Use of Widely Available Android Sensors in Collision Detection

Literature Review

Gavin Wiener
University of Cape Town
wnrgav001@myuct.ac.za

ABSTRACT

Every year Android smart devices are either replaced, recycled, or become out-dated. This leaves substantial amounts of computational power and sensors unused. This paper aims to re-purpose these Android devices and their sensors with a focus on transport and safety. Past implementations and re-purposing of smart-phones were looked at to see if the any significant issues were run into or any atypical usage was implemented. Using mostly accelerometers and GPS the detection of accidents, potholes, and ultimately crowd-sourcing the events were explored multiple times with the biggest issues being establishing thresholds of when event actually occurred as opposed to regular driving usage (loose or mounted). The use of just these two sensors was found to have high accuracy once the thresholds like hard braking, the device being dropped, are established but the variation between motor vehicles made it harder to achieve consistency between varied vehicles.

CCS Concepts

•**Hardware** → *Sensor devices and platforms;* Wireless devices;

Keywords

Android; collision detection; sensors, fleet management

1. INTRODUCTION

These concerns are not Android specific, they apply to the industry, but the Android platform was chosen because they have entry-level/budget smart-phones which have gained wide use in developing countries and worldwide [13, 11]. The sensors available in Android smart-phones vary between devices but the sensors available which are accepted as standard functionality have been available since Android 2.3 (Gingerbread) which was released 2010. It is assumed any Android device released after 2010 will include these sensors because majority devices after use either of the operating systems released after Gingerbread [7].

Android operating system promotes experimentation and alternate firmwares. Re-purposed Android devices can potentially use custom firmware to improve performance and release resources for a specific re-purposing.

Section 2 will describe the most common sensors available Android devices and their typical usage. *Section 3* will introduce several fields of interest which were focused on in research papers and previous projects which re-purposed android devices for atypical usage, and specifically which sensors were used. *Section 5* will analyse these projects regarding shared issues and concepts, and potential opportunities these atypical usages may not have considered.

DOI: 10.1145/1235

2. COMMON SENSORS AND INPUTS AVAILABLE TO ANDROID DEVICES

This section will discuss the most common sensors and inputs available to Android and those which have been found to be most implemented when devices are re-purposed. Android sensors can classified into three major categories; **motion**,**position**, and **environment**. The sensors can be software-based or hardware-based; hardware-based sensors are physical components and software-based sensors will implement one or more hardware-based sensors to provide for all intents and purposes an additional sensor with which developers can provide functionality. Audio capture using the device's microphone has also found use by re-purposing projects. [8]

The most consistently implemented sensors among Android devices, and found to be to used consistently between re-purposing projects are the accelerometer, three-axes gyroscope, and proximity sensors;

2.1 Hardware-based

Accelerometer The accelerometer is a hardware-based sensor. It measures acceleration across all three-axes in m/s^2.

Audio Detection Audio detection and recording provided via a microphone (the same microphone used during voice calls).

Gyroscope Gyroscope is hardware-based and measures the rotation about three-axes in *radians per second,rad/s.*

Magnetic Field Measures geomagnetic field across three-axes in *microtesla* units, μT.

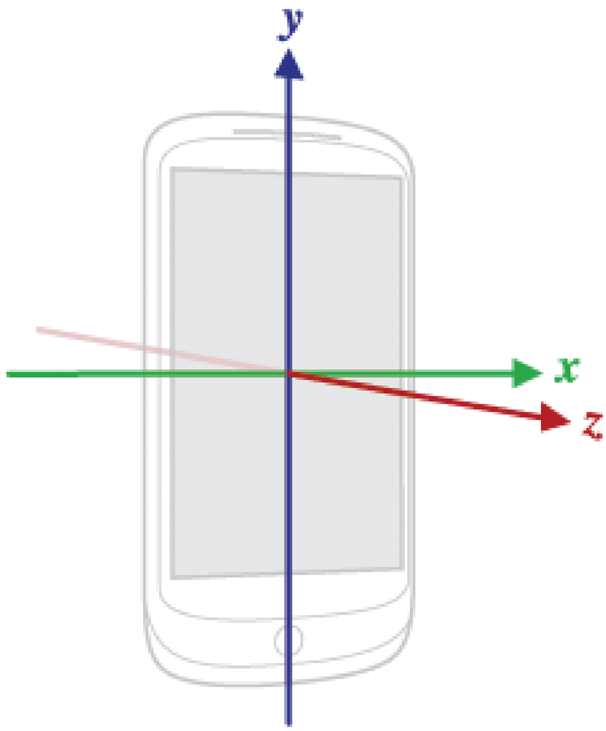

Figure 1: The visual display on three-axes available [8].

Proximity Proximity is hardware-based. It measures the distance between the front of the device's view screen and an object in front of it in *cm*. It's prominent use is deactivating the keypad and touch capabilities when the device is held next to a user's ear.

2.2 Software-based

Orientation It implements the gyroscope and accelerometer and its typical use is adjusting an application if the user uses the device in portrait or landscape, or the developers intend a user to use a specific application in a specific orientation.

3. PREVIOUS RE-PURPOSED IMPLEMENTATIONS AND PROJECTS

This section will review several projects which implement Android devices and their sensors specifically with a transport and safety focus.

Android devices have been re-purposed many times for transport. Some focused were popular and many papers had already focused on the topic but they attempted to improve the previous papers attempts or felt the previous papers had overlooked important factors which affected the implementations accuracy. Re-purposed smart devices have been found to be ideal because the non-smart-phone devices installed in vehicles are expensive, proprietary, and could be replicated with high accuracy with smart devices, thus it would provide benefits to a lot more than the relatively small number of drivers operating those vehicles as opposed to the number of smart-phone users who are also vehicle operators. Almost every transport focused re-purposed device implemented the accelerometer and Global Positioning System ("GPS") sensors, and there was often a strong focus on crowd-sourcing. This section will introducing several papers and the algorithm/s which were used.

The most popular topics were; pothole detection and more broadly monitoring road quality for early notification, accident and collision-detection

3.1 Pothole Detection and Road Quality Approximation

Pothole detection and road quality estimation would be impossible without the accelerometer (except in one case). The papers use the values along all three-axes when a driver encounters a pothole i.e. sudden drops into a pothole, or ditch and wobbling to the lack and right (bumpy roads). Detecting potholes along isn't useful though; once a driver has a encounter a pothole, they are more likely to avoid the pothole the next time they drive the same route. Therefore; crowd-sourcing is also a focus for each paper. The crowd-sourcing isn't atypical, pothole data is uploaded from each device running the application to a central server, and other drivers running the application benefit from others' data. Crowd-sourcing benefits the driver, but most papers also described the opportunity for government programmes or departments which focus on road maintenance can use the data and crowd-sourced data to preempt serious road quality deterioration. Further; if a driver is notified that a road has poor quality, the driver is likely to avoid the road and thus further prevent additional deterioration.

3.1.1 Crowd-sourcing

The most benefit drivers and government can gain is from crowd-sourcing the data received by the sensors if a pothole or accident is detected. A great issue which is further discussed later is that there can many false positives which is due to the difference between vehicle types and anomalies on the road. Therefore, collation when the data is received on the server is often used; clustering and or machine learning and then a set is build of significant road anomalies like potholes and damage[5, 6]. Machine learning may suffer from th Determining the direction of travel was even possible by *Pothole Patrol* [14] Central collection of data obviously raises the concern of privacy. *Road Condition Monitoring App* was the only paper which raised the concern of privacy, this is the same concern Google has had with Google Maps. As data is being collected along the driver's whole path, and some applications mentioned in the next section even able to recreate driver's paths, this can create safety issues for a user of application, a driver's routine, work, and home could be determined. Thus the application does not store any data regarding the driver and their vehicle, and it only saves and posts data when a pothole event is detected [6]. Most data was sent using a RESTful API [14], though data won't be sent using any identifiable data, no papers mentioned any form of encryption or secure connection use. Intermittent connectivity when WiFi or GPS was approached, a buffer of sorts was implemented, which means some of the implementations wouldn't be real-time but as most of the data needs to be collated anyway to be confirmed before it is pushed to clients, the driver would not be negatively effected unless the network loss affected multiple drivers along the same path. Crowd-sourcing further deters issues like GPS inaccuracy [15] which was found to have a 3.3 meter standard deviation [14]. If GPS was not inaccurate, acceleration could have been calculated as a derivative of speed (which can be calculated between GPS readings) [15].

3.1.2 Thresholds

Most of the papers focused on the fact that accidents will generate g-forces and sensors reactions significantly greater and different than a user would ever experience in daily usage. Establishing the thresholds was necessary. This also raised the question of mounting the device the letting the device roam freely in the car or be on the user's body [5]. The *Pothole Patrol* established that braking, door slamming, sudden swerves all generate sudden acceleration. *Pothole Patrol* was one of the few papers which made assumption that a true pothole will only effect one side of the car more often than not because drivers have a tendency to avoid anomalies in the road and that during the night is more likely to experience a full on pothole hit compared to during the day.[5] *BusNet* did not implement smart-phone devices but it used the same sensors which are available. *BusNet* lacked storage, like an older smart-phone may, therefore a threshold was set on acceleration before the device even started recording data. Additionally, a car's suspension can greatly affect the acceleration, devices need a calibration period, and suspension changes over time [16]

Real time pothole detection using Android smart-phones with accelerometers established 4 algorithms: Z-Thresh registers an event if the z-value is greater than a threshold. Z-Diff registers an event if the difference between subsequent readings is greater than threshold. This would indicate a large difference between subsequent values, and beyond the thresholds of regular driving. Stdev; events are registered if a reading is greater than the standard deviation (establishes a baseline for each vehicle). The previous algorithm would attempt to combat the issue of different vehicles having different responses to the same events and movements. G-ZERO which registers an event if there is *0g* reading across all axes, indicates free-fall when dropping into a pothole. Z-Diff was found to have the highest accuracy using known potholes. [10]

3.1.3 Models

Most of the pothole detection systems made use of data value thresholds along the three-axes and filtering to determine if a pothole event occurred. A formula model which found a linear relationship between acceleration and road quality - the worse the quality of the road the slower a driver will drive [4]. This is in contrast with other papers which identified large acceleration values along the axes.

3.2 Accident and Collision-Detection

Accident and collision detection saw more variation among the algorithms and approaches than pothole detection, though there was a still a focus on establishing thresholds (the thresholds were greater).

Most approaches relied solely on the smart device's sensors. Some high-end vehicles already include collision detection (and emergency service notification), but most importantly all vehicles after 2001 are required to have an on-board unit which records values regarding speed, engine emissions and airbag deployment, one paper implemented the on-board unit with the device's sensors [15]. The symbiosis of vehicle unit and phone does create a problem of scaling as the driver would be required to purchase the connector, and the technical aspect may deter drivers being willing to use the system (and crowd-sourcing is often a key factor to these systems).

3.2.1 iBump

iBump implemented an algorithm called *Dynamic Time Warping* ("DTW") [6]. DTW compares received time series data against templates (which are established in accident simulator described in the following few sentences). Testing the accident collision is than harder pothole detection, the researchers can't crash cars themselves, and a large enough sample size will be difficult to achieve. An accident simulator was built to test *iBump*. A holder runs along a track into a metal barrier. The paper did not consider accidents which are not head-on collisions, the metal barrier was ran into straight on, but vehicle are accidents are often not head-on. 'Head-one' accidents are often off center. This may effect the g-forces experienced and thus their thresholds, it is unlikely though. The paper discussed the limitations of a smart device accelerometers compared to devices installed by the vehicle manufacturer, but the g-forces experienced during an actual vehicular accident are a lot greater than the thresholds of regular device dropping and movements, and even the capabilities of the device sensors. The Android accelerometer was found to have a 5G thresh-hold which is less that 20G which were found to be experienced in severe accidents and death conditions experienced greater than 50Gs [15] and thus some implementations chose to actually not to use accelerometer in device because it can't register the forces high enough.

iBump's DTW and accident simulation correctly predicted accidents at 98% [1].

3.2.2 Car Crash Detection on Smartphones

The 2015 paper titled "Car crash detection on smartphones" reviewed past attempts at use of sensor data for collision-detection with Android smartphones and ultimately used location data in-addition to the accelerometer. The paper also expanded on the thresholds implemented in previous papers.The approach required establishing a threshold in g-forces for the accelerometer whereby they could state, if the threshold was met, this was not possible under regular driving conditions, including hard-braking and rapid acceleration. Dropping the smart-phone was also used to establish this threshold. The paper identified that $3g$'s is near the upper limit of what a device will experience in daily usage, they had to establish this as they did not mount the phone as a few other papers did. Falling has a distinctive pattern. As mentioned testing collisions is not feasible, the paper used test data by in a US American governmental database which freely provides velocity and accelerometer data from crash tests. [9]

WreckWatch further considered how the sensor data could be filtered from the regular usage by not only implementing thresholds on the accelerometer, but by also making use of the devices microphone and thus acoustic data. The acoustic data shared some issues with establishing the thresholds for the accelerometer. Looseness of phones, dropping the phone, environment effects (sounds of other cars, people in the car) would potentially generate false positives. Built-in services for the vehicle will obviously be better. *WreckWatch* had the same assumption with the acceleration thresholds that accidents would generate data values greater than any regular usage of the vehicle (and more extreme usage like shouting and music), specifically the sound of impact and airbag deployment. Mounting the phone allows the accelerometers to detect forces on the car, car safety oftens soften users experiences and would allow closer capabilities of the event data recorders in cars but with smart-phone sensors. The limit with smart-phone sensors was seen though when *WreckWatch* implemented the acoustic data because the microphone would clip at approximately 145dBs, which could just about be reached by the upper limits of daily usage (shouting and music). Airbag deployment was beyond this clipping threshold, which means that the application could not always differiate between the different events.

3.2.3 Reckless Driving Detection

Following many of the collision implementations; detection of accidents can be achieved with either matching templates or thresholds. Drunk driving detection was achieved with pattern matching. Various actions each had a template and the more patterns/templates matched by the received data would increase the chances that an occupant is drunk or driving recklessly. The cues; lane position maintenance problems (weaving, drifting, swerving), speed problems (accelerating or decelerating suddenly, braking erratically and stopping inappropriately), judgement problems (slow response to signals, no headlights and wide turns). Swerving and drifting affect lateral acceleration, either abnormal movements or back or forth. Spastic acceleration affects longitudinal acceleration. Instead of relying on mounting the device, calibrating to the orientation was implemented, this makes more sense considering a drunk driver or reckless driver is unlikely to go through the effort of mounting a phone. A driver actively using the application another concern. A human running can achieve approximately *2 meters per second*. Most other activities are less than *1 meters per second* and thus able to detect when the user is in a car and moving. This implementation used overlaps in sensor data to ensure they didn't miss periods, otherwise templates of the established reckless driving would not match accurately. This conflicts with other papers which were concerned regarding battery usage of polling the sensors too much. Those the devices would be re-purposed, it would not be ideal to have the device plugged in at all times [3].

3.2.4 Motorbike Safety

Previous papers made minimal or no mention of motorbikes regarding their safety. As mentioned, this may be that the movement and g-forces experience by a motorbike and its driver are different to a car. The use of gyroscope only (as opposed to a strong focus on the accelerator) was successfully implemented to automatically turn on motorbike turning signals. Steering and leaning activated the automatic signals because it created a change in the yaw value (z-value i.e. left and right) relative to the ground level. The required value was Âś 0.5 rad/s. This approach (a threshold) was different to the one used in the reckless driving paper which used templates and comparison with current data to determine turning and weaving. This radian per second threshold may be implementable across cars. The focus of the use appears to be appears to be aimed at drivers who have forgotten to indicate, the automatic signals would only initiate once a turn has been initialised [2]. It could not replace manual signalling because it would defeat the purpose of signalling in motor-vehicle i.e. preemptive indicating to other drivers your intention before the action begins. Potentially a sensor mounted on the driver's helmet may be an alternative, driver's may look in the direction they intend to turn before they initiate the turn.

3.3 Human-Robot Interaction Safety

One project called *ChibiFace* was one of the few re-purposed projects which did not focus on transport (it appears researched has a inclination towards transport because many of the sensors are already implemented in transport). The *ChibiFace* project introduced the use of Android tablets with human-robot interaction, specifically safety for the human operators and or those who many work in proximity to robotics. The project mentioned but did not discuss the use of the other sensors which were used in the previous projects. Tablets do not function for range estimation, these are hardware implemented. The researchers did not implement the proximity sensors of the device but implemented the camera and facial detection to estimate the human's distance from the machine - the size of the human face relative to it's usual size. This may have a crossover functionality with transport, a windshield facing camera may be used to estimate the distance from another car and thus if the travelling distance is safe (the driving conditions would have be considered, the server or phone could poll weather conditions from a trust service). *ChibiFace* attempted to use the microphone to determine if a human is nearby but environment noise and a single microphone was not able to localise sounds (environment noise was also an issue with collision detection) [12].

4. CRITICAL DISCUSSION

Battery life (unplugged and older devices have a worse battery lives) and being able to use the phone while other applications were still being used was a concern for several papers. Limiting the number of times the sensors are polled and which calculations were run on the local device as opposed to the server were considered, especially the polling of GPS [14]. A balance has to be created the number of calculations done locally and also the data sent. The more calculations done locally, the less data has to be sent but the more the battery is drained [6]. *WreckWatch* expanded further to determine if the user was actually in still in the car. They determined if the car was stopped at a light, driver got out of the car, but polling to see if the car had moved within a certain time period [14]

None of the mentioned papers explicitly discussed the use of motorbikes. Collecting data using motorcycle drivers may be possible with calibration, a motorbike driver's body is used to turn the bike, their pockets or backpacks are the most likely storage place for their phone (not many motorcycle drivers would mount their phone externally). The g-forces exerted (and thus thresholds) on the operators body (and on the bike) are likely to be different compared to a car. A bike operator may not have the incentive to use the application, as they will not likely to be able to hear the notifications of upcoming potholes, and avoiding potholes is likely to be easier on a motorcycle than a car.

The repeated issues seen between the projects especially using the accelerator is one of actually filtering through all the data that the sensors are producing. Some of the projects found mounting the devices was easiest, compared to establishing additional thresholds which would occur from regular use or the devices falling. Mounting the device creates overhead for the user, a consideration could be installing re-purposed devices in government service vehicles which are used regularly across varied routes e.g. waste collection and disposal, police vehicles. Using government vehicles would also reduce the inconsistency between vehicles and the need to establish additional thresholds because different vehicles have varied suspension and experience different forces [10]. A garbage truck will not experience the same vibrations and g-forces along axes as a commercial sedan. Multiple papers mentioned preemptive response and the use of the collected data for road maintenance (it is a government service).

5. CONCLUSION

Re-purposing Android devices and its sensors have found to be useful and accurate in multiple transport settings. Transport and safety appear to have an inclination towards the devices because the sensors available (though not as capable regarding thresholds) are often the same included in manufacturer installed on-board systems, specifically the accelerometer. The device not having a direct installation into the vehicle's system does mean it detects events which are not vehicle related and difficulty arises in filtering these events from each other using thresholds which would not occur under regular smart-device usage, only in unlikely events of potholes and accidents (in this situation). Thus the other sensors must be implemented beyond the accelerometer like gyroscope and orientation sensors to assist with additional filtering. The largest benefit of the captured event data arises when it is crowd-sourced; sent to a web-server, collated with others events data to filter false positives, and then pushed to other users for notification. The implementations are not perfect but the additional processes implemented provide useful systems and successful re-purposed implementations.

6. REFERENCES

[1] F. Aloul, I. Zualkernan, R. Abu-Salma, H. Al-Ali, and M. Al-Merri. ibump: Smartphone application to detect car accidents. In *Industrial Automation, Information and Communications Technology (IAICT), 2014 International Conference on*, pages 52–56, 2014. ID: 1.

[2] B.-H. Chen, S.-K. Wong, and W.-C. Chang. Motorcycle ride care using android phone. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pages 1525–1530, New York, NY, USA, 2015. ACM.

[3] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan. Mobile phone based drunk driving detection. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–8, 2010. ID: 1.

[4] V. Douangphachanh and H. Oneyama. Formulation of a simple model to estimate road surface roughness condition from android smartphone sensors. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6, 2014. ID: 1.

[5] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 29–39, New York, NY, USA, 2008. ACM.

[6] A. Ghose, P. Biswas, C. Bhaumik, M. Sharma, A. Pal, and A. Jha. Road condition monitoring and alert application: Using in-vehicle smartphone as internet-connected sensor. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 489–491, 2012. ID: 1.

[7] G. Inc. Dashboards.

[8] G. Inc. Sensors overview.

[9] J. Lahn, H. Peter, and P. Braun. Car crash detection on smartphones. In *Proceedings of the 2Nd International Workshop on Sensor-based Activity Recognition and Interaction*, WOAR '15, pages 12:1–12:4, New York, NY, USA, 2015. ACM.

[10] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo. Real time pothole detection using android smartphones with accelerometers. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pages 1–6, 2011. ID: 1.

[11] MyBroadband. Android crushing its competitors in south africaâĂŹs mobile race, May 25 2015.

[12] B. Nurimbetov, A. Saudabayev, D. Temiraliuly, A. Sakryukin, A. Serekov, and H. A. Varol. Chibiface: A sensor-rich android tablet-based interface for industrial robotics. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, pages 587–592, 2015. ID: 1.

[13] I. I. Research. Smartphone os market share, 2015 q2.

[14] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt. Wreckwatch: Automatic traffic accident detection and notification with smartphones. *Mobile Networks and Applications*, 16(3):285–303, 2011.

[15] J. Zaldivar, C. T. Calafate, J. C. Cano, and P. Manzoni. Providing accident detection in vehicular networks through obd-ii devices and android-based smartphones. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 813–819, 2011. ID: 1.

[16] K. D. Zoysa, C. Keppitiyagama, G. P. Seneviratne, and W. W. A. T. Shihan. A public transport system based sensor network for road surface condition monitoring. In *Proceedings of the 2007 Workshop on Networked Systems for Developing Regions*, NSDR '07, pages 9:1–9:6, New York, NY, USA, 2007. ACM.