UCGE Reports
Number 20351

Department of Geomatics Engineering

Indoor Location Based Services

(URL: http://www.geomatics.ucalgary.ca/graduatetheses)

by

Ehsan Mohammadi

Date

September, 2011

UNIVERSITY OF CALGARY

Indoor Location Based Services

by

Ehsan Mohammadi

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

September, 2011

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Indoor Location Based Services " submitted by Ehsan Mohammadi in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE.

Dr. Andrew J.S. Hunter
Supervisor, Department of Geomatics Engineering

Dr. Steve H.L. Liang
Department of Geomatics Engineering

Dr. Kyle O'Keefe
Department of Geomatics Engineering

Dr. Lina Kattan
Department of Civil Engineering

Date

Abstract

With the advent of mobile devices and the Internet, location based services (LBS) has become an important part of our daily lives. Location based services integrate location of a mobile device with other information to provide different types of services to a user. Thus, knowledge of the location and position of a user is an essential part of a location based service.

In this research indoor positioning and multiple choices of best paths to a destination is investigated as two main use cases of LBS applications. Compared with outdoors, indoor environments are highly structured and more complicated. Thus, describing indoor environments for LBS applications requires models that can support complexity. Global positioning systems have difficulties fixing positions inside buildings; so other techniques and methods are required for indoor positioning.

This thesis presents the research and implementation of location fingerprinting methods for indoor positioning, along with the results of testing each method. Analysis suggest that KNN (K=2) analysis provides the most precise results. In addition, in this research, the all possible paths as a pathfinding solution was implemented and compared with K shortest paths solution. The all possible paths solution is more time efficient than K shortest paths and can also provide a base for other network analysis.

Dedication

To the memory of my mother, **Ozra** and my father, **Ata**

No son ever had better parents.

Acknowledgements

I would like to thank my supervisor Dr. Andrew J.S. Hunter for giving me the opportunity to work on this research. His help and guidance throughout this research is much appreciated. I also would like to thank my family for all their supports through my life.

Ehsan Mohammadi

Calgary, Canada.

September, 2011.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

With the advent of mobile devices and the Internet, location based services (LBS) has become an important part of our daily lives [1]. Location based services integrate location of a mobile device with other information to provide different types of services to a user [2]. Thus, knowledge of the location and position of a user is an essential part of a location based service. Different infrastructure components are necessary for a location based service to work. These components are: mobile devices, communication networks, positioning systems, and data/service providers [3].

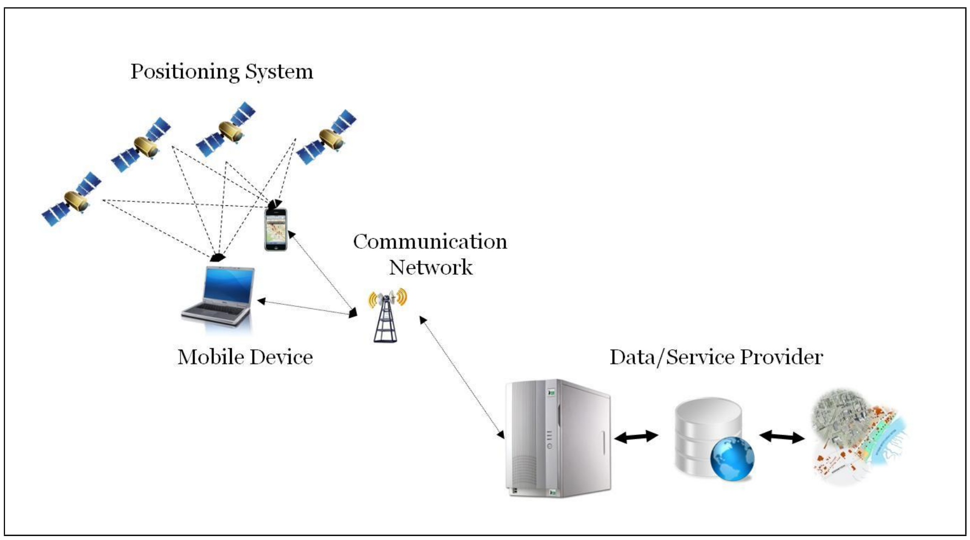


Figure 1.1: LBS Infrastructure Components

LBS has many different applications. Recommending social events, requesting the nearest business, turn by turn navigation, locating people on a map [4], sending messages to certain geographic areas (i.e. geocasting) [5] are some examples of LBS applications.

A simple scenario for using LBS is guiding a person to the closest emergency center. For this purpose position of the person should be revealed to the system (positioning component). This can be done on a mobile device (e.g., laptop, PDA, smartphone). In order to send information to and receive information from the service provider, a communication network is essential. The communication network provides connection between a mobile device and the service provider. The service provider receives the position of a mobile device and analyses the spatial (e.g., routes, traffics) and non-spatial data (e.g., time, date) to find the most appropriate route to the closest emergency center, and sends the result to the user through the communication network.

Location based services are different from other paper or internet based services in that they are aware of the context in which they function [3]. "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves" [6]. Context can be any source of useful information for the intended application. In a location based service any piece of context information can fall in one of the following categories: user's identity, location/position, time, orientation of movement, navigation history, purpose of use, social and cultural situation, physical surroundings and system properties [3]. According to this definition of context, LBS can be considered a context-aware system in which position of the user and real world surroundings compose the main part of the context information, so more relevant messaging and services can be sent to a user if information received from the user is adjusted to their context [2].

As mentioned, in LBS most context information depends on the knowledge of a user's position or can be extracted from the history of a user's position. Thus, obtaining the position of a user accurately is necessary to provide location based services [7]. A positioning system ideally offers the coordinates of an object in a spatial reference system and keeps the track of the object as it moves through the space [8]. Various technologies have been proposed to determine the location of users in various environments. However, these technologies can be separated into two main categories, depending on whether they are operating in outdoor or indoor environments.

Outdoor positioning technologies can be divided into two further categories: Satellite-based positioning and Network-based positioning [9]. GPS, GLONASS, Galileo and Beidou are examples of satellite-based positioning systems, while Cell of Origion (COO), Time of Arrival (TOA), Angle of Arrival (AOA) and Time Difference of Arrival (TDOA) are examples of techniques used in network-based positioning. The cell of origin (COO) method uses the network Base Transceiver Station (BTS) to identify the user in the cell area. The angle of arrival (AOA) method is based on triangulation and determines the position of a mobile device using intersection of directional lines between the mobile device and at least two base stations. The time of arrival (TOA) method determines the position of a mobile device using intersection of lines between the mobile device and at least two base stations. Distances between the mobile device and stations are obtained using time of wave propagation. The time difference of arrival (TDOA) is based on trilateration and uses the time difference measured from two stations to define a hyperbolic curve as an estimation of the position of the mobile device. Using an extra station provides a new hyperbolic and the intersection of two hyperbolic curves give the position of the mobile device [10].

However indoor positioning systems (IPS) are intended to provide location estimation for wireless devices such as laptops, PDAs and smartphones inside buildings and closed environments such as stores, hospitals, warehouses, campuses, hotels, airports and factories [11]. However, no single positioning system supports both indoor and outdoor environments to an acceptable quality [12]. GPS is currently the defacto standard for positioning in outdoor environments [13]; but for indoor positioning, there is no globally accepted standard.

Indoor positioning systems are mainly based on the following technologies: Active Radio Frequency Identification (RFID), Infrared (IR), Ultra-Wideband (UWB) and WiFi [8]. An Active RFID infrastructure requires scanners -with known position- to be installed through a facility that interrogates tags attached to moving objects. The position of the tag is identified as the position of the scanner, which detects the tag [14]. RFID technology uses proximity analysis to detect tags approaching a scanner [14]. Similar to RFID, an IR system determines the presence of an object. Objects are equipped with Infrared transmitters. Receivers installed through a building detect transmitted signals and determine approximate position of objects [8]. UWB systems are similar to the previous technologies, but operate using radio signals with a very wide bandwidth. The positioning methods used by this technology are generally based on time-of-arrival techniques.

LBS deals with other issues beside positioning such as various applications, context awareness, mobile device abilities, privacy, wireless mobile networks, spatial and spatio-temporal analysis, database management systems, and spatial indexing [3]; while this research deals only with positioning and path finding as a type of spatial analysis.

1.2 Problem Statement

In this research indoor positioning and multiple choices of best paths to a destination is investigated as two main use cases of LBS applications. The first issue, indoor positioning, refers to the "Where am I?" problem; and the second one, path finding, refers to the "How can I reach my destination?" problem.

Although the accuracy of standard GPS is ample for many applications, using GPS in certain areas is not possible [15]. Since a clear line of sight is required between a GPS receiver and satellites, receivers have difficulty fixing positions inside buildings or between tall obstacles (i.e. urban canyon effect) [15]. Network-based (Cellular) positioning has some advantage over GPS. They produce stronger signals that operate indoors and are unaffected by urban canyons. However, cells in network-based systems vary in size depending on terrain and the number of users [15]. Big cells result in inaccurate positions which make it inappropriate for applications that need accurate positioning results as accurate user location information enables a wide range of location dependent applications [13].

Indoor positioning is the main interest of this research versus outdoor positioning. In the previous section different indoor positioning methods were introduced, however, using existing infrastructure is preferred for positioning rather than deploying additional infrastructure [7]. As such, this research focuses on using WiFi technology that is already widely deployed in many facilities for indoor positioning.

Indoor positioning (geolocation) based on techniques such as Time Of Arrival (TOA), Time Difference Of Arrival (TDOA) and Angle of Arrival (AOA) is only reliable when signal line of sight is dominant, which limits these techniques in indoor environments [16]. So approaches based on signal strength are more frequently explored for indoor areas [16].

Location fingerprinting is the main technique of indoor positioning investigated in this research. When positioning using location, fingerprinting signal strength of unknown points are compared with a collection of signal strengths from known points (calibration points) and positions are inferred using a comparison algorithm. Other techniques like triangulation can also be utilized, but triangulation and direction often yield highly erroneous results in indoor environments [7].

Brunato et al. [17] classifies positioning systems into two main categories: first those assisted by dedicated hardware and second those that can operate with no need for additional hardware. Indoor location fingerprinting belongs to the second category, as there is no need for additional hardware installments, which reduces total system cost. Along with its advantages, location fingerprinting has some drawbacks. The main drawback of location fingerprinting is that it becomes costly as the area to be covered and number of users becomes large [7]. Larger areas imply a larger number of calibration points. However, in indoor areas finding the position with the accuracy of a room is sufficient. Gathering extensive training data for location fingerprinting can be very costly and requires substantial deployment effort. Therefore, minimizing the number of training observations is preferable and demands a more adaptable model [11]. The goal of this research is to identify an indoor positioning technique with sufficient accuracy to locate someone in a room with a reduced number of calibration points [18]. Taking into account a large number of calibration points provides us with better positioning accuracy, but it is also more costly and sometimes redundant. Given these conditions a problem statement for this research is:

Given the structure of indoor environments, which location model is best for positioning purposes, so we can infer the location of the user, instead of determining the coordinate of the position?

Based on this problem statement instead of calculating the position, our IPS should be able to infer the location. This use case can be further refined to:

Given the limitations of indoor location fingerprinting, how can we use signal information in its full capacity to assist the positioning system to function properly with a small number of calibration points.

Path finding in indoor environments is another concern of this research. Destination choice and path selection are two critical characteristics of human path finding [19]. Different users have their own preferences and choices for path finding. Not only do we need to select and follow paths that are compatible with our preferences, we need to develop models capable of finding solutions to these path selection problems.

Paths in indoor environments are not as explicit as paths are in outdoor environments. Indoor path finding difficulties are caused by lack of signs and information, unfamiliarity with the interior design of a building and problems with finding main entrances [20]. Thus our location model should also support path finding for indoor areas. Considering this, we can posit the following additional problem statement:

Given the structure of indoor environments, our spatial model should also support path finding requirements.

Based on the spatial model for indoor areas, we can provide users with the best paths to their destination. But these paths are not unique and can differ depending on users' situation. So our next problem statement is:

Due to different situations and preferences of users of indoor location based services, we must develop our system so it can consider various situations and preferences to suggest best paths to a destination. We may have more than one best path to a specific location, so the system should be able to apply changes dynamically for this purpose.

1.3 Research Objectives

So far, different research fields are introduced in this chapter. In order to make the research feasible and reasonable, the scope of the research has been narrowed to these items:

1. Development of an indoor positioning approach, using existing wireless infrastructure. This research attempts to find the best match for a user's position from a predefined set of points or locations. Location fingerprinting is the principle method for positioning, however, instead of calculating the position; we

attempt to match the signal patterns at the location of user with signal patterns in a calibration dataset. So the emphasize is on reducing the number of calibration points and consequently reducing the cost of system deployment.

2. Extracting most suitable paths for users and suggesting most likely paths, based on parameters such as user's preferences. This part of the research mainly attempts to find and suggest more than one path to a destination. The results of multiple-paths can satisfy various requirements of a user.

1.4 Significance of Research

This research tries to demonstrate the significance of using signal distribution patterns as a method for location detection or positioning. In other words, this research tries to use both signal strength and its variability for positioning purposes. If variability is not considered, most methods are not capable of determining position with few calibration points. This research also hopes to illustrate the importance of finding all possible paths to a destination in contrast to just the shortest path. Defining all possible paths becomes more significant when dealing with routes that undergo dynamic changes. The application of all possible paths is not restricted to indoor location based services. Most applications that deal with network analysis can benefit from the results of all possible paths.

1.5 Organization of Thesis

This chapter provided an overview of two main indoor LBS use cases: Indoor positioning and indoor path finding. The specific research problems related to these two have also been described and the objectives and research goals are identified. Chapter two opens a detailed discussion of location models for indoor environments and examines different aspects of those models. Chapter three presents comprehensive details of indoor positioning and indoor path finding. Different techniques for location fingerprinting including Artificial Neural Network (ANN), Support Vector Machine (SVM) and K Nearest Neighbour (KNN) are discussed, and solutions for path finding are presented. Chapter four details the study area, data and methods used for this study. This chapter provides statistical distances for the location fingerprinting methods and examines different algorithms for path finding. Chapter five presents the results of both positioning and path finding methods. The final conclusions and suggestions for future works are provided in chapter six.

Chapter 2

Basic Indoor Location Structure and Model

As mentioned in the previous chapter, positioning in indoor environments aims at defining the location of a user within an indoor area. In other words we try to find the position at room granularity, rather than a pair of coordinates at meter or centimeter accuracy. Besides positioning, for the path finding problem we need to implicitly define a network mesh of walkable paths in indoor environments, as opposed to a one dimensional road network. In order to address these issues we require an understanding of different location models and their application for indoor areas. This chapter looks at various location models in detail for both positioning and path finding applications.

2.1 Indoor Environment Structure

People spend most of their lives in indoor areas [14], such as office buildings, malls, airports, etc; and these spaces are becoming increasingly complex and elaborate. For indoor location based services, detailed knowledge of the indoor environment is necessary. Unlike outdoor areas, indoor areas have no explicit network of paths; also spaces are separated from each other by walls and partitions, which effects human's cognition of the environment [14].

A well-formed representation of spatial knowledge is a major requirement for any location based application. Maps for indoor environments should provide both geometries of entities and implicit meanings of entities, also known as semantics [21]. The other difference between indoor and outdoor environments is the hidden complexity of indoor environments. Most buildings have more than one floor.

Although, building floors are usually stored in 2D in most databases [22], connections between different floors require a 3D model.

Topology and relations between features are required for navigational purposes [22]. Considering that there are different types of navigational tasks such as finding the optimal path to a destination, or finding a suitable path for people with disabilities [23], the node (vertex)/link (edge) model can be used for defining topological relations between the main objects of a building [22].

For this purpose, different models have been proposed, so far. The next section deals with concepts related to these models. Since basic queries in location based services are based on the location of physical objects [23], location models should be chosen to describe both geometries of objects, and relations between them properly. For instance, in order to search for objects in an area, within relations should be defined in a well known manner [23].

2.2 Indoor Environment Structure

Compared with outdoors, indoor environments are highly structured and more complicated [1]. Thus, describing indoor environments for LBS applications requires models that can support complexity. Generally, a location model should preserve both geometric information and underlying semantics of a location to be useful for an indoor location based service. However, semantics are generally application specific and will differ from one application to other [21].

A location model is the central part of any LBS, and stores a representation of static and mobile objects in the real world (e.g., buildings, points of interest, people) [23]. In addition, a location model helps the system to understand and capture the semantics of the physical world [24].

A proper location model provides the information necessary for location browsing, navigation, and query processing [1]. Attributes like speed, distance, and even highly dynamic attributes such as current traffic conditions should form part of the model [23]. A good location model should be able to provide an object's position, distance functions, topological relations, and orientation in order to allow positioning, ranging, routing and navigational queries [23].

Location models can be classified according to an application's requirements, and type of queries performed by a location based service [23]. Two basic classes of location models are defined as geometric and symbolic systems [23].

2.2.1 Geometric Location Model

In a geometric location model features are presented by geometrical figures [23]. A Euclidean space is typically used as the fundamental reference to represent objects and features in geometric models. Basic components of these models are points, lines, polygons, and volumes [1]. This is the general model that is used for mapping purposes. Most outdoor LBS applications use this model to represent the real world to clients. A set of coordinates in the space is used to describe objects in the geometric model [1]. This model is also compatible with positioning systems such as GPS.

In geometric models, coordinate tuples describe positions. These coordinate tuples can be global or local, based on the reference coordinate system that defines them. Geometric coordinates can be used to calculate distance, direction, area and other geometric attributes of features. In addition, some of the topological relations between different features can be extracted from these coordinates [23]. GPS, as a globally operating system, provides coordinates as a longitude and latitude pair, with reference to the WGS84 coordinate system.

Accurate position representation and flexible information retrieval are the main advantages of using geometric models [1]. Although, geometric models do not provide relationships between entities in the real world, some topological relations can be derived from the geometries of features (e.g., containment) and the others have to be defined explicitly (e.g., connections) [23].

Figure 2.1 shows a geometric model of an indoor area. Different rooms within the building are illustrated by polygons, i.e., boundaries of polygons model the boundaries of partitions in the real world.

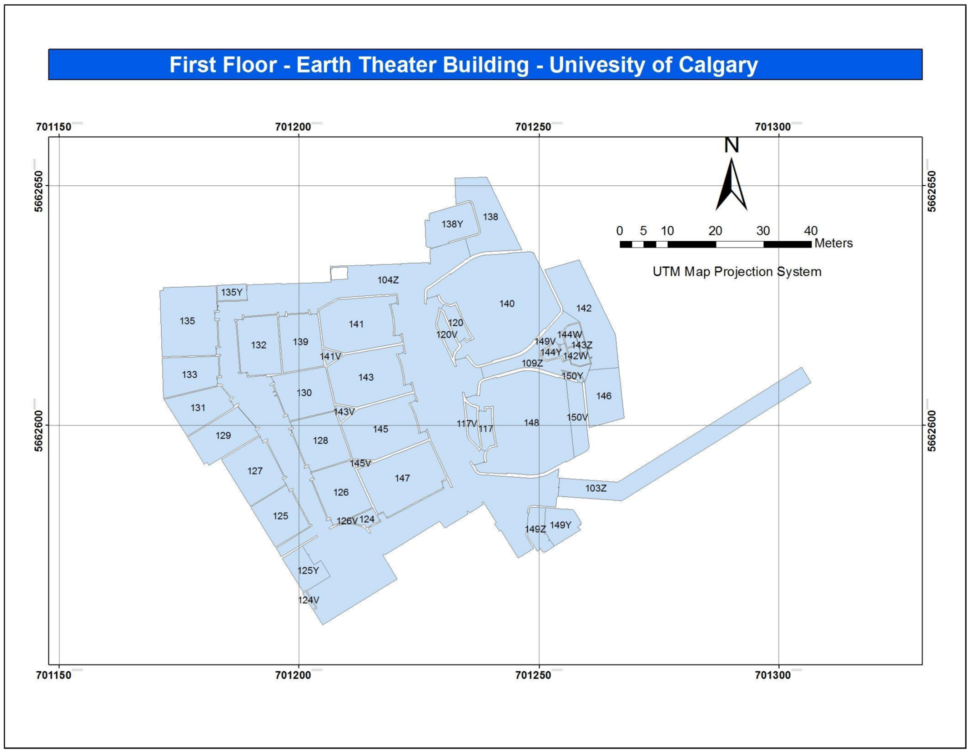


Figure 2.1: Representing an indoor environment according to the UTM geometric model

2.2.2 Symbolic Location Model

In symbolic models all objects and features are presented by symbols and referred to by names [1]. Symbolic coordinates define positions in the form of abstract symbols. Additional information such as geometrical and topological relationships of features are not included in a symbolic model [23], but can be attached to symbols explicitly. Symbolic models attempt to represent logical entities and their semantics. Although symbolic models do not deal with the geometry of the entities, substantial effort can be required to describe real world entities [21].

Set-based, hierarchical, and graph-based models are the dominant symbolic models for location presentation [23]. The main advantage of symbolic models is that they represent spatial relationships with semantic information, which makes it easy for people to understand [1]. Two fundamental semantics in a symbolic model are: the topological relation and distance between locations [21]. Each of the models have their own advantages and drawbacks, which are described below:

Set-based model: In a set-based location model, all symbolic coordinates belong to a global set L. Locations that consist of several symbolic coordinates are defined by a subset of the set L. This model can be used to find contained and overlapping features. Also for range queries, this model can be used where features in a particular range can be defined by a subset of L [23].

Set-based models can represent containment relationships without using the geometry of objects in the space [1]. The following example shows how this model represents real world entities.

$GlobalSet = L = UniversityofCalgary$

$L = \{EngineeringBldg., \ldots, CCITBldg., 1stfloor, \ldots, 12thfloor\}$

$EngineeringBldg. = \{ABlock, \ldots, FBlock\}, CCITBldg. = \{\ldots\}, ICTBldg. = \{\ldots\}, \ldots$

$CBlock = \{C101, C102, \ldots, C228, \ldots, C332\}, A = \{\ldots\}, B = \{\ldots\}, \ldots$

$STBuilding = \{ST120, ST125, \ldots, ST149Y\}$

Containment can be inferred from set operations. If an entity belongs to a set or is a subset of a superset, we can infer that entity is contained by its superset in the real world. Overlapping relations can also be inferred from set-based analysis. If two sets have an entity or other set in common, in other words, if two sets intersect, then we can infer that they overlap in real world [23].

Hierarchical model: Hierarchical models consist of a set of locations, which are ordered according to their spatial containment. (i.e. a location l_1 is an ancestor of other location l_2 if l_2 is spatially contained by l_1) [23]. Ordering these symbols forms a hierarchy that takes the shape of a tree or a lattice [21]. In hierarchical model, objects and locations are described by unique names (known as symbols).

A hierarchical model is consistent with human cognition of indoor space [1], which makes it suitable for visualization on mobile devices. However this model is not suitable for presenting moving objects and only accessibility relations can be modeled by this method [1]. Smaller scales of geometric models correspond to root levels of hierarchical models, and larger scales correspond to leaf levels. When using this approach, if different locations overlap each other, a more general lattice-based model can be applied, where each location can have more than one ancestor [23]. Figure 2.2 represents a hierarchical symbolic model of the Schulich School of Engineering of the University of Calgary.

Hierarchical models are good for some queries. For example, they naturally support range queries as they are based on a containment relation [23]. But for other queries they are not very good; for instance, they do not provide means to model interconnections between locations [23] and accessibility cannot be extracted from this model, which makes them improper for navigational purposes.

Graph-based model: a graph-based approach uses vertex (node)/edge (link) concepts to model connections between different locations [23]. A graph can describe connectivity and accessibility of floors if the relations are captured appropriately [14]. In graph-based models the emphasis is on physical connectivity and accessibility

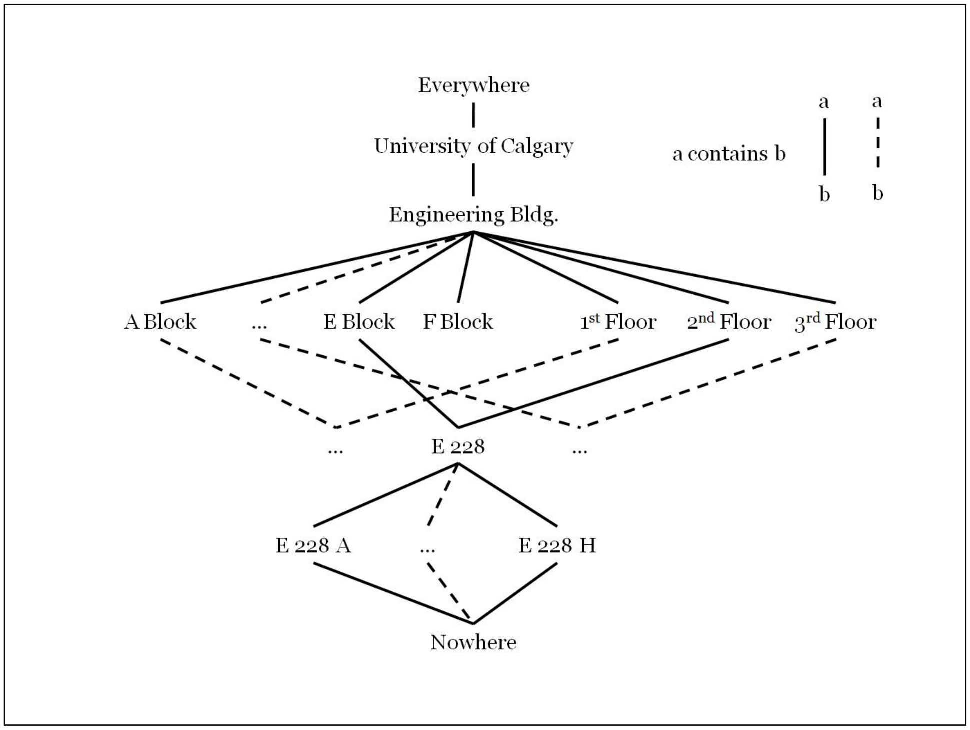


Figure 2.2: Representing an indoor environment using a hierarchical model

between neighbor locations.

Jensen et. al. [14] proposes the use of connectivity base graphs and accessibility graphs to demonstrate connections between different partitions of a building. According to connectivity based graph models, each partition in a floor plan (e.g. room, staircase, and hallway) is represented as a vertex in the base graph. In addition, exterior areas of an indoor space are represented by a single vertex. Edges are used to capture physical connectivity between vertexes (i.e. partitions). Figure 2.3 demonstrates a graph-based model for the indoor environment presented in figure 2.1 (geometric model).

Graph-based models works well for queries related to path finding, navigation and nearest neighbor analysis [23]. However, in order to get trustworthy results this model requires some geometric refinements. In other words, the value (i.e. weight) of the edges should be compatible with geometrical distance between vertexes, or time of

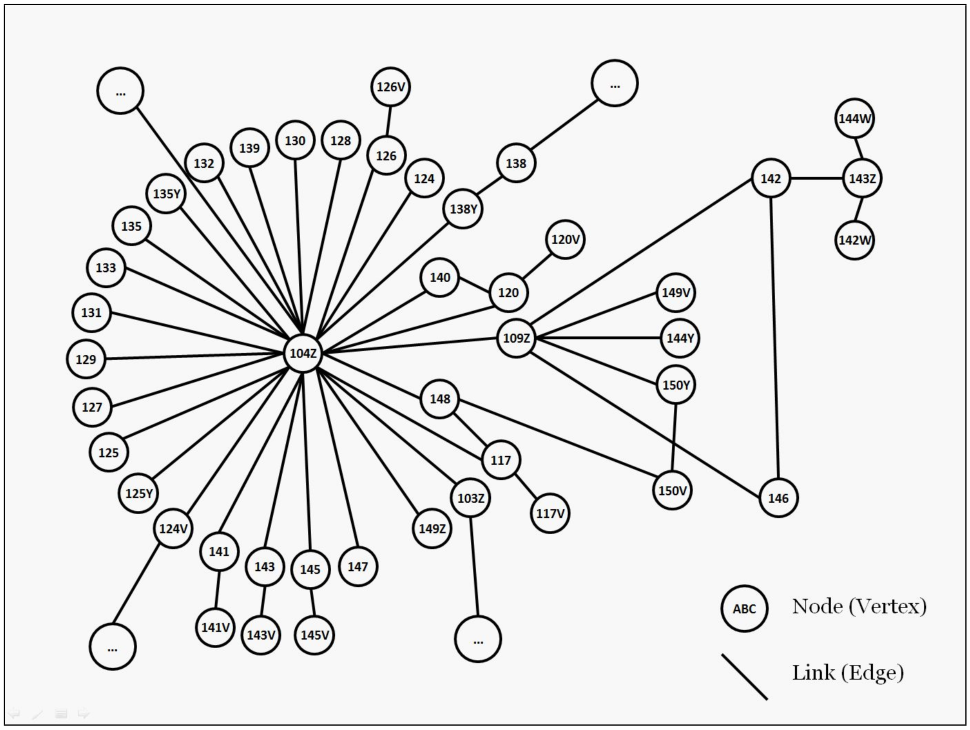


Figure 2.3: Representing an indoor environment with graph-based model

travel; depending on the model adopted.

Location-exit model: Graph-based model have been studied by different researchers and some have proposed variations of the graph-based model. For example, Li and Lee [1] propose an exit-location space model, which preserves topology and distance semantics between locations. Here, location refers to a bounded geometric area with one or more exits and an exit is a point on the boundary that is used for entering or leaving a location. For this type of model Q-analysis [25] is applied to analyze the topological structure of the location model. Through application of this model, containment and overlap relations between two entities can be revealed [24].

The location-exit model is a well-defined symbolic model for presenting indoor environments. ER diagrams can be used to define the underlying infrastructure to represent this model. Entities of this model are real world entities such as buildings, rooms, and corridors [21]. This model can be categorized as a graph-based symbolic model which can be derived from the geometric representation of the indoor space [24]. Figure 2.4 illustrates the same environment presented in figure 2.1 as a location-exit model.

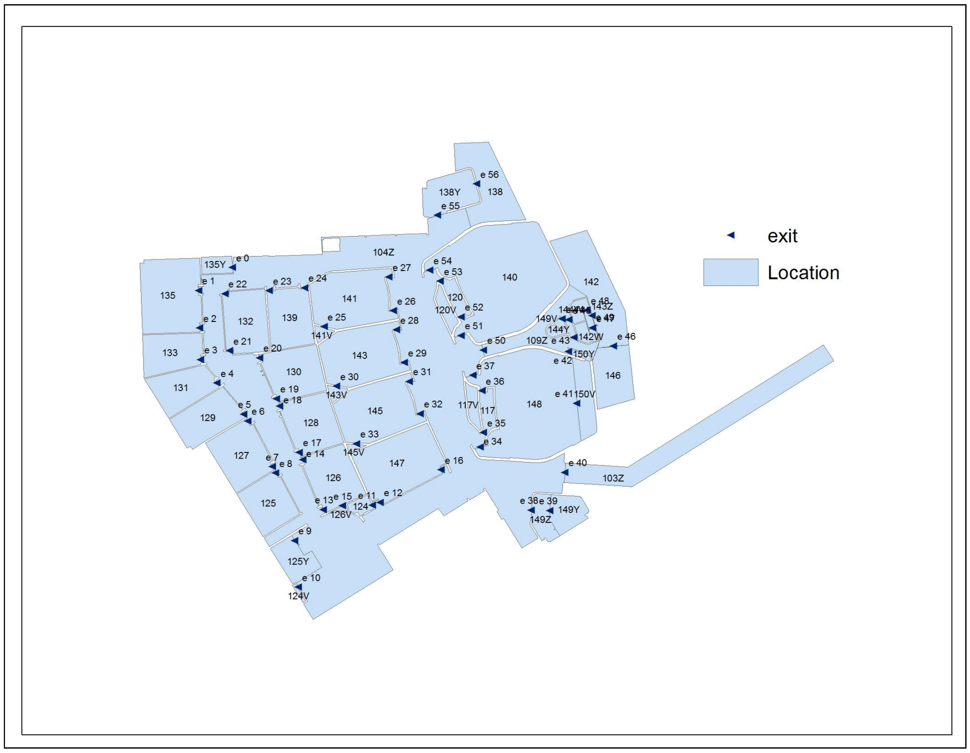


Figure 2.4: Location-exit model of the science theater, University of Calgary

The location-exit model creates a graph from locations represented by vertexes and exits, represented by edges which connect corresponding locations of vertexes to each other. In this model the optimal path is the one with the minimal number of locations/exits that must be passed through to reach from one location to another [24].

2.2.3 Hybrid Location Model

Hybrid models, which are combinations of both geometrical and symbolic models, are proposed by some researchers to overcome the weaknesses of these two models [1]. Both geometric models and symbolic models have their own advantages and disadvantages, which make them suitable for some applications. In order to take advantage of both, these two models are combined.

The main idea behind hybrid location models is that features are organized in a hierarchical model within a specific geometrical space at each level of the hierarchy [1]. Embedding a graph model in the geometric model at each level provides a base for some topological relationships between entities. In addition, network analysis and path finding algorithms require graph models. So, combining different models can provide an appropriate foundation for some basic queries for location based services. Table 2.1 summarizes some properties of these location models and their support for queries such as positioning, range and nearest neighbor analysis [23].

Table 2.1: Properties of location models

	Supported Coordinates		Supported Queries			Modeling Effort
	Symbolic	Geometrical	Positioning	Range Query	Nearest Neighbour	
Set-based	Yes	No	Good	Good	Basic	High
Graph-based	Yes	No	Good	Basic	Good	Medium
Hierarchical	Yes	No	Good	Good	Basic	Medium
Hybrid	Yes	Yes	Good	Good	Good	High/Very High

It is worth noting that during the integration process of a graph-based model with a geometric model, some locations may be represented by more than one node. For example, a rectangular corridor that has four corners, each corner will be represented by one node. According to Hlscher et al. [26] horizontal areas may have more than one choice point (points on which one can decide to take different routes) or directional change. In such areas each choice point, or each point of directional change is a node in a graph-based model.

2.3 Definition of Positioning in Indoor Environments

"The determination of physical location of an object is usually referred to as positioning, location estimation/identification, localization or geolocation" [27]. In indoor positioning some aspects of the environment are observed and the location or position is inferred. In general, all positioning systems are based on three primary techniques: Triangulation, Scene Analysis, and Proximity [28].

When using triangulation, the distance between a point of interest and some reference points are measured to determine the point of interests' position. Distance can be measured directly, by measuring angles, by time-of-flight or using attenuation [28]. In scene analysis, we try to match the pattern of a variable with a pre-observed set of observations; while in proximity positioning, we assign the position of the base station from which the client is observed, as the position of the client.

Location information can be presented in different formats [23]. However, we prefer to choose a format that is most compatible with our location model. Based on the output of the positioning system, these systems can be divided into two classes: geometric and symbolic positioning systems [5], compatible with geometric and symbolic location models.

Other classification of positioning systems distinguishes between two kinds of positioning: discrete and continuous. In discrete positioning, locations are selected from a set of points acquired during a data collection step (Microsoft RADAR is such a system `http://research.microsoft.com/en-us/projects/radar/default.aspx`); while continuous positioning locates a mobile terminal within an infinite set of positions [18].

Using latitude, longitude, and altitude for positioning is an example of a continuous positioning system, which is utilized by Global Positioning Systems (GPS). These values (Latitude, Longitude, and Altitude) describe positions in outdoor environments properly, but for indoor environments GPS is not very useful, and locations in indoor environments are often recognized in conjunction with other locations [1], which require a comprehensive understanding of the location model.

Geometric positioning systems use coordinates to represent the position of a client, however, symbolic positioning systems reveal the symbolic location identifier as the position of the client. Even if the indoor positioning system provides coordinates as the result of a positioning process, it should be able to convert coordinates to entities represented in a symbolic location model to fulfill the requirements of indoor LBS [5].

2.4 Definition of Path Finding in Indoor Environments

Navigation is one of the major use cases of indoor location based services [22]. The main purpose of navigation is to guide a client from his/her current location to a desired destination. This guidance is performed by finding and suggesting a path and providing instructions that enable the user to follow the path. Path finding abilities in indoor environments rely on the spatial properties of buildings (i.e. structure of the building) or on signs inside the building [20]. If a client is new to an indoor environment, navigation and path finding becomes difficult. Four major reasons that make path finding difficult in indoor environments are: lack of people to guide, complex interior design, unclear signs, and difficulty in locating buildings and main entrances [20].

The classical methods of path finding are based on graph theory (i.e. node/link structure) [22]. Graphs are abstract constructions made of nodes and relations (links) between nodes, represented by edges. Because of their flexibility, extensibility, and general framework, graphs are used to solve many real world problems [29], including path finding for location based services.

In mathematics, a graph (G) is an abstract representation of a set of objects where some pairs of the objects are connected by links. The interconnected objects are represented by mathematical abstractions called vertices (V), and the links that connect vertices are called edges (E). Typically, a graph is depicted in diagrammatic form as a set of dots for the vertices, joined by lines or curves for the edges ($G = (V, E)$) [30]. If the edges in a graph have no orientation the graph is undirected, otherwise it is a directed graph. The connections between locations in an indoor area can be modeled as a graph and is usually referred to as a network. Indoor path finding solutions are based on a network inferred from the graph-based model in which accessibility/connectivity between locations is modeled.

In order to solve the path finding problem, some definitions are required:

Network: Network is a connected linear graph consisting of nodes/vertices (p) and links/edges (s). Value/weight of a link is a positive number ($v(s)$). Each link has an initial node ($i(s)$) and a terminal node ($t(s)$) [31]. Figure 2.5 depicts a network of possible connections between some locations in an indoor area.

Path: A path (P) is a set of links $\{s_1, s_2, \ldots, s_j, \ldots, s_z\}$ with the following properties:

1. $i(s_1) = i_0$ is the node of origin
2. $t(s_z) = t_d$ is the node of destination, and
3. $t(s_j) = i(s_{j+1})$

Figure 2.6 illustrates a path between two locations in an indoor network.

This is a general definition of a path and can be applied properly within a graph-based location model. Considering the location/exit model, a path from location a to b is a sequence of exits $(x_1, x_2, \ldots, x_s)$ where $x_1 \in a$, $x_s \in b$ and $\forall i, x_{i+1}$ is directly reachable from x_i. A path can also be assumed to be a sequence of locations, as each exit belongs to a location [21].

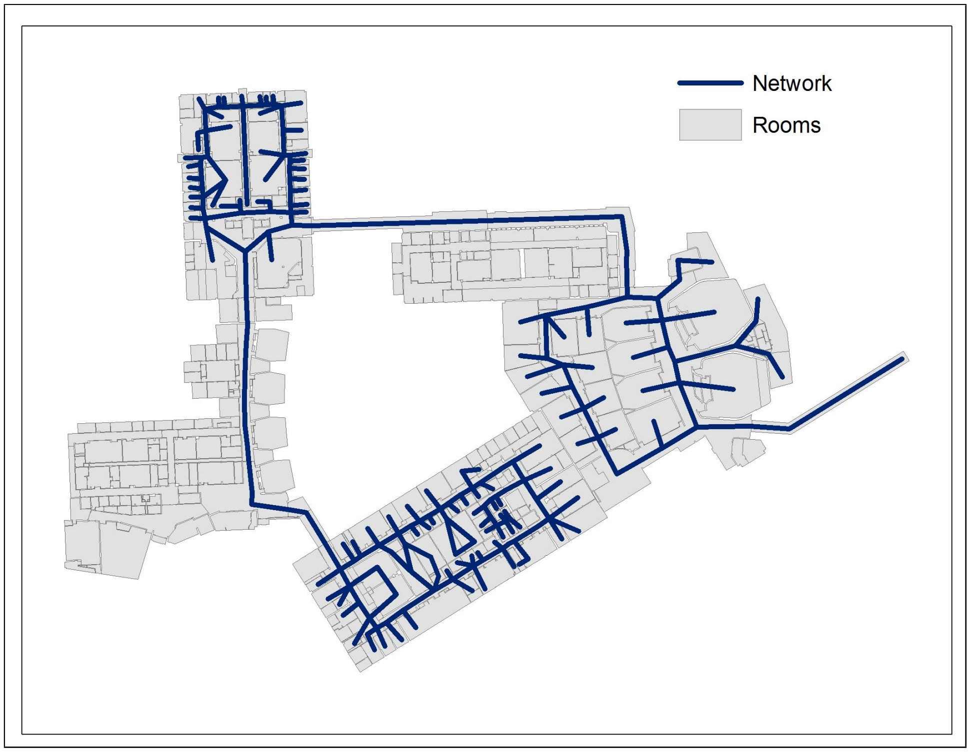


Figure 2.5: Network of connections between locations in an indoor area

The value of a path $v(P)$ is the sum of the values of the links defining the path [31]. If the graph model has emerged from a geometrical model, the link value can be represented by the true link length, making the graph suitable for optimization solutions such as finding the shortest path between two nodes on a network. However, other metrics can also be considered, such as time, cost, impedance, etc. In these cases the shortest path is not necessarily shortest in distance, but smallest in value (e.g., time, cost, impedance, etc.).

In the symbolic location/exit model, an exit x is directly reachable from exit y, if and only if there exists a physical route from y to x which involves no other exits [21]. In this model, the value of each link (i.e. connection) is assumed to be one and the value of a path corresponds to the number of exits through the whole path.

Navigation aims at finding an acceptable path in the network and guides the user

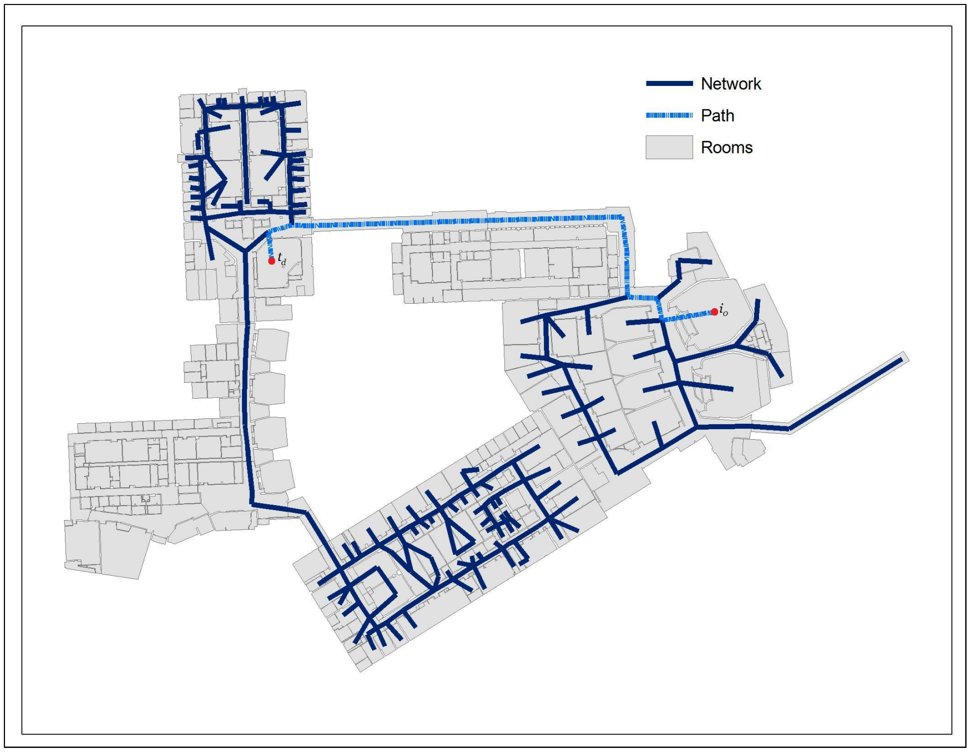


Figure 2.6: A path between two arbitrary locations in an indoor area

through the path [19]. The process of navigation for people includes sign or landmark recognition, turn angle estimation, path sequencing, network comprehension, etc. For humans, navigation in space can be based upon external representations such as maps and graphs, or based on internal representations derived from experiences [32].

Wolbers and Hegarty [32] and Klippel et al. [33] compare and distinguish differences between human spatial cognition of paths (as cognition maps) and technological representation of paths (as cartographic maps). Liu et al. [34] discusses the design of a functional user interface that provides directions that are compatible with human spatial cognition. In their study, instead of cartographic maps, images and directional arrows have been integrated into the interface to present the user with appropriate navigation aids.

There are three different strategies for human navigation in an indoor environment [26]:

1. the central strategy, which relies on navigating through well-known parts of the building, 2. the direction strategy, which aims at horizontal routes to the destination, and 3. the floor strategy, in which the navigator heads toward the vertical position of the destination.

Human cognition of space often contains errors, so technical methods can help overcome human cognitive limitations [19]. As mentioned previously, the node/link structure of a network can be used to model the network of paths and is supportive of navigation [22]. Humans use view and place graphs for path finding purposes [29]. In this type of graph all places that can be viewed from a place can be connected together. The same model can be utilized in order to model a network for an indoor environment. The first step is to build a 3D topological model of an indoor area, consisting of rooms, corridors, stairs and halls [22]. In a multi-level complex indoor environment with inconsistencies in each floor and dead-end locations, appropriate strategies are important for effectively solving the path finding problem [26]. The graph model for path finding should be able to support these strategies. However, studies reveal that humans prefer to use a set of central paths (i.e. main paths or skeleton) in an indoor environment, as the foundation of their navigational strategies [26].

The overall graph model for a multi-level indoor environment can be summarized using the following method: Each floor can be described as a network consisting of nodes and links. For indoor environments vertical links are used to model connection between floors (e.g., stairs and elevators) [22]. Each location may be presented by one node (e.g., rooms) or more than one node (e.g., corridors). In each location, doors, point of choices, junctions, and points of change in angles can be modeled as nodes. Links are designed in a way that each node is connected to all nodes that are visible from that node. Connections between different floors are also modeled by links. For example ladders, stairs, or elevators that connect different floors can be modeled by links.

Navigation can provide a huge number of paths from an origin to a destination, based on the desires of users, strategies for navigation, different times and situations, and flow of the crowd. Most applications attempt to minimize or optimize the value (i.e. cost) of the path. Algorithms like Dijkstra shortest path algorithm [35] can be used for this purpose. However, in some other applications not only the shortest path, but a number of shortest paths can be utilized for this purpose. K shortest path problem deals with these situations. In chapter 4, this problem will be discussed in detail.

2.5 Summary

In this chapter we investigated different location models for positioning and path finding. Symbolic models are more compatible with positioning and location definition of indoor environments, while geometric models are better for outdoor areas. However, we may be interested in geometric models for positioning and convert the results to a symbolic model. For path finding, a graph-based model is the best approach for modeling the network of paths between different locations. However, this model is not explicit for indoor environments and we need to reconstruct this model from a geometric model of the environment. Hybrid models try to take most advantage of both geometric and symbolic models by combining them together. Different applications require different location models to function properly. Hybrid models provide us with the opportunity to choose different aspects of each model and combine them together according to our needs.

Chapter 3

Indoor Positioning and Path Finding Techniques

In this chapter different technologies for indoor positioning are reviewed and methods for location fingerprinting are discussed. This chapter presents basic concepts for different location fingerprinting methods that have been implemented. Each of these methods adopts a unique approach for solving the positioning problem in indoor environments and addresses the first objective of this thesis.

3.1 Review of Technologies

Common positioning systems find location information with the help of a satellite system [2] and these systems are helpful when a direct sight exists between satellites and the receiver, which makes it attractive in outdoor environments where there are few obstacles blocking the signals from satellites.

Indoor positioning requires different infrastructure. There are a wide range of indoor sensors utilizing various techniques for positioning purposes. Infrared and laser transmitter/receiver systems, ultrasonic sensor/actuator systems, computer vision systems, physical contact, close proximity radio identification and WiFi systems are examples of indoor positioning systems [27]. Each system has its own advantages and disadvantages and choosing one system depends on the user's technology and the application. Some of the most popular indoor positioning systems and their properties are summarized below:

Active Badge: (`http://www.cl.cam.ac.uk/research/dtg/attarchive/ab.html`)

Active Badge is the oldest indoor positioning system. In this system a badge attached to a person transmits infrared (IR) signals periodically (e.g. every 15 seconds) and a set of sensors detect these signals. Each badge has its own unique code. A master control station, connected to the network of sensors, records received signals and determines the position of the badge with respect to the corresponding sensor by assigning the position of the sensor to the badge [36].

Active Bat: (`http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/`) Active Bat is an indoor positing system with accuracy in the range of centimeters. Active Bat systems utilizes a range based technique, which works by finding the distance to a minimum of three reference nodes and then uses the multi-lateration technique, or hyperbolic positioning [37] to find the position of the client. The time difference of arrival of infrared signals is used to find the distance from the reference nodes [37].

Cricket: (`http://cricket.csail.mit.edu/`) Cricket combines RF and ultrasound technologies for indoor positioning. Receivers in this system listen to both RF signals and ultrasonic pulses and calculate distance from each beacon using the difference in propagation speeds between RF and ultrasound (i.e. time difference).

RADAR:
(`http://research.microsoft.com/en-us/projects/radar/default.aspx`) RADAR is an RF-based positioning system developed by the Microsoft Research Group. This system uses signal strength information from wireless transmitters and determines the location based on the propagation model of the signal between receivers (e.g. laptops, PDAs) and transmitters (i.e. wireless network) [38].

Active Badge, Active Bat, Cricket and RADAR are some examples of available indoor positioning systems. These systems could find their applications in different areas, although each of these systems requires its own type of hardware/software infrastructure. "WiFi-based positioning system (WPS) emerged as an idea that can solve the positioning in certain situations (like indoors), taking advantage of the rapid growth of wireless access points in urban areas" [16]. The next section provides more details on WiFi positioning.

3.2 WiFi Positioning

In WiFi positioning, existing infrastructure within buildings and available technologies in personal mobile devices are utilized to calculate a user's indoor position [39]. WiFi networks are usually used to provide internet access for users in an indoor environment. (e.g. a department, hospital, airport, home). A connection is made when access points (APs) talk to wireless adapters installed in mobile devices (e.g., laptops or PDAs).
A list of Access Points and their location usually is the only requirement of this technology [39]. Most wireless adapters provide received signal strength (RSS) information from local access points either as a power measure or signal to noise ration (SNR) as connection quality [17]. The power of the signal usually ranges between -100[dBm] and -20[dBm] and connection quality is usually provided by ranges and colored-bars that indicate signal quality.
A variety of information can be extracted from received signals. From this information, signal strength and signal to noise ratio have the most importance; However, signal strength shows better correlation with location than the signal to noise ratio (SNR) [40], and for this reason most WiFi positioning systems utilize signal strength rather than signal to noise ratio.
WiFi characteristics and use of signal strength for positioning will be elaborated in next sections; but it is noteworthy to mention here that utilizing the existing WiFi infrastructure, with minimum additional devices, for positioning remains the most important advantage of WiFi systems [7] in comparison with other indoor positioning systems.

3.2.1 WiFi Characteristics

The WiFi system is based on IEEE 802.11 standards. ”IEEE 802.11 is a set of standards for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands.” [41] WiFi is designed for providing wireless connection in a local area and each WiFi station can roughly cover 300 meters in open space [17]. Signals that facilitate communication between wireless stations (a.k.a Access Points) can be used for positioning.

WiFi location estimation employs physical attributes of the received signal strength (RSS) from the access point, the angle of arrival of the signal, and time difference of arrival. However, received signal strength is the only attribute that most available hardware can currently measure without adding additional tools and therefore cost [11]. Measurements can be performed during normal operation of the wireless system [17]. In addition, WiFi positioning systems can work with any type of firewall, or other restrictive policies [17].

As mentioned, radio signals can be used to detect the position of a device [17]. This can be done by modeling signal propagation through space. Signal propagation modeling is based on a signal predictor variable, where the independent (observed) variable is signal strength (in dBm), and the dependent (predictor) variable is the distance from a reference station, and the main estimation parameter is the exponent α, which determines path loss of the signal from the signal source [16].

Under the assumption that signal strength is related only to the distance between the transmitter and the receiver, following the Motley-Keenan propagation model [16] the distance between a receiver and the transmitter is:

$$P(r) = P(r_0) - 10\alpha log(\frac{r}{r_0}) \tag{3.1}$$

where $P(r)$ [in dBm] is the power received by a given Mobile Station (MS) whose distance from a given transmitter or AP is r (meters). r_0 is the reference distance from the transmitter and $P(r_0)$ is the signal power at that reference distance. The parameter α indicates path loss with respect to an increase in distance [16].

In free path loss environments (e.g., a Fresnel ellipsoid), $\alpha = 2$, and in indoor environments α is close to 3. For indoor environments this parameter needs to be tuned to characterize the structure of the building. α is important as it may significantly affect the estimation of the distance [42]. In a laboratory situation, WiFi signal decays linearly with log distance and a simple triangulation can be used to estimate position of a point in a 2D coordinate system when receiving signals from three different access points [11]. However, investigations show the range estimation from this simple propagation model leads to poor positioning, as it does not take into account the effect of obstacles such as walls [42]. In addition, in practice, activities and obstacles block signals and add significant noise to received signal strength measurements [11].

Deriving a practical model for an indoor environment and its electromagnetic signal propagation is extremely complex [17]. Equation 3.1 does not consider obstacles such as walls found in an indoor environment through which signals pass. So in practice the signal strength may be overestimated [16]. Considering these situations, an empirical model can be expressed as:

$$P(r) = P(r_0) - 10\alpha log(\frac{r}{r_0}) - l.WAF \tag{3.2}$$

Where l is the number of walls between the transmitter and the receiver; and WAF is the wall attenuation factor [16]. An alternative radio wave propagation model, based on

Friis relations, is defined by:

$$\frac{P_R}{P_T} = G_R G_T (\frac{\lambda}{4\pi d})^2 \tag{3.3}$$

where: P_R and P_T are, respectively, the Signal Strength (SS) received and emitted. G_R and G_T are, respectively, the receiver and transmitter antenna gains. λ is the carrier wavelength and, d is the distance between the receiver and the transmitter [18]. The Friis relation expresses signal strength as a function of distance in free space [18].
Since WiFi technology is accessible and low cost [18] and the infrastructure is being deployed in most indoor areas, utilizing this technology for positioning is becoming more popular. Two different methods can be adopted for this purpose. Mathematical methods in which the relation between distance and signal strength is modeled, and empirical methods in which position is determined by matching the data from unknown locations with data from a large set of known locations, using different algorithms. Empirical methods are also known as location fingerprinting and are detailed in next section.

3.2.2 Location Fingerprinting

Taheri et al. state that: "Location fingerprinting is a technique for location sensing on 802.11 Wireless Local Area Networks (WLAN), using commodity WLAN cards and no additional hardware tags" [13]. Deployment of positioning systems based on fingerprinting can be divided into two main phases. First, performing a site-survey to collect signal strength patterns from different APs (a.k.a offline phase/training phase/calibration phase). Second, online phase, reporting the signal strength measurements of a mobile device to a server to determine the mobile device's location, based on a comparison of measured signal strength and measurements from the offline phase (a.k.a positioning phase) [40].

In other words, the location fingerprinting method uses signal characteristics (e.g., RSSI) from a set of known points to compare against signal received by a mobile device to infer the position of the mobile device. The whole process includes collecting signal strength information for each calibration point for which position is known and building a model from the information to predict user's positions. The model building phase creates a predictive model that maps signal strengths to locations [11]. During the online phase the predictive model is used to estimate location of a client using received signal strengths [11].

Location fingerprinting techniques make use of signal maps rather than time or direction of arrival for determining the location of a client [40]. In WLANs an easily available signal characteristic is the received signal strength (RSS) [7]. Since, it is far easier to obtain signal strength over other characteristics such as multipath characteristics, time of arrival or angle of arrival [7].

In a heterogeneous environment, signal strength is a complex function of distance, geometry of a buildings [43], construction material, and signal propagation issues (i.e. refraction, reflection, and multipath) that can cause the signal strength to correlate poorly with distance [28]. The reason is that signal strength is not immune to the effect of radio interference, resulting from building materials [13]. So the main idea of location fingerprinting is to consider all these factors together in one model rather than modeling each of them separately.

In comparison to GPS, WiFi does not calculate the range between receiver and access point using timing information [42]. In order to find the range, a model relating range to signal strength should be developed; however, signal strength varies over distance, direction, time and hardware simultaneously [13]. One of the advantages of the location fingerprinting method is that it is independent of these variations. It is also independent of the variations of signal strength due to obstructions and multipath fading effects [7], assuming that no changes are made in infrastructure. In other words, the location fingerprinting technique does not require detailed knowledge of the access points' location nor the buildings characteristics [43], as long as these locations and characteristics remain unchanged during both the calibration and positioning phase. Users expect that location detection or positioning software be trained quickly, so time for collecting calibration data sets and training the system is minimized [17]. Finding a functional relationship between the position of a mobile user and raw RSSI measurements is a complicated task. Performing the inverse problem of finding positions based on RSSI is more complicated and demanding [17]. Based on these requirements different location fingerprinting techniques have been investigated.

Taheri et al. [13] used proximity-matching algorithms to determine the location of a mobile device. These algorithms are based on least threshold and Euclidean distance. Interpolation and averaging methods are implemented to increase the accuracy of positioning results. Hermersdorf [39] used a counting and weighting methodology to relate APs in a mobile device to APs in each section of a building. In this method the MAC address of the most visible AP with the strongest signal is detected and the section of the building in which the physical Access Point with the same MAC address exists is selected as the location of the device.

In location fingerprinting a location algorithm compares an observed set of values with a known set to determine location [13]. Based on a database of pre-recorded measurements of Received Signal Strength Intensity (RSSI), sampled from different locations within a building, position/location is estimated by inspecting the RSSI values a mobile device currently receives [12].

The standard location fingerprinting methods use supervised learning techniques and algorithms. The training data comprises vectors of signal strength for each known location. The dimension of each vector equals the number of access points from which signal strengths are measured at the location [11].

A wide range of location fingerprinting methods and algorithms has been proposed. The K Nearest Neighbour algorithm, Support Vector Machine, and Artificial Neural Networks [11] are some of the dominant algorithms and methods that have been adopted. The next section is dedicated to describing these location fingerprinting algorithms.

3.2.3 K Nearest Neigbour

K Nearest Neighbour is a technique for classification. This method assigns the location for a sample to the calibration sample that is closest in signal space [43]. For a location fingerprinting task, a grid of known points (with known position) along with their signal strength information is collected from observable APs and forms the training/calibration dataset, also called a radio map. A radio map (radio fingerprint) is a database of RSSIs associated with the position of points for which these RSSIs are collected [44]. In this method the distance between the vector of received signal strength at an unknown point and the points in the calibration dataset is calculated. This distance is not a geometrical distance, but a distance in signal space. The K closest points from the calibration dataset have the smallest signal distances from the unknown point. The position of the unknown point is calculated as a simple, or weighted, average of the positions of the K selected points [43].

According to [43] if the training set consists of N samples, in order to compute the position corresponding to a set of signal strengths $(RSSI_1, RSSI_2, \ldots, RSSI_m)$ this process should be followed:

1. For each $i \in 1, \ldots, N$ compute the distance d_i between $(RSSI_1^i, RSSI_2^i, \ldots, RSSI_m^i)$ and $(RSSI_1^u, RSSI_2^u, \ldots, RSSI_m^u)$ in which $RSSI_j^i$ is the observation of signal strengths from access point j at point i; and $RSSI_j^u$ is

the observation of signal strength from access point j at the unknown point u. d_i is computed by equation:

$$d_i = \sqrt{\sum (RSSI^i - RSSI^u)^2} \tag{3.4}$$

2. Find the k smallest distances: $d_i^1, d_i^2, \ldots, d_i^k$ where d_i^j is the j^{th} closest distance from a vector in training dataset to the unknown point. It is necessary to mention that each point in the calibration dataset has a known position, a pair of coordinates (x, y).

3. Compute the estimated position as the average of the positions recorded for the k samples selected at the previous step:

$$\begin{aligned} x &= \sum \frac{x_i^j}{k} \\ y &= \sum \frac{y_i^j}{k} \end{aligned} \tag{3.5}$$

Figure 3.1 illustrates the K Nearest Neighbour algorithm.

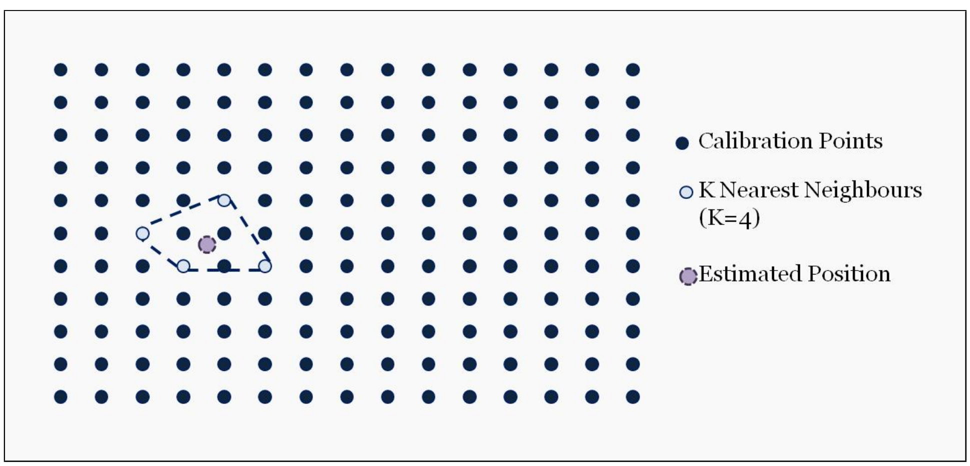


Figure 3.1: K Nearest Neighbour Algorithm

3.2.4 Support Vector Machine

Support Vector Machine is a set of classification methods used to analyze data and detect patterns [45]. SVM was originally developed for binary classification [46]. In other words it takes a set of input data and predicts the class it belongs to [45]. This method can be generalized to predict for more than two classes.

The idea behind SVM is to construct a hyperplane or a set of hyperplanes that have the maximum distance from samples of nearest classes [45]. This distance is referred to as the "margin". SVM uses a train and predict paradigm, in which classification parameters are determined using training/calibration data and then used for prediction with new data.

If the data samples construct a set of vectors (x) and two binary classes are represented by $+1$ and -1 values, the best hyperplane can be described by equation $w.x - b = 0$. So the problem can be changed to finding the values of w and b that provides the maximum margin. Details of the algorithm can be explored in [45]. Figure 3.2 illustrates the concept of SVM.

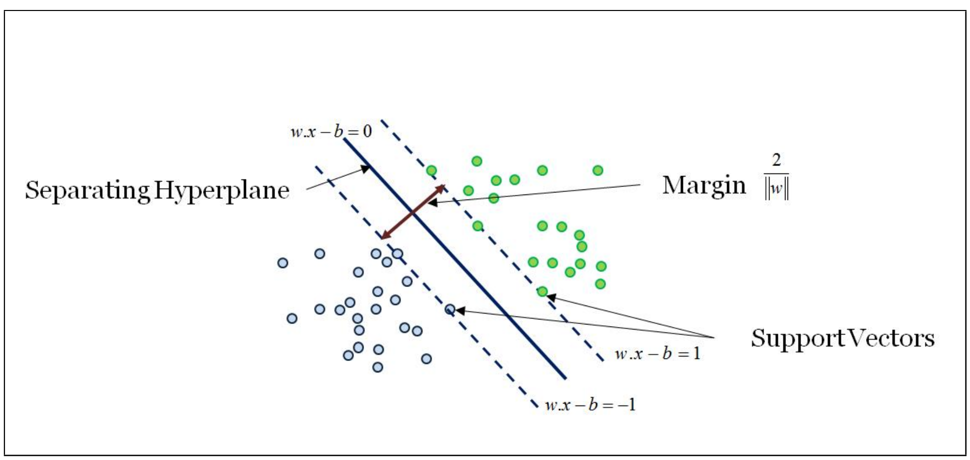


Figure 3.2: SVM concept

The original problem may be stated on a finite dimension space, but SVM uses kernels to project samples to higher dimensions/spaces to let the classes be linearly separable [45]. Common kernels utilized by SVM include:

1. Dot product: no mapping performed and only the optimal separating hyperplane is calculated; $K(x, y) = x.y$.
2. Polynomial functions: $K(x, y) = (x.y + 1)^d$, where the degree d is given.
3. Radial Basis Function (RBF): $K(x, y) = e^{-\gamma(||x-y||)^2}$ with parameter γ.
4. Sigmoid (neural) kernel: $K(x, y) = tanh(ax.y + b)$ with parameters a and b.

SVM, as a machine learning technique, is applied for general classification, regression, and outlier detection [46], and can be employed to classify signal strengths into different classes (each class represents a particular area such as a room, a corridor, etc.) for positioning purposes.

In location fingerprinting the x vector refers to signal strength observations from different APs at each point. Each point in an indoor environment belongs to a location such as a room or a corridor. As mentioned before, SVM basically looks for an optimal segmentation by creating hyperplanes between classes based on the maximizing distance between closest points within each class. In practice, SVM looks for the parameters that illustrate the hyperplane. After detecting hyperplanes, online signal strengths can be used as input for the SVM model and the classes (e.g., room, corridor, floor) can be predicted.

3.2.5 Artificial Neural Network

The complexity of deriving position from signal strength, and an incomplete understanding of the physics of an environment has lead to the employment of a flexible model based on a network of functions (i.e. neural networks) [47]. Artificial Neural Network (ANN) models and automated learning techniques provide effective solutions to estimate the position of a mobile device from signal strengths received at the device [43]. In fact, ANN tries to find a functional relationship between covariates (input variables) such as RSSIs and response variables (output variables) such as location [48]. The concept of ANN is based on human neural network in which neurons (neural cells) are connected to each other through synapses. ANN employs a similar concept known as multi-layer perceptron (MLP) [48]. MLP has a directed graph structure which consists of vertices (neurons) and directed edges (synapses). ANN typically includes three main layers: input layer, hidden layer, and output layer. Layers consist of vertices which usually are fully connected by edges [48]. Figure 3.3 depicts a scheme of an Artificial Neural Network.

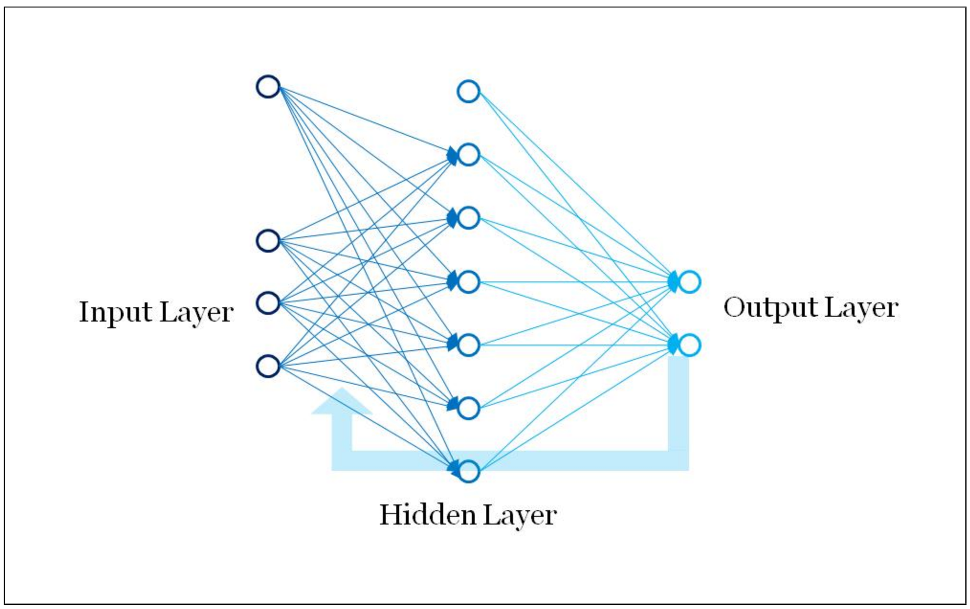


Figure 3.3: Scheme of an Artificial Neural Network

The ANN method consists of two steps: training and prediction. In the training phase a set of data samples consisting of input data (received signals at each calibration point) and output data (position of each calibration point) are fed to ANN. In location fingerprinting, neural networks are trained on observed data [48], which is the calibration/training dataset consisting of signal strengths associated with positions of calibration points. The objective of the training algorithm is to build a model with acceptable generalization capabilities for off-training/prediction set values [43].

Assuming the input layer has n neurons $x = (x_1, x_2, \dots, x_n)$, the hidden layer has m neurons $h = (h_1, h_2, \dots, h_m)$, and the output layer has k neurons $o = (o_1, o_2, \dots, o_k)$, ANN attempts to establish a linear function between neurons of each layer to produce the same outputs from corresponding inputs. These linear functions can be described as:

$$\begin{gathered} h_j = g_j(w_{0j} + \sum_{i=1}^{n} w_{ij} x_i) \\ 1 \leq j \leq m \end{gathered} \tag{3.6}$$

and

$$\begin{gathered} o_j = f_j(w_{0j} + \sum_{i=1}^{m} w_{ij} h_i) \\ 1 \leq j \leq k \end{gathered} \tag{3.7}$$

In other words, ANN tries to find the values of w (also referred to as the weight/value of each synapses) through an iteration process.

In general, ANN can approximate any complex functional relationship [48]. The computational cost increases exponentially with higher orders of complexity [48].

In location fingerprinting, neural networks are capable of establishing a relationship between signal strengths and position, utilizing linear and non-linear transformation and a large number of free parameters [43]. Finding the parameters of the model requires a supervised learning/training strategy, in which calibration data are used to construct the model and derive the parameters. These parameters are generalized to solve the positioning problem using new signal strength data [47].

As with the previously mentioned methods, distance from the access points and knowledge of the position of the access points are not required [43]. Also in practice an artificial neural network with a single hidden layer is sufficient, while the number of hidden neurons is generally large [43].

3.3 Summary

In this chapter major indoor positioning technologies were reviewed; including: Active Badge, Active Bat, Cricket, RADAR, and WiFi. We noted that WiFi is easier to utilize as most indoor areas are equipped with WiFi infrastructure as a mean for Internet access. Also we had a brief review of location fingerprinting methods applicable over WiFi systems. KNN, SVM, and ANN methods were discussed as dominant location fingerprinting methods and basic concepts were provided for implementing and testing the methods.

Chapter 4

Methodology

In this chapter we implement different methods of location fingerprinting and provide results from each method. We also introduce statistical distances as a method for location fingerprinting whereby instead of predicting position, we detect the points from the calibration set that have the best statistical match in terms of signal distribution with our observations in real time.

The path finding problem is also discussed in this chapter. Methods including shortest path, K shortest path, and all possible paths are discussed and compared. A database approach for finding all possible paths is also tested in this chapter.

4.1 Study Area

The first floor of Science Theaters at University of Calgary was chosen as the study area. This floor includes corridors and classrooms, and connects the Biological Science building to the ICT, math science and social science buildings. Figure 4.1 shows a map of the study area.

To take RSSI measurements an INSPIRON 6400 DELL laptop with Windows XP operating system was used. This laptop was equipped with a Dell Wireless 1390 WLAN Mini-Card with a BCM4311/BCM2050 chipset.

The software package used for collecting RSSI was inSSIDer software, which is a free and open source software application, licensed under the Apache License Version 2.0. The source code is freely available from a public repository at:
`https://github.com/metageek-llc/inSSIDer-2`.

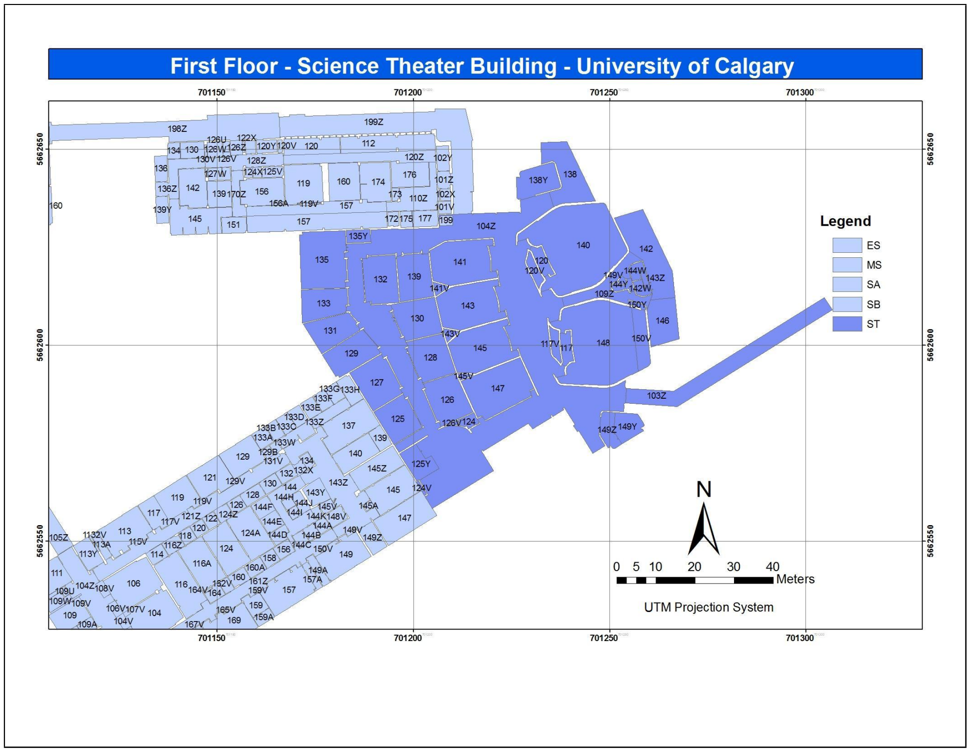


Figure 4.1: First floor - Science Theater building

The source code for the inSSIDer software were written in the C# programming language. inSSIDer scans access points within reach of computer's WiFi antenna, tracks signal strength over time, and determines their security settings (including whether or not they're password-protected). One can customize inSSIDer according to their needs. Figure 4.2 shows a snapshot of the interface of inSSIDer.

In this research inSSIDer version 1.2.7.0311 was utilized. The interface was customized to ask the user for the unique identification value of points. A function was added to provide information regarding signal strength, time and location for each test point in a flat file.

The file content is structured as follows: MAC address, SSID, Channel, SignalAverage, SignalSum, SignalNum, FirstSeen, LastSeen, SignalSd, PointId. Table 4.1 describes these parameters.

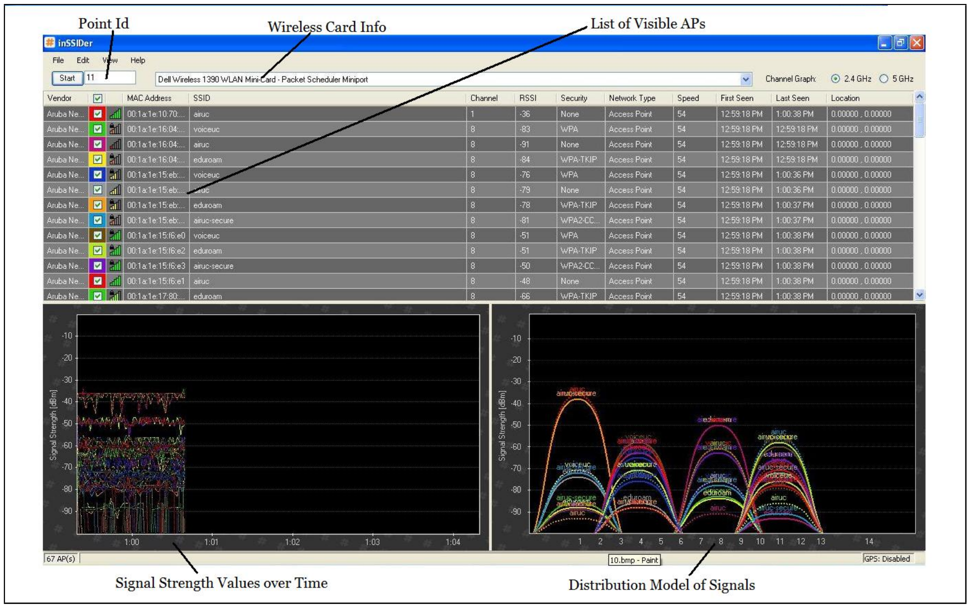


Figure 4.2: Interface of inSSIDer includes a list of visible access points, their distribution graph and signal strength of each access point over time

4.2 Dataset

Science Theater's first floor measures approximately 100 meters by 120 meters. A grid of 201 points covering the whole area was defined, but due to some physical constraints (closed areas, occupied points, and steep areas) 99 points were collected in total. These points have known coordinates (x, y) in the UTM projection system. The measurements were made over three different days. To collect signal information each point was visited between 10 to 25 seconds. Figure 4.3 depicts the position of these points. Through the data collection process 197 distinct MAC addresses were observed. Some of these MAC addresses belong to access points on other floors. Each access point transmits signals using multiple MAC addresses. In other words, the relation between access points and MAC addresses is one-to-many. There are two ways to detect and classify MAC addresses from the same AP. One approach makes use of the employed naming convention by the university. In our study the first 5 characters of the MAC

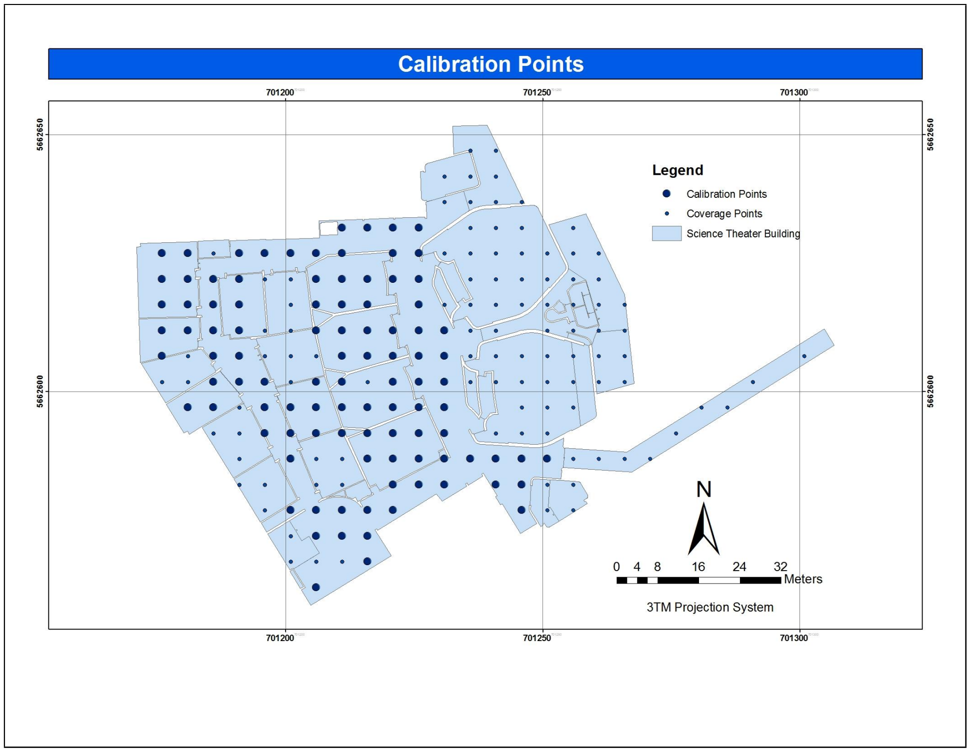


Figure 4.3: Points of collected signal information

addresses were identical. Table 4.2 classifies and shows an example of this classification.

Table 4.1: Parameters recorded by inSSIDer

Parameter	**Description**
MAC address	The MAC address of the access point transmitter for which inSSIDer receives signals. The MAC address consists of six hexadecimal characters within the range of 00 and ff, separated by a colon (e.g. 00:1a:1e:15:eb:c1). Each physical access point can contain several transmitters each with a unique MAC address.
SSID	The ID, or name of the access point. This name is not unique and usually states the group of access points to which the AP belongs. For example, airuc is the SSID for access points belonging to the over air network at the University of Calgary.
Channel	The signal channel. WiFi signals currently use frequency ranges of 2.4 GHz, 3.6 GHz and 5.0 GHz. Each range is divided into multitude of channels. There are 14 channels (1 to 14) in 2.4 GHz frequency used by WiFi networks in Canada.
SignalAverage	The average signal strength received over a time interval from a unique MAC address at a point. Typical values range from -100 to -20.
SignalSum	The summation of signal strength observations over time.
SignalNum	The number of times a signal was detected by theWiFi device.
FirstSeen and LastSeen	The time and date that a signal was first seen or last seen at a point. The format is DateTime mm/dd/yyyy hh:mm:ss.
SignalStd	The standard deviation of the signal.
PointID	The identification code of a point provided by the user at the time of observation.

Table 4.2: Examples of the main MAC addresses at the University of Calgary and observed MAC addresses

<table>
<tr><th>MAC address —SSID</th><th>Main MAC address</th></tr>
<tr><td>00:0b:86:c7:bc:20 — airuc
00:0b:86:c7:bc:21 — voiceuc
00:0b:86:c7:bc:22 — eduroam
00:0b:86:c7:bc:23 — airuc-secure</td><td>00:0b:86:c7:bc:20</td></tr>
<tr><td>00:0b:86:c7:be:20 — airuc
00:0b:86:c7:be:21 — voiceuc
00:0b:86:c7:be:22 — eduroam
00:0b:86:c7:be:23 — airuc-secure</td><td>00:0b:86:c7:be:20</td></tr>
<tr><td>00:0b:86:c7:c6:60 — airuc
00:0b:86:c7:c6:61 — voiceuc
00:0b:86:c7:c6:62 — eduroam
00:0b:86:c7:c6:63 — airuc-secure</td><td>00:0b:86:c7:c6:60</td></tr>
<tr><td colspan="2">The AirUC Wireless Internet service is the unencrypted wireless local area network at the University of Calgary that provides users with network connectivity.
Eduroam (EDUcation ROAMing) is a wireless network service that allows students, staff, and faculty from educational institutions to securely access the Internet while visiting other member universities. Users simply use credentials from their home institution and they are granted access.
AirUC-Secure is the encrypted wireless network at the University of Calgary.</td></tr>
</table>

The other approach used was correlation between signal strengths. Signals from the same AP show strong positive correlation. Figure 4.4 depicts an example of a correlation plot between signals from different MAC addresses from the same APs. These MAC addresses are: 00:1a:1e:17:80:61 — airuc, 00:1a:1e:17:80:62 — eduroam, and 00:1a:1e:17:80:63 — airuc-secure. For testing the positioning algorithms we tend to use the strongest signals that can be observed on the first floor.

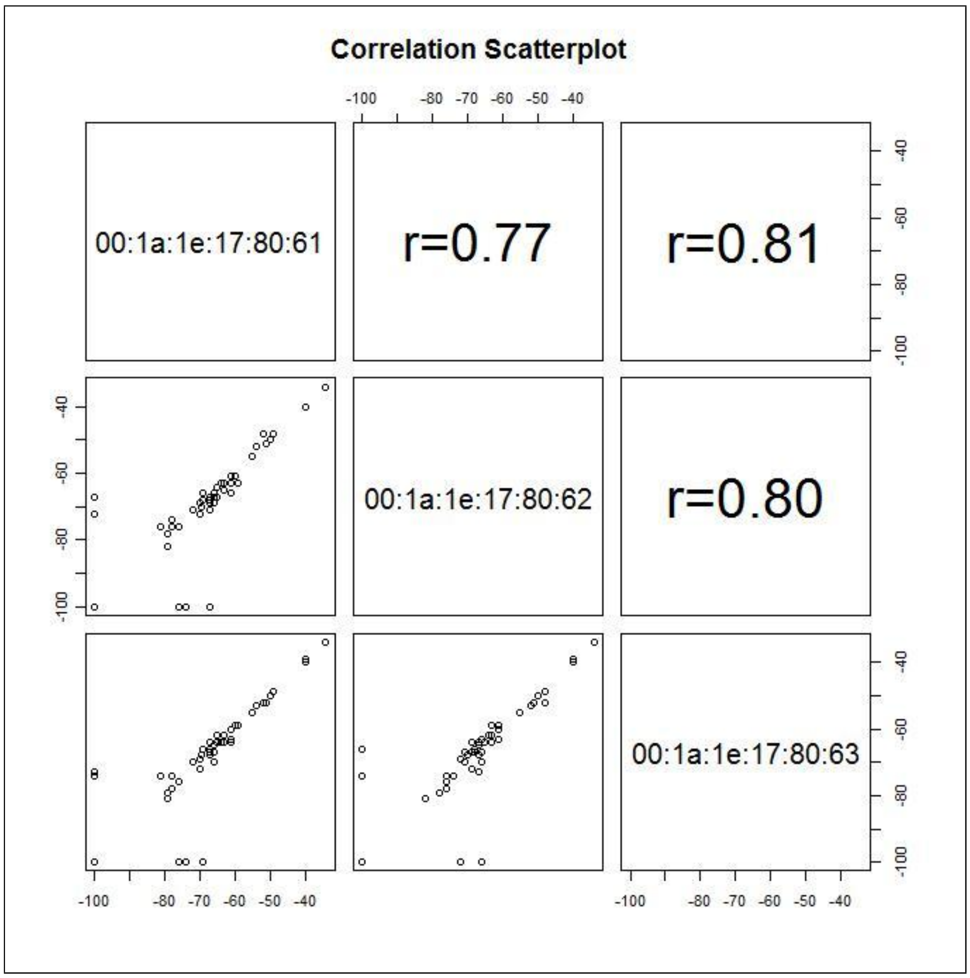


Figure 4.4: An example of a correlation scatter plot between signal strength values of MAC addresses from the same AP

4.3 Methodology for Positioning

As mentioned in the previous sections, RSSI data was collected from the first floor of the Science Theaters building at University of Calgary. The number of APs that can be seen by a mobile device varies depending on the RSSI, path loss, interference, and multipath fading at different locations[7]. These effects make the pattern of signal strength unique at each location and are known as fingerprinting of an AP. Prasithsangaree et al. compares the performance of fingerprinting systems using fine and coarsely spaced calibration points[7]. Choosing the grid spacing is a major challenge when implementing location fingerprinting. Large grid spacing reduces granularity/accuracy and small grid spacing increases cost with no guarantee of increase of precision [40]. In this study the distance between adjacent points was 5m. However, an objective of this research is to identify a fingerprinting method that decreases the number of calibration points as much as possible. However, from an operation perspective the granularity of positioning depends on the application [49]. RSSI samples at each test point can be presented as a vector: $R = (p_1, p_2, \ldots, p_n)$. Each component in this vector is assumed to be a random variable with the following assumptions:

1. The random variable $p_i[dBm]$ for all i are mutually independent. For this reason we remove redundant signals from different MAC addresses originating from the same APs.
2. The random variable $p_i[dBm]$ are normally Gaussian distributed[40].

Considering these assumptions we can test different location fingerprinting methods with our collected data. These methods include K Nearest Neighbour (KNN), Support Vector Machine (SVM), and Artificial Neural Network (ANN).

4.3.1 KNN, SVM, and ANN Methods

K Nearest Neighbour: To test the KNN method the R package "knnflex" was selected. This package allows us to test our positioning method and calculate precision and accuracy of the method. For K, values ranging from 1 to 16 were examined and the results were compared. Figure 4.5 to figure 4.8 demonstrate the results of this method for some values of K (K=1,3,7,11).

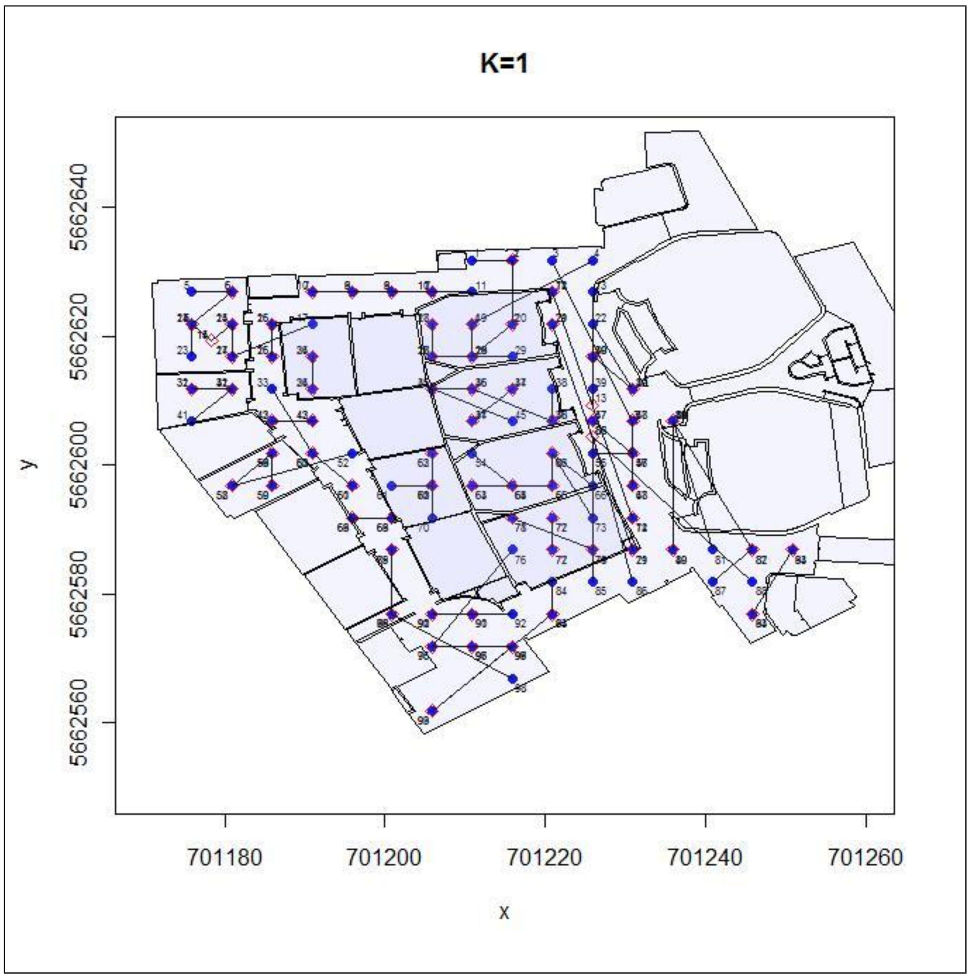


Figure 4.5: Results of KNN method, K=1

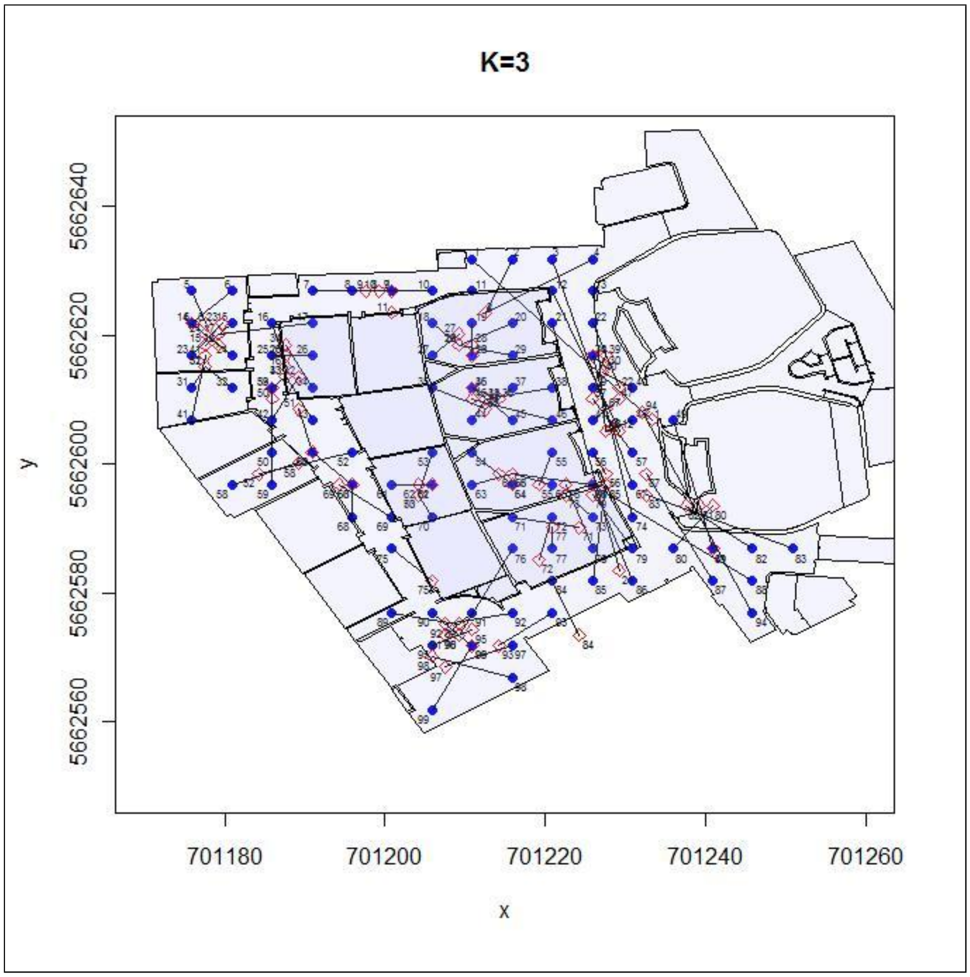


Figure 4.6: Results of KNN method, K=3

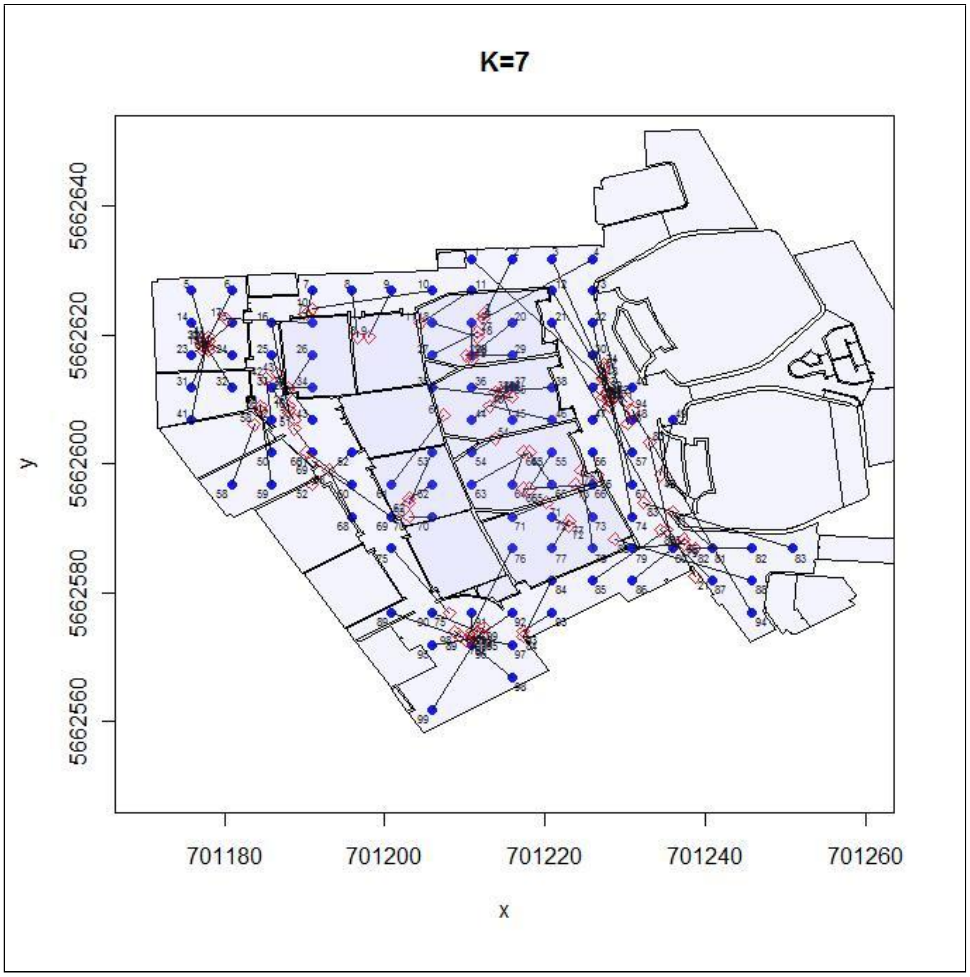


Figure 4.7: Results of KNN method, K=7

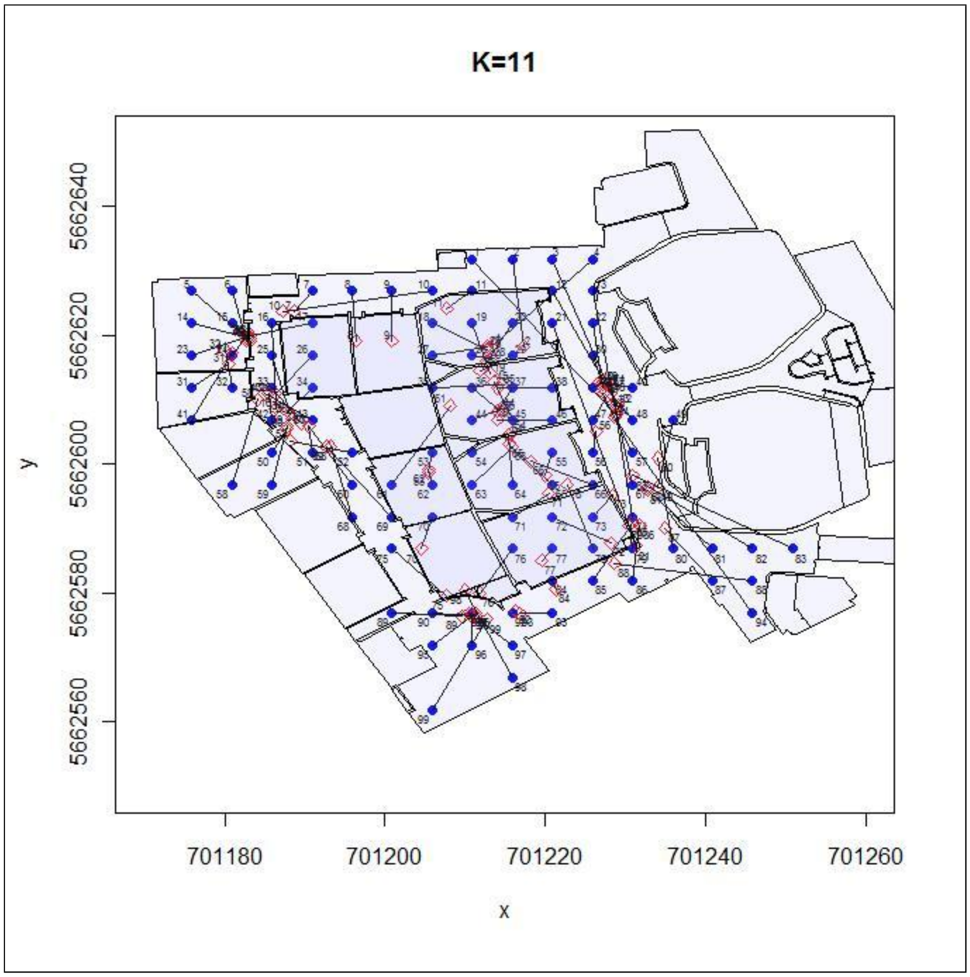


Figure 4.8: Results of KNN method, K=11

This method is very dependent on the granularity and number of calibration points. As it is obvious from the figures, as K increases, more calibration points are engaged which may increase precision, but decrease accuracy. Figure 4.10 depicts a plot of RMSE with respect to K. RMSE values for K=1, 3, 7, 11 are respectively 12.78m, 11.63m, 11.37m, and 11.28m. According to the quadratic curve fit to the RMSE data for K=1 to K=16, K=9 gives the best result for which the RMSE is 11.40m. A one-way ANOVA test was used to test for error differences between KNN neighbourhoods of 1 to 16 calibration points. Error derived from the sixteen neigbourhoods did not significantly differ, $F(15,1568)=0.188$, $MSE=47.786$, $p=1.000$. This result was confirmed by pairwise comparison using Fisher's Least Significant Difference (LSD) test as all group means differ by less than 3.475 m, the minimum mean difference needed to differentiate a significant difference between means. However, K=2 gave the best results with a mean error of 9.007 m (se=0.756 m). Figure 4.9 depicts the results from the LSD test in the form of a bar graph showing that all neighbourhoods belong to the same group with respect to location accuracy.

The KNN algorithm requires knowledge of the position of calibration points in order to function; so this method is only appropriate for use with geometrical spatial model and is not compatible with other spatial models (e.g symbolic models). Discussion of the advantages, disadvantages, and effects of this method on precision and accuracy is left for the next chapter.

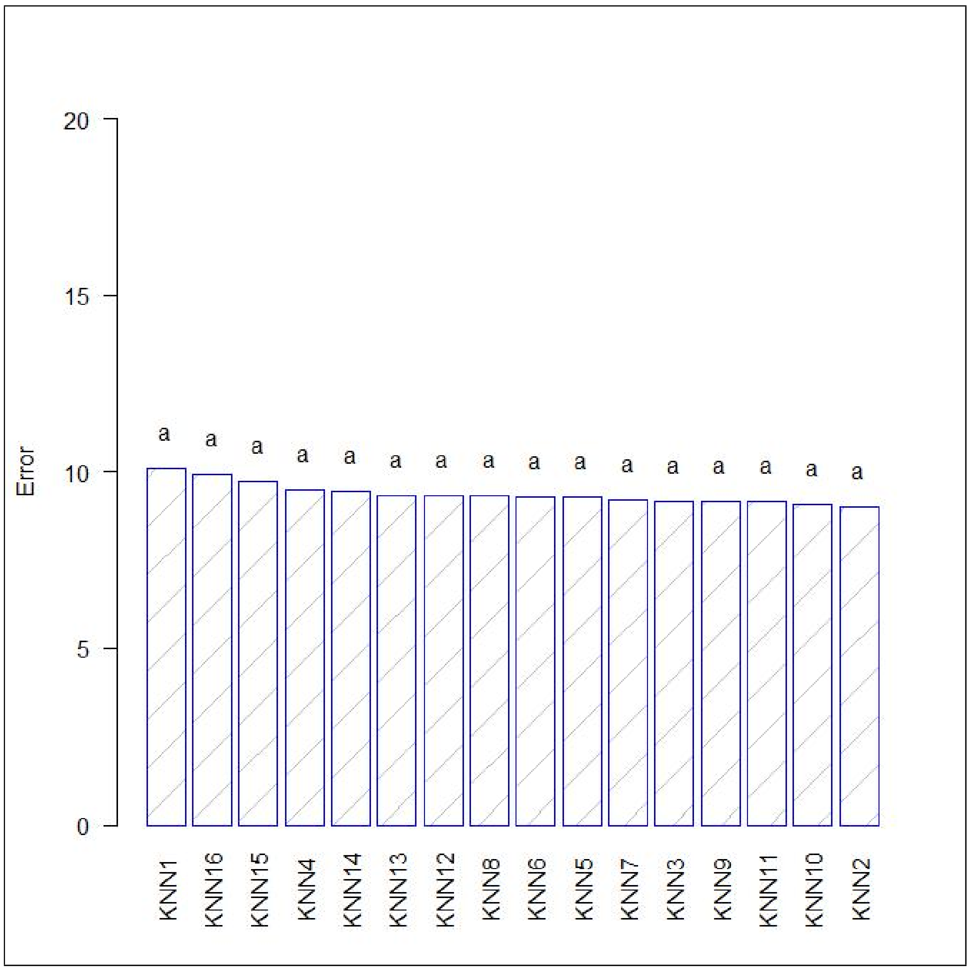


Figure 4.9: Result of ANOVA test for KNN suggests no significant difference among K from 1 to 16.

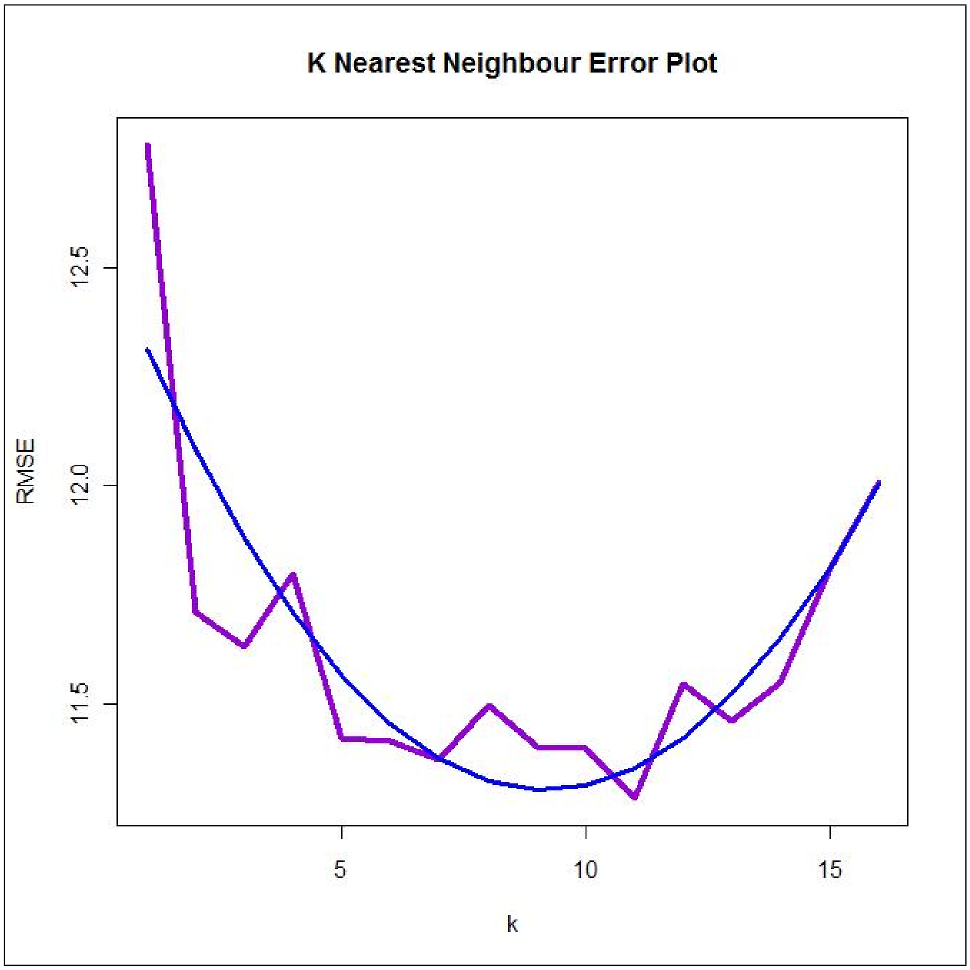


Figure 4.10: RMSE with respect to K, the number of neighbours used. (Violet line shows the real RMSE and blue line shows a quadratic fitted to RMSE)

Support Vector Machines: To test the SVM method the R package "e1071" was utilized. "e1071" is designed to estimate a solution via classification, regression, or SVM. Using this package, four different kernels can be tested with SVM. They include: linear, polynomial, radial basis, and sigmoid kernels. In practice, all calibration points are fed into SVM. Through the cross validation process leave one out, each point was removed from the solution once and SVM was trained with the other points; then the position of that single point was predicted. Figures 4.11 to figure 4.14 provides the results of positioning with SVM utilizing different kernels.

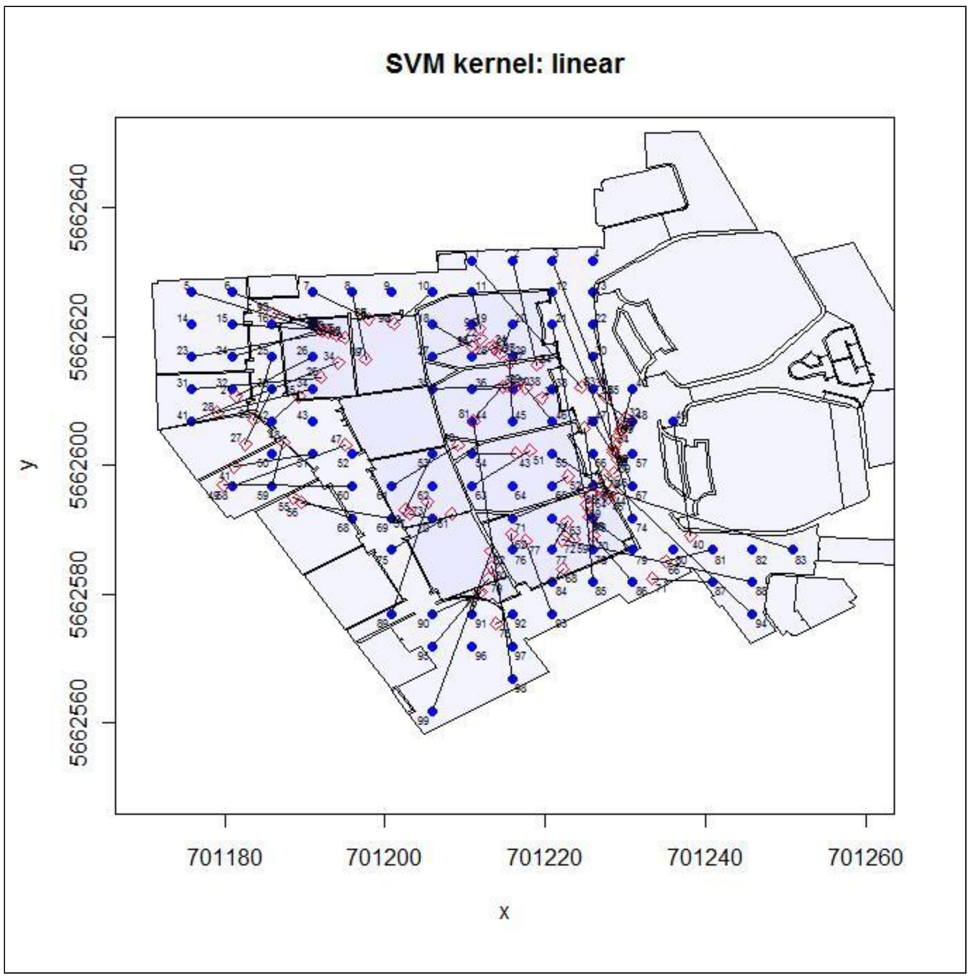


Figure 4.11: Results of SVM method, kernel:linear

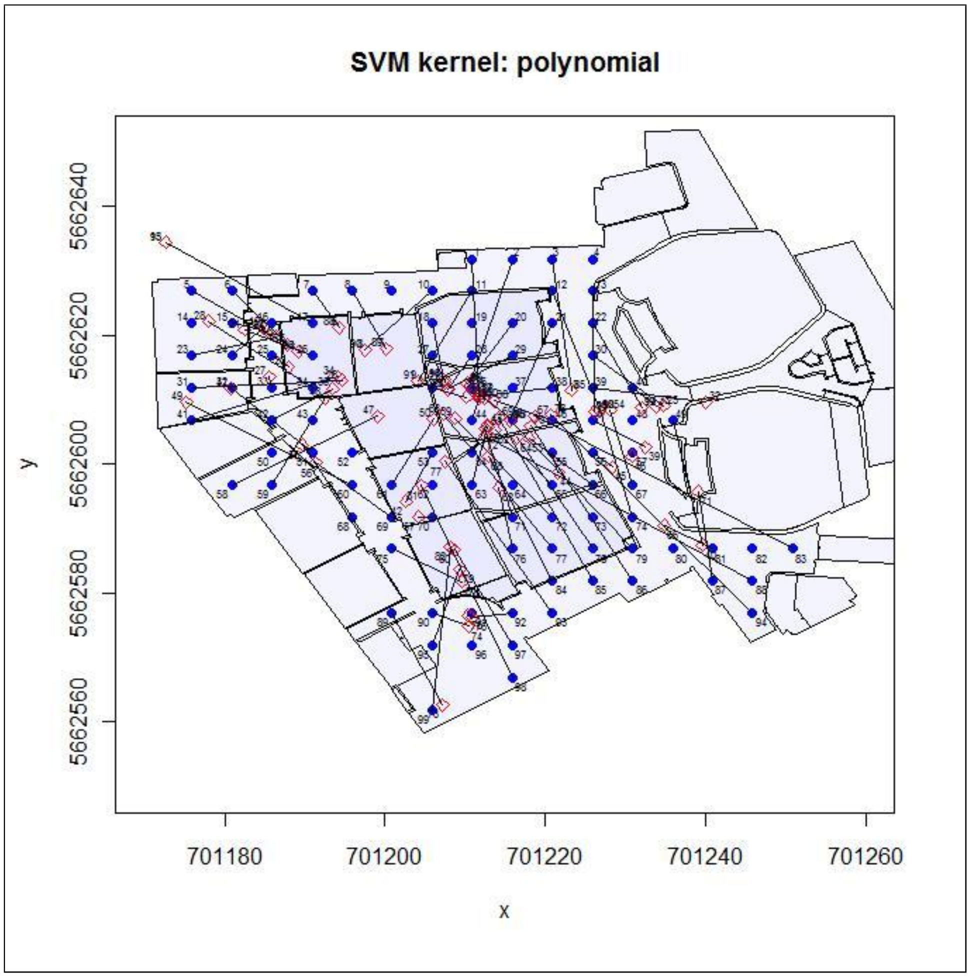


Figure 4.12: Results of SVM method, kernel:polynomial

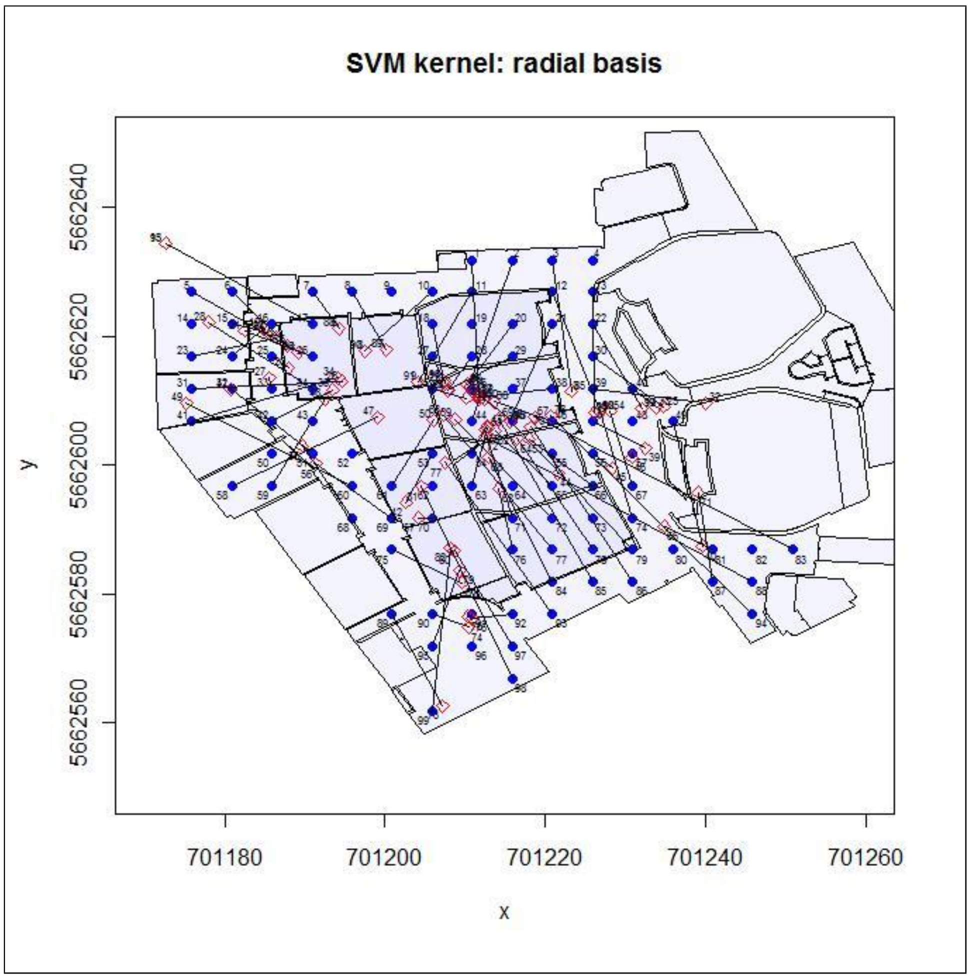


Figure 4.13: Results of SVM method, kernel:radial basis

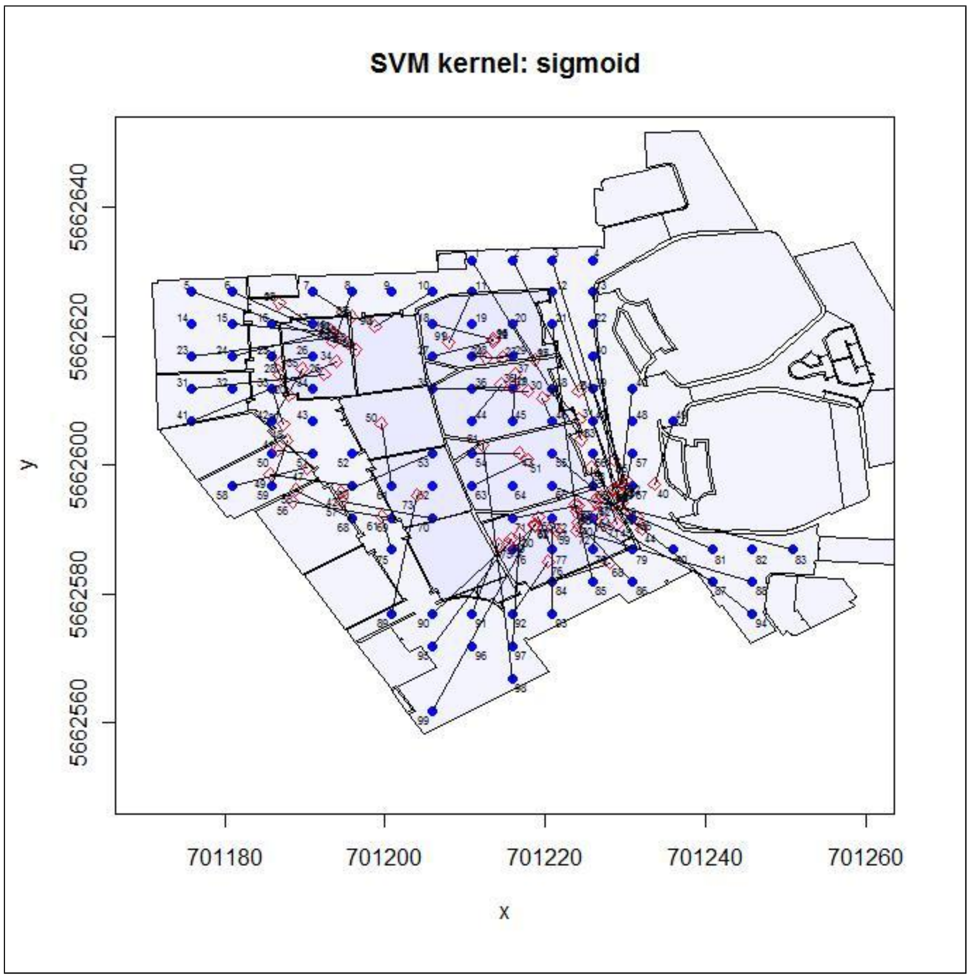


Figure 4.14: Results of SVM method, kernel:sigmoid

SVM does not give better precision in comparison to KNN method, since the best precision from SVM was 13.05m and worst precision from KNN was 12.78m for K=1. However, it can work with symbolic (non-geometric) spatial data models. In other words, SVM deals with positioning problem efficiently if instead of coordinates (x, y) for the calibration points, we use the location of points (e.g. room name, location ID, unique code). The RMSEs for linear, polynomial, radial basis, and sigmoid kernels were respectively 13.50m, 16.31m, 16.31m, and 15.03m. A one-way ANOVA was used to test for error differences between SVM kernels. Error derived from the four kernels did not significantly differ, F(3,324)=1.651, MSE=70.772, p=0.178. This result was confirmed by pairwise comparison using Fisher's LSD test as all group means differed by less than 3.488 m. Figure 4.15 presents the graphical results from the LSD test in the form of a bar graph showing that all kernels belong to the same group with respect to location accuracy. The SVM linear kernel gave the best results with a mean error of 11.212 m (se=0.836m). Again advantages and disadvantages of this method will be described in the next chapter.

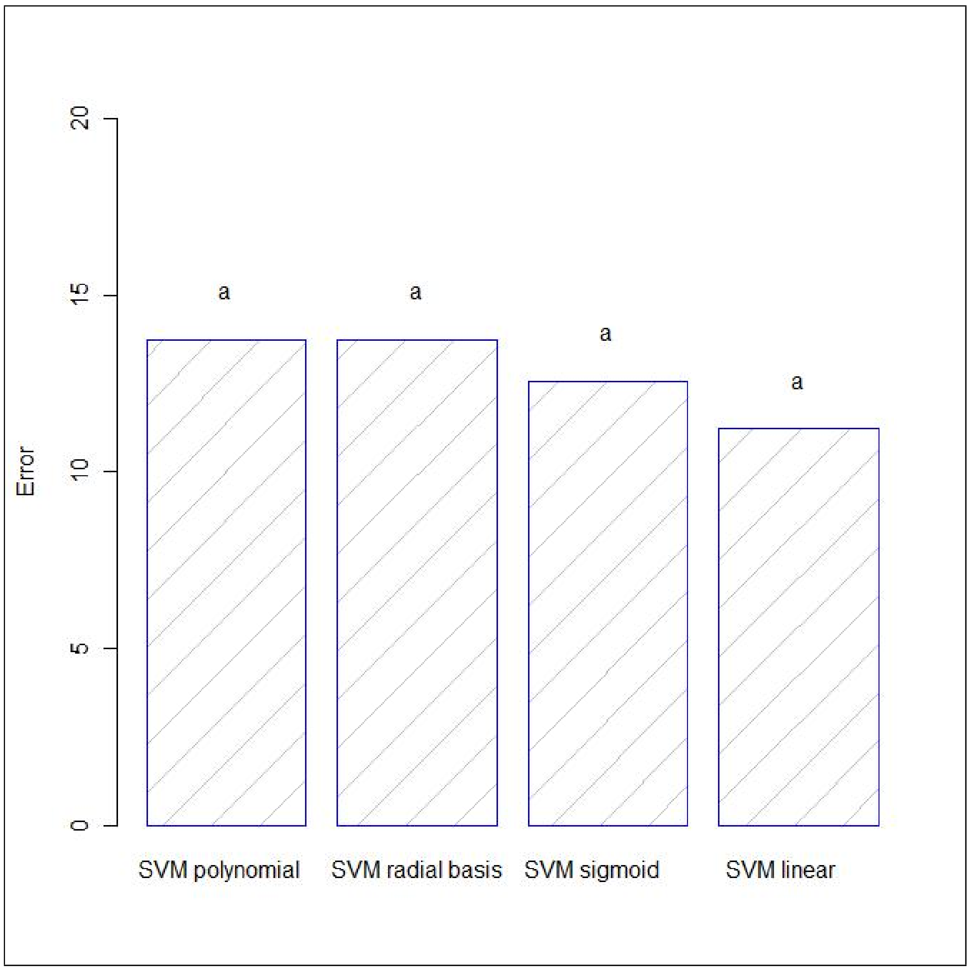


Figure 4.15: Result of ANOVA test for SVM suggest no significant difference among the different kernels

Artificial Neural Network: ANN can be described as a model for mapping signal strength data to position data [47]. Input layer consists of RSSI and output layer consists of position of calibration data. The hidden layers within ANN assign a new weight to synapses during each iteration until results converge. Through this process and based on the number of iterations, each synapse may take different weights. As such, multiple attempts to analyse data using ANN may give different results.

To test the ANN method the R package "neuralnet" was employed. Cross validation was used during the training process. Each point was taken out in a loop and the other points were used to train the network; then the single point was introduced to system for prediction. Figure 4.16 shows the result of running ANN with our calibration data. The main feature of this method was the time it took for training and prediction.

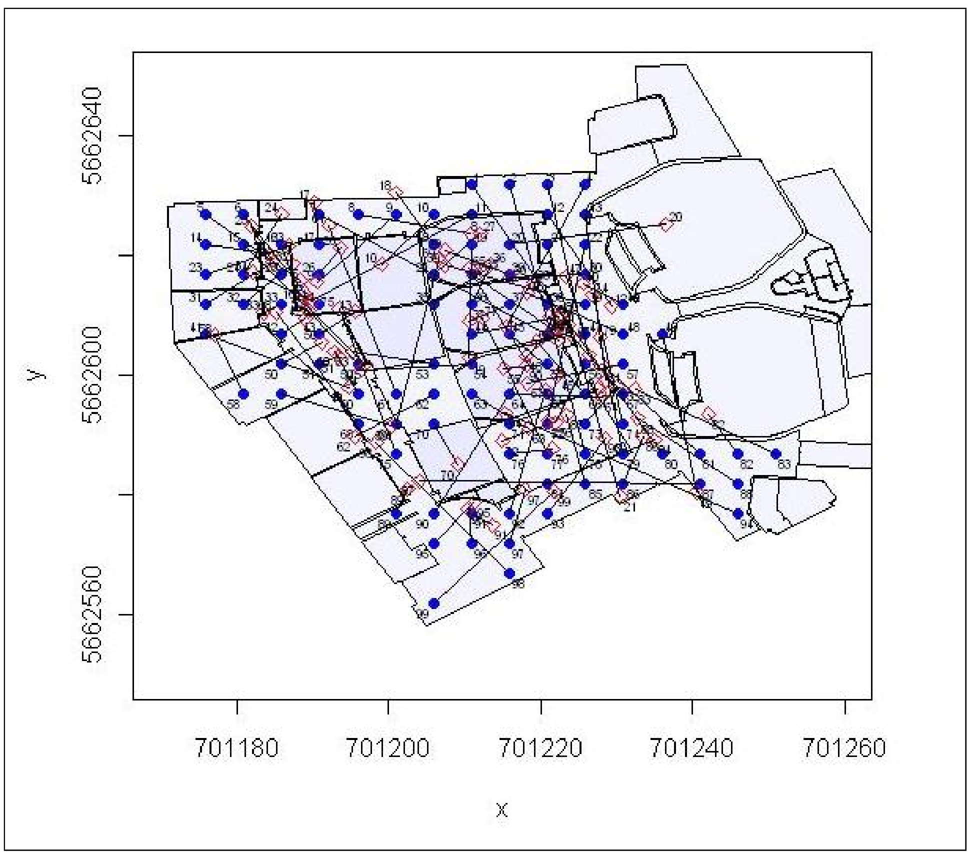


Figure 4.16: Results of ANN method with 16 hidden neurons

The main parameter that affects the precision of results in ANN is the number of neurons in the hidden layer. A one-way ANOVA was used to test for error differences between ANNs with varying numbers of neurons, 4, 8, 14, 16, and 32. Error derived from the ANN with differing numbers of neurons significantly differed, $F(4,265)=11.136$, $MSE=186.995$, $p=2.302e\text{-}08$. Pairwise comparison using Fisher's LSD (minimum mean difference = 7.450 m) revealed that neurons of 4, 8, and 14 produced significantly lower errors than neurons of 16 and 32. test as all group means differed by less than 7.450 m. Fourteen neurons in the hidden layer gave the best results with a mean error of 16.351m (se=1.364m). Over 5 implementations of ANN with 4, 8, 14, 16, and 32 neurons in their hidden layer Figure 4.17 shows that 4, 8, and 14 neurons (labeled [group] b on the plot) in the hidden layer result in similar location accuracies, as do 16 and 32 neurons (labeled [group] a), but their errors (group a) are significantly greater.

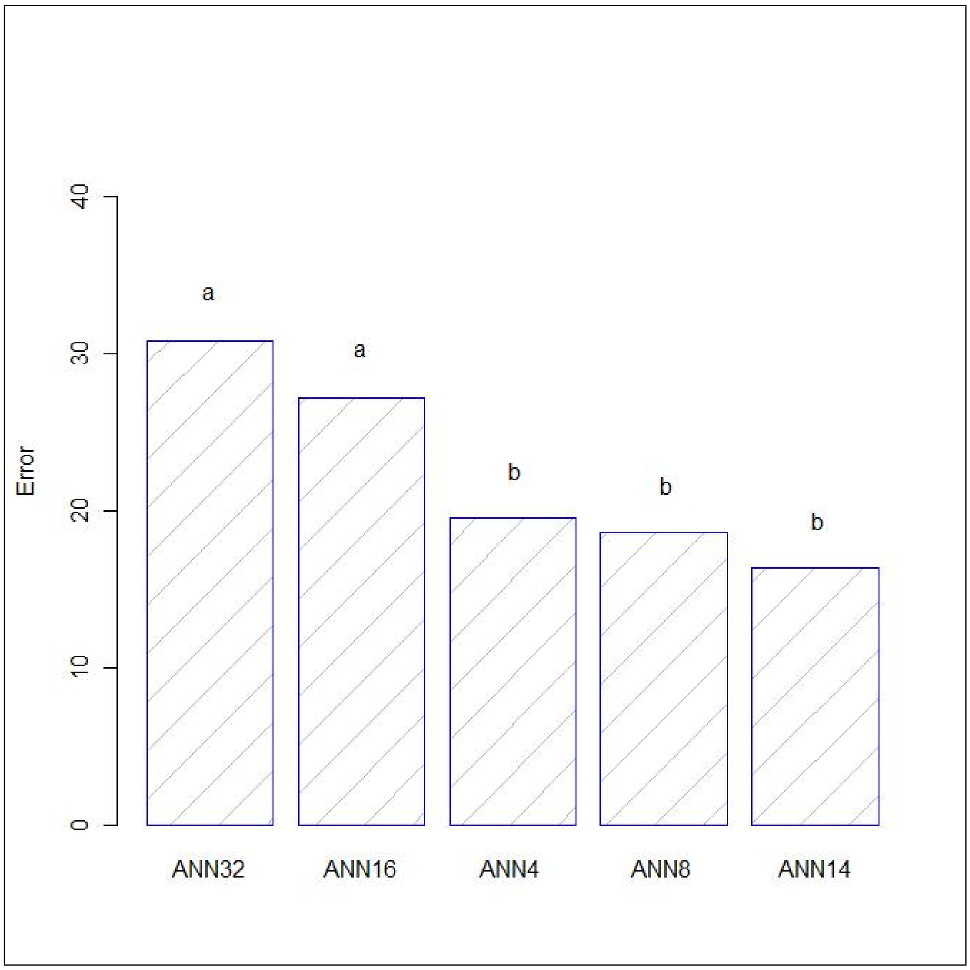


Figure 4.17: Results of ANOVA test on ANN errors

4.3.2 Statistical Distances

KNN, SVM, and ANN location fingerprinting methods are based on measuring the signal strength values. These values are usually the average signal strengths received at a point. None of these methods consider signal distribution (i.e. standard deviation and mean value of signal strength). Statistical distances attempt to find similar signal distributions assuming that signals in certain locations show the same distribution patterns.

In statistics, a statistical distance quantifies the distance between two statistical objects (e.g. two samples, two random variables, or two probability distributions). In other words, a statistical distance can measure similarity between two distribution patterns. Bhattacharyya and Mahalanobis distances are two statistical distances investigated in this research.

The Mahalanobis distance measures a distance based on correlation between two set of variables. This distance can identify patterns in the set of variables. Generally, the Mahalnobis distance between multivariate vectors $x = (x_1, x_2, \ldots, x_n)^T$ and $y = (y_1, y_2, \ldots, y_n)^T$ with covariance matrix S is defined as [50]:

$$D_M = \sqrt{(x-y)^T S^{-1} (x-y)} \tag{4.1}$$

In other words, Pythagorean theorem assumes that x and y are uncorrelated and that the variances are the same. But when the distributions of the two vectors differ, the covariance matrix must be considered in the model [51].

In the case of location fingerprinting, vectors x and y can be replaced by vectors of signal strengths from the calibration set and prediction set. In order to estimate the covariance matrix we need to calculate the standard deviation of signal strengths at both calibration and predication time. Figure 4.18 and figure 4.19 demonstrate two results using Mahalanobis distance for location fingerprinting. The first figure is the result of using signal strengths which are greater than -55dBm and second figure is the result of using signal strengths greater than -60dBm. The first model resulted in an RMSE=19.26m and the second an RMSE=16.66m.

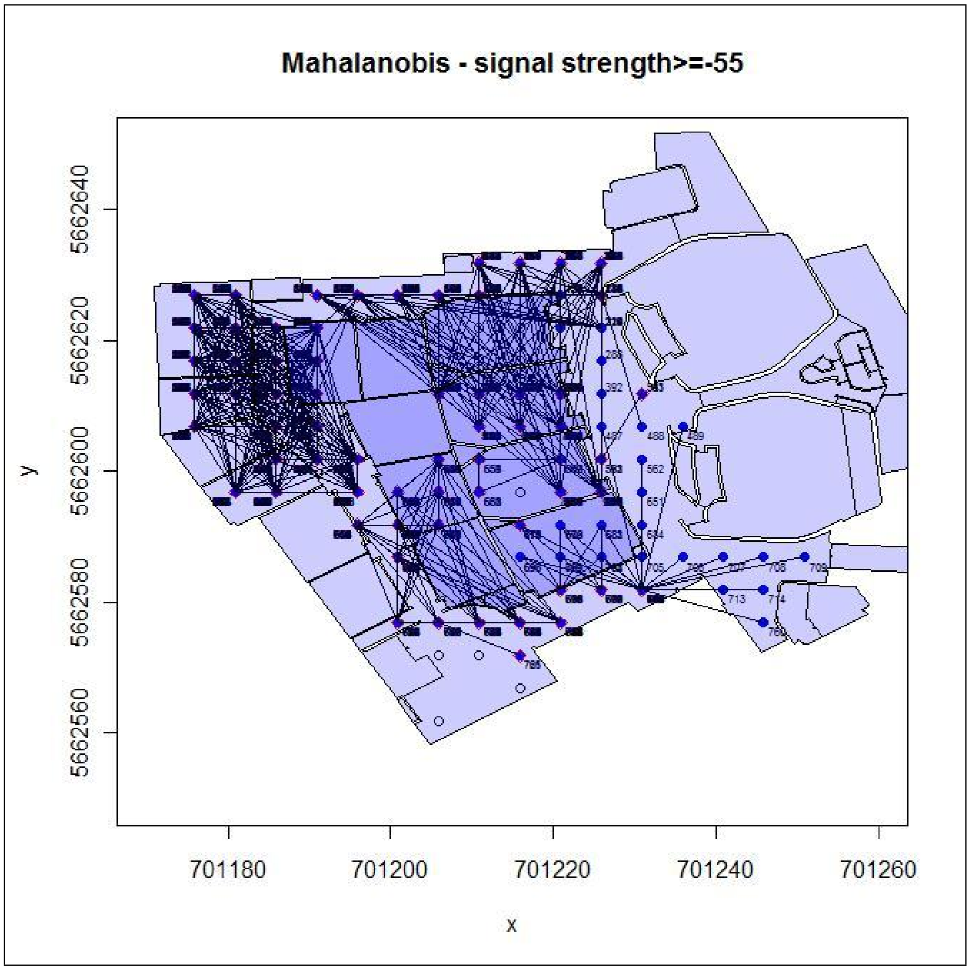


Figure 4.18: Results of Mahalanobis distance method

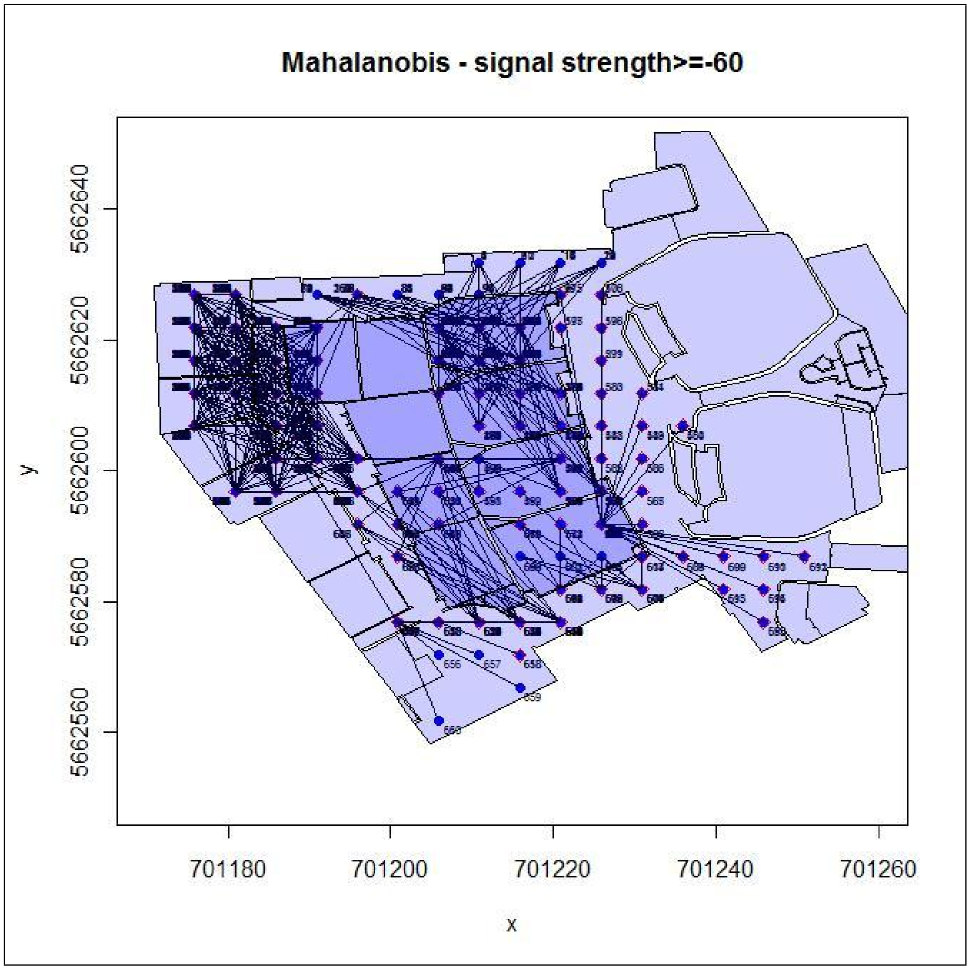


Figure 4.19: Results of Mahalanobis distance method

The Bhattacharyya distance also measures the similarity of two discrete or continuous probability distributions. For two multivariate Gaussian distributions $p_1 = N(m_1, P_1)$ and $p_2 = N(m_2, P_2)$, the Bhattacharyya distance can be defined as [52]:

$$D_B = \frac{1}{8}(m_1 - m_2)^T P^{-1}(m_1 - m_2) + \frac{1}{2}ln(\frac{detP}{\sqrt{detP_1 detP_2}}) \tag{4.2}$$

where

$$P = \frac{P_1 + P_2}{2} \tag{4.3}$$

Again, p_1 and p_2 distributions can be replaced by signal strength distributions obtained during calibration and prediction. Figure 4.20 and figure 4.21 illustrate the results of location fingerprinting using the Bhattacharyya distance. The first figure takes into account signals with average signal strength greater than or equal to -55[dBm], and second figure considers signal strengths greater than or equal to -60[dBm]. Their RMSE values are 19.34m and 16.49m respectively. A one-way ANOVA was used to test for error differences between the Bhattacharyya and Mahalanobis methods for signal strengths greater than -55[dBm] and -60[dBm]. Error derived from the statistical distances differed significantly, F(3,2490)=15.718, MSE=55.822, p=3.98e-10. Pairwise comparison using Fisher's LSD (minimum mean difference = 1.12 m) revealed that the Bhattacharyya method using signals greater than -55[dBm] produced significantly greater errors than the other tests. The Bhattacharyya method using signals greater than -60[dBm] gave the best results with a mean error of 14.845m (se=0.288m). Figure 4.22 shows that Bhattacharyya distance for signal strengths greater than -60[dBm] and Mohalnabis distances greater than -55[dBm] and -60[dBm] gave similar location accuracies (labeled [group] b), and that the Bhattacharyya distance for signals greater than -55[dBm] (labeled [group] a), are significantly greater.

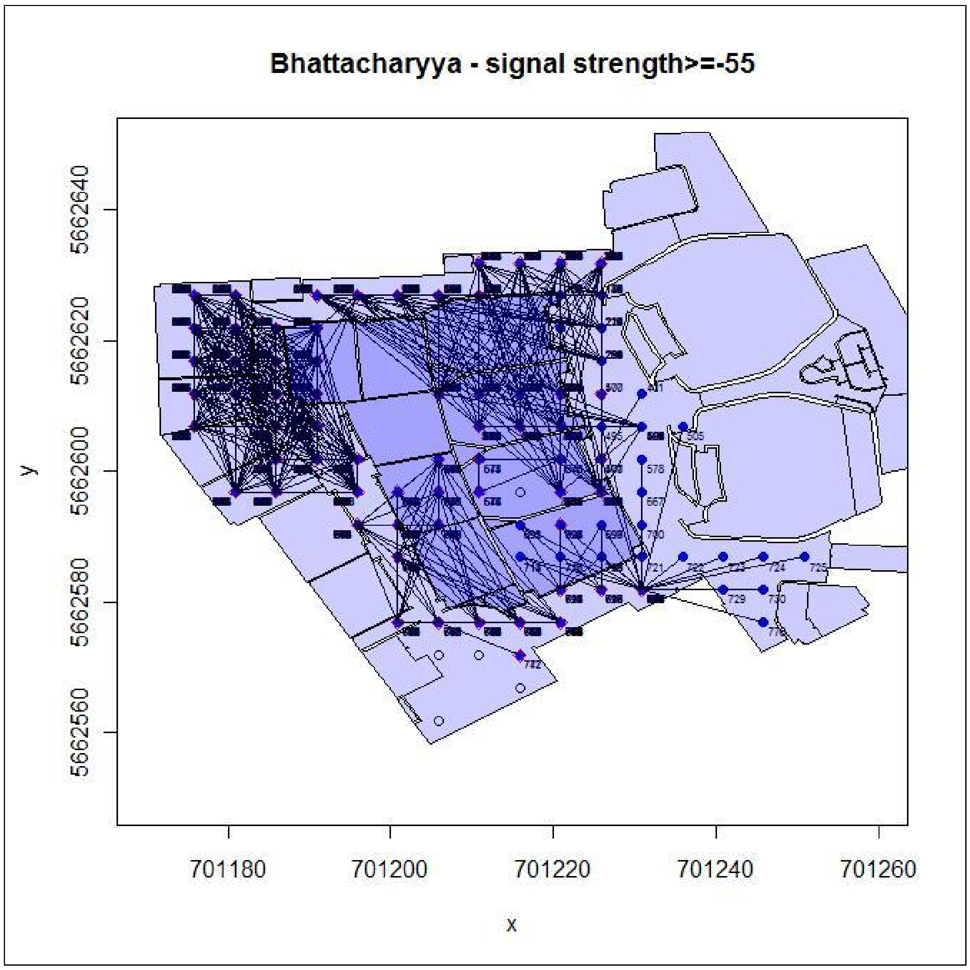


Figure 4.20: Results of Bhattacharyya distance method

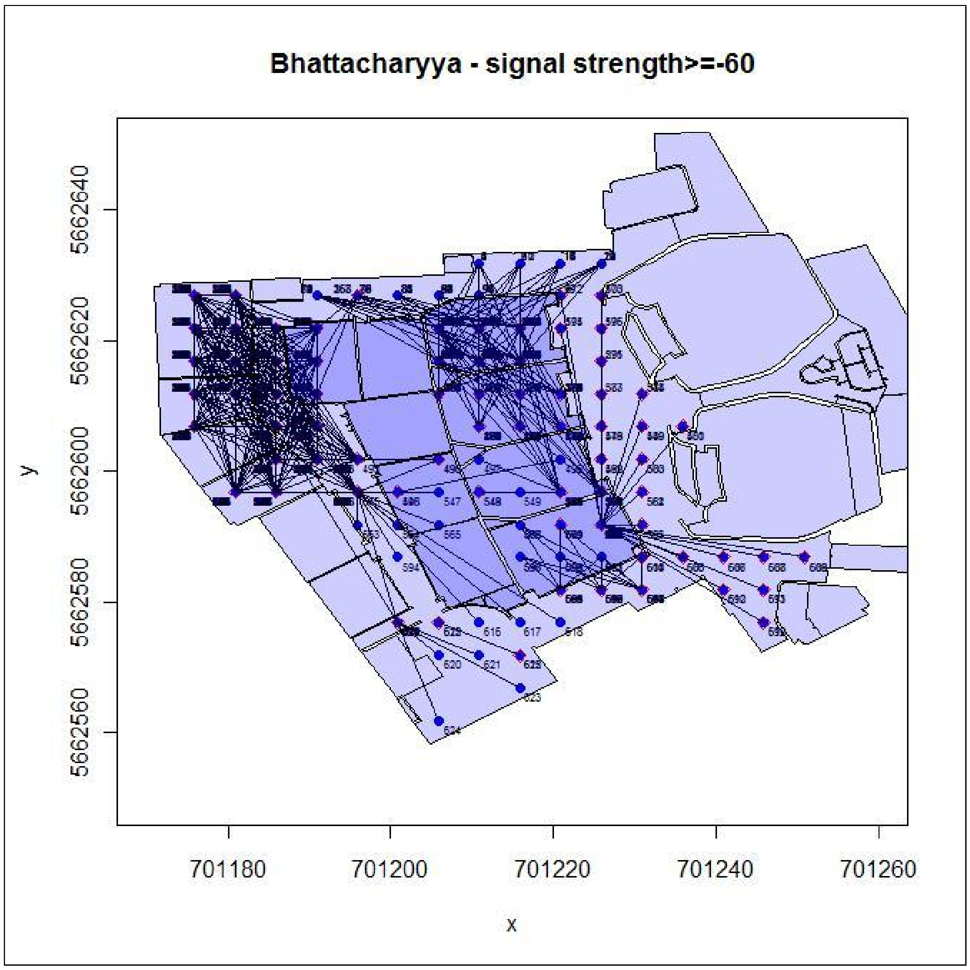


Figure 4.21: Results of Bhattacharyya distance method

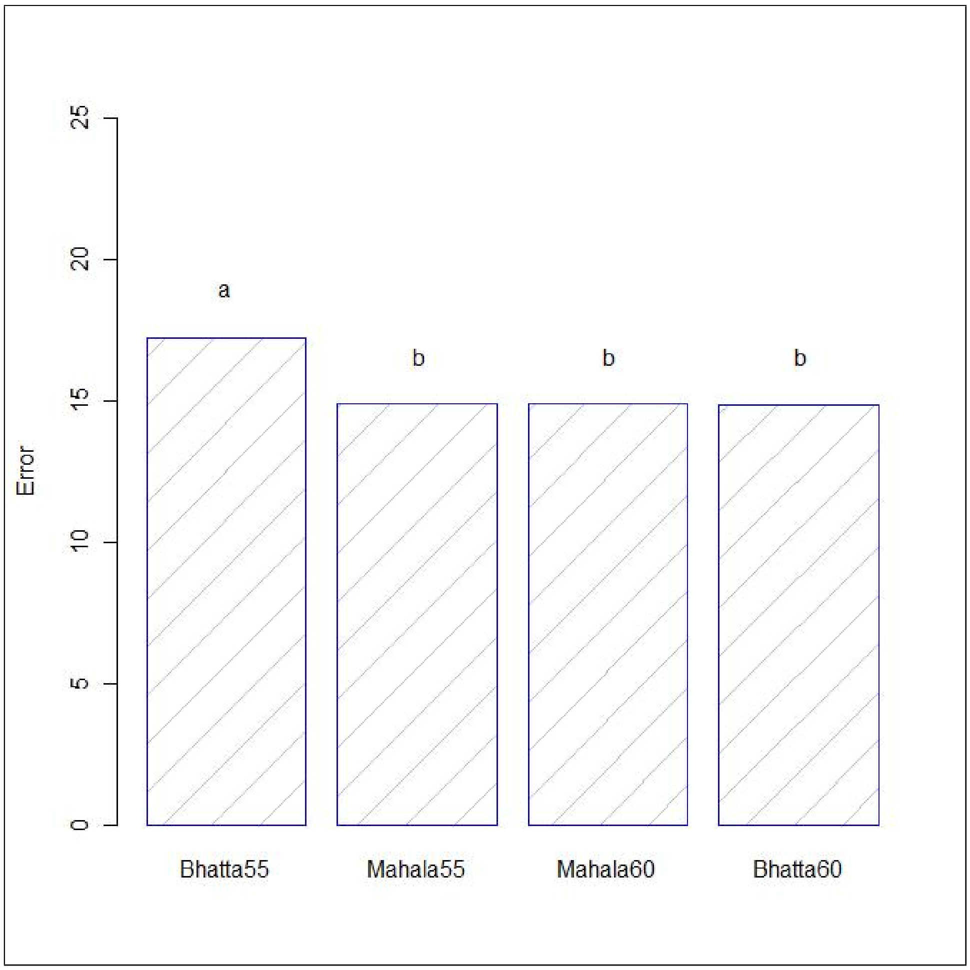


Figure 4.22: Results of ANOVA test on statistical distances

The idea behind using statistical distances is to reduce the number of calibration points. One of the main disadvantages of location fingerprinting for indoor positioning is the cost and time required for collecting calibration data. Statistical distances show that points in the same areas have similar patterns, which means we can replace a set of points with same patterns with just one representative location.

4.4 Methodology for Path Finding

Roads, trails and other movement facilities for outdoor environments are explicitly built as transportation networks for different types of commuters. Paths in indoor environments are not as explicit as those outdoors [53]. So, the concept of connections and movement paths for indoor areas are derived from human cognition maps. In other words, the paths that people use to commute in different locations can be used to model the movement network. For this purpose, a typical approach is to construct a large graph that covers the whole area with a fine network of possible paths [53]. Path finding solutions can then be founded on this graph or mesh surface.

Human navigation and path finding is often conceived as a suboptimal system, since it is not based on mathematical solutions. Human navigation has some critical characteristics [19]:

1. A human's navigation system adapts to the environment in which navigation occurs.
2. Navigation follows body motion and self perception of motion over time.
3. The mental image of the environment affects the nature, type, speed and direction of motion.
4. Routes are considered as long or short based on the process towards or away from the destination.

5. Paths are considered to be non-symmetric [19].

In LBS applications, path finding refers to algorithms that provide a path (or a set of paths) from an origin to a destination based on some criteria or constraint. Shortest (distance) path solutions are the well-known solutions among path finding algorithms. The next sections provide details of shortest path algorithms and elaborate the difference between these algorithms.

4.4.1 Shortest Path Problem

A best path through a network from an origin to a destination is a path having the least value/weight [31]. Value/weight refers to any metric that describes the effort to traverse the path: length, time, cost, or roughness of the path. These metrics can be assigned to the graph which describes the path network in an environment. However, as mentioned, the best path is the one that minimizes the total weight/value of the path. For some applications distance is the most important characteristic of a path and can be extracted directly from geometry of the network. However, the meaning of distance can vary. For many applications distance means Euclidean distance, for some others distance means network distance (e.g. applications looking for nearest point of interest), while some applications accept distance as energy consumption through a path (e.g. robot controllers) [21].
Hu and Lee [21] state that distance between locations in an indoor environment is the summation of their path distances. However, distance should satisfy the following conditions:

1. bounded: $\forall$ location A and B, $distance(A, B) < \infty$;
2. zero-flexibility: $\forall$ location A, $distance(A, A) = 0$;
3. transitive inequality: $\forall$ location A, B, C,

distance(A,B)+distance(B,C)¿=distance(A,C)$; symmetryisnotaconditionofdistance; inotherwords,$

Since paths in complex indoor environments are implicit it is necessary to construct a detailed map of paths which connects different places together [53]. This type of map can be compared with road maps for outdoor areas with respect to network analysis. Dijkstra's algorithm is one of the best known algorithms to find the best/shortest path. This algorithm works on a weighted graph in which the edges weights are non-negative. Most of the applications that try to find the best/shortest path rely on this algorithm.
Dijkstra's algorithm: In this algorithm, distances are represented as a weighting function w from the set of edges E consisting of positive reals $\mathbb{R}^+$. An additional target weighting function $t : N \rightarrow \mathbb{R}^+$ is used to store the minimum distances from the start node s to each node in the graph. Dijkstra's algorithm initializes the target weights t to infinity, except for the starting node $t(s) \leftarrow 0$, and the nodes adjacent to s. The algorithm then proceeds to traverse the entire graph from the start node, at each step sorting the unvisited nodes $N \backslash V$ into ascending order of target weight and recalculating the minimum target weights t [54].
Algorithm 1 describes details of Dijkstra's algorithm and Figure 4.23 to figure 4.30 depicts an example of finding shortest paths between two nodes in a network [54].

Input: Undirected simple connected graph $G = (N, E)$, starting node $s \in N$, weighting function $w : E \rightarrow \mathbb{R}^+$, target weighting function $t : N \rightarrow \mathbb{R}^+$

initialize $t(n) \leftarrow \infty, \forall n \in N$, visited node set $V \leftarrow s$

set $t(s) \leftarrow 0$

for all $m \in N$ such that edge $sn \in E$ **do**

set $t(n) \leftarrow w(sn)$

While $N \neq V$ **do**

find, by sorting, $n \in N \backslash V$ such that $t(n)$ is minimized

add n to V

for all $m \in N \backslash V$ such that edge $nm \in E$ **do**

$t(m) \leftarrow min(t(m), t(n) + w(nm))$

Output: Graph weights $t : N \rightarrow \mathbb{R}^+$

Algorithm 1: Dijkstra's Algorithm [54]

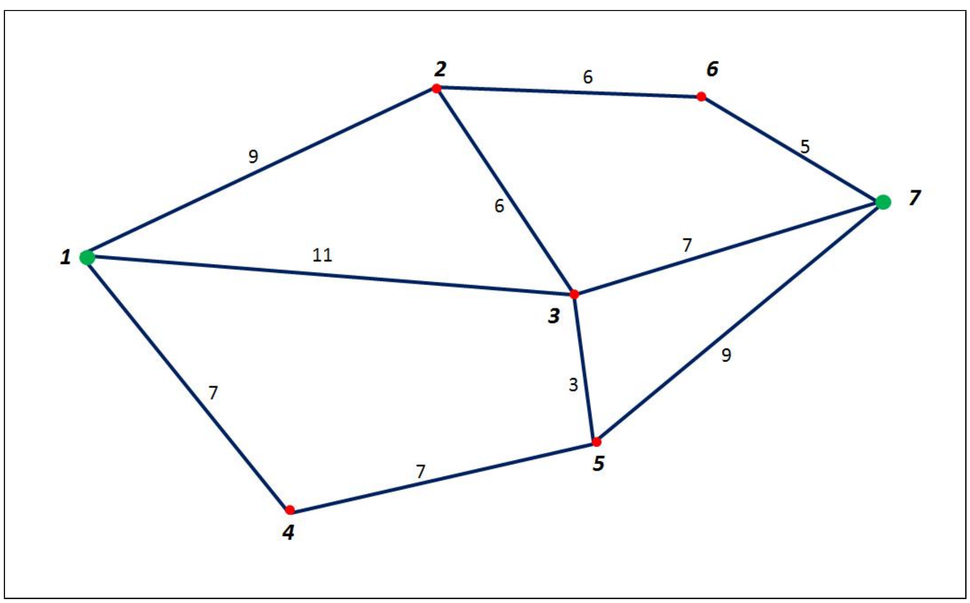
4. 

Figure 4.23: Dijkstra's algorithm to find shortest path between nodes 1 and 7.

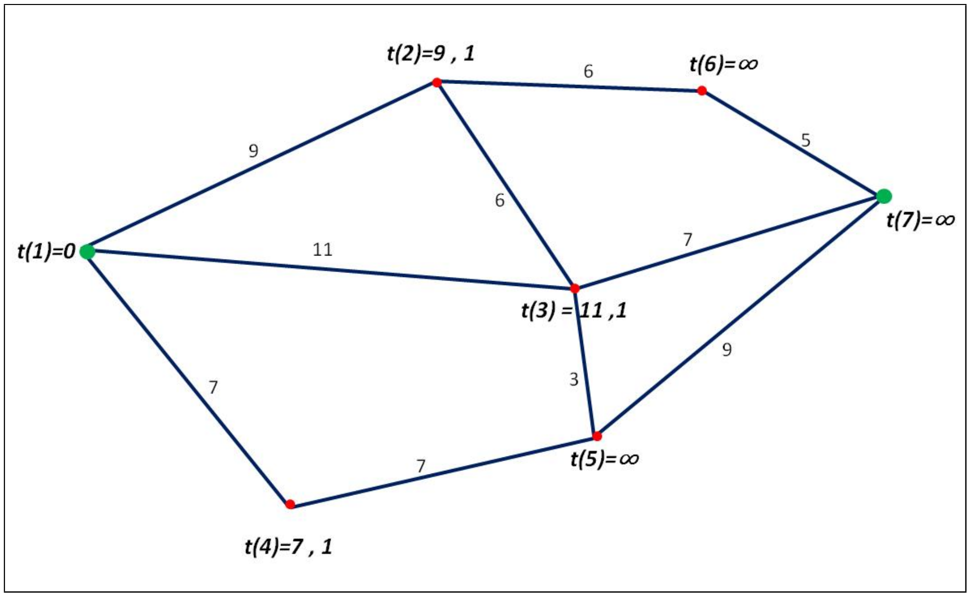


Figure 4.24: Dijkstra's algorithm: 1^{st} iteration. (t(node)=value , previous node)

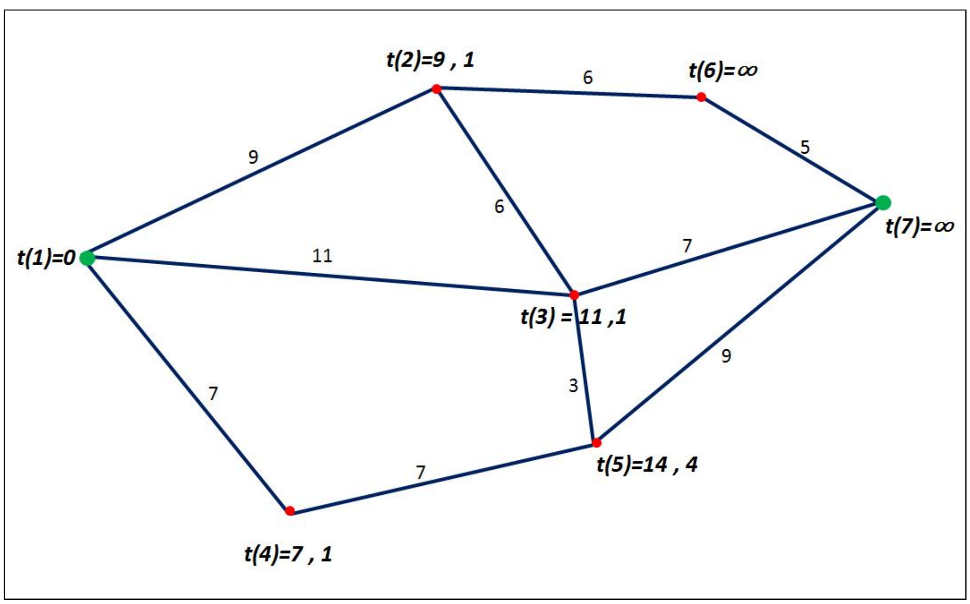


Figure 4.25: Dijkstra's algorithm: 2^{nd} iteration. (t(node)=value , previous node)

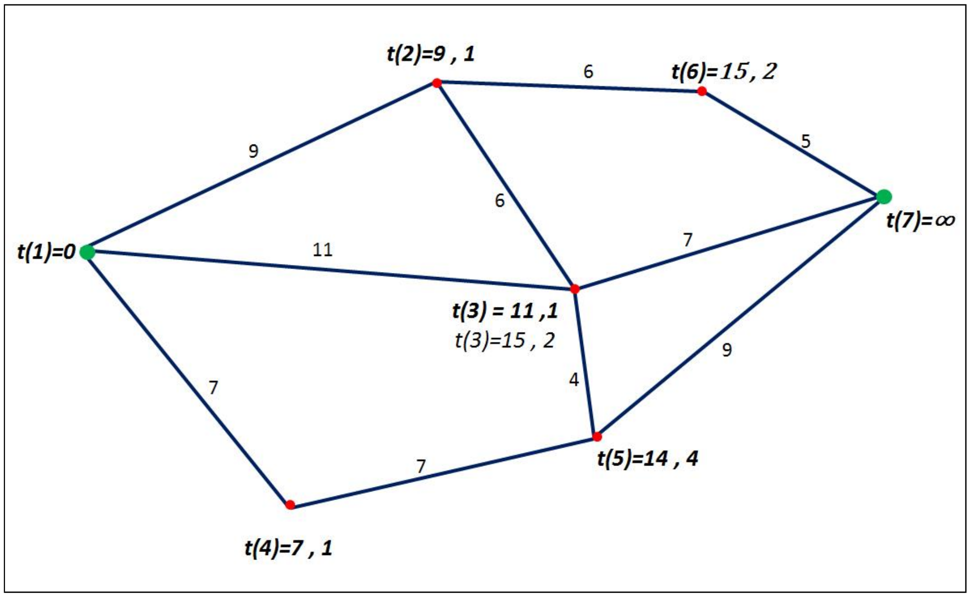


Figure 4.26: Dijkstra's algorithm: 3^{rd} iteration. (t(node)=value , previous node)

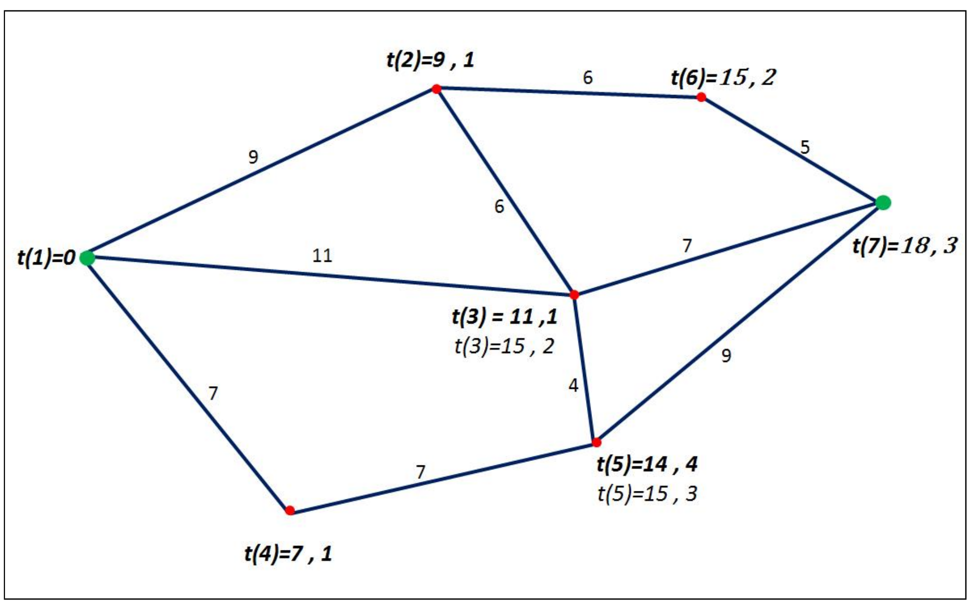


Figure 4.27: Dijkstra's algorithm: 4^{th} iteration. (t(node)=value , previous node)

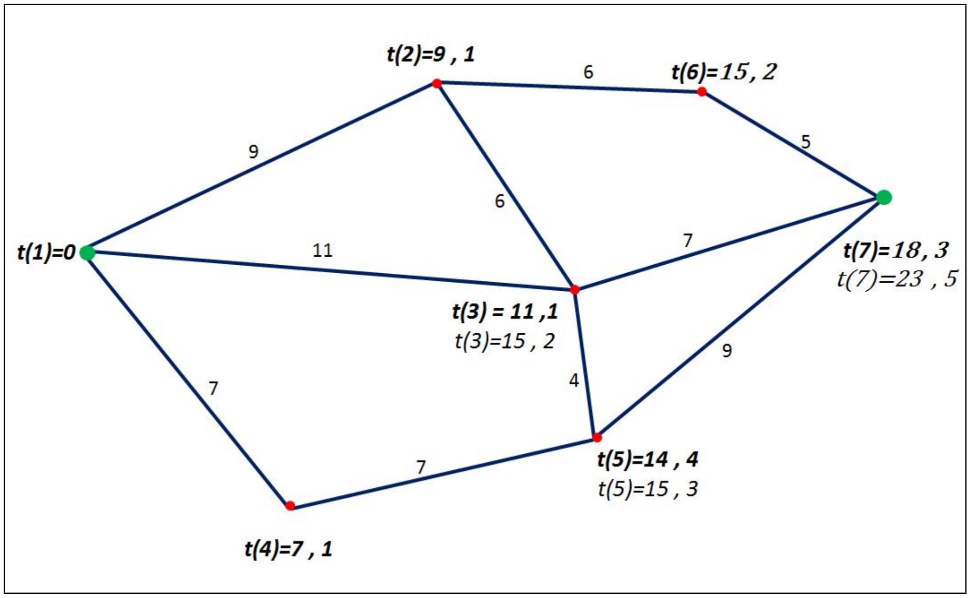


Figure 4.28: Dijkstra's algorithm: 5^{th} iteration. (t(node)=value , previous node)

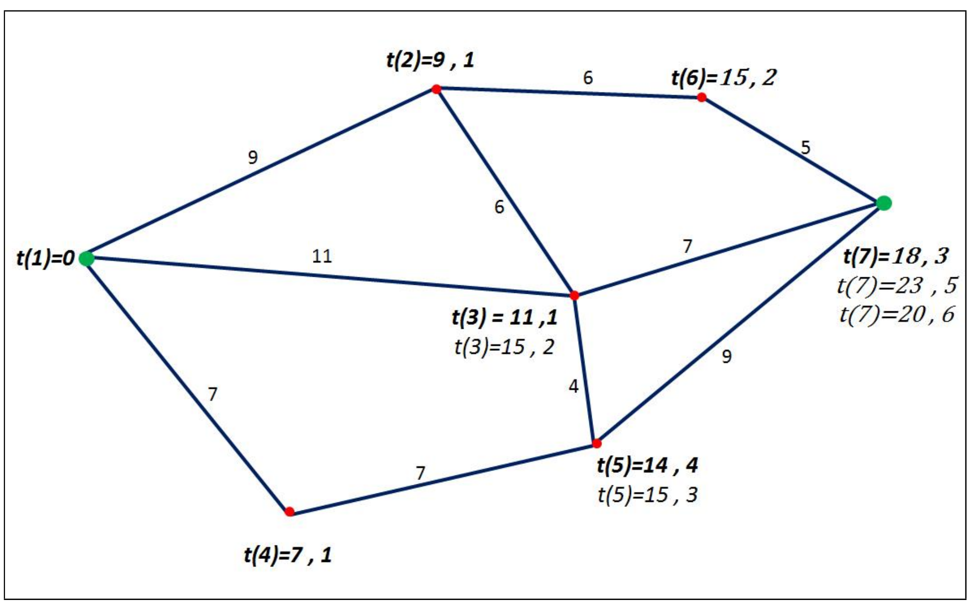


Figure 4.29: Dijkstra's algorithm: 6^{th} iteration. (t(node)=value , previous node)

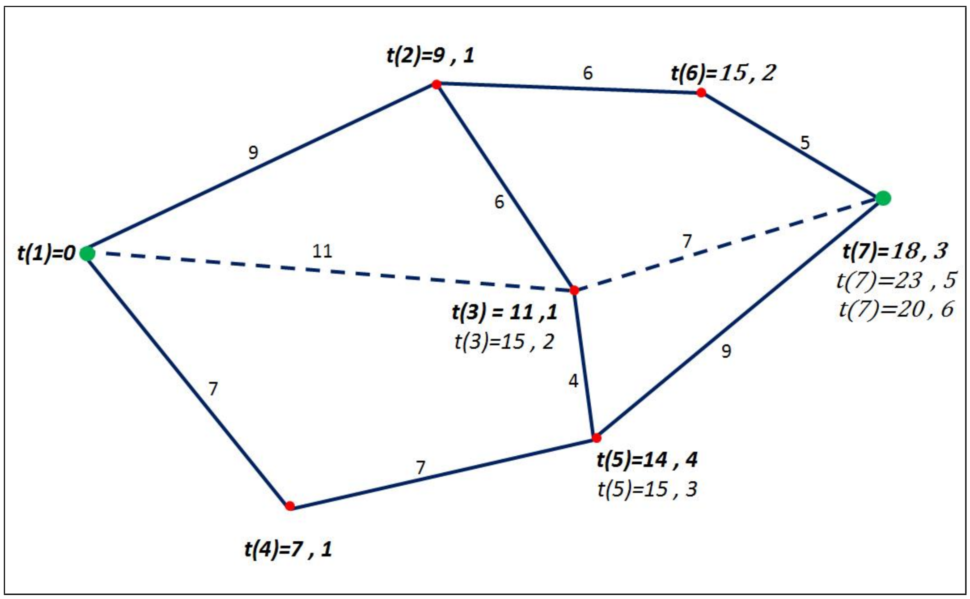


Figure 4.30: Dijkstra's algorithm: shortest path between nodes 1 and 7. (t(node)=value , previous node)

Dijkstra's algorithm is able to calculate all shortest paths from a single node to all other nodes in a network, which makes it a single source shortest path algorithm.

4.4.2 K Shortest Paths Problem

Sometimes it is desirable to know the second (third, etc.) shortest path in a network in addition to the shortest path [55]. There may be different reasons behind finding the second (third, etc.) shortest path. Some applications require alternative routes to the shortest path for emergency reasons. Other applications may provide alternative paths that are not that different from the shortest path, but offer different characteristics. Also understanding the alternative paths from one node to another provides flexibility for applications. For example, some applications may incorporate other constraints (e.g. weather, indoor paths vs. outdoor paths, visiting specific areas or points, etc.) that divert the route from the shortest distance. In these situations finding more than one shortest path is a significant task.

For indoor LBS applications, providing a number of paths is of benefit because [19], people prefer to use a wide range of criteria for path finding in indoor environments. These criteria may include: shortest distance, quickest distance, fewest turns, most scenic/aesthetic, first notices segment, longest leg first, many curves, many turns, different paths from previous ones and shortest leg first [19]. To satisfy some of these conditions we may come up with alternatives for the shortest path.

The problem of finding more than one shortest path is defined as the K shortest path problem. By definition, a solution to the K shortest paths problem returns a set of K shortest paths between two locations given a particular weighting function [31]. Solution to the K shortest paths are based on minimum trees. A minimum tree is the basic to the computation of the shortest paths which was developed by Moore and Dantzig [31]. In short, a minimum tree is a subset of the links in a network that define minimum paths from a specific node to all other nodes in the network [31]. Dijkstra's algorithm is also considered to be a minimum tree case.

Dijkstra's algorithm finds the first shortest path. The second shortest path is the one with the smallest diversion from the first shortest path that can be extracted. The next shortest path can be found using the same technique used for second shortest path. Figure 4.31 to figure 4.33 illustrate 1^{st}, 2^{nd} and 3^{rd} shortest paths on the graph of Figure 4.23.

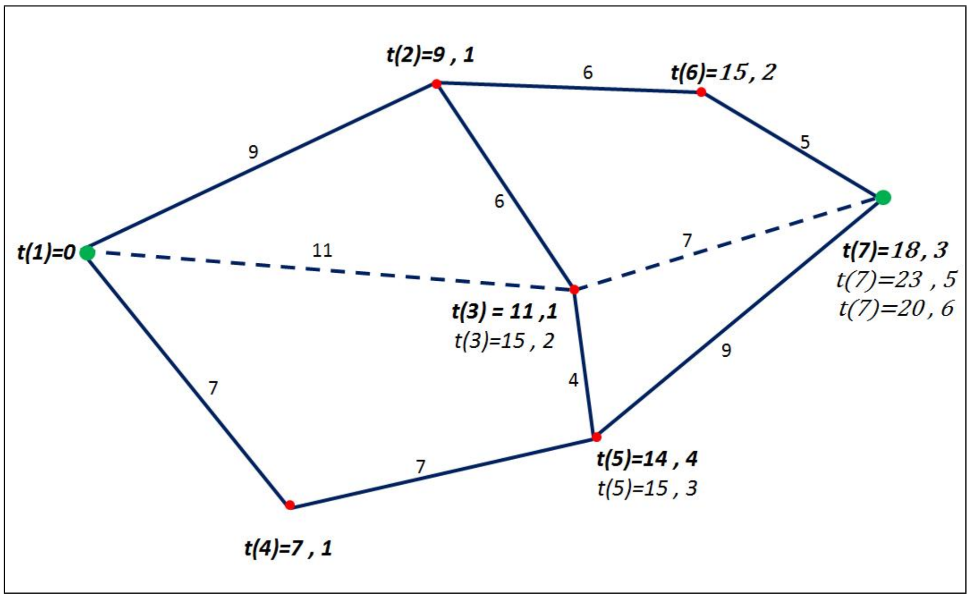


Figure 4.31: 1^{st} shortest path

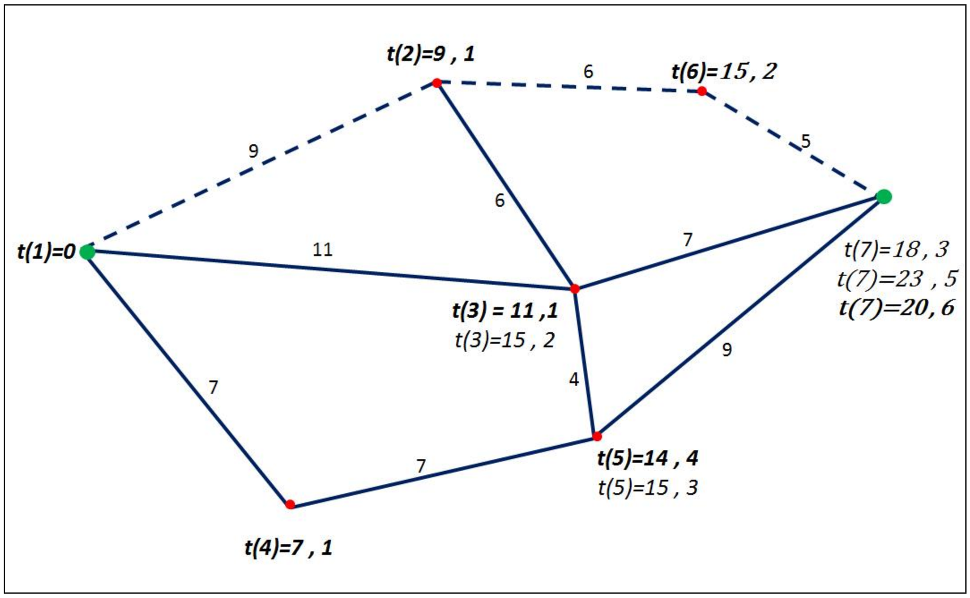


Figure 4.32: 2^{nd} shortest path

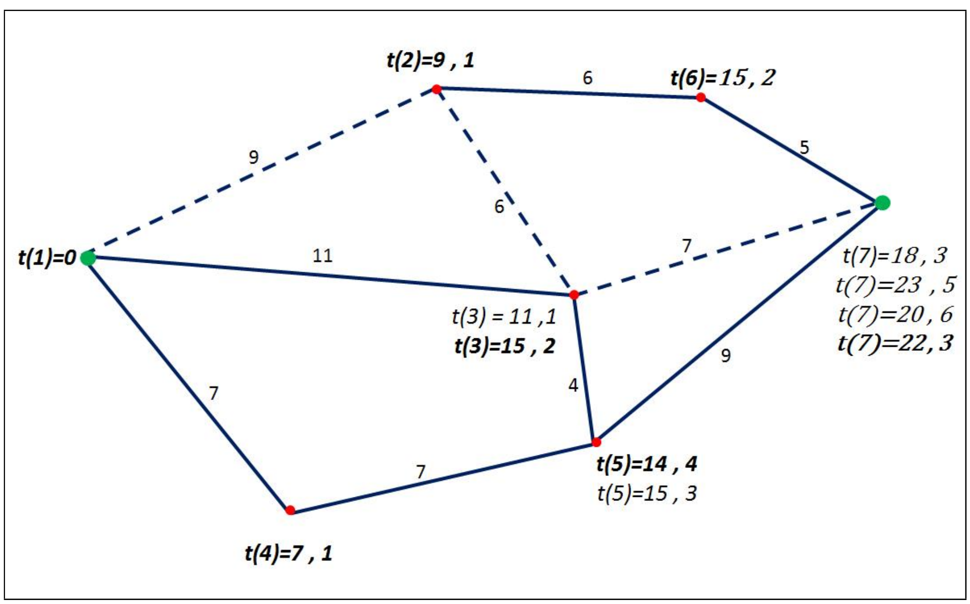


Figure 4.33: 3^{rd} shortest path

The K shortest path problem can be used to solve the shortest path problem for different situations and provide choices for users so that they may select the best path that fits their requirements; but, can be costly to calculate K shortest path for each request. The problem becomes more significant when K changes for each request or the value of edges in the graph change dynamically (e.g. changes by traffic congestion). Managing all possible paths in a database and changing the edge values dynamically can be a solution to this problem. The next section deals with the all possible paths problem.

4.4.3 All Possible Paths Problem

The all possible paths problem is an alternative to the K shortest path problem where the value of K may range from a minimum (i.e. 1) to maximum number of paths dynamically. Besides solving this problem, all possible paths methods try to find solutions for other problems such as the travelling salesman problem (TSP), in which several destinations must be visited within a network.

To deal with all the possible paths problem we start with a complete graph. By definition, "a complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge." Considering this definition we can conclude that:

1. A complete graph on n vertices has $\frac{n(n-1)}{2}$ edges (K_n).
2. The number of all possible paths from a node to another node equals:

$$No_{allpossiblepaths} = \sum_{i=0}^{n-2} P_i^{n-2} = \sum_{i=0}^{n-2} \frac{(n-2)!}{(n-2-i)!} \tag{4.4}$$

3. The number of times edges are visited (except for direct edge which equals 1)

equals:

$$\frac{No_{allpossiblepaths} - 1}{n - 2} \tag{4.5}$$

4. The minimum number of edges in a path is 1 and the maximum number of edges in a path is $n - 1$.

Figure 4.34 and figure 4.35 illustrate two examples of complete graphs and all possible paths between two arbitrary nodes.

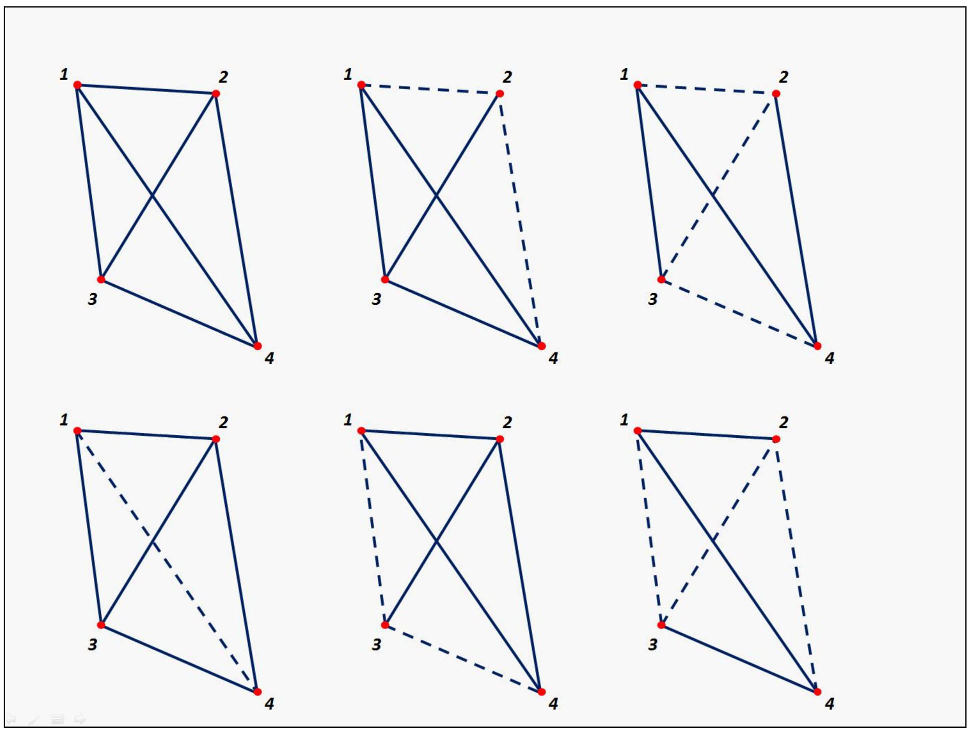


Figure 4.34: All possible paths from node 1 to node 4. $N = 4, K_n = 6, No_{allpossiblepaths} = 5$

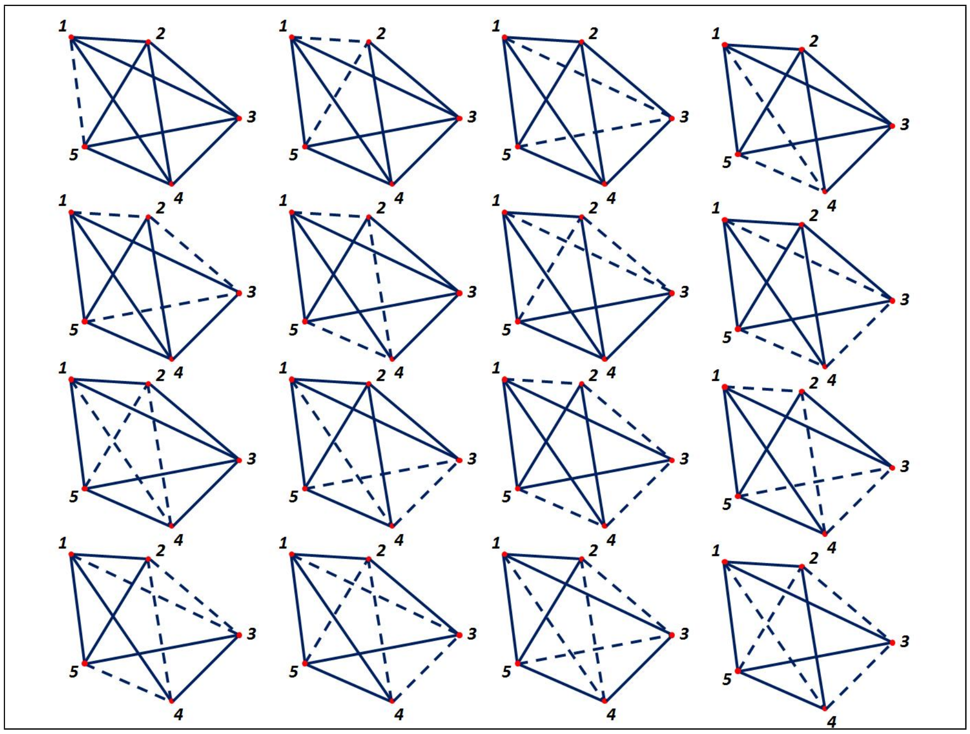


Figure 4.35: All possible paths from node 1 to node 5. $N = 5, K_n = 10, No_{allpossiblepaths} = 16$

Finding all paths from a source/origin to a target/destination and sorting them by total weight can be considered as an alternative K shortest path solution. This method has two main disadvantages. First, this method requires a very large number of computations and memory addresses. Second, it takes the same effort to solve problems for small or large K [56]. However, we can address these disadvantages and provide more efficient solutions.

In order to reduce the computation effort we can substitute mathematical computations for finding all paths with a simple join operation in a database. We assume that for each edge/segment of the graph there are start and end point attributes which store unique ids of the start and end points of each edge/segment in addition to length or value/weight/cost of the edge/segment following the arc/node topological structure. We have to provide the database with origin and destination nodes. The process starts with an iteration in which database tries to find all segments that start from that node. Then the algorithm searches all other segments for which the start point is equal to the end point of previous segments. This process iterates until the end node of the path is reached. It is important to know whether or not the graph is directed or undirected. If the graph is undirected we have to examine both start and end points of each edge/segment. But if it is directed, matching the start point of each segment with the end point of previous segments is sufficient. Figures 4.36 to figure 4.40 illustrate this algorithm.

It takes 4 join operations to find all three possible paths in this example. The results are provided in figure 4.38 to figure 4.40. In the worst case it takes $\| E \|$ join operations to detect all possible paths in which $\| E \|$ is the number of edges/segments in the graph. All possible paths algorithm is compatible with database operations and can be stored in a database. The results of all possible paths algorithm is independent of the length of segments which makes it appropriate for dynamic situations in which the value of

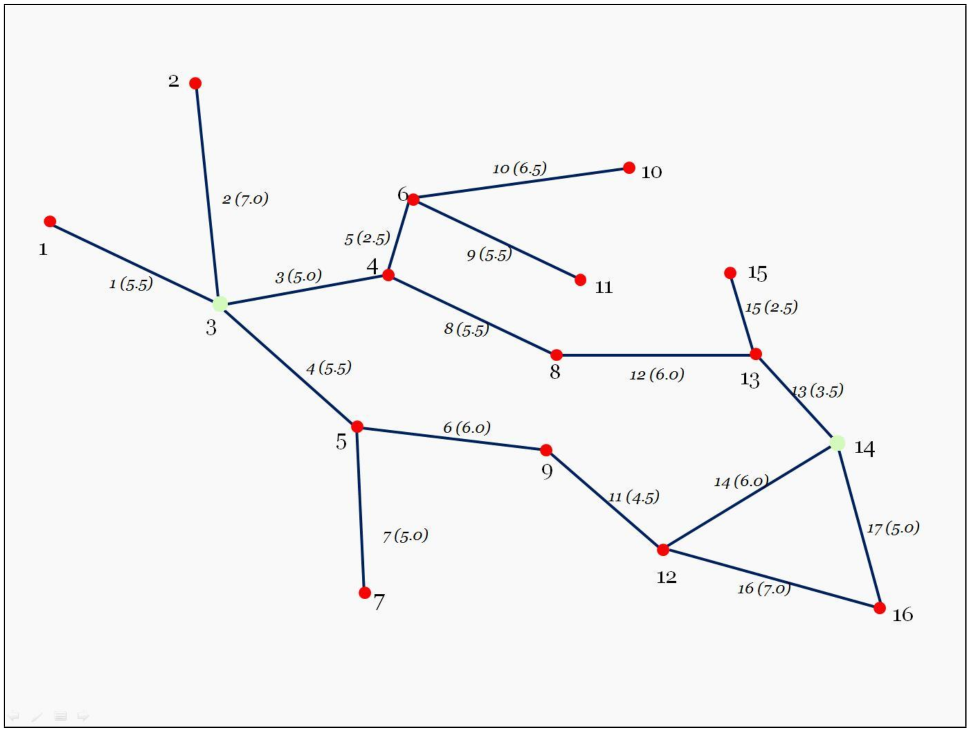


Figure 4.36: Find all possible paths from node 3 to node 14

segments change permanently. Also the all possible paths algorithm provides better results of path finding when there are more than one destination. The database can be queried to identify which paths meet all requested destinations. Beside these advantages the results of all possible paths can be used for data mining purposes. For example all destinations in a particular area may be connected to other areas by a specific node. Using all possible paths database, one can detect these nodes.

4.5 Summary

This chapter discussed methodologies for indoor positioning and path finding utilized in this research. Location fingerprinting algorithms including KNN, SVM, ANN, and statistical distances were tested. In this chapter we observed that statistical distances find the best match for signal distribution patterns and can be used to reduce the

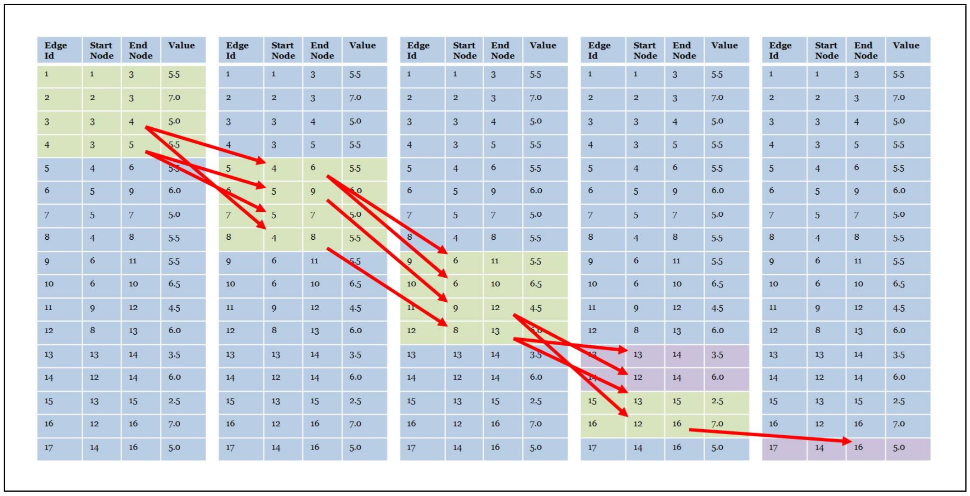


Figure 4.37: Finding all possible paths using database join operation

number of calibration points or to filter calibration points for other methods. Shortest path algorithms including Dijkstra, K shortest path and all possible paths were investigated for their applications in path finding problems. K shortest path solutions provide different options for path finding and the all possible paths provide all routable paths between two nodes regardless of their values, which can be used in different applications from traffic management to pedestrian distribution.

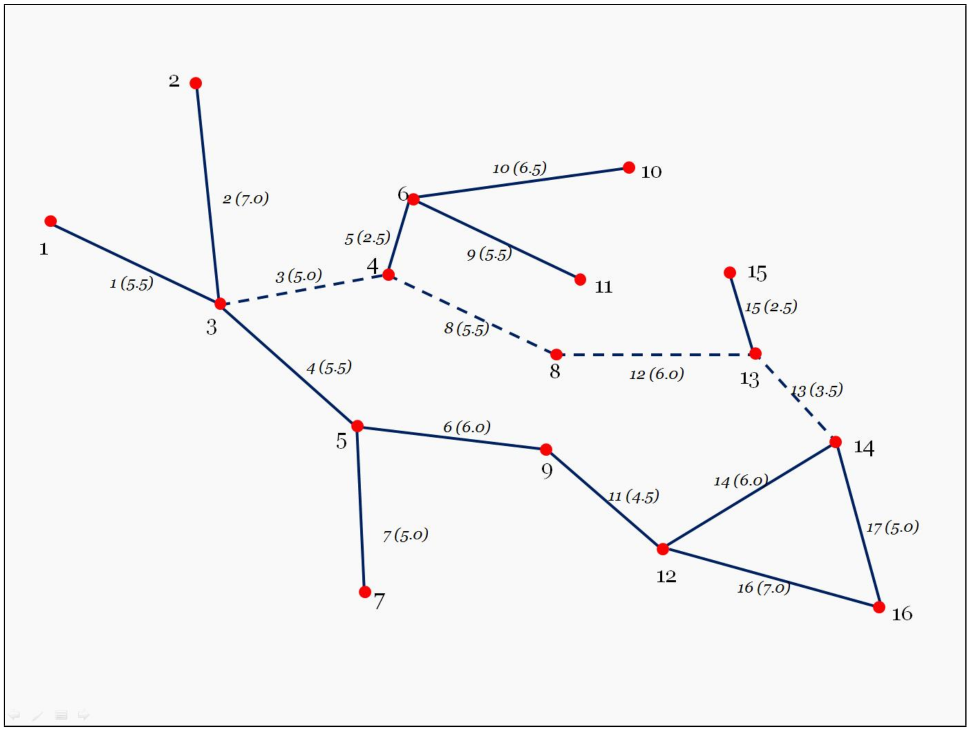


Figure 4.38: 1^{st} path: $3 \rightarrow 4 \rightarrow 8 \rightarrow 13 \rightarrow 14$; total weight=20

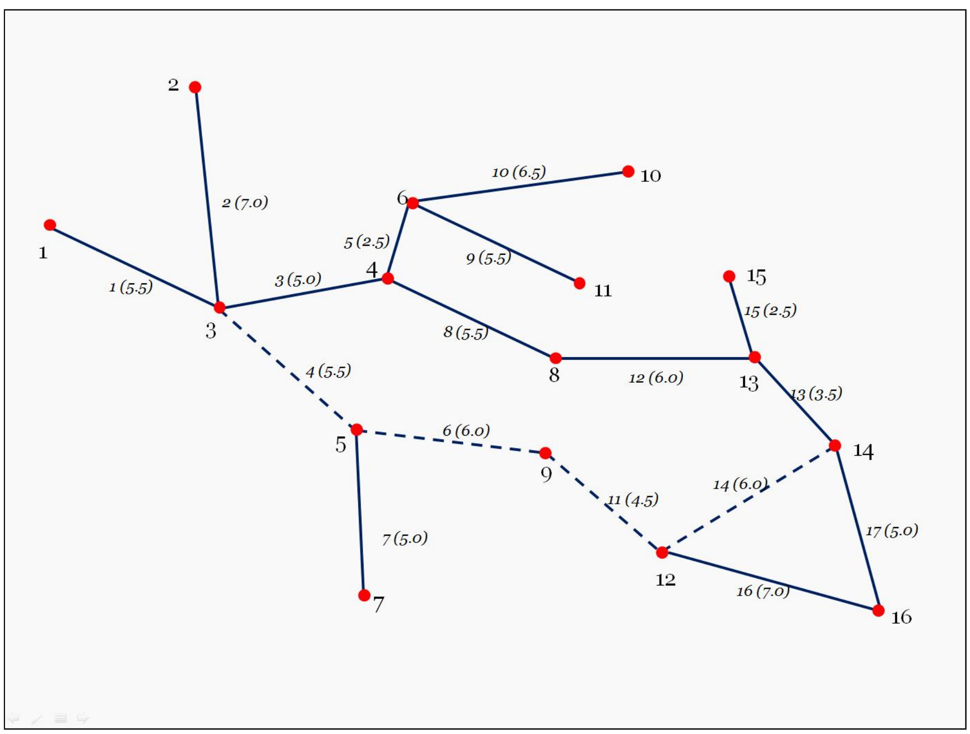


Figure 4.39: 2^{nd} path: $3 \rightarrow 5 \rightarrow 9 \rightarrow 12 \rightarrow 14$; total weight=22

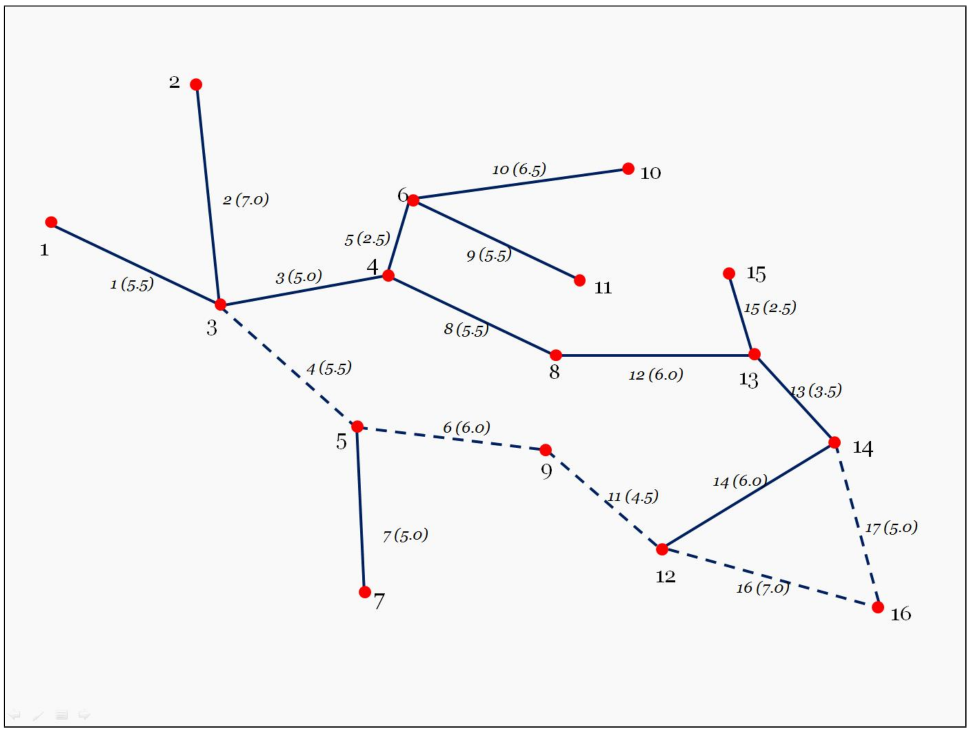


Figure 4.40: 3^{rd} path: $3 \rightarrow 5 \rightarrow 9 \rightarrow 12 \rightarrow 16 \rightarrow 14$; total weight=28

Chapter 5

Results and Discussion

The results of this research are presented in this chapter for both positioning and path finding. Findings from the implementation of different fingerprinting methods are detailed and comparisons are made between each method. ANOVA is used to compare each method. For different path finding solutions, details of algorithms are discussed and algorithms' time complexity are measured and compared. This chapter addresses two objectives of this research. First, using signal information to its full capacity for positioning purpose; and second, providing path finding solutions for clients with different requirements or preferences.

5.1 Results for Positioning

Data collection in this study showed that strong observed signals are not necessarily from APs on the same floor. In other words, at some points signal strengths from APs on different floors may be stronger than signals from APs of the same floor. However, indoor positioning systems are qualified by six criteria: symbology, error, location rate, scalability, cost, and centralized vs. distributed computing [57]. Accepting these criteria, we compare stated location fingerprinting methods.

Symbology refers to the location model used for positioning. The K Nearest Neighbour (KNN) method requires coordinates of neighbour points in order to solve the position of the unknown point. This method works for geometric location models. For Artificial Neural Network (ANN), we must provide the system the location of calibration points. With this method position is required as a pair of coordinates;

making it suitable for use with a geometric location model.

The Support Vector Machine method differs from KNN and ANN; since it can work on geometric coordinates of the points and the symbolic location of the points, while KNN and ANN can work just on geometric coordinates. Advanced SVM methods can use numerical values for classification or prediction, which means they can utilize a geometric location model. SVM also is suitable for use with symbolic location models where the location of points is expressed via symbols (or classes). For statistical distances the location of the control points are not required. These methods just look for the best match among the calibration points and the position or location of the best fitting calibration point is assigned to the unknown point; so these methods are independent of a location model and can work on both geometric and symbolic location models.

Error is a parameter used to evaluate the precision or accuracy of the positioning method. In order to assess the precision of a positioning method we usually accept Root Mean Square Error (RMSE) as a measure of error in the system. The lower the RMSE, the more precise the method.

The KNN method is very sensitive to the granularity of the calibration points and the number of calibration points used (K). In this work the best precision achieved was RMSE=11.28m for K=11. It is important to note that the space between points was 5m and the grid was not regular. The linear kernel produced the optimal solution for SVM with an RMSE=13.50m. ANN did not provide a better result. RMSE for ANN was 17.62m. However we can consider ANN to be a non-deterministic method which may produce different results from the same input data. The best results for statistical distance methods belongs to Bhattacharyya distance for signal strengths more than -60[dBm] which resulted in an RMSE of 16.49m.

In order to evaluate accuracy we can also assess percent of points that are predicted to be in their original location. However, for KNN method this is difficult to assess as K becomes large, because more points are involved in the prediction process. In order to have a good understanding of accuracy this work suggests that K=1 or K=11 are best. For K=1, an accuracy of 88% was achieved and for K=11 accuracy was 80%. SVM using different kernels gave the following results: linear kernel 72%, polynomial and radial basis kernels is 75% and for sigmoid kernel 70%. Evaluation of ANN from an accuracy point of view is difficult. One reason is that the parameters required to find the best structure of ANN can only be achieved through a long process of trial and error and these parameters cannot be generalized for other areas. However, for the final test of ANN we reached an accuracy of 67% with 16 neurons in hidden layer. The accuracy for statistical distances for different methods (Mahalanobis and Bhattacharyya) is almost identical. For the Bhattacharyya distances, an accuracy of 81% was achieved; while for Mahlanobis distance the best accuracy achieved was of 80%.

Location rate refers to the changes in location we can predict. If the method can provide a continuous positioning solution (e.g. KNN, ANN and numerical SVM), continuous prediction is possible. This implies utilizing a geometric location model. However, if we choose SVM or statistical distances, which use a symbolic location model, the best rate depends on the granularity of the points or locations in the calibration dataset.

Scalability refers to the fact that we can generalize a method from a local system to a global system. None of the location fingerprinting methods can be readily generalized. All these methods depend strictly on the infrastructure in which they work. Also each method may use different parameters in different areas, which means calibration in one area may not provide appropriate parameters for prediction in other areas.

Cost is a parameter that usually refers to the amount of time or CPU usage to predict the location, as well as time to install infrastructure and collect calibration data.

Through the experiment the fastest method was KNN followed by statistical distance methods, then SVM, and ANN was the slowest. However, we can also consider the computation effort required for each method. KNN and statistical distance utilize simple mathematical concepts which are fast to compute. However, SVM and ANN use more complicated math models and also compute many iterations that contain a number of loops, which makes them more difficult to compute.

The data collection phase in location fingerprinting is also cumbersome, and a large number of signal strength observations along with their position/location information should be collected; and if the positions of the access points change, the data collected in that vicinity will no longer be valid and must be collected again [44]. Statistical distance methods are the only methods that can work on a small number of calibration points chosen properly for each location. For the other methods, KNN, SVM, and ANN, the more calibration points collected the better the results achieved.

Centralized vs. Distributed refers to the architecture of the positioning system. All of these methods can either be centralized in a server or distributed over the network. For this study, the fingerprinting methods used information collected on the client side to predict position, we did not focus on the architecture of the server.

Table 5.1 summarizes best results for all methods according to criteria mentioned.

Table 5.1: Summary of location fingerprinting results

	Symbology	**Error**		**Location Rate**	**Cost**
		Precision	Accuracy		
KNN	Geometric	11.28m	88%	Continuous	Low
SVM	Geometric/Symbolic	13.50m	75%	Continuous/Discrete	Medium
ANN	Geometric	17.62	67%	Continuous	High
Statistical Distances	Neutral	16.49m	81%	Discrete	Low

To compare the error results of each method a one-way ANOVA was used to test for error differences between the four methods using the best result from each method as described in Chapter 4.. Error derived from the methods differed significantly, F(3,212)=15.283, MSE=61.295, p=4.84e-09. Pairwise comparison using Fisher's LSD (minimum mean difference = 4.013m) revealed that KNN2 and SVM produced significantly lower errors than the other tests, and that KNN2 gave the best results with a mean error of 7.780m (se=0.941m). Figure 5.1 ishows that KNN2 and SVM gave similar location accuracies (labeled [group] b), and that the Bhattacharyya distance for signals greater than -55 dBm and ANN with 14 neurons in the hidden layer (labeled [group] a) also gave similar, but significantly worst results. As such, we can conclude that, given the data, KNN (K=2) provides overall the most precise results, and ANN (hidden=14) provides the least precise results.

5.2 Results for Path Finding

The complexity function is the best criteria to compare the results of different path finding algorithms, and is described as a function of time and memory use. Here, we consider three different algorithms and compare them using time complexity: Dijkstra's shortest path, K shortest paths and all possible paths. The worst-case running time for the Dijkstra algorithm on a graph with n nodes and m edges is $O(n^2)$ [54]. However, time complexity for the K shortest simple path algorithm for a graph with n nodes and m edges is $O(k(m + nlogn))$ in undirected graphs and $O(kn(m + nlogn))$ in directed graphs. [58].

The process of finding K shortest path when we have all possible paths saved in a database for a graph of n nodes and m edges includes search for the start node with $O(logn)$ complexity [54], search for end node with $O(logn)$ complexity [54] and sorting the results with $O(klogk)$ complexity in the worst case [59]; which results in a complexity of $O(2logn + klogk)$ in the worst case, where k is the number of all possible

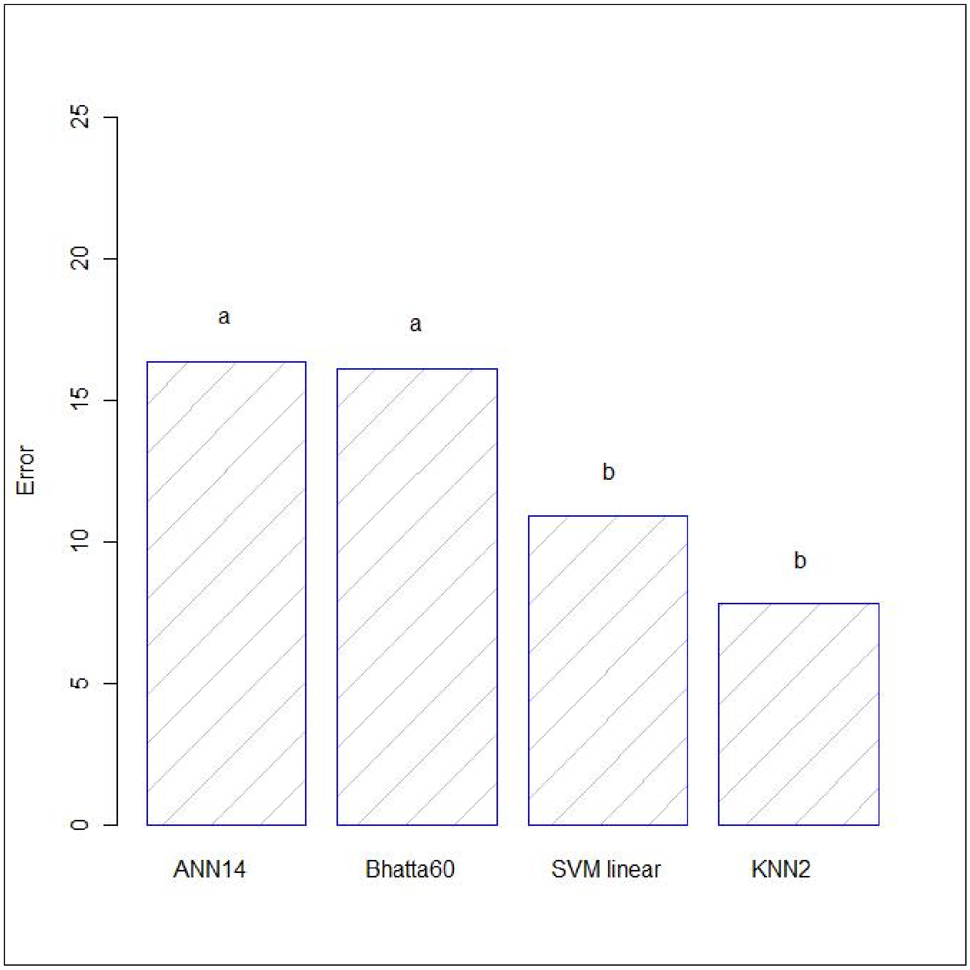


Figure 5.1: Result of ANOVA test for the best results of each fingerprinting methods (ANN, Statistical distance, SVM, KNN)

paths between two nodes. This method requires more effort when we deal with dynamic changes in edge weights. However in order to prepare a database of all possible paths we need to perform a preprocessing phase. The preprocessing phase for a complete graph of n nodes and $\frac{n(n-1)}{2}$ edges has a running time of $O(n(n-1)^2)$. This value is the result of finding all possible paths from n sources to $n-1$ destination performing $n-1$ (maximum number of edges in a path) joins in the database.

Finding and storing all possible paths in a database has other advantages. For example, we can extract the maximum value of K from a source to a destination, so the algorithm can act more efficiently with large values of K. We can also distinguish the results of all possible paths for different types of users. For example we can consider users' preferences and extract those paths that consider those preferences. When the results of a path finding algorithm is limited to a set of specific paths, the all possible paths algorithm provides more efficient results in comparison with other algorithms. We can also test and extract patterns and trajectories for both edges with weights or independent of weights. As all the edges of paths are saved in a database, performing spatial queries and enhancing the flexibility of algorithms using spatial operators becomes more valuable.

5.3 Summary

In this chapter some characteristics of different location fingerprinting methods were discussed. KNN provided the best precision. However, statistical distances can be used to find the best match in signal distribution and filter the candidate calibration points. So a combination of KNN and statistical distances can both reduce the number of calibration points and provide precise results. From a path finding perspective, all possible paths can be used to store paths between two distinct nodes. This method suggests better performance when compared to the K shortest path algorithm, which makes it appropriate for dealing with dynamic changes in the network. Also, providing the paths in a database can aid combining path finding analysis with other spatial analysis methods.

Chapter 6

Conclusion

This chapter summarizes findings of this research as related to the objectives stated in the first chapter. Also some enhancements for future works are suggested.

6.1 Summary of Findings

The main advantages of location fingerprinting methods are their reliance on standard wireless networks and their ability to preserve privacy as positioning is executed by the client. Also during the training phase, knowledge of the position of access points is not required [47]. This advantage is general to all positioning methods.

In this research, we tested different methods of location fingerprinting. KNN, SVM, and ANN methods have been suggested and tested by other researchers. In this research we implemented statistical distances to understand distribution similarities in the signal space and determine location based on those similarities.

Through the testing process, KNN and statistical distances methods showed efficient solutions to indoor positioning problem. Time and memory consumption for estimation by these two methods offer improvement over the SVM and ANN methods, which makes them more suitable for implementation for mobile and portable devices.

The ANN and SVM methods are more complicated and contain hidden processes, which makes them ambiguous. However, KNN and statistical distances have clearly defined processes and give developers full control over optimization of the positioning process.

The KNN method is both fast and reliable. The main disadvantage of this method is its reliance on a fine grid of calibration points. In order to improve the accuracy of this method we need to provide a finer mesh of calibration points. Statistical distances showed that choosing well located calibration points is more important than their granularity, as points in the same area show similar signal distribution patterns, which makes them promising methods for reducing the number of calibration points.

In order to combine path finding solutions with a positioning solution for indoor environments we need to project the results of positioning into a graph. There are different location models that are suitable for this process. Map matching algorithms are a solution to this problem.

For path finding problems we need to build a graph that reflects movement corridors within the indoor environment. However, indoor movement corridors differ from road networks in the dimensionality of movement, which is usually represented as a line by road networks, whereas the path in an indoor environment is generally represented by a mesh of paths. This is one of the reasons why solution to the shortest path problem does not provide the shortest path.

The K shortest paths algorithm is a solution that provides more than one choice for navigating from one place to another. This solution can consider people's preferences and local environmental context. For example, if people have a range of times within which to navigate from a place to another, they may prefer to choose more than one destination along the way, or perhaps they have special location needs (i.e. the disabled). In these instances K shortest paths offer a good solution.

While K shortest paths provide alternatives it is also time consuming to generate K shortest paths, especially when the value of K increases. Extracting all possible paths using relational operators found in a database is an alternative for K shortest paths solution. However, the all possible paths method has its own advantages over K shortest paths; such as flexibility for networks that change dynamically. These advantages make this method more suitable for dynamic applications, planning applications, and pattern mining applications.

6.2 Future Research

As noted by Prasithsangaree et al. [7] the relationships between deployment issues, accuracy, size of database, robustness of system and performance needs more investigation. As location fingerprinting tends to be used on mobile and portable devices, considering hardware issues is inevitable.

There are some concerns in the data collection process that need further assessment. First of all, RSS received from the same APs at the same location may vary with orientation [40]. We need to either design new types of receivers to consider this issue or develop algorithms to compensate for this problem. Additionally, the amount of noise received with the signal information is problematic. We need to develop methods to filter out the noise and fluctuation commonly observed in WiFi signals. Fluctuations in signal strength can be minimized if some filtering (e.g. Kalman filtering) is applied [42], or if signal strength is measured at the AP so that signals can be differentiated. Collection of sufficient calibration is also a limitation of location fingerprinting. This research assumes that enough time is spent on data collection in both the calibration and positioning phases. However, in practice we need to position users as they walk, which means data collection must be rapid. Another issue arises with the reverse problem- when a user is staying in one location for a long time. Periodic updating of mobile position generates a large number of messages between the client and server (i.e. high traffic/bandwidth cost) [12].

Yeung and Ng [60] propose utilizing the asymmetrical nature of signal strength to improve the accuracy of positioning. The asymmetry of the signal is the result of differences between transmission power of access points and mobile devices.

Another issue that can be considered during future work is positioning for areas with a small number of access points. In our research we ran our tests in an environment with enough access points, such that at every point we could observe at least three access points. However, for some areas (e.g. residential apartments or houses) a user may only have one or two access points that are visible. Some research such as [27] focus on indoor positioning based on only one access point, utilizing a Monte Carlo sampling method.

From a path finding point of view, the all possible paths is a promising solution that allows the combination of different parameters such as spatial relations between paths with other features, along with the weight/value of edges. This method needs further investigation to provide solutions for more path finding problems such as the travelling salesperson problem, and the longest path problem.

Bibliography

[1] D. Li and D. L. Lee, "A topology-based semantic location model for indoor applications," in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, ser. GIS '08. New York, NY, USA: ACM, 2008, pp. 6:1–6:10. [Online]. Available: http://doi.acm.org/10.1145/1463434.1463443

[2] S. Spiekermann and H. U. Z. Berlin, "General aspects of location-based services," in *In Location-Based*. Morgan Kaufmann Publishers, 2004.

[3] A. E. Stefan Steiniger, Moritz Neun, "Foundations of location based services," Department of Geography, University of Zurich, Lecture Note, 2006.

[4] D. Quercia, N. Lathia, F. Calabrese, G. D. Lorenzo, and J. Crowcroft, "Recommending social events from mobile phone location data," *Data Mining, IEEE International Conference on*, vol. 0, pp. 971–976, 2010.

[5] F. DÃ̋rr and K. Rothermel, "On a location model for fine-grained geocast," in *UbiComp 2003: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, A. Dey, A. Schmidt, and J. McCarthy, Eds. Springer Berlin / Heidelberg, 2003, vol. 2864, pp. 18–35, 10.1007/978-3-540-39653-6-2. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39653-6-2

[6] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," in *In HUC 99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. Springer-Verlag, 1999, pp. 304–307.

[7] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless lans," in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, vol. 2, sept.

2002, pp. 720 – 724 vol.2.

[8] E. Inc, "Comparison of wireless indoor positioning technologies," Ekahau Inc., Tech. Rep., 2005.

[9] N. Chan, "Introduction to location-based services," *The Oresund Summer University*, August 2003.

[10] T. Kos, M. Grgic, and J. Kitarovic, "Location technologies for mobile networks," in *Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on*, june 2007, pp. 319 –322.

[11] D. Madigan, E. Einahrawy, R. Martin, W.-H. Ju, P. Krishnan, and A. Krishnakumar, "Bayesian indoor positioning systems," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2, march 2005, pp. 1217 – 1227 vol. 2.

[12] M. Kjrgaard, G. Treu, and C. Linnhoff-Popien, "Zone-based rss reporting for location fingerprinting," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. LaMarca, M. Langheinrich, and K. Truong, Eds. Springer Berlin / Heidelberg, 2007, vol. 4480, pp. 316–333, 10.1007/978-3-540-72037-9-19. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72037-9-19

[13] A. Taheri, A. Singh, and A. Emmanuel, "Location fingerprinting on infrastructure 802.11 wireless local area networks (wlans) using locus," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, nov. 2004, pp. 676 – 683.

[14] C. Jensen, H. Lu, and B. Yang, "Graph model based indoor tracking," in *Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on*, may 2009, pp. 122 –131.

[15] T. D'Roza and G. Bilchev, "An overview of location-based services," *BT Technology Journal*, vol. 21, pp. 20–27, 2003, 10.1023/A:1022491825047. [Online]. Available: http://dx.doi.org/10.1023/A:1022491825047

[16] Y. Chen and H. Kobayashi, "Signal strength based indoor geolocation," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 1, 2002, pp. 436 –439.

[17] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless lans," *Computer Networks*, vol. 47, no. 6, pp. 825 – 845, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128604002610

[18] F. Lassabe, P. Canalda, P. Chatonnay, and F. Spies, "Indoor wi-fi positioning: techniques and systems," *Annals of Telecommunications*, vol. 64, pp. 651–664, 2009, 10.1007/s12243-009-0122-1. [Online]. Available: http://dx.doi.org/10.1007/s12243-009-0122-1

[19] R. Golledge, "Path selection and route preference in human navigation: A progress report," in *Spatial Information Theory A Theoretical Basis for GIS*, ser. Lecture Notes in Computer Science, A. Frank and W. Kuhn, Eds. Springer Berlin / Heidelberg, 1995, vol. 988, pp. 207–222, 10.1007/3-540-60392-1-14. [Online]. Available: http://dx.doi.org/10.1007/3-540-60392-1-14

[20] C. A. Lawton, "Strategies for indoor wayfinding: The role of orientation," *Journal of Environmental Psychology*, vol. 16, no. 2, pp. 137 – 145, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0272494496900110

[21] H. Hu and D.-L. Lee, "Semantic location modeling for location navigation in mobile environment," *Mobile Data Management, IEEE International Conference on*, vol. 0, p. 52, 2004.

[22] P. yves Gilliéron, D. Bchel, I. Spassov, and B. Merminod, "Indoor navigation

performance analysis," in *In Proceedings of the European Navigation Conference GNSS 2004*, 2004.

[23] C. Becker and F. DÃ̋rr, "On location models for ubiquitous computing," *Personal and Ubiquitous Computing*, vol. 9, pp. 20–31, 2005, 10.1007/s00779-004-0270-2. [Online]. Available: http://dx.doi.org/10.1007/s00779-004-0270-2

[24] D. Li and D. L. Lee, "A lattice-based semantic location model for indoor navigation," in *Mobile Data Management, 2008. MDM '08. 9th International Conference on*, april 2008, pp. 17 –24.

[25] R. Atkin, "From cohomology in physics to q-connectivity in social science," *International Journal of Man-Machine Studies*, vol. 4, no. 2, pp. 139 – 167, 1972. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020737372800294

[26] C. Hlscher, T. Meilinger, G. Vrachliotis, M. Brösamle, and M. Knauff, "Up the down staircase: Wayfinding strategies in multi-level buildings," *Journal of Environmental Psychology*, vol. 26, no. 4, pp. 284 – 299, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0272494406000582

[27] G. Zruba, M. Huber, F. Kamangar, and I. Chlamtac, "Indoor location tracking using rssi readings from a single wi-fi access point," *Wireless Networks*, vol. 13, pp. 221–235, 2007, 10.1007/s11276-006-5064-1. [Online]. Available: http://dx.doi.org/10.1007/s11276-006-5064-1

[28] J. Hightower and G. Borriello, "Location sensing techniques," IEEE Computer, Tech. Rep., 2001.

[29] G. Franz, H. A. Mallot, J. M. Wiener, and K. Neurowissenschaft, "Graph-based models of space in architecture and cognitive science - a comparative analysis," in *Proceedings of the 17th International Conference on Systems Research, Informatics*

and Cybernetics (InterSymp2005), Architecture, Engineering and Construction of Built Environments, pages 3038. The International Institute for Advanced Studies in System R, 2005, pp. 30–38.

[30] M. D. Michael Worboys, *GIS: A Computing Perspective*, 2nd ed. CRC PRESS LLC, 2004.

[31] W. Hoffman and R. Pavley, "A method for the solution of the nth best path problem," *J. ACM*, vol. 6, pp. 506–514, October 1959. [Online]. Available: http://doi.acm.org/10.1145/320998.321004

[32] T. Wolbers and M. Hegarty, "What determines our navigational abilities?" *Trends in Cognitive Sciences*, vol. 14, no. 3, pp. 138 – 146, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661310000021

[33] A. Klippel, P. U. Lee, S. Fabrikant, D. R. Montello, and J. Bateman, "The cognitive conceptual approach as a leitmotif for map design," in *In Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance. Papers from the AAAI 2005 Spring Symposium. March 21-23.* AAAI Press, 2005, pp. 90–95.

[34] A. L. Liu, H. Hile, H. Kautz, G. Borriello, P. A. Brown, M. Harniss, and K. Johnson, "Indoor wayfinding:: developing a functional interface for individuals with cognitive impairments," in *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, ser. Assets '06. New York, NY, USA: ACM, 2006, pp. 95–102. [Online]. Available: http://doi.acm.org/10.1145/1168987.1169005

[35] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959, 10.1007/BF01386390. [Online]. Available: http://dx.doi.org/10.1007/BF01386390

[36] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, pp. 91–102, January 1992. [Online]. Available: http://doi.acm.org/10.1145/128756.128759

[37] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," *Wireless Networks*, vol. 8, pp. 187–197, 2002, 10.1023/A:1013767926256. [Online]. Available: http://dx.doi.org/10.1023/A:1013767926256

[38] H. Araklian, "Indoor and outdoor location sensing," RWTH-AACHEN, Tech. Rep.

[39] M. Hermersdorf, "Indoor positioning with a wlan access point list on a mobile device," *ACM*, October 2006.

[40] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2, march 2004, pp. 1012 – 1022 vol.2.

[41] "IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1 –1184, 12 2007.

[42] F.Evennou and F. Marx, "Improving positioning capabilities for indoor environments with wifi," *IST*, 2005.

[43] R. Battiti, A. Villani, and T. L. Nhat, "Neural network models for intelligent networks: Deriving the location from signal patterns," in *IN PROCEEDINGS OF AINS2002, (UCLA)*, 2002.

[44] M. Robinson and I. Psaromiligkos, "Received signal strength based location estimation of a wireless lan client," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, march 2005, pp. 2350 – 2354 Vol. 4.

[45] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998, 10.1023/A:1009715923555. [Online]. Available: http://dx.doi.org/10.1023/A:1009715923555

[46] D. Meyer, *Support Vector Machines: The interface to libsvm in Package e1071*, Techische Universitat Wien, Austria, 2004.

[47] R. Battiti, T. L. Nhat, and A. Villani, "Location-aware computing: A neural network model for determining location in wireless lans," Tech. Rep., 2002.

[48] G. F. Fritsch S, "neuralnet: Training of neural networks," *The R Journal*, vol. 2/1, pp. 30–38, June 2010.

[49] I. Guvenc, "Enhancements to rss based indoor tracking systems using kalman filters," in *In GSPx & International Signal Processing Conference*, 2003.

[50] P. Mahalanobis, "On the generalized distance in statistics," *Indian National Science Academy*, 1936.

[51] S. O. Carl J Huberty, *Applied MANOVA and Discriminant Analysis*, 2nd ed. WILEY, 2006.

[52] A. Bhattacharyya, *On a measure of divergence between two multinomial populations.* Sankhya, 1946, vol. 7.

[53] J. Ye, L. Coyle, S. Dobson, and P. Nixon, "A unified semantics space model," in *Location- and Context-Awareness*, ser. Lecture Notes in Computer Science, J. Hightower, B. Schiele, and T. Strang, Eds. Springer Berlin / Heidelberg, 2007,

vol. 4718, pp. 103–120, 10.1007/978-3-540-75160-1-7. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75160-1-7

[54] M. D. Michael Worboys, *GIS: A Computing Perspective*, 2nd ed., M. D. Michael Worboys, Ed. CRC PRESS LLC, 2004.

[55] S. E. Dreyfus, "An appraisal of some shortest-path algorithms," *Operations Research*, vol. 17, no. 3, pp. pp. 395–412, 1969. [Online]. Available: http://www.jstor.org/stable/168375

[56] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. pp. 712–716, 1971. [Online]. Available: http://www.jstor.org/stable/2629312

[57] J. A. Tauber, "Indoor location systems for pervasive computing," August 2002.

[58] D. Eppstein, "Finding the k shortest paths," *SIAM J. Comput.*, vol. 28, pp. 652–673, February 1999. [Online]. Available: http://dx.doi.org/10.1137/S0097539795290477

[59] O. Astrachan, "Bubble sort: an archaeological algorithmic analysis," *SIGCSE Bull.*, vol. 35, pp. 1–5, January 2003. [Online]. Available: http://doi.acm.org/10.1145/792548.611918

[60] W. M. Yeung and J. K. Ng, "Wireless lan positioning based on received signal strength from mobile device and access points," *Real-Time Computing Systems and Applications, International Workshop on*, vol. 0, pp. 131–137, 2007.