Background Positioning for Mobile Devices - Android vs. iPhone

Denis Huber
School of Electrical and
Computer Engineering
Technical University Berlin
Email: hubeealh@mailbox.tu-berlin.de

***Abstract*—This seminar paper covers the current techniques available for mobile phone positioning and their application. GPS, WiFi and cellular based positioning methods are explained in detail, and current trends regarding the optimization of existing algorithms are analyzed. The algorithms talked about include fingerprinting and different time of arrival methods. Current solutions in the mobile phone market from Apple and Google which use several positioning technologies in their products are investigated. Especially the APIs from Apple and Google who deliver access to these technologies. Location Based Services (LBS) which need continuous monitoring of the users position are leading to a high energy demand. This problem is not solved yet satisfactory, and different approaches for solving this problem are available.This paper investigates some of the proposed algorithmic solutions and gives an outlook to the future.**

I. Introduction

Smartphones with high bandwidth internet access and GPS receivers have become very widespread these days and have repressed traditional mobile phones. Two of the leading companies in the mobile phone operating system market today are Apple with their iPhone iOS and Google with Android OS. One of the major features of these operating systems is the ability of running a wide array of third party applications available for purchase in special App Stores. A special kind of applications has emerged in the last few years, the so called Location-based services (LBS) [1]. These are applications using the ability of smartphones to localize the geographical position of their users outdoors as well as indoors. With applications ranging from showing the users position on the map or routing to new kinds of services like Location-based gaming, Location-based advertising or even mobile dating, many companies are developing LBSs today. But also security applications like for example the tracking of 911 calls, which is now mandatory by law in the United states [2], are using the location information provided by mobile phones these days. While some applications may need only very rough position fixes others may need a very accurate location determination. One of the major challenges in localization today is to deliver this accuracy without consuming too much of the limited amount of battery power available on the mobile devices. In order to realize Location-based services a lot of different techniques are available today including GSM, WiFi or GPS positioning.

These different methods as well as new approaches of improving them are covered in section 2. In section 3 a look under the hood of the positioning capabilities of the iPhone and Android OS's will be provided, as well as the API calls available for developers. In section 4 the problem of continuous tracking by LBS's leading to high power consumption will be discussed. Different approaches to solve this problem will be presented. The final section gives an outlook for the future and concludes this paper.

II. Overview of Positioning Systems

Modern smartphones today can use up to three different methods to locate their position. Satellite Positioning, Wifi Positioning and Cellular Positioning (Figure 1). In the following subsections all three will be explained in detail. Generally these positioning methods have different characteristics which can be measured by the following criterias like described by Hightower et. al [3].

- **Accuracy.** The difference between the true position and the calculated position.
- **Precision.** The closeness of a number of position values to their mean value.
- **Power consumption.** The amount of electrical power needed for determining the position measured in Watts per second.
- **Latency.** The time needed for obtaining the position fix. Also generally known as TTFF (Time to first fix) measured in seconds.
- **Availability.** Not all positioning methods are available in every situation.

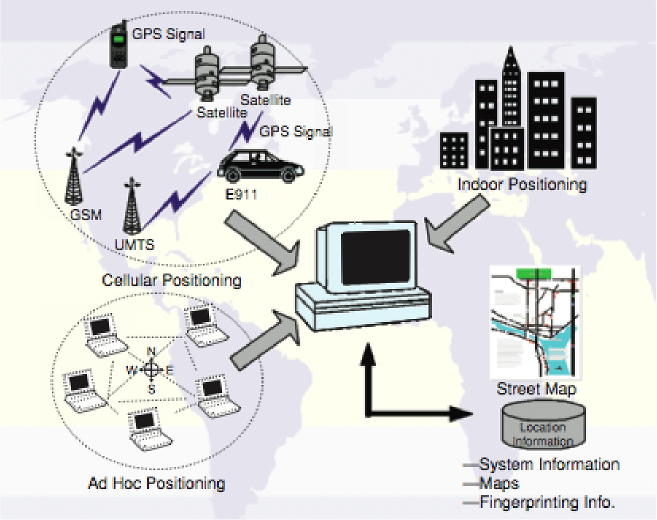


Fig. 1. Overview of Position Systems from [4]

A. Satellite Positioning

Satellite Positioning is a technology originally invented by the US Military. The most prominent satellite positioning method today is the Global Positioning System (GPS). It uses up to 32 satellites to determine the position of a Mobile Station (MS) [5]. There are two types of signals sent by the satellites. The Precise Positioning Service (PPS) and the Standard Positioning Service (SPS). While PPS is reserved for military use and is encoded, SPS is free for public use. A mobile device equipped with a GPS Receiver catches up at least four of these satellite signals which are broadcasting time and orbital information and calculates the position from these. GPS is very accurate and precise technology, providing accuracy up to 20 meters, but it consumes also a lot of power on the end device. Another downside of GPS is that satellite signals can not be received indoors, and therefore it is not suited for indoor positioning. Also the TTFF can be up to 15 minutes long because when the device was off for a longer period of time the complete Almanac database (location of all satellites in the orbit) must be reloaded into the device. Most smartphones today however use Assisted GPS (A-GPS) like described in [4], a technique where the traditional GPS Receiver is assisted by cellular positioning methods and the assisting data is downloaded in the background via IP connection. This results in better accuracy, faster TTFF times and less power consumption. However IP connectivity may not be available in all situations and power consumption is still higher than in other positioning methods.

As GPS Receivers have become less and less cheap they are integrated in nearly every smartphone today becoming a standard. Beside that a lot of improvement in the Hardware of GPS Receivers has been done lately, and recent studies have shown that they can be functional in certain indoor environments which can bee seen in [6].

B. Cellular Positioning

Cellular networks have expanded globally in the last decade and reached almost worldwide coverage. At the moment there are mainly two types of cellular networks in operation. The Global System for Mobile Communication (GSM) and the Universal Telecommunication System (UMTS) often also referred as 2G and 3G networks. While the next generation of networks Long Term Evolution (LTE) is already in the pipeline, it will take again probably two years since this standard will be integrated in the network infrastructure and on the mobile devices. While work is already in progress in specifying positioning for the next generation of cellular networks, in this paper only techniques used for 2G and 3G networks are explained.

Cellular networks are basically divided into cells which ideally have a round shape. Each of this cells is associated with a cellular tower (base station). As a user moves through the cells his mobile device will connect to the base station with the best signal strength available, which should be in most cases the nearest one according to his position. The basic form of cellular positioning therefore is to use the location of this base station, whose position is already known and stored in a database. This method is referred as cell identification (cell ID). The accuracy delivered from this method relies on the cell size which can be up to 35 km in radius for 2G networks and a little bit less in 3G due to technical restraints. For this reason a method called timing advance (figure 2) is used. This technique measures the time signals take to arrive from the mobile device to the base station. The mobile station can then be located in a circle around the base station. This can be improved even further with the use of multiple antennas which divide the circle into sections. These method is referred as enhanced cell ID (E-CID) [7].
When mobile devices are in reach of multiple base stations more complex algorithms can be used, which utilize the Time Difference of Arrival (TDOA) or Angle of Arrival (AOA). See Kuepper [8] for a complete overview of these techniques. Generally all of these methods need a high density of cell towers to function properly and TDOA, due the need of timing measurements reliability, is not well suited for 3G networks.

The accuracy of cellular positioning lies far behind GPS, but regarding it's power consumption, nonavailability indoors and additional hardware costs, research in cellular positioning techniques regarding the improvement of their accuracy is considered well. For example Ibrahim and Youssef propose a new algorithm based on a hidden Markov model which is especially suited for low end GSM phones and which delivers quite good results [9] and Nurmi et. al provide a grid based solution [10]. Also a probabilistic method based on RSSI fingerprinting is provided by Ibrahim and Youssef [11]

C. WiFi Positioning

In general WiFi positioning takes a similar approach as the cell ID algorithm in cellular based positioning. Instead of cell towers the geographical information of WiFi Access Points (AP) is kept in a database which can be looked up for positioning. This information is usually gathered through war driving. This means that people equipped with GPS Receivers and WiFi clients are driving around collecting the data. As a key the unique MAC Address of the access points is being used. As WiFi networks have only an average range of 60 meters a good accuracy is provided naturally. Also the TTFF times are low, while no connection must be established to the Wifi Network and the database lookup may only take some seconds. Traditionally WiFi positioning was regarded as a indoor only solution, but as wireless Internet access gained more and more popularity, now there's also a very good outside coverage in urban areas. Jones et.al provided a good research work in analyzing the density of access points in the United states [12].

WiFi positioning can either be performed passive or active from the mobile terminal viewpoint. The active method uses the uplink of a 802.11 WiFi enabled device. A beacon, which is basically a broadcast signal, is sent by the terminal.The access points receive this information and evaluate it in the network.This method is generally referred as network-based positioning.The other method available uses the downlink.The terminal passively scans beacons from the access points, known as Basic Service Set Identifier's (BSSI), which include several information including their signal strength. If no beacon information is available for a longer period of time, the terminal can send a probe request and all access points available in range will send their BSSI back. This is referred as active scanning. Generally WiFi positioning algorithms which use upstream or downstream can be divided in three categories as described in [8].

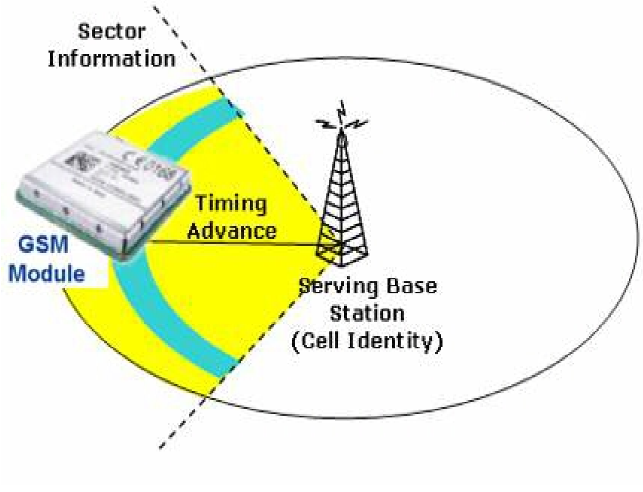


Fig. 2. Cell ID with timing advance and sectored antennas

- **Proximity sensing.**The position of the access point with the best signal quality received is adopted.

- **Lateration.**The distance between the access point and the terminal is determined by the path loss experienced by the beacon during transmission.

- **Fingerprinting.**A database with RSS patterns observed earlier is compared with the actual RSS pattern received by the access points. The best match is then adopted as the actual position.

From these methods proximity sensing is the easiest to implement. Though it suffers from accuracy as the range of a signal broadcasted by access points can vary from some to a hundred meters. Accuracy and precision can therefore not be guaranteed and the algorithm is only suited for granular positioning. Lateration on the other side is only possible when the access points are aligned in a certain way. As this is far from reality in most cases this method is not used in daily practice. The last method remaining is fingerprinting. In this method a radio map,in which the signal strength for a certain position is stored, is generated offline with the already introduced uplink method. This information is stored in a database. The database can be either stored on the device or in the network, the latter being the better option because of limited processing power and memory available on the handsets. In the online phase, the terminal requests it's position from the network in sending the actual measured RSSI information. This information is matched against the existing data from the radio map in the database. The point with the minimum distance to the received one from the terminal is then calculated and will be taken. This is referred as the Nearest Neighbor in Signal Strength method [13]. Beside this simple deterministic approach also different probabilistic methods are possible which can deliver better results but are more complex to implement. Honkarvirta et. al provide a very good overview of these algorithms [14]. RSSI fingerprinting is very promising in terms of accuracy when enough access points are used, as already existing implementations like RADAR [13] show very good results with an accuracy of up to only a few meters. This can surely be improved in the future as a lot of researchers are currently active in this area. A major advantage of WiFi positioning however is that no additional hardware is required, as most devices already implement the 802.11 standard and the access points are already present and don't need any further modification.

D. Summary

As seen in this section so far, all of the positioning methods have different characteristics. With GPS in advance in terms of accuracy and position but weak in energy consumption, and WiFi best suited for indoor application but not available everywhere. While cellular positioning is almost available everywhere today, and very energy efficient it still lacks accuracy and precision. An in depth comparison of all positioning techniques on the iPhone 3G is provided by Zandbergen [7].

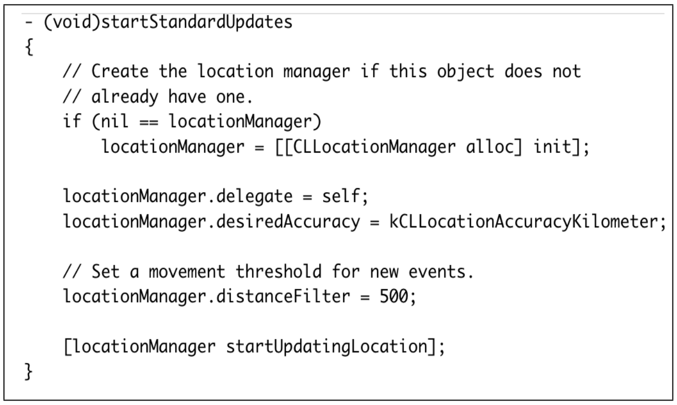
III. Smartphone Specific Positioning

In this section an overview of the current smartphone positioning techniques is provided, by looking at the two most popular mobile Operating Systems (OS) today. These are Google's Android OS and Apple's iOS running on the iPhone. Also third party solutions available for different platforms will be covered.

A. Apple iPhone

Apple recently introduced the fourth version of the iPhone. Simultaneously they also released their renewed operating system iOS, currently available in Version 4.2, which added new features including multitasking. The first generation of iPhones did not have GPS Receivers. Location determination was realized with GSM and WiFi positioning using Google's location databases. With the second and third models Apple built in GPS Receivers and used Skyhook as their main location data service. Skyhook is a company specialized in providing location data services, and will be discussed in the Third Party section. On the newest iPhone 4 model however Apple now use their own database of location information, and like tests have shown the location determination accuracy and performance has been improved with it [17].

Developers can access the positioning features of iOS 4.2 via the Core Location API provided by Apple [18]. There are two major options available. The first one is the standard location service (Figure 3). For the standard location service a new object of the *CLLocationManager* class has to be created.Then a delegate object of type *CLLcationManagerDelegate* and a desired accuracy value must be assigned to it. Also a *distanceFilter* is set to the value in meters when the location should be updated. Finally the *startUpdatingLocation* method is called, and the device begins to track it's position, updating it when the user moved out of the distance set in the *distanceFilter*. The assigned delegate will be notified when location information becomes available or changes. This method as proposed by Apple works best for positioning applications running in the foreground e.g. Turn by Turn Navigation. The higher the value in the *CCLocationManagerAccuracy* property is set the more precise the position data. This also leads to higher energy consumption as GPS will be used when available. However the real value of precision relies on the availability of the methods on the infrastructure side and can change drastically during runtime. The second method for tracking is called significant update change. Like in the standard method a *CLLocationManager* object must be created and associated with a delegate. But instead of calling *startUpdatingLocation* the method *startMonitoringSignificantLocationChanges* is called. With this method the application can be suspended and is woken up automatically if a significant location change took place. This is the method proposed by Apple if location services should run in the background. It consumes much less energy as it is relying on cellular positioning only, but is much more inaccurate and therefore not suited if slight position changes must be tracked [19]. With the new multitask feature it is also possible to run the standard location service in the background. But as this consumes lots of power Apple describes it as the least desirable option [20]. Another new feature interesting for LBS developers is the new monitoring region feature introduced with iOS 4. With this new feature it is possible to monitor a shaped region which can be highlighted on a map. This is realized via the *startMonitoringForRegion* method provided by the API. When the mobile phone reaches the boundary of the monitored region either the *didEnterRegion* or *didExitRegion* will be called depending if the user wants to be notified entering or exiting the region. However this method tracks the user continuously, hence consumes a lot of battery power. As we can see developers do not have much control over the positioning hardware. Instead the API manages and encapsulates the decision which technology is used. Developers have either the option to use the more precise standard location service resulting in high energy consumption or use the significant update feature with less precision. Apple however provides no data about accuracy and precision of the positioning features of the iPhone. Results of a field test done by Zandbergen with the older model iPhone 3 can be looked up here [7]



```
- (void)startStandardUpdates
{
    // Create the location manager if this object does not
    // already have one.
    if (nil == locationManager)
        locationManager = [[CLLocationManager alloc] init];

    locationManager.delegate = self;
    locationManager.desiredAccuracy = kCLLocationAccuracyKilometer;

    // Set a movement threshold for new events.
    locationManager.distanceFilter = 500;

    [locationManager startUpdatingLocation];
}
```

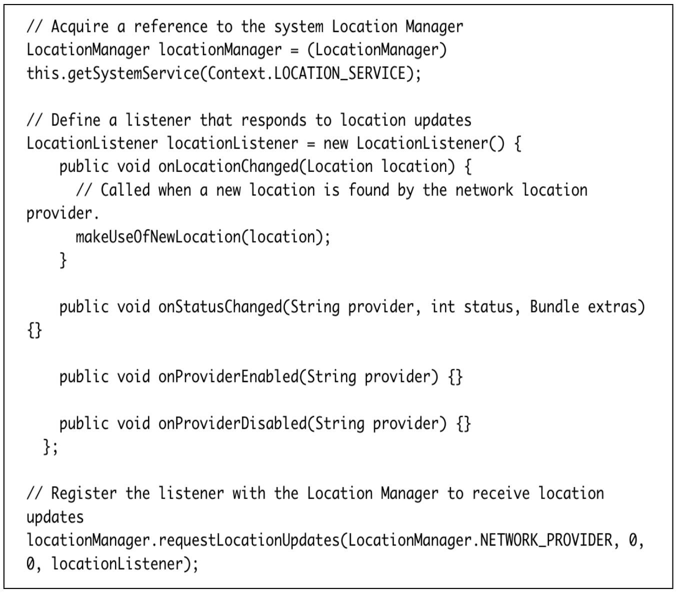
Fig. 3. Starting a standard location service in Core Location [18]

B. Google Android

Google recently unveiled the newest Version of Android which is version 2.3 now. In contrary to Apple, Google is focusing their attention on the mobile operating system

Android only, and does not produce any hardware. This means that their operating system is used on a lot different hardware devices from companies like HTC, Motorola, Samsung and others and not all positioning methods may be available on a specific phone e.g. GPS. Google use their own location databases for localization. Though it is possible for vendors of Android powered phones to use third-party localization like Skyhook, Google makes effort in preventing this [21].

Developers can obtain location data using the Android API [22] in the following way (Figure 4). First a reference to *LocationManager* must be acquired from the system. Then a new *LocationListener* object is assigned to it, which has to implement several callback methods. With the call of *requestLocationUpdates* the device begins to locate it's position. The parameters passed to this method are very essential here. The first parameter expects a *LocationProvider*. This can be either Network_Provider or GPS_Provider. The first option means that only cell and WiFi based location methods will be used, while with the second option only GPS will be utilized. It is also possible activating both with calling *requestLocationUpdates* twice with both different parameters. With the second parameter a minimum time interval between location updates can be set. The third parameter allows to set the minimum change in distance before getting new location updates. With the fourth parameter the *LocationListener* instantiated before is handed over. As we can see Android developers have a little more control over the Hardware in comparison to iPhone developers. They can decide which positioning method they want to use, in choosing the desired *LocationProvider*. Also the fact that *LocationManager* operates systemwide allowing the developers to run only the *LocationListeners* in the background is a good solution. But there are still shortcomings in terms of controlling the energy consumption and accuracy. Choosing only between two location providers is still too granular and in the effort of building energy saving applications which on the other hand can provide good accuracy. However Google doesn't provide any data about the position accuracy on Android devices.



```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
      // Called when a new location is found by the network location
provider.
      makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras)
{}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
  };

// Register the listener with the Location Manager to receive location
updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0,
0, locationListener);
```

Fig. 4. Define a Location Listener in Android [23]

C. Third Party APIs

There are a couple of third-party solutions for localization on the market like Skyhook Wireless's XPS System, SoftGPS from Devicescape, PlaceEngine, Navizon or Polaris Wireless [16]. The most prominent and most successful till now being Skyhook. Skyhook has a large amount of location data from Wireless LAN access points which they gathered through wardriving or information by private person's who can provide Skyhook their location data in form of MAC-Address and relating GPS coordinates through their website [24]. According to Skyhook their location data covers up to 70 % of the US rural areas and many places around the world with over 250 million access points in their database. Developers can download their Software Development Kit (SDK) for a number of platforms including Android, but excluding the iPhone. For platforms where the SDK is not available, Skyhook provides a portable Core Engine for developers and manufactures which can be set up on top of the underlying OS [25]. Once the SDK or the Core Engine is running on the device, developers are enabled to use a the Skyhook API for location tracking. Skyhook's XPS provides hybrid location determination with GPS, WiFi and cellular techniques and the company holds several patents most of them regarding WiFi positioning. They indicate that their hybrid system offers an accuracy of up to 10 meters with an availability of 99.8 % and a TTFF time of only 1 second [26] like shown in table I, though accurate information under what circumstances this data was collected is not provided. Also no information about energy consumption is provided. Their website gives an overview with sample codes how one time location fixes on certain platforms can be achieved, but no information regarding constant location tracking and energy saving can be found there [27]. Skyhook as already mentioned in a section before lost a big customer with Apple and it's relationship with Google is not the best, so their future remains unclear at the moment.Apple and Google seem to take the location market seriously and do not want to share it with other company's, though they might face some patent trial issues with Skyhook [28].

TABLE I
SKYHOOK' CORE ENGINE PERFORMANCE ACCORDING TO THEMSELVE [26]

	Core Engine	**GPS**	**A-GPS**
Accuracy	10 meters	10 meters	30 meters
Availability	99,8%	80%	95%
TTFF	1 sec	65 sec	30 sec

IV. LBS and Energy Consumption

As already mentioned in the introduction Location-based services are gaining more and more popularity these days. Many of these applications do need a location fix at a time or need only rough position estimation like for example the city the user is currently staying in. In this case energy consumption is not a big problem, while GPS is only turned on for a short period of time or not needed at all because the accuracy of other methods is already sufficient. However another category of applications has evolved in the last years which are known as proactive Location-based services. The most prominent examples here are navigation applications. These applications show the users current position on a map and therefore need to track him constantly while he is moving around. As high accuracy is needed in this case GPS is constantly turned on which leads to a very high energy consumption and the battery of the mobile phone will be drained quickly (Figure 5).

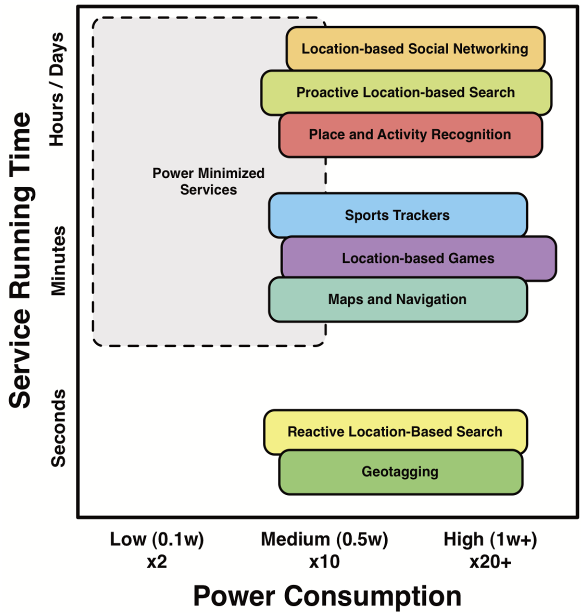


Fig. 5. Power consumption of typical Location Based Services from [29]

This problem has currently gained a lot of attention in the LBS research community [29]. One solution provided by Constandache et al. [30] is to turn on GPS only in certain time intervals and calculate the track between them with probabilistic methods. This leads to less energy consumption, but the position errors are quite high in tracks with a lot of turns. The idea has been improved by Youssef et al. with their GAC Project [31]. The main idea here is that most of the modern smartphones are equipped with compasses and accelerometers which can be used to calculate the track between two GPS position fixes. With longer time intervals between the GPS fixes energy consumption can be reduced significantly (Figure 6), though accuracy suffers. But in comparison to Enloc the accuracy achieved is better especially in urban areas with a lot of turns (Figure 7). Furthermore accuracy can be improved wit map matching where the position data calculated is mapped against a street map. This technique is already widely used in car navigation systems. Youssef et al. recently gained an technology award from Google for this project. Google also will implement this system in future versions of Android.

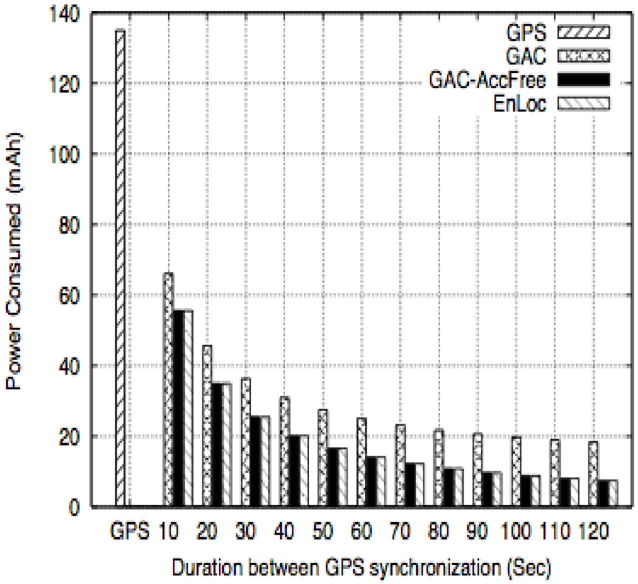


Fig. 6. Energy Consumption with GAC from [31]

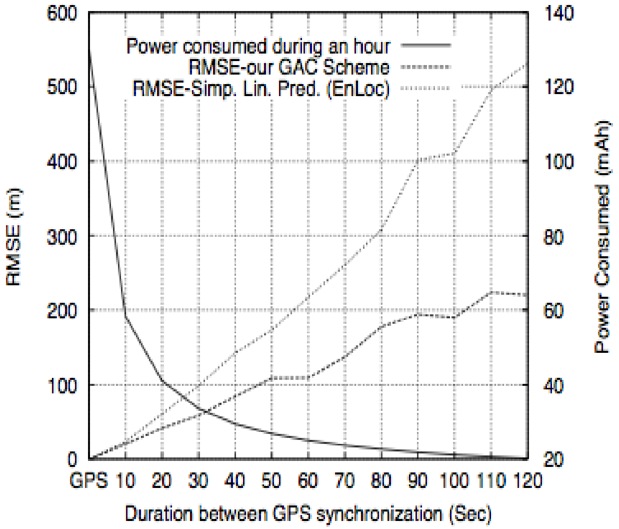


Fig. 7. Accuracy with GAC from [31]

V. Conclusion and Outlook

Positioning methods have improved a lot in the recent years, but there are still many challenges to master. GPS can be improved for indoor positioning and cellular based algorithms have to become more accurate. Of course also the infrastructure plays an important role and the density of WiFi and cellular stations will certainly increase in next years leading to better accuracy especially with WiFi. Also new technologies like LTE or the announced european satellite positioning system Galileo will play an important role in the future of positioning. On the manufacturer side Apple and Google should make positioning more transparent for developers and provide a better energy management. This could be possible with the solutions presented in this paper. A good approach would also be to implement a power simulation already integrated in the SDK, so that developers can test their applications for energy consumption already in the early stage of development.

References

[1] C. Post and S. Woodrow, "Location is everything: Balancing innovation, convenience, and privacy in location-based technologies," *Ethics and Law on the Electronic Frontier*, no. 6.805/STS.487, December 2008.

[2] Y. Zhao, "Standardization of mobile phone positioning for 3g systems," *Communications Magazine, IEEE*, vol. 40, no. 7, pp. 108 –116, July 2002, 13.12.2010.

[3] J. Hightower and G. Borriello, "A survey and taxonomy of location systems for ubiquitous computing," University of Washington, Computer Science and Engineering, Tech. Rep., 2001.

[4] G. Sun, J. Chen, W. Guo, and K. Liu, "Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 12 – 23, July 2005.

[5] P. H. Dana. (2010, December) Global positioning system overview. [Online]. Available: http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html

[6] M. Kjaergaard, H. Blunck, T. Godsk, T. Toftkjaer, D. Christensen, and K. Groenbaek, "Indoor positioning using gps revisited," in *Pervasive Computing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6030, pp. 38-56.

[7] P. A. Zandbergen, "Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning," *Transactions in GIS*, vol. 13, pp. 5–25, 2009.

[8] A. Kuepper, *Location-Based Services: Fundamentals and Operations*. John Wiley & Son, 2005.

[9] M. Ibrahim and M. Youssef, "A hidden markov model for localization using low-end gsm cell phones," *CoRR*, vol. abs/1010.3411, 2010.

[10] P. Nurmi, S. Bhattacharya, and J. Kukkonen, "A grid-based algorithm for on-device gsm positioning," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*, ser. Ubicomp '10. New York, NY, USA: ACM, 2010, pp. 227–236. [Online]. Available: http://doi.acm.org/10.1145/1864349.1864385

[11] M. Ibrahim and M. Youssef, "Cellsense: A probabilistic rssi-based gsm positioning system," Tech. Rep. arXiv:1004.3178, April 2010.

[12] K. Jones and L. Liu, "What where wi: An analysis of millions of wi-fi access points," in *Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on*, May 2007, pp. 1 –4.

[13] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000.

[14] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche, "A comparative survey of wlan location fingerprinting methods," in *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, 2009, pp. 243 –251.

[15] P. A. Zandbergen, "Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning," *Transactions in GIS*, vol. 13, 2009.

[16] U. Bareth, "Algorithmic optimizations for energy-efficient monitoring of spatial objects on smartphones," in *Proceedings of the 40th Annual Conference of the Gesellschaft für Informatik e.V. (INFORMATIK 2010)*, vol. P175. Leipzig, Germany: Köllen Druck + Verlag GmbH, Sep 2010, pp. 577–582.

[17] F. Schuil. (2010, December) Geopositioning on the iphone 4 is doing just fine without skyhook and google. [Online]. Available: http://thenextweb.com/location/2010/08/24/geo-positioning-on-the-iphone-4-is-doing-just-fine-without-skyhook-and-google

[18] (2010, December) Cllocationmanager class reference. Apple. [Online]. Available: http://developer.apple.com/library/ios/#documentation/CoreLocation/Reference/CLLocationManager_Class/CLLocationManager/CLLocationManager.html

[19] (2010, December) Location awareness programming guide. Apple. [Online]. Available: http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/LocationAwarenessPG/Introduction/Introduction.html

[20] (2010, December) Background execution. Apple. [Online]. Available: http://developer.apple.com/library/ios/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/BackgroundExecution/BackgroundExecution.html

[21] (2010, December) Skyhook:google forced motorola to drop our location service. Engadget. [Online]. Available: http://www.engadget.com/2010/09/17/skyhook-google-forced-motorola-to-drop-our-location-service-de/

[22] (2010, December) Android api locationmanager. Google. [Online]. Available: http://developer.android.com/reference/android/location/LocationManager.html

[23] (2010, December) Obtaining user location. Google. [Online]. Available: http://developer.android.com/guide/topics/location/obtaining-user-location.html

[24] (2010, December) Skyhook coverage. Skyhook. [Online]. Available: http://www.skyhookwireless.com/howitworks/coverage.php

[25] (2010, December) Skyhook device support. Skyhook. [Online]. Available: http://www.skyhookwireless.com/devices/devicesupport.php

[26] (2010, December) Skyhook performance. Skyhook. [Online]. Available: http://www.skyhookwireless.com/howitworks/performance.php

[27] (2010, December) Skyhook examples. Skyhook. [Online]. Available: http://www.skyhookwireless.com/developers/example.php

[28] (2010, December) Skyhook will take the location battle to court. Gigaom. [Online]. Available: http://gigaom.com/2010/08/03/skyhook-will-take-the-location-battle-to-court/

[29] M. Kjaergaard, "Minimizing the power consumption of location-based services on mobile phones," *Pervasive Computing, IEEE*, 2010.

[30] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. P. Cox, "Enloc: Energy-efficient localization for mobile phones," in *INFOCOM*, 2009, pp. 2716–2720.

[31] M. Youssef, M. A. Yosef, and M. N. El-Derini, "Gac: Energy-efficient hybrid gps-accelerometer-compass gsm localization," *CoRR*, vol. abs/1004.3174, 2010.