Trinity College Dublin

Masters Thesis

Smooth Route:

Cyclist Routing With Crowd-Sensed Road Metrics

Author:
Cian McElhinney

Supervisor:
Prof. Siobhán Clarke

A thesis submitted in fulfilment of the requirements
for the degree of

M.A.I. in Computer Engineering

Submitted to the University of Dublin, Trinity College, May, 2016

DECLARATION

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university

____ ______________________

Name Date

Abstract

Smooth Route - Cyclist Routing With Crowd-Sensed Road Metrics

by

Cian McElhinney

MAI in Computer Engineering

Trinity College Dublin

Supervisor: Siobhán Clarke

The quality of a road is of great importance to cyclists, affecting their comfort and safety as well as the condition of their bicycle. Defective road segments lead to reduced safety and comfort, causing cyclists to swerve to avoid obstacles such as potholes and cracks that can cause punctures. Long stretches of uneven, deteriorated roads cause vibrations which can damage the bicycle.

Existing approaches to automate road assessment focus on using only a smartphone, these are not accurate, can be difficult to set up and they immobilise the users phone for the duration of the journey or they uses expensive embedded computers.

The aim of this project was to investigate if attaching a dedicated sensor to bicycles could accurately classify sections of roads. This approach attaches a custom 3-axis accelerometer in a 3D printed case to a bicycle that connects over Bluetooth Low Energy to the users phone where acceleration and GPS location data is stored. Data from the cyclists is uploaded to a central server where sections of roads are clustered and classified. The data is visualised and the user is can automatically find the smoothest route for their journey. The approach in this project allows for a simple set-up without heavily relying on a smartphone. 437 rides were recorded with 10 different users, this was done over a 5 month period, covering 3500 km and including 600 unique roads. This procedure was able to find anomalies to within under a meter of their true location and classify road segments into 3 distinct categories based off their average smoothness. A 93% agreement was found between ranking of roads with this system and the participants opinion.

Contents

List of Figures

List of Listings

List of Tables

Acknowledgments

I wish to show my appreciation to my supervisor Prof. Siobhán Clarke for the support, feedback and guidance received throughout the course of this project. I am very grateful for the support I received from Dr. Mauro Dragone who provided suggestions to to overcome technical obstacles encountered. I would like to thank Dr. Vivek Nallur for helping with the integration of this project and Smart-GH.

I would also like to thank all the participants that volunteered to use this system.

Chapter 1

Introduction

1.1 Background and Motivation

Knowing the quality of a road is very important to cyclists. Defects in a road can cause a cyclists velocity to suddenly change on collision or cause them to swerve in avoidance. This can cause the cyclist to collide with the pavement or alongside traffic. In 2008, 13 cyclists in Ireland were killed, 27 sustained serious injury and 308 were reported to have relatively minor bumps and scrapes [5]. Studies have shown the correlation between road deficiencies and fatal road accidents. A 2008 study of 248 fatal road accidents in Sweden found that 59% were related to poor road quality [36]. As seen in Figure 1.1, the majority of fatal road accidents were attributed to roads that were not compliant with road safety criteria.

Apart from the risk of injury, bad road sections cause discomfort to the cyclist as well as damaging their bicycles and personal possessions. Potholes and cracks result in pinch punctures, this is when the tire and inner tube are squeezed between a defect in the road and the wheel rim causing a tear in the inner tube. Long sections of bumps cause a large amount of stress on the axles increasing the chance of it failing. This ability to plan smooth routes is valuable for commuters who bring expensive equipment such as a laptop on their bikes, casual cyclists looking for a relaxing route and more serious athletes requiring smooth routes for their high-end bikes. This project aims to facilitate the cyclist in finding the smoothest and safest route. There has been research into the area of analysing roads using various modes of transport and sensors. These include Street Bump [6], which works by placing a smartphone on a cars dashboard which records accelerometer and location data from the user's smartphone, Biketastic [28] which attaches a user's smartphone to their bike in a custom designed mount and records accelerometer and location data from the phone and vCity Map [42], where a user places a smartphone in their pocket and it records both accelerometer and sound data as well as location data from the user's phone while they cycle. All of these rely on a user's smartphone as the sole input of accelerometer data, this leads to inconsistent data from different phone manufacturers with the phone being in different positions and moving freely. Also in the first two cases this immobilises the user's phone for

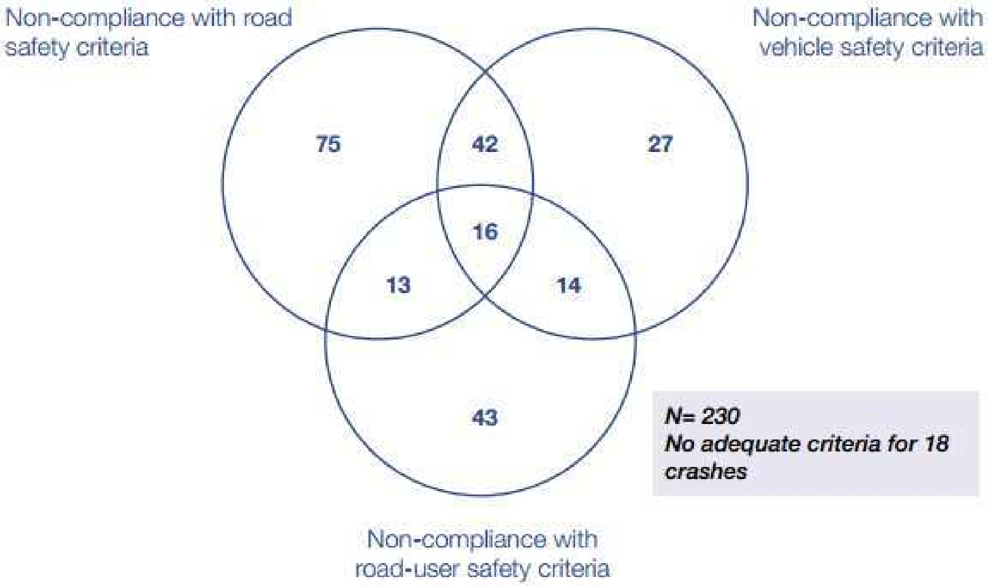


Figure 1.1: Contributing Factors in 248 Fatal Crashes in Sweden

the duration of the journey which is not practical.

Everyday large volumes of cyclists commute to and from the city. There has been a continual rising trend in the number of cyclists in Dublin increasing from 4839 in 2006 to 10349 in 2014 [8]. A large amount of these commuters are equipped with numerous sensors in the form of their smartphones, bike computers, exercise monitors, navigation devices, smart bike lights or even smart bike locks, to name a few. These sensors are not being utilised.

The motivation behind this project is to take advantage of the 20698 total commutes to and from the city center and to help plan safe and enjoyable routes. The benefits gained from an automatic system that can analyse and classify sections of road include saving money and time manually surveying roads. The metrics from manual periodic checks quickly become outdated, whereas an automatic system can continuously update it's model of all locations by taking advantage of the crowd.

1.2 Dissertation Layout

Chapter 2 will compare different research approaches and will look at specific projects which aimed to tackle the problem of automating the surveying of roads. Chapter 3 will examine the technical implementation and design of this project. Chapter 4 will evaluate the results of this project. In Chapter 5 we will discuss the future applications and the contributions of this project.

Chapter 2

Background Research

This chapter gives some insight into related state of the art research into crowd sensing road quality. I will examine the main approaches along with their advantages and disadvantages. Four similar projects will be compared and contrasted in the related work section.

2.1 Sensor Types

The three main sensing approaches considered were using smartphone sensors, dedicated sensing and a combination of the two.

2.1.1 Smartphone Sensing

Modern smartphones are equipped with a vast array of sensors and radios while being easily connected with the Internet. These sensors include 3-axis accelerometers, gyroscopes, magnetometers, pressure and temperature sensors. With radio transmitters and receivers including WiFi, GPS, Bluetooth, FM and Infrared. The popularity of smartphones mean that almost everyone has a compact sensing computer with them at all times, this allows for applications to be quickly distributed to a large volume of people with ease. For recording the quality of a road the sensors and radios of interest are the 3-axis accelerometer, the GPS receiver and the network and WiFi radios.

While the smartphone allows for a large amount of values to be recorded it does come with a number of limitations and downfalls. The quality and accelerometer frequency limitations vary between different phones. This can lead to inconsistent readings between users. The reliance on a smartphone leads to excessive battery use. With battery life being a valuable resource, users can be reluctant to use applications that draw a lot of power. To ensure accurate readings from the accelerometer the user's phone must be immobilised and fixed to the vehicle for the entire recording duration. This means the user can not use their phone to check directions, control music or interact with their phone without distorting readings. With a phone in a cyclist pocket there will be unwanted acceleration from the user's legs moving up and down as well as the persons legs acting as shock absorbers to the bumps resulting in inaccurate readings. Similarly if the phone is placed on a car dashboard there will be unwanted movement with the phone sliding along the dashboards surface which leads to readings that are not a true representation of what the vehicle is sensing.

One solution would be to attach your phone directly to the vehicle, while this would overcome some of the inaccuracies, it would require additional equipment to mount it and increases the complexity and time required to set up. In the case of bicycles most people would not be comfortable mounting their expensive phone on a bicycle, as users would see this as a risk of their phone falling out and breaking.

2.1.2 Fully Isolated Dedicated Sensing

Sensors are now becoming cheaper and more widespread. While sensors are relatively cheep, adding processing power, storage and a method to transmit data over the Internet quickly increases the cost. An approach to reduce cost would be to remove the ability to transfer data wirelessly over the Internet and instead require the data to be manually retrieved. This would require far more storage of the raw readings, increase the latency in the time it takes for the data to be uploaded and processed and would add a large amount of effort and time for the user. The requirement of an additional device removes a large amount of potential users that one could get from a method using a smartphone. After the initial disadvantages of distributing and setting up the dedicated sensor, there are a great deal of advantages. There is far more control over what type of sensor to use and how sensitive they should be. This results in higher and more reliable recording frequencies and accuracies than that achieved by smartphones.

Using a dedicated sensor removes any negative implications associated with using a smartphone. The user does not have to worry about disrupting measurements by using their phone or worry about attaching it to a mount on a bicycle. This approach will also have no impact on a phones battery life as it is a fully separate device.

Having a fully dedicated device means it can be attached in a fixed position for an indefinite amount of time. This will allow for easy distinction between the X, Y and Z axis and will provide a more true representation of what forces the vehicle is experiencing. A fixed device will require less time and effort from the user per journey as they will not need to take an action to set up the recording.

2.1.3 Dedicated Sensing and Smartphone Hybrid

Using a fully dedicated sensor or just a smartphone both have advantages and disadvantages. Many of these counteract each other, most criteria that a smartphone excels in are those that a dedicated senors fails on and vice versa. For example, a smartphone makes it very easy to transmit data across the Internet where as a separate sensor will require manual extraction of the data or expensive WiFi or mobile telecommunications technologies like 3G added to the device.

The third and final sensing method requires using both a dedicated sensor fixed to the vehicle but making use of the phone for the other actions that do not require being mounted to the vehicle such as processing, storing, recording location and transmitting it over the Internet. This is achieved by recording batches of reading on the sensing device and then transmitting the data to the smartphone over Bluetooth Low Emery (BLE).

These devices can be smaller and have a better battery life than a fully isolated sensor as they are only responsible for sensing and transmitting data over BLE to the phone. This also means that they can be much cheaper to produce. Using a dedicated sensor to do the sensing means the smartphone does not have to use as much battery life reading accelerometer measurements. Similarly to a fully isolated dedicated sensor a hybrid would share the advantages of increase accuracy, distinction of axes and ease of use while not immobilising or putting the user's phone in a potentially dangerous position.

Finally using a hybrid approach combats the problem of the reduction in potential users. This is done in two ways. This hybrid device is smaller and cheaper meaning it is easier to distribute to a large amount of users. Secondly as this device requires far less functionality than that of a fully isolated sensor, we can integrate with and make use of a wide range of smart devices that are already attached to many bicycles and utilise their sensing capabilities.

The disadvantages to using a hybrid approach are that it requires some interaction from the user and their smartphone, using some of their battery. While it is not as prohibitive as the fully isolated device it can require the purchase of additional hardware reducing the size of the potential user base.

2.2 Vehicles

Cars and bicycles are the main vehicles that are appropriate to sense the quality of a road. Both of these vehicles pose their own advantages and disadvantages.

2.2.1 Cars

Cars generally drive in the same locations on a road, along two parallel lines on either side of the lane, if there is a pothole or defect along these paths normally they do not or can not swerve to avoid them meaning meaning the probability of it being sensed is high. Even though there is going a be a high success rate for finding defects along these two paths, any other pothole or damaged sections of the road will not be sensed, leading to many affected road sections going unnoticed. A large distance of road can be covered with cars and only a few users, but it is limited to just streets and excludes areas such as separate cycle tracks and parks that do not allow the driving of cars. The speed cars travel at on different roads varies greatly with some being limited to 30 km/h with traffic limiting them to even slower speeds, while others are 120 km/h. This large difference in speed would make it difficult to compare forces experienced at different speeds. A combination of the high speed reached by cars as well as their large mass and advanced suspension means that many bumps and potholes would go unnoticed even if they were directly hit.

2.2.2 Bicycles

Using bicycles to sense a true representation of road quality has many advantages. Cyclists may offer a far more dense coverage of routes and can travel where other vehicles are not permitted to. Cyclist will travel along different paths on the same road depending on traffic, parked cars, pedestrians or debris on the road. Because of this cyclists might eventually cover a large percentage of the width of the road. Many cyclists already have smart devices on their bicycles, making it easy for them to integrate with a hybrid sensing model of using a smartphone and a fixed device. In contrast to a car, a bicycle has much smaller tyres, lower mass and less suspension giving it a more true feel of the surface below it. As discussed, cyclists will take a number of different paths along the same road. This could mean that a bad pothole is being avoided by every cyclist making it difficult to locate them. This is not a problem as if a well calibrated 3-Axis accelerometer is being used, swerves along a horizontal plane can be found to show where cyclists are actively avoiding an area. Though in reality, no pothole will be successfully avoided one hundred percent of time, sometimes rain fills a hole making it difficult to see or the cyclist has no option to swerve because of alongside traffic. This results in a pothole being eventually picked up if enough users travel by that location. If a pothole does manage to be avoided all the time without causing the cyclist to swerve to avoid it then it would be safe to say that this defect is not an issue and can be ignored.

2.3 Related Work

In this section we look at four similar projects which have attempted to automate the detection of road quality.

2.3.1 Street Bump

Street Bump [6] is an iPhone app [37] created for the city of Boston. This application works by starting a ride on the app and placing the smartphone on the cars dashboard or in a cup holder, the application records accelerometer and location data from the user's smartphone. This is then uploaded where it is processed and bad potholes are found and labelled. The use of a smartphone has meant that many users from around the world can easily download and start using it. The choice of sensor and vehicle has meant a reduction in accuracy. The aim of this project is to locate actionable obstacles which the authors describe as features such as potholes and sunk castings which are caused by nature and require attention to be taken as opposed to non-actionable obstacles such as speed bumps, railway crossings or cobblestones which need no attention. Street Bump aims to allow a city council find and fix affected areas.

The authors first extract features, this is done on a binary basis where any reading with an acceleration value greater than 0.4g and a speed greater than 5 miles per hour is extracted along with a 0.25 second window from before the incident. Next they measure features of interest such as mean, standard deviation and range of the acceleration values. The readings are also passed through a high pass filter to remove any value below 0.4g, the aim of this was to remove noise from small bumps in the road and from the phone moving around on the dashboard. Machine learning algorithms were used, Support Vector Machines (SVM), AdaBoost and Logistic regression [13], to classify actionable obstacles and non-actionable obstacles. Using a large set of training data, classifiers learnt to distinguish between these two types of obstacles. This application was reported as having "too many false positives" [38]. As well as the number of false positives there would have been a large amount of missed defects due the to low coverage from a car.

2.3.2 The Pothole Patrol

The authors of The Pothole Patrol [10] discuss how they located and classified different types of defects in the road by deploying embedded computers equipped with 3-axis acceleration sensors and GPS in cars. For their study 7 taxis were used. The choice of vehicle contributed to 2,492 distinct kilometers being covered in 10 days. The use of an embedded dedicated sensor meant that they could record at 380 Hz. To produce an accurate location for each measurement this project uses linear interpolation, this was necessary as the sensor records 380 measurements per second where as the GPS receiver is limited to 1 Hz, 1 measurement per second. A difficulty with using a fully isolated system over a smartphone is the ability to upload measurements. This project used opportunistic WiFi connections provided by participating open WiFi access points to upload the recorded data, the downside to this is that it required adding additional storage to the embedded computers to buffer the data in between connections.

When attaching the sensor to the car, the authors faced the problem of where to place it. The three positions considered were attached to the windshield, attached to the dashboard, in the glove box and attached to the embedded computer itself which was not firmly attached to the vehicle. After a series of tests driving over the same bumps, it was found that while it was easy to attach the sensor to the embedded computer, this resulted in a lot of signal noise. Consequently, they firmly attached the accelerometer to the dashboard inside the cars glove box, which is a relatively easy location to install the sensors and which keeps the sensors out of the way of passengers in the cabin.

This project aimed to classify road segments into a number of different classes including smooth roads, expansion joints, railroad crossings, potholes and manholes. They further went on to attempt to classify driving behaviours such as a hard stop and a sharp turn. It was found that it was too difficult to manually label large volumes of data for training purposes. Instead, loosely labelled training data was used. This involved providing a rough number of the features experienced on a road. For example a given reading of a road might be labelled as containing some distribution of manholes, smooth sections, a few potholes but no railroad crossings or expansion joints. The loosely labelled data was then used in conjunction with a small data set of carefully labelled data. To improve the accuracy of the system as well as removing some of the false positives clustering was used. The clustering of multiple events in the same location allowed them to remove unusual false classifications. The authors report that when classifying potholes they had a 92.4% success rate with 7.3% of them being incorrectly classified as a railroad crossing and 0.3% as an expansion joint.

2.3.3 Biketastic

Biketastic [28] attempts to create a platform that consists of mounting a smartphone in a custom attachment to a bicycle that will use an application to record acceleration and noise values as the user cycles. The user then has the opportunity to share the data recorded on that route. The motivation behind this project was to help cyclists find a "good" route to travel. This was achieved by presenting the user with visualisations of raw data. To use the application, a manual calibration step had to be taken at the beginning of each ride. This was reported to be cumbersome with users saying "...for short trips it was a hassle." [28]. A manual calibration process allows for the differentiation of axes but requires more effort by the user. The authors aimed to describe segments of a journey as smooth or rough by analysing the change in variance in 5 second windows. To combat the challenge of indicating the size of a pothole becoming difficult due to the varying speed levels, an experimental scaling parameter was found. When visualising the route, clustering was used to group readings within a distance threshold. The clustering technique used, can be described as if a point is within a certain minimum distance from the preceding points center, it is considered part of the cluster. This continues until a point is reached that exceeds the distance threshold, and if the cluster is at least four points, then all the points are replaced with the center point [46]. A study was conducted with 12 participants and a total of 208 routes were recorded over the testing time.

2.3.4 vCity Map

vCity Map uses crowd-sensing to visualise the city environment for two kinds of information: sound and road conditions [42]. This project consists of a smartphone application that records accelerometer, location and sound data. This application is started at the beginning of a journey and the user places the phone in their pocket while cycling. To analyse road conditions they must be able to separate the two input signals of acceleration, the cyclists leg moving up and down and the acceleration experienced by the bicycle as a result of the road surface. They applied independent component analysis(ICA) with the Fourier transform to the raw acceleration signals to separate them. ICA is a method to estimate sources even if source signals are unknown. Once the signals had been separate, the acceleration signal representing the road surface was analysed. The authors proposed four different road conditions: flat, positive step, negative step, and convex step. These can be seen in Figure 2.1

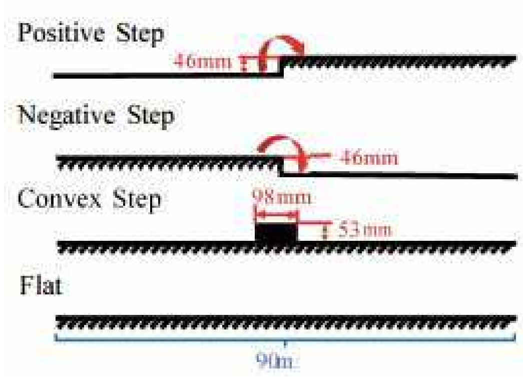


Figure 2.1: Four classes of road conditions

To test the system's accuracy users cycled over a known bump from each of the four classes in a controlled environment 50 times. These were then fed into their system where they were analysed and classified. It was found that it successfully classified the flat 100% of the time, positive steps 76%, negative steps 84% and convex step had a true positive rate of 68%.

While these success rates were quite high, in a real world environment where the user is not cycling at a consistent rate and over bumps varying in size, it would be expected that these results would be lower.

2.3.5 Comparison

These four projects aimed to achieve similar goals. To gain an insight into what attributes would be best to use in this project, Smooth Route, the previous projects will be compared and contrasted. The most favourable attributes of each project will then be used for the development of this project.

Target Use Case and Motivation

The Pothole Patrol and Street Bump share the same goal, to allow city workers to locate potholes that need fixing. Both of these projects make an effort to classify recorded bumps to let city works know what bumps are intended to be there, like speed bumps and expansion joints and which require attention to be fixed, like potholes. Biketastic makes it clear their goal is to allow users to find nice and enjoyable routes to take. This is evident from letting users view and compare other users rides to find a good route for them to take based on noise and smoothness. This is also seen by how they describe themselves as almost a social network allowing users to annotate routes or upload pictures. While it can be seen that the motivation of Biketastic is to allow cyclists to find a good route, the authors make no attempt to automatically find that route for the user but instead they present the user with visualisations of the data. vCity did not make it clear what their aim of their project was. It appears their motivation was to see if it was possible to use just a smartphone in a cyclists pocket. While these projects seem to target different use cases for the data recorded, all of these applications could be used interchangeably to a degree.

Sensor

The Pothole Patrol was the only project to use a fully isolated sensing device, this project was reported to have very accurate results, with both a high rate of accuracy locating features as well as classifying them into many different groups. In comparison with Street Bump which had a lower success rate with fewer types of features being classified, it is evident that the type of sensor as well as it being firmly attached played a big role in this gain in accuracy.

The location of the sensor and if it is fixed or not is evidently important, looking at Street Bump and The Pothole Patrol the main difference is that the sensor is fixed to the car in The Pothole Patrol. I believe this was the main attribute to the higher accuracy and less noise experienced by The Pothole Patrol. On inspecting Figure 2.2 and Figure 2.2 it can be seen that the two main differences are that The Pothole Patrol records at a far higher frequency. This can be attributed to the sensor type allowing a sampling rate of 380Hz as opposed to 40-50Hz which was achieved by Street Bump. The other main difference between these two graphs is the level of noise. With The Pothole Patrol it is clear where the bump is and where the smooth areas are. With Street Bump it is a far noisier graph, this difference can be linked with the placement of the device with the smartphone being free on the dashboard where as the sensor is fixed to the dashboard in The Pothole Patrol.

When comparing the two bicycle projects it can be seen that Biketastic recorded cleaner data and was more robust, as vCity was only seen to work well under lab conditions. Biketastic was reported to be a lot of effort from the users side where as vCity only required putting your phone in your pocket.

A final consideration to take when picking the sensor type is to consider cost. Using a pure smartphone approach requires no additional purchase by the user. Having a custom mount, like with Biketastic, will require a small or even no expense with some users already owning mounting enclosures for their phone. This is an important decision to make as to whether it is it worth adding more hardware to have the sensing device fixed to the vehicle if it removes a large percent of potential users.

Vehicle

The Pothole Patrol and Street Bump are limited in the coverage of the width of the road and the locations where cars are permitted to drive. Both of these projects could only record measurements of features in the road that either spanned the width of the road, such as speed bumps and railway crossings or features that appeared directly in line with the path

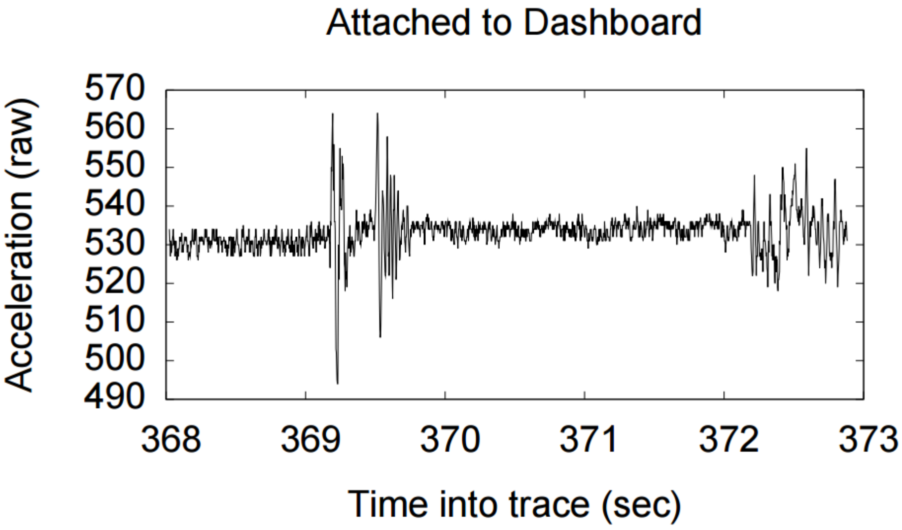


Figure 2.2: The Pothole Patrol: Raw accelerometer data [10]

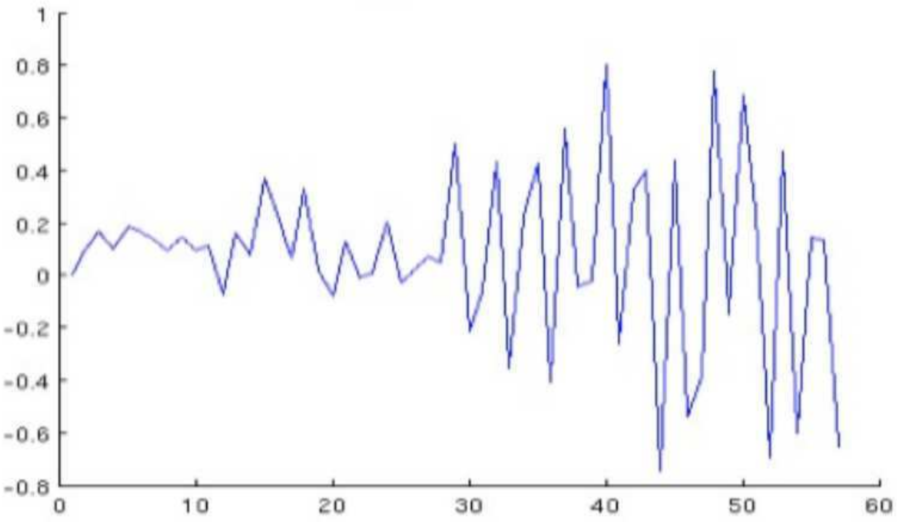


Figure 2.3: Street Bump: Raw accelerometer data [6]

taken by the cars tyres. It is evident from the two car based systems that their sensitivity is very low this can be seen in photographs included of typical potholes as well as by looking at their raw data. As seen in Figure 2.4, The Pothole Patrol describe this as a "Typical small pothole". For a cyclist this pothole is far from typical or small. If a cyclist were to hit that pothole at speed it would result in a high probability of a crash or damage to the bike.

Looking at the raw accelerometer data from both The Pothole Patrol and Street bump

Figure 10: Typical small pothole with missing pavement.

Figure 2.4: The Pothole Patrol: Typical small pothole [10]

it can be seen that the magnitude of the acceleration experienced over a pothole is approximately 1m/s^2 and 5m/s^2 respectively. When compared with readings 10-15m/s^2 for vCity it can be seen that the mass and suspension of a car play a huge part in dampening the signal received by the sensors. When measuring fine details about a road, such as if it is a freshly paved road or an old gravel road it is very difficult to tell the difference between the two in a car but on a bicycle those two road surfaces are massively different.

Ease of Use

Ease of use and practicality are important aspects to consider when developing a system to be used by the public. If the application obstructs the user by too much or requires a lot of their time, many will simply not use it. With the three projects that rely on the smartphone as the sole source of sensor data users will be reluctant to use it as it will require they do not touch their in anyway for the duration of the journey. With Biketastic this level of impracticality and difficulty of use is particularly bad not only is the user's phone immobilised for the duration of the ride but the user must attach it in a specific way to a mount on the bike and then perform a manual calibration. The unwillingness of a user to attach their expensive smartphone to a mount on a bike further reduces the practicality of this approach.

Axes Distinguished

The Pothole Patrol appears to distinguish between three axes of acceleration. This separation of the three signals allows them to perform detailed analysis such as classifying sharp turns and sudden breaking which would cause a large spike in acceleration along the X-axis and Y-axis respectively. As the device is fixed in the same location for each journey it is easy to get the three axes separated. Similarly for Biketastic the three axes can be distinguish after a manual calibration is carried out, but the authors seem to ignore the the X and Y-axis and just focus on the Z-axis. Street Bump attempts to isolate the three axes by defining Z as the direction of gravity with the X-axis pointing North with the X-axis aligning with the direction of travel, this solution poses a few problems outside lab conditions, for example, if the phone is with in a cup holder or free on the dashboard allowing it freely rotate throughout the journey. In vCity two axes are considered, the X-axis which aligns with the direction of motion and the Y-axis which points in the direction of gravity. The X-axis is used to help separate the motion of the user's leg from that of the signals of the road, this results in this project isolating one axis of acceleration. In a similar way to Street Bump, vCity is vulnerable to the phone moving and rotating in the user's pocket outside of lab conditions. From these projects the importance of distinguishing between the axes of acceleration can be seen. Having well defined axes allows for more detailed classification and helps to clean signals, removing unwanted noise.

Speed Adjustment

When a vehicle hits a obstacle in the road the acceleration experienced is going to be affected by the speed of the vehicle, this is the premise of speed bumps. To obtain more accurate reading the speed at the time of impact should be considered. The authors of Street Bump make no direct mention to the use of speed to adjust acceleration but they do explain how speed is used in the training and classification phase of their system, this leads the reader to believe that speed is accounted for when analysing measurements. The Pothole Patrol directly discusses how speed impacts the magnitude of the acceleration sensed. The authors further explain how this is particularly useful for highways to prevent false positives where the car is travelling at particularly high speeds. Biketastic uses an experimentally obtained scaling parameter to scale the acceleration of each reading. vCity makes no mention of taking speed into consideration as their experiments were strictly performed at 13 km/h.

2.4 Research Approach

Before it can be decided what favourable attributes are wanted, we first need to define the motivation behind this project. The goal of this project is to develop an easy to use and easily distributed system that can locate areas that are good and bad for cycling. These areas must include dedicated cycle tracks and along side traffic cycle lanes. It must also include the full span of the width of roads. It should have a high level of accuracy that can distinguish between a freshly paved smooth road and a slightly older more weathered road.

To ensure this system is easy to use while still being relatively easily distributed, a hybrid solution was picked for the sensor. A bicycle was chosen as the vehicle. It is the only choice that fulfils the requirements of coverage as well as the requirement of distinguishing between reasonably good and very good roads. An automatic method of calibration would be favourable as it saves time and effort for the user. This will make it possible to isolate all three axes of acceleration. Speed adjustment will be used to account for the variation in speed by different cyclists at different points so as not to penalise fast sections of road. Once an adequate amount of data has been recorded it will be displayed to the user in a similar way as done by Biketastic but with the added feature of automatically finding the best route for the user.

2.5 Ethics

When dealing with members of the public ethical consideration must be taken. Ethical approval was granted before any participants were included in this project, this required adhering to the ethical guidelines presented by the Ethics Committee. Participants were free to withdraw from the study for any reason without any penalties at any time. The participants were told that the data gathered is done so anonymously and will be treated with full confidentiality.

These considerations were taken into account when developing the system and when gathering participants. In particular privacy and anonymity were given careful thought when developing the Android application by the inclusion of privacy zones and the aggregation of data.

Chapter 3

Implementation

3.1 Overview

Developing on the knowledge received from the projects studied in the research section, the architecture of this project was designed. A high level overview of the chosen design can be seen in Figure 3.1.

A Bluetooth enabled accelerometer is attached to the user's bicycle, which will communicate over Bluetooth Low Energy to an application running on an Android smartphone. The Android application will record location data and store it along with the acceleration data, these two data sources are uploaded to a Heroku web service. This web service is responsible for storing the raw data on Amazon S3 [2]. An entry for the ride is pushed to a Redis [29] queue with the metadata denoting where in S3 this ride is stored. A daemon process removes entries from the Redis queue, retrieves the uploaded files from S3 and processes them. An infrequent scheduled Cron process performs more processing and adjusts the data. The resulting data is stored in a MongoDB [41] database. This data is then fed into custom mapping user interfaces and into Smart-GH [21] where automatic routing is achieved.

3.2 Hardware

For the hardware a MetaWear R Series [18] was chosen. The MetaWear is composed of a Nordic Semiconductor nRF51822 BLE SoC, this provides a 2.4 GHz transceiver that can operate over Bluetooth Low Energy, a ARM Cortex-M0 32 bit processor, 256 KB of flash program memory and 16 kB or RAM. The MetaWear has an integrated 3-axis accelerometer that can sense ±8g as well as an ultra-bright RGB LED. It has a max data rate of 800Hz which is far higher than any smartphone sensor. A micro-USB can be used to provide power and charge the battery.

Initially two early access beta version of the MetaWear were used to create prototypes. These early versions were not assembled. A lithium polymer battery had to be soldered to the board. Flux was applied to the end of the leads to ensure the joining of the metal on

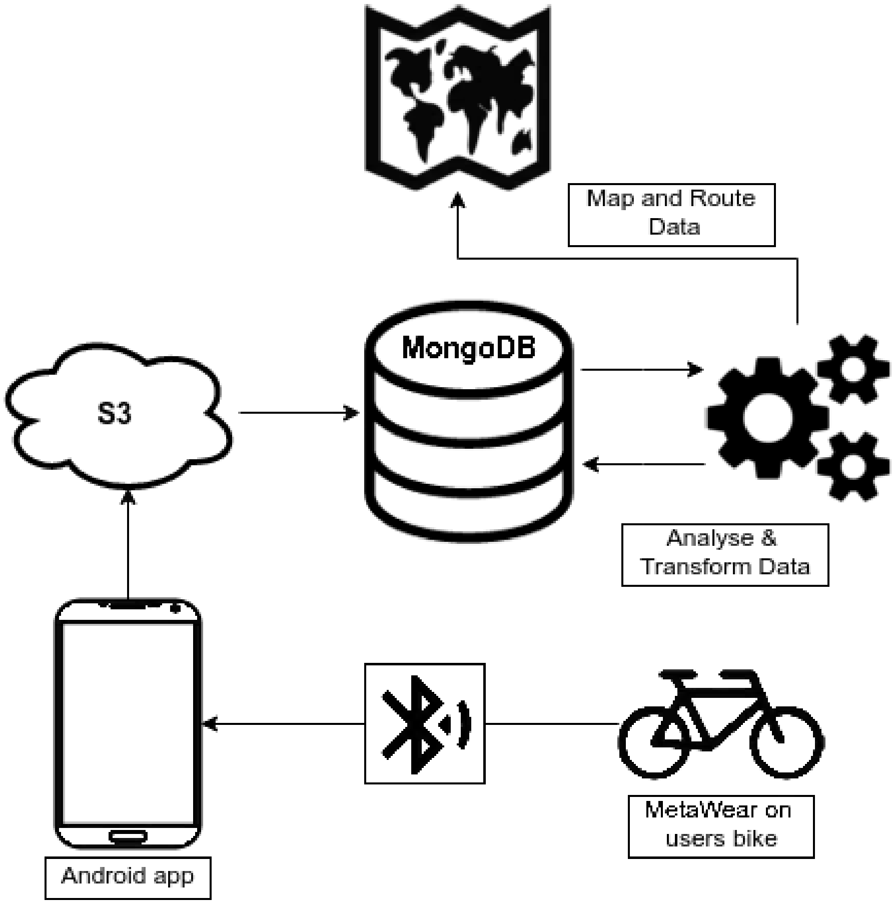


Figure 3.1: System Diagram

the board. As seen in Figure 3.2 the ground wire of the battery was carefully soldered to 3 V_{GND} and the positive lead was soldered to 2 V_{BAT}.

Once the hardware had been proven to work 10 more MetaWears were ordered. These were fully assembled and were smaller than the prototypes that were initially used for development.

These devices were assigned unique names, BIKE0 through to BIKE9. This was with the aim to make it easier to keep track of how many were currently distributed and who had which device.

3.3 Enclosure

For the sensor to record an accurate reading for a road surface it was clear from the previous research that the device must be securely fixed to the bicycle. A number of designs and variations of those designs were looked at.

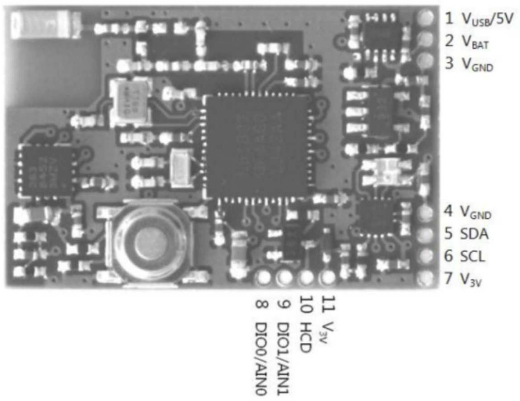


Figure 3.2: MetaWear R

3.3.1 3D Modelling

The final design was inspired by a commonly used bicycle light. The design consists of a hard plastic internal case which holds the MetaWear, with a soft rubbery plastic used to attach to the bicycle as well as providing friction to stop it from twisting around the pole it is connected to. The inner hard plastic case was modelled with a combination of SketchUp [1] and OpenSCAD [25].

The inner hard plastic case can be broken into three main aspects, the top half of the case, the bottom half of the case and the hook protruding from the bottom half for the elastic to attach on to.

SketchUp is a GUI based 3D modelling program. This was used for the designing the top and bottom half of the body of the case. Small dimples were used on the lip of the case so the two parts could snap together. A hole on the back was created to align with the USB charging port of the MetaWear.

OpenSCAD is a 3D compiler that uses a programming language similar to JavaScript. This was mainly used for designing the hook. OpenSCAD was choose for this as it is effective at turning a mathematical formula for a curve into a 3D model. OpenSCAD also offers the ability to easily parametrise modules. This functionality was particularly useful for designing the hook as we iterated through a number of different lengths, widths and curvatures to create the strongest model that fit best to a bicycle. Simply changing the value of one variable meant that a full model could be changed and rendered to a 3D CAD file with ease. Listing 1 shows the use of modules and variables to construct a 3D model.

When rendered this example produces a 3D model seen in Figure 3.3. In this example a hoop is created by drawing a circle and then subtracting a smaller circle to produce a hoop with a thickness defined by the parameter passed in to the hoop module, this is represented in semi-transparent red. A curve created by taking the intersection of a hoop with a square to cut out a section of the hoop. Finally this curve is linearly extruded to a height of 5cm.

The body of the model designed with SketchUp was then combined with the finalised

```
module hoop(thickness, radius) {
    difference() {
        // draw circle
        circle(rad);
        // remove center
        circle(rad-thickness);
    }
}
module curve(size, thickness, rad) {
    intersection(){
        // make a hoop
        hoop(thickness, rad);
        // only keep part of hoop that intersects with this square
        square([size, size], center=true);
    }
}
// create curve of height=5cm, width=15cm, thickness=0.8cm, radius=20cm
linear_extrude(height = 5)
    curve(15, 0.8, 20);
```

Listing 1: Sample OpenSCAD: Code for hook

hook within OpenSCAD, this meant that the body of the case could easily be scaled and transformed from within OpenSCAD based off provided parameters. Netfabbs 3D Model Repair tools [20] were used to ensure connectivity and that there were no holes throughout the model. The full finalised case design can be seen in Figure 3.4.

3.3.2 3D Printing

To obtain a solid form of the 3D model described in the previous section, an Ultimaker Original [44] 3D printer was used. This 3D printer uses a Plexiglas build plate and has the ability to print using Polylactic acid (PLA) and Acrylonitrile butadiene styrene (ABS). For the choice of material ABS was initially considered, ABS has a higher melting point and is a stronger plastic with a more flexibility when compared to PLA. While still strong PLA is more rigid and offers a wider range of colours and levels of translucency. ABS is known to be more difficult to print with as it can warp and curl making it difficult to print precise models, this can be combated using a heated build plate.

It was decided that PLA would be the better material to use for this project as the 3D printer being used did not have a heated build plate it would be difficult to print cases with high levels of accuracy. This was important as the two sections of the case were modelled to perfectly fit together and even a millimetre difference due to warping would cause the parts not to fit together. The other deciding factor for the use of PLA was its ability to

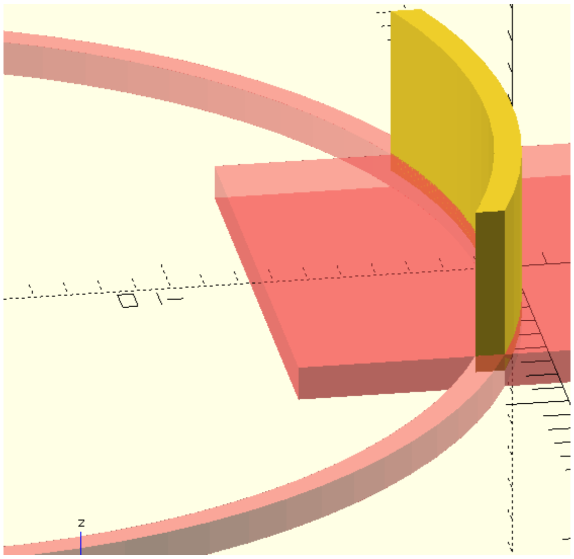


Figure 3.3: Sample OpenSCAD: Model of hook

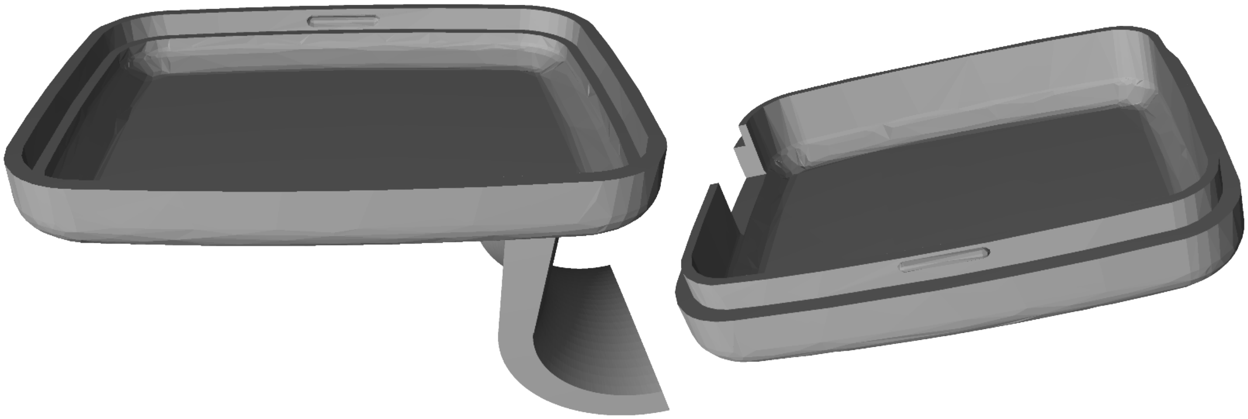


Figure 3.4: Case Top and Bottom

be translucent, this was important as we would like the ability to take advantage of the ultra-bright RGB LED built into the MetaWear.

The method of 3D printing used by the Ultimaker Original is additive manufacturing. This entails slicing the 3D model into layers, these two dimensional layers are then broken into one dimensional lines. The instructions on where these lines should go are serialised into a G-Code [45] format and written to an SD card.

Before the print can begin, it must be ensured that the bed is level and at the correct height. This process is known as calibration and involves positioning the print head in each of the four corners of the build plate and adjusting that corners height until it just about touches the print head.

Using the parametrised model created with OpenSCAD, over 20 different models were printed. Various dimensions of the case were tested to find the ideal internal volume for MetaWear to tightly fit into. The thickness of the walls were experimented with to find the optimum size to provide protection and rigidity while still permitting the light from the LED to pass through. A curve around the Y-axis provides a hook for the elastic to clip around. Testing on different configurations of the model were performed on the hook. A curved or arched shape around the Z-axis was found to result in the strongest shape. The arch displays greater resistance to fracture while in tension, compared to a flat shape. The degree to which the arch is curved was chosen based on iterative designs to find the best resistance to the tension forces experienced from the elastic enclosure attaching it to the bicycle. Ultimately the final design was chosen based on the merits of its resistance to both tensile and compressive stresses. Figure 3.5 shows a sample of the different configurations and materials tested.

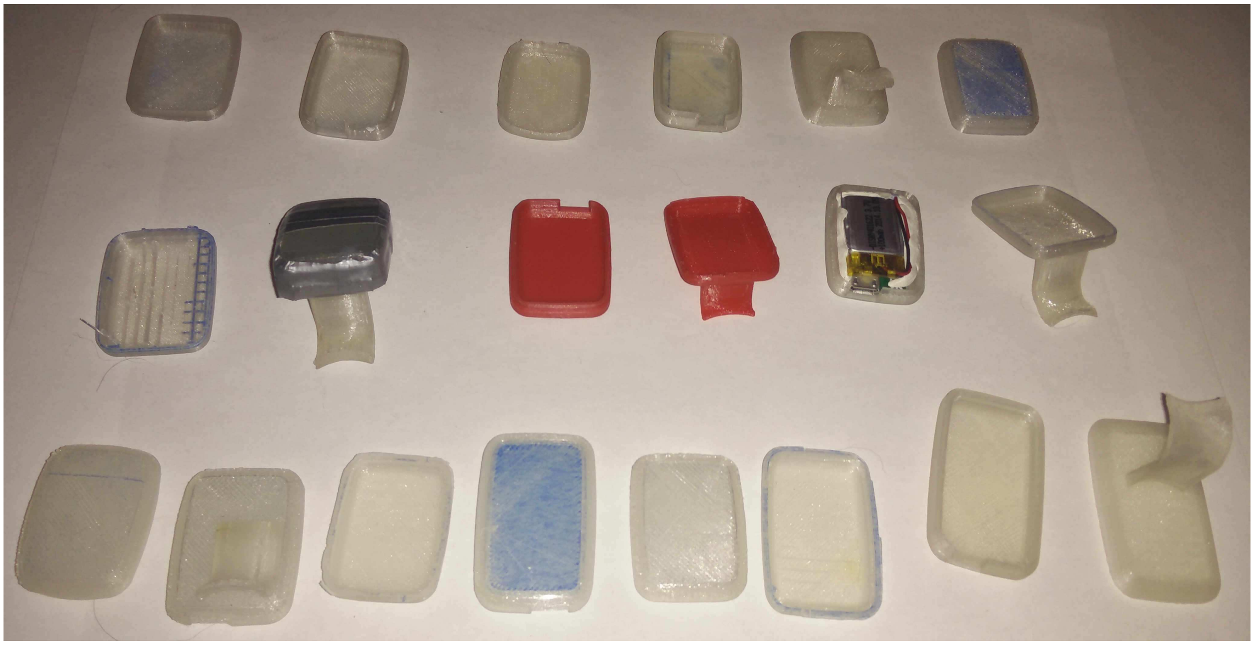


Figure 3.5: A Collection of Prints Testing Parameters and Materials

For the outer elastic casing, low cost rubber bike light attachments were used. These were obtained for free as they are normally disposed when the lights costing one euro break or run out of battery. The final device can be seen in Figure C.1.

3.3.3 Placement of Device

Three locations on the bike were considered for the device to go. These were the handlebars, seat post and the handlebar stem.

The first location looked at was the handlebars, these seemed like the obvious choice but it became apparent that this was not an option for every user. Some bicycle, especially road bikes, have very little room on their handlebars. The little room there is, is usually already in use with lights, bells or other accessories. This can be seen in Figure C.1 where the limited space on the handlebars in already in use with a bell and a light.

We contemplated placing the device on the seat post, this is also a popular location for bike lights to be attached and tends to have free space. A concern with this placement was that it may be more difficult to calibrate the orientation of the device. This transpired to not be an issue providing the seat post was not fully perpendicular to the road. This is known as the seat tube angle and ranges from 70 to 80 degrees meaning the calibration system would work for this placement. If this location was chosen then the functionality of mirroring notifications would be lost. On experimenting with this placement it was observed while it was accurately calibrated due to the generic calibration process that the device tends to rotate and get nudged by the user's leg as they cycle.

It was found that the handlebar stem of the bicycle was the perfect location. Here the user can easily see notifications, while not taking up any space on their handlebars. The stem was the chosen location recommended to users to attach the device.

3.4 Android Application

The core functionality of the Android app is to communicate with the MetaWear device attached with to the bicycle, to store the sensor data received and upload this data. A secondary functionality was added to the application to have the LED on the MetaWear mirror the LED notification light found on most Android phones. SMS and call notifications were picked to be mirrored as they are seen as having the highest priority. A bright purple light flashes for the duration of the time the phone is ringing and an orange light flashes 4 times for any received SMS message.

3.4.1 Privacy

The ethical implications of this project were considered paying particular attention to the main areas discussed in The Data Protection Acts 1988 and 2003 [7] that govern the processing of personal data. It was ensured that the user is in full control of the data which meant no data was recorded or uploaded without the user's consent. Location and sensor data is only recorded after the user pressed start and this recording is stopped as soon as the user touched stop. It was important that any personal data or data that could be linked to an individual was not recorded. It would be possible to look at the produced heat map and see where a user resides or commutes to, for example, viewing a driveway that has a lot of points recorded on it, or noticing that a common routes start and end point. These privacy concerns were tackled with two approaches. The first precaution taken was the use of a privacy zone, this feature allowed the user to select a distance that would be omitted from the start and end of every journey. The default value for this was 60 meters meaning that the first and last 60 meters of the journey are never stored. The user was in full control of this value and could change it to any value of their choosing.

Secondly all data recorded is anonymously aggregated making it impossible to discern the difference between two overlapping journeys by one or more users. With these two precautions it would mean that if the user went the opposite direction, for example left as opposed to right, to or from their home or workplace entrance or if they, or any other user, cycled past their home or workplace entrance even once then it would be impossible to recognise any starting or finishing points for routes.

An approach of automatically uploading data on behalf of the user was not taken as it was felt that the most ethical approach was to let the user choose when and what data they wish to upload.

3.4.2 Design

It was important to keep this app simple and attractive to ensure continued use. Material Design as defined by Google [17] was implemented, this was with the aim to provide a familiar looking application to the user, one which they intuitively know how to use.

The application consists of three views or activities, the settings menu, main screen, and previous rides.

CSV files are used to store logged sensor and GPS data. A SQLite database is used to store metadata about a ride, this contains the start and end location, the time and duration, what suspension type was used for this journey, the MAC address of the device used for that ride and if it has been uploaded yet. Shared Preferences are used to store key-value pairs for data relating to the user and the use of the application.

Settings Menu

When the user first starts the app they are prompted to fill out their chosen settings as seen in Figure B.1. Clicking on the Privacy Zone option launches a number picker dialogue allowing the user chose the size of their privacy zone. The LED notifications for calls and SMS can be toggled on and off separately. Finally the user can pick their bike type from a range of 1-5 where 1 represents a racer and 5 a full suspension mountain bike, this can be viewed in Figure B.2. These settings were stored in using Androids Shared Preferences key-value store [30]. This consists of an XML document recording key-value pairs that are stored internally on the user's phone meaning that it can only be accessed by this application and when the user uninstalls the app all files stored for the app will be removed.

Main Screen

For all subsequent launches of the app after the first time the user is shown the main screen. When a user presses the start button for the first time, or if they have not yet connected to a device, they will be prompted to select a device from scanned devices, seen in Figure B.3. The chosen devices name and MAC address is then stored in the Shared Preferences. All later launches will use this device unless changed by pressing "SCAN NEW DEVICES" which will launch the Bluetooth Device Picker again. In the lower left hand corner of the main screen, the device name and the first two groups of its MAC address can be seen. The colour of this text varies between red, yellow and green where red represents not connected, yellow is for trying to connect and green signifies that it is currently successfully connected. When the device is connected its battery status is queried and displayed beside the device. GPS status is displayed in the upper left hand corner. This consists of the user's speed in km/h and the accuracy of the GPS measurement. The center of the screen displays what the application is currently doing. "Ready" is displayed when nothing has been started, once a ride has started the number of that ride is shown along with the status of the recording and the elapsed time for this ride. When the user touches the start button an animation is played to change the colour and text of the button to that in Figure B.5. The center message is updated to display "Waiting on GPS and BLE" is shown, when GPS signal is found or the device is connected then they are removed from the message. Once both GPS signal has been established and a Bluetooth connection has been made to the MetaWear device then the message changes to "Recording".

This activity is responsible for creating a SQLite entry for each ride when the user presses start.

Ride History

A Rides History view was added to show a list of all journeys taken. This section can be accessed by pressing the menu button or pulling in from the left to display a sliding drawer menu, seen below in Figure B.7, here the user may select to view rides or access the settings menu. On the rides history page a large "SYNC ALL" button is fixed at the top of the screen with a scrollable list of previous rides below.

The large sync now acts to upload all rides that have yet to be uploaded. Once a ride has successfully been uploaded the icon, colour and content of the text changes to clearly show that ride has been synchronised with the web server.

A list of material cards are used to represent a single ride, this is a similar style to that seen in popular Android apps such as the Play Store or Gmail. Each card displays the starting, A, and finishing, B, position of a journey, the time and duration with the title of the card being either "Morning", "Afternoon", "Evening" or "Night" depending on the time the ride was taken. The names of the street shown at the start and end of the journey are obtained by reverse geocoding the latitude and longitude of at the two points. Reverse geocoding is the process of converting a latitude and longitude into a human readable street address. Google's HTTP geocoding API [40] was used on the first and last location of the journey when the journey ends. If an address is unavailable or the user does not have Internet access at that point then the raw latitude and longitude is used. This was purely for cosmetic reason to provide the user with a reminder of where they went to and from. Touching the blue location for the start or end of a journey sends an intent to open this location in the users default map application, for example Google Maps. The user may delete a ride simply by swiping it to the left or right, again this is similar behaviour that is seen in many popular apps like Gmail or Google Keep, because of this the user is intuitively familiar with these actions. When a ride is deleted by swiping its card off the screen, a Snackbar is shown at the bottom of the screen seen in Figure B.9.

A Snackbar is defined as providing lightweight feedback about an operation, they automatically disappear after a timeout or after user interaction elsewhere on the screen. Snackbars can be swiped off screen. Snackbars can contain an action [34].

The Snackbar provides information about the ride number that was deleted and gives the user a window of time when they can undo this deletion.

This approach of allowing the user to undo their deletion was achieved by adding the ride to a queue of pending deletion rides. If the user presses undo before the Snackbar is dismissed by time or the user interacting with anything other than the Snackbar, then that ride is removed from pending deletions, a new Snackbar message is shown to say the ride has been restored and the ride is pushed back to its position in the list of cards. If a user does not press undo the ride remains in pending deletions until the Snackbar is dismissed at which point the ride and all its associated data will be permanently erased from the phone.

MetaWear Library

MetaWear provide an API [19] for communication with the MetaWear board, this API makes it possible to achieve a number of actions with the MetaWear board. Small programs can be loaded over BLE onto the MetaWear that can sample the accelerometer at a given frequency and perform simple calculations and operations on board on the data. The board can log measurements within its own 16 kB of RAM and have the phone download them in bulk, this saves bandwidth in comparison to continually sending measurements. Requests can be sent with the API to query battery level or operate the LED. To control the LED three bytes with a value limited between 0 and 31 are sent to board to represent the intensity of the red, green and blue LED.

Location Service

In Android a service is defined as an application component representing either an application's desire to perform a longer-running operation while not interacting with the user [32]. For this application, four services were used, location recording, MetaWear recording, uploading rides and to scan for new devices.

Services let an application component run in the background regardless of whether the screen is off or the application is closed. This was achieved by acquiring a partial wake lock for that service, this prevented the CPU going asleep but allows for the screen to be turned off. At the end of each service it is important that each wake lock is released.

A location service is launched by the main activity when the user presses the start button, the service is passed the ID of the ride that has just begun. A file name derived from the ride ID and the start timestamp is used to create the CSV file that the data is logged to. This file name takes the form of <rideID>_<timestamp>_<suspension_type>_<geo|acc>.csv, for example 46_1450272730_2_geo.csv. This naming convention ensures uniqueness across file name and it serves a secondary purpose of storing some metadata about a ride making it possible to still make use of the data even if the metadata gets erased.

A request is made to the Location Manager to receive updates on the location of the phone. Within this request the location provider is chosen from either a low accuracy cell tower network provider and a higher accuracy GPS provider. A minimum time and distance is also specified. These define the minimum change in time and distance to have passed before getting updated. For this application the GPS provider was chosen as accuracy and speed data is very important. The minimum allowed time and distance was picked to ensure regular updates of the location. The real world maximum rate achieved is 1 Hz.

The location data that is recorded consists of timestamp, latitude, longitude, accuracy, speed, altitude and bearing. For each reading received speed, altitude and bearing may not be available. This is the case because speed is obtained as a result of the Doppler effect on the GPS signals meaning if the user is moving at a slow pace or stationary it is possible they have a GPS fix but since the Doppler effect can not be observed at that time, speed may not be available, similarly for bearing if the user is not moving or moving slowly it is difficult to know the direction they are travelling, again resulting in this value being unavailable. Altitude may not be available if there is a high level of error associated with the reading. These values are continually logged to a CSV file. This method of continual logging in small batches means that if the battery on the phone suddenly ran out of power or the app unexpectedly ended then the abrupt ending of the ride will not matter and will just mean that ride was cut short.

As discussed previously the idea of a privacy zone was used to ensure the start and end of a journey is never recorded. The privacy zone around the starting point was added by recording the first location in a local variable and only starting to log readings to a file once a location is greater than the required distance from the starting point. The zone around the end point was implemented by having a first in, first out buffer for readings. When a reading is initially recorded it is pushed onto the buffer, only once the first reading is greater than the required distance from the newest reading does it get removed and appended to a file.

Bluetooth Acceleration Service

A Bluetooth service to record accelerometer data is created at the same time as the location service. This service is also passed the ride ID of the newly created ride. An added action is also passed to this service, these actions can be to start the service or to tell the service that a SMS has been received or a call has started or stopped ringing, this information is then used to flash the LEDs on the MetaWear.

When this service is started, it first attempts to connect to the MetaWear. The MAC address is fetched from the Shared Preferences. A BLE connection is then made to this device and when connected a request is made to get the battery status of the device. If this attempt fails the service continually tries to connect until a successful connection has been established. The operations to record the acceleration values are sent to the MetaWear, when these have been successfully registered by the MetaWear and recording has begun a request is made to flash the LED green twice.

The data recorded in this service is the timestamp and the acceleration about the X,Y and Z-axis. These values are also continually logged to a CSV file defined by the previous naming convention.

If this service receives a message with an action of CALL_START then a sequence of rapid bright purple flashes of the LED is made and sent to the MetaWear to be executed, if the action is CALL_END then a command is sent to stop all flashing and if the action is SMS a sequence of 4 orange flashes is sent. The purple flash consisted of the red LED being at full intensity, 31, and the blue LED being at a lower intensity of 20. Similarly orange was created by sending a value of 25 for red and 10 for green. To stop the possibility of accidentally flashing indefinitely if, for example if the CALL_END action was never received, a timeout of 60 seconds was used to stop flashing unless instructed to continue.

These services had to expect there to be times when no GPS signal was available or that the MetaWear device was unreachable or had no power left in its battery and to handle these cases gracefully.

Both of these services also provided queryable data that the main activity could read to update the UI. This included the user's current speed and the accuracy associate with the most recent location reading, the battery level and connection status of the MetaWear.

Upload Service

For data collected to be uploaded a number of different approaches were looked at. The chosen approach was to use the Android Asynchronous HTTP Client provided by loopj [3].

When this service starts it queries the SQLite database to find all the rides that have not been uploaded yet. If there are no rides that have not been uploaded then the service ends. Otherwise each ride is uploaded over HTTP using a HTTP PUT request, this request is idempotent meaning if the app accidentally uploads the same ride twice, it will be treated as the same ride and not two separate rides.

A unique URL to PUT the data is created by concatenating the MetaWear MAC address, a unique ID for the smartphone and the ride ID being uploaded. The CSV files corresponding to that ride are loaded into memory and if no CSV file is found or that file is empty then that ride is skipped. A multipart HTTP request is formed comprising of the metadata relating to the ride and the contents of the two CSV files, the acceleration data from the MetaWear and the location data from the phones GPS.

An attempt is made to upload these to a web service running on Heroku. This web service returns a 200 OK HTTP status code if the file is successfully uploaded. On receiving an asynchronous HTTP response, if it is successful, the entry for that ride in the SQLite database is updated with the time this ride was uploaded at.

BLE Scanning Service

A Bluetooth scanning service used to scan for near by devices. This service is launched when the user first launches the app or they scan for a new device. A 16 bit service UUID is used to limit scanned results to just MetaWear devices. This stops other devices such as phones or smart watches being found. This service returns all the scanned MAC address of near by devices and they are then displayed in a dialogue for the user to choose from.

LED Notification

An intent filter was used to tell Android that we want a specific part of the application to run when a call or SMS message is received.

When a call or SMS event is triggered, Android checks what applications are registered to filter this intent. A Broadcast Receiver was used to receive these events, this receiver inspected the received intent and sends a CALL_START, CALL_END or SMS action to the Bluetooth acceleration service if a ride is ongoing.

NFC Integration

To increase the ease of use and make it possible to use the device while wearing gloves a small NFC sticker was stuck to the case. This was placed under the rubber outer casing to increase resistance to weathering. A photograph of this NFC tag can be seen in Figure 3.6.

To start a ride a user can touch their phone, providing it has NFC capability, to the device. This will toggle a ride to start or end. This feature was added by first manually writing the same URI corresponding to the application to each NFC sticker. An intent filter was created in a similar way to that of receiving SMS and call intents. Only scanned NFC tags with the appropiate URI is registered by the application.

NFC tags are not capable of starting and stopping services but they can start activities. To get around this problem a blank hidden activity was used. This activity is started when a NFC tag is scanned, it is then responsible for checking if a ride is currently in progress, if there is then that ride is signaled to stop, otherwise a new ride is started. This hidden activity launches the main view of the application so the user gets visual feedback that the ride has been started.

Ongoing Notifications

Ongoing notifications were used to reduce the possibility of people forgetting to stop a ride. These act similarly to normal notifications except they can not be dismissed.

When a ride is being recorded an ongoing notification is displayed as seen by the bike icon in the upper left hand corner in Figure B.5. This notifications persists even if the user

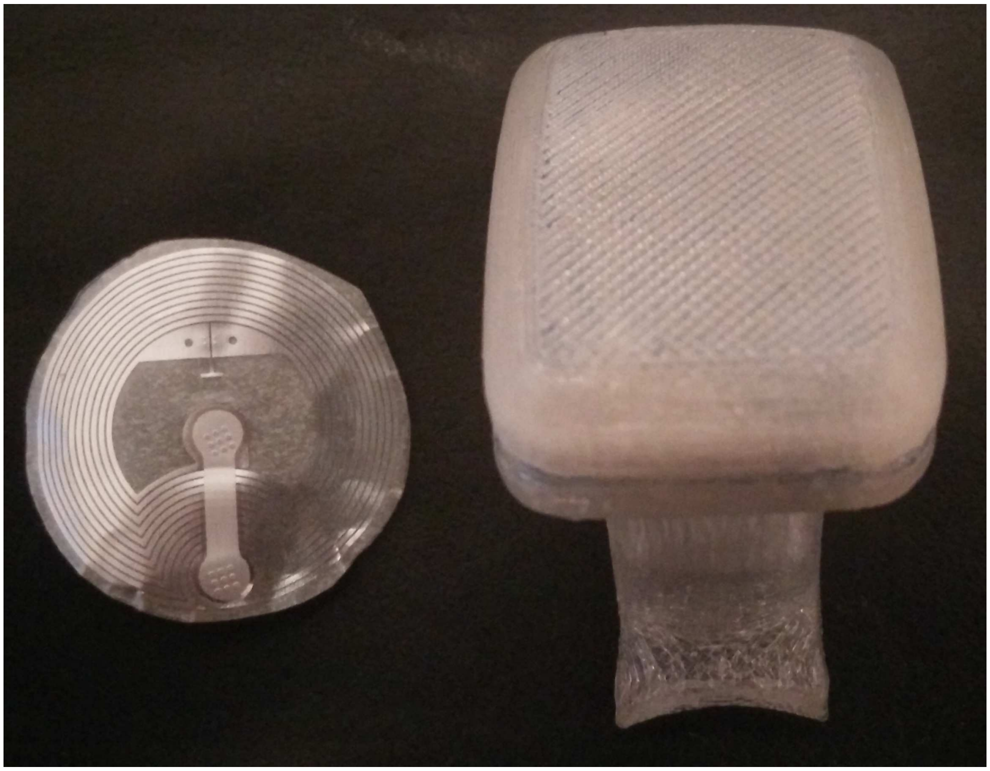


Figure 3.6: NFC tag beside 3D Printed Case

closes the application or opens a different app. This serves to remind the user that a ride is currently being recorded. Pulling down the notification bar reveals more information as seen in Figure B.6, informing the user that a ride is recording and clicking on this message will bring the user to the main page of the application. Linking the notification to the applications main screen makes it easy for the user to navigate back to stop a ride.

3.4.3 Obstacles

When developing this application a number of implementation issues had to be overcome. The first issue encountered was when using Bluetooth on Android devices older than Lollipop. These older versions have a different API to interact with Bluetooth. To support these older devices a lot of the code had to be duplicated and rewritten to use these older and more limited APIs.

When storing the data two main approaches were considered. A custom web service and storing the data in S3 and MongoDB or to use SiteWhere [33]. SiteWhere is a time series database wrapper for MongoDB. It provides a UI to browse data, a HTTP interface for inserting events and an Android library. Initially SiteWhere seemed like the ideal solution. While integrating their library it was found that it was out of date and targeted a far older version of Android.

As soon as this version issue was overcome another problem arose with SiteWhere.

SiteWhere only allows measurements to be uploaded individually over HTTP. This makes it extremely slow to upload a ride as it has to continually open and close a HTTP connection. For an average 25 minute ride recording acceleration data at 50Hz and location data at 1Hz a total or 76,500 readings are recorded, $25minutes \times 60seconds \times 50hz = 75000accelerometer\_readings$ and $25minutes \times 60seconds \times 1hz = 1500location\_readings$ totaling $75000 + 1500 = 76500total\_readings$.

Making 76500 request to the server for a single typical user with one ride is very expensive and not practical. It was found that a small ride took over an hour to upload while connected to WiFi on a new Android phone. With the average user making two journeys per day and there being 10 users, it was clear that this approach would not work.

The CSV files containing the 76500 readings are very small being typically under 3 MB which is the only size of an average photograph. It would be expected that this would be uploaded almost instantly.

A different approach of hosting a web service on Heroku was taken which lets files be streamed on a single HTTP connection reducing the upload time to one or two seconds per ride. This is a much more reasonable time.

3.5 Web Server

For the reasons discussed previously a custom web service was created to handle uploading of the rides.

This web server was create with Flask [11] in Python. A URL that accepts PUT requests was exposed that takes the form of /<phone_id>-<device_address>/ride/<int:ride_id>. When a file is uploaded the phone ID, MetaWear MAC address and ride ID is parsed and the validity of these is checked.

The acceleration and location CSV files are examined to see if they meet the maximum size constraint and are in the appropiate format. These files are then read from the socket into memory and sent to S3 where these raw files will be indefinitely stored. A metadata JSON document is created, this records the location of the files in S3, phone ID, MetaWear MAC address, ride ID and suspension type for that ride. The metadata is pushed to a Redis Queue, a copy of this document is stored in S3 for backup and debugging purposes. A 200 OK status code is returned if the this uploaded process is successfully.

This web server was deployed with HTTPS support to increase the security and reduce the possibility of participants data getting leaked, this was enabled by Heroku.

A private GitHub repository was used to host the code for the server, this was connect to Heroku to allow for one click deploys.

3.6 Storage

The storage technologies used were Redis, S3, MongoDB, CSV and JSON files.

3.6.1 Redis

Redis is a fast in-memory database. Data is stored using key-value pairs. Redis was used to provide a queue messaging service for the system. A Python class was written to push metadata documents to the end of a queue and to retrieve documents from the start of a queue. This was implemented by using a list in Redis under the key of the queue name. To put an entry on the queue, it is serialised into JSON and the RPUSH command was used to push a it to end or right hand side of a queue. To get an entry from the queue, the BLPOP command is used. This executes a blocking list pop from the start or left of the queue. This allows the a worker process to long poll the queue with the get method.

Amazon's Simple Queue Service was considered for the role of a messaging queue. This was not picked as it was felt that less reliance on proprietary services would be better, as well as this a Redis database is needed for the integration with Smart-GH meaning this queue can piggyback of this already deployed database.

Smart-GH reads the information needed to rate roads from the Redis database. This is stored with the road name as the key and quality of that road as the value.

3.6.2 S3

S3 is an online file storage web service offered by Amazon. This was chosen to store to raw files as their free tier offers 5 GB of storage that get reset every month, i.e. every month, 5 GB of new data can be stored for free. Amazon provides libraries for most languages making it seem as though the files are locally stored.

The function of S3 for this project was to store the raw acceleration and location CSV files along with their JSON meta files. This meant that workers running on any machine would all have access to the same data. As Amazon resets the 5 GB limit every month, there was no negative drawbacks to never removing the raw data. Indefinitely storing the raw readings made it very easy to safely test and develop without the worry of permanently removing important information. It also meant that post processing or additional analysing of the data could be done at any time.

3.6.3 MongoDB

MongoDB is a highly scalable document-oriented database. The primary reason for using this instead of a relational database like MySQL was because of its geospatial support. MongoDB makes it very easy to perform geospatial queries like find all documents within X meters of this point or inside a given polygon, with location data being a large part of this project it was a crucial feature to have. The secondary reason for using MongoDB was due to there already being a MongoDB running on a college server.

This database served the role of storing rides metadata, recorded measurements and data derived from the raw readings.

3.6.4 CSV and JSON files

CSV and JSON files are used for storage and transfer of data throughout the project. As discussed, CSV files are used to store the raw accelerometer and location data both on the smartphone, in S3 and to transfer it between them. CSV files were also used to store the ranking of roads.

JSON files are used to store the metadata for a ride, they are also used in the heat mapping process with the quality of a location being stored in JSON files which are feed to the JavaScript UI.

3.7 Web Worker

Once a ride is successfully uploaded and added to the Redis queue, a worker asynchronously pulls the ride from the queue and process it. The act of decoupling the processing of the data to the uploading of the data means that the web server can quickly complete the uploading of a ride, saving time for the user. With this architecture more workers or web services can easily be created to deal with the required load.

The processing and analysing of the data was managed by two different processes. The first is a web worker daemon that continually transforms and loads rides from the Redis queue into a MongoDB. This process retrieves the ride as soon as it is placed into the Redis queue. The second is a Cron job scheduled to run every N days, this process aggregates all of the rides and produces the data for the UI maps and road quality ratings for Smart-GH.

The web worker is a Python daemon running in a continual loop, when a ride has been pushed to the Redis queue the daemon retrieves it, joins locations with acceleration measurements, performs light processing and inserts it in the MongoDB database.

The location CSV data is first loaded into memory from S3. This CSV data is used to initialise a path. The idea of a path is used to represent a single route of location data. The path is responsible for cleaning and smoothing the GPS data by snapping it to the road and then reverse geocoding each coordinate to resolve a street address. A ride is used to represent acceleration measurements that have an associated location.

3.7.1 Snapping to Road

The purpose of snapping a point to the road is to not attribute a reading to the wrong road. As well as computing and storing the snapped coordinate, the raw GPS points are also stored.

The Google Snap to Roads API [35] was used to achieve this. This API takes up to 100 GPS points collected along a route, returns a similar set of data, with the points snapped to the most likely roads the user was travelling along. It was seen that the 100 readings limit was quite restricting and did not always produce coordinates snapped to the correct road due to them being so close to each other. This was due to the user's location being recorded every second meaning 100 readings would be only 100 seconds, if the user was travelling slowly or stopped for part of those 100 seconds the API would find it difficult to tell what direction and on what road the user was travelling. This effect was particularly noticed at cross road traffic lights when the user intended on going straight. But when they were stationary at the side of the road it could appear that they were taking the left turn due to errors in GPS accuracy.

To combat this two precautions were taken. Every 5th point was taken and fed into the API, this would then return 100 anchor points that were 5 GPS readings apart covering at least 500 seconds of the route. Next an overlap of at least 20 was used when entering 100 contiguous points which means that the last 20 from each batch are used to seed the first 20 for the next batch. This method results in every batch, excluding the first, having at least 20 from the previous batch snapped to the correct road and 16 from every 5th point in the last 80 locations totalling at least 36 points out of every batch of 100 already being snapped correctly. The size of the overlap was calculated as seen in Listing 2. In this example a path of 1380 locations results in an overlap of 24 points, with a step size of 77, this means the first request would contain points 0-99 and the next request would have points 77-176 followed by 154-253. This pattern is continued for all 18 batches.

```
# e.g. num_locations = 1380
interval_size = 100 # max size of batch supported by API
min_overlap   = 20  # minimum number of points to overlap

num_batches = ceil(num_locations / ( interval_size - min_overlap ))
step_size   = ceil(num_locations / num_batches)

# num_batches = 18
# step_size   = 77 -> overlap = 23
```

Listing 2: Snap to Road interval overlap

For each point in a batch the API returns the corresponding snapped point if available. If the user is not cycling on a road or the accuracy for the reading is low then that point may get ignored and not snapped. The process of snapping data aids in reducing the number of roads that are wrongly attributed, it also increases the accuracy when reverse geocoding location street addresses.

3.7.2 Reverse Geocoding Locations

Having a street address to associate with an individual reading has a number of uses. Knowing the street name plays a big part in how Smart-GH routes, it allows for roads to be rated and compared and it makes it easier to analysis and cluster readings from different rides.

After researching a number of different online reverse geocoding APIs, it was found that Googles API offered a large number of requests for free, it does not rate limit and allows for parallelising multiple requests. It was also found that using Google for both snapping the location data and reverse geocoding worked well together. This can be seen with Googles place_id being used across a number of products to represent a unique location.

When looking up a GPS point it is checked if a snapped coordinate exists for that point, it it does then that point is used. For a given GPS point the API returns the top most likely addresses along with the exact point of each address, some of these are vague address with just the county or country the point is in, others are more accurate pointing to an exact street number. The most accurate result is taken if available, otherwise that point does not get an address.

Individually geocoding every point in a path is very expensive both in terms of processing time and using up the limited requests from the free tier. For example a 40 minute journey contains about 2400 GPS points, at an optimistic rate of 4 requests per second, it would take at least 10 minutes just to look up each point. This would be a huge bottleneck.

Two features were added to reduce the reliance on Google and lessen this bottleneck.

Before looking up a point with Google its own paths cache is checked. For each point in a path the 50 previous locations are checked. If any of these are within a given distance, 2 meters was chosen to ensure high accuracy, and they are a true address, this point takes that address as its own. We define a true address as one that has been looked up with the API and a precise location was found. This method of checking its path greatly reduces the reliance on Google and having the previous locations in memory result in a hit being a few thousand times quicker than making a HTTP request. This approach means that there will only be at most one request every 2 meters, leaving a large amount of points still needing to be looked up with Google.

The second feature added aims to further diminish the bottleneck and the reliance on Google by checking the MongoDB in a similar way to that of checking the paths cache. After a location has not been found in the paths cache and before it is looked up remotely with Googles API, it is first checked in the local MongoDB if any other ride has ever gone within 2 meters of that point and if it was looked up with the API to make it a true address. If one is found then that address is taken as the address of a given point. A geospatial index is placed on the coordinates of a measurement in the MongoDB database, this makes it very quick to access data within a radius of a given point. MongoDB caches the most recently accessed data in RAM. If the working data set fits in RAM, MongoDB serves all queries from memory. Because of this a geospatial query can be as much as ten to a hundred times quicker than looking it up online.

The power of these features is seen when a new user takes a route for the first time and the majority of it is already known to the system due to other users taking overlapping routes.

Only true address points are considered to remove the possibility of addresses creeping along a road, where it continually takes the location of the point before and if the user never travels more than 2 meters per second then the full journey would end up with just one street address.

3.7.3 Merge Location and Acceleration Measurements

Acceleration data and location data are both indexed by timestamp, with the MetaWear recording at 50Hz and GPS recording at a maximum of 1Hz, there is going to be at least 49 measurements that have no location associated with them. Given that acceleration values are recorded with an accuracy of a single microsecond and location is recorded with only a precision of one second, it is very likely that none of these timestamps will directly correlate.

The process of assigning a location to an acceleration measurement involved linearly interpolating between the two nearest locations and calculating an intermediary location. Every measurements timestamp was looked up in the path to find the two nearest locations, this can be seen in Listing 3.

```python
prev_location = locations[0]
for location in locations:
    # >= handles if measurement_timestamp equals a location timestamp
    if (measurement_timestamp >= prev_location.timestamp and
                            timestamp < location.timestamp):
        return (prev_location, location)
    prev_location = location
```

Listing 3: Finding to two nearest locations

If a measurement has a timestamp between two locations in the path, then those two locations are used. These two points are linearly interpolated to find the most likely position at the time of the measurement. This is seen in Listing 4. The percentage a measurements timestamp is from the first location is calculated and then used to calculate the distance the measurement is from the first location. The direction of travel is computed by taking the bearing from the first point to the second. The bearing and distance is used to travel that length from the first point in the direction towards the second point.

A new location is then created and initialised with the destination_point and computed bearing. The speed and altitude of this new location is interpolated in a similar manner using the percentage from the first location. The accuracy is assigned the worse value of the two locations used in the interpolation. This was done as measurements will only be as good as the least accurate measurement. If the acceleration measurements timestamp was closer to the first location then that geocoded address is used if available, otherwise the address of

```
# how close the timestamp is to the 1st nearest location
percent_from_loc1 = ( (measurement_timestamp - loc1.timestamp)/
                            (loc2.timestamp - loc1.timestamp) )
interpolated_distance = percent_from_loc1 * loc1.distance_to(loc2)

bearing = loc1.bearing_to(loc2)

d = VincentyDistance(meters = interpolated_distance)
start = Point(latitude = loc1.latitude, longitude = loc1.longitude)
destination_point = d.destination(point=start, bearing=bearing)
```

Listing 4: Interpolating two nearest points

the second location is used, if the timestamp is closer to the second location the opposite is done. This process was done with the raw GPS values before being snapped to a road. If it is found that the two nearest locations have snapped locations these snapped coordinates are interpolated in a similar way to give a snapped to the road location for the measurement.

To ensure two location points which are far apart are not used, limits were put in place. These were chosen were as follows: a 5 second limit to the nearest of the two locations, a 20 second limit between the two nearest locations, a 50 meter maximum distance to the nearest location and a 150 meter maximum distance between the two nearest locations. If any of these limits were broken then that measurement would receive no location.

If these limits were not used, a measurement could get a location with a high level of error. For example if GPS signal was lost mid journey and found a few kilometres later this could result in the points, mid journey, receiving locations with an error in the order of kilometres. While omitting locations from these measurements results in less geolocated data, it means that any measurement having a location will be accurate. This approach of choosing quality over quantity was taken throughout the project. It can be seen that if enough users adopted using this system then quantity would not be an issue.

3.7.4 Separation into Sections

Measurements now consists of a 3-axis acceleration value and a location. The concept of a section is used to split up parts of a ride where the cyclist is continually cycling. This separation can be caused by a tempory lose in GPS or connection to the accelerometer or due to the user stopping. Sections make it easier to process contiguous chunks of a rides. This allows one to assume that readings next to each other in a section are next to each other by location and time as well and that the user is moving at this point.

For every measurement in a ride it is checked if they are moving at a cycling speed. This was restricted to over 8 km/h. The average walking speed for a human is about 5 km/h [43], so anything above 8 km/h can be assumed that it is not the user pushing their bicycle.

If the measurement has a speed greater than 8 km/h it is pushed into a section and this is continued until a measurement where the user is not moving at a cycling pace or a gap in GPS locations is encountered. This measurement is ignored and the next good measurement is used to create a new section. Sections are stored by assigning each measurement in a section an integer that acs as an index for the section it is a part of. A visual representation of a ride split into sections can be seen in Figure 3.7.

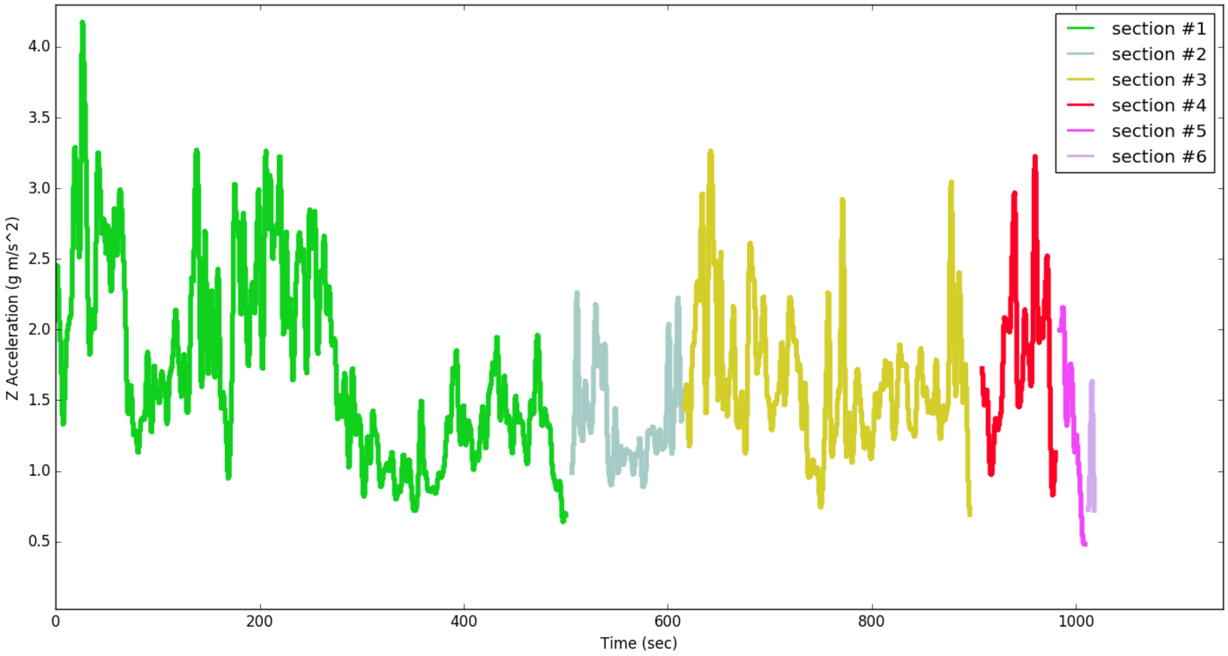


Figure 3.7: Sample ride with 6 sections

3.7.5 Axial Calibration

Participants were advised to attach the device to their bicycle on the stem as seen in Figure C.1. Sometimes they might put it on pointing left, other times pointing right. They may, out of habit of using other bike lights, attach it to the main handle bar. As well as users attaching it to different locations on the bicycle, they will also attach it at slightly different angles around the pole each time. All of these different placements make it impossible to directly compare the raw acceleration data in an accurate and fair way.

The raw acceleration data from two rides can be seen in Figure 3.8 and Figure 3.9. These graphs show how for different rides, the mean acceleration along the axes are very different. Directly comparing the readings from the Z-axes, it can be seen that there is a difference of 0.634 in the mean resulting in all the readings in the first graph appearing to be far larger.

Designing a Method

The initial approach taken was influenced by the method of manual calibration used by Biketastic in the research section. This method involved holding the bicycle still at the start of the ride and waiting for the accelerometer to record measurements of the stationary bike. These measurements were then used to rotate all further measurements. A process was developed to mimic this action automatically. The approach taken was to scan the acceleration data for stationary sections. These were defined to be groups of points longer than 3 seconds where all acceleration data for the 3 axes were approximately equal and the magnitude of these 3 axes was approximately equal to gravity, these approximations were with a precision of 2 decimal places. These stationary points tended to be at traffic lights or at the start or end of a journey. Calibrating by these points seemed like a good idea but turned out to have two major flaws. When a cyclist is stopped at a traffic light they rest the bike at an angle against their leg with the handlebars also turned to the side at a variable angle. The second flaw came when a ride has no stationary points. This could happen if recording starts during the ride and is ended before the cyclist stops. A sample of this can be seen in the first sample ride shown above. This would result in the ride remaining uncalibrated.

Figure 3.10 is a zoomed in section of Figure 3.9 with the exception of stationary sections not being removed. Inspecting the yellow X-axis it can be seen that the two stationary points do not align with the true mean value with there being a 0.14 difference. It can also be seen that including these stationary increased the overall mean along the X-axis by 0.011. In this sample this was caused by the bicycle pointing to the right while the cyclist was stopped. This realisation lead to the opposite approach being taken where instead of being relied on, the stationary points were fully omitted from the calibration process.

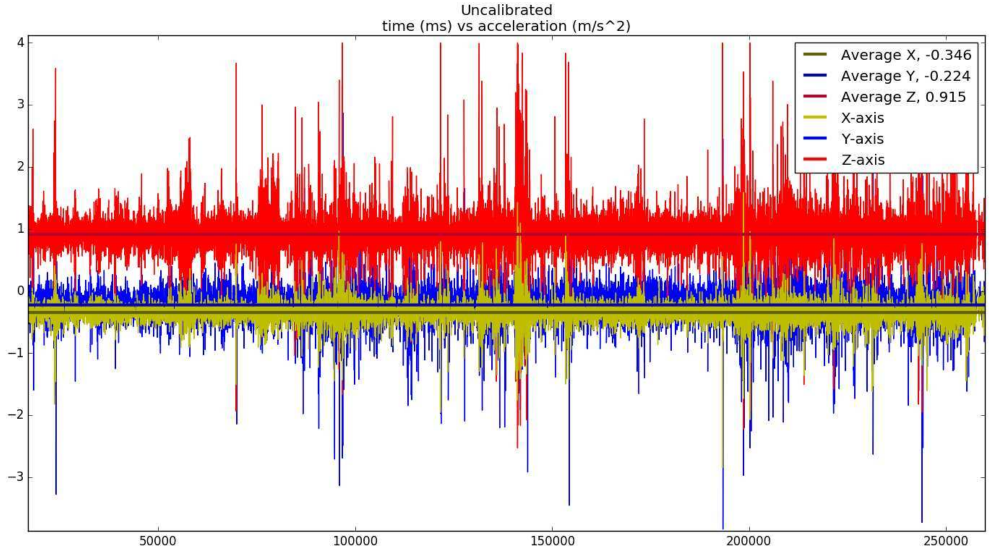


Figure 3.8: Uncalibrated continuous ride with no stationary points

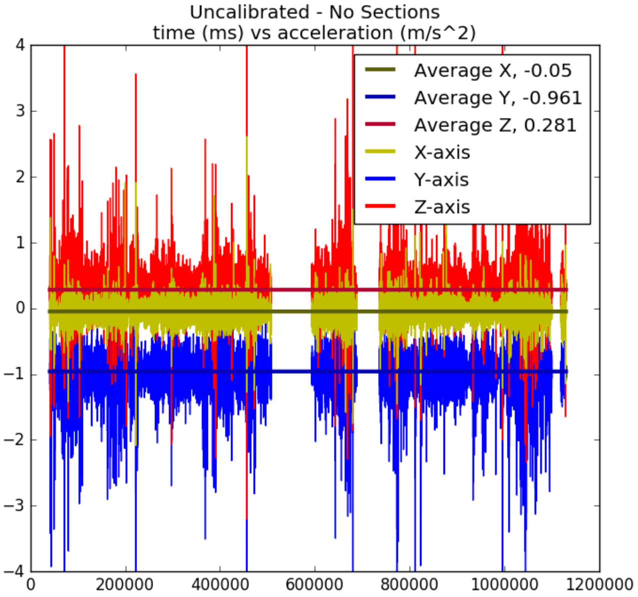


Figure 3.9: Uncalibrated ride with two stationary points ignored

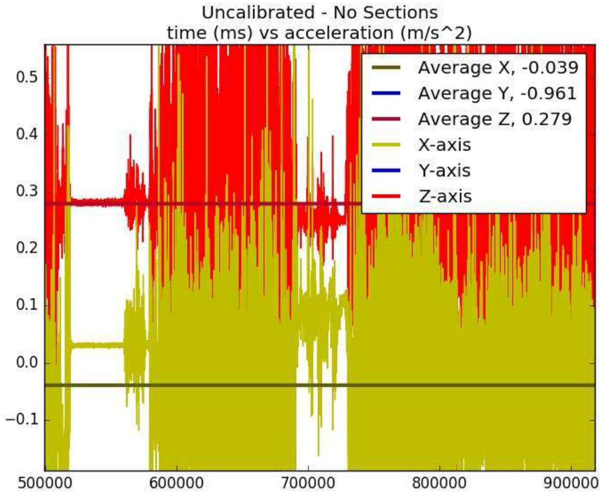


Figure 3.10: Enlarged Figure 3.9 with stationary points kept

After the mean vector to align is found two approaches were looked at to rotate it to align with gravity along the Z-axis. The first approach will be described here along with how the error was found.

This approach used the cross product to derive the axis perpendicular to both the mean vector and gravity. The dot product was then used to compute the angle between the vector and the Z-axis about the derived axis. All measurements were then calibrated by rotating them about the derived axis by the magnitude of the angle computed. This worked well providing the device was rotated about just one axis. For example if the device was perfectly centered on the stem about its Y-axis and the angle of the stem with the frame rotated it 30 degrees about its X-axis, then this approach would work well as the derived axis would end up being the Y-axis. This approach fails when there were rotations about multiple axes.

For example if the device was rotated 30 degrees about both the X and Y axis, i.e. the stem is at a 30 degree angle with the ground, X-axis, and the device is rotated 30 degrees about the stem, Y-axis. The vector for this position can be calculated by rotating $< 0, 0, 1 >$ about the X axis by 30 degrees followed by the Y axis resulting in a vector, $V =< 0.4333, 0.5, 0.75 >$. Calculating the cross product between V and and the Z-axis produces the derived axis $V \times Z = C =< -0.5, 0.4333, 0 >$. The angle between C and the X-axis can be calculated with the dot product to get $40.89°$, this represents the amount this vector has been rotated about the Z-axis. This calculation can be seen represented in Figure 3.11 where V is in black, the cross product, C colored in pink and the angle between the C and the Y-axis is in green.

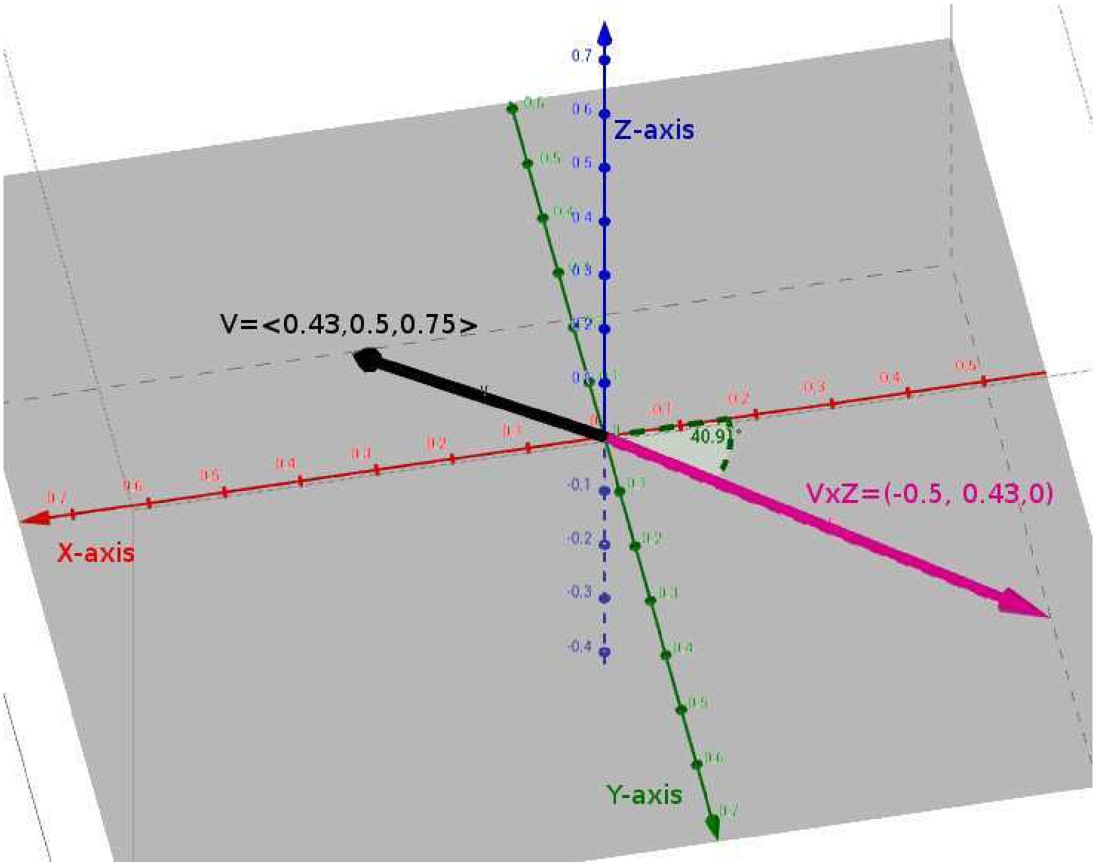


Figure 3.11: Rotating vector about derived axis

Seen in the above example, a small rotation of 30 degrees about the X and Y axes results in the calibrated measurement being aligned with gravity but rotated 41 degrees about the Z-axis.

The error in this approach was noticed when the placement of the device resulted in a rotation about the Z-axis of over 60 degrees causing the X and Y-axis to become almost flipped where the X-axis was closer to the true Y-axis and vise versa.

Integrating Axial Calibration

To correct this, the following calibration process was designed. The goal of this process is to compute a matrix that will align these measurement vectors with the acceleration felt by gravity which is pointing along the Z-axis, G=<0, 0, 1> and that they are orientated as defined in Figure 3.12.

The mean value of the X, Y and Z-axis from all measurements in sections is computed. These 3 averages produce a 3D vector V=<avg_x, avg_y, avg_z>. Due to the conservation of energy, the magnitude of this vector will be approximately equal to that of grav-

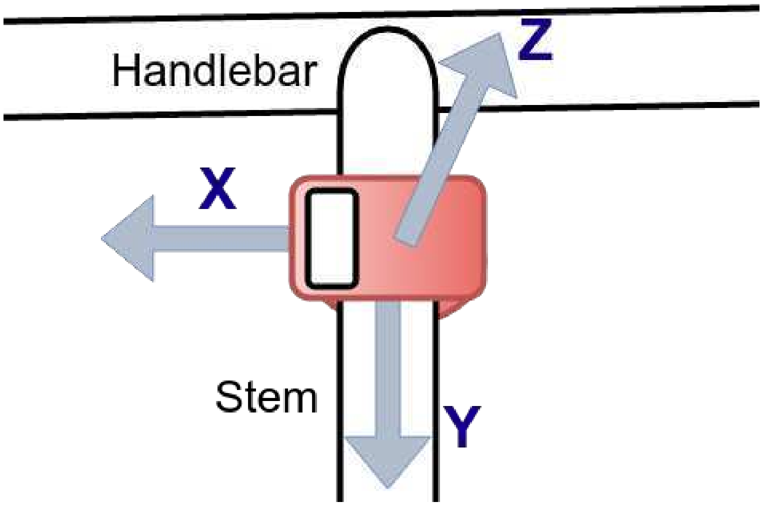


Figure 3.12: Placement of device and axis definition

ity. It was seen in every ride that this mean vector was approximately equal to gravity to within 3 decimal places. In the above example in Figure C.1, the mean vector was $< -0.345562864417, -0.223629958118, 0.91466921245 >$. The magnitude of this can be calculated to be 1.00301g m/s^2.

The magnitude of the angle about the Y-axis between the vector, V, and the YZ plane is calculated as follows using the dot product:

$angle = cos^{-1}(V.z/\sqrt{V.x^2 + V.z^2})$

A right-hand rule, counter-clockwise system was used to compute a rotation matrices. To know if it should be rotated clockwise or counter-clockwise the sign of the X component is checked. If it is positive, the angle is multiplied by -1 causing the rotation to be in the clockwise direction. A rotation matrix, M_y, about the Y-axis with this angle is them computed. M_y is applied to the vector, V forming a new vector, V_2 which lies on the YZ plane.

A similar process is now used to calculate a matrix to rotate V_2 about the X-axis to complete the alignment with gravity. The angle between V_2 and the Z-axis is calculated as follows:

$angle = cos^{-1}(V.z)$

The sign of the Y component of V_2 is checked. If it is positive, then the angle is multiplied by -1. A rotation matrix, M_x, about the X-axis is obtained using this angle. Applying M_x to V_2 would result in <0, 0, 1> aligning it to gravity. The final rotation matrix was computed by performing the dot product on M_x and M_y.

$M = M_x \cdot M_y$

The Python code to perform this calibration can be seen in Listing 5.

This rotation matrix is then applied to all the measurements in a given ride. This is done

```
avg_x = sum(x)/len(x) # find the average value for each axis
avg_y = sum(y)/len(y)
avg_z = sum(z)/len(z)
V = norm([avg_x, avg_y, avg_z]) # normalise vector
angle_rot_y = acos( V[2] / sqrt(V[2]**2 + V[0]**2) )
if V[0] > 0: # Check if clockwise/counter-clockwise
    angle_rot_y *= -1.0
# rotation matrix around Y-axis
rot_matrix_y = np.array([
                        [cos(angle_rot_y),  0, sin(angle_rot_y)],
                        [0, 1, 0],
                        [-sin(angle_rot_y), 0, cos(angle_rot_y)]
                    ])
V2 = rotate(V, rot_matrix_y)

angle_rot_x = acos( V2[2] )
if V2[1] > 0:
    angle_rot_x *= -1.0
# rotation matrix around X-axis
rot_matrix_x = np.array([
                        [1, 0, 0],
                        [0,  cos(angle_rot_x), sin(angle_rot_x)],
                        [0, -sin(angle_rot_x), cos(angle_rot_x)]
                    ])
# combine two matrix
rot_matrix = np.dot(rot_matrix_x, rot_matrix_y)
```

Listing 5: Calculating calibration matrix

by using the dot product, $calibrated\_measurement = rot\_matrix \cdot measurement$. As well as rotating, each measurement is also zeroed, this involves subtracting gravity, i.e. removing 1 from the value of the Z-axis. A sample of a ride after being calibrated and zeroed can be seen in Figure 3.8.

All the measurements in a ride have now been rotated about the Y axis followed by the X axis to align them with gravity. With these two rotations it can be assured that the Z acceleration values from multiple rides are along the same axis. This is not true for horizontal acceleration on the XY plane. Users were advised to place the device as seen in Figure 3.12 with it pointing to the left while they are sitting on the bicycle. As people are not perfect and sometimes forget to put it on pointing left or right or that this position is not very practical for their bicycle. It was found that it was not acceptable to just assume they had it on the correct way.

To combat this a final two steps were added to the calibration process. The goal of the

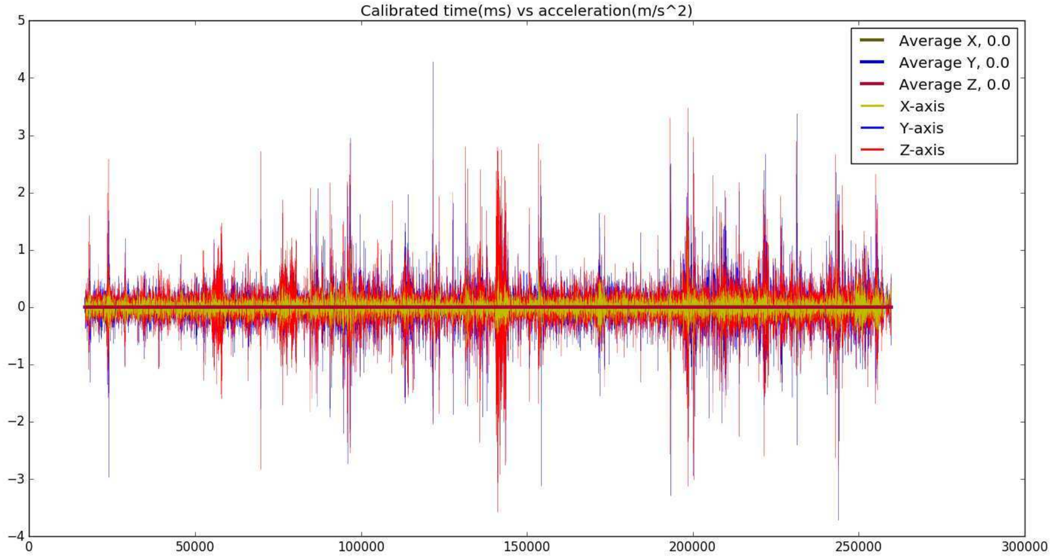


Figure 3.13: Sample Calibrated and Zeroed ride

first step is to find out if the device is attached to the stem or handlebar and if it is on the handlebar to rotate it so that it is in line with the stem. The basis of this relies on the acceleration along the Y-axis being correlated to that of the Z-axis. This happens as when a bicycle collides with a bump, it is not accelerated directly up, part of that acceleration is along the direction the cyclist is travelling. This correlation can be seen in Figure 3.14. This allows for the discerning of the X-axis and Y-axis. The absolute value of 3 axes have a Gaussian filter applied with a standard deviation value of sigma = 120. With the accelerometer recording at 50Hz, 120 provides around a 2.5 second window over which the smoothening occurs.

It is known at this point that the Z-axis is correctly orientated, the two axes are compared with the Z-axis. This was done by calculating the root-mean-square deviation (RMSD) between the X and Z axis and the Y and Z axis.

$RMSD\_X - Z = \sqrt{\sum_{i=1}^{n}(X_i - Z_i)^2/n}$

$RMSD\_Y - Z = \sqrt{\sum_{i=1}^{n}(Y_i - Z_i)^2/n}$

$RMSD\_ratio = RMSD\_X - Z/RMSD\_Y - Z$

A ratio is calculated by dividing the RMSD for the X-axis by the value for the Y-axis. If this value is lower than 1 then the X-axis is closer to the Z-axis. To fix this, each measurement is rotated by 90 degrees about the Z-axis. Experimentally it was found that there was a factor of 5 or more between the two axes, with the true Y-axis being at least 5 times closer to the Z-axis. If the Y-axis is closer then it is already aligned correctly. As the only places the device can be attached are at right angles, the stem and handlebars, it is safe to assume that either the X or Y axis aligns with the Z-axis and not an angle between them.

The final step attempts to find out if the device is pointing left or right. The principle behind this is that the acceleration in the positive Y direction is closer to the positive Z

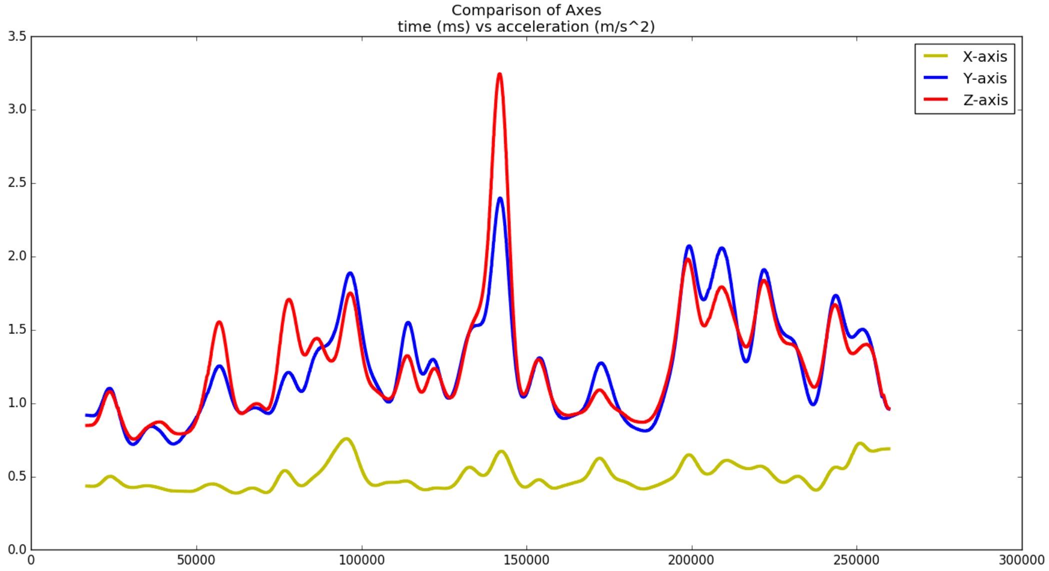


Figure 3.14: Comparing axes of acceleration

direction than that of the negative Y direction. This is due to when a cyclist hits a bump they are accelerated both backwards and up at the same time.

In a similar why to what was described before with the exception of the data not having been made smooth and not using the absolute value of the data, the RMSD of the Y and Z-axis and the -Y and Z-axis were compared. A ratio was computed and if it was found -Y was closer then all measurements were rotated by 180 degrees about the Z-axis. Experimentally it was found that there was a factor of 1.5-2.5 between -Y and Y when compared to the Z-axis.

Finally these geolocated and fully calibrated measurements are inserted into a MongoDB database. As a ride could have hundreds of thousands of measurements and individually inserting them would be very time intensive. MongoDB provides the ability to perform write operations in bulk. This involves building a bulk object of all the documents required to be inserted and sending this to the database. To further speed up insertions, an unordered bulk operation can be used. This can be used when the documents being inserted do not affect or rely on each other. With unordered bulk operations MongoDB will execute inserts in parallel. Consideration was used when deciding what values to be indexed. Indexing fields massively increases query speeds and without indexes on coordinates geospatial queries are not possible. Indexing fields has downfalls though. In terms of space each geospatial index takes up approximately 1.5% of the space used by the database. Indexes can dramatically slow down inserts as each new value being inserted has to be indexed as well this becomes more of a problem as the size of the database grows, because of this only the minimum number of indexes were used.

3.7.6 Error/Fails handling

Before beginning processing and inserting a ride, this web worker tries to connect to the MongoDB database, if it fails it sleeps for a given time denoted by a binary exponential backoff limited at 5 minutes. After sleeping, it will try to connect again and this process will be continued until a successful connection is established. When the ride has been processed and is being inserted, if this fails due to lost connection with the database, the ride is pushed back onto the queue to be tried again later. This approach was necessary as the MongoDB database being used was not always available online. If this was not being done it could result in a ride being removed from the queue and if there was a connection error while inserting then that ride would be lost.

3.8 Scheduled Job

A Cron process is triggered to run every N days and normally takes about an hour to run. For the small group of participants used it was found that every 3 days was regular enough for this to be executed. If the demand was there, it could have been run as regularly as every few hours. Discussed below are the jobs carried out by in this process.

3.8.1 Acceleration Adjustment

The magnitude of the acceleration felt due to bumps on the road varies with the speed the cyclist is travelling. As discussed in the research section, cyclists travel at different speeds, making it difficult to directly compare measurements.

This step uses supervised machine learning to find the correlation between speed and acceleration experienced in the X,Y and Z-axis. Every measurement in the database that is in a section is queried. To improve the speed of this query and memory used, the returned values are limited to just the speed and the 3 axes of acceleration.

Linear regression is used to find the equation of the line of best fit for speed verses each acceleration axis. The method used to fit the line was least squares. This approach aims to minimise the squared error between each point and the line being fit. The equation of the line being fit can be represented by $speed = m * acceleration + c$ where m and c are the two parameters being obtained. These two parameters are then stored for each axis.

A target speed at 5.5 m/s or approximately 20km/h was chosen as the baseline speed. Each measurement is now scaled to its equivalent value if it had been travelling 5.5 m/s. A scaled value is obtained by the code seen in Listing 6.

After adjusting acceleration by the speed these issues were overcome. It was discovered that this step did not need to be run very often as the adjustment parameters, m and c, were seen to change very little as more data was added to the system. If the speed of this Cron job was becoming a concern, this step could be safely skipped once it has been run at least once with a large sample size. But having said that, this step is executed quickly taking under a minute for a database of 20 million measurements. Another approach that

```
target_speed = 5.5 # m/s
line_y2      = m*target_speed + c
line_y1      = m*input_speed  + c
ratio        = line_y2/line_y1
scaled_accel = input_acceleration * ratio
```

Listing 6: Scaling acceleration by speed

could be taken to ensure this step executes quickly would be to query the newest N million measurements and compute the adjustment parameters for them. Where N could be picked based on time constraints.

Each measurement document was then updated to have an additional field of the adjusted acceleration. The reason why it was chosen to update each measurement in the database as opposed to only storing the parameters and computing the adjusted acceleration whenever needed, was to make it easy to search by these adjusted acceleration values. For example writing the adjusted acceleration to each document means that a simple query can be made to find all measurements with an adjusted acceleration along the Z-axis greater than a given value. Without storing these values all measurements would need to be loaded into memory on the application side and have their acceleration adjusted each time and then compared to the required value. Every acceleration value used after this point has been adjusted based of its speed.

3.8.2 Peak detection

Peak detection is used to find all the local maximas in a given ride. The previously described concept of sections were used to ensure measurements next to each other in the database are truly next to each other. Querying by sections offer an easy way to ensure the measurements have been checked to be moving at cycling pace and that they have an accurate location associated with them.

All the measurements from a ride that are found to be in sections are loaded into memory from the database. Measurements are then separated into their sections. Peak detection is then performed for each axis on a section. A Gaussian filter with a sigma value of 20, correlating to about half a second, is used to smooth the section. This value of 20 was then used as the look ahead value when performing the peak detection, this acted as the window size to look ahead from a peak candidate to determine if it is the actual peak. A sample section after peak detection can be seen in Figure 3.15: where "+" symbols represent peaks.

A document object is created for each peak, storing the ID of the measurement corresponding this peak, the axis and value, and the geolocation of this peak. These were then bulk inserted into a peak collection. A sample of this document can be seen in Listing 14.

It should be noted that the value associated with a peak is not the acceleration felt at that point. Due to the Gaussian filter peaks are smoothened to a lower value. The advantage of using this value over the actual raw acceleration reading is that this method treats a longer

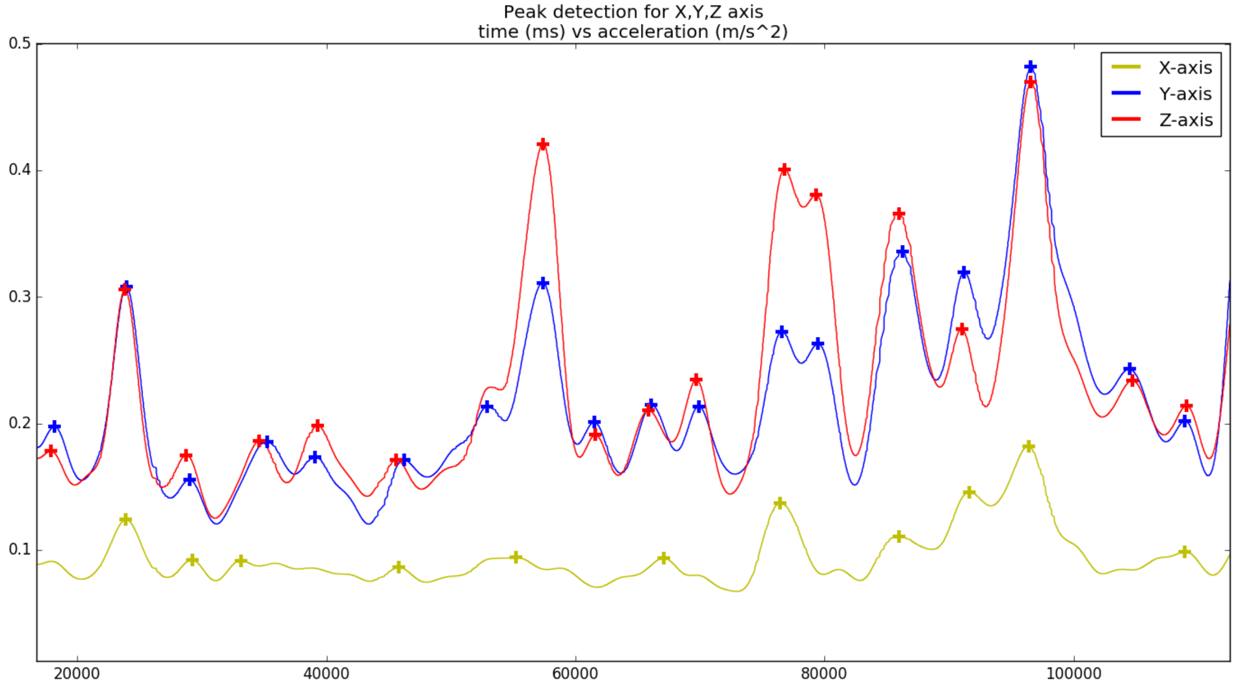


Figure 3.15: Sample peak detection on X,Y and Z axis

lasting bump with a smaller acceleration as worse than a large acceleration over a very short period. This helps to remove noise and outlying single values.

3.8.3 Peak Classification

Peaks are assigned a category of low, medium and high, where low represents a peak from a smooth road and high represents a peak on a bad bump. These categories are also represented as category 0, 1 and 2 respectively. To decide the thresholds that define these 3 categories, it was decided that good represents the lower 50% of all peaks, medium represents the next 45% and high represents the worst 5%. To compute these thresholds, peaks are separated into bins at 0.01 intervals for their peak value. This data is then used to decide the threshold values for the 3 categories. This was achieved by finding the bin that contained reading at the 50% point and doing the same for 95%. These two thresholds were found to be approximately 0.225 and 0.45 and can be seen in Figure 3.15. With these limits the categories can be defined as:

$0.000 \leq Low < 0.225$
$0.225 \leq Med < 0.225$
$0.450 \leq High$

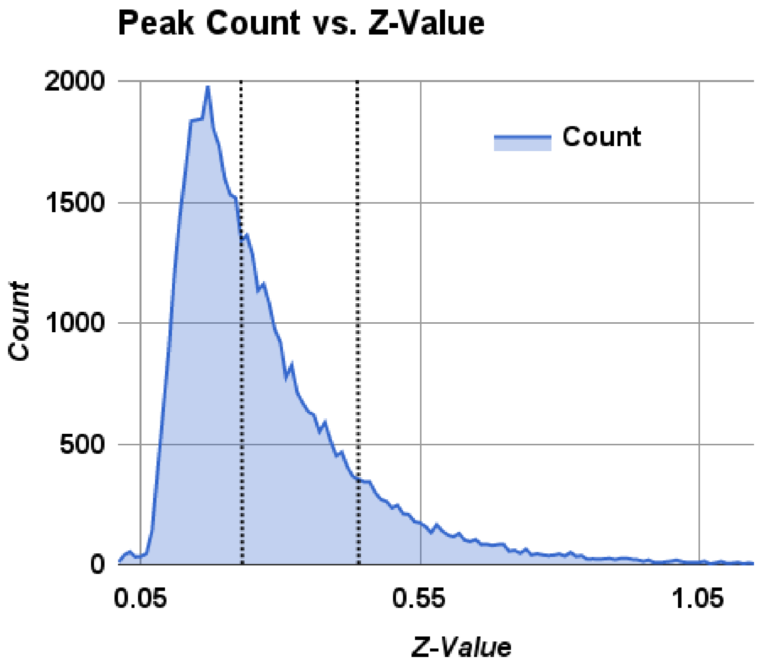


Figure 3.16: Sample peak detection on X,Y and Z axis

3.8.4 Clustering

The objective of this step is to find a single numerical value to represent every distinct part of a road. This was necessary as in the peak collection, a given segment of road could have hundreds of peaks within a few meter radius. This clustering process uses unsupervised machine learning to determine the number of each feature classified by the three categories described previously.

Designing a Method

Different methods of clustering were considered and tested before deciding mean shift was the best fit for this application. Figure 3.17 shows a visual comparison of the five clustering methods that were compared.

K-means clustering provides a fast way to cluster data into an already known number of clusters. This method was not suitable as K-means is developed to work with a Cartesian coordinate system and not geospatial. The biggest downfall of K-means is that the number of clusters K needs to be known prior, while metrics like the Davies-Bouldin index can be used to calculate K this serves to add more complexity thus increasing the executing time.

DBSCAN was the recommend method for clustering location data by a number of online sources. This uses a nearest neighbour approach, if a point is within a distance limit it is

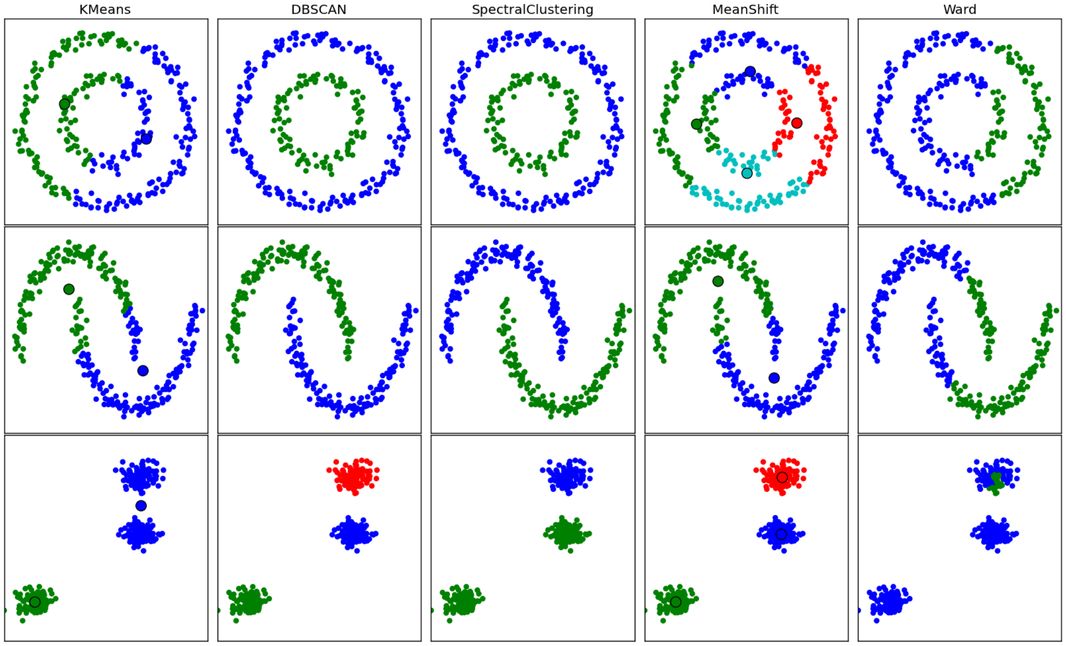


Figure 3.17: Comparison of different clustering methods

joined into a cluster. After some testing seen in Figure 3.18 it was clear that this approach results in a number of features getting clustered together into one long cluster. Looking at the center in red numbered 92, it is evident that this is between at least 2 different features. It was also found that this algorithm was particularly slow to execute.

Spectral clustering was not used for similar reason to that of DBSCAN. Ward clustering uses a hierarchical clustering technique, as seen in the comparison diagram it is difficult to fine tune the level at which clusters are separated. As well as this Ward has a high time complexity and uses a nearest neighbour metric to cluster making it not suitable for this project.

The chosen technique was mean shift. Mean shift works by computing the center of two points. All points within a radius of this center are clustered together and the center, or mean, is recalculated. The kernel is shifted and the process is continued until the cluster converges to one point. This is repeated until all points are labelled. The advantages of this are that the maximum size of a cluster can be defined. The benefits in this can be seen in both the comparison diagram and in Figure 3.19. On comparing this with DBSCAN, it is clear that this approach is far more practical for clustering an unknown number of bumps on a road. It was also found that it scales well and dealt with large volumes of data quickly.

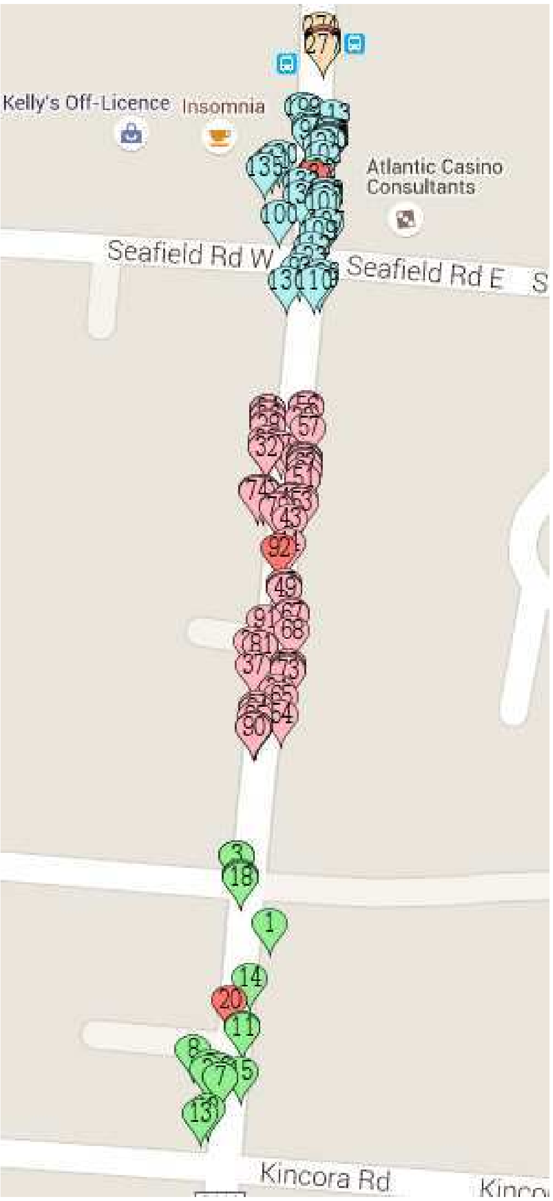


Figure 3.18: DBSCAN of road section

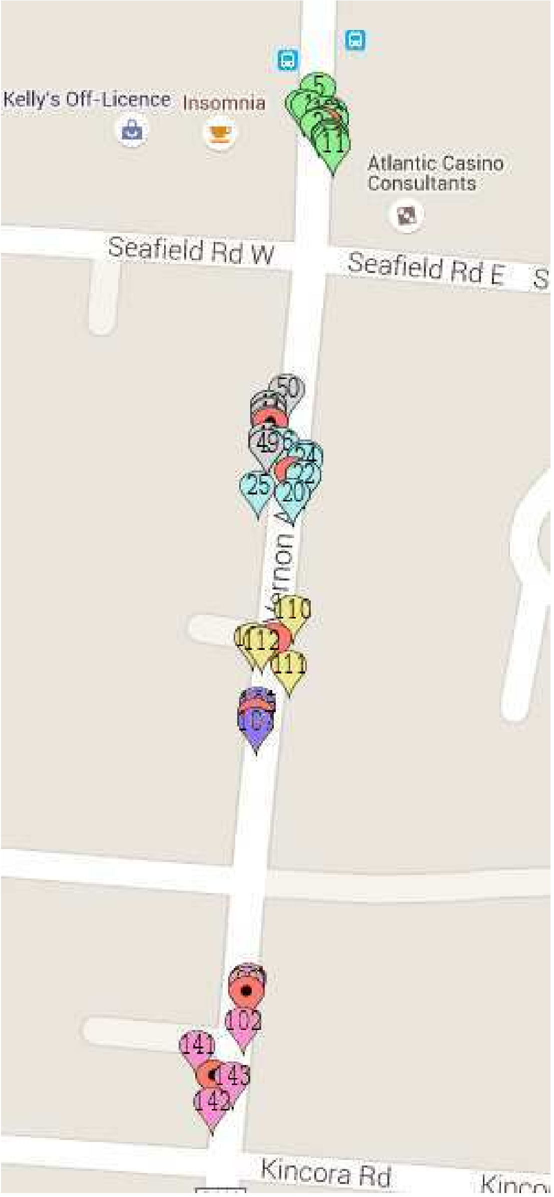


Figure 3.19: Mean Shift of road section

Clustering a road

The distinct query is used to get a unique list of all roads in the peak collection. Each of these roads is individually clustered and the results are written to a cluster collection in the MongoDB database. The objective of a cluster is for its center to represent a single value for the road at that point.

This step begins by querying all the peaks from each category on the given road. An adapted version of the mean shift algorithm to deal with latitude and longitude is used to cluster all the points from the category within a maximum cluster size of 10 meters. This size was chosen to deal with GPS error which was seen to vary between 3 and 15 meters. If a value greater than 10 meters was used, it would result in too many different features getting incorrectly clustered together, anything less than 10 meters was seen to result in many single bumps getting separated into a number of smaller ones due to GPS error.

The low category 0 is first clustered with a threshold minimum cluster size of 1 to ensure every peak with a low category is made into a cluster. No minimum cluster size was used for the lowest cluster as it will be seen later that single cluster value will get overtaken by a higher cluster and it was important that every road segment gets a value, even it is in an area that is infrequently travelled.

This process is repeated for the middle and higher category with minimum cluster sizes of 1 and 2 respectively. The same bandwidth of 10 meters was used for all categories. These 3 steps can be seen in the start of the for loop in Listing 7.

```
for road in roads:
    clusters0 = cluster_road(road, min_cluster_size=1, cat=0, bwidth=10)
    clusters1 = cluster_road(road, min_cluster_size=2, cat=1, bwidth=10)
    clusters2 = cluster_road(road, min_cluster_size=3, cat=2, bwidth=10)
    clusters  = merge_clusters(clusters0, clusters1, clusters2, bwidth=10)
    insert_clusters(clusters)
```

Listing 7: Clustering roads

Next the 3 clusters were merged. The merging process attempted to find what cluster should be used to represent a given area. If a low category has hundreds of points for a cluster but a medium or high cluster has very few points for a cluster in that location then it is likely that the higher cluster was caused by something like a mobile piece of debris or an easily avoidable bump. In that case the smaller high cluster is ignored and the lower category is assigned to that section. This step also acts to filter out old readings. When a large low and higher category clusters are compared the most recent N points are considered. The chosen N for this system was 100 which correlated to about 1 month for popular routes and with more users a higher value could be used and that would provided more up to date data.

This process was implemented by looping through each cluster in category 2. It is checked if this cluster representing a bad section of road should be removed. The center of this category 2 cluster is compared with the center of all lower clusters, 0 and 1, along that road. If any of these lower centers are within a radius of 10 meters of the higher cluster then the two clusters progress to decide if the higher or lower cluster should be removed. A ratio is computed by dividing the number of category 2 points by the number of the lower points in the lower category. This ratio is then further divided by the difference in categories. This is to help differentiate between the cases where a road segment is on the borderline between two adjacent categories, 0 and 1, and it is trying to be decided which to pick. In this case there only needs to be 4 points in category 1 for every 100 in category 0 for that segment to be classified as a 1. In the case where a road segment is perfectly smooth except for a few stray points of category 2, there would need to be at least 8 category 2 points for every 100 in category 0 for the category 0 cluster to be removed. This is because it is a far bigger claim to say a smooth road is actually very bad than it is for a good road to be classified as medium. This algorithm can be seen in Listing 8.

```
higher_center      = higher_cluster['center']
higher_cluster_cat = higher_cluster['category']
higher_point_count = len(higher_cluster['points'])

for lower_cluster in lower_clusters:
    lower_center = lower_cluster['center']
    if distance(lower_center, higher_center) < bandwidth:
        lower_point_count = len(cluster['points'])

        ratio        = higher_point_count/lower_point_count
        category_dif = higher_cluster_cat - lower_cluster['category']
        ratio       /= category_dif

        if ratio < 0.04:
            # remove higher point

# if higher point is not removed
#     remove all lower points near it
```

Listing 8: Comparing higher cluster with all lower clusters

If the higher point is not removed then that point is dominant and every point within the bandwidth distance is removed. This process is repeated comparing category 1 with 0.

After all overlapping clusters have been removed the 3 categories can be safely combined and inserted into the MongoDB database.

Generate Heat Map Data

The final step in the clustering process is to generate the data to be used by the heat mapping library. This requires making a JSON file that takes the form of a two dimensional array, where the inner arrays store the latitude, longitude and value at that point, this can be seen in Listing 9.

All clusters are queried from the cluster collection, the location of the center of the cluster is used for the point on the map. The value is the mean value of each point in the cluster.

This data is then written to 3 separate JSON files, one for each category. Using 3 separate files makes it easy for the UI to toggle the 3 categories on or off independently.

```
[
    [53.33279273, -6.26052540, 0.6597],
    ...
    [53.38633324, -6.26083575, 0.4717]
]
```

Listing 9: Heat Map JSON File

Clustering Snapped and Raw data

This whole process is repeated for both coordinates snapped to a road and for the raw GPS points. Some cycle tracks do not register as roads making them not recognised or incorrectly merging them with a near road. This can be seen when comparing Figure 3.20 and 3.21, in this example it is clear from the raw data that the cycle track is of good quality, while the road is bad. Using only the snapped data the clusters from the cycle track get grouped with that of the road resulting in a false model of that area.

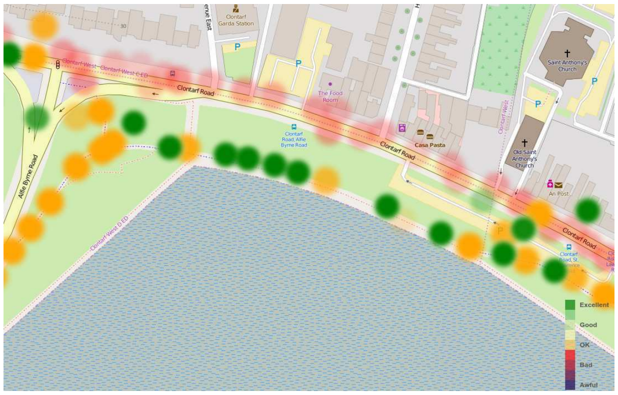


Figure 3.20: Mapping Raw Cluster data

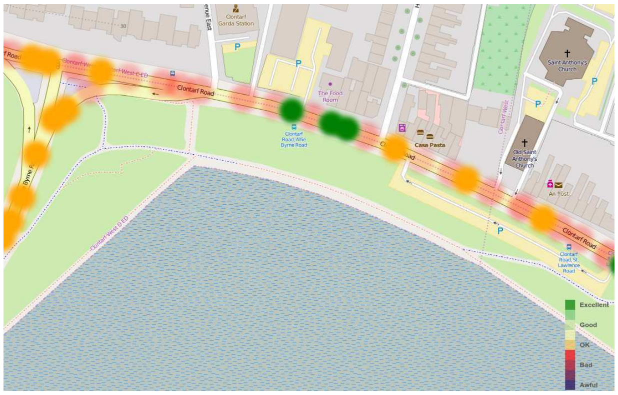


Figure 3.21: Mapping Snapped Cluster data

3.8.5 Ranking Roads

Assigning a road section a score plays a number of roles in this project. For integration with Smart-GH, road ratings are read from a Redis database and those values are used for routing purposes. To be able to see a table of roads with a clear score assigned to them is a useful tool for a cyclist to have. Road ratings were used to conduct a survey where participants were given the roads rated in the top and bottom 25 percent in this system for their area. This was used to see how this systems ratings compare with the public opinion.

The quality of a road is quite a subjective measurement. Is a smooth road with a few huge potholes better than a slightly bumpy road with no potholes?

Rating by Mean Acceleration

One approach taken was to find the mean acceleration felt, this meant on a smooth road, and potholes would play a small part in reducing the score of that road.

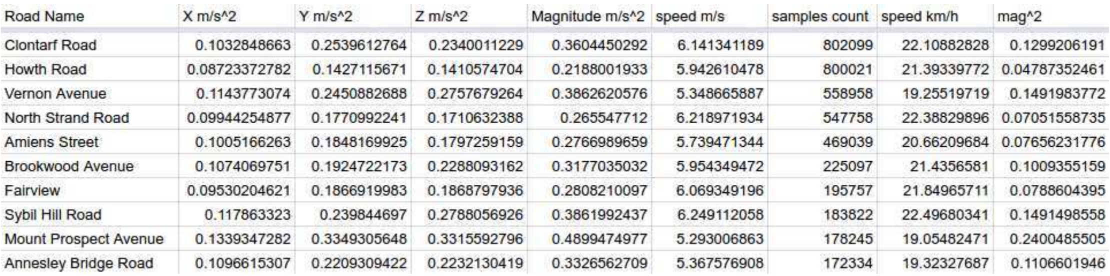
A database index placed on the road name meant all measurements for a given road could be quickly retrieved. Using MongoDBs distinct method, a unique list of all the road names is extracted.

```
road_names = db.measurements.distinct('location.address.street')
for road_name in road_names:
    measurements = db.measurements.find({
            'location.address.street': road_name,
            'section':{'$exists': True},           # is in a section
            'location.position':{'$exists': True}, # has been snapped
        })
```

Listing 10: Query to get unique road name

For each road a query is made to find all the measurements on that road, limiting to only return results that are in a section and that have a position that has been snapped to a road to reduce the chance of incorrect attribution between roads. If the number of measurements on that road was found to be under a threshold value, then that road was omitted. A value of 50 was experimentally chosen for the minimum number of measurements on a road.

The mean value of the magnitude of each axis for each road is calculated and written to a CSV file and to a Redis database. The CSV files were displayed in a HTML table for easy viewing. Figure 3.22 shows an extract of the table of the average values for each road.



Road Name	X m/s^2	Y m/s^2	Z m/s^2	Magnitude m/s^2	speed m/s	samples count	speed km/h	mag^2
Clontarf Road	0.1032848663	0.2539612764	0.2340011229	0.3604450292	6.141341189	802099	22.10882828	0.1299206191
Howth Road	0.08723372782	0.1427115671	0.1410574704	0.2188001933	5.942610478	800021	21.39339772	0.04787352461
Vernon Avenue	0.1143773074	0.2450882688	0.2757679264	0.3862620576	5.348665887	558958	19.25519719	0.1491983772
North Strand Road	0.09944254877	0.1770992241	0.1710632388	0.265547712	6.218971934	547758	22.38829896	0.07051558735
Amiens Street	0.1005166263	0.1848169925	0.1797259159	0.2766989659	5.739471344	469039	20.66209684	0.07656231776
Brookwood Avenue	0.1074069751	0.1924722173	0.2288093162	0.3177035032	5.954349472	225097	21.4356581	0.1009355159
Fairview	0.09530204621	0.1866919983	0.1868797936	0.2808210097	6.069349196	195757	21.84965711	0.0788604395
Sybil Hill Road	0.117863323	0.239844697	0.2788056926	0.3861992437	6.249112058	183822	22.49680341	0.1491498558
Mount Prospect Avenue	0.1339347282	0.3349305648	0.3315592796	0.4899474977	5.293006863	178245	19.05482471	0.2400485505
Annesley Bridge Road	0.1096615307	0.2209309422	0.2232130419	0.3326562709	5.367576908	172334	19.32327687	0.1106601946

Figure 3.22: Table of the average value for each road

The information shown here includes the average value for each axis, the magnitude of the 3 axes, the speed in both meters per second and kilometres per hour, the number of measurements on that road and the magnitude squared to help separate roads that may appear to have similar values.

Rating by Cluster Value

A second approach was taken to order roads by average cluster value. To achieve this all clusters on a road were summed and the average value was computed. The goal of using this method was to compute a more fair rating. If with the previous method, a short bad section was popular among the participants then whole road average would be very low. Where as using a clustering approach meant that these popular short sections would only be assigned single values at the same rate as other sections. For example, at a popular crossroads the users may be on a second road for just a few meters but this could represent thousands of points, these would play a large part in the average of the second road. If this section was unrepresentative of the second road then it would cause the roads overall score to not be its true score. Using a clustering approach would mean this crossroads would only get one or two clusters causing only a slight change to its overall score.

This approach operated very similarly to the previous method with a difference in querying the Clusters collection to retrieve all clusters on that road. Each cluster had a value computed by calculating the mean value of each point in the cluster. The number of clusters and total number of points in each cluster were summed, these values are used to filter out any roads that have the same amount of clusters as total points. This removes roads that only have clusters with one point. These are normally roads that have only been ridden once. The road was then given a score by getting the average cluster value for each cluster on a road.

3.9 User Interface and Visualising Data

A heat map was used to visualise this data, this map can be viewed as a stand along heat map or overlaid on top of Smart-GH. A number of custom maps using colored markers were generated. A personalised survey was generated to compare the results of this system with the opinion of the public.

3.9.1 Heat Map

To visualise this data the Leaflet heat map plugin was used [14]. This is an open source library that uses OpenStreetMap. Using the JSON data generated from the clustering process a heat map were created and deployed to same Heroku server hosting the uploading service.

Using JavaScript, HTML and CSS a legend depicting the color gradient was add. The ability to toggle on and off specific categories was also added.

3.9.2 Smart-GH

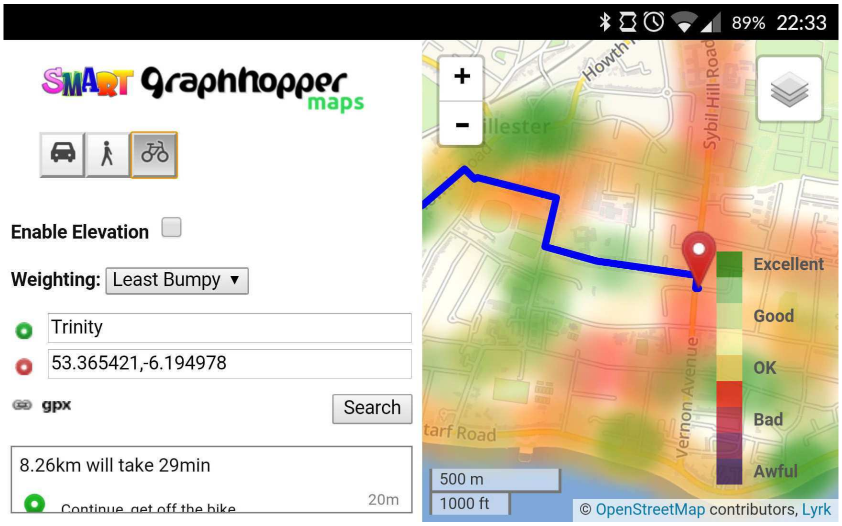
GraphHopper [12] is an open source routing library and server using OpenStreetMap written in Java that runs on Apache Tomcat. Smart-GH is a fork of GraphHopper developed in the

Future Cities Research Centre in Trinity College Dublin that provides routing by an arbitrary data source.

The integration process involved locating and downloading the OSM file for the greater Dublin area as the map provided was limited to just the city center. A new routing option of "Least Bumpy" was added, a weighting class was created to implement the weighting interface. When GraphHopper is calculating the value of a route, a method in this least bumpy class is called to calculate the average surface value for a given road, this method queries the Redis database and returns the value of that road. If no road is found, a default value is used. This default value is assigned the value of the bottom 20th percentile. The reasoning behind this is that if there is multiple routes close to each other and one has never been taken, then it is likely that this road has never been cycled on for a reason and therefore should only be taken if the surrounding routes are particularly bad.

Overlaying the heat map required adding the code used in the previous section to the UI packaged with GraphHopper. As OpenStreetMap is also used by GraphHopper the integration was quick to achieve.

As seen in Figure 3.23 and Figure 3.24 these maps are easily viewable on mobile devices.



Enable Elevation ☐

Weighting: Least Bumpy ▼

○ Trinity

○ 53.365421,-6.194978

gpx Search

8.26km will take 29min

Continue, get off the bike 20m

Figure 3.23: Mobile Smart-GH

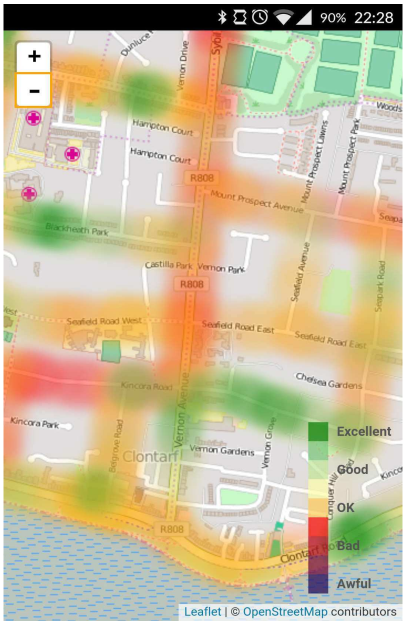


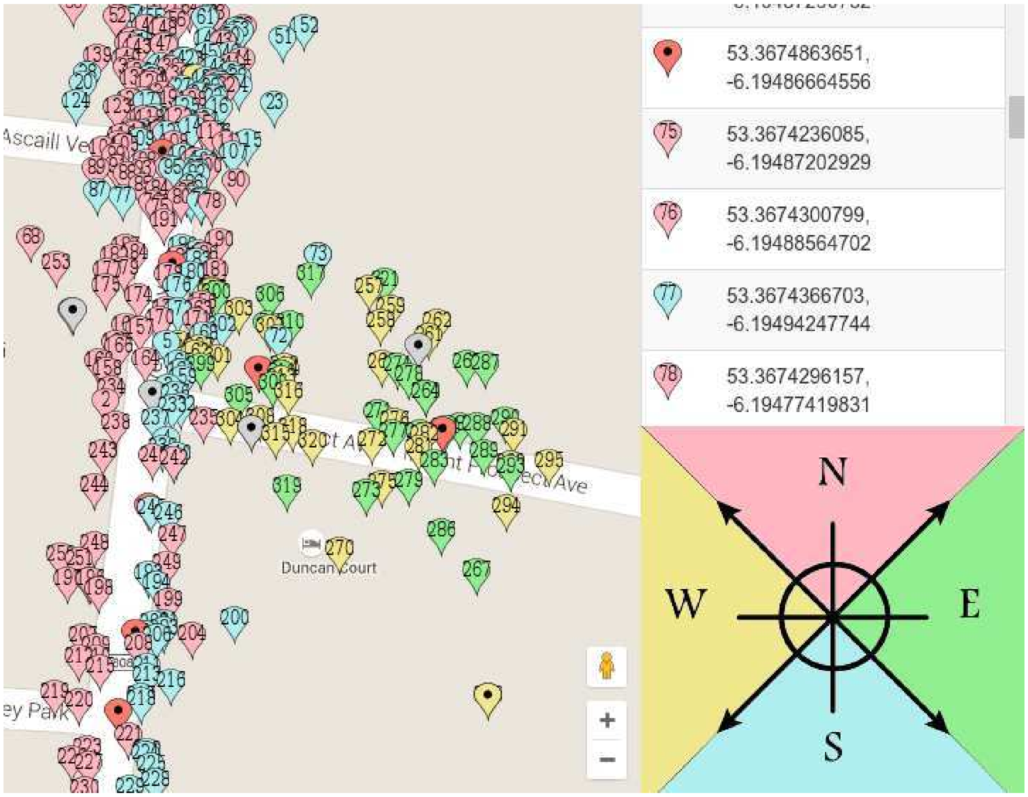
Figure 3.24: Mobile HeatMap

GitHub was used to help with the integration, this allowed for this to be worked on independently without the worry of damaging their code. GitHub was used for this entire project for version control, this made the integration process smoother.

3.9.3 Custom Maps with Markers

Custom maps were generated using Google Maps and colored Markers. These maps served the purpose of providing additional detailed information over the clustered heat maps.

As seen in the previous maps in Figure 3.18 and Figure 3.18, the detail provided by these markers allowed for a deeper understanding of the data.



	53.3674863651, -6.19486664556
75	53.3674236085, -6.19487202929
76	53.3674300799, -6.19488564702
77	53.3674366703, -6.19494247744
78	53.3674296157, -6.19477419831

Figure 3.25: Colored Markers Denoting Direction

The use of color markers in Figure 3.25 meant the bearing of the point can easily be seen, the color of the center marker with a dot denotes the category of the bump where categories 0,1 and 2 were represented with tan, grey and red respectively. The direction of individual point can be seen by the color of the marker where green is east, yellow is west, blue is south and north is pink.

3.9.4 Auto Generated Survey

Surveys were generated to compare the results from this system with the opinion of the users. For a given users phone ID, a distinct list of all road they had travelled on was queried. The roads that were ranked in the top and bottom 25% were found. For each of these roads the number of measurements recorded for this user was counted, if this value was above a threshold, the road was appended to a list. This threshold was to ensure that user has spent a considerable amount of time on this road. This list of roads was then randomly shuffled and HTML was generated for each road with its title and a link to this road on Google Maps. The generated HTML was then fed into SurveryMonkey using their "Add Answers in Bulk" tool [39]. A unique link was then generated for each user and sent to them. This survey can be seen in Figure F.1.

Chapter 4

Results and Evaluation

In this chapter we will evaluate this project and discuss results obtained. Possible improvements and changes will be investigated with their advantages and disadvantages compared. This evaluation will consist of:

- Hardware
  - Did the electronic hardware and enclosure fulfil the requirements
  - What improvements could be made?
- Android Application
  - What worked well and what did not
- Improvements gained from 3 stage reverse geocoding
- Correlation between acceleration sensed and speed travelled
- Rating roads using clustering and mean approach
- How this system compared to participants opinion

4.1 Electronic Hardware

The objective of the hardware was to record accelerometer data, transmit the data over BLE to the user's phone and to flash an LED to display notifications. The MetaWear achieved all of these but had one downfall, the battery life. The MetaWear came packaged with a 100 mAh battery, this provided approximately a week of battery life, with regular use lowering this. As the MetaWear was always on, even if the user did not use their device for a week or two, it would be out of battery.

This poor battery life resulted in a large amount of journeys being wasted due to the MetaWear not having power, this can be seen in Figure 4.1.

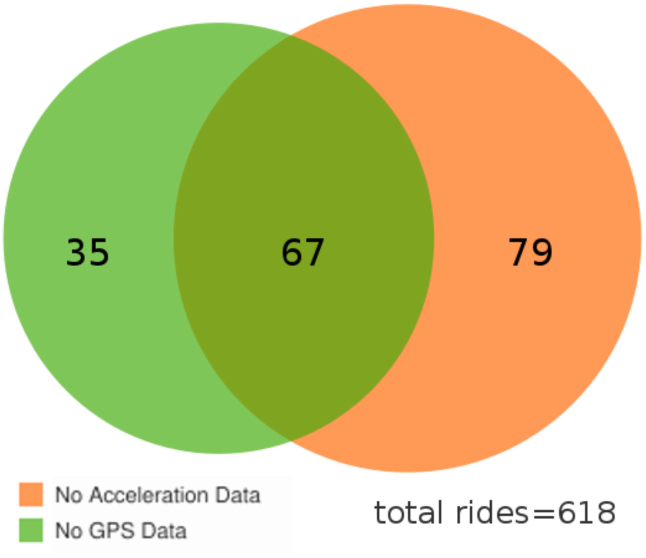


Figure 4.1: Number of Rides with Incomplete Data

Out of a total of 618 rides recorded, it was found that 181 had incomplete data, the majority of these has no acceleration data. This can be partly attributed to the battery having no power. This could be also caused by a ride being so short that there was not enough time for the MetaWear to connect. The connection time was seen to range from a few seconds to a one or two minutes.

The large overlap between rides with no GPS and those with no accelerometer data would support the idea that these were short rides or accidental ones that were started and instantly stopped without being deleted. This leaves at least 79 rides that were lost due to no acceleration data.

If I were to repeat this project I would have invested in a larger battery. Researching online with the same supplier that provided the batteries for the MetaWear showed other batteries with a capacity 5 times as large for an extra $5 [27]. While the bigger battery would mean the overall volume of the device would be increased, it would only bring a 1.5mm increase in the thickness. This increase would result in a size close to that of the prototype boards which were initially used. Larger enclosures were created for these MetaWears and it was found that they attach to a bicycle equally well and discretely.

Even though the number of rides with missing data is large, there was still a large volume of rides which had good data making up for those missed.

4.2 Enclosure

The purpose of the 3D printed case was to protect the device and attach it in a fixed position on the bicycle. The devices were used in all weather conditions for the 5 month period, no cases broke or became damaged and neither did any of the MetaWears. The enclosures had a 100% success rate serving their purpose.

4.3 Android Application

When searching for participants, a number of people could not partake due to the phone they had not being supported. The majority of those had Apple phones with two having old Android phones that did not support BLE. While this made it more difficult to get participants, it was quickly overcome by finding different volunteers.

A focus group was organised to receive some qualitative feedback about the Android application. Participants felt it was easy and intuitive to use but all of them agreed that more could have been done to alert them when the battery level was low or fully discharged. In terms of letting the user know when the battery level was low, the application displayed the battery percentage in a small font on the bottom of the screen during the ride. Participants reported that they did not pay attention to this value, this was partially due to it only being displayed a few seconds into the ride after the connection had been made, at which point the phone was back in the user's pocket. It was reported that the most common way participants knew the battery had no power was to notice that "it hadn't flashed for a few days". On inspecting the raw data it was seen that for some users these "few days" were at times longer than a week. It is evident that this was a large contributing factor in data seen in Figure 4.1.

Some proposed solutions to this problem were to alert the user at the beginning of a ride if it had not connected the previous time, or to alert the user after a number of failed connection attempts, the application used in this project kept trying to connect indefinitely until stopped, or to alert the user at the end of a ride if the battery is running low. It can be argued that if these precautions were in place from the start there would be more data collected.

It was also suggested that rides should be automatically uploaded and that there could be a setting to turn this feature on or off meaning the user was still in control of their data. This feature was considered during the course of this project but was dismissed as it was felt that trying to push to many updates to users would be annoying and could result in different participants running different versions of the application. It was also felt that since this method worked well and the developing time could be more effectively used elsewhere in the project.

The inclusion of NFC was seen to make the application easy to use especially during the winter when participants wore gloves which do not work for touch screens.

Overall participants reported that they "got into a habit of using it" when cycling. This shows how the application can be quickly become a regular part of a cyclists commute.

4.4 Reverse Geocoding Efficiently

Three methods were used to look up a GPS coordinate. The use of Google Maps Geocoding API meant that a very detailed addresses were found for GPS points. A sample of this precision can be seen in the Listing 12 where details on the building number is included.

As discussed previously relying only on this API would be expensive in terms of both time and money. A hybrid system was created which provided far quicker look ups and did not rely heavily on Google.

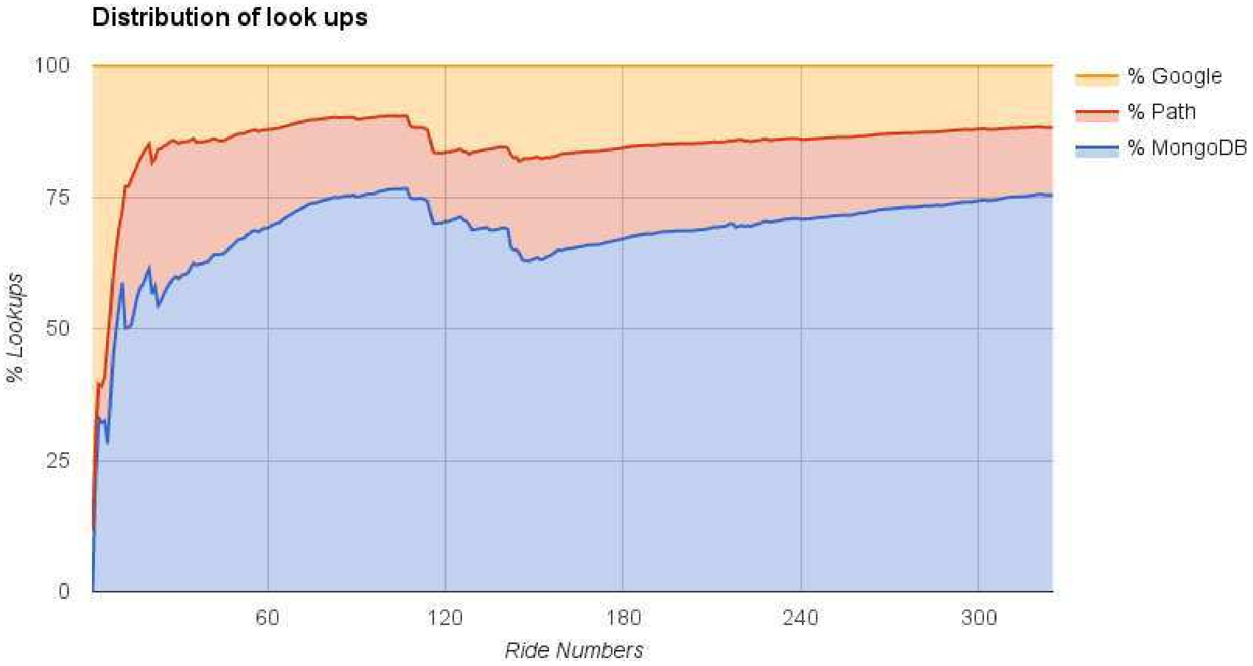


Figure 4.2: Comparison of Reverse Geocoding Methods

Initially all look ups were fulfilled by Google and the paths cache. This quickly became overtaken by database lookups. Figure 4.2 shows the running total percentage of how locations are reverse geocoded. It can be seen in this graph how the look ups in the path and on Google correlate with each as only true addresses retrieved from online can be fetched from the paths cache. Without the use of the path, there would be approximately twice as many look ups online. The occasional declining spikes in the look ups fulfilled by the MongoDB database were caused by either getting new participants or by users cycling to new locations never visited with the system. It can also be seen that as time goes on, these spikes become far smaller, even new routes have the majority of their journey already recorded by a different user. This shows how this approach improves upon itself, learning more locations as time goes on.

This approach of using 3 methods to look up a location meant that out of a total of over half a million locations reverse geocoded, only fifty thousand were looked up with the online

API with the rest being served from memory or disk.

4.5 Acceleration Adjustment

The acceleration of each measurement was adjusted to combat the problem of measurements recorded at different speeds.

A scatter plot consisting of 25 million points of speed vs Z-axis acceleration can be seen in Figure 4.3, where the blue line represents the line of best fit for the Z-axis.

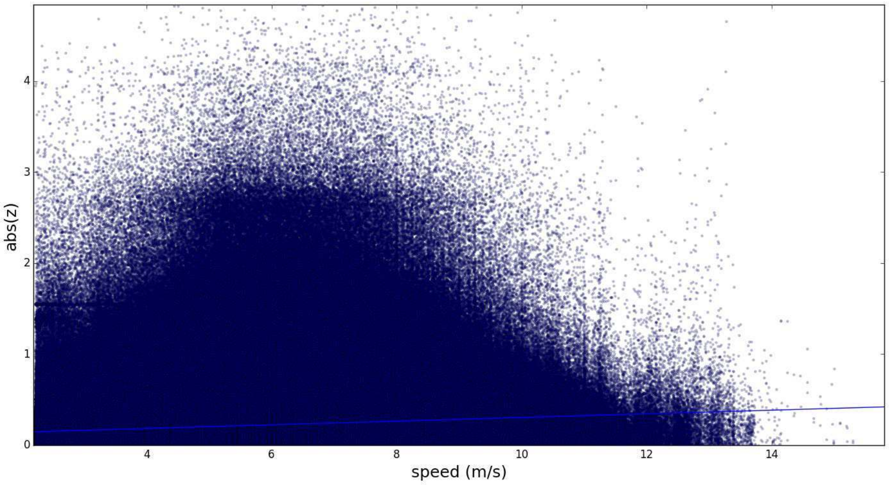


Figure 4.3: Scatter plot of Speed vs. Z-axis

While this line appears to be almost linear, rescaling the Y-axis of the graphs shows the extent to which acceleration increases with speed. This can be seen in Figure 4.4 where the computed lines of best fit can be seen for the X, Y and Z-axis in red, green and blue respectively.

On inspecting this graph we see that at a leisurely 10 km/h (2.78 m/s), which is 2 km/h above what we defined cycling to be earlier, the value for Z-axis acceleration is 0.156 m/s^2. At a faster, but still common pace of 36 km/h (10 m/s) we see the Z-axis acceleration has approximately doubled to 0.311 m/s^2. This shows how there can be a dramatic difference is the acceleration experienced based on the speed the cyclist is travelling.

The strong correlation between the Y and Z-axis can be seen Figure 4.4 where the two lines of best fit are almost identical. On inspecting the Table 4.1 it can be seen that the parameters for the Y and Z axes are very similar.

From this graph it would appear that speed has very little, if any, effect on the X-axis with an increase of only 28% from 0.089 to 0.1142 over the same speed difference used for the previous calculation with the Z-axis where a 100% increase was seen. It can be argued that

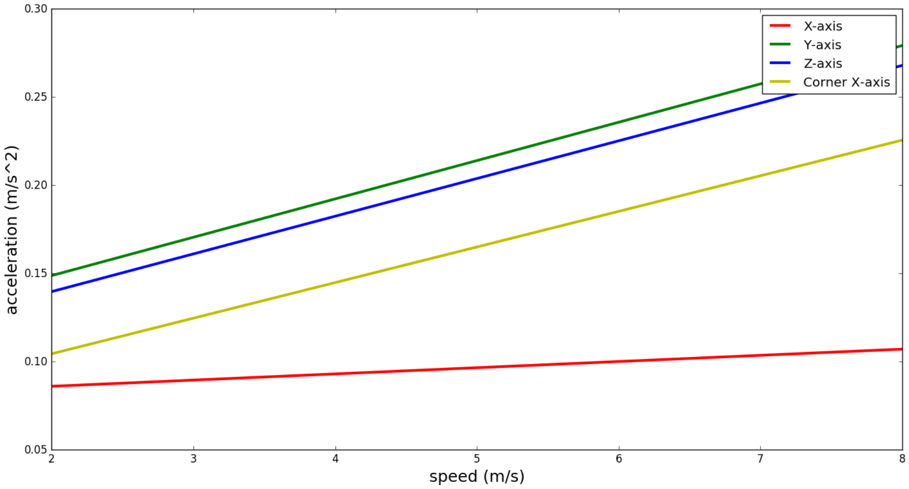


Figure 4.4: Speed vs. X,Y,Z-Axis & Corner X-axis

Axis	**m**	**c**
X	0.00351	0.079046
Y	0.02173	0.105364
Z	0.02137	0.096995
Corner X	0.02018	0.064113

Table 4.1: All measurements: Parameters for acceleration adjustment

this is due to cyclists slowing down when coming to a corner and cycling quickly on a straight sections. To test this hypothesis two known sharp turns were studied. All measurements within 7 meters of the corners were queried resulting in over two thousand readings. The line of best fit was computed and the results can be seen in yellow in Figure 4.4 and Table 4.1 where the slope can be seen to be very similar to that of the Y and Z-axis.

A linear curve was used to fit the relationship between speed and acceleration because it was observed that previous research had scaled their acceleration values linearly. It can also be seen in the equation for impulse imparted, $J = m * v$ where J is the impulse imparted, m is the mass and v is the velocity that there is a linear correlation between impulse and velocity.

The importance of adjusting the acceleration with relation to speed can be seen by looking at a sample fast paced road. The "Carrickbrack Road" in Howth is steep down the side of a hill where cyclists reach high speeds. The average speed recorded for this road in this project was 29.095 km/h, with a Z-axis acceleration of 0.185 m/s^2 placing it in position 194 out of 514, see Table F.2. Calculating the average acceleration before adjustment resulted in a value of 0.23812 m/s^2 which would place it in position 326, dropping it 25% down the list.

4.6 Road Ratings

Rating the different roads was an important process in this project. The ordered roads were used to produce a survey to give to participants, they were also used within Smart-GH to allow routing by the smoothest road. This survey can be seen in Figure F.1 with the results of these ratings can be seen in Table F.1 and Table F.2.

4.6.1 Comparing Methods

Comparing these two techniques for scoring roads it can be seen that the clustering approach results in fewer roads than just calculating the mean for that road. The clustering method ignores roads that are only briefly travelled on as there is not enough measurements to form clusters.

On evaluating these techniques it can be seen that the averaging approach without clusters included some roads which were never travelled on. These were roads were close to a road being travelled on but due to GPS error a few points were registered on the incorrect road. Using clusters improved upon this, removing incorrect roads which resulted in fewer roads being labelled.

A difficulty for both methods is roads at intersections perpendicular to the direction the cyclist is travelling, that are never travelled on other that at the intersection. These roads can get a score even though there was never a real journey on that road. The clustering approach can combat this by looking at the total number of clusters verses the number of points. For example looking at Preston Street we see 86 points but only 4 clusters, due to the low amount of clusters but high amount of points it can be suspected that this is just a road that a user passes through or passes close to. On inspection it is seen that this is a side road, perpendicular to a main road, which has never been travelled on and the values are due to GPS error in measurements on the main road. Using only the road averaging method this fact can not be seen.

The difficulty in defining what makes a good road makes it hard to label one method as categorically better. A big difference in these two methods is seen when part of a road is travelled on much more than another, averaging all points on the road weights the overall score based on popular sections while clustering gives a more fair score for the total road. It can be argued that if a section is more popular then that location deserves a higher weighting. For this project it was reasoned that using the clustering method gave more accurate and fairer results.

On comparing the best and worst rides for each method, it can be seen that they produce similar results. This correlation is seen to grow when you only consider routes with a large point count. Looking at Figure 4.5 we see a histogram with the difference in positions for each road between both methods. This histogram consists of 374 roads that were in common on both lists. It can be seen that the majority of roads are within 15 places of each other throughout both lists. However there is still a discrepancy between the two lists highlighting the differences between the to methods.

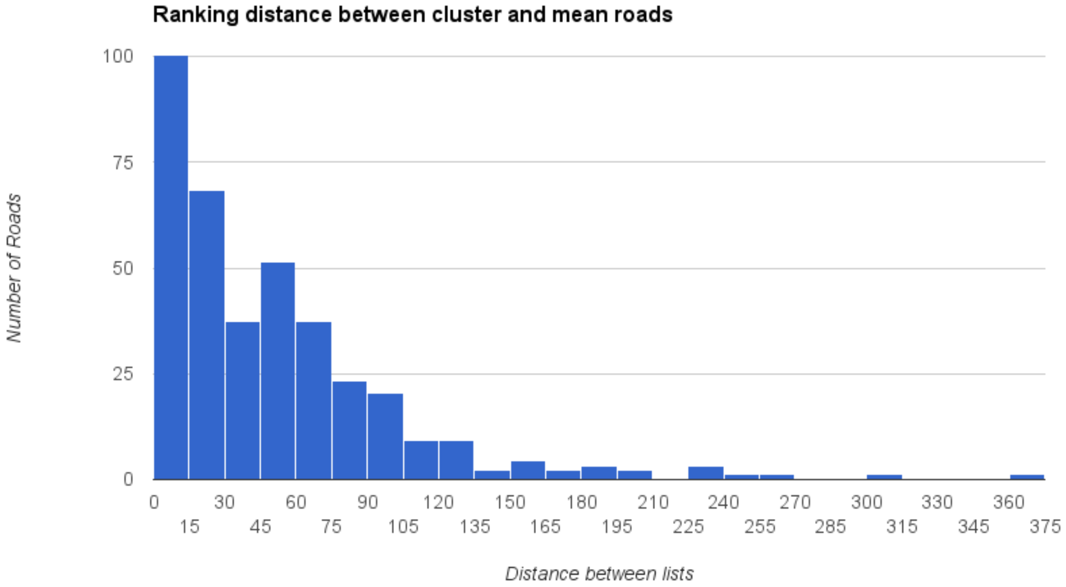


Figure 4.5: Difference between quality rankings for each road for each methods - bucket width = 15

One major difference can be seen with "Suffolk Street". By the clustering method this road is rated as the 6th best where as it is the 2nd worst by the averaging method. This discrepancy can be linked to the number of readings on that road. It is seen that there is less than 37 seconds of data for this, because of this and how few clusters are found the clustering approach ignores outlying single bad results. All bad readings for that road would be outliers as there is so few readings. One approach to combat this problem would have been to have enforced a strict minimum cluster size for all categories but this would have resulted in a number of road having no values associated with them due to the low number of participants used.

With more participants or if this was conducted over a longer time, then the size of these clusters would grow and increase the overall accuracy meaning we could safely enforce a strict threshold for cluster sizes.

4.6.2 Survey Results

Surveys were distributed to participants with the best and worst roads that they were seen to travel on. The surveys consisted of 20 roads on average depending on how many roads that user went on. Participants were encouraged to only rate roads which they were confident about and to ignore any that they could not remember if they were good or bad. Participants were given a week to complete the survey meaning they could read the roads on their survey, check on a map where they were and then pay particular attention to them during their commute. The results of these personalised surveys were aggregated and compared to their position within this system.

	Participant - Good	**Participant - Bad**
Smooth Route - Good	38	2
Smooth Route - Bad	3	24

Table 4.2: Survey Results

From this survey, 8 of the participants responded. Each participant gave a response for an average of 8 roads. The results can be seen in Table 4.2. From this table a 93% agreement can be seen between the ratings by the system and those by the participants. It can be argued that one of the three roads rated as good by the participants but bad by the system is actually a bad road. It was seen that two other participants rated this road as bad and on manual inspection it was seen that this was in fact a very poor road. This would bring the agreement to over 94% if that user had labelled it as bad.

4.6.3 Difficulties

A difficulty or flaw with the process of ordering roads for routing came from relying on road names. There is not a "one to one" mapping from a road name to a particular road, there is not even a "one to many" mapping. It was seen that there is a "many to many" mapping between road names and actual roads. This is seen with roads that have common names, for example the road "Church Road" appears 5 times throughout just Dublin. And with regional roads which are made up of multiple smaller sections of other roads. An example of this can be seen with R808 that includes parts of Vernon Avenue and Gracefield Road but all of Brookwood Avenue and Sybil Hill Road. Reverse geocoding coordinates along this path could result in either or both names being returned. This was attempted to be overcome by storing both the long name, e.g. Vernon Avenue and the short name, e.g. R808. But at times just the short name was returned and it was unknown which road it was a part of. Another issue with relying on road names comes when dealing with cycle tracks, most cycle tracks do not have a name resulting in the measurement either having no address or being incorrectly assigned to a nearby road.

The reliance on roads was seen to have a negative effect relating to how measurements along a cycle track were snapped to the road beside the cycle track this can be seen in the previous shown example in Figures 3.20 and 3.21.

It was difficult to decide on a cut off point for when roads are disregarded. It can be seen that roads with a low point count, result in unexpected results. An approach could have been taken to omit roads with a point count or ratio between point count and cluster count below a threshold. For the purposes of this report it was chosen for this threshold to be low to allow for unpopular roads to be seen for completion and referencing purposes.

4.6.4 Improvements

A different approach was proposed towards to end of this project, this was to use OSM Node IDs [22]. These are unique IDs relating to points along a route. These routes included community defined routes, like cycle tracks and paths through parks that do not have to have a road name. Using OSM IDs would also mean routing could be more accurate as it could calculate the best route based off just the Node IDs which are part of a given route. This method was only discovered after implementing the approach which relies on road names and the associated problems were discovered. At this point it was too late to change the architecture of the system. Using OSM Node IDs and OSM Routes would help to combat the errors that arose when snapping a unknown cycle track to a road. As OSM routes are community generated and do not rely on the route having a street name, there are a lot more smaller paths and cycle tracks included.

4.7 Cluster

As discussed, clusters serve a number of uses in this project. Within the field of visualising location data, a major problem is how to view the data in a meaningful way. A big barrier to this is the aggregation of data. Exposing a JavaScript engine to the gigabytes of map data would not be a possible solution. Naive aggregation approaches still leave more than one value for a given location. This was a major problem with the heat map library used, if an area had more than one reading then that area's value would be added to form a single value. This initially resulted in popular routes appearing as far worse than rarely travelled roads. Clustering solved this and ensured popular areas do not overshadow all other areas.

4.8 User Interface

The user interface was developed for users to browse the data freely and to allow them to route by a given parameter. The interactive heat map can be viewed in Appendix D. This screenshot in Figure D.1 shows the volume of the data that was collected by just 10 casual cyclists. This shows how with only a few users, a large area can be covered quickly.

The results of the routing with Smart-GH can be seen in Figure D.2. Here the blue line on the top diagram represents the quickest path taking 22 minutes where as the blue line on the bottom image represents the least bumpy route avoiding some particularly bad areas but taking 3 minutes longer. It can be seen in this Figure how users can toggle different categories on and off. This makes it possible to easily locate specific potholes and bad areas.

4.9 Case Study

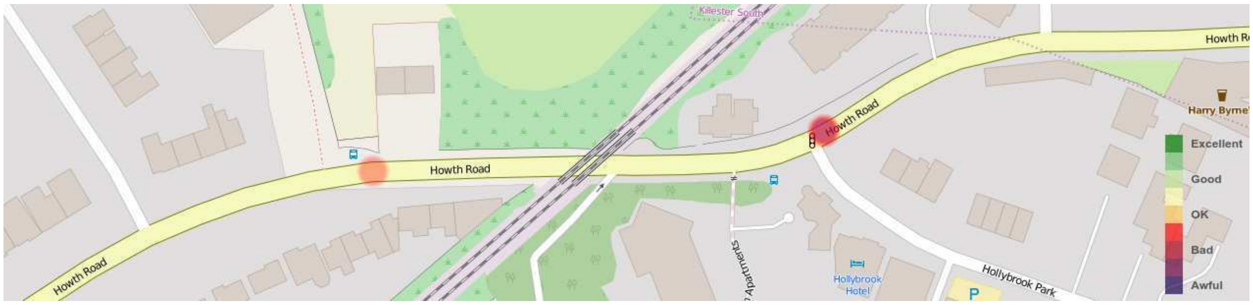


Figure 4.6: Howth Road Heat Map - Category 2

The Howth Road was used to perform a case study. This road was chosen as it was observed that there were very few category 2 potholes with one particularly big one that we can study.

Inspecting the heat map with only category 2 clusters visible it can be seen that there are only 2 sections with category 2 clusters, these can be seen in Figure 4.6. The darker feature on the right was chosen to study.

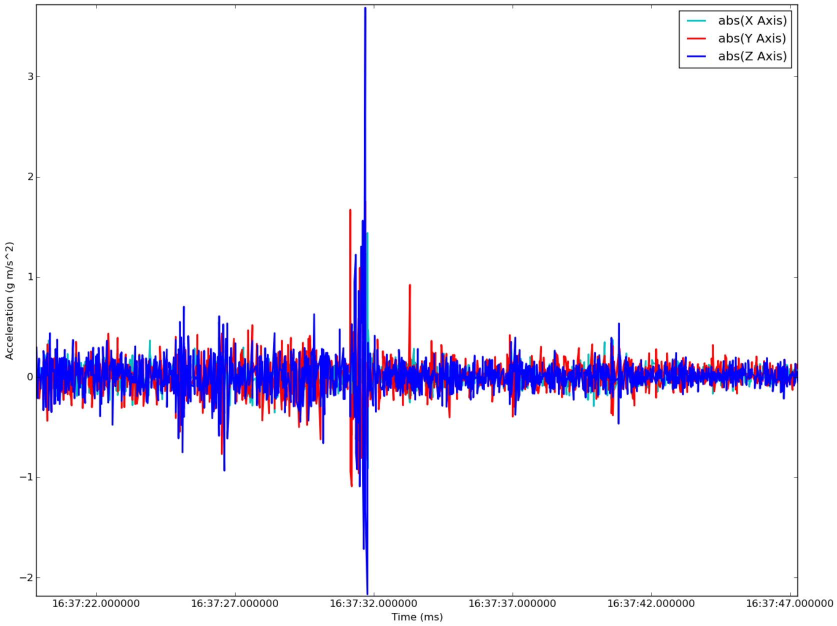


Figure 4.7: Howth Road Acceleration Data

Looking at the calibrated data for that section of Howth Road for a single journey, this pothole can be clearly seen in Figure 4.7. This sample was taken by querying a ride at

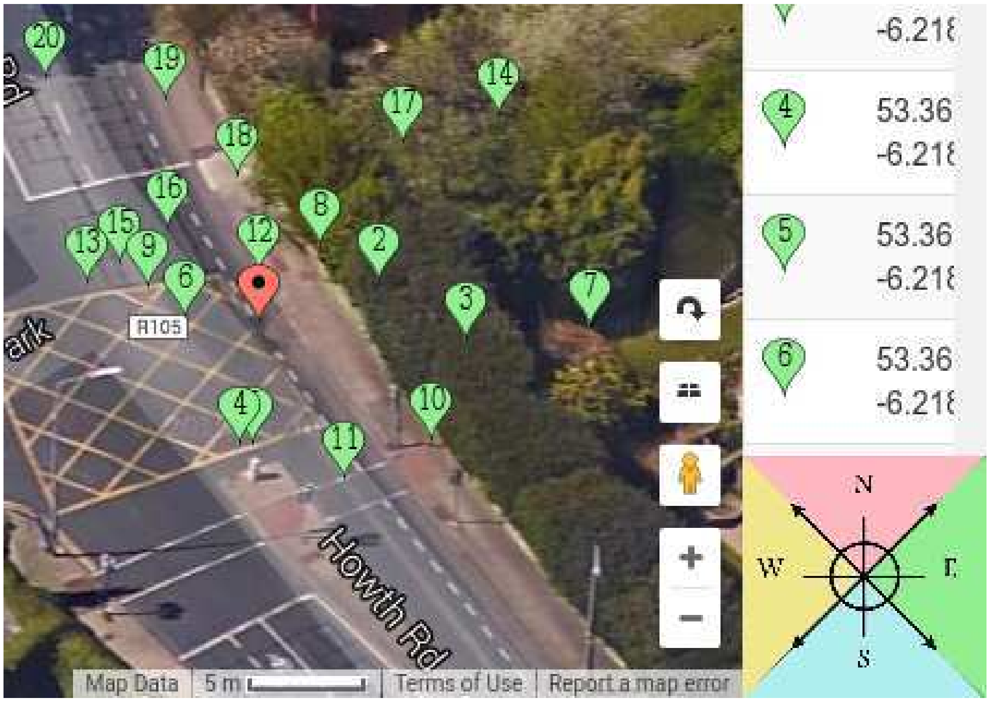


Figure 4.8: Howth Road Cluster Satellite Image

random that travelled along Howth Road. All measurements on that ride within a 350 meter radius of the target area were queried and plotted. It can be seen how there are no other spikes of a similar size near, with the next largest peak being approximately 4 times smaller.

Using the custom maps created, all the points that make up that cluster can be seen along with that clusters center. This is seen in Figure 4.8. Looking at this image we can see how all the points making up that cluster are green meaning they are travelling east and are not from the other side of the road. This site was manually inspected and a photograph of the feature was taken, this is seen in Figure 4.9. This photo is a good example of an unavoidable pothole, due to the traffic island ahead, regular traffic and size of the hole it is almost always unsafe to avoid it. It is not apparent from the photo that this pothole fluctuates massively and sharply on the smooth looking center, this deception makes it even more dangerous.

On inspecting the location of the center of the cluster, it can be seen that it is in the cycle lane at the corner of the yellow box. On comparing this with the actual photograph we see how the center of the cluster is within a few centimetres of the actual location even though each individual point had an error placing it a few meters from the actual point. This shows how when a sample size grows, so does the statical accuracy converging towards the actual location. This reinforces the notion that as the number of users grows the accuracy of the system does too.

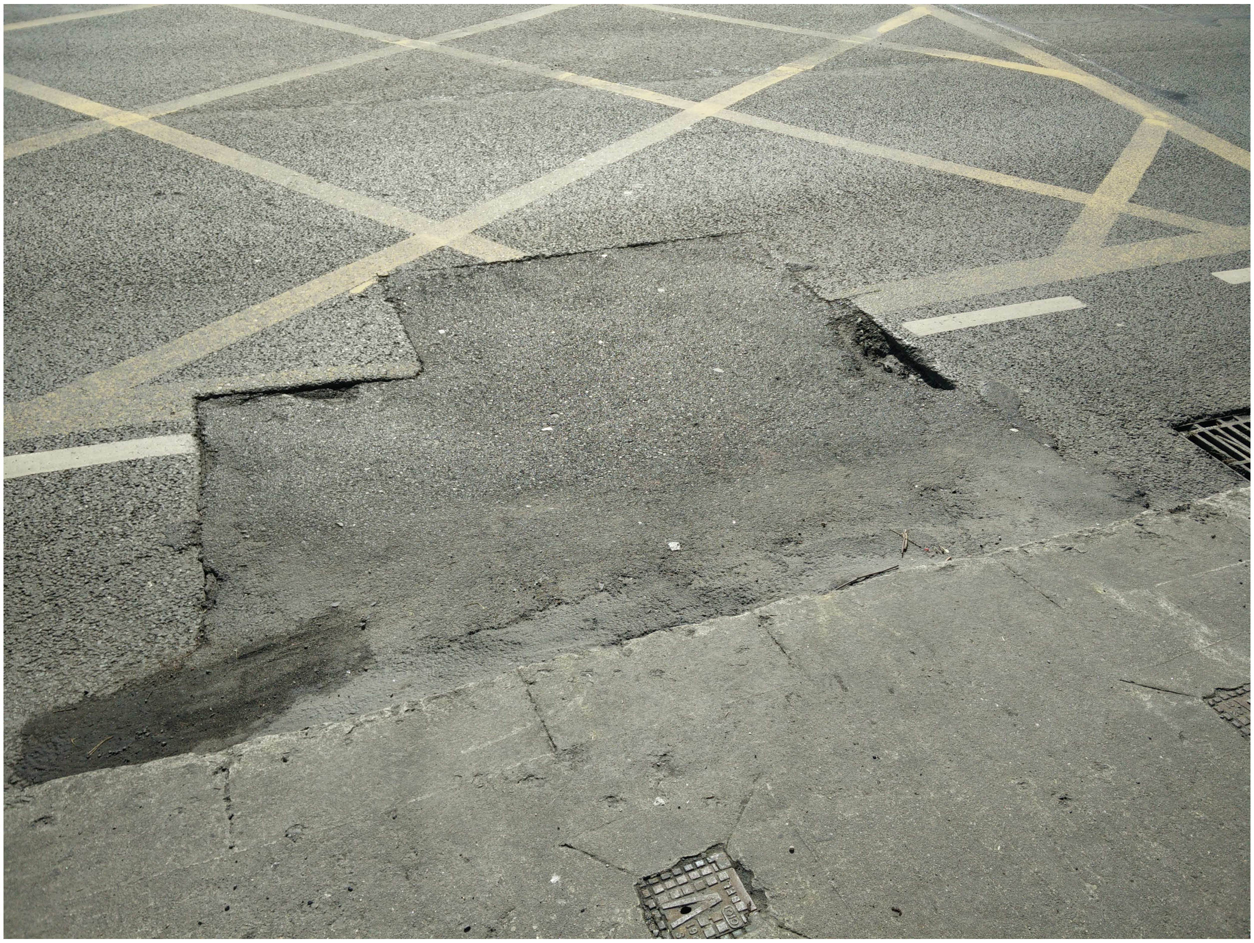


Figure 4.9: Howth Road Photograph

Chapter 5

Conclusion

When comparing this project with similar projects from the research section it can be seen how this excelled in some areas more than others. The routing, automatic calibration, road rating, reverse geocoding and visualisation of the data proved very effective. This project focused on retrieving accurate clean data. It is felt that some other projects did more in terms of classifying features in more detail than just 3 categories. This approach was taken as a cyclist does not care if it is a pothole, sunken manhole, expansion joint or a bad speed bump, the cyclist just cares about the smoothest and safest route.

5.1 Overview

The aim of this project was to provide a cheap and crowd sensed approach to survey the quality of a road. It has been demonstrated that dedicated sensors can work in tandem with with a smartphone to sense road quality and provide routing capabilities based on this data.

5.2 Contribution

This project introduces novel approaches to the problems encountered while implementing this system.

5.2.1 Dedicated Sensing and Smartphone Hybrid

A unique approach to the problem of crowd sensing was chosen. Using cheap sensors, 3D printed cases and a smartphone provides the best of both systems and can be seen to eliminate many of the problems associate with similar projects that relied solely on on device.

5.2.2 Auto-Axial Calibration

Using the idea of sections, aligning mean acceleration with gravity and correlating acceleration along the positive Z and Y axis, where the positive Z-axis points up along the line of gravity and the positive Y-axis is in along the path of motion pointing backwards. The advantages of this technique can be seen in the accuracy of the calibrated measurements. The autonomous nature of this approach means it requires no input from the user making it easy to use and reducing any human error associate with manual calibration.

The use of sections would mean that if a system existed where the device periodically rotates and was not always in a fixed position then this system could easily deal with that by calibrating each section individually. This approach is not seen in any of the papers researched and has to ability to add value to projects of a similar nature.

5.2.3 3 Step Reverse Geocoding

To combat the problem of geocoding coordinates a 3 step technique was developed. The ability to check the path in RAM for a given ride and a geo-indexed database of all known location offers a huge advantage by reducing the reliance on a third party and offers exponential speed increases.

5.2.4 Location Interpolation

While location interpolation is not a novel idea, the method used in this project incorporated unique aspects. These were interpolating a snapped and raw location, speed, altitude and computing a bearing for that point. The best address from the two closest points to that measurement was assigned while having a limit to ensure accuracy. This interpolation pipeline was not seen in related work with only The Pothole Patrol reporting that just the location was interpolated. It was noticed that many related works would either not mention how location was linked to measurements or state that they just took the previously recorded location.

5.2.5 Acceleration Adjustment

The importance of scaling acceleration with relation to speed was seen in this project. It can be seen that related projects which deal with speed and acceleration often omit this relationship. Without this aspect good roads where cyclist tend to travel quickly were punished and slow bad roads were rewarded. This project proposed an approach using sections to limit only cycling paced measurement where a linear relationship between acceleration and speed was found through linear regression. This project proposed an argument for why speed was not seen to affect horizontal acceleration along the X-axis by studying only sharp corners and displaying how the data would suggest that there is a strong correlation between X-axis acceleration and speed when it is limited to areas where cyclists turn.

5.2.6 Category Based Clustering

This project used an adapted mean shift algorithm with an original approach to cluster data into different categories based of calculated peaks. This technique allows for the increase in accuracy in locating an individual feature, removing noisy outlying readings, fading old data out and provides the ability to easily visualise data by giving a single value for a given location. The fading of old data also helps with ensuring this system operates quickly and efficiently.

5.2.7 Integration with Smart-GH

Integrating with Smart-GH provided a large geolocated data set by which routing was achieved. On inspecting related work, each project just provides the data for the user to view without any attempt to route using that data. The flaws with relying on roads in this project have been discussed with the suggestion that using OSM Node IDs would provide better results. Future projects attempting to route journeys based on location data can learn of the challenges faced with this project in the hope of improving theirs.

5.3 Future Work

The value of this system comes from a large amount of user's spread over different areas with routes being covered multiple times. To achieve a wide geographical coverage with a large user base, the best approach would be to integrate Smooth Route with an existing commercial smart product.

With this only requiring updated software to become integrated with the smart device, it could be rapidly be introduced to a wide audience. These smart devices include: BeeLine [4] navigation device, Haize [24] navigation and notification device, PowerPod [26] power meter, Lumos [16] lit up helmet, SeeSense ICON [31] bike light, Noke [23] U-lock and the Garmin Edge Explore [9] bike computer. These devices are just a small sample of the potential devices that could integrate with Smooth Route.

Another approach to increase the user base for this project would be open source it. Providing the hardware specifications, 3D model files for the case, the Android application and documentation for the use of the HTTP endpoint would mean anyone could build their own and integrate it with this project. Having the HTTP endpoint documented and the application open sourced would mean other developers could integrate with already existing 3D party smart devices removing the need to construct their own hardware.

There are many uses for the data collected within this project. One future use would be to put more energy into classifying features. As discussed this was not a priority for this project as its aim was to find the smoothest route without worrying about what the bumps were. A more detailed classification system would mean the city council could easily see areas that need repair.

Another future use for this data that has yet to be explored would be to use the detailed acceleration maps produced by this system to aid in geolocating a device. Applying particle filters in a similar way to that described by the authors of Learning and inferring transportation routines [15] could mean detailed geolocating in a urban environment could be achieved be by matching areas with an acceleration profile similar to that experienced by the cyclist.

This project displayed the importance of adjusting acceleration by speed. It was seen how this relationship along the X-axis was dependant on the location of the cyclist, if they were travelling straight or taking a sharp corner. An interesting idea to be investigated further would be the idea of location based adjustment where specific features such as corners or speed bumps were scaled differently.

The type of bicycle was recorded for each reading, this information was never used as all the participants had very similar road bikes. It would be interesting to see how different bikes from racers to mountain bikes experience the same potholes.

This project mainly focused on acceleration along the Y and Z-axis, it would be interesting to further study X-axis acceleration, to find particularly sharp turns or if there are areas where users are regularly swerving.

The success and high levels of accuracy achieved in this project provides good foundations for future work to be built on. The most favourable future path would be to integrate with a commercial smart device, gaining a vast user base in a short period of time.

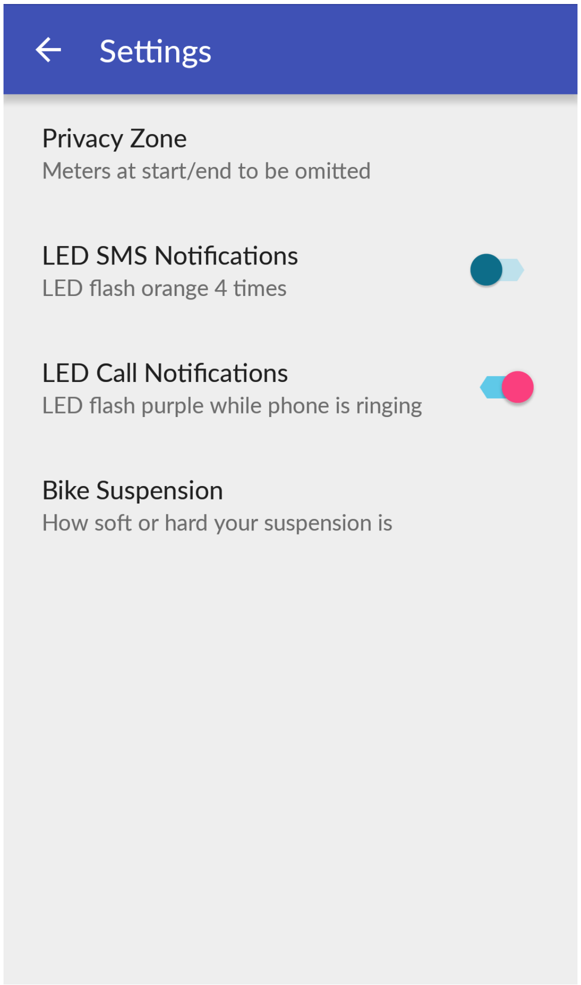
Appendix A

Abbreviations

BLE	**B**luetooth **L**ow **E**nergy
CSV	**C**omma **S**eparated **V**alues
JSON	**J**avaScript **O**bject **N**otation
XML	E**x**tensible **M**arkup **L**anguage
mAh	**m**illi**A**mpere **h**our
OSM	**O**pen**S**treet**M**ap
HTTP	**H**yper**T**ext **T**ransfer **P**rotocol
HTML	**H**yper**T**ext **M**arkup **L**anguage
CSS	**C**ascading **S**tyle **S**heets

Appendix B

Android App



Settings

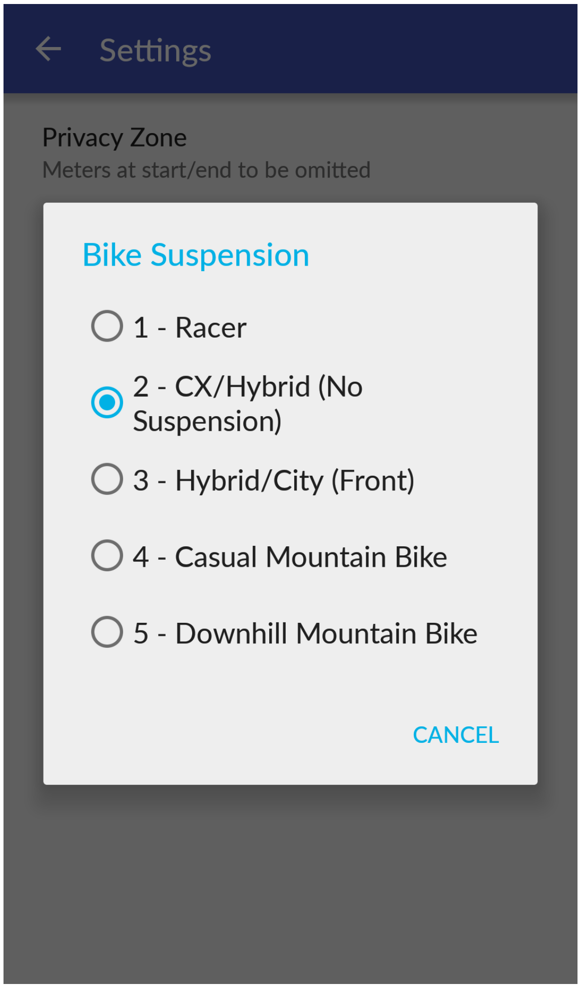
Privacy Zone
Meters at start/end to be omitted

LED SMS Notifications ○
LED flash orange 4 times

LED Call Notifications ●
LED flash purple while phone is ringing

Bike Suspension
How soft or hard your suspension is

Figure B.1: Settings page



Settings

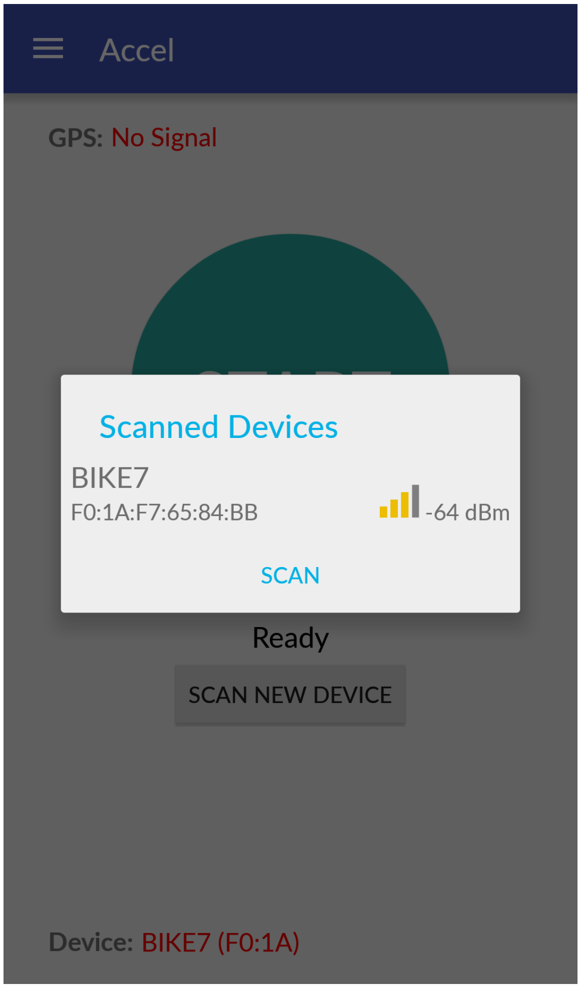
Privacy Zone
Meters at start/end to be omitted

Bike Suspension

○ 1 - Racer
● 2 - CX/Hybrid (No Suspension)
○ 3 - Hybrid/City (Front)
○ 4 - Casual Mountain Bike
○ 5 - Downhill Mountain Bike

CANCEL

Figure B.2: Suspension Picker



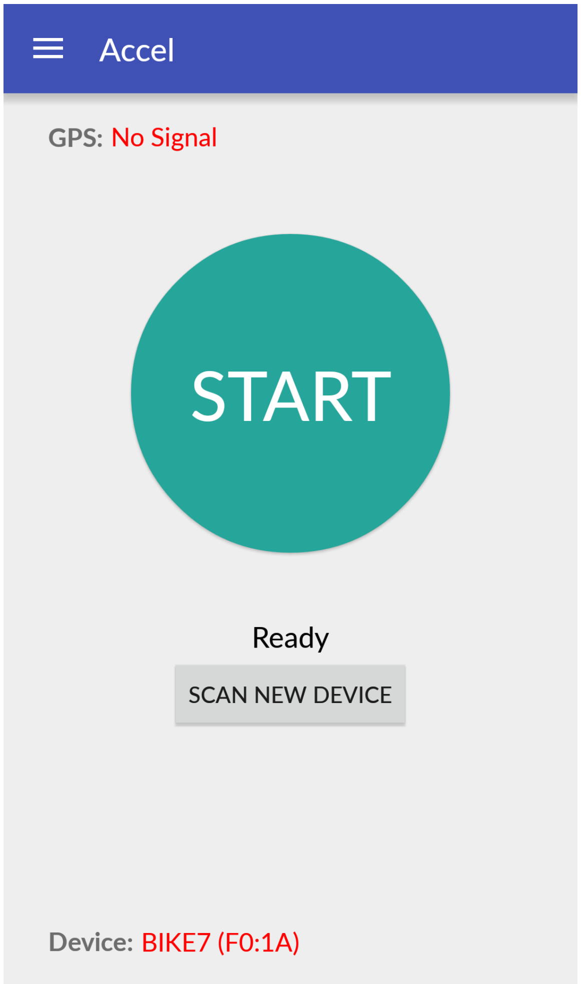
GPS: No Signal

BIKE7
F0:1A:F7:65:84:BB -64 dBm

Ready

Device: BIKE7 (F0:1A)

Figure B.3: Bluetooth Device Picker

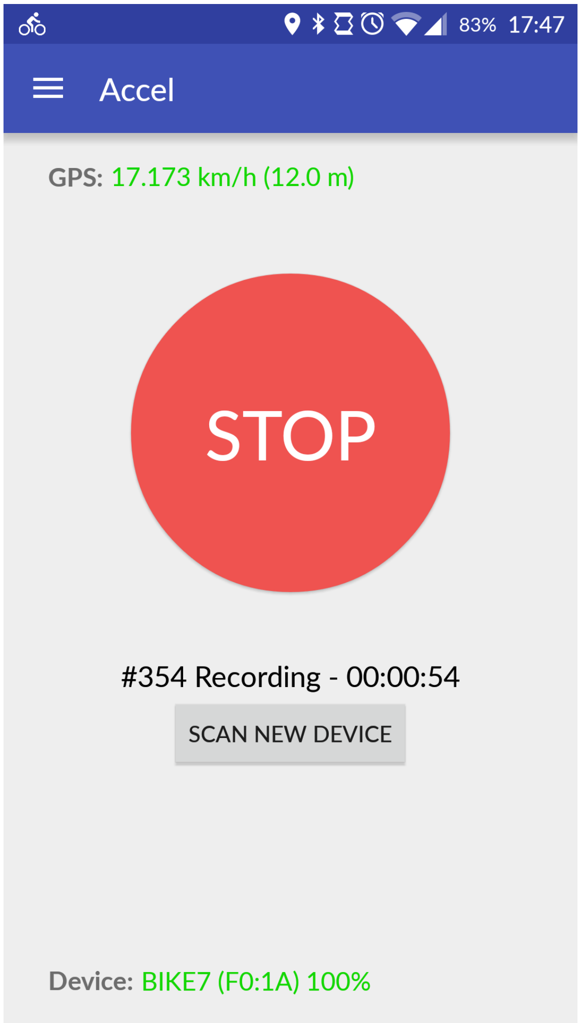


GPS: No Signal

Ready

Device: BIKE7 (F0:1A)

Figure B.4: Main Screen

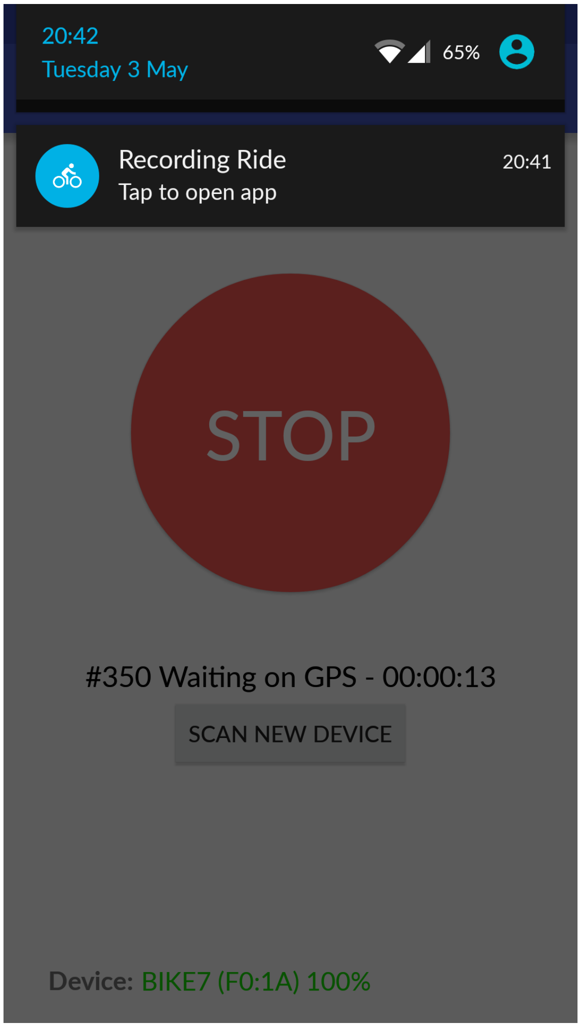


GPS: 17.173 km/h (12.0 m)

#354 Recording - 00:00:54

Device: BIKE7 (F0:1A) 100%

Figure B.5: Ongoing Ride



Recording Ride

Tap to open app

#350 Waiting on GPS - 00:00:13

Device: BIKE7 (F0:1A) 100%

Figure B.6: Recording Ride Notification

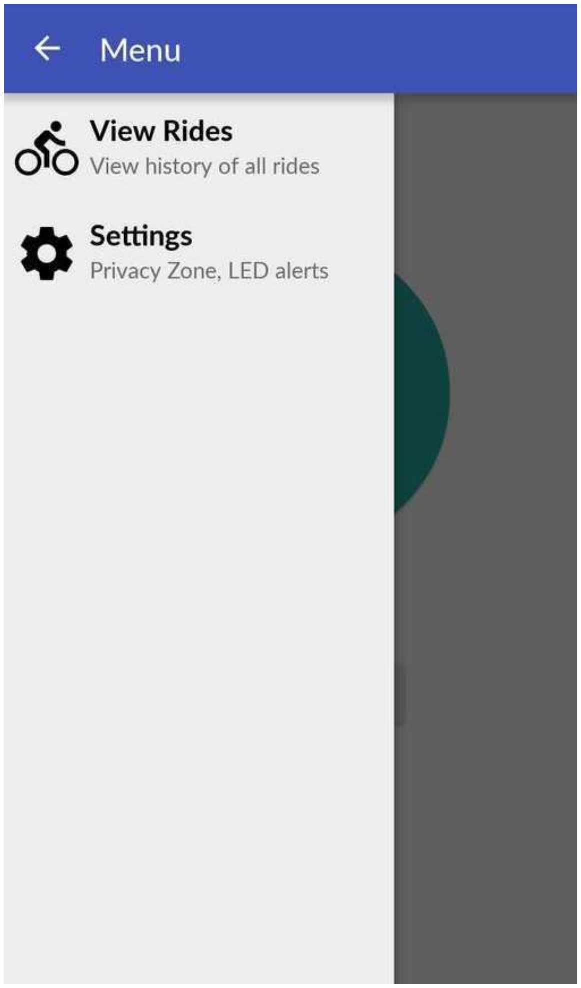
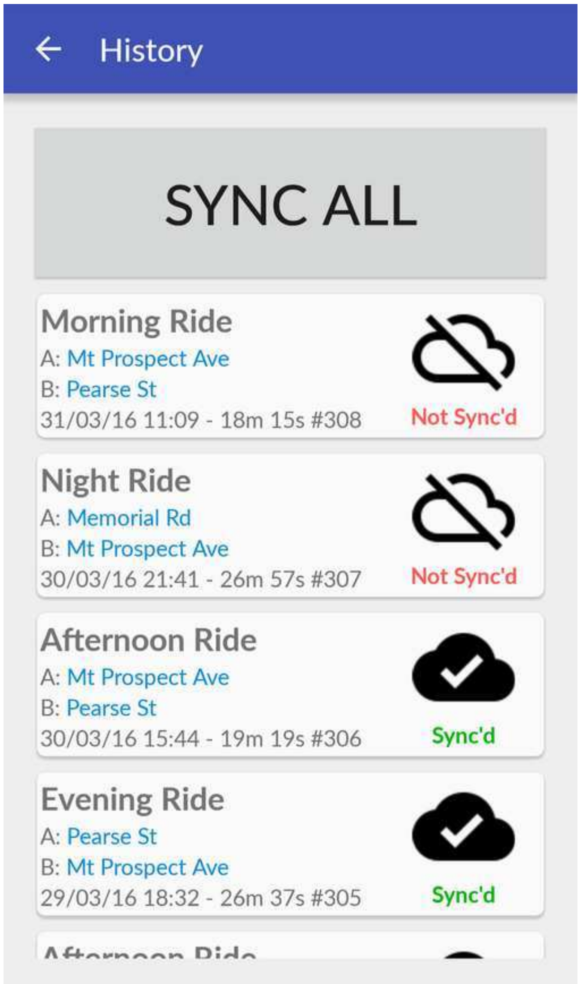


Figure B.7: Sliding Drawer



Morning Ride
A: Mt Prospect Ave
B: Pearse St
31/03/16 11:09 - 18m 15s #308
Not Sync'd

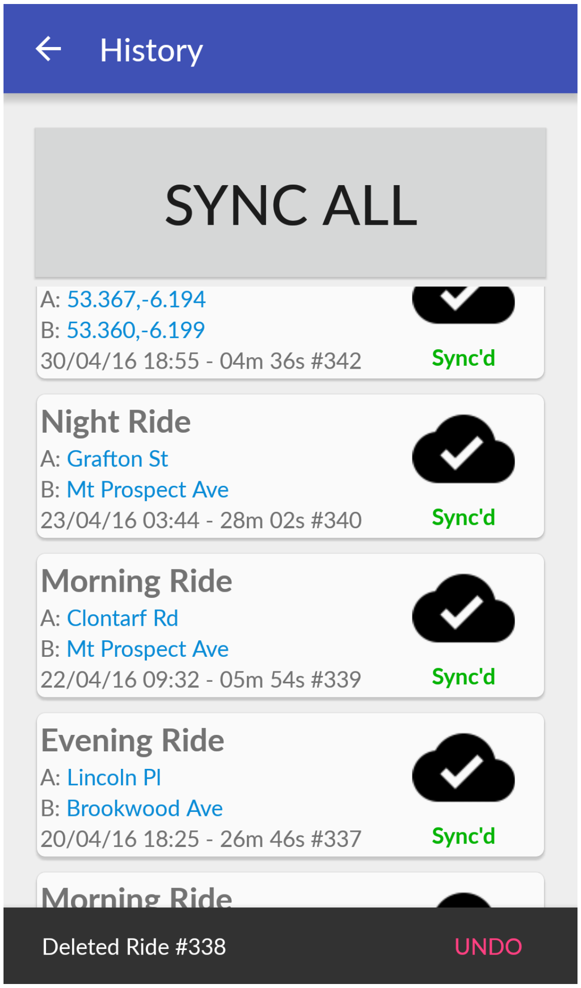
Night Ride
A: Memorial Rd
B: Mt Prospect Ave
30/03/16 21:41 - 26m 57s #307
Not Sync'd

Afternoon Ride
A: Mt Prospect Ave
B: Pearse St
30/03/16 15:44 - 19m 19s #306
Sync'd

Evening Ride
A: Pearse St
B: Mt Prospect Ave
29/03/16 18:32 - 26m 37s #305
Sync'd

Afternoon Ride

Figure B.8: Rides View



A: 53.367,-6.194
B: 53.360,-6.199
30/04/16 18:55 - 04m 36s #342
Sync'd

Night Ride
A: Grafton St
B: Mt Prospect Ave
23/04/16 03:44 - 28m 02s #340
Sync'd

Morning Ride
A: Clontarf Rd
B: Mt Prospect Ave
22/04/16 09:32 - 05m 54s #339
Sync'd

Evening Ride
A: Lincoln Pl
B: Brookwood Ave
20/04/16 18:25 - 26m 46s #337
Sync'd

Morning Ride

Deleted Ride #338

Figure B.9: Deleted Ride, Snackbar with undo action

Appendix C

Hardware Photographs

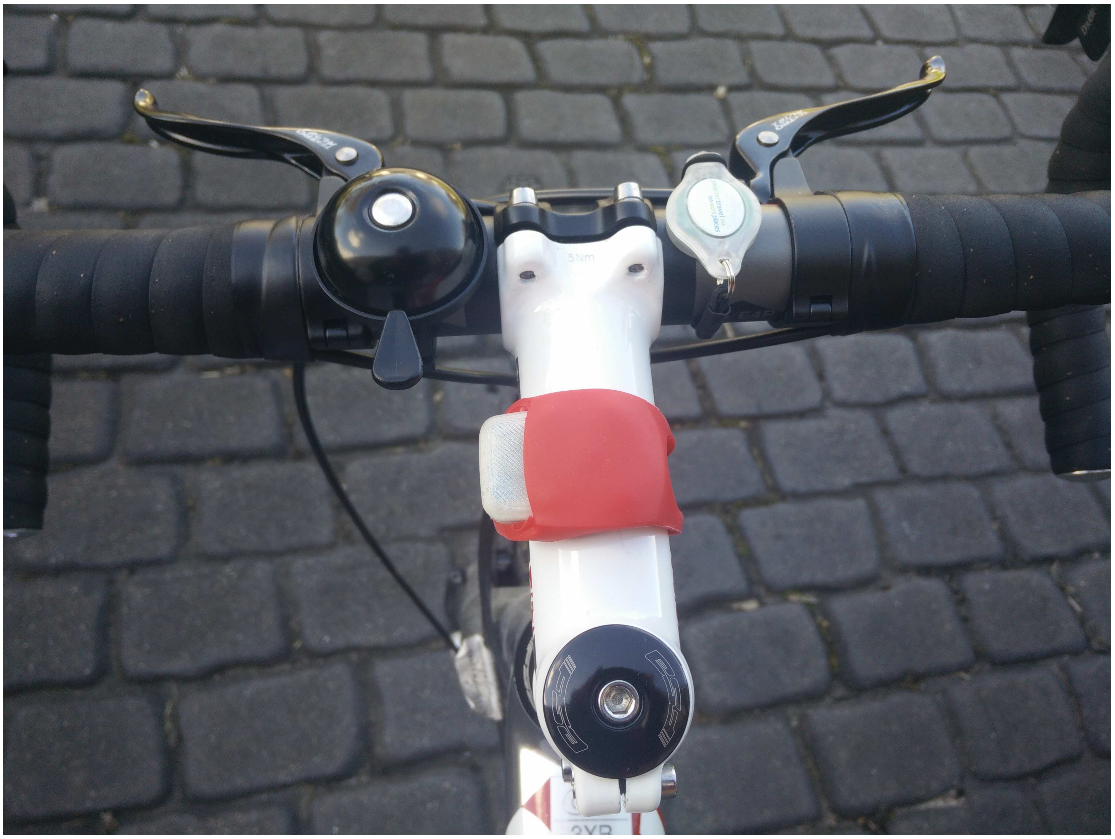


Figure C.1: Device attached to bicycle stem

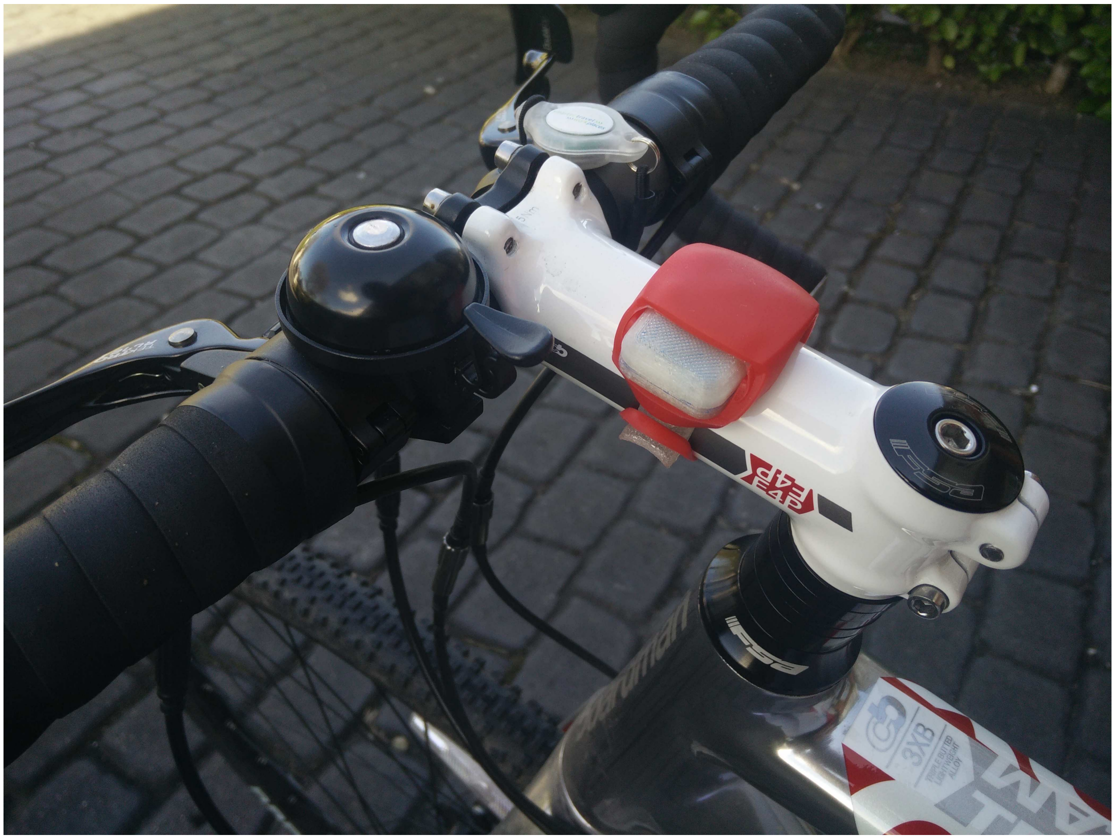


Figure C.2: Device attached to bicycle stem

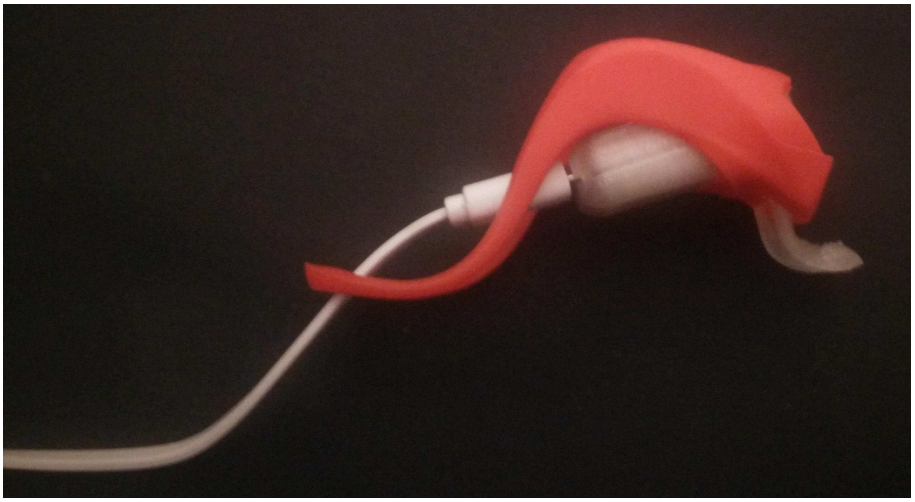


Figure C.3: USB Charging Cable connect to device

Appendix D

User Interface

D.1 Live Heat Map

A public version of the heat map that is hosted on Heroku can be accessed here:
Smooth Route Heat Map - https://smoothroute.herokuapp.com.
A screenshot of this can be seen in Figure D.1.

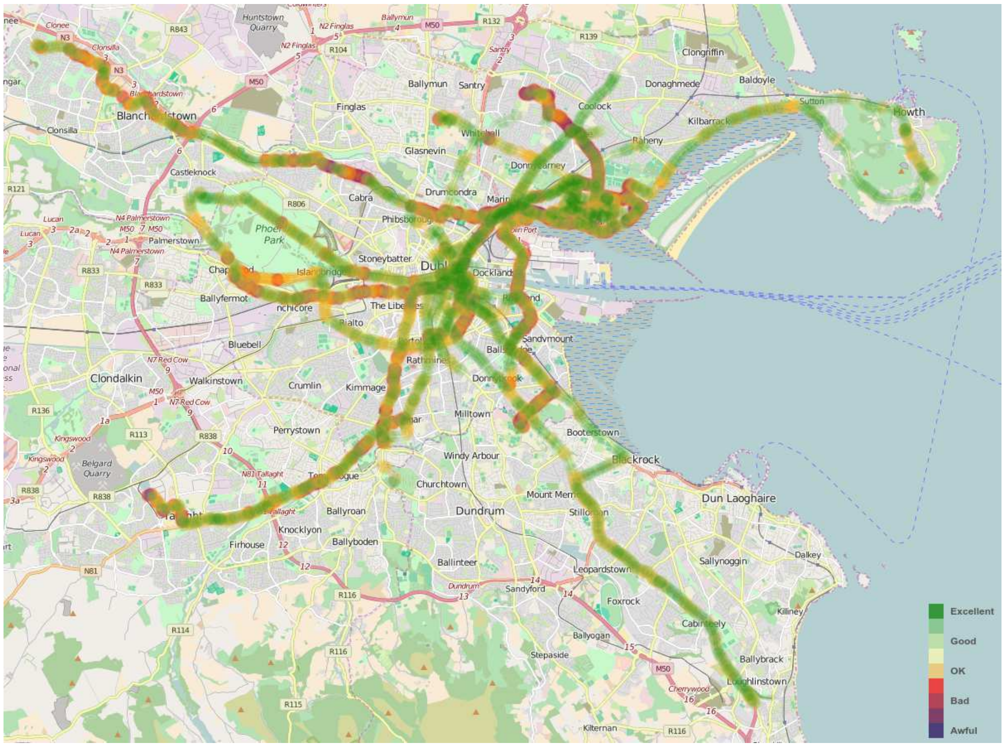
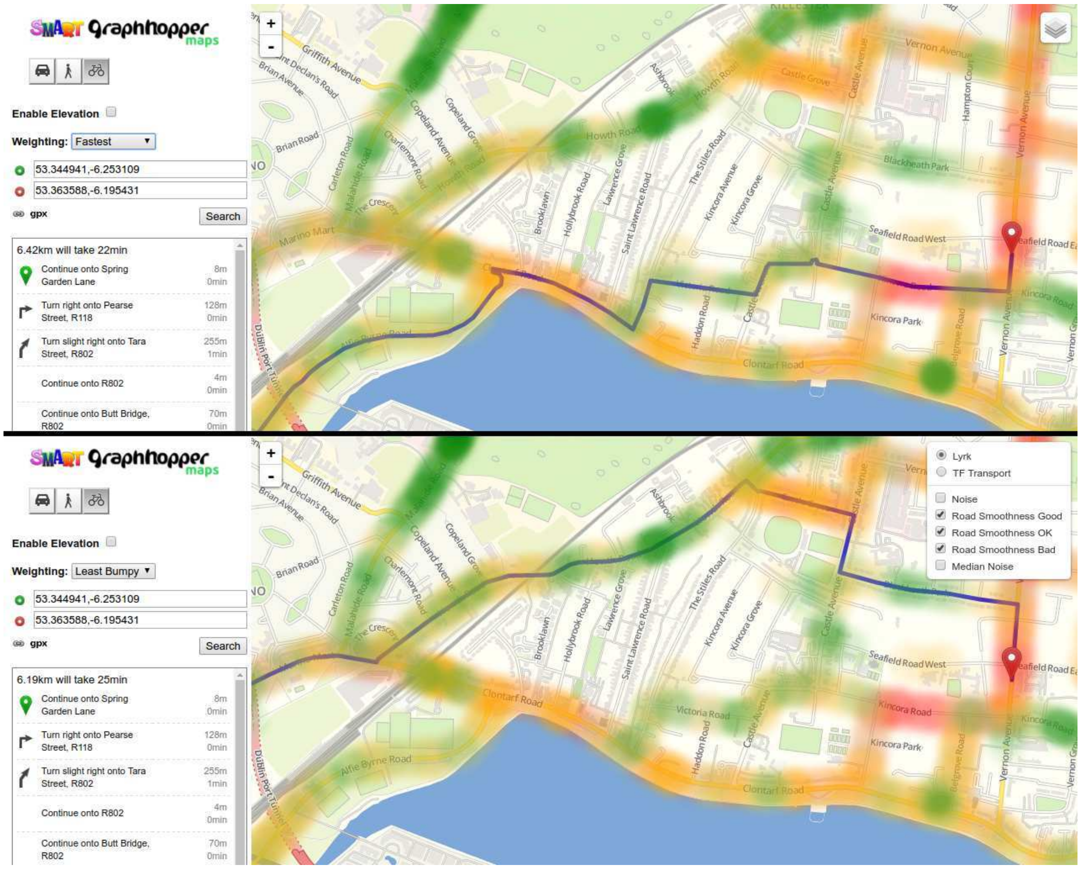


Figure D.1: Heat Map on Heroku

D.2 Smart-GH Routing

The blue line in Figure D.2 represents the route. The heat map ranging from green to red to represent good to bad. The top image displays routing by the fastest route with the lower image showing routing by least bumpy.



Enable Elevation □

Weighting: Fastest

53.344941,-6.253109

53.363588,-6.195431

6.42km will take 22min

Continue onto Spring Garden Lane	8m 0min
Turn right onto Pearse Street, R118	128m 0min
Turn slight right onto Tara Street, R802	255m 1min
Continue onto R802	4m 0min
Continue onto Butt Bridge, R802	70m 0min

Enable Elevation □

Weighting: Least Bumpy

53.344941,-6.253109

53.363588,-6.195431

6.19km will take 25min

Continue onto Spring Garden Lane	8m 0min
Turn right onto Pearse Street, R118	128m 0min
Turn slight right onto Tara Street, R802	255m 1min
Continue onto R802	4m 0min
Continue onto Butt Bridge, R802	70m 0min

● Lyrk
○ TF Transport
□ Noise
☑ Road Smoothness Good
☑ Road Smoothness OK
☑ Road Smoothness Bad
□ Median Noise

Figure D.2: Routing by least bumpy

Appendix E

Snippets

```
{
    "_id": ObjectId("66f9fc1551686f7879a2efbd"),
    "acceleration": {
        "x": -0.226900867608054,
        "y": -0.0578086027912058,
        "z": -0.343113280906029,
        "magnitude": 0.41539446531226
    },
    "acceleration_adjusted": {
        "x": -0.200986228738158,
        "y": -0.0427927046156691,
        "z": -0.251091474572218
        "magnitude": 0.324458947038854,
    }
    "ride_id": ObjectId("56f9fc1351686f7879a25c34"),
    "timestamp": ISODate("2016-01-20T14:46:02.906Z"),
    "section": 2,
    "location": {...}
}
```

Listing 11: Sample JSON Document

```
{
    "location": {
        "bearing": 212.810745274789,
        "altitude": 72,
        "speed": 9.41334,
        "accuracy": 8,
        "position": {
            "type": "Point",
            "coordinates": [
                -6.22537113581372,
                53.3691836733065
            ]
        },
        "raw_position": {
            "type": "Point",
            "coordinates": [
                -6.22538955827217,
                53.3691907752951
            ]
        },
        "address": {
            "city": "Dublin City",
            "country": "Ireland",
            "street_short": "R107",
            "place_id": "ChIJLVAZAk4OZ0gRAq7lDXFx-mU",
            "number": "110",
            "county": "Dublin",
            "geocoded": true,
            "suburb": "Dublin",
            "street": "Malahide Road",
            "formatted_address": "110 Malahide Rd, Dublin, Ireland"
        }
    }
}
```

Listing 12: Sample Expanded Location

```
{
    "_id" : ObjectId("571ccef151686f2f9fb9163e"),
    "device_address" : "f0b19e6cfc1f",
    "rotated" : 0,
    "phone_id" : "c8d766f0f87eda08",
    "number" : 37,
    "vehicle" : 2,
    "started_at" : ISODate("2016-03-30T21:41:16.000Z")
}
```

Listing 13: Sample ride document

```
{
    "_id" : ObjectId("56fe997051686f242f742223"),
    "ride_id" : ObjectId("56f9ff6551686f7879b0898b"),
    "peak_measurement_id" : ObjectId("56f9ff6551686f7879b09229"),
    "value" : 0.587795943145369,
    "axis" : "z",
    "location" : {...}
}
```

Listing 14: Sample peak document

```
{
    "_id" : ObjectId("57312ad051686f7c62001b23"),
    "category" : 2,
    "snap" : true,
    "road_name" : "Howth Road",
    "center" : {
        "type" : "Point",
        "coordinates" : [
            -6.218595808203,
            53.3669381491447
        ]
    },
    "points" : [
        {
            "lat" : 53.3669840699292,
            "lng" : -6.21853011549187,
            "bearing" : 59.2791457779181,
            "value" : 0.514858723446057,
            "timestamp" : ISODate("2016-05-09T20:54:49.000Z")
        },
        ...
        {
            "lat" : 53.3670453735787,
            "lng" : -6.21869520103589,
            "bearing" : 59.279109172663,
            "value" : 0.505388199927735
        }
    ]
}
```

Listing 15: Sample cluster document

Appendix F

Road Ratings

F.1 Ordered by Cluster Average

Best roads Cluster

#	Road Name	Value	Point Count	Cluster Count
1	Beaumont Road	0.06169	16	14
2	George's Quay	0.11590	4	3
3	Phibsborough Road	0.12096	233	65
4	Saint Lawrence Terrace	0.13033	20	10
5	Leeson Street Bridge	0.13224	12	2
6	Suffolk Street	0.13316	5	4
7	Tom Clarke Bridge	0.13525	2	1
8	Dunluce Road	0.13845	10	3
9	N3	0.13918	200	16
10	Ballsbridge Avenue	0.14161	32	8
11	Mayor Street Lower	0.14386	18	13
12	Swords Road	0.14498	22	20
13	Charlemont Road	0.14665	5	4
14	Lower Leeson Street	0.14840	2	1
15	Belgard Square West	0.14888	14	2
16	Portland Row	0.14964	14	7
17	Lombard Street West	0.15070	8	2
18	Claremont Court	0.15159	39	11
19	Hume Street	0.15172	33	14
20	The Grove	0.15200	30	3
21	Morgan Place	0.15203	185	19
22	Morehampton Road	0.15234	70	44
23	Wynnsward Drive	0.15240	8	3
24	Leeson Lane	0.15246	14	3

25	Shandon Gardens	0.15258	759	89
26	Bells Lane	0.15310	16	6
27	Nassau Street	0.15327	30	20
28	Farmleigh View	0.15431	51	14
29	Copeland Grove	0.15643	6	5
30	Georgian Village	0.15689	4	2
31	Clare Street	0.15757	76	13
32	Haverty Road	0.15802	4	2
33	Old Abbey Street	0.15883	28	3
34	Howth Road	0.15909	8192	598
35	Saint Nessan's Terrace	0.15911	39	18
36	McAuley Avenue	0.16013	14	12
37	White's Road	0.16058	485	156
38	Burgh Quay	0.16145	14	12
39	Claremont Crescent	0.16227	7	6
40	Poolbeg Street	0.16237	6	5
41	Church Lane	0.16376	18	5
42	Navan Road	0.16382	1986	231
43	Ely Place	0.16444	75	22
44	River Road	0.16483	855	91
45	Camden Street Lower	0.16529	37	28
46	O'Connell Street Upper	0.16536	39	31
47	Leeson Street Upper	0.16540	90	59
48	Rathmines Road Lower	0.16677	554	94
49	Greenfield Road	0.16693	86	74
50	Merrion Street Lower	0.16717	35	9
51	Carrickbrack Road	0.16770	236	204
52	James Larkin Road	0.16876	210	129
53	Gardiner Street Upper	0.16995	28	22
54	Saint Stephen's Green	0.17182	161	65
55	Thormanby Road	0.17198	16	14
56	Rathmichael Manor	0.17200	36	18
57	George's Dock	0.17407	13	11
58	Shelbourne Avenue	0.17428	104	12
59	Rock Road	0.17440	126	100
60	Sean Macdermott Street Upper	0.17469	27	13
61	Wexford Street	0.17473	27	14
62	Temple Street	0.17506	14	12
63	Malahide Road	0.17512	2072	654
64	Pembroke Road	0.17557	102	66
65	Copeland Avenue	0.17600	2	1
66	Drumcondra Road Upper	0.17654	54	43

67	Grafton Street	0.17684	7	5
68	Sarsfield Road	0.17740	154	34
69	Bessborough Avenue	0.17804	4	3
70	Ormond Quay Lower	0.17976	121	29
71	Donnybrook Road	0.17994	76	55
72	Howth Terrace	0.18019	16	12
73	Westhaven	0.18108	15	13
74	James's Street	0.18148	26	4
75	Harcourt Street	0.18161	26	19
76	Wellington Road	0.18267	6	2
77	Bailey Green Road	0.18401	71	27
78	R138	0.18495	42	24
79	Christ Church Square	0.18508	30	8
80	Albert College Park	0.18603	31	17
81	Rathgar Road	0.18621	944	135
82	Grand Canal Bridge	0.18739	12	10
83	Dublin Port Tunnel	0.18757	49	22
84	Nicholas Street	0.18786	63	20
85	Saint Lawrence Road	0.18974	4	3
86	Shamrock Place	0.19079	9	5
87	Bray Road	0.19164	234	183
88	Conyngham Road	0.19312	202	113
89	Ely Place Upper	0.19440	17	13
90	Grosvenor Court	0.19462	5	4
91	Saint Nessan's Close	0.19475	34	14
92	Northumberland Road	0.19481	217	96
93	Church Street	0.19543	18	10
94	Harcourt Road	0.19705	35	16
95	Bushy Park Road	0.19799	18	16
96	Harbour Road	0.19858	12	10
97	Ranelagh Road	0.19889	106	62
98	Bramblefield Drive	0.19964	26	7
99	Gardiner Street Middle	0.20021	28	14
100	Macken Street	0.20029	40	25
101	Mount Merrion Avenue	0.20050	171	125
102	R137	0.20085	365	88
103	Wellington Lane	0.20099	5	4
104	Wolfe Tone Quay	0.20158	157	39
105	Glengarriff Parade	0.20169	6	4
106	Crampton Avenue	0.20170	110	34
107	Knockmaroon Hill	0.20177	78	36
108	Inns Quay	0.20427	121	30

109	Causeway Road	0.20465	5	4
110	N11	0.20532	549	386
111	McAuley Road	0.20554	24	14
112	Inchicore Parade	0.20563	77	17
113	Fenian Street	0.20939	6	4
114	R114	0.21094	53	12
115	Cathal Brugha Street	0.21162	45	18
116	Seaview Avenue North	0.21195	134	21
117	Stiles Court	0.21207	7	6
118	Earlsfort Terrace	0.21248	60	36
119	Holles Street	0.21265	6	4
120	Collins Avenue West	0.21274	36	26
121	Landen Road	0.21279	175	25
122	Hogan Place	0.21286	4	2
123	Townsend Street	0.21287	517	94
124	Mount Street Lower	0.21335	117	56
125	Summerville	0.21556	109	15
126	Littlepace Park	0.21624	20	8
127	Littlepace Gallops	0.21644	1299	128
128	Balglass Road	0.21696	34	14
129	Ellis Quay	0.21714	132	26
130	Rathfarnham Wood	0.21780	26	19
131	Chesterfield Avenue	0.21843	1116	363
132	Bachelors Walk	0.21883	159	38
133	Beresford Place	0.22128	418	45
134	Tallaght Road	0.22272	723	185
135	Third Avenue	0.22362	145	21
136	O'Connell Street Lower	0.22417	12	10
137	Dodder View Cottages	0.22471	23	9
138	Clanbrassil Street Lower	0.22564	301	57
139	Canal Road	0.22763	158	31
140	BÃşthar Ghort Na nGrÃąs	0.22863	14	5
141	South Lotts Road	0.22978	26	22
142	Fourth Avenue	0.23038	99	20
143	Grand Canal Street Upper	0.23050	28	16
144	Ormond Quay Upper	0.23073	124	31
145	The Drive	0.23073	4	2
146	Charlemont Street	0.23086	57	27
147	Foster Place	0.23105	8	4
148	Brookwood Grove	0.23174	18	11
149	N81	0.23184	1568	401

150	Arran Quay	0.23191	140	41
151	Custom House Quay	0.23243	6	5
152	Kincora Grove	0.23254	52	30
153	Merrion Road	0.23288	3373	327
154	Leeson Street Lower	0.23321	48	34
155	R806	0.23365	220	17
156	Merrion Square North	0.23365	42	21
157	Clanbrassil Street Upper	0.23393	244	41
158	New Street South	0.23478	245	38
159	Brookwood Park	0.23492	10	3
160	Stillorgan Road	0.23512	844	538
161	North Road	0.23571	19	8
162	Balkill Road	0.23650	102	50
163	R148	0.23699	50	16
164	Collins Avenue East	0.23744	35	25
165	Victoria Road	0.23860	216	45
166	Kilmainham Lane	0.23871	613	79
167	Memorial Road	0.23871	133	22
168	Blackheath Park	0.23913	329	76
169	North Strand Road	0.23976	4352	156
170	Watermill Road	0.23999	10	8
171	Dorset Street Lower	0.24287	38	24
172	Collins Avenue Extension	0.24338	99	52
173	Sean Macdermott Street Lower	0.24363	84	40
174	Canon Place	0.24375	43	10
175	Fairview	0.24444	1267	120
176	Grand Canal Street Lower	0.24575	16	12
177	Dublin Road	0.24582	354	216
178	Merlyn Park	0.24582	9	2
179	Richmond Street South	0.24766	158	25
180	Anglesea Road	0.24852	146	64
181	R802	0.25017	163	18
182	South Great George's Street	0.25022	51	28
183	Ringsend Road	0.25087	8	7
184	Eccles Street	0.25125	22	8
185	Ballsbridge Park	0.25130	223	31
186	Seafield Avenue	0.25199	54	29
187	Dolphin Road	0.25199	128	77
188	Sarsfield Quay	0.25283	41	12
189	Belgard Road	0.25294	178	32
190	Leinster Street South	0.25305	26	10
191	Parnell Road	0.25424	118	72

192	Cookstown Road	0.25444	93	14
193	Seville Place	0.25547	4	2
194	Eden Quay	0.25951	146	36
195	Marino Mart	0.25964	1023	99
196	Prince's Street South	0.25968	10	6
197	Gardiner Street Lower	0.26057	98	48
198	Amiens Street	0.26133	3803	194
199	Lucan Road	0.26371	43	10
200	Cookstown Way	0.26408	48	10
201	Ranelagh	0.26597	56	33
202	Pelletstown Manor	0.26738	66	10
203	Dunseverick Road	0.26831	149	26
204	Main Street	0.26968	1120	121
205	Herbert Cottages	0.27040	37	13
206	Pearse Street	0.27135	898	144
207	South Circular Road	0.27219	164	71
208	O'Hara Avenue	0.27269	8	2
209	Westland Row	0.27299	182	49
210	Royal Canal Way	0.27334	297	39
211	Vernon Drive	0.27367	15	4
212	Talbot Court	0.27516	58	22
213	Trinity Terrace	0.27586	10	2
214	Preston Street	0.27635	86	4
215	Beatty's Avenue	0.27643	112	19
216	Rathborne Avenue	0.27782	8	6
217	Alfie Byrne Road	0.27882	1813	123
218	R105	0.27927	201	33
219	Grand Parade	0.28041	295	52
220	Lincoln Place	0.28060	145	24
221	The Stiles Road	0.28079	181	25
222	Grove Road	0.28181	644	109
223	Patrick Street	0.28192	189	38
224	Templeogue Road	0.28239	2350	470
225	Mountjoy Square West	0.28242	21	8
226	Martin's Row	0.28257	222	44
227	Rathfarnham Road	0.28316	471	52
228	Killarney Street	0.28381	14	7
229	Westbourne Road	0.28435	116	12
230	Tower Road	0.28549	104	51
231	Fitzwilliam Square East	0.28638	32	10
232	Drumcondra Road Lower	0.28649	261	62
233	M50	0.28655	123	29

234	Belgard Square East	0.28705	10	2
235	Mount Drummond Avenue	0.28963	10	2
236	Conquer Hill Road	0.29111	10	6
237	North Circular Road	0.29165	45	19
238	Talbot Memorial Bridge	0.29179	175	12
239	East Wall Road	0.29191	2547	271
240	Belgard Square North	0.29196	523	91
241	Dawson Street	0.29432	22	14
242	Castleview	0.29462	15	9
243	Thomas Street	0.29470	617	81
244	Redmond's Hill	0.29621	13	6
245	Snugborough Road	0.29667	1224	92
246	Harold's Cross Road	0.29740	2291	220
247	Orwell Road	0.29742	36	21
248	Buckingham Street Upper	0.29903	5	2
249	D'Olier Street	0.29921	5	3
250	Belgrove Park	0.29978	35	7
251	Rathborne Drive	0.29983	113	10
252	Inchicore Road	0.30021	748	83
253	Tara Street	0.30104	476	58
254	Royal Canal Avenue	0.30117	1052	87
255	Maiden's Row	0.30209	99	20
256	Anglesea Street	0.30273	8	3
257	Sandwith Street Lower	0.30345	5	3
258	Aungier Street	0.30374	33	18
259	Irishtown Road	0.30425	812	58
260	Saint Ignatius Road	0.30492	8	4
261	Ratoath Estate	0.30658	78	19
262	R111	0.30835	6	4
263	Merrion Square South	0.30900	165	35
264	R811	0.30906	92	30
265	Huntstown Drive	0.31170	79	14
266	Clonliffe Road	0.31266	1424	121
267	Bridge Street	0.31345	201	37
268	R113	0.31407	425	76
269	Annesley Bridge Road	0.31476	1790	121
270	Roebuck Castle	0.31605	14	8
271	Zion Road	0.31696	19	9
272	Thornberry Square	0.31776	6	4
273	Clontarf Road	0.31844	12075	689
274	Summerfield Rise	0.31916	14	2
275	Lord Edward Street	0.32130	306	33

276	Magennis Place	0.32168	4	2
277	Suir Road	0.32222	26	12
278	Lansdowne Road	0.32243	384	26
279	Thomas Street West	0.32256	191	27
280	Creighton Street	0.32441	32	17
281	Grattan Crescent	0.32736	90	12
282	Terenure Road East	0.32743	294	70
283	Adelaide Road	0.32809	112	27
284	Christchurch Place	0.32814	281	21
285	Kincora Avenue	0.32826	28	10
286	Prospect Road	0.32965	19	9
287	Fortfield Road	0.32990	6	2
288	Old Navan Road	0.33124	464	51
289	Sheepmoor Grove	0.33183	345	33
290	Merrion Square West	0.33283	170	23
291	Herbert Road	0.33296	101	23
292	Fergus Road	0.33546	24	2
293	Westmoreland Street	0.33844	137	34
294	City Quay	0.33870	215	52
295	Terenure Road North	0.33935	847	71
296	Collins Avenue	0.34277	121	65
297	Whitworth Road	0.34336	1016	120
298	Seapark Road	0.34380	327	76
299	Parnell Street	0.34429	67	28
300	Whitestown Park	0.34623	837	83
301	R815	0.34667	4	2
302	Merrion Street Upper	0.34805	103	22
303	Castlekevin Road	0.34999	244	45
304	Ashfield Lawn	0.35134	4	2
305	College Green	0.35331	344	56
306	Dame Street	0.35361	442	65
307	Meath Street	0.35470	10	2
308	Dollymount Avenue	0.35518	387	73
309	Belgrove Road	0.35559	326	34
310	Mark Street	0.35650	7	4
311	Saint Ignatius Avenue	0.35689	6	2
312	Seafield Road West	0.35841	219	66
313	Huntstown Wood	0.35934	300	64
314	Cumberland Street South	0.36077	4	2
315	Huntstown Way	0.36106	489	63
316	Charlotte Way	0.36423	6	2
317	R121	0.36442	362	41

318	O'Connell Bridge	0.36592	49	14
319	Inchicore Terrace North	0.36709	143	22
320	Fitzwilliam Lane	0.36819	8	2
321	Hollybrook Park	0.37016	35	10
322	Seafield Road East	0.37038	134	55
323	Mark's Lane	0.37112	4	2
324	Hawkins Street	0.37194	45	14
325	Martin Savage Park	0.37344	171	28
326	Parkgate Street	0.37450	63	20
327	Castle Avenue	0.37477	1128	171
328	Huntstown Green	0.37881	81	11
329	Nutley Lane	0.38053	1639	126
330	R109	0.38065	72	17
331	Archers Wood	0.38241	95	27
332	Lombard Street East	0.38262	325	52
333	Fitzwilliam Street Upper	0.38543	164	30
334	Tritonville Road	0.38644	323	26
335	Greenfield Crescent	0.38732	174	20
336	Castle Grove	0.38879	635	66
337	Chapelizod Road	0.39024	201	90
338	Saint Patrick's Villas	0.39593	65	8
339	Albert College Avenue	0.40061	15	6
340	Saint Laurence's Road	0.40082	737	94
341	Thorncastle Street	0.40372	766	63
342	Huntstown Park	0.40743	45	5
343	Ashfield Way	0.40908	4	2
344	Bow Lane West	0.40973	357	49
345	College Street	0.41715	80	23
346	East-link Toll Bridge	0.42653	162	19
347	Greenfield Park	0.42983	1350	103
348	Terenure Place	0.43231	70	11
349	Vernon Avenue	0.43273	4270	260
350	Fitzwilliam Street Lower	0.44231	61	12
351	Sandwith Street Upper	0.44581	7	3
352	Bayview Avenue	0.44819	13	3
353	Kincora Road	0.45313	1093	187
354	Luke Kelly Bridge	0.45333	179	16
355	Fitzwilliam Place	0.45689	140	26
356	Brookwood Avenue	0.46259	1098	157
357	Moss Street	0.46646	359	32
358	Denzille Lane	0.46785	7	3
359	Shaw Street	0.47476	153	18

360	Mount Prospect Avenue	0.47768	1572	192
361	Gracefield Road	0.48047	302	45
362	Cornmarket	0.48542	80	10
363	Newbridge Avenue	0.48669	510	37
364	Ardlea Road	0.48942	670	111
365	Ratoath Road	0.49776	461	65
366	Ballyboggan Road	0.50316	144	23
367	Fairview Strand	0.51601	547	51
368	High Street	0.51621	166	19
369	Sybil Hill Road	0.52934	1178	153
370	Kilmore Road	0.54660	377	77
371	Oulton Road	0.55025	269	41
372	R131	0.55405	84	17
373	Dollymount Park	0.59079	532	20
374	Castle Road	0.60068	8	2
375	Royal Canal Way, Glasnevin	0.61363	732	52
376	Kilbarron Road	0.61692	297	61
377	R814	0.61782	16	2
378	Ballyshannon Road	0.61832	128	24

Table F.1: Order of roads by cluster

F.2 Ordered by Average

#	**Road Name**	**X**	**Y**	**Z**	**Speed**	**Point**
		m/s^2	**m/s^2**	**m/s^2**	**km/h**	**Count**
1	Beaumont Road	0.165	0.142	0.074	25.757	7044
2	Lawrence Grove	0.062	0.080	0.082	17.404	625
3	Ossory Road	0.065	0.090	0.083	24.341	147
4	Towerview Cottages	0.038	0.123	0.091	27.207	155
5	Leinster Street North	0.061	0.106	0.093	24.525	5603
6	Swords Road	0.135	0.094	0.094	25.019	4554
7	Dalcassian Downs	0.066	0.117	0.095	21.264	8470
8	Ashbrook	0.053	0.099	0.096	20.754	133
9	Talbot Street	0.066	0.097	0.099	11.763	100
10	Leeson Lane	0.064	0.126	0.100	16.730	1626
11	Saint John's Wood	0.076	0.118	0.101	8.684	531
12	Slaney Road	0.055	0.116	0.102	23.822	1223
13	Clareville Grove	0.057	0.130	0.103	21.627	80
14	Botanic Road	0.042	0.122	0.104	32.335	111
15	Hatch Street Lower	0.078	0.135	0.104	11.848	152
16	R102	0.057	0.115	0.107	20.431	19893

17	Leeson Street Bridge	0.105	0.145	0.107	10.405	1990
18	Strandville Avenue	0.094	0.112	0.109	23.466	52
19	Dunsink Lane	0.059	0.116	0.109	22.053	22130
20	Deerhaven Walk	0.050	0.126	0.110	20.670	148
21	Mill Lane	0.065	0.121	0.110	18.861	32139
22	Claremont Lawns	0.058	0.123	0.110	23.705	23095
23	Phibsborough Road	0.068	0.126	0.111	20.014	19697
24	Pace Avenue	0.095	0.140	0.111	13.603	2501
25	Deerhaven Avenue	0.049	0.139	0.112	21.886	179
26	Shandon Park	0.068	0.130	0.114	20.738	18709
27	Charlemont Road	0.083	0.131	0.116	23.166	310
28	Finglas Road	0.050	0.136	0.117	30.541	200
29	Saint David's Wood	0.071	0.112	0.120	17.646	607
30	N3	0.061	0.131	0.120	25.363	13672
31	Shandon Gardens	0.062	0.133	0.121	19.979	86349
32	Saint Attracta Road	0.056	0.128	0.122	21.009	3647
33	Richmond Row	0.104	0.167	0.122	15.216	250
34	Farmleigh View	0.060	0.107	0.122	16.095	5222
35	Prospect Avenue	0.049	0.152	0.124	33.192	178
36	Killester Avenue	0.065	0.113	0.125	11.034	72
37	Dunluce Road	0.091	0.131	0.125	13.170	723
38	Howth Road	0.080	0.125	0.126	21.103	1110662
39	Shamrock Place	0.090	0.124	0.127	17.060	527
40	Newcomen Avenue	0.077	0.112	0.127	21.880	279
41	Herbert Park	0.084	0.128	0.128	17.352	126
42	Morgan Place	0.058	0.137	0.128	28.780	17388
43	Ballsbridge Avenue	0.057	0.120	0.129	18.972	3386
44	Wynnsward Drive	0.067	0.153	0.129	15.054	787
45	Evora Park	0.094	0.134	0.130	11.780	65
46	Hume Street	0.070	0.143	0.131	18.942	3731
47	Claremont Court	0.055	0.137	0.131	25.146	2503
48	Deerhaven Park	0.067	0.156	0.132	25.160	1358
49	Memorial Road	0.094	0.147	0.132	19.398	43174
50	Decies Road	0.082	0.118	0.132	19.024	305
51	Claremont Crescent	0.062	0.140	0.132	23.067	15394
52	Buckingham Street Lower	0.045	0.164	0.132	16.768	61
53	River Road	0.065	0.150	0.134	20.347	75481
54	Clare Street	0.082	0.144	0.135	16.196	10312
55	Old Abbey Street	0.090	0.142	0.137	24.459	1810
56	Saint Lawrence Terrace	0.114	0.190	0.137	12.018	1947
57	Sean Macdermott Street Upper	0.085	0.124	0.138	19.220	1719
58	Saint Nessan's Terrace	0.092	0.152	0.138	9.824	3976

59	Navan Road	0.065	0.152	0.139	22.684	187412
60	The Grove	0.065	0.152	0.139	22.732	2113
61	Carne Court	0.051	0.159	0.139	21.753	232
62	Rathmines Road Lower	0.103	0.157	0.139	23.739	48344
63	White's Road	0.064	0.112	0.140	25.902	36477
64	Nicholas Street	0.057	0.153	0.141	19.660	6443
65	Upper Merrion Street	0.111	0.160	0.141	13.716	430
66	Ormond Quay Lower	0.063	0.120	0.141	19.274	12178
67	Dorset Street Upper	0.083	0.171	0.142	17.450	230
68	Bessborough Avenue	0.083	0.141	0.142	13.873	250
69	Ely Place	0.096	0.165	0.142	17.725	8599
70	Bells Lane	0.067	0.163	0.143	19.905	913
71	Merrion Street Lower	0.140	0.171	0.144	18.781	2712
72	O'Connell Street Upper	0.102	0.132	0.145	20.525	3134
73	Morehampton Road	0.096	0.173	0.145	17.400	6053
74	Malahide Road	0.086	0.143	0.146	19.396	195593
75	Third Avenue	0.076	0.176	0.148	17.088	18256
76	Leeson Street Upper	0.093	0.180	0.149	17.552	6614
77	Iveragh Court	0.067	0.133	0.151	16.370	449
78	Saint Stephen's Green	0.093	0.178	0.151	16.035	12306
79	Ely Place Upper	0.092	0.156	0.151	19.015	2536
80	Beresford Place	0.085	0.155	0.151	21.881	78189
81	Causeway Road	0.131	0.141	0.151	10.756	172
82	Oriel Street Upper	0.070	0.190	0.152	24.503	212
83	Rathgar Road	0.104	0.166	0.152	20.886	87248
84	Vernon Drive	0.079	0.138	0.152	19.093	3881
85	Estate Cottages	0.081	0.158	0.152	20.255	4895
86	Shelbourne Avenue	0.073	0.144	0.152	15.364	19623
87	Sarsfield Road	0.075	0.126	0.153	20.971	13040
88	Seaview Avenue North	0.099	0.190	0.153	16.929	15090
89	Booterstown Avenue	0.121	0.204	0.153	9.257	75
90	Deerhaven View	0.071	0.149	0.153	22.678	718
91	Merlyn Park	0.089	0.165	0.153	27.035	1501
92	Wexford Street	0.068	0.187	0.153	17.774	2340
93	Blunden Drive	0.099	0.162	0.153	19.665	329
94	Westhaven	0.061	0.183	0.154	21.295	3086
95	Greenfield Road	0.080	0.153	0.154	19.885	6965
96	McAuley Avenue	0.087	0.158	0.154	16.957	1323
97	Crinan Strand	0.094	0.195	0.155	22.788	180
98	Saint Philomena's Road	0.061	0.150	0.155	24.743	303
99	Shelbourne Road	0.073	0.154	0.155	18.053	12572
100	Hollybrook Park	0.091	0.150	0.156	24.348	8126

101	Frankfort Cottages	0.054	0.129	0.156	18.803	76
102	Fourth Avenue	0.067	0.172	0.157	19.110	14048
103	Wellington Road	0.094	0.147	0.157	26.593	406
104	Clanbrassil Street Lower	0.068	0.177	0.157	23.021	39036
105	Littlepace Park	0.086	0.176	0.157	15.284	3886
106	Talbot Memorial Bridge	0.102	0.167	0.157	21.759	37477
107	Donnybrook Road	0.097	0.185	0.158	17.838	6361
108	George's Quay	0.103	0.152	0.158	19.423	396
109	Inns Quay	0.077	0.133	0.159	21.847	12183
110	James Larkin Road	0.095	0.157	0.159	25.347	19082
111	Belgard Square West	0.065	0.177	0.159	19.230	1000
112	Harcourt Road	0.107	0.171	0.159	18.959	4897
113	Pembroke Road	0.097	0.186	0.160	24.049	11409
114	Rathmichael Manor	0.101	0.220	0.160	14.587	3862
115	R806	0.071	0.176	0.160	21.157	29463
116	R815	0.079	0.160	0.160	21.527	1937
117	Harbour Road	0.091	0.171	0.160	23.332	733
118	R802	0.105	0.174	0.161	20.353	31642
119	Inchicore Parade	0.093	0.139	0.161	19.384	14522
120	Landen Road	0.072	0.133	0.162	21.867	27403
121	Northumberland Road	0.109	0.193	0.163	23.305	18464
122	Summerville	0.104	0.178	0.163	14.477	10237
123	Joshua Lane	0.075	0.165	0.163	21.235	102
124	Howth Terrace	0.103	0.191	0.164	12.001	2451
125	Clanbrassil Street Upper	0.081	0.192	0.164	19.945	32113
126	Victoria Road	0.107	0.197	0.164	19.146	23566
127	Leeson Street Lower	0.109	0.197	0.165	16.685	11516
128	North Strand Road	0.094	0.169	0.165	22.209	764539
129	Albert College Park	0.058	0.168	0.165	16.056	2174
130	Haverty Road	0.078	0.125	0.165	18.911	447
131	Grand Canal Bridge	0.138	0.214	0.166	20.525	827
132	New Street South	0.071	0.189	0.166	21.914	27937
133	Townsend Street	0.087	0.151	0.167	21.073	57226
134	R137	0.068	0.194	0.167	25.189	29783
135	Bailey Green Road	0.091	0.182	0.167	14.660	5990
136	Dodder View Cottages	0.083	0.173	0.167	23.900	4844
137	The Drive	0.078	0.192	0.167	25.722	2651
138	James's Street	0.059	0.133	0.167	16.083	2882
139	R138	0.095	0.189	0.169	17.394	2961
140	Christ Church Square	0.084	0.206	0.169	15.223	4392
141	Ormond Quay Upper	0.076	0.146	0.169	21.591	12383
142	Wolfe Tone Quay	0.101	0.159	0.171	29.227	12937

143	The Moorings	0.054	0.129	0.171	13.867	6705
144	Littlepace Gallops	0.080	0.192	0.171	22.297	134597
145	Amiens Street	0.094	0.175	0.172	20.645	679215
146	Ellis Quay	0.085	0.147	0.172	22.335	11041
147	Cookstown Way	0.076	0.180	0.173	15.363	4464
148	Conyngham Road	0.089	0.157	0.173	26.990	16661
149	Inchicore Square North	0.085	0.133	0.173	23.131	153
150	Crampton Avenue	0.078	0.179	0.174	16.154	21948
151	R114	0.099	0.182	0.174	19.857	5645
152	Marino Crescent	0.100	0.175	0.174	12.472	824
153	The Willows	0.061	0.169	0.174	19.598	175
154	Lindsay Grove	0.090	0.221	0.174	19.560	1041
155	Priorswood Road	0.093	0.176	0.175	24.518	356
156	Bachelors Walk	0.088	0.153	0.175	17.641	17033
157	Pace Road	0.112	0.194	0.175	14.073	196
158	McAuley Road	0.084	0.175	0.175	14.572	2057
159	Preston Street	0.093	0.178	0.175	21.343	7758
160	George's Dock	0.089	0.192	0.175	13.440	2131
161	Herbert Cottages	0.083	0.172	0.175	22.859	5744
162	Harbourmaster Place	0.091	0.201	0.176	18.856	2731
163	Dublin Port Tunnel	0.093	0.181	0.177	20.167	2791
164	Saint Fintan Terrace	0.075	0.176	0.177	22.772	376
165	Lombard Street West	0.069	0.178	0.177	24.142	518
166	Merrion Road	0.097	0.193	0.177	25.018	378621
167	Deerhaven Close	0.076	0.196	0.177	20.527	285
168	Canal Road	0.100	0.183	0.177	24.710	16281
169	M11	0.148	0.251	0.177	18.319	358
170	Saint Nessan's Close	0.092	0.184	0.178	13.162	3288
171	Ballyfermot Road	0.056	0.181	0.178	14.241	153
172	Tara Street	0.090	0.178	0.178	19.069	84164
173	Kilmainham Lane	0.081	0.153	0.178	17.772	81087
174	Pearse Street	0.108	0.180	0.178	19.330	153616
175	The Park	0.081	0.140	0.179	25.418	262
176	Mount Street Lower	0.146	0.205	0.180	22.148	12517
177	Fairview	0.088	0.178	0.180	21.804	249170
178	Chesterfield Avenue	0.092	0.158	0.180	30.052	88392
179	Ranelagh Road	0.072	0.207	0.180	19.876	9380
180	Copeland Grove	0.138	0.196	0.181	14.774	182
181	Collins Avenue West	0.084	0.216	0.181	18.990	5440
182	R148	0.075	0.165	0.181	22.739	4690
183	Temple Street	0.077	0.175	0.182	19.200	1625
184	R105	0.095	0.186	0.182	18.851	49360

185	Church Lane	0.069	0.150	0.182	13.849	1386
186	Earlsfort Terrace	0.098	0.197	0.183	20.073	10498
187	Burgh Quay	0.093	0.167	0.183	20.539	2413
188	Saint Lawrence Road	0.096	0.203	0.183	16.869	369
189	Mount Merrion Avenue	0.128	0.234	0.184	16.349	15383
190	Sandymount Avenue	0.105	0.194	0.184	23.862	816
191	Lagan Road	0.079	0.216	0.185	17.859	4997
192	Arran Quay	0.088	0.151	0.185	26.611	13570
193	Marino Mart	0.090	0.184	0.185	19.248	215260
194	Carrickbrack Road	0.103	0.198	0.185	29.095	27535
195	Richmond Street South	0.117	0.209	0.186	17.467	14594
196	Macken Street	0.462	0.197	0.187	18.467	3025
197	Abbey Street Lower	0.109	0.191	0.187	21.785	264
198	West Terrace	0.098	0.158	0.187	23.609	168
199	Blackheath Park	0.088	0.201	0.187	23.059	32932
200	Belgard Road	0.073	0.202	0.187	20.357	16847
201	Ashington Mews	0.070	0.217	0.188	15.447	220
202	Nassau Street	0.118	0.240	0.188	18.525	2103
203	Thormanby Road	0.097	0.195	0.189	15.250	1795
204	Saint Patrick's Terrace	0.088	0.135	0.189	22.808	78
205	Foster Place	0.078	0.217	0.190	15.823	999
206	Patrick Street	0.081	0.221	0.190	21.753	25316
207	Anglesea Street	0.093	0.219	0.190	17.471	724
208	Alfie Byrne Road	0.100	0.200	0.191	23.921	225599
209	Tallaght Road	0.080	0.232	0.191	24.496	67763
210	Cathal Brugha Street	0.111	0.165	0.193	17.918	4080
211	Cookstown Road	0.071	0.245	0.193	20.056	9153
212	Georgian Village	0.073	0.161	0.194	17.038	585
213	Pelletstown Manor	0.067	0.211	0.195	21.879	10108
214	Gardiner Street Upper	0.101	0.176	0.195	20.027	5307
215	N81	0.078	0.246	0.196	24.335	136635
216	Rock Road	0.127	0.257	0.197	23.081	12908
217	Main Street	0.081	0.220	0.197	21.806	150770
218	Grand Parade	0.108	0.199	0.198	24.973	28975
219	Inchicore Terrace South	0.079	0.169	0.198	19.877	713
220	Bramblefield Drive	0.087	0.225	0.198	19.396	3491
221	Bray Road	0.103	0.224	0.199	22.498	26885
222	Clinchs Court	0.103	0.171	0.199	25.549	301
223	Harcourt Street	0.115	0.243	0.199	15.752	3546
224	Beatty's Avenue	0.098	0.204	0.199	22.603	14664
225	N11	0.118	0.242	0.200	20.473	57089
226	North Wall Quay	0.099	0.234	0.200	10.594	409

227	Eden Quay	0.085	0.205	0.200	20.544	24891
228	Church Street	0.142	0.256	0.201	13.599	2359
229	Grosvenor Court	0.108	0.196	0.201	16.472	433
230	Fortfield Road	0.088	0.252	0.202	18.948	783
231	South Lotts Road	0.121	0.247	0.202	15.667	3774
232	Fitzwilliam Square East	0.114	0.219	0.202	27.341	5393
233	Annesley Bridge Road	0.100	0.201	0.202	19.118	300153
234	Sean Macdermott Street Lower	0.127	0.177	0.202	20.354	7070
235	Sheriff Street Lower	0.108	0.184	0.203	21.035	204
236	Lincoln Place	0.102	0.222	0.203	17.553	16136
237	Laurence Brook	0.058	0.154	0.203	14.401	9175
238	Camden Street Lower	0.074	0.223	0.204	20.133	4285
239	Sarsfield Quay	0.116	0.184	0.204	27.426	3958
240	Trinity Terrace	0.088	0.226	0.204	20.070	1008
241	Wyattville Road	0.096	0.227	0.206	21.311	410
242	Merrion Square North	0.241	0.240	0.206	17.077	4317
243	Royal Canal Way	0.075	0.223	0.206	20.195	43332
244	Eccles Street	0.127	0.266	0.207	11.758	2042
245	Maiden's Row	0.064	0.163	0.209	15.689	20153
246	Balglass Road	0.121	0.237	0.209	10.410	2834
247	Wellington Lane	0.091	0.263	0.209	26.495	1305
248	Belgard Square North	0.077	0.246	0.209	21.212	55384
249	Grove Road	0.100	0.224	0.210	24.320	65255
250	Westbourne Road	0.094	0.220	0.210	13.757	15101
251	Drumcondra Road Lower	0.093	0.226	0.210	19.348	47088
252	Ballsbridge Park	0.085	0.203	0.211	15.785	29045
253	Rathfarnham Road	0.106	0.218	0.211	19.646	57940
254	Leinster Street South	0.107	0.228	0.212	16.066	2576
255	BÃşthar Ghort Na nGrÃąs	0.130	0.169	0.212	17.591	1288
256	Canon Place	0.085	0.193	0.214	16.434	3198
257	East Wall Road	0.105	0.228	0.214	24.771	284158
258	Kilgobbet Grove	0.107	0.238	0.215	23.140	52
259	Charlemont Street	0.078	0.252	0.215	18.942	5476
260	Irishtown Road	0.109	0.232	0.215	24.908	93024
261	Mount Drummond Avenue	0.101	0.241	0.215	25.585	1370
262	Talbot Court	0.085	0.237	0.216	22.716	16600
263	Prince's Street South	0.078	0.194	0.216	13.738	1642
264	The Stiles Road	0.100	0.210	0.216	16.424	19508
265	Templeogue Road	0.079	0.256	0.216	22.867	237823
266	Bass Place	0.116	0.219	0.216	21.108	273
267	Westland Row	0.110	0.243	0.216	14.891	25211

268	South Great George's Street	0.093	0.270	0.217	16.715	6353
269	Collins Avenue Extension	0.075	0.240	0.217	15.103	10063
270	Harold's Cross Road	0.098	0.239	0.217	24.409	270595
271	Fownes Street Upper	0.090	0.186	0.218	20.790	267
272	Prospect Road	0.099	0.260	0.219	15.918	3422
273	Brookwood Park	0.090	0.174	0.219	22.399	1420
274	Exchange Place	0.134	0.296	0.219	13.552	76
275	Grand Canal Street Upper	0.423	0.263	0.219	14.271	3333
276	Brookwood Avenue	0.103	0.187	0.220	21.367	249594
277	Balkill Road	0.106	0.227	0.220	11.505	15249
278	Royal Canal Avenue	0.083	0.249	0.222	20.061	136154
279	Stillorgan Road	0.119	0.259	0.222	20.559	100795
280	Knockmaroon Hill	0.085	0.248	0.222	13.231	8591
281	Martin's Row	0.065	0.171	0.222	15.545	23748
282	Gardiner Street Middle	0.117	0.210	0.223	16.876	3704
283	Whitestown Park	0.090	0.249	0.223	21.723	90146
284	Clontarf Road	0.100	0.242	0.224	22.827	1729772
285	Lucan Road	0.085	0.165	0.224	18.186	7023
286	Court View	0.068	0.261	0.225	25.851	180
287	Seafield Avenue	0.090	0.203	0.225	21.177	5249
288	Cope Street	0.088	0.195	0.225	21.268	68
289	Clonliffe Road	0.083	0.257	0.225	22.285	171421
290	City Quay	0.115	0.233	0.225	22.418	31540
291	Camden Street Upper	0.095	0.245	0.226	22.707	1000
292	North Road	0.099	0.181	0.226	18.585	2127
293	Brookwood Glen	0.058	0.219	0.227	18.432	100
294	R132	0.120	0.185	0.228	22.370	512
295	R811	0.119	0.251	0.228	19.748	10996
296	Mayor Street Lower	0.138	0.353	0.228	11.439	1880
297	Trim Road	0.159	0.256	0.229	20.038	14794
298	Watermill Road	0.098	0.167	0.229	20.663	1758
299	Merrion Square South	0.120	0.259	0.229	21.189	21647
300	Bayview Avenue	0.109	0.224	0.230	25.220	5892
301	Sheepmoor Grove	0.096	0.266	0.230	21.398	44947
302	Whitworth Road	0.089	0.255	0.230	21.111	152008
303	Belgrove Park	0.073	0.181	0.231	12.822	8235
304	Bramblefield Court	0.084	0.261	0.232	21.854	1146
305	BÃşthar Mhuirfean	0.234	0.343	0.232	19.402	51
306	Bridge Street	0.123	0.245	0.232	19.488	24004
307	Dublin Road	0.113	0.233	0.232	23.934	34863
308	Killarney Street	0.162	0.255	0.233	14.608	678

309	Dunseverick Road	0.101	0.213	0.233	16.726	13839
310	Lord Edward Street	0.104	0.238	0.233	19.992	36639
311	Stiles Court	0.118	0.249	0.233	17.574	846
312	Christchurch Place	0.097	0.234	0.233	17.965	31531
313	Huntstown Wood	0.090	0.268	0.233	19.215	47258
314	Rathborne Drive	0.077	0.259	0.233	21.872	9950
315	Shaw Street	0.111	0.242	0.234	16.534	36514
316	Kincora Road	0.120	0.229	0.235	19.915	161151
317	Lombard Street East	0.107	0.225	0.235	18.543	50636
318	Merrion Square West	0.105	0.244	0.235	19.777	19992
319	Belgard Square East	0.082	0.306	0.235	20.776	719
320	Inchicore Square East	0.088	0.178	0.235	23.769	428
321	Kincora Drive	0.081	0.202	0.236	17.922	75
322	O'Connell Street Lower	0.114	0.219	0.236	17.432	2555
323	Eustace Street	0.074	0.173	0.237	17.553	108
324	Castlekevin Road	0.106	0.188	0.237	19.569	26972
325	Grattan Crescent	0.101	0.197	0.238	18.507	10820
326	Dolphin Road	0.120	0.241	0.238	25.516	13911
327	Roebuck Castle	0.119	0.223	0.238	12.441	2678
328	M50	0.093	0.282	0.238	24.286	11155
329	Snugborough Road	0.090	0.284	0.239	21.176	123722
330	Dame Street	0.095	0.240	0.240	19.930	61896
331	O'Hara Avenue	0.117	0.229	0.240	28.535	667
332	Inchicore Road	0.094	0.203	0.240	21.032	71642
333	Drumcondra Road Upper	0.125	0.196	0.242	18.640	8664
334	Kincora Grove	0.097	0.261	0.242	19.312	5653
335	R121	0.100	0.276	0.242	22.225	44411
336	Mountjoy Square West	0.104	0.219	0.243	17.615	2508
337	Portland Row	0.107	0.217	0.243	18.513	1733
338	Thomas Street	0.105	0.206	0.244	23.370	55387
339	Buckingham Street Upper	0.134	0.232	0.244	15.065	870
340	Adelaide Road	0.118	0.264	0.244	19.342	10631
341	Copeland Avenue	0.103	0.232	0.246	14.742	73
342	Rathborne Close	0.077	0.273	0.246	22.479	1075
343	Castleview	0.067	0.214	0.246	15.201	1032
344	Parnell Road	0.130	0.263	0.246	28.024	15493
345	Brookwood Grove	0.113	0.247	0.247	17.507	2548
346	Gardiner Street Lower	0.108	0.264	0.247	21.210	13290
347	Castle Avenue	0.113	0.250	0.248	17.631	165224
348	Collins Avenue East	0.088	0.300	0.248	19.857	8581
349	Huntstown Green	0.102	0.269	0.248	17.237	11959
350	Lansdowne Road	0.095	0.233	0.248	14.051	42226

351	Parkgate Street	0.100	0.205	0.249	25.394	5704
352	Parc Na Silla Rise	0.143	0.348	0.249	18.076	52
353	Sandwith Street Lower	0.141	0.238	0.250	15.089	4423
354	Inchicore Terrace North	0.101	0.208	0.250	24.385	22829
355	R808	0.125	0.274	0.250	20.236	308
356	Moss Street	0.108	0.241	0.251	20.885	71118
357	Yellow Road	0.107	0.177	0.251	23.290	360
358	East-link Toll Bridge	0.130	0.270	0.251	23.716	29778
359	South Circular Road	0.122	0.260	0.251	18.490	18828
360	Ranelagh	0.092	0.286	0.252	18.585	8215
361	Rathborne Avenue	0.088	0.274	0.252	19.670	4099
362	Westmoreland Street	0.102	0.282	0.252	17.257	15064
363	Phelan Avenue	0.109	0.275	0.253	21.465	152
364	Fenian Street	0.124	0.256	0.253	20.436	2239
365	College Green	0.101	0.246	0.253	18.321	52994
366	Parnell Street	0.118	0.252	0.253	15.538	8243
367	Terenure Road East	0.121	0.257	0.253	20.443	28702
368	Terenure Road North	0.103	0.279	0.254	24.655	85772
369	Anglesea Road	0.118	0.278	0.254	26.961	13889
370	Old Navan Road	0.096	0.296	0.254	20.583	55780
371	North Circular Road	0.103	0.277	0.255	21.455	6442
372	Belgrove Road	0.108	0.236	0.255	16.979	40286
373	Castle Grove	0.100	0.228	0.255	18.500	118673
374	R113	0.084	0.263	0.257	20.924	35058
375	Merrion Street Upper	0.115	0.263	0.257	17.252	11971
376	Thomas Street West	0.101	0.220	0.258	23.083	20609
377	Huntstown Drive	0.080	0.251	0.258	19.605	10222
378	Newtown Road	0.110	0.250	0.258	15.679	1002
379	Seapark Road	0.133	0.294	0.258	20.371	32006
380	Vernon Avenue	0.108	0.234	0.258	19.174	733715
381	Fergus Road	0.110	0.244	0.259	21.865	1241
382	Empress Place	0.131	0.224	0.260	21.862	133
383	Herbert Road	0.106	0.252	0.260	13.866	14414
384	Bramblefield View	0.109	0.323	0.261	22.034	427
385	Collins Avenue	0.103	0.292	0.261	19.101	19099
386	Petty Lane	0.108	0.250	0.262	17.467	643
387	Rathfarnham Wood	0.075	0.222	0.262	14.557	5459
388	Dorset Street Lower	0.105	0.238	0.262	17.633	5215
389	Nutley Lane	0.122	0.278	0.262	23.034	196480
390	Kilmore Road	0.119	0.212	0.262	21.413	60725
391	Mark Street	0.128	0.308	0.263	13.686	2254
392	College Street	0.103	0.249	0.263	16.394	21496

393	Grand Canal Street Lower	0.300	0.273	0.264	21.206	3210
394	Ratoath Estate	0.093	0.308	0.264	20.370	21987
395	Martin Savage Park	0.099	0.296	0.267	17.391	36423
396	Seafield Road West	0.115	0.274	0.269	23.930	20542
397	Bushy Park Road	0.131	0.280	0.270	29.331	4218
398	Ardlea Road	0.107	0.205	0.271	21.233	107283
399	Hogan Place	0.133	0.281	0.272	18.600	1330
400	Ringsend Road	0.154	0.343	0.273	21.428	3208
401	Tower Road	0.089	0.211	0.273	17.215	12528
402	Tritonville Road	0.127	0.289	0.274	22.791	40285
403	Sybil Hill Road	0.115	0.241	0.274	22.356	208821
404	Temple Lane North	0.107	0.283	0.274	19.346	487
405	Hardwicke Place	0.099	0.273	0.274	17.299	225
406	Arranmore Avenue	0.135	0.306	0.274	10.995	449
407	Orwell Road	0.123	0.296	0.275	20.852	4085
408	Exchange Street Upper	0.142	0.239	0.276	18.314	138
409	Albert College Avenue	0.075	0.302	0.276	12.917	1871
410	Bow Lane West	0.091	0.215	0.276	22.694	37667
411	Cambridge Park	0.135	0.356	0.276	24.596	153
412	Nerney's Court	0.108	0.258	0.276	17.968	75
413	Knock Riada	0.071	0.219	0.278	15.532	680
414	Vernon Court	0.114	0.276	0.278	24.904	456
415	Dollymount Avenue	0.136	0.319	0.279	17.560	33829
416	O'Connell Bridge	0.108	0.295	0.281	18.185	6325
417	Creighton Street	0.125	0.279	0.281	20.621	4098
418	Redmond's Hill	0.109	0.347	0.281	20.873	1588
419	Gracefield Road	0.109	0.226	0.281	19.863	48366
420	R827	0.101	0.331	0.284	20.883	102
421	Thorncastle Street	0.121	0.289	0.284	21.845	84705
422	Saint Patrick's Villas	0.128	0.278	0.286	22.294	10726
423	Fitzwilliam Street Upper	0.121	0.288	0.287	20.669	17679
424	Oulton Road	0.143	0.296	0.288	18.874	47512
425	Mark's Lane	0.145	0.273	0.288	11.677	344
426	Fleet Street	0.095	0.258	0.289	16.965	1146
427	Talbot Downs	0.107	0.365	0.290	20.364	485
428	R111	0.143	0.309	0.290	24.891	1568
429	Suir Road	0.115	0.276	0.291	24.565	4224
430	R109	0.109	0.233	0.291	24.724	7017
431	Huntstown Park	0.110	0.324	0.291	18.354	3617
432	Saint Ignatius Avenue	0.117	0.343	0.291	15.316	488
433	Sheriff Street Upper	0.145	0.320	0.293	29.455	671
434	Archers Wood	0.124	0.317	0.293	22.667	15336

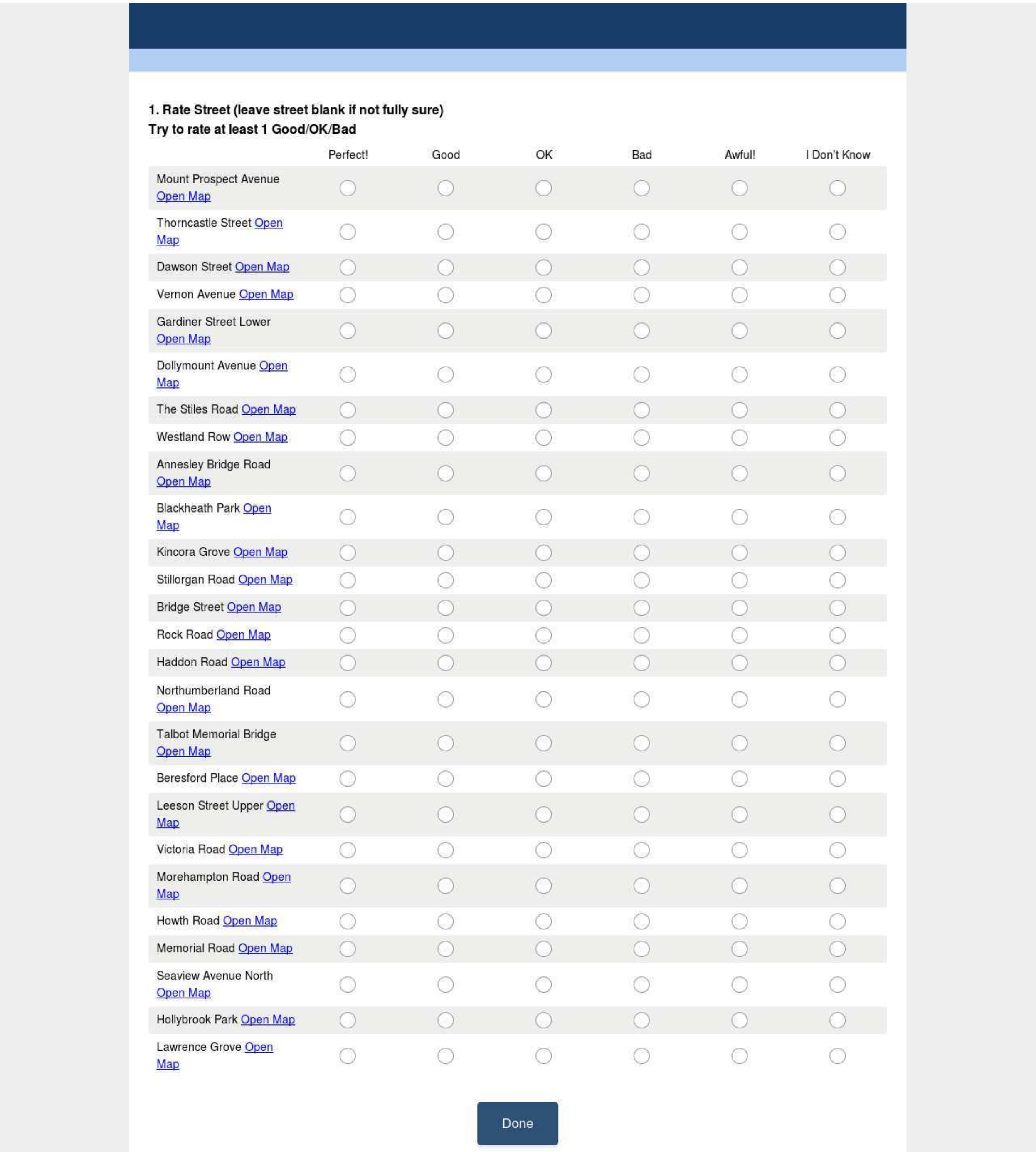
435	Fitzwilliam Lane	0.121	0.288	0.293	24.983	462
436	Saint Laurence's Road	0.098	0.225	0.293	23.749	74596
437	Meath Street	0.139	0.240	0.294	22.392	655
438	Huntstown Way	0.102	0.312	0.294	20.947	59625
439	Hill Street	0.108	0.297	0.294	22.921	2176
440	Nutley Park	0.115	0.304	0.294	30.298	103
441	Dawson Street	0.131	0.323	0.295	17.479	3630
442	Saint Ignatius Road	0.088	0.330	0.295	20.662	2422
443	Kincora Avenue	0.124	0.310	0.296	17.043	2962
444	Zion Road	0.134	0.308	0.296	21.588	3688
445	Thornberry Square	0.104	0.349	0.296	20.993	8479
446	Ballyshannon Road	0.139	0.265	0.296	19.865	21694
447	Summerfield Rise	0.085	0.347	0.299	19.773	504
448	Wellington Quay	0.131	0.339	0.299	24.957	53
449	D'Olier Street	0.106	0.266	0.301	18.094	417
450	Glengarriff Parade	0.110	0.338	0.301	22.852	1678
451	Poolbeg Street	0.115	0.310	0.304	18.215	1035
452	R803	0.113	0.222	0.307	14.093	262
453	Sandwith Street Upper	0.123	0.305	0.307	18.511	1790
454	Magennis Place	0.127	0.261	0.308	15.628	1577
455	Saint Andrew's Street	0.099	0.376	0.308	13.041	803
456	Ballyboggan Road	0.094	0.363	0.308	18.571	22920
457	Seafield Road East	0.130	0.308	0.308	21.676	17884
458	Greenfield Park	0.126	0.321	0.309	18.857	161391
459	Aungier Street	0.101	0.362	0.310	18.835	5718
460	Fitzwilliam Street Lower	0.123	0.309	0.310	21.349	6499
461	Park Lane East	0.083	0.175	0.310	21.525	94
462	Terenure Place	0.105	0.345	0.311	18.435	8442
463	Luke Kelly Bridge	0.125	0.364	0.311	16.925	21178
464	Knockmaroon Road	0.121	0.202	0.316	12.353	82
465	Kilbarron Road	0.134	0.256	0.317	21.896	39397
466	Nottingham Street	0.118	0.336	0.318	24.691	76
467	Mount Prospect Avenue	0.128	0.317	0.319	18.768	227396
468	Fitzwilliam Street	0.142	0.321	0.323	22.564	641
469	Greenfield Crescent	0.129	0.353	0.324	25.443	16795
470	Fitzwilliam Place	0.149	0.337	0.324	25.762	14133
471	Newbridge Avenue	0.139	0.345	0.326	21.847	62041
472	Dean Street	0.130	0.405	0.327	17.349	402
473	Rathborne Way	0.097	0.355	0.328	20.605	567
474	Pinewood Court	0.118	0.380	0.328	18.709	1668
475	Hawkins Street	0.115	0.291	0.329	16.644	9043
476	Carnlough Road	0.101	0.408	0.330	22.629	941

477	R131	0.138	0.339	0.331	20.939	14296
478	Bow Lane East	0.101	0.428	0.333	20.257	255
479	Fairview Strand	0.113	0.366	0.333	18.288	72251
480	Conquer Hill Road	0.139	0.279	0.334	24.162	2359
481	Castle Road	0.147	0.389	0.336	16.803	957
482	Charlotte Way	0.136	0.457	0.337	16.049	1660
483	Ratoath Road	0.108	0.399	0.338	19.135	75219
484	York Street	0.097	0.484	0.338	17.390	127
485	Cumberland Street South	0.120	0.329	0.347	18.457	2998
486	Castleside Drive	0.080	0.291	0.351	15.342	501
487	Holles Street	0.122	0.369	0.352	24.856	1287
488	Ashfield Court	0.154	0.391	0.359	21.212	232
489	Broombridge Road	0.127	0.416	0.359	18.498	336
490	Chapelizod Road	0.119	0.300	0.361	27.054	29920
491	Saint Gabriel's Road	0.136	0.414	0.362	22.130	1385
492	Trinity Street	0.122	0.429	0.373	15.305	949
493	Alexandra Road	0.153	0.434	0.375	21.632	842
494	Ashfield Way	0.122	0.412	0.378	19.986	2345
495	York Road	0.151	0.372	0.381	15.301	17183
496	Barrow Road	0.136	0.454	0.386	18.163	62
497	Bannow Road	0.112	0.435	0.390	20.341	16344
498	Lee Road	0.117	0.436	0.392	17.643	34840
499	High Street	0.148	0.351	0.396	27.451	13998
500	Cornmarket	0.141	0.330	0.403	26.761	5431
501	Seville Place	0.104	0.324	0.403	20.045	717
502	R814	0.188	0.448	0.412	20.763	1550
503	Coburg Place	0.096	0.306	0.414	23.337	103
504	Dollymount Park	0.150	0.410	0.428	15.707	50440
505	R817	0.152	0.490	0.436	18.227	253
506	Saint Brendan's Avenue	0.158	0.468	0.438	19.128	51
507	Con Colbert Road	0.152	0.394	0.444	23.686	52
508	Winetavern Street	0.148	0.474	0.445	18.540	151
509	Denzille Lane	0.140	0.409	0.450	19.367	1754
510	Ashfield Lawn	0.111	0.514	0.463	22.613	270
511	Boyne Street	0.161	0.354	0.470	20.705	111
512	Bow Bridge	0.147	0.376	0.473	22.957	5016
513	Rathdown Drive	0.133	0.744	0.483	15.751	102
514	Royal Canal Way, Glasnevin	0.128	0.582	0.490	19.121	68254
515	Back Lane	0.150	0.381	0.500	26.105	503
516	Pembroke Cottages	0.229	0.523	0.523	12.022	185
517	Suffolk Street	0.197	0.704	1.147	11.726	1889

518	Wicklow Street	0.500	0.905	2.019	9.225	540

Table F.2: Order of roads by mean Z-axis acceleration

F.3 Road Rating Survey



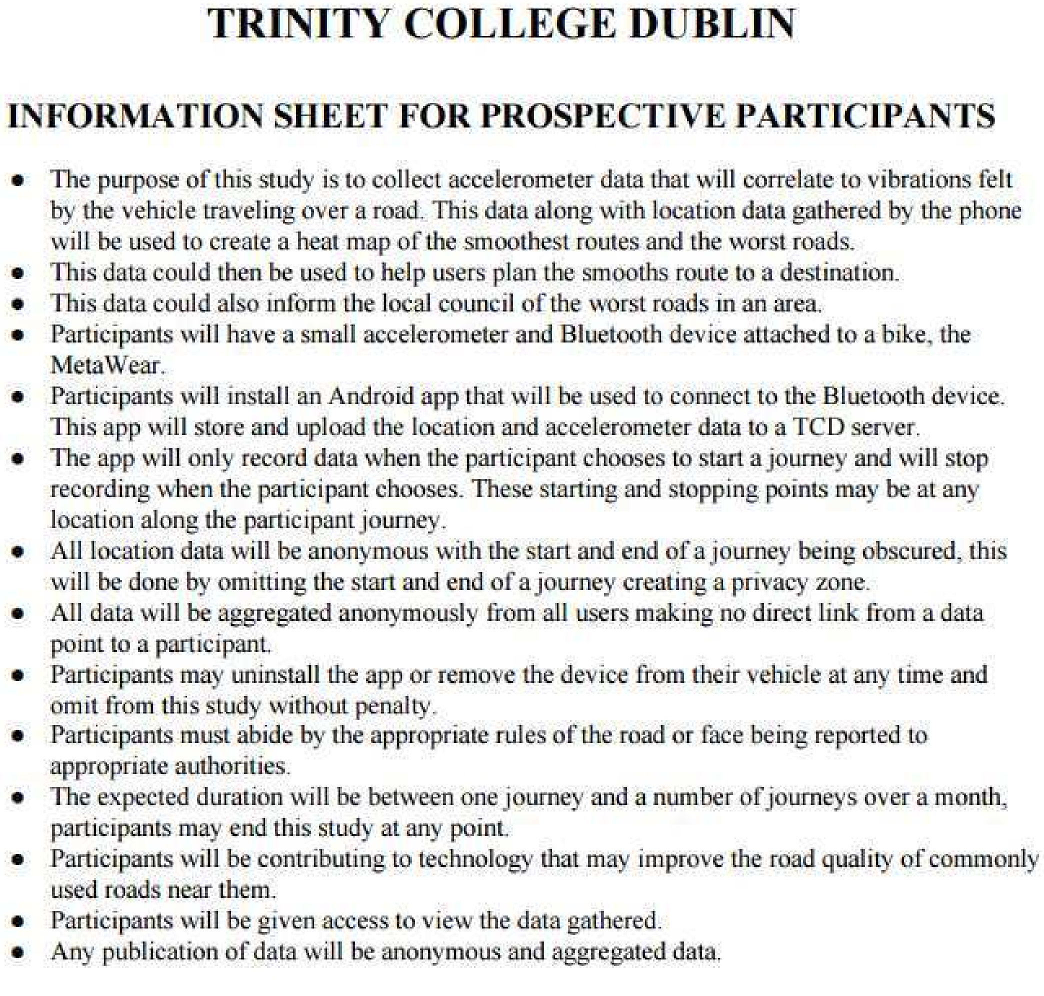
1. Rate Street (leave street blank if not fully sure)
Try to rate at least 1 Good/OK/Bad

	Perfect!	Good	OK	Bad	Awful!	I Don't Know
Mount Prospect Avenue Open Map	○	○	○	○	○	○
Thorncastle Street Open Map	○	○	○	○	○	○
Dawson Street Open Map	○	○	○	○	○	○
Vernon Avenue Open Map	○	○	○	○	○	○
Gardiner Street Lower Open Map	○	○	○	○	○	○
Dollymount Avenue Open Map	○	○	○	○	○	○
The Stiles Road Open Map	○	○	○	○	○	○
Westland Row Open Map	○	○	○	○	○	○
Annesley Bridge Road Open Map	○	○	○	○	○	○
Blackheath Park Open Map	○	○	○	○	○	○
Kincora Grove Open Map	○	○	○	○	○	○
Stillorgan Road Open Map	○	○	○	○	○	○
Bridge Street Open Map	○	○	○	○	○	○
Rock Road Open Map	○	○	○	○	○	○
Haddon Road Open Map	○	○	○	○	○	○
Northumberland Road Open Map	○	○	○	○	○	○
Talbot Memorial Bridge Open Map	○	○	○	○	○	○
Beresford Place Open Map	○	○	○	○	○	○
Leeson Street Upper Open Map	○	○	○	○	○	○
Victoria Road Open Map	○	○	○	○	○	○
Morehampton Road Open Map	○	○	○	○	○	○
Howth Road Open Map	○	○	○	○	○	○
Memorial Road Open Map	○	○	○	○	○	○
Seaview Avenue North Open Map	○	○	○	○	○	○
Hollybrook Park Open Map	○	○	○	○	○	○
Lawrence Grove Open Map	○	○	○	○	○	○

Figure F.1: Participants Road Rating Survey

Appendix G

Ethics

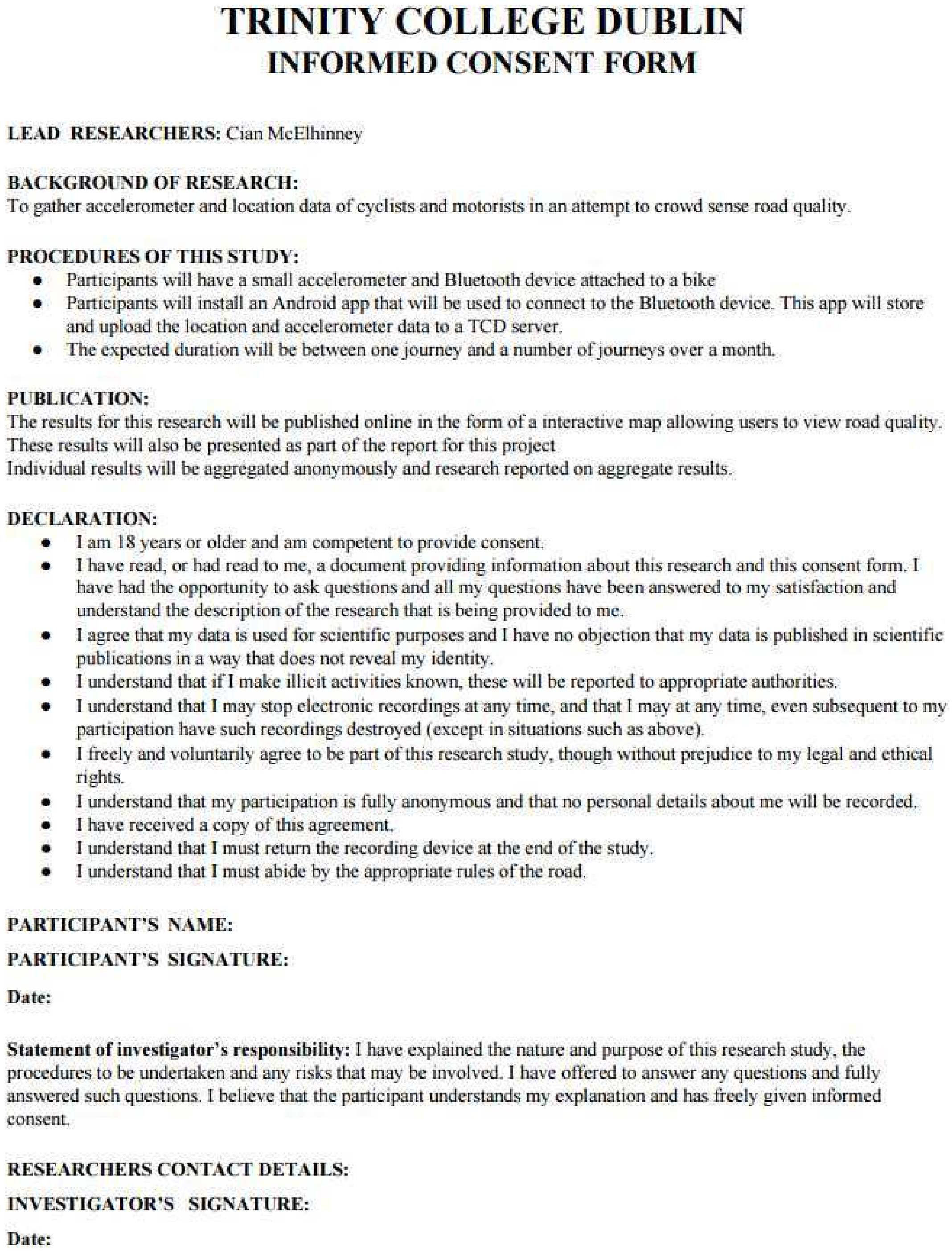


TRINITY COLLEGE DUBLIN

INFORMATION SHEET FOR PROSPECTIVE PARTICIPANTS

- The purpose of this study is to collect accelerometer data that will correlate to vibrations felt by the vehicle traveling over a road. This data along with location data gathered by the phone will be used to create a heat map of the smoothest routes and the worst roads.
- This data could then be used to help users plan the smooths route to a destination.
- This data could also inform the local council of the worst roads in an area.
- Participants will have a small accelerometer and Bluetooth device attached to a bike, the MetaWear.
- Participants will install an Android app that will be used to connect to the Bluetooth device. This app will store and upload the location and accelerometer data to a TCD server.
- The app will only record data when the participant chooses to start a journey and will stop recording when the participant chooses. These starting and stopping points may be at any location along the participant journey.
- All location data will be anonymous with the start and end of a journey being obscured, this will be done by omitting the start and end of a journey creating a privacy zone.
- All data will be aggregated anonymously from all users making no direct link from a data point to a participant.
- Participants may uninstall the app or remove the device from their vehicle at any time and omit from this study without penalty.
- Participants must abide by the appropriate rules of the road or face being reported to appropriate authorities.
- The expected duration will be between one journey and a number of journeys over a month, participants may end this study at any point.
- Participants will be contributing to technology that may improve the road quality of commonly used roads near them.
- Participants will be given access to view the data gathered.
- Any publication of data will be anonymous and aggregated data.

Figure G.1: Information sheet given to participants



TRINITY COLLEGE DUBLIN
INFORMED CONSENT FORM

LEAD RESEARCHERS: Cian McElhinney

BACKGROUND OF RESEARCH:
To gather accelerometer and location data of cyclists and motorists in an attempt to crowd sense road quality.

PROCEDURES OF THIS STUDY:

- Participants will have a small accelerometer and Bluetooth device attached to a bike
- Participants will install an Android app that will be used to connect to the Bluetooth device. This app will store and upload the location and accelerometer data to a TCD server.
- The expected duration will be between one journey and a number of journeys over a month.

PUBLICATION:
The results for this research will be published online in the form of a interactive map allowing users to view road quality.
These results will also be presented as part of the report for this project
Individual results will be aggregated anonymously and research reported on aggregate results.

DECLARATION:

- I am 18 years or older and am competent to provide consent.
- I have read, or had read to me, a document providing information about this research and this consent form. I have had the opportunity to ask questions and all my questions have been answered to my satisfaction and understand the description of the research that is being provided to me.
- I agree that my data is used for scientific purposes and I have no objection that my data is published in scientific publications in a way that does not reveal my identity.
- I understand that if I make illicit activities known, these will be reported to appropriate authorities.
- I understand that I may stop electronic recordings at any time, and that I may at any time, even subsequent to my participation have such recordings destroyed (except in situations such as above).
- I freely and voluntarily agree to be part of this research study, though without prejudice to my legal and ethical rights.
- I understand that my participation is fully anonymous and that no personal details about me will be recorded.
- I have received a copy of this agreement.
- I understand that I must return the recording device at the end of the study.
- I understand that I must abide by the appropriate rules of the road.

PARTICIPANT'S NAME:

PARTICIPANT'S SIGNATURE:

Date:

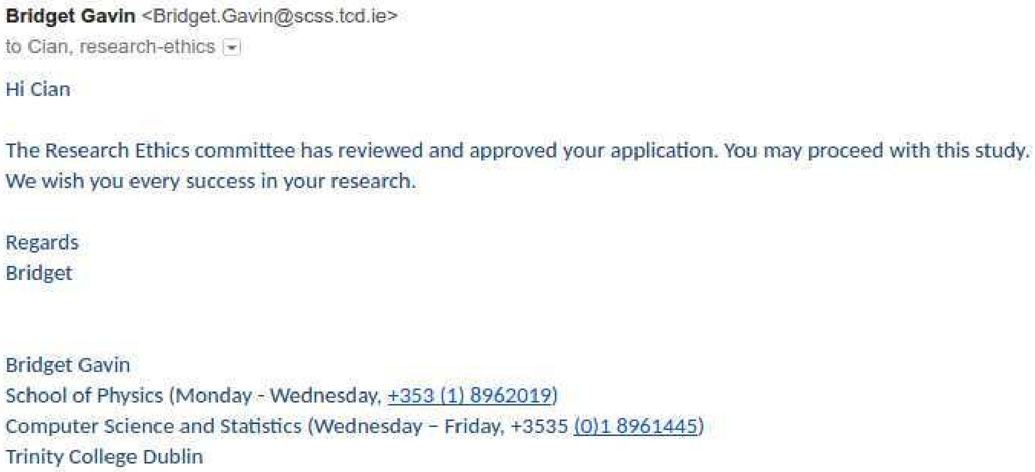
Statement of investigator's responsibility: I have explained the nature and purpose of this research study, the procedures to be undertaken and any risks that may be involved. I have offered to answer any questions and fully answered such questions. I believe that the participant understands my explanation and has freely given informed consent.

RESEARCHERS CONTACT DETAILS:

INVESTIGATOR'S SIGNATURE:

Date:

Figure G.2: Consent form signed by participants



Bridget Gavin <Bridget.Gavin@scss.tcd.ie>
to Cian, research-ethics

Hi Cian

The Research Ethics committee has reviewed and approved your application. You may proceed with this study.
We wish you every success in your research.

Regards
Bridget

Bridget Gavin
School of Physics (Monday - Wednesday, +353 (1) 8962019)
Computer Science and Statistics (Wednesday – Friday, +3535 (0)1 8961445)
Trinity College Dublin

Figure G.3: Confirmation of ethical approval

Bibliography

[1] *3D for Everyone | SketchUp*. http://www.sketchup.com/. (Accessed on 04/30/2016).

[2] *Amazon Simple Storage Service (S3) - Cloud Storage*. https://aws.amazon.com/s3/. (Accessed on 04/29/2016).

[3] *Android Asynchronous Http Client*. http://loopj.com/android-async-http/. (Accessed on 05/04/2016).

[4] *Beeline - navigate your city, your way*. https://beeline.co/. (Accessed on 05/16/2016).

[5] *Bicycle Accidents in Ireland*. 2008. URL: http://www.injury-compensation.ie/bicycle-accidents-ireland/ (visited on 03/20/2016).

[6] Theodora S Brisimi et al. “Sensing and classifying roadway obstacles: The street bump anomaly detection and decision support system”. In: *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*. IEEE. 2015, pp. 1288–1293.

[7] Data Protection Commissioner. “Data Protection Acts 1988 & 2003-Informal Consolidation”. In: *Web accessed May* (2009).

[8] National Transport Authority Dublin City Council. *Report on trends in mode share of vehicles and people crossing the Canal Cordon 2006 to 2014*. http://www.dublincity.ie/sites/default/files/content/RoadsandTraffic/Traffic/Documents/DownloadThe2014Cor pdf. (Accessed on 05/17/2016). 2014.

[9] *Edge Explore 1000 | Garmin*. https://buy.garmin.com/en-US/US/into-sports/cycling/edge-explore-1000/prod522791.html. (Accessed on 05/16/2016).

[10] Jakob Eriksson et al. “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*. ACM. 2008, pp. 29–39.

[11] *Flask âĂŤ Flask Documentation (0.10)*. http://flask.pocoo.org/docs/0.10/. (Accessed on 05/05/2016).

[12] *GraphHopper: Open source routing library and server using OpenStreetMap #route #planner*. https://github.com/graphhopper/graphhopper. (Accessed on 05/10/2016).

[13] Trevor Hastie et al. “The elements of statistical learning: data mining, inference and prediction”. In: *The Mathematical Intelligencer* 27.2 (2005), pp. 83–85.

[14] *Leaflet/Leaflet.heat: A tiny, simple and fast heatmap plugin for Leaflet.* https://github.com/Leaflet/Leaflet.heat. (Accessed on 05/10/2016).

[15] Lin Liao et al. “Learning and inferring transportation routines”. In: *Artificial Intelligence* 171.5 (2007), pp. 311–331.

[16] *Lumos Helmet - A Next Generation Bicycle Helmet.* https://lumoshelmet.co/. (Accessed on 05/16/2016).

[17] *Material design - Google design guidelines.* https://www.google.com/design/spec/material-design/introduction.html. (Accessed on 05/02/2016).

[18] *MbientLab | MetaWear R series.* https://mbientlab.com/metawearr/. (Accessed on 04/29/2016).

[19] *MetaWear Android API âĂŤ MetaWear Android API 2.5.9 documentation.* https://mbientlab.com/androiddocs/latest/. (Accessed on 05/03/2016).

[20] *Model Repair Service.* https://modelrepair.azurewebsites.net/. (Accessed on 04/30/2016).

[21] Vivek Nallur, Amal Elgammal, and Siobhán Clarke. “Smart route planning using open data and participatory sensing”. In: *Open Source Systems: Adoption and Impact.* Springer, 2015, pp. 91–100.

[22] *Node - OpenStreetMap Wiki.* http://wiki.openstreetmap.org/wiki/Node. (Accessed on 05/15/2016).

[23] *Noke U-Lock | Keyless Bluetooth Smart Bike Lock | Smartphone Security.* http://noke.com/pages/ulock. (Accessed on 05/16/2016).

[24] *onomo.* http://www.onomo.net/. (Accessed on 05/16/2016).

[25] *OpenSCAD - Documentation.* http://www.openscad.org/documentation.html. (Accessed on 04/30/2016).

[26] *PowerPod.* http://www.powerpodsports.com/. (Accessed on 05/16/2016).

[27] *Rechargeable lithium polymer battery catalogue from PowerStream.* http://www.powerstream.com/li-pol.htm. (Accessed on 05/12/2016).

[28] Sasank Reddy et al. “Biketastic: sensing and mapping for better biking”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM. 2010, pp. 1817–1820.

[29] *Redis.* http://redis.io/documentation. (Accessed on 04/30/2016).

[30] *Saving Key-Value Sets | Android Developers.* http://developer.android.com/training/basics/data-storage/shared-preferences.html. (Accessed on 05/03/2016).

[31] *See Sense ICON.* http://seesense.cc/features/. (Accessed on 05/16/2016).

[32] *Service | Android Developers.* http://developer.android.com/reference/android/app/Service.html. (Accessed on 05/04/2016).

[33] *SiteWhere*. http://www.sitewhere.org/documentation/overview/. (Accessed on 05/05/2016).

[34] *Snackbar | Android Developers*. http://developer.android.com/reference/android/support/design/widget/Snackbar.html. (Accessed on 05/04/2016).

[35] *Snap to Roads | Google Maps Roads API*. https://developers.google.com/maps/documentation/roads/snap. (Accessed on 05/06/2016).

[36] Helena Stigson, Maria Krafft, and Claes Tingvall. “Use of fatal real-life crashes to analyze a safe road transport system model, including the road user, the vehicle, and the road”. In: *Traffic Injury Prevention* 9.5 (2008), pp. 463–471.

[37] *Street Bump*. 2011. URL: http://www.streetbump.org (visited on 03/20/2016).

[38] *Street Bump: Crowdsourcing Better Streets, but Many Roadblocks Remain âĂŞ Digital Innovation and Transformation*. https://digit.hbs.org/submission/street-bump-crowdsourcing-better-streets-but-many-roadblocks-remain/. (Accessed on 04/20/2016).

[39] *SurveyMonkey: Add Answers in Bulk*. http://help.surveymonkey.com/articles/en_US/kb/Adding-Answer-Choices-in-Bulk. (Accessed on 05/10/2016).

[40] *The Google Maps Geocoding API | Google Maps Geocoding API*. https://developers.google.com/maps/documentation/geocoding/intro#ReverseGeocoding. (Accessed on 05/04/2016).

[41] *The MongoDB 3.2 Manual âĂŤ MongoDB Manual 3.2*. https://docs.mongodb.org/manual/. (Accessed on 04/30/2016).

[42] Yoshito Tobe et al. “vCity Map: Crowdsensing towards visible cities”. In: *SENSORS, 2014 IEEE*. IEEE. 2014, pp. 17–20.

[43] Inc TranSafety. “Study compares older and younger pedestrian walking speeds”. In: *Road Management & Engineering Journal* (1997).

[44] *Ultimaker Original+*. https://ultimaker.com/en/products/ultimaker-original. (Accessed on 04/30/2016).

[45] *What is G-Code - Ultimaker Wiki*. http://wiki.ultimaker.com/What_is_G-Code. (Accessed on 04/30/2016).

[46] Changqing Zhou et al. “Discovering personal gazetteers: an interactive clustering approach”. In: *Proceedings of the 12th annual ACM international workshop on Geographic information systems*. ACM. 2004, pp. 266–273.