# Implementation of a WiFi-based indoor location system on a mobile device for a university area

Dennis Núñez Fernández
*Universidad Nacional de Ingeniería*
Lima, Peru
dnunezf@uni.pe

***Abstract*—This paper describes the implementation of an indoor location system on a mobile phone for the Faculty of Electrical, Electronic and Telecomunications at the Universidad Nacional de Ingeniería, Peru. The proposed system makes use of WiFi signals and the Bayes filter in order to predict the user location. The principal advantage of the proposed method in comparison with similar methodologies is that our system does not need additional hardware or special equipments. As the experimental results show, the system achieves 92.31% for position accuracy and works in real-time. In addition, the proposed methodology could be used for similar location tasks.**

***Index Terms*—Indoor Location System, WiFi, Received Signal Strength, Position Fingerprint, Bayes Filter.**

## I. Introduction

Current GPS (Global Positioning System) and related location systems can obtain high accuracy in outdoor environments, however, for indoor environments they obtain low level position accuracy because the satellite signals can not be efficiently received in indoor regions [1]. As a result, the implementation of indoor location systems has attracted many researchers due to the need for these kind of systems in real applications like asset tracking, proximity marketing, immersive experiences, indoor navigation, augmented reality, controlling robots in a warehouse, and more.

At present, there exist many indoor location systems [2], including infrared (IR), ultrasonic, radio frequency identification (RFID), wireless local area networks (WLAN), Bluetooth, ultra wideband (UWB), and others [3] [4]. In [5], a precise WiFi-based indoor location system using Monte Carlo (MC) filter is proposed, however, the system is intended for tracking precise trajectories, which it is not suitable for our project. In [6], an indoor positioning system for smart buildings is used, but the system makes use of RFID technology, which results in additional costs. In a more recent work, [7] describes a novel system that uses machine learning techniques, nevertheless the system is intended for centimeter level location and relies on visible-light technology. In addition, indoor Google Maps [8] provides guidance in buildings, but its accuracy is not enough, according to our experience.

In this work we developed a system for indoor location in the Faculty of Electrical, Electronic and Telecomunications at the Universidad Nacional de Ingeniería, Peru. This system does not demand additional hardware such as RFID devices, Bluetooth beacons or implementation of special electronic hardware that result in additional costs. Therefore, by only using six pre installed WiFi access points and a mobile phone our system is capable to achieve a great position accuracy. In order to have an efficient predictor that consumes low computational resources and works in real-time, a predictor based on Bayes filter was implemented.

## II. Methodology

### A. Overview

The proposed indoor location system operates in real-time and makes use of six pre installed WiFi access points (APs), whose location is unknown. This system is intended to work on Android mobile devices with low computational resources and using low power consumption. In order to accomplish response time, computational demand and energy requirements, we employ a simple but effective estimator based on Bayes filter.

The diagram of the entire system is presented in the Fig. 1. As the image shows, the proposed methodology has two phases: offline phase and online phase. During the first one, offline phase, we captured the WiFi levels several times per region and obtain an RSSI table per region, then the RSSI data is divided into training and testing datasets, later we construct the histogram and clean the training RSSI dataset by using Gaussian curve. During the online phase, Bayesian filter is implemented in order to perform analysis and confusion matrix by using the testing dataset.

Finally, the Bayesian filter is implemented in an Android mobile device. This filter uses the captured WiFi levels and the cleaned RSSI database to predict the most likely region.

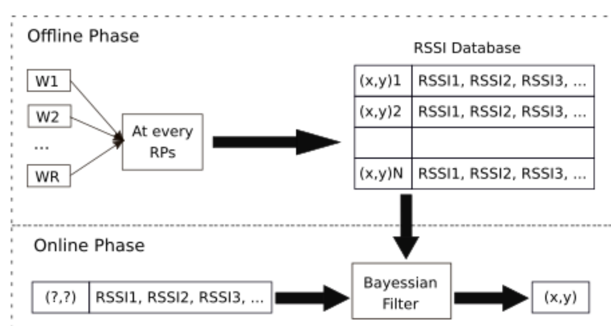



Fig. 1: Diagram for the proposed system

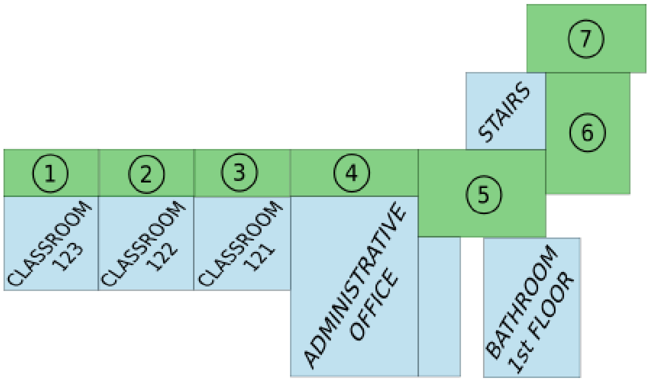



Fig. 2: Distributions of regions for the first floor

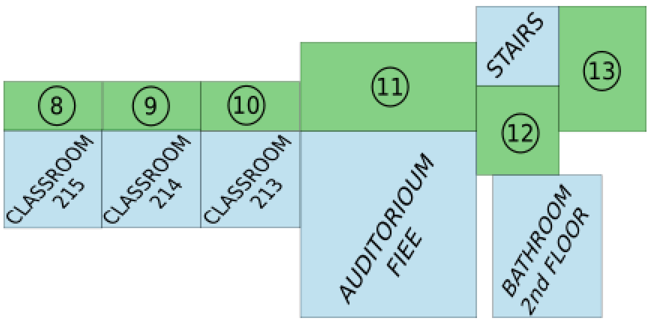



Fig. 3: Distributions of regions for the second floor

## B. Data Recolection

As mentioned in the previous section, the environment is the Faculty of Electrical, Electronic and Telecomunications at the Universidad Nacional de Ingeniería, Peru. So, first of all, the main areas are identified and the total environment is divided into regions larger than 2x2 meters and less than 5x5 meters. Fig. 2 and Fig. 3 depict such as distribution of the regions. We should notice that only the green regions (access areas to different main environments) will be used for indoor location since these regions helps to find the desired main environment for visitors purposes, which are the blue regions.

After the division of the regions, an application in Java for Android devices is implemented to scan and capture the WiFi information (SSID/MAC/RSSI are extracted for all APs). This app is used in order to capture 150 samples per region, then this data was saved in a CSV file.

The captured samples in each region were taken by starting at the central point of the selected region and then moving around different points of such a region. This process of data recollection was taken at different times during three non consecutive days, at day and night.

Later, the captured data is saved in CSV files in the internal memory of the mobile device. Finally, the data is moved from the device to a Personal Computer and the best APs are selected for the final captured dataset.

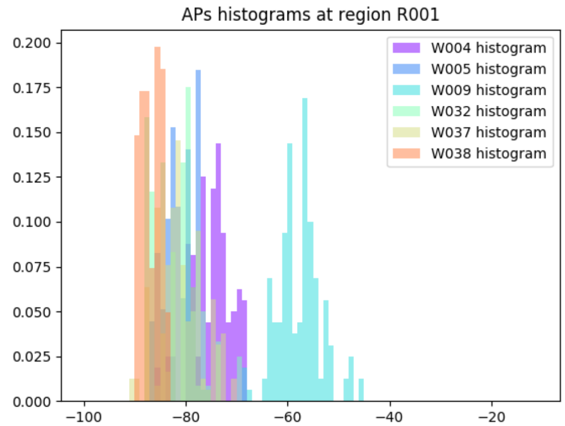



Fig. 4: Sample of the original histograms

## C. Pre-processing

After we have the final captured data on a PC, the data is put in order, obtaining an RSSI table per region. Then, in order to perform the offline analysis of Bayesian filter, the dataset is divided in training/testing datasets. So, we randomly shuffled the samples of each CSV file, and then divide in 80% for training (120 samples) and 20% (30 samples) for testing.

Using the training RSSI dataset, the normalized histograms (for each AP and region) are calculated. However, since the original training RSSI database is formed of a limited number of samples (affected by reflections and scattering), the calculated histograms are composed by missing RSSI values and noisy histogram values, see Fig. 4.

Several papers related to indoor location and based on RSSI measurement, assume its Gaussian probability density function (PDF) [9] [10]. This is excused by relation to PDF of radio-receiver's noise or together with influence of average white Gaussian noise (AWGN) radio-channel which is generally modelled by a normal PDF.

Therefore, the original histograms are cleaned by approximating each histogram to a Gaussian PDF. Fig. 5 shows the formulation of a Gaussian PDF, which is defined by two parameters: $\mu$ (mean) and $\sigma$ (standard deviation). In our case, these parameters were extracted from the 150 samples.

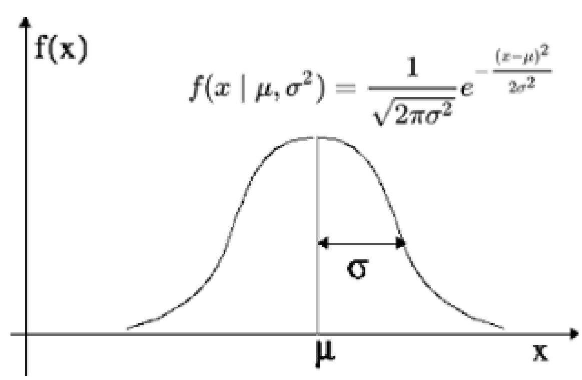



Fig. 5: Description of Gaussian PDF

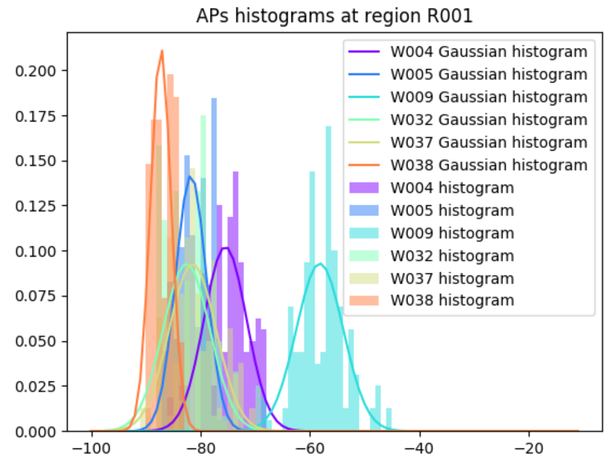



Fig. 6: Sample of original and cleaned histograms

Fig. 6 shows the original histograms and the cleaned histograms (Gaussians histograms). As the figure shows, the original histograms can be well approximated by Gaussian PFDs. This new distribution completes the missing values of the RSSI histogram and modify the values which were disturbed by strong reflections and scattering. Consequently, after clean the normalized histograms, a new dataset composed only of the normalized Gaussian histograms is obtained.

Before performing the Bayesian filters, one AP table for each access point is constructed. Each AP table will be made up of the union of the Gaussian histograms of different regions which have the same AP. The Table I shows its composition.

TABLE I: Examples of histogram tables

(a) AP table for router 1 (W1)

| **Region** | **Gaussian histograms** |
|---|---|
| Region 1 | W1 Gaussian histogram at Region 1 |
| Region 2 | W1 Gaussian histogram at Region 2 |
| ... | ... |
| Region N | W1 Gaussian histogram at Region N |

(b) AP table for router 2 (W2)

| **Region** | **Gaussian histograms** |
|---|---|
| Region 1 | W2 Gaussian histogram at Region 1 |
| Region 2 | W2 Gaussian histogram at Region 2 |
| ... | ... |
| Region N | W2 Gaussian histogram at Region N |

### *D. Bayesian Filter*

The Bayesian recursive estimator is able to infer the posterior state based on current sensed and prior knowledge [11]. As formulated in (1), $A$ is the event we want the probability of, and $B$ is the new evidence that is related to $A$ in some way. Thereby, the posterior value $P_{(A|B)}$ is estimated by the multiplication of the likelihood $P_{(B|A)}$ (the probability of observing the new evidence) and the prior value $P_{(A)}$ (the probability of our hypothesis without any additional prior information). Furthermore, $P_{(B)}$ is called the marginal likelihood, which is the total probability of observing the evidence, see (2). In this way, the RSSI histogram tables (or AP tables) generated in previous subsection are used to calculate the $P_{(B|A)}/P_{(B)}$ relationship. So, by a series of iterations, the prior value $P_{(A)}$ is updated with the current estimated posterior value $P_{(A|B)}$ at the end of each iteration.

$$P_{(A|B)} = \frac{P_{(B|A)}P_{(A)}}{P_{(B)}} \tag{1}$$

$$P_{(B)} = \sum_{Y} P_{(B|A)}P_{(A)} \tag{2}$$

The implementation of the Bayes-based estimator is shown in the Algorithm 1. This takes the AP tables and current WiFi measure as inputs, and return the estimated region and its probability. In this way, the algorithm start with a uniform prior probability distribution (lines 6 and 7) and recursively calculate the probability of the posterior region based on AP tables (line 11), then the probability (line 12) and predicted region (line 13) are calculated. After evaluating over all regions, the highest probability and its corresponding region are calculated (lines 16 and 17). Finally, based on the previous result, prior region is updated with the estimated posterior region (line 21) in order to be used in the next iteration.

The implementation of the Algorithm 1 is implemented in Python language for evaluation of performance of the Bayes estimator, and implemented in Java language for real-time inference on an Android mobile device.

**Algorithm 1** Bayes Algorithm

1: $N = number\ of\ regions$
2: $R = number\ of\ routers$
3: $W_r = AP\ table\ for\ router\ r$
4: **procedure** BAYESESTIMATOR($w_1, w_2, ..., w_R$)
5:     # Starting with uniform distribution
6:     $priorW_{1,2,..,R} = [1/N; 1/N; ...; 1/N]_{Nx1}$
7:     $probability = (100/N)\%$
8:     **while** probability $< 95\%$ **do**
9:         # Performing Bayes
10:         **for** r from 1 to R **do**
11:             $posteriorW_r = norm(priorW_r \times W_r[:, w_r])$
12:             $prob_r = max(posteriorW_r)$
13:             $pred_r = where(posteriorW_r == prob_r)$
14:         **end for**
15:         # Finding the highest probability
16:         $probability = max(prob_{1,2,..,R})$
17:         $r\_best = where(prob_{1,2,..,R} == probability)$
18:         $prediction = pred_{r\_best}$
19:         # Assigning the new prior
20:         **for** r from 1 to R **do**
21:             $priorW_r = posteriorW_{r\_best}$
22:         **end for**
23:     **end while**
24:     Return $prediction, probability$
25: **end procedure**

## III. Experimental Results

To evaluate the performance of the indoor localization system, the Bayes estimator algorithm is carried out in the testing dataset. The results were summarized in a confusion matrix, which is a specific table layout that allows us to easily visualize the performance of our prediction system, see Fig. 7. In this matrix, each column represents the instances in a predicted region while each row represents the instances in an actual region. In addition to confusion matrix, the overall accuracy of our Bayes prediction algorithm is 92.31%.

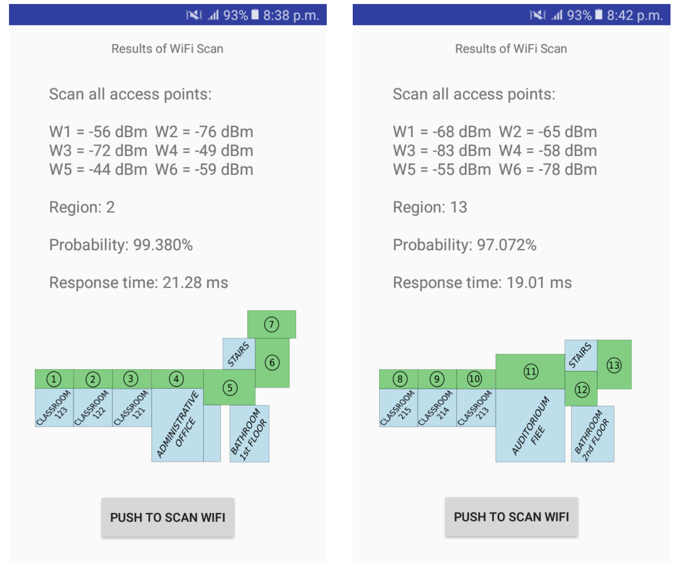
| | R01 | R02 | R03 | R04 | R05 | R06 | R07 | R08 | R09 | R10 | R11 | R12 | R13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R01 | 0.97 | 0 | 0 | 0 | 0 | 0.01 | 0.01 | 0 | 0 | 0.01 | 0 | 0 | 0 |
| R02 | 0 | 0.94 | 0 | 0 | 0 | 0.01 | 0 | 0.02 | 0 | 0 | 0.03 | 0 | 0 |
| R03 | 0.01 | 0 | 0.94 | 0 | 0.01 | 0 | 0 | 0.01 | 0.02 | 0 | 0.01 | 0 | 0 |
| R04 | 0 | 0 | 0 | 0.91 | 0 | 0.04 | 0 | 0 | 0.02 | 0.01 | 0 | 0.02 | 0 |
| R05 | 0 | 0.04 | 0 | 0.03 | 0.89 | 0 | 0 | 0.02 | 0 | 0.01 | 0 | 0.01 | 0 |
| R06 | 0 | 0 | 0.05 | 0 | 0 | 0.9 | 0 | 0.01 | 0 | 0.02 | 0 | 0.02 | 0 |
| R07 | 0 | 0 | 0 | 0.02 | 0 | 0 | 0.87 | 0 | 0 | 0.05 | 0 | 0 | 0.06 |
| R08 | 0.01 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0.96 | 0 | 0.02 | 0 | 0 | 0 |
| R09 | 0 | 0 | 0.04 | 0 | 0 | 0 | 0.02 | 0 | 0.9 | 0 | 0.04 | 0 | 0 |
| R10 | 0 | 0.02 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0.97 | 0 | 0 | 0 |
| R11 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0.02 | 0.01 | 0 | 0 | 0.96 | 0 | 0 |
| R12 | 0.01 | 0 | 0 | 0.03 | 0 | 0 | 0 | 0.03 | 0 | 0 | 0 | 0.92 | 0.01 |
| R13 | 0.01 | 0 | 0.02 | 0 | 0 | 0.02 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0.94 |

Fig. 7: Confusion matrix

The application for indoor localization was tested on a Samsung J2 Android device (with Android version 6.0.1). Then, we tested the app in different positions at different regions, and verified if the developed localization system works properly. As expected, we verified that the system correctly predicted the locations for almost all positions inside different regions. Some results are shown in Fig. 8. As we can see, the system correctly predicts the locations with a probability above 97% for tests conducted in different regions.

## IV. Conclusion

In this paper, we introduced the implementation of a WiFi-based indoor location system to guide persons at the Faculty of Electrical, Electronic and Telecomunications at the Universidad Nacional de Ingeniería, Peru. In order to achieve a high position accuracy with a fast response time and using low computational resources, we make use of a simple but effective Bayes estimator. In fact, experiments show promising results, obtaining an accuracy of 92.31% and a response time of around 20 milliseconds. In addition to the results, the proposed methodology could be used for similar localization tasks such as asset tracking, immersive experiences, augmented reality, controlling robots and more.




(a) Result at region 2 (b) Result at region 13

Fig. 8: Online results on an Android device at first and second floor

## References

[1] H. Motte, J. Wyffels, L. De Strycker, J.-P. Goemaere, "Evaluating GPS Data in Indoor Environments," Advances in Electrical and Computer Engineering, vol.11, no.3, pp.25-28, 2011, doi:10.4316/AECE.2011.03004

[2] W. Sakpere, M. Adeyeye Oshin, and N. B. Mlitwa, A State-of-the-Art Survey of Indoor Positioning and Navigation Systems and Technologies, South African Comput. J., vol. 29, no. 3, pp. 145197, 2017.

[3] F. Zafari, A. Gkelias, K. Leung: A Survey of Indoor Localization Systems and Technologies, arXiv:1709.01015v3, 2019.

[4] R. F. Brena, J. P. Garca-Vzquez, et al., "Evolution of indoor positioning technologies: A survey," Journal of Sensors, Volume 2017 (2017), 21 pages.

[5] A. Dhital, P. Closas and C. Fernndez-Prades, "Bayesian filters for indoor localization using wireless sensor networks," 2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), Noordwijk, 2010, pp. 1-7.

[6] M. V. Moreno-Cano, M. A. Zamora-Izquierdo, Jos Santa, and Antonio F. Skarmeta. 2013. An indoor localization system based on artificial neural networks and particle filters applied to intelligent buildings. Neurocomput. 122 (December 2013), 116-125.

[7] X. Li, Y. Cao and C. Chen, "Machine Learning Based High Accuracy Indoor Visible Light Location Algorithm," 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), Xi'an, 2018, pp. 198-203. doi: 10.1109/SmartIoT.2018.00043

[8] Indoor Google Maps. (Online Web, accessed on January 2014) Available: http://maps.google.com/help/maps/indoormaps/.

[9] L. Chruszczyk, Statistical Analysis of Indoor RSSI Read-outs for 433 MHz, 868 MHz, 2.4 GHz and 5 GHz ISM Bands, International Journal of Electronics and Telecommunications, vol. 63, no. 1, pp. 3338, 2017.

[10] K. Kaji and N. Kawaguchi, "Design and Implementation of WiFi Indoor Localization based on Gaussian Mixture Model and Particle Filter," 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, NSW, 2012, pp. 1-9.

[11] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. 2010. Growing an organic indoor location system. In Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10). ACM, New York, NY, USA, 271-284.