Calibree*: Calibration-free Localization using Relative Distance Estimations

Alex Varshavsky◇, Denis Pankratov◇, John Krumm† and Eyal de Lara◇

◇Department of Computer Science, University of Toronto
{walex, delara}@cs.toronto.edu, denis.pankratov@gmail.com
†Microsoft Research
jckrumm@microsoft.com

Abstract. Existing localization algorithms, such as centroid or fingerprinting, compute the location of a mobile device based on measurements of signal strengths from radio base stations. Unfortunately, these algorithms require tedious and expensive off-line calibration in the target deployment area before they can be used for localization. In this paper, we present Calibree, a novel localization algorithm that does not require off-line calibration. The algorithm starts by computing relative distances between pairs of mobile phones based on signatures of their radio environment. It then combines these distances with the known locations of a small number of GPS-equipped phones to estimate absolute locations of all phones, effectively spreading location measurements from phones with GPS to those without. Our evaluation results show that Calibree performs better than the conventional centroid algorithm and only slightly worse than fingerprinting, without requiring off-line calibration. Moreover, when no phones report their absolute locations, Calibree can be used to estimate relative distances between phones.

1 Introduction

The most widespread localization technology available today is the Global Positioning System (GPS) [7, 17]. Although accurate in open environments, GPS does not work well indoors, in urban canyons, or in similar areas with a limited view of the sky. In addition, GPS is installed in only a small portion of the mobile phones in use today. ABI research reported that the number of mobile phone subscribers with GPS equipped devices constituted only 0.5% of the total number of subscribers in 2006, but it estimates that this number will grow to 9% by 2011 [19]. As a result, several alternative localization algorithms have been proposed, including *centroid* and *fingerprinting* [18, 3]. The main drawback of these algorithms is that they require off-line calibration in the target deployment area before they can be used for localization.

Calibrating a fingerprinting system is tedious and expensive, involving physically sampling signal strengths at many locations. For instance, a recent effort

* [kaw-**li**-bri] is a Russian word for hummingbird

by Intel Research Seattle to sample the GSM radio environment in the Seattle metropolitan area took 3 months to complete, costing about US$ 30,000 [3]. If the locations of cell towers are not readily available, centroid methods need the same type of calibration to compute the unknown locations of the cell towers. Another disadvantage of relying on physical sampling is that it gives only a snapshot of signal strengths at a particular time. Cell towers can be added, removed, blocked, renamed, or moved, partially invalidating an expensive calibration run.

In this paper, we present a novel localization technique, called Calibree, that requires no such calibration or maintenance. Calibree takes advantage of a small number of phones with known locations to determine locations of a larger set of phones. Calibree has two stages. In the first stage, Calibree computes relative distances between mobile phones that detect at least one GSM cell tower in common by comparing their GSM signatures. We define a *GSM signature* as a set of GSM cell towers that a phone detects and the signal strengths at which the phone hears these towers. To estimate relative distances, Calibree computes a regression formula in real time, based on a snapshot of GSM signatures and absolute locations from a small number of phones. In the case when no phones report their absolute locations, Calibree reverts back to using the last computed regression formula.

In the second stage, Calibree combines the pairwise distance estimations into a graph, in which nodes represent mobile phones and weighted edges represent likely distances between the phones. If a small number of mobile phones are able to report their absolute positions (e.g., through GPS), these phones are anchored at their known locations. Calibree then computes likely locations for all other phones by modelling the graph as a constraint problem and estimating mobile phone positions in order to minimize overall constraint violations using a mass-spring minimization method [4]. If no phones report their absolute locations, Calibree cannot compute absolute locations, but it can estimate relative distances between pairs of mobile phones. This is useful for gaming and social-mobile applications [16], where knowing a relative distance to another mobile phone is sufficient.

When computing relative distances between phones, Calibree takes into account only the ranked list of cell towers that phones hear sorted by the signal strength. Not using the actual signal strength as part of the relative distance estimation makes Calibree independent of the particular phone model it is running on, as the relative ranking of cell towers has been shown to be independent of the specific phone used [8].

We evaluated Calibree in the University of Toronto campus, located in downtown Toronto, Canada and a quiet residential neighborhood, located on the outskirts of Toronto. The results show that with only a small number of phones having GPS, Calibree outperforms the centroid algorithm and is comparable to the fingerprinting algorithm, achieving up to 147m median error. This result is very promising because Calibree achieves similar accuracy to existing localization algorithms, without requiring off-line calibration. Moreover, Calibree continues to work well and outperforms both the centroid and fingerprinting algorithms

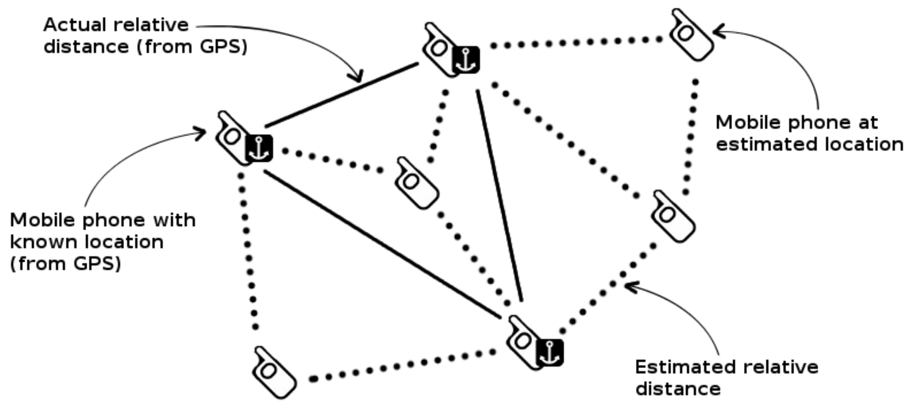

Fig. 1. Absolute positioning of mobile phones using Calibree.

when tested on measurements collected inside 20 buildings scattered around the University of Toronto campus. This result shows that Calibree is effective at propagating the absolute location information from phones located outdoors to the phones located indoors. Finally, even when no absolute locations are known, Calibree estimates relative distances between phones more accurately than the centroid algorithm and only slightly worse than the fingerprinting algorithm.

The rest of this paper is organized as follows. We describe Calibree, centroid and fingerprinting algorithms in detail in Section 2 and present our evaluation results in Section 3. Section 4 discusses the differences between Calibree and related research efforts. Finally, we present our conclusions in Section 5.

2 Localization Algorithms

In this section, we describe Calibree, the fingerprinting and the centroid algorithms. We compare performance of these algorithms in Section 3.2.

2.1 Calibree

Calibree is a localization algorithm that estimates locations of GSM phones based on a snapshot of all phones' GSM signatures and absolute locations of a small number of these phones. If no absolute locations are known, Calibree can be used to estimate relative distances between any two mobile phones.

Figure 1 demonstrates how Calibree solves the problem of estimating absolute phone locations. The anchored mobile phones obtain their locations through GPS and feed these locations to Calibree, which uses them to estimate locations of phones without GPS. Calibree has two stages. In the first stage, Calibree computes a regression formula based on GSM signatures and known locations of

GPS-equipped phones, and then uses this formula to estimate relative distances between phones without GPS that overhear at least one common cell tower. In the second stage, Calibree uses a graph-based algorithm to estimate locations of phones without GPS. We next describe these two stages in detail.

Estimating Relative Distances In the first stage, Calibree computes relative distances between pairs of mobile phones based on their GSM signatures. Recall that a GSM signature consists of a set of GSM cell towers that a phone detects and the signal strengths at which it hears these towers. Computing distances based on pairs of GSM signatures involves extracting features from the signatures and feeding the extracted values into a pre-generated formula.

We explored several possible features, including: number of common cells, number of cells not in common, Spearman coefficient, Euclidean distance in signal space, ratio of the number of cells in common to the total number of cells and a boolean variable indicating whether the phones hear the same serving cell.

To identify which features to use, we experimented with a number of different combinations of features, each time recording the median error of relative distance predictions using a given set of features. We found three features that both achieve good accuracy and that are insensitive to a particular phone model. The features are:

Common cells: The number of cell towers that are common to the two GSM signatures.

Uncommon cells: The number of cell towers that are not common to the two GSM signatures.

Spearman coefficient: The Spearman coefficient [12] between rankings of common cell towers by signal strengths.

These features use information about cell towers and relative signal strengths only. Previous studies have shown [8] that these parameters are cell phone model agnostic.

Given GPS coordinates and GSM signatures from a number of phones, Calibree generates a formula for predicting pairwise distances between phones by applying the multiple regression method to the features extracted from the GSM signatures and the distances computed from the GPS coordinates. We experimented with polynomials of different degrees. The evaluation procedure was the same as for selecting the features. We observed that while polynomials of high degrees often suffer from overfitting, a degree two polynomial gives consistently good performance and results in more accurate predictions than a linear function.

Figure 2 shows the general form of the formula that Calibree uses for relative pairwise distance estimations. In this formula, x_i stands for the value of the ith feature. The constants a_{ij}, b_i, and d are fitted from the GPS coordinates and GSM signatures of phones with GPS. These constants are recomputed each time the graph is built, which makes Calibree implicitly adaptive to changes in cell tower configurations. Once Calibree computes the constants a_{ij}, b_i and d, it is ready to estimate relative distances between pairs of mobile phones.

$$Distance(x_1, x_2, x_3) = \sum_{i=1}^{3} \sum_{j=i}^{3} a_{ij} x_i x_j + \sum_{i=1}^{3} b_i x_i + d$$

Fig. 2. The general form of the regression formula for relative pairwise distance estimates.

Note that when two mobile phones overhear no common cells, the Spearman coefficient is not defined and the Common cells feature always evaluates to 0. Therefore, Calibree does not estimate relative distance between phones that detect no common cells, but rather assumes that these phones are located far away and lets the graph-based algorithm deal with these cases.

Estimating Absolute Locations To compute absolute positions of mobile phones, Calibree uses the relative distance estimates obtained from the previous step as well as the absolute positions of a small number of mobile phones with GPS. With this data, Calibree builds a graph, in which nodes represent mobile phones and weighted edges represent estimated distances between mobile phones.

The graph models a geographic coordinate system: each node has a corresponding latitude/longitude coordinate and the distance between nodes is calculated using the Haversine formula [15]. Calibree initializes the coordinates with the actual mobile phone positions for the GPS-equipped phones and with random values for other phones. This initial placement of nodes results in a discrepancy between the weights on the edges and the distances between nodes in the graph. The goal of Calibree is to find a placement of nodes such that the overall discrepancy is minimized. Calibree doesn't change the coordinates of mobile phones with GPS during its runtime since these are known to be very close to correct.

To better understand the problem, imagine that every pair of nodes with an edge between them is connected using a spring with a relaxed length equal to the estimated relative distance between the corresponding mobile phones. As Calibree adjusts the locations of the nodes, it recomputes the lengths of the springs based on the Haversine formula, which calculates the distance between two latitude/longitude coordinates. Calibree's goal is then to find a placement of nodes such that the lengths of the springs are as close as possible to the relaxed lengths of the springs. More formally, if we denote two nodes as x and y, the relaxed spring length as r_{xy} and the current spring length as c_{xy}, Calibree needs to find a node placement that minimizes the error function:

$$Err = \sum_{x} \sum_{y} |r_{xy} - c_{xy}| \tag{1}$$

Calibree minimizes this function by iteratively refining node coordinates. Each spring exerts a force on the pair of the nodes that it connects. The magnitude of the force is taken to be the difference between the current and the relaxed lengths of the spring. We denote the direction of the force from node y to node x by a unit vector $unit(x, y)$. The force on node x from node y is then:

$$F_{xy} = |r_{xy} - c_{xy}| \times unit(x, y)$$

The net force on node x from all other connected nodes is just the sum of all individual forces:

$$F_x = \sum_{y \neq x} F_{xy}$$

Once Calibree calculates the net force, it is then ready to move the node in the direction of that force. However, applying the full force on the node would result in oscillations in node positions because many of the current node positions are incorrect. Therefore, Calibree applies only a portion of the original force on the node. The portion of the applied force is controlled by a parameter δ. Finding the right value for δ is important, because large values will result in large oscillations of node positions and consequently Calibree may not reach equilibrium altogether, while small values will result in slow convergence. We experimented with different values of δ and found that a value of 1.0×10^{-8} provides a good compromise between the speed of convergence and no oscillations.

Formally, if we let $coor_x$ be the vector coordinate of node x, the new coordinate after applying the force is:

$$coor_x = coor_x + \delta \times F_x$$

Calibree stops minimizing the error function when the refinement of node coordinates results in a negligible change in the total error Err, controlled by another parameter $tolerance$. For smaller values of δ, Calibree needs to use smaller $tolerance$ values, or otherwise Calibree might terminate prematurely.

A special case occurs when a pair of nodes has no relative distance prediction. This is a result of two mobile phones detecting no common cell towers. In this situation Calibree assumes that the two phones are more than a certain threshold distance away. To compute this threshold we looked at the cumulative distribution function of distances between mobile phones that detect no cells in common and picked the 25^{th} percentile, which turned out to be about $500m$. Pairs of nodes that correspond to mobile phones that detect no common cells therefore have a special spring connecting them that exerts force only if the actual distance between nodes is smaller than $500m$. This is because Calibree does not know how far the two nodes are, but it does know that the nodes are likely to be at least $500m$ apart. Formally, the force between nodes x and y that detect no common cells is calculated as:

$$F_{xy} = \begin{cases} |500 - c_{xy}| \times unit(x, y) & \text{if } c_{xy} < 500 \\ 0 & \text{otherwise} \end{cases}$$

It is possible for Calibree to end up in a local minimum equilibrium, in which node position refinements are small, but the overall error Err is still large. We

```
▷ input: graph G = (V,E) and relative distance estimation function r
▷ output: coordinates of nodes in V
CALIBREE-ABSOLUTE(G, r)
 1  Err_cur ← ∞
 2  repeat
 3        Err_prev ← Err_cur
 4        for each x in V
 5        do
 6           if x has a fixed location
 7              then continue
 8           F ← 0
 9           for each y ≠ x in V
10           do
11              if (x, y) ∈ E
12                 then F ← F + |r_xy − c_xy| × unit(x, y)
13                 else F ← F + max((500 − c_xy) × unit(x, y), 0)
14           coor_x ← coor_x + δ × F
15        Err_cur ← Error(G)
16    until |Err_prev − Err_cur| > tolerance
```

Fig. 3. The pseudo-code of a graph-based stage of Calibree.

used an optimization that proved to work consistently well in pushing Calibree towards global equilibrium. With our optimization, when Calibree achieves a local equilibrium state, instead of terminating, it repositions a randomly chosen non-fixed node in the average of the current locations of all its connected nodes. This is repeated several times with different mobile phones.

Figure 3 shows the pseudo-code for the graph-based stage of Calibree. We leave out the above optimization for simplicity. The algorithm receives as input two parameters: (a) G, a graph with a set of nodes V and a set of edges E and (b) r, a pre-computed function of relative pairwise distances between nodes. The algorithm computes a new set of coordinates that minimizes the total error as defined in Equation 1. The outermost loop runs until the error difference between two successive iterations is less than the *tolerance* value. The inner loop keeps calculating the net force exerted on each node and updating their coordinates in the direction of the force. *Error* function on line 15 calculates the current total error of graph G.

CALIBREE-ABSOLUTE, shown in Figure 3, works correctly even when no absolute phone locations are known. In this case, although the orientation of the nodes is arbitrary, the algorithm may be used to estimate relative distances between any two nodes in the graph.

2.2 Fingerprinting

The fingerprinting algorithm [1] relies on the fact that signal strengths observed by mobile phones exhibit temporal stability and spatial variability. In other words, a given cell tower may be heard stronger or not at all a few meters away, while at the same location the observed signal strength is likely to be similar tomorrow and next week.

The fingerprinting algorithm requires a calibration phase, in which a mobile phone moves through the target environment, recording the strengths of signals emanating from radio sources (e.g., GSM cell towers). At the end of calibration, the fingerprinting algorithm creates a mapping from radio measurements to locations where these measurements were observed. Since the fingerprinting algorithm does not model radio propagation, a fairly dense grid of radio scans needs to be collected to achieve good accuracy. The original RADAR experiments, for example, collected measurements of WiFi signal strengths about a meter apart [1]. In our implementation we collected measurements every two meters on average.

Once the calibration phase is complete, a mobile phone can estimate its location by performing a radio scan and feeding it into the fingerprinting algorithm, which estimates the phone's location based on the similarity between the phone's radio scan and the measurements recorded during the calibration. The similarity of signatures can be computed in a variety of ways, but it is common to use the Euclidean distance in signal space [11, 3]. The fingerprinting algorithm then estimates the location of a mobile phone to be the location of the measurement in the mapping with the smallest Euclidean distance in signal space to the radio scan. If a cell tower is not present in one of the measurements, we substitute its signal strength with the minimal signal strength found in this measurement.

2.3 Centroid

To estimate phone locations, the centroid algorithm [3] needs to know the locations of GSM cell towers. However, since this information is typically kept confidential by service providers and it is not available to third parties, the centroid algorithm has to estimate positions of cell towers by reverting to the same physical sampling of the radio environment employed by the fingerprinting algorithm. Once the physical sampling is complete, the centroid algorithm can estimate locations of cell towers in the environment by positioning a cell tower in a location where the signal strength from that cell tower was observed the strongest. In our experiments, we used the same calibration data for the centroid algorithm as we did for the fingerprinting algorithm.

Once the positions of cell towers are known and a mobile phone performs a radio scan, the centroid algorithm computes the location of the mobile phone as an average of locations of cell towers that appear in the radio scan. Typically, giving a higher weight during averaging to cell towers with stronger signal strength yields better localization accuracy.

3 Evaluation

In this section, we describe our data collection process, present our experimental results and then discuss the usage model of Calibree.

3.1 Data Collection

To evaluate the accuracy of our localization algorithms, we collected GSM measurements on the streets of the University of Toronto campus. The university covers an area of approximately $1km^2$ and it is located in the downtown core of Toronto, Canada. We gathered additional traces covering a residential area of similar size located on the outskirts of the city. To test the localization accuracy of Calibree indoors, we also collected several measurements inside 20 university buildings.

To collect the measurements, we used a Pocket PC T-Mobile MDA and an AudioVox SMT 5600 phone, connected to a Holux GPSlim236 GPS via Bluetooth. Both the PDA and the phone ran Intel's POLS [3] data collection software, which gives access to identities and signal strengths of up to 8 GSM cells. We walked through the target area at a speed of 2 meters per second and sampled the radio environment and GPS unit at a rate of one sample per second.

We collected two sets of traces for each area, to be able to train and test our algorithms on different traces. Full traces for the downtown and the residential areas contain about 6000 and 5000 measurements, respectively. Since different network operators use different cell towers, Calibree, fingerprinting and centroid algorithms work only when training and testing traces use the same operator. To support multiple operators, separate traces need to be collected for each operator. We collected all traces using Rogers, a single GSM network operator available in Toronto, Canada.

3.2 Experimental Results

In this section, we compare accuracy with which Calibree, the fingerprinting algorithm and the centroid algorithm estimate absolute phone positions and relative distances between phones.

Absolute Positioning We trained Calibree "on-line" by randomly picking a number of points from the testing trace, simulating mobile phones with GPS, and using their GSM signatures and known locations to train the regression formula. To test the three algorithms, we randomly picked 50 points from a testing trace, simulating 50 mobile phones, estimated their absolute locations using each of the three algorithms and calculated the localization error using the actual phone locations from the trace. For example, if Calibree picks 25 points to train the regression formula and another 50 points for testing, it will use all 75 points to construct and solve the mass-spring graph. The 25 points that Calibree uses for creating the regression formula do not change their location during the runtime of Calibree and are not included in the calculation of accuracy. To reduce random effects and smooth the graphs, we repeated this procedure 40 times for each experiment. The following experiments use downtown traces unless otherwise specified.

Figure 4 plots the 50th and 95th percentile error of Calibree as a function of the number of phones with GPS. The results show that although the localization

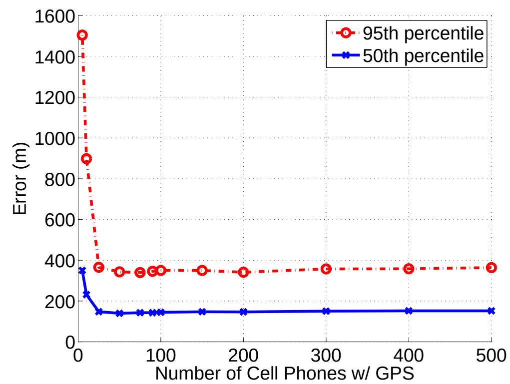

Fig. 4. The effect of the number of phones with GPS on the 50th and 95th percentile localization error of Calibree.

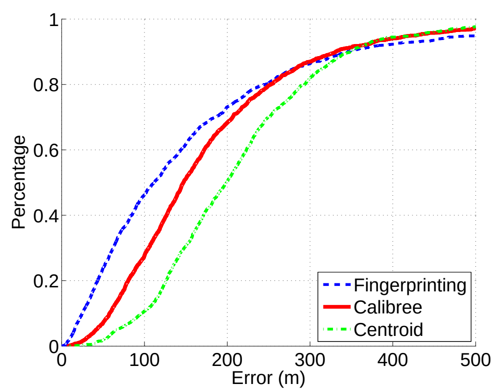

Fig. 5. Cumulative distribution function of absolute localization error of Calibree, the fingerprinting and centroid algorithms, evaluated in the downtown area.

accuracy generally improves with the larger number of phones with GPS, the error levels off at 25 GPS-equipped phones. Note that 25 phones with GPS in the area of $1km^2$ is not too many, given that the average population density of Toronto including residential areas is 4000 $people/km^2$ and it is much higher in the downtown area.

Figure 5 and Figure 6 show the cumulative distribution function (CDF) of absolute localization error for the fingerprinting algorithm, the centroid algorithm and Calibree with 25 phones having GPS, evaluated in the downtown and the residential areas, respectively. In the downtown area, the fingerprinting and centroid algorithms achieved comparable accuracy to previously reported implementations, $112m$ and $200m$ median error, respectively. Calibree achieved $147m$ median error, which is better than the centroid algorithm and slightly worse than the fingerprinting algorithm. In the residential area, the errors are

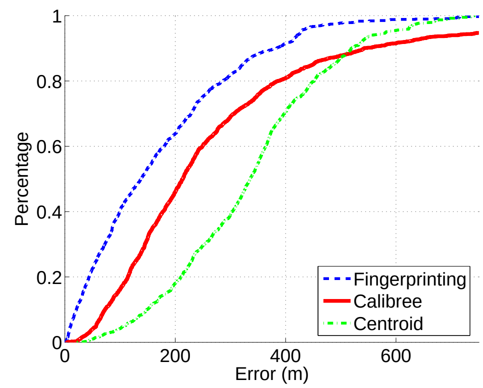

Fig. 6. Cumulative distribution function of absolute localization error of Calibree, the fingerprinting and centroid algorithms, evaluated in the residential area.

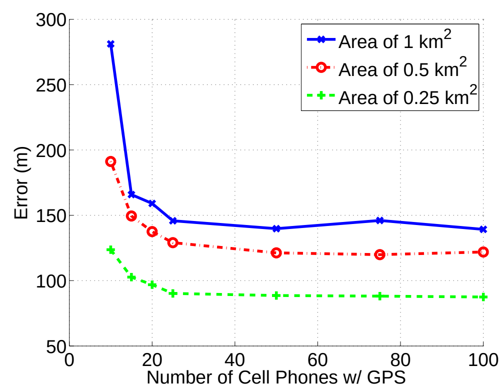

Fig. 7. Median localization error for areas of different size as a function of the number of GPS devices in the area.

larger for all three algorithms; however, the general picture looks very similar - fingerprinting algorithm and Calibree show good performance, achieving $138m$ and $214m$ median error respectively, while the centroid algorithm does poorly with $335m$ median error. These results are very encouraging because Calibree achieves similar accuracy to existing localization algorithms, without requiring off-line calibration. We note that, with 25 phones acting as real time calibration points, our calibration density is much less than that of the fingerprinting or centroid algorithms, yet Calibree still achieves accuracies in the same range as these much more labor-intensive algorithms.

We conjectured that there is a correlation between the area size under consideration and the number of GPS-equipped phones required to achieve similar localization accuracy. To test this correlation, we experimented with limiting the traces collected to only half and quarter of the original area size. Figure 7

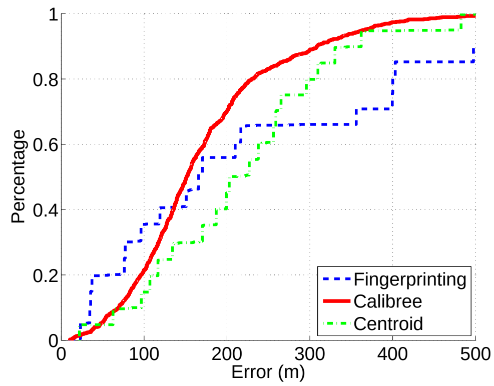

Fig. 8. Cumulative distribution function of absolute localization error for mobile phones located indoors.

shows the median localization error for area sizes of $1km^2$, $0.5km^2$ and $0.25km^2$ as a function of the number of GPS-equipped phones in the area. The results confirm that it is typically the case that increasing the target area requires more GPS-equipped phones in the area to achieve comparable accuracy. For instance, Calibree achieves $123m$ median error with 10 GPS devices in the area of $0.25km^2$, $149m$ median error with 15 GPS devices in the area of $0.5km^2$ and $147m$ median error with 25 GPS devices in the area of $1km^2$.

Finally, we tested the localization accuracy of Calibree, the fingerprinting and centroid algorithms on measurements collected inside 20 buildings of our university. The training of Calibree was performed as explained previously by picking 25 random points outdoors and estimating the regression formula. For testing all algorithms, we used 20 testing points, each collected in a different building. Because we knew where the buildings are located, we marked the ground truth of each of the 20 testing points manually in our trace. Figure 8 shows the CDF of absolute localization error for each of the three algorithms. Calibree achieves better localization accuracy than both the fingerprinting and centroid algorithms, reaching $151m$ median error vs. $165m$ median error for the fingerprinting algorithm and $203m$ for the centroid algorithm. Interestingly, the $151m$ median error that Calibree achieves on indoor measurements is very close to the median error of $147m$ that Calibree achieves on measurements taken outdoors. The results suggest that Calibree is effective in propagating the absolute location information from phones located outdoors to the phones located indoors.

Relative Positioning In this section, we show that even when no phones report their absolute locations, Calibree is effective at predicting relative distances between phones. Note that although our regression formula computes distances between mobile phones that detect at least one common cell, Calibree is also able to predict distances between phones that have no cells in common. To compute relative positions, the centroid and fingerprinting algorithms first compute

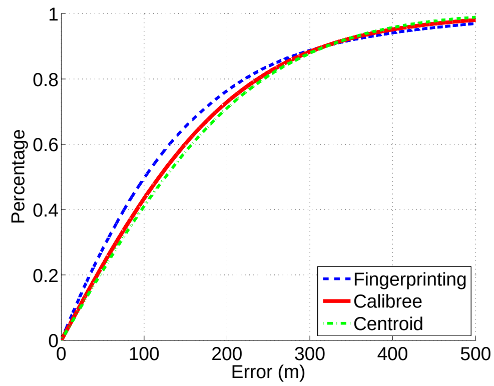

Fig. 9. Cumulative distribution function of pairwise distance estimation errors for Calibree, fingerprinting and centroid algorithms.

absolute phone positions and then calculate relative distances directly. Calibree, on the other hand, was trained using GSM signatures and absolute locations from 25 phones used in previous experiments.

Figure 9 shows the CDF of estimated relative distances between phones for Calibree, centroid and fingerprinting localization algorithms. The results show that Calibree is able to estimate distances between phones with a similar accuracy to that obtained by the centroid and fingerprinting algorithms, without the need to compute absolute locations first.

3.3 Discussion

Although we developed and tested Calibree off-line with stored data, we envision it as a real time Web service in its final form. Mobile clients would transmit their measured signal strengths and, if available, their GPS coordinates, to a central Calibree server. The server would run our algorithm and make the computed, absolute coordinates available to authorized subscribers. Clients with satisfactory GPS availability would not need the Calibree service, because they already know their location. These clients could be enticed to contribute their data by micro-payments, discounted subscriber rates, discounted location data for other users (assuming authorization), or an offer of free position data from Calibree whenever they lose GPS satellite connectivity. For privacy, GPS data and the associated signal strengths could be transmitted to the service completely anonymously, although this would complicate the process of giving compensation for those users. Alternatively, or in addition to regular GPS-equipped users, the service could exploit GPS and signal strength data from taxis, police cars, municipal vehicles, garbage trucks, and delivery vehicles, many of which are already equipped with GPS and cell phones. Cell phone companies and ordinary users could set up static GPS and cell phone stations to transmit absolute coordinates in regions of particular need, although Calibree ideally exploits only mobile users to avoid the need for extra infrastructure.

4 Related Work

The Calibree localization algorithm is related to several research efforts in ubiquitous computing and sensor networks. We next describe key distinctions between these efforts and ours.

4.1 Relative Distance Estimation

Several projects have suggested ways to compute ranging estimates between wireless devices. In SpotON [6], tags use received radio signal strength information as an inter-tag distance estimator. Relate [5] uses a combination of ultrasound and radio communication to infer relative position and orientation between specialized USB dongles. Calibree differs from these efforts in the way it computes relative distances between nodes. Instead of using peer-to-peer measurements between devices, Calibree estimates distances between devices based on measurements of signals from static beacons – cell towers in our case.

The technique Calibree uses for estimating pairwise distances was inspired by the NearMe wireless proximity server [8]. NearMe showed that it is possible to calculate the relative distance between WiFi devices based on their WiFi signatures. In contrast, Calibree applies this technique to GSM instead of WiFi, and it uses the computed distances to find absolute phone locations with help from GPS measurements from a small number of phones. Furthermore, Calibree takes advantage of a network of pairwise distance estimates to refine the location results, while NearMe stopped after computing just individual pairwise distances.

4.2 Graph-based Location Estimation

Self-mapping [9] is a graph-based algorithm for mapping radio beacon (e.g., WiFi APs or GSM cell towers) locations given a small seed of known beacon locations and a set of radio scans. The main idea behind self-mapping is that if a radio measurement contains two beacons, these beacons are located within twice the maximum transmission range of each other. A better distance estimate may be obtained by using a radio propagation model. Self-mapping combines the estimated distances into a graph and solves for beacon positions using an iterative error minimization algorithm.

A number of related localization techniques have been developed for sensor networks, mainly to improve network routing. Some of these approaches assume that a small number of beacon nodes with known locations are available [13, 10] and compute locations of other nodes in the network. Other approaches assume no such knowledge and compute only the relative node locations [2, 14].

Calibree applies similar techniques to localize GSM mobile phones instead of radio beacons or sensor nodes. However, Calibree estimates relative distances between phones using a regression formula derived in real time from a set of GSM signatures and their absolute locations. In contrast, self-mapping estimates distances between beacons based on a simple radio propagation model, while sensor network efforts either rely on sheer node-to-node connectivity or estimate relative distances using peer-to-peer measurements of time of flight [10] between sensor nodes. Finally, Calibree uses a different graph-based algorithm for constraint satisfaction: a variation of the Vivaldi algorithm [4].

Vivaldi is a distributed algorithm for predicting communication latency between Internet hosts without requiring explicit round trip time measurements between them. For that purpose, Vivaldi assigns each host a synthetic coordinate such that distances between the coordinates accurately predict the latency between hosts. Vivaldi uses round trip time measurements to estimate relative distances between network nodes, combines these measurements into a graph and then solves the graph constraint system using a distributed version of the mass-spring minimization method. In contrast, Calibree estimates relative distances based on phones' measurements of GSM cell towers and it calculates absolute, not relative, phone positions based on a seed of known locations of GSM signatures measured by devices with GPS.

5 Conclusions

In this paper, we presented Calibree, a novel GSM localization algorithm that does not require the tedious calibration phase that normally accompanies cell tower localization algorithms. Calibree uses a relatively small number of GPS-equipped mobile phones to compute the locations of mobile phones without GPS. Calibree takes advantage of the fact radio signatures are a reasonably good basis for computing the relative distances between mobile phones that detect at least one cell tower in common. Our algorithm combines these relative distance measurements with GPS measurements from some of the phones in an error-minimization procedure to compute the absolute locations of all the phones.

Our experimental results showed that the accuracy of Calibree is comparable to traditional, calibration-intensive algorithms for cell phone localization. Even as the number of cell phones equipped with GPS grows, Calibree will remain relevant, as it works for phones that are located indoors or otherwise unable to detect GPS satellites. Calibree is also an effective technique for computing the relative locations of a group of phones even if none of them have GPS. This can be useful for proximity-based applications and games.

Extensions to Calibree could include a technique to enforce spatial continuity of the inferred locations. As it is, Calibree computes locations based on measurements from a single instant in time. It may be possible to improve accuracy and robustness by smoothing or filtering location estimates across time, using probabilistic motion models. Such models could include path constraints on the mobile nodes, such as a network of streets, railways, and pedestrian paths. Considering both time and path constraints, it may even be possible to compute absolute locations from a sequence of relative locations by finding the unique set of absolute paths that could give rise to the inferred relative paths. Calibree could be extended to deal with the error inherent in GPS. As it is, the algorithm assumes that GPS measurements are always correct, although we know that GPS has its own error characteristics, including occasional outliers. The spring-mass scheme could be easily extended to account for some error in the GPS estimates.

References

1. P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of INFOCOM*, pages 775–784, 2000.
2. S. Capkun, M. Hamdi, and J.-P. Hubaux. GPS-free positioning in mobile ad-hoc networks. *Cluster Computing Journal*, 5(2):157–167, Apr. 2002.
3. 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*, Irvine, California, Sept. 2006.
4. F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *Proceedings of SIGCOMM*, pages 15–26, 2004.
5. M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for co-located mobile devices. 2005.
6. J. Hightower, R. Want, and G. Borriello. SpotON: An indoor 3d location sensing technology based on RF signal strength. Technical Report 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, Feb. 2000.
7. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer-Verlag, 3rd edition, March 2001.
8. J. Krumm and K. Hinckley. The nearme wireless proximity server. In *Proceedings of the Sixth International Conference on Ubiquitous Computing (UbiComp)*, pages 283–300, Sept. 2004.
9. A. LaMarca, J. Hightower, I. Smith, and S. Consolvo. Self-mapping in 802.11 location systems. In *Proceedings of the Seventh International Conference on Ubiquitous Computing (Ubicomp 2005)*, Lecture Notes in Computer Science, pages 87–104. Springer-Verlag, September 2005.
10. K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.
11. 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 2005)*, September 2005.
12. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, October 1992.
13. C. Savarese, J. Rabay, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of USENIX Technical Annual Conference*, 2002.
14. Y. Shang and W. Ruml. Improved mds-based localization. In *Proceedings of Infocom*, 2004.

15. R. W. Sinnott. Virtues of haversine. *Sky and Telescope*, 68(2), 1984.
16. I. E. Smith. Social mobile applications. *IEEE Computer*, 38(4):84–85, April 2005.
17. Sprint. Location based services network overview. Technical report, 2005.
18. A. Varshavsky, M. Chen, E. de Lara, J. Froehlich, D. Haehnel, J. Hightower, A. LaMarca, F. Potter, T. Sohn, K. Tang, , and I. Smith. Are GSM phones THE solution for localization? In *IEEE Workshop on Mobile Computing Systems and Applications*, April 2006.
19. GPS-Enabled Location-Based Services (LBS) Subscribers Will Total 315 Million in Five Years, http://www.abiresearch.com/abiprdisplay.jsp?pressid=731.