UBIQUITOUS INTERACTION ON WIRELESS MOBILE DEVICES

SIDDHARTH SINGH

NATIONAL UNIVERSITY OF SINGAPORE

2004

UBIQUITOUS INTERACTION ON WIRELESS MOBILE DEVICES

SIDDHARTH SINGH
(B.Tech(Hons.), I.I.T., Kharagpur)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Abstract

This thesis is based on the research work done on ubiquitous interaction for mobile devices.

In order to improve *human – mobile device* interaction, we first focus on developing a location tracking technology that can be used easily by *location aware* mobile devices. We present an indoor location tracking system that makes use of the existing wireless LAN (WLAN) and does not need any additional infrastructure. We present a new location region based approach for location detection that allows faster real time tracking. A location region approach is particularly useful for *location aware* devices which can directly obtain information about their location area. We use a *mobile device assisted, centrally based* architecture which causes minimal disruption of normal WLAN traffic and allows transparent updates to the tracking system. We compare the tracking accuracies using different pattern recognition techniques and present our results.

We then look into improving *human – mobile device* interaction through development of new user interfaces. We focus on consumer mobile phones and investigate their use as an augmented reality (AR) interface for real world interaction. We describe the design and implementation of our core system and also present a user study on the effectiveness and feasibility of augmented reality on mobile phones. This project makes use of readily available consumer mobile phones without making any hardware modifications to them.

Thirdly, we focus on improving the interaction among mobile device users through the development of an *Anywhere Gaming* system. This allows mobile device users to play a game with other mobile device and stationary device users in real time, thus greatly improving interaction.

Six papers based on this thesis work have been published at international conferences. Two US patents have also been filed.

Acknowledgements

I wish to thank Dr. Adrian David Cheok for his patient and invaluable guidance in my research work. It was really a great experience working with him. I would also like to thank Dr. Rahim Leyman for his help and support.

My colleagues at the Mixed Reality Lab have been wonderful and they have created a friendly, cooperative and a very cheerful atmosphere to work in. It was fun to work with them.

I am really thankful to my friend Sandeep, who helped me in innumerable ways during my stay in Singapore.

Finally, I would like to thank my parents and my brother for all their love and encouragement. They are the reason for all my success. I dedicate this work to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Pervasive computing refers to the ubiquitous presence of computing in both mobile and embedded environments, with the ability to access and update information anywhere, anyplace and anytime. This idea has been around for a long time, but only now is pervasive computing truly taking root, mainly due to the proliferation of wireless networks and the development of lighter and more powerful mobile devices.

In this thesis we focus our attention on ubiquitous interaction on mobile devices. We use the term *interaction* for *human – mobile device* interaction, as well as for interaction among mobile device users.

Human – mobile device interaction can be improved in two ways:

1. By developing smarter mobile devices
2. By improving the user interface of mobile devices

Interaction can be greatly enhanced through the use of smart mobile devices that are more responsive to the needs of their owners. A leading way of achieving this goal is through the development of *location-aware* mobile devices. These devices can change their behavior in subtle ways depending on the physical location of the user. This leads to a more enjoyable experience by the user. But these location-aware devices need some location tracking technique that will allow them to know their location. Thus location tracking techniques are vital to the development of location-aware mobile devices. In this thesis we will investigate the location tracking problem for indoor environments.

Interaction with mobile devices can also be enhanced through the development of suitable interfaces that are user-friendly and natural to use. We will investigate the use of augmented reality (AR) as user interfaces for mobile devices.

We are also interested in improving interaction among mobile device users. For this purpose, we will investigate the use of networked computer games. We are particularly interested in gaming systems that allow *mobile device users* as well as *stationary device users* to interact with each other in a seamless way.

1.1 Improving interaction through location-aware systems

Public wireless data networks like GPRS[1] and CDMA[2] are being rolled out, while inexpensive Wi-Fi networks are rapidly turning up in corporate, private and public locations. Several companies worldwide are working on extending the range of Wi-Fi from meters to kilometers through the use of smart antennas and mesh network technologies. The FCC[3] has also recently allocated unlicensed spectrum for ultrawideband networks, which promise much higher speeds and less interference than Wi-Fi networks. On the hardware front, mobile devices are growing more capable and less expensive. New form factors like tablets and smart phones are appearing, color displays are becoming standard, and pen input is getting better. Computing power is also being embedded into everyday things such as cars, roads, cameras and vending machines. In the future, many of these devices will be on the network, sensing, monitoring, controlling and informing us about our environment. These recent advances have spurred an interest in *location-aware computing.*

Location-aware computing is a recent field of interest that has great potential in the areas of personal security, navigation, tourism and entertainment. It involves the use of location-aware devices that can detect their physical position, or allow other devices to detect their position. Based on this location detection ability they can provide a variety of services to the user. Some examples of location-

---

[1] General Packet Radio Service

[2] Code Division Multiple Access

[3] Federal Communications Commission, an independent US government agency

based services are those that provide answer to questions such as “Where am I?”, or “Where is the nearest restaurant?” Therefore there is a need for having accurate location positioning systems [1]. GPS is the most popular positioning system but its drawback is that it does not work well in indoor environments. Although different tracking systems have been proposed for indoor environments, they require additional infrastructure to be installed. Additionally, most of them require specialized receivers and transmitters that are never carried by common users. This has stifled their growth beyond the research stage.

In order to develop a location tracking system that will be readily adopted by common users and developers of location-aware applications, it is necessary to develop a tracking system that can be setup easily with minimal cost. Wi-Fi, or WLAN, is gaining widespread popularity and many corporate offices and public locations provide Wi-Fi coverage. Using the wireless signals, it is possible to come up with a location tracking system that is cheap and can easily be setup. In the first part of this thesis, we will deal with developing a WLAN-based tracking system.

1.2 Improving interaction through better user interfaces

Apart from improving interaction through location-aware mobile devices, it is also important to develop suitable interfaces for mobile devices so that *human – mobile device* interaction becomes intuitive.

The current trend towards pervasive computing suggests future work environments will comprise of a range of information displays and interaction devices. These will include normal desktop computers or even notebook computers together with 3D immersive displays. A lot of research has been focused on the creation of new interaction systems using Augmented Reality (AR. Augmented Reality (AR) is a variation of Virtual Reality (VR). VR technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composted with the real world. Therefore, AR supplements reality, rather than completely replacing it. Ideally, it would appear to the user that the virtual and real objects coexisted in the same space. The superimposition of graphics, audio, and other sense-enhancements into the real world would result in a single scene commonly referred to as an "augmented" scene (see Figure 1.1). In an augmented reality scene, what the viewer sees is part real, part virtual. Such augmented scenes can be used to create an enhanced environment.

Enhanced environments aim to enhance user's daily experiences by providing them with AR services that blend into their surroundings. Such environments are natural complements to mobile computing, since they can assist the user directly in several situations. One of the most important aspects of such an environment is how users interact with it. This should be as intuitive and unobtrusive as possible, which makes traditional computing interfaces such as keyboards, mice and monitors undesirable.

Figure 1.1: “Augmented” scenes

Techniques that allow a user to interact with an enhanced environment in a more natural manner are preferable. Thus there is a need to design (or select) suitable mobile devices that enable users to interact with this enhanced environment. This project explores the concept of using a mobile phone as an Augmented Reality (AR) interface for real world interaction (see figure 1.2).

Figure 1.2: Mobile phone as an Augmented Reality interface

1.3 Improving interaction among mobile device users

So far we have focussed on improving the *human – mobile device* interaction through development of *location-aware* devices and better user interfaces. However, it is also important to improve interaction among mobile device users.

This can be done through the development of interesting networked applications that compel mobile device users to use it and interact with other users. Computer games and console based video games are hugely popular and attract large number of users. Their popularity has grown rapidly over the past few years. The main contributing factor has been the development of multiplayer games that allow humans to play against each other. Clearly, computer games can be used as an important tool in improving interaction among mobile device users.

The development of suitable games for mobile devices is more challenging due to the small size of the screens and the fact that mobile devices are not powerful enough to support CPU and graphics intensive games. Developing multiplayer games for mobile devices is even more complicated due to the existence of a wide variety of communication techniques (GPRS, Bluetooth, CDMA). Thus, a truly useful game for mobile users would be one that provides a seamless gaming experience across multiple device types and connectivity techniques. This would allow creation of a larger playing field and thus greatly promote interaction.

The third part of this thesis investigates the concept of *Anywhere Gaming* system that can allow players on mobile devices to interact with other mobile device players as well as players on stationary devices.

1.4 Contributions of this Thesis Work

The main objective of this thesis is to look at ways of improving ubiquitous interaction for mobile devices. We have made the following contributions towards this goal:

1. Development of a location tracking system in order to support *location-aware* mobile devices.

   Our location tracking system is designed for indoor environments and uses the existing WLAN infrastructure. Thus the system can be setup easily and is affordable.

2. Development of augmented reality (AR) interface for commercial mobile phones.

   We developed the core technology and built two AR applications using it. We also conducted user surveys to study the effectiveness and feasibility of augmented reality on mobile phones.

3. Development of a networked game for mobile phone users.

   Our gaming system, the *Anywhere Gaming System* allows mobile device users to play against each other and also against stationary device users (PC users).

This allows interaction of humans on different devices (stationary or mobile), irrespective of the communication channel being used by their device.

1.4.1 Publications

This thesis is based on the following patents and research papers.

- **Pending Patents**

  1. Adrian David Cheok, Siddharth Singh, Ng Guo Loong, "Augmented Reality for Mobile Phones", applying for Singapore patent and US patent through the Industry & Technology Relations Office (INTRO), NUS.

  2. Siddharth Singh, Adrian David Cheok, Ng Guo Loong, "Marketing Applications for Augmented Reality on Mobile Phones", applying for Singapore patent and US patent through the Industry & Technology Relations Office (INTRO), NUS.

- **Published/Accepted Conference Papers**

  1. Siddharth Singh, Adrian David Cheok, Soh Chor Kiong, "Anywhere Gaming System", in Proc. of Australian Workshop on Interactive Entertainment, (IE2004), Sydney, 2004, pp. 23-26.
     (Status: Published).

2. Siddharth Singh, Adrian David Cheok, Guo Loong Ng, Farzam Farbiz, "3D Augmented Reality Comic Book and Notes for Children using Mobile Phones", ACM Interaction Design & Children, IDC 2004, June 1-3, 2004, College Park, Maryland, USA.

   (Status: Published).

3. Siddharth Singh, Adrian David Cheok, Soh Chor Kiong "A Step Towards Anywhere Gaming", ACM Advances in Computer Entertainment Technology, ACE 2004, June 3-5, 2004, Singapore.

   (Status: Published).

4. Siddharth Singh, Adrian David Cheok, Guo Loong Ng, Farzam Farbiz, "Augmented Reality Post-It System", ACM Advances in Computer Entertainment Technology, ACE 2004, June 3-5, 2004, Singapore.

   (Status: Published).

5. Siddharth Singh, Adrian David Cheok, Guo Loong Ng, Farzam Farbiz, "AR Post-It: A Location Dependent Messaging System", Mobility Conference 2004, Singapore, 2-4 August, 2004.

   (Status: Published).

6. Siddharth Singh, Adrian David Cheok, Guo Loong Ng, Farzam Farbiz, "A Ubiquitous, Location dependent Messaging and Information Access System", Mobile and Ubiquitous Information Access Workshop, Glasgow, Scotland, 13th September, 2004.

(Status: Accepted, to be published).

- **Journal Papers**

  1. Siddharth Singh, Adrian David Cheok, "WLAN-based Location Tracking Using *Location Region* Approach".

     (Status: Submitted and being reviewed).

1.5 Thesis Organization

The structure of this thesis work is organized as follows:

We first describe the WLAN based tracking system in chapter 2. We describe our location tracking approach, the system architecture, and the location tracking algorithms.

Then in chapter 3 we describe the augmented reality system for mobile phones. We present our prototype system and also describe the two AR applications that were developed using it. We also present the user surveys that were conducted to gain feedback on our system.

In chapter 4, we describe our *Anywhere Gaming* system. This system allows mobile device and stationary device users to simultaneously play together, thus promoting interaction between them. We describe the system architecture and provide the implementation details.

Finally in chapter 5 we provide a conclusion.

Chapter 2

WLAN Based Indoor Location Detection

2.1 Introduction

The proliferation of mobile devices and the advancements in wireless communication techniques have created a pervasive computing environment. This, in turn, has led to an interest in *location-aware* systems and services. Location-aware systems can be extremely useful in a variety of contexts. Some of the potential applications are:

- *Smart devices:* A smart mobile device that can change its behavior and user interface according to the physical location would be greatly enhance the user's experience. For example, smart phones that can automatically go into the *silent mode* in meeting rooms.

- *Computer tourist guides:* In museums, tourists can carry smart mobile devices that will display information according to the physical location of the tourist.

- *Security:* In places of high security risk, such as airports, it may be desirable for security agencies to know the physical location of their personnel at all times. This would allow them to effectively respond to emergencies.

Clearly, location-aware services require some mechanism for tracking the location of the mobile users. For outdoor locations, the Global Positioning System (GPS) [2] allows reliable location tracking in any part of the world. GPS uses a network of 24 satellites (figure 2.1) that was put into orbit by the U.S. Department of Defence. GPS receivers are available commercially and they can use the signals from the satellites to calculate the exact location of the user. There is no subscription or setup fee to use the GPS. This is the key reason for the popularity of GPS - the infrastructure required for location tracking is already in place.

Figure 2.1: 24 GPS satellites orbit the earth.

Today, GPS is used widely by commercial aircrafts and ships for navigation. It is also being used for navigation across city-wide areas. For example, Kane Gear

[3] has developed an application called *Kane Car Pilot* that uses GPS to provide location information on a Pocket PC. The information is displayed using a map of city (figure 2.2). Although GPS is ideal for outdoor location tracking, it requires clear open spaces to work and therefore cannot be used inside buildings.

Figure 2.2: The *Kane Car Pilot* application for Pocket PC type PDAs. It uses GPS to provide navigation information for automobile drivers.

For indoor locations, there is no preferred location tracking system that is as popular as GPS is for outdoor locations. The suitable contenders for indoor environments are infra-red, ultrasonic, and radio frequency (RF) based systems. Among these, the infra-red based systems are limited in range and require a clear line-of-sight between the transmitter and receiver to work properly. The ultrasonic based systems require installation of extra hardware in order for them to work, and thus are not easy to setup. RF-based systems also require transmitters and receivers, but with the growing popularity of Wi-Fi, technically known as IEEE 802.11 Wireless LAN, the infrastructure needed for an RF-based tracking system is already in place in most indoor locations. Thus a WLAN based location tracking system has the following advantages:

1. It does not require any hardware installation.
2. It does not require use of specialized mobile devices (for use as transmitters/receivers).
3. A WLAN based tracking systems adds to the value of the wireless network, by allowing users to use it for data services as well as for location-aware services. Thus, the organization that owns the WLAN will be open to make the necessary software updates that are needed for the tracking system.

We therefore chose to develop a WLAN based tracking system.

2.1.1 Location Detection Techniques

All the location tracking systems, whether for indoor locations or outdoor locations, use a few basic techniques to track the mobile user. Hightower and Borellio [4] have proposed the following taxonomy for these location sensing techniques:

1. Triangulation
2. Scene Analysis
3. Proximity

Figure 2.3 gives a summary of this taxonomy.

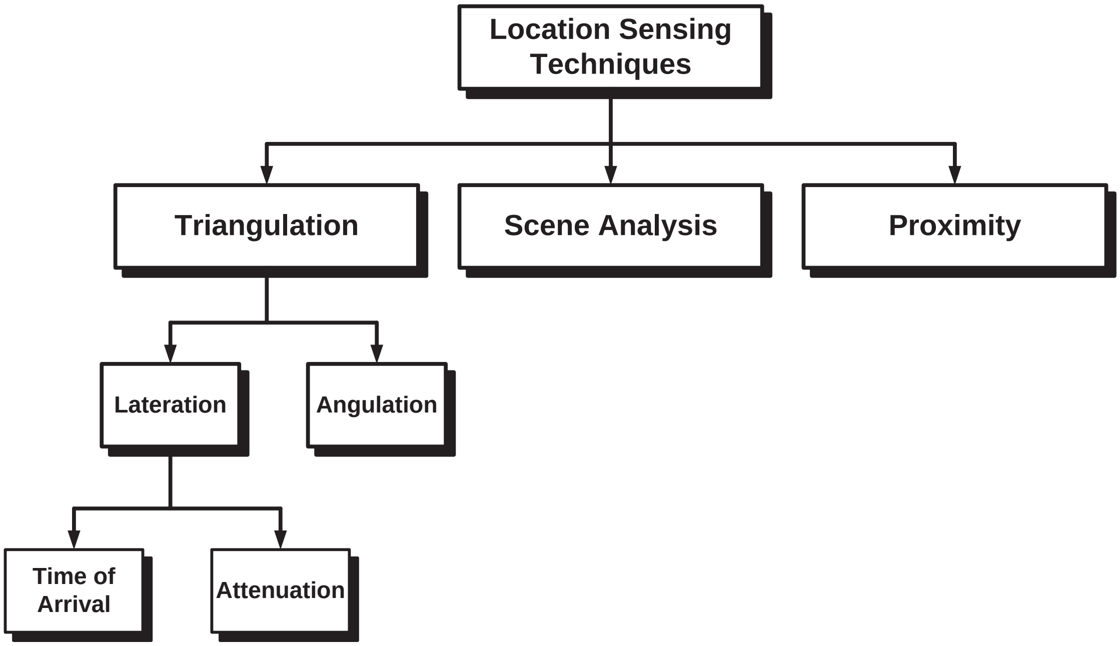


Figure 2.3: Taxonomy for location detecting techniques

2.1.2 Triangulation

This technique makes use of the geometric properties of triangles to compute object locations. Either distance measurements or angle measurements can be used in triangulation. Thus triangulation can be further subdivided as *lateration* and *angulation*.

Lateration

Lateration computes the position of an object by measuring its distance from mulfiple reference positions. Calculating an object's position in two dimensions requires distance measurements from 3 non-collinear points as shown in Figure 2.4. In 3 dimensions, distance measurements from 4 non-coplanar points are required. Domain-specific knowledge may reduce the number of required distance measurements.

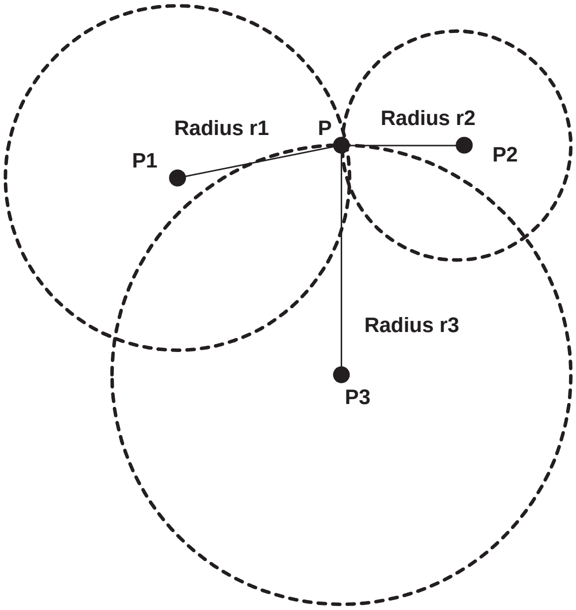


Figure 2.4: Lateration: P1,P2 and P3 are the reference points used to calculate the position of point P.

The distance to the reference points can be measured by the following techniques:

1. **Time-of-arrival :** In this method, distance between a reference point, say R, and the point of interest, say P, is calculated by measuring the time it takes to travel from R to P at a constant velocity. This is done by transmitting a signal at one point and receiving it at the other. By measuring the time difference between the transmission and reception of the signal, the distance between the two points can be measured. Time-of-arrival location sensing systems include GPS [2], the Active Bat Location System [5], the Cricket Location Support System [6], Bluesoft [7], and PulsON$^{\text{TM}}$ Time Modulated Ultra Wideband technology [8].

2. **Attenuation :** Attenuation is the weakening of a signal as it propagates away from its source. Given a function correlating attenuation and distance for a type of emission and the original strength of the emission, it is possible to estimate the distance from an object to some point P by measuring the strength of the emission when it reaches P. For example, a free space radio signal emitted by an object will be attenuated by a factor proportional to $1/r^2$ when it reaches point P at distance r from the object.

   In environments with many obstructions such as an indoor once space, measuring distance using attenuation is usually less accurate than time-of-arrival. Signal propagation issues such as reflection, refraction, and multipath cause the attenuation to correlate poorly with distance resulting in inaccurate and imprecise distance estimates. The SpotON [9] ad hoc location system implements attenuation measurement using low-cost tags. SpotON tags use radio signal attenuation to estimate inter-tag distance and exploits the density of tag clusters and correlation of multiple measurements to mitigate some of the signal propagation difficulties.

Angulation

Angulation is similar to lateration except that angles are used instead of distances for determining the position of an object. In general, two dimensional angulation requires two angle measurements and one length measurement such as the distance between the reference points. In three dimensions, one length measurement, one azimuth measurement, and two angle measurements are needed to specify a precise position. Angulation implementations sometimes choose to designate a constant reference vector (e.g. magnetic north) as 0°.

Angulation based location detection is used by phased array antennas. Multiple antennas are kept at a known separation and the time of arrival of the signal is measured. Then, using the difference in arrival times of the signal and the known seperation of the antennas, it is possible to calculate the angle from which the signal was emitted.

2.1.3 Scene Analysis

The scene analysis location sensing technique uses features of a scene observed from a particular vantage point to draw conclusions about the location of the observer or of objects in the scene. Usually the observed scenes are simplified to obtain features that are easy to represent and compare (e.g., the shape of horizon silhouettes such as Figure 3 as seen by a vehicle mounted camera [10]). In *static* scene analysis, observed features are looked up in a predefined dataset that maps them to object locations. In contrast, *differential* scene analysis tracks the difference between successive scenes to estimate location. Differences in the scenes will correspond to movements of the observer and if features in the scenes are known to be at specific positions, the observer can compute its own position relative to them.

The advantage of scene analysis is that the location of objects can be inferred using passive observation and features that do not correspond to geometric angles or distances As we have seen, measuring geometric quantities often requires motion or the emission of signals, both of which can compromise privacy and can require more power. The disadvantage of scene analysis is that the observer needs to have access to the features of the environment against which it will compare its observed scenes. Furthermore, changes to the environment in a way that alters the perceived features of the scenes may necessitate reconstruction of the predefined dataset or retrieval of an entirely new dataset.

The scene itself can consist of visual images, such as frames captured by a wearable camera [11], or any other measurable physical phenomena, such as the electromagnetic characteristics that occur when an object is at a particular position and orientation. The Microsoft Research RADAR location system is an example of the latter [12]. The observed features, signal strength values in this case, correlate with particular locations in the building but do not directly map to geometric lengths and angles describing those locations.

2.1.4 Proximity

A proximity location sensing technique entails determining when an object is "near" a known location. The object's presence is sensed using a physical phenomenon with limited range. There are three general approaches to sensing proximity

1. **Detecting physical contact.** Detecting physical contact with an object is the most basic sort of proximity sensing. Technologies for sensing physical contact include pressure sensors, touch sensors, and capacitive field detectors.

Capacitive field detection has been used to implement a Touch Mouse [13] and Contact, a system for intra-body data communication among objects in direct contact with a person's skin [14].

2. **Monitoring wireless cellular access points.** Monitoring when a mobile device is in range of one or more access points in a wireless cellular network is another implementation of the proximity location technique. Examples of such systems include the Active Badge Location System [5] and the Xerox ParcTAB System [15], both of which use diffuse infrared cells in an office environment.

3. **Observing automatic ID systems.** A third implementation of the proximity location sensing technique uses automatic identification systems such as credit card point-of-sale terminals, computer login histories, land-line telephone records, electronic card lock logs, and identification tags such as electronic highway E-Toll systems, UPC product codes, and injectable livestock identification capsules [16]. If the device scanning the label, interrogating the tag, or monitoring the transaction has a known location, the location of the mobile object can be inferred.

2.2 Related Work

We will now describe the related work in *indoor* location tracking systems. We have categorized these tracking systems into *Non RF-based systems* and *RF-based* *systems*. This is illustrated in figure 2.5.

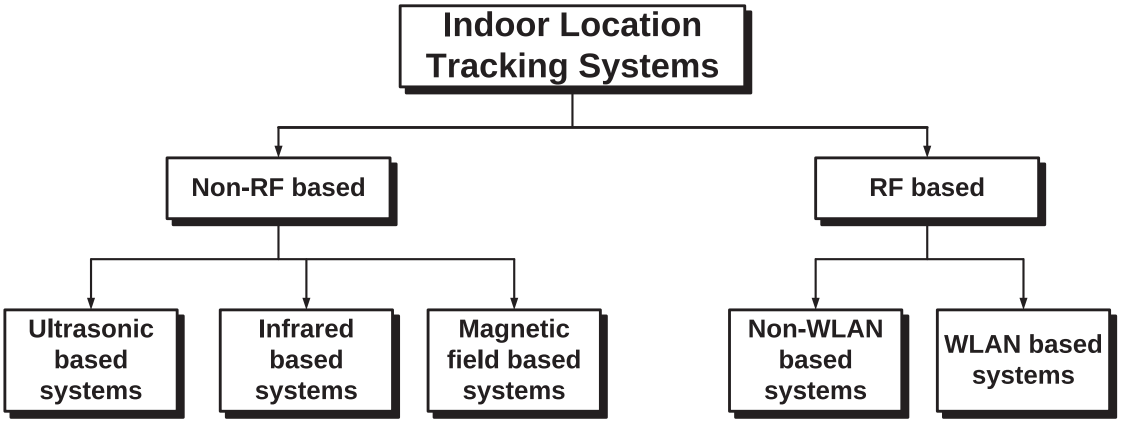


Figure 2.5: Location tracking systems for indoor environments

2.2.1 Non RF-based Systems

This category of location tracking systems make use of ultrasound, infra-red (IR), or magnetic fields.

Ultrasonic based systems

The Active Bat Location System [5] measures distance from indoor mobile tags, called Bats, to a grid of ceiling mounted ultrasound sensors. A Bat's 3-dimensional position can be determined using only 3 distance measurements because the sensors in the ceiling are always above the receiver. The geometric ambiguity of only 3 distance measurements can be resolved because the Bat is known to be below the sensors and not in the alternate possible position on the next floor or roof above the sensor grid.

While the Active Bat provides highly accurate location estimates, it is costly to setup due to the requirement of specialized ultrasound sensors.

Infrared based systems

Examples of IR based systems are the Active Badge Location System [17] and the Xerox PARCTAB System [15]. The PARCTAB system consists of palm-sized mobile computers that can communicate wirelessly through infrared (IR) transceivers to workstation-based applications. The infrared transceiver used in the PARCTAB project is shown in figure 2.6.

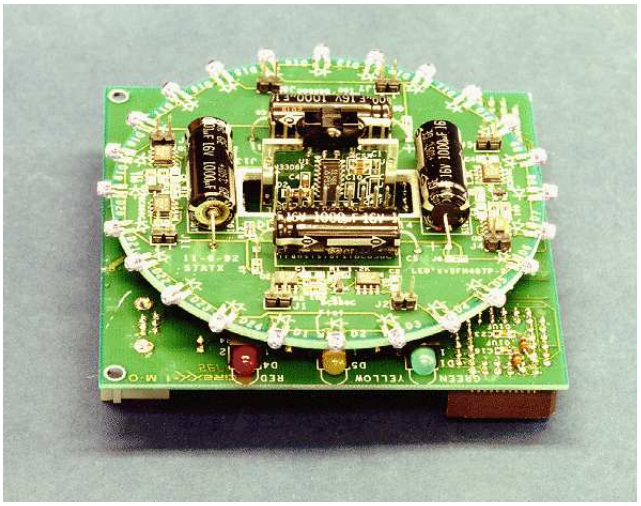


Figure 2.6: The PARCTAB transceiver.

In the Active Badge System [17], sensors are installed at various locations in a building. The mobile users need to wear a badge that emits an IR signal every 10 seconds. The sensors in the building pick up these transmissions and relay it to a central location manager software.

Another IR based system is described by Azuma [18]. The system requires attaching IR transmitters to the ceiling at known locations. An optical sensor on a head mounted unit senses the IR beacons, which enables the system software to determine the user's location.

All these IR based tracking systems have the disadvantage that specialized hardware needs to be installed within the buildings. Another problem with IR based tracking systems is that they perform poorly in the presence of direct sunlight. Thus the system's accuracy will be affected in a room with windows.

Magnetic field based systems

A system based on pulsed direct current (DC) magnetic fields has been developed by Ascension Technology Corporation [19]. This requires placement of multiple sensors on a user's body (e.g., using data gloves). The output of these sensors is used to determine the user's location and orientation to a high degree of precision. Although this technique is popular in the computer animation industry, it is too expensive. Moreover, it requires attaching sensors to the person who needs to be tracked. These factors make it unsuitable for use by common users.

2.2.2 RF-based Systems

Non-WLAN based systems

One of the earliest RF-based tracking system was the *Duress Alarm Location System* (DALS) [20]. This requires the mobile user to wear a RF transmitter on her body. Receivers are installed throughout the installation. The received signal strength is then used to determine the user's location. The DALS system requires use of specialized mobile devices (transmitters) as well as the installation of specialized infrastructure.

Time Domain [8] has developed PulsON^{TM} chipsets and chip designs based upon the Time Modulated Ultra Wideband (TM-UWB) architecture. PulsON^{TM} technology transmits pulse trains of individual ultra-wideband 'pulses' at very precise time intervals. This results in a low power, noise-like signal that can transmit data, voice and video communications and can be used as a positioning and tracking device. TM-UWB radios can be used to measure distance in a method analogous to the approach taken by the Global Positioning Satellite System (GPS). TM-UWB radios can measure time with three orders of magnitude better resolution than GPS receivers. This permits sub-centimeter range resolution to be achieved quickly and with very little effort. One special advantage of TM-UWB is that because of the time resolution of the technology and the pulse nature of the transmitted signal, the system works especially well in the presence of multipath, e.g., inside buildings and other highly-cluttered environments.

The Cricket Location Support System [6] is another location tracking that uses RF and ultrasonic signals to perform location tracking. Although Cricket is not a pure RF based system, we have mentioned it under this section since use of RF was made. Cricket requires installation of small transmitter boards that can transmit RF and ultrasonic signals. These boards are mounted on the ceiling or high on the wall. Cricket also requires special receivers that can be built using off the shelf equipment.

WLAN based systems

One of the first WLAN based tracking tracking system was the RADAR tracking system [12] developed at Microsoft Research. RADAR uses a database of signal strength measurements created by observing the radio transmissions of an 802.11 wireless networking device at many positions throughout a building. The location of other 802.11 network devices is then computed by performing table lookup on the stored database. RADAR employs simple Euclidean distance measurements for estimating the user location.

Another WLAN based tracking system was developed by Roos et al. [21]. This tracking system uses the signals recorded in the offline stage to build a model for the distribution of signals at various location. A probability model was built for different locations. During the online tracking stage, the offline probability model was used to predict the location of the user. Bayes' method was employed in order to make use of the offline probability distribution model.

Youssef et al. [22] describe a WLAN based tracking system. The location sensing technology uses a joint clustering of access points and joint probability distributions and involves an offline and an online phase. In the offline phase signal measurements are taken at several points or training locations. At each training location, a model for the joint probability distribution of the visible access points at that location is stored. The training phase results in a collection of probability distribution models for the signals at each point. Maximum likelihood estimation was used to calculate these joint probabilities. The online phase consists of collecting samples from some access points at an unknown location. The strongest access points are used to determine the cluster to search within for the most probable location. Then Bayes' theorem is used to estimate the probability of each location within the cluster using the observed tuples and the collected data during the offline phase.

Newbury Networks, a Boston based company, has launched products based on WLAN based tracking [23]. They have developed a location-based product called “Digital Docent”, which provides location based content on a mobile client. Such information can be useful in museums, exhibitions and education campuses by providing dynamic information based on a users current location. This application is based on a core technology called LocaleServer. One of the key capabilities of LocaleServer is the ability to detect locations of wireless devices within range of the WLAN. The advantage that LocaleServer has above traditional networking tools is that it can detect all 802.11 traffic within range of the network, not just the traffic on the network. LocaleServer can also be used for tracking and monitoring all 802.11 devices and equipment [23]. Detailed technical information for the LocaleServer was unavailable.

Table 2.1 summarizes these WLAN based tracking systems.

Researchers	**Tracking Approach**	**Tracking Algorithm**	**Advantages & Disadvantages**
Bahl et al. [12]	Offline training by sampling signals at different physical locations	Euclidean distance measurements	Easy to implement.
Roos et al. [21]	Offline training by sampling signals at different physical locations	Probabilistic technique	Use of probabilistic technique provides greater information about the signal distribution
Youssef et al. [22]	Offline training by sampling signals at different physical locations	Probabilistic technique along with clustering of APs	Faster tracking since clustering of APs reduces the searching of the database
Newbury Networks [23]	Detailed technical information was unavailable.	Detailed technical information was unavailable.	Detailed technical information was unavailable.

Table 2.1: Summary of previous WLAN tracking systems.

2.3 Unique features of our WLAN tracking system

Our WLAN based tracking system differs from previous work in the following ways:

- We have adopted a *location region* based approach for tracking the location of a mobile user. Previous WLAN tracking systems have adopted a *coordinate system* based approach. The advantages of the *location region* approach are discussed in section 2.3.1.

- We present results about how the placement of APs affects the tracking accuracy. Previous research on WLAN based tracking systems has not investigated how the placement of APs affects the location estimation accuracy. In our work we have investigated this aspect.

- We implement our tracking system using an architecture that is scalable to allow tracking across multiple WLANs. It causes minimal disruption of normal WLAN data traffic and allows easy software updates of the tracking system. We use a *mobile device assisted, centrally based* architecture for WLAN based tracking system. Previous research on WLAN tracking systems has not discussed the architectural issues in detail. Thus, it is unclear how a location-aware application can interact with a WLAN based tracking system. For the development of location-aware applications, it is important to clearly spell out how the interaction between the *location-aware application* and the *location tracking system* takes place. In this thesis we discuss the system architecture in detail. We will recommend certain measures that will allow easy integration of multiple WLAN based location tracking systems. This would enable the development of truly ubiquitous location-aware services.

2.3.1 Advantages of the *location region* based approach

We propose using a *location region* based tracking system because of the following reasons:

1. It leads to a less complicated training process, compared to that of previous WLAN tracking systems.
2. It can better aid location-aware services and mobile devices in indoor envi-

ronments.

3. The tracking system gives the location information as a direct output. Thus no additional database needs to be checked.

4. It leads to faster real-time tracking.

Previous WLAN-based tracking mechanisms have used a coordinate system approach in which the user's location is represented in terms of coordinates, usually a 2-D cartesian coordinates. The drawback of this approach is that it complicates the training process. During the training process, signals need to be sampled at different physical locations. The person who is sampling the signals needs to specify his physical location in terms of coordinates. *But how does he know his location in terms of mathematical coordinates?* As an illustration, suppose the trainer is standing in front of his office to sample the signals. While he knows his location (*I am in front of my office*), how can he specify it mathematically?

An inconvenient and impractical solution is to measure the physical distances and specify the coordinates accordingly. An alternative solution, is to use a software application that can display a map of the location. The person will then be able to specify his location by clicking at the appropriate point on the map. However, this method requires that the physical location of the trainer correspond exactly to the *point-of-click* in the map. If the map is not accurate enough, it will be difficult for the trainer to find his location point on the map. Thus the wrong coordinates can be specified while training the system.

The second reason for favoring the *location region* approach is based on the needs of location-aware mobile devices and other location-aware services. In indoor environments, these location-aware devices and services do not need information about the precise location of a user. They only need to know if the user has entered (or left) a particular location region. As an illustration, consider a smart phone that can enter into the *silent mode* when the user is in the meeting room. For this, the phone only needs to know that the person has entered the meeting the room. It does not need to know the exact coordinates of the user.

The third reason for favoring the *location region* approach is the location information is obtained directly. In the coordinate system based approach, the tracking system gives only the coordinates of the user. However, this information by itself is not useful. The coordinates of the user have to be compared with the coordinates of various locations using another database. Only then the nearest location can be found. This is illustrated in figure 2.7.

The fourth reason for favoring the *location region* approach is that it leads to faster real-time tracking. This is because a smaller database needs to be maintained for the *location region* approach.

These observations clearly show that that a *location region* based tracking approach will be more practical for location-aware devices in indoor environments.

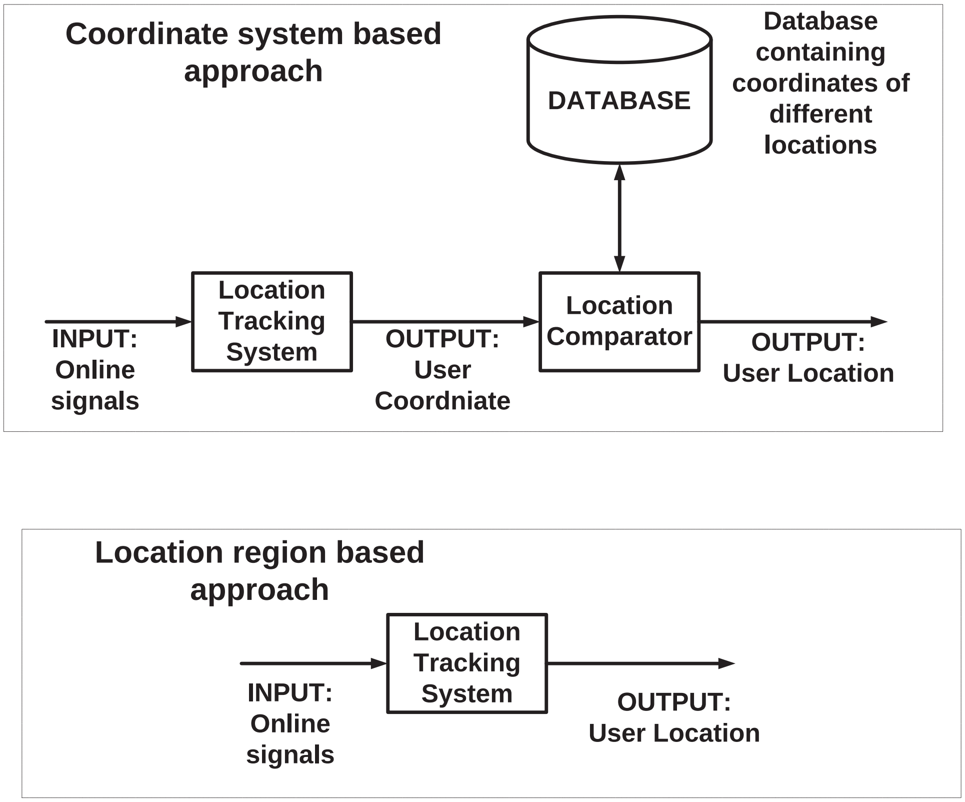


Figure 2.7: Comparison of coordinate system based approach and location region based approach.

2.4 Selection of Location Sensing Technique for our Tracking System

The first step in designing a location tracking system is deciding the location sensing technique that should be adopted. This decision is reached by considering the limitations imposed by the chosen medium (RF, IR, ultrasonic) and the capabilities of the transmitters/receivers of the medium. We will now investigate the suitability of the different location sensing techniques (*triangulation*, *proximity*, and *scene analysis*) for our WLAN-based tracking system.

2.4.1 Suitability of ***Triangulation*** technique

Triangulation techniques essentially use of the geometric properties of triangles to compute object locations. Triangulation can be done either *lateration* or through *angulation*. Triangulation can work only if we are able to precisely take distance measurements (for *lateration*) or angle measurements (for *angulation*).

Angulation technique requires specialized antennas at the receivers. Thus, this technique cannot be used in a WLAN based tracking system, unless significant hardware changes are made to the mobile devices and the WLAN infrastructure. Therefore triangulation using AOA technique is ruled out for a WLAN-based system.

The other option is to use the lateration technique and measure distance between the WLAN transmitters and receivers. For lateration, we have the option of using *signal attenuation* or *time of arrival* techniques.

Signal attenuation technique requires the use of a function that can correlate attenuation and distance. In free space, this relationship is very simple and can be mathematically written as:

$$P_{loss} = \left(\frac{4\pi d}{\lambda}\right)^2 \tag{2.1}$$

where λ is the wavelength and d is the distance.

But for an indoor environment, the path loss is more complex due to the multipath effect. Figure 2.8 illustrates the *multipath effect*, in which a signal's strength and timing are altered as it bounces off physical structures.

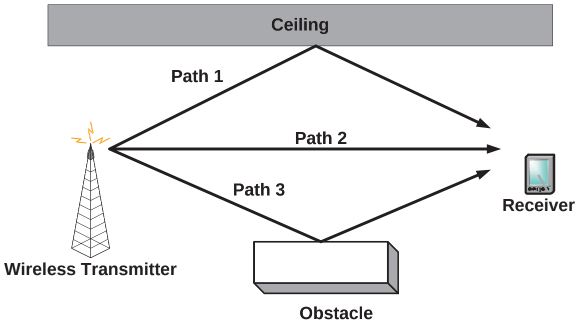


Figure 2.8: Multipath effect.Transmitted signals are reflected or diffracted by structures, changing the signals' timing, strength, and quality.

The multipath effect is more pronounced in an indoor environment than in an outdoor environment. Although researchers have carried out experiments and proposed propagation models for indoor environments [24], these models are still not accurate enough for location tracking purposes. Therefore we had to rule out the use of signal attenuation technique for our tracking system.

Lateration using *time of arrival* is also not feasible for WLAN-based systems because of the lack of precise clocks at the receivers and transmitters. Radio waves travel at very high speeds ($\approx 3 \times 10^8 m/s$). Due to the very short distance between the WLAN transmitters and receivers ($< 50m$), it is impossible to know the time difference between transmission and reception of a WLAN signal.

2.4.2 Suitability of *Proximity* technique

The proximity technique works by determining when a mobile device is “near” a known location. Proximity techniques are simple to implement but the accuracy is limited by how we actually define “near”. It can be applied to WLAN-based systems by monitoring the access points and detecting when a mobile device is in its range. However, since the access points have large coverage areas, it would provide very rough estimates of location. Therefore we had to rule out this location technique.

2.4.3 Suitability of *Scene Analysis* technique

The *Scene Analysis* techniques make use of “feature matching” to predict the user’s location. Features are extracted from different places over the desired area where tracking needs to be done. During the tracking, the pre-stored features are compared with the new features and the user’s location is predicted.

The *scene analysis* sensing technique is ideally suited for a WLAN based tracking system. In a wireless environment, signal strength can be used as a feature. It is easy to extract this feature using the existing WLAN enabled mobile devices.

This does not require making any hardware modifications to the mobile devices. Therefore, we have adopted the *scene analysis* location sensing technique for our tracking system.

2.5 Our WLAN-based Tracking System

Our WLAN tracking systems operates in two phases:

- The *offline* training phase
- The *online* tracking phase

In the offline phase there is no tracking of the mobile device. The offline phase is used to train the location tracking system. In the online phase, the mobile devices can be tracked in real time.

Our WLAN-based tracking system comprises of 3 separate modules:

1. The Signal Sampling Module
2. The Signal Preprocessing Module
3. The Location Tracking Module

The *signal sampling module* and the *signal preprocessing module* are used in the offline training phase, while the *location tracking module* is used only during online tracking phase. Figure 2.9 shows the functioning of the signal sampling module and the signal preprocessing module during the offline training phase. Figure 2.10

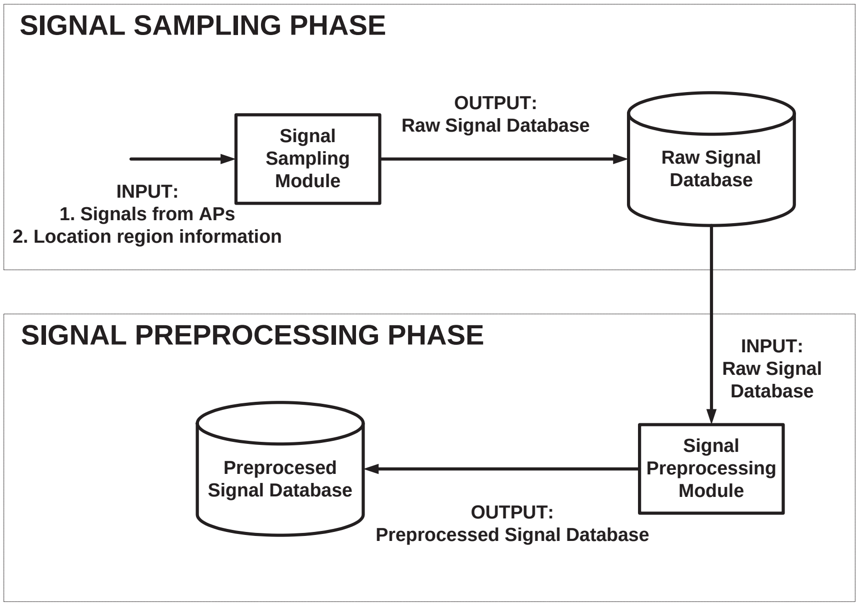


Figure 2.9: Block diagram for the offline training phase modules

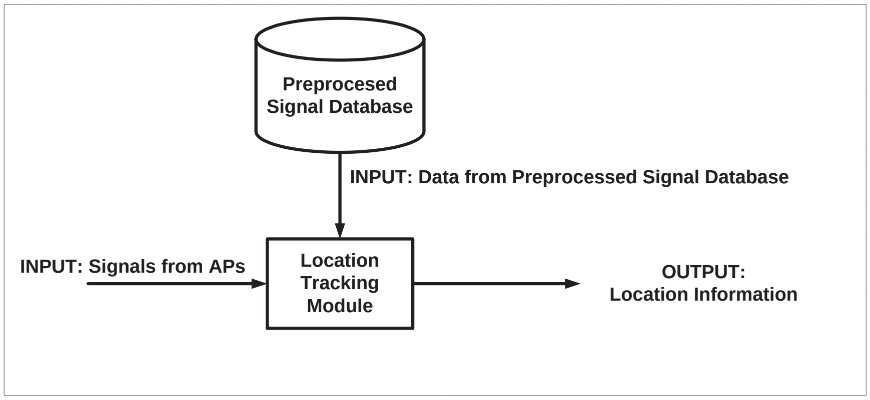


Figure 2.10: Block diagram for the location tracking module

shows the functioning of the location tracking module during the online tracking phase.

As shown in figure 2.9, the offline training phase comprises signal sampling and then its preprocessing. In the signal sampling phase, we first identify the region over which we want to perform location tracking. This entire region is termed as the *area of interest*. The only constraint when deciding on the the *area of interest* is that it should have WLAN coverage. Using a mobile device which has the signal sampling module installed, signals are sampled at different physical locations within the area of interest. These raw signal samples are then stored in a database.

After the signals have been collected, they are pre-processed to extract useful information. Pre-processing entails performing some mathematical operations on the raw data. The exact mathematical operation depends on the tracking algorithm that is to be used during the online phase.

Once the offline phase is complete, the system is ready for location tracking. First the signals being received by a mobile device are recorded. These signals, which we term as the *online signals*, are then compared with the preprocessed signals in the database. Various algorithms can be used to determine the best match between the online and the offline signals. Once a best match is found, the corresponding location region for the offline signal is chosen as the estimated location region.

2.6 Implementation of the Signal Sampling and the Signal Preprocessing Modules

The signal sampling module needs to be able to record the WLAN signals being received by a mobile device. Every WLAN enabled mobile device needs a wireless Network Interface Card (NIC) to be able to access the WLAN. Although the NICs can record the WLAN signals, this information is made available only to the operating system. This is because an operating system segregates the available system memory into *kernel space* and *user space*. The kernel space is reserved for running the kernel and device drivers. Any user application cannot access the kernel space directly. Therefore the information available to the device drivers of the NICs cannot be directly seen by an application in the user space. Therefore it is necessary to design a new software routine to capture the signal strength information from the NIC.

2.6.1 Selection of NIC for development work

NIC for WLAN mobile devices can be of the following types:

1. **PCMCIA card:** These types of cards are the most commonly used. PCMCIA stands for *Personal Computer Memory Card International Association*. PCMCIA is an organization consisting of some 500 companies that has developed a standard for small, credit card-sized devices, called PC Cards. These NICs are detachable. Figure 2.11 shows commercially available PCMCIA

cards from three different vendors.

2. **Wireless USB adapters:** These types of wireless NICs require that the mobile device have a USB slot. These are detachable. Figure 2.12 shows commercially available wireless USB adapters from two different vendors.

3. **In-built card:** These NICs are integrated into the mobile device and are non-detachable.

Of these, the PCMCIA cards are the most widely used. Almost all the commercially available PCMCIA NICs are compatible with the Microsoft Windows operating system. We therefore decided to develop our routine for a Microsoft Windows OS based mobile device and a PCMCIA NIC.

Figure 2.11: PCMCIA network interface cards.

2.6.2 Implementation of the *signal capturing routine*

The *signal capturing routine* was implemented as a DLL (dynamic link library) so that it could be loaded at runtime. This would allow it to be used with other

Figure 2.12: Wireless USB Adapters.

modules as well.

The *signal capturing routine* makes use of the NDIS User Mode I/O Protocol (NDISUIO). NDISUIO provides an interface between a user-mode application and NDIS using the *DeviceIoControl* API. NDISUIO is a connection-less, NDIS 5.1 compliant protocol driver and it allows applications running in user-mode to establish connections to the NICs.

NDIS is short for *Network Driver Interface Specification*. The primary purpose of NDIS is to define a standard API for NICs. The details of a NIC's hardware implementation is wrapped by a *Media Access Controller* (MAC) device driver in such a way that all NIC's for the same media (e.g., Ethernet) can be accessed using a common programming interface. NDIS also provides a library of functions (sometimes called a *wrapper*) that can be used by MAC drivers as well as higher level protocol drivers (such as TCP/IP). The wrapper functions serve to make development of both MAC and protocol drivers easier as well as to hide (to some extent) platform dependencies. Early versions of NDIS were jointly developed by

Microsoft and the 3Com Corporation. Current NDIS versions used by Windows For Workgroups (WFW), Windows 9X Windows NT, Windows 2000, Windows XP and Windows Server 2003 are Microsoft proprietary specifications.

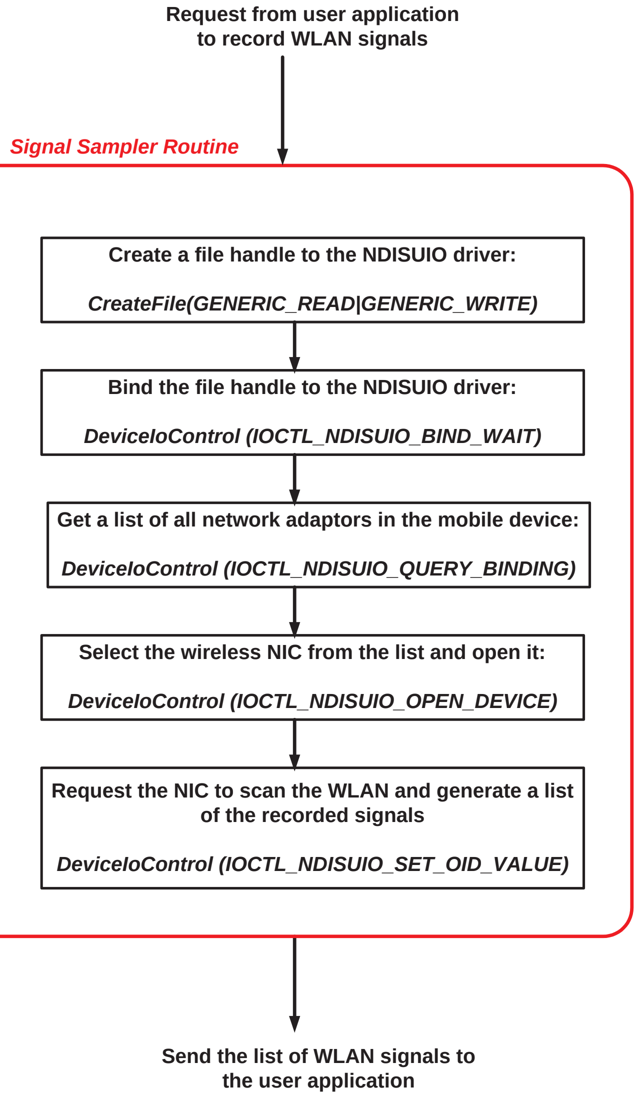


Figure 2.13: Flowchart of the *signal capturing routine*

Figure 2.13 gives the flowchart of the *signal capturing routine* showing how the signal strength information is captured from the NIC using the NDISUIO protocol. The italicized words in the flowchart refer to the NDISUIO APIs along with the most relevant parameter for the API.

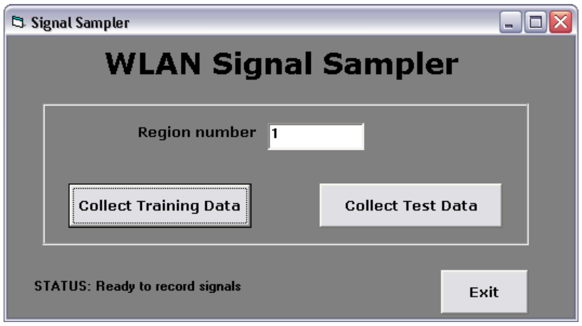
When the *signal capturing routine* is called by any application, it calls the the *CreateFile* function. The *CreateFile* function can be used to create or open a file, directory, physical disk, or communications resource. This function returns a handle that can be used to access the object. In our case, we request the *CreateFile* function to open the NDISUIO driver by listing the first parameter in the function call as “\\\\.*Ndisuio*”. Once a handle is created, it is bound using the *DeviceIoControl* API. This function is called with one of the parameters specified as *IOCTL_NDISUIO_BIND_WAIT*.

If the bind is successful, then the NIC is requested for a list of all the network adapters that the NDISUIO can bind to. This includes ethernet adapters as well as wireless network adapters. The function returns a list of devices. The ethernet interface is always listed first followed by the wireless interface. The wireless interface is selected and opened using the *DeviceIoControl* API (with the parameter set as *IOCTL_NDISUIO_OPEN_DEVICE*).

Once the wireless interface (the NIC) is open, we can send different requests to it. For example, we can request it to tell the MAC address of the associated access point (AP). In figure 2.13, we have requested the NIC to scan the WLAN and find all the available APs. Once the NIC has completed scanning, it returns the MAC address of the various APs and their corresponding signal strengths in decibel units.

2.6.3 The Signal Sampling Module

The *signal sampling* module was developed as a stand-alone application using Visual Basic 6 for the front-end graphical user interface. It used dll developed in C++ for storing the raw data into files. It also made use of the *signal capturing routine* we had described in the previous section. Figure 2.14 shows a screen-shot of the *signal sampling* module.



Region number 1

STATUS: Ready to record signals

Figure 2.14: Signal sampler application.

2.6.4 The Signal Preprocessing Module

The task of the *signal preprocessing module* is to use the raw data collected by signal sampling module and extract useful features from it. The features that need to be extracted from the raw signals could vary depending on the the location tracking algorithm that is used for location detection. In section 2.8 we will discuss in detail all the algorithms that were used by the *signal preprocessing* module.

The *signal preprocessing* module was developed as a stand-alone application. Visual Basic 6 was used for the development of the front-end graphical user interface. The algorithms for preprocessing the raw signals were implemented as a dll (dynamic link library) using C++. Figure 2.15 shows a screenshot of the Signal Preprocessing Application.

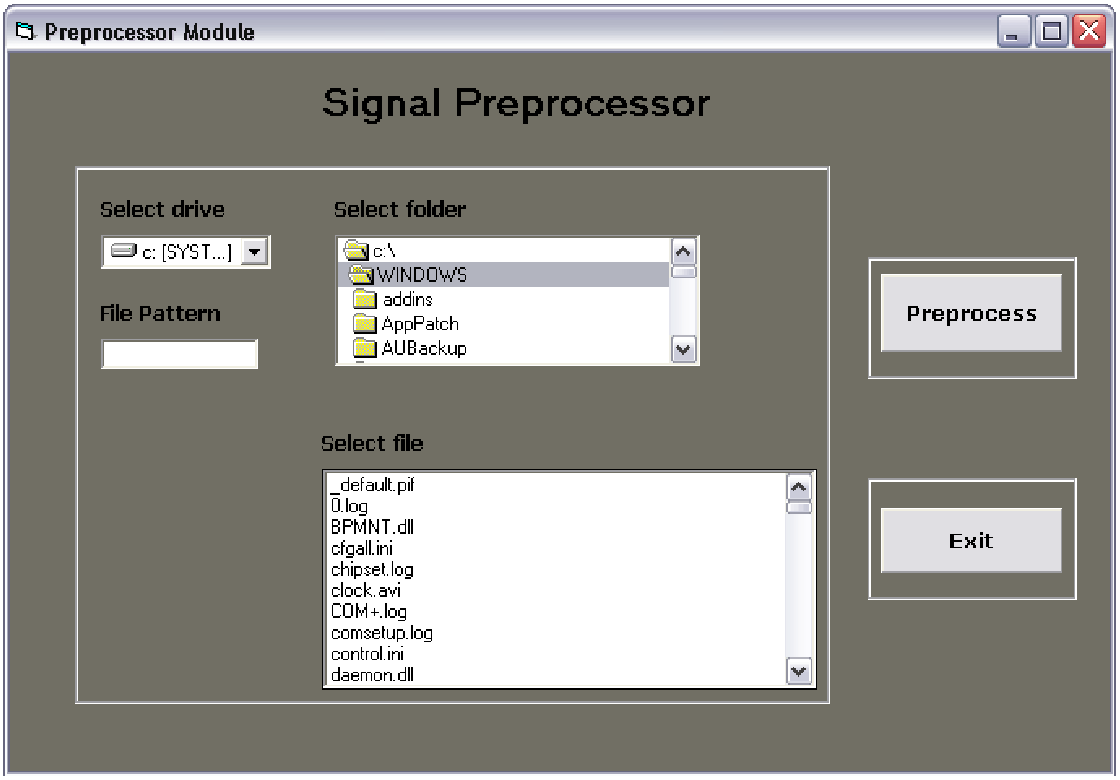


Figure 2.15: GUI for the *signal preprocessing* module

2.7 Implementation of the Location Tracking Module

The design of the location tracking module requires consideration of two issues:

1. What location tracking algorithm should be used?

2. Where will the location tracking module reside, i.e., on the mobile device or on some central server?

The actual location tracking algorithms will be discussed subsequently in section 2.8. In this section we will deal with the second question, which is essentially about the architectural issues.

The location tracking intelligence can be provided either to the mobile device, or to a centrally located device. Table 2.2 compares the advantages and disadvantages for each of these.

Based on these considerations, we implemented a *mobile device assisted, centrally based* architecture for the tracking system.

Centrally based tracking means that the *location tracking module* resides on a central server. *Mobile device assisted* means that the mobile device assists the centrally *located location tracking* module by scanning the WLAN and preparing a *signal report*. The signal report contains a list of the signal strength received from different APs. This signal report is then sent to the central server.

2.7.1 System Architecture

Figure 2.16 illustrates the concept of a *mobile device assisted, centrally based* tracking architecture. The key components are:

1. **Central Location server:** The *location tracking module* resides on this server, allowing it to detect the location of a mobile device. The central server may receive a request for tracking a mobile device by any of the following:

TRACKING DONE BY	ADVANTAGES	DISADVANTAGES
Mobile Device	- Easier enforcement of privacy. - Easy to implement, because signal capturing and location tracking done by the same device.	- Difficult to update tracking system. All mobile users have to install software updates. - Mobile device needs to spend its resources when doing the computations for the tracking algorithm. - If user goes to a new WLAN, he will have to download new data (preprocessed signal database) for the tracking to work properly.
Central Server	- Easy updates of tracking system, transparent to the mobile users. - A powerful server can be employed to do tracking for multiple mobile devices. - Each WLAN can have its own *Location server*, allowing users to migrate seamlessly from one WLAN to another.	- Enforcement of privacy is not simple - Implementation is more complex, since the server has to be robust enough to handle multiple requests. - Need to decide whether the signal capturing for online tracking will be done by the access points, or by the mobile device.

Table 2.2: Comparison of *mobile device based* and *central server based* location tracking architectures.

- The mobile device itself.

- Some other mobile device or an external application.

Appendix D discusses in detail the various scenarios.

2. **Mobile device:** Within a WLAN there can be many mobile devices. These devices will be contacted by the central location server whenever their location needs to be tracked. Appendix D suggests steps that can be taken to protect user privacy.

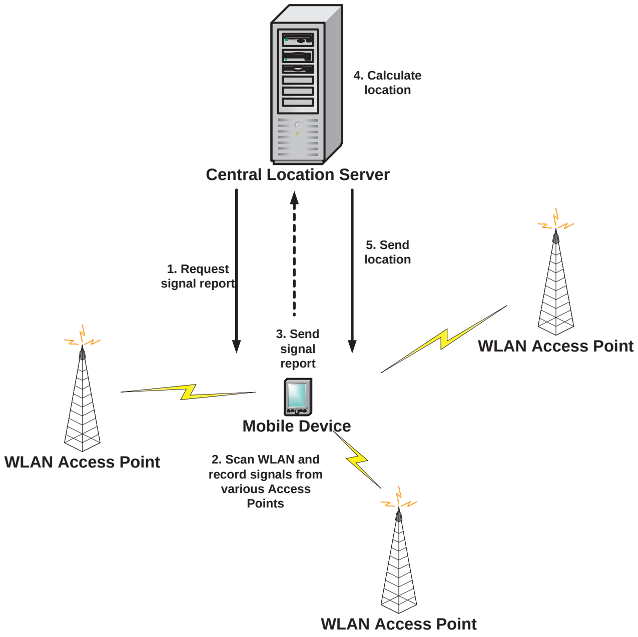


Figure 2.16: *Mobile device assisted, centrally based* tracking architecture: Mobile device assists by scanning for signals, while the central server does the actual tracking

The main advantages of this architecture are:

1. **Minimal disruption of WLAN traffic:** A mobile device assisted system will lead to lesser traffic disruption. This is because within a WLAN there

are many APs and not all of them will be within range of the mobile device that we want to track. By forcing all the APs to scan for a single mobile device instead of letting them continue with their normal activity, we will be disrupting traffic for all the other mobile devices. Contrast this with the case where the mobile device scans the WLAN instead. Here, the disruption is limited only to the mobile device that is being tracked. The traffic of the other mobile devices will not be affected.

2. **Support for easy updates to the location tracking system:** It may so happen that the tracking algorithm needs to be modified in order to improve the accuracy, or that the preprocessed signal database needs to be updated because of changes in the building layout. In a centrally based system, all these changes can be done at the central server and the individual mobile users will not have to be bothered.

2.7.2 Implementation of the Location Server

We implemented a simple location server for our WLAN-based tracking system. The location server is entrusted with two tasks:

1. Handling requests for tracking the location of a mobile device.
2. Doing the actual tracking once it has received the required signal strength information from a mobile device.

The location tracking intelligence was provided by the *location tracking module*. For handling requests from mobile devices, we implemented a *communication module*. Figure 2.17 shows the relation between the *communication module* and the *location tracking module*.

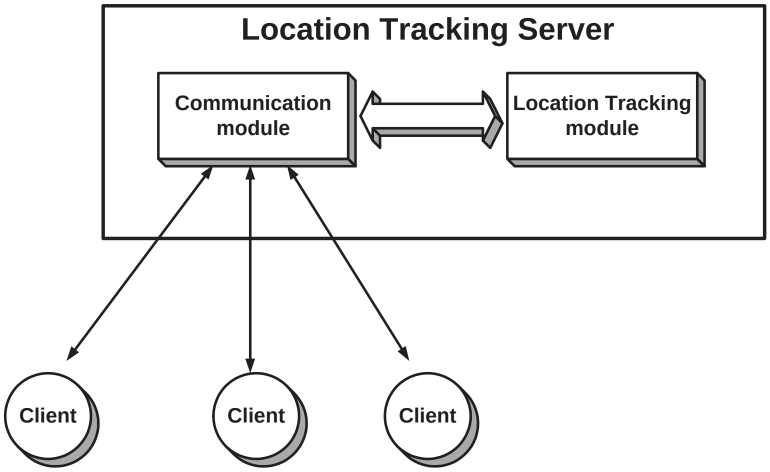


Figure 2.17: Interaction of the *communication module* and the *location tracking module*: The *communication module* handles location tracking requests from clients. It obtains the necessary online signals from them and passes the information to the *location tracking module*, which then computes the location and returns it to the *communication module*.

For our WLAN-based tracking system, the location server was implemented on a Windows XP machine. To handle location tracking requests, the *communication module* opens a TCP/IP listening socket and waits for requests. Once a request is received, it opens another socket to handle the request. This enables it to continue listening for new requests while serving the current request. The *communication module* then contacts the mobile device with a request to scan the WLAN. Once it receives a list of signals, it passes the information to the *location tracking module* which tracks the location and returns the information to the *communication* *module*. This location is then sent to the mobile device. Figure 2.18 shows the flowchart of the location server. The location server handles all requests for location tracking. We did not implement any procedures for enforcing the privacy of mobile devices as this would shift our focus away from our primary task. However, we present the guidelines for privacy enforcement in appendix D.

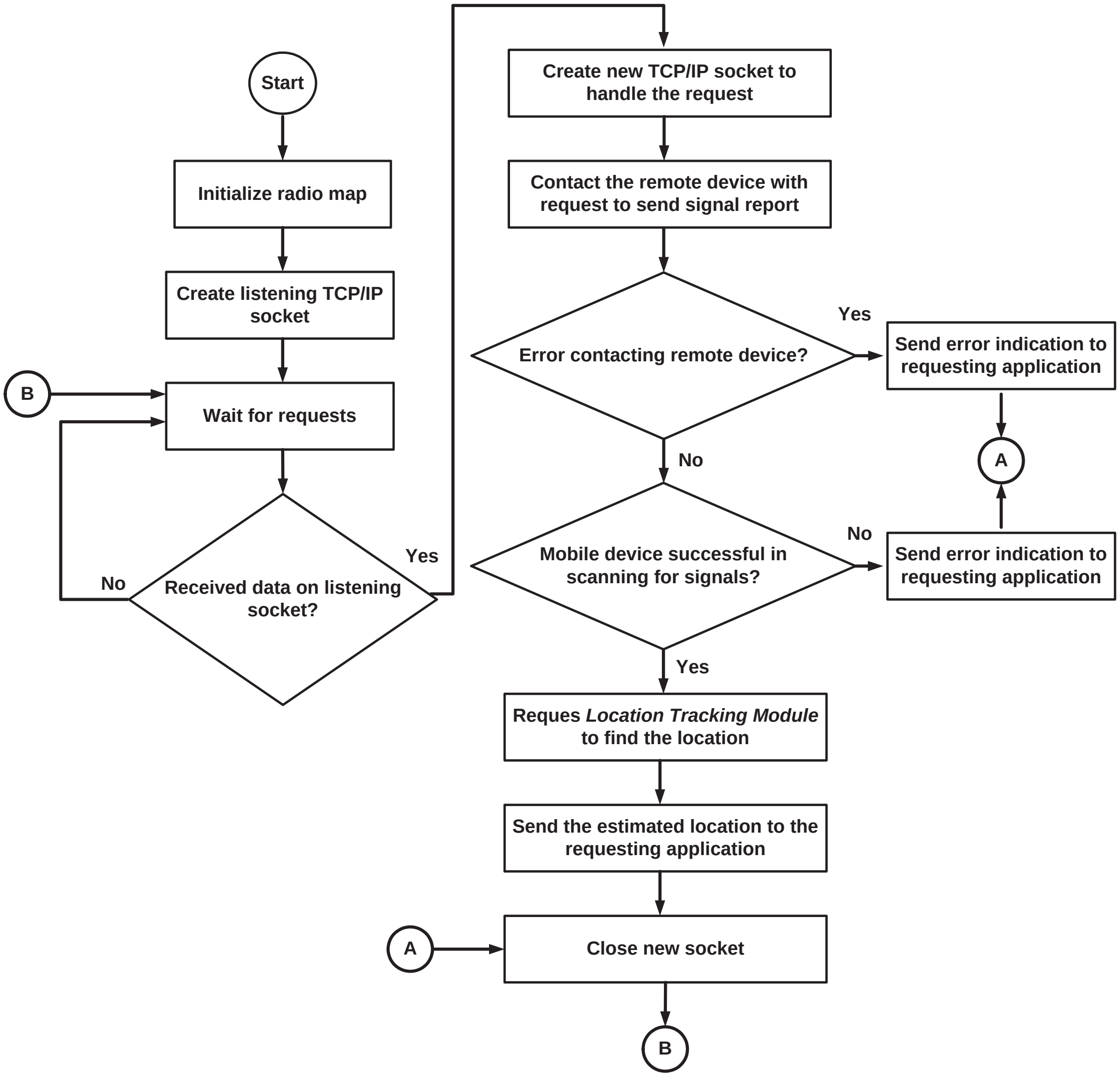


Figure 2.18: Flowchart of the location server

2.8 Algorithms for the *Signal Preprocessing* and *Location Tracking* Modules

2.8.1 A pattern recognition approach for location tracking

The main task of a pattern recognition system is to provide decisions based on given samples, by which incoming patterns can be classified. As an example, consider two pattern classes, C_1 and C_2, as illustrated in figure 2.19. Each class contains a number of *sample patterns*, each of which is a vector $\boldsymbol{x} = (x_1, x_2)$ in the $x_1 - x_2$ plane. In figure 2.19, samples belonging to pattern class C_1 are denoted by unfilled circles, while those belonging to pattern class C_2 are denoted by filled cirles. For any incoming pattern $\boldsymbol{x}$, known *a priori* to belong to C_1 or C_2, we want to find the best way for making the estimate.

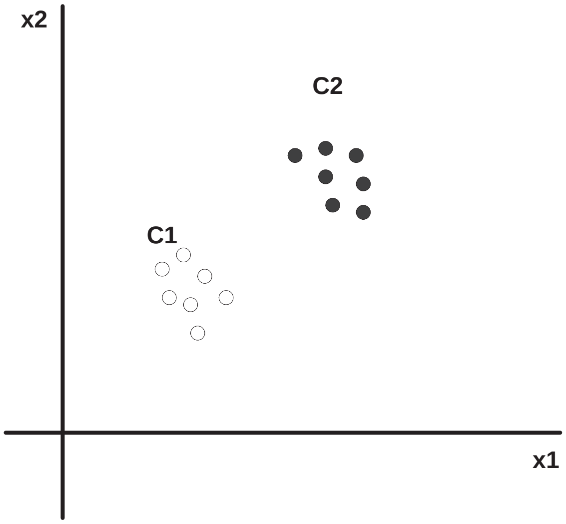


Figure 2.19: The pattern classification problem.

In our WLAN tracking system, the pattern classes are the different *location* *regions*. Each *location region* contains a number of sample patterns, in the form of the the preprocessed signals. Each sample pattern can be represented as a vector $\boldsymbol{S} = (s_1, s_2, \ldots, s_n)$, where n is the number of APs in the WLAN, and s_k is the signal from the k^{th} access point. Thus the problem is essentially one of matching an incoming signal pattern to its correct class (i.e., location region).

We used different algorithms to determine the best pattern recognition approach for our system. We will now present these algorithms.

We evaluated the tracking results using different variations of the *template matching* and *statistical pattern matching* techniques. These are well known classifications of pattern recognition techniques [25] [26].

We used the following methods for our location tracking:

1. Minimum Distance method
2. k-Minimum Distance method
3. Cluster-Center Distance method
4. Bayes' criteria using individual locations
5. Bayes' criteria using clustered locations

The first three algorithms are based on the concept of finding the minimum distance between an incoming sample pattern and the pre-stored sample patterns. They differ in the definition of distance. The first two algorithms use the point-to-point distance measurements. The *Cluster-Center Distance* algorithm uses the point-to-set distance measurements.

We have also used the Bayes' method for our location tracking system. We would like to point out that although the Bayes' method has been used by previous WLAN tracking systems [21] [22], it has *not* been used in the context of *location region* based location tracking. Therefore we wanted to see how the probabilistic performed in comparison with the other methods. We use the Bayes' method in two different ways, as described in subsequent sections.

All the algorithms that we will describe use the following notations:

L_i **:** The i^{th} *location region*.

l_{ij} **:** The j^{th} individual location in the i^{th} *location region*, L_i. Each *location region* contains a number of individual locations.

$\boldsymbol{S}$ **:** The signal vector recorded during the offline training period. $\boldsymbol{S} = (s_1, s_2, \ldots, s_n)$, where n is the number of APs in the WLAN, and s_k is the signal from the k^{th} access point.

$\boldsymbol{Z}$ **:** The signal vector recorded during the online training period. $\boldsymbol{Z} = (z_1, z_2, \ldots, z_n)$, where n is the number of APs in the WLAN, and z_k is the signal from the k^{th} access point.

AP_k **:** The k^{th} access point.

2.8.2 Minimum Distance Method

The concept of this algorithm is illustrated in figure 2.20. The figure uses a 2-dimensional space in order to better illustrate the concept. L1 and L2 are two location regions, containing a number sample patterns. The problem is to classify the incoming sample pattern $\boldsymbol{Z}$.

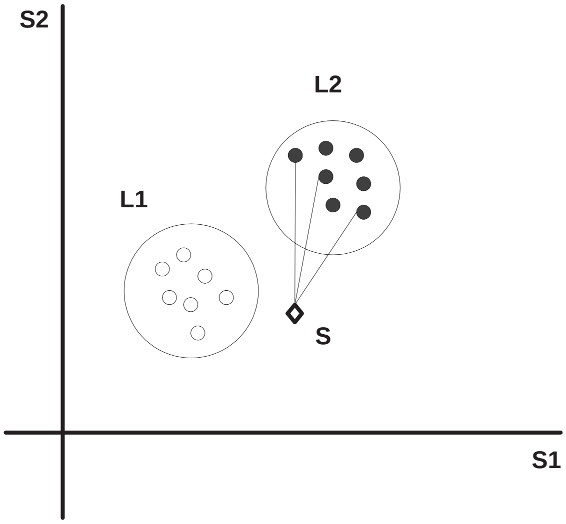


Figure 2.20: Point-to-Point Distance measurement

The minimum distance method works by finding the closest sample to $\boldsymbol{Z}$, say $\boldsymbol{S}$. Then $\boldsymbol{Z}$ is classified into the class that contains the sample pattern $\boldsymbol{S}$. We now present the algorithm for this method.

Preprocessing Algorithm

Step 1. Open the database containing the raw signals.

Step 2. Until EOF, do

1. Average signals for each individual location.

2. Store each signal pattern in a table as $(L_i, l_{ij}, AP_k, \boldsymbol{S})$

Step 3. Divide the large table into smaller tables by sorting data according to the AP_k. Each smaller table now contains entries of the form (L_i, l_{ij}, S_k)

Step 4. Save the tables to database file.

Tracking Algorithm

Step 1. Obtain incoming pattern $\boldsymbol{Z} = (z_1, z_2, \ldots, z_n)$.

Step 2. Open database containing pre-processed signal samples. Until EOF reached, do

1. Read next signal pattern $\boldsymbol{S}$, and its location region L.
2. Measure the distance, d, between $\boldsymbol{Z}$ and $\boldsymbol{S}$, using $d = \left[\sum_{i=1}^{n} (z_i - s_i)^2\right]^{1/2}$.
3. If $d < d_{min}$, set

$$d_{min} = d$$

$$\boldsymbol{S}_{nearest} = \boldsymbol{S}$$

$$L_{estimated} = L$$

Step 4. Return $L_{estimated}$ as the estimated location region.

Figure 2.21 gives the flowchart for this algorithm.

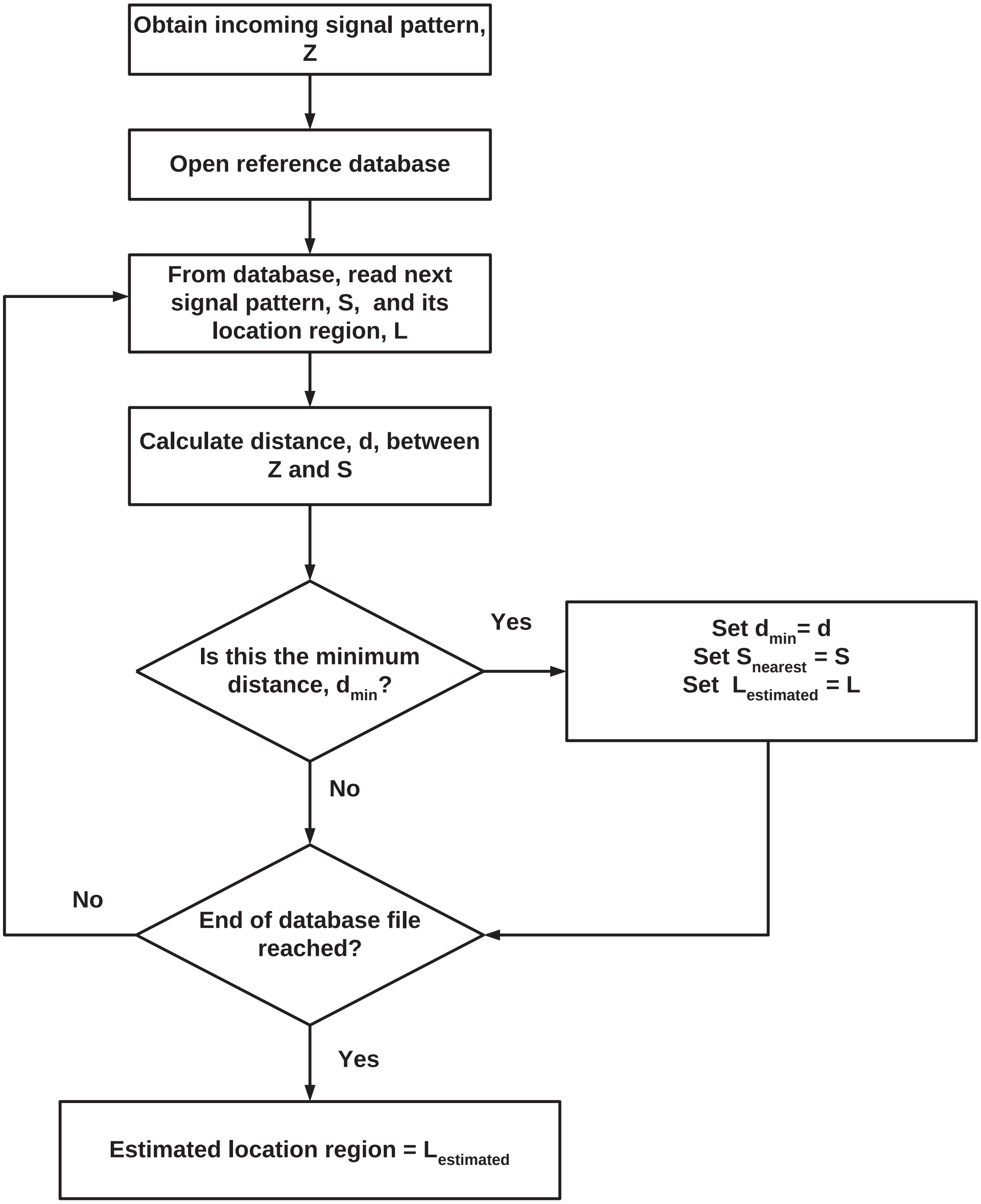


Figure 2.21: Flowchart for Minimum Distance Method based tracking

2.8.3 k-Minimum Distance Method

This algorithm is similar to the previous algorithm. However, in this case we measure the k nearest sample patterns to the incoming sample pattern. Then, the estimated location is the one containing the maximum number of these k nearest sample patterns.

We just present the tracking algorithm below since the offline pre-processing algorithm is the same as the one in the Minimum Distance method.

Step 1. Obtain incoming pattern $\boldsymbol{Z} = (z_1, z_2, \ldots, z_n)$.

Step 2. Open database containing pre-processed signal samples. Until EOF reached, do

```
1. Read next signal pattern $\boldsymbol{S}$, and its location region L.
2. Measure the distance, $d$, between $\boldsymbol{Z}$ and $\boldsymbol{S}$, using
   $d = \left[ \sum_{i=1}^{n} (z_i - s_i)^2 \right]^{1/2}$.
3. For $j = 1$ to $k$
          If $d < d_{min}(j)$, then
             For $r = j + 1$ to $k$
                $d_{min}(r) = d_{min}(r-1)$ and $\boldsymbol{S}_{nearest}(r) = \boldsymbol{S}_{nearest}(r-1)$
                $d_{min}(j) = d$ and $\boldsymbol{S}_{nearest}(j) = S$
             Exit For
   Exit For
```

Step 3. In the set $\boldsymbol{S}_{nearest}(k)$, find the location region, $L_{maxcount}$, that has maximum frequency of occurrence. Set $L_{estimated} = L_{maxcount}$.

Step 4. Return $L_{estimated}$ as the estimated location region.

Figure 2.22 gives the flowchart for this algorithm.

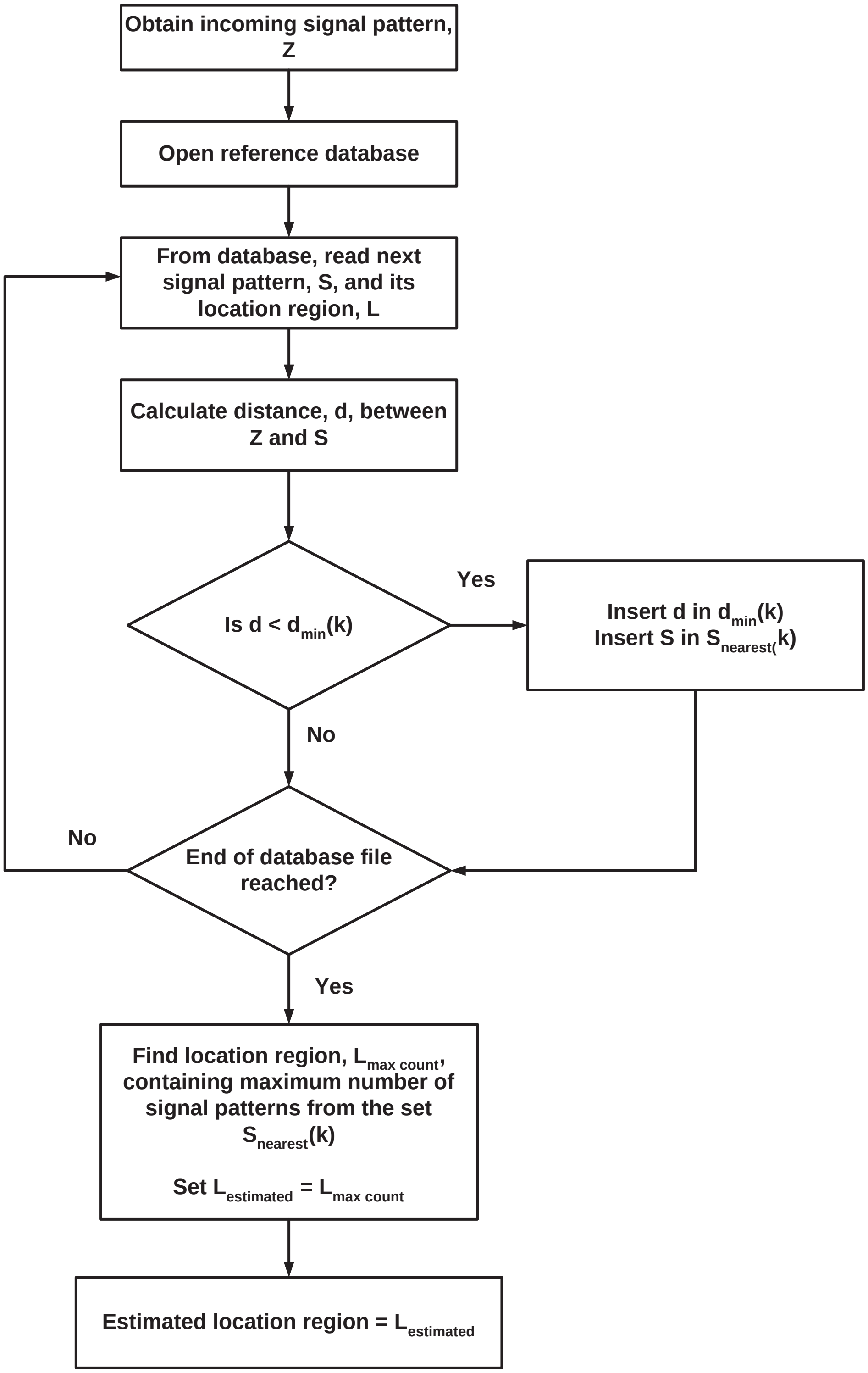
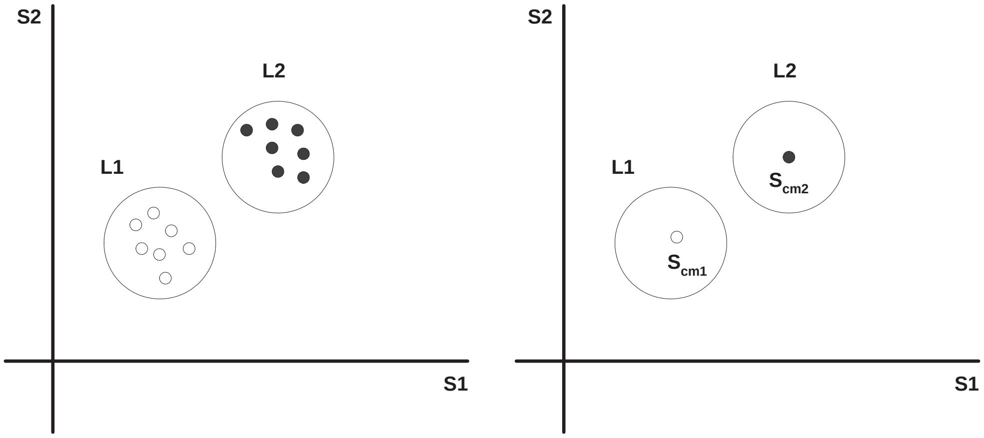


Figure 2.22: Flowchart for k-Minimum Distance Method based tracking

2.8.4 Cluster-Center Distance Method

This method is uses *point-to-set* measurements instead of *point-to-point* measurements as in the *minimum distance* and the *k-minimum distance methods*. This is illustrated in figure 2.23.

First, the raw signals are preprocessed to obtain the center of mass of each cluster. Then, during the online tracking stage, the distance of the incoming pattern and each cluster center is measured. The estimated location is the one having its center nearest to the incoming pattern. The advantage of this method is that it allows us to maintain a very small database of signal patterns. This translates to faster tracking compared to the previous two tracking methods. We present the algorithm below. The flowchart is given in figure 2.24.



(a) Before preprocessing

(b) After preprocessing. Each cluster represented by its center of mass

Figure 2.23: Illustration of the preprocessing for Distance-to-Cluster-Center method

Preprocessing Algorithm

Step 1. Open the database containing the raw signals.

Step 2. Until EOF, do

1. Average signals for each individual location.
2. Store each signal pattern in a table as $(L_i, l_{ij}, AP_k, \boldsymbol{S})$

Step 3. Create a table for each *location region.* Each entry in the table contains entries of the form $(\boldsymbol{S}, AP_j, l_{ij})$

Step 4. Calculate cluster center of each *location region*, L_i using

$\boldsymbol{S_{cm_i}} = \frac{\sum_{j=1}^{n} \boldsymbol{S_j}}{n}$

where $n =$ total number of individual locations in a given *location region*

Step 5. Store entries in tables sorted by *location region*, L_i. Each entry in a table is of the form $\boldsymbol{S_{cm_i}}$

Tracking Algorithm

Step 1. Obtain incoming pattern $\boldsymbol{Z} = (z_1, z_2, \ldots, z_n)$.

Step 2. Open database containing preprocessed signal samples. For each *location region*, L,do

1. Measure the distance, d, between the $\boldsymbol{Z}$ and $\boldsymbol{S_{cm}}$ using
   $d = \left[\sum_{i=1}^{n} (z_i - s_{cmi})^2 \right]^{1/2}.$

2. If $d < d_{min}$, set

$$d_{min} = d$$

$$L_{estimated} = L$$

Step 3. Return $L_{estimated}$ as the estimated location region.

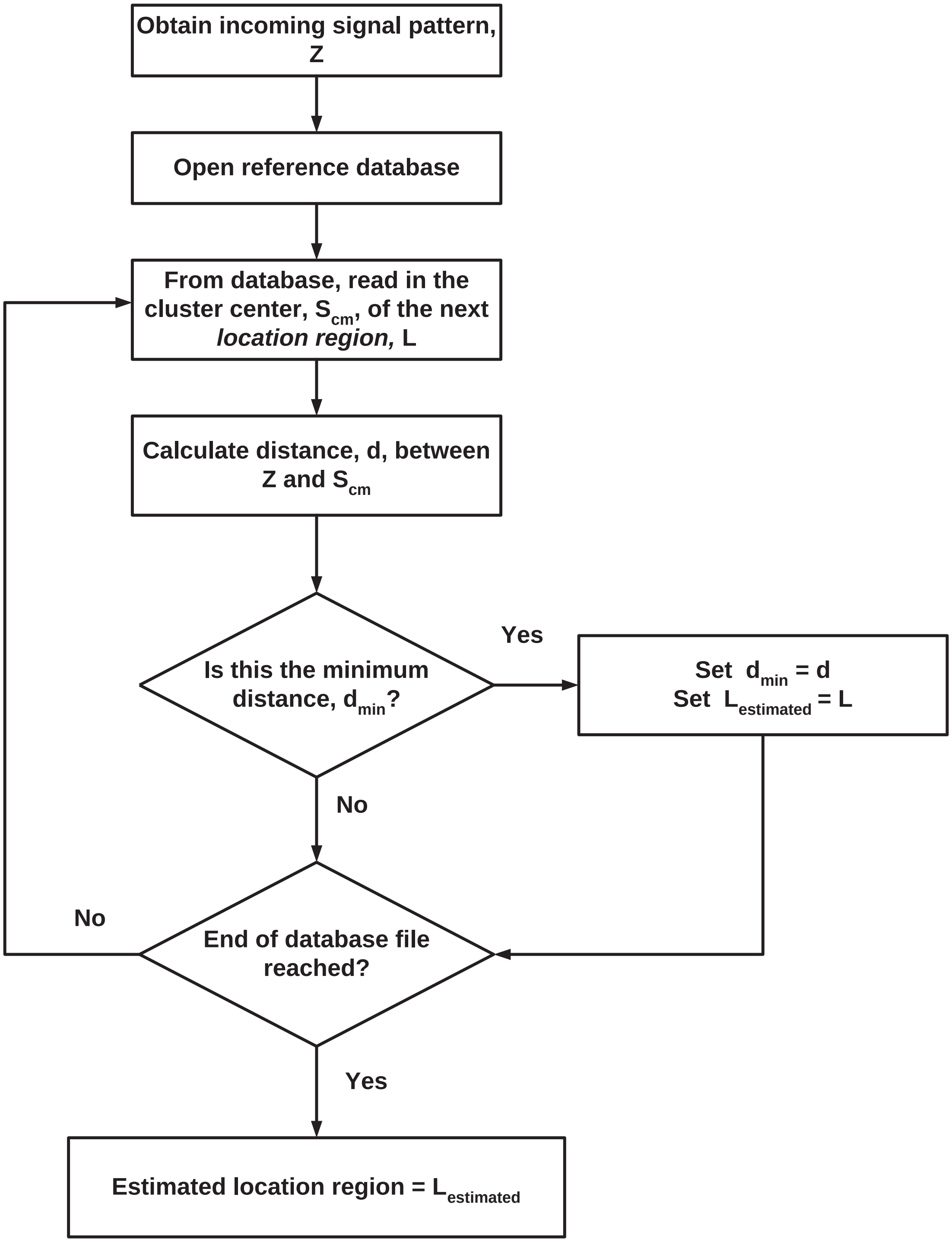


Figure 2.24: Flowchart for the *Cluster-Center distance* method based tracking

2.8.5 Bayes' criteria using individual locations

The Bayes' formula is a statistical approach to the problem of pattern classification. In pattern recognition we are interested in obtaining a rule based on a set of features that is useful for classification of objects into one of the g possible classes. Such a rule is called and allocation or decision rule. In statistical pattern recognition, the probability distribution of the features and Baye's formula are used to obtain a decision rule. We will briefly explain the Bayes' theorem and how it can be applied in the context of our WLAN tracking system.

If there is an object that is to be classified on the basis of a feature vector $\boldsymbol{X}$, into one of g possible classes $(C_1, C_2, \ldots, C_g)$, then the probability that the object is classified into class i when $\boldsymbol{X}$ is observed can be described by $P(C_i|\boldsymbol{X})$. This is the exactly the problem statement in the case of WLAN based tracking, where we want to assign an incoming signal pattern as belonging to one of the *location regions.*

Bayes' theorem gives to the solution to this problem. It states:

$$P(C_i|\boldsymbol{X}) = \frac{P(C_i|\boldsymbol{X})P(C_i)}{P(\boldsymbol{X})} \tag{2.2}$$

In the context of the WLAN tracking, $\boldsymbol{X}$ is nothing but the incoming signal pattern. $(C_1, C_2, \ldots, C_g)$ are the *location regions.* Using Bayes' theorem, we can classify the estimated location region to be the one which has the highest probability, $P(C_i|\boldsymbol{X})$.

The probabilities on the right hand side of (2.2), are pre-determined in the following way:

$P\boldsymbol{X}|(C_i)$ **:** This is nothing but the probability of finding a given signal pattern at location region C_i. This can be done by sampling signals in different physical locations and noting the probabilities of occurrence of different signals.

$P(C_i)$ **:** This is nothing but the probabilities of occurrence of different regions. If it is equally likely for a user to be in any of the regions, then

$$P(C_1) = P(C_2) = \ldots = P(C_g)$$

Otherwise higher probabilities can be assigned to regions which are more popular (e.g., canteen area can be given a higher priority than corridors).

$P(\boldsymbol{X})$ **:** This is constant for all locations

Thus Bayes' theorem can be easily used for location tracking purposes.

In our case, we used Bayes' criteria in two alternative ways. The first was to use the Bayes' method and try to estimate the physical location of the user. Once a location is determined, we then find the *location region* containing this physical location. We call this approach as *Bayes' criteria using individual locations.*

In the second method, which we have termed as *Bayes' criteria using clustered locations*, probabilities are stored for each location region, instead of individual physical locations. This approach results in faster tracking during the real-time tracking phase.

We will first present the algorithm for the *Bayes' criteria using individual locations.*

Preprocessing Algorithm

Step 1. Open the database containing the raw signals.

Step 2. For each individual location, l_{ij}

For all the signals recorded from each access point,

Calculate the *a priori* probabilities using $p(s_{ij_k}) = n/N$

where,

n = number of occurrences of signal s during the sampling process N = number of samples taken at that location

$p(s_{ij_k})$ = probability that signal from the k^{th} AP has a value of s

Step 3. Create tables sorted by AP. Each entry in the table contains entries of the form $(p(\boldsymbol{S}_{ij}), l_{ij}, L_i)$

where, $p(\boldsymbol{S}_{ij})$ = set of probabilities of all the signal samples that were recorded at physical location l_{ij}

Tracking Algorithm

Step 1. Obtain incoming pattern $\boldsymbol{Z} = (z_1, z_2, \ldots, z_n)$.

Step 2. Open database file. Until EOF reached, do

1. For next physical location, l_{ij}, calculate probability that the incoming signal belongs to this location, using Bayes' theorem

   $P(l_{ij}|\boldsymbol{Z}) = P(\boldsymbol{Z}|l_{ij})$, where

   $P(\boldsymbol{Z}|l_{ij}) = P(z_1|l_{ij})P(z_2|l_{ij})\ldots P(z_n|l_{ij})$

2. If $P(l_{ij}|\boldsymbol{Z}) > P_{max}$, set

   $P_{max} = P(l_{ij}|\boldsymbol{Z})$

   $l_{estimated} = l_{ij}$

Step 3. Find the location region, L, that contains the physical location $l_{estimated}$ and set $L_{estimated} = L$.

Step 4. Return $L_{estimated}$ as the estimated location region.

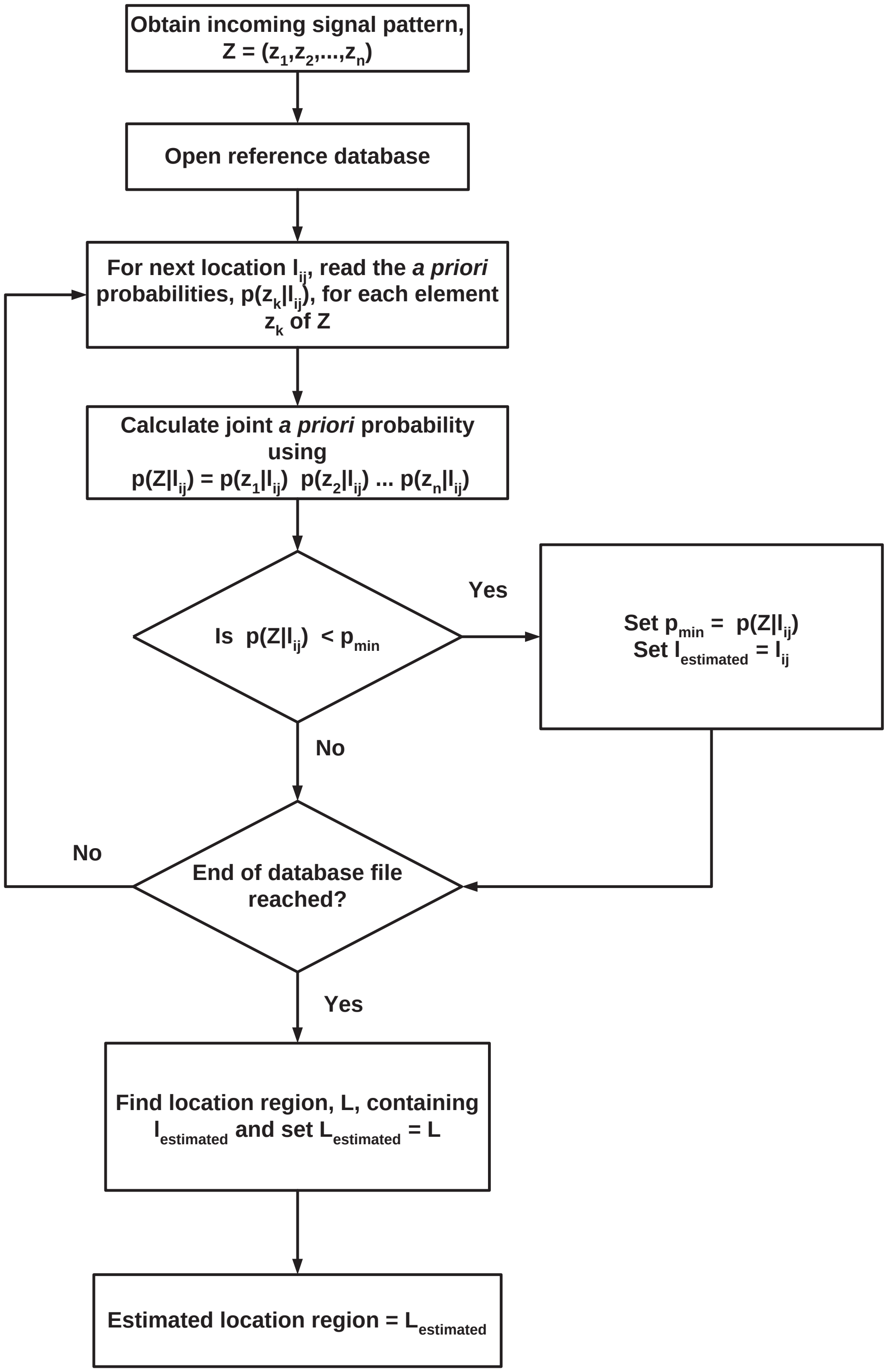


Figure 2.25: Flowchart for the *Bayes' Individual Location* Method

2.8.6 Bayes' criteria using clustered locations

In this method, we preprocess the data differently and store the probabilities for each *location region*, instead of storing for each *individual* physical location. Since in each *location region* we have taken samples at a number of individual locations, this approach allows us to have a smaller database. This, in turn, leads to faster real-time tracking.

We now present the algorithms and flowchart for this method.

Preprocessing Algorithm

Step 1. Open the database containing the raw signals.

Step 2. For each individual location, l_{ij}

For all the signals recorded from each access point,

Calculate the *a priori* probabilities using $p(s_{ij_k}) = n/N$

where,

n = number of occurrences of signal s during the sampling process N = number of samples taken at that location

$p(s_{ij_k})$ = probability that signal from the k^{th} AP has a value of s

Step 3. Create tables sorted by AP. Each entry in the table contains entries of the form $(p(\boldsymbol{S}_{ij}), l_{ij}, L_i)$

where, $p(\boldsymbol{S}_{ij})$ = set of probabilities of all the signal samples that were recorded at physical location l_{ij}

Step 4. For each location region, L_i,

Combine the probabilities of different individual locations, $p(\boldsymbol{S})$, to get the *a priori* probability for the entire *location region*, $p(\boldsymbol{S_L})$

Step 5. Create tables sorted by AP. Each entry in the table contains entries of the form $(p(\boldsymbol{S_L}), l, L)$

Tracking Algorithm

Step 1. Obtain incoming pattern $\boldsymbol{Z} = (z_1, z_2, \ldots, z_n)$.

Step 2. Open database file. Until EOF reached, do

1. For next *location region*, L_i, calculate probability that the incoming signal belongs to this location, using Bayes' theorem

   $P(L_i|\boldsymbol{Z}) = P(\boldsymbol{Z}|L_i)$, where

   $$P(\boldsymbol{Z}|L_i) = P(z_1|L_i)P(z_2|L_i)\ldots P(z_n|L_i)$$

2. If $P(L_i|\boldsymbol{Z}) > P_{max}$, set

   $$P_{max} = P(L_i|\boldsymbol{Z})$$

$$L_{estimated} = L_i$$

Step 3. Return $L_{estimated}$ as the estimated location region.

2.9 Experimental Results

We conducted experiments to find the answers to the following questions:

1. Which tracking method gives the best results?
2. How does the placement of the access points affect the tracking accuracy?

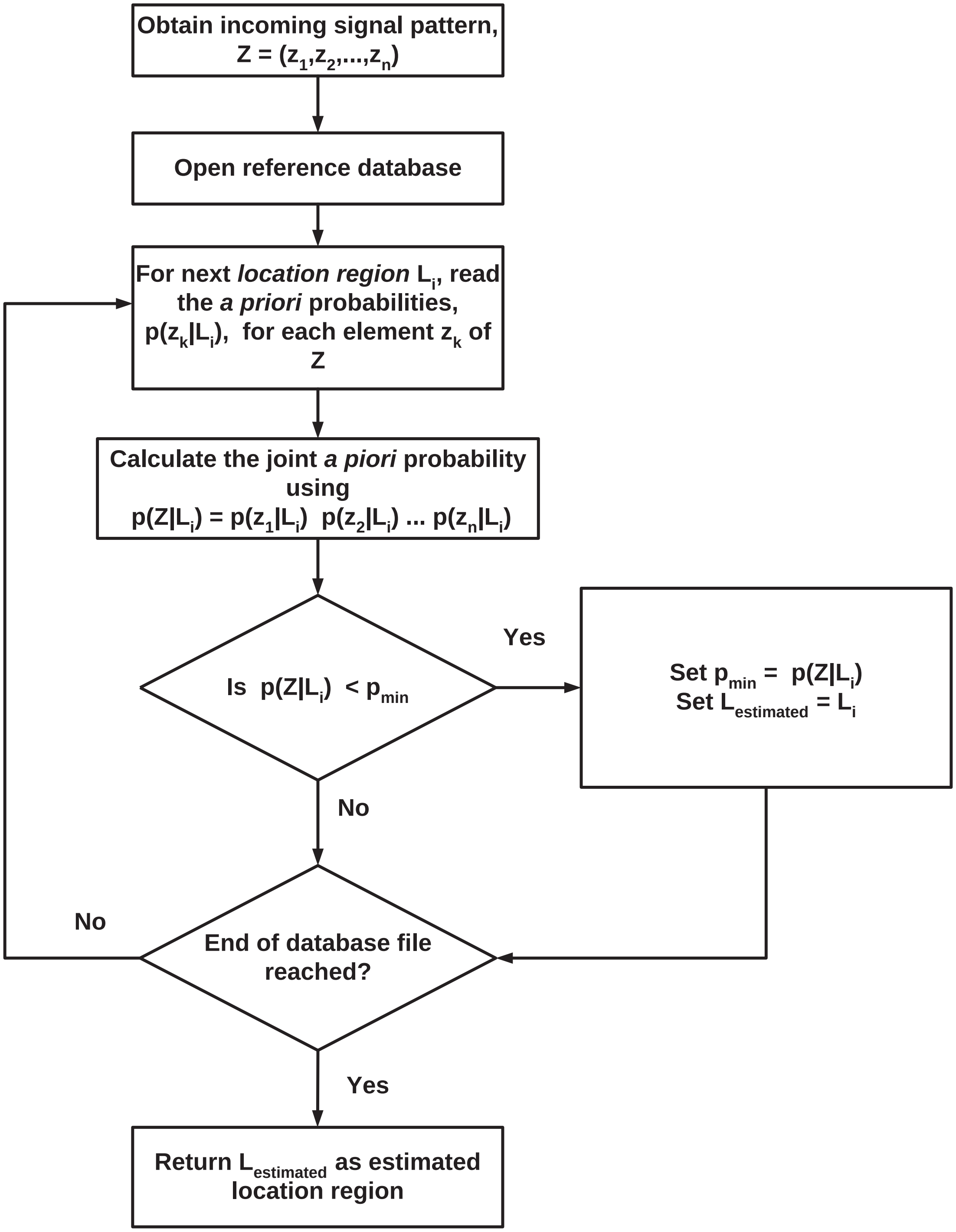


Figure 2.26: Flowchart for the *Bayes' Clustered Region* Method

2.9.1 Experimental Testbed

The experiments were performed in the Mixed Reality Laboratory in the Engineering building. Figure 2.27 shows the floor plan of the room. We used 8 WLAN routers from two vendors (LinkSys and Prolink) for use as the access points. These access points were setup on the walls at different locations as shown in figure 2.27.

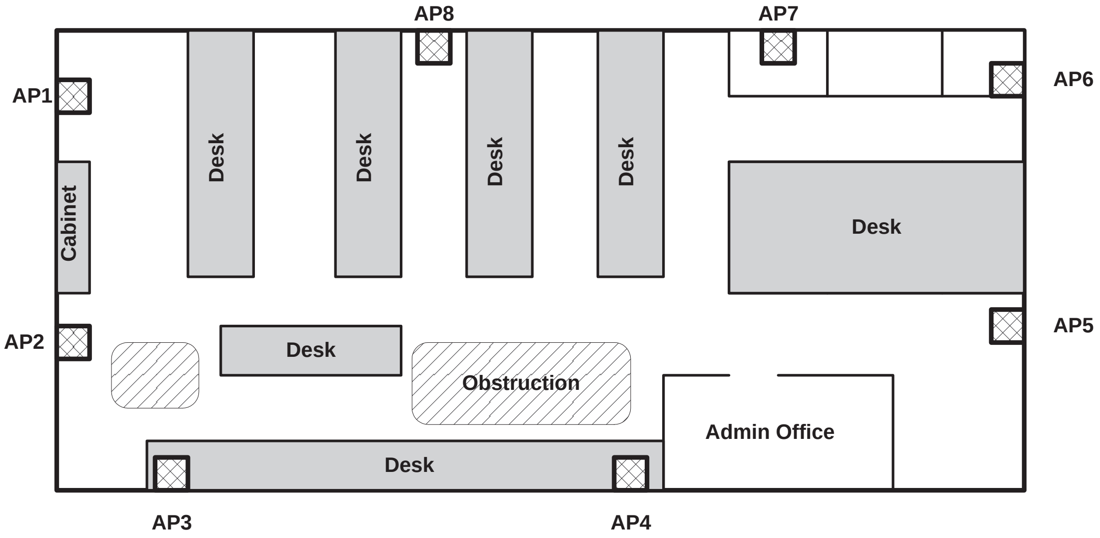


Figure 2.27: Floor plan of the test bed.

2.9.2 Training phase

The training phase comprises the signal sampling phase and the signal preprocessing phase. For the signal sampling, we used a Sony VAIO (PCG-161P) as the mobile device during the sampling process. It was equipped with an Orinico Gold WLAN card. Figure 2.28 shows a picture of the Orinoco card and the mobile device.

First the area of interest was subdivided into 3 regions and represented in

Figure 2.28: Hardware used for the signal sampling.

numerical terms as regions 1,2 and 3, as illustrated in figure 2.29.

In each region, signals were sampled at all the likely locations where a user could be present. These locations were spaced at a distance of 1.5m. It should be noted that the clustered region approach allowed for an easier and faster training since we did not have to bother with specifying our exact training location. Had we chosen the coordinate system approach, then we would have to make sure that the physical location matched exactly with the specified coordinates. Figure 2.30 shows the locations at which the signals were sampled.

For each physical location 100 signal samples were taken at the rate of 1 sample/second. In each sample, the signal strength from various access points was recorded. The application recorded the signal strength of the beacon packets that were transmitted by each AP. After the raw signals were collected, they were preprocessed using the algorithms mentioned in sections 2.8.2 to 2.8.6.

Tables 2.3-2.10 show the average values of the signals from various access points.

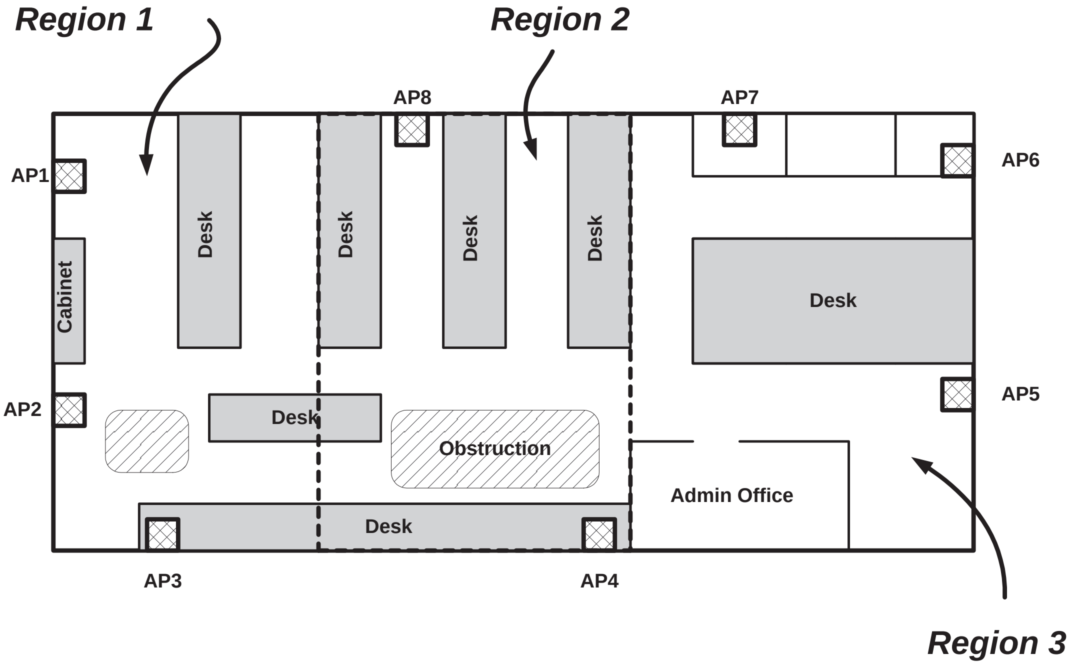


Figure 2.29: Region division of the area of interest.

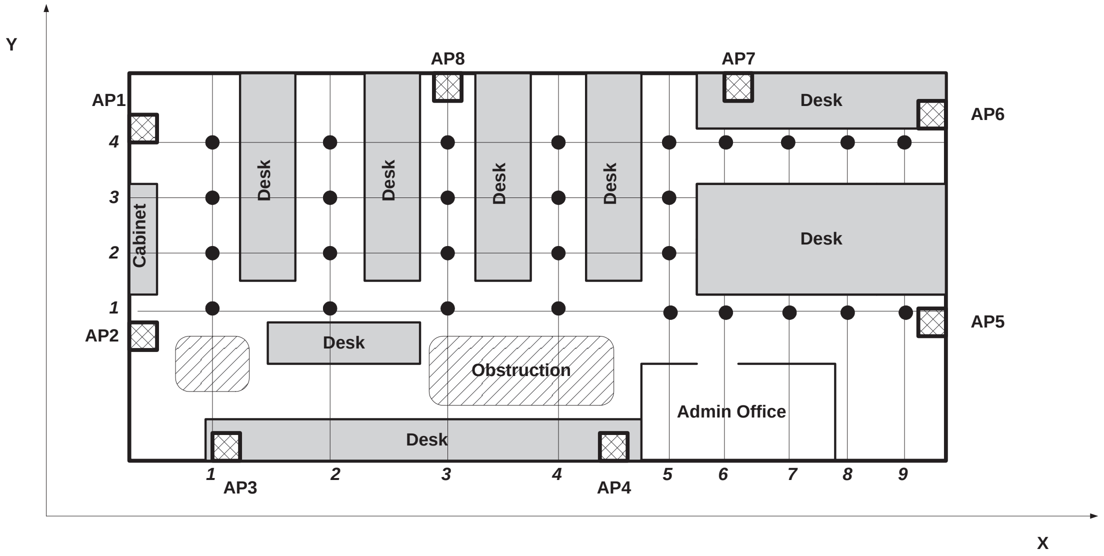


Figure 2.30: Locations at which signals were sampled for the training data. The figure is not drawn to scale. The floor plan has been divided into x, y coordinates for use in tables 2.3 to 2.10.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-44.86
(1,2)	-42.63
(1,3)	-44.62
(1,4)	-42.16
(2,1)	-45.33
(2,2)	-43.47
(2,3)	-48.47
(2,4)	-45.78
(3,1)	-54.03
(3,2)	-52.16
(3,3)	-50.24
(3,4)	-53.03
(4,1)	-50.32
(4,2)	-58.28
(4,3)	-47.55
(4,4)	-57.43
(5,1)	-55.2
(5,2)	-59.36
(5,3)	-60.57
(5,4)	-54.02
(6,1)	-53.83
(6,4)	-58.05
(7,1)	-65.78
(7,4)	-58.21
(8,1)	-57.94
(8,4)	-58.01
(9,1)	-57.33
(9,4)	-62.26

Table 2.3: Average signal strengths in the testbed from AP1. Coordinates refer to locations shown in figure 2.30.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-38.06
(1,2)	-37.32
(1,3)	-39.41
(1,4)	-41.86
(2,1)	-33.22
(2,2)	-40.29
(2,3)	-44.9
(2,4)	-43.29
(3,1)	-42.03
(3,2)	-32.34
(3,3)	-26.13
(3,4)	-46.51
(4,1)	-28.28
(4,2)	-42.67
(4,3)	-13.78
(4,4)	-41.79
(5,1)	-17.58
(5,2)	-20.62
(5,3)	-15.02
(5,4)	-20.79
(6,1)	-22.46
(6,4)	-22.36
(7,1)	-27.73
(7,4)	-24.03
(8,1)	-30.82
(8,4)	-15.4
(9,1)	-28.31
(9,4)	-26.68

Table 2.4: Average signal strengths in the testbed from AP2. Coordinates refer to locations shown in figure 2.30.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-42.84
(1,2)	-48.41
(1,3)	-48.74
(1,4)	-47.93
(2,1)	-42.9
(2,2)	-43.7
(2,3)	-49.14
(2,4)	-39.69
(3,1)	-29.92
(3,2)	-45.01
(3,3)	-44.58
(3,4)	-26.46
(4,1)	-37.74
(4,2)	-44.04
(4,3)	-47.82
(4,4)	-29.43
(5,1)	-42.19
(5,2)	-37.9
(5,3)	-46.13
(5,4)	-24.52
(6,1)	-43.37
(6,4)	-20.72
(7,1)	-19.87
(7,4)	-19.43
(8,1)	-32.75
(8,4)	-25.69
(9,1)	-22.2
(9,4)	-29.79

Table 2.5: Average signal strengths in the testbed from AP3. Coordinates refer to locations shown in figure 2.30.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-10.52
(1,2)	-15.8
(1,3)	-17.94
(1,4)	-12.28
(2,1)	-12.34
(2,2)	-15.49
(2,3)	-11.79
(2,4)	-17.11
(3,1)	-24.08
(3,2)	-13.93
(3,3)	-11.81
(3,4)	-13.87
(4,1)	-15.73
(4,2)	-5.95
(4,3)	-11.59
(4,4)	-11.13
(5,1)	-35.55
(5,2)	-16.24
(5,3)	-13.61
(5,4)	-21.53
(6,1)	-17.79
(6,4)	-19.45
(7,1)	-14.24
(7,4)	-14.69
(8,1)	-7.84
(8,4)	-10.91
(9,1)	-8.56
(9,4)	-7.82

Table 2.6: Average signal strengths in the testbed from AP4. Coordinates refer to locations shown in figure 2.30.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-8.97
(1,2)	-14.28
(1,3)	-11.25
(1,4)	-18.8
(2,1)	-14.76
(2,2)	-11.24
(2,3)	-13.85
(2,4)	-9.74
(3,1)	-20.74
(3,2)	-11.97
(3,3)	-8.28
(3,4)	-18.7
(4,1)	-17.2
(4,2)	-14.06
(4,3)	-13.95
(4,4)	-14.07
(5,1)	-13.21
(5,2)	-22.64
(5,3)	-15.26
(5,4)	-13.78
(6,1)	-15.4
(6,4)	-35.22
(7,1)	-33.28
(7,4)	-29.03
(8,1)	-9
(8,4)	-38.02
(9,1)	-40.65
(9,4)	-42.69

Table 2.7: Average signal strengths in the testbed from AP5. Coordinates refer to locations shown in figure 2.30.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-50.2
(1,2)	-24.62
(1,3)	-19.48
(1,4)	-25.83
(2,1)	-18.7
(2,2)	-29.6
(2,3)	-36.71
(2,4)	-48.18
(3,1)	-15.62
(3,2)	-12.04
(3,3)	-43.62
(3,4)	-32.17
(4,1)	-27.05
(4,2)	-45.71
(4,3)	-45.38
(4,4)	-24.94
(5,1)	-20.69
(5,2)	-43.35
(5,3)	-46.29
(5,4)	-44.25
(6,1)	-35.02
(6,4)	-41.88
(7,1)	-42.9
(7,4)	-36.78
(8,1)	-40.47
(8,4)	-36.58
(9,1)	-41.58
(9,4)	-31.20

Table 2.8: Average signal strengths in the testbed from AP6. Coordinates refer to locations shown in figure 2.30.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-27.15
(1,2)	-17.92
(1,3)	-16.96
(1,4)	-26.1
(2,1)	-33.9
(2,2)	-28.61
(2,3)	-26.39
(2,4)	-17.54
(3,1)	-34.37
(3,2)	-43.57
(3,3)	-47.32
(3,4)	-35.33
(4,1)	-42.33
(4,2)	-42.01
(4,3)	-38.92
(4,4)	-38.44
(5,1)	-35.1
(5,2)	-40.45
(5,3)	-32.76
(5,4)	-34.39
(6,1)	-40.45
(6,4)	-42.77
(7,1)	-43.11
(7,4)	-45.88
(8,1)	-34.38
(8,4)	-28.01
(9,1)	-38.08
(9,4)	-43.21

Table 2.9: Average signal strengths in the testbed from AP7. Coordinates refer to locations shown in figure 2.30.

Location Coordinates (x, y)	**Average Signal Strength (dBm)**
(1,1)	-42.49
(1,2)	-36.7
(1,3)	-40.02
(1,4)	-38.24
(2,1)	-30.67
(2,2)	-38.08
(2,3)	-29.17
(2,4)	-30.66
(3,1)	-34.14
(3,2)	-34.52
(3,3)	-30.9
(3,4)	-22.97
(4,1)	-37.06
(4,2)	-31.95
(4,3)	-31.6
(4,4)	-31.58
(5,1)	-31.9
(5,2)	-39.37
(5,3)	-38.29
(5,4)	-36.58
(6,1)	-41.26
(6,4)	-13.75
(7,1)	-22.64
(7,4)	-15.68
(8,1)	-34.11
(8,4)	-16.77
(9,1)	-31.71
(9,4)	-28.41

Table 2.10: Average signal strengths in the testbed from AP8. Coordinates refer to locations shown in figure 2.30.

2.10 Experimental Results for Location Tracking Algorithms

The test data was collected separately, and the test locations were chosen randomly in each region. In each region we collected at least 50 test samples. We measured the location tracking accuracy for the different location tracking algorithms that were discussed in section 2.8. We have defined location tracking accuracy as,

$$Location\,Tracking\ Accuracy = (Accurate\ Predictions/Total\ Test\ Samples) \times 100$$

In order to find the best performing algorithm we tested the location tracking accuracy of the algorithms for the following 4 cases:

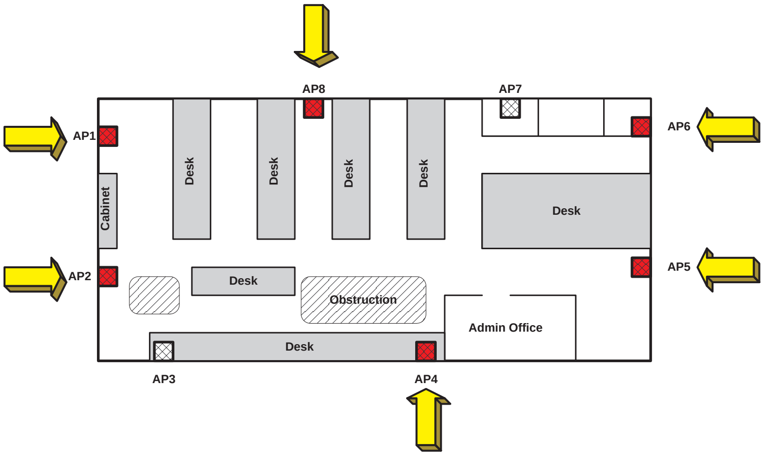
1. Using signals from 8 access points - This involved all the APs in our testbed.
2. Using signals from 6 access points - Figure 2.31(a) shows the access points that were used.
3. Using signals from 4 access points - Figure 2.31(b) shows the access points that were used.
4. Using signals from 2 access points - Figure 2.31(c) shows the access points that were used.

2.10.1 Performance of different tracking methods when number of APs was varied

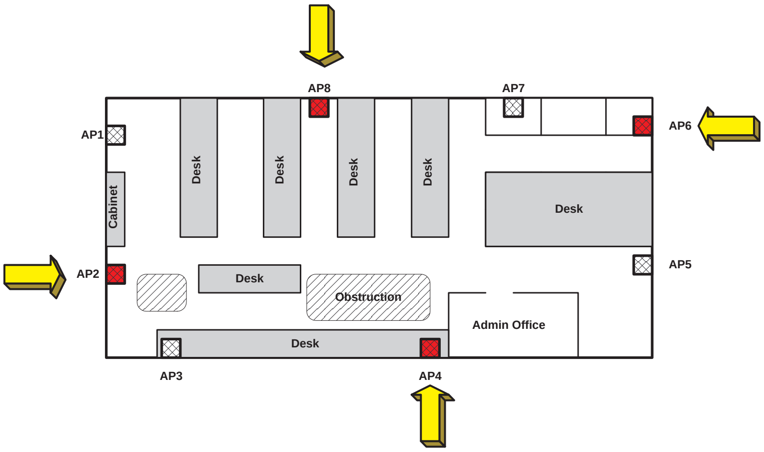
Figure 2.32 shows the location tracking accuracy for different regions when the *Bayes' clustered region* method was used. Figure 2.33 shows the location tracking accuracy for different regions when the *Bayes' individual region* method was used. Figure 2.34 shows the location tracking accuracy for different regions when the *cluster-center distance* method was used. Figure 2.35 shows the location tracking accuracy for different regions when the *minimum distance* method was used. Figure 2.36shows the location tracking accuracy for different regions when the *k-minimum distance* method was used. We used the value of $k = 5$.

2.10.2 Effect of varying the value of k in the *k-Minimum Distance* method

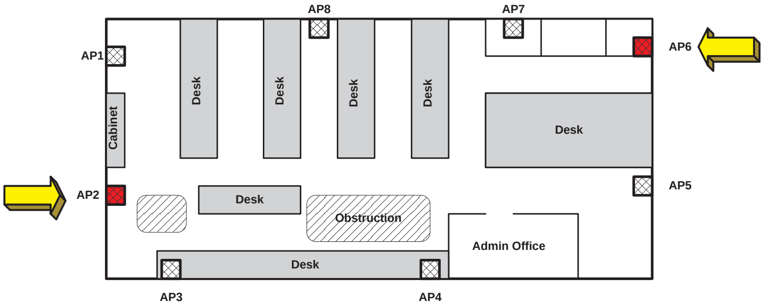
Figure 2.37 shows the effect of varying the value of k. As we use larger values of k, we are in effect measuring the distance to more and more individual locations in each region. This increases the memory requirements of the tracking system and also slows down the real-time tracking speed.



(a) Tracking using 6 APs

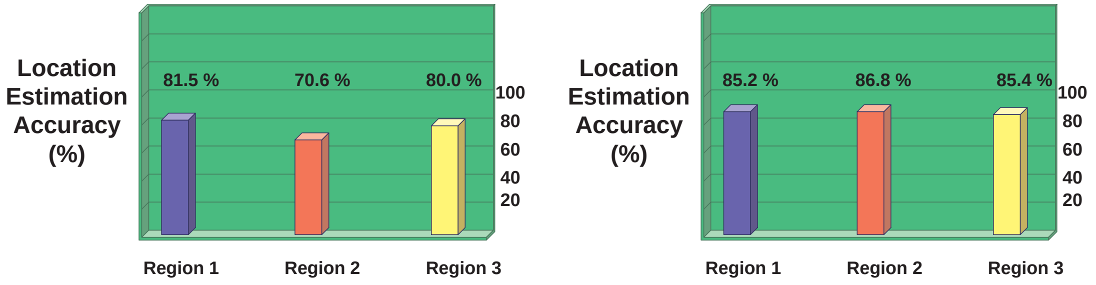


(b) Tracking using 4 APs



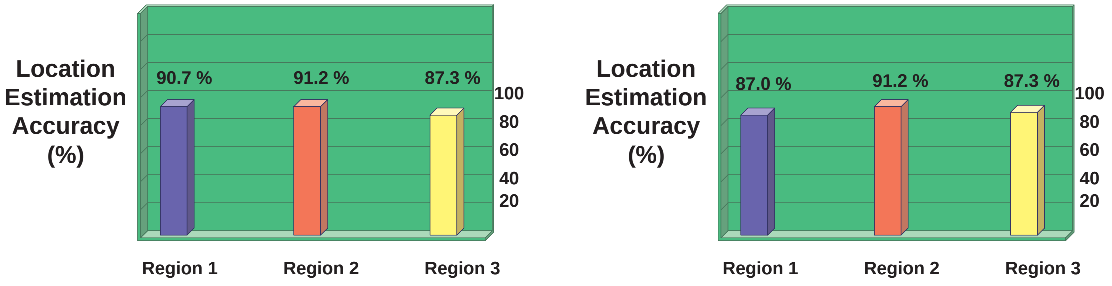
(c) Testing using 2 APs

Figure 2.31: Location of the APs used in various phases of our testing.



Using 2 APs

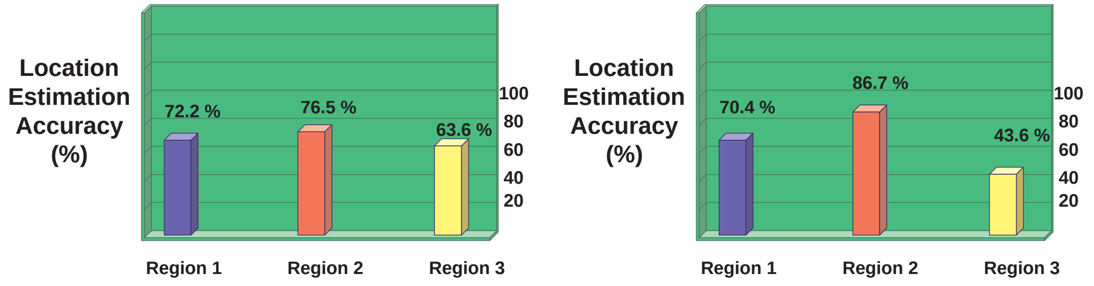
Using 4 APs



Using 6 APs

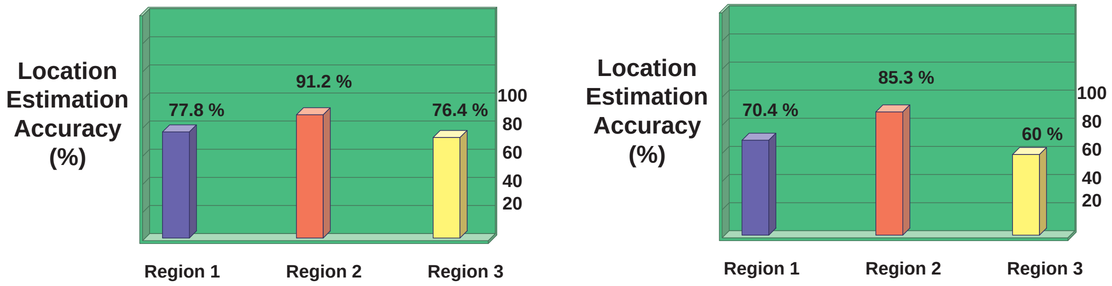
Using 8 APs

Figure 2.32: Tracking accuracy using the *Bayes' Clustered Location* method.



Using 2 APs

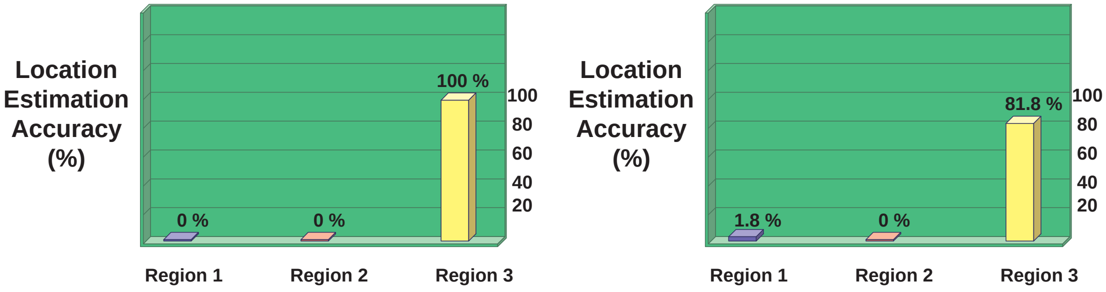
Using 4 APs



Using 6 APs

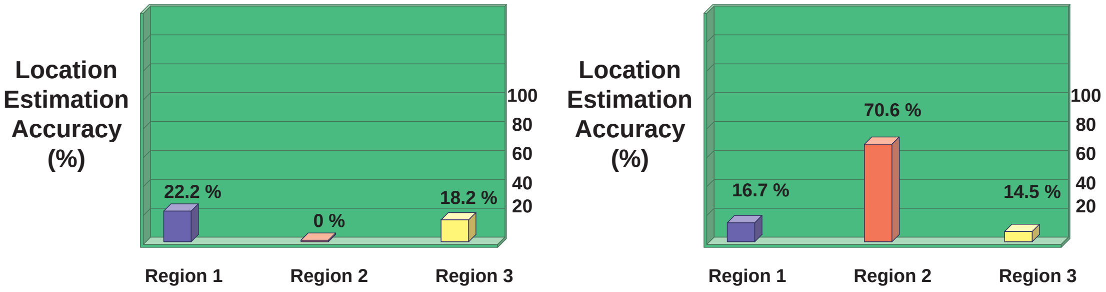
Using 8 APs

Figure 2.33: Tracking accuracy using the *Bayes' Individual Location* method.



Using 2 APs

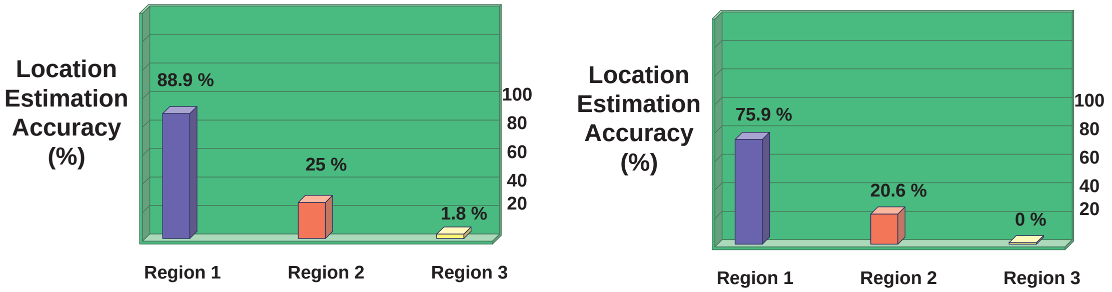
Using 4 APs



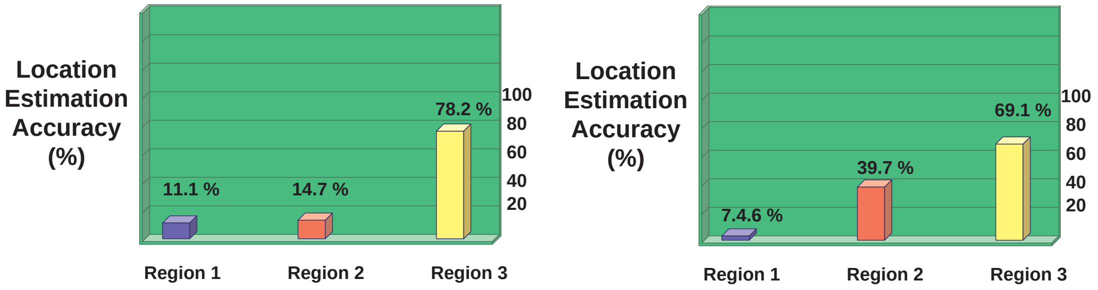
Using 6 APs

Using 8 APs

Figure 2.34: Tracking accuracy using the *Cluster Center Distance* method.



Using 2 APs **Using 4 APs**



Using 6 APs **Using 8 APs**

Figure 2.35: Tracking accuracy using the *Minimum Distance* method.

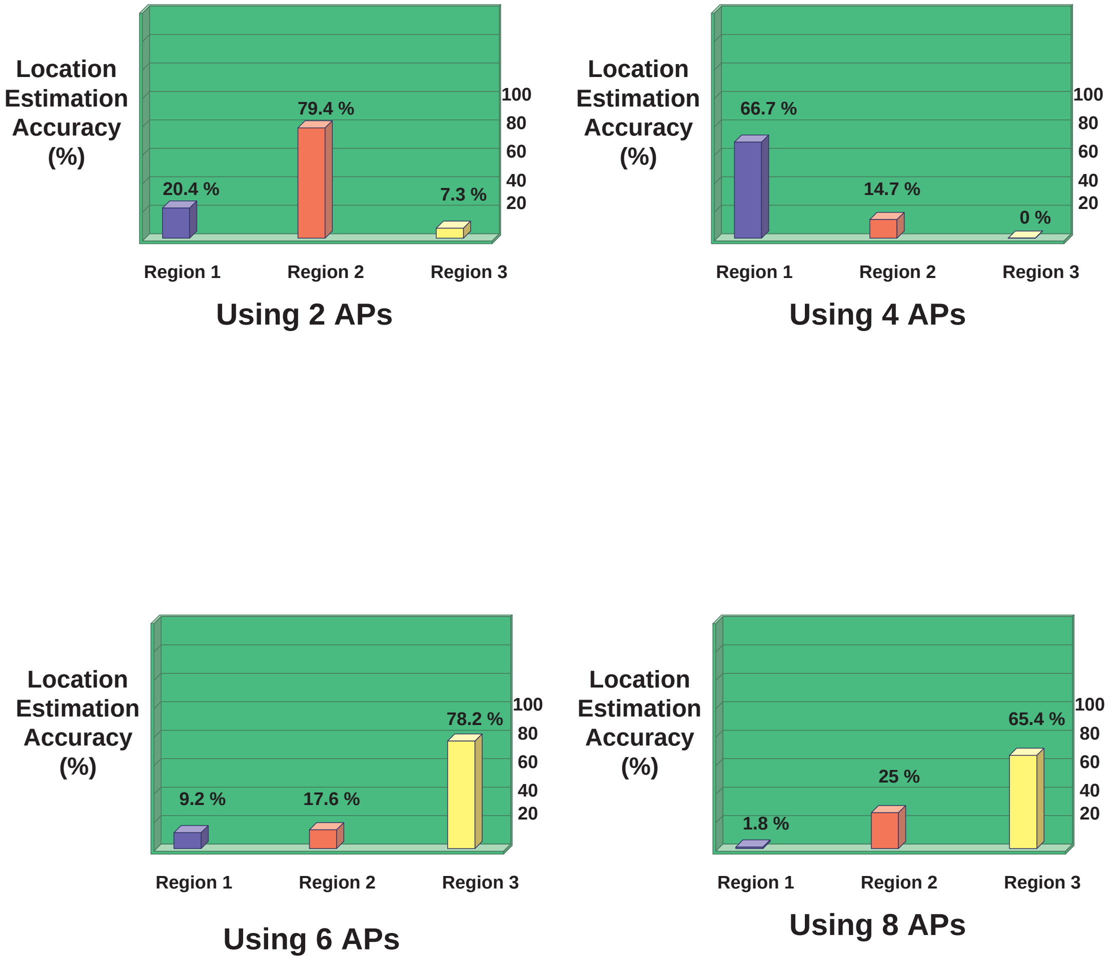


Figure 2.36: Tracking accuracy using the k *Minimum Distance* method. $k = 5$ was used.

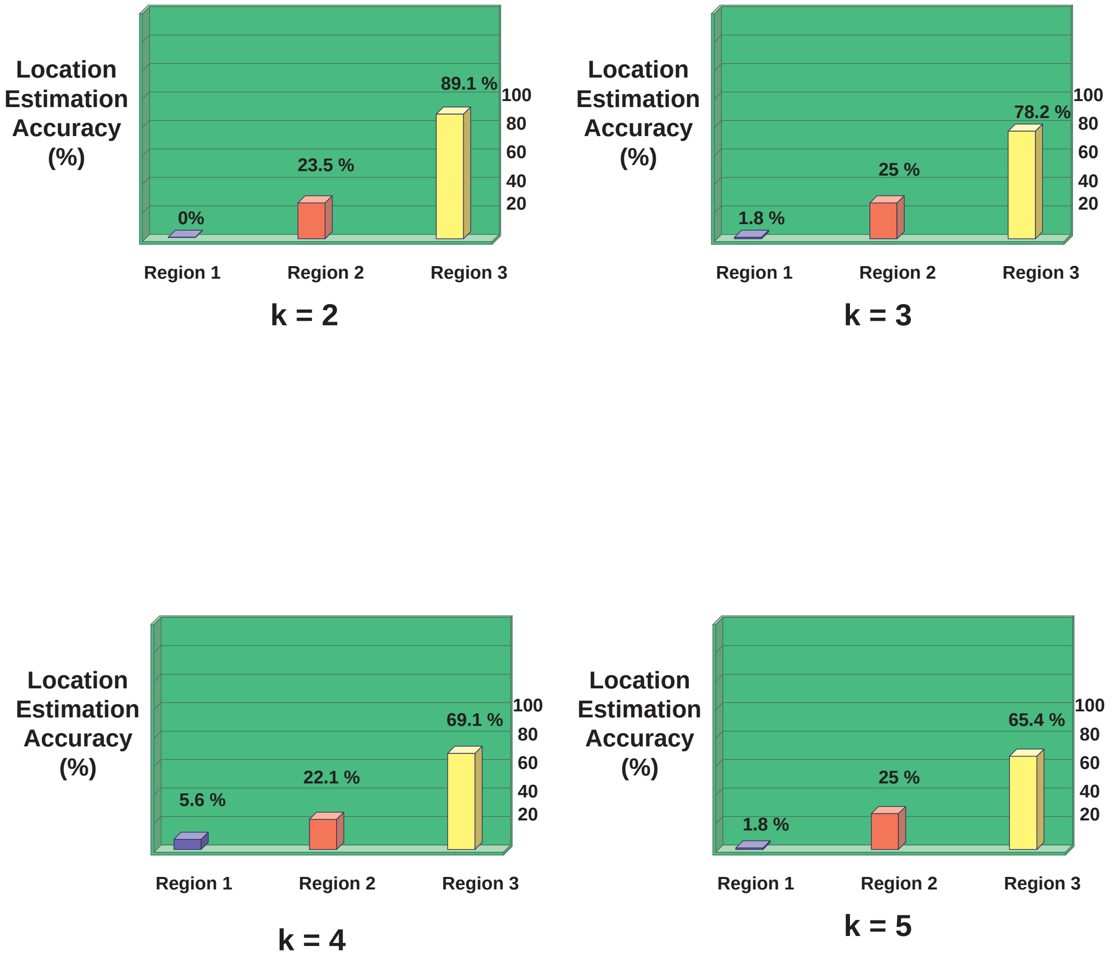
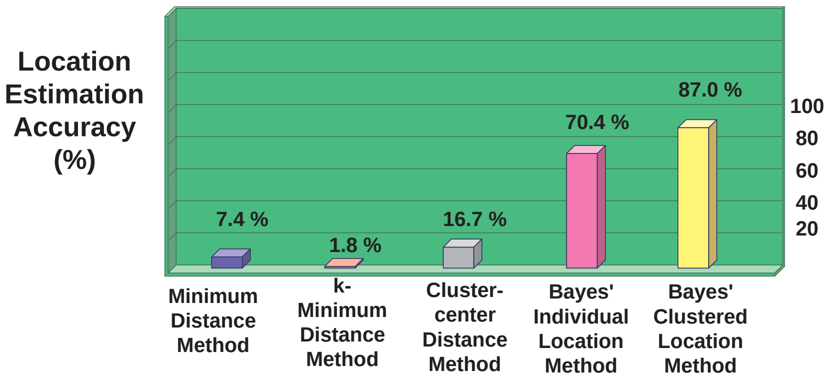


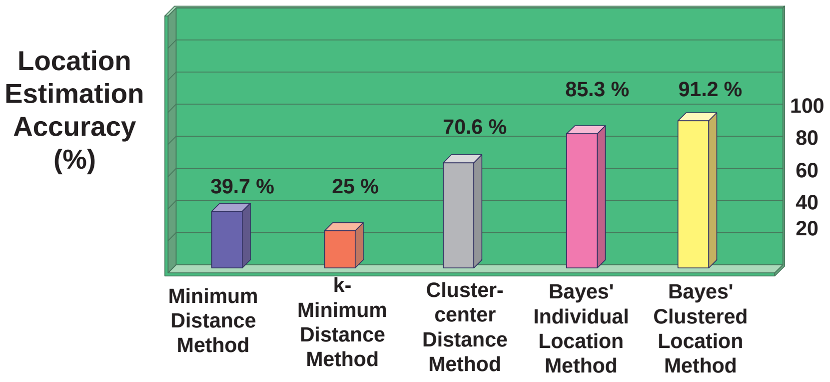
Figure 2.37: Tracking accuracy using the *k Minimum Distance* method with 8 APs. Different values of k were used.

2.10.3 Comparison of different tracking methods

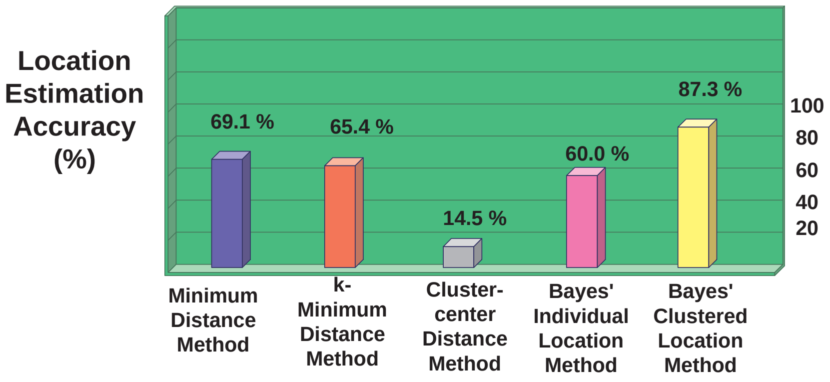
We compare the performance of the different tracking methods. Figures 2.38(a) to 2.41(a) give the comparison for each individual region. Figures 2.42 to 2.45 show the comparison over the entire region.



(a) Accuracy for region 1

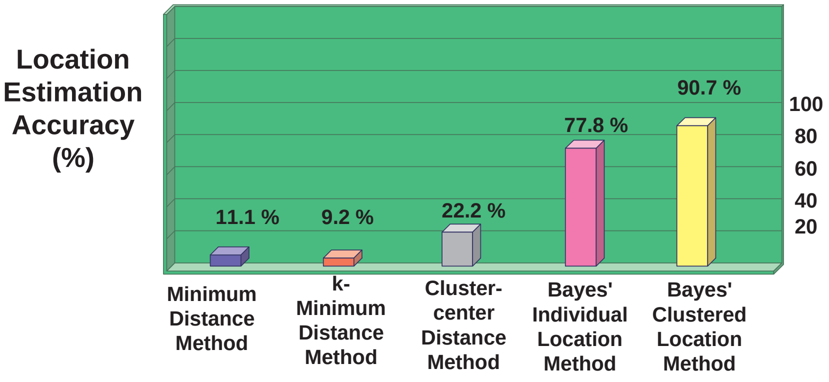


(b) Accuracy for region 2

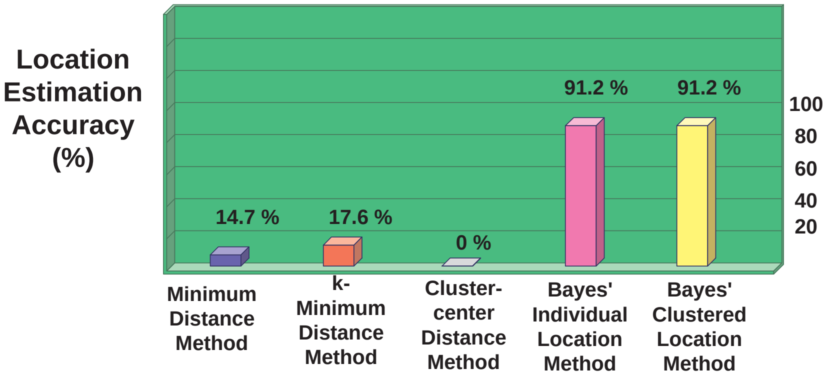


(c) Accuracy for region 3

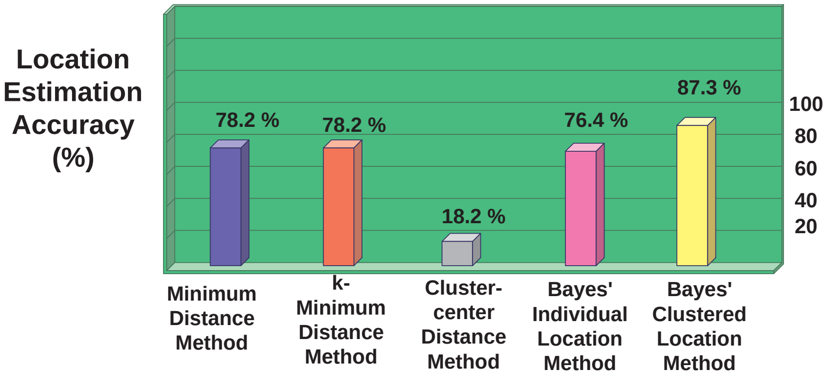
Figure 2.38: Comparison of tracking accuracies of different methods - 8 APs were used for tracking



(a) Accuracy for region 1

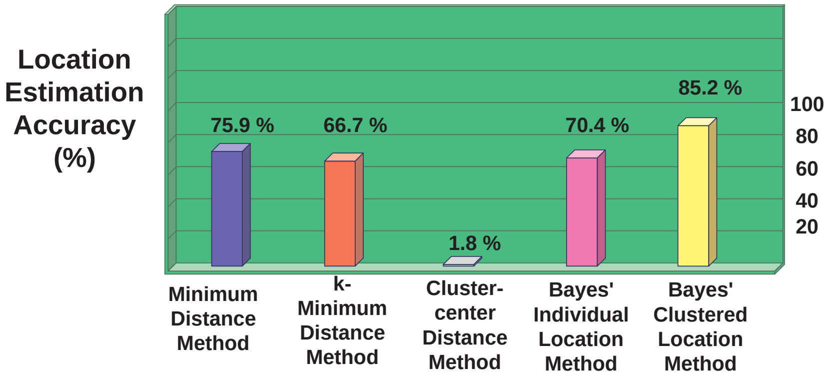


(b) Accuracy for region 2

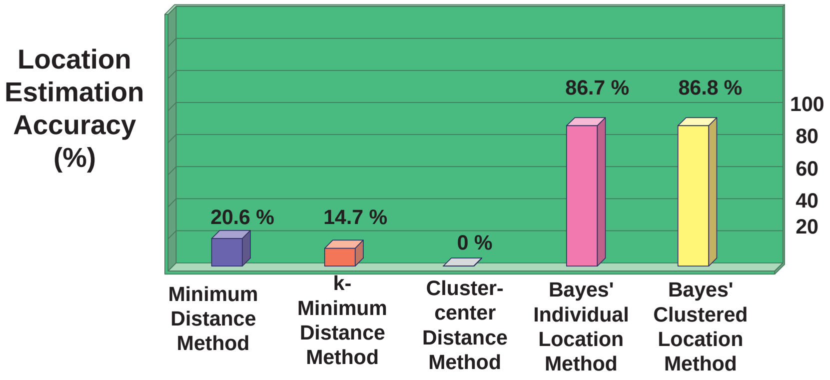


(c) Accuracy for region 3

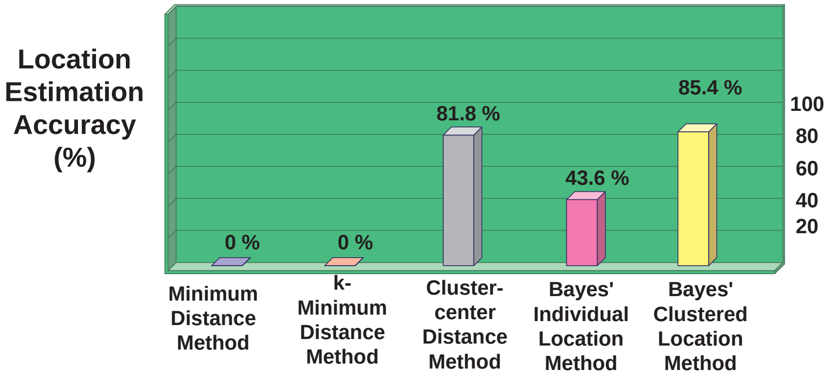
Figure 2.39: Comparison of tracking accuracies of different methods - 6 APs were used for tracking



(a) Accuracy for region 1

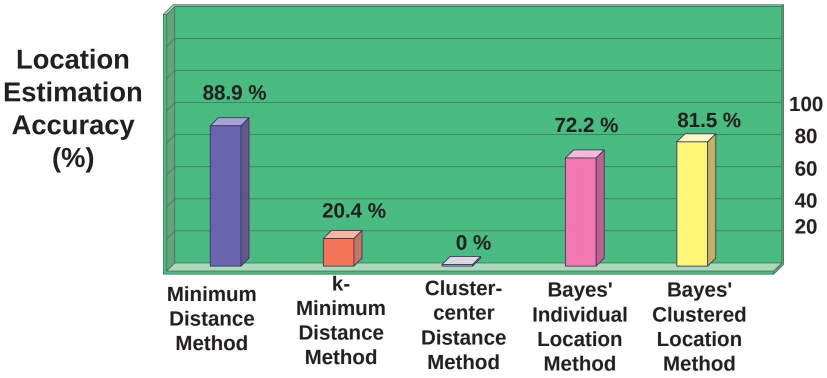


(b) Accuracy for region 2

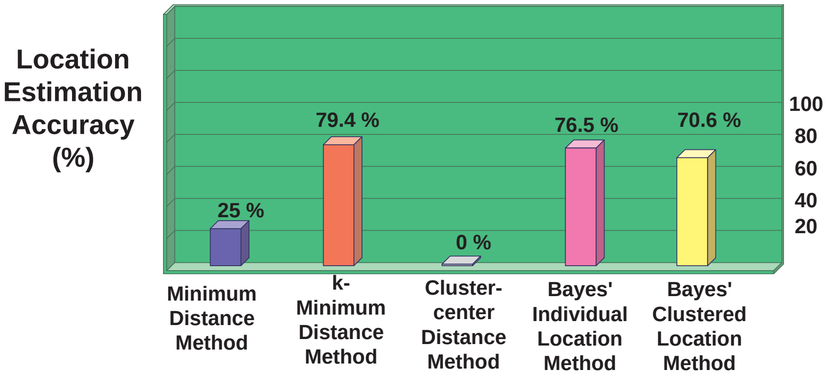


(c) Accuracy for region 3

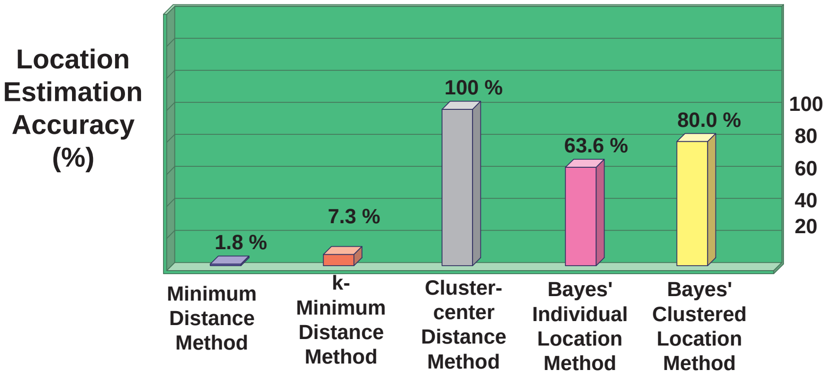
Figure 2.40: Comparison of tracking accuracies of different methods - 4 APs were used for tracking



(a) Accuracy for region 1



(b) Accuracy for region 2



(c) Accuracy for region 3

Figure 2.41: Comparison of tracking accuracies of different methods - 2 APs were used for tracking

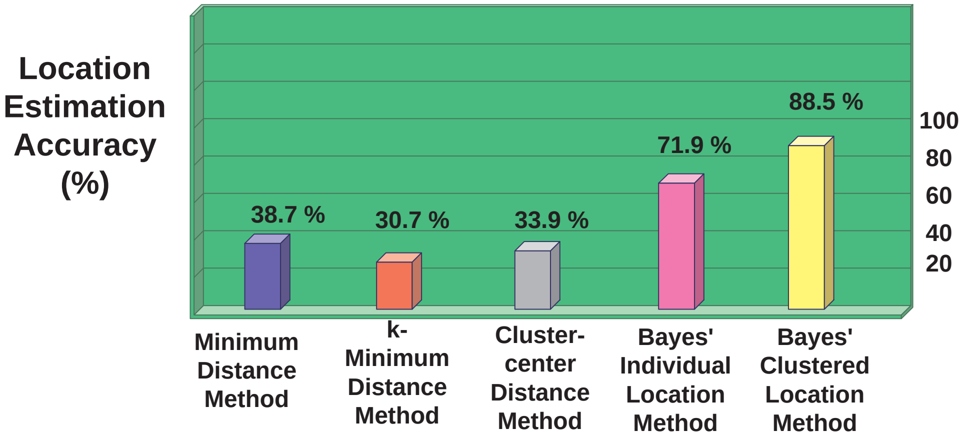


Figure 2.42: Comparison of tracking accuracies over the entire region (combining regions 1,2, and 3) - 8 APs were used for tracking.

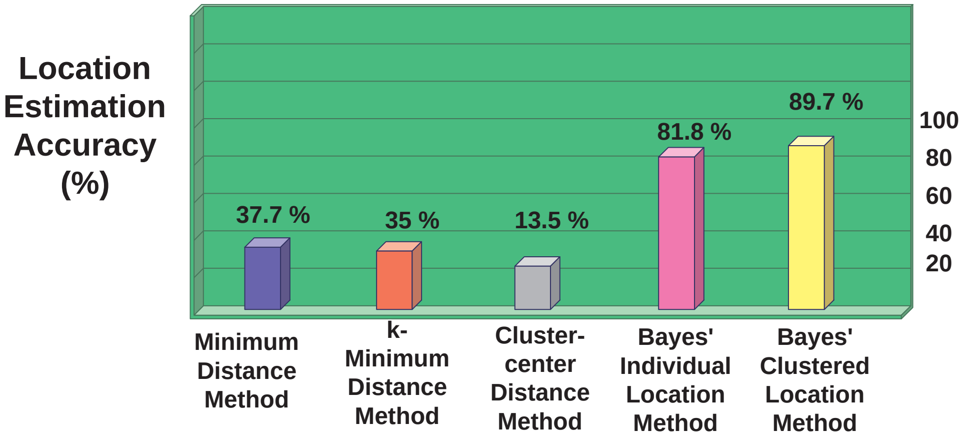


Figure 2.43: Comparison of tracking accuracies over the entire region (combining regions 1,2, and 3) - 6 APs were used for tracking.

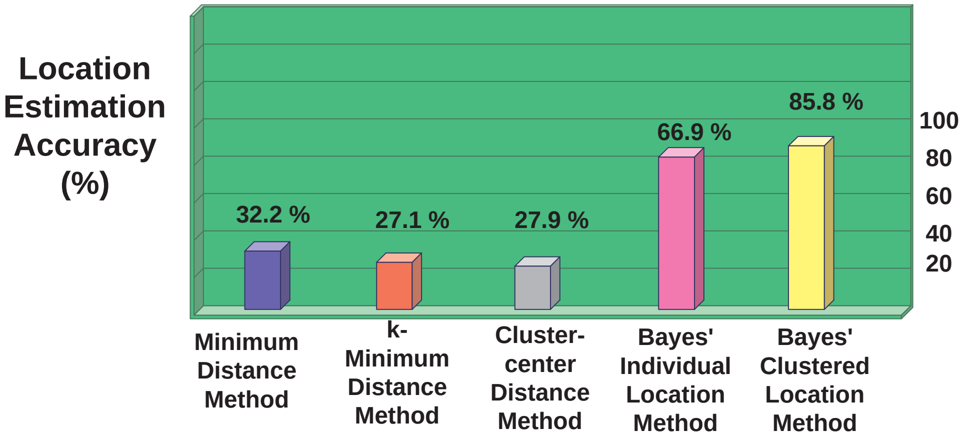


Figure 2.44: Comparison of tracking accuracies over the entire region (combining regions 1,2, and 3) - 4 APs were used for tracking.

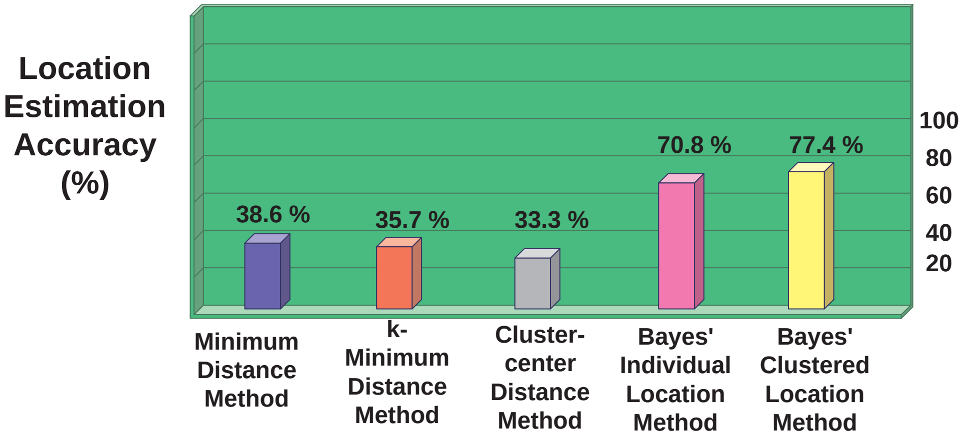


Figure 2.45: Comparison of tracking accuracies over the entire region (combining regions 1,2, and 3) - 2 APs were used for tracking.

Tracking Method	**Tracking Accuracy for the entire region**	**Advantages & Disadvantages**
Minimum distance	• Using 8 APs: 38.7% • Using 2 APs: 38.6%	Faster than k-Minimum distance.
k-Minimum distance	• Using 8 APs: 30.7% • Using 2 APs: 35.7%	Slower than the *Minimum distance* method because extra overhead of manipulating the list of k nearest signals. Extra memory is also required
Cluster-Center distance	• Using 8 APs: 33.9% • Using 2 APs: 33.3%	Faster than both *Minimum distance* and *k-Minimum distance* methods, but not robust enough in their comparison
Bayes' Individual Location	• Using 8 APs: 71.9% • Using 2 APs: 70.8%	Performs better than the distance based methods. Slow real time tracking since probabilities from all the individual locations have to be used
Bayes' Clustered Location	• Using 8 APs: 88.5% • Using 2 APs: 77.4%	Gives the best performance. Faster real-time tracking compared to *Bayes' Individual Location.* Offline preprocessing takes longer time.

Table 2.11: Comparison of different tracking algorithms used in the experimental tests

Looking at the results from figures 2.31 to 2.45, and table 2.11, we can infer that the *Bayes' Clustered Region* approach gives the best tracking results. It performs consistent tracking across all the three regions, and is reasonably robust when lesser number of access points are used.

2.11 Experimental Results for Placement of Base Stations

We wanted to see how the placement of the base stations affects the tracking accuracy of the system. Since the *Bayes' Clustered Region* approach gave the most accurate tracking results, we will use it for investigation into this aspect.

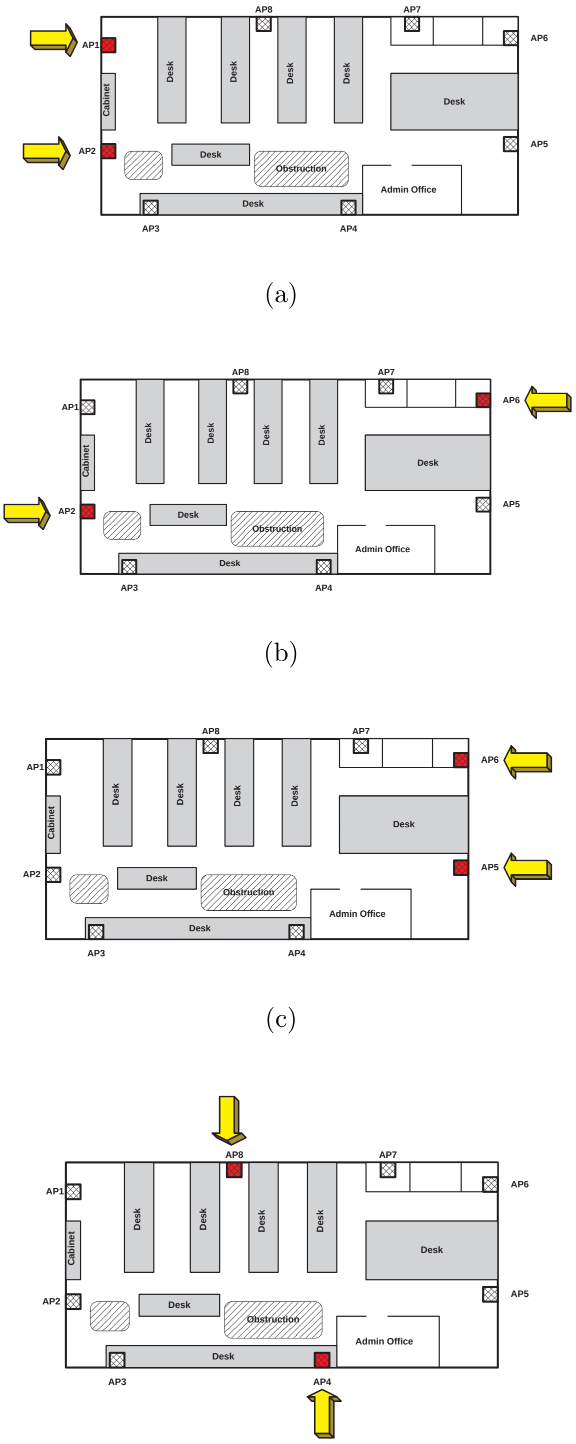
2.11.1 Placement of 2 base stations

We compared the tracking accuracies using the configurations shown in figure 2.46. The tracking results are shown in figure 2.47.

When the APs located were located in region 1 (see figure 2.46(a), the tracking accuracies for region 1 and 3 were high. This was also the case when the APs were place in region 3 (see figure 2.46(a)). This is because the placement of the APs at one extreme allowed the distinct patterns of signals in the nearest and the farthest regions. The nearest region will have the strongest signals, while the farthest region will have the weakest signals. Thus regions 1 and 3 showed high accuracy for configurations shown in figures 2.46(a) and 2.46(a).

We also expect that the tracking accuracies to be less when the access points are located at a central location, as in figure 2.46(d). This is indeed so, as can be seen in figure 2.47(d). This is because the location of the APs is such that the signals in different regions have similar patterns.

Based on the results, we recommend that when 2 APs are available, placing them in a central location should be avoided.



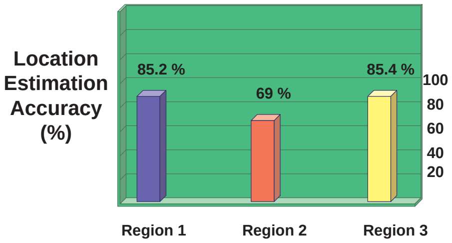
(a)

(b)

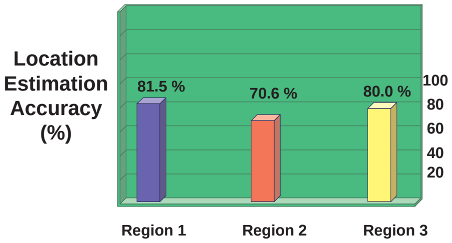
(c)

(d)

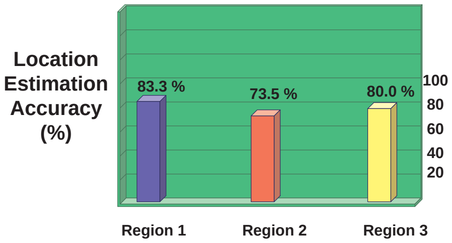
Figure 2.46: Different location configurations of 2 APs.



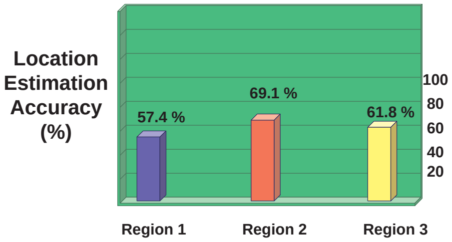
(a) Accuracy for configuration shown in figure 2.46(a)



(b) Accuracy for configuration shown in figure 2.46(b)



(c) Accuracy for configuration shown in figure 2.46(c)



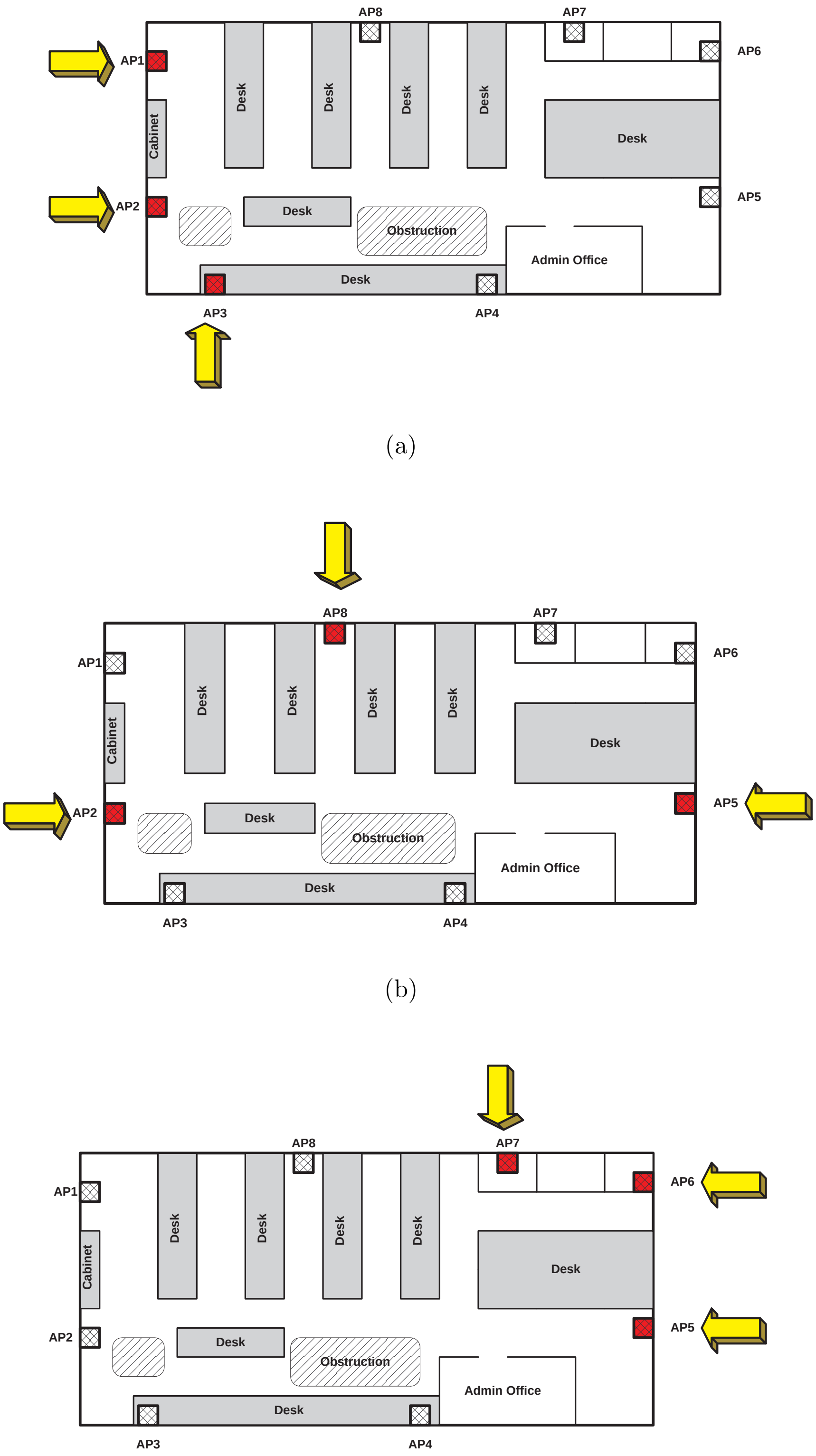
(d) Accuracy for configuration shown in figure 2.46(d)

Figure 2.47: Location accuracies for different configurations.

2.11.2 Placement of 3 base stations

When comparing the tracking accuracies using 3 access points, we used the configurations shown in figure 2.48. In figures, 2.48(a) and 2.48(c), the access points are all located in one region. In figure 2.48(b), we have used one access point from each location region.

The results of the tracking are shown in figure 2.49. As can be seen, the best tracking results were obtained when there was one access point installed in each location region. Clustering all the access points in one region did not provide as good tracking results.

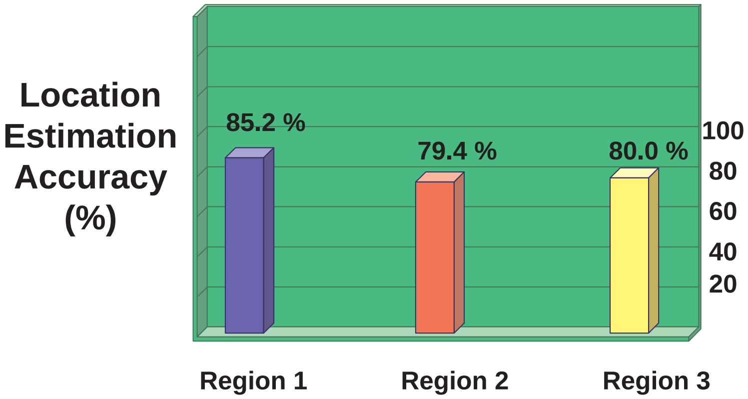


(a)

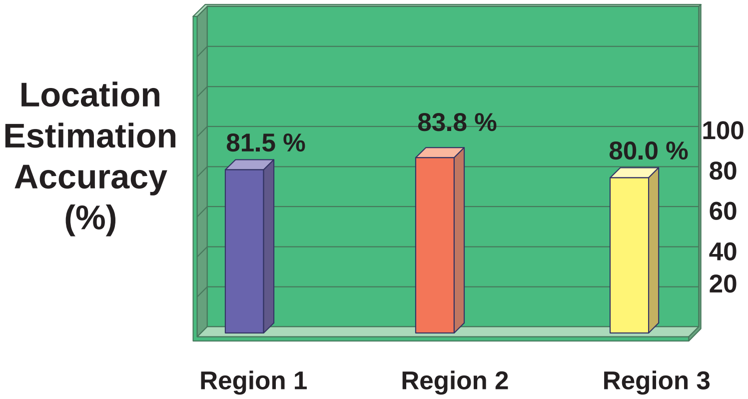
(b)

(c)

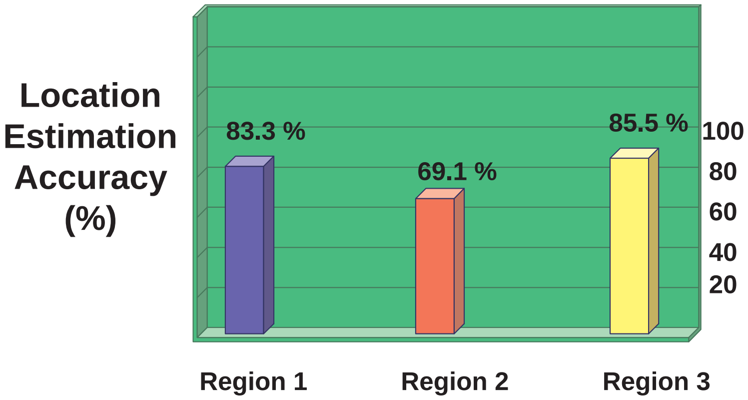
Figure 2.48: Different location configurations of the 3 APs.



(a) Accuracy for configuration shown in figure 2.48(a)



(b) Accuracy for configuration shown in figure 2.48(b)



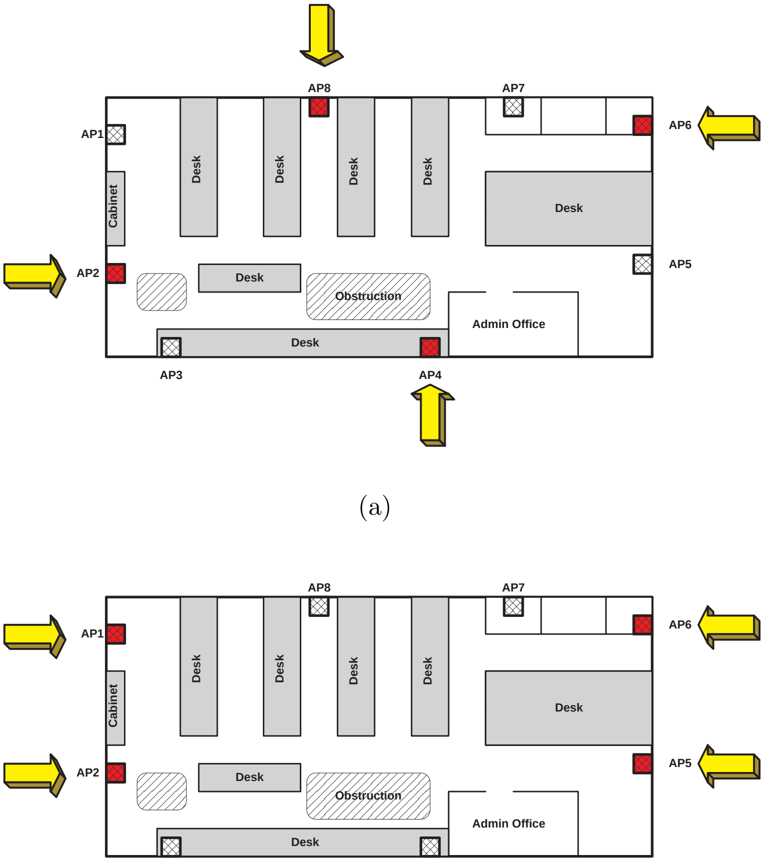
(c) Accuracy for configuration shown in figure 2.48(c)

Figure 2.49: Location accuracies for different configurations of 3 APs.

2.11.3 Placement of 4 base stations

When tracking with 4 access points, we tested the accuracies for the configurations shown in figure 2.50. In figure 2.50(a), the access points are evenly spread out. In figure 2.50(b), 2 access points each are installed at the opposite ends of the room.

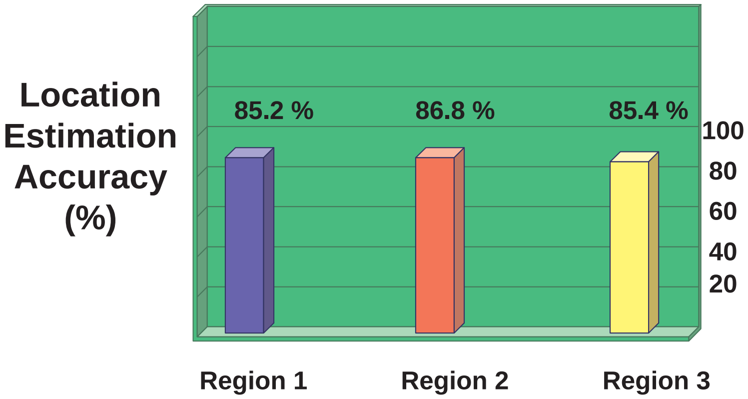
Figure 2.51 shows the tracking accuracies for the corresponding configurations. Here also, placing the access points at the opposite extreme ends allowed us to get better tracking accuracies. In this sense the results are similar to those obtained using 2 access points.



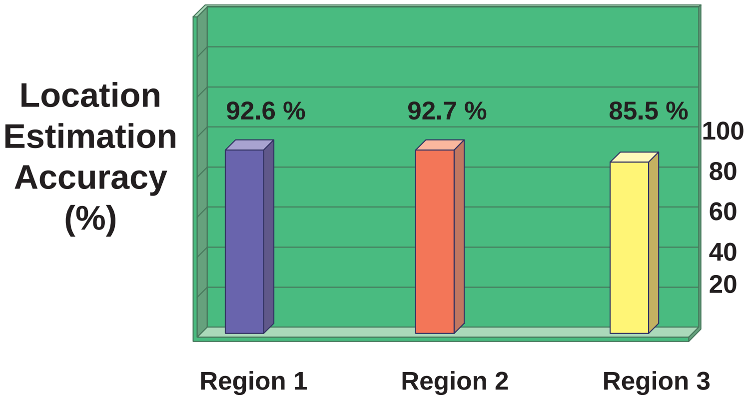
(a)

(b)

Figure 2.50: Different location configurations of the 4 APs.



(a) Accuracy for configuration shown in figure 2.50(a)



(b) Accuracy for configuration shown in figure 2.50(b)

Figure 2.51: Location accuracies for different configurations of 4 APs.

2.12 Conclusion

In this paper we presented our location tracking system for indoor locations. Our system makes use of the existing WLAN infrastructure without making any hardware modifications to the mobile devices or the infrastructure itself. We used a new approach, the *location region* approach, for tracking the mobile devices. The key advantages of this approach are:

- Leads to less complicated training process, compared to that of previous WLAN tracking systems.
- It can better aid location-aware services and mobile devices in indoor environments.
- The tracking system gives the location information as a direct output. Thus no additional database needs to be checked.
- It leads to faster real-time tracking.

We implemented our tracking system making use of three modules –

1. Signal Sampling Module : For sampling of offline signals
2. Signal Preprocessing Module: For pre-processing the offline signals
3. Location Tracking Module: For doing the actual tracking of mobile devices. This module made use of the preprocessed signals and the online signals from the mobile device.

We used a *mobile device assisted, centrally based* architecture when implementing the location tracking module. Thus, the module was installed on a central server, called the Location Server. The mobile device that needs to be tracked assists the location server by scanning for signals from the access points and sending a signal report to the location server. Steps for enforcing privacy and various scenarios for this architecture have been discussed in appendix D.

The main advantages of this architecture are:

1. Minimal disruption of WLAN traffic.
2. Support for easy updates to the location tracking system.

For the tracking algorithms, we employed five different pattern recognition methods. These were:

1. Minimum Distance method
2. k-Minimum Distance method
3. Cluster-Center Distance method
4. Bayes' criteria using individual locations
5. Bayes' criteria using clustered locations

Based on the test results we found that the *Bayes' method using clustered locations* gave the best tracking results. Using 8 access points, the tracking accuracy was 88.5% for the entire area of interest. When only 2 access points were used, the accuracy was 77.4%.

Chapter 3

Augmented Reality on Mobile Phones

3.1 An Overview of Augmented Reality

Augmented Reality (AR) can be described as a system that supplements the real world with virtual objects that appear to coexist in the same space as the real world. The development of AR is tightly coupled with other related fields of research such as Mixed Reality (MR) and Virtual Reality (VR). Being multidisciplinary in nature, AR encompasses computer vision, image processing, computer graphics, machine learning and human-computer interaction techniques.

3.1.1 Difference between AR and VR

Augmented Reality (AR) is a variation of Virtual Reality (VR). In VR, a user is completely immersed inside a synthetic environment. In contrast, An AR environment is a combination of the real scene viewed by the user and a virtual scene generated by the computer. Therefore, AR supplements reality, rather than completely replacing it. Milgram [27] describes a taxonomy that shows the relationship between augmented reality and virtual reality. The Reality-Virtuality continuum defined by him is shown in Figure 3.1. The real environment and virtual environment are located at the two extreme ends of this continuum. Real environment defines any environment consisting solely of real objects. Virtual environment defines environments consisting solely of virtual objects generated by the computer. The region between real environment and virtual environment is known as Mixed Reality (MR) where the real world and virtual world objects are combined in a single display. AR is closer to the real world. It describes that class of displays that consists primarily of a real environment, with graphic enhancements or augmentations. Augmented Virtuality (AV), on the other hand, describes that class of displays that enhance the virtual experience by adding elements of the real environment. Figure 3.2 shows the illustration of reality, mixed reality and virtual reality.

A more specific definition to AR is given in [28]. An AR system is expected to have the following properties:

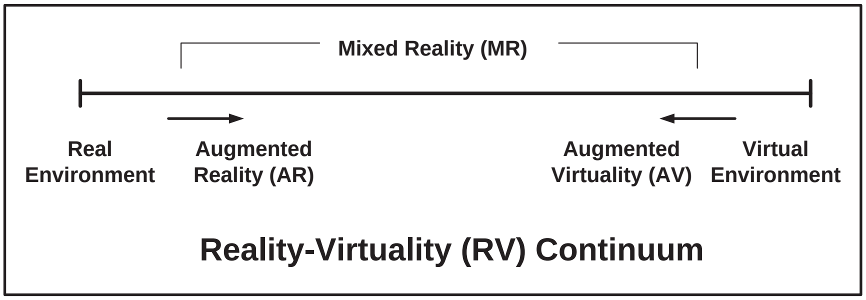


Figure 3.1: Reality-Virtuality Continuum chart

Figure 3.2: Illustration of Reality, Mixed-Reality and Virtual Reality

1. Combines real and virtual objects in a real environment.
2. Registers (aligns) real and virtual objects with each other.
3. Runs interactively, and in real time.

3.1.2 Applications

AR has great potential in various applications because it can enhance a user's perception of and interaction with the real world. Only recently have the capabilities of real-time video image processing, computer graphic systems and new display technologies converged to make possible the display of a virtual graphical image correctly registered with a view of the 3D environment surrounding the user. Researchers working with augmented reality systems have proposed them as solutions in many domains. The areas that have been discussed range from entertainment to military training. In addition, some of its potential applications include medical imaging, maintenance and repair, annotation and visualization. This section will highlight some of the proposed applications for augmented reality.

A simple form of augmented reality has been in use in the entertainment and news business for quite some time. Whenever you are watching the evening weather report the weather reporter is shown standing in front of changing weather maps. In the studio the reporter is actually standing in front of a blue or green screen. This real image is augmented with computer generated maps using a technique called chroma-keying. It is also possible to create a virtual studio environment so that the actors can appear to be positioned in a studio with computer generated decorating.

A more elaborate form augmented reality in entertainment is shown the outdoor game ARQuake[29]. ARQuake is an augmented reality version of the popular Quake game. It is designed for play outdoors using augmented reality technology. Using a head mounted display (HMD), the user can see both the Quake game, and also the real world at the same time. ARQuake takes its view information from a GPS and orientation sensor, and so as you walk around, Quake moves in sync with the real world. Monsters and buildings appear to sit in the real world as though they were really there, and then you can play a game of Quake while you are in the real physical world. Figure 3.3 shows the AR view when the game is played.

Figure 3.3: Augmented reality view when ARQuake is played

Many future military operations are expected to occur in urban environments. These complex, 3D battlefields introduce many challenges to the dismounted warfighter. Better situational awareness is required for effective operation in urban environments. However, delivering this information to the dismounted warfighter is extremely difficult. For example, maps draw a user's attention away from the environment and cannot directly represent the three-dimensional nature of the terrain.

To overcome these difficulties, the Battlefield Augmented Reality System (BARS) [30] was developed in the Virtual Reality Laboratory at the Naval Research Laboratory, Washington, DC. The system consists of a wearable computer, a wireless network system, and a tracked see-through head-mounted display (HMD). The computer generates graphics that, from the user's perspective, appear to be aligned with the actual environment. For example, a building could be augmented to show its name, a plan of its interior, icons to represent reported sniper locations, and the names of adjacent streets.

3.1.3 Challenges in Augmented Reality

There are many challenges in the research of augmented reality. One of the most important issues is the registration problem. The objects in the real and virtual worlds must be properly aligned with respect to each other, or the illusion that the two worlds coexist will be compromised. To achieve this, sensor-based and vision based registration techniques or a hybrid of both have been employed to solve the problem. This thesis focuses mainly on one type of vision-based registration technique called fiducial tracking.

3.1.4 Registration Methods for Augmented Reality

Registration research in augmented reality basically evolves around 2 major techniques: vision-based and sensor-based. As registration is a difficult problem to be solved by a single method, there has been a trend to incorporate two or more techniques for registration. Such systems employ hybrid-tracking techniques (such as magnetic and video sensors) to exploit strengths and compensate weaknesses of individual tracking technologies. Sensor-based techniques are mostly accomplished by using magnetic, mechanical, ultrasonic or optical technologies. Due to the extensive infrastructure that is required to support the tracking, the displacement of the sensor-based AR system is restricted. These sensors are also easily affected by perturbations from the environment and the accuracy of the camera pose measurements are also known to be poor, thus limiting their usefulness for AR applications[31].

For this thesis, the AR registration method used is fiducial-based tracking, which is a vision-based technique. Fiducial-based tracking works by deliberately introduces artificial fiducials such as paper markers into the scenes where virtual objects are to be introduced. Registration is accomplished by tracking the known contour shapes or by calculating the centroids of the markers. For example, ARToolkit [32] tracks the 4 outer corners of the square markers. The 3-D world coordinates of the marker features are measured a priori and given the 2-D coordinates of the detected features in the image, a correspondence between 3-D and 2-D is set up. Pose estimation techniques [33] can then be employed to recover the 3-D camera pose.

These markers are inexpensive to produce and methods are simple and can be implemented in real-time using normal desktop computer. However, camera tracking can be easily lost as it is only based on a few known features of the markers. For example, in the case of the ARToolkit, a partial occlusion to any one of the four outer corners will fail the tracking completely. To lessen the effect of varying lighting conditions, these markers are often painted in black. Also, relatively simple geometrical patterns are used in the markers to facilitate template matching to identify the markers.

3.2 Related Work

The Mobile Augmented Reality Systems (MARS) project [34] presented in 1999 by Columbia University was one of the first truly mobile augmented reality setups which allowed the user to freely walk around while having all necessary equipment mounted onto his back. This project demonstrates one of the earliest prototype of wearable computers.

Other researchers have focussed on wide-area tracking based AR applications. A number of tracking methods have been used e.g., GPS, inertial sensors [35], and computer vision [36]. Optical tracking is another approach that has been looked into. The ARToolKit in such example. It tracks fiducial markers using cameras.

When PDAs were mass-marketed, Augmented Reality projects that are based on off-the-shelf PDAs surfaced. Most Augmented Reality projects that use these devices primarily as displays and outsource most challenging tasks such as rendering, tracking and application intelligence to stationary servers. In the AR-PDA project [37] a server/client architecture was developed for digital image streaming from and to an application server. The PDA acted as a client whereas the server was responsible for natural feature tracking and rendering tasks. A similar system, the Augmented Reality PDA [38], was also developed by Siemens.

In addition, researchers at Daimler-Chrysler had developed an interesting server/client based AR project where they used analog video transmission from and to the handheld device [39].

AR projects on PDAs are not restricted to server/client based. The Handheld AR project [40] from Vienna University of Technology uses a PDA running the AR toolkit. This has allowed the PDA operate without external servers. However, it still provides the option to use wireless technologies if available to offload the AR processing to a server.

This project differs in using the mobile phone instead of the PDA as an AR interface. Since more people own mobile phones than PDAs it is intuitive to find out whether mobile phones are suitable as an AR interface.

The CyPhone project [41] implemented by University of Oulu, Finland in 2000 was one of the first attempts to bring augmented reality to mobile phones. In addition to voice calls, it has been designed to support context-specific and multi-user multimedia services in an augmented reality manner. Context-awareness has been implemented with GPS-based navigation techniques and a registration algorithm, capable of detecting a predefined 3-D model or a landmark in the environment. This system consists of a small CPU which is strapped around the waist and a HMD for AR viewing. It is relatively bulky and not commercially available.

To date, this project is one of the first in the world which focusses on the use of an off-the-shelf mobile phone as an AR interface. No extra hardware is added onto the phone. The main advantages of the prototype system are as follows:

- *Ease of setup:* Since no extra hardware is added to the phone, it is possible to quickly load an AR software to an off-the-shelf mobile phone and start using it.

- *Ease of use:* No head mounted displays (HMDs) have been used, thus the user does not have to carry any extra hardware to view the AR application.

3.3 Mobile Phones and Augmented Reality

Mobile Augmented Reality makes it possible to create new kind of services and applications. Some example applications of mobile AR include personal navigation, guidance systems, security, entertainment, e-commerce and personal services. If AR services should be offered to a wide range of possible users, standard consumer electronic devices have to be applied. By making AR work in mobile phones we face new challenges in positioning, registration, system performance and energy consumption. Currently, most mobile AR systems use specialized laptops, wearable computers (e.g. Cybernaut, Espresso) or special AR hardware rather than mobile phones.

Augmented Reality applications require processing power that is beyond the capability of present day mobile phones. Existing processors in mobile phones like Sony Ericsson P800's ARM9 are not powerful enough for AR processing. Therefore, most of the computation has to be done on a server which is connected to the mobile phone. The most popular wireless connection today is Bluetooth [42] and GPRS . If the system has to be deployed over city-wide geographical regions, GPRS is the obvious choice. Bluetooth can be used if the server is close enough (10 - 100m). Any realistic AR application for mobile phones should use the phone only to capture a raw input image (and some simple preprocessing, if possible), transfer them to an AR server and receive some form of augmented result.

Another shortcoming of mobile phones is their limited graphics display capabilities and small screen size. Hence, the interface on the phone should be simple and intuitive. It should merely act as a viewport of the AR application, with the bulk of the processing taking place in a AR server.

We expect that future mobile phones will significantly increase their effectiveness to a level similar to today's standard personal computers. Furthermore, the evolution of cellular networks and related technologies has recently been very fast. The transfer rates and low latencies for packet data required by many AR applications are feasible using the upcoming 3rd generation cellular networks. The bandwidth of wireless network standards like UMTS [43], which is based on 3G systems, will allow streaming interactive video and audio in real time.

Our philosophy for the AR phone system is that the interface on the phone should be intuitive and relatively passive. It should simply act as a viewport of the AR application, with the bulk of the processing taking place in an AR server. This keeps the applications that need to run on the phone simple, allowing them to be deployed across a range of phones with different capabilities.

Following our design philosophy, we developed three separate modules -

1. Mobile Phone Module
2. AR Server Module

3. Wireless Communication Module

The mobile phone module runs on a mobile phone enabling it to communicate with a AR server module via the wireless communication module, which is deployed using the low-power, low-range Bluetooth technology. The AR server module, which runs on a desktop/laptop, processes the image input and returns the augmented results to the phone for display to the user. The images are sent through the system as compressed image (see Figure 3.4).

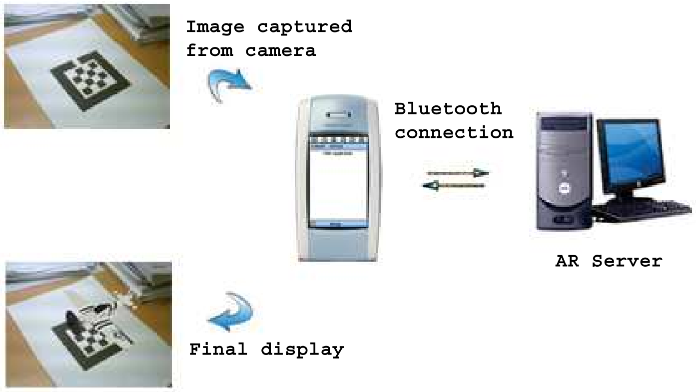


Figure 3.4: Prototype system flowchart

3.4 Novelties

To exploit this new idea of interfacing Augmented Reality with mobile phones, two applications has been built. They are the *AR Catalogue* and *AR Post-It*.

The first AR application is the *AR Post-It*, which aims to replace the conventional Post-It pads which is commonly used in workplaces and households. This system combines the speed of traditional electronic messaging with the tangibility of paper based messages. The key concept of the AR Post-It system is that the messages are location specific. In other words, the messages are displayed only when the intended receiver is within the relevant spatial context. This is done by deploying a number of fiducial markers in different locations. Messages are posted remotely over the internet and the sender can specify the intended recipient as well as the location of the message. The messages are stored in a server, and downloaded onto the phone when the recipient uses his phone's digital camera to view a marker. The phone connects to the server using Bluetooth.

The second AR application is the *AR Catalogue* which aims to enhance the reading experience of consumers. In addition to viewing 2D images of objects on normal catalogues, handouts or magazine, readers can use the AR phone system to view those objects in full 3D. These 3D virtual characters are rendered into the actual scene captured by the mobile phone's camera. Moreover, these 3D characters can be viewed from different perspectives allowing consumers to interact with them. A server is used to do the image processing tasks and the phone connects to the server using Bluetooth.

3.5 Implementation of the Prototype System

3.5.1 System Design

Hardware Details

- **Mobile Phone**

  The user interface to the system has been developed on a Sony Ericsson P800 mobile phone running version 7.0 of the Symbian OS. The P800 is a good choice since it includes a built-in camera, a large color screen and Bluetooth communication in the standard model. Table 3.1 shows the specifications of this mobile phone.

- **Desktop Computer**

  A Pentium 4, 2.0 GHz computer was used as a server to handle the AR processing and communicating with the mobile phone over Bluetooth. The computer was equipped with 256MB RAM and an *NVIDIA GeForce 4 MX 440* video card.

- **Bluetooth Adapter**

  A Bluetooth adaptor is needed by the AR server module to establish a wireless connection with the mobile phone. We used a TDK Bluetooth adaptor that had a range of 50m in free space and 10m in closed room.

Feature	Description
Picture	
System	Triple band GSM 900/1800/1900
Weight	148g without flip 158g with flip
Screen	1/4 VGA color touchscreen 4096 colors 40 x 28 mm/208 x 144 pixels (closed) 40 x 61 mm/208 x 320 pixels (open)
Camera image size Color depth	up to 640 x 480 pixels 16 million colors (24 bit)
Wireless Connectivity	Bluetooth technology GPRS Infrared port
Development Environment	C++, Java, Visual Basic

Table 3.1: Sony Ericsson P800 specifications

Software Details

The prototype system comprised two modules:

1. *Mobile Phone module*: Responsible for user interface and image capturing of the fiducial markers.
2. *AR Server module*: Responsible for image processing and marker recognition tasks.

3.5.2 Mobile Phone Module

The mobile phone module is entrusted with the tasks of image capturing using the phone's digital camera, transfering it to the AR server via Bluetooth and displaying the augmented result on the phone. Figure 3.5 gives the flowchart of the mobile phone module.

The mobile phone module was installed on the Sony Ericsson P800. Since the P800 does not support capturing video from the internal camera, the only alternative left was to capture still images, send it to the server and wait for the augmented result. At this point a new image could be captured and the process repeated as often as desired. The P800's digital camera supports image capture at three resolutions: 640×480, 320×240, and 160×120.

For the wireless communication, the mobile phone module used Bluetooth L2CAP connection. L2CAP (Logical Link Control and Adaptation Layer Protocol)is a Bluetooth protocol that provides connection-oriented and connectionless

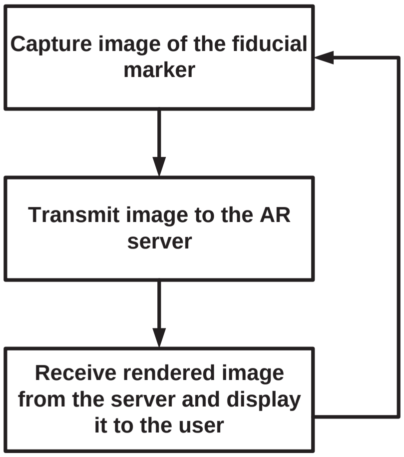


Figure 3.5: Mobile Phone Module flowchart

data services to upper layer protocols.

The algorithm of the mobile phone module can be divided into two parts. The first part of the algorithm focusses on capturing an image and sending it to the AR server module. It has the following steps:

1. Software starts up and reserves the camera on the mobile phone for this application.
2. Create a memory buffer to store one image and the viewfinder.
3. User starts inquiry of Bluetooth devices and selects the AR server.
4. Mobile phone initiates L2CAP connection with AR server.
5. If connection is successful, the software displays video stream from the camera on the viewfinder.
6. User clicks the capture button and the mobile phone captures a image, resizes its resolution to 320×240 and stores it in the memory buffer.

7. Apply JPEG[44] compression to the data in memory buffer and write it into temporary file.
8. Read temporary JPEG file into memory as binary data.
9. Binary data is broken into packets smaller than 672 bytes each.
10. Send a “start” string to the server to indicate the start of an image.
11. Send one packet of data to the server and wait for confirmation from server.
12. When receive confirmation, send the next packet until the all the packets from that image are sent.
13. Send an “end” string to the server to indicate the end of an image.
14. Wait for AR server module to return rendered image.

Binary data of the image is broken into packets of 672 bytes each due to constraints in the L2CAP protocol. This will be further explained in Section 3.6. Figure 3.6 shows the flowchart of the first part of the algorithm.

The second part of this algorithm focusses on receiving the rendered image from the AR server module and displaying it on the screen. It has the following steps:

1. Receive one packet of data of the rendered image from the AR server module.
2. Append binary data to memory buffer.
3. Send a confirmation packet to the AR server module.

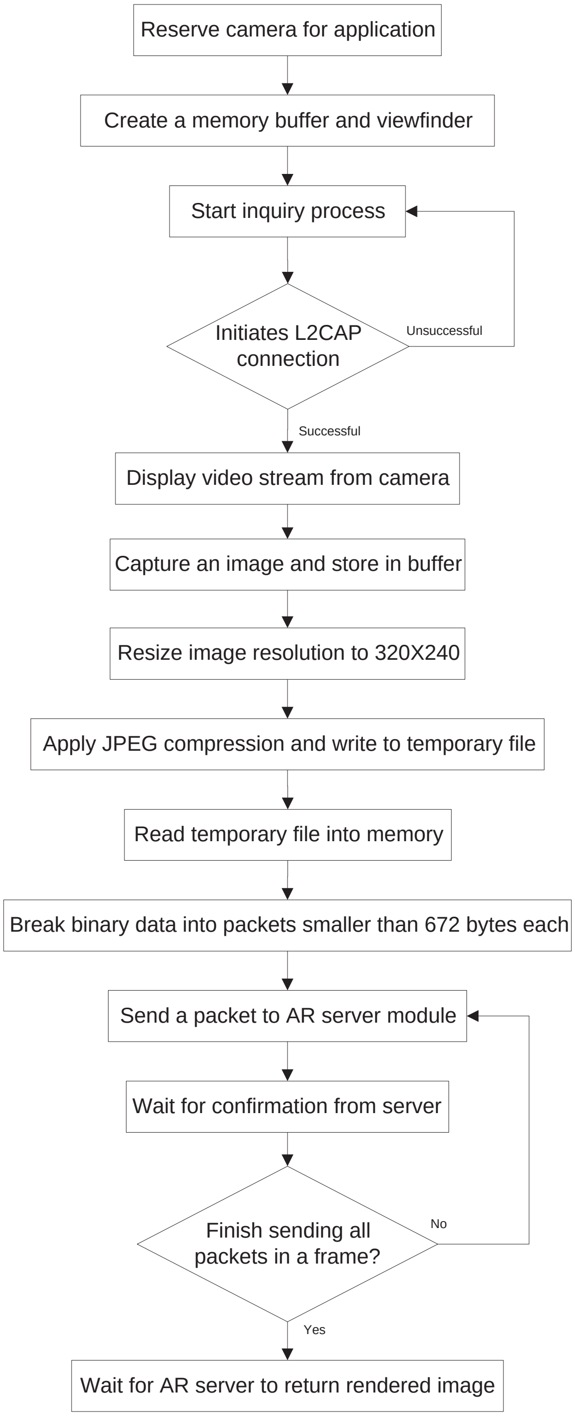


Figure 3.6: Flowchart showing how the mobile phone captures an image and sends it to the AR server module

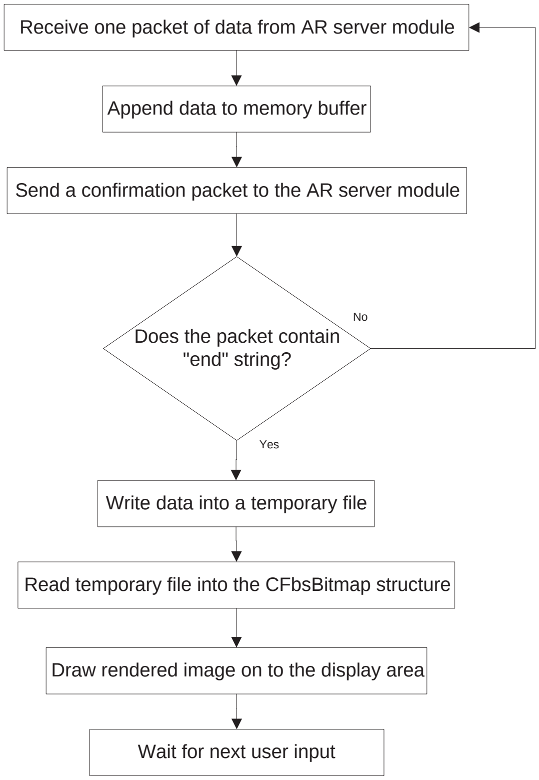


Figure 3.7: Flowchart showing how the mobile phone receives an image from the AR server module and displays it on to the screen

4. Wait for AR server module to send the next packet until the string “end” is received

5. Write binary data of rendered image in memory buffer to a temporary file.

6. Read temporary file into the CFbsBitmap structure.

7. Draw rendered image on to the display area.

The CFbsBitmap format is internal to Symbian UIQ SDK. Figure 3.7 shows the flowchart of the second part of the algorithm.

3.5.3 AR Server Module

The AR server is a generic module that is able to receive image input from the mobile phones and perform processing and manipulation of the data. The processing and manipulation of data is done mainly using the MXR Toolkit [45], which is a marker-based tracking toolkit developed in our lab. The MXR Toolkit consists of a wide range of routines to help with all aspects of building mixed reality applications. The AR server module examines the input image for a particular fiducial marker. If a marker is found it will attempt to recognize the pattern in the center of the marker. The MXR Toolkit allows us to differentiate two different markers with different patterns even if they are placed side by side. Hence, this allows us to overlay different virtual objects on different markers. After processing of the image, the result is returned to the phone. The success rate of pattern recognition is dependent on the resolution of the image, the size of the fiducial marker and the distance between the mobile phone and the fiducial marker. These issues will be further discussed in section 3.10. Figure 3.8 illustrates the basic operation of the MXR Toolkit.

3.6 Wireless Communication for the AR Phone System

It is important to select a suitable wireless communication technique so that the data transfer between the phone and the server can be done in the fastest possible

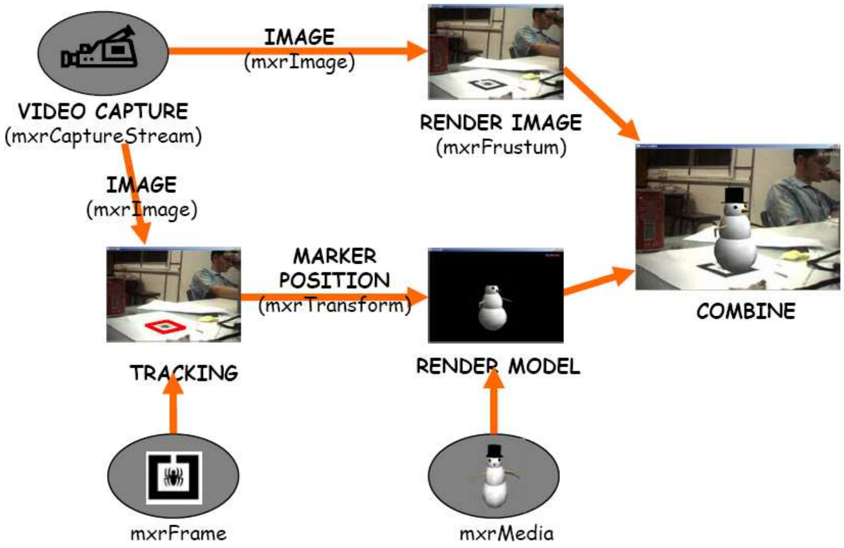


Figure 3.8: Operation of the MXR Toolkit

way. The chosen wireless technique should ideally allow user roaming over wide areas.

The most popular wireless technologies for data communication for mobile phones are Bluetooth and GPRS. GPRS communication makes use of the existing GSM network and thus allows city wide roaming flexibility to the user. It operates in the 900 MHz frequency band and has a maximum theoretical data rate of 171.2 kbps. Bluetooth is a short range communication technology that operates in the 2.4 GHz ISM [1] frequency band. It has a range of 10m in closed spaces and 100m in open spaces and has maximum data rate of 721 kbps and 432 kbps for asymmetrical and symmetrical connections respectively.

For our project, Bluetooth was the preferred choice because of the following reasons:

---

[1] Industrial, Scientific, and Medical

1. It is relatively easy and flexible to implement as compared to GPRS.
2. Accessing the ISM frequency band is free of charge, while accessing the GPRS network requires payment to the service provider. Thus for laboratory conditions, it is far cheaper to carry out all the testing using Bluetooth.

3.6.1 Bluetooth Protocols

Bluetooth is based on a stacked protocol model in which the communication is divided into several layers. The core lower layers consist of the Baseband, the Link Manager Protocol (LMP), the Host Control Interface (HCI). The core higher layers are the Logical Link Control and Adaptation Protocol (L2CAP), and the Service Discovery Protocol (SDP). The L2CAP employs the concept of channels to keep track of where data packets come from and where they should go. SDP defines how a Bluetooth client's application shall act to discover available Bluetooth servers' services and their Bluetooth characteristics.

Apart from the core protocols, Bluetooth has a cable replacement protocol. This is the RFCOMM protocol, whose purpose is to emulate a serial port. The protocol covers applications that use serial ports of the kind used in PCs. Thus, RFCOMM emulates RS-232 control and data signals over the Bluetooth baseband. It provides transport capabilities for upper level services, such as OBEX (Object Exchange protocol).

The OBEX, or IrOBEX, as is the correct term, is an optional application layer protocol designed to enable units supporting infrared communication to exchange a wide variety of data and commands in a resource-sensitive standardized fashion. OBEX uses a client-server model and is independent of the transport mechanism and transport API. The OBEX protocol also defines a folder-listing object, which is used to browse the contents of folders on remote device. RFCOMM is used as the main transport layer for OBEX.

Figure 3.9 shows the Bluetooth hierarchy.

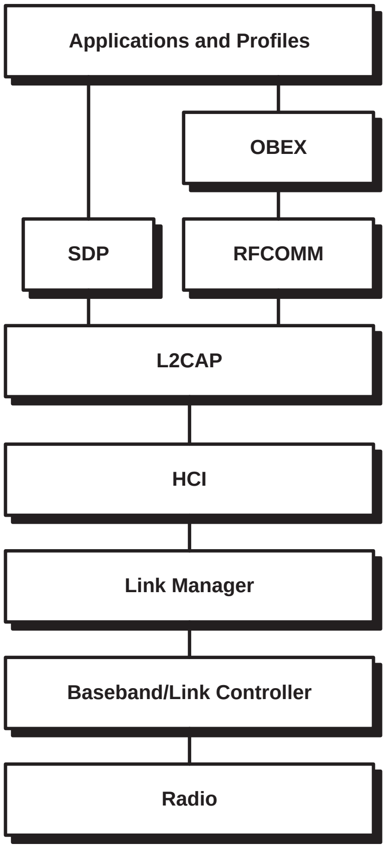


Figure 3.9: Different Layers in the Bluetooth Hierarchy

3.6.2 Protocol Selection

The Bluetooth protocol stack consists of optional protocols as well as the core protocols and therefore we had the following three options were available for the transfer of data between the mobile phone and the AR server:

1. *Using OBEX protocol:* The advantages of this method are:

    - High level implementation
    - Has parity check
    - Simple programming interface

    However, this method has the disadvantage of low data transfer rate, since the OBEX layer makes use of RFCOMM as well as the L2CAP protocols at the lower layers. Thus additional header bits of the lower layer protocols encapsulate the OBEX packet, decreasing the amount of useful data transferred per second.

2. *Using RFCOMM protocol:* The advantages of this method are:

    - High level implementation
    - Has parity check
    - Higher data rate as compared to the OBEX implementation.

    The disadvantage here is that the programming implementation is slightly more complicated than the OBEX implementation.

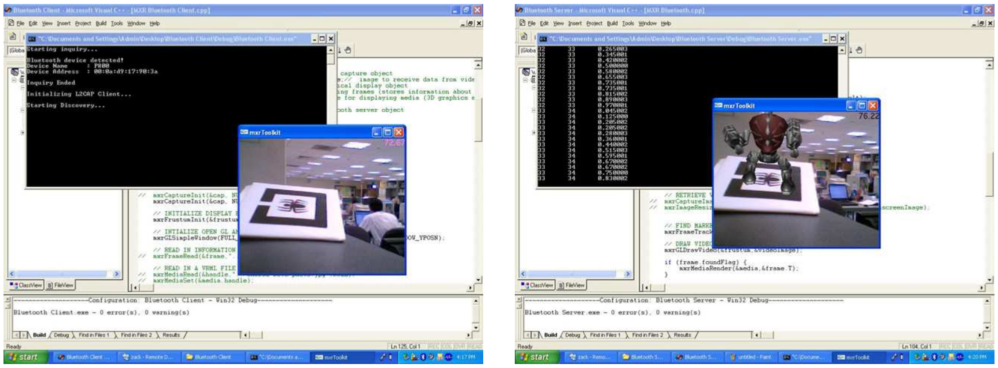
3. *Using L2CAP:* The advantage of this implementation is that it has a higher data rate than the RFCOMM and OBEX implementations.

   However, the disadvantages are:

   - Low level implementation, there the programming implementation is the more complicated than the RFCOMM and the OBEX implementations.
   - No parity check, only CRC (Cyclic Redundancy Check) in the baseband

We chose to implement the L2CAP method because it offers superior data transfer rates compared to the other 2 methods. Although there is no parity check in L2CAP, CRC (Cyclic Redundancy Code) in the baseband is sufficient to detect errors in data transmission. It is important to note that the default Maximum Transmission Unit (MTU) is 672 bytes, as specified in [46]. Thus, L2CAP will not transmit packets greater than 672 bytes. Although it is possible to use a larger default value, no mechanism to determine the optimum value exists. We therefore did not change the default value and used simple segmentation and reassembly to break down the large image files ($5000 - 15000$ bytes after JPEG compression) into L2CAP packets of size 672 bytes each. These L2CAP packets were then reassembled at the receiving end to get back the original image.

The segmentation and reassembly of L2CAP packets was first tested using two desktop PCs, one of them capturing raw input images from a web camera and transferring the images to a remote PC using Bluetooth. Figure 3.10 shows the screen shots from the client and server PCs. The client PC acted as the mobile phone, while the server PC's task was to reassemble the segmented L2CAP packets and attempt to recognize the markers in the received image. Figure 3.10(a) shows the raw input image on the client PC, and figure 3.10(b) shows that the server correctly reassembled the L2CAP packets and then successfully recognized the markers in the image.



```
Starting inquiry...

Bluetooth device detected!
Device Name      : P800
Device Address   : 00:0a:d9:17:90:3a

Inquiry Ended

Initializing L2CAP Client...

Starting Discovery...
```

```
--------------------Configuration: Bluetooth Client - Win32 Debug--------------------

Bluetooth Client.exe - 0 error(s), 0 warning(s)
```

```
32      33      0.265003
32      33      0.345001
32      33      0.420002
32      33      0.500000
32      33      0.580002
32      33      0.655003
32      33      0.735001
32      33      0.735001
32      33      0.815002
32      33      0.890003
32      33      0.970001
33      34      0.045002
33      34      0.125000
33      34      0.205002
33      34      0.205002
33      34      0.280003
33      34      0.360001
33      34      0.440002
33      34      0.515003
33      34      0.595001
33      34      0.670002
33      34      0.670002
33      34      0.750000
33      34      0.830002
```

```
--------------------Configuration: Bluetooth Server - Win32 Debug--------------------

Bluetooth Server.exe - 0 error(s), 0 warning(s)
```

(a) Client PC screen shot

(b) Server PC screen shot

Figure 3.10: Desktop screen shots

3.7 Augmented Reality Post-It Messaging System

3.7.1 The need for location messaging

The advent of mobile devices such as mobile phones, PDAs and laptops has allowed people to send and receive messages from anywhere and at anytime. While this has greatly improved communication, it has also led to an overabundance of digital messages, making it difficult to sort through and remember information. These digital messages are delivered instantaneously without regard for the receiver's spatial and temporal contexts. While time-based applications are easy to develop, it is much more difficult to design location-dependent message delivery systems due the following constraints:

- Selecting/Designing an appropriate mobile device for use as an interface for message display
- Selecting/Developing a suitable location tracking system

Other researchers have looked at systems that can provide location-dependent messaging. One such system is the Wearable Remembrance Agent System [47]. It is based on the concept of providing the user with context based information. This system uses five types of contexts, including spatial, to provide cues to a wearable computer user. The wearer of the wearable computer can see one-line summaries of notes-files, old email, papers, and other text information that might be relevant to the user's current context. To display the whole text described in the summary, the wearer hits a quick chord on the chording keyboard.

Starner et al [48] have described a location based messaging system called the *Locust Swarm.* This system is essentially a infrared network of small boards, called Locusts, that are placed in grills beneath the overhead fluoroscent lights. Each Locust consists of a 4MHz PIC 16C84 microcontroller, a RS232 line voltage converter, infrared receiver, infrared LED, 6" × 6" 9V solar cell, and a voltage regulator. The board is approximately 1" × 3". Since this system requires the use of specialized equipment it will not be cost effective to be deployed over city-wide geographical regions.

Another location based messaging system is the *MemoClip* which is a wearable clip that sends location based reminders to users [49]. A user can associate information to be remembered with a description of a location, download it onto the Memoclip, and then get notified when she enters the selected location. For location detection, *LocationBeacons* are installed at the places of interest. The LocationBeacons are programmable and broadcast the location information which can be read by the MemoClip. This system is similar to the Locust Swarm system in that it requires installation of specialized equipment for location detection.

The major drawback of these systems is that they require installation of specialized hardware for the purposes of location tracking. Moreover, users will have to carry a mobile device that they would not carry otherwise. These factors make it unlikely that the system will find widespread acceptance with the common user. Ideally, the mobile device should be available easily in the consumer market and require minimal hardware modification. From a perspective of mobility, it is essential that the device be light enough to be carried easily for long periods of time. A mobile phone fits this role perfectly because it is a portable device which is carried or worn by a user for long periods. In contrast to notebooks and personal digital assistants (PDAs), mobile phones can be carried around without any ergonomic drawbacks since they were designed for exactly that purpose. They are sufficiently small and lightweight enough to be held in one hand or placed in a pocket.

An effective location based messaging system requires a reliable and cost-effective location detection technique. Ideally, the location detection technique should provide tracking over a wide geographical area. GPS technology [2] allows precise location detection, but requires specialized receivers. Moreover it does not work indoors. Cellular based technology provides location for large geographical areas, but current precision level is 200-300 meters which is not precise enough for providing fine-grained location based messaging. Other techniques such as ultrasonic [50] require specialized receivers and infrastructure and are therefore not cost-effective. However, visual tracking is an effective means since it is precise and does not require the installation of any additional infrastructure. The new generation of mobile phones have enough features, such as digital cameras and sufficient processing power, that make them suitable for use as a visual tracking device.

3.7.2 The importance of paper

Although the use of paper-based communication has decreased after the arrival of the internet, its usefulness is evident from the large stack of papers visible in most offices. In the book, *The Myth of the Paperless Office*, Harper et al. [51] mention that in spite of the digital tools available, paper still has the following use:

- As a tool for managing and coordinating action among co-workers in a shared environment.
- As a medium for information gathering and exchange.

- As an artifact for information gathering and exchange.
- As a means of archiving information for groups of co-workers.

A vital, but often overlooked, advantage of paper is that it is easy to stick at different locations and thus enable spatial messaging. For example, in an office, use of Post-It notes as personal reminders, or as messages for colleagues is common. Messages can be posted in the relevant spatial context and thus are easy to remember. For example, reminders to buy something on the way to the office can be posted onto the car's dashboard, reminders to reheat a refrigerated dish can be posted onto the refrigerator. When sending digital reminders, this important spatial context is lost.

But paper has the following drawbacks [51]:

- Paper must be used locally and cannot be remotely accessed.
- Paper occupies physical space and thus requires space for its use and storage.
- Paper requires physical delivery.
- A single paper document can be used only by one person at a time. If more than one person tries to read the same document, it is inconvenient.
- Paper documents can only be used for the display of static visual markings. They cannot display moving images or play sounds.

Moreover, paper messages do not support privacy. A message left on someone's desk can be read by all, unless it has been sealed in an envelope. Even then it is possible for anyone to simply tear off the envelope to read the message.

This has motivated us to come up with a messaging system that combines the advantages of paper-based Post-It notes and the electronic messages. In other words, the messages should be digital to allow remote posting and fast delivery, but they should also be *postable* just as Post-It notes. In order to allow messages to be viewed in a natural way, we have made use of AR technology so that messages can be displayed in their natural environment. A web server has been developed that allows remote users to post messages to a desired user and location. The sender needs to provide the following information when sending the message:

1. The recipient's phone number
2. The location where the message should be received.

The messages are stored on the web server until the intended recipient arrives at the specified location. The messages stored in the web server are also accessible by an AR server. The task of the AR server is to process the messages stored in the web server and generate an AR version of the message. This AR message is displayed to the recipient on her mobile phone when she reaches the specified location. A key advantage of the AR Post It system is that no sophisticated tracking technology is required to determine the location of the mobile user.

Table 3.2 summarizes the features and interactions of the AR Post-It system:

The distinguishing features of the AR Post-It system are:

1. *Location based messaging:* Messages are delivered only in the appropriate

Feature	**Explanation**
Location based messaging	Messages are delivered only in the appropriate location. Recipient uses her mobile phone to view at a marker. The message is displayed as an augmented image on her phone.
Remote Access	Messages can be posted remotely over the internet. Sending the message is similar to sending an email. The sender needs to provide the phone number of the recipient. Messages can also be sent over the mobile phone.
Privacy protection	Unlike paper Post-It notes which can be seen by everyone, the AR Post-It message will be visible only to the person to whom the message has been posted.
Neatness	Since the messages are electronic, the mess of paper is avoided.

Table 3.2: Summary of features of the AR Post-It System.

location.

2. *Remote Access:* Messages can be posted remotely over the internet.

3. *Privacy protection:* Unlike paper Post-It notes which can be seen by everyone, the AR Post-It message will be visible only to the person to whom the message has been posted.

4. *Neatness:* Since the messages are electronic, the mess of paper is avoided.

Figure 3.11 shows a screenshot from the phone. Figure 3.12 illustrates the privacy aspect of the AR Post-It system. Two users are viewing the same marker but they receive different messages on their phone. Thus different senders may post messages to the same location and yet the recipient will be able to view only the message meant for her. This can allows us to have message boards that allow messages to be viewed only by a select group of users.

Boil the milk!

Figure 3.11: Screen shot from the mobile phone showing the AR message display.

Another advantage of the AR Post-It system is user privacy.

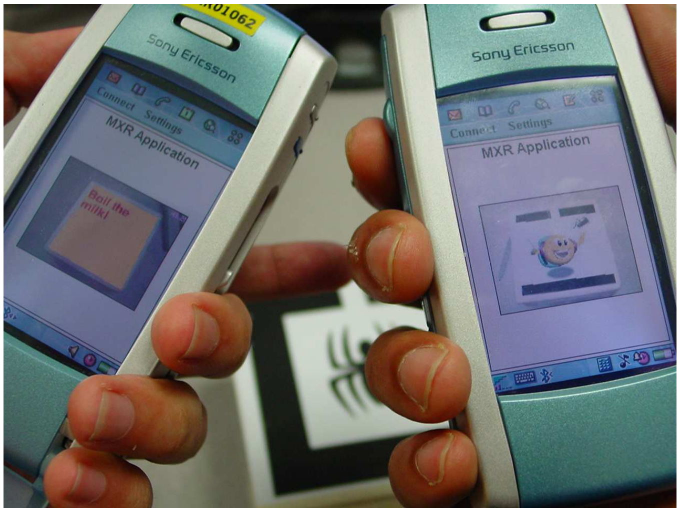


Figure 3.12: Different messages displayed when viewing the same marker. This allows greater privacy than paper based Post-It notes.

3.8 AR Post-It System Description

3.8.1 System Overview

Figure 3.13 outlines how messages are posted and retrieved. The sender can post messages using the phone or a web based application by specifying the recipient and the location where the message is to be posted. Once posted, the message is stored on a server until the specified receiver retrieves the message. There is the additional requirement that the recipient should be at the location specified by the sender, otherwise the message will not be displayed.

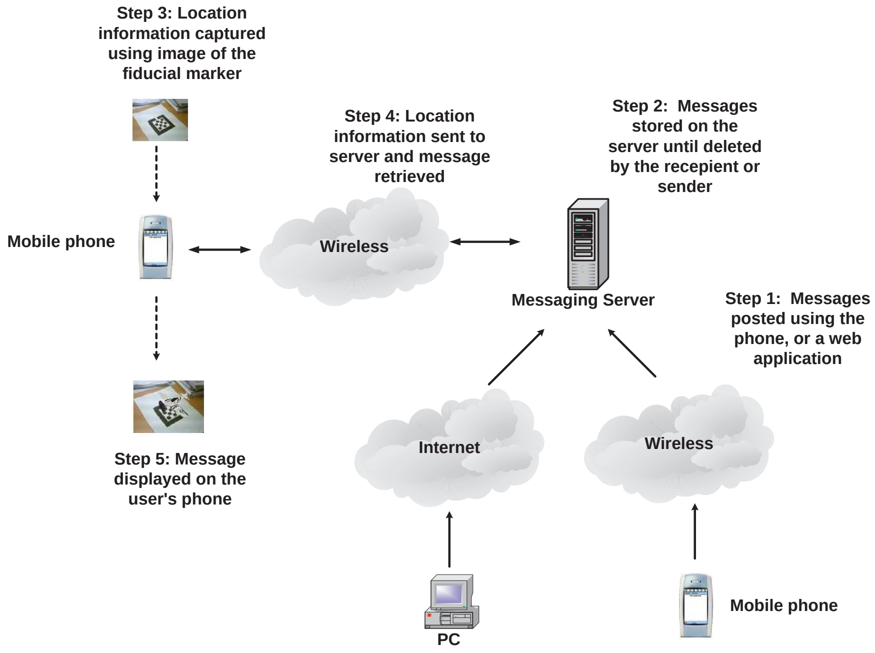


Figure 3.13: Posting and receiving messages.

The AR Post-It system has been developed using the basic AR Phone prototype system. It has the following three modules:

1. Mobile Phone module

2. AR Server module

3. Web Server module

The mobile phone module and the AR Server are based on the basic prototype system. We will therefore not discuss these modules in detail. However, the web server module is an additional module that was developed for the AR Post-It system.

Web Server Module

The web server module allows remote users to post messages to a desired user and location. The web server stores these messages in a database that is also accessible to the AR server module. The relation between the AR Server, the Web server and the database can be thought of as a post-box message delivery system. The web server posts messages to the post box (database), and the AR server retrieves the messages from the post box (database). Figure 3.14 illustrates this concept. Figure 3.15 shows the login page of the AR Post-It Web server. Users first need to register themselves in order to use the system for sending and receiving messages. Figure 3.16 shows the registration page. If a user wishes to post messages only, she does not need to provide her phone number. However, if she wants to receive messages as well, providing phone number is mandatory. This is because when multiple phone users are connected to the AR server, they are distinguished based on their phone numbers. Figure 3.17 shows the page for posting of messages.

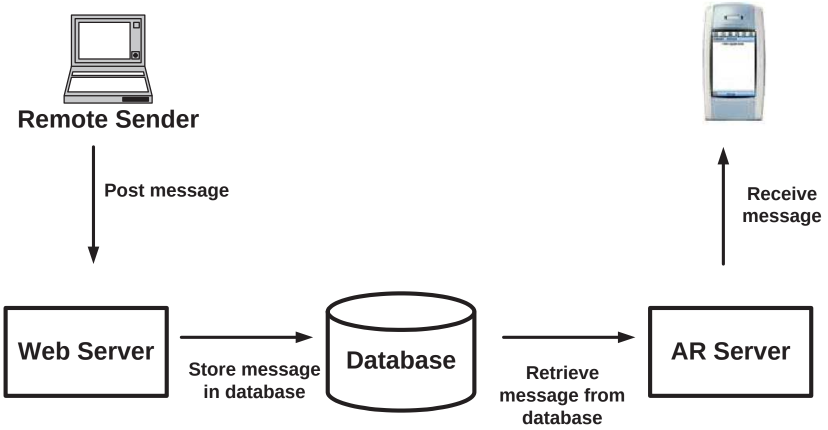


Figure 3.14: Interaction between AR server and Web Server.

Registered Users Login Here
Enter your login name and password to post a message to someone.

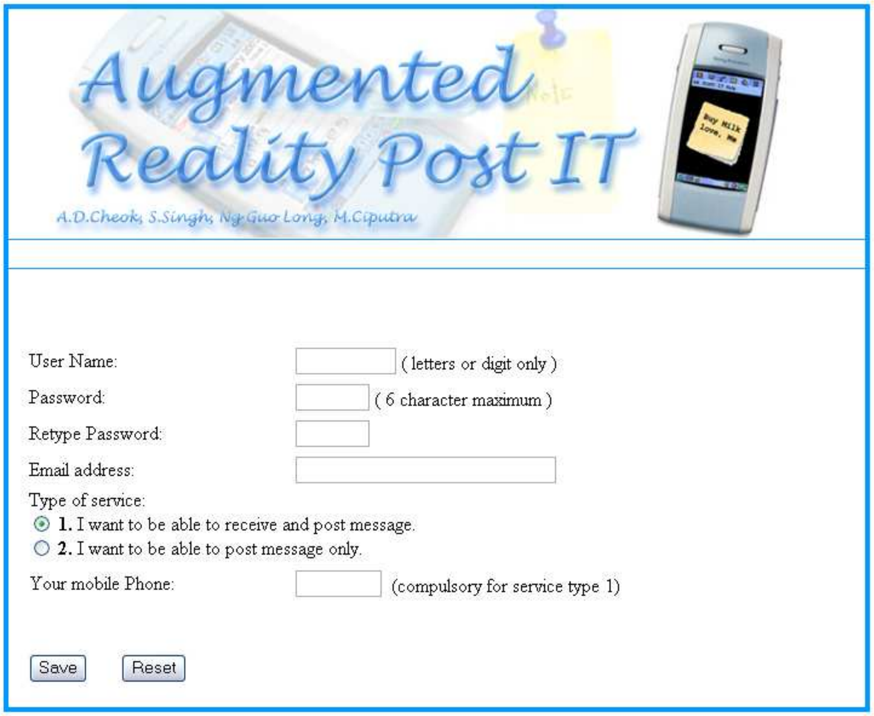
Name
Password:

New to Augmented Reality Post-It?

Get an account with us for a new way of message posting using augmented reality.

- What is Augmented Reality ?
- How does Augmented Reality Post-It work ?
- What are the advantages over SMS and email?

Figure 3.15: Login page of the web server.



User Name: (letters or digit only)

Password: (6 character maximum)

Retype Password:

Email address:

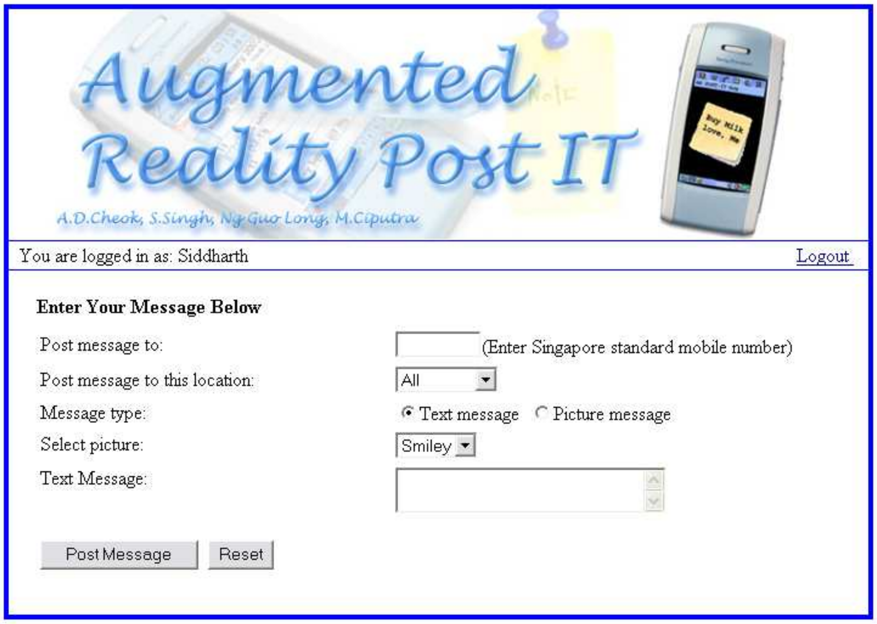
Type of service:

● **1.** I want to be able to receive and post message.

○ **2.** I want to be able to post message only.

Your mobile Phone: (compulsory for service type 1)

Figure 3.16: Registration page.



You are logged in as: Siddharth

Enter Your Message Below

Post message to: (Enter Singapore standard mobile number)

Post message to this location: All

Message type: ● Text message ○ Picture message

Select picture: Smiley

Text Message:

Figure 3.17: Web page for posting messages.

3.8.2 User study

We conducted a user study to compare the AR Post-It system with the digital and paper messaging systems. This user study was conducted separately from the one mentioned in section 3.12 and they both have different aims as well. The user study mentioned in section 3.12 focusses on augmented reality on mobile phones, whereas this user study focusses on the AR Post-It system and response of users towards a location based messaging system.

The Participants

The survey involved 16 people, 3 females and 13 males. Most of them were aged between 22 and 24, with one participant being of 34 years of age.

Survey Approach

The survey was conducted in two phases. In the first phase we asked the users to fill up a questionnaire in order to know which messaging system they normally used under different circumstances. This was done by asking two types of questions:

1. *Behavior based* questions

2. *Scenario based*

In the *behavior based* questions, the users were asked about how frequently they used various messaging systems (paper based and digital based). Some of these questions were:

- How often do you use paper Post-It notes?
- What do you use paper Post-It notes for (e.g., posting self reminder messages, communicating with others) ?
- How often do you use email to communicate with others?
- How often do you use SMS to communicate with others?

In the *scenario based* questions, the users were presented with 4 different scenarios and asked to select the mode of communication they would use under those circumstances. In the first phase, they could only choose from choose from digital (email and SMS) and paper Post-It notes. In the second phase, once they had used the AR Post-It system, they are also given the option of the AR Post-It system. The scenario based questions are listed in table 3.3.

In the second phase, the users were introduced to the concept of location based messaging systems. After this they were shown a demonstration of the AR Post-It system, and allowed to spend 5-10 minutes using the system. Then they were asked to fill up another questionnaire to get feedback about the AR Post-It system. This questionnaire contained two types of questions:

1. Questions about the AR Post-It system *per se*
2. *Scenario based* questions

The questions about the AR Post-It system were related to issues that were about the system itself, e.g., screen clarity, ease of use. In the scenario based questions, we presented the users with the same scenarios that were presented to them in phase 1 of the survey. But this time, we also gave them the option of using the AR Post-It system.

3.8.3 AR Post-It Survey Results

Response to questions asked in phase 1

Table 3.4 lists some of the questions that were asked during phase 1. Figure 3.18 shows the response to these questions.

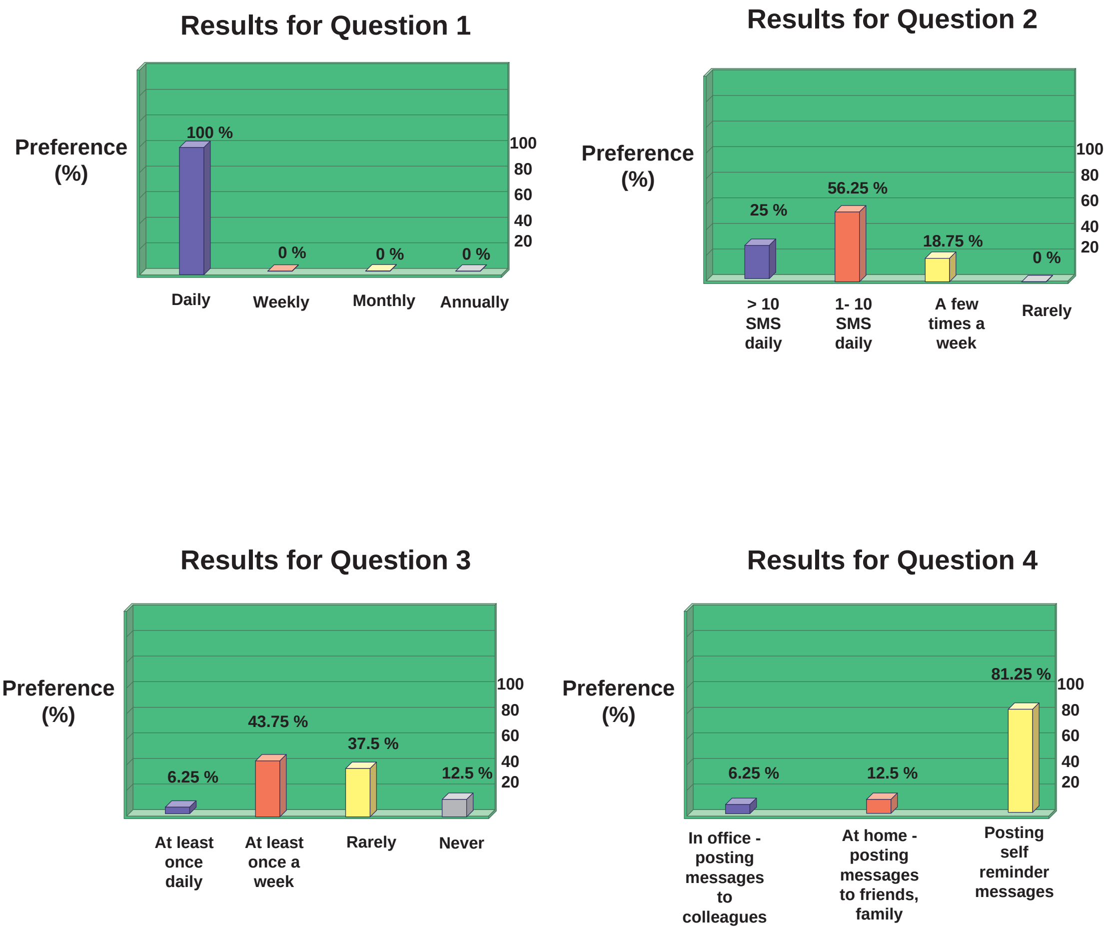


Figure 3.18: Responses to questions asked in phase 1. The questions are listed in table 3.4

Question No.	Question
1	You are in the office at your desk and want to leave a message for a friend who sits in the same room. The person is not currently at his desk. Which messaging system would you prefer?
2	You are in the office at your desk and want to leave a message for a friend who sits in another floor of the same building. Which messaging system would you prefer?
3	You are in the office and want to remind your roommate to reheat the food when he gets home. Which messaging system would you prefer?
4	You are at home and want to send a short personal message to your friend who is in his office. Which messaging system would you prefer?

Table 3.3: Scenario based questions

Question No.	Question
1	How often do you use email?
2	How often do you use SMS?
3	How often do you use paper Post-It notes?
4	For which purpose do you mainly use paper Post-It notes?

Table 3.4: Some of the questions asked during phase 1 of the survey

All the volunteers extensively used digital (email and SMS) messaging systems whereas paper Post-It notes were not used as regularly. The main use of paper Post-It notes was for posting reminder messages for oneself.

The response to the scenario based questions is shown in figure 3.20. Here the users had not been presented with the option of AR Post-It system. This was because we wanted to know how the behavior of users would change once they had the option of the AR Post-It system.

Response to questions asked in phase 2

In phase 2, the users were shown the AR Post-It system and allowed to use it. We asked some questions related to the functioning of the AR Post-It system. The questions have been listed in table 3.5 and the corresponding responses are listed in figure 3.19.

Most of the users found the AR Post-It to be a useful addition to the traditional electronic messaging systems. They also acknowledged the privacy feature of the AR Post-It system. Users also appreciated the fact that the AR Post-It system did not force them to carry any additional mobile device. When the users were specifically asked whether they would be willing to carry any additional device for enabling location based messaging, almost all the users responded in the negative. One user responded:

> “No, unless it is as small as a credit card.”

Thus, the fact the AR Post-It system did not use any additional device for

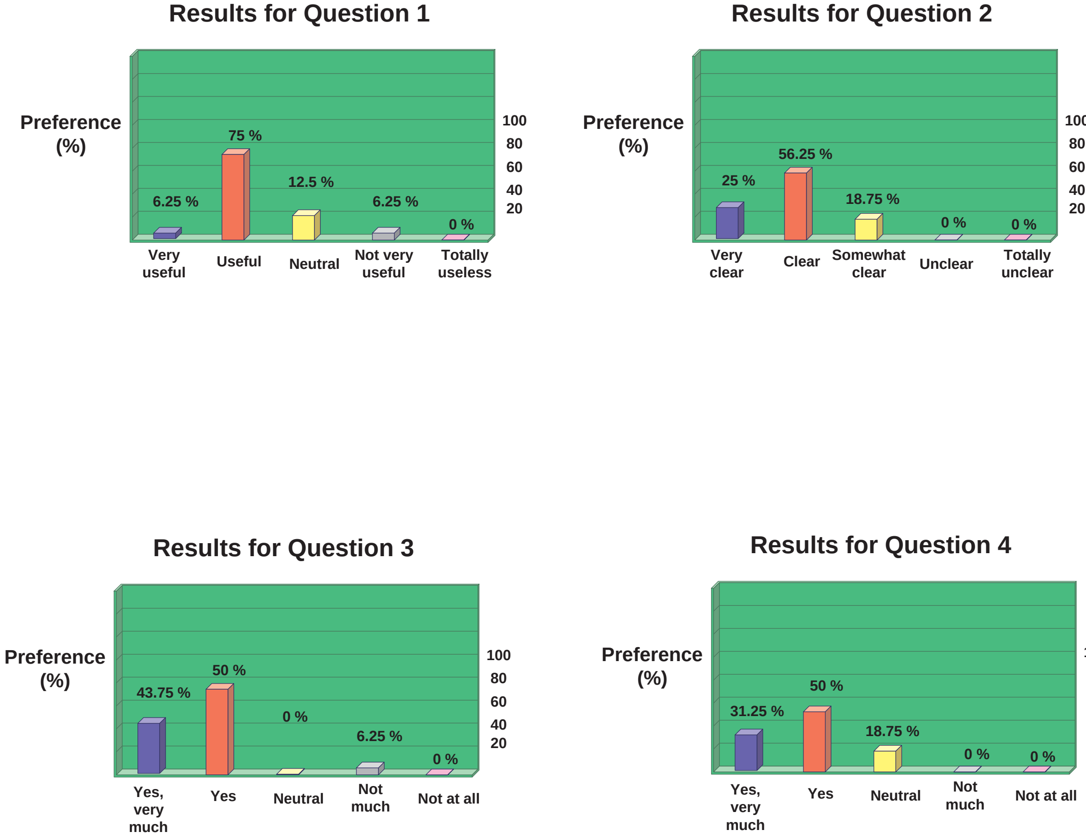


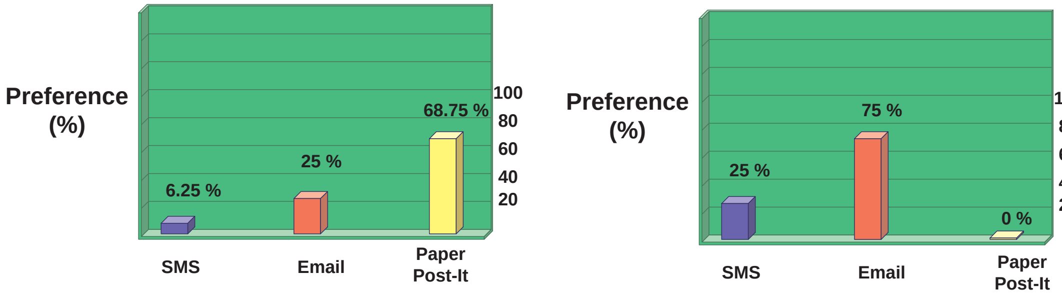
Figure 3.19: Responses to questions asked in phase 2. The questions are listed in table 3.5

tracking or displaying the messages was viewed positively by the users.

Response to *scenario based* questions

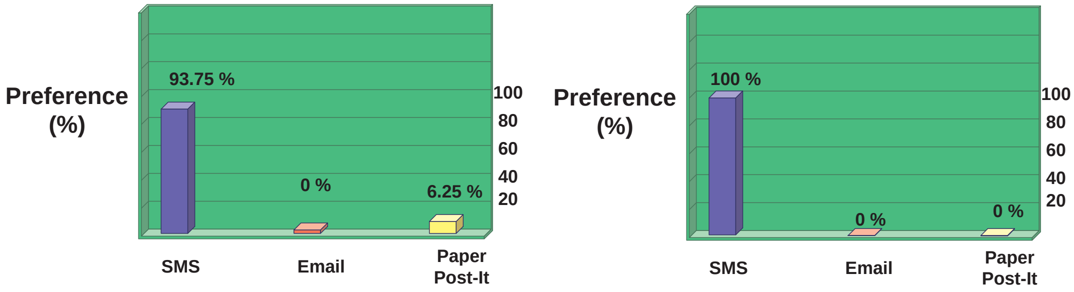
Figures 3.20 and 3.21 show the response to the *scenario based* questions. Figure 3.20 shows the response during the first phase of the survey. Figure 3.21 shows the response when the AR Post-It was also included as an option.

When the message had to be sent to someone in the same room, or in the same building (scenarios 1 and 2 in figure 3.21), AR Post-It was the preferred messaging system. Although some users also pointed out that scribbling on paper was quicker than using AR Post-It, many of them preferred the AR Post-It due to better privacy provided by it. SMS was preferred over the AR Post-It whenever the message was a personal message and not a reminder message (scenario 4 in figure 3.21).



Scenario 1: Sending messages within the same room

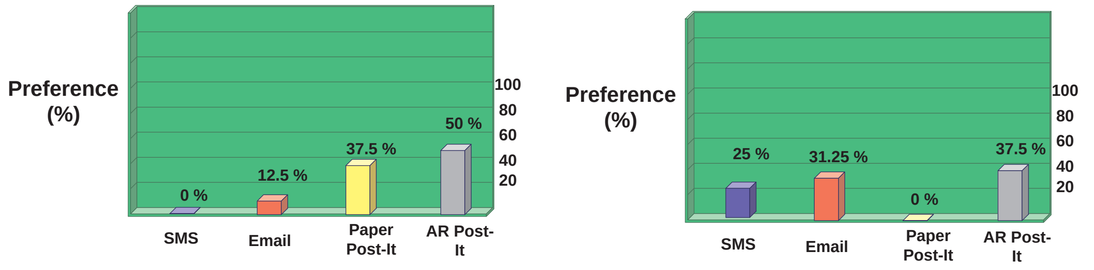
Scenario 2: Sending messages within the same building



Scenario 3: Sending reminder message to remote location

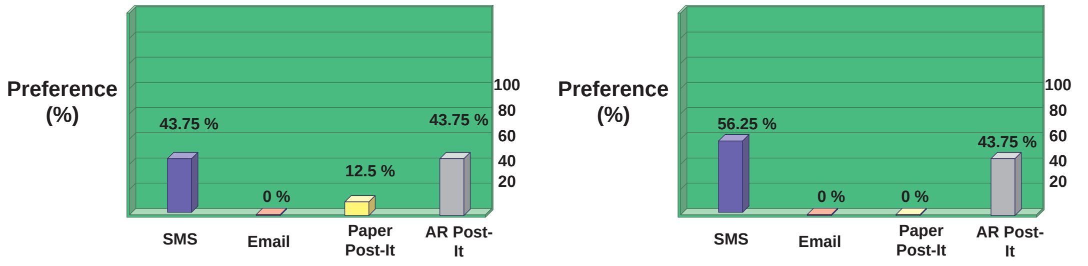
Scenario 4: Sending personal message to remote location

Figure 3.20: Responses to *scenario based* questions during phase 1 of the survey. Users were not presented with the option of AR Post-It. The exact questions are listed in table 3.3



Scenario 1: Sending messages within the same room

Scenario 2: Sending messages within the same building



Scenario 3: Sending reminder message to remote location

Scenario 4: Sending personal message to remote location

Figure 3.21: Responses to *scenario based* questions during phase 2 of the survey. Users had been shown the AR Post-It system, and were also given the option of choosing it. The exact questions are listed in table 3.3

QUESTION NO.	QUESTION
1	How useful is the AR Post-It system as a location based messaging system?
2	How clear was the message display?
3	Compared to paper Post-It notes, does the AR Post-It system provide better privacy?
4	Do you feel the AR Post-It system is a useful addition to traditional electronic messaging systems (SMS/email)?

Table 3.5: Some of the questions asked during phase 2 of the survey

3.9 AR Catalogue System

The AR Catalogue system aims to enhance the commonly used catalogues by allowing the users to view relevant 3D information on their phones. The AR Catalogue system was developed using our basic prototype of the AR phone system. Thus the component modules are the mobile phone module and the AR server module.

The distinguishing features of the AR Catalogue system are:

1. *Full 3D display:* Users can view pictures in full 3D. Thus additional information can be obtained.
2. *Mobility:* Since the AR Catalogue system uses lightweight mobile phones for display, mobility of users is not hindered.
3. *Multiple character display:* Multiple 3D characters can be displayed at the same time. This enhances the utility of the system.

The system architecture of the AR Catalogue system is displayed in figure 3.22.

We now present user pictures of the AR Catalogue system. As mentioned earlier, the system allows posting of 3D characters. Figure 3.23(a) shows a 3D image of Kerropi the frog being displayed on the phone. Figure 3.23(b) and 3.23(b) show the actual screen shots captured from the Sony Ericsson P800 mobile phone.

The AR Catalogue system allows multiple markers to be recognized at the same time. This allows great flexibility to the user because she does not have to bother

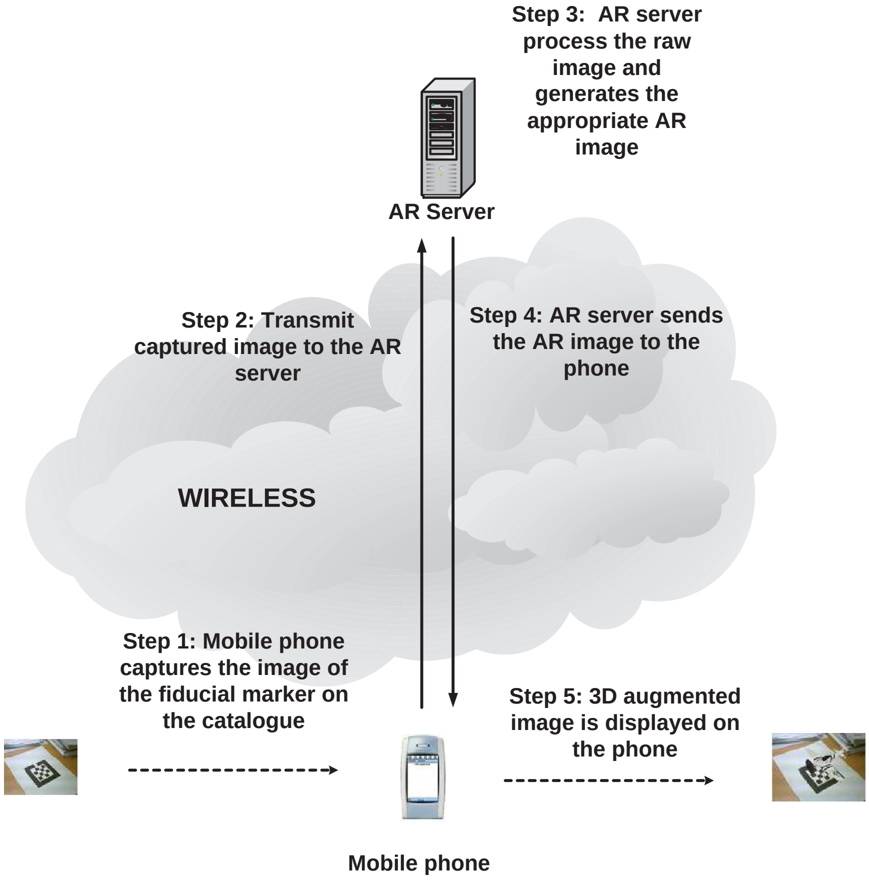
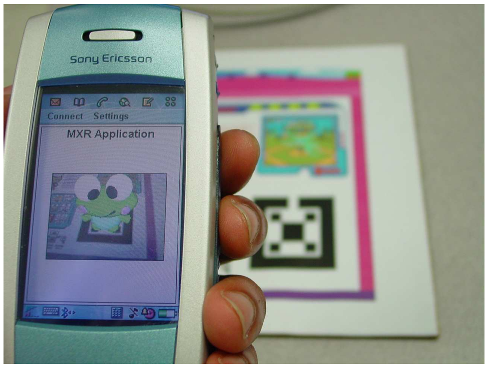


Figure 3.22: AR Catalogue system architecture.

about how to position the phone camera. It also allows a user to view multiple messages together. Figure 3.24 illustrates that multiple markers can be recognized at the same time. The image on the left shows the orientation of the markers. The image on the right shows that the mobile phone displays 3 different virtual objects on the 3 markers.

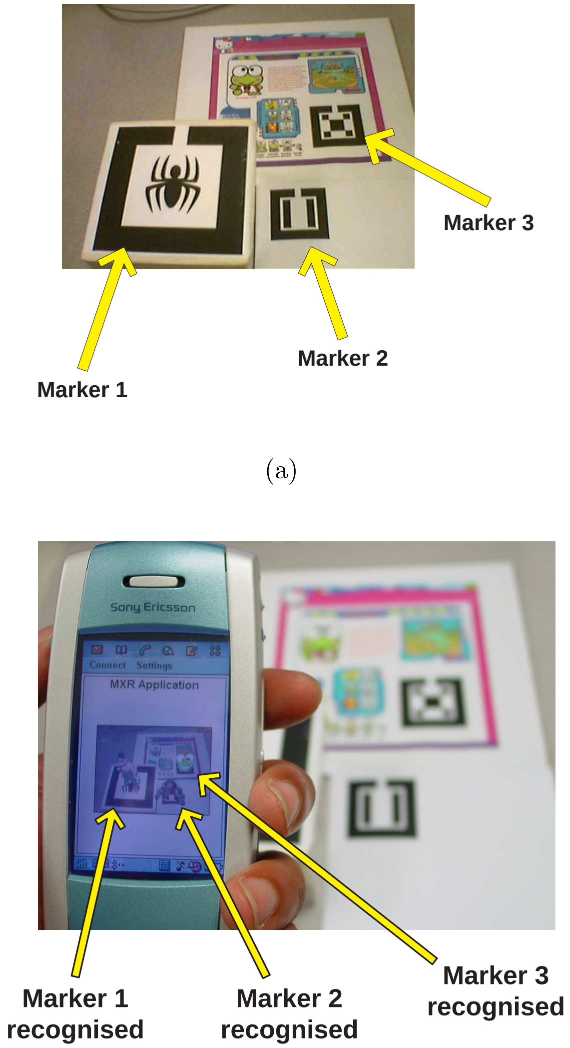


(a) 3D AR image being displayed on the phone.

(b) Screen shot from the mobile phone

(c) Screen shot from the mobile phone

Figure 3.23: Pictures and screenshots of the 3D Catalogue system



(a)

(b)

Figure 3.24: Multiple markers can be recognized at the same time.

3.10 Results

In general, the AR phone system had a high degree of robustness. It was able to deliver accurate marker tracking and pattern recognition consistently.

3.10.1 Performance Issues

The mobile phone was able to display an augmented image fairly quickly after the capture of the original image. Depending on the chosen input resolution, the round trip time varies from 2 to 7 seconds. In mathematical terms, we can term the total time delay

$$T_d = t_{capture} + t_{transfer} \tag{3.1}$$

where,

$t_{capture}$ = time taken by the mobile phone camera to capture a raw image

$t_{transfer}$ = round trip image transfer time between the mobile phone and the AR server

It was found that the biggest bottleneck lies with the image transfer time (see Table 3.6) with larger files leading to higher transfer times.

3.10.2 Fiducial Marker Tracking and Pattern Recognition

The success rate of marker tracking and pattern recognition is dependent on the resolution of the image, the size of the fiducial marker and the distance between the mobile phone and the fiducial marker. It was found that image capture at

Resolution	Image Capture Time (s)	JPEG Size (KBytes)	File Transfer Time (s)	Total Time (s)
$640 \times 480 \times 12$ bit	1.5 - 2.0	12 - 27	4.0 - 5.0	5.5 - 7.0
$320 \times 240 \times 12$ bit	1.0 - 1.5	6 - 10	2.0 - 3.0	3.0 - 4.5
$160 \times 120 \times 12$ bit	0.5 - 1.0	2 - 3	1.5 - 2.0	2.0 - 3.0

Table 3.6: Capture time, JPEG size, file transfer time and total time delay for images at three different resolutions.

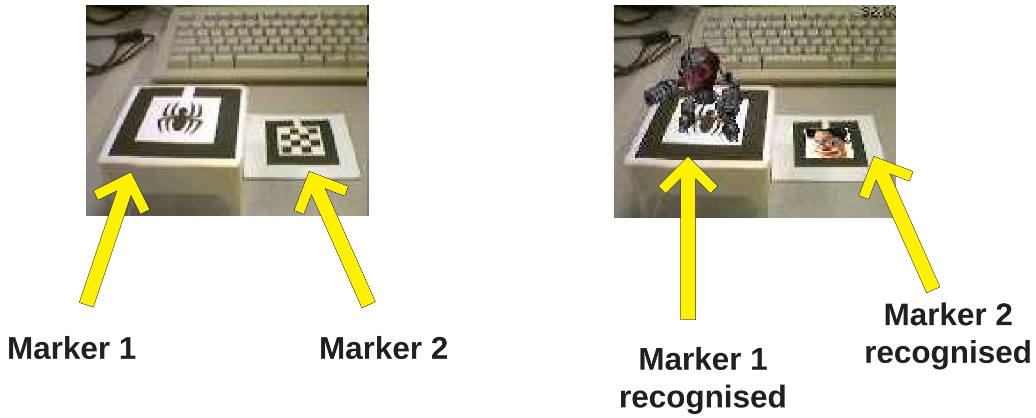


Figure 3.25: Tracking and differentiating the pattern on the markers at 320 × 240 resolution. The image on the left shows the image captured by the P800. The image on the right shows the final rendered image displayed on the P800.

320 × 240 was adequate for the purposes of marker recognition. Figure 3.25 shows multiple marker recognition at 320 × 240. Table 3.7 shows the marker tracking results that were obtained by doing pattern recognition on two fiducial markers of size 10cm × 10cm.

Resolution	Distance from marker (m)	Marker Tracking	Pattern Recognition
640 × 480 × 12 bit	0.50	ok	ok
	1.00	ok	ok
	1.50	ok	ok
	2.00	ok	ok
	2.25	ok	failed
	2.50	failed	failed
320 × 240 × 12 bit	0.50	ok	ok
	1.00	ok	ok
	1.50	ok	ok
	1.75	ok	failed
	2.00	ok	failed
	2.25	ok	failed
	2.50	failed	failed
160 × 120 × 12 bit	0.50	ok	ok
	0.75	ok	ok
	1.00	ok	failed
	1.25	failed	failed

Table 3.7: Marker tracking and pattern recognition of images captured at different resolution and distance.

3.11 Discussion

One important consideration when implementing this module is the varying light conditions at different locations. When capturing images under low-light conditions, it is possible that the tracking algorithm in the AR server module might not work effectively. Another consideration is the resolution of the captured image. Images with higher resolution will undoubtedly give better tracking and pattern recognition results (see Table 3.7). However, such image has bigger file size, and it take a long time to transfer this file to the AR server module (see Table 3.6). It is a trade off between better tracking and pattern recognition and file transfer speed.

Considering these two factors, it is important that users have the flexibility to change the brightness, contrast and image resolution under different conditions to obtain optimum results. Pull-down menus with options to change these parameters have been created for this purpose.

There is also some complication in sending the image to the PC server as the Symbian UIQ SDK can only saves the data obtained from the camera to a CFbsBitmap format which is internal only to Symbian OS. Hence, if raw data captured from the camera is sent to the AR server module, the server will not be able to access the image data. To overcome this, the data in CFbsBitmap format is first converted to a general format like bitmap or JPEG before sending it to the server. JPEG is the preferred choice simply because it is a more compressed format which can significantly save bandwidth when transferring to the AR server module.

The performance of the AR phone system can be improved by reducing the image transfer between the phone and the AR server. Although mobile phones are not designed for CPU intensive tasks it is still possible to implement basic image processing tasks such as the thresholding of images. This would mean far less data would need to be transmitted over the wireless channel and should increase performance. We wanted to implement a simple thresholding, but could not load our graphics tool kit, the MXRToolkit [45], onto the phone because of the existing proprietary operating system(Symbian OS version 7.0). Moreover, there is no 3D graphics library present on the P800 mobile phone. Although there have been prior attempts to create libraries similar to OpenGL [52] such as PocketGL [53] and TinyGL [54], these projects are either restricted to games or incomplete and discontinued. In the future, as multimedia content on phone becomes more popular, we should expect the development of 3D graphics libraries for mobile phones.

Bluetooth has a theoretical maximum data rate of 723kbps while the GPRS wireless network has a maximum of only 171.2kbps. However, the user will not experience anything close to this because this data rate assumes no error correction. Hence, it is not desirable to run AR services over both Bluetooth and GPRS wireless network. 3G systems offer a lot of potential in terms of bandwidth. Although the maximum user data rate offered in year 2002 is 384kbps, it is expected to increase to about 2 Mbps in near future [55]. In addition, HSPDA will offer data speeds up to 8-10 Mbps (and 20 Mbps for MIMO systems). Wireless network standards like 3G systems, should lead to a great performance improvement in our prototype application in future. However, such systems are still largely unavailable to end users currently.

3.12 User Study

A preliminary user study was conducted to gauge user interest in AR applications on mobile phones. The survey was conducted with the following aims in mind:

1. To gauge the first reaction of users towards mobile phone based AR applications.
2. To know which genre of AR applications would interest mobile phone users the most.

3.12.1 AR Applications used for the Survey

We used two applications for the survey:

- *AR Post-It*
- *AR Catalogue*

3.12.2 Survey Approach

The survey was divided into three phases:

The first phase was the *introduction* phase. Volunteers were first given a short introduction about AR in general. Then, the concept of the AR Post-It and AR-Catalogue applications were explained.

The second phase was the *user trial* phase. Volunteers spent about 10 minutes using the AR Post-It and the AR-Catalogue applications. Assistance was given when necessary.

The third phase was the *feedback* phase in which we sought responses from the users. For this, the volunteers were asked to fill up a survey form. The questions in the survey form were of two types:

1. *Close-ended questions:* These questions were multiple-choice questions that aimed to categorize participant's responses. The aim of these questions was to find the spontaneous responses of the users.
2. *Open-ended questions:* These questions aimed to find the participants' opinions about their first encounter with AR on mobile phones. The aim was to

avoid bias that could have resulted from suggesting responses to users and allow users to give more expansive answers. One example of such open-ended questions is:

> Please tell about the problems faced when using the system.

After the feedback form was filled, a short discussion was carried out with the users to assess their enthusiasm and also to get feedback on any issue that might have missed in the questionnaire.

3.12.3 The Participants

In total 32 responses were collected, 11 female and 21 male. The participants were aged between 20 to 33 years with most of them between 22 and 24.

All the participants were computer users. Six of them rated themselves as “beginners” while the others rated themselves as “advanced” or “intermediate”. Twenty-eight of them owned mobile phones. Half the participants had no prior experience with any kind AR applications.

QUESTION NO.	QUESTION	*Reason for asking*
i	How do you feel, in terms of level of excitement, about the concept of mobile phone based Augmented Reality (AR) applications?	*To investigate the excitement level with regard to AR applications on mobile phones*
ii	How comfortable did you feel while using the AR system?	*To find out ease of use of a mobile phone as an AR device.*
iii	How did you find the screen display, in terms of clarity?	*To investigate whether screen display on a mobile phone is good enough for AR applications*
iv	Would you prefer using a phone with a larger screen display size?	*To investigate whether screen display on a mobile phone is good enough for AR applications*
v	How do you find mobile phone based AR applications as compared to AR on other devices such as PCs, wearable computers, and PDAs?	*To investigate the suitability of mobile phones as an AR interface as compared to other devices such as PCs, wearable computers, and PDAs*
vi	Which type of AR applications would you like to have on your mobile phone?	*To find which kind of AR applications would be popular with users*
vii	How useful were these two applications in demonstrating the possible uses of mobile phone based AR?	*Whether our demos were useful to the laymen in helping them understand and appreciate AR*
viii	Having used these two applications, do you look forward to other AR applications on mobile phones?	*Whether our demos generated interest in the participants*

Table 3.8: Some of the questions from the user study

3.12.4 Questions

Table 3.8 lists some of the questions that were asked in the user study. The results to these questions are shown in Figure 3.26 and Figure 3.12.4.

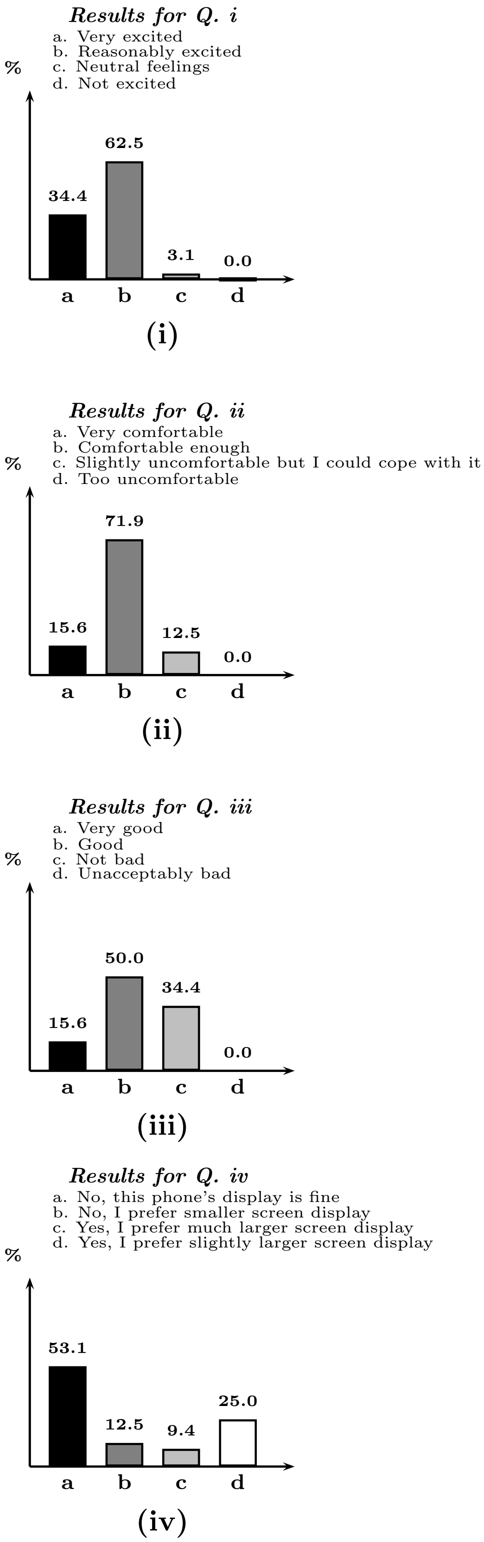
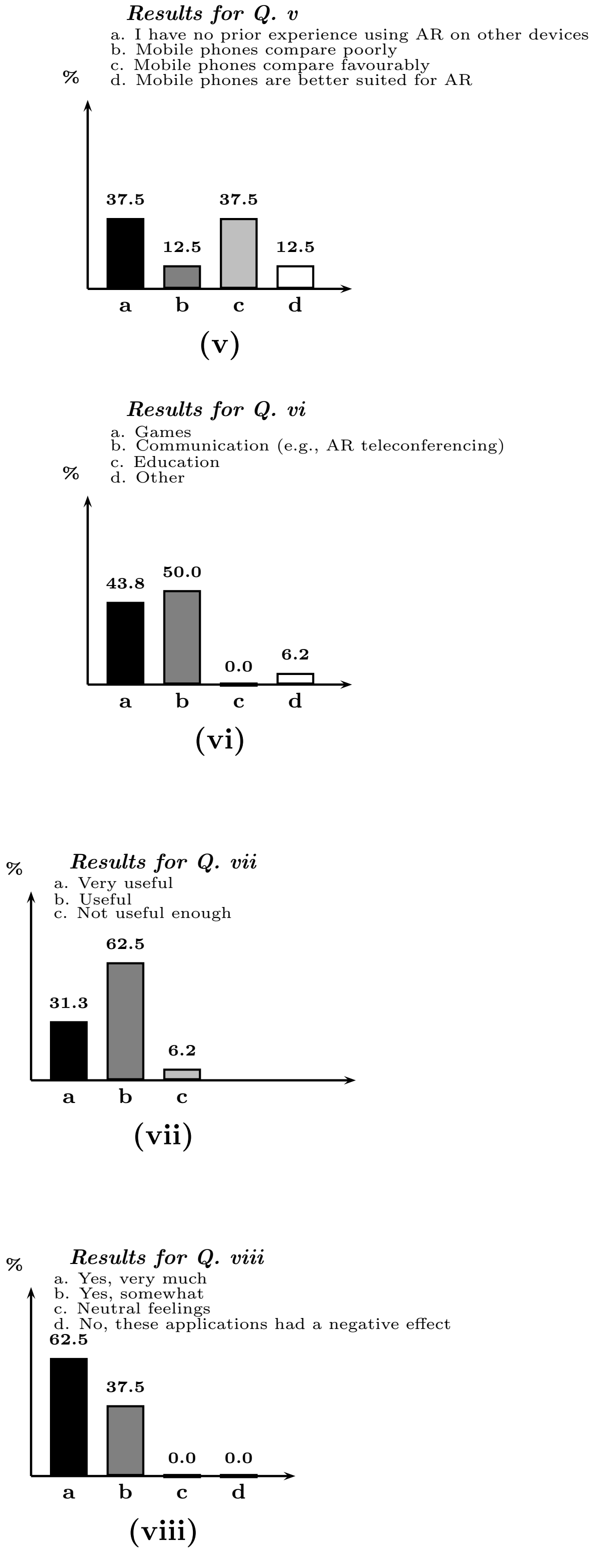


Figure 3.26: Results for questions i to iv.



(v) (vi) (vii) (viii)

Figure 3.27: Results for questions v to viii.

3.12.5 Findings

The primary advantage of deploying AR on mobile phones is that mobile phones have a high penetration rate. More people own mobile phones than PDAs. In our survey group, 28 users owned a mobile phone while only 6 owned a PDA. While these mobile phones were not the high end mobile phones that could support AR applications, these numbers show how widespread the use of mobile phones has become. Also, mobile phone users tend to migrate to higher end phones either by selling/discarding their old phones or by using the exchange offers provided by various phone companies. This means that in the near future, it is reasonable to expect that there will be a significant increase in the number of users which have AR capable mobile phones.

When asked about which AR applications would the users like to have on their mobile phones, most users indicated either games or communication applications (Figure 3.12.4, question vi). Users were asked if the applications we had developed were useful enough (as a technology demonstrator). Most users rated our system either "very useful" or "useful" (Figure 3.12.4, question vii).

Suitability of mobile phones for AR

Almost all the users had a positive expectation about the concept of AR on mobile phone. Only 1 user reported having neutral feelings(Figure 3.26, question i).

The major drawback of AR on mobile phone would be the small form factor provided by the limited screen size. The P800 has a screen size of 320 x 208 pixels. However, from the responses, it was found that the small screen size was not perceived as a hinderance (Figure 3.26, question iv). Only 3 users said they wanted a much larger screen display. 17 of them were comfortable with the P800's screen size while 8 of them wanted a slightly larger screen size. Curiously, 4 of them actually preferred a smaller display.

The reason users found the screen size to be acceptable was probably due the fact mobile phones are used frequently for sending text messages. All the users had indicated that other than voice communication, the most common use of mobile phones was for sending text messages. Mobile phones also doubled up as schedulers and organizers. Thus the users have long adapted to the small screen size of mobile phones.

Screen clarity was another issue that would be an important factor in gauging the suitability of mobile phones for AR. However, none of the users found the clarity to be unacceptably bad. 21 users rated the screen clarity from "very good" to "good". 11 rated it to be "not bad" (Figure 3.26, question iii).

Effect of previous experience of AR applications

Participants' response to the mobile phone based AR system could have been affected by their prior exposure to AR systems on other devices. We wanted to study whether users having prior exposure to AR reacted differently than users having no previous exposure to AR systems.

For the purpose of analysis, all users were separated into two groups -

Group	Total Users	*Present display is fine*	*Prefer smaller display*	*Prefer much larger display*	*Prefer slightly larger display*
A	16	11	1	0	4
B	16	7	2	3	4

Table 3.9: Perception of screen size in different groups

Group	Total Users	*Very good*	*Good*	*Not bad*	*Unacceptably bad*
A	16	1	11	4	0
B	16	3	6	7	0

Table 3.10: Perception of screen clarity in different groups

1. Group A: Users of this group had prior experience of AR applications.
2. Group B: Users of this group had *no* prior experience of AR applications.

Both group A and group B had 16 users each.

With regard to the perception of the screen size, responses in group A were mostly in one category. Responses in group B had more variation (see table 3.9).

With regard to the perception of screen clarity, users in different groups gave somewhat different responses (see table 3.10). Users in group B gave an average rating. It seems users who had previous AR experience were more forgiving towards mobile phone based AR applications.

AR on mobile phone vs AR on other devices

The survey also aims to find out how users compare the mobile phone based AR with AR applications that run on other devices. Users in group A consisted of 16 participants who had some prior experience of AR applications. The other devices on which they had viewed AR applications were PCs and/or wearable computers.

Among the 16 users who had prior experience of AR applications, 14 said that mobile phones compared favorably (or better) as AR interfaces (Figure 3.12.4, question v). In fact 3 users said that mobile phones were better suited for AR because they were easy to carry. Only 2 in this group said that mobile phones compared poorly to other devices. The reason cited by them was the slow image capturing speed of the P800.

Most of the users did not cite the slow speed of mobile phones as a drawback. Looking at the background of the users in this group, it was found that 13 of them had used HMD based AR applications. When asked about using HMD based AR systems, these users said they didn't like using such systems. So one important reason why these users rated mobile phones favourably could be that the mobile phone system was HMD-less.

3.13 Conclusion

In this chapter we presented a mobile phone based augmented reality system. We developed a prototype of the system that consisted of a mobile phone module and an AR server module. The prototype system was then used to develop two AR applications for mobile phones - the *AR Post-It* and the *AR Catalogue* systems.

The AR Post-It was a location based messaging system that made use of augmented reality to enhance the messages as well as to provide location detection capability. AR Post-It has the advantages of digital messaging (remote access, fast delivery) and paper messaging (sticking messages on objects).

The distinguishing features of the AR Post-It system are:

1. *Location based messaging:* Messages are delivered only in the appropriate location.
2. *Remote Access:* Messages can be posted remotely over the internet.
3. *Privacy protection:* Unlike paper Post-It notes which can be seen by everyone, the AR Post-It message will be visible only to the person to whom the message has been posted.
4. *Neatness:* Since the messages are electronic, the mess of paper is avoided.

The AR Catalogue system allowed mobile phone users to view 3D augmented reality pictures on their mobile phones. This was developed with the primary purpose of enhancing the paper based catalogues. The distinguishing features of the AR Catalogue system are:

1. *Full 3D display:* Users can view pictures in full 3D. Thus additional information can be obtained.
2. *Mobility:* Since the AR Catalogue system uses lightweight mobile phones for display, mobility of users is not hindered.
3. *Multiple character display:* Multiple 3D characters can be displayed at the same time. This enhances the utility of the system.

We also discussed the issues involved in building a mobile phone AR system. Our experiments show that image capturing at 320×240 resolution leads to optimal performance of the AR phone system. Images captured at the lower resolution of 160×120 reduce the success rate of marker recognition while images captured at 640×480 considerably increase the transfer time between the phone and the AR server. Using the core technology of the AR phone system, we designed two simple AR applications for the mobile phone - AR Post-It and AR Catalogue. These AR applications were then used in order to obtain user feedback about AR on mobile phones. From the user survey we found that

1. The concept of AR applications on mobile phones generated a lot of enthusiasm. This shows that there is a potential for having successful AR applications. In other words AR on mobile phones can go beyond being just a technology demonstrator.
2. Most users are very comfortable using mobile phones for AR applications.
3. The small screen size of mobile phones is not a hinderance to the success of AR applications on mobile phones.

Hence we could conclude that mobile phones are suitable as an AR interface.

We consider the outcomes to be important because they represent one of the first 3D Augmented Reality application that is immediately useful and runs on socially acceptable, off-the-shelf mobile phone. The business and games market pushing video conferencing and mobile recreational activities will fuel the demand for faster and more capable mobile phones. We can reasonably expect mobile phones with built-in 3D acceleration to become available soon. This trend is manifested in the appearance of products with built-in wireless connectivity and cameras which will allow us to use completely unmodified off-the-shelf devices for Augmented Reality applications. Therefore, mobile phones can be viewed as an important tool for future manifestation AR applications.

Chapter 4

Anywhere, Any-device Gaming

4.1 Chapter Overview

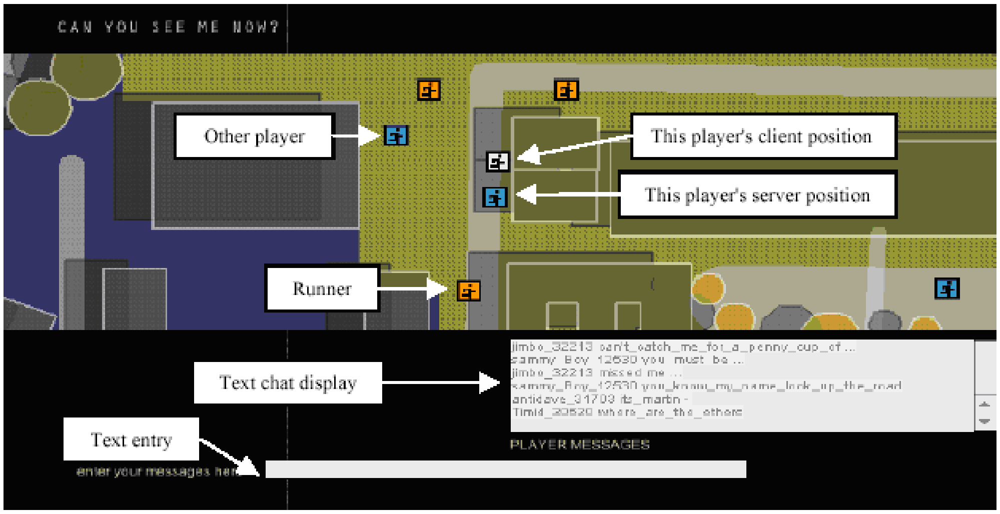
In this chapter we describe a multi-player networked Pong game that can be played in real-time on mobile devices and stationary PCs. Through this game we intend to explore the realm of the next generation of games which, we believe, will incorporate elements of mobility, multiple connectivity methods and playability on different types of devices. The system comprises the Sony-Ericsson P800 phone, a central game server, and personal computers. The client-server model has been used so that all interaction is done through the game server. The server can host multiple games at the same time. Each game has two players and a number of spectators who can view the game. The spectators can interact with each other using text messages. The Pong game is not the end to itself. This work is the core of an ongoing research project on multiple platform, multiple player mobile entertainment.

4.2 Background and Motivation

Video games have undergone a lot of change in their architecture. The early video games could be played only on stand-alone terminals. Most of these games were single player games. The games that were multi-player required the players to connect to the same terminal and sit close together. With the arrival of the internet age, multi-player games evolved to the next level. Players could now sit far away, each connected to his own PC. Examples of such games are Quake2 [56], Unreal Tournament [57]. But the individual player still had to remain seated at a fixed place. The advent of powerful mobile devices has removed this restriction. One such example is the recently launched Nokia N-Gage [58]. These mobile devices are equipped with enough processing power enabling them to run even traditional PC based games. For example, Nokia's N-Gage supports the hugely popular PC game *Lara Croft Tomb Raider* [59].

Anywhere Gaming System uses mobile phones as part of the system where game players can interact with one another. There are some previous works done using other platforms in entertainment. One of them came from the University of Nottingham [60]. "Can You See Me Now" (CYSMN) and Bystander were created. This research is to explore the experience in which mobile user can collaborate with online user with the intention to establish a relationship between human who are operating in different context.

These games allow online participants collaborate with mobile participants on the city streets. It utilizes personal computers, PDAs and GPS devices to allow players to engage in city wide mixed reality games. Figure 4.1. shows the online player interface from CYSMN.



jimbo_32213 can't_catch_me_for_a_penny_cup_of ...
sammy Boy 12530 you must be ...
jimbo_32213 missed me ...
sammy_Boy_12530 you_know_my_name_look_up_the_road
antidave_31703 its_martin -
Timid_22520 where_are_the_others

Figure 4.1: The online player interface from CYSMN

Mobile player (runner) would carry a PDA to register their movement around the city and sent back to the server. This position can be seen as the Runner in Figure 1. Basically, the runners are to find the stationary players. This research utilized lots of hardware which are required to be equipped by the runner. This is shown in the figure 4.2. Anywhere Gaming System requires only mobile handset as the mobile user which has a larger pool of potential users to interact with.

Another important mobile game is known as ARQuake[61]. It is an AR extension of the popular computer game Quake. Using wearable computer equipped with global positioning system, ARQuake can be played indoor and outdoor. However it is a single player game with practically no social interaction.

Anywhere Gaming system also has aspects derived from pioneering work that

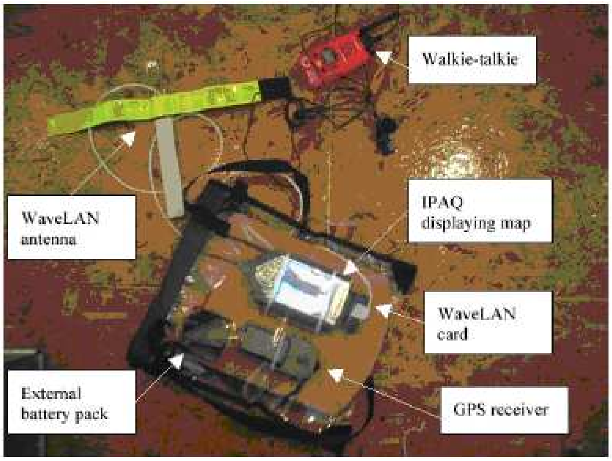


Figure 4.2: Runners' equipment

has been developed on ubiquitous gaming. Multi-player mobile gaming is demonstrated in *Pirates!* [62]. *Pirates!* implements the game on PDAs with proximity sensing technology to incorporate a players contextual information (such as physical co-location of players and objects in the world) into the game context as important elements of the game mechanics. However it lacks multi-playability.

However, cross-device games have only recently been explored. By cross-device games we mean games that involve stationary (computers, game consoles, etc.) as well as mobile devices and allow real-time data exchange between them. Sony Corporation released the PocketStation for its highly popular game console Sony Playstation [63] in 1999. PocketStation allows users to save their game data from the PlayStation and play the game on the portable PocketStation. These devices enable players to continue the same game on two different devices, and thus are a step forward towards anywhere gaming. However they are still not true anywhere gaming devices because they don't allow simultaneous play of a game among multiple players on different devices.

4.3 Chapter Organization

In this chapter we describe a cross-device Pong game. This game is the first stage of work in our ongoing research on novel multi-platform, multi-device *Anywhere Gaming* system. The rest of the chapter is organized as follows: section 4.4 briefly describes the concept of the *Anywhere Gaming* system. Section 4.5 describes the Pong game we have implemented. Section 4.6 describes the architectural components of the Pong game system. Finally, we present conclusion and future work in section 4.7.

4.4 Anywhere Gaming System

The user base of PC based online games is very large and the future potential of mobile gaming is huge. However at present, both these gaming worlds are relatively separate, and there is no seamless link from the PC based gaming environment to the wireless mobile environment.

We are developing a new system, *Anywhere Gaming*, which will enable players to access the broadband gaming environment on the internet even when they are moving. Players will be able to play the same game using multiple devices such as PCs, mobile phones, PDAs, and laptops. This will allow the player to access the game from anywhere at anytime.

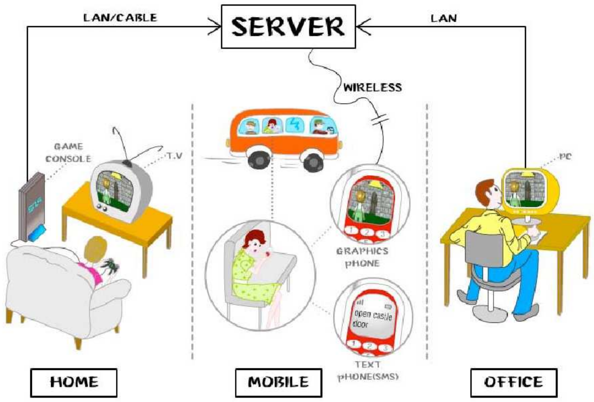


Figure 4.3: *Anywhere Gaming* system

A typical scenario is shown in figure 4.3. It shows three players playing a multi-player game using different devices. One of the players is connected through her game console. Another is playing the game on her mobile phone while travelling on the bus. The third player is using his PC to play the game. As can be seen from the figure, the most important component in this system is the server which has to support multiple connectivity methods for the same game being played simultaneously on multiple devices.

The server can also be used to support seamless migration from one device to another when playing an online game. For example, consider a player playing an online game such as *EverQuest II*[64]. At home she can play the game on her game console, connected to the internet. Then in the day time while on the move or sitting on a bus or train, she can continue to play the game seamlessly. In other words, there is no interruption no matter what device she is using. Later in the day, she may decide to continue playing using her PC. The *Anywhere Gaming* system will handle connections from all these diverse devices.

yoshiboy>> Go Red Go!!

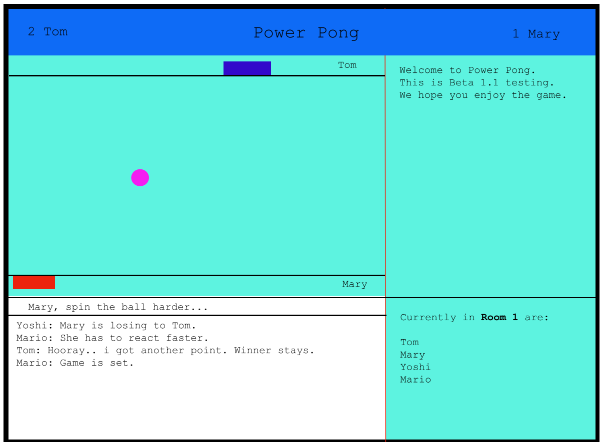
Figure 4.4: Screen view on the P800

4.5 Game Description

Our initial game is a modification of the famous Pong game. In the original Pong game, 2 players can play at a time. The aim of the game is to score goals by hitting a ball over the other end of the screen that is protected by the opponent. Each player uses a keyboard or joystick controlled bar to hit the ball. The game continues till someone first scores a certain number of goals.

Our modification allows the 2 players to play over a network (Bluetooth, Wireless LAN, or LAN). We refer to these 2 players as “pong players”. Apart from the pong players, there are “spectators” who can view the game on their devices, which can be mobile or stationary. We will refer to these spectators as “pong spectators”. Pong spectators interact with other pong spectators using text messages. These messages are also visible to the pong players. This promotes interaction between many persons simultaneously.

Figure 4.4 shows the screen view on the P800. The text message at the bottom



Welcome to Power Pong.
This is Beta 1.1 testing.
We hope you enjoy the game.

Mary, spin the ball harder...

Yoshi: Mary is losing to Tom.
Mario: She has to react faster.
Tom: Hooray.. i got another point. Winner stays.
Mario: Game is set.

Currently in **Room 1** are:

Tom
Mary
Yoshi
Mario

Figure 4.5: Screen shot of the game on the PC

of the screen is a comment from a spectator. Figure 4.5 shows a screen shot of Pong game on the PC. We have added features such as "angle shot","direct shot" and speed control. Angle shot allows players to hit the ball in a different direction. Based on laws of Physics, if the ball is moving in the same direction as the moving bar, the ball rebounds with a faster speed after collision. However due to constraints of playability, we have put an upper limit on the maximum speed of the ball.

The unique part about our modified Pong game is that it allows seamless integration of wireless and wired communication, and allows the game to be played on mobile as well as stationary devices.

4.6 System Description

There are three main architectural components in the game

1. The central game server

2. The mobile device

Figure 4.6: System Architecture

3. The stationary device

Figure 4.6 describes the system architecture. We have used a client-server model for the game. As shown in figure 4.6 the clients can be either mobile devices, such as a mobile phone, or stationary PCs.

The central game server: The central game server is the crucial component of the system. It is responsible for keeping all the clients synchronized. All the data transfer between various clients takes place through this central server. To support connectivity with both wired and wireless devices, the central server is equipped with multiple connectivity technologies. The present version of the central server supports Bluetooth, LAN and Wireless LAN connections. Within the server, there are different processes running. One of them is the game engine, while others handle network connections. These processes communicate using standard Interprocess Communication (IPC) techniques.

*The mobile device:*For our system, we chose the Sony-Ericsson P800 phone as the mobile device. The P800 runs on Symbian OS version 7.0. It has Bluetooth capability and a stylus touch screen display. The P800 communicates with the central server using Bluetooth. To play the game, the player uses a stylus to tap on the screen.

Stationary device: The stationary device is a PC running on Windows OS. It communicates with the server using the LAN. To play the game from a stationary device, the pong player can use the keyboard keys. Pong spectators can use the keyboard to type their messages.

4.6.1 Server Architecture

The task of designing the *Anywhere Game* server is the most crucial aspect of building the Anywhere Gaming system since the server is the heart of the system. We will first present the reasons for having a server based system.

4.6.2 Choosing the Communication Architecture

In order for players to be able to communicate with each other in the most effective way possible, it is need to decide how to set up the logical connections between the players. There are two ways by which players can communicate:

1. Client-Server
2. Peer-to-peer

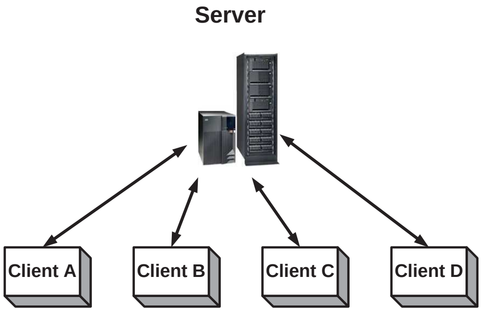


Figure 4.7: Client-Server architecture

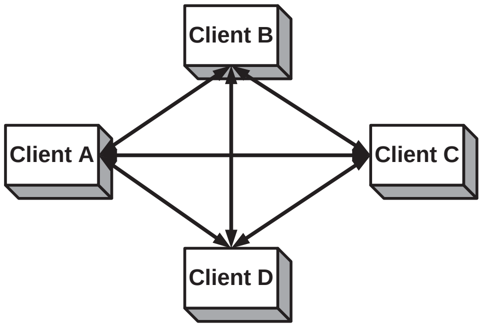


Figure 4.8: Peer-to-Peer architecture

Figures 4.7 and 4.8 show these architectures.

Client-Server Architecture

In client-server architecture, players send and receive all data through a central server. The advantage of this architecture is that each player has to maintain only a single connection to the server, no matter how many players are participating in the game. This greatly reduces network traffic on the players' side, since players have to send their messages only once. This is illustrated in figure 4.9 where client A wants to send packet to the other clients. To do so, it needs to send only a single packet to the server. The server will then forward the packet to clients B,C, and D. Thus the bandwidth requirement on the client side is low.

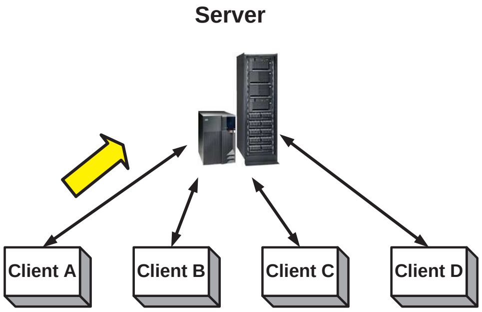


Figure 4.9: Traffic load on client side remains constant even if the number of participating players increases

The other advantage of using this architecture is that server can take an active role in managing the gameplay. This can range from coordinating some aspects of the game that are not player dependent. For example, in an adventure game, the weather is an external factor for the players and should be handled by the server rather than the players. The clients should only tell the server what their player is doing. This also prevents the players from cheating since they have no control over the server.

Another advantage is that game can be easily extended by doing a software upgrade on the server side. Thus minor bug fixing on server side and adding new levels are simple tasks.

The major disadvantage of this architecture is the problem of latency. Since the messages have to travel along two communication pathways, i.e, *client* → *server* → *other clients*, the time taken for the packet to reach its destination can be high.

Another problem is that servers can be become bottlenecks if the number of players is too high. But this can be avoided by using a number of servers and distributing the load among them.

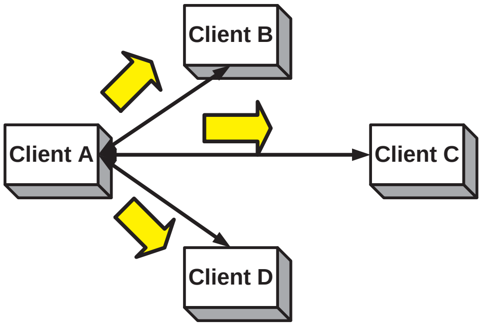


Figure 4.10: Traffic load on client side increases linearly with number of participating players

Peer-to-peer Architecture

In peer-to-peer architecture, all messages travel directly from the sending player to the receiving player, or set of receiving players. Thus there is no need for a server to pass the messages to other players. Since the messages have to travel only one way ($sending\ player \rightarrow receiving\ player$, instead of $sending\ player \rightarrow server \rightarrow receiving\ player$) the packet travel time is about twice as fast as that of the client-server architecture.

However, the disadvantage is that this architecture results in a higher network traffic and requires each player to have high bandwidth to be able to run the game smoothly. This is illustrated in figure 4.10 where client A wants to send packets to the other players. Since there are 3 other players, it has to send 3 packets on the network. Contrast this with figure 4.9 where client A had to send only 1 packet.

For example if there are 10 players in a game, then each player has to send the game packet to 9 other players simultaneously. Compare this with the client-server architecture where the each player would have to send only 1 packet to the server.

The other disadvantage of this is architecture is that the games become vulnerable to cheating since players can manipulate the client software.

Selection Decision

Having had a look at both these architectures, the client-server architecture is chosen because of the following reasons:

1. *Different communication methods:* The Anywhere Gaming system should support game play between devices using different connectivity methods. Players cannot be expected to use devices that have all the different connectivity methods in-built into them. For example it would be very difficult to find a PC that has support for LAN, WLAN, Bluetooth, GPRS etc.

   It is simpler to use a single server that can support all connectivity methods and let it communicate with the different devices. If in the future we need to add support for another connectivity method, all we need to do is upgrade the server only.

2. *Bandwidth on different connectivity methods:* The peer-to-peer architecture would have placed too large a load on the client side of the network. For example, Bluetooth has much smaller bandwidth than LAN. Therefore, as the number of players increase, the Bluetooth based device will be become the bottleneck as it won't be able to simultaneously send multiple packets.

3. *Reducing load on client processor:* In an Anywhere Gaming system some devices will have lower processing power than other devices and will take

longer time to do their calculations. As a result players using such devices will be at a disadvantage. In order to avoid this it is important to reduce the processing load on the clients. In the client-server server architecture, we can shift the common tasks to the server side thus ensuring fairness to all players.

4. *Preventing hacking:* The client-server architecture prevents malicious players from cheating and hacking.

4.6.3 Matchmaking Services

Before the game can actually begin, there has to be some way for players to be able to discover other players and decide with whom they want to play. On LANs (Local Area Networks), this is a trivial task, because players can broadcast their presence to everybody else. But if the players are on separate LANs the task becomes more difficult.

This is where matchmaking services are involved. Typically these services run on a central server. It is possible that the match-maker service runs on a different server than the one where the game-play takes place. The main purpose of the matchmaking service is to provide a place where players can meet and exchange information they need for further communication in the game. The two basic services that the matchmaker provides are:

- Creating new games
- Join games created by other players

More sophisticated matchmakers may provide services such as user account management, chatroom lobby so that players can discuss strategies before starting any game, information on player profiles and win/loss records.

4.6.4 Server Overview

Our *Anywhere Gaming* system is a server-driven system. The server can host multiple games and for each game it maintains the authoritative game state. Specifically, the server performs the following tasks:

- Provide match-making service so that new players can join as Pong players.
- Create new game instances.
- For each game instance, maintain game state and update all member players.

Figure 4.11 explains the architecture of the server. The *match-maker* handles connection from new players and presents them with a list of available games to join. Players can also initiate a new game if they desire. The *match-maker* maintains an upper-limit on the number of game instances that can be created on the server.

Figure 4.12 shows how the *match-maker* handles new connections from various devices. The *match-maker* has two processes:

1. The *match-maker* core process that does the house-keeping work of maintaining the list of players and other related work
2. The communication process that listens for requests from Bluetooth-device players and LAN-device players.

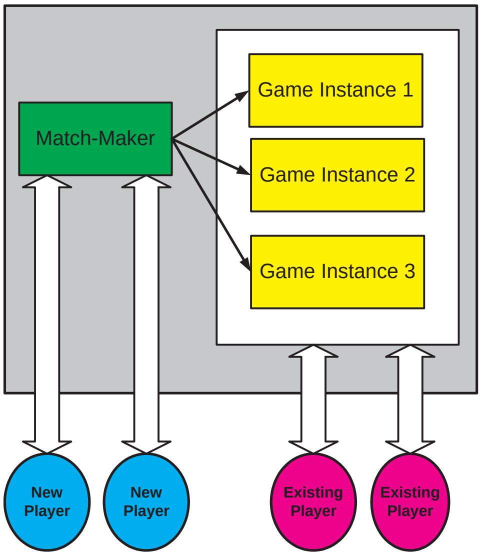


Figure 4.11: Game server architecture

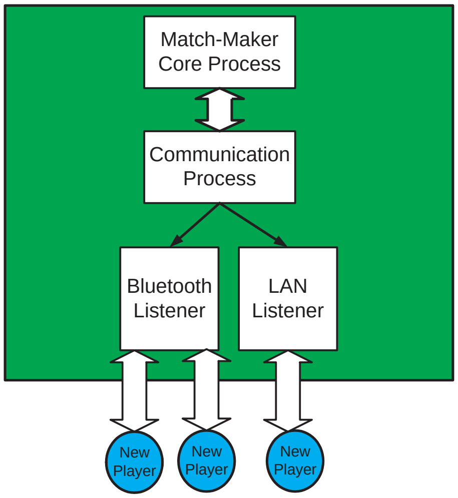


Figure 4.12: Processes within the *Match-Maker*

Once two players have joined as Pong players, the *match-maker* creates a new game instance and the game begins. Figure 4.13 gives the flowchart for the match-maker process. The flowchart has simplified a bit by lumping the Bluetooth and LAN connections together. Thus in figure 4.13, "Listen for data any socket" means that it can listen for data on any TCP/IP socket on the LAN as well as any data from the Bluetooth devices.

Each game has two Pong players currently. An expansion of the game is considered such that more players could join the game. Figure 4.14 shows the processes within a game instance. Each game instance contains two processes - the *game-state* process, and the *game-communication* process. The *game-state* process is concerned with maintaining the state of the game, while the *game-communication* process is responsible for handling connections with various players. It creates a player thread for each player who is connected to the game. Events reported by various players are relayed to the *game-state* process. The *game-state* process then updates its state and instructs the *game-communication* process to update the other players.

4.6.5 Bandwidth and Latency Issues

The critical problems of bandwidth and latency will be discussed in this section.

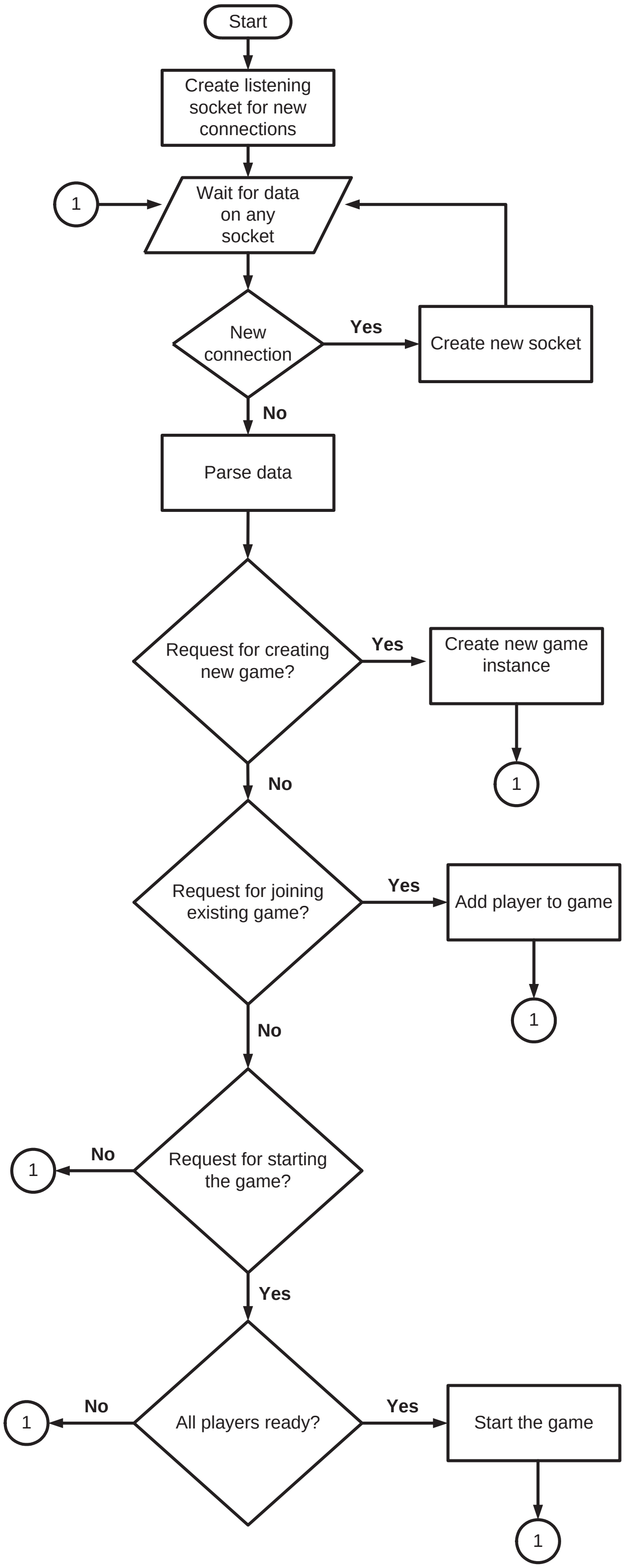


Figure 4.13: Flowchart of the matchmaker

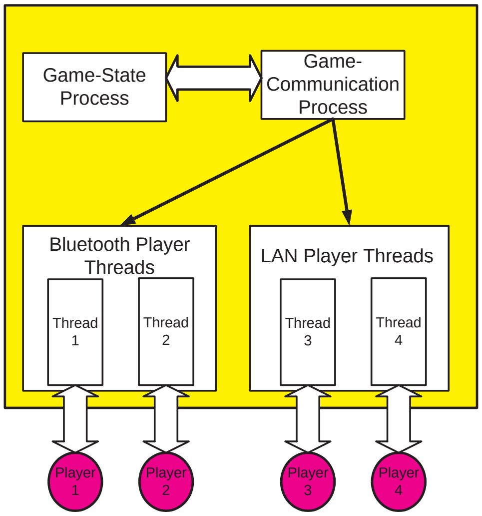


Figure 4.14: Processes within a game instance

Bandwidth

The Anywhere Gaming system supports connectivity on different communication methods. It is important that clients who are using low bandwidth methods do not suffer when the number of players increases.

Therefore when designing the network protocol, i.e., what information should be sent out on the network and in what format, one have to made sure that it would be small enough for the lowest bandwidth connectivity method. This was ensured in the following way:

1. *Use numbers instead of string to describe player state:* For example, use the number 1 to indicate player is ready, instead of saying something like "player ready"
2. *Use bit fields instead of complete numbers:* For example if there are only 8 possible player states, it is possible to specify all possible states using 3 bits

of data. Therefore, when implementing the protocol in C++, we used the declaration

```
unsigned i : 3;
```

instead of

```
unsigned i;
```

For our system, the Bluetooth connectivity method had the lowest bandwidth.

Latency

One of the biggest problems in designing an Anywhere Gaming System was the issue of latency. For example if a packet takes 200 ms to travel from sender to receiver, then any action made by one player will be seen by other players after 200 ms. For turn based games such as chess this is not a problem, since other players have to wait for other players before they can make their own move. But in action games each player moves independently of others and thus it is important that all players have the same view of the virtual world.

In our system, to solve the problem through the so called *Dead Reckoning* method is chosen. The idea behind this technique is that an approximation of the truth is likely to be as good as the truth itself. Therefore, additional information is transmitted with the network messages. For example, instead of simply sending player positions, the current orientation and speed are also sent. This makes it possible for other players to estimate the subsequent movements and likely positions, until the next message arrives with the updated information.

4.7 Conclusion and Future Work

The Pong game implemented in our project allows interaction of multiple mobile and stationary devices. We used Bluetooth connectivity for the mobile devices and LAN connection for the stationary devices. One drawback of using Bluetooth was that the mobile device users had a limited mobility of 10 meters from the game server. In the future, high speed 3G connections such as CDMA can be used instead of Bluetooth. In this way, multimedia phone users would be able to play games even if they are far away from the game server. This work is a first step on our main research goal of “Anywhere Gaming”.

Chapter 5

Conclusion

In this thesis we focussed our attention on ubiquitous interaction on mobile devices. We identified three approaches to improving interaction with mobile devices:

1. By development of location-aware mobile devices. This improves *human – mobile device* interaction.
2. By improving the user interface of mobile devices. This improves *human – mobile device* interaction.
3. By developing applications that promote interaction between different mobile device users.

In order to facilitate the development of location-aware mobile devices, we developed an indoor location tracking system that uses only the existing WLAN infrastructure. Our tracking system follows a *location region* based approach that directly identifies the location region of the user. We used a *mobile device assisted,* *centrally based* tracking architecture for our system. This allows easy updates to the tracking system and causes minimal disruption of the WLAN traffic. We compared the tracking accuracies using different pattern recognition algorithms and found that the Bayes' Clustered Region approach gave best tracking results. We also conducted experiments to study how the placement of the access points affects the tracking accuracy.

In the second part of the thesis, we focussed our attention on developing better user interfaces for mobile devices. For this, we explored the use of augmented reality on mobile phones. We developed a prototype of the system using the commercially available Sony Ericsson P800 mobile phone. Using our prototype system we developed two applications - the AR Post-It system and the AR Catalogue system. The AR Post-It aims to complement the existing paper and digital messaging systems by allowing users to send digital messages that are location based. The other application that we developed was the AR Catalogue system. It allowed mobile phone users to view 3D augmented reality pictures on their mobile phones. This was developed with the primary purpose of enhancing the paper based catalogues. We conducted user studies to study the response of users towards augmented reality applications on mobile phones. The user feedback confirmed our belief that augmented reality on mobile phones have exciting commercial potential.

In the third part of the thesis, we focussed our attention on developing applications that promoted interaction among mobile device users. For this purpose, we identified networked games as the most promising genre of applications. We presented our work on an *Anywhere Gaming* system that allowed mobile devices and PCs to play a simple Pong game in real time. This game showed that it is possible to allow interaction among different mobile device users, as well as among mobile device and stationary device users.

Appendix A

Contents of the CD-ROM

A CD-ROM attached to the inside back cover of this thesis provides a soft copy version of the thesis, as well as the source codes for the applications mentioned in this thesis. It has three main folders. The contents of these folders are as follows:

- **Thesis :** Contains a soft copy of the thesis. The file name is `Thesis.pdf`.
- **WLAN Tracking Software :** Contains the source code for the WLAN Tracking Applications.
- **AR Phone Software :** Contains the source code for the mobile phone applications.

Appendix B

Tutorial for the WLAN Tracking Software

Appendix B explains how to install and run the WLAN Tracking software while Appendix C explains the same for the AR applications for the mobile phone. We have used the following conventions in the tutorials in appendices B and C:

- Sans-serif fonts are used to display file names and folders:
  C:\Windows\System32.
- Commands that need to be typed in the MS-DOS window are displayed in a typewriter-like font:
  `copy file1 file2`.
- Menus and submenus are displayed in emphasized and boldface text:
  Start→Run.

Module	**Purpose**
The Signal Sampling Module	Collection training data as well as test data
The Signal Preprocessing Module	Preprocessing of the training data
The Tracking Comparator Module	Used to compare the tracking accuracies of different methods. Requires preprocessed training data and the test data
Online Location Tracking Modules	Used to track the location of a mobile device in real-time. Consists of two modules, one of which needs to be installed on the mobile device while the other needs to be installed on the location server. The module on the location server needs preprocessed training data.

Table B.1: The purpose of the different modules that were developed for the WLAN Tracking Research.

The source codes for the WLAN tracking application are stored on the CD in the folder WLAN Tracking Software. This folder contains the following sub-folders:

- Signal Sampling Module
- Signal Preprocessing Module
- Tracking Comparator Module
- Online Location Tracking Modules
- Dll

Table B.1 shows the different modules that were developed for the research work and also briefly mentions their purpose.

B.1 System Requirements

B.1.1 Hardware

The WLAN applications requires the following hardware:

- **A WLAN-enabled mobile device:** The mobile device should be running on Windows XP operating system, and should be able to connect to a WLAN through a PCMCIA WLAN card.
- **PCMCIA WLAN card:** This is required for the mobile device so that it can connect to the WLAN.
- **WLAN Routers:** For setting up a testbed for research purposes, WLAN routers are required. Otherwise, the existing WLAN infrastructure can be used.
- **Desktop Computers:** Two desktop computers are required for doing the real time tracking. One of these acts as a location server, while the other computer is used to display the location. Ideally, these computers should be running on Windows XP.
- **WLAN Adaptors:** These are needed for the desktop computers so that they can connect to the WLAN.

B.1.2 Software

The mobile device requires pre-installation of the Windows Driver Development Kit (DDK). In addition, other steps need to be carried out for proper configuration.

B.1.3 Configuring the mobile device

The mobile device needs to be configured so that it can record the WLAN signals. Therefore, the DDK needs to be installed and compiled. Then the compiled files need to be copied into the Windows system folder. We will now explain these steps in detail.

The DDK is available free of charge from Microsoft. Insert the DDK CD into the CD-ROM drive and install the DDK. The DDK is always installed at the root directory. For this tutorial, we will assume that the DDK has been installed in C:\. Once the DDK has been installed, the drivers need to be compiled. Perform the following steps for compiling the DDK:

1. Start a build environment for the DDK. For this go to ***Start→DDK→Build*** and click on ***Free Build Environment***. This will start the DDK command prompt.
2. From the command prompt, change the working directory to C:\WinDDK\2600.1106\src.
3. At the command prompt, type the following: `build - cZ`

After the drivers have been built, we need to replace the ndisuio.sys in the

Windows system directory (C:\Windows\System32) with the DDK compiled files. However, a simple copy-paste will not work, since Windows XP will not permit changes to the system files. Therefore an additional software, called the *Recovery Console*, needs to be installed. Perform the following steps for installing the *Recovery Console*:

1. Open the command prompt window. This can be done by going to ***Start→Run***. In the MS-DOS command prompt, type: `cmd`
2. Insert the Windows XP setup CD into the CD-ROM drive.
3. If E is the CD-ROM drive, then type the following at the command prompt:
   ```
   E:\i386\winnt32.exe /cmdcons
   ```

Now that the *Recovery Console* has been installed, perform the following steps for replacing the system file:

1. Copy the compiled ndisuio.sys file to the root Windows directory (C:\). The compiled file is within the DDK folder and the path is: C:\WinDDK\2600.1106\src\network\ndis\ndisuio\sys\obj\i386.
2. Restart Windows XP and choose ***Microsoft Windows Recovery Console*** at the boot menu.
3. This will start the *Recovery Console* application instead of Windows XP. Now change the working directory to the Windows root directory by typing:
   ```
   cd C:\
   ```

4. Replace the Windows ndisuio.sys file with the compiled ndisuio.sys by typing:

   ```
   copy ndisuio.sys C:\Windows\System32
   ```

This completes the configuration steps associated with the DDK. On the software CD, open the folder WLAN Tracking Software\Dll, and copy the file Wrapi.dll to the C:\Windows\System32 folder. The mobile device is now ready to track the WLAN signals.

B.2 The Signal Sampling Module

The Signal Sampling Module is used to collect the training data as well as the test data. The training data is used to train the tracking system, while the test data needs to be collected when we wish to compare the tracking accuracies of different algorithms.

On the software CD, open the folder WLAN Tracking Software, and copy the folder Signal Sampling Module onto the mobile device's C:\ folder.

B.2.1 Compiling the Signal Sampling Module

The WLAN tracking software requires knowledge about the identities of the access points (APs) in the WLAN. Perform the following steps:

1. Open the folder C:\Signal Sampling Module\DataCollectionDll and then open the VC++ workspace DataCollectionDll.dsw.

2. In the DataCollectionDll.cpp file, change the appropriate variables to specify the number of APs that are to be used for tracking, and also enter the MAC addresses of all these APs.
3. Then compile and build.
4. Copy the generated DataCollectionDll.dll file (from the folder C:\Signal Sampling Module\DataCollectionDll\debug) into the Windows system folder (C:\Windows\System32).

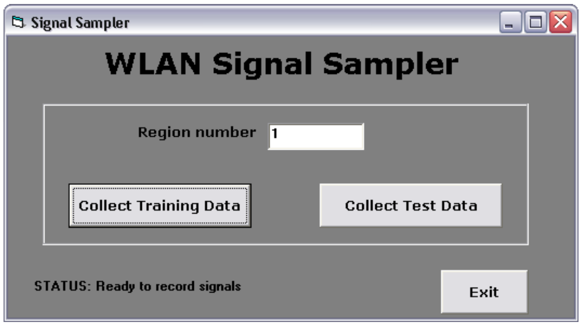
B.2.2 Collecting Training Data

Perform the following steps:

1. Windows XP provides a service called the *Wireless Zero Configuration*. This needs to be disabled to enable the tracking of the WLAN signals. Go to ***Start→Control Panel→Performance and Maintenance→Administrative Tools→Services***. Then disable the *Wireless Zero Configuration* service.
2. Open the command prompt, and type

   ```
   net start ndisuio
   ```

3. Open the folder C:\Signal Sampling Module\Signal Sampler and run the Signal Sampling module by double clicking on Signal Sampler.exe. A GUI application will be started. This is illustrated in figure B.1.
4. Enter the region number in which data is being collected. Note that in each region there can be multiple individual locations. The region number should



Region number 1

STATUS: Ready to record signals

Figure B.1: The signal sampler.

be changed only when data is being collected in a different region.

5. Click on the button ***Collect Training Data***. This will begin collection of the signal samples. The sampling time at each location is 100 seconds.
6. Record the signal samples at all the required individual locations.
7. The raw signals are stored in `.txt` files. A `.txt` file is generated for each AP that is present in the WLAN. These files are named after the MAC address of the respective APs, e.g. `00e098abff3c.txt`.

B.2.3 Collecting Test Data

The procedure for collecting the test data is similar to that for collecting the training data. However, in step 5, click on ***Collect Test Data*** instead of ***Collect Training Data***. Collection of test data at each individual location requires 5 seconds.

It should be noted that collection of the test data is not required if we just want to do real-time tracking of a mobile device. The test data is required only if we wish to compare the tracking accuracies of different algorithms.

The test data is stored in .txt files with the following name format: online_00e098abff3c.txt

B.3 The Signal Preprocessing Module

The Signal Preprocessing Module is used to preprocess the raw training data. The preprocessing can be done on any computer, but for this tutorial we will do it on the mobile device. On the software CD, open the folder WLAN Tracking Software, and copy the folder Signal Preprocessing Module onto the mobile device's C:\ folder.

Perform the following steps in order to use this module:

1. Copy all the raw signal files (e.g., 00e098abff3c.txt) into the C:\ folder.
2. Open the C:\Signal Preprocessing Module\Preprocessor folder and copy the WLAN Preprocessor.exe file to the C:\ folder.
3. Open the C:\Signal Preprocessing Module\DllWlanPreprocessor\debug folder and copy the DllWlanPreprocessor.dll file to the Windows system directory (C:\Windows\System32).
4. Run the WLAN Preprocessor.exe program in the C:\ folder. This will open a GUI application, as shown in figure B.2.
5. On the GUI application, change working directory to C:\Signal Preprocessing Module\Preprocessor by using the drop down lists.
6. Select all the raw signal files (e.g., 00e098abff3c.txt) by clicking on them.
7. Once all the raw signal files have been selected, start the preprocessing by clicking on the ***Preprocess*** button.

Preprocessed File Name	**Used For**
min_dist_00e098abff3c.txt	Minimum Distance method, k-Minimum Distance method
cluster_center_00e098abff3c.txt	Cluster-Center Distance method
bayes_ind_00e098abff3c.txt	Bayes' criteria using individual locations
bayes_clust_00e098abff3c.txt	Bayes' criteria using clustered locations

Table B.2: Four different preprocessed files are generated from the raw file, 00e098abff3c.txt.

8. The module will generate four different preprocessed files to be used for various tracking techniques. Table B.2 shows the preprocessed files that are generated, and the algorithms which use these files.

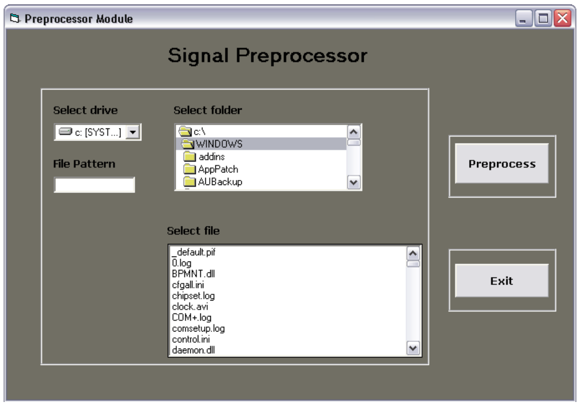


Figure B.2: GUI for the *signal preprocessing* module

B.4 The Tracking Comparator Module

The tracking comparator module is used to compare the tracking accuracies of the different tracking algorithms. The module can be installed on any computer, but for this tutorial we will install it on the mobile device. On the software CD, open the folder WLAN Tracking Software, and copy the folder Tracking Comparator Module onto the mobile device's C:\ folder.

Perform the following steps in order to use this module:

1. Copy all the preprocessed signal files (e.g., min_dist_00e098abff3c.txt, bayes_ind_00e098abff3c.txt) as well as the test data files (e.g., online_00e098abff3c.txt) into the C:\ folder.
2. Open the C:\Tracking Comparator Module\Tracking Comparator folder and copy the OnlineTrackCompare.exe file to the C:\ folder.
3. Open the C:\Tracking Comparator Module\OnlineTrackCompareDll\debug folder and copy the OnlineTrackCompareDll.dll to the Windows system directory (C:\Windows\System32).
4. Run the OnlineTrackCompare.exe program that is located in the C:\ folder.
5. From GUI, change working directory to C:\Tracking Comparator Module by using the drop down lists.
6. Select all the raw signal files (e.g., 00e098abff3c.txt) by clicking on them.
7. Once all the raw signal files have been selected, start the tracking comparison by clicking on the ***Compare Tracking Methods*** button.

Tracking Method	File names containing the results
Minimum Distance method	min_dist_results.txt
k-Minimum Distance method	k_min_dist_results.txt
Cluster-Center Distance method	cluster_center_results.txt
Bayes' criteria using individual locations	bayes_ind_results.txt
Bayes' criteria using clustered locations	bayes_clust_results.txt

Table B.3: The files that contain the tracking results.

8. The module will compare the tracking accuracies of different methods using the test data. Table B.3 shows the file names that are generated for the various tracking algorithms:

B.5 The Online Location Tracking Modules

These modules enable tracking of a mobile device in real time. The following three computers need to be used during the real-time tracking:

- **Mobile computer:** This is the mobile device that needs to be tracked
- **Location Displayer computer:** This is the computer on which the location of the mobile computer is shown in real-time using a GUI. This computer sends a request to the *location server* computer to track a specific mobile device.
- **Location Server:** This is the computer which does the actual location tracking, and returns the estimated location to the *location displayer* computer.

Before running the modules, we need to ensure that all these 3 computers are on the same WLAN and can "see" each other. The following steps can be followed:

1. Attach a WLAN router to the location server.

2. Restart the mobile device and the *location displayer* computer and use peer-to-peer networking configuration so that both of them are now attached to the location server's WLAN router. A number of tutorials and HOW-TOs are available on the public domain to show how this can be done.
3. Make sure the mobile device and the *location displayer* computer can *ping* the location server.

B.5.1 Tracking module for the mobile device

On the software CD, open the folder WLAN Tracking Software\Online Location Tracking Modules, and copy the folder WLANScanner onto the mobile device's C:\ folder.

Perform the following steps in order to use this module:

1. Perform steps 1 and 2 mentioned in section B.2.2.
2. Open the C:\WLANScanner folder and open the WLANScanner.dsw VC++ project. In the file WLANScanner.cpp, make sure that the number of APs and their MAC address are correct. Compile and build the project.
3. Run the application. It will start and then wait for signal scanning requests from the location server. Once the request comes, it will scan the WLAN and return the signal values to the location server.

B.5.2 Tracking module for the Location Server

On the software CD, open the folder WLAN Tracking Software\Online Location Tracking Modules, and copy the folder Location Server onto the location server's C:\ folder.

Perform the following steps in order to use this module:

1. Copy all the preprocessed signal files for the *Bayes' Clustered Region* method(i.e., bayes_clust_results.txt) into the C:\Location Server folder.
2. In the C:\Location Server folder, open the Location Server.dsw VC++ project. In the Location Server.cpp file, ensure that the IP address of the mobile device is correct. Compile and run the application. The location server will start and wait for tracking requests from the *location displayer* computer.

B.5.3 Tracking module for the *Location Displayer* computer

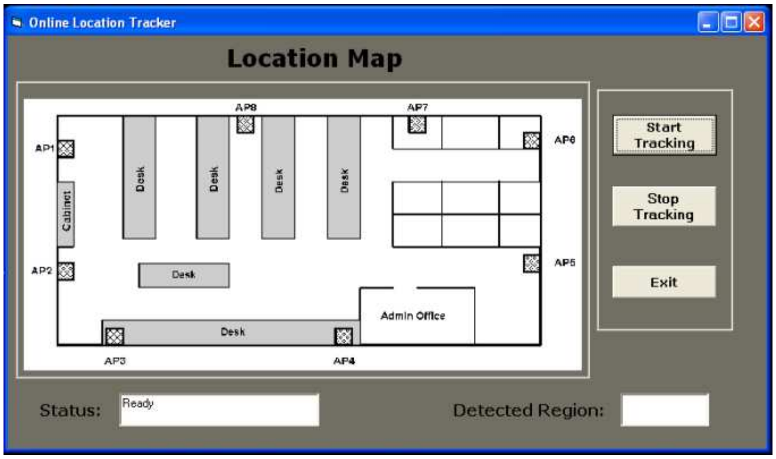
On the software CD, open the folder WLAN Tracking Software\Online Location Tracking Modules, and copy the folder Location Displayer onto the C:\ folder of the *Location Displayer* computer.

Perform the following steps in order to use this module:

1. Open the C:\Location Displayer\DllAspScan folder and open the DllAspScan.dsw VC++ project. In the DllAspScan.cpp file, make sure the IP address of the location server is correct. Then compile and build the project. From the debug

subfolder, copy the DllAspScan.dll file to C:\Windows\System32.

2. Open the C:\Location Displayer\Location Announcer folder and double-click on the Location Announcer.exe file. This will start the GUI application, as shown in figure B.3. It will send a scan request to the location server to track the mobile device. The application updates the mobile device's location every 10 seconds.



Status: Ready

Detected Region:

Figure B.3: GUI application for the *location displayer* computer.

Appendix C

Tutorial For the Augmented Reality Applications

On the CD, the folder AR Phone Software contains the following files:

- glut-3.7.6-bin.zip
- Bluetooth.zip
- AR Post-It.zip
- AR Catalogue.zip
- Mobile phone module.zip
- MXR Toolkit.zip
- pgrflycapture1.4a018.exe

C.1 Hardware Requirements

The AR application requires the following hardware:

- **A Desktop Computer :** The recommended system requirement for the desktop computer is Pentium 4 2.0GHz Processor, a 64Mb Graphic card and 512Mb RAM. Ideally, the computer should have the Windows 2000 or Windows XP operating system.
- **A Sony Ericsson P800 mobile phone**
- **A Bluetooth Adaptor (USB or Serial type)**

If the users want to compile and build the source code, then all the steps from given in sections C.2 – C.8 should be performed. If the users merely want to test the programmes without looking at the source, installations steps in section C.4.1, C.6, C.7 and C.8 are still required.

C.2 Integrated Development Environments

The software for *AR Post-It* and *AR Catalogue* is developed using Microsoft Visual C++ 6.0, the mobile phone module is developed using Codewarrior for Symbian Pro 2.0. Both IDEs contain a fully integrated editor, compiler, and debugger that will be used in this project. Unfortunately, they are not freeware and have to be purchased. To install Microsoft Visual C++ 6.0 and Codewarrior for Symbian Pro 2.0, the user can follow instructions given by the installation wizard.

C.3 GLUT

GLUT is the OpenGL Utility Toolkit, a window system independent toolkit for writing OpenGL programs. Only *AR Catalogue* uses GLUT. The `glut-3.7.6-bin.zip` contains all the necessary file to use it. Alternatively, the latest version of glut can be downloaded from

http://www.xmission.com/ nate/glut.html.

Assuming that the Microsoft Visual C++ 6.0 has been installed into the default location, the following steps are to be follow:

1. Unzip the `glut-3.7.6-bin.zip` to a temporary folder.
2. Place the `glut32.lib` in `C:\Program Files\Microsoft Visual Studio\VC98\Lib`.
3. Place the `glut32.dll` in `C:\WINDOWS\system32`.
4. Place the `glut.h` in `C:\Program Files\Microsoft Visual Studio\VC9\Include\GL`.
5. To create a console application, go to ***Project Settings*** in Microsoft Visual C++ 6.0.
6. Under the ***Link*** menu, add `Opengl32.lib` , `glu32.lib` and `glut32.lib`

Step 5 and Step 6 are only relevant if the user wishes to build a GLUT application from scratch. To run the programme of this project, steps 5 and 6 can be skipped.

C.4 Bluetooth Software Installation

The following installations are only applicable for running the *AR Post-It* and *AR Catalogue* which uses Bluetooth functions running on the desktop.

C.4.1 TDK USB Bluetooth Stack Installation

The installation of TDK USB Bluetooth Stack is direct and simple. One thing to bear in mind is not to connect the Bluetooth USB Adaptor until the software is installed.

1. Unzip the Bluetooth.zip into a temporary folder.
2. Open the Setup.exe.
3. Click on the ***Install Bluetooth Stack***.
4. Follow the instructions given by the installation wizard.
5. Connect the Bluetooth USB Adaptor once the software is installed.

C.4.2 Widcomm Software Development Kit (SDK) Installation

The installation of the Widcomm SDK must come after the Bluetooth Stack installation mentioned earlier.

1. Open the same folder extracted earlier from Bluetooth.zip.
2. Open the same Setup.exe.

3. Click on ***Install DK and Code Samples***.
4. Follow the instructions given by the installation wizard.

The SDK consists of a DLL, a library file and two header files.

1. WidcommSdk.dll should be placed in a directory that is in the DLL search path, typically the working directory for the executable or the Windows system directory.
2. Applications should be linked with WidcommSdk.lib.
3. BtIfDefinitions.h is a header file that defines constants and structures used by the DK and applications.
4. BtIfClasses.h is a header file that defines the classes offered by the DK.

C.5 Symbian UIQ 2.0 SDK Installation

The installation of Symbian UIQ 2.0 SDK must come after the installation of Codewarrior for Symbian Pro 2.0 in order for them to work properly. Symbian UIQ 2.0 SDK can be downloaded from *http://www.symbian.com/developer/sdks_uiq.asp*. For installation, the user can follow instructions given by the installation wizard.

C.6 MXR Toolkit Installation

The MXR Toolkit consists of a two DLLs, two library files and two header files. Only *AR Post-It* and *AR Catalogue* use the MXR Toolkit.

1. mxrSDK.dll and mxrMedia.dll should be placed in a directory that is in the DLL search path, typically the working directory for the executable or the Windows system directory.
2. Applications should be linked with mxrSDK.lib and mxrMedia.lib.
3. mxrSDK.h and mxrMedia.h are header files that defines constants and structures and classes used by the toolkit and applications.

For the MXR Toolkit to work properly, the installation of DirectX SDK version 8.0 or above and Point Gray Research Camera drivers are required.

DirectX SDK can be downloaded at *http://www.microsoft.com/downloads/* while Point Gray Research Camera drivers can be installed by running pgrflycapture1.4a018.exe which is included on the CD.

C.7 PC Suite for P800 Installation

PC Suite for P800 comes in a CD together with the purchase of the mobile phone. This installation is necessary to enable connectivity between the mobile phone and the desktop computer. Installation is fairly straight forward as it starts automatically when the CD is inserted into the CD-ROM.

C.8 SIS Package Installation

A SIS package file is an installation file which will port the application (mobile phone module) in the encapsulated in package file to the Sony Ericsson P800 mobile phone. Once the PC Suite for P800 installation is done, the mobile phone module can be installed into Sony Ericsson P800 mobile phone by double clicking the file MXR Client.sis which can be found after decompressing the file Mobile phone module.zip.

C.9 Running The Applications

Three programmes are provided in the CD for users to run, and they are stored in AR Post-it.zip, AR Catalogue.zip and Mobile phone module.zip. AR Post-it.zip and AR Catalogue.zip contains the softwares for the *AR Post-It* system and *AR Catalogue* respectively. Mobile phone module.zip contains the SIS package which needs to be installed on to the phone as described in Appendix C.8.

The instructions for running the *AR Post-It* system and *AR Catalogue* are exactly the same.

C.10 AR Post-It

To run this system, the server application must setup and run first before the client application.

C.10.1 Running The Server Application

The steps are as follow:

1. AR Post-it.zip into the desktop computer.
2. Create a directory with the name AR Post-It in C:\ and extract the zip files into it, i.e. C:\AR Post-It.
3. To begin the server application, click on AR Post-It shortcut in C:\AR Post-It directory.
4. The application will launch a window and wait for the client application to connect.
5. To exit the application, hit the “ESC” key.

C.10.2 Running The Client Application

Prior it the steps shown below, users must install the SIS package on to the P800 mobile phone by following the instructions in Appendix C.8.

1. Open the flip (or virtual flip) on the P800 mobile phone to change to PDA mode.
2. From the list of application, choose the ***MXR Client*** application.
3. After the client application is launched, click on the ***Connect*** pull down menu and click on ***Select Remote***.

4. A dialog box will appear and the application will search for nearby discoverable Bluetooth devices. This searching process can take up to half a minute. Click on the desktop where the server application is running.
5. From the pull down menu again, click ***Connect*** → ***Connect*** If a connection is established successful, there will be a notification message.
6. Once connected, the camera viewfinder will appear. Users can start taking snapshots by pressing the camera button on the right side of the mobile phone.

C.11 AR Catalogue

The instructions for running the *AR Catalogue* are exactly the same as those for the *AR Post-It* system.

Appendix D

Architecture for Ubiquitous WLAN Tracking

D.1 Introduction

The central aim of this chapter is suggest recommendations on how to enable WLAN-based location tracking across multiple WLANs. This would allow users to experience a truly ubiquitous computing environment.

In his famous paper on ubiquitous computing, Mark Weiser had said [65]:

> The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Following the same philosophy, we believe that for WLAN-based tracking systems to gain acceptability, the user should not have to concern herself with the low-level details such as updating the radio map, or downloading a new radio map whenever she goes to a new WLAN. In this chapter we propose a new framework for WLAN-based tracking systems with the following aims in mind:

1. Support non-local roaming (i.e., user can seamlessly migrate to a new WLAN)
2. Support non-local lookup (i.e., the object of interest should not have to be located in the same WLAN)
3. Support easy updating of the tracking system without troubling the user

In chapter 2 we had introduced the architecture of a *mobile device assisted, centrally based* tracking system. The key advantages of this architecture were:

1. Minimal disruption of WLAN traffic, and
2. Support for easy updates to the location tracking system

We will now propose addition of some system components that would allow integration of the tracking systems of different WLANs.

The rest of the chapter is organized as follows: in section D.2 we present an overview of the system architecture and elaborate on the design guidelines. In section D.3 we describe the system operation using case scenarios and show how the various system components interact with each other. Then, in section D.4, we provide a discussion on the issues of user privacy and authentication. Finally, we provide a conclusion in section D.5.

D.2 Overview of the system architecture

Tracking systems based on other technologies such as GPS or cellular telephony contain infrastructure that is owned entirely by a single organization. Therefore it is easier to implement policies.

However, WLANs are installed and owned by different organizations. Therefore it is important to design the system such that different WLANs can implement their own location services, and if desired, a number of different WLANs can form a group so that specific location services can be provided seamlessly across all these WLANs. We had the following guidelines in mind when designing the system:

1. **Support for non-local roaming:** In order to have a ubiquitous location service, it is essential not to restrict the user's movement to within a single WLAN. That would defeat the very purpose of pervasive computing. Moreover, from the point of user convenience, the mechanism should be transparent to the user. This would allow the user to move seamlessly from one WLAN to another without botheration.
2. **Support for non-local lookup:** The system architecture should be such that a remote user can avail of location services in a far-away place. For example, a user on her desktop PC can track her friend who is using her mobile device in some WLAN. This would allow a wider variety of location services to be provided to the user and thus enhance the usefulness of the existing WLAN infrastructure.

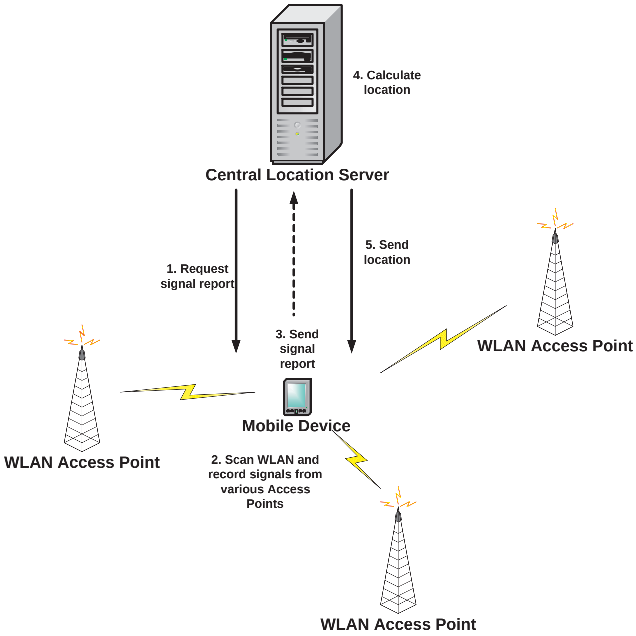


Figure D.1: Mobile device assisted tracking.

We propose the addition of some new system components to the *mobile device assisted, centrally based* architecture that was introduced in chapter 2. Figure D.2 illustrates the key components of the system. They are:

1. **Application server:** Application server is responsible for providing location based services. It is concerned not with locating a mobile device, rather using that information to provide a variety of services. For example, based on the mobile device's location the application server can tell it the location of the nearest bus stop. Each WLAN can have one application server. An application server may also be setup so that it serves a number of WLANs.
2. **Location server:** Each WLAN has a single location server. The location server is responsible for detecting the location of a mobile device. It detects the location of a mobile device upon receiving a request from the application

server.

3. **Mobile device:** Within a WLAN there can be many mobile devices. Mobile devices can request the application server for some location service. They may also receive requests from the location server for a signal report.
4. **Home Location Register (HLR):** The HLR is a global database in which information about all the mobile devices is registered.
5. **Visitor Location Register (VLR):** The VLR is a local database associated with a single WLAN and contains the list of only those mobile devices that are within its WLAN. The VLR can be accessed by the location server and the application server.

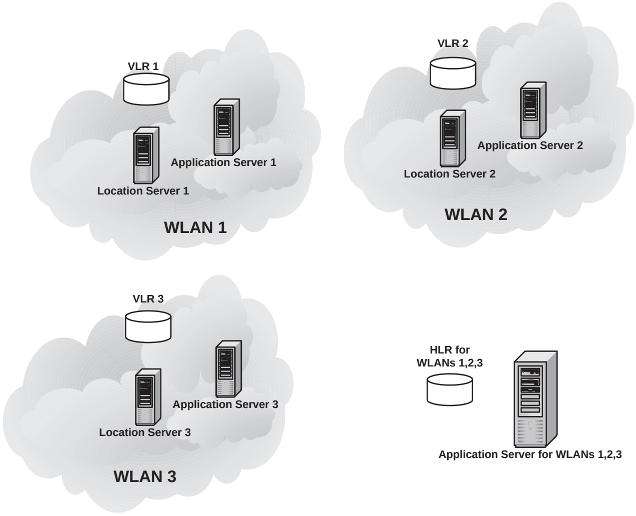


Figure D.2: The application server provides location services. Each WLAN has a location server that is responsible for location tracking. The mobile device can move seamlessly across different WLANs.

D.3 System operation

Broadly, location services can be classified either as *self-triggered* or as *externally-triggered.*

In a self-triggered service, the user's location is calculated only when requested by the user. Typically these services are information services, either personal location (where am I?) or service location (where is the nearest cafe?). In an externally-triggered service, the user's location needs to be detected when requested by other users or maybe because the user enters a new region. We will use case scenarios to discuss the operation of the system for both these types of services. All case scenarios consist of two basic steps:

1. Register the mobile device with the WLAN.
2. Request the application server for the desired location service.

Section D.3.1 explains step 1, while sections D.3.2, D.3.3 and D.3.4 discuss step 2 for three case scenarios.

D.3.1 Service discovery and registration

In order to use a location service, the mobile device will first have to indicate its interest by registering with the WLAN. Depending on the user's preference, the registration can either be *automatic* or *user-initiated.* In automatic registration, whenever the mobile device enters a new WLAN and associates with an AP, it will automatically be registered. In user-initiated registration, the registration will take place only when the mobile device initiates the process. The list of mobile devices that have registered themselves is maintained in a two-level hierarchical database, called the Visitor Location Register (VLR) and the Home Location Register (HLR). Each WLAN has a VLR, and there is one HLR for all the WLANs.

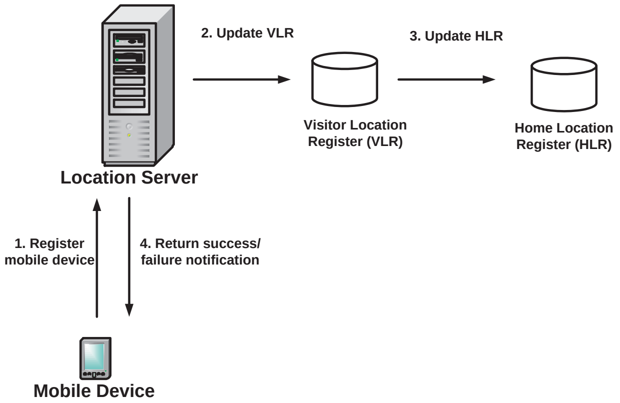


Figure D.3: Registering a mobile device with the WLAN.

Figure D.3 illustrates a user-initiated registration process. First the mobile device register itself through the location server. The VLR and HLR are both update. If the operation is successful, the location server returns a success notification; otherwise a failure notification is issued.

D.3.2 Self-triggered location service

In this section we consider the following scenario: A mobile user has entered a building with WLAN coverage and needs to know the location of the nearest cafe. We assume that the mobile device has already been registered, either automatically or through user initiation. It is thus eligible for availing of location services.

Figure D.4 illustrates the sequence of events for this location service.

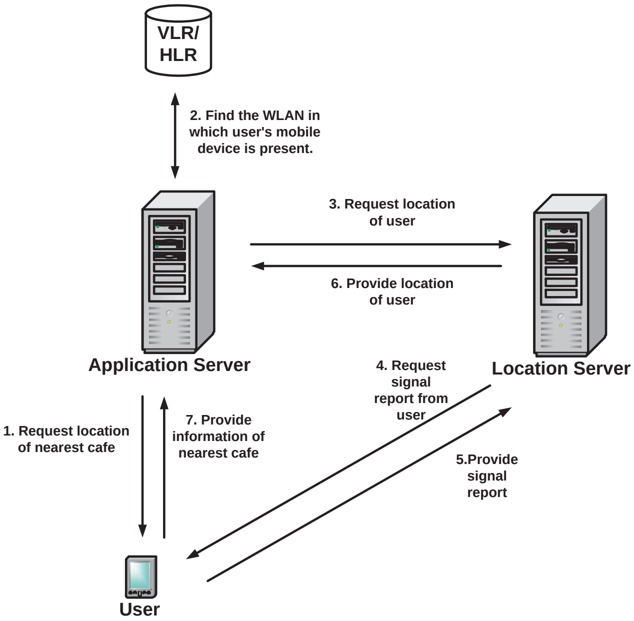


Figure D.4: Illustration of how a self-initiated service works.

First the user contacts the application server and requests it for the location of the nearest cafe. The application server looks up the VLR/HLR to find the WLAN in which the user is present. The application server then requests the corresponding location server to detect the location of the mobile device. The location server contacts the mobile device and requests it to send a signal report. The mobile device scans the WLAN and records the signal strength from various APs. It then sends the signal report to the location server. Based on the signal report, the location server "guesses" the location of the mobile device and sends it to the application server. The application server then finds the cafe nearest to the mobile device.

D.3.3 Externally triggered location service - both mobile users

In this section we consider the following scenario: A mobile device user, say user 1, in a WLAN wishes to find the the location of her friend, say user 2. Both, user 1 and 2, are in the same WLAN and both are using mobile devices. We assume that both users have already registered their mobile devices with their respective WLANs.

Figure D.5 illustrates the sequence of events for this location service.

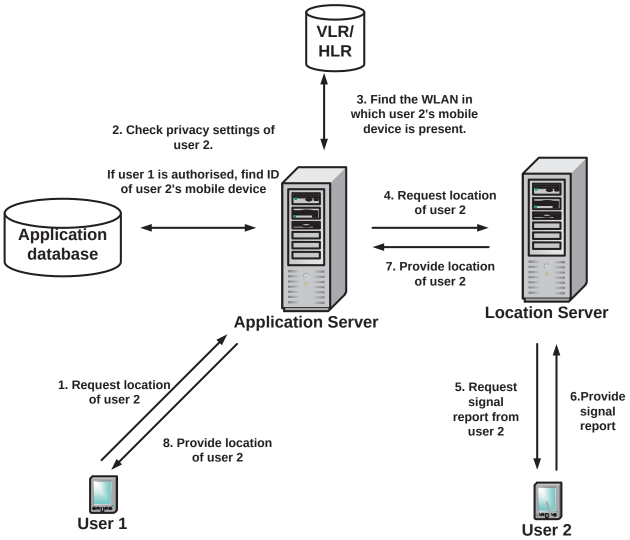


Figure D.5: Illustration of how an external user initiated service works.

First user 1 contacts the application server and requests it to find the location of user 2. The application server has a database that contains the privacy preferences of user 2. The application server finds that user 2 has given access privilege to user 1. The application server then looks up the VLR/HLR to find the WLAN in which user 2 is present. It then contacts the corresponding location server with a request to detect the location of user 2. The location server contacts user 2's mobile device and requests for a signal report. User 2's mobile device scans the WLAN and sends a signal report to the location server. Based on the signal report the location server "guesses" the location of user 2 and sends it to the application server. The application server then shows the location of user 2 to user 1.

D.3.4 Externally triggered location service - one mobile device user, one stationary device user

In this section we consider the following scenario: A desktop PC user, say user 1, wants to know the location of her friend, say user 2, who is in some WLAN. We assume that the user 2 has already registered her mobile device with the WLAN.

Figure D.6 illustrates the sequence of events for this location service.

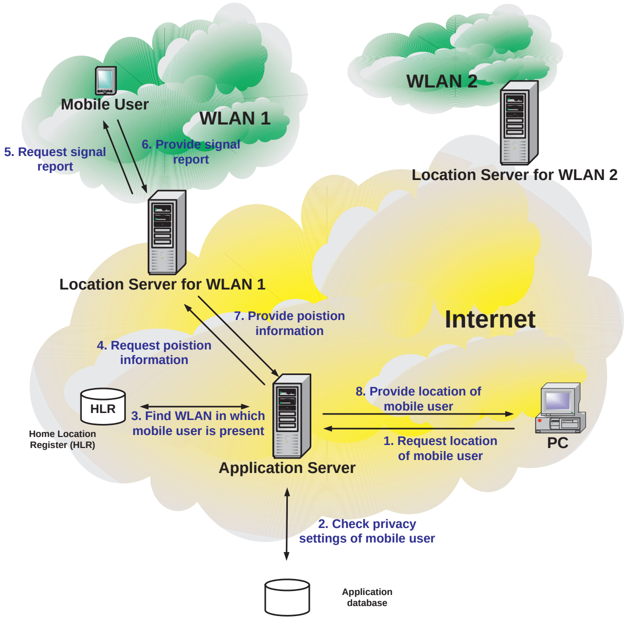


Figure D.6: A location service requested by a non-WLAN user over the internet.

This scenario is similar to the one in section D.3.3 except that here the service is requested by person who is not using a mobile device and is not on a WLAN. The PC user contacts the application server and requests the location of the mobile user. The application user checks the application database to see if the PC user is authorized. It finds that the PC user is authorized and so it looks up the HLR/VLR to find the WLAN in which the mobile user is present. The application server then contacts the corresponding location server and requests it to send the location of the mobile device. The location server contacts the mobile device for a signal report. Upon receiving the signal report the location server determines the position of the user and sends it to the application server. The application server then displays the information in a suitable format and sends it to the PC user.

D.4 Discussion

D.4.1 Mobility Management

Our mobility management scheme adopts the IS-41 standard [66] and uses a two level hierarchy for databases - the HLR, and the VLR. This entire database (HLR and VLRs) is commonly referred to as location-information-database (LID).

Updating the LID

As a mobile device moves across different WLANs it becomes necessary to know in which WLAN the mobile device is presently in. This would allow an application server to page the appropriate location server.

Different WLANs will typically be in seperate geographical areas and there will be no overlapping coverage. So when a user is leaving a WLAN, he will quite likely turn it off (or suspend) his mobile device. Later, when he arrives in a new WLAN he will turn it on again. We can therefore use the following broad guidelines:

1. Register the mobile device whenever it is turned on, if user has allowed so. Otherwise register the device when the user specifies.
2. Deregister when the mobile device is turned off.
3. If the mobile device is suspended, do nothing.

Figure D.7 shows in detail how the LIDs are updated when a mobile device enters a new WLAN. We assume that the mobile device was suspended before migrating to the new WLAN. Therefore it was not deregistered in its previous WLAN. If the mobile device had been shut down before migrating, the sequence of steps would have been shorter because step 4 in figure D.7 would not be necessary. For the present scenario, the tasks are executed in the following sequence:

1. The mobile device migrates from WLAN 1 to WLAN 2 (step 1 in figure D.7).
2. The mobile device contacts the VLR of WLAN 2 and registers itself (step 2 in figure D.7).
3. VLR 2 sends a location update to the HLR (step 3 in figure D.7).
4. The HLR finds that the mobile device was previously registered with VLR 1. It updates its entry and sends an update to VLR 1 (step 4 in figure D.7).

5. VLR 1 deregisters the mobile device.

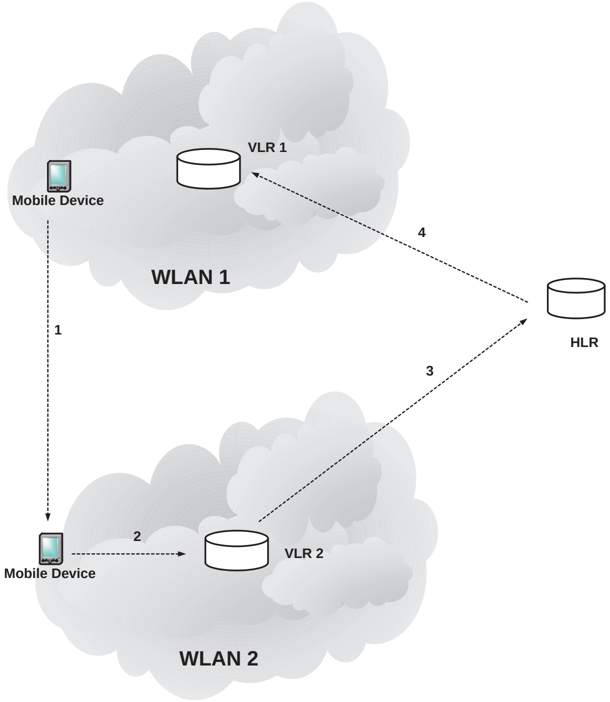


Figure D.7: Location update procedure for mobility management.

LID Architecture

When an application server needs to find the WLAN in which a mobile device is present, it will need to do a database lookup. As the number of mobile devices increases, it is vital to design the LID so that the lookup requires minimum time. Moreover, the LID should be fault tolerant.

Two common architectures for LIDs are *distributed* and *centralized* architecture. A centralized architecture is easier to implement and has better connection delay performance, since the location information can be accessed by a single query. However a centralized architecture is not fault tolerant and is difficult to scale. Also, as the number of mobile devices increases, it will increase the load on the LID and thus degrade the Quality of Service (QoS). A number of optimizations have been suggested to improve the performance of centralized LIDs. One suggestion is to have a dynamic hierarchical database architecture [67]. The proposed architecture is based on that of the IS-41 standard with the addition of a new level of databases called directory registers (DR's). Another suggestion is to have a user-profile replication scheme [68] so that a mobile device's location can be available locally. Querying the HLR is necessary only when the information is not available locally.

A distributed architecture, on the other hand, is more fault tolerant and better scalable than a centralized architecture, but suffers from delay during location lookup due to increased signaling and control traffic flow. Different optimization techniques have been suggested to improve the performance. Krishnamurthi et al [69] have proposed an algorithm for optimally balancing the load in a distributed architecture. Jain et al [70] have proposed using forwarding strategy to reduce network signaling and database loads in exchange for increased CPU processing and memory costs.

So, for larger groups of WLANs, a distributed architecture may prove more beneficial, while smaller groups of WLAN may get better performance out of a centralized LID architecture.

D.4.2 Authentication

Any distributed system is susceptible to a multitude of threats from malicious users. In general, there are two categories of threats. The first category, *host compromise*, involves disrupting the normal activity of a host or even taking control of it. These techniques are beyond our scope and we will not discuss it further in this paper. Interested readers are referred to [71] for an overview. The second category, *communication compromise*, involves threats associated with message communication. Threats in this category can be further subdivided into *passive* and *active* threats.

Passive threats involve eavesdropping of transmitted messages in order to extract information. We will not consider these type of threats in our discussion since they affect the security of the entire communication system, not just our tracking system.

Active threats involve the deliberate insertion of spurious messages to confuse the receiver. For example, in our proposed architecture, the mobile device scans the WLAN upon receiving a request from the location server. This introduces the possibility that a malicious user can flood an unsuspecting mobile device with spurious requests, causing it to scan the WLAN over and over again.

Traditional authorization methods such as public key cryptography can be used to identify unauthorized requests. In public key cryptography, there are two keys - an encryption key and a decryption key. The encryption key allows the sender to encrypt messages before sending it over the communication channel. The decryption key allows the receiver to recover the message from its encrypted form. The encryption key is made public while the decryption key is kept secret.

A widely used public key cryptosystem is the RSA [72]. In RSA, encryption-decryption key pairs satisfy the following commutative property:

$$\forall\ m \in M : \forall (k, k^{-1}) \in K_E \times K_D : \{\{m\}_{k^{-1}}\}_k = m \tag{D.1}$$

where M is the message space, and $K_E \times K_D$ is the set of encryption/decription key pairs. The representation $\{x\}_y$ is used to denote the encryption of message x if y is the encryption key. If y is the decryption key, it represents the decryption of message x. So if k is the public key, and k^{-1} is the secret key, then $\{m\}_{k^{-1}}$ can be used as the location server's signature on message m. This is because only the location server knows the secret key. This way the mobile device will know that the request is genuine.

D.4.3 User Privacy

A user's location information should be revealed based on two criteria: *user preferences* and *organizational preferences.*

User preferences refer to access control settings as specified by the user. The user may set preferences such as with options such as "allow none to track me", or "allow all to track me", or "allow only these to track me". User preferences are not location specific (e.g. *allow my spouse to track my location*) and should therefore be enforced by the application server at application level.

Organizational preferences refer to access control as specified by the organization (e.g., company, university) where the WLAN is installed. For example, a company might have the policy: “Do not reveal the people inside the meeting room”. Organizational preferences are location specific and should therefore be enforced by the location server.

D.5 Conclusion

In this chapter, we presented a centrally based, mobile device assisted architecture for WLAN-based tracking systems.

Each WLAN has a location server, and an optional application server. The location servers are entrusted with the task of locating the mobile devices, while application servers are responsible for providing location services. The mobile devices assist in tracking by scanning the WLAN and preparing a signal report that is sent to the location server. A group of WLANs can join together to provide ubiquitous location services across all these WLANs. Our architecture allows new WLANs to easily join an existing group.

The mobility management technique is based on the IS-41 standard and employs a two-level hierarchical database - a single HLR and multiple VLRs. The HLR is a global database that contains information about all the mobile devices from all the WLANs. The VLR is a local database that maintains information about mobile devices from a single WLAN.

The main benefits of the proposed architecture are:

- *Support for non-local roaming:* By having the tracking logic reside on a central server, users can roam seamlessly across different WLANs.
- *Support for non-local lookup:* The architecture is such that even remote users can avail of location services.
- *Less traffic disruption:* A mobile device assisted tracking system leads to lesser traffic disruption in the WLAN. Only the mobile device that is being tracked will have to suspend its normal traffic activity briefly.
- *Easy updating of location tracking technique:* The radio map is stored on the location server and therefore it is easier to make updates if the building layout changes. Also, it would be easier to change the tracking algorithm since the changes have to be made only at the server and the mobile users do not have to be involved.

Bibliography

[1] G. Chen and D. Kotz, “A survey of context-aware mobile computing research,” Tech. Rep. TR2000-381, Dartmouth Computer Science, 2000.

[2] P. Enge and P. Misra, “Special issue on gps: The global positioning system,” in *Proceedings of the IEEE*, pp. 3–172, January 1999.

[3] K. Gear. http://www.kanegear.com/.

[4] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,” *IEEE Computer*, pp. 57–61, August 2001.

[5] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, “The Anatomy of a Context-aware Application,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1999),*, (Seattle,Washington), pp. 59–68, ACM Press, August 1999.

[6] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-support System,” in *Proceedings of MOBICOM 2000*, (Boston, MA), pp. 32–43, ACM, ACM Press, August 2000.

[7] AeroScout. http://www.aeroscout.com/.

[8] "Pulson technology: Time modulated ultra wideband overview," tech. rep., Time Domain Corporation,, 7057 Old Madison Pike, Huntsville, AL 35806, 2001.

[9] J. Hightower, R. Want, and G. Borriello., "SpotON: An Indoor 3d Location Sensing Technology based on rf Signal Strength," Tech. Rep. UW-CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, Washington, February 2000.

[10] J. R. Beveridge, C. R. Graves, , and C. E. Lesher., "Local Search as a Tool for Horizon Line Matching," in *Image Understanding Workshop*, (Los Altos, CA), pp. 683–686, ARPA, Morgan Kaugmann, February 1996.

[11] T. Starner, B. Schiele, and A. Pentland, "Visual Context Awareness Via Wearable Computing," in *International Symposium on Wearable Computers*, (Pittsburgh, PA), pp. 50–57, IEEE Computer Society Press, October 1998.

[12] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proc IEEE INFOCOM 2000*, vol. 2, (Tel Aviv, Israel), pp. 775–784, March 2000.

[13] K. Hinckley and M. Sinclair, "Touch-sensing Input Devices," in *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI 1999)*, ACM, 1999.

[14] K. Partridge, L. Arnstein, G. Borriello, and T. Whitted, "Fast intrabody signaling." Demonstration at Wireless and Mobile Computer Systems and Ap-

plications (WMCSA), December 2000.

[15] R. Want, B. Schilit, N. Adams, R. Gold, K. Petersen, D. Goldberg, J. Ellis, and M. Weiser, *The Parctab Ubiquitous Computing Experiment*, ch. Mobile Computing, Chapter 2, pp. 45–101. No. 0-7923-9697-9, Kluwer Publishing, February 1997.

[16] R. Want and D. M. Russell, "Ubiquitous Electronic Tagging," *IEEE Distributed Systems Online*, vol. 1(2), September 2000.

[17] R. Want, A. Hopper, V.Falco, and J. Gibbons, "The active badge location system," in *ACM Transactions on Information Systems*, vol. 10 of *1*, pp. 91–102, January 1992.

[18] R. Azuma, "Tracking requirements for augmented reality," in *Communications of the ACM*, vol. 36 of *7*, pp. 50–51, July 1993.

[19] Ascension Technology Corporation. http://www.ascension-tech.com/.

[20] T. W. Christ and P. A. Godwin, "A prison guard duress alarm location system," in *Proc. IEEE International Carnahan Conference on Security Technology*, October 1993.

[21] P. Myllymäki, T. Roos, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to wlan user location estimation," in *Proceedings of the 3rd IEEE Workshop on Wireless Local Areas Networks*, vol. 7, (Boston, USA), September 2001.

[22] M. A. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh, “A probabilistic clustering-based indoor location determination system,” Tech. Rep. CS-TR-4350 and UMIACS-TR-2002-30, University of Maryland, March 2002.

[23] N. Networks. http://www.newburynetworks.com/.

[24] S. Y. Seidel and T. S. Rappaport, “914mhz path loss prediction models for indoor wireless communications in multifloored buildings,” in *IEEE Trans. Antennas Propagat.*, vol. 40, pp. 207–217, February 1992.

[25] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York: John-Wiley & Sons, 1973.

[26] T. Pavlidis, *Structural Pattern Recognition.* New York: Springer-Verlag, 1977.

[27] Milgram, Paul, and F. Kishino, “A taxonomy of mixed reality virtual displays,” in *IEICE Transactions on Information and Systems E77-D, 9*, pp. 1321–1329, September 1994.

[28] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, “Recent advances in augmented reality,” in *IEEE Computer Graphics and Applications*, pp. 34–47, December 2001.

[29] B. Thomas, B. Close, J. Donoghue, J. Squires, P. D. Bondi, M. Morris, and W. Piekarski, “Ar quake: An outdoor/indoor augmented reality first person application,” in *In 4th Int’l Symposium on Wearable Computers, Atlanta, Ga, USA*, pp. 139–146, October 2000.

[30] M. A. Livingston, L. J. Rosenblum, S. J. Julier, D. Brown, Y. Baillot, J. E. Swan, J. L. Gabbard, and D. Hix, "An augmented reality system for military operations in urban terrain," in *Proceedings of Interservice / Industry Training, Simulation & Education Conference (I/ITSEC)2002*, p. 89, Dec 2002.

[31] R. Azuma, "A survey of augmented reality," in *Presence, 6(4)*, pp. 355–385, 1997.

[32] ARToolkit. http://www.hitl.washington.edu/artoolkit/.

[33] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proc. IWAR*, pp. 85–94, October 1999.

[34] T. Hollerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway, "Exploring mars: Developing indoor and outdoor user interfaces to a mobile augmented reality system," in *Computer and Graphics*, pp. 779–785, Elsevier Publishers, Dec 1999.

[35] E. Bachmann, D. I., U. Usta, M. R., X. Yun, and M. Zyda, "Orientation tracking for humans and robots using inertial sensors," in *International Symposium on Computational Intelligence in Robotics and Automation (CIRA 99), Monterey, CA*, pp. 187–194, 1999.

[36] M. Ribo, P. Lang, H. Ganster, M. Brandner, C. Stock, and A. Pinz, "Outdoor ar tracking," in *CG and A 2002*, vol. 22, pp. 54–63, Nov 2002.

[37] J. Gausemeier, J. Fruend, C. Matysczok, B. Bruederlin, and D. Beier, "Development of a real time image based object recognition method for mobile ar-devices," in *2nd international conference on Computer graphics, Virtual Reality, visualisation and interaction in Africa*, pp. 133–139, 2003.

[38] C. Matysczok, C. Reimann, and W. Rosenbach, "Augmented reality pda," in *The First IEEE International Augmented Reality Toolkit Workshop, Darmstadt*, 2002.

[39] H. Regenbrecht and R. Specht, "A mobile passive augmented reality device, mpard," in *Proceedings to ISAR*, October 2000.

[40] D. Wagner and D. Schmalstieg, "First steps towards handheld augmented reality," in *Proceedings of the 7th International Conference on Wearable Computers, White Plains, NY, USA*, pp. 21–23, 2003.

[41] T. Pyssysalo, T. Repo, T. Turunen, T. Lankila, and J. Roing, "Cyphone: Bringing augmented reality to next generation mobile phones," in *Proceedings of DARE 2000 on Designing augmented reality environments*, pp. 11–21, ACM Press New York, USA, 2000.

[42] http://www.bluetooth.org.

[43] 3GSystems. http://www.umtsworld.com/technology/overview.htm.

[44] http://www.jpeg.org.

[45] http://www.realspace.com.sg.

[46] J. Inouye, "Logical link control and adaptation protocol." http://www.bluetooth.com, 2001.

[47] B. J. Rhodes, "The wearable remembrance agent: a system for augmented memory," in *Wearable Computers, Digest of Papers, 1st International Symposium on Wearable Computers*, (Cambridge,Mass, USA,), pp. 123–128, 13-14 October 1997.

[48] T. Starner, D. Kirsch, and S. Assefa, "The locust swarm: an environmentally-powered, networkless location and messaging system," in *Wearable Computers, Digest of Papers, 1st International Symposium on Wearable Computers*, (Cambridge,Mass, USA,), pp. 169–170, 13-14 October 1997.

[49] M. Beigl, "Memoclip: A location based remembrance applicance," in *2nd International Symposium on Handheld and Ubiquitous Computing (HUC2000)*, vol. 4 of *2000 and Personal Technologies*, (Bristol, UK), pp. 230–234, Springer Press, Sept. 25-27 2000.

[50] M. Hazas and A. Ward, "A high performance privacy-oriented location system," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003)*, pp. 216 – 223, 23-26 March 2003.

[51] A. J. Sellen and R. H. Harper, *The Myth of the Paperless Office*. Cambridge, Massachusetts: The MIT Press, 2002.

[52] http://www.opengl.org.

[53] SundialSoft, "Pocketgl." http://www.sundialsoft.freeserve.co.uk/pgl.htm.

[54] Bellard and Fabrice, "Tinygl." http://fabrice.bellard.free.fr/TinyGL.

[55] http://www.umtsworld.com/technology/wcdma.htm.

[56] www.quake2.com.

[57] www.unrealtournament.com.

[58] www.n-gage.com.

[59] www.tombraider.com.

[60] M. Flintham, R. Anastasi, S. Benford, T. Hemmings, A. Crabtree, C. Greenhalgh, T. Rodden, N. Tandavanitj, M. Adams, and J. Row-Farr, "Where on-line meets on-the-streets: Experiences with mobile mixed reality games," in *Conference on Human Factors in Computing Systems*, (Ft. Lauderdale, Florida, USA), pp. 569 – 576, ACM Press New York, NY, USA, 2003.

[61] B. Thomas, B. Close, J. Donoghue, J.Squires, P.D.Bondi, and W.Piekarski, "First person indoor/outdoor augmented reality application:arquake," *Personal and Ubiquitous Computing*, vol. 6, no. 1, pp. 75–86, 2002.

[62] S.Bjork, J. Falk, R. Hansson, and P. Ljungstrand, "Pirates! - using physical world as a game board," in *Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction*, (Tokyo, Japan), 2001.

[63] www.playstation.com.

[64] http://everquest2.station.sony.com.

[65] M. Weiser, "The computer for the 21st century," *Scientific American*, pp. 94–100, September 1991.

[66] "Cellular radio-telecommunications intersystem operation: Automatic roaming," Tech. Rep. IS-41.3-B EIA/TIA, EIA/TIA, 1991.

[67] J. S. M. Ho and I. F. Akyildiz, "Dynamic hierarchical database architecture for location management in pcs networks," in *IEEE/ACM Trans. Networking*, vol. 5, pp. 646–661, October 1997.

[68] N. Shivakumar and J. Widom, "User profile replication for faster location lookup in mobile environments," in *ACM/IEEE MOBICOM 95*, pp. 161–169.

[69] G. Krishnamurthi, S. Chessa, and A. K. Somani, "Optimal replication of location information in mobile networks," in *ICC '99, IEEE International Conference on Communications*, vol. 3, pp. 1768 – 1772, IEEE, 6-10 June 1999.

[70] R. Jain and Y.-B. Lin, "An auxiliary user location strategy employing forwarding pointers to reduce network impacts of pcs," in *1995 IEEE International Conference on Gateway to Globalization*, vol. 2, (Seattle), pp. 740–744, IEEE, 18-22 June 1995.

[71] D. E. Denning, *Cryptography and Data Security.* Reading,Mass.: Addison-Wesley Publishing Co., 1982.

[72] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. ACM*, vol. 21, no. 2, pp. 120–126, 1978.