Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling

Philipp Bolliger[1], Kurt Partridge[2], Maurice Chu[2], Marc Langheinrich[3]

[1] Institute for Pervasive Computing, ETH Zurich, Switzerland
[2] Palo Alto Research Center, Palo Alto, CA, USA
[3] Faculty of Informatics, University of Lugano, Switzerland

Abstract. Wireless signal strength fingerprinting has become an increasingly popular technique for realizing indoor localization systems using existing WiFi infrastructures. However, these systems typically require a time-consuming and costly training phase to build the radio map. Moreover, since radio signals change and fluctuate over time, map maintenance requires continuous re-calibration. We introduce a new concept called "asynchronous interval labeling" that addresses these problems in the context of user-generated place labels. By using an accelerometer to detect whether a device is moving or stationary, the system can continuously and unobtrusively learn from all radio measurements during a stationary period, thus greatly increasing the number of available samples. Movement information also allows the system to improve the user experience by deferring labeling to a later, more suitable moment. Initial experiments with our system show considerable increases in data collected and improvements to inferred location likelihood, with negligible overhead reported by users.

1 Introduction

WiFi localization has shown great promise for indoor positioning, yet has not achieved ubiquitous commercial success yet. One difficulty has been the construction of an accurate mapping between signal strength patterns and physical locations. The signal strength patterns depend not only on the distances between WiFi radios, but also on other factors such as the positions of physical objects, which reflect or block signals. This complication may be partially overcome by either performing calculations with detailed models of the environment, or by collecting a dense dataset of fingerprints and their associated true locations [2]. In this paper, we focus on the latter approach, as it is generally more accurate and it is easier to collect this data.

Even so, collecting labeled fingerprint samples can be tedious. Signal readings must be collected every few meters or so, with pauses of tens of seconds at each position to get an accurate reading. This process must be repeated if the infrastructure or environment changes substantially. Commercial deployments usually conduct such surveys as part of deployment, however in some installations, such as private homes, consumers may not have the patience for this process.

Previous work has explored end-user labeling of locations [1, 4, 8, 5]. End-user labeling allows labels to be added as needed, in the places that users most frequently visit. However, second-by-second signal fluctuations mean that the fingerprint stored with a label may not match future measurements. Ideally, an end-user labeled fingerprint would also be collected over an interval of several tens of seconds, much as is done during formal calibration stages. Users, however, may not have the patience to comply with this restriction.

In this paper, we present PILS, an adaPtive Indoor Localization System that addresses the challenges of end-user labeling. PILS explores a technique that extends the applicability of a user-provided label from an instant to an interval over which the device is stationary. The stationary state is detected using an accelerometer, which allows PILS to detect location changes autonomously, and consequently collect stationary interval measurements without explicit user intervention.

Using intervals also enables a different kind of labeling. By detecting intervals of device immobility, the system can defer location labeling to a more appropriate time, and refer to longer time periods that are easy for users to remember (e.g., "Where were you between 9:15 am and 10:05 am today?"). This greatly improves the user experience, as users need not provide labels while at the labeled location, where they are likely engaged in some other activity. We call this technique *asynchronous interval labeling.*

The reminder of this paper is structured as follows. The next section explains how PILS relates to other indoor localization systems, in particular, those that follow a user-labeling approach. Section 3 then describes asynchronous interval labeling in detail. Sections 4 and 5, respectively, describe the prototype implementation of PILS and the results of the initial experiments. We close with discussion and conclusions in Sections 6 and 7.

2 Related Work

Research on indoor location systems has been popular for several years [2, 10, 17, 25]. Location systems typically output Cartesian coordinates, which, for indoor settings, are often mapped to rooms based on available map data [17]. Like other systems [6, 9], we output symbolic location data (such as room identifiers) directly. However, rather than using authoritative designations, we collect room labels directly from end-users during their use of the system.

WiFi fingerprinting [2] has been particularly popular for indoor positioning, because it requires no new hardware infrastructure for sites that already have WiFi. With resolution to a few meters, it can usually support room-level localization. To achieve such high accuracy from the noisy WiFi signal, however, such systems require prior manual calibration. For example, King et al. [17] were able to achieve an average error distance of less than 1.65 meters, but required prior calibration with 80 measurements every square meter (20 measurements each at 0°, 90°, 180°, and 270° orientations). Even though a single active WiFi scan takes only 250 ms, the time needed to measure all four orientations and to move between locations quickly adds up to tens of seconds per reference point. In total, the training phase for a small 100 m^2 building could take well over one hour. In addition, the training may miss longer-term variations as described in Section 3. While training time can be reduced by modeling the environment [12], this approach is less accurate and requires additional information (such as floorplans) that are not always available or easy to input.

Ashbrook and Starner [1] describe how significant places may be determined by detecting stationary GPS signals, which can later be clustered and labeled. Froehlich et al. [8] also identify significant places using GSM signals. Both approaches identify places on a building-sized scale rather than a room-sized scale, and neither use an additional sensor such as an accelerometer to detect true motion stability.

Bhasker et al. previously explored collecting calibration data during use, rather than in a separate training step [4]. Their localization system employs a two stage process. First it computes geometric location. The result is shown on a map, and can be corrected if necessary. Corrections are treated as virtual access points and given higher priority when calculating locations. However, this method requires having a map and interrupting the user's primary activity to collect input. The system also allows only one correction per location.

Our earlier work, Redpin [5], also collected calibration information from end-users. In contrast to Bhasker et al.'s geometric approach, Redpin generates symbolic location identifiers with room-level accuracy. If the current location cannot be predicted accurately, the user is prompted to enter a label for his or her current location. By allowing multiple measurements for the same location and by combining from GSM, WiFi, and Bluetooth, Redpin can provide the correct symbolic location in nine out of ten cases, as evaluated in a two-day experiment with ten users on one 30-room floor.

Other location systems also perform motion detection. Krumm and Horovitz's LOCADIO [19] uses WiFi signal strength to both localize a device and infer whether it is moving. However, due to the natural fluctuation of signal strength readings even when standing still, this motion detection's error rate is 12.6%, which results in a high number of false state transitions (e.g., from "stationary" to "moving") during experimental use (24 reported when only 14 happened).

King and Kjærgaard [16] also use WiFi to detect device movement, reporting results similar to Krumm and Horovitz's on a wider variety of hardware. They use motion data to minimize the impact of location scanning on concurrent communications: If the device is stationary, the system does not need to recompute its position (which might interfere with regular communications as both activities share the same WiFi card). In contrast, we use motion information not only for positioning, but also to aid the training: If the device is stationary, the system can collect stable WiFi measurements.

Accelerometer-based detection of stability has been a popular research topic in the context of activity recognition [3, 15, 20, 22]. Kern [15] showed how to use an accelerometer to distinguish a moving state (walking, running, or jumping) from a stationary state (standing or sitting). Since our use emphasizes the stability of the device, rather than the activity of the person, we do not require a worn device. Moreover, many of today's laptops, PDAs, and mobile phones already contain accelerometers to protect system operations (by parking disk heads before possible impact) or to support the user interface (by detecting screen orientation).

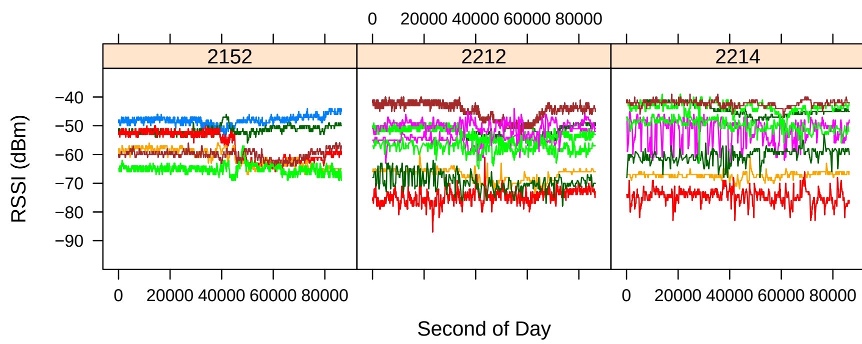
3 Interval Labeling with PILS

The method of location fingerprinting using WiFi signals assumes that the pattern of *mean signal strengths* received in one location differs from the pattern observed in another location. Unfortunately, various effects, including interference from interposing static objects as well as reflections off neighboring objects, make the relationship between the signal means and location difficult to predict in practice [11, 18, 21, 24]. Less well-documented are sources of variance in the signal, although there has been some work studying these effects over a one day period [13].

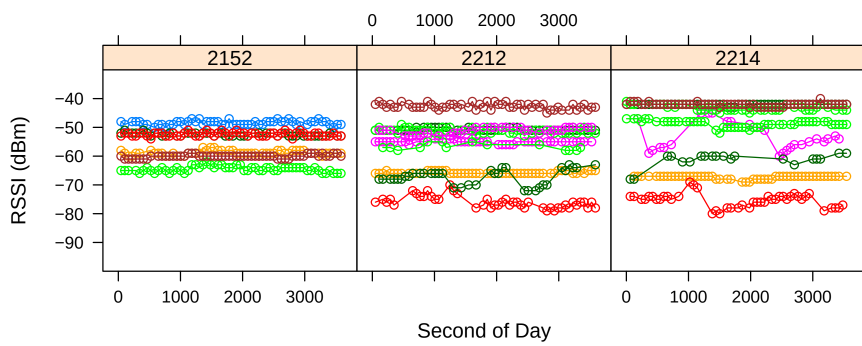
To understand the significance of those mean variations in signal strength, we performed a small experiment. We positioned five laptops in different rooms of our office building. For one week, each laptop did an active WiFi scan every minute and recorded the access points' unique identifiers (BSSID) and received signal strengths (RSS). Figure 1(a) shows the signal strength variation for three laptops over the course of a day. Different lines correspond to the signal strengths from different access points. Rooms 2212 and 2214 are adjacent to each other, and Room 2152 is further away. Room 2212 and 2214's patterns resemble each other much more than either of them do 2152, illustrating how these readings can be used to determine position. However, the graph also shows that there is short-term variation from minute-to-minute as well as longer-term fluctuations. The short-term fluctuations arise not only from the motion of people—average per-access point variance on low-traffic weekends was still 68% of the variance during the week. Additionally, different access points have different variances. Figure 1(b) shows the detail of the first hour, with individual scans now indicated by circles. This shows how readings can appear in scans at different rates independent of the mean received signal strengths.

The long-term variance, which is especially noticeable during the day in Room 2212, shows that for nearby locations it may not suffice to build the radio map only once. The best way to cope with long-term variance is to update the map frequently by taking measurements at different times of the day and days of the week. This addresses not only variations of unknown causes, but also infrastructure changes such as failing or replaced access points.

These signal traces also show that the best way to reduce the error caused by short-term signal variance is to average a large number of measurements taken during a short time. However, collecting measurements is tedious and not something an end-user is very eager to do. So, two challenges are: How can a



(a) RSSI measurements over the course of a day



(b) Detail from above showing the first hour

Fig. 1. Signal strength variations from three laptops. Rooms 2212 and 2214 are adjacent to each other, and Room 2152 is further away. Signal variations happen on different timescales, ranging from a few minutes to several hours.

system get users to contribute many labeled measurements to the system *without interrupting their work routine*? And how can a system continue to update the radio map over days and weeks, again *unobtrusively*?

Our method of interval labeling addresses these two challenges. Labels provided by end users are applied not only to the immediate signal strength measurement, but to all measurements taken during the interval while the device was stationary, at the same place. Figure 2 gives an example of the process of interval labeling. Using data from the accelerometer, PILS partitions time into alternating periods of "moving" and "stationary" as indicated in the second row of the figure. (The implementation of the motion detection process is described in Section 4.3.) Whenever the system is stationary, it continuously adds measurements to the interval. When it detects movement, it stops taking measurements until the device rests again, at which time a new interval begins.

In addition to increasing the number of WiFi measurements that can be associated with a location label, intervals can improve the user experience of labeling. Because intervals are known to be periods of immobility, they can be more easily labeled asynchronously. A user is more likely to remember their location during the entire interval (knowing its starting time and duration) than they are likely to remember their location at a specific instant. This gives the system the freedom to postpone labeling until a more convenient time such as the start of the next stationary period, or when the user returns to their desk. This can help the system reduce the obtrusiveness of any explicit location prompts.

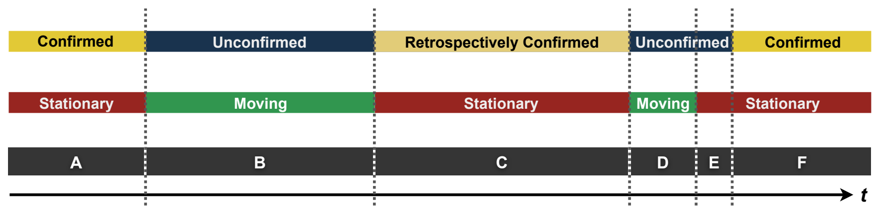


Fig. 2. Interval labeling allows the user to update the radio map with all data taken while the device is stationary. Because intervals provide more cues to users (starting time, ending time, and duration of interval), users are more likely to remember where they were during an interval than at an instant.

4 The PILS System

Figure 3 gives an overview of the three main system components: a *scanner* to listen for announce beacons, a *locator* to compare current measurements with the assembled radio map from a fingerprint database, and a *motion detector* to inform the locator about interval boundaries (i.e., changes between the moving state and stationary state).

4.1 Hardware

PILS requires a WiFi communications module and an accelerometer in the terminal—two components that are often available in today's laptops and high-end mobile phones. We implemented our initial prototype on Mac OS X 10.5 using MacBook Pros (revision B and C).

In particular, the 15-inch machines that we used featured a WiFi network card from Atheros or Broadcom, as well as a Kionix KXM52-1050 three-axis accelerometer with a dynamic range of +/− 2g and bandwidth up to 1.5KHz.

In our office environment, there were sixteen access points to cover about 70 rooms with a combined area of 1000m^2. Access points were installed at a density of about 0.23 access points per room, or 1 access point per 62.5m^2 of office area. Typically five access points were visible at any location.

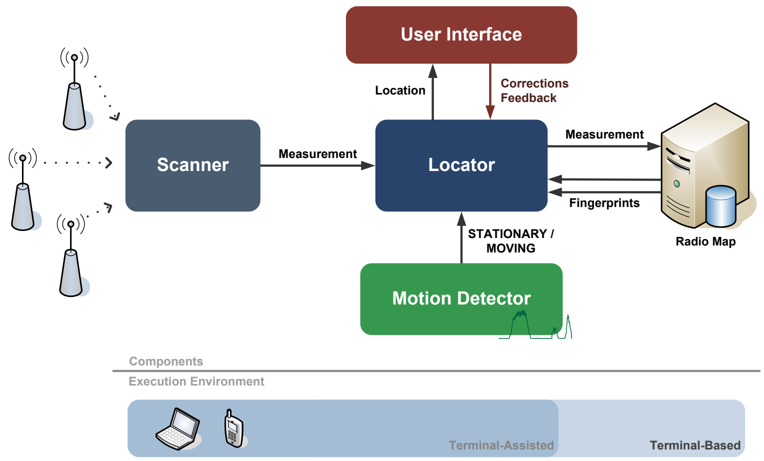


Fig. 3. Our terminal-based system has four components. The signals observed by the Scanner are sent to the Locator, which estimates the location from the radio map stored in the Radio Map. The Motion Detector informs the Locator whether the device is stationary or moving, and the User Interface collects the labels. For low-power devices, a terminal-assisted approach could outsource location estimation to a central server.

4.2 Locator

Our approach to location fingerprinting is to learn a probabilistic model of the likely readings of received signal strength (RSS) of WiFi beacons for each location we are interested in. With these learned models, we estimate the device's location by choosing the model that gives the maximum likelihood. Our probabilistic model is similar to the approach taken by Chai and Yang [7], except that we use normal distributions for RSSI rather than quantizing RSSI values and using histograms. As long as the RSSI values are not multi-modal, such a unimodal approach still offers good performance while being computationally much simpler. By keeping only the mean and variance, updates are very fast and do not use much memory. In addition, the larger number of free parameters in a histogram approach is more susceptible to overfitting when there is not much data.

Each received signal strength reading is stored as a pair consisting of the access point's BSSID and the measured indicator of its signal strength, i.e., $b_t = (BSSID_t, RSSI_t)$, with $RSSI_t$ being the received signal strength from the WiFi access point with unique identifier $BSSID_t$ at time t.

For each location l we learn a model of the readings received by a device in location l. For a set of n readings $\{b_1, ..., b_n\}$ in location l, we adopt the following model for the likelihood of the set of readings:

$$P_l(b_1, ..., b_n) = \prod_{i=1}^{n} p_l(BSSID_i) \cdot N(RSSI_i; \mu_l(BSSID_i), \sigma_l^2(BSSID_i)) \quad (1)$$

where N is the normal distribution and $p_l(BSSID)$ is the probability that the reading in location l comes from WiFi access point BSSID. We model each reading to be independently generated from a normal distribution with mean $\mu_l(BSSID_i)$ and variance $\sigma_l^2(BSSID_i)$, which can be different for each access point.

Given a set of n readings $\{b_1, ..., b_n\}$ in location l, the model parameters which maximize the likelihood of the readings are given by:

$$p_l(bssid) = \frac{R_{bssid}}{n}$$

$$\mu_l(bssid) = \frac{1}{R_{bssid}} \sum_{i:BSSID_i=bssid} RSSI_i$$

$$\sigma_l^2(bssid) = \frac{1}{R_{bssid}-1} \sum_{i:BSSID_i=bssid} (RSSI_i - \mu_l(bssid))^2$$

where $R_{bssid} = |\{b_i|BSSID_i = bssid\}|$ is the number of readings that came from WiFi access point *bssid*. Note that a location l will not get readings from all access points. For those access points which were not part of the readings for learning the model, we set $p_l(bssid)$ to a very small value, e.g., 10^{-15}. The parameters $\mu_l(bssid)$ and $\sigma_l^2(bssid)$ can be chosen in any way as long as the product of p_l and the normal distribution is small.

To estimate the most likely location $\hat{l}$ from a set of readings $\{b_1, ..., b_n\}$, we can compute Eq. 1 and find the maximum likelihood location as follows:

$$\hat{l} = \arg_l \max P_l(b_1, ..., b_n) \ .$$

In practice, we compute $\log P_l(b_1, ..., b_n)$ because it is numerically stable and the monotonic property of the logarithm guarantees the same answer for $\hat{l}$.

4.3 Motion Detector

The motion detector is a discrete classifier that reads the accelerometer to determine whether the device is being moved or whether it is stationary. Classification needs to be somewhat forgiving so minor movements and vibrations caused by readjusting the screen or resting the computer on one's lap are still classified as "stationary." Only significant motion such as walking or running should be classified as "moving."

To classify the device's motion state, the motion detector samples all three accelerometer axes at 5 Hz. It then calculates the acceleration magnitude and subtracts it from the previously sampled magnitude. To prevent misclassification of small movements as "moving," the signal is smoothed into a moving average of the last 20 values. Figure 4 shows that this method yields a sharp amplitude increase in the magnitude delta whenever the user is walking. The classifier includes hysteresis with different threshold values when switching between the moving and stationary states. The exact threshold values were established

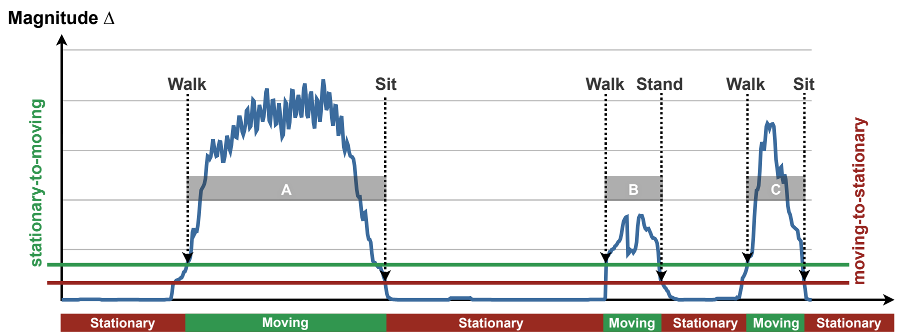


Fig. 4. Example data from the motion detector. As soon as the magnitude delta exceeds the stationary-to-moving threshold, the device is considered to be moving. This holds as long as the magnitude delta does not fall below the moving-to-stationary threshold.

through a series of informal experiments. Figure 4 shows the motion magnitude trace of a user going from his office to a colleague's office (A) and back (B and C), with two stationary phases in between: a longer discussion at the colleague's office and a brief chat in the hallway. The sequence bar at the bottom of the figure shows the motion detector's output. Due to the use of a moving average, the system imposes a small delay of 2-4 seconds before values fall below the threshold for the stationary state, however this does not appear to degrade performance.

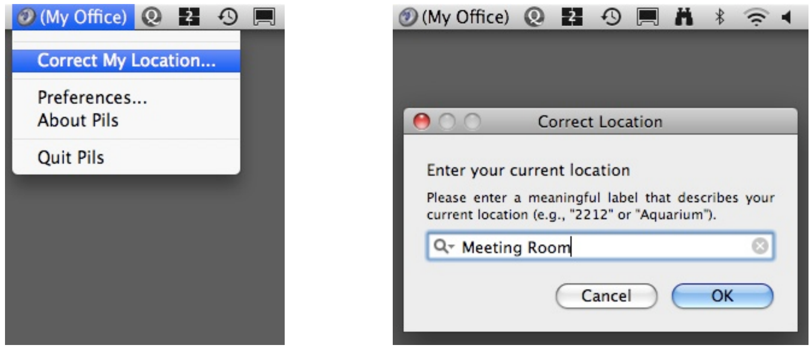
5 Evaluation

To get a better sense of whether interval labeling would work well in practice, we conducted a user study. The study examined whether users would voluntarily correct incorrect location predictions, what the characteristics of the labeled intervals were, and whether labeling increased the system's confidence in the user's location.

5.1 Experimental Setup

We recruited 14 participants, all researchers at one of our institutions. Participants installed a custom application on their MacBooks. The software placed an extra menu in the right side of the menu bar, as shown in Figure 5. Users were instructed to correct the system if they saw that it incorrectly guessed the location. This was also the mechanism for adding new labels to the system. The users gained no benefit from the application other than the satisfaction of making the correction. The study ran for five weeks, which included the winter holiday period.

To remind users about the study and to provide additional feedback to the user about the system's inferences, the user could optionally enable a voice announcement of "moving" and "stationary" when the device transitioned between



(a) The user corrects an erroneous inference through the “Correct My Location...” menu option.

(b) The user can enter any label for the current location by a simple dialog.

Fig. 5. User interface for collecting label corrections: The system’s prediction of the room is placed in the menu bar to provide ambient awareness.

moving and stationary states. Music could also optionally be played while the device was in the moving state. However, as the laptops went to sleep when their lids were closed, the music typically did not continue for the entire moving duration.

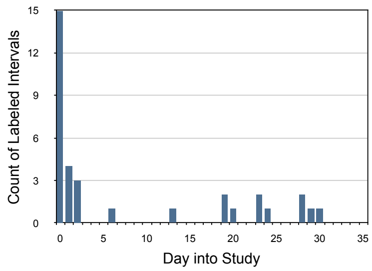
Location inferences were made on the users’ laptops, however all WiFi measurements and labeled data were uploaded to a server for later analysis.

5.2 Results

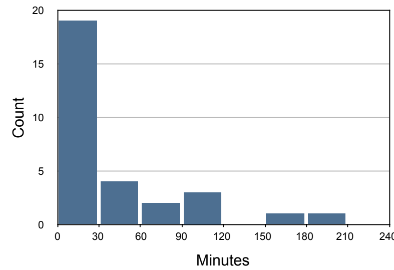
WiFi Scans and Label Frequency When running, the program conducted an active WiFi scan once every five seconds. A total of 322,089 WiFi measurements were taken. Each scan contained on average 6.6 beacons, with a standard deviation of 4.4.

Users labeled 31 intervals, with a surge on the first day, and declining frequency afterward (see Figure 6(a)). However, users continued to correct the system at a roughly constant rate until the end of the experiment, despite not receiving any reminders about the study other than the ambient awareness in the menu bar. Furthermore, continued labeling was not concentrated in a couple individuals—the contributions after the tenth day came from five different participants. All these results suggest that providing corrections is a low-overhead activity that can be sustained for at least a month.

Interval Characteristics Figure 6(b) shows a histogram of interval durations. Most intervals were only a few minutes long. Of those under a half hour, five lasted less than a minute, and sixteen less than ten minutes.



(a) Number of new labels added per day. Around a third of the labels were added on the first day. The decline and later uptake in labeling likely resulted from the holiday schedule.



(b) Histogram of labeled interval durations. Most intervals lasted less than a half hour. Note that there is an outlier not shown on the graph at 21.3 hours.

Fig. 6. Label Frequency and Interval Durations

Generally, users provided labels at the beginning of an interval. 28 intervals were labeled within the first two minutes. Of the remaining three intervals, one was labeled at the end of a half-hour interval, and two others were labeled in the middle of multi-hour intervals. From these observations we conclude that since users chose to enter corrections when arriving at a new place, this is the best opportunity for a more proactive system to query users for location data.

Benefits of Labeling Intervals To understand how much the system benefitted from interval labeling, we examined the recorded data more closely. A sample of 1,000 WiFi measurements was drawn. Each scan was classified according to its most likely location, given the labels that the system knew about at the time the scan was taken. Two classifiers were compared, one that learned from all WiFi scans in previously labeled intervals, and another that learned only from the WiFi scan at the instant a label was assigned.

Figure 7 compares the distribution of maximum log-likelihoods for the class returned by each classifiers. The graph does not include the scans whose WiFi likelihood scores were zero, as explained in the caption. For the over 92% of scans in the instant labeling condition, the likelihood value gives no information about which label is best. Likelihood values can be computed, however, for over half of the scans in the interval labeling condition. Furthermore, even when a likelihood value is computed, the values are, in general, relatively higher in the interval labeling condition, which indicates greater certainty of the answers.

5.3 Survey

Following the user study, we surveyed participants to better understand the user experience. We felt that it was important to get users' perspective on both the

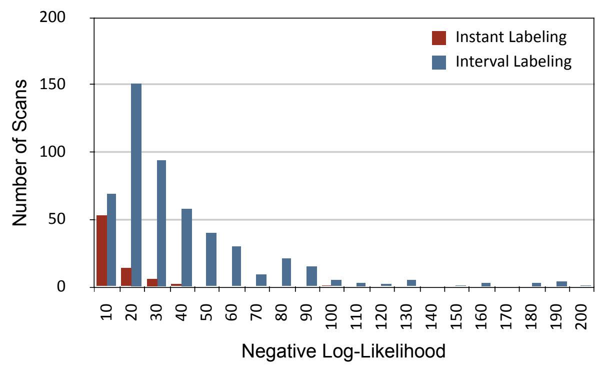


Fig. 7. Distribution of the log-likelihoods of 1,000 random WiFi scans, excluding those with zero likelihood (which include 484 for Interval Labeling, and 924 for Instant Labeling). The proportionally higher likelihood scores indicates that WiFi scans are more likely to find labels when using Interval Labeling than when using Instant Labeling.

accuracy of the system as well as the overhead involved in collecting the labels. Eleven of the participants responded to the survey.

Participants' perceptions about the system accuracy were mixed. On a Likert scale from 1–7, where 1 stands for "strongly disagree," responses to "PILS often showed a wrong or missing label" had a mean of 3.0 and standard deviation of 1.9. But in response to "the accuracy got better over time," responses averaged 4.3 with a standard deviation of 0.8.

In free responses, participants offered several improvement suggestions, such as reducing the latency to make an estimate and improving the autocompletion of labels. Two participants appreciated the music that played when the laptop was moving. One found it to be not only a useful form of feedback about the system's operation, but also an interesting prompt for social engagement. The other wanted to be able to choose the music from their iTunes library.

6 Discussion and Future Work

Previous work by King and Kjærgaard [16] has shown that knowing whether the user or device is in motion can be beneficial for several reasons. In our system, motion detection allows us to improve end-user data collection by supporting interval labeling instead of single measurements only. We use a very simple heuristic to differentiate between stationary and mobile intervals, yet it worked well in our prototype. In only very few cases did the motion detector report a false stationary state, while false reports of moving states never occurred.

Although these results indicate that prompting users for feedback when they arrive at a new place could minimize interruptibility, we did not focus on this aspect in this work. We plan to more thoroughly investigate this process in the future. We envision several options worth exploring, such as the active applications on the device, the time of day, or mouse and keyboard activity. We also plan to incorporate the results from interruptibility research into this process [14, 23], as well as games for user labeling [26]. For example two users might give the same label for a room to win points.

Asynchronous labeling can also ensure that only "important" labels are solicited, such as the places that the user stays for long time periods or visits repeatedly. If the user stays at an unknown place for only a few minutes, PILS can omit the prompt, thus further reducing the intrusiveness of the system.

Our initial results from both the experimental study and the survey give a strong indication that the accuracy of location fingerprinting can be improved by interval labeling. However, about one third of the survey participants reported that accuracy seemed to decline over time, which could have arisen from long-term signal fluctuations or overfitting effects in the radio map. Consequently, we plan to evaluate how long to keep old measurements in the radio map, and the optimal number of measurements in a location fingerprint.

7 Conclusion

In this paper, we have presented a method to improve end-user supported location fingerprinting. By using the built-in accelerometer to detect motion, it is possible to observe and record WiFi signals during long periods. This greatly increases the number of WiFi measurements associated with a single given label. In addition, by making intervals the unit of labeling, the labeling process can be performed at a less obtrusive time, since users are more likely to recognize intervals of stability than they are to recall their locations at instants. Motion detection can also reduce the computational burden of inferring location when a label is given.

Our user study shows that labels can be collected without greatly burdening users, and that when such labels are applied to intervals, the maximum-likelihood of a new WiFi measurement is much higher than it would be if only instants were labeled.

We plan to further investigate how to improve accuracy through interval labeling. Moreover, we intend to study how to use retrospective labeling to increase the amount of labeled data while minimizing user effort.

References

1. D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.
2. P. Bahl and V. Padmanabhan. Radar: an in-building rf-based user location and tracking system. *INFOCOM, Tel Aviv, Israel*, Jan 2000.
3. L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive Computing (PERVASIVE)*, pages 1–17, Jan 2004.

4. E. Bhasker, S. Brown, and W. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. *Pervasive Computing and Communications (PerCom)*, Jan 2004.
5. P. Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. *Workshop on Mobile Entity Localization and Tracking in GPS-less Environment Computing and Communication Systems (MELT), San Francisco*, 2008.
6. P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A probabilistic room location service for wireless networked environments. *Ubiquitous Computing (Ubicomp)*, pages 18–34, Jan 2001.
7. X. Chai and Q. Yang. Reducing the calibration effort for location estimation using unlabeled samples. *Pervasive Computing and Communications (PerCom)*, Jan 2005.
8. J. Froehlich, M. Chen, I. Smith, and F. Potter. Voting with your feet: An investigative study of the relationship between place visit behavior and preference. *Proceedings of Ubicomp 2006*.
9. A. Haeberlen, E. Flannery, A. Ladd, and A. Rudys. Practical robust localization over large-scale 802.11 wireless networks. *International Conference on Mobile Computing and Networking (MobiCom)*, pages 70–84, Jan 2004.
10. A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2-3):187–197, Jan 2002.
11. A. Hossain, H. Van, Y. Jin, and W. Soh. Indoor localization using multiple wireless technologies. *Mobile Adhoc and Sensor Systems (MASS)*, Jan 2007.
12. Y. Ji, S. Biaz, S. Pandey, and P. Agrawal. Ariadne: A dynamic indoor signal map construction and localization system. *International Conference On Mobile Systems, Applications And Services (MobiSys)*, (151-164), Apr 2006.
13. K. Kaemarungsi. Design of indoor positioning systems based on location fingerprinting technique. *Dissertation, School of Information Sciences, University of Pittsburgh*, Jan 2005.
14. N. Kern, S. Antifakos, B. Schiele, and A. Schwaninger. A model for human interruptability: experimental evaluation and automatic estimation from wearable sensors. *International Symposium on Wearable Computers (ISWC)*, 2004.
15. N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Troster. Wearable sensing to annotate meeting recordings. *Personal and Ubiquitous Computing*, Jan 2003.
16. T. King and M. B. Kjaergaard. Composcan: adaptive scanning for efficient concurrent communications and positioning with 802.11. *International Conference On Mobile Systems, Applications And Services (MobiSys)*, pages 67–80, Jan 2008.
17. T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. *Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, Aug 2006.
18. D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. *Technical Report TR2003-467, Dartmouth College*, Jan 2003.
19. J. Krumm and E. Horvitz. Locadio: Inferring motion and location from wi-fi signal strengths. *Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, 2004.
20. J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. *Pervasive Computing (PERVASIVE)*, pages 1–16, Jan 2006.
21. H. Lim, L. Kung, J. Hou, and H. Luo. Zero-configuration, robust indoor localization: Theory and experimentation. *INFOCOM, Barcelona, Spain*, 2006.

22. M. Mathie, A. Coster, N. Lovell, and B. Celler. Detection of daily physical activities using a triaxial accelerometer. *Medical and Biological Engineering and Computing*, Jan 2003.
23. Y. Noy, T. Lemoine, C. Klachan, and P. Burns. Task interruptability and duration as measures of visual distraction. *Applied Ergonomics*, Jan 2004.
24. S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu. Transfer learning for wifi-based indoor localization. *Association for the Advancement of Artificial Intelligence (AAAI) Workshop*, page 6, May 2008.
25. B. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The parctab mobile computing system. *Workstation Operating Systems*, Jan 1993.
26. L. von Ahn and L. Dabbish. Labeling images with a computer game. *Conference on Human factors in computing systems (CHI)*, Jan 2004.