Uniwide WiFi Based Positioning System

William Ching[1] Rue Jing Teh[1], Binghao Li[2], Chris Rizos[2]
[1]*School of Computer Science and Engineering, University of New South Wales*
[2]*School of Surveying and Spatial Information Systems, University of New South Wales*
wching@cse.unsw.edu.au; rueteh@cse.unsw.edu.au; binghao.li@unsw.edu.au

Abstract

Millions of people travel between countries and regions on a daily basis being exposed to unfamiliar environments. Knowing where you are and how to get to places within a restricted time can make people's lives easier. Today, WiFi access points (APs) are common everywhere, especially on university campuses, in hotels, hospitals, shopping centres, and city central business districts. This project, dubbed WiPos (WiFi-Positioning), had the objective to develop a positioning system using WiFi APs deployed across a university to locate one's position in a building under conditions where GPS could not be used. When relatively accurate user position is available, location based services (LBS) can be provided to users, including timetable of a lecture room, location of the nearest vending machine, and so on. This project involved developing a server and a client (running on the Android platform) to handle positioning and LBS transactions. Testing has shown that the university's WiFi network is sufficient to provide 'room to room' accuracy.

1. Introduction

GPS is no doubt today's most popular positioning system, widely used in our daily life. Handheld GPS units (including those within mobilephones) have a horizontal accuracy of about 10 metres in open environments [1]. However, GPS does not work well within buildings due to obstructions to the line of sight with the satellites [2].

Cell towers are currently used to provide an estimate of location for devices such as mobilephones. The technique primarily used is known as 'Cell ID'. Position is calculated based on which tower the mobile device is connected to. The accuracy of this technique is dependent on the density of cell towers - a higher density means a higher accuracy, as the service area of the cell tower is smaller. In general the cell tower density is sparse, hence the accuracy of this technique is quite low - being of the order of hundred of metres to several kilometres. Other techniques, such as time of arrival, time difference of arrival and angle of arrival may also, in principle, be used in mobilephone network positioning. However, the indoor positioning accuracy ranges from several tens of metres to hundreds of metres because of non-line-of-sight error and other errors [3, 4]. These techniques are therefore not suitable for the finer positioning requirements within a building.

WiFi is an attractive positioning technology due to the widely deployed WiFi access points (APs) and the growing number of WiFi-enabled mobile devices on the market. WiFi APs can be found almost everywhere - on university campuses, in hotels, shopping centres, etc. The global WiFi AP shipments are forecast to exceed 70 million by 2010. Shipments of WiFi-enabled mobilephones will double in number by the end of 2010, compared to January 2008 [5]. There are many WiFi APs located throughout the University of New South Wales (UNSW) that form the University Wide WiFi Network known as 'Uniwide'. The density of UNSW WiFi APs is far greater than the density of mobilephone cell towers in the area. Hence WiFi positioning, even uses a simple "Cell ID-like" approach, offers a much higher accuracy than that of indoor GPS or the mobile network. This higher density is required in order to provide sufficient WiFi coverage to service the entire (or most) university since WiFi signals only extend at most about 100 metres, and often much less.

WiFi was therefore chosen over other techniques for UNSW indoor campus positioning. The system is known as 'WiPos' (WiFi-Positioning). The aim of this project was to provide UNSW students with location based services such as finding a lecture room and related timetable, finding the nearest vending machine, etc. For most applications room level accuracy is good enough. Hence in this paper, the unit of accuracy will be the room rather than metres.

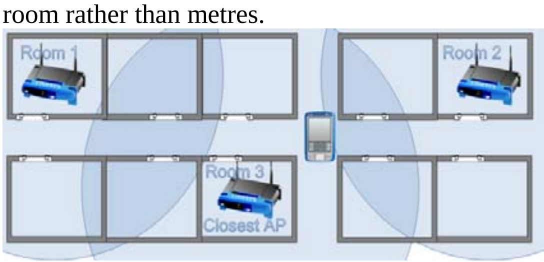

Figure 1. WiFi positioning - Cell ID

2. WiFi positioning technology

Cell ID, Triangulation, Trilateration and Fingerprinting are the most popular techniques used in WiFi positioning.

2.1 Cell ID

Cell ID WiFi-based positioning reports back the closest AP as the location of the mobile client/user. Cell ID positioning can be extended by the user reporting back areas where it can 'see' some combination of APs. This requires the mapping of the coverage area to determine which areas can detect which APs (see Figure 1). Similar to Cell ID in mobile network positioning, the accuracy of WiFi-based Cell ID positioning depends on the density of the APs and the size of each AP's cell coverage.

2.2 Triangulation

Given the coordinates of APs, and the angle between client/AP and North, Triangulation can give a positioning result by calculating the location of the client. However, Triangulation requires a directional antenna, which is expensive and not available in mobilephone devices. Furthermore, the NLOS error is huge in indoor environments which degrades the angle measurements, and impacts on the positioning result significantly [6, 7].

2.3 Trilateration

Ideally, given the coordinates of APs and the distance from APs, Trilateration (or Multilateration) can also determine an accurate result, as shown in Figure 2. However, it is difficult to accurately measure the distance from the client/user to each AP. Using the time taken to send a packet from the client to the AP will not yield an accurate result as there are many factors that will affect the time for a packet to arrive. In addition, to obtain a reasonable level of precision, the device must operate at a minimum of 1GHz.

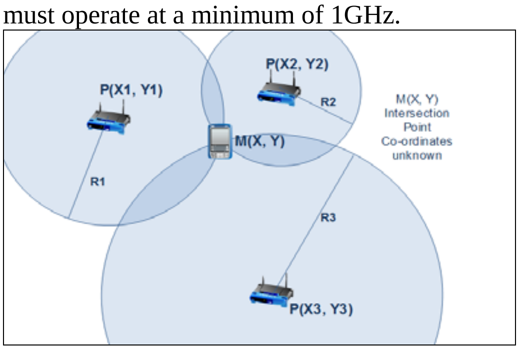

Figure 2. WiFi positioning - Trilateration

Using signal strength also yields a relatively inaccurate distance as there are no ideal models for estimating the distance between client/user and an AP using signal strength alone. This is because the environment through which the signals propagate is so variable, different from location to location due to walls and objects blocking signals [8, 9].

2.3 Fingerprinting

Each location has a unique set of detectable APs and associated signal strengths. This set is known as a 'fingerprint'. Fingerprinting involves surveying the coverage area and recording the WiFi fingerprints across the area and storing this data in a database (see Figure 3). Finding the location of a client or user involves measuring the current fingerprint at the unknown location and performing a comparison procedure against the fingerprint data stored in the database in order to find a match (see Figure 4). The matched fingerprint's position will be the estimated location of the client [9, 10].

This technique does not require knowledge of the positions of the APs (needed in the Cell ID, Triangulation and Trilateration techniques). This technique also does not require an estimation of distance between AP and client, and therefore static objects in the environment (that affect signal strength) do not affect the system. However, if such an object is removed, or there are significant structural modifications to a building, the system may give degraded results. The coverage area must then be re-surveyed [8, 11].

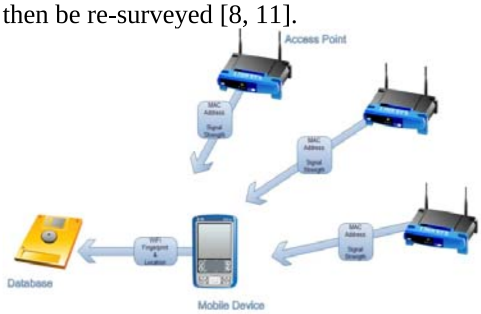

Figure 3. Fingerprinting survey

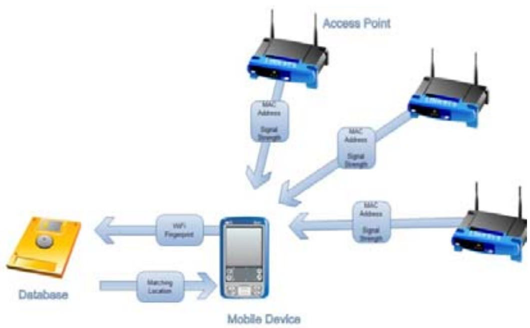

Figure 4. Fingerprinting positioning

2.3.1 Survey. The Survey stage involves creating a database of WiFi 'fingerprints' by physically surveying the signal environment of the coverage area (Figure 3).

The direction of the device influences the system accuracy. As discussed in [9, 12, 13] the direction in which the device points affects the signal strength measurements between APs (refer to Figure 5).

A solution to this problem is to add additional fingerprints for each direction during the Survey stage. By considering the direction of the device it is possible to have up to 95% accuracy with the system, as opposed to 55% when pointing direction is not considered [12].

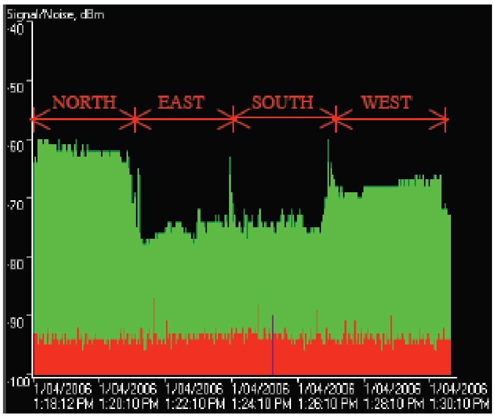

Figure 5. Orientation's affect on signal strength

The environment changes over time due to room renovations, movement of large office furniture, deployment of new APs or removal of exisiting APs. Hence the fingerprint database must be updated - this is a major drawback of the Fingerprinting technique. This can be done by the administrators of the system, however this is rather labour intensive. An alternative approach is to allow the users to update the database. Users can use signal survey client software on their mobile device, and measure add fingerprints to the database - but 'trusting' the users is an issue. For example, a user may enter an incorrect location name, which will cause problems in the database. It is also possible for users to merely provide feedback to the system administrators. If the user is located correctly, the user can give positive feedback; the feedback can be confirmed by counting the number of positive/negative feedback messages in that specific area.

Another problem that exists with the Fingerprinting technique is the constantly changing environment caused by the movement of people. WiFi signal strengths are changed by small amounts constantly. This is a problem when surveying the area as initial data may become inaccurate very quickly, or the survey data was not truly representative of the average 'traffic' in a coverage area. Regular updates to the database discussed above may help to reduce these errors.

2.3.2 Positioning. The Positioning stage involves the mobile device querying the database of fingerprints for a match with the current fingerprint – in order to obtain a position, as shown in Figure 4.

As discussed in the previous section, a changing environment would affect the accuracy of the fingerprint data in the database. This problem also exists in the Positioning stage as the current fingerprint of the client's location may be inaccurate, and poor results may be obtained when trying to find a match in the database. The orientation of the client also affects the signal strengths received by the user's device. This is an issue in the Positioning stage, and may affect the performance of the matching algorithms in the database. In the Uniwide WiPos design this problem is handled at the Survey stage.

The Positioning stage involves matching fingerprints - the current fingerprint of the client against a database of fingerprints. There are two way of comparing fingerprints: the deterministic approach and the probabilistic approach.

The deterministic approach calculates the difference between the reference fingerprint and each fingerprint in the database [8]. The fingerprint in the database with the smallest difference from the reference fingerprint is selected to be the closest matching fingerprint. This fingerprint's location coordinates will be returned as the user's position. From [10] the deterministic approach offers a possible 1-3 metre level of accuracy.

The probabilistic approach involves calculating the probability that a user is at a specific location. The probabilistic approach requires the Survey stage to sample more of one reference point so a distribution of signal strengths can be generated [11, 14, 15]. This possibly requires more sampling in the Survey stage. During positioning, the system calculates the probability of the user being at each reference point recorded in the database. The one with the highest probability is assumed to be the user's location.

The authors have chosen the deterministic approach as it is easier to implement while offering only a small impact in performance (against the probabilistic approach). The deterministic approach is good enough for 'room to room' positioning accuracy (assuming each room is at least 3 metres across).

3. System design and development

Currently, the UNSW is a map-based positioning system. This system consists of map boards, maps online and kiosks around the university to show the location of specific buildings and facilities. However, there are several disadvantages of such a system:

- Have to locate the map board itself,
- requires the basic geography map reading skills,
- does not locate rooms within a building, and
- not available everywhere.

One of the aims of this project was to develop a replacement system.

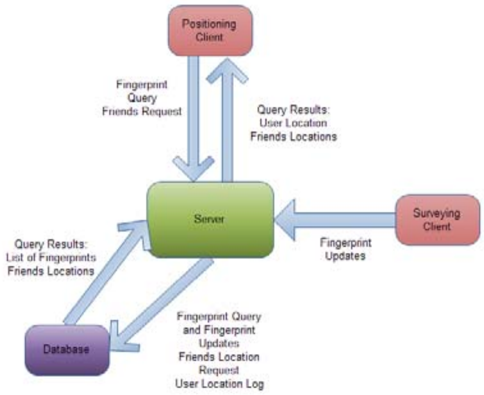

Figure 6. System block diagram

As shown in Figure 6, the system consists of three main components:

- Surveying client
- Positioning client
- Server and Database

Four reference points in each room (assumed rectangular) were chosen - each reference point was about 1 metre from each corner of the room. Non-rectangular rooms had reference points measured at each corner of the room. Long corridors were divided into sections, each section treated as a separate 'room'.

Figure 7. T-Mobile G1 with mock user interface

3.1 Client

The positioning client was the Android platform, in particular the T-Mobile G1 phone shown in Figure 7. This platform was chosen for several reasons. Firstly, G1 supports WiFi, but it also incorporates 2G and 3G mobile connectivity which can be used to communicate with the server when the WiFi signal is too weak to be used for accessing the internet. Secondly, the Android platform provides a convenient API for the wireless adapter - which means it is easy to gather AP details. Thirdly, its display can show large graphical and customisable elements - the touch screen makes navigation using a map more flexible. Lastly, it has built-in GPS and a magnetometer which may be used in future work. Figure 8 gives the positioning client block diagram.

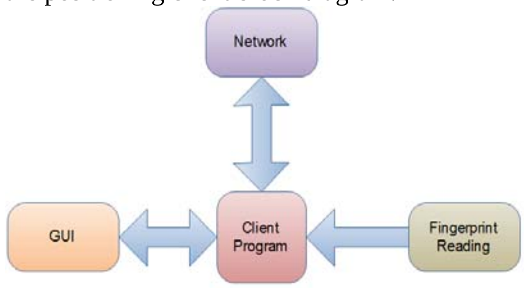

Figure 8. Positioning client block diagram

The Graphic User Interface (GUI) of the system displays a map with the following features (refer to Figure 9):

- The user can scroll the map in all directions.
- The user can zoom in/out.
- The user can switch between a main campus map, the user's current location floor map, and the user's searched location map.

Current location and searched location is plotted on the main map. If the user switches to a floor map, those locations are plotted on the floor map.

In the main mode, the user is presented with zoom buttons, a locate button (centre) and a switch map button (right arrow). The "buddy" icon displayed here is merely a remnant of a proposed friends system, not implemented due to time constraints and has remained for possible future application development.

Figure 9. Search, Survey UI (left) and Map UI (right)

A search mode is provided for searching for locations. The user can switch to this mode via the menu button. This mode permits the user to select a building, then to select a floor in the building, and then to type in the room name/number. The user then presses the Search key, which invokes a search for the selected room's coordinates and map to be displayed. The user may press the Find key to locate the device's current location. The UI then automatically switches to the main map mode. The Scan button performs a scan of APs in the area and displays them on the screen. The Save button saves the current WiFi fingerprint with details of the building, level and room selected.

When the project was started there was no Java or Android API that supported customised map features. The Google Maps API did not support this feature either. Hence the authors have had to develop their own system for dealing with maps.

A dark maps theme is chosen to improve visibility of text as shown above. User location and searched location are displayed on the map as a small bubble. These bubbles and the associated text are colour-coded. Green is used to represent a searched location and red is used to represent the user's location.

3.2 Server

WiPos incorporates several modules that support the client functionality. These include the server database, the network connection between the system and the network cloud, a console user interface for administrators and authorised parties to access the server, and finally the fingerprint matching system that performs the positioning calculations. Figure 10 is a diagram of the server system, which maps out the design structure of all components.

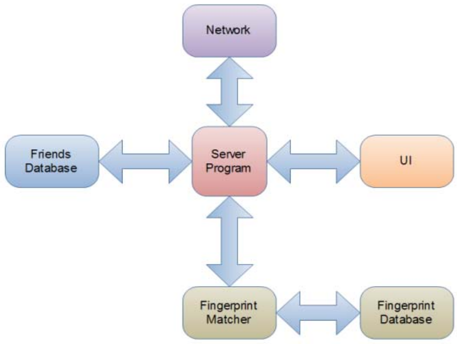

Figure 10. Server block diagram

The network component joins the server to the network cloud (Uniwide network system) to allow data information transfer to and from the server. The server program is the main controller system for all other components. At this component several modules are run concurrently. The server itself is not a single main function, but rather consists of sub-modules (i.e. classes).

The WiPos server system was designed to support multiple clients' requests simultaneously. By connecting the clients via a Java socket connection, the server is able to listen on a fixed port to receive requests, and create a thread that is treated as a separate process to execute concurrently with other requested processes. Each request is concerned with one of the following four functions:

- **SAVE:** This function takes in the details of the location (i.e. room name/number, level and building name) and a list of APs that are detected by the client program to create a fingerprint that can be saved to the database. A 'fingerprint' can be successfully saved if only the location details to be saved already exist in the database. (This requirement originated from being unable to find an existing Java API such as Google Maps that supports the customised maps functionality.)
- **SEARCH:** The Search function allows users to search for a location or a destination with the given location details (such as building name, level, room name, name of map, and x and y coordinates for the customised map). This function can be broken down into three categories:
  - **Search a room:** the user only knows the room name. However, if the room name entered is not unique (i.e. the name of the room may be the same but reside in two different buildings), the location "not found" will be returned to the user. A solution could be to have the server return both building names and ask the user to identify the correct building. However,

this solution has been left for future development.
    - **Search a building:** the user only knows the name of the building.
    - **Search a room and building:** the user knows both the room and the building name.
- **FIND:** The Find function locates the user's current location. A list of APs are passed in and matched with the database for the closest fingerprint found (nearest neighbour), then the user's location details are sent back to the client's application.

Each request received is then parsed, invoked and returned according to the function identified.

The following software was used to implement the WiPos server:

- **Eclipse**: a multi-language software development environment comprising an IDE and a plug-in system to run and maintain the server.
- **PostgreSQL:** an open source object-relational database system.
- **pgAdmin:** an open source administration and development platform for PostgreSQL.
- **JDBC driver:** an API for the Java programming language that defines how a client may access the WiPos database created in PostgreSQL. JDBC API also provides the means to query and update the database.

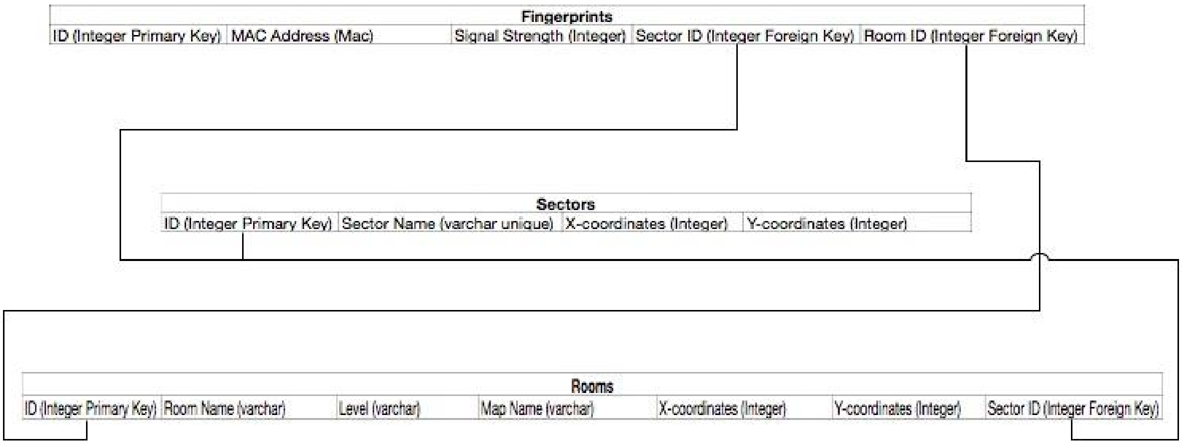

Figure 11. Database entity relationship diagram

3.3 Database

The final database design has been redesigned to a more simplified schema to accommodate the needs of the client system.

Each relational table (i.e. fingerprint, sectors and rooms) was designed in such a way that the deletion of an entry triggers a cascading deletion on all related entries in other tables (refer to Figure 11). A deletion may happen in one of the following scenarios:

- If an AP has been removed from the building.
- If a building has been demolished.
- If the university wish to keep a particular room private.

The WiPos database uses several tools and methods to access the data stored in the PostgreSQL database. For the server to obtain permission to access the database the following procedures are followed:

1. The server program creates a new DBConnectionFactory object that searches for the Java database connectivity (JDBC) driver and establishes a connection to the PostgreSQL database through the JDBC API. This connection is deemed to be successful only if the correct Uniform Resource Locator (URL), username and password to the database is provided.
2. Once the connection to the database is established a Structured Query Language (SQL) statement is created to query the database. This prepared statement is executed and sent to the database. When the query reaches the database, the PostgreSQL database system executes the query request to return the results of the query.
3. After all queries are executed the connection to the database is closed.

Maintenance of the database is done through a program called pgAdmin. This application allowed the authors to develop the database design in SQL and directly communicate with the database server without additional drivers required. By using pgAdmin it is possible to directly access and delete any information within the database upon request. Information would only be deleted from the database if one of the deletion requirements is satisfied. Currently, WiPos database maintenance is carried out manually. Since the server was designed for testing purposes, future applications can be developed to provide a more user-friendly interface to interact directly with the database.

The WiPos database and server is not in any way protected against hacking - anti-hacking features are left for future development. In addition to security, a database maintenance program can also be incorporated to delete any APs that are prominent to the rest of the data. Prominent APs relate to fingerprint sets that contain a noticeable AP which has a much lower signal strength to the rest. Thus, by extracting these APs, it may lead to an improvement in positioning accuracy.

3.4. Localising search

This algorithm is embedded within the server program's Find function to make queries on the database, and to execute a finer search on results from a database query. The positioning algorithm consists of three steps:

1. The algorithm first makes a query to the database retrieving a list of fingerprints. The criterium for selecting the set of fingerprints is that the AP has the highest signal strength is the same as the strongest AP received from the client. This reduces the number of fingerprints for a finer search. However, if a temporary AP is placed close to the device, this step fails. Hence the algorithm only looks for an AP with the SSID of Uniwide and other registered APs.
2. The algorithm then retrieves all the fingerprints from the database with the most matches of APs with the fingerprint from the client. This reduces the number of fingerprints for a finer search. The Manhattan distance between a database fingerprint and the client fingerprint is calculated.
3. The fingerprint from the database with the smallest Manhattan distance from the client fingerprint is taken to be the matching fingerprint. Calculating the Manhattan distance involves looking for matching MAC addresses between two fingerprints and finding the difference between the two signal strengths corresponding to the pair of MAC addresses.
4. The fingerprint (location) is returned to the client.

3.5 Communication and other issues

Socket programming is used to connect client to server. This approach assumes that the client is connected to the internet via Uniwide or any other means (such as the mobilephone network), as the server is connected to the internet.

There are two fingerprint data structures in the system, the client-to-server data structure and the database fingerprint data structure. The client-to-server data structure consists of only a list of APs and their associated signal strengths. The database fingerprint data structure consists of a list of APs, their associated signal strengths as well as the location name of the fingerprint.

The following formats are utilised for communicating with the server:

- Find – Sends fingerprint to server, waits for reply
  - FND=<MAC Address>_<Signal Strength>;<MAC Address>_<Signal Strength>;
- Search – Sends Details in drop down boxes and waits for reply
  - SCH=<Building Name>;<Level Name>;<Room Name>
- Save – Sends details in drop down boxes and fingerprint to server
  - SAV=<Building Name>;<Level Name>;<Room Name>=<MAC Address>_<Signal Strength>;<MAC Address>_<Signal Strength>;

The following formats are used for returning data to the client:

- Used for both FND reply and SCH reply
  - LOC=Building{x,y};Level;Room{x,y};mapname
  - If building / room not found protocol: LOC=BLD/NA;L/NA;RM/NA
- Reply to Save
  - ADDED – Fingerprint was saved to the database
  - NOTADDED – Where there was an error in saving the fingerprint to the database

Maps are stored on the client device to save bandwidth - sending the maps wirelessly from the server to the client. Since the maps are unlikely to change very much over many years, it is feasible to notify users of map updates and have them download the maps when updates are available.

The Android platform has a limit on memory usage which affects the size of the map being displayed. This limit allows about 1.3 mega-pixel for a displayed image before the program crashes. Due to this limitation, the maps on the client are less than or equal to 1.3 mega-pixel in resolution. This problem can be resolved by tiling each map.

The fingerprint database is stored on the server. When there are multiple devices and the database must be updated, then updating a central database is much easier than updating every database stored on client devices. However this does require a connection between the client and server to perform positioning - but the benefits of having a convenient way of updating the database outweigh the complexity of needing a connection.

Currently map coordinates are stored on the server and sent to the client. Ideally the map coordinates should be stored with the maps. Future versions of the system will have coordinates with the maps.

For the GUI to plot a marker showing where the user is located, and where a searched location is, there must be some database storing information on where on a map is a room located. This data is stored on the server.

4 Tests and results

Intensive tests had been carried out to evaluate the system. The tests can be classified into two categories: system performance tests and feature testing.

4.1 System performance tests

This series of tests have tested the performance of the system, including testing for accuracy.

Level differentiation testing

This test investigated how well the system could determine which level the user is on within a building. This test was carried out in the following steps:

1. Select an area on one level
2. Invoke *Find* function several times and record the locations
3. Move to an area on a level above or below the current level
4. Invoke *Find* function several times and record the locations

Result interpretation:

- For each location calculate the number of times the system gives the correct level. This indicates how accurate the system is at level differentiation.
- Calculate the median number of levels away from the correct level the system's returned level is. This shows the actual accuracy of the system.

Assumption:

- The surveying data of two or more levels in a building are available.

Results Summary

The test was carried out in the Electrical Engineering Building on Level 3 and Level 4 at 4 PM on Friday 9th October 2009. Student population was low.

The level differentiation tests indicate that the system is capable of differentiating between levels: the system was 100% accurate when the client was not in a stairwell. However, when the client was in the stairwell, the reported location may jump between levels, only 71% of the locations were correct.

Basic accuracy testing

This test determined how accurate the system positions the client device on a level in a building, and how well the system distinguishes the client moving between rooms. The following was the test procedure:

1. Select five test rooms in a building: two lecture theatres, two tutorial rooms and one study room.
2. For each room, invoke three or more *Find* functions at three or more points inside the test room. Record the results.
3. For each test room, invoke three or more *Find* functions at three or more points outside, but next to the test room. Record the results.

Result interpretation:

- For each test room calculate the percentage of correct positions. This indicates how accurate the system is for room-level positioning.
- Calculate the number of rooms away from the real location.

Assumptions:

- The entire, or most of the, building/floor used for testing has been surveyed.
- Server is running.

Results Summary

The test was carried out in Level 1 of the Australian School of Business Building. The area was surveyed at two different times. During each survey, the student levels were relatively low - there were students studying in some open areas and there were lessons being conducted in the lecture theatres. Each room that could be accessed without interrupting or interfering with other people was surveyed. Four reference points per study room and six reference points per lecture theatre and tutorial room were surveyed.

The accuracy test was carried out on Friday 9th October 2009. The obtained results indicate that the system is capable of positioning the client device to within 2 rooms away from the actual (true) location. The system usually positioned the client device to the correct room (about 46% of all the tests) or to a room next to the correct room (98%). Rarely did the system report the client device to be two rooms away from the real location. This result was obtained in the case of a tutorial room - could be explained by the small size of the tutorial rooms and their close proximity.

If the accuracy in metres is considered instead, the system can give 5-10 metres (about the size of a room) accuracy, which is better than the indoor positioning accuracy of GPS [2]. The system can be improved by using finer surveying techniques such as considering orientation and measuring more reference points, or using more sophisticated algorithms.

Changing environment testing

This tested the system at different times of the day, checking room to room accuracy with varying levels of people in the area. Environments to test include night time environment (few people) which had been introduced previously, between class hours when people are moving around and during class hours (many people). The following was the test procedure:

1. Selecting a time of day and area with many people.
2. Invoke three or more *Find* functions at three or more points in the area and record the results.
3. Wait till the lecture is ended and students are leaving and moving to another class.
4. Invoke three or more Find functions at three or more points in the area and record the results.

Result interpretation:

- For the room calculate the number of times the system positions correctly. This indicates how accurate the system is at providing locations in highly populated areas.
- Calculate the median number of rooms away from the actual location is the system's returned position. The median shows the accuracy of the system in highly populated areas.

Assumption:

- The same as basic accuracy testing.

Results Summary

This test was conducted on a working day at 2:30 PM when there were many people in the area but low movement (during the class hours), and heavy traffic at 3 PM when students were moving from lecture to lecture.

The system behaved significantly differently compared to the basic tests results. In areas with a high population the positioning system performed poorly. When there were many people in the area but not moving around much, the system performed not as well as when there were low numbers of people. The tests showed that the system accuracy was 47% for exact room accuracy and 87% for next room accuracy during times when there were many lectures and tutorials. It was even worse when there was heavy traffic in the environment - the accuracy was as low as 32% for exact room accuracy and 63% for next room accuracy.

Obviously the change of the environment has a big impact on the positioning performance of the WiFi Fingerprinting technique. It is one of the significant problems of this technology. Further investigation will be conducted.

4.2 Feature testing

This test confirmed the correctness of the search function. This test examined the server connectivity, server's search algorithms and map coordinate correctness. This test used the following procedure:

1. Start client program.
2. Select search function.
3. Select a Building and a Level.
4. Type in a room number/name, select from available list.
5. Press search button.

Result interpretation:

- The system should switch to map mode and display the correct level map.
- The system should display the searched location marker in the correct position on the map.
- The main map should display the searched location's building correctly.

Assumptions:

- Server is running.
- There are building data with level data and room data. Building and room data include map names and map coordinates.

Results Summary

Tests on the search function indicate that this feature was working correctly. During the Survey stage, the search function had been used multiple times to find out where on the map the device was and what room to attach to each 'fingerprint'.

5. Conclusions

The WiPos system described here shows that an existing WiFi network such as Uniwide in UNSW could be used for the positioning of users throughout most of the university. The use of the WiFi Fingerprinting technique provides a level of accuracy sufficient to position the client device correctly within a room most of the time. The error is estimated to be approximately 5- 10 metres.

Since there is a constantly changing environment, a probabilistic WiFi fingerprinting algorithm might be more suited when used with Uniwide. Future investigations will be carried out.

Allowing users to contribute to the system to provide reference points to the system can increase the accuracy of the system. This user contribution is very useful if a probabilistic algorithm is used.

Integration with cell tower and GPS is left for future development.

6. References

[1] C. Tiberius, "Standard Positioning Service-Handheld GPS Receiver Accuracy," *GPS World,* pp. 46-51, 2003.

[2] B. Li, P. Mumford, A. G. Dempster and C. Rizos, "Secure User Plan Location (SUPL): concept and performance," *GPS Solutions,* vol. 14, pp. 153, 2010.

[3] C. Drane, M. Macnaughtan and C. Scott, "Positioning GSM telephones," *IEEE Communications Magazine,* vol. 36, pp. 46-54, 1998.

[4] H. Laitinen, S. Ahonen, S. Kyriazakos, J. Lahteenmaki, R. Menolascino and S. Parkkila, "*Cellular location technology*," 2000.

[5] ABI Research, "Wi-fi IC market share analysis and forecasts," 2009.

[6] E. Furey, K. Curran and P. Mc Kevitt, "HABITS: A History Aware Based Wi-Fi Indoor Tracking System," *PGNET 2008,* 2008.

[7] D. R. Hill, *A History of Engineering in Classical and Medieval Times.* Routledge, 1996,

[8] B. Li, Y. Wang, H. K. Lee, A. Dempster and C. Rizos, "Method for yielding a database of location fingerprints in WLAN," *IEE Proceedings-Communications,* vol. 152, pp. 580-586, 2005.

[9] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *IEEE Infocom*, 2000, pp. 775-784.

[10] B. Li, J. Salter, A. G. Dempster and C. Rizos, "Indoor positioning techniques based on wireless LAN," in *First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications, Sydney, Australia,* 2006, pp. 13-16.

[11] M. Youssef and A. Agrawala, "The Horus location determination system," *Wireless Networks*, vol. 14, pp. 357-374, 2008.

[12] B. Li, J. Kam, J. Lui and A. G. Dempster, "Use of directional information in wireless LAN based indoor positioning," in *Symp. on GPS/GNSS (IGNSS2007), Sydney, Australia*, 2007,

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

[14] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki and D. S. Wallach, "Robotics-based location sensing using wireless ethernet," *Wireless Networks*, vol. 11, pp. 189-204, 2005.

[15] J. Salter, B. Li, D. Woo, A. G. Dempster and C. Rizos, "802.11 positioning in the home," in *5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008,* 2008, pp. 598-602.