Elevation App

CS 6365 Project Proposal

Elevation App is a Light-Weight Android Application that leverages the micro-sensors on a smart phone viz., Accelerometer and Gyroscope to determine the gross and net change in elevation of the phone from a starting point.

Sandeep Manchem

2/7/2014

Contents

Introduction

The ubiquitous and all pervasive Smartphones have become a part and parcel of our current lifestyle. One of the key reasons for their success apart from internet connectivity and cloud support is that they are equipped with a myriad of micro-sensors that redefine "*All in one*". Two such crucial micro-sensors are the *Accelerometer* and the *Gyroscope*. So far these two sensors have been used primarily for games and motion detection. There have been attempts to use them for a wider range of applications but few have succeeded owing to the noisy data they generate. This project attempts to develop filters for the Accelerometer and Gyroscope data in order to compute the gross ascent and descent of the Smartphone from a starting point.

Motivation

Several mobile applications are available that enable a user to compute their jogging/hiking statistics viz. distance covered, average speed, maximum speed, route taken and so on. But very few apps show you the elevation achieved during the jog/hike, the gross ascent and descent, which I believe are useful pieces of information for a healthy workout session and essential to compute calories burned. The ones that do provide elevation information do it using GPS location information which does not work indoors and which is not accurate for short distances. The motivation behind building the Elevation app is to leverage the built-in *Accelerometer* and *Gyroscope* to compute the gross and net ascent and descent during a jogging/hiking session without relying on GPS.

Objectives

The purpose of the project is to successfully filter the *Accelerometer* and *Gyroscope* data from a Smartphone and use that to compute the elevation achieved by the device starting from an initial point of origin. Subsequently develop this into a Mobile Application on the Android Platform that computes the gross ascent and descent achieved by the user during an activity viz. Jogging/Hiking. The algorithm to successfully filter out the raw sensor information is expected to be heavy on resources and computation. In order to keep the mobile app light-weight, the raw data is compressed and sent over to a backend server that crunches them and

transfers the final result to the phone. The data is also stored on a database for further analysis of the sensor information and improve accuracy over time. The mobile app will also feature a rating mechanism that takes feedback from the user as to the accuracy of the app, so that the filters can be adjusted and customized for the user to improve accuracy.

Proposed Algorithm

The algorithm that is being developed for filtering the raw data of the *Accelerometer* and *Gyroscope* involves several layers of filters to weed out irrelevant readings. The algorithm will be heuristic and rely on the empirical data obtained and tested upon. An *Accelerometer* provides the acceleration of the phone along the 3 axes (X, Y and Z), whereas the *Gyroscope* will give the degree of rotation along the same 3 axes. *Gyroscope* helps us determine the orientation of the phone; where as the *Accelerometer* helps us determine its state of motion.

Since we are interested in computing the elevation achieved by the phone, we collect readings along the Y-axis, exclude the acceleration due to gravity component and apply our filters. The basic idea is that using the acceleration of the phone along the Y-axis at several points of time, we can compute its velocity along the same direction from which we can derive the distance travelled.

The Gyroscope readings will tell us the amount of rotation achieved by the phone and correlating this to the accelerometer readings, we can eliminate the force applied that resulted in rotating the phone about the axis but did not contribute to its movement along the same axis.

High magnitude of acceleration for a very short period of time does not result in any effective movement and is an indication of a jerking motion. Therefore, to begin with we will focus on acceleration of similar magnitude for extended periods of time that indicate steady motion and take it from there.

Phases of Algorithm Development

1. The first step is to achieve a successful determination of elevation achieved during a simple vertical motion i.e., lifting the phone vertically up from a table top and keeping it back down on it.

2. Once we successfully detect and measure a simple vertical motion, we extend the algorithm to detect inclined motion that is moving the phone along an inclined plane and determine the elevation achieved.
3. The next natural step is to try and determine the elevation achieved in a more day-to-day scenario like travelling uphill in a car. This scenario will help us get a reading of all forces that act on the phone while in a car and provide us a wealth of information to build filters. This scenario is not as complicated as that of walking or hiking which will be the final stage.
4. After having built the app for car drives on smooth uphill/downhill roads, we will attempt to do the same while climbing up the stairs. This is less complex compared to running and hiking as one foot is always stationary (assuming that the phone will be in the users pant pocket) and the elevation is uniform at every step and the surface is smooth unlike during a hike. Also it doesn't involve a jump as is during running.
5. The final stage is to try the app while running outside on uneven surfaces and determine the kind of forces and their magnitude acting on the phone and develop filters for them.

System Components and Architecture

The Android app makes the front-end and the back-end comprises of a *Node.js Server* which is an event driven non-blocking I/O platform, and *Mongo DB* which is a highly available lightweight document database.

Front-End

Light-Weight Android App

The Android App interface will be intuitive and will have options to select the kind of activity viz. walking, climbing, jogging, nature hike etc and simply start and stop the app. The final results will show the gross and net ascent and descent during the session along with a graph, elevation vs. time, indicating the elevation of the user at different times.

The Android App will poll the Accelerometer and Gyroscope sensors at a very high frequency and compress the raw data using the *DEFLATE* Algorithm for Android and send it across to the Node.js Server in batches using the data connection. This

will minimize not only power consumption but also the bandwidth requirements of the app.

Back-End

The Backend system will involve a Node.js server and a Mongo DB database.

Node.JS

Node.js is a server side JavaScript execution engine written on top of Google Chrome's V8 JavaScript interpreter. Node.js utilizes JavaScript as its scripting language, and achieves high throughput via non-blocking I/O and a single-threaded event loop. Node.js is used to build the Elevation Computation Engine and to host the web interface which allows users to login and verify their application usage (workout) history. Node.js also provides easy integration with MongoDB which is used in our backend.

Node.js is particularly efficient for this scenario as it is very efficient at handling multiple users. The data from the Android app is received and a copy saved in the MongoDB and the data is decompressed and the various filters are applied to determine the current elevation compared to the starting point. At the end of the session, the final results are sent to the phone and a copy saved in the Database.

MongoDB

MongoDB is a cross-platform documented-oriented database that uses JSON-like documents with dynamic schemas as opposed to traditional relational database tables– different tables and ease and speed of MongoDB, which makes it ideal for our application. It is not only easy to use but also really fast in terms of performance. We use the database for storing user information, sensor data and usage statistics. The Data stored can be used to evaluate the performance and accuracy of the system and identify patterns in its usage which will help customize the filters for the user.

Related Work

The Accelerometer and Gyroscope sensors have been used for a variety of gaming and other applications that involve motion detection. They have also been used in Pedometer apps like Runtastic. The existing applications that compute altitude change rely on GPS location information which is used to query a publicly

available NASA website to get altitude information for that location and use that to compute the elevation difference. One such example is the Altitude android app.

Planned Schedule

The following tasks will be completed by the dates mention alongside them.

***Feb 21*:** Android App Interface and Back-end Infrastructure

Mar 7: Phase 1 of Algorithm Development

Mar 14: Phase 2 of Algorithm Development

Mar 21: Phase 3 of Algorithm Development

Mar 28: Phase 4 of Algorithm Development

April 11: Phase 5 of Algorithm Development

April 18: Testing and Consolidation

April 25: Demo and Final Report

Challenges

The primary challenge of the app lies in developing the filters. It's an open ended research problem with hardly any success stories available as on date.

Successful and reliable transfer of data over wireless networks (WiFi and 3G/4G Data Connection) to the backend needs to be achieved.

Future Work

The Android App could be further developed to provide social media access wherein a user can post the workout statistics directly to social media like Facebook, Twitter and Google+. The App can be integrated with Google Maps and use GPS location information to provide a route map of the hike/jog.

A more ambitious proposal, once sufficient accuracy has been achieved with the filters, is to use this app to map out the gradient of the terrain. For example by

making every student in Georgia Tech to use this app and gathering the data collected during the day, the exact terrain gradient on all the routes used within the campus can be mapped into a nice 3D map, which can be used for various purposes like road planning.

Deliverables

1. Demo of Application
2. Project source code
3. Final Report

References

[1] http://developer.android.com/reference/android/hardware/Sensor.html

[2] http://developer.android.com/reference/java/util/zip/Deflater.html