SITE-SPECIFIC RSS SIGNATURE MODELING FOR WIFI LOCALIZATION

By

Brian J. Roberts

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Electrical and Computer Engineering

May 2009

APPROVED:

____

Prof. Kaveh Pahlavan, Thesis Advisor

____

Prof. William R. Michalson, Committee Member

____

Prof. Allen H. Levesque, Committee Member

____

Prof. Fred J. Looft, Department Head

Abstract

A number of techniques for indoor and outdoor WiFi localization using received signal strength (RSS) signatures have been published. Little work has been performed to characterize the RSS signatures used by these WiFi localization techniques or to assess the accuracy of current channel models to represent the signatures. Without accurate characterization and models of the RSS signatures, a large amount of empirical data is needed to evaluate the performance of the WiFi localization techniques. The goal of this research is to characterize the RSS signatures, propose channel model improvements based on the characterization, and study the performance of channel models for use in WiFi localization simulations to eliminate the need for large amounts of empirical data measurements.

In this thesis, we present our empirical database of RSS signatures measured on the Worcester Polytechnic Institute campus. We use the empirical database to characterize the RSS signatures used in WiFi localization, showing that they are composed of connective segments and influenced by the access point (AP) location within a building. From the characterization, we propose improving existing channel models by building partitioning the signal path-loss using site-specific information from Google Earth. We then evaluate the performance of the existing channel models and the building partitioned models against the empirical data. The results show that using site-specific information to building partition the signal path-loss a tighter fit to the empirical RSS signatures can be achieved.

Acknowledgements

I would first like to express a very special thanks to Professor Kaveh Pahlavan, for his guidance, support, and encouragement during this research. Through the thesis process I learned from Prof. Pahlavan not only about how to perform scholarly research, but also life lessons he told through metaphoric stories. I will keep this experience with me always.

I wish to thank the members of the M.S. thesis committee, William R. Michalson, and Dr. Allen Levesque for their valuable time and comments on my research.

To my family and friends, who acted as a valuable support team giving me the confidence and motivation to complete this life achievement, I send you my deepest thanks.

Finally, thanks to Dr. M. Heidari of SkyHook Wireless who provided the RSS data used to populate the empirical database.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

WiFi localization using received signal strength (RSS) is attracting attention as a complement to the Global Positioning System (GPS) for indoor and outdoor urban areas where GPS does not perform satisfactorily [1][2]. The growth of WiFi access points (AP) and mobile WiFi devices have spawned many new applications, which use WiFi for both connectivity and localization. The idea of WiFi localization was first introduced by Bahl for indoor location [3]. Bahl's concept took advantage of the growth of WiFi enabled devices and massive deployment of WiFi AP's. Bahl set the ground work for Roos, who used a probabilistic approach to increase the accuracy of indoor location [4]. From these works, location-based services companies have successfully commercialized the idea for indoor WiFi localization [2].

WiFi localization became feasible for outdoor environments as the deployment of WiFi APs and WiFi enabled mobile devices became ubiquitous. For outdoor areas it is not feasible to collect large amounts of calibration data as done for indoor techniques. Place Lab overcame this barrier through the use of a probabilistic algorithm, which only required sparse calibration data [5]. The WiFi localization techniques used by outdoor localization algorithms are based on RSS signatures [5][6]. A RSS signature for a given AP comprises a series of signal detections over a specified measurement path. To provide localization, the RSS's of the APs detected by a mobile device are compared with a large database of location-tagged RSS signatures using a variety of algorithms. The RSS signature database is collected through a costly and time-intensive process of war-driving

[7]. The war-driving process involves systematically driving through an entire area where WiFi localization is to be performed to collect location-tagged RSS measurements for each detected AP.

1.2 Motivations

To evaluate the performance of WiFi localization algorithms, a large database of RSS signatures is needed. The RSS database can be created empirically through measurements or theoretically through channel models. Collecting large amounts of empirical data is costly, time intensive, the AP environment is inflexible, and the actual locations of the AP's are unknown. These limitations affect the types of research that can be performed using empirical measures. For example, it would be difficult to investigate the effects of AP density and distribution on localization algorithm performance. Each AP layout scenario would need to be setup and measured. On a large-scale this would not be feasible.

Therefore theoretically building the RSS signature database using an accurate channel model is preferred, but it is important to use a model that accurately characterizes the channel. Accuracy is important as it has been shown that the effects of the physical layer characteristics impact the performance of algorithms and can even affect the ranking among algorithms in performance evaluations [8]. To date, no evaluation of the accuracy of WiFi channel models to simulate the RSS signatures used by outdoor WiFi localization algorithms has been performed.

1.3 Contributions of the Thesis

The objective of this thesis is to characterize the RSS signatures used in WiFi localization for a campus environment and propose improvements to existing models based on the findings to allow for theoretical RSS signature database creation.

We began our research by characterizing the RSS signatures on the WPI campus using a significant empirical RSS database. From this characterization we found the following. The RSS signatures were composed of connectivity segments whose length's best fit a lognormal distribution. The detected mean RSS values were on the lower end of the WiFi receiver sensitivity limit. There was not a tight relationship between the distance from the AP to the measurement point and the RSS value. The RSS values of AP's located on ground to third floors followed the same distribution. Finally, it was observed that the RSS signature behavior was affected by the location of the AP within the building in respect to the distance from the exterior wall.

From the results of our RSS signature characterization we proposed a new building partitioned modeling method. The building partitioned method made use of site-specific information readily available from Google Earth^{TM} to find the coordinates of the building footprint containing the AP. Using the building footprints the new models were able to account for the location of the AP relative to the exterior wall of the building.

Lastly, we evaluated the performance of two existing models (JTC and 802.11) along with the proposed building partitioned models against the empirical data. The evaluation showed that by adding site-specific information to the channel models a 10% performance increase is gained.

The long term goal of this research is to allow for the development and evaluation of WiFi localization algorithms using a channel model and eliminate the need for large and costly empirical data collection.

1.4 Thesis Organization

This thesis is organized into six chapters. Chapter 1 gives an introduction to our research including background motivation and contributions. Chapter 2 explains the concept behind WiFi localization and the different methods of implementation. Chapter 3 we present our RSS signature database and discuss the work performed to characterize the RSS signatures in a campus environment. In Chapter 4, the current models used for RF channel modeling are presented and we propose improving these models by introducing a building partitioning method. In Chapter 5, we analyze the performance of each channel model against the empirical data using our defined performance criteria. Lastly in Chapter 6, we discuss the conclusions found during this research and future work.

Chapter 2: WiFi RSS Localization

2.1 Introduction

WiFi localization is broken in to three phases the data collection, offline, and real-time phases. The data collection and offline phases are performed before the real-time phase and provide the search space for the localization algorithms used in the real-time phase (see Figure 2-1).

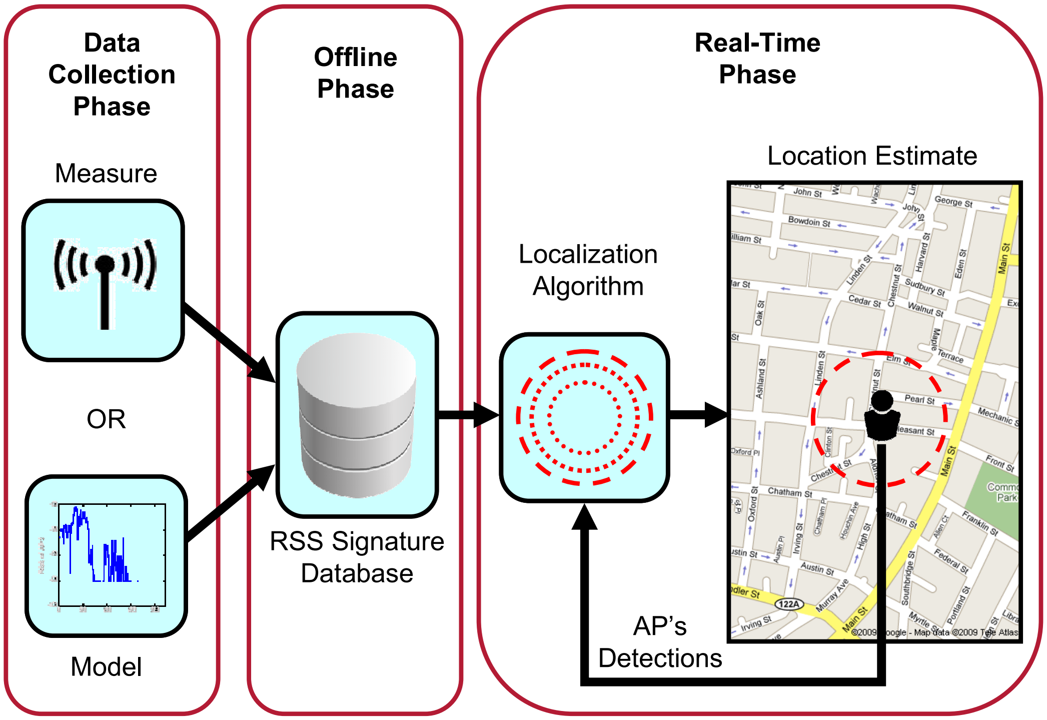


Figure 2-1 WiFi Localization Process

In the data collection phase an RSS database is created to provide a search space for the real-time phase. The RSS database is composed of location-tag RSS measurements from detectable AP's in the area where localization is to be performed. The data collection can be done empirically by measuring the RSS of AP in a systematic fashion or theoretically computed using channel models. In the offline phase, the data from the data collection phase is transformed to minimize computation time during the real-time phase, reduce the size of the database and/or increase accuracy of the location estimate. In Section 2.2, the two options for performing the data collection phase are discussed. In Section 2.3, the localization algorithms used in the real-time phase are discussed.

2.2 Data Collection Phase and RSS Database Creation

In the data collection phase the goal is to create the RSS database to provide a search space for the real-time phase. The RSS database can be built either by empirically measuring AP's RSS in a systematic fashion over the designated area for localization or by theoretically computing the AP's RSS's using a channel model. Both methods have advantages and disadvantages. Empirically collecting the RSS's is costly and time consuming while computing the RSS database using channel models requires information about the location of the APs and the path-loss environment which may not be available. In Section 2.2.1, we describe the empirical data collection process and in Section 2.2.2 we describe the 2.2.2 the theoretical data generation process using channel models.

2.2.1 Empirical Data Collection

Collecting the RSS database empirically is performed by war-walking or war-driving. The process entails systematically measuring RSS data from the detectable AP's over the area where localization is to be performed. Each RSS measurement is tagged with a location reference and stored as tuples of the form {RSS, Location}. The process of measuring the RSS's is time consuming and costly. Additionally, the process may need to be repeated as APs are moved. The process of empirically collecting the RSS database differs from indoors to outdoors due to the increased scale of location data and location reference used.

2.2.1.1 Indoor Data Collection Process

When collecting the RSS data indoors a scale floor plan is needed for the location reference. The area to be surveyed is divided up into a grid defining the locations where RSS measurements will be taken (Figure 2-2). At each location several RSS measurements are taken by a mobile receiver and the location is recorded. It is recommended that the measurements are taken facing in each direction to mitigate shadow fading effects. Indoor surveys for data collections cost on the order of thousands of dollars and consist of on the order of tens of AP's. This process is much easier to collect and update for indoor environments than outdoors [7].

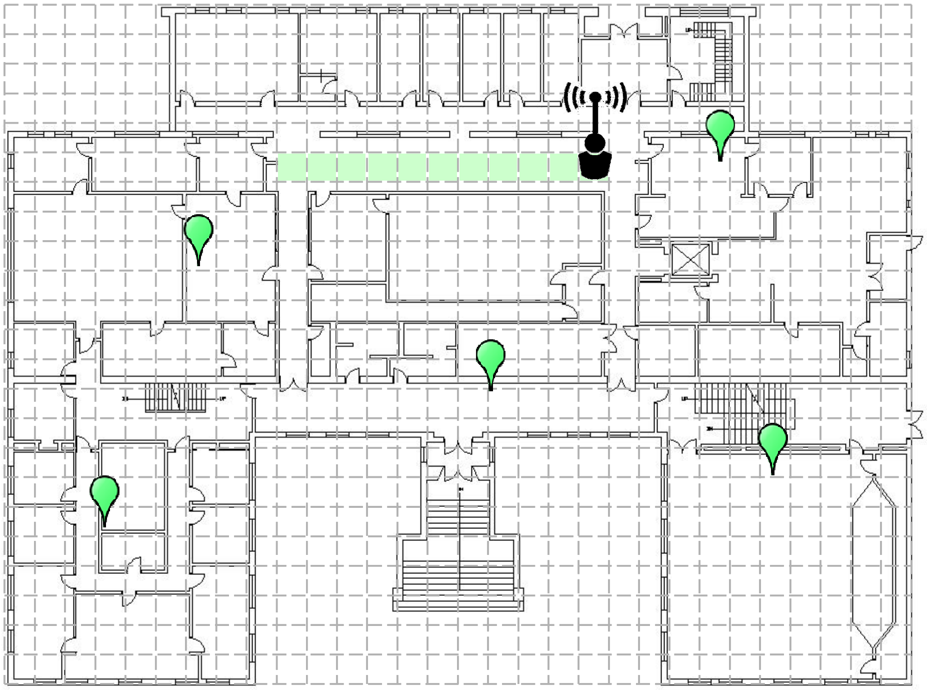


Figure 2-2 Indoor empirical RSS data collection notional diagram

2.2.1.2 Outdoor Data Collection Process

For outdoor areas it is not feasible to collect large amounts of calibration data as done for indoor techniques. Place Lab over came this barrier by the use of a probabilistic algorithm, which only required sparse calibration data [5]. The WiFi localization technique used for outdoors is based on WiFi RSS signatures [5][6]. A WiFi RSS signature for a given AP is the series of signal detections over a specified measurement path.

To collected empirical data outdoors latitude and longitude coordinates are used for location reference. The process of collecting this outdoor data as been dubbed war-driving or war-walking and involves driving or walking on road ways and paths to collect RSS measurements (Figure 2-3).

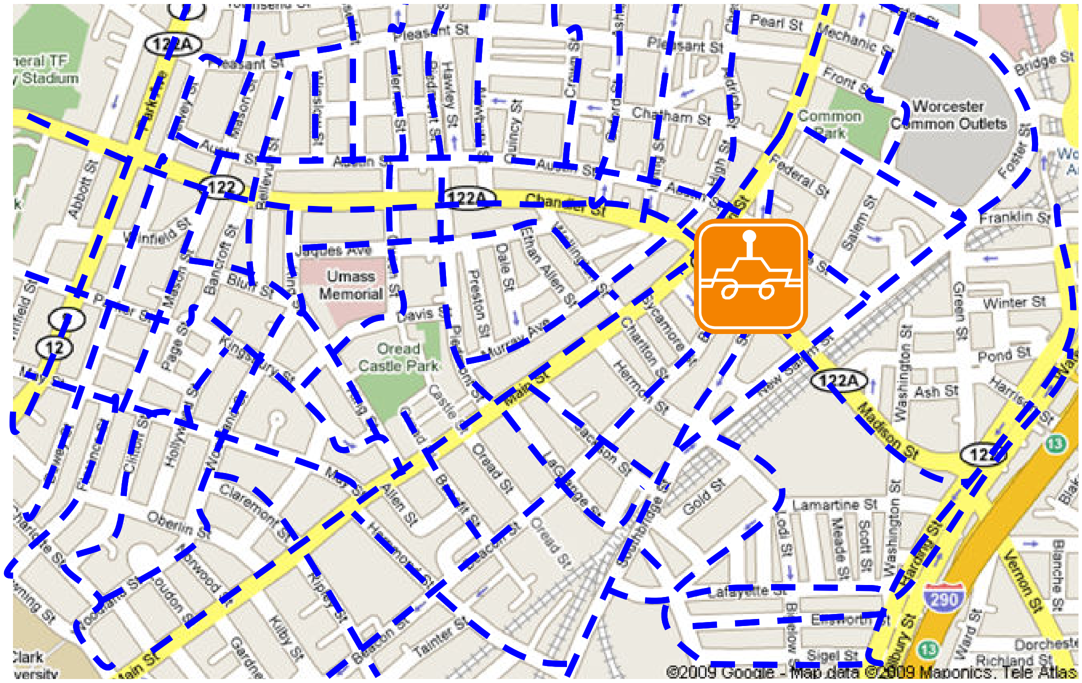


Figure 2-3 Outdoor empirical RSS data collection notional diagram

The two methods differ only by the means of travel. Both use GPS to obtain accurate latitude longitude coordinates. Given this, war-driving is not affective in areas where GPS does not work properly. War-driving produces a much larger database than indoors on order of millions of APs and is much more expensive to collect (~$5M per US survey).

2.2.2 Theoretical Data Generation Using Channel Models

An alternative to collecting empirical RSS measurements is to generate the data points theoretically using channel models. Equation (2-1) gives the equation for computing the RSS of a transmitter distance, *d*, in meters from the receiver.

$$RSS(d) = P_t - L_p(d) \tag{2-1}$$

Where P_t is the transmit power 40dBm for the WiFi standard and L_p is the path-loss in dB over the signal path (Figure 2-4) [9].

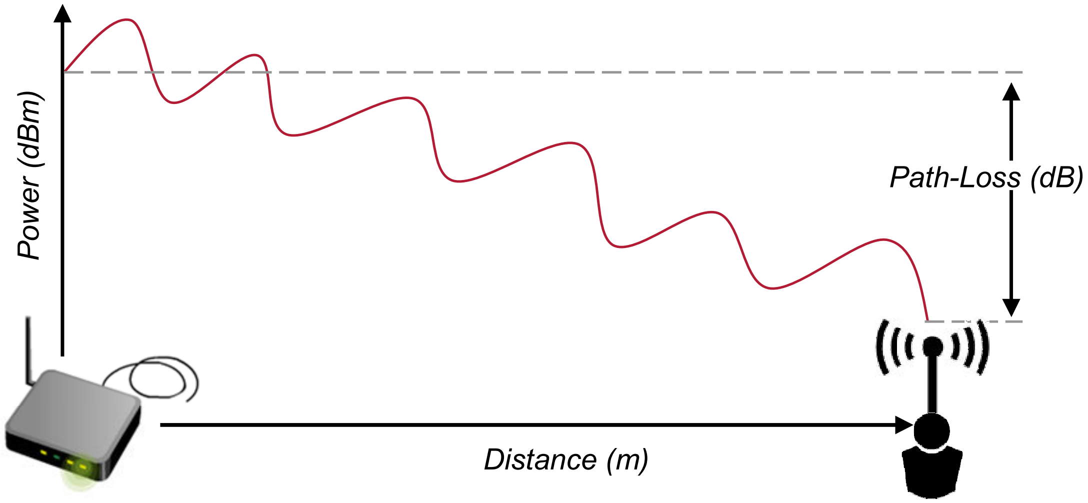


Figure 2-4 Channel Modeling notional diagram

The first item to note about using a channel model to compute the RSS database is that the locations of the APs must be known to calculate *d* unlike the empirical method. This is not an issue if the APs are a part of a planned infrastructure where their locations have been documented. There are several methods to compute L_p each with their strengths and weaknesses. In this paper we chose to focus on two path-loss channel models the Joint Technical Committee (JTC) model and the Distance Partitioned 802.11 model.

2.3 Localization Algorithms

In the real-time phase the AP's detected by the mobile receiver are matched to the RSS signature database to provide a location estimate. The performance of these algorithms is measured the difference in the estimated position from the actual position. In this Section we present several localization algorithms used in WiFi RSS localization.

2.3.1 Centroid

The centroid algorithm is a simple one, which reduces the size of the RSS database and therefore is used most for outdoors [10]. After the data collection phase during the offline phase the mean location for each AP RSS detection is calculated. The mean location serves at the AP location estimate. The estimate for each AP is then stored in a database. During the real-time phase the mobile device location is estimated by finding the centroid of all AP's detected.

2.3.2 Nearest Neighbor / Fingerprinting

The nearest neighbor algorithm was first proposed for indoor localization in RADAR [3]. It was later used for outdoor localization, but at a much coarser measurement granularity in reference [10] where it was defined as fingerprinting. The basic idea is that every location has a unique fingerprint based on the detected AP's and RSS's. The nearest neighbor or fingerprinting algorithms uses this idea to locate a mobile receiver by matching the AP RSS location fingerprint to a database of location tagged fingerprints collected in the data collection phase. The matching fingerprint if found by determining the Euclidean distance of each fingerprint. The location of the fingerprint that is the closest match to the detected fingerprint is estimated to be the current location of the mobile receiver. The distance error formula is given by

$$d_{err} = \sqrt{\sum_{i=1}^{n} \left[\left(Pr_i - Pr'_i \right)^2 \right]} \tag{2-2}$$

where n is the number of detected AP's, and Pr is the RSS of detected by the mobile receiver and Pr' is the fingerprint AP RSS.

2.3.2.1 Multiple Nearest Neighbor

The multiple nearest neighbor algorithm expands of the nearest neighbor algorithm by using the closes k neighbors. The locations of the k neighbors are averaged to estimate the location of the mobile receiver. The idea is that there are usually multiple neighbors with relatively close computed distances given the inherent variation in the RSS's and there is no reason to use only the nearest neighbor.

2.3.2.2 Ranking

The ranking algorithm is a variation of the nearest neighbor that uses relative RSS instead of absolute to overcome differences in WiFi cards [10]. The rankings are then compared using the Spearman rank-order correlation co-efficient

$$rs = \frac{\sum_i \left(R_i - \overline{R}\right)\left(R'_i - \overline{R'}\right)}{\sqrt{\sum_i \left(R_i - \overline{R}\right)^2} \sqrt{\sum_i \left(R'_i - \overline{R'}\right)^2}} \tag{2-3}$$

where $\overline{R}$ and $\overline{R'}$ the means of the rank vectors for the *i* AP's.

2.3.3 Particle Filters

In the past particle filters have been used in robotics to combine streams of sensor data into location estimates [11]. A particle filter is a probabilistic approximation algorithm for a Bayes filter [10]. In location estimation, Bayes filters probabilistically estimate a person's or object's location from noisy sensor observations [12]. Bayes filters represent the uncertainly of the location estimate at time *t* by a probability distribution of a random variable x_t called a belief. Particle filters represent the beliefs by sets of samples or particles:

$$Bel(x_t) \approx S_t = \left\{ \left\langle x_t^{(i)}, w_t^{(i)} \right\rangle \mid i = 1, \ldots, n \right\} \tag{2-4}$$

where $x_t^{(i)}$ is the location, and $w_t^{(i)}$ are the nonnegative weights called importance factors [12]. The importance factors represent the probability that a device will hear the observed scan if it were at the position of the particle and are trained using the measurements taken in the data collection phase [10].

Chapter 3: Empirical WiFi RSS Signature Analysis

3.1 Introduction

This section details the efforts to characterize the RSS signatures in a campus environment. To perform the characterization, an empirical database of location-tagged RSS measurements was built using measurements made around Worcester Polytechnic Institute (WPI), in Worcester, Massachusetts. The WPI campus was chosen because the WiFi AP infrastructure was known. The RSS data was then characterized to understand the behavior of the RSS signatures in the environment where they were produced. From the characterization, modifications to the current channel models are proposed to add site-specific information for dynamic parameters.

3.2 Description of the Empirical RSS Database

To characterize the RSS signatures, an extensive database of empirical RSS measurements was built. The database consisted of location-tagged RSS measurements from APs over measurement paths at and around the WPI Campus. The data was collected using a standard off-the-shelf WiFi card for the RSS measurements, and a GPS for accurate position information. The measurements were taken on foot while walking at a steady pace.

Figure 3-1 shows the paths taken during the data collection (indicated by the yellow lines) as well as the AP locations (indicated by the green balloon markers) and Table 3-1 lists the main attributes of the empirical RSS database.

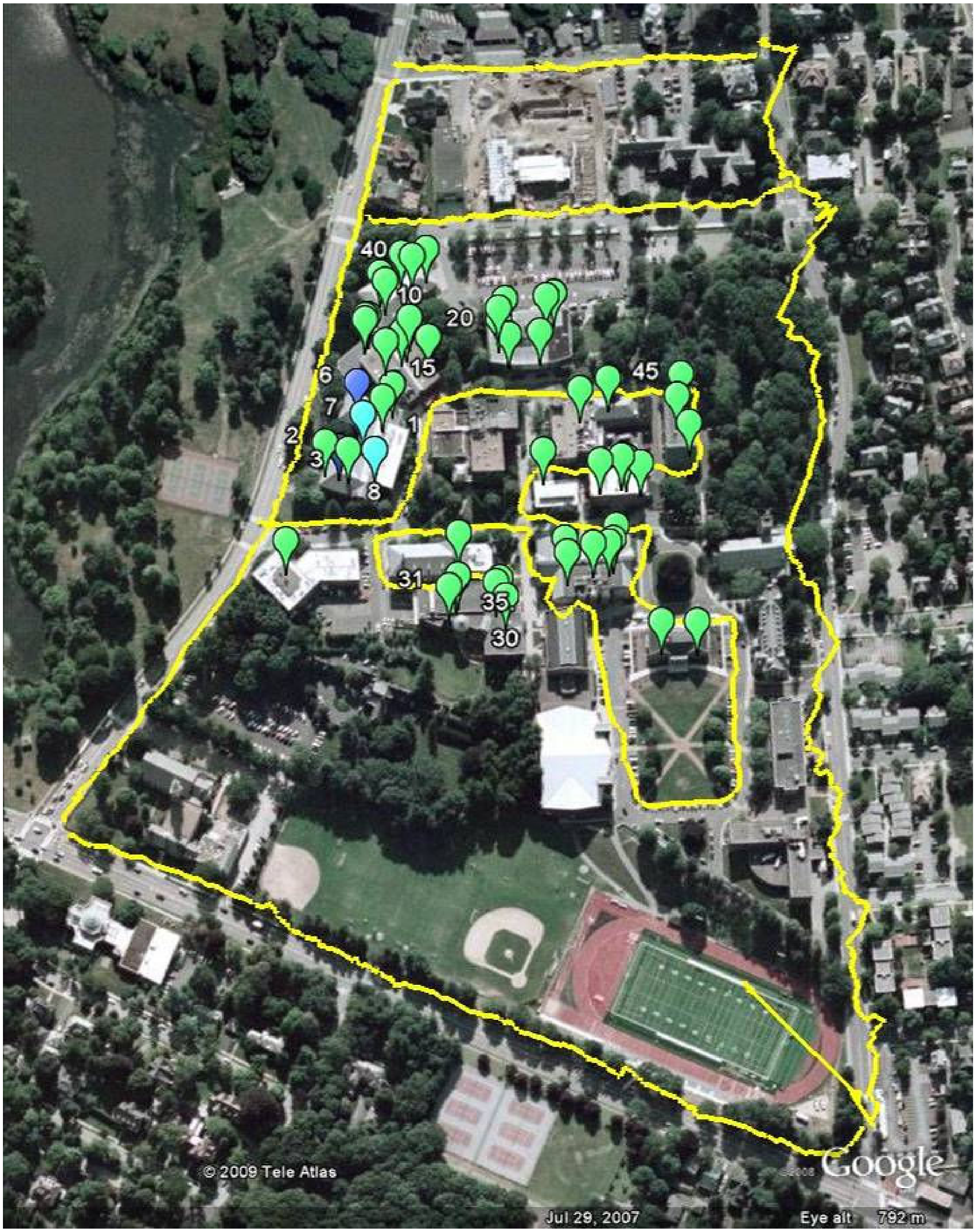


Figure 3-1 Empirical RSS database measurement paths indicated by the yellow lines and AP locations shown as green balloons

Table 3-1 Attributes of the empirical RSS signature database collected for characterization

Attribute	Value
Buildings containing access points	13
Number of access points	47
Number of AP signal detections	21,732
Number of measurement sample points	3,833
Measurement path distance	6,452m
Number of unique measurement paths	14
Range of floors containing access points	0-3
Range of RSS values detected	-90dB to -55dB
Sample rate of RSS measurements	1/s

Figure 3-2 shows the distribution of the distances between the AP's and measurement points.

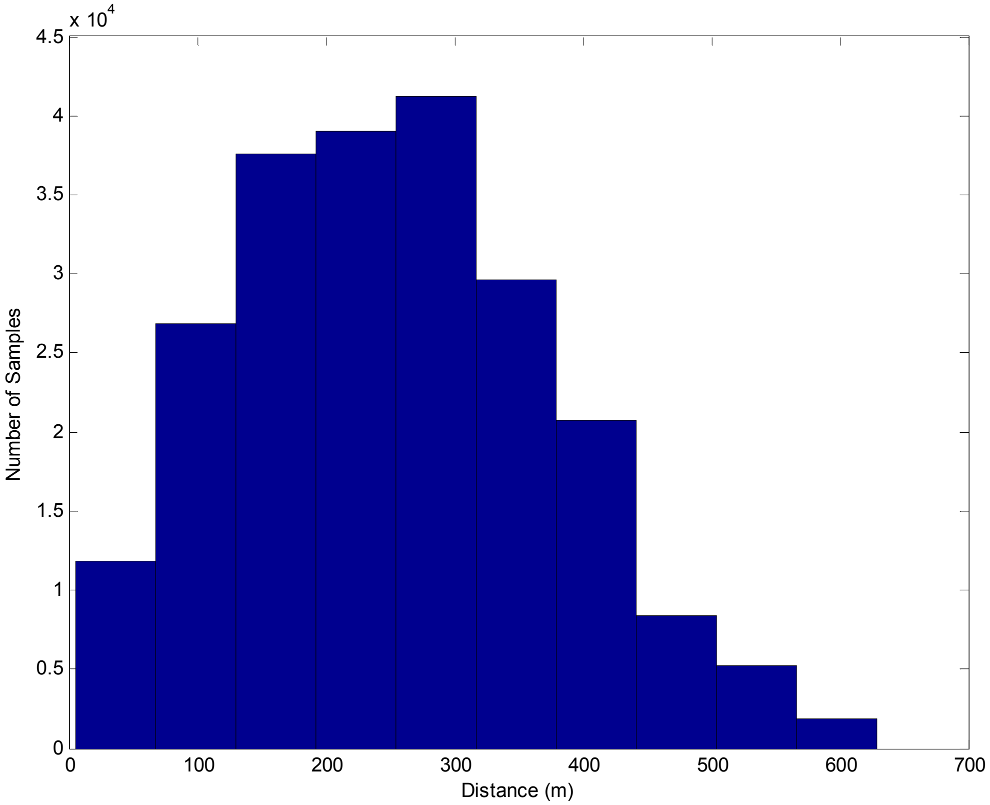


Figure 3-2 Histogram of distance between AP and measurement point

The WPI campus was chosen for several of reasons: it has a high concentration of APs, the locations of the APs were known, and it fit the scenario for a mobile WiFi user. The high concentration of APs was representative of urban areas where WiFi localization is most effective. It was a necessary to know the locations of the APs to use the channel models, which require the distance from the AP to the measurement point to be known. The campus fit the mobile WiFi scenario because there are available measurement paths both along streets and sidewalks frequented by mobile WiFi users, and there is a high concentration of APs in the surrounding buildings.

The empirical RSS database was broken into two sets of data: the location-tagged RSS measurements and the AP location information. The RSS measurements were formatted such that each measurement path was a separate ACSII text file. The RSS database consisted of 14 paths and therefore contained 14 path files. Each file consisted of the latitude and longitude (supplied by the GPS), Service Set Identifier (SSID), Media Access Control (MAC) address, timestamp, and RSS. The MAC addresses were used as unique identifiers for each AP since the SSID can be the same for multiple AP's, as was the case for the AP's on the WPI campus. The latitude and longitude coordinates where used to identify the unique measurement points. If multiple samples existed at the same coordinate on a path they where averaged to produce one RSS for each AP for each coordinate on a path.

The AP location information was formatted with an entry for each AP on the WPI campus. The AP parameters stored in the database were the MAC address, name of the building it was located in, the floor it was on, the latitude and longitude, and SSID. Again the MAC addresses were used as unique identifiers for the AP's. Seventeen AP's were located on ground floors, seventeen on first floors, sixteen on second floors and ten on third floors in buildings on campus.

3.3 WiFi RSS Signature Characterization

Using the RSS database, the behavior of the RSS measurements was investigated. To aid in the analysis of the RSS behavior the measurements were spatially grouped into RSS signatures. An RSS signature is the series of spatially related RSS measurements along the measurement path for a given AP also referred to as a network availability trace. The goal of this analysis is to understand the behavior of the signatures instead of the behavior of a specific measurement point. This allows for a more coarse characterization which is all that is necessary for the WiFi localization application.

To reduce the complexity a subset of the database was used for the initially characterization. After the analysis was completed for the subset we then applied this to the entire database. The Atwater Kent Laboratory (AK) building was chosen for the subset as it contained seven AP's over three floors and had measurement paths that covered three sides of the building. The AP's are spread throughout the building to maximize coverage inside the building. AP's 1-3 are located on the first floor AP 5 and 6 are located on the second floor and AP's 7 and 8 are on the third floor. Note that AP 4 was removed from AK and therefore was not in the RSS database. Figure 3-3 shows the measurement path around the building and the locations of the AP's.

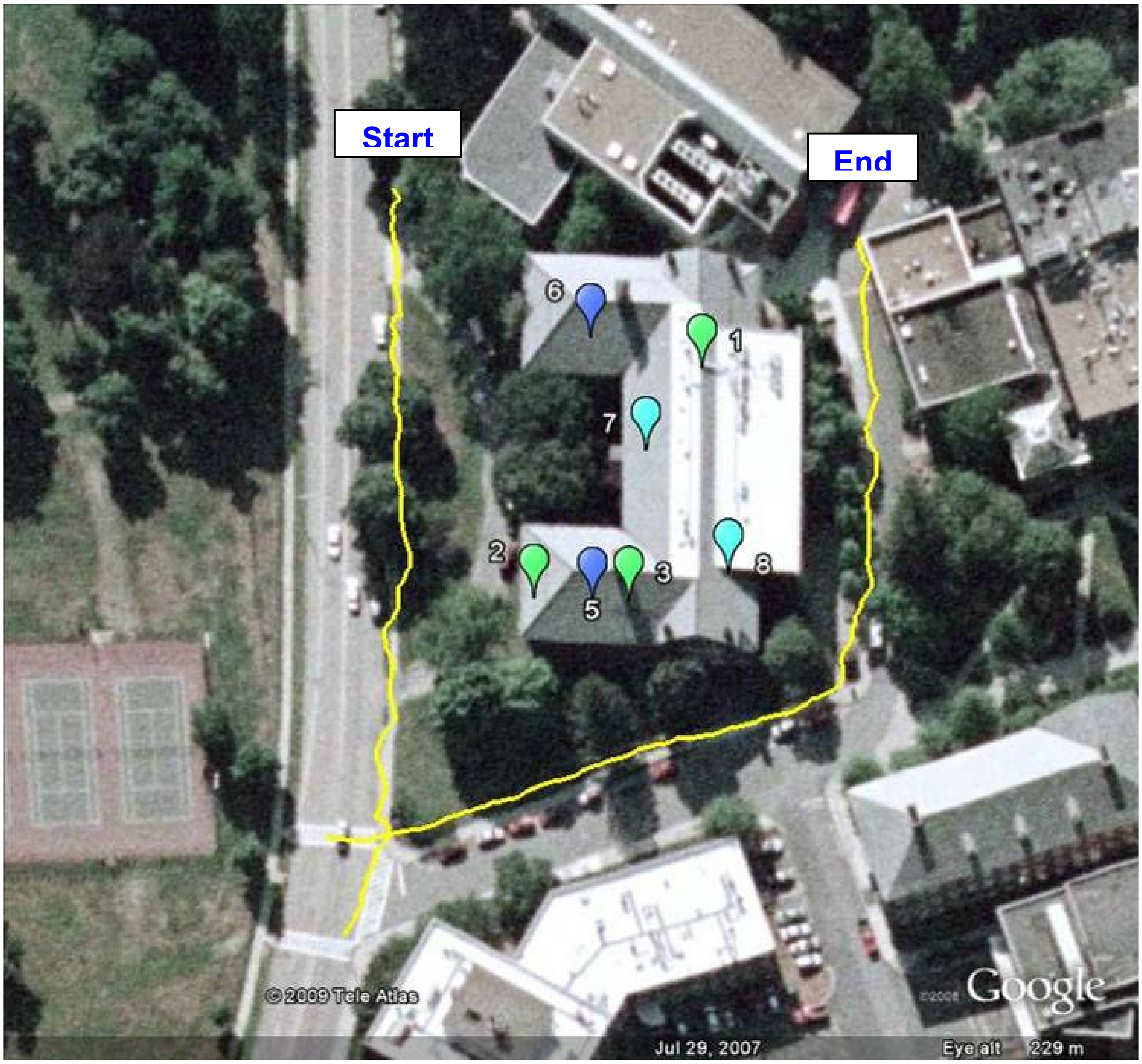


Figure 3-3 Measurement Path of RSS signatures in Atwater Kent

We began our analysis by looking for a relationship between the RSS and the distance between the AP and the measurement point. In simple channel models the RSS is calculated as a function of the distance between the AP and the receive [5]. To understand if this relationship holds true for the indoor to outdoor RSS measurements, the RSS measurements were plotted versus the distance between the receiver and the AP as shown in Figure 3-4.

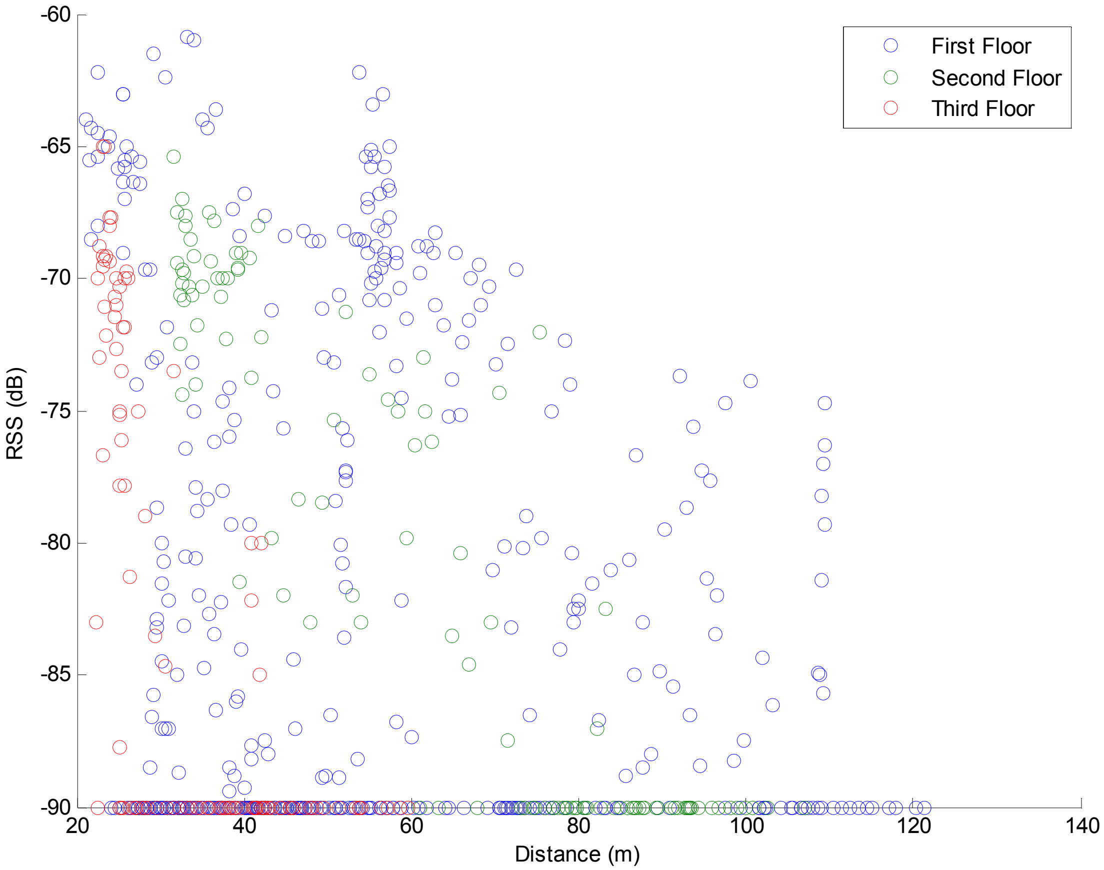


Figure 3-4 Scatter plot of RSS for the AP's located in AK versus the distance between the AP and the receiver showing that distance cannot be used as a sole predictor for RSS

From the plot it can be seen that for a given distance up to about 110 meters the RSS's range from -90dB to more than -75dB. Given this more than 25dB wide variation we can see that using the distance from the AP to the receiver as the sole predictor for RSS would not produce accurate results. Additionally, we observed that many of the RSS values were at -90dB. Figure 3-5 shows the distribution of the RSS's and clearly shows that RSS values at -90dB are fifty percent more likely than any other value.

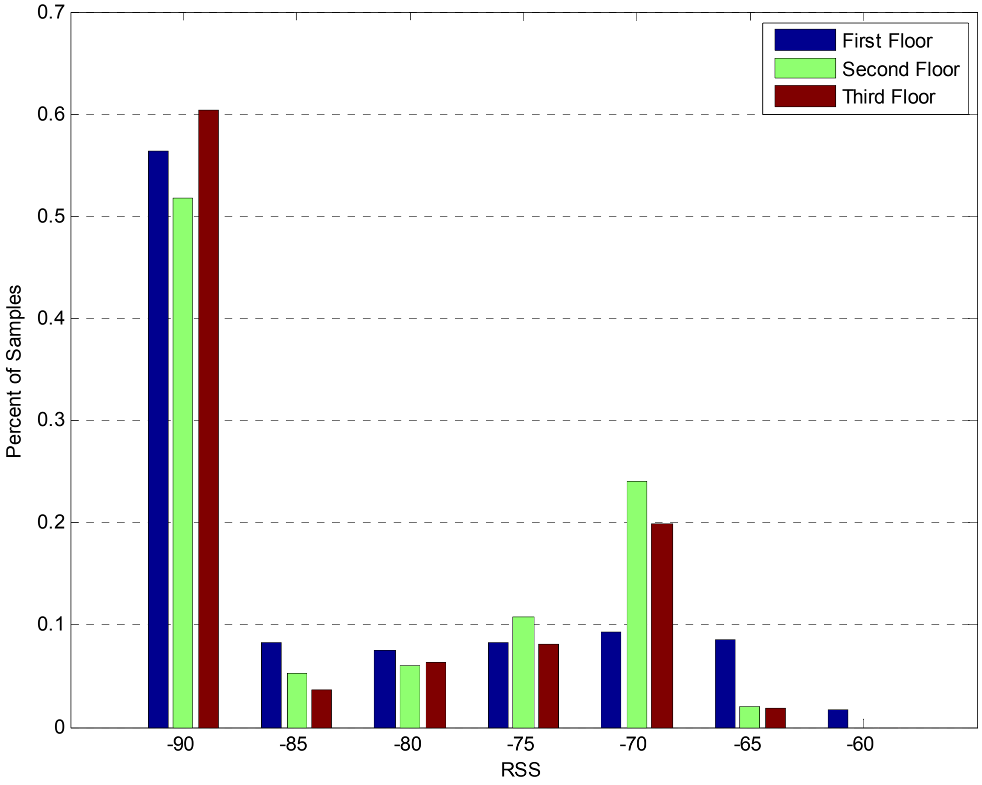


Figure 3-5 Histogram of RSS values for AK subset showing that RSS values of -90dB are measured fifty percent more than other values

Another observation noted from both Figure 3-4 and Figure 3-5 is that the AP's on floor ranging from the first to third show the same trend of RSS values. One possible theory is that the AP signal no matter what floor it is located on travels out through the exterior wall of the building and not down through the floors, which would reduce the RSS. If this theory is true the only difference for an AP located in the third floor from an AP located on the first would be the added vertical distance for an AP would not significantly affect its RSS.

To understand the RSS value distribution relative to the measurement paths the RSS signatures for the seven AP in AK were then plotted in Figure 3-6.

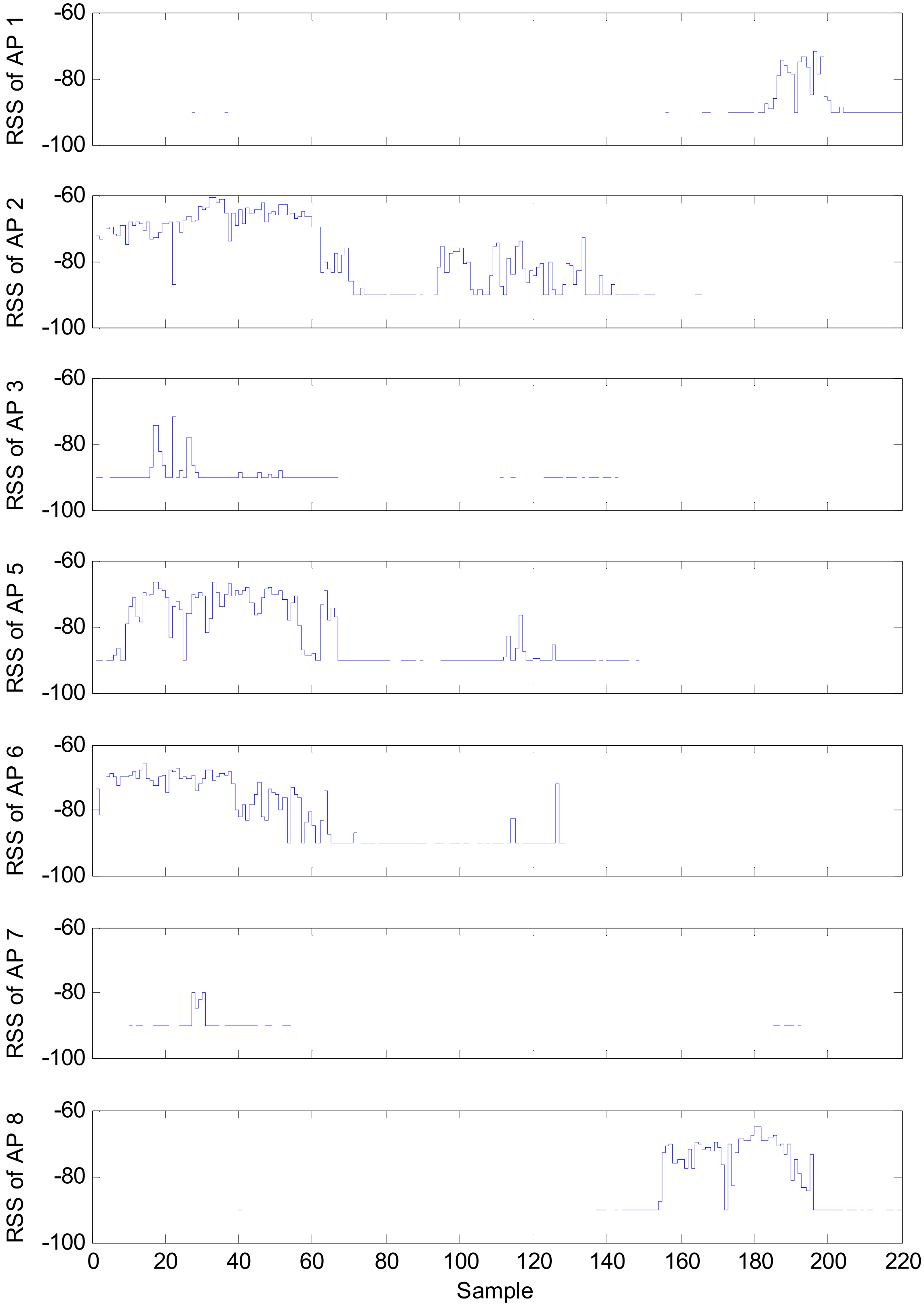


Figure 3-6 Atwater Kent AP's RSS signatures showing connectivity segments

From the plots it can be seen that the RSS signatures are composed of connectivity segments of varying length. Connectivity segments are defined as AP signal detections in adjacent samples along the measurement path. The location and length of the connectivity segments are the main characteristics that affect WiFi localization algorithms. From the plot it can be seen that the connectivity segments end in flat lines at -90dB, which accounts for the high rate of -90dB values noted previously. Given the location of these segments of -90dB being just before the AP signals become undetectable it can be assumed to be due to the sensitivity of the WiFi access card.

The next step in our analysis was to co-locate the RSS signatures in the environment where they were measured to identify factors contributing to their behavior. Google Earth^{TM} was used to plot the RSS signatures along the path they were measured. Since the RSS values were in negative decibels a value of 100 was added to each to make them positive. Figure 3-7 shows the relative RSS signature (in red) for AP 1 in AK on the measurement path around the building. From the figure it can be seen that the RSS signature is present on only one side of the building. The signature is present on the same side of the building as AP 1. This is to be expected because the signal should have less physical obstructions to penetrate the closer the AP is to the exterior of the building.

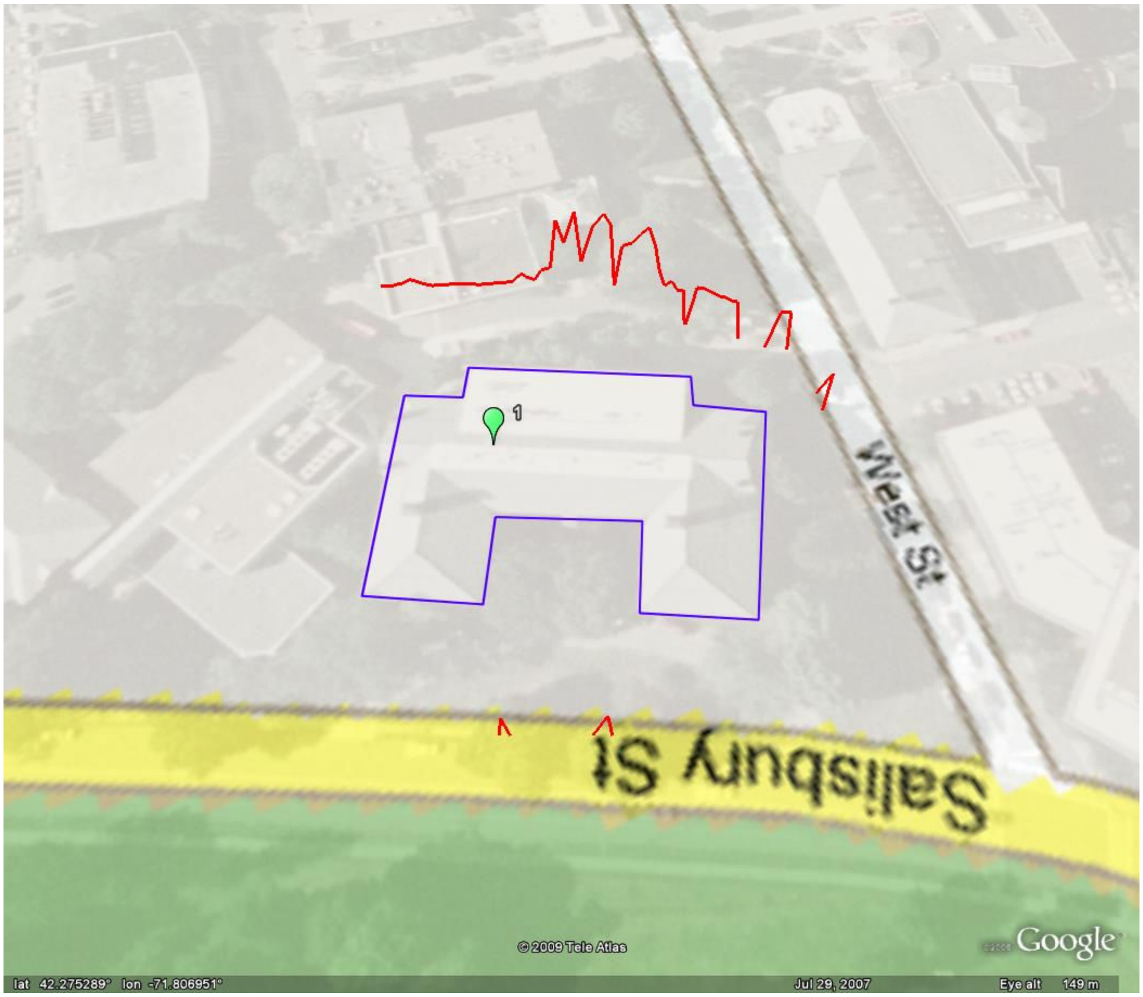


Figure 3-7 RSS signature of AP 1 in AK plotted on Google Map showing the location and length relative to the location of the AP in the building

AP 1 was located on the first floor of AK, we then performed the same inspection for AP which was located on the second floor of AK to determine if AP to building location to RSS signature location relationship continued. Figure 3-8 shows the RSS signature for AP 5 in AK for the path around the building. Again it can be seen that the RSS signature is present on the same side of the building as the AP is located. There is no signature on the left side of the building even though the measurement paths were about the same distance away from the AP. The difference between the two measurement points was the amount of building the signal had to pass through before being received at the on the left side compared to the right, which was more than twice as much.

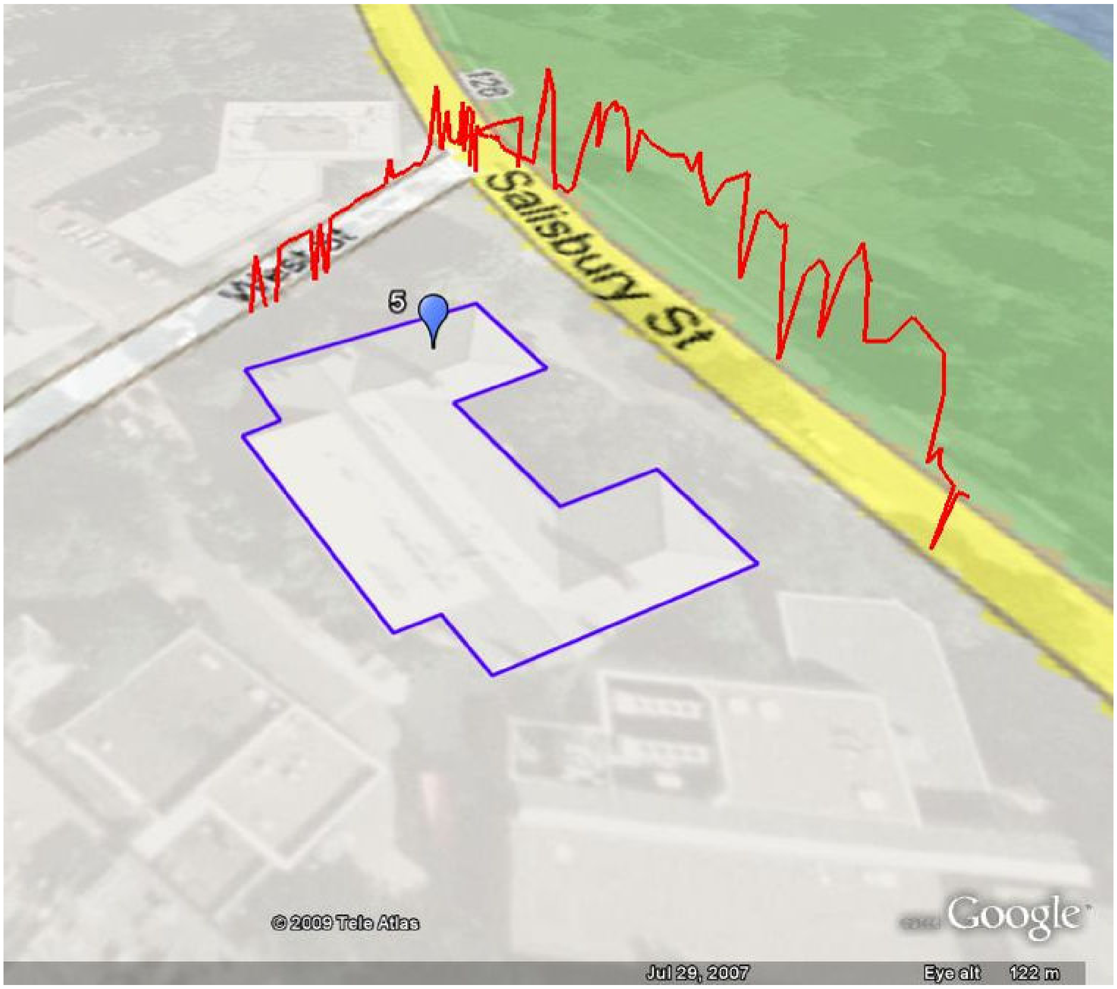


Figure 3-8 RSS signature of AP 5 in AK plotted on Google Map showing the location and length relative to the location of the AP in the building

To better understand this relationship the RSS signature was overlaid on the distance from the AP to the measurement point and the distance from the AP to the edge of the building. Figure 3-9 clearly shows that location of the AP from the edge of the building has a signification impact on if a signal is detected from an AP at a given measurement point.

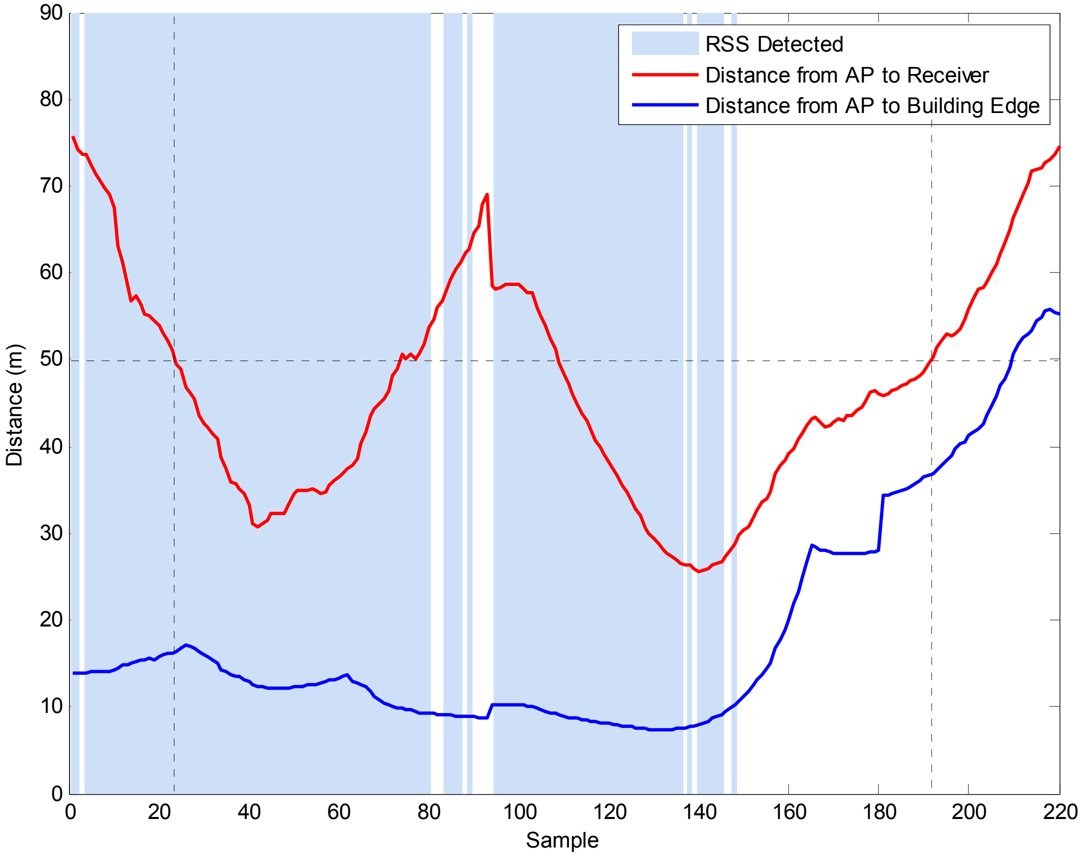


Figure 3-9 RSS signature detections around AK for AP 5 overlaid on plots of the distance from the AP to the receiver and the distance from the AP to the edge of the building

For example, at sample point 24 and 192 the measurement points were both 50 meters away from the AP at sample 24 there was signal detected by the AP, but at sample 192 there was not. This was because at sample point 24 there was only 16 meters of building for the signal to pass through whereas at sample point 192 there was 37 meters of building to pass through. Given this observation, it would be very important for a channel model to account for the location of the AP relative to the building it is within for the model to be accurate.

To understand how the RSS signatures may change over a given measurement path measured on separate occasions, a subset of the database was used for the analyzed.

Figure 3-10 shows the two measurement paths taken on Salisbury Street on two separate locations and the AP's located in AK.

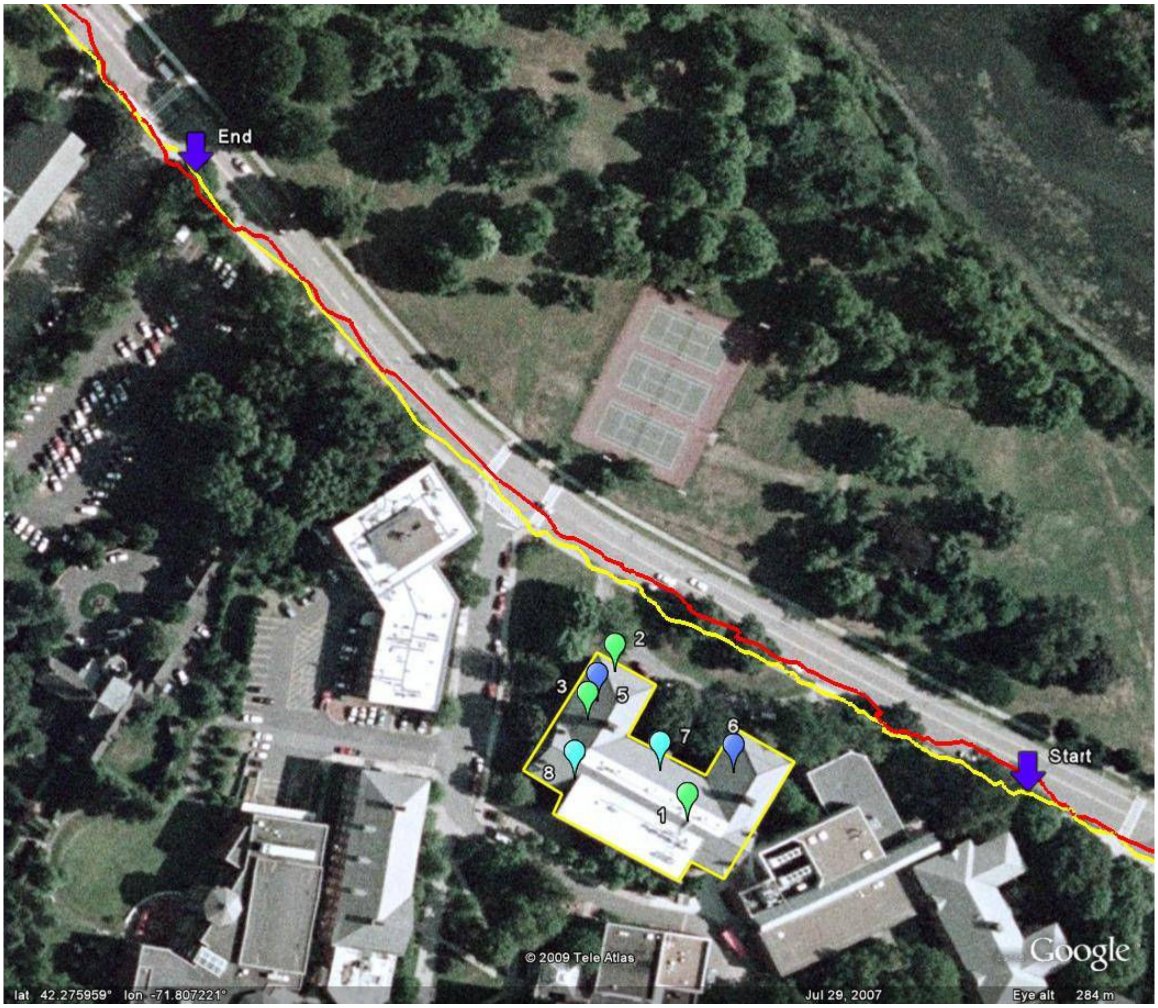


Figure 3-10 Two measurement paths on Salisbury Street measure on separate locations to compare the change in RSS signature

The RSS signatures were then plotted for each AP in AK in Figure 3-11. From the plot it can be observed that the RSS values at almost the same position can vary as much as 20dB from one measurement path to the next. On the other hand, it was found that the two RSS signatures had ~90% overlap in connectivity segments.

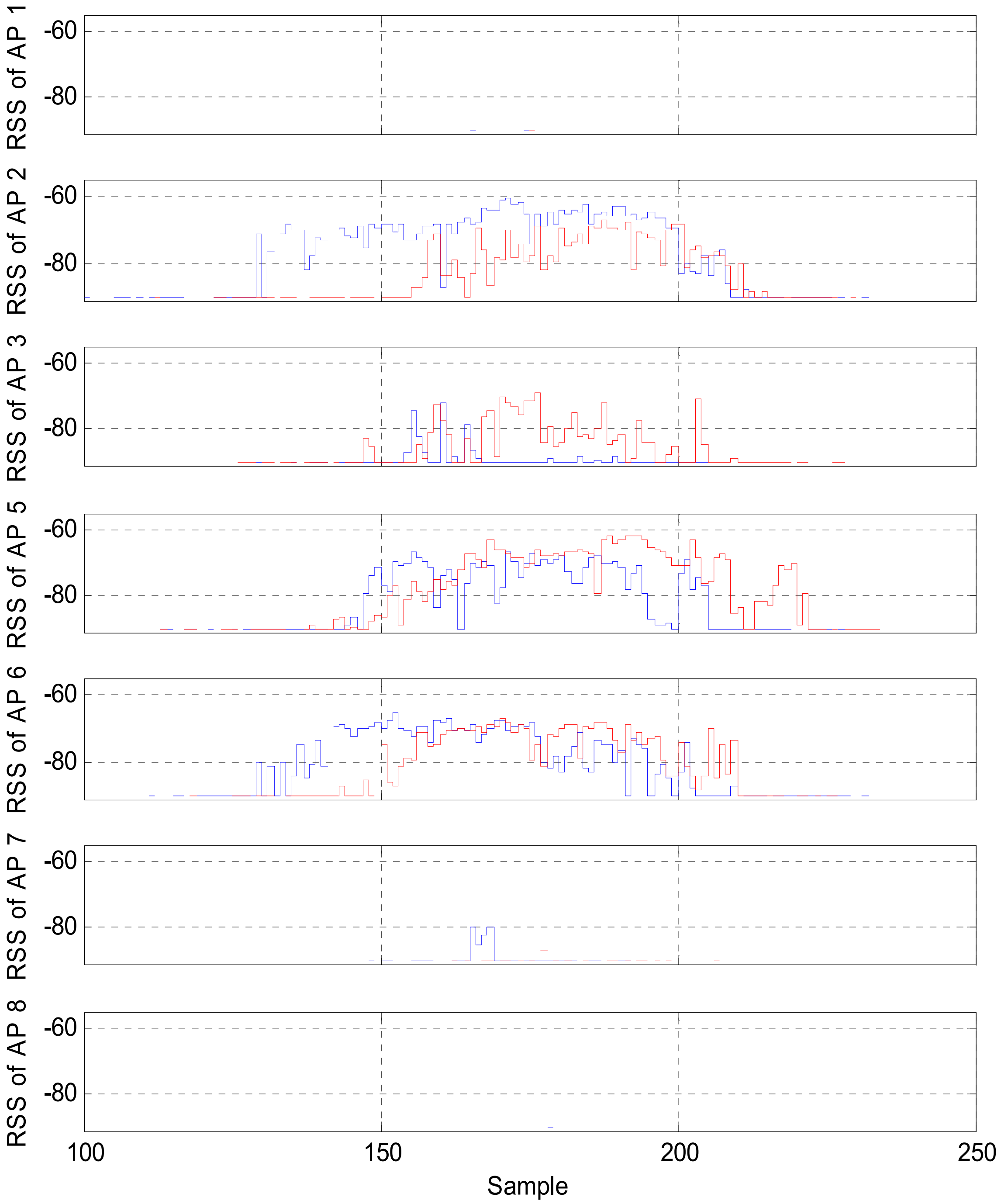


Figure 3-11 AK AP RSS signatures plotted for the two measurement paths on Salisbury Street on separate occasions

From the observations seen in the AK data subset we used the entire database to confirm that they held true. Figure 3-12 is a scatter plot confirming the observation seen in the

AK data subset that the distance between and AP and the receiver is not a good predictor of the RSS value at that point. Again the RSS values range from -90dB to less than -60dB for the same distance between the AP and receiver at difference measurement points.

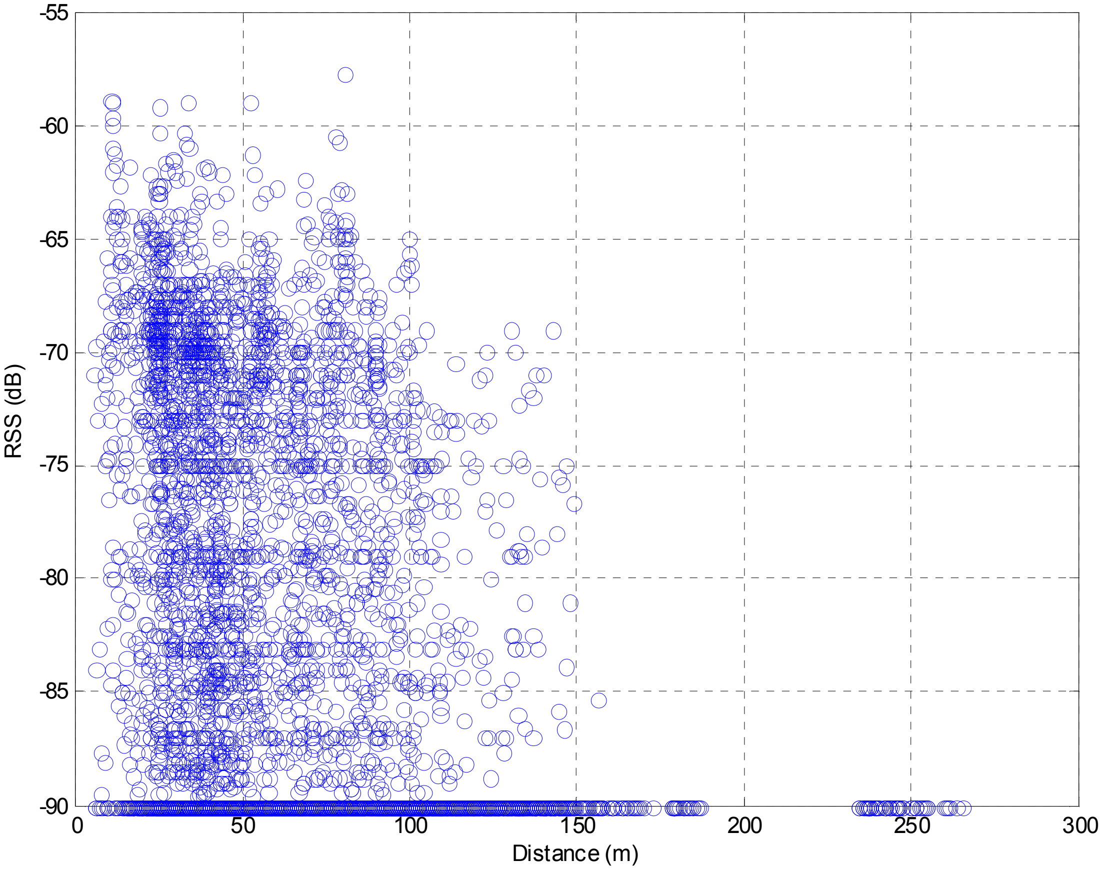


Figure 3-12 Scatter plot of RSS for all AP's versus the distance between the AP and the receiver showing that distance cannot be used as a sole predictor for RSS

Figure 3-13 shows that the distribution of the RSS values for the entire database follows the same trend as the AK data subset. There is a high amount of values at -90dB which is the lower limit of the WiFi receiver detectable range. There is a more apparent trend in this figure that AP's on the ground and third floors have more percentage of -90dB values.

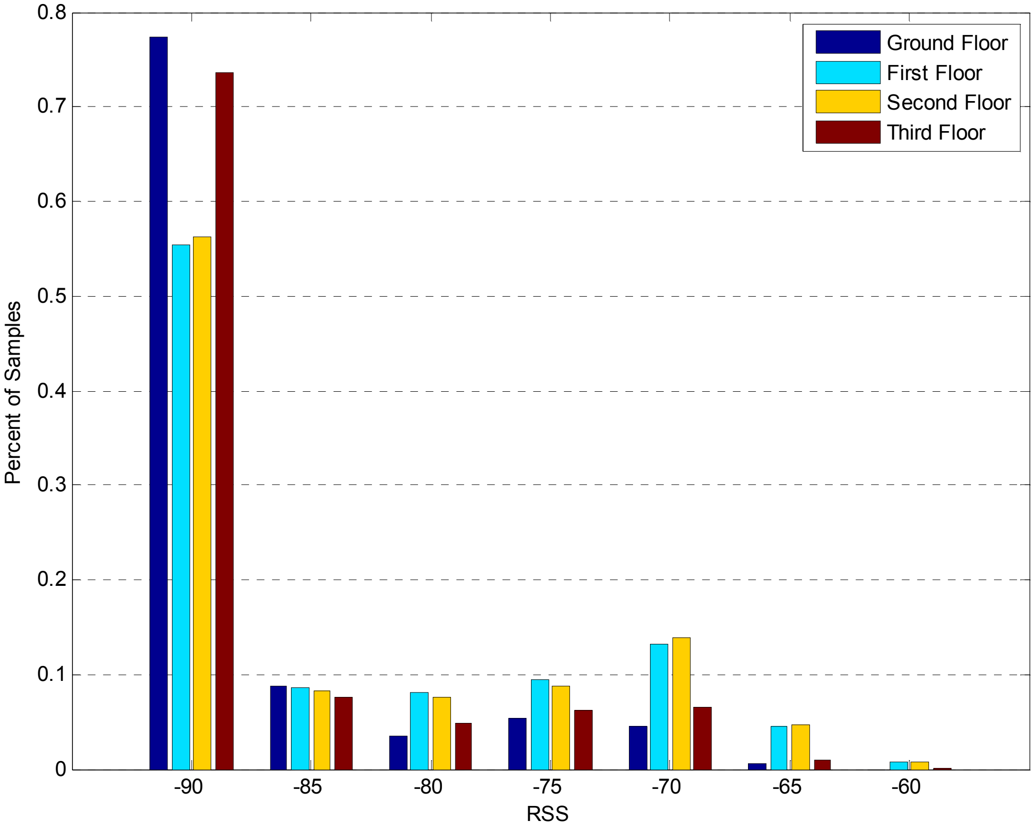


Figure 3-13 Histogram of RSS value showing the distribution in RSS signature database

To characterize the distribution of the RSS signature connectivity segment lengths, three distribution functions were fit to the RSS signature connectivity segment lengths CDF. The functions chosen were exponential, lognormal, and Rayleigh. The exponential distribution was chosen because it describes the distribution of random events. The lognormal distribution was chosen because it has been shown to describe large-scale variations in RSS measurements [13]. Lastly, the Rayleigh distribution was chosen because it has been shown to describe small-scale RSS measurement variations [13]. Table 3-2 gives the CDF formulas for each of the distributions.

Table 3-2 Formulas of distributions for characterization

Distribution	**Definition**	
	CDF	***Parameters***
Exponential	$F(x)=(1-exp(-\lambda x))$	$\lambda > 0$ rate
Lognormal	$F(x)=\frac{1}{2}\left(1+erf\left(-\frac{ln(x)-\mu}{\sigma\sqrt{2}}\right)\right)$	μ - mean σ^2 - standard deviation
Rayleigh	$F(x)=1-exp\left(\frac{-x^2}{2\sigma^2}\right)$	$\sigma > 0$

Figure 3-14 shows the three distributions functions fit to the empirical RSS signature connectivity segment lengths with the best fitting distribution being the lognormal distribution. For completeness, the remaining probabilistic distributions were fit to the data and are show in Appendix B.

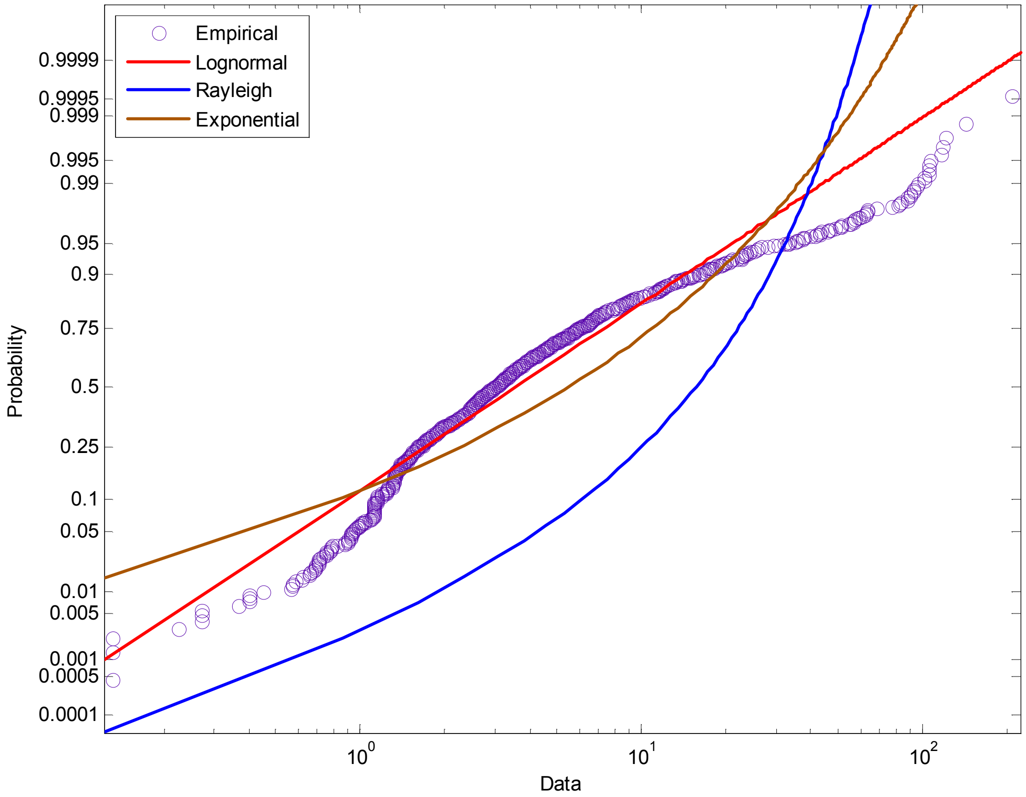


Figure 3-14 CDF of RSS signature connectivity segments fit to probabilistic distributions

Table 3-3 gives the RSS signature attributes for the AP's on the WPI campus. The table indicates that the mean RSS for each AP is relatively low (i.e. near the lower bound of the wireless card sensitivity limit of -90dB). This is to be expected as the measurement were all collected outdoors at distances ranging in the hundreds of meters away the AP's located inside buildings. The mean RSS is also consistent across AP's in difference buildings and floors with a standard deviation of 1.7dB. Additionally from the table it can be seen that an RSS signature on average has 24 connectivity segments (CS) with a mean length of 9 meters.

Table 3-3 AP RSS signature attribute table

AP MAC Address	Building	Floor	Smps	Mean RSS	STD RSS	CS	Mean CS Length	Coverage Distance
00 20 D8 2E 90 00	Atwater Kent	First	263	-87.4	4.9	10	9.34	93.40
00 20 D8 2E 97 40	Atwater Kent	First	1322	-86.0	7.0	24	17.12	410.89
00 20 D8 2E E6 C0	Atwater Kent	First	790	-89.0	3.0	23	12.34	283.92
00 20 D8 2E 8E 00	Atwater Kent	Second	1201	-88.5	3.4	19	20.33	386.31
00 20 D8 2E A9 00	Atwater Kent	Second	915	-84.4	8.5	24	14.67	352.11
00 20 D8 2E F5 40	Atwater Kent	Third	156	-84.4	9.4	26	3.42	88.99
00 20 D8 2E 88 80	Atwater Kent	Third	669	-87.1	6.9	20	10.52	210.42
00 16 CA 32 BD C0	Fuller Labs	Ground	334	-87.1	6.7	20	8.10	162.03
00 16 CA 32 91 40	Fuller Labs	First	505	-85.9	6.9	22	9.38	206.41
00 16 CA 32 BD 80	Fuller Labs	Second	448	-87.4	6.6	22	8.91	195.99
00 16 CA 32 8F 80	Fuller Labs	Second	100	-85.9	8.4	16	3.82	61.10
00 16 CA 32 8E 80	Fuller Labs	Second	22	-87.7	5.8	3	3.59	10.78
00 16 CA 32 82 C0	Fuller Labs	Third	690	-86.4	6.6	117	3.75	438.40
00 20 D8 2E 45 C0	Library	First	28	-90.0	0.0	7	3.14	21.99
00 20 D8 2E EB 40	Library	First	837	-82.5	10.7	23	17.19	395.44
00 20 D8 2E D3 00	Library	Second	552	-89.0	3.0	46	6.67	306.96
00 20 D8 2F 19 80	Library	Second	904	-87.3	7.0	43	9.19	395.02
00 20 D8 2E 51 C0	Library	Third	841	-85.8	9.3	49	6.86	336.31
00 20 D8 2E B8 40	Library	Third	432	-88.1	5.3	42	5.68	238.76
00 20 D8 25 52 80	Higgins Labs	First	92	-86.9	5.4	10	5.36	53.64
00 16 CA 3A 8A 80	Higgins Labs	First	412	-87.3	6.4	20	8.24	164.80
00 20 D8 25 1E 00	Higgins Labs	Second	131	-90.0	0.0	7	6.69	46.84
00 16 CA 38 3D 80	Higgins Labs	Second	218	-86.8	7.0	18	5.74	103.25
00 20 D8 24 37 40	Higgins Labs	Third	465	-85.3	8.2	8	17.85	142.80
00 20 D8 23 DD 80	Campus Center	First	69	-85.1	8.6	2	11.30	22.60
00 20 D8 23 C4 80	Campus Center	Second	314	-83.9	8.7	5	17.26	86.32
00 20 D8 2E CA 00	Campus Center	Second	132	-86.9	6.4	10	6.98	69.81
00 20 D8 23 C5 C0	Campus Center	Third	156	-85.3	9.9	17	5.08	86.35
00 20 D8 23 C6 80	Campus Center	Third	293	-88.9	4.0	24	5.89	141.27
00 16 CA 32 95 40	Olin Hall	First	247	-83.8	7.8	17	6.08	103.32
00 20 D8 25 BD 80	Kaven Hall	Ground	374	-87.8	5.6	21	7.11	149.39
00 20 D8 25 C1 00	Kaven Hall	Ground	366	-87.9	4.8	34	4.69	159.32
00 20 D8 25 B2 40	Kaven Hall	First	745	-86.1	6.5	37	8.23	304.40
00 20 D8 25 F3 80	Kaven Hall	Second	481	-87.1	7.6	32	6.65	212.91
00 15 E8 E6 A7 00	Stoddard Labs	First	220	-82.4	9.5	7	12.26	85.81
00 15 E8 E6 9D 80	Stoddard Labs	Third	565	-86.4	7.4	45	5.99	269.33
00 16 CA 32 BD 00	Stratton Hall	Ground	303	-85.4	9.7	8	13.65	109.17
00 16 CA 32 A3 80	Stratton Hall	First	617	-86.9	6.2	18	11.88	213.79
00 16 CA 32 9E 00	Stratton Hall	First	595	-84.1	10.7	30	7.97	239.09
00 16 CA 32 C3 C0	Stratton Hall	Second	460	-88.2	4.3	19	9.05	171.93
00 16 CA 32 A2 00	Stratton Hall	Third	283	-86.2	7.5	20	7.19	143.89
00 16 CA 32 B9 80	Boynton Hall	First	560	-88.4	5.9	48	5.58	268.00
00 16 CA 32 BA 80	Boynton Hall	Second	402	-86.1	7.8	30	6.12	183.72
00 20 D8 2E A0 40	Bartlett Center	First	559	-85.9	7.5	16	12.51	200.09
00 20 D8 2E B0 00	Bartlett Center	First	407	-88.4	4.3	20	7.25	144.90
00 20 D8 2E 97 80	Bartlett Center	Second	637	-85.6	7.3	30	7.86	235.69
00 20 D8 2E A5 C0	Bartlett Center	Second	620	-87.7	4.7	23	9.62	221.35

Chapter 4: RF Channel Models

4.1 Introduction

In Chapter 4, we present two standard channel models and propose improvements based on the RSS signature characterization performed in Chapter 3. The first model is the single-gradient multi-floor (SGMF) model, which was the recommend for PCS band by the Joint Technical Committee (JTC). The second model is the distance partitioned multi-gradient single-floor (MGSF) model path-loss model, which is the basis of the indoor channel model used for the 802.11 WiFi standard [15]. The proposed improvements to these models are to use site-specific information available from Google Earth$^{\text{TM}}$ to building partition the path-loss by adding a dynamic wall breakpoint and a path loss for the exterior wall of the building.

4.2 Single-Gradient Multi-Floor (SGMF) Model

The single-gradient multi-floor (SGMF) model is recommended for 1900MHz PCS bands by the Joint Technical Committee (JTC) to describe the path-loss in multistory buildings [9]. The idea behind this model is that if the AP and receiver are located on the same floor the path-loss is dictated by the distance from the AP to the receiver using a distance power-gradient. If the AP is located on a difference floor than the receiver a floor penetration loss is added to the distance dictated path-loss. The path-loss in the SGMF model is given by

$$L_p = L_0 + L_f(n) + 10\alpha \log(d) \tag{4-1}$$

where L_0 is the path-loss over the first meter, $L_f(n)$ is the attenuation attributed to each floor, n is the number of floors between the transmitter and receiver, α is the distance-power gradient, and d is the distance between the transmitter and receiver. Table 4-1 gives the set of parameters suggested for three different environments, where L_0 has been adjusted to accommodate the 2.4GHz frequency used by WiFi devices.

Table 4-1 JTC Suggest SGMF Model parameters for residential, office, and commercial environments

Parameter	Description	Environment		
		Residential	Office	Commercial
d	Distance between trans. and recv. (m)	NA		
L_o	Path-loss over first meter (dB)	40	40	40
α	Distance-power gradient	2.8	3	2.2
$L_f(n)$	Path-loss of floors (dB)	$4n$	$15+4(n-1)$	$6+3(n-1)$

Figure 4-1 shows an example for the SGMF model where the AP is located on the third floor of a building and the mobile receiver is located on the first floor.

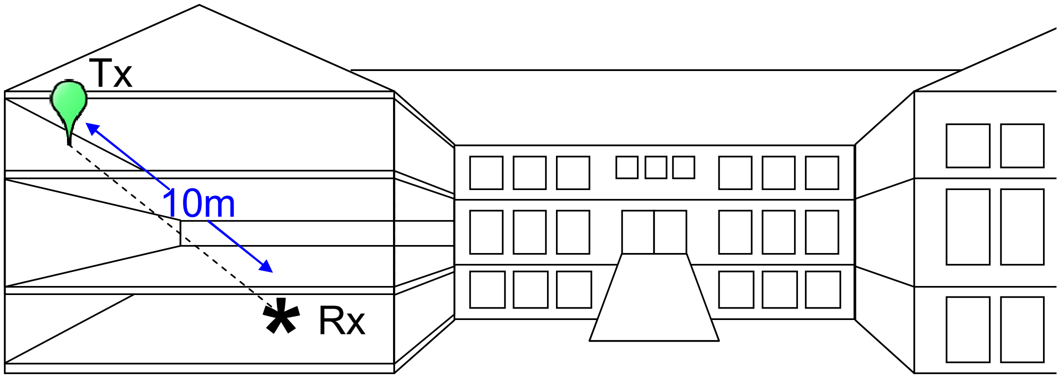


Figure 4-1 JTC model notional diagram showing an AP located on the third floor of a building and the receiver on the first floor

In this example *n* would equal 2 and *d* would be 10m. The SGMF model assumes that the strongest signal path is through the floors between the AP and the receiver. This holds to for most cases when indoors.

4.3 Multi-Gradient Single-Floor (MGSF) Model

The Multi-Gradient Single-Floor (MGSF) model most recently has been used to model the WiFi propagation path-loss in indoor environments. The MGSF model makes use distance partitioning to allow for multiple distance-power gradients to describe the path-loss. The MGSF is the recommend model for the 802.11 standard [9]. The basis for the use of distance partitioning is the assumption that the propagation path-loss from the AP to receiver does not follow a uniform gradient. Given this assumption, the propagation path is partitioned into two sections using a breakpoint distance parameter, d_{bp}. Each section uses a separate distance-power gradient parameter, α_1 and α_2, to characterize the path-loss for each section. Equation (4-2) gives the formula for the distance partitioned MGSF model,

$$L_p = L_0 + \begin{cases} 10\alpha_1 \, log(d) & ;d<d_{dp} \\ 10\alpha_1 \, log(d_{bp}) + 10\alpha_2 \, log(d/d_{dp}) & ;d>d_{dp} \end{cases} \tag{4-2}$$

where L_p is the path-loss over distance *d* in dB, L_0 is the path-loss over the first meter in dB, α_1 and α_2 are the distance-power gradients for the path sections one and two respectively, and d_{bp} is the breakpoint distance in meters. Table 4-2 gives suggested parameter sets for three environments defined for 802.11 standard in reference [10].

Table 4-2 Suggested 802.11 MGSF parameters for residential, office, and commercial environments

Parameter	Description	Environment		
		Residential/ Small Office	Typical Office	Commercial
d	Distance between trans. and recv. (m)	NA		
L_o	Path-loss over first meter (dB)	40	40	40
α_1	Distance-power gradient of section 1	2	2	2
α_2	Distance-power gradient of section 2	3.5	3.5	3.5
d_{bp}	Breakpoint distance (m)	5	10	20

The non-uniform path-loss assumption holds true for most indoor environments. When the receiver is close to the AP, it is likely that they are located in the same room, where the path-loss would be about the same as free space. As the receiver moves farther away from the AP, it becomes more and more likely that there will be structural partitions between the AP and the receiver, which would increase the path-loss. Figure 4-2 shows an example to illustrate the assumption behind the breakpoint in the distance partitioned MGSF model.

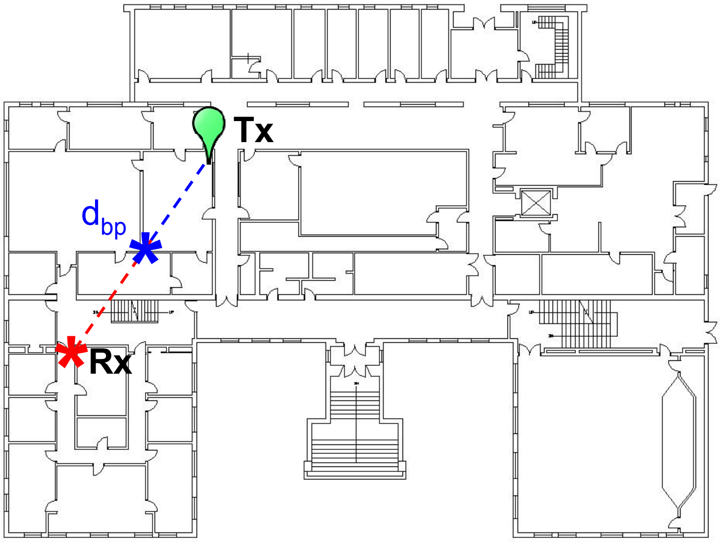


Figure 4-2 Distance Partitioned MGSF Model notional diagram showing the placement of the breakpoint

The example idealizes the breakpoint assumption as the signal does not pass through any physical obstructions up to the breakpoint after which the signal passes through three walls.

4.4 Proposed Site-Specific Channel Models using Google Earth

The current channel models were intended for indoor-to-indoor channel modeling. To be useful for WiFi localization simulations, a channel model must represent the propagation path-loss from an AP inside a building to a receiver outside of a building. The path-loss environment outside a building is free space with trees, lamp posts, signs, and other buildings posing obstacles to the receiver. The small-scale obstacles cause scattering of the propagating wave while the large-scale obstacles like the surrounding buildings cause reflections of the propagating wave, which may interfere constructively or destructively at the receiver [14].

In Section 3.3, it was observed that RSS signatures are influenced by the location of the AP relative to the exterior wall of the building. Neither of the current channel models used any site-specific information to predict the RSS signatures. We hypothesized that adding site-specific information to the current channel models would improve their performance.

Collecting site-specific information through a site survey would be just as costly and time consuming as collecting empirical RSS data. But with the advent of Google Earth^{TM} site-specific information has been become free and readily available. Figure 4-3 shows the aerial view of the WPI campus where the buildings on campus can be clearly seen as well as the pathways and roads. Using Google Earth^{TM} the footprint of each building on campus can be found as also shown in Figure 4-3. The footprints of each building are outlined and stored simply as the corner latitude and longitude coordinates that make up the polygon.

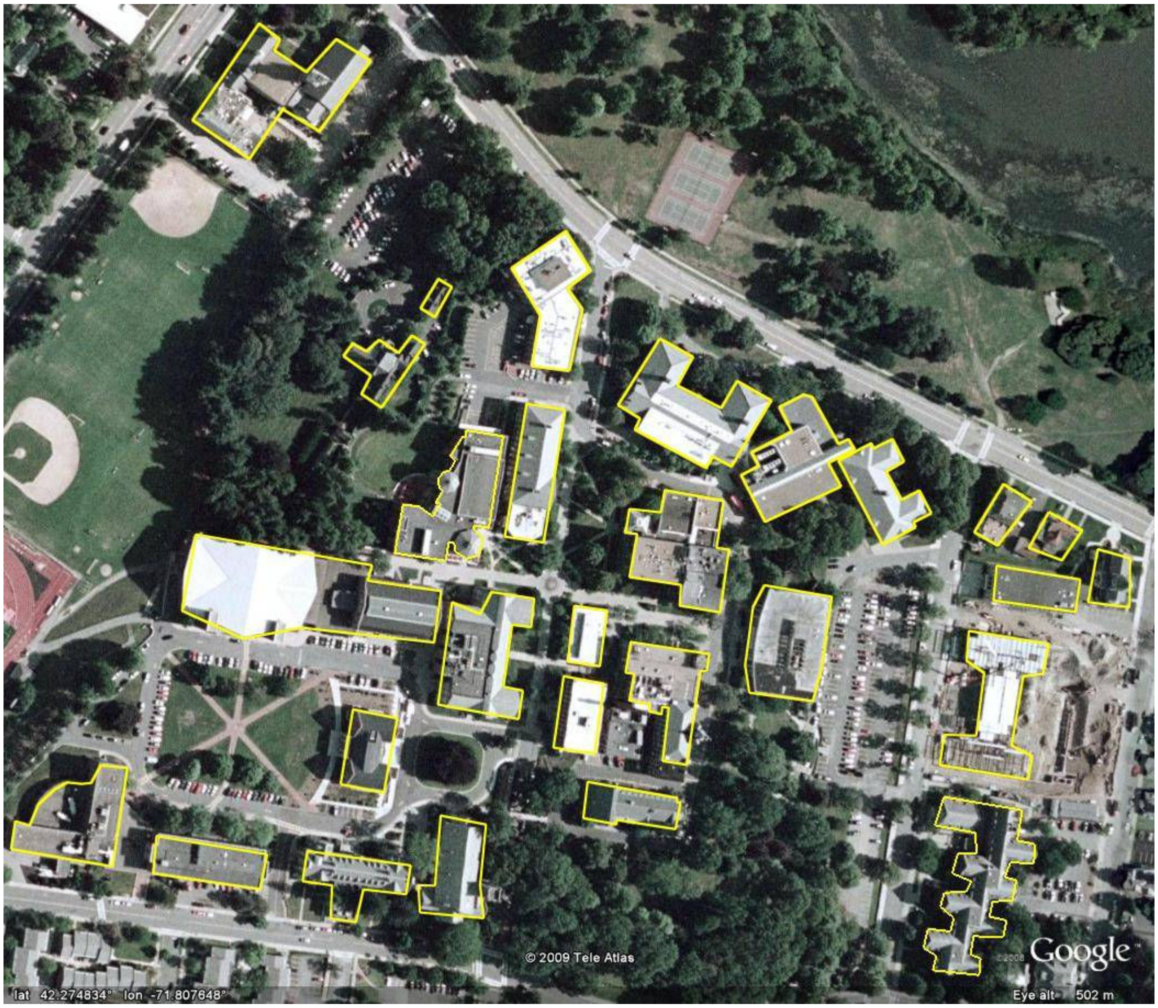


Figure 4-3 Google Earth aerial view of the WPI campus with the building footprints outlined in yellow

With the building footprint available, we propose a building partitioned method to model the path-loss from an AP inside a building to a receiver outside. The building partitioned method is implemented with a dynamic wall breakpoint and exterior wall penetration loss.

The dynamic wall breakpoint is defined by the distance from the AP to the exterior wall of the building. This dynamic parameter uses the building footprint information from Google Earth to calculate the distance for the breakpoint. Figure 4-4 shows a conceptual example of the wall breakpoint locations given the location of the AP, exterior wall, and sample measurement points, *i* to *i+n*.

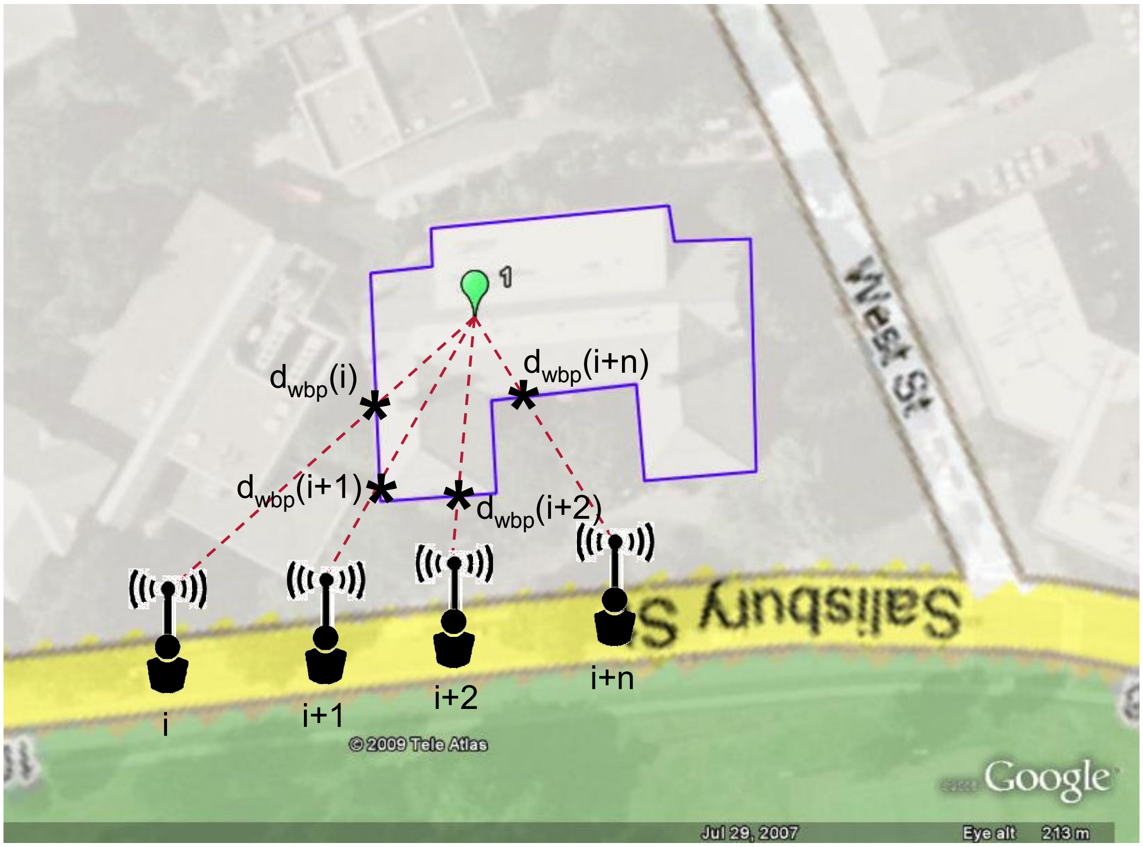


Figure 4-4 Wall breakpoint notional diagram showing the wall breakpoint locations for the measurement location i to i + n

To use the wall breakpoint model in a simulation the exterior walls of the building containing the AP's are outlined in Google Earth to produce the building footprints. The building footprints are used to find the line intersections of the signal path and building walls to define the dynamic breakpoint for each AP and sample.

In addition to the dynamic wall breakpoint, a path-loss for the exterior wall was added to account for the penetration loss of the exterior walls of buildings. The building penetration loss equates to a one-time loss in signal strength. The wall path-loss parameter can be varied depending on the type of buildings in a simulation. Residential buildings have less path-loss than brick commercial buildings. The building penetration loss values can be found in previous research, like reference [13] states that the average path-loss at ground floor-level is 12.8dB at 2.3GHz.

Both of the building partitioned parameters where incorporated in the SGMF and MGSF models.

4.4.1 SGMF Building Partitioned Model

As preciously discussed, the unmodified SGMF model is based on the assumption that the path-loss is solely a function of the distance from the AP to the receiver and the floors between. During the characterization of the RSS signatures it was shown that the location of the AP relative to the exterior of the building also influences the RSS signature, therefore two improved models are proposed. First we add a dynamic wall breakpoint to the SGMF to produce a model we denote SGMF+BP. The formula for the SGMF+BP model is given by

$$L_p = L_0 + L_f(n) + 10\alpha_1 \, log(d_{wbp}) + 10\alpha_E \, log\left(\frac{d}{d_{wbp}}\right) \tag{4-3}$$

where L_p is the path-loss over distance d in dB, L_0 is the path-loss over the first meter in dB, $L_f(n)$ is the attenuation attributed to each floor, n is the number of floors between the transmitter and receiver, α_1, and α_E are the distance-power gradients for the respective path sections, and d_{wbp} is the dynamic AP specific wall breakpoint in meters.

We then add the exterior wall penetration loss to the SGMF+BP to produce a model we denote as SGMF+BPWL. The SGMF+BPWL formula is given by

$$L_p = L_0 + L_f(n) + 10\alpha_1\, log(d_{wbp}) + 10\alpha_E\, log\left(\frac{d}{d_{wbp}}\right) + L_W \tag{4-4}$$

where L_w is the path-loss for the exterior wall in dB.

4.4.2 MGSF Building Partitioned Model

The distance partitioned MGSF model is not ideal for modeling indoor-to-outdoor scenarios due to its basic assumption not holding true. The assumption holds true up to the exterior wall of the building, but after that the path-loss environment changes to free space with trees, lamp posts, signs, and other buildings posing obstacles to the receiver.

The wall breakpoint model extends the distance partitioned MGSF model to allow for proper simulation of the path-loss environment outside of buildings. It was implemented by adding a second dynamic breakpoint, d_{wbp}, with an associated distance-power gradient, α_E, for the path section outside the exterior wall of the building. We denoted this improved model as MGSF+BP and its formula is given by

$$L_p = L_0 + \begin{cases} 10\alpha_1\, log\left(d_{wbp}\right) + 10\alpha_E\, log\left(d / d_{wbp}\right) & ;d_{bp} > d_{wbp} \\ 10\alpha_1\, log\left(d_{bp}\right) + 10\alpha_2\, log\left(d_{wbp} / d_{bp}\right) & \\ \qquad + 10\alpha_E\, log\left(d/d_{wbp}\right) & ;d > d_{bp} \end{cases} \tag{4-5}$$

where L_p is the path-loss over distance d in dB, L_0 is the path-loss over the first meter in dB, α_1, α_2, and α_E are the distance-power gradients for the respective path sections, d_{bp} is the static breakpoint distance in meters, and d_{wbp} is the dynamic AP specific wall breakpoint in meters.

We then add the exterior wall penetration loss to the MGSF+BP to produce a model we denote as MGSF +BPWL. The MGSF +BPWL formula is given by

$$L_p = L_0 + \begin{cases} 10\alpha_1\, log\left(d_{wbp}\right) + 10\alpha_E\, log\left(d/d_{wbp}\right) + L_w & ;d_{bp} > d_{wbp} \\ 10\alpha_1\, log\left(d_{bp}\right) + 10\alpha_2\, log\left(d_{wbp} / d_{bp}\right) & \\ \qquad + 10\alpha_E\, log\left(d/d_{wbp}\right) + L_w & ;d > d_{bp} \end{cases} \tag{4-6}$$

where L_w is the path-loss for the exterior wall in dB.

Chapter 5: Channel Model Performance Analysis

5.1 Introduction

In Chapter 5, we evaluate performance of the channel models discussed in Chapter 4 to predict the WiFi RSS signatures characterized in Chapter 3. We begin by describing the framework used for channel model simulation and evaluation. We then define a criterion to measure the performance of each channel model and then use this criterion to optimize the parameters for the wall breakpoint models. Finally we evaluate the performance of each channel model.

5.2 RSS Signature Simulator and Evaluator

To evaluate the channel models to reproduce the RSS signatures an RSS signature simulator and evaluator was needed. MATLAB was used to create a simulation environment to implement the channel models and evaluate their performance. Figure 5-1, shows the flow and components of the channel model simulator.

The empirical RSS signature database described in Section 3.2, was used for the measurement path coordinates. It was important to use the coordinates where the empirical data was collected for in the performance evaluation of the channel models. The building footprint database was built from site-specific information exported from Google Earth as described in Section 4.4, was used to calculate the wall breakpoint distance. The AP database was supplied by WPI and contained the AP latitude and longitude coordinates which were used to calculate the distances between the AP's and the measurement points. Using the three databases the two distance parameters were calculated and input into the channel model. The channel model RSS signatures output were stored in a model RSS signature database.

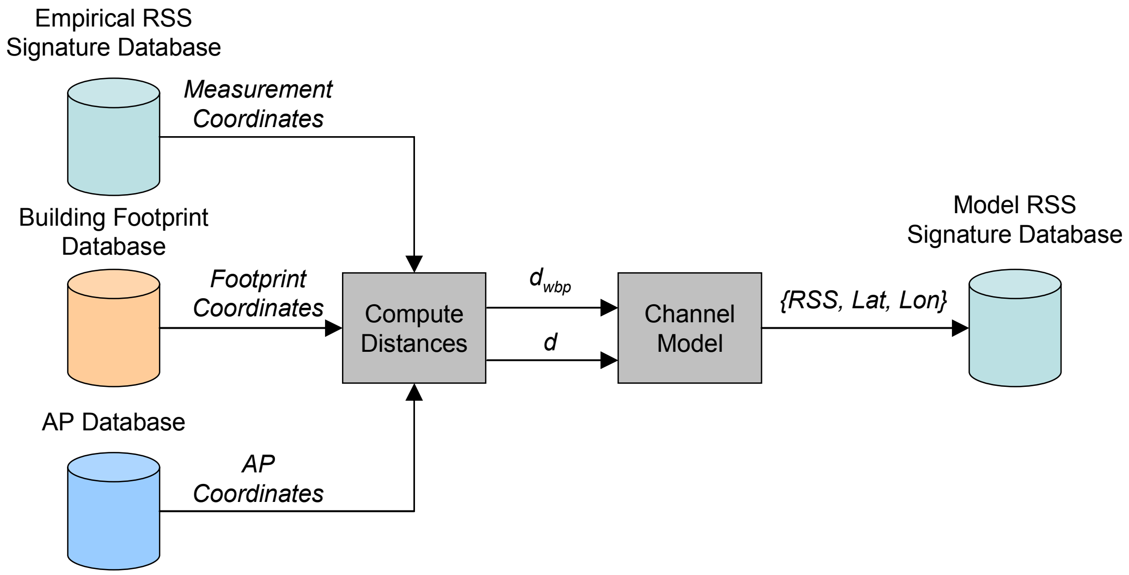


Figure 5-1 Model simulation flow diagram

The RSS signature evaluator was created to implement the performance evaluation criteria, *K*, which is defined in the next section. Figure 5-2 shows the components and flow of the performance evaluator.

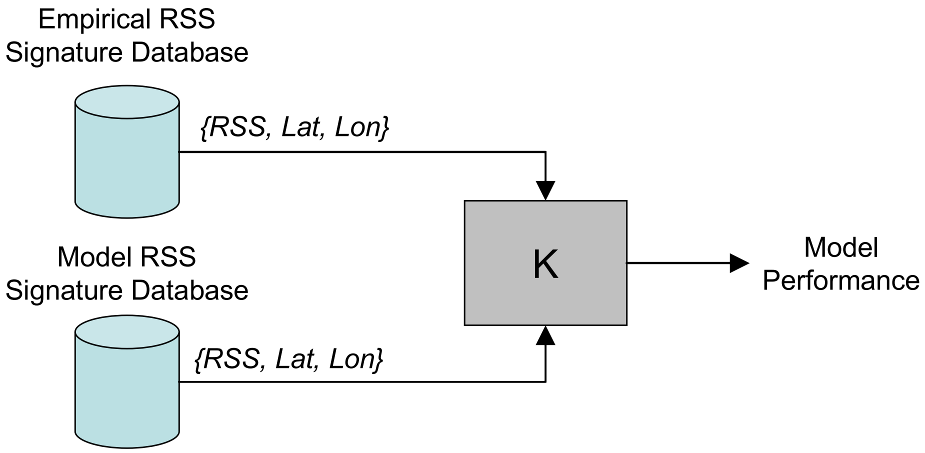

Empirical RSS Signature Database
{RSS, Lat, Lon}
Model RSS Signature Database
{RSS, Lat, Lon}
K
Model Performance

Figure 5-2 Performance evaluation flow diagram

Using the results of the RSS signature simulator and the empirical RSS signature database the evaluator computes the performance of the channel model.

5.3 Definition of Performance Evaluation Criteria

To evaluate the performance of the channel models, binary hypothesis testing was employed to measure their ability to accurately produce the empirical RSS signatures. To use binary hypothesis testing, the model results and empirical data were treated as two binary sets, *D* and *H*, in the sample space, *S*. The sample space consisted of the locations where RSS measurements were recorded. For each AP there was an empirical set consisting of the locations where a signal was detected, *H*. Similarly, for each AP the channel model produced a set consisting of the locations where a signal was predicted, *D*.

In binary hypothesis testing, the empirical data are used to find the probabilities of the possible outcomes. There are four possible outcomes for each sample: the model predicts a detection and the empirical data shows a detection (D_1, H_1), the model predicts a detection and the empirical data does not show a detection (D_1, H_0), the model does not predict a detection and the empirical data does not show a detection (D_0, H_0), and last, the model does not predict a detection and the empirical data does show a detection (D_0, H_1). Given these four possible outcomes, Bayes' theorem (4) was used to find the likelihood of correct model predictions [16].

$$P(D \mid H) = \frac{P(H \mid D)P(D)}{P(H)} = \frac{P(D \cap H)}{P(H)} \tag{4}$$

In Equation (5), the two correct model prediction likelihoods were averaged and used as the performance metric, *K*. The performance metric was use to evaluate each channel model's ability to match the empirical RSS signatures.

$$K = \frac{P(D_0|H_0) + P(D_1|H_1)}{2} \tag{5}$$

5.4 Building Partitioned Modeling Parameter Optimization

The distance-power gradient for the path section from the exterior wall to the receiver, α_E, is unknown. We hypothesize that it should be greater than 2, which is the distance-power gradient for free space and 3.5 which is the gradient used by 802.11 for the second path in the MGSF model. Before the performance of the building partitioned models was able to be evaluated these optimized value for the distance power-gradient needed to be found. The performance metric defined in Section 5.1, was calculated over a range of 2-10 for the distance-power gradient for the building partitioned models. For the remaining parameters typical office environment values were used as defined in Chapter 4.

The performance of the each model was plotted over the range of distance-power gradients (see Figure 5-3).

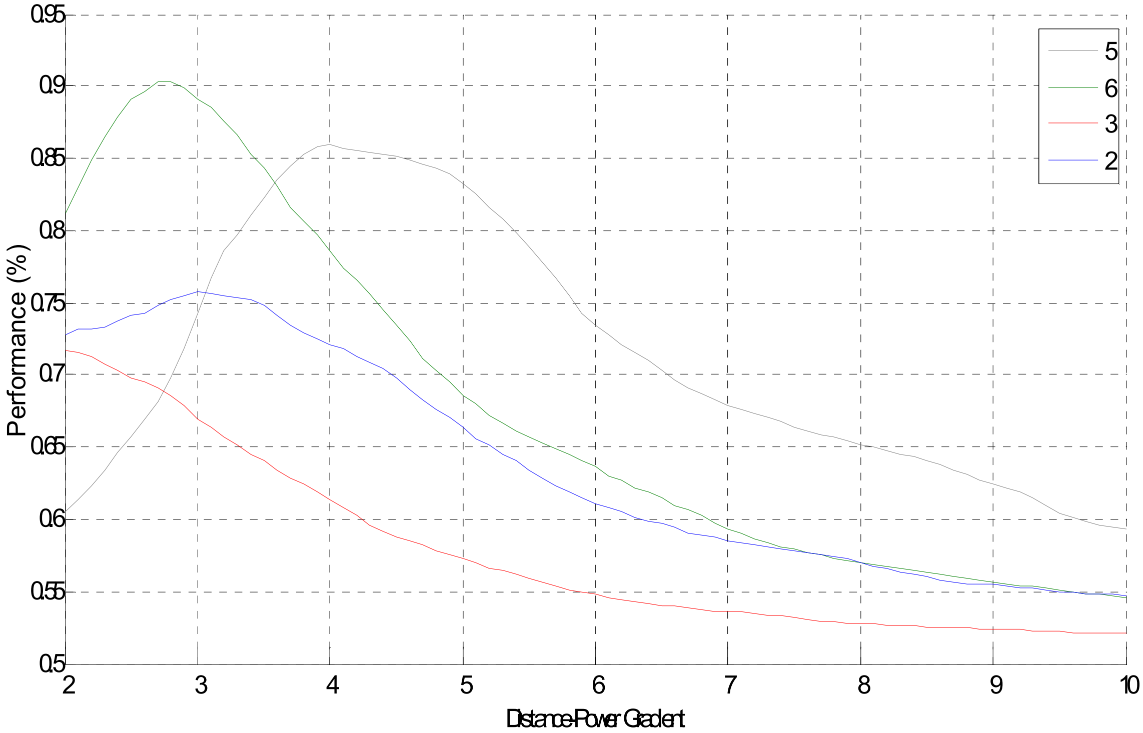


Figure 5-3 Building partitioned models parameter optimization

From the above figure the optimal distance-power gradients were found. Table 5-1 shows the parameter values used for each model.

Table 5-1 Building partitioned models parameters

ID	Model	d_{bp}	d_{wbp}	α_1	α_2	α_E	$L_f(n)$	L_W
1	SGMF (JTC)			3.0			15+4(n-1)	
2	SGMF+BP		Wall	3.0		3.0	15+4(n-1)	
3	SGMF+BPWL		Wall	3.0		2.0	15+4(n-1)	12.8
4	MGSF (802.11)	5		2.0	3.5			
5	MGSF+BP	5	Wall	2.0	3.5	4.0		
6	MGSF+BPWL	5	Wall	2.0	3.5	2.8		12.8

The optimal power-gradients range between 2.0 and 4.0 for the model variations. The optimal external path distance-power gradient for the SGMF+BP model was the same as the internal path. This indicates that adding a building partitioned distance-power gradient to the SGMF model will not improve its performance. The SGMF+BPWL however, had an optimal external distance-power gradient of 2 or free space, indicating that adding a path-loss for the exterior wall of the building has an effect of the power gradient.

The MGSF+BP model's distance-power gradient was larger than the internal path distance-power gradient, which does not fit with the known path-loss environment. The interior paths should have higher path-loss due to interior wall and other physical obstructions. Whereas the exterior path should be closer to free space path-loss as discussed in Section 4.4. The distance-power gradient for this model is most likely artificially high due to the absence of the exterior wall path-loss and should result in lower performance than the other model with the exterior wall path-loss. The MGSF+BPWL model which includes path-loss for the exterior wall of the building has a lower distance power gradient for the exterior path than the interior path. This was the expected case as discuss above and should show better overall performance.

5.5 Performance Analysis Results

Using the simulator and evaluator the performance of the existing and proposed models were evaluated. The parameters used for each model as the same as listed in Table 5-1. Each of the building partitioned models where compared back to their respective base model, either the SGMF (JTC) or the MGSF (802.11). Table 5-2 gives the results for each model.

Table 5-2 Channel model performance results

ID	Model	Opt Metric		
		Mean	Max	Min
1	SGMF (JTC)	79.7%	98.4%	50.5%
2	SGMF+BP	79.7%	98.4%	50.5%
3	SGMF+BPWL	74.1%	96.2%	49.7%
4	MGSF (802.11)	82.0%	88.3%	51.9%
5	MGSF+BP	86.0%	91.9%	55.2%
6	MGSF+BPWL	91.2%	96.3%	62.7%

From the results it can be seen that both of the current base models have similar performance. The SGMF model had a higher peak performance but the MGSF model had a slightly higher mean performance.

The SGMF+BP model, which added the wall breakpoint to the SGMF model, showed no improvement to performance. This was expected as the optimal value for the exterior distance-power gradient was found to be the same as the interior gradient. The SMGMF+BPWL, which added both a wall breakpoint and wall penetration loss to the SGMF model, showed a decrease in performance over the base model. It is hypothesized that the decrease in performance is because the floor path-loss included in the original SGMF model is not present for the indoor to outdoor signal path being modeled. This is because the assumption from the indoor to indoor scenario that the signal passes through the number of floors between the AP and receiver does not hold true for the indoor to outdoor scenario. If they are both indoors the strongest signal path is most likely through the floors between the AP and receiver, but if the receiver is outdoors then the strongest signal path is more likely to be through the exterior wall for example through a window. The floor path loss in the original model is most likely accounting for the wall path loss for the indoor to out door scenario and therefore with the addition of the wall penetration loss parameter the wall path loss is being double counted. A solution to this problem could be to modify how the number of floors between the AP and receiver is calculated. For the WPI campus the floor of the AP is not a significant driver to the RSS values but for environments with building with more floors this parameter may become useful.

Opposed to the SGMF base model, the MGSF base model showed improved performance when the building partitioning parameters were added. The addition of the wall breakpoint showed a 6% increase, but when the wall path-loss was added the performance increased by 10%. The results follow the expectation that the MGSF+BPWL model with path-loss for the exterior wall of the buildings would outperform the models without the wall path-loss.

Given that the MSGF+BPWL model performance the best out of the building partitioned models it was used in the further analysis of the model predictions. Figure 5-4 is a plot showing the performance of the two base models and the best performing building portioned model, MGSF+BPWL over all of the AP’s in the database.

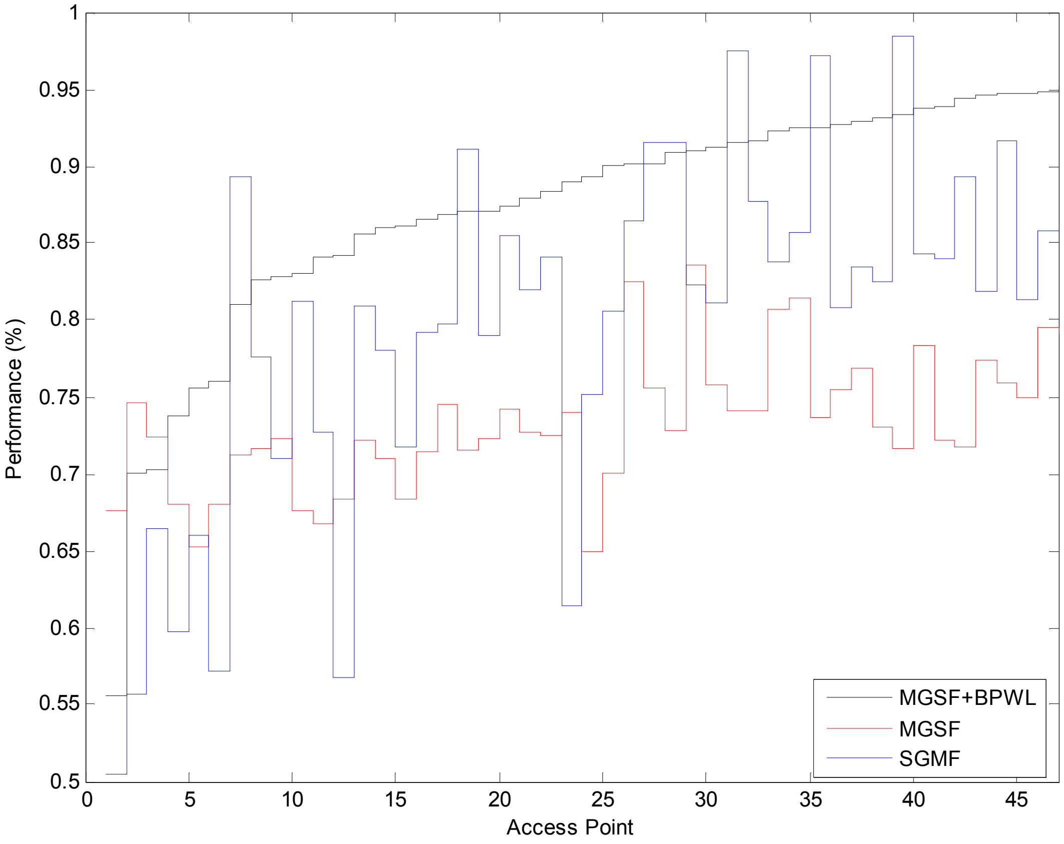


Figure 5-4 Channel model performance comparison over AP's

The figure shows that the building portioned model outperforms the base models for 38 out of the 47 AP's.

To understand the models predicted RSS values in comparison to the empirical data the mean RSS for each AP was found. Figure 5-5 is a plot of the mean RSS for empirical measurement, the two base models, and the best performing building partitioned model, MGSF+BPWL, for each AP in the database.

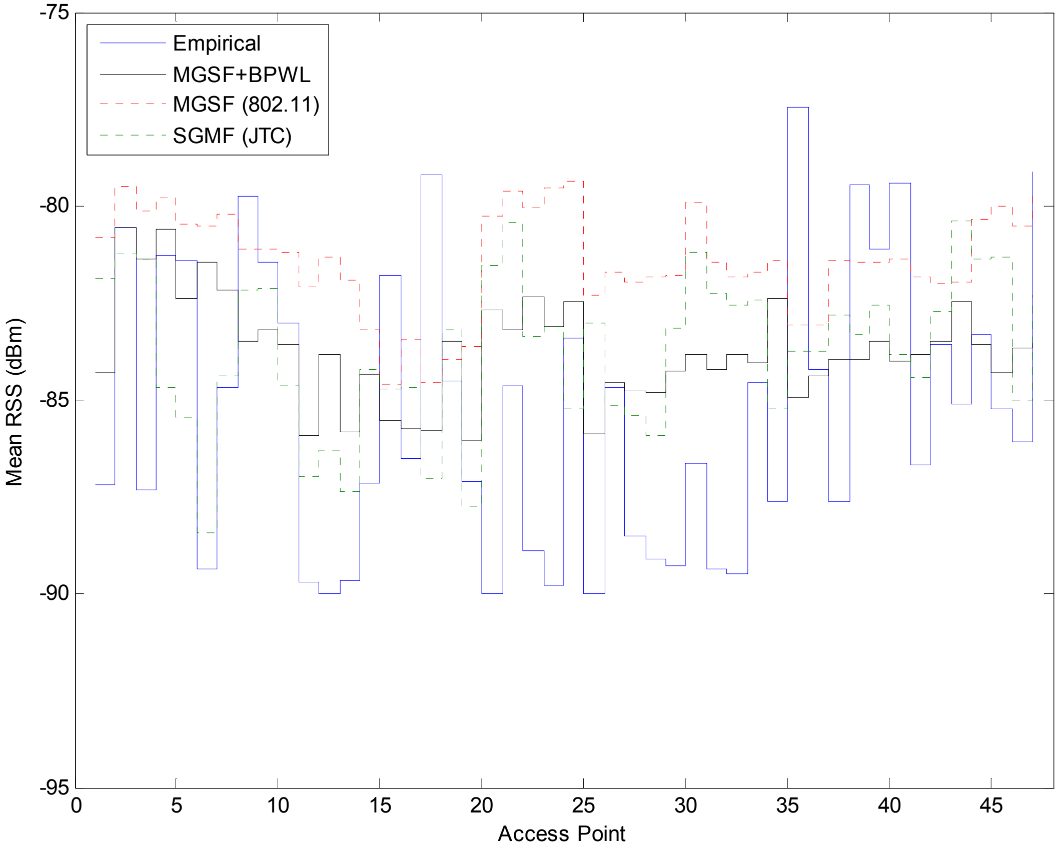


Figure 5-5 Channel model mean RSS comparison over database AP's

The plot shows that the models on average over predicted the mean RSS for the AP's. The empirical RSS mean values also show more variation between AP's than the models.

Table 5-3 gives the mean difference for the channel models from the empirical RSS measurements.

Table 5-3 Mean RSS channel model difference from empirical measurements

		Difference from Empirical (dB)	
ID	Model	Mean	STD
1	SGMF (JTC)	3.4	3.8
4	MGSF (802.11)	4.7	3.9
6	MGSF+BPWL	3.3	3.8

The SGMF and MGSF+BLWL show similar performance to match the mean RSS of the empirical measurements. In general the difference from the empirical mean RSS's are not significant given that the RSS variation from shadow fading is 5dB.

An additional parameter for interest is the coverage distance predicted by each AP. The coverage distance is directly related to the RSS signature. Figure 5-6 show the coverage length of the each AP for the empirical measurements and the model predictions.

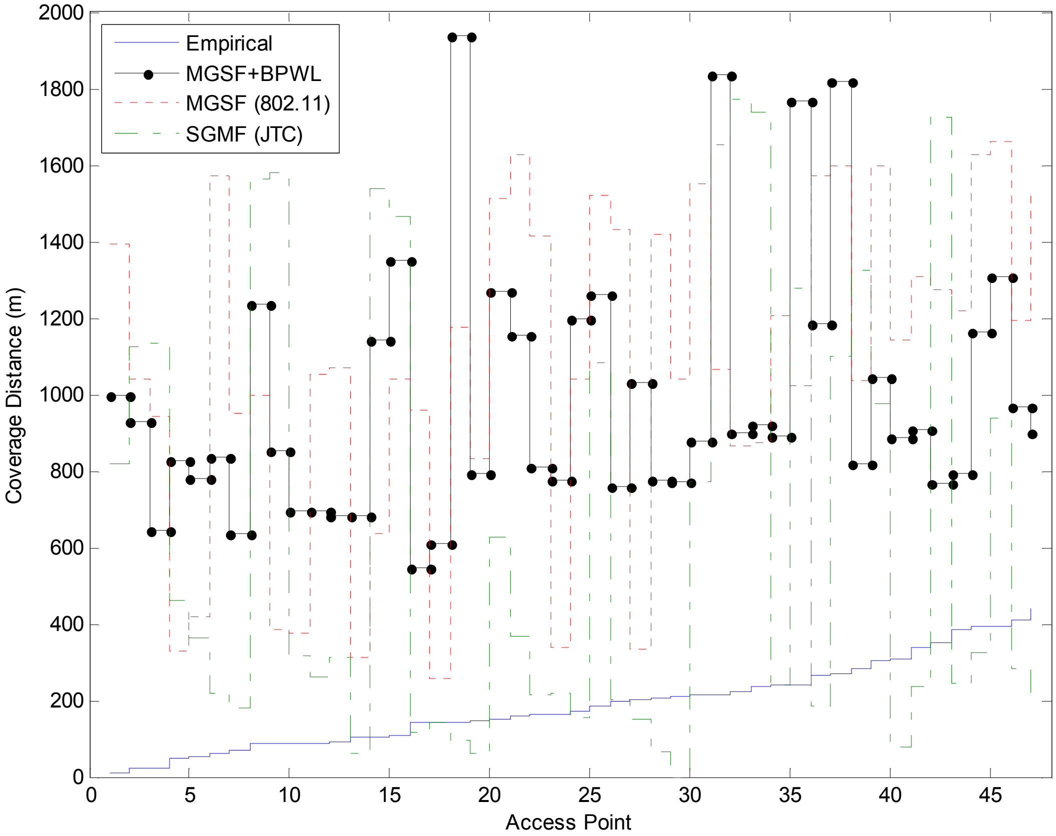


Figure 5-6 Coverage distance model prediction to empirical data comparison for each AP

From the plot it can be seen that the all three of the models do not predict the AP coverage distance of the RSS signature accurately. The errors from the empirical distance are on the order of hundreds of meters as given in Table 5-4.

Table 5-4 Model coverage prediction difference from the empirical data

		Difference from Empirical (m)	
ID	**Model**	**Mean**	**STD**
1	SGMF (JTC)	517	590
4	MGSF (802.11)	890	381
6	MGSF+BPWL	798	328

The predication error is hypothesized to be due to the buildings surrounding the buildings containing the AP's. These surrounding buildings serve as obstacles causing path-loss that are not accounted for in any of the models. The over predicted coverage could also be attributing to the higher mean RSS predicted by the models. To overcome this short fall the footprints of the surrounding building could be added to future models.

The distribution of the connectivity segment lengths of the models was also compared to the empirical data. Figure 5-7 shows the CDF of the empirical, SGMF model, MGSF model and MGSF+BPWL model RSS signature connectivity segments.

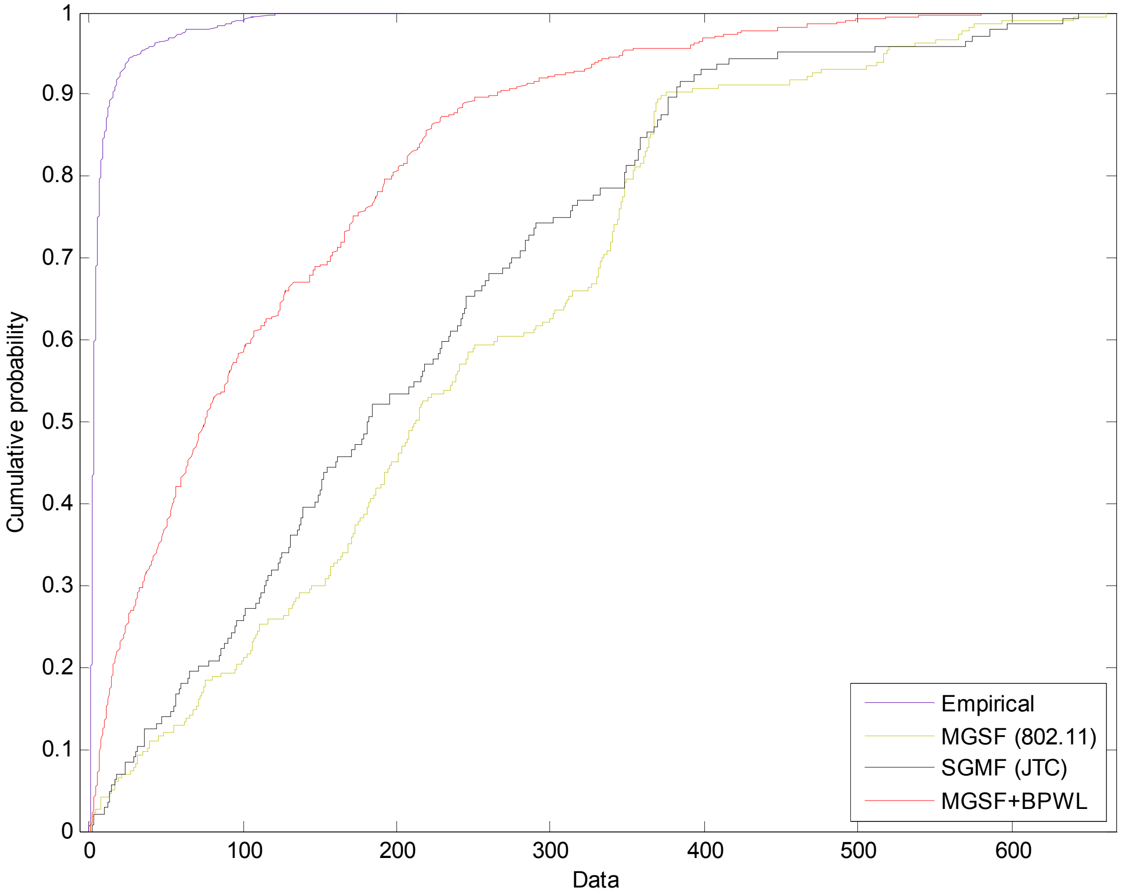


Figure 5-7 CDF of empirical and model RSS signature connectivity segments

From the plot it can be seen that the MGSF+BPWL model creates RSS signature connectivity segments which have a tighter fit to the distribution of the empirical data.

To complete the evaluation of the models we used the AK data subset discussed in Section 3.3. Figure 5-8 is a plot of the empirical and model RSS signatures for compassion on the measurement path around AK for each AP. The first observation made was that the MGSF+BPWL model out performed the MGSF (802.11) for all of the AP's. But this was not the case for the SGMF (JTC), the MGSF+BPWL model out performed the it for the first floor AP's (1-3) but had the same performance for the 2nd floor AP's (5-6) and performed worse on the third floor AP's (7-8).

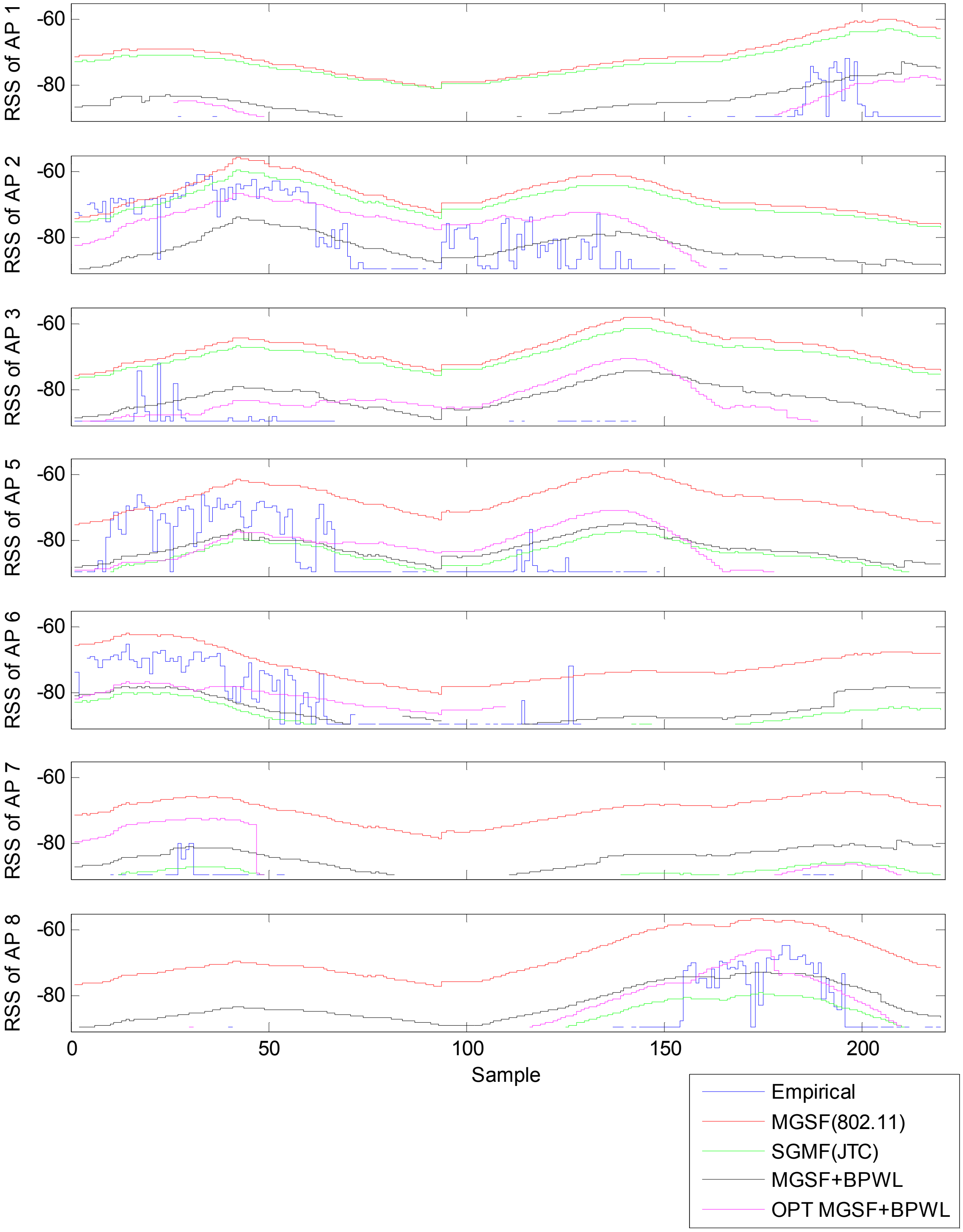


Figure 5-8 Comparison of empirical and model RSS signatures for the AK AP's on the database subset defined in Figure 3-3

Given this observation we see that MGSF+BPWL model has room for improvement. As an exercise we tried to optimize the MGSF+BPWL model parameters to see if we could improve it performance. We found that if we let $\alpha_1 = 5$ and $\alpha_E = 2.8$ we could improve the performance for the AK subset to ~90%. This RSS signature is denoted as OPT MGSF+BPWL in Figure 5-8.

Chapter 6: Conclusions and Future Work

6.1 Summary and Conclusions

WiFi localization algorithm development requires a large database of RSS signature information. There are two methods to build the RSS database: empirically through measurements or theoretically using channel models. The empirical method is not preferred for research and development of WiFi localization algorithms because it is costly, time consuming, and the environment scenarios are inflexible. Therefore the theoretical method using an accurate channel model is preferred.

In support of using the theoretical method to build an RSS database we characterized the RSS signatures on the WPI campus using a significant empirical RSS database. From this characterization we found the following. The RSS signatures were composed of connectivity segments whose length's best fit a lognormal distribution. The detected mean RSS values were on the lower sensitivity limit of the WiFi receiver. There was not a tight relationship between the distance from the AP to the measurement point and the RSS value. The RSS values of AP's located on ground to third floors followed the same distribution. Finally it was observed that the RSS signature behavior was affected by the location of the AP within the building in respect to the distance from the exterior wall.

From the results of our RSS signature characterization we proposed a new building partitioned modeling method. The building partitioned method made use of site-specific information readily available from Google Earth to find the coordinates of the building footprint conation the AP. Using the building footprints the new models were able to account for the location of the AP relative to the exterior wall of the building.

Lastly, we evaluated the performance of two existing models (JTC and 802.11) along with the proposed building partitioned models against the empirical data. The evaluation showed that by adding site-specific information to the channel models a 10% performance increase is gained.

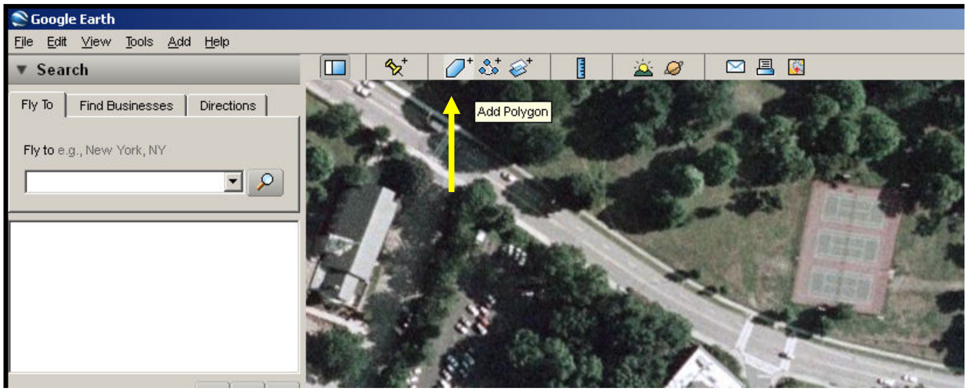
The work performed in this research is the first steps to allow researchers in WiFi localization to evaluate algorithm performance and study environmental and AP distribution effects without the need to collect vast amounts of empirical data.

6.2 Future Work

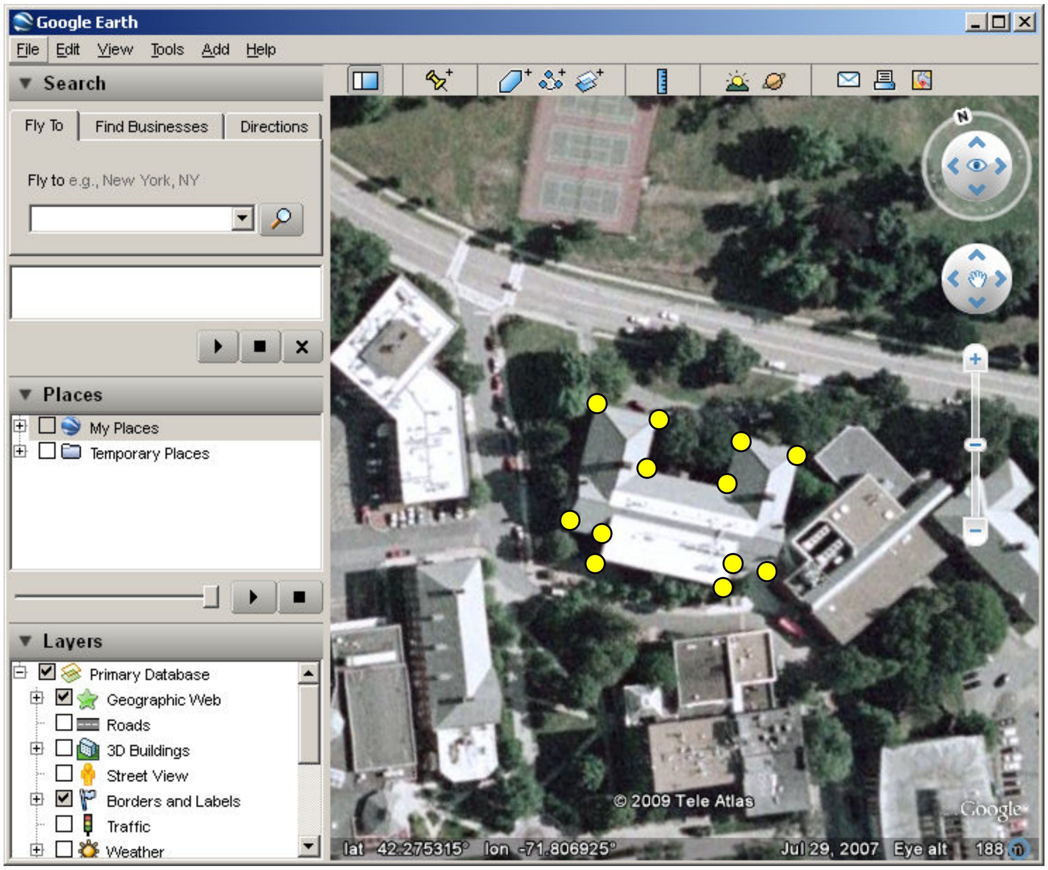
The research presented in this thesis leaves several areas for future work, including continued model improvements, environment characterization, AP density and distribution analysis, and algorithm performance evaluation. In the area of channel model improvements, more site-specific information is available that was not employed in this initial research like accounting for the neighboring buildings in the path-loss model. In the area of environment characterization, we only analyzed empirical RSS signatures in a campus environment. It is unknown whether these signatures behave significantly different in environments such as residential and urban canyon areas. Based on these environment characterizations there may be further research needed to design and evaluate models for theses different environments. In the area of AP density and distribution, the channel models studied in this research can be used to simulate RSS signatures for varying AP densities and distributions (i.e. random and grid) to understand the effects of localization algorithms. The last area of future work would be to use the channel models to evaluate the performance of different localization algorithms such as centroid, nearest-neighbor, and particle filter. The resulting locations could be compared back to the actual and empirical locations to understand the effect of using channel models on the localization algorithms. The goal would be to minimize the difference between the empirical and the model predicated locations.

Appendix A Exporting Building Footprints from Google Earth

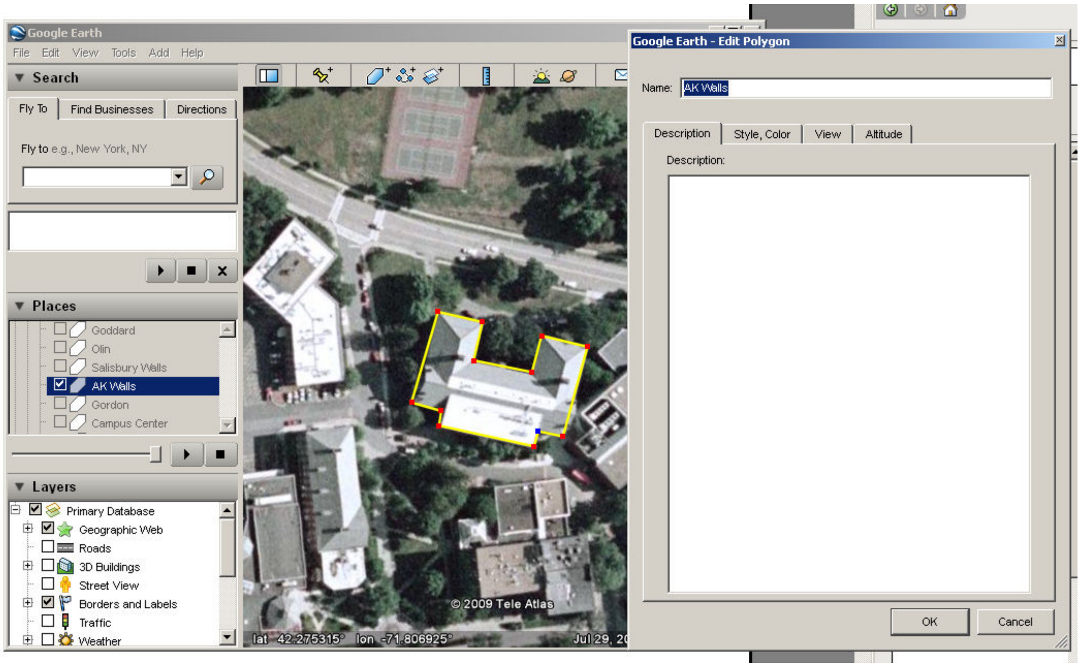
1. Click on “Add Polygon”



2. Click on the corners of the building to be outlined in consecutive order

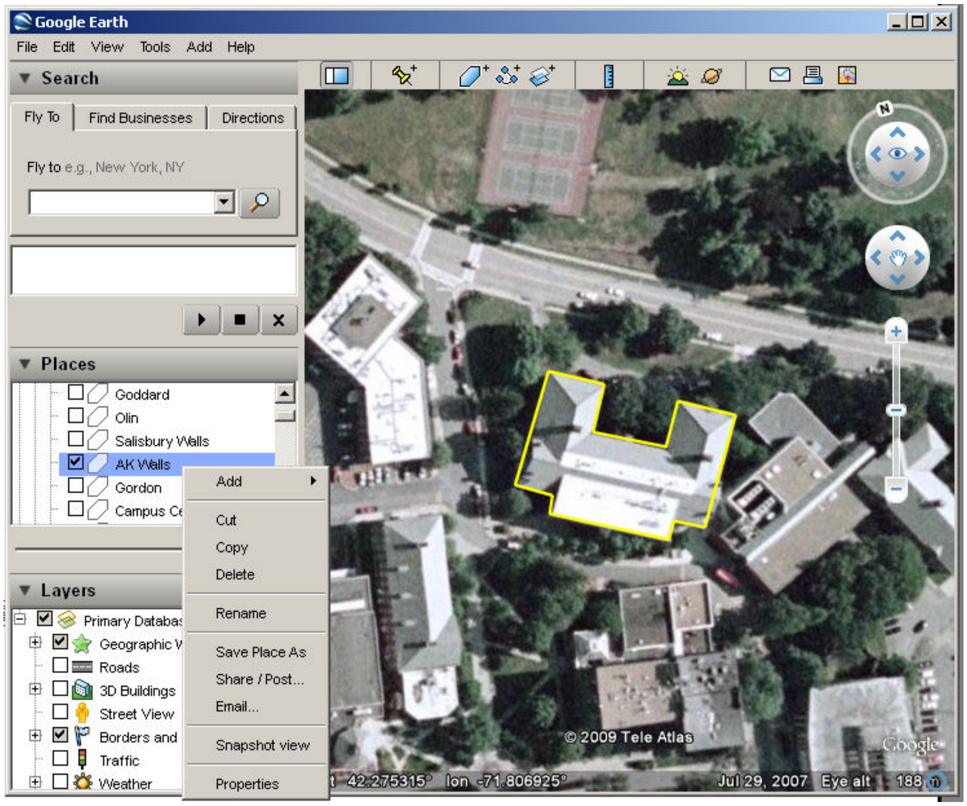


3. Enter a name for the polygon and click "OK"

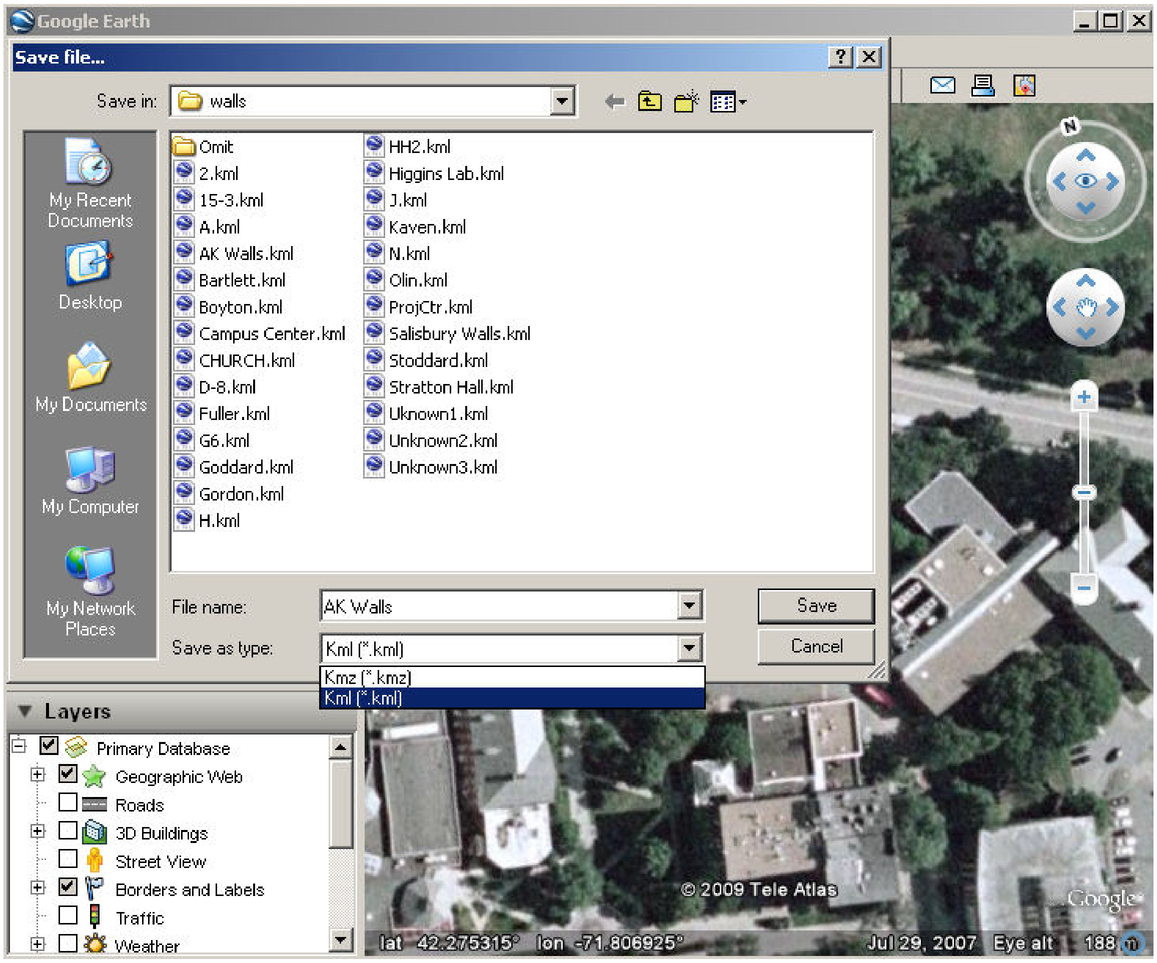


Name: AK Walls

4. Right click on the polygon in the "Places" area and click "Save Place As"



5. Save the polygon as a Kml type



Save in: walls

File name: AK Walls

Save as type: Kml (*.kml)

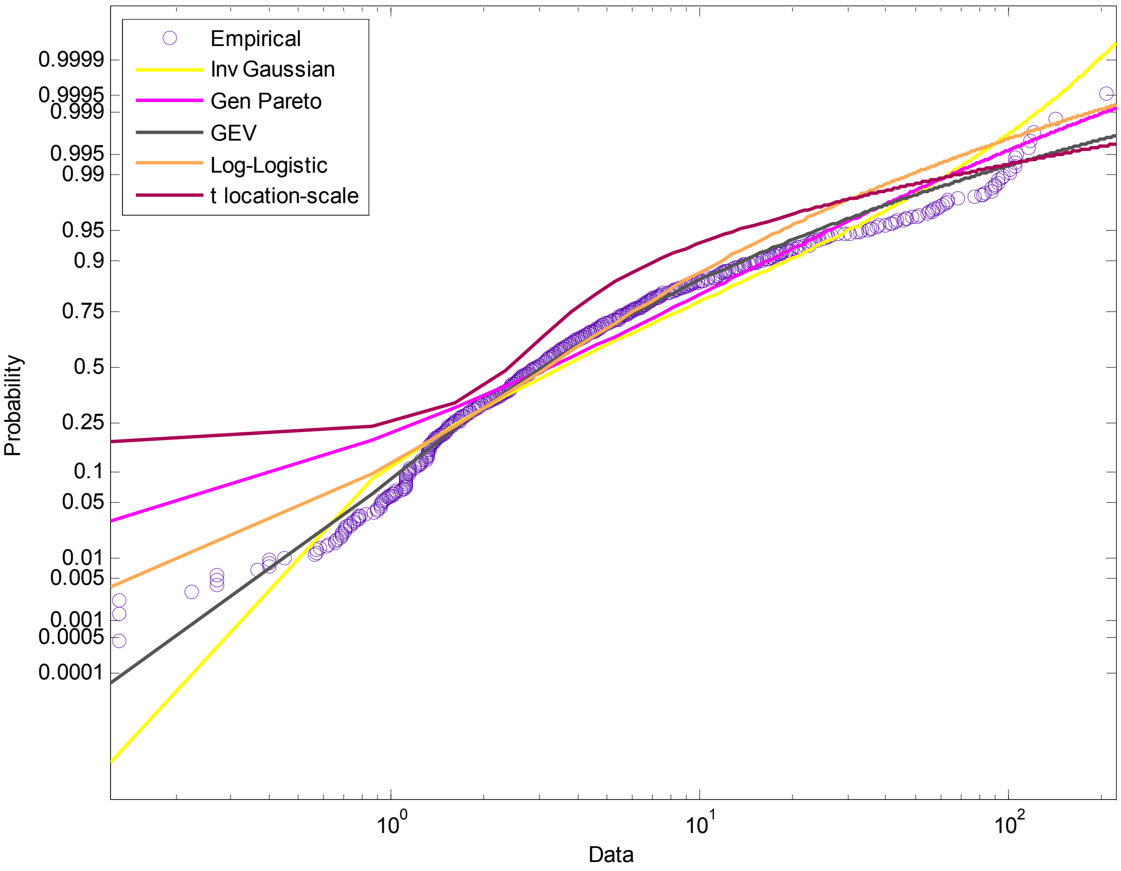
6. The Kml file containing the building footprint can now be read into MATLAB using the Google Earth Toolbox.

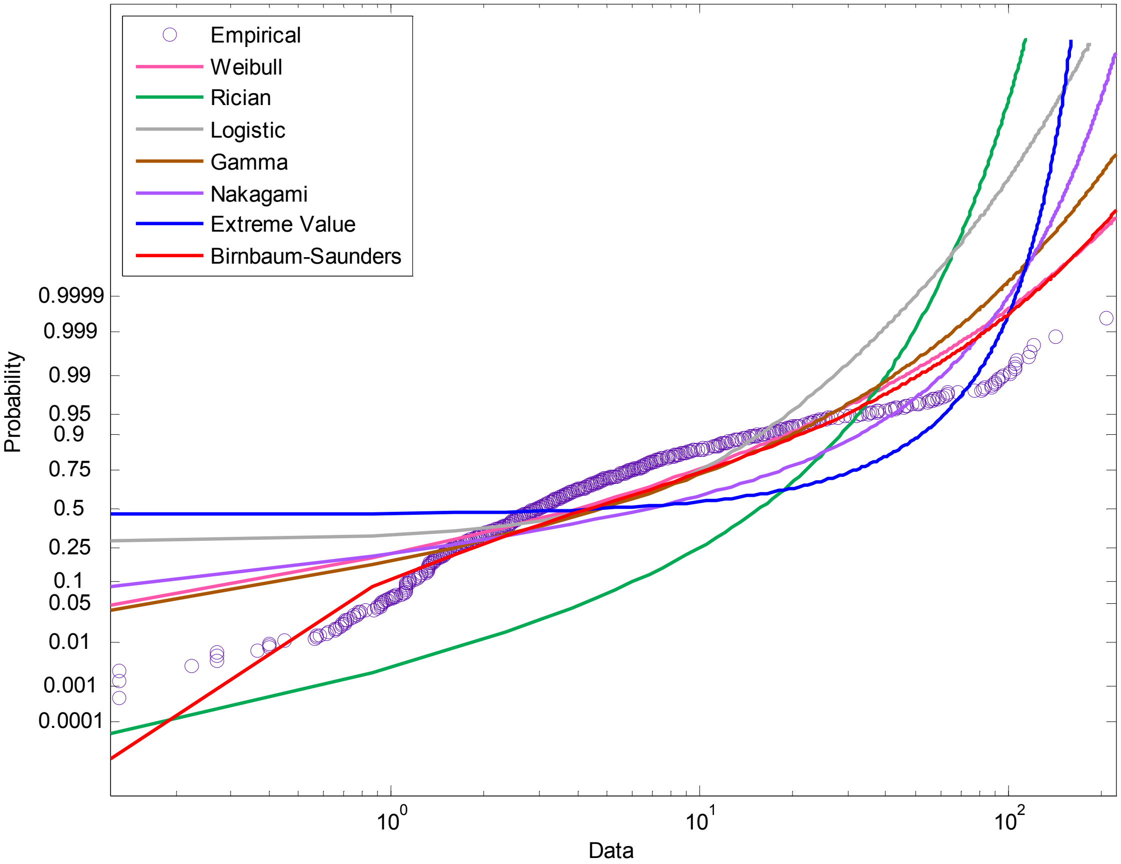
AP Database

AP MAC Address	Building	Floor	Longitude	Latitude	SSID
00 20 D8 2E 90 00	Atwater Kent	First	71°48'24.67"W	42°16'30.54"N	WPI Wireless
00 20 D8 2E 97 40	Atwater Kent	First	71°48'25.57"W	42°16'31.91"N	WPI Wireless
00 20 D8 2E E6 C0	Atwater Kent	First	71°48'25.91"W	42°16'31.47"N	WPI Wireless
00 20 D8 2E 8E 00	Atwater Kent	Second	71°48'25.79"W	42°16'31.64"N	WPI Wireless
00 20 D8 2E A9 00	Atwater Kent	Second	71°48'24.10"W	42°16'30.99"N	WPI Wireless
00 20 D8 2E F5 40	Atwater Kent	Third	71°48'25.01"W	42°16'31.01"N	WPI Wireless
00 20 D8 2E 88 80	Atwater Kent	Third	71°48'26.07"W	42°16'30.93"N	WPI Wireless
00 16 CA 32 BD C0	Fuller Labs	Ground	71°48'22.41"W	42°16'30.56"N	WPI Wireless
00 16 CA 32 91 40	Fuller Labs	First	71°48'22.31"W	42°16'30.54"N	WPI Wireless
00 16 CA 32 BD 80	Fuller Labs	Second	71°48'22.31"W	42°16'30.48"N	WPI Wireless
00 16 CA 32 8F 80	Fuller Labs	Second	71°48'22.89"W	42°16'29.74"N	WPI Wireless
00 16 CA 32 8E 80	Fuller Labs	Second	71°48'23.13"W	42°16'30.23"N	WPI Wireless
00 16 CA 32 82 C0	Fuller Labs	Third	71°48'22.97"W	42°16'29.86"N	WPI Wireless
00 20 D8 2E 45 C0	Library	First	71°48'23.68"W	42°16'27.68"N	WPI Wireless
00 20 D8 2E EB 40	Library	First	71°48'22.79"W	42°16'26.58"N	WPI Wireless
00 20 D8 2E D3 00	Library	Second	71°48'22.89"W	42°16'27.78"N	WPI Wireless
00 20 D8 2F 19 80	Library	Second	71°48'22.83"W	42°16'26.75"N	WPI Wireless
00 20 D8 2E 51 C0	Library	Third	71°48'23.77"W	42°16'27.03"N	WPI Wireless
00 20 D8 2E B8 40	Library	Third	71°48'22.67"W	42°16'27.61"N	WPI Wireless
00 20 D8 25 52 80	Higgins Labs	First	71°48'29.64"W	42°16'27.42"N	WPI Wireless
00 16 CA 3A 8A 80	Higgins Labs	First	71°48'29.94"W	42°16'26.43"N	WPI Wireless
00 20 D8 25 1E 00	Higgins Labs	Second	71°48'30.04"W	42°16'27.43"N	WPI Wireless
00 16 CA 38 3D 80	Higgins Labs	Second	71°48'30.01"W	42°16'26.53"N	WPI Wireless
00 20 D8 24 37 40	Higgins Labs	Third	71°48'29.95"W	42°16'26.86"N	WPI Wireless
00 20 D8 23 DD 80	Campus Center	First	71°48'30.28"W	42°16'30.00"N	WPI Wireless

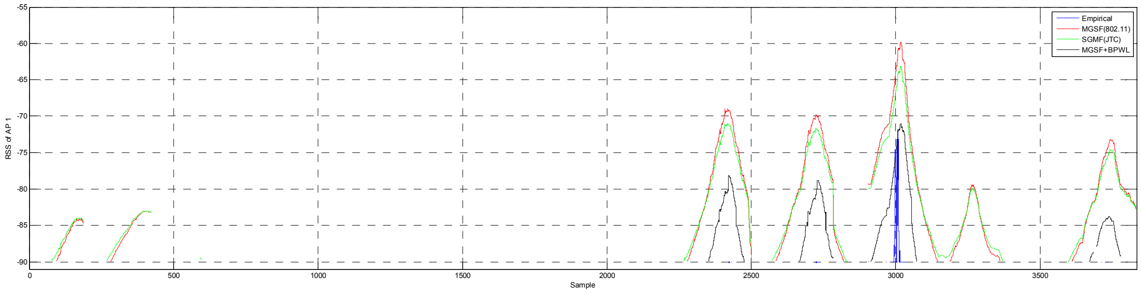
AP MAC Address	Building	Floor	Longitude	Latitude	SSID
00 20 D8 23 C4 80	Campus Center	Second	71°48'30.36"W	42°16'28.92"N	WPI Wireless
00 20 D8 2E CA 00	Campus Center	Second	71°48'30.13"W	42°16'29.82"N	WPI Wireless
00 20 D8 23 C5 C0	Campus Center	Third	71°48'30.44"W	42°16'29.02"N	WPI Wireless
00 20 D8 23 C6 80	Campus Center	Third	71°48'30.03"W	42°16'29.77"N	WPI Wireless
00 16 CA 32 95 40	Olin Hall	First	71°48'28.87"W	42°16'29.57"N	WPI Wireless
00 20 D8 25 BD 80	Kaven Hall	Ground	71°48'21.46"W	42°16'29.98"N	WPI Wireless
00 20 D8 25 C1 00	Kaven Hall	Ground	71°48'20.82"W	42°16'29.00"N	WPI Wireless
00 20 D8 25 B2 40	Kaven Hall	First	71°48'20.94"W	42°16'29.32"N	WPI Wireless
00 20 D8 25 F3 80	Kaven Hall	Second	71°48'20.80"W	42°16'29.53"N	WPI Wireless
00 15 E8 E6 A7 00	Stoddard Labs	First	71°48'25.60"W	42°16'26.46"N	WPI Wireless
00 15 E8 E6 9D 80	Stoddard Labs	Third	71°48'25.47"W	42°16'25.84"N	WPI Wireless
00 16 CA 32 BD 00	Stratton Hall	Ground	71°48'27.76"W	42°16'25.81"N	WPI Wireless
00 16 CA 32 A3 80	Stratton Hall	First	71°48'27.70"W	42°16'26.36"N	WPI Wireless
00 16 CA 32 9E 00	Stratton Hall	First	71°48'28.00"W	42°16'25.54"N	WPI Wireless
00 16 CA 32 C3 C0	Stratton Hall	Second	71°48'27.70"W	42°16'25.90"N	WPI Wireless
00 16 CA 32 A2 00	Stratton Hall	Third	71°48'27.57"W	42°16'26.43"N	WPI Wireless
00 16 CA 32 B9 80	Boynton Hall	First	71°48'25.78"W	42°16'24.33"N	WPI Wireless
00 16 CA 32 BA 80	Boynton Hall	Second	71°48'27.17"W	42°16'24.37"N	WPI Wireless
00 20 D8 2E A0 40	Bartlett Center	First	71°48'31.94"W	42°16'25.74"N	WPI Wireless
00 20 D8 2E B0 00	Bartlett Center	First	71°48'32.71"W	42°16'25.07"N	WPI Wireless
00 20 D8 2E 97 80	Bartlett Center	Second	71°48'32.49"W	42°16'25.80"N	WPI Wireless
00 20 D8 2E A5 C0	Bartlett Center	Second	71°48'32.16"W	42°16'24.98"N	WPI Wireless

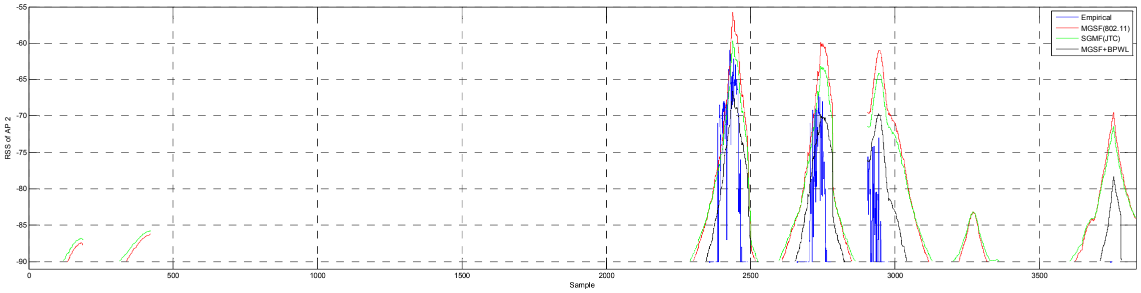
Appendix B RSS Signature Segment Length Additional Distribution Fitting

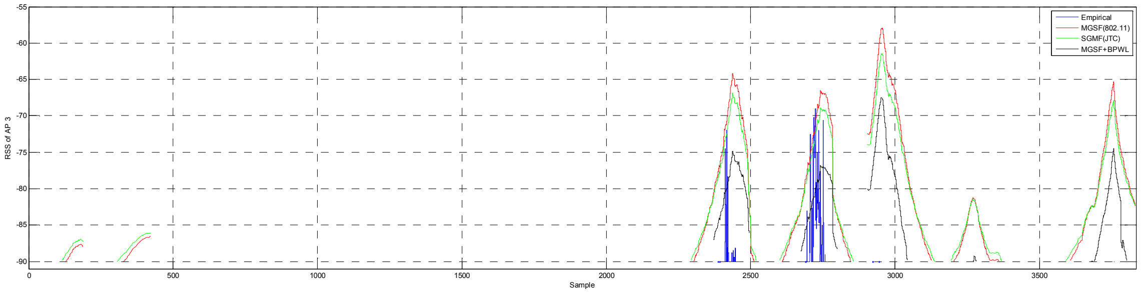


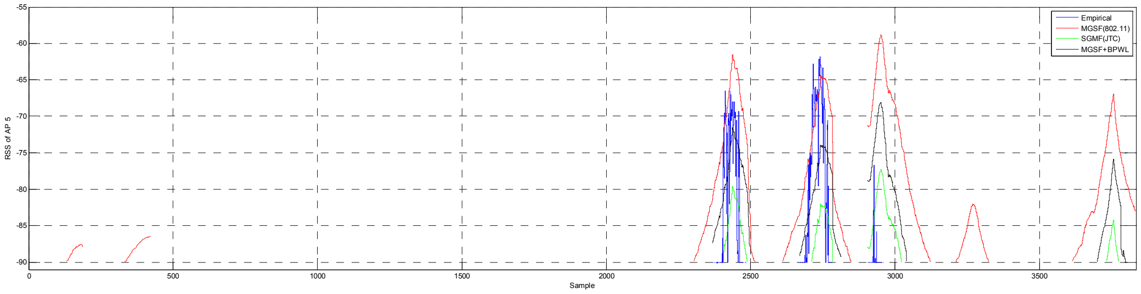

Empirical
Weibull
Rician
Logistic
Gamma
Nakagami
Extreme Value
Birnbaum-Saunders
Probability
0.9999
0.999
0.99
0.95
0.9
0.75
0.5
0.25
0.1
0.05
0.01
0.001
0.0001
10^0
10^1
10^2
Data

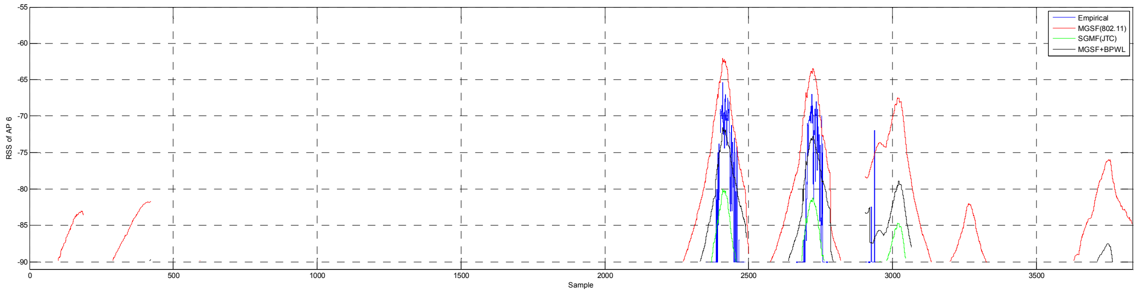
Appendix C Plots of the Empirical and Model RSS Signatures

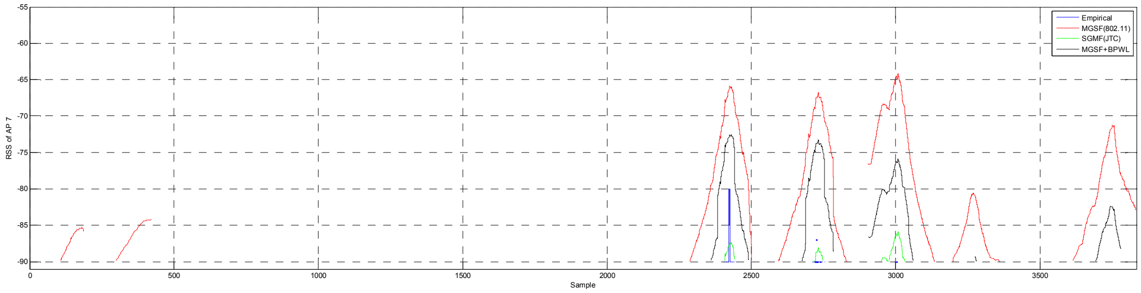


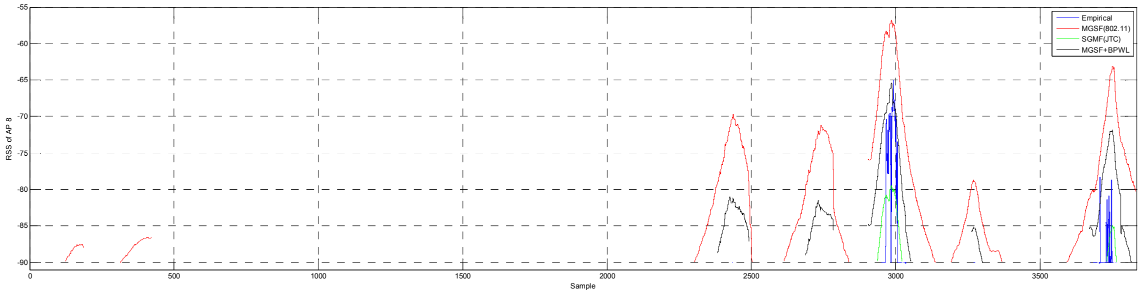


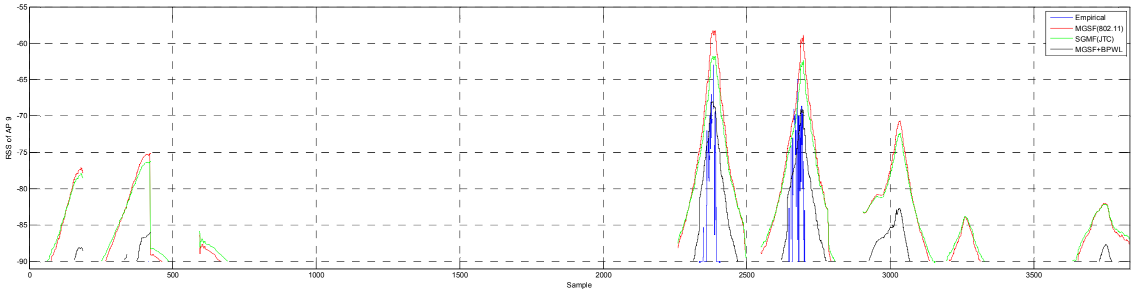


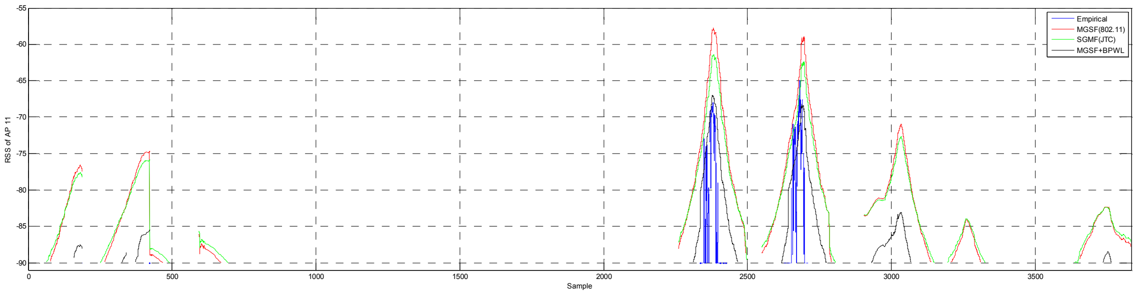



Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL
RSS of AP 6
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500

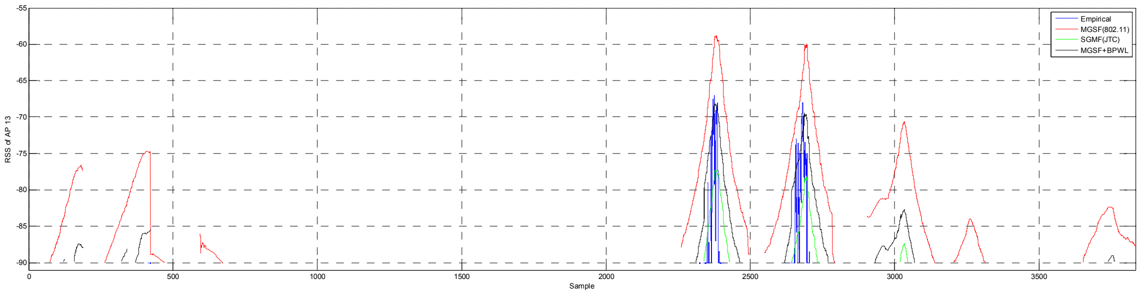

Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL
RSS of AP 7
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500


Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL
RSS of AP 8
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500

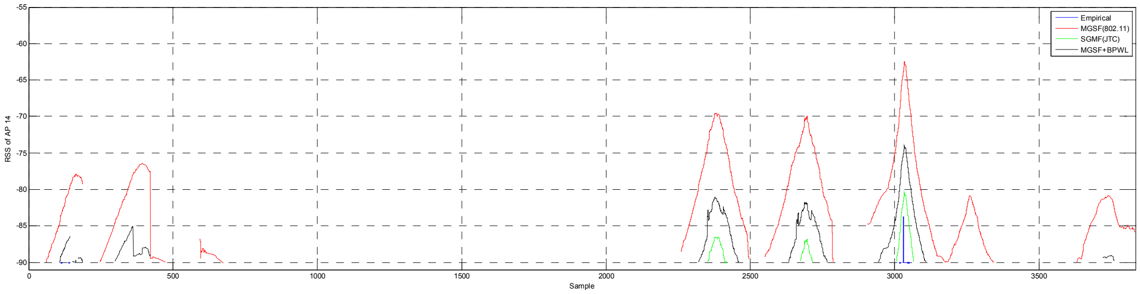

Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL
RSS of AP 9
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500



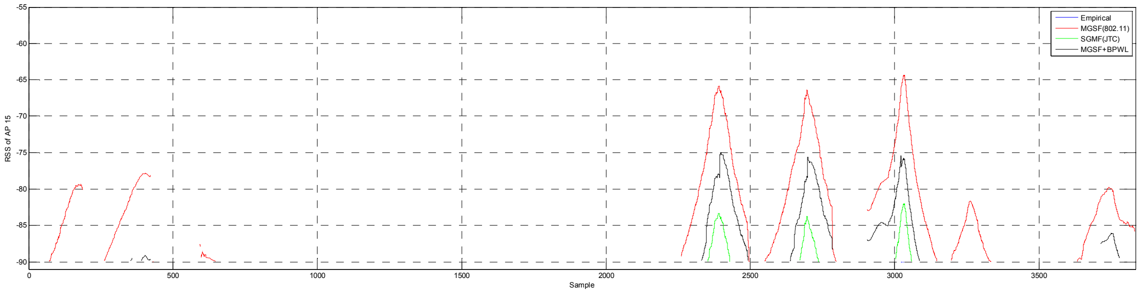
RSS of AP 11
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



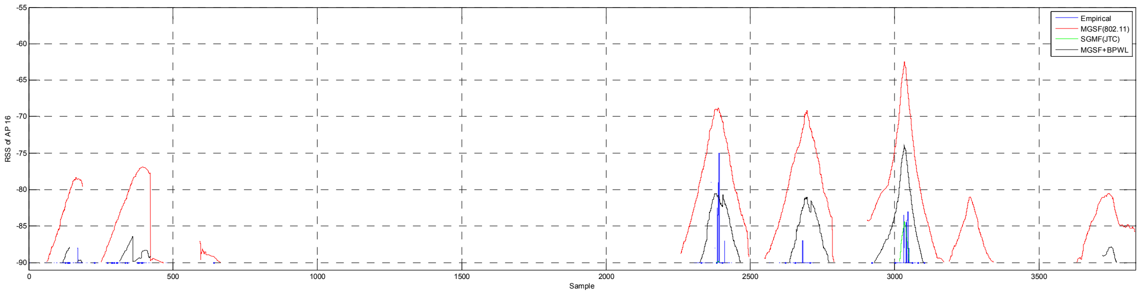
RSS of AP 13
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



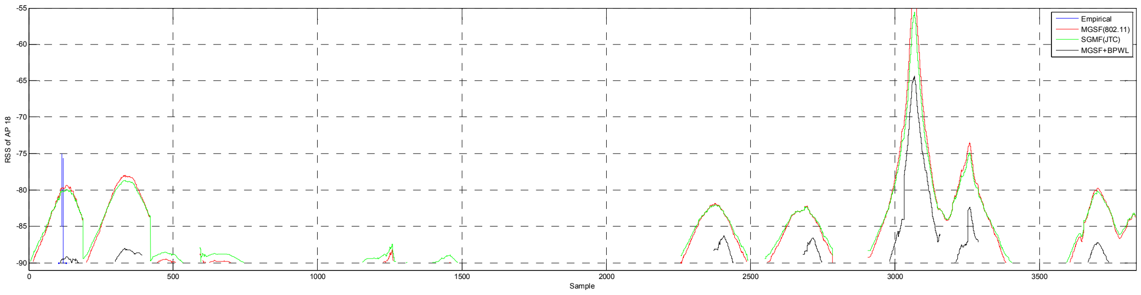
RSS of AP 14
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



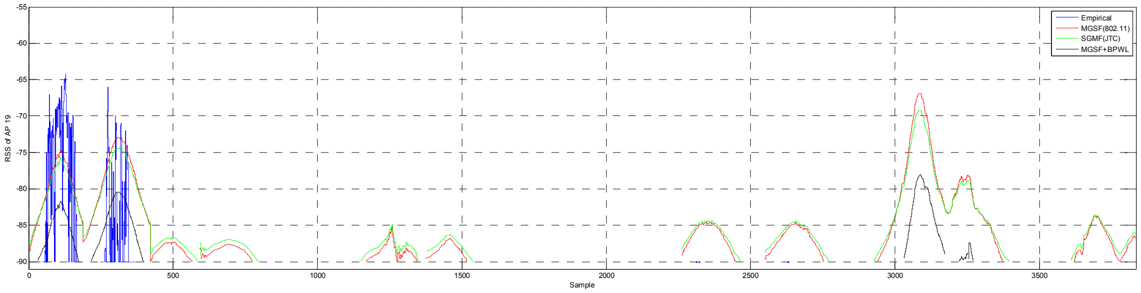
RSS of AP 15
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



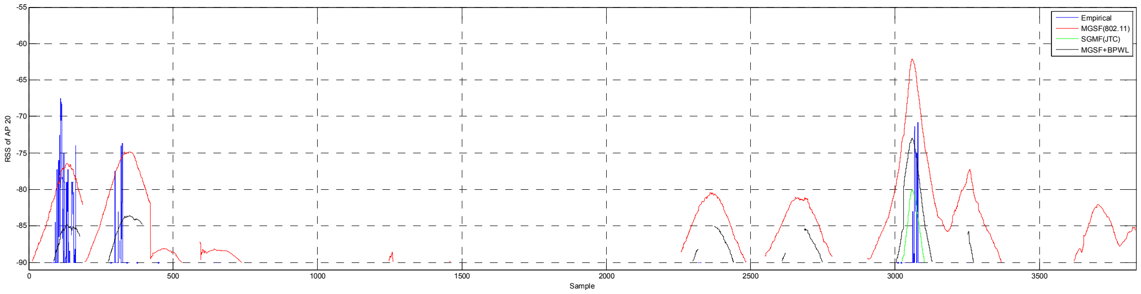
RSS of AP 16
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



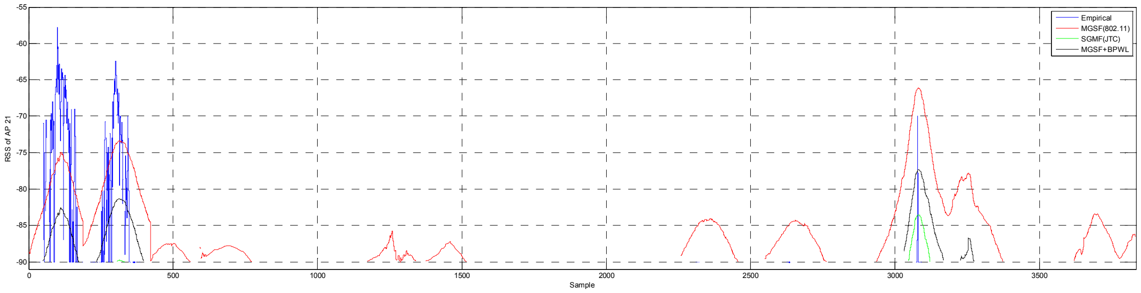
RSS of AP 18
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



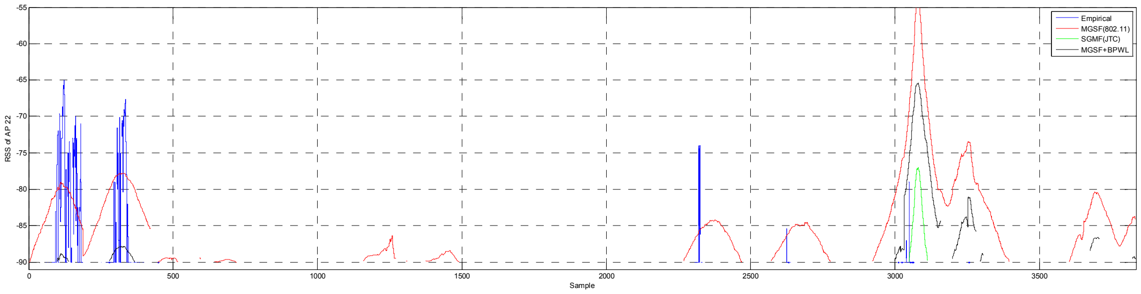
RSS of AP 19
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



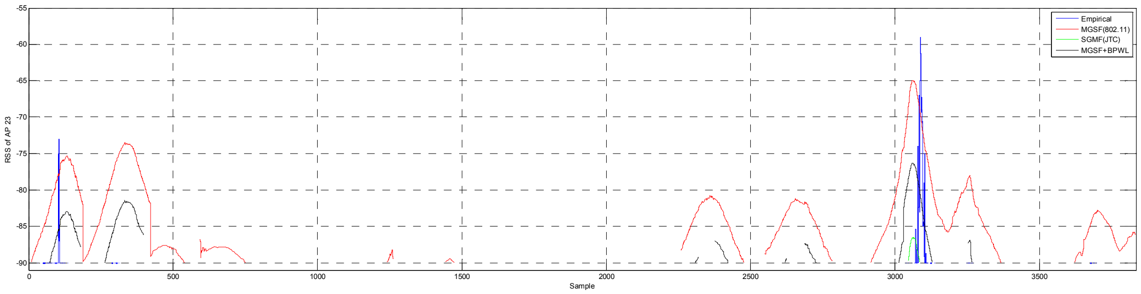
RSS of AP 20
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



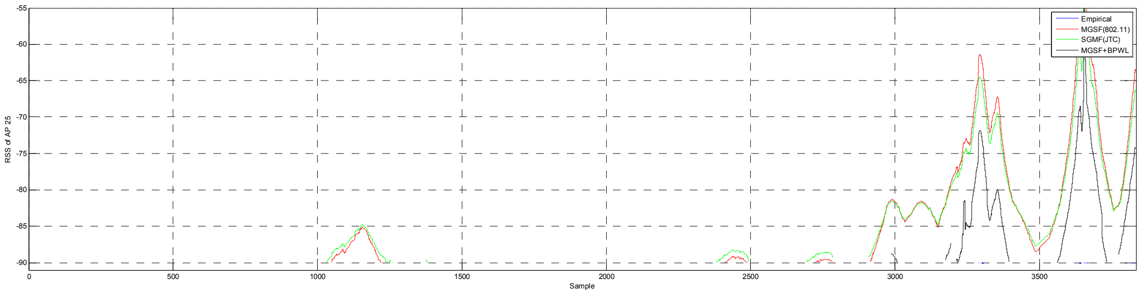
RSS of AP 21
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



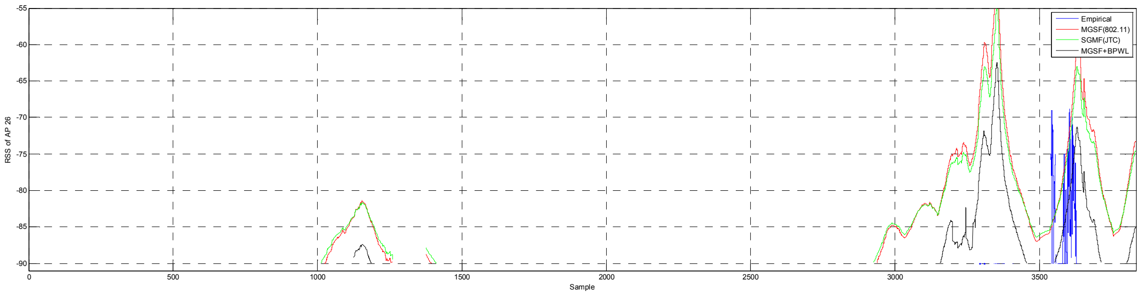
RSS of AP 22
Sample
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



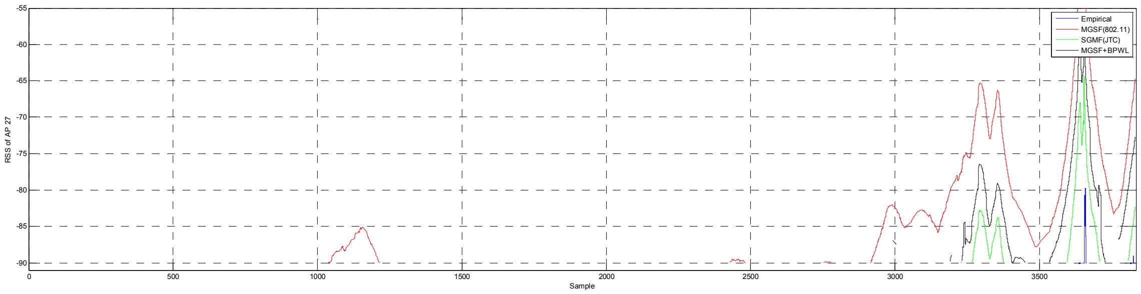
RSS of AP 23
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



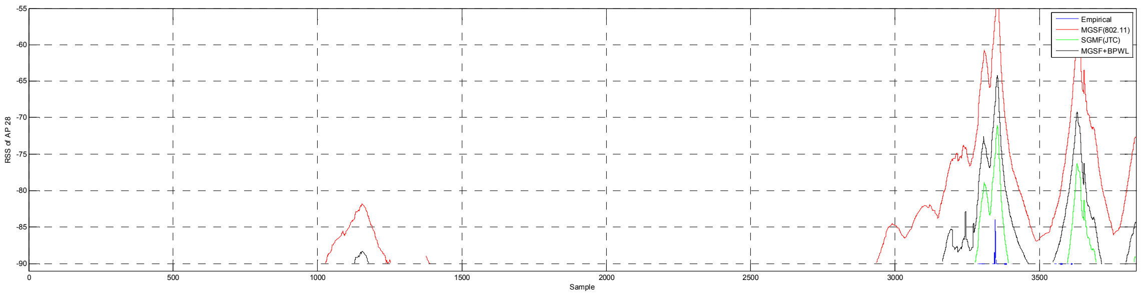
RSS of AP 25
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



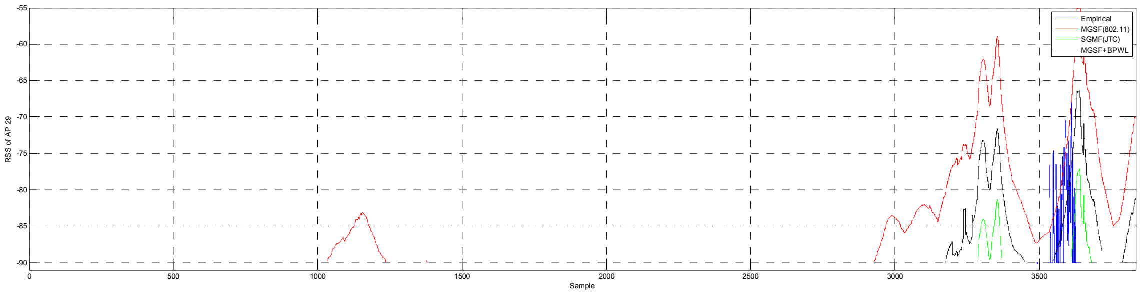
RSS of AP 26
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

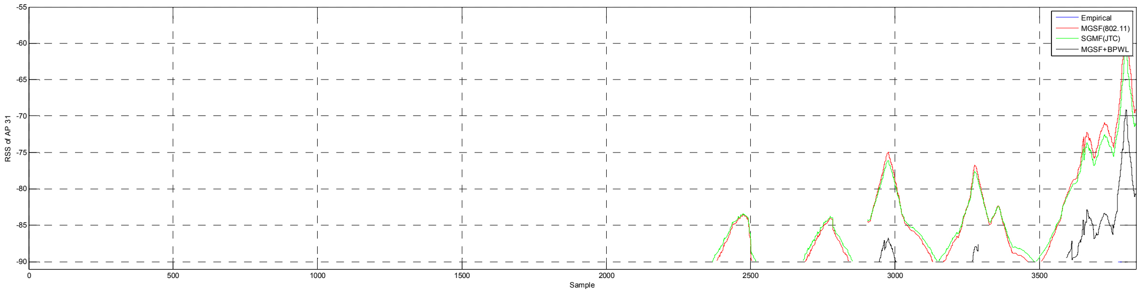


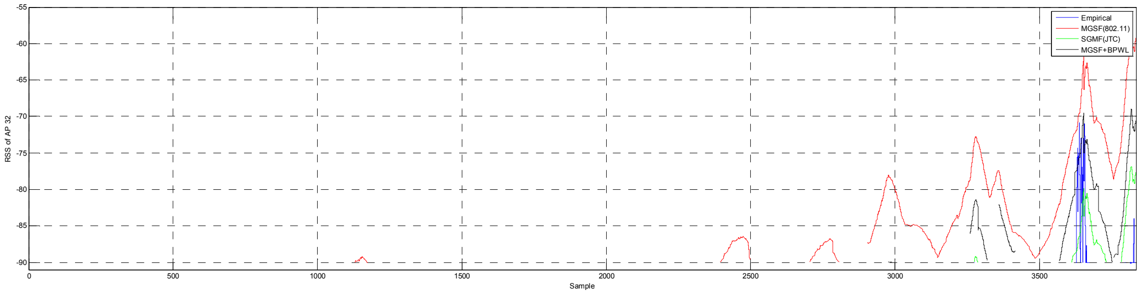
RSS of AP 27
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

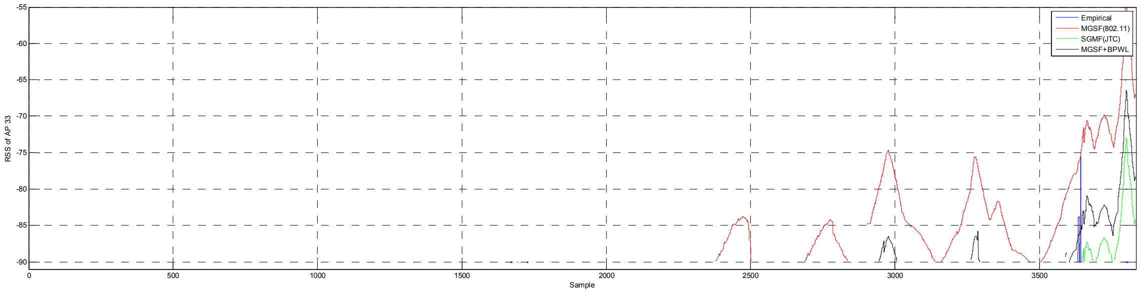


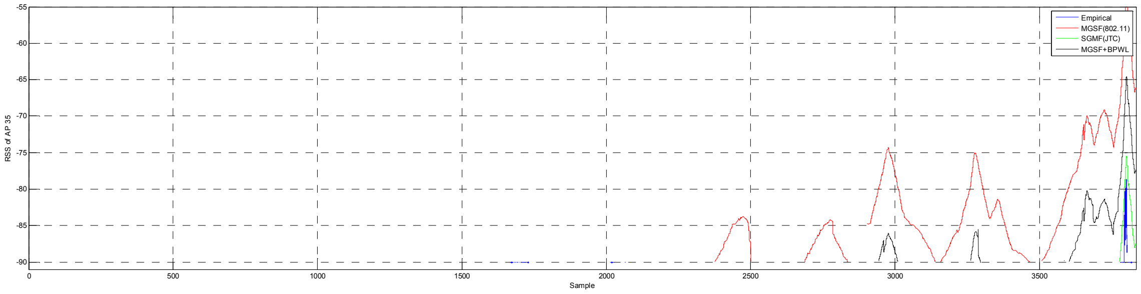
RSS of AP 28
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

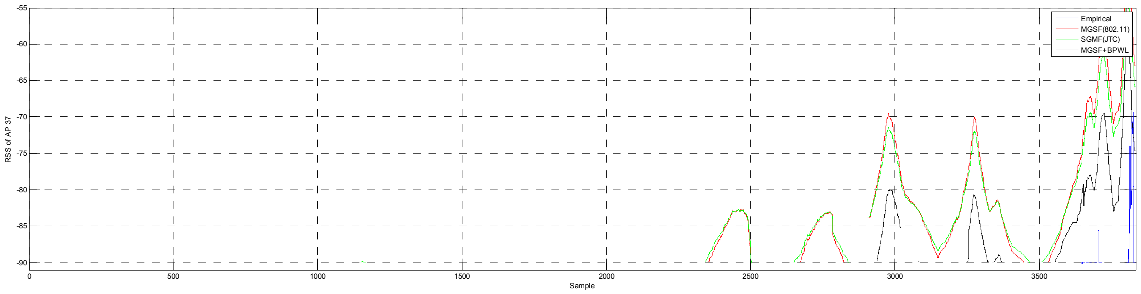

-55
-60
-65
-70
-75
-80
-85
-90
RSS of AP 29
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL


-55
-60
-65
-70
-75
-80
-85
-90
RSS of AP 31
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

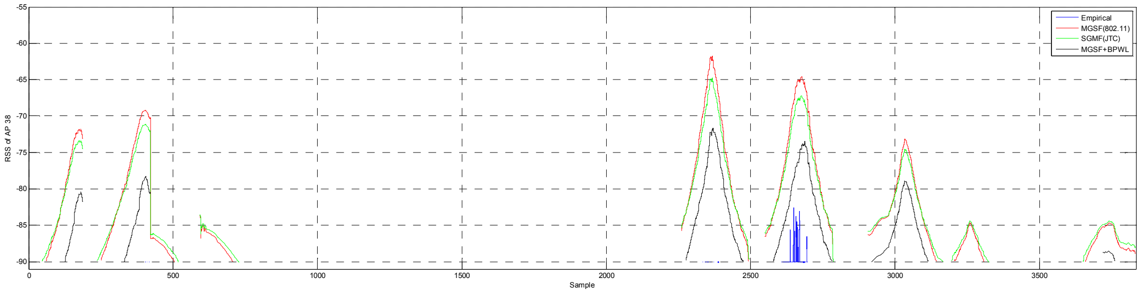

-55
-60
-65
-70
-75
-80
-85
-90
RSS of AP 32
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL


-55
-60
-65
-70
-75
-80
-85
-90
RSS of AP 33
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

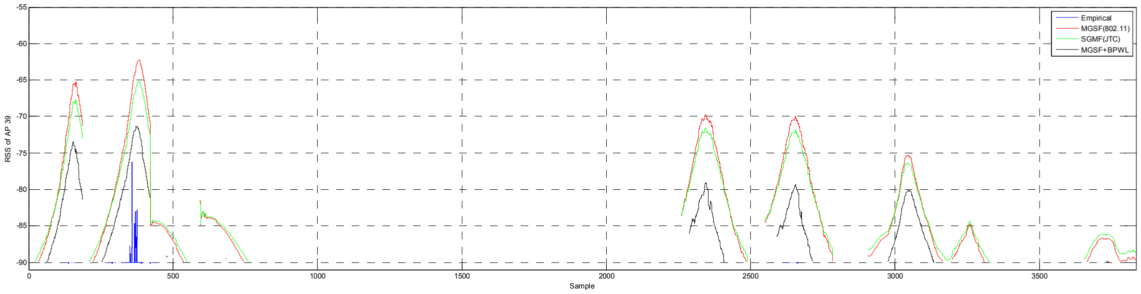

-55
-60
-65
-70
-75
-80
-85
-90
RSS of AP 35
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



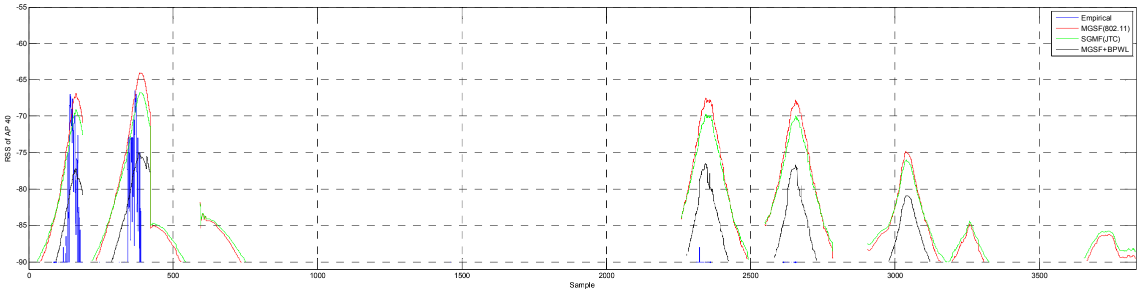
RSS of AP 37
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



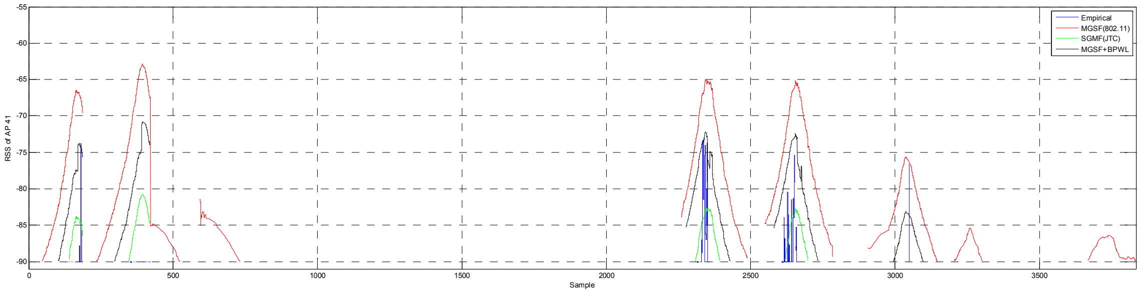
RSS of AP 38
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL



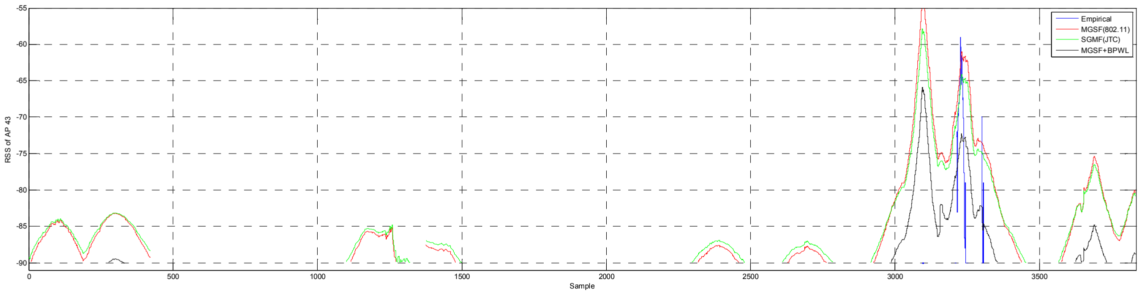
RSS of AP 39
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

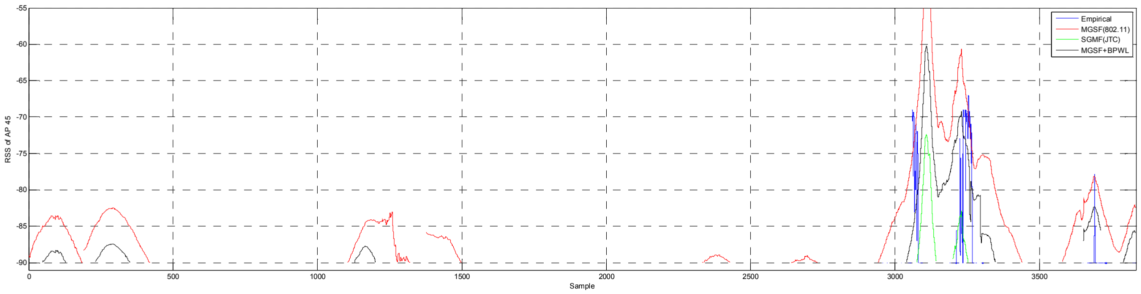


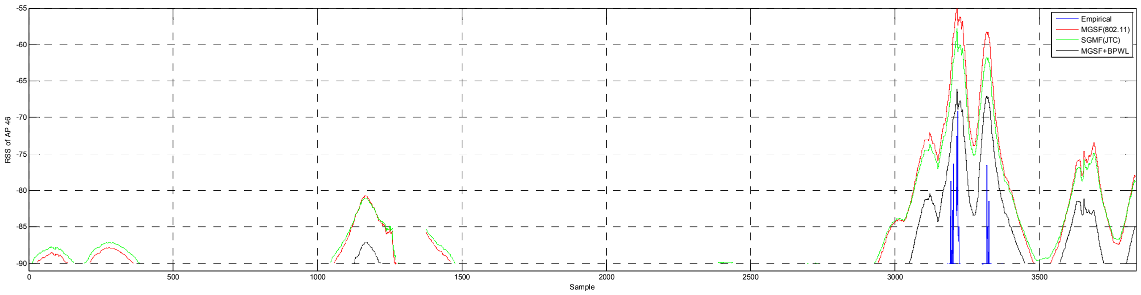
RSS of AP 40
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

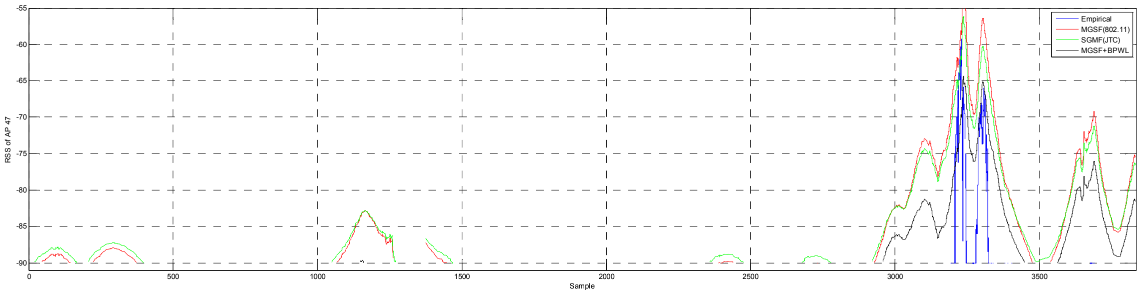


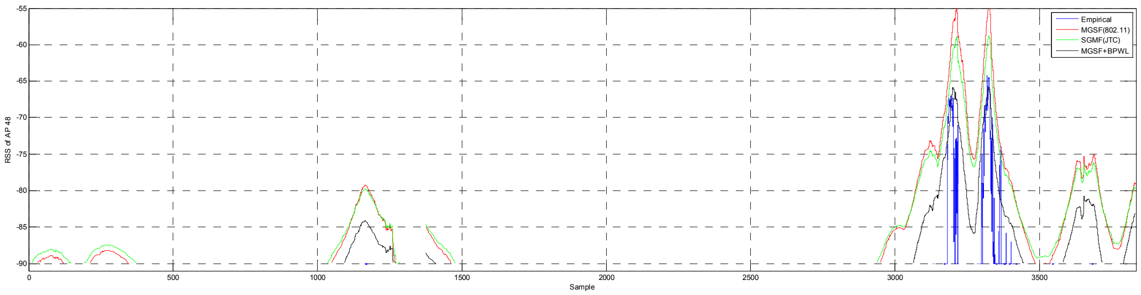
RSS of AP 41
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

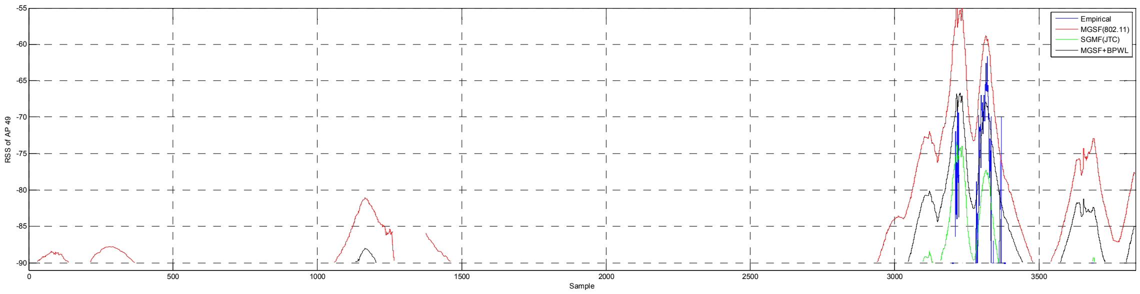

RSS of AP 43
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

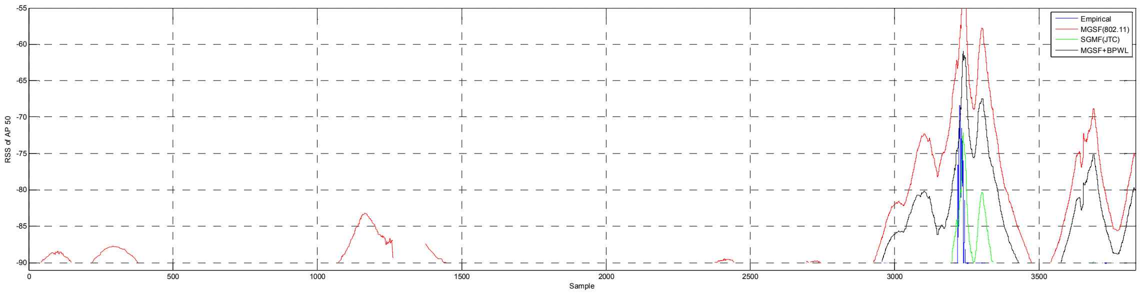

RSS of AP 45
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

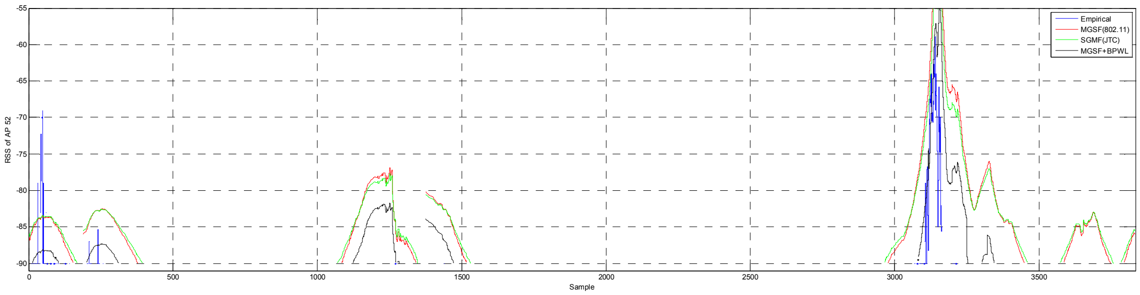

RSS of AP 46
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

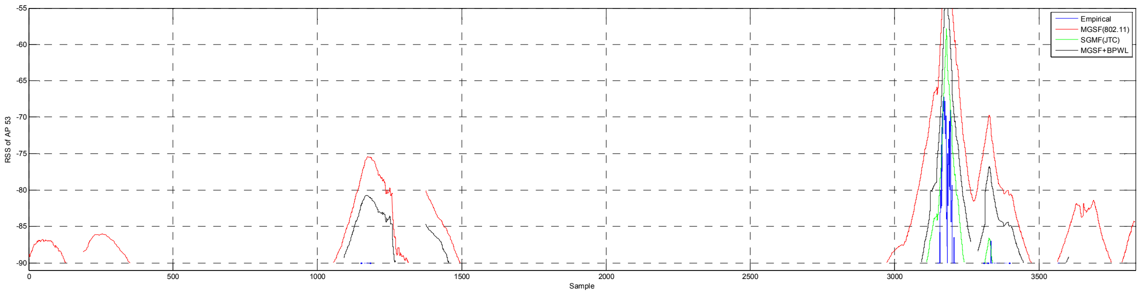

RSS of AP 47
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

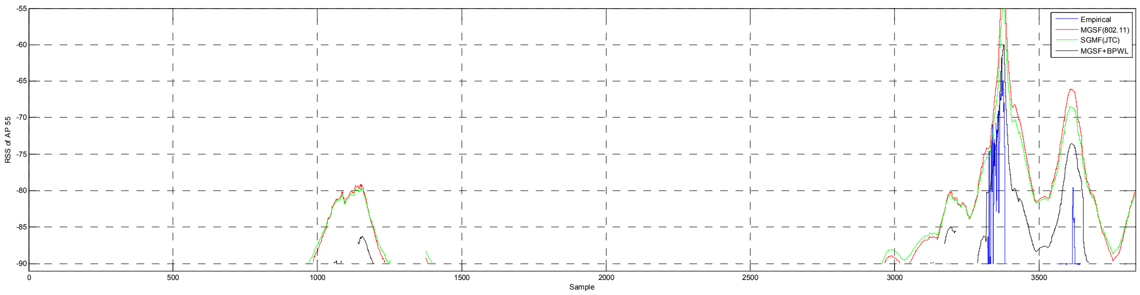

RSS of AP 48
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

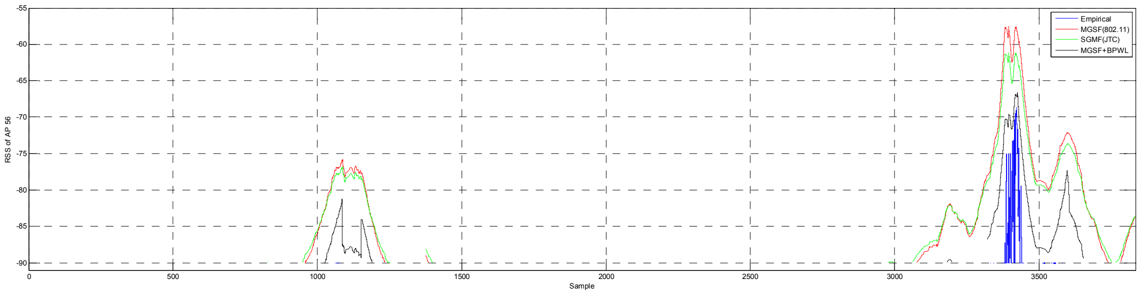

RSS of AP 49
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

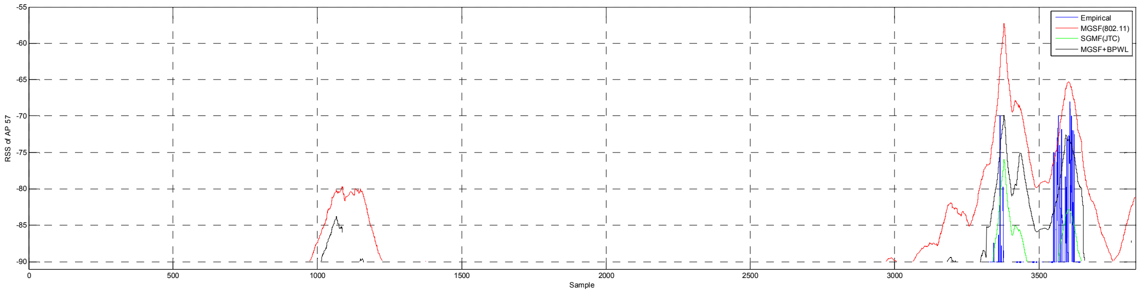

RSS of AP 50
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL

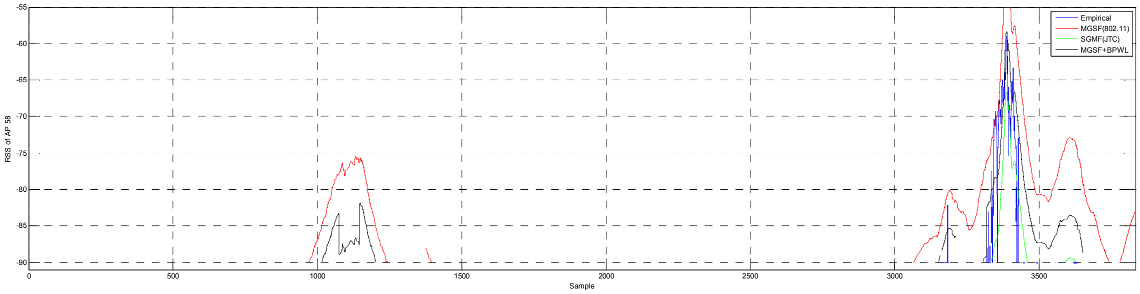

RSS of AP 52
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL


RSS of AP 53
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL


RSS of AP 55
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample
Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL


Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL
RSS of AP 56
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample


Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL
RSS of AP 57
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample


Empirical
MGSF(802.11)
SGMF(JTC)
MGSF+BPWL
RSS of AP 58
-55
-60
-65
-70
-75
-80
-85
-90
0
500
1000
1500
2000
2500
3000
3500
Sample

Appendix D MATLAB Code

D.1. MATLAB Function to read Google Earth kml files

The code following code was downloaded from MATLAB® Central File Exchange at http://www.mathworks.com/matlabcentral/fileexchange/13026 and was authored by Amy Farris

```
function [lat,lon,z] = read_kml(fileName)
%READ_KML reads a Google Earth kml file into Matlab
% Reads in lat,lon,z from a simple path file.
%
%  All the data in the data file must EITHER be on one line, for
example:
%   -73.6513,40.4551,0 -73.3905,40.5214,0 -73.0589,40.5956,0
% OR each point must be on its own line, for example:
%   -73.237171, 40.627311, 0.0
%   -73.242945, 40.626360, 0.0
%
%  I have tried to make this code as robust as possible, but it may
still
%  crash if there is a space in the wrong place in the data file.
%
% Example:
%   [lat,lon,z] = read_kml('test.kml');
%
% where test.kml looks like:
% <?xml version="1.0" encoding="UTF-8"?>
% <kml xmlns="http://earth.google.com/kml/2.1">
% <Placemark>
%   <name>test_length</name>
%   <description>junk</description>
%   <LineString>
%     <tessellate>1</tessellate>
%     <coordinates>
% -73.65138440596144,40.45517368645169,0 -
73.39056199144957,40.52146569128411,0 -
73.05890757388369,40.59561213913959,0 -
72.80519929505505,40.66961872411046,0 -
72.61180114704385,40.72997510603909,0 -
72.43718187249095,40.77509309196679,0 </coordinates>
%   </LineString>
% </Placemark>
% </kml>
% afarris@usgs.gov 2006 November

%% open the data file and find the beginning of the data
fid=fopen(fileName);
if fid < 0
    error('could not find file')
end
done=0;
while done == 0
```

```
    junk = fgetl(fid);
    f=findstr(junk,'<coordinates>');
    if isempty(f) == 0
        done = 1;
    end
end
ar = 1;
% junk either ends with the word '<coordinates>' OR
% some data follows the word '<coordinates>'
if (f + 13) >= length(junk)
    % no data on this line
    % done2 is set to zero so the next loop will read the data
    done2 = 0;
else
    % there is some data in this line following '<coordinates>'
    clear f2
    f2=findstr(junk,'</coordinates>');
    if isempty(f2) == 0
        %all data is on this line
        alldata{ar} = junk(f+13:f2-1);
        % done2 is set to one because the next loop does not need to
run
        done2 = 1;
    else
        % only some data is on this line
        alldata{ar} = junk(f+13:end);
        ar = ar + 1;
        % done2 is set to zero so the next loop will read the rest of
the data
        done2 = 0;
    end
end

%% Read in the data
while done2 == 0
    % read in line from data file
    junk = fgetl(fid);
    f=findstr(junk,'</coordinates>');
    if isempty(f) == 1
        % no ending signal, just add this data to the rest
        alldata{ar} = junk;
        ar = ar + 1;
    else
        % ending signal is present
        if f < 20
            % </coordinates> is in the begining of the line, ergo no
data
            % on this line; just end the loop
            done2 = 1;
        else
            % the ending signal (</coordinates>) is present: remove it,
            % add data to the rest and signal the end of the loop
            f2 = strfind(junk,'</coordinates>');
            alldata{ar} = junk(1:f2-1);
            ar = ar + 1;
            done2 = 1;
        end
    end
end

%% get the data into neat vectors
% either all the data is on one line, or each point is on its own line
f = strfind(alldata{1}(:)',',');
if length(f) > 2
    % more than one coordinate on each line, this is hard b/c there is
no
    % comma between points (just commans between lon and lat, and
between
    % lat and z)  ie;  -70.0000,42.0000,0 -70.1000,40.10000,0 -
70.2,....
    %
    %  I have to divide the string into Latitude, Longitude and Z
values
    % using the locations of both commas and spaces.
    %
    % turn alldata into regular vector so it is easier to work with
    data = cell2mat(alldata);
    % now find all commas
    fComma = strfind(data, ',');
    % find all spaces
    fSpace = strfind(data,' ');
    a=1;
    fC = 1;
    % have to do first point seperately b/c line may not begin with a
space
    lon(a) = str2num(data(1:fComma(fC)-1));
    lat(a) = str2num(data(fComma(fC)+1:fComma(fC+1)-1));
    z(a) = str2num(data(fComma(fC+1)+1:fSpace(1)-1));
    a=a+1;
    fS=1;
    % go thru all the points in the line
    for fC = 3: 2: length(fComma)
        lon(a) = str2num(data(fSpace(fS)+1:fComma(fC)-1));
        lat(a) = str2num(data(fComma(fC)+1:fComma(fC+1)-1));
        if fS  < length(fSpace)
            z(a) = str2num(data(fComma(fC+1)+1:fSpace(fS+1)-1 ));
        else
            % have to handle last point seperatly b/c line may not end
with
            % a space
            z(a) = str2num(data(fComma(fC+1)+1:end ));
        end
        a=a+1;
        fS=fS+1;
    end
 else
    %each point is on its own line
    for i = 1 : size(alldata,2)
        fComma = strfind(alldata{i}(:)',',');
        lon(i) = str2num(alldata{i}(1:fComma(1)-1));
        lat(i) = str2num(alldata{i}(fComma(1)+1:fComma(2)-1));
        z(i) = str2num(alldata{i}(fComma(2)*1:end));
    end
 end
```

```
fclose(fid);
[a,b]=size(lat);
lat=reshape(lat,max(a,b),min(a,b));
lon=reshape(lon,max(a,b),min(a,b));
z=reshape(z,max(a,b),min(a,b));
```

D.2. MATLAB ReadAPdatabase

```
function [AP, AP_MAC, AP_lat, AP_lon, AP_fl] = readAPdb(filename,
pathName)


fid = fopen([pathName filename]);
APdb = textscan(fid, '%s%s%s%s%s%s%s%s', 'delimiter', ',');
fclose(fid);


APfilename = APdb{2};
AP = APfilename(2:end);
APfilename_char = char(APfilename(2:end));
AP_MAC(:, [1 2 4 5 7 8 10 11 13 14 16 17]) = ...
  APfilename_char(:, [1 2 4 5 7 8 10 11 13 14 16 17]);
AP_MAC(:, [3 6 9 12 15]) = ' ';

AP_fl_cell = APdb{4};
AP_fl = -ones(length(AP_fl_cell)-1,1);
AP_fl(strcmp('Ground', AP_fl_cell(2:end))) = 0;
AP_fl(strcmp('First', AP_fl_cell(2:end))) = 1;
AP_fl(strcmp('Second', AP_fl_cell(2:end))) = 2;
AP_fl(strcmp('Third', AP_fl_cell(2:end))) = 3;

if sum(AP_fl == -1) ~= 0
  error('AP_fl')
end

AP_lon_cell = APdb{5};
AP_lon_char = char(AP_lon_cell);
AP_lon = str2angle(AP_lon_char(2:end, 2:end-1));

AP_lat_cell = APdb{6};
AP_lat_char = char(AP_lat_cell);
AP_lat = str2angle(AP_lat_char(2:end, 2:end-1));
```

D.3. MATLAB Empirical Data Pre-Process

```
clc
clear
```

```
close all
format short

%% Params
src = 'SkyHook';
if strcmp(src, 'SkyHook')
  header_len = 3
else
  header_len = 5
end
pathName = '..\Scanning_data\file_2_run1\'

APrange = [1:3 5:8];
%% constants
% FileName = 'APdatabaseOUT.csv';
PathName = '..\WirelessMon Data\';
outputPath = '..\GE_kmls\';
s_path = '..\MAT_data\';
%%
md_i = 1;

%% get data
tic

[AP, AP_MAC, AP_lat, AP_lon] = readAPdb('WPI - Access Points List.csv',
'..\');

files = dir(pathName);

RSSI_on_path = cell(max(APrange), length(files)-sum([files.isdir]));
f_i = 1;
for fnum = 1:length(files)
  if ~files(fnum).isdir
    files(fnum).name
    files_out{f_i} = files(fnum);
    %load data
    [sampleInd t_stamp cell_MACaddr cell_RSSI la lo flr_num inout] =
...
      getAPdata(src, files(fnum).name, pathName);
    num_samples = length(sampleInd)-header_len;

    %format data
    RSSI = str2double(cell_RSSI(header_len:end));
    lat = str2double(la(header_len:end));
    lon = str2double(lo(header_len:end));
    t_stamp = t_stamp(header_len:end);

    %find uinque pts
    uu_lat = lat(1);
    uu_lon = lon(1);
    num_upts = 1;
    for ii = 2:length(lat)
      if uu_lat(num_upts) ~= lat(ii) || uu_lon(num_upts) ~= lon(ii)
        num_upts = num_upts+1;
        uu_lat(num_upts) = lat(ii);
        uu_lon(num_upts) = lon(ii);
      end
    end

    cell_MACaddr=cell_MACaddr(header_len:end);

    for APsel = 1:length(AP_MAC)%APrange
      %create mask for current AP
      AP_mask = strncmpi(AP(APsel, :), cell_MACaddr, 17);
      fprintf('AP %d sum_AP_Mask %d\n', APsel, sum(AP_mask))

      %Apply mask
      AP_masked_RSSI = RSSI(AP_mask);

      %mask the AP lat and lon
      AP_masked_lat = lat(AP_mask);
      AP_masked_lon = lon(AP_mask);

      RSSI_on_path_t = nan(1, num_upts);

      %if AP was detected in file
      if ~isempty(AP_masked_lat)

        %find AP unique points
        APuLat = AP_masked_lat(1);
        APuLon = AP_masked_lon(1);
        uAPpts = 1;
        num_uAPpts = 1;
        for AP_masked_i = 2:length(AP_masked_lat)
          if APuLat(num_uAPpts) ~= AP_masked_lat(AP_masked_i) || ...
              APuLon(num_uAPpts) ~= AP_masked_lon(AP_masked_i)
            num_uAPpts = num_uAPpts+1;
            APuLat(num_uAPpts) = AP_masked_lat(AP_masked_i);
            APuLon(num_uAPpts) = AP_masked_lon(AP_masked_i);
            uAPpts(num_uAPpts) = AP_masked_i;
          end
        end

        %avg RSS at same location
        AP_masked_mRSSI = ones(num_uAPpts,1);
        for APupts_i = 1:num_uAPpts
          AP_masked_mRSSI(APupts_i) = mean(AP_masked_RSSI(AP_masked_lat
== APuLat(APupts_i) & AP_masked_lon == APuLon(APupts_i)));
        end


        %create mask from unique Ap points and unique Lat Lon locations
        pts_mask = false(num_upts,1);
        for APupts_i = 1:num_uAPpts
          for upts_i = 1:num_upts
            if APuLat(APupts_i) == uu_lat(upts_i) && APuLon(APupts_i)
== uu_lon(upts_i)
              pts_mask(upts_i) = true;
            else
              pts_mask(upts_i) = pts_mask(upts_i) | false;
      end
     end
    end

    if sum(pts_mask) == num_uAPpts
     RSSI_on_path_t(pts_mask) = AP_masked_mRSSI;
     display('found points')
    else
     error('');
    end
   end
   %store results
   RSSI_on_path{APsel, f_i} = RSSI_on_path_t;

%   if mod(APsel-1, 4) == 0
%    figure
%    ff= 1;
%   else
%    ff= ff+1;
%   end
%   subplot(2,2,ff);
% %   stairs(AP_masked_RSSI);
%   stairs(RSSI_on_path_t);
%   title([files(fnum).name ' AP ' num2str(APsel)]);
% %   axis([0 length(AP_masked_RSSI) -100 -50])
%   axis([0 length(RSSI_on_path_t) -100 -50])

  end%APs

  unique_lat{f_i} = uu_lat;
  unique_lon{f_i} = uu_lon;
  f_i = f_i+1;
 end%~dir
end%files
time_s = sprintf('%d-%02d-%02d_%02d%02d%02d', fix(clock));
filename = [s_path  'Emp3_' time_s];
save(filename, 'RSSI_on_path', 'unique_lat', 'unique_lon', 'files_out')
toc
```

D.4. MATLAB Wall Breakpoint Database Builder

D.4.1. MATLAB runFindBreakPt

```
clc
clear
close all
format short
```

```
%% constants
s_path = '..\MAT_data\';

%% get data

%Get AP data
[AP, AP_MAC, AP_lat, AP_lon] = readAPdb('WPI - Access Points List.csv',
'..\');

%Get Empirical data
files = dir(s_path);
datenum = 0;
for file_i = 1:length(files)
  if strncmp('Emp', files(file_i).name, 3)
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  error('No processed empirical data run "runEmpData"');
end

display(['Loading ' fname]);
E_data = load([s_path fname])

%%
wf_dir = '..\GE_kmls\walls\';

wall_files = dir(wf_dir);

wLat_cell = cell(1);
wLon_cell = cell(1);
%
% if ~exist('opt', 'var')
%   opt = 0;
% end

build_i = 1;
for wf_i = 1:length(wall_files)
  % if not a dir
  if ~wall_files(wf_i).isdir
    %read AK Wall coordinates from file
    [wlat, wlon, z] = read_kml([wf_dir wall_files(wf_i).name]);
    wLat_cell{build_i} = wlat;
    wLon_cell{build_i} = wlon;
    build_i = build_i+1;
  end%if not dir
end%for files

%% find breakpoints
tic
RSS_size = size(E_data.RSSI_on_path);
short_dbp = cell(RSS_size);
```

```
walls = cell(RSS_size);
mask = cell(RSS_size);
% long_dbp = cell(length(AP), length(E_data.unique_lat));
for file_i = 1:RSS_size(2)
  fprintf('file_i %d of %d\n', file_i, RSS_size(2))
  for AP_i = 1:RSS_size(1)
    fprintf('AP %d of %d\n', AP_i, RSS_size(1))
    [short_dbp{AP_i, file_i} walls{AP_i, file_i} mask{AP_i, file_i}]=
...
      findBreakPt(E_data.unique_lat{file_i}, E_data.unique_lon{file_i},
AP_lat(AP_i), AP_lon(AP_i), wLat_cell, wLon_cell);
%     long_dbp{AP_i, file_i} = findBreakPt(E_data.unique_lat{file_i},
E_data.unique_lon{file_i}, AP_lat(AP_i), AP_lon(AP_i), 1);
  end
end

%% Save
time_s = sprintf('%d-%02d-%02d_%02d%02d%02d', fix(clock));
filename = [s_path  'wallBps_' time_s];
save(filename, 'short_dbp', 'walls', 'mask');%, 'long_dbp');
toc
```

D.4.2. MATLAB findBreakPt

```
function [dbp walls mask] = findBreakPt(lat_masked, lon_masked, AP_lat,
AP_lon, wLat_cell, wLon_cell)


if ~exist('wLat_cell', 'var')
  wf_dir = '..\GE_kmls\walls\';

  wall_files = dir(wf_dir);

  wLat_cell = cell(1);
  wLon_cell = cell(1);
  %
  % if ~exist('opt', 'var')
  %   opt = 0;
  % end

  build_i = 1;
  for wf_i = 1:length(wall_files)
    % if not a dir
    if ~wall_files(wf_i).isdir
      %read AK Wall coordinates from file
      [wlat, wlon, z] = read_kml([wf_dir wall_files(wf_i).name]);
      wLat_cell{build_i} = wlat;
      wLon_cell{build_i} = wlon;
      build_i = build_i+1;
    end%if not dir
  end%for files
end
% if opt == 0
```

```
  dbp = -1*ones(1, length(lat_masked));
% else
%   dbp = zeros(1, length(lat_masked));
% end

mask = ones(1, length(lat_masked));
walls = zeros(1, length(lat_masked));

%Find the shortest intersection point for each measurement coordinate
for m_pts = 1:length(lat_masked)
  bld_cnt = 0;
  for bldg_i = 1:length(wLat_cell)
    dbp_b = 0;
    if ~isempty(wLat_cell{bldg_i})
      %find intersections
      [lati, loni] = polyxpoly(wLat_cell{bldg_i}, wLon_cell{bldg_i},
...
        [lat_masked(m_pts)  AP_lat], [lon_masked(m_pts)  AP_lon]);

      %If find farthest intersection in building
      for i_pts = 1:length(lati)
        dbp_t = distdim(distance(AP_lat, AP_lon, lati(i_pts), ...
          loni(i_pts)), 'deg','m');
        % find farthest wall
        if dbp_t > dbp_b
          dbp_b = dbp_t;
          walls(m_pts) = length(lati);
          %             kmlStr = ge_point(loni,lati,0);
          %             ge_output('..\GE_kmls\i_points.kml', kmlStr)
        end
      end

    end%if there are walls in bldg
    %see if building intersection is shortest path
    if(dbp_b ~= 0)
      bld_cnt = bld_cnt+1;
      if bld_cnt > 1
        mask(m_pts) = 0;
      end
    end

    if (dbp_b < dbp(m_pts)) || (dbp_b ~= 0)
      dbp(m_pts) = dbp_t;
      walls(m_pts) = length(lati);
    end
  end %for num buildings

  %Check to see if no walls are found
  if dbp(m_pts) == 0
    display('No wall intersection found');
    dbp(m_pts) = distdim(distance(AP_lat, AP_lon, lat_masked(m_pts),
...
      lon_masked(m_pts)), 'deg','m');
  end
end % for num of measurement points
```

D.5. MATLAB Find Performance findProbAgivenB Function

```
function [OPm n] = findProbAgivenB_v2(A, B, AP, files, mask, opt)

num_same = 0;
t_sum_B = 0;
t_length = 0;

i = 1;
AA = [];
BB = [];
if ~exist('opt', 'var')
  for f = files
    if 1%sum(mask{AP, f}) > 1
      A_t  = A{AP, f}';
%       AA = [AA; A_t(logical(mask{AP, f}))];
AA = [AA; A_t];
    end
    BB = [BB; B{AP, f}];
  end
else
  for f = files
    for ap_i =AP
      if 1%sum(mask{ap_i, f}) > 1
        A_t  = A{ap_i, f}';
%         AA = [AA; A_t(logical(mask{ap_i, f}))];
        AA = [AA; A_t];
      end
      %       AA = [AA; A{ap_i, f}'];
      BB = [BB; B{ap_i, f}];
    end
  end
end



%   AA = A';
%   BB = B;
%   AA(AA<0) = 1;
%   BB(BB<0) = 1;
  AA(~isnan(AA)) = 1;
  BB(~isnan(BB)) = 1;
  AA(isnan(AA)) = 0;
  BB(isnan(BB)) = 0;
%   sum(AA)
%   t_sum_B = t_sum_B+ sum(BB==1);
  A1_i_B1 = AA==1 & BB==1;
  A0_i_B0 = AA==0 & BB==0;
  A_u_B = AA==1 | BB==1;
  A_i_nB = AA==1 & BB==0;
  A_e_B = AA==BB;
%   t_length = t_length + length(n);
%   num_same = num_same + sum(n);
  p_A1_i_B1 = sum(A1_i_B1)/length(A1_i_B1);
  p_A0_i_B0 = sum(A0_i_B0)/length(A0_i_B0);
  p_A_u_B = sum(A_u_B)/length(A_u_B);
  p_A_e_B = sum(A_e_B)/length(A_e_B);
  p_A_i_nB = sum(A_i_nB)/length(A_i_nB);
  p_B = sum(BB)/length(BB);
  p_A = sum(AA)/length(AA);
  p_A_g_B = p_A1_i_B1/p_B;
  p_B0_g_A0 = p_A0_i_B0/(1-p_A);
  p_B1_g_A1 = p_A1_i_B1/p_A;
  p_A_g_nB = p_A_i_nB/(1-p_B);
  n =p_A/(1-p_A);
  OPm(i) = (p_B0_g_A0+p_B1_g_A1)/2;
  i = i+1;
```

D.6. MATLAB Channel Model Parameter Optimizer

D.6.1. MATLAB runOPChanModels

```
clc
clear
close all
format long

%% Params
models = ['Y']    %
sigma1 = 5;
smp_delta = 1;
a1 = 5.6;
a2 = 4.7;

%% constants
% FileName = 'APdatabaseOUT.csv';
PathName = '..\WirelessMon Data\';
outputPath = '..\GE_kmls\';
s_path = '..\MAT_data\';

%% get data

%Get AP data
[AP, AP_MAC, AP_lat, AP_lon, AP_fl] = readAPdb('WPI - Access Points
List.csv', '..\');

%Get Empirical data
files = dir(s_path);
datenum = 0;
emp_fn = 'Emp';
for file_i = 1:length(files)
  if strncmp(emp_fn, files(file_i).name, length(emp_fn))
```

```
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  error('No processed empirical data run "runEmpData"');
end

display(['Loading ' fname]);
E_data = load([s_path fname])
clear fname

% Get Wall breakpoint data
files = dir(s_path);
datenum = 0;
wb_fn = 'wallBps';
for file_i = 1:length(files)
  if strncmp(wb_fn, files(file_i).name, length(wb_fn))
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  warning('No processed Wall breakpoint data run "findWallBPs"');
else

  display(['Loading ' fname]);
  Wbps_data = load([s_path fname])
  clear fname
end

%% build shadow fading
% sf = makeSF(5, 1);
sf = zeros(1000,1);
%% Process Channel Models

tic

RSS_size = size(E_data.RSSI_on_path);
for m = 1:length(models);

  %   if models(m) == 'M' ||models(m) == 'U'
  %     a1 = 2:0.1:5;
  %     a2 = 2:0.1:5;
  %   elseif models(m) == 'N'
  %     a1 = 1;
  %     a2 = 2:0.1:5;
  %   elseif models(m) == 'O'
  %     a1 = 1;
  %     a2 = 25:1:50;
  %   elseif models(m) == 'Q' ||models(m) == 'R' ||models(m) == 'P'
  %     a1 = 2:0.1:5;
  %     a2 = 25:1:50;
  %   end
  M_data = struct('Lp', {}, 'a1', {}, 'a2', {});

  for fnum = 1:RSS_size(2)
    fprintf('fnum %d of %d\n', fnum, RSS_size(2))

    for APsel = 1:RSS_size(1)
      s = sprintf('AP %2d', APsel);
      fprintf(s)
      if 1%sum(Wbps_data.mask{APsel, fnum}) > 1
        mask = logical(Wbps_data.mask{APsel, fnum});
        lat_masked = E_data.unique_lat{fnum};
        lon_masked = E_data.unique_lon{fnum};
%         lat_masked = lat_masked(mask);
%         lon_masked = lon_masked(mask);
        wbps = Wbps_data.short_dbp{APsel, fnum};
%         wbps = wbps(mask);


        dist = zeros(length(lat_masked), 1);
        for d_i = 1: length(lat_masked)
          dist(d_i) = distdim(distance(AP_lat(APsel), AP_lon(APsel),
lat_masked(d_i), ...
            lon_masked(d_i)), 'deg','m');
        end


        for a1_i = 1:length(a1)
%           a1s = sprintf('a1 %d of %d', a1_i, length(a1));
%           fprintf(a1s)
          for a2_i = 1:length(a2)

            M_data(a1_i, a2_i).Lp{APsel, fnum} = chanModel(lat_masked,
...

              lon_masked, models(m), AP_lat(APsel), AP_lon(APsel), ...
              AP_fl(APsel), wbps, dist, sf, a1(a1_i), a2(a2_i));
            M_data(a1_i, a2_i).a1 = a1(a1_i);
            M_data(a1_i, a2_i).a2 = a2(a2_i);

          end %a1
%
%           for i = 1:length(a1s)
%             fprintf('\b')
%           end
        end%a2
      end % if there are non-masked points

      %clear line
      for i = 1:length(s)
        fprintf('\b')
      end

    end % for APS
```

```
%     fprintf('\n')
  end %for files

  %% Save
  time_s = sprintf('%d-%02d-%02d_%02d%02d%02d', fix(clock));
  filename = [s_path  'OPM1_nomask_' time_s models(m)];
  save(filename, 'M_data', 'models');
end%for models
toc
```

D.6.2. MATLAB runOpProbs

```
if exist('M', 'var')
  keep('M')
else
  clear
end
clc
format short
%%

s_path = '..\MAT_data\';
%% get data

%Get AP data
[AP, AP_MAC, AP_lat, AP_lon] = readAPdb('WPI - Access Points List.csv',
'..\');

%Get Empirical data
files = dir(s_path);
datenum = 0;
fn = 'Emp';
for file_i = 1:length(files)
  if strncmp(fn, files(file_i).name, length(fn))
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  error('No processed empirical data run "runEmpData"');
end

display(['Loading ' fname]);
E_data = load([s_path fname])
clear fname

%Get Model data
if ~exist('M', 'var')
  files = dir(s_path);
  datenum = 0;
  fn = 'OPM1_';
  for file_i = 1:length(files)
    if strncmp(fn, files(file_i).name, length(fn))
      if files(file_i).datenum > datenum
        fname = files(file_i).name;
      end
    end
  end

  display(['Loading ' fname]);
  M = load([s_path 'OPM1_nomask_2009-04-26_150537Y'])
  clear fname
end
M_data = M.M_data;
clear M
%Get 802 Model data
files = dir(s_path);
datenum = 0;
fn = '802M';
for file_i = 1:length(files)
  if strncmp(fn, files(file_i).name, length(fn))
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  error('No processed 802.11 Model data run "runChanModels"');
end

display(['Loading ' fname]);
M802_data = load([s_path fname])
clear fname

%Get Wall breakpoint data
files = dir(s_path);
datenum = 0;
wb_fn = 'wallBps';
for file_i = 1:length(files)
  if strncmp(wb_fn, files(file_i).name, length(wb_fn))
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  warning('No processed Wall breakpoint data run "findWallBPs"');
else

  display(['Loading ' fname]);
  Wbps_data = load([s_path fname])
  clear fname
end
```

```
% JTC_data = load([s_path 'JTC_2009-03-01_152231'])
%Get JTC data
files = dir(s_path);
datenum = 0;
fn = 'JTC';
for file_i = 1:length(files)
  if strncmp(fn, files(file_i).name, length(fn))
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  warning('No processed JTC_data');
else

  display(['Loading ' fname]);
  JTC_data = load([s_path fname])
  clear fname
end

%%

Lp_size = size(M802_data.Lp);

LpC= cell(Lp_size(1), Lp_size(2));
LpD= cell(Lp_size(1), Lp_size(2));
LpE= cell(Lp_size(1), Lp_size(2));
LpF= cell(Lp_size(1), Lp_size(2));
LpJTC= cell(Lp_size(1), Lp_size(2));
M_size = size(M_data);

for APsel = 1:Lp_size(1)
  for fnum = 1:Lp_size(2)
    LpC{APsel,fnum} = scale_LP(M802_data.Lp{APsel,fnum,1});
    LpD{APsel,fnum} = scale_LP(M802_data.Lp{APsel,fnum,2});
    LpE{APsel,fnum} = scale_LP(M802_data.Lp{APsel,fnum,3});
    LpF{APsel,fnum} = scale_LP(M802_data.Lp{APsel,fnum,4});
    LpJTC{APsel,fnum} = scale_LP(JTC_data.M_data.Lp{APsel,fnum});
    for a1_i = 1:M_size(1)
      for a2_i = 1:M_size(2)
        M_data(a1_i, a2_i).Lp{APsel, fnum} = scale_LP(M_data(a1_i,
a2_i).Lp{APsel, fnum});
      end
    end
  end
end


%%

opt_P = 0;%zeros(1, Lp_size(1));
fRange = 1:Lp_size(2)
```

```
APrange = [1:3 5:9 11 13:29 31:35 37:41 43 45:50 52 53 55:58];
ii = 1;
for fnum = 1:1
%   fprintf('fnum %d of %d\n', fnum, Lp_size(2))
  for APsel = 1:1
%     fprintf('AP %2d of %d\n', APsel, Lp_size(1));
    if APsel == 4
      for a_i = 1:961;
        ii = ii+1;
      end%alphas
    else
      for a1_i = 1:M_size(1)
        for a2_i = 1:M_size(2)

          P_M(a1_i, a2_i) = findProbAgivenB_v2(E_data.RSSI_on_path,
M_data(a1_i, a2_i).Lp, APrange, fRange , Wbps_data.mask, true);


          if P_M(a1_i, a2_i) > opt_P(APsel)%round(P_M(a1_i, a2_i)*100)
> round(opt_P(APsel)*100)
            opt_P(APsel) = P_M(a1_i, a2_i);
            opt_a1(APsel) = M_data(a1_i, a2_i).a1;
            opt_a2(APsel) = M_data(a1_i, a2_i).a2;
            opt_a1_i = a1_i;
            opt_a2_i = a2_i;
          end

        end
      end%alphas

      [P_EM_C(APsel) n] = findProbAgivenB_v2(E_data.RSSI_on_path, LpC,
APrange, fRange, Wbps_data.mask, true);
      P_EM_D(APsel) = findProbAgivenB_v2(E_data.RSSI_on_path, LpD,
APrange, fRange, Wbps_data.mask, true);
      P_EM_E(APsel) = findProbAgivenB_v2(E_data.RSSI_on_path, LpE,
APrange, fRange, Wbps_data.mask, true);
      P_EM_F(APsel) = findProbAgivenB_v2(E_data.RSSI_on_path, LpF,
APrange, fRange, Wbps_data.mask, true);
      P_EM_JTC(APsel) = findProbAgivenB_v2(E_data.RSSI_on_path, LpJTC,
APrange, fRange, Wbps_data.mask, true);
    end
  end%AP

end%files
opt_a1
opt_a2
opt_a1_i
opt_a2_i
opt_P = opt_P'

P_EM_C = P_EM_C'
P_EM_D = P_EM_D'
P_EM_E = P_EM_E'
P_EM_F = P_EM_F'
P_EM_JTC = P_EM_JTC'
```

```
opt_P/P_EM_C
figure(1)
hold on
plot(2:0.1:10, P_M)
ylabel('Performance (%)')
xlabel('Distance-Power Gradient')
grid on
```

D.7. MATLAB Channel Model Simulator

```
clc
clear
close all
format long

%% Params
models = ['CDEF']    %802.11 model'C' 'D' 'E' 'F'

%% constants
% FileName = 'APdatabaseOUT.csv';
PathName = '..\WirelessMon Data\';
outputPath = '..\GE_kmls\';
s_path = '..\MAT_data\';

%% get data

%Get AP data
[AP, AP_MAC, AP_lat, AP_lon, AP_fl] = readAPdb('WPI - Access Points
List.csv', '..\');

%Get Empirical data
files = dir(s_path);
datenum = 0;
emp_fn = 'Emp3';
for file_i = 1:length(files)
  if strncmp(emp_fn, files(file_i).name, length(emp_fn))
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  error('No processed empirical data run "runEmpData"');
end

display(['Loading ' fname]);
E_data = load([s_path fname])
clear fname
```

```
%Get Wall breakpoint data
files = dir(s_path);
datenum = 0;
wb_fn = 'wallBps';
for file_i = 1:length(files)
  if strncmp(wb_fn, files(file_i).name, length(wb_fn))
    if files(file_i).datenum > datenum
      fname = files(file_i).name;
    end
  end
end

if ~exist('fname', 'var')
  warning('No processed Wall breakpoint data run "findWallBPs"');
else

  display(['Loading ' fname]);
  Wbps_data = load([s_path fname])
  clear fname
end
%%
% sf = zeros(1,1000);
sf = makeSF(5, 1);
%% Process Channel Models
tic
RSS_size = size(E_data.RSSI_on_path);
Lp = cell(RSS_size(1), RSS_size(2),length(models));

for fnum = 1:RSS_size(2)
  fprintf('fnum %d of %d\nAP ', fnum, RSS_size(2))

  for APsel = 1:RSS_size(1)
    s = sprintf('%2d of %d ', APsel, RSS_size(1));
    fprintf(s)
%     if sum(Wbps_data.mask{APsel, fnum}) > 1
      mask =
ones(size(E_data.unique_lat{fnum}));%logical(Wbps_data.mask{APsel,
fnum});
      lat_masked = E_data.unique_lat{fnum};
      lon_masked = E_data.unique_lon{fnum};
%       lat_masked = lat_masked(mask);
%       lon_masked = lon_masked(mask);
      wbps = Wbps_data.short_dbp{APsel, fnum};
%       wbps = wbps(mask);
%       y = zeros(length(lat_masked), 1);
%
%       for y_i = 1: length(lat_masked)
%         if y_i == 1
%           y(1) = distdim(distance(lat_masked(2), lon_masked(2),
lat_masked(1), ...
%             lon_masked(1)), 'deg','m');
%         else
%           y(y_i) = distdim(distance(lat_masked(y_i), lon_masked(y_i),
lat_masked(y_i-1), ...
%             lon_masked(y_i-1)), 'deg','m');
%         end
%       end

      dist = zeros(length(lat_masked), 1);
      for d_i = 1: length(lat_masked)
        dist(d_i) = distdim(distance(AP_lat(APsel), AP_lon(APsel),
lat_masked(d_i), ...
          lon_masked(d_i)), 'deg','m');
      end

      for model_i = 1:length(models)
        fprintf(models(model_i))
        Lp{APsel, fnum, model_i} = chanModel(lat_masked, ...
          lon_masked, models(model_i), ...
          AP_lat(APsel), AP_lon(APsel), AP_fl(APsel), wbps, dist, sf);
      end
      %clear line
      for i = 1:length(models)
        fprintf('\b')
      end
%     end % if there are non-masked points

    %clear line
    for i = 1:length(s)
      fprintf('\b')
    end

  end % for APS
  fprintf('\n')
end %for files

%% Save
time_s = sprintf('%d-%02d-%02d_%02d%02d%02d', fix(clock));
if strcmp(models, 'CDEF')
  filename = [s_path  '802M3_nomask_' time_s];
else
  filename = [s_path  'newM_' time_s models];
end
save(filename, 'Lp', 'models');
toc
```

D.7.1. MATLAB ChanModel

```
function Lp = chanModel(lat_masked, lon_masked, model, AP_lat, ...
  AP_lon, AP_fl, Wbps_data, dist, sf, in_a1, in_a2)
% flr_h =3;
L0=40;
walls = zeros(1, length(lat_masked));
wall_lp = 0;
pts = length(lat_masked);
closest_wall = 0;
```

```
farthest_wall = 1;
model_type = 0;

%define model types
MT_802 = 1;
MT_802_2bp = 2;
MT_JTC = 3;
MT_JTC_1bp = 4;
MT_OJTC = 5;
MT_OJTC_1bp = 6;
dwbp = Wbps_data;
switch model
  case 'C'
    model_type = MT_802;
    dbp= 5*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
  case 'D'
    model_type = MT_802;
    dbp= 10*ones(1, pts);
    a1 = 2;
    a2=3.5;
  case 'E'
    model_type = MT_802;
    dbp= 20*ones(1, pts);
    a1 = 2;
    a2=3.5;
  case 'F'
    model_type = MT_802;
    dbp= 30*ones(1, pts);
    a1 = 2;
    a2=3.5;
  case 'G'
    model_type = MT_802;
    if exist('Wbps_data', 'var')
      dbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
closest_wall);
    end

    a1 = 3.4;
    a2 = 2;
    wall_lp = 0;
  case 'H'
    model_type = MT_802;
    if exist('Wbps_data', 'var')
      dbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = 3;
    a2 = 2;
    wall_lp = 0;
  case 'I'
    model_type = MT_802;
    if exist('Wbps_data', 'var')
      dbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
closest_wall);
    end
    a1 = 2;
    a2 = 3.5;
    wall_lp = 0;
  case 'J'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    dbp = 10*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
    a3 = 3.3;
    wall_lp = 20;

  case 'K'
    model_type = MT_802;
    if exist('Wbps_data', 'var')
      dbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
closest_wall);
    end
    a1 = 3.5;
    a2 = 2;
    wall_lp = 10;
    walls= ones(1, pts);
  case 'L'
    model_type = MT_802;
    if exist('Wbps_data', 'var')
      dbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = in_a1;
    a2 = in_a2;
    wall_lp = 0;
  case 'M'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon, 
farthest_wall);
    end
    dbp = 5*ones(1, pts);
    a1 = 2;
    a2 = in_a1;
    a3 = in_a2;
    wall_lp = 0;
  case 'N'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon, 
farthest_wall);
    end
    dbp = 10*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
    a3 = in_a1;
    wall_lp = 12.8;
    walls = ones(1, length(lat_masked));
  case 'O'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon, 
farthest_wall);
    end
    dbp = 10*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
    a3 = 2;
    wall_lp = in_a2;
    walls = ones(1, length(lat_masked));
  case 'P'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon, 
farthest_wall);
    end
    dbp = 10*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
    a3 = in_a1;
    wall_lp = 0;
    walls = ones(1, length(lat_masked));
  case 'Q'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    dbp = 5*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
    a3 = in_a1;
    wall_lp = 0;
    walls = ones(1, length(lat_masked));
  case 'R'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    dbp = 20*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
    a3 = in_a1;
    wall_lp = in_a2;
    walls = ones(1, length(lat_masked));
  case 'S'
    model_type = MT_OJTC_1bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = 3;
    a2 = in_a1;
    num_fl = [0 0 1 2];
    wall_lp = 0;
    walls = ones(1, length(lat_masked));
  case 'T'
    model_type = MT_OJTC_1bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = 3;
    a2 = in_a1;
    num_fl = [0 0 1 2];
    wall_lp = 12.8;
    walls = ones(1, length(lat_masked));
  case 'U'
```

```
    model_type = MT_JTC_1bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = 3;
    a2 = in_a1;
    num_fl = [0 0 0 0];
    AP_fl= ones(size(AP_fl));
    wall_lp = 0;
    walls = ones(1, length(lat_masked));
  case 'V'
    model_type = MT_JTC_1bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = 3;
    a2 = in_a1;
    num_fl = [1 0 1 2];
    wall_lp = 0;
    walls = ones(1, length(lat_masked));
  case 'W'
    model_type = MT_JTC_1bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = 3;
    a2 = in_a1;
    num_fl = [1 0 1 2];
    wall_lp = 12.8;
    walls = ones(1, length(lat_masked));
  case 'X'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    dbp = 5*ones(1, pts);
    a1 = 2;
    a2 = 3.5;
    a3 = in_a1;
    wall_lp = 12.8;
    walls = ones(1, length(lat_masked));
  case 'A' %JTC frat
    model_type = MT_JTC;
    a1 = 3;
    num_fl = [1 0 1 2];
  case 'Z' %JTC orginal
    model_type = MT_OJTC;
    a1 = 3;
    num_fl = [0 0 1 2];
  case 'B' %JTC no floors
    model_type = MT_JTC_1bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    a1 = 3;
    a2 = in_a1;
    num_fl = [0 0 0 0];
    AP_fl= ones(size(AP_fl));
    wall_lp = 12.8;
    walls = ones(1, length(lat_masked));
  case 'Y'
    model_type = MT_802_2bp;
    if exist('Wbps_data', 'var')
      dwbp = Wbps_data;
    else
      warning('no stored wall breakpoints!!');
      dwbp = findBreakPt(lat_masked, lon_masked, AP_lat, AP_lon,
farthest_wall);
    end
    dbp = 5*ones(1, pts);
    a1 = 2;
    a2 = in_a2;
    a3 = in_a1;
    wall_lp = 12.8;
    walls = ones(1, length(lat_masked));
end
if ~exist('dist', 'var')
  make_dist = true;
  dist = zeros(pts, 1);
else
  make_dist = false;
end

% init path loss array
Lp = zeros(pts, 1);

%loop over sample points
for i = 1:pts

  if make_dist
    dist(i) = distdim(distance(AP_lat, AP_lon, lat_masked(i), ...
      lon_masked(i)), 'deg','m');
  end
  switch model_type
    case 1 %802.11

      if (dist(i) < dbp(i) || dbp(i) == 0)
        Lp(i) = L0 + 10*a1*log10(dist(i))...
          + walls(i)*wall_lp...
          + sf(round(dist(i)));
        %       keyboard
      else
        Lp(i) = L0 + 10*a1*log10(dbp(i))...
          + 10*a2*log10(dist(i)/dbp(i))...
          + walls(i)*wall_lp...
          + sf(round(dist(i)));
      end


    case 2 %802.11 with wall breakpoint
      %if AP is less than dbp meters from the measurement point and no
wall
      %intersection was found because path was close to building
      if (dist(i) < dbp(i)) && dwbp(i) == 0
        Lp(i)= L0 + 10*a1*log10(dist(i))...
          + walls(i)*wall_lp...
          + sf(round(dist(i)));

        %if AP is less than dbp meters away from the wall use only dwbp
      elseif (dbp(i) > dwbp(i)) && dwbp(i) ~= 0
        Lp(i)= L0 + 10*a1*log10(dwbp(i))...
          + 10*a3*log10(dist(i)/dwbp(i))...
          + walls(i)*wall_lp...
          + sf(round(dist(i)));

        %if no wall intersect was found because path was close to
building
      elseif dwbp(i) == 0
        Lp(i)= L0 + 10*a1*log10(dbp(i))...
          + 10*a2*log10(dist(i)/dbp(i))...
          + walls(i) * wall_lp...
          + sf(round(dist(i)));

        %normal case use both breakpoints in order
      else
        Lp(i)= L0 + 10*a1*log10(dbp(i))...
          + 10*a2*log10(dwbp(i)/dbp(i))...
          + 10*a3*log10(dist(i)/dwbp(i)) ...
          + walls(i)*wall_lp...
          + sf(round(dist(i))) ;
      end
    case 3 %JTC
      fRat = dwbp(i)/(dist(i)-dwbp(i));
      if AP_fl == 1 || AP_fl == 0
        Lp(i) = L0 + 10*a1*log10(dist(i))...
          + sf(round(dist(i)));
      elseif AP_fl == 2
        if fRat <= 1
          Lp(i) = L0 + 10*a1*log10(dist(i))...
            + sf(round(dist(i)));
        else% 1 floor inbetween
          Lp(i) = L0 + 15 ...
            + 10*a1*log10(dist(i))...
            + sf(round(dist(i)));
        end
      else
        if fRat <= 0.5
          Lp(i) = L0 + 10*a1*log10(dist(i))...
            + sf(round(dist(i)));
        elseif fRat <= 2 % 1 floor inbetween
          Lp(i) = L0 + 15 ...
            + 10*a1*log10(dist(i))...
            + sf(round(dist(i)));
        else %2 floors betwen
          Lp(i) = L0 + 15+4*(2-1) ...
            + 10*a1*log10(dist(i))...
            + sf(round(dist(i)));
        end
      end

    case 4 %JTC with wall breakpoint
      fRat = dwbp(i)/(dist(i)-dwbp(i));
      if AP_fl == 1 || AP_fl == 0
        if (dist(i) < dwbp(i) || dwbp(i) == 0)
          Lp(i) = L0 + 10*a1*log10(dist(i))...
            + walls(i)*wall_lp...
            + sf(round(dist(i)));
          keyboard
        else
          Lp(i) = L0 + 10*a1*log10(dwbp(i))...
            + 10*a2*log10(dist(i)/dwbp(i))...
            + walls(i)*wall_lp...
            + sf(round(dist(i)));
        end
      elseif AP_fl == 2
        if fRat <= 1
          if (dist(i) < dwbp(i) || dwbp(i) == 0)
            Lp(i) = L0 + 10*a1*log10(dist(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
            keyboard
          else
            Lp(i) = L0 + 10*a1*log10(dwbp(i))...
              + 10*a2*log10(dist(i)/dwbp(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
          end
        else
          if (dist(i) < dwbp(i) || dwbp(i) == 0)
            Lp(i) = L0 + 15+4*(num_fl(AP_fl+1)-1)...
              + 10*a1*log10(dist(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
            keyboard
          else
            Lp(i) = L0 + 15+4*(num_fl(AP_fl+1)-1)...
              + 10*a1*log10(dwbp(i))...
              + 10*a2*log10(dist(i)/dwbp(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
          end
        end
      else
        if fRat <= 0.5
          if (dist(i) < dwbp(i) || dwbp(i) == 0)
            Lp(i) = L0 + 10*a1*log10(dist(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
            keyboard
          else
            Lp(i) = L0 + 10*a1*log10(dwbp(i))...
              + 10*a2*log10(dist(i)/dwbp(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
          end
        elseif fRat <= 2
          if (dist(i) < dwbp(i) || dwbp(i) == 0)
            Lp(i) = L0 + 15 ...
              + 10*a1*log10(dist(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
            keyboard
          else
            Lp(i) = L0 + 15 ...
              + 10*a1*log10(dwbp(i))...
              + 10*a2*log10(dist(i)/dwbp(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
          end
        else
          if (dist(i) < dwbp(i) || dwbp(i) == 0)
            Lp(i) = L0 + 15+4*(num_fl(AP_fl+1)-1)...
              + 10*a1*log10(dist(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
            keyboard
          else
            Lp(i) = L0 + 15+4*(num_fl(AP_fl+1)-1)...
              + 10*a1*log10(dwbp(i))...
              + 10*a2*log10(dist(i)/dwbp(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
          end
        end
      end
    case 5 %JTC
      if AP_fl == 1 || AP_fl == 0
        Lp(i) = L0 + 10*a1*log10(dist(i))...
          + sf(round(dist(i)));
      else
        Lp(i) = L0 + 15+4*(num_fl(AP_fl+1)-1) ...
```

```
          + 10*a1*log10(dist(i))...
          + sf(round(dist(i)));
      end
    case 6 %JTC + WB
      if AP_fl == 1 || AP_fl == 0
          if (dist(i) < dwbp(i) || dwbp(i) == 0)
            Lp(i) = L0 + 10*a1*log10(dist(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
            keyboard
          else
            Lp(i) = L0 + 10*a1*log10(dwbp(i))...
              + 10*a2*log10(dist(i)/dwbp(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
          end
      else
          if (dist(i) < dwbp(i) || dwbp(i) == 0)
            Lp(i) = L0 + 15+4*(num_fl(AP_fl+1)-1) ...
              + 10*a1*log10(dist(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
            keyboard
          else
            Lp(i) = L0 + 15+4*(num_fl(AP_fl+1)-1) ...
              + 10*a1*log10(dwbp(i))...
              + 10*a2*log10(dist(i)/dwbp(i))...
              + walls(i)*wall_lp...
              + sf(round(dist(i)));
          end

      end
    otherwise
      error('No model type selected')
  end %switch
end %for pts
```

D.8. MATLAB Plot on Google Earth

```
fnum = 1;
for fnum = 1:2
  AP = 1;
  % files(fnum+3).name(1:end-4)
  outputPath = '..\GE_kmls\';
  % files = dir('..\Scanning_data\file_2_run1');


%   RSS = 100+E_data.RSSI_on_path{AP,fnum};
  % RSS = 100-Mnew_data.Lp{AP,fnum};
  % RSS = 100+M_d(19, 21).Lp{AP,fnum};
  RSS = 100+LpM{AP,fnum};
  % RSS = 100+LpF{AP,fnum};
  RSS(isnan(RSS)) = -1;


  lat = E_data.unique_lat{fnum};
  % lat = lat(logical(Wbps_data.mask{AP,fnum}));
  % lat = lat(RSS~=0);
  lon = E_data.unique_lon{fnum};
  % lon = lon(logical(Wbps_data.mask{AP,fnum}));
  % lon = lon(RSS~=0);
  % RSS(logical(Wbps_data.mask{AP,fnum})) = 20;
  % RSS(logical(~Wbps_data.mask{AP,fnum})) = -1;
  kmlStr1 = ge_plot3(lon, lat, RSS);
  % ge_output([outputPath files(fnum+3).name(1:end-4) '_path.kml'],
kmlStr1)
  ge_output([outputPath 'SigPath_ap' num2str(AP) 'f' num2str(fnum)
'_dist.kml'], kmlStr1)
end
```

References

[1] Workshop on Opportunistic RF Localization for Next Generation Wireless Devices; Future Directions, Technologies, Standards and Applications; June 16-17, 2008; Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA, USA

[2] Stantchev, V., Schulz, T., Trung Dang Hoang, and Ratchinski, I., "Optimizing Clinical Processes with Position-Sensing," IT Professional , vol.10, no.2, pp.31-37, March-April 2008

[3] Bahl, P. and Padmanabhan, V.N., "RADAR: an in-building RF-based user location and tracking system ," INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.2, no., pp.775-784 vol.2, 2000

[4] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen. A Probabilistic Approach to WLAN User Location Estimation. *International Journal of Wireless Information Networks*, 9(3), July 2002.

[5] LaMarca, A., et al., Place lab: Device positioning using radio beacons in the wild. In Proceedings of International Conference on Pervasive Computing (Pervasive) (June 2005).

[6] Minkyong Kim, Jeffrey J. Fielding, and David Kotz, Risks of using AP locations discovered through war driving, In Proceedings of The 4th International Conference on Pervasive Computing (Pervasive 2006) pp. 67-82 (2006)

[7] Pahlavan, K., Xinrong Li, and Makela, J.P., "Indoor geolocation science and technology," Communications Magazine, IEEE , vol.40, no.2, pp.112-118, Feb 2002

[8] M. Takai and J. Martin. Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks. Proc. Of MobiHoc 2001, Oct. 2001.

[9] Pahlavan, K. and Levesque, A., Wireless Information Networks: John Wiley & Sons, 1995.

[10] Y. Cheng, Y. Chawathe, A. LaMarca and J. Krumm, Accuracy Characterization for Metropolitan-Scale Wi-Fi Localization, in: *Proceedings MobiSys* (2005) pp. 233–245.

[11] Hightower, J. & Borriello, G. Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp) (Springer-Verlag).(2004)

[12] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian Filtering for location estimation. IEEE Pervasive Computing, 2(3):24-33, July-September 2003.

[13] A. DeToledo, A. Turkmani, and J. Parsons, “Estimating coverage of radio transmission into and within buildings at 900, 1800, and 2300 mhz,” *IEEE Personal Communications*, vol. 5, no. 2, pp. 40–47, 1998.

[14] J. B. Andersen, T. S. Rappaport, and S. Yoshida, “Propagation measurements and models for wireless communications channels,” *IEEE Commun. Mag.*, vol. 33, no. 1, Jan. 1995.

[15] V. Erceg, L. Schumacher, P. Kyritsi, D. S. Baum, A. Molisch, and A. Y. Gorokhov, Indoor MIMO LAN Channel Models, in IEEE Stand. 802.11-03/161r0, Mar. 2003.

[16] M. Barkat, *Signal Detection and Estimation*, Artech House, Inc., 2005