Optimal Receivers Location for Passive Indoor Positioning Based on BLE

Roald de Brouwer[1,2], Joaquín Torres-Sospedra[3], Sergio Trilles Oliver[2] and Rafael Berkvens[1]

[1] *University of Antwerp - imec, IDLab - Faculty of Applied Engineering, Antwerp, Belgium*
[2] *Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain*
[3] *UBIK Geospatial Solutions S.L., Castellón, Spain*

Abstract

Indoor Positioning Systems are attracting many researchers, being Bluetooth Low Energy gaining some attention recently. The recent need to track people, ensuring low density of people in some areas, made the research community to provide unintrusive tracking. In contrast to traditional direct fingerprinting, where the user utilizes their mobile to estimate its location based on the BLE infrastructure, we propose the other way around. The users *wear* a beacon which is sensed by the monitors deployed and inverse fingerprints are generated for a particular object of interest. We propose in this paper a simulation-based approach to retrieve the most interesting monitor distribution, providing a real-world validation. The results show "passive" positioning errors of 3 meters in an office sensed with 5 monitors.

Keywords

BLE, Tracking, Passive fingerprinting

1. Introduction

The proliferation of smartphones and other wireless devices increases every year. This growth process is expected to have 4.3 billion smartphone users by 2023 [1]. This results in a wide range of services including indoor localization which is a process of obtaining a device or user location indoors.

In the last decades, indoor localization has mainly been investigated in industrial sectors, wireless sensor networks, and robotics. The increasing communication capabilities on the smartphones and wearables results in localization possibilities. Hence, the user can also be localized and many different services and applications can be developed. These services apply in the health sector [2], disaster management [3], surveillance [4], and many other sectors. The Internet of Things (IoT) could also benefit from indoor localization services such as smart buildings [5], and Machine Type Communication (MTC) [6].

IPIN 2021 WiP Proceedings, November 29 – December 2, 2021, Lloret de Mar, Spain

Roald.deBrouwer@student.uantwerpen.be (R. de Brouwer); info@jtorr.es (J. Torres-Sospedra); strilles@uji.es (S. Trilles Oliver); rafael.berkvens@uantwerpen.be (R. Berkvens)
http://jtorr.es (J. Torres-Sospedra); http://www3.uji.es/~strilles/ (S. Trilles Oliver)
0000-0003-4338-4334 (J. Torres-Sospedra); 0000-0002-9304-0719 (S. Trilles Oliver); 0000-0003-0064-5020 (R. Berkvens)

The focus in this paper will be on a monitor based localization system. The process in which a set of anchor nodes or reference nodes passively obtains the position of the user or entity connected to the reference node. Monitor based localization is primarily used for tracking the user and then accordingly providing different services [7]. The monitors in the environment will sense the Bluetooth Low Energy (BLE) signals in order to estimate their source location. In contrast to active localization performed on smartphones, the users will carry a BLE beacon whose battery may last months or even years. In this way, sensing relies on the infrastructure deployed and the calculations to estimate the position can be shifted to an external server which does not depend on a battery. Whereas traditional device-based active localization systems collect BLE signals from the reference nodes and perform the location calculation (either on the device or using wireless communications to send to an external server). These additional workload (extra calculations for computing the position estimate or communications to an external server) will consume more battery and lead to a negative user experience, when using a smartphone as the positioning device [8].

For BLE-based active localization, several beacons can be deployed in the environment to support, for instance, a smartphone to get its location. As the cost of a beacon is low compared to other positioning technologies, a dense deployment is feasible to ensure good positioning accuracy. In addition, beacons are usually battery-powered, that may last several months, thus requiring low maintenance. In contrast, a monitor-based approach is more complex as the monitors deployed are more expensive and require a wired power supply and internet connectivity (Ethernet is more robust than wireless communications). Installing a monitor is not as easy as just sticking a BLE beacon in the ceiling and, therefore, a good analysis to locate the monitors is needed to maximize the performance of the Indoor Positioning System (IPS).

Choosing the best possible reference node setup in a room is essential for a good monitor-based positioning system. Therefore, in this paper, a simulator is built to mimic the radio propagation in an indoor office. The simulator is configured to evaluate one hundred random reference node setups and 8-12 self-chosen logical setups for the indoor office. Furthermore, the simulation is also configured to evaluate scenarios where the number of BLE monitors goes from one to ten. The results of the simulation are evaluated to choose the best configurations for each scenario (from 1 to 10 monitors).

In a next step, the best setups for 3 to 5 monitors (reference nodes) are unrolled in the office to assess their real-world performance. Thereafter, the results of both the simulation and the real-world setups are compared. With this comparison, we want to demonstrate if a simulation can be a good first approach to determine what a good reference node setup is.

The paper is presented as follows. Section 2 gives an outline of the existing systems. In Section 3 there is a more in-dept explanation about the used materials and methods, followed by section Section 4 in which the results of the simulator and the physical setups is shown. Section 5 consists of a discussion of these results, after which a general conclusion about the performed work is presented in section Section 6.

2. Related work

Techniques such as Received Signal Strength (RSS), time of arrival, angle of arrival, or combinations of these techniques are commonly used in wireless indoor localization systems. The consolidation of these techniques with technologies (i.e., infrared, ultrasound, Radio Frequency (RF) waves, vision, etc.) are used to obtain a position. These techniques and technologies are described by Zafari et al. [7] together with their advantages and weaknesses.

Indoor localization systems entirely based on RF waves are commonly used for the reason that the accuracy can be relatively high depending on the available resources and used technique. RF chips have an on-board RSS block that can determine the received power of the incoming signal. These RSS values can be transformed to an estimated distance between two communicating devices causing these positioning systems to be relatively cheap in comparison with other systems which use a different technology. Different conversions from signal strength to distance can be used but all these conversion algorithms are influenced by propagation effects (fading, scattering, absorption, reflection, etc.), resulting in an unreliable conversion from signal strength to distance. Additionally, the battery restrictions imposed by some autonomous devices (as smartphones) make them a perfect match for integration in, for instance, smartphone-based applications.

By using the fingerprinting technique and generating radio maps the propagation effects can be partly circumvented. That approach was introduced in 2000 [9] and is still widely adopted by the indoor research community. In fast changing environments fingerprinting may become more complicated because the generated radio maps would need to be updated for the unstable environment. However, the other way around, where the sensors deployed in the environment collect the RSS values emitted by the mobile devices is less frequent.

Lee et al. [10] proposed an indoor location tracking service, which was based on BLE and Raspberry Pi Scanners. The authors introduced a nursing hospital for patient monitoring as the main evaluation area with several BLE scanners, one per room and four in a main large common area. However, they did not clarify the procedure to obtain its optimal placement for that use-case environment.

Schneider et al. [11] proposed a low-cost indoor tracking system able to detect, collect, process, and store user movement and presence data in indoor environments. For that purpose, it was supposed that the users were carrying a beacon/tag or a smartphone acting as a "smart beacon" and several low-cost off-the-shelf ESP32 units were deployed at specific points of interest to infer the user's location by proximity. However, that approach was prone to error because of the RSSI fluctuation. The authors proposed additional processing layers and simplified the procedure to estimate the position, they registered delimited places (e.g., offices) avoiding halls and corridors and other transitory locations and they applied a time-window of 30 s to smooth the RSSI values. Again, no indications about where to deploy the scanners were provided.

Kolakowski et al. [12] proposed a tracking system based on sensing nodes equipped with an ARM Cortex M4F microcontroller, barometer, Ultra-wideband (UWB), Wi-Fi, and two BLE modules. The Tags were able to send UWB and BLE messages to allow their tracking. The system accuracy and precision were evaluated in a furnished apartment for static object localization and moving person (dynamic) localization. Authors provide a map with the location of the anchor nodes, without justifying their location.

Rajkumar and Deshpande [13] rbased their tracking system on PSoC BLE nodes, which allowed two arrangements. Despite these nodes being able to receive the signals from the mobile target, they were used as mere transmitters (BLE Beacons), using the smartphones as a mobile receiver.

Belka et al. [14] proposed a tracking system based on BLE to acquire relevant information about the behavior of visitors in touristic locations. For such kind of applications, the positioning accuracy has low relevance as the spatial size of the point of interest is a few or tens of meters. The authors proposed beacons with either the CSR101x or CSR102x series chipset depending on the BLE version. Several analyses were shown, showing the problem of overlapping RSSI values in proximity with close points of interest. Again, the location of the sensory network based on Raspberry Pi was not considered.

Most of the works that we have analyzed on monitor-based passive localization are either based on proximity to the anchors or in reducing the variability of the RSSI values, being fingerprint-based solutions usually not considered. Given the success of fingerprint-based solutions in “active” localization using BLE beacons as anchors and smartphones as monitors, we will study in this paper the other way around. In contrast to active localization where one may deploy a denser set of anchors (BLE beacons), in a monitor-based solution the density of anchors is expected to be lower as the monitor infrastructure is expected to be more expensive. However, most of the reviewed works do not present a study on where to place those anchors to maximize accuracy.

There are several techniques for determining the optimal receiver locations in RF positioning. Cramér-Rao Lower Bound (CRLB) calculates the lowest possible variance of an unbiased estimator. It has been adopted numerous times for location estimation, and then optimized with changing node locations, e.g., [15, 16]. The Dilution Of Precision (DOP), also referred to as Horizontal DOP (HDOP) or Geometric DOP (GDOP) in terrestrial localization, is a metric from satellite navigation systems that relates the satellite constellation to the expected precision of the location estimate. The satellite constellation can be regarded as the anchor or receiver locations. In ultrasound networks, [17] used the estimated DOP to optimize the receiver locations. Finally, [18] used the mutual information in a UWB localization setup to improve receiver locations.

These works differ from our approach in the sense that they compute an additional metric (CRLB, DOP, or mutual information) to estimate the quality of the IPS system, while our approach directly simulates the expected localization error in an environment to determine the optimal receiver locations.

3. Material and Methods

This section is devoted to introducing the materials and methods used to carry out the experimental work reported in this paper. We first propose the positioning engine for “inverse” fingerprinting, then we describe the developed simulator to retrieve the best distribution of monitors, and finally, we show the hardware used in the experiments.

3.1. Inverse fingerprinting model

The traditional fingerprint approach work as follows. A user with a collection device gathers fingerprints at well known locations (also known as reference points). At every reference point, 1 or more fingerprint vectors are collected and stored with information about its location. Each fingerprint vector is composed of the RSS values from the detected Access Points (in Wi-Fi) or BLE Beacons. The set of fingerprints collected forms the radio map. For positioning, a new operational fingerprint is collected and then is compared to all the fingerprints in the radio map, the location information of the most similar ones is used to estimate the position of the operational fingerprint.

The concept of fingerprint and radio map vary with the proposed tracking solution. Now, the fingerprint does not characterize the RSS values collected by a device to obtain its position, but the RSS values collected by multiple monitors for a particular beacon for a short period of time. For calibration, a known beacon will be placed in well known locations and at well-known periods of time. i.e., if the calibration beacon was placed in x,y from 10:00 to 10:05, we can generate multiple reference fingerprints for that calibration beacon using the RSS registered by the monitors. This approach requires a central element (server) to join the information from all the monitors and generate the resulting fingerprints, which will have as many elements as monitors are deployed in the environment. If the monitor does not detect a particular beacon, a default low value is used in the fingerprint vector. For the operational stage, the monitors are tracking all the BLE advertisements which are sent to the central element to generate the operational fingerprint. As the monitors are not providing synchronized RSS readings, a short time window is needed to generate the latest operational fingerprints, one per recently detected beacon.

In both cases, active and inverse fingerprinting, the k-NN algorithm is the approach providing the most direct implementation. The input for this algorithm consists of reference points with their RSSI value (i.e., the radio map) and the operational fingerprints (the RSSI vectors). Based on a chosen amount of neighbors the algorithm is going to compare the RSSI values of the operational fingerprints with those of the reference points, resulting in an estimated position.

For evaluating indoor positioning systems based on fingerprinting, the average positioning error (the Euclidean distance between the estimated position and the exact location they were taken) is used. Additionally, we also consider the standard deviation of the positioning error. Not only the error but also the variability are considered to compare the setups.

3.2. Simulator tool

A synthetic data generator tool (simulator) was developed to model the signal propagation between any BLE beacon present in the environment and all the monitors deployed for monitoring purposes. The tool allows users to introduce the location of a predefined number of monitors in the environment using a map and, later, generate the training "inverse" fingerprints in a regular grid distribution. For the simulator, we have followed the same approach as for regular fingerprinting, but the anchors correspond now to the monitors and the moving devices to the BLE beacons (emitters). For simplicity, we can assume a perfect synchronization among all the monitors.

In particular, we have simulated the main office of the Geospatial Technologies Research Group at Universitat Jaume I, which is around 16 m by 12 m. A grid of 1 m was implemented, having a total of 183 reference points. For evaluation, we choose a random point falling within the limits of each of the 153 cells in the grid set-up.

The simulator discretizes the office as a matrix. Every cell in the matrix has a different number that represents free space or objects such as a table, bookshelves, or pillars. i.e., the matrix can be considered as an indexed-based image. The simulator uses then all the reference and the testing points to generate the "inverse fingerprints" as the distance between each monitor and each point is known. As commonly done in the literature, we have used the path loss model to calculate the RSSI values for all the fingerprints:

$$rssi(d) = rssi(d_0) - 10 \cdot n \cdot log_{10}(\frac{d}{d_0}) - abs - X \tag{1}$$

Where $rssi(d)$ is the RSSI value at a specific distance d and $rssi(d_0)$ the RSSI at a reference distance $d_0 = 1$ m. In the simulator, the path loss exponent is $n = 2$ and X is the normal random variable which causes a log-normal distribution of the received power. In the simulator a X is used with a $\sigma = 4$. abs refers to the absorption and is accumulated over all the obstacles between emitter and receiver. The simulator calculates the line from the point to every receiver and checks through which type of points the direct signals travel (table, bookshelves or pillars). Based on this, the RSSI value is degraded. In contrast to other works where the attenuation is given by obstacle type, we also consider the thickness of the obstacle and the attenuation values are provided per meter. Fig. 1 shows an example of generated propagation model.

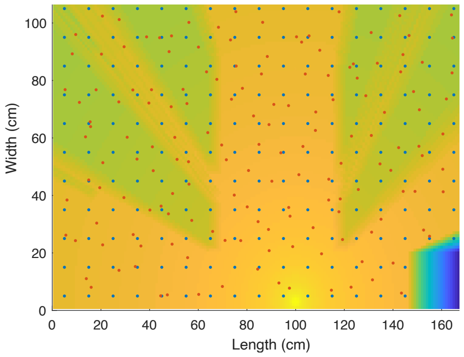


Figure 1: Signal propagation for monitor at [10m,0m] and grid distribution.

In the selected environment, the bookshelves degradation is -25.82 dB/m, the pillar is -20.56 dB/m and the table has no degradation in the simulator. The values for the bookshelves and pillars are based on experiments executed in the office. A Raspberry Pi and beacon are placed one meter from each other with the obstacle (bookshelf, pillar) in between. The average RSSI value collected during two minutes is compared with the same setup without the obstacle. Based on this comparison and the obstacle current thickness, the degradation is acquired. The table/desktop has no degradation because in this proof of concept most beacons are above the tables causing the influence of the tables negligible.

The path loss model used in this simulator is a basic approach to estimate the RSSI value. In order to deal with all the other influences on the RSSI value, a random Gaussian noise component is considered. Random Gaussian noise with $\sigma = 4$ is added to the degraded RSSI values at the receivers of all 187 reference points. For every reference point, 10 RSSI values are saved with different random Gaussian noise in order to deal with the propagation effects. The same principle is done for the test points regarding the noise but only one RSSI value for every test point is used. Next, the data described above is put in the k-NN algorithm with an increasing amount of neighbors (k) going from one to $1,830$. This is the maximum amount of neighbors that can be used in order to compare the RSSI values of the reference points with the test points. Once the training and testing data are generated, the accuracy of the simulated setup is assessed using the k-NN algorithm.

For the environment analysis, a simulation was performed where the total number of receivers in the room increased from 1 to 10. For each number of receiver(s), a total of 100 random setups (i.e. with different random distribution of monitors) are simulated. Moreover, for the 3, 4 and 5 receiver setups, we have added some manual monitor distributions. Those manual configurations were based on our past experience in signal propagation and the environment. In addition, for each setup, the training and testing fingerprint were also generated multiple times with a different random seeds in each run, 100 runs/trials in our case. As the path loss model and the testing samples location have random components, the 100 runs with different random seeds provide a more general behavior of the simulated setup.

The choice to emphasize on this, is because a trade-off has to be made between the most accurate location possible on the one hand, but on the other hand, keeping the deployment cost of the used receivers low. Increasing the number of receivers in a room will benefit the localization purposes, but also comes with a higher deployment price.

In summary, the simulator tool could be seen as a basic 2-D ray-tracing simulation based on Additive white Gaussian noise (AWGN) without considering neither reflection nor diffraction. Despite being a very simple approach, starting from simulated data can help in understanding some of the difficulties the system may have in the operational phase and which configurations may be considered irrelevant.

3.3. Empirical experiments

Once the simulations are all over, and all configurations have been assessed, the best self-selected setup and the best random setup are deployed in a real-world environment. The Raspberry Pi's are mounted to the ceiling of the office, a particular beacon is placed in different locations for calibration purposes, and a set of beacons are used for estimating their position using the reference data of the calibration beacon.

First, the locations of the reference and test points must be determined. In the office a 1 by 1 meter grid is measured and marked with laser precision, resulting in the identification of 140 reference points. Several reference points are not considered because their location is inaccessible or are not relevant for the use of the system. A table located on top of a reference point is not interesting because a person is not going to be standing on top of a table. A total of 20 static beacons are randomly placed in the office at different heights and will not move during the experiment. These test points are also precisely measured.

Second, in each reference point the RSSI is collected for 20 seconds in all the monitors. This 20-second window is split up into ten, two-second windows where the average RSSI is calculated. By doing this the same concept is created as with the simulation. Every reference point will have 10 RSSI values with which to compare. The RSSI collection for the test points is done differently. Over the entire time span of the experiment, a window of 2 s is taken 17 times and the RSSI values of the static beacons are saved. This is the test data for the k-NN algorithm that is compared to the reference points. When for some reason no data was present at a specific moment a value of -200 dBm is allocated for this point. This is done both for the reference and test points. The k-NN algorithm will recognize this value and not use it.

Finally, all the beacons are configured with a transmission power of -4 dBm and a sample frequency of 450 ms.

4. Empirical results

This section is split into simulation (for selecting the best setups) and real-world experiments to assess them.

4.1. Simulation results

The simulation results are showed in Table 1. This table shows the top five setups of all the simulated setups going from one to ten Raspberry Pi's. For every number of Raspberry Pi's 100 random setups are simulated. Every setup is repeated 100 times in order to obtain the general results with different noise vectors. Thereafter, the mean and standard deviation over these one hundred repetitions are calculated. These values are used in the following equation to compare the setups with each other:

$$Score = Mean + STD \tag{2}$$

Where Mean is the average of all the positioning errors calculated by the simulator for all the repetitions for a specific setup. The standard deviation corresponds to the STD in the formula which is also calculated over all the positioning errors for a specific setup. Table 1 shows the mean and the standard deviation in brackets for the top five. The mean provides a general indication of the performance of the setups and by using the standard deviation a statistical element is added. The simulation shows that increasing the number of Raspberry Pi's corresponds with better accuracy. The difference in accuracy is lower with more Raspberry Pi's (6-10 RPi's) which is one of the reasons why the focus of the experiments is on the 3-5 Raspberry Pi's. We also focus on that number of RPi's because of the deployment costs, which do not only in the price but also in providing powering and wired networking.

For all simulations and experiments, a value for $k = 11$ has been used because around this point the best accuracy is obtained. Fig. 2 shows the positioning error when k increases. The bending point for the setups in the graph occurs between $k = 10$ and $k = 20$. When a lower value for k is used the estimated location appears in a completely wrong position because the comparison was made with too few reference points. When the value for k is to large, the expected location will usually be in the middle, causing a large difference between edge points and the expected location. Resulting in a larger overall positioning error.

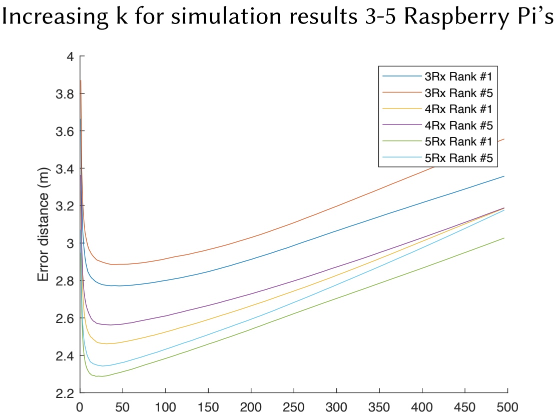


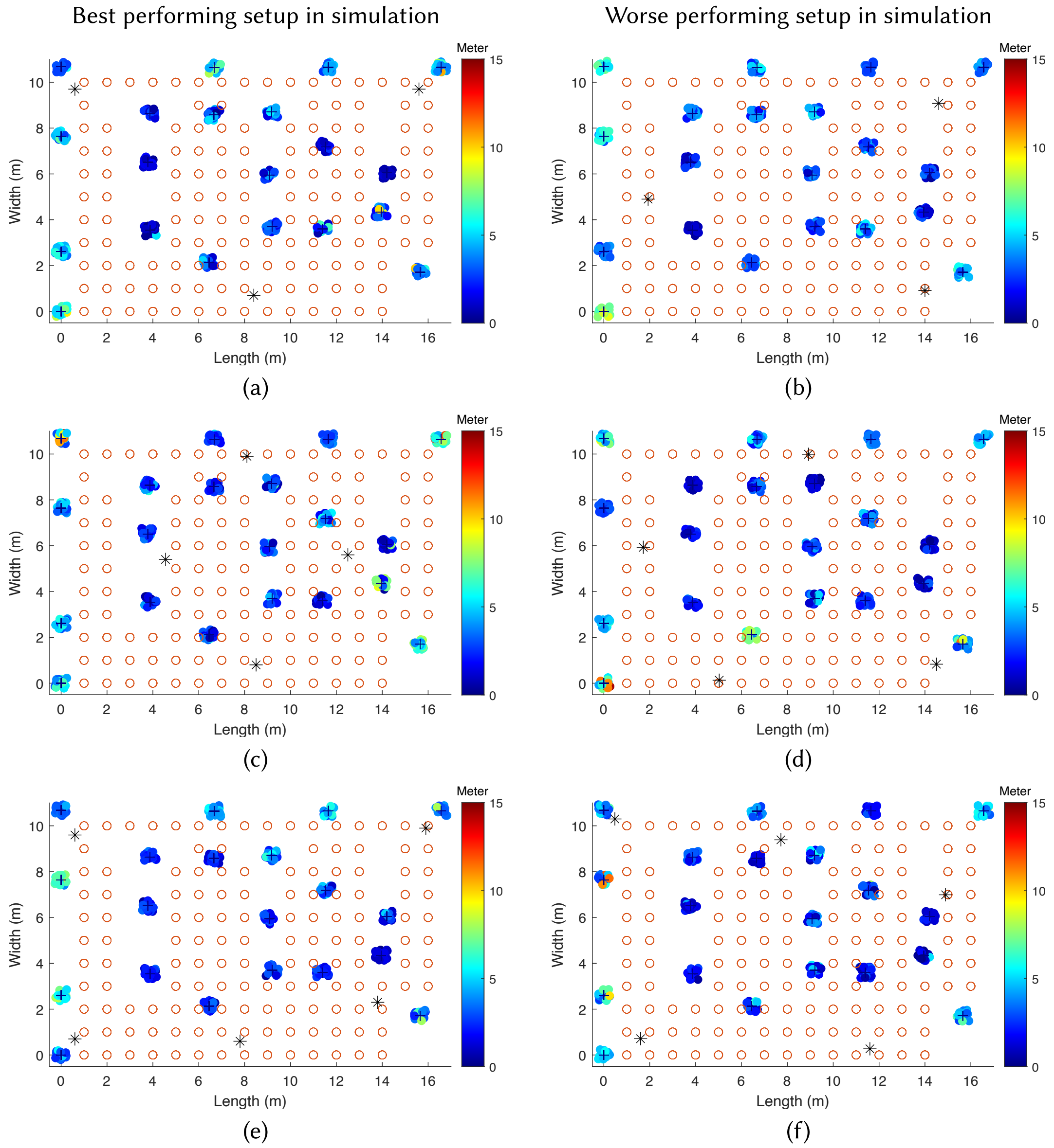
Figure 2: Visualization of the increasing k in the six configurations evaluated with 3, 4, and 5 RPi.

Table 1
Simulation results

	1 RPI	2 RPI	3 RPI	4 RPI	5 RPI	6 RPI	7 RPI	8 RPI	9 RPI	10 RPI
Rank #1	3.11 (1.53)	2.2 (1.32)	1.79 (1.09)	1.54 (0.95)	1.38 (0.85)	1.26 (0.83)	1.13 (0.75)	1.09 (0.68)	0.97 (0.65)	0.96 (0.62)
Rank #2	3.14 (1.52)	2.21 (1.3)	1.82 (1.18)	1.54 (0.97)	1.4 (0.85)	1.27 (0.82)	1.14 (0.73)	1.09 (0.73)	1.01 (0.65)	0.96 (0.62)
Rank #3	3.11 (1.59)	2.26 (1.39)	1.88 (1.11)	1.55 (0.97)	1.4 (0.92)	1.29 (0.81)	1.14 (0.75)	1.09 (0.72)	1 (0.67)	0.95 (0.65)
Rank #4	3.14 (1.54)	2.29 (1.4)	1.85 (1.19)	1.52 (1.06)	1.41 (0.92)	1.3 (0.81)	1.15 (0.74)	1.09 (0.73)	1 (0.67)	0.95 (0.65)
Rank #5	3.15 (1.58)	2.34 (1.33)	1.88 (1.15)	1.57 (0.98)	1.42 (0.91)	1.28 (0.85)	1.15 (0.74)	1.09 (0.72)	1.02 (0.69)	0.97 (0.62)

Fig. 3 shows the results of the experiments. The left column shows the best setup for RPi's 3-5 and the right column shows the worst setup for the same number of RPi's. On all the subgraphs the office is visualised with all the 140 reference points as red circles. The black asterisks correspond to the deployed Raspberry Pi's on the ceiling. The black plus signs indicate the exact spot of the static beacons. These static beacons are located on different heights and sometimes in bookshelves. Random Gaussian noise is added to the estimated position for better visualization purposes. The color of these points indicates the error of that measurement going from 0-15 meters.

Fig. 3a is the best setup for three Raspberry Pi's. The left border of the office shows the biggest positioning error on the measurements. The bounding box for the three Raspberry Pi's covers half of the office resulting in accurate center points. Fig. 3b is the worst setup for three Raspberry Pi's. The bounding box of the Raspberry Pi's in this setup is smaller resulting in a bigger error on the border case.



(a) (b) (c) (d) (e) (f)

Figure 3: Visualization of the positioning errors in the six configurations evaluated with 3 RPi (a,b), 4 RPi (c,d) and 5 RPi (e,f). On the left, the best performing setup and on the right the worst performing setup, according to the simulation.

Fig. 3c has a diamond shaped Raspberry Pi setup. The bounding box of this setup is smaller than these of the three Raspberry Pi's setups. The center points are showing the lowest positioning error but on the left and right side, the border points are performing worse than for the setups with three Raspberry Pi's. In Fig. 3d the bounding box of the Raspberry Pi's is larger but also more randomly distributed. Based on the experiment, the setup in Fig. 3d performs better than the setup in Fig. 3c. This is against the expectations of the simulation. The main cause is that the bounding box of the Raspberry pi's in Fig. 3c is smaller than in Fig. 3d. The simulation only takes into account the largest obstacles and not the area surrounded by three or more Raspberry Pi's.

The Raspberry Pi setup in Fig. 3e is the best based on the simulation. A Raspberry Pi is placed in every corner of the office which creates the largest bounding box. Overall, the center points have the lowest positioning error but the border points show bigger positioning errors. The setup in Fig. 3f has a more random placement of Raspberry Pi's resulting in a slightly worse performing setup.

In general, all the plots shows that some of the testing locations are challenging as they fall out of the convex hull formed by the reference points.

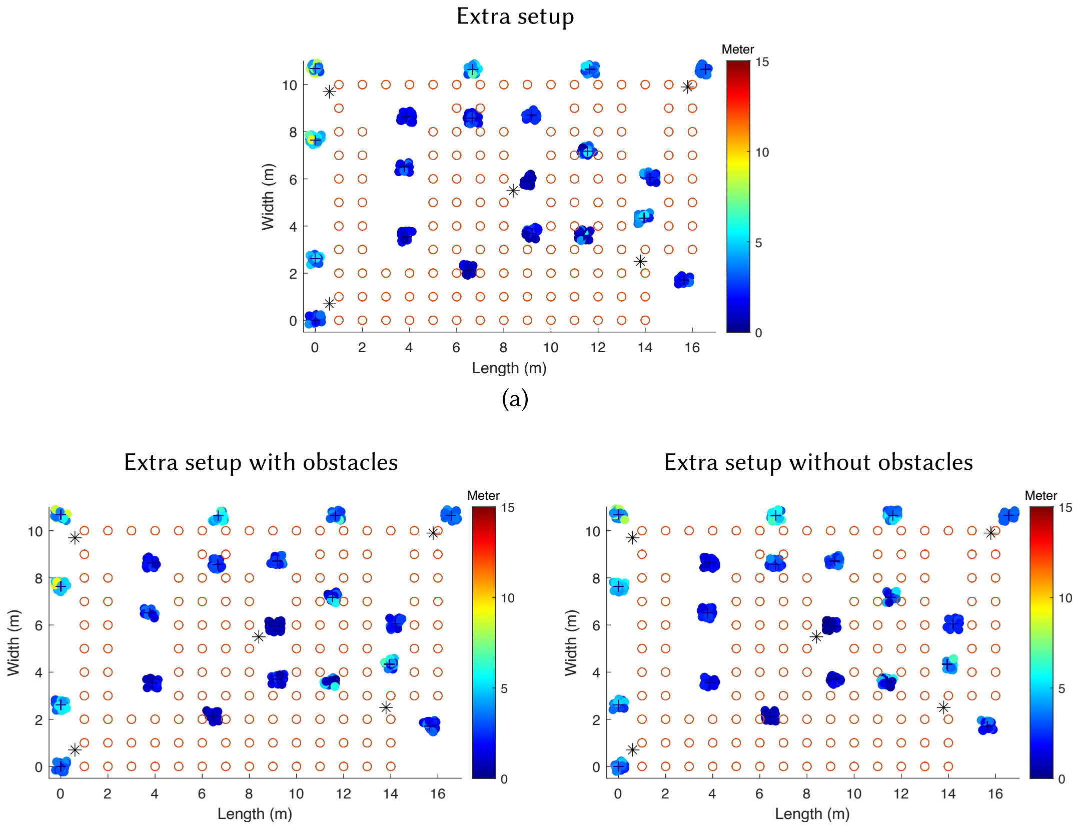
The median and standard deviation for the experiments are shown in Table 2. For three and five Raspberry Pi's the estimated best setups perform better than the estimated worst setup which is in line with the simulation results. The results for the experiments with four Raspberry Pi's are not in line with the simulation results because the estimated best setup performs worse than the expected worse setup. This is due to the small bounding box of the diamond shaped setup. In order to have a consistent relative good accuracy a point has to be surrounded by three or more Raspberry Pi's.

Table 2
Experiments results

	Best	Worst
3 RPI	3.36 (2.03)	3.52 (1.77)
4 RPI	3.59 (2.33)	3.37 (2.43)
5 RPI	3.11 (1.75)	3.15 (1.96)

Based on the experiments in Fig. 3, an extra setup has been tested which should give better results. Fig. 4a shows this setup with a Raspberry Pi in every corner of the office and one in the middle. Generally, the overall positioning is better than all the previous setups. The largest positioning errors occurred in the left border points. In this part of the office, there are many other obstacles such as large plants, banners and boxes. To check the impact of these obstacles on the positioning error, two batches of data were collected. The first batch of data was collected for 20 minutes with the obstacles in the office. The second batch of data was collected for 20 minutes without these obstacles in the office. The measurement with the obstacles is showed in Fig. 4b and the one without obstacles is showed in Fig. 4c.

The results of the extra setup are shown in Table 3. Where the "Normal" column shows the mean and standard deviation of this setup calculated over the 80 minutes of the experiment. In order to see if the bigger obstacles on the left of the office tend to influence the positioning error. The two other columns show the mean and standard deviation of the same setup but with and without the obstacles such as plants, banners and boxes. During a period of 20 minutes the RSSI value is captured and the estimated position is calculated with the same radio map. Whereby, one can only compare the results of columns two and three in Table 3. Removing the bigger objects from the office provides a small improvement on the general positioning errors.



(a)

(b)

(c)

Figure 4: Visualization of the positioning errors in the extra setup evaluated with 5 RPi with mesaruments during whole experiment (a), with obstacles (b) and without obstacles (c) in the office.

Table 3
Experiments results with the additional setup results

	Normal	With obstacles	Without obstacles
5 RPI	2.77(1.64)	2.90(1.72)	2.92(1.60)

5. Discussion

What can be deduced from Table 1 is that using more Raspberry Pi's ensures better accuracy. In addition, a low average does not mean that one setup is better than another. By taking the standard deviation into account, this ensures that a more normally distributed setup will be better. The setups with Raspberry Pi's going from 6–10 show little difference between each other. The error (and variability) for one or two Raspberry Pi's is to high for desired localization purposes.

Table 2 shows the mean and standard deviation for the experiments. In the case of 3 and 5 Raspberry Pi's the best setup performs better than the worst setup. Only, for the 4 Raspberry Pi's the expected worst setup has a better result than the expected best setup. Generally, increasing the amount of Raspberry Pi's is beneficial for localization accuracy. The setup in Fig. 3c does not follow this statement, but the area covering three or more Raspberry Pi's is smaller than in Fig. 3d. Hereby, the corner points are showing greater deviations, increasing positioning error.

The corner points are causing the most problems for all setups. Especially, the corner points on the left side. These corner points are not surrounded by multiple Raspberry Pi's and only one side has recorded reference points in the radio map. The left side of the office consists of different objects such as plants, banners and boxes which absorb the RF signals. The beacons are located on compressed wood which also absorbs part of the RF signals.

The extra setup shows the overall best results. The four Raspberry Pi's are placed in every corner resulting in the largest possible coverage. The fifth Raspberry Pi is placed in the middle of the office in order to surround most of the points by three or more Raspberry Pi's. Comparing the results in columns two and three in Table 3 indicates that removing the major objects in the left corner of the office improves the positioning error slightly.

Overall, we hypothesise that the being within the bounding box of the Raspberry Pi's may be a relevant factor to obtain low positioning error. However, this may not be possible in all locations. Adding Raspberry Pi's helps to reduce the positioning error, but this is accompanied by a higher deployment cost.

6. Conclusions

In this paper, we designed a basic simulator which determines the best possible location for 1-10 BLE receivers in an indoor environment. This is done by the inverse fingerprinting model and path loss equation. The simulator takes into account the major obstacles in the room such as pillars, bookshelves and tables. Based on the path loss model the RSSI value of the beacon is calculated at the receivers. Using a k-NN algorithm and the radio map of the environment the mean positioning positioning error is determined for 153 testing points and more than 100 monitor distributions. The final evaluation metric considers the average and standard deviation over 100 independent runs with different initialization, which is used to compare different BLE receiver setups. In a next phase, two setups for 3–5 BLE receivers are deployed in an indoor environment. The results of these real-world setups are compared with the results of the simulator in order to check if the simulator is a good first approach.

The results of the real-world setups indicate that more factors influence the RSSI value other than the major obstacles. Only the use of the path loss model in the simulator is a basic approximation of reality. In order to simulate the real things such as propagation effects, smaller obstacles, specific materials, occupancy, humidity etc. needs to be taken into account. Eventually, the goal of the simulator is to give a good first approach. Comparing the results of the simulation and the experiment of the same setups show that the expected best setup performs better than the expected worse setup. The bounding box of the BLE receivers turns out to be important and needs to be taken into account in the simulator. The simulator needs room for improvement but already gives a first idea of which setup is better than one another.

The results of all the experiments show an average positioning error of around 3.15 m. The biggest error is in the corner points so that the average error in the middle is lower. Which makes it usable for our monitor application where we want to see what part of the office is overcrowded. The office consists of three compartments and with the measured average positioning error you are able to determine in most cases in which compartment someone is.

Having a good simulator allows assessing a BLE receiver setup before deploying. It shows which setup may perform the best. As future work, we plan to add the bounding box of the receivers in the simulation calculation will improve the overall decision making as this turned out to be a determining factor in the results of the experiments. Also additional concepts, such as GDOP, will be included on top of the current version.

Acknowledgments

We would like to thank Erasmus+ program, which allowed R. de Brouwer to perform his Master Thesis at Universitat Jaume I. Authors acknowledge funding from Ministerio de Ciencia e Innovación through postdoctoral programmes "Torres Quevedo" (PTQ2018-009981) and "Juan de la Cierva" (IJC2018-035017-I). We also thank the reviewers for their comments to improve this contribution.

References

[1] S. O'Dea, Number of smartphone users worldwide from 2016 to 2023, https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, 2021. [Accessed May/2021].

[2] J. Wyffels, J. De Brabanter, P. Crombez, P. Verhoeve, B. Nauwelaers, L. De Strycker, Distributed, signal strength-based indoor localization algorithm for use in healthcare environments, IEEE Journal of Biomedical and Health Informatics 18 (2014) 1887–1893. doi:10.1109/JBHI.2014.2302840.

[3] S. Winter, M. Tomko, M. Vasardani, K.-F. Richter, K. Khoshelham, M. Kalantari, Infrastructure-independent indoor localization and navigation, ACM Comput. Surv. 52 (2019).

[4] P. Desai, K. S. Rattan, Indoor localization and surveillance usingwireless sensor network and pan/tilt camera, in: Proceedings of the IEEE 2009 National Aerospace Electronics Conference (NAECON), 2009. doi:10.1109/NAECON.2009.5426659.

[5] F. Zafari, I. Papapanagiotou, K. Christidis, Microlocation for internet-of-things-equipped smart buildings, IEEE Internet of Things Journal 3 (2016) 96–112. doi:10.1109/JIOT.2015.2442956.

[6] T. Taleb, A. Kunz, Machine type communications in 3gpp networks: potential, challenges, and solutions, IEEE Communications Magazine 50 (2012) 178–184. doi:10.1109/MCOM.2012.6163599.

[7] F. Zafari, A. Gkelias, K. K. Leung, A survey of indoor localization systems and technologies, IEEE Communications Surveys Tutorials 21 (2019) 2568–2599. doi:10.1109/COMST.2019.2911558.

[8] M. Svalastog, Indoor positioning-technologies, services and architectures., 2007. Master's Thesis, University of Oslo, Oslo, Norway.

[9] P. Bahl, V. Padmanabhan, Radar: an in-building rf-based user location and tracking system, in: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), volume 2, 2000, pp. 775–784 vol.2. doi:10.1109/INFCOM.2000.832252.

[10] J. Lee, J. Kim, S.-W. Lee, Y. Ko, A location tracking system using ble beacon exploiting a double-gaussian filter, KSII Transactions on Internet and Information Systems 11 (2017) 1162–1179. doi:10.3837/tiis.2017.02.031.

[11] F. Schneider, A. Branco, A. M. B. Rodrigues, F. Carvalho, S. D. J. Barbosa, M. Endler, H. Lopes, Yet Another BLE Technology Based Tracking System, Association for Computing Machinery, New York, NY, USA, 2021, p. 771–774. URL: https://doi.org/10.1145/3412841.3442095.

[12] J. Kolakowski, V. Djaja-Josko, M. Kolakowski, K. Broczek, Uwb/ble tracking system for elderly people monitoring, Sensors 20 (2020). URL: https://www.mdpi.com/1424-8220/20/6/1574. doi:10.3390/s20061574.

[13] S. R. Jondhale, R. S. Deshpande, Grnn and kf framework based real time target tracking using psoc ble and smartphone, Ad Hoc Networks 84 (2019) 19–28. URL: https://www.sciencedirect.com/science/article/pii/S1570870518305195. doi:https://doi.org/10.1016/j.adhoc.2018.09.017.

[14] R. Belka, R. S. Deniziak, G. Łukawski, P. Pięta, Ble-based indoor tracking system with overlapping-resistant iot solution for tourism applications, Sensors 21 (2021). URL: https://www.mdpi.com/1424-8220/21/2/329. doi:10.3390/s21020329.

[15] J. Li, I.-T. Lu, J. Lu, Cramer-rao lower bound analysis of data fusion for fingerprinting localization in non-line-of-sight environments, IEEE Access 9 (2021) 18607–18624. doi:10.1109/ACCESS.2021.3053994.

[16] S. Du, B. Huang, B. Jia, W. Li, Optimal anchor placement for localization in large-scale wireless sensor networks, in: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2019, pp. 2409–2415. doi:10.1109/HPCC/SmartCity/DSS.2019.00336.

[17] A. Comuniello, A. Moschitta, A. De Angelis, Ultrasound tdoa positioning using the best linear unbiased estimator and efficient anchor placement, IEEE Transactions on Instrumentation and Measurement 69 (2020) 2477–2486. doi:10.1109/TIM.2019.2958011.

[18] A. Schenck, E. Walsh, J. Reijniers, T. Ooijevaar, R. Yudanto, E. Hostens, W. Daems, J. Steckel, Information theoretic framework for the optimization of uwb localization systems, in: 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2018, pp. 1–8. doi:10.1109/IPIN.2018.8533802.