Optimizing Mobile Phone Self-Location Estimates by Introducing Beacon Characteristics to the Algorithm

Sebastian Hubrich, Kevin Curran

School of Computing and Intelligent Systems
University of Ulster, Northern Ireland, UK
Email: kj.curran@ulster.ac.uk

Abstract
Positioning technologies that use Global System for Mobile Communication (GSM) networks for location estimation (such as the Privacy Observant Location System (POLs) and the Place Lab framework) lack the accuracy that other positioning technologies like the Global Positioning System (GPS) have. GPS receivers are capable most of the time of placing a person within 10 metres of a known location. Place Lab is an open platform framework implemented in Java for client-side location sensing that can calculate a position estimate from various beacon sources, such as GSM beacons. The POLs framework is a counterpart of Place Lab for Windows Smartphone devices which provide the tools to develop location-based services quickly.

There is a lack of accuracy however when the location estimation algorithm uses only GSM readings. Measurements that have been made with Place Lab show a median accuracy of 232m in downtown areas. Place Lab and POLs do not need additional hardware, apart from the mobile phone itself however their lack of accuracy compared to GPS is significant. Due to this rather poor accuracy, the use of those frameworks is limited to applications where the accuracy is not crucial. This paper presents the results of improving the accuracy of location estimation in urban areas by extending the algorithm used in the POLs and Place Lab frameworks to take into account the beacon properties, Effective Radiated Power (ERP) and beacon height when estimating a position. The extended algorithm based on beacon properties outperforms the centroid algorithm by over 30%.

Keywords: location estimation, effective radiated power (ERP), location positioning, GPS

1 Introduction

The first GSM network went live in 1991. Since then, the number of customers has increased to 2 billion users. Today, GSM accounts for 82% of the global mobile market, and 29% of the global population uses GSM technology (GSM-Association, 2006). As the network progressed, so did the technology. Service providers already offer some location-based services, mainly in the areas of fleet-tracking and child-tracking. Most of these services are purely network oriented; meaning that the person carrying the phone being tracked has no influence on this service, or direct benefit from it. There have also been cell-ID based location technologies available in which the location technology is network based but the location request and benefit of same is the mobile user. Another interesting field is that of location-aware devices, such as the GPS-receiver or the location-aware mobile phone. At a time when 2 billion users already have a mobile phone, there is tremendous interest in having a piece of technology that eliminates the need for an extra device. Hence, location-aware mobile phones could be of interest in a wide area of applications and services; a few examples are friend-finder, restaurant-finder, and routing and tourist information. With the emergence of web-based technologies such as web services, the technical capability of location-based services that can be used from every mobile phone is now available, using technologies such as GPRS or EDGE. In particular, governments in numerous countries have expressed their need to locate mobile phone users, especially for locating the origin of emergency calls. In the US, the Federal Communication Commission stated their requirements in the E911 mandate, which calls for location information at a scale between 50m and 300m in most cases (FCC, 1996). The European counterpart of E911 is E112.
The Place Lab architecture is as an open platform for client-side location sensing (LaMarca et al., 2005). Place Lab currently runs on Windows XP, Linux (only 802.11 beacons), Mac OS X (only 802.11 beacons), Pocket PC 2003, Symbian (GSM and Bluetooth beacons). Place Lab can use three different beacon types for its location sensing: 802.11 Wi-Fi, GSM and Bluetooth. The current Place Lab location algorithms used on Smartphones use the CellID for database mapping to estimate a location. The CellID database is filled by means of a so-called “stumbling” process, meaning that a user drives around with a GPS device attached to a Smartphone or notebook, automatically recording signal readings that are saved in a text file together with the matching GPS reading. By post-processing this data (the so-called log-cooking process), the cooker algorithm looks for the strongest signal reading for any given CellID, takes the GPS measurement for this reading and saves it as a CellID fingerprint in another text file (the actual database for later lookup). The POLS (Privacy Observant Location System) framework is a Dot Net (.NET) implementation of Place Lab for Microsoft Smartphones that uses a centroid tracker for location estimates. The centroid tracker takes all the readings from the Mobile Station (MS) and searches its database for the location (Latitude/Longitude) of each CellID reading. Using those locations, the average value for latitude and longitude is calculated, which becomes the location estimation output of the centroid tracker.

During field studies, LaMarca et al. (2005) discovered that, because of its smaller cell size, they obtained the best results with the Wi-Fi beacon type as long as the beacon density is sufficient. The 802.11-based location estimates showed a median accuracy of 20.5m in urban areas with 100% coverage, whereas GSM showed a median accuracy of 107.2m with 100% coverage. Due to the limitations of the Nokia cell phone used for the field studies, all GSM-based results were calculated using a single GSM cell reading. Non-mobile Bluetooth beacons have not yet reached sufficient density in the wild for tests to be possible. Place Lab’s (or rather, Symbian’s) limitation of investigating only a single cell in GSM-based location sensing is a major drawback. Nevertheless, Place Lab (and POLS in its Dot Net implementation for Microsoft Smartphones) provides a framework to build upon. It is worth noting that the POLS framework for Microsoft Smartphones can retrieve all the GSM cell readings from the phone’s memory. Silventoinen and Rantalainen (1995) found that the performance of a locating algorithm relying solely on the CellID is rather poor while pico or nano cells are not being used very frequently. Other methods of mobile station location technologies that they investigated involved design changes (hardware and/or software) at either the BTS, the MS or both.

The development of location aware devices built upon the already existing GSM technology is a promising approach, considering the increased interest in location-based services. The beauty of this approach is that the physical properties now being used to provide location estimates are simply those needed to keep the system working (that is, to provide mobile communication). This approaches to location estimation can be done anywhere in the world where there is GSM coverage, independently of the network operator and without any additional cost.

Hence, this research investigates the impact on the location estimation of adding beacon specific properties to the tracker algorithm. Properties such as the transmitting power and antenna height can be used to generate weighting factors, which can be taken into account in the algorithm so as to improve the precision of the location estimation. These beacon properties will be used to apply a radio wave propagation model (basically the Hata formula solved for distance) to improve the location estimation. Results of field experiments conducted by Cheng et al. (2005) and Tonteri (2001) show that working with weighted properties is a promising approach to improve location estimation. Cheng et al. (2005) improved the performance of their algorithm in downtown areas by 4.1%, by using only the received signal strength as a weighted property, while Tonteri (2001) showed that adding an additional propagation parameter (direction of transmission) to a statistical signal propagation model further improves the precision of the location estimation.

We aim to demonstrate that adding beacon characteristics to the self-location algorithms will significantly improve the performance of an MS self-location estimate in urban areas. We also aim at providing better location estimation for Smartphones, by improving the performance of the algorithms alone, without the need for an external device such as a GPS receiver, or network-supported tracking features (which often have to be paid for). This reduces to some degree the possibility of an invasion of privacy, since the location estimation takes place entirely within the Smartphone device. No outside party has access to these estimates, unless it is explicitly granted for use in possible future applications which would be necessary for many location-based services (Hong, 2003).
2 Mobile Self Location Estimation

Here we present a prototype that has been developed which incorporates an alternate runtime algorithm whilst logging data for each algorithm on the MS. By analysing the log files and cross-checking the results from the algorithms with GPS readings made concurrently on a predefined route, the performance of the optimized algorithm can be measured and compared against the traditional centroid tracker algorithm. Numerous extensions were added to the POLS (privacy observant location system) framework. In particular a new beacon representation was needed that has fields for the extended properties and a tracker algorithm was needed to take the extended properties into account when delivering location estimate updates. A new tracker algorithm was developed that takes account of the extended properties and weights them accordingly while delivering location estimates. This class adds 4 new fields to it:

- Verified (Boolean flag which determines if the beacon has been verified)
- TX (Enumeration that determines the ERP level of a beacon in steps 1 to 4)
  - 1 = large, for ERP > 1000W
  - 2 = medium, for ERP 100W – 1000W
  - 3 = small, for ERP 10W – 100W
  - 4 = very small, for ERP <= 10W.
- Environment (Enumeration that determines the area around the beacon in steps 0 to 2)
  - 0 = rural
  - 1 = suburban
  - 2 = urban.
- Height (The antenna height of a beacon, 30m – 200m)

The tracker uses a combination of a recognized and thoroughly tested empirical radio wave propagation model and a simple averaging method to estimate the location of a MS. The Hata Model is a radio wave propagation model that was developed after extensive measurements of urban and suburban radio propagation losses. Okumura et al. (1968) published many empirical curves useful for planning cellular systems. These were reduced to a convenient set of formulas known as the Hata Model. The Hata Model covers 150 MHz to 1.5 GHz transmitter frequency with a base station antenna height of up to 200m (Hata, 1980). The basic formula for the median propagation loss given by Hata is:

$$L(dB) = 69.55 + 26.16 \log f_{MHz} - 13.82 \log h_1 - a(h_2) + (44.9 - 6.55 \log h_2) \log d_{km} - K$$

Where:
- $L_{dB}$ = Path loss in dB
- $f_{MHz}$ = Frequency in megahertz
- $h_1$ = Base station antenna ht (30m–200m)
- $h_2$ = Mobile station height (1m – 10m)
- $d_{km}$ = link distance in kilometer
- $K$ = Correction factor for suburban areas

Figure 1 shows the development of the path loss component in dB with increasing distance, according to the Hata radio wave propagation model.
As the extended beacon properties include the ERP value of the beacons and the MS reports with every measurement, the received signal strength and the path loss at the MS can now be calculated and the Hata formula can be solved for distance. The result is the estimated distance between the MS and the BTS. The Hata formula solved for distance (km):

\[
L = 55.55 - 26.16 \log(f) + 13.82 \log(h_1) + a(h_2) + K_f
\]

\[
d = 10 \frac{L}{44.9 - 6.55 \log(h_1)}
\]

The POLS framework reports the received signal strength at the MS as a positive integer value. This signal value represents the C1 (path loss criterion) value from the GSM protocol. Following the reasoning below, the received signal strength from the MS therefore translates into:

\[
L(dBm) = C_1 - 110(dBm)
\]

Tabbane (1997) states that in the GSM, the main criterion used to choose the target BTS and to trigger a handover is the path loss criterion (C1). The path loss criterion parameter C1 is defined as

\[
C_1 = (RxLev - RxLevAm - MAX((MSTxPwr - MSMaxTxPwr),0))
\]

Where:
- MSTxPwr is the maximum allowed handset transmitting power in this cell.
- MSMaxTxPwr is the maximum handset transmitting power
- RxLevAm is the minimum level of signal needed to log into a cell. It is assumed that this is the same number on every network (-110dBm)

The C1 value is used for cell selection and reselection. This is calculated by the MS and is used for deciding which cell to connect to. The path loss criterion is more useful than just \(RxLev\), since it takes \(MSTxPwr\) and \(MSMaxTxPwr\) into account. \(MSMaxTxPwr\) is the phone’s maximum output in dBm. The reason why TX power is factored into C1 is so that the MS only connects to cells where it has a reasonable chance to be heard by the base station if it transmits.

\(MAX((MSTxPwr - MSMaxTxPwr),0)\) is equal for multiple handsets in the same cell; therefore, the resulting equation is \(C_1 = (RxLev - RxLevAm)\). The ERP value has been stated in watts; therefore, a transformation into dBm is needed to feed the value into the formula. As described above, the Swiss Federal Agency for Communication (BAKOM) states the maximum ERP value of a BTS location. That is, it is the sum of all ERP values of all antennas (directional and omnidirectional) of all providers at this location if the maximum possible MSs are logged into the cells and are receiving at the same time. Examinations throughout the test area revealed that almost all BTS locations in the test area have 3 antennas which cover 360° in 120° sections. The value used for the ERP in this study is therefore the value defined by BAKOM divided by 3. Additionally, a correction factor can be applied to take into account the unlikely eventuality that all antennas at a BTS location are transmitting with maximum power. Despite the lack of detailed knowledge of the ERP value of each cell, it is a good starting point for calculating the distance circles. The results of the prototype testing (in a later section) show that the values are in a reasonable range. The formula to convert Watt into dBm is

\[
dBm = 10 \log(ERP * 1000)
\]

To visualize the radio wave propagation model, BTSs are represented by points on a plane in Euclidian space. The radii of the circles are the calculated distances of the MS from the BTSs. The intersections of the circles around the BTSs show the points where the MS is believed to be. Figure 2 illustrates three possible constellations of 2 BTSs and the ways the distance circles might or might not intersect.

1. Intersecting
In this scenario, the intersection points E and F (denoted by red x labels) are the candidates for the MS’s position.

2. Non-intersecting and apart

In Figure 3, the circles do not intersect and so are apart from each other. In this case, the midpoint of the shortest distance between the two circles is regarded as the candidate point.

3. Non-intersecting and contained

The circles do not intersect as shown in Figure 4 but one circle is contained in the other. Thus, the same logic as in scenario 2 is applied, so the midpoint of the shortest distance between the two circles is regarded as the candidate.
The BPExtendedTracker algorithm obtains current GSM measurements by a callback method whenever a new measurement is available (it must be noted this technique will not work for 3G). Each measurement consists of a set of "reading" objects, each object containing the beacon id (CellID) and the signal strength of that reading. The mapper file data structure contains the CellID as the lookup property plus additional information about that beacon. For each beacon, the current path loss (in dBm) is calculated as described above. A mobile measures and reports the serving cell and the next strongest cells. That is a total of 7 circles; one for each cell with the centres on the cell antenna coordinates. A mobile transmission site most commonly consists of three cells pointing roughly 120 degrees apart. For each identified BTS in the measurement, the algorithm calculates the probable distance to the MS using the Haversine formula (Sinnott, 1984). Using these distances, the intersection points and the geographic coordinates are determined; this is possible because the geographical location of each BTS has been found by a stumbling process and stored in the local lookup file. A centroid algorithm is applied to the calculated intersections, giving a first location estimate by averaging the latitude and longitude values. During a second run through, each intersection coordinate is tested against the first location estimate and its distance from it is calculated (using the Haversine formula) and stored. Finally, the intersections are sorted in descending order of this calculated distance criterion, and the upper 50% are eliminated as being too far away. A further averaging run using the remaining coordinates is then performed. The outcome of this last centroid averaging estimation is then considered to be the current location estimate and is broadcast to the subscriber. See Section 4.4, Field Study, for a screenshot. We believe this approach to be a less computational intensive approach than the least squares estimation approach.

3 Evaluation

In order to examine the effectiveness of the new algorithm in urban areas we chose the city of Schaffhausen (see Figure 5), which provides suitable beacons throughout the city and has a sufficient level of building development to produce some multipath propagation effects. Although those effects are not expected to be as significant as in cities packed with skyscrapers, they should still be noticeable. Whereas a GPS receiver can experience difficulty with its signal where there are building structures and narrow streets, it is expected our GSM algorithm will obtain even more accurate results the more “urban” a city gets. This is because GSM providers need to deal with multipath propagation effects by increasing the numbers of BTSs and so more reference points are provided for the algorithm.

Figure 5 displays the locations of the test area beacons. These two figures provide a picture of the test area where the field experiment outlined below has been performed, and Table 1 outlines the numerical data about the test area.
The traditional “stumbling” process used by the POLS framework logs only the GPS position, together with the received signal strength, for each CellID. During the “log-cooking” process all available log-files are scanned. For any CellID found to have a signal strength higher than a predefined threshold the logged GPS position is selected to be stored in a text file (newer versions of the POLS framework allow a MySQL database as an alternative). The result is a text file with a CellID identification string and its logged GPS position stored on each line. The CellID identification string consists of MCC (Mobile Country Code), MNC (Mobile Network Code), LAC (Location Area Code) and CellID (Cell ID Number). A typical CellID identification string has the following format: 228:03:7002:26596. A typical entry in the mapper file after the log-cooking process is shown in Table 2.

<table>
<thead>
<tr>
<th>MCC</th>
<th>MNC</th>
<th>LAC</th>
<th>CellID</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Verified Location</th>
<th>ERP</th>
<th>Env</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>228</td>
<td>03</td>
<td>7002</td>
<td>26596</td>
<td>47.700042</td>
<td>8.636265</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 2: Typical mapper entry

The ExtendedBeacon algorithm uses the additional beacon properties: ERP, Verified, Environment and Antenna Height, enabling the algorithm to relate the received signal strength at the MS with the ERP value and to calculate the path loss component. To collect those extended properties that are not caught by the traditional “stumbling” process, the procedure below was applied. The Swiss Federal Agency for Communication (BAKOM) provides an online map showing all radio beacons with their ERP values. Using this map for the area of interest, a route for the traditional stumbling process was defined. The goal was to enable the stumbling process to “see” all the CellIDs of all BTSs in the test area with the highest possible signal strength. Therefore, each BTS on the BAKOM map was circled at decreasing radii, while running the stumbling process on an MS. After a “log-cooking” process with a signal threshold of 50, a reading was made for each CellID in the test area, with its approximate GPS location in the mapper file (approximate due to the way the log-cooker tries to estimate the exact location of a beacon from the stumbling log files). The BAKOM BTS coordinates were delivered in the Swiss coordinate system (Swiss Grid), based on the Bessel-Ellipsoid and called CH1903. The coordinates had to be transformed to the WGS84 ellipsoid format by using the application provided by the Swiss authorities. Using the BAKOM map as reference and Google Maps for latitude/longitude coordinates, the mapper file was modified so that each listed CellID now has the correct location. This manual correction of the stumbling/log-cooking procedure was done to ensure a deterministic database for both algorithms to work with. Otherwise, it would not be known whether a location estimate derivation (in relation to the current position estimate from a GPS receiver) comes from an incorrect database or from the algorithm itself. The 100% correct mapping of a CellID to a GPS location is not as important with the traditional algorithm, because there are no additional properties tied to a specific beacon (CellID). Because of the extended properties at the correct location, it is very important to map the ERP value to the correct CellID. After each logged CellID had been uniquely mapped to a BTS and a correct location, the extended properties of this BTS (ERP and height) were added to the mapper file. As the research was limited to urban areas, Environment was always set to 2.

The additional values represent (corresponding to the given example):

1 = Flag determining whether or not CellID location was verified (0/1)
4 = ERP (1/2/3/4 enumeration for large/medium/small/very small)
2 = Environment (0/1/2 enumeration for rural/suburban/urban)

http://www.swisstopo.ch/de/online/calculation/navref/?jsessionid=2hjy96jfrjvuv
We evaluated our new ERP algorithm against the traditional algorithm by performing field experiments in which each of them was tested at a location or path whose geographical coordinates were known. The MS then logged the resulting data from both algorithms in a data file. The location estimates of both algorithms could then be compared with one another to display the performance of each algorithm in comparison to a GPS reading. The aim of this field experiment is to compare the accuracy of the location estimates of each algorithm in a controlled and defined environment. An application has been developed that runs on the MS and uses both tracker algorithms, and a GPS tracker as well. The GPS tracker is triggered every 1000ms to update its location. As soon as the new GPS position is available, the two tracker algorithms are triggered to update their location estimate. This ensures that there is a close coordination in time between the GPS position and tracker algorithm updates. Each complete set of position estimates (GPS, Centroid and BPExtended) is appended to a log file for postprocessing.

In multi-path environments (urban areas) it is often necessary to measure location with four antennas to overcome the multi-path radio propagation effects or indeed to use estimates of the location of the MS with its previous positions (Wylie and Holtzman, 1996). Our solution however was to calculate the position with the measured time from three BSs and compare these measurements with stored measurements to locate the nearest predefined point. Finally we corrected the estimated position by the coefficient factor of the nearest point. This however does have a performance hit on our algorithm and we felt that given more time, we could optimise it by increasing the predefined points in the more frequented locations. This could be done automatically at the end of say, a week by taking measurements from multiple phones and therefore more points would exist for the heavily populated areas thus reducing the error.

Beginning with the GPS tracker, all three algorithms estimate their location every 1000ms and record the results together with a timestamp in a log file. By post-processing, splitting this log file to get a file for each algorithm, and importing all three resulting log files into Google Earth, visual feedback is received on the location estimates along the defined test path and a median error rate can be defined for each algorithm against the results from the GPS tracker. Each reading in the log file consists of a GPS entry and an entry for each tracker algorithm. A small C# command line tool was been developed to perform the following post-processing on the logged data:

- Eliminate invalid readings, that is, readings where the GPS entry states that there were fewer than 3 satellites seen. In this case, there are no valid reference coordinates against which to evaluate the tracker algorithms.
- Extract the GPS log data entries and copy these readings into a .kml file (which is a XML format for Google Earth) that can be imported into Google Earth to visualize the path taken.
- Similarly, extract the Centroid tracker log data entries and copy them into a .kml file.
- Do the same for the BPExtended tracker log data entries.
- For each valid reading, determine the distance between the location estimate that each tracker made and the reference GPS coordinates, determined using the Haversine formula (Sinnott, 1984). The distances are then exported to another log file called MeanError.txt where the coordinates are stored with their respective distances to the reference coordinates.

Additional field studies ‘in the wild’ were done with both prototype and traditional algorithms, by running tests at known locations (in terms of their physical addresses, rather than their GPS readings). By observing the resulting location estimates, a more subjective impression of the performance of the algorithms in ‘real world’ environments was gained. The prototype algorithm was added to the PhoneMapDemo software, which was extended to allow for switching the algorithms that update the location estimates. The prototype algorithm was expanded to display the estimated location on the map, also to display the beacons in range of the MS as overlays on the map (see Figure 8), with the CellID of each recognized beacon and the received signal strength. The algorithm in use can be changed while the software is running; internally, a complete refresh is performed and the next position estimate is delivered from the selected algorithm. If the BPExtended tracker algorithm is the one in use, the software displays the Hata distance circles on the map and updates them with each measurement. The circle intersections are marked visually as well, by orange dots. The Centroid position is shown as a red dot, while the location estimate from the BPExtended tracker is shown as a purple dot.
Figure 7: Screenshot showing Hata Distance Circles

Figure 7 shows a screenshot of the POLS Map Demo using the proposed algorithm with Hata distance circles. The coloured squares are the BTSs that the MS can currently “see” with their respective positions on the map. In the top left-hand corner, the C1 value (path loss component) for each BTS is shown. The red dot is the location estimate from the Centroid algorithm, whereas the purple dot is the location estimate from the BPExtended algorithm (which in this example is more accurate by 200m). The green arrow denotes the physical location from which the test has been conducted.

4 Results

Here we investigate the impact on the estimation accuracy of the location algorithm in urban areas of adding the beacon properties ERP and antenna height to the algorithm. By analysing the data gathered in the field experiments, it is possible to compare the performance of each location algorithm, in terms of median errors in the location estimate. GPS readings are not 100% accurate, but with 10m accuracy for 95% of the time, they provide a suitably accurate base for comparing the performance of the algorithms (Hulbert and French, 2001). The log file grew to 600kB during the field experiment, using a 2kB mapper file as a lookup source. It is worth mentioning that the mapper file covered the test area plus its significant surroundings. In contrast to the software developed for the field experiment, the prototype developed for the field study aimed at providing a more subjective and completely visual comparison between the two algorithms. No log data files are created; instead, every measurement, every reading and every location estimate update is instantly mapped onto the display of the MS as circles, dots and squares. The result was a file for each tracker algorithm that can be imported into Google Earth to visualize the results of the field experiment.
Figure 8: Imported data from field experiment

As shown in Figure 8, the red path denotes the GPS coordinates from the path we took through the test area. The purple path denotes the BPExtended tracker algorithm location updates during this field experiment. The turquoise path visualizes the location estimates from the Centroid tracker. It can be clearly seen from the data that the BPExtended tracker algorithm is subject to more variation than the Centroid tracker, while it tends to give better location estimates. This behaviour has been confirmed by the field study. Figure 9 shows the location estimate errors in metres for each valid measurement point (that is, for each measurement point that has a valid GPS reading). The variation mentioned earlier can be seen very clearly in this chart. The Centroid error gradient is relatively smooth when compared to the BPExtended error gradient. The median location estimate error for the field experiment is 230.1m for the Centroid algorithm and 158.8m for the BPExtended algorithm. The Place Lab and POLS framework were aimed at providing a relatively easy-to-use entry framework to client-based location estimation technology that can be used by anyone with a mobile phone, they were limited in their choice of available tracker technologies, because they are built upon a relatively small database (mapper file content) and the measurements of the MS. Basically, the mapper file stores only the ID and coordinates for each beacon. These coordinates, gathered during the stumbling process, are subject
to errors due to the gathering process (averaging the places where the highest signal strengths in each cell have been observed).

Figure 9: Location estimation errors

A recent examination revealed an average error of 56m (Chen et al., 2006). This average error (added to the location estimation error) and the small size of the working data set is the price to pay for the comfortable way in which the stumbling processes work. Adding beacon properties to the mapper file immediately broadens the possibilities available for tracker algorithms. However, this complicates the initial process of stumbling the beacon information. We chose to add ERP and antenna height as relevant beacon properties to the mapper file and to evaluate those properties with the prototype algorithm. When an ERP property was available, it was possible to use a radio wave propagation model to calculate actual path loss values; the Hata model was chosen as a widely recognized and proven model. The underlying formula has been solved for distance, so that the model revealed the distance of the MS from the beacon in question. Calculating this distance for all beacons currently sensed by the MS provides the circle scenario described earlier.

Figure 8 displays the data which was linked into Google Earth to get visual feedback on the location estimates the algorithms performed. For reference, a GPS position tracker has been included as well (red path). From the visualized data it can be seen that the BPExtended algorithm (purple path) produces more variation in its estimates than the traditional Centroid algorithm (turquoise path). These variations can be classified as ‘jumping location estimates’. This is because the Centroid algorithm always delivers the geographical mean of the BTS locations which limits the resulting location estimate to only 1 set of coordinates for a given set of ‘seen’ BTSs.
Figure 10 visualizes the scenario when the MS ‘sees’ two BTSs. The centroid algorithm will always deliver the coordinates at point x as an estimate when those two BTSs are seen. In contrast, the BPExtended algorithm models the radio wave propagation in urban areas, which results in many different possible coordinates for a given set of ‘seen’ BTSs — depending on the signal strength of each BTS at the current MS location at the time the algorithm performs the measurement. Because of the fact that the received signal strength at the MS can vary due to many factors (scattering, attenuation, etc.) even if the MS does not move, the BPExtended algorithm shows more variation in its delivered estimates.

Figure 11 shows the range of possible location estimates the algorithm can deliver (depending on the received signal strength) for the two seen BTSs. In this scenario the blue circles visualize the maximum possible distance when following the Hata formula. That is, the blue circles are the farthest point from a particular BTS where a signal can still be received at the MS. Depending on the actual received signal strength at the MS, these circles change their radii and new intersections are calculated. The yellow coloured area displays the possible zone where those intersections can occur if the blue circles represent the maximum distance. As the BPExtended algorithm relies on the calculated intersections this scenario shows the difference in the location estimate possibilities for the two algorithms and explains the “jumping location” effect that was found during the tests. This effect is clearly visible at the purple line in
During the calculation of the location estimate the BPExtended algorithm needs to calculate possible intersections of the Hata distance circles. As soon as the algorithm has revealed the intersection coordinates, a first mean coordinate value of those intersections is calculated. The algorithm then compares the distance from each intersection to this calculated mean value and eliminates the intersections farthest away from the calculated mean value. From the remaining intersections, another mean coordinate value is calculated which is then the actual location estimate the algorithm delivers. The second post-processing run, in which the distance to the mean value is calculated, improves the location estimate, because while calculating the intersection points the algorithm always reveals two intersections, out of which only one is likely. This second post-processing run is therefore aimed at eliminating the unlikely intersections.

While the results reveal that significant improvements to the location estimation accuracy can be achieved by adding beacon properties to the algorithm, these improvements in their current state are merely academic in nature. The traditional process of gathering the data with which the algorithms can work (stumbling) is relatively easy to perform and straightforward. In contrast, gathering the data needed for the beacon properties algorithm proved to be more demanding and time-consuming. Table 4 compares the steps that were necessary to obtain the data.

<table>
<thead>
<tr>
<th>Traditional stumbling</th>
<th>Extended stumbling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run stumbling application at MS</td>
<td>Obtain map with BTS locations for area of interest</td>
</tr>
<tr>
<td>Run log cooking process on stumbled data</td>
<td>Run stumbling application at MS. Circle the BTS positions to get distinct readings for the different antenna directions (different CellID’s). Note the antenna height</td>
</tr>
<tr>
<td>Run log cooking process on stumbled data</td>
<td>Check the log cooking result with Google Maps and the BTS locations from step 1 for plausability. Map each cellID to a BTS</td>
</tr>
<tr>
<td></td>
<td>Add the extended properties (the antenna height noted in step 2 as well as the ERP value). In Switzerland the ERP value is stated on the same map as the BTS locations</td>
</tr>
</tbody>
</table>

Table 4: Stumbling process

Mapping the CellID’s to the BTSs often required a visit to the area in question while closely monitoring the CellID and signal reading. Especially when beacons are standing close to each other, it is not always clear from the log data to which BTS the CellID belongs. Following the reasoning above
it becomes obvious that the process of adding the data may not always be appropriate for a real-world application as it is indeed very time-consuming.

5 Conclusion

This research examined the impact of adding beacon properties to a mobile phone self location estimation algorithm through a radio wave propagation modelling algorithm (the widely recognized Hata model). The process of adding beacon properties to the data that the algorithms can use provides more possibilities for algorithms to calculate location estimates. To further improve the location estimation accuracy, an averaging method is applied after the radio wave propagation model has estimated a location because, under normal conditions, the Hata distance circles provide the algorithm with a maximum of $28^2$ intersection points. Each pair of intersection points has one unlikely location point that we have tried to eliminate using this downstream averaging process.

We conclude that adding beacon characteristics to the self-locating algorithm of mobile phones can increase the location estimation accuracy in urban areas by an average of 31% or 71.3 metres thus demonstrating that adding beacon properties to the algorithm can significantly improve location estimation. Additionally, in our quest to provide superior location estimation for Smartphones without the need for an external device such as a GPS receiver, or network-supported tracking features (which often have to be paid for), we have reduced the possibility of an invasion of privacy, since the location estimation takes place entirely within the Smartphone device. Hence, no outside party has access to these estimates, unless it is explicitly granted for use in possible future applications which would be necessary for many location-based services.

References


