DIRECT INFERENCE OF LOCATION-RELATED CONTEXT FROM WIRELESS SIGNAL STRENGTH

BY GAYATHRI CHANDRASEKARAN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Marco Gruteser and Richard P. Martin

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2011

ABSTRACT OF THE DISSERTATION

Direct Inference of Location-related Context from Wireless Signal Strength

by Gayathri Chandrasekaran

Dissertation Director: Marco Gruteser and Richard P. Martin

In this dissertation we derive location-related context like mobility-states, co-mobility, speed and decelerations directly from the wireless signal strength information. The key insight is that the time-series of signal strength is robust to environmental factors that typically negatively affect the RSS-based localization systems. Therefore, inferring these physical properties directly from the time-series of wireless signal strength is more accurate than deriving them from location estimates.

We apply correlation and time warping algorithms to the time series of wireless signals to infer these properties. Our trace-driven experimental approach shows that our inference techniques can work with minimal infrastructure, are computationally efficient, requires no explicit user participation and can produce higher accuracies than location-based systems. We have also experimentally identified the factors that limit the accuracy of indoor localization and have proved the existing assumptions behind theoretical lower bounds of indoor localization incorrect.

Our results will enable new context aware applications, because accurate estimates of co-mobility and speed offer a richer set of primitives available to applications. Such applications can derive user mobility states like walking, running, driving or social states, such as if a user is in a meeting or alone.

List of Tables

List of Figures

Acknowledgements

I would like to express my heartfelt gratitude to

- My husband. My husband Mr. Ravi Vridhagiri patiently waited as I took my time to complete my PhD and gave me a comfortable life throughout without letting me worry about running a family. His resourcefulness and a positive "can do" attitude has always inspired me. The numerous number of hours he spent driving the car at nights with me to help me collect my experimental vehicular datasets requires a special mention. Thanks for all the help!

- My Parents. My parents (Dr. E. Chandrasekaran and Mrs. Banumathy Chandrasekaran) always kept my priorities ahead of theirs and encouraged me throughout my PhD. Right from my childhood days, my parents have held high hopes on me and have always guided me to success. Mere words cannot do justice to their sacrifices.

- My friends at WINLAB. I would like to thank Mesut Ali Ergin for his immense help during my early years of graduate study. I have always been inspired by the clarity in his thoughts, his insights to research and his hands on approach to answering a lot of research problems. During the last year and half of my PhD, I enjoyed my discussions with Tam Vu and have been quite surprised by his abilities to interact with everyone around without any inhibitions. I would also like to thank my friends Sanjit Krishnan Kaul, Kishore Ramachandran, Chandrasekaran Raman, Begum Turgut and John-Austen Francisco for making my graduate life joyful.

- I would like to extend special thanks to Prof. Thu D. Nguyen of the CS department for his efforts at inspiring me to teach. I initially started my discussions with him when I TA-ed his Architecture course. Through this course, I realized that I had special interest for teaching. He further helped me realize my teaching potential by encouraging me to

handle Operating Systems course independently as a lecturer. He served as my mentor for this course and helped me get through teaching the course with ease.

- My advisors, Prof. Marco Gruteser and Prof. Richard P. Martin. My advisors taught me several things about research and writing and shaped me up as a researcher. They both gave me sufficient academic freedom to pick a problem of my choice, supported me with their grants and guided me through the various intermediate steps with enough patience. Without their help, I would be nowhere.

- Ivan seskar for his help in interpreting my experimental results at various stages. His hands on approach has always been a great source of inspiration. I would also like to extend special thanks to all the other faculty members of WINLAB, specially Professor Ray Choudhury, for sharing their wisdom with me during the course of my PhD.

- Dr. Alex Varshavsky for being my mentor at AT&T Labs and working with me and guiding me on my Speed Estimation project.

- Prof. Lawrence Rabiner for taking the time to review my work on the speed estimation project and suggesting ways to improve the results.

- Prof. Vinod Ganapathy and Dr. Ramon Caceres for serving in my thesis committee.

- My in-laws for their indispensable help on the home-front when we needed it the most.

- My dear daughter Raghavi Aiyer. She has been the greatest source of joy in my life. Though her addition in my life has been pretty recent (18 months), her influence on me is immense.

Table of Contents

Chapter 1

Introduction

The wireless industry has seen a tremendous growth within the last decade to the extent that it has penetrated into the lives of every human being. The pervasiveness of the wireless devices have opened up immense opportunities to study user context and provide information relevant to the user's current situation/activity. The term "context" has had several definitions in the past. The most popular definitions are the ones given by Schilit and Theimer [82] which defines context as *Location, Nearby People or Object and Changes to those Objects*. Anind K. Dey redefined context in [26] as *Emotional State, Focus of attention, location and orientation, date and time, nearby objects*. He further refined the definition in his thesis [27] to *Any information that characterizes the situation of the person, object or the entity*. The common elements in all these definitions are Who (Identify), When (Time), Where (Location) and What (Activity). In this thesis, we focus on inferring location-related context, which is defined as, the set of physical properties that could be derived from accurate location estimates. Figure 1.1 depicts some of the location-related properties like speed, accelerations, mobility state and co-mobility information that we consider in this thesis.

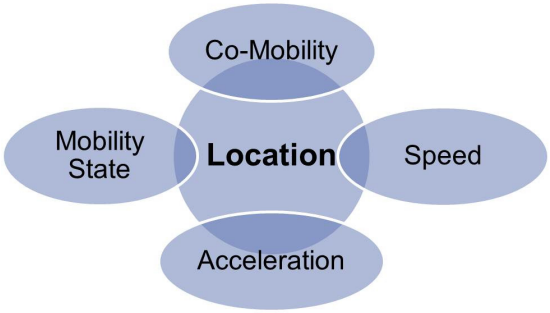


Figure 1.1: Location-related Context

All these properties can be derived accurately if we had perfect location estimates. While

location estimates from GPS have been accurate outdoors, the power drain due to the GPS has been shown to be two orders of magnitude higher than that of using GSM [59]. Also, GPS can have very poor accuracy indoors. Other outdoor location estimation techniques like GSM based localization [18] or Wi-Fi based localization [10] have low accuracies in the order of 60m and 40m respectively. Due to the noisy nature of the wireless medium, the indoor localization estimates using Wi-Fi suffer a median error of 3m. While adding more receivers could bring down the indoor localization errors, there still exists a trade-off between adding more resources for improving localization accuracy versus reusing the communication infrastructure to sustain reasonable localization accuracy. Therefore, deriving the other location-related properties from location estimates that exhibit high errors may not be the most optimal means to deriving them.

My dissertation addresses the above issue by inferring the location-related properties like speed, co-mobility, mobility states and decelerations directly from the time-series of wireless signal strength. The RSS-time series exhibits several interesting properties that make them an attractive candidate for inferring other location-related properties.

- The presence of continuous RSS samples in the time-series ensures that inferences about the present model the past as well. For example, if the last predicted speed was 40mph, the current speed cannot differ substantially from 40mph. This property makes the present inferences more meaningful.

- Analyzing the time-series for deriving location related properties does not require any environmental modeling and could reuse the minimal communication infrastructure.

- The transient distortions in RSS introduced by shadow fading and small scale fading do not have a significant impact on the overall shape of the time-series. Therefore, the location-related properties that are derived from the time-series are robust to these fades.

My thesis is divided into three parts. The first part of my thesis [15] systematically analyzes the factors that result in indoor location estimation errors. The goal of this work is to show that the factors that typically result in localization errors indoors cannot be eliminated even with dense receiver deployments which makes deriving other location-related properties accurately very challenging. We do this by experimentally evaluating the different Received

Signal Strength (RSS) based indoor localization techniques in a 400 node (20×20) high density indoor wireless testbed [5]. Through the experiments, we

- Evaluate the different indoor localization algorithms on a common deployment framework with similar number of training and testing points.
- Understand the effect of different environmental/external factors like receiver density, environmental noise, training density, etc. on the performance of the different classes of localization algorithms

Our results show that the fundamental performance limitation of any RSS based localization system stems due to the simplistic assumptions behind modeling the path loss and the environmental noise. We show that merely increasing the receiver density without considering the environmental noise and the resulting path loss model observed at the receiver can degrade localization accuracy. From our results, the best achievable indoor localization accuracy in an *optimally-placed dense* environment was in the order of a $24cm$. However, the typical errors that were observed with just 4 receivers (similar to that of the communication infrastructure) tends to be close to 3m. Therefore, using these incorrect location estimates as a means to estimating other properties like speed or co-mobility would be error prone.

The second part of my thesis [13, 14] therefore focuses on using a continuous time series of RSS observations to detect a location-related property called Co-Mobility. The key observation behind the technique lies in the fact that when environmental factors like shadow fading affect the SNR observed from the co-moving devices, all co-moving transmitters get equally affected. So, monitoring a continuous time series of RSS discounts the effects of transient fades that are very common in the wireless environments and takes advantage of the similarity of the RSS fades over long durations. I show that this technique does not require environmental modeling, is more resilient to noise and can function with minimal infrastructure and calibration.

The third part of my thesis [16, 17] extends this observation in co-mobility to a scenario where the transmitters move along the same path but at different times and different speeds. Moving along the same paths result in similar fades except in the time axis where the traces are either stretched or compressed in time depending on the relative speeds of the traces. We

use this observation, apply correlation and derivative dynamic time warping techniques to determine the speed of one trace relative to the other. Through real experiments, we show that our technique can achieve speed estimation accuracies close to $\pm 5mph$

In Summary, the overall contributions of my dissertation are:

- Analyzing the lower bounds of indoor localization in a high density indoor Testbed [5]. Results show that the main factor influencing the performance of lateration based localization algorithm is the underlying assumption about the propagation model. It also demonstrates that increasing the receiver density beyond a limit can have a negative impact on the performance of algorithms.

- Studying the accuracy-infrastructure tradeoff for RSS based localization systems.

- Proposing techniques for detecting transmitter co-mobility. The time series of signal strength values observed from the transmitters that move together exhibit a very high correlation (0.8) with each other in our results. We generalize this observation for co-moving transmitters having different radios (802.11 and 802.15.4). We also use RSS traces from GSM enabled co-moving mobile phones outdoors to show that such an inference can be extended outdoors.

- Tracking vehicular speed variation with high accuracy $\pm 5mph$ using GSM enabled mobile phones. We show that the GSM RSS profile (time-series) observed from the mobile phones on vehicles moving along a given road segment are similar except in the time axis due to their relative speed up. Using a training GSM RSS trace from a vehicle moving at known speeds, we apply time warping techniques to estimate the unknown speed of the moving vehicle.

Chapter 2

Background

In this section, we classify the related work into three major categories. We begin with a review of the existing indoor and outdoor localization systems. We then continue reviewing some of the specific location-related context inference systems, namely, co-mobility detection systems and vehicular speed detection systems.

2.1 Indoor and Outdoor Localization systems

In this section, we summarize the experimental setting and the reported accuracy limits for several Wi-Fi based indoor localization algorithms. We then continue to give a brief overview of the other indoor localization schemes that use other radios such as 802.15.4, UWB, IR, GSM, etc. We conclude this section with a discussion of the outdoor location determination algorithms and their reported accuracies.

2.1.1 Wi-Fi based Indoor Localization System

RADAR [11], the first localization algorithm for IEEE 802.11 transmitters in this category, uses RF Fingerprint information (vector containing known locations of transmitter along with a measure of the observed signal strength at different receivers) observed at three receivers and performs a nearest neighbor matching algorithm to determine the location of the transmitters with a three meters median accuracy. Several other systems worked on enhancements to RADAR. [83] proposed two different localization systems, namely, CMU-PM and CMU-TMI where the CMU-PM algorithm performed pattern matching similar to RADAR with enhanced training set while the CMU-TMI performed triangulation, pattern matching and interpolation by first profiling the environment, interpolating the RSS on locations where the measurements were not made and finally matching the obtained RSS. The reported median errors were 1m

Technique	Type	Area (m^2)	Num APs	Num Training	Median Error(m)	75^{th} % Error(m)	Max Error(m)
RADAR [11]	Classification(Scene Matching)	22.50×43.49	3	70	2.93	4.69	24.99
AURA [83]							
CMU-PM	Classification(Scene Matching)	Not Specified	5	17	0.98	3.29	>9.75
CMU-TMI	Classification(Scene Matching with interpolation)	Not Specified	5	17	1.94	3.29	8.50
LEASE [54]	Classification(Scene Matching) with Interpolated Grid	68.58×43.90	5	100	2.29	n/a	n/a
		76.20×53.34	4	100	0.61	n/a	n/a
Ref [94]	Classification(Probabilistic)	68.28×35.94	4	110	1.07	1.22	7.32
HORUS [95]	Classification(Probabilistic)	68.28×35.94	21	172	0.39	0.55	4.99
		11.80×33.13	6	110	0.51	0.90	4.99
M1 [60]	Lateration(Bayesian inference)	60.96×24.38	4	115	5.49	6.71	27.43
		64.00×42.67	5	215	5.49	6.1	27.43
Ref [21]	Lateration with LLS	60.96×24.38	4	286	6.1	9.14	42.67
	Lateration with NLS	60.96×24.38	4	286	3.35	6.1	33.53

Table 2.1: Summary of the Reported Experimental Accuracies for Various Localization Algorithms

and 2m for CMU-PM and CMU-TMI respectively. [54] performed RSS interpolation along with pattern matching with heavy training (more than 100 training locations) on two different environments and reported median errors of 2m and 0.6m in the two environments. [94, 95] used the RSS distribution in every location and applied probabilistic techniques to determine location. [55] was a Wi-Fi based context inference system which first identifies if the transmitter was mobile or stationary and then refines its location determination system to account for the state using Hidden Markov Model (HMM). [32] uses Bayesian learning algorithm on RF fingerprints observed at three or more receivers to obtain a median 802.11 localization accuracy of 3-4 meters. The most accurate 802.11 location system to date is [57] which uses Hidden Markov Model and Bayesian inference derived from observations at nine different receivers yielding a median accuracy of one meter. While most systems based on Wi-Fi uses signal strength, AeroScout [1] uses 802.11-based TDOA location solution and [33] uses AOA based solution.

2.1.2 Other Indoor Localization Systems

There are several non Wi-Fi based indoor localization systems. SpotON and LANDMARC systems use RFID tags to determine location and report a median accuracy of 2m. UWB based systems [8, 72] transmit ultrashort pulses on a wide set of frequencies and use precise Time Of Arrival measurements to achieve a positioning accuracy of 10cm. [92] used infrared to perform indoor positioning. However, all the above techniques require specialized infrastructure to be deployed for performing localization. On the other hand, the Wi-Fi based systems typically re-use the existing wireless infrastructure. A number of systems [68, 91] use the GSM mobile cellular network for indoor localization. The key idea that behind this technique is the use of wide signal-strength fingerprints. The wide fingerprint includes the six strongest GSM cells and readings of up to 29 additional GSM channels and this can yield a median accuracy of 5m.

2.1.3 Outdoor Localization Systems

Global Positioning Systems [35] is the most widely used outdoor positioning system. The GPS receiver uses timing information to perform distance measurements to the orbiting satellites and in turn uses trilateration to determine its location. In indoor environments, the absence of line of sight between the GPS receiver and the satellites results in poor location estimation accuracy. Although GPS is quite accurate outdoors, not every mobile device is equipped with this. The power consumption of GPS is also quite high which mandates the need for alternate means of positioning. Cell Phone Positioning is the widely used alternative to GPS. The cell identification (Cell-ID) method can identify the tower that is closest to the mobile phone thereby identifying the mobile phone within a 100-200m range depending on the cell size. Better cellular phone based techniques based on RSS fingerprinting were developed by [18] that reports accuracies in the order of 60m. The Placelab [10] project took a slightly different step where they tried to use the Wi-Fi beacons captured through War-Driving to locate a Wi-Fi enabled device outdoors. The accuracies reported by this technique varied from 20 to 40m depending on the Wi-Fi availability.

2.2 Co-Mobility detection systems

The previous work on detecting co-located and co-moving objects have either been based on absolute location of the transmitters obtained using localization indoors and GPS outdoors or from p can be classified roximity sensing using short range infrared (IR) or Bluetooth devices. We know of no other work that infers co-location or co-movement directly from signal strength measurements. In this section we classify the related work into three main categories.

2.2.1 Mobility Detection Systems

Several earlier studies have concentrated on distinguishing mobile and stationary transmitters. [84] determines mobility from GSM traces using seven different metrics one of which is the variance in Signal Strength which is similar to our approach. Similarly, [63] discusses detecting mobility from RSSI in WLAN. LOCADIO [55] again used variance to detect mobility and combined it with a two state Hidden Markov Model (HMM) to eliminate oscillations between the static and mobile states. We build on this work— detecting mobility is an integral component of the DECODE technique.

2.2.2 Proximity-based Co-Mobility detection

Proximity based co-location inference techniques mainly consist of using short range IR or Bluetooth devices to estimate distance between the transmitters. The Reality Mining project [31] [30] used Bluetooth capable GSM phones to record the other nearby bluetooth devices and transmit them to the central server for inferring social interaction patterns. SpotOn system [45] used radio signal attenuation to estimate the relative distance between the special tags. Though these techniques look attractive for co-location detection, they require tracking software on the devices themselves and are effective only for detecting devices that have the same technology. Our scheme is more generic as it involves measurement of RSSI which is common to GSM, WLAN, Zigbee, Bluetooth.

2.2.3 Distance Threshold based Co-Mobility Detection

This detection technique involves estimating the locations of different transmitters over time and deriving conclusions about co-movement based on the distance between the estimated positions of the transmitters. Recent efforts have resulted in a plethora of methods to determine the locations of transmitters as discussed in Section 2.1;

As described in Section 2.1.1, there are several Wi-Fi based indoor localization systems [11, 21, 54, 57, 60, 83, 94, 95] that use RSS from fixed wireless APs to derive user positions. Further, the average localization accuracy employing RSS in a 802.15.4 (Zigbee) network [19] and an active RFID system [20] is about the same with median errors around 3-4m when using four receivers. While the recent papers [56, 61] have reported a higher accuracy localization techniques, these techniques require transmitters to perform synchronized communications which is not common across typical transmitters that we analyze in this work. Further, these papers have not reported the accuracy in a mobile environment questioning its applicability for the detection of co-movement.

Intuitively one can derive co-movement information with threshold detection on the distance between two transmitters. Compared to DECODE (in signal space) all these localization systems require three or more receivers to work in concert, whereas DECODE can be used even with just one receiver. In addition, the accuracy results reported for Wi-Fi localization raise questions about the precision of such a detection approach. We will further address these question in detail in our work.

2.3 Vehicular Speed Estimation systems

In this section, we review the existing studies on vehicular speed estimation and classify the existing work on vehicular speed estimation based on the modality of sensing as follows.

2.3.1 Fixed Infrastructure based sensing

By far the most common of highway speed estimation system is the inductive loop detectors [22, 23, 44] which are based on on-road sensors embedded in the pavement. Traffic cameras [28] have also been installed on roads that uses a sequence of image captured on several

cameras on the road to calibrate the speed of a moving vehicle. [23] has shown that speed estimation errors using loop detectors for a vehicle traveling at over 50mph can be in the order of 20mph to 120mph. Besides that, they suffer from their limited reliability and high installation cost, which makes it hard to maintain significant coverage on the road network.

2.3.2 Smartphone based sensing

Using GPS enabled smart-phones for sensing [4, 47] has gained huge popularity in the recent times due to its negligible deployment cost. These techniques, if adopted by a large number of users, can provide very accurate speed estimation on most roadways. However, frequent sampling of the GPS unit can result in fast battery drain on the mobile phone. [89] tried to overcome some of the energy limitations by sub-sampling the GPS and combining the Wi-Fi outdoor positioning along with map-matching to estimate speeds with high accuracy. Still, energy consumption remains higher than approaches that use existing phone signals. It also require software modifications on each handset which makes bootstrapping the service more difficult.

2.3.3 Cellular phone (GSM) based sensing

Unlike the smartphone based sensing, these techniques rely on the location of the cellular phone over time calibrated either using triangulation of the GSM signal strength [97] over time or Fingerprint matching of the phone successive signal strength readings [18] or the location where the cellular phone handsoff between towers [39, 88]. [85] uses the rate of change of RSS between successive samples to determine the speed. While all of the above techniques can overcome the bootstrapping (since the provider already has access to the signal strength information from phone) and energy issues that were present in smartphone based sensing, these can only estimate average speeds over segment of length typically over 100m. [9] uses the GSM network to infer traffic volume from call volume. None of these techniques can be used to track small variations in speed that are important for several traffic engineering applications. We differ from all the above techniques by estimating speeds with high accuracy. In addition, we are the first to show the possibility of using GSM signal strength for tracking temporary speed variations (for, example bottlenecks causing slowdowns).

2.3.4 Doppler shift-based sensing

Finally, [93, 98] makes use of the doppler shift in frequency caused by the moving transmitter to estimate speed. [93] can only perform coarse speed classification while [98] can predict the actual speed of the mobile. But the latter assumes the presence of strong Line of Sight(LOS) component between the transmitter and the receiver which can make this technique impractical.

2.4 Summary of Related Work

In this section, we describe how our work differs from all the above mentioned related works. First, we do not localize devices over time to infer these location-related properties. Therefore, our technique does not require infrastructure set up and calibration. Secondly, our work [13] was the first of its kind to make inferences about co-mobility directly from the signal space with just a single receiver. While there have been a lot of other proximity based sensing techniques proposed in the past for inferring social interactions, all these require the end users to upload their bluetooth traces to a central server in order to infer co-mobility. In contrast, our technique does not require bootstrapping, since, it works by sensing the existing communications from the wireless devices. The work that was closest in spirit to ours was [55], which observed the variance in RSS to infer mobility states but it did not extend its inference for co-mobility detection. Thirdly, the speed estimation algorithms that have been proposed in the literature using localization or handoffs can only estimate average speeds over a period of time due to the inaccuracies associated with localization or handoff zone prediction. Our work on speed estimation [16] differs from the rest by tracking fine-grained speed variations. While GPS based techniques could also track speed variations, they require explicit user participation in uploading the collected traces to a central server and impose a heavy battery drain on the end-user devices. Our technique, however, senses the existing cellular phone communication signals to infer vehicular speeds which therefore, does not impose any additional battery drain and eliminates the need for bootstrapping the system.

Chapter 3

Analysis of the Accuracy Limits for Signal Strength based Localization Systems

3.1 Introduction

Location is essential for many emerging applications from a diverse set of areas including asset tracking, workflow management, geographic routing, and physical security. Wireless networks offer an unprecedented potential for realizing many of these applications. Given that wireless devices are carried by many people and attached to many objects and all modern radio chipsets include the hardware necessary to measure and report the received signal strength (RSS) of transmitted packets, there is a tremendous cost and deployment advantage to re-using the existing RSS infrastructure of the communication network for signal strength-based localization purposes.

Over the past years, algorithmic advances have yielded accuracy improvements from RADAR's [11] median 3 m error to less than 1 m median error [54]. A significant further improvement to about 40 cm median error has been obtained using a larger number of landmarks (base stations), 21 instead of the 3–5 used in previous experiments [95]. Since radio environments are becoming increasingly dense, this points to possible further accuracy improvements through using additional measurement nodes. Particularly, cooperative localization techniques [69, 70] where clients also contribute RSS measurements could provide readings from tens to hundreds of nodes. The limits of localization performance in such settings remain an open question.

In this chapter, we thus perform an empirical quantification of the accuracy limits of RSS localization on commodity wireless hardware. We try to understand the important factors that limit the accuracy of indoor localization by performing experiments in a controlled extremely dense laboratory environment with a single transmitter and up to 369 landmarks, which represents an ideal scenario for localization algorithms. Traces were collected using the ORBIT

testbed, which is a 400 node indoor wireless experimental apparatus placed in a 3600 sq ft. area. Using the ORBIT platform allowed us to capture long, high quality packet traces in a dense environment free of major shadowing and with limited multipath effects.

We use a combination of theoretical as well as trace-driven analysis on this dataset. Our theoretical work uses a traditional Cramér-Rao Bound (CRB) analysis, which has previously been used to establish bounds on location estimation variance [69]. We then use a trace-driven emulation to characterize the performance of different algorithms. In order to show the generality of our results as well as compare localization strategies, we used algorithms with widely divergent mathematical foundations. They range from classification approaches such as RADAR [11] over probability density exploration methods such as H1 [34] to multi-lateration such as Non-Linear Least Squares (NLS) [21].

Specifically, we found that:

- RSS based localization can achieve median errors as low as 0.24 m, with a maximum error of 1.5 m. Interestingly, while NLS performed the best under perfect synthetic conditions, it has the worst performance for real RSS observations, with a mean error of 1.6 m and a maximum error of 5.4 m.

- classification and probability density exploration algorithms had fundamentally worse performance using perfect synthetic input, because of a combination of their inherent discretization effects as well as the leave-one-out technique we employ to create our testing dataset.

- for lateration-based approaches, which assume a signal-to-distance function, quality of the RSS measurements is more important than the quantity of measurements. A subset of 179 landmarks whose data yield a good signal-to-distance fit provided best localization performance. Simply increasing the number of landmarks over this actually increased the median error from 24 cm to 58cm.

- classification algorithms are qualitatively less sensitive to variances and noise in the input set than lateration-based algorithms. Given RSS measurements that deviate substantially from standard models, these algorithms maintained good average and worst-case performance.

- accuracy improvements leveled off with about 100 nodes, the lower-bound of localization performance appears still limited by discrepancies between the underlying algorithmic models and the actual signal-propagation effects of indoor environments. For example, individual node differences due to differences in connectors, thermal effects, and local noise floors, as well as multi-path effects caused by diffraction, reflection and shadowing are not explicitly accounted for in any of the algorithms and account for the remaining errors.

Three important implications of our results are the following. First, the CRB for unbiased estimators [51,70], which is widely used for assessing localization performance limits, does not represent an actual lower bound on localization performance for all algorithms, likely because its assumptions of unbiased estimators or normally distributed measurement errors do not hold for these algorithms. Second, the strategy of minimizing the square root of the sum of the residuals leaves much room for improved accuracy, so localization systems based on classifiers or probability density exploration are preferable to least squares. We explore these effects more in Section 3.4. Third, significant accuracy improvements are still possible, likely by algorithms that incorporate more accurate models of measurement noise.

3.2 Baseline Algorithms

3.2.1 Lateration Based Algorithms

Lateration-based algorithms [21, 58, 66] explicitly model the signal-to-distance effect on RSS. They estimate the position of the transmitter by measuring the distance to multiple receivers. In [60], the authors use a Bayesian graphical model based on lateration to find a location estimate. We select our representative subset from lateration-based algorithms as *Non-Linear Least Square (NLS)* [21], and *Bayesian Networks (M1)* [60], and explain them briefly next.

Non-Linear Least Square (NLS)

In NLS, estimating the true location of the transmitter (x, y) can be viewed as an optimization problem where the actual locations of the reference points (x_i, y_i) are known apriori and the distance estimates d_i are obtained from the signal-to-distance relationship. The problem then

becomes solving for the optimal $(\hat{x}, \hat{y})$ that minimizes the sum of residuals:

$$(\hat{x}, \hat{y}) = \arg\min_{x,y} \sum_{i=1}^{N} \left[\sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i \right]^2 \tag{3.1}$$

Bayesian Networks (M1)

The M1 algorithm uses Bayes Nets which encode dependencies and relationships among a set of random variables. The vertices of a Bayes Net graph correspond to the variables and the edges represent dependencies [41]. The networks used for localization encode the relationship between the RSS and the (x, y) location using a simple log-distance propagation model.

The M1 strategy describes the joint probability density of (x, y) as a function of the observed RSS. However, in general, there is no closed form solution for the returned joint distribution. Therefore, we use Markov Chain Monte Carlo (MCMC) sampling to draw samples from the joint density [37]. The resulting samples allow us to approximate the true PDF of the (x, y). M1 selects the averages of the drawn samples as the (x, y) location estimate.

3.2.2 Classification Based Algorithms

Classification algorithms, a.k.a matching algorithms, do not rely on a model of signal strength and distance relationship. Rather, they match RSS observations against an existing signal map. The term classification, as used in the machine learning sense, implies that the goal of the classifier is to map a potentially large input space into a much smaller space of *labels*. In the case of localization, the labels are a set of discrete (x, y) locations.

Previous approaches [11], [83] and [54] are examples of classical fingerprint matching algorithms. The location of a node is estimated by matching its fingerprints to the closest one in the signal map. Matching algorithms in [95], and [94] employ probabilistic inference to estimate the location of a node. Also, [78] uses Bayesian inversion to return the location that maximizes the probability of the RSS vector. The authors of [40] apply the same technique to the robotics domain and experimentally show that 83% of the time, the location error is within 5 ft.

Deriving closed form solutions for the lower-bound of most of these algorithms is not trivial, or the bound may not be existent at all. Indeed, framing localization as a map matching

problem naturally lends itself to machine-learning approaches, for which many algorithms have provably no closed form solution. We select our representative subset from classification-based algorithms as *RADAR*, *Gridded-RADAR (GR)* and *Highest Probability (H1)* and continue with a brief overview of these algorithms.

RADAR

[11] is a classical scene matching localization algorithm where the signal map, a set of fingerprints with known (x, y) locations, is provided as an input to the offline phase of the algorithm. In the online phase, when presented with a fingerprint of a node with unknown location, RADAR returns the location of the 'closest fingerprint from the signal map.

Gridded-RADAR (GR)

is an improvisation over RADAR where measurement area is sub-divided into a regular grid and the signal map provided in the offline phase is interpolated over the entire grid. The online phase is similar to RADAR with the exception that the "closest" fingerprint in signal space is chosen from the interpolated signal map. This approach has an advantage of obtaining a much finer-grained resolution as the regions which are not covered by the signal map can also be returned as location estimates.

Highest Probability (H1)

Given an area divided into a discrete set of points called tiles, the strategy used by H1 is to return the most likely (x, y) by finding the highest probable tile using Bayes' rule over the set of RSS values. In order to find the likelihood of the RSS-matching for each tile in isolation, H1 assumes that the distribution of the RSS for each receiver follows a Gaussian distribution. This assumption significantly simplifies the computations with little performance loss. Using Bayes' rule, H1 computes the probability of being at each tile on the floor, L_i, given the fingerprint of the localized object $\bar{S}_l$ as

$$P\left(L_i|\bar{S}_l\right) = \frac{P\left(\bar{S}_l|L_i\right) \times P\left(L_i\right)}{P\left(\bar{S}_l\right)}. \tag{3.2}$$

However fingerprint $\bar{S}_l = (\overline{s_{lj}})$ is some constant c and with no prior information about the exact object's location, H1 assumes that the object to be localized is equally likely to be at any location on the floor, i.e., $P(L_i) = P(L_j), \forall i, j$. Thus, Equation 3.2 can be rewritten as

$$P\left(L_i|\bar{S}_l\right) = c \times P\left(\bar{S}_l|L_i\right). \tag{3.3}$$

Without having to know the value c, H1 can just return the tile L_{max}, where $L_{max} = argmax(P\left(\bar{S}_l|L_i\right))$, by computing $P\left(\bar{S}_l|L_i\right)$ for every tile i on the floor. Up to this step H1 is very similar to the traditional Bayesian approaches [40, 94], with the exception of the Gaussian and variance assumptions.

Finally, [69] and [29] studied establishing theoretical lower bounds for the achievable localization performance using estimation techniques that employ unbiased estimators. Table 3.1 summarizes the median, 75^{th} percentile and maximum errors for various localization algorithms that were studied experimentally. Overall, we can see that none of the approaches have experimented with more than 21 access points and the state of the art approach [95] has its median, 75^{th} percentile and the max errors as 0.39 m, 0.55 m, and 4.99 m, respectively.

In addition to studying RSS-based localization in a high density setting with hundreds of landmarks, and reporting localization errors an order of magnitude lower than the state-of-the-art, in this work, we also focus on understanding the factors that limit the RSS-based localization performance from achieving near-zero errors.

3.2.3 Cramér-Rao Bound

Localization can be defined as an estimation problem where measurements like wireless signal strength, angle or time of arrival are provided to an estimator (i.e. the localization algorithm) to obtain the most likely position in the assumed coordinate system. In estimation theory, the Cramér-Rao bound (CRB) has been derived as a lower-bound on the variance of an estimator [73]. Although CRB has been applied to certain classes of biased estimators [43, 90], it is commonly used to bound the variance of unbiased estimators [51]. The CRB has frequently been used by researchers to assess localization techniques [29, 69, 70].

The CRB for an unbiased estimator is obtained from the inverse of the Fisher Information

Matrix (FIM) [51], given as

$$\mathcal{I}(\theta) = E\left\{\left[\frac{\partial \ln f(p|\theta)}{\partial \theta}\right]\left[\frac{\partial \ln f(p|\theta)}{\partial \theta}\right]^T\right\}, \tag{3.4}$$

where θ is the unknown parameter to be estimated from measurements p, which follows a probability density function $f(p|\theta)$. Intuitively, the FIM is an estimate of the curvature of the log likelihood function $\ln f(p|\theta)$. If the curvature is sharp, the parameter estimation becomes more accurate making the lower bound on the variance of the estimator(CRB) very small. The received power at a landmark location (x_n, y_n) from a transmitter at (x, y) can be modeled as [36]

$$P(x_n, y_n) = P_0 - 10\gamma \log_{10}(d_n/d_0) + S_n \quad \text{(dB)}, \tag{3.5}$$

where $d_n = \sqrt{(x_n - x)^2 + (y_n - y)^2}$ with $n = 1...N$ landmarks; P_0 is the received power at the reference distance d_0 from the source; γ is the path loss exponent; and S_n is the random variation of the signal measurements and assumed to come from an i.i.d. Gaussian distribution $\mathcal{N}(0, \sigma^2_{RSS})$. For the case of a single unknown transmitter location and N landmarks, the CRB for the variance σ^2 of an unbiased location estimator is given as

$$\sigma^2 \geq \frac{\mathcal{I}(\theta)_{xx} + \mathcal{I}(\theta)_{yy}}{\mathcal{I}(\theta)_{xx}\mathcal{I}(\theta)_{yy} - \mathcal{I}(\theta)^2_{xy}} \tag{3.6}$$

where $\mathcal{I}(\theta)_{xx}$ and $\mathcal{I}(\theta)_{yy}$ are the diagonal blocks and $\mathcal{I}(\theta)_{xy}$ and $\mathcal{I}(\theta)^T_{xy}$ are the off-diagonal blocks of the FIM in Eq.3.4. Details of the derivation can be found in [70].

Note that the above CRB, for any localization technique using RSS information, critically depends on:

- the number and topology of the landmarks and the transmitters to be localized,
- the ratio of the RSS standard deviation to the propagation constant (σ_{RSS}/γ), characterizing the signal and the propagation environment
- the assumption of RSS fluctuations due to an i.i.d. Gaussian distribution with a common variance σ^2_{RSS}.

Figure 3.1: (a)The movable node (Transmitter) attached to an antenna on a portable mast, (b)The 400 node ORBIT experimental testbed

3.3 Testbed Experiments

In this section, we begin by describing our experimental objectives and then explain how we performed experiments in order to fulfill these objectives.

3.3.1 Objectives

The objectives of our experiments are:

- To quantify the limits of different localization algorithms—M1, H1, GR, and NLS—under a dense, indoor landmark deployment with limited shadowing and multipath fading
- To understand how these limits compare to the CRB
- To understand how variations in RSSI observations limit the localization algorithms from achieving perfect results in this laboratory environment

3.3.2 Experimental Methodology

We performed our experiments on ORBIT [75], a large scale indoor wireless testbed. The ORBIT testbed consists of 400 small form-factor PCs, with two IEEE 802.11a/b/g wireless interfaces per node. The nodes are suspended from the ceiling as shown in Figure 3.1(a) and are placed in a 20×20 regular grid with an inter-node separation of 91.44cm (3ft) spanning a

total area of 3600 sq ft. We collected packet data traces from a subset of 369 nodes using one of the identical Atheros 5212 based 802.11a/b/g NIC in every node. The remaining 31 nodes were down for maintenance.

The data trace collection spanned two days and all 369 fixed ORBIT nodes were configured as IEEE 802.11a receivers operating at 5GHz, channel 44. The receivers used the Tshark packet sniffer utility to log the received signal strength indicator (RSSI) for every received packet. We used a movable ORBIT node attached to a portable antenna mast as our packet transmitter. Figure 3.1(b) shows the rubber-duck antenna mounted on the mast. The antenna rig was used to raise the movable node's antenna to the same plane receiver antennas were located as well as keeping transmitter antenna orientation the same throughout all transmitter locations. The transceiver diversity options were disabled in all our radios to eliminate unwanted RSSI oscillations. The transmitter was placed at 400 different locations, one below each one of the suspended ORBIT node for collecting the training dataset. The per-packet data trace collected at each of the receiver was then post processed to yield an average RSSI measurement over 1000 packets for the transmitter at every location. This resulted in a dataset with 400 training Points and 369 receivers for each training point. Note that our testing point locations overlapped with the landmark locations.

To evaluate the different algorithms, we use the well-known leave-one-out approach where the 400 point training set is split into 400 sets of 399 training points and one testing point. The difference between the known actual location (x, y) of the testing point and the localization algorithm's estimate $(\hat{x}, \hat{y})$ of the testing point derives the estimation error.

3.4 Results

In this section, we begin by highlighting our important contributions and provide detailed insights into our results. Table 3.1 summarizes our key findings. They are:

1. The M1 algorithm achieved the lowest median localization error of about 0.24 m, exceeding the best prior experimental result [95] by a factor of two

2. Adding a lot of landmarks can reduce the localization accuracy for lateration based algorithms. The best performance was achieved with a carefully selected cleaned subset of

	Scaled Results			Sanitized Results		
	Median (m)	Max (m)	Std.Dev. (m)	Median (m)	Max (m)	Std.Dev. (m)
M1	.58	26.87	1.87	.24	1.60	0.25
NLS	2.01	13.44	2.41	1.62	5.37	0.79
GR	.31	1.74	0.30	.36	1.97	0.32
H1	.33	1.82	0.29	.39	1.70	0.33
CRB	—	—	0.62	—	—	0.75

Table 3.1: Summary of Localization Accuracy and Precision

about half the landmarks

3. The general CRB for unbiased estimators is a poor benchmark of localization precision for the algorithms tested

The following subsection describe each of these results in detail.

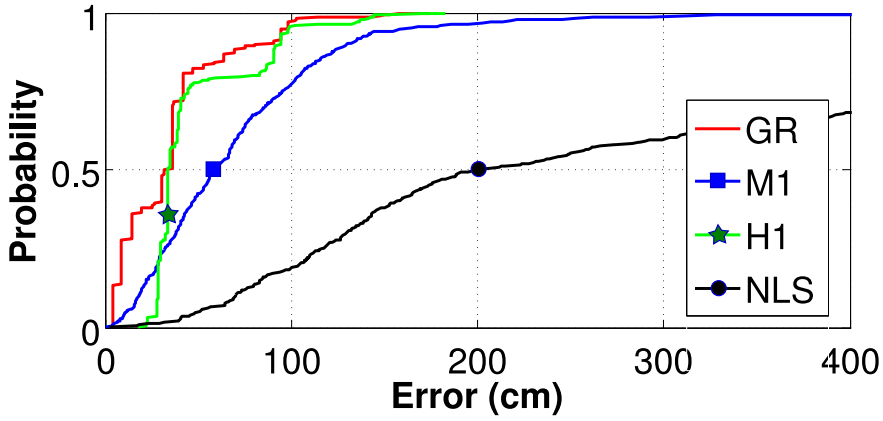


Figure 3.2: Error CDF plots representing localization performance of the selected algorithms from ORBIT experiment using all available 369 landmarks

3.4.1 High-density Localization Performance

Figure 3.2 plots the error CDF for the lateration algorithms (M1 and NLS) and the matching algorithms (GR and H1) with the training data collected from all 400 locations and 369 landmarks in the 20 by 20m space. We call this training set the *scaled dataset*. This dataset allows conclusions about how far localization error can be reduced with extreme measurement resources.

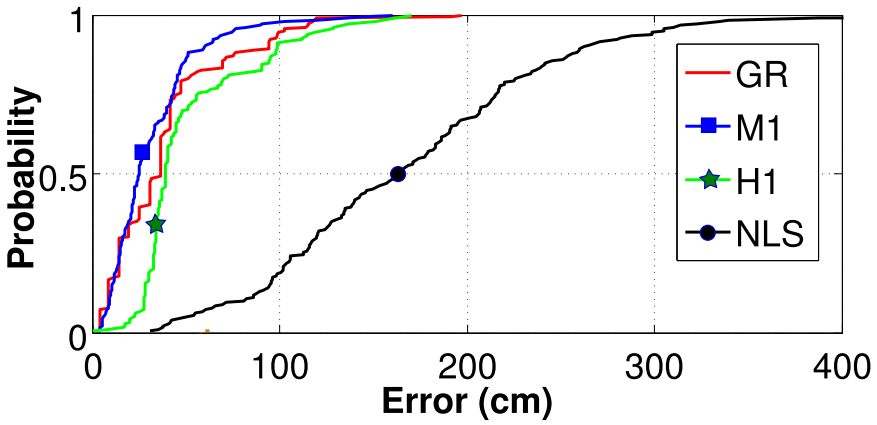


Figure 3.3: Error CDF plots representing localization performance from ORBIT experiment using the "sanitized" dataset with 179 Training and 179 Landmarks

GR and H1 have the best performance with median errors of 0.31 m and 0.33 m, respectively exhibiting only an 16% improvement over the best prior reported median accuracy of 0.38 m with 21 landmarks.

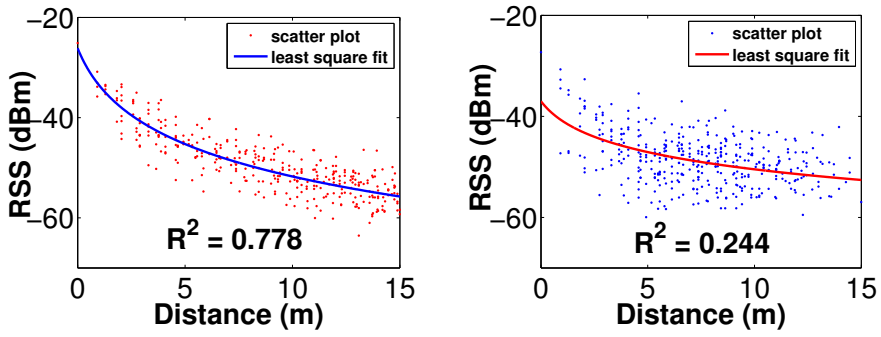
Table 3.1 shows the median and maximum errors for all four algorithms. The lateration algorithms, in particular NLS, show much higher errors both in terms of median and maximum. M1's maximum error of about 27 m exceeds that of H1 and Gridded RADAR 15-fold. This motivates us to further explore the causes for such outliers.

3.4.2 Sensitivity to Data Quality

We define the *data quality of a landmark* based on how well the signal strength measurements for the landmarks match a distance to RSS propagation model fitted on the data. The lateration algorithms estimate the propagation parameters from the measurements based on this distance to RSS fit. Recall that in free space, the signal power decays linearly with log distance.

Figure 3.4 plots the distance to RSS relationship together with the fitted free-space propagation model for two different landmarks using the same set of training points. We observe that the quality of the fit in terms of R^2 differs significantly.

To investigate the relatively poor performance of the lateration algorithms in the scaled dataset, we sanitize the data by removing low quality landmarks whose co-efficient of determination $R^2 < 0.5$. We empirically determined that this threshold significantly improves localization performance. After filtering, 179 landmarks and the corresponding 179 training



(a) High Quality Landmark (b) Low Quality Landmark

Figure 3.4: Plots showing the co-efficient of determination(R^2) for 2 different landmarks (a) Landmark with Good fit, $R^2 = 0.778$ (b) Landmark with poor fit, $R^2 = 0.2443$

points remain. We refer to this dataset as the *sanitized* (or cleaned) dataset.

Figure 3.3 plots the performance of different localization algorithms with the sanitized dataset. We can see that M1 outperforms all other algorithms, it achieves a median error of 0.24 m, about half the error reported by state of the art RSS-based localization algorithms. The median error for NLS has also improved from 2.01 m for the scaled dataset to 1.62 m for the sanitized dataset. Note also that the max errors for M1 dropped from 26.87 m to 1.60 m showing a 94% improvement.

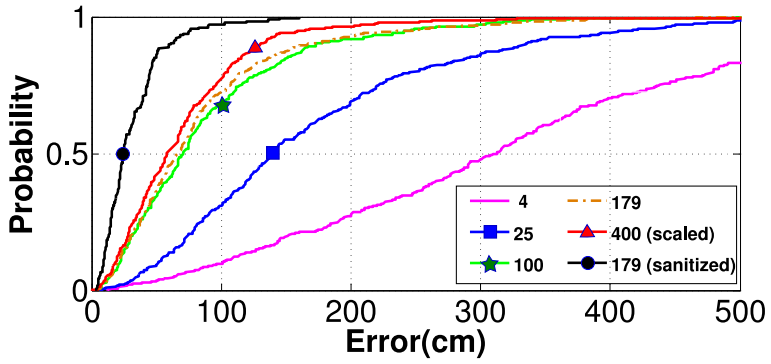


Figure 3.5: Effect of Scaling the Number of APs

Figure 3.5 plots the error CDF for M1 for varying number of landmarks (or access points) —4,25,100,179 and 400. In each scenario, the access points were deployed in a regular, equally spaced fashion. Additionally, we also plot M1's result for the sanitized dataset. While increasing the number of access points significantly reduces the error for M1, the results show diminishing returns. The reduction in error from 100 to 400 landmarks is minor compared to

reduction from 4 to 25. Note also that, the sanitized dataset with 179 landmarks significantly outperforms all results with arbitrary selection of landmarks.

These results suggest that lateration algorithms generally are very sensitive to data from low-quality landmarks that cannot be fitted on a propagation model. For the lateration algorithms, increasing landmark density is less effective than selecting high-quality landmarks. The matching algorithms, however, remain very robust to these data quality issues.

3.4.3 Performance with Synthetic Data

We have shown that after data sanitization, a median localization accuracy of 0.24 m is achievable. To explore possibility of further improved localization, we study the localization performance starting with an ideal noise-less RSSI dataset, which we create synthetically to conform to the well-known path-loss model given in Eq. 4.2. Then, by using empirical observations from our ORBIT experiments, we model two different classes of noise that affect RSSI data and perturb this perfect synthetic dataset according to the models. Finally we compare the performance of the localization algorithms using the perturbed data with the ORBIT experiments to validate our modeling.

The parameters used to create the synthetic dataset are given in Table 3.2. These were obtained from a detailed channel measurement study in the ORBIT room [53], which determined the path gain at reference distance P_0, and the path loss exponent γ. To facilitate comparisons with the best case ORBIT experiments, our synthetic dataset consists of 179 landmarks with RSSI information from 179 different locations that fall 30cm away from each landmark. This replicates the transmitter locations used in the actual ORBIT experiments for the sanitized dataset.

Parameter	**Value**
Path gain P_0 @ 2.4 GHz @ 1 meter	-42.934 dB
Path Loss Exponent (γ)	1.96
Transmit Power	10 dB
Antenna Gain	2 dB
Cable Losses	1 dB

Table 3.2: Summary of the Parameters Used in Synthetic Data Generation

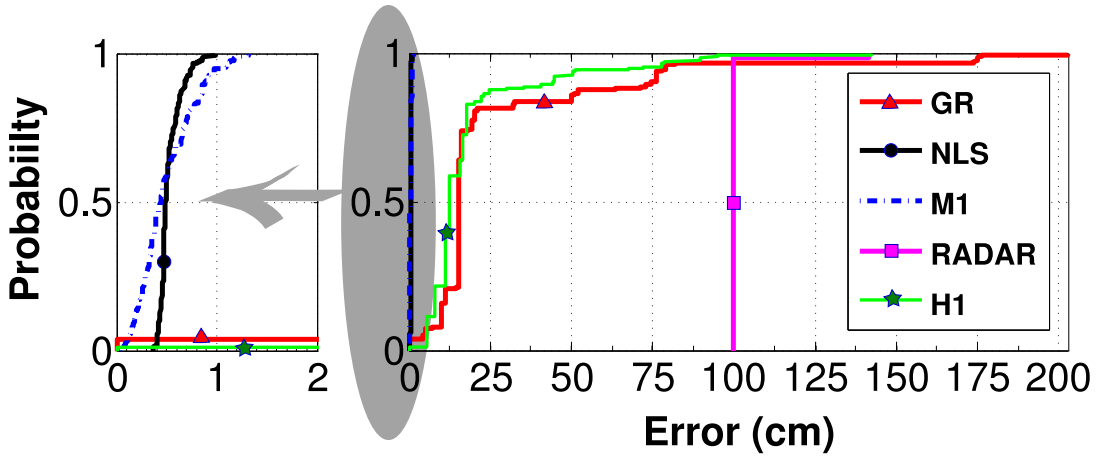
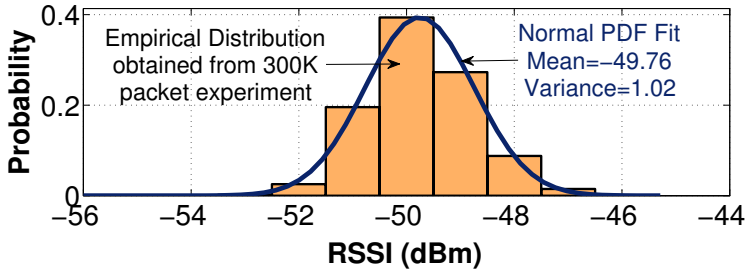


Figure 3.6: Localization performance with synthetically generated noise-less data for 179 landmarks. The Fig in the left plots the Zoomed-in error CDF to show M1 and NLS performance and the one on right plots the Error CDF for all five algorithms

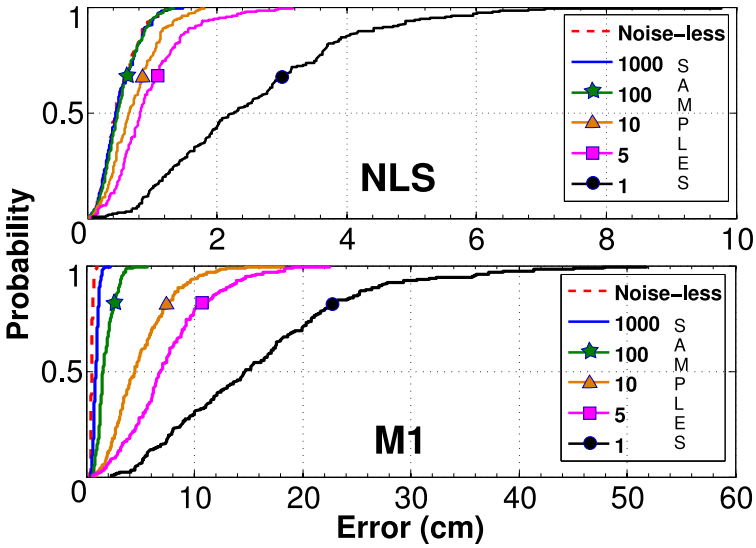
Noise-less Performance: Figure 3.6 presents localization performance for five different algorithms using the noise-less dataset. Lateration algorithms M1 and NLS perform very well—both result in sub-centimeter accuracy for 99% of the time, and strictly below 1.5 cm all the time. Classification algorithms RADAR and GR are limited with the discrete number of fingerprints (classes) to which a given testing point can be associated. Due to the leave-one-out method of testing, RADAR can only match a testing point to the nearest possible landmark, which is 3-feet away in the ORBIT grid setting. Consequently we observe that RADAR has 99% of its error accumulated exactly at 3-feet (about 91 cms). The CDF for GR, which works with 2 inch (5.08 cm) grid-sizes, shows a step-like behavior and achieves a median error of 15 cm (worth almost 3 grid points). Similarly, the tile-discretization effect inherent to H1 results in a median error of 12 cm with this ideal noise-less dataset.

The performance discrepancy between Figures 3.3 and 3.6 indicate that the input dataset in our ORBIT experiments contains significant noise on RSSI observations.

RSSI Noise Analysis: To investigate this discrepancy, we measured the distribution and stability of RSSI readings on a single transmitter-receiver pair. Without environmental mobility and a time-invariant channel, the RSSI observations from an ideal pair of transmitter-receiver should be constant over the time. Figure 3.7(a) shows the RSSI distribution of 300.000 packets



(a) Single Link RSSI Distribution



(b) 179 Landmark Localization Performance with Type I Noise

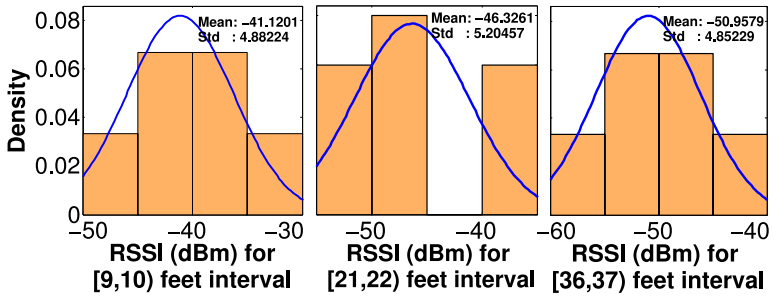
Figure 3.7: (a) Empirical RSSI distribution from single-link 300K packet experiment and the normal distribution fit for Type I Noise model (b) Effect of the number of RSSI samples used for averaging on Type I Noise perturbed synthetic data for Nonlinear LS and Bayesian Network Algorithms

over a given ORBIT link.[1] Clearly, variations of 1–2 dB exist and we categorize this as Type I noise. Detailed discussions on the potential causes of the noise observed on RSSI are deferred to Section 3.5.

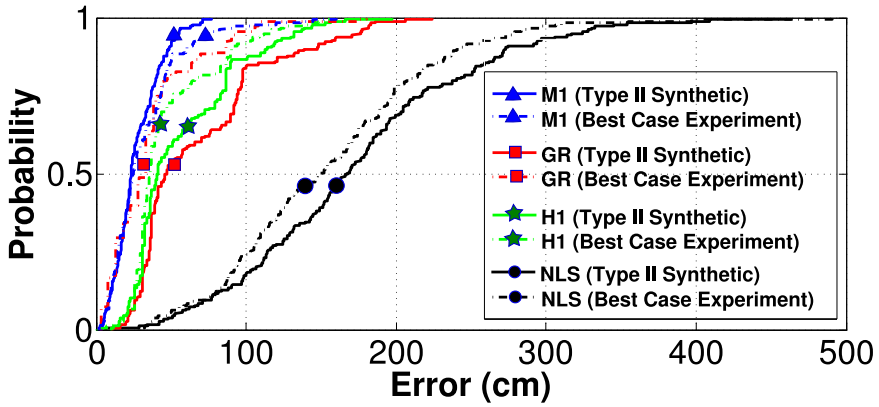
Type I Noise Case: To understand how this variance affects localization, we have perturbed our noise-less synthetic dataset according to the variance of the normal distribution fit on the observed noise of 1.02 db as shown in Figure 3.7(a). The RSS for the synthetic data follows $\mathcal{N}(\mu(\theta), 1.02)$ where $\mu(\theta)$ is the mean received power which is $P_0 - 10\gamma \log_{10}(d_n/d_0)$ as given

[1]Note that these experiments were conducted remotely 4AM in the morning with no human presence and no 802.11 interference. Our tests with other combinations of off-the-shelf cards (i.e., Atheros 5212 and Intel ProWireless 2945) exhibited the same behavior.

by Eq 4.2. We tested both M1 and NLS localization algorithms with this perturbed dataset. We observed that using single RSSI measurements resulted in increased median localization error, from sub-centimeters up to of 14.6 cm and 2.2 cm for M1 and NLS respectively, as shown in Figure 3.7(b). RSSI averaging, however, reduced this effect—the mean of 1000 RSSI observations removed nearly all localization error due to this noise. Thus, the Type I noise do not account for the observed discrepancies, since the high-density experiments were conducted with the mean of 1000 packet RSS readings.



(a) Three example PDF fits for Landmark #5



(b) Localization Performance with TypeII Noise

Figure 3.8: (a) Example empirical RSSI distributions obtained per landmark at each foot distance separation for Type II Noise (b) Performance of localization under Type II Noise

Type II Noise Case: Next we model the noise observed in Fig. 3.4, by dividing TX-RX distance into 1 ft buckets for each receiver and fitting a normal distribution on the data that fall in each bucket, as illustrated in Figure 3.8(a). We categorize this noise as Type II . Using these standard deviations($\hat{\sigma}$) obtained from the PDFs, we create another synthetic dataset whose RSS follows $\mathcal{N}(\mu(\theta), \hat{\sigma}^2)$ and again evaluate the localization performance. Results are depicted in Figure 3.8(b). Note that these results match the experimental results well, the discrepancy is

less than 20 cm for all algorithms.

These results suggest that the noise that limits localization performance is not due to short term measurement noise on individual nodes, but rather due to variations across nodes and locations. We discuss more on RSSI noise in Section 3.5.

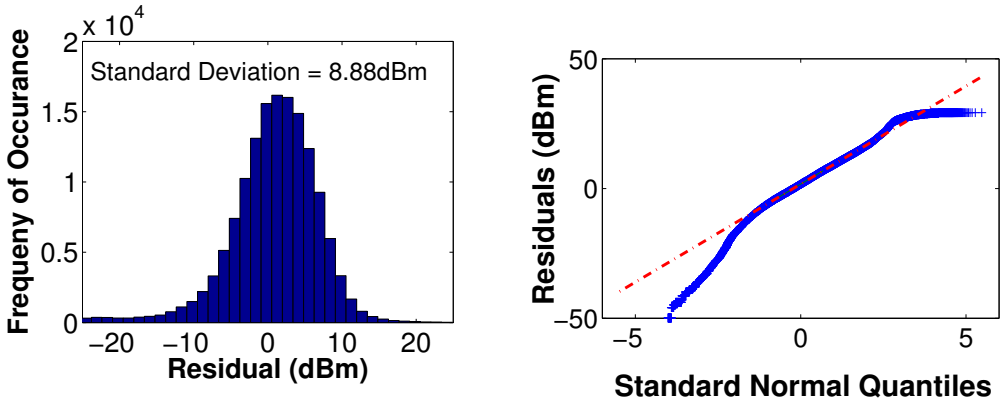
3.4.4 Comparisons with Cramér Rao Lower Bound

Following the Gaussian distribution assumption in Section 3.2.3, we have calculated the standard deviation (*stddev*) of the averaged RSS sample residuals (σ_{RSS}) from our experimental dataset of 400 nodes to be 8.880, as illustrated in Figure 3.9(a). Also, the path-loss exponent γ in the ORBIT room was previously measured to be 1.701, using precise measurement equipment [53] at 5.1GHz UNII band. Using this σ_{RSS}/γ ratio of 5.220 together with the 179 (i.e., sanitized) and 369 (i.e., scaled) landmark topologies as inputs, we calculated the CRB for each unknown transmitter position with the help of a Matlab script. The median value of the stddevs obtained from this CRB calculation is reported in Table 3.1 (in meters) together with stddevs of errors from the localization algorithms we have evaluated.

In both the scaled and sanitized cases, the CRB does not provide a lower bound on the variance of the localization error, except for the NLS algorithm (both cases) and M1 algorithm (only for scaled case). It is important to note that the CRB is only a bound on the variance of the localization error, but not a bound on the mean of the error. Therefore the CRB should be used as a benchmark for the precision of various unbiased localization algorithms but not the accuracy of them.

We identify two reasons why the CRB provided little value in comparing the precision of localization algorithms we tested.

RSS Distribution: The CRB (Eq.3.6) assumes that the RSS sample residuals come from a zero mean Gaussian distribution with a stddev of σ_{RSS}. To verify if this assumption holds, we examined averaged RSS samples shown in Figure 3.9(a) with a quantile-quantile (Q-Q) plot given in Figure 3.9(b). We observe that RSS data fit the normality assumption only between -2 and +2 quantiles. Also, a chi-square test for the normality of the data indicates that the normality hypothesis can not be accepted with 95% confidence. It is likely that our RSS samples come from a more complex composite distribution, similar to the observation in [70]. Since



(a) RSS Residual Distribution

(b) Q-Q Plot for Residuals

Figure 3.9: Investigating Gaussian distribution assumption of RSS samples from 400 node dataset

the Gaussian distribution assumption does not strictly hold, the CRB can not be expected to provide a strict bound for the localization error variance. In theory [74], the RSS residual in the ORBIT testbed environment which has a strong LOS component with minimal shadowing should be characterized using the Ricean distribution instead of Gaussian.

Estimator Bias: The CRB in the form of Eq.3.6 is not applicable to biased estimators. It is quite possible that the majority of localization algorithms are biased. In fact, non-linear least squares is known to generally be a biased estimator. Also, RADAR intuitively appears biased due to the limited number of training points that positions are matched to. The median error for biased estimators can be lower than the bound for an unbiased estimator. Deriving a general form of the CRB for a biased estimator would require knowledge on the gradient of the estimator bias [42].

3.5 Discussion

Our results leave unanswered questions with regards to the lower bounds of localization performance using RSS. In this section, we discuss how resources, node quality and algorithm choice impact the lower bounds of localization performance.

First, high accuracy requires a large number of observation points, which is in agreement with prior work [95]. However, motivation for many of the prior works has been building a localization system using only minimal additional infrastructure. We have also shown that

EQUIPMENT	Calculation	How the RSSI is measured by the card (i.e., exact sampling from the packet training sequence, before or after AGC elements etc.). How interference and noise affect RSSI calculation. Details of the particular algorithm running in DSP.
	Quantization	The way RSSI is quantized (i.e., 0-31, 0-63, 0-100). Averaging used. Non-linearity of rcvd power vs. reported RSSI
	Faults	Leaking, or improperly terminated RF circuitry. Broken antenna cable or connectors.
	Manufacturing/Design	Differences in the manufacturing process. Variance of the quality of the circuit components used. Cheap transceiver design with fluctuating TX power and receiver path gains.
ENVIR.	Thermal Noise	Observed in the receiver electronics dependent on the ambient temperature. Also other forms of cyclo-stationary noises.
	Shadow Fading	Caused by the blocking of direct, reflected, diffracted, and scattered signal copies from the transmitter.
	Multipath Fading	Caused by multiple copies of the received signal through (a possible) line-of-sight component, and its reflections, diffractions, scattering, each delayed wrt power-delay profile of the physical environment around TX-RX pair.

Table 3.3: Sources of noise for RSSI relevant to our experiments

the additional infrastructure can leverage the RSS measurements on existing communication waveforms by re-using the enormous investment in commodity chipsets.

Our results show that high quality RSS measurement is critical to localization performance, and that the measurement and reporting variances across devices limit the accuracy. Although the minute-scale averages of RSS observations are found to be stable (Figure 3.7), significant variance still exists when signal-to-distance fits are considered (Figure 3.8(a)). The exact breakdown for the causes of this distribution remains unknown. Table 3.3 provides a non-exhaustive list of the sources of noise that might have potentially led to non-ideal RSSI observations for our experiments. Note that the list omits important items like mobility and external interference as they were not existent in our controlled experiment. *Calculation* of RSSI from a received IEEE 802.11 packet is only outlined by the standard [48] and implementation details for any given wireless card remain the manufacturers' intellectual property. Nonetheless in our experiments, we use 369 identical m-PCI Atheros 5212 IEEE 802.11a/b/g cards manufactured at close proximity in time, thus RSSI observations likely come from identical calculation algorithms. Also, in our experiments, *faults* are easily detectable as we have close proximity RSSI observations for every landmark from which outliers could be eliminated. Fading, as demonstrated by precise measurements in the ORBIT room [53], is time-invariant in the absence of environmental mobility, thus the multipath profile of the channel in the room is static. In our experiments, fading is visible not because it varies the received signal power for a given link over the time,

but because the RSSI is observed from a static transmitter at 179 discrete locations in the room, each capturing a different fading profile. This makes several of the possible sources unlikely, pinpointing the exact reasons remains an open problem.

A last open issue is that we believe, there is still room for better algorithmic methods to extract localization performance from the traditional approach of finding the best fit that minimizes the residuals. The M1 algorithm is a first step in exploiting such prior information in a manner that goes beyond traditional classifiers, but our work raises the question if additional information could still be extracted by clever algorithms without resorting to classifiers.

3.6 Summary of Key Ideas

To summarize, the key contributions of this chapter are:

- We investigated the lower bound of RSS based localization algorithms through a dense, high-precision wireless testbed. We found that high average accuracies, on the order of 0.2 m, are possible using commodity hardware in our configuration. We also showed that the maximum error can be reduced to about 1.6 m, which is also an encouraging result. However, we have to note that such high accuracies were produced in a very high density environment with 179 landmarks in a $20m \times 20m$ area and this is not typical in actual indoors deployments.

- Our results show that the precision of a number of algorithms exceed the theoretical lower bounds commonly calculated for localization techniques using Cramér-Rao Bound (CRB) analysis for unbiased estimators. This raises questions about the validity of the assumptions underlying this analysis, particularly with regard to bias and normally distributed measurement errors.

- Our results also demonstrate that the choice of algorithms is important, in that we observed least squares approaches have the worst performance on real data sets. We also found that classification-based algorithms are more robust to poorer quality data than lateration approaches. Our results also point to the possibility of further improvements from increasing the quality of the RSS observations, raising the node density in real

deployments, or adding algorithmic enhancements.

- Finally, our trace driven simulations along with the experimental observations show that the main cause of localization algorithms indoors is the inability of the algorithms to model the dynamic multi-path environment. There are several ray-tracing algorithms proposed in the literature to model multipath indoors but none has been shown to capture the dynamics of the environment. This leads us to a conclusion that localization errors cannot be completely removed in an indoor environment and that using localization results for deriving other location-related properties may not always be the optimal means to deriving them. In the next few chapters, we look at the possibility of using the time-series of RSS for deriving location-related properties.

Chapter 4

DECODE: DEtecting CO-Moving Wireless DEvices using Signal Strength

4.1 Introduction

Many location-aware applications benefit from an accurate estimate of the underlying physical properties of wireless devices. One such physical property is *co-movement*, which describes whether a set of wireless transmitters are moving together on a common path. Co-movement information could be used to infer containment relationships, indicating for example that two devices are owned and carried by the same person, or that several tagged objects are placed on the same pallet. It could also be used to infer social relationships if the transmitters are carried by different persons or for optimizing localization system performance.

While it is straightforward to derive co-movement relationship from position coordinates and trajectories generated by a localization system, our study in Chapter 3 suggests that sufficiently accurate and precise data is not always available. Indeed, our evaluation of a bayesian WiFi localization system (M1) [32] shows that the location estimation errors lead to bias and variance in the Euclidean distance between two co-moving transmitters, making detection of co-movement difficult. Global Positioning System (GPS) accuracy is also frequently degraded in urban canyons [25], and even if signals are available, GPS receivers are not commonly used in portable devices due to their high energy consumption. For indoor environments, localization systems require the presence of multiple landmarks or receivers, which adds infrastructure cost. Coarse co-movement information can also be obtained from connectivity through short-range radios [31]. This, however, requires tracking software to be installed on all mobile devices, it can not easily be inferred through infrastructure solutions alone.

4.1.1 Overview of DECODE

In this chapter, we propose the DECODE technique which detects co-movement through correlated signal variations over time rather than directly measuring the signal difference between two transmitters. The technique can either work in signal space, using Received Signal Strength (RSS) indicator values, or in position space, using location coordinates derived from the signals.

DECODE can exploit commonalities in signal power variations, because certain fading patterns of co-moving transmitters are similar. The wireless communications literature [77] distinguishes shadow and multi-path fading effects that attenuate or amplify a signal in addition to the path loss due to communication distance. Shadow fading refers to obstacles in the environment that attenuate the transmitted signal, when it travels through the object. The magnitude of this effect depends on the material and width of the object (e.g., about 10dB attenuation was observed when an outside antenna was moved inside of a vehicle [46]). Multi-path fading describes the effect that objects in the environment reflect and scatter the transmitted signal, so that the signal often arrives at the receiver along multiple paths. The signal components constructively or destructively interfere, leading to fast changes in received signal strength if the position of the receiver changes by merely one-half the wavelength of the communication frequency used (about 59mm for ISM Band 2.4GHz [65] can result in signal strength changes exceeding 20dB). As transmitters or receivers move, the time varying attenuation due to these effects will be unique for each path in space. Two receivers co-moving with a separation of less than one-half wavelength can be trivially detected because they will experience nearly identical signal power curves, assuming same transmission power and antennas). For high communication frequencies in the unlicensed band, however, only few transmitters will be sufficiently close to allow such straightforward detection. Thus co-movement detection has to allow significant difference in signals due to multi-path fading.

Thus, this work presents the DECODE technique, which detects co-moving transmitters by correlated signal changes introduced by the shadow fading component in measured signals. While the multi-path component of the signal differs, transmitters separated less than a few

meters will often still observe commonalities in shadow fading since larger objects in the environment tend to block all direct signal paths to the co-located transmitters. To isolate the shadow fading component, DECODE first extracts periods of high signal variance from the observed signal strength traces over time. When operating directly in signal space, DECODE removes high-frequency multi-path components of the signal and calculates a correlation coefficient over the filtered signal. A high correlation coefficient indicates co-movement of the transmitters. When operating in location space, it calculates correlation over a time-series trace of coordinates reported by a localization algorithm. Localization algorithms typically average signals over time and thus also largely filter out multi-path effects. Shadow fading can manifest itself as errors in the localization output, which DECODE can exploit. One key advantage of applying DECODE in signal space is that, in typical indoor or urban outdoor environments where shadow fading exists [49], DECODE requires only one receiver to detect co-movement, while localization systems require signal measurements from multiple receivers.

4.1.2 Uses of Co-Movement Information

Many applications can benefit from co-movement information. Some of the important ones are:

- **Mapping Devices to Persons:** Many location-aware application such as Friend finders are tracking devices as a proxy to infer the position of the device owner. The proliferation of mobile devices and distinct radio technologies on each mobile device make monitoring this mapping of devices to their owners increasingly cumbersome. For example, as a mobile device moves from an outdoor to an in-building location, it may be tracked by a variety of different technologies each using a different device identifier (usually a radio MAC address). By monitoring co-movement of different transmitters a localization system may be able to infer which devices belong to the same owner, or which addresses represent the same device.

- **Social Network Mining:** Recent work [31] has sought to infer social relationships from mobile device connectivity patterns. Applications for such techniques include automatically determining access control policies and viral marketing. Current techniques monitor Bluetooth advertisement messages to determine when and how long devices from different owners meet. This requires software on mobile devices. The co-movement techniques could also extract this information through external observations (from a communications base station).

- **Localization Optimizations:** Knowing that two mobile devices move together helps collaborative positioning mechanisms that provide lower energy consumption or better accuracy. For example, one device could power down its GPS receiver to conserve energy, while the other device's receiver still provides accurate position updates. In challenging environments for localization, position estimates may also be improved through redundancy.

4.2 Motivation for DECODE

If current indoor localization systems can provide sufficiently accurate location coordinates, one would detect co-movement based on the distance between the two transmitters remaining below a threshold of a few meters. To verify this intuition, we conduct an experiment in an office environment with coordinates reported by a WiFi localization system using the RSS bayesian localizer M1 [32]. M1 is a lateration-based bayesian algorithm which encodes the relationship between the RSS and 2-dimensional cartesian location coordinates using a simple log-distance propagation model. Using a training set (a vector of RSS for different known (x, y) locations), M1 determines the propagation parameters for each of the receivers. It then derives the joint probability density of (x, y) as a function of the observed RSS for the point to be localized and uses the mean of the derived pdf to estimate the unknown location (x, y). M1 has been shown to provide qualitatively comparable accuracy to current state-of-the-art WiFi localization algorithms [32].

In our experimental setup, two IEEE 802.11g (WiFi) transmitters, send 10 packets per second on the same channel, while moving together with about six inch separation within the

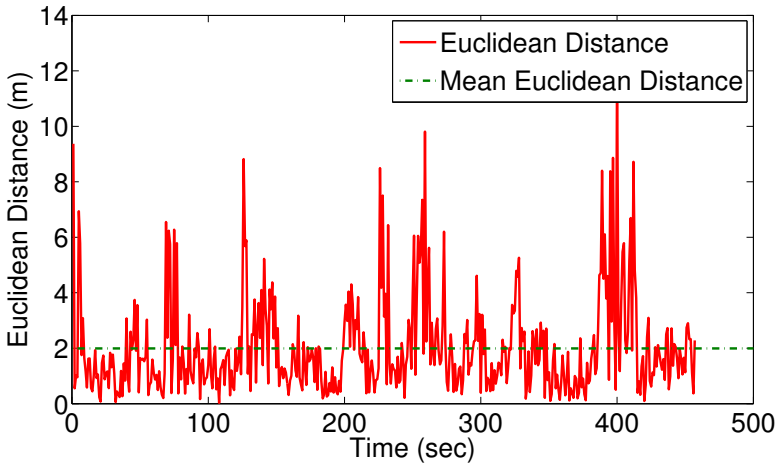


Figure 4.1: Euclidean distance between the localized (X, Y) positions for a co-moving transmitter pair

office space at a constant speed of 1m/sec. Four receivers recorded the observed Received Signal Strength (RSS) from this transmitter pair. We localize these co-moving pair of transmitters in the $85m \times 50m$ cubicle office environment using M1 every second, based on the average RSS reported over the last second. More details on the testbed setup are provided in Section 4.4.1.

The Figure 4.1 plots the Euclidean distance in geographic space between the localized points for the pair of co-moving devices over time. We can see that the distance varies between 0.3m and 12m with a mean distance of 2m. This high variance in the Euclidean distance can be attributed to the following causes.

1. Typical RSS based localization algorithms exhibit a relatively high median localization error of 3m even under no mobility. This localization inaccuracy can increase with mobility thereby resulting in high distance variance.

2. The Localization algorithm estimates X and Y for every transmitter independently before the Euclidean distance metric combines the estimated X and Y from each transmitter. It is possible that errors add up temporarily. It is also possible that a bias in the estimated values for one of the parameters could result in continuously high Euclidean distance estimates.

These high distance errors suggest that the distance threshold detection approach cannot

accurately determine co-moving transmitters. This further motivates the DECODE technique, which we will describe next.

4.3 Decode System Design

The environment in which wireless communication takes place affects the received signal power (or signal-to-noise ratio). The key idea underlying the DECODE technique is exploiting shadow fading, signal attenuation due to objects blocking the path of communication. Two transmitters in close proximity will be similarly affected by surrounding buildings, furniture, or passing people. Therefore, the observed signal power from these transmitters should be correlated. This similarity in signal strength in turn should also translate to correlations in localization errors.

DECODE captures these similarities by calculating the correlation coefficient over a time-series trace of signal strength or location coordinate values. The correlation coefficient measures the strength of a linear relationship between two random variables. Thus the correlation coefficient captures similarities in the changes of two values, even if the absolute values are different. DECODE uses the Pearson's product moment correlation coefficient [12], a preferred method for quantitative measures such as the RSSI traces used. For comparison, we also evaluated another measure of correlation, Spearman's Rank correlation coefficient [62]. Unless otherwise mentioned, correlation coefficient will refer to Pearson's product moment correlation coefficient r_{xy} in the remainder of this work. For n samples each from two random variables X and Y, it is defined as

$$r_{xy} = \frac{\sum x_i y_i - n\overline{x}\,\overline{y}}{(n-1)S_x S_y} \tag{4.1}$$

where S_x and S_y are the sample standard deviations. The correlation coefficient lies in the interval $[-1, 1]$, where 0 indicates no correlation, +1 indicates maximum positive correlation, and -1 indicates maximum negative correlation. We empirically determined a correlation coefficient threshold of 0.6 (see section 4.4.4), values that exceed this threshold indicate co-movement.

Received signal strength, however, also significantly varies due to multi-path fading. It

can introduce received signal strength changes of more than 20dB between locations separated only by half the wavelength of the carrier frequency, if no line-of-sight path to the transmitter is available. These variations render the similarities due to shadow fading difficult to detect. To address this challenge, DECODE calculates a moving average over signals, which acts as a low-pass filter to reduce or remove multi-path effects.

Movement also helps detection of shadow fading similarities, because co-moving transmitters will experience received signal strength changes due to shadowing at similar points in time (e.g., two co-moving transmitters would pass a building corner at the same time). Intuitively, higher speed of the transmitters will increase the frequency of these changes and thus facilitate co-movement detection. Therefore, DECODE will focus on periods of high signal variance, which typically correspond to movement.

Figure 4.2 illustrates the system design and key processing steps of the DECODE system, which can use received signal strength or location-coordinates for estimation. Both approaches share a number of common data collection and preprocessing steps.

In both cases, the receiver measures the received signal strength for packets emitted from transmitters. It reports a *transmitter identifier*, *signal strength* and a *reception timestamp* for each observation to the DECODE processing unit, usually over an existing wired network infrastructure. In our prototype, we have implemented DECODE by monitoring the RSSI indicators reported for each packet reception by the receiver. RSSI has been shown to be a good indicator of channel quality [86], hence it should provide adequate information about fading patterns. RSSI is also available across all wireless technologies, which allows measuring co-movement across different transmitters. For each transmitter, DECODE first performs time alignment and interpolation to facilitate later processing in the face of missing samples. It then extracts periods of high signal variance, which are likely to correspond to movement of transmitters.

This is followed by RSS- or location-specific processing steps. Finally, correlation coefficients are calculated for each transmitter pair and correlation values exceeding a specified threshold indicate co-movement of a transmitter pair.

In the following subsections, we give details of the common, RSS-specific, and location-specific components of DECODE.

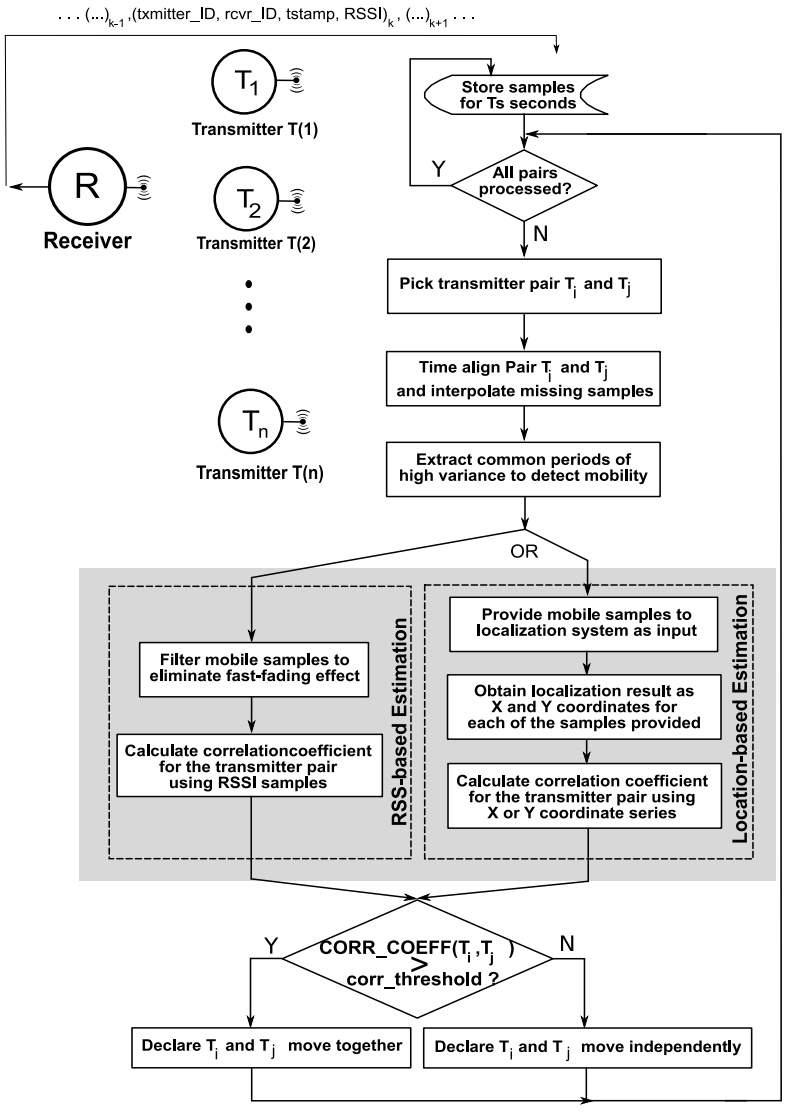


Figure 4.2: System diagram and data flow

4.3.1 Common Components

The common preprocessing steps include time alignment and extraction of high variance periods.

Time alignment: The following co-movement detection seeks to compare RSSI values observed at the same time from different transmitters. The packets originating from transmitters attached to different devices may not be synchronized in time. Even if one attempts to synchronize transmitters attached to the same device, the inherent channel access delays will cause packets from these different transmitters to arrive at the receiver on slightly different times.

Depending on wireless channel conditions, packets are also lost due to collisions or path loss. Thus, the time alignment step synchronizes the samples received from two transmitters. Given the packet traces for two transmitters, our implementation matches every packet from the first transmitter with the last prior packet transmission from the second transmitter. If a sample is missing from the second transmitter, this procedure replaces the missing sample with the last observed sample from the second transmitter.

Extracting high variance periods: Recall that DECODE focuses on periods of mobility because during these periods it can observe correlated signal changes due to shadow fading, and during these periods it can filter out multi-path fading. Several techniques have been proposed to detect mobility [55, 63, 71, 85]. Of these, we choose the straightforward signal-strength variance threshold-detection technique. DECODE divides the RSSI traces into blocks. It then extracts and concatenates all blocks where the variance exceeds the specified threshold. We empirically determined the variance threshold to be three (see 4.4.6 for further discussion) and a suitable block size of five seconds for variance calculation.

4.3.2 RSS-Estimation Components

If DECODE operates using RSS data, this is followed by filtering out multipath fading and computing correlation over RSS values.

Filtering out multi-path fading: While fading is common in communication channel, the fast fading component where the signal varies in amplitude and phase over short periods of times does not contain useful information about the shadowing profile of the environment. The variance due to fast fading should thus be removed from the RSSI traces to allow calculation of correlation primarily over shadow fading components. DECODE uses a moving window averaging process with a window size of 10 packets (1sec). Figure 4.3 shows an example of this filtering effect. Before filtering the received RSSI values vary by about 10 dB on timescales of less than 100 ms. After processing, only slow variations remain, which are expected from shadow fading.

Co-movement detection: The final step involves calculating correlation co-efficient on the processed signal strength values from the transmitter pair under consideration. If the resulting

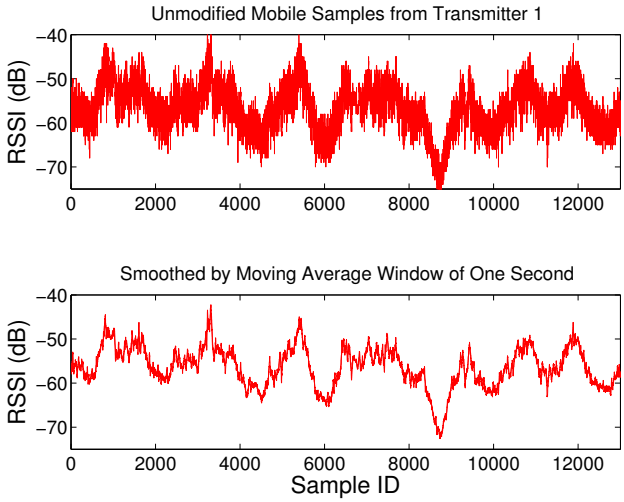


Figure 4.3: Smoothing data to remove fast-fading.

correlation co-efficient exceeds a certain threshold, we classify the transmitter pair to be co-moving. We give details on determining this threshold in Section 5.6.

4.3.3 Location-Estimation Components

The location based estimation approach calculates the same correlation metric over time-series location coordinate data, but it requires data from several receivers to be available and a calibrated localization system. Our localization system relied on an existing signal map of the building, which discretizes spaces and contains an observed signal strength vector (each value corresponding to a different receiver) for each known locations.

RSS Fingerprint Generation: The input to the location system is a fingerprint, an $R \times T_s$ matrix containing RSS values, where R refers to the number of receivers (four in our setup) and T_s to a time window in seconds. To generate these fingerprints, receivers report the *transmitter identifier*, *signal strength* and a *reception timestamp* for every transmitted frame to DECODE. After generating a time aligned sample for the transmitter pairs at each of the receiver and extracting the high variance periods, the resulting RSS samples for each transmitter-receiver pair are averaged over one second intervals and entered into the fingerprint matrix (one matrix per transmitter). If the interval contains no observations for a specific transmitter-receiver pair, the fingerprint generator fills in a localization algorithm-specific default value of -99.

Localization: We use a bayesian solver [32] called M1 to perform localization. M1 is initially provided with a signal map (or training set) containing measurements from 88 different locations in 2D space within the building we carried out the experiment. M1 then transforms each fingerprint matrix into a $2 \times T_s$ matrix of cartesian location coordinates over time, one location estimate per second.

Co-Movement detection: The final step involves detecting co-movement from the (X,Y) estimations at every second for different transmitters. To verify whether a pair of transmitters move together, we estimate their similarity in X or Y coordinates using correlation co-efficient. If the correlation co-efficient for X or Y is over a certain threshold, we declare the transmitter pairs to be moving together. While it may be possible to combine the inference about the correlation in X and the correlation in Y, we do not address this in this work.

4.4 Experimental Evaluation

In this section, we begin with a description of our wireless testbed followed by an explanation of the experimental methodology. We then present our results that highlights DECODE's performance in the location space. We conclude this section by presenting the detection of co-moving devices in the location space.

4.4.1 Experimental Methodology

The measured environment is a typical office environment with partitioned cubicle offices. The experiments were performed during normal office hours where one could expect dynamic changes in the environment as a result of the mobility of the people within the office. We set up both IEEE 802.11b and IEEE 802.15.4 receivers within the office space and place them at strategic locations as shown by stars in Figure 4.4. The WiFi receivers(landmarks) in these four locations operated in promiscuous mode in 2.4GHz, ISM Band Channel 1 to capture all the packets in this particular channel. A Tmote Sky mote configured as receiver was attached to each of the landmarks to capture packets originating from Zigbee transmitters. These motes operated in 2.4GHz, ISM Band Channel 16.

We used four IEEE 802.11b cards and four Tmote Sky motes as transmitters where a pair

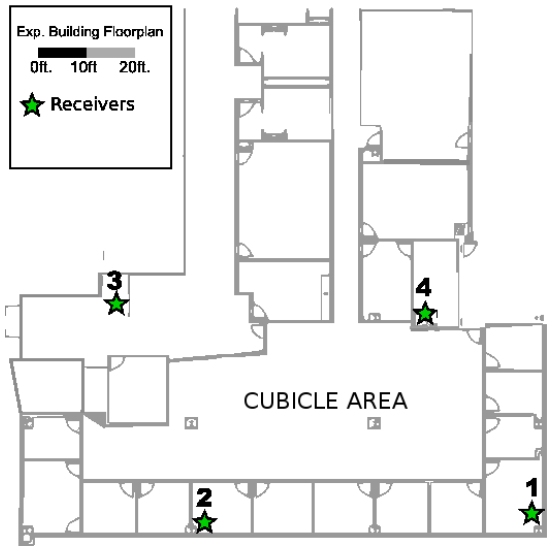


Figure 4.4: Floorplan of the experiment environment and the node placement

Figure 4.5: Nodes and the transmitters used in experiments

of WiFi cards and a pair of motes were placed together in the first laptop and the other pair of WiFi cards and motes were placed together in the second laptop as illustrated in Figure 4.5. The motes were battery powered. The WiFi cards were connected to the configured APs and pinged the AP at the rate of 10packets/sec with a transmit power of 15dBm. And the motes were configured to transmit packets at the rate of 10 packets/sec at 0dBm. We use the ORBIT infrastructure for capturing and logging each IEEE 802.11 and IEEE 802.15.4 packet from these transmitters to be stored in a SQL database. For each packet, we logged the transmitter's MAC address(Mote ID in case of motes), the receiver's MAC address(Mote ID in case of motes), RSSI and the time when the packet was captured. We also recorded the ground truth about which transmitter pairs were moving together along with the speed and the start and the end times of the different static and mobile periods of these transmitters manually. We note that we

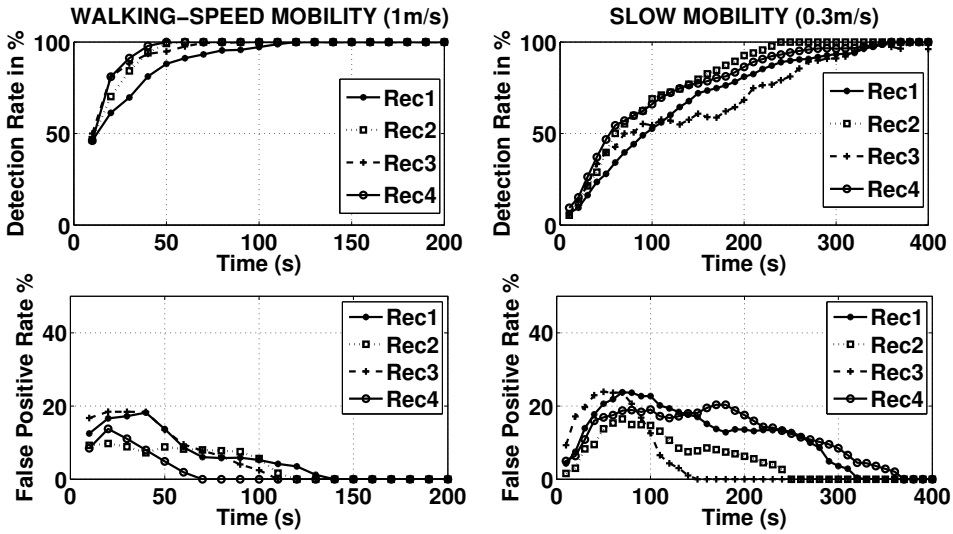


Figure 4.6: IEEE 802.11 network: Effectiveness of DECODE in terms of detection rate and false positive rate. The left side plots are for the Walking-Speed Mobility experiment and the right side plots are for the Slow Mobility experiment.

set up pairwise transmitters in our experiments to show how DECODE works, but our approach could be applied to a set of transmitters that are co-moving.

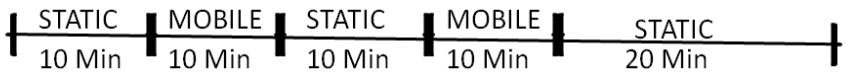


Figure 4.7: The Experimental Procedure

Two experimenters carried one laptop each (that contains two WiFi and two motes) and conducted the experiment. The total experiment lasted for one-hour with alternating static and mobile periods as shown in Figure 4.7. The authors were walking at 0.3m/sec(1ft/sec) for about 20 minutes. We call this experiment period *Slow Mobility*. We chose very slow speeds because this represents the most challenging case. The same experiment was repeated once more where the moving speed of the transmitters was increased from 0.3m/sec to 1m/sec (normal human walking speed). We refer to this second experiment period as *Walking-Speed Mobility*. We refer to these experiment traces as the complete traces.

To analyze the effect of mobility on the results, we then also create mobile-only traces by extracting and concatenating the two 10 minute mobile periods into a 20 minute mobile trace. Using this technique we both create a slow-mobile and a walk-speed mobile trace. We then use

a time-based sliding window of time interval T_s seconds to slice each of the above datasets into overlapping test traces. We vary the time interval T_s from 10 second to 400 seconds in steps of 10 seconds. For, example $T_s = 100s$ would generate 1101 test traces of duration 100s from the 1200s of data.We used these different sliced datasets with different time intervals T_s in our results to report the detection rates and false positives.

4.4.2 Evaluation Metrics

We will evaluate the effectiveness of RSS based DECODE in the following three categories: (1) performance evaluation in terms of the detection rate of co-moving transmitters and the corresponding false positive rate; (2) sensitivity study under different packet sampling rates and various correlation coefficient thresholds; and (3) generality investigation across different correlation methods and wireless networks. Finally, we will study the effects of mobility detection on the performance of DECODE.

4.4.3 Effectiveness of DECODE's RSSI based detector

To evaluate the performance of DECODE, we first examine the detection rate and the false positive rate of determining the co-mobile transmitters. Figure 4.6 depicts the detection rate and the false positive rate as a function of time with respect to each receiver for the IEEE 802.11 network for both Slow Mobility as well as Walking-Speed Mobility experiments.

We compute the correlation coefficient for the samples accumulated over the last T_s seconds and if the computed correlation coefficient is larger than 0.6, the pair of transmitters are declared to be co-mobile. Otherwise, this pair of transmitters are declared to be not moving together. A detailed discussion of the choice of the threshold is presented in Section 4.4.4. In our 20 minutes of mobile trace, we repeat the above procedure for all the generated data subsets of duration T_s seconds. We then estimate detection rate as the percentage of times DECODE correctly reports co-mobility when the pair of transmitters are indeed moving together and False positive rate as the percentage of times DECODE incorrectly reports co-mobility when the transmitters are *not* moving together.

Figure 4.6 shows that in both the Walking-Speed Mobility and Slow Mobility experiments,

DECODE is able to detect all co-moving and non-co-moving pairs over all the data subsets accurately. We can also see that, increasing the observation time T_s improves the co-mobility detection rate while reducing the likelihood of observing spurious matches.

We found that the mobility speed also has an impact on the time required to achieve high detection rate and low false positive rate. In the Walking-Speed Mobility experiment, it takes about 130 seconds to detect all co-moving data subsets. Whereas it takes around 370 seconds to achieve the same in the Slow Mobility experiment. This suggests that, with higher speed, more shadow fading effects can be observed within a shorter duration, leading to improved detection performance.

The results of the Slow Mobility experiment represent detection performance of DECODE under challenging conditions. For the rest of this section, we provide analysis by using the Walking-Speed Mobility experiment since it represents more typical scenarios for devices carried by humans.

4.4.4 Sensitivity to Sampling Rate and Correlation Coefficient Threshold

We now study the sensitivity of our scheme with respect to the different correlation coefficient thresholds and sampling rates, which we define to be the "packet transmission rate per transmitter". To this end, we further process the Walking-Speed mobile trace and extract 0.5,1,5 and 10 packets every second from the trace to generate datasets corresponding to sampling rates of 0.5,1,5, and 10 pps respectively. These four datasets are further sliced into several data subsets with time interval T_s seconds similar to our previous study for estimating detection and false positive rates.

Figure 4.8 presents the detection rate and false positive rate as a function of time for packet sampling rates of 0.5 packets per second (pps), 1 pps, 5 pps, and 10 pps, respectively, observed at receiver-2 (we do not present the results from other receivers as the performance is very similar). The threshold of the correlation coefficient is empirically determined to be 0.6. We found that for the sampling rates of 1 pps, 5 pps, and 10 pps, the time taken to achieve 100% detection rate and 0% false positive rate is similar—about 130 seconds. With the low 0.5 pps the time to reach 100% detection rate increases marginally to 150 seconds. This is encouraging as it

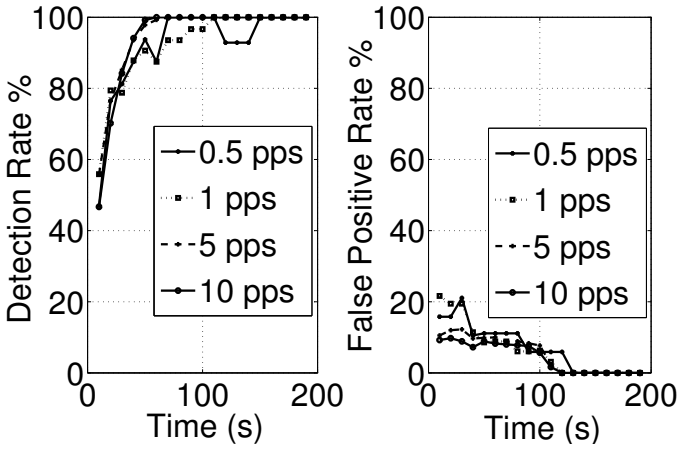


Figure 4.8: IEEE 802.11 network: Sensitivity of DECODE vs. sampling rate.

indicates that DECODE is not very sensitive to sampling rates in the 1 pps range. This insensitivity can be because, a higher sampling rate would not provide additional gain compared to lower sampling rate as long as the lower sampling rate is fast enough to capture the "shadowing events".

This insensitivity to sampling rate also allows reducing the overall channel utilization, in a system design that relies on explicitly transmitted beacons to allow co-movement detection. The transmission overhead would be negligible. For example, assuming a minimum packet length of 29 bytes(28 Bytes of Frame and 1 Byte of Payload), an 802.11b station transmitting one packet per second at 11 Mbits/s PHY rate takes $603.27\mu sec$ [3] which accounts only for 0.06% of channel utilization.

We next analyze the sensitivity of DECODE to the correlation coefficient thresholds τ. Choosing an appropriate threshold will allow our detection scheme to be robust to false detections. Figure 4.9 presents the detection rate and the false positive rate for τ equaling 0.4, 0.5, 0.6, 0.7 and 0.8, respectively. As expected, we observe that the detection rate takes longer to reach 100% as the threshold goes up, while the false positive rate drops to 0% quicker. The threshold $\tau = 0.6$ achieves the best balance with a false positive rate remaining below 10% at all times and the detection rate reaching 100% nearly as fast as the smaller thresholds 0.4 and 0.5. Hence, we chose a threshold of 0.6 for all other experiments.

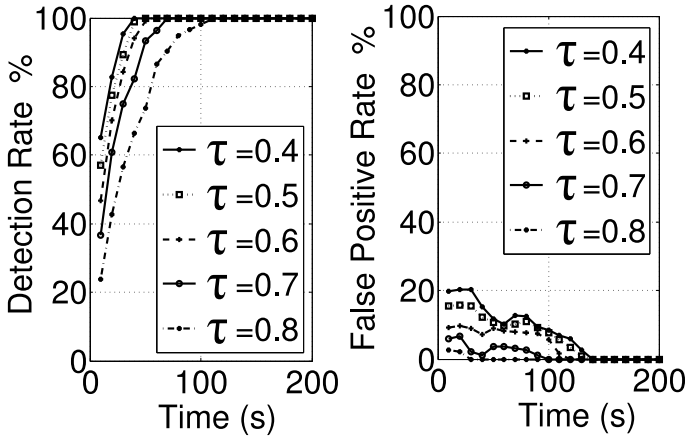
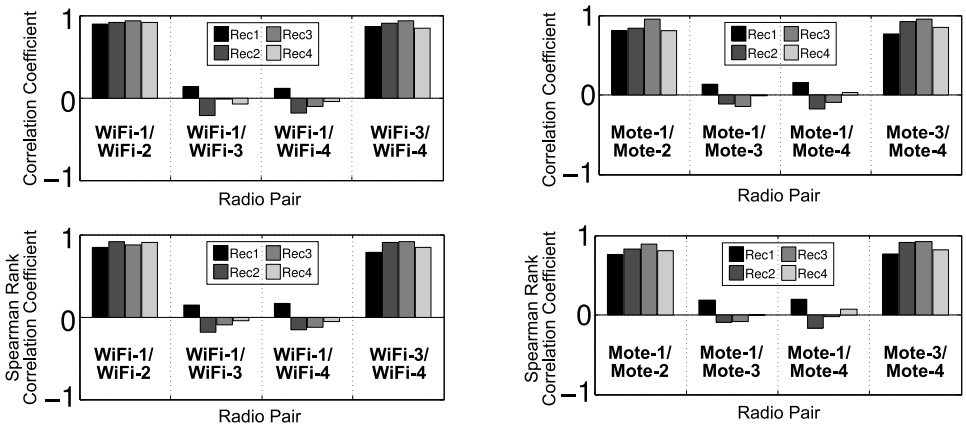


Figure 4.9: Sensitivity of DECODE to Correlation Co-efficient Threshold. We pick a threshold of 0.6 for Co-Movement.



(a) IEEE 802.11 network

(b) IEEE 802.15.4 network

Figure 4.10: Comparison of correlation coefficient methods for WiFi and Mote radio pairs.

4.4.5 Generality of RSSI based DECODE

We now study the generality of DECODE in using different correlation methods to determine co-moving transmitters and its generality across both IEEE 802.11 as well as IEEE 802.15.4 networks.

Different Correlation Methods: We compare our correlation coefficient method (i.e., Pearson's product moment correlation coefficient) with Spearman's rank correlation coefficient in Figure 4.10(a) and 4.10(b) for the IEEE 802.11 network and the IEEE 802.15.4 network respectively. The correlation coefficients are computed for all the co-moving and non-co-moving pairs of transmitters. Note that we refer to the Pearson's product moment correlation coefficient

method as correlation coefficient in the figure.

We observed that both the correlation coefficient methods perform similarly for the co-moving and the non-co-moving pairs of transmitters. For the co-moving pairs, the correlation coefficients from both methods are above 0.6, while for the non-co-moving pairs, both have values of correlation coefficient below 0.2.

Different Wireless Networks: Figure 4.11 presents the results of correlation coefficient calculated across an 802.11 transmitter and an 802.15.4 transmitter.

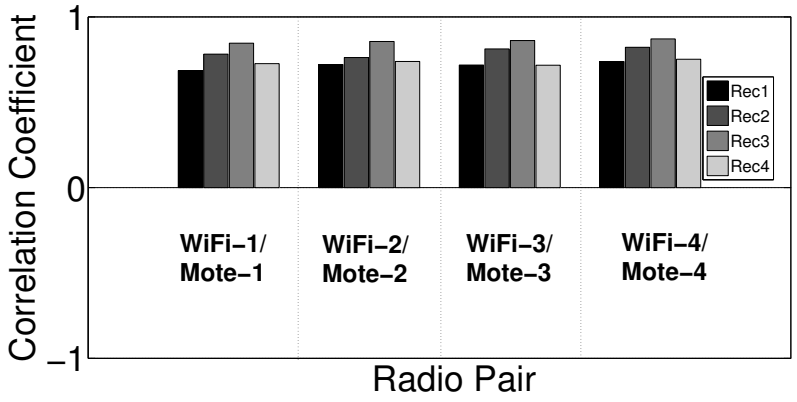


Figure 4.11: Correlation Coefficient for Co-located Mote and Wifi

We found that the correlation coefficients for co-moving pairs for both the 802.11 as well as for the 802.15.4 are consistently high (larger than 0.6) across all receivers. This is because, when there is an obstruction to the Line-of-Sight signal component due to walls and other objects, both the WiFi and the mote transmitters experience similar shadowing effect as they are placed close enough. Though the actual amount of the degradation of signal differs, the relative effects are the same. Since Pearson's correlation coefficient method removes the sample mean from its estimation, similar relative performance is enough to capture co-moving transmitters. This result is strong evidence that our approach is generic across different networks.

4.4.6 Significance of Mobility Detection for DECODE

In this section, we examine how mobility detection impacts the performance of DECODE.

Effects of Mobility Detection: Figure 4.12 plots the correlation coefficient at all 4 receivers

for co-moving transmitters in the Walking-Speed Mobility experiment. The correlation coefficient is computed over the entire duration of the experiment as well as just over the mobile periods.

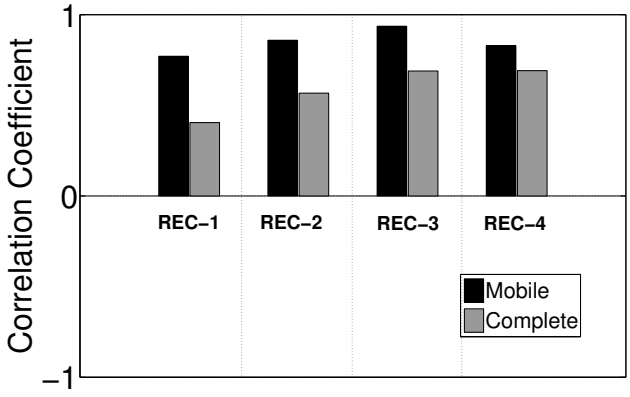


Figure 4.12: 802.11 network: Calculation of the correlation coefficient over the entire experimental period and over the mobile periods only. There is a 20% improvement in the correlation coefficient values when applied over mobile periods only.

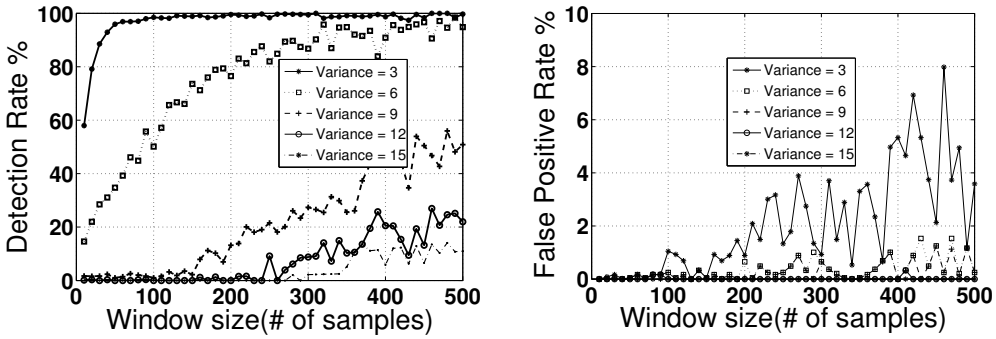


Figure 4.13: Effects of variance threshold and sample window size on mobility detection.

We found that the mobility detection helps increasing the values of the correlation coefficient for co-moving transmitters by an average of 20%. During static periods, the co-moving transmitters do not experience significant changes in shadow fading, but may experience small scale fading effects that differ from one transmitter to the other (if the separation is more than about 6cm ($\lambda/2$) for 2.4GHz). Thus, including static periods in the calculations tends to reduce overall detection performance, particularly if the static periods are long compared to the mobile periods.

These results support our approach of first extracting mobile (high variance) periods.

Thresholds for Variance and Window Size: For mobility detection, there are several metrics available as shown in [84]. However, we found that using a simple metric, variance of RSS, is sufficiently effective. Further, two parameters are important when using the RSS variance to detect mobility: the threshold of variance and the number of RSS samples on which the variance is calculated. Figure 4.13 plots the trade off between the detection rate and the false positive rate for different variance thresholds and different window sizes for the co-moving WiFi transmitter pair.

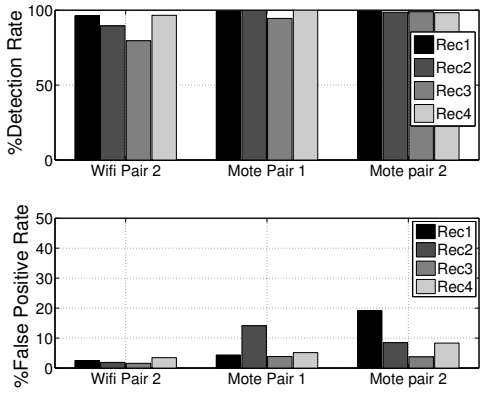


Figure 4.14: Mobility detection of co-moving transmitters under windowsize =50, variance Threshold = 3.

We observed that the variance threshold of three has the highest detection rate with false positives less than 10% for all window sizes. We choose a window size of 50, where the detection rate is over 96% and the false positive is less than 1%. We estimate the correctness of these parameters across all transmitters to check the result consistency. Figure 4.14 plots the detection rate and the false positive rate for mobility detection across the rest of the 6 transmitters including both WiFi and mote transmitters. The results from Figure 4.14 proves that our results are consistent across all the transmitters with high detection rate and less than 10% false positive rate under a window size of 50 and a variance threshold of three.

4.4.7 Co-Movement Detection in Location Space

As pointed out in Section 2.3, the Euclidean distance between the pairs of transmitters is not a very accurate estimator for co-movement detection. In this section, we evaluate the DECODE

correlation estimation applied to individual co-ordinates in location space.

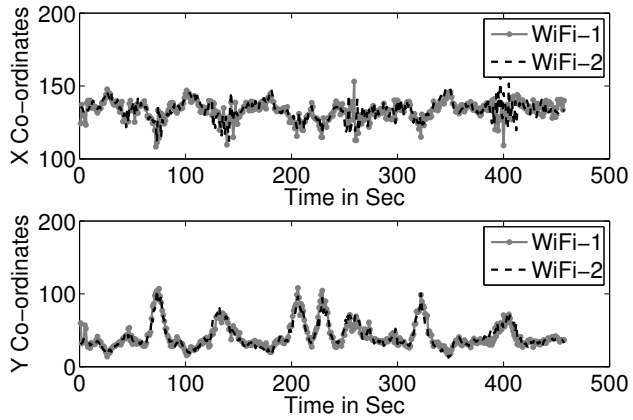


Figure 4.15: Localized X and Y positions for a pair of co-moving wifi devices

Figure 4.15 plots the localized X and Y positions over time for a pair of co-moving WiFi devices that were attached to the laptop 1 and moving at a speed of 1m/sec. We can observe that the X and Y co-ordinates estimated by the localization system for the two co-mobile transmitters are very similar, but some differences exist. These differences can be attributed to the sensitivity of the localization algorithm to small scale fading, which can affect both transmitters differently and resulted in the high variance in Euclidean distance, as was shown in Figure 4.1.

However, by calculating the correlation coefficient over the localized X position and the correlation co-efficient over the localized Y positions, we can achieve similar detection performance to the signal space technique. This is possible because the correlation co-efficient can ignore the absolute values and can capture the relative trend in the way the X and Y co-ordinates vary (e.g, shadow fading is likely to lead to similar localization errors for both transmitters).

We evaluate the total time taken to achieve a 100% detection rate and 0% false positive rate. We define the detection rate as the percentage of times the correlation co-efficient computed for a co-moving pair is above 0.6 and false positive as the percentage of times the correlation co-efficient for a non-co-moving pair is above 0.6. Figure 4.16 plots the detection rate and false positives as a function of time. Note that for simplicity, we have calculated correlation separately for the X and Y coordinates. We can see that it takes nearly 200 and 90 seconds for

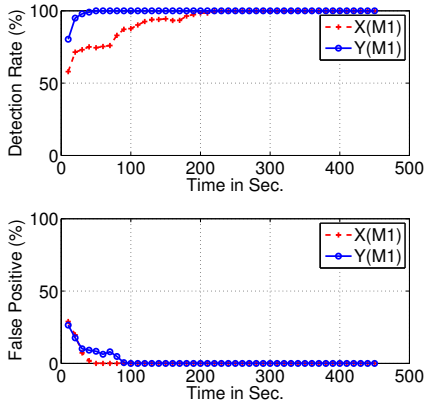


Figure 4.16: Effectiveness of correlation coefficient applied over the localized X and Y locations for co-movement detection

the X and Y co-ordinates respectively to achieve a 100% detection rate with 0% false positive rate. The corresponding time taken by DECODE in signal space was 130 seconds. While these times are comparable, there are several advantages of using signal space DECODE over location space DECODE—we discuss them in the Section 4.6

4.5 Simulation with Different Channel Parameters

After observing encouraging results in the experimental indoor environment, we evaluate now whether these experimental results as presented in section 4.4 are consistent with results from simulation models and whether they can be generalized to indoor and outdoor environments with different propagation parameters. We also analyze the effect of shadow fading on the detection time.

Environment	**PathLoss Exponent**	**De-Correlation Distance (m)**	$\sigma_{CorrelatedShadowing(db)}$	$\sigma_{Noise(db)}$	**Detection Time (sec)**
Indoor-1	2.5	2	2	2.3	108
Indoor-2	2.5	2	4.3	0	81
Outdoor-1	2.8	5	2	2.4	94
Outdoor-2	2.8	5	4.4	0	70

Table 4.1: Summary of the Parameters Used in simulations along with the total detection time

Our simulation methodology involves generating the received power at a receiver from

three transmitters, two of which are moving together on the same path and the third transmitter following a different path. To allow comparison with the experimental results, the path taken by the moving transmitters in the simulation was derived from the experiment paths described in section 4.4.1.

The simulator generates received power levels as follows. From [36], we know that the received power at a receiver from a transmitter can be modeled as

$$P(d) = P_0 - 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + S_\sigma + \delta, \tag{4.2}$$

where d is the distance between the transmitter and the receiver; P_0 is the received power at the reference distance d_0 from the transmitter; γ is the path loss exponent; S_σ represents shadow fading (i.e. correlated shadowing) which follows zero mean and σ standard deviation Gaussian distribution and δ is the random noise.

To simulate correlated shadowing, the S_σ for different positions must satisfy the following exponential constraint [38]:

$$E[S_\sigma(P_i)S_\sigma(P_j)] = \sigma^2 e^{-D_{ij}/D_c}, \tag{4.3}$$

where $S_\sigma(P_i)$ and $S_\sigma(P_j)$ are the shadow fading at location P_i and P_j, respectively. D_{ij} is the distance between the positions P_i and P_j. D_c represents the decorrelation distance, which can range from 1-2m indoors to many tens of meters outdoors. We generate such correlated Gaussian random variables S_σ by multiplying uncorrelated Gaussian random variables with the upper triangular matrix from a cholesky decomposition of the correlation matrix [87]. In our case, the correlation matrix is initialized with the desired correlation values e^{-D_{ij}/D_c} between each transmitter (position) pair.

As shown in Table 4.1, we considered four scenarios with different propagation parameters, two for indoor environments and two for outdoor environments. For the indoor environments, we chose standard deviation of the received power by measuring in our experiment environment. Since this standard deviation combines both correlated shadowing and random noise, we simulate two indoor scenario with different assumptions on the level of shadowing and noise. While Gudmundson's exponential [38] decay model has been proposed for medium to large cellular networks in the outdoor environments, [49] has shown that this exponential model can

be adapted for analyzing the spatial correlation arising from shadowing in the indoor environments. We obtained the other indoor and outdoor parameters including the propagation exponent from other reported measurements [49, 96]. The last column of the table also shows the result, the total time taken for detecting co-movement without false positives. The results show similar detection times across all four scenarios, indicating that DECODE is not very sensitive to propagation parameters. This is encouraging and shows that DECODE can be expected to also work in outdoor environments with typical parameter settings.

While the simulation results show slightly lower detection times, 80-108 seconds compared to 130 seconds in the experiment, the results are on the same order of magnitude. The difference can be attributed to modeling and measurement inaccuracies. We measured the standard deviation in power (4.3db) within the office environment several months after conducting the DECODE experiments. Also, the simulation model assumes that power measurements follow a Gaussian distribution $\mathcal{N}(P_0 - 10\gamma \log_{10}(d_n/d_0), \sigma^2_{RSS})$, which may not be fully accurate.

The indoor results also show that increasing the correlated shadowing reduces the overall detection time from 110 sec to 81 sec. A similar trend can be observed in the outdoor results. This indicates that the presence of correlated shadow fading leads to faster detection and is beneficial for DECODE.

4.6 Discussion

In this section, we discuss the feasibility of detecting transmitters that are static and located within close proximity. We continue the discussion by giving out the advantages of operating in the signal space in comparison to the location space. We finally conclude this section by discussing the impact of missing samples on co-mobility detection.

4.6.1 Feasibility of Detecting Co-Location

The co-movement detection results described so far raise the question whether the DECODE technique can also be used to detect stationary co-located transmitters. Ostensibly, an environment with high surrounding mobility could lead to similarly high signal variance even though the transmitters and receivers are stationary, because the moving objects can temporarily block

Receiver1	Receiver 2	Receiver 3	Receiver 4
0.4630	-0.1140	0.2753	0.4362

Table 4.2: Correlation Coefficient for the time interval t=3100 seconds to t=5100 seconds.

transmission paths, which changes shadow and multipath fading patterns.

To investigate whether human mobility in a cubicle office environment is sufficient to also allow detection of co-located stationary transmitters, we performed an experiment where a pair of mote transmitters were attached to the main doorway within the WINLAB office, which is an area with frequent human traffic (it is located next to a printer and water cooler providing additional traffic).

Table 4.2 shows the correlation co-efficients obtained for the stationary transmitter pairs by each of the receivers over a 2000 seconds interval (the transmitters actually moved when the door was opened, but this occurred only twice in this period). Note that all correlation co-efficients are far below the 0.6, our correlation threshold for co-movement detection. Note also that some of the receivers show correlation coefficient values near zero, which suggests that reducing the detection threshold would not be very effective. Thus, these results show that in a typical office environment, surrounding mobility is unlikely to induce sufficient correlated fading to allow use of the DECODE technique for detecting co-located transmitters (even with the extended 2000s measurement period, compared to the 130s period that was sufficient for co-movement detection as shown previously).

4.6.2 RSSI-Based vs. Location-Based Detection

While accuracy of DECODE in both signal and location-space is similar, applying DECODE in signal-space provides several advantages, particularly if location information is not needed for other applications. However, there are challenges to be addressed before one could assume localization systems are sufficient for the purpsohn2006mobilityoses of co-location detection:

- **Generality:** Most localization systems use the already computed training set to determine the location associated with any fingerprint. However, this approach requires the TX power settings during the training and the testing phase to be same in order to estimate the correct location. With a wide variety of wireless devices in the environment,

this requirement makes localization technique highly sensitive and error prone, while RSS-based co-movement detection is more agnostic to these issues. Also, different radio technologies may need different localization systems with different accuracy limits, making co-location detection for radios belonging to different technologies non-trivial using these systems, while we showed that the RSS-based technique can be used across wireless technologies.

- **Localization Overheads:** Calculating absolute location of a device takes time and requires signal information from multiple points of contact (e.g., three reference points for trilateration), which may not be available at all times.

- **Infrastructure Costs:** Investing in the localization infrastructure, including the equipment costs as well as maintaining signal maps, beacon or landmark (receiver) positions etc., might be costly. The RSS-based co-movement detection techniques only requires a single receiver, in comparison.

4.6.3 Impact of Missing Samples on Co-Mobility Detection

As explained in section 4.3.1, the time alignment step involves replacing the missing samples from a transmitter with its last observed sample. While this step aids in comparison of signal strengths from transmitter pairs, excessive replacement of missing samples could overstate the correlation between transmitter pairs. To this end, we analyze the percentage of times the missing samples have been replaced during this step for a Wifi-Wifi pair at Receiver-1.

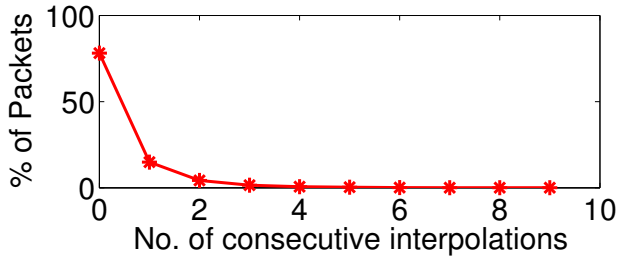


Figure 4.17: Histogram of the percentage of consecutive missing samples from a Wifi transmitter observed at Receiver-1

Figure 4.17 shows that, very few consecutive packet losses occur. 80% of packets have been correctly received. 15% are intermittent single packet losses and only 5% of packets

are consecutive packet losses. We also verified these packet loss rates across all the other transmitter receiver pairs and found similar results. Since DECODE uses a moving window average over 1 second of received packets to remove multipath fading, interpolation of these packet losses has little effect on the correlation results.

4.7 Summary of Key Ideas

To summarize, the key contributions of this chapter are as follows.

- We presented DECODE, a system that detects co-moving wireless devices by exploiting the similarity in shadow fading for the packets transmitted from a set of transmitters.
- We showed that our technique can work in both the signal space and its corresponding location space, but that the signal space approach provides the key advantage that only a single base station is needed.
- We also demonstrated the generality of our technique by detecting co-movement of wireless transmitters having different radios, namely, IEEE 802.11b/g WLAN and IEEE 802.15.4 Mote devices.
- Through real experiments in an indoor office environment, we showed that DECODE can achieve a true positive rate of 100% with 0% false positive given 130 seconds of signal strength data from mobile transmitters. Our experiments also demonstrate that DECODE is insensitive to the sampling rate and a sampling rate as low as 1 packet/sec should be sufficient for perfect detection.
- We found that DECODE's effectiveness is quite sensitive to transmitter mobility. We found that observing RSS variance is sufficient to detect mobility 96% of the times with a false positive rate of less than 1%. Therefore, detecting mobility has a straightforward solution and does not limit the DECODE system.
- Finally, the fact that the time-series of RSS is robust to fades brings in a possibility of extending this idea to deriving other location-related properties. In the next chapter, we look at two such properties, namely, Speed and decelerations.

Chapter 5

Tracking Vehicular Speeds using GSM Signal Strength from Mobile Phones

5.1 Introduction

In this chapter, we consider the problem of estimating average and fine-grained speed variations of a vehicle from cellular handset signals. While average speed estimates can serve as a good indicator of traffic congestion, more fine-grained speed traces could benefit a larger number of transportation applications. For example, fine-grained speed trace could improve estimating and pinpointing traffic congestion, particularly on arterial roads with traffic signals. Since fine-grained speed traces reveal where on a road segment vehicles slow down, it becomes easier to distinguish speed variations due to congestion from slowdowns due to red traffic lights. Fine-grained speed traces also reveal whether traffic is flowing slow but smoothly or in a stop-and-go fashion. It can also show where frequent lane changes occur that cause traffic shock waves. These factors have a significant effect on accident rates and gasoline consumption, and would therefore be important to monitor on a larger scale. Techniques to determine vehicle speed from cell phone signals are particularly useful because they do not incur the high infrastructure costs of traffic cameras or loop detectors embedded into the roadway [22, 23, 44]. While fine-grained speed traces can also be obtained through networked in-vehicle GPS devices, cell phone signals can readily be collected from a much larger number of vehicles. Collecting cell phone signal strength readings at the base station imposes no energy overhead, since the cell phones that are active on call periodically transmit Network Measurement Reports (NMR) containing signal strength readings to the base station.

Our Approach. In this work, we propose two different algorithms to estimate the speed of a moving vehicle, namely, correlation algorithm and Derivative Dynamic Time Warping (DDTW) algorithm. Both these techniques rely on the observation that the received signal strength (RSS)

from a moving cell phone handset in a vehicle remains similar over multiple passes along a given road segment except on the time axis since the traces are either stretched(or compressed) in time depending on the speed with which the vehicle moves. The algorithms then estimate the relative stretch(or compress) factors with respect to a RSS trace collected from a vehicle moving at known speeds to derive the speed of the moving vehicle. The correlation algorithm assumes that the compress(or stretch) factor is uniform over the entire length of the trace whereas the DDTW algorithm accounts for a variability in the compress(or stretch) factor over the length of the trace.

5.2 Motivation

To illustrate the insights underlying the correlation algorithms, consider the signal traces in Figure 5.1. These were obtained from a mobile phone passing three times along the same road segment, twice at the same speed of 25mph and once at 50mph. Note how the two 25 mph traces are very similar. Also, note how the faster trip produced a similar RSS trace except for being compressed by a factor of two. This illustrates how despite signal variations due to fading the signal strength on an average remains relatively stable at the same location, as also previously reported in [18] for stationary handsets. Thus, if we can algorithmically determine that the faster trace is a scaled version of the slower trace by a factor of 2 and we are given the speed of the slower trace, we can calculate the speed of the faster trace. The correlation algorithm makes use of this observation to determine the speed of a moving vehicle.

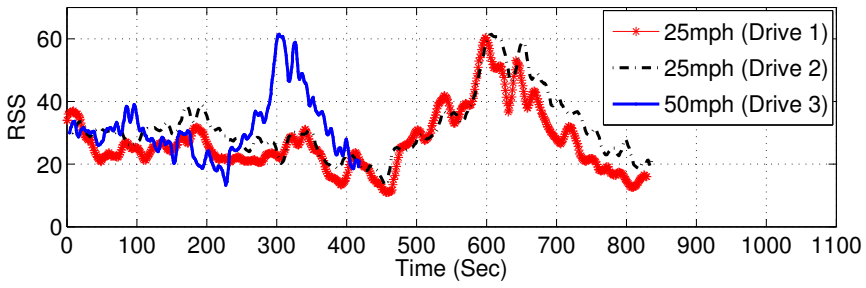


Figure 5.1: Stability of RSS over time

Let us consider another scenario where the vehicles move along the same road segment with varying speeds (not uniform over the entire segment). In this case, the testing trace may

not have a uniform scale factor with respect to the reference trace and the stretch factor can vary over the length of the trace. Fig. 5.2 plots the instantaneous speed and RSS trace from the associated cell tower for two vehicle trips along the same stretch of a road moving at different speeds. The vehicle drove roughly at the same speed during the first 150 seconds of both trips, but then it slowed down in the first trip and sped up in the second.[1] The graph shows how the RSS traces remain similar over the first part of the trace, where the vehicle traveled at the same speed, and depart when the vehicle varied its speed in the later part of these two trips. Note also, how the trace from the slower trip is essentially a stretched version of the faster trip in the second part of the trace. For example, the dip below an RSS value of 20 dB occurs in the same location in both trips but due to the speed difference, the graph shows them at different times.

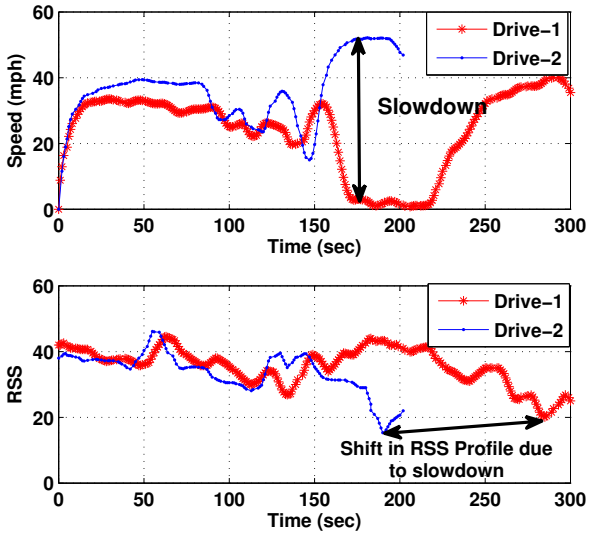


Figure 5.2: Stability of RSS over time

The key idea underlying our technique is to stretch (or compress) the RSS trace until it best matched the reference trace. Since the instantaneous speed over the reference trace is known, the algorithm can convert these stretch factors into instantaneous speed estimates for the test RSS trace. The time-warping algorithm makes use of this observation to estimate instantaneous speeds.

[1]The car traveled the same distance in both cases and stopped at the same physical location. However, due to the speed difference, the first trip took about 300 seconds while the second only lasted 200 seconds.

For both these techniques, we assume that training RSS profiles and their speeds are available for road segments under study. These could be collected as part of the service provider signal measurements to determine coverage. We also assume that the approximate starting location and the road segment the vehicle travels on is known, for example by monitoring handoff locations as shown in prior work [12].

5.3 Baseline algorithms

We choose three representative algorithms - *Localization Algorithm*, *Handoff Algorithm* and *Normalized Euclidean Distance Algorithm*, that have previously been used for estimating vehicular speed and detecting bottlenecks in road segments, as baseline approaches for comparing our algorithm with.

Localization Algorithm: This method as implemented by several commercial products [2, 79] estimates the speed of a mobile phone between two points by estimating the phone's locations at the two points, calculating the distance the phone has traveled and dividing it by the time traveled. In this work, we use the fingerprinting [68] algorithm for determining phone's location. The algorithm uses the RSS fingerprints obtained from 7 neighboring towers at different known locations as the training. When an RSS fingerprint is obtained from a mobile at an unknown location, the algorithm estimates the euclidean distance in signal space between this obtained fingerprint and all the training fingerprints and determines the location to be the location of the training fingerprint that yields the minimum euclidean distance.

Handoff Algorithm: The Handoff algorithm [39] involves detecting the location of the mobile based on existing knowledge of *handoff zones*. A handoff zone is the most probable location in a given road segment where the mobile switches from the current base station to a new one. Whenever a handoff occurs in the testing trace, the location of the mobile is estimated to be the location of the most probable handoff zone. A handoff typically occurs when the signal to noise ratio (SNR) drops below a certain threshold. It turns out that on any given road segment, the locations where the SNR drops below the threshold remain stable. The average speed estimate is then the distance between the previous predicted handoff location and the current predicted handoff location divided by the total time between the previous and

the current handoffs.

Normalized Euclidean Distance Algorithm: This algorithm detects speed changes during speed tracking, e.g., slowdowns, by calculating the normalized euclidean distance between consecutive GSM measurements and declaring a slowdown when the distance falls beyond a certain threshold. The normalized Euclidean distance between two RSS measurements A and B, having n common cell towers is defined as:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}/n \tag{5.1}$$

Note that Euclidean distance between successive samples from a mobile phone is directly proportional to the distance the phone moves in physical space, which in turn depends on how fast the phone moves. While we cannot derive an accurate speed estimate from this relation, we can still predict regions where there are slowdowns. We experimented with multiple other metrics suggested in [84], but found the normalized euclidean distance to work the best. Hence, we chose to use this algorithm for comparison with our mechanism.

5.4 Speed Estimation Algorithms

Speed estimation algorithms fall under two categories, namely, average speed estimation algorithms and fine-grained speed tracking algorithms. We propose Correlation Algorithm that can estimate average speeds with high accuracy and Derivative Dynamic Time Warping (DDTW) algorithm that detects fine-grained speed variations. Though DDTW can be used for estimating average speeds as well, the correlation algorithm is computationally more efficient than DDTW when detecting average speeds. The following two subsections explain these algorithms in detail.

5.4.1 Correlation Algorithm

The Correlation algorithm matches a measured received signal strength trace to a training trace obtained from the same road and infers the speed from the amount of stretching or compression that yields the best match. It assumes that the approximate starting location and the road segment the vehicle travels on is known, for example by monitoring handoff locations as shown in prior work [81].

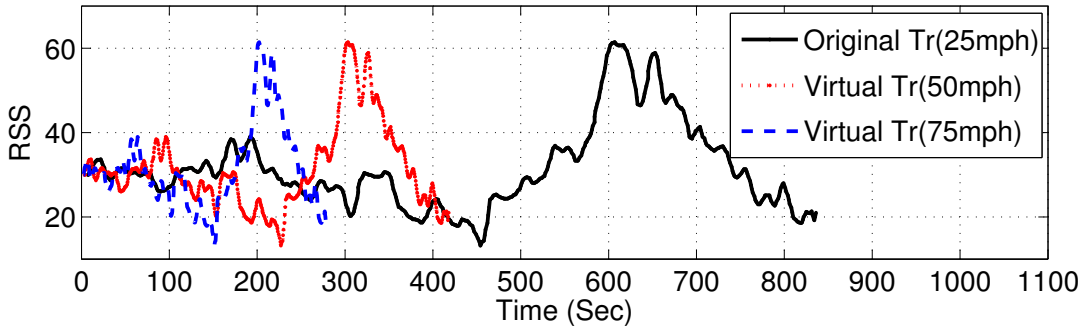


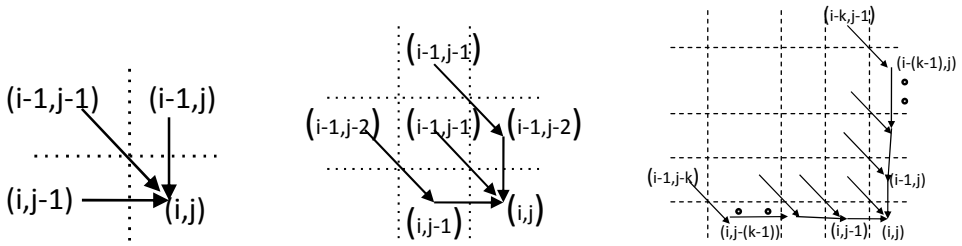
Figure 5.3: Generating Virtual training traces from actual training trace.

The Correlation algorithm creates k scaled versions of the training trace in its training phase. The number of different scaled versions k depends on the desired speed estimation accuracy. In our implementation we chose 1mph steps up to a maximum speed of 80mph, yielding $k = 80$ scaled versions of a training trace. The scaled versions are subsampled or interpolated, so that each retains a sampling rate of 1 measurement/sec. Figure 5.3 shows an example of creating two scaled profiles from a training profile of 25mph. Note that the total time to completion of the 25mph trace is twice as long as that for the 50 mph trace.

In the second step, its matching phase, the algorithm seeks to determine which of the k scaled versions of the training trace best fit a given measured trace, which we refer to as testing profile. To this end, it calculates the pearson's product moment correlation co-efficient [12] over the t seconds of the testing profile with t seconds of each of the scaled training profiles. The algorithm chooses the speed associated with the scaled trace that maximizes correlations. The overall running time of this algorithm is $O(k.n)$ where n is the number of RSS samples in the input trace and k is typically a constant that we decide ahead of time. Note that the correlation algorithm yields an average speed over the entire length of the road and cannot identify speed variations within the road segment. To address this issue, we propose an algorithm — Derivative Dynamic Time Warping (DDTW) which is explained in the following section.

5.4.2 Derivative Dynamic Time Warping Algorithm

Dynamic Time Warping is a classic dynamic programming algorithm which has been widely used for optimal alignment of two time series datasets and was particularly popular for applications like speech processing [64, 80], data mining [50, 67], and gesture recognition [24].



(a) Unconstrained (b) Constrained, $E_{MAX} = 2$ (c) Constrained, $E_{MAX} = k$

Figure 5.4: DDTW local constraints that restrict the admissible paths to every location within the matrix:

In particular, we use a variant of the DTW algorithm called Derivative Dynamic Time Warping (DDTW) [52], which exploits the same principle as DTW but for the input data, where, instead of the time-series of RSS, we use the time-series of derivative of RSS. As observed previously [52], if the two RSS profiles varied only on the time axis and not on the absolute values of RSS at any given location, DTW would have been sufficient. But RSS in an outdoor environment typically suffers varying amount of shadow fading under different environmental conditions which also alters its absolute value in any given location. DDTW can overcome this difference in the y-axis by working with derivatives of RSS where only the slope of RSS matters and not the absolute values. For example, if $A = (a_1, a_2, ...a_M)$ is a time series of RSS measurements collected over M time points, the input to DDTW is $A' = (a'_1, a'_2, ...a'_M)$, the derivative of A which is defined as

$$a'_i = \frac{(a_i - a_{i-1}) + (a_{i+1} - a_{i-1})/2}{2} \quad 1 < i < M. \tag{5.2}$$

Given two RSS profiles - A and B with lengths of M and N samples respectively, DDTW constructs a distance matrix $d[M \times N]$ which is defined as:

$$d(i, j) = (a'_i - b'_j)^2 \tag{5.3}$$

where a'_i and b'_j are the i^{th} and j^{th} elements of the derivative of the RSS profiles A and B respectively. With this $d[M \times N]$ as the input to the algorithm, DDTW returns a warping path $P = (p_1, p_2, ...p_k)$ where $p_l = (x, y) \in [1 : M] \times [1 : N]$ for $l \in [1 : k]$ as shown in Figure 5.5. The warping path must satisfy the following conditions:

1. **Boundary Condition**: $p_1 = (1, 1)$ and $p_k = (M, N)$. This ensures that the warping path always starts at $(1, 1)$ and ends at (M, N).

2. **Monotonicity Condition**: If p_{k-1} = (c, d) and $p_k = (e, f)$, we have $e - c \geq 0$ and $f - d \geq 0$. The monotonicity condition ensures that the matching always progresses in the forward direction of time.

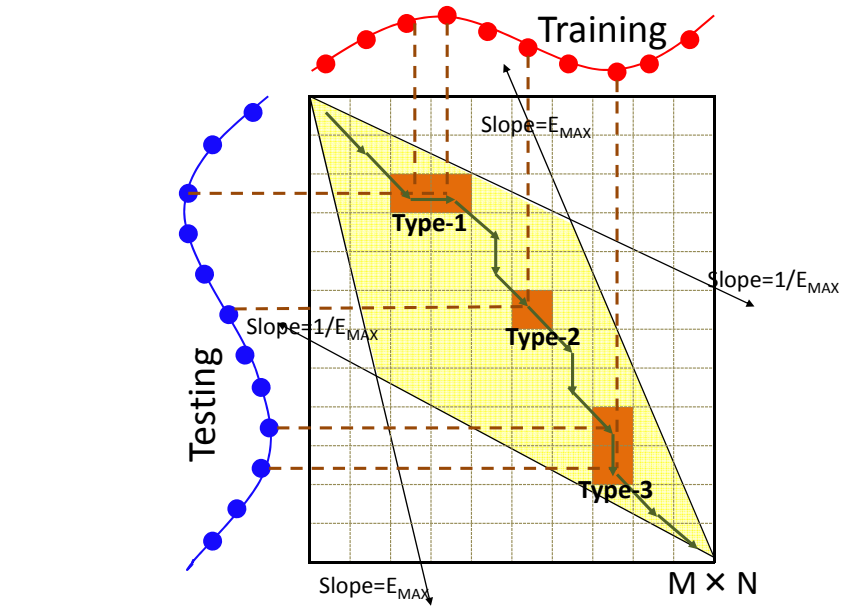
3. **Global Constraints**: Global constraints are constraints that limit the region in which the warping path can exist. In addition, global path constraints also guarantee the existence of a path from $(1, 1)$ to (M, N). Figure 5.5 illustrates the region for warping path generation. The region enclosed within the parallelogram is the region that corresponds to the global constraints. In Figure 5.5, E_{MAX} is defined as the maximum allowable expansion (or compression) in time axes of one time series with respect to the other, and is chosen to be $max(2, \lceil \max(M, N)/\min(M, N) \rceil)$. The ratio $\lceil \max(M, N)/\min(M, N) \rceil$ defines the amount of expansion of one trace relative to the other. Accordingly, the sides of the parallelogram are set to have slope values of E_{MAX} and $1/E_{MAX}$.

4. **Local Constraints**: Finally, Local constraints define the set of admissible step-patterns. There are three types of step progression: *horizontal*, *vertical* and *diagonal*. As shown in Figure 5.4, different kinds of local constraints are possible. For example, Figure 5.4(a) shows the most unrestrictive step constraint where (i, j) can be reached from one of its three neighbors $(i - 1, j - 1), (i - 1, j), (i, j - 1)$. Whereas Figure 5.4(b) and 5.4(c) illustrate more constrained progressions where a diagonal progression is forced for every E_{MAX} horizontal or vertical progressions.

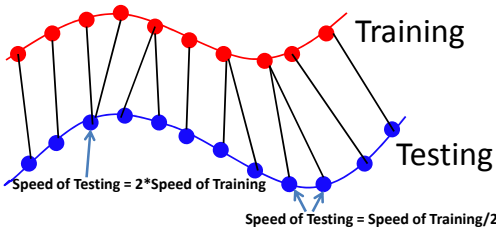
To generate a warping path, DDTW constructs a cost matrix $C[M \times N]$ which represents the minimum cost to reach any point (i, j) in the matrix from $(1, 1)$ using a dynamic programming formulation. For example, in Figure 5.4(a), (i, j) can be reached from one of its three neighbors, namely, $(i-1, j-1)$,$(i-1, j)$, and $(i, j-1)$, and the algorithm picks the neighbor that has the minimum cost. This relation can be shown as:

$$C(i, j) = d(i, j) + \min(C(i - 1, j - 1), C(i, j - 1), C(i - 1, j)). \tag{5.4}$$

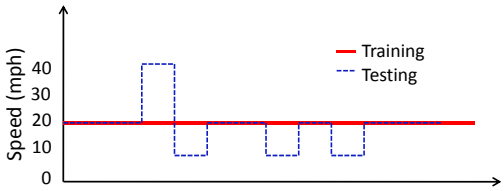
However, using an unconstrained local constraint as shown in 5.4(a) can lead to an undesirable effect called "singularities" [52] where either one sample point in the testing is mapped to a very large number of samples in training (unrestricted horizontal progression) or many points in testing map to the same point in training (unrestricted vertical progression). This effect as



(a) Cost Matrix $C(M, N)$ and the warping path generation



(b) Optimal alignment of training and testing



(c) Estimated speed from a warping path

Figure 5.5: Illustration of vehicular speed estimation from DDTW.

observed previously [64] can be minimized by using a more constrained topology for forward progression. In this work, we thus take an approach of using the constrained DDTW with a maximum expansion of E_{MAX}. our local constraints resemble the ones in Figure 5.4(b) and 5.4(c). For example, 5.4(b) forces a diagonal progression before every horizontal or vertical progression, whereas 5.4(c) allows up to (E_{MAX}) horizontal or vertical progressions before forcing a diagonal progression. For a complete description of local constraints, we refer the readers to [64]. The local constraints that we use in this work allow up to E_{MAX} vertical or horizontal progressions before forcing a diagonal progression and the cost matrix $C(i, j)$ corresponding to this local constraint can be formulated as

$$C(i,j)=\min\Big[\min_{1\le r\le E_{MAX}}\Big(C(i-1,j-r)+\sum_{j1=j-(r-1)}^{j} d(i,j_1)\Big),$$
$$\min_{2\le r\le E_{MAX}}\Big(C(i-r,j-1)+\sum_{i1=i-(r-1)}^{i} C(i_1,j)\Big)\Big]. \tag{5.5}$$

Note that the optimal path to (i, j) depends only on the values of (i', j') where $i' \le i$ and

$j' \leq j$. From the cost matrix, the algorithm derives a warping path P by back-tracking the constructed cost matrix from (M, N) to $(1, 1)$. While backtracking, the path that the algorithm chooses from any point (i, j) will be the (i', j') that resulted in optimal $C(i, j)$. We will next explain how we use the warping path to estimate the speed of the testing trace.

Estimating Vehicular Speed from DDTW's warping path The DDTW algorithm returns a warping path P between the points $(1, 1)$ to (M, N). This warping path defines the optimal alignment between the training and the testing RSS traces, i.e, it maps the RSS samples in the testing trace to the RSS samples in the training trace. As explained in Section 5.1, there is a direct correlation between the speed of vehicle and the overall shape of the RSS curve and the optimal mapping of the RSS-curves can yield a speed estimate for the testing trace relative to the training.

We define three types of matching between training and testing traces: *Type-1*, *Type-2*, and *Type-3*. If one point in the testing trace is mapped to k points in the training trace, the resulting speed estimate for the testing trace is k times that of the training. We call this as Type-1 matching as shown in Figure 5.5(a). The figures illustrate this for $k = 2$. Similarly, Type-3 matching is when k points in the testing trace are mapped to one point in the training trace, speed of the testing trace is $1/k$ times the training speed. Finally, Type-2 match is when one point in testing maps to exactly one point in training trace. In this case, speed of testing equals speed of training.

We note that the estimated speed from time warping is always a multiple of the training speed. For example, if the training speed at any instance is 20mph, the resulting testing speed can only be multiples of 20mph such as 60mph, 40mph, 20mph or 10mph. Figure 5.5(b) plots the optimal alignment between the training and the testing RSS traces obtained from DDTW and illustrates the speed derivation procedure for the testing trace. Figure 5.5(c) plots the estimated speeds for the testing trace given a training trace with a constant speed of 20mph.

We observed that there are speed fluctuations in the estimated speed. In order to remove these fluctuations, we apply a moving window smoothing filter over the estimated speed which averages the speed estimates within the entire window to produce a single speed. The choice for the window size should not be too large since this might smooth out all variations leaving a very coarse speed estimate. Similarly, having a very small window size may result in the

overall speed estimation to be highly fluctuating. We will evaluate the length of the optimal smoothing window in Section 5.6.

5.5 Applications of Fine-Grained Speed Tracking: Slowdown Detection

In practice, most traffic engineering applications do not require the instantaneous speeds of vehicles and are more concerned about regions of bottlenecks. Such bottlenecks in road networks can in turn be detected from vehicular speeds by observing the regions where vehicles typically slowdown or by observing the normalized euclidean distance where the normalized euclidean distance between successive samples go below a threshold. We will next provide a formal definition of slowdown and describe the scheme for slowdown detection.

We define a slowdown as a sudden reduction in the speed of a moving vehicle by more than τ mph to a value below μ mph. The duration of the slowdown is the period of time the speed remains below μ mph. A slowdown is detected by sequentially scanning the input trace. The input trace can be the groundtruth speed data derived from GPS readings, DDTW estimated speed, speed estimate from the Localization algorithm, or the Normalized Euclidean Distance from Normalized Euclidean Distance algorithm. Our scheme identifies *peaks* and *dips* in the input trace. A peak occurs in the input trace at any given point when its first derivative (slope) at that point changes from positive to negative. Similarly dips occur when the slope changes from negative to positive. Our scheme initially assign a very low value to the first detected peak and a very high value to the first detected dip. As the scheme proceeds scanning the trace, the peak value is adjusted to the highest observed peak. Similarly, the dip value is adjusted to the lowest observed dip. After every adjustment of the peak and the dip, if $(peak - dip) > \tau$, and $dip < \mu$, a slowdown is declared. The duration of this slowdown is then the period of time the dip remains below μ. Finally, the peaks and dips are reset to the lowest and highest values respectively and the scheme repeats until all slowdowns are detected in the specific trace.

The main challenge in identifying slowdowns accurately lies on the choice of τ and μ. We performed an empirical study on 18 of our GPS traces that lasted for a total of 6.4 hours and picked a threshold of 25mph for τ since most breaking events involved slowing down the vehicle from 40-45mph speed limit in arterial roads to a very slow speed of around 5-10mph.

Our choice for μ is 20mph because most residential regions have a speed limit of 25mph or more and we do not want to classify those residential regions as bottlenecks.

While a choice of 25mph and 20mph for τ and μ fits the ground truth speed from GPS, these thresholds need not be the same for the speeds estimated from either DDTW or Localization. and the normalized euclidean distance estimated by Normalized Euclidean Distance algorithm. For example, we showed in Section 5.4.2 that the speed estimate from DDTW at every instance is a multiple of the training speed which in turn requires a moving window smoothing filter to be applied over the estimated speed to get the speed estimate. However, due to this smoothing, an actual speed change of 25mph in the ground-truth speed may only correspond to a speed change of 15mph in the estimated speed.

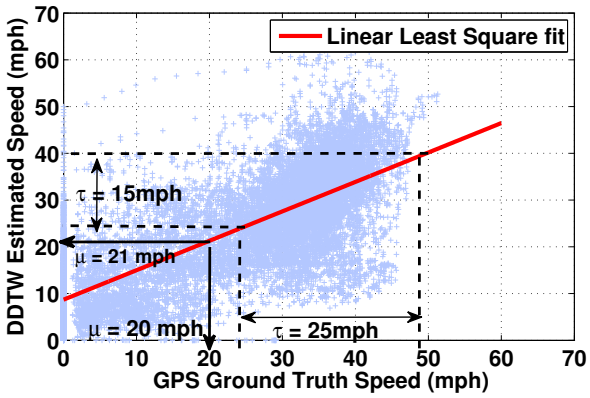


Figure 5.6: Least square fit between the ground-truth speed from GPS and the estimated speed from DDTW.

In order to capture this relationship between the ground-truth speed and the estimates from other algorithms, we performed a regression analysis using a linear least square fit over the data sets. The inputs to the regression analysis are the ground-truth speed from GPS and the output from any of the three estimation algorithms under study: DDTW, Localization and Normalized Euclidean Distance. For instance, Figure 5.6 shows a scatter plot of the ground-truth speed from GPS versus the estimated speed from DDTW, and the corresponding fitted line obtained from linear least square fit. The slope of the fitted line determines that a drop in ground-truth speed by 25mph corresponds to only 15mph drop in the estimated speed from DDTW. Similarly, a μ value of 20mph in ground-truth corresponds to 21mph in estimated speed from DDTW. Table 5.1 summarizes the values for τ and μ obtained using the above process for the different

slowdown estimation algorithms.

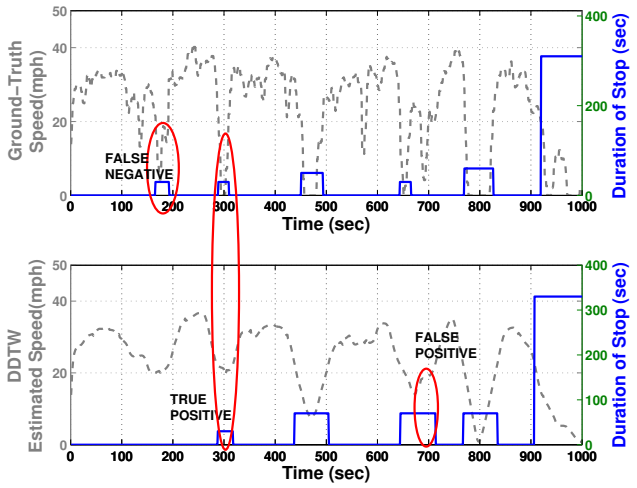


Figure 5.7: Figure Illustrating the metrics for quantifying the slowdown detection performance, namely, false negative, true positive and false positive

The slowdown detection algorithm takes any of the four inputs, namely, *ground-truth speed trace*, *estimated speed trace from DDTW*, *estimated speed trace from Localization* or *normalized euclidean distance trace from Norm.Euc.Dist algorithm*, along with their respective τ and μ values. Figure 5.7 illustrates the results of slowdown detection performed on a 1000 seconds long ground-truth speed trace from GPS and the corresponding estimated speed from DDTW respectively. We treat the slowdown detection obtained from the GPS data as the ground truth. The y-axis on the left side corresponds to the speed of the traces, while the y-axis on the right represents the duration of each of the identified slowdown locations.

Finally, we will use three metrics: *True Positive*, *False Positive* and *False Negative* to quantify how well the the different algorithms detect bottlenecks in Section 5.6. In the slowdown detection, true positives are the time periods when the ground-truth slowdowns coincide with

	τ	μ
Ground-Truth Speed from GPS	25 mph	20 mph
Estimated Speed (DDTW)	15.73 mph	21.28 mph
Estimated Speed (Localization)	4.92 mph	20.84 mph
Norm. Euc. Dist.(Norm. Euc. Distance Algorithm)	4.15 dBm	26 dBm

Table 5.1: Thresholds τ and μ for the slowdown estimation algorithms

the slowdowns estimated in the result of the algorithm under consideration, i.e.,DDTW or Localization, or Normalized Euclidean Distance. Whereas false positives occurs when slowdown is detected in our scheme under consideration but not in the ground-truth. False negative is when the slowdown exists in the ground-truth, however, is not detected by our scheme.

5.6 Experimental Evaluation

We begin this section with a description of our experimental setup. We then present the speed estimation performances for the two proposed algorithms — correlation and DDTW and compare their performances with other state of the art speed estimation algorithms.

5.6.1 Experimental Setup

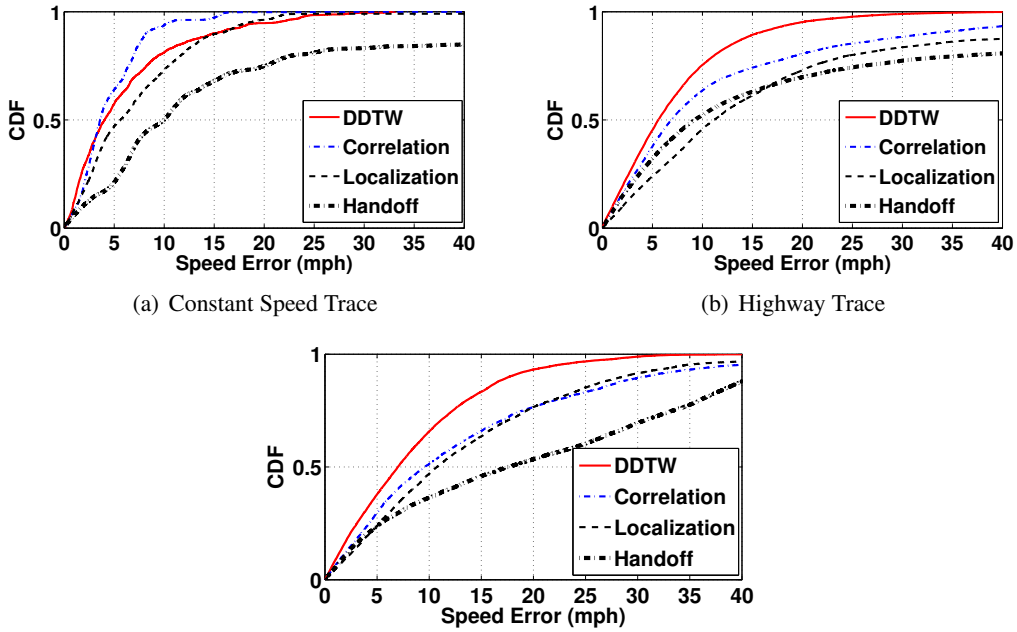
Our evaluation studies the accuracy of our speed estimation algorithms on a received signal strength dataset collected with serveral handsets over a period of one month both outdoors and indoors. Outdoors, we used GSM enabled HTC Typhoon phones running the Intel-POLS [6] software on the AT&T network for all our experiments. The software records the time, cell tower description (Cell ID, MNC, MCC, LAC, IMEI), and the received signal strength from the 7 strongest cells once every second. We used Holux GPSlim236 GPS receivers paired with the mobile phones through bluetooth for logging the ground truth location information for all the traces. Our outdoor traces were collected under varied driving scenarios, in three different environments.

1. **Highway trace**: Two experimenters collected traces while driving from home to work at varying speeds for over a month. The common route in these drives had a 14 mile highway stretch without traffic lights, which contributed a total of 36 traces.
2. **Constant Speed trace**: This experiment involved driving on a 5 mile stretch of road thrice at three different speeds: 25mph, 40mph and 55mph.
3. **Arterial roads trace**: This experiment involved 18 drives, with each drive spanning a distance of 10 miles on arterial roads with traffic lights. We chose the roads with traffic lights to create traces that have high variability in speeds.

Our indoor traces were collected on a 802.11b Wi-Fi network. We collected 9 traces in which the experimenter placed a laptop equipped with 802.11b WG511T Wi-Fi card in a cart and moved along a corridor measuring 228 ft in length thrice at three different speeds (1ft/sec, 2ft/sec, 4ft/sec) while sending out packets at the rate of 2 pkts/sec. Three receivers were placed along the corridor: one closer to the beginning of the corridor, one in the middle and one closer to the end of the corridor. The receivers were 802.11b/g enabled, configured to listen on channel 6 in monitor mode and log the packets using Tshark [7] packet sniffer utility. In total, we had 550 seconds of data logged at each of the receivers.

Unless otherwise noted, we used the outdoor datasets for all experiments in this work.

5.6.2 Speed Estimation Accuracy



(a) Constant Speed Trace

(b) Highway Trace

(c) Arterial Roads Trace

Figure 5.8: Speed Estimation Accuracy comparison across four algorithms: DDTW, Correlation, Localization, and Handoff

In this section, we evaluate the accuracy with which DDTW and Correlation algorithms can estimate speed and compare it to the accuracy achieved by the Localization and the Handoff algorithms described in Section 2.3. We do not include results for the Normalized Euclidean

Distance algorithm in this section since the algorithm can be used only to detect speed variations and cannot be used to estimate speeds. Figure 5.8 plots the CDF of the error between the actual and the estimated speeds for all the four algorithms — DDTW, Correlation, Localization and Handoff collected under three different outdoor driving scenarios, namely, constant speed traces, highway traces and arterial road traces. For the constant speed traces, the correlation algorithm has the least median error of $4mph$ followed by the DDTW, Localization and Handoff algorithms respectively. This result is very encouraging since it shows that the correlation algorithm can be very effective for scenarios where speed changes are not too high. DDTW on the other hand outperforms all the other algorithms under highway and arterial road traces. In the highway traces, DDTW exhibits a median errors of $5mph$ whereas the correlation, localization and handoff algorithms exhibit $7mph$, $12mph$ and $10mph$ respectively. The performance gain of DDTW over correlation algorithm is not very significant ($\pm 2mph$) in the highway traces. This can be attributed to the fact that vehicles in highway typically move at close to constant speeds. On the other hand, on arterial roads, DDTW outperforms the other three algorithms significantly, achieving a median error of 6.5 mph, which is almost twice lower than that achieved by the other three algorithms. This can be attributed to the fact that DDTW has been specifically designed to detect high speed variations (which is the case on arterial roads) whereas, the remaining algorithms can only estimate average speeds. In evaluations that follow, we will only present the results of DDTW on the arterial road traces unless otherwise noted since the other two scenarios – Constant speed and Highway traces do not exhibit a lot of variability in speeds. We also present comparison of DDTW only with the localization algorithm since correlation and localization algorithms have similar performance on arterial roads and the handoff algorithm has a poorer performance compared to localization.

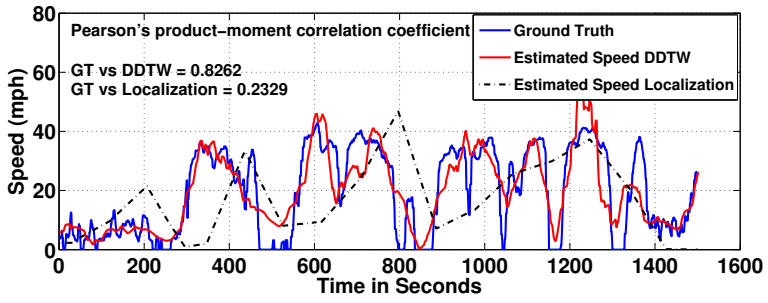


Figure 5.9: Ground truth and estimated speeds of DDTW and Localization

In order to illustrate the effectiveness of DDTW, Figure 5.9 plots the ground truth (actual) speed of a vehicle obtained through GPS, as well as the estimated speeds of the DDTW and the Localization algorithms. The drive took 1500 seconds to complete. The figure shows that the speed estimated by DDTW matches the actual speed very well, whereas the Localization algorithm performs poorly. To quantify how closely the two algorithms follow the actual speed, we calculated the Pearson's product-moment correlation coefficients between each of the algorithms and the actual speed. The Pearson's correlation coefficient measures a linear dependence between two variables. The coefficient of 1 means very strong positive correlation. The coefficient of 0 means no correlation. The Pearson's correlation coefficients are shown in the upper left corner of the figure. The actual speed and the DDTW algorithm exhibit very strong correlation of 0.83. The Localization algorithm, on the other hand, has a weak correlation with the actual speed of 0.34. We also calculated the Pearson's correlation coefficients for all the testing traces combined. The correlation between the estimated speed of DDTW and the actual speed was strong with a correlation coefficient of 0.75, whereas the correlation between the estimated speed of Localization and the actual speed is poor with a correlation coefficient of just 0.11. The above results are very encouraging and we believe that the speed estimates from DDTW would be suitable for a wide range of traffic engineering applications. In the next section, we will evaluate a method that can be used by an application that detects regions of bottleneck on a road. This requires detecting when a vehicle significantly reduces its speed.

5.6.3 Slowdown Detection Accuracy

In this section, we evaluate the accuracy with which DDTW, Localization and Normalized Euclidean Distance algorithms can detect slowdowns, as defined in Section 5.5. Recall that a true positive occurs when an algorithm correctly predicts that there is a slowdown. A false positive occurs when an algorithm predicts that there is a slowdown but there is none. A false negative occurs when an algorithm doesn't predict a slowdown and there is one. Please refer to Figure 5.7 for an illustration of all these metrics.

We present our results using Precision, Recall and F-measure [76]. Precision captures the percentage of correct slowdown predictions and it is defined as the total duration of true positives divided by the sum of total duration of true positives and false positives. The higher is

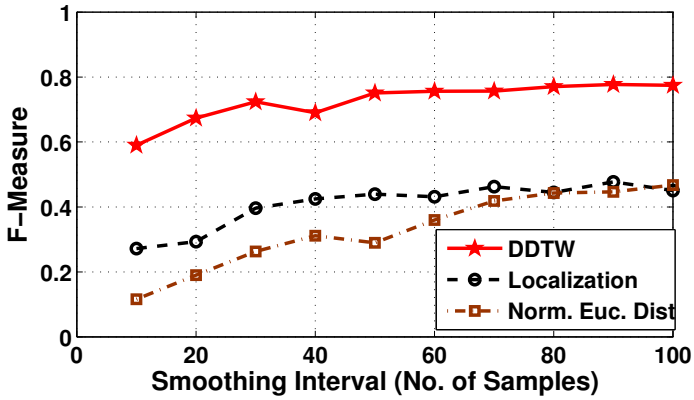


Figure 5.10: The effect of the smoothing interval on the F-measure.

the Precision, the more accurate an algorithm's estimations are. Recall captures the percentage of actual slowdowns that were detected and it is defined as the total duration of true positives divided by the sum of true positives and false negatives. The higher the Recall, the more actual instances of a slowdown an algorithm has predicted correctly. F-measure is used to estimate the Precision/Recall tradeoff and it is defined as follows:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5.6}$$

The higher the F-measure, the better is an algorithm's slowdown detection accuracy. In our case, it is possible to trade off Precision for Recall by changing the smoothing interval, as defined in Section 5.4.2. Figure 5.10 plots the F-measure for different smoothing intervals for the DDTW, Localization and Normalized Euclidean Distance algorithms.

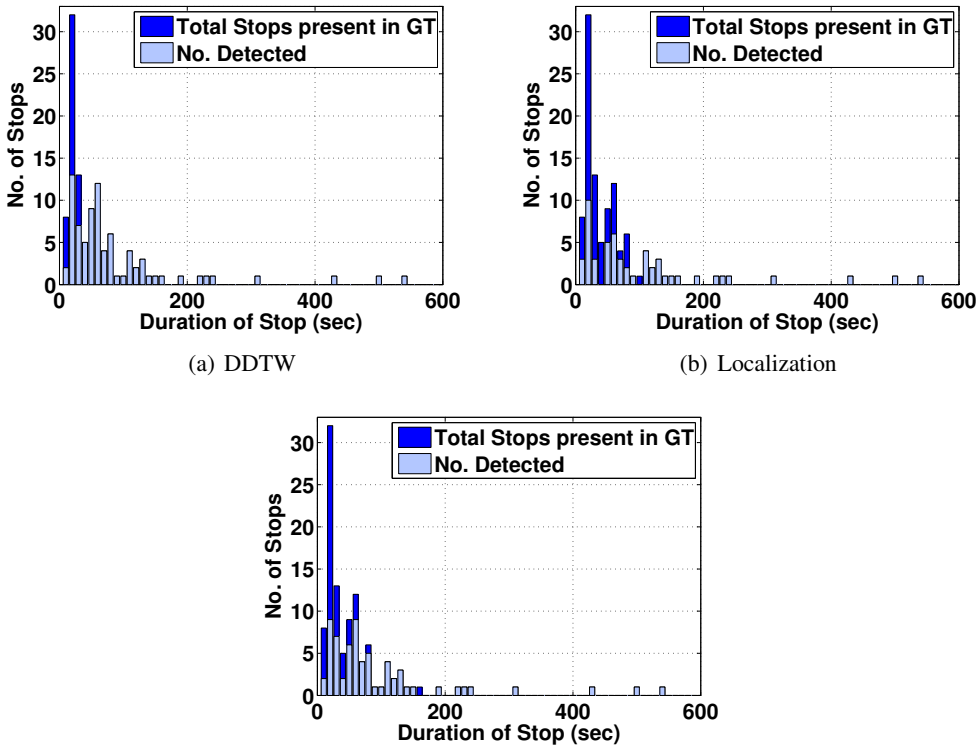
The figure illustrates that the F-measure for DDTW is almost twice as high as the F-measure for Localization and Normalized Euclidean Distance algorithms across the entire range of smoothing intervals. This indicates that the DDTW has higher Precision and Recall compared to the other algorithms. We pick the smoothing interval that achieved the highest recall value for each algorithm, which in this case, was 50, 90 and 100 for DDTW, Localization and Norm. Euc. Dist respectively.

Table 5.2 summarizes the Precision and Recall values for DDTW, Localization and Normalized Euclidean Distance algorithms for their respective optimal smoothing intervals derived from their F-Measure in Figure 5.10. DDTW significantly outperforms the other two algorithms achieving Precision of 0.68 and Recall of 0.84. Its Precision is 94% higher than that

	Precision	Recall
DDTW	0.68	0.84
Localization	0.38	0.63
Normalized Euclidean Distance	0.39	0.59

Table 5.2: Slowdown Detection Performance of DDTW, Localization and Normalized Euclidean Distance Algorithms.

of Localization and 74% higher than that of Normalized Euclidean Distance. Its Recall is 40% higher than that of Localization and 42% higher than that of Normalized Euclidean Distance.



(a) DDTW

(b) Localization

(c) Normalized Euc Distance

Figure 5.11: No. of slowdowns with different durations predicted by (a)DDTW (b)Localization (c)Normalized Euclidean Distance

Next, we study the impact of the duration of a slowdown on the ability of the algorithms to detect it. The intuition tells that it should be easier to detect slowdowns of a longer duration. The duration of a slowdown is defined in Section 5.5 as the total time the speed remains below the threshold of μ mph.

Figure 5.11 plots a histogram of the number of slowdowns of a given length that appear in the trace and the number of slowdowns that are correctly detected by each of the three

algorithms. Although all algorithms can correctly detect all slowdowns of 120 seconds or more, only DDTW detects all slowdowns that are longer than 30 seconds. The algorithms cannot detect slowdowns of a short duration because of the smoothing that is applied to average out the oscillations in speed predictions, which in turn results in smoothing out abrupt speed changes that last for short durations.

5.6.4 Effect of Alignment Error on Speed Estimation Accuracy

In this section, we study the effect of the alignment error between the training and testing traces on the speed estimation accuracy of DDTW. Recall from Section 5.4.2 that introducing an alignment error results in applying DDTW on training and testing traces that are shifted in time by the value of the alignment error. Note that although we study the effect of alignment error of up to 500m, a typical GSM based localization system has a median localization error of less than 100m [18]. Therefore, it is reasonable to assume that, in practice, DDTW would achieve speed estimation accuracy that is equivalent to the one obtained with a 100m alignment error.

Alignment Error(m)	Median Error (mph)
0	5.2
100	6.5
200	7.12
500	8.57

Table 5.3: Effect of alignment error on speed estimation accuracy

Table 5.3 summarizes the median error in miles per hour for different alignment errors. When a localization systems provides an accurate location estimate, DDTW suffers from no alignment errors and has a median speed estimation error of 5.2 mph. When an alignment error of 100m is present, the accuracy of DDTW degrades slightly to 6.5 mph. Even in this case, DDTW performs much better than the Localization algorithm that achieves the median speed estimation accuracy of 13 mph.

5.6.5 Indoor WiFi-based Experiment

We finally verify if DDTW technique can be used across a different wireless technology and a different environment for the same purpose of speed estimation. To this end, we use the

Wi-Fi data in which we performed 9 indoor Wi-Fi experiments where the experimenter moved between the given two points in a long (228ft) corridor thrice at three different speeds [2], namely : 1ft/sec(0.68mph), 2ft/sec(1.36mph), 4ft/sec(2.72mph) while sending out packets at the rate of 2 pkts/sec. We had three Wi-Fi receivers, each recording the RSS from this transmitter.

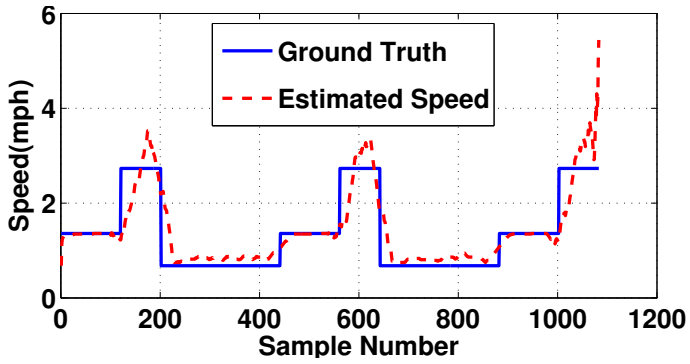


Figure 5.12: Speed estimates from DDTW on indoor environment using RSS from receiver-1

We can see from Figure 5.12 that the estimated speed closely follows the ground truth. The median error for this receiver is 0.1527mph. The median errors on receiver-2 and receiver-3 were 0.1388mph and 0.1527mph. This result is encouraging since this proves the generality of the proposed mechanism across different wireless technologies and shows that it is effective at detecting even very small speed changes indoors.

5.7 Summary of Key Ideas

To summarize, the key contributions of this chapter are

- We proposed two speed estimation algorithms, namely correlation algorithm that was effective at predicting average speeds and DDTW algorithm that was effective at tracking fine-grained speed variations. The algorithms used the received signal strength from mobile phones and exploits the stability of signal strength measurements over time on any given road segment to optimally align the training and testing signal strength traces and derive speed.

[2]The corridor had markers on floor separated by 1ft. The experimenter placed the laptop on a cart and moved over n markers within a second to generate a RSS trace with a speed of nft/sec

- We experimentally evaluated our algorithms on real signal strength measurements captured with mobile phones on various road road-segments and showed that correlation algorithms is very effective at estimating speeds that do not vary much and DDTW algorithm is effective even at predicting highly varying speeds. We show that the speed estimated by DDTW has a very high correlation with the ground-truth speed reported by the GPS and exhibits a median error within $\pm$ $6.5mph$ across the 6.4 hours of driving traces. The high speed estimation accuracy enables the DDTW algorithm to effectively predict bottlenecks in road segments with a precision of 68% and a recall of 84%.

- Additionally, to demonstrate the generality of our proposed speed tracking technique across different environments and radio technologies, we experimented with Wi-Fi RSS traces from an indoor environment and successfully showed that our technique can be used to detect walking speeds indoors as well.

Chapter 6

Conclusions and Future Work

The ubiquity of wireless devices has resulted in a wide variety of location based services (LBS) being deployed. While the location traces derived from GPS enabled devices seem to be the dominant source of input to these LBS, the heavy battery drain imposed by the GPS devices coupled with the increasing privacy concerns at sharing one's location can make such LBS impractical in a larger scale. An alternative to using GPS devices would be to either use cellular phone's location or location derived from other wireless devices, namely, Wi-Fi or Bluetooth or 802.15.4, etc. Using this alternative form of location is attractive since they consume an order of magnitude less energy than GPS and they do not require explicit user participation as the location estimation can be performed at the infrastructure end. However, these techniques report a median location accuracy ranging from $3m - 60m$. Therefore, using these location estimates to derive other location-related context like speed, co-mobility, mobility states and decelerations can result in poor estimates. In this thesis, we address this problem by estimating the other location-related context directly from the wireless signal strength.

We started with the categorization of the factors that results in location estimation errors for indoor RSS-based localization schemes. We verified a variety of factors like landmark density, training density, localization algorithms through extensive experimentation on a high-density 400 node wi-fi testbed. Our results suggested that increasing the landmark density with no regards to the underlying model in the algorithm can deteriorate the localization accuracy for lateration based algorithms. Our results also suggested that the most important factor that influences the localization accuracy is the presence of multi-path that results in the underlying propagation model to be violated in the lateration algorithms. By choosing the right set of landmarks that obey the propagation model, we showed that we could achieve a median localization errors as low as $0.24m$. However, in practice, such a high density deployment seems

impractical and the median errors achieved by the different algorithms with the typical 4 landmark deployment remained close to 3m. Therefore, we looked at ways to bypass the location determination step for inferring other location-related properties like co-mobility and speed.

The second part of my thesis dealt with determining co-mobility directly from wireless signal strength. In our approach, we estimated the correlation between the time series of RSS from different devices rather than comparing their RSS measurements at discrete time instances. Therefore, the transient differences in RSS fades experienced by the co-moving devices did not affect the performance of co-mobility detection. Our results also suggested that our approach was generic enough at detecting co-moving wireless devices that belonged to different radio technologies, namely 802.11b and 802.15.4. We note that our technique works with just one wireless receiver in range without the need for any calibration whereas, localization techniques require at least 3 receivers in range and would exhibit lower accuracy without calibration. Our observations also paved a way to extending this framework for inferring more location-related properties like speed.

The third part of my thesis extended on the observation behind co-mobility detection systems to a more generic setting where the wireless devices now moved over the same path but at different times and different speeds. The observation that moving along any given path at a given speed produces similar RSS fades was the key to speed estimation. If we had an RSS trace for any given path collected at a known speed (training trace), the speed estimation problem was reduced to mapping the collected RSS trace (testing trace) to the RSS trace with known speed. This mapping between the training and the testing traces served as the means to estimating the speed for the testing trace. We applied correlation and dynamic time warping techniques to derive this mapping. Our results showed that we could estimate vehicular speed outdoors using GSM RSS from mobile phones in vehicles within a median error of $\pm 5mph$. This result also encouraged us to verify if our technique was suitable at detecting bottlenecks (decelerations) on road segments and we found that our algorithm could detect bottlenecks that result in slowdowns lasting longer than 30 seconds. We also showed the generality of our algorithm by extending it indoors to verify its applicability for detecting walking speed estimation using 802.11b radios instead of GSM. In contrast, the existing systems that used the GSM RSS to localize phones over time and derive vehicular speeds exhibited errors in the orders of $\pm 12mph$.

Overall, our results suggest that using the time-series of RSS directly to infer location-related properties is more accurate then deriving the same from location estimates. The arguments that support this observation are as follows:

- Time-series of RSS has a continuous collection of signal strength measurements. When a continuous time-series is analyzed, the analysis automatically models the history which can make the inferences about present more meaningful. In contrast, localization systems make location estimates from discrete fingerprints collected in any given location. The exception to this rule are localization systems that employ particle filters which estimates the probability of being at the next position based on the current location. However, these systems still require a mobility model/building model to capture the transitions that happen from current location to the next location.

- Secondly, analyzing the time-series does not require any environmental modeling or infrastructure set-up whereas the localization systems require at least four access points to estimate the location of a transmitter within an error of 3m and requires extensive training to model the propagation parameters.

- Thirdly, the distortions in RSS caused by shadow fading (the changing environment) cannot typically be modeled in any localization system since these fades change dynamically between the modeling phase and the execution phase. We however showed that such distortions can actually work to our advantage if we analyzed the time-series.

Our results also bring out another important trade-off that exists between adding more resources to improve localization accuracy versus reusing the existing communication infrastructure with the detection techniques outlined in this paper. Our results pointed out that increasing the landmark density from 4 to 25 brought down the median errors of localization from 3m to 1m. However, the common argument for using RSS for localization in the literature is the ability to reuse the communication infrastructure. The communication infrastructure is usually deployed to maximize the coverage with minimal number of access points. Due to this, finding 25 access points around a given transmitter is not practical unless more access points were deployed specifically for the purpose of improving localization. Our scheme that utilizes

the time-series of signal strength makes all its conclusions with measurements at a single access point. Therefore, at any point in time, if the transmitter is in communication range to at least one access point (the same requirement as that of the communication infrastructure), our scheme can estimate the other location-related properties with high accuracy.

As an extension, we envision using the speed and co-mobility information derived using RSS to in turn improve the accuracy of localization. Our research could be extended to derive a new paradigm of indoor localization called Path-based localization where indoor environments could be modeled as a series of paths rather than discrete locations in space. The problem now becomes matching time-series of testing RSS with time-series of training RSS for location determination. Note that using a single receiver is still sufficient for location estimation if we used the time-series. Doing it this way can also take advantage of the fact that time-series of RSS are robust to transient fades.

There is also a need to rethink the way we would design the future context aware stacks. For example, instead of performing localization to derive the state of the transmitter, we could first derive mobility states from signals which can in turn be used as a trigger for localization if the transmitter's state is mobile. We believe that including the other richer set of primitives that can directly be inferred from signal strength measurements can augment the amount of information available for the context inference systems and that our work proposes ways to derive these other primitives with high-accuracy.

References

[1] Aeroscout company. http://www.aeroscout.com/.

[2] AirSage Inc. http://tinyurl.com/c782oz.

[3] Cisco Inc., Maximum Throughput Calculations for 802.11b WLAN. http://tinyurl.com/kv3yjn.

[4] http://traffic.berkeley.edu/.

[5] Orbit testbed. http://www.orbit-lab.org/.

[6] Privacy observant location system (pols). http://pols.sourceforge.net/.

[7] Tshark - network protocol analyzer and traffic dumper. http://tinyurl.com/zc98g.

[8] Ubisense company. http://www.ubisense.net.

[9] Vodaphone's traffic online. http://www.its-munich.de/pdf/DynInfoBY/4-Birle.pdf.

[10] A.LaMarca, Y.Chawathe, S.Consolvo, J.Hightower, I.Smith, J.Scott, T.Sohn, J.Howard, J.Hughes, F.Potter, J.Tabert, P.Powledge, G.Botriello, and B.Schilit. Place lab: Device positioning using radio beacons in the wild. In *In Proceedings of the third International Conference on Pervasive Compiting Pervasive*, pages 116–133, Munich, Germany, May 2005.

[11] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM*, March 2000.

[12] Keith G. Calkins. E-Book, An Introduction to Statistics. http://tinyurl.com/2jlrsu, 2008.

[13] Gayathri Chandrasekaran, M.A. Ergin, Marco Gruteser, Richard P. Martin, Jie Yang, and Yingying Chen. Decode: Detecting co-moving wireless devices. In *MASS*, pages 315 – 320. IEEE, 2008.

[14] Gayathri Chandrasekaran, Mesut Ergin, Marco Gruteser, Rich Martin, Jie Yang, and Yingying Chen. Decode : Exploiting shadow fading to detect co-moving wireless devices. *Transactions on Mobile Computing (TMC)*, 8, no. 12, Dec 2009.

[15] Gayathri Chandrasekaran, Mesut Ergin, Marco Gruteser, Rich Martin, Jie Yang, and Yingying Chen. Empirical evaluation of the limits on localization using signal strength. In 6^{th} *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks(SECON)*, Rome, Italy, Jun 2009. IEEE.

[16] Gayathri Chandrasekaran, Tam Vu, Alex Varshavsky, Marco Gruteser, Richard P. Martin, Jie Yang, and Yingying Chen. Tracking vehicular speed variations by warping mobile phone signal strengths. In *Ninth Annual IEEE International Conference on Pervasive Computing and Communications(PERCOM)*, Seattle, USA, March 2011.

[17] Gayathri Chandrasekaran, Tam Vu, Alex Varshavsky, Marco Gruteser, Richard P. Martin, Jie Yang, and Yingying Chen. Vehicular speed estimation using received signal strength from mobile phones. In *Proceedings of* 12^{th} *ACM International Conference on Ubiquitous Computing, (UBICOMP)*, Copenhagen, Denmark, Sep 2010. ACM.

[18] Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith, and Alex Varshavsky. Practical metropolitan-scale positioning for gsm phones. In *UBICOMP*, LNCS, pages 225–242. Springer-Verlag, Sept 2006.

[19] Y. Chen, J. Francisco, W. Trappe, and R. P. Martin. A practical approach to landmark deployment for indoor localization. In *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, September 2006.

[20] Yingying Chen, Gayathri Chandrasekaran, Eiman Elnahrawy, John-Austen Francisco, Konstantinos Kleisouris, Xiaoyan Li, Richard P. Martin, Robert S. Moore, and Begumhan Turgut. Grail: A general purpose localization system. *Sensor Review, Special edition*, 28, No. 2:115 – 124, 2008.

[21] Yingying Chen, Francisco J.-Austen, Wade Trappe, and Richard P. Martin. A Practical Appalsoroach to Landmark Deployment for Indoor Localization. In *SECON*, 2006.

[22] Benjamin Coifman. Using dual loop speed traps to identify detector errors. In *Transportation Research Board*, pages 47–58, 1999.

[23] Benjamin Coifman. Improved velocity estimation using single loop detectors. In *Transportation Research Part A*, pages 863–880, Dec 2001.

[24] Andrea Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *RATFG-RTS '01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, page 82, Washington, DC, USA, 2001. IEEE.

[25] Youjing Cui and Shuzhi Sam Ge. Autonomous vehicle positioning with gps in urban canyon environments. *IEEE Transactions on Robotics and Automation*, 19(1):15–25, February 2003.

[26] Anind K. Dey. Context-aware computing: The cyberdesk project. In *AAAI 1998 Spring Symposium on Intelligent Environments*, pages 51–54. AAAI Press., 1998.

[27] Anind Kumar Dey. *Providing architectural support for building context-aware applications*. PhD thesis, Atlanta, GA, USA, 2000. AAI9994400.

[28] D.J.Dailey, F.W.Cathey, and S Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. In *Intelligent Transportation Systems*. IEEE, Jun 2000.

[29] S. Dulman, P. Havinga, A. Baggio, and K. Langendoen. Revisiting the Cramer-Rao Bound for Localization Algorithms. In *4th IEEE/ACM DCOSS Work-in-progress paper*, June 2008.

[30] Nathan Eagle and Alex Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 4(2):28–34, 2005.

[31] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.

[32] E. Elnahrawy, X. Li, , and R. P. Martin. The limits of localization using signal strength: A comparative study. In *Proceedings of the First IEEE International Conference on Sensor and Adhoc Communcations and Networks (SECON)*, pages 406–414, oct 2004.

[33] Eiman Elnahrawy, John-Austen Francisco, and Richard P. Martin. Adding angle of arrival modality to basic rss location management techniques. In *In Proceedings of IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, San Juan, PR, Feb 2007.

[34] Eiman Elnahrawy, Xiaoyan Li, and Richard P. Martin. The limits of localization using signal strength: A comparative study. In *SECON*, 2004.

[35] P. Enge and P. Misra. *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamuna Pr, 2001.

[36] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi. An empirically based path loss model for wireless channels in suburban environments. *IEEE J. Sel. Areas Commun.*, 17(7):1205–1211, jul 1999.

[37] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. Bayesian data analysis, 2004.

[38] M. Gudmundson. Correlation model for shadow fading in mobile radio systems. In *Electronics Letters*, page 2145âĂŞ2146, 1991.

[39] D. Gundlegard and J.M. Karlsson. Handover location accuracy for travel time estimation in gsm and umts. In *IEEE Intelligent Transportation Systems Conference(ITSC)*, March 2009.

[40] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom*, pages 70–84, New York, NY, USA, 2004. ACM.

[41] David Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, March 1995.

[42] Alfred O. Hero, Jeffrey A. Fessler, and Mohammad Usman. Exploring estimator bias-variance tradeoffs using the uniform cr bound. *IEEE Tr. Sig. Proc*, 44:2026–2041, 1996.

[43] A.O. Hero. A cramer-rao type lower bound for essentially unbiased parameter estimation. Technical Report 890 DTIC AD-A246666, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, 1992.

[44] J. Herrera and A.M.Bayen. Traffic flow reconstruction using mobile sensors and loop detector data. 87^{th} *TRB Annual Meeting*, 2008.

[45] Jeffrey Hightower, Roy Want, and Gaetano Borriello. Spoton: An indoor 3d location sensing technology based on rf signal strength. Technical Report UW CSE 2000-02-02, University of Washington, Department of Computer Science, February 2000.

[46] C. Hill and T. Kneisel. Portable radio antenna performance in the 150, 450, 800, and 900mhz bands outside and in-vehicle. *IEEE Transactions on Vehicular Technology*, 40(4):750–756, 1991.

[47] Baik Hoh, Marco Gruteser, Ryan Herring, Jeff Ban, Daniel Work, Juan-Carlos Herrera, Alexandre M. Bayen, Murali Annavaram, and Quinn Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys '08*, pages 15–28, New York, NY, USA, 2008. ACM.

[48] IEEE. IEEE Standard 802.11b - Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specications: High Speed Physical Layer(PHY) in the 2.4 GHZz Band, 1999.

[49] N. Jalden, P. Zetterberg, B. Ottersten, A. Hong, and R. Thoma. Correlation properties of large scale fading based on indoor measurements. In *IEEE Wireless Communications and Networking Conference(WCNC)*, March 2007.

[50] Tamer Kahveci, Ambuj Singh, and Aliekber GÃijrel. Similarity searching for multi-attribute sequences. In *In Proc. of SSDBM*, pages 175–184, 2002.

[51] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. PTR Prentice-Hall, Englewood Cliffs, NJ, 1993.

[52] Eamonn J Koegh and Michael J. Pazzani. Derivative dynamic time warping. In *SDM*, 2001.

[53] H. Kremo, J. Lei, I. Seskar, L. Greenstein, and P. Spasojevic. Characterization of the orbit indoor testbed radio environment. In *Proc. IEEE VTC-2007 Fall*, pages 946–950, October 2007.

[54] P. Krishnan, A. S. Krishnakumar, Wen-Hua Ju, Colin Mallows, and Sachin Ganu. A system for lease: Location estimation assisted by stationary emitters for indoor rf wireless networks. In *INFOCOM*, October 2004.

[55] J. Krumm and E. Horvitz. Locadio: inferring motion and location from wi-fi signal strengths. In *MOBIQUITOUS*, pages 4–13, Aug 2004.

[56] Branislav Kusy, Akos Ledeczi, and Xenofon Koutsoukos. Tracking mobile nodes using rf doppler shifts. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 29–42, New York, NY, USA, 2007. ACM.

[57] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Guillaume Marceau, Lydia E. Kavraki, and Dan S. Wallach. Robotics-based location sensing using wireless ethernet. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 227–238, New York, NY, USA, 2002. ACM.

[58] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Netw.*, 43(4):499–518, 2003.

[59] Kaisen Lin, Aman Kansal, Dimitrios Lymberopoulos, and Feng Zhao. Energy-accuracy trade-off for continuous mobile device location. In *MobiSys*, pages 285–298, 2010.

[60] David Madigan, Eiman Elnahrawy, Richard P. Martin, Wen-Hua Ju, P. Krishnan, and A.S. Krishnakumar. Bayesian Indoor Positioning Systems. In *Infocom*, 2005.

[61] Miklós Maróti, Péter Völgyesi, Sebestyén Dóra, Branislav Kusý, András Nádas, Ákos Lédeczi, György Balogh, and Károly Molnár. Radio interferometric geolocation. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 1–12, New York, NY, USA, 2005. ACM.

[62] MathWorld. Spearman's Rank Correlation Co-efficient. http://tinyurl.com/3ymsre, 2007.

[63] K. Muthukrishnan, M. Lijding, N. Meratnia, and Paul Havinga. Sensing motion using spectral and spatial analysis of wlan rssi. In *EuroSSC*, October 2007.

[64] C. Myers, L. Rabiner, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 28(6):623–635, 1980.

[65] P. Nepa, G. Manara, S. Mugnaini, G. Tribellini, S. Cioci, G. Albasini, and E. Sacchi. Differential planar antennas for 2.4/5.2 ghz wlan applications. In *IEEE Int. Symposium of Antennas and Propagation Society*, pages 973–976, July 2006.

[66] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS). In *GLOBECOM*, 2001.

[67] Vit Niennattrakul and Chotirat Ann Ratanamahatana. On clustering multimedia time series data using k-means and dynamic time warping. In *MUE*, pages 733–738, Washington, DC, USA, 2007. IEEE.

[68] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara. Accurate gsm indoor localization. In *UBICOMP*, Sept 2005.

[69] N. Patwari, J.N. Ash, S. Kyperountas, III Hero, A.O., R.L. Moses, and N.S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):55 – 69, 2005.

[70] Neal Patwari, Alfred O. Hero III, Matt Perkins, Neiyer S. Correal, and Robert J. OâĂŹdea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2137–2148, August 2003.

[71] Neal Patwari and Sneha K. Kasera. Robust location distinction using temporal link signatures. In *Proc of ACM MOBICOM*, September 2007.

[72] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 32–43, Aug 2000.

[73] C.R. Rao. Minimum variance and the estimation of several parameters. *Proc. of Cambridge Philos. Soc.*, pages 280–283, 1946.

[74] Theodore S. Rappaport and Theodore Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, 2001.

[75] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Proc. IEEE WCNC*, volume 3, pages 1664–1669, March 2005.

[76] Rijsbergen and C. J. Van. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 1979.

[77] Zhigang Rong and Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[78] Teemu Roos, Petri Myllymäki, and Henry Tirri. A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, 1(1):59–69, 2002.

[79] B. Rutten and R. Schafer. The connected car. In *TOMTOM Mobility Solutions Whitepaper*, June 2007.

[80] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26:43–49, 1978.

[81] R. Sankar and L. Civil. Traffic monitoring and congestion prediction using handoffs in wireless cellular communications. In *IEEE VTC*, pages 520–524, 1997.

[82] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE Computer Society, 1994.

[83] Asim Smailagic, Daniel P. Siewiorek, Joshua Anhalt, David Kogan, and Yang Wang. Location sensing and privacy in a context-aware computing environment. *IEEE Wireless Communications*, 9:10–17, 2001.

[84] Timothy Sohn, Alex Varshavsky, Anthony LaMarca, Mike Y. Chen, Tanzeem Choudhury, Ian Smith, Sunny Consolvo, Jeffrey Hightower, William G. Griswold, and Eyal de Lara. Mobility detection using everyday gsm traces. In *UbiComp*, pages 212–224, September 2006.

[85] Timothy Sohn, Alex Varshavsky, Anthony LaMarca, Mike Y.Chen, Tanzeem Choudhury, Ian Smith, Sunny Consolvo, Jeffrey Hightower, William G. Griswold, and Eyal de Lara. Mobility detection using everyday gsm traces. In *Proceedings of the Eighth International Conference on Ubiquitous Computing (Ubicomp 2006)*, Lecture Notes in Computer Science, pages 212–224. Springer-Verlag, September 2006.

[86] Kannan Srinivasan and Philip Levis. Rssi is under appreciated. In *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*, May 2006.

[87] James H. Stapleton. Linear Statistical Models. Wiley Series in Probability and Statistics, 1995.

[88] S Thajchayapong, W. Pattara-atikom, N. Chadil, and C. Mitrpant. Enhanced detection of road traffic congestion areas using cell dwell times. In *IEEE Intelligent Transportation Systems Conference(ITSC)*, Toronto, Ont., Sept 2006.

[89] Arvind Thiagarajan, Lenin Ravindranath Sivalingam, Katrina LaCurts, Sivan Toledo, Jakob Eriksson, Samuel Madden, and Hari Balakrishnan. Vtrack: Accurate, energy-aware traffic delay estimation using mobile phones. In *SenSys*, Berkeley, CA, Nov 2009.

[90] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory. Part I: Detection, Estimation, and Linear Modulation Theory*. John Wiley and Sons Inc, New York, 1968.

[91] Alex Varshavsky, Anthony LaMarca, Jeffrey Hightower, and Eyal de Lara. The skyloc floor localization system. In *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2007)*, pages 125–134. IEEE Computer Society Press, March 2007.

[92] Roy Want, Andy Hopper, Veronica Falcao, and Jonathon Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, jan 1992.

[93] C. Xiao, K. D. Mann, and J. C. Olivier. Mobile speed estimation for tdma-based hierarchical cellular systems. In *IEEE VTC*, pages 2456–2460, Sept 1999.

[94] Moustafa Youssef, Ashok Agrawal, and A. Udaya Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 143–150, March 2003.

[95] Moustafa Youssef and Ashok Agrawala. The horus location determination system. *Wirel. Netw.*, 14(3):357–374, 2008.

[96] X. Zhao, J. Kivinen, P. Vainikainen, and K. Skog. Propagation characteristics for wideband outdoor mobile communications at 5.3 ghz. In *IEEE Journal On Selected Areas In Communications*, April 2002.

[97] Yilin Zhao. Standardization of mobile phone positioning for 3g systems. *IEEE Communications Magazine*, pages 108–116, 2002.

[98] Yahong Rosa Zheng and Chengshan Xiao. Mobile speed estimation for broadband wireless communications over rician fading channels. *Trans. Wireless. Comm.*, 8(1):1–5, 2009.

Vita

Gayathri Chandrasekaran

Education

- PhD., Computer Science, WINLAB Rutgers University, 05/2011
- M.S., Computer Science, WINLAB Rutgers University, 01/2009
- Graduate Student, Computer Science, Ohio State University, 09/2004-06/2006
- MSc.(Tech) Information Systems, Birla Institute of Technology and Science(BITS), Pilani, 06/2004

Experience

- Instructor for Operating Systems, 01/2011-05/2011
  Department of Computer Science, Rutgers University
- Graduate Teaching Assistant, 09/2009-12/2010
  Department of Computer Science, Rutgers University
- Summer Intern, 05/2009-08/2009
  AT&T Labs, Florham Park, NJ
- Summer Intern, 05/2007-08/2007
  Nokia Research Center, Palo Alto
- Graduate Research Assistant, 09/2006-05/2009
  WINLAB, Rutgers University

Publications

- Gayathri Chandrasekaran, Tam Vu, Alex Varshavsky, Marco Gruteser, Richard P. Martin, Jie Yang, and Yingying Chen. Tracking vehicular speed variations by warping mobile phone signal strengths. In Ninth Annual IEEE International Conference on Pervasive Computing and Communications(PERCOM), Seattle, USA, March 2011.
- Gayathri Chandrasekaran, Tam Vu, Alex Varshavsky, Marco Gruteser, Richard P. Martin, Jie Yang, and Yingying Chen. Vehicular speed estimation using received signal strength from mobile phones. In Proceedings of 12th ACM International Conference on Ubiquitous Computing, (UBICOMP), Copenhagen, Denmark, Sep 2010. ACM.

- Gayathri Chandrasekaran, Mesut Ergin, Marco Gruteser, Rich Martin, Jie Yang, and Yingying Chen. Decode : Exploiting shadow fading to detect co-moving wireless devices. Transactions on Mobile Computing (TMC), 8, no. 12, Dec 2009.

- Gayathri Chandrasekaran, John-Austen Deymious, Vinod Ganapathy, Wade Trappe, and Marco Gruteser. Detecting identity spoofs in 802.11e wireless networks. In GLOBECOM, Hawaii, USA, December, 2009. IEEE.

- Gayathri Chandrasekaran, Mesut Ergin, Marco Gruteser, Rich Martin, Jie Yang, and Yingying Chen. Empirical evaluation of the limits on localization using signal strength. In 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks(SECON), Rome, Italy, Jun 2009. IEEE

- Gayathri Chandrasekaran, M.A. Ergin, Marco Gruteser, Richard P. Martin, Jie Yang, and Yingying Chen. Decode: Detecting co-moving wireless devices. In MASS, pages 315-320. IEEE, 2008.

- Yingying Chen, Gayathri Chandrasekaran, Eiman Elnahrawy, John-Austen Francisco, Konstantinos Kleisouris, Xiaoyan Li, Richard P. Martin, Robert S. Moore, and Begumhan Turgut. Grail: A general purpose localization system. Sensor Review, Special edition, 28, No. 2:115-124, 2008.

- Gayathri Chandrasekaran, Mesut Ergin, Marco Gruteser, and Rich Martin. Bootstrapping a location service through geocoded postal addresses. In 3rd Intl. Symposium on Location- and Context-Awareness (LoCA, held with UBICOMP), Germany, Sep 2007. IEEE.

- High Throughput MAC Layer Multicasting over Time-Varying Channels, Ai Chen Gayathri Chandrasekaran, Dongwook Lee, and Prasun Sinha, Elsevier Computer Communications (COMCOM) , Volume 32, Number 1, pp 94-104, Jan. 200

- Dongwook Lee, Gayathri Chandrasekaran, Mukundan Sridharan, and Prasun Sinha. Association management for data dissemination over wireless mesh networks. Elsevier Computer Networks, 2007.

- HIMAC: High Throughput MAC Layer Multicasting in Wireless Networks, Ai Chen Gayathri Chandrasekaran, Dongwook Lee, and Prasun Sinha, IEEE MASS, Oct. 2006.

- Dongwook Lee, Gayathri Chandrasekaran, and Prasun Sinha. Optimizing broadcast load in mesh networks using dual association. In Workshop on Wireless Mesh Networks (WiMesh). IEEE, Sep 2005.