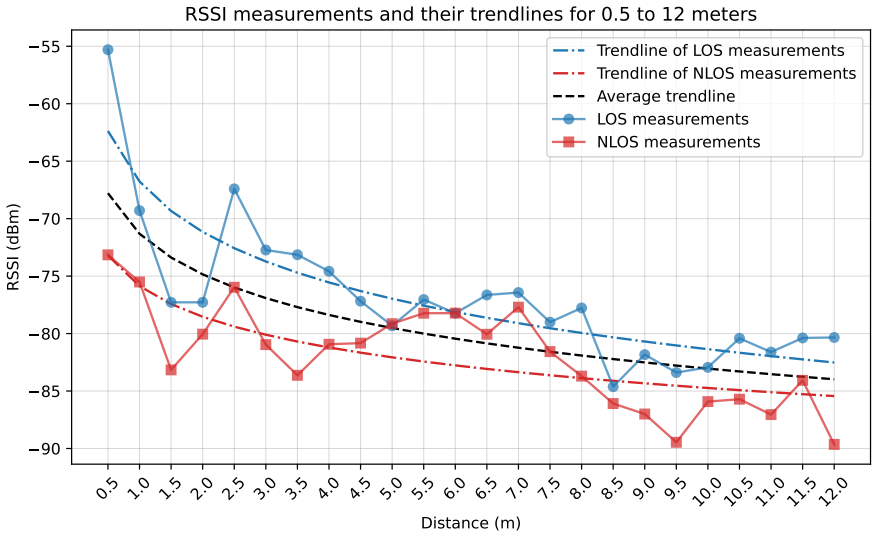


Fig. 12: Average RSSI measurements and their trendlines at distances between 0.5 and 12 meters

responding distance. The dashed lines represent the fitted trendlines for the LOS and NLOS measurements. The average of both trendlines is also shown.

To compare the fitted models with the log-distance path loss model we plot them together in Figure 13. For the log-distance path loss model, a transmission power of -56 dBm is used, which is the average of the RSSI values obtained from the experiments discussed in Section 3.3. For the path loss exponent a value of 2.0 is used. Looking at the plot, the log-distance path loss model yields significantly higher RSSI values than the fitted models.

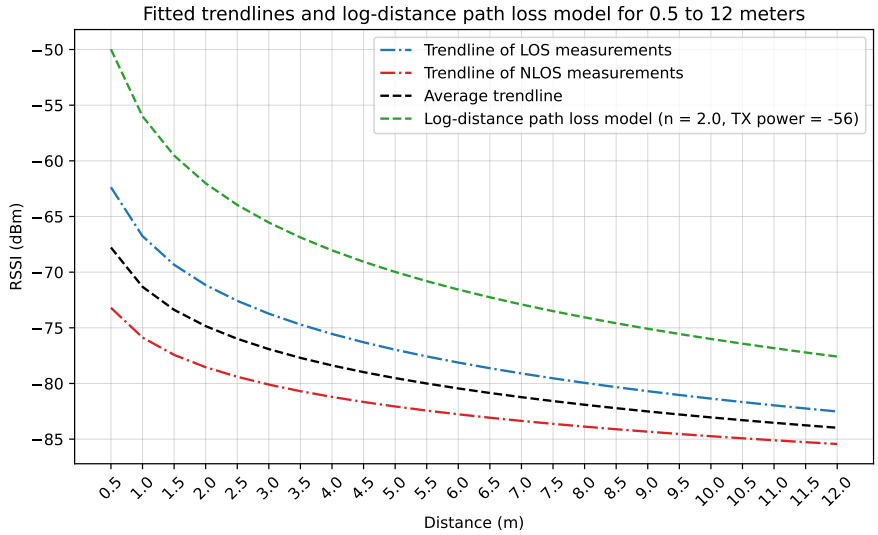


Fig. 13: Fitted model trendlines and the log-distance path loss model

3.5.3 Advertising channel identification

As discussed in Section 3.2, the advertising packets broadcast by the BLE beacons are send on three different channels. While the beacons can transmit on all three channels simultaneously, the radio chip in smartphones can generally only receive packets on a single channel at once. This is a significant problem because the RSSI measurements are affected by an uneven channel gain, resulting in different measurements for different channels [4, 18]. This problem is compounded by the fact that the BLE radio does not relay information about on which channel a beacon was received to the smartphone operating system [18]. As a result, the channel-dependent error can not easily be mitigated.

One simple way to overcome this problem is to configure the beacons to only transmit on a certain channel. Unfortunately, this was not an option for the beacons used in this paper. Alternatively, an algorithm can be used to try and detect on which channel the received BLE was transmitted. To this end, Gentner et al. introduced a novel method to identify the BLE advertising channel on Android devices [18]. This method was implemented in the IPS developed for this paper. Unfortunately, the results of channel identification using measurements at different distances did not show a significant separation between the different channels. Because of this, and the fact that the channel identification is quite rudimentary, we chose to not utilize the channel identification for the distance estimation.

3.6 Positioning

The final step towards locating the smartphone receiver is positioning itself. In this step the estimated distances from the previous step are used by a positioning method to estimate a position.

For the IPS developed for this paper, three positioning methods are implemented: trilateration, Weighted Centroid Localization (WCL), and probability-based positioning. Fingerprinting was also considered, but we wanted positioning methods that are robust to changes in the environment, and do not require a time-consuming offline phase.

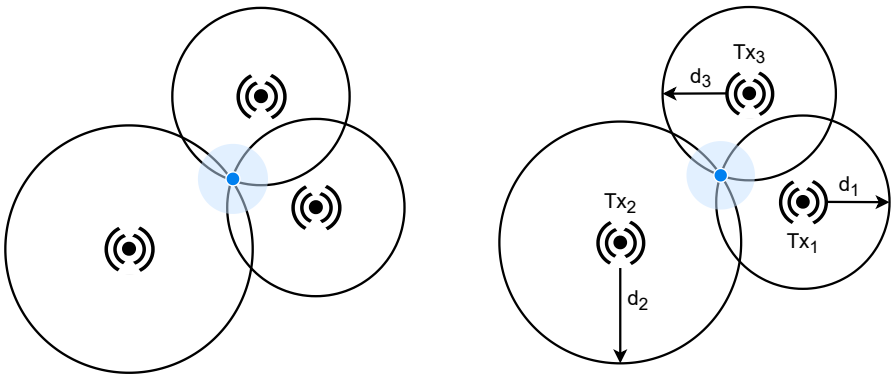
3.6.1 Trilateration

In geometry, trilateration is defined as the process of determining absolute or relative locations of points by measurement of distances [19]. Once the distances to at least three transmitters are known a 2D-position can be calculated. For three dimensions, the distances to at least four transmitters are required. When more than the required amount of transmitters are detected, only the closest ones are used.

For each transmitter, the set of possible positions of the receiver can be determined based on the distance between the transmitter and receiver. In two dimensions, the set of possible positions equates to a circle given by Equation 4,

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}, \tag{4}$$

where d_i is the distance between transmitter i (Tx_i) and the receiver, and (x_i, y_i) is the known reference position of transmitter i. In Figure 14 three transmitters are shown, along with the corresponding circles on which the receiver might be positioned. To find the position of the receiver, the intersection



(a) (b)

Fig. 14: Trilateration

of the three circles has to be calculated [3, 16]. This is done by solving the following system of equations,

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}, \text{ for } i = 1, 2, 3. \tag{5}$$

Again, as the name suggests, trilateration requires three distances in order to find a position estimate. A similar thing is true for the positioning methods discussed in the upcoming sections. Because of this, by default, a position estimate is only provided when three or more beacons are detected. When a position estimate is required while less than three beacons are detected, the following cases occur. If only one beacon is detected, the position is estimated as the location of the detected beacon. If two beacons are detected, the midpoint between the two beacons is calculated, weighed by the distance similar to WCL.

To estimate the distance using trilateration, the intersection of the three circles defined by the distance from each beacon is calculated. Because the distance estimation is imperfect, there are three possible cases that can arise:

Case 1. The three circles have a single intersection point. In this case the intersection point is used as the estimated position. This case is shown in Figure 14.

Case 2. Only two of the circles intersect. In this case there are two intersection points to be considered. For the position estimate, the intersection point closest to the third beacon is used.

Case 3. None of the circles intersect. In this case Weighted Centroid Localization is used.

In reality the situation in case 1 rarely occurs, if ever. Case 2 is the most likely to arise, and case 3 only occurs when the three circles are non-overlapping, or when circles are contained in other circles.

3.6.2 Weighted Centroid Localization (WCL)

The next method is a multilateration method called Weighted Centroid Localization (WCL), first introduced by Blumenthal et al. in 2007 [20]. Because it is a multilateration method, it utilizes all detected transmitters to estimate the target's position. It works by calculating the weighted mean of the known coordinates of nearby transmitters. Transmitters that are closer to the receiver are weighed higher and contribute more to the final predicted position. The position of the weighted centroid is defined by Equation 6 [20, 21].

$$(x, y) = \frac{\sum_{i=1}^{n} (x_i, y_i) \cdot w_i}{\sum_{i=1}^{n} w_i}, \tag{6}$$

where (x, y) is the predicted position, n is the number of considered transmitters, (x_i, y_i) are the coordinates of the i-th transmitter and w_i is the weight allotted to the i-th transmitter.

The weight is inversely proportional to the distance, and is given by Equation 7,

$$w_i = \frac{1}{d_i^g}, \tag{7}$$

where d_i is the distance to the i-th transmitter and g is a parameter that controls the weight drop off at larger distances.

An important limitation of the weighted centroid method is that it only produces reasonably results if the receiver is within the convex hull of the installed transmitters, since no negative weights are considered and thus the predicted position can not go outside of the convex hull [21]. Fortunately, this is the case when the beacons are installed on and around the walls of the building.

In the definition of the weight used in WCL (given by Equation 7), a parameter g is introduced that controls the weight drop off when the distance increases. In Figure 15 the weight is plotted against distances from 0.5 to 12 meters, for weight exponents g from 0.5 to 2.0 with 0.25 increments. As evident

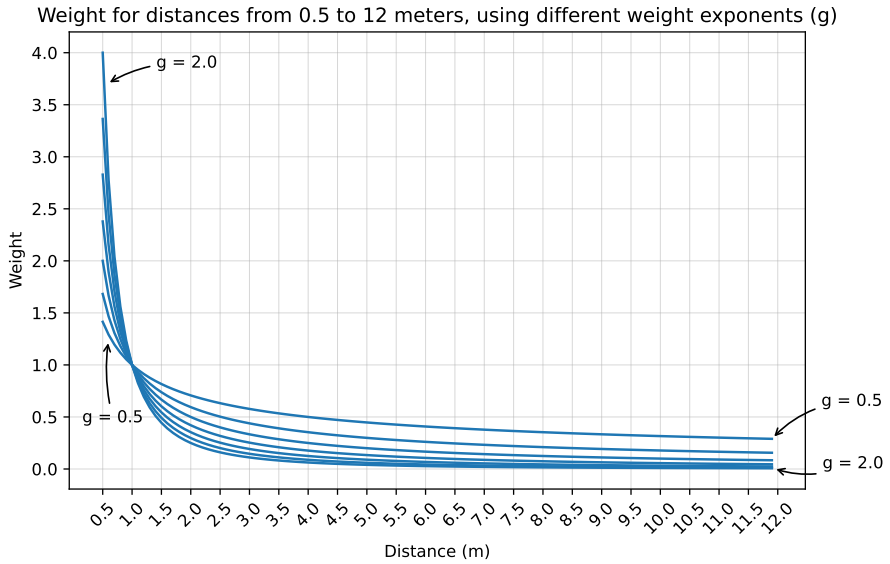


Fig. 15: Weight plotted for distances from 0.5 to 12 meters, for weight exponent (g) values between 0.5 and 2.0, with 0.25 increments

from the plot, increasing the weight exponent g not only increases the rate at which the weight decreases for longer distances, it also results in vastly higher weights for shorter distances.

3.6.3 Probability-based positioning

Finally, we have the probability-based positioning method, introduced by Knauth et al. [22]. This method is an alternative to fingerprinting, but does not require prior fingerprint collection. It only requires the indoor environment to be divided into a grid of points [21, 22]. The probability-based positioning method makes use of a parametric probability density function $p(d, d_i)$. The function describes, for an estimated distance d_i to transmitter i, the probability p for the receiver to be at a distance d from the transmitter's position. A typical probability density function is defined in Equation 8 [21],

$$p(d, d_i) = \frac{1}{(d - d_i)^2 + c}, \tag{8}$$

where c is a parameter influencing the sharpness of the function. The probability is higher if the distance d is closer to the estimated distance d_i.

Equation 8 gives the probability for a single transmitter. To get the probability for all transmitters we multiply the probability of each transmitter to get a residual probability, as given by Equation 9,

$$p((x_j, y_j)) = \prod_{i=1}^{n} p(|(x_i, y_i) - (x_j, y_j)|, d_i), \tag{9}$$

where (x_j, y_j) are the coordinates at which the probability is calculated, n is the number of transmitters, (x_i, y_i) are the coordinates of the i-th transmitter and d_i is the estimated distance to the i-th transmitter.

Now that we can calculate the probability at a certain position using all transmitters we can go to the final step of probability-based positioning. In this step we divide the floor plan into discrete coordinates and then loop over these coordinates. At each coordinate we calculate the residual probability given by Equation 9. After looping over all coordinates the point with the highest residual probability is chosen to be the predicted position.

One drawback of this method is that it can get computationally expensive quite fast, and might not be feasible for very large buildings [21]. This can be solved by approximating the location using another (cheaper) positioning technique or by performing probability-based positioning on a coarser grid. Then the approximate position is used to perform a finer search on an area surround it. Another final thing to note is that the spacing of the grid has to be sufficiently small as to not unnecessarily decrease the accuracy.

3.6.4 Confidence indicator

To enrich the position estimate a confidence indicator is added. The concept is similar to that of dilution of precision (DOP) used by the GPS [23]. The idea of dilution of precision is to state how errors in the measurements will affect the final position estimation. Another source of errors that is relevant for RSSI-based positioning, is the distance estimation. Therefore, the confidence

indicator introduced in this section is comprised of two parts. One focusing on the error within the measurements, and one on the error of the distance estimation. Only the closest three beacons are used to calculate the confidence indicator.

The first part of the confidence indicator is based on the standard deviations of the RSSI measurements, observed in Figure 11. This is a separate component because, as noted before, there is no direct relation between the standard deviation and the distance. To calculate the deviation confidence, first the average standard deviation of the measurements windows corresponding to the closest three beacons is calculated. This calculation is shown in Equation 10,

$$\overline{\sigma} = \frac{1}{3}\sum_{j=1}^{3}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{X}_j)^2}, \tag{10}$$

where $\overline{X}_j$ is the mean of the measurements window for the j-th beacon, x_{ij} is the i-th measurement in the measurements window for beacon j, and n is the length of the measurements window. The average standard deviation is then used to calculate the deviation confidence given by Equation 11,

$$\textbf{deviation confidence} = \mathrm{e}^{-\overline{\sigma}}. \tag{11}$$

Calculating the deviation confidence this way accomplishes two things. Firstly, because of the minus sign the value is scaled to a value between 0 and 1. And secondly, when the average standard deviation increases, the deviation confidence decreases exponentially. This is useful because when the receiver is moving the standard deviation is likely to increase, resulting in a lower deviation confidence, as desired. Conversely, when the receiver is stationary the standard deviation is likely to be lower, resulting in a higher deviation confidence.

The second part of the confidence indicator is based on the RSSI. It is dubbed the "distance confidence" because the distance is directly linked to the RSSI. The distance confidence decreases when the RSSI increases because the likelihood of NLOS-conditions increases. It is calculating by linearly interpolating between 0 and 1, using the minimum RSSI ($RSSI_{\min}$) and the maximum RSSI ($RSSI_{\max}$). The resulting line equation yielding the distance confidence is given by Equation 12,

$$\begin{aligned} a &= (RSSI_{\max} - RSSI_{\min})^{-1}, \\ b &= -(a \cdot RSSI_{\min}), \\ \textbf{distance confidence} &= a \cdot RSSI_{\mathrm{mean}} + b, \end{aligned} \tag{12}$$

where $RSSI_{\mathrm{mean}}$ is the average of the filtered RSSI of the three closest beacons. For the beacons used in this paper, the minimum and maximum RSSI are -100 dBm and -40 dBm respectively.

Finally the confidence indicator is obtained by averaging the deviation confidence and the distance confidence, as shown in Equation 13,

$$\textbf{confidence indicator} = \frac{\text{dev. conf.} + \text{dist. conf.}}{2}. \tag{13}$$

The deviation confidence and distance confidence are abbreviated as dev. conf. and dist. conf. respectively.

4 Experiments

In this section, we discuss the setup and design of the experiments to evaluate our IPS. We will introduce the environment in which the experiments were conducted, along with the reference beacon locations and the ground truth path. Utilizing the ground truth path, we present a method to interpolate between the ground truth coordinates to obtain ground truth estimates for all positions estimated by the IPS. We will also show effective methods to explore the parameters of the system and discuss the error metrics used to evaluate the experiments.

4.1 Experiment setup

The experiments were conducted on the first floor of a residential building, consisting of a living room and a kitchen. A detailed floor plan of the first floor is given in Figure 16. The floor plan includes the furniture and a grid with cells of 20 by 20 centimeters. The total area of the first floor is about 61 m^2 and it has a bounding box of about 12 by 9.6 meters.

4.1.1 Beacon locations

The beacons were spread uniformly along the walls of the living room on the first floor, as shown in Figure 16. Each green dot represents a beacon, and the associated label indicates the beacon's name. The coordinates of each beacon are stored in the MongoDB database, as discussed in Section 3.1.3.

An important thing to note is that every beacon was placed at the same height. This was done because the implemented positioning methods are only capable of providing position estimates in a two dimensional space. By placing all beacons at the same height, positioning happens at the plane that intersects all beacons, which can be directly translated to 2D coordinates. If necessary, all methods discussed in Section 3.6 can be adapted to operate in three dimensions.

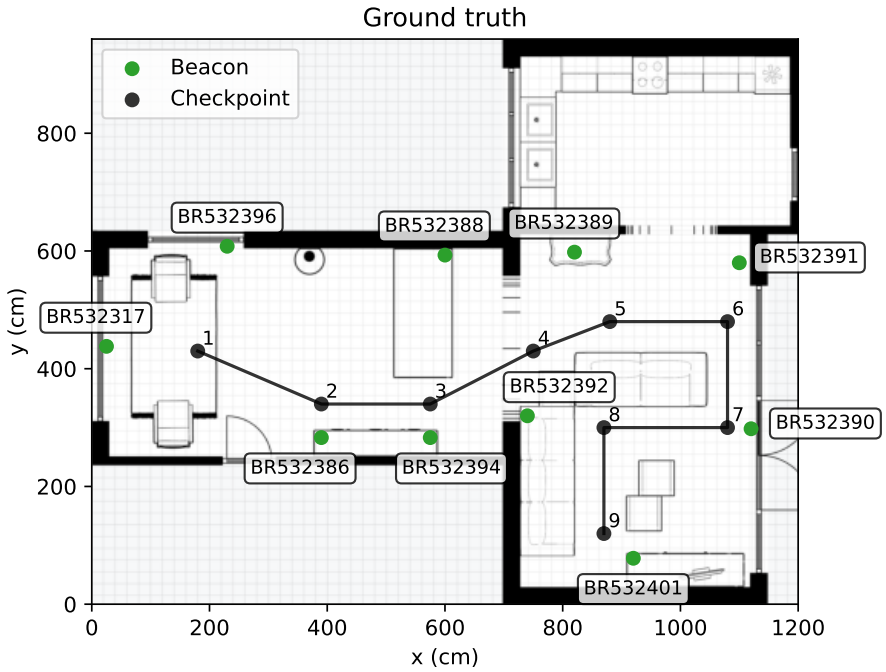


Fig. 16: Floor plan showing the beacon placement and ground truth

4.1.2 Ground truth

The ground truth is given by a path, or so-called trace, that is defined by nine checkpoints. These checkpoints are shown in Figure 16. The checkpoint numbers indicate the direction in which the ground truth path is traversed.

The checkpoints were chosen in such a way that they represent a ground truth trace that covers the complete living room.

To make sure that the ground truth path was followed, the checkpoints were carefully marked on the ground to guide the positioning subject. Furthermore, while traversing the ground truth path, the positioning subject walked in a straight line from checkpoint to checkpoint.

4.1.3 Ground truth interpolation

A problem with evaluating IPSs is that every single position estimate requires a complementary ground truth definition in order to calculate the positioning error. A common way to solve this problem is to use static evaluation [24, 25]. When using static evaluation one or more ground truth points are defined, and the positioning system is used to obtain static position estimates at these reference points, without moving [24]. A drawback of static evaluation is that

it is potentially difficult to capture dynamics that are involved when the positioning subject moves and potentially obstructs lines of sight between beacons and the receiver. Furthermore, sampling at a large amount of ground truth reference points might require significant effort.

To solve the drawbacks of static evaluation, dynamic evaluation methods are used. An example of a such a method is dynamic evaluation using a reference positioning system. In this case, a positioning system with higher accuracy is used as a reference for the evaluation of the target system [24]. The reference positioning system is ideally at least an order of magnitude more accurate than the target system, and consequently, is likely to be more expensive [25]. Unfortunately, such a reference system was not available for this paper.

The method used to evaluate the IPS created for this paper was introduced by Osa et al [24], and falls into the dynamic evaluation category. It uses the predefined geometrical path shown in Figure 16. Furthermore, it requires the positioning subject to indicate when each checkpoint is reached by pressing the checkpoint button shown in the Android application when recording is started. This way, the timestamps corresponding to each checkpoint are recorded. Now, for each position estimate, a corresponding interpolated checkpoint/ground truth point can be generated. This is done by determining between which checkpoints the position estimate falls and, using the timestamps of the position estimate and the checkpoints, linearly interpolating between the previous and upcoming checkpoints. Figure 17 shows an example using linear interpolation. In this example, the first checkpoint is reached at a time of 1 second, and

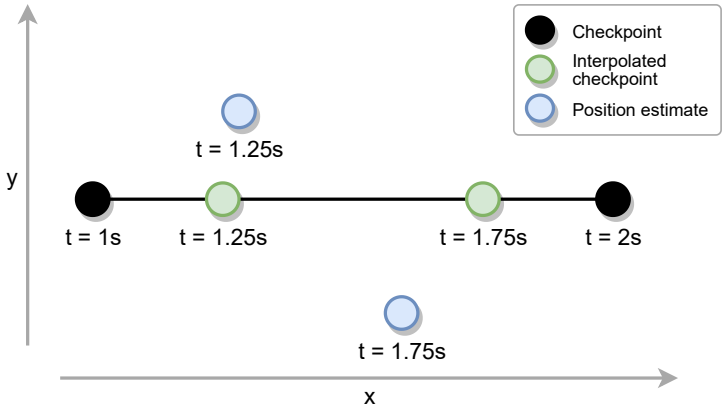


Fig. 17: Ground truth interpolation, adapted from [24]

the second checkpoint is reached at a time of 2 seconds. The checkpoints are represented by black dots. In the time between the times that the checkpoints were reached, two position estimates are provided. These position estimates are represented by blue dots and have timestamps of 1.25 seconds and 1.75

seconds. The corresponding interpolated checkpoints are depicted by the green dots and lay on the between the two checkpoints.

Linearly interpolating between checkpoints only works when the positioning subject moves in a straight line between checkpoints and maintains a constant velocity. As such, the experiments were performed while adhering to these constraints.

4.2 Experiment parameters

The RSSI filtering method, window size, distance model and positioning methods were described in detail in Section 3. Table 2 lists all the different parameters explored in the experiments, along with the considered values. For the window size five different values are considered, capped at 20 measurements to avoid delays in the distance estimation. The values of the weight exponent and probability sharpness are capped at 3.5, since increasing these parameters further would have a diminishing impact on the resulting weights and probabilities.

Table 2: Experiment parameters and the corresponding values

Parameter	Values
RSSI filtering method	Mean, median, mode
Window size	1, 5, 10, 15, 20 measurements
Distance model	Log-distance path loss model, fitted LOS, fitted NLOS, fitted average
Path loss exponent (n)	1.5 – 3.5, with 0.1 increments
Positioning method	Trilateration, Weighted Centroid Localization (WCL), probability-based positioning
Weight exponent (g)	0.5 – 3.5, with 0.5 increments
Probability sharpness (c)	0.5 – 3.5, with 0.5 increments

4.3 Replaying RSSI measurements

The parameters listed in Table 2 all affect the performance of the IPS. However, some parameters are also affected by changes in other, related parameters. For example, changes in the window size affect the filtering methods. This makes it hard to test changes in a parameter in isolation. Furthermore, RSSI measurements differ between experiments making it even more challenging to objectively compare results.

To account for these difficulties, a system was implemented that can replay the RSSI measurements as they were received by the smartphone. This is

done by using the stored RSSI measurements and corresponding timestamps, along with the timestamps recorded for each position estimate. The system calculates the distance and position estimates exactly like the Android application, but it is not bound by time delays between the RSSI measurements, as all measurements are readily available. Consequently, it can apply different positioning techniques with arbitrary parameters to the same data set of collected traces — in a matter of seconds. Using this system, all parameter combinations can be efficiently explored and objectively compared since they are evaluated using the same RSSI measurements.

There are a total of 4680 possible combinations of the parameters given in Table 2. The path loss exponent only has to be considered when the log-distance path loss model is used, and, similarly, the weight exponent and probability sharpness only have to be considered when WCL and probability-based positioning are used. Furthermore, when the window size equals one, no RSSI filtering can be applied.

4.4 Error metrics

When each position estimate in a trace has a corresponding ground truth, the positioning error (also referred to as the localization error) can be determined. To do so, the distance between each estimated position and its ground truth is calculated using the Euclidean distance. Equation 14 defines the Euclidean distance in two dimensions,

$$d(p, g) = \sqrt{(p_x - g_x)^2 + (p_y - g_y)^2}, \tag{14}$$

where $d(p, g)$ is the Euclidean distance between the position estimate p and ground truth point g. Furthermore, the subscripts indicate the dimension, x or y. The calculated distances for each estimated position are aggregated into a single statistical metric that defines the positioning error.

For the experiments in this paper, five different positioning error metrics are considered: the mean error, root-mean-square error, median error, 75^{th} percentile error, and 90^{th} percentile error.

The mean positioning error is perhaps the most commonly used metric to evaluate Indoor Positioning Systems. It is defined as the average of the distance errors, as given by Equation 15.

$$\overline{E} = \frac{1}{n} \sum_{i=1}^{n} e_i, \tag{15}$$

where n is the number of calculated Euclidean distance and e_i is the i-th distance error.

Another metric based on the mean is the root-mean-square error (RMSE). When compared to the mean error, it penalizes larger errors more heavily. It is defined by taking the square root of the mean of the squared error between the distance error and the target distance error. Because the target distance error

is zero, it can be dropped from the equation. The resulting RMSE formula is given by Equation 16,

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}. \tag{16}$$

Metrics utilize the mean are relatively sensitive to outliers. This is not the case for the median error. The median error is the middle value of the sorted list of distance errors, and can also be referred to as the 50^{th} percentile error — the distance error below which 50 percent of the other distance errors fall. The 75^{th} and 90^{th} percentile errors can be used to get an indication of the number of outliers.

5 Results and discussion

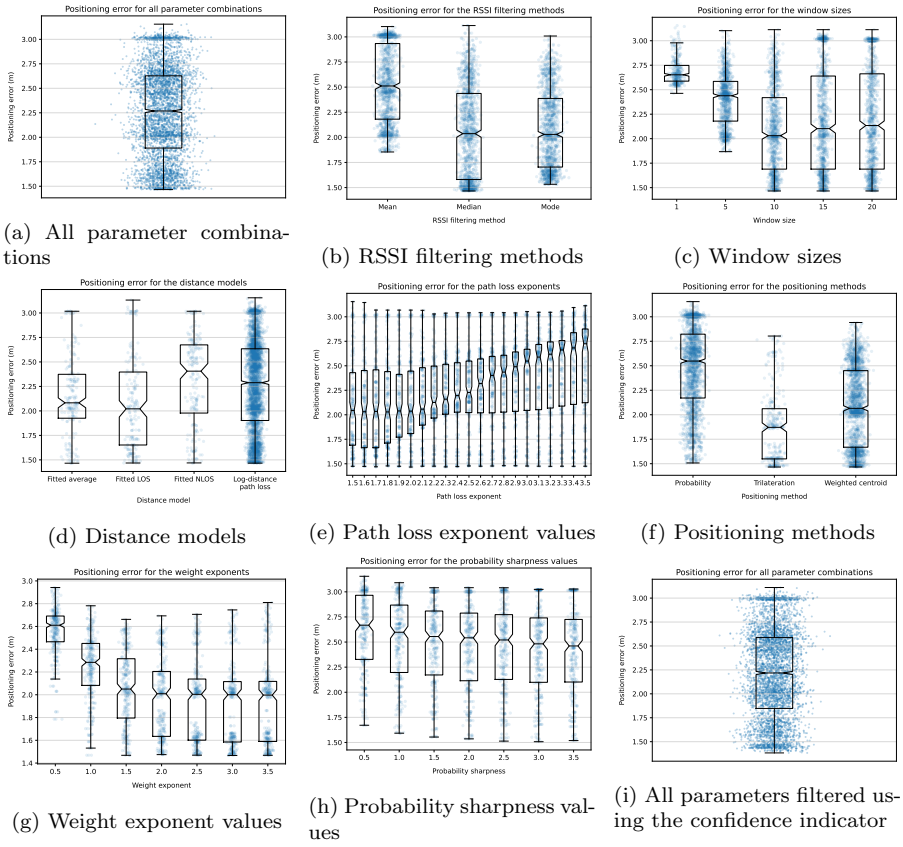
The ground truth path (given in Section 4.1.2) was traversed ten times, resulting in ten different data sets consisting of RSSI measurements and the corresponding positioning data. For each data set, 4680 different sets of processing steps were performed by replaying the RSSI measurements (as discussed in Section 4.3) — resulting in a total of 46800 post-processed traces. Each trace is generated using a unique combination of the parameters listed in Section 4.2, Table 2. For the final positioning error of each parameter combination, the ten positioning errors corresponding to each data set were averaged.

We discuss and briefly summarize the effect of each parameter on the positioning error, evaluating the best performing parameter combination and the overall performance of the IPS. Finally, we also discuss the effect of the confidence indicator.

5.1 Parameter exploration

In order to evaluate the effect of each parameter on the positioning error, the discrete values of the parameters were used to create notched box plots. The notches represent the 95% confidence interval of the median, determined using a Gaussian-based asymptotic approximation [26]. Overlaid on the box plots are scatter plots of the corresponding data points. Each data point represent a unique parameter combination, where the specified parameter value is kept fixed. As such, the data points in the scatter plots are subsets of the 4680 different parameter combinations. To reduce overplotting, and increase legibility, random (normally distributed) noise is added to the data points in the x-direction — in a process called "jittering". Furthermore, the data points are slightly transparent to better visualize larger concentrations of data points.

The positioning errors in Figure 18 are obtained using the median positioning error metric. The other error metrics are omitted, not only for simplicity's sake, but also because using different error metrics when comparing parameter



(a) All parameter combinations

(b) RSSI filtering methods

(c) Window sizes

(d) Distance models

(e) Path loss exponent values

(f) Positioning methods

(g) Weight exponent values

(h) Probability sharpness values

(i) All parameters filtered using the confidence indicator

Fig. 18: Average positioning error for the indicated parameters, data points are jittered to increases legibility

values does not change the relation between the parameter values, and their relative effect on the positioning error.

The complete set of the average positioning error for every possible parameter combination is shown in Figure 18a. Looking at this figure, there is a wide spread of positioning errors for the different parameter combinations. The best parameter combinations result in a median positioning error of about 1.5 meters, while the worst combinations result in errors of above 3 meters. The best combinations are most interesting, and are further explored in Section 5.2. The median error of all parameter combinations is about 2.27 meters (95% CI 2.252 m – 2.285 m). This value is useful to contextualize whether a certain parameter value has a positive or negative effect on the overall positioning error.

5.1.1 RSSI filtering method

The first parameter to be explored is the RSSI filtering method, discussed in detail in Section 3.4. The filtering method is the method used to condense the RSSI measurements in the measurements window into a single value, while trying to deal with the variance between the measurements. There are three different RSSI filtering methods: the mean, median and mode. The results for each method are presented in Figure 18b.

From Figure 18b it is evident that the mean filtering method performs significantly worse than the other two methods, with a median positioning error of about 2.51 meters (95% CI 2.48 m – 2.54 m) and no data points with an error below 1.8 meters.

The median filtering method and the mode filtering method perform equally well; the median positioning error of the median filtering method is about 2.04 meters (2.00 m – 2.07 m), and median error of the mode filtering method is about 2.03 meters (2.00 m – 2.06 m). The similarity in performance between these two methods is partly explained by the design decision to use the median in cases where the mode is undefined, thus resulting in overlapping data. Nonetheless, the best performing parameter combinations using the median RSSI filtering method have a lower positioning error than the filtering method that uses the mode.

5.1.2 Window size

For the window size five different values were considered: 1, 5, 10, 15 and 20. The window size controls the maximum amount of measurements that are contained at once in the measurements window. In Figure 18c, the results for the different window sizes are shown. As stated before, a window size of one equates to no RSSI filtering as the RSSI measurements are used directly in the distance estimation. Consequently, there are only a third of the parameter combinations (data points) for a window size of one when compared to the number of data points for the larger window sizes.

The median positioning error for parameter combinations with a window size of 1 is about 2.65 meters (95% CI 2.64 m – 2.67 m), which is well above the median of the other window sizes. Additionally, all parameter combinations with a window size of 1 have a positioning error above 2.45 meters. The worst parameter combinations — with a positioning error of above 3.1 meters — also all have a window size of 1. These observations indicate that RSSI filtering is an effective way of reducing variances and, by extension, the positioning error.

With a median position error of about 2.03 meters (2.00 m – 2.07 m), a window size of 10 seems to perform the best out of the explored values. Noteworthy however, is that the performance of the window size is related to the travelling speed as discussed in Section 3.4.1. For the experiments a casual walking speed of around 5 km/h was maintained. At this speed the window sizes of 15 and 20 also perform relatively well, as they have a median

positioning error of 2.10 meters (2.06 m – 2.15 m) and 2.13 meters (2.09 m – 2.18 m) respectively.

The similar error ranges for the window sizes of 10 and above are explained by the fact that the median and mode RSSI filtering methods can often result in the same filtered RSSI measurement for the different window sizes.

5.1.3 Distance model

The distance models are used to estimate the distance to a beacon based on the (filtered) RSSI measurement, as discussed in Section 3.5. There are four different distance models used in the IPS, the log-distance path loss model, and the models obtained by fitting distance – RSSI measurements. The results of using these distance models are shown in Figure 18d.

The log-distance path loss model has a lot more data points than the other distance models because of the large range of explored path loss exponent values, which are only relevant when the log-distance path loss model is used.

The optimal distance method depends on the indoor environment; buildings with frequent Line-of-Sight (LOS) obstructions might be better fit to use the log-distance path loss model or the fitted NLOS model, while open-plan buildings are ideal for the fitted LOS model. The fitted average model is a compromise between both situations. For the indoor environment used in our experiments, discussed in Section 4.1, the fitted LOS model performs the best. It has a median positioning error of 2.02 meters (95% CI 1.94 m – 2.10 m), which is considerably lower than the median error of the complete set of parameter combinations. Notable is that it also has the largest range of positioning errors of all the distance models. The fitted average model also performs relatively well with a median positioning error of 2.08 meters (2.03 m – 2.13 m).

The fitted NLOS model has a median positioning error of 2.41 meters (2.33 m – 2.48 m), making it the worst performing distance model. This is in line with expectations as it can be thought of as the counterpart of the LOS distance model. Finally, the log-distance path loss model performs second to worst, with a median positioning error of 2.29 meters (2.27 m – 2.31 m).

Despite the difference in median performances of the different distance models, the best performing parameter combinations for each distance model have a similar positioning error of about 1.5 meters.

Path loss exponent One of the parameters used in the log-distance path loss model is the path loss exponent. The path loss exponent affects the rate at which the RSSI decreases over distances, as explored in Section 3.5.1. The positioning error for values between 1.5 and 3.5, with 0.1 increments, is shown in Figure 18e.

The positioning error seems to increase as the path loss exponent increases. This is consistent with the previous results that showed that the fitted LOS has the lowest positioning error since lower path loss exponents equate to a slower RSSI drop-off, as associated with Line-of-Sight (LOS) conditions [17].

The path loss exponent 1.8 results in the lowest positioning error with a median value of 2.03 meters (95% CI 1.95 m – 2.11 m), while the path loss exponent with a value of 3.5 has the worst median positioning error of 2.73 meters (2.64 m – 2.81 m). There is a difference of about 0.7 meters between the median errors of the lowest and highest explored values of the path loss exponent. The median positioning error obtained using a path loss exponent of 1.8 is comparable to the median error using the fitted LOS and fitted average distance model.

5.1.4 Positioning method

The positioning method is the parameter that has the most direct effect on the positioning error, as it specifies the method that is used to estimate the positioning subject's position. As discussed in Section 3.6, three positioning methods were implemented: probability-based positioning, trilateration and Weighted Centroid Localization (WCL). The positioning error for the parameter combinations of the different positioning methods are shown in Figure 18f.

The trilateration positioning method has fewer data points since no additional parameters were explored, as opposed to the WCL and probability-based method where the weight exponent and the probability sharpness value are considered.

Looking at Figure 18f, trilateration clearly has the best median performance with a positioning error of 1.87 meters (95% CI 1.83 m – 1.92 m). Noteworthy is that there is a clear separation between two groups of data points for the trilateration positioning method. The group with the lower positioning error consists of parameter combinations that have a window size of above 10, where the RSSI filtering is done using the median or mode filtering method. Conversely, the group with a higher positioning error mostly consists of parameter combinations that have window sizes of 5 or 1, and that use the mean RSSI filtering method.

The probability-based positioning method has a median positioning error of 2.55 meters (2.53 m – 2.57 m), making it the worst performing positioning method. Furthermore, none of the parameter combinations that use probability-based positioning achieve a positioning error below 1.5 meters; and all combinations with positioning errors above 3.0 meters can be attributed to the probability-based positioning method.

Finally, when WCL is used, the median positioning error is 2.07 meters (2.04 m – 2.09 m) and the range of positioning errors is similar to that of the trilateration method.

Weight exponent An important parameter in WCL is the weight exponent, as discussed in Section 3.6.2. For WCL, the coordinates of the received beacons are weighed inversely to their distance. The weight exponent controls the extent to which the weight decreases as the distance increases. Higher values for the weight exponent result in steeper weight drop offs. For our experiments, six

different values were explored: 0.5, 1.0, 1.5, 2.0, 2.5, 3.0 and 3.5. The resulting positioning errors are shown in Figure 18g.

Figure 18g shows that the positioning error seems to decrease as the weight exponent increases, indicating that more aggressive weighing of the beacon coordinates, based on the corresponding distances, improves the performance. This is only true to a certain extend and the effect seems to flatten off at weight exponents of 2.0 and above. The median positioning error corresponding to the weight exponent value of 0.5 is 2.61 meters (95% CI 2.59 m – 2.63 m), while the median positioning error for the weight exponent with a value of 2.0 is 2.01 (1.96 m – 2.06 m) meters — rivalling the median error obtained using trilateration.

As the weight exponent increases, the spread of the positioning error seems to increase as well, even though the median error decreases. This can be explained by the fact that for higher weight exponents, the position estimation is dominated by the closest beacons — the beacons with the lowest filtered RSSI measurement. As such, the accuracy of the RSSI measurements corresponding to these beacons has a greater effect on the overall positioning error, increasing the dependence on the effectiveness of the RSSI filtering. This is the same phenomena observed for the trilateration positioning method; for the weight exponent with a values of 2.0 and above there is split between the parameter combinations with windows sizes of 10 or greater, and those with window sizes below 10.

Probability sharpness The final parameter that we explored is the probability sharpness. The probability sharpness is only relevant when probability-based positioning is used. As discussed in Section 3.6.3, it controls the bias towards position estimates that have a distance to the corresponding beacons close to the actual, estimated distances. As the probability sharpness increases, the bias towards the estimated distances decreases. Four different probability sharpness values were explored: 0.5, 1.0, 1.5, 2.0, 2.5, 3.0 and 3.5. The results are shown in Figure 18h.

The median positioning error seems to monotonically decrease as the probability sharpness increases. Furthermore, the boundaries of the error range decrease until the probability sharpness value of 2.5. The decline in the positioning error is not much, and the subsequent median positioning errors fall only just below the 95% confidence interval that corresponds to the previous median positioning error. The median positioning error of the probability sharpness with a value of 0.5 is 2.67 meters (95% CI 2.61 m – 2.72 m). On the other end of the explored value range, the median positioning error is 2.46 meters (2.41 m – 2.52 m) for the probability sharpness value of 3.5.

5.1.5 Summary

To summarize, RSSI filtering is effective as indicated by the increase in positioning errors for lower window sizes. For our experiments the window size of 10 resulted in the best performance, closely followed by the window sizes of 15

and 20. As expected, the median and mode are better suited for RSSI filtering than the mean.

All of the fitted distance models, except the fitted NLOS model, had a lower median positioning error than the median error for the log-distance path loss model. However, the log-distance path loss model performs about equally well as the best performing, fitted LOS positioning method, when the path loss exponent is set to a value of 2.0 to 3.5. Furthermore, out of the implemented positioning methods, trilateration resulted in the lowest median positioning error but the performance is equal to Weighted Centroid Localization (WCL) when a weight exponent of 2.0 to 3.5 is used. Both positioning methods showed a clear divide between parameter combinations with window sizes below 10, and sizes of 10 and above. Finally, probability-based positioning had the worst median performance. While increasing the probability exponent resulted in lower positioning errors, all of the errors were still significantly higher than the errors obtained using the other two positioning methods.

5.2 Best results

The best performing parameter combination is selected based on the average positioning error over all of the error metrics discussed in Section 4.4. There are three best parameter combinations that perform equally well because the performance of these parameter combinations is the same for the window sizes of 10, 15 and 20 due to the median filtering method. The error metrics in meters along with the parameter values are shown in Table 3. The window sizes are shown as a set of multiple values, corresponding to the parameter grouping.

Table 3: Positioning error metrics and parameter values for the best performing parameter combination

Mean	RMS	Median	75^{th} Percentile	90^{th} Percentile
1.59 ± 0.319	1.83 ± 0.408	1.48 ± 0.283	2.10 ± 0.434	2.68 ± 0.882

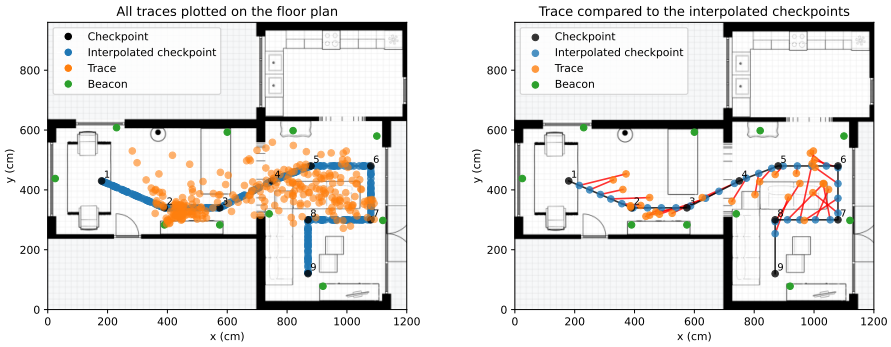
Filtering method	Window size	Distance model	Path loss exponent	Positioning method	Weight exponent	Probability sharpness
Median	{10, 15, 20}	Log-distance path loss	2.4	WCL	3.0	N/A

The median positioning error for the best parameter combination is 1.48 ± 0.283 meters. Regarding the other error metrics, there is a big jump between the median positioning errors and the corresponding 75^{th} percentile errors. A similar sized jump happens between the 75^{th} percentile and 90^{th} percentile. These large increases indicate that there are a significant number of relatively

large positioning errors. Reducing the errors in the upper 25^{th} percentile would greatly reduce the mean and root-mean-square positioning errors.

The values corresponding to the best performing parameter combination are in line with the observations in Section 5.1 that explored the impact of each parameter. The median RSSI filtering method is used, and the window sizes are exclusively above five. Furthermore, the log-distance path loss model is used and positioning is done using WCL.

In Figure 19, the traces corresponding to the best performing parameter combination are plotted. The traces for of all 10 sets of RSSI measurements



(a) All traces corresponding to the best parameter combination

(b) Single trace compared to the interpolated ground truth

Fig. 19: Traces corresponding to the best parameter combination

are shown in Figure 19a. Looking at this figure, the estimated positions at checkpoints 1, 6 and 9 seem to creep towards the "center of mass" of the beacons. This is caused by the nature of WCL; it considers all beacons for the weighted average position resulting in position estimates at the edges of the ground truth path to be skewed more towards the center than position estimates closer to the center of the beacons.

In Figure 19b, one of the 10 traces shown in Figure 19a is plotted. The error for each point of the trace is indicated by a red line between the trace point and the corresponding interpolated checkpoint. Again, the bias towards the center of all beacons is clearly visible.

5.3 Confidence indicator

To evaluate the confidence indicator, introduced in Section 3.6.4, we filtered the points in the 10 base traces based on whether the corresponding confidence indicator was higher than a certain confidence threshold. For the confidence threshold we chose the mean confidence indicator value minus the standard deviation, resulting in a value of 0.256. Figure 20 shows the distribution of the

confidence indicator values as well as the confidence threshold, indicated with a dashed red line. The confidence indicator values seem to follow a normal

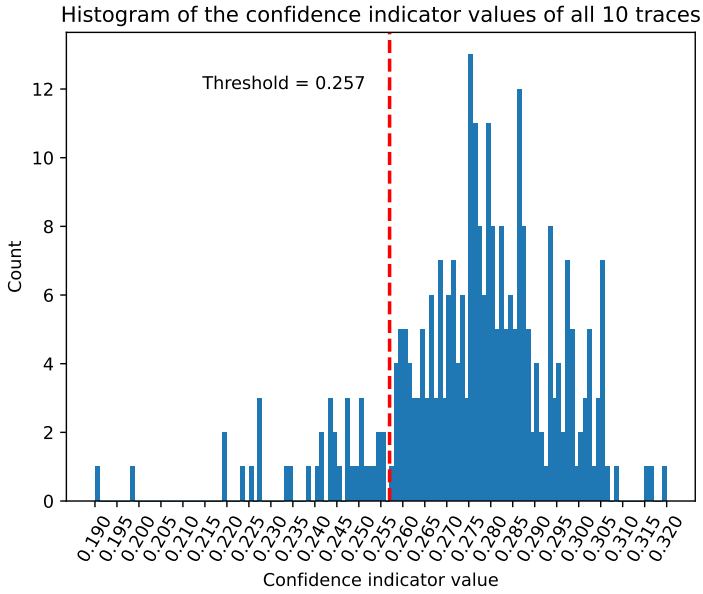


Fig. 20: Histogram of the confidence indicator values

distribution that is skewed to the left, resulting in a left-tailed normal distribution. The left tail is cut off by the confidence threshold. Now, to determine whether the excluded position estimates correspond to high positioning errors, we plot the positioning error for all parameter combinations in Figure 18i.

We can compare the results shown in Figure 18i with the results from Figure 18a. The median positioning error of the parameter combinations using the filtered traces is 2.22 meters (95% CI 2.201 m – 2.235 m), five centimeter less than the non-filtered results — a small, but statistically significant difference.

To get a better idea of which position estimates are filtered, the best parameter combination using probability-based positioning is plotted with the filtered trace points crossed out. The results are shown in Figure 21. The probability-based positioning parameter combination is chosen because of the high 90^{th} percentile positioning error. Qualitatively, it seems that filtering based on the confidence indicator is successful in removing most position estimates that lie far away from the ground truth trace. However, not all outliers are eliminated and some position estimates with a relatively low positioning error are removed.

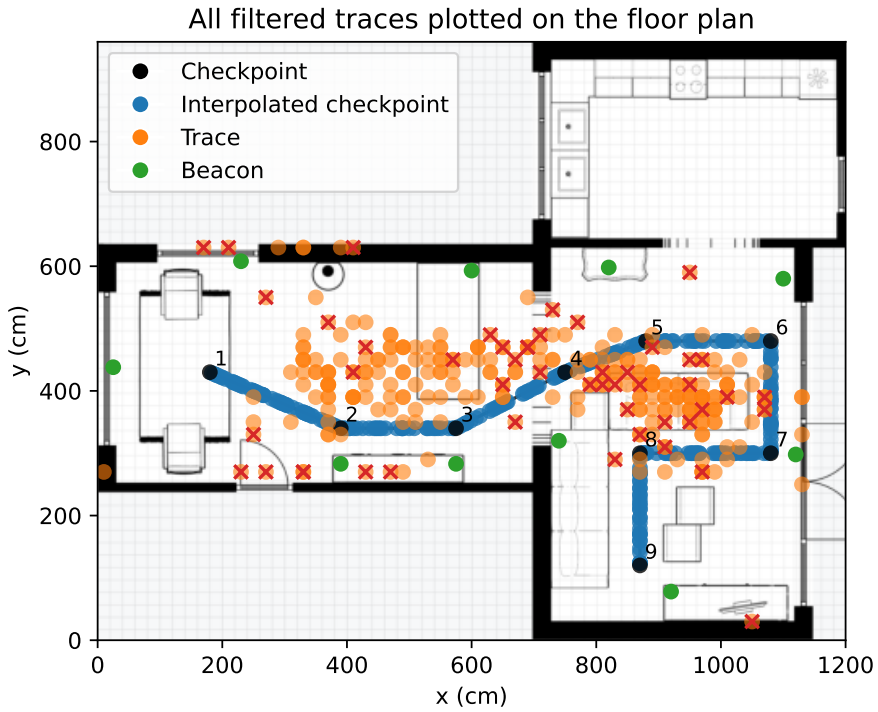


Fig. 21: All traces corresponding to the best probability-based parameter combination, using traces that are filtered based on the confidence indicator

6 Conclusions

In this paper, an Indoor Positioning System (IPS) is implemented that runs locally on a smartphone and requires minimal setup before it is able to operate. At the core of this IPS are four main steps. First, the RSSI measurements are collected. These measurements are then filtered by taking the mean, median or mode of a measurements window that holds the last n measurements, where n is a configurable window size parameter. Next, the RSSI measurements, obtained from the filtering step, are used to estimate the distances between the corresponding beacons and the smartphone receiver. To this end four distance estimation models were explored, the well-known log-distance path loss model and three models based on fitting a trendline to data obtained by measuring the RSSI at distances from 0.5 to 12 meters (with 0.5 meter increments). The fitted trendlines include a trendline for data obtained in Line-of-Sight (LOS) conditions, a trendline for data obtained in Non-Line-of-Sight (NLOS) conditions, and a trendline that averages the two. Finally, the distance estimates from the distance estimation models are used as input to a positioning method in order to estimate the position of the smartphone receiver. Three positioning

models were explored and implemented: trilateration, Weighted Centroid Localization (WCL), and probability-based positioning. Fingerprinting was not considered because of the extensive setup required. Furthermore, a confidence indicator is proposed that serves as a metric to indicate the reliability of a position estimation. This confidence indicator is computed based on standard deviations in the measurements window and the RSSI values for each detected beacon.

To evaluate the IPS, several experiments were performed in which the position of the positioning subject (carrying the smartphone) was estimated periodically, while traversing a predefined path. Because every position estimate requires a complementary ground truth to calculate the positioning error, additional ground truth points were generated. This was done by interpolating between checkpoints with known timestamps using the timestamps of each received RSSI measurement. The positioning error is obtained by first calculating the Euclidean distance between the ground truth points and the estimated positions, and then using the Euclidean distance to compute five common positioning error metrics: the mean error, root-mean-square-error, median error, 75^{th} error, and 90^{th} percentile error.

The IPS has seven different parameter that affect the performance of the system. Because the effects of these parameters are hard to study in isolation, and the positioning error can not easily be directly compared across different experiments (without an excessive number of experiments), a system was implemented to replay the RSSI measurements. This system behaves identically to the IPS, and was used to exhaustively explore the combinations of the values for the different parameters.

For the conducted experiments, the results showed that a window size of 10 significantly decreased the positioning error compared to smaller measurement windows. Furthermore, the median RSSI filtering method was most effective in filtering the variance between RSSI measurements. The best performing distance estimation models for our experiment environment were the fitted LOS model, and the log-distance path loss model using a path loss exponent between 1.5 and 2.0. Finally, the positioning methods with the lowest median positioning error were trilateration and WCL with a weight exponent between 2.0 to 3.5. The best performing parameter combinations had a median positioning error of about 1.48 ± 0.283 meters, and used the median filtering method, window sizes between 10 and 20, the log-distance path loss model, and Weighted Centroid Localization with a weight exponent between 2.0 and 3.5. Finally, filtering out position estimates using thresholding of the confidence indicator yielded a slight reduction in the positioning errors.

With a median positioning error of 1.48 ± 0.283 meters, the positioning error achieved in this paper is lower than the majority of positioning errors presented in the related work discussed in Section 2. Furthermore, works that achieved a lower positioning error were either evaluated in an environment with a small area, or used fingerprinting and required an extensive calibration phase. That being said, the test environment used in this paper was comparatively small, and the beacons were relatively densely deployed. Nevertheless, the test

environment was reasonably complex, and the deployed beacons only had a broadcast frequency of 2 Hz.

While the results presented in Section 5 are promising, with the best performing parameter combinations yielding errors of around 1.5 meters, there is still a lot of room for improvements. First off, the Bluetooth 5.1 specification revealed by the Bluetooth Special Interest Group (SIG) in January 2019 supports Angle of Arrival (AOA) measurements enabling triangulation techniques that could lead to large accuracy improvements. As of writing however, support for Bluetooth 5.1 and above is still very limited, even in the newest smartphones [27].

Another improvement that could be made is using BLE beacons that are capable of broadcasting BLE advertising packets at a higher frequency than 2 hertz (once every 500 ms). This would enable larger window sizes and more aggressive filtering. It would also be helpful in exploring the effects of different travelling speeds. In this paper, only static window sizes were considered due to inaccuracies in potential velocity estimation using the on-board accelerometer. As an alternative, future work could look into heuristics that could be used to dynamically adjust the window size. A possible heuristic is using step length estimation and step detection to approximate the velocity.

Finally, an interesting avenue of research that was not explored in this paper, is optimal BLE beacon placement and the effects of varying the number of deployed beacons. Furthermore, only a single ground truth path was traversed due to time and space constraints. Evaluating the IPS using multiple ground truth paths and multiple indoor environments would result in more rigorous experiments and results.

Declarations

Conflict of interest. The authors declare no competing interests.

Availability of code and data. In the interest of open science, both code and data are made publicly available[4]. Any use must include a citation to this paper.

References

1. IndustryARC. Indoor Positioning and Navigation Market - Forecast (2020 - 2025). `https://www.researchandmarkets.com/reports/4531980/indoor-positioning-and-navigation-market`, 2020.
2. Germán Martín Mendoza-Silva, Joaquín Torres-Sospedra, and Joaquín Huerta. A meta-review of indoor positioning systems. *Sensors*, 19(20), 2019.

[4] `https://github.com/rriesebos/indoor-positioning-system`

3. Wilson Sakpere, Michael Adeyeye-Oshin, and Nhlanhla B.W. Mlitwa. A state-of-the-art survey of indoor positioning and navigation systems and technologies. *South African Computer Journal*, 29:145 – 197, 00 2017.
4. Ramsey Faragher and Robert Harle. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.
5. Pavel Kriz, Filip Maly, and Tomas Kozel. Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems*, 2016, 2016.
6. Yuan Zhuang, Jun Yang, You Li, Longning Qi, and Naser El-Sheimy. Smartphone-based indoor localization with bluetooth low energy beacons. *Sensors*, 16(5):596, 2016.
7. Santosh Subedi, Goo-Rak Kwon, Seokjoo Shin, Suk-seung Hwang, and Jae-Young Pyun. Beacon based indoor positioning system using weighted centroid localization approach. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 1016–1019. IEEE, 2016.
8. Sebastian Sadowski and Petros Spachos. Rssi-based indoor localization with the internet of things. *IEEE Access*, 6:30149–30161, 2018.
9. Baichuan Huang, Jingbin Liu, Wei Sun, and Fan Yang. A robust indoor positioning method based on bluetooth low energy with separate channel information. *Sensors*, 19(16):3487, 2019.
10. Liu Liu, Bofeng Li, Ling Yang, and Tianxia Liu. Real-time indoor positioning approach using ibeacons and smartphone sensors. *Applied Sciences*, 10(6):2003, 2020.
11. Jacopo Tosi, Fabrizio Taffoni, Marco Santacatterina, Roberto Sannino, and Domenico Formica. Performance evaluation of bluetooth low energy: A systematic review. *Sensors*, 17:2898, 12 2017.
12. Apple. iBeacon - Apple Developer. `https://developer.apple.com/ibeacon/`, 2021.
13. Core Specification Working Group. Bluetooth Core Specification. `https://www.bluetooth.com/specifications/specs/core-specification/`, 12 2019.
14. OnePlus. OnePlus 6 specs. `https://www.oneplus.com/nl/6/specs/`, 2021.
15. Mai Al-Ammar, Suheer Alhadhrami, Abdulmalik Al-Salman, Abdulrahman Alarifi, Hend Al-Khalifa, Ahmad Alnafessah, and Mansour Alsaleh. Comparative survey of indoor positioning technologies, techniques, and algorithms. In *2014 International Conference on Cyberworlds*, pages 245–252, 10 2014.
16. Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3):2568–2599, 2019.
17. Theodore S Rappaport et al. *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey, 1996.

18. Christian Gentner, Daniel Günther, and Philipp H Kindt. Identifying the ble advertising channel for reliable distance estimation on smartphones. *arXiv preprint arXiv:2006.09099*, 2020.
19. The Editors of Encyclopædia Britannica. Trilateration. In *Encyclopædia Britannica*. Encyclopædia Britannica, Inc., 2016.
20. Jan Blumenthal, Ralf Grossmann, Frank Golatowski, and Dirk Timmermann. Weighted centroid localization in zigbee-based sensor networks. In *2007 IEEE International Symposium on Intelligent Signal Processing*, 2007.
21. Stefan Knauth. Study and evaluation of selected RSSI-based positioning algorithms. In *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, pages 147–167. Elsevier, 2019.
22. Stefan Knauth, Alfonso A Badillo Ortega, Habiburrahman Dastageeri, Tommy Griese, and Yentran Tran. Towards smart watch position estimation employing rssi based probability maps. In *Proceedings of the First BW-CAR Baden-Württemberg CAR Symposium on Information and Communication Systems (SInCom 2014), Furtwangen, Germany*, page 75, 2014.
23. Richard B Langley et al. Dilution of precision. *GPS world*, 10(5):52–59, 1999.
24. Carlos Martínez de la Osa, Grigorios G Anagnostopoulos, Mauricio Togneri, Michel Deriaz, and Dimitri Konstantas. Positioning evaluation and ground truth definition for real life use cases. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, 2016.
25. Daniel Becker, Fabian Thiele, Oliver Sawade, and Ilja Radusch. Cost-effective camera based ground truth for indoor localization. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 885–890. IEEE, 2015.
26. Robert McGill, John W Tukey, and Wayne A Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.
27. GSM Score. Bluetooth 5.1+ support. `https://www.gsmscore.com/model-finder/filter/`.