Efficient Spatial Tracking for iOS Devices

Georgios Stylianou
European University Cyprus
6 Diogenes Str. 1516, P.O. Box 22006
Nicosia, Cyprus
g.stylianou@euc.ac.cy

ABSTRACT

We propose a simple and effective yet novel method designed for iOS smartphones that autonomously tracks its user's destinations and sparse trajectory. The smartphone application manages to do that 24/7 unobtrusively in the background and without consuming noticeable by the user battery energy. In this article we present the method and our highly encouraging early experimental results.

Keywords

trajectory and destination tracking, iOS, energy efficient

1. INTRODUCTION

Smart phones that can unobtrusively learn their users, can later provide to them smart personalized services. One aspect for every smartphone user is its spatial transitions. Trajectory tracking has been studied during the last five years and several solutions have been proposed. During these studies it was demonstrated that the smartphone's GPS sensor is energy hungry. As a result the proposed methods attempt to become energy efficient by minimizing the total time the GPS sensor is powered up. These solutions include combinations of accelerometer, GPS, cell tower id, Wi-Fi signals and bluetooth and run only on android and older Nokia devices.

It is unquestionable that the energy reduction they achieve is significant. However, it is also true that they exhibit the following issues: (a) these methods cannot run in the background of iOS devices, (b) the iOS development API does not allow to query cell towers or Wi-Fi access points for signal strength or the use of bluetooth in the background.

2. PROPOSED METHOD

Our aim is to develop a method that can run solely in the background of an iOS smartphone and can do (a) sparse trajectory tracking, (b) destination tracking, and (c) minimize the energy footprint.

2.1 iOS Location Services

The CoreLocation framework, part of iOS development API, provides two approaches for acquiring the smartphone's current location: (a) the standard location service that uses the GPS sensor and (b) the significant location change service that uses other means such as cell towers and Wi-Fi access points. When using (a) the smartphone provides locations continuously or every some distance (see distanceFilter). It also provides two parameters that when used properly can aid in energy savings. These are the 'desired Accuracy' (e.g. in meters) and the 'activity Type' (e.g. automotive navigation). When using (b) the smartphone provides a location at a minimum of every five minutes or every 500 meters of displacement. This service uses very low energy as it doesn't power up the GPS sensor, but rather relies of cell towers and Wi-Fi access points. This service is suitable for applications that want to acquire the user's initial location and then want to know when that location changes.

Our experiments showed that neither of the services can alone properly provide destination and/or trajectory tracking. The standard location service provides too many locations (fig. 1, Left) and is energy hungry where as the significant location service provides too few locations (fig. 1, Middle) and several times fails to capture a destination because movement stopped prior to the next location update. As a result a hybrid approach (fig. 1, Right) can overcome the limitations of each service and offer both good location and trajectory and destination tracking while minimizing the energy footprint.

2.2 Proposed Approach and Initial Results

Our approach aims to optimize (a) the interplay between the standard location service (denoted as GPS) and the significant location service (denoted as SLU) and (b) to satisfy the strict conditions of the iOS operating system (OS) for background running applications (apps).

In algorithm 1, we provide our approach. It is important to explain some important aspects of it.

1. At any given time we activate either the GPS or the SLU service and never both services. Consequently we always know which service provides a new location.

2. We know from the CoreLocation framework's guidelines and our experiments that SLU provides new locations in a variable frequency that depends on the density of the cell towers. We have experimentally discovered that in rural areas where cell towers have much less density that location updates are much sparser

MOBIQUITOUS 2015, July 22-24, Coimbra, Portugal

DOI 10.4108/eai.22-7-2015.2260189

Figure 1: Trajectory tracking examples. Left: GPS service using distanceFilter=$500m$. Middle: SLU service. Right: Proposed approach that combines both services.

than in urban areas. For example, we observed that for a specific rural area the new location is provided after 16 minutes for a distance of about $13Km$. In order to overcome this problem we schedule GPS location updates to occur t seconds (e.g. $t = 120$) after each SLU location update. Then, if we acquire two or more consecutive GPS locations that are accurate, we compute the distance between the last three locations. If "no" distance (e.g. $< 100m$) was traveled then we don't schedule for more GPS location updates, otherwise we continue scheduling for more. This feature serves two purposes: (a) to ensure we can get the location of a destination (when "no" distance was traveled) and (b) to reduce the gap between infrequent SLU locations, thus enabling among others better total distance computation and more accurate, although sparse, trajectory tracking.

3. We discard inaccurate locations. We consider a location as inaccurate when it has intolerable positioning error and this depends on whether the smartphone is connected to a Wi-Fi access point or not. The reason is that there are occasions that SLU tends to acquire locations with large errors of the range of 500 meters to some kilometers, even though the smartphone is stationary. These errors are discarded when the phone is connected to a Wi-Fi access point for which we have a known good position. As large errors may also occur in rural areas due to the lower density of cell towers we keep these locations unless we get a more accurate location, with a similar timestamp, using GPS.

Regarding (b) iOS OS may suspend a background running app for several reasons including consuming resources needed by other apps or expiration of its background running time. In addition, we experimentally observed among others that (i) if a timer is scheduled to fire in the background after 3 minutes, it will not necessarily fire in 3 minutes but whenever the iOS OS allows it to and (ii) if an app runs switches on the GPS service with $distanceFilter = 0$ in the background, this app gets suspended after a $4 - 8$ location updates. As a result, we had to overcome these strict rules, many of them undocumented, in order for the app to be consistently running in the background.

We have evaluated the performance by doing experiments: (a) to evaluate the ability of trajectory and destination tracking and (b) to measure the energy footprint. Figure 1, right shows visually the improvement of trajectory tracking over just SLU (Middle) for a routine day of the author. We observe that several routes are more accurately approximated. It also, manages to capture all destinations: home, work and friends house and arrivals and departures from each destination. We have also measured the energy footprint of our approach by measuring the battery consumption for an idle iPhone 4 and the same phone running our app on the background. These early results reveal an extra 0.12% battery consumption per hour or extra 2.88% for 24 hours. It is important to note that as battery consumption is related to the amount of spatial transitions, more thorough results will reveal more precise numbers.

```
set GPS_Loc=0; i=0, j=0;
while true do
    get new location $P_i$;
    if P_i is SLU location then
        set GPS_Loc=0;
        if location is Accurate then
            save location to L_j;
            increase j;
            set timer to swap service in $t$ seconds;
        end
        else
            discard location;
        end
    end
    if P_i is GPS location then
        if location is Accurate then
            save location to L_j;
            increase j;
            stop GPS and start SLU;
            increase GPS_Loc;
            if $GPS\_Loc >= 2$ then
                if $distance(L\_j, L_{j-1}, L_{j-2}) >= D$ in Km
                then
                    set timer to swap service in $t$ seconds;
                end
                Else  set GPS_Loc=0;
            end
        end
    end
    increase i;
end
```

Algorithm 1: Overview of the spatial tracking algorithm.

3. CONCLUSION AND FUTURE WORK

We described the early results of a background running destination and sparse trajectory tracking service designed to run efficiently on iOS devices. Even though these results are highly encouraging as this service provides good overall results and runs stably in the background 24/7, we face the issue of how to discard locations with high positioning error, that sometimes appear, without losing important information. Further, we will run extensive experiments to evaluate it and improve it, using different devices and carriers.