WiFi RSS Fingerprinting Indoor Localization for Mobile Devices

Wondimu K. Zegeye *, Seifemichael B. Amsalu †, Yacob Astatke*, Farzad Moazzami*
*Department of Electrical and Computer Engineering
Morgan State University, Baltimore, MD 21251, USA
Email: {wozeg1,yacob.astatke,farzad.moazzami}@morgan.edu
†Department of Electrical and Computer Engineering
N.Carolina A&T State University, Greensboro, NC 27411 USA
Email: samsalu@aggies.ncat.edu

***Abstract*—While WiFi-based indoor localization is attractive, the need for a significant degree of pre-deployment effort is a key challenge. In this paper, indoor localization with no pre-deployment effort in an indoor space, such as an office building corridor, with WiFi coverage but no a priori knowledge of the placement of the access points(APs) is implemented for mobile devices. WiFi Received Signal Strength(RSS) in the considered environment is used to build radio maps using WiFi fingerprinting approach. An offline RSS fingerprint of the environment is compared with an online RSS measurement to estimate the location of a user. Two architectures are developed based on this localization algorithm. The first one involves a client-server approach where the localization algorithm runs on the server whereas the second one is a standalone architecture and the algorithm runs on the SD card of the mobile device. Experimental results in the considered environment validate the approach for both architectures.**

***Keywords*-RSS;Android;WiFi Fingerprinting;Indoor Localization;**

I. Introduction

Location-based services (LBS) require on-the-fly localization of the user to provide their services and related information. Indoor localization applications, such as indoor navigation, require precise user location. Since positioning services such as GPS remain inefficient for indoor use, other sensor techniques should be considered [1]. Because of the advancement in WiFi technologies and the increasing capability of mobile devices, locating a user in an indoor environment became a good starting point for providing indoor LBS.

Localization/positioning techniques which use WiFi signal strength measurements are mainly categorized into two. The first one involves mapping of labeled data gathered offline before the system operation. This mainly involves building radio map of the environment based on WiFi RSS fingerprinting. The second one relies on mathematical modeling methods to determine the device's location. As in the case of [2] [3], trilateration is used to determine the location of a user by measuring its distance from multiple reference points such as Access Points (APs) positions.

Indoor localization techniques which are based on mapping of labeled data [4][5] exploit RSS to determine location by a reverse function which uses RSS fingerprinting or radio maps resulting in coarse measurements. As a result, these methods involve a calibration step where labeled data are collected before the system operation.

Mathematical modeling methods determine a device's location by measuring distance or angle information between nodes such as APs and the mobile device in the localization process. These methods are based on range location (Range-based). The information of range or angle between APs and the mobile device is obtained. Range is the premise of location; precise range is the assurance of accurate location. Only a certain number of reference APs are combined with effective localization algorithm if they ensured feasible location task, lots of equipment is included and arranged in the process. At the same time, in the case of the receiving mobile device, information of signal strength and angle between APs is needed to measure and convert it into distance information. With the help of geometric or mathematical relationships, it is possible to carry out the location task after achieving information of distance or orientation between APs. Typical indoor localization algorithms, the methods based on Range, mainly include trilateration or triangulation and maximum likelihood techniques [6]. The common ranging methods are as follows: Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA) and RSSI ranging method.

While WiFi-based indoor positioning is attractive, the need for a significant effort in pre-deployment of an infrastructure is a key challenge. A 2014 Microsoft Indoor Localization Competition explores and compares a diverse set of technical approaches to indoor Localization [7]. The approaches presented by the competitors are mainly categorized into two: Infrastructure-free and Infrastructure-based. Infrastructure-free approaches have focused on leveraging already existing WiFi, FM, TV, GSM, geo-magnetic and sound signals to enable indoor localization through detailed fingerprinting. Infrastructure-based approaches rely on the deployment of

978-1-5090-1496-5/16/$31.00 © 2016 IEEE

customized RF-beacons such as RFID, infrared, ultrasound, Bluetooth, Short-range FM transmitters, lights, and magnetic signal modulators to enable accurate position estimation.

RADAR [8] which is the state-of-the-art RF wireless networks localization system is based on empirical signal strength measurements as well as a simple yet effective signal propagation model. Data collection considers user's location (x,y), and record of the direction (d) (one of north, south, east, or west) that the user is facing at the time the measurement is made. In addition, the the mobile host records tuples of the form (t,x,y,d) during the off-line phase where t is the time stamp. WiFi RSS measurements are associated with the 3 specific APs used to cover the testing area. The localization algorithm of this paper is simpler and the APs used for localization are selected among the many reachable APs based on visual inspection of the radio maps.

In this paper, an indoor localization system which makes use of an already deployed infrastructure in the considered environment, such as an office building corridor, with WiFi coverage but no a priori knowledge of the placement of the Access Points (APs) is implemented for Android OS based mobile devices. Any user carrying WiFi-enabled Android OS based devices such as Smart phones traverse this environment in a normal course. Once the RSS is sampled in the area under consideration, the estimated location of the user is computed by a reverse function as an optimization problem which makes use of this sampled RSS values. The approach implemented in this paper depends on a grid-based representation of the considered area.

This paper implements two different WiFi based architectures to locate a person carrying a mobile device in an indoor space. The first architecture is client-server whereas the second one is a standalone architecture.

II. WiFi RSS Fingerprinting Indoor Localization Algorithm

The basic idea behind the localization algorithm is building radio maps of the RSS in the considered indoor environment and setting the problem of localization as an optimization problem: given RSS value measurements on-the-fly of an unknown location, the function is reversed to calculate the estimated location.

This approach does not depend on specific features or beacons. It uses a grid-based [9][10] representation of the indoor environment. In the offline phase of this approach, radio maps of the considered area are built from the sampled RSS values. Then, the estimated location of the device is computed by a reverse function which solves the optimization problem, the online phase (estimation phase).

A significant number of WiFi based localization techniques [5] [11] use Neural Network (NN) to estimate location using RSS measurements. The NN results in an approximation of the radio environment and behaves like a black-box, thus it is difficult to understand and find a mechanism of improving its performance.

Alternatively, RSS sampling can be used as an input for a mathematical modeling of the RF fields. Starting from this model approximation, a function that estimates a device's location can be formulated. This approach provides a foundation for strong improvements and extensions (e.g. weighted APs, non-linear weighting of RSS measurements, mutual positioning of users, etc.)which are not easily allowed by an NN .

The first step (offline) of the algorithm involves spatial sampling of RSS values in the testing corridor by using the WiFi fingerprinting Android application developed. The application scans and collects the information including the RSS value associated with the reachable APs at each sampling location. The set of APs is defined as:

$$A = \{A_k\} \quad k = 1, ..., |A| \tag{1}$$

where $|A|$ is the cardinality of the set of APs. Radio maps showing the distribution of the RSS are then built for each AP. The choice of APs used for positioning is based on the signal coverage and the value of $\Delta = RSS_{max} - RSS_{min}$. The higher the value of Δ, the better the discrimination as can be seen in Figure 3.

The set of radio maps built for each of the APs are obtained by interpolating the data coming from the sampling procedure (offline phase).

We can visually inspect the radio maps to understand the placement of the APs and how the sampled RSS values vary in our measurement area. The interpolation of the RSS values to built these radio maps uses fitting surfaces computed with a triangle-based interpolation and these surfaces are represented by

$$R = \{R_k\} \quad k = 1, ..., |R| \tag{2}$$

with the constraint that $|R| = |A|$.

To make sense of the interpolation, spatial sampling interval should satisfy Nyquist criterion. As RSS value variations at grid points separated by a distance smaller than 1m are not meaningful or comparable to signal fluctuations. As a result, distance separations greater than 1m among grid points for RSS measurement are used for interpolations to make sense.

Each R_k represents a grid map with indices by i-rows and j-columns, where i = $1, ..., I$ and $j = 1, ..., J$.

Once we are done with the fingerprinting (offline) phase, we move to the localization algorithm where rss_k is the RSS value measured from the k^{th} AP in the unknown location. We define and use the following matrix notation to simplify the computation:

$$\hat{R}_k = [\hat{r}^{i,j}] \tag{3}$$

where

$$\hat{r}^{i,j} = rss_k \quad \forall i, j \tag{4}$$

For each (i, j)-th point of the map, the Euclidean distance among the values in the $|R|$ radio maps and the corresponding measured values $\hat{R}_k$ is defined as:

$$D_{i,j} = \sqrt{\sum_{k=1}^{|A|} (R_k - \hat{R}_k)^2} \tag{5}$$

Where $\hat{R}_k$, $k = 1, ..., |R|$ are the grid maps.

From the resulting matrix $D_{i,j}$, the minimum value represents the approximated user location estimate:

$$[\hat{x}, \hat{y}] = \arg_{i,j} \min D_{i,j} \tag{6}$$

III. Experimental Setup

The experiments were performed in an indoor environment, which is some portion of the test building ground floor corridor.
Figure 1 shows the layout of the ground floor corridor where the experiments were performed. For the particular experiment, neither additional APs were deployed nor existing APs were moved. Thus, the experiments were performed using the already existing WiFi infrastructure of the building. The measurements were performed during working hours with people walking around and the WiFi network being used.

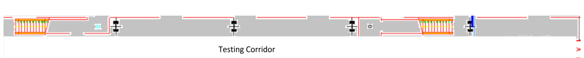


Fig. 1: Layout of the testing corridor

A. Data Collection (WiFi RSS Fingerprinting)

A corridor of dimension 72x2.4m as shown in Figure 1 is used as a testing area. It is divided into 3x31 grid points. Using the WiFi Fingerprinting Android application developed, the WiFi RSS values are collected at each of the grid points. This training data is stored in `XML` format shown in Listing 1 which consists of the very important parameters for localization such as the pixel values of the collection points (grids labeled with IDs from 0 to 92) which corresponds to the relative physical locations and the RSS values corresponding to each of the reachable APs (Table I) at collection points. Radio maps are generated as shown in Figure 2. The Fingerprint parsing utility (MATLAB code) in Figure 2, parses the raw `XML` in Listing 1 and associates `MAC` of an AP and its `RSS` value for the 93 grid points. This process is repeated for all of the 10 reachable APs to build their corresponding radio maps.

Listing 1: *Fingerprinting Raw XML Scan*

```
<?xml version='1.0' encoding='UTF-8'
   standalone='yes' ?>
 <data><grid id="0"><X>364.0</X><Y>95.0</
    Y><Z>0.0</Z>
  <wifi>
   <rssi mac_ap="00:13:46:1d:5b:a0" ssid
      ="AP_SSID" channel="6" encryption="
      on">-45</rssi>
        .....
  </wifi>
  </grid>
   <grid id="1"><X>1271</X><Y>540</Y><Z
      >0.0</Z>
    <wifi>
         <rssi mac_ap="00:13:46:1d:5b:a0"
             ssid="AP2_SSID" channel="6"
            encryption="on">-46</rssi>
              ....
         </wifi>
  </grid>
.......................
</data>
```

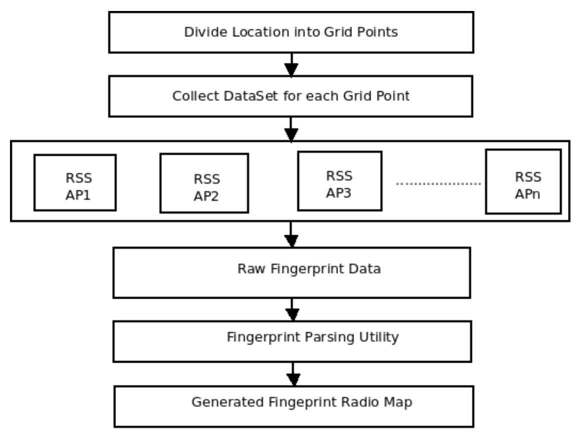


Fig. 2: WiFi Fingerprint Flowchart

B. System Requirements

The initial system is aimed toward Android users due to its open source system development and large user base. Both smart-phones and tablets with an Android OS can install the application that allows scanning of WiFi RSS data on the fly for user's location computation based on the radio maps which are already saved on the SD card.

During the development of this work, we used several devices such as the remote laptop computer (Intel(R) Core(TM) i3 CPU M370 @2.40GHz, RAM 4.0 GB, Speed 1066.0 MHz) which is used as a server for the client-server architecture. The server is accessible remotely via a TCP socket once the client (the tablet computer) is connected to one of the APs.

MAC_AP	Delta(dBm)	RSS_Min(dBm)	RSS_Max(dBm)
00:1a:30:b8:01:80	18	-85	-67
00:13:46:1d:5b:a0	30	-83	-53
00:14:7c:ae:79:62	26	-83	-57
00:14:7c:b9:75:6a	21	-81	-56
00:1a:30:b8:01:80	23	-93	-70
00:11:20:06:a0:e0	36	-78	-42
00:11:20:68:b1:b0	31	-71	-40
00:11:20:68:b2:50	41	-81	-40
00:14:7c:af:52:72	40	-91	-51
00:22:3f:19:20:0c	37	-88	-51

TABLE I: Reachable APs

MAC_AP	Delta(dBm)	RSS_Min(dBm)	RSS_Max(dBm)
00:11:20:06:a0:e0	36	-78	-42
00:11:20:68:b1:b0	31	-71	-40
00:11:20:68:b2:50	41	-81	-40
00:14:7c:af:52:72	40	-91	-51
00:22:3f:19:20:0c	37	-88	-51

TABLE II: 5 APs selected for Localization

And our Android-based mobile device is a Samsung Galaxy Tab 10.1 which is not only used as a client for the client-server architecture but also for the standalone architecture. The WiFi fingerprinting application developed for data set (training data) collection also runs on the tablet.

C. Radio Maps

The parsed fingerprinting data provides a coarse RSS measurement for all of the reachable APs. This original measurement has 3x31 grid points representing 93 Cartesian coordinate points of our `jpg` file of the layout of the corridor. Using a triangle-based interpolation, this coarse measurement is interpolated over the rest of the Cartesian points of the `jpg` file which do not correspond to any measurement points. This interpolation generates radio maps as `CSV` file of 7x138 RSS values as shown in Figure 5 which is some portion of AP4's interpolated RSS values. The (x,y) values on Figure 3 represent the pixel values of the actual `jpg` image of the layout of the testing area labeled with the corresponding interpolated RSS values.

The interpolation technique employed in this paper is entirely based on the RSS values collected at the measurement points. It does not take into consideration the radio propagation behavior of an indoor environment unlike RADAR [8]. This makes it to be flexible and it can be used in any environment. Especially in indoor environments where signal reflections makes it difficult to model the environment using radio propagation behavior.

After interpolation, the heat maps generated for each of the reachable APs were visually inspected and 5 APs which provide a better cumulative coverage of the testing area are selected. Figure 3 shows the interpolated radio maps corresponding to the 5 APs selected for location estimation during testing. These selected APs together with their Δ values are summarized in Table II.

D. Location Estimation

The Location estimation procedure for both architectures that we implemented, client-server and the standalone, follows the flow chart of Figure 4. It involves applying the localization algorithm on the radio maps of the 5 APs selected for the localization which are `CSV` files of the interpolated RSS samples and the received RSS values on-the-fly.
For the client-server architecture, the radio maps are saved at the location of the server program which is implemented in `C++` where matrix operations makes use of the `GNU GSL` matrix library for fast computations. When a user requests for location estimation, an `RSS` measurement collected on-the-fly will be sent from the *client Android Application* of the Android based device via a socket to the server as an `XML` file as shown in Listing 2. The received `XML` file will be parsed at the server and location estimation is performed at the sever by running the localization algorithm. The estimated location value will be sent as an `XML` file consisting of the estimated locations (Cartesian coordinates) to the client Android Application which in turn will be parsed by it and the estimated location will be displayed for the user by the same application.

Listing 2: *RSS Online Scan*

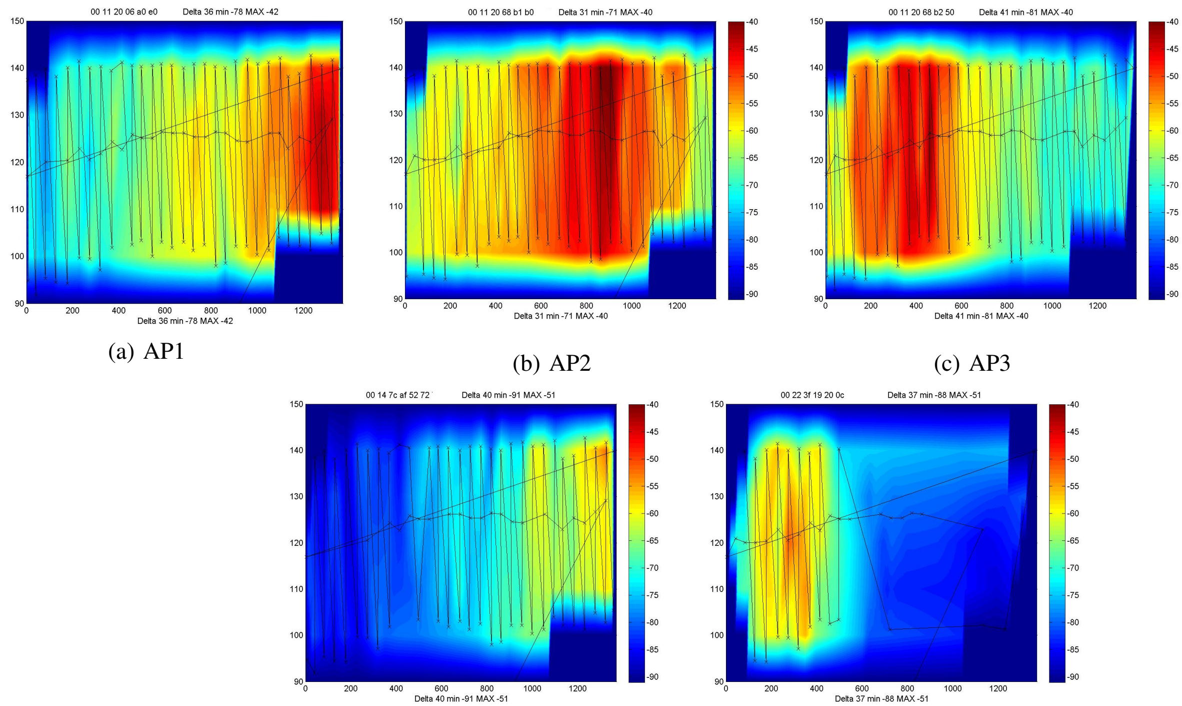
```
<?xml version='1.0' encoding='UTF-8'
   standalone='yes' ?>
  <wifi>
   <AP>00:13:46:1d:5b:a0 -45</AP>
   <AP>00:11:20:06:a0:e1 -45</AP>
        ....
 </wifi>
```

In case of the standalone architecture, the localization algorithm is implemented using Java Matrix Package (`JAMA`) for matrix computations integrated with the *localization Android application*. In other words, one Android application is developed which makes use of the `JAMA` library for matrix manipulation of the radio maps and online scanning of `RSS` values at testing points. A user of the standalone architecture must have the radio maps saved on the SD card of the device in order to use the application. Thus, when a user looks for his/her position, online measurement is collected as in Listing 2 and parsed locally on the same device by the same application. Using the localization algorithm, location estimation is computed by loading the `CSV` radio maps already saved on the SD card of the device. Once the location is estimated, it is returned by the same application. The response time of this algorithm in the standalone architecture is about 220ms.

IV. RESULTS AND ANALYSIS

A. First Experiment (Initial Result)

The first test involves portion of the same corridor of dimension 72x2.4m as shown in Figure 1. It is divided into 3x31 grid points and WiFi RSS for each of the points was recorded



(a) AP1 (b) AP2 (c) AP3

(d) AP4 (e) AP5

Fig. 3: Radio Map of 5 APs used for Positioning

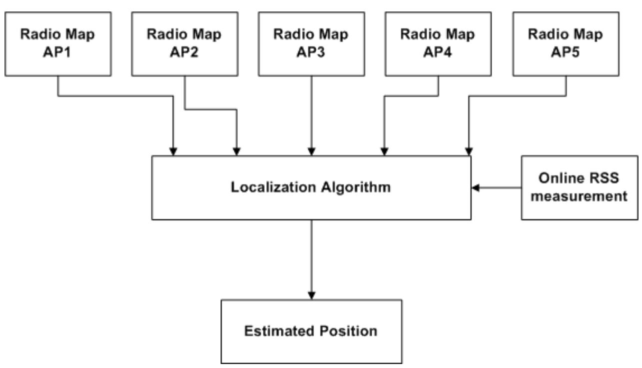


Fig. 4: Location Estimation Procedure

DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH
-99	-99	-99	-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99	-99	-99	-99
-62.021	-60.899	-59.777	-58.808	-58.356	-58.734	-59.292	-59.924	-99	-99
-61.143	-60.106	-59.655	-59.203	-58.752	-59.093	-60.106	-61.616	-99	-99
-59.722	-59.477	-59.231	-58.986	-58.671	-60.288	-61.798	-63.307	-99	-99
-56.163	-55.26	-54.335	-53.411	-52.486	-54.493	-58.738	-62.984	-99	-99
-99	-99	-99	-99	-99	-99	-99	-99	-99	-99

Fig. 5: AP4 interpolated RSS CSV file (radio map) corresponding to Right side section of the corridor

using the fingerprint application developed. The interpolated CSV file (radio map) is a 7x138 matrix of RSS values. For the localization 5 APs were chosen of all the detected APs based on their coverage and the value of Δ. From Figure 5, we can see the right side section of the building interpolated CSV file (radio map) of AP4 of Figure 3.

From this experiment a mean $\mu = 7.46m$ and standard deviation $\delta = 5.18m$ of accuracy were achieved and the system can locate a device (user) under 10m 67% of the time over 42 realizations.

B. Second Experiment(Improved and Final Result)

The following improvement techniques are implemented.

- Discard the first 10 measurements.
- Insert 2 seconds of delay between each scan.
- Take the average RSS value of 5 measurements.

Discarding the first 10 measurements prevents from including measurements of previous scans. Adding a delay between scans provides a period of time for detecting broadcasts. Besides, the averaging of the RSS values over 5 measurements enables the applications to have a relatively stable RSS values. This is reasonable as RSS values are known to slightly vary on the same location of the considered environment for several different measurements.

From this experiment, a mean $\mu = 3.46m$ and standard deviation $\delta = 2.46m$ of accuracy were achieved and the system can locate a device (user) under 5m 80% of the time over 62 realizations. As a result, a 47% improvement in accuracy is achieved compared to the first test.

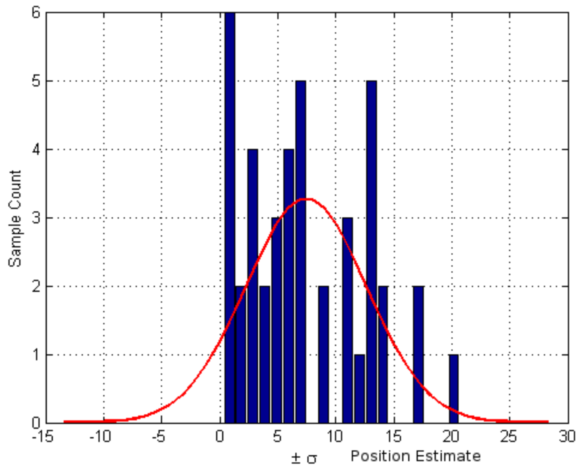


Fig. 6: Accuracy test corresponding to the first test

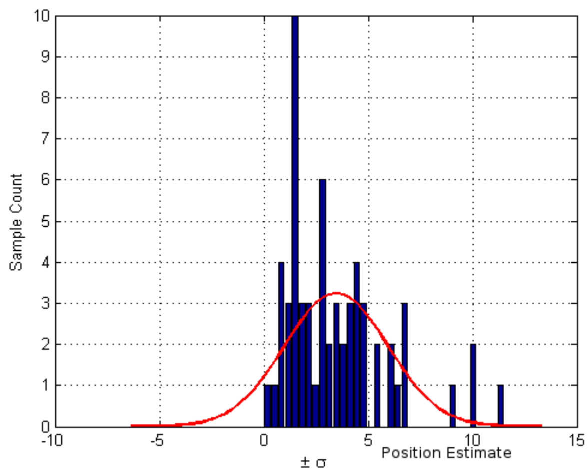


Fig. 7: Accuracy test after improvement

C. Accuracy Calculation

The localization error of the user can be defined as the distance between the ground-truth Cartesian location (x, y) and its Cartesian location estimation $(\hat{x}, \hat{y})$ as shown in Figure 8.
The average error $(\bar{e})$ over n number of realizations is computed as the average of e over n.

$$e = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \tag{7}$$

In Figure 8, the green label (square) represents the ground-truth Cartesian location and the red label (circle) its corresponding Cartesian location estimation for a particular measurement of 10 testing points.

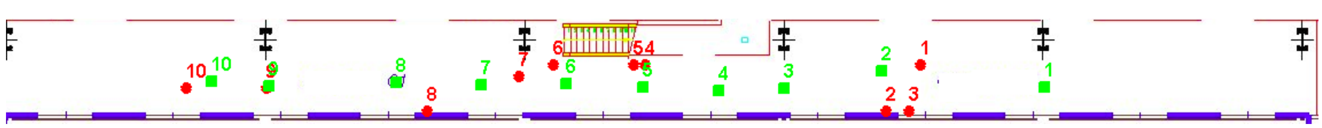


Fig. 8: Ground-truth vs Estimated-location

V. Conclusion

In this paper, RSS fingerprint based WiFi localization which uses an existing infrastructure of an indoor environment is examined. It involves building radio maps using RSS values measured offline and uses a reverse function for solving the localization problem as an optimization problem. The implementations and experiments presented in this paper can provide a foundation for the integration of other navigation techniques such as outdoor positioning and Inertial Navigation Systems (INS). Combining indoor positioning techniques with existing outdoor positioning systems can allow a full indoor/outdoor spatial coverage which can greatly enhance a user's experience. Future work will investigate the accuracy of the algorithm in a wider test area and its possibility of application in a multi-story building.

References

[1] Solomon Chan, Gunho Sohn, "Idoor Localization Using WI-FI Based Fingerprinting and Trilateration Techniques For LBS Applications",*International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXVIII-4/C26

[2] A.Smailagic and D.Kogan, "Location sensing and privacy in a context-aware computing environment," *Wireless Communications*, vol.9, no.5, pp.1017, 2002.

[3] Best PracticesLocation-Aware WLAN Design Considerations, retrived from http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/WiFiLBS-DG/wifich5.html

[4] R.Battiti,T.L. Nhatand A.Villani,"Location Aware Computing: A Neural Network Model for Determining Location in Wireless LANs," in *Technical Report # DIT-02-0083*, February 2002.

[5] U.Ahmad, A.Gavrilov, U.Nasir, M.Iqbal, S.J. Cho, and S.Lee, "In-building localization using neural networks," *in Engineering of Intelligent Systems, 2006 IEEE International Conference on* 2006, pp. 16.

[6] Robin Henniges ,"Current approches of Wifi Positioning", *SERVICE-CENTRIC NETWORKING-SEMINAR* WS2011/2012, TU-Berlin, 2012

[7] Dimitrios Lymberopoulos et.al. "A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned", *IPSN '15*, April 14-16, 2015, Seattle, WA, USA.

[8] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE INFOCOM*, pp. 775-784, 2000.

[9] Brevi, D., F. Filepppo, R. Scopigno, F. Abrate, B. Bona, S. Rosa, and F. Tibaldi, "Hybrid localization solutions for robotic logistic applications", *IEEE International Conference on Technologies for Practical Robot Applications (TePRA 2009)*, November 2009.

[10] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46-57, 1989.

[11] R. Battiti, A. Villani, R. Villani, and T. L. Nhat, "Neural network models for intelligent networks: Deriving the location from signal patterns," in *Proceedings* of AINS2002, UCLA, 2002.