IMPROVEMENT OF RSS-BASED INDOOR POSITIONING SYSTEMS BY ENHANCING LOCATION SENSING ALGORITHMS

by
Bulut ALTINTAŞ

Submitted to the Institute of Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Computer Engineering

Yeditepe University
2013

IMPROVEMENT OF RSS-BASED INDOOR POSITIONING SYSTEMS BY ENHANCING LOCATION SENSING ALGORITHMS

APPROVED BY:

Assist. Prof. Dr. Tacha Serif: …………………………………………
(Supervisor)

Prof. Dr. Şebnem Baydere: …………………………………………

Assist. Prof. Dr. Barış Aktemur: …………………………………………

DATE OF APPROVAL: …./…./2013

ACKNOWLEDGEMENTS

I am especially thankful to my supervisor, Tacha Serif, whose support and guidance from the initial to the final level enabled me to accomplish this research.

I also offer my regards and blessings to my friend Kadir Yüceer, who supported me in any respect during the completion of the project.

I would like to thank all the instructors in Computer Engineering department of Yeditepe University, as they shared their wisdom and enlightened me on the way through becoming an engineer.

Finally, I would also like to extend my thanks to the Scientific and Technological Research Council of Turkey (TÜBİTAK) who provided valuable scholarship during my Master Thesis.

ABSTRACT

IMPROVEMENT OF RSS-BASED INDOOR POSITIONING SYSTEMS BY ENHANCING LOCATION SENSING ALGORITHMS

Advances in mobile technologies have changed the way users interact with devices and other users. These new interaction methods and services are offered by the help of intelligent sensing capabilities, utilizing context, location and motion sensors. However, indoor location sensing is mostly achieved by measuring radio signal (WiFi, Bluetooth, GSM etc.) strength and nearest neighbor identification. The most common algorithm adopted for Received Signal Strength (RSS)-based location sensing is K Nearest Neighbor (KNN), which calculates K nearest neighboring points to estimate location. Nevertheless, fluctuation on the received signal strength is one of the crucial problems in the RSS-based KNN algorithm. Adopting the fluctuated signals for positioning may lead to inaccurate results. In this study, the accuracy of the RSS-based KNN algorithm is targeted to be enhanced by eliminating the effects of fluctuated signals. For this purpose three separate enhancements are applied to the KNN algorithm. In the first proposed system, the accuracy of the KNN algorithm is attempted to be incremented by exploiting wireless mesh network capabilities. This approach is to share the location data among devices and utilize them into the location estimation. However, the results showed that the proposed systems could not provide the intended accuracy improvement. The second proposed system aims to apply k-means clustering to improve the KNN algorithm by enhancing the neighboring point selection. In the proposed method, k-means clustering algorithm groups nearest neighbors according to their distance to mobile user. The evaluation results indicate that the performance of clustered KNN is closely tied to the number of clusters, number of neighbors to be clustered and the initiation of the center points in k-mean algorithm. The third system aims to improve the KNN algorithm by integrating a short term memory (STM) where past signal strength readings are stored. In this proposed approach, the signal strength readings are refined with the historical data prior to comparison with the environment's radio map. The results indicate that the performance of enhanced KNN-STM outperforms the KNN algorithm. Moreover, as an application, the proposed location sensing system is utilized in a location-aware system that accesses patient records.

ÖZET

YER BELİRLEME ALGORİTMALARININ İYİLEŞTRİLMESİ İLE ASG-TABANLI KAPALI ORTAMLARDA POZİSYONLAMA SİSTEMLERİNİN GELİŞTİRİLMESİ

Mobil teknolojilerdeki gelişim, kullanıcıların aletlerle ve diğer kullanılarla olan etkileşimidi değiştirdi. Bu yeni etkileşim yöntemleri, akıllı algılama yetenekleri, içerik, yer ve hareket sensör kullanımları ile sunulmaktadır. Bu doğrultuda, bina içi yer tespiti radyo (Wi-Fi, Bluetooth, GSM vs.) sinyal güçleri ölçülüp, yakın komşular belirlenerek gerçekleştirilmektedir. Alınan sinyal gücüne (ASG) bağlı K Yakın Komşu (KYK) algoritması bu amaçla en sık kullanılan yöntemdir. Bununla beraber, radyo sinyallarındaki dalgalanmalar ASG tabanlı KYK algoritmasındaki en önemli sorunlardan biridir. Dalgalı sinyal güçlerini yer belirme işleminde kullanmak ise hatalı sonuçlara sebebiyet vermektedir. Bu çalışmada, ASG tabanlı KYK algoritmasının yer belirleme netliği, sinyal dalgalanmalarının etkileri elemine edilerek iyileştirilmesi hedeflenmiştir. Bu amaç doğrultusunda üç ayrı iyileştirme yöntemi, var olan sisteme uygulanmıştır. Birinci sistemde, KNN algoritmasının netliği kablosuz örgü ağlar kullanılarak arttırılmaya çalışıldı. Bu yaklaşımda, mobil aletler yer bilgilerini ve toplanan sinyal güçlerini kendi aralarında paylaşarak yer belirlenmesine bulunur. Fakat bu sistemle hedeflenen performansı kazancına elde edilemedi. İkinci sistem, k-means kümeleme yöntemini uygulayarak KYK algoritmasındaki komşu seçim sürecini iyileştirmeyi hedefler. Değerlendirmeler, kümelenmiş KYK algoritması performansının, küme sayısı, kümelenecek komşu sayısı ve küme merkezlerinin başlangıç değerleri ile sıkıca bağlı olduğunu gösterdi. Üçüncü sistem ise, geçmiş sinyal gücü okumalarının tutulduğu kısa dönemlik hafızanın KYK algoritmasına eklenerek performans arttırımını sağlamak amaçlanmıştır. Bu sistemde, sinyal güçleri ortamdaki sinyal haritası ile karşılaştırılma yapılmadan önce, geçmiş sinyal güçleri ile zenginleştirilir. Analizler, geliştirilen bu yeni KYK algoritmasının, normal KYK algoritmasından daha iyi sonuçlar verdiğini gösterdi. Bunlara ek olarak, önerilen iç ortamlarda yer berlirleme tekniği, Hasta Bilgilerine Konum Bilgisi ile Erişme sistemi içerisinde konum belirleme methodu olarak kullanılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

α	Threshold value for signal strength variation
ß	Threshold value for each APs
ρ	Maximum size of short term memory
A-GPS	Assisted GPS
AOA	Angle Of Arrival
AP	Access Point
CF	Compact Framework
EPE	Ekahau Positioning Engine
FTP	File Transfer Protocol
GPS	Global Positioning System
KNN	K Nearest Neighbor
KNN-STM	K Nearest Neighbor with Short Term Memory
KWNN	K Weighted Nearest Neighbor
LBS	Location Based Service
m	meter
MSS	Mobile Support System
MU	Mobile User
NNSS	Nearest Neighbor in Signal Space
OS	Observation Stack
PDA	Personal Digital Assistant
RF	Radio Frequency
RFID	Radio Frequency Identification
RP	Reference Point
RSS	Received Signal Strength
SS	Signal Strength
STM	Short Term Memory
TOA	Time Of Arrival
TDOA	Time Difference Of Arrival
Wi-Fi	Wireless Fidelity (IEEE 802.11a/b/g wireless networking)
WLAN	Wireless Local Area Network

1. INTRODUCTION

Mobile information technologies have been improving rapidly during the last two decades. Nowadays the handheld devices can provide most of the functionalities of desktop/laptop computers. Ease of use (touch screen, small-size), merged functionality as being both a mobile phone and a computer, and the eye-catching designs make these devices the most popular for users who want to carry their computers in their pockets. That’s why the number of companies that focus on developing services for mobile devices and academic studies in this area has increased remarkably. So far, numerous services have been developed aiming to increase usability, efficiency, intelligence of handheld computers. Especially, context-aware systems aim to accomplish certain tasks according to the contextual information captured by the device (like temperature, location, noise, etc.). Accordingly, Location Based Services (LBS) concentrates on the services that can be provided based on the location information of the device.

Hence, WLAN-based location sensing, and especially fingerprinting technique, has been accepted as a viable solution for indoor location identification where global positioning systems fall short [1]. RSS-based indoor location sensing utilizes the radio signal strength for the location estimation. The strength of the signal gives information about distance to the transmitter where signal is emitted. If this strength value is high, it implies that the transmitter is close to the receiver. RSS-based location sensing systems are designed bearing in mind this vital information. The most common algorithm adopted for RSS-based location sensing is K Nearest Neighbor (KNN), which is one of the most popular fingerprinting techniques. KNN algorithm calculates K nearest neighboring points to the Mobile User (MU) and determines the coordinates of user according to the neighbors’ known coordinates.

As an example, Altintas [2] developed a location-aware system for indoors in his under-graduate term project. In this study, the positions of the mobile users were determined by RSS-based location sensing approach which utilized the KNN algorithm. This study pointed out that RSS-based fingerprinting technique should be enhanced to improve the accuracy of the positioning. Altintas' location sensing system, which is summarized in Chapter 3, is utilized as base system in this Master Thesis.

The analysis on the Altintas' location sensing system and literature research point that severe fluctuations in received signal strength is one of the crucial problems in RSS-based indoor positioning systems. Even at a fixed location, the Signal Strengths (SSs) received by a mobile device at different times have large discrepancy. This makes evaluation via signal strengths unreliable. Hence, the device may select group of neighbors that are not actually nearest neighbors in KNN algorithm and this will definitely lead to an error in location estimation. In such a scenario, when the nearest neighbors are miscalculated, the location of the mobile user is determined inaccurately since the coordinates of the mobile user is calculated according to the nearest RPs. Therefore, in order to improve the accuracy of the RSS-based fingerprint technique, a solution should be brought to this issue.

The objective of this thesis is to improve the RSS-based indoor positioning system by enhancing the location sensing algorithms. Accordingly, the aim is to improve the performance of the base location sensing system by developing three novel systems. In the first proposed system, the accuracy of the fingerprinting algorithm is attempted to be improved by exploiting wireless mesh network capabilities. This approach is to share the location data among devices and improve the precision of the estimation. The second proposed system aims to improve the KNN algorithm by enhancing the neighboring point selection by applying k-means clustering approach. In this system, k-means clustering algorithm will group nearest neighbors according to their distance to the mobile user. Then the closest group to the mobile user will be utilized to calculate the MU's location. The third proposed system aims to improve the KNN algorithm by integrating a short term memory (STM) where past signal strength readings will be stored. In this system, the signal strength readings will be refined with the historical data prior to comparison with the environment's radio map. This thesis also presents an application of our proposal, which is a location-aware electronic health record system, and its evaluation. This system senses the location of the physician by utilizing enhanced fingerprinting technique, and retrieves the relevant patient's medical data on to the physician's mobile device.

Accordingly, the structure of this thesis report is as follows: Chapter 2 introduces the keystones and main methods of indoor positioning techniques, while Chapter 3 summarizes the base RSS-based fingerprinting system developed as the under-graduate term project. Chapter 4 describes the first enhancement on the base system; mesh network aided indoor positioning. Chapter 5 expands on the second proposed enhanced fingerprinting via k-means clustering while Chapter 6 presents the third proposed system which is KNN algorithm with Short Term Memory (KNN-STM). An applied system – Location-aware access to patient records – is detailed in the Chapter 7. Finally, Chapter 8 includes overall discussion of the proposed systems and the report is concluded with a discussion about the possible feature work in Chapter 9.

2. BACKGROUND AND RELATED WORK

There are several methods that could be used to sense (or calculate) location. Today, the most popular location detection method is utilizing the Global Positioning System (GPS); however this technology is limited to outdoor environments, since GPS receivers requires a clear view of the sky to calculate its location. Sun et al. [1] compare network-aided positioning techniques for both indoors and outdoors, such as GPS, Assisted-GPS (A-GPS), outdoor cellular network positioning, ad-hoc network based and WLAN-based positioning. They point out that GPS requires long acquisition time, high power consumption and high unit cost, and more importantly GPS is not suitable for indoor positioning. To overcome these limitations, other techniques and technologies such as ultrasound, infrared, Radio Frequency Identification (RFID) tags and radio signals have been developed and evaluated for indoor location sensing. Hazas et al. [3] developed a location detection system which utilized broadband ultrasound to determine the position of the user. In order to achieve this, a special receiver and transmitter hardware named Dolphin (Figure 2.1) was designed. Dolphin initially observes the distance between the mobile user, who carries a transmitter, and the receiver and then detects the mobile user’s position using the measured distance in signal space.

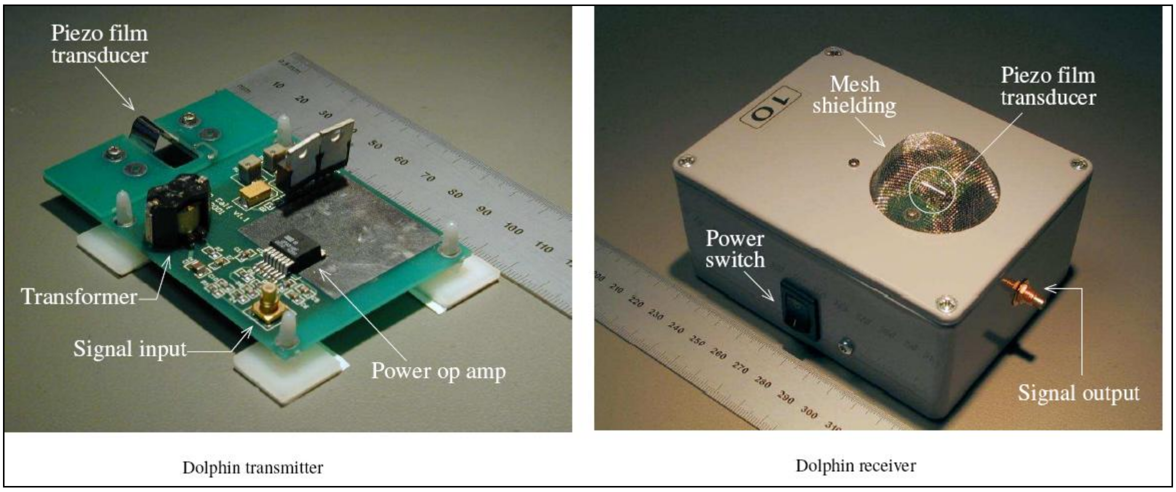


Figure 2.1. Dolphin transmitter and receiver [3]

From a different perspective, Want et al. [4] proposed an indoor positioning system utilizing the infrared signals. In this study, a tag in the form of an ‘Active Badge’ (Figure 2.2) that emits a unique code for approximately a tenth of a second every 15 seconds (a beacon) has been developed, which determines the location of an individual automatically. These periodic beacons are picked up by a network of sensors placed around the host building on known locations. A master station, also connected to the network, polls the sensors for badge ‘sightings’, processes the data, and then makes it available to clients in a visual form. A mobile user carrying an active badge emits infrared signals while s/he moves around the building. The network sensors pick up the signals that contain a unique code for every mobile user. Finally, master station detects the location of the mobile user based on the known location of the network sensor, which received the infrared signals from the active badge.

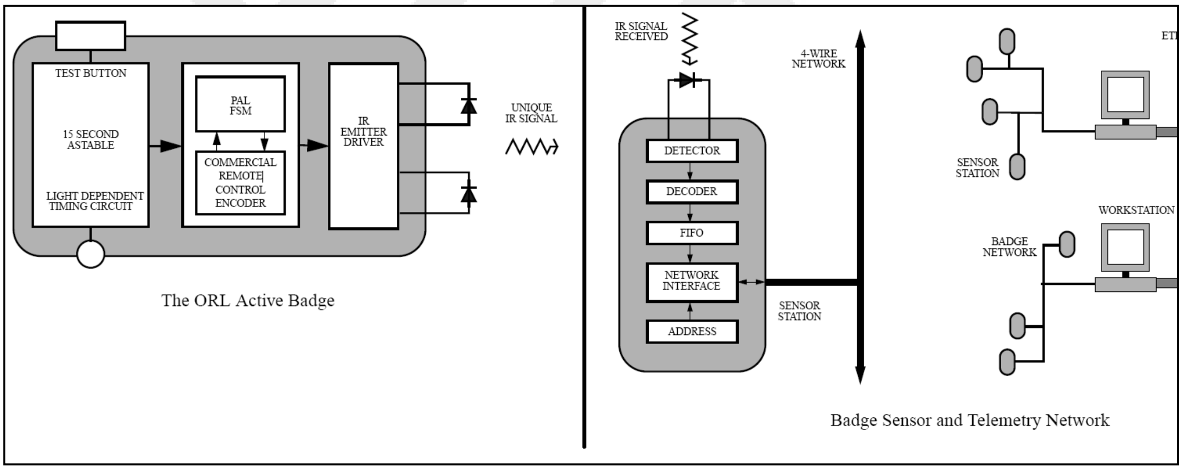


Figure 2.2. Active Badge [4]

Ni et al. [5] designed a location sensing prototype system, named LANDMARC that uses RFID technology for locating objects inside buildings. In this work, they point out that LANDMARC - Location Identification based on Dynamic Active RFID Calibration - improves the overall location sensing accuracy by utilizing the concept of reference tags – see Figure 2.3 for RFID readers and tags used in the LANDMARK project. Accordingly, in order to detect the RFID tags in the environment, RFID readers were placed known locations which are called Reference Points (RPs). After that, the nearest readers to the

RFID tag are determined and then the position of the tag is calculated through K-Nearest Neighbor algorithm.

Figure 2.3. The RFID reader and tag used in prototype system [5]

Bahl et al. [6] worked on methods of determining the mobile users' position based on the radio signals in the buildings. The idea in this work is to determine the mobile user's coordinates according to the received radio signals' characteristics. These characteristics can be strength of the signal, angle of signal arrival or time difference of signal arrival. In this study, strength of signal characteristic was utilized to calculate the distance between the Access Points (APs) and the mobile users' handheld device. The signal strength received by the device is high when the device is near to the AP; signal strength is low when it is far away from AP. Based on this radio signal property and mathematical formulas/algorithms, mobile users' position was calculated.

All these technologies - infrared, ultrasound, RFID and special radio signals - require extra hardware components or hardware modifications which make the development of location detection systems hard. Nevertheless, among all these technologies, due to its penetration and availability, IEEE 802.11b/g/n (Wi-Fi) radio signals and access points (APs) have been the most popular. Bahl et al. [6] developed one of the first location-aware systems over a Radio Frequency (RF) local-area wireless network (i.e. WLAN). Their service, which represented that WLAN signals could be utilized to detect the mobile user location with approximately 3 meters tolerance, was called RADAR.

The user community of IEEE 802.11 a/b/g/n wireless networks (Wi-Fi) is in the increase in recent years. WLAN provides local wireless access to network architecture by taking the advantage of the IEEE 802.11. It is apparent that, WLAN was not designed and implemented with location sensing purposes in mind. However, in the recent years there is abundance of research on radio signal strength (SS) from access points (AP) or stations which also include WLANs. These studies point out that it possible to infer the position of a device with signal strength information from visible APs [6,7,13]. In Bodker et al. [7] study, while someone was walking around the hallway of one building, SS was recorded every second for four different APs. These records (Figure 2.4) point out that signal strengths could be utilized to calculate how far mobile user stand from APs since the signal strength changes according to the distance between the mobile user and APs.

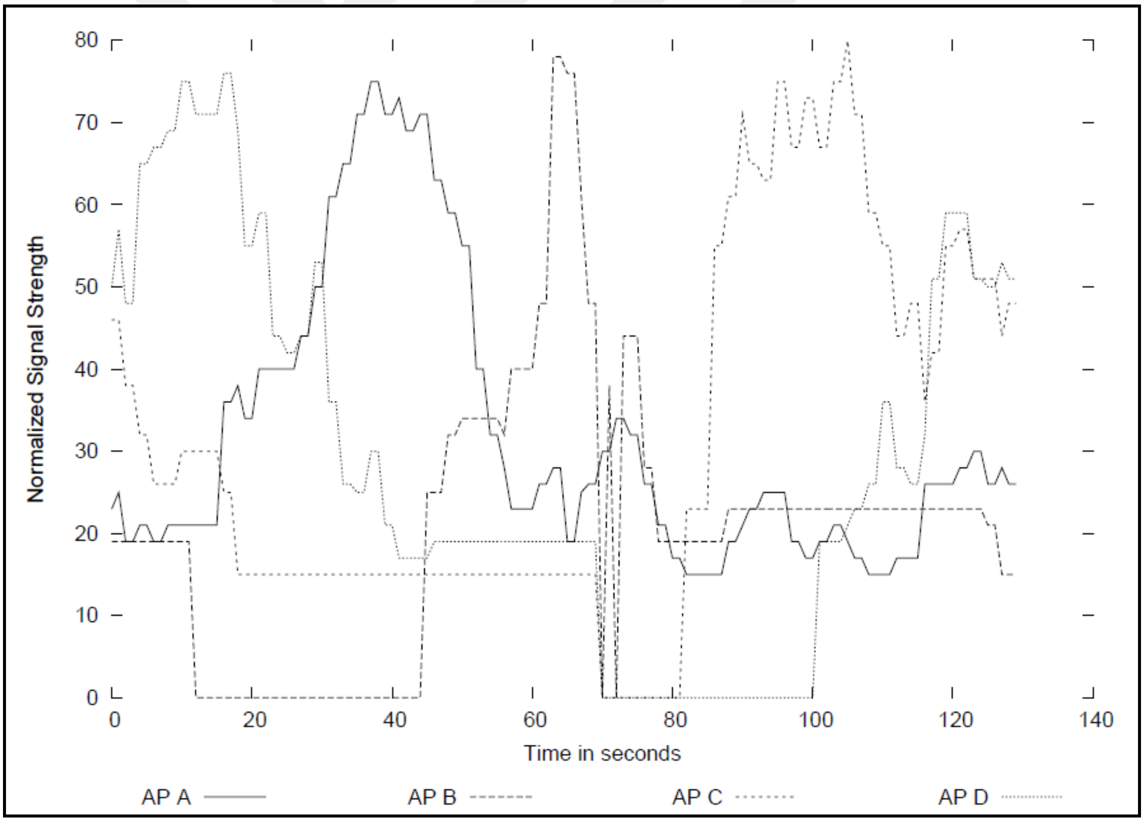


Figure 2.4. Signal strength readings vary with location of the device [7]

Various Wi-Fi-based approaches that use radio frequency signal to measure mobile user's distance to APs have been implemented and tested [6-18]. Namely, these approaches are

Time-of-Arrival (TOA) [8]/Time-Difference-of-Arrival (TDOA) [9], Angle-of-Arrival (AOA) [10] and Received-Signal-Strength (RSS) [11-18].

Llombart et al. [8] developed a Wi-Fi based location detection system by utilizing Time-of-Arrival (TOA) approach. The use of TOA measurements for Wi-Fi positioning is a promising alternative. TOA is time that the radio signal spends from transmitter to receiver. In this technique, distances between the MU and several APs are estimated from TOA measurements which utilize Wi-Fi frames. Then, the MU's position can be calculated by making use of a trilateration or tracking algorithms. Geometrically, each obtained distance provides a circle centered at the corresponding AP, on which the target device must lie. By using at least three APs to solve ambiguities, the target's position can be obtained as the intersection of the circles.

Yamasaki et al. [9] developed a TDOA location system based on IEEE 802.11b, including several techniques such as access point (APs) synchronization, leading edge timing observations, and averaging with diversity. Their measurements are based on the time difference of signal transmitted through the AP to the MU and then back to the AP. In order to achieve this, AP was modified to measure the time of the signal movements. Their findings showed that their system could detect the mobile user's location with 2.4 meters tolerance.

From a different perspective, Yang et al. [10] developed a location detection system based on Wi-Fi where they used angle of arrival (AOA) of the received signals to determine location of the mobile use. They developed a hybrid system by merging the TDOA and AOA method. However, the measurement and calculation operations in this system were very complex which made the implementation and development of such application very hard.

According to the literature, implementing TOA, TDOA and AOA is cumbersome and costly tasks, since they all require hardware and/or software modifications on APs. On the other hand, RSS-based location detection can be implemented without any modification on the AP end; furthermore, collecting RSS data is a straightforward task to perform. Currently, the most esteemed solutions for indoor positioning employ Received Signal

Strength (RSS) – based techniques. There are two well-known and popular techniques that make use of RSS. These are Trilateration and Fingerprinting, where utilize RSS observations to detect user’s location.

It will be helpful to explain these techniques briefly. *Trilateration* observes the distance of mobile user to the three or more known locations and then determine the Mobile User (MU) position based on these distances. By exercising the ability to measure the distances from the MU to each Base Station (BS), trilateration makes use of the hypothetically drawn circles (Figure 2.5). Each circle has with a BS as its center, and its radius is equal to distance from MU to the corresponding BS. Intersection of these circles gives the location of the MU. In the wireless home network context, this approach requires that locations of the APs should be known since they are used as the Base Station and a propagation model should be determined to calculate the distance between the AP and MU.

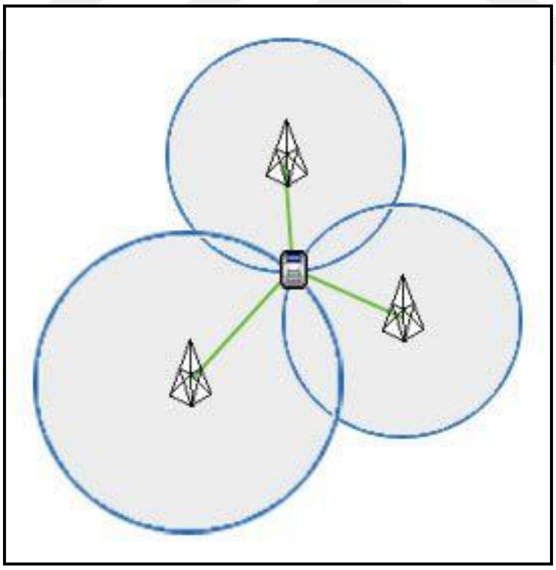


Figure 2.5. The case of trilateration

In fingerprinting technique, signal strength observations from a set of access points are used to determine the nearest fingerprints which will be used to identify a specific location. Thus, a location fingerprint database or a radio map should be constructed before trying to estimate the location of MU.

There is plenty of work that rely on Wi-Fi signals to determine location. One of them is developed by Ekahau [11]. Ekahau is a company selling a Wi-Fi-based positioning system called Ekahau Positioning Engine (EPE). The EPE can be used to determine the location of Wi-Fi enabled devices such as laptops, PDAs and similar devices. The EPE relies on the fingerprinting algorithm. Another project about positioning system is the Skyhook Wi-Fi positioning system [12]. The Skyhook wireless provides no detailed information on how their positioning system works except that it is based on a database containing the locations of the APs. It tries to compensate for different signal propagation model for each AP hence it uses trilateration. As mentioned earlier, one of the mile stones projects in location sensing is RADAR [6]. This project results point out that usage of signal strength information to determine a position is viable. Furthermore, their comparison findings on fingerprinting and trilateration highlight that fingerprinting gives much more accurate results – fingerprinting yielding a median resolution of 2-3 meters and trilateration yielding 5 meter median resolution.

In that sense, trilateration has some disadvantages because it may not be possible to get absolute coordinates of APs. Furthermore, it also requires knowledge of radio signal propagation model to map and match signal strength values to distance values. On the other hand fingerprinting does not rely on a propagation model for positioning and does not need to know the position of APs. In addition, since fingerprinting technique has more accurate results than the trilateration model, it is used much more factor.

Another comprehensive study on fingerprinting technique was performed by Li et al. [13]. In this study, Wi-Fi based positioning systems were developed with both trilateration and fingerprinting techniques. The finding of this study is in line with RADAR's [6] and confirms that fingerprinting algorithm outperforms than the trilateration technique. Another important result of this study is the optimal value of the K in KNN algorithm. Trials were performed with various K values – 2, 3, 4, 5 and 6 – and best accuracy results were obtained with the K value of 4.

Fingerprinting employs either deterministic or probabilistic method to determine the user's position. Deterministic method calculates the distances between the MU and the reference points in signal space. Then nearest reference points are utilized to sense the MU's location. On the other hand, probabilistic method makes a comparison between the probabilistic distribution of signal strengths and real-time signal strength readings at reference points. Then best matching fingerprints according to the distribution function results are selected to determine the mobile user position. Lin et al. [14] performed a performance comparison between these methods. Based on their analyses and experiments, KNN algorithm – a deterministic method – provides the best overall performance for fingerprinting-based indoor positioning. Hence, we focus on KNN-based fingerprinting technique in this study.

Altintas [2] implemented a prototype location-aware system for indoors in his work. In this study, the positions of the mobiles users were determined by RSS-based location sensing approach which utilized the KNN algorithm. His working prototype test results showed that RSS-based fingerprinting technique could be improved to decrease the mean distance error – i.e. improve the accuracy of location recision. This Master Thesis was inspired by that work.

Similarly, Taheri et al. [15] developed RSS-based software-only, platform independent tool for tag-less location sensing on 802.11 infrastructure WLANs for indoor environments, called LOCUS. Their study showed that in-building interference and AP occlusion from certain locations effect signal strengths which cause poor RSS data for positioning. Thus, this worked showed that signal strength variation reduce the precision of the location sensing algorithm.

As mentioned before, fingerprint database, which is constructed for the purpose of nearest neighbor calculation, is the one of the main parts of the fingerprinting technique. However, the fingerprints that construct the fingerprint database or radio map may change with time when environmental changes occur. Re-generation of fingerprints for all locations to maintain an up-to-date RSS database incurs high operational cost, which is impractical in dynamically changed environment. In order to overcome this issue, various studies have been performed. One of them was performed by Michael et al. [16]. In this work, the fingerprint database was generated automatically which saved operational costs. This method requires Medium range (1-2 meters) RFID tags and a person who carry a specific device which consists of a RFID reader and a special Wi-Fi scanner. When the person is moving around the area of interest, the specific coordinates are detected by RFID reader and unique fingerprint for that coordinates are automatically generated by the special Wi-Fi scanner. This proposed system significantly simplifies the deployment of fingerprint based indoor positioning system in the buildings.

Koweerawong et al. [17] also worked on fingerprinting databases. They developed a method to estimate the RSS fingerprint of a specific location from a set of neighboring re-measured RSS fingerprints, called which was “feedbacks”. The proposed method searches for new feedbacks and some old RSS fingerprints in the cut-off area and then applies plane-interpolation to calculate the new RSS fingerprint for a specific location. Re-measured RSS fingerprints (feedbacks) provided better positioning correctness in this study.

The literature reviews shows that the main drawbacks of KNN-based fingerprinting technique are the interference of signals in the environment, which decrease the quality of the signal strength, and frequent constructional changes in the building, which ruin the radio map of the building i.e. fingerprints. These issues are the main cause of poor precision of a KNN-based location detection system. Therefore, various studies, which aim to improve the accuracy of location estimation, are conducted to overcome these issues. Accordingly, in this study, the performance of KNN-based fingerprinting technique is attempted to be improved extensively. For this purpose, three separate enhancement approaches to the existing base system are proposed. First approach will use the help of Wireless Mesh Networks (WMNs) to provide extra environmental data to KNN algorithm. Second approach will apply k-means clustering algorithm to enhance the accuracy of the KNN-algorithm and the third approach will utilize a short term memory to store previous RSS observation, which will utilized in the tuning process of estimated location. All three approaches aim to improve the performance of KNN algorithm. Therefore, prior to the details of these approaches, the methodology and findings of Altintas’s under-graduate project, which implements the KNN-algorithm, is summarized in the following chapter.

3. BASE SYSTEM: RSS-BASED FINGERPRINTING SYSTEM

As mentioned in the background chapter, this master thesis was inspired by the graduation project of Altintas [2], which was undertaken in 2009. In the graduation project, RSS-based fingerprinting was developed by applying the KNN algorithm. This project, which will be called base system, is the origin of this Master thesis. This chapter describes the methodology, implementation and evaluation of the base system.

3.1. METHODOLOGY OF THE BASE FINGERPRINTING SYSTEM

RSS-based fingerprinting technique is composed of two phases: Training (Offline) phase and Tracking (Online) phase.

3.1.1. Training (Offline) Phase

The objective of training phase is to build the fingerprint database which will be used in the tracking phase. In this phase (Figure 3.1a), signal strengths from APs are collected at pre-identified locations, which are called reference points (RPs). At this stage, it is vital that RPs are evenly and homogenously distributed in target area, since the mobile user's location is determined based on the surrounding RPs.

At each RP, sufficient number of observations is carried out in order to reveal the characteristic feature of signal strength. The collected observation set is in the form of **O** = $\{S_1, S_2, \ldots, S_n\}$ where S_i denotes the signal strength obtained from AP_i. Through the observations, median signal strength values of APs at each RP are calculated and then inserted into fingerprint database, which is called radio map of the environment.

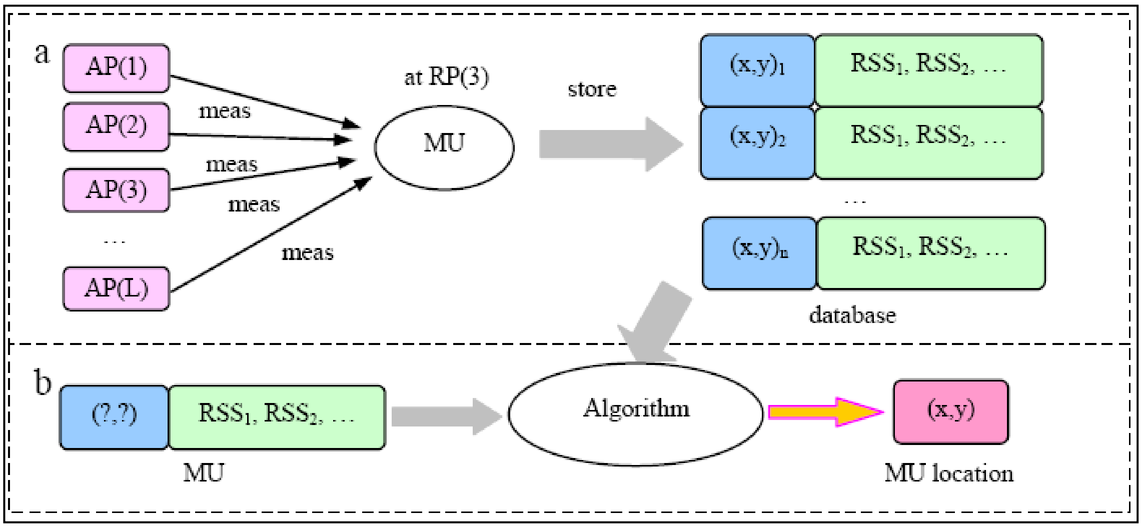


Figure 3.1. Two phases of fingerprinting: (a) training phase and (b) positioning phase

3.1.2. Tracking (Online) Phase

In the tracking phase (Figure 3.1b) MU's surrounding AP SSs are compared with the fingerprint database, which is generated in the training phase, to identify the best matching RPs. These real time SS readings are acquired by making a few observations consecutively. In our implementation, reading number was set to a mediocre value of three, so that the observation requests do not overwhelm the wireless adaptor and at the same time collect enough data to be normalized. The steps of this operation are as follows:

- Make three consequent observations and insert them into observation stack (OS). // **OS** = $\{o_1, o_2, o_3\}$ where OS size is equal to 3 (Figure 3.2).
- Calculate median signal strength value for each AP through three observations and generate real time signal strength vector (SS_{rt}). // $\mathbf{SS_{rt}}$= $\{s_1, s_2, .., s_n\}$ where total number of APs is n.
- Use SS_{rt} to compare and match with fingerprint database.

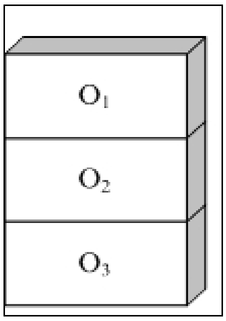


Figure 3.2. Observation stack of KNN

As mentioned in the background – chapter 2 –, fingerprinting technique generally utilizes either deterministic or probabilistic method in the tracking phase. Deterministic method calculates the distance between the mobile users and the Reference Points (RPs). Then a matching algorithm is applied to find the most appropriate RPs based on the calculated distance. Accordingly the coordinates of the mobile user is calculated based on the matched RPs. On the other hand, probabilistic method utilizes real time signals observations in the building for the purpose of matching operation. This matching operation utilizes distribution functions, which are generated based on the signal propagation. Then, the user's location is determined by the fingerprints, which are computed with the help of the distribution functions. Lin et al.'s study [14], which is detailed in the background chapter, points out that deterministic method, and especially K-Nearest Neighbor (KNN) algorithm, yields the best accuracy results when used for RSS-based indoor positioning. Therefore, KNN algorithm was chosen to implement the location detection system proposed in the graduation project.

K-Nearest Neighbor (KNN) algorithm is one of the basic deterministic algorithms, which is popularly used for RSS based indoor positioning. This algorithm attempts to reveal RPs with the minimum distance to mobile user in signal space. KNN algorithm utilizes Euclidean distance (Eq. 3.1 where q = 2) in signal space to calculate distances between the MU and RPs. Then RPs with the minimum distance to the mobile user are determined and utilized in the calculation of the MU's coordinates.

Signal distance between mobile user and RPs are calculated by applying Eq. 3.1. This equation makes use of real time signal readings – $\mathbf{SS_{rt}}= \{s_1, s_2, .., s_n\}$ – and the SS vectors –$\{S_1, S_2, .., S_n\}$ –, which are observed in the offline phase.

$$L_q = \left(\sum_{1}^{n} |s_i - S_i|^q \right)^{1/q} \tag{3.1}$$

The quantity $\mathbf{L_q}$ is a positive real value. A lower $\mathbf{L_q}$ value indicates a smaller difference between the compared vectors and a higher value indicates bigger difference. Therefore, the location of the MU is calculated according to the smallest $\mathbf{L_q}$ values.

In this process, as required, KNN algorithm selects K minimum $\mathbf{L_q}$ values which correspond to nearest K RPs. Using this data, the average of these RPs' coordinates is calculated. This calculation result translates as the coordinates of the mobile user. Li, B. et al. [18] has conducted an experiment to determine the most appropriate value for the K parameter in the KNN algorithm. Furthermore, an experimental setup was used to evaluate the affects of RP number to the precision of the sensed location. In this study, five APs were located to known locations and 132 RPs were generated as fingerprints. Location estimations were performed at 30 distinct unknown positions. To investigate the effect of the granularity, the number of RPs was intentionally reduced to 99, 66, 33, and 16. However, the RPs were spread as evenly as possible in the test area. Hence, five fingerprinting databases were generated each having 132, 99, 66, 33, 16 RPs respectively. KNN algorithm with K values of 2, 3, 4, 5, 6 and K Weighted Nearest Neighbor (KWNN) algorithm with K values of 2, 3, 4, and 5 are tested at 30 distinct test points with different fingerprint databases. The difference between KNN and KWNN algorithms is that, while KNN algorithm utilizes the average of the nearest RPs' coordinates directly, KWNN algorithm calculates the weighted average of the nearest RPs' coordinates. Table 3.1 lists all the mean distance errors - precision of the positioning results - , which are computed by the both KNN and KWNN algorithms with different K values and different fingerprint databases. These results point out that when the granularity of the RPs is large, KNN algorithm performs better. Moreover, KNN algorithm with K value of 3 or 4 yield the best positioning results. This indicates that utilizing the two nearest neighbors is not enough (some of the useful information has been ignored) to determine the MU's position. On the other hand, too many nearest neighbors could decrease the accuracy of the estimation. The last important output from this study is that KWNN method slightly improves the accuracy of estimation compared to the KNN method.

Table 3.1. Mean distance error using different algorithm (unit: m) [18]

	NN	2NN	3NN	4NN	5NN	6NN	2WNN	3WNN	4WNN	5WNN
Test1 (132 RPs)	1.75	1.47	1.29	1.23	1.38	1.31	1.49	1.29	1.19	1.31
Test2 (99 RPs)	1.63	1.52	1.38	1.31	1.36	1.39	1.53	1.37	1.27	1.30
Test3 (66 RPs)	1.74	1.47	1.51	1.60	1.52	1.60	1.48	1.44	1.49	1.43
Test4 (33 RPs)	1.78	1.93	1.94	1.72	1.99	2.12	1.79	1.79	1.64	1.75
Test5 (16 RPs)	2.55	2.34	2.65	2.98	3.41	3.99	2.11	2.28	2.45	2.69

For simplicity and estimation accuracy purposes, the KWNN method with K value of 4 was preferred in this base system. The RPs' distances to MU were utilized as weights so that the closest RP has the biggest weight and the furthest RP has smallest weight. Simple mathematical representation of this method is as follow.

$$x = \frac{\sum_{i=1}^{4}(x_i \ . \ d_{4-i})}{\sum_{i=1}^{4} d_i} \tag{3.2}$$

$$y = \frac{\sum_{i=1}^{4}(y_i \ . \ d_{4-i})}{\sum_{i=1}^{4} d_i} \tag{3.3}$$

In the KWNN method, the selected nearest RPs are stored in the list in ascending order based on the distance. x_i x_i represents the x coordinates of the RPs and, y_i y_i represents the y coordinate of the RPs. d_i implies the signal distance between the MU and RPs. As shown in the equation (Eq. 3.2, 3.3), the nearest RPs coordinates (x_1, y_1) is multiplied by the biggest distance (d_4) among the K RPs. The purpose of this operation is to increase the contribution of the nearest RPs to location estimation.

3.2. IMPLEMENTATION OF BASE FINGERPRINTING SYSTEM

This project was developed with .NET Compact Framework (CF) technology. C# was preferred as implementation language. Default .NET CF does not provide a library to observe the signal strengths from the mobile device network adapter. Therefore, a robust shared-source library from OpenNETCF.org [19], which is named Smart Device Framework, was utilized to read the signal values from the mobile device's network adapter. This library was developed by OpenNETCF Consulting to enable developing of feature-rich application such as displaying network adapter properties, discovering nearby wireless access points etc.

This framework contains a single namespace, "OpenNETCF.Net", but a lot of classes for various purposes such as File Transfer Protocol (FTP), network statistics, Bluetooth etc. In the base system project, four classes were used to access the signal strength values from the surrounding APs. These are **Adapter**, **AccessPoint**, **AdapterCollection** and **AccessPointCollection**.

Algorithm 3.1. Usage of class in the smart device framework library

```
public List<Classes.WiFiSignature> readMeasure()
{
   m_adapters = Networking.GetAdapters();
   List<Classes.WiFiSignature> lst = new
List<Classes.WiFiSignature>();
   m_currentAdapter = adapter[0];
   if (m_nearbyAPs == null)
   {
      m_nearbyAPs = m_currentAdapter.NearbyAccessPoints;
   }
   else
   {
      m_nearbyAPs.Refresh();
   }
   foreach (AccessPoint ap in m_nearbyAPs)
   {
      Classes.WiFiSignature entry = new
Classes.WiFiSignature(ap.Name,BitConverter.ToString(
ap.MacAddress),ap.SignalStrengthInDecibels);
      lst.Add(entry);
   }
   return lst;
}
```

The use of these classes is shown in the Algorithm 3.1. In this code snipped, *m_adapters* is an object of AdapterCollection class which includes network adapters on the device. *m_currentAdapter* is an object of Adapter class which includes all available access points and *m_nearbyAPs* is an object of the AccessPointCollection class which includes access points which are accessed currently.

By utilizing these classes, the APs' radio signals are observed from the network adapter of the mobile device. This operation is performed firstly in the training (offline) phase to generate the fingerprint database. In order to achieve this, mobile device is located to the known coordinates. Then, the signals from surrounding APs are observed 30 times by calling the `readMeasure()` method at each position. For each reference point, median value of the APs' signal strengths is calculated and inserted into the local database as fingerprint. The base system's local database was created with Compact SQL Server and included four tables in this study. These tables are Observation and WiFiSignature which store the fingerprints observed in the offline phase (Figure 3.3). Remaining two tables, Room and Coordinate, are utilized in the positioning (online) phase in order to determine the room where the MU exists.

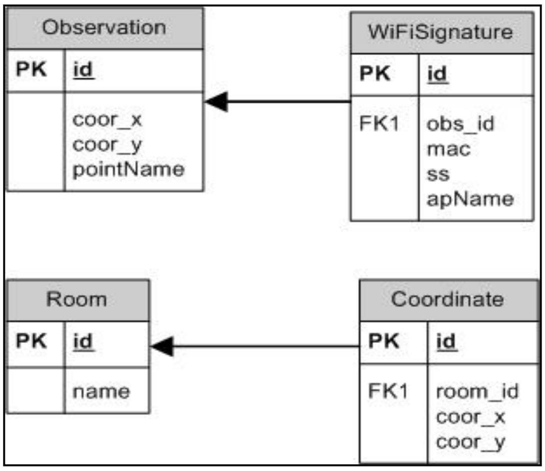


Figure 3.3. Local database model diagram

As mentioned earlier, Observation and WiFisignature tables store information about training phase. Observation table involve the coordinates of the each Reference Point and their name. WiFiSignature table involves the MAC address and the median signal strength value of each access point for all reference points. Matching operation, i.e. the Euclidian distance calculation utilizes this stored information (i.e. fingerprints). On training phase, real-time signal strength observation is obtained by the same function, `readMeasure()`. At unknown locations, this function collects real-time signal strengths. Then these values are utilized in the Euclidian distance equation to calculate the nearest RPs in the building. This equation is implemented in `NNSS()` function in the project (Algorithm 3.2).

Algorithm 3.2. Nearest Neighbor in Signal Space (NNSS) function

```
public List<Classes.NNSSDistance>
NNSS(List<Classes.WiFiSignature> lt)
{
   int i = 0;      int total = 0;       int index=0;
   List<Classes.NNSSDistance> distanceList = new
List<Classes.NNSSDistance>();
   foreach (Classes.Observation ob in obsList)
   {
      i = 0;
      if (ob.list.Count <= lt.Count)
      {
         foreach (Classes.WiFiSignature ssList in ob.list)
         {
            index = indexCalc(ssList.mac, lt );
            if (index == -1)
               continue;
            else{
               total += square(ssList.calSS - lt[index].calSS);
               i++;
            }
         }
      }
      else{
         foreach (Classes.WiFiSignature ssList in lt)
         {
            index = indexCalc(ssList.mac, ob.list);
            if (index == -1)
               continue;
            else{
               total += square(ssList.calSS -
ob.list[index].calSS);
               i++;
            }
         }
      }
   Classes.NNSSDistance entry = new.Classes.NNSSDistance();
   entry.distance = ((Math.Sqrt(Math.Abs(total)))*1)/1;
   entry.obsId = ob.id;
   entry.numberOfAP = i;
   distanceList.Add(entry);
   total = 0;
   }
   return distanceList;}
```

In this algorithm, Equation (3.1) is implemented with $q=2$. *obs* is the list of the RPs in the area and it includes the median SS values from each AP at each RP. On the other hand, *lt* is the list of the signal strength observed from APs at unknown location. These two collections, first one that includes the fingerprints and the other that includes the real-time signal observations, are utilized together to calculate the distance of MU to the RPs.

As mentioned in the previous section, for simplicity and location estimation accuracy purposes, K value was set to 4 in the KWNN algorithm. The algorithm utilizes the distance values as weights. In order to increase the contribution of the nearest RP to the result, the coordinates of the nearest RP are multiplied by the farthest away RP's distance among the K (4) chosen RPs. Similarly, to decrease the effect of the furthest RP, its coordinates are multiplied with the closest RP's distance in the chosen RP group with K (4) members. Hence, this algorithm boils down to Equation (3.2) and Equation (3.3), which both then are used to calculate the coordinates of the mobile user. The results of these equations give us the *x* and *y* coordinates of the MU.

3.3. EVALUATION AND ANALYSIS OF BASE FINGERPRINTING SYSTEM

In the trials, existing WLAN infrastructure in the floor of Yeditepe University department of Computer Engineering building was utilized. This WLAN included three D-Link Wireless APs, one U.S. Robotics AP and one Cisco Aironet 340 series AP. In the development process, the project was tested on three different PDAs. These were Samsung i900, HP IPAQ 6515 and HP IPAQ 5500. The system was developed on Visual Studio 2008 and .NET compact framework.

The indoor positioning system test bed was set up at our university's school building. The test bed was 500 sq. meters (approx. 5381 sq. ft.) area, which was covered by five 802.11b/g APs. In Figure 3.4, 48 RPs are marked with a red cross.

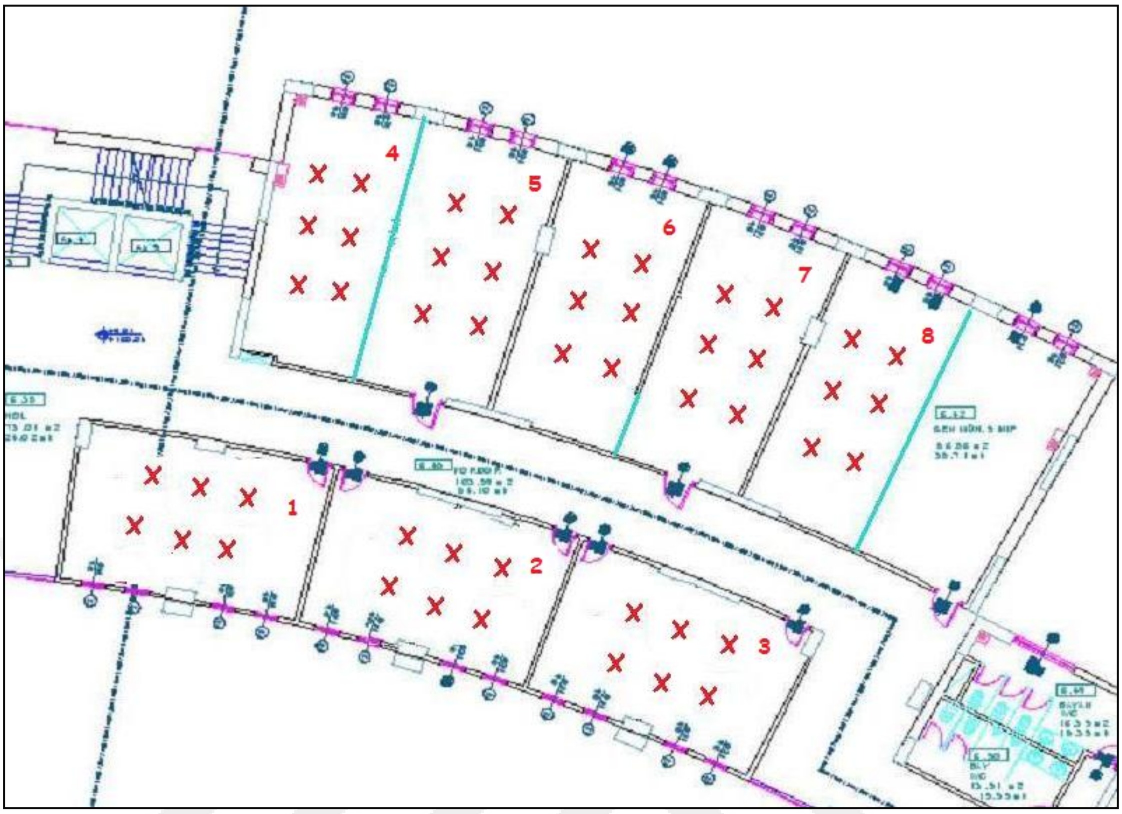


Figure 3.4. Floor plan of the test bed and RPs marked as red crosses

The evaluation of the system showed that using the above algorithm, a mean distance error (i.e. the average distance between the MU real position and estimated position) of approximately 4.5 meters and standard deviation of approximately 3.2 meters (Figure 3.5) can be achieved. These test results also indicated that the location of mobile user can be estimated roughly in the test bed. This error rate can be acceptable by some kind of location-based systems which do not require precise location estimation. For instance, finding the nearest printer in the building to print documents does not require precise estimation. On the other hand, some kind of location-based systems need more precise estimations. For example, accessing the patient records when entering the patient room requires exact location of the user in order to determine in which room he/she exists. In this study, the system was able to determine the correct location of the user only half of the time during our trials.

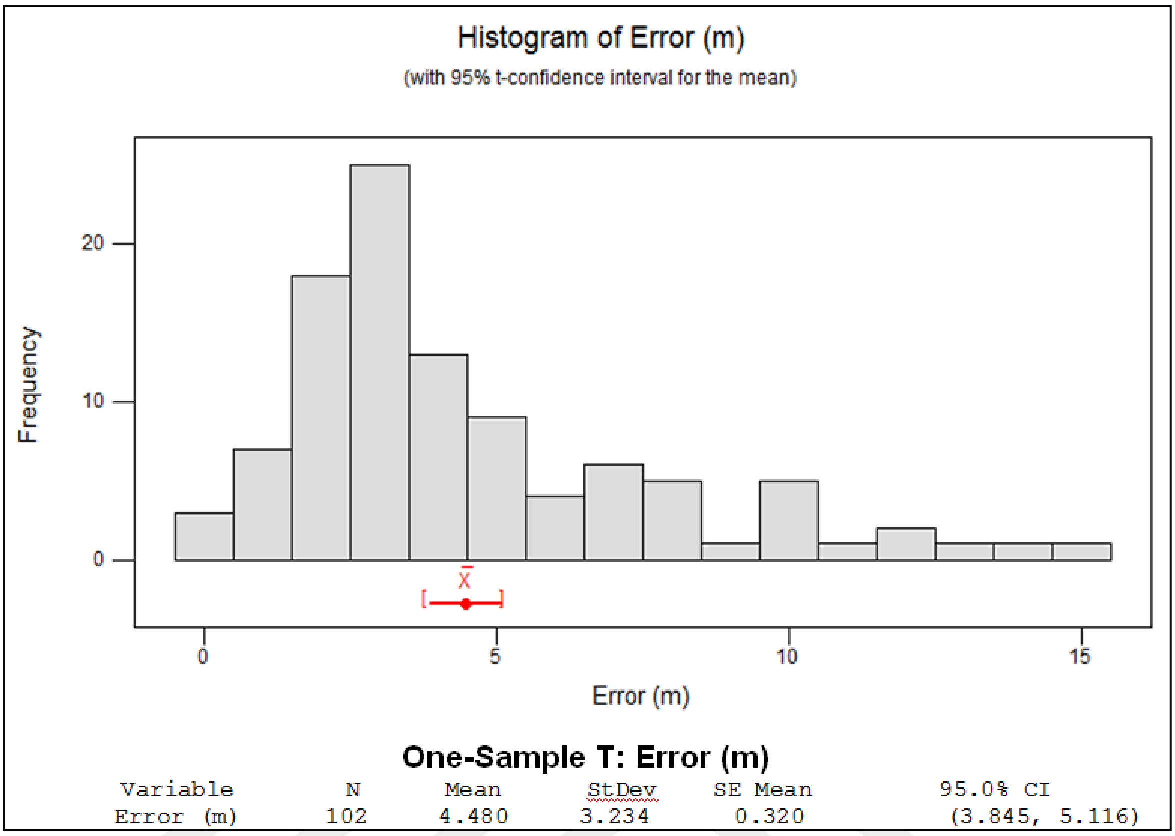
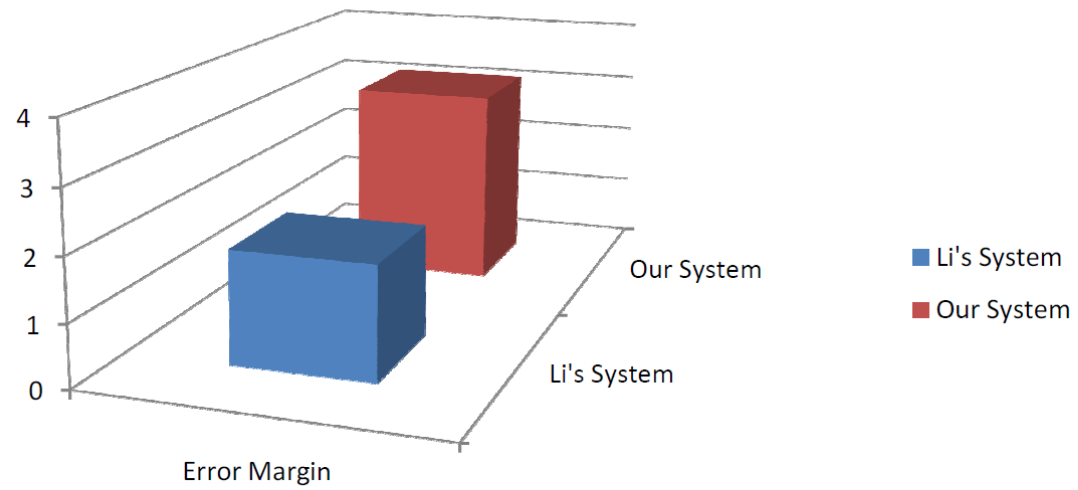


Figure 3.5. Test results

When the test results were compared with the similar studies, which are detailed in the background chapter, it can be said that the obtained results are similar with most of these related works. A closer look to the test results pointed out that the location estimation algorithm implemented in this system produced more precise results in the area where the numbers of available APs were high. Since the existing WLAN infrastructure was utilized during the trials, the distribution of APs in the test bed was not homogeneous and caused extreme fluctuations in the number of accessible APs and SS. The number of available APs could be very low at the some parts of test bed, which this as a return decreased the estimation precision. In these circumstances, the real time signal observations were not enough to run the matching algorithm - KNN. It could be easily derived from here that the quality of location estimation algorithm's output and the precision of the calculated location are directly related to the number of APs available in the surrounding environment.

Moreover, when this study was compared with the Li et al.'s study [13], it was seen that their proposed system provides better location estimation precision which implies a smaller mean distance error. The comparison of these two works can be seen in the Figure 3.6. Both systems utilized the same fingerprinting technique. When the reason of the difference between these two works was investigated, it was clearly seen that the number of RPs in Li et al.'s study is significantly higher than this study. The number of RPs in the test bed is considerably important because, when you increase the granularity of the RPs, you increase the probability of the matching exact nearest RPs in the environment.



* Li, B. et al., 'Indoor Positioning Techniques Based on Wireless LAN', Auswireless Conference 2006, pp 1-7, 2007

Figure 3.6. Compression between our system and Li's system

Further analysis on the trials showed that strength of the Wi-Fi radio signal frequently fluctuates based on the environmental factors. When a radio wave encounters objects in its path, it gets affected by them. For instance, when a person walks through the corridor in the building, or even a microwave starts to work, signal waves oscillate. The most significant factors that affect the propagation of radio waves are reflection, diffraction, scattering, and refraction. These factors highly affect the precision of location estimation by reducing the signals quality received from APs. The reduced signals quality cause miscalculation and false correct results while determining the nearest RPs. In other words, the matching process may conclude with a nearest RP which is not actually the nearest to the mobile user. As a result, this issue decreases the accuracy of overall location estimation.

In order to overcome this issue, enhancements to the matching process between the RSS observations and the fingerprints are proposed in the scope of this Master thesis. The findings and knowhow gains from the base systems lead us to design these enhancements on the KNN algorithm. Accordingly, three new enhancement approaches are designed, developed and evaluated in this study. The first approach attempts to share the estimated location information among the mobile users to provide extra data for the matching (comparison) process. Following chapter details the development phase of this approach and the evaluation results of the prototype.

4. MESH NETWORK AIDED RSS-BASED POSITIONING

In fingerprinting algorithm, the most challenging step is to determine the nearest K neighbors. In this step, the real-time RSSs readings are compared with the ones recorded as fingerprints. However, signal strengths are prone to vary in time due to the environmental blocking and reflecting issues, which causes inconsistencies in the RSS readings. This makes signal strength based evaluations unreliable. As a result, the algorithm may select a group of erroneous nearest neighbors and lead to an error in location estimation. The main purpose of this study is providing a new way to determine nearest RPs wisely. In order to be able to do this, the nearest RPs calculation process is expanded with couple of refinement steps. The new approach attempts to take advantage of other mobile devices' location information and RSS observations in the vicinity.

The proposed system's general architecture consists of three separate architectures that cooperate. First one implements the fingerprinting algorithm, which makes use of the real time signal strengths and previously recorded fingerprints to estimate the device's location. Second one provides the mesh network backbone, so that devices can communicate with each other. Then, the third architecture comes in use, which performs trilateration and tunes the estimated location in the first step. A basic illustration of the system architecture is Figure 4.1.

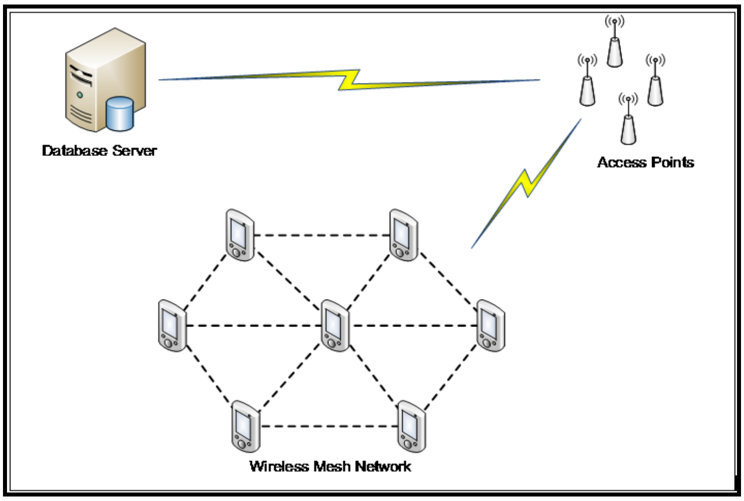


Figure 4.1. The proposed system architecture

Accordingly, the system follows a three-phase estimation scheme. In the first phase, location of MU is estimated by the base fingerprinting technique. This technique is already developed in our base system which is detailed in the Chapter 3. Then, in the second phase, location information and RSS observations of neighbor mobile users are received via WMN. If a mobile device, which has not been localized previously, is requested location information, it calculates its current location with fingerprinting and shares this information. But if a mobile device has already calculated its position with new proposed system, it then shares the tuned results as its location. After extra data is transferred from neighbors, trilateration is performed by utilizing the collected information. The purpose of the trilateration is determining the actual nearest RPs around the mobile user. Therefore, the RPs, which are nearest to the trilateration's result, are assumed as the actual nearest RPs. As a final step, in the third phase, fingerprinting is reapplied to refine MU's location; but this time, it takes nearest RPs in the new estimated region into consideration. After adding the extension to the base system, it is then called 3-Phase Location Estimation (Figure 4.2) since it consists of three phase: first step is, initial fingerprinting; second step is trilateration to get rid of miscalculated nearest neighbor and third is re-fingerprinting.

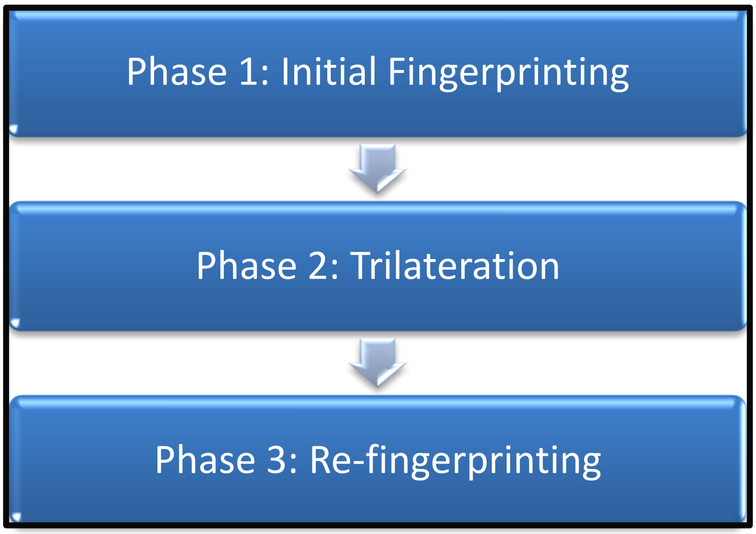


Figure 4.2. 3-Phase location estimation

Main motivation of the project is to apply this system in crowded areas, such as museums, schools, work places, and other public areas. It is assumed that there will be some devices with bad estimations and there will be some with more accurate ones. The objective here is to make use of more accurate estimations to get rid of the bad ones. Hence, the system has to be able to distinguish them.

Our prior experiments have shown that fingerprinting gives more accurate results when the device is close to APs. This is a finding that should be kept in mind. Another thing is, it is believed that enhancement of the fingerprinting accuracy will lead the 3-Phase approach to better estimations. If all the devices have good estimations, this continuous trilateration would end up minimizing the error margin for all of the devices. However, if the accuracy of the initial fingerprinting is poor then tuning operation would distribute faulty and inaccurate location data, which overall will distort the location algorithm outcome of every mobile device.

4.1. RELATED WORK OF MESH NETWORK AIDED INDOOR POSITIONING

In literature there are not many studies about merging Wireless Mesh Networks (WMNs) with Location Based Services (LBS), however, some out of these few do stand out. To begin with, Jimenez [20] proposed a Real Time Location System (RTLS) for WMN in his thesis. The system proposed by Jimenez uses RSS for distance measurement and location estimations. A prototype of this system is presented using off-the-shelf components. Trials with the proposed system show that it is capable of locating nodes with 18% accuracy regardless of the network size.

From a different perspective, Thomas et al. [21] from US Army Communications-Electronics Research Development and Engineering Center (CERDEC) studied with some new practical findings pertaining to vehicular networks, specifically convoy communications and unmanned vehicle control. Besides considering of principles and rules of military convoys, experiments and simulations with military scenarios are performed to improve vehicular networking for the military use in this study. GPS tracing and its effects on throughput improvement and reliability on vehicular communications were evaluated. Another example that targets vehicular environments was developed by Movaghar [22]. In this study, the author designs a universal intelligent vehicular ad-hoc network, where its backbone is constructed with a hierarchical WMN. In the study a novel addressing scheme was also proposed, so that a node’s address indicates its position relative to the leader node. The process of location detection was one of the core objectives of this project, which was ultimately aiming to simplify route-discovery process and minimize the routing overhead.

Although the studies described above are recent, most of them target to improve the basic capabilities, reliabilities and performances of WMS. This is the case because WMN technology is still in its emerging state. Hence, none of the work detailed above is directly aiming to enhance indoor location detection by the use of WMNs.

4.2. MESH NETWORK METHODOLOGY IN POSITIONING

Mesh network-aided RSS-based positioning system's methodology consists of four subsections.

4.2.1. Wireless Mesh Network Routing Protocols

Wireless Mesh Networks (WMNs) are popular especially because of their self-healing and self-organization characteristic features. Mesh clients serve as nodes in WMN. Data is transmitted from one node to another, through the network by multiple hops. The property of multi-hopping comes with the ability to develop various routing protocols for WMNs. Popular routing platforms with various routing protocols are examined and compared by Riggio et al. [23] – see Table 4.1 for a detailed comparison of these WMN routing platforms. Even though Mesh Connectivity Layer (MCL), with its loadable windows driver that implements an interposition layer between the link and the network layer, seems to be the most suitable solution for a mesh application on Windows CE platform. It will not be used in the implementation of the prototype, because the MCL project was 'retired' and not supported any further. On the other hand, however, Optimized Link State Routing (OLSR) Daemon runs on almost every platform and it is supported actively by the developer company and community.

Table 4.1. Comparison of WMN routing platforms [23]

Name	Protocol	Platform	Implementation	Routing layer
AODV-UCSB	AODV	GNU/Linux	Kernel module w/ user space routing logic	3
AODV-UIUC	AODV	GNU/Linux	User space	3
AODV-UU	AODV	GNU/Linux	Kernel module w/ user space routing logic	3
CUWiN	HSLS	NetBSD	Kernel space	3
DSR-UU	DSR	GNU/Linux	User space	3
JAdhoc	AODV	GNU/Linux, Windows, Zaurus	User space	3 3
Kernel-AODV	AODV	GNU/Linux	Kernel space	3
MCL	LQSR (DSR-like)	Windows	Loadable Windows driver	2.5
Monarch Project	DSR	FreeBSD	Kernel space	3
OLSRD	OLSR	GNU/Linux, Windows, MAC OS X, *BSD	User space	3
QOLSR	OLSR	GNU/Linux	User space	3
Roofnet	SrcRR (DSR-like)	GNU/Linux, *BSD	User space	2.5

4.2.2. OLSR Daemon

OLSR Daemon is an open source implementation of the Optimized Link State Routing protocol. It is able to create a mesh network between devices that have ad-hoc support and runs on any wireless network card. A version of it is also available to run with Ethernet devices on PCs. Since it is a layer 3 solution, it is compatible with almost every platform that someone can think of, such as Windows, Linux, Android, etc. This project is developed in C programming language.

There is also another OLSR daemon project which is developed by Moviquity [25]. This application also provides mesh networking ability, and is developed for Windows Mobile and implemented in C# programming language. It comes with a user-friendly graphical user interface (Figure 4.3). It can be deployed on a PDA as a project that has OpenNETCF.Net library within.

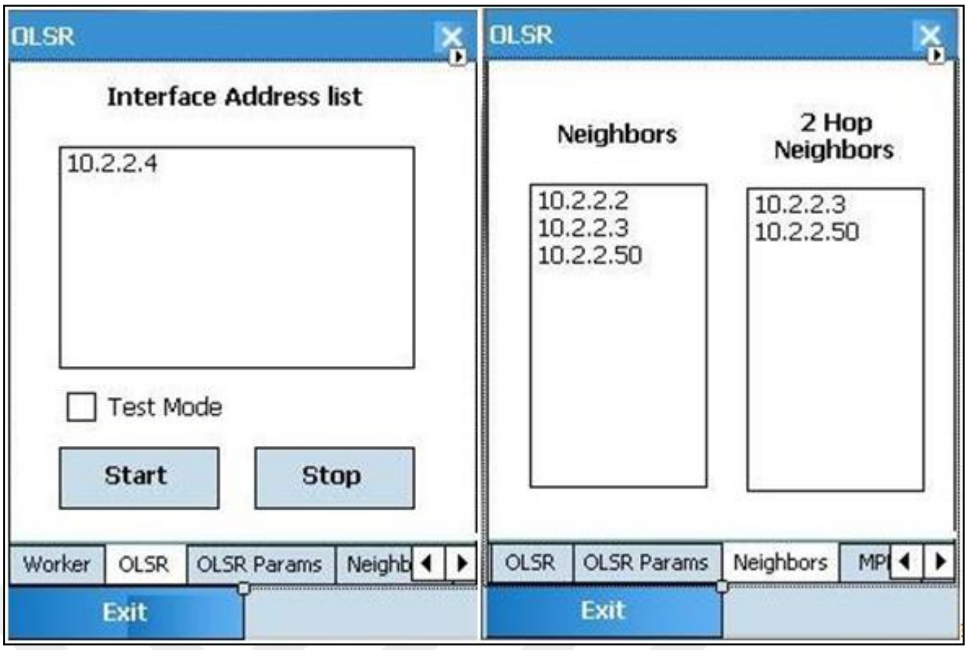


Figure 4.3. Moviquity OLSR daemon sample GUI

4.2.3. UDP Broadcasting

Mesh networks are basically peer to peer (P2P) networks, but offer more services. These services (self-healing, self-constructive, etc.) usually have an effect on the topology and the routing paradigms. In mesh networks the communication principles stay the same as in P2P networks. UDP datagrams are used for data transmission in both P2P and mesh networks.

Since mesh networks are self-constructive, every device joins the mesh network by announcing its request to do so. Accordingly, the device that wants to join the network has to let all devices in its range know about its existence. This kind of announcing scheme in networks is called broadcasting. When a device broadcasts a chunk of data, it’s received by all nodes that are connected to the corresponding broadcast channel. In mesh networks, this broadcast channel happens to be the whole network, since all devices work in ad-hoc mode and are aware of the devices around them. In addition to ad-hoc, mesh network can make use of intermediate nodes to provide a connection between two nodes which are not in each other’s range. So, any broadcast message reaches to every node in the mesh, even though the some of the nodes are not connected to each other directly.

4.2.4. 3-Phase Location Estimation

In this proposed system, firstly the mobile user observes signals strengths from APs in real-time and determines the nearest RPs via the base fingerprinting system. Then, the mobile user collects other mobile users' location information and their RSSs observations through the mesh network. In the next step, the proposed system attempts to execute trilateration to assist mobile user on location estimation process. The locations of the surrounding users are utilized as the center of the circles in the trilateration. The distances between the mobile user and the surrounding users, are the radiuses of the circles for trilateration. As mentioned earlier, the distances are calculated in signal space (signal distance) by using the Euclidian distance equation (Eq. 3.1). However, metric distances are required in the trilateration instead of signal distances. Therefore, the output values of the Euclidean distance equation - the radiuses of the circles - are converted to metric distance (i.e. distance in meters) as it is required for trilateration. In order to make this conversion, initially the ratio between the signal distance and the metric distance is determined. The methodologies employed to calculate the conversion ratio is detailed in the implementation section of this study (Section 4.3.1)

4.3. MESH NETWORK IMPLEMENTATION IN LOCATION SENSING

The implementation of the proposed system is composed of two facets; the adapted trilateration and peer-to-peer communication between the devices, which is described in the following two sections.

4.3.1. Adapted Trilateration

The main purpose of using trilateration in this study is to improve the accuracy rate in identifying the four nearest neighbors. The trilateration approach is customized for this project so that the distances between the MU and three neighbors are calculated based on the conversion ratio. For this purpose, every device calculates its own conversion ratio. To calculate this ratio, the signal distances between the device and RPs are measured by utilizing the real-time signal strength observation and the RPs' fingerprints. Then, metric distances between the device and RPs are calculated according to the device coordinates and RPs coordinates. Finally the correlation coefficient of these two values is obtained as the conversion ratio.

In an ideal world, the ratio between the signal distance and the metric distance would be constant. However, since the signal strengths themselves are not constant and stable for a specific location, the ratio changes from device to device and even from time to time. Therefore, we come up with three different implementation approaches to overcome the instability problem and convert the signal distance to metric distance. In the first one, the mobile device utilizes its own calculated ratio. In the second approach, the mobile device utilizes its neighbor's calculated ratio, which is determined by neighbor using its own RSS readings and location. In the last one, the average of these two ratios is calculated and used as a common ratio on both of the devices.

4.3.2. Peer to Peer Communication

To facilitate the peer to peer communication .NET.Sockets libraries are used in the implementation of the prototype. In the prototype system, by nature every node circulates a join request as soon as it detects that it is in a meshed network environment. Once a node has three or more neighbors, it starts the second estimation phase, in which the location of the node is *tuned* by trilateration. Assuming that the trilateration results are accurate, it moves to the third phase, where it uses fingerprinting. The outcome of the last fingerprinting is considered to be final and exact estimation of the device's location.

4.4. MESH NETWORK-AIDED POSITIONING EVALUATION

The trials were performed in the same test bed with the base system. One of the outcomes of the base system trials was that when the number of accessible APs is high, fingerprinting provides better estimations. For that very reason, due to the high number of AP accessibility, the mesh network-aided fingerprinting trials performed in rooms Number 1, 2 and 3 - see Figure 3.4. Under these circumstances, the mean distance error of the base system was improved to around 4 meters. Accordingly, 3-Phase Location Estimation aimed to take advantage of this refinement. Moreover, in order to compare the proposed and the base system, both estimate the position of the mobile user with the same real-time

RSS reading. And then, these estimated positions are compared with the actual location of the mobile user to obtain their distance errors (i.e. accuracy).

In the following set of evaluations, the signal space (SS) distance and metric distance conversion methods are studied. In this context, this conversion can be made using mobile devices own ratio calculation, neighbor's ratio or by taking the average of both ratios. The findings of 30 trial results are discussed below.

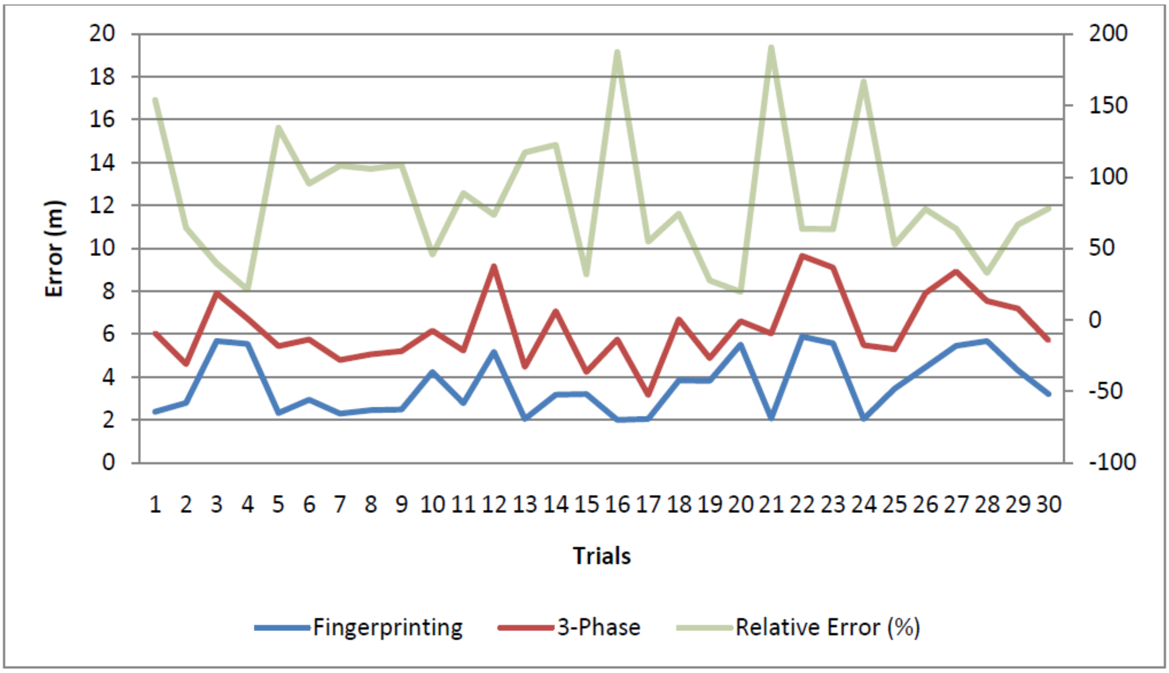


Figure 4.4. Error rates when the system uses the device's own SS/metric ratio

Figure 4.4 shows the error rates of the scenario where the device uses its own SS/metric distance ratio. For this scenario, the mean distance error value is 3.63 meters for the first phase of the estimation, where only fingerprinting algorithm had performed. After three phases were performed, however, it was observed that the mean error value was set to 6.26 meters. This value, of course, is not acceptable for an indoor positioning system as it will surely lead to unwanted outcomes. This figure (Figure 4.4) also contains the relative error graph, where it depicts the difference between fingerprinting and 3-phase approach in percentage. Figure 4.5 highlights the mean error between two approaches in meters.

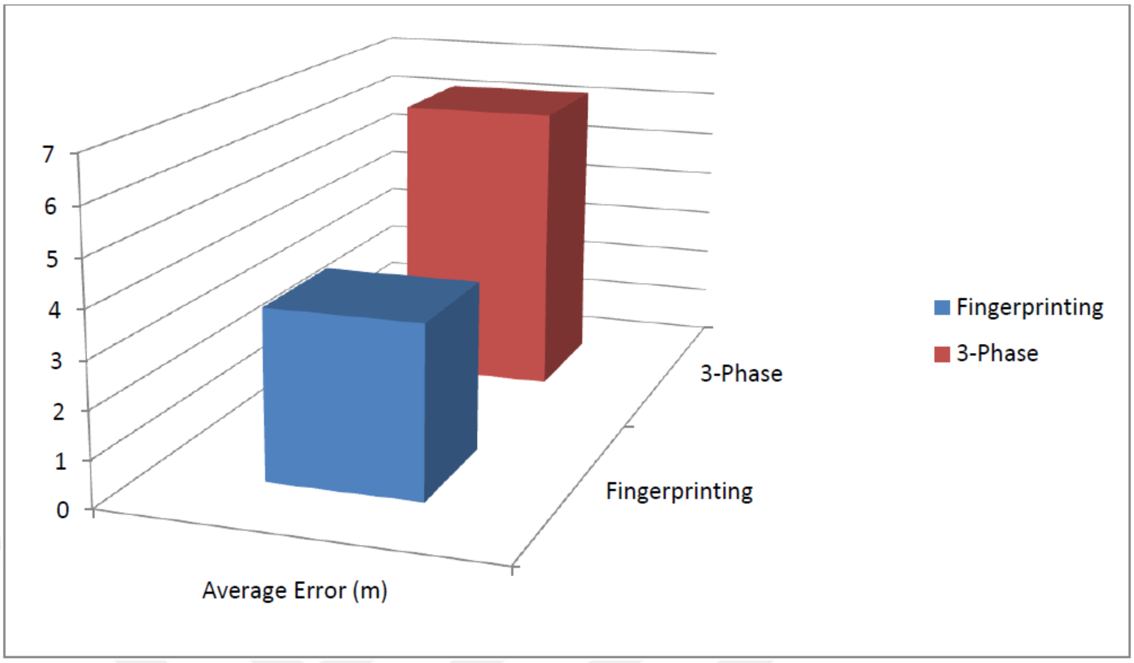


Figure 4.5. Mean error values when the system uses the device's own SS/metric ratio

Figure 4.6 presents the results of the second scenario, where neighbor’s SS/metric ratio was used to calculate the distance.

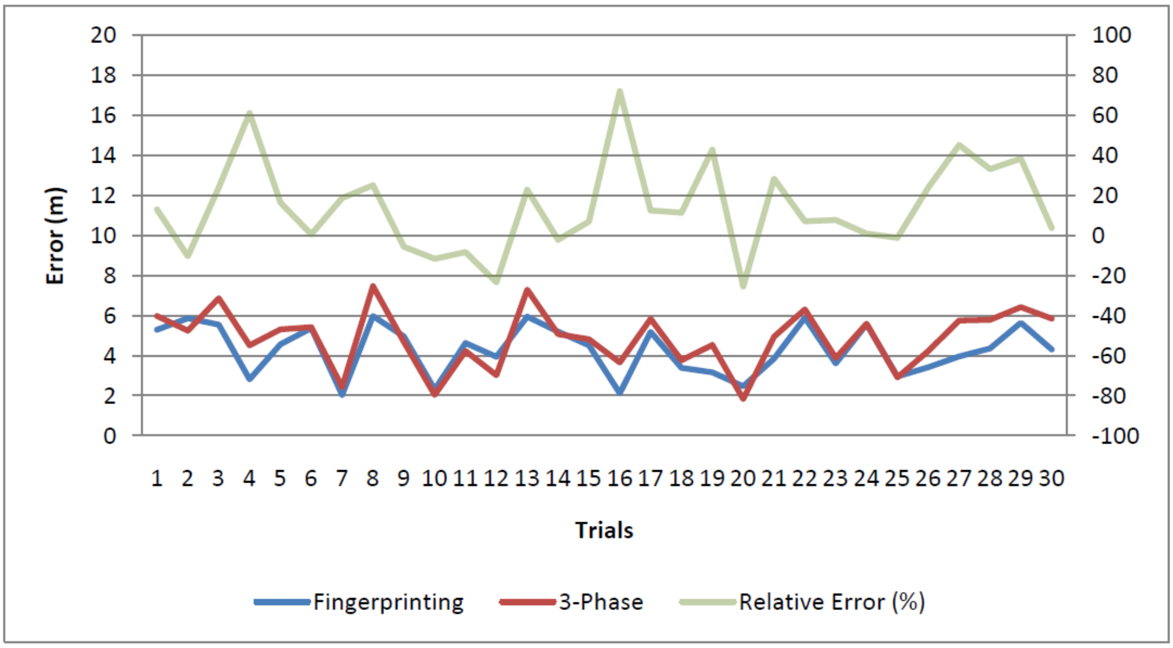


Figure 4.6. Error rates when the system uses the other device's SS/metric ratio

For the second scenario, average distance error of fingerprinting algorithm is approximately 4.3 meters. With the same RSS readings average distance error of three phases approach is approximately 4.86 meters. According to the average distance errors, which are depicted in Figure 4.7, it is obviously seen that three phase approach still does not outperform the base system. However, in a few trials, three phase approach performed more accurate location estimation then fingerprinting technique.

By the way, it should be emphasized that, the mean error value of the base system was 4.3 meters in the second scenario. However, in the previous scenario, the mean distance error of the base system was less than 4 meters. The main reason of this variation is the time difference of the trials. When the scenarios are tested on different days, the environmental factors such as the number of person and number of running computers in the test bed were not the identical. In order to overcome this issue, each trial performs both fingerprinting and three phase approach with the same RSS observations to make a rational comparison.

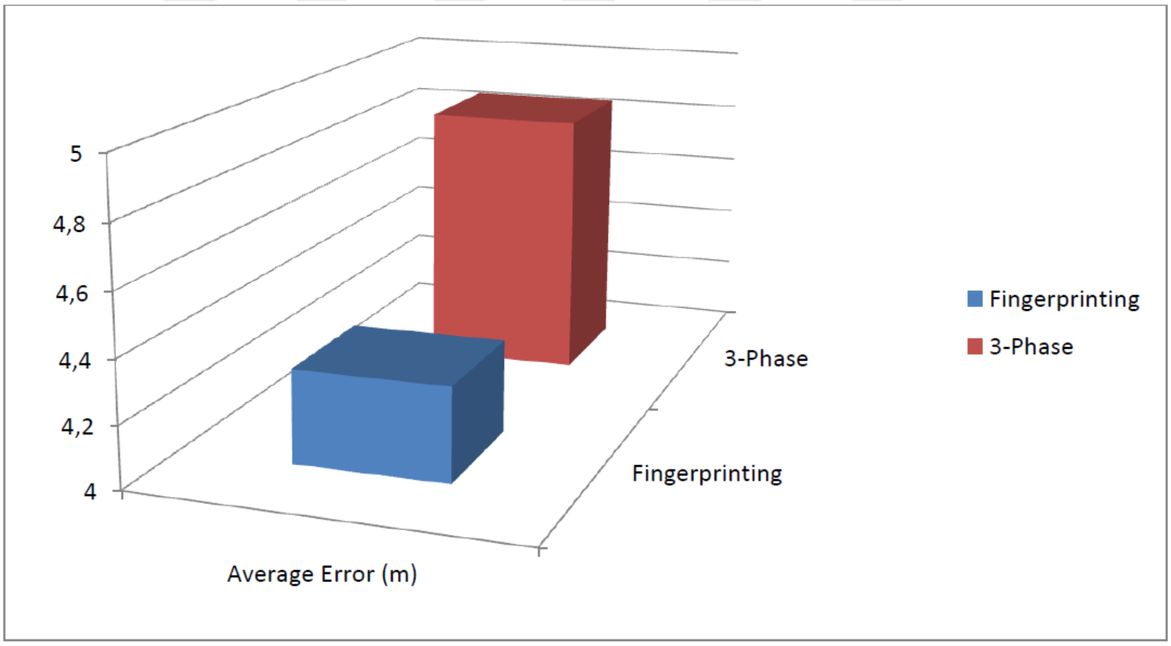


Figure 4.7. Mean error values when the system uses other device's SS/metric ratio

For the last scenario, three phase approach utilizes average of two devices’ SS/metric ratios to calculate the distances. The results of this scenario are shown in the Figure 4.8.

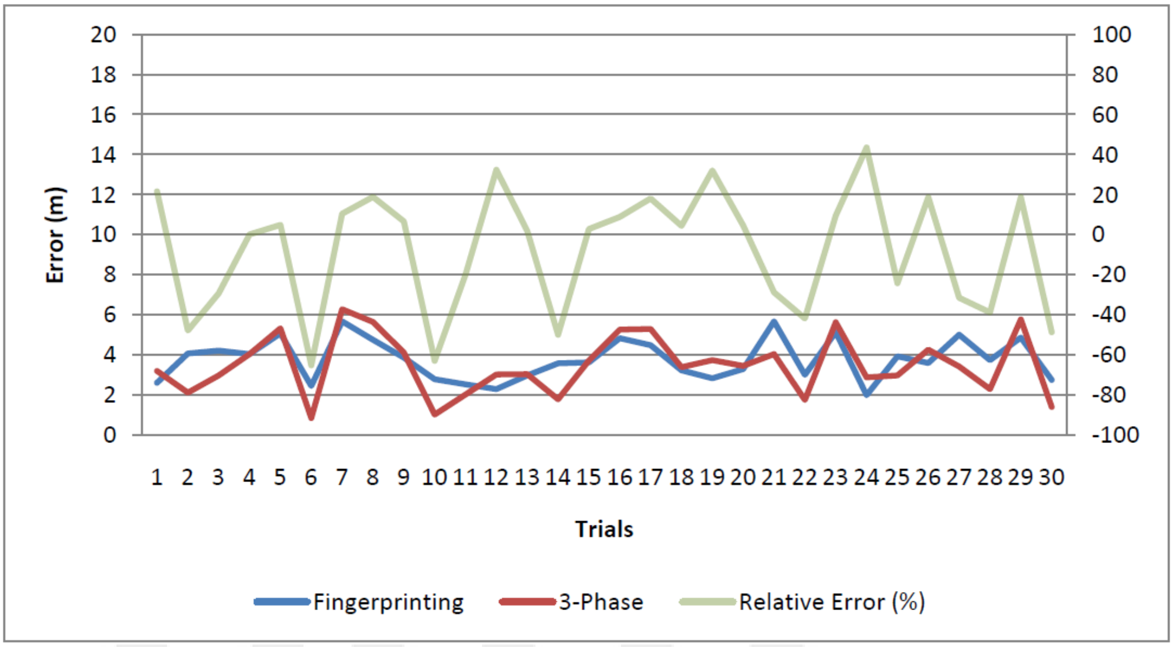


Figure 4.8. Error rates when the system uses the average SS/metric ratio

The base system's mean distance error was approximately 3.78 meters in the trials. However, mean distance error of three phase approach is approximately 3.65 meters. According to the mean distance errors, three phase location estimation slightly outperforms the base fingerprinting system. The comparison of the mean distance errors are highlighted in Figure 4.9.

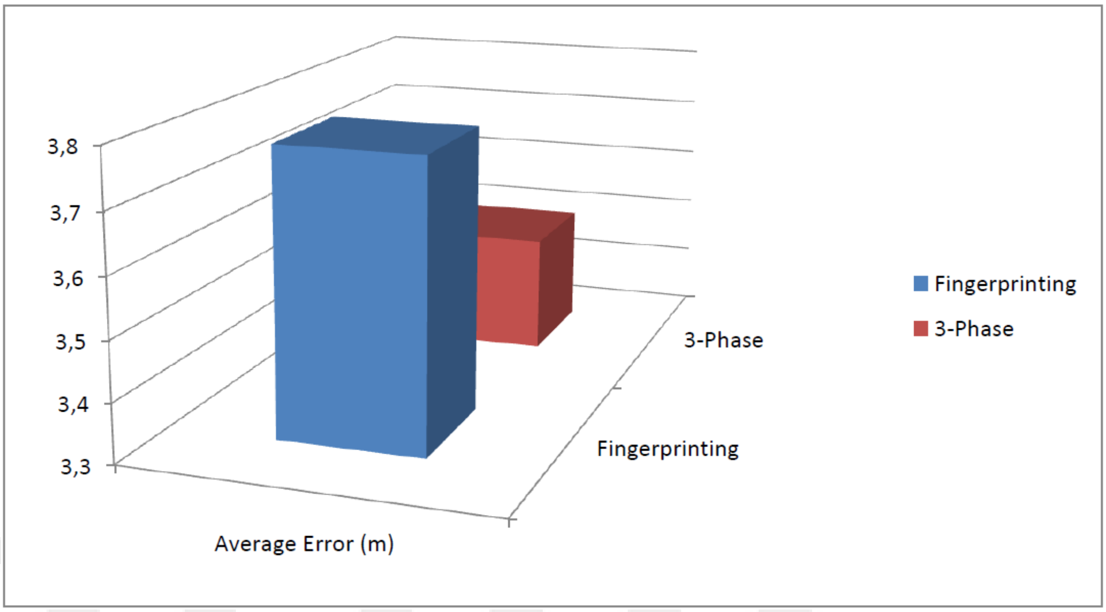


Figure 4.9. Mean error values when the system uses the average SS/metric ratio

Comparison of all three scenarios' relative distance errors is shown in the Figure 4.10. When the device uses its own SS/metric ratio in the algorithm, three phase approach causes accuracy loss up to 85 percent. Instead of using its own ratio, if the algorithm utilizes the neighbor's SS/metric ratio, three phase approach is again losing up to 14 percent of its accuracy. However, in the last scenario, when the average of the SS/metric ratio of both devices is used, the prototype provides accuracy gain up to 3.6 percent.

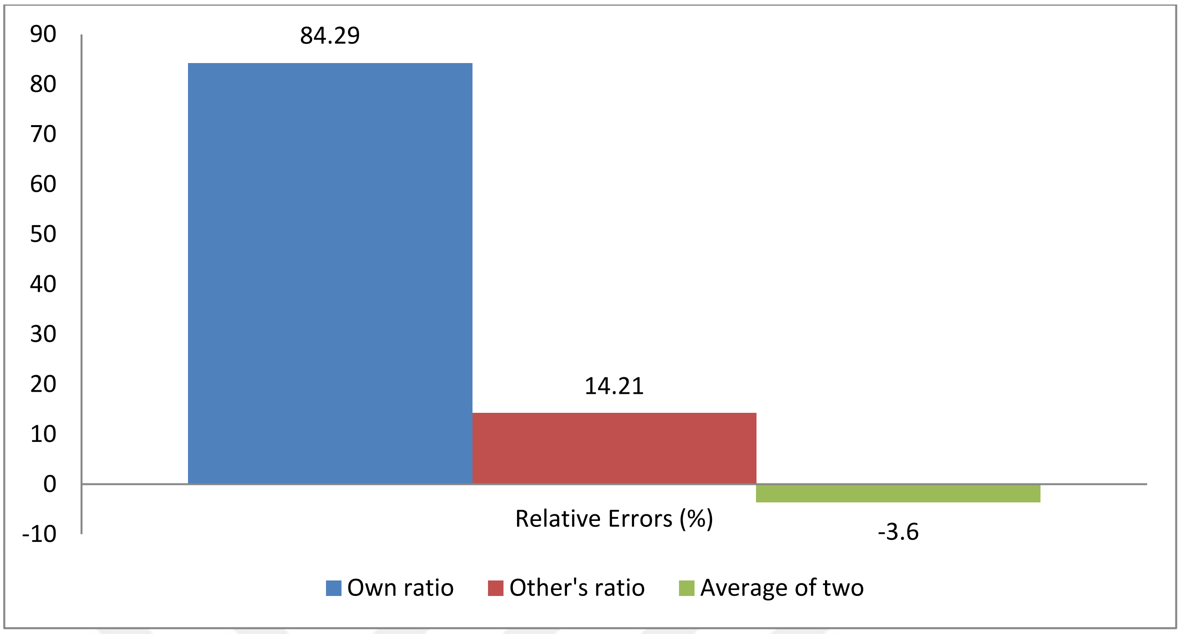


Figure 4.10. Averages for relative errors

Based on the trials' results, it can be said that the proposed system could not provide a significant improvement for location estimation. One of the main reasons of this situation is the limited number of devices, which we utilize in the trials. As it has been emphasized before, the objective of this study is share location data among the mobile devices in the vicinity so that each device can use this data to calibrate and refine its location estimation. The motivation of this study is to utilize this system in crowded area where each mobile device could help out each other by providing additional environmental data. Bearing in mind that our trials were limited with only 4 devices we had in hand, it was not possible to evaluate the "choosing the suitable neighbor" approach in the trilateration phase. This phase required large number of mobile devices to be deployed in a relatively small area to obtain realistic results. In our trials the number of devices was limited with 4 and, due to this fact; same three neighbors were utilized for trilateration, which eventually produced non-realistic results.

On the other hand, another main reason of inaccurate results is the performance of mesh network. For example, the daemon, which is utilized in the prototype to deploy the WMN, had some connection problems during the trials. However, WMNs are still in the growing state and there are many ongoing studies aiming to make mesh network communication better and faster. Improvements provided on the WMN would probably overcome connection issues, and accordingly it will increase the accuracy of the three phase approach.

As a result, an intended improvement could not be provided with the proposed system in this circumstance. The Mesh-Aided fingerprinting system does not decrease the mean distance error in many trials. It is concluded that the developed system usually could not provide enhancement on the nearest RPs selection which was the main goal of the study. In other words, the environmental factors such as reflection, diffraction and refraction could not be eliminated in this study. Accordingly, in order to achieve this goal and provide better nearest RPs selection, a new system was proposed. In the new system, adding a clustering method to the nearest RPs selection is targeted. The proposed system would attempt to eliminate the miscalculated nearest neighbors by grouping the accurate nearest RPs and utilizing them into estimation. The proposed system is detailed in the following chapter as Enhanced Fingerprinting via k-means Clustering.

5. ENHANCED FINGERPRINTING VIA K-MEANS CLUSTERING

As emphasized in previous chapters, the environmental effects such as reflection, diffraction and scattering could affect the accuracy of location estimation negatively. Effectively, these cause selection of non-nearest RPs as nearest RPs due to the signal fluctuations. Therefore, this study aims to improve the accuracy of KNN algorithm by enhancing the neighboring point selection process by applying k-means clustering. In the proposed method, k-means clustering algorithm groups nearest neighbors according to their distances to mobile user and uses the closest group to the mobile user to calculate the MU's location. The evaluation results indicate that the performance of clustered KNN is closely tied to the number of clusters, number of neighbors to be clustered and the initiation of the center points in k-mean algorithm [26].

5.1. BACKGROUND OF CLUSTERING IN FINGERPRINTING

One of the first positioning systems that utilizes clustering algorithm is developed by Ma et al. [27]. In this work, which was called Cluster Filtered KNN (CFK), KNN algorithm was improved by utilizing clustering technique to partition the neighbors into multiple clusters and one these clusters was chosen as a delegate. The results showed that KNN improved with average-linkage agglomerative Hierarchical Clustering (HC) outperforms KNN. Similarly, Sun et al. [28] developed KNN-FCM hybrid algorithm for WLAN based indoor positioning systems. In order to improve the KNN performance, they applied fuzzy c-means (FCM) clustering algorithm. K-nearest neighbors, determined by KNN algorithm, are classified into numerous clusters through FCM and one of them is chosen to calculate user position. Their simulation results indicate that KNN-FCM hybrid algorithm generally has better results than KNN when the distance error is less than 2 meters. From a different perspective, Roshanaei and Maleki [29] improved KNN algorithm by selecting the best number of nearest neighbor as K value dynamically. Dynamic-KNN (D-KNN) algorithm, which combines AOA and KNN algorithm as a hybrid method, utilizes the adaptive antenna system to determine the user’s location by intersecting several obtained AOAs. Thus, selection of the K neighbors within the intersecting AOAs results with the best K values. Their simulation results show that D-KNN outperforms KNN.

5.2. K-MEANS CLUSTERING METHODOLOGY IN FINGERPRINTING

In order to improve the KNN algorithm performance, k-means clustering algorithm is deployed. K-means is one of the simplest learning algorithms that solve the well known clustering problem.

It classifies a given data set through a certain number of clusters. The algorithm aims to find the centre point of a cluster by minimizing the distance between the cluster centre and members of the same cluster. Suppose we have a set of RPs as nearest neighbors, which are determined by KNN algorithm, x_j, j=1,..,N and we would like to organize them into K clusters $C = C_1, C_2,.., C_k$. The algorithm is composed of the following steps – see Algorithm 5.1 [30].

Algorithm 5.1. k-Means Clustering Algorithm

i. Initialize K centroid points which represent initial group centre point (centroid).
ii. Calculate distances between RPs and centroids.
iii. Assign each RP to a cluster that has the closest centroid.
iv. When all RPs are assigned, recalculate clusters centroids.
v. Repeat step ii, iii and iv until there is no change for each cluster.

After clustering operation is completed, the average distance between RPs and mobile user are calculated for each cluster. Then, the cluster with the lowest average value, cluster with the closest proximity to mobile user, is favored as the delegate cluster to determine the user's current position. The current position of user is obtained from cluster center (centroid) of delegate cluster.

The performance of the k-means algorithm varies based on the number of clusters, *k*, number of points in data set, *q*, and initialization of centroids. In our system, to prevent abundant calculations, we set the number of clusters to two. Accordingly, in our experiments one of the objectives was also to identify the optimal number of RPs to be employed in the clusters.

The k-means algorithm does not necessarily find the most optimal clustering sets. The algorithm is significantly sensitive to the initial selection of the centroids and the number of the clusters. Therefore, the approach taken to initialize K centroids and the number of clusters are important. One popular initialization approach is to randomly choose centroid points. On the other hand initialization can also be performed after obtaining information on the arrangement of the points by a coarse analysis of the data set.

In order to analyze the arrangement of the RP set, a dynamic initialization algorithm is developed in our system. This algorithm attempts to place centroids as far as possible from each other. To achieve this, algorithm performs coarse grouping of RPs by their IDs, which are assigned consecutively to neighboring RPs. It takes the average of the RP ID numbers in the data set, then assign nearest RP ID to the average value as first centroid and furthest RP as second centroid.

The algorithm first specifies a set of RPs which are relatively closer to each other than the remaining RPs. Following the creation of closest RP set, one centroid is selected from the members and one centroid is selected from the non-members of this set.

5.3. K-MEANS CLUSTERING IMPLEMENTATION IN POSITIONING

The proposed system was implemented on Visual Studio 2008 and .NET Compact Framework as in the base system. The existing WLAN infrastructure in the school building was utilized. The WLAN infrastructure was composed of three D-Link Wireless APs, one U.S. Robotics AP and one Cisco Aironet 340 series AP. In the development process, the project was tested on three different PDAs. These were Samsung i900, HP IPAQ 6515 and HP IPAQ 5500.

During the trials, the same test bed that was used to evaluate the base system was utilized. However, the number of RPs was increased based on the analysis of the base system, as described in the Section 3.3. With this update there are 57 RPs in the test bed instead of 48 RPs which are marked with a red cross (Figure 5.1). In order to compare the base system and the proposed system, both systems estimate the location of the MU in the trials. To reiterate, the test bed was 500 sq. meters (approx. 5381 sq. ft.) area, which was covered by five 802.11b/g APs.

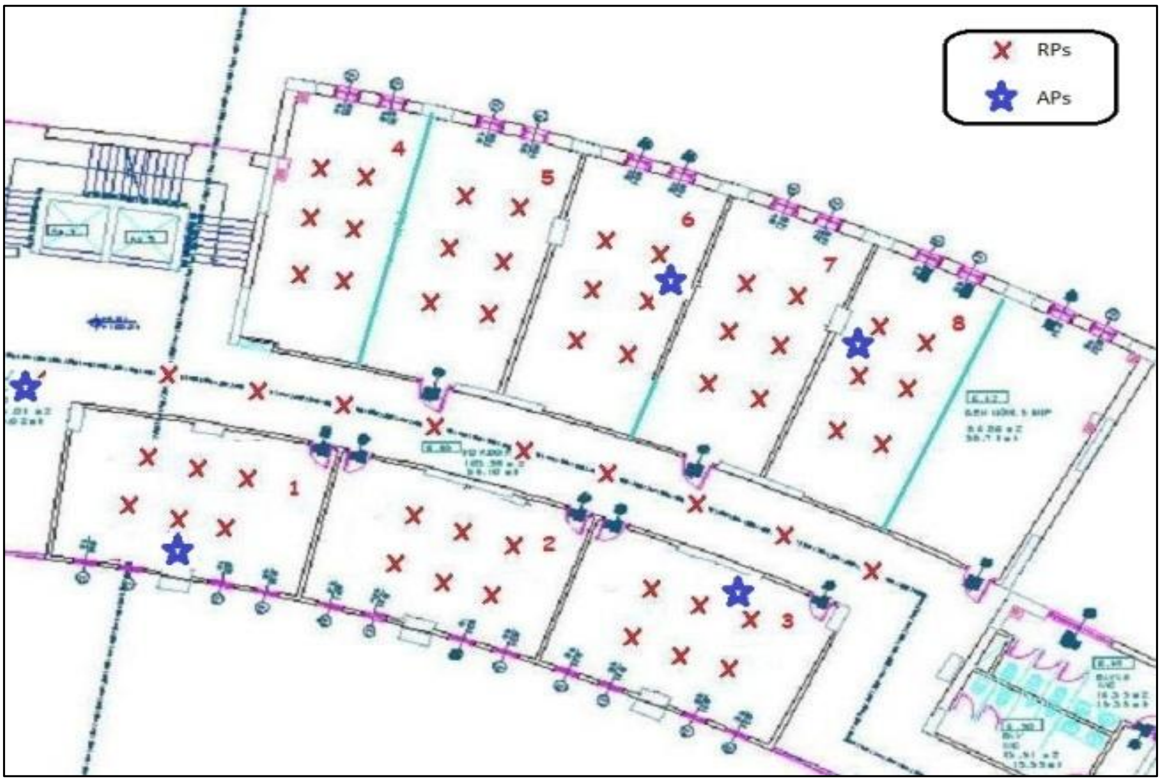


Figure 5.1. Floor plan of the test bed and new RPs

The k-means clustering algorithm mentioned in the previous sub-section was implemented so that the mobile user location is estimated by both base and proposed systems in all trials. This provides an opportunity to compare and see whether the prototype system provides accuracy improvement or not. The k-means clustering was implemented in C# programming language – and its source code can be found in Appendix A.

5.4. TEST RESULTS AND EVALUATION OF PROPOSED SYSTEM

All tests are performed stationary at pre-identified points in test bed area. These tests are aimed to observe the performance variation with various q values – the number of RPs within a cluster. Tests are performed with q value of 5, 7 and 9. For each q value, a number of measurements were carried out at 150 different pre-identified test locations by utilizing the fingerprinting algorithm and the findings were compared with actual coordinates within the building.

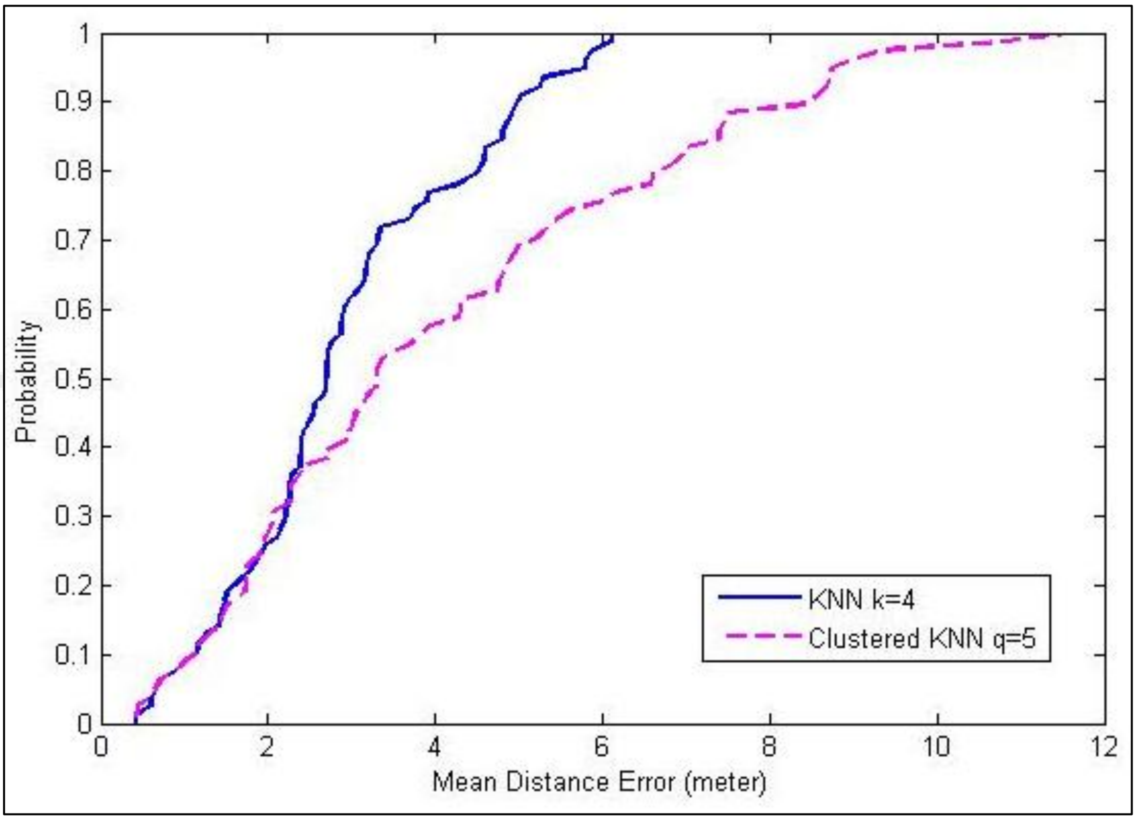


Figure 5.2. Comparison between Clustered KNN with $q = 5$ and KNN

As shown in Figure 5.2, the findings of the trials indicate that KNN algorithm performs better than Clustered KNN algorithm when q is equal to 5, since delegate cluster consists of two or three RPs. As a result, this scenario pans out with a less accurate prediction of user's location. This is caused by severe fluctuations in the received signal strength for mobile user even at fixed location. In such case, some distant RPs can be determined as nearest neighbors even if they are not. Nevertheless, miscalculated neighbors can be compensated by averaging three or four nearest neighbors. That is why K is mostly chosen to be three or four in KNN fingerprinting algorithms and, clustered KNN algorithm with two RPs cannot compensate miscalculated neighbors. This can be clearly seen in Figure 5.2, where it shows a correlation between clustered KNN algorithm and KNN algorithm up to 2 meters distance error. This is the case because distance error under 2 meters means that delegate cluster rarely includes miscalculated neighbors.

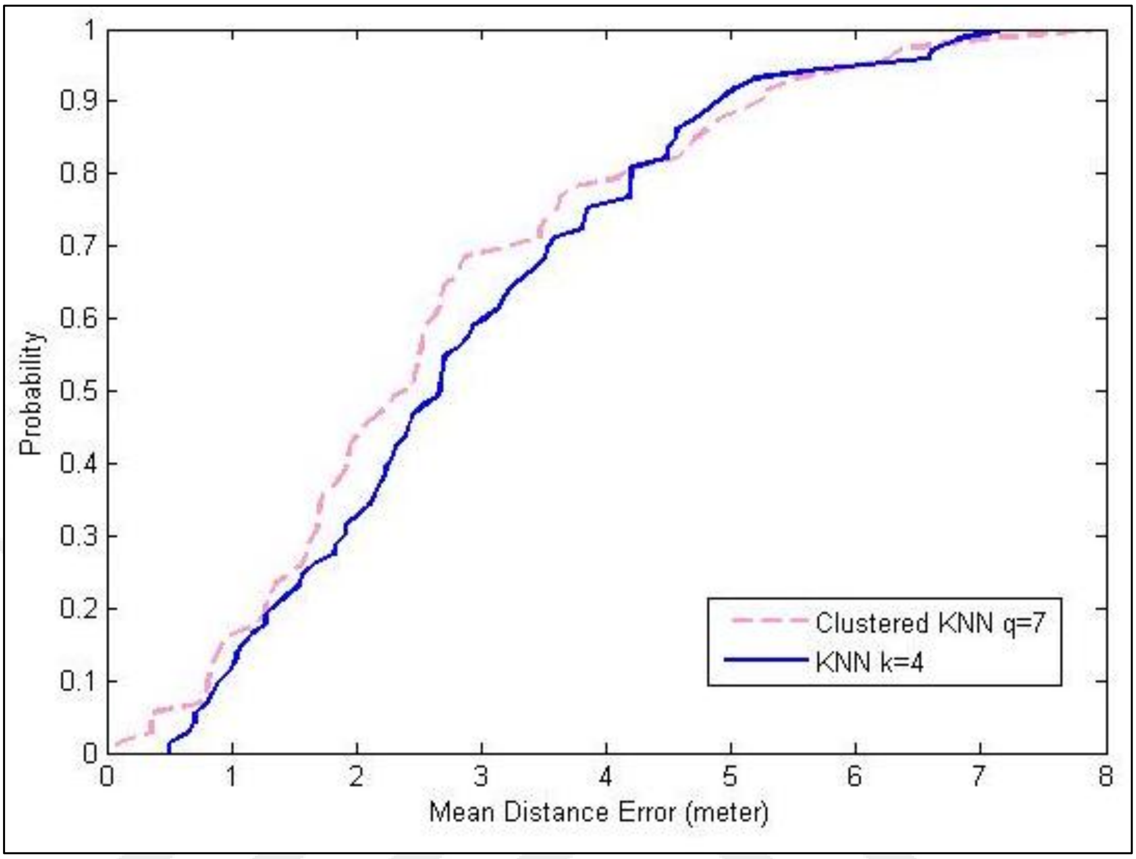


Figure 5.3. Comparison between Clustered KNN with *q* = 7 and KNN

Clustered KNN algorithm with q equal to 7 outperforms KNN algorithm as shown in Figure 5.3. The reasons for this is the size of the delegate cluster, which mostly it contains at least three or four RPs. As a result miscalculated neighbors can be eliminated by averaging these RPs. Moreover, the clustering algorithm attempts to coarsely group by utilizing the initialization algorithm described in section 5.2. This initialization algorithm eliminates the miscalculated nearest neighbors. Consequently, filtering miscalculated RPs within the nearest neighbors set in two stages help the system to predict user position more accurately than KNN algorithm.

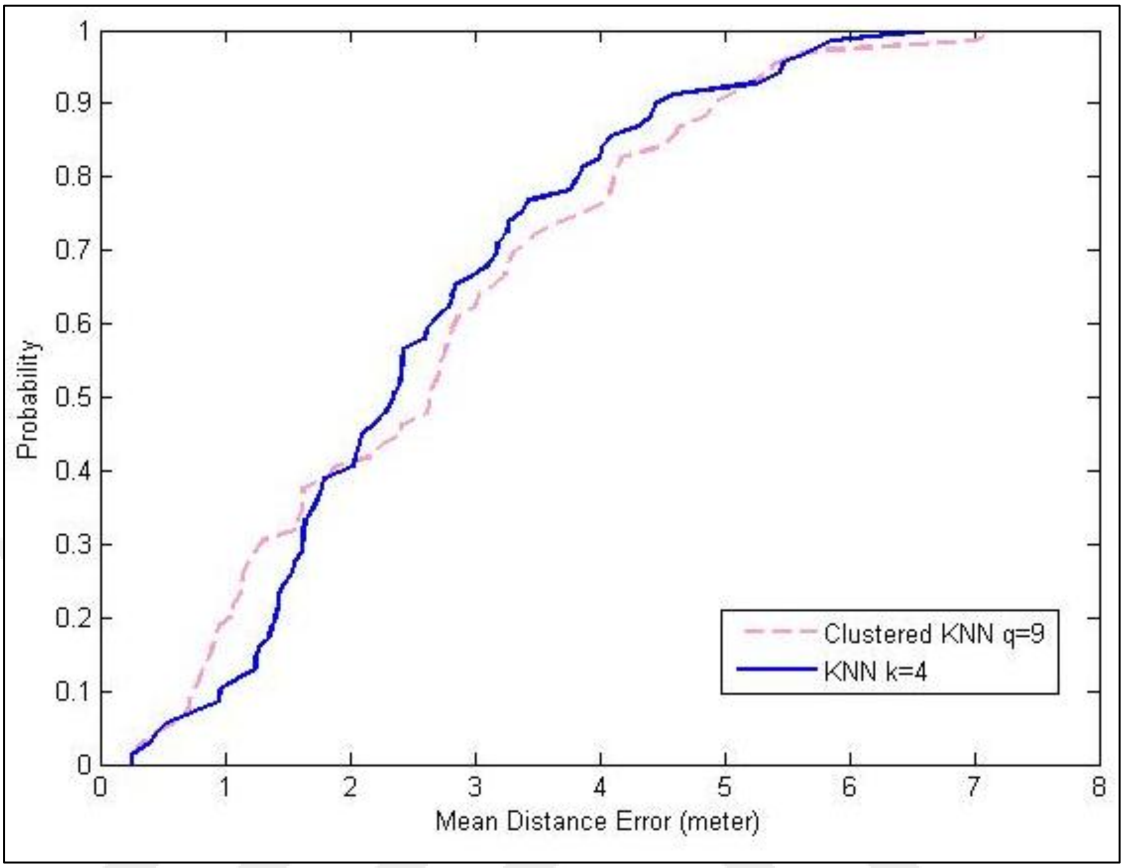


Figure 5.4. Comparison between Clustered KNN with q = 9 and KNN

Our experiments show that when q equal to 9, clustered KNN algorithm performs worse than KNN algorithm. The number of RPs in delegate cluster is around five and six. By setting q to nine we enlarge the selected group size thus increasing the probability of miscalculated nearest neighbors. As a result, the possibility of eliminating miscalculated neighbors caused by signal strength fluctuations decreases. Therefore, no more than one wrongly selected neighbor in delegate cluster can be compensated; consequently, user position cannot be predicted as accurate as in the KNN algorithm. However, Figure 5.4 also indicates that clustered KNN algorithm outperforms KNN when distance errors are less than two meters. It is believed that this is due to fewer numbers of miscalculated RPs.

Table 5.1. Average distance errors & standard deviation of test results

	Clustered KNN		KNN	
	Mean	Std. Dev.	Mean	Std. Dev.
q=5	4,11	2,7	2,9	1,45
q=7	2,7	1,73	2,89	1,6
q=9	2,68	1,66	2,60	1,44

From a different perspective, distance error mean of clustered KNN algorithm with variable q values can be compared with its relative KNN algorithm results. To conduct these tests, single signal strength is measured and distances to RPs is calculated with both clustered KNN algorithm and KNN algorithm. For example average distance error of clustered KNN algorithm, when q is set to 7, is approximately 2,7 meters and with the same received signal strength, the average distance error of KNN algorithm is approximately 2.9 meter (Table 5.1).

Mean distance errors and standard deviations of the trials show that standard deviation of clustered KNN algorithm is larger than of KNN algorithm. This is the case even if mean distance error of clustered KNN is less than KNN algorithm. We believe that this largely depends on the initialization algorithm employed to select the foremost centroids. Depending on the accuracy of the initially selected centroids, k-means clustering algorithm, by selecting a delegate cluster which is far away from the mobile user’s actual location, either produces more acceptable results or more inaccurate results.

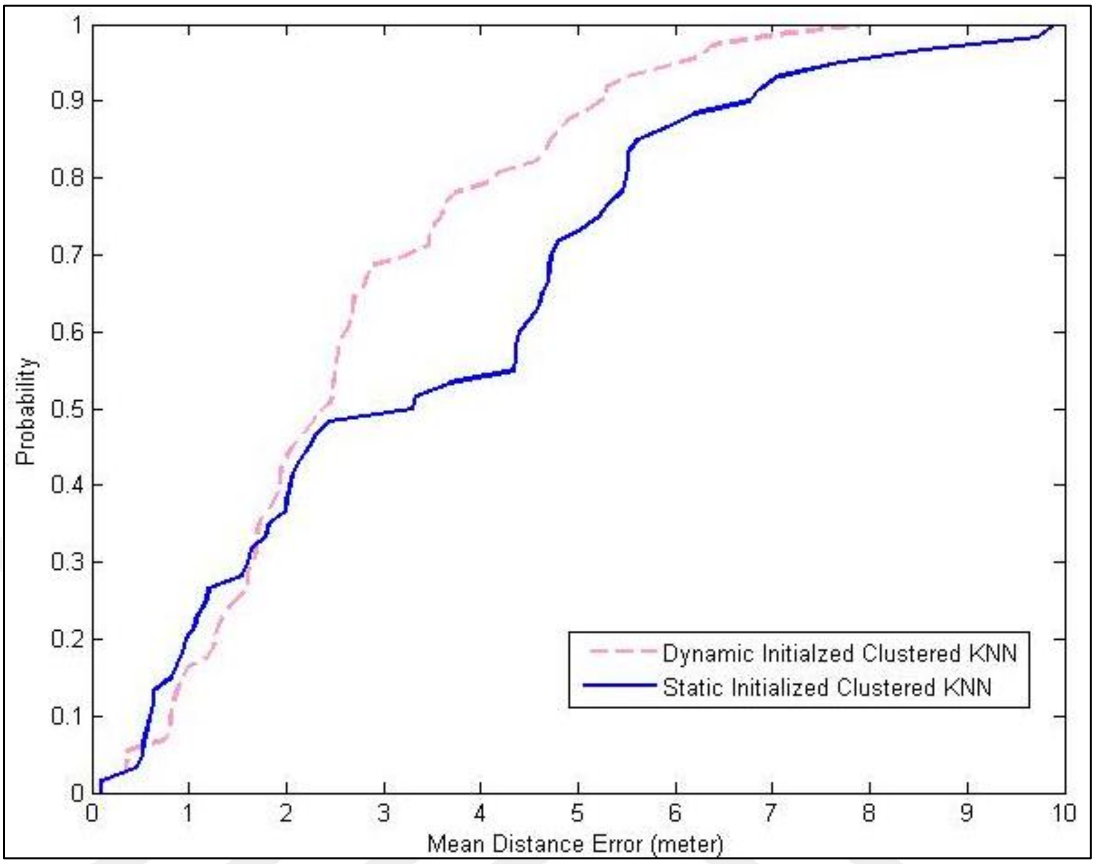


Figure 5.5. Comparison between static initialization and dynamic initialization

Furthermore, the static and dynamic initialization process of centroids was also compared and its effect on the k-means clustering algorithm was investigated. Since our typical initialization process is conducted dynamically, we already had the accuracy data set ready for this group. However, to collect the accuracy rate for static initialization, two centroids were identified based on their distance to the mobile user. One centroid was chosen to be the closest neighbor to the mobile user and the second was the most distant. Subsequently, when the k-means algorithm was employed, the results indicated that dynamically initialized centroids outperformed statically initialized centroids as shown in Figure 5.5. This confirms that coarse grouping of adjacent RPs method improves and refines the selection of accurate centroids.

As a final analysis, this proposed clustered KNN algorithm is compared with the Ma et al.'s [27] study (Figure 5.6). As mentioned in the section 5.1, Ma et al. utilized clustering technique to improve the KNN algorithm and called their proposed system as Cluster filtered KNN (CFK). The comparison of these two work shows that our clustered KNN algorithm improves KNN algorithm more than the CFK algorithm.

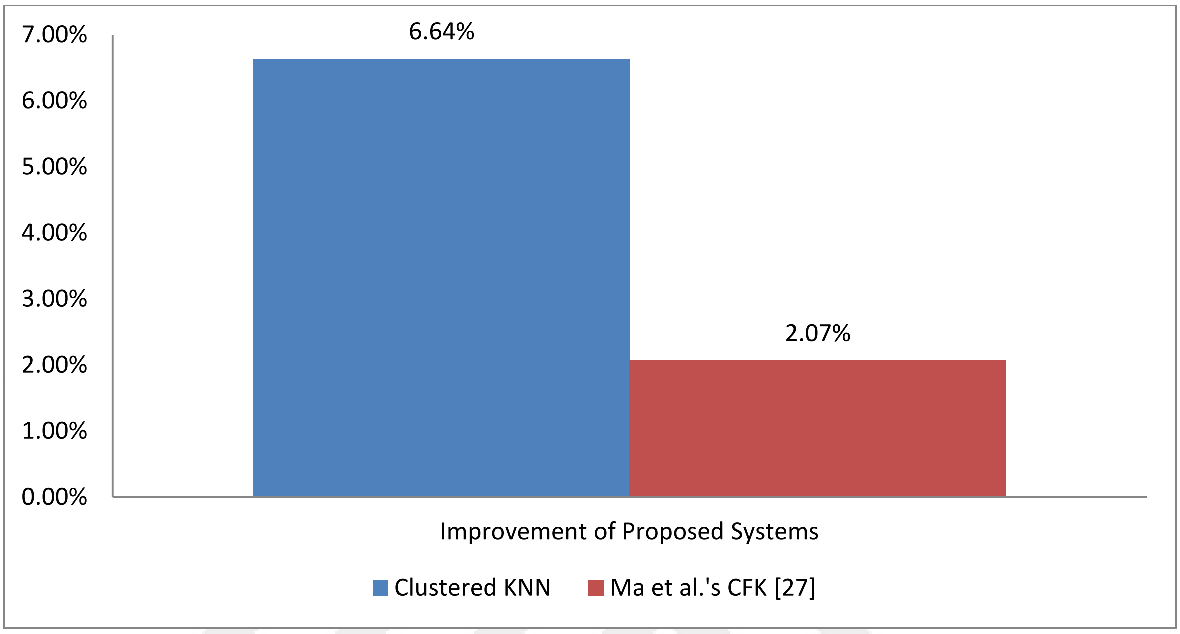


Figure 5.6. Comparison of Clustered KNN with Ma et al.'s CFK

To sum up, in this chapter, a RSS-based positioning approach that utilizes clustered KNN to estimate user location was evaluated. The results obtained throughout our evaluations indicate that the clustered KNN algorithm outperforms KNN algorithm when the number of nearest neighbors to be clustered is selected accurately. Thus, the goal of this study, which was eliminating the miscalculated nearest neighbors, is achieved in the proposed system.

Furthermore, with the help of the findings of this proposed system, a new approach was designed and implemented to improve the accuracy of the base system. The idea of the new system is to utilize the near-real time signal observations in the fingerprinting technique to provide extra data to eliminate the miscalculated nearest RPs. In order to use near real time RSS observations, they are stored on the device for a short time period. The new proposed system, which is called KNN with Short Term Memory (KNN-STM) is described in the following chapter.

6. RSS-BASED SHORT TERM MEMORY TECHNIQUE (KNN-STM)

In the pursuit of eliminating or reducing the number of miscalculated nearest RPs that pose problem in location estimation, which is the objective of this thesis, we reach to the third refinement method. In the third method, it aims to improve the KNN algorithm by integrating a short term memory (STM) where past signal strength readings are stored. Considering the limited movement capabilities of a mobile user in an indoor environment, user's previous locations can be taken into consideration to derive his/her current position. Hence, in the proposed approach, the signal strength readings are refined with the historical data prior to comparison with the environment's radio map.

Algorithms which use historical data such as the coordinates of user’s previous positions and his/her speed have already been developed in the past and can be found in the literature. However, to the best of our knowledge, no previous work has utilized a memory which stored the user’s previous signal strength observation values to improve the KNN algorithm performance. Our evaluation results indicate that the performance of enhanced KNN outperforms KNN algorithm [31]. In the following sections, the details of this study are described.

6.1. KNN-STM RELATED WORK

As mentioned above, KNN results with non-satisfaction may inaccurate data due to the indoor environmental factors that cause reflection, diffraction and scattering of the radio waves. In order to overcome these issues, advanced techniques and methods are being developed and integrated to KNN. As an example, the performance of the KNN was integrated to KNN. As it was described in the previous chapter, the performance of the KNN was improved extensively by the deployment of an auxiliary clustering algorithm. On the other hand, the accuracy of KNN algorithm can be improved by utilizing past collected data such as previous positions, speed of the mobile user. Thus, Khodayari e. al. [32] have developed a RSS-based fingerprinting method based on historical data, which estimates the current position of the mobile user by taking into account his/her previous positions and speed, in addition to KNN. In this study, they predicted the next probable location of user by taking into consideration his/her last two recorded locations. The speed and direction of the user was determined from these recorded locations, and the user's next position was predicted based on his/her direction. Accordingly, neighbors between the previous position and predicted position, which were named Predicted Neighbors, were used to calculate the user's current location. Their test results indicated that the improved KNN, named Predicted KNN, performed better than the KNN algorithm.

On the other hand, prior position data can also be utilized to calculate next position of the users by employing a probabilistic method. As an example, Xiang et. al. [33] have developed a WLAN-based indoor positioning system using a probabilistic method.

6.2. SYSTEM DESIGN AND IMPLEMENTATION OF KNN-STM

Movement of the mobile users in an indoor environment is generally limited to short displacements. Due to physical limitations, mobile users cannot make drastic movements in indoor environments. Hence, the mobile user cannot instantaneously move to another position too far from its prior position. Furthermore; mobile user cannot travel between two locations in a short time, especially, if there are walls between these locations. Thus, current position of the user can be associated with his/her prior position.

The characteristic of mobile user movements motivates us to take into account previous position information to predict next position of user. In order to achieve this, we add a short term memory to a location sensing system, which stores recent signal strength observations as historical data. Since, in indoor environments, it is only possible to have short displacement of users at a time, this should only cause a small signal strength variation between two locations. This means that user's previous signal strength readings that are obtained for position sensing, can significantly contribute in determining user's next position.

In the proposed method, we enhance the generation of $\mathbf{SS_{rt}}$ vector so that it can be used in the matching operation. In order to achieve this, the size of the OS is increased to store the previous signal strength observations. Although the observation size is increased, the number of observations performed at each position remained the same. By doing so the short term memory –observation stack (Figure 6.1b) – stored three times more signal strength observation data for each previous positions and current position instead of (Figure 6.1a). Accordingly, the vector $\mathbf{SS_{rt}}$ is generated by taking the median values of the SSs of each AP which are stored in the short term memory. The $\mathbf{SS_{rt}}$ vector is generated using the closest signal strength values – considering that users can only make short displacements in an indoor area – which indicates the small variations of SS between the two positions. This helps in the elimination of unexpected signal strength values which can occur due to the reflection, diffraction and scattering of the radio waves.

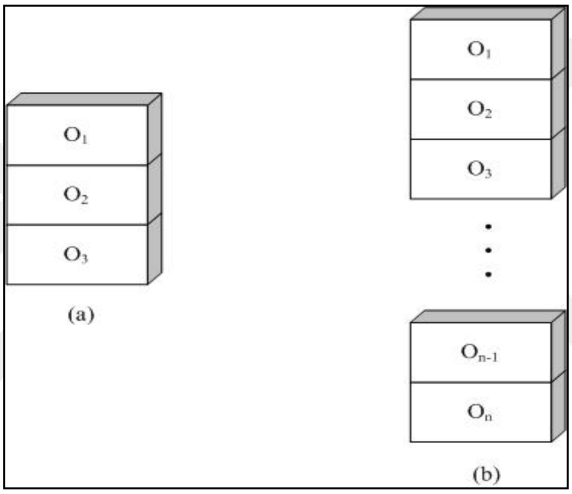


Figure 6.1. (a) Observation stack of KNN and (b) observation stack of improved KNN

However, in case the user is moving very rapidly and skips from one place to a very far corner of a building. This action results with big variations in signal strengths readings from a number APs. In this case, the historical data is ignored and removed from the short term memory. The position sensing is then performed only with the last three SS observations.

In the proposed method, to detect the movements of mobile user and identify whether there is large displacement between his/her last two positions or not, we utilize three variables. First one is the threshold value for signal strength variation (α), which is used to identify whether the difference of signal strengths between the current position and prior position is acceptable or not. Although the comparison based on α value provides information about user displacement, it is insufficient to decide whether to ignore the historical data or not.

Since, when the signal strength differences of one or two AP(s) exceed α and remaining APs signal strength difference are acceptable, the α threshold value conclude that the user displacement is not large enough to ignore the past collected data. Thus, we need a second variable to define a threshold value for each AP (ß), which is used to hold the number of APs whose signal strengths difference exceed α for in the last two position. Lastly, the third variable holds the maximum size of short term memory (ρ) which is utilized in the method.

Algorithm 6.1 KNN-STM algorithm

```
Set α (Signal Strength Threshold), β (Access Points Threshold) and ρ
(Window Size), Counter=0;
Obtain SSs from APs three times and insert into set T;
for each observation j in set T do
 |  if (SS_i^Tj - SS_i^Wj) is greater than α then
 |   |  counter++;
 |  end
end
if counter is greater than β then
 |  Empty Window;
end
Insert elements of T into Window W ;
if Window Length is greater than ρ then
 |  Remove first three element in W ;
end
```

Considering the variables described above, the steps that have to be taken to generate the $\mathbf{SS_{rt}}$ vector (Algorithm 6.1) are as follows:

- Define the size of short term memory (ρ), Signal Strength Threshold (α) and Access Point Threshold (β)
- Make three observations and insert them into temporary stack (TS) // TS = $\{O_1, O_2, O_3\}$
- Count of cases where difference of corresponding past SS elements and current TS readings are greater than the Signal Strength Threshold (α)

- If the number of above cases is greater than AP threshold (β), then empty the window. If not, insert elements of TS to OS
- If the OS size exceeds the maximum STM size (ρ), remove three observations which enter to OS first (FIFO).

Similar to the previously described studies, this proposed system was also implemented on Visual Studio 2008 and .NET Compact Framework as in the base system. In order to compare the base system and the proposed system under the same circumstances, existing WLAN infrastructure in the school building and the same test bed was utilized. As the same area of the school building was used; the test bed was 500 sq. meters (approx. 5381 sq. ft.) area, which was covered by five 802.11b/g APs which are marked with blue stars and 57 RPs that are marked with a red cross (Figure 5.1).

6.3. KNN-STM TEST RESULTS AND ANALYSIS

In order to compare the KNN with the improved KNN- STM algorithm, user position is determined by running both KNN and KNN-STM algorithms with same Signal Strengths that were collected by two separate mobile devices. For each different values of α, β and ρ set, fifty trials were performed.

6.3.1. Effect of Signal Strength Threshold (α)

Due to the movement limitation of users in indoor environments, users pass two or three RPs between the consecutive two location estimations. Accordingly, when the signal strength difference between the two or three RPs are analyzed, it is revealed that variation of SS from an AP varies between 5 and 15. Thus, we perform the experiments with three different threshold values which are 10, 15 and 20. As shown in the Table 6.1, when α. was set to 10 or 15, the improved method performs better than the KNN algorithm. When α is set to 20, KNN-STM method produces extremely less accurate predictions of user's location. This is caused by holding the previous positions data even if user's displacement is large. On the other hand, the reason for obtaining more accurate location predictions with KNN-STM algorithm is because, when α is set to 10 or 15 the signal strength difference between the consecutive positions does not exceed 15 in general. As a result, the historical data is significant to determine the current position, when user passes two or three RPs during the traveling from a position to another in the environment.

6.3.2. Effect of Access Points Threshold (β)

Even though the signal strength threshold provides information about user's displacement, it is inadequate to decide to whether the displacement of user a major leap and if it should disregard the previous signal observations or not. The prediction of user's position also depends to the number of APs whose signal strength difference between the consecutive positions exceeds threshold value (α). Since the number of APs in the surrounding environment is five, we carry out the experiments with two different access point threshold values – 2 and 3. The results indicate that KNN-STM surpasses the performance of KNN algorithm when ß is set either 2 or 3 and α is set either 10 or 15 (Table 6.1).

Table 6.1. Test results (mean distance error)

ρ	β	α	KNN	KNN – STM	Performance
9	2	10	2,50	2,19	12,4%
		15	2,29	2,11	8,11%
		20	3,05	5,33	-74,8%
	3	10	2,66	2,08	21,85%
		15	2,52	2,11	16,25%
		20	2,49	5,10	-105%
12	2	10	2,38	2,10	11,87%
		15	2,20	2,11	4,41%
		20	2,79	2,91	-5,53%
	3	10	2,44	2,23	8,79%
		15	2,32	2,06	10,85%
		20	N/A	N/A	

6.3.3. Effect of Maximum STM Size (ρ)

The maximum size of the short term memory is the last variable that we test to identify its effects on the performance of the KNN-STM algorithm. The tests are performed with two different STM sizes – 9 and 12 – to store historical data in 3 or 4 SS observation triplets. The results of the experiment points out that the selected ρ values are appropriate for KNN-STM algorithm when α is set to 10 or 15 and ß is set to 2 or 3. Also, it has to be mentioned that when the maximum STM is set to a value larger than 12, it increases the CPU and memory usage, thus reduces the device responsiveness. On the other hand, setting ρ to a value smaller than 9, reduces the observation stack and decreases the level of accuracy.

6.3.4. Evaluation of KNN-STM

The empirical cumulative distribution functions of the KNN and KNN-STM algorithms in Figure 6.2 confirms that the developed algorithm achieves its goal and eliminates the miscalculated nearest neighbors that existed in the KNN algorithm. Subsequently, the enhanced algorithm predicts the position of the user more accurately than the KNN.

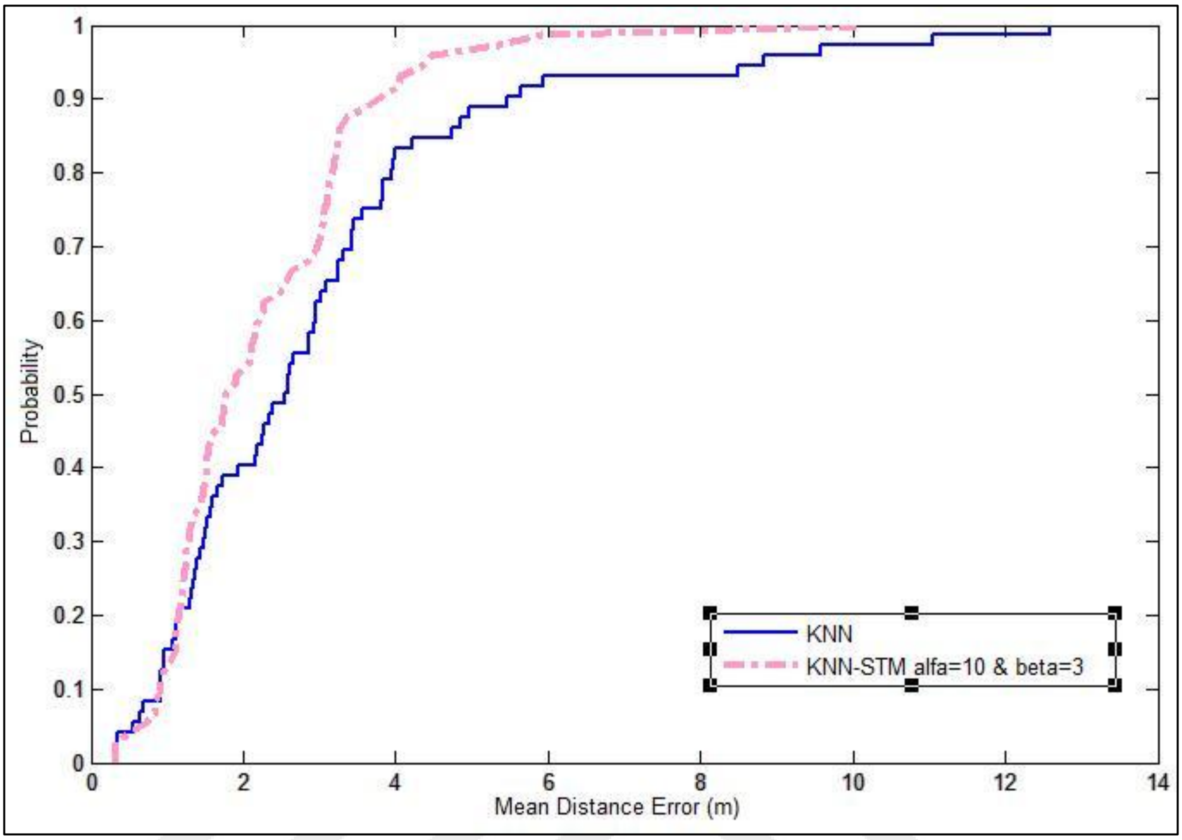


Figure 6.2. Comparison between KNN and KNN-STM (α=10, β=3 and ρ=9)

Furthermore, KNN-STM senses the position of the user more consistently than KNN since the short term memory provides a foresight regarding the next possible position of user. The trial results point out that the mean distance errors of KNN algorithm varies between 2.2 meters and 3 meters. However, the mean distance error of KNN-STM varies between 2.06 meters and 2.19 meters when the signal strength threshold value (α) is equal or smaller than 15.

As a final analysis, KNN-STM is compared with the Khodayari et al.'s [32] study, which is detailed in section 6.1. Figure 6.3 shows that Khodayari et al.'s proposed system's performance gain up is more than my KNN-STM. However, it should be emphasized that in Khodayari et al.'s trials, the mean distance error of traditional KNN algorithm was about 3.95 meters in their test bed. Then, this error rate was reduced to the 2.65 meters when their proposed system was applied. However, in our trials, traditional KNN algorithm's mean distance error was approximately 2.5 meters. And, KNN-STM reduced this error rate to approximately 2 meters.

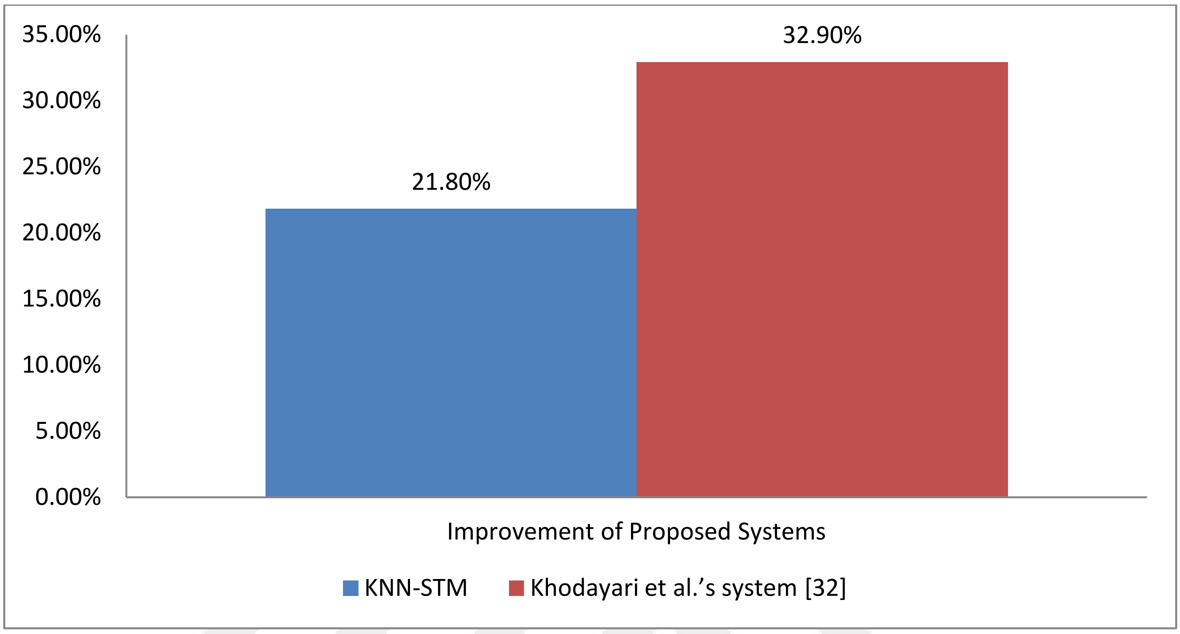


Figure 6.3. Comparison of KNN-STM with Khodayari et al.'s system

The following chapter describes the implementation and evaluation of a prototype location-based services aided application using the know-how gained through out location sensing trials. This work aims to implement a location-aware handheld tool for the medical personnel to access, keep track and update patient health record. The core feature of this tool is its ability to detect location and download relevant patient records according to the practitioner's location. In the scope of this work, the usability assessment and location estimation evaluation are carried out.

7. LOCATION-AWARE ACCESS TO PATIENT RECORDS

Emerging wireless technologies enabled medical personnel to work effectively anywhere/anytime. With the advances in medical technology, treatment of patients can start in ambulances and high-risk patients can be monitored in 24-hour bases. However, on the other hand, reports [34] on healthcare indicate that now medical personnel have “more to know, more to do, more to manage, more to watch, and more people involved than ever before”, which draws the attention to the needs of healthcare personnel.

Therefore, the focus of this chapter will be the implementation and evaluation of a location-aware patient record system. Location-aware medical healthcare record systems have been implemented in the past and these could be found in the literature. However, to the best of our knowledge, no previous work has applied RSS-based fingerprinting technique for location detection in a hospital environment.

The proposed location-aware data access application’s architecture is shown in the Figure 7.1. It is composed of two parts; (1) location sensing, where PDA observed signal strengths of access points then locates the position and determines the room the user is in; and (2) accessing to patient records, where PDA downloads the relevant patient data for the physician’s use.

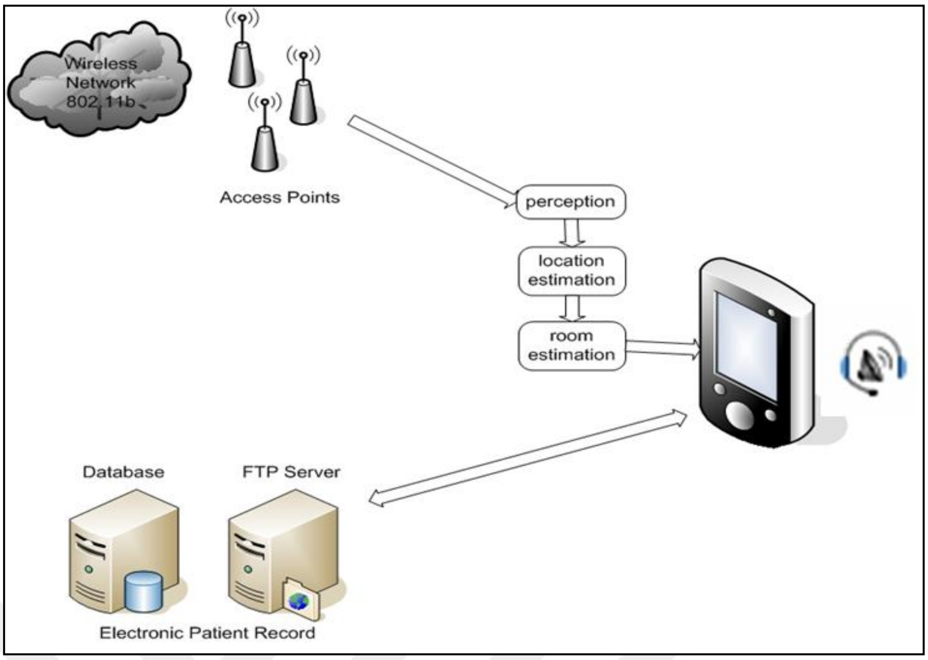


Figure 7.1. System architecture

In order to create a realistic patient record database, contents of currently used medical files was requested from the Haydarpaşa Numune Hospital, Istanbul, Turkey. Additional interviews with physicians were held in order to capture and understand the practical uses of patient files in their daily routine. It has been elicited that the hospital has kept all the patient records, such as personal information, diagnosis data, prescription history, on physical files called ‘Medical Examination and Observation Paper’. In particular, the interviews with staff revealed that the main difficulty with the existing system was finding, accessing and recording data to the patients’ physically stored health records. In brief, the identified requirements were:

- Easy access to patient personal information and general health records
- Read/write or listen/record history of patient easily
- Review laboratory test results or request new tests
- Quick access to previous diagnosis/ prescription/ medication or write new medications or prescription

Bearing in mind that the target users of this application would be physicians and nurses with limited technical knowledge and expertise, the location-aware patient records system was designed with a user friendly interface and flip-pad-like analogy. This metaphor was specifically chosen to ease the transition from flip-boards and flip-charts to electronic record system.

The system which satisfied all of the above requirements was developed using Visual Studio 2008 and .NET compact framework on PDA. In addition, electronic patient records (EPR) database was implemented using MS SQL Server 2008.

The PDA side of the location-aware patient records system is developed using .NET Compact Framework and three auxiliary libraries (Figure 7.2). These auxiliary libraries are OpennetCF, which is used to collect the SSs from the WiFi adapter; PInvokeCS, which is used to record and play audio files; and lastly, FTP Client library to facilitate the audio file transfer between the EPR database server and the PDA.

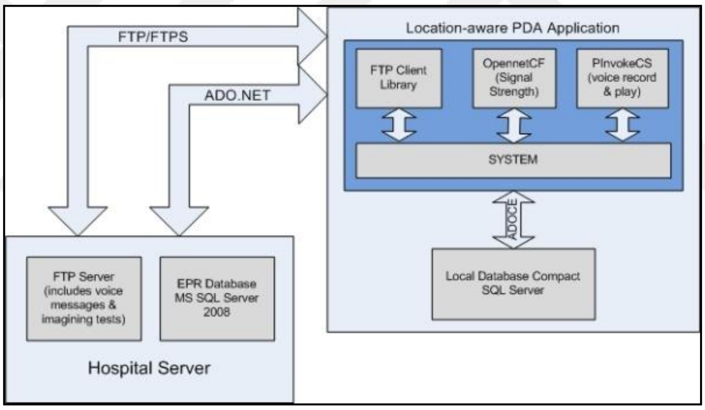


Figure 7.2. Detailed architecture of the system

The PDA user interface is divided into four main tabs that contain all the patient data. Hence, ‘Personal Information’ tab (Figure 7.3.a) includes general information about the patient; ‘History’ tab (Figure 7.3.b) includes patient’s previous information, answers to routine health check questions and last physical examination results.

‘Laboratory’ tab (Figure 7.3.c) includes past laboratory test results, laboratory test request form, test result images (CT, X-RAY, MRI etc.). Finally, the ‘Patient Follow Up’ tab contains information about previous prescriptions and diagnoses (Figure 7.3.d).

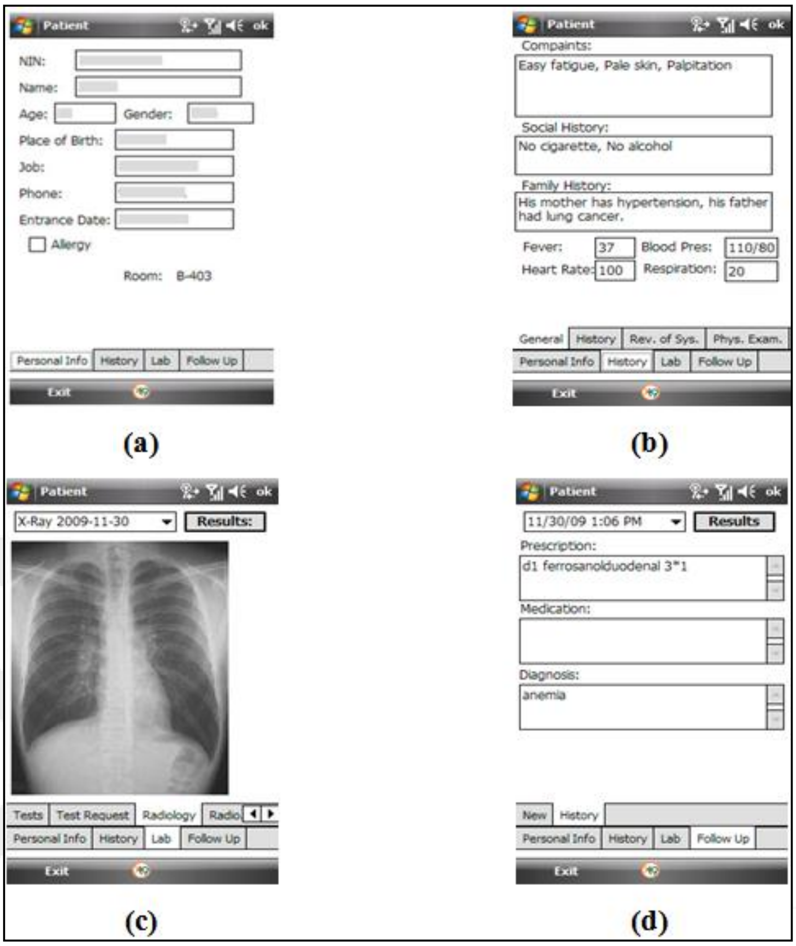


Figure 7.3. (a) Patient information screen (b) medical history information (c) radiology test results (d) prescription and diagnoses

The system evaluations took place at the 8-room test bed area, which was located at the university's medical school building. In order to make our evaluation as realistic as possible, solely university's existing wireless network infrastructure was used.

The participants were selected among medical school students. The prototype location-aware hospital ward system was evaluated by 10 users - 4 female and 6 male. Participants' age ranged between 20 and 24. All of them had basic IT knowledge and were frequent Internet users. However, none of them had previous Smartphone or PDA experience.

The location-aware hospital ward system evaluation had two facets: (1) user interface and usability assessment and (2) location sensing precision measurement. To evaluate usability and ease of use, participating student doctors were given a one-page user manual and asked to access the patient records of the residents in the test bed area using PDAs for the period of 24 hours. At the end of the evaluation period, users were asked to complete a questionnaire (Appendix B.), in which their opinions over a range of statements were recorded on a 5-point Likert scale, where 5 corresponded to the most positive response, and 1 to the most negative. The prominent questions from this questionnaire are evaluated in this study (Table 7.1). Users were also asked to note any other comments they might have had in respect of the developed system. Moreover, to evaluate the location sensing capabilities of the system, every time a change was sensed, the PDA application asked users whether it had picked up the correct patient's records or not and recorded their responses.

In general, the evaluation results collected from participating student doctors were positive [35]. Users found the laboratory test request interface easy to use and helpful. These results are especially encouraging, since none of the participating users had prior PDA experience. It is believed that the reason for Question 1 collecting high opinion score is because the new approach saves the doctors from filling bulky test request form papers.

Table 7.1. Questions used to evaluate the PDA application

#	Question	Mean	Std. Dev.
Q1	It is easy to request new laboratory & radiology tests	4.8	0,42163
Q2	Playing audio-notes is easy	3.7	1,33749
Q3	The application's user interface is consistent and easy to follow	4.7	0,48304
Q4	Accessing to previous lab test result is easy	4.4	0,69920
Q5	It is easy to make mistakes while using this application	2.3	0,94868
Q6	It is useful to have on the move access to digital patient records	4.2	0,78881
Q7	It is easy to add a new prescription for the patient	3.5	1,50923

Furthermore, strongly positive results were obtained regarding the application user interface – Question 3. It is also believed that this was the case because a patient flip chart analogy was followed to design the interface. Thus, the user student doctor had no issues traversing or locating a specific data within the patient records.

Lastly, responses to Question 6 highlight the importance of on the move and timely data access needs of physicians. The proposed system calculates devices location and downloads relevant patient records without requiring any interaction from its user. In the open-ended feedback section, this feature has been acknowledged to be the most useful by three of our 10 participants.

8. DISCUSSION

The previous work by Altintas [2] was considered as a base system and this thesis is built upon the idea of improving the location detection by enhancing the algorithms behind the scenes. The analyses on the base system pointed out that one of main problems of the KNN algorithm was not too successful in selecting the nearest RPs. In order to overcome this issue, three separate enhancement approaches to the existing base system were proposed, developed and evaluated in the scope of this Master thesis.

The first approach was the mesh network-aided location sensing system. The new approach attempted to take advantage of the neighboring mobile device location information and RSS observations. The idea here was to have abundance of data to work on to determine the nearest RPs. However, the evaluation test results point out that the proposed system could not achieve better location estimation accuracy. It is believed that this was the case due to the limited number of mobile devices deployed in the environment – in our case only four mobile devices.

When the mesh network-aided fingerprinting system did not provide better estimation results, a new approach was proposed to eliminate miscalculated nearest RPs. This second approach made use of the analysis of the first proposed system's trials. Deeper analysis of the trials showed that the fifth or sixth nearest RP – not in the selected and utilized four RPs – could be actual nearest RPs. As an example, in a scenario, selected four nearest RPs includes a RP which is too far from the others. For this scenario, the second approach aims to get rid of the problematic (i.e. miscalculated) RP, which is the main cause of the high error rated location sensing. For this purpose, this second approach attempts to take out the problematic RP from to nearest RPs group, so that it would have no contribution on the location sensing. In order to achieve this, a clustering algorithm was added to the existing system to group closer RPs. Afterwards; the closer group of RPs was used in the location estimation process. The second approach achieved to an improvement of 6.64 percent accuracy, which is satisfactory for indoor environments.

Based on experience gained from the previous approaches, third study was carried out to enhance the accuracy of location estimation. The idea of this third approach was to utilize the near-real time signal observations in the fingerprinting technique to eliminate the inaccurate nearest neighbor from the nearest RP group. Considering the limited movement capabilities of a mobile user in an indoor environment, user's previous locations can be taken into consideration to derive his/her current position. For this purpose, this study attempted to improve the KNN algorithm by integrating a short term memory (STM) where past signal strength readings are stored. The mean of distance error of KNN-STM algorithm with the proper parameters (thresholds is approx. 2.08 meters where the mean distance error of KNN algorithm is approx. 2.66 meters.

Overall, Figure 8.1 highlights the performance improvements achieved by the three proposed approaches. The performance improvement indicates the mean distance error reduction for this study. The KNN-STM approach got the best accuracy improvements in the experiments. Clustered KNN algorithm also achieved a valuable performance enhancement. However, mesh network-aided proposed system could not reach the intended performance gain.

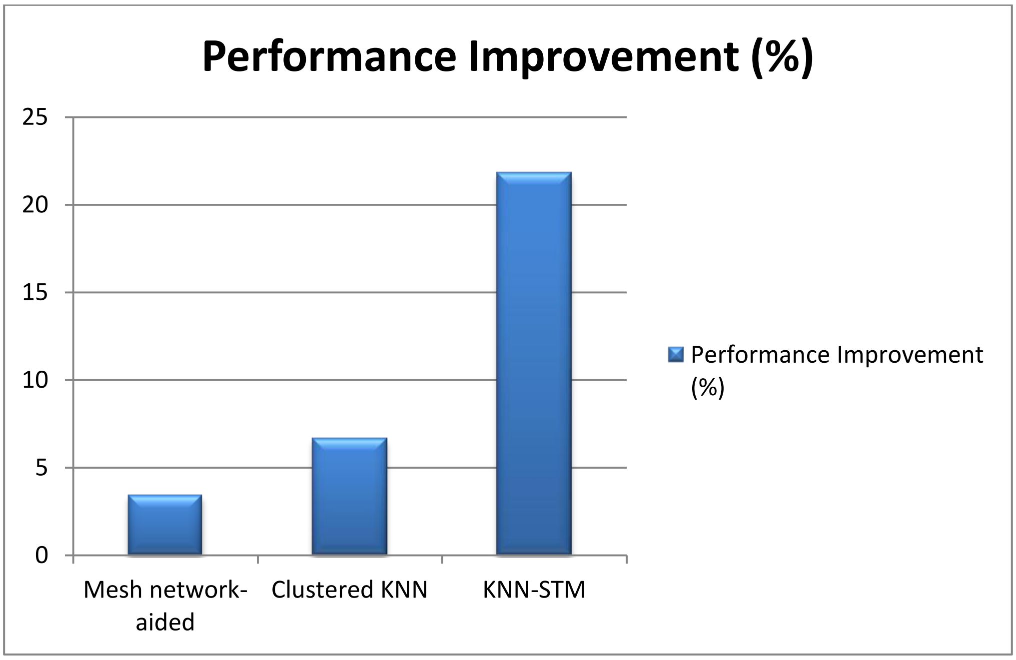


Figure 8.1. Performance comparison between the proposed systems

Finally, a location-aware electronic health record system was developed as an applied system. This system can sense the location of the physician by utilizing fingerprinting technique, and retrieve the relevant patient's medical data on to the physician's mobile device. Furthermore, this system also enables medical personnel to transcribe post-it-like, audio-notes, and facilitate communication among physicians on other shifts by posting location-based notes. The prototype system's location precision and usability evaluation results indicate that the proposed system is conceived as easy to use, accurate, and efficient tool.

9. CONCLUSION & FUTURE WORK

The most common algorithm adopted for Received Signal Strength (RSS)-based location sensing is K Nearest Neighbor (KNN), which calculates K nearest neighboring points to estimate location. The literature review and the analysis in them highlight that the main cause of the low accuracy in the KNN algorithm is the radio signal fluctuations. The inaccurate signal strength observations, due to fluctuation, lead to inaccurate nearest neighbors' calculations. This study aspires to enhance the RSS-based location sensing system, which its accuracy suffers from severe signal oscillations, by three new approaches. The common goal of these three approaches was to eliminate the miscalculated nearest neighbors.

The first proposed system – mesh network-aided location sensing – utilizes the location and RSS observations of the other mobile devices in the vicinity. The idea is to increase the amount of data so a refinement phase can be executed on top of the traditional location detection process. For this purpose, a mesh network is setup among the mobile devices in the environment, to enable them to share their locations and signal observations. Then, a refinement phase is added to the KNN algorithm. However, the evaluation test results point out that the proposed system could not achieve better location estimation accuracy. It is believed that this was the case due to the limited number of mobile devices deployed in the environment – in our case only four mobile devices.

In the light of the first proposed system's evaluation results, where showed no improvement in the elimination of the miscalculated nearest neighbors, a second system was developed with the same purpose. The aim of the proposed system is to apply k-means clustering to improve the KNN algorithm by enhancing the neighboring point selection process. Nearest neighbors are grouped based on their distance to mobile user with k-means clustering algorithm and then utilized to estimate MU's location. The trials show that accuracy of the KNN algorithm can be improved by selecting correctly the number of the clusters, the number of the neighbors to be clustered and the center points in the k-means algorithm.

Bearing the findings of the clustered KNN prototype and the knowhow gained in the last two systems, a third enhancement scheme is also proposed with the same objective. In this system, a short term memory is integrated to the KNN algorithm to store the historical signal observations, which are used to refine the real-time signal strength values prior to their comparison with the fingerprints. The evaluations indicate that the performance of enhanced KNN outperforms KNN algorithm.

Overall, considering all three proposed system evaluation results, it can be concluded that the accuracy of the pure KNN algorithm is improved by integration of auxiliary methods such as k-means clustering and historical RSS memory. In other words, the proposed methods and techniques, at least up to a level, help to tackle the RSS-based positioning system's the main problem, which is the miscalculation of neighbors due to the signal fluctuations.

Moreover, an applied system named location-aware patient record access was also developed in this study. This system can sense the location of the physician by utilizing fingerprinting technique, and retrieve the relevant patient's medical data on to the physician's mobile device. The prototype system's location precision and usability evaluation results indicate that the proposed system is conceived as easy to use, accurate, and efficient tool.

As future work, a comprehensive location sensing system, which includes all the three proposed systems, could be developed and evaluated. In the scope of this Master thesis, unfortunately, the proposed systems were separately developed and tested in different time periods. Therefore, the performance gain of the proposed systems– especially Clustered KNN and KNN-STM – could not be integrated. Thus, this comprehensive system could make use of these proposed systems simultaneously and provide much more accurate estimation results.

Nowadays, besides the WiFi adapter, most of the Smartphones and PDAs are equipped with extra sensors, such as accelerometer and digital compass. The information collected from these sensors can be interpreted as user's walking distance and orientation. Thus, the proposed positioning system, which entails all three techniques, can benefit from these sensors to improve the accuracy of the estimated location.

Last but not least, it has be emphasized that one of the disadvantages of using a fingerprinting based method is the high labor and time cost for collecting the required fingerprint database during the offline phase. Therefore, a successful algorithm to facilitate a calibration-free radio map generation would decrease the workload on the offline phase of fingerprinting technique.

REFERENCES

1. Sun, G., Chen, J., Guo, W. and Liu, K. J. R., “Signal processing techniques in network-aided positioning: a survey of state-of the-art positioning desig.”, *IEEE Signal Processing Magazine*, Vol. 22, no. 4, pp. 12-23, 2005.

2. Altintas, B., *Location-aware Access to Patient Records*, Under-graduate Thesis Report, Yeditepe University, 2009.

3. Hazas M. and Ward, J., “A novel broadband ultrasonic location system”, *Proc. 4th Int. Conf. Ubiquitous Computing*, pp. 264-280, 2002.

4. Want R., Hopper, A., Falcao V., and Gibbons, J., “The active badge location system”, *ACM Transactions on Information Systems*, Vol. 10, no 1, pp. 91-102, 1992.

5. Ni, L. M., Liu, Y., Lau Y.C., and Patil, A. P.,“Landmarc: indoor location sensing using active rfid”, *First IEEE International Conference on Pervasive Computing and Communications*, pp. 407-415, 2003.

6. Bahl P. and Padmanabhan, V. N., “RADAR: an in-building RF-based user location and tracking system”. *NFOCOM 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2, pp. 775-784, 2000.

7. Bodker, K., Jorgensen, B. and Pedersen, E. “Community Based Wi-Fi Positioning”, M.S. thesis, Aalborg University, 2006.

8. Llombart M., Ciurana M. and Barcelo-Arroyo F., “On the scability of a novel WLAN positioning system based on time of arrival measurements”, *Proceedings of the 6th Workshop on Positioning, Navigation and Communication*, pp. 15-21, 2008.

9. Yamasaki, R., Ogino, A., Tamaki, T., Uta, T., Matsuzawa N., and Kato, T., "TDOA location system for IEEE 802.11 b WLAN", *IEEE Wireless Communications and Networking Conference*, Vol. 4, pp. 2338-2343, 2005.

10. Yang, C., Huang Y. and Zhu, X. "Hybrid TDOA/AOA method for indoor positioning systems", The Institution of Engineering and Technology Seminar on Location Technologies, pp. 1-5, 2007.

11. Ekahau, Inc., Ekahau Positioning Engine, [Online], http://www.ekahau.com/?4140 [retrieved 30th June 2013], 2013.

12. Skyhook Wireless, Inc., Location Technology, [Online] http://www.skyhookwireless.com [retrieved 30th February 2013], 2013.

13. Li, B., Salter, J., Dempster A. G. and Rizos C., 'Indoor Positioning Techniques Based on Wireless LAN', First IEEE International Conferenec on Wireless Broadband and Ultra Wideband Communications, pp 1-7, 2006.

14. Lin, T. and Lin, P., "Performance Comparison of Indoor Positioning Techniques based on Location Fingerprinting in Wireless Networks", *International Conference on Wireless Networks, Communications and Mobile Computing*, pp. 1569-1574, 2005.

15. Taheri A., Singh, A. and Agu, E. "Location Fingerprinting on Infrastructure 802.11Wireless Local Area Networks (WLANs) using Locus", *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, 2004.

16. Gunawan, M., Li, B., Gallagher, T., Dempster, A. G. and Restcher, G., "A new method to generate and maintain a WiFi fingerprinting database automatically by using RFID", *International Conference on Indoor Positioning and Indoor Navigation*, 13-15 Nov 2012, pp. 1-6, 2012.

17. Koweerawong, C., Wipusitwarakun, K. and Kaemarungsi, K., “Indoor Localization Improvement via Adaptive RSS Fingerprinting Database”, *International Conference on Information Networking (ICOIN),* pp. 412-416, 2013.

18. Li, B., Dempster, A., Rizos, C. and Barnes, J., “Hybrid method for localization using WLAN”, *Spatial Sciences Conference*, pp 341- 350, 2005.

19. Windows Embedded Developer Center., *Building a Wi-Fi Discovery Application with the .NET Compact Framework 2.0,* [Online] http://msdn.microsoft.com/en-us/library/aa446491.aspx [retrieved 30th June 2013], 2013.

20. Jimenez, A. and Abdiel M. S., *M-loc: Real Time Location System for Wireless Mesh Networks*, Thesis, University of Puerto Rico, Mayaguez (Puerto Rico), 2009.

21. Preuss M. and Thomas, S., “Wireless, Mesh and Ad-hoc Networks; Military Convoy Location and Situation Awareness”, *Sarnoff Symposium*, pp 1-5, 2008.

22. Movaghar, S. R., “Universal Vehicular Ad-hoc Network Design: The Case for Dynamic Addressing and Relative Positioning System”, *International Conference on Application of Information and Communication Technologies (AICT)*, pp 1-5, 2009.

23. Riggio, R., Miorandi, D., Chlamtac, I., Scalabrino, N., Gregori, E., Granelli, F. and Yuguang, F., “Hardware and Software Solutions for Wireless Mesh Network Test Beds”, *Communications Magazine, IEEE,* Vol 46, pp 156-162, 2008.

24. OLSRD project, An *Ad-hoc Wireless Mesh Network Daemon*, [Online] http://www.olsr.org [retrieved 30th of June 2013], 2013.

25. Moviquity Inc., *OLSR PDA & PC Daemon*, [Online] http://www.moviquity.com [retrieved 30th of June, 2013], 2013.

26. Altintas, B. and Serif, T., “Clustered KNN: A WLAN-based indoor positioning algorithm enhanced with k-means clustering”, *European Wireless 2011*, 2011.

27. Ma, J., Li, X., Tao, X. and Lu, J. "Cluster filtered KNN: A WLAN-based indoor positioning scheme", *World of Wireless, Mobile and Multimedia Networks WoWMoM*, pp. 1-8, 2008.

28. Sun, Y., Xu Y., Ma L. and Deng, Z., "KNN-FCM Hybrid Algorithm for Indoor Location in WLAN", *Power Electronics and Intelligent Transportation System PEITS*, Vol. 2, pp. 251-254, 2009.

29. Roshanaei, M. and Maleki, M., "Dynamic-KNN: A novel locating method in WLAN based on Angle of Arrival", *Industrial Electronics & Applications ISIEA*, Vol. 2, pp. 722-726, 2009.

30. Xu, R. and Wunsch, D., "Survey of clustering algorithms", *IEEE Transactions on Neural Networks*, vol. 16, pp. 645–678, 2005.

31. Altintas, B. and Serif, T., "Indoor Location Detection with a RSS-based Short Term Memory Technique (KNN-STM)", *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, 2012.

32. Khodayari, S., Maleki, S. and Hamedi, E., "A RSS-bsed fingerprinting method for positioning based on historical data" *Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pp: 306-310, 2010.

33. Xiang, Z., Song, S., Chen, J., Wang, H., Huang, J. and Gao, X. "A wireless LAN-based indoor positioning technology", IBM *Journal of Research and Development*, Vol. 48 , pp. 617-626, 2004.

34. Committee of Quality Health Care in America IoM. Crossing the quality chasm. National Academy Press, Washington, DC, USA, 2001.

35. Altintas, B. and Serif, T., "Location-aware Patient Record Access: Patient Identification Using Fingerprinting Technique", World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium, 2011.

APPENDIX A: K-MEANS CLUSTERING IMPLEMENTATION IN THE FINGERPRINTING

Algorithm A.1. k-means clustering implementation in the fingerprinting

```
if (isCLS)
{
        List<Classes.ClusterNode> cluster1 = new
List<termProject1.Classes.ClusterNode>();
        List<Classes.ClusterNode> cluster2 = new
List<termProject1.Classes.ClusterNode>();
        Classes.Point centroid1 = new
termProject1.Classes.Point();
        Classes.Point centroid2 = new
termProject1.Classes.Point();
        int min = 60, minIndex = 0, max = -1, maxIndex = 0, mean
= 0, meanIndex = 0, temp1 = 100;

        for (int i = 0; i < N; i++)
        {
             mean += sevenNode[i].observation.id;
             if (sevenNode[i].observation.id > max)
             {
                  max = sevenNode[i].observation.id;
                  maxIndex = i;
             }
             if (sevenNode[i].observation.id < min)
             {
                  min = sevenNode[i].observation.id;
                  minIndex = i;
             }
        }
        mean /= N;
        for (int i = 0; i < N; i++)
        {
             int dif = Math.Abs(mean -
sevenNode[i].observation.id);
             if( dif < temp1)
             {
                  temp1 = dif;
                  meanIndex = i;
             }

        }

        centroid1 = sevenNode[meanIndex].observation.pt;
        if ((mean - min) > (max - min))
              centroid2 = sevenNode[minIndex].observation.pt;
        else
              centroid2 = sevenNode[maxIndex].observation.pt;

        for (int i = 0; i < N; i++)
        {
```

```
                if (calculateDistance(centroid1,
sevenNode[i].observation.pt) < calculateDistance(centroid2,
sevenNode[i].observation.pt))
                {
                    cluster1.Add(sevenNode[i]);
                }
                else
                    cluster2.Add(sevenNode[i]);
        }
        if ((max - min) < N + 2)
        {
                userPosition = new termProject1.Classes.Point();
                int s = 0;
                for (int i = 0; i < 4; i++)
                {
                    userPosition.x +=
(sevenNode[i].observation.pt.x) * (int)sevenNode[3 -
i].distance;
                    userPosition.y +=
(sevenNode[i].observation.pt.y) * (int)sevenNode[3 -
i].distance;
                    s += (int)sevenNode[i].distance;
                }
                userPosition.x /= s;
                userPosition.y /= s;
         }
         else
         {
                while (true)
                {
                List<Classes.ClusterNode> c1 = new
List<termProject1.Classes.ClusterNode>();
                List<Classes.ClusterNode> c2 = new
List<termProject1.Classes.ClusterNode>();
                if (cluster1.Count == 0 || cluster2.Count == 0)
                     break;
                else
                {
                     int x = 0, y = 0, s = 0;
                     for (int i = 0; i < cluster1.Count; i++)
                     {
                         x += cluster1[i].observation.pt.x *
(int)cluster1[i].distance;
                         y += cluster1[i].observation.pt.y *
(int)cluster1[i].distance;
                         s += (int)cluster1[i].distance;
                     }
                     centroid1.x = x / s;
                     centroid1.y = y / s;
                     x = 0; y = 0; s = 0;
                     for (int i = 0; i < cluster2.Count; i++)
                     {
                         x += cluster2[i].observation.pt.x *
(int)cluster2[i].distance;
                         y += cluster2[i].observation.pt.y *
(int)cluster2[i].distance;
                         s += (int)cluster2[i].distance;
                     }
                     centroid2.x = x / s;
```

```
                        centroid2.y = y / s;

                        for (int i = 0; i < N; i++)
                        {
                            if (calculateDistance(centroid1,
sevenNode[i].observation.pt) < calculateDistance(centroid2,
sevenNode[i].observation.pt))
                            {
                                c1.Add(sevenNode[i]);
                            }
                            else
                                c2.Add(sevenNode[i]);
                        }
                        if (compareCluster(cluster1, c1))
                        {
                             break;
                        }
                        else
                        {
                             cluster1 = c1;
                             cluster2 = c2;
                        }

                  }
            }
}
```

APPENDIX B: USER QUESTIONNAIRE

Intelligent Hospital Application

Software Evaluation Form

User Questionnaire

Evaluator Details	
Name:	**Age:**
Role:	**Sex: M / F**

Please tick (☑) the appropriate box below considering

5- Strongly Agree, 4- Agree, 3- Neutral, 2- Disagree, 1- Strongly Disagree

Effectiveness					
The PDA is easy to carry around with me	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
The medical application is intuitive	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
A user manual for the application would be helpful	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Patient record download times could be enhanced	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
I could use the PDA for other activities in my life	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
It is useful to have ward-based access to patient records	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
I still believe medical personnel should keep hard copies of patient records	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Transferring data from the PDA application to main database is easy	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
It is useful to have access to patient prescription history	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Having access to previous laboratory test results is helpful	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Having access to patient health history is helpful	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Having the ability to record/play voice messages is helpful	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Having access to radiology test results are useless	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐

Ease of Use					
The application's user interface is consistent and easy to follow	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
I can navigate through the program with difficulty	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Program responds appropriately to any erroneous input	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
It is easy to make mistakes using this program	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Text on each screen is clear	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Character recognition/on-screen keyboard is easy to use	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
It is easy to make new laboratory & radiology tests requests	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐

Interface and Media Quality					
Visual inspection of medical images (MRI, X-RAY etc.) is easy	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
It is difficult to navigate through tests results	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
It is easy to understand how to record/play voice message	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
Working on a multi-tab user interface is cumbersome	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐
The recordings are comprehensible when they are played through PDA speakers	1 ☐	2 ☐	3 ☐	4 ☐	5 ☐

Overall					
Summary Evaluation	**Poor** ☐	**Fair** ☐	**Satisfactory** ☐	**Good** ☐	**Excellent** ☐

Comments/Notes/Request:……………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
……………………………………………

Figure B.1. User questionnaire