Utilization of User Feedback in Indoor Positioning System

A.K.M. Mahtab Hossain⋆, Hien Nguyen Van† and Wee-Seng Soh⋆

⋆*Department of Electrical & Computer Engineering, National University of Singapore, Email: {g0500774, elesohws}@nus.edu.sg.*

†*Department of Electrical & Computer Engineering, University of Maryland, Email: hien@umd.edu.*

Abstract

We propose an interpolation-based fingerprinting technique utilizing user feedback which does not require an exhaustive training phase typically seen in the indoor localization solutions. We argue that the contribution of users' feedback to any positioning system is two-fold. Firstly, users' feedback greatly help in fine-tuning an under-trained positioning system with proper filtering. Secondly, if users are *well-behaved*, our experimental results show that the participation of end-users can actually assist in the construction of a positioning system incrementally from scratch. We also show that user feedback-based positioning system adapts quite well when surroundings change. Our present system is built upon Bluetooth.

Key words: Positioning Systems, Indoor Localization, User Feedback, Bluetooth, Wireless Networks.

1. Introduction

Location awareness is expected to be an integral part of future ubiquitous computing environment [1]. In order to enjoy the benefits of pervasive computing, the knowledge of location with some degree of accuracy is obligatory in both outdoor (e.g., GPS) and indoor scenario. Recently, there has been a growing interest in indoor localization techniques that rely on in-building communications infrastructure (e.g., Wi-Fi, Bluetooth, etc.) mainly because it allows the design of an easily deployable low-cost positioning system. Most of these approaches utilize location fingerprinting techniques [2, 3, 4], where some location-dependent signal parameters are collected at a number of locations as location fingerprints in an "offline training phase". During the "online location determination phase", the signal parameter obtained is *compared* with those training data to estimate the user location.

The procedure of creating the training database of signal parameters entails a laborious offline phase because the location system administrator needs to take readings at every selected location of interest. Moreover, if for any unforeseeable reason, the setup changes (e.g., due to renovation, rearrangement of furniture, etc.), the whole training phase needs to be repeated again in the changed environment. In this paper, we propose an idea where the end-users can actually contribute to the construction of a positioning system incrementally, as well as the fine-tuning of an under-trained system. As a result, the aforementioned drawbacks of the fingerprinting techniques may be relieved. We define *user feedback* as the information about a user's actual position as indicated by the user to the system, either explicitly or implicitly.

We claim that the contribution of users' feedback to any positioning system is two-fold. Firstly, user feedbacks greatly help in fine-tuning an under-trained positioning system with proper filtering of the malicious feedbacks. Secondly, if users are *well-behaved*, our experimental results show that the participation of end-users can actually assist in the construction of a positioning system incrementally from scratch. UCSD's ActiveCampus project also tries to solve the indoor localization problem with the help of user feedbacks [5]. They utilize the corrections made by users on their estimated positions similar to us. However, their interpretation of the user's correction is simply a location and its received signal strength (RSS) signature pair, similar to a traditional training sample, which is completely different from how we interpret user feedback.

We contend that the combination of user feedback together with interpolation methods could eliminate the need for an exhaustive training phase, as the need for signal strength survey by administrators has been the key obstacle for the mass deployment of fingerprint based indoor positioning system. Our system can be particularly beneficial for large area deployment where it is quite demanding on the system administrator's part to visit all the possible areas and tirelessly perform the training phase. A user's feedback may not always truly reflect his/her actual location either due to the user's carelessness while giving feedback or deliberate ill intentions. Therefore, we define a Region of Confidence (RoC) with each estimated position to provide a measure of likelihood of a user's position, which is not just useful to the user when they give feedback; but also helps to assign *credibility* to each individual feedback in order to aid its incorporation into our system.

The important issue of adapting the positioning system seamlessly when its surroundings change (without performing the entire training phase all over again), has been overlooked in most fingerprint-based localization research. In our work, we emphasize that, a positioning system that exploits user feedbacks would guarantee reasonable performance over a longer period even if its surroundings change. This is crucial as the environments in a real system could constantly change, and it will be very difficult and demanding if system administrators need to monitor such changes and having to perform the signal strength survey all over again every time it changes. Apart from the above novelties, we have also denoted the signal strength signature of a user feedback in an efficient way and proved it analytically. In the following, we summarize our objectives:

- We try to relieve the exhaustive training phase of a traditional fingerprint-based positioning system through user participation in both explicit and implicit ways.

- We show that user feedback can greatly help in fine-tuning an under-trained positioning system which is already in operation. Moreover, under certain assumptions on user behavior and with the help of our interpolation method, we show that a positioning system solely based on user feedback could be built from scratch.

- We also show that, with the help of user feedback, changes in surroundings could be detected, allowing the system to adapt to the new environment in a seamless manner.

We hold the view that, user feedback may be obtained in both explicit and implicit ways. Explicit feedback is the one where user specifically inputs a correction to the system's estimation of his/her whereabouts. Implicit feedback is indirectly obtained from the user by the system without the user being aware of it. In the following, we list a several ways in which user feedback (explicit or implicit) may be obtained:

- In an indoor scenario, a user may sometimes know where he/she is at present, but he/she may wish to obtain the route to another place within the same building from there. By explicitly inputting a more accurate starting point than what the system suggests, he/she can obtain a more refined route from the system than the original suggested route.

- In a commercial system, a user who volunteers to provide explicit feedback in an area he/she is familiar with, may earn credits for using the positioning service in an unfamiliar area later on.

- User trails as in [6] could be utilized to provide implicit feedback. In [6], the user trail is recorded as an ordered sequence of landmarks (e.g., access points (APs), card readers, etc.) where he/she has visited. To formulate a user feedback from trails, we could collect the signal strength samples of a user device between the user's visit to two successive landmarks. Since the start and end positions of the user are known (i.e., the two landmarks' positions), the intermediate locations could be interpolated by applying some assumptions on the user movement (e.g., constant speed). Subsequently, these interpolated locations could be correlated with the signal strength samples collected and treated as user feedback.

- Various landmarks (e.g., APs, tags, card readers, etc.) installed at several fixed positions in the building could act as continuous sources of implicit feedback as well.

The rest of the paper is organized as follows. In Section 2, we discuss our user feedback model and explains its various components elaborately. Section 3 describes our user feedback based positioning system in detail and presents experimental results and findings. In Section 4, we provide a brief description of related works. Finally, we depict in Section 5 the conclusions drawn and future work.

2. User Feedback Model

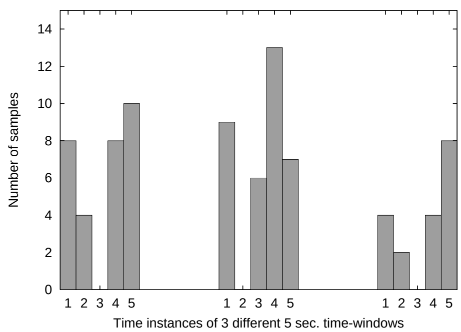
As previously mentioned, user feedback is the information about a user's actual position as indicated by the user to the system either explicitly or implicitly. In this section, we discuss how the user feedback is visualized from a positioning system's point of view. Whenever a user inputs feedback to the system, it is interpreted as, $\mathcal{F} = (\mathcal{L}, \mathcal{S}, w)$, where

$$
\begin{aligned}
\mathcal{L} &= [x\ y]^T = \text{the position indicated by a user,} \\
\mathcal{S} &= [\bar{S}_1\ \bar{S}_2\ \ldots\ \bar{S}_K]^T = \text{the RSS signature of the feedback captured at the } K \text{ APs,} \\
w &= \text{the degree at which a system believes the feedback, i.e.,} \\
&\quad\ \text{the } \textit{credibility} \text{ or } \textit{weight} \text{ of each individual feedback.}
\end{aligned}
$$

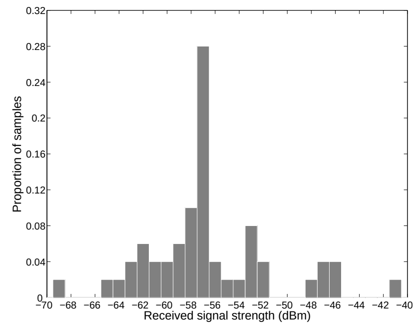
Next, we elaborately discuss all three components of a user feedback in Section 2.1, 2.2 and 2.3, respectively.

2.1. Location Indicated by User, $\mathcal{L}$

A user feedback is obtained when the user indicates his/her actual position to the system (either explicitly or implicitly). This location information is generally interpreted by the system as Cartesian coordinates (i.e., $[x\ y]^T$) in an indoor environment. In practice, there might be some uncertainties involved when a user tries to indicate his/her actual position at the time of providing



(a) Number of samples collected per second over three 5-sec windows. It can be seen that, some slots are empty.



(b) Histogram of signal strength samples received at an AP when the mobile device used in training is stationed at a particular location.

Figure 1: Characteristics of signal strength samples utilized to define the signature of user feedback

feedback to the positioning system. In case of *explicit* feedback, these uncertainties might arise owing to the carelessness on the user's part while pinpointing his/her location on the map, or he/she may deliberately provide inaccurate location information. In case of *implicit* feedback, these uncertainties arise when the system's predicted location deviates from the user's actual position. We will discuss two different user models in Section 3.3 which try to broadly emulate these two types of user behaviors while providing feedback.

2.2. Signature of a User Feedback, $\mathcal{S}$

We first discuss our choice of a user feedback's signature, and then prove that it is an efficient one. During the offline training phase of a fingerprint-based positioning system, we know that the system administrator positions himself/herself at a particular location of interest for the RSSs to be measured at the APs. The RSSs perceived at the APs actually denotes the signature of that particular location. We also utilized the RSSs measured at the APs during a user feedback to denote its signature in a similar way taking into account some additional details. For example, in order to denote the signature of a user feedback, we sample the signal strengths perceived at APs over a 5-second window, and instead of using a single sample from each AP, the *mean* of all the samples over the 5-second window has been used. Furthermore, the time when a user clicks his actual position in the map is treated as the median of that window. Our approach is taken in view with the following facts:

(i) Whenever a user clicks to input feedback, it is reasonable to assume that he/she has been at that particular location for a while. Hence, we have chosen the clicking instant of the user as the median of the 5-second window, rather than the beginning of the window.

(ii) The probability that an AP fails to collect any sample from the mobile node (MN) during a user feedback is greatly reduced as well. Fig. 1(a) shows some cases when our AP failed to receive any sample from the MN within certain slots of a user feedback's time-window. If the probability that an AP receives a sample from an MN is q, then the probability that an AP receives at least one sample within the 5-second window can be expressed as, $1 - (1 - q)^{5m}$, where m is the number of packets sent by the MN within a 1-second slot and each 1-second slot is assumed to be independent. For example, if $q = 0.5$ and $m = 2$, the probability of getting a sample at the AP during a user feedback increases from 0.75 to 0.999 when a 5-second window is considered compared to 1 second.

(iii) Capturing more samples should provide more information about the signal strength distribution at a particular location, which generally has a tail (see Fig. 1(b)). The use of just a single sample would be unlikely to work well.

(iv) The *mean* of all the collected samples' signal strengths inside the time-window is an efficient unbiased estimate of a user feedback's signature compared to any other linear combination of the samples' RSSs. This can be realized with the help of Theorem 1.

Theorem 1. *Suppose S_k denotes the signal strength distribution of the samples collected at the k^{th} AP during a user feedback. If S_{ki} specifies the i^{th} sample's RSS of the n samples observed inside the time-window at that AP, then the linear combination of observations $\sum_{i=1}^{n} a_{ki} S_{ki}$ is an unbiased estimate of $E(S_k)$ given $\sum_{i=1}^{n} a_{ki} = 1$. It is also the most efficient one when $a_{ki} = \frac{1}{n}, \forall i \in \{1, 2, \ldots, n\}$.*

Proof. This can be proved with basic estimation theory properties. The proof is shown in Appendix A.

Corollary 1. *If $\overline{S}_k$ is an efficient unbiased estimate of the signal strength samples' signature collected at the k^{th} AP inside a time-window, then for a positioning system with K APs, $\mathcal{S} = [\overline{S}_1 \ \overline{S}_2 \ \ldots \ \overline{S}_K]^T$ is indeed an efficient unbiased estimate of a user feedback's signature.*

Proof. Corollary 1 can be realized by extending Theorem 1 for all the K APs, together with the assumption that the APs are independent of each other [7].

2.3. *Credibility or Weight of a User Feedback, w*

Without the credibility factor, w, a user feedback is typically a traditional training sample of location and RSS signature pair $(\mathcal{L}, \mathcal{S})$ from a positioning system's perspective. The traditional training samples are generally collected by a positioning system's administrative people. Therefore, all the samples are treated with equal importance. On the other hand, the sources of user feedbacks can be different entities (e.g., system administrators, normal users, intruders etc.). Consequently, there should be certain credibility factor associated with each feedback given, i.e., a measure for the system to believe that the user is actually at his/her claimed position. In many ways, this approach is similar to *location verification* technique which ensures that the claimed source location is associated with a high level of trust. Existing location verification techniques [8, 9, 10, 11] either accept/reject a source's location claim. They generally require specialized hardware (incorporated with non-RF technologies) to verify a source's location claim more precisely [8, 9] or the accuracy level within which the location claim is verified, is set to be quite large [10, 11]. However, our positioning system has certain implications which makes the use of these location verification techniques infeasible:

(i) Our positioning system is built upon RF technology (Bluetooth) preferably using off-the-shelf hardware in order to provide location service in a cost-effective way. Consequently, the more precise solution to verify a location claim with the help of specialized hardware is not applicable.

(ii) The accept/reject policy of the existing location verification techniques would restrict the user feedback to have only one of the two extreme values, i.e., $w \in \{0, 1\}$. If a strict margin is set for incorporating the user feedback, then many useful feedback might be filtered out. On the contrary, if a large tolerance level is set, many malicious user feedback might be incorporated which may ultimately cause the actual accuracy offered by the system to deteriorate.

Therefore, instead of an accept/reject policy of the existing location verification techniques, we come up with a strategy which assigns *relative* weights to the user feedbacks utilizing their credibility. Later on, it will be shown that, this approach actually helps in fine-tuning an existing positioning system to achieve better accuracy. Next, we elaborately describe how the user feedbacks are assigned relative weights based on their credibility while being incorporated into the system. In order to realize this, we first describe the "Region of Confidence (RoC)" concept, which subsequently helps to derive our weight assignment policy for each individual feedback.

2.3.1. Region of Confidence

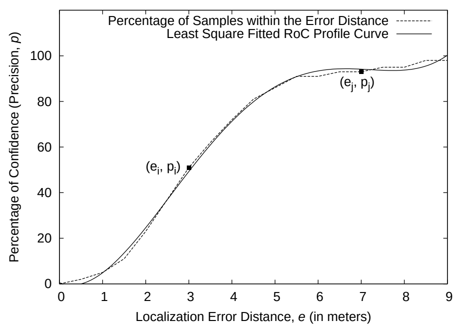
We define a system parameter, RoC, which gives a measure of the system's overall accuracy and precision[1]. We express RoC as a two-parameter entity, i.e., (e, p), where the parameters e, and p, denote the accuracy grain size and the expected precision of the system, respectively. In localization literature, the term "accuracy" generally indicates the grain size of the position information provided (in some works, the accuracy grain size is referred as "localization error distance" as well), while the term "precision" specifies how often we could attain that accuracy [13]. For example, if a positioning system can determine positions within 3 meters for about 90 percent of the measurements, that particular system qualifies to be 90% precise in providing 3-meter accuracy. Intuitively, a higher precision would compel the system to provide a coarser accuracy, and similarly, in order to achieve finer accuracy, the system may turn out to be not so precise. We define RoC in a way that considers both requirements, in order to facilitate our feedback-based positioning system. In general, RoC provides a measure of likelihood of a user's estimated position and also influences the weights that would be associated with the feedbacks which we describe later.

In order to create the "Precision vs. Accuracy" graph of Fig. 2(a), which we term as "RoC profile graph", first we assume that our positioning system is already in an initial state with some training samples. Now, we inspect its performance when *well-behaved* users' (whose claimed locations do not deviate from their actual locations by a large margin) feedbacks are incorporated into the system in order to obtain the "RoC profile graph". It can be seen that, the shape of our "RoC profile graph" has a similar trend as those "Precision vs. Accuracy" curves found in existing localization literature [14, 15]; it shows that the precision, p, increases with larger accuracy grain size or localization error distance, e. Intuitively, the "RoC profile" may not be fully reflective of the system's actual state with only a limited number of user feedbacks. As we gather more and more user feedbacks, we can approximate "RoC profile" more accurately (using the feedbacks as both training and testing samples). In the following section, we depict how the "RoC profile" has been utilized to derive the trend of the credibility to be assigned to a user feedback.

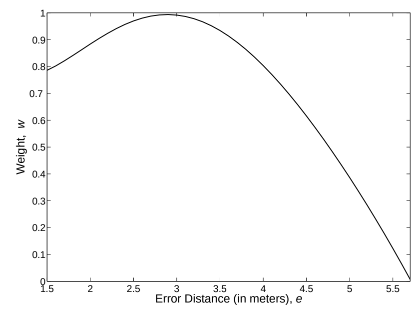
2.3.2. Feedback Weight Assignment Policy

Since user feedbacks may contain dubious information, we should not treat all feedbacks with equal importance. Whenever a user claims to be at a particular location via feedback, that

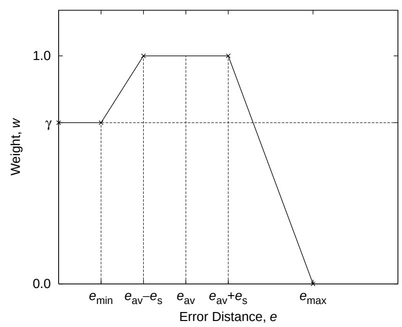
[1]Note that, our definition and purpose of RoC is quite different from an earlier work. In [12], RoC was formed utilizing simple geometry in order to fight *aliasing*, i.e., to eliminate physically different locations which have similar signatures in signal space.



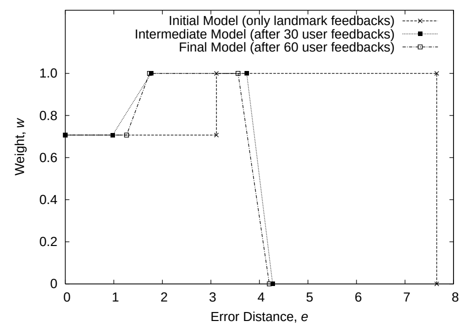
(a) RoC profile graph showing that precision increases with larger accuracy grain size or localization error distance.



(b) Feedback weights' profile generated from the RoC profile shown in Fig. 2(a) using Eqn. (2)



(c) Our simplified feedback-weight assigning model pertaining to Fig. 2(b).



(d) Various stages of our feedback-weight assigning model as the number of user feedbacks increases.

Figure 2: Illustration of how we approximate the feedback-weight assigning model from the RoC profile graph, as well as its variation when different number of feedbacks are incorporated

information is associated with a certain degree of credibility. In order to calculate this credibility factor, consider a positioning system where n user feedbacks have been utilized as test samples to obtain the "RoC profile". Subsequently, for any point (e, p) of Fig. 2(a), it is obvious from the definition of RoC that, $p \times n$ user feedbacks' estimated positions do not deviate from its actual one by more than e. In other words, if we think of a circle with the accuracy grain size or localization error distance, e, as radius, then $p \times n$ user feedbacks can be thought to be inside it. Now, suppose if we increase the radius e by a small amount Δe (i.e., p also increases in Fig. 2(a)), then Δn new user feedbacks fall inside the new area. So, the proportion of user feedbacks falling inside the area $[\pi(e + \Delta e)^2 - \pi e^2]$ is $\frac{\Delta n}{n}$. Consequently, we denote the probability of occurrence of a user feedback inside this unit area as,

$$\lambda = \frac{\frac{\Delta n}{n}}{[\pi(e + \Delta e)^2 - \pi e^2]} \approx \frac{\Delta n}{2\pi n e \Delta e}. \tag{1}$$

Subsequently, we define the *weight* or *credibility* of the i^{th} user feedback utilizing (1) as follows:

$$\omega_i = \frac{\lambda_i}{\max\{\lambda_1, \lambda_2, \ldots, \lambda_n\}}. \tag{2}$$

Note that, ω_i is just the normalized form of λ_i so that $\omega_i \in [0, 1)$. Now, let us investigate the rationale behind choosing such a weight assignment criteria. Consider two user feedbacks, i and j with RoC (e_i, p_i) and (e_j, p_j), respectively. Their positions in the "RoC Profile Graph" are shown in Fig. 2(a) where $e_i < e_j$. Following similar steps which were involved in obtaining (1), we have,

$$\lambda_i \approx \frac{\Delta n_i}{2\pi n e_i \Delta e} \quad \text{and} \quad \lambda_j \approx \frac{\Delta n_j}{2\pi n e_j \Delta e}.$$

The parameters e's and Δn's have certain effects in the above expressions:

(i) $e_i < e_j$ implies the accuracy of the i^{th} feedback's estimated position by the system is higher than that of the j^{th} user feedback. Therefore, from the system's perspective, it is natural to believe the i^{th} user feedback more than the j^{th} one.

(ii) Consider the number of user feedbacks, Δn_i and Δn_j which fall into the two new areas that have been formed by extending the radius e_i and e_j by the same amount, Δe, respectively. If $\Delta n_i > \Delta n_j$, then a greater number of user feedbacks which are used to create the "RoC Profile Graph", falls into the i^{th} feedback's new area than that of the j^{th} feedback's area. Consequently, it is natural for the system to believe the i^{th} user to be more *well-behaved* since the system's "RoC Profile Graph" had been created utilizing the *well-behaved* users' feedbacks as mentioned in the previous section. Therefore, it is only fitting to assign more weight to the i^{th} user feedback than the j^{th} one.

From Fig. 2(a), using the numerical values of the parameters, $n = 44$, $\Delta e = 0.5\text{m}$, $e_i = 3\text{m} < e_j = 7\text{m}$, and $\Delta n_i = 5 > \Delta n_j = 1$, we find, $\lambda_i > \lambda_j$. In other words, the i^{th} user feedback is more believable than the j^{th} user feedback from our positioning system's perspective. Next, we describe our ultimate simplified weight assignment policy for each individual feedback taking into account the aforementioned facts.

By utilizing the RoC profile together with (2), we obtain the trend for weights to be associated with user feedbacks as shown in Fig. 2(b). We observe that, the weight's maximum occurs when the accuracy grain size or the localization error (e) of the user feedback's estimated position is close to our system's average localization error ($\approx$ 3m), and decreases as the estimation error becomes larger. Since it is desirable to have a weighting scheme that is simple and yet capable of evolving with time as more user feedbacks become available, we define a feedback-weight assigning model as follows. A maximum weight of 1 shall be assigned when the localization error (e) of a user feedback's position is within one standard deviation (e_{s}) from the average error (e_{av}), as shown in Fig. 2(c). This is in accordance with the view that our system is fairly accurate and therefore, we expect the system's estimated positions' errors to be around this average quantity. Assigning maximum weight around one standard deviation of this average helps to build, and subsequently, fine-tune the system gracefully. From $e_{\text{av}} + e_{\text{s}}$ to e_{max} (maximum error), the weight follows a similar trend as in Fig. 2(b). The horizontal dotted line (i.e., $w = \gamma$) of Fig. 2(c) indicates the filter of our weighting scheme. We associate a constant weight, γ (which is 3 dB lower than w_{max}), to the user feedbacks when the estimation error is less than e_{min}, in the view that our system's predictions of these positions are already quite good. The weight assignment

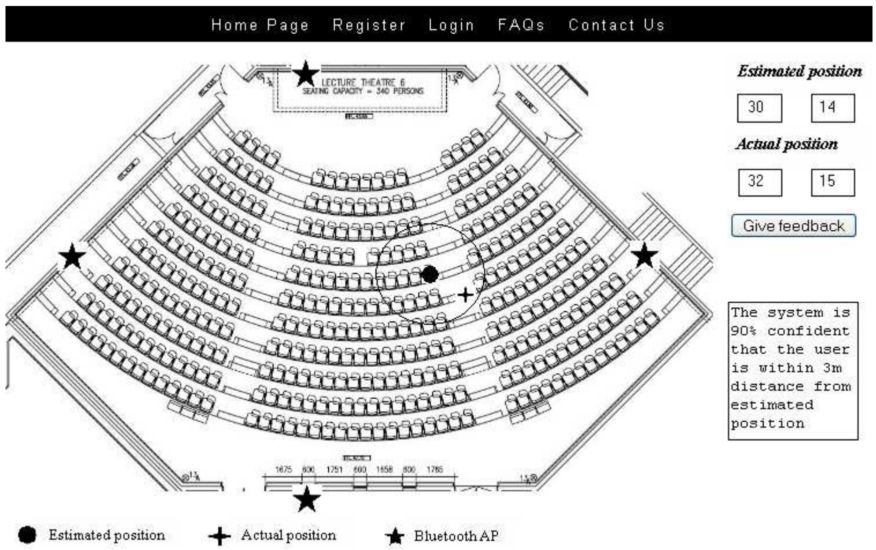


Figure 3: Interface for user feedback input – the experimental testbed is a lecture theater in campus. For each user, the system's estimate of his/her position together with the RoC is shown. The users can click on their positions within the map and press "Give feedback" button to provide feedback.

policy for the i^{th} user feedback of our model as shown in Fig. 2(c) can be summarized as,

$$w_i = \begin{cases} \gamma & e_i \leq e_{\text{min}} \\ 1 + (1-\gamma)(\frac{e_i - e_{\text{av}} + e_{\text{s}}}{e_{\text{av}} - e_{\text{s}} - e_{\text{min}}}) & e_{\text{min}} < e_i < e_{\text{av}} - e_{\text{s}} \\ 1 & e_{\text{av}} - e_{\text{s}} \leq e_i \leq e_{\text{av}} + e_{\text{s}} \\ 1 + (\frac{e_i - e_{\text{av}} - e_{\text{s}}}{e_{\text{av}} + e_{\text{s}} - e_{\text{max}}}) & e_{\text{av}} + e_{\text{s}} < e_i < e_{\text{max}} \\ 0 & e_i \geq e_{\text{max}} \end{cases}$$

where $e_i = \sqrt{(x_i - x_{\text{est}})^2 + (y_i - y_{\text{est}})^2}$, $[x_i \; y_i]^T$ is the i^{th} user's claimed location, and $[x_{\text{est}} \; y_{\text{est}}]^T$ is the system's estimate of that user's position.

Fig. 2(d) shows the evolvement of our feedback-weight assigning model as user feedbacks are increasingly incorporated. Our initial system only consists of *landmark* feedbacks (e.g., the feedbacks from the 4 APs). Two other stages of our system are shown in Fig. 2(d) where 30 and 60 *well-behaved* user feedbacks are subsequently incorporated. The definition of various user feedbacks (e.g., landmark, well-behaved etc.) can be found in Section 3.4. For each stage of the system, 44 testing samples which are completely different from the incorporated user feedbacks are utilized to obtain the error model. As can be seen, this model helps to improve the accuracy of our system, since both the average error and its standard deviation decreases with increasing number of user feedbacks.

3. User Feedback based Positioning System

In this section, we elaborately discuss our user feedback based positioning system, and present our experimental results and findings.

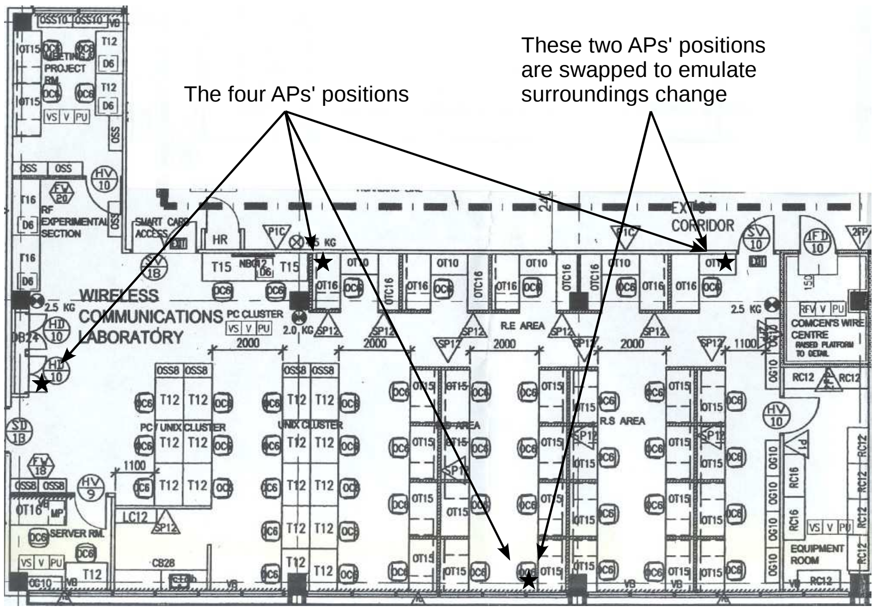


Figure 4: Our second experimental testbed which is located within a research laboratory.

3.1. User Interface and Experimental Testbed

We start by providing a brief description of our user interface used to input explicit feedbacks into our system. Fig. 3 shows the interface for a user to input feedbacks that are to be incorporated into our positioning system. We can observe from the interface that, a user is always provided with the system's estimation of his/her position (i.e., the shaded circle on the map) together with the RoC. Subsequently, the user can choose to inform the system about his/her actual location by clicking on the corresponding position within the map, and pressing the "Give feedback" button.

We have two experimental testbeds. The first is located inside an amphitheater of our school, which spans over an area of 540 m^2. The second is located within a research laboratory having an area of 214 m^2, and includes many small cubicles for the research students. We have used four Aopen MP945 Mini PCs to serve as our access points (APs), which are placed near the ceilings. The locations of these APs in the two testbeds are shown in Fig. 3 and 4 respectively, marked as stars. Each MP945 is incorporated with BT-2100 Class 1 Bluetooth adapters, which keep on scanning for Bluetooth packets by issuing inquiries periodically. All our mini PCs run SuSe 10.1 Linux distribution with the latest BlueZ protocol stack [16].

3.2. Usage of User Feedback in Positioning Algorithms

Depending on the positioning algorithm used, there are various ways how a user feedback can be utilized. In the following, we briefly describe the two approaches we have undertaken in order to make use of the user feedback into our positioning algorithm.

(i) We utilize interpolation technique to create the RSS signature of a *fictitious* training point where no training sample has been taken. Unlike a typical fingerprint-based positioning system that requires an exhaustive sample collection phase, interpolation helps to achieve the same goal with much fewer training samples. In addition, it is advantageous in our case since the user feedback locations may not be uniform over the entire localization area.

Interpolation technique can help to fill up the voids in the training database where no user feedback has been obtained. We have used weighted linear regression to generate the interpolated RSSs exploiting spatial similarity like our previous work [17]. In order to deduce a fictitious training point j, each AP's RSS is formulated according to (3) (given in Appendix B), exploiting the signal strength values collected at the APs for user feedbacks. If there are K APs, K different regression equations will be formed in order to deduce a single fictitious training point's fingerprint. However, the difference from our previous work [17] is – whereas the weight in (4) corresponds only to the spatial similarity factor; here, the user feedback's credibility factor is also taken into consideration regarding the weight calculation. The spatial similarity weights are assigned taking into account the property which basically states that the RSSs observed at neighboring locations tend to exhibit similar properties [7]. In our experiments, we have chosen this spatial similarity weight to be inversely proportional to the distance between a certain fictitious point j and the actual training point i (i.e., $\frac{1}{d_{ji}}$). In Appendix B, we provide the details about how interpolation technique predicts the RSS of a *fictitious* training point where real training samples are not collected or obtained through user feedbacks.

(ii) We have used two well-known localization algorithms (i.e., weighted K-Nearest Neighbors (KNN) and Bayesian) [2, 4] where the user feedbacks' weights are utilized to denote the weights of the algorithms.

3.3. User Models

In this section, we describe our two user models which try to emulate the two broad categories of the user behavior while giving feedbacks. These "user feedback behavior" models are utilized in the experiments to emulate the *real* user feedbacks from our collected data.

(i) *User Model 1:* In case of explicit feedback, the user may be unfamiliar with the surroundings, and subsequently fails to pinpoint his/her actual position on the map. In case of implicit feedback, the system's interpretation of the user's location may not be fully accurate. We model this phenomenon as, $[x\ y]^T = [x_a + N(0, \sigma^2) \quad y_a + N(0, \sigma^2)]^T$, where x_a and y_a denote the actual location coordinates when no uncertainty is involved and $N(0, \sigma^2)$ is a normal distribution with zero mean and variance σ^2. We assume that this is the most common model of a user's feedback and it is also capable of modeling many different user feedbacks (by varying σ). For example, we know that a *well-behaved* user is the one whose claimed location does not deviate from his/her actual location by a large margin. For experimental purposes, we model a *well-behaved* user as one where the uncertainty parameter of the feedback position (i.e., σ^2) does not exceed the system's ultimate achievable average accuracy. Since our positioning system can offer 3m average accuracy, we assume that the feedback position of a *well-behaved* user regarding our system conforms to the equality, $\sigma = \sqrt{3}$m.

(ii) *User Model 2:* In case of explicit feedback, there may be some feedbacks where the user feels totally unsure about his actual position corresponding to the map. In case of implicit feedback, the system's interpretation of the user's location may be way off. We model this phenomenon as, $[x\ y]^T = [U(0, x_{\max}) \quad U(0, y_{\max})]^T$, where $x_{\max}$ and $y_{\max}$ depict the maximum possible location coordinates of the testbed and $U(\cdot)$ denotes a uniform distribution over the range. The feedbacks given by those who try to sabotage the positioning system intentionally, also fall into this category.

3.4. Classification of User Feedback

Based on our user models of the previous section and the weight assignment policy for each individual feedback discussed in Section 2.3.2, we classify user feedbacks into four categories:

(i) *Super-user feedback:* These are the feedbacks provided by system administrators and alike, and they are expected to be included into the system with 100% belief (i.e., $w = 1$).

(ii) *Regular-user feedback:* We consider the feedbacks from ordinary users who use the positioning system's services to be the mainstay in the fine-tuning of our system. These are the most common type of feedbacks which are amalgamated with some uncertainties. Our *User Model 1* discussed in the previous section tries to emulate this particular type of feedback.

(iii) *Landmark feedback:* The APs can be regarded as sources of feedbacks as well, since they also transmit radio signals, and their locations are known and fixed. We have four such APs in each of our experimental testbeds as shown in Fig. 3 and Fig. 4, respectively. Note that, the RSS signature vector of this type of feedback comprises of $K - 1$ components instead of K, because one of the K APs is actually considered as an MN here. We fill this void with the maximum RSS rating corresponding to our Bluetooth adapter. Apart from that, various other devices (e.g., beacons, card-readers, tags etc.) installed at several fixed positions in the building could act as continuous sources of landmark feedback too. Landmark feedback is a form of super-user feedback (just that the sources are static fixed points) since it is always believed with $w = 1$. Therefore, the inclusion of such static fixed points as a source of feedback will increase the number of super-user feedbacks, and subsequently, will have positive impact on localization accuracy.

(iv) *Spurious-user feedback:* The feedbacks given by those users who are oblivious about their surroundings, and also those who aim to sabotage the positioning system, are harmful. Instead of fine-tuning the system to achieve better accuracy, these spurious-user feedbacks could make the positioning error larger if incorporated. Our weight-assignment policy of Section 2.3.2 ensures that these types of feedbacks are filtered out.

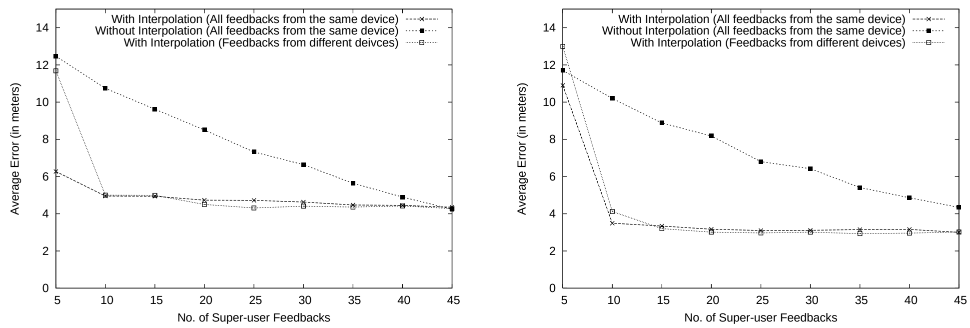
3.5. Results and Findings

The results of Section 3.5.1 are based on the experimental data of our lecture theater testbed (Fig. 3) while the results presented in Sections 3.5.3 and 3.5.4 are obtained from our research laboratory testbed's data (Fig. 4).

3.5.1. Interpolation aids our user feedback based positioning system

In order to demonstrate the usefulness of our interpolation based approach described in Section 3.2, we have carried out an experiment that only considers super-user feedbacks, where all feedbacks are assigned the maximum weight (i.e., $w = 1$). As can be seen from Fig. 5(a) and 5(b), the system that uses interpolation easily outperforms the one that does not.

Since different users are expected to carry devices with heterogeneous hardware, selecting RSS as a location fingerprint could easily hamper a user feedback based positioning system. RSS is known to vary quite significantly at a particular location for different device hardware even under the same wireless conditions [17, 18, 19, 20]. As a result, we have chosen a robust location fingerprint, namely, *Signal Strength Difference (SSD)*, since it is argued to be able to accommodate devices with heterogeneous hardware solutions unlike the RSS [17]. We also verified our system's robustness when the users input their feedbacks using different types of devices (e.g., Bluetooth Class 1 or 2 devices), which could easily occur in a real deployment scenario. Fig. 5(a) and 5(b) show similar performance for both cases, regardless of whether the user feedbacks are given using only one type of device or not.



(a) When KNN localization algorithm is used.

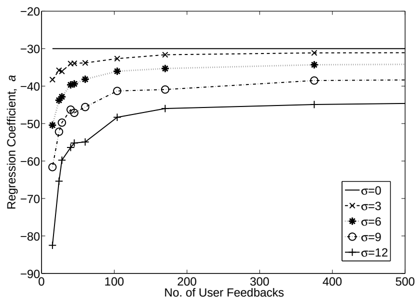
(b) When Bayesian localization algorithm is used.

Figure 5: Demonstration of how interpolation helps to improve our positioning system's accuracy – only super-user feedbacks are considered here.

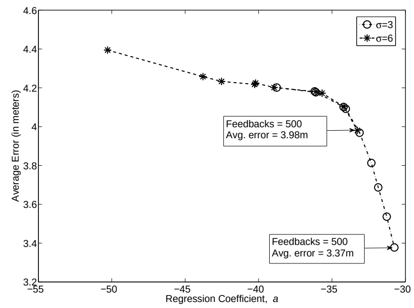
3.5.2. Evolvement of user feedback based positioning system

In this experiment, we investigate the prospect of creating a positioning system utilizing only regular-users' feedbacks from scratch. We try to estimate the linear regression coefficients for the equation in (3) (given in Appendix B) which are necessary for generating the interpolated training points from user feedbacks. Here, we emulate different types of users by changing the value of σ of "*User Model 1*" which we have defined in Section 3.3. We contend that if the two linear regression coefficients (i.e., a and b) computed from regular-user feedbacks can somehow match the coefficients computed from super-user feedbacks, then our interpolation-based approach should perform equally well even though these feedbacks have uncertainties. We see from Fig. 6(a) that using feedbacks from users exhibiting lower uncertainty (e.g., $\sigma = 3$) can almost achieve the same a as the case when no uncertainty is involved ($\sigma = 0$). Furthermore, it can be noted from Fig. 6(a) that increasing the uncertainty in user feedbacks have the effect of swaying the estimated a values away from the $\sigma = 0$ case. Similar observations have been made with the other coefficient, b.

In our interpolation-based approach, we first calculate the regression coefficients (i.e., a and b) for all the APs at an interpolated point making use of the user feedbacks as training samples. Subsequently, the RSS signatures of the APs at every interpolated point are calculated, and all of them are then treated as normal training samples together with the user feedbacks in our localization algorithm. Table 1 lists the average localization errors when a significant number (= 500) of user feedbacks with different values of uncertainty parameter, σ, are being considered. We see that the average accuracy (3.37m) achieved for $\sigma = 3$m case is very close to the accuracy when there is no uncertainty (3.1m). This is expected since the calculated a value for $\sigma = 3$m case after 500 feedbacks is very close to the a value obtained for $\sigma = 0$ (see Fig. 6(a)). The higher uncertainty cases (e.g., $\sigma = 6$m, $\sigma = 9$m, etc.) report coarser accuracy as can be seen from Table 1, which is also justified according to their curves shown in Fig. 6(a). Therefore, we can approximate the regression coefficients of our interpolated points more accurately for user feedbacks with lower uncertainty which in turn yields better localization accuracy. In a nutshell, we argue that if we decide to build our system with user feedbacks from scratch, our interpolation-based approach may still enable us to achieve reasonable accuracy, provided that the user behavior does not stray too drastically. Note that the results for this particular experiment



(a) The values of regression coefficient, a, required for predicting the RSS at an AP when various user types of "*User Model 1*" (i.e., different σ's) are considered.



(b) Two cases of Fig. 6(a), σ = 3m and σ = 6m, are picked to show the corresponding average errors of the system with the calculated a's.

Figure 6: Simulation results of how different user behaviors affect the regression coefficient a values and correspondingly, influence the system's achievable average accuracy.

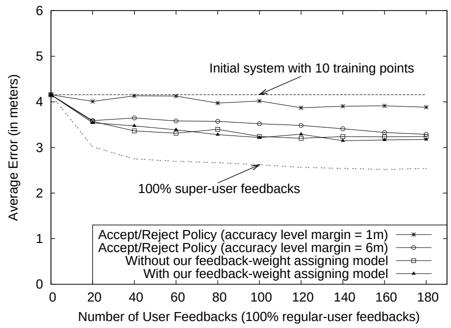
are obtained through simulation, unlike the others in this paper where real experimental data are used.

Table 1: Relationship between the Uncertainty Parameter, σ, and Average Localization Error

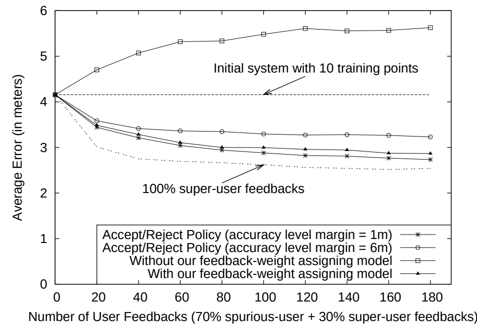
Number of User Feedbacks	Uncertainty Parameter, σ of "*User Model 1*"	Average Localization Error (in meter)
	$\sigma = 0$	3.1
	σ = 3m	3.37
500	σ = 6m	3.98
	σ = 9m	4.71
	σ = 12m	6.18

3.5.3. Fine-tuning of an existing positioning system utilizing user feedbacks

In this section, we wish to show that we could fine-tune a positioning system in order to achieve finer accuracy by exploiting our feedback-weight assigning model, irrespective of any assumption on user behavior. For this experiment, we choose two different combinations of user feedbacks where one consists of only *well-behaved* regular-user feedbacks while the other comprises of 70% spurious-user and 30% super-user feedbacks. In both cases, we assume that the positioning system is already running with some feedbacks (4 landmark feedbacks + 6 super-user feedbacks) so that we can approximate the initial "RoC Profile". Consequently, we can come up with the feedback weight-assigning model of Section 2.3.2 from this initial state of our system. We consider 137 testing points to evaluate the localization errors which are completely different from the user feedback points. As more user feedbacks become available, the weight-assigning model continuously updates itself in a similar manner as previously shown in Fig. 2(d), which helps to fine-tune the system.



(a) 100% *well-behaved* regular-user feedbacks.



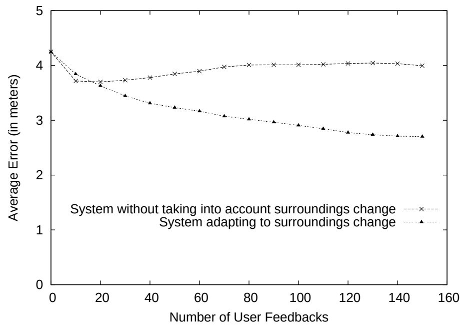
(b) 70% spurious-user feedbacks + 30% super-user feedbacks.

Figure 7: Performance comparison of our feedback-weight assigning model with other options in the fine-tuning of an under-trained positioning system.

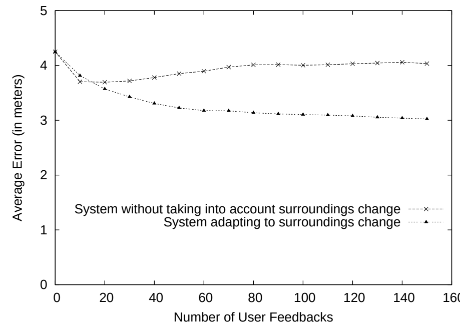
The two horizontal lines of Fig. 7(a) and 7(b) at 4.16m represent the initial system's performance with only 10 training points. Our feedback-weight assigning model shows that the system's performance improves when more feedbacks are incorporated. Without our feedback-weight assigning model, the system's performance deteriorates when spurious-user feedbacks dominate as can be seen from Fig. 7(b). For 100% spurious-user feedbacks scenario (the results are omitted for brevity), our system's performance remains relatively unchanged from the initial system's performance. This means that our feedback-weight assigning model could shield the system from the adverse effect of this type of feedbacks. For the well-behaved user case, the feedback positions may turn out to be very close to the actual positions which will eventually make them a bit similar to super-user feedbacks. The inclusion of super-user feedbacks into the system always helps regardless of whether we are using our model or not. Therefore, the "without feedback-weight assigning model" might have been seen to perform almost similar to (or even slightly better than) our model in Fig. 7(a). Our model's effectiveness over the "without feedback-weight assigning model" can be realized when different types of feedbacks are mixed (e.g., one instance can be seen in Fig. 7(b)).

We also compare the accept/reject policy of location verification techniques discussed in Section 2.3 to incorporate a user feedback with varying accuracy level margins. If the accuracy level margin is set too large ($\approx$ 6m), a number of spurious-user feedbacks may get through to the system, thereby causing it to perform worse. Setting a strict margin (e.g., 1m) may overcome this issue as can be seen from Fig. 7(b). However, if the accuracy level margin is set too strict, many of the well-behaved regular-user feedbacks are rejected. Consequently, the system's performance does not improve much over the initial system when this type of feedback dominates as revealed in Fig. 7(a). On the contrary, our feedback-weight assigning model is quite automated (no need for manual setting of accuracy level margin) and is shown to perform reasonably well in the presence of different types of feedbacks.

The 100% super-user feedback curves in both Fig. 7(a) and 7(b) show the performance when the feedbacks are given by super-users only (i.e., $w = 1$). This performance is comparable to the traditional fingerprint-based system where all the samples are collected exhaustively by administrators. This provides a performance benchmark for the user feedback based positioning system.



(a) 30% regular-user + 70% super-user feedbacks.



(b) 70% regular-user + 30% super-user feedbacks.

Figure 8: Adaptation of our system when it perceives that the surroundings have changed.

Note that we have only provided Bayesian algorithm's results; the KNN algorithm's results show similar trends, and are not included here.

3.5.4. Effect of change of surroundings on our user feedback based positioning system

One of the major drawbacks of existing fingerprint-based positioning systems is that it is not adaptable to environmental changes, i.e., the training phase has to be repeated all over again for the changed surroundings. Our system does not suffer from such shortcomings since user feedbacks are continuously employed to fine-tune it. Furthermore, our system's whole process of adapting to the changed environment is automated, and does not require any outside intervention. In order to help perceive that there is a change in the surrounding, we exploit landmark feedbacks. Since the landmark feedbacks from the APs are continuous, the system can approximate the APs' positions all the time. We infer that there is a change in surrounding when the estimated positions of all the APs deviate quite significantly from their actual positions. Algorithm 1 (in Page 39) describes the adaptation process of our positioning system. From algorithm 1, we see that, when the system perceives its surroundings to have changed, it enters into the adaptation mode. In this mode, all the previously incorporated user feedbacks are associated with an exponential outdate-factor together with their assigned weights. As a result, new user feedbacks are given more importance.

In order to emulate a change in the surrounding in our experiments, we swapped the positions of two of our APs as shown in Fig. 4. This serves our purpose of creating a changed environment since the two APs' signal strength signatures change quite significantly. Our initial system consists of 50 super-user feedbacks from the *old* setting and we utilize 137 testing points from the *new* setting to evaluate the localization errors. The two curves of Fig. 8(a) and 8(b) depict the performances of two systems where one system is incorporated with our surroundings change algorithm and the other one is not. As can be seen from the figures, the system which could realize the change in surroundings, performs significantly better in the *new* setting as more user feedbacks are incorporated into the system. For this experiment, we choose two different combinations of user feedbacks where in one scenario, the super-user feedbacks dominate (Fig. 8(a)), while in the other, the regular-user feedbacks dominate (Fig. 8(b)). In both scenarios, our system could adapt seamlessly with the surroundings change. Note that, the super-user dominating scenario demonstrates lower localization error for the same number of user feedbacks compared

Algorithm 1 Adaptation of our feedback-based positioning system

System State: A positioning engine with n samples or user feedbacks. Let $(e_{\text{ex}}, p_{\text{ex}})$ denote the system's expected RoC. It is a tunable parameter for the system administrator within which he/she expects the K landmarks' (e.g., APs') estimated positions to be verified. If all the K landmarks' estimated positions deviate from $(e_{\text{ex}}, p_{\text{ex}})$, the system infers surroundings change, and enters the adaptation mode. The system returns to normal mode again when all the K landmarks' estimated positions are within the system's expected RoC $(e_{\text{ex}}, p_{\text{ex}})$. The landmarks' positions are estimated continuously by the system from the landmark feedbacks.

for every *new* batch of N feedbacks collected **do**
    if all K APs' estimated positions deviate from $(e_{\text{ex}}, p_{\text{ex}})$ **then**
        {adaptation mode}
        $h \leftarrow \alpha$ {outdate factor: α small constant}
    else
        {normal mode}
        $h \leftarrow 0$ {no outdate factor}
    end if
    for $i = 1$ to n **do**
        $w_i \leftarrow \exp(-h) \times w_i$ {outdating older samples' weights if $h \neq 0$}
    end for
    $n \leftarrow n + N$
    calculate the interpolated RSS signatures as discussed in Appendix B
    run localization algorithm (e.g., Bayesian or KNN) with only the feedbacks having $w_i \geq \gamma$ as test samples among the *new* feedbacks {w_i and γ are defined in 2.3.2}
    update feedback-weight assigning model's parameters (i.e., $e_{\text{min}}, e_{\text{av}}, e_{\text{s}}$ and e_{max}) of Section 2.3.2.
end for
goto 1

to the regular-user dominating scenario which is justifiable. The presence of spurious-user feedbacks also does not affect the adaptation process (the results are omitted for brevity). This is due to our feedback-weight assigning model which is found to be successful in dealing with them in the previous section. We have also observed in our experiments that around $20 \sim 30$ user feedbacks are required for the system to return to its normal mode (i.e., to leave its adaptation mode). This state transformation occurs when the landmark feedbacks start to give better estimations of the APs' positions again.

4. Related Work

The current research efforts for indoor positioning systems can largely be divided into two main categories:

- Those that make use of angle of arrival (AoA), time of arrival (ToA), and time difference of arrival (TDoA) methodologies. This family of localization techniques relies on specialized hardware (e.g., RF tags, ultrasound or infrared receivers, etc.) and extensive deployment of dedicated infrastructure solely for localization purpose [21, 22, 23].

- Those that utilize the correlation between easily measurable signal characteristics (e.g., RSS) and location. These location fingerprinting solutions try to build a positioning system on top of existing infrastructure (e.g., Bluetooth networks) [2, 3, 4, 15] in a cost-effective way.

Our work focuses on fingerprint-based positioning systems, i.e., the second category above. Due to space limitation, we only provide an overview about some existing approaches under this category. Avid readers may consult [13, 24] for more in-depth discussions. Location fingerprinting techniques became popular with RADAR [2], mainly because of the unavailability of appropriate radio signal propagation models for indoor environments. It also opened the door for many different approaches to be applied for the indoor localization problem. For example, Nibble [3] is one of the first systems to use a probabilistic approach for location estimation. To date, Ekahau's Positioning Engine Software [25] claims to be the most accurate location system based on probabilistic model; they claim a one-meter average accuracy with a short training period. Battiti *et al.* applied statistical learning theory [26] and neural networks [27] to tackle the indoor localization problem.

The application of user feedback to incrementally build or fine-tune an indoor positioning system has not been sufficiently explored in existing literature. Although UCSD's ActiveCampus project tries to build a positioning system incrementally by incorporating user-based survey mechanisms [5], their user inputs are treated no differently from an administrator's fingerprinting data, and are accepted with complete trust. As shown earlier in Fig. 8(b), such a positioning system in which all user inputs are assigned a feedback-weight of w = 1 by default would suffer in terms of accuracy when spurious user feedbacks are aplenty.

We discussed some location verification techniques (i.e., positioning system verifying the location claimed by a source node) [8, 9, 10, 11] in Section 2.3, and outlined their viability when adopting them for our system. These verification techniques are pertinent to our work since our system's feedback-weight assigning model determines the *credibility* (i.e., weight w) of a feedback, which is similar to verifying a user's claim (in case of explicit feedback) or the system's claim of a user's location (in case of implicit feedback). We also compare our feedback-weight assigning model's performance with the accept/reject policy of location verification techniques in Fig. 7(a) and 7(b), and discuss the results in Section 3.5.3. From the results, we observe that, it is hard to set a unique tolerance level (i.e., the margin within which the feedbacks are accepted) of the accept/reject policy in order to make it work efficiently across various mixtures of user feedbacks. On the contrary, our model turns out to be robust across different mixtures of user feedbacks.

5. Conclusion and Future Work

In this paper, we propose a novel idea where users can take part in fine-tuning an under-trained positioning system. Our feedback-weight assigning model which assigns *relative* weights to user feedbacks, fine-tunes an under-trained positioning system, thereby, helps it to achieve finer accuracy. We also show that, if users are *well-behaved*, we can actually construct a positioning system incrementally from scratch exploiting our interpolation-based techniques with the user feedbacks. We contend that the exhaustive training phase seen in the traditional location fingerprinting techniques might be relieved through it. Through the use of landmark feedbacks, we could successfully infer changes in the environment, and switch our system's mode to be more adaptable. The whole procedure is quite dynamic, and requires no intervention from the positioning system administrator's part.

In summary, we conclude that our user feedback-based positioning system is fairly accurate, cost-effective, robust and requires no or very little training phase. In the following, we list the limitations of our work, and also mention the future directions that we foresee:

- We have implemented our system in two testbeds – one is placed inside an amphitheater while the other is within a research laboratory. Our system performed quite well in both scenarios as can be seen from the results. It should also be verified with other types of testbeds that introduce more multipath effects and obstructions in order to be more conclusive.
- Our current positioning system is based on Bluetooth wireless technology, but it can easily be extended to accept feedbacks from devices using other technologies as well (e.g., Wi-Fi).
- We actually performed an offline training phase like a traditional fingerprinting system to collect the experimental data ourselves. Subsequently, these data are amalgamated with uncertainties to emulate the various user models as discussed in Section 3.3. The practical deployment of our system would allow us to obtain feedbacks from different types of users who may use the positioning service. This could potentially allow us to model real users' behaviors more accurately.
- A positioning system exploiting smart ways to obtain implicit feedbacks from users (e.g., [6]) may eliminate the need for explicit feedbacks altogether. This approach might even have greater potential rather than solely depending on users' goodwill for obtaining feedbacks. The practical deployment of such a system may fetch interesting results in this regard.

A. Proof of Theorem 1

The linear combination $\sum_{i=1}^{n} a_{ki} S_{ki}$ is an unbiased estimate of $E(S_k)$ because, $E(\sum_{i=1}^{n} a_{ki} S_{ki}) = \sum_{i=1}^{n} a_{ki} E(S_{ki}) = E(S_k) \sum_{i=1}^{n} a_{ki} = E(S_k)$. Since the estimate is unbiased, then the particular combination that is most efficient is the one which minimizes the variance, $\mathrm{var}(\sum_{i=1}^{n} a_{ki} S_{ki}) = \sum_{i=1}^{n} {a_{ki}}^2 \,\mathrm{var}(S_{ki}) = \mathrm{var}(S_k) \sum_{i=1}^{n} {a_{ki}}^2$. Consequently, the problem can be reformulated as, minimize $\sum_{i=1}^{n} {a_{ki}}^2$ subject to $\sum_{i=1}^{n} a_{ki} = 1$. Now, using basic optimization theory, it directly follows that the particular linear combination $\frac{1}{n} \sum_{i=1}^{n} S_{ki}$, or the sample *mean*, $\bar{S}_k$, is the most efficient unbiased estimator of $E(S_k)$.

B. Calculation of RSS at Interpolated Training Points

Suppose, there are n user feedbacks for which the real measurements of RSSs have been taken at the K APs. Our goal is to emulate the RSSs of K APs for J possible interpolated training points utilizing those real measurements of user feedbacks.

As in [17], the linear regression RSS prediction formula based on the log-normal shadowing model takes the following form,

$$\hat{y}_{ki} = a_k x_{ki} + b_k, \tag{3}$$

where $\hat{y}_{ki}$ = the predicted RSS of the k^{th} AP when the MN is at i^{th} training point, $a_k = -10\beta$, $x_{ki} = \log(d_{ki})$ and $b_k = P_{\text{r}}(d_0)|_{\text{dBm}} + 10\beta\log(d_0)$. Utilizing both the spatial similarity and user feedback credibility factors, the weighted least mean square minimization function for our linear regression model can be written as,

$$R^2(a_{jk}, b_{jk}) = \sum_{i=1}^{n} c_{ji}[y_{ki} - (a_{jk}x_{ki} + b_{jk})]^2 \tag{4}$$

where y_{ki} = real measurement of RSS at the k^{th} AP when the MN is at i^{th} training point, $x_{ki} = \log(d_{ki})$ = log distance of k^{th} AP from the i^{th} training point, $c_{ji} = \frac{u_{ji}\times v_i}{\sum_{i=1}^{n} u_{ji}\times v_i}$, u_{ji} = normalized weight considering spatial similarity of RSS = $\frac{1/d_{ji}}{\sum_{i=1}^{n} 1/d_{ji}}$, v_i = normalized weight for i^{th} feedback considering its credibility = $\frac{w_i}{\sum_{i=1}^{n} w_i}$, d_{ji} = distance of interpolated point j from the i^{th} training point, a_{jk}, b_{jk} = regression coefficients of the linear RSS prediction formula of the k^{th} AP for j, $k \in \{1, 2, \ldots, K\}$, and $j \in \{1, 2, \ldots, J\}$. Note that, depending on the interpolated point j, for which the RSS will be predicted, the associated weight of spatial similarity factor (i.e., u_{ji}) changes. Hence, an additional subscript is used in (4) to denote the regression coefficients for an AP w.r.t. different interpolated points compared to (3).

Denote,

$$\mathcal{Y}_k = \begin{bmatrix} y_{1k} \\ y_{2k} \\ \vdots \\ y_{nk} \end{bmatrix}, \ \mathcal{X}_k = \begin{bmatrix} 1 & x_{1k} \\ 1 & x_{2k} \\ \vdots & \vdots \\ 1 & x_{nk} \end{bmatrix}, \ \mathcal{C}_j = \begin{bmatrix} c_{j1} & 0 & 0 & \ldots & 0 \\ 0 & c_{j2} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & c_{jn} \end{bmatrix} \text{ and } \mathcal{B}_{jk} = \begin{bmatrix} b_{jk} \\ a_{jk} \end{bmatrix}.$$

Using these matrix notations, now we differentiate (4) w.r.t. $\mathcal{B}_{jk}$ and set it to zero,

$$\begin{aligned}
& \frac{\partial}{\partial \mathcal{B}_{jk}} \left[\left(\mathcal{Y}_k - \mathcal{X}_k \mathcal{B}_{jk} \right)^T \mathcal{C}_j \left(\mathcal{Y}_k - \mathcal{X}_k \mathcal{B}_{jk} \right) \right] = 0 \\
\Rightarrow \quad & \frac{\partial}{\partial \mathcal{B}_{jk}} \left[\left(\mathcal{Y}_k{}^T - \mathcal{B}_{jk}{}^T \mathcal{X}_k{}^T \right) \mathcal{C}_j \left(\mathcal{Y}_k - \mathcal{X}_k \mathcal{B}_{jk} \right) \right] = 0 \\
\Rightarrow \quad & \frac{\partial}{\partial \mathcal{B}_{jk}} \left[\mathcal{Y}_k{}^T \mathcal{C}_j \mathcal{Y}_k - \mathcal{B}_{jk}{}^T \mathcal{X}_k{}^T \mathcal{C}_j \mathcal{Y}_k - \mathcal{Y}_k{}^T \mathcal{C}_j \mathcal{X}_k \mathcal{B}_{jk} + \mathcal{B}_{jk}{}^T \mathcal{X}^T \mathcal{C}_j \mathcal{X}_k \mathcal{B}_{jk} \right] = 0 \\
\Rightarrow \quad & 2\mathcal{B}_{jk}{}^T \mathcal{X}_k{}^T \mathcal{C}_j \mathcal{X}_k - 2\mathcal{Y}_k{}^T \mathcal{C}_j \mathcal{X}_k = 0 \\
\Rightarrow \quad & \mathcal{X}^T \mathcal{C}_j{}^T \mathcal{X}_k \mathcal{B}_{jk} = \mathcal{X}_k{}^T \mathcal{C}_j{}^T \mathcal{Y}_k.
\end{aligned}$$

If the matrix $\left(\mathcal{X}^T \mathcal{C}_j{}^T \mathcal{X}_k \right)$ is non-singular, the regression coefficients are given by the formula,

$$\mathcal{B}_{jk} = \left(\mathcal{X}^T \mathcal{C}_j{}^T \mathcal{X}_k \right)^{-1} \mathcal{X}_k{}^T \mathcal{C}_j{}^T \mathcal{Y}_k. \tag{5}$$

For a particular interpolated point j, the regression coefficients $\mathcal{B}_{jk}$ of the k^{th} AP's signals can be obtained through (5). Consequently, the RSS of the k^{th} AP for an interpolated point j can be emulated as,

$$\text{RSS}_{jk} = a_{jk} \log d_{jk} + b_{jk}. \tag{6}$$

Plugging the values of a_{jk}, b_{jk} and d_{jk} (the distance of the interpolated point j from k^{th} AP) into (6), we finally obtain the RSS fingerprint for j considering only AP k. To deduce the RSS vector comprising of all the K APs for a particular interpolated point j, we have to follow the same procedure for all $k \in \{1, 2, \ldots, K\}$. Finally, in order to obtain the RSS vector of the K APs for all the J interpolated points over the localization area, we have to repeat the whole calculation of this section for all $j \in \{1, 2, \ldots, J\}$.

Note that, when all user feedbacks are believed equally, we have, $c_{ji} = \frac{u_{ji} \times \frac{1}{n}}{\sum_{i=1}^{n} u_{ji} \times \frac{1}{n}} = \frac{u_{ji}}{\sum_{i=1}^{n} u_{ji}} = u_{ji}$. In other words, only spatial similarity weight factor would be taken into consideration in calculating the RSS signatures of the interpolated points.

References

[1] T. Kindberg, A. Fox, System software for ubiquitous computing, IEEE Pervasive Computing 1 (1) (2002) 70–81.

[2] P. Bahl, V. N. Padmanabhan, RADAR: An in-building RF-based user location and tracking system, in: Proc. IEEE INFOCOM, Israel, 2000, pp. 775–784.

[3] P. Castro, P. Chiu, T. Kremenek, R. R. Muntz, A probabilistic room location service for wireless networked environments, in: Proc. UbiComp'01, Atlanta, GA, 2001, pp. 18–34.

[4] M. A. Youssef, A. Agrawala, A. U. Shankar, WLAN location determination via clustering and probability distributions, in: Proc. PERCOM, 2003.

[5] E. S. Bhasker, S. W. Brown, W. G. Griswold, Employing user feedback for fast, accurate, low-maintenance geolocationing, in: Proc. PerCom, 2004, pp. 111–120.

[6] D. Papadogkonas, G. Roussos, M. Levene, Analysis, ranking and prediction in pervasive computing trails, in: Proc. IET Intelligent Environments (IE08), Seattle, 2008.

[7] K. Kaemarungsi, P. Krishnamurth, Properties of indoor received signal strength for WLAN location fingerprinting, in: Proc. MobiQuitous'04, San Diego, CA, 2004, pp. 14–23.

[8] N. Sastry, U. Shankar, D. Wagner, Secure verification of location claims, in: Proc. ACM WiSe '03, NY, USA, 2003, pp. 1–10.

[9] D. Chaum, Distance-bounding protocols (extended abstract), in: In Theory and Application of Cryptographic Techniques, 1993, pp. 344–359.

[10] D. Singelee, B. Preneel, Location verification using secure distance bounding protocols, in: Proc. IEEE MASS, Washington, USA, 2005.

[11] M. D. Corner, B. D. Noble, Zero-interaction authentication, in: Proc. MobiCom '02, New York, NY, USA, 2002, pp. 1–11.

[12] Y. Gwon, R. Jain, T. Kawahara, Robust indoor location estimation of stationary and mobile users, in: Proc. IEEE INFOCOM'04, 2004, pp. 1032–1043.

[13] J. Hightower, G. Borriella, Location systems for ubiquitous computing, IEEE Computer 34 (8) (2001) 57–66.

[14] E. Elnahrawy, X. Li and and R. P. Martin, The limits of localization using signal strength: A comparative study, in: IEEE SECON, Santa Clara, CA, 2004.

[15] E. Elnahrawy, X. Li and and R. P. Martin, Using area-based presentations and metrics for localization systems in wireless lans, in: Proc. IEEE WLN, 2004.

[16] BlueZ, Official linux bluetooth protocol stack.
URL `http://www.bluez.org`

[17] M. Hossain, H. Nguyen Van, Y. Jin and W.-S. Soh, Indoor localization using multiple wireless technologies, in: Proc. IEEE MASS, Pisa, Italy, 2007.
URL `http://www.ece.nus.edu.sg/stfpage/elesohws/mass07.pdf`

[18] P. Tao, A. Rudys, A. M. Ladd, D. S. Wallach, Wireless LAN location-sensing for security applications, in: Proc. ACM WiSe'03, San Diego, CA, 2003, pp. 11–20.

[19] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, L. E. Kavraki, Practical robust localization over large-scale 802.11 wireless networks, in: Proc. ACM MobiCom'04, Philadelphia, PA, 2004, pp. 70–84.

[20] M. B. Kjærgaard, C. V. Munk, A calibration-free solution for handling differences in signal strength, in: Proc. IEEE PerCom'08, 2008.

[21] R. Want, A. Hopper, V. Falcão, J. Gibbons, The active badge location system, ACM Trans. on Information Systems 10 (1) (1992) 91–102.

[22] A. Ward, A. Jones, A. Hopper, A new location technique for the active office, IEEE Personal Communications 4 (5) (1997) 42–47.

[23] N. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in: Proc. MobiCom, 2000, pp. 32–43.

[24] K. Pahlavan, X. Li, J. Maleka, Indoor geolocation science and technology, IEEE Communications Mag. 40 (2) (2002) 112–118.

[25] Ekahau, Wi-fi rtls and site survey solutions.
URL http://www.ekahau.com/

[26] R. Battiti, M. Brunato, A. Villani, Statistical learning theory for location fingerprinting in wireless LANs, Tech. Rep. DIT-02-0086, Universita di Trento, Dipartimento di Informatica e Telecomunicazioni (Oct. 2002).

[27] R. Battiti, T. L. Nhat, A. Villani, Location-aware computing: a neural network model for determining location in wireless LANs, Tech. Rep. DIT-02-0083, Universita di Trento, Dipartimento di Informatica e Telecomunicazioni (Feb. 2002).