Thesis for the Degree of Doctor of Philosophy

Rapid Development of Flexible and Custom-resolution Indoor Location Systems

by

Uzair Ahmad

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Republic of Korea

August, 2008

Thesis for the Degree of Doctor of Philosophy

Rapid Development of Flexible and Custom-resolution Indoor Location Systems

by

Uzair Ahmad

Supervised By

Prof. Sungyoung Lee, Ph.D.

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Republic of Korea

August, 2008

Rapid Development of Flexible and Custom-resolution Indoor Location Systems

Uzair Ahmad

Submitted to
the Faculty of the Graduate School of Computer Engineering
in Partial Fulfillment of the Requirements
for the Degree of
Ph.D.

Thesis Committee:

Prof. Chae, Oksam. Ph.D. (Chairman)

Prof. Park, Chong Kuk. Ph.D.

Prof. Kim, Tae-Seong. Ph.D.

Prof. Lee, Young-Koo. Ph.D.

Prof. Lee, Sungyoung. Ph.D.

Prof. Brian J. d'Auriol. Ph.D.

Dedicated to my Family

Acknowledgements

Without the support of a large number of people, writing this thesis would be a much more difficult. I would like to thank all of them especially:

- My research supervisors and mentors Professor Sungyoung Lee, Professor Andrey Gavrilov, Professor Young-Koo Lee and Professor Brian J. d'Auriol, for reading this thesis and giving valuable feedback and suggestions throughout my research. I an indebted to Professor Sungyoung Lee for giving me the opportunity to work in UC Lab and allowing me to independently work on my areas of interest. I especially feel obliged to Professor Andrey for guiding my research on a minute scale and Dr. Brian for extending his precious time and expertise to refine my research and writing skills. I am also grateful to Dr. Arshad Ali for always providing me his insightful guidance and support.

- Our *'Halmoni'* Mrs Lee, Young-Seob for all the love and care she generously extended to Hamna and us. All her support substantially helped me to focus on my research.

- All my friends and colleagues for making the time at Kyung Hee as good as it was. I especially owe a lot to the support and help of Umar Kalim, Kamrul Hassan, Hyowon Hong, Song Shi Heon, Tahir Rasheed, Bilal Ahmad, Obaid Amin, Shaikh Riaz Ahmad, Raazi, Shoaib, Hassan Jameel, Faraz Rasheed, Jehad Sarkar, Julius, Mamun, Saad, Anjum, Cho Seong-jin, NQ Hung, Nakyong Yang, Young-Koo Han, Seo Sang-jin and last but not least our Lab Assistant Mrs Seoungae Kim.

Abstract

Location determination of mobile devices can enable several applications in the modern Ubiquitous Computing environments. Recently, the IEEE 802.11 standard network based location determination systems have been emerging as an effective and economical indoor location technology. The existing approaches for developing such location systems require a laborious phase of sensory data collection to develop a radio map of the target site. The radio map is used to train pattern classification models for mapping the signal strengths to certain locations. However, the localization capability of such classification models is faced with the prevalent noise and temporal fluctuations in the signal measurements due to indoor multi path propagation effects. These fluctuations may cause a high degree of overlap in the neighboring points of a signal space resulting in location estimation errors. Another limitation of previous approaches is the rigidity of the underlying location model. Such methods distribute the target environment into an inflexible and monolithic Euclidean space or grid of cells. The resulting location determination system reports location of a mobile terminal as (x,y) coordinates of the cell containing the device. However, many indoor location based applications require: i) semantically meaningful location information instead of coordinates, and ii) customizable resolution of the location model.

This thesis proposes an integrated middleware infrastructure, named Ababil, to realize indoor location systems. Ababil (arabic name of Swallow) is known to have the location-awareness [36] and location based adaptability [35] faculties akin to the ideas presented in

this thesis. The ababil bird can localize itself *on the fly* with the help of landmarks, e.g. mountains and rivers, as well as the magnetic field of earth. Similarly, the Ababil middleware utilizes radio beacons and their signal strength signatures for mobile device localization. A new rapid development life cycle is introduced along with the tools and subsystems to support each phase of this cycle such as: i) sensor data collection and analysis system, ii) pattern recognition methods for custom resolution and flexible location determination system, and iii) a reflective component framework for delivering location based services to the mobile stations.

A key result of this research is the improvement in overall accuracy of the system by utilizing the temporal signal detect-ability information. A novel methodology is proposed to incorporate this information into the design of a pattern classifier. This approach not only improves the localization performance but also enables easy manageability of a large scale complex system. This method is implemented as an integrated Radio Map Data Analysis and Decision Support System which provides automated support for sensory data preprocessing, visualization, and modularization and classification complexity analysis. On top of the modular approach, an online, incremental learning method, ConSelFAM, is proposed which does not require creation of a radio map; thus reducing the time to develop a system. This method can learn to classify the input sensory data to the corresponding locations in real time without any lab time training and evaluation phase. Besides the fine resolution localization, a new methodology is presented for development of coarse resolution, semantically meaningful location systems in multi-floor buildings. This methodology is composed of an Election algorithm for real time discovery of location-to-signal associations and a novel signal-to-location translation model. It is demonstrated via extensive experiments in real life environments that the proposed methods provide significantly better accuracy compared to state-of-the art location determination systems.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Emerging mobile computing application and ubiquitous computing environments are required to provide location sensitive behavior to the end users. Consequently location systems are on the verge of becoming a part of daily lives of the end users in smart environments like shopping malls, campuses and airports. Advances in sensing technologies have resulted in several novel positioning techniques to support many potential location based applications. Location aware computing space is defined mainly by these three complementary technologies as shown in Figure 1.1. Until recently, the indoor localization has been subject to costly infrastructure and special hardware devices mounted on the objects of interest [62, 63, 53]. However, signal strength based location estimation is an attractive choice because of its economic viability and pervasive availability of Wireless LANs i.e. WiFi. Such indoor positioning systems can enable several location based applications in future ubiquitous computing environments. Although such location awareness is desirable, but the accuracy of such systems have been a subject of intensive research [30, 51, 13, 12, 34, 37].

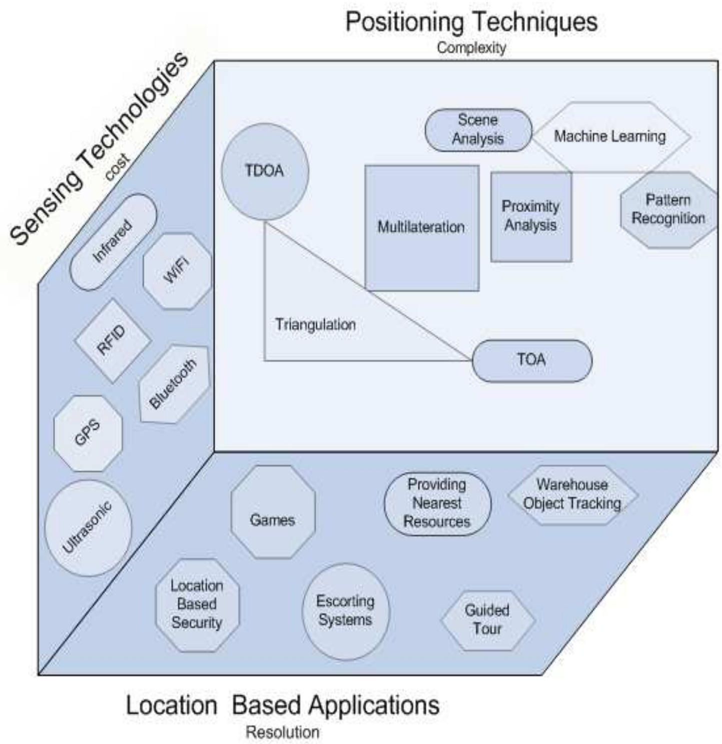


Figure 1.1: Location Aware Computing Space

1.1 Signal Strength Based Positioning: Basic Concept

The fundamental assumption underlying RSS based location estimation is that received signal strengths of different access points, or signal sources, exhibit complex but distinguishable patterns at a particular location. If these patterns could be captured at each location then pattern recognition machines can be trained to learn the relationship between RSS fingerprints and each target location. From a pattern recognition standpoint, a particular set of access points constitute n-dimensional input space which is often referred to as a *Radio Map*(see for e.g. [26, 30, 64]). Suppose that n access points define a signal space R^n which covers target location space L, then this relationship can be represented as

$$F : R^n \rightarrow L \tag{1.1}$$

More specifically,

$$l_j = f(x^n) \tag{1.2}$$

where $l_j \in L$ and $x^n \in R^n$.

The creation of the *Radio Map*, also called *Site Calibration*, involves capturing this information and storing observed signal strengths vectors in a data store as shown in Fig 1.2.

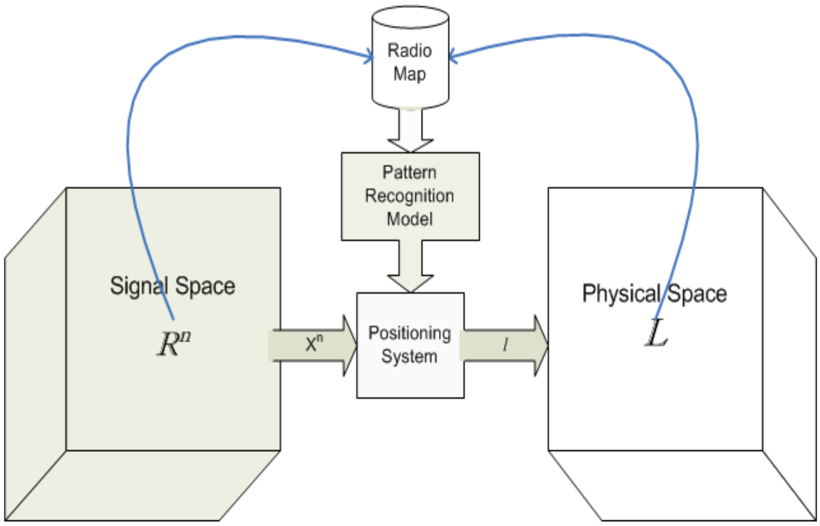


Figure 1.2: Basic Concept: the relationship between signal and location space

Once the *Radio Map* feature space is created, it is used to develop a mapping function, given in Equation 1.2; between j^{th} target location l_j and respective signal strength patterns. This function is later employed to estimate location of a device with given RSS values.

1.2 Motivation and Goals

The main focus of this research is to provide end-to-end middleware support for indoor location aware computing. This support includes providing novel positioning methods to overcome limitations of existing approaches as well as a framework to develop location aware adaptive applications.

As the optimality and feasibility of location estimation using radio signal of modern wireless network, especially IEEE 802.11, is well established [61, 64]. This provides ample opportunity for location aware computing to become pervasive because of no special requirement of the infrastructure which was conventionally perceived as a bottleneck. However the realization of such location systems faces sever limitations in terms of scalability, extendibility, repair-ability and long development time. Secondly, the metric definition of sense of location is not required by many location dependant applications. Instead the definition of *location* is often determined by the semantics of target environment. On the positioning end, this thesis strives to device efficient positioning methods complacent to these requirements.

In addition to the development of positioning methods, the service layer of location aware computing puts forward a core requirement to the end user applications, which is the *adaptability*. The location aware applications and environments are expected to adapt and reconfigure their behaviors in order to achieve certain contextual goals. However, this adaptability incurs prohibitive computational resources and time. To this end, an efficient and generic adaptive framework is presented to lower the time and computational costs of behavior switching.

1.3 Scope and Methodology

Indoor Location Aware Computing research spans three main areas: i) systems support, ii) Infrastructure, and iii) positioning methods. Here a succinct description of the scope of the proposed Ababil is provided to prepare the didactic design of the rest of the thesis. A general research taxonomy is shown in Figure 1.3 in which thick-bordered blocks are the exact topics where contributions of this thesis lie. A prime concern of Ababil is to integrate previously disparate topics into a coherent middleware infrastructure. In this taxonomy, the filled arrow heads indicate the sub topics and empty arrow heads indicate the targeted integration among related topics.

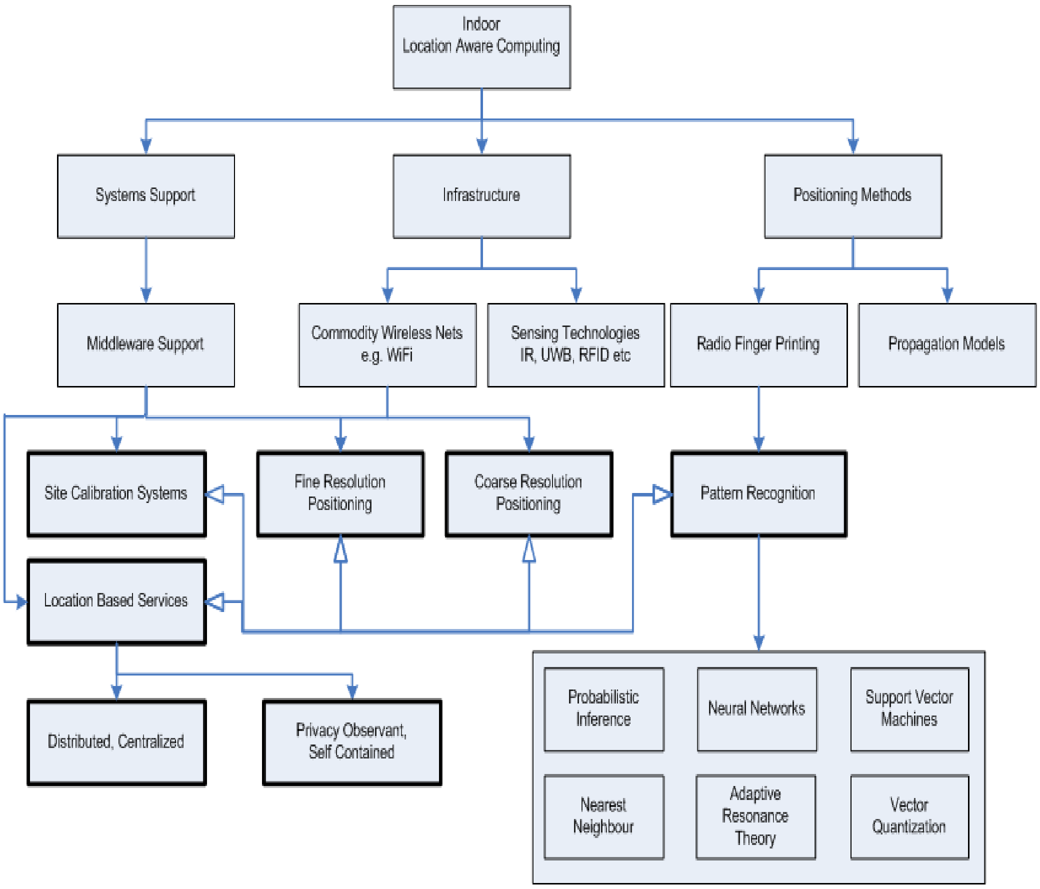


Figure 1.3: Indoor Location Aware Computing Research Taxonomy

Modularity is an effective and natural solution to overcome complexity. It simplifies a complex system by enhancing the understanding of interdependencies and dynamics of the system. Besides, it allows systematic growth in system and localize the reconditioning and repair. It is hypothesized that it can bring these benefits to location systems as well. Research methods applied in this thesis are:

- Literature surveys and reviews.
- Qualitative comparison of classification and prediction methods and algorithms based on their declared and/or determined properties.
- Quantitative comparisons of methods and algorithms based on quality measures.
- Design of models and architectures based on requirements and results obtained from literature studies.
- Design and implementation of algorithms and protocols.
- Evaluation of the proposed concepts and methods via prototype implementations.

1.4 Thesis Contributions and Significance

The contributions this thesis together with its significance in the context of contemporary Signal Strength based Positioning systems are as follows:

- Due to the multi-path, Non Line of Sight propagation effects: a temporal or permanent loss of radio signals at some areas is a common occurrence. The Ababil incorporates this information into the design of the classification method such that the overall location recognition performance improves significantly compared with other systems. Ababil discovers subsets of all access points which are maximally-covering as well as separating the target locations in term of Visibility Clusters. These clusters guide the design of the location estimation system as detailed in the Chapter 3. This approach reduces the error in estimated compared with previous approaches as well as facilitate the development process of a large scale location system.
- Ababil incorporating online and incremental learning classification to reduce the development time of a location systems. A modified version of Adaptive Resonance

Theory based pattern recognition system, Fuzzy ArtMap is proposed for this purpose. In this system, the sense of sub-spaces in the Radio Map feature space is modeled as a Context-field. Resulting learning system provides comparable accuracy and, on top of that, the capability to develop a location system without the prior requirement of a radio map. This capability enables a rapid development life cycle in contrast to the conventional ones.

The conventional approaches to Signal Strength based Positioning follow a laborious development life cycle. In fact, the accuracy of location estimates is directly related to the amount of signal samples which require intensive human effort. This thesis provides a rapid development life cycle which not only reduces the time to develop a positioning system but also provides competitive accuracy compared to other systems.

- The discovery of maximally-covering and separating subspaces is realized via a brief site survey instead of a detailed Radio Map. Thus, the output of this survey becomes an input to the above mentioned systems as well as provide a region level location information to several applications such as group oriented location based services. This thesis provides a new methodology to achieve both objectives. This capability eliminates the limitations of a rigid, inflexible location model from location system development by allowing the target applications to define the locations and areas according to their semantic requirements.

  The requirement of location information may vary from application to application [2]. However, the grid-based location model of the existing positioning systems is rigid and monolithic; targeting specific applications. This is because of the sequential nature of the classic development life cycle, which forces creation of the location space long before an application is developed. The Ababil middleware enables creation of a flexible location model which is created as part of application development process.

- The Ababil middleware infrastructure puts forth a new system development life cycle which guides the developers through well defined development phases and provide support of realizing the concepts into actual system. The rapid development life cycle is based upon application driven methodology contrary to the conventional approaches. It enables on demand, incremental location system development and an integrated component framework for performing the development activities.
- An end-to-end development support is provided for realizing an indoor location-aware computing system. For each phase of the development life cycle, the Ababil offers tools, components and design patterns for implementing the deliverables for next phase. These include: i) a distributed, collaborative sampling system, ii) a radio map data engineering toolkit for analyzing the data characteristics and selection of appropriate classifiers, iii) component libraries for embedding variety of classifiers as objects into the applications, and iv) reflective component framework for adaptivity in location-aware applications.

1.5 Results Evaluation

Within the scope of this thesis, a conceptual framework for applying pattern recognition methods to location systems has been developed, based on the idea of capturing the unique patterns the wireless signal exhibit on different locations. By recognizing these patterns and analyzing the device trajectory, it is possible to localize and track mobile objects in the indoor settings. This general concept has been refined into a flexible, custom resolution location system development middleware for ubiquitous computing environments. It is based on six steps: sensor data acquisition, feature extraction, preprocessing, labeling and classifier training and localization. In these steps, recognized location information is inferred from low-level received signal strength sensor data. To demonstrate its flexibility and suitability for wide range of location based applications, it has been implemented in a component-based, layered architecture which supports both distributed as well as self-contained applications.

An evaluation of the middleware and its implementation has been done with an extensive sensor data set collected in a realistic scenario. For both the training and testing step, different methods have been compared quantitatively on the same data set to select the best algorithms for this scenario. Because not all of those algorithms are currently available as modules in the framework, some have been evaluated using Matlab and other available toolkits. Most of the methods have been implemented using Matlab and then componentized in C#. A real location based application, The Smart Meeting Manager, was also developed on top of the Ababil. It incorporates locations of the 'presenters' to manage a conference meeting. Upon arrival of the meeting chair and a presenter, it launches the appropriate power point presentations and send notifications to the interest attendees who are not present in the meeting.

1.6 Organization

The proposed middleware and subsystems presented in this thesis are coherently interfaced with each other. In fact, the organization of this thesis is guided by the default interconnection of the subsystems involved in Ababil. A holistic architecture of the Ababil Middleware is shown in Figure 1.4. In the following a precise description is provided regarding the individual subsystems and how they are interfaced with other.

Location Based Applications In a Location-aware Computing system, the indoor applications reside at the top layer to provide the location-driven services to the end users. Architecturally, these applications can be deployed on mobile devices or on high end server machines in a distributed manner. In both cases, these application are interfaced with the

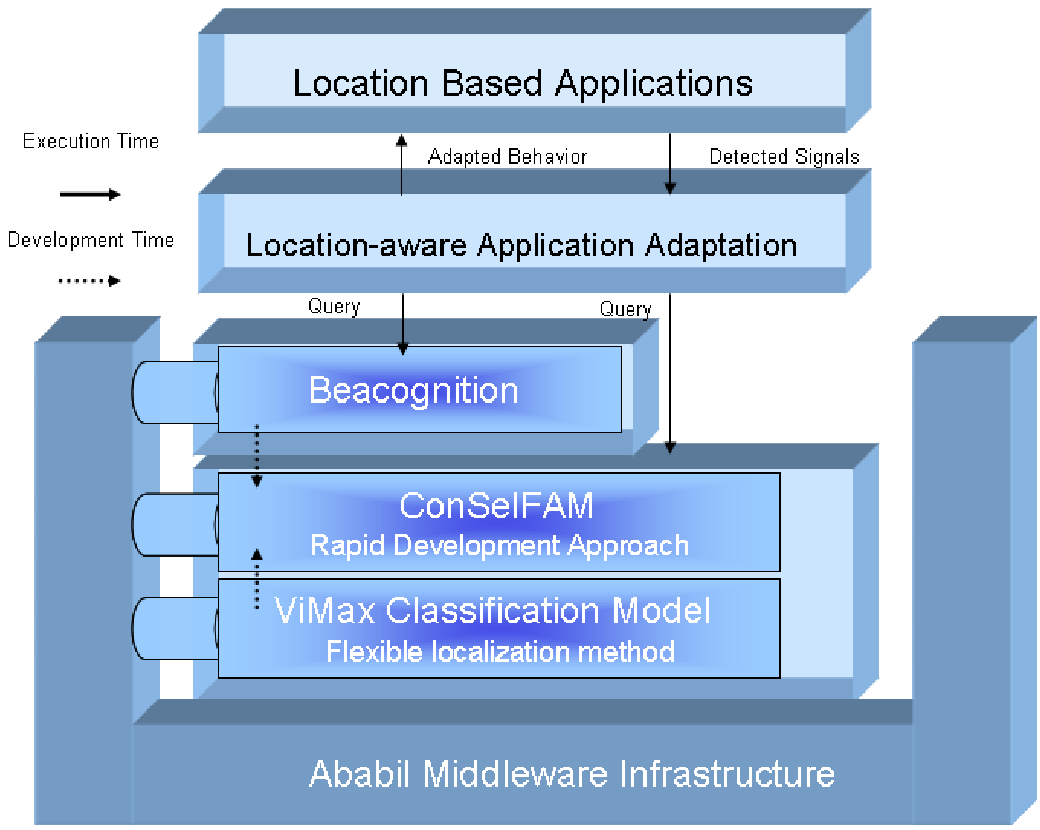


Figure 1.4: Global Architecture of the Ababil Middleware Infrastructure

middleware subsystem via a simple protocol of seamlessly feeding signal reading to the lower layers and, depending upon the response, actuating appropriate functions for the user.

Location-aware Application Adaptation A primary capability of any Location-based application is to autonomously adapt its behavior depending upon the location of carrier mobile device. This layer can also be deployed on single machine or on server. Besides adaptation, its responsibilities include to broker the location application and remotely deployed positioning systems by hiding networking details and, if necessary, users identity.

The components residing on this layer receive signal readings from application layer and pass them down to the appropriate positioning systems in terms a 'Location Query'. Upon receiving the reply of this query, the adaptation takes place [10]. The requirement of location information can vary for different application. These requirements can be divided into two main categories as follows.

Fine Resolution Positioning The fine resolution positioning system resoles a location query into a metric sense of location. Ababil's realization of this requirement has two further facets. The components of this layer are interfaced with the application adaptation layer for exchanging the signal readings, e.g Location Query, and resulting Location information.

The ViMax Model: The ViMax model for localization provides a novel methodology to improve the localization capability of the classification machines. This model is applied to a variety of pattern classification methods to evaluate the generalizations of the performance. As detailed in Chapter 2, this model can effectively resolve a location query into (x,y) coordinate of querying mobile device. However, like other location fingerprinting based positioning methods, the ViMax model requires extensive site calibration, preprocessing and provides location information in terms of coordinates in a rigid Euclidean location space.

The ConSelFAM Model: The ConSelFAM system enables radio map free, online, incremental learning methodology for developing positioning systems. The Chapter 3 presents this methodology as well as its additional advantages. Similar to the ViMax model, it receives location query from adaptation layer, resolves the query and provides the location information back. As is demonstrated in Chapter 3, incorporating the ViMax model improves the localization performance. However, contrary to the aim of rapid development, the ViMax requires the radio map prior to the learning starts.

Low Resolution Positioning Besides the fine resolution positioning methods, Ababil provides a coarse resolution but semantically meaningful location awareness scheme, namely *Beacognition*, which is presented in the Chapter 4. The Beacognition also have similar interfaces with the upper layer as fine resolution positioning layer. However, it does not impose any special structure on the location query and provides back a region-grained location information.

Chapter 2

Related Work

Most of location aware computing middleware research differs with respect to target application scenarios but agrees on a common set of features that are required by a broad range of location aware applications. These features include support for several positioning techniques, trajectory tracking, publish subscribe support for spatial messaging and grouping and modeling and representation of physical space inside middleware. Middleware approach allows separation of developing location technologies from developing location based applications. Due to this logical exclusion most work in Location Aware Computing research has focused on location awareness in either of two dimensions i) Location estimation or positioning systems ii) Location based applications. This separation of concerns benefits in designing generic middleware architectures by using off the shelf location technologies. At the same time it limits the provision of integrated framework that allows both development of positioning system and location based services.

2.1 Location Systems

Recently, 802.11 wireless network based location systems have gained a significant attention from the research community [30, 32, 33, 65, 38, 8] as well as industry [26, 25]. This is mainly due to the pervasive availability of 802.11 in indoor environments and proliferation of wireless network enabled commodity hand held devices. The WiFi location systems can be broadly categorized based upon: i) the resolution of location estimate and ii) location inference method.

These system can also be divided based on the underlying location estimation methods present in both coarse grained and fine grained systems. The low resolution systems often incorporate simple inference methods such as k-nearest neighbors [60]. However the fine resolution systems employ more complex classifiers such as statistical modeling [38, 65, 33], vector quantization [50] or neural networks [6].

2.2 Fine Resolution Localization

There has been several efforts to develop RSS based location systems. Several pattern classification and machine learning methods e.g. bayesian classification and filtering[13], k-nearest neighbors[30],[51] , GPS like triangulation[57] and kalman filtering[37] have been employed for this purpose.

Nearest Neighbors based pattern recognition technique and its derivatives have been used in pioneer works on RSS based location estimation. Microsoft's RADAR system reported 2.65 meter distance error [30]. K. Pahlavan *et al.* used K-Nearest Neighbors and achieved 2.8 meter distance error[51]. Nearest Neighbor and its variants require a database of sample RSS readings at the estimation time for pattern matching. As the area and number of target locations grow, the size of the database dramatically increases prohibiting sufficient scalability.

Some research works have also employed GPS like triangulation method for location estimation. Asim et al achieved 4.5 meter location estimation error in a target area of 60 square meters[57]. Triangulation methods work on the assumption that signal strength decays only as function of distance of receiver device from sender access points. Nevertheless signal strength decay is a function of several factors of the indoor environments which undermine the validity of this assumption. Such deterministic localization techniques face difficulty while applied in the indoors. As noticed by [30, 51], the fluctuations in the RSS values make it hard to model the propagation and requires detailed environmental information.

Probabilistic approaches like Bayesian networks have also been employed for such systems but are computationally exhaustive and difficult to scale. Andrew et al reported 1.5 meter average distance error but only for 30 square meter area test bed [13]. As the area and number of target locations and wireless access points increase, the complexity of bayesian structures grow and become computationally expensive.

Battiti *et al.* have reported their research on using feed forward back propagation neural network on small scale (624 square meter area using 3 access points) location estimation system [12]. Learning Vector Quantization networks were used to develop location estimation system for 350 square meter area using 5 access points[50].

2.3 Coarse Resolution Localization

Coarse resolution location systems, such as Intel's Place Lab [26], provide 20 to 30 meter accuracy in the outdoor scenarios, whereas fine resolution location systems claim up to three meter accuracy in the indoor environments. The fine resolution systems model the physical space into geometrical grid [30] or topological cells [38]. Such systems require high density sampling, referred to as radio map, of target environment which provides the basis for developing a mapping function between physical space and signal space.

The BeaconPrint [40] employs GSM and 802.11 beacon response-rate signatures to learn and recognize places in an outdoor environment. The Nibble [32] system builds a bayesian network to capture the dependencies between signal to noise ratio (SNR) and corresponding locations. It supports incremental development of a system, however prior knowledge of beacons and training data is required for offline calculation of the conditional probabilities. The NearMe [45] system provides rapid development without radio map scheme. It is a proximity server which provides a list of neighboring devices based upon similarity of their beacon signatures. The SkyLoc [60] system specifically addresses floor recognition problem in multi story buildings. It improvises GSM (Global System for Mobile communications) beacons instead of 802.11 access points.

	Multi-floor	Offline Training	Environment al Modeling	Probabilistic	Deterministic	Coarse Resolution	Fine Resolution
Ababil	*			*		*	*
RADAR[1]		*		*			*
Smailgic[3]			*		*		*
NTT[6]		*		*			*
Near Me[7]			*		*	*	
Rice[5]	*	*		*			*
Place Lab[8]					*	*	
HORUS[11]				*			*
Cisco[12]	*	*	*		*		*

Figure 2.1: Related Work: a feature comparison

2.4 Middleware Support for Location Aware Computing

There have been several research efforts both in academia and research community to build novel positioning systems e.g. Infrared based [62], ultrasonic positioning [63]],[53],[59] and radio signal strength based [30]. Similarly, with respect to location based service development, several middleware solutions have been proposed to provide necessary services that support location based applications. Bohn presented middleware architecture for super distributed RFID tag infrastructure for location awareness [14]. This middleware abstracts underlying RFID infrastructure from applications and enable several interesting location sensitive applications but resulting systems is required to have dense tag deployment (39 per m2) and location aware devices need to have RFID reader and antenna mounted on them. Murphy et al used LIME middleware for Location Aware computing [49]. LIME is a coordination Middleware for mobile computing that helps decoupling behavior from communication in mobile applications. Authors discuss how physical context, particularly location, of mobile device can also be incorporated into LIME to support location aware applications. An interesting application is built on top of this middleware to track mobile crew in a disaster scenario. This work has limited scope as it supposes that location is already available, through GPS or other technology, and deals only with application aspect of location awareness in coordination middleware. Ying Chin et al presented a location operating reference model (LORE) middleware infrastructure for location aware computing [19]]. LORE facilitates fusion of multiple location sensors, tracking of moving objects and spatial queries. In later work, Authors integrated Location Based Service (LBS) Middleware with multimedia services to develop location based messaging application [20]. Gianpaolo et al introduced the idea of a location aware publish subscribe middleware in [21]. This middleware solution offers location based notifications to subscribers. Subscription can be made with respect to location of both subscriber and publisher and both in terms of absolute or relative locations of subscriber or publishers. Like previous solutions, focus of this middleware remains restricted to location information service floor. MiddleWhere proposed by Ranganathan et al provides middleware infrastructure that provides several desirable features including incorporation of multiple sensing technologies, spatial database representation of physical world and statistically handling temporal nature of location information [54]. MiddleWhere performs fusion of different location estimates from multitude of sensors and develops spatial probability distribution of target object's location. It employs desktop login, finger print logins, card swipes, RF badges and UbiSense ultra wide band location technology [59] to build location reasoning engine. Bottazzi et al presented middleware architecture Allocation and Group Aware Pervasive Environments (AGAPE) [28] for location aware environments. This middleware targets novel application of location awareness for dynamically grouping mobile objects according to their physical neighborhood. AGAPE provides primary constructs and artifacts to build location based groups in pervasive computing environments. Authors used WLAN based location technology for building proof of concept prototype on top of AGAPE middleware.

Chapter 3

A Modular Classification Model for Localization

3.1 Introduction

WiFi radio signals follow a complex propagation model because of multi-path effects. These are caused by building structure and environmental factors such as human activity and neighboring devices. These factors as well as the distance between transmitter and receiver contribute to the loss of a signal in certain areas. The signal availability of a particular access point at a given location is referred to as its *visibility* in this chapter. Previously several pattern classification methods have been reported, as explained in chapter 1, for RSS based indoor location determination. However most of the previous results consider a small scale location estimation problem. Nevertheless, scaling up the target area introduces the visibility issue such that signals of some access points are not detectable at certain locations. This phenomena introduces *missing* values in radio map feature space. This chapter discusses an alternative visibility modeling approach which effectively improves the RSS based location determination accuracy despite missing values.

3.1.1 Optimality of Signal Strength Based Positioning

The optimality of using signal strengths can be evaluated analytically as shown by Mechael [61]. Given a signal space of n access points, the set of signals x^n observed at distinct locations L is defined as $R^n = f(L) := \{f(l)\,|l \in L\}$. Thus, for any location l, a signal vector $X = [x_1, x_2, x_3 ... x_n]$ is available. Inversely, a set of locations where x^n is observed can be denoted as

$L = f^{-1}(R^n) := \{l|l \in L, f(l) \in R^n\}$.

The inverse image of X is the set of locations L_l where same signal is observed. As shown in Fig. 3.1, the elements of L_l are not distinguishable solely based upon the X. Therefore, for each X, the corresponding set of locations can then be used to estimate the distance error as suggested by [61].

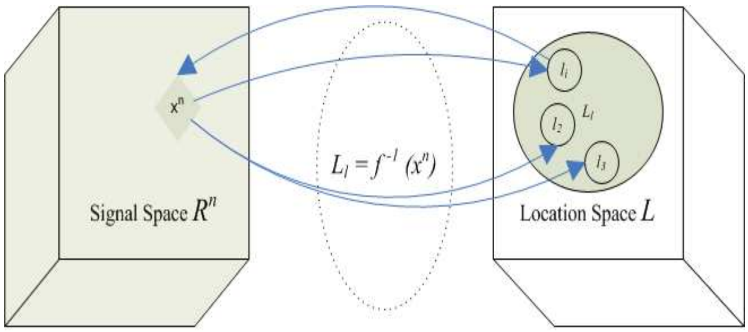


Figure 3.1: Functional Mapping Between Signal and Location Space

A significant characteristic of indoor radio wave propagation is the fluctuating nature of the signals due to multi-path effects. A deterministic model, as described above, of localization problem cannot provide applicable solution to the real life environments. In order to incorporate the fluctuations in signal strengths, the X is represented as

$\bowtie_v := \{(min, max)\,|min, max \in R^n, |min_i - max_i| \leq v, 1 \leq i \leq n\}$.

The **X** $\in R^n$ compliant to the $\bowtie_v$ will comprise all fluctuations observable at a location.

The inverse image of **X** then represents all those locations which are indistinguishable such that,

$L_l^{\bowtie_v} := \{p | p \in L, f(p) \bowtie_v f(l)\}.$

Fig. 3.2 illustrates the scenario of location error in case of signal fluctuations. The fluctuation model provides the worst case distance error estimate because it is supposed that all observable fluctuations are equally probable to take place. However, the fluctuations occur in a probabilistic fashion in the real environments. These propagation characteristics are discussed in the next section using large number of signals from a real environment.

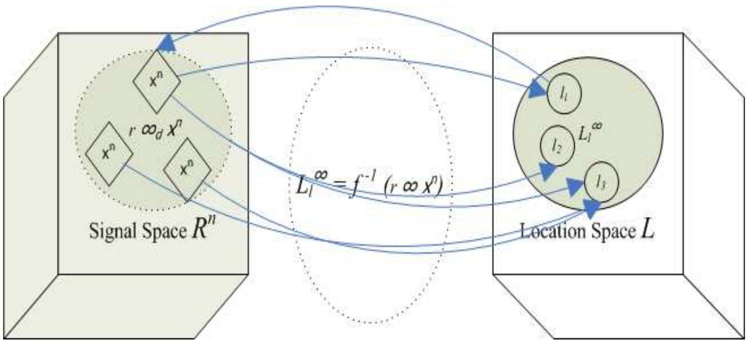


Figure 3.2: Mapping Between Signal Fluctuations and Locations

3.1.2 Real Environment Data Characteristics

After conceptualizing the expected location estimation error, it is worthwhile to analyze the characteristics of Signal Strength data in a real environment. An elongated Signal Strength measurement experiment is presented here to show the noisy and probabilistic nature of signals in an indoor environment. The experiment was conducted in a hall-like laboratory which is partitioned into small desk-cabins for twenty people. The environment reflects a typical office settings, obstructions and noise factors. At each location, signal strength of four access points were collected for four hours. The access points, signal sources, are deployed outside the lab. The used access points belong to a public network operator, KTF Korea Telecom, therefore their deployment can not be manipulated in any way to affect the signal measurements in the observation laboratory. Four points were designated, approximately 3 meters apart from each other, for taking the observations. At each location 36000 scans were performed for an extended period in working hours.

Fig. 3.5 shows the signal strength plots of each access point at different locations. The same experiment was repeated two more times after 30 days and 40 days respectively. These data plots illustrate the noisy nature of indoor radio wave propagation. The range of signal strength may vary even for a stationary device as wide as 20 40 dBm. These fluctuations form a major component of the error in the location estimates. From location estimation standpoint, some important observations can be noticed as follows:

- **Fluctuations and the Sources:** Despite the signal sensing device is stationary, the received signal strengths often fluctuate. This is due to the obstacles and noise prevalent in the indoors as well as the sensing hardware. Another important source of variation in the RSS is the distance between APs and the sensing device. Increasing the distance between signal source and receiver reduces the RSS even though the exact relationship between distance and RSS is concealed by the fluctuations.

- **Signal Strength Patterns:** It can be seen that at different locations the range of fluctuations changes, which means that at different locations the signal strengths form a different pattern. This interesting pattern can be observed via plotting their statistical properties such as mean and variance. Fig. 3.6 plots the variance of fluctuations observed at one location as vertical bars. Connecting the average signal strength of each bars show the underlying pattern which reflect the discriminatory information contained in the signal strengths.

- **Location Discriminating APs:** The really discriminating access points are fewer

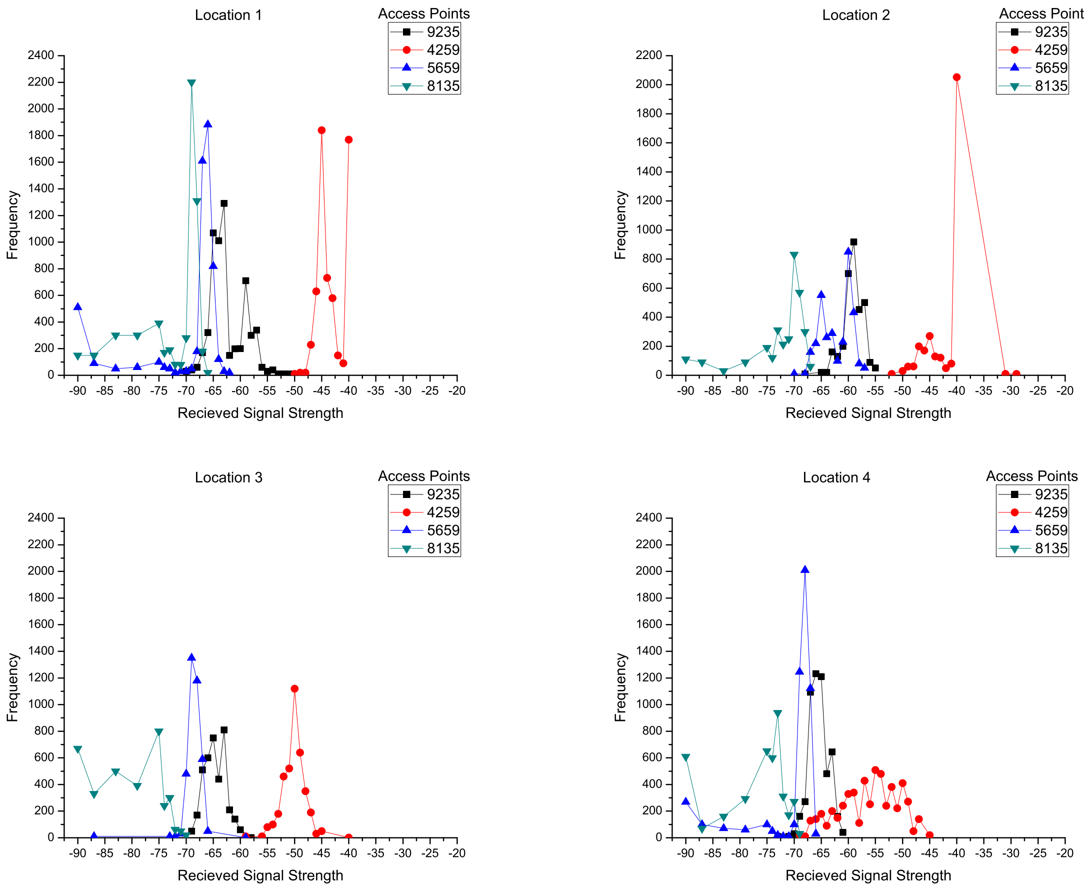


Figure 3.3: Day 1: Position of the access point 4259 is right outside the observation room. The other three access points are situated in adjacent corridors of the neighboring floors.

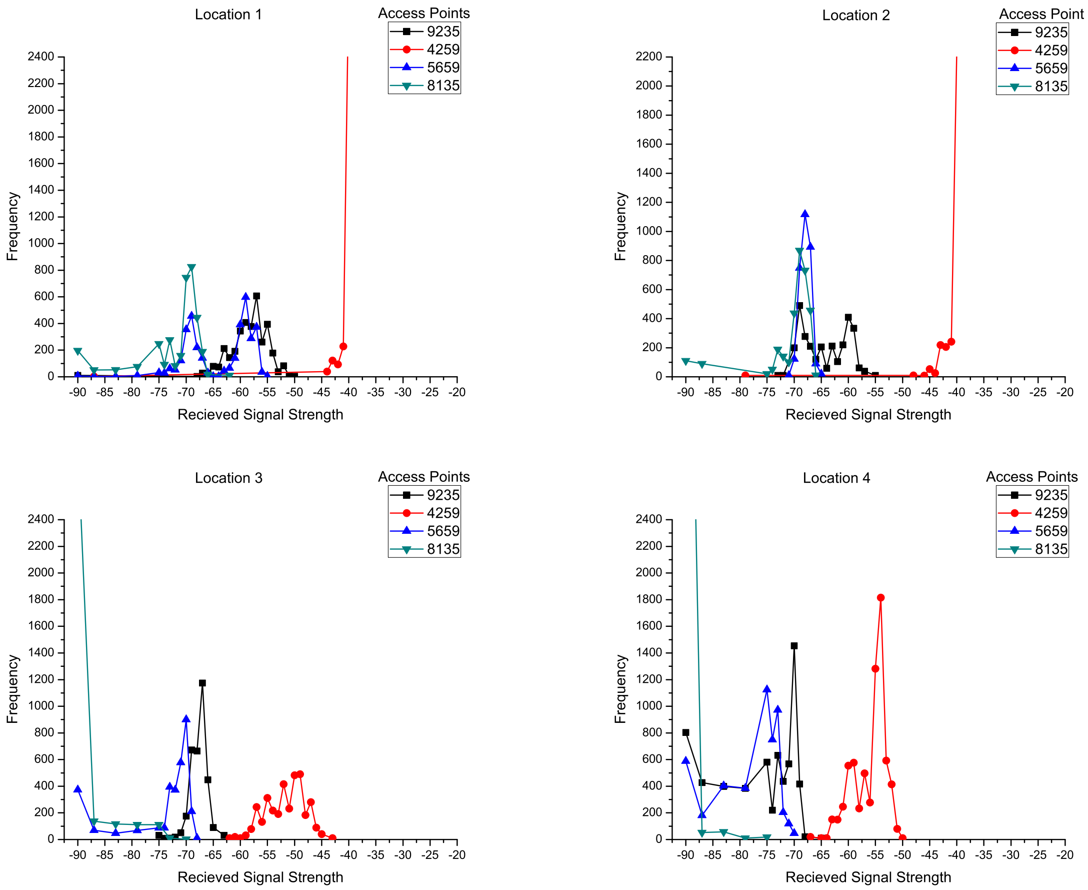


Figure 3.4: Day 2: Visualizing Signal Strengths of WiFi APs in a Real Environment.

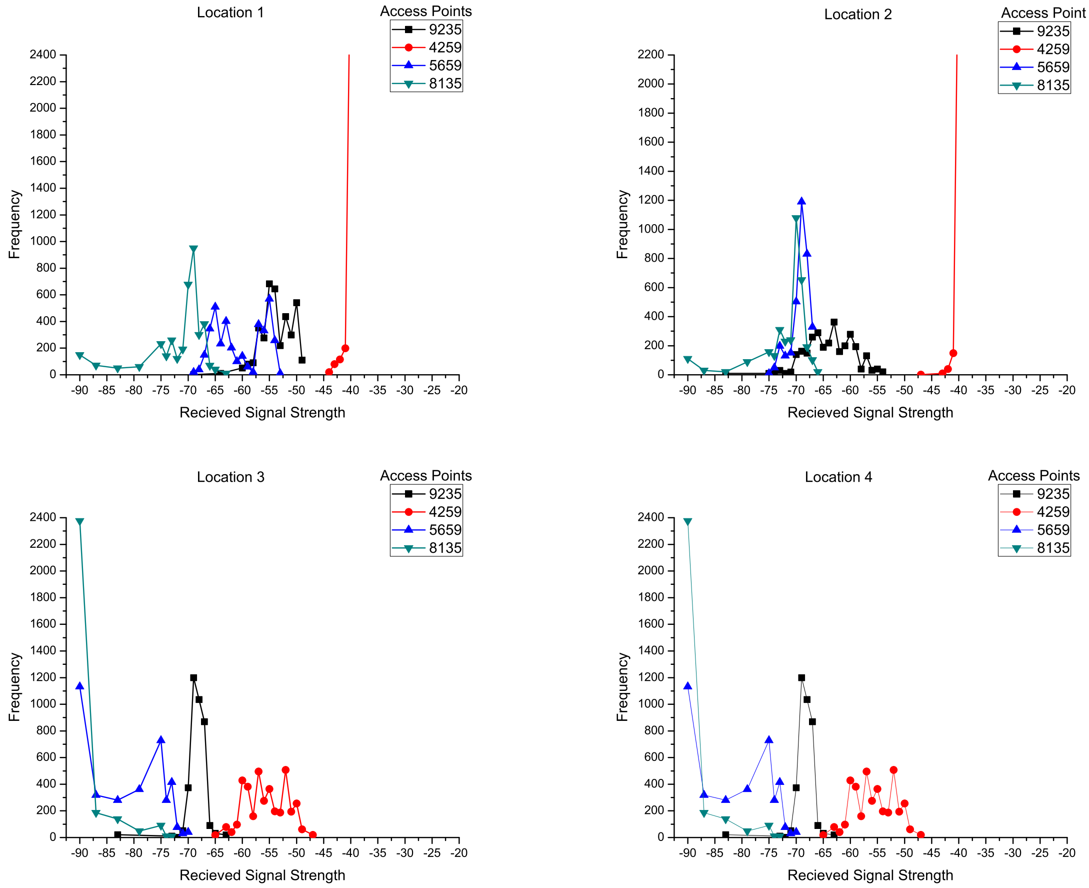


Figure 3.5: Day 3: Visualizing signal strengths of WiFi APs in a real Environment.

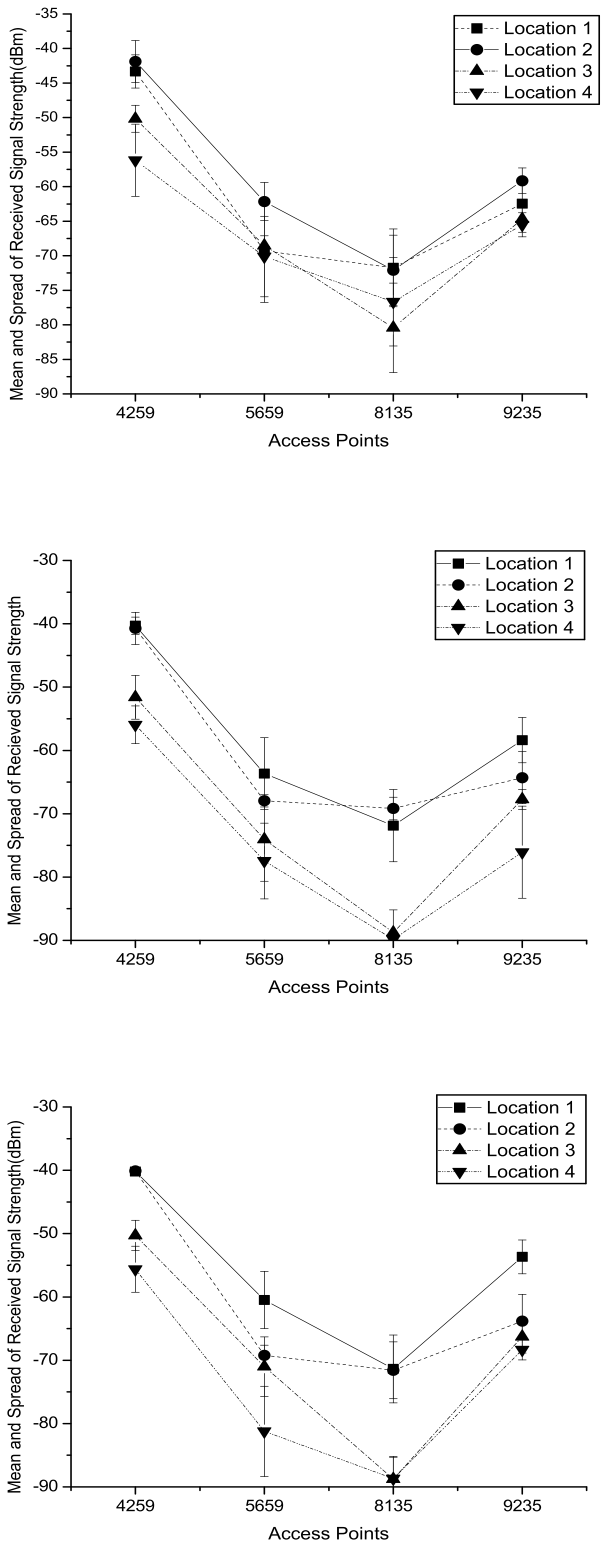


Figure 3.6: Three graphs showing Received Signal Strength patterns on three different days. Each graph shows the pattern as mean and spread of signal strengths (Y-Axis) created by four APs (X-Axis) at four neighboring locations in a lab room.

among all the available signals at two or more neighboring locations. This is an important property which can guide the design of the underlying classification method.

- **Quasi-stability in Time:** The Fig. 3.6 plots show that even though observations are made at distant time points (40 days) apart, the Signal Strength Patterns exhibits quasi-stable traits. This is very important and required characteristic from location estimation stand point.

- **Overlapping Boundaries:** However, despite the distinguishable information, it is quite obvious that there is a considerable overlap in signal strength values at different locations which, from a pattern recognition algorithm standpoint, results in non-linear and complex class boundaries. Moreover, due the high noise levels, it is difficult to capture the all possible RSS values within short time. Therefore the time-difference between training and test patterns should be used to assess a realistic expected location estimation error.

- **Multi-model, Non Uniform Densities:** At a unique location, the probability density of the RSS is formed according to the particular surrounding environment which is often unique. This is because the profile, shape of the probability density function, of the signal strengths of an AP received at location 'x' is not necessarily identical to the one received at location 'y'. Therefore, assuming a generalized density formation for RSS at all locations is naively simplistic.

3.2 Limitations of Previous Approaches

Previous approaches assume that all input signals are available at every location, mainly due to small problem area of their location system. This assumption leads to the monolithic, large and complex classification machine for location estimation. The classifier learns to estimate all locations, or classes, using whole radio map feature space. This approach is referred to as *all-classes-one-classifier* in literature [55]. It is observed that in many complex pattern classification tasks where the number of classes is large and input space is noisy, the *all-classes-one-classifier* either may not learn pattern-to-class function or may take very long to learn[55]. Due to intermittent visibility of the WiFi radio signals, all access points are not accessible at all target locations all the time. It is hypothesized that this phenomena imposes a primary limitation on performance of *all-classes-one-classifier* approaches. Since a particular access point corresponds to one dimension of the radio map feature space , therefore invisibility of signals introduce *missing* values in RSS features. Previous approaches handle this situation by representing unknown values with very low RSS such as -100 dBm; however, this practice causes redundancy in the feature space. Redundant RSS values contain little information to influence the discrimination ability of feature space; instead they can negatively effect performance of classifier in two ways. First, redundant features increase sample size to dimensionality ratio. It is well known that classification error is mainly determined by ratio of training sample size and feature space dimensions [42], [23]. Second, unknown features do not compactly represent the target locations in terms of signal measurements. Non-representative features reduce the efficiency of classifier and contribute to increase computational complexity as well as memory requirements.

3.3 Modular Classification Model for Location Estimation

Modularity is principally proven to be an effective method for improving performance in many pattern recognition problems (see [55]). However no single approach suffices for every problem. Each modular design is mainly guided by the specific nature of the problem and related goals. An intuitive approach to overcome the limitations mentioned in Section 4.2 is to decompose the location estimation task into several sub-tasks based on *a priori* knowledge about visibility of signals in the target area. This can be achieved by partitioning the input signal space into several feature spaces and training individual classification modules to learn the association between signal strengths and respective locations. Let RSS^d represent input signal space

$$RSS^d = (rss_{i1}, rss_{i2}, rss_{i3} \dots rss_{id}) \tag{3.1}$$

where i is the i^{th} input vector and each input vector is composed of signal strength values of d access points. This signal space provides connectivity to mobile device in a finite output space A which is divided into m disjoint locations.

$$A = \{l_1, l_2, l_3 \dots l_m\} \tag{3.2}$$

The learning procedure of RSS pattern classifier requires a training data set τ containing pairs of n input vectors and corresponding output locations.

$$\tau = (RSS_i^d, l_j)_{i=1,j=1}^{n,m} \quad l_j \in A \tag{3.3}$$

In previous approaches a single large classifier is trained to learn signal-to-location associations using τ. This thesis proposes to partition both input as well as output spaces based on visibility properties of radio signals in a way that the input space is divided into an arbitrary collection of nonempty subsets which constitute the actual RSS^d once combined

$$RSS^d = (RSS_1^c \bowtie RSS_2^c \bowtie RSS_3^c \bowtie \dots RSS_q^c) \tag{3.4}$$

such that $c < d$ and q is total number of partitions in the input space. The output space is partitioned correspondingly and each subset RSS_i^c defines a region R which is a nonempty set of locations in the output space.

$$R_i = \{l_{i1}, l_{i2}, l_{i3} \dots l_{il}\} \forall l_{il} \in A \tag{3.5}$$

and

$$A = \bigcup_{i=1}^{q} R_i$$

This partitioning of the input and the output spaces is driven by visibility properties of access points in the area.

Separate classification modules can be employed, based on visibility dependency between (3.4) and (3.5), such as a subset of access points defines the input vector of each classifier and the locations in each corresponding region R_i become its outputs.

$$(R_i \longleftrightarrow RSS_i^c)_{i=1}^q \tag{3.6}$$

Fig. 3.7 shows schematic diagram of modular classification system for RSS based location estimation. This approach leverages several desirable features of location systems which

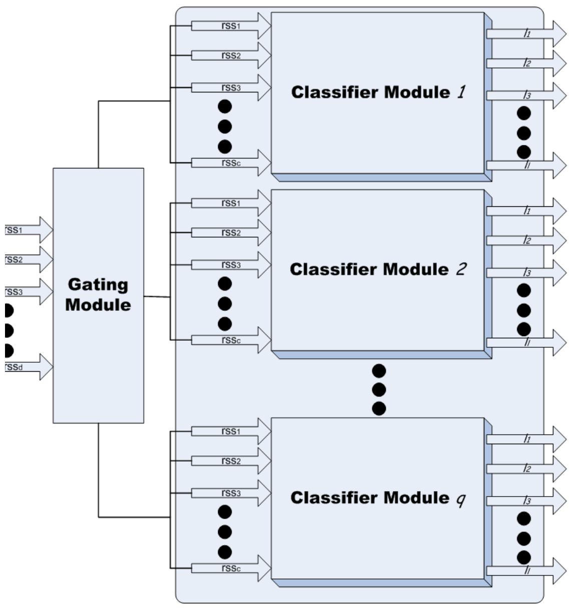


Figure 3.7: The Modular Classification Model

cannot be realized using previous approaches.

1. Insulation refers to separating the concerns by localizing the learning and recognition of *related* patterns and classes into individual classifiers.

2. The high dimensions and *low sample size* contributes to learning and generalization ability of the classifier [42]. Signal strength based location estimation faces the issue of small sample size because only finite number of samples can be collected at calibration time.

3. Parallelism is a common trait of modular designs which allows faster training of classification systems. Since a large and complex problem is divided into several simpler sub-problems. It take lesser time to learn pattern-class associations. Moreover several classifier modules can be trained in parallel, which further reduces training time.

3.4 Visibility Modeling

It is needed to inspect the visibility properties of radio signals in order to support the propositions made by modular classification model in the previous section. This point is elucidated with the help of visibility data of 10 access points deployed in a real life environment shown in Fig. 3.8. The proposed visibility modeling approach captures *visibility* information about radio signals in the form of a visibility matrix. The visibility matrix provides a systematic way to select more representative features from the radio map feature space and then design separate classification modules using those features.

3.4.1 Visibility Matrix and Clusters

As mentioned in Section 4.2, in certain locations the signal strength of an access point drops to undetectable levels, e.g. less than -100 dBm. Suppose that at j^{th} location, the signal scanning operation is performed N_j times and the signal of i^{th} access point is detected p_i times. Then the visibility probability of a given access point i at location j is written as

$$P_{ij} = \frac{p_{ij}}{N_j} \tag{3.7}$$

such that $P_{ij} = 0$ implies no visibility and $P_{ij} = 1$ shows always visible signal. In Fig. 3.8, the target locations are listed on the edge of each graph and P_{ij} is shown as shaded area from center (which represents no visibility) to the edge (representing always visible) of each graph. The visibility probabilities of d access points at m locations can be combined into an $m \times d$ matrix.

$$VisibilityMatrix := (P_{ij})_{m \times d} \quad 0 \leq P_{ij} \leq 1 \tag{3.8}$$

This visibility matrix resolves each location l_j into the corresponding visibility probabilities of all access points.

$$[P_{ij}]_{i=1}^{d} = VisibilityMatrix(l_j) \tag{3.9}$$

The visibility probability of each access point at all target locations is calculated as part of site calibration process as explained in section 3.5.1. Notice that, locations 11,12,14,20 and 34 are missing in Fig. 3.8 graphs because of restricted access. Careful observation of the visibility graphs, shown in Fig. 3.8, reveals that among total d access points there exists q subsets or clusters of c access points which are visible at different regions. Let RSS_i^c denote i^{th} cluster of c access points which belongs to the whole input space RSS^d. Let R_i denote the corresponding i^{th} region where members of RSS_i^c has desirable visibility probability. Then, the *Visibility clusters* define all the visibility associated pairs of *access point clusters* and their corresponding *regions* such that

$$\begin{aligned} &[RSS_i^c]_{i=1}^{q} = VisibilityCluster([R_i]) \\ &[R_i] = VisibilityCluster(RSS_i^c) \end{aligned} \tag{3.10}$$

where $[RSS_i^c]_{i=1}^{q} \in RSS^d$ and $[R_i] \in A$ as mentioned in equations (3.4) and (3.5). Notice that all visibility associations among signal space and location space, as formulated in (3.10), are defined over the *visibility matrix*.

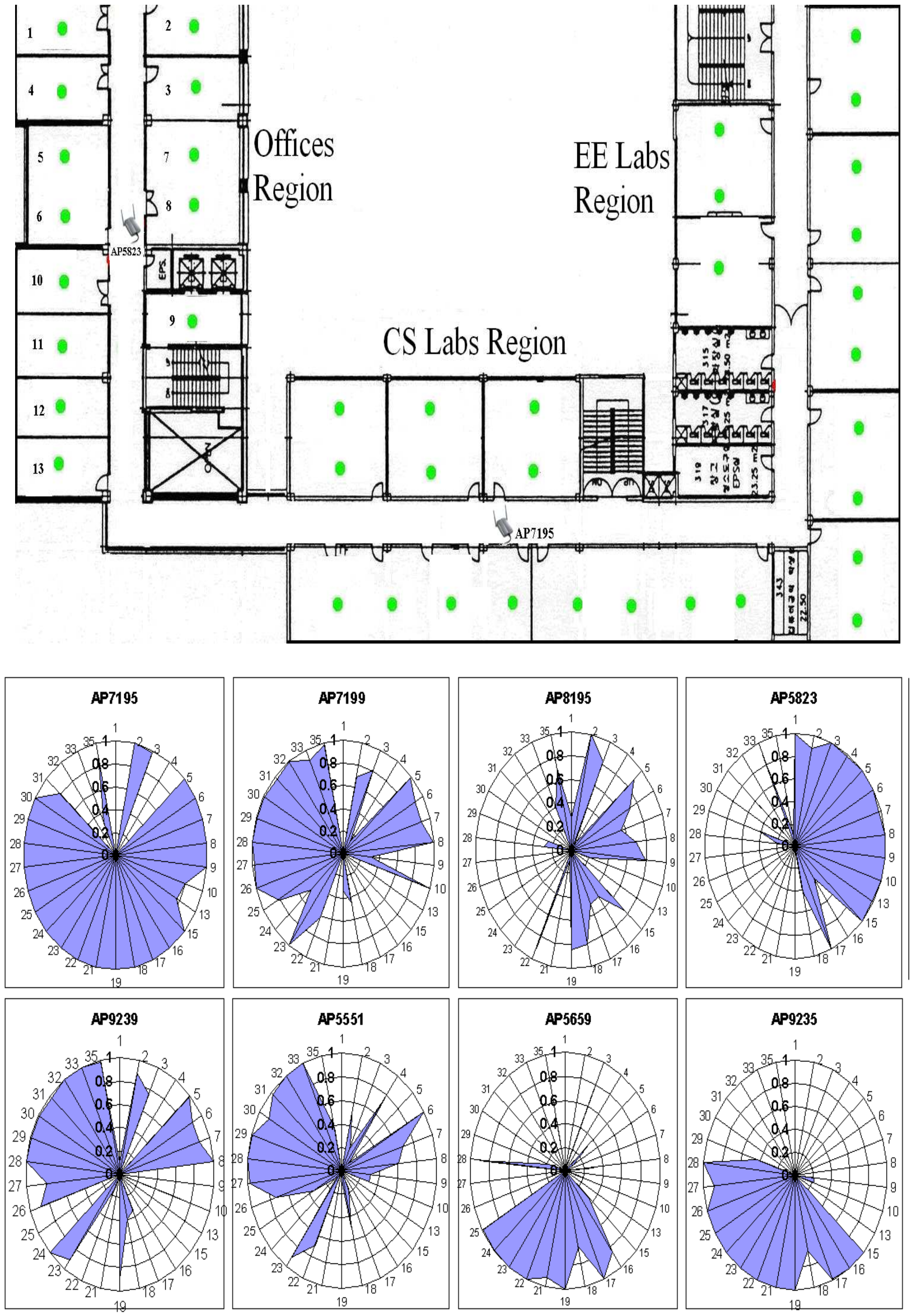


Figure 3.8: Top: Experimental site map, Bottom: Visibility of eight APs in the experimental site

3.4.2 Extracting Visibility Clusters

Once a visibility matrix is available, then it is desirable to automatically extract i) those access point clusters which have more *visibility* probability than certain threshold P_τ and ii) all locations where members of each cluster are *visible*. Let c denote the desired number of features in subspace and r denote the set of locations in corresponding region. Even though c can vary across different access point clusters but, for simplicity of discussion, suppose that c remains constant for all clusters. Algorithm 1 extracts these clusters from the visibility matrix given c and P_τ.

Even though formation of visibility clusters tends to vary in different sites due to the specific physical layout and environmental conditions at each site. Some general heuristics are presented here which largely effect this formation. Combination of three parameters influences the selection of final members of visibility clusters; , a) P_τ visibility probability threshold b) c features in subspace c) d the total number of features in the radio map, in three respects;

1. Total number of visibility clusters denoted as q. For an admissible visibility P_τ, smaller values of c and larger values of d produce several access point clusters to be visible at similar or overlapping regions. This is because if there are n access points visible in a region and $c < n$, then $\frac{n!}{c!}$ different combinations of access point clusters can become visible in that region. As described in Equation 3.6, since each visibility cluster defines inputs and outputs of the classifier module, it is desirable to generate less number of modules in order to reduce overall complexity of the modular classification system.

2. Total number of omitted locations that can not be covered by any of the clusters. Notice that even though c can take possibly all values less than d, nevertheless any arbitrary access point cluster does not necessarily provide enough coverage in terms of the number of locations.

3. Separability of new subspaces defined by access point clusters. It refers to those

Algorithm 1 Algorithm for Extracting Visibility Clusters from Radio Map

$L_{[j]}$: Collection of all target locations

i, j and k: Index of access point cluster, a target location and an access point respectively

$AP_{[k]}$: Collection of all access points

p_{kj}: Visibility probability of a k^{th} access point at j^{th} location

$R_{[i]}$: Collection of regions

$RSS_{[i]}$: Collection of access points clusters

$APSet$: temporary collection of access point

for Each Location j in $L_{[j]}$ **do**

  Initialize i, k, newAPset, and APSet

  for Each Access Point k in $AP_{[k]}$ **do**

    if $p_{kj} \quad satisfies \quad P_{\tau}$ **then**

      $Add \quad AP_{[k]} \quad to \quad APSet$

    else

      Move to next access point $AP_{[k+1]}$

    end if

  end for

  if APSet has c elements **then**

    set newAPSet flag to **true**

    for Each access point cluster i in $RSS_{[i]}$ **do**

      if $APSet$ *is member of* $RSS_{[i]}$ **then**

        Add $L_{[j]}$ to $R_{[i]}$

        set newAPSet flag to **false**and quit

      else

        Move to next access point cluster $RSS_{[i+1]}$

      end if

    end for

    if newAPSet is **true then**

      Add new APSet to $RSS_{[i]}$

      Add $L_{[j]}$ to $R_{[i]}$

    end if

  end if

  Move to Next Location $L_{[j+1]}$

end for

ambiguous vectors in feature space which belong to more than one locations or classes. Neighboring locations often receive similar signal strengths especially when situated in one room or a corridor. A straightforward solution to this issue is to incorporate more access points into feature space. Nevertheless the exact number of access points required to achieve separability depends on a particular site. The separability of training samples is measured as the ratio of separable samples and total samples representing one location.

$$s^q_{i=1} = \sum_{j=1}^{m} \frac{T_j - O_j}{T_j} \tag{3.11}$$

where T_j denotes the total samples and O_j represents the number of overlapping samples in the feature space for j^{th} location. Summation of these ratios at every location in i^{th} region gives separability of i^{th} access point cluster. Fig. 3.9 shows separability of different visibility clusters.

In order to further explain above heuristics some example visibility clusters, which exist in one of our experimental sites, are presented here. Effect of changing total number of access points used to define the radio map feature space is shown in tables 3.1 and 3.2. It is evident from these clusters that increasing features, 8 access points in table 3.1 and 11 in table 3.2, results in increased number of visibility clusters q. Three locations are omitted in this clustering. Nevertheless only three features may not result in completely separable signal vectors. Table 3.3 shows *visibility* clustering results with different parameters. In this case q is 5 and omitted locations are 3. This clustering produces no overlap in training data of different locations as can be seen in Fig. 3.9. This shows that adding more access points to the feature space enriches discrimination information by making all sample training vectors separable. However acceptable values of visibility probability P_τ restrict subspace dimensionality c to 4 and further increment of access points in c results in no visibility clusters. Consequently the total number of access points d were increased to 12. Table 3.4 shows visibility clusters where $c = 5$ in 12 access point radio map. Although this clustering results in only 3 clusters and completely separable training samples but 16 locations are omitted.

Table 3.1: Visibility Clusters A: $d = 8$, $c = 3$,$P_\tau = .55$

Access Point Clusters	Regions
AP7195AP9239AP5659	3,19,23,24,28
AP7195AP5659AP9235	16,17,18,19,21,22,23,24,25,28
AP7195AP7199AP9239	2,3,5,6,7,10,23,26,27,28,29,30,31,35
AP5551AP7199AP9239	2,6,7,8,23,26,27,28,29,30,31,32,33
AP7195AP8195AP5823	2,3,6,7,9,15,17
AP7195AP7199AP5659	3,22,23,25,28
AP7195AP5551AP7199	2,6,7,22,23,26,27,28,29,30,31

Table 3.2: Visibility Clusters B: $d = 11, c = 3, P_\tau = .55$

Access Point Clusters	Regions
AP5551AP7199AP9239	6,7,8,23,26,27,28,29,30,31,32,33;
AP7195AP5551AP7199	6,7,22,23,26,27,28,29,30,31
AP7195AP7199AP9239	2,3,5,6,7,10,23,26,27,28,29,30,31,35
AP7195AP8195AP5823	2,3,6,7,9,15
AP7195AP5659AP9235	16,17,18,19,21,22,23,24,25,28
AP7195AP7199AP5659	3,22,23,25,28
AP7195AP5659AP8135	16,17,21,22,23,24,25,28
AP7195AP9239AP5659	3,19,23,24,28

So far it is confirmed that low values of visibility probability P_τ introduce redundant features in the radio map. This reduces the location estimation accuracy and increases the

Table 3.3: Visibility Clusters C: $d = 11, c = 4, P_\tau = .55$

Access Point Clusters	Regions
AP7195AP5659AP9235AP8135	16,17,18,19,21,22,23,24,25,28
AP5551AP7199AP9239AP6079	6,8,28,29,30,31,32,33
AP7195AP5551AP7199AP9239	2,6,7,23,26,27,28,29,30,31
AP7195AP7199AP9239AP6079	3,5,6,10,28,29,30,31,35
AP7195AP8195AP5823AP5535	2,3,6,7,9,15

Table 3.4: Visibility Clusters D: $d = 12, c = 5, P_\tau = .50$

Access Point Clusters	Regions
AP7195AP7199AP9239AP6079AP5535	3,5,6,10,29,35
AP7195AP7199AP9239AP8195AP5535	2,3,5,6,7,35
AP5551AP7199AP9239AP6079AP9207	6,8,28,29,30,31,32,33

complexity of the classification system as shown in experimental results. On the other hand, low values of c correspond to decreased discrimination ability of access points in each cluster and higher values render either no clustering or result in a large number of omitted locations.

Since visibility clusters correspond to a set of access points which are visible at a set of locations, this information is used to develop binary decision rules, as shown in table 3.5, to invoke a specific classifier module for a particular signal input. In this table the availability or absence of signal from a particular access point is represented as either 1 or 0 respectively. The gating module shown in Fig. 3.7 receives a d-dimensional input feature space vector rss^d, presented in Equation 3.1, and converts it into appropriate sub-space vectors, Equation 3.4, based on visibility status of access points. This converted input vector is then routed to appropriate classifier module depending upon the decision rules in table 3.5.

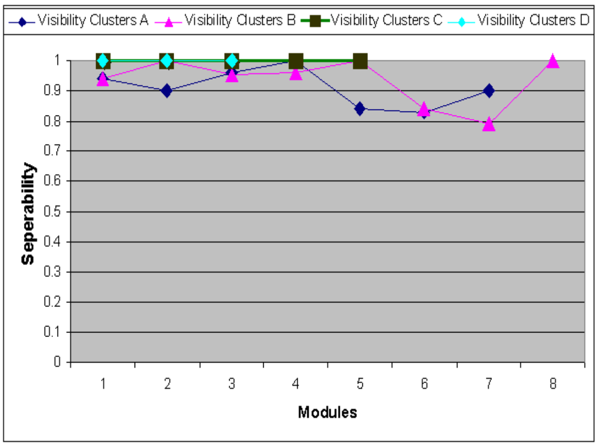


Figure 3.9: Separability of the Visibility Clusters

Table 3.5: Binary Visibility Decision Rules

7195	7199	9239	5551	8195	5823	5659	9235	Module
1	1	1	0	0	1	0	0	m1
1	1	1	0	1	0	1	1	m2
1	0	0	0	1	1	0	0	m2
1	0	0	0	0	0	1	1	m4
1	0	0	0	1	0	1	1	m5

3.5 Experimental Design

In order to evaluate modular classification model in real life environment; several experiments were conducted in a campus building. The selected sites were located on 3^{rd} floor of computer engineering department. This floor has a versatile environment containing class rooms, labs and offices. The MATLAB Neural Network Tool Box [47] was sued for training the location classifier. Nevertheless, in order to actually deploy trained classifiers onto mobile devices, component-based software libraries CompoNet [1] were developed to encapsulate arbitrary size and structure of neural networks into a software object which can be invoked as a function from other programs. Experiments were carried out using a public network of 3COM IEEE 802.11 (a, b, g) WiFi access points at two different sites. In order to actually scan signal strengths, two types of consumer devices with built in WiFi Network Interface Cards (NIC) were used for RSS scanning: 1) HP iPAQ Pocket PC running Windows CE and 2) Toshiba M30 Laptops running Windows XP. Fig. 3.8 show map of the target site with target locations marked as small filled circles. Akin to typical pattern classification tasks, RSS based location system development is divided into feature-space creation, preprocessing, classifier training and testing phases. The experimental design and test setup is discussed in following sub sections.

3.5.1 Sensor Data Collection

Site calibration phase involves scanning RSS patterns at discrete target locations. Signal strengths of access points can be scanned passively and actively. In former case each access point periodically broadcasts an announcement packet. Mobile devices can parse that packet to know the signal source Basic Service Set Identifier (BSSID) and signal strength. The BSSID contains MAC address of access point which is used to identify different access points. In active scanning case, the mobile device broadcasts a query signal to access points. In response to the query signal each access point sends a reply signal back to querying device containing its BSSID. The active scanning method is used for scanning signal strengths. For each type of devices, customized software modules called Calibration Agents were developed to access NIC hardware. Calibration agent invokes the scanning process on an adjustable frequency and parses response signals of each access point. As presented in section 3.4, a visibility matrix is developed during calibration phase in addition to the radio map. This is actualized through a histogram based data collection method. Apart from hardware interfacing, the calibration agent builds a histogram data structure in its memory to store scanned data. Sensor measurements contain pairs of access point BSSID and Received Signal Strength Indicator (RSSI) values $O^d_{i=1} = [BSSID_i, RSSI_i]$. Calibration agent parses each pair O from radio signals and establishes a separate histogram for individual access points. The visibility probability of access points is computed such as for N signal scan observations at j^{th} location, each histogram provides the total number of times the signal from i^{th} access point is detected $p_{ji} = \sum_{i=1}^{n} x_i$. The ratio of N and p_{ji} gives visibility probability as explained in (3.7).

Prevalent method for site calibration uses a network card interface to extract RSS values from hardware and a graphical program with image map of site which allows developers to pinpoint their location. This practice is tiring for sufficient calibration of large sites. In order to alleviate the required labor for data collection task, a distributed site calibration system is developed which allows multiple devices to simultaneously calibrate site in short time [9]. RSS based location estimation is directly affected by the degree of how closely sample signal data represent the real life radio signals. Therefore un-customary efforts were undertook to gather sample signal strengths. Instead of creating the radio map at one time and then dividing it into training and test samples, the target site was calibrated for seven days and at different timings of each day. Each target location was calibrated multiple times by different people and devices. Data of alternate days was used for, respectively, training and testing the classifiers. During sample data collection routine activities were taking place which are peculiar to typical indoor environments such as lecture rooms, labs and admin offices.

3.5.2 Location Estimation Error Measures

Each location is identified in two ways: a) unique identification number (ID) and b) cartesian coordinates on xy-plane. The former was used for classifier training and later for error analysis. For classifier training, class labels or target locations are assigned in the range of

1 to 35. This ID is then encoded into sparse array which has all 0-valued elements except the one at the index of ID which contains 1. This array has zero values in all elements for the location which was not accessible for calibration. The location estimation error is measured in two aspects:

a) Absolute deviation of location estimate from actual location is measured as Mean Absolute Error (MAE)

$$|MAE| = \frac{1}{N}\sum_{i=1}^{N}\left|l_i - \hat{l}_i\right| \tag{3.12}$$

where N is the total number of training or test patterns, l_i is the estimated location and $\hat{l}_i$ is the true location of i^{th} pattern. Deviation between l_i and $\hat{l}_i$ is calculated as euclidian distance

$$\triangle_i = \left|l_i - \hat{l}_i\right| = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{3.13}$$

where (x_i, y_i) are corresponding coordinates of l_i location estimate of i^{th} pattern and $(\hat{x}_i, \hat{y}_i)$ is the actual location.

b) Relative deviation of location estimate, denoted as e_r, is measured relative to some tolerable error threshold denoted as Γ. It implies that relative error reflects severity of error in location estimate by allowing some deviation which is not more than Γ. An estimate is less severe if it is relatively closer to actual location than the one which is farther away. Thus E_r gives a percentage of total N estimates in which $\triangle_i$ is admissible or less than Γ.

$$E_r = \frac{1}{N}\sum_{i=1}^{N}\left[(\triangle_i \leq \Gamma)\right] \tag{3.14}$$

where threshold $\Gamma_{i=1}^{3}$ produces three severity of error values averaged over all locations.

3.5.3 Selection of the Classification Methods

It is observed that the performance of classification methods is closely related to the characteristics of the underlying data [58]. Therefore an appropriate selection of these methods should be considering the RSS data characteristics detailed in the Section 3.1.2. The choice of classifier is mainly determined by two factors; i) the capability to approximate non-linear and complex class boundaries and minimize the classification error, ii) no requirement of certain density assumption in the data, and iii) learn the relationship between points in location space and signal space with limited number of samples.

K-Nearest Neighbors Algorithm

The k-NN is one of the most popular nonparametric method which does not make any density assumptions. Different variation of this method are applied to the location estimation problem as reported in the literature [30, 51, 46]. In this method certain distance measure, often Euclidean formulae 3.15, is calculated between the measured metrics, RSS, and all training samples in the Radio Map.

$$d = \sqrt{\sum_{i=1}^{n} (AP_i^{Trg} - AP_i^{Tst})^2} \tag{3.15}$$

The location estimate is determined to be the one associated with the minimum distance d. Despite its robustness to the noisy measurements and competitive performance, the simple k-NN method can become computationally expensive when the size of the Radio Map increases due to larger area of coverage. Besides considering other more sophisticated pattern classification methods, we have adopted a heuristic approach for applying k-NN in order to keep it computationally light weight. This heuristic is derived from the fact that a mobile device can move only to the next possible location from a given current location. A priori information about the next possible set of locations reduces the search space for the algorithm as shown in Fig. 3.10.

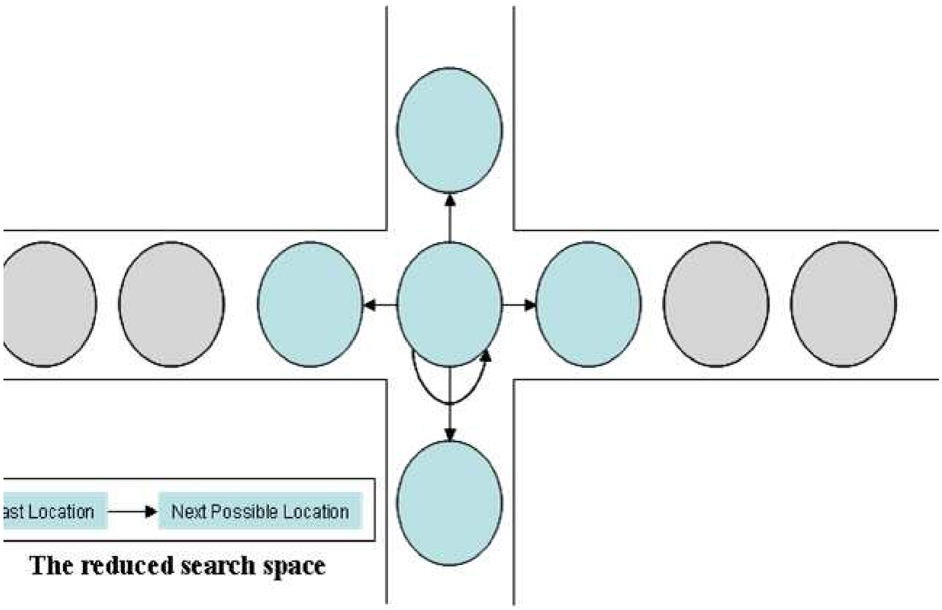


Figure 3.10: Reducing the Search Space for k-NN.

The reduction in the search space needs less computational resources, however it requires a detailed modeling of the location space and the simple location determination becomes a location tracking task. The memory requirements of such setting requires both the reference entities: i) whole sample radio map and ii) the location model to be available on the mobile device. Due to the resource constraints, this solution may not be feasible for most of the devices. Therefore, this heuristic is best applicable in distributed settings where a back end server keeps both the reference entities. However, this setting might not be deemed suitable for the privacy sensitive applications.

Learning Vector Quantization

Learning Vector Quantization (LVQ) classifier employs non-parametric nearest neighbor pattern recognition algorithm based on Kohonen's self-organizing-maps [43]. Fig. 3.11 shows application of an arbitrary structure of a Learning Vector Quantization (LVQ) network for location estimation. The classifier takes individual components of RSS vector as input and produce an estimate of most likely location of the device which is reporting these RSS values.

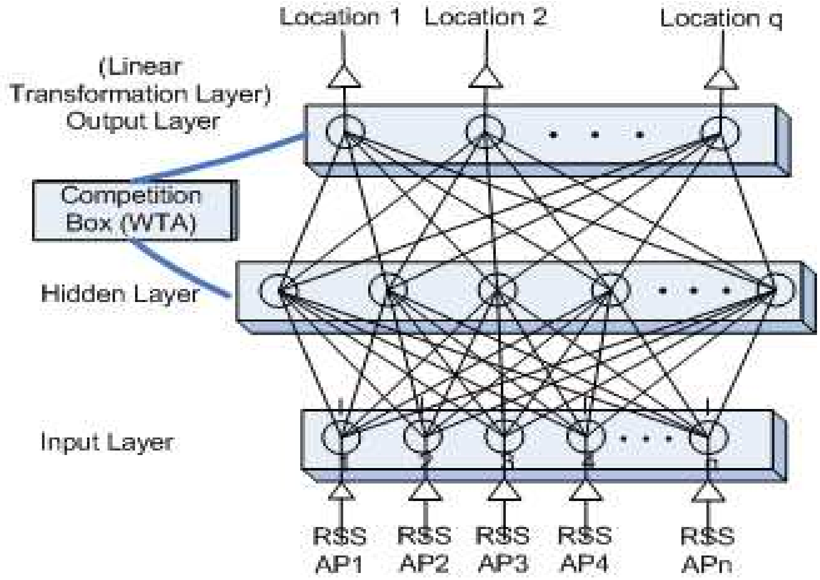


Figure 3.11: Employing Learning Vector Quantization for Location Estimation

The design of LVQ network specifies three layers of neurons. Input layer contains as many neurons as components of input vector I. Hidden layer contains competitive neurons to represents subclasses and output layer consists of neurons which represent actual classes in the input space.

The hidden or competitive layer learns to classify input vectors in much the same way as the competitive layer of self-organizing-maps. Input neurons and competitive layer neurons are interconnected through IW weights. When an RSS input is applied to the network; the distance between each hidden layer neuron and input vector is computed. Then Winner Takes All (WTA) competition is applied to find the closest subclass, the winner, neuron IW_w. Once the input vector is classified at hidden layer the third layer, also called linear transformation layer, transforms the output of competitive layer into target classification vectors. Learning occurs by adjusting the IW_w weights in such a way as to move it closer to input vector if classification result is correct

$$IW_w = IW_w + \alpha(I - IW_w) \tag{3.16}$$

or farther if input is incorrectly classified.

$$IW_w = IW_w - \alpha(I - IW_w) \tag{3.17}$$

Learning rate of the network is determined by the parameter α. Number of hidden layer neurons influences the learning as well as generalization capability of Learning Vector Quantization networks. Several Learning Vector Quantization networks were employed with different hidden layer neurons. Table 3.8 lists the training results of best performing LVQ classifiers. Analogous to the naming convention of multi layer perceptron experiments, modular networks are denoted as $mLVQ_i$ and non modular ones are denoted as LVQ in results tables.

Multi Layer Perceptron

MLPs are general-purpose, flexible models which can, given enough number of hidden neurons and samples, approximate any non-linear function of arbitrary complexity with high degree of accuracy [41, 58]. This classifier is chosen for its ability capture the non-linearity in the signal to location mappings without imposing any density assumptions on the input data. Another attractive feature of this method is that the domain knowledge can be incorporated into the design of an MLP by means of architectural choices such as hidden layers, units, transfer functions so on [24]. Previously, the application of MLP for location estimation is reported by Battiti *et al.* in [12]. Fig. 3.12 shows an arbitrary structure of a Multi Layer Perceptron (MLP) network for location estimation. Network takes individual components of the RSS vector as input and produces an estimate of most likely location of the device which is reporting these RSS values.

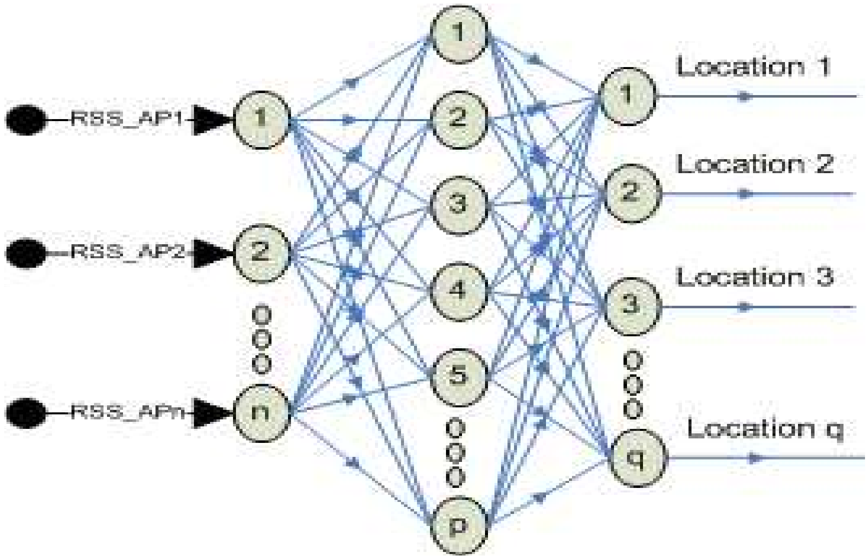


Figure 3.12: Employing Multi Layer Perceptron for Location Estimation

3.6 Experimental Results

The *Radio Map* feature space is used to train location classifiers. During training phase, preprocessing of feature space is one important step to encode inputs and outputs into a format suitable for classification method. The range normalization is applied on the *Radio Map* feature space as shown in Equation 3.18.

$$rss_{norm} = 2((rss - rss_{min})/(rss_{max} - rss_{min})) - 1 \tag{3.18}$$

where rss_{norm} is normalized signal strength. RSS values fall in the range of -100 dBm to -10 dBm [52], these values were used as global minima rss_{min} and maxima rss_{max} for normalizing all features. Separate multi layer perceptron based location estimation systems were trained for the target site. For the sake of comparative analysis both the non-modular, represented as MLP, and modular, represented as $mMLP_i$, location estimation systems were trained. The radio map has 11 dimensions access points for training MLP and for the mMLP classifiers the training data was extracted using tables 3.3 visibility clusters.

Several multi layer perceptron networks were trained with different parameters such as hidden layer neurons, learning function, transfer functions and training epochs. Training results of only the best performing networks for the target site are represented in Table 3.6 shows training results of for site 2. Detailed discussion of the effects of different parameters on location estimation performance is given in [4].

As results tables show in 'Training Function' column, two training algorithms were employed to learn pattern-location pairs: i) Levenberg-Marquardt (LM) developed by Hagan *et al.*[39] and ii) Moller's Scaled Conjugate Gradient (SCG) presented in [48]. In our experiments, LM algorithm achieved nearly zero training error in fewer iterations but its requirement of computational resources is prohibitive for training large networks. On the other hand SCG achieved comparable performance and does not require excessive computational as well as memory resources during training.

Topology column contains structure of respective neural network which is represented as $I - H - O$ where I is input neurons, H is hidden layer neurons and O is output layer neurons. Epochs column shows number of training iterations that respective network took to achieve Mean Absolute Error (MAE) which is listed in last column. One common property that all networks in our experiments share is the choice of transfer functions which is *logsig*moidal function $logsig(n) = 1/(1 + e^{-n})$ at hidden layer neurons and *tansig*moidal function $tansig(n) = 2/(1 + e^{-2n}) - 1$ at the output neurons.

Table 3.6: Training Results of Multi Layer Perceptron Classifiers

Module	Training Function	Topology	Epochs	Training MAE
MLP	SCG	11-70-35	2000	0.021
mMLP1	LM	4-20-10	88	0.021
mMLP2	LM	4-20-8	80	0.082
mMLP3	LM	4-25-10	86	0.0184
mMLP4	LM	4-20-9	50	0.0047
mMLP5	LM	4-15-6	45	0.009

Table 3.7: K-Nearest Neighbors Algorithm using Visibility Clustering

Visibility Cluster	MAE	$E_r \leq 1$	$E_r \leq 2$	$E_r \leq 3$
VC1	0.96	0.72	0.88	0.98
VC2	0.32	0.95	0.96	1
VC3	0.09	1	1	1
VC4	0.12	0.97	0.99	1
VC5	0.88	0.78	0.94	0.99
VC6	0.25	0.96	0.96	1

Modular classification, mMLP and mLVQ, results are shown in Tables 3.9 and 3.10. Both classifiers exhibit similar performance, in terms of absolute and relative errors, except module 3 and 5.

Overall performance of modular classification system is compared with non-modular classifier in Table 3.11. In this site MLP performed better than LVQ classifier both in terms of absolute and relative error. However LVQ benefits from modularity significantly more than MLP with respect to absolute distance error. The mMLP and mLVQ produced relative

Table 3.8: Training Results of Learning Vector Quantization Classifiers

Classifier	Learning Function	Topology	Epochs	Training MAE
LVQ	lvq1	8-100-35	150	0.018
mLVQ1	lvq1	5-30-10	45	0.03
mLVQ2	lvq1	4-30-8	20	0.009
mLVQ3	lvq1	4-25-10	40	0.027
mLVQ4	lvq1	4-25-9	25	0.057
mLVQ5	lvq1	4-15-6	15	0.048

Table 3.9: Modular Approach Results of Modular MLP

Classifier	Test MAE	$E_r \leq 1$	$E_r \leq 2$	$E_r \leq 3$
mMLP1	0.4140	85%	90%	91%
mMLP2	0.7286	91%	92%	94%
mMLP3	0.4957	81%	95%	96%
mMLP4	0.556	91%	92%	99%
mMLP5	0.9806	68%	88%	88%

positioning error $E_r \leq 1$ for 84% and 80% times respectively. On the other hand, non-modular MLP and LVQ achieve similar performance if $E_r \leq 3$. Which means that severity of error of modular classification system is 3 times better in this site than monolithic counterparts. As the tolerable error threshold relaxes up to 3 positions, the modular classification system consistently provided superior results in comparison with non-modular approach.

3.7 Summary

The location estimation performance is compared with monolithic, non-modular classification approach using two famous neural networks 1) Multi Layer Perceptron and 2) Learning

Table 3.10: Modular Approach Results of Modular LVQ

Classifier	Test MAE	$E_r \leq 1$	$E_r \leq 2$	$E_r \leq 3$
mLVQ1	0.5571	87%	91%	91%
mLVQ2	0.7214	91%	91%	92%
mLVQ3	0.9760	66%	92%	93%
mLVQ4	0.4087	93%	95%	95%
mLVQ5	1.25	61%	78%	94%

Table 3.11: Summarized Results

Classifier	MAE	$E_r \leq 1$	$E_r \leq 2$	$E_r \leq 3$
MLP	1.00	70%	78%	85%
mMLP	0.63	84%	92%	95%
LVQ	2.40	54%	68%	78%
mLVQ	0.78	80%	90%	93%
k-NN	0.73	77%	80%	86%
k-NN(VC)	0.43	84%	90%	95%

Vector Quantization. Same neural networks were employed to realize the modular classification model. Comparative results show superiority of modular approach in terms of both absolute error and relative error across different sites. The note worthy improvement in location estimation accuracy is observed consistent across different sites. With respect to absolute error measure compared to non-modular approach the accuracy of modular approach is improved 1.58 and 3 times respectively for MLP and LVQ classifiers. All methods benefit from modular approach in terms of relative error in the same way. On the basis of extensive experimental results, it is concluded that modular classification model achieves significant improvement in location estimation accuracy as well as enables systematic expansion in coverage area of location systems.

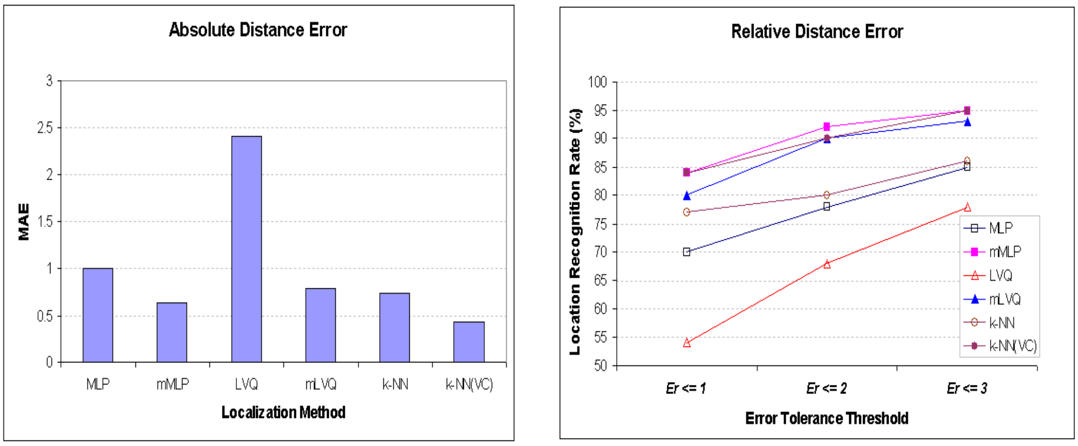


Figure 3.13: Right: Relative Distance Error, Left: Absolute Distance Error

Chapter 4

A Rapid Development Approach

4.1 Introduction

An end-to-end development life cycle of RSS based location system can be divided into two major stages and five different phases. Fig. 4.1 shows general schematic of development life cycle. It comprises two stages, in bottom-up direction, containing different development phases and respective subsystems. A vertical line is shown which again divides the subsystems into two modes of development. *Offline* mode subsystems require lab time for data collection, preprocessing, classifier training and optimization etc. Whereas *online* mode means that subsystems can be made with plug in components as described in [7], [10] and [5].

One of the major draw backs of existing approaches is that it requires extensive and laborious sensor data collection or *Site Calibration*. Partly due to this problem, existing systems focus on evaluating classification methods on small scale location systems while, as discussed by [3], development of large scale location system presents certain challenges which degrades the accuracy of classifier. These issues are not particularly addressed in previous work. Section 4.2 identifies and discusses these limitations in detail.

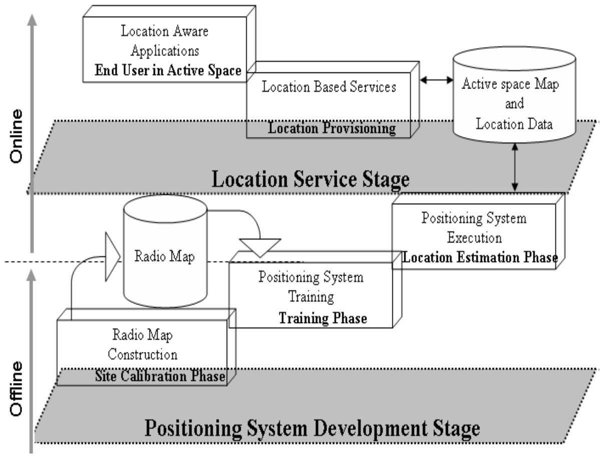


Figure 4.1: Typical Development Life Cycle of Location Systems

The design goals of an ideal approach should address these issues as discussed in section 4.3. This chapter demonstrates the efficacy of online incremental learning approach through proof of concept implementations in real environment. Main contribution of this work is to provide succinct analysis of issues, lay down the design and implementation of our approach for *rapid*ly developing *large scale* and incrementally *extensible* location systems and extensively evaluate it in comparison with previous approaches. Primary motivation of this work is to exploit 'fuzzy arithmetic' and 'adaptive resonance theory' based *online and incremental* learning systems , Fuzzy Art [16] and Fuzzy ArtMap [15], for achieving our design goals. An overview of their learning dynamics is given in section 4.4. A **Con**text-aware, **Sel**f-scaling **F**uzzy **A**rt**M**ap system is presented in section 4.5. The experimental design and test setup technical details are elaborate in section 4.6 and section 4.6.2 presents comparative results analysis.

4.2 Limitations of Radio Map Based Approach

Despite several research results have proved the feasibility of RSS based location systems, there are still some issues which prohibit wide scale availability of this technology. This section offers a discussion of these issues and limitations in previous approaches.

4.2.1 Laborious Development and Reconstruction

Developing a location system fundamentally amounts to *Site Calibration* and *Classifier Training* phases, as can be seen in Fig 4.1, but special nature of indoor radio wave poses *reconstruction* problem as well. Such environments as super markets and hospitals might experience restructuring, even though not frequently, which most likely changes signal strength distributions. This can potentially render a location system ineffective and require reconstruction of classifier even if only a small part of building is affected. Thus both *development* and *reconstruction* of location system become a laborious task.

RSS based location estimation is a multi-class classification problem which requires extensive sensor data collection in order to create *Radio Map* which provides training data for classification machines to learn signal to location relationship. It has been reported by several researchers that detailed radio map, in terms of number of samples per location, is crucial component for finer granularity and higher accuracy. Recently some researchers have proposed statistical interpolation of RSS data to reduce the effort and time required for constructing radio map. Radial basis functions have been employed by [29] to interpolate the missing data of un-calibrated locations. [27] proposed a method of using unlabeled samples for reducing the sampling rate at each location and number of locations. It should be noticed that even if the actual calibration points are reduced, these techniques still need to calibrate the area. Despite an extensive research being put into enhancing calibration speed, construction of Radio Map has been a major hurdle in wide acceptability for this technology.

As the number of locations increase, pattern classification task becomes harder due to overlapping classes and imbalance in class data. Mostly, it takes several training and parameter tuning sessions in order to make a classifier to yield sufficient accuracy. Same time is required to retrain the classifier in case of *reconstruction* as changes in signal distribution in a small area require retraining whole classifier. Modular classifier approach tries to localize this problem by portioning the location space into subspaces and train several redundant as well as sperate modules for each subspace [3].

4.2.2 Increasing The Scope of Location System

Location systems might be required to recognize new target locations. This requirement is reffered to as "increasing the scope of location system" which involves two aspects of i) *extensibility* and ii) *expansibility*. Before these concepts are described in the context of location estimation, a formal definition of *range* and *scope* of location system is required. Radio wave obeys inverse-square law in free space which states that

$$signalstrength \approx 1/r^2$$

where r is distance between transmitter and receiver. Indoor environment further impose several environmental factors which collectively reduce signal strength outside a certain region to be undetectable by the receivers. This physical property of radio signals ultimately defines the range of a location system. Assume that a classifier is trained to learn association between signal space S and location space A, where S is composed of n access points. *Range* of this system can be formalized as

$$S \supseteq (AP_1, AP_2, AP_3, ...AP_n) \tag{4.1}$$

whereas $S \rightarrow A$.

Similarly, pattern classifier that categorizes RSS input vectors into a set of n target locations $(a_1, a_2, a_3...a_n)$, the *scope* of a location system is an area A such that

$$A^S \supseteq (a_1, a_2, a_3, ...a_n) \tag{4.2}$$

Definition 1: Extending means to increase the *scope* of location system without changing its *range*.

According to this definition there could be a new set of m locations $(a'_1, a'_2, a'_3...a'_m)$ to be recognized by the classifier which was already trained using same dimensions of the input space for area A.

$$A^S \supseteq (a_1, a_2, a_3...a_n) \cup (a'_1, a'_2, a'_3...a'_m) \tag{4.3}$$

Definition 2: Expanding means to increase the *range* of location system in order to increase its *scope*. This implies that there could be a new set of p locations $(b_1, b_2, b_3...b_p)$ that a location system should learn to recognize in addition to already learned A. Whereas these locations may or may not be within the *range*, Equation 4.1, of location system. Let S' denote a new set of access points

$$S' \supseteq (AP_{n+1}, AP_{n+2}, AP_{n+3}, ...AP_m)$$

required to form additional signal space which includes these locations. Then a range increment can be represented as

$$A^{S \cup S'} \supseteq (a_1, a_2...a_n) \cup (a'_1, a'_2...a'_m) \cup (b_1, b_2...b_p) \tag{4.4}$$

Notice that increasing the range necessarily increase the scope of system but not otherwise.

Although this requirement seems to be very basic capability that location systems should posses but, so far, no such capability is improvised by previous systems. In order to achieve this scalability, using previous approaches, Radio Map feature space is required to be extended to include training RSS pattern-location data and then retraining of classifier with extended radio map. In case of retraining with new feature space, most of the *offline* training based classifiers face the *Stability-plasticity dilemma* which states that learning new pattern-class mappings causes erosion of previous knowledge acquired by classifier during early training. Another method to overcome this problem is to retrain classifier with whole Radio Map (that includes both old and new training data) which is tedious and cumbersome.

4.3 Online and Incremental Learning Approach

In this chapter, a novel approach for building location systems based on ConSelFAM neural network is presented. Fuzzy ArtMap is a generalized ArtMap (also called Predictive Art)[15] network which can handle analog input patterns and performs *online* and *incremental* learning of pattern-class pairs presented in any arbitrary order. This approach effectively overcomes the limitations, presented in section 4.2, as well as offers several desirable features which cannot be realized using previous methods. Unlike previous approaches, our approach does not require Calibration Phase and *offline* (or lab time) Training Phase by means of online learning which shortens development time dramatically. Fig. 4.2 shows how our approach transforms location system development life cycle.

The following sections explain how this approach meets the design goals such as reducing excessive development and training time, increasing system *scope* and addressing the *visibility* problem.

4.3.1 Rapid Development via Online Learning

Fuzzy ArtMap classification system learns pattern class pairs *online*, which implies that *Radio Map* feature space need not be created prior to model training. This property enables such location systems that can be built without calibration phase and *offline* model training phase. Previous approaches are based on *offline* training based pattern recognition methods

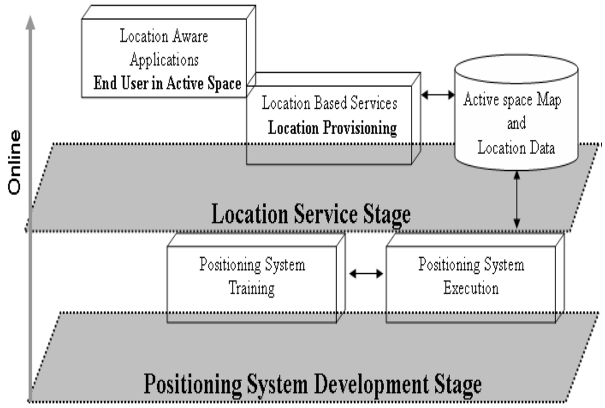


Figure 4.2: Rapid Development Approach

which incurs long site calibration phase and training phase onto development life cycle as shown in Fig. 4.1. By virtue of online learning capability of Fuzzy ArtMap both phases can be removed from development life cycle, as shown in Fig. 4.2, and rapid location system development can be realized.

4.3.2 Extensibility and Expansibility via Incremental Learning

Flexible and Dynamic expansion of location system is easy and straight forward in our approach. As discussed in section 4.2.2, by expanding location system flexibly and dynamically it means to incorporate new locations thus increasing the *scope*, Equation 4.1,4.2, of location system . Fuzzy ArtMap is capable of *incremental* learning and demonstrates stable learning of classes when exposed to a new set of pattern-class pairs [15]. This capability allows flexible learning of new locations without requiring retraining with old and new feature space as is the case with previous systems. Nevertheless Fuzzy Art requires that capacity of network, in terms of number of categories (locations in our case) that network can learn, need to be fixed prior to learning. Prior fixation, of number of categories that network can learn, means that once RSS patterns of a fixed number of locations, with respect to the capacity of network, are learned by a network then more locations can not be incorporated or learned by that network. This limits the application of Fuzzy ArtMap in terms of dynamically expanding a pattern classification problem or, in this case, location system. The Fuzzy ArtMap system is extended such that it does not require prior fixation of capacity of network thus allowing network to self-scale itself as new categories are presented to it. This solution is explained in section 4.5.1 in more detail.

4.3.3 Visibility Awareness

Previous researches have shown that *Visibility Matrix* based approach of *modular classifiers* improves location accuracy. This method partitions input space as well as output space based on visibility probability of a set of access points over a cluster of locations. Then separate classifiers, called modules, are trained for each partition as shown in previous chapter. Although modular approach improves overall accuracy but, obviously, this method increases complexity and might take longer periods of training. It is hypothesized that improvising the visibility information as *context* and enabling the classifier to recognize RSS patterns based on it shall enhance the localization performance. Comparative results show that this context-awareness achieves comparable accuracy to modular approach and greatly simplifies the visibility-aware learning. The learning dynamics of context-awareness are explained in section 4.5.2.

4.4 Fuzzy Art and Fuzzy ArtMap

Fuzzy Art is Adaptive Resonance Theory based self organizing neural network for real time autonomous learning environments. Fuzzy ArtMap is composed of a pair of Fuzzy

ART neural networks denoted as Fuzzy ART$_a$ and Fuzzy ART$_b$. A brief, and sequential, explanation of Fuzzy Art system is provided first and then Fuzzy ArtMap system in this section. Fuzzy Art combines of fuzzy set theory and adaptive resonance theory (ART) [15] to accomplish unsupervised, incremental and online learning of analog valued input vectors presented to system in arbitrary order.

The learning dynamics and Ababil's implementation of of Fuzzy Art neural network system is presented here. A comprehensive treatment of Fuzzy Art characteristics can be found in [16]. Fig. 4.3 shows topological structure of Fuzzy Art.

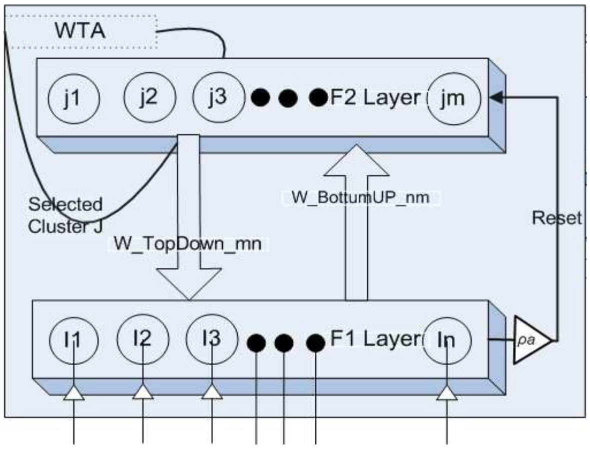


Figure 4.3: Fuzzy Art Network Structure

It consists of two processing layers F$_1$ and F$_2$. Each neuron of F$_1$ layer is linked, through bottom up synaptic connections W_BottomUp_nm, to all neurons of F$_2$ layer and vice versa. Adaptive weights of bottom up and top down synaptic connection, denoted as W_TopDown_mn, bear same value in Fuzzy Art systems. F$_2$ layer neurons represent learned categories in input space. Each F$_2$ neuron is refereed to as *committed* (if it already represents a category) or *uncommitted* (it it is not representing any category). Fuzzy Art learning is governed by a *choice* parameter α, a *vigilance* parameter ρ and and learning rate parameter β. The network initialization creates n and m neurons in F$_1$ and F$_2$

layers respectively and synaptic connections get initialized. ART systems can employ a special preprocessing method, so called *complement coding*, to input vector I. Orienting subsystem determines system choice of m F_2 layer categories for this input vector. This is achieved by evaluating a *choice function* by F_1 layer neurons. For every input vector $I = (I_1, I_2, I_3 ... I_M)$, this function produces a ranking list based on component wise fuzzy distance $T(_j)$, between input pattern and synaptic connection weights, and a choice parameter α.

$$T(_j)_{j=1}^{n} = |I\,(j)| \wedge W\,(j)\,/\alpha + |W\,(j)| \tag{4.5}$$

$T(_j)$ ranking list is adaptively fed into a Winner Takes All (WTA) filter and resulting winner neuron is, tentatively, selected as category of current input pattern. Vigilance subsystem confirms, or dismisses, this decision based on externally adjustable vigilance parameter ρ. In case of dismissal relearning continues by reevaluating the ranking list until a satisfactory category is found or learning capacity, denoted here as n, of system is reached. Learning ensues once a category choice satisfies vigilance subsystem. All weights that belong to confirmed category J neuron are updated as following.

$$W_J = \beta\,(I \wedge W_J) + (1 - \beta) W_J \tag{4.6}$$

One of the distinguishing properties of Fuzzy Art neural network system is that it can output a *don't know* response, which means that network do not assign an input pattern to any categories if it is very dissimilar to all categories. This capability is realized by means of learning capacity concept. When all F_2 layer neurons become *committed* and an input is encountered which does not qualify to be a member of any category then network outputs a *don't know* response.

The topological structure of Fuzzy ArtMap neural network is presented in Fig. 4.4. Fuzzy ART modules ART_a and ART_b self-organize category grouping for separate input sets I (feature RSS vector reported by mobile device) and Y (encoded location information). Map Field is inter-ART module that controls the learning of an associative map from ART$_a$ recognition categories to ART$_b$ recognition categories. This is achieved by connecting F$_2$ Layer, so called ${F_2}^b$, neurons of ART$_b$ to Map Field nodes with one-to-one non-adaptive links in both ways. On the other hand each F$_2$ layer, referred to as ${F_2}^a$, neuron of ART$_a$ is connected to all Map Field nodes via W_mp adaptive links. Since Map Field represents a mapping from both ${F_2}^a$ and ${F_2}^b$, it is denoted as F^{ab}. This map does not directly associate feature vectors with encoded class labels but rather associate the compressed codes of groups of I and Y.

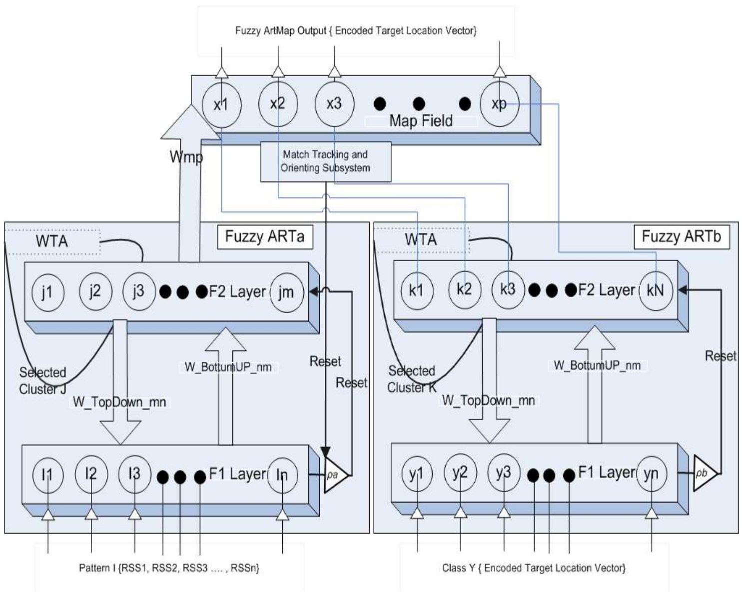


Figure 4.4: Fuzzy ArtMap Structure

During learning of pattern-class pairs if a mismatch occurs at Map Field between ART$_a$ category and ART$_b$ category then system increases vigilance parameter of ART$_a$ so that ART$_a$ can categorize this pattern in different category. This mechanism allows network to capture novel features that can be incorporated through learning new ART$_a$ recognition category. Activation of Map Field results in output signal from each F^{ab} node, a vector cor-

Algorithm 2 The Self-scaling Fuzzy Art Online Learning Algorithm

for Each Input I in RSS^d **do**
  Apply Range Normalization on I
  Apply Complement Coding: $I = \{I, I^c\}$
  for Each Component i in I **do**
    Apply I to F1 Layer: $F_1(i) = I(i)$
  end for
  if F_2 is *empty* **then**
    Create new F_2 Node j
    for Each Node i in F_1 **do**
      Create New Synaptic Connection S
      Initialize Weight: $W_i = I_{(i)}$
      Connect S to i and j
    end for
  end if
  for Each Node j in $F2$ **do**
    Evaluate Choice Function: $T(j) = |I(j) \wedge W_{td}(j)|/\alpha + |W_{td}(j)|$
  end for
  Apply WTA(Winner Takes All): $T_J = \max T(j)$
  Orientate: $W_J = T_J(W)$
  Calculate the Size of the Hyper box: $s = |I \wedge W|/|I|$
  if Vigilance violates: $s \geq \rho$ **then**
    $T_J = -1$ and Goto Apply WTA:
  else
    Turn off all F_2 Nodes except J: $T(j) = 0 \quad \forall \quad j \neq J$
    Update Weights: $W(j) = \beta(I \wedge W(j)) + (1 - \beta)W(j)$
  end if
end for

responding to target location, that eventually becomes out put of Fuzzy ArtMap network. Learning RSS-location pair occurs if Fuzzy ArtMap network is presented with both RSS input vector I and target location vector Y. Location estimation occurs in case only RSS input vector is presented to the system. Activation of F^{ab} occurs both in case of learning mode and estimation modes. Match tracking and orienting subsystem allows Fuzzy ArtMap network to establish different categories for similar RSS inputs at ART_a as well as allows very different RSS inputs to form categories that belong to same location. This is achieved by activating orienting subsystem only when ART_a makes a location estimate that does not confirm with actual location provided to ART_b. This condition starts match tracking by adjusting ART_a vigilance parameter in such a way that estimation error is removed. A simplified version of Fuzzy ArtMap is adapted, presented in [56], which employs only one Fuzzy Art optimized for hardware implementation of Fuzzy ArtMap system. Simplified Fuzzy ArtMap exhibits same learning and recall performance as original Fuzzy ArtMap and its learning algorithm is shown in [56].

4.5 ConSelFAM

Context-aware, Self-scalable Fuzzy ArtMap extends original Fuzzy ArtMap neural network system in order to realize *on the fly* location system development and reconstruction which is aware of visibility clusters present in signal space. *On the fly* development essentially refers to capability of classifier to learn RSS to Location association at the time a signal is detected by mobile device.

4.5.1 Self-scaling Fuzzy Art

In original Fuzzy Art, learning capacity of system need to be fixed in order to initialize a network. All F_2 layer neurons are said to be *uncommitted* before learning starts and as soon as a new input, which was not encountered until that point, is sensed by system an *uncommitted* F_2 layer neuron becomes *committed* to represent this category onwards. This process goes on until all patterns in input space are categorized or learning capacity is reached. The *don't know* response feature of Fuzzy Art systems is realized by means of fixed capacity. But, at the same time, fixed capacity puts *extensibility* limitation on Fuzzy Art which implies that system cannot learn beyond a certain number of categories. As discussed in section 4.2.2, *extensibility* is very basic function that location estimation system should perform. The Self-scalable Fuzzy Art learning algorithm is presented in 2. This self-scaling variant preserves *don't know* response feature of original Fuzzy Art but in this scheme network capacity is externally adjustable, increment-only parameter that can be increased as a network is required to learn more categories. Main difference in original Fuzzy Art and self-scaling Fuzzy Art is in network initialization method. Self-scalable Fuzzy Art algorithm initializes a network without any F_2 layer neurons and include new F_2 neurons by means of self-scaling subsystem. This way if capacity is reached but more learning is required then a neuron is incorporated dynamically. The original Fuzzy Art is replaced with self-scalable Fuzzy Art in order to enable Fuzzy ArtMap network to learn new RSS patterns as they appear as well as to learn new locations to achieve the *extensibility* defined by quation 4.3.

4.5.2 Context-awareness

In order to achieve *expansibility* in scope, a classifier must be able to learn different input spaces without confusing different classes or locations as same due to similar patterns. Moreover, a signal input space might be partitioned into several subspaces due to physical properties of signals as described in section 4.3.3. Therefore a particular subset M_i of total access points represents a subset of all locations, referred to as visibility cluster, where M_i is always visible. This implies that each set of access points in Visibility Matrix is important *context* that can improve pattern recognition capability. Therefore, instead of training a separate pattern recognition module for a each visibility cluster the ConSelFAM incorporate this contextual information into one classifier thus making it *context-aware*. By context-awareness it means to equip a classifier with specific domain knowledge such that it can differentiate among different input spaces. This capability is realized in Fuzzy ArtMap system by defining a mechanism to embed contextual knowledge, visibility cluster in our application, into classifier. A Context Field subsystem is introduced into Fuzzy ArtMap neural network as can be seen in Fig. 4.5 which enables system to distinguish between different contexts thus enhancing its learning as well as generalization capability.

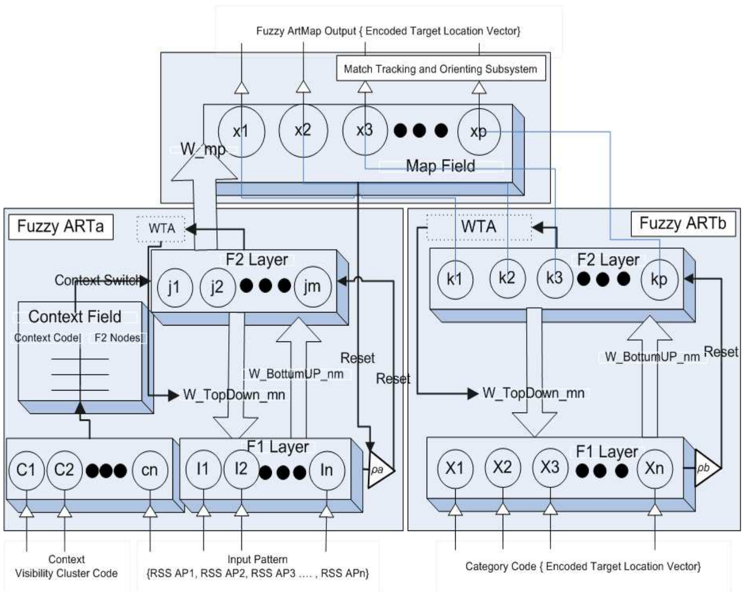


Figure 4.5: Context-aware, Self-scaling Fuzzy ArtMap Structure

In this scheme two types of inputs are presented to network a) classification context code b) input pattern. Classification context code is defined by Equation 4.2 as scope of a location system. Thus context-code in our application is visibility status of different access points. This information governs the further learning and recall operations of system. Learning dynamics of Context-aware Fuzzy ArtMap are similar to original one except that

Algorithm 3 The Fuzzy ArtMap Online Learning Algorithm

Initialize Fuzzy Art: $W_{nm} = 1$

Initialize MapField: $W_{mp} = 1$

Initialize Vigilance Parameter: ρ = r

Present RSS Input I and Location Y pair

Find F_2 Node M to represent I (Algorithm 2)

repeat

  if No representative F_2 found **then**

    Increase vigilance: $\rho = |I \wedge W|/|I| + \epsilon$

    Find F_2 Node to represent I (Algorithm 2)

  end if

until A representative F_2 node M is found

Update Weights: $W(nm) = \beta(I \wedge W(m) + (1 - \beta)W(m)$

Update Map field weights: $W_m(new) = Y \cap W_{mp}(old)$

Initialize Vigilance Parameter: $\rho = r$

Algorithm 4 The Context-aware, Self-scaling Fuzzy ArtMap Online Learning Algorithm

Initialize Fuzzy Art: $W_{nm} = 1$ and ρ

Initialize MapField: $W_{mp} = 1$

Present Input I, Context C and Location Y

if Context C Exists **then**

  Load Context:

  Find F_2 Node M to represent I (Algorithm 2)

  repeat

    if No representative F_2 found **then**

      Increase vigilance: $\rho = |I \wedge W|/|I| + \epsilon$

      Find F_2 Node to represent I (Algorithm 2)

    end if

  until A representative F_2 node J is found

  if F_2 Node M is *new* **then**

    Associate M with C

    Update Map field weights: $W_{mp}(new) = Y \cap W_{mp}(old)$

    Associate M with Y

  else if Map field Category is not associated with M **then**

    Increase ρ and research

  else

    Update Map field Categories

    Update F_2 Node

  end if

end if

it maintains contextual knowledge as a special hash table. As a particular set of access points, visibility cluster, is detected by the device a context-switching happens inside the network. This switching directs the network to perform *choice function* and *vigilance check* on a particular set of F_2 nodes which belong to this context. Once a particular context is loaded next operations of learning or recall take place the associative learning of Map Field connections ensues. The implementation of Self-scaling, Context-aware Fuzzy Art and Fuzzy ArtMap networks is done in $C\#$ programming language and it is available as open source [1].

4.6 Experimental Design

4.6.1 Preprocessing

Adaptive Resonance Theory based learning systems suffers from category proliferation problem as characterized by Moore [11]. In order to overcome this problem a data preprocessing technique, namely Complement Coding, is proposed by Carpenter et al in [16]. Besides overcoming category proliferation problem, this technique allows network to reduce effect of presentation frequency of an input pattern as well as order of presenting input patterns to Fuzzy ArtMap, as explained in [15],[17]. The Fuzzy ArtMap models were developed with and without complement coding in order to evaluate its efficacy in location estimation problem. Complement coding requires input pattern values to fall in range of 0 to 1 but actual RSS values range between -10 dBm to -100 dBm. A scaling normalization was applied on raw RSS input vector so that all values are transformed in range of 0 to 1. Same preprocessing is applied for Multi Layer Perceptron location classifiers in both training and test phases. The location estimation error is measured similar as described in previous chapter.

Fuzzy Art Map

Table 4.6.1 presents training results of Fuzzy ArtMap network with and without complement coding. Fuzzy ArtMap model trained with complement coding is denoted as FAM-CC and without compliment coding is denoted as FAM.

Table 4.1: Fuzzy ArtMap Results on Training

	FAM-CC	FAM
F2 Clusters	36	167
MAE	.066	.018
Unclassified	0	11
Miss classified	.02	.006
$e_r \leq 1$	.02	.006
$e_r \leq 2$	.02	.006
$e_r \leq 3$	0	0

Complement coding controls category proliferation problem and classifies all RSS vectors successfully but classification performance is slightly affected. Training Fuzzy ArtMap without complement coding results in higher accuracy but increases indecisiveness, *don't know* response in terms of unclassified RSS patterns, as well. During testing phase, This problem aggravated and FAM could not classify 389 patterns.

ConSelFAM

For Context-aware, Self-scalable Fuzzy ArtMap experiments the visibility matrix was generated in data collection phase. Each tuple of this matrix represents a cluster of locations where a subset of access points are always visible as described in section 4.3.3. Thus each cluster corresponds to separate classification *context.* Table 4.2 presents training results of for each context of classification.

Table 4.2: ContextAware Fuzzy ArtMap Results

Context	F2 Clusters	Epochs	Training MAE
5	20	3	0
4	27	3	0
3	36	3	0
2	29	3	0
1	26	3	0

Although ConSelFAM requires viability matrix to be known before learning starts, it requires less memory resources in terms of F_2 layer clusters than Fuzzy ArtMap. More importantly it does not require radio map creation and converges to optimal error in just 3 epochs, contrary to *offline* training based methods, which takes most of development time in previous approaches. Modular classifier approach improves overall location accuracy but depends on visibility matrix to be established as prerequisite. Contrary to other methods Fuzzy ArtMap and ConSelFAM system takes only 3 epochs to achieve 0 MAE.

4.6.2 Test Results

Here comparative testing results of different classifiers using same test radio map are presenterd. Both absolute and relative error measurements for each classifier are given in the following.

Table 4.3 shows performance of ConSelFAM for each visibility cluster or *context*. The same information for modular MLP and modular LVQ methods is provided in previous chapter. Notice that relative error performance of ConSelFAM is better than both mMLP and mLVQ which means that estimated location deviation from actual location is mostly limited to neighboring locations.

Table 4.3: ConSelFAM Results

Context	Test MAE	$e_r \leq 1$	$e_r \leq 2$	$e_r \leq 3$
1	0.85	85%	90%	92%
2	0.79	83%	100%	100%
3	0.56	99%	100%	100%
4	1	95%	99%	100%
5	0.42	100%	100%	100%

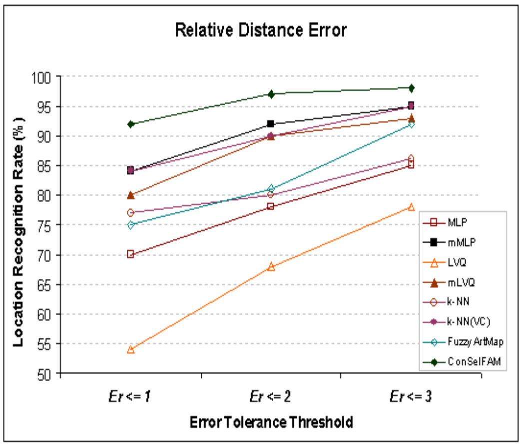
Table 4.4: Summarized Comparative Results on Test Radio Maps

Method	MAE	$e_r \leq 1$	$e_r \leq 2$	$e_r \leq 3$
FAM-CC	1.06	75%	81%	92%
ConSelFAM	0.72	92%	97%	98%
MLP	1.03	75%	79%	91%
mMLP	0.63	83%	91%	93%
LVQ	2.60	55%	69%	80%
mLVQ	0.79	79%	89%	93%
k-NN	0.73	77%	80%	86%
k-NN(VC)	0.43	84%	90%	95%

ConSelFAM showed best relative error performance which is consistent in training and test phases. Although LVQ benefits most significantly from *visibility* based modularity, overall performance of this method was lesser than other methods.

4.7 Summary

ConSelFAM is suitable for several classification problems which require context-awareness and scalability. An RSS based location system in real environment is developed to confirm



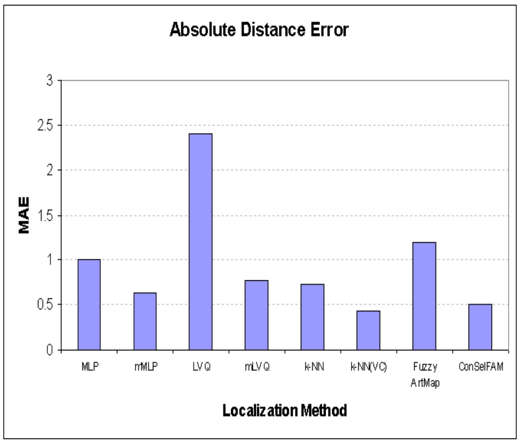


Figure 4.6: Right: Relative Distance Error, Left: Absolute Distance Error

the applicability of ConSelFAM pattern classification system. The location estimation performance is compared with other classification methods such as Multi Layer Perceptron, Learning Vector Quantization and Modular variants. On the basis of extensive experimental results, it is concluded that ConSelFAM provides competitive location estimation accuracy as well as leverages novel features which can not be realized using previous methods such as 1) *Rapid system development and Reconstruction* 2) *Flexible and dynamic expansion of system.*

Chapter 5

Semantically Meaningful Localization

This chapter presents a new methodology, Beacognition, for real-time discovery of the associations between a signal space and arbitrarily defined regions, termed as Semantically Meaningful Areas (SMAs), in the corresponding physical space. It lets the end users develop semantically meaningful location systems using standard 802.11 network beacons as they roam through their environment. The key idea is to discover the unique associations using a beacon popularity model. The popularity measurements are then used to localize the mobile devices. The beacon popularity is computed using an 'election' algorithm and a new recognition model is presented to perform the localization task. Such a location system is implemented in a five story campus building. The comparative results show significant improvement in localization by achieving on average 83% SMA and 88% Floor recognition rate in less than one minute per SMA training time.

5.1 Introduction

Location awareness is a key enabling technology for ubiquitous computing spaces. Although satellite based localization, e.g. GPS, is a defacto positioning method; many researches have pointed out its shortcomings such as low availability in the indoors and the requirement of special hardware [26, 45, 44, 65]. The potential ubiquitous computing applications can be realized by embedding the sense of location into commodity devices connected through the modern communication networks. There have been several branches of this research which utilize different radio communication networks for localization such as FM radio [44], GSM channels [18], and 802.11 access points [26, 2].

Most of these systems strive to estimate the actual location of the mobile device in terms of geometrical distance. However, as reported in [40], the geometrical definition of location is not suitable or required for many emerging indoor location enhanced applications. Instead such applications perceive a location as an 'area of interest' identified by everyday names e.g. Gents Garments Section, Food Court or Customer Service Center in a super market. The 'area of interest' which can have arbitrary boundaries serves richer semantic value and provides a defining block of a location system.Such an area is referred to as *Semantically Meaningful Area* (SMA). Two main categories of several applications of such localization scheme are: i) multimedia content adaptation systems, e.g. tour guides, advertisement and collaborative games, and ii) high level activity recognition for context-aware ubiquitous computing spaces. Some studies have demonstrated such localization via beacon identification [26, 40, 60] or using probabilistic modeling [32, 38]. However, these methods require the prior information such as beacon positions, target locations or propagation models to train a localization algorithm. Such a localization scheme results in longer development time. Moreover, multi-floor environments pose a *resembling signal space* phenomena [31] which renders previous approaches ineffective to distinguish between adjacent floors.

The main contribution is a new semantically meaningful localization methodology, referred to as radio **Bea**con signal measurement and re**cognition** (Beacognition), which includes two components. First, an interactive, election algorithm which discovers the best representative beacons for an SMA. This discovery is performed online by presenting the beacon signatures to the algorithm. Second, a new location recognition method which computes the most probable location of the mobile device based upon an intuitive model. Beacognition is employed to build a semantically meaningful location system using IEEE 802.11 standard network beacons in a real multi-floor environment. In general, this methodology is equally beneficial for localization in other wireless networks.

The Beacognition targets a coarse resolution but semantically meaningful location system in multi-floor buildings. The salient features of this methodology are as follows: i) it provides interactive development scheme to facilitate location system development, ii) no requirement of any prior knowledge lowers the entry barrier for location system developers, iii) the location to signal space mappings are discovered while a device is roaming through the environment.

The intuitive reason behind better localization performance of Beacognition has two aspects: i) unlike other approaches of using all detectable beacons as references, the Beacognition discovers only the best-representative beacons and their features via the Election algorithm. Apparently, using all detected beacons as references may be sufficient to distinguish different locations. However, due to the noise in indoor environments and resembling signal space problem, this approach faces difficulty to accurately recognize different locations, and ii) the Beacognition SMA recognition model measures the signal strength as well as ranking similarity between the detected beacon set and the reference beacons. Moreover, the ranking similarity measure gives weight not only to the rank of a detected beacon but also to the missing beacons. This approach better resolves the confusion between similar points in signal space representing different points in physical space.

5.2 Beacon Based Localization Primer

The core issue of inferring location information from radio beacons is to discover the association between a signal space and respective physical space. The area where the signal of an identifiable beacon can be detected is referred to as *Signal Coverage Area* (SCA). A device can infer its location whenever it receives signal from a particular beacon by searching in a list of SCAs. The boundary of an SCA depends on several factors, e.g. propagation environment, transmitter power and frequency band. The meaningfulness of an SCA boundary may often not coincide with the semantic requirements of an indoor location based service.

Multiple neighboring radio beacons, typical of urban and indoor environments, create overlapping signal space which can leverage important clues for localization. The overlapping signal space exhibits two important complementary facets for inferring location information.

1. The single beacon Distinguishable Signal Coverage Area (sDSCA) is the area where signal of multiple beacons can be detected but the signal intensity of a beacon remains stronger than others. A device can be straightforwardly localized with the location of a beacon with the strongest strength.

2. The multiple beacon Distinguishable Signal Coverage Area (mDSCA) is created by overlapping signal space of more than two beacons. It represents more complex boundaries in the physical space given that the appropriate associations between signal and physical space are discovered. The localization is performed by comparing the membership similarity between set of detected beacons v with the all mDSCAs in ($mList$). Then the maximally similar mDSCA to v represents the location of the device as following.

$$l = \arg\max_{i \in N}(\|v \cap mList_i\|) \tag{5.1}$$

It is understandable that all locations identifiable by beacons may not be the SMA for a location enhanced system. Ideally, the semantic needs of a location based application system should define the boundaries of an SMA. The formation of best representative beacons and their related properties is referred to as Semantically Meaningful Area Recognition Template (*smart*).

In multi-floor environments, beacon based localization face a resembling signal space at vertically similar areas on adjacent floors. This phenomenon is also observed in a cross floor signal propagation study for three radio frequencies in a multi-floor indoor environment [31]. The authors identify that, in certain situations, signal levels tend to remain quasi-constant in adjacent floors. Furthermore, the signal attenuation for a single floor separation is often lower than same-floor signal attenuation. These cross floor propagation characteristics implicate the localization task and require special care for discovering *unique* associations as well as adequate similarity measures for accurate recognition.

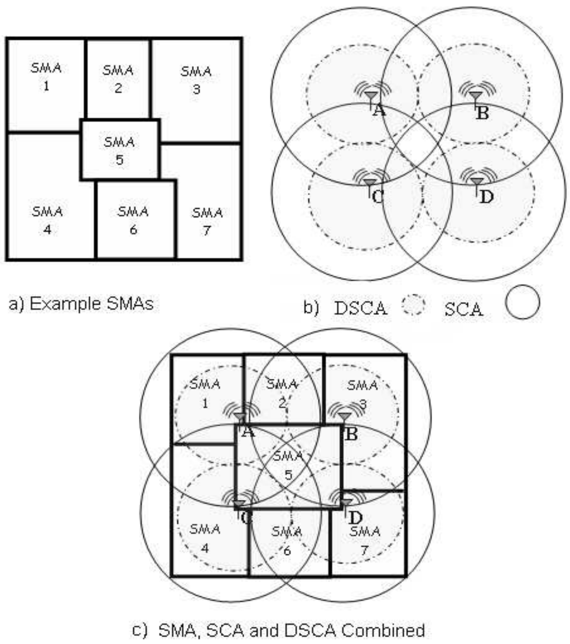


Figure 5.1: Example of $SCA, DSCA$ and $gDSCA$

5.3 The Election Algorithm

The algorithm views all detectable beacons in an SMA as contestants of an election, hence termed the election algorithm. It ranks all contestants based upon the votes they 'win' from different points in an SMA. Once a valid *smart* is discovered, the less representative beacons are pruned out so that only the winner survives. Two important operating conditions of this algorithm distinguish it from the conventional localization algorithms: i) the signal to location mapping is created in real-time while the device is roaming in the target area and ii) no prior information about beacons, e.g. beacon ID, position or radio map, is available.

5.3.1 A Motivating Analogy

The intuition behind this algorithm derives from an analogy with the political election process in the real life where multiple candidates contest for winning the representative offices (*smart*) of a constituency (SMA). The beacons are analogous to the candidates and the individual points in physical space are the voters. The polling results produce a ranking which reflects the degree of representativeness, or popularity, of each candidate such that highest office is awarded to the candidate who wins maximum votes from a sample population in that constituency, then the next office is given to the next highest winner.

The polling process of Election algorithm differs from conventional ones in the time of voting and results computation and duplication of votes. In ordinary elections, polling takes place simultaneously at different locations of a constituency at a given time and the results are compiled afterwards. This polling method is applicable because all the contestants know their constituencies. However, consider a situation in which candidates do not know their constituencies but an election must be held. In the same sense, The aim is to 'discover' appropriate beacon representatives for SMAs while ID or position of any beacon is not known. Owing to the special nature of the task, the algorithm conducts polling sequentially and results are compiled on the fly as the voting continues. Allowing duplicate votes is another deviation from the normal elections. However, since the main objective is to facilitate end users to develop location systems, it is not possible for a developer to visit certain pathway only once. Therefore the algorithm detects the duplicate voting internally and only unique *smart* formations are created.

5.3.2 Notations Involved

A brief description of notations is required before the learning dynamics of Election algorithm are explained. It concerns objectification of two concepts i) a radio beacon and ii) a set of beacons. Every object possesses some properties and provides interfaces to access those properties. I use $A_{[b]}$ notation to represent each group where A is the group name and b indexes over group members. Each group provides interfaces to perform some basic operations onto the group. These interfaces are mentioned using "." after the concerned object such as; $A.Count$ gives total members of the group, $A.Contains(beacon)$ answers group membership enquiry and $A.Add/A.Remove$ allows adding/removing beacons to/from a group.

An individual beacon is represented as $B_{(d,nd,dP,x)}$ possessing four properties. Algorithmic description uses boldface to show value of any of these properties e.g. $B_{(\mathbf{d},nd,dP,x)}$ gives value of d. The d holds total number of times the beacon is detected since learning started, nd is number of times a beacon was not detected, dP is the detection persistence and x is a boolean flag which indicates the system to remove a beacon from the candidates set.

Listing 5 shows different sets of beacons as well as related representations of computations will be explained while discussing the learning dynamics.

Listing 5 Notations concerning Algorithm [6]

Parameters

$m\tau$: Missing Beacon Tolerance

τ: Limit on Size of a SMART

Define Global

$C_{[c]}$: Collection of beacons appear in latest scan

$L_{[l]}$: Collection of beacons appeared in last scan

$M_{[j]}$: Collection of missing beacons

$T_{[k]}$: Collection of beacons trail

$ChangeDetected$: A boolean flag

S_c: Scan count

m: Consistently missing beacons

5.3.3 The Learning Dynamics

The Election algorithm forms conceptual groupings of beacons and continuously performs simple set operations on these groups in order to finally discover the $gDSCA$. Membership of these groups is bound to change as the moving device keeps on scanning the beacons from place to place. Algorithm 6 provides an abstract description of operational flow which is composed of three main steps.

Scanning Beacons

In the first step, at an arbitrary location a the network is scanned to detect the beacons who claim to be the representatives of that location. Upon each scanning operation, or polling call, all detected beacons are grouped as latest scan set, $C_{[c]}$, and the previously scanned beacons are assigned to another group denoted as $L_{[l]}$. Objective of this grouping is twofold; i) Detecting change in signal or location space ii) Tracking detection/absence of an individual beacon. Detecting change in location or signal space is important from sys-

Algorithm 6 The Election Algorithm

Define Variables (Listing 5)
Scan Beacons and Track Detections
{see procedure 7}
while $ChangeDetected$ is 1 **do**
    Determine detection/absence persistence
    {see procedure 8}
    if $m > 0$ **then**
        Remove insignificant beacons From T
        {see procedure 9}
        if $T.Count \leq \tau$ **then**
            SMART $\leftarrow T$
        end if
    end if
end while

tem development point of view. Considering the election analogy, for the sake of fairness, it is necessary that at each location one beacon casts only one vote. Which requires that equal number of scanning operations should be performed at all locations. However, it is very unlikely that the human carrier of learning device will keep a consistent speed. The computations can get biased towards detected beacons in case carrier stays at a location longer than the other. However, if scanning/voting results are similar at two adjacent locations then it would not affect the final results. Based upon this intricacy, I repose the issue of fairness by changing the equal vote counting condition to only dissimilar vote counting. It means that system learns only when there is a change in signal space. This change is important even if carrier is stationary or mobile at scanning time. It eliminates the consistent speed constraint from system developer as well as redundant operations. On the other side, Even though location is changed but there is no change in signal space then no learning shall take place. Machine perceives changed signal space in three cases when; i) a new beacon appears in $C_{[c]}$ ii) a beacon is missing in $C_{[c]}$ which was detected in $L_{[l]}$ iii) both (i) and (ii). The record of respective detected and absent beacons are updated once change in signal space is found. The system maintains two other groupings as well; 'beacon trace' $T_{[k]}$ and missing beacons $M_{[j]}$. The $T_{[k]}$ is the superset which contains all beacons which get detected since learning started. While $M_{[j]}$ are all the beacons which appeared at some point but vanished later. Besides updating detection/absence record, system adds newly detected beacon of case (i) to the T and missing beacons of case (ii) to M. Clearly, no change in $T_{[k]}$ and $M_{[j]}$ occurs if the signal space remains unchanged.

$$T_{[k]} \cup C_{[c]} \setminus T_{[k]} \Rightarrow \{c : c \in C_{[c]} and c \ni T_{[k]}\} \tag{5.2}$$

$$M_{[j]} \cup T_{[k]} \setminus C_{[c]} \Rightarrow \{k : k \in T_{[k]} and k \ni C_{[c]}\} \tag{5.3}$$

Procedure 7 Scan Beacons and Track Detections

if $L_{[l]}$ is $C_{[c]}$ **then**
    $ChangeDetected \leftarrow 0$
    $Stop$
else
    $ChangeDetected \leftarrow 1$
end if
for c^{th} beacon b in $C_{[c]}$ **do**
    $b_{(\mathbf{d++},nd,dP,x)}$
    if $\neg T.Contains(b)$ **then**
        $T.Add(b)$
    end if
end for
for k^{th} beacon b in $T_{[k]}$ **do**
    if $\neg C.Contains(b)$ **then**
        $b_{(d,\mathbf{nd++},dP,x)}$
        $M.Add(b)$
    end if
end for

Detection/Absence Persistence

In case of change in signal space, the learning continues to the second step. The distinguishing nature of the election method is to compile the results while voting is taking place in a continuum. This requires watching the candidates who are always or mostly detected. This information is captured as detection persistence dP of each individual beacon since the first scan (voting) took place. The dP is a measured as the ratio of detection count and total number of scans (or voting calls).

$$dP = \frac{Detectioncount}{Totalnumberofscans} \tag{5.4}$$

The system computes dP for each beacon who is member of $T_{[k]}$ and detection count S_c denominator is incremented for next round of voting. Notice that if the detection of change in signal space has subtle impact on dP as well. If device stays at a location for extended period then temporal absence of a beacon at that location can cause unfair drop in dP which results in weakening its candidacy for becoming the representative. The temporal absence of beacons is commonplace phenomenon especially in indoor environments. It can be observed in figure **??**, which shows a snapshot of the real data about detection of beacons in one of the target fields. This happening can also cause abrupt removal of a beacon from candidates set and When a beacon appears again after short absence all its previous reputation, in terms of dp, is lost. A cushion is provided to overcome this potentially perturbing situation. The $m\tau$ is an externally specifiable parameter which allows the system to tolerate temporarily missing beacons. Due to this mechanism an absent beacon remains in beacon trace superset $T_{[k]}$ until it is consistently not detected more than $m\tau$ times. Even though it slowly degrades beacon reputation but prohibits abrupt removal of a legitimate beacon. Once a beacon is not detected even for extended time, system marks it as removable from the beacon trace as a natural consequence. Besides marking, all such beacons increment the removable beacon count m so that further procedures can take place.

Procedure 8 Computing Detection/Absense Persistence

$m = 0$
if $changeDetected$ **then**
  for k^{th} beacon b in $T_{[k]}$ **do**
    $b_{(d,nd,\mathbf{dP},x)} \leftarrow b_{(\mathbf{d},nd,dP,x)}/S_c$
  end for
  $S_c \leftarrow S_c + 1$
  for j^{th} beacon b in $M_{[j]}$ **do**
    if $b_{(d,\mathbf{nd},dP,x)} > m\tau$ **then**
      $b_{(d,nd,dP,\mathbf{x})} \leftarrow 1$
      $m = m + 1$
    end if
  end for
end if

Removal consolidation

The final goal of first two steps was to segregate the overlapping beacons in two sets; i) The beacon trace $T_{[k]}$ which gives a ranking of all contestants according to their reputation measured as dP and ii) The missing beacons set $M_{[j]}$ which is a set of beacons who could not qualify as legitimate candidates. When one or more beacons are perceived as missing by the system, the removal consolidation ensues for a possible SMART formation in third step. At this stage Election algorithm enacts four further sub groupings of beacons trace based upon there detection (as well as absence) persistence reputation; SMART, 'To be removed' $R_{[i]}$, Immature beacons $iM[i]$, Weak beacons $wK_{[i]}$ and Missing beacons. The dP divides missing beacons into two subgroups; i) 'To be removed' and ii) Conclusive. Each of them have opposite role to play. Distribution of these groups with respect to dP is shown in figure 5.2.

Cautious creation

The 'to be removed' missing beacon set initiates the removal process so that the representative offices should be assigned to legitimate SMART. However, system takes a cautious approach to avoid redundant creation of $gDSCA$ where largely similar SMART represent nearby locations. As a by product, this approach gives another chance for make up to the temporarily absent beacons. This approach delays SMART creation until members of all other groups are less than SMART size τ.

$$(iM.Count + wK.Count + R.count) \leq \tau \tag{5.5}$$

Immediate creation

System can encounter a situation which calls for urgent creation of SMART. It occurs when the members of 'conclusive set' who won majority votes ($dP > .7$) but are not available for this location onwards anymore. Therefore it immediately eventuates the creation of a reliable $gDSCA$.

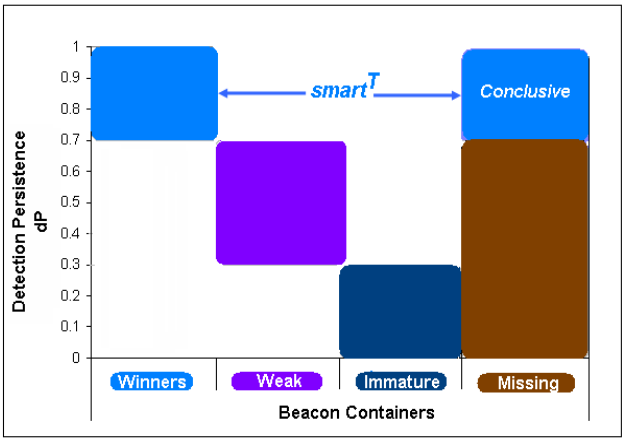


Figure 5.2: The Beacon Containers: decomposition of the beacon trace

The discovery of $gDSCA$s is governed by two parameters i) $m\tau$ missing beacon tolerance and ii)τ SMART Size limit.

Beacon Removal and Transitive *smart*

The decomposition of the T_b into beacon containers ensues the removal of missing and less-representative beacons from the system. At this point, the system observes another externally specified constraint of maximum smart size denoted as τ:

$$\|T_b\| - (\|m_B\| + \|w_B\| + \|i_B\|) \leq \tau \tag{5.6}$$

It is observed that, in many cases, fewer beacons represent large gDSCAs and more beacons represent small gDSCAs [2]. This constraint further delays the final creation of a *smart* giving another chance to the temporarily missing beacons to make up. There are two other situations when the system creates a transitive smart: i) when a conclusive beacon is detected to be missing, and ii) the developer intrudes the polling process to explicitly

Procedure 9 Removal Consolidation

```
$R_{[i]}$: Collection of 'to be removed' beacons
$iM_{[i]}$: Collection of immature beacons
$wK_{[i]}$: Collection of weak beacons
$iC$: Insignificant beacons count
for $j^{th}$ beacon $b$ in $M_{[j]}$ do
    if $b_{(d,nd,dP,\mathbf{x})}$ is $1$ then
        $R.Add(b)$
    end if
end for
for $k^{th}$ beacon $b$ in $T_{[k]}$ do
    if $b_{(d,nd,\mathbf{dP},x)} < .40 \wedge b_{(d,nd,dP,\mathbf{x})}$ is $1$ then
        $iM$.Add(b) and $iC = iC + 1$
    else if $b_{(d,nd,\mathbf{dP},x)} < .70 \wedge b_{(d,nd,\mathbf{dP},x)} > .40 \wedge b_{(d,nd,dP,\mathbf{x}}$ is $1)$ then
        $wK$.Add(b) and $iC = iC + 1$
    end if
end for
smart Formation: Procedure10
```

Procedure 10 *smart* Formation

```
if (T.Count − iC) ≤ τ then
    for i^th beacon b in R_[i] do
        T.Remove(b)
    end for
    if (T.Count > τ then
        if (T.Count − iM.Count) ≥ τ then
            for i^th beacon b in iM_[i] do
                T.Remove(b)
            end for
        else
            Remove only τ − (T.Count − iM.Count)
        end if
        if (T.Count − wK.Count) ≥ τ then
            for i^th beacon b in wK_[i] do
                T.Remove(b)
            end for
        else
            Remove only τ − (T.Count − wK.Count)
        end if
    end if
    Form new SMART with the T
else
    Continue
end if
```

create an SMA. Once the second case happens, system tries to find the best representative beacons and create a corresponding *smart* immediately. However, this intrusion is treated as just a suggestion. If there is no corresponding distinguishable signal space, then, the system ignores this suggestion. Ultimately, the remains of t_B are the ones who have won the election in *this* SMA. At this point, these beacons are referred to as transitive *smart*^{T}.

Detection of a Redundant *smart*

In order to avoid the redundancy, before finalizing the *smart*^{T} to be a *smart*, the system ensures that a similar set is not already existing. A similar *smart* could have been formed previously in two cases.

Case 1. The developer is roaming in the same SMA even after detection of a valid *smart*. This case can be detected by measuring the similarity between *smart* S_a and *smart*^{T} S_b using Dice's coefficient [22] denoted as d^C. It measures asymmetric information in two sets which may contain dissimilar elements. The d^C reflects the weight of common elements in two *smart*s, S_a and S_b, as follows:

$$d^C = \frac{2\,||S_a \bigcap S_b||}{||S_a|| + ||S_b||} \tag{5.7}$$

Case 2. The developer has reentered in an SMA which is represented by an already valid *smart*. This case requires more careful treatment for the redundancy detection because only d^C can cause *smart*^{T} to be regarded as redundant due to the similar signal space on adjacent floor. The election algorithm measures this similarity as an average of signal strength confidence and d^C. Measuring the signal strength confidence is explained in section 5.4.1.

In either case, if the similarity between S_a and S_b is more than certain threshold, 0.70 in the experiments, then *smart*^{T} is considered to be redundant thus ignored. However, since the device is roaming in the same SMA, the statistical information of common beacons is updated.

Creation of smart

After passing the redundancy check, the system creates a *smart* by associating the final set of beacons and their statistical information with *this* SMA. The control shifts to scanning state if developer wants to continue otherwise it stops at this point. The structure and example members of a smart are shown in table 5.3.3.

Table 5.1: Structure of a *smart*.

smart			
SMA	Recognition Template		
SMA name	MAC	SS Mean	SS Variance
this SMA	4259	-47	1.3
	5659	-53	2.4
	9235	-69	1.1

5.4 The Location Recognition Model

The second component of the Beacognition methodology is to recognize the location of a query beacon signature. The influence of the election analogy does not end at the discovery of *smart*s and formation of SMAs. The final localization decision is again made based upon a voting strategy. At that time, a device puts forward the detected beacons as location query v to all representatives, *smart*s , for finding its appropriate location. The member beacons of respective *smart*s vote their confidence to represent the query beacon set v. The location recognition model evaluates the vote of confidence of each *s* in all *smart*s S. Consequently, the location of querying device is inferred as the SMA of the winner *smart* w who is most confidant to represent v as follows:

$$w = \arg\max_{s \in S} f(s, v) \tag{5.8}$$

where $f(s, v)$ is computed as an average of two components of the recognition model as discussed next.

5.4.1 Signal Strength Confidence

Signal strength of a beacon carries important clue for resolving confusion about the correct location among neighboring SMAs. A simple signal strength similarity measurement formula is devised to compute the combined impact of all beacons in a *smart* on the localization within [0,1] range.

$$\begin{aligned} & sv^S = \textstyle\sum_{i=1}^{n} \frac{1}{n} \quad where \quad n = |s| \\ & \Rightarrow \left[i | \forall \left(s_i = v_i \right) \in \left(s \cap v \right) and \left(\frac{\left| v_i^{ss} - s_i^{\mu} \right|}{s_i^{\sigma}} \right) \leq 2 \right] \end{aligned} \tag{5.9}$$

It assumes that within an SMA, the spread of signal strengths follows a normal distribution. By definition, a standard normal distribution can explain 95.4% of the variation within 2 standard deviations from the mean. Therefore, if the normalized signal strength falls in the range [-2,2] then it is considered to be in favor of the *smart* s.

5.4.2 Ranking Confidence

The ranking confidence sv^R is measured on signal strength based ranking of *smart* s' and query beacon signature v'. The sv^R measures it in the range of 0, for no confidence, to 1, for high confidence. The ranking similarity measurement needs special treatment because of two common situations which are ignored in the previous works: i) the s' and v' may contain different beacons, and ii) the number of beacons in both signatures are different. First the difference between two sets is measured as a fraction of ‘excitation’ and (*Ex*) and 'inhibition' (*In*) of the differences in ranking. Subtracting the total difference from 1 gives the ranking confidence as given in equation 5.10.

$$sv^R = (1 - \tfrac{Ex}{In}) \tag{5.10}$$

Let s_i and v_j denote the individual beacons in s and v where i and j are their respective ranks. The Ex is calculated as:

$$Ex = \left\{ \begin{array}{lll} Ex + |i - j| & if & s_i = v_j \in c \\ Ex + 1 & if & v_j \in d \end{array} \right\} \tag{5.11}$$

where $c = s \cap v \quad and \quad d = s - v$. The In is calculated as:

$$In = \left\{ \begin{array}{lll} In + (i + j + (k - i)) & if & s_i = v_j \in c \\ & and & i = j \\ In + (i + j) & if & s_i = v_j \in c \\ In + 1 & if & v_j \in d \end{array} \right\} \tag{5.12}$$

where $k = \|s\|$. The *Ex* is the summation of all differences in the rank of common beacons in s and v as well as uncommon beacons. The *In* sums up the similarities for common beacons to inhibit the excitement in differences. The occurrence of a beacon at the same rank in both sets points to high similarity between them for that beacon. This event inhibits the difference with more weight such as $In = In + (i + j + (k - i))$. However, the weight of a missing element adds a constant 1 to both Ex and In, in order to avoid detection of '0' difference for completely different sets.

5.5 Experimental Environment and Beacon Data Collection

In order to ensure that the experimental environment reflects the settings available in common multi-floor buildings, Three important steps were taken: i) The used WiFi access points belong to a public network 'Nespot' deployed by Korea Telecom and the deployment and position of the beacons is not manipulated to better suit required SMA boundaries, and ii) the boundaries of SMA were arbitrarily defined independent of the access point deployments, and iii) Multiple data sets were collected on different day in order to evaluate the affect of the temporal variability in signal space which can cause biased location estimation performance. Extensive experiments have been conducted in a five floor campus building which contains multiple departments as well as associated labs and class rooms. Table 5.2 lists the SMAs defined in five floors of engineering building and 802.11 access points deployed in respective floors.

Actual devices used were hp iPAQ PDA model h4150 device running Pocket PC Embedded 2003 version of Microsoft Windows. The PDA has in built 802.11 network interface card. The WiFi signal collection system was implemented using C# in .Net to acquire data for training and test evaluations of the system. Since signal space can exhibit different properties at different times due to the indoor environmental factors. In order to evaluate the effect of time on localization capability, the training and test data collection spanned over one month at different days and times of the day. Besides time difference, the effect of the size of the training data was also evaluated by employing different sizes of the training set. Table 5.3 shows a listing of training data sets which were used for training the system as well as other systems given in [60], [45] and [38]. For the sake of demonstrating short development time, the training beacons were collected while a volunteer was walking on a normal speed in the SMAs. For the largest training set, Trg_4, the average time spent in each SMA was less than two minutes while the device was scanning the beacons every second. For the test data collection, total 15 hours were spent in all SMAs (on average 45 minutes per SMA) and collected 20,000 beacon signatures. All methods were first trained on each training set and tested on the extended test data set.

5.6 Experimental Results

The Election algorithm performance is evaluated in a real indoor environment. The signal coverage of WiFi beacons varies from place to place between as low as 2 to as high as 16 beacons. For the sake of clarity only one scheme is presented here where 4 Semantically Meaningful Areas were defined as the areas or interest.

5.6.1 The Affect of Missing Beacon Tolerance

Results show that $gDSCA$ formation depends on the missing beacon tolerance and $SMART$ size. In order to characterize the variance of performance, the $gDSCA$ formation results are presented with variation of these two parameters. Figure 5.3 bar graphs show number of $gDSCA$ changes with respect to $m\tau$ and τ. It is interesting to note that increasing the tolerance parameter results in lesser number of $gDSCA$ discoveries. From system development stand point zero tolerance can identify numerous $gDSCA$s in signal space but machine requires lot of human attention at the same time. Moreover, lesser $m\tau$ values increase the risk of redundant $SMART$s being created. The redundancy not only increases the memory and computational requirements, it adversely affect the recognition ability of the algorithm.

The recognition rate of Election algorithm is shown in figure 5.6.1 bar graphs for different values of $m\tau$ and τ. The SMA recognition error is measured as ratio of incorrect assignments to the total number of training or test vectors. The recognition error generally reduces for middle order options of $m\tau$ and τ. Nevertheless for the extreme values the error aggravates.

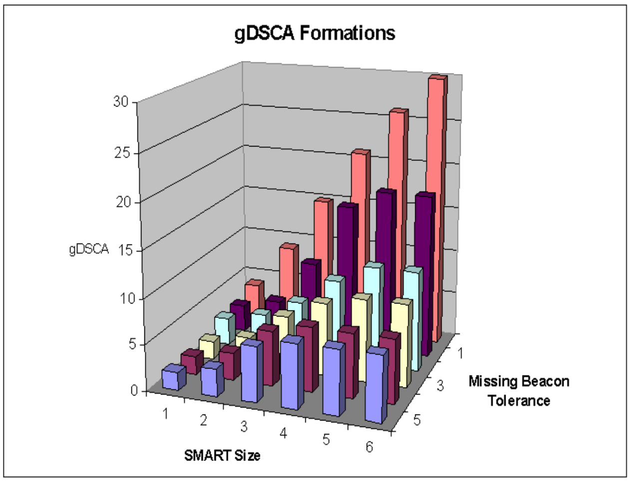


Figure 5.3: The $gDSCA$ Formations

5.6.2 Comparative Evaluations in a Multi-floor Environment

The localization performance is evaluated in two respects. Firstly, the SMA recognition is measured as: i) 'Correct SMA' is the ratio of correct estimates in total test signatures. ii) '1 SMA off' is the ratio of estimates which deviate from actual SMA not more than 1 neighboring SMA in total test signatures. Secondly, floor recognition-ability of each method is computed as ratio of 'Correct Floor', '1 Floor Off', '2 Floors Off' deviations of estimates from the actual floor.

NearMe

The NearMe system provides a list of neighboring devices by comparing the beacon signature of querying device with the signatures received from all other devices connected to the system. It proposes a general heuristic for computing the physical distance of two devices from their signatures. An elaborate description of this method can be found in [45].

In order to evaluate the SMA and Floor recognition ability of NearMe system following test setting was created. Suppose that j^{th} test signature $T_j^{S_i}$, collected in i^{th} SMA S_i, repre-

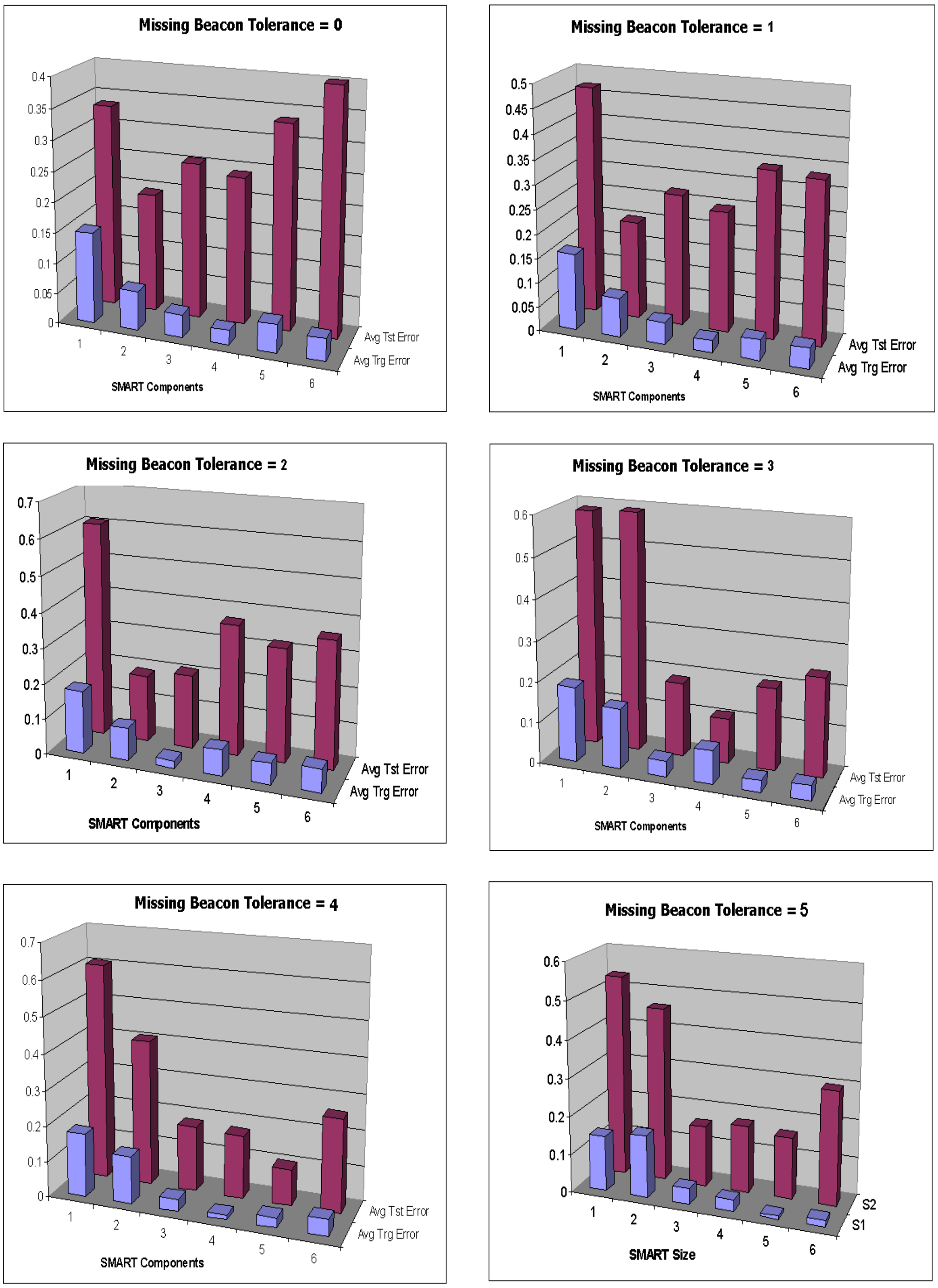


Figure 5.4: Affect of variations in Missing Beacon Tolerance $m\tau$ and $SMART$ size τ (x-axis) on SMA Recognition-rate (y-axis) of $SMART$. The z-axis show aggregated error for both training and test beacon vectors.

sents the device which wants to enquire for neighboring devices. Similarly, the signatures in the k^{th} training beacon set Trg$_k$ are collected from devices which are located all over the building including the S_i nearby $T_j^{S_i}$. Let n denote the number of training signatures which were collected from S_i. For finding the nearby devices, guided by the NearMe heuristic, the distance of each test signature $T_j^{S_i}$ with all the beacon signatures in Trg$_k$ is computed. Since the exact number of neighboring devices is known to be n. Therefore nearest n distances should be the devices which are within the proximity of query device. If the NearMe system qualifies m remote devices to the neighborhood of $T_j^{S_i}$ then the localization error e^{S_i} is computed as $e^{S_i} = \frac{1}{N}\sum_{j=1}^{N}\frac{m_j}{n}$ where N is the total number of test signatures. The SMA recognition and Floor recognition results of NearMe system are shown in Figures 5.8 and 5.9 respectively. The SMA recognition-ability is similar across different experiments. The Floor recognition results show that it can successfully localize within 2 floors. However the exact floor recognition rate remains lower than 60% on average.

SkyLoc

SkyLoc system employs GSM beacons to identify the floor of a mobile device in a multi-floor building [60]. It employs the k-nearest neighbors algorithm for floor identification. The collection of GSM signatures were not possible due to the unavailability of specific hardware and GSM infrastructure. Therefore, all experiments were conducted using only the 802.11 network beacons and the SkyLoc system was implemented to evaluate its performance in such environments. Detailed results for each floor are given in Figure 5.10 and overall localization performance is shown in Figures 5.5. The main reason behind performance degradation of SkyLoc is an essential incapability of Euclidean distance in signatures to represent the physical distance. Incidently, if majority of the beacons in t and v are different but a minority, or just one beacon, is similar then the distance can be minimal. The possibility that a non-representative beacons may exhibit similar signal strength at different floors makes this distance almost zero. Thus majority of physically distant locations fall in k nearest neighbors of signal space. This effect is referred to as overshadowing of non-representative beacon over representative beacons. Due to the same reasons the SMA recognition performance is degraded. Individual SMA recognition results are not presented here due to space constraints.

Rice

A topological location model based localization scheme is presented in [38], here referred to as ‘Rice‘. It divides the target environment into cells akin to the SMAs. Like typical radio map based location systems, it requires a priori information about the signal space to capture the signal strength variations at different cells of the target environment. A succinct description of their method can be found in [38]. This system is implemented for the target environment with same data preparation and modeling specifications. The only difference is the number of beacons used for training the system whereas ‘Rice‘ requires higher density network for claimed localization. However, they have shown that lowering the density of beacons shall reduce the localization accuracy. The implementation results confirm this as shown in Figures 5.11 and 5.12 respectively.

Beacognition

The performance of Beacognition is evaluated using the same data sets as used for the other systems. Even though it is a real-time discovery method, that is, it does not require a priori availability of training signatures, for the sake of comparisons the target device is fed with the same training and test data sets. In [2], an initial version of the election algorithm is presented, here referred to as 2-D *smart*, for single floor scenarios. It was demonstrated that the 2-D *smart* could achieve 87% accuracy in a two dimensional physical space. However in a multi-floor scenario its performance suffers from the resembling signal space problem as with the other systems. The enhanced version of Beacognition for multi-floor localization is referred to as 3-D *smart*. Two types of experiments were conducted to evaluate the performance of the 3-D *smart* with the 2-D *smart* as well. For both types, the values of τ and $m\tau$ parameters were chosen as 5 and 3 respectively. Figures 5.13 and 5.14 show performance of the 3-D *smart*.

Due to the enhanced election algorithm and similarity measurement model, the 3-D *smart* significantly outperforms other methods. The average correct floor recognition performance over all experiments is 89%. Whereas the SMA recognition is 83% for correct SMA and 86% for 1 SMA Off. Overall performance of all methods is summarized in Figure 5.5. The requirement of minimum SMA and Floor recognition rates largely depends on target applications. However, for coarse resolution based applications the expected base line 83% SMA recognition rate is very practical while the location information also contains the semantic meanings. In order to sustain sporadic inaccurate estimates, a general heuristic can be applied which utilizes: i) averaging the response of multiple location queries, and ii) the last known location and build a graph model to estimate next most probable location given the location estimate.

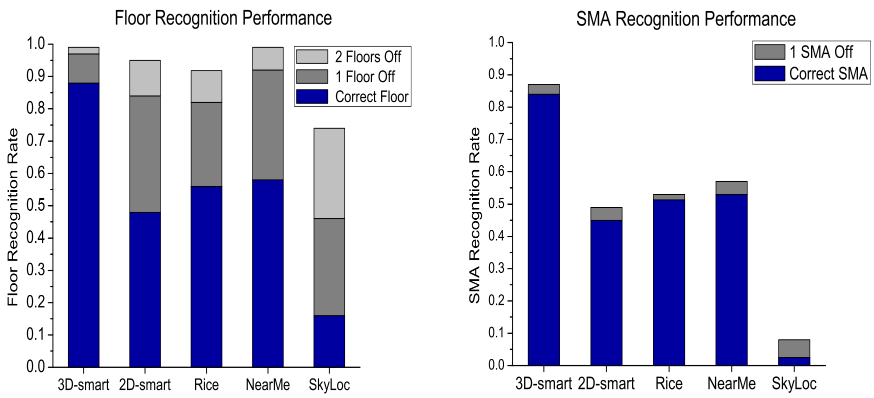


Figure 5.5: Overall Recognition Results: averaged over all training data sets

In order to evaluate the effect of training data size of localization performance a series of experiments were conducted with varying data amounts. For each experiment a portion of the training data sets Trg$_i$ was used one by one. Figure 5.6 shows the SMA recognition performance of 3-D *smart* with respect to different training sets and samples sizes. The 'Correct SMA Training' is the ratio of correct estimates against total training signatures. It reflects that how successfully Election algorithm can create SMAs from training data. The Floor recognition results are shown in Figure 5.7. It is noted that both SMA and Floor recognition performance improves upon increasing the training samples. However, as can be observed from Trg4 experiments, increasing the training signatures beyond 50 does not have significant effect on the performance. It suggests that a device needs to roam in an SMA for less than one minute while scanning beacons every second.

Another straightforward merit of the proposed methodology is that it requires minimal storage for storing its knowledge about location to signal space mapping. Once the winner beacons are discovered, all it needs to store for an SMA S is those best representative beacons along with their popularity information. In the experiments the size of the file containing all *smart*s remains less than 9 kB. Contrarily, the other systems require all training signatures collected from S to be stored. Later at the recognition time the computational requirements of the proposed method are also significantly lower than other systems. Each test signature, location query, is compared only with the *smart*s whereas other methods compare it with all training signatures.

Even though the experiments were conducted using only IEEE 802.11 WLAN access points, the underlying radio infrastructure does not matter as long as the joint coverage area of the beacons is sufficient enough for the target applications. Beacognition can be applied to other commonly available radio signals (e.g RFID, Bluetooth, GSM, CDMA) for indoor localization very easily. However, the short range beacons such as RFID shall require denser deployments on large scale.

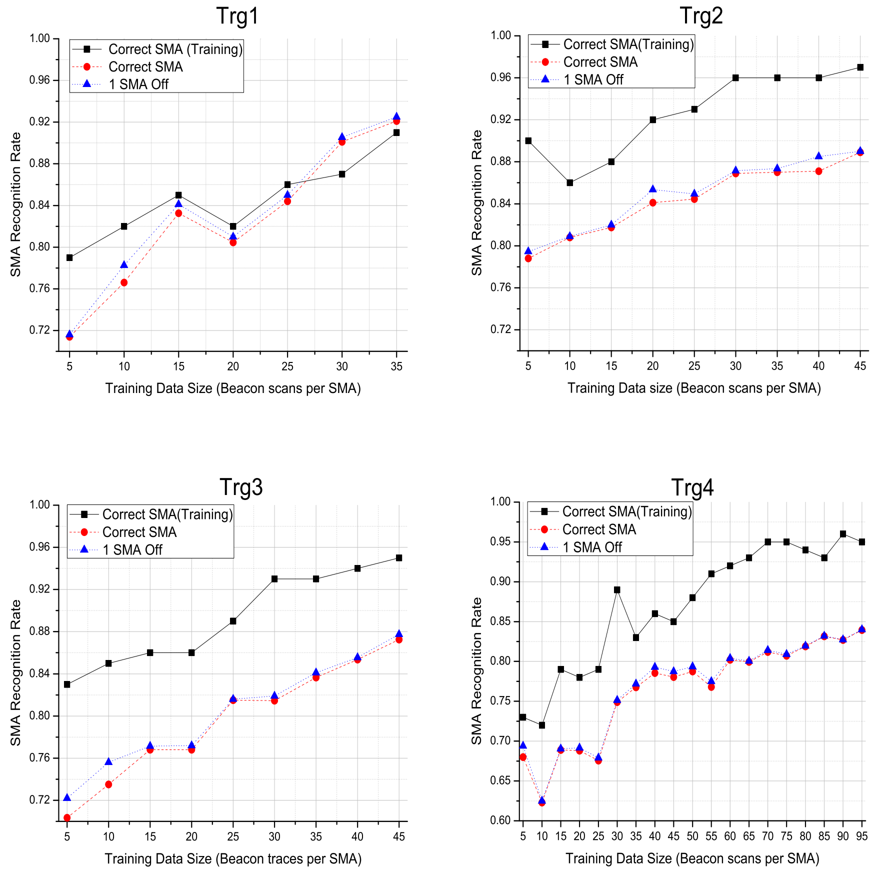


Figure 5.6: Different training data size. The legend of Trg1 applies to all graphs.

5.7 Summary

This chapter offers a new methodology for indoor semantically meaningful localization using IEEE 802.11 network beacons. It aims at two objectives: i) short development time by providing an real-time discovery algorithm which requires no prior knowledge about signal and physical space, and ii) it involves end users in the system development by providing an interactive development scheme. Both objectives serve the purpose of rapid development of indoor location systems in a multi-floor environment. The comparative results show that

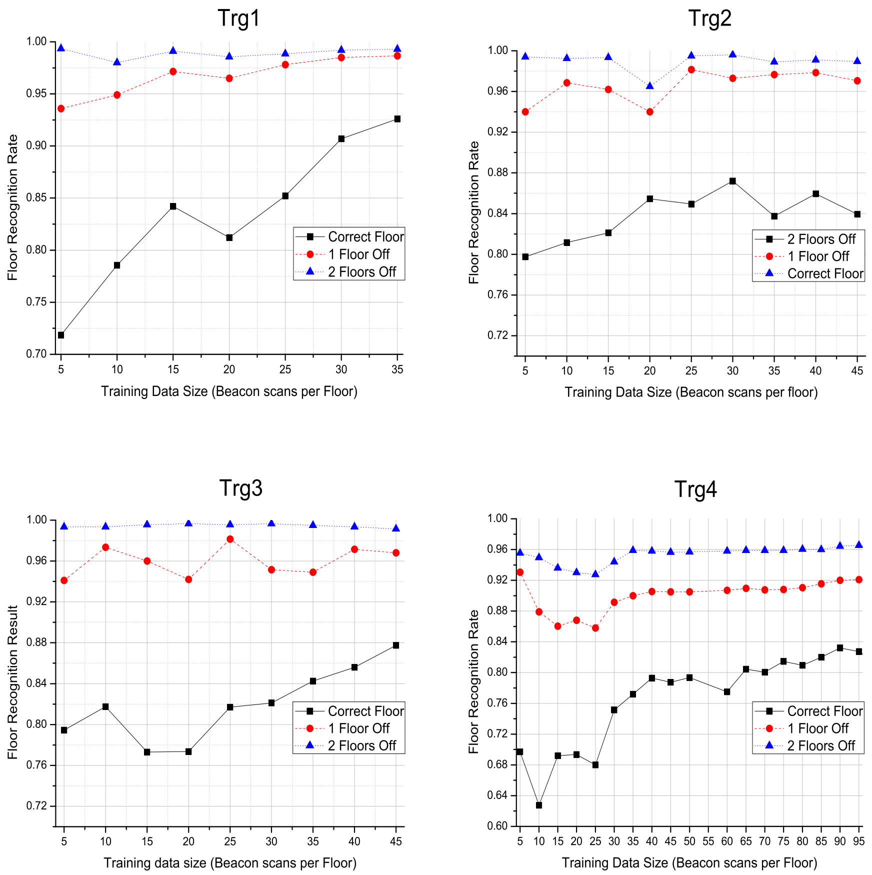


Figure 5.7: Different training data size. The legend of Trg1 applies to all graphs.

Beacognition achieves significantly better performance, both in terms of SMA recognition as well as Floor recognition. The *smart*s require minimal storage which makes it suitable for stand alone, privacy observant location systems. In this case a device can localize itself by passively scanning the environment without even being connected to the network. However, a centralized location system is feasible as well by placing all *smart*s on a location server while mobile devices place their location queries in terms of detected beacon signatures. Despite significantly better performance of Beacognition, it inherits a limitation of the beacon-based localization systems which is the requirement of fairly dense deployment of radio beacons. In case of a sparse deployment the boundaries of the target SMAs may not exactly correspond to the needs of target applications.

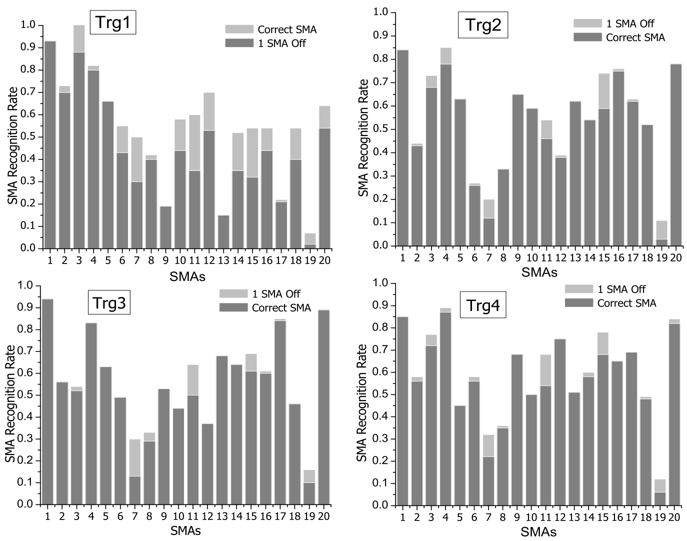


Figure 5.8: NearMe SMA Recognition Results

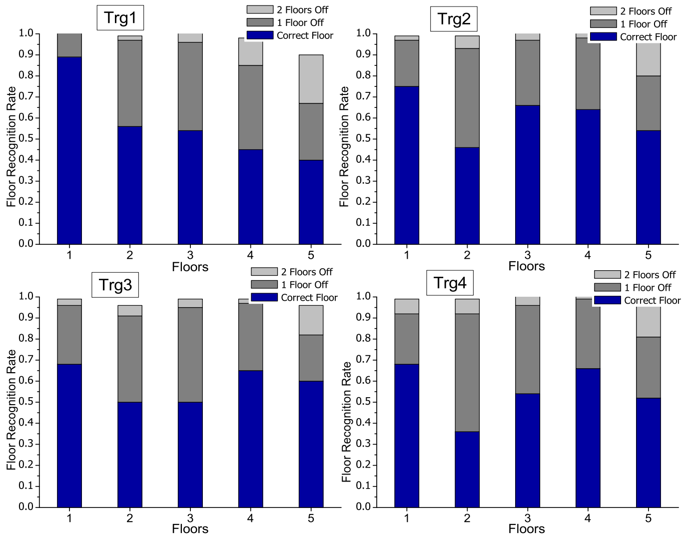


Figure 5.9: NearMe Floor Recognition Results

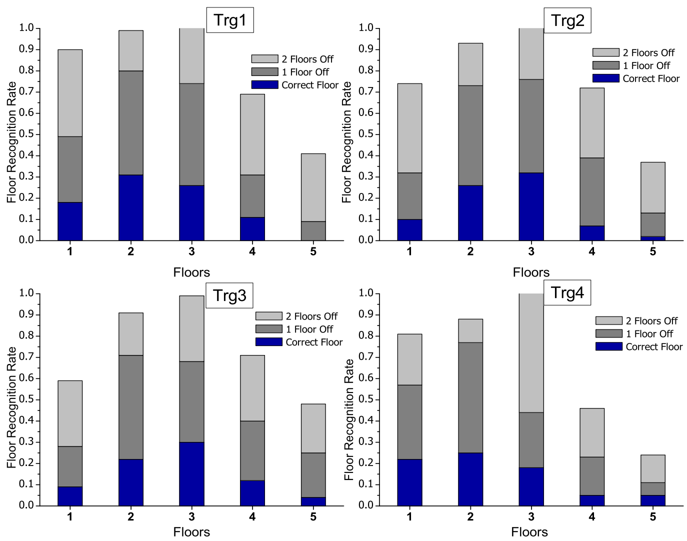


Figure 5.10: SkyLoc Floor Recognition Results

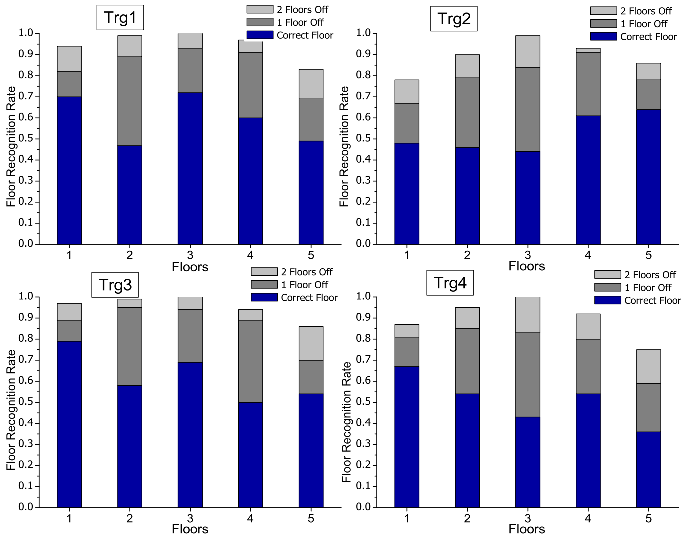


Figure 5.11: Floor Recognition Results of Rice

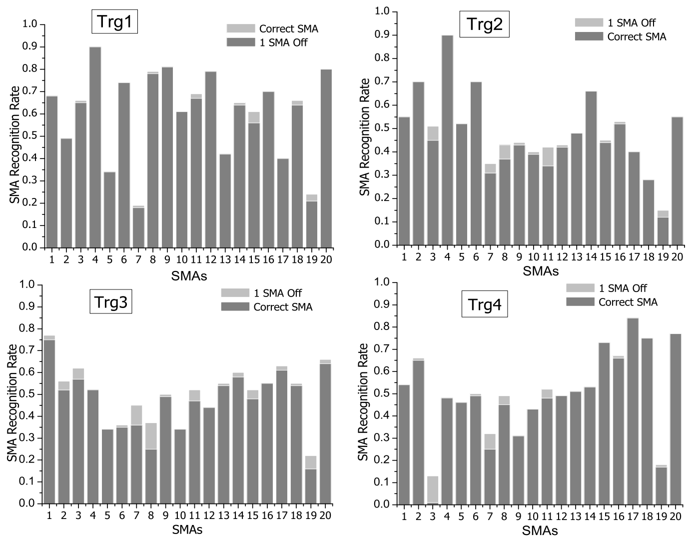


Figure 5.12: SMA Recognition Results of Rice

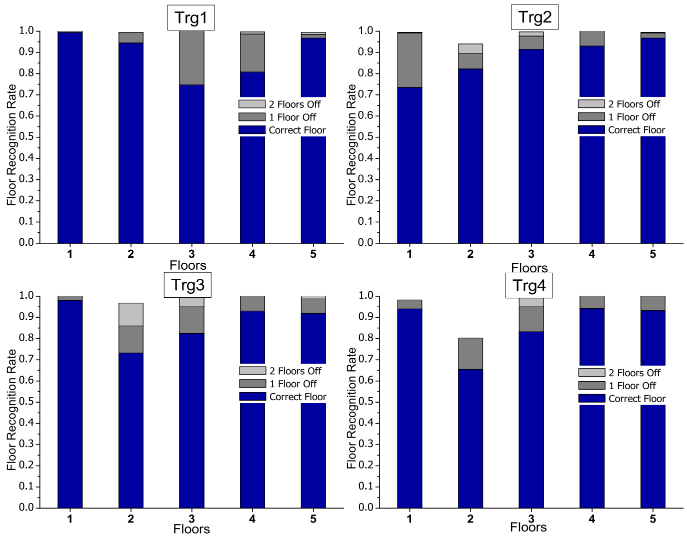


Figure 5.13: 3-D smart Floor Recognition Results

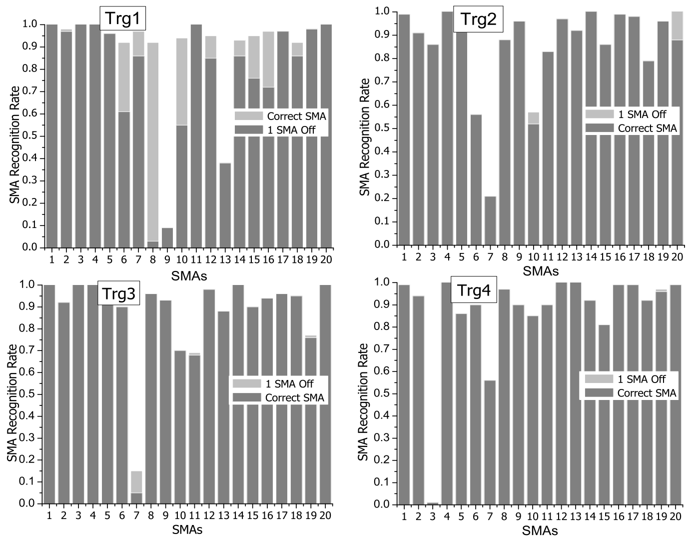


Figure 5.14: 3-D smart SMA Recognition Results

Table 5.2: Experimental Physical Space and 802.11 Beacons.

Floor	SMAs	Beacon MAC(Last four Digits)
1	Comp Physics Labs	8135
1	Photo-Electronics Labs	5139
1	Institute of Natural Sci	5035
1	Natural Sci Lec Rooms	5883
2	Robotics Labs	9235
2	Biomed Lecture Rooms	7199, 9207
2	App Biomedical Engg	
2	Admin Offices	8203
3	Comp Engg Labs	7195
3	Radio Engg Labs	9239
3	Imp/Img Res Labs	2243
3	Faculty Offices	5823
4	Stdt Unions Offices	
4	Radio Engg Rooms	5551
4	Bio-Medical Labs	
4	Lecture Rooms	5535, 5543
5	Laser Engineering Labs	5659
5	Communication Labs	6079
5	Astrophysics Labs	
5	Micro/Ultrasonic Labs	5559

Table 5.3: Training beacon data specification

Data Set	Scans/sec per SMA	File Size(kB)
Trg_1	40	51
Trg_2	50	62
Trg_3	50	67
Trg_4	100	201

Chapter 6

Conclusions and Future Work

Received Signal Strength based location systems are poised to enable indoor positioning systems due to their economic viability. Intrinsically, signal strength based location estimation is a pattern classification problem. Large scale realization of these systems face with the visibility issue which introduces null values in the radio map feature space resulting in sparsity and redundancy. A modular classification model is presented to overcome the visibility problem by incorporating the prior knowledge about signal visibility into design of the classification system. This is achieved by partitioning high-dimensional and sparse radio map feature space into low-dimensional but compact subspaces. Two location systems were developed for different sites in order to confirm the applicability and robustness of modular approach in real life environments. It is demonstrated through development of location systems in two real life environments that signal visibility based decomposition of radio map enables development of location systems in arbitrarily large target sites.

Although the modular classification approach provides high accuracy and shorter training time still it requires both creation of radio map as well as visibility matrix of signal sources. ConSelFAM classifier design is inspired by modular classification but the only prior information it requires for learning is visibility matrix. It incorporate this prior information about visibility as classification *context*. A rapid development of classification systems can be realized driven by *online*, *incremental*,*self-scaling* and *context-aware* learning properties of ConSelFAM. *Extensibility* and *expansibility* are realized by introducing *Self-scaling* subsystem in Fuzzy Art and *Context Field* subsystem in Fuzzy ArtMap.

The Beacognition achieves significantly better performance, both in terms of SMA recognition as well as Floor recognition. The *smart*s require minimal storage which makes it suitable for stand alone, privacy observant location systems. In this case a device can localize itself by passively scanning the environment without even being connected to the network. However, a centralized location system is feasible as well by placing all *smart*s on a location server while mobile devices place their location queries in terms of detected beacon signatures.

There are four major future works that can be pursued. Firstly, a baseline location determination system can be enhanced to provide object tracking and activity recognition for smart environments. This would require investigation of existing and development of new filtering techniques and efficient methods to create location models. Secondly, the accuracy and reliability of location estimates can be enhanced by incorporating hybrid signal spaces such as WiFi, Blue tooth and GSM/CDMA cell towers. Thirdly, incorporation of the Ababil middleware for pursuing the application oriented research. A very demanding area is to develop novel location based applications for assisting business management processes and daily living tasks to the elderly people. Fourthly, since the hosting devices of the indoor location based applications are resource constrained, efficient methods for rendering guiding maps and directions is an important future work. An XML based active space markup language is proposed in [10] which can be enhanced to achieve this goal.

Bibliography

[1] Uzair Ahmad. Componet: Componentization of neural networks. `http://www.sourceforge.net/projects/componet/`, 2006.

[2] Uzair Ahmad, Brian J. d'Auriol, Young-Koo Lee, and Sungyoung Lee. The election algorithm for semantically meaningful location-awareness. In *MUM '07: Proceedings of the 6th IEEE/ACM international conference on Mobile and ubiquitous multimedia*, pages 55–63, Oulu, Finland, December 2007. ACM.

[3] Uzair Ahmad, A. Gavrilov, Sungyoung Lee, and Young-Koo Lee. Modular multilayer perceptron for wlan based localization. In *19th IEEE International Joint Conference on Neural Networks*, pages 3465–3471, Vancouver, Canada, July 2006. IEEE.

[4] Uzair Ahmad, A. Gavrilov, U. Nasir, M. Iqbal, Seong Jin Cho, and Sungyoung Lee. In-building localization using neural networks. In *IEEE International Conference on Engineering of Intelligent Systems*, pages 1–6, Islamabad, Pakistan, March 2006. IEEE.

[5] Uzair Ahmad, Andrey Gavrilov, Sungyoung Lee, and Young-Koo Lee. Componet: Programmatically embedding neural networks into ai applications as software components. In *19th IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 194–201, Patras, Greec, October 2007.

[6] Uzair Ahmad, Andrey Gavrilov, Young-Koo Lee, and Sungyoung Lee. Context-aware, self-scaling fuzzy artmap for received signal strength based location systems. *Journal of Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 12(7):699–713, 2008.

[7] Uzair Ahmad, Sungyoung Lee, Mahrin Iqbal, Uzma Nasir, A. Ali, and Mudeem Iqbal. Reflective middleware for location-aware application adaptation. In *Computational Science and Its Applications*, pages 1045–1054. Springer, Lecture Notes on Computer Science, 2005.

[8] Uzair Ahmad, Young-Koo Lee, Sungyoug Lee, and Chongkug Park. Indoor location estimation using radio beacons. In *ICMIT'07: International Conference on Mechatronics and Information Technologies*, volume 6794, Gifu, Japan, January 2007. The International Society for Optical Engineering.

[9] Uzair Ahmad, Young-Koo Lee, and Sungyoung Lee. A distributed and parallel sampling system for efficient development of radio map. In *Proceedings of the 2007 International Conference on Information & Knowledge Engineering (IKE2007)*, pages 272–279, Las Vegas, Nevada, USA, June 2007. CSREA Press.

[10] Uzair Ahmad, Uzma Nasir, Mahrin Iqbal, Young-Koo Lee, Sungyoung Lee, and In-ook Hwang. On building a reflective middleware service for location-awareness. In *RTCSA '05: Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 439–442, Washington, DC, USA, 2005. IEEE.

[11] Moore. B. Art and pattern clustering. In *Connectionist models summer school*, volume 4, pages 174–183, 1988.

[12] Roberto Battiti, Alessandro Villani, and Thang Le Nhat. Neural Network Model for intelligent networks: deriving the Location from signal patterns. In *The First Annual Symposium on Autonomous Intelligent Networks and Systems*, 2002.

[13] Ladd A.M. Bekris, K.E. Marceau, G. Rudys, A. Wallach, and D.S. Kavraki. Using wireless Ethernet for localization. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 1, pages 402–408, 2002.

[14] J. Bohn. Prototypical implementation of location-aware services based on a middleware architecture for super-distributed rfid tag infrastructures. *Personal Ubiquitous Comput.*, 12(2):155–166, 2008.

[15] G. A. Carpenter and S. Grossberg. Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3:698–713, 1992.

[16] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[17] G.A. Carpenter and S. Grossberg. *Handbook of Brain Theory and Neural Networks*. MIT Press- Cambridge, 2003.

[18] Mike Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, and Jeffrey Hightower et al. Practical metropolitan-scale positioning for gsm phones. In *Ubicomp*, pages 17–21, Orange County, California, USA, September 2006. IEEE.

[19] Y. Chen, X. Y. Chen, F. Y. Rao, X. L. Yu, Y. Li, and D. Liu. Lore: an infrastructure to support location-aware services. *IBM J. Res. Dev.*, 48(5/6):601–615, 2004.

[20] Ying CHEN, Wei LU, Xiaoyan CHEN, Lihua TANG, Fangyan RAO, Qingbo WANG, and Long ZHANG. Location aware messaging - integrating lbs middleware and converged services. In *ICEBE '05: Proceedings of the IEEE International Conference on e-Business Engineering*, pages 419–426, Washington, DC, USA, 2005. IEEE Computer Society.

[21] Gianpaolo Cugola and H.-Arno Jacobsen. Using publish/subscribe middleware for mobile systems. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(4):25–33, 2002.

[22] L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302, 1945.

[23] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection for gene expression data. In *IEEE Bioinformatics Conference*, pages 523–529, August 2003.

[24] Richard O. Duda, Perter E. Hart, and David G. Stok. *Pattern Recognition, Second Edition*. WILEY INTERSCIENCE, 2000.

[25] Ekahau Positioning Engine. Ekahau inc., 2008.

[26] Anthony LaMarca et al. Place lab: Device positioning using radio beacons in the wild. In *Proceedings of Pervasive*, pages 116–133, Munich, Germany, 2005.

[27] B. Li et al. Method for yielding a database of location fingerprints in wlan. *IEE Proceedings - Communications*, 152(5):580–586, October 2005.

[28] D. Bottazzi et al. Agape: a location-aware group membership middleware for pervasive computing environments. In *ISCC '03: Eighth IEEE International Symposium on Computers and Communication*, pages 1185–1192. IEEE, July 2003.

[29] John et al. Minimizing calibration effort for an indoor 802.11 device location measurement system. Technical report, Microsoft Research, MSR-TR-2003-82, November 2003.

[30] P. Bahl et al. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM 2000*, pages 775–784, March 2000.

[31] P. Vlach et al. Cross-floor signal propagation inside a contemporary ferro-concrete building at 434, 862, and 1705 mhz. *IEEE Transactions on Antennas and Propagation*, 47(7):1230–1232, July 1999.

[32] Paul Castro et al. A probabilistic room location service for wireless networked environments. In *Ubicomp*, pages 18–34. IEEE, 2001.

[33] Teemu Roos et al. A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, 1(1):59–70, January 2002.

[34] Dieter Fox, Jeffrey Hightower, Lin Liao, and Dirk Schulz. Bayesian filters for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, September 2003.

[35] T. Fuchs, A. Haney, T.J. Jechura, F.R. Moore, and V.P. Bingman. Daytime naps in night-migrating birds: behavioural adaptation to seasonal sleep deprivation in the swainson's thrush, catharus ustulatus. *Animal Behaviour*, 72(4):951–958, October 1959.

[36] Donald R. Griffin. The sensory basis of bird navigation. *The Quarterly Review of Biology*, 19(1):15–31, March 1944.

[37] I. Guvenc, C. T. Abdallah, R. Jordan, and O. Dedeoglu. Enhancements to rss based indoor tracking systems using kalman filters. In *GSPx International Signal Processing Conference*, April 2003.

[38] 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 '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2004.

[39] M. T. Hagan and M. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.

[40] Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian Smith, and Jeff Hughes. Learning and recognizing the places we go. In *Ubicomp*, pages 159–176. IEEE, 2005.

[41] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[42] A.K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. *Handbook of Statistics*, 2:835–855, 1987.

[43] T. Kohonen. *Learning vector quantization*. MIT Press, 1995.

[44] John Krumm, Gerry Cermak, and Eric Horvitz. Rightspot: A novel sense of location for a smart personal object. In *Ubicomp*, volume 2864, pages 36–43. LNCS, Springer, 2003.

[45] John Krumm and Ken Hinckley. The nearme wireless proximity server. In *The Sixth International Conference on Ubiquitous Computing*, pages 283–300. IEEE, September 2004.

[46] Teddy Mantoro and C. W. Johnson. nk-nearest neighbor algorithm for estimation of symbolic user location in pervasive computing environments. In *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, pages 472–474, Washington, DC, USA, 2005. IEEE Computer Society.

[47] MATLAB. Matlab nerual network toolbox. `http://www.mathworks.com/products/neuralnet/`.

[48] Martin Fodslette Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

[49] Amy L. Murphy, Gian Pietro Picco, and Gruia-Catalin Roman. Lime: A coordination model and middleware supporting mobility of hosts and agents. *ACM Trans. Softw. Eng. Methodol.*, 15(3):279–328, 2006.

[50] Tomoaki Ogawa, Shuichi Yoshino, and Masashi Shimizu. Location determination method for wireless systems based on learning vector quantization. *NTT Network Innovation Laboratories*, 1(9):27–36, 2003.

[51] J.P. Pahlavan, K. Xinrong Li Makela. Indoor Geolocation Science and Technology. In *IEEE Communications Magazine*, volume 40, pages 112–118, February 2002.

[52] Steve Pope. Issues related to rssi measurement.

[53] Nissanka Bodhi Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *6th ACM MOBICOM*, Boston, MA, August 2000.

[54] Anand Ranganathan, Jalal Al-Muhtadi, Shiva Chetan, Roy Campbell, and M. Dennis Mickunas. Middlewhere: a middleware for location awareness in ubiquitous computing applications. In *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 397–416, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[55] M. Sarkar. Modular pattern classifiers: a brief survey. *IEEE International Conference on Systems, Man, and Cybernetics*, 4:2878–2883, Aug 2000.

[56] M. Teresa Serrano-Gotarredona, Bernabe Linares-Berranco, and G. Andreaou Andreas. *Adaptive resonance theory microchips-Circuit Design Techniques*. Springer, 2005.

[57] D. Smailagic, A. Kogan. Location Sensing and Privacy in a Context Aware Computing Environment. In *Pervasive Computing*, volume 9, pages 10–17, October 2002.

[58] So Young Sohn. Meta analysis of classification algorithms for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1137–1144, 1999.

[59] UbiSense. Local position system and sentient computing.

[60] Alex Varshavsky, Eyal de Lara, Jeffrey Hightower, Anthony LaMarca, and Veljo Otsason. Gsm indoor localization. *Pervasive Mobile Computing*, 3(6):698–720, 2007.

[61] Michael Wallbaum. A priori error estimates for wireless local area network positioning systems. *Pervasive Mob. Comput.*, 3(5):560–580, 2007.

[62] Roy Want, Andy Hopper, ao Veronica Falc and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992.

[63] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, Oct 1997.

[64] M. Youssef and A. Agrawala. On the optimality of wlan location determination systems. In *In Communication Networks and Distributed Systems Modeling and Simulation Conference*, January 2004.

[65] Moustafa A. Youssef, Ashok Agrawala, and A. Udaya Shankar. Wlan location determination via clustering and probability distributions. In *Percom*, pages 23–26. IEEE, March 2003.