



COMPUTER SCIENCE HONOURS FINAL PAPER 2016

Title: Mobile On-board Vehicle Event Recorder:
MOVER

Author: Luke Bell

Project abbreviation: MOVER

Supervisor: Tommie Meyer and Francois Mouton

Category	Min	Max	Chosen
Requirement Analysis and Design	0	20	5
Theoretical Analysis	0	25	3
Experiment Design and Execution	0	20	15
System Development and Implementation	0	15	10
Results, Findings and Conclusion	10	20	17
Aim Formulation and Background Work	10	15	10
Quality of Paper Writing and Presentation	10		10
Quality of Deliverables	10		10
Overall General Project Evaluation	0	10	0
Total marks			80

Mobile On-board Vehicle Event Recorder: MOVER

Project Paper

Luke Bell
University of Cape Town
lukekingsleybell@gmail.com

ABSTRACT

Through the rapid development of smart-phone technology in recent years, many smart-phone users have old devices, equipped with useful technologies, that are no longer in use. These devices are valuable resources that can be harnessed to improve users' lives. This project aims at doing so by helping improve road safety. An Android application – MOVER (Mobile On-board Vehicle Event Recorder) – was designed and built with the purpose of detecting car accidents through the use of acceleration thresholds. Driving data was gathered and crash simulations were run. With this data, testing and analysis were conducted in order to find as accurate a threshold as could be reached to separate normal driving from accident situations. With this application, users can leverage their mobile devices, be they old or new, to improve road safety - for themselves, and their area as a whole. A promising level of accuracy was achieved, but significant improvements can be made to the application. There is a wide gap for future work in the field, and hopefully through the development of this application, the door can be opened for other researchers to investigate and test such future work.

1. INTRODUCTION

The total number of users owning a mobile device has grown significantly in the past decade [10]. Market penetration of mobile devices has increased, and significantly so in developing countries [3]. Many Android devices, after users have upgraded to a newer smart-phones, are left at home, unused [10]. This is a waste of processing power as well as useful sensor hardware. The processing power as well as useful hardware of these mobile devices can be used every day to improve users' lives, without any additional costs to users.

This project aims at harnessing the power of these unused resources in such a way as to improve road safety. Users can leverage their devices' power while driving (in a way that does not take their focus away from driving) to help im-

prove road safety for themselves as well as their area in general. Through the use of mobile phone sensors, specifically the accelerometer, an algorithm was developed that detects collisions on the road in real-time, to an acceptable degree of accuracy.

With the real time detection of car accidents on the road, response times of emergency, traffic and police services to accident scenes can be significantly decreased. This has been shown to lead to greater road safety and better traffic management in areas [16]. Although there are numerous projects aimed at addressing the problem of accident detection in cars, most of these propose solutions that rely on and require new technology that can be expensive or inaccessible for most users. These solutions would also require a lot of time before they can be expected to be implemented in a large portion of cars on the road. By developing a mobile application to detect car accidents in real time, the technology required to solve the problem of accident detection will be accessible to most users, thereby solving the accessibility problem of other solutions that rely on newer technologies. This solution can be implemented in any car, and thus can theoretically be implemented by far more drivers in a far shorter time space than other solutions.

An Android application, MOVER (Mobile On-board Vehicle Event Recorder), was developed and tested for the purposes of developing and testing an accurate and efficient algorithm that can detect car collisions and other accidents. Through the gathering and analysis of driving data and crash simulations, appropriate accelerations thresholds were set and tested that could separate normal driving from crash situations.

The rest of this paper is structured as follows. *Section 2* discusses the various ways in which Android devices have been used to improve road safety and traffic management, with subsections that explore solutions to monitoring traffic, detecting when users are driving, detecting potholes, and finally, detecting accidents. *Section 3* describes the development of the MOVER Android application. *Section 4* describes how testing was done on the application and how data was gathered and analysed, followed by how the test data was used to develop an algorithm and threshold to detect collisions. *Section 5* goes through the results of testing the final version of the application, *Section 6* discusses these results, considers the shortfalls of the project, and indicates how they could have been mitigated. *Section 7* breaks down the conclusions made from the results, and finally, *Section 8* suggests what future work can be done to improve the application, or build on top of it.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

2. BACKGROUND

Research aimed at integrating smart-phone technology into users' cars fall into three categories, namely *adding convenience while driving*, *gathering and providing data on traffic situations* and *improving road safety*. There is also research that looks at the same areas, but focuses on improving technology within cars, or building external infrastructure, instead of using mobile phones [5, 6, 11, 14]. The approach of adding technology to cars or building external technology specifically aimed at addressing these areas will take considerably longer, and will bear considerably more cost to users, than developing mobile applications. Users can download and use a mobile app without any considerable cost in time or money. This makes the mobile application approach a far more accessible one than the approach of developing external technology. It is important to note that users should not be using their devices while driving, as this has proven to cause a lack in focus on driving and decreases overall road safety [13, 18]. Thus, applications developed for the project's purposes must run in the background, without the requirement of user input during driving. Distracting audio or visual outputs must be avoided for the same reason.

2.1 Traffic monitoring

An Android application, DriveAssist [9] was developed to provide a user interface for data collected from Vehicle-to-X (V2X) services that come built with many modern cars. This service, comprising of Vehicle-to-Vehicle and Vehicle-to-Infrastructure systems, gathers information from all cars connected through a wireless network to provide traffic knowledge to users relevant to their current area. Users are able to – in real time – view incidents on the road that are near them, so as to avoid danger, as well as find alternate routes where and when necessary for arriving at their destinations with less delay [9]. This application is an inefficient solution to the problem faced in this project because it relies on technology outside of the user's smart-phone. The V2X framework is only found in a small proportion of cars that are on the road, only introduced in 2012, and first implemented in 2013.

2.2 Driving detection

One way in which smart-phones can aid in improving road safety is to protect users from the smart-phones themselves. Through driving detection, a device can be locked, essentially preventing the driver from using their phone in a way that impairs their driving ability.

Chu et al [8] developed a Driver Detection System (DDS) which uses smart-phone sensors to detect when a user is driving a car. The system is able to, with a success rate of over 80%, recognise when a user is inside a moving vehicle, as well as detect when a user is the actual driver of the vehicle. This is done through the processing of various micro patterns that separate passengers from driver. For example, a driver will regularly move their right foot to manoeuvre the driving pedals. The intended application of the system is to prevent drivers from receiving notifications on their phones while driving, for increased road safety. Another application of the system is driving analytics for insurance companies, who would be able to track their customers driving habits and adjust premiums and such accordingly. Though the DDS system can be implemented in any car, with the only requirement being that the user has a smart-phone, the

system does not achieve the projects aim in significantly increasing users' road safety, or make the activity of driving any more convenient for users.

2.3 Pothole detection

Mednis et al. [15] explored the concept of using smart-phones to automatically detect potholes while driving. They proposed a system in which road authorities would have access to automatically generated statistical data related to damaged areas on the road, allowing the fixing of damaged areas to happen in a more efficient and organised manner. Four different detection algorithms were tested, all using the accelerometer sensor found on most Android devices. A success rate of over 90% was achieved on a test track over multiple runs, showing the effectiveness of the algorithms tested as well as the developed software. This application of Android devices in cars does not require any other technology, making it widely available to all car and smart-phone owners. But this system is a solution for road maintenance, and does not *significantly* improve users' lives while they are driving.

2.4 Accident detection

Accident detection can be vital with regard to preventing as much harm as possible to those involved in accidents. Often, the biggest problem in preventing permanent injuries is the time taken for emergency services to be notified about an accident, fetch whoever was injured and deliver them to the nearest hospital. Through accident detection systems, this time can be reduced considerably, which could result in far less permanent injuries and deaths on the road. Another application for accident detection is for insurance purposes. Being able to track where and when customers were involved in road accidents will be very useful to insurance companies.

Lahn et al [12] used Android smart-phones and their sensors to detect car crashes using a software application that makes use of a pipeline architecture to filter and combine sensor data in order to recognise crashes. Their application was very good at recognising crashes, with 100% of test data crashes recognised. But they had a problem with false negatives, where the application would detect a crash where there hadn't been one. This solution to car collision detection is very relevant to the project and was be closely studied with the intention of improving on their application and algorithm.

WreckWatch is an Android application developed in 2011 by White et al. [17] which proved to be very successful in the detection of car accidents. It was developed as a means of increasing road safety, with the driving idea behind it being that a decrease in emergency services response time to accidents and an increase in the situational awareness related to an accident would decrease road injuries and deaths [16]. The application makes use of a well-tested algorithm, with inputs from multiple sensors on a users smart-phone, which determines if a given circumstance is indeed a car accident or not. When a user's phone detects an accident, it automatically sends data related to the accident to emergency services, including geographic location and user medical information, who then are dispatched to the scene of the accident. WreckWatch also allows for bystanders of accidents to report on an incident. Witnesses to an accident can provide additional information to emergency services, or notifying them of the accident in the case where the drivers phone has

been destroyed, or the driver doesn't have the application installed on their device. Through information sent to emergency services via victims' and bystanders' devices, a higher situational awareness is given to the emergency services dispatched to the scene of the accident. This higher situational awareness allows for more efficiency in dealing with the problems associated with the accident [4]. The application runs on a client-server architecture model, with user information stored and various computations done on the server, and each user's smart-phone device acting as a client, sending requests to the server. This solution to accident detection is low cost to users, who only need a smart-phone to take full advantage of the application's features.

Zaldivar et al [19] developed a similar system to Wreck-Watch, the key difference being that instead of relying on Android sensors to detect accidents, an On Board Diagnostics II (OBD II) interface is used, a system built into the vehicle with various sensors to detect faults and accidents. The OBD interface then communicates through wireless technology with a user's smart-phone that then alerts emergency services of an accident. Although OBD technology has been required in all cars manufactured since 2001, it is not found in older cars, making this solution not applicable to as wide an audience as that which WreckWatch is applicable to, with the only requirement being that a user has a smart-phone.

3. PROTOTYPE

The MOVER application was developed as a prototype application to illustrate a concept with focus being kept on time-efficiency. This allowed a significant amount of time to be available for testing and data gathering. Basic functionality was implemented: a log-in and sign-up screen, and a main activity that displayed current accelerometer value as well as GPS position. Accompanying the values in the main activity is a Google map fragment showing current GPS position.

The application communicates with a server through HTTP requests, allowing for secure log-in and sign-up, as well as the posting of accident data. All requests are translated into SQL database queries for communication with a database located on the server. The final prototype was released as a Beta version to the Google Play store, and can be found at the following link: goo.gl/WJzG48.

Logging functionality was implemented, where acceleration data is written to a local file stored on the mobile device running the application. Acceleration values for logging are taken with a time resolution of 0.3 seconds, recording the maximum acceleration value achieved every 0.3 second window.

4. TESTING METHODS AND ALGORITHMS

4.1 Approach

Testing of the MOVER application was done through gathering drive data from car trips driven with an Android phone running the app, as well as crash simulations, performed with shopping trolleys. With these tests, normal driving acceleration patterns were recognised and categorised as well as various crash situations. Through the analysis of the test data, threshold acceleration values were reached that separated normal driving from certain collisions. All normal driving and crash simulation data can be found in a shared

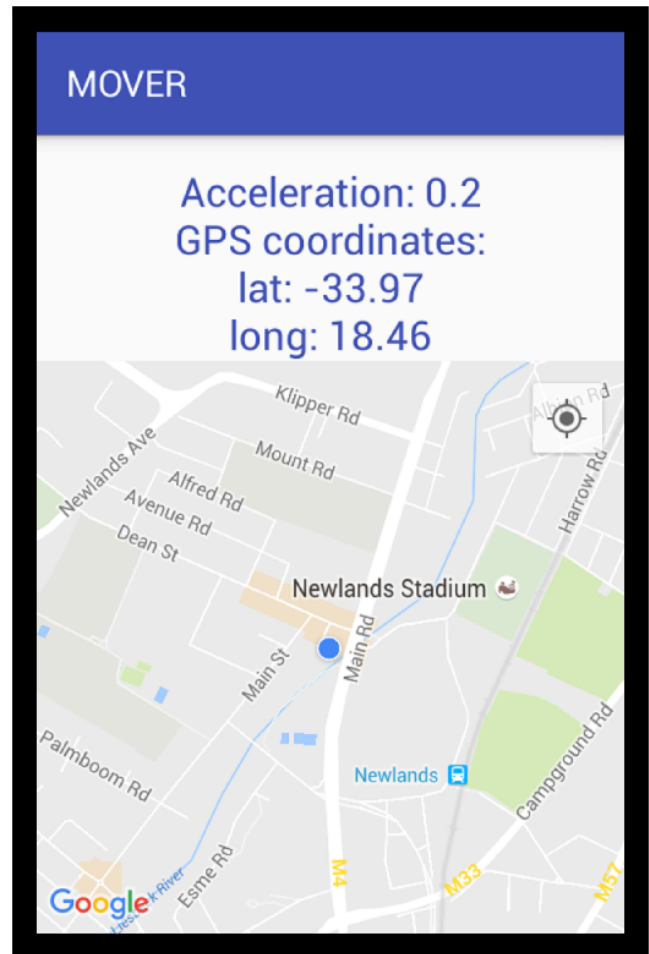


Figure 1: Screenshot of Mover application

Google Drive folder at the following link: <https://goo.gl/0lfrBX>.

While the application is running, acceleration values are logged to a local file on the device. Every 300 milliseconds, the maximum acceleration value measured for that window is written to file. These log files were used for the analysis of acceleration data gathered during testing.

4.2 Gravity and Filters

Android accelerometer data comes as a 3-dimensional vector. Using each dimension of this vector individually was not a viable option for accident detection, because the orientation of the recording device in a car could not be ensured. Thus, only the magnitude of the acceleration vector was used for testing and analysis.

Raw accelerometer data captured from android phones, is only 0 when the phone is free falling. While the phone is at rest, the accelerometer will read at approximately 9.8 m.s^{-2} , because of the force of gravity. To normalise acceleration data, gravity was accounted for, by subtracting 9.8 from every acceleration value received followed by taking the absolute value of the subtracted result, ensuring only positive values were recorded.

A low pass filter was also used to test if such a filter could allow for more efficient identification and classification of acceleration spikes. Since collisions will result in a spike in

acceleration, this would be very useful for collision detection. However, the filter proved to in fact cause these spikes to become less efficient to identify. Hence, only the raw acceleration values – with gravity accounted for – were used for analysis.

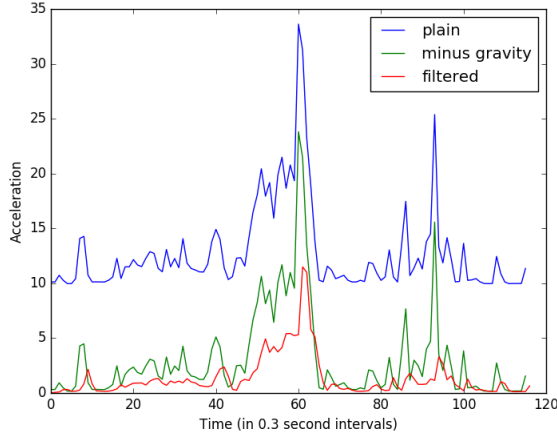


Figure 2: Example of acceleration data with raw values, values with gravity accounted for, and filtered values

4.3 Crash Simulation Environment

Real car crash data could not be used for analysis in the search for accurate threshold values to use for collision detection. Crash simulation data was gathered by crashing shopping trolleys in numerous different crash situations. It was decided that trolleys were the closest thing to cars that were available to crash. Although real car accident speeds could not be achieved with trolleys, data from trolley crashes could be extrapolated reliably to mimic the data that would be generated from a high speed crash. For the tests, a mobile device was fixed to a trolley, by means of cable ties, and the application was run on the device, recording all acceleration values, while the trolley was put through various different crash situations.

5. RESULTS

5.1 Driving Data

Normal, non-crash driving data were collected by running a number of tests where the application recorded acceleration values while driving certain distances in a car. Driving tests ranged in distance from 10km to 500km. All data were logged to a local file, and for analysis, the data were then graphed, as shown in Figure 3. This example graph visualises acceleration data from a long-distance driving trip of approximately 3 hours. The x-axis represents each 0.3 second window that acceleration data was recorded at, and the y-axis measures magnitude of acceleration (in $m.s^{-2}$).

A total of 10 tests drives were performed. This number was limited to the availability of a car. However, aggressive driving techniques were used in order to achieve higher than

normal acceleration values during drives. Aggressive driving techniques included both accelerating from stand-still as quickly as possible (Pushing the accelerator to the floor), as well as hard breaking to stop.

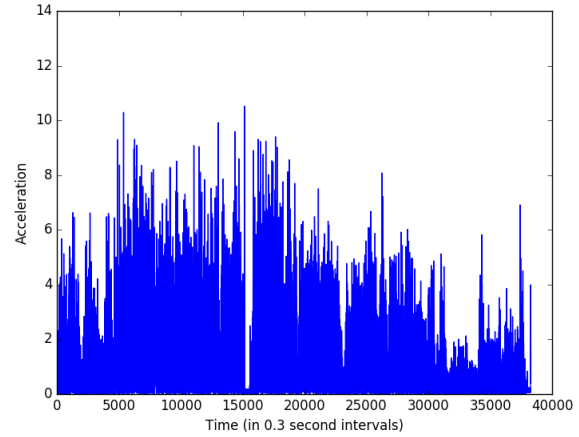


Figure 3: Example of driving test with approximately 500km covered

The maximum acceleration value recorded during the driving tests was $14.84m.s^{-2}$, which probably happened during a sharp brake action.

To consider boundary cases, where acceleration values while driving may be similar to those of a crash, extreme cases were looked at. The fastest 0-100 km/h acceleration in a car on record took 1.513 seconds[2]. The equation of motion below can be used to calculate the acceleration achieved in this record.

$$Velocity_{final} = Velocity_{initial} + Acceleration * Time$$

$$100 \text{ km/h equates to } 27.77 \text{ m.s}^{-1}.$$

$$27.77 = 0 + Acceleration * 1.513$$

$$\begin{aligned} Acceleration &= 27.77/1.513 \\ &= \mathbf{18.36 \text{ m.s}^{-2}} \end{aligned}$$

Although there is no world record for braking acceleration, the Bloodhound Super Sonic Car (SSC) was used as an extreme braking case. The Bloodhound SSC was designed to break land speed records, and can travel at speeds above 1600 km/h. At full braking force, the car's velocity decreases by approximately 66 miles per hour, per second[1]. Or 105.6 km/h per second. Using the same equation as above, this translates to an acceleration magnitude of approximately $29m.s^{-2}$.

Clearly these extreme case values are well above what values can be achieved in normal driving conditions. They serve as outlier cases that mark acceleration points below which all normal driving data falls.

5.2 Crash Simulation Data

Tests for crash situations were conducted by attaching a mobile device to a shopping trolley and acting out various crash situations. Two main crash situations were tested:

collisions with a wall, and collisions with another trolley. Two separate crash tests were performed, with a total of 13 collisions. The speed at which collisions were tested ranged from 5km/h to 15km/h (fast walking to moderate running speeds). All data was logged, just as was done for the normal driving data, to a local file, and the data was then graphed. Figure 4 shows an example of a crash test graph, containing acceleration spikes from two separate collisions.

In this example crash graph, there are clearly two acceleration spikes, which both correspond to crashes against a wall. The first spike is considerably larger than the second, and this is due to the different speeds at which collisions took place across the different crashes. The first collision was tested at running speed, approximately 15km/h, while the second crash only happened slow jogging speed, approximately 10km/h. This accounts for the different acceleration spikes shown on the graph.

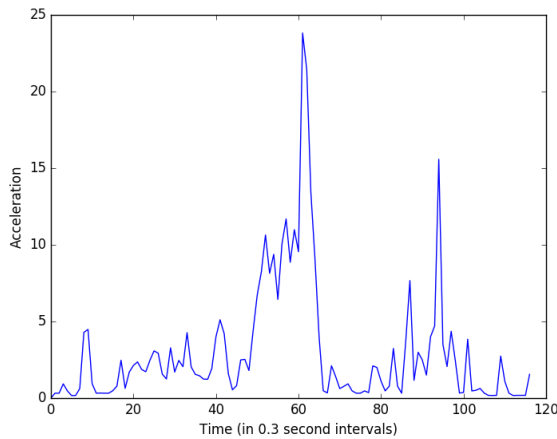


Figure 4: Example crash simulation with two collisions

Wall crashes were either head on, running the trolley straight into the wall. Or they were side on, crashing into the wall at an approximately 45 degree angle, as shown in Figure 5 below. Crashes with other trolleys were either head on with the other trolley standing still, or head on with the other trolley moving towards the crash trolley.

Different crash situations all showed different spike patterns and magnitudes. For example, direct collisions with the wall, as shown in the top right of Figure 5, produced acceleration spikes considerably higher than side on collisions, shown in the top left of Figure 5. Figure 6 is a graph representing four crashes, the first was a direct wall collision, and the next three were side on. All collisions were tested at similar speed, but the direct crash has a considerably higher acceleration spike than the side on crashes. This is because a direct crash causes the vehicle to come to a complete halt during the collision, blocking its entire path. In a side on collision, however, the wall is only blocking part of the vehicle's motion, and the vehicle will continue to move (in a slightly different direction) after the collision, therefore not feeling the same force (and acceleration change) that would occur in a direct collision. Similarly, collisions with one stationary trolley produced smaller acceleration spikes than collisions with trolleys moving towards each other.

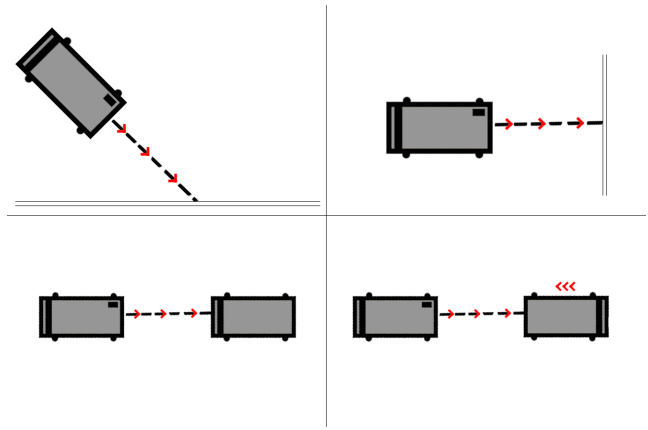


Figure 5: Sketch showing different crash test scenarios

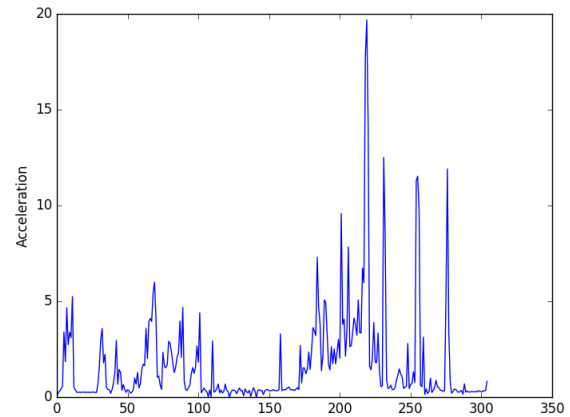


Figure 6: Crash graph showing 1 direct collision and 3 side on collisions

The maximum acceleration spike from a test crash was 23.82 m.s^{-2} , while the minimum was 10.55 m.s^{-2} . Although these values can be argued to be similar to values achieved with normal driving – especially for the minimum spike value – the tests were performed at very low speeds, and the results can be extrapolated to give an estimate of acceleration values achieved from similar crashes that occur at higher speeds.

5.3 Threshold

20m.s^{-2} was used as the threshold value to separate normal driving and collisions. This value, which is substantially higher than any acceleration value achieved through normal driving tests, was chosen conservatively. False positives – detecting collisions without there being any collision – should be completely avoided through this high threshold value. Avoidance of false positives means that no resources will be wasted or alarm raised for situations that aren't accidents on the road. While some very low speed collisions (Car travelling at less than 20km/h) may not go above the threshold, any serious collision will cause acceleration to surpass the threshold, thus resulting in successful detection through the

application.

After the final threshold was finalised, the data from the already completed driving tests were run through the application manually, confirming that no false positive collision was detected. Another aggressive driving test was carried out as well, also resulting in no false positive detections.

A final trolley crash test was also conducted. Crash test speeds were higher than in previous trolley tests – at least 12km/h – and this ensured that collisions were detected in every crash. There were no false negatives, where no collision was detected despite there being an actual collision that occurred. All collisions were detected successfully.

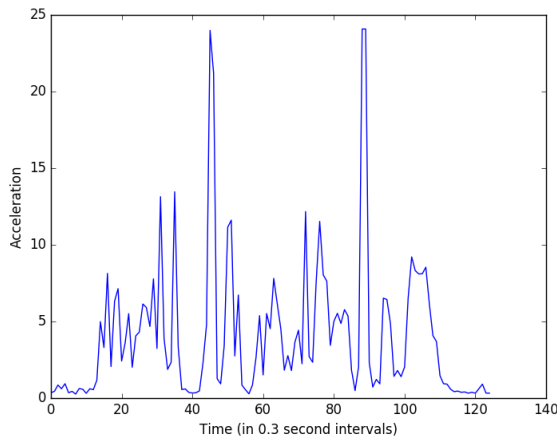


Figure 7: Final Trolley Crash Test

6. DISCUSSION

Due to the limited time and resources available for the completion of this project, compromises had to be made regarding the development and testing of the application. Far more can be implemented in terms of features within the app, allowing users to view and interact with their driving data from within the app itself for example. App design was not put into major consideration and many improvements could be made on the look and feel of the application, allowing for automatic log in, by remembering user's credentials for example.

Testing of the application was also done in a limited capacity. Far more driving data could be captured. Crowd sourcing could be used to gather data pertaining to different cars in different areas, and through the gathering of substantially more driving data, more accurate results could be achieved. Crash simulations, although useful, could also be improved by using real cars in real accidents. This could be done through crowd sourcing as well, or by using crash test operations that already crash cars on a regular basis. In addition to improving the quality of crash tests, the quantity of tests could also be increased to gather more crash data for analysis and possibly improve the accuracy of collision detection.

7. CONCLUSIONS

The final tests of the application with the threshold in place show that success was achieved in implementing a collision detection tool using android mobile devices. Normal driving is highly unlikely to trigger a false positive collision detection due to the high-valued threshold. Collisions that occur at low speeds, however, may go by undetected due to acceleration not exceeding the threshold. False negatives, where a collision occurs but goes by undetected, could only happen at speeds below 15km/h, or for very minor collisions, for example knocking a side mirror into a street light.

Although a promising level of accuracy was achieved in collision detection with this project's outcome, it can definitely be improved upon. Using only a basic threshold can be improved by means of other analytical techniques, for example processing the data around periods of spiked acceleration to confirm or deny a collision. Another possible improvement could be the inclusion of machine learning to analyse all crash and normal data [7]. Using gathered data from users of the application, an algorithm could continuously learn how to better differentiate between the patterns found in the data related to different situations. Hence being able to detect collision more accurately.

The application lends itself to solutions in traffic monitoring and management areas, and functionality could be built in to show users data regarding their surrounding area and traffic. Insurance companies may be interested in the application as they could track their clients driving habits, adjusting premiums depending on your driving history. Although there are already solutions available that insurance companies use, most of them rely on technology that can be expensive. Using users' mobile phones as sensors instead would be far cheaper and easier to implement across a company's client base. Emergency services could also use the application to help lower their response times, possibly using the application to send them alerts whenever a collision is detected. All these applications require additional features and functionality to be built on top of the application.

8. REFERENCES

- [1] The bloodhound project. <http://www.bloodhoundssc.com/forces>. Accessed: 2016-10-28.
- [2] Fastest 0-100 km/h acceleration - electric car. <http://www.guinnessworldrecords.com/world-records/fastest-0-100-kmh-acceleration-electric-car>. Accessed: 2016-10-28.
- [3] S. Asongu. The impact of mobile phone penetration on african inequality. *International Journal of Social Economics*, 42(8):706–716, 2015.
- [4] A. Blandford and B. W. Wong. Situation awareness in emergency medical dispatch. *International journal of human-computer studies*, 61(4):421–452, 2004.
- [5] L. Bouraoui, S. Petti, A. Laouiti, T. Fraichard, and M. Parent. Cybercar cooperation for safe intersections. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 456–461. IEEE, 2006.
- [6] M. Celidonio, D. Di Zenobio, E. Fionda, G. G. Panea, S. Grazzini, B. Niemann, L. Pulcini, S. Scalise, E. Sergio, and S. Titomanlio. Safetrip: a bi-directional communication system operating in s-band for road safety and incident prevention. In *Vehicular*

Technology Conference (VTC Spring), 2012 IEEE 75th, pages 1–6. IEEE, 2012.

- [7] M. Chong, A. Abraham, and M. Paprzycki. Traffic accident analysis using machine learning paradigms. *Informatica*, 29(1), 2005.
- [8] H. Chu, V. Raman, J. Shen, A. Kansal, V. Bahl, and R. R. Choudhury. I am a smartphone and i know my user is driving. In *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*, pages 1–8. IEEE, 2014.
- [9] S. Diewald, A. Möller, L. Roalter, and M. Kranz. Driveassist-a v2x-based driver assistance system for android. In *Mensch & Computer Workshopband*, pages 373–380, 2012.
- [10] C. Downie and K. Glazebrook. *Mobile phones and the consumer kids*. Citeseer, 2007.
- [11] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *Intelligent Transportation Systems, IEEE Transactions on*, 1(2):108–118, 2000.
- [12] 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*, page 12. ACM, 2015.
- [13] D. Lamble, T. Kauranen, M. Laakso, and H. Summala. Cognitive load and detection thresholds in car following situations: safety implications for using mobile (cellular) telephones while driving. *Accident Analysis & Prevention*, 31(6):617–623, 1999.
- [14] F. J. Martinez, C.-K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni. Emergency services in future intelligent transportation systems based on vehicular communication networks. *Intelligent Transportation Systems Magazine, IEEE*, 2(2):6–20, 2010.
- [15] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo. Real time pothole detection using android smartphones with accelerometers. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–6. IEEE, 2011.
- [16] C. Thompson, J. White, B. Dougherty, A. Albright, and D. C. Schmidt. Using smartphones and wireless mobile sensor networks to detect car accidents and provide situational awareness to emergency responders. In *ICST Conf., June*, 2010.
- [17] 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.
- [18] F. A. Wilson and J. P. Stimpson. Trends in fatalities from distracted driving in the united states, 1999 to 2008. *American journal of public health*, 100(11):2213–2219, 2010.
- [19] 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. IEEE, 2011.