Published in International Conference on ComNet, CIIT and ITC 2013

GeoDivert: Traffic and road condition estimation and Interpretation using smart phone sensors

[1]Prajakta Jadhav, [2]Moupali Dasgupta, [3]Casilda Mendonza, [4]Kevin Pinto and [5]Mahalaxmi Sridhar
Department of Information Technology, Mumbai India
[1]prazzzi.pj@gmail.com, [2]dasguptamoupali@gmail.com, [3]casilda.mendonza@gmail.com
[4]kevinpnt133@gmail.com and [5]mahalaxmi90sridhar@gmail.com

ABSTRACT

Estimating and monitoring road traffic in an effective and efficient way has been a challenge since many years. Some of these methods require deploying dedicated hardware such as GPS devices and accelerometers in vehicles or cameras on roadside and near traffic signals. All these methods are expensive in terms of money spent, energy consumed and human effort required. We propose GeoDivert a method which uses sensors in smart phones to manage these complexities. We started off extending the prior work done by Wolverine (IIT-Bombay) to propose use of accelerometer, magnetometer and GPS sensors to estimate different traffic and road conditions. We apply machine learning techniques on incoming streams of data from accelerometer, magnetometer and GPS sensors to identify frequent braking, which represents heavy traffic, and a lot of bump events, which represents road anomalies.

***Keywords*: Android, Orientation Sensor, Accelerometer Sensor, Magnetometer Sensor, Web Server, LED Display**

I. INTRODUCTION

With growing number of vehicle users, traffic is growing day by day. It is desirable to have a mechanism by which people can know, in real time, about the traffic condition in the routes on which they wish to travel. As a result, working on traffic monitoring has gained significant attention in recent times. We have been designing an android application which collects the sensor readings, processes the data and identifies brakes or bumps if any. We propose a system which collects information from several phones in a locality and processes that information to estimate the road and traffic condition in that locality. This information can be shown on an annotated map on the web site which can be accessed by the smart phone. The sensing part in *GeoDivert* would be similar to the Wolverine system, which uses smart phone sensors for traffic state monitoring. Further, we propose the design of an entire road traffic detection system, with the smart phones at the lowest sensing layer and including a web service to infer a big picture of the traffic state in the vicinity that can be sent back to the smart phone to be used by the user or an LED display (only proposed). This system includes sensing on the smart phone, local communication on smart phones for co-operative sensing, computation on a web service to get a holistic picture of the traffic and presentation of this information on the web site.

II. BACKGROUND

To set a scene for this paper, we begin with a brief overview of Wolverine (IIT Bombay).

Wolverine uses machine learning techniques on incoming streams of data from accelerometer, magnetometer and GPS sensors to identify frequent braking, which represents heavy traffic and a lot of bump events, which represents road anomalies. Capabilities was the conclusion come up with, because of which

the threshold based methods for event(brake/bump) detection do not work the same way on all the phones.

Our project is a further extension on the above scope wherein the tapped sensor data is brought to use by sending it to a remote server that processes the information and then broadcasts it to the subscribed smart phone users thereby benefitting them with useful information about the traffic and road conditions.

III. IDEA

The project about building a road and traffic state monitoring system using smart phones. The aim of the system is to collect road and traffic state data, use it to produce valuable information regarding road conditions and traffic jams and to deliver that information in a useful format back to the user. In order to fulfill this objective, the system architecture involves an application running on the smart phone and a web service running on the cloud. The operation of the system is as follows.

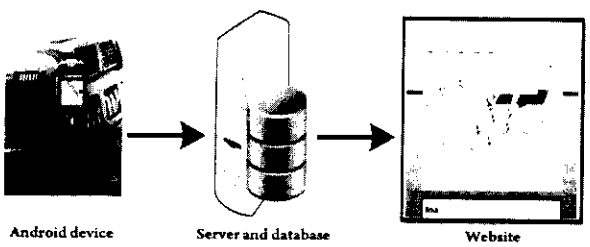

Figure 1. Overview of the system

- **Event Detection:** The idea is to collect data from the accelerometer, magnetometer and GPS and then process this to detect braking and bump events. It then attaches a time and location tag to this event data and sends it across to the web server for further processing.
- **Inter-device cooperation:** Keeping the sensors on is costly in terms of energy and if two smart phones are in the same vehicle, keeping the sensors off on one of them would be beneficial. In order to do this, the system uses inter-device wireless communication and determines whether the two phones are on the same vehicle, by observing the traces of the past events.
- **Inference:** The web server receives the event traces of several smart phones along with the time and location tags. Using this information, the web server infers higher level traffic conditions – such as the presence of a traffic jam at a certain location, a bumpy road.
- **Presentation:** The web service needs to send over the inferred events to the smart phone running the application. The smart phone sends over its location, and the web service responds with events of interests in the vicinity of this location. These events are displayed on a map on the phone, so that the user of the application can choose to take alternate routes based on this.

IV. DETAILS

4.1 About Sensors

There are basically 3 types of sensors available in smart phones – environment sensors, position sensors and motion sensors; out of which we will make use of the position and motion sensors.

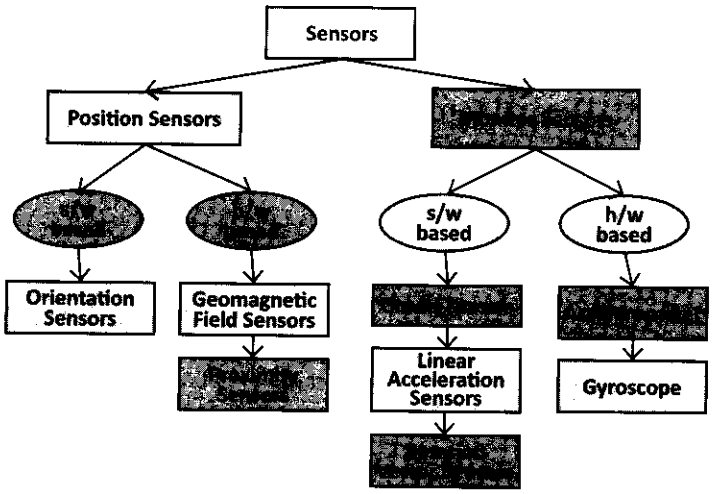

Figure 2. Classification of sensors

A. MOTION SENSORS:

Motion sensors are useful for monitoring device movement, such as tilt, shake, rotation, or swing. The movement is usually a reflection of direct user input (for example, a user steering a car in a game or a user controlling a ball in a game), but it can also be a reflection of the physical environment in which the device is sitting (for example, moving with you while you drive your car). All of the motion sensors return multi-dimensional arrays of sensor values for each SensorEvent. For example, during a single sensor event the accelerometer returns acceleration force data for the three coordinate axes and the gyroscope returns rate of rotation data for the three coordinate axes. These data values are returned in a float array along with other SensorEvent parameters.

B. POSITION SENSORS:

In the first case, you are monitoring motion relative to the device's frame of reference or your application's frame of reference; in the second case you are

monitoring motion relative to the world's frame of motion sensors by themselves are not used to monitor device position, but they can be used with other sensors, such as the geomagnetic field sensor, to determine a device's position relative to the world's frame of reference.

Position sensors are useful for determining a device's physical position in the world's frame of reference. For example, you can use the geomagnetic field sensor in combination with the accelerometer to determine a device's position relative to the magnetic North Pole. You can also use the orientation sensor (or similar sensor-based orientation methods) to determine a device's position in your application's frame of reference. Position sensors are not typically used to monitor device movement or motion, such as shake, tilt, or thrust. The geomagnetic field sensor and orientation sensor return multi-dimensional arrays of sensor values for each SensorEvent. For example, the orientation sensor provides geomagnetic field strength values for each of the three coordinate axes during a single sensor event. Likewise, the orientation sensor provides azimuth (yaw), pitch, and roll values during a single sensor event. The proximity sensor provides a single value for each sensor event. Figure 2.2 summarizes the position sensors that are supported on the Android platform.

V. MECHANISM

5.1 DETERMINING ACCELEROMETER ORIENTATION

As described earlier, the accelerometer (or rather the mobile that contains this accelerometer) can be in any orientation in the vehicle. We need to find the readings of the accelerometer along Y' and Z' axes of the vehicle to find braking and bump events respectively, which we use for estimating traffic and road conditions. When the phone is kept idle in some arbitrary orientation, the accelerometer shows some non-zero readings, giving illusion that the phone is accelerating in those directions with the specified acceleration, which is actually not accelerating. This is because the force of gravity which acts vertically downwards is factorized along the axes of the accelerometer. Similarly, even when the device is actually accelerating in some direction, the readings does not corresponding to only the actual acceleration, but also includes this added factor of gravity. We need to calculate this force acting along the axes of the phone in that orientation and subtract it from readings. This can be done once we know the orientation of the phone. To transform the vector from reference coordinate system to target coordinate system, one method is to find the angles by which the axes of the reference coordinate system need to be rotated around each of the axes, to align with the axes of the target coordinate system. Each of these rotations can be expressed in the form of a rotation matrix. For example, consider the reference coordinate system XY in 2-D plane, is rotated at an angle θ in counter-clockwise direction, then the rotation matrix equivalent to this rotation can be represented as

$$R=\begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

And a column vector V in reference coordinate system can be transformed to a new coordinate system just by multiplying it with the rotation matrix, as shown

$$V' = R(\theta) \times V$$

5.2 REORIENTATION FROM PHONE'S AXES TO GEOMETRIC AXES

The accelerometer can give *Gravity Vector*, which acts vertically downwards towards the center of the earth. The magnetometer can give the *Magnetic Vector*, which acts towards the magnetic north. Cross product of these two vectors gives us a vector perpendicular to the plane of these two vectors and acts along magnetic west (using right hand rule). We call it *East West Vector*. Again cross multiplying the *Gravity Vector* with *East West Vector*, we get *North South Vector*, which acts horizontally towards magnetic north. We do not present the exact calculation here due to lack of space. Now, we have three mutually perpendicular vectors, representing the geometric coordinate system. The rotation matrix representing this system as target system can be formed as

$$R^{-1} = \begin{bmatrix} \text{East West Vector} \\ \text{North South Vector} \\ \text{Gravity Vector} \end{bmatrix}$$

This rotation matrix is in fact a 3 × 3 matrix, which it is equivalent to. Each of the rows represents the projections of the geometric axes onto the phone's coordinate axes. The obtained rotation matrix represents the angles of rotation of the phone around the device's axes to align with geometric axes. The orientation of the phone with respect to geometric coordinate system can be calculated from the rotation

matrix with simple trigonometric calculations. Once we have the three angles, we calculate the factor of gravity that is added to the accelerometer readings along each axis and subtract it from rotation matrix. Multiplying the obtained rotation matrix with the column vector representing the modified acceleration values gives us the acceleration values of the phone along each of the axes of geometric coordinate system. i.e., we can get the acceleration values along magnetic west direction, magnetic north direction and along the gravity vector direction. Here we need to observe that, magnetic north and the true north are deviated by a small angle, hence the magnetic west and true west also. We will calculate this deviation in the next section.

5.3 REORIENTATION FROM GEOMETRIC AXES TO VEHICLE'S AXES

Once we have the acceleration values along the magnetic north and magnetic south, transforming them onto vehicle's axes is simple if we know the angle at which the direction of motion of the vehicle deviates from the magnetic north. We calculate the magnetic declination, which is the deviation of magnetic north from true north, which we can calculate from the magnetometer readings and GPS readings. We also calculate the bearing as described in, which is the angle that a line joining two latitude-longitude points makes with geometric north. We can extract the latitude/ longitude information from the location fixes returned by GPS. A GPS fix is the location identified by the GPS receiver. To reduce the possibility of miscalculation of this bearing due to the lack of accuracy in GPS fixes, we take average of bearings calculated from first fix to next 30 fixes. Once we know the bearing and the magnetic declination, we can transform the values from geometric axes to vehicle's axes with the help of a rotation matrix representing a rotation of the geometric axes at an angle bearing – declination

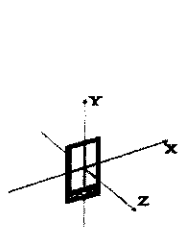

Fig. 4.2 Phone Axes

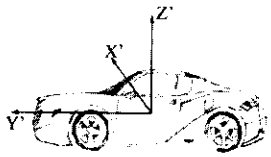

Fig. 4.3 Vehicle Axes

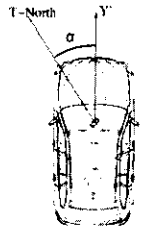

Fig. 4.4 Direction of Motion

Figure 3: Orientation of phone with vehicle axis

The data processed by the mobile can be sent to a central server, which can use the information received to annotate maps accessed by the users through this application. It can also be used to send Alerts -

- On the phone of the subscribed users
- LED displays

5.4 MECHANISM ON THE DEVICE:

The User puts the application on the handset with the gps and internet connection on. Then he has to press the "send" button on the application. This action would trigger an event where 3 values will be sent to the database on the web-server .These three values sent comprise of the following information:

1) The Z-direction acceleration value tapped from the accelerometer as described earlier gives the elevation of device w.r.t to its orientation with the vehicular axis.

2) Latitude and longitude that give the geoposition of the device i.e. the current location of the device and thereby the vehicle.

The server will plot the position of that device on a Google map on the website. This is just to show the current location of the device on the map.

The main function of the application is to determine the traffic and road conditions using these three values which can be done as follows:

5.5 LOGIC ON THE WEB SERVER:

There are thresholds set for estimating traffic and road bumps which are D and Z respectively

1) If the number of devices in a particular area [(Latitude ± x, Longitude ± y)] increases by the threshold D, then along with the position markers there will be an info Window on the map indicating suggestions to the driver such as "diversion needed, Traffic ahead".

2) Also if any device registers a z-acceleration value of more than Z, then a black dot will be marked on the map that indicates a bump.

The current position of the device with time stamp is stored in the database. So if the user is in constant motion his marker will move along with him. However the refresh rate is 5mins.The moment a new value from the same device comes up, the older values are discarded from the database. Hence the Server is a self-sufficient and stand-alone system, which may or may not require supervision.

VI. TABLES

Mode	Life Time Includes Phone Idle	Power (mW) For Given Mode Only
Phone Idle	24h 18m	182.7
Bluetooth (BT) Idle	22h 13m	17.1

BT Device Enquiry	10h 46m	229.5
BT Service Discovery	7h 53m	380.0
WiFi Idle	4h 39m	771.8
WiFi Beacon (Sending)	4h 36m	782.0
WiFi Scan (Receiving)	2h 59m	1298.8
GPS	5h 32m	617.3
Microphone	10h 54m	223.3
Accelerometer (per sec.) Accel. with Bluetooth	24h 5m 19h 56m	1.65 40

Table 1: Battery consumption of an Android Phone

Features	Bump/ Pothole Detection	Traffic Estimation	Activity Recognition	Hardware				
Apps				Accelero Meter	Magneto Meter	GPS	Microph One	GSM Antenna
Traff Sense	No	Yes	No	Yes	–	Yes	Yes	Yes
Wolverine	Yes	Yes	No	Yes	Yes	Yes	–	–
Taplogger	No	No	Yes	–	–	–	–	–
Street Bump	Yes	No	No	Yes	–	Yes	–	–
Traffline		Yes	–	–	–	–	–	Yes
GeoDirect	Yes	Yes	Yes	Yes	Yes	Yes	No	No

Table 2: Comparison table between the various existing and the proposed application

VII. RELATED WORK

i) Traffic Sense (Microsoft, April 2009)
TrafficSense, a system that performs rich sensing by piggybacking on smart phones that users carry around with them. TrafficSense leverages sensors besides GPS, accelerometer and microphone, in particular to glean rich information, e.g. the quality of the road or the noisiness of traffic. The use of an accelerometer introduces the challenge of virtually reorienting it to compensate for the arbitrary orientation of the phone that it is embedded in. Once the accelerometer is virtually reoriented, we need to design efficient and robust bump, brake and honk detectors in order to infer road and traffic conditions. The smart phone used had a Windows Mobile 5.0 operating system, unlike the rest that use an Android OS.

ii) Wolverine (IIT Bombay, January 2012)
Wolverine uses machine learning techniques on incoming streams of data from accelerometer, magnetometer and GPS sensors to identify frequent braking, which represents heavy traffic, and a lot of bump events, which represents road anomalies. The proposed system was different from Nericell system in terms of the set of sensors used and the reorientation algorithm itself. The above mentioned sensors were used in their work, which are available in many of the current generation smart phones. Different phones have sensors with different sensing capabilities was the conclusion come up with, because of which the threshold based methods for event(brake/bump) detection do not work the same way on all the phones. Hence, instead of threshold based heuristics for determining the traffic and road conditions, machine learning techniques (K-means clustering and Support Vector Machine (SVM)), which can be more precise in determining the events were used.

iii) TapLogger (IBM, April 2012)
Accelerometers on smart phones have been widely exploited by context-aware applications to infer the current contexts of smart phones, such as transportation mode inference (i.e. inferring if the user is walking or taking vehicles and activity recognition (i.e. inferring if the user is walking, sitting, or running. Differently, in their application, the goal is not in context inference but detecting the occurrences of tap events. These tap events are comparatively small fragments in the sequence of collected readings. To detect the tap events, they utilize the unique change pattern of external force on the smart phone during tap events. When TapLogger is in the logging mode, the Sensor Listener service detects tap events by observing the readings of accelerometer. Services running in background (e.g., Sensor Listener) are not allowed to receive touch event information from the touchscreen.

iv) Street Bump (Connected Bits, June 2012)
Street Bump, an Android app piloted by the City of Boston, the app allows the smart phone's accelerometer to do the job of sensing potholes. If you're driving and you hit a pothole while the application is loaded, Street Bump pairs up data about the size of the bump with a GPS coordinate and sends that to a city database.

v) Traffline (Birds Eye System, July 2012)
Using traffline motorists will be able to view traffic congestion spots along the route of their journey simply by visiting the traffic police's official website. Traffline sources data from public transport vehicles moving across city streets. The information is then overlaid on a digital map or provided in text format to users. Motorists can also receive free SMS and email alerts if they subscribe to the service. The system relies

VIII. CONCLUSION

Traffic state estimation is thus possible by tracking the location of an android device, bumps/pothole state estimation is possible by tapping sensor data and this data is put to use by sending it to a server where it is being processed to estimate traffic and road conditions. Also, this project is a scalable system, as many users having smartphones and the application installed can participate.

REFERENCES

[1] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee, "TRAFFICSENSE: Rich monitoring of road and traffic conditions using mobile smart phones". April 2008 Microsoft Research India, Banglore.

[2] Ravi Bhoraskar and Nagamanoj Vankadhara, WOLVEINE: Traffic and road condition estimation using smart phone sensors. M. Tech Report, June 2012 IIT Bombay.

[3] "TAPLOGGER". April 2012 IBM. http://www.ibm.com/developerworks/opensource/library/os-android-sensor/

[4] "STREET BUMP". June 2012 Connected Bits. http://www.dailymail.co.uk/news/article-2176783/City-releases-motion-detecting-Street-Bump-app-automatically-detects-reports-potholes-driving.html http://streetbump.org/

[5] "TRAFFLINE". July 2012 Birds Eye System. http://www.birdseyetech.com/index.html

[6] Sensor Overview. http://developer.android.com/guide/topics/sensors/sensors_overview.html

[7] Tapping into Android's sensors. http://www.ibm.com/developerworks/opensource/library/os-android-sensor/

[8] Android sensor fusion. http://www.thousand-thoughts.com/2012/03/android-sensor-fusion-tutorial/

[9] Tutorial on coding android's accelerometer. http://www.techrepublic.com/blog/app-builder/a-quick-tutorial-on-coding-androids-accelerometer/472