Energy-Efficient Positioning for Smartphones using Cell-ID Sequence Matching *†

Jeongyeup Paek‡ Kyu-Han Kim⋆ Jatinder P. Singh⋆ Ramesh Govindan‡

‡ Department of Computer Science
University of Southern California
{jpaek, ramesh}@usc.edu

⋆ Deutsche Telekom R&D Laboratories USA
Los Altos, CA, USA
{kyu-han.kim, jatinder.singh}@telekom.com

ABSTRACT

Many emerging location-aware applications require position information. However, these applications rarely use celltower-based localization because of its inaccuracy, preferring instead to use the more energy-hungry GPS.

In this paper, we present CAPS, a Cell-ID Aided Positioning System. CAPS leverages near-continuous mobility and the position history of a user to achieve significantly better accuracy than the celltower-based approach, while keeping energy overhead low. CAPS is designed based on the insight that users exhibit consistency in routes traveled, and that cell-ID transition points that the user experiences can, on a frequently traveled route, uniquely identify position. To this end, CAPS uses a cell-ID *sequence matching* technique to estimate current position based on the history of cell-ID and GPS position sequences that match the current cell-ID sequence. We have implemented CAPS on Android-based smartphones and have extensively evaluated it at different locations, and for different platforms and carriers. Our evaluation results show that CAPS can save more than 90% of the energy spent by the positioning system compared to the case where GPS is always used, while providing reasonably accurate position information with errors less than 20% of the celltower-based scheme.

Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Real-time and embedded systems; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Algorithms, Design, Experimentation, Performance, Measurement

*This material is based upon work that Jeongyeup Paek conducted during his internship at *Deutsche Telekom Inc. R&D Labs USA*.

†This material is based upon work supported by the National Science Foundation (NSF) under Grant Nos. 0121778, 0520235 and 0905596. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

Keywords

Energy-Efficient, Adaptive Positioning, Smartphone, Localization, GPS, Cell-ID, Sequence Matching, Smith-Waterman Algorithm

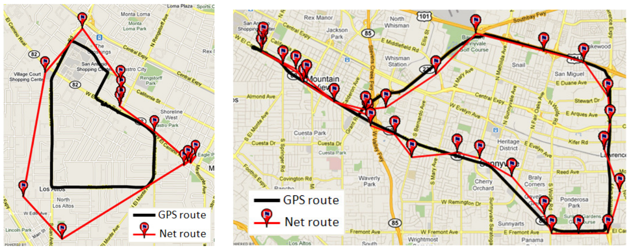
1. INTRODUCTION

Many mobile applications for smartphones require position information, and the usage of these applications is rapidly increasing. For instance, many mobile services such as mobile search, advertisement, social networking, and mobile gaming (e.g., Second Life) heavily use users' position information to provide location-informed responses to search queries or to continuously log user locations [9, 10, 12, 13, 18, 22, 23, 27].

Although there exist several positioning systems, they have some inherent limitations. Celltower-based localization is highly energy-efficient, but it incurs errors as high as 500 meters, which may be insufficient for the applications mentioned above. Because it is more accurate, the GPS (Global Positioning System) is widely used by many applications. However, it is also well-known that GPS is extremely power-hungry. Keeping GPS activated continuously would normally drain the battery on a smartphone in less than 12 hours, even in the absence of any other activity [17, 24, 30, 32]. This is a significant roadblock on the way to all-day smartphone usage. Ideally, an appropriate positioning system should be able to provide accurate position information, while spending minimal energy.

These trade-offs have prompted the development of a class of *lightweight positioning systems*, which has explored a large part of the energy-accuracy tradeoff space. These systems either relax accuracy requirements to limits that would be acceptable to many applications (for example, to within 100m) or aggressively use other *cues* to determine when and when not to turn on GPS, or both [7, 16, 17, 21, 24]. Examples of such cues include other sensors on the phone (such as the accelerometer, the Bluetooth interface, the wireless interface, the audio device), or past user behavior. Implicitly or explicitly, these systems make some assumptions about the environment (e.g. the availability of WiFi) or about user activity (e.g. walking or riding on a bus). We envision a future where smartphones will contain implementations of different lightweight systems, each suited to different environments and/or user activities, and selectively triggered under the appropriate circumstances.

In this paper, we consider a part of the design space of lightweight positioning systems that has not been explored yet, a system for highly mobile users who travel long distances in a predictable fashion. Examples of such users include those commuting to work using public transit or other modes of transportation that do not offer convenient handset charging. The constituency of such users is large outside North America: for example, over 5.2 million peo-



(a) 'Los Altos Route' (b) 'Sunnyvale Route'

Figure 1: Celltower-based localization vs. GPS reported route: trace comparison.

Figure 2: Celltower-based localization vs. GPS comparison, multiple iterations

ple take buses everyday in Seoul, Korea, and in large cities in India, over 60% of the population uses mass transit in some form or another [25]. The consistency of transit commuters allows the positioning system to use a history of previously traveled routes to cheaply estimate current position.

Our Cell-ID Aided Positioning System (CAPS) achieves position accuracy comparable to that of GPS while achieving very low energy usage. At its core, CAPS is based on the observation that, for mobile users with consistent routes, the cell-ID transition point that each user experiences can often uniquely represent the current user position. Specifically, CAPS estimates current user position using the cell-ID information freely available on smartphones, along with the history of past GPS coordinates. It stores cell-ID sequences and GPS readings taken by the mobile device during daily use in a sequence database, and then finds a matched sequence in his history using a modified Smith-Waterman algorithm. With adequate history, CAPS can extract accurate position information from this sequence database without, in many cases, activating GPS.

We have implemented CAPS on Android-based smartphones and have evaluated its implementation in several locations, on different platforms and cellular networks. CAPS is implemented as an application service that does not require any change in underlying mobile platforms. Further, its implementation is compact (e.g., 42 KB code footprint) and it has low computational complexity. Our evaluation results reveal that CAPS can improve energy efficiency by more than 90 % compared to a GPS-only approach. Furthermore, it provides reasonably accurate position information with errors less than 20 % of celltower-based triangulation.

Thus, our main contributions can be summarized as follows:

- We present a new energy-efficient positioning scheme that provides better accuracy than a celltower-based scheme for various location-based services.
- We apply a novel cell-ID sequence-matching algorithm that significantly reduces energy in obtaining position information.
- We have implemented the proposed system in the Android OS and demonstrate its improvement in accuracy and energy efficiency in several locations, and on a variety of platforms and carriers.

The remainder of this paper is organized as follows. Section 2 explains the challenge of energy-efficient positioning, Section 3 discusses the central observations that motivate our approach. Section 4 presents a detailed design of CAPS, while Section 5 describes our extensive evaluation. Section 6 discusses some interesting issues that arise in the design of CAPS. Finally, Section 7 summarizes related work, and Section 8 concludes the paper.

2. MOTIVATION

In this section, we motivate our work by presenting several experimental results on the inaccuracy of celltower-based localization. We demonstrate that, at least for the areas in which we have collected data (locations in Northern and Southern California, USA), there is significant error in positions reported by celltower-based localization, often as high as a few kilometers. Finally, based on these observations and motivated by the high energy cost of GPS, we argue for a new energy-efficient positioning system.

To understand the inaccuracy of celltower-based localization, we measure positions using the *network-based* localization option in the Android Location API. For this reason, we label cell-tower based localization as NET. We wrote software that continuously logged raw GPS readings and NET readings every 2 seconds using the Location API provided by Android OS on a Nexus One phone. During our experiments, the WiFi interface was disabled, to ensure that the Location API did not attempt to use visible WiFi access points for localization.

Figure 1(a) plots two traces collected using this software by driving in a vehicle in the streets of Los Altos, CA. (We will explain our experimental methodology and the features of these traces in more detail in Section 5). The thin gray line with flags in the figure represents all the discrete positions reported by NET, and the thick black continuous curve represents GPS logs. The Google Maps API v3 was used to plot the recorded points in the figure.

From the figure, we can observe two major limitations of NET. First, even though the location API was invoked every 2 seconds, and there was continuous movement, NET reports only a small number of discrete positions (the flags in the figure). Second, its positioning error, which is calculated as the difference between NET and GPS, is extremely high — a median error of 390 meters with some error values as high as a few kilometers.

To verify that this result is not an artifact of a single unfortunate experiment, we have conducted several experiments for the same route (Figure 2). Different runs along the same route produce significantly different positions at the same location. Moreover, on a different route (Figure 1(b), in Sunnyvale, CA), the error is also high. For this route, the error may not be apparent visually, but it is evident from Figure 3 that the error is high. The figure shows the positioning error CDF for all four different routes that we have

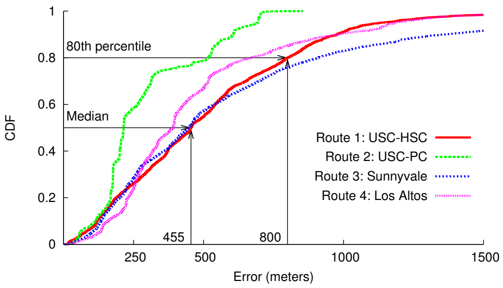


Figure 3: CDF of position error for celltower-based localization for four different routes.

experimented on. It shows that NET has a median positioning error of around 400 meters and an 80^{th} percentile error of around 800 meters. Taken together, these results confirm that NET provides extremely inaccurate position information and that its positioning information is consistently inconsistent — different readings are obtained at different times at the same physical location.

Due to its inaccuracy, many location-based applications rarely use cell-tower based positioning, despite its energy efficiency. Instead, they often prefer to use GPS which is more accurate but power-hungry. As our measurements on several platforms show (Table 1), GPS can drain the phone battery within 10 hours when used continuously, and in 40 hours even when used for 1 minute per every 5 minutes.[1] On the other hand, obtaining positioning information from NET, or even accessing the ID of the currently visible celltower, requires a negligible amount of power on the devices that we have experimented with.

One approach to reducing the energy spent by GPS is to periodically duty-cycle GPS at a fixed interval. However, as other work has shown [24], determining the periodicity is a significant challenge and periodic duty-cycling can incur high error if the periodicity is not well-matched to the rate of user movement. Furthermore, periodic duty-cycling may not help reduce energy significantly because GPS can take up to several tens of seconds after activation to get a position fix, and many GPS receivers have a power-down delay ranging from 5 seconds to 60 seconds (Table 1). A recent approach to adaptive duty-cycling, RAPS [24], achieves much of its energy savings by avoiding GPS activation indoors; as we show later, it is possible to achieve substantial energy savings beyond RAPS, even in outdoor settings.

In summary, we have made three important observations in this section: i) Celltower-based localization is extremely inaccurate, ii) GPS consumes significant energy, and iii) periodic or adaptive duty-cycling may not achieve significant energy savings under all conditions. This motivates the need for a positioning system that expends minimal energy while providing position information with reasonable accuracy. In what follows, we propose a novel approach—Cell-ID sequence matching—to meet this goal.

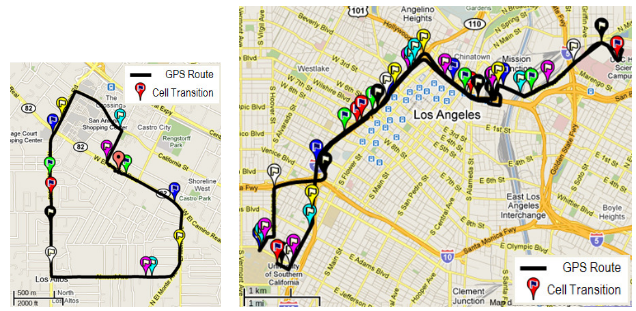
3. USING CELL-ID SEQUENCES FOR POSITIONING

Celltower-based positioning systems use triangulation to estimate user positions. In contrast, we explore an approach that uses the sequence of celltower identifiers (henceforth, cell-IDs) traversed by a user to estimate position. Such identifiers are readily available

[1] We have used the Power Monitor device from Monsoon Solutions Inc. for our power measurements.

Phone	GPS On Power	Turn-off Delay		1min/5min
		Time	Avg.Power	Est.power
Google G1	387 mW	5 sec	same as On	98.9 mW
Nexus One	327 mW	5 sec	same as On	88.4 mW
MotoDroid	363 mW	60 sec	same as On	105.4 mW
Nokia N95	372 mW	30 sec	158 mW	140.6 mW

Table 1: Power consumption measurement result for four different types of phones.



(a) Los Altos Route (b) USC-HSC Route (Los Angeles)

Figure 4: Cell-ID transition points in our traces.

on many phones, and we leverage spatio-temporal consistency in user mobility to aid position estimation.

Motivating Examples. To illustrate our approach, consider the following scenario. Suppose a user's cell-ID changes from 1 to 2 as shown in Figure 5(a).[2] This information alone is not sufficient to determine which of the three positions A, B, or C the user is currently in. Now consider the scenario in Figure 5(b), which has additional information about the underlying geography. Since B and C point to positions at sea, they are less likely candidates for the user's location. The user is more likely to be at A.

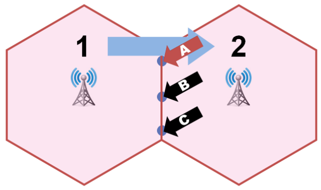
Beyond geography, user travel patterns can be leveraged to estimate position. Consider the example shown in Figure 5(c). If it is known that the user traverses the road running diagonally from top-left to bottom right to go to work, a positioning system may be able to infer that the user's position is B.

Aside from spatial consistency in user behavior, *temporal* consistency can also be exploited, as shown in Figure 5(d). The figure plots a user's usual commute in solid black curves. Now, if the upper route is one that the user uses in the AM, and the lower route in the PM, a cell-ID change from 2 to 1 at 9:00AM would most likely be at position A on the map.

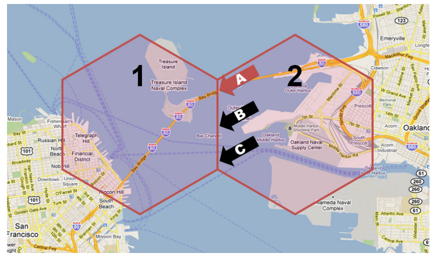
Finally, beyond just a single transition, a *sequence of cell-IDs* has information that can be exploited to estimate position, as shown in Figure 5(e). In the figure, a user's often-used routes are drawn in black solid curves on a map, and a hypothetical cell-ID map is overlayed on top of it. Now, as an example, if the user's mobility results in a sequence of cell-ID changes [4-3-2-1], it is possible to guess that the user is following the upper route, and the user's current position is the point labeled as A. On the other hand, if the user's recently experienced sequence of cell-IDs is [5-2-1], then it is likely that the user is at position C.

Observation 1. For highly mobile users (such as those using public transit systems), we can leverage the sequence of observed cell-ID

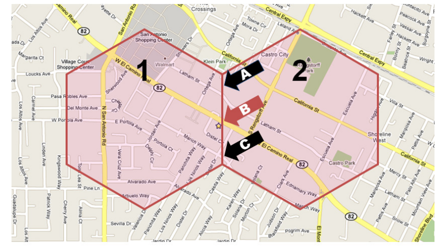
[2] The hexagonal representation of cells is for illustrative purposes only. Cells are irregularly shaped, may overlap, and may enclose one another.



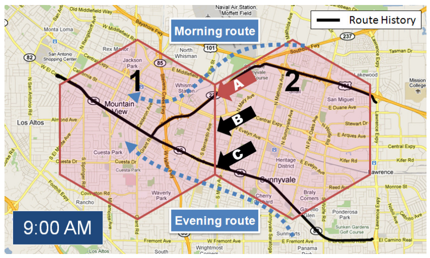
(a) Cell-ID has changed from 1 to 2



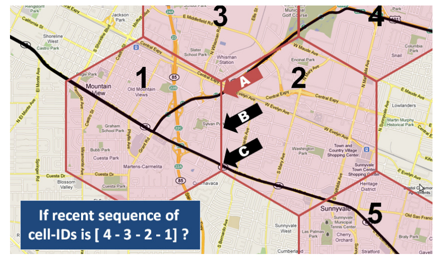
(b) Given underlying physical environment



(c) At a frequently visited area



(d) 2→1 change at different time-of-day



(e) Sequence of cell-ID changes: '4-3-2-1'

Figure 5: Abstract view of cell-ID transitions

transitions, together with the spatio-temporal consistency in user behavior, to *cheaply* estimate positioning.

Reducing Positioning Error. Using cell-ID sequences alone can incur significant error. We illustrate this using results from traces obtained using the software described in Section 2.

Figure 4(a) depicts the cell-ID transition points in one of our traces from the *Los Altos* Route. The black solid curve in the figure plots the GPS reported positions and each flag represents the positions when the cell-ID has changed. One can visually see from this figure that the cell-ID reported by the phone changes approximately after every 500 meters traveled. When driving close to the speed limit of the traveled roads, this translates into cell-ID changes roughly every one to two minutes. Similar results can also observed from other routes in different cities. For example, Figure 4(b) shows the same plot (on a different scale) from the *USC-HSC* route in Los Angeles.

Table 2 summarizes all the results from four different routes (in different cities) used in our experiments. In this table, the third column shows the number of cell-IDs observed during one experimental run, for both distinct cell-IDs and the cell-ID transitions. The fourth column is the average number of cell-IDs observed per kilometer traveled, and the fifth column is the average time spent and distance traveled within each cell-ID. Finally, the last column shows the positioning error of the celltower-based localization, measured relative to the GPS position.

From these results, we conclude that a user experiences cell-ID transitions every 500-600 meters traveled, or around 1$\sim$2 cell-ID transitions per kilometer traveled. These numbers are comparable to the positioning accuracy of the celltower-based triangulation which, as we have argued in Section 2, is inadequate.

Observation 2. Spatial GPS sampling and interpolation *between* cell-ID transition points, can reduce positioning error. Pure interpolation, assuming a certain speed of movement and based on the time elapsed since the last transition can improve positioning accuracy (Figure 6). Better yet, sampling GPS at a few locations within the cell, and then using interpolation, can reduce error even further.

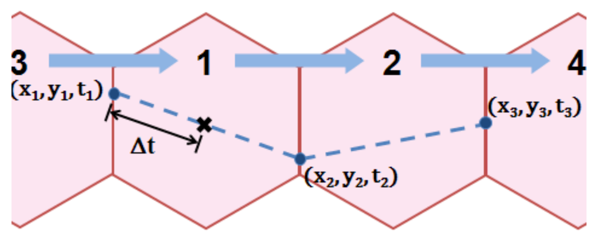


Figure 6: Position estimation using cell-ID transition points and interpolation – simple approach

CAPS leverages these two observations in order to obtain acceptable accuracy and low energy cost, as described in the next section.

4. THE CAPS DESIGN

This section presents the design of CAPS. First, we overview the design features and overall operation of CAPS. We then present a detailed description of key components in CAPS.

4.1 Overview

CAPS is a positioning system that uses cell-ID sequences to efficiently locate a user's position with reasonable accuracy, and its design has the following salient features:

- *Use of spatial and temporal mobility history*: CAPS exploits the fact that people often take similar routes every day and thus experience consistent cell-ID transition points. These points can be used to uniquely represent the user position on those routes. Based on the route history associated with GPS readings, CAPS estimates the user position within the route that she has used in the past.
- *Use of Cell-ID sequence matching*: CAPS adopts a sequence matching technique to efficiently determine a user's posi-

Route	Length	Number of Cell-IDs		Cell-IDs per Kilometer		Avg. per Cell-ID		NET Position Error	
		Distinct	Transitions	Distinct	Transitions	Time Spent	Distance	Median	80 percentile
1: USC-HSC	24.6 km	29	41	1.2 /km	1.7 /km	73.7 sec	598 m	534 m	887 m
2: USC-PC	4.5 km	11	15	2.4 /km	3.3 /km	126.1 sec	301 m	167 m	384 m
3: Sunnyvale	26.6 km	29	36	1.1 /km	1.4 /km	62.9 sec	739 m	341 m	715 m
4: Los Altos	7.1 km	10	13	1.4 /km	1.8 /km	66.3 sec	546 m	347 m	632 m

Table 2: Statistics of observed Cell-ID transitions and Celltower-based localization error for the four different routes used for evaluation.

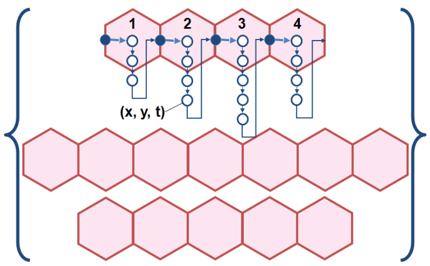


Figure 7: Conceptual view of the database structure.

```
DB SEQ  :  1 2 3 4 5 6 7 8
CURRENT :      9 1 4 5 6
--------------------------
MATCH   :          4 5 6
```

Figure 8: Example of sequence matching

tion. CAPS identifies a cell-ID sequence in the user's history which matches with the current cell-ID sequence. This essentially places the user's position on a route that she has used in the past, thus allowing the positioning system to estimate its position while avoiding the use of energy-intensive GPS.

- *Opportunistic learning*: CAPS opportunistically learns and builds the history of a user's route for future usage. Using a small memory footprint, CAPS maintains the user's past routes and triggers GPS, if necessary.

CAPS consists of three core components — sequence learning, sequence matching and selection, and position estimation — that are intertwined. The sequence learning component populates a local *sequence database* with cell-ID sequences when it detects that the database has insufficient information about the cell-ID that the user currently is in. The sequence matching and selection component match the user's current sequence of recently visited cell-IDs with sequences learned in the database to find the best match sequence. The position estimation component uses the last cell-ID within the matched sequence to interpolate the user's position.

In the following subsections, we will describe each of the three components in detail.

4.2 Sequence Learning

CAPS opportunistically learns cell-ID sequences and stores them in a *sequence database*. This database consists of a collection of cell-ID sequences (Figure 7). Intuitively, each sequence represents a route taken by a user. For each cell-ID in a sequence, CAPS maintains a list of $< position, timestamp >$ tuples, where *position* represents a GPS reading, and *timestamp* is the time at which that reading was taken. This latter information is used to estimate how much time a user spent in a cell in the past, and also for position interpolation (discussed in Section 4.4).

Learning is triggered by two conditions: either when a user sees a cell-ID which is not present in the database, or when the user has stayed in the current cell far longer than during previous visits. If either condition is satisfied, CAPS turns GPS on, and reads both GPS readings and cell-IDs. Specifically, whenever CAPS obtains a GPS reading, it inserts this $< position, timestamp >$ tuple to a list associated with the *current cell-ID*. Whenever there is a change in the cell-ID, CAPS inserts the previous cell-ID to a *learning cell-ID sequence*.

Then, whenever the length of the learning cell-ID sequence exceeds a certain threshold (e.g., 20), it inserts this sequence into the database and starts a new learning cell-ID sequence. When doing this, the new learning sequence is pre-loaded with the final 4 cell-IDs of the previous learning sequence; this ensures continuity in sequence matching. For example, assume that the length threshold is 10 and current learning sequence is [1-2-3-4-5-6-7-8-9-10]. When the 11th cell-ID is encountered, CAPS inserts [1-2-3-4-5-6-7-8-9-10] into the database, and starts a new sequence with [7-8-9-10-11]. Thus, any sub-sequence of [1-2-3-4-5-6-7-8-9-10-11] with length 5 will fully match with the database without discontinuity (we explain the choice of length 5 in Section 4.3). This algorithm ensures responsiveness — recent history is entered into the database quickly enough so that it can be used soon.

While learning cell-ID sequences, CAPS also attempts to estimate the phone's current position in the same way as when GPS is turned off using the algorithms described in Section 4.4. If a "good" position estimate is found (defined later), it turns off GPS and terminates the learning.

Learning is also enabled when the phone or GPS has been turned off for a significant amount of time (e.g. 20 minutes). In this case, CAPS starts off with an empty learning cell-ID sequence.

4.3 Sequence Matching and Selection

CAPS continuously tracks the cell-ID, and maintains a sequence of recently-visited cell-IDs as a *current cell-ID sequence*. It *matches* this sequence against the sequence database, to come up with one or more matching sequences. It then *selects* the best match sequence.

Matching. CAPS uses modified Smith-Waterman [26] algorithm for cell-ID sequence matching. This is a well-known algorithm for performing local sequence alignment; that is, for determining similar regions between two nucleotide or protein sequences in bio-informatics [26]. Instead of looking at the entire sequence, the Smith-Waterman algorithm compares segments of all possible lengths and optimizes a similarity measure. It uses dynamic programming, and guarantees the optimal local alignment with respect to a scoring system (how to weight the value of match, mismatch, and gaps in a sequence) being used. Using this algorithm, CAPS determines whether there is any match between the sequence of recently-visited cell-IDs and the sequence database.

We have modified the Smith-Waterman algorithm for use in CAPS.

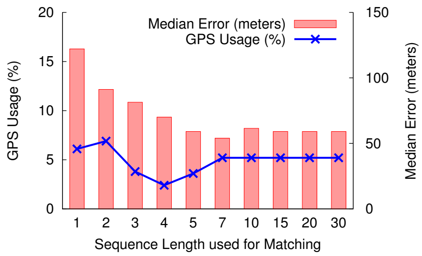


Figure 9: Parameter selection experiment — length of the current cell-ID sequence used for matching.

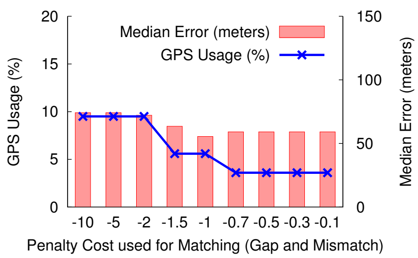


Figure 10: Parameter selection experiment — penalty cost of gaps and mismatches in matching.

CURRENT SEQ	1 2 3 4 5	Match	Gap	Mismatch	Score
DB SEQ 1	1 7 3 4 5	4	0	1	3.5
(match)	1 X 3 4 5				
DB SEQ 2	1 2 3 6 4 5	5	1	0	**<u>4.5</u>**
(match)	1 2 3 – 4 5				
DB SEQ 3	6 7 2 3 4 5 8	4	0	0	4.0
(match)	2 3 4 5				
DB SEQ 4	6 1 3 7 5 9	3	1	1	2.0
(match)	1 – 3 X 5				

Figure 11: Examples of sequence matching against database.

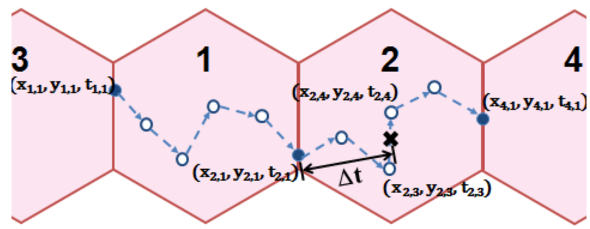


Figure 12: CAPS Position Estimation Example

We have added a constraint that ensures that the last cell-ID of the recently-visited cell-ID sequence must be in the matched subset; this ensures the correct semantics of a match. Second, we use a penalty cost of -0.5 for gaps and mismatches, so we penalize any gaps or substitutions in cell-ID sequence matching with score of -0.5 (a matched cell-ID is given a score of 1.0). We motivate this parameter choice below. The output of the sequence matching algorithm contains, for each match in the database, the match score, the actual sequence matched, and the length of the match. Figure 8 shows an example of a sequence matching where the current cell-ID sequence [9-1-4-5-6] is matched with a sequence [1-2-3-4-5-6-7-8] in the database.

Two parameters are critical to the performance of this sequence matching algorithm. The first is the length of the recently-visited cell-ID sequence used for matching against the database, which is set to 5 in our experiments. The second is the penalty cost used for gaps and mismatches in finding a matching with best score, and CAPS sets its value to -0.5.

The length of the recently-visited cell-ID sequence determines how far back CAPS looks to find a matched sequence in the database. If the length is too long, CAPS unnecessarily searches too far in the history to determine a user's current route, and risks conflating two distinct routes. On the other hand, if the length is too short, CAPS would not be able to leverage the structure of the route.

To find a sweet spot, we have conducted extensive trace-driven simulations with various length values and examined the accuracy and energy performance. Figure 9 plots the result for the HSC route. As shown in the figure, the accuracy performance is best when the length is 7, and the energy cost is minimum when the length is 4. Simulation on other routes also exhibited qualitatively similar results. There is a tradeoff between energy and accuracy in that range: using a longer sequence increases the chance of a match (see the mismatch penalty discussion below) and GPS activation is avoided, but the match may be incorrect for reasons we have discussed above. To balance both sides, we have selected an intermediate value, 5, as the length used in our implementation.

We have also performed a similar evaluation to decide the mismatch penalty, and Figure 10 plots the results. This figure shows that it is better to penalize the gaps and mismatches with a cost less than the benefit given to a match. That is, for example, when matching current sequence [1-2-3-4-5] against the database, CAPS will prefer [1-2-9-4-5] over [3-4-5] since their scores are [4 - 0.5 = 3.5] vs [3]. More examples of matching with gaps and mismatches are shown in Figure 11. However, the performance was relatively insensitive to a choice of mismatch penalty between -0.1 to -0.9. We have used -0.5 as the penalty cost whereas the original Smith-Waterman algorithm used -1.

Sequence Selection. After running the sequence matching algorithm on all routes in the database, CAPS may have more than one matched sequence with a positive match score. CAPS selects the match with the highest score. If there are multiple matches with the highest score, it uses the following tie-break rules, in order. First, select the match whose time-of-day is close (within an hour before or after the current time), exploiting the user's temporal consistency. If that fails, select the longer sequence. If that fails also, match the tail of current sequence (where a user is now) to head of a sequence in the database so that CAPS can predict what will come next. This last rule helps in finding a match when the current cell-ID sequence is very short, which happens when the phone is turned on after being off, or when the user comes out of an area where GPS was not available for a long time (e.g. indoors). Finally, if none of above tie-breaking rules were able to decide on a match, CAPS simply prefers a newer sequence to older ones. If no valid match is found, CAPS turns on GPS since it is not possible to estimate current position.

4.4 Position Estimation

After selecting the best match sequence, CAPS uses temporal interpolation to estimate position. Specifically, given the selected sequence, and the time elapsed since a user moved into the current

Route	Length	Moving Time	Stop Time	Velocity		Inter-sample Dist.		Avg. Sampling Interval	Mobility Type
				Avg.	Max.	Avg.	Max.		
1: USC-HSC	15.2 miles	51 min	60 min	17.7 mph	85.7 mph	20.1 m	78.6 m	2.6 sec	bus
2: USC-PC	2.8 miles	31 min	10 min	5.3 mph	40.3 mph	6.1 m	74.0 m	2.6 sec	bus
3: Sunnyvale	16.5 miles	37 min	20 min	26.4 mph	93.4 mph	34.5 m	95.0 m	2.5 sec	car
4: Los Altos	4.4 miles	15 min	0 min	19.0 mph	56.2 mph	21.6 m	50.2 m	2.4 sec	car

Table 3: Four different travel routes used for evaluation.

cell-ID, CAPS uses the history of corresponding GPS points that are associated with the cell-ID and estimates a current position.

Figure 12 depicts an example of this position estimation procedure. In this figure, hexagons represent cell-ID boundaries, solid circles are the cell-ID transition points in a cell-ID sequence [3-1-2-4], and donut-shaped circles within each cell are additional GPS points associated with that cell-ID. Assume that a user's current position (marked as cross) is within cell-ID 2 and her current cell-ID sequence is [3-1-2]. Also, assume CAPS selects the sequence [3-1-2-4] from the database. Now, if the elapsed time, denoted as Δt, since the user's entrance to the current cell-ID is within the range of two measurement points (i.e., $t_{2,3}, t_{2,4}$)) in the history — $(t_{2,3} - t_{2,1}) \geq \Delta t \geq (t_{2,4} - t_{2,1})$— then CAPS performs an interpolation as follows to estimate the current position.

$$(x_{est}, y_{est}) = \left(x_{2,3} + (x_{2,4} - x_{2,3}) \cdot \frac{\Delta t - (t_{2,3} - t_{2,1})}{t_{2,4} - t_{2,3}} \right.$$
$$\left. , \qquad y_{2,3} + (y_{2,4} - y_{2,3}) \cdot \frac{\Delta t - (t_{2,3} - t_{2,1})}{t_{2,4} - t_{2,3}} \right)$$

If the time spent in the current cell-ID since the entrance into that cell-ID far exceeds the departure time of that cell-ID from the history, then CAPS turns GPS on.[3] This is required since the user's behavior has deviated from the history that CAPS has selected to perform estimation on; CAPS would either need to learn new behavior of the user or find a better matching sequence.

However, the cell residence time may exceeds the expected residence time by small values because of small variations in velocity. For values of this difference below 90 seconds, CAPS continues to perform estimation by "looking ahead" into the predicted next cell-ID (cell-ID 4 in the figure), That is, Δt is still calculated as $t_x - t_{2,1}$ even though the expected departure time of cell-ID 2 has passed, and the GPS points within the cell-ID 4 are used as if they were points within cell-ID 2 even though cell-ID 4 has not been observed yet. When cell-ID 4 is finally reached, CAPS will perform a new sequence matching and "rewind" to the start location (in its history) for cell-ID 4. Ideally, if there are sufficient number of GPS points within each cell, this lookahead and rewind may not be necessary. However, sometimes GPS reception is bad (e.g. in urban canyons) and there could be an insufficient number of GPS points within each cell-ID. In this case, lookahead provides a better interpolation baseline between the last GPS point of the previous cell-ID to the first point of the next expected cell-ID. Once the next cell-ID is known, rewinding gets us to a last known good state from which estimation can continue.

Finally, when GPS is on, CAPS returns the GPS reading, and uses its position estimate to monitor the accuracy of its estimation algorithm. CAPS decides to turn GPS off again when its estimation error falls to within a configurable tolerance limit (80m in our implementation).

[3]We use 90 seconds as the threshold based on the observation in Table 2; the user stays within a cell-ID for average of 60∼120 seconds.

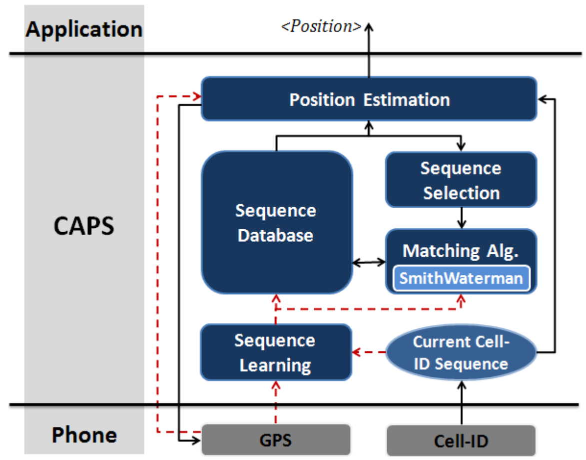


Figure 14: CAPS implementation block diagram

5. PERFORMANCE EVALUATION

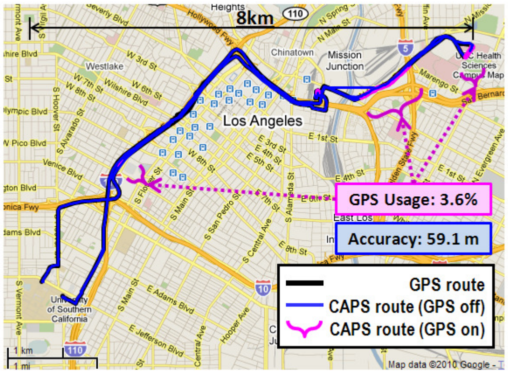
In this section, we present evaluation results from real world experiments using a complete implementation of CAPS on a smartphone.

5.1 Implementation and Methodology

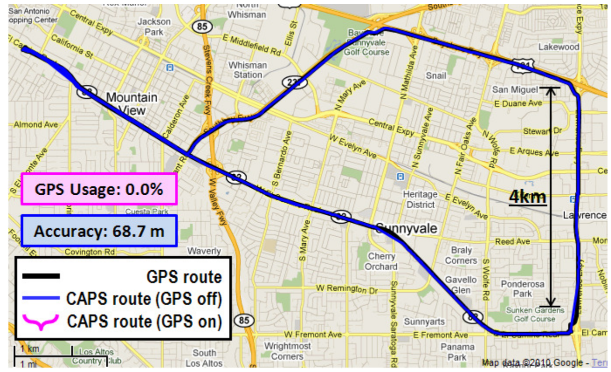
We have implemented CAPS in Java using the Android API for the Android smartphones. Figure 14 shows the block diagram of the CAPS software implementation. The Cell-ID is obtained from the phone, and CAPS runs its algorithms to provide the smartphone application with a position estimate whenever requested. CAPS may turn the GPS on and off depending on its estimates. Control flow occurs on the gray dotted line only when the GPS is turned on, upon which sequence learning begins and the position is obtained directly from GPS.

CAPS has been tested to work on Android OS versions 1.5, 1.6, 2.1, and 2.2. Conceptually, our algorithm is platform-independent and can work on any GPS-equipped smartphone that exports cell-ID information. In most of our experiments, we have used the Nexus One Android smartphone on a T-Mobile GSM network. In Section 5.5, we present results for experiments done with Nexus One phones on AT&T's GSM network, and also with MotoDroid phones on Verizon's CDMA network.

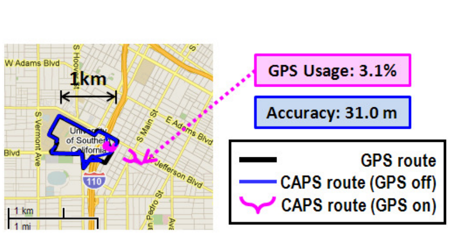
We have experimented with our CAPS implementation on four different routes, with qualitatively different characteristics, in and around the cities of Los Altos, Sunnyvale, and Los Angeles in California. Table 3 summarizes the characteristics of the routes used for the experiments. The columns of this table show, in order: the label for the route, the trip length, the time spent moving, the time that the user was stationary at the destination before returning to the starting point, the instantaneous velocity between two consecutive sample points, the distance between two consecutive samples, and the average time between two consecutive samples. Travel distances ranged from 2.8 miles to 16.5 miles, and the average velocity



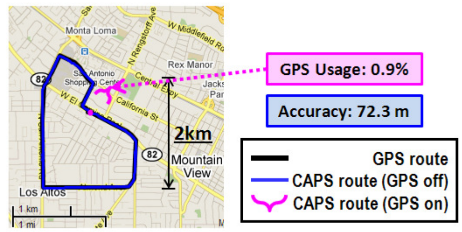
(a) HSC Route



(b) Sunnyvale Route



(c) PC Route



(d) Los Altos Route

Figure 13: CAPS experiment result for all four routes at *same scale*.

ranged from 5.5 mph to 26.4 mph, with occasional instantaneous speeds of over 90 mph. Although the software was programmed to collect logs every 2 seconds, the inter-sample time was about 2.5 seconds due to occasional software delays. Each round of our actual experiment ran for approximately 15 minutes for the shortest route, and 51 minutes for the longest one [4].

We have used two main evaluation metrics to show the performance of CAPS. *GPS Usage* represents the percentage of time that the GPS was turned on. *Positioning Accuracy* measured as the median distance between the estimated position and the position reported by always-on GPS. We seeded each phone with six iterations worth of prior GPS history. We conducted several runs, and report the result of subset of these; other runs show qualitatively similar results.

5.2 Energy Savings

Figure 13 depicts the CAPS experiment results on *equally scaled maps* for our four routes. It shows the GPS trace using a thicker black curve, and the CAPS trace with thinner gray curves, where the CAPS trace is drawn with two different shades of gray for the cases when GPS is off and GPS is on. The figures show that CAPS closely matches, in fact almost overlaps completely, with GPS while turning on the GPS only for a very small amount of time. Indeed, the two traces are indistinguishable in the figures. For example, on the HSC route (Figure 13(a)), GPS was turned on only 3.6% of the time compared to the always-on GPS, with the median positioning error of only 59.1 meters. On the Sunnyvale route (Figure 13(b)), CAPS closely estimated the GPS trace with median error of 68.7 meters *without using GPS at all*.

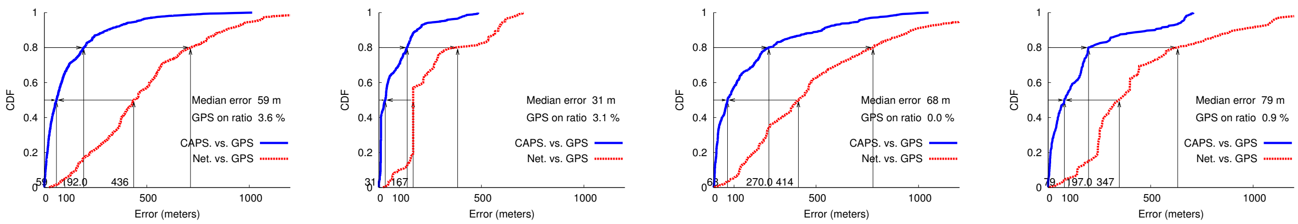
[4] For USC-HSC route only, there was no log during 60 minutes when the user was stationary at the destination since he went into a building where GPS was not available. For all other routes, the logs were continuous throughout.

To compare CAPS with celltower-based localization, consider Figure 15, which compares the cumulative distributions of the error for both CAPS and the celltower-based localization for all four routes that we have experimented on. It shows that CAPS improves positioning accuracy significantly; in all of the four subfigures, CAPS' CDF is distinctly to the left of the NET CDF.

Table 4 summarizes the main results. CAPS achieves median positioning error of below 75 meters while turning on the GPS less than 4% of the time compared to the always-on GPS across all four routes. Considering that the width of a road can often be as large as 20 meters, and also the fact that user can move more than 20 meters within the sampling interval (Table 3), we believe that positioning error of 50$\sim$80 meters is reasonably accurate. Furthermore, the error of CAPS is only 20% of that of the NET scheme, a significant improvement over the NET scheme with minimal GPS usage. CAPS' positioning errors are "on-route": errors occur mostly because CAPS incorrectly estimates where the user is *on* the route, rather than off it. These errors are introduced by temporal interpolation, since a user may encounter different levels of traffic or different stop light durations on a trip, relative to prior trips stored in the sequence database. Overall, our results clearly demonstrate that CAPS is able to achieve reasonable positioning accuracy while spending minimal energy.

Route	CAPS		NET	CAPS/ NET
	GPS Usage	Med.Error	Med.Error	Error Ratio
USC-HSC	3.6 %	59.1 m	534 m	11.1 %
USC-PC	3.1 %	31.0 m	167 m	18.5 %
Sunnyvale	0.0 %	68.7 m	341 m	20.1 %
Los Altos	0.9 %	72.3 m	347 m	20.8 %

Table 4: Summary of the CAPS experiment results and their comparison to NET error for the four different routes.



(a) HSC Route (b) PC Route (c) Sunnyvale Route (d) Los Altos Route

Figure 15: CDF of the positioning error of CAPS (vs. celltower-based) for experiment on Los Altos Route.

Route	CAPS		Periodic GPS 4%	
			w/o extrapolation	w extrapolation
	GPS Usage	Med.Error	Med.Error	Med.Error
USC-HSC	3.6 %	59.1 m	166.1 m	109.8 m
USC-PC	3.1 %	31.0 m	14.5 m	24.0 m
Sunnyvale	0.0 %	68.7 m	323.6 m	174.6 m
Los Altos	0.9 %	72.3 m	208.9 m	197.0 m

Table 5: Summary of the positioning error of fixed-interval periodic GPS (with and without extrapolation for position estimation) with comparable GPS usage (On ratio of 4%) as CAPS.

5.3 Comparison with Periodic GPS Activation

In this subsection, we investigate how CAPS compares with a scenario where GPS is periodically activated with fixed interval. The simplest approach to trade-off accuracy for energy in GPS-based location services is to periodically duty-cycle GPS. However, it has been shown in [24] that periodic duty-cycling of GPS at a fixed interval may introduce significant, potentially unbounded, error without necessarily providing significant energy benefits because periodic duty-cycles are oblivious to the actual mobility. We have conducted experiments to see how CAPS compares with the periodic GPS in terms of energy-efficiency and positioning error. We also contrast both schemes with celltower-based localization.

To conduct this experiment, we used one phone that was programmed to collect a GPS reading as well as a celltower-based positioning reading every 2 seconds. We then sub-sample the GPS readings to simulate different periodic rates. To compare the positioning accuracy, we use two methods to calculate the error of the periodic GPS scheme. The first one is a naive approach wherein the estimate of the current position is set as the most recent GPS reading. The second method uses an extrapolation approach where the estimate of the current position is taken as extrapolation of the two previous GPS readings based on the time passed since the last reading.

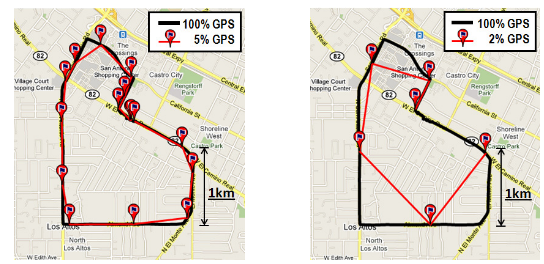
Figure 16(a) plots the GPS trace for the Los Altos Route when GPS was sub-sampled at 5%. The route reported by this 5% GPS looks visually similar to that reported by CAPS (Figure 13(d)). However, CAPS achieved 0.9% GPS activation rate in our evaluation which is far less than 5%. Comparing with a lower rate GPS, for example 2% GPS in Figure 16(b), it is clear that the periodic GPS with comparable activation rate has much higher inaccuracy. The same holds for the Sunnyvale Route (Figure 17(a)).

In terms of positioning error, periodic GPS with 5% activation ratio results in as coarse-grained positioning as celltower-based localization, far below the quality of CAPS. Furthermore, the extrapolation method for estimating the current position using periodic GPS readings does not improve accuracy much as visually shown in Figure 18. In fact, it provides position estimates that are often off-route (unlike CAPS) due to the nature of extrapolation.

Table 5 summarizes the comparison between CAPS and periodic GPS with comparable GPS usage of 4% for all four different routes. While CAPS keeps the median positioning error below 75 meters and GPS usage ratio below 4%, periodic GPS with 4% duty-cycling has significantly higher positioning error, as high as 300 meters, except for one route. Even if extrapolation is used for position estimation, the accuracy is not as good as CAPS. The reason that error of periodic GPS is pretty good for the *USC-PC* route is because the route itself is very short, restricted to a small region as shown in Figure 17(b), and was traversed at low speeds. Thus, the magnitudes of the position errors are small.

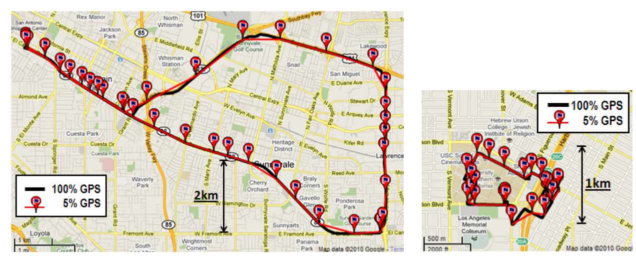
The results in this subsection clearly shows the benefits of CAPS; it has better energy saving over periodic GPS invocation scheme with comparable accuracy, and better accuracy compared to periodic GPS invocation scheme with comparable energy cost.

Finally, we have not compared CAPS with RAPS [24] for a simple reason. The latter was designed explicitly for pedestrian use and those components which estimate mobility (its space-time history and its accelerometer-based measurement of activity ratio) were optimized for slower speeds than the ones we have explored in this paper. As a result, we believe that RAPS would exhibit much lower energy-efficiency in order to achieve comparable positioning error to CAPS.



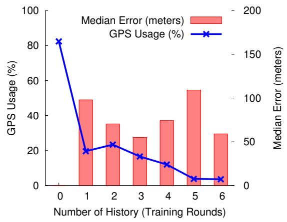
(a) Periodic GPS 5% (b) Periodic GPS 2%

Figure 16: Periodic GPS, 5% and 2%, on Los Altos Route

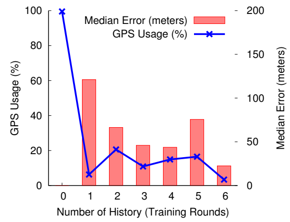


(a) Sunnyvale Route (b) PC Route

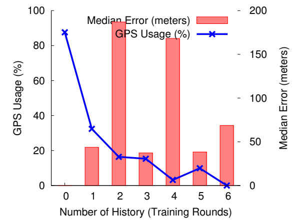
Figure 17: Periodic GPS 5%



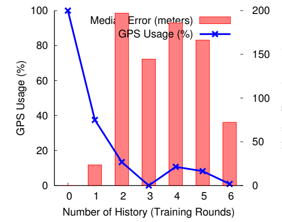
(a) HSC Route



(b) PC Route



(c) Sunnyvale Route



(d) Los Altos Route

Figure 19: Learning performance of CAPS: As the number of history logs (training rounds) increases, GPS Usage (On ratio) decreases while keeping the median positioning error around 100m.

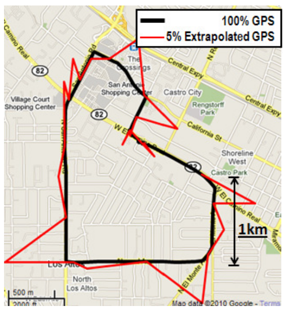


Figure 18: Periodic GPS 5% with extrapolation, Los Altos Route

5.4 Learning of CAPS

In this section, we investigate the impact of varying the training set (measured by the number of history logs) given to CAPS for learning to the performance of CAPS. For this experiment, we used one phone and ran seven iterations of CAPS experiment on the same route with different training set sizes ranging from zero to 6.

Figure 19 plots the relationship of the training set size to the error and GPS usage performance of CAPS. There are several interesting things to note from this result. First, when no (zero) training data was used, the error is close to zero since CAPS keeps the GPS on almost all the time. Second, as the training set size increases, the GPS usage decreases. This is the expected behavior; as CAPS learns more about the history of cell-IDs and GPS coordinates, it requires less use of GPS to estimate its current position. Third, the median error may increase during the learning phase even though the GPS usage decreases. This is also not surprising since CAPS gets less help from GPS and may not have learned enough to provide accurate estimate without GPS. Most interestingly, by the time CAPS has reached the seventh iteration, the GPS usage has already reduced to below 4%, a point which is already extremely low and where there is not much room for further reduction. This in turn means that the median error will not increase further even if the learning progresses further, but indicates that with about a week's worth of training, many commuters can hope to get reasonably accurate positioning with extremely low energy-usage.

5.5 Platform and Carrier Independence

So far, all the results we have discussed have used *Nexus One* phones on the *T-Mobile* network. However, we have evaluated CAPS on other platforms and carriers also. Specifically, we have conducted experiments on the *AT&T* network using *Galaxy S* phones, and also for *MotoDroid* phones on the *Verizon* network. The purpose of this experiment is to show that CAPS is platform and network independent. Table 6 summarizes the results for all platform-carrier combinations that we have experimented with. It shows that the accuracy of the NET scheme, cell-ID density, the error and GPS usage performance of CAPS, and their comparison to the NET error are all qualitatively similar. Thus, the main observation from these results is that CAPS performs well, regardless of the network carrier or the platform.

5.6 Comparison with a WiFi-based Positioning System

We also compare the accuracy of CAPS with that of a WiFi-based positioning system. Dense urban areas are expected to have a high density of WiFi APs, and prior work on WiFi-based positioning has shown promising results [19, 4]. We compare the performance of CAPS against a commercial WiFi-based positioning system, WPS, from Skyhook Wireless [5]. It determines location based on Skyhook's database of known WiFi access points. We implemented a tool that collects location readings from WPS every 2 seconds using Nexus One Android phones and conducted experiments in Los Angeles downtown area.

The key advantage of WPS is that it provides position information in urban areas and indoors, when information for those areas exists in their database. However, since it needs to hear beacons from at least 3 known WiFi APs before fixing a location, it does not work well when a mobile moves fast (i.e. driving) or where the deployment of WiFi APs is sparse. In addition, WPS consumes energy comparable to GPS when location information is requested frequently (e.g. every 2 seconds in our experiment). This is because of the high cost of WiFi scanning as well as communication overhead with the Skyhook database server.

Table 7 summarizes WPS results for the USC-HSC and USC-PC routes in Los Angeles. Each result is averaged over 5 experiments on each route. There are several observations to make from this table. First, the median positioning error of WPS is over 300 meters

[5] Skyhook Wireless, http://www.skyhookwireless.com/

Route:	**USC-HSC**	**USC-PC**
Median Error	339 m	323 m
WPS Availability	55 %	76 %
WPS ERROR	—	—
+ *LOCATION_CANNOT_BE_DETERMINED*	21 %	19 %
+ *NO_WIFI_IN_RANGE*	24 %	5 %
Median Error Only When Available	176 m	322 m
WPS Accuracy > 500 m	68 %	79 %
Error > 5000 m	1.2 %	0.4 %
Median Error Only When Available, FILTERED Out Error > 5000m	67 m	124 m

Table 7: Summary of the WPS experiment results.

Phone / Network Carrier		CAPS		NET	CAPS/ NET	Cell-ID Transition	
		GPS Usage	Med.Error	Med.Error	Ratio	per kilometer	Distance
Nexus One	**T-Mobile (GSM)**	3.6 %	59.1 m	534 m	11.1 %	1.7 /km	597 m
Nexus One	**AT&T (GSM)**	2.2 %	98.8 m	462 m	21.3 %	1.8 /km	552 m
MotoDroid	**Verizon (CDMA)**	9.3 %	85.8 m	609 m	14.1 %	1.6 /km	643 m
Galaxy S	**T-Mobile (GSM)**	7.5 %	101.7 m	856 m	11.8 %	0.6 /km	1644 m

Table 6: Summary of the CAPS experiment results on USC-HSC route for various network providers and phone types.

for both routes. This is far greater than what has been reported in the literature [19, 4]. When we looked further into the source of error, we found that WPS returned a location for only 55% and 76% of the experiment time on each route, respectively. During the remaining 45% and 24% of the time, respectively, WPS failed reporting one of two errors, *LOCATION_CANNOT_BE_DETERMINED* and *NO_WIFI_IN_RANGE*. The main reason for this, we believe, is because we were moving in a vehicle at a speed that WPS was not designed for.

We noticed two more things in our experimental data; 1) WPS reported a horizontal accuracy greater than 500 meters for around 70% of its readings, which means that it was not very confident about its location information, and 2) there were readings (1.2% and 0.4% for each route, respectively) whose errors were greater than 5 kilometers!! The erroneous readings included locations in Long Beach ($\sim$45 km from our experiment site), Texas ($\sim$2400 km), New Orleans ($\sim$3000 km), and Washington DC ($\sim$4000 km). We do not know whether this is due to attack on WPS [28], or legitimate move of a personal AP, or a software bug in WPS library. We filtered out those readings whose locations were more than 5km away from our experiment site, after which the median error of WPS, after filtering, becomes 67 meters and 124 meters for USC-HSC route and USC-PC route, respectively. Thus, we conclude that the accuracy performance of WPS similar to that of CAPS in some cases, but can incur high energy cost for continuously scanning WiFi access points.

5.7 Effects of Time-of-Day for Sequence Selection

As described in Section 4.3, CAPS prefers to select the cell-ID sequence whose time-of-day is close to the current time to exploit a user's temporal consistency. This raises two questions; 1) is the use of time-of-day information for sequence selection actually beneficial? and 2) what if the user traverses her regular route at a different time-of-day? To answer these questions, we performed trace-based simulation. Here, we used the trace data used in Section 5.2 but adjusted timestamps. Specifically, all the Los Altos training set's timestamps were shifted to the morning, and the Sunnyvale training data's timestamps were shifted to the afternoon. Then, two versions of same test data were used on each route, one in the morning and one in the afternoon, for cross validation. Note that we used the data from the Los Altos and Sunnyvale routes because their routes overlap and this can confuse position estimation, as shown below.

Table 8 summarizes the simulation results. CAPS performs the best (i) when time-of-day information is used for sequence selection and (ii) when the test data has the same time-of-day as the training data. When the test is being done at a different time-of-day from the training data, or if the time-of-day information is not used for the sequence selection, then the median error increases. Time-of-day is helpful in disambiguating routes that overlap with each other; without that information, CAPS position estimation can give incorrect readings until a user encounters a new cell-ID. Figure 20 illustrates such a scenario. The right-middle part of the *Los Altos* route overlaps with the top-left part of *Sunnyvale* Route. In this example, even though the user has taken a right-turn (left figure, *Los Altos* route), CAPS may estimate its position as though the user went straight down the road, following the *Sunnyvale* route. At that point, CAPS may not be able to distinguish these two routes until it encounters the next cell-ID or spends more time[6] in the current cell than expected, since the user has started his journey from an overlapping part of the cell-ID sequence. When time-of-day information is used, the temporal similarity tie-break rule would prevent this inaccuracy. Finally, the reason the error on the Los Altos route looks more significant than that on the Sunnyvale route is because the Sunnyvale route is much longer relative to the overlapped portion of the route, and thus the error gets amortized over the whole route.

Route:		Los Altos	Sunnyvale
Time-of-Day Alg.	**Time of test data**	Med.Error	Med.Error
WITH time-of-day	Matching	72.3 m	68.7 m
	Different	135.2 m	69.0 m
WITHOUT time-of-day	Matching	135.2 m	67.1 m
	Different	135.2 m	67.1 m

Table 8: Summary of the time-of-day simulation results.

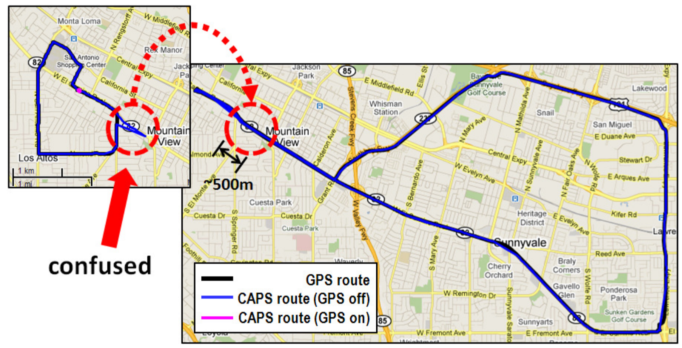


Figure 20: CAPS may confuse two routes.

6. DISCUSSION

In this section, we discuss a few details of CAPS that are worth mentioning.

CAPS is designed mainly for large-scale movements of mobile devices (e.g. during commutes). When users spend time indoors where GPS is not available, we envision the use of other lightweight positioning systems. For example, RAPS [24] proposed a celltower-RSS blacklisting technique to avoid turning on the GPS indoors, saving energy. When integrated with RAPS, entry and exit from locations without GPS availability can be modeled as "cell-ID" transitions in CAPS.

CAPS has been evaluated only in urban areas where cell-tower density is high. How CAPS will perform in rural settings with

[6] 90 seconds in our parameter settings.

lower cell-tower density is an open question. Accuracy may or may not suffer: intuitively, a lower cell-tower density may result in fewer cell-ID sequences and greater uncertainty within each cell-ID, therefore lower accuracy. But at the same time, users generally move larger distances in rural areas since the density of buildings and other waypoints is also less, and the road network is also less complex, so it may not matter that there are fewer cell-ID sequences to train on. Which of these factors impacts performance can only be determined by experimentation, and we have left this to future work.

CAPS did not make use of the underlying geography in this paper. Although human-inaccessible areas are implicitly avoided in CAPS' history, it would be interesting to evaluate the benefit of using geography: CAPS may potentially benefit from, for example, map-matching [6, 14]. We have left this to future work as well.

In its current form, CAPS cannot disambiguate physically distinct routes that produce identical cell-ID sequences. As long as spatial divergence in routes is significant enough to pass through at least one distinct cell, CAPS will work well. However, when a user diverges from her route slightly without impacting the cell-ID sequence, CAPS' accuracy may suffer. In these cases, techniques that incorporate other environmental cues [2, 6, 24] may be required to deal with these deviations.

Prior work has investigated whether the accuracy of cell-tower localization can be improved by other methods, such as using cell-tower signal strength. These approaches rely on knowledge of cell-tower locations and signal strength information from multiple cell-towers, using which an accuracy of 70–200 meters is achievable just by using the centroid method, and even better accuracy can be achieved using fine-grained fingerprinting [29, 3]. However, commodity smartphones do not export celltower locations and signal strength from multiple celltowers to the application programmer. CAPS is pragmatic in its reliance on current cell-ID information.

Managing the sequence database may be an important issue for CAPS. The size of the database file for our experiment, including all four routes with six rounds of history for each route, is around 780kB without any optimization. If the user traverses new routes often and thus the number of routes increases, this size may grow substantially. Furthermore, the cell-IDs might change over time, due to network failure or upgrades. To overcome these challenges, it might be possible to keep the database on the cloud, crowd-source its collection, have the cloud service age entries appropriately [1], and dynamically download the relevant parts of the database.

7. RELATED WORK

Many techniques have been proposed to improve energy efficiency in positioning systems. RAPS [24] trades-off location accuracy for reduced energy use. It uses a combination of spatio-temporal location history, user activity, and celltower-RSS blacklisting to selectively activate GPS only when necessary to reduce position uncertainty. It also proposes sharing position readings among nearby devices using Bluetooth in order to further reduce GPS activation. However, RAPS is mainly designed for pedestrian use, and a significant portion of the energy savings come from avoiding GPS activation when it is likely to be unavailable. On the other hand, CAPS is designed for higher user mobility, and is able to provide continuous positioning without turning on the GPS as long as sufficient location history exists.

Other pieces of work use different cues to determine when to activate GPS. EnTracked [17] attempts to duty-cycle location determination. Unlike CAPS, it schedules position updates by using accelerometer-based mobility estimation to save energy. You et al. [31] propose a signal-strength based indoor localization scheme that adapts a sampling rate to a target's estimated mobility level for energy-efficient operation. However, unlike CAPS, their scheme is tailored only for RF-based indoor localization. Farrell et al. [11] present an algorithm that determines, based on a positioning uncertainty model, when to perform GPS updates for energy-efficient positioning. However, they simply use a maximum walking velocity for uncertainty calculation, and the model has been evaluated only in simulation. Deblauwe and Treu [8] propose GSM signature-based triggering to avoid the unnecessary triggering of GPS. They use the difference between the device's current GSM measurements and the ones taken the last time GPS was switched on. However, their approach is designed to detect entering and leaving of a zone, and does not estimate user position while GPS is off nor does it attempt to turn GPS off outside of the zone. Also, it assumes that celltower RSS information from multiple celltowers are available on the smartphone, which current APIs do not support.

Several researchers have explored the energy-accuracy tradeoff by exploiting characteristics of different positioning methods. EnLoc [7] uses dynamic programming to find the optimal localization accuracy for a given energy budget and decides which one of GPS, WiFi, or GSM localization methods to use. Micro-Blog [13] exploits the accuracy-energy tradeoff of GPS, WiFi, and GSM based localization for energy-aware localization. Specifically, depending on the accuracy requirement of the application, it uses a lower-energy method over a higher-energy method when possible. A-Loc [21] proposes to dynamically trade-off location accuracy and energy use, based on probabilistic models of user location and sensing errors. These models are used to choose the best among different localization methods and tune energy expenditure to meet location accuracy requirements specified by applications. In contrast, CAPS achieves high energy-efficiency and reasonable accuracy by leveraging freely-available cell-ID sequences obtained from a history of user's mobility.

Recently, positioning systems that make use of history have been proposed. StarTrack [15] continuously tracks a mobile user for smartphone applications. It uses a track abstraction and employs similarity matching algorithms to find mobile users who share same routes. CAPS is different from StarTrack in using cell-ID sequences to determine a user's position. Escort [5] is a positioning system that obtains cues from social encounters and leverages an audio beacon infrastructure. CAPS uses a user's personal history of routes to determine a position and does not require any additional infrastructure support.

Finally, in [20] a variant of Smith-Waterman algorithm is used to assess the similarity between possible trajectories of best connected nodes along the path of a mobile sink in a sensor network. The trajectories are hierarchically-clustered into characteristic mobility pattern profiles. The intent is to identify suitable relay nodes along the predicted path of the mobile node to stash data to be delivered there till the mobile node passes by. In contrast, CAPS uses sequence matching with historical cell-ID sequences to predict current location of a smartphone in a cellular network.

8. CONCLUSIONS

In this paper, we presented CAPS, a cell-ID aided positioning system for smartphone applications. It is based on the observation that users exhibit consistency in their everyday routes and the cell-ID sequence that the user experiences can often provide an accurate estimate of the user position. By monitoring the cell-ID transitions and using a history of GPS readings obtained within a cell, CAPS efficiently estimates a user's current position, with reasonable accuracy, using its cell-ID sequence matching technique. We have evaluated CAPS through real-world experiments using a prototype implementation, at different locations, and on different platforms, and wireless operators, and show that CAPS reduces GPS on-time by more than 90% relative to the case when GPS is always-on, while still providing accurate position information.

9. REFERENCES

[1] G. Ananthanarayanan, M. Haridasan, I. Mohomed, D. Terry, and C. A. Thekkath. Startrack: a framework for enabling track-based applications. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, 2009.

[2] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. 2009.

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

[4] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *Proceedings of 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys'05)*, Seattle, Washington, June 2005.

[5] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. "Did You See Bob?": Human Localization using Mobile Phones. In *MobiCom '10: Proceedings of the 16th annual international conference on Mobile computing and networking*, pages 149–160, 2010.

[6] I. Constandache, R. R. Choudhury, and I. Rhee. Towards Mobile Phone Localization without War-Driving. In *Proceedings of IEEE Infocom*, 2010.

[7] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox. EnLoc: energy-efficient localization for mobile phones. In *INFOCOM Miniconference'09*, 2009.

[8] N. Deblauwe and G. Treu. Hybrid gps and gsm localization - energy-efficient detection of spatial triggers. In *WPNC'08: Proceedings of the 5th Workshop on Positioning, Navigation and Communication*, 2008.

[9] S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. MetroSense project: people-centric sensing at scale. In *First Workshop on World-Sensor-Web: Mobile Device Centric Sensory Networks and Applications WSW06, SenSys 06*, 2006.

[10] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proceedings of 6th International Conference on Mobile Systems, Applications, and Services (MobiSys'08)*, 2008.

[11] T. Farrell, R. Cheng, and K. Rothermel. Energy-efficient monitoring of mobile objects with uncertainty-aware tolerances. In *Proceedings of 11th International Database Engineering & Applications Symposium (IDEAS'07)*, 2007.

[12] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay. Myexperience: A system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of 5th International Conference on Mobile Systems, Applications, and Services (MobiSys'07)*, 2007.

[13] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *Proceedings of 6th International Conference on Mobile Systems, Applications, and Services (MobiSys'08)*, 2008.

[14] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar. Autowitness: locating and tracking stolen property while tolerating gps and radio outages. In *Proceedings of 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys'10)*, Nov. 2010.

[15] M. Haridasan, I. Mohomed, D. Terry, C. A. Thekkath, and L. Zhang. StarTrack Next Generation: A Scalable Infrastructure for Track-Based Applications. In *OSDI'10: Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation*, Berkeley, CA, USA, 2010. USENIX Association.

[16] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. SensLoc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*, 2010.

[17] M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer. EnTracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, 2009.

[18] M. H. Krieger, M.-R. Ra, J. Paek, R. Govindan, and J. Evans-Cowley. Urban Tomography. *Journal of Urban Technology*, 17:21–36, 2010. http://tomography.usc.edu/.

[19] A. Lamarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *In Proceedings of the Third International Conference on Pervasive Computing*, pages 116–133. Springer, 2005.

[20] H. Lee, M. Wicke, B. Kusy, O. Gnawali, and L. Guibas. Data Stashing: Energy Efficient Information Delivery to Mobile Sinks through Trajectory Prediction. In *IPSN'10: Proceedings of the IEEE/ACM International Conference on Information Processing in Sensor Networks*, 2010.

[21] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-Accuracy Aware Localization for Mobile Devices. In *Proceedings of 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, June 2010.

[22] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of 6th ACM International Conference on Embedded Networked Sensor Systems (SenSys'08)*, Nov. 2008.

[23] M. Motani, V. Srinivasan, and P. S. Nuggehalli. PeopleNet: Engineering a wireless virtual social network. In *Proceedings of 11th ACM International Conference on Mobile Computing and Networking (MobiCom'05)*, 2005.

[24] J. Paek, J. Kim, and R. Govindan. Energy-Efficient Rate-Adaptive GPS-based Positioning for Smartphones. In *Proceedings of 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, June 2010.

[25] S. K. Singh. Review of Urban Transportation in India. *Journal of Public Transportation*, 8(1):79–98, 2005.

[26] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 – 197, 1981.

[27] T. Sohn, K. A. Li, G. Lee, I. Smith, J. Scott, and W. G. Griswold. Place-its: A study of location-based reminders on mobile phones. In *UbiComp'05: Proceedings of the 7th International Conference on Ubiquitous Computing*, 2005.

[28] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun. Attacks on public wlan-based positioning systems. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, 2009.

[29] A. Varshavsky, M. Y. 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 *Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications*, 2006.

[30] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of 7th International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, 2009.

[31] C.-W. You, P. Huang, H. hua Chu, Y.-C. Chen, J.-R. Chiang, and S.-Y. Lau. Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization. *Elsevier Ad Hoc Networks*, 6(8):1221–1237, 2008.

[32] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving Energy Efficiency of Location Sensing on Smartphones. In *Proceedings of 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*, June 2010.