Small UAVs-supported Autonomous Generation of Fine-grained 3D Indoor Radio Environmental Maps

Ken Mendes, Filip Lemic, Jeroen Famaey
Internet and Data Lab (IDLab), Universiteit Antwerpen - imec, Belgium
Email: {name.surname}@uantwerpen.be

***Abstract*—Radio Environmental Maps (REMs) are a powerful tool for enhancing the cognitive awareness of various communication and networked agents by providing localized radio measurements of an environment of interest. Generating REMs is a laborious undertaking, especially in complex 3-Dimensional (3D) environments, such as indoors. To address this issue, we propose a system for autonomous generation of fine-grained REMs of indoor 3D spaces. In the system, multiple small indoor Unmanned Aerial Vehicles (UAVs) are used for 3D sampling of signal quality indicators. The collected readings are streamlined to a Machine Learning (ML) system for its training and, once trained, the system is able to predict the signal quality at unknown 3D locations. The system enables autonomous REM generation and can be straightforwardly deployed in new environments. The system also supports REM sampling without self-interference and is technology-agnostic, as long as the REM-sampling receivers features suitable sizes and weights to be carried by the UAVs. In the demonstration, we instantiate the system design using two UAVs and show its capability of visiting 72 waypoints within 10 min and gathering thousands Wi-Fi samples. Our results also include an instantiation of the ML system for predicting the Received Signal Strength (RSS) of known Wi-Fi Access Points (APs) at locations not visited by the UAVs.**

I. Introduction

A Radio Environmental Map (REM) documents radio signal properties over a given geographic area. These properties can among others include the frequency, protocol and technology of the radio signal, as well as an indication of the quality of the signal (e.g., Received Signal Strength (RSS), Signal-to-Noise Ratio (SNR), or Channel State Information (CSI)), and are stored together with the location where they were measured. These REMs and the data they hold can then be used for a variety of purposes, for example as an aid in cognitive radio networks [1], for Radio Frequency (RF) localization [2], or for optimizing network discovery and handover procedures [3].

As a trade-off to their utility, the generation of REMs is a burdensome process, particularly for 3D environments [4]. Apart from the significant labor needed for carrying out a measurement campaign, an additional complexity stems from the fact that the measurements have to be correlated with the physical locations at which they were collected, suggesting the need for accurate localization of the entity collecting such measurements. The complexity of REM generation is further exacerbated by the fact that such generation often has to be performed periodically, as the REMs can become obsolete due to long-term changes in the signal propagation [5]. Finally, as a variety of different wireless technologies often co-exist [6], the generation of the corresponding REMs is ideally to be done with a single tool for the coexisting technologies.

From the above discussion, there is a need for a system for autonomous generation of 3D REMs, which has been recognized in the community. One of the most promising approaches for such generation is to utilize UAVs as carriers of the REM-sampling devices [7], [8]. This is primarily due to their high degrees of freedom in terms of their optimal positioning in the environment of interest, as well as due to their accurate navigation capabilities [7], [8]. In such systems, the samples generated by the REM-sampling devices are correlated with the UAV-originating location at which the measurements were taken, providing the primitive for REM generation. Such systems have mostly been proposed for *outdoor scenarios* and Global Positioning System (GPS)-enabled large outdoor UAVs [9]. These systems are intuitively not suitable for fine-grained generation of REMs indoors because the UAVs are not practically utilizable in space-constrained and more complex environments. In addition, the GPS-originating location information is not suitable for generating fine-grained REMs due to the relatively large localization errors that such information unavoidably features [10]. This is despite the fact that small UAVs are expected to be an enabler of a variety of indoor applications in domains such as cultural and creative industry [11], Industry 4.0 [12], and civil security [13].

Given their envisaged utility in complex indoor spaces, we argue that UAVs could be designed in a way that the generation of fine-grained indoor REMS can be piggybacked to their application-specific tasks, which would in turn be used for optimizing the communication capabilities and contextual-awareness in the deployment environment. Toward the attainment of the outlined vision, we propose a system for generating fine-grained 3D REMs in small *indoor* scenarios. The system can be viewed as a toolchain consisting of small-indoor UAVs with fine-grained localization capabilities supported by an Ultra Wide-Band (UWB)-based positioning system. The UAVs are envisioned to serve as the carriers of the REM-sampling receivers and provisioners of location annotation for the sampled measurements. The location-annotated measurements are envisioned to be streamlined as training inputs to an Machine Learning (ML)-based part of the toolchain, which, once trained, is able to predict the signal quality at locations not visited by the UAVs.

The most similar existing effort to our work is [14], where the authors propose a UAV-based system for the generation of a training dataset for 3D Wi-Fi fingerprinting-based indoor localization. In contrast to their work focusing solely on Wi-Fi-based fingerprinting, we argue that REMs can be beneficial and utilized more broadly, for example in optimizing the positioning of UAVs serving as mobile relays [15] or planning the extensions of any wireless networking infrastructure by adding Access Points (APs) or base stations to cover "dark" connectivity regions in an environment [16]. Hence, our system abides to two additional design requirements compared to [14]. The first one is a modular design of the interface between the UAV-based system and an REM-sampling device. This requirement allows for a simple integration of different REM-sampling devices (e.g., Wi-Fi, LoRa, BLE, mmWave) with the UAV, extending the REM capabilities beyond traditional Wi-Fi. The second is to guarantee no self-interference between the wireless communication for controlling the UAVs and the REM-sampling device the UAV is carrying, allowing highly repeatable measurement collections with minimized external influences. Finally, we envision the possibility of seamlessly integrating additional UAVs into the toolchain, allowing for sequential data collection and scalable REM generation.

We instantiate the proposed design on two CrazyFlie UAVs to autonomously gather IEEE 802.11b/g/n Wi-Fi beacon frame data in the 2.4 GHz Industrial, Scientific, and Medical (ISM) band. In a real-world indoor environment, we demonstrate that the UAVs are each able to visit 36 waypoints over a 10 min period and collectively gather thousands of Wi-Fi beacon data samples. Finally, we instantiate the remaining part of the toolchain on several contemporary regression- and neural network algorithms, train the system using the data collected by the UAVs, and demonstrate a reasonable prediction accuracy at locations not visited by the UAVs.

II. System Overview

The REM generation is envisioned to be initiated from the control station by providing a set of waypoints to be visited by a fleet of UAVs. The first UAV is envisioned to visit a subset of the provided points, with the main limitation on the number of points that can be visited stemming from the constrained battery capabilities of the UAV. At each location, the UAV instructs the REM-generating receiver to collect the signal quality indicators of interest, and upon each response it reports the obtained results to the control station. This procedure is repeated until all points have been visited or all UAVs in the fleet have depleted their batteries. To visit the instructed locations, each UAV requires a means for self-localization, which is supported through its localization system consisting of a client mounted on the UAV and a set of infrastructural devices (i.e., anchors) for the client's localization.

The main design requirements for the envisioned system include i) accurate location-annotated sampling for streamlined generation of fine-grained 3D REMs, ii) straightforward deployment of the system in unknown complex indoor environments, iii) support for technology-agnostic REM-generating receiver, iv) guaranteed mitigation of self-interference.

We have compared a range of commercial off-the-shelf UAVs based on the design requirements. This comparison can be found in the accompanying technical report [17]. We have decided to utilize BitCraze Crazyflie 2.1 (Figure 1), primarily as it is an open hardware and software platform. Crazyflie 2.1 UAVs come with a FreeRTOS-based operating system and a radio and Bluetooth LE transceivers for control. They also feature an accelerometer, gyroscope, magnetometer, and a high precision pressure sensor through its 10-DOF Inertial Measurement Unit (IMU). These capabilities can be extended by adding up to two expansion boards or decks as shown in Figure 2. In this work, both expansion slots are used: one for the Loco Positioning Deck (LPD), the other for the integration of an REM-generating receiver. The Crazyflie provides a set of interfaces that can be addressed over 20 pins for communication with each expansion deck. Figure 3 documents the pin allocations, where one can choose between an I^2C, SPI, STM32, and two UART interfaces.

A. Interfacing with REM-generating Receivers

The FreeRTOS Crazyflie 2021.06 firmware-flavored custom driver is responsible for interfacing with an REM-generating receiver. The driver should support: i) initializing and ii) checking the state of the receiver, iii) instructing the receiver to collect a measurement, and iv) enabling parsing of the output of the previous instruction. For integration with the UAV, the user is required to provide the driver for the REM-generating receiver to react to the four specified instructions. In terms of hardware integration, the user can choose between UART and I2C interfaces available on the UAVs. We argue that such an *integration procedure is straightforward as it is supported by well-known hardware interfaces and a four instructions-long C driver*, as long as the REM-generating receiver features suitable size (i.e., USB-dongle dimensions) and weight (up to 20 grams) to be carried by the UAV.

B. UAV Localization

We use the Crazyflie's Loco Positioning System (LPS) to provide accurate indoor positioning. This system works with a tag, the LPD, and multiple anchors distributed in the environment. The system can be deployed by simply positioning of the localization anchors, measuring their coordinates relative to a chosen origin, and initializing their automated calibration for synchronizing their transmission schedules. Once the localization anchors are self-calibrated, they can be used for localizing the UAVs and consequently for the generation of a 3D REM. Given that the procedure for the REM-generating system is relying solely on physical deployment, localization, and self-calibration, we argue that *it can be utilized for rapid deployments in complex new environments*.

The LPS is based on the Decawave DWM1000 and uses UWB technology for communication and localization. The tag can estimate its own position based on the UWB signals received from the anchors. The localization is then performed using either the Two-Way Ranging (TWR) procedure or different flavors of the Time Difference of Arrival (TDoA) procedure, the latter featuring slightly better accuracy and supporting simultaneous localization of multiple UAVs. Regardless of the utilized localization procedure, the LPS is able to localize the tag in a 3D environment at the range of up to 10 m [14].

Figure 1: Customized Crazyflie 2.1 UAVs

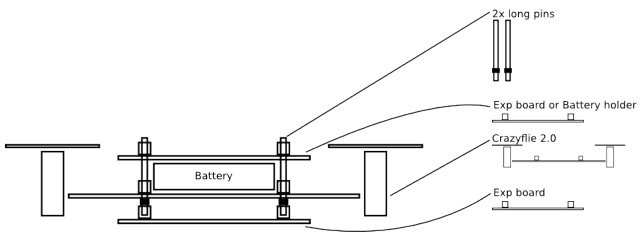

Figure 2: Crazyflie expansion boards

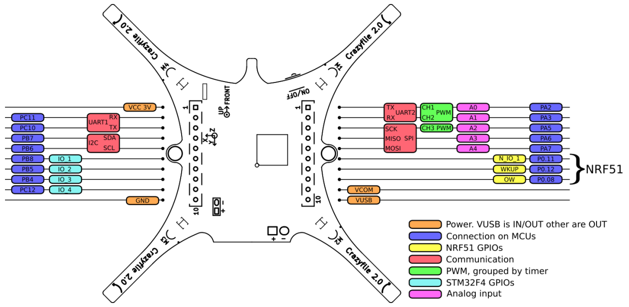

Figure 3: Crazyflie interfaces and pin allocation

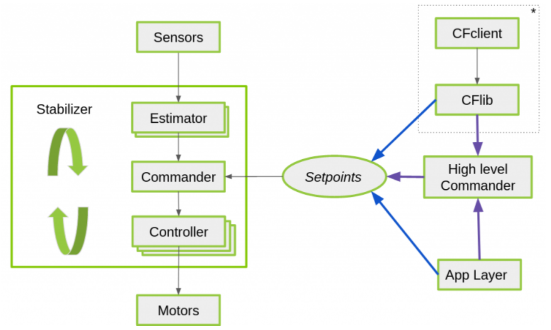

Figure 4: The Crazyflie commander framework

A minimum of four anchors are required to support UAV localization in 3D environments. A Crazyflie equipped with an LPS deck makes use of an extended Kalman filter to estimate its state (orientation and position), its implementation is based on [18]. An increase in the number of anchors increases the robustness and accuracy of this process. Hence, Bitcraze advises to use at least six such anchors to mitigate potential negative effects (e.g., no Line of Sight (LoS) to an anchor, fluctuating tag orientation). Chekuri and Won have shown in their tests [14] that localization with 6 anchors can achieve an accuracy of 9 cm when the UAV is hovering, this is important as we will record the UAV's position and scan while the UAV is holding its position and orientation. In summary, *the system is able to generate location-annotated measurements for REM generation with decimeter-level accuracy at 10 m range*.

C. UAV Communication and Control

UAV control is done through a Python application that uses the Crazyflie Python library. The application can communicate with the UAV to send instructions to move to a waypoint, scan for signal quality indicators and other relevant information, parse the results, and store them for later processing. It runs through the following sequence: i) initialize and instruct the UAV to take-off. For every configured waypoint, make the UAV: ii) move to the waypoint defined with the coordinates $\langle x, y, z \rangle$, iii) initiate an on-demand scan, iv) shutdown the Crazyradio while the scan is running, v) restart the radio connection after the scan is done, vi) fetch the scan results, parse and store them. Communicating with and controlling the Crazyflie UAVs remotely can be done using a custom USB dongle called the Crazyradio. This radio uses a nRF24LU1 chip providing 126 channels and works with a custom protocol for communication: the Crazyradio RealTime Protocol (CRTP). These 126 channels are uniformly distributed over the 2400 MHz - 2525 MHz frequencies.

There are three potential sources of self-interference from the system to the REM-generating receiver. The first two are the positioning system and the propulsion of UAV's rotational engines, although [14] showed that these types of interferences have a negligible effect on the REM-generating receiver operating in 2.4 GHz ISM band. The third potential source of interference is the Crazyradio. The interference generated by the Crazyradio while the System Under Test (SUT) is scanning for available APs at a given location is illustrated in Figure 5. The figure shows the average number of APs detected at different 2.4 GHz Wi-Fi channels for 6 operating frequencies of the Crazyradio and in case when the radio was turned off. As visible (and with more details provided in Section III), the interference from the Crazyradio is significant, irrespective of its operating frequency.

Hence, our experimentation setup features the (default) possibility of automatically turning off the Crazyradio while performing a measurement. To avoid self-interference, the radio is turned off right before the scan starts and restarted again once the scan is finished. The Crazyflie will go into *position hold* mode while the radio is turned off. Small adjustments were made to the firmware to enable operation when the radio is turned off. First, the `CRTP_TX_QUEUE_SIZE` was increased so that full scan results can be temporarily stored until the radio comes back online and the results can be sent to the controlling application. Second, the `COMMANDER_WDT_TIMEOUT_SHUTDOWN` was increased to 10 sec. This timeout is a safety measure, if there is no setpoint received within this interval, the Crazyflie will shut down under the assumption that something went wrong. The default value does not allow to bridge the radio shutdown period.

When the UAV loses its radio connection, it also loses its ability to get new waypoints (i.e., target locations) from the base station. When no new setpoint is received for over 500 ms, the UAV will set its attitude angles (pitch, roll and yaw) to 0 in order to keep itself stabilized. Figure 4 details how the base station's custom Python client can forward waypoints to the *Commander* in the UAV's firmware through the *CFlib* library. To make the UAV hold its position after shutting down the radio, an extra FreeRTOS task was added to the ESP8266 driver that feeds back the scanning position every 100 ms to the UAV's commander during such a scan. This task gets resumed at the start of the scanning task and suspended at the end of it so that it does not interfere with regular waypoint activities. This feedback process results in *the UAV having stability and guaranteed lack of interference while scanning*.

III. Validation

A. Collection of 3D REM-generating Measurements

We demonstrate[1] one instantiation of the proposed system design for small UAVs-supported autonomous generation of fine-grained 3D indoor REMs. The system was deployed in a living room of an apartment in a large apartment building in Antwerp, Belgium. The 3D environment for the UAVs to scan is a rectangular cuboid of 3.74 m long (x-axis), 3.20 m wide (y-axis) and 2.10 m high (z-axis). At each of the 8 corners of the cuboid, an anchor was placed and localized for enabling the UAVs to estimate their locations within the environment. Once the self-calibration procedure was finished, using the controlling application we have instructed a fleet of Crazyflie UAVs to generate an REM of the environment.

We have opted for generating an REM of 2.4 GHz ISM Wi-Fi. This was supported through AI Thinker ESP-01 modules with Espressif Systems ESP8266 Wi-Fi chips. The modules were soldered on Crazyflie prototyping decks and enabled to interact with the UAVs as one of their expansion decks. Using the Crazyflie 2021.06 firmware release as basis, a custom driver communicates with the ESP-01 module over its UART interface by sending AT instructions and parsing the output. Since the module is only used to scan for available APs, it suffices that the driver supports just the following AT instructions: i) `AT` - testing AT start-up, ii) `AT+CW_MODE_CUR` - setting the current Wi-Fi mode (to put the module in station mode), iii) `AT+CWLAP` - listing the available APs and scanning for Wi-Fi beacons, `AT+CWLAPOPT` - formatting the output of the `AT+CWLAP` instruction to $\langle ssid, rssi, mac, channel \rangle$ tuples.

Wi-Fi REM was selected due to self-interference with CrazyRadio, as its frequency range overlaps with the 2.4 GHz band that the Wi-Fi modules use. To get an idea of how pronounced this expected interference is, the Crazyradio was run on different frequencies over its range in 25 MHz increments (2400, 2425, 2450, 2475, 2500 and 2525 MHz). At each of these frequencies, 3 AP scans were done using the ESP-01 module on the Crazyflie. To generate a baseline for comparison, an additional three scans were performed with the Crazyradio turned off. The interference generated by the Crazyradio while scanning for APs is illustrated in Figure 5. It shows for every Wi-Fi channel the average count of APs that were detected over the 3 runs and shows this for 6 different frequencies of the Crazyradio as well as the radio turned off. The Wi-Fi channels that did not feature any detected AP were left out for clarity. These scans were done in a short timespan and with the Crazyflie and Crazyradio in a fixed position. Not only does this data clearly show that the interference of the Crazyradio is significant, it also demonstrates the benefits of our design decision to turn off the Crazyradio while performing REM-generating measurements.

To mitigate interference among UAVs, the UAVs are run in a sequence, not jointly. The LPS is configured to use the TDoA-based localization procedure. The Crazyflie is advertised as having a flight time of up to 7 min depending on how it is used. This is, however, without the weight and power consumed by the LPD and the custom ESP8266 deck. There are also several other factors that can influence the UAV's endurance including flight and scan parameters, the choice between TWR and TDoA or the distance to the anchors, to name a few. To get a notion of the UAV's endurance in a baseline scenario, a UAV was manually flown until it became less responsive and its motions erratic, considering a fully charged standard battery, eight active anchors in TWR mode, periodic scanning mode with an interval of 8 sec, with a beacon scan duration of around 2 sec. The UAV was kept in a steady position about 1 m above ground level for the duration of the test. The UAV was able to perform 36 scans over a timespan of 6 min and 12 sec before it experienced erratic behaviour.

Obviously, the endurance is expected to be lower when the UAVs visit different locations and scan more frequently. With this constraint in mind, 72 locations evenly spread over the volume were identified, with each UAV responsible for scanning 36 of them. The UAVs had 4 sec to fly from a location to another and 3 sec for scanning. Thus, scanning 36 locations was expected to take at least 4 min and 12 sec. Contributing to that the time required for takeoff and landing, the UAVs were expected to operate at their operating limits.

The UAVs were controlled by a base station, i.e., a laptop running the custom Python client software. The client was responsible for sending the UAVs to a waypoint and instructing the scanning. Once the scanning at a waypoint was finished, the UAV would send the results back to the client for parsing and storing for further processing. The client was configured to be able to control multiple UAVs with a matching set of waypoints and parameters such as radio address, starting position, and yaw. While in the paper we show the operation of a two-UAV system, the system can be scaled by simply adding sets of waypoints and above-mentioned parameters.

Using this setup, data was collected for further analysis and processing. A total of 2696 samples were collected, 1495 by UAV A and 1201 by UAV B. During data collection, UAV A was active for 5 min and 3 sec and UAV B for precisely 5 min. In the collected samples, there were 73 distinct Medium Access Control (MAC) addresses and 49 Service Set Identifiers (SSIDs), with the mean RSS of around -73 dBm. When looking at the samples collected per UAV and scanned location (cf., Figure 6), we see no issues with the number of samples collected by UAV B, although this number

[1] Demonstration video: https://youtu.be/fxDkR-Qat6w

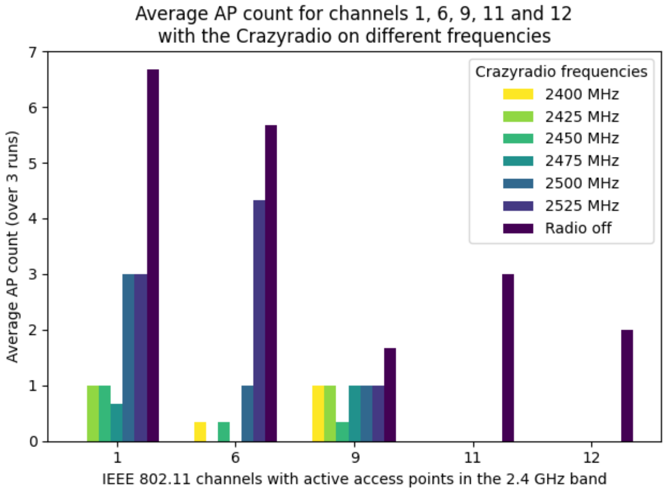

Figure 5: The number of APs detected per IEEE 802.11 channel with the Crazyradio set at different frequencies or completely turned off

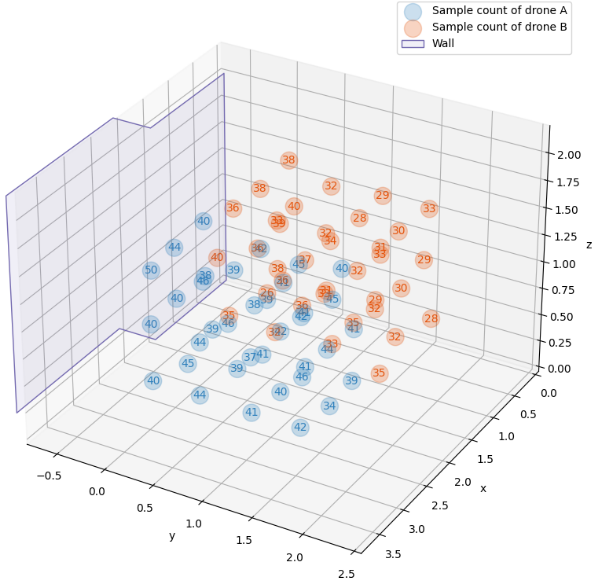

Figure 6: Number of samples per UAV and scanned location

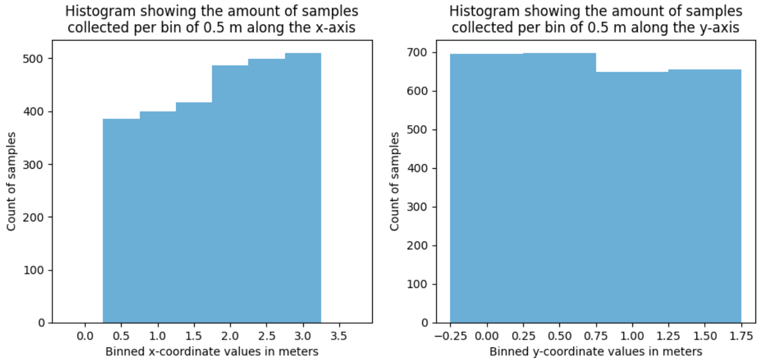

Figure 7: Histograms showing the number of samples collected per bin of 0.5 m along the x and y-axis

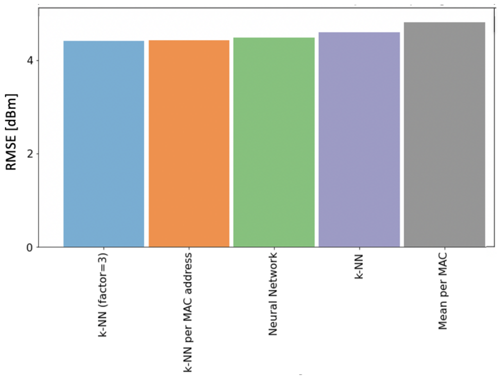

Figure 8: RMSE of prediction for different models

is generally lower than for UAV A. There are environmental factors that can play a role, however i) the positive x-axis and negative y-axis point towards the center of the building where we can expect to see more signals, and ii) there is a wall segment that is 40 cm wider where UAV B's measurements are taken compared to UAV A, as illustrated in Figure 6.

We expected to observe a gradual increase in the number of RSS observations (more APs) toward the center of the building, irrespective of which UAV collected the samples). An illustration of this can be seen in Figure 7, which shows a histogram per axis that groups the x and y values in bins of 0.5 m with the height representing the number of samples collected by the UAVs in that bin. We can clearly see that the number of samples collected increases with an increasing x-coordinate and a decreasing y-coordinate.

B. Generation of Fine-grained 3D REMs

A few pre-processing steps were taken before utilizing the data in an ML model. Since SSIDs can be shared between devices, they were generally not used. Instead, RSS readings were grouped based on their MAC addresses. The timestamps were left out of consideration as well. The time difference between the first and last collected sample was less than 10 min, hence considered as irrelevant. MAC addresses with less than 16 samples were dropped, since the goal was to predict RSS values of APs with sufficient number of measurements. Finally, MAC and channel features were considered as categorical and one-hot encoded. This pre-processing results in 2565 retained samples (131 dropped).

Since the data is locational as it represents signal quality in a 3D space, a k-nearest neighbor (kNN) regressor was considered. As features, the x, y, z coordinates were chosen, as well as the one-hot encoded MAC addresses. This one-hot encoding has the advantage that samples with a different MAC address are considered farther away than similar samples with the same MAC address. The kNN regressor was configured to use Euclidean distance by setting $metric$=minkowski and p=2, as yielded by the the grid search with an exhaustive set of hyperparameters. Similarly, the $weights$ and $n_{neighbors}$ parameters were tuned using a grid search where the optimal values were $weights = distance$ and $n_{neighbors}$=3.

We have considered the samples with different MAC addresses to be further distinguishable by multiplying the one-hot encoded values by the factor of 3 and setting the $n_{neighbors}$ parameter to 16, as yielded by the grid search. Moreover, as an intuitive alternative to assigning samples with different

MAC addresses a greater distance, we considered a kNN estimator per MAC address. To achieve that, we kept the hyperparameters of these regressors the same as previously, and took samples with the same MAC address into account, reducing the feature set to only the x, y, z coordinates.

The last solution we considered was a neural network (NN). In particular, we have built and tested the NN for different configurations, including; i) multiple hidden layers with a varying amount of nodes, ii) normalized RSS values, iii) multiple inputs: 1 for the x, y, z coordinates and 1 for the hot-encoded MAC addresses that get combined into a common hidden layer, and iv) different activation functions and optimizers. The optimized NN had an input layer for the x, y, z coordinates and the one-hot encoded MAC addresses, sigmoid activation function, hidden layer with 16 fully connected nodes, linear activation function, output layer with a single node for the prediction, and Adam optimizer.

The accuracy of the considered estimators was measured based on the Root Mean Square Error (RMSE) of their predictions. In order to have an unbiased view on an estimators' predictive capacity, the preprocessed data was split into a training (75%) and test (25%) sets. For those estimators that require an additional validation set for tuning their hyperparameters, the validation set was taken out of the training set. In order to assess more elaborate estimators we used a baseline estimator that always returns the mean per MAC address.

Figure 8 shows a comparison of the RMSEs for the different utilized predictors. The predictor generally utilizing the mean per MAC address resulted in an RMSE of 4.8107 dBm. The k-NN-based regressors generally yielded slightly better performance than the baseline, with the best performing one being the k-NN algorithm where the one-hot encoded MAC feature was multiplied by the factor of 3 and $n_{neighbors}$=16, resulting in the RMSE of 4.4186 dBm. Finally, the optimally tuned NN with a single hidden layer of 16 nodes yielded the RMSE of 4.4870 dBm. While this is slightly better compared to the baseline, it does fall short of the best kNN solution, as indicated in the figure. This collection of regressors was used with a relatively small set of collected samples, which explains their comparable performance. Nonetheless, we believe that this exercise demonstrates the toolchain-like utilization of the proposed system with different ML tools.

IV. Conclusion

We have proposed a system for autonomous collection of 3-Dimensional (3D) Radio Environmental Maps (REMs)-generating measurements. The main components of the system include Ultra Wide-Band (UWB) localization-enabled Unmanned Aerial Vehicles (UAVs) acting as the carriers of REM-generating devices. The location-annotated measurements for 3D REM generation are then streamlined in a Machine Learning (ML) entity for fine-grained prediction of signal quality indicators at locations not visited by the UAVs.

Future work will include instantiating the system on a larger number of UAVs, followed by exhaustive data collection for a set of representative environments and different wireless technologies. These datasets will be used for deriving the fundamental limitations on the density of 3D REMs. The UAV localization is currently utilizing UWB, hence the REM-generating device has to operate in a different frequency band to mitigate the self-interference effects. This represents an obvious limitation of our system. To address this issue, future work will focus on integrating the BitCraze's infrared system called *Lighthouse* for UAV localization, which features comparable precision, while requiring less anchors and being cheaper. In addition to further self-interference mitigation, this effort is expected to make the system even easier to deploy.

Acknowledgments

Filip Lemic was supported by the EU Marie Curie project "Scalable Localization-enabled In-body Terahertz Nanonetwork" (nr. 893760). In addition, this work received support from the University of Antwerp's Research Fund (BOF).

References

[1] H. B. Yilmaz, T. Tugcu, F. Alagöz, and S. Bayhan, "Radio environment map as enabler for practical cognitive radio networks," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 162–169, 2013.

[2] F. Lemic *et al.*, "Enriched training database for improving the wifi rssi-based indoor fingerprinting performance," in *IEEE Consumer Communications & Networking (CCNC)*, IEEE, 2016, pp. 875–881.

[3] S. Santi, T. De Koninck, G. Daneels, F. Lemic, and J. Famaey, "Location-based vertical handovers in wi-fi networks with ieee 802.11 ah," *IEEE Access*, vol. 9, pp. 54 389–54 400, 2021.

[4] G. Caso *et al.*, "Vifi: Virtual fingerprinting wifi-based indoor positioning via multi-wall multi-floor propagation model," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1478–1491, 2019.

[5] E. Meshkova *et al.*, "Experimental spectrum sensor testbed for constructing indoor radio environmental maps," in *IEEE Dynamic Spectrum Access Networks (DySPAN)*, IEEE, 2011, pp. 603–607.

[6] F. Lemic *et al.*, "Location-based discovery and vertical handover in heterogeneous low-power wide-area networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 150–10 165, 2019.

[7] D. Romero *et al.*, "Aerial spectrum surveying: Radio map estimation with autonomous uavs," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2020, pp. 1–6.

[8] F. Nex and F. Remondino, "Uav for 3d mapping applications: A review," *Applied geomatics*, vol. 6, no. 1, pp. 1–15, 2014.

[9] P. Chhikara *et al.*, "Dcnn-ga: A deep neural net architecture for navigation of uav in indoor environment," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4448–4460, 2020.

[10] D. Lymberopoulos *et al.*, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *ACM Information Processing in Sensor Networks*, 2015, pp. 178–189.

[11] J.-L. Poza-Luján *et al.*, "Indoor drones for the creative industries: Distinctive features/opportunities in safety navigation," in *Drones and the Creative Industry*, Springer, 2018, pp. 129–141.

[12] L. Wawrla, O. Maghazei, and T. Netland, "Applications of drones in warehouse operations," *Whitepaper. ETH Zurich, D-MTEC*, 2019.

[13] M. M. Molina *et al.*, "Ethics for civil indoor drones: A qualitative analysis," *Micro Air Vehicles*, vol. 10, no. 4, pp. 340–351, 2018.

[14] A. Chekuri and M. Won, "Automating wifi fingerprinting based on nano-scale unmanned aerial vehicles," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, IEEE, 2017, pp. 1–5.

[15] I. Rubin and R. Zhang, "Placement of uavs as communication relays aiding mobile ad hoc wireless networks," in *MILCOM 2007-IEEE Military Communications Conference*, IEEE, 2007, pp. 1–7.

[16] K. Daniel and C. Wietfeld, "Using public network infrastructures for uav remote sensing in civilian security operations," DORTMUND UNIV (GERMANY FR), Tech. Rep., 2011.

[17] K. Mendes, *Drone-based Autonomous Generation of 3D Maps of Indoor Radio Environments*. University of Antwerpen, 2020.

[18] M. W. Mueller *et al.*, "Fusing uwb range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *Robotics and Automation (ICRA)*, IEEE, 2015, pp. 1730–1736.