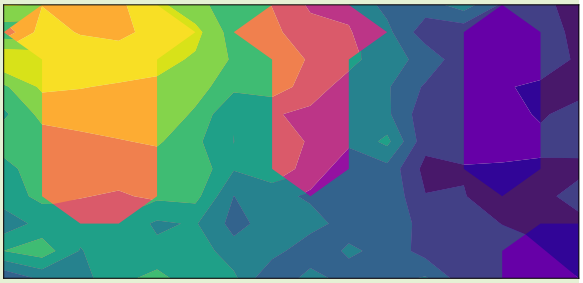
Influence of Measured Radio Map Interpolation on Indoor Positioning Algorithms

Tomas Bravenec, *Graduate Student Member, IEEE*, Michael Gould, Tomas Fryza, *Senior Member, IEEE* and Joaquín Torres-Sospedra

***Abstract*— Indoor positioning and navigation increasingly has become popular and there are many different approaches, using different technologies. In nearly all of the approaches the locational accuracy depends on signal propagation characteristics of the environment. What makes many of these approaches similar is the requirement of creating a signal propagation Radio Map (RM) by analysing the environment. As this is usually done on a regular grid, the collection of Received Signal Strength Indicator (RSSI) data at every Reference Point (RP) of a RM is a time consuming task. With indoor positioning being in the focus of the research community, the reduction in time required for collection of RMs is very useful as it allows researchers to spend more time with research instead of data collection. In this paper we analyse the options for reducing the time required for the acquisition of RSSI information. We approach this by collecting initial RMs of Wi-Fi signal strength using 5 ESP32 micro controllers working in monitoring mode and placed around our office. We then analyse the influence the approximation of RSSI values in unreachable places has, by using linear interpolation and Gaussian Process Regression (GPR) to find balance between final positioning accuracy, computing complexity, and time requirements for the initial data collection. We conclude that the computational requirements can be significantly lowered, while not affecting the positioning error, by using RM with a single sample per RP generated considering many measurements.**



***Index Terms*— Indoor Localization, Indoor Positioning, Interpolation, Radio Map, RSSI, Wi-Fi, Wireless communication**

I. Introduction

CURRENTLY the use of existing wireless communication protocols for indoor positioning and navigation is a growing research topic. The exploitation of radio interfaces and the signal propagation characteristics proved to be a valid approach for Indoor Positioning System (IPS). Since every building is different, be it due to the floor plan, furniture placement or just anchors of the IPS being placed in different locations, the indoor positioning must be adapted for each building. Regardless of technology, the interior heterogeneity is the reason why a majority of IPSs rely on an analysis of the environment first. This analysis can be very time consuming, affected by the size and complexity of the environment, required accuracy and technology employed.

A. Technologies for Indoor Positioning Systems

Indoor localization can be done using several possible wireless technologies ranging from Bluetooth [1]–[3], Wi-Fi [4], Ultra-Wideband (UWB) [5], [6], ZigBee [7], [8], visible light [9] to millimetre wave radar [10] and others employing computer vision [11] or dead reckoning [6], [12] which can work without depending on previously created radio infrastructure. Each of the technologies has its own advantages and disadvantages. Almost every type of User Equipment (UE) made in last decade contains a Wi-Fi interface. This, combined with the widespread availability of Wi-Fi infrastructure, makes Wi-Fi based IPS easy to deploy. Usage of Bluetooth Low Energy (BLE) beacons requires deployment of new infrastructure, but most UE devices are already equipped with BLE wireless interfaces. UWB technology achieves the highest positioning accuracy, but the hardware is not widespread at this moment, mainly because of its higher cost [13].

The authors gratefully acknowledge funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreement No. 813278 (A-WEAR: A network for dynamic wearable applications with privacy constraints, http://www.a-wear.eu/) and No. 101023072 (ORIENTATE: Low-cost Reliable Indoor Positioning in Smart Factories, http://orientate.dsi.uminho.pt/). This work does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of its content. *(Corresponding author: Tomas Bravenec.)*

Tomas Bravenec is with the Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castellón, Spain and also with Department of Radio Electronics, Brno University of Technology, 616 00 Brno, Czechia (e-mail: bravenec@uji.es).

Joaquín Torres-Sospedra is with the Algoritmi Research Centre, University of Minho, Guimarães, Portugal (e-mail: jtorres@algoritmi.uminho.pt).

Michael Gould is with the Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castellón, Spain and also with Esri, Inc. (e-mail: gould@uji.es).

Tomas Fryza is with the Department of Radio Electronics, Brno University of Technology, 616 00 Brno, Czechia (e-mail: fryza@vut.cz).

B. Approaches to Indoor Positioning & Localization

There are three main approaches to locating humans inside a building [14], [15] using radio networks. First there is active positioning when the UE collects and processes the data to obtain its own location. Second and third approaches are passive and neither device nor user do anything to obtain their position, because the anchors analyze either the data packets transmitted by the UE or detect changes in the radio signal propagation. These are not the only options for indoor positioning, because other approaches to indoor positioning do not rely on radio networks.

1) Active Positioning: The active approach to indoor localization works by comparing Received Signal Strength Indicator (RSSI) of the mobile UE devices to previously collected database of measurements known as Radio Map (RM). Comparison to RMs is called fingerprinting, a very popular technology for IPS [16]–[18] because Wi-Fi networks are widely available and do not require any extra deployment costs. The fingerprints are created by the collection of measurements of RSSI at each Reference Point (RP) which is also known as the offline phase of fingerprinting. This approach can be extended by sensor fusion of RSSI and collection of sensor data available to the UE [19]. Another way of active positioning is not by using the signal strength, but by employing range-based localization using radio information like Time of Arrival (ToF), Time Difference of Arrival (TDoF) and Angle of Arrival (AoA) [6], [20], which is currently employed by UWB [21] for centimeter level positioning. The accessibility of this information is dependent on the available capabilities of individual UEs.

2) Passive Device-Dependent Localization: Very similar to the active approach is passive localization that relies on users carrying devices. In this case the positioning infrastructure collects the data transmitted by the UE that is required for obtaining the position. This approach works in almost the same way as the active approach, using a previously collected map of fingerprints.

3) Passive Device-Free Localization: A third approach uses passive localization of users, without relying on the users carrying any device with them. This approach is based on observing abnormal changes in signal propagation created by the interference of human body in the radio environment. In this case we can observe fluctuations in RSSIs or variations in the Channel State Information (CSI) [22]. This approach is becoming more popular as CSI is more stable than RSSI: it is a frequency response to the environment unlike the mere single value of RSSI and it can also benefit from multi-path signal propagation [23], [24].

4) Indoor positioning without Radio Networks: Other approaches such as usage of computer vision [11], or dead-reckoning [6], [12], [25], [26] allow alternative indoor positioning. Dead-reckoning requires the initial location of the user and then by using sensor functions of the UE it incrementally tracks the position of the user. By combining an accelerometer, gyroscope and magnetometer the direction, orientation and speed of movement of the user can be obtained [27], [28]. Additionally with a barometer the movement in vertical axis can be roughly determined [25], [29].

In this paper we focus on the passive positioning determined by the network, assuming the users are carrying a Wi-Fi enabled device. We chose this approach as a follow up to our previous works [30], [31], focused on presence detection using Wi-Fi packet sniffers. This passive with-device method is described in Section I-B.2. We look into the localization using only RSSI based wireless maps. The creation of RMs have several drawbacks:

- They are very susceptible to the changes in the environment, be it moving of furniture, Access Point (AP) or anything else that changes the signal propagation in the area of interest. Any change to the environment may require recreating the RM from scratch.
- The creation of the RM can be very time consuming. The usual approach to creating wireless maps is to collect data at each RP, and the time required for the data collection is very dependent on the size of the space and the density of the RPs.
- Usability of the map of the radio environment is quite limited, since the wireless signal propagates differently in each environment. This makes it very difficult to reuse the same indoor positioning system in different locations.
- Not all locations in the environment can be easily accessed for RSSI collection. This means the positioning accuracy in these spots can suffer.

The ability to create RMs more quickly is very useful for the research community focused on indoor positioning. Spending less time on the measurements collection can simplify the adaptation of IPS for new environments. The difference may be in hours, or even days depending on the environment.

C. Contributions

As opposed to previous works, we do not focus only on the accuracy of the interpolated RM or positioning accuracy, but also on other aspects like:

- *We created a comparison of covariance functions used by Gaussian Process Regression (GPR) and the influence the function have on the generated RM;*
- *The system uses inverted roles of transmitting/receiving devices compared to most works found in the literature. With the anchors receiving the RSSI signals transmitted by the device;*
- *We have compared the interpolated RMs with the measured RM to provide statistical significance;*
- *We provide a new k selection for k-Nearest Neighbors (kNN) use in IPS depending on the radio map features.*
- *We also provide the entire software pipeline ensuring reproducibility and replicability.*

In addition to the above-mentioned contributions, this work also encompasses the entire process from data collection, through clear presentation and selection of hyper-parameters, using different comparison metrics and ending with the significance testing using statistics. We consider these aspects crucial and believe that the scientific community can build upon our work, using it as a baseline for similar research.

II. Related Work

In majority of works related to RM interpolation, the authors work with active positioning, in which the UE collects information necessary for positioning that is transmitted by beacons [32]. We focus on the inverse approach in which we use sniffers to collect information useful for localization transmitted by the UE. This server-side localization is not as commonly used [33]. There has been past research interest using server-side localization with software defined radios [34] as well as Wi-Fi probe requests [35] for passive room occupancy tracking employing RSSI triangulation. Neither one of these works deal with the interpolation of measured data to improve localization accuracy.

Several existing works explore the option of interpolation and extrapolation of collected RMs. Some of these, which look into the interpolation problematic of RMs take an approach of simple linear interpolation, while others use more complex methods. The authors of [36] measured RSSI values at some of the positions and used linear interpolation and the Delaunay (triangulation) algorithm to approximate the rest of the RM, but they did not explore the positioning errors. Apart from these algorithms, the authors of [37] propose to use Kriging algorithm to build a fingerprint database out of sparsely collected data. Another approach to the interpolation of RMs is the use of signal processing graphs [38]. The use of crowdsourcing and subsequent interpolation of acquired RMs also has been explored [39], [40]. More complex approaches to the RM interpolation can utilise deep learning [41]. The kNN and Inverse Distance Weighting (IDW) algorithms are also capable of approximating missing values in RM [42]. Collection of RMs can be improved by the use of inertial sensors for support in the Wi-Fi RSSI location tracking [43]. The authors of [44] also analyzed the use of linear interpolation for approximation of RSSI values inside the convex hull made from the RPs. Additionally, they use minimum, mean and gradients for extrapolation past the convex hull. Apart from this process of using 2 algorithms, they also evaluated the use of kNN and IDW for mixed interpolation and extrapolation of RMs.

In this work we explore the influence the interpolation of RSSI values in missing RPs has on the accuracy and compute requirements in final positioning algorithms. There are several prior works exploring the RM enhancements using GPR for indoor positioning with BLE [45] and with Wi-Fi [46]–[48]. There are also works in which authors explored the use of GPR with Wi-Fi signals for outdoor positioning in industrial environments [49]. Unlike previously published works, we also focus on the differences between created RMs, from the accuracy, computing performance, and statistical significance.

III. Dataset Collection

For the evaluation of the influence of RM interpolation on both the accuracy and computational intensity of subsequent localization algorithms, we collected baseline RM data in our office. The office, whose floor plan is shown in Fig. 1, has a rectangular shape with dimensions of $16.71\,\mathrm{m}$ by $10.76\,\mathrm{m}$.

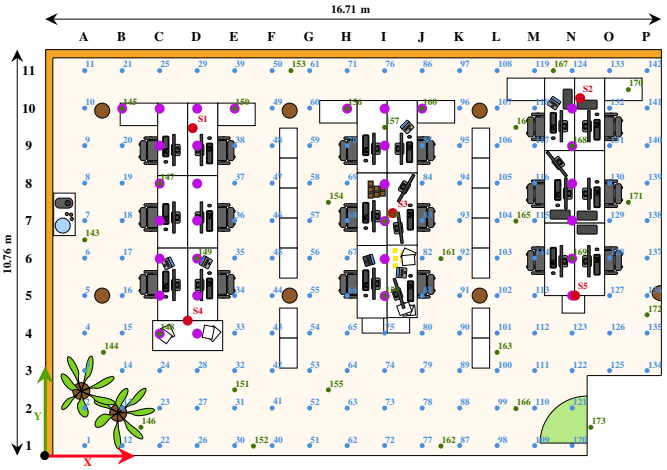


Fig. 1: Floor plan of the office space of INIT department at UJI, Spain. (*S1*–*S5*: ESP32 Sniffer locations, 1–142: Reference Points, 143–173: Evaluation Points).

A. Data Acquisition & Format

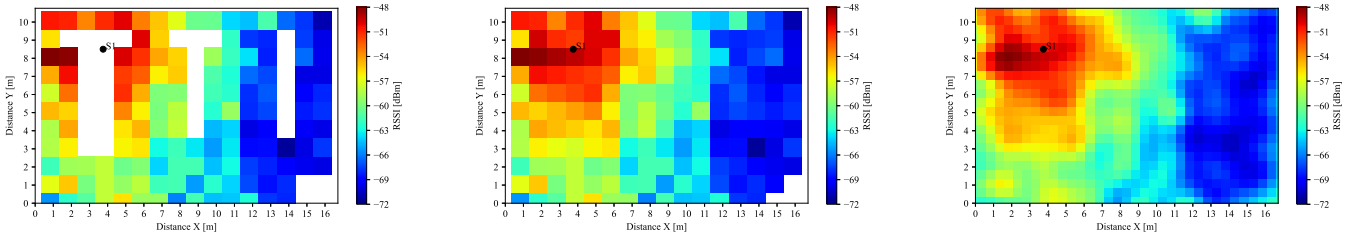
The dataset used for the analysis of the influence of data interpolation was collected using ESP32 micro controllers with firmware based on the one used in our previous works [30], [31]. The ESP32 firmware we used in the past collects Wi-Fi management frames, specifically probe requests, and stores them in a standardized binary packet capture file compatible with Wireshark (network protocol analyser), Scapy (Python package for packet manipulation) and other network traffic analysis tools. We expanded the firmware with the ability to collect RSSI measurements and Media Access Control (MAC) address filtering of incoming probe requests.

To give us control over the transmission of data, we also created a secondary firmware for probe request transmission. This firmware, apart from the ability to send a predefined amount of probe requests when the user presses a button, records the time range in which all probe requests are transmitted. The board employing this firmware was then moved around the 142 reference locations available on the floor of our office to create a Wi-Fi radiomap of the office space. In addition to the 142 RPs, there are 31 locations in the grid we used in our office, that are not easily accessible due to the furniture and equipment. These locations are highlighted in Fig. 1

Additionally, to evaluate the accuracy of the interpolated RMs, we created a small dataset with values collected in both the inaccessible spaces as well as in between RPs of the grid. This allows for better evaluations of locations in inaccessible spaces as well as assessment of the influence the denser RP grid has on positioning accuracy with RSSI values collected in locations outside of the original grid.

The code used in the ESP32 micro controllers, for both the passive sniffers and for the transmitter is publicly available. More details about the availability of the code and reproducibility are in Section VI.

During the data acquisition, we used 5 ESP32 micro controller boards for the collection of probe requests. We chose to use 5 to reduce the influence of RSSI fluctuations on the signal stability by having more data. The reason for 5 sniffers,



(a) RM with measured only data (b) RM with interpolated data (c) GPR approximation of RM

Fig. 2: Visualization of RSSI RM captured by *S1* (ESP32 placement) compared with RM with interpolation of missing values in unreachable RPs using the quickhull algorithm implementation of Python SciPy package.

is coverage of each corner of the office and last one in the center of the room, approximately equidistant from the ESP32s placed in the corners. During the pre-processing of the data, the probe requests were split according to time ranges recorded by the transmitter board. Then we used the sequence numbers available in probe requests for measurement synchronization between all 5 boards. From the probe requests, we only select RSSI and we store them in Comma-Separated Values (CSV) file format including the measurement coordinates.

B. Data Description

In the office, we emitted 50 probe requests at each RP which were all collected by the 5 ESP32 micro controller boards *S1-S5* used for data collection. The period between two consecutive emitted probe requests was 50 ms as it is a good compromise between too short a time period between 2 probe requests and having to spend too much time at one RP. During the transmission, a person stood and rotated around the RP. The reason for this was to simulate a shadow of a person close to the UE in the radio environment and by changing the direction the ESP32 sending probes faces, we reduced the influence of directionality of the Wi-Fi antenna on the board.

With 142 RP easily accessible and 50 samples per RP the dataset consists of 7100 samples. Furthermore, we took 10 % of the data (710 samples) as an evaluation split for the main grid data. For splitting the data we used the helper function of Python package sklearn *train_test_split*. The split and data shuffle is reproducible using *random_state* parameter set to 0. To be able to evaluate the positioning accuracy in spaces we originally did not measure samples in, we collected another 25 samples at 31 locations (see the points 143–173 highlighted in green in Fig. 1) bound to a denser 0.5 m grid excluding all the RPs defined by the 1 m grid. We collected only 25 samples for these locations to avoid creating a bias in the evaluation set, by having significantly more samples in the extra locations. These 775 samples extend the evaluation split of the dataset to fairly evaluate positioning accuracy using interpolated RMs in the hard-to-access locations unused for the RM measurement.

IV. Methodology

In the RM processing we used several algorithms to achieve an approximation of data samples at hard-to-reach RPs and to acquire a higher density of RPs with approximated data in these locations. For the sake of simplicity, we will present the visualizations only for the mean of all values of ESP32 sniffer *S1*. The algorithms we used for data processing are linear interpolation and GPR.

A. Linear interpolation

Before application of more complex methods, we first consider all measurements in a 1 m grid, and we approximate values in unreachable RPs. To achieve this, we used linear interpolation for grid data of the Python SciPy package [50], which uses an implementation of the quickhull algorithm [51]. The Python package SciPy contains implementation of several algorithms, of which we use the Delaunay triangulation to fill in the missing data for hard-to-access RPs. The RM with only the Measured Data (MD) by the ESP32 sniffer *S1* is in Fig. 2a while the resulting RM with approximated missing data using this interpolation technique can be seen in Fig. 2b. In Fig. 2a and 2b the centers of the cells representing the RSSI are matching the RPs aligned in 1 m grid, while in the denser Fig. 2c the grid is 0.5 m. The X and Y axes limits of the figures represent the boundaries of the office.

B. Gaussian Process Regression

Using Gaussian Process Regression [52], [53], we create a model representing the radio space which allows us to get an approximate RSSI value at any coordinates we need. This allows us to generate a RM of signal propagation that has as many RPs as we want.

To use GPR, the selection of covariance function is required. Most commonly used covariance function is the Squared Exponential (SE). In total we selected 3 covariance functions used previously in the literature [47], the SE function, Matern and Rational Quadratic (RQ) functions. We also used compounds of these covariance functions to evaluate the influence on the created RMs. In total we have tested 7 functions. In addition to the 3 functions and their combinations, we have also tested the SE function with fixed length scale parameter. As the deciding metric, we chose the 95^{th} percentile, as it presents majority of prediction. From Table I we can see the differences between positioning accuracy achievable by RMs with different covariance functions are negligible. For proceedings in further evaluation we chose the covariance function SE with fixed length scale, as it provided the lowest 95^{th} percentile in most variations of RM enhancements and in the rest of the cases, the difference is marginal.

TABLE I: Comparison of difference in 95^{th} percentile of positioning accuracy employing RMs created using different covariance functions. In bold is highlighted the best performing covariance function for each RM.

RM	$\frac{Samples}{RP}$ [-]	RM Grid [m]	SE [m]	SE Fixed [m]	Matern [m]	RQ [m]	SE+Matern [m]	SE+RQ [m]	Matern+RQ [m]
Measured RM	50	1.0	**4.73**	**4.73**	**4.73**	**4.73**	**4.73**	**4.73**	**4.73**
RM with LID	50	1.0	**4.68**	**4.68**	**4.68**	**4.68**	**4.68**	**4.68**	**4.68**
RM by GPR trained on MD	50	1.0	4.81	**4.72**	4.85	4.85	4.85	4.79	4.79
	50	0.5	**4.86**	4.98	5.08	5.26	5.08	5.24	5.26
RM by GPR trained on LID	50	1.0	4.73	4.71	**4.66**	4.70	**4.66**	4.72	4.70
	50	0.5	4.89	**4.85**	5.05	5.10	5.05	5.11	5.12
RM by GPR trained on Selection of LID	50	1.0	**4.91**	4.92	4.94	5.02	4.94	5.02	5.05
	50	0.5	5.24	**5.20**	5.28	5.21	5.28	5.36	5.47
RM by GPR trained on MD	1	1.0	5.01	**4.87**	5.13	5.05	4.99	4.99	4.99
	1	0.5	5.34	**5.27**	5.33	5.38	5.62	5.33	5.62
RM by GPR trained on LID	1	1.0	4.95	4.95	4.93	4.95	4.92	**4.91**	**4.91**
	1	0.5	5.34	5.31	5.35	5.30	**5.30**	5.43	5.37
RM by GPR trained on Selection of LID	1	1.0	5.23	5.23	5.36	5.18	6.05	5.18	**5.15**
	1	0.5	5.69	5.69	5.54	5.52	8.49	5.51	**5.26**

The collected dataset has 50 RSSI values per RP and board ($S1$–$S5$). In our first approach, we create 50 GPR models per board, which gives us 50 RMs, each with varying level of RSSI in each RP, that is, the i-th RM is generated using the i-th sample of every RP. In this manner the generated information better mimics the RSSI fluctuations in real setups.

In the second approach, we trained only one GPR model over all available samples, and this approach has 2 advantages:

- Every new RSSI sample is obtained considering information from 50 real measurements. This results in lower noise and reduced influence of outliers in the dataset.
- The total amount of samples is equal to the amount of RPs, subsequently the compute requirements can be much lower compared to solutions using the same distribution but several RSSI values per RP.

Unlike the first approach, having 1 sample per position is not ideal for minimizing the influence of RSSI fluctuations of the Wi-Fi signal on the positioning accuracy. However, the lower amount of reference samples can result in significantly lower compute requirements for the final IPS.

The usage of IPS has issues in situations where we use incomplete data, with lower accuracy around locations with missing data points. Unlike using just linear interpolation, GPR approximates data with a machine learning model, which means that in short distances to the edge of the measured RM it is possible to extrapolate the data by passing the model coordinates that are outside of the room boundaries. The extrapolation can be seen using the top view in Fig. 2c or visualised in 3D in Fig. 3. Three-dimensional visualisations of the mean output of GPR approximated RMs with 50 samples are presented in Fig. 3 with different input data. We omitted the visualization of GPR generated RMs using all input data to create one GPR model, as the surface plots were almost identical to the visualised mean in Fig. 3.

1) Using only MD: In Fig. 3a, only the MD in accessible RP were used for the model training. In the location of the sniffer *S1*, where the device is closest to the network interface of the sniffer, the assumption is the RSSI of the tracked device would be the highest. Using only the MD however, provides different results showing a drop in RSSI value.

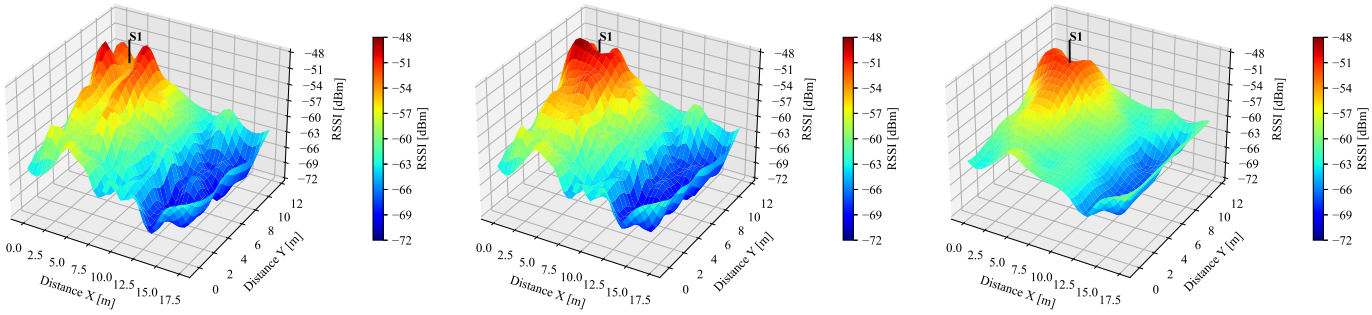
2) Using Linearly Interpolated Data (LID): The second visualization of using GPR to generate denser RM is in Fig. 3b. In this case, we included linearly interpolated RSSI values in inaccessible places as input of the GPR. As can be seen, around the placement of the sniffer *S1*, there is no longer a drop in RSSI as is expected when the devices are in close proximity to each other. Looking at the radio propagation approximation further, we can see that in places further from the sniffer the linear interpolation did not significantly change from Fig. 3a. The similarity to merely using linear interpolation is high, the main difference being the possibility to extrapolate past the edges of the room. We chose to use kNN for testing, which is known to have higher error closer to the walls. The reason being the lack of RPs past the walls results in more neighbours being towards the centre of the room. Extrapolating the RMs past the boundaries of the room allows us to battle lower accuracy at the edges of our office.

3) Using Selection of LID: To explore the ability of GPR to reconstruct RMs, we again used RSSI values with LID in inaccessible RPs. In this case we used the input grid of 2 m, practically skipping every 2^{nd} value on both axes. The approximated RM is visualized in Fig. 3c and is without much of the details present in the RM using as an input all data in the 1 m grid. i.e., the approximation is smother.

We selected point A1 (see in Fig. 1) as starting point for the lower density grid for multiple reasons:

- When starting on the 2^{nd} row of the grid, instead of at 5, we loose samples at 6 points in Y axis.
- By starting with line *B*, instead of *A* we lose more points in proximity to the sniffer locations.
- Since kNN uses k neighboring values to determine the final result, having points around the edges of the room can increase accuracy 1 m away from the wall, which we assume to be more likely place for user to be, rather than staying right against the wall.

Even by starting at *A1* the peak in RSSI close to the placement of the sniffer *S1* is lower in Fig. 3c in comparison to RMs using denser input grid in Fig. 3a and Fig. 3b. By reducing the number of points in close proximity we lose even more detail in the RM.



(a) RM approximation created by GPR with only MD, new RPs distributed in $0.5\,\mathrm{m}$ grid

(b) RM approximation created by GPR with interpolated data at missing RPs with new RPs distributed in $0.5\,\mathrm{m}$ grid

(c) GPR approximation of RM with interpolated data at missing RPs, using every 2^{nd} RP in the interpolated data

Fig. 3: 3D visualization of RMs created using GPR out of only MD, interpolated data with approximated values in unreachable RPs using the quickhull algorithm implementation of Python SciPy package with $1\,\mathrm{m}$ and $2\,\mathrm{m}$ input grid.

V. Analysis & Results

We analysed the influence of RM interpolation in 4 ways. First, we will look at the changes in accuracy, then at the evaluation speed of kNN based IPS. We follow this with a look at the balance between the necessary time for RM gathering, processing time and accuracy of the kNN based IPS. Finally, we use a t-test to determine if there is a statistical difference between the measured RM and all other RMs.

Although there are several enhanced kNN implementations, we have chosen the plain kNN. It is simple, does not require additional hyperparameters to set and is efficient when computing the position centroids. We have placed the reference locations in regular grids of $1\,\mathrm{m}$ and $0.5\,\mathrm{m}$, with the reference data equally distributed over the operational area except near the borders. In a larger operational area with an uneven distribution of reference locations and variable density of fingerprints, a dynamic approach would fit better [54], [55]. Nevertheless, we evaluated other alternatives including Weighted k-Nearest Neighbors (wkNN), Adaptive Weighted k-Nearest Neighbors (aw$k$NN) [54], Self-Adaptive Weighted k-Nearest Neighbors (saw$k$NN) [55], Signal Tendency Index – Weighted k-Nearest Neighbors (sti-kNN) [56] and Distance & Feature Weighted k-Nearest Neighbors (dwfw$k$NN) [57] on the measured RM, producing differences of the Mean Absolute Error (MAE) around $1\,\mathrm{cm}$ to $2\,\mathrm{cm}$ in the best cases.

The measured RM seems to not be sensitive to the positioning method given the density of samples, the number of devices sensing the environment and the lack of missing (not detected) values in the dataset. In this work, we are focusing on the data itself rather than the method, making kNN a good choice for position estimation.

To make the accuracy comparison impartial to the selected k, each RM will benefit from a different k value, depending on the density of the RPs in a grid, amount of reference samples available at each RP and a proximity to the unknown location. Because of this we have formulated a new equation that selects k accordingly:

$$k = \left\lceil \left(\frac{2g}{d} + 1 \right)^2 \sqrt{2n} \right\rceil , \qquad (1)$$

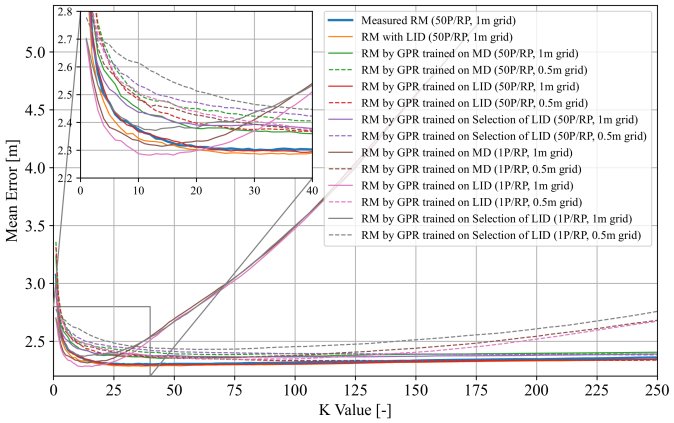


Fig. 4: Comparison of RM processing on the mean positioning accuracy of IPS depending on k of kNN algorithm.[1]

where $[\cdot]$ represents the standard rounding function, g equals to the displacement around the central point (in our case $1\,\mathrm{m}$), d presents the grid density as the distance between neighboring RPs (in our case $1\,\mathrm{m}$ or $0.5\,\mathrm{m}$), and n is the number of reference samples per RP. Additionally, g must be a multiple of d. Selected values of k for each RM are in Table II. The validity of the selection, can be checked by looking at Fig. 4. There we can see the values selected by the equation, match with the k with which each RM achieved best mean accuracy.

A. Influence of processed RMs on positioning accuracy

For positioning accuracy evaluation of the final IPS we use the MAE metric. To evaluate the behaviour of kNN based IPS we measured the influence of k on the accuracy from $k = 1$ up to $k = 250$, when the improvements in achieved accuracy diminished. The results are shown in Table II. The dependency of the accuracy on the selected value of k for all tested RMs are in Fig. 4. Out of the 14 RMs we can easily see a trend. The kNN based on any of the RMs generated using GPR with $0.5\,\mathrm{m}$ grid achieve worse accuracy than the kNN based only on the MD. The difference is not very significant as it differs in the worst cases by $25\,\mathrm{cm}$ from the MD baseline.

[1]Higher resolution Figures are available in the GIT repository [58].

TABLE II: Overview of accuracy and normalized performance achieved with final IPS based on kNN.

Final IPS Input RM	$\frac{Samples}{RP}$ [-]	RM Grid [m]	Selected k [-]	kNN Reference Samples [-]	MAE [m]	Median Error [m]	75th percentile [m]	95th percentile [m]	RMSE [m]	Normalized Time [-]
Measured RM	50	1.0	90	6390	2.32	2.12	2.92	4.68	2.62	1.00
RM with LID	50	1.0	90	7940	2.30	2.10	2.93	4.68	2.60	1.25
RM by GPR trained on MD	50	1.0	90	8800	2.37	2.17	3.07	4.74	2.68	1.38
	50	0.5	250	40 800	2.39	2.16	3.12	4.94	2.73	6.52
RM by GPR trained on LID	50	1.0	90	8800	2.30	2.11	2.95	4.72	2.61	1.39
	50	0.5	250	40 800	2.34	2.13	3.00	4.82	2.66	6.39
RM by GPR trained on Selection of LID	50	1.0	90	8800	2.37	2.16	3.02	4.95	2.72	1.39
	50	0.5	250	40 800	2.38	2.13	3.08	5.11	2.78	6.35
RM by GPR trained on MD	1	1.0	13	176	2.32	2.05	2.93	4.90	2.66	0.03
	1	0.5	35	816	2.38	2.09	3.08	5.24	2.79	0.13
RM by GPR trained on LID	1	1.0	13	176	2.29	2.01	2.89	5.01	2.65	0.03
	1	0.5	35	816	2.39	2.07	3.10	5.26	2.81	0.13
RM by GPR trained on Selection of LID	1	1.0	13	176	2.37	2.10	3.03	5.28	2.77	0.03
	1	0.5	35	816	2.46	2.04	3.26	5.58	2.96	0.13

The best accuracy was achieved by GPR generated RMs with 1 sample per RP, trained on MD and LID. However the accuracy started dropping with the value of k being higher than 16. That is due to the nature of kNN with low number of features; due to that this sensitivity to the k is only present in kNN based on the smaller RMs. The most consistent results were gained by using LID. The improvements in the accuracy were not as big as using GPR generated RMs with 1 sample per RP, but the accuracy was for every tested value of k better than using only the MD for training the kNN model. As expected, the lowest accuracy was achieved by using samples collected at every 2nd RP, where the difference from the baseline is consistently only about 10 cm worse than using all of the MD. That decrease in accuracy can be acceptable as the time needed to create a RM significantly decreases.

To provide more error metrics apart from MAE, we used also median error, 75th and 95th percentile as well as Root Mean Squared Error (RMSE) metrics [59]–[64]. All of these metrics are in Table II. To visually show the distribution of the error, in Fig. 5 we present the Cumulative Distribution Function (CDF) for each of the RMs. The figure also includes expanded views to the median and 95th percentile. When looking at the median, we can see that more than 50 % of evaluation values were predicted with higher accuracy using RMs using only a single sample per RP. This trend though does not continue and when looking at the 95th percentile, the best results are achieved by the RM using linear interpolation, with only measured data and RM created using GPR on LID.

Another factor of accuracy that the indoor positioning depends on is dependency of positioning error on the distance from the edges of the room. Due to fewer neighboring RPs, more neighbors of kNN algorithm are further from the room edge resulting in higher error. This we present in Fig. 6. Fig. 6a is our baseline using only the measured data, where we see that the error increases mostly towards the edge of the office, while in the middle it stays relatively low. For comparison we selected RM approximations created using GPR aligned to a 1 m grid with 50 and 1 sample per RP due to their accuracy performance (see Figs. 6b & 6c). Interpolation does not remove large positioning errors near edges of the office completely, but interpolated RMs provide lower errors there in both cases. While using the RM with only one sample per RP, the error in the rest of the room increased.

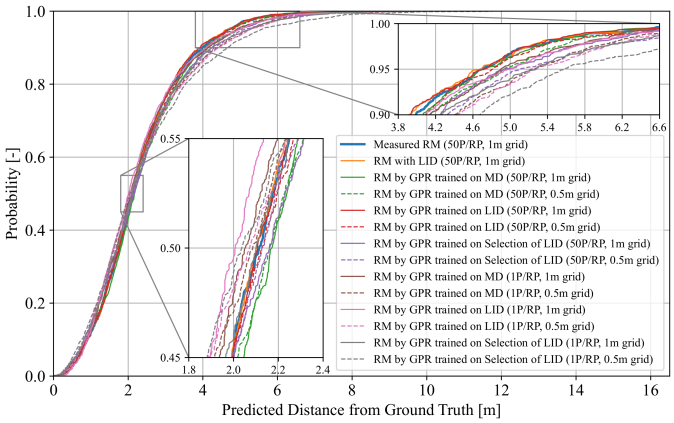
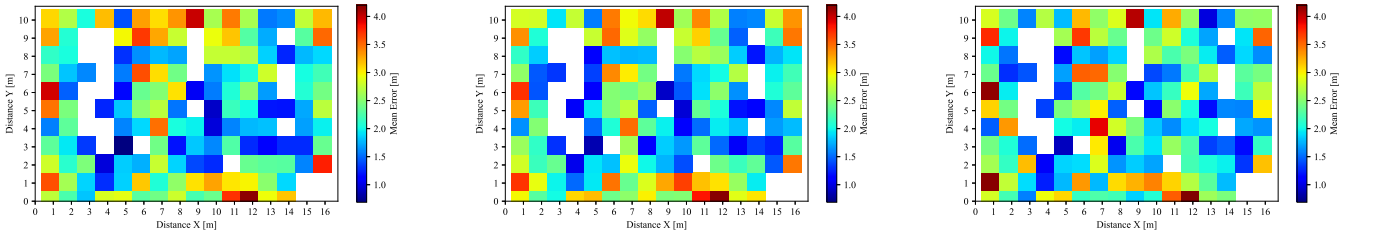


Fig. 5: Comparison of RM processing on the CDF of positioning error, highlighting median and 95th percentile.[1]

B. Compute Requirements Based on RM Complexity

To evaluate the compute performance depending on the RM used for kNN based IPS we normalized all evaluation run times to the baseline performance of RM created out of only MD. We decided to use normalized times to remove the computing hardware from the question and present relative performance increase or decrease depending on the RM that was used. The normalized performance results are in Table II.

The results, as expected, depend on the number of samples in a given RM. This means the RMs with RPs spread out in a 0.5 m grid, instead of 1 m grid used for collecting the measurements, contain approximately 500 % more samples. The performance difference in these cases is unsurprisingly much slower, as can be seen in Fig. 7. Similarly, the RM using LID is a bit slower than the baseline. RMs created using GPR with 1 m grid contain more samples than RM created with LID which again results in slightly slower performance.



(a) RM by Measured Data (MD)

(b) RM by GPR trained on LID, with 50 samples per RP in 1 m grid as output

(c) RM by GPR trained on LID, with 1 sample per RP in 1 m grid as output

Fig. 6: Visualization of MAE of kNN based IPS depending on the RP.

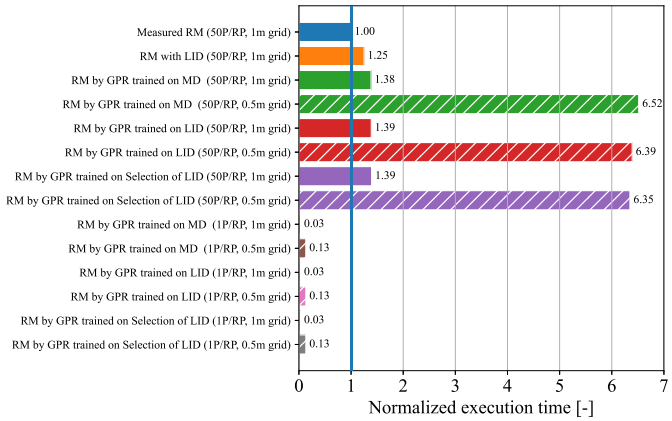


Fig. 7: Comparison of RM processing on relative compute requirements of IPS based on kNN algorithm.

Following the same pattern, the RMs with just 1 sample per RP achieve the best run time performance. Due to the small size of such models, the time required for computation resulted in a fraction of the time required for any RM with 50 samples per RP. Slightly worse computing performance is achieved by the kNN models with just 1 sample per RP but with 0.5 m grid. These models do not suffer from the same issue of the accuracy getting worse with higher values of k, which is due to the RMs having approximately 500 % more samples, but their accuracy gets a big hit.

C. Reductions in RM Collection Time

The baseline MD with 142 RP took us 2 hours and 13 minutes to collect, without considering the time required to mark the reference positions where to stand and broadcast the probe requests. For a relatively small space like our office, this is very time-consuming task and this time increases according to the environment size. The evaluation samples from an additional 31 RPs in about 29 minutes.

Using these collections times, we can estimate time required to collect data in a 0.5 m grid, or in 2 m grid, depending on the application needs. The comparison of the time required to create baseline RM in our office, and approximations through cross-multiplication for RMs with different number of RPs are in Table III. In the table we also compare different densities of grid size as well as collecting data in only accessible as well as in all RP.

TABLE III: Approximation of time required for RM collection in our office acquired using cross-multiplication.

RM type	Grid [m]	RPs [-]	Approximated Time [hh:mm]
Accesible RP	2.0	41	00:38
	1.0	**142**	**02:13**
	0.5	497	07:45
All RP	2.0	47	00:44
	1.0	173	02:42
	0.5	656	10:14

When it comes to the time required for collecting the data of RM, the most time efficient method according to our research is collecting data in a 2 m grid. The accuracy of such RM is lower than collecting RMs using the common and more precise 1 m grid, but the difference in positioning error is not big enough to make a huge difference in positioning of humans. Such RM can be constructed with 400 % fewer RPs, which can save time. In case the application requires higher accuracy, reducing the RP grid is not a good option.

D. Statistical Analysis of Tested RMs

We have used the two-side Paired-Sample t-test with a 5 % significance level to compare the measured RM to the RMs employing LID and GPR in terms of positioning error. These statistical results are reported in Table IV.

From these results, we can see that using linear interpolation to fill in samples at missing RPs significantly improves measured data. The other 2 cases were the measured data improved with GPR trained on the measured data expanded by linear interpolation with both 50 and 1 sample per RP aligned in 1 m grid. For both of these cases, there is no statistical significance as the pairwise difference between individual errors may have a mean equal to zero. In the case of the model obtained with the RM trained with GPR, 1 sample per RP aligned in 1 m grid, the differences in positioning error with respect to the model training with the measured map are not statistically conclusive, but the k-NN matching time is considerably faster. In the remaining cases the proposed solutions are worse in terms of positioning accuracy, being the t-test rejecting the null hypothesis in most of cases.

VI. Data and Software Availability

For full reproducibility of this work, all code and scripts are publicly available for download. The source code for

TABLE IV: t-test and Pearson correlation results comparing the positioning error of the measured RM with others. t-test tests the null hypothesis that the pairwise difference between the accuracy with the measured RM and each of the others has a mean equal to zero, where h indicates whether t-test rejects ($h = 1$) or not ($h = 0$) the null hypothesis at 5% significance level.

Final IPS Input RM	$\frac{Samples}{RP}$ [-]	RM Grid [m]	RM Samples [-]	t-stat [-]	α [-]	Reject Null (h) [-]	Correlation [-]
RM with LID	50	1.0	7940	2.63	0.01	1	0.98
RM by GPR trained on MD	50	1.0	8800	−5.49	0.00	1	0.95
	50	0.5	40 800	−5.32	0.00	1	0.91
RM by GPR trained on LID	50	1.0	8800	1.65	0.10	0	0.98
	50	0.5	40 800	−1.98	0.05	1	0.94
RM by GPR trained on Selection of LID	50	1.0	8800	−3.15	0.00	1	0.89
	50	0.5	40 800	−3.49	0.00	1	0.85
RM by GPR trained on MD	1	1.0	176	−0.19	0.85	0	0.89
	1	0.5	816	−3.17	0.00	1	0.83
RM by GPR trained on LID	1	1.0	176	1.72	0.09	0	0.8
	1	0.5	816	−3.51	0.00	1	0.83
RM by GPR trained on Selection of LID	1	1.0	176	−2.90	0.00	1	0.85
	1	0.5	816	−5.15	0.00	1	0.77

the ESP32 firmware for transmission and collection of probe requests including radiotap information is available from the GitLab repository [58] under the Public Domain licence. The repository also includes Python scripts for data pre-processing and analysis. The collected dataset is also publicly available for the scientific community and is published on Zenodo [65].

The firmware was written for the ESP32-CAM board, due to the onboard SD card slot and compact size. There are several changes that may be necessary to the firmware, in case the deployment board is different, as the SD card slot can use either an internal SDMMC peripheral or SPI communication bus. This depends on the specific board (ESP32-CAM uses an SDMMC peripheral). Changes to the firmware are also necessary to the *config.h* of both firmwares, specifically to set Service Set Identifier (SSID) and password of available Wi-Fi network in the deployment location. Furthermore, there is a need to update the MAC address filter in *sniffer.c* to correspond to the MAC address of the device transmitting the probe requests, be it UE or another ESP32 with probe request sender firmware.

VII. Conclusions

In summary, we gathered Wi-Fi RM of our office using ESP32 micro controllers and analysed the influence the approximation of missing samples of the RM has on the accuracy of kNN based IPS. We tested the approximation of samples at missing RP using linear interpolation and GPR algorithms. Even though we focused on server-side positioning, this approach for enhancing RMs can be used in active positioning as well. From our testing we conclude that the benefits of using interpolated RMs depend on selected processing. Interpolation of RM can increase compute performance or reduce the time required for data collection, while preserving accuracy. For instance, in *RM by GPR trained on Selection of LID*, the collection time quartered (distance between surveyed points is doubled) but the accuracy with respect to the *Measured RM* only dropped $5\,\mathrm{cm}$ (MAE and $11\,\mathrm{cm}$ (75th Perc. & RMSE).

RM with missing samples in hard to access spaces can be completed by using interpolation algorithms. From our testing the most stable performance in the positioning accuracy of the indoor positioning was using linear interpolation of data for RM creation. This approach provided consistently improved accuracy over the measured RM, regardless of the chosen value of k in kNN algorithm. The difference in accuracy though is not high and is in the cm range. Due to more samples, using this RM was slower by $25\,\%$, compared to our baseline with just measured data. The reason for the slowdown is due to the interpolated RM containing more samples and for kNN the computational complexity grows with the amount of reference samples.

Regarding the accuracy, for the RMs created by GPR, with 1 sample per RP aligned in $1\,\mathrm{m}$ grid, the positioning error with respect to the measured RM is not statistically conclusive, however the compute requirements of kNN algorithm are significantly lower. The RM created by using linear interpolation significantly improved measured data and for the cost of higher compute requirements performed consistently better than using just measured data.

In our future work we plan to further focus on the server-side positioning of users. We will further explore possibilities of passive with-device tracking using Wi-Fi and the privacy implications that come with such systems.

List of Acronyms

AP	Access Point
AoA	Angle of Arrival
aw$k$NN	Adaptive Weighted k-Nearest Neighbors
BLE	Bluetooth Low Energy
CDF	Cumulative Distribution Function
CSI	Channel State Information
CSV	Comma-Separated Values
dwfw$k$NN	Distance & Feature Weighted k-Nearest Neighbors
GPR	Gaussian Process Regression
IDW	Inverse Distance Weighting
kNN	k-Nearest Neighbors
LID	Linearly Interpolated Data
MD	Measured Data
MAC	Media Access Control
MAE	Mean Absolute Error
IPS	Indoor Positioning System
RM	Radio Map
RMSE	Root Mean Squared Error

RP	Reference Point
RQ	Rational Quadratic
RSSI	Received Signal Strength Indicator
saw$k$NN	Self-Adaptive Weighted k-Nearest Neighbors
sti-kNN	Signal Tendency Index – Weighted k-Nearest Neighbors
SE	Squared Exponential
SSID	Service Set Identifier
ToF	Time of Arrival
TDoF	Time Difference of Arrival
UE	User Equipment
UWB	Ultra-Wideband
wkNN	Weighted k-Nearest Neighbors

REFERENCES

[1] A. Basiri *et al.*, "Indoor Location Based Services Challenges, Requirements and Usability of Current Solutions," *Computer Science Review*, vol. 24, 2017.

[2] A. J. Martín *et al.*, "BLE-Based Approach for Detecting Daily Routine Changes," in *2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, IEEE, 2021.

[3] L. Polak *et al.*, "Received Signal Strength Fingerprinting-based Indoor Location Estimation Employing Machine Learning," *Sensors*, vol. 21, no. 13, 2021.

[4] M. N. Husen and S. Lee, "Indoor Human Localization with Orientation using WiFi Fingerprinting," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, 2014.

[5] L. Flueratoru *et al.*, "High-Accuracy Ranging and Localization With Ultrawideband Communications for Energy-Constrained Devices," *IEEE Internet of Things Journal*, vol. 9, no. 10, 2021.

[6] A. Alarifi *et al.*, "Ultra WideBand Indoor Positioning Technologies: Analysis and Recent Advances," *Sensors*, vol. 16, no. 5, 2016.

[7] Z. Y. Dong, W. M. Xu, and H. Zhuang, "Research on ZigBee Indoor Technology Positioning Based on RSSI," *Procedia Computer Science*, vol. 154, 2019.

[8] J. Zhen, B. Liu, Y. Wang, and Y. Liu, "An Improved Method for Indoor Positioning based on ZigBee Technique," *International Journal of Embedded Systems*, vol. 13, no. 3, 2020.

[9] Á. De-La-Llana-Calvo *et al.*, "Weak Calibration of a Visible Light Positioning System based on a Position-Sensitive Detector: Positioning Error Assessment," *Sensors*, vol. 21, no. 11, 2021.

[10] J. A. Paredes, F. J. Álvarez, M. Hansard, and K. Z. Rajab, "A Gaussian Process Model for UAV Localization Using millimetre Wave Radar," *Expert Systems with Applications*, vol. 185, 2021.

[11] S. Yang, L. Ma, S. Jia, and D. Qin, "An Improved Vision-based Indoor Positioning Method," *IEEE Access*, vol. 8, 2020.

[12] A. Riady and G. P. Kusuma, "Indoor Positioning System using Hybrid Method of Fingerprinting and Pedestrian Dead Reckoning," *Journal of King Saud University-Computer and Information Sciences*, 2021.

[13] J. Kunhoth, A. Karkar, S. Al-Maadeed, and A. Al-Ali, "Indoor Positioning and Wayfinding Systems: a Survey," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, 2020.

[14] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, 2019.

[15] F. Liu *et al.*, "Survey on WiFi-based Indoor Positioning Techniques," *IET communications*, vol. 14, no. 9, 2020.

[16] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based Positioning in Wireless Local Area Networks," *IEEE transactions on mobile computing*, vol. 6, no. 6, 2007.

[17] S.-H. Fang, T.-N. Lin, and K.-C. Lee, "A Novel Algorithm for Multipath Fingerprinting in Indoor WLAN Environments," *IEEE transactions on wireless communications*, vol. 7, no. 9, 2008.

[18] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-Signal-Strength-based Indoor Positioning using Compressive Sensing," *IEEE Transactions on mobile computing*, vol. 11, no. 12, 2011.

[19] A. Poulose, J. Kim, and D. S. Han, "A Sensor Fusion Framework for Indoor Localization using Smartphone Sensors and Wi-Fi RSSI Measurements," *Applied Sciences*, vol. 9, no. 20, 2019.

[20] Y. Wang and K. Ho, "Unified Near-Field and Far-Field Localization for AOA and Hybrid AOA-TDOA Positionings," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, 2017.

[21] O. Abdul-Latif, P. Shepherd, and S. Pennock, "TDOA/AOA Data Fusion for Enhancing Positioning in an Ultra WideBand System," in *2007 IEEE International Conference on Signal Processing and Communications*, IEEE, 2007.

[22] E. Gönültaş *et al.*, "CSI-based Multi-Antenna and Multi-Point Indoor Positioning using Probability Fusion," *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, 2021.

[23] K. Wu *et al.*, "CSI-based Indoor Localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, 2012.

[24] D. He *et al.*, "3-D Spatial Spectrum Fusion Indoor Localization Algorithm based on CSI-UCA Smoothing Technique," *IEEE Access*, vol. 6, 2018.

[25] J. Geng *et al.*, "Smartphone-based Pedestrian Dead Reckoning for 3D Indoor Positioning," *Sensors*, vol. 21, no. 24, 2021.

[26] S. Jeong, J. Min, and Y. Park, "Indoor Positioning Using Deep-Learning-Based Pedestrian Dead Reckoning and Optical Camera Communication," *IEEE Access*, vol. 9, 2021.

[27] A. Tsanousa *et al.*, "Combining RSSI and Accelerometer Features for Room-Level Localization," *Sensors*, vol. 21, no. 8, 2021.

[28] H. Mehrabian and R. Ravanmehr, "Sensor Fusion for Indoor Positioning System through Improved RSSI and PDR Methods," *Future Generation Computer Systems*, 2022.

[29] J. Li *et al.*, "Improved Height Estimation Using Extended Kalman Filter on UWB-Barometer 3D Indoor Positioning System," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[30] T. Bravenec, J. Torres-Sospedra, M. Gould, and T. Fryza, "What your wearable devices revealed about you and possibilities of non-cooperative 802.11 presence detection during your last ipin visit," in *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2022.

[31] T. Bravenec, J. Torres-Sospedra, M. Gould, and T. Fryza, "Exploration of User Privacy in 802.11 Probe Requests with MAC Address Randomization Using Temporal Pattern Analysis," *arXiv e-prints*, 2022.

[32] R. Melamed, "Indoor Localization: Challenges and Opportunities," in *Proceedings of the International Conference on Mobile Software Engineering and Systems*, 2016.

[33] D. Jaisinghani *et al.*, "Experiences & Challenges with Server-Side WiFi Indoor Localization using Existing Infrastructure," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2018.

[34] Z. Li, T. Braun, and D. C. Dimitrova, "A Passive WiFi Source Localization System Based on Fine-Grained Power-based Trilateration," in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, IEEE, 2015.

[35] J. Scheuner *et al.*, "Probr-A Generic and Passive WiFi Tracking System," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, IEEE, 2016.

[36] J. Racko, J. Machaj, and P. Brida, "Wi-Fi Fingerprint Radio Map Creation by Using Interpolation," *Procedia engineering*, vol. 192, 2017, 12th international scientific conference of young scientists on sustainable, modern and safe transport.

[37] C. Liu *et al.*, "A Kriging Algorithm for Location Fingerprinting based on Received Signal Strength," in *2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, IEEE, 2015.

[38] A. E. C. Redondi, "Radio Map Interpolation using Graph Signal Processing," *IEEE Communications Letters*, vol. 22, no. 1, 2017.

[39] J. Bi *et al.*, "A Method of Radio Map Construction based on Crowdsourcing and Interpolation for Wi-Fi Positioning System," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2018.

[40] J. Bi *et al.*, "Fast Radio Map Construction by using Adaptive Path Loss Model Interpolation in Large-Scale Building," *Sensors*, vol. 19, no. 3, 2019.

[41] R. Hashimoto and K. Suto, "SICNN: Spatial Interpolation with Convolutional Neural Networks for Radio Environment Mapping," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE, 2020.

[42] A. Kiring *et al.*, "Wi-Fi Radio Map Interpolation with Sparse and Correlated Received Signal Strength Measurements for Indoor Positioning," in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, IEEE, 2020.

[43] P. Brida, J. Machaj, J. Racko, and O. Krejcar, "Algorithm for Dynamic Fingerprinting Radio Map Creation using IMU Measurements," *Sensors*, vol. 21, no. 7, 2021.

[44] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-Based Interpolation and Extrapolation Methods for RSS-based Localization with Indoor Wireless Signals," *IEEE transactions on vehicular technology*, vol. 64, no. 4, 2015.

[45] F. Yin, Y. Zhao, F. Gunnarsson, and F. Gustafsson, "Received-Signal-Strength Threshold Optimization using Gaussian Processes," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, 2017.

[46] B. F. D. Hähnel and D. Fox, "Gaussian Processes for Signal Strength-based Location Estimation," in *Proceeding of robotics: science and systems*, Citeseer, 2006.

[47] A. Bekkali *et al.*, "Gaussian Processes for Learning-based Indoor Localization," in *2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, IEEE, 2011.

[48] P. Richter and M. Toledano-Ayala, "Revisiting Gaussian Process Regression Modeling for Localization in Wireless Sensor Networks," *Sensors*, vol. 15, no. 9, 2015.

[49] F. Duvallet and A. D. Tews, "WiFi Position Estimation in Industrial Environments using Gaussian Processes," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008.

[50] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature methods*, vol. 17, no. 3, 2020.

[51] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull Algorithm for Convex Hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, 1996.

[52] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, 2011.

[53] E. Schulz, M. Speekenbrink, and A. Krause, "A Tutorial on Gaussian process Regression: Modelling, Exploring, and Exploiting Functions," *Journal of Mathematical Psychology*, vol. 85, 2018.

[54] S. Liu, R. De Lacerda, and J. Fiorina, "Performance Analysis of Adaptive K for Weighted K-Nearest Neighbor Based Indoor Positioning," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022.

[55] J. Hu, D. Liu, Z. Yan, and H. Liu, "Experimental Analysis on Weight K -Nearest Neighbor Indoor Fingerprint Positioning," *IEEE Internet of Things Journal*, vol. 6, no. 1, 2019.

[56] H. Zou *et al.*, "WinIPS: WiFi-Based Non-Intrusive Indoor Positioning System With Online Radio Map Construction and Adaptation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, 2017.

[57] X. Liang, X. Gou, and Y. Liu, "Fingerprint-Based Location Positoning Using Improved KNN," in *2012 3rd IEEE Int. Conf. on Network Infrastructure and Digital Content*, 2012.

[58] T. Bravenec, *Radio Environment Map Interpolations*, 2023. [Online]. Available: https://gitlab.com/tbravenec/rm_interpolations.

[59] ISO, "Information technology – Real time locating systems – Test and evaluation of localization and tracking systems (ISO/IEC 18305:2016)," *International Organization for Standardization*, 2016.

[60] F. Potorti, A. Crivello, P. Barsocchi, and F. Palumbo, "Evaluation of Indoor Localisation Systems: Comments on the ISO/IEC 18305 Standard," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2018.

[61] F. Potorti, A. Crivello, and F. Palumbo, "The EvAAL Evaluation Framework and the IPIN Competitions," in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, Elsevier, 2019.

[62] F. Potortí *et al.*, "Off-Line Evaluation of Indoor Positioning Systems in Different Scenarios: The Experiences From IPIN 2020 Competition," *IEEE Sensors Journal*, vol. 22, no. 6, 2021.

[63] T. Yang, A. Cabani, and H. Chafouk, "A Survey of Recent Indoor Localization Scenarios and Methodologies," *Sensors*, vol. 21, no. 23, 2021.

[64] J. Bi *et al.*, "PSOSVRPos: WiFi Indoor Positioning Using SVR Optimized by PSO," *Expert Systems with Applications*, vol. 222, 2023.

[65] T. Bravenec, J. Torres-Sospedra, M. Gould, and T. Frýza, *Supplementary Materials for "Influence of Measured Radio Environment Map Interpolation on Indoor Positioning Algorithms"*, Zenodo, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7193602.

Tomas Bravenec (GS'19) received the M.S. degree in electronics and communications from Brno University of Technology, Czechia in 2019. He is currently pursuing the joint-Ph.D. degree at University Jaume I Spain and at Brno University of Technology, Czechia, working as an Early Stage Researcher (ESR) within the A-WEAR project. His research interests include machine learning, indoor localization, and privacy and security issues related to wearable applications.

Michael Gould (PhD, SUNY Buffalo) is associate professor of Information Systems at Universitat Jaume I, Spain. His research and teaching focuses on geospatial information, smart cities, and spatial data infrastructures. He has been PI and researcher on several European and global projects, and was chair of the Association of Geographic Information Laboratories in Europe (AGILE). Since 2009 he also is Global Education Manager at the software company Esri, Inc. where he works on capacity development projects around the world.

Tomas Fryza (SM'20) received the Ph.D. degree in electronics and communication technologies from the Brno University of Technology, Brno, Czechia, in 2006. He has been an Associate Professor with the Department of Radio Electronics, Brno University of Technology, Czechia since 2010. From 2013 to 2021 he was a Vice Head of the Department of Radio Electronics. He was visiting with Simula Research Laboratory, Oslo, Norway, CSC - IT Center for Science, Espoo, Finland, and Umeå University, Umeå, Sweden, in 2012, 2018, and 2022, respectively. He served as an Executive Editor of Radioengineering journal. His research interests include the development of optimized codes for embedded systems, intelligent data analysis, machine learning, and signal processing.

Joaquín Torres-Sospedra received his PhD from Universitat Jaume I in 2011. He is currently an MSCA postdoctoral fellow at the University of Minho (Guimarães, Portugal), where he works on Indoor Positioning (Wi-Fi, BLE, VLC, Machine Learning) for Industrial applications. He has authored more than 160 articles in journals and conferences; and has supervised 16 Master's and 2 PhD Students. Currently, he is supervising 6 PhD students. He is the chair of the IPIN International Standards Committee and IPIN Smartphone-based off-site Competition.