The SkyLoc Floor Localization System

Alex Varshavsky Anthony LaMarca Jeffrey Hightower Eyal de Lara

University of Toronto
{walex,delara}@cs.toronto.edu

Intel Research Seattle
{anthony.lamarca,jeffrey.r.hightower}@intel.com

Abstract

When a mobile user dials 911, a key to arriving to the emergency scene promptly is knowing the location of the mobile user. This paper presents SkyLoc, a GSM fingerprinting-based localization system that runs on a mobile phone and identifies the current floor of a user in tall multi-floor buildings. Knowing the floor in a tall building significantly reduces the area that emergency service personnel have to canvas to locate the individuals in need. We evaluated our system in three multi-floor buildings located in Washington DC, Seattle and Toronto. Our system identifies the floor correctly in up to 73% of the cases and is within 2 floors in 97% of the cases. The system is robust as it works for different network operators, when the training and testing sets were collected with different hardware and up to one month apart. In addition, we show that feature selection techniques that select a subset of highly relevant radio sources for fingerprint matching nearly double the localization accuracy of our system.

1 Introduction

The pervasiveness of mobile phones makes them an ideal platform for summoning emergency services. To ensure prompt emergency response, network operators need to be able to pinpoint the location of the mobile phone quickly and accurately. Unfortunately, despite significant research [2, 12, 8] and several FCC mandates [19] to provide accurate mobile user location information, cellular companies are still unable to determine the whereabouts of mobile users with enough accuracy. This is particularly the case for indoor environments where lack of direct line-of-sight significantly reduces the effectiveness of GPS and other localization techniques based on triangulation [12].

Automatic mobile phone-based localization is highly desirable for emergency calls as relying on the individual seeking assistance to provide localization information is both error prone and time consuming, and may not even be possible if they are disoriented or incapacitated. Moreover, accurate localization technology would make possible the wide spread deployment of life-saving devices carried by the user such as fall or heart attack monitors, which automatically call for assistance in case of emergency.

This paper reports our experiences with SkyLoc, a system that runs on a GSM mobile phone and determines the floor within a building on which a user is located. Floor level localization significantly reduces the area that emergency service personnel have to canvas to locate individuals in a large indoor environment. For example, the Empire State Building has a total floor area of 204,385 m^2 spread over 102 floors [1]. Floor-level localization reduces the area that needs to be searched by more than 99% to just 2,000 m^2 (about 18,000 ft^2). To the best of our knowledge, we are the first to address the problem of localizing users using mobile phones in tall multi-floor buildings. This is an important problem since correct floor localization in emergency situations may be a matter of life or death.

SkyLoc determines the floor on which a user is located using GSM signal strength fingerprinting. Fingerprinting relies on a training phase in which a radio map of the environment of interest is constructed by taking a series of radio measurements in multiple locations. A measurement records the strength at which signals emanating from a group of radio sources are heard at a given location. Once the training phase is complete, a client can estimate its location by matching the current measurement to the set of measurements collected in the training phase.

We tested SkyLoc in 3 tall buildings located in Toronto, Seattle and Washington D.C. Our initial experience with the system showed that simple fingerprint matching approaches used in previous research [2, 13] resulted in low floor level localization accuracy. To increase the accuracy of the system, we introduced the use of feature selection techniques [3] for matching signal strength fingerprints. We show that the implicit assumption made by previous work, whereas the larger the number of radio sources that are used for matching fingerprints the better the localization accuracy, is in fact incorrect. By using only a subset of highly relevant radio sources for fingerprint matching we were able to nearly double the accuracy of the system.

Overall, SkyLoc correctly identifies the floor up to 73% of the time and is within 2 floors up to 97% of the time. Moreover, we show that our system is robust. It works when tested across a number of GSM network operators, and when training and testing sets are collected by different hardware and up to one month apart.

The remainder of this paper is organized as follows. Background on GSM and fingerprinting is presented in Section 2. We describe our prototype implementation of the SkyLoc system in Section 3. Section 4 presents our floor classification algorithms. Section 5 describes our data collection process and presents evaluation results. Finally, we discuss related work in Section 6 and conclude in Section 7.

2 GSM Fingerprinting

Global System for Mobile Communication (GSM) is the most widespread mobile telephony standard in the world, with deployments in more than 210 countries by over 676 network operators [5]. In North-America, GSM operates on the 850 MHz and 1900 MHz frequency bands. There are 299 non-interfering physical channels available in the 1900 MHz band, and 124 in the 850 MHz band, totaling 423 physical channels.

A GSM cell is allocated a number of physical channels based on the expected traffic load and the operator's requirements. Channels are allocated in a way that both increases coverage and reduces interference between cells. The channel to cell allocation is a complex and costly process that requires careful planning and typically involves field measurements and extensive computer-based simulations of radio signal propagation. As a result, the mapping between cells and channels rarely changes, a very desirable property for fingerprinting-based systems.

Every GSM cell has a special Broadcast Control Channel (BCCH) used to transmit, among other things, the identities of neighboring cells to be monitored by mobile stations for handover purposes. While GSM employs transmission power control both at the base station and the mobile device, the data on the BCCH is transmitted at a full and constant power. It is these BCCH channels that we use for fingerprinting.

Fingerprinting-based location techniques [2] take advantage of the fact that the strength of radio signals in the wireless bands used by GSM and 802.11 networks exhibits considerable spatial variability at the 1-10M level, but is consistent in time. In other words, a given radio source may be heard stronger or not at all a few meters away, and the signal strength from a given source at a given location is likely to be similar tomorrow and next week. In combination, this means that there is a radio profile that is feature-rich in space and reasonably consistent in time. Fingerprinting-based location techniques take advantage of this by capturing this radio profile for later reference.

Fingerprinting requires a preliminary *training phase* in which a radio map of the environment is constructed by taking a series of fingerprints in multiple locations. Each fingerprint is composed of several signal strength readings, one for each radio source in range (e.g., 802.11 access points, GSM base stations, FM radio [8] or TV stations). Once the training phase is complete, a client can estimate its location by performing a radio scan (or equivalently collecting a *testing fingerprint*) and feeding it to a *localization algorithm*, which estimates the client's location based on the similarity of the signal strength signatures between the testing and the training fingerprints. The similarity of signatures can be computed in a variety of ways, but it typically involves finding fingerprints in the training set that have the same radio sources with similar signal strengths. We elaborate on fingerprint matching in Section 4.

3 The SkyLoc Implementation

SkyLoc is a system that runs on a GSM mobile phone and determines the floor within a building on which a user is located. The system is implemented in C# and was tested on an AudioVox SMT 5600 phone shown in Figure 1. The phone runs Windows Mobile 2003 operating system. The SkyLoc system has two components: a data collection application called PlaceLogger and a fingerprint matching and visualization application called SkyLoc.

The PlaceLogger supports creating a hierarchical representation of places visited by a user and then collecting GSM measurements for these places (e.g., floors in a building). The screen shot of the PlaceLogger application is shown in Figure 2. The upper part of the PlaceLogger is a tree of places entered by a user. In our case, the tree has a depth of 2, having the names of buildings as root nodes and the floors as leaf nodes. The PlaceLogger allows scrolling through the nodes, adding new nodes, deleting nodes or selecting nodes. Once the user selects a node, she can press the `Enter Place` button to start the data collection process. To stop the data collection, the user presses the `Exit Place` button. The lower part of the screen shows the name of the place for which measurements are being collected and the number of measurements collected so far at the place.

The SkyLoc application shows the same hierarchical view of places recorded by the PlaceLogger. However, once loaded, SkyLoc continuously takes GSM measurements, matches them to the training measurements collected by the PlaceLogger, and presents the classification results to the user. The results are represented in a hierarchical manner. First, the probability of being at a leaf node is calculated

Figure 1. Audiovox SMT 5600 Smart Phone

Figure 2. PlaceLogger collection application

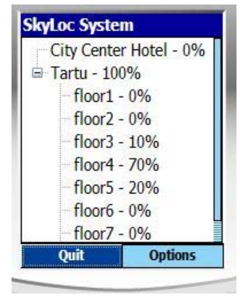


Figure 3. SkyLoc floor display application

and then these probabilities are propagated up the tree to the roots. The screen shot of the SkyLoc application is shown in Figure 3. The `Options` menu allows selecting the current window size and the algorithm for matching between the current measurement and the stored fingerprints.

Our preliminary experiences with the system are encouraging. Adding support for a building by collecting training measurements is quite easy and not very time consuming. Moreover, as we show in Section 5, the system has good accuracy.

Although we envision our system being eventually adapted, deployed and maintained by network operators or other 3rd party companies, we decided that the fastest way to get the system up and running today is to implement it as a stand alone application running on a mobile phone. In case of emergency, when a user dials 911 or the fall detection monitor instructs the phone to dial 911, the phone may calculate the current floor locally and transmit it to the emergency services. We analyze the run times and the memory requirements of our current system in Section 5.6.

4 Algorithms

In this section, we describe the classification algorithms we use to determine the floor in tall multi-floor buildings. All algorithms use fingerprinting as their underlying technique. That is, the algorithms are given two separate sets of training and testing data, which contain fingerprints collected on floors of a building. The algorithm then tries to determine the floor the testing fingerprints were taken on based on the fingerprints in the training set. To determine the floor given a testing fingerprint, our algorithm does the following: (1) Scans through all training fingerprints and calculates the Euclidean distance in signal space between the current testing fingerprint and all the training fingerprints; (2) Predicts the floor of the testing fingerprint as the floor of the training fingerprint with the smallest Euclidean distance.

4.1 Naive

The naive fingerprinting algorithm uses all available radio sources to compute the Euclidean distance between the training and testing fingerprints based on the assumption that the more radio sources are available the better the localization accuracy will be. For example, if a training fingerprint contains signal strength readings for 3 sources $\{R_1^{tr}, R_2^{tr}, R_3^{tr}\}$ and a testing fingerprint has signal strength readings for the same 3 sources $\{R_1^{tst}, R_2^{tst}, R_3^{tst}\}$ then the Euclidean distance between the two fingerprints using the naive fingerprinting approach will be calculated as:

$$\sqrt{(R_1^{tr} - R_1^{tst})^2 + (R_2^{tr} - R_2^{tst})^2 + (R_3^{tr} - R_3^{tst})^2} \quad (1)$$

In practice, however, some of the sources may be either too noisy, too stable across all floors, or simply inappropriate for fingerprinting-like algorithms [3] and including them in the calculation of the Euclidean distance may actually reduce localization accuracy. The interesting question is then how to identify these "bad" sources, so that only the remaining set of sources is used for matching in a given building. To perform the task of finding and eliminating "bad" sources, we next make use of a machine learning technique known as `feature selection`.

4.2 Feature Selection

We use feature selection techniques to identify "bad" sources and eliminate them from the Euclidean distance calculation. For example, if source 2 is identified as "bad", its readings can be ignored, and the Euclidean distance between the training and the testing fingerprint can be calculated as:

$$\sqrt{(R_1^{tr} - R_1^{tst})^2 + (R_3^{tr} - R_3^{tst})^2} \quad (2)$$

Formally, the feature selection problem is defined as follows. We assume that a building has N floors $F = \{F^1, ..., F^N\}$ and that a set of training data for each of the floors is available. Each sample in the training set consists of a feature vector $f = \{f_1, ..., f_K\}$. In our case, the feature vector is a vector of K signal strength readings corresponding to K available sources. We are interested in finding a ranking of the feature set $R = \{R_1, ..., R_K\}$ based on the usefulness of the features in classifying between floors correctly. Moreover, we are interested in finding a cut-off fingerprint τ for the ranked feature set such that adding features beyond τ does not significantly improve classification accuracy. In our case, the feature selection process will result in a subset of cells overheard during the training process. Localization algorithm is then going to use signal strength readings from these cells for fingerprint matching.

Including only the most relevant features in the matching process will not only improve the localization accuracy, but will also reduce memory consumption and significantly speed up the matching process. This is important because our goal is to run our localization system on real phones with limited memory and slow processors.

The simplest approach would be to try all possible combinations of features on the training data and pick the features that result in the best performance. However, such search is exponential in the number of features and therefore intractable. Therefore, existing feature selection algorithms revert to a heuristic search in the exponential feature state space. We have implemented and evaluated 3 techniques for feature selection: `Forward Selection`, `Backward Elimination` and a new `Per-Floor Feature Selection`.

4.2.1 Forward Selection and Backward Elimination

The Forward Selection algorithm starts with an empty set of features and in each step adds one additional feature to the set. The feature that is being added to the set is the feature whose addition results in the maximal increase in accuracy. In contrast, Backward Elimination algorithm starts with a set that contains all available features and then, at each step, removes features from the set. Once again, the feature that is being removed is the feature whose removal results in the largest increase in accuracy. Forward Selection and Backward Elimination are two variants of greedy feature selection that although do not necessarily select the best features, usually achieve good accuracy [3].

4.2.2 Per-Floor Feature Selection

The main idea behind the Per-Floor Feature Selection algorithm is that instead of selecting features that increase classification accuracy across all floors, the algorithm selects a different set of features that increase localization accuracy for a specific floor. In other words, for each floor F^i the algorithm produces a ranking of features $\{R^i_1, ..., R^i_K\}$ that can be used by a naive fingerprinting classifier to identify whether a testing fingerprint is on the floor i or not. Each per-floor classifier is assigned a weight based on its accuracy at identifying the floor correctly. At the end, the algorithm combines the classification results for each of the per-floor classifiers into the final classification decision.

5 Experiments

This section first describes our data collection process and then presents our evaluation results.

5.1 Data Collection

We collected fingerprints in the hallways of 3 buildings: (a) City Center Hotel, Washington D.C., USA; (b) University Hotel, Seattle, WA, USA; and (c) Tartu building, Toronto, ON, Canada. The buildings are shown in Figure 4. City Center Hotel is a 9-storey building, located in a quiet midtown residential area of Washington DC. University Hotel is a 12-storey building located in a midtown commercial area of Seattle. Finally, Tartu is a 16-storey building, located in downtown Toronto. Taking fingerprints in different cities and different urban environments allowed us to assess the robustness of SkyLoc in various environments. Moreover, we collected fingerprints during the day when people were present on the floors. Although we knew that this will result in less impressive localization results, we believe this approach better represents the true achievable accuracy.

Table 1 summarizes the number of fingerprints collected per-floor for each of the buildings[1]. The uneven number of fingerprints collected per floor is the result of us increasing the number of training and testing fingerprints collected with every new building in the hope of achieving even better localization results. Ironically, as we show in Section 5.5, the number of training fingerprints had little bearing on the localization accuracy.

We collected fingerprints for several available network operators simultaneously (using different phones), scanning the network every second. Once we started the data collection, we walked with an average speed of about 2m/s on each of the floors, collecting fingerprints. To investigate the effects of using different hardware for training and testing and the effects of separating the training and testing in time, we collected additional fingerprints in City Center Hotel two days after the initial fingerprints were collected and in University Hotel a month after the initial fingerprints were collected. In both cases, we collected fingerprints using different instances of the AudioVox phone.

[1]The buildings are sorted by height

(a) City Center Hotel, Washington D.C.

(b) University Hotel, Seattle, Washington

(c) Tartu, Toronto, Ontario

Figure 4. The tall multi-floor buildings where the data was collected.

	City Center Hotel	University Hotel	Tartu
`Number of floors`	9	12	16
`Fingerprints per floor`	110	30	130
`Training file size`	66KB	33KB	320KB

Table 1. Characteristics of the 3 buildings under study

5.2 Accuracy Evaluation

In this section, we evaluate how accurately the algorithms presented in Section 4 can differentiate between floors in tall multi-floor buildings. The algorithms are (a) `Naive` fingerprint matching algorithm that uses all available radio sources for matching; (b) `Forward Selection (FS)` algorithm; (c) `Backward Elimination (BE)` algorithm; and (d) `Per-Floor Feature Selection (PFFS)` algorithm. As described in Section 5.1, we collected separate traces for training and testing data in each of the 3 buildings and we used those traces as input to the 4 algorithms.

Figure 5 summarizes the accuracy with which the algorithms can correctly determine the current floor, be 1 floor off (predict the adjacent floor as the correct floor) and be 2 floors off. The PFFS algorithm performs the best, achieving 57% correct floor classification accuracy for both the City Center Hotel and the University Hotel buildings and 94% and 90% of correct classifications within 2 floors, respectively. The FS and BE algorithms achieve comparable accuracy with up to 52% of correct floor classifications and up to 96% of correct classifications within 2 floors.

The Naive fingerprinting algorithm achieves relatively low accuracy compared with the other feature selection algorithms (33% of correct floor classifications as opposed to 57% for PFFS). We found that the main reason for a low performance of the Naive algorithm is the apparent difference of the training and testing data on many floors of the buildings under study. Although the presence of people on the floors may have increased the discrepancy, we believe the main reason for the discrepancy lies in the way a mobile phone picks cells and channels to listen to. According to the GSM specification [5], the phone gets the list of neighboring cells to listen to from the associated cellular tower, which is not necessarily, but often, the tower with the strongest signal strength. The way the phone picks the associated tower depends on the strength and quality of the signal received from neighboring cells and on additional parameters, such as the time the phone was associated with the cell. Overall, this occasionally results in the phone picking different associated cells for the training and testing data on the same floor, which in turn results in low localization accuracy. Fortunately, even when the associated cells are different, there is still an overlapping in the cells picked by the two associated cells. It is these common cells that the feature selection algorithms use to achieve higher localization results.

One might expect to see better localization accuracy for lower buildings because less floors means less chance of getting the current floor wrong. For example, in a building with only 3 floors even an algorithm that guesses the current floor at random will be correct roughly 33% of the time. However, the results show that the accuracy in higher buildings has not decreased significantly. For example, PFFS, our best performing algorithm, achieved 93%

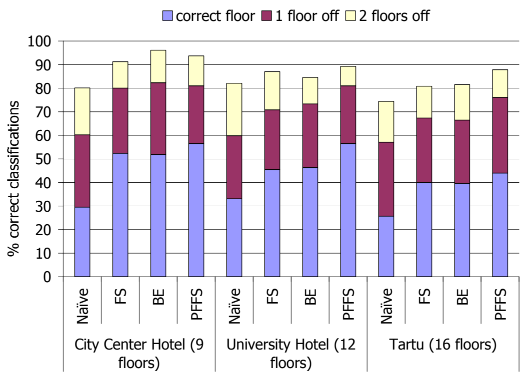


Figure 5. Accuracy results across all building

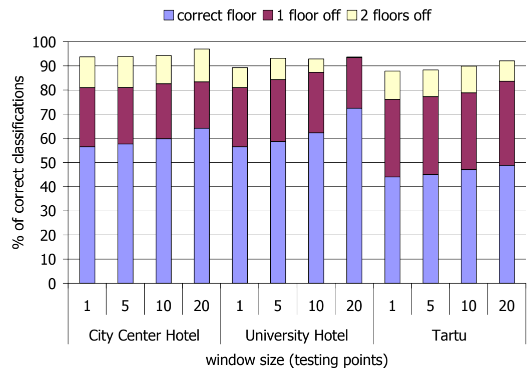


Figure 6. The effect of windowing on top of the PFFS algorithm

accuracy within 2 floor in City Center Hotel, 89% in University Hotel and 87% in Tartu. As the following analysis shows this is mainly due to the fact that when the classifier is wrong, it is usually wrong within 1 or 2 floors and therefore increasing the number of floors may not necessarily affect accuracy. For instance, the radio environment on a 2nd floor might be similar to the one on the 3rd or the 4th floor, but it is as drastically different from the one on the 10th floor as it is from the one on the 20th.

5.3 Windowing

The previous section showed localization results for testing fingerprints classified independently of one another. In practice, the classification decision need not necessarily be made on a single testing fingerprint, but may be made based on a stream of testing fingerprints.

We implemented a simple algorithm that makes the classification decision based on a fixed-size sliding window of testing measurements. For example, if the window size is 10, the classification decision is based on the current measurement and the nine preceding measurements. The windowing algorithm first classifies each measurement in the window individually, and then selects the current floor as the most frequently appearing floor among the individual classifications.

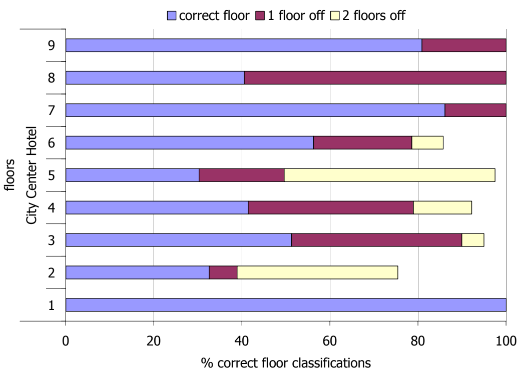
Figure 6 shows the effect of basing the classification decision on a stream of testing fingerprints with a fixed window size. The figure plots the classification accuracy for the PFFS algorithm when the number of testing fingerprints in the window varies from 1 to 20. For the City Center Hotel, 97% of the testing fingerprints were classified within 2 floors; whereas, for the University Hotel, the accuracy of correctly identifying the current floor has increased from 56% to 73%, reaching 94% of classifications within 1 floor.

Although in areas with large number of misclassifications, windowing does not help much, it does help to remove outliers when the overall performance is good, and we believe it should be used by localization systems. Our prototype SkyLoc system, described in Section 3, uses the windowing technique. In the future, we plan to look at sequential filtering techniques [15] to improve the accuracy of the system.

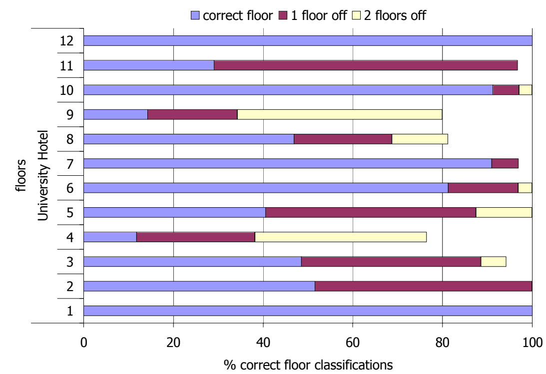
5.4 Per-Floor Analysis

Figures 7(a), 7(b) and 7(c) show the breakdown of the per-floor classification accuracy of the PFFS algorithm for City Center Hotel, University Hotel and Tartu building, respectively. For example, the full bar for the 1st floor in the City Center Hotel means that all testing fingerprints collected on this floor were classified correctly. The figures reveal that the bad performance on some of the floors pulls the overall performance down. For example, no testing fingerprints were classified correctly on the 3rd floor in the Tartu building. We believe that identifying and improving accuracy on the "bad" floors will drive our results higher up, and we are planning to investigate this matter in the near future.

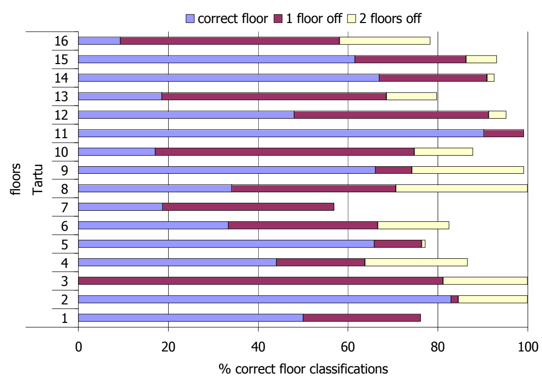
We expected the middle floors to have higher errors than the top and bottom floors because these floors have more "competition", or a larger number of adjacent floors. Although it is the case in City Center Hotel and University Hotel, it is not the case in the Tartu building. Also, we expected higher floors to have larger localization errors because of the possible lower coverage. This seems to be not the case. Our data showed that there is no difference in average signal strength between the floors in the building across all 3 buildings.



(a) City Center Hotel



(b) University Hotel



(c) Tartu

Figure 7. The breakdown of classification accuracy per floor for the PFFS algorithm.

5.5 Sensitivity Analysis

In this section, we quantify the sensitivity of the classification accuracy to different network operators, different hardware, separation of training and testing in time and the number of training fingerprints collected per floor.

Figure 8 shows the localization results for the University Hotel for different network operators. The results suggest that our system works across different network providers, as there seems to be no significant difference in terms of achievable accuracy between different network operators. The results for City Center Hotel and Tartu buildings (not included) show a similar trend.

Figure 9 shows that collecting the training and testing fingerprints with different phones does not significantly affect localization accuracy. In the University Hotel, the percentage of correct floor classifications has reduced from 57% to 56% and for the Tartu building it has reduced from 44% to 35%. In both buildings, the percentage of correct classifications within 2 floors has slipped 2%. Although we have experimented with different phones we used the same phone models. In the future, we plan to investigate how the effect of taking fingerprints with a different phone model impacts the overall localization accuracy.

Figure 10 shows the effect of taking the training and testing fingerprints 2 days and a month apart for the City Center Hotel and the University Hotel. The results show that taking testing fingerprints a few days or even a month apart does not significantly affect localization accuracy. For the City Center Hotel, the percent of correct floor classifications within 2 floors did not change, and the number of correct floor classifications has reduced from 57% to 54%. For the Tartu building, the performance actually increased, rising to 48% correct floor classifications from 44%.

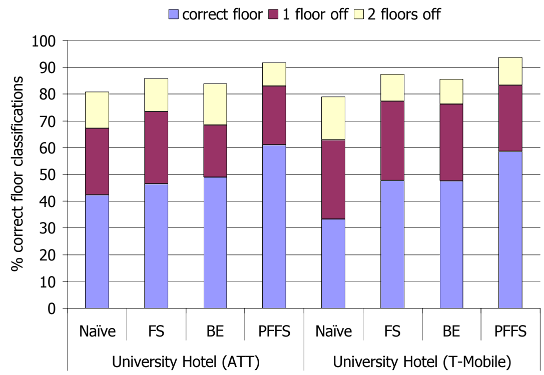


Figure 8. The effect of varying network operators

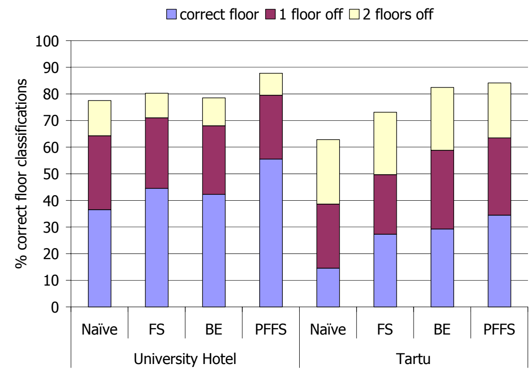


Figure 9. The effect of collecting testing and training measurements with different phones

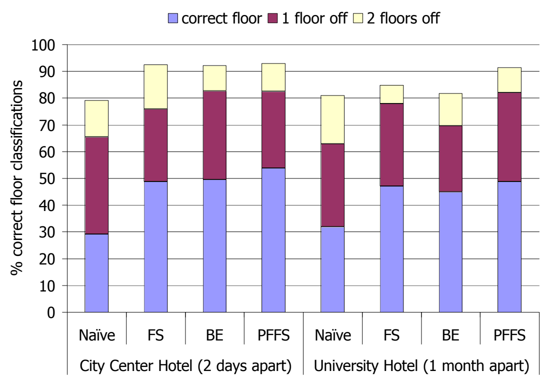


Figure 10. The effect of taking the training and testing fingerprints 2 days and a month apart

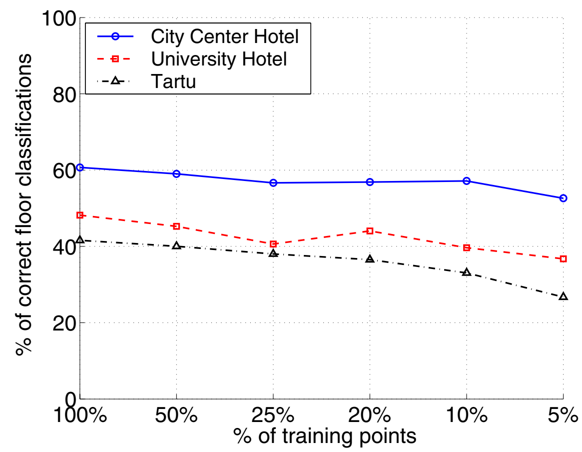


Figure 11. The effect of reducing the number of training fingerprints

Figure 11 shows the effect of reducing the number of training fingerprints collected per-floor for each of the 3 buildings. The figure plots the percentage of correct floor classifications as a function of the percentage of training fingerprints used. For example, 58% of the testing points were classified correctly in the City Center Hotel with both one forth and one tenth of the originally collected training points. Surprisingly, the reduction in accuracy between 100% of training fingerprints to only 5% is small across all buildings. This is a thrilling result because it means that only a small number of training fingerprints needs to be collected per-floor, or in other words we could train any of the buildings under study in less than 30 minutes and still achieve good localization results.

5.6 Preliminary Performance Evaluation

In this section, we present our preliminary performance evaluation of the SkyLoc system in terms of memory and storage footprint and localization run times.

The amount of training data that needs to be stored on the phone depends on the building size. The taller the building and the larger the floor size, the larger the training file. Our current prototype stores the data in a raw text format without performing any storage optimizations. The training file sizes are summarized in Table 1. It follows that with the current flash card sizes of 1GB it is possible to store training files of more than 7000 buldings on a single card. Moreover, the training files may be stored in an archive file (e.g., zip) most of the time and extracted only on demand. This optimization reduces the storage requirement on the phone by an order of magnitude (archiving the 320KB training file from the Tartu building produces a 30KB zip file). Note that instead of storing all fingerprint maps on the phone, the phone may be able to simply download them upon entering a building.

The SkyLoc application takes about 200KB storage space including all the necessary libraries. When loaded it takes about 1600KB of memory, plus any additional memory needed for the training data. So, the SkyLoc application and the Tartu building training file take about 2MB of memory out of the 32MB available on our AudioVox SMT 5600 phone.

Next, we measured the scalability of SkyLoc in terms of the time it takes to locate a single testing fingerprint on AudioVox's 200Mhz ARM processor. Determining the location of a fingerprint requires matching the fingerprint against the current training set. Note that in order to locate a fingerprint there is no need to match the fingerprint to all fingerprints stored on a phone, but only to a set of relevant fingerprints. One approach that we found to work well in practice is matching only against training fingerprints that have at least one cell ID in common with the current testing fingerprint.

We conducted a series of experiments, each time varying the training file size and measuring the time it takes to locate a single testing fingerprint. On average, it takes 0.002 seconds to match a single testing fingerprint to a single training fingerprint or equivalently the phone can match a testing fingerprint to 500 training fingerprints a second. For instance, in the University Hotel, it takes about 0.72 seconds to localize a fingerprint. We are planning to develop faster fingerprint matching techniques in the future.

5.7 Discussion and Recommendations

Should floor identification be added to the E911/E112 specifications, we recommend regulatory bodies start with the requirement of "within 2 floors of the actual floor number 95% of the time." We have demonstrated that the 2 floor-95% goal is achievable in software on mobile phones and thus it represents a good starting point for any discussions of extending regulations of the third dimension. While a lower error margin might be necessary for some E911 scenarios, we believe regulation works best if it starts with what is possible and then evaluates if it is sufficient.

The largest barrier to wide-scale adoption of our approach is probably the requirement to gather training data for each building. However, we believe such calibration could be made a part of the regulated zoning procedures for large buildings and is probably low overhead compared to the many stringent building codes and maintenance procedures already in place for a multi-floor building like elevator maintenance and emergency exit lighting and signage. The fact that calibration maps seem capable of being transferred between devices without significantly impacting accuracy also supports this deployment model.

6 Related Work

Indoor location systems have successfully employed a variety of technologies including infrared [7, 20] and ultrasound [14, 16]. While these system can achieve accuracies of a few centimeters, their requirement for custom infrastructure has hindered their wide-scale deployment. On the other hand, fingerprinting-based localization systems provide accurate indoor localization by making use of the existing wireless infrastructure, obviating the need for infrastructure investment and greatly increasing the possible area in which the system will work.

Most of the research on fingerprinting-based localization has focused on the use of 802.11 radio sources [2, 18, 10, 17, 6]. GSM-based fingerprinting, however, has the main advantage that it works with existing mobile phones, whereas 802.11 is available only on high-end PDAs and laptops. Moreover, due to higher coverage, GSM works in more places than 802.11 does.

A number of systems have used GSM to estimate the location of mobile clients. Place Lab [12, 4] system employed a map built using war-driving software and a simple radio model to estimate a mobile phone's location with 100-150 meter accuracy. Laasonen *et al.* used the transition between GSM cell towers to build a graph representing the places a user goes [9]. Laitinen *et al.* [11] used GSM-based fingerprinting for outdoor localization. They have collected sparse fingerprints from the 6-strongest cells, achieving 67^{th} percentile accuracy of $44m$.

In our previous work, we demonstrated that GSM-based indoor localization is feasible [13]. We collected measurements with a specialized Sony Ericsson GSM modem connected to a laptop and using a naive fingerprinting algorithm we reported within-floor median error of 5 meters and showed that it is possible to differentiate between a couple of floors in a building. In contrast, this paper reports on our experiences with SkyLoc, a fully functioning system that runs on a commodity mobile phone and identifies the current floor in tall multi-floor buildings. This paper also pioneers the use of feature selection techniques for the domain of localization and shows that it can significantly improve localization accuracy. Finally, this paper investigates practical deployment issues, such as using different hardware for training and testing, using different network operators and collecting testing measurements up to one month after the training measurements were collected.

7 Conclusions and Future Work

We presented SkyLoc, a localization system that identifies the current floor of a mobile phone user in tall multi-floor buildings. Knowing the floor in a tall building significantly reduces the area that emergency service personnel have to canvas to locate the individuals asking for help. We evaluated our system in three multi-floor buildings located in Washington DC, Seattle, and Toronto. Our system identifies the floor correctly in up to 73% of the cases and is within 2 floors in 97% of the cases. Our system is robust; it works for different network operators, when the training and testing sets were collected with different hardware and up to one month apart. We also showed that feature selection techniques that select a subset of highly relevant radio sources for fingerprint matching nearly doubled the localization accuracy of our system.

In future direct extensions to this work, we plan to test the system in higher buildings, compare with technologies like pressure sensors that require hardware modifications, and conduct a user study to test satisfaction with our prototype in the context of usage scenarios like building rescue. We also plan to improve our system in terms of memory and CPU consumption. Finally, we plan to test the applicability of feature selection techniques to within-floor localization as well.

References

[1] Empire State Building.

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

[3] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. In *Journal on Artificial Intelligence*, 1997.

[4] M. Y. Chen, T. Sohn, D. Chmelev, D. H. J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical metropolitan-scale positioning for gsm phones. In *Proceedings of the Eighth International Conference on Ubiquitous Computing (UbiComp)*, Irvine, California, Sept. 2006.

[5] J. Eberspacher, H.-J. Vogel, and C. Bettstetter. GSM switching, services and protocols, 2001.

[6] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Philadelphia, PA, Sept. 2004.

[7] A. Hopper, A. Harter, and T. Blackie. The active badge system. In *Proc. of INTERCHI-93*, pages 533–534, Amsterdam, The Netherlands, 1993.

[8] J. Krumm, G. Cermak, and E. Horvitz. RightSPOT: A Novel Sense of Location for Smart Personal Objects. In *Proceedings of Ubicomp*, 2003.

[9] K. Laasonen, M. Raento, and H. Toivonen. Adaptive on-device location recognition. In *Proceedings of the Second International Conference on Pervasive Computing*, pages 287–304. Springer-Verlag, 2004.

[10] A. Ladd, K. Bekris, G. Marceau, A. Rudys, L. Kavraki, and D. Wallach. Robotics-based location sensing using wireless ethernet. In *Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2002.

[11] H. Laitinen, J. Lahteenmaki, and T. Nordstrom. Database correlation method for GSM location. In *Proceedings of the 53rd IEEE Vehicular Technology Conference*, Rhodes, Greece, May 2001.

[12] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device Positioning Using Radio Beacons in the Wild. In *Proceedings of the Third Annual Conference on Pervasive Computing*, 2005.

[13] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate gsm indoor localization. In *Proceedings of the Seventh International Conference on Ubiquitous Computing (UbiComp)*, Tokio, Japan, Sept. 2005.

[14] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Mobile Computing and Networking*, pages 32–43, 2000.

[15] V. Seshadri, G. V. Zaruba, and M. Huber. A bayesian sampling approach to in-door localization of wireless devices using received signal strength indication. In *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, 2005.

[16] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. In *IEEE Personnel Communications, 4(5)*, pages 42–47, Oct. 1997.

[17] M. Youssef, A. Agrawala, and U. Shankar. WLAN Location Determination via Clustering and Probability Distributions. In *IEEE PerCom 2003*, March 2003.

[18] Ekahau, http://www.ekahau.com.

[19] Federal Communications Commission Report and Order 96-264: Revision of the commission's rules to ensure compatibility with Enhanced 911 emergency calling systems, July 1996, http://www.fcc.gov/bureaus/wireless /orders/1996/fcc96264.txt.

[20] Versus Technologies, http://www.versustech.com.