A SEQUENTIAL DETECTION APPROACH TO INDOOR POSITIONING USING RSS-BASED FINGERPRINTING

by

Negar Etemadyrad

A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Master of Science
Electrical Engineering

Committee:

_____	Dr. Jill K. Nelson, Thesis Director
_____	Dr. Bernd-Peter Paris, Committee Member
_____	Dr. Daniel M. Lofaro, Committee Member
_____	Dr. Monson Hayes, Chairman, Department of Electrical and Computer Engineering
_____	Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering

Date: _____ Spring Semester 2017
George Mason University
Fairfax, VA

A Sequential Detection Approach to Indoor Positioning Using RSS-Based Fingerprinting

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science at George Mason University

By

Negar Etemadyrad
Bachelor of Science
Ferdowsi University of Mashhad, 2013

Director: Dr. Jill K. Nelson, Associate Professor
Department of Electrical and Computer Engineering

Spring Semester 2017
George Mason University
Fairfax, VA

Dedication

To maman, baba, and Neshat.

Acknowledgments

I would like to sincerely thank my advisor, Dr. Jill Nelson, for her contributions, and guidance throughout my master's studies. I would also like to express my gratitude to the members of my committee, Dr. Bernd-Peter Paris, and Dr. Daniel Lofaro, for their time, and valuable feedback.

I am thankful to my parents, for their love and trust in me, and thanks to my sister, Neshat, for her care and encouragement.

Finally, thanks to Marjan, Nazanin, and all my dearest friends, for the fun and support.

Table of Contents

List of Figures

Abstract

A SEQUENTIAL DETECTION APPROACH TO INDOOR POSITIONING USING RSS-BASED FINGERPRINTING

Negar Etemadyrad

George Mason University, 2017

Thesis Director: Dr. Jill K. Nelson

Indoor positioning has received significant research and commercial attention during the past two decades. Real world applications include navigating in hospitals, office buildings, warehouses, and parking garages. In this work, we focus on localizing a robot navigating an unfamiliar building. The proposed approach uses received signal strength (RSS) fingerprinting from nearby Wi-Fi access points (APs). RSS fingerprinting approaches are common in indoor localization, and their popularity is in part due to the relatively low hardware cost, taking advantage of the equipment that generates Wi-Fi signals in a wireless local area network (LAN). Moreover, fingerprinting-based localization has significantly lower computational complexity than competing highly sophisticated mathematical techniques, such as probabilistic methods, compressive sensing positioning, and Hidden Markov Models.

This thesis presents a sequential detection approach to RSS-based positioning that employs a Bayesian metric to identify the most likely path traveled by the indoor user, given a time series of RSS measurements. The RSS measurements are collected as the robot travels and are passed as input to the proposed algorithm to identify the best user path estimate. A trellis is used to model different possible paths the user could travel, and the Viterbi algorithm is applied to find the most likely path. As a contribution of this work, a

sequential metric is developed which takes advantage of Bayes' rule in combination with the k-nearest neighbors (KNN) algorithm to approximate the likelihoods, conditioned on observed RSS, of the paths in the trellis.

The performance of the proposed indoor localization algorithm is evaluated using data collected in the Nguyen Engineering building at George Mason University. Data was collected at varying times of day and across several days. Training (offline) RSS data was collected at a grid of location points and stored in a database for use during the test (online) phase. Various combinations of training and test points were drawn randomly from the collected data and provided as input to the KNN algorithm for estimating the conditional likelihood of observations given location. Results show that the sequential detection approach achieves strong performance even when only a small series of RSS measurements are available. In fact, in most cases, there is great improvement in performance, reported both in terms of average distance error and probability of correct path estimation, when a sequence of two sets of RSS measurements is used rather than collecting measurements at only a single location.

Chapter 1: Introduction

A recent growing interest in location-based services motivates the need for developing accurate localization solutions with moderate computational requirements. Among all real world applications, our application of interest lies in robotics. Specifically, we are interested in developing approaches to allow a robot to self localize within a given map. Considerable research and work is done in robotic mapping, where an autonomous robot is designed to localize itself in an area using the corresponding map of the area.

Considering indoor localization as our research area, our goal is to design a system to allow a robot to self localize while navigating an unfamiliar building. There are a wide range of applications in this realm. Examples include navigation in parking lots, hospitals, airports, and transportation stations; location detection of products stored in a warehouse; locating medical personnel or equipment in a hospital, firemen in a building on fire, police dogs trained to find explosives in a building; and finding tagged maintenance tools and equipment scattered around a plant [1]. The information used by these localization systems could be radio waves, magnetic fields, acoustic signals, or other sensory information collected by mobile devices.

Global positioning systems (GPS) are generally not an appropriate option for indoor localization, since the signals will be attenuated and scattered by walls and indoor obstructions. The methods applicable to the indoor localization problem, mainly include non-radio and wireless technologies. The former provides high accuracy but requires costly equipment and installations. Hence, we focus our discussion on wireless technologies, which are the basis for the indoor localization approach we propose. Various wireless systems are suggested by the literature. Depending on the location positioning algorithm and the wireless technology that connects different devices, these technologies are grouped into several classes.

The most common algorithms include triangulation, scene analysis, and proximity, all of which are discussed in more detail in Chapter 2.

With recent advances in research in this field, there is significant commercial interest in development and improvement of indoor localization systems. Various commercial systems, based on several different technologies, are suggested by the market. The most common technologies employed in wireless localization include GPS, radio-frequency identifier (RFID), cellular-based, ultra-wide-band (UWB), wireless local area network (WLAN) and bluetooth. These approaches and systems are discussed in more detail in Chapter 2.

Due to the nature of the indoor environment, time-varying elements such as shadowing, multi-path fading, non-line of-sight (NLOS) path, and the presence of moving objects and reflecting surfaces, lead to variations in RSS (received signal strength) at a fixed location over time. Therefore, developing a sufficiently accurate model for indoor radio propagation is often impractical. To avoid the need for a model, we propose a solution that relies on RF-based scene analysis. RF scene analysis algorithms are based on RSS fingerprinting approaches. In these methods, fingerprints (features which are vectors formed from received signal strength) are collected at various points in the area of interest during an offline phase before running the designed localization approach. In the online phase, when the robot performs self localization, the online measurements are matched to the available fingerprints or database of the area. The robot observes Wi-Fi signals available in the environment using, for example, an antenna that interfaces through a USB connection to a laptop running Linux OS. The RSS is provided using the existing infrastructure, which is an easily accessible and low-cost approach and eliminates the need for additional hardware installation. RSS-based fingerprinting is relatively simple to implement compared to techniques that rely on more sophisticated measurements such as angle-of-arrival (AOA) and time-of-arrival (TOA). This makes it a popular approach since there are often limits on the processing unit of the mobile robot, and hence there is need for a positioning algorithm with moderate computational requirements.

Because physical changes inside the building, as well as possible changes in network

configuration, cause variations in RSS values at a given point, accurate location estimation requires more than instantaneous RSS measurements as input. In addition to considering RSS measurements over time and evaluating the likelihood of a trajectory rather than individual locations, we also construct the localization algorithm to exploit odometry information (such as distance traveled) when available. Including such information allows for the elimination of many paths from consideration in the estimation process, thereby improving performance and reducing complexity.

This thesis proposes a sequential detection based algorithm for estimating the path traveled by a user (robot) given a set of RSS measurements collected as the user moves. The KNN classification method and a Bayesian likelihood metric are employed to formulate a novel framework for indoor localization. The goal is to design an algorithm that can take RSS measurements as input and estimate the robot's location with high accuracy while maintaining moderate computational complexity.

As an example, consider the area of interest, to consist of a simple rectangular hallway, as depicted in Figure 1.1. The access points of the WLAN (wireless local area network) are located inside the area, and their locations may be unknown. By selecting a number of grid points along the hallway (the finer the resolution is desired, the more points must be chosen), the offline and online data collection can be performed. In the former, the robot sits at each of the locations for a short period of time, receiving and recording RSS measurements from APs. In the online phase, the user moves within the hallway while observing RSS in the environment. Figure 1.1 shows three APs generating Wi-Fi signal; however, the total number is generally much larger for a real-world scenario. A detailed explanations of the data collection process used for this work is provided in Chapter 5.

The remainder of the thesis is organized as follows. In Chapter 2, background and related work for indoor localization are discussed. The proposed algorithm is described in Chapter 3, and simulated performance results are included in Chapter 4. The main discussion of the algorithm, simulations and performance results for the experimental data collected are given in Chapter 5. Brief discussion on data analysis and challenges is provided

in Chapter 6. Conclusions and comments on future work form Chapter 7, the last chapter of this thesis.

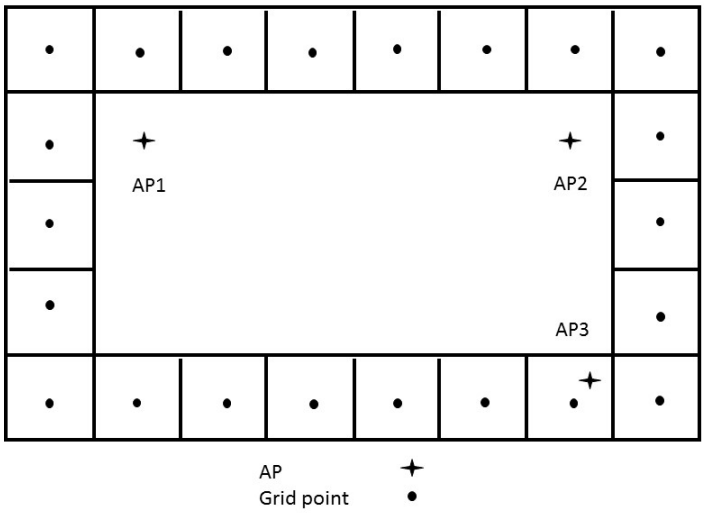


Figure 1.1: An example area of interest for self localization. The dots represent grid points at which off-line measurements are collected. The plus signs represent access points in the environment.

Chapter 2: Background and Related Work

Various RSS-based schemes for indoor localization have been proposed and evaluated in the existing research literature. In a comprehensive overview of wireless indoor positioning solutions by Liu et al. [1], three typical approaches – triangulation, scene analysis, and proximity – are discussed.

Triangulation is based on mathematical and geometric properties and equations. In one derivative of this approach, called lateration techniques, multiple reference points are considered, and the distance of the user from these points is computed. To do so, either the attenuation of the emitted signal is used or the signal velocity is multiplied by travel time. The second branch of triangulation, called angulation techniques is based on angle of arrival estimation. Using directional antennas or an array of antennas, the direction or angle of multiple base stations or beacon stations is determined by drawing the circle radius that connects the mobile user to these stations. The final location estimate is the intersection of these direction lines.

Scene analysis refers to a class of indoor localization techniques that are based on fingerprinting. Fingerprinting refers to techniques that estimate the position of a user by matching a fingerprint composed of signal properties (e.g. RSS values from several access points) to a set of fingerprints in a database. Fingerprinting approaches include an offline survey of the indoor area of interest in addition to an online phase. The former is performed by dividing the space into a grid, or number of discrete points (locations). At each point, the RSS from all visible APs is measured and saved as a reference for future comparison. In the online phase, the user is traveling in the space and relates RSS values it collects in real time to those collected in the offline phase in order to estimate its location. RSS fingerprinting is the most common approach in scene analysis according to [1].

The features included in a fingerprint are characteristics of the RSS that change with location. The largest challenge of this approach is the variations of RSS at a fixed location with time caused by shadowing, multi-path fading, etc. for which developing an analytical model is often impractical. The literature suggests five different location fingerprinting-based approaches including: probabilistic methods, KNN (K nearest neighbors), neural networks, SVM (support vector machines) and SMP (smallest M-vertex polygon).

The final RF-based approach described in [1] is referred to as proximity techniques. A dense grid of antennas with known locations are used to detect the mobile user. The user is estimated to have the same location as the antenna that receives the strongest signal strength.

As described in [1], to evaluate the localization approach used, accuracy alone is not enough. In fact precision, complexity, robustness, scalability, and cost are also considered for comparing different systems, and there is usually a trade-off among these metrics. While accuracy is defined as the average Euclidean distance between the true and estimated location, the precision considers the distribution of the distance error by defining the cumulative density function (CDF) of the error. Complexity is usually defined as software complexity or computing complexity of the algorithm. Considering the influence of the hardware used, evidently, the more powerful the processing unit is, the faster the computations are performed. Accordingly, if there is a central server for processing, the complexity reduces significantly comparing to when the mobile user takes care of all computation using a limited power resource. In many references, complexity is defined as the computing time. However, we could also ignore the hardware used for localization; in which case complexity would be defined as the total number of operations of the algorithm under investigation. Robustness, as defined for positioning techniques, as the ability to localize the user with acceptable accuracy, even when the input to the algorithm is unreliable in some way. For example, some measuring units don't function well and are unable to report accurate data. Scalability of a positioning system defines how well it operates when the scope of the problem changes. In

some cases, performance worsens when the distances between APs and measuring units increases. Cost could include money, time, space, weight and energy. Among all performance metrics, in this work, we have evaluated our proposed method, by reporting accuracy, as described in Chapter 5.

In another survey on indoor positioning by F. Seco et al. [2], mathematical approaches (versus physical model of localization systems) based on RF signals are discussed. These include geometry-based methods, minimization of a cost function, fingerprinting, and Bayesian techniques. In the cost function based approach, minimizing the cost function results in maximizing $p(r|x)$, where r is the measurement vector and x is the current position of the mobile user. In Bayesian methods, the location of the user at each time instant is obtained based on all previous measurements up to that instant. In this case, the localization algorithm, includes a prediction step in which the a priori estimate of the position is obtained based on the previous data. In the correction phase, that follows, the new set of measurements are employed to compute the posterior probability using Bayes rule. These two steps are applied each time a new measurement is reported to update the location of the user. As an advantage of this iterative method, the estimate can be improved each time new data is reported, and unreliable measurements may have a smaller effect on performance. Evaluation of the four aforementioned methods is performed in [2] in terms of the feasibility of each algorithm (under which conditions/constraints does it work well), implementation requirements, and immunity to non-line-of-sight (NLOS) propagation.

Since we propose a fingerprint-based technique for indoor localization, we focus in more detail on related fingerprinting approaches. A relatively simple but well-known solution applicable to location fingerprinting is the K-nearest-neighbors (KNN) classification method. As one of the simplest machine learning algorithms, KNN is used both for classification and regression. In KNN classification, a set of data points are collected which form a database representing all possible classes. Using this set, an unknown data point should be assigned to the class specified by the majority vote of its K nearest neighbors. As shown in Figure 2.1, for example, the green circle is the test point that needs to be classified, and the red

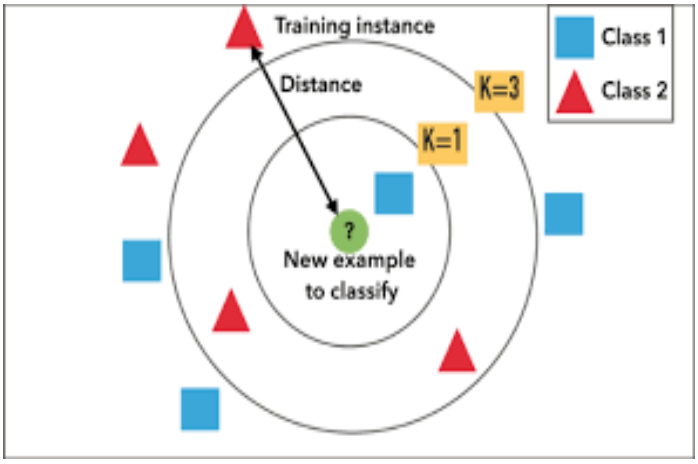


Figure 2.1: An example of KNN classification[3].

triangles and blue squares represent training points from the two possible classes. For K=1, the green circle would be assigned to class 1, while for K=3, there are two triangles versus one square, resulting in the majority vote for class 2.

Applying KNN alone can lead to poor performance, and the accuracy is reported to be low. A main drawback of this algorithm, according to the literature, is that all K neighbors are treated equally without considering the relative distance from the neighbors to the test point. In fact, the closer a neighbor to the test point, the more impact it should have in classifying the test point. Accordingly, many modified KNN algorithms are suggested to improve performance by weighting neighboring points based on their distances to the test point. As one example, Dempster-Shafer Theory (DST), also referred to as evidence theory, is applied [4]. Shafer proposed the theory in 1976 for combining the decision results of multiple classifiers to generate a more reliable classification procedure. As another contribution of [4], DST is applied to imbalanced data sets, as well. When the total number of training points assigned to one class is much larger than the number assigned to the other class, KNN will likely fail. Experimental results show that this problem is solved by using the modified KNN methods proposed in [4].

Projection approaches have been shown to improve performance and reduce complexity in RSS fingerprinting. These methods are performed before running the localization algorithm, as a preprocessing step applied to the data. In [5], MDA (multiple discriminant analysis) and PCA (principal component analysis) are exploited concurrently to form a dynamic hybrid projection (DHP) that is input dependent and solves the problems caused by the weaknesses of both approaches. Specifically, according to [5], PCA ignores the specific RSS structures available at each class, and MDA falls into numerical issues when number of training samples is too small. The projection methods are also reported to solve problems (the curse of dimensionality) caused by the existence of a large number of APs that provide redundant or possibly unreliable information. Such high dimensional fingerprints are impractical in terms of both storage and computation requirements. In DHP, a transition matrix is designed to transform RSS data to a new space, resulting in the formation of regions or classes that are better separated than in the original space. The online measurements are later compared to these new regions to decide the best or closest match. For the estimation (localization) process, the authors in [5] have developed an algorithm based on a Bayesian approach. The prior probability is assumed to be uniform, and the posterior probability is computed using the Gaussian density function, parameters of which are obtained using the maximum likelihood technique.

In [6], a machine learning technique known as CaDet is proposed to reduce measurement vector dimensionality. Combining information theory, cluster analysis, and a decision tree algorithm, an intelligent selection of a subset of the available APs is performed. This approach saves the power for the user by reducing the total amount of computation required.

More complex methods to address the indoor localization challenge are presented in [7], where the authors provide a comprehensive theoretical review of probabilistic approaches including Bayesian models, smoothing, compressive sensing (CS), and random field differentiation. They concentrate on a Kullback-Leibler divergence (KLD) metric that compares multivariate RSS distributions. The four mentioned methods can be categorized as,

A. Bayesian Models: including the Naive Bayes classifier, state-space models and particle

filters, random field differentiation.

B. Comparing RSSI Distributions using multivariate gaussians: presented as model-free smoothing or comparing distributions using Kullback-Leibler divergence.

C. Probabilistic Kernel Regression: as in plain KLD kernel (KL), KLgauss (a kernel built using parametric gaussian), DistMean (a modified version of KLgauss) and a hybrid of KL and DistMean (hybrid).

D. Compressive Sensing

The performance of these algorithms is compared by collecting real data in two different scenarios: dynamic and static (in terms of the movement of the user while collecting the data). The measurements are then used to compare the performance of six algorithms including the four mentioned kernel regression methods, in addition to CS and multivariate gaussian (MvG). The results indicate that for the static scenario, MvG has the best performance, while in the dynamic one, the superiority of simple kernel regression over the rest of the methods is confirmed.

A localization algorithm based on a two-stage (coarse-fine) approach is then proposed in [7]. This includes a coarse localization followed by a fine step. The former includes partitioning the points of grid into several clusters, using affinity propagation algorithm in the offline phase. The coordinates of the mobile user are determined using compressive sensing in the second stage. CS is also used to reconstruct the full database when measurements of only a few points on the grid are available.

CS based positioning has been discussed in detail in [8]. In this method, a sparse vector is introduced to represent the location estimate for the user. A coarse localization indicates in which cluster the user is located; a fine localization follows. The former includes partitioning the points of grid into several clusters using the affinity propagation algorithm in the offline phase. Since the RSS depends on the orientation of the antenna, data collection in the offline phase, in addition to clustering, are performed separately for each orientation. The clustering can cause great localization error if the correct group is not decided, however the

error is equal to the size of the wrong cluster in the worse case. Before running CS on the chosen cluster for deciding the exact location, an AP selection procedure is applied to reduce the dimension of the measurements. The approaches suggested by [8] include strongest APs, fisher criterion, and random combination. Finding and applying the appropriate method results in much less computation and energy required by the limited resources of the mobile user, while running cs in fine localization step when the coordinates of the mobile user is determined. CS is also used in [8] to reconstruct the full database of fingerprints from measurements at a small number of points on the grid.

Kaemarungsi and Krishnamurthy developed an analytical model which considers an RSS vector from the offline phase and an RSS vector from the online phase as the inputs to the algorithm [9]. The proposed framework is formed by computing the mean received signal strength as the transmit power of the AP attenuated by the mean path loss, using the distance of the transmitter and receiver and the mean path loss model. The authors used this in combination with the two RSS vectors to form probability of correct estimation for some assumed system model setup. The influence of number of APs, standard deviation of RSS, path loss exponent, and grid spacing on accuracy and precision is investigated. The related results are then validated with simulation.

The authors of [10], have employed the difference of RSS measurements (signal strength difference or SSD), rather than their absolute values, as the location fingerprints. They also take advantage of interpolation based techniques to reproduce the database of fingerprints in changed environments based on a few sample RSS measurements. Finding good APs, referred to as anchors, for which the signals are very location dependent, is another contribution of the paper.

A variety of recent research is based on Hidden Markov Models (HMMs) for indoor localization. In [11], for example, online RSS measurements are combined with movement information, and the forward algorithm is used to estimate the set of most likely positions, all of which are then averaged to compute the user location estimation. El. Khoribi et al. [12] approach the problem by using simulated training data based on a log-distance path

loss model in addition to collected measurements. This data is used to model the motion dynamics of a mobile user as a Markovian first order process.

In another application of HMM for pedestrian navigation [13], the probability distribution for user position is updated each time new movement and RSS information is provided. In the movement update, dead reckoning (estimating the position using information on direction and distance traveled) is used for estimating the step length traveled by the user, for which acceleration measurements and compass heading, obtained by magnetic field calculation, are employed. In the measurement update, online RSS data is correlated with the stored values in the database to find the new distribution of the user location.

In this thesis, we focus on the design of an algorithm that estimates user trajectory based on a time series of RSS measurements. In the proposed solution, we focus on combining a Bayesian-based sequential estimation approach with RSS fingerprinting (and kinematic information when available) to achieve reliable trajectory estimation while maintaining low computational complexity. Given online RSS measurements, kNN is used to approximate the probability that the user is in a given location, motion constraints are used to build a location transition model, and the most likely path (sequence of locations visited) is determined via a trellis search using the Viterbi algorithm. The performance of the algorithm using real data shows the performance improvement over kNN alone. While the proposed method employs HMM like many existing approaches, it differs in that it leverages kNN to maintain moderate complexity.

To describe the proposed algorithm, consider a given indoor area in which a mobile user is to be localized, within the map of the area. This map may have a metric or a topological representation. In the metric framework, the objects are located in a two dimensional space with known and exact coordinates. The most important weakness of this map, is its sensitivity to noise which affects computation of distances between objects located in the space. The topological framework, draws a graph related to the area of interest, in which nodes represent places and edges display their appropriate relations [14]. Since there is typically uncertainty in both structures, many techniques suggest probabilistic representation. This

helps address the uncertainty which naturally appears both in robot's position and the geometric features of the system [15]. In this work, we focus on the metric framework, where we assume exact coordinates of all points on the grid are given.

Using any of the three mentioned representations, the process and work on building the map, is called map learning. A commonly used approach to map learning is simultaneous localization and mapping (SLAM). In this approach, the map of the unknown environment is built by a mobile user, while the user travels and self localizes within the area, using the same mentioned map. SLAM operates by using several different types of sensors. There are laser scans or visual features which provide information of many points within an area, and tactile sensors which provide information about points very close to the robot. In practice, SLAM tasks combine visual and tactile elements. VSLAM (visual SLAM), has been the focus of attention recently, due to increasing availability of cameras in almost any mobile device [16].

Solving the SLAM problem requires significant processing and computational power, which is often beyond the capabilities of simple, low-power robots designed for localization. Rather than solving SLAM, the focus of this work is to present a novel approach which uses the map of the indoor area, but localizes the mobile user with much less computation and processing.

While our proposed approach also requires the map of the area to perform localization, it removes the burden on the user to create the map. This would require that the map (with offline fingerprints) be provided. To make it more clear, note that while the location of the user within the area is continuous in nature, the proposed approach relies on a discrete approximation to the space. A grid of discrete points (locations) is defined to cover the area, and offline RSS observation fingerprints are collected at each of these points. These measurements serve as labeled training data that can be used in KNN to classify RSS fingerprints observed during the online phase. During the online phase, the user travels some unknown path, reporting the RSS values observed in the environment. The goal of the designed algorithm is to find the most likely path given the online measurements and the

constraints on the user's movement. More detailed explanations and equations related to the algorithm, in addition to performance evaluation for the proposed method, are discussed in Chapters 3-6.

Chapter 3: Algorithm Description

The proposed algorithm for indoor location estimation employs a Bayesian metric to approximate the likelihood of a particular path given a set of RSS fingerprint observations. Consider a given indoor area in which a mobile user is to be localized. While the location of the user within the area is continuous in nature, the proposed approach relies on a discrete approximation to the space. A grid of discrete points (locations) is defined to cover the area, and offline RSS observation fingerprints are collected at each of these points. These measurements serve as labeled training data that can be used in KNN algorithm to classify RSS fingerprints observed during the online phase. This part of the data collection, is where a considerable amount of labor work could be required, depending on the method used for creating the database of RSS measurements. The process could be performed manually, in which the hardware and related setup for data collection is located separately at each point of the grid. Labor work is required to move the hardware from one point to the next, after RSS data is collected and saved at each location for some fixed time duration (time is measured manually as well). This would continue until all location of the grid are covered. In addition to the considerable amount of time and energy required by the labor in this method, there exist some possibility for errors caused in measuring time, locating the setup consistently at all points, etc. As another, more efficient approach, automated data collection is suggested. This includes programming a robot to move through some path on the grid and wait at each location for the duration of interest, while collecting RSS data in the environment. The path is designed so that the robot stops at all points of the grid exactly once. This method saves for labor work considerably, specifically if we are dealing with a large grid. However, some work will be required to make sure the process is performed with enough accuracy. The process used for data collection in this work will be addressed and discussed in detail in chapter 5.

Let L_1,...,L_N denote the locations of the mobile user at time indices 1 through N and r_1,...,r_N denote the RSS fingerprint observations at time indices 1 through N. To estimate the path of the user, we find the sequence of locations L_1,...,L_N that maximizes the conditional probability $P(L_1, ..., L_N|r_1, ..., r_N)$. The nature of the tracking and localization problem allows us to simplify the conditional probability into an expression that can be updated sequentially as new online RSS fingerprint observations are obtained. Using Bayes theorem, we have

$$P(L_1, ..., L_N|r_1, ..., r_N) = \frac{P(r_1, ..., r_N|L_1, ..., L_N)P(L_1, ..., L_N)}{P(r_1, ..., r_N)} \tag{3.1}$$

$$\propto P(r_1, ..., r_N|L_1, ..., L_N)P(L_1, ..., L_N), \tag{3.2}$$

noting that $P(r_1, ..., r_N)$ is equal for all possible paths. Since we can reasonably assume that the RSS fingerprint observation at time i depends only on the location at time i (and not on the location at other times), we can write the conditional probability as

$$P(r_1, ..., r_N|L_1, ..., L_N)P(L_1, ..., L_N) = \prod_{i=1}^{N} P(r_i|L_i)P(L_1, ..., L_N) \tag{3.3}$$

$$= \prod_{i=1}^{N} P(r_i|L_i) \prod_{j=2}^{N} P(L_j|L_{j-1})P(L_1), \tag{3.4}$$

where the second equality follows from the fact that, conditioned on the location at time *i-1*, the location at time i is independent of all locations before time *i-1*.

Assuming all starting locations are equally likely, $P(L_1)$ will be equal for all possible

paths. Hence, we obtain the following form for the likelihood metric:

$$P(L_1, ..., L_N | r_1, ..., r_N) \propto \prod_{i=1}^{N} P(r_i|L_i) \prod_{j=2}^{N} P(L_j|L_{j-1}). \tag{3.5}$$

According to (3.5), computing the likelihood metric requires computing the conditional likelihood of the RSS observations, $P(r_i|L_i)$, and the location transition likelihood, $P(L_i|L_{i-1})$. In general, the transition likelihood function models any known behaviors or kinematic constraints of the mobile user. In our work, we assume that the user moves to a neighboring location in each time step, with each neighbor being equally likely. As a result, $P(L_i|L_{i-1})$ is nonzero only when L_i is a neighbor of L_{i-1}. The following sections describe our approach to computing the conditional observation likelihood.

To compute the conditional likelihood of the RSS observations, $P(r_i|L_i)$, two approaches were investigated. In the first approach, a mathematical model for path loss was used to predict observed RSS at a given location; this approach was evaluated via simulation. To address more practical scenarios, a kNN-based approach that relies on labeled training data is considered. This approach was evaluated using experimental data. Details of the two approaches are provided in Sections 3.1 and 3.2. Use of the Viterbi algorithm to compute the overall path likelihood is described in Section 3.3.

3.1 Path-loss Based Modeling of Conditional Likelihood of RSS

In using a mathematical model to describe the expected RSS observations, we assume knowledge of both the location grid points and the access points within the indoor space of interest. To determine the signal strength from AP j received at point k of the field, the mean path loss model is employed as [9]

$$Pl(d_{j,k}) = Pl(d_0) + 10\alpha \log_{10}(d_{j,k}) \tag{3.6}$$

where $Pl(d_{j,k})$ is the mean path loss, $d_{j,k}$ is the physical distance of the k -th grid from the j -th AP in meters, and $Pl(d_0)$ is defined as the free-space path loss at the known reference distance given by $d_0 = 1m$. The variable α denotes the path-loss exponent, which varies depending on the building type, corresponding to each indoor area. The mean RSS can then be computed as

$$E\{RSS_{j,k}\} = P_t - Pl(d_{j,k}) \tag{3.7}$$

where P_t is the transmit power of the AP of interest. The simulated value for $RSS_{j,k}$ is then computed by adding Gaussian noise with mean zero and variance σ^2. The numerical values we used for our simulations are discussed in Chapter 4 with the simulation results.

To compute $P(r_i|L_i)$, we consider the RSS distribution for each AP separately. The signal strength observed at each point of the grid, from a specific AP, is a Gaussian distributed random variable with mean given by equation (3.7) and variance σ^2. If the user location L_i is known, the RSS distribution at that location, can be determined for each AP. $P(r_i|L_i)$ for one AP would then be the value of this distribution at the reported observation vector $(\mathbf{r_i})$.

After computing these probabilities for each AP separately, $P(\mathbf{r_i}|L_i)$ can be written as

$$P(\mathbf{r_i}|L_i) = P(r_{i,1}, r_{i,2}, ..., r_{i,n}|L_i) \tag{3.8}$$

where $r_{i,1}$, ..., $r_{i,n}$ represent observations corresponding to AP1, ..., APn. Assuming these are statistically independent random variables, (3.8) can be simplified as

$$P(\mathbf{r_i}|L_i) = P(r_{i,1}|L_i)P(r_{i,2}|L_i)...P(r_{i,n}|L_i), \tag{3.9}$$

and the conditional likelihood of the RSS observations is then computed using the Gaussian

distribution previously described.

3.2 KNN-based Estimation of Conditional Likelihood of RSS

Because the physical environment is changing over time and is often too complex to be accurately modeled, and since the locations of the APs may be unknown, an analytical model of the RSS observations may not provide reliable information in this context. As an alternative to analytical models, we can employ an offline training phase in the algorithm to provide information about the RSS fingerprints we expect to see at various locations within the building. When an offline data collection phase is included, KNN operating on the training data collected in the offline phase can be used to compute the conditional observation likelihood, which is given by

$$P(r_i|L_i) = \frac{P(L_i|r_i)P(r_i)}{P(L_i)} \propto P(L_i|r_i), \tag{3.10}$$

assuming all locations are equally likely at time i when not conditioned on RSS observations. In KNN, a sphere is defined that includes the k training points closest (smallest Euclidean distance) to the observed RSS fingerprint r_i. Using KNN, $P(L_i|r_i)$ can be approximated by the fraction of training samples that bear the label L_i in a sphere around r_i. Suppose m of the k points in the sphere around r_i bear the label L_i; the conditional probability is then approximated as $P(L_i|r_i) \approx \frac{m}{k}$. As an example, consider Figure 3.1, in which x_u, needs to be classified. For $k = 5$ in KNN, 4 closest neighbors (among 5), belong to class 1, and 1 neighbor belongs to class 3. Accordingly, the probability for x_u to belong to class 1 equals $\frac{4}{5}$, and the probability that it belongs to class 3 is $\frac{1}{5}$. Obviously, the point belongs to class 2 with probability zero.

Note that if none of the k training samples within the sphere around a fingerprint observed at time index i bear the correct location label, all trajectories that include the correct location at that time index will be eliminated from consideration. To overcome this

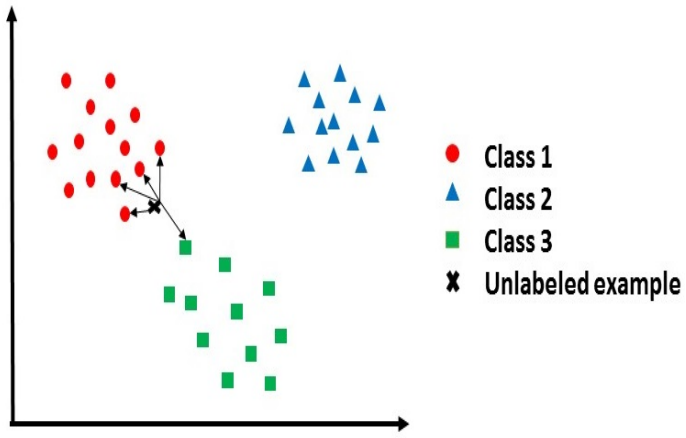


Figure 3.1: An example of KNN classification with three classes

issue, a small bias term is added to each posterior probability, $P(L_i|r_i)$, and the results are normalized to maintain a total probability of 1.

3.3 User Path Estimation Using Viterbi Algorithm

Having developed approaches to compute the observation and transition likelihoods, $P(L_i|r_i)$ and $P(L_j|L_{j-1})$, we now consider how to compute the overall path (trajectory) likelihood conditioned on RSS observations, $P(L_1, ..., L_N|r_1, ..., r_N)$. As described in [17], a trellis structure can be used to describe possible transitions from one location (state) to the next. As depicted in Figure 3.1, the nodes correspond to states at each time stamp, and the arrows represent transitions from one state to the next. Clearly, there are totally four states, and two possible transitions from one state to the next, at each time stage.

Given the Markov structure of the conditional likelihood, the Viterbi algorithm can be used to determine the most likely path in a computationally efficient manner. The complexity of the Viterbi algorithm is $O(T.K^2)$, where T is the sequence length and K is the

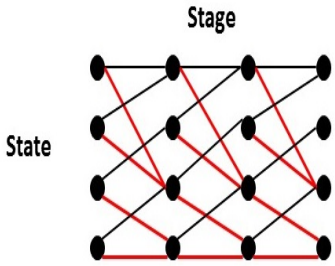


Figure 3.2: Example of a trellis that could be used in the Viterbi algorithm

total number of states. Clearly, complexity grows linearly with path length, dramatically improving computational requirements relative to a brute-force comparison approach. Information about constraints on user motion reduces the number of transitions exiting each state and makes the search much less complex. Providing further motion constraints, results in a good estimate of the distance and direction traveled by the user. This eliminates many impossible transitions of the trellis and reduces complexity even further.

As an alternative to the Viterbi algorithm for maximum likelihood sequence detection, the stack algorithm is suggested in [18]. Rather than searching a trellis, an iterative search through a tree is performed to find the most likely path or state sequence. Starting from the initial state, the algorithm builds and extends the best path by choosing the one with the largest metric at each iteration. Tree-search based approaches have been shown to provide reduced complexity relative to trellis-search approaches for a variety of applications. Development of such a metric is beyond the scope of this work and is considered as future research.

Chapter 4: Simulated Performance Results

In this chapter, the performance of the proposed sequential algorithm, is evaluated using simulations that model path loss based RSS measurements. Offline and online data are generated using a linear path loss model, and the Viterbi algorithm is applied to find the best estimate for the path traveled by the user. The influence of white Gaussian noise on the performance of the algorithm is investigated by empirically computing the probability of correct path estimation for different noise levels.

4.1 Path loss Based RSS Estimation

A set of grid of points and a set of AP locations, all with known coordinates, are defined. The grid points are located as shown in Figure 4.1. The location of the APs are chosen randomly within the square grid of possible locations. We assume that the user reports RSS observations for five time indices (sequence length = 5) and that three APs are active in the environment.

We use the linear path loss model described by (3.6) and the mean RSS calculation described by (3.7) to compute the expected RSS at each point from each AP. The parameter values $Pl(d_0)$, P_t , and α are chosen, consistent with [9], to be equal to 41.5 dBm, 15 dBm and 3.3, respectively.

Offline data is then simulated by adding white Gaussian noise with mean zero and variance σ^2 to the vector of RSS values (one element for each AP) obtained using (3.7) for each point of the grid. Online, or test data, is also generated using the same mean RSS vector and different realizations of the additive white Gaussian noise at five points of the grid that form our path of interest for the given simulation.

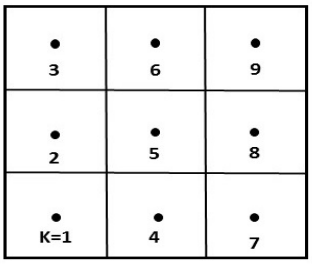


Figure 4.1: Map of locations for the simulated environment. Points of the grid are labeled with k=1, ..., 9. Three APs are randomly located inside the square grid and used for simulated RSS vector generation.

4.2 Sequential Localization Using Trellis Search for Simulated RSS Measurements

To provide an example of the trellis constructed for a length-5 sequential search, consider the true path $\{2, 5, 6, 9, 8\}$, meaning that the user has started from location 2 and traveled the path of locations 5, 6, 9 and finally stopped at 8. It is assumed that the the travel consists of five consecutive time stamps and that the user has reported an observed RSS vector at each location (and each time index). Assume motion information is available from the user and is provided to the algorithm as the possibility for the user to move to its neighboring points on the grid at each time index. Given this information, a set of possible starting points and movements can be defined. Accordingly, a trellis can be used to describe all possible starting states and all possible transitions from one state to the next, as depicted in Figure 4.2. The gray lines represent all possible transitions from one state to another, and the red lines show the most likely path, given the RSS observation at each stage. Clearly, nine different possibilities should be investigated to find the best

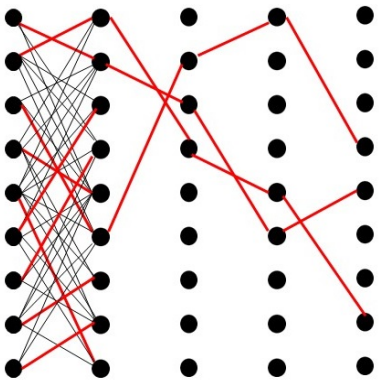


Figure 4.2: Trellis for the given example

path estimate. To do so, the likelihood of all possible paths should be computed using the metric in (3.5), where, $P(r_i|L_i)$ is computed assuming a Gaussian distribution for RSS at each location, in addition to the assumption for the APs to be statistically independent, as described in Section 3.1. For $P(L_j|L_{j-1})$, we note that this is equally likely for neighbors of each location, so depending on the number of neighbors, this probability can be computed and used in finding the likelihood for each path, in the trellis.

4.3 Performance results

To evaluate the performance of the proposed sequential technique given by the metric in (3.4), a set of simulations are developed in Matlab. As shown in Figure 4.1, 9 points on a square grid are considered as the area of interest. Five different paths with length 5 are considered. These include $\{2, 5, 6, 9, 8\}$, $\{1, 2, 3, 6, 5\}$, $\{3, 6, 9, 8, 7\}$, $\{1, 2, 5, 8, 9\}$, and $\{3, 2, 1, 4, 7\}$. For each path, in 1000 different simulations, the offline and online data are generated separately (for a fixed σ^2), and the shortest path is estimated using trellis search (through the trellis in Figure 4.2.) via the Viterbi algorithm. The total number of times that the sequential technique is able to correctly estimate the user's path (all locations correctly

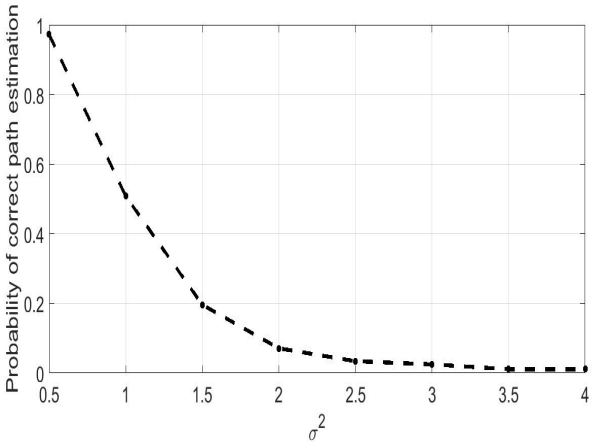


Figure 4.3: Simulated performance results

estimated) is divided by 1000, to find the empirical probability of correct path estimation (PCPE) for each path. The results are then averaged over all five paths mentioned. These simulations are repeated for different values of σ^2, ranging from 0.5 to 4, to obtain the plot depicted in Figure 4.3. As the performance plot shows, the highest PCPE equals 0.972, and the lowest is 0.011. Also, the plot decays monotonically with σ^2.

To explain why a lower value of σ^2 leads to a higher probability of correct estimation, note that for $\sigma^2 = 0$, the offline and online RSS are exactly the same, and equal to the value computed by (3.7). As a result, the conditional probability $P(r_i|L_i)$, when r_i is reported at the true location L_i, is equal to one for all five online measurements reported. For other values of σ^2, a smaller variance, means the noisy measurements are more tightly distributed around the mean, so the conditional likelihoods take on higher values at the correct locations, explaining the higher PCPE, at lower σ^2.

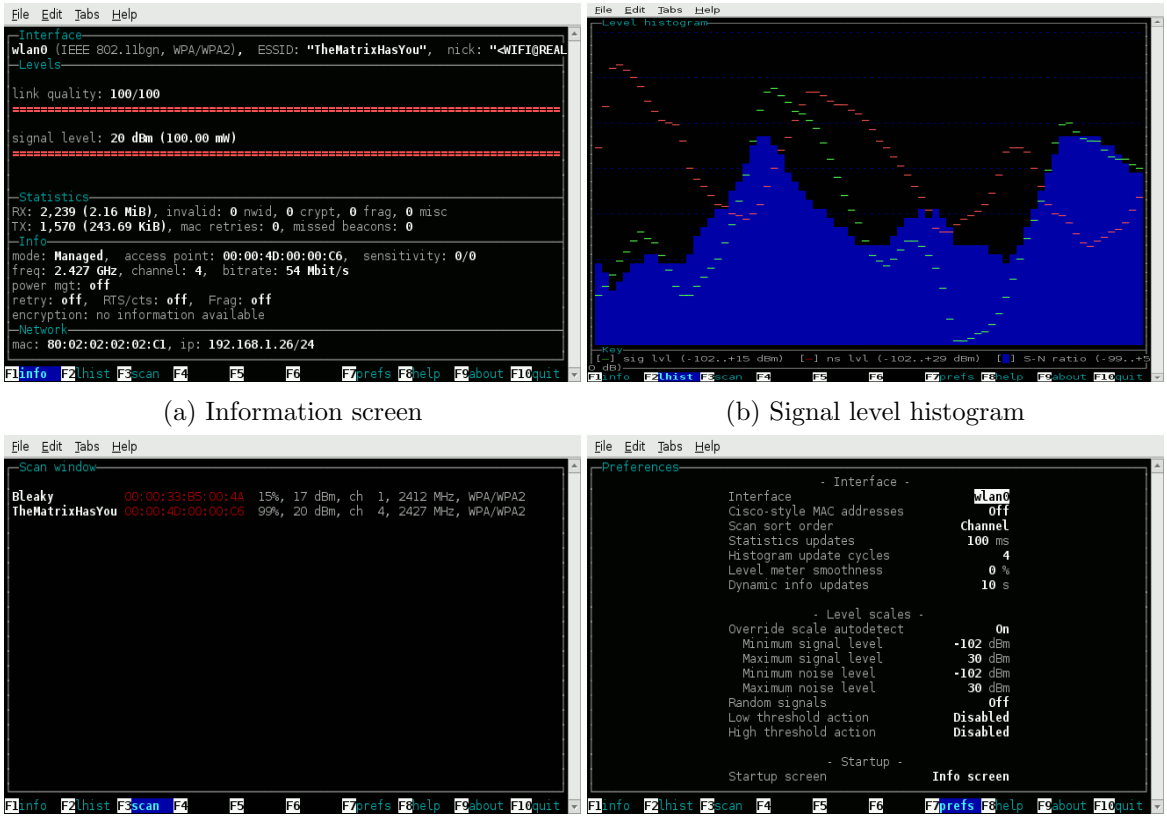
Chapter 5: Experimental Results

In this chapter, we focus on KNN-based estimation of the conditional likelihood of RSS observation, as described in Chapter 3. The performance of the proposed sequential technique is evaluated by collecting real RSS measurements in the Nguyen Engineering Building at George Mason University. This provides the training and test sets required by the KNN algorithm. After describing hardware and software used for collecting data, the layout for two location scenarios, a hallway and a study lounge, both on the third floor of the Nguyen Engineering Building, is explained. Details regarding measurements collected at each set of locations, in addition to performance results, are included.

5.1 Hardware and Software Used for Data Collection

To collect RSS measurements in both the offline and online phases, an external USB antenna was used. The chosen antenna was an Alfa Network AWUS051NH v2, an IEEE802.11 a/b/g/n dual-band 2.4 GHz /5 GHz wireless USB antenna. The antenna, was connected to a laptop running Ubuntu 14.04 OS.

The Wavemon utility was used to save the RSS data collected by the antenna. Wavemon is typically used to monitor wireless networks in the environment. Information regarding signal and noise levels, packet statistics, device configuration, and network parameters of the wireless networks is provided by the utility. More specifically, this information can include a periodic update of the ESSID (Extended Service Set Identification), MAC (Media Access Control) address, and signal level in dBm, for the available APs [19]. As depicted in Figure 5.1(a), the Wavemon Info screen shows an overview of network parameters and statistics. Figure 5.1(b) shows the signal level histogram, which describes variations in signal levels



(a) Information screen

(b) Signal level histogram

(c) Scan window

(d) Preferences

Figure 5.1: Screenshots of the wavemon screens

over time. As the key at the bottom of the figure describes, the green plot shows signal levels in time, the red one corresponds to noise levels, and the blue graph is the signal-to-noise ratio (SNR). An example of the scan window can be seen in Figure 5.1(c). ESSID, time stamp, MAC address, channel, and frequency are provided in the scan. Finally, the preferences window is shown in Figure 5.1(d). Options such as scan rate, statistics update, etc., are configurable. In the data sets collected in this work, we set these parameters to 4000 ms for statistics updates and 2 for the dynamic info updates.

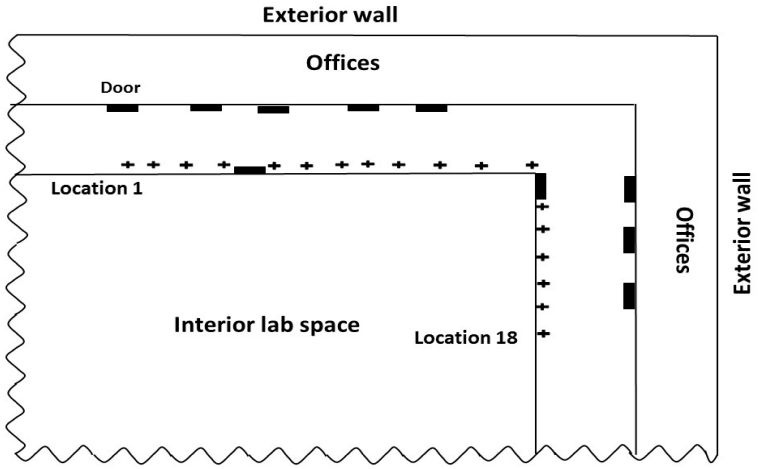


Figure 5.2: Map of the experiment environment in the ECE department

5.2 Experiment Scenario I: Hallway

For localization evaluation, we considered two different scenarios. The layout for the first scenario is depicted in Figure 5.2. The area of interest is located on the third floor of the engineering building. Eighteen locations were chosen along a hallway. Adjacent locations were separated by one meter in most cases, separation distances were sometimes larger to accommodate office doors. Two sets of data collection were performed for this scenario. We refer to them as the April and August data sets, denoting the month in which each was collected. Analysis of both sets is discussed in Performance Results, Section 5.2.3.

5.2.1 Basic Data Collection Information

As described above, training data for the KNN algorithm was collected during an offline phase using Wavemon, an external USB antenna, and a laptop running Linux OS. The area of interest was marked with tape (either on the carpet floor or on the bottom of the wall) at the desired locations for consistent reference. The laptop and antenna (both hardware

position and orientation) were always located at a fixed and constant spot in relation to the tape marks. For this scenario, the laptop was placed on a cart and easily moved from one location to the next. The duration of the scan process was 1 minute for each point of the grid. During this time, there were unpredictable interruptions in Wavemon. These interruptions were shown as "searching for data" on the Wavemon screen and resulted in jumps in the timestamps stored in the file of collected data. This issue is discussed further in Chapter 6.

Using a manual timer, the one-minute scan time was measured, and Wavemon was stopped. A text file of the data collected was automatically generated and saved onto the laptop. The setup was defined so that the name of the file included date and start time of the scan. As an example, `"WaveMon_Log_2016-08-10_1428.log"` is the data file collected on August 10, 2016 at 14:28. This includes 7 different columns of timestamp, channel, frequency, MAC address, Mode, ESSID, and signal strength. The data points are then built out of this file for each location.

We considered APs (MAC addresses) that were part of the secure campus wireless network, MasonSecure, since these APs are expected to provide a more reliable set of RSS data with fewer variations over time. While there were many visible MasonSecure APs, not all provided useful information. Accordingly, we explored options to further reduce the dimensionality of the RSS fingerprints by performing an AP selection procedure, which also eliminates redundancy in signal information. In our simulations, for unseen or unavailable access points, we assumed signal strength equal to -100 dBm. For localization purposes, we considered only APs whose average signal strength values were greater than -45 dBm. After investigating all RSS measurements, four APs were found which met the required condition. Using these APs resulted in strong performance without incurring unneeded computation.

5.2.2 Performance Results

For the April dataset, the data collection was performed on three consecutive days: Sunday, April 10 (starting at 4:16 p.m.), Monday, April 11 (starting at 12:54 p.m.), and Tuesday,

April 12 (starting at 8:11 p.m.), all in 2016. These three collection days and times were chosen to represent a variety of levels of activity (and hence noise scenarios) in the building. The offline data was collected by scanning each location for one minute.

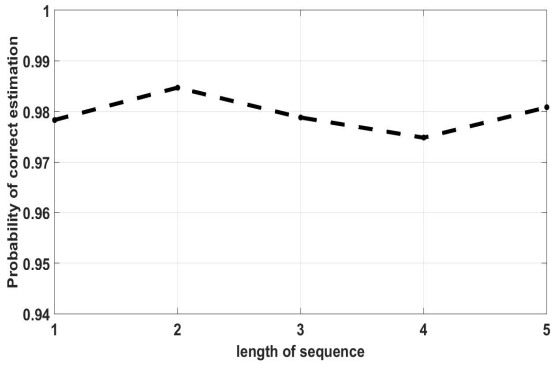
To apply the sequential metric, 1000 training points (fingerprints) were selected randomly from the offline data, one third from each collection day. The test data was drawn randomly from the second collection day (Monday); 10 fingerprints (online measurements) were drawn for each location visited by the user. The KNN algorithm was applied to each fingerprint (test point) separately, and the results were averaged over 10 points to obtain the final KNN estimation. The value of k for KNN was set to 15.

The performance of the sequential approach, presented in terms of the probability of correctly estimating all locations in a path as a function of path length, is shown in Figure 5.3(a). Results are averaged over 100 realizations of online fingerprint measurements for each possible path traveled. The percentage of paths estimated correctly is larger than 97% across all path lengths considered. Paths only up to length 5 are considered, as it is assumed that a relatively small window for hard decisions will be applied in the Viterbi algorithm.

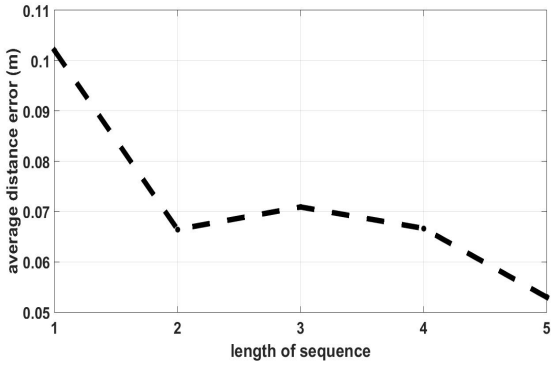
Figure 5.3(b) shows the average distance error between the locations in the true and estimated paths, again as a function of path length. Distance error for each path, normalized by path length, is computed as

$$\frac{1}{N}\sum_{i=1}^{N} d(L_i, \hat{L}_i), \tag{5.1}$$

where N denotes path length, d denotes Euclidean distance, L_i is the true location at time i, and $\hat{L}_i$ is the estimated location at time i. Distance error is then averaged over 100 realizations of online fingerprint observations for each possible path. As seen in Figure 5.3(a), average distance error shows a general trend of decreasing as path length increases, indicating the value of a time series of RSS fingerprints in reducing the performance-degrading



(a) Probability of correct path estimation as a function of path length



(b) Average distance error as a function of path length

Figure 5.3: Performance results for data collected in April along third floor hallway

effects of time variation in RSS. This is also confirmed by comparing the performance results for path lengths of 1 and 2 in Figures 5.3(a) and 5.3(b). The improvement in sequence estimation gained by increasing the path length from 1 to 2 is clearly observable. Note, however, that average distance error increases when we move from a path length of 2 to a path length of 3. As an element for future study, the paths that were estimated incorrectly should be separated and explored in more detail. Other combinations of training and test data could also be used to further explore the results.

A second set of data at the same locations was collected on August 10, 2016. Abdulwahaab Arif performed the August data collection using the Pioneer 3DX, the most popular

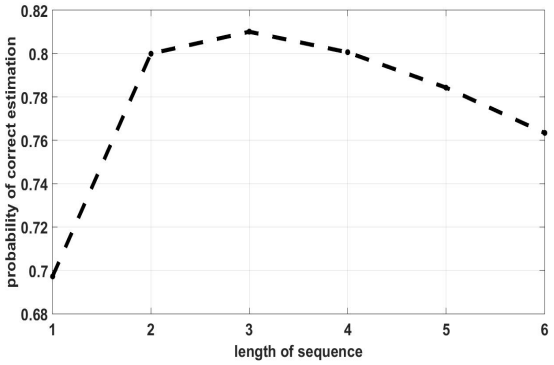
research mobile robot used for localization/mapping purposes [20]. A laptop running Linux OS was placed on a Pioneer 3DX, shown in Figure 5.4, and was used for collecting data along the third floor hallway in five different training sets, all collected in the same afternoon. Five additional locations were added to the end of the path, shown in Figure 5.2, continuing in a line from location 18. Each set included 23 log files of scans representing the Wi-Fi data recorded for a duration of one minute at each of the 23 locations. The starting time stamps of data collection for the five sets were Set 1 – 1:55 pm, Set 2 – 2:28 pm, Set 3 – 3:01 pm, Set 4 – 6:04 pm, and Set 5 – 6:35 pm.

Figure 5.4: A picture of the Pioneer 3DX used for data collection [21].

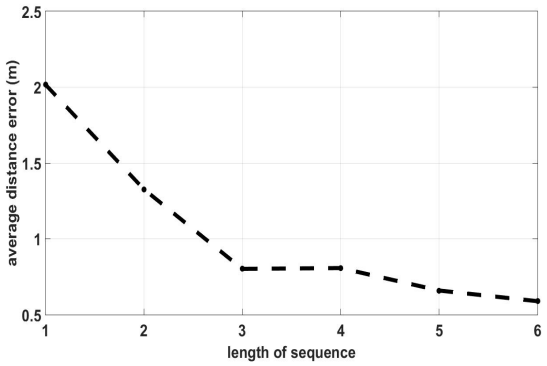
Most sets are separated by a gap of approximately 30 minutes. The longer gap between sets 3 and 4 resulted from a need to charge the laptop battery before continuing.

The proposed sequential localization approach can be applied to each of the five data sets separately or to their combination. The performance results for pulling both training (offline) and test (online) points from set 1 is depicted in Figure 5.5. The number of realizations for each possible path is 30, and 100 training points are used for each location. 10 online data points are used as the test data for each location visited by the user. The probability of correct path estimation is above 0.68 for all sequence lengths, and there is a dramatic improvement of 0.1 when moving from $N = 1$ (path length) to $N = 2$. The

probability of correct path estimation reaches its maximum value (approximately 0.81), for length $N = 3$, and then degrades slightly when moving toward $N = 6$. The average distance error, however, decreases almost monotonically, with a 1.4 meter reduction from $N = 1$ to $N = 6$, showing the power of a sequence of measurements compared to RSS values at a single location.



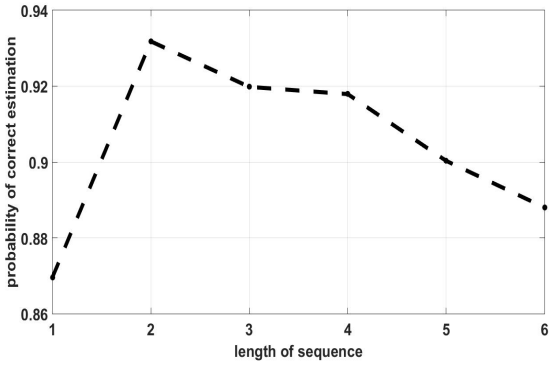
(a) Probability of correct path estimation as a function of path length



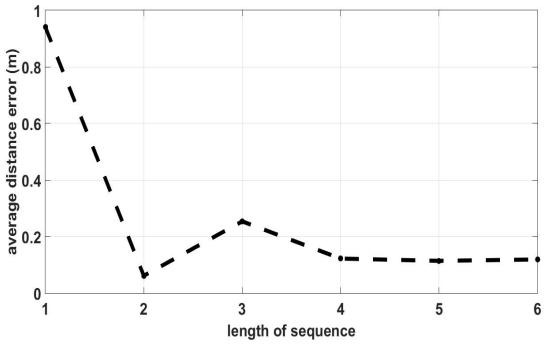
(b) Average distance error as a function of path length

Figure 5.5: Performance results for data collected in August along the third floor hallway, Set 1

To further evaluate performance, the training and test data points were drawn randomly from each of Sets 2, 3, 4, and 5. The simulation parameters (including number of realizations, training and test points) are the same as Set 1. The results for each data Set are shown in Figures 5.6 through 5.9.



(a) Probability of correct path estimation as a function of path length



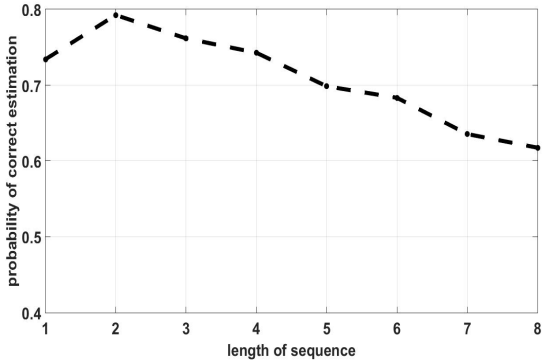
(b) Average distance error as a function of path length

Figure 5.6: Performance results for data collected in August along the third floor hallway, Set 2

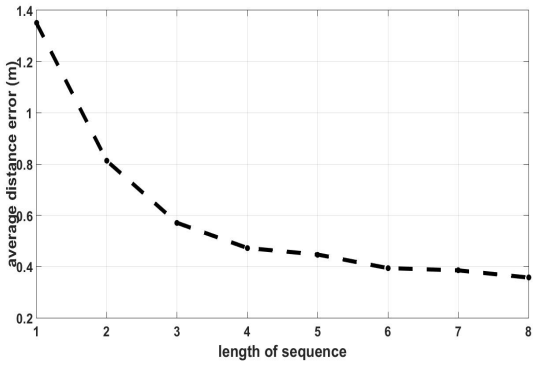
As Figure 5.6 (corresponding to Set 2) shows, the probability of correct estimation is roughly between 0.87 to 0.93, which is greater than for Set 1; the probability of correct path estimation rises and falls somewhat as sequence length increases. As observed for Set 1, the greatest performance improvement occurs in both performance plots when adding the second measurement to the first one. The variations as N increases from 2 to 6 are much smaller.

For Set 3, the fraction of paths estimated correctly decreases with increasing path length, as Figure 5.8 shows. However, the average distance error decreases monotonically with

increasing path length. This indicates that although more paths are being estimated incorrectly, the true and estimated sequences have closer locations (in terms of Euclidean distance) to one another for longer paths.

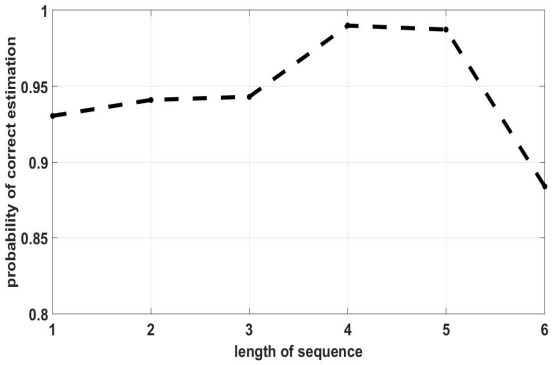


(a) Probability of correct path estimation as a function of path length

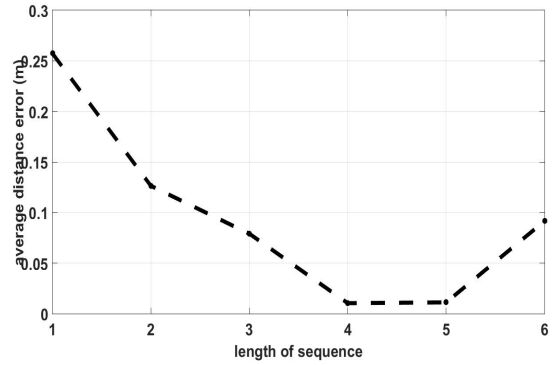


(b) Average distance error as a function of path length

Figure 5.7: Performance results for data collected in August along the third floor hallway, Set 3



(a) Probability of correct path estimation as a function of path length

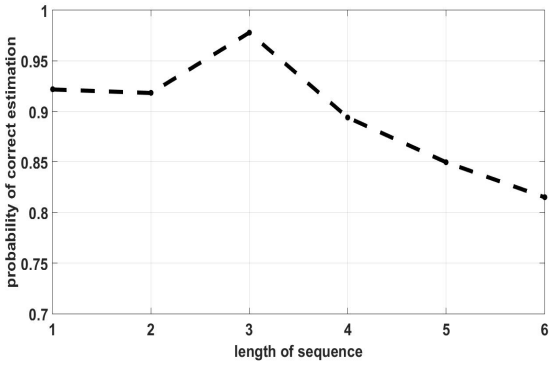


(b) Average distance error as a function of path length

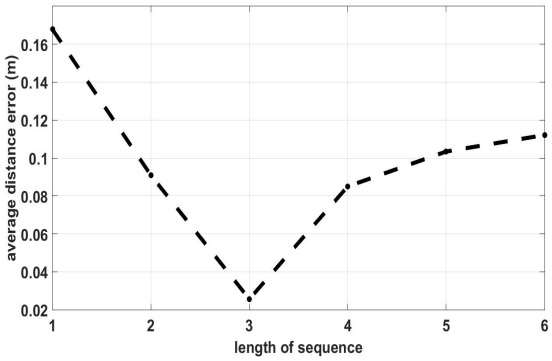
Figure 5.8: Performance results for data collected in August along the third floor hallway, Set 4

Sets 4 and 5 display similar behaviors. For both, the first performance plot (Figures 5.8 (a) and 5.9 (a)), shows worse performance for length 6 than for length 1. However, looking at performance in terms of average distance error, (Figures 5.8 (b) and 5.9 (b)), the error rate has decreased to a noticeable extent, emphasizing the importance of the second metric (distance error) in deciding the optimum sequence length.

Finally, as a comprehensive evaluation of the algorithm, we combined all 5 sets together. We pulled 50 training points for each class as the offline data and 10 test points as the online data. 20 realizations were made for each path traveled by the user. As the plots in Figures 5.10(a) and 5.10(b) show, the probability of correct path estimation is greater than 0.86 in



(a) Probability of correct path estimation as a function of path length

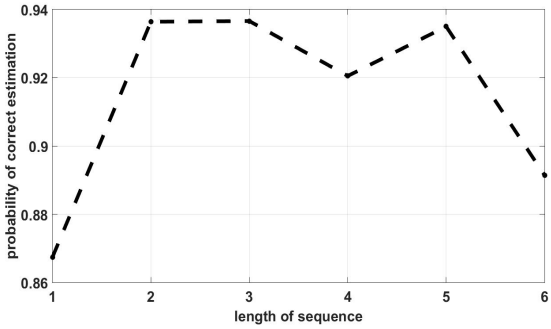


(b) Average distance error as a function of path length

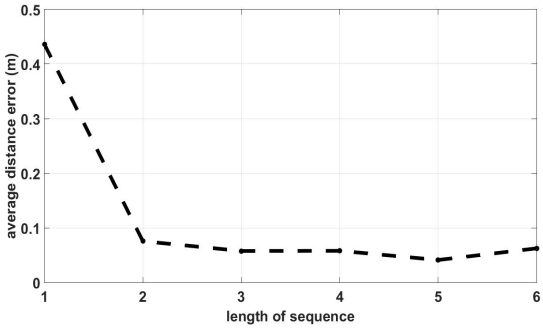
Figure 5.9: Performance results for data collected in August along the third floor hallway, Set 5

all cases, and average distance error decreases with a large slope from length 1 to length 2 and stays less than 0.1 meters for lengths 2 to 6. Note that Figure 5.10(b), shows that average distance error is monotonically decreasing with increasing path length. This could be a result of combining all sets, and incorporating variations in RSS data across one day. The sequential metric also performs much better with the combination of 5 sets. More investigation on the reasons for this improvement over individual sets is a topic for future work.

To evaluate robustness over time, we analyzed the performance of the proposed localization technique when the training and test samples came from different datasets. To perform



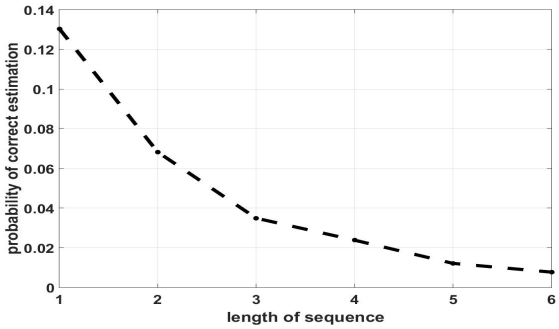
(a) Probability of correct path estimation as a function of path length



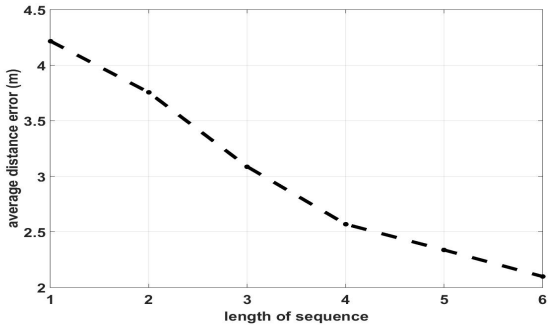
(b) Average distance error as a function of path length

Figure 5.10: Performance results for data collected in August along the third floor hallway, Combination of 5 sets

the evaluation, we pulled 50 training points out of Set 1 and 10 test points out of Set 3 for each location. The performance results for this combination are shown in Figure 5.11. Performance has degraded dramatically relative to the cases in which test and training were drawn from the same set(s). The probability of correct path estimation is less that 0.14 for all path lengths, and the average error is between 2 and 4.5 meters. These results indicate that there is a considerable difference between the times at which Set 1 and Set 3 were collected. Further investigation of this issue is warranted in extensions to this work.



(a) Probability of correct path estimation as a function of path length



(b) Average distance error as a function of path length

Figure 5.11: Performance results for data collected in August along the third floor hallway, separating training and test sets

5.3 Experiment Scenario II: Study Lounge

To further evaluate the performance of the proposed algorithm, we considered another set of locations, defining a more challenging experimental condition. This time a study lounge on the third floor of the Nguyen Engineering Building at George Mason University, as depicted in Figure 5.12, was selected for data collection. As the figure shows, this area contains a roughly square grid of locations, providing more choices for the user movement to neighboring points. Searching through the wider range of possibilities, requires more time and computation using the proposed algorithm. This creates a more difficult and more realistic localization problem.

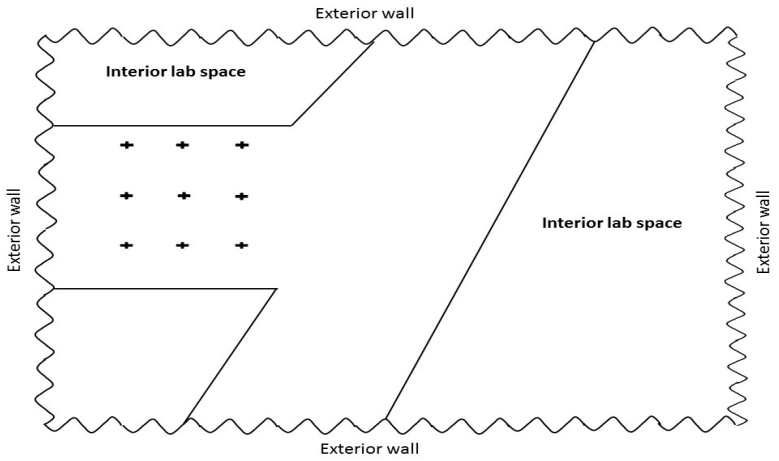


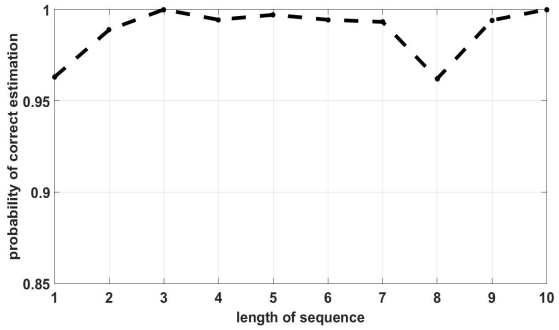
Figure 5.12: Map of the experiment environment in the Bioengineering Department study area

As in the first experimental scenario, only the APs with average signal strength values of greater than -45 dBm were chosen for localization purposes. For this scenario, three APs met this selection criterion.

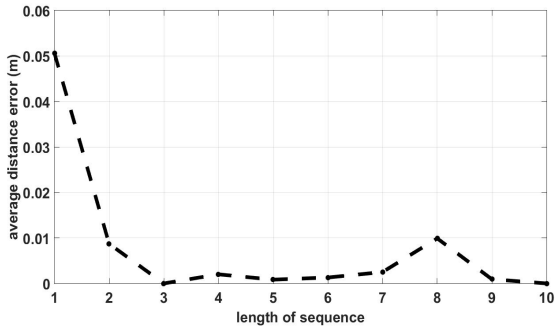
5.3.1 Performance Results

Two sets of data collection were performed in the study lounge: Monday, April 11, 2016 (starting at 5:11 p.m.) and Tuesday, April 12, 2016 (starting at 9:40 p.m.), representing a busy and a quiet scenario, respectively. The proposed algorithm was then applied to a set of training points, including 25 points from each day, resulting in 50 total points for each location. 10 test points were chosen from the April 12 data set for each realization. For each path chosen for localization, 30 realizations were performed. Ultimately, for each path length considered, the results were averaged over all (30) realizations, in addition to the length of the sequence. The related performance plots are shown in Figures 5.13(a) and 5.13(b). As seen in the figures, the proposed sequential technique can estimate the user path correctly in more than 95% of the realizations. In general, the average distance error decreases with path length. The exception is for paths of length $N = 8$. More research

and analysis is suggested to understand why there is a decrease in the probability of correct estimation and a sudden increase in the average distance error, when $N = 8$. The plots in these figures also indicate that the performance is very close to ideal for the sequence lengths 3, 5, and 10, where almost all paths are estimated correctly.



(a) Probability of correct path estimation as a function of path length



(b) Average distance error as a function of path length

Figure 5.13: Performance results for data collected in April in the Bioengineering Department study area

Chapter 6: Data Analysis and Challenges

In this chapter, we study the five data sets collected on a single day in August 2016, to show the variations of RSS values over time. A strange periodic behavior in the RSS plots is observed and addressed as well. The analysis in this chapter provides some explanation for why fingerprint-based localization using this data is challenging. We chose the five mentioned sets, since these provide a broader representation of the observations. In this analysis, we consider all observed APs (in the secure campus network) and plot RSS values versus time, at a few specific locations. Also, variations of RSS measurements over a longer time are investigated for a fixed location, fixed AP, and different training sets. At the end of the chapter, we briefly mention the challenges we faced in working with all these data sets.

6.1 Data Analysis

Considering all locations contained in the five data sets collected on August 10 2016, from 1:55 p.m to 6:35 p.m, 14 Mason Secure APs were observed. These were named as AP 1 to AP 14 for consistent reference. To better understand the behavior of the observed RSS at these APs, RSS measurements (dBm) are plotted versus time stamp at a fixed location for training sets 1 through 5. All Mason Secure APs seen in training set 3 at locations 1, 10, and 23 are plotted in Figures 6.1, 6.2, and 6.3, respectively. Clearly, not all APs among the 14 mentioned are seen at all locations. When present (active), each AP generates a roughly constant (variation of no more than 1 dBm) signal strength during the full measurement window. Looking at Figure 6.1, which shows data from location 1, AP 1 is the most active in terms of duration of presence (active duration), and AP 6 is the least. APs 2, 7 and 9 have the same active duration, but their pattern of presence is not exactly the same. AP

2 is active only for the first 25 seconds, which includes 3 cycles (each cycle is defined as a roughly 4 seconds of presence, in which the AP is active). APs 7 and 9 are also active for 3 cycles, but with a longer gap between cycles 2 and 3.

Some APs, for example AP 6 at location 1, and APs 3, 5, and 6 at location 10, are active only once, and for less than five seconds, during the full scanning time. Comparing with locations 1 and 10, fewer Mason Secure access points are observed at location 23, according to Figure 6.3. Two totally new (not observed at location 1 or 10) Mason Secure access points, including AP 8 and AP 11, become active in training set 3 at location 23. An explanation to these periods of time during which we do not see APs, can be mentioned here. By looking at log files of data, there are gaps (or jumps) in timestamps, implying that Wavemon did not collect the measurements continuously during the 1 minute period. In fact, in addition to inactivity of the AP, Wavemon could have been inactive throughout the process as well. At this point, by looking at Figures 6.1 to 6.3, in addition to the same RSS plots for the rest of locations, we can conclude that which APs are visible is a function of location.

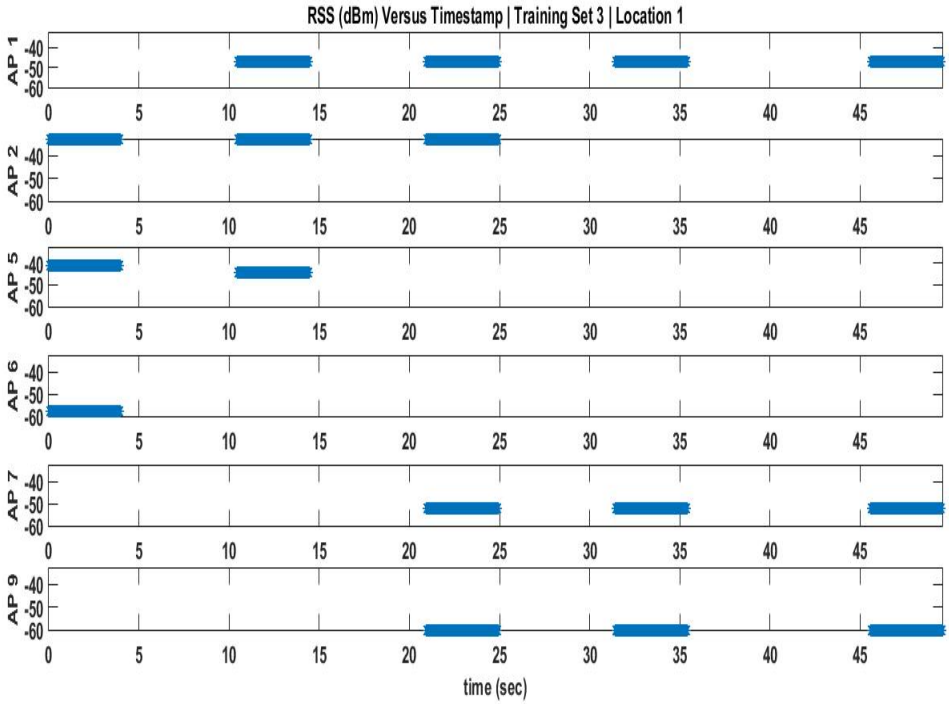


Figure 6.1: Observed RSS as a function of time at location 1 for training set 3

In another analysis, variations of RSS measurements over a longer time are investigated for a fixed location, fixed AP, and different training sets. Figure 6.4 shows plots of RSS versus timestamp, for AP 1 (MAC address: 08:CC:68:91:10:35) at location 1, for training sets 1 through 5. Note that both signal strength and signal presence vary significantly across the five training sets. For training set 5, the most activity is recorded. Two possible reasons for this are as follows : First, AP 1 might have been less active when collecting measurements for sets 1 through 4, comparing to set 5. Secondly, there is the issue with gaps (or jumps) in timestamps as mentioned before, implying that Wavemon could be interrupted in collecting the data.

AP 1 has a very different pattern of presence at location 10 compared to locations 1 and 3, as shown in Figure 6.5. Very few observations from AP 1 are collected for training set 5 (only 1-2 seconds at the beginning). The signal strength is reduced comparing to location 1

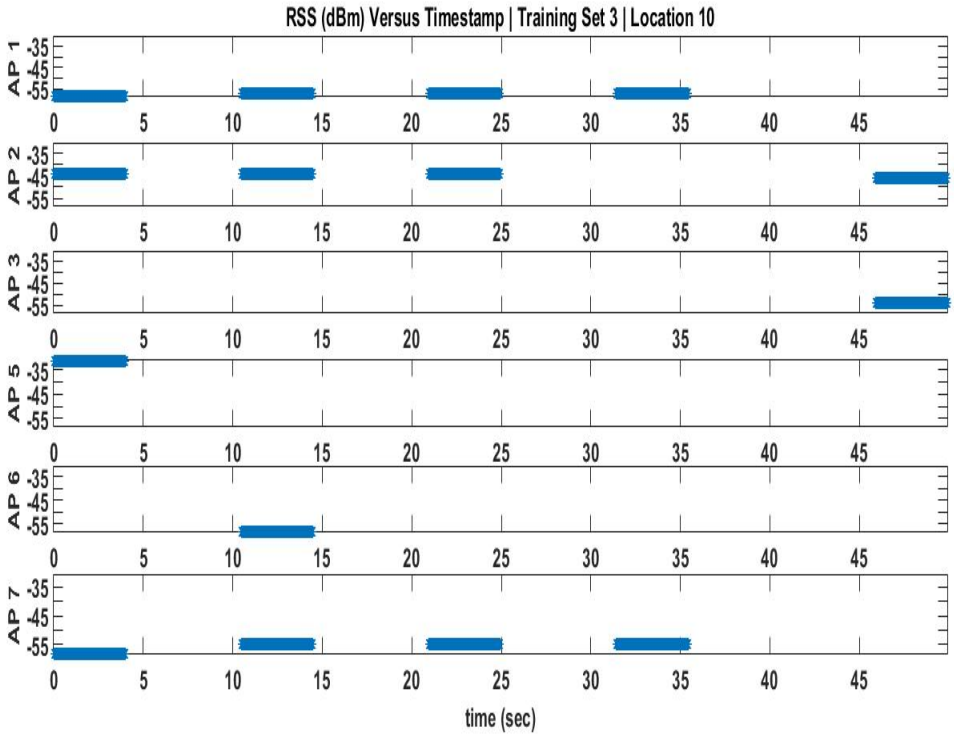


Figure 6.2: Observed RSS as a function of time at location 10 for training set 3

(in Figure 6.4) for all training sets (for location 1, all RSS values are greater than -50 dBm, while for location 10, all are less than or equal to -55). As a similarity of Figures 6.4 and 6.5, the duration of presence, is the same for all sets and both locations 1 and 10.

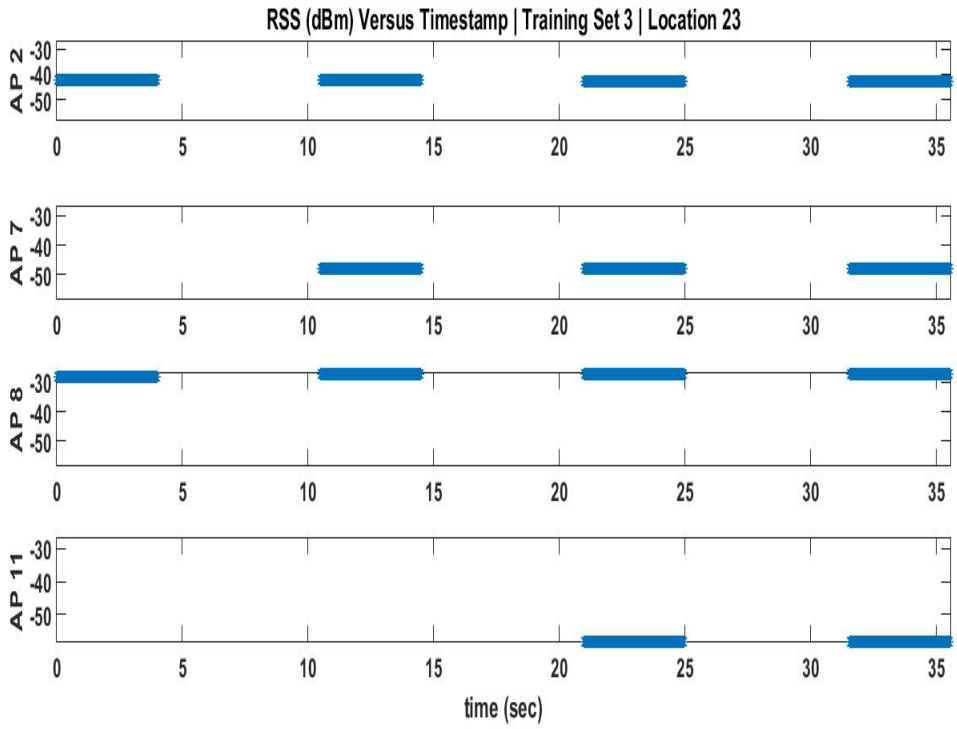


Figure 6.3: Observed RSS as a function of time at location 23 for training set 3

6.2 Challenges Faced

In addition to the variation in RSS measurements, that made evaluating the proposed localization algorithm more difficult, the other main challenge faced in working with the experimental data, include inconsistent and missing data. The first one, could be caused by some APs being inconsistent in their activity, meaning they were active for some time and inactive over other periods. The significant length of the periods of inactivity makes this explanation fairly unlikely however. The missing data, refers to interruptions in the scan process in Wavemon, the reason for which is unclear, also resulted in time windows during which no data was observed (in this case, the APs are active, but our software fails to detect RSS or record the measurements). Different strategies were suggested to deal with these 2 issues. These included, ignoring the absent APs, considering a very small value close to

zero for their RSS, estimating the distribution of the data based on the known values, or simulating the missing data with a simple shadowing model. In dealing with missing AP measurements, we chose the second approach, where we assumed -100 dBm (close to zero mW) for the signal strength of unobserved APs.

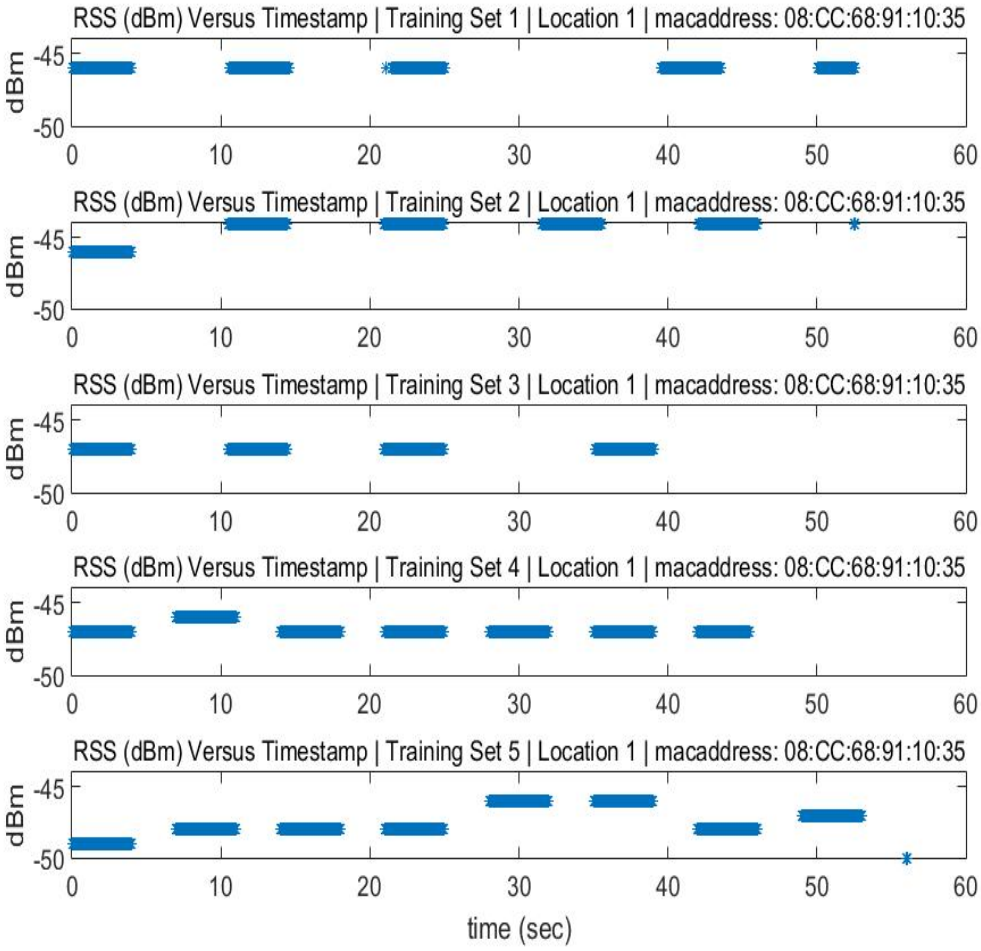


Figure 6.4: Observed RSS as a function of time at location 1 for all training sets

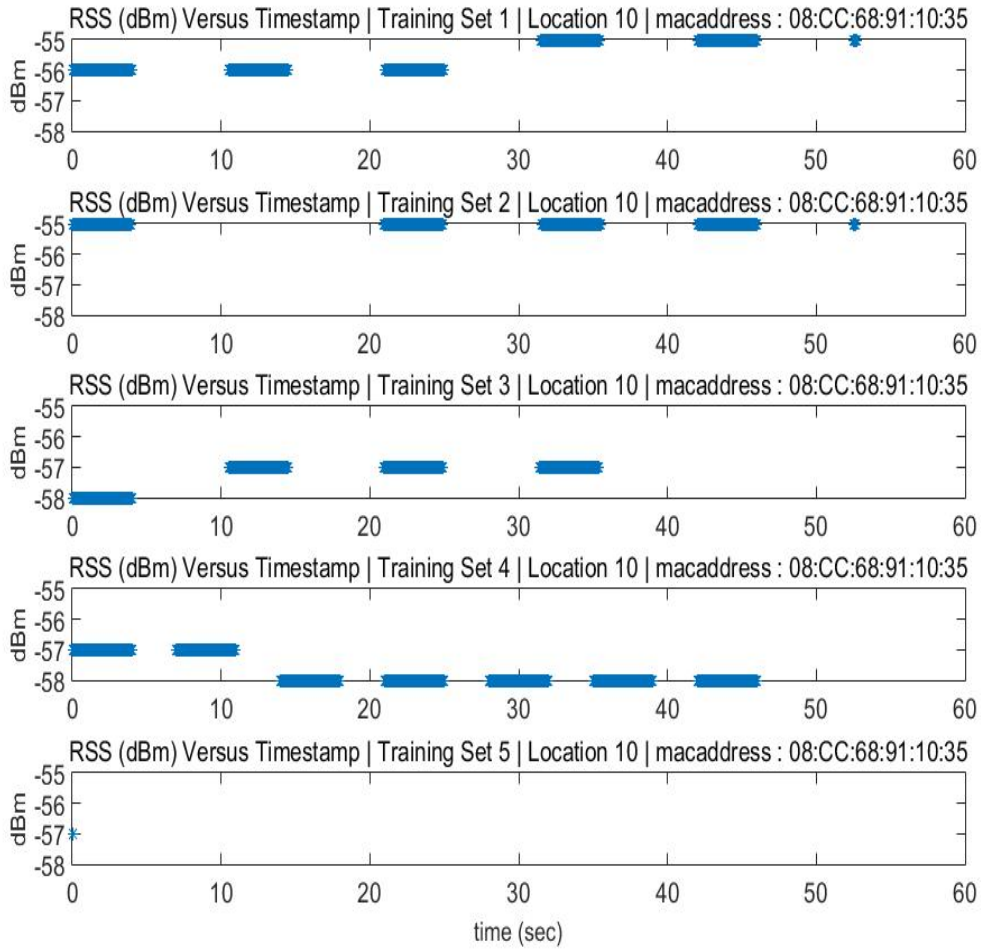


Figure 6.5: Observed RSS as a function of time at location 10 for all training sets

Chapter 7: Conclusion and Future Work

This thesis presents an efficient solution to the indoor positioning problem, both in terms of hardware cost and computational complexity. A sequential detection approach that combines KNN and trellis search to perform RSS-based positioning has been presented. Using a Bayesian metric, the most likely path, conditioned on a time series of RSS observations, traveled by the user is determined. The KNN algorithm is applied to compute position likelihoods conditioned on each signal strength measurement. Experimental results on real data collected in an indoor environment show strong performance of the proposed method, even for short sequences.

There are several possible extensions to this work. For example, a natural next step would be to investigate the performance of the proposed algorithm, under practical operating conditions. This includes investigating performance, in more challenging physical spaces, when a tighter grid of possible locations is considered. Also, the path traveled by the user, could be generated by collecting online data using a robot traveling a continuous path, while gathering Wi-Fi measurements in the environment. With this approach, online measurements could occur at any location rather than only at grid locations.

Secondly, with all the variation observed in RSS over time, it would be wise to collect more offline and online data. New measurements can provide additional information about the behavior of access points, which can then answer some of the questions posed in Chapter 6. In collecting more data, it would be logical to consider different locations within the Nguyen Engineering Building, other campus buildings, and other indoor environments beyond campus.

Additionally, a variety of AP selection techniques can be used when performing dimensionality reduction. With additional datasets, the robustness of competing AP selection algorithms could be evaluated across time and locations. For example we considered AP

selection using information theory, in addition to PCA (principal component analysis). The simple method of selecting the strongest APs, resulted in the best performance among the methods we considered for the dataset we investigated. In considering additional datasets, however, one may find that more sophisticated AP selection techniques may find the most descriptive and informative features.

As a final suggestion for extending this work, efforts should be made to understand the periodic breaks in measurements collected by Wavemon, and alternative software for RSS measurement collection should be explored.

Bibliography

Bibliography

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C (Applications and Reviews), IEEE Transactions on,* vol. 37, no. 6, pp. 1067-1080, November 2007.

[2] F. Seco, A. R. Jimenez, C. Prieto, J. Roa, and K. Koutsou, "A survey of mathematical methods for indoor localization,"*Proceedings of the IEEE International Symposium on Intelligent Signal Processing,* Budapest, Hungary, pp. 9-14, August 2009.

[3] M. Luk, (2016, July. 20), Time-Series Analysis: Wearable Devices using DTW and KNN, *SFL Scientific* [Online]. Available:

https://sflscientific.com/case-studies/2016/6/4/time-series-analysis-fitbit-using-dtw-and-knn

[4] L. Wang, L. Khan, and B. Thuraisingham, "An effective evidence theory based k-nearest-neighbor (KNN) classification,"*Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology,* Sydney, Australia, pp. 797-801, 2008.

[5] S. Fang, and C. Wang, "A dynamic hybrid projection approach for improved wi-fi location fingerprinting," *Vehicular Technology, IEEE Transactions on,* pp. 1037-1044, 2011.

[6] Y. Chen, Q. Yang, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation,"*Knowledge and Data Engineering, IEEE Transactions on,* pp. 877-888, 2006.

[7] P. Mirowski, D. Milioris, P. Whiting, and T.K. Ho, "Probabilistic radio-frequency fingerprinting and localization on the run," *Bell Labs Technical Journal, Alcatel-Lucent,* pp. 111- 133, 2014.

[8] C. Feng, W.S.A. Au, S. Valaee, and Z. Tan, "Received signal-strength-based indoor positioning using compressive sensing," *Mobile Computing, IEEE Transactions on,* pp. 1983- 1993, 2012.

[9] K. Kaemarungsi, and P. Krishnamurthy, "Modeling of Indoor Positioning Systems based on Location Fingerprinting,", *IEEE Computer and Communications Societies, Twenty-third AnnualJoint Conference,* Vol. 2., 2004.

[10] AKM. M. Hossain, H. N. Van, Y. Jin and W. Soh, "Indoor localization using multiple wireless technologies," *Mobile Adhoc and Sensor Systems, IEEE International Conference on.,* 2007.

[11] M.K. Hoang, J. Schmalenstroeer, C. Drueke, D.H. Tran Vu, and R. Haeb-Umbach, "A hidden Markov model for indoor user tracking based on WiFi fingerprinting and step detection," *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*, 2013.

[12] R.A. El-Khoribi, H. S. Hamza, and M. A. Hammad, "Indoor localization and tracking using posterior state distribution of hidden Markov model," *Proceeding of the 8th International ICST Conference on Communications and Networking,* China, 2013.

[13] J. Seitz, J. Jahn, J.G Boronat, T. Vaupel, S. Meyer, and J. Thielecke. "A hidden markov model for urban navigation based on fingerprinting and pedestrian dead reckoning," *In Information Fusion (FUSION), 13th Conference on,* pp. 1-8, 2010.

[14] T. Sebastian, "Exploring artificial intelligence in the new millennium," Robotic mapping: a survey, pp. 1-36, 2003.

[15] P. Hbert, S. Betg-Brezetz, and R. Chatila, "Probabilistic map learning: Necessity and difficulties," *Reasoning with Uncertainty in Robotics, Springer Berlin Heidelberg,* pp. 307-321, 1996.

[16] M. Magnabosco, and T.P. Breckon, "Cross-spectral visual simultaneous localization and mapping (SLAM) with sensor handover," *Robotics and Autonomous Systems 61.2,* pp. 195-208, 2013.

[17] G.D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE,* pp. 268-278, 1973.

[18] J.K. Nelson, and H. Roufarshbaf, "A tree search approach to target tracking in clutter," *Information Fusion, 12th International Conference on, IEEE,* pp. 834-841, 2009.

[19] Wavemon - A wireless network monitor, (n. d.). *Ubuntu manuals*[Online]. Available: *http* : *//manpages.ubuntu.com/manpages/xenial/man*1*/wavemon*.1.*html*. Accessed Mar. 22. 2017

[20] Pioneer P3-DX, (n. d.). *Omron*[Online]. Available: *http* : *//www.mobilerobots.com/ResearchRobots/PioneerP*3*DX.aspx*. Accessed Mar. 22. 2017

[21] Equipment. (n.d.). *Technische Fakultat AG Technische Informatik* [Online]. Available: *https* : *//www.ti.uni*−*bielefeld.de/html/research/equipment.html*. Accessed Mar. 22, 2017.

Curriculum Vitae

Negar Etemadyrad received her BS degree in Electrical Engineering from Ferdowsi University of Mashhad, Mashhad, Iran in 2013. Later, she continued her MS studies in Electrical Engineering at George Mason University, Fairfax, Virginia, and received her MS as a secondary degree in Spring 2017. Along with her master's studies, she was the first author of one conference paper, and worked as a teaching, as well as research assistant, at George Mason University. She received the Outstanding Academic Achievement Award for 2017 from ECE Department, at GMU. Her area of active research includes communications, signal processing, localization, and tracking.