

Energy-Efficient GPS Usage in Location-Based Applications

Joy Dutta, Pradip Pramanick and Sarbani Roy

Abstract GPS is one of the most used services in any location-based app in our smartphone, and almost a quarter of all Android apps available in the Google Play store are using this GPS. There are many apps which require monitoring your locations in a continuous fashion because of the application's nature, and those kinds of apps consume the highest power from the smartphones. Because of the high-power draining nature of this GPS, we hesitate to take part in different crowd-sourced applications which are very much important for the smart city realization as maximum of these applications use GPS in real time or in a very frequent manner for the realization of participatory sensing in a smart city scenario. To resolve this, we have introduced an energy-efficient context-aware approach which utilizes user's mobility information from the user's context and as well smartphone's sensing values from the inbuilt accelerometer, magnetometer, and gyroscope of the smartphone to provide us a very close estimation of the present location of the user without using continuous GPS. It is an energy-efficient solution without sacrificing the accuracy compared to energy saving which will boost the crowd to take part in the smartphone-based crowd-sourced applications that depend on participatory sensing for the smart city environment.

Keywords Energy efficient • GPS • Location estimation • Sensor fusion
Smart city

J. Dutta (✉) · P. Pramanick · S. Roy
Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700032, India
e-mail: joy.dutta.in@ieee.org

P. Pramanick
e-mail: pradipcyb@gmail.com

S. Roy
e-mail: sarbani.roy@ieee.org

1 Introduction

The fundamental thought behind location-based information systems is to connect information pieces to positions in outdoor and indoor space. To locate anything on earth, the importance of GPS is immense as of now there are not many alternatives available other than GPS in terms of location-based services. However, in some systems, GLONASS and DARPA are used along with GPS to precise the user location.

GPS has an assortment of uses ashore, adrift, and noticeable all around. Usually, GPS is usable wherever aside from where it is difficult to get the signal, for example, inside most structures, in caverns and other underground areas, and submerged. Its use is not bounded by only driving directions, which comes to our mind first, but there are plenty of other applications of GPS which we are using in our day-to-day life. Importance of GPS is endless, and in present days, maximum user-centric applications use the context of the user which also includes its location information to provide the user more optimal and valid feedback from the various smartphone-based applications. Some most recent applications in terms of smart city include various crowd-sourced and IoT-based applications where location-based participatory sensing is used. For example, GPS is used in various crowd-sourced environmental applications like air quality monitoring or even in opportunistic crowd sensing also for the realization of smart city [6, 7]. It is also used for various IoT-based smart city [8] related applications too. Practically, use of this GPS is very common and used widely across various types of applications.

However, GPS's battery depleting conduct is most recognizable amid the underlying procurement of the satellite's route message. During this, the phone is unable to enter a sleep state, which is sometimes essential to save energy. A-GPS (Assisted GPS) partially solves this, by sending the navigational message to the mobile device over a cellular data network or even Wi-Fi. As the bandwidth of either of these greatly reduces the 50 bps of the GPS satellites, the time spent powering the GPS antenna or avoiding sleep is greatly abridged. Also, the GPS signal is weaker than the other signals. It requires more amplification stage and other signal strength, increasing circuits than other sensors. Generally, mapping software needs more processing power, and it prevents the phone from going to sleep mode, thus increasing the energy consumption of the device.

As discussed in [4], the energy consumption of a GPS is the highest among all the sensors embedded in a smartphone. It is also a known fact that smartphones are battery-powered devices, and we cannot really afford the luxury to drain the battery. Also, it is now clear that it is a very important part, and we cannot skip using GPS. Also, we need to note that, if we use GPS continuously, the accuracy will be the highest but draining of the battery will be the highest as well, which is a serious issue. So, the solution is to use the sensor optimally without much compromise with the accuracy. We need to handle this one carefully such that we can do a balanced trade-off between the accuracy and the power consumption in location-based applications.

2 Related Work

There are works on using low power sensors such as accelerometer to minimize duty cycle time of GPS [1, 5]. In some sensing scenarios, the location information can be post-processed when the data are uploaded to a server; in such cases, CO-GPS [9] allows a sensing device to aggressively duty cycle its GPS receiver and log some raw GPS signal for post-processing. In [5], authors use the location-time history of the user to estimate the user velocity and adaptively turn on GPS only if the estimated uncertainty in position exceeds the accuracy threshold. Whereas in [10], methods to reduce the amount of GPS calculations and transmissions without sacrificing the tracking capabilities of the applications are based on a state machine within which modified Kalman filter is implemented to handle the presence of noise in GPS data.

From the above discussion, we found that there is a research challenge that how to optimize power consumption without sacrificing the accuracy much. Also, we found that inbuilt smartphone sensors are not utilized properly. Nowadays, even on low budget smartphones also we have all the sensors available. So here, we plan to utilize the same properly and for that, we are using the concept of sensor fusion [12] on the smartphone-based sensors to achieve the high level of accuracy. Here, we propose a solution to find a user's real-time location by merging sensors like GPS, accelerometer, gyroscope, and magnetometer data of the smartphone more accurately. Accelerometer requires much less power than GPS and using Kalman filter, the distance can be estimated by double integration of accelerometer values [2]. Using gyroscope and magnetometer orientation of the device can be found. Now, using the orientation and estimated distance by calculating the speed, next location can be predicted.

The main contribution of our work is twofold. First, we detect user's mobility using a low duty cycle accelerometer and turn on GPS only when user is having significant mobility and second, we periodically turn on GPS for trading off between accuracy and power efficiency. The technique of periodic GPS is described in [5] but the main difference in our work is that instead of only taking speed of the user as a period tuning parameter, we estimate the next location using accelerometer along with orientation sensors (gyroscope and magnetometer) and adaptively change the period taking estimation error into account. Our approach also improves the accuracy of the trajectory between two GPS points by location estimation and map-matching.

The rest of this paper is organized as follows: the proposed approach has been discussed next followed by evaluation section. This paper ends with conclusion.

3 Proposed Approach

In our proposed approach, we are detecting user's mobility to determine whether to keep GPS on and also instead of continuously using it to get location information, GPS is periodically turned on to re-synchronize estimated location to the actual location. The details of the proposed technique are described below.

Significant motion sensor is an ultralow power sensor; essentially a virtual sensor made from accelerometer values can detect a significant change in the user's activity, thereby indicating user's mobility [3]. This sensor can be effectively used to turn off or lower the sampling rate of GPS to save power when the user is not mobile. GPS can again be turned on or set to a higher sampling rate once this sensor triggers. We have used this one for our application development.

However, this sensor is not present in older devices. But user's mobility can again be detected by low power using accelerometer. Accelerometer reports acceleration values in three directions X, Y, and Z. Continuous positive acceleration values in X or Y axis (depending upon the orientation of the device) confidently indicate movement of the user using filters for noise removal [4].

If the position and horizontal orientation of the device are fixed, which is true for many application areas including vehicle-related ones, further optimization can be done to reduce GPS usage to a much greater extent using the inbuilt gyroscope sensor. We can measure the change in angle from the gyroscope directly. In our case, accelerometer along with gyroscope is playing the key role in our approach in predicting the user's path when the GPS is off.

To explain the scenario, let us consider that the user moves from point P1(φ_1, λ_1) to point P2(φ_2, λ_2) where φ is latitude and λ is longitude. Initially, GPS is turned on and used to get location samples with a high rate for a short amount of time. Then, distance between these two points given by the law of cosines which gives well-conditioned results down to distances as small as a few meters on the earth's surface as

$$D = \cos^{-1}(\sin \varphi_1 \cdot \sin \varphi_2 + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda) \cdot R \quad (1)$$

where R is the earth's radius (mean radius = 6,371 km).

Time interval T can be calculated from timestamp. Now average speed (S) during the synchronization interval from point P1 to P2 is $S = D/T$. Now the path between P1 and P2 is divided by small time interval t where t is the sampling period the application requires. Let d is the distance covered in time t. Again, intermediate location coordinates starting from point P1 over a smaller distance d can be estimated which essentially gives the GPS trajectory. The smaller distance interval, d can be found from accelerometer. Let a be the acceleration along the axis in which the user is moving, and v is the velocity assuming user goes straight in such a small interval. Then, v and d can be found by

$$\begin{aligned}
 v &= \int a \, dt \\
 d &= \int v \, dt
 \end{aligned}
 \tag{2}$$

By taking distance values from y axis of the accelerometer and orientation change in radians over small time interval t of the total synchronization interval T , and by summing up all these, we get the final estimated location $P2' (\varphi2', \lambda2')$ for a larger time interval T using Eq. (3) below:

$$\begin{aligned}
 \varphi2 &= \{ \sin^{-1}(\sin(\varphi1) * \cos(d/R) + \cos(\varphi1) * \sin(d/R) * \cos(\theta)) \} \\
 \lambda2 &= \lambda1 + \tan^{-1} \{ (\sin(\theta) * \sin(d/R) * \cos(\varphi1)), (\cos(d/R) - (\sin(\varphi1) * \sin(\varphi2))) \}
 \end{aligned}
 \tag{3}$$

Initially, we are checking whether the user is mobile or not, based on the significant motion sensor or accelerometer, when the former is unavailable. When the user moves and the re-synchronization timeout occurs, then we are taking user's location based on GPS. If the user is not moving at that instant, then it waits for user's movement and then collects location information. And set the synchronization interval based on the context of the user. Between each synchronization, we take the accelerometer and gyroscope values and internally estimate distance in smaller time interval and velocity in that interval from Eq. (2). Using these calculated values, we are calculating next point from Eq. (3) until next re-synchronization interval occurs which can be further improved using opportunistic calibration method if you consider driving a car [11]. However, note that when the map-matching algorithm is applied along with our proposed approach for real-time location estimation, it is giving us significantly better result as shown in Fig. 1 (theoretically) which has been cross checked real time in the evaluation section. If the user is not within the synchronization interval, then we repeatedly check the status of the user and accordingly set the context for the user. After each re-synchronization interval, we are calculating actual distance traversed between two points using Eq. (1) which in turn helps the system to predict more accurately the next synchronization interval in the next turn.

If the user traverses in a straight line, then the difference between estimated locations $P2'$ and actual location $P2$ is usually smaller than curved path. Here, the re-synchronization interval is very crucial. For a longer re-synchronization interval (T), the error is estimated with respect to the present location of the user tends to be high. So, choosing this interval is balancing the accuracy vs. battery consumption which requires attention. To make the system intelligent, we make this interval context-dependent, i.e., based on the user's speed and traveling mode, this interval changes internally in a dynamic way, which is again doing an excellent job in terms of accuracy. Here, in each synchronization phase, GPS is turned on to get actual

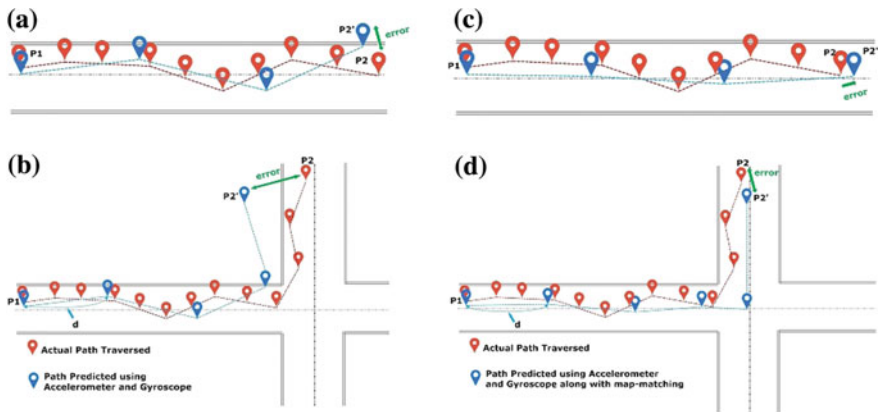


Fig. 1 Path estimation without map-matching in straight road segment (a), turning section of a road and with map-matching in (b), straight road segment (c), and turning section of a road (d)

location, and user's current location is set to this to correct the error. Also, new average speed is calculated after each T period.

If the path is not straight and a curved one, then the difference between $P2$ and $P2'$ can be large due to the combined noise of accelerometer, gyroscope and magnetometer. The GPS trajectory estimated by Eq. (3) is further improved by interpolation.

4 Evaluation

We have tested the proposed approach in real-life traffic conditions in the Jadavpur University area and get the results as shown in Figs. 2, 3 and 4. Here, we have denoted real GPS traversal path to be in red color, and the path that we are calculating is set to be in blue color. Also note that $P1$, $P2$, $P3$, and $P4$ points shown in the map outputs are re-synchronization points, where estimated location synchronizes with actual GPS-based location.

Initially, we have taken straight road and tested how our proposed approach is performing in real life. We tested the synchronization interval to be 2 min, and we found the estimation is very close to the actual GPS whereas when we make the interval 4 min, the error in the estimation slightly increases but still very much accurate and valid for our everyday applications as shown in Fig. 2a, b. Next, when the proposed approach is merged with map-matching we get the result as shown in Fig. 2c, d which is actually improving system's accuracy.

Next, we have tested the efficiency by taking a left turn while walking. Here, also we have considered three different types of synchronization interval to validate our claim. It is clear from Fig. 3a, b that accuracy is high if the synchronization interval is small.

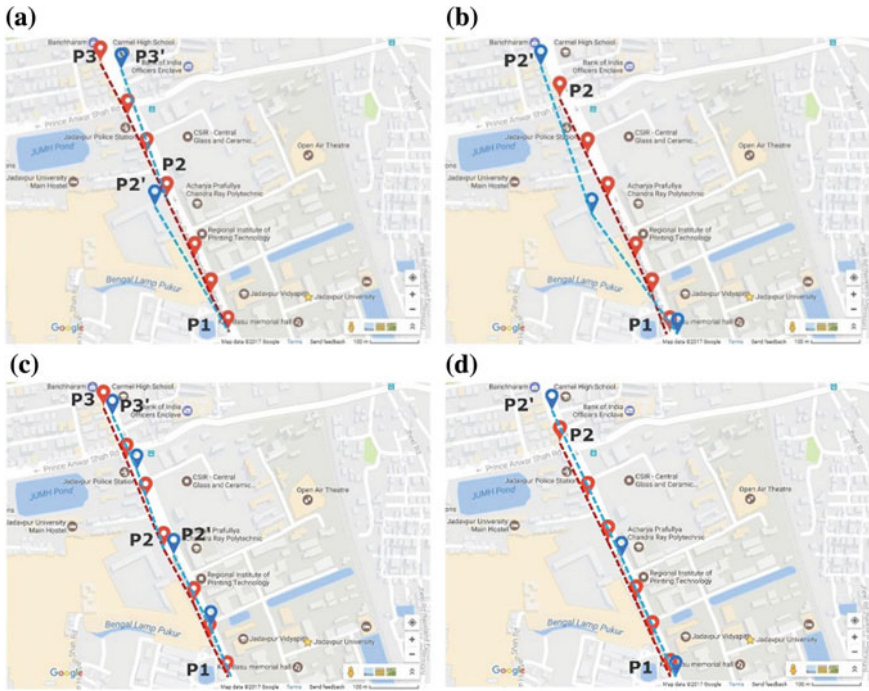


Fig. 2 Case studies: Path estimation without map-matching when user walking (a), biking in a straight line and path estimation with map-matching when user (b), walking (c), and biking in a straight line with 2 min synchronization interval (d)

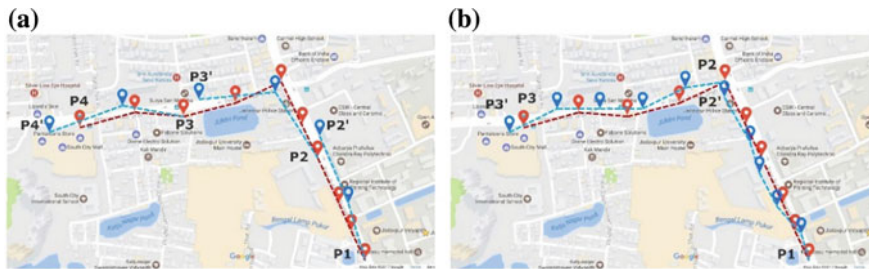


Fig. 3 Case studies: Path estimation without map-matching when user taking turn while walking with GPS synchronization interval 2 min (a), and 4 min (b)

Also note that in these types of cases, map-matching is not essential as the speed of the user is low. So, using only accelerometer and gyroscope, we can estimate the location of the user efficiently as shown above. As the walking speed of a human is not very fast, our estimated location is very much closer to the actual location point of any individual. Thus, our proposed scheme is very much valid from this

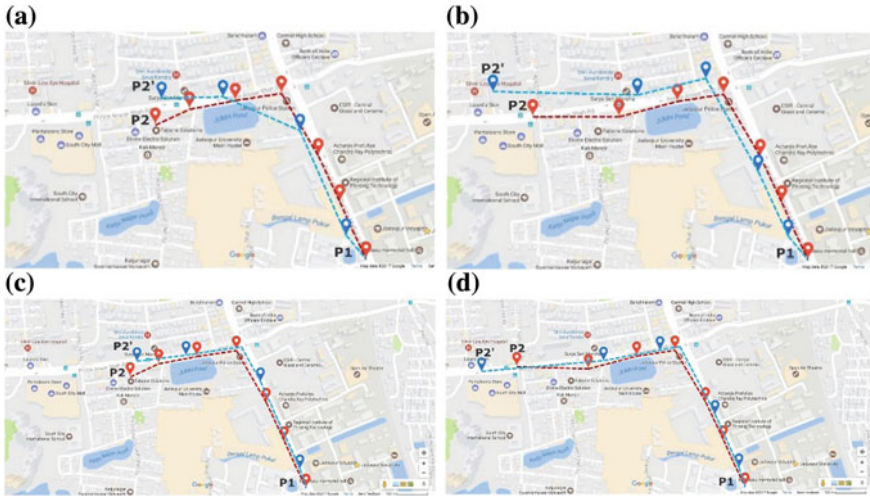


Fig. 4 Case studies: Path estimation without map-matching when user taking turn while biking with GPS synchronization interval 2 min (a), 4 min and path estimation with map-matching when user taking turn while biking with GPS synchronization interval (b), 2 min (c), and 4 min (d)

viewpoint also. However, even if we apply map-matching here, that will improve the accuracy which is of no harm. Practically, when the synchronization interval increases, we are becoming more dependent on the accelerometer and gyroscope values. However, it is clear from Figs. 2, 3 or from Fig. 4 also that without using map-matching also, the system is proving us very much descent result.

Now, the crucial part is predicting the location of a moving car and validating the same. This was the most challenging task. Here, due to the speed factor, the collected GPS points on the road are less as compared to walking because we have used the same sampling rate for all the cases. Now, we can see from Fig. 4 that the low synchronization interval gives better accuracy, as expected, when compared to all time GPS on a system which is acceptable in most of the applications.

According to this research result, turning the GPS periodically on/off along with intelligent sensing does not affect much the accuracy as we are gathering information from other energy-efficient sensors.

5 Conclusion

Here, we have used the GPS sensor optimally without affecting location accuracy of the user using the concept of sensor fusion. As GPS is a power-hungry sensor, our approach is to add intelligence in the system such that the system can read the user, understand the state, and work accordingly without affecting the performance. Here, we have introduced the context-aware sensing which, when assembles with

accelerometer, magnetometer, and gyroscope, is providing us almost the same result as if you are using GPS continuously. Here, using different real-life test cases, we have validated the same by using the location estimation and map-matching for the trajectory between different GPS points.

Acknowledgements The research work of the first author is funded by “Visvesvaraya PhD Scheme, Ministry of Communications & IT, Government of India”.

References

1. Abdesslem, F.B., Phillips, A., Henderson T.: Less is more: energy-efficient mobile sensing with senseless. In Proceedings of the 1st ACM workshop on Networking, Systems, and Applications for Mobile Handhelds (2009)
2. Singhal, T., Harit, A., Vishwakarma, D.N.: Kalman filter implementation on an accelerometer sensor data for three state estimation of a dynamic system. *Int. J. Res. Eng. Technol.* (2012)
3. Muthohar, M.F., Nugraha, I.G.D., Choi, D.: Exploring significant motion sensor for energy-efficient continuous motion and location sampling in mobile sensing application. *Int. J. Technol.* 38–49 (2016)
4. Kjærgaard, M.B., Langdal, J., Godsk, T., Toftkjær, T.: Entracked: energy-efficient robust position tracking for mobile devices. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services. ACM, New York, USA (2009) 221–234
5. Paek, J., Kim, J., Govindan, R.: Energy-efficient rate-adaptive GPS-based positioning for smartphones. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services. ACM, New York, USA (2010) 299–314
6. Dutta, J., Gazi, F., Roy, S., Chowdhury, C.: AirSense: opportunistic crowd-sensing based air quality monitoring system for smart city. In: Proceedings of IEEE Sensors. Orlando, FL, USA (2016). <https://doi.org/10.1109/icsens.2016.7808730>
7. Dutta, J., Chowdhury, C., Roy, S., Midya, A.I., Gazi, F.: Towards smart city: sensing air quality in city based on opportunistic crowd-sensing. In: Proceedings of the 18th International Conference on Distributed Computing and Networking. Hyderabad, India. ACM. (2017). <https://doi.org/10.1145/3007748.3018286>
8. Dutta, J., Roy, S.: IoT-fog-cloud based architecture for smart city: prototype of a smart building. In: Proceedings of 7th International Conference on Cloud Computing, Data Science & Engineering. Noida, India, pp. 237–242 (2017). <https://doi.org/10.1109/confluence.2017.7943156>
9. Liu, J., et al.: CO-GPS: energy efficient GPS sensing with cloud offloading. *IEEE Trans. Mob. Comput.* 15(6), 1348–1361 (2016). <https://doi.org/10.1109/TMC.2015.2446461>
10. Taylor, I. M., Labrador, M. A.: Improving the energy consumption in mobile phones by filtering noisy GPS fixes with modified Kalman filters. *IEEE Wireless Communications and Networking Conference, Cancun, Quintana Roo, Mexico 2006–2011.* (2011). <https://doi.org/10.1109/wcnc.2011.5779437>
11. Khaleghi, B., El-Ghazal, A., Hilal, A. R., Toonstra, J., Miners, W. B., Basir O. A.: Opportunistic calibration of smartphone orientation in a vehicle. In: *IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks.* Boston, MA, USA (2015). <https://doi.org/10.1109/wowmom.2015.7158210>
12. Abyarjoo, F. et al.: Implementing a sensor fusion algorithm for 3D orientation detection with inertial/magnetic sensors. *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering.* Springer, Cham pp. 305–310 (2015)