Wi-Fi Localization Using RSSI Fingerprinting

Eduardo Navarro[#1], Benjamin Peuker[#2], Michael Quan[#3]

Advisors: Dr. Christopher Clark[#4], Dr. Jennifer Jipson[*5]

[#]*Computer Engineering,* [*]*Child Development*

California Polytechnic State University

1 Grand Avenue; San Luis Obispo, CA 93405; USA

[1]eonavarr@calpoly.edu, [2]bpeuker@calpoly.edu, [3]mquan@calpoly.edu

[4]cmclark@calpoly.edu, [5]jjipson@calpoly.edu

Abstract— **Wireless Local Area Networks using Wi-Fi is becoming more and more ubiquitous. As such, they provide a potential pre-built infrastructure for small area localization. This project serves as a proof of concept for a playground child tracking system to be deployed at Cal Poly's Child Development Playground Lab. The two main options for doing Wi-Fi localization are triangulation and fingerprinting. Triangulation involves mapping signal strength as a function of distance while fingerprinting creates a probability distribution of signal strengths at a given location and uses a map of these distributions to predict a location given signal strength samples. The triangulation method did not show promising results and the fingerprinting method had promising results with various ways of making predictions.**

***Keywords*— Wi-Fi, Localization, RSSI, Triangulation, Fingerprint, Markov, Nearest Neighbor**

I. Introduction

While in the process of working on a Computer Engineering Capstone Project, the client had an idea of tracking children on the Cal Poly Child Development Playground Lab and using the positional data to infer how they interacted with their environment and each other. This could provide a great research platform for Child Development. Additionally, the team's advisor, Dr. Chris Clark, expressed an interest in incorporated the positional data into behavioral models of Artificial Intelligence agents.

II. Background and problem statement

There are several existing technologies used for geo-location. The playground provided an environment where the choice of tracking method was not trivial. The following are some of the choices considered:

GPS is one of the most common tracking solutions in the world. Unfortunately, the Child Development Playground Lab is situated towards the core of the school's campus and is enclosed by walls. Additionally, part of the playground resides indoors and could not be accurately tracked by a GPS system. Because of these issues, the GPS solution was no longer considered as a viable option.

RFID was one of the first methods considered. Readers were to be placed at multiple Points-of-Interest (playground structures) to detect if a child was nearby. Passive RFID tags are cheap and used no power and scales really well to the number of children being tracked. However, one large drawback was the very short range (inches) of passive RFID. The granularity needed for tracking would require hundreds of points throughout the playground. Active RFID increases the range, but uncertainty is introduced and triangulation would be required to pinpoint the location of a child.

Wi-Fi localization using RSSI (Received Signal Strength Indicator) readings was also considered as a potential solution. This was ultimately determined to be the most realistic solution for the playground after researching the technique and acquiring ideas from several white papers that describe successful Wi-Fi localization implementations [1]. Specifically, two major types of localization were considered after initial research.

A) Triangulation

Wi-Fi triangulation's goal is to map RSSI as a function of distance. This method requires a steep linear characterization curve in order to be properly implemented. Functions describing these curves are then used with live RSSI values as input to generate an (x,y) location prediction. This method was considered first due to its relatively simple implementation.

B) Fingerprinting

Wi-Fi Fingerprinting creates a radio map of a given area based on the RSSI data from several access points and generates a probability distribution of RSSI values for a given (x,y) location. Live RSSI values are then compared to the fingerprint to find the closest match and generate a predicted (x,y) location.

Due to the playground location and the potential to use existing Wi-Fi infrastructure, Wi-Fi localization was ultimately determined to best suit our needs.

III. Experimental setup

To perform Wi-Fi localization, a node and several receivers are needed. Any Wi-Fi enabled device with ad-hoc capability can be used for the node. For our tests, a Dell Mini 1012 Netbook was used. For receivers, Linksys WRT54GL

routers with the DD-WRT custom firmware were used. Specifically, DD-WRT was used because it provided straightforward access to RSSI information.

A. *Lab Test Setup*

Initial tests were done to characterize RSSI as a function of distance from a router (for use with triangulation) and 1-dimensional fingerprinting.

1) *RSSI CHARACTERIZATION*

The goal of this test was to obtain a characterization curve of RSSI as a function of distance. A router was placed on a table and a team member carried a netbook and walked away from the router. At 1 meter increments, RSSI and noise values were manually collected from the router's web interface.

2) *One-Dimensional Wi-Fi Fingerprinting*

The goal of this test was to obtain the fingerprint of reference points on a line. Two routers were placed 8 meters apart. Reference points were placed 1 meter apart between the two routers. The fingerprint was then used to predict the location of the node given live RSSI readings.

B. *Playground Setup*

The RSSI characterization from the lab test showed that RSSI was not usable for triangulation; as such, Wi-Fi fingerprinting was used for the actual implementation.

The following flowcharts outline the Wi-Fi fingerprinting and localization process.

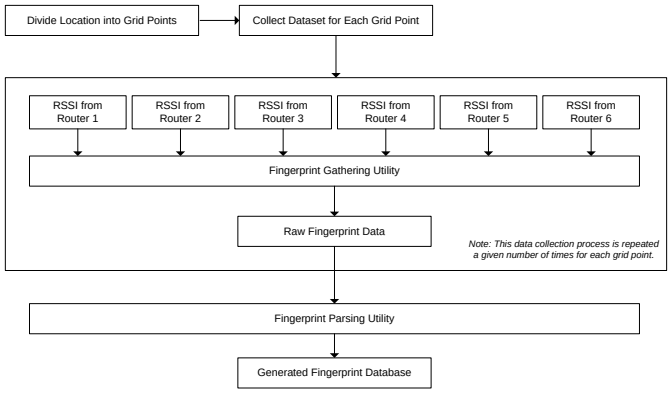

Fig. 4 – Fingerprint Flow Chart

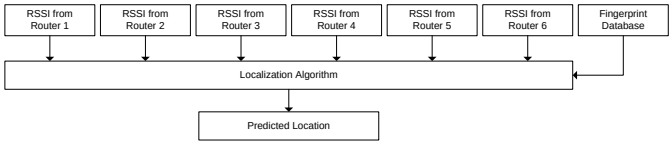

Fig. 5 – Prediction Flow Chart

Various components were needed to fingerprint the playground efficiently. After obtaining a blueprint of the playground [2], the following information was determined. (Note that due to the units used in the blueprint, subsequent tests used feet instead of meters)

1) *Router Placement and Picking (x,y) Coordinates*

Fig 1 shows the placement of the routers and reference points. 10 foot increments were chosen for (x,y) coordinates starting with the location (5,5) and incrementing in both directions until the entire playground is covered.

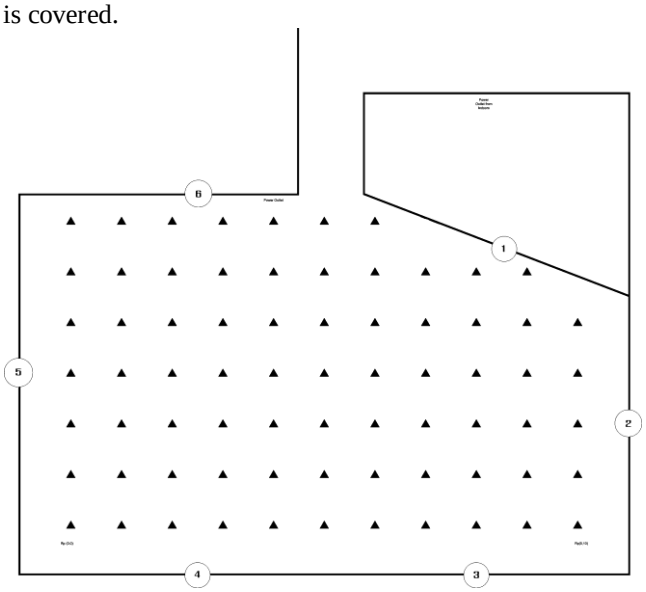

Fig. 1 – Router and fingerprint reference points. Circles represent routers and triangles represent reference points.

Fig. 2 – Sample router placement on playground

In addition, two primary software utilities were developed in order to expedite the collection and generation of fingerprint data.

1) *Fingerprint Gathering Utility*

Manually collecting data to fingerprint the playground was considered to be infeasible due to the

large datasets required to create a valid fingerprint. A Fingerprint Gathering Utility with a GUI was thus developed in C# in order to facilitate data collection at each reference point.

This utility looked at the HTTP status pages generated by each router in order to collect data. Specifically, the router status pages contain the live RSSI data needed to create a fingerprint at each location.

To gather data with this utility, the user first specifies a series of router status page URLs and router authentication details in an included configuration file (urls.ini).

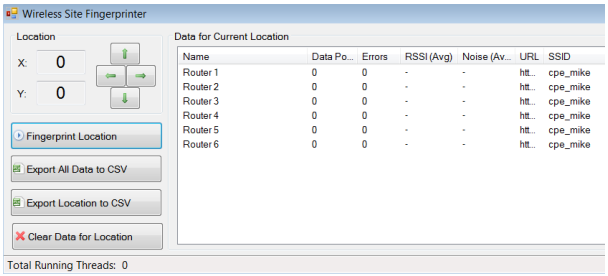

Fig. 3 – Fingerprinting Tool

2) Parser and Fingerprint Map Generator

The parser takes the raw RSSI readings as input and generates the data necessary to build the RSSI probability distribution for each reference point.

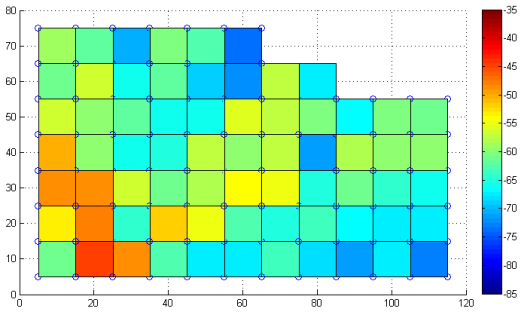

Fig. 4 – Overhead Playground Fingerprint Map for Router 4. Color map indicates RSSI.

C. Prediction Methods

There are two prediction methods used to determine the location of the node based on the fingerprint data.

1) Nearest Neighbor

The nearest neighbor method simply calculates the Euclidean distances between the live RSSI reading and each reference point fingerprint. The minimum Euclidean distance is the Nearest Neighbor and the likely (x,y) location.

Euclidean Distance:

$$\sqrt{\sum_{i=1}^{6}(R_i - FP_i)^2}$$

Two versions of Nearest Neighbor are used: unconstrained search-space and constrained search-space. Unconstrained search-space looks at the entire fingerprint map to find the closest match. Constrained search-space only searches within a given distance from a previously predicted location. The idea is that a moving object can only travel up to a maximum distance from its previous location within the time it takes to collect a live RSSI reading and searching through the entire map is unnecessary. This also has the effect of ignoring predicted locations that are closer based on Euclidean distance but physically impossible as the next location based on the previous location.

2) Markov Localization

Markov Localization makes use of the statistical data of the fingerprint to guess the most likely position. This is done in two steps: Prediction and Correction.

Prediction Step:

$$p(L_t) = \sum_{L_{t-1}} p(L_t \mid L_{t-1}) p(L_{t-1})$$

$p(L_t)$ is the probability of being at location L at time t. $p(L_t \mid L_{t-1})$ is the probability of being at location L at time t given the previous location L at time t-1. This in effect constrains the search-space to the most likely region based on what is physically possible based on what we know about the object's motion.

Correction Step:

$$p(L_t \mid R[]) = p(R[] \mid L_t) p(L_t) * N$$

$p(L_t \mid R[])$ is the probability of being at location L at time t given the RSSI values R[] we received at time t. $p(R[] \mid L_t)$ is the probability of having RSSI values R[] given a location (the probability density function generated from the fingerprint data) and $p(L_t)$ is the probability of being at that location (from prediction step). N is a normalization factor.

IV. Results

A. Lab Test Results

The lab test results ultimately determined which method we used for the full implementation.

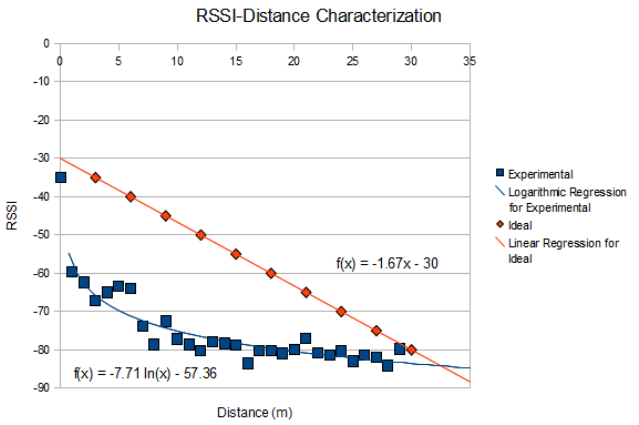

Fig. 5 – RSSI-Distance Characterization

The RSSI characteristic (Fig.5) show an almost constant curve outside the 10 meter mark. The result also shows a very large variation in RSSI readings for a given distance relative to the change in RSSI as we move away from the router. This means that an RSSI for a given distance can fluctuate more than the difference in RSSI between two distances! From this we concluded that Wi-Fi triangulation will not work for the purposes of this project.

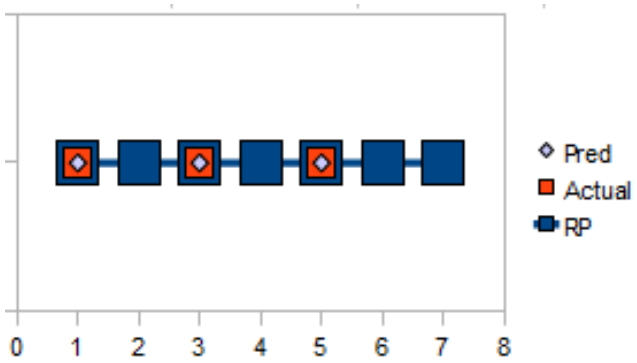

Fig. 6 – 1-D Fingerprint Test

The fingerprinting test showed more promising result than the RSSI characterization (the basis for the triangulation method) as can be seen in Fig.6. As such, the fingerprinting method for localization was pursued further. Additional tests were then performed using this method with the full number of routers (6). These tests yielded similarly promising results so it was decided that fingerprinting would be the method used to implement Wi-Fi Localization.

A simulated random walk was made using sample (x,y) locations from the collected RSSI readings for fingerprinting. This represents a test run on the same day as fingerprinting. The data was then plugged into the various prediction methods in an attempt to rebuild the path. The results shown in Figs. 8, 9, 11 and 12 show the predicted path overlaid on the actual path. Another random walk was performed 1 week later to test how much the fingerprint data changed over time.

The results from the second run were very poor, showing that the fingerprint drifted over the course of 1 week.

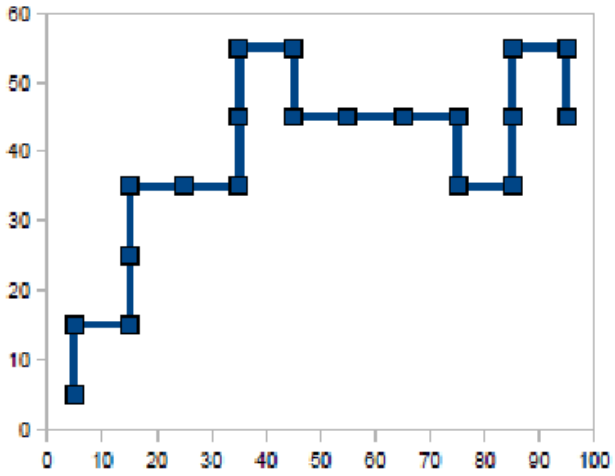

Fig. 7 – Actual Path.

1) Nearest Neighbor

Nearest Neighbor worked surprisingly well in our test. In the unconstrained version, the prediction got "lost" (predicted location >20 ft away) a few times but quickly recovered since the entire map was searched. The constrained version (within 10 ft) was slightly more accurate. However, if it does get lost, there is little chance of recovery since it will only consider the best match within that local region, none of which would be correct.

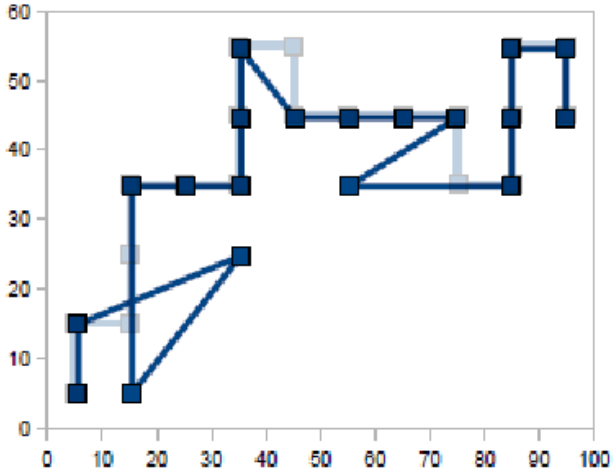

Fig. 8 – Unconstrained Nearest Neighbor.
(Darker predicted path overlaid on lighter actual path)

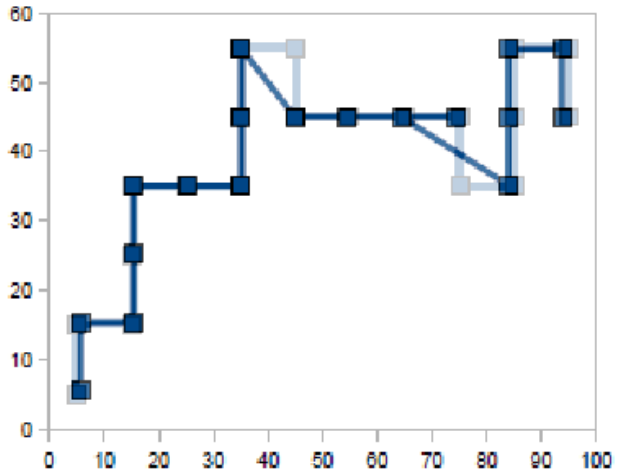

Fig. 9 – Nearest Neighbor constrained within 10 ft.

2) Markov Localization

For the Markov Localization tests, it is assumed that there is equal probability for the object to move to any location within a region of a given distance from the previously predicted location. A Gaussian distribution was used for the probability density function.

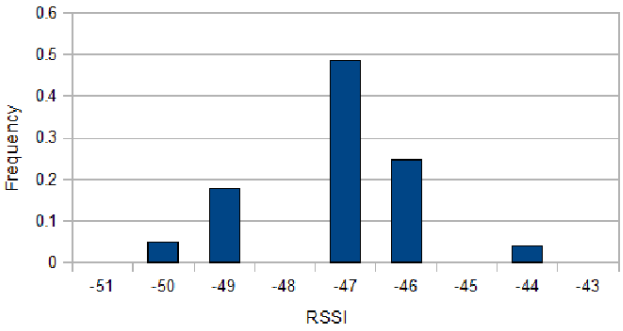

Fig. 10 – Sample Router Histogram.

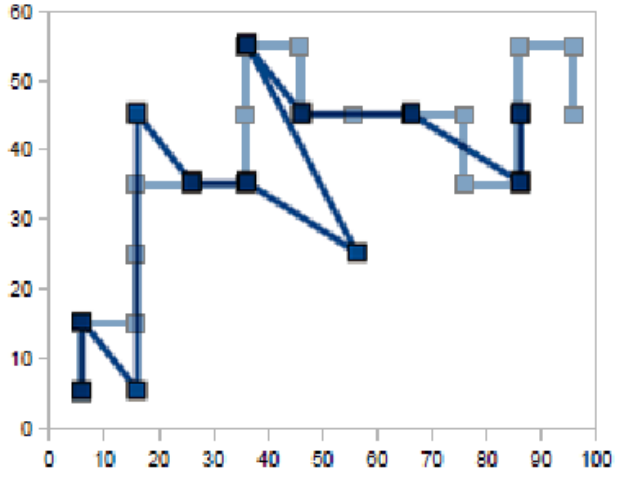

Fig. 11 – Unconstrained Markov

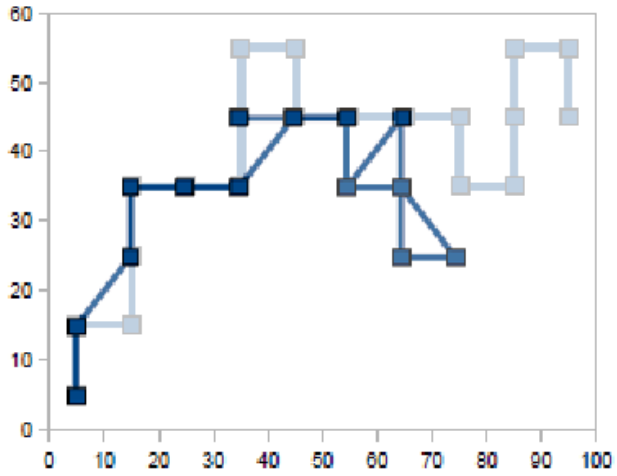

Fig. 12 – Markov within 10 ft

Markov Localization worked well until it deviates enough and gets lost. When Markov guessed wrong and the actual location was outside of the search radius, predictions became inaccurate. Since the search-space is constrained, once it gets lost, recovering was unlikely.

Since Markov looked at a probability map, a probability density function that better represents the variation of RSSI values is needed. The simplification to a Gaussian distribution may be sufficient for the majority of the fingerprint data; however, a few routers exhibited behavior that was not Gaussian. This inconsistency could have contributed to the error.

Method	Min Error	Mean Error	Max Error
Unconst. NN	0	3.62	22.36
NN within 10ft	0	1.21	14.4
Unconst. Markov	0	4.33	28.28
Markov within 10ft	0	1.99	44.17

Table 1 – Distance Error (in ft) for each method.

V. Conclusions and Future Work

The results were promising for localization with live RSSI data taken soon after fingerprinting. It was determined that the fingerprint data drifts over time breaking the system. Markov localization did not perform as well as Nearest Neighbor because probability distributions used for fingerprint data approximated Gaussian when in reality some of the RSSI distributions were not Gaussian.

More work could be done in the future to develop a probability distribution function that better represents the variation in RSSI readings. The full implementation also needs to deal with the fingerprint drift by either developing a fast calibration system or faster fingerprinting method. Accuracy can also be improved by improving the density of

reference points; a faster fingerprinting method and exploration of data interpolation will be especially helpful for this. Further work could also be done to develop a more accurate child motion model for use with the Markov prediction step. On the research platform side of the project, tools for better data analysis and applications for the collected data could be developed.

ACKNOWLEDGMENT

The group would like to thank Dr. Christopher Clark for overseeing our project and providing input, Dr. Jennifer Jipson for the idea of the project, and Tina Risse for helping the group acquire equipment to make the project possible.

REFERENCES

[1] U. Grossmann, M. Schauch, S. Hakobyan, "RSSI based WLAN Indoor Positioning with Personal Digital Assistants," *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Sept. 2007

[2] Oasis Landscape Architecture and Planning, "Cal Poly Child Development Department Pre-School Lab – Playground Accessibility Improvements." Blueprint. June 2008

[3] C. Clark, "CPE485-AMR-Lecture 8- Markov Localization." Winter 2010.

APPENDIX

1.) Fingerprinting Tool – frm_main.cs – provides the GUI and high-level functionality of the fingerprinting utility

2.) Fingerprinting Tool – URL.cs – provides data storage and parsing methods for a given router URL

3.) Fingerprint Parsing Tool – FingerprintParser.cpp – parses fingerprint data into cvs. formats of average values and standard deviations.

4.) Markov Code – Markov.cpp – runs dynamic data sets through the Markov prediction algorithm

5.) Fingerprint (x,y) Averages – avg_10f_tr.csv

6.) Fingerprint (x,y) Standard Deviations – stdev_10f_tr.csv