

RESEARCH

Open Access

Enhancing Wi-Fi fingerprinting for indoor positioning using human-centric collaborative feedback

Yan Luo¹, Orland Hoerber² and Yuanzhu Chen^{2,3*}*Correspondence: yzchen@mun.ca²Department of Computer Science,
Memorial University of
Newfoundland, St. John's, NL,
Canada³Department of Computer Science,
The University of Regina, Regina, SK,
CanadaFull list of author information is
available at the end of the article

Abstract

Position information is an important aspect of a mobile device's context. While GPS is widely used to provide location information, it does not work well indoors. Wi-Fi network infrastructure is found in many public facilities and can be used for indoor positioning. In addition, the ubiquity of Wi-Fi-capable devices makes this approach especially cost-effective.

In recent years, "folksonomy"-like systems such as Wikipedia or Delicious Social Bookmarking have achieved huge successes. User collaboration is the defining characteristic of such systems. For indoor positioning mechanisms, it is also possible to incorporate collaboration in order to improve system performance, especially for fingerprinting-based approaches.

In this article, a robust and efficient model is devised for integrating human-centric collaborative feedback within a baseline Wi-Fi fingerprinting-based indoor positioning system. Experiments show that the baseline system performance (i.e., positioning error and precision) is improved by collecting both positive and negative feedback from users. Moreover, the feedback model is robust with respect to malicious feedback, quickly self-correcting based on subsequent helpful feedback from users.

Introduction

After over a decade of research and development, location-aware services have gradually penetrated into real life. They assist human activities in a wide range of applications, from productivity and goal fulfillment to social networking and entertainment. Traditionally, location-aware applications have been confined to outdoor environments. Relatively less research has explored the potential applicability of similar services for indoor settings. However, in large indoor environments such as airports, libraries, or shopping centres, location-awareness can increase the quality of service provided by these facilities.

Large scale deployment of indoor location-awareness is much more difficult due to two technical challenges. First, GPS can not be deployed for indoor use because GPS signals can not reach indoor receivers. Second and more importantly, due to complicated indoor environments such as building geometries, the movement of people, and the random effects of signal propagation, triangulation-based approaches are much less effective [1]. In addition, interference and noise from other devices can also degrade the accuracy of positioning. On the other hand, such challenges provide researchers with great

opportunities for innovative indoor positioning techniques. Some early indoor positioning technologies used infrared, laser, and/or ultrasonic range finders, yielding fairly good system performance in field tests [2]. The disadvantages of such an approach are its size, complexity, and cost, which render it infeasible for mobile devices.

A number of researchers have been working on using Wi-Fi infrastructure for indoor positioning, even though it was not specifically designed for this purpose [3]. Due to the infeasibility of indoor triangulation, most of these systems use a fingerprinting approach based on the Received Signal Strength (RSS) transmitted by nearby Wi-Fi access points [3]. Typically, such an approach consists of a training phase and a positioning phase. In the training phase, each survey position is characterized by location-related Wi-Fi RSS properties called *Wi-Fi RSS fingerprints* [4]. During the positioning phase, the position likelihood is calculated based on the current Wi-Fi RSS measurements.

For Wi-Fi fingerprinting, fine-grained system training is normally required to achieve high accuracy and resolution. This results in significant costs in terms of initial configuration and ongoing maintenance in order to continuously adapt to environmental changes and Wi-Fi infrastructure alterations. Such alterations are not uncommon due to system malfunctions, equipment upgrades, or simply turning on and off Wi-Fi access points controlled by individual users. A great deal of effort has been made by researchers to reduce such costs. A potentially effective way is to let users provide feedback to facilitate the construction and continual maintenance of the RSS fingerprints database.

In this article, we propose a Wi-Fi based indoor positioning system that includes an integrated human-centric collaborative feedback model. In the proposed prototype, we define an efficient and robust user feedback model, where the initial likelihood distribution calculated by the positioning system will be compensated before being presented to the user. Further, the user can participate in how the compensation works in the future by providing feedback.

The rest of this article is organized as follows. An overview of the related work is provided in Section 'Related work'. In Section 'Baseline Wi-Fi fingerprinting indoor positioning system', we describe a baseline Wi-Fi fingerprinting framework. The details of the proposed user feedback model are explained and discussed in Section 'Human-Centric collaborative feedback model'. This user feedback model is tested and evaluated in comparison to the baseline method in Section 'Evaluation'. This article concludes in Section 'Conclusions and future work' with a summary of the primary contributions of this work and an overview of future work.

Related work

The Wi-Fi-based positioning technology has a very promising application prospect mainly because of the ubiquitous and inexpensive nature of Wi-Fi infrastructure. Also, Wi-Fi is widely used and integrated in various electronic devices. Thus, the Wi-Fi based positioning systems can also reuse these mobile devices as tracking targets to locate users, which is a less intrusive way to provide location-aware services.

Wi-Fi fingerprinting-based indoor positioning

Due to the infeasibility of signal propagation model-based distance estimation, more and more researchers have employed a Wi-Fi fingerprinting-based approach, which is more robust, accurate, and cost-effective in real indoor environments. However, its system

performance is highly dependent upon the elaborate training process and ongoing maintenance efforts. Also, in the positioning phase, random propagation effects of signal propagation introduced by complex indoor environments may result in large RSS fluctuations or access point (AP) loss [5] (i.e., APs which cannot be heard), which could cause the fingerprints in the database to become inefficient and result in large positioning errors. These shortcomings not only imply a high system overhead and training cost, but also vulnerability to environmental alteration. However, if such a system is enhanced with a self-learning ability adapting it to the environmental changes, such inaccurate positioning outcomes can be compensated. For mobile devices carried by people, such a self-learning capability and positioning compensation could come from end users for free. Users could provide feedback to the positioning service based on their knowledge of the surroundings. They may choose to accept, reject, or supply specific information (i.e., their known location) to modify the system results after being given the estimated position.

Human feedback to fingerprinting-Based positioning process

Active Campus [6] is an early system integrating user feedback. It allows users to update the training data incrementally for future use. When the system location is incorrect, users can click on the correct location and suggest new positions. Similarly, Redpin [7] uses a “folksonomy”-like approach, where many users train the system while using it. Gallagher et al. [8] focus on the adaptation of Wi-Fi infrastructure alteration. They investigate a new method to utilize user feedback as a way of monitoring changes in the wireless environment. Users are prompted to send their RSS measurements to a remote positioning server. The server can then update the Wi-Fi RSS fingerprints in the database based on the observations from the users.

Park et al. [9] propose a user promotion mechanism. They argue that in a human-centric positioning system, it is useful to only prompt users for their location when the system error is large. They propose a mechanism to convey the system’s spatial confidence in its prediction based on a *Voronoi Diagram*, and the system only prompts users whenever its confidence falls below a threshold. Therefore, the size of the Voronoi cell naturally represents the spatial uncertainty associated with prediction of the bound space. Once the size of the current Voronoi cell is beyond a threshold, the system will prompt users to provide feedback.

The above approaches refine the existing Wi-Fi RSS fingerprint-based positioning system with the integration of human-centric feedback. However, a potential pitfall is that the model constructed during the training phase could also be negatively affected by unreliable or misleading user feedback. Thus, it is crucial that the feedback from users should be given proper weight or credibility, rather than blind acceptance or rejection. Hossain et al. [10] propose a simple credibility rating. In their system, a user’s estimation is given a higher credibility weight if the suggested position has a small discrepancy with the system. In fact, according to the observation of our preliminary experiments, the system results are mostly close to user’s true position. However, they are occasionally very far away from the true position due to insufficient Wi-Fi RSS data or large variance. In that case, if user’s feedback follows the system’s estimation and is assigned a high weight, it in fact becomes an outlier feedback and could bring large interference to future positioning queries. Such negative effects from outlier user feedback should be eliminated. A straightforward solution could be using clustering algorithms to filter outliers [9].

In this article, we will discuss a more general and efficient framework using a wider variety of user feedback. Such a framework is endowed with a high degree of system robustness when a large number of users provide correct feedback. Even when incorrect feedback is provided, the system is able to quickly recover by incorporating subsequent corrective feedback.

Baseline Wi-Fi fingerprinting indoor positioning system

We start by first introducing a baseline Wi-Fi fingerprinting system. The implementation of this baseline system is similar in many respects to the systems in the literature [11]. However, it is also refined to be more robust and suitable for integrating and processing user feedback.

Training phase

The system training is conducted for each survey point in a two-step process. The first step is to collect multiple Wi-Fi scans in order to stabilize the average of RSS readings and calculate the variances. The variance is used to detect the environmental interference level, where a large variance tends to cause unreliable positioning results. The following step utilizes the information collected by these Wi-Fi scans to generate an RSS fingerprint for each survey position.

Collect raw Wi-Fi RSS data

At each survey position, system administrators use a mobile device to scan for the beacon frames transmitted by nearby Wi-Fi APs. In each Wi-Fi scan, beacon frames from different APs are received and converted to a list of 3-tuples, which contains the MAC address of an AP, the RSS in dBm, and timestamp. Note that a single scan may not be able to capture beacon frames from all nearby APs due to the different beacon frame broadcasting periods or severe signal fading. Also, the collected RSS values have a natural variation when indoors, which is unavoidable. To compensate the RSS fluctuation and obtain complete AP information, a sufficiently large number of scans is needed to create an RSS fingerprint. As a result, in a given period of sampling, the device logs a time series of RSS vectors. These vectors are then used to construct the Wi-Fi RSS fingerprints for each measured location in the training grid.

Generate system anchors

The statistics are extracted from the raw Wi-Fi measurement data to generate an Wi-Fi RSS fingerprint for each survey position.

A Wi-Fi RSS *fingerprint* is defined as a vector of 5-tuples (i.e., *MAC*, *Timestamp*, *RSS Mean*, *Count*, and *RSS Variance*), describing a set of APs. The definition and explanation for each field are given as follows.

Given the i -th AP in a Wi-Fi RSS fingerprint, each AP determines one dimension of such a vector:

- *MAC*: The MAC field contains its MAC address, denoted as M_i .
- *Timestamp*: The time of creating the fingerprint is stored in the Timestamp field, denoted as t .
- *RSS Mean*: The RSS Mean \bar{r}_i is an average of the Wi-Fi RSS over the sampling period.

- *Count*: The value of Count is the number of occurrences of the AP during the sampling period, denoted C_i , which is a very important indicator for the reliability of this AP. For a fixed number of Wi-Fi scans, a large Count value means that the AP can be heard for most of the time, indicating that the AP will have a more reliable estimation of its RSS value.
- *RSS Variance*: RSS Variance contains the variance of the measured RSS from the AP, denoted σ_i . The fluctuation level of the current Wi-Fi environment at a certain survey position can be estimated by analyzing the Wi-Fi RSS fingerprint. Each AP has its own mean and variance, which can not provide a global description about the current Wi-Fi environment. In order to estimate the fluctuation level of the entire environment, we use the weighted average of RSS Variance for each AP. The occurrence or the value in the Count field for each AP is utilized as the weight. The collective RSS variance for this fingerprint is defined as

$$\sigma_{F_s} = \frac{\sum_{i \in F_s} \sigma_i C_i}{\sum_{i \in F_s} C_i},$$

where F_s is its RSS fingerprint.

At the end of the training phase, each survey position is associated with an RSS fingerprint containing APs that describe the specific location. For each survey position \mathcal{P}_s in the system, we define a *system anchor* A_s as

$$(\mathcal{P}_s, F_s)$$

The system anchors are reference points to determine the positions of mobile devices.

Note that it is quite possible for the RSS measurements to vary throughout the day, based on cyclical activities such as the number of people within the building, their use of electronic devices, etc. In order to carefully explore the benefits of the core contribution of this work (i.e., the inclusion of human-centric feedback within the positioning process), we found it necessary to simplify the problem domain and assumed that the RSS measurements are stable over time. As a result, we conducted all testing and experimentation at a consistent time of day to avoid temporal-based variances in the RSS measurements. A further discussion on how to extend this work to the more realistic situations of time-varying RSS measurement is provided in Section ‘Conclusions and future work’.

Positioning determination phase

In the positioning phase, live Wi-Fi measurements will be collected and used to query the fingerprint database. Using only a few Wi-Fi scans during the positioning phase may generate a large error due to lack of informative RSS data. For experimental purposes, the prototype implementation allows for a variable number of Wi-Fi scans to evaluate system performance (Figure 1).

Suppose the total Wi-Fi scan number is S and the i -th scan will generate an RSS vector R_i , $i \in \{1, 2, 3, \dots, S\}$. Given N system anchors, when the first RSS vector is formed, we use it to calculate the likelihood L_j , $j \in \{1, 2, 3, \dots, N\}$ of it matching the fingerprint for each system anchor. Each subsequent scan should lead to a cumulative estimation result with a decreasing error. As such, the estimated result will become more and more reliable as more RSS vectors are used.

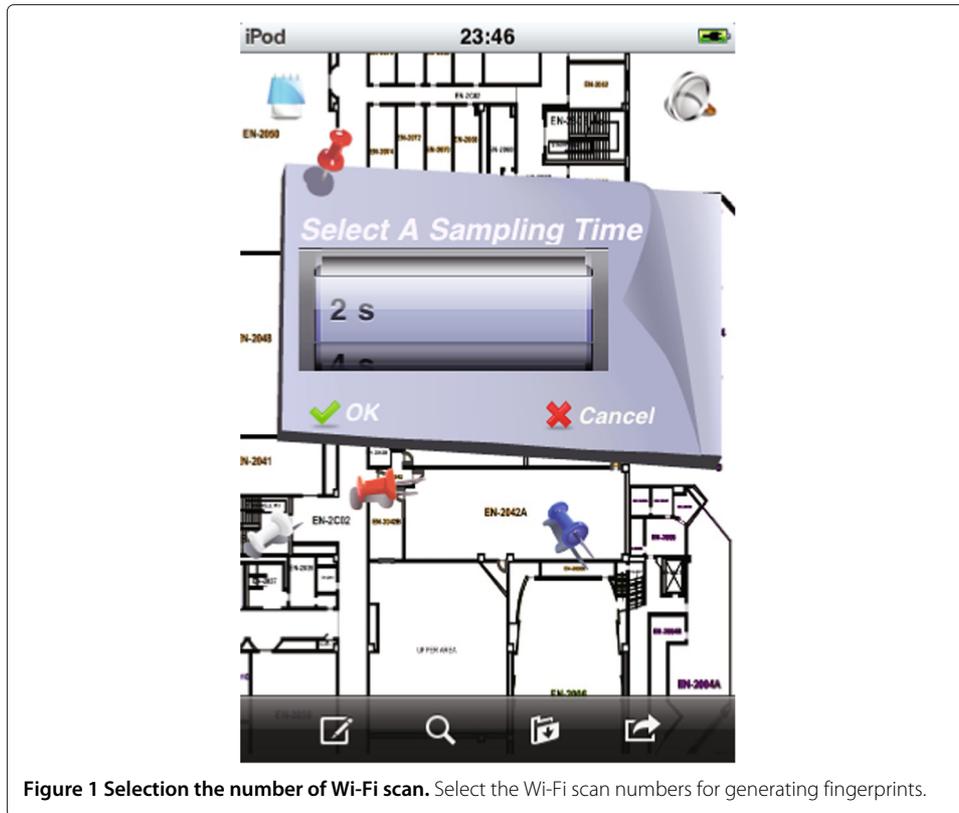


Figure 1 Selection the number of Wi-Fi scan. Select the Wi-Fi scan numbers for generating fingerprints.

Position likelihood distribution

In terms of our baseline system, we use sparse vectors containing all n APs and a Gaussian kernel to calculate the likelihood for each system anchor, which is robust and efficient according to the results of our preliminary experiments.

The Gaussian kernel method was originally used in support vector machines (SVM) to classify data [12], and has also been found to be very efficient for RSS vectors likelihood calculation [11,13,14].

Given an RSS live measurement (observation) vector generated at location \mathcal{P} as $R_{\mathcal{P}_s}$, the resulting likelihood estimate between $R_{\mathcal{P}_s}$ and fingerprint F_i in system anchor A_{s_i} is the sum of n equally weighted density functions

$$L(R_{\mathcal{P}_s}, F_i) = \sum_{k=1}^n K_G(r_{M_k}; r_{F_k}),$$

where r_{M_k} is the RSS of k -th AP in the live measurement vector $R_{\mathcal{P}_s}$ and r_{F_k} is the RSS Mean of k -th AP with the same MAC address in fingerprint F_i . Note that when r_{M_k} or r_{F_k} is an impossible value (e.g., -100 dBm), we just ignore this dimension. K_G denotes the Gaussian kernel or radial basis function (*Gaussian RBF*), whose value depends on the distance from the centre. It is given as

$$K_G(r_{M_k}; r_{F_k}) = \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{(r_{M_k} - r_{F_k})^2}{2\delta^2}\right),$$

where δ is an adjustable parameter that determines the width of the Gaussian kernel and the centre is r_{F_k} .

In terms of Wi-Fi RSS, whose value domain is $[-90dBm, -30dBm]$, δ less than 0.05 or greater than 0.5 could lead to a weak discrimination ability of Gaussian RBF. In the particular environment, we have to tune the δ value in order to archive adequate system performance.

After the likelihood calculation, each system anchor has a likelihood for being the true position of the device. Instead of just returning a single estimation, the system selects the top- k system anchors as candidates in order to provide redundant true position information. The main reason is that the true position may not always be in the system anchor with the highest likelihood. The next step is to choose a representative from these top- k candidates as the system's estimation of the position.

Position selection

A naïve approach would be to use the weighted mean of the top- k anchors as the estimation for the position. However, if one or more outliers exist, the weighted mean position could be pulled far away from the cluster formed by other system anchors. As a result, this mean position could be a meaningless point in the physical space.

Instead, we can use an approach to the vertex p -centres problem [15] to determine the representative of the top- k anchors. It is a computationally expensive problem for general k . However, in our case, we only consider the case of $p = 1$, i.e., the 1-centre problem. Since the value of k could be very small (less than five), we do not analyze the algorithm complexity at this point.

In particular, the vertex 1-centre for our positioning system is the system anchor point that minimizes the maximum distances from itself to the other top- $(k - 1)$ anchor points. These distances are weighted with the likelihood estimated as above. For two indices $i, j = 1, 2, \dots, k$, we minimize the following over all values for i

$$\max_{j \neq i} \frac{D(i, j)}{L_i},$$

where $D(i, j)$ is the Euclidean distance between anchors A_{S_i} and A_{S_j} and L_i is the likelihood of A_{S_i} . By choosing the vertex 1-centre, the resulting anchor takes advantages of both its likelihood and the positioning information shared by other top- k anchors.

Human-Centric collaborative feedback model

Before discussing the user feedback model in detail, it is useful to begin by identifying three types of user input that can be collected within a human-centric collaborative feedback system:

- *Positive feedback* is generated when users reject the estimated position and suggest a location based on their knowledge. In such a case, the system can accept the updated information from the users. The result is that the system may create new anchors from the users' suggestions, called *user anchors*.
- *Negative feedback* indicates that the users do not believe the estimated position, and are unable to make any suggestion as to their current location. In this case, the system should reduce the positioning likelihood of the returned location in the future.
- *Null feedback* occurs when users choose not to provide any feedback. The assumption here is that the estimated position is accurate, and that there is no need to make any modification to the positioning model.

Next, we will present the general idea of our user feedback model. Assume that the model has N (system and user) anchors, and the likelihood of the i -th ($i = 1, 2, \dots, N$) anchor is denoted as L_i . Before ranking these anchors based on the likelihood vector L , our user feedback model compensates each L_i with two factors, α_i and β_i as

$$L' = \begin{cases} \beta_i L_i & \text{if } A_i \text{ is a system anchor, and} \\ \alpha_i \beta_i L_i & \text{if } A_i \text{ is a user anchor} \end{cases}$$

Due to the temporal or permanent random interfering factors of complex indoor environments, the reliability of system anchors will be reducing. In order to solve this problem, we design the β factor to gradually reduce the likelihood of system anchors as negative feedback is received. As mentioned before, the system estimation is provided by the vertex-1 centre of top- k anchors. However, if this estimation receives negative user feedback, this means that the user believes that they are not near this location which is an indication that the data stored for these top- k anchors may not be accurate. As a result, the model reduces their likelihood by updating the β factors for these top- k anchors. If more and more users provide negative feedback on a system anchor, it may never be selected as one of the top- k anchors. The β factor thus gives the system an ability to forget outdated or unreliable knowledge.

On the other side, new knowledge (user anchors) will be added into the database via positive user feedback. However, when a user anchor is first created, its likelihood is reduced by the discounting effect of the small initial α value. The rational is that the system can not assess the reliability or credibility of a newly created user anchor (which may be from a malicious user). However, as more and more similar user anchors are generated to confirm it, its α factor will be increased. Once some user anchors become sufficiently reliable, they may appear to be within the top- k anchors to affect the system estimation. Also, the β factor could affect user anchors should they receive negative feedback. The user anchor and α factor enable the system to absorb new knowledge about the Wi-Fi environment.

As such, future users can take advantage of the knowledge shared by previous users. Also, they are encouraged to provide feedback to benefit subsequent users. As a result, the positioning model can be consistently updated via the user feedback model thus designed. Later in this section, we will explain how to calculate the α and β factors in detail.

Positive feedback

Suppose likelihood calculation is finished, and each system anchor A_{s_i} , ($i \in \{1 \dots N\}$) has a likelihood value L_i . For positive user feedback, users try to tell the system their estimations by providing suggestion positions. Note that these estimations could be close to the true position (accurate feedback) or still far away from it (inaccurate feedback).

Whenever the system receives a user-suggested location associated with its current RSS measurement, denoted as *user fingerprint*, the system creates a temporary user anchor (A_u). If this anchor is sufficiently similar to an existing user anchor in the model, it is merged with it, and the α factor is updated. Otherwise, it becomes a new user anchor, with the associated α factor set to a very small initial value. It indicates that the newly create user anchor is not as reliable as system anchors at the beginning.

Temporary user anchor

Since a user's suggested position could be arbitrary, saving these suggestions separately would bloat the model significantly. Therefore, we use discrete locations by dividing the study area into an $m \times n$ grid. That is, any position within a grid cell is represented by the centre of the cell. This grid-based selection of the position is enabled directly in the user interface provided to the user (Figure 2).

Note that the resolution of this grid could be different from the resolution as used in the training phase. We can set smaller grid space because the system training from users is cost-effective. This helps to efficiently reduce the grid space between system anchors. Thus, the resolution of entire system could be refined.

Within each grid cell, its geometric centre is used to represent the positions of all temporary user anchor points falling into it.

We thus define the user anchor A_u as:

$$A_u = (\mathcal{P}_u, F_u),$$

where \mathcal{P}_u is the grid cell centre that contains the user suggested position and F_u is the user fingerprint summarized from the current Wi-Fi RSS measurement.

Anchor merge

A newly generated positive feedback could be either converted to a new user anchor point or merged with an existing user anchor point based on their similarity. As mentioned

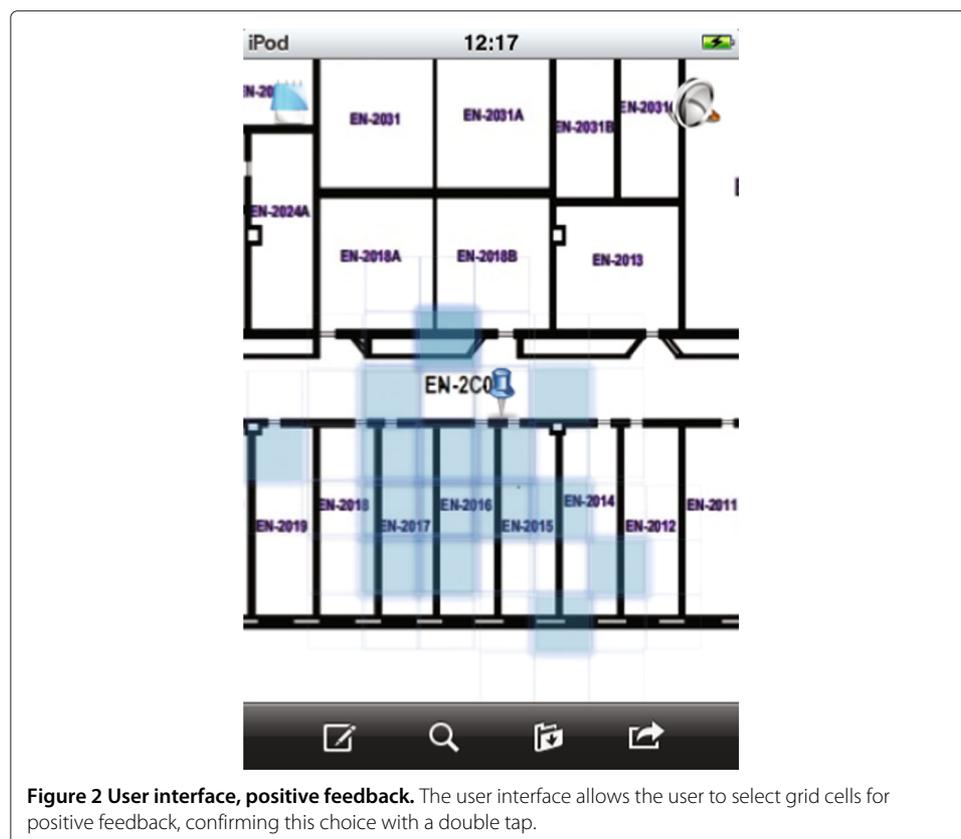


Figure 2 User interface, positive feedback. The user interface allows the user to select grid cells for positive feedback, confirming this choice with a double tap.

before, we believe that positive feedback represented by a user anchor point should gradually become reliable if more and more similar user anchor points are generated to confirm it. Before we discuss how to update the reliability of user anchors, we define the similarity between two user anchor points.

Given user anchor points A_{u_i} and A_{u_j} , $i \neq j$, their similarity is determined by two aspects:

- *Wi-Fi RSS fingerprint similarity*: A natural measurement mechanism is the cosine similarity in the range of $[0, 1]$. Thus, the Wi-Fi RSS fingerprint similarity F_u is given as:

$$s_{F_u} = \begin{cases} 1 & \text{if } \cos(F_{u_i}, F_{u_j}) > a \\ 0 & \text{otherwise} \end{cases},$$

where F_{u_i} and F_{u_j} are Wi-Fi RSS fingerprints of user anchor points A_{u_i} and A_{u_j} respectively. They are all sparse vectors of n dimensions; a is the threshold for Wi-Fi RSS fingerprint similarity.

- *Physical position similarity*: If two user anchor points share the same geometric centre of a grid as their position. They are considered as similar in position.

As a result, we claim that two user anchor points are similar if they satisfy both of the two similarity conditions above.

A temporary user anchor A_{u_i} is thus merged with the existing user anchor A_{u_j} in the same cell if their fingerprints are sufficiently similar. If multiple anchors already exist in the same cell as A_{u_i} , we only consider the most similar one, denoted A_{u_j} . If the similarity between A_{u_i} and A_{u_j} is greater than a threshold, the temporary user anchor is regarded as the same as the existing one, and therefore is merged with it.

The α factor

Whenever a temporary user anchor is merged with an existing user anchor in the system, the associated α factor is updated. For user anchor A_{u_i} , we define α_i as

$$\alpha_i = \frac{1}{a + e^{-x}}, \text{ with } x \geq 0 \text{ and } 0 < a \leq 1,$$

where the variable x has a cumulative effect and a is a parameter controlling the initial and maximum values of α_i . When a user anchor A_{u_i} is first created, its original likelihood will be reduced by a small α . As more positive feedback is provided in support of it, its α factor gradually increases until it reaches an upper limit.

Thus, the magnification capability of the α factor is $\frac{a+1}{a}$. The increment of x is defined as

$$\Delta x = \frac{\frac{T}{T_s} + e^{-\sigma_F}}{b} \text{ with } b > 0.$$

The pace of the increase of x is controlled by a few aspects:

- An independent parameter b , which compensates the increasing velocity of x . When there are many users (e.g., in a large shopping centre), we may not want to trust their individual estimation much. Instead, we can rely on the convergence effects of large amount of users to evolve the mode. However, when there are only a few users (e.g., in a depot), we assign each individual feedback a much higher weight.

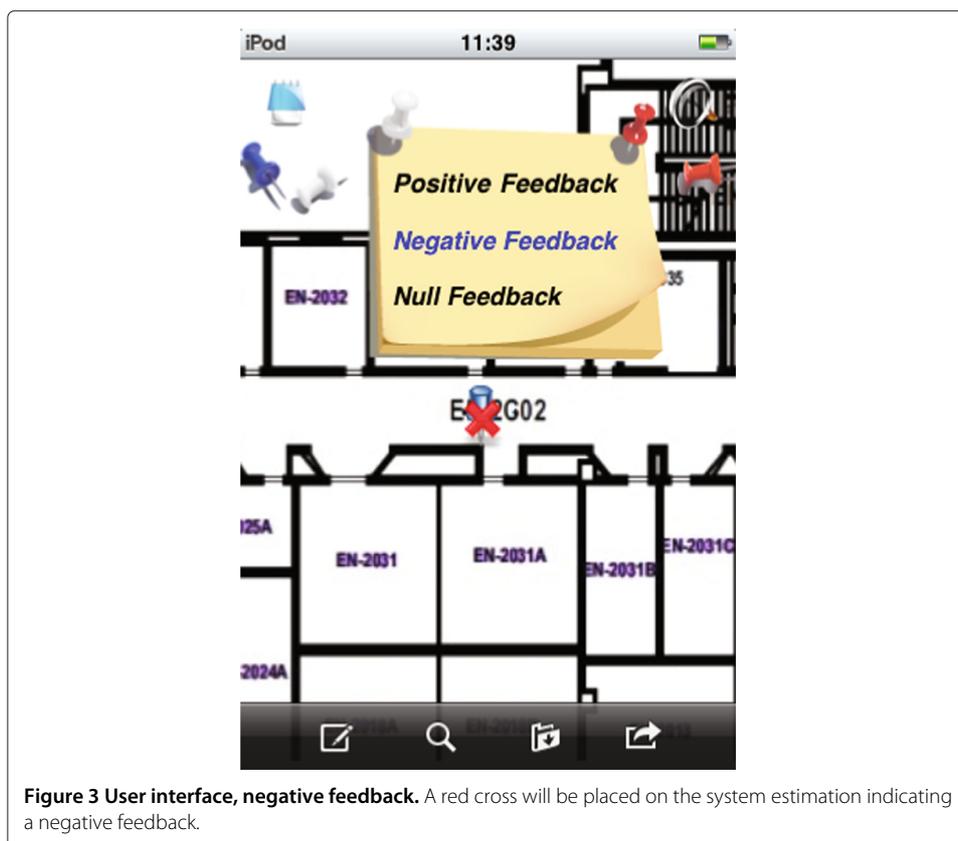
- The variance of the current RSS fingerprint, σ_F . The user feedback generated in the environment with small RSS variance will have a greater influence on the evolution speed of the model.
- If T is the number of Wi-Fi scans used in the positioning query and T_s is the number of Wi-Fi scans used during system training, their ratio $\frac{T}{T_s}$ also reflects the credibility of this positive feedback.

As a result, the α factor increases fastest with the first few instances of the user anchor, becoming stable once a sufficient number of feedback events are received. The rationale for this design is to allow the system to quickly adapt to new information provided by the users, but without this feedback overpowering the system.

Negative feedback

Suppose the system delivers a position from the top- k anchors according to their likelihood ranking, but the user believes this location to be incorrect and cannot provide any further information regarding the actual location (Figure 3). The negative user feedback to this estimated position can also provide valuable information to the system.

Typically, when a user rejects the position estimated by the system, the reason could be that the user is nowhere near any of the anchors known by the system. In this case, none of the top- k anchors would truly represent a good estimate. Therefore, we should try to decrease their likelihoods simultaneously.



Given an anchor A_i , we use a negative user feedback factor β_i to reduce its likelihood according to the accumulation of negative feedback received. Similar to the positive feedback model, the negative factor model also has fast adaptability. Accordingly, we define β_i as

$$\beta_i = e^{-x}.$$

When an anchor is given a negative feedback, we give x in above formula the same increment Δx used in the positive user feedback.

The value of β is inversely related to x , such that β will decrease from the initial value 1 to its limit zero as x increase from zero to infinity. As a result, if more and more users reject the same set of anchors, they will not be chosen as the top- k due to the small value of the β factor.

Evaluation

Experimental settings

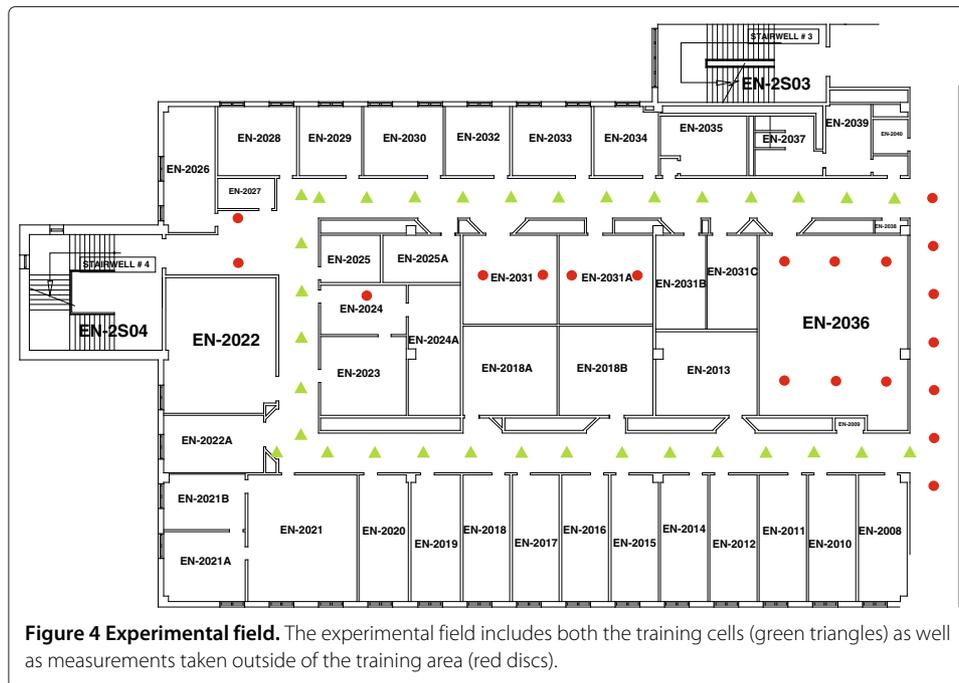
The system evaluation consisted of two phases. The first phase was to analyze the performance of the baseline system without user feedback in field tests. The accuracy and precision of the baseline system was calculated. By analyzing these two performance metrics, we can determine whether or not our baseline system is suitable for comparison purposes. During the second phase, we explored how the proposed user feedback model improved the system performance.

Experiments and evaluations with this feedback model were conducted in an complex indoor office environment, which is the part of the 2nd floor of the Engineering Building at Memorial University. The reason we chose this experimental field is that we can fully control our evaluation process under this environmental setting. The space was divided into a grid using a 3×3 m cell size. 33 positions were selected within the hallways for training the baseline system (denoted the *training area*), and an additional 20 positions were selected as untrained positions for testing purposes (denoted the *non-training area*). A diagram of the setting is provided in Figure 4. System anchors were created in the training area only. Note that the non-training area lacks valid system or user anchors. It can be treated as an area that is the result of environment alteration, a new Wi-Fi coverage area, or a region that was neglected in the training of the system.

The prototype system was developed for iPhone OS 3.1.2; experiments were conducted using the Apple iPhone and iPod Touch devices.

The system training was conducted during semester break (April, 2010). Each RSS fingerprint had been generated by extracting features from 20 Wi-Fi scans, which took approximately two minutes. The baseline system evaluation was conducted during the summer semester (May - July, 2010) with much more interference from other people and their electronic devices. Thus, the RSS data provided by users are more capable of describing the Wi-Fi characteristics of the current environment.

As mentioned earlier in Section 'Human-Centric collaborative feedback model', the parameters in the feedback model are used to adjust the rate of change of the α and β factors (i.e., the sensitivity of our user feedback model). In production environments, the sensitivity of the user feedback model will depend on the number of users and the degree of trust of those users. For the purpose of evaluation, we increased the sensitivity of the



user feedback model in order to speed up the rate at which the system is able to learn from user feedback.

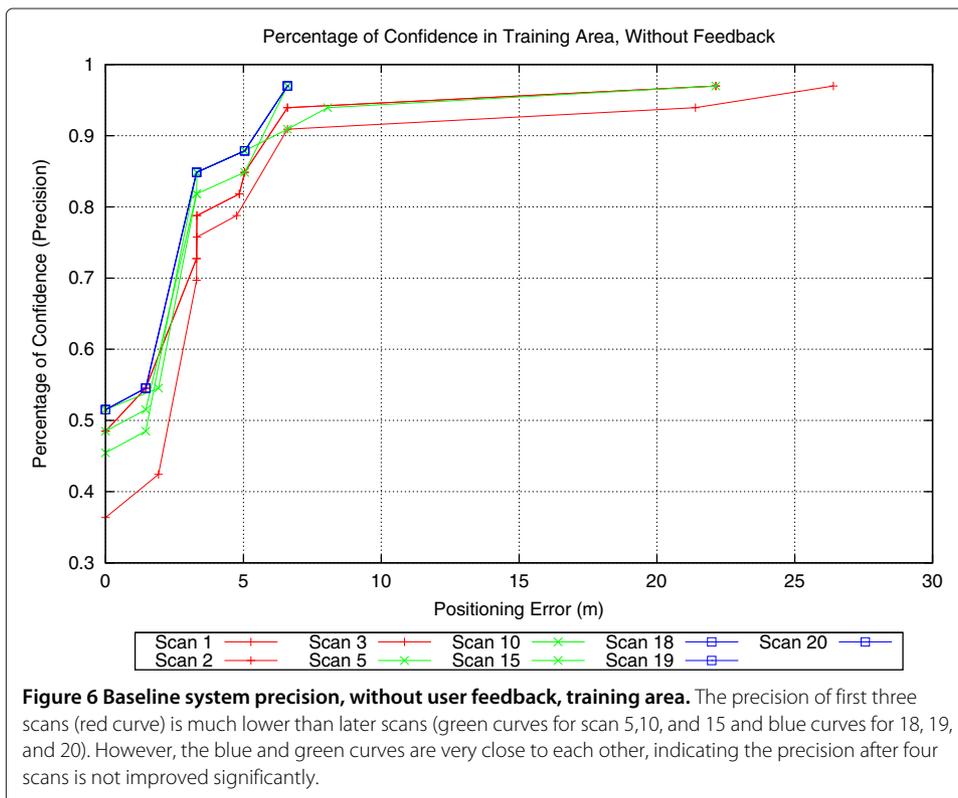
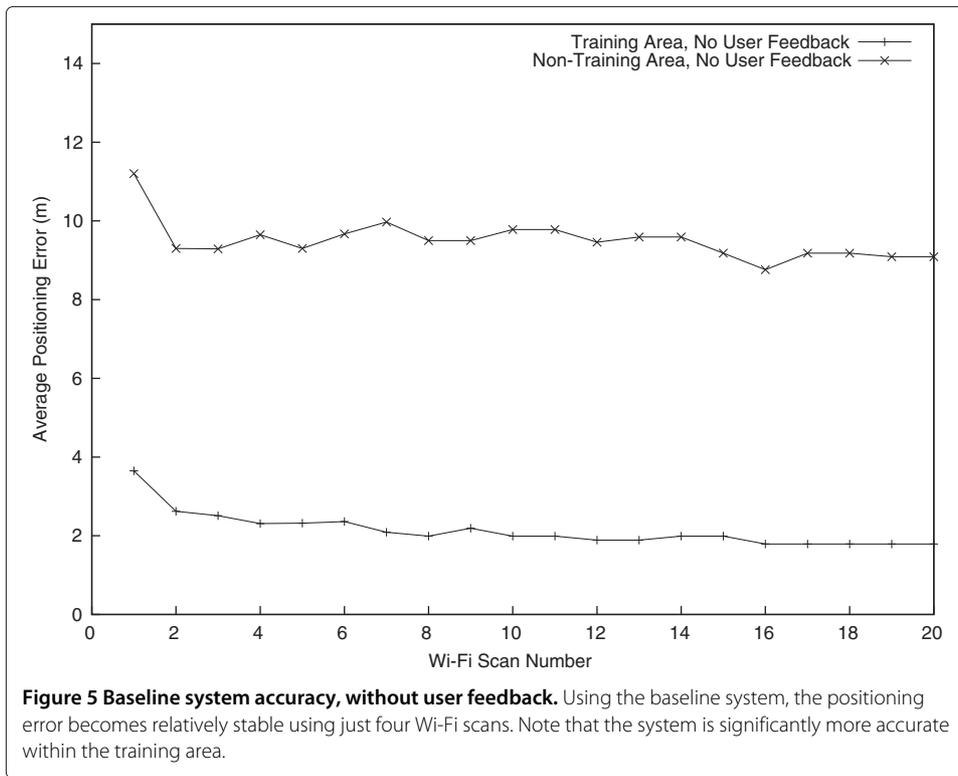
We set the value of parameter α to be 1, which means that the magnification factor of parameter α is 2. The value of parameter β was set to be 0.6. As such, according to the design of our user feedback, these parameter setting will weight the first four users much larger than subsequent users, which grants the system a fast learning ability.

Baseline system evaluation

Since the time that a user is willing to spend waiting for a positioning result influences the service quality, we have conducted an experiment to investigate the relationship between time (i.e., the number of Wi-Fi scans) and system performance. We use the baseline system to determine the smallest number of Wi-Fi scans (measured at one scan per second) needed for the system to produce a reasonably accurate result. At the same time, the performance of our baseline system can be evaluated with respect to other similar systems described in the literature.

In the training area, for each survey point, we have collected 20 scans of the Wi-Fi RSS, using these incrementally to query the positioning system. The average positioning error after each scan is plotted as the bottom curve in Figure 5. We can observe that for a small number of scans, the system has an error between 2 and 4m. As more scanned RSS data are used (i.e., greater than four), the accuracy stabilizes at around 2m.

The system precision, as another very important metric for system performance, is plotted in Figure 6. We selected the positioning precision for 9 out of the 20 scans, illustrating three phases of Wi-Fi sampling. The early phase consists scans 1, 2, and 3 (red curves). In this phase, due to insufficient Wi-Fi RSS data, the precision is low. The second phase includes scans 5, 10, and 15 (green curves), it is in the middle of the Wi-Fi sampling and has more Wi-Fi RSS data than the first phase. The last phase is at the end



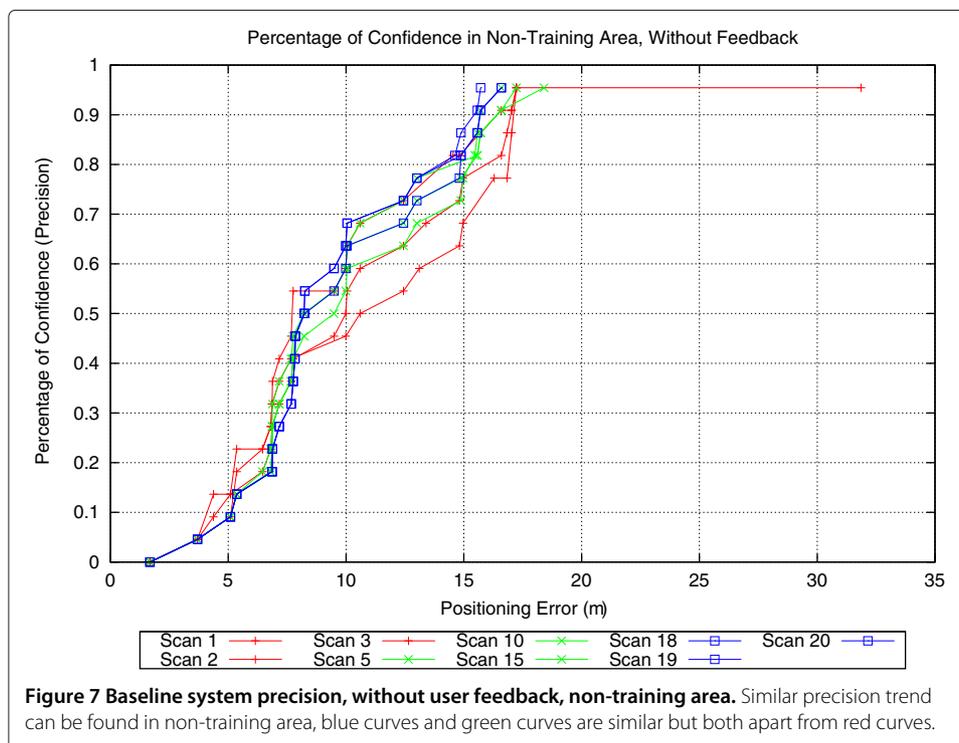
of Wi-Fi sampling (scans 18, 19, and 20), which includes all RSS vectors (blue curves). From Figure 6, we can see that the green and blue curves are very close to each other, which means that a scan number greater than four will not generate significant precision improvement. However, if the Wi-Fi scan number is small (i.e., less than four), the probability of generating outliers is considerably high.

Similarly, in the non-training area, we also collected 20 scans for each position. We plotted the positioning accuracy for the number of scans as the top curve in Figure 5 and positioning precision in Figure 7. In this case, the system performance is significantly lower than in the training area due to the lack of system anchors. However, in both the training area and non-training area, four scans provide a reasonable trade-off between performance and positioning time. Therefore, we use this as the number of scans in the remainder of our experiments.

According to the analysis of our baseline system, the average positioning error is between 2m and 4m, respectively, depending on the Wi-Fi sampling time. It is in fact only marginally worse than the 0.7m to 4m average positioning error yielded by the best-performing but intensively trained Horus system (using 100 Wi-Fi scans and much smaller grid space of 1.52 m and 2.13 m) [16]. Thus, we believe this baseline system is qualified to evaluate the value of the proposed human-centric collaborative feedback model.

Collaborative feedback model evaluation

In order to evaluate the benefits of the collaborative feedback model, we have defined a number of different scenarios that represent specific types of behaviours of users. While we do not claim that any of these evaluations represents what would occur in real world

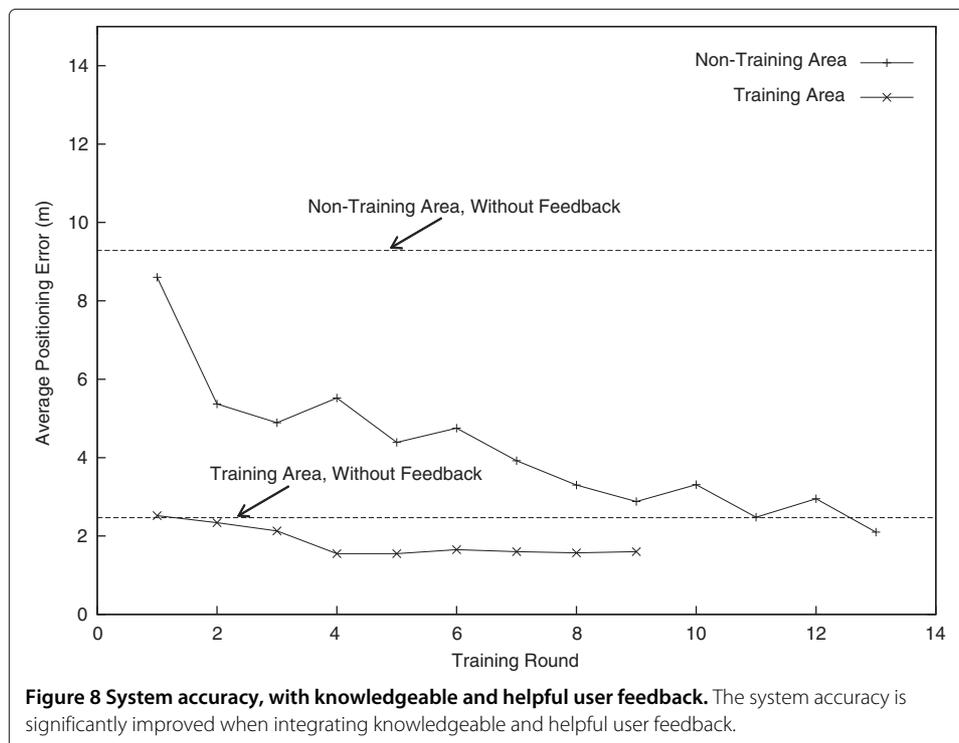


use, they allow us to examine how the system will react to different types of feedback. Our future plans for real-world field trials are discussed in Section ‘Conclusions and future work.’

Knowledgeable and helpful feedback

Next, we investigate how the user feedback model improves the system performance. In this scenario, whenever the system returns a position that does not match the true position of the user, feedback was provided. We modelled the user as being knowledgeable and helpful; whenever the position was inaccurate, the user suggested positive feedback 80% of the time, and negative feedback 20% of the time. We believe it is a reasonable choice for situations where users are highly motivated to provide accurate and positive feedback. In fact, there may be many other users who are providing null feedback (i.e., using the system and trusting the results). However, since such types of users do not affect the evolution of the model, they are not discussed here.

Within the training area, we define a *round* as a traversal of all grid cells. In a round, the user stops at each survey position to scan the RSS for nearby APs (using four scans). If the result is correct, the user moves to the next position. Otherwise, the user provides feedback before moving on. The average positioning accuracy after nine such rounds of visiting and testing each position is plotted in Figure 8. In the course of providing this user feedback, the positioning error within the training area improved from approximately 2.5m to 1.5m after just four rounds. From there, little change was observed. Note that the baseline system accuracy ranged from 4m to 2m without feedback. At this point, with the integration of human-centric collaborative feedback, the system performance is further improved even in the well trained area.

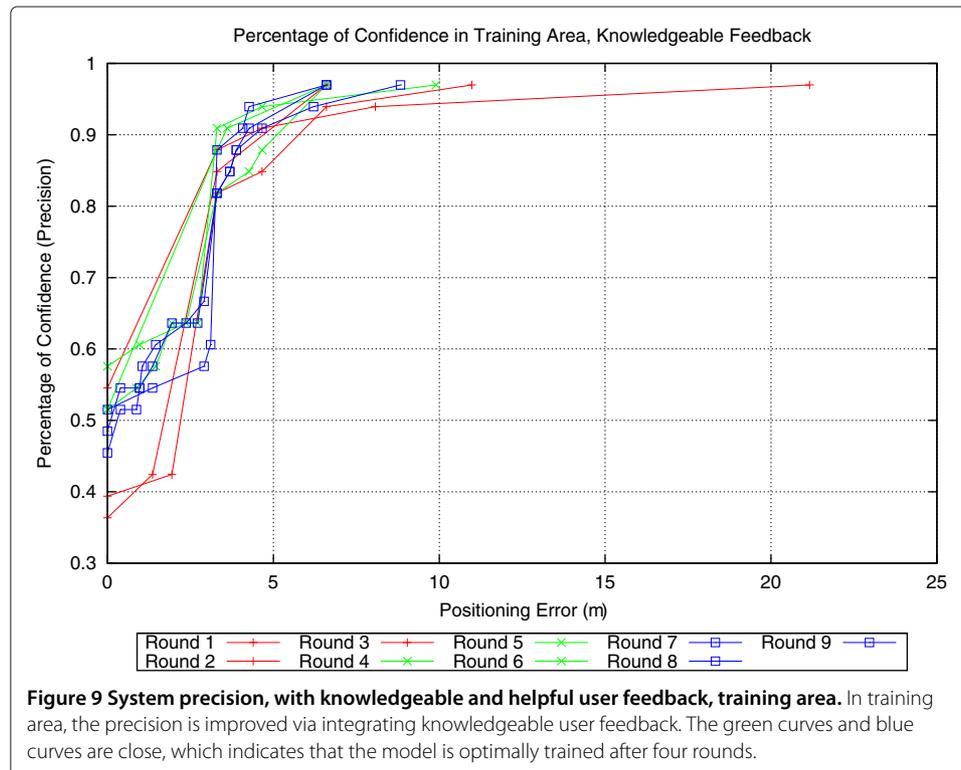


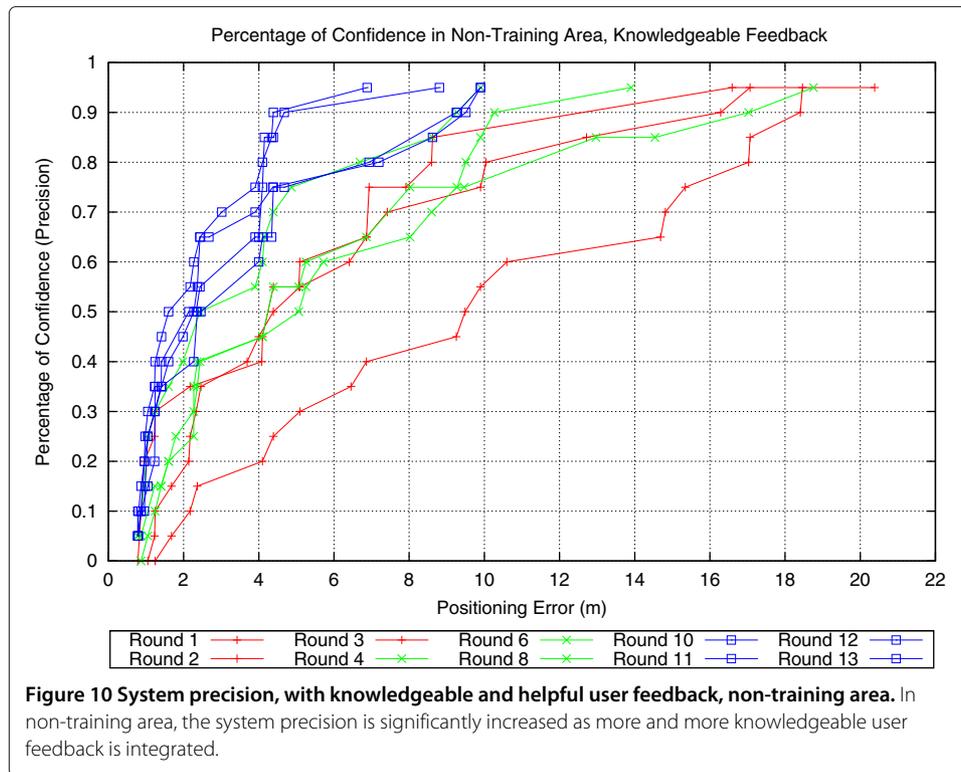
The precision is also improved after four rounds of user-involved positioning within the training area, as we can see in the green and blue curves which are closer to the y axis than red curves shown in Figure 9. Furthermore, green and blue curves are close to each other, which indicates that the model reaches its optimal performance after approximately four rounds of knowledgeable and helpful feedback.

Within the non-training area, the experiment followed the same procedure as in the training area, producing the data plotted in Figure 8. Because there was no training data in these regions, the initial positioning error was rather large. However, after 13 rounds of collecting user feedback, the error decreased from 9m to 2m. The precision is also significantly increased as plotted in Figure 10. As a result, the system performance in an area that had not been previously trained became comparable to the training area.

The reliable user feedback contains information (user fingerprint) that best characterizes the current Wi-Fi RSS features. Such helpful information can help the system to improve the performance. At the beginning of the test within the non-training area, the model contained only system anchors, and therefore could only return the position of a system anchor (i.e., within the training area) to the user. These positions were often far from the true position of the user. As a result of the positive feedback, user anchors were added and the relative weight of these anchors were enhanced by the α factor. Similarly, with the negative feedback, the weight of the system anchors were reduced by the β factor. As a result, the positioning accuracy increased as more user anchors become valid candidate positions.

What this means for indoor positioning systems is that the system training and maintenance costs can be reduced significantly by relying on knowledgeable and helpful end users





working on a partially trained system, eventually achieving the same level of accuracy as a fully trained system. Also, the resolution of the positioning system is improved because many reliable user anchors fill the gap between system anchors, thus reducing the grid space or increasing the grid resolution.

At this point, the optimal combination of different types of user feedback is not considered. To conduct experiments testing each possible combination is impractical within a limited time period. In fact, this problem can be explored if we could use a simulation testbed. We can collect a large amount of real Wi-Fi RSS data to simulate the Wi-Fi scans. When the simulated positioning process is finished, virtual user positive or negative feedback can be generated to the evolve the model. As such, the system performance with an arbitrary combination of positive and negative feedback could be estimated.

Mixed feedback

In a real environment, user feedback can be either helpful or malicious. At this point, we assume that the accuracy of user feedback follows the normal distribution. Thus, the feedback from malicious users should exist as outliers. We could employ some supervised classification algorithms such as logistic regression or SVM to classify the malicious users. However, the Wi-Fi RSS fingerprinting based positioning is essentially an unsupervised or instances-based approach (similar to KNN). For instance-based learning, we can cluster different user feedback based on their RSS features and locations, which avoids labeling whether the user is benign or malicious. As described in the previous section, we take the grid-based clustering approach with predefined centres. The reliability of each cluster is compensated by our user feedback model. Furthermore, the performance of instance-based approaches is in fact highly dependent on whether we will have a large

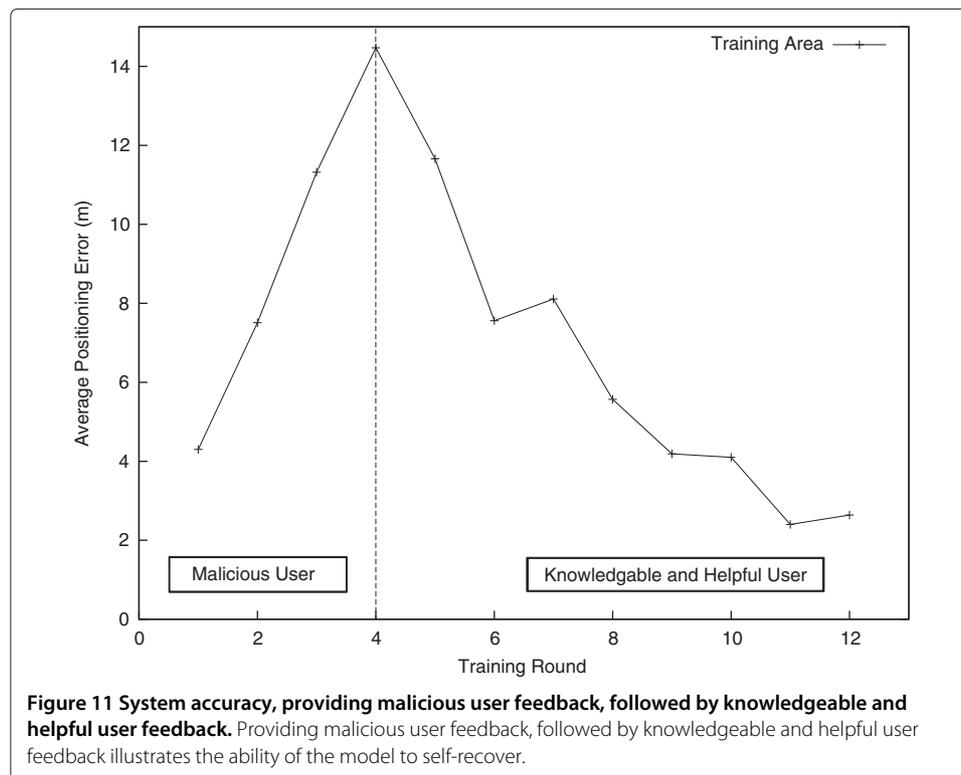
dataset or the noise level in training dataset. Thus, if the noise level is very high (e.g., all user feedback are from malicious users), the performance of the system will not be acceptable.

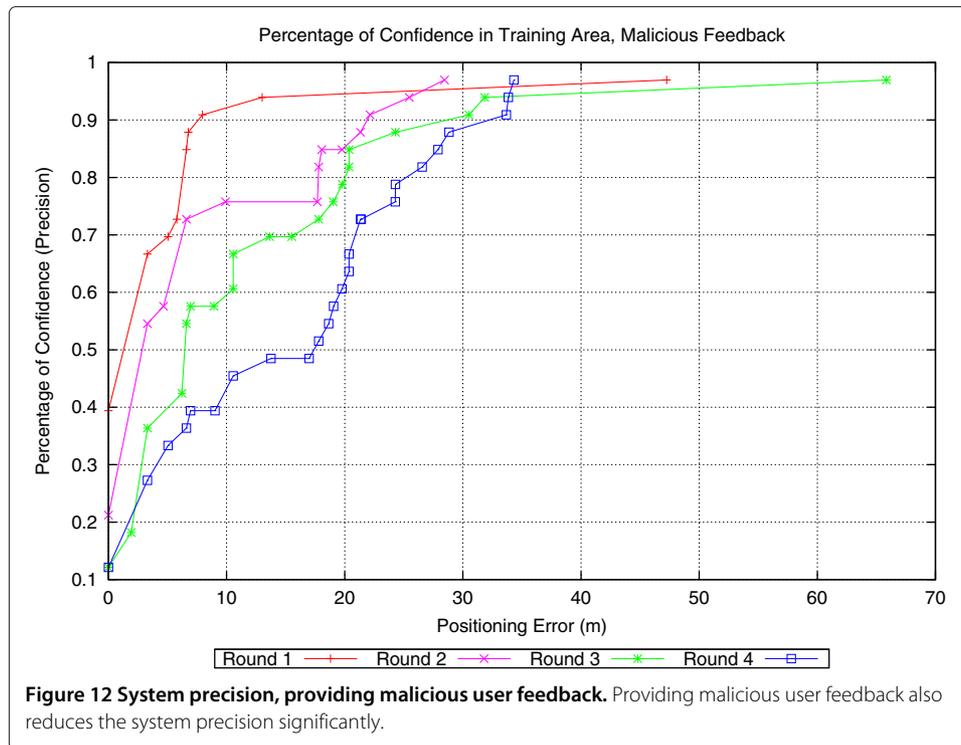
In this experiment, we test the model to determine its ability to recover from incorrect feedback. In particular, we model the user feedback as completely malicious at the beginning and as completely informative thereafter. Such a behaviour is not typical but it provides a “worst case scenario” study of the system, followed by its ability to recover from incorrect or malicious feedback.

Our focus here is on the training area only. As seen in the previous experiments, the non-training area can become nearly as good as the training area with sufficient user feedback. As such, we expect similar results within the non-training area as the training area with respect to mixed feedback.

During the initial phase of this experiment, whenever the system returns a correct position estimation, the malicious user has a 50% chance of either providing negative feedback of suggesting a random false position. When the system is incorrect, the malicious user provides null feedback. Following a similar methodology as the previous experiments, such malicious feedback was provided for four rounds. Another eight rounds of feedback from a knowledgeable and helpful user was then collected.

The position errors for this experiment are plotted in Figure 11. We observe that the system error starts out with around 4m and quickly increases to 14m as a result of the malicious feedback. At the same time, the system precision is also reduced to an unacceptable level, shown as the red curves in Figure 12. With an error of 14m and extremely low precision, the system is considered to be fairly disturbed by the malicious users. At





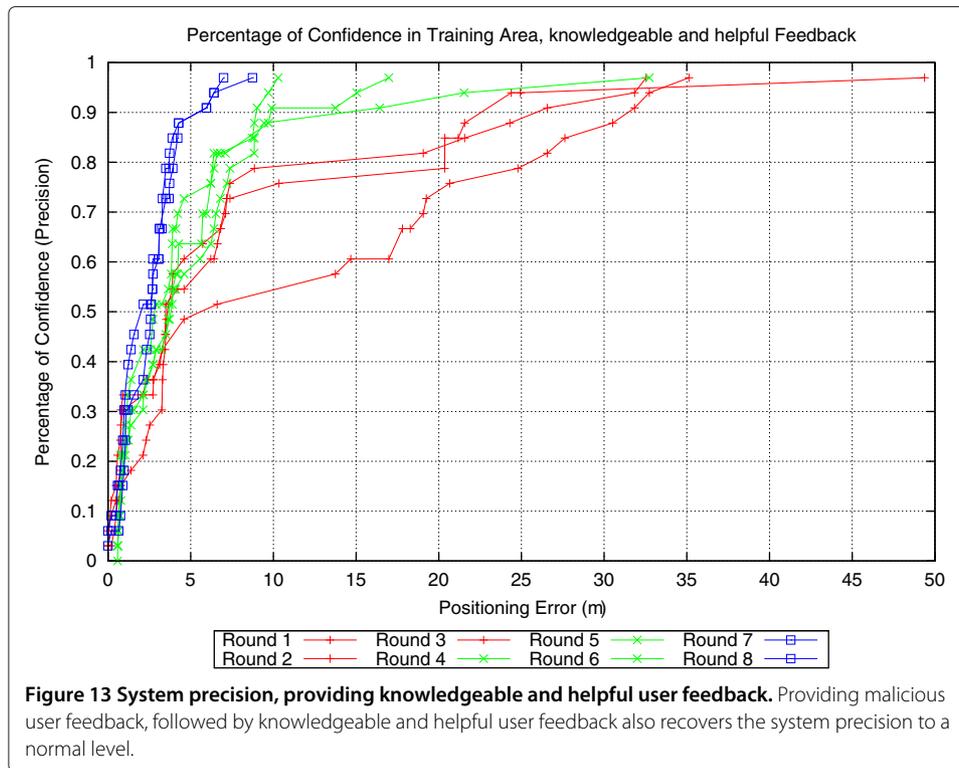
this point, we turn the user into knowledgeable and helpful to provide positive feedback whenever the system is incorrect.

The user behaviour in this case is the same as in the previous subsection. The helpful feedback quickly corrects the significant positioning errors, recovering to the starting accuracy after five rounds of feedback, and below 3m after eight rounds. At the same time, the system precision is stabilized as indicated by the blue curves in Figure 13. As a result, our system has recovered from the low accurate state by integrating helpful and knowledgeable feedback.

In real life, helpful and malicious feedback are often mixed together to feed the model. As such, the phenomena described in this experiment might be rarely observed. However, it in fact provides the “worst-case”. If the model can eliminate the negative effect introduced by continuous malicious or unreliable user feedback, then it is reasonable to deduce that it is robust to malicious user feedback in more moderate or general cases.

Conclusions and future work

In this article, the primary contribution is the presentation and evaluation of a user feedback model which receives and processes human-centric collaborative feedback. The proposed user feedback model adjusts the positioning results via placing a compensation mask over the likelihood vector (distribution) generated in the positioning phase. The history of both positive feedback and negative feedback will affect the compensation ability of such a mask. In general, positive feedback generates user anchors and enhances their reliability. On the other hand, negative feedback reduces the trustworthiness of an anchor. All user feedback will be assigned low compensation power when first created and can be enhanced with multiple similar feedback events. As such, this user feedback model



should be able to gradually update the system's knowledge and guide the system to learn the changes in the Wi-Fi indoor environments. Based on these principles, we have built a prototype and conducted experiments to evaluate it. Experimental results show the ability of the model to improve upon the positioning accuracy and precision in both regions that have been trained, as well as in nearby regions that do not include sufficient anchors. The model is also shown to be robust with respect to malicious feedback, quickly recovering based on helpful user feedback.

In general, storing arbitrary user feedback could require very large storage space and the computational cost of typical clustering algorithms (such as k -means) is high. However, the anchor merge mechanism proposed in our user feedback model merges all similar user anchors which avoids the need to store every user anchor. Furthermore, the grid-based clustering in the user feedback model only needs to cluster each user anchor within the same grid-cell, which significantly reduces the calculation time. As a result, even with the addition of the user feedback mechanisms to the positioning system, the resulting approach remains efficient.

Such a feedback model can be further refined and enhanced in a number of interesting ways. The first refinement is to use the temporal aspect of user feedback, such that different times (morning, noon, and night) of a day or different dates (weekdays, weekends, and holidays) is used to generate different RSS data patterns. For example, in a university cafeteria, due to the interference from human bodies and electronic devices, the RSS measurement generated during lunch time could be very different from that in the early morning. As such, the user feedback generated during lunch time may mislead the positioning activities at other times of the day. In order to solve this problem, the model should take advantage of the timestamp within the RSS fingerprint, limiting the candidate

anchors to those that were created at about the same time of the day. This could increase the accuracy of the system in environments with time-related changes in human activities. More complex approaches could be developed that dynamically learn the features of when the RSS measurements are changing, using this to partition the data to generate different models for different times of the day. Also, we can introduce a forgetting mechanism which could remove user feedback from the system. It could be used to address situations where malicious feedback has been received but subsequent helpful feedback is not available.

The second aspect for refinement of this approach is to perform cross platform validation. In real indoor environments, users could carry different types of mobile devices. Due to the diversity of manufacture technologies in wireless network interface cards, the RSS generated by different Wi-Fi chips could also be different. However, our entire implementation and experiments are conducted on Apple iPhone and iPod Touch, which indicates its limitation in field validation. At this point, we argue that the system performance could be improved if the diversity of Wi-Fi chips in different mobile devices is considered. The simplest but efficient approach is to create individual fingerprints database for each type of mobile device. It might improve system performance with a large system overhead. More intelligently, an RSS compensation mechanism can be integrated to automatically adjust RSS patterns among different mobile devices.

The evaluations of the proposed approach have allowed us to validate the ability of the system to learn useful information from the collaborative feedback provided by the users. However, the specific scenarios were somewhat contrived and do not represent realistic user behaviour. As such, field trials within a real-world positioning domain (e.g., new students using the system to find their way around a university campus) are currently in the planning phases.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors contributed equally to this work. YL designed and implemented the proposed prototype, and prepared the manuscript; OH analyzed and described the theoretic human-centric feedback model, supervised its analysis and edited the manuscript; YC defined the problem domain and refined the system, designed and developed the system evaluation process, supervised its analysis and edited the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This work was funded by scholarships provided by the School of Graduate Studies at Memorial University to the first author, as well as NSERC Discovery Grants held by the second and third authors.

Author details

¹Department of Computer Science, The University of Western Ontario, London, ON, Canada. ²Department of Computer Science, Memorial University of Newfoundland, St. John's, NL, Canada. ³Department of Computer Science, The University of Regina, Regina, SK, Canada.

Received: 8 August 2012 Accepted: 30 January 2013

Published: 6 March 2013

References

1. Ladd AM, Bekris KE, Rudys AP, Wallach DS, Kavradi LE (2006) On the feasibility of using wireless ethernet for indoor localization. *IEEE Trans Wireless Commun* 5(8): 555–559
2. Harter A, Hopper A, Steggles P, Ward A, Webster P (2002) The anatomy of a context-aware application. *Wirel Netw* 8(2): 187–197
3. Bahl P, Padmanabhan VN (2000) RADAR: an in-building RF-based user location and tracking system. In: *Proceedings of 19th IEEE international conference on computer communications*. pp 775–784
4. Kaemarungsi K, Krishnamurthy P (2004) Properties of indoor received signal strength for WLAN location fingerprinting. In: *Proceedings of international conference on mobile and ubiquitous systems: networking and services*. pp 14–23

5. Kaemarungsi K, Krishnamurthy P (2004) Modeling of indoor positioning systems based on location fingerprinting. In: Proceedings of 23rd IEEE international conference on computer communications. pp 1012–1022
6. Bhasker ES, Brown SW, Griswold WG (2004) Employing user feedback for fast, accurate, low-maintenance geolocationing. In: Proceedings of the Second IEEE annual conference on pervasive computing and communications. pp 111–120
7. Bolliger P (2008) Redpin-adaptive, zero-configuration indoor localization through user collaboration. In: Proceedings of the first ACM international workshop on mobile entity localization and tracking in GPS-less environments. pp 55–60
8. Gallagher T, Li B, Dempster AG, Rizo C (2010) Database updating through user feedback in fingerprint-based Wi-Fi location systems. In: Proceedings of ubiquitous positioning indoor navigation and location based service. pp 1–8
9. Park J, Charrow B, Curtis D, Battat J, Minkov E, Hicks J, Teller S, Ledlie J (2010) Growing an organic indoor location system. In: Proceedings of the 8th international conference on mobile systems, applications, and services. pp 271–284
10. Hossain AM, Van HN, Soh WS (2010) Utilization of user feedback in indoor positioning system. *Pervasive and Mob Compu* 6(4): 467–481
11. Kushki A, Plataniotis KN, Venetsanopoulos AN (2007) Kernel-based positioning in wireless local area networks. *IEEE Trans Mob Compu* 6(6): 689–705
12. Schölkopf B, Smola AJ (2002) *Learning with kernels*. MIT Press, Cambridge
13. Haeberlen A, Flannery E, Ladd AM, Rudys A, Wallach DS, Kavraki LE (2004) Practical robust localization over large-scale 802.11 wireless networks. In: Proceedings of the 10th annual international conference on mobile computing and networking. pp 70–84
14. Vossiek M, Wiebking L, Gulden P, Wiehardt J, Hoffmann C, Heide P (2003) Wireless local positioning. *IEEE Microw Mag* 4(4): 77–86
15. Kariv O, Hakimi SL (1979) An algorithmic approach to network location problems. I: The p -centers. *SIAM J Appl Math* 37(3): 513–538
16. Youssef M, Agrawala A (2005) The horus WLAN location determination system. In: Proceedings of the 3rd international conference on mobile systems, applications, and services. pp 205–218

doi:10.1186/2192-1962-3-2

Cite this article as: Luo et al.: **Enhancing Wi-Fi fingerprinting for indoor positioning using human-centric collaborative feedback.** *Human-centric Computing and Information Sciences* 2013 **3**:2.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
