Low Energy and Sufficiently Accurate Localization for non-Smartphones

Kuldeep Yadav*, Vinayak Naik*, Amarjeet Singh*, Pushpendra Singh*, and Umesh Chandra+

*Indraprastha Institute of Information Technology, Delhi, India

+Nokia Research Centre, Bangalore, India

***Abstract*—**

Location-aware mobile applications are steadily gaining popularity across the world. However, lack of Global Positioning System (GPS) and absence of Wi-Fi infrastructure prevent users with non-Smartphones (majority of population in developing countries) from using location-aware applications as their phones do not have access to their current location. Existing GSM based approaches such as Cell ID-based works on non-Smartphones but they require access to a comprehensive database of Cell IDs. Such a database either does not exist or is very limited in developing countries.

In this paper, we propose a novel GSM-based approach of using Cell Broadcast Service (CBS) messages for getting current location on the phone. Proposed approach does not depend on a comprehensive database and can run on programmable low end phones. We demonstrate the effectiveness of our approach on data collected in New Delhi, India across two different operators and propose two space-time history based algorithms to improve upon the localization accuracy of our baseline CBS approach. The proposed algorithms provide up to 35% improvement in accuracy over the baseline method. Further, we compare accuracy of our CBS-based approach with that of Cell ID-based approach and also, present a multimodal approach that uses combination of both CBS and Cell ID (wherever available) to improve the localization accuracy.

I. Introduction

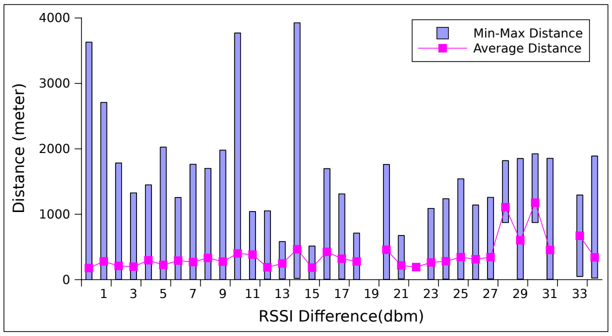
Location has been an integral part of user context in delivering context-aware services such as navigation, activity recognition, local business search, and friend finder services. Interestingly, all context aware services do not require same level of accuracy for current location. For instance, navigation applications require high level of accuracy (~10 m) whereas if one has to share location with online social networks, even location accuracy of hundreds of meter will suffice. Many technologies/approaches are available to measure user's current location on mobile phone of which the primary approaches are as follows:

1) **GPS**: Highly accurate (~10-100 meter) satellite based approach and most common for high-end phones. However, it consumes high energy, requires special hardware, and only works outdoors [13].
2) **WiFi based Positioning**: A perceptual map of wireless APs identifier with respective signal strengths and approximate location is created by wardriving and stored in a database [18]. The mobile phone queries this database to estimate the current location. Though, it can work indoors but it consumes high energy and requires special hardware besides needing a WiFi infrastructure, which does not exist in majority of developing countries.
3) **GSM based Positioning**: There are two kinds of GSM positioning approaches i.e Base station assisted and Base station independent. Base station assisted approaches require installation of sophisticated infrastructure on base station and hence require assistance from the operator [13]. Base station independent GSM positioning approach is based on Cell ID where, like WiFi based positioning system, a perceptual map of GSM Cell towers is created using war-driving and this is queried to estimate the current location of the phone [17].
   This method does not require any extra hardware and easily works on phones having GSM connection (~85% of phones) [19]. This approach have lower accuracy than GPS and WiFi (~100-1500 meters). The localization accuracy primarily depends on coverage of Cell ID database, Cell ID density in an area, and visibility of Cell IDs on phone.

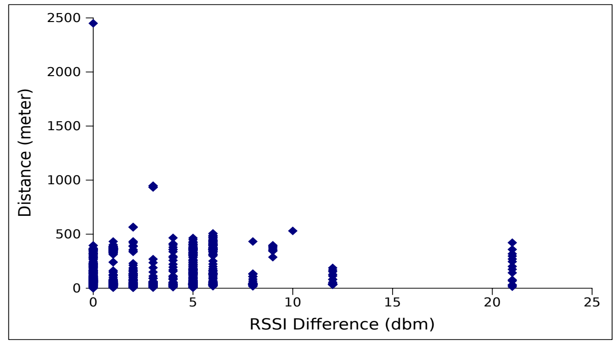
Mobile phones having GPS and Wi-Fi chips are expensive, so a large number of non-Smartphones[1] do not have these capabilities. It is predicted that for the next five years, over 50% of the phones will not have GPS [19]. In 2011, ratio of non-Smartphones to smartphone sales was 2:1 and majority of these phones had capability of internet connectivity also [2]. Apart from cost, mobile phones are highly energy constrained and continuous use of GPS and WiFi drains the battery quickly. For the class of applications that do not require fine grained location accuracy, Cell ID based GSM localization is better suited due it its wide availability and low power consumption. In fact, for low-end phones (without GPS/Wi-Fi capability) this is best suited [19]. However, GSM based localization needs to overcome following limitations:

1) According to GSM standards, a phone can receive signals from seven different Cell towers [17]. However, most of the phones can access (using APIs)

[1]Also, known as low-end phones or feature phones
[2]http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats



(a) In whole dataset, Min-Max bars representing minimum and maximum distance between two position into same cell ID. For instance, for a difference of 14 dBm in RSSI, distance between those points can range from 0 to 4000 meter.



(b) Only for single cell ID, X-Y scatter plot for showing variance in RSSI difference vs distance between two positions into same Cell ID. For instance, two points with RSSI difference 21 dBm can have a distance of 0 to 500 meter.

Figure 1: RSSI analysis on self collected dataset

only one Cell tower to which the phone is currently connected [14]. Access to only a single Cell ID offers coarse grained accuracy.

2) For Cell ID based localization, perceptual map (Cell ID database) has to be created by wardriving. Wardriving is not scalable because it is practically impossible to cover each and every street of a country to create a database of Cell Ids. Although, there are few crowd-sourcing based open source Cell ID database, e.g. OpenCellID, they only have few entries and often becomes obsolete due to lack of participation.

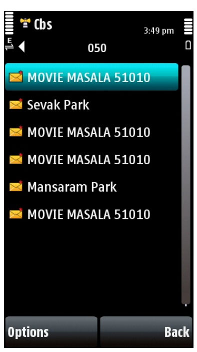


Figure 2: A native application in Nokia Symbian phone displaying last six received CBS Messages

We propose using Cell Broadcast Service (CBS) messages to provide localization for low end phones. CBS is a GSM standard in which nearby Cell towers broadcast their locality name. A phone can receive CBS messages from only one Cell tower to which it is currently connected. Proposed scheme removes the necessity of building Cell ID database and can support location aware services, that do not require fine grained accuracy. Figure 2 shows a native application, in a Nokia Symbian phone, displaying last six received CBS messages.

The primary contributions of the paper are as follows:

1) The first comprehensive study (to the best of our knowledge) to propose and evaluate CBS messages to provide localization for low-end mobile phones. We provide:
   a) An architecture of a system that uses CBS messages for localization
   b) Challenges in a realizing working localization system
2) Space-time history based algorithms to improve localization accuracy over the baseline approach. We provide:
   a) Evaluation of accuracy of the algorithms using 58 real world GSM traces for two different operators
   b) Improvement of up to 35% in accuracy over the baseline approach using the proposed algorithms
3) Comparison of CBS and Cell ID based approaches and a multimodal approach of using CBS and limited Cell ID database to improve upon the availability and accuracy.

The paper is organized as follows. Section II describes related work and problem definition. Section III presents CBS based localization architecture and Section IV discusses challenges associated with it. Section V presents two algorithms, which enhance accuracy of CBS-based localization over that of baseline approach. In Section VI, we present evaluation of proposed algorithms using collected data. In Section VII, we talk about using multimodal approaches to improve localization accuracy. Finally, we conclude in Section VIII with a discussion on potential applications built using CBS based localization.

II. Background and Problem Formulation

Prior work related to GSM-based localization can be divided into two categories: (A) Cell ID based Approaches and (B) Fingerprinting based approaches.

A. *Cell ID-based Approaches*

In this approach, Cell IDs are fetched using phone APIs, and looked up in an existing database to provide localization. To the best of our knowledge, none of the mobile phone operators reveal exact location of the Cell towers. Hence, using crowd sourcing/war driving data, cell tower location is approximated, which could be several hundred meter away from its actual location. If there are multiple visible Cell IDs, the approaches compute some function, e.g. centroid, of all the geo-coordinates (latitude and longitude) obtained from the database.

As discussed above, there are limitations on how much visibility phone APIs provide to third party developers for accessing Cell IDs. Many of prior works assume that phone APIs provide access to multiple Cell IDs, as far as seven, at a time [17].Our experience supported by other prior work [4], [16] shows that for several phones including Nokia S40, S60 phones, Samsung Android phones only provide access to only one Cell ID to which the phone is currently connected. This significantly reduces accuracy of the localization as compared that obtained had there been access to seven Cell IDs. Google Mobile Maps'(GMM) My Location[3] app works on a single Cell ID-based approach, where it provides a median localization error of 656.37 meter for a rural area and 503.89 meter for an urban area [3]. The localization error depends on density of cell towers. Since in urban areas, density of cell towers is high, so this method provides good location accuracy.

As identified in Section I, it is hard to get a comprehensive database of cell IDs. There are some proprietary databases, such as one used by GMM, which are not publicly shared. There exist open source initiatives, e.g. OpenCellID[4] and Cell Spotting[5], that build their database using crowd-sourcing. To check the coverage of open source cell ID databases, we selected two widely used operators in New Delhi. We call them X and Y for anonymity. On our self collected dataset of Cell IDs for operators X, we observed that out of 252 cell IDs, OpenCellID contained only 65. For operator Y, the number was only 21 out of 164 as shown in Table I. We cannot find out comprehensiveness of the GMM as it is not publicly available. However, given low penetration of Android phones in rural India, we postulate that the database will be underpopulated.

Crowd-sourcing for building cell ID database seems to be in-effective due to (A) lack of incentives as people need to incur airtime charges for contributing to the databases and (B) lack of GPS-enabled phones in developing countries.

Operator	No of cell IDs	Found on OpenCellID	%
X	252	65	31%
Y	164	21	13%

Table I: Success rate of Open Cell ID (most extensive open source database of cell IDs) on our dataset collected in New Delhi region

[3]http://www.google.com/mobile/maps/
[4]www.opencellid.org
[5]www.cellspotting.com

B. Fingerprinting-based Approaches

In this approach, RSSI (Received Signal Strength Indication) is also collected along with Cell IDs during war-driving. Typically, a fingerprint constitutes Cell IDs, their associated RSSI, and GPS locations that are represented in a vector form. For this approach, database size is larger and more effort is needed during war-driving. During the tracking phase, Cell ID(s) and associated RSSI are compared with stored vector space of fingerprints using KNN (K Nearest Neighbor) to user's location. Here, KNN uses euclidean distance in RSSI space as a metric to find closest stored fingerprint [17]. This approach gives better accuracy than the cell ID-based approaches since granularity of stored information is more. However, it requires more storage and computational capabilities.

Continuous war-driving effort is required in this approach because signal strength keeps on fluctuating due to changes in physical environment. It works good when there is a visibility of seven cell towers and their respective RSSIs. Recent results demonstrates that RSSI measure from single cell tower is not a good measure to calculate movement [4].

We conducted our own study to find out whether RSSI is a good metric for localization. An RSSI difference is the absolute change in the RSSI, for a given Cell ID, when user moves from one location to another. In our database, we had 24064 unique RSSI difference values from 410 unique cell IDs. We plot maximum, minimum, and average distances for each RSSI difference. As seen in Figure 1a, the average difference is almost constant for RSSI difference ranging from 1 to 9 dBm. We zoom in on one cell ID and plot the data (Refer Figure 1b). We observed the similar behavior for RSSI difference ranging from 1 to 6 dBm. This concludes that RSSI is not a good measure for GSM-based localization as one observes similar RSSI values between two points with large physical distance between them.

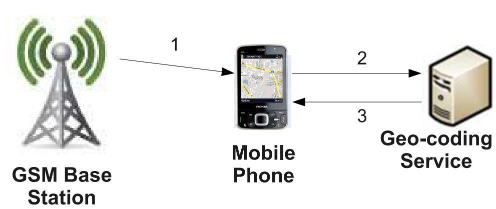


Figure 3: Architecture of CBS based Localization System

III. Architecture of CBS-based Localization and Pilot Collection of Data

The CBS messages are broadcast by Cell towers to all the phones in its range [2]. CBS is defined in the phase II of GSM standard 3.49 [1]. The users need not pay airtime charges to receive CBS messages, even while roaming outside of their home area. The CBS messages are commonly used to broadcast information about weather forecast, landmarks/area names, news, announcement by governments, etc. All this information can be broadcasted simultaneously on different channels. A cell tower typically broadcasts the locality/landmark name, where it is located. Channel 50 is reserved for broadcasting location/area names. Most of the phones come with built-in APIs to capture CBS messages.

A. Architecture of CBS-based Localization Scheme

In this subsection, we describe architecture of our proposed solution to use landmark names in the CBS messages to find users' locations. Figure 3 shows architecture of our working CBS-based approach. The data flows as depicted by numbered arrows in the Figure 3 correspond to the following:

1. GSM base station broadcasts CBS messages, each containing a CBS string mentioning location name or advertisement. The messages are received by our application running on the phone.

2. If the message content is a location, then the phone checks for geo-coordinates of the landmark in its local cache. If it is not available, the application makes a request to a cloud-based geo-coding service.

3. In reply to the geo-coding request, the geo-coding service returns geo-coordinates of that location to the phone. The application adds it to local cache of the phone. Cloud based geo-coding service is likely to get request from many phones, using which it builds a cache of all location names with their geo-coordinates. Phones can download this global cache proactively to avoid frequent requests to the cloud.

Above described approach is the most basic way of estimating a user's location using CBS messages and called as baseline approach. Baseline approach is identical to Cell ID approach described in Section II-A.

B. Pilot Collection of Data

To characterize the accuracy of CBS-based localization approach, we collected CBS messages for operators X and Y in urban setting of New Delhi, India. Five volunteers ran our data collection application for three months. We collected this data to measure accuracy of the baseline approach and design algorithms to improve upon the baseline accuracy.

Our data collection application is written in J2ME. We ran the application on Nokia S60 and Nokia S40 phones. Though we have collected data using Nokia phones, we have found that nearly all Java-enabled phones provide APIs to receive CBS messages. For example, phones from Samsung, Sony Ericsson, Black Berry, etc work fine but their APIs to get other location information like Cell ID differs since each manufacturer gives proprietary APIs to access information. Our application ran on all of these platforms with minor modifications.

The application collects CBS messages on channel 50, records the location message, time stamp of reception, Cell ID, MCC (Mobile Country code), MNC (Mobile Network Code), and GPS coordinates (if GPS is available on the phone). Volunteers were given choices to start and stop application at any point of time. After collecting each trace, participants tagged their activity as walking or traveling and upload it using one of following methods. – (a) using phone's data connection or (b) transferring it to PC first and then uploading it using PC's Internet connection.

Nearly half of our traces did not have GPS coordinates due to volunteers being indoor. For consistency purpose, we have only considered the traces, which had GPS values nearly all the time. We list out some of the statistics about the dataset in Table II. We analyze the collected data in the next section and list out challenges in using CBS message for localization.

State	X	Y	Combined(X+Y)	Avg Duration (Minutes)
Travelling	27	12	10	46
Walking	12	7	7	65

Table II: Number of travelling and walking traces in CBS dataset across two different operators X and Y

IV. Challenges in CBS based Localization

Data from our pilot study brought forth non-trivial challenges that require addressing before even the baseline approach can be used effectively. We addressed some of these challenges in our prior work [9], [10] but this paper presents comprehensive analysis with a bigger dataset.

A. Filtering of Advertisement Messages

CBS messages contain advertisements in addition to location names. It is essential to filter out these advertisements. We found that number of advertisements differ among operators X and Y, as shown in Table III.

Operator	Total CBS Messages	Advertisements(%)
X	3106	48%
Y	1173	60.53%

Table III: Percentage of Advertisement CBS Messages in our Dataset collected for operator X and Y

It was observed that advertisements contain some common patterns such as special characters ('*','#','%','@') or continuous digits like ('55050'). Using these two discriminators, we designed a regular expression to filter at runtime all the advertisements at the phone itself [10]. We got 100% accuracy in filtering advertisements when the regular expression was applied off-line to 4279 CBS messages in our dataset.

B. Geocoding of Location Names

As per our architecture, CBS location messages need to be geo-coded using a geo-coding service i.e. Google Maps. Among all the on-line maps services, we found Google Maps to be most effective in geo-coding our location names, and we have used it for all our experiments. We obtained 143 unique CBS location names in our dataset, among which 30% of location names could not be geo-coded by Google

Maps at first. We call them false negatives. Primary reasons for occurrence of false negatives are the following:

1) Location names may exist differently (in the geo-coding service), e.g. there could be a spelling difference, use of short hand abbreviations, or with a completely different name. For example, 'Matiyala' and 'Matyala', 'Uttam Nagar' and 'Uttam Ngr', 'Dwarka Sec-3' and 'Sec-3 Dwarka'.
2) There is no publicly available extensive GIS database.

We have employed following approaches on the location names, that could not be geo-coded directly by Google Maps:

1) Sanitizing Location Names : To resolve the ambiguity present in location names, we do a pre-processing of landmark names before sending them to the geo-coding service. Pre-processing algorithm apply following steps to sanitize the CBS location names:
   a) Replace special '-' character by a space, so 'Dwarka Sec-02', 'Dwarka Sec-2' and 'Sec-2-Dwarka' are converted to 'Dwarka Sec 02', 'Dwarka Sec 2' and 'Sec 2 Dwarka' respectively.
   b) Numerical characters in the location name are separated out from surrounding text characters e.g. converting 'Dwarka Sec2' to 'Dwarka Sec 2'.
   c) After removing special characters from location name, search for popular abbreviations in location names like 'NGR', 'SEC', 'VHR' etc., and replace then with its full form like 'NGR' for 'Nagar' followed by a space. We have manually populated this mapping table from the location names.

   After pre-processing by the above algorithm, Google Maps service was able to geo-code nearly 50% of the false negatives. Other 50% (15% of total) of the names were not present on the map service or existed with a different name. For instance, the location 'Dwarka Mor' exists on Google Maps and can be geo-coded, but same location with a different name 'Kakrola Mor' does not exist on Google Maps.
2) Use of on-line map based business search services: For the location names that are completely missing from digital maps or exist with a different name, we took help of the data present in on-line map-based business search services like Google local search. These business name are often collected through crowd-sourcing, so many of the location names (not found on Google maps otherwise) were present in business names. After retrieving business names, we applied K-means to approximate geo-coordinates for a location. However, currently we could not verify a location's geo-coordinates automatically and we leave it to future work.

We believe that a common algorithm that can work for all the names is hard to achieve due to non-standard nomenclature for CBS messages and poor GIS database (specially in developing countries). However, it is still a one-time task to geo-code the names which are not automatically geo-coded by any service and requires much less effort than the wardriving task used by Cell ID-based approaches.

C. Inaccuracy of Geo-coding Services

We are using geo-coding services, e.g. Google Maps, for finding geo-coordinates of the location names. There are inherent errors within these services, e.g. a location called "Dwarka Sec-3" (a neighborhood in Delhi) is not mapped with geo-coordinates representing the central location of that neighborhood. Such errors vary from one location name to another and get introduced in the result. From our dataset, we got an aggregate of 143 unique location names from CBS messages. During each of the traces, we also collected the GPS coordinates and mapped them to the corresponding location names received at that instant. Since there were multiple GPS coordinates mapped to a single location name, we took an average of all the GPS coordinates collected for a given location name and define that as the calculated GPS coordinate for the corresponding location name. We then calculate error in localization for a location name as the distance between the calculated GPS coordinates and the geo-coordinates returned by the geo-coding service for that location name.

Figure 4 shows a bar graph of distribution of error in terms of percentage of the location names. Out of 143 location names, 15% of the names could not be geo-coded. For about 58% of the names, which were successfully geo-coded, geo-coding error is more than 600m. This motivates us to build algorithms, which can reduce the error introduced by geo-coding.

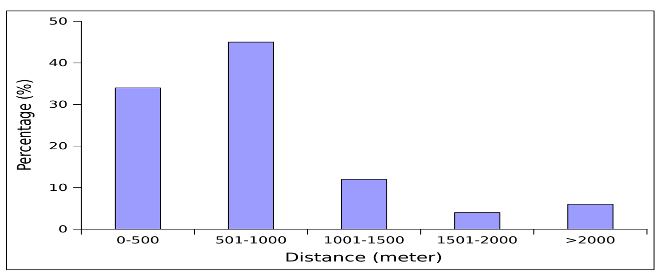


Figure 4: Distribution of error from inaccurate Geo-coding services. For 58% of the names, error is more than 600m.

D. Heterogeneity in Location Names Among Operators

Similar to Cell ID-based approaches, CBS-based approaches also suffer from operator heterogeneity, i.e. broadcasted CBS location names for a particular place differ across operators. This may affect localization accuracy. In Section VI, we will show an impact of operator heterogeneity on accuracy of localization by analyzing results from experiments with different operators. The challenge then is to tolerate this heterogeneity.

V. Algorithms To Improve Localization Accuracy

Baseline CBS based localization takes the most recently received CBS message's geo-coordinates to approximate the location of the user. Baseline approach does not always give good results due to two inherent errors: one is caused by geo-coding service (described in Section IV-C) and other due to the fact that CBS location names may be far away from user's actual location. A key insight towards reducing the impact of these errors is that we are not taking into account history of the locations visited by the mobile user in the recent past.

To account for location history, we form a vector of location names received in the past. When the user is stationary, the phone often receives multiple distinct location names as it can associate with different cell towers that are in geographic proximity at different time instances. These location names sometimes may include locations that, in reality are far away from user's current location. However, the frequency of such location names is far smaller than frequency of location names that are in close proximity to the current location. We hypothesize that this frequency difference is a factor of distance between Cell Tower and the user. Therefore, a weighted average based approach where the weight given to each location name is dependent on the frequency of received messages with the corresponding location name (within fixed time window) will intuitively work well for improving the localization accuracy. We call this approach *FrequencyWeighted* .

For a slow moving user, since the conditions are similar to a static user, the *FrequencyWeighted* approach should ideally provide better localization accuracy. However, a fast moving user will probably be in the range of a cell tower for a short duration and hence will receive a small number of (often only a single) CBS messages with the corresponding location. However, it may happen that the currently received location name corresponds to a location in real world that is ahead on the path of the user while the previously received location name was behind on the path of the user (a typical case when the location name is received immediately on crossing the cell boundary). Therefore, weighted average of the geo-coordinates of received location names with higher weight given to those that are received most recently and exponentially reducing the weights of location names received in the past will intuitively improve the localization accuracy. We call this approach *TimeWeighted* .

A. FrequencyWeighted *Algorithm*

The *FrequencyWeighted* algorithm considers all CBS location messages received in a fixed time window duration δ.As discussed above, this algorithm will intuitively improve localization accuracy in the case of static or slow moving user. The algorithm first extracts all the location geo-coordinates, in time window δ behind the current time, with their corresponding frequency of occurrence and then it computes weighted average from the extracted location geo-coordinates to find the location in terms of geo-coordinates of the user. The detailed description of this algorithm can be found in [11].

Time window parameter δ needs to be tuned according to user's mobility as a high value of δ could consider old location names and a low value could unnecessarily discard recent location names.

```
Algorithm: TimeWeighted
Input: Location Vector(LocVector) containing CBS location name,
       reception time stamp(ReceptionTime),
       GeoCoordinates[Lat,Lon] and a time out interval λ minutes
begin
    Index=0;
    RunningCoordinates= LocVector[Index].Geo-Coordinates;
    Index = Index + 1;
    while Index < LocVector.Size do
        TimeDifference = LocVector[Index].ReceptionTime -
        LocVector[Index-1].ReceptionTime;
        if TimeDifference < λ then
            RunningCoordinates = (RunningCoordinates +
            LocVector[Index].Geo-Coordinates)/2;
        else
            RunningCoordinates=
            LocVector[Index].Geo-Coordinates;
        end
        Index = Index + 1;
    end
    EstimatedCoordinates = RunningCoordinates;
    return EstimatedCoordinates;
end
```

Algorithm 1: Pseudocode of *TimeWeighted* Algorithm

B. TimeWeighted *Algorithm*

Assuming that CBS messages are received at more than a certain minimum rate, once every λ minutes, *TimeWeighted* algorithm considers all the received CBS messages in the past to calculate the current location of the user. In other words, whenever there is a long gap (more than λ minutes) in the reception, the algorithm forgets past history of messages and starts accumulating new history. The pseudocode of *TimeWeighted* algorithm is given in Algorithm 1. At the first location instance, the calculated location is same as the current geo-coordinates because there is no history available. Thereafter, the calculated location is the average of the current observed location and previously calculated location. As a result, the weights of previous location messages decrease exponentially with time.

We developed a service for mobile phones that implement both *FrequencyWeighted* and *TimeWeighted* algorithms. It receives all CBS location messages and stores them in a location vector. All these locations are geocoded using Google Maps API. Whenever any application needs current location of the user, the service takes location vector as an input and returns calculated coordinates.

It is important to note that our approach (aimed for low-end phones) cannot assume any means, e.g. accelerometer or GPS, to measure the speed of the user and accordingly adapt the averaging policy for improved localization. We therefore compare the two approaches - *FrequencyWeighted* and *TimeWeighted* with the baseline approach empirically for cases with slow and fast user speed.

VI. Evaluation of The Algorithms' Accuracy

We now describe the empirical evaluation of the two algorithms, *FrequencyWeighted* and *TimeWeighted* , explained in the previous section, using our self collected real-world dataset. We used point-based localization approach as a baseline for comparison which is identical to the one used by cell ID based localization approach, including service providers like Google as described in Section II-A. We use localization error as our evaluation metric. It is the distance between actual location (GPS Coordinates) and predicted location (CBS based approach). For simplicity purpose, we discuss only one operator's result (referred to as operator Y) in detail. However, at the end of Section VI-A and Section VI-B, we also briefly present results for operator X.

As hypothesized earlier, the accuracy of the algorithms could depend on the speed of travel. Hence, we collected traces for two different motions of walking and traveling. We define walking as movement at an average speed of approx 3.5Km/h and traveling as movement at average speed of approx 30Km/h.

A. Traveling Traces

Let us first analyze the effect of varying input parameters on the performance of two algorithms. For *TimeWeighted* algorithm, λ is a time-out parameter, which is necessary to forget old history. Empirically, we found optimum λ to be 2 minutes since it gave the least median localization error for all the traveling traces. We, therefore, have used λ as 2 minutes for evaluating the performance of *TimeWeighted* algorithm. For *FrequencyWeighted* algorithm, parameter δ is used to fix the time window within which it considers the received CBS messages to perform weighted average. Empirically, we found optimum δ to be 2 minutes for traveling traces since it gave the least median localization error for all of traveling traces. We, therefore, have used δ=2 for evaluating the performance of *FrequencyWeighted* algorithm.

Figure 5a compares the Cumulative Distribution Function (CDF) of localization error for *TimeWeighted* and *FrequencyWeighted* algorithm with the baseline approach. Both *TimeWeighted* and *FrequencyWeighted* algorithms perform consistently better than baseline. The improvement in localization accuracy for *TimeWeighted* and *FrequencyWeighted* over baseline is approximately 12% and 16% respectively, as shown in Table IV.

Let us discuss intuition for performance of the two algorithms for traveling case. Typical rate of arrival of CBS message is 1 per minute. With λ fixed to 2 minutes and average speed of traveling trace as 30Km/h, if no CBS message is received for 2 minutes, the user has approximately moved by 1Km from the location of previously received CBS message. It is therefore better for *TimeWeighted* algorithm to discard the history of CBS messages than to consider them for future calculation of localization. Similarly, with δ fixed to 2 minutes, *FrequencyWeighted* algorithm will only consider CBS messages received within a distance of 1Km for calculation of localization, giving weights based on frequency of each CBS message received. This will mostly translate to average of two distinct CBS messages received in the 2 minute interval. Therefore, in case of traveling trace with correspondingly fixed parameter values, the two algorithms differ in that *FrequencyWeighted* algorithm never considers any CBS message outside the 2 minute window while the *TimeWeighted* algorithm gives any message outside the 2 minute window a small weight in case there is no time out in received rate of CBS messages. Soon after the time out, for the first 2 minutes, calculated localization for the two algorithms will be same.

Traces	Baseline	*TimeWeighted*	*FrequencyWeighted*
Travelling	621.40	549.82	521.52
Walking	712.94	462.54	644.85

Table IV: For operator Y, median localization error comparison of *TimeWeighted* and *FrequencyWeighted* algorithms with baseline for walking and traveling traces

Traces	Baseline	*TimeWeighted*	*FrequencyWeighted*
Travelling	688.2	618.29	615.14
Walking	466.69	382.8	386.56

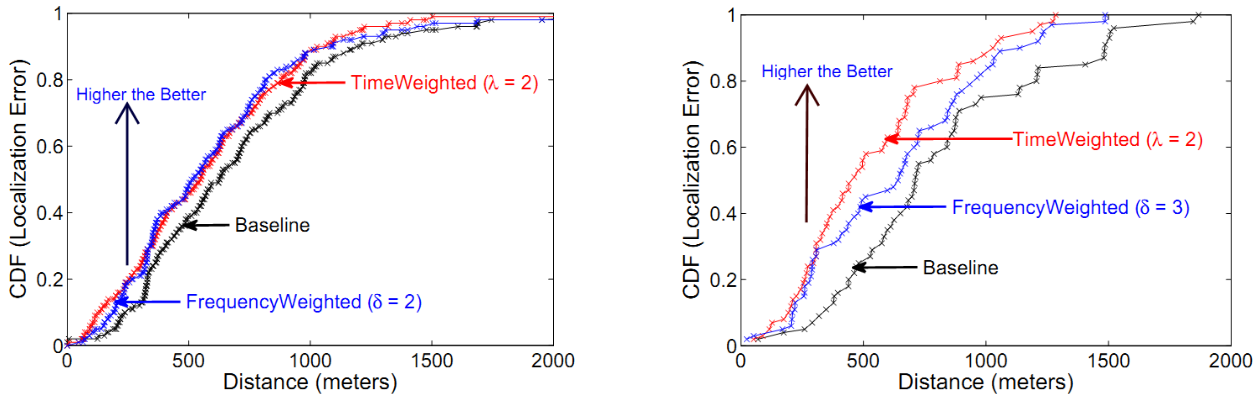
Table V: For operator X, Median localization error comparison of *TimeWeighted* and *FrequencyWeighted* algorithms with baseline for walking and traveling traces

For operator X, both algorithms perform equally good as compared to baseline. The improvement in localization accuracy for *TimeWeighted* and *FrequencyWeighted* over baseline is approximately 10% and 11% respectively, as shown in Table V.

B. Walking Traces

In walking traces, there is no such instance where CBS location messages were not received for a significant amount of time. However, we still kept λ equal to 2 minutes for *TimeWeighted* algorithm to maintain uniformity across both traveling and walking traces. For the case of *FrequencyWeighted* algorithm, we again empirically calculated the most optimal value of δ that came out to be 3 minutes. Intuitively, higher value of δ, as compared to the case of traveling traces, is justified since longer history of CBS location messages will be useful due to lower speed.

Figure 5b show the CDF plot of *TimeWeighted* and *FrequencyWeighted* algorithm performance as compared to



(a) Comparison for traveling traces (b) Comparison for walking traces

Figure 5: CDF plots for *TimeWeighted* and *FrequencyWeighted* algorithms w.r.t to Baseline for operator Y

baseline for walking traces. As shown in Table IV, overall *TimeWeighted* and *FrequencyWeighted* give an accuracy improvement of approximately 35% and 10% respectively over the baseline approach.

Intuitively, we had hypothesized *FrequencyWeighted* algorithm to provide higher localization accuracy than *TimeWeighted* algorithm for walking traces (as also discussed in Section V). However, empirical study showed otherwise. Close observation of the collected data revealed that the walking traces contained a lot of location names, that were farther located, 1200-1500m, from cellphone's actual location. This noise, particularly, gets added by the geo-coding service and presence of distant location names, which are among the challenges mentioned in Section IV. Effect of this noise can also be seen in terms of higher point-based localization error for walking traces (712.94m) as compared to traveling traces (621.4m).

Although *FrequencyWeighted* algorithm is hypothesized to have better accuracy for walking traces but, if the message containing distant location name is repeated within the δ time interval, it will have significant effect on the location computed by *FrequencyWeighted* algorithm (with fixed δ). On the other hand, for *TimeWeighted* algorithm, when such a CBS message with distant landmark name is received most recently, the calculated location is inaccurate. However, as the time progresses the weight of the CBS message with distant location is reduced, correspondingly resulting inaccuracy is reduced in calculated location as well.

We also conclude that our initial assumption that fast and slow motion patterns would demand different approaches for improved localization was empirically found incorrect on our collected data. As shown here, *TimeWeighted* algorithm that was hypothesized to handle fast motion suffices for slow motion as well since it tolerates the noise added by the geocoding service for real data. However, we believe that the localization accuracy may vary across different environments. Therefore, an approach that can adapt based on accurate location input known intermittently from an oracle (in physical world through the GPS coordinates from intermittently turned on GPS or from a GPS enabled phone in close proximity) will reduce error in localization accuracy significantly. For operator X, baseline accuracy was good due to good quality of landmarks. Improvement in localization accuracy for *TimeWeighted* and *FrequencyWeighted* over baseline is approximately 18% and 17% respectively, as shown in Table V.

C. Impact of Operator Heterogeneity on Accuracy

We observed that different operators provide different CBS location names as well as with different time interval (broadcast cycle). We analyze the impact of operator heterogeneity on localization accuracy by collecting walking and traveling traces with two different phones, each having a operator X and Y. For a fair comparison across two different operators, we selected traces which were collected together for Operator X and Y on the same geographic path described in Table II.

Table VI shows the median localization error for the three different approaches across both the operators. Although the individual errors are different for each operator, we observe that *TimeWeighted* algorithm consistently performs better for both the operators. This empirically confirms with our finding that *TimeWeighted* algorithm is able to tolerate different broadcast cycles of the operators.

Algorithm	**Walking**		**Traveling**	
	X	Y	X	Y
Baseline	670.71	641.08	530.87	712.94
TimeWeighted	577.11	581.38	318.5	462.54
FrequencyWeighted	562.41	529.82	343.07	644.85

Table VI: Median localization error comparison of different algorithms

VII. Multimodal Approaches with CBS-based Localization

CBS-based approach is a wardriving-free localization technique, that provides an alternative to Cell ID approach.

In this section, we compare accuracy of CBS-based localization with Cell ID based approach. Cell ID database availability is variable in different areas because it depends on various other factors such as network (GPRS/EDGE/HSDPA), operator etc. As shown in Table I, open source Cell ID database such as OpenCellID have very limited coverage across both the operators, we used Google Cell ID database[6] that has a good coverage (˜90% in our dataset) of operator X's Cell IDs for 2G network but very limited for operator Y. Therefore, it makes sense to combine CBS and Cell ID based approaches to improve the overall localization.

A. Comparison with Cell ID based Approach

As *TimeWeighted* (TW) algorithm performed equally good for both walking and traveling traces across different operators, we will use *TimeWeighted* algorithm for further experiments. Figure 6 presents a CDF for a comparison between CBS with TW algorithm and Google's Cell ID based approach. Cell ID data was taken from Google Cell ID database. The median localization error of CBS only with TW algorithm was 585.81 meter as compared to 254.11 meter provided by Cell ID based approach. Following are the primary reasons of high localization error of CBS approach as compared to that of Cell ID based approach:

1) The Cell ID database is more granular than CBS location messages because different Cell IDs in an area may broadcast same location name as was also observed in our collected data.
2) Cell ID database geo-coordinates are mostly collected on main streets using war-driving/crowd-sourcing. Since, most of our data is also collected from such streets, it produces low error as compared to CBS based approach.

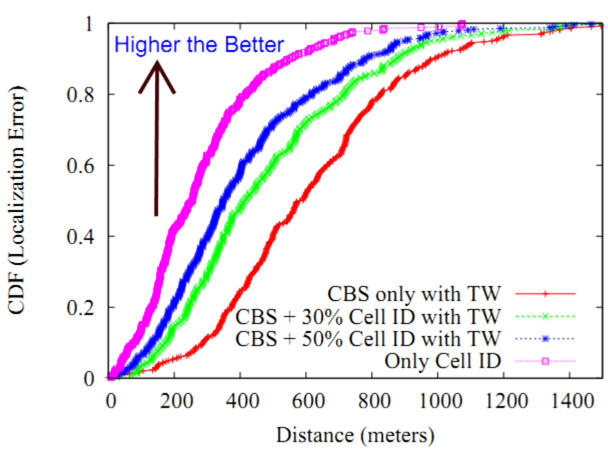


Figure 6: Comparison of CBS with TW algorithm with (A) only Cell ID based approaches and (B) combinations of CBS with TW algorithm and Cell ID

[6]Google does not provide official APIs to access Cell ID database

B. Cell ID + CBS based Approaches

We investigated whether limited Cell ID database can be used in conjunction with CBS based localization for improving localization accuracy. A combined localization scheme can use Cell ID whenever it is available in the database or otherwise make use of CBS. Based on GMM Cell ID data, we simulated two different scenarios where 30% or 50% of Cell IDs can be found in a Cell ID database for each trace. Assignments of Cell ID coordinates in a traces was done randomly. After combining geo-coordinates from CBS and available Cell IDs, we have applied *TimeWeighted* algorithm to further refine accuracy.

Figure 6 presents the comparison of different combinations of Cell ID and CBS with only Cell ID and only CBS based approaches. We have found that if 30% of Cell IDs can be found in the Cell ID database, it can result in 29.2% improvement in localization accuracy whereas for 50% of Cell IDs in same traces, this improvement can be up to 40.2%. Cell ID based localization require a pre-built Cell ID database, which is limited in many parts of the world. Whereas CBS based localization provides coarse grained accuracy. Therefore, combining of these two approaches can provide a robust and sufficiently accurate localization for low end phones because Cell ID-based localization provides good accuracy where as CBS-based scheme can improve the overall availability of localization.

VIII. Conclusion and Future Work

Proposed CBS based localization approach removes the necessity of war-driving or building a Cell ID database for GSM based localization. Evaluation using real-world traces show that proposed approach can provide reasonably good accuracy which is sufficient for many location based services. Hence, CBS-based localization is a promising solution for non-smartphones and provides them with an opportunity to access location based services without any extra infrastructure. The localization accuracy provided by the baseline solution is low due to geo-coding noise produced by non-standardization of CBS location names which may be far from actual location.

Proposed algorithms, *TimeWeighted* and *Frequency-Weighted* reduce impact of these errors by taking space time history. Using empirical evaluation, we observed that *TimeWeighted* can work for both walking and traveling traces across two different operators. Our algorithms do not pose any special requirements at backend or phone client and can be easily deployed in real world. Though Cell ID based approach provides good accuracy than CBS based approach whenever fine grained data is available but they have limited availability or provide low accuracy whenever there is limited war-driving data.

We proposed a multimodal approach of combining Cell ID, wherever available, with CBS messages to provide accurate localization and increased availability. We already built some real world applications using CBS based localization and will provide APIs so that application developers can use it in their applications.

Some of the real world applications, that can be built using CBS based localization for low cost phones includes:

1) Activity Classification: CBS message reception rate can be used to do binary activity classification (walking/static and travelling). We observed that number of CBS messages (location names + advertisements) received per minute is higher than two in walking whereas it is lower than two in traveling traces.
2) Location Sharing and Local Search: Most of the people in developing countries, like India, do not use digital maps for navigation and searching local businesses [12]. We built a local search application, which estimates current location using CBS location names and fetches relevant entries from local business database in vicinity of current location with a choise of GPRS/SMS based communication. CBS location names can also be used to share location with user's friends and across her social networks.
3) Trajectory Matching: Many location aware services require access to a trajectory (route travelled), which is built using periodic location samples. Such services include fleet management, mobile object/asset tracking applications [19], etc. We have found that combination of CBS and Cell ID information further combined with street map data can result in low cost but accurate trajectory matching.

In future, we plan to combine CBS based localization approach with GPS to reduce energy consumption by periodically sampling GPS. Also, we are building a model from the collected data to theoretically investigate the optimality of *TimeWeighted* approach.

IX. Acknowledgements

This work is partially funded by a research grant from Nokia Research Centre and a PhD Fellowship by Microsoft Research, India. We would like to thank anonymous reviewers for detailed reviews and helping improving the quality of the paper. Finally, we would like to thank all the volunteers who helped us in collecting data.

References

[1] http://cell-broadcast.blogspot.com/2005/11/history-and-importance-of-cell.html

[2] http://www.gsmhelpdesk.nl/en/helpdesk/helpdesk.php?id=57

[3] Mohamed Ibrahim and Moustafa Youssef,A Hidden Markov Model for Localization Using Low-End GSM Cell Phones, http://arxiv.org/abs/1010.3411.

[4] Jeongyeup Paek, Joongheon Kim, Ramesh Govindan, Energy-Efficient Rate-Adaptive GPS-based Positioning for Smartphones, MobiSys'10.

[5] GPS and Mobile Handsets. http://www.berginsight.com/ReportPDF/ProductSheet/bi-gps4-ps.pdf.

[6] Constandache, I., Choudhury, R. R., and Rhee, I. 2010. Towards mobile phone localization without war-driving, IEEE Infocomm'2010.

[7] Robert P. Biuk-Aghai, GSM-Based Provider-Independent Positioning Method, Location Asia 2007, Hong Kong, China, 13-14 September 2007.

[8] Jain, S., Ghosh, R.K., Shyamsundar, R.K., Engineering location based pathfinding on Indian road networks over low end mobile phones, Communication Systems and Networks 2010, pp.1-9, Jan 5-9, 2010.

[9] Kuldeep Yadav, Vinayak Naik, Amarjeet Singh, Pushpendra Singh, Ponnurangam Kumaraguru, and Umesh Chandra, Challenges and novelties while using mobile phones as ICT devices for Indian masses: short paper, NSDR'10.

[10] Kuldeep Yadav, Vinayak Naik, Pushpendra Singh, Amarjeet Singh, Alternative localization approach for mobile phones without GPS. In Middleware '10 Posters and Demos Track.

[11] Kuldeep Yadav, Vinayak Naik, Amarjeet Singh, and Pushpendra Singh, Low Energy and Sufficiently Accurate Localization for Non-Smartphones, IIIT-D Technical Report (IIITD-TR-2011-009), Nov 2011.

[12] Arun Kumar, Dipanjan Chakraborty, Himanshu Chauhan, Sheetal Agarwal, Nitendra Rajput, FOLKSOMAPS: Towards Community Driven Intelligent Maps for Developing Regions, ICTD 2009.

[13] Yilin Zhao, Mobile phone location determination and its impact on intelligent transportation systems," , IEEE Transactions on Intelligent Transportation Systems , vol.1, no.1, pp.55-64, Mar 2000.

[14] Paek, J., Kim, J., and Govindan, R. 2010. Energy-efficient rate-adaptive GPS-based positioning for smartphones. Mobisys'10.

[15] Telecom Regulatory Authority of India : http://www.trai.gov.in/WriteReadData/trai/upload/PressReleases/756/pr7sep.pdf

[16] Petteri Nurmi et al, A Grid-Based Algorithm for On-Device GSM Positioning, UbiComp 2010.

[17] Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith, and Alex Varshavsky, Practical Metropolitan-Scale Positioning for GSM Phones, UbiComp 2006.

[18] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm, Accuracy characterization for metropolitan-scale Wi-Fi localization. In Proceedings of MobiSys'05.

[19] Arvind Thiagarajan and Lenin S. Ravindranath and Hari Balakrishnan and Samuel Madden and Lewis Girod, Accurate, Low-Energy Trajectory Mapping for Mobile Devices, NSDI'11.