International Journal of Advances in Computer Science & Its Applications – IJCSIA
Volume 4: Issue 4 [ISSN 2250-3765]
Publication Date : 27 December,2014

Mobile Geofencing Applications: Design Issues and Performance Evaluations

Siu-Ting Wong and Sheung-On Choy

***Abstract*—Geofencing is becoming more and more important, and there is a lack of standards or metrics to evaluate whether an application is designed appropriately and performs well enough. In this research, an application is built to test about functions of geofencing in Android OS, and it is also a tool for assisting the evaluation of performance. Three performance metrics, namely power consumption, accuracy and responsiveness, are studied. To study the power consumption, an indirect approach is adopted, which is measurement of power endurance of the devices, instead of direct measurement of actual power consumption by the application. Field testing is used to study the accuracy and responsiveness of the application. The test is performed in the format that testers bring along the device, which has the testing application being installed, to walk through a path going through fences' boundaries which are set before testing. Different location services are enabled and disabled alternately and repeat the test a few times, in order to study the difference of performance between distinct positioning approaches.**

***Keywords*—geofencing, android application, design issues, performance evaluations, power consumption, power endurance, accuracy, responsiveness**

I. Introduction

Mobile technologies develop rapidly in recent years, especially after the born of Android OS and iOS. This thriving trend has catalyzed the development of mobile applications, functionality of applications keep broadening. Location-based services (LBSs) are one of the most innovative features in these applications, which provide suitable user-desired functions or information automatically, according to users' current location. Geofencing is a relatively new family member of LBSs in the previous mentioned operating systems (OSs), and people generally believe that geofencing will become more and more useful and essential in near future by considering its wide-range of usages.

General definition of geofencing refers to the concept that users first define some virtual boundaries around particular geographical areas. Once a transition over a boundary is detected, devices would make notification, or perform some users desired actions. When it comes to smartphones, applications with geofencing features are supposed to run in the background, which means the phones can monitor the transition over a boundary and perform other tasks at the same time. This introduces another challenge for application developers, as they are impeded by the battery capacity, which implies that requests for locating current position should be reduced or otherwise those phones would easily run out of power.

Google Play services provide various features in Android OS, including location-based services. Geofencing is included in the Google Play services since the version 3.1, and devices starting from Android OS 2.2 platform (Froyo) are supported [1]. The Google Play services provide an interface of geofencing for application developers, and this greatly reduces the difficulties for developers to build a geofencing related applications. No need for developers to implement their own geofencing framework and algorithms allows them to focus on other matters.

In order to develop meaningful geofencing applications, algorithms determining users' location around some virtual boundaries are required to be designed well, so that minimum power is consumed and hence devices could run other applications meanwhile as long as possible. On the other hand, location information need to be accurate, actions taken should be responsive. In other words, applications cannot consume much power, no matter the application is running in foreground as well as background, while having least effect on locating beyond those virtual boundaries.

However, GPS sensor used for locating is usually a power-hungry component [2], which especially affects the endurance of a mobile device if it is frequently activated. Therefore applications are not expected to request GPS information continuously, which means that applications must release GPS resource and regain it after a period of time from time to time. Accuracy is hence reduced and seems cannot coexist with power endurance in this case.

Therefore, a good design of geofencing algorithm is very important for applications to work well and properly, otherwise applications may be not responsive enough to the transition of boundaries, or consume a great amount of energy in a very short period of time. The following will introduce two approaches in designing an algorithm, one of them switch between different positioning technologies and makes use of motion sensors, while another team design some scheduling strategy to determine the timing of sensors activation.

Ryoo et al. [3] try to classify all cases into 3 situations. "Fence-level state" is determined by their algorithm, which aims at using another locating technology, cell-tower triangulation to replace GPS positioning as much as possible in different scenes. In addition, "motion-level state" manages the activation of fence-level state. According to the fence-level state, different positioning methods will be applied. They even apply motion sensors, instead of any positioning method, to

Siu-Ting Wong, Sheung-On Choy
Engineering Science Team, the Open University of Hong Kong
Hong Kong

estimate the location in some scenes. Their design works successfully, but their classification of motion-level state requires further improvement. An accurate prediction strategy seems essential on determining the situation to enable GPS sensor.

On the other hand, Man et al. [4] successfully reduce the GPS sensor usage by 90% using their scheduling strategy, which determine the time to wakeup according to the distance between geofence and the devices. However, they do sacrifice the accuracy. One limitation of their work is sensors delay, which affects the estimation result. An algorithm to overcome this constraint is required, or we may need to use sensors with less delay.

Of course there are many other approaches that may enhance the performance of a geofencing application, but it is hard to ensure that all algorithms or approaches will do a good job as expected. And hence in this research, we try to evaluate performance of a geofencing application, in three dimensions, which are power consumption, accuracy and responsiveness.

II. Methodologies

A. Application Building

A prototype is built and different features are added to it, which is used as trial and testing purpose. After studying APIs, testing is also required, in order to examine the actual result when it comes a practical case, but not a theoretical one. Simulation of functions and some features that are combined from other APIs are important, which can examine the compatibility between them. Feasibility of some concepts can be tested on a prototype as well, which ensure the project would not work on a wrong direction. Besides, a visual preview of the application sometimes stimulates other ideas as well. Prototyping is another kind of record of the project progress. Different APIs are simply combined into one prototype at once, and improvement is made from time to time. The process of improvement or making changes on the prototype is valuable for future development and reference.

To test the feature of prototype, real devices testing is much preferable. For the reason that an application simulator on a laptop or desktop fails to simulate the reality environment, testing on a real Android device becomes an essential step. Besides, each function or component from different APIs is tested separately before they are gathered. This approach allows developers to combine or separate functions or components from various APIs effectively later on.

When designing an UI, developers need to consider the usability of the application. The UI must be straight forward to users, so they do not need to put so much effort on learning how to use the application. Besides, since various APIs would be combined to work together, redistributing different features into various pages allow clearer division of works of different features. Grouping different parts of the prototype into different categories such as information, user interaction, data storage and so on. This helps for future implementation. Actions react to the transition beyond virtual boundaries should be as simple as possible in this stage, in order to prevent unnecessary error to the prototype. Functionality is not a main concern at this stage.

B. Performance Evaluation

Since a geofencing application will require Google Play services, it is not easy to separate and record those services called by a specific application. Hence indirect measurement to power consumption, power endurance is adopted to indicate the power consumption of an application. One advantage of using indirect measurement is that it is easy to obtain a control sample or in other word, a reference, by simply reset the testing device to format factory and not install any custom application, and allow it to run. A reference helps us to analyze the results of other data sets effectively.

On the other hands, field testing is performed to evaluate accuracy and responsiveness. The reason for adopting field testing is that simulator may not reflect actual situation since there are too much external factors in real world which are difficult to be reproduced and simulated, and those external factors could greatly interfere the results. Multiple times of field testing are performed and statistical tools will be used to help to analyze the resulting data.

III. Implementation

A. Features Detail of the application

- Custom Geofence: a custom geofence class is implemented, the class contain all information that the original geofence class required. The reason for implementing a custom geofence class is that it is more convenient to manage if there are other extra information or setting are related to one specific geofence, such as actions in response to a transition. Those information are usually specifically describing a particular geofence, using other medium to save may confuse developers.
- Geofence Store: a class is built to manage all custom geofences, developers can create, read, update or delete record through an instance of this class. Not only for saving custom geofences, some temporary settings are also saved or updated through this class. Generally to say, this class act as the manager of data storage in the application.
- Geofence Adder/Remover: the two classes are implemented, for the duty of keep adding or removing geofences automatically. The reason for implementing these classes is that sometimes not only one geofence is being added. Since Android supports for mass geofences registration, requirement to add/remove multiple geofences would be a tedious task if developers need to handle and add/remove all geofence separately. Hence a geofence adder/remover makes the task easier for developers.
- SMS: SMS is one of the actions that responding to a transition. At the moment users create a geofence, if

SMS is selected as the action in response to transition, they are required to provide the receiver phone number and content they wish to send out. Those settings are saved to respective custom geofence instance and will be retrieved once a transition is detected, and a SMS is sent out automatically.

- Alarm: alarm is another action respond to a transition. Users can either choose to send a SMS or make an alarm. In this project, since the functionality of alarming is not the concerning part, not much setting is provided for users.
- History Recorder: since studying the performance is the objective of this project, it is essential to have a tool to record all transitions, in order for tracing afterwards. Things being recorded include the geofence id, type of transition, time and position that the device detects a transition.

B. *Performance Evaluation*

Before any tests, some preparations are required to do. First, the device being tested is reset to factory default. This practice ensures no other factors would interfere the results. Then, the application developed in this project is installed and two geofences are added. Geofences added would have the same center point, but different radius. Fig. 1 shows the two geofences used in the project. OUHK is set as the center point, the latitude and longitude of that point is 22.31613983 and 114.18034192 respectively, and the radii of them are 100m and 200m.

Figure 1. Two geofences used for testing

There are 7 cases that are tested. Table I shows that cases are derived according to different location providers being adopted. The last case is to act as a control, in which all services are disabled. Android provides two choices for users, which are GPS satellite and network positioning only. In order to set up the environment, hardware that is not the testing target is disabled in order to satisfy the testing environment.

TABLE I. TABLE TYPE STYLES

Case no.	Location Provider
1.	All location providers
2.	Wi-Fi only
3.	GPS only
4.	3G only
5.	Network (3G + Wi-Fi)
6.	AGPS (3G + GPS)
7.	None of location providers (control)

To evaluate power consumption of the application, the first thing is to charge the device to 100%. For the reason that number displayed may not be accurate, once the battery is said to be 100% full, the device will be left along for charging for at least 3 hours. Another approach is to use a battery monitoring application to monitor the current battery capacity. After the battery is fully charged, tester brings along the device and allowed it to run. Tester is required to have a regular life style, which means that he or she would keep go in and go out the fence regularly. This is to prevent inaccurate results are obtained. The device is allowed to run, until it reaches a low battery level. Tester will record the day the device has lasted for. One system tool that is able to show the time the device lasts for is used.

To evaluate the accuracy and responsiveness, it is more suitable to let users to carry the device and test it in a real situation, and test about the offset of the time and position. Refer to the two fences created in Fig. 1, the route for the test is designed as Fig. 2. Crosses in Fig. 2 are the reference points, testers are asked to record the time they pass through those points every time. After testing, those records will be compared with the data stored in the history, and work out the offset, of the time and position the device detects transition. The process is repeated a few times, and each time there are two entries and two exits. The remaining task is to work out the mean and the standard deviation, and then plot those data on a graph.

Figure 2. Route during the test

IV. Results

A. Graphs

Fig. 3 to Fig. 5 are graphs showing the results of data collected in this research.

B. Observation

Refer to Fig. 4 and Fig. 5, they show the mean and the standard deviation of distance and time offset, and a phenomenon is observed from the graph. Cases within Wi-Fi (Cases 1, 2 and 5) always have response with various time delays, while cases without Wi-Fi (cases 3, 4 and 6) are hard to observe any response towards a geofence. Fig. 3 shows the days that the device last long when the application is running in the background. The power consumed basically according to services enabled, except two cases, which activate GPS and APGS only (cases 3 and 6).

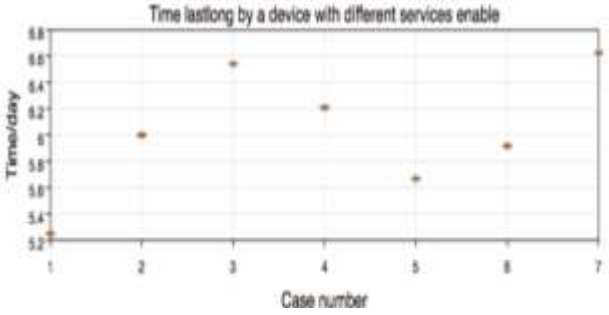

Figure 3. Power Endurance

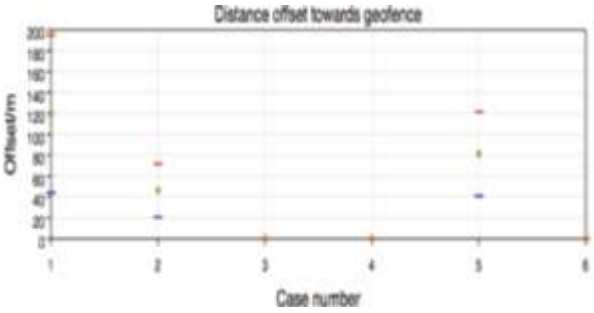

Figure 4. Distance offset

Figure 5. Responding Time offset

V. Discussion

A. Wi-Fi Positioning

Since the tested target is an urban area, where high density of wireless APs are available, Wi-Fi positioning (cases 1, 2 and 5) always have response. This situation allows Wi-Fi positioning works best and hence always has response. Wi-Fi positioning work well in urban area and have median error range, which is about 50m [5]. The error is relative small compared to the radii of the two geofences.

B. Non Wi-Fi Positioning

There are difference between cases 3, 6 and case 4. Cases 3, 6 do not have any response throughout the test process, while case 4 detects a transition of entry or exit, at the time the geofence is registered. Reason for cases 3 and 6 should be the strategy of geofence service prevent calling for GPS, in order to allow the device last as long as possible. Without proactive call for GPS service, Android's geofence framework prevent the use of GPS component, hence it would be much less responsive, even no response no matter someone enters or exits a geofence and stays for a long time. On the other hand, case 4 does have response to transition at some, but does not in other time. Reason for this phenomenon should be the geofence radius is too small to tolerate the error that Cell-Id positioning may have. Since the Wi-Fi component are forced to close, only dispersed cell towers are available, this greatly reduce the accuracy. When the accuracy is low, the responsiveness become low as well, since there is a great correlation between accuracy and responsiveness. High accuracy implies that there is higher chance that the result will reflect the actual position of users, and hence the responsiveness will therefore become higher if reasonable

number of fix is requested in some period. Otherwise, low accuracy means the chance of reflecting the users actual position is low, so if the radius of a geofence is not large enough to tolerate the error, if would result in not having any response.

C. Design Issues

After the development of a mobile geofencing application, followed by its performance evaluation, some properties and design issues of the build-in geofence can be outlined. First, GPS is rarely used as the method to obtain position in order to achieve power efficient, so developers need to take care the case that if only GPS is available, in order to prevent the application become nonfunctional. Or other approach is to provide an option for users, which leave the right of choosing suitable strategy to users. Second, error of positioning method does have significant effect on the accuracy and responsiveness, especially in the case the radius of geofences are smaller than the error range. No matter GPS, Wi-Fi or Cell-Id positioning are adopted, developers may need to set a constraint for the choice of radius size for users, according to current available positioning method, this improve the performance of the application, as well as prevent the users from setting a unreasonable small geofence. Not only at the time a geofence is set will be monitored, but also the moment that users disable some position providers should be concerned. The application may listen for the change of different location providers' availability, and prompt a remind dialogue if there is a need to modify the geofence radius.

D. Limitations

There are some limitations for the evaluation approaches adopted. First, the approaches to evaluate accuracy and responsiveness require a real device, as well as someone to take the device to go in and out some geofences. In this project, the radii of geofences are only 200m at most, which means tester is required to walk around four to five hundred meters in one iteration. Because of the budget is tight, only one device is available, and hence each time only one tester is allowed to perform the test. Efficiency is not good if the route being tested expended from a few hundred meters to a few kilometers, while it is hard for the tester to perform such a long walk so many time, or ask other tester to wait until he comes back. The situation will be eased if more budgets are allowed, which means that more devices, or even using transportation are possible. Another limitation in the project is that there is not any effective approach that can directly measure the energy usage by an application if some system features, such as Wi-Fi and network are used. An indirect approach to estimate the energy usage is adopted in this project. Inadequate information may be obtained, while there is not any method to validate the data. One measure is taken in the project to mitigate effect from other applications, which is reset the device to factory default. This action reduces the chance that the battery is consumed by other background application and causes an inadequate result.

VI. Conclusion

A. Conclusion

Throughout the report, issues of designing a geofencing application are studied, and some supporting technologies are studied as well. Besides, this research tries to study further into the performance of an application, and evaluate by experiment. Although results obtained differ from the prediction, it is a motivation that encourages us to test the geofence thoroughly.

B. Future Works

Although some approach used to evaluate performance is introduced and applied in the project, data being collected are not enough to define metrics about the performance. Much more samples should be obtained and analyse, in order to conclude a reasonable and acceptable metrics for other developers to take reference. Hence, a database may be built to store such sample data, not only from one tester, or one region, but may be collectively work out by many testers and researchers from all over the world. The reason is that not only urban areas require geofencing, but also some rural areas which may find dangerous and not suitable for people to get close to those areas. It is hard to perform such a test in Hong Kong, but easy in other countries. Besides, the approach to evaluate the power consumption is not a good design. In the project, due to the difficulties to obtain the actual energy consumed by the geofencing application in a direct way, an indirect way that measure the run time of the device is adopted. If there are other ways to obtain the information directly in future, other researchers are not suggested to adopt this approach, since there are many factors would interfere the results, especially the device is allowed to run in such a long time. Number of transition detected, or other background application make different degrees impact on the results obtained..

References

[1] Social Gaming, Location, and More in Google Play Services | Android Developers Blog,
http://android-developers.blogspot.hk/2013/05/social-gaming-location-and-more-in.html

[2] J. Paek, J. Kim and R. Govindan. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In Proceedings of the 8th international conference on Mobile systems, applications, and services. San Francisco, California, USA, 2010.

[3] J. Ryoo, H Kim and S.R. Das. Geofencing: Geographical-fencing based energy-aware proactive framework for mobile devices. Quality of Service (IWQoS), 2012 IEEE 20th International Workshop on.

[4] Y. Man, E.C. Ngai and Y. Liu. A glimpse of energy-efficient location-triggered mobile application design and implementation in collection scenarios. Mobile Ad-hoc and Sensor Networks (MSN), 2012 Eighth International Conference on
Digital Object Identifier: 10.1109/MS

[5] Bareth, U., "Simulating Power Consumption of Location Tracking Algorithms to Improve Energy-Efficiency of Smartphones," Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual , vol., no., pp.613,622, 16-20 July 2012 doi: 10.1109/COMPSAC.2012.87