# Database Updating Through User Feedback in Fingerprint-Based Wi-Fi Location Systems

Thomas Gallagher, Binghao Li, Andrew G. Dempster, and Chris Rizos
School of Surveying and Spatial Information Systems
University of New South Wales
Sydney, Australia
t.gallagher@unsw.edu.au

*Abstract*—**Wi-Fi fingerprinting is a technique which can provide location in GPS-denied environments, relying exclusively on Wi-Fi signals. It first requires the construction of a database of "fingerprints", i.e. signal strengths from different access points (APs) at different reference points in the desired coverage area. The location of the device is then obtained by measuring the signal strengths at its location, and comparing it with the different reference fingerprints in the database. The main disadvantage of this technique is the labour required to build and maintain the fingerprints database, which has to be rebuilt every time a significant change in the wireless environment occurs, such as installation or removal of new APs, changes in the layout of a building, etc. This paper investigates a new method to utilise user feedback as a way of monitoring changes in the wireless environment. It is based on a system of "points" given to each AP in the database. When an AP is switched off, the number of points associated with that AP will gradually reduce as the users give feedback, until it is eventually deleted from the database. If a new AP is installed, the system will detect it and update the database with new fingerprints. Our proposed system has two main advantages. First it can be used as a tool to monitor the wireless environment in a given place, detecting faulty APs or unauthorised installation of new ones. Second, it regulates the size of the database, unlike other systems where feedback is only used to insert new fingerprints in the database.**

***Wi-Fi positioning; fingerprinting; user feedback; database updating; crowdsourcing.***

## I. Introduction

With the recent exponential growth of the smartphone market, the most advanced technologies are now available to the general public. Indeed, most pocket-sized phones are now equipped with Wi-Fi and Bluetooth capabilities, and an increasing number of them are equipped with assisted and high sensitivity GPS chips. This technological breakthrough, along with the development of mapping software and location-enabled databases, has enabled programmers to incorporate information about a user's location in their software, opening up a whole new field of applications. For example, the "Locale" application for Android adjusts mobile phone settings depending on its location, switching on Wi-Fi when at home for instance [1]. However, most of these applications still rely mainly on GPS to infer the user's position, and are thus restricted to outdoor operation where the device has a clear view of the sky.

The next step in delivering ubiquitous and accurate positioning is to provide coverage for indoor locations, where users spend much of their time. This field has attracted a lot of research in the past few years, and different technologies have emerged as potential candidates to provide accurate indoor positioning. One is to use Wi-Fi signals, which can have many advantages. First, no new infrastructure hardware is needed as more and more transmitters are installed each day. Also, nearly all new mobile phones are now equipped with a Wi-Fi chip. Second, it can deliver room-level accuracy, which is good enough for many applications, including asset-tracking, location-based advertising, location-based information for users, etc. Finally, it has a very low Time To First Fix (TTFF) when compared to GPS, and also drains much less battery power, which is a critical issue for mobile devices.

Two techniques have been investigated using Wi-Fi signals for positioning, both based on the measurement of signal strengths by the device to be located. The first, trilateration, attempts to convert these measurements into distances between the device and the Wi-Fi access point (AP). The second technique, referred to as fingerprinting, first requires the construction of a database (DB) of signal strengths from different APs at different reference points in the desired coverage area. The location of the device is then obtained by measuring the signal strengths at its location (its "fingerprint"), and comparing it with the different reference fingerprints in the DB.

The trilateration technique suffers from the difficulty in establishing an accurate model to convert signal strength measurements into distances, due to the complexity of indoor environments and the underlying nature of Wi-Fi signals. It also requires the exact location of each AP, which is not an easily available piece of information. The other existing technique, fingerprinting, is considered a better candidate for ubiquitous indoor positioning, and has already been used in commercially available systems such as Ekahau [2], and Skyhook Wireless [3]. However, the main disadvantage of this technique is the labour (and therefore cost) required to build and maintain the DB. Not only must an extensive survey be performed before the initial deployment of the positioning system, but every time the environment changes significantly,

978-1-4244-7879-8/10/$26.00 ©2010 IEEE

or when new APs are introduced, the initial survey is compromised and may have to be carried out again. Neither Ekahau or Skyhook have addressed these issues as the first one requires administrators to perform an extensive and very accurate survey of the area to be covered, and the second one funds drivers to drive around major cities, recording signal strength data as they do so.

Even though most research in Wi-Fi positioning is currently focused on accuracy, [4] and [5] have used fingerprinting and investigated ways to reduce the burden of the initial survey. [4] has described a sector-based indoor positioning system, where a medium-sized, five storey building was surveyed in 40 minutes. [5] has introduced a system relying on a list of APs present in the building, hence requiring no survey at all. Another important cost is the one associated with maintaining the DB over time. If not, the performance of the system degrades rapidly as new APs are installed and old ones removed. A way to reduce the cost of implementation of indoor positioning systems is to make the users themselves contribute to the survey effort. By periodically asking the user for its true position, the system can use this information to continuously update its fingerprints DB, and therefore maintain a reasonable quality of service over time.

Such a system is presented in this paper. It relies on an initial DB constructed by a skilled surveyor, which is then updated using user feedbacks. The paper is organised as follows. The next section reviews related work in the field of user feedback in Wi-Fi fingerprinting. Section 3 introduces our system, and shows how it can be used not only to maintain the fingerprints DB, but also to monitor changes in the wireless environment which is, to the best of authors' knowledge, the first attempt of its kind. Section 4 presents the results of tests conducted at the University of New South Wales, Sydney. We first measure the impact of wireless environment changes on the system's accuracy, and then demonstrate how our system reacts to such changes. We also show how erroneous user entries affect final accuracy. Finally, section 5 concludes the paper and discusses future work.

## II. Related Work

In the recent years, the concept of "crowdsourcing" as applied to Wi-Fi positioning has gained more interest. We will discuss the work of five investigators, more details can be found in [6] – [10].

Park et al. describe in [6] a system similar to ours, i.e. a sector-based indoor positioning system based on fingerprinting. They investigate solutions to a number of challenges, such as when to prompt the users for feedback, and how to filter erroneous entries. However, they do not use the feedback as a means of monitoring the wireless environment, and their system differs from ours as it doesn't rely on an initial DB built by a skilled surveyor.

In [7], Lee et al. study the impact of different parameters of the DB, such as the density of fingerprints per room, or the number of APs a device can see in a given room, on the final accuracy of the system. They also point out another challenge for crowdsourced radio maps. As the users own different devices, which can have very different sensitivities, their contribution to the DB must be adjusted according to the device they own. More work is needed in this area to check how the variety of devices may impact on the final accuracy of the system.

In [8], the results of a year-long study of a user-trained Wi-Fi positioning system are presented. Their main findings are that, first most of the contributions are made by a small number of active users, which mainly correct the system in their own areas of interest (their office for instance). Second, they point out the sources of errors in their system: differences in device sensitivity, erroneous entries by the users, access point locations moved (old data does not expire), and changes in MAC addresses in the wireless environment. They do not attempt to address any of these issues.

Bhasker et al. present in [9] a system "ActiveCampus", which relies initially on a signal propagation model linking the SS observed by the user to its distance to the APs. Accuracy is then improved by contributions from the users. Their system suffers from poor accuracies in the initial phase, and from the fact that it needs to know the APs' physical locations – information that may be hard to obtain and which may vary a lot over time.

Finally, [10] describes a system "Redpin", which uses Wi-Fi, GSM and Bluetooth as sensing devices. As in our system, it generates fingerprints from users' feedback, but doesn't require an initial survey phase, hence delivering poor accuracies in the early stages of its roll out.

## III. Underlying System Architecture

The system used here is a server / client based system. It is extensively described in [4] and [12]. We will just review the main features of our system.

### *A. Client*

The development platform for the client is the Android©-powered HTC Dream© smartphone. This platform was chosen because the Dream© is equipped with Wi-Fi and GPS, and because the Android© platform is open-source, and therefore provides an extensive API for developers, allowing full control over the hardware of the phone. The user can access three main functions when using the software:

- Search function: the user can search for a specific room or level throughout the university campus. The software displays the indoor map, and a marker at the searched room.
- Indoor map function: displays the location of the user on an indoor map.
- Outdoor map function: displays the location of the user on the main campus map.

### *B. Server*

The server was designed to support multiple clients' request simultaneously. It listens on a specific port to receive requests, and creates a separate thread to handle each request. The server can react to five different requests:

- Find: it processes the signal scan results sent by the user and computes the user's location.
- Find outdoor: same as the Find function, but the location is returned in the format latitude / longitude.
- Save: this function is used by the surveyors to build the initial fingerprints DB.
- Search: this function returns parameters for the client to be able to display the location searched by the user.
- Update: this function manages the user-feedback. It will be described later in this paper.

### C. Algorithm used for positioning

The positioning algorithm implemented in the server is based on a classic fingerprinting technique but with a few differences.

The DB contains 'fingerprints', which are of the following form:

$$f_{l_i} = \{l_i, (MAC_0, SS_0, MAC_1, SS_1, ...)\},$$

where $l_i$ represents the location at which the fingerprint is taken, and $(MAC_0, SS_0, MAC_1, SS_1, ...)$ represents the results of scanning the Wi-Fi network, $MAC_i$ being the mac address of AP $i$ and $SS_i$ being the associated signal strength. (Further details about how to build the DB will be presented in the next part of this section.) When the user requests its location, the phone sends a vector of signal strengths to the server of the form $(MAC_0, SS_0, MAC_1, SS_1, ...)$. The server then tries to find in the DB the fingerprint which is the *closest* to the vector sent by the user. The criterion used here is the signal distance between the vector sent by the user, and each of the vectors recorded in the DB:

$$d_i = \sqrt{\sum_{j=1}^{n} \left| ss_j - SS_j \right|^2},$$

where $i$ is the index of the fingerprint in the DB, and $n$ is the number of APs in the vector sent by the user. The fingerprint with the shortest distance to the data sent by the user has its location extracted and sent back to the phone.

However, the size of the DB quickly increases with the area surveyed. Testing all the fingerprints every time a location is requested would be too inefficient. The solution is to constrain the DB search to a small number of fingerprints before computing the signal distances. Moreover, an efficient algorithm to do so also reduces the probability of picking the wrong fingerprint, and hence increases the accuracy of the system, as shown in [11] and [4]. To do that, the simplest method is to select the strongest signal AP in the scan results sent by the user, and find all the fingerprints in the DB which contain this AP. A more efficient method is to select several APs in the scan result instead of one, and find all the fingerprints in the DB which contain all these APs. This method can also be improved by inspecting the signal strengths of the APs in the scan result. For instance, if APs $a$ and $b$ are in the scan result, with respective signal strengths $ss_a$ and $ss_b$, the DB can be reduced by searching for all the fingerprints that contain APs $a$ and $b$, and where the signal strengths are "close" to $ss_a$ and $ss_b$. As detailed in [11], these algorithms must not be too restrictive in order not to eliminate the correct fingerprint. [4] showed that the most efficient parameters for the algorithm are:

- Matching half of the APs sent by the user.
- Retaining only fingerprints which signal strengths values are within a 15dBm range (+/- 15dBm).

It is important to note that this algorithm does not provide any mechanism to deal with the appearance or disappearance of new APs.

### D. Database design

Our DB design is slightly different to the classic fingerprinting DB, where a fingerprint is associated with a unique set of coordinates for so-called 'reference points'. This approach was rejected as it makes the survey phase longer and more complex. Indeed, every time a reference point is surveyed in the field, the surveyor's position has to be translated into pixels on the map before inserting the fingerprint into the DB. Here, the surveyor only has to enter the room in which he/she is, and then walk around the room holding the phone to record the fingerprints, making it much quicker to construct the DB. The disadvantage of this method is that the system will not locate the phone inside a room, it will only return the room in which the user is. However, room-level accuracy is adequate for most applications. Figure 1 shows how the western part of the School of Surveying and Spatial Information Systems was divided.

## IV. UPDATING THE DATABASE

### A. Algorithm used to exploit users' feedback

The DB is updated every time a user sends a feedback to the server. In the current state of our system, the user is periodically asked if the returned location is correct and, if not, to indicate the correct one. The client then binds the location entered by the user and the scan results used to compute the erroneous location, and sends both to the server. In order to update the DB, we introduced two new columns in the existing fingerprints DB:

- Score: each AP in the DB is given a number of "points". That score is subject to the constraint:

  $$0 \le score \le MAX\_SCORE$$

  MAX_SCORE is a constant whose value depends on the number of users that contribute to the feedback, and on how many feedbacks they provide. How this value is set needs to be investigated and is beyond the scope of this paper. For our tests, we chose an arbitrary value of 30. In a given room, if an AP is present in several fingerprints, its score is the same in each fingerprint. However, the score of an AP can vary from room to room. To summarise, the score is an indicator of the confidence one can have that a given AP is present in a given room.

- Pending: this is a Boolean value, i.e. it can only take the values TRUE or FALSE. In the current system it is used to differentiate between the APs discovered by the users, in which we have less trust, and the APs discovered by the surveyor when carrying out the initial signal survey. We believe it can also act as filter for erroneous entries by users, but more work is needed to prove that.


Fig. 1. Western part of the School of Surveying and Spatial Information Systems, UNSW, as represented in the fingerprints database.

The values of these two fields are updated every time a new feedback is sent by a user. We chose to only consider "strong" APs both in the feedback and in the DB when updating. "Strong" APs are APs whose signal strengths are above a threshold value. The main reason is that we use these two fields to decide whether an AP is present or not in a given room. Strong APs are more likely to be present in all the fingerprints in a given room. Weak APs' presence or not is more random. For our experiments, we chose a threshold of -80dBm, but more research is needed to determine this value. The updating takes place in two phases, illustrated in figures 2 and 3:

- Phase one: the score and pending fields of the existing fingerprints are updated, considering the incoming feedback. For each strong AP present in the room, we check if it is present or not in the feedback. If it's not, the AP is suspected to have been switched off; and its score in the DB is reduced by one. If the algorithm found the AP in the feedback, there are two possibilities. If the APs' pending status is false, it means that the AP has already been confirmed to exist and be strong in that room. Hence its score is set back to MAX_SCORE. If the APs' pending status is true, it's an unconfirmed AP in that room, and its score is just increased by one. When the score of all the strong APs in the room have been updated, phase two is executed.

- Phase two: the score and pending status of the feedback APs are computed. For each strong AP in the feedback, the algorithm goes through the APs stored in the DB for that room. If a match is found, the pending and score fields are read and assigned to the feedback AP. If not, a new AP is suspected to have appeared in the room. Its score is set to 1 and its pending status set to true. Once all the score and pending status have been computed, the feedback fingerprint is written in the DB.

### B. *Detecting new or switched off APs*

The score and pending fields in the DB can be used as a means of monitoring the wireless environments. Our algorithm can detect whether an AP existing in the DB has been switched off or removed, and whether a new AP has been installed:

- When an APs' score reaches 0, and its pending status is false, this means that a large number of users have *continuously* reported that this AP is missing in a given room. Indeed, even if occasional users send feedback without that AP, which can occur in specific places within a given room, as soon as a user sees this AP again, its score is set back to MAX_SCORE. Hence for an AP to be deleted, it needs to be absent from MAX_SCORE consecutive feedbacks. When such an event occurs, this AP is deleted from all the fingerprints containing it in the considered room. A message is also sent to the administrator to notify him that an AP has been switched off in this area.

- If a new AP has been installed and switched on, most of the feedback users will send in the concerned rooms will contain that new AP. Its score will gradually increase until it reaches MAX_SCORE. When it does, its pending status is set to false, meaning that this AP is confirmed to be present, as a large number of users have reported it. A notification message is also sent to the administrator.

### C. *Challenges*

The main challenge to overcome when using user-feedback to maintain a fingerprints DB is to find a way to detect erroneous entries by users. As we will see later in this paper, such errors significantly compromise accuracy. We believe a strategy for filtering a limited number of erroneous entries is to not take into account the fingerprints where an APs' pending status is set to true for computing the users' location. Indeed, if a user inputs the wrong room when giving feedback, the algorithm should detect some new APs that are not recorded in the DB for that room. It will set the scores and pending status of these respectively to 1 and true. As more feedback comes in, these APs will not appear anymore in the feedback and the score will rapidly decrease to 0 again.

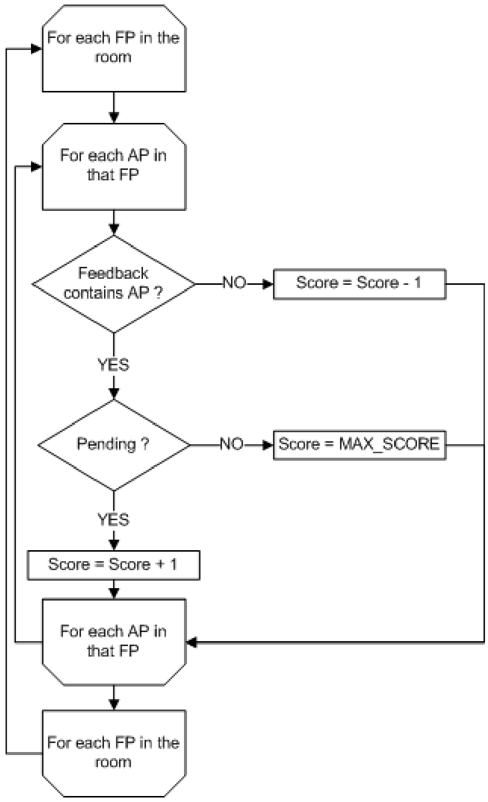


Fig. 2. Algorithm used to update the score and pending status of the existing fingerprints in a room (phase one).

Another issue that appeared when designing the system is that even though a strong AP will be in most fingerprints in a room, there is also a chance that it doesn't appear in some particular spot in that room. Hence, if a lot of feedback is given in that particular spot, the system is tricked into believing that that AP has been switched off. A solution to this problem is to limit the number of feedbacks a single user can give over a period of time in a single room. Moreover, a large number of feedbacks in one room over a short period of time is not useful as changes in the wireless environment usually take place at a slower pace (over several days or weeks).

## V. TEST RESULTS

### A. Introduction

All the tests took place on the 4th floor of the Electrical Engineering building, located on the main campus of the University of New South Wales (UNSW), in Sydney. We used six APs managed by the university that are always switched on, and two extra APs of our own that we could switch on and off as the testing took place. Four test rooms were chosen, located in the middle of the level. When our two extra APs were switched on, all eight APs could be seen by the smartphone in all four of these rooms. Figure 4 shows the positions of the APs, and of the test rooms.

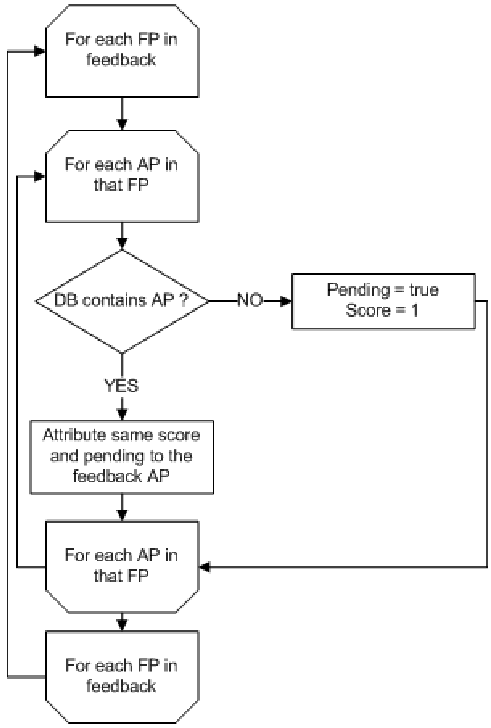


Fig. 3. Algorithm used to compute the score and pending status of the APs in the feedback (phase two).

We divided our test into four experiments. First, we show that an increased number of fingerprints provided by feedbacks, in a stable wireless environment, can significantly improve accuracy. Second, we demonstrate the effect on accuracy of a changing wireless environment. Then, we show how the addition or deletion of an AP is detected by the system. Finally, we study the impact of erroneous entries on the final positioning accuracy.

### B. Effect on accuracy of the number of feedbacks in a stable wireless environment

In this experiment the two extra APs were switched off. First, a surveyor constructed a DB containing ten fingerprints in each room. Then, 200 samples were gathered in the four test rooms. To simulate feedbacks, a surveyor walked once again around each room, gathering this time 50 scan results. This assumes that the users will give balanced feedbacks, i.e. cover the entire level. In reality this will of course not be the case. How to encourage users to give feedback in certain places, and what are the negative effects of unbalanced feedbacks, is beyond the scope of this paper. The feedbacks were then sent gradually sent to the server, and stopped at certain intervals to perform the 200 location tests in the four rooms. Figure 5 shows the results of this experiment.

Fig. 4. Location of the APs used for the tests (the yellow ones were always switched on, the red ones could be switched on and off), and of the test rooms (in the yellow rectangle).

As can be seen the system clearly benefits from receiving more feedbacks, even in a stable wireless environment. The improvement is more marked when the number of fingerprints is low in the initial DB. As more and more feedback arrives, the performance seems to reach an upper bound.

### C. *Effect on accuracy of the system in a changing wireless environment*

This experiment is intended to demonstrate the difference between a positioning system without user feedback and a system that incorporates it, in a changing wireless environment. To do that, we first show the changes in terms of accuracy when switching on two new APs that are not recorded in the DB. For this experiment a surveyor first constructed a DB of 60 fingerprints per room, with the two extra APs switched off. Then, the two APs were switched on, and 300 samples were taken in each of the four test rooms. All these samples will of course contain the two new APs, which are not recorded in the DB. Figure 6 shows how the system reacted.

As can be seen if no measures are taken on the server side to deal with new APs, the effect on accuracy is dramatic. Because the new APs are not recorded in the DB, the server fails to deliver a position 20% of the time. When a position is computed, it has 26% less chance of being within one-room than previously. A simple solution to allow the system to be able to compute a position, even when new APs are introduced, is to check if they exist in the DB, and if not, to ignore them. However, they are two issues with such an approach. The first, and most important one, is that over time more and more new APs will be introduced and old ones discarded, reaching a point when the whole signal survey has to be done again. The second one is that by using this technique, the algorithm does not take advantage of the new APs installed to increase the accuracy of the system. Indeed, the more APs installed in the area, the better the accuracy. Figure 7 demonstrates that. Here, the two extra APs were switched on, and a surveyor constructed a database containing 60 fingerprints per room. Then, 300 samples were taken in each room containing the extra APs. Two subsets of these samples were then sent to the server, the first one was the one with the raw samples, and in the second one we removed the new APs. When utilising the two extra APs, the accuracy increases quite significantly, delivering 18% more correct positions than when filtering them out.

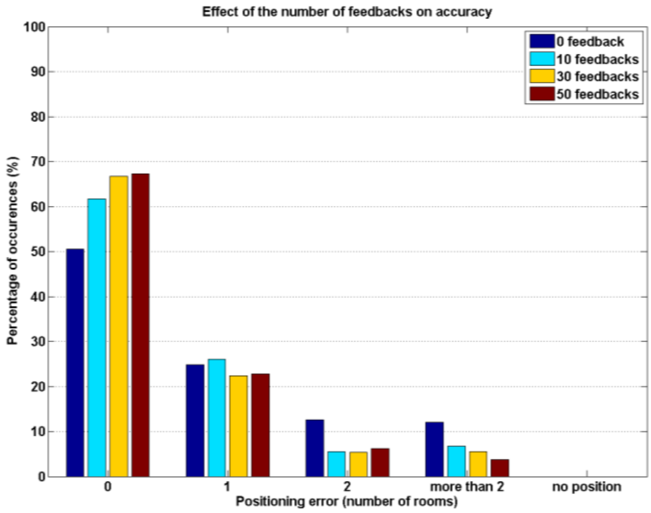


Fig. 5. Effect of the number of feedbacks on the accuracy in a stable wireless environment.

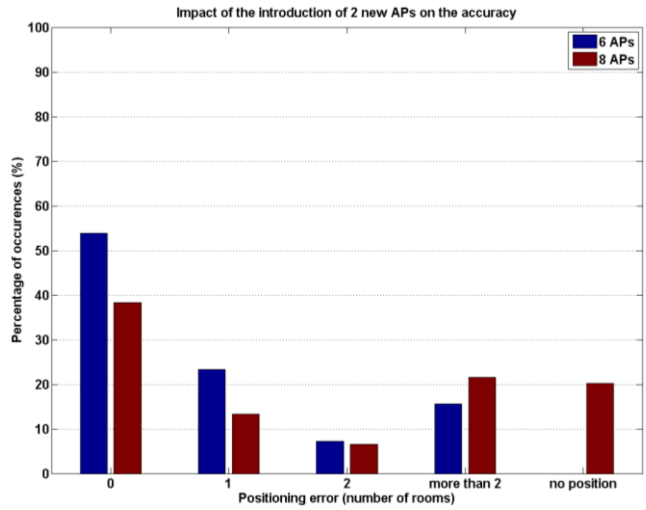


Fig. 6. Effect of the introduction of two new APs on the accuracy.

The next experiment demonstrated the efficiency of the system on accuracy. As a starting point, we switched off the 2 extra APs, and built a DB containing 60 fingerprints per room. We then switched on the two APs, and collected 50 samples in each test room. Finally, 40 scan results were collected in every room of the level, to act as user feedback. The following process was then executed: 50 samples were sent, then a feedback was applied, then the same 50 samples were sent before another feedback was sent. We repeated that sequence until we ran out of feedbacks. Figure 8 shows the results.

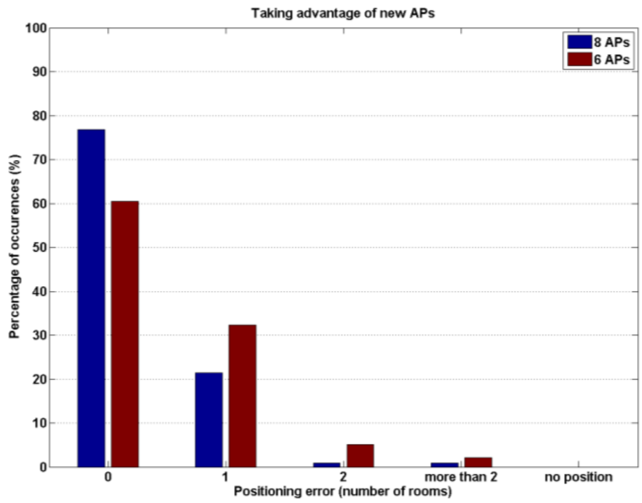


Fig. 7. Taking advantage of new APs.

As can be seen the system performed as expected. At first, the DB does not know about the two new APs, hence the accuracy is poor and the server fails to compute a position in a relatively high number of cases. As the first feedbacks containing the new APs come in, they are recorded in the DB, hence the rapid improvement in accuracy observed in the left part of the graph. Then, the accuracy improves more slowly as more and more fingerprints are written in the DB.

In the next part of this paper, we will show how the system detects new APs, or switched off ones.

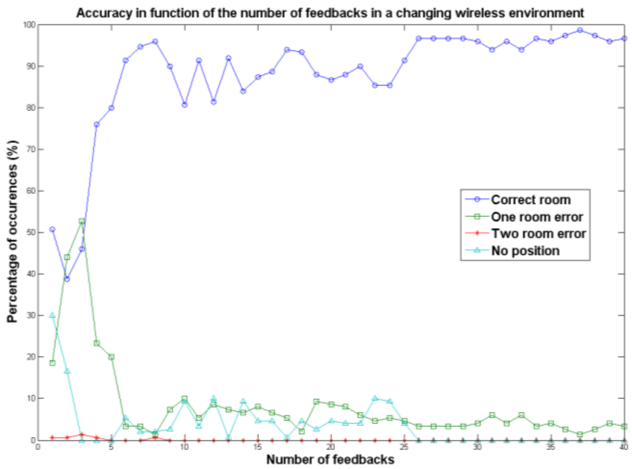


Fig. 8. Accuracy as a function of the number of feedbacks in a changing wireless environment.

### D. Detection of changes in the wireless environment

In this experiment we investigate how the evolution of the score field in the DB reflects changes in the wireless environment. A first test involved switching off the two extra APs and creating a DB of 60 fingerprints per room. We then switched on the APs and send 60 consecutive feedbacks in all the rooms of the level. Figure 9 shows the evolution of the scores of the two extra APs against the number of feedbacks, in a particular room (4Q5_1). As can be seen the feedbacks correctly detect the new APs, and the score steadily grows until it reaches MAX_SCORE. At that point the pending status of the APs is set to false, and a message is sent to the administrator to notify him of the presence of two new APs in the area.

Figure 9 also illustrates the opposite scenario. The DB was built with the two extra APs switched on, and the feedback was sent with them off. The scores in that case decrease steadily until they reach 0. At that point, the APs are deleted from all the fingerprints in that room, and a notification is sent to the administrator.

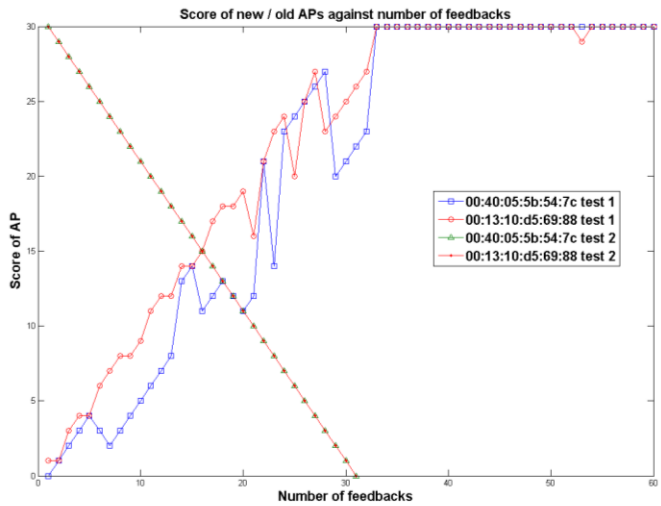


Fig. 9. Score of APs against number of feedbacks. In test 1, the two APs were switched on after the DB was built. In test 2, they were turned off after the DB was built.

### E. Effect of erroneous entries on accuracy

In this experiment we studied the impact of erroneous entries on the accuracy of the system. There are two types of erroneous entries: involuntary mistakes or intentional errors aimed at reducing the system's efficiency. When asked for their location before sending feedback, users can easily make an involuntary mistake by selecting the wrong room. Most of the time it will be an adjacent room. In the second case, users can send erroneous feedbacks, either because they deliberately want to compromise the system, or just because they're frustrated with being asked for feedback and just randomly select a room. In our experiment we just simulated the first case.

We first constructed a DB of 60 fingerprints per room, with the two extra APs switched off. We then switched on these two APs, and gathered 60 extra feedbacks in every room on the level. Four subsets of data were then built, each with different levels of erroneous entries (0%, 5%, 10%, and 20%). To "generate the mistakes", we made the assumption that when a user makes a mistake, there is a 75% of chance of selecting an adjacent room, and 25% of selecting a room which is two rooms away. After the 60 updates were applied, 300 samples were tested in each room. Figure 10 details the results.

As expected, accuracy decreases as more erroneous feedbacks are sent. In our experiment the decrease is relatively limited, due to the nature of the errors introduced by us. In a real world situation we think the decrease would be much more important as the intentionally inserted errors would have much worst consequences than the involuntary mistakes.

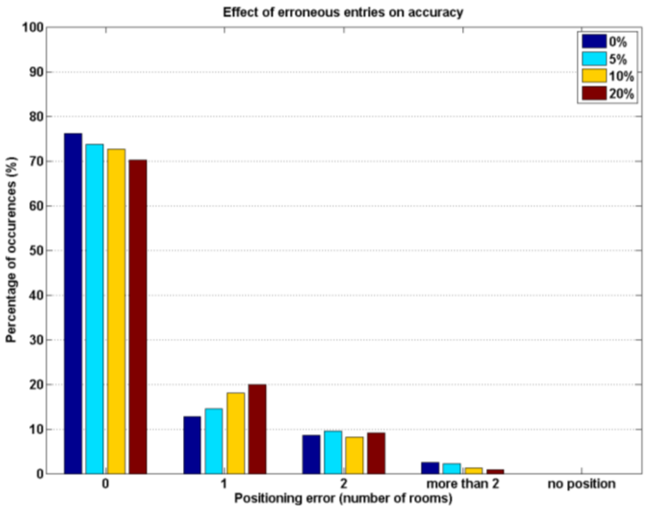


Fig. 10. Effect of erroneous entries on the accuracy of the system.

## VI. CONCLUSION

In this paper a Wi-Fi positioning system based on fingerprinting was described. It relies on an initial database built by a skilled surveyor to deliver high accuracy as soon as the system is running. To maintain the database, feedback is asked periodically of the users. The novelty of our algorithm is that the feedback is also used to monitor changes in the wireless environment, allowing the administrator to detect eventual problems with the grid of access points used for positioning. However, many issues still need to be resolved. The main one is to be able to detect and discard intentional or involuntary erroneous feedbacks, as they will quickly decrease the accuracy of the system. Another challenge is to know when and where to ask feedback of users. Depending on the users' behaviour towards the system, he or she can be asked for more or less feedbacks. Moreover, it is important for the system to have feedback in as many of the rooms as possible. We need to find a way to ask users for feedback where none is available. Future work will tackle these challenges, with the aim to perform a long duration and large scale study across the whole university campus.

## REFERENCES


[1] Locale for Android. http://www.twofortyfouram.com

[2] Ekahau. http://www.ekahau.com

[3] Skyhook Wireless. http://www.skyhookwireless.com

[4] T. Gallagher, B. Li, A.G. Dempster, and C. Rizos, "A sector-based campus-wide indoor positioning system," submitted to the 2010 International Conference on Indoor Positioning and Indoor Navigation, September 15-17, 2010, Zurich, Switzerland.

[5] M. Hermersdorf, "Indoor positioning with a WLAN access point list on a mobile device," *Proceedings of Workshop on World-Sensor-Web at th 4th ACM conference on Embedded Networked Sensor Systems (WSW'06 at Sensys'06),* October 31st 2006, Boulder, Colorado, USA.

[6] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie, "Growing an organic indoor location system," *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 271-284, June 15-18, 2010, San Fransisco, California, USA.

[7] M. Lee, C. Yu, H. Yang, and D. Han, "Crowdsourced radiomap for room-level place recognition in urban environments," *IEEE PerCom Workshop on Smart Environments (SmartE),* 2010.

[8] A. Barry, B. Fischer, and M. Chang, "A long-duration study of user-trained 802.11 localization," *Proc. Mobile Entity Localization and Tracking in GPS-less Environments,* pp. 197-212, September 2009, Orlando, Florida, USA.

[9] E.S. Bhasker, S.W. Brown, and W.G. Griswold, "Employing user feedback for fast, accurate, low-maintenance geolocationing," *Proc.PerCom PerCom*, pp 111-120, March 2004, Orlando, Florida, USA.

[10] P. Bolliger, "RedPin: Adaptive, zero-configuration indoor localization," *Proc. Location and Context Awareness*, pp 55-60, September 2008, Oberpfaffenhofen, Germany.

[11] B. Li, I. Quader, and A.G. Dempster, "On outdoor positioning with Wi-Fi", *Journal of GPS*, 7(1), pp. 18-26, 2008.

[12] W. Ching, R.J. Teh, B. Li, and C. Rizos, "Uniwide Wi-Fi based positioning system", *IEEE International Symposium on Technology and Society*, paper 26, pp. 180-189, 2010, Wollongong, Australia.