

---

# Satellite Navigation based on traffic within cities

---

Dylan Tyrie-Dron -  
40203045

Submitted in partial fulfilment of  
the requirements of Edinburgh Napier University  
for the Degree of  
BEng (Hons) Software Engineering

School of Computing

November 26, 2018

### **Authorship Declaration**

I, (Dylan Tyrie-Dron), confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work in this dissertation following the School's ethical guidelines.

*Signed:*

*Date:*

*Matriculation no:*

### **General Data Protection Regulation Declaration**

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below one of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

## **Abstract**

## Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
<b>2</b>	<b>Literature Review</b>	<b>8</b>
2.1	Traffic Research . . . . .	8
2.1.1	. . . . .	8
2.1.2	. . . . .	8
2.1.3	. . . . .	8
2.1.4	. . . . .	9
2.2	Google Maps . . . . .	9
2.2.1	Google Maps Elevation API . . . . .	9
2.3	TomTom maps . . . . .	10
2.3.1	Map tiles . . . . .	10
2.3.2	Map layers . . . . .	10
2.3.3	Traffic data . . . . .	11
2.3.4	Traffic data . . . . .	12
2.4	Routing Research . . . . .	12
2.4.1	. . . . .	12
2.4.2	. . . . .	12
2.4.3	. . . . .	12
2.4.4	. . . . .	12
2.4.5	. . . . .	13
2.4.6	. . . . .	13
2.4.7	. . . . .	13
2.4.8	. . . . .	13
2.4.9	. . . . .	14
2.5	Car average speed Research . . . . .	14
2.5.1	Floating car data . . . . .	14
2.6	Traffic sensor Research . . . . .	14
2.6.1	RFID . . . . .	14
	<b>Appendices</b>	<b>15</b>
<b>A</b>	<b>Project Overview</b>	<b>15</b>
<b>B</b>	<b>Diary Sheets (or other project management evidence)</b>	<b>18</b>

## List of Tables

**List of Figures**

1	PathElevationRequest . . . . .	9
2	Zoom level 1 . . . . .	11

## **Acknowledgements**

Insert acknowledgements here

I would like to thank my cat, dog and family.



## 1 Introduction

The aim of this project is to describe a way of navigating through a city through the use of real-time traffic data. This will be achieved by testing different pathfinding algorithms for example: (?). Research will also be carried out on traffic and other data affecting vehicles getting to their destinations faster. This will be done by rigorously testing car speeds at certain times in the same place within the city Edinburgh. This should give data that can input into a pathfinding algorithm. The results will then be compared to similar experiments developed by others as a way of finding out which method was the most accurate, cost-effective and efficient to travel through a city and discussing which methods are the most suitable for getting through any city. The main interface used to convey the outcome of this project is an android application. It is also thought to make use of API's like: "OpenstreetMaps" to speed up the development.

## 2 Literature Review

### 2.1 Traffic Research

Traffic

#### 2.1.1

Some conditions for road users that affect the speed of cars and therefore traffic: road weather changes, car accidents and traffic congestion. (?)

#### 2.1.2

Difference between distance based navigation (a) and time based navigation based navigation (b). As you can see the time based navigation works out that though the distance is greater for a car to travel from points 1-5 it takes less time than from points 1-3-5. (pic)

#### 2.1.3

To find traffic congestion, find cars within one road or a set of roads that are averaging the lowest average speed and put the cars on different routes (repeat process) until the average speed has been brought up again to a more suitable average speed (20 km/h). (?)

Equation to determine the speed weighting for cars in the current area: (cars in average speed cluster) / (total number of cars) \* (average speed of cars in the current area). (equation) (?)

### 2.1.4

Traffic congestion is measured by detecting the average speed of the vehicles and if the vehicles are moving slower than 10km/h, then a road is congested (?).

## 2.2 Google Maps

Google maps was introduced in 2005, revolutionising the way maps worked when display on websites as it allowed users to interact with the map (?).

### 2.2.1 Google Maps Elevation API

Google Maps allows users to see the elevation of locations through the "terrain layer". This layer shades in different parts of the map to show which locations are higher than others. This data is taken from the "Elevation API". This allows users of the API to find the elevation of certain locations by providing the "LocationElevationRequest" with a GPS coordinate which will in return provide the user with an elevation for that GPS coordinate in meters. This API also allows the user to request a set of GPS coordinates along a path by providing the "PathElevationRequest" with two or more GPS coordinates along a path along with a number of samples for the path to be split up into, for example if a the path had two GPS coordinates and was split up into 16 samples then the "PathElevationRequest" would return a graph with 16 elevations between the two coordinates specified (?).

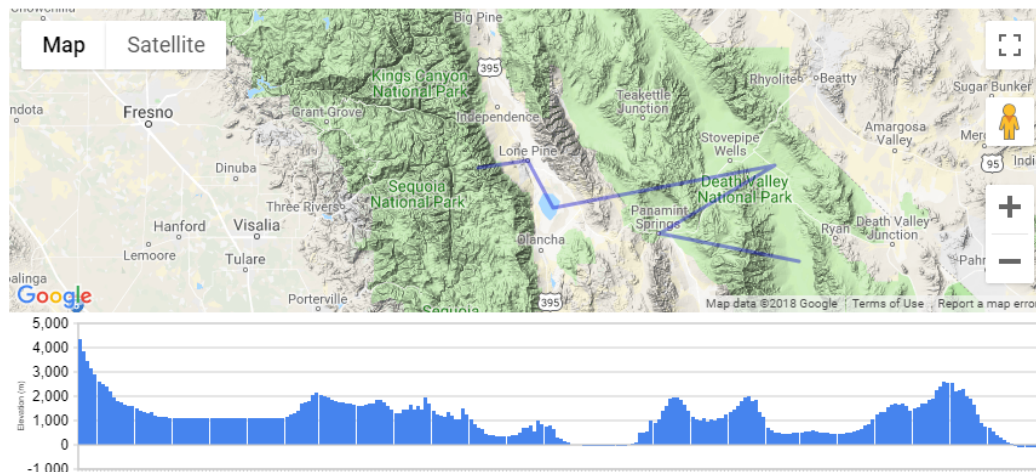


Figure 1: **PathElevationRequest**- This is what PathElevationRequest returns for 6 coordinates at 256 samples.

## 2.3 TomTom maps

TomTom was founded in 1991 beginning in the business-to-business section of the technology market. They then went on to develop personal digital assistants (PDAS) in the early 90s, going on to become the market leader in PDA software with navigation applications such as "RoutePlanner" and "Citymaps". In 2002, the TomTom "Navigator" provided the first affordable and easy-to-use navigation system to customers in Europe.

This increased demand for portable navigation devices, which lead to the development of the Portable Navigation Device (PND). This became the fastest selling consumer technology device in history and by 2016 TomTom had sold over 78 million PNDs.

Nowadays, TomTom provide customers with several navigation-based products, including a navigation app for android and wrist devices for sports enthusiasts, reaching over 800 million people a day (?).

### 2.3.1 Map tiles

The TomTom maps API provides the user to view the maps using raster tile images or vector tile images. The difference is subtle when viewing the maps, however in theory, raster tiles should be more detailed as the tile is coloured per pixel whereas vector imaging colours the tile based on the objects inside it (roads, rivers, etc.).

The number of tiles shown depends on the level of zoom that the user has defined (by using a pinching gesture on the screen). There are 23 different levels of zoom with 0 showing the whole world in one tile and zoom level 22 showing 2 to the power of 22 tiles.

### 2.3.2 Map layers

The TomTom maps API also allows the maps to be separated into different layers to make it easier to update the map or to show places in different languages etc. There are three layers defined in the TomTom documentation: Basic, Hybrid and Labels. The Basic layer shows all the maps features as an end user would see it. The Hybrid layer is a step down from that as it takes the polygons that outline the birds eye view of the buildings so that the view of the map is focussed on the roads more. The Labels layer shows only the labels of the named places shown on the map in the position they would be in, in the Basic layer. The allows the developer to see places more clearly which allows the addition of places to be easier and also to check whether the names of the places are in the correct place and are spelt correct in their respective languages.

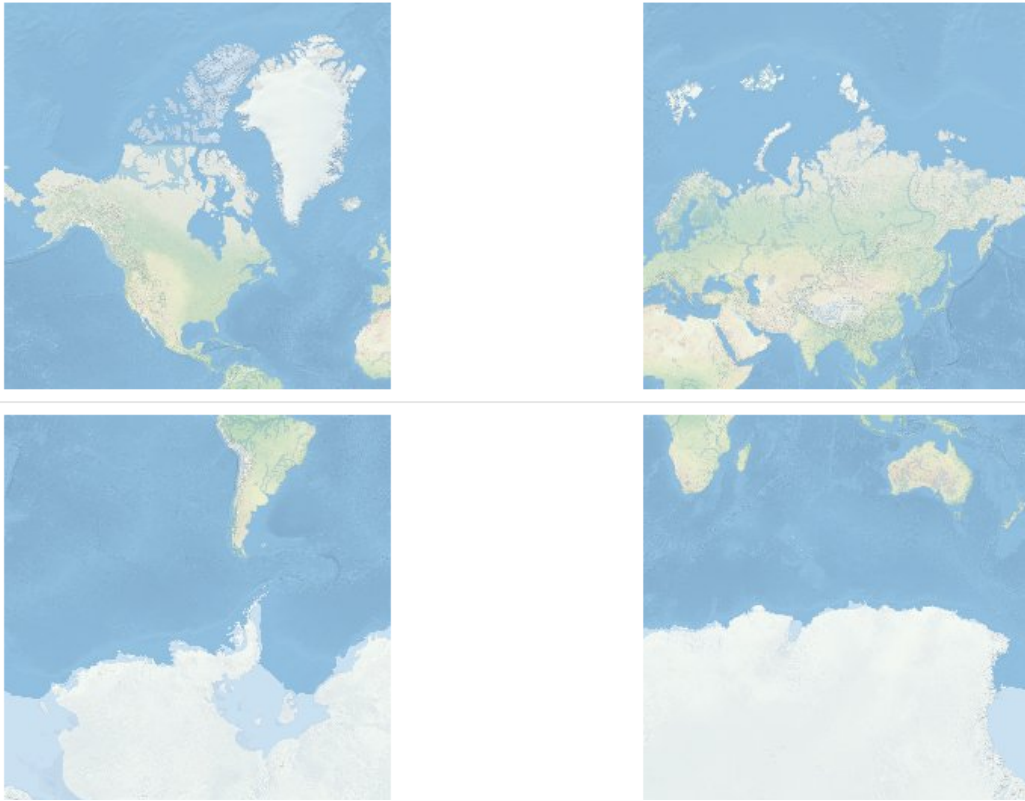


Figure 2: **Zoom level 1**- This is Zoom Level 1 which is 4 tiles in total.

The map also has styles that affect the way that the layers are shown: main and night. The main style is more of a light theme where all the usual map colours are used, however, the night style will use colours that are not too intrusive to the user at night (?).

There are also some layers that are not directly related to the map, these include: traffic incident road highlighting,

### 2.3.3 Traffic data

TomTom maps provide a traffic data API to allow developers to use real-time traffic data that TomTom collects. The data is split up into traffic incidents and traffic flows. The Traffic incidents API allows the user to be informed of traffic jams due to road incidents that will affect the route that the user takes. The traffic flow API tracks the users speed on the routes that they are taking so that the next user can benefit from the road being used before them to provide them with more accurate routing and predicted travel times. The traffic incident API updates every 2 minutes to give the user (and the

software) the chance to change their route in accordance to the latest traffic incident updates. The traffic flow API is updated every minute.

The traffic incident API will return information to the user and the system that details the impact of certain traffic incidents within the tile the user is currently in, or in fact along the route that the user is taking.

#### **2.3.4 Traffic data**

### **2.4 Routing Research**

#### **2.4.1**

Data stored from real-time cars in database to affect the routing algorithm: long/lat, time stamps, driving speed and orientation (data uploaded four times a minute to the server). (?)

#### **2.4.2**

Compare the routing algorithm time with other routing algorithms like: Heristic1 (time-based algorithm), Heristic2 (time-based algorithm), Dijkstra (distance-based). (?)

#### **2.4.3**

Based on data in Beijing, 60% of the total time is spent on intersections, although, 2.7% of all the intersections were main intersections (400 intersections) and they took 18% of the total intersection-time suggesting that a large proportion of the total intersection-time was wasted on main intersections. (?)

#### **2.4.4**

OpenStreetMap description: the map of an area (region, province, city, etc) is implemented as an oriented graph characterised by two main elements: nodes and ways.

Nodes represent important positions on the map identified by GPS coordinates (lat and long) corresponding to Points of interest (POIs), intersections, points of change in direction on the same road (curves), etc. Thus, between two nodes a linear segment is defined. Each node is associated with an id and with a list of tags that describes the characteristics of the node. The number of nodes on a road depends on the type of road (typically urban roads are described with a higher amount of nodes with respect to highways).

Ways are ordered sets of nodes constituting open or closed polygons, representing possible paths from the start node to the end node. The nodes of a way are listed consecutively and this allows identifying stretches of consecutive roads. (ref)

#### 2.4.5

here is picture explaining the different types of segments (description of each needed before pic) (pic)

#### 2.4.6

Graphhopper puts the base node (where the car starts) into both forwards and backwards lists (two opposite-direction lists that display nodes that the car has passed) until the second GPS reading is taken and then removes the nodes from list that is not needed after the second reading is put into the right list. (ref)

#### 2.4.7

The usual states of traffic that popular systems implement are: flowing, slow, very slowed and blocked. Another state was added by (ref) which is absent. Which means there is either no traffic or a total absence of vehicles, this is used for incident detection. The usual traffic systems will check the number of vehicles that have crossed a segment in an amount of time.

#### 2.4.8

In the (ref) the system checks if there is sufficient vehicles on each road segment and sets the traffic states based on a few rules:

- If there is insufficient vehicles: If there is 0 vehicles then set state to absent else then set the state to flowing

Else (if there is enough vehicles) Find the median speed of the cars in the road segment and compare the median speed to a percentage of the speed limit that is close enough to the actual speed limit. If the percentage of speed limit is reached (p1) or a higher speed is found, then set the traffic state to flowing. If it is within a set percentage range lower (p2) then the state is set to slower. If the speed is reaching an even lower percentage of the speed limit (p3) then the state is set to very slow. Else the traffic is blocked.

In the experiments used in this paper:  $p1 = 50\%$  ,  $p2 = 40\%$  and  $p3 = 30\%$  and the sufficient number of cars was set to 4.

#### 2.4.9

Google traffic takes account of four different traffic states: (i) Normal speed of traffic (green colour) (ii) Slower traffic (yellow colour) (iii) Congestion (red colour) (iv) Blocked or stop-and-go traffic (dark red colour)

### 2.5 Car average speed Research

#### 2.5.1 Floating car data

Problems with tracking average speed is that GPS is not entirely accurate. (?)

Intersection delay is a factor that contributes to route calculation. (?)

### 2.6 Traffic sensor Research

#### 2.6.1 RFID

Disadvantages of active RFID: - Cost per tag - Need for battery maintenance/ higher maintenance costs - Higher probability of interfering with other devices using similar frequency bandwidths in the area (ISM networks) - Propriety communication protocols

By contrast, advantages of passive RFID tags are: - Theoretically, the obtained distance of seven meters based on static experiments in (?) is a good enough range for reading passive RFID tags as cars are normally just over two meters away from the pavement and based on the data rate (40Kbps-protocol: EPC GLOBAL 1G2 ISO 18000-6C) requires only twenty-five milliseconds to measure a tag. - In theory, the passive tag in this paper can measure cars passing at speeds of up to one hundred kilometers an hour, which therefore suggests that for most use-cases in the city, cars can be read by passive tags. - Another advantage is that the complex anti-collision protocol used in (?) is not necessary due to the reader, used in the experiments, being able to read multiple tags on the same plate.

The conclusion drawn from form this comparison is that the passive tag advantages outweigh the use of active tags, in this situation, as the active tags would be "overkill". These results are then validated by the car-in-motion sensor test results - at fifty kilometers an hour, ten tags were read in about 250 milliseconds. However, this approach is limited as the RFID reader needs constant *or at least for the few seconds that a car passes the road sign with an RFID reader attached* line of sight with the car's RFID tag as anything such as a parked vehicle could block the reading. (?)

# Appendices

## A Project Overview

Dylan Tyrie-Dron

40203045

### Initial Project Overview

#### SOC10101 Honours Project (40 Credits)

#### Title of Project: Satellite Navigation based on traffic within cities

#### Overview of Project Content and Milestones

The aim of this project is to describe a way of navigating through a city through the use of real-time traffic data. This will be achieved by testing different pathfinding algorithms for example: \cite{Dijkstra1959}. Research will also be carried out on traffic and other data affecting vehicles getting to their destinations faster. This will be done by rigorously testing car speeds at certain times in the same place within the city Edinburgh. This should give data that can input into a pathfinding algorithm. The results will then be compared to similar experiments developed by others as a way of finding out which method was the most accurate, cost-effective and efficient to travel through a city and discussing which methods are the most suitable for getting through any city. The main interface used to convey the outcome of this project is an android application. It is also thought to make use of API's like: "OpenstreetMaps" to speed up the development.

#### The Main Deliverable(s):

- A report describing and evaluating the main application for this project and discussing why this project is a potential improvement over other older projects and current technologies.
- A mobile app that can find the most effective way of getting to a certain destination by car, given the data that it is presented with using pathfinding algorithms(ref).
- A way to store some data that will help simulate the experience of using the mobile app to navigate through traffic in the city. This will be done by storing the likes of the number of cars that are on roads that may be used to affect the algorithm that calculates the route to the user's destination.

#### The Target Audience for the Deliverable(s):

The main audience that this project will be targeting is the road users in a city, however researchers may use this project to their benefit to use for their own projects. Developers could also use this project to improve their own software or to expand this project.

#### The Work to be Undertaken:

- Investigating the best way to implement the mobile app e.g. calculating the outcome of the algorithms using a server computer and having the phone retrieve the results of these calculations or to just do the calculations through the mobile phone.
- Approaching the project in an agile way: creating diagrams and plans to plan development of the project, setting up sprints etc. so that the project runs smoothly.
- Implementing the initial mobile app.



Dylan Tyrie-Dron

40203045

- Testing the data by comparing it to real world routes and to also test the different ways of retrieving the data and finding out how accurate the app will be.
- Fixing bugs and adding extra features to the app to flesh out the project.
- Evaluating each stage of the process to ensure that lessons have been learnt whilst undertaking the task at hand.

**Additional Information / Knowledge Required:**

- Expanding my knowledge on how pathfinding works.
- Finding a way of making an app look more polished (compared to previous mobile application of my own)
- Linking the data collected to affect the algorithm and the overall route calculated. (too vague)
- Finding out if what types of sensors could be used if this project was to be used using real traffic data.
- Possibly linking a database built on a server to a mobile app if that is the efficient method of storing the data.
- Expanding on my previous knowledge and experience of Android Studio.
- Learning how to integrate an API or more.

**Information Sources that Provide a Context for the Project:**

- Papers
- Library
- Google Scholar

**The Importance of the Project:**

This project would potentially be a useful tool to use to navigate through cities in the future and therefore theoretically could be important to people wanting to get to certain places fast or to find the most suitable route for them. However, this project is not necessarily important in current cities as the project would need data that is potentially not available to users right now.

Dylan Tyrie-Dron

40203045

**The Key Challenge(s) to be Overcome:**

- Integrating one or more API's
- Implementing the most efficient code/approach when creating the app.

## B Diary Sheets (or other project management evidence)

### EDINBURGH NAPIER UNIVERSITY

#### SCHOOL OF COMPUTING

#### PROJECT DIARY

**Student:** 40203045-Dylan Tyrie-Dron

**Supervisor:** Brain Davison

**Date:** 03/10/18

**Last diary date:** -

**Objectives:**

- Read two papers
- get IPO completed

**Progress:**

- Read two papers
- Got most of the IPO done to be checked on 03/10/18 (question to be asked about it)

**Supervisor's Comments:**

-Brian happy with progress made however recommends that the IPO should be finished quickly so that it can be checked by second marker.

**EDINBURGH NAPIER UNIVERSITY**

**SCHOOL OF COMPUTING**

**PROJECT DIARY**

**Student:** 40203045-Dylan Tyrie-Dron

**Supervisor:** Brain Davison

**Date:** 10/10/18

**Last diary date:** - 03/10/18

**Objectives:**

- Read two papers
- Gantt chart finished
- Start looking at exercises for the project: make a small map's activity in android studio
- Finish IPO

**Progress:**

- Read one paper
- Finished IPO
- Gantt chart finished
- Did not get started on exercises due to an increase of coursework activity on both other modules.

**Supervisor's Comments:**

- Missed supervisor meeting this week

**EDINBURGH NAPIER UNIVERSITY**

**SCHOOL OF COMPUTING**

**PROJECT DIARY**

**Student:** 40203045-Dylan Tyrie-Dron

**Supervisor:** Brain Davison

**Date:** 17/10/18

**Last diary date:** - 10/10/18

**Objectives:**

- Read two papers
- Start looking at exercises for the project: make a small map's activity in android studio

**Progress:**

- Read two papers
- looked at a small google maps activity

**Supervisor's Comments:**

**EDINBURGH NAPIER UNIVERSITY**

**SCHOOL OF COMPUTING**

**PROJECT DIARY**

**Student:** 40203045-Dylan Tyrie-Dron

**Supervisor:** Brain Davison

**Date:** 24/10/18

**Last diary date:** - 17/10/18

**Objectives:**

- Read two papers
- Gantt chart
- read references tutorial

**Progress:**

- Read references tutorial
- Read one paper

**Supervisor's Comments:**

**EDINBURGH NAPIER UNIVERSITY**

**SCHOOL OF COMPUTING**

**PROJECT DIARY**

**Student:** 40203045-Dylan Tyrie-Dron

**Supervisor:** Brain Davison

**Date:** 31/10/18

**Last diary date:** - 24/10/18

**Objectives:**

- Read two papers
- Do some more exercises on Openstreetmaps
- lit review draft

**Progress:**

- Read one paper

**Supervisor's Comments:**