

Interactionless Calendar-Based Training for 802.11 Localization

Andrew J. Barry, Noah L. Tye, Mark L. Chang

Franklin W. Olin College of Engineering

Needham, MA, USA

andrew.barry@alumni.olin.edu, noah.tye@students.olin.edu, mark.chang@olin.edu

Abstract—This paper presents our work in solving one of the weakest links in 802.11-based indoor-localization: the training of ground-truth received signal strength data. While crowdsourcing this information has been demonstrated to be a viable alternative to the time consuming and accuracy-limited process of manual training [2], one of the chief drawbacks is the rate at which a system can be trained. We demonstrate an approach that utilizes users’ calendar and appointment information to perform interactionless training of an 802.11-based indoor localization system. Our system automatically determines if a user attended a calendar event, resulting in accuracy comparable to our previously published large-scale crowdsourced deployment. We find that no other user interaction is necessary to train the system to that level of accuracy when calendar data are available. In ideal conditions, this technique can reduce training time by over a factor of six.

Keywords—Location measurement; calendar; crowdsourcing; localization; location representation; location-based services

I. INTRODUCTION

As computing becomes increasingly mobile, location systems, and the applications that leverage them, become more a part of everyday life. With the proliferation of laptops, sensor-laden smartphones, a blanket of wireless access points, and hybrid localization techniques, consumers are beginning to demand location-aware capabilities of their hardware and software. For most of these devices, localizing outdoors with GPS is accurate and provides good coverage. But once indoors, or in an otherwise GPS-denied environment, coverage and accuracy suffer dramatically. Commercial solutions, such as those from Skyhook Wireless [13], attempt to provide location by using a combination of GPS, cell-tower triangulation, and 802.11-based techniques. In GPS-denied environments, the localization relies primarily upon a database of access point signatures acquired by a fleet of 802.11-scanner-equipped vehicles that are limited to scanning from public roads. Therefore, while the quality of location data may be adequate in and around some public indoor spaces, private spaces—large corporate offices for instance—will not necessarily be well covered.

To localize accurately indoors, most proposed systems catalog the received signal strength (RSS) of 802.11 access-points throughout an indoor space, and use those signatures to compare against a user’s current signal scan results to determine location [5], [4], [1], [15]. Most systems require an extensive manual training phase in order to achieve usable

accuracy. We proposed a crowdsourced system [2], similar to [14], [3], that requires very little interaction and minimal dedicated training time, yet achieves very high (room-level) accuracy. One of the oft-cited drawbacks of crowdsourced data is the long effective training time required to achieve good results. In effect, it is the chicken and the egg problem: without good manual training data, users will not utilize (and therefore train) the system; but without users utilizing the system (and actively providing training data), the system will not improve its accuracy.

In this work, we propose a novel solution to this “cold start” problem. By overlaying calendar and appointment data that is tagged with location information (such as a meeting room or office), every report from a user’s device will be automatically bound to a location. Thus, instead of requiring a user to manually associate semantically-relevant room information with a signal scan (*fingerprint*), we do so automatically. Because users’ calendars are not perfectly accurate, we treat every calendar/RSS pair as a noisy sensor. Our use of a clustering algorithm allows us to determine when a user attended or skipped an event, enabling us to extract accurate location data.

II. RELATED RESEARCH

There has been a rich history of published work that attempts to solve the indoor localization problem. Beginning with the Active Badge [15] and Cricket [12] systems, researchers were able to demonstrate reasonable room-level localization. However, these systems required specialized hardware that had to be location-bound and installed by trained personnel.

The growing ubiquity and dense coverage of 802.11 access points saw researchers use them as fixed-location radio beacons. Microsoft’s RADAR [1], and later Haeberlen, et. al. [4], show successful room-level indoor localization using just 802.11 signal strength information. These approaches, while accurate, require the system to be trained to create a database of location and signal strength tuples. This necessitates a substantial up-front investment in time and effort, something that may be a barrier to adopting an indoor localization system. Many more infrastructure-focused approaches are described in [5].

The focus of research then shifts to alleviating the burden of up-front training and high infrastructure costs. Intel Re-

search demonstrated an algorithm that can estimate location through proximal sensing, and expands its known area with continued use [7]. While the initial cost of this system is minimal, the time required to achieve acceptable localization accuracy and wide coverage is high.

Closer to our own work, in approaches described by Teller et. al. [14] and Bolliger [3], training and correction data are collected using a crowdsourced approach rather than via trained personnel. Instead of triangulating estimated absolute positions, these approaches estimate position by returning symbolic representations of physical spaces. This is accomplished by comparing wireless beacon scans to stored *fingerprints* of signal strengths collected at user-annotated locations. The advantages of crowdsourced data sources are obviating the requirement for significant up-front training, and quality training of the system by users in actual physical places of interest.

Unfortunately, purely crowdsourced training data has one significant drawback: the acquisition of training data is completely dependent on user participation. In the next section, we describe our novel approach to solving this “cold-start” problem without requiring any separate intervention from the user.

III. INTERACTIONLESS TRAINING WITH CALENDARS

Almost every medium- to large-scale organization occupies some sort of dedicated physical space; from a floor of a larger building, an entire building, to even an entire campus spanning several buildings. It is not uncommon for these organizations to employ some sort of shared calendaring mechanism, such as Microsoft Exchange, Google Calendar, or Lotus Notes, to help their employees both coordinate work and find one another for interactions. In the latter, the calendar can almost serve as a sensor [9]—providing room-level localization information with some degree of certainty.

Instead of relying on users to manually associate a vector of signal-strengths in *signal space* with a semantically-relevant *physical space*, we can automate training by using the location field of shared appointments in users’ calendars as the physical space annotation for each signal scan. In the ideal case users’ mobile devices, such as laptops and smartphones, would be loaded with signal scanning software, and every user would share accurate and complete appointments in their calendars. With this approach, as people move around to different physical locations to attend scheduled meetings, the system would be trained with accurate data automatically.

There are several benefits to this approach. Most importantly, the “cold-start” issue is largely eliminated. With no infrastructure costs other than the installation of software, users will train the system with accurate data as they progress through their normal day without changing their behavior. With a wide deployment of scanning software, the system will be able to localize accurately in all visited spaces

almost immediately. Secondly, this approach no longer requires any user intervention other than allowing software to be run in the background of their mobile devices to scan and report to a central server. In a sense, it is a modified form of crowd-sourcing, requiring *no actual interaction* from the crowd. Finally, this approach will continually provide new localization data. With the right filtering or clustering mechanisms, the system can continually adapt to changes in the environment, such as the movement of access points, deployment of new access points, the reconfiguring of physical spaces, and the addition of new physical spaces.

Calendar-based training does have some limitations, however. The most obvious is the reliance on location-annotated, shared calendar appointments. If users do not utilize this field in their appointments, or if users do not share their appointment data, system accuracy will suffer. In particular, if a meeting of many people has no annotation of location, that space will not be trained. However, the lack of calendar data would have to be widespread among users to significantly affect the system. As long as there are some users with annotated and shared appointment data that utilize that room, the space will be successfully trained given enough time.

We note that relying on appointment event locations causes the localizer to succeed only in areas where users have specified these events. Locations without such events, such as warehouses, corridors, factories, residence halls, or apartment complexes will exhibit particularly poor performance. Thus, a calendar-trained localizer is most applicable in an office or other structured environment where users regularly schedule appointments and share their calendars.

Perhaps more serious than a lack of data, however, is the case when a user annotates an appointment, but does not take their mobile device to that location. If we treated each calendar/scan intersection as valid training data, this would become a problem. Fortunately, we demonstrate that we can successfully detect and mitigate this situation by employing a clustering algorithm to filter inbound data. So long as we have a plurality of users in the annotated location, the algorithm will ignore these false data.

What cannot be accounted for easily is if all attendees at a meeting with annotated calendars hold the meeting at a different location than noted in the appointment. Since there is no convenient mechanism to override the calendar data other than every attendee changing the appointment location information in their calendar, the system will incorrectly annotate the wrong location. This, however, can be mitigated using the same clustering technique. If there are enough correct localizations in the new location, the moved meeting will be ignored. And even if the new meeting location is incorrectly trained, with enough correct training data, the cluster will eventually move to be associated with the correct location.

IV. EXPERIMENTAL DESIGN

To demonstrate the viability of our system, we perform two experiments. In the first, *Playback Simulation*, we combine our extensive existing localization database (from [2]) with calendar data from our users. We simulate, over time, the training of our system with only calendar/fingerprint intersections—exactly as if we had deployed a calendar-based training system from the beginning—and compare the results with our previous crowdsourced results.

In the second experiment, *Ideal Deployment Simulation*, we motivate a more likely use case: that of wide-scale deployment within an organization that has more consistently location-annotated calendar data, such as a corporate office. We compare these results to all of our previous results.

A. *Playback Simulation*

We chose to implement this experiment using our currently running localization system’s recorded wireless access point data (tagged with a user ID) and publicly available calendar archives. Using logged data presents an issue, however, in that it is time invariant. Theoretically we could access all two years of data to perform correlations which are not possible when training the system in real time. To emulate a deployed system, we perform all tests over a period of simulated time. As the simulated time moves forward, more data becomes available to the localizer based on when users actually provided valid calendar/fingerprint intersections. In this way, we assess the localizer’s accuracy as it would have appeared throughout the past two years.

To determine appointment and wireless data intersections, we capture a user’s wireless scan data during the entirety of his or her appointments’ durations. We expand appointments that include repetition to consider every instance of the event. One might consider limiting the time range that appointments correlate to scan data to reduce error when users are late or leave early, but we find that the signal-space clustering algorithm is adequate for this task.

To test our system’s performance we captured signal strength data in every unlocked and accessible room in the academic building. In each room we recorded two wireless scans and the true room number. These data allow us to determine the true accuracy of our system and compare the performance of the calendar-trained system to the user-trained system.

It is important to note that users do not use a completely consistent naming schema for rooms, although many users label rooms in a form similar to “AC 204,” where “AC” is consistent and the room number varies. We chose to filter our data to include only rooms labeled in this way, although in the future work section we describe our thoughts on extending this parser to garner both denser and better labeled data.

B. *Ideal Deployment Simulation*

The *Playback Simulation* does not necessarily demonstrate the expected use-case, as the existing data was trained based on opt-in usage of a small subset of our community. In contrast, we anticipate the benefit of using our calendar-based approach is for *new* installations of localization systems into organizations that utilize a calendaring system throughout the entire organization. In order to demonstrate the potential effectiveness of our approach, we simulate an idealized environment.

One idealized environment might be a corporate setting, determined both by the personnel and their behavior, as well as the physical spaces. We envision employees of this organization to utilize company-issued mobile devices—both smartphones and laptops—have average work days that include some meetings with others, and have location-annotated appointments in their calendars. Deployment of the system then only consists of installing a scanning and reporting application on all devices. We argue that with managed IT services and an expectation of calendar usage, these assumptions are within reason for any organization that would consider deploying a localization system.

In this simulated environment, we will show that we can achieve very high rates of accuracy in an extremely short time span with nearly zero investment in training and infrastructure. In essence, we obtain all the benefits of crowdsourced data without the long time-lag often associated with the approach.

We choose to use our institution’s course schedule to simulate this type of environment. For one semester, we determined when and where every class was taught along with each classes’ enrollment information. We then ran four simulations, assuming that 10%, 25%, 50%, and 100% of students were using our system. In each case, we assumed that 35% of participating users did not attend lecture. In this way, we simulated the high data-density we expect to occur in a corporate environment while retaining an appropriate amount of erroneous reports.

To provide our simulated students with ground-truth data, we used the combination of our user and calendar generated data. Whenever a simulated student “gathered” data, we copied the appropriate room’s data from those sources. For the 35% of users localizing in incorrect locations, we selected a random point.

As in our existing system, each simulated user localized every 5 minutes during all events regardless of if he or she was attending. With these data providing a simulated ground truth over time, we performed the same operations as in the *Playback Simulation*, including signal-space clustering, localization, and accuracy testing.

V. SYSTEM ARCHITECTURE

As in [2], we use a client-server architecture to perform localizations. Clients capture data and calendar information

which are transmitted to a server that determines ground truth, trains the system, and estimates locations. We extract location data from the users' calendars which, when combined with relevant wireless signal strength data, is used to train the localizer. Thus, to be used as training data, we require an event to be ongoing while signal strength scans are recorded.

A. Deployment Site

While we hypothesize that this system would perform best in a corporate setting, we use a college campus to perform our study. Olin College is a small residential engineering college with approximately 300 students and a campus encompassing over 300,000 square feet. Classes take place almost exclusively in a single building, so we limit our test to that area where calendar events are dense. We collect calendar data from a shared Microsoft Exchange system and wireless network data from students who chose to participate in the previous study focused on active user-entered localization data. To participate, students run a small client program on their notebook computer that records wireless access point data along with a user ID and timestamp.

B. Localization Method

Like in [2], [1], we use a received signal strength (RSS) indicator to perform a simple nearest neighbor search in signal space for localization. Essentially, the localizer traverses every known data-point and computes the Euclidean distance in signal space to the point under consideration. The localizer returns the point corresponding to the minimum computed distance. We chose to continue using this localizer to show that the interactionless component of the system does not require a specialized localization algorithm, and could easily be extended to the more accurate and complex systems like those in [6] and [11].

C. Calendar Integration

To acquire calendar data, we accessed users' publicly shared Microsoft Exchange calendars, converted them into the standard iCal format, and imported the iCal files into the system. While our users run Microsoft Outlook, this conversion step generalizes the system, allowing it to be used with almost any modern calendaring application. Each appointment entry provides us with some or all of the following: a user ID, a time range, and a location string. Calendar appointments that do not include all of these elements lack a critical piece of data and are thus not useful for training.

1) *Calendar/802.11 Intersection*: In addition to a user ID, a time span, and a location string, an 802.11 scan must have occurred during the appointment for it to be useful as training data. As our localizer software running on both laptops and Android-powered mobile devices reports a wireless signal strength fingerprint once every five minutes,

we are able to capture, on average, twelve points per hour of appointments.

2) *Calendar Filtering via Signal-Space Clustering*: Users are not always located where their calendar indicates. Calendars are often double-booked, users do not attend all meetings, meetings move without being updated in the system, and users do not bring their wireless devices to all appointments. Our crowdsourced localization system reports that users are located where calendar events specify 68% of the time. Therefore, to extract useful training data, we must filter the calendar appointments.

As noted in [10], even without an existing localization mechanism, the collected wireless signal strength data provides insight. In general, a user either attends a meeting in the location specified, or does not. If not, the user's actual location has little correlation to where the appointment indicates. Thus, given a reasonable meeting attendance rate, we find that clusters of accurate data are easy to find in signal space. We are able to prune almost all events where users do not attend meetings, providing us with interactionless and accurate training data.

To filter the raw data, we use a clustering algorithm based on distance in signal space. We have 76 wireless access points on campus so we cluster in 76-dimensional space. For each location we compute a mean vector by taking the arithmetic mean of each component in the individual vectors. We then calculate the Euclidean distance between each vector and the mean vector. Unlike in [10], the inherent size of a calendar dataset allows us to use a majority voting system to identify outliers, implemented by rejecting entries with a distance greater than its associated location's median. This is a harsh filter but, like [8], we have found that relatively few erroneous points can significantly reduce accuracy. Listing 1 shows this algorithm in pseudo code form.

```

for location in All_Locations:
    meanVector = ComputeMeanVector(All_Events[location])

    for event in All_Events[ location ]:
        distances .append(EuclideanDistance(event, meanVector))

    medianDistance = ComputeMedian(distances)

    for event in All_Events[ location ]:
        if EuclideanDistance(event, meanVector) < medianDistance:
            Filtered_Events .append(event)

```

Listing 1. Simple Signal Space Clustering Algorithm.

To simulate a time-dependence, we ran this algorithm once for each day, retraining and retesting the system on each iteration.

VI. RESULTS: PLAYBACK SIMULATION

A. Time-Based Analysis

We measure the quality of a localization system with two statistics, *coverage* and *accuracy*. Coverage describes the

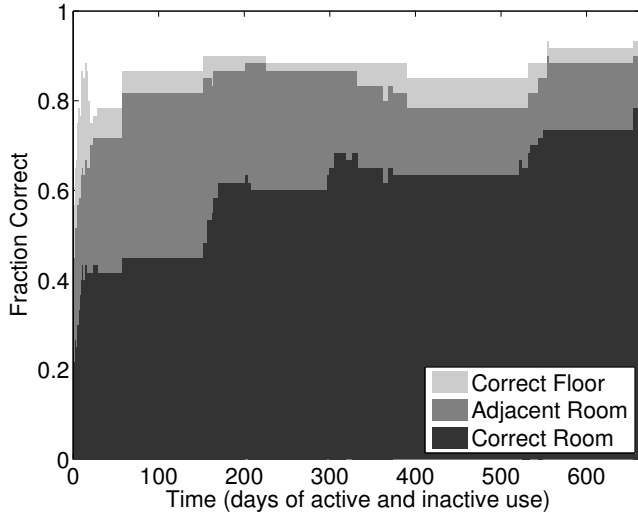


Figure 1. Accuracy of calendar-trained system over time. Dark gray indicates fraction of localizations resulting in a correct prediction. Middle gray indicates fraction of predictions in a correct or adjacent room, and light gray indicates fraction of localization predictions on the correct floor.

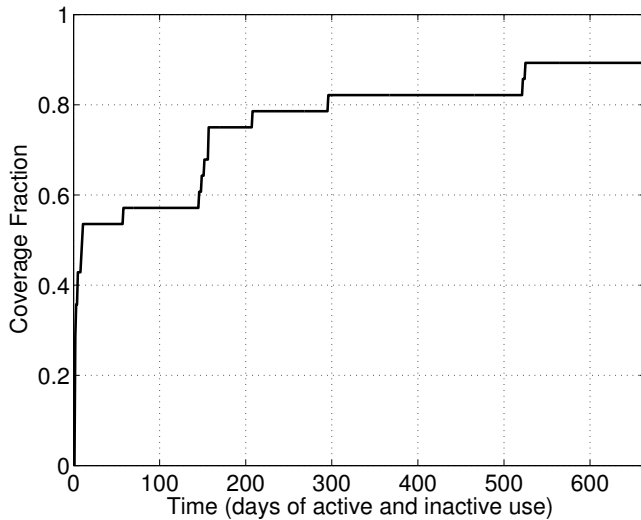


Figure 2. Fraction of rooms with training data over time. As users have meetings in new locations, the localization system’s coverage improves. The dramatic increase near Day 145 is a result of students returning to campus with a new calendar schedule. Note the strong correlation between coverage and accuracy (Figure 1).

fraction of locations with training data, while accuracy denotes how often the localizer correctly determines position.

We find that the system achieves 50% room-level accuracy (85% in adjacent rooms) after 148 days of active use (Figure 1). We define the system to be in active use when users are resident on campus, thus excluding intersession periods such as winter break. When the system is not in active use, its accuracy neither rises nor falls.

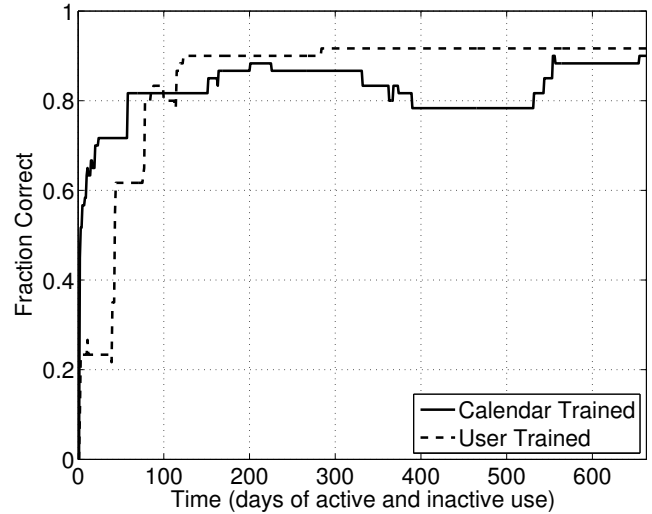


Figure 3. Comparison of correct or adjacent room accuracy over time between calendar-trained (solid) and user-trained (dashed) systems. Note that, unlike the user-trained system, the calendar based localizer is not monotonically increasing because its additional data is not always correct. The horizontal segments in the user-trained accuracy (before Day 100) are related to development periods when the system was not in active use.

Of the 28 rooms we localized, 89% had at least one signature bound to that location by the end of our experiment. Figure 2 shows how coverage in the calendar-trained localizer changes with time. After 372 days of active use, no signatures for new rooms were added to the database.

Three rooms remained without coverage: a biology lab, a professor’s research lab, and a mechanical project space. These rooms had no data because students rarely use their computers there as there are very few classes, labs, or meetings scheduled there.

B. Comparison to User-Trained Localization

The user-trained system took 18 days of active use to localize to the correct room in 50% of trials and 47 days of active use to localize to the correct room or an adjacent room in 85% of trials. Figure 3 compares the accuracy over time of this crowdsourced localizer with the calendar-trained localizer. Note that Figure 3 includes development periods (visible as plateaus near the beginning of the plot) where the user-trained system was unusable and therefore we do not regard them as periods of active use. The above statistics do not include these days.

The accuracy of the user-trained system stabilized after 55 days of active use, at 70% room-level accuracy and 90% including adjacent rooms. The calendar-based system proved slower than its user-trained counterpart, requiring 381 days of active use to achieve equivalent room-level accuracy. It took only 71 days of active use, however, to achieve adjacent room-level accuracy within 10% of the user-trained system.

The localizer’s accuracy decreases when data are added

that cause the clustering algorithm to fail. This occurs when a significant number of users are not present at their appointments but are still providing data (for instance, by leaving their laptop running at a different location), skewing the values of location cluster means.

One phenomenon observed in the crowdsourced system is that users seem to lose interest, and the number of new binds decreases. As a result, accuracy plateaus. In contrast, the calendar-trained localizer continually accumulates training data (Figure 4). Thus, the training set is in constant flux, ensuring that the system will adapt to variations in the environment such as new network access points, additional furniture, changing architecture, and even relabeled rooms.

We find that a disadvantage of the calendar-based system is that, while long-term accuracy continually improves, short-term accuracy fluctuates. The system’s constant stream of new data implies that some incorrect data is also being added, potentially reducing accuracy until enough correct data overcomes the issue. Users rarely submit bad data to the user-trained localizer, explaining why user-trained accuracy rarely decreases.

Accuracy in the calendar-trained localizer does drop after Day 373. This could be for several reasons. As already mentioned, since the calendar-trained system’s data is automatically harvested rather than explicitly entered by users, bad data is much more likely. Wireless access points are sometimes moved by network administrators to improve wireless performance. This change not only invalidates old data points but also makes the clustering filter incorrectly discard valid data points. The access points also have automatic and manual gain control, the use of which has a similar effect as physically moving the access points. Nevertheless, the localizer adapts and recovers after 37 days of active use, and after 11 more days of active use the localizer is more accurate than it was before the dip.

C. Clustering Analysis

Our clustering filter is a deliberately aggressive algorithm, discarding approximately one third to one half of all incoming data. Ideally we would evaluate the algorithm’s performance based on ground-truth data, but we do not have a mechanism to determine when users attended their calendar events. As a substitute, we use our user-trained localization system as ground truth, but only compute statistics for floor-level accuracy to mitigate the impact of the user-trained system’s localization error on our results.

On the final day of simulation, we find that the algorithm accepts 85% of the valid data and discards 76% of the incorrect data, discarding 33% of total data. Clearly, we discard a significant amount of correct, useful data. This state is acceptable, however, because incorrect training data is highly detrimental while collecting more intersections costs little.

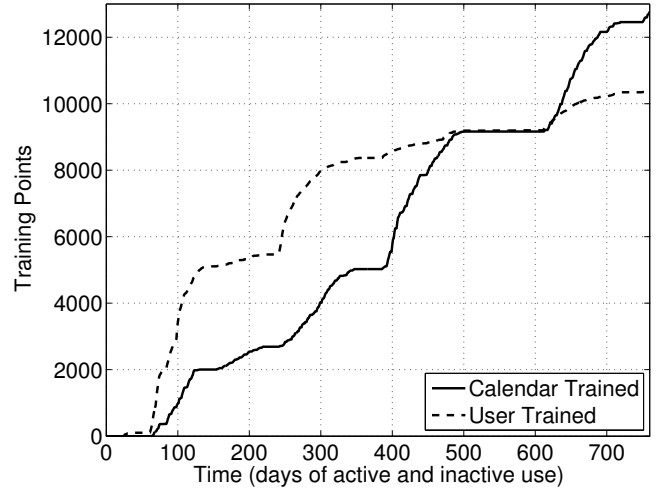


Figure 4. Comparison between calendar-based and user-based binds over time. Notice that as the system becomes more accurate, users bind less. Thus, we expect the user-trained system to become less reactive to environment changes as it ages, while the calendar-based system will continuously acquire new data. We are not surprised to see few binds for both systems in the summer and January terms when students are not on campus.

D. Experimental Details

We recorded over 1.8 million wireless scans over a period of more than two years from 278 users, of which 99 had published calendars. These shared calendars contained 47,955 events and 93,005 single event-occurrences (many events repeat). Users with public calendars accounted for 725,533 wireless scans with only 85,237 (11.7%) of these scans overlapping with a calendar event. We were able to parse location data on 28,732 (33.7%) of these. We expect that in a more favorable environment, such as in an institution-sponsored program, the amount of useful raw data available would be substantially higher.

Based on the localization data from our previously published system, we find that users are in the location that their calendar predicts in approximately 65% of cases. We evaluate our clustering algorithm based on that system’s output to estimate how successful the algorithm is. In other words, we can use our previous system to determine how well our clustering algorithm is performing.

E. User Behavior

In our calendar data we see a less typical mass interaction curve than in our previous crowdsourced approach. One quarter of users provide most of the data, however this group is much larger than that in other mass interaction applications[16]. We find that 25% of users provide 52.3% of all calendar data while 25% contribute 56.3% of all wireless network data. These data are more level than those of our user-trained system and of MIT’s crowdsourced localization system [14].

We find that recurring appointments are critical to the system’s success, accounting for 84.8% of all appointment/wireless data intersections and 87.1% of intersections determined to be in the correct place by the user-trained system. Users’ individual reliability varies greatly, giving an approximately uniform distribution when considering users with a significant number of intersections.

VII. RESULTS: IDEAL DEPLOYMENT SIMULATION

Accuracy, measured only for rooms where classes are scheduled, increases much more quickly in the data-rich environment. As the number of participants increases, the rate of accuracy increase improves in parallel (Figure 5). We note that our signal-space clustering successfully mitigates the increasing number of incorrect location updates that accompanies the additional users.

With 100% participation, we find that the system takes 8 days to achieve full accuracy at the room level, compared to the approximately 150 days for the non-ideal system. As expected, it takes longer with fewer users, taking 15, 50, and 64 days for 50%, 25%, and 10% participation, respectively. Notably, we find that to achieve stable accuracy at the floor level, all simulations require less than one week of training data.

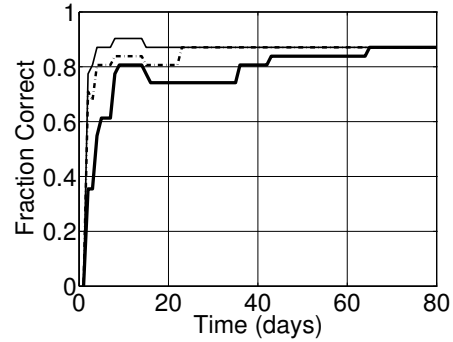
Even in the case of only 10% user participation, we find that the data-rich environment achieves similar accuracy in less than one half of the time required for the user-trained system (15% of the time required for the demonstrated calendar-based system). With increased participation, this duration decreases further, to less than 15% and 5% of the user-trained and calendar-trained systems respectively.

While the simulation’s time-to-accuracy exceeds that of the two other methods, its final-value accuracy does not. This result stems from the simulator’s reliance on the combined data of the user and calendar trained systems. The final accuracy of the user and calendar trained systems is limited to all available data, a limitation that the simulator faces as well.

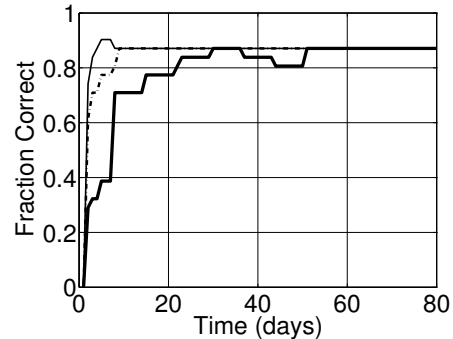
VIII. FUTURE WORK

A. Integration with Other Data Sources

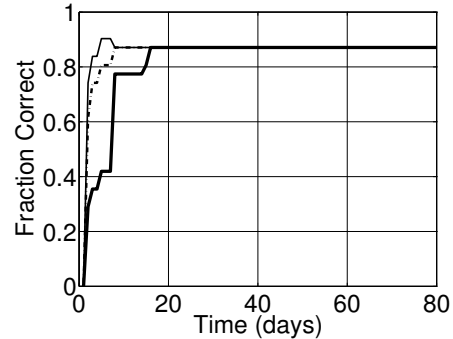
There are a multitude of other data sources that could improve our system’s localization performance. We are considering integration with a room-scheduling system and a student-course scheduling system, both of which provide user-identified location information at a specific time. We are also considering data sources that do not traditionally supply location information, such as a desk telephone. Answering the telephone indicates that a user is most likely at his or her desk, allowing us to bind all data close in signal space to the user’s workspace. We are considering this same strategy for capturing data from desktop computer idle times, screensavers, instant messaging, and even printing applications.



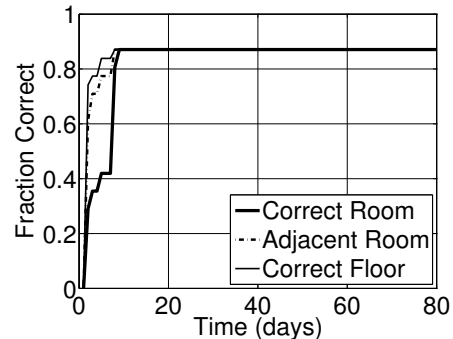
(a) 10% participating



(b) 25% participating



(c) 50% participating



(d) 100% participating

Figure 5. Simulation results for 10%, 25%, 50%, and 100% of users participating. Solid bold indicates correct room, dashed indicates adjacent rooms, and solid thin indicates correct floor. Notice that the system becomes accurate more quickly as the number of users increases.

It is important to note that this type of bind can be extremely helpful to the system. Users rarely schedule meetings at their own desk, but there is likely to be a large number of wireless scans in the area, since the user spends a significant amount of time there. Thus, extracting a bind from these non-traditional data sources allows us to correlate the past data taken at the user's desk, dramatically increasing the number of binds and thus accuracy in the area.

B. Calendar Correlation

A second method of improvement might stem from correlating calendars. We propose studying users' calendars in groups to determine which meetings users are likely to attend. Employees are very likely to attend meetings with their boss, but less likely to attend meetings where they are less important. Such correlation could be as complex as necessary or as simple as the difference between the user being in the "To" or "CC" field in an appointment notification.

C. User-Defined Location Labels

We might expand our coverage by removing the restriction that appointments must label locations in a defined format. However, user-defined names introduce the potential for duplicate location labels, uninformative names, etc. Signal-space analysis might allow us to identify duplicate labels, which could help mitigate the issue.

D. Identifying Unnamed Locations

A final technique we are considering allows the system to name locations that never appear in calendar data. These locations tend to be areas where one works alone, such as an office or dormitory. We propose correlating long stays in these areas as a way to determine their status. For example, at a workplace, a laptop might be left on overnight in a user's office. A college student might turn his or her laptop off late at night before going to sleep. If these cases occurred in a predictable and identifiable manner, we might name them "[Username's] Office" or "[Username's] room," depending on application. We believe that this type of inference, combined with new data sources and improved correlation systems, can significantly improve the performance of the localizer described here.

IX. CONCLUSION

We have proposed a novel method for training an indoor wireless localization system that utilizes location-annotated calendar data to perform interactionless training of the system. Our experiments have shown that this approach yields equal or better accuracy than crowdsourced training approaches, yet requires a fraction of the training time. We have performed experimental verification of our approach on a working database of location fingerprints, and have motivated and demonstrated the performance of more typically expected deployment scenarios.

We believe this approach is the beginning of a larger set of novel approaches that incorporate a wide range of inputs as location sensors to further improve localization accuracy, reduce training time, and provide even more robust localization in all locales.

ACKNOWLEDGEMENTS

We thank the community at Olin College for their ideas, testing, and feedback. Andrew Barry and Noah Tye are supported by F. W. Olin Scholarships.

REFERENCES

- [1] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. IEEE Infocom 2000*, pages 775–784. Proc. IEEE Infocom 2000, IEEE CS Press, 2000.
- [2] A. Barry, B. Fisher, and M. L. Chang. A long-duration study of user-trained 802.11 localization. In *Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, pages 197–212, Sept. 2009.
- [3] P. Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *MELT '08: Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 55–60, New York, NY, USA, 2008. ACM.
- [4] Haeberlen, Flannery, et al. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Sept. 2002.
- [5] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, Aug. 2001.
- [6] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Efelsberg. Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. In *Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH)*, Los Angeles, CA, USA, September 2006.
- [7] A. LaMarca et al. Self-mapping in 802.11 location systems. In *UbiComp 2005: Ubiquitous Computing*, pages 87–104. LNCS 3660, Springer, 2005.
- [8] M. Lee, C. Yu, H. Yang, and D. Han. Crowdsourced radiomap for room-level place recognition in urban environment. In *IEEE PerCom Workshop on Smart Environments (SmartE)*, 2010.
- [9] E. Mynatt and J. Tullio. Inferring calendar event attendance. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 121–128, New York, NY, USA, 2001. ACM.
- [10] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an Organic Indoor Location System. In *8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*, June 2010.

- [11] I. Paschalidis, K. Li, and D. Guo. Model-Free Probabilistic Localization of Wireless Sensor Network Nodes in Indoor Environments. *Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, pages 66–78, Sept. 2009.
- [12] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *6th Ann. Int'l Conf. Mobile Computing and Networking (Mobi-com 00)*, pages 32–43. ACM Press, 2000.
- [13] Skyhook Wireless. <http://www.skyhookwireless.com>.
- [14] S. Teller et al. Organic indoor location discovery. Technical Report MIT-CSAIL-TR-2008-075, MIT, Dec. 2008.
- [15] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 40(1):91–102, Jan. 1992.
- [16] S. Whittaker, L. G. Terveen, W. C. Hill, and L. Cherny. The dynamics of mass interaction. In *In Conference on Computer-Supported Cooperative Work (CSCW)*, Nov. 1998.