Group Project: RSSI Localization via ESP32

Wong Chor Sing
u3579076@connect.hku.hk
HappyIOT, The University of Hong Kong
Hong Kong

Chun Wing Yee
u3569511@connect.hku.hk
HappyIOT, The University of Hong Kong
Hong Kong

Lee Kwok Lam
lemuelkl@connect.hku.hk
HappyIOT, The University of Hong Kong
Hong Kong

Ko Berrick Eldridge
bko11@connect.hku.hk
HappyIOT, The University of Hong Kong
Hong Kong

1 SYSTEM OVERVIEW

The scanning system consists of three main components with a MQTT architecture: the ESP32, the host machine, and a Python GUI program. The ESP32 is responsible for scanning the Wi-Fi signals and sending the related data to the host machine. The host machine is responsible for receiving the data from the ESP32 as a broker in MQTT. The Python GUI program is for sending control commands, processing the data, and displaying the results. The three components communicate with each other using the MQTT protocol. The ESP32 acts as a subscriber and a publisher; the host machine acts as a broker by using Mosquitto, and the Python GUI program also acts as a subscriber and publisher. The ESP32 publishes the data to the topic SENSOR_DATA to the broker, and the Python program subscribes to this topic to receive the data from the broker. The Python program publishes the commands to the topic COMMANDS to the broker, and the ESP32 subscribes to this topic to receive the commands from the broker. The ESP32 receives the start and stop commands from the broker to start and stop the scanning process. The Python GUI program receives the data from the ESP32 via the broker and further processes the data to get the device's location.

2 ALGORITHM DESIGN

2.1 Maximum RSSI based Localization

This is a simple and brute algorithm that simply locates the client as the location of the AP with the highest RSSI.

2.2 Weighted Centroid

There are three approaches to this method.

(1) Use the inverse square of the distance as the weight. $w_d = 1/(distance^2)$
(2) Use the inverse of signal strength as the weight. $w_s = 1/|RSSI|$
(3) Use the inverse square of signal strength as the weight. $w_s s = 1/(RSSI^2)$

The first approach is the most common one; it first requires calculating the distance between the locations of each access point and label. The second approach is to reduce the impact of the signal strength. The third approach is to reduce the impact of signal strength and increase the impact of signal strength difference.

After calculating the weights, the device's location is calculated by taking the weighted average of the locations of the access points.

2.3 Ranging-based Localization using Least Squares method

This ranging-based algorithm first estimates the distances between the client and the respective access points. With these distances, the least squares method is used to carry out trilateration for localization.

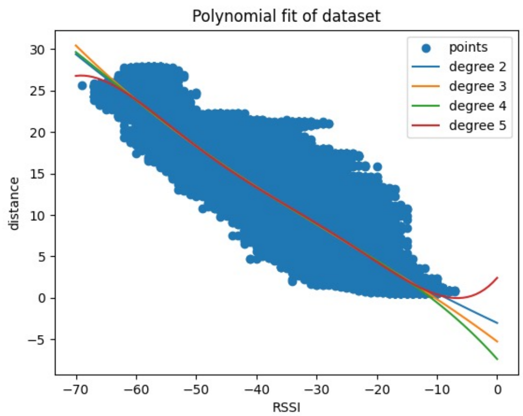
The distance estimation with the Log-Distance Path Loss model [3],

$$P_d = P_{d0} - 10\gamma \log\left(\frac{d}{d_0}\right)$$

where γ is the path loss exponent that depends on the particular environment, in the case of this project, γ is chosen to be 3, assuming that the data measurement happens in an office with a hard partition. P_{d0} is calibrated to $-13.5dBm$ to best fit of given data.

2.4 Ranging-based Localization using polynomial approximation

This algorithm is similar to the one in the above section in that both are ranging-based. This algorithm differs by using polynomial approximation as a trilateration for localization. As mentioned in the paper [2], we perform polynomial regression on the data set, which means an n-th degree polynomial to model the relationship between RSS and the distance. The best fitting is confirmed by plotting the polynomials as in figure 2.4; hence, the polynomial coefficients are obtained through 'Polynomial.fit()' from the Numpy library.



To reduce the computational cost, we adopted the Linear Least Square approach to calculate the solution of location estimation by solving $Ax = b$ with:

$$A = \begin{pmatrix} x_1 - \frac{1}{N}\sum_{i-1}^{N} x_i & y_1 - \frac{1}{N}\sum_{i-1}^{N} y_i \\ \vdots & \vdots \\ x_N - \frac{1}{N}\sum_{i-1}^{N} x_i & y_N - \frac{1}{N}\sum_{i-1}^{N} y_i \end{pmatrix}$$

$$b = \frac{1}{2}\begin{pmatrix} (x_1^2 - \frac{1}{N}\sum_{i-1}^{N} x_i^2) + (y_1^2 - \frac{1}{N}\sum_{i-1}^{N} y_i^2) \\ -(d_1^2 - \frac{1}{N}\sum_{i-1}^{N} d_i^2) \\ \vdots \\ (x_1^2 - \frac{1}{N}\sum_{i-1}^{N} x_i^2) + (y_N^2 - \frac{1}{N}\sum_{i-1}^{N} y_i^2) \\ -(d_N^2 - \frac{1}{N}\sum_{i-1}^{N} d_i^2) \end{pmatrix}$$

denoted x_i is the estimated client location; A is represented by the coordinates of the access points and b as the coordinates and the distances between access points and the client locations.

2.5 Decision Tree Model

Originally, a purely statistical machine learning approach, which is fundamentally Fingerprinting, was proposed. This approach uses a decision tree to approximate f for $y = f(X)$ where X is in a set of vectors of length equal to the number of APs and each element represents the RSSI of that respective AP, and where y is in the set of all possible coordinates in a location.

A decision tree regressor is a machine-learning algorithm for regression tasks. It partitions the input data into segments based on a series of decision rules. Each segment represents a leaf node of the tree, and it contains the predicted output value.

The decision tree regressor starts with a root node representing the entire dataset. At each step, it selects a feature and a threshold to split the data into two subsets. The splitting is done to minimise the error between the predicted values and the actual target values. This process recursively applies to each subset, creating branches and additional nodes.

To make predictions, the decision tree regressor traverses the tree from the root node to a leaf node based on the feature values of the input sample. The predicted output value at the leaf node is then used as the regression prediction for the input.

The limitation of this approach lies in the fact that it disregards the location information of APs. The location of the APs is constant throughout a dataset, making it a constant variable in a statistical machine-learning model that carries no information. This also implies that a model trained on data collected in one environment would not work in a new environment. Data collection must first be done at the desired location (the client location and the respective RSSI values from APs). Regardless, this approach was still demonstrated in the two provided datasets, and the results were successful.

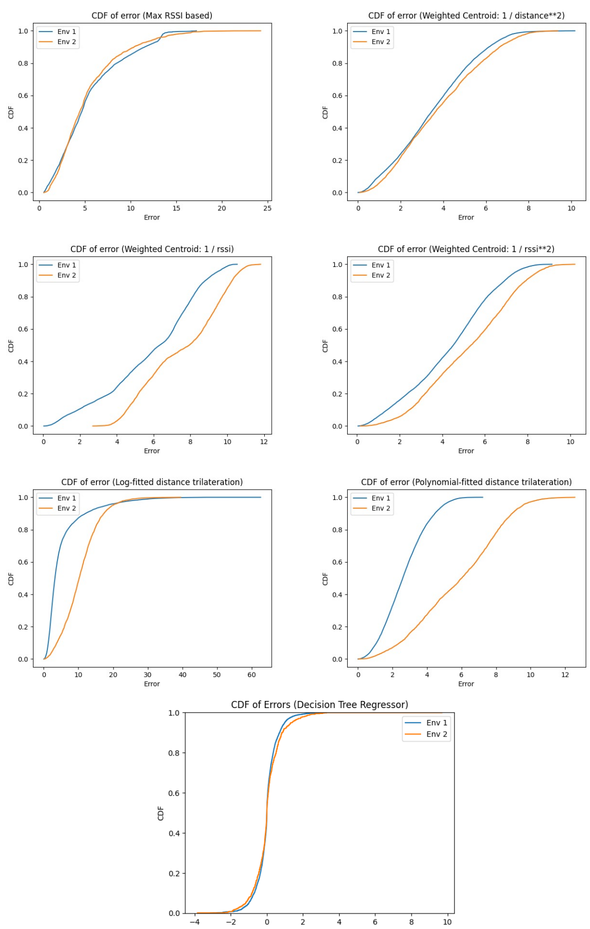
3 EVALUATION

Algorithm	Env	Mean Error	Median Error
Max. RSSI based Lclztn.	1	5.50139	4.65842
	2	5.32448	4.44184
Weighted Centroid			
$1/d^2$	1	3.54381	3.45199
	2	3.82854	3.63933
$1/\lvert RSSI\rvert$	1	5.85162	6.33400
	2	7.54293	7.93781
$1/RSSI^2$	1	4.32412	4.51856
	2	5.23210	5.34775
Ranged L.S. Mthd.	1	5.26615	3.22587
	2	10.57802	10.20996
Ranged Poly. Approx.	1	2.68872	2.601412
	2	5.81265	6.00782
Decision Tree	1	-0.00054	-0.00165
	2	0.04557	0.00467

Table 1: Summary of the Performance Metrics of Algorithms

Table 1 summarises the performance metrics of the algorithms proposed in this report. We believe this performance is close to if not to the practical limit of RSSI localization. The accuracy of RSSI-based localization can be influenced by environmental conditions such as signal interference, multipath propagation, and obstacles. These factors can introduce errors in the distance estimation and subsequently affect the localization accuracy. Also, the performance can vary depending on the deployment scenario. Factors such as the density and placement of access points, the distribution of reference points, and the presence of dynamic objects in the environment can impact the algorithm's effectiveness. Additionally, the performance may be influenced by the hardware and firmware characteristics of the devices used for measurements. Variations in the sensitivity and calibration of the RSSI measurements across different devices can introduce inconsistencies and affect the overall accuracy of the localization.

It is important to note that for the Decision Tree approach, two trees were fitted for the two environments respectively.



Figures 1 to 5 display the cumulative distribution functions (CDF) of errors for the proposed algorithms. In the case of the Decision Tree approach, the steep slope observed at 0 indicates a high probability density for minimal errors. This suggests that the approach exhibits a high level of accuracy, particularly when compared to others.

4 REAL-WORLD EXPERIMENTS

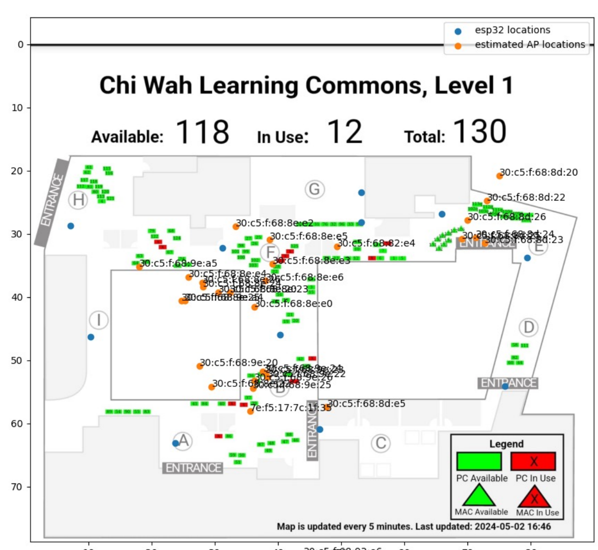
As LE2 is closed for examination, we are not allowed to enter and collect the data by ourselves. Therefore, the system is only tested in the Chi Wah learning commons through war-driving activity.

4.1 War-Driving

4.1.1 Data Collection. To minimize the error, we have collected 3 data samples at 11 locations inside Chi Wah Learning Commons. The locations are marked on the floor plan and the data includes the SSID, RSSI, and the corresponding MAC addresses. The locations are then converted as xy-coordinates, and the RSSI of the same APs are averaged among the 3 samples. It is important to note that some APs are not detected in some locations. In this case, the RSSI is set to $-\infty$ for computational purposes. These processed data are then written into a pickle file in a format aligned with the given dataset.

4.1.2 Localization. Basis filtering is done to the collected data; we keep the AP if it appears over 50% of the dataset. The processed data is then parsed to the script using the method in section 2.4 for calculation. The localization algorithm will then first convert the RSSI to distance and then use triliteration to calculate the device's location.

4.1.3 Result. The APs found by using the algorithm are marked on the floor plan [1], and the corresponding information (SSID, MAC address, locations) is summarized in the table 2. The result is expressed in terms of xy-coordinates on the floor plan for simplicity. For your reference, the image is also included in the project folder.



MAC address	SSID	locations
30:c5:0f:68:8e:26	Y5ZONE	28.02, 37.72
30:c5:0f:68:9e:25	CSL	36.00, 54.37
30:c5:0f:68:8e:e6	Y5ZONE	37.95, 37.33
30:c5:0f:68:8e:25	CSL	24.76, 40.52
30:c5:0f:68:8e:e5	CSL	38.66, 30.92
30:c5:0f:68:8e:23	Wi-Fi.HK via HKU	32.44, 39.28
30:c5:0f:68:8e:24	CSL Wi-Fi Roam	28.07, 38.40
30:c5:0f:68:9e:23	Wi-Fi.HK via HKU	37.98, 52.00
30:c5:0f:68:9e:26	Y5ZONE	36.26, 53.21
30:c5:0f:68:8e:e3	Wi-Fi.HK via HKU	39.14, 34.77
30:c5:0f:68:9e:22	eduroam	38.38, 52.70
30:c5:0f:68:9e:24	CSL Wi-Fi Roam	37.61, 51.74
30:c5:0f:68:8e:e2	eduroam	33.24, 28.87
30:c5:0f:68:9e:20	HKU	27.58, 50.84
30:c5:0f:68:8e:20	HKU	30.54, 39.30
30:c5:0f:68:8e:e4	CSL Wi-Fi Roam	25.86, 36.81
30:c5:0f:68:8d:23	Wi-Fi.HK via HKU	72.70, 31.43
30:c5:0f:68:8e:e0	HKU	36.27, 41.55
30:c5:0f:68:8e:22	eduroam	29.49, 54.20
30:c5:0f:68:8d:24	CSL Wi-Fi Roam	71.28, 30.58
30:c5:0f:68:8d:26	Y5ZONE	69.96, 27.80
30:c5:0f:68:8d:22	eduroam	73.04, 24.79
30:c5:0f:68:8d:25	CSL	69.05, 30.81
30:c5:0f:68:8d:20	HKU	74.99, 20.82
30:c5:0f:2a:74:76	Y5ZONE	16.90, -15.00
30:c5:0f:68:82:e4	CSL Wi-Fi Roam	49.27, 31.98
30:c5:0f:68:9e:a5	CSL	17.98, 35.18
30:c5:0f:68:8d:e5	CSL	47.72, 57.42
30:c5:0f:68:9e:a4	CSL Wi-Fi Roam	25.29, 40.55
30:c5:0f:68:93:e6	Y5ZONE	44.07, 80.75

Table 2: Estimated APs locations and their identification

5 SUMMARY

In summary, we have successfully developed a comprehensive localization system using Wi-Fi RSSI and ESP32. This system has proven to deliver good outcomes for client and AP localization tasks. Throughout our research, we thoroughly examined five distinct localization algorithms, incorporating wireless technologies and statistical analysis expertise. Mathematical analysis has extensively discussed and supported the algorithm designs, ensuring their credibility. To assess performance, we have systematically evaluated each algorithm using two provided datasets, employing key metrics such as mean error, median error, and CDF. Furthermore, real-world experiments conducted during war-driving have demonstrated the functionality of our system and its adaptability to various environments.

6 CONTRIBUTION STATEMENT

Every member shares approximately the same workload.

REFERENCES

[1] 2024. *Chi Wah Learning Commons, Level 1*. https://hkuesd.hku.hk:8000/pcavail/lc1a.svg

[2] Yingying Chen Jie Yang. 2009. Indoor Localization Using Improved RSS-Based Lateration Methods. (2009), 6.

[3] The University of Hong Kong. 2024. COMP3516: Data Analytics for IoT, Lecture 9 Wireless Localization. (2024), 35.