

Chapter 1

Introduction

In recent years, the availability of global positioning system (GPS) data has increased dramatically, not only in the transport industry but also for human behaviour in general. The cause for this can be linked directly to the rapid development of GPS systems and their integration into mobile devices and fleet monitoring devices, as well as the increase in availability of dashboard-mounted travel guidance- and monitoring systems (Lou et al., 2009). This spread of GPS-enabled systems and devices has led to an increase in accessibility of transport GPS trajectories; a sequence of GPS points recorded in chronological order. These trajectories are beneficial for many applications such as location-based service, urban planning and traffic management, by giving insight into understanding users' moving behaviour. Ghiani et al. (2015) focuses on the application of GPS to improve the performance of route planning algorithms on solid waste collection vehicles and states that by analyzing vehicle trajectory data companies can significantly improve the services they provide. Although the benefits of GPS trajectories, especially in the transport industry, are evident through the use of intelligent transport systems (ITS) as described in Quddus et al. (2006), the administration of the data can impose a considerable financial burden due to data warehousing. For applications where detailed location accuracy is not required, the data polling frequency is reduced to decrease operating costs by reducing data storage requirements. However, the resulting traffic information, albeit smaller in storage requirements, will be less accurate. The outcome of this is that if the polling frequency is low, for example, 1 GPS point every minute or 1 GPS point every 500 m, one will not be able to identify the route of the vehicle with a high level of certainty (Lui & Morikawa, 2006; Miwa et al., 2012). Fortunately, there are many successful attempts documented in academic literature and conference papers of the development of map-matching algorithms that could accurately process the low polling frequency within a required level of accuracy.

A GPS monitoring system will record the location of the device at specific frequencies using trilateration, a method to determine the relative positions of objects using the geometry of triangles. Frequency is often expressed in hertz (Hz) and represents the number of observations per second. That is, if GPS observations are 1 s apart they have a frequency of 1 Hz, while observations that are 5 s apart will be expressed as $1/5 = 0.2$ Hz. GPS trajectories used in this project are often at very low frequencies and this dissertation will refer to these low frequency trajectories using the inverse of frequency, period, to define a trajectory. For example a 0.01 Hz frequency can also be referred to as a *100 s period*. This dissertation uses the two terms interchangeably, but with preference to *period* for frequencies below 1 Hz.

In this study the specific use case for a map-matching algorithm was to analyse the routes from waste collection vehicles. Not only to identify the route a vehicle took but

also to infer from the results which parts of the route the vehicle was servicing and which parts were merely used for travelling from one service area to another, called *deadheading*. This will allow researchers to allocate the waste collected by the vehicle to a very specific area. To achieve this, the speed of the vehicle was used as a proxy to determine whether it is deadheading or collecting waste, since the travelling speed between the two activities differs significantly.

Suppose a research project is tasked with analysing the waste collection in Cape Town, South Africa. The project team have all the waste collection data in terms of weight of solid waste collected, date and specific vehicle. The travel information for each vehicle the was recorded using off-the-shelf **GPS** monitoring systems to get **GPS** trajectory data for every trip the vehicle took. Due to data storage limitations, the team of the aforementioned research project had to gather **GPS** points at a low polling rate of one sample every 30 s (0.033 Hz) to reduce the number of points stored per route. Figure 1.1 shows an example of a portion of such a trajectory for a fictitious trip that was recorded from a waste collection vehicle that operated in the area shown. The *true path* is the actual route that the fictitious driver travelled, and would have been recorded manually by the driver.

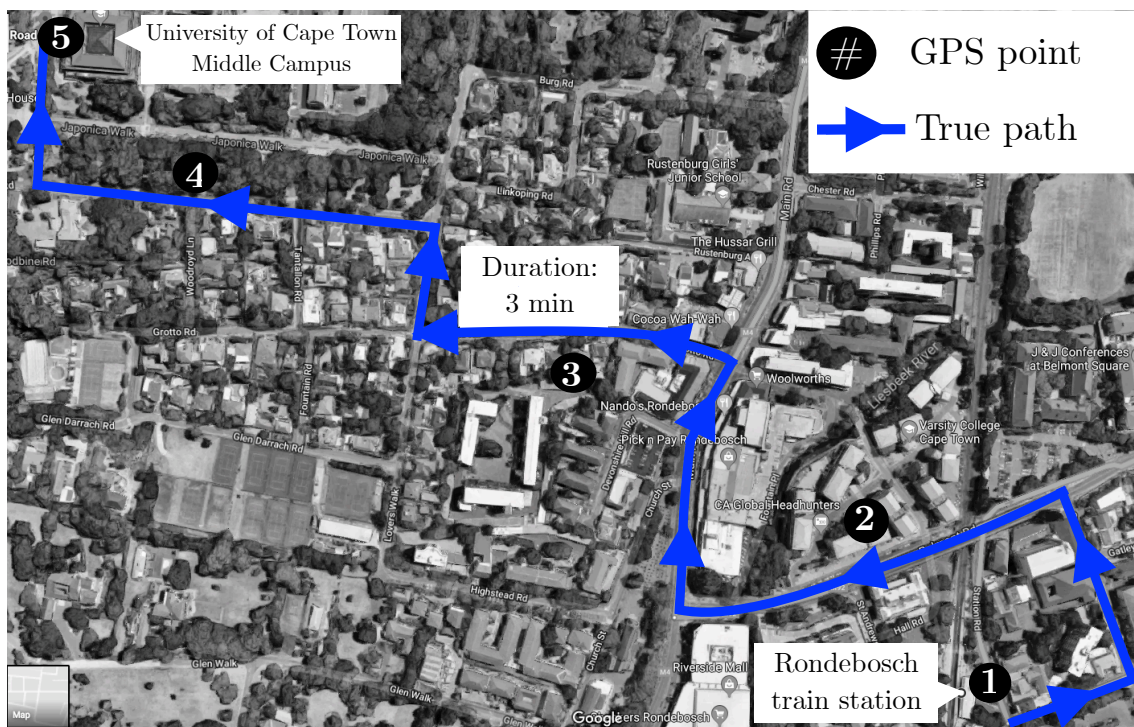


Figure 1.1: Example of a low-sampling **GPS** trajectory between Rondebosch train station and University of Cape Town Middle Campus

Using only the **GPS** points, the researchers want to determine:

1. The most probable route the driver took
2. The most probable speed at which the driver traversed each link

1.1 Defining map-matching

Map-matching is the process of aligning a sequence of observed user positions with the road network on a digital map (Lou et al., 2009). Some researchers, such as Yin & Wolfson (2004), refer to map-matching as *snapping*. Map-matching is often a fundamental step in processing data before it can be used in applications, such as moving object management, traffic flow analysis, and determining driving directions. Typically, a recorded GPS data point consists of an entry of the latitude, longitude, and timestamp. A GPS trajectory is then made up of a set of these data points. Due to the limitations of GPS devices caused by inherent error in accuracy, as well as the error caused by a low-sampling rate, map-matching algorithms are required to evaluate alternatives and give the most probable route the object travelled. The accuracy of map-matching could differ significantly depending on GPS accuracy and the sampling rate (Yin & Wolfson, 2004). According to Lou et al. (2009), low-sampling-rate GPS trajectories, i.e. one point every 2–5 min, are becoming more and more available but unfortunately, most current map-matching approaches only deal with high-sampling rate trajectories, typically one point every 1–30s, and such approaches are less effective for low-sampling rates. Because the sampling frequency plays such a vital role in determining the accuracy of the map-matching algorithm, distinct methods have been developed to cater for the different data sets.

Consider GPS points 3 and 4 from Figure 1.1 as enlarged in Figure 1.2. This example illustrates the effect that low polling frequency GPS trajectories, as well as the inherent error in GPS data points, have on the accuracy of map-matching. There are seemingly three possible routes that the vehicle could have taken and still generated GPS points 3 and 4 on their exact locations.

- **Route A:** Continue on Grotto Road and up Woodroyd Lane.
- **Route B:** Instead of Woodroyd Lane use Tantallon Road.
- **Route C:** Turn right into Lovers Walk and then Left into Stanley Road.

Without a proper map-matching algorithm it is unclear how to find the most probable route that the vehicle could have taken. The numbered black dots in Figure 1.2 indicate the GPS points and the order in which they were recorded, while the arrows indicate the path that the driver recorded manually after the trip. This was used to determine the accuracy of the calculated path. According to the driver’s recording, this route is 1.6 km long and should take approximately 3 min to complete. As expected, due to GPS error, the GPS points are not exactly on the specific roads the vehicle travelled on. The start of the route appears to be the Rondebosch train station entrance in Station Road and the end at the University of Cape Town (UCT) Middle Campus entrance in Cross Campus Road. In order to digitally map these coordinates to the most likely path that the driver took, the research team needed to match the GPS coordinates to a digital map of the road network using a map-matching algorithm.

1.2 Research design

The objective for this study was to develop a map-matching algorithm that could be used to identify the most probable routes for vehicles from a dynamic sampling rate GPS data set. The primary focus of this study was to create the algorithm in Java so that it could be used to map a GPS trajectory to a Multi-Agent Transport Simulation (MATSim) network. This algorithm could then be used to build upon the work of Joubert & Axhausen

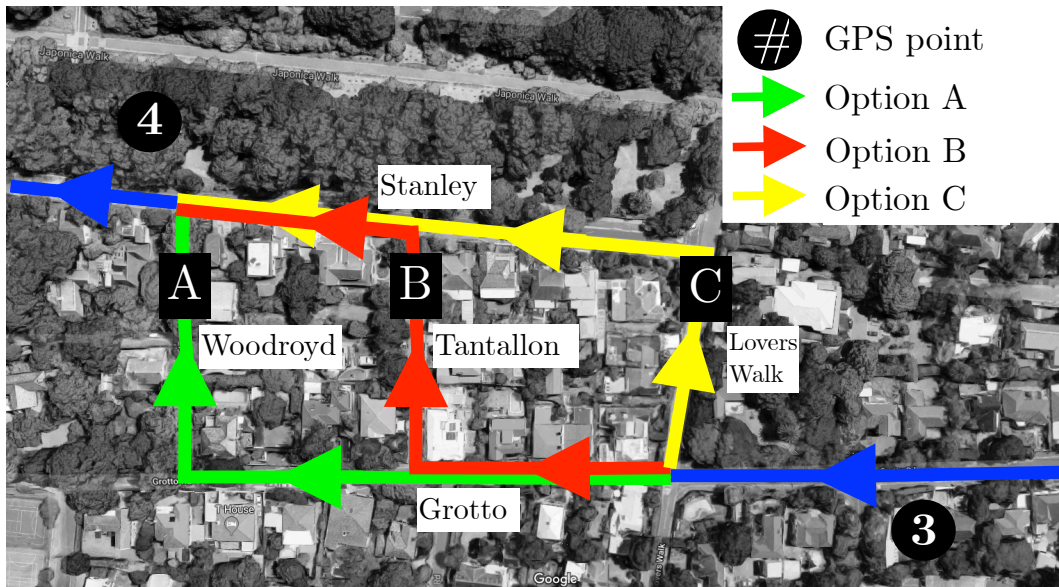


Figure 1.2: Possible routes with low-sampling GPS trajectories

(2011), who used a similar data set to identify freight activities in South Africa. They used activity locations, along with the activity chain characteristics to generate a representative synthesised populations of freight agents that were later used in a large-scale agent-based transport simulator. The simulator is used to provide decision support on transport infrastructure by enabling the ability to simulate *what if* scenarios. The artefacts of this study could provide another means of validating the model output and assist in improving the representation of the agents in the population. This algorithm will be able to provide the most probable routes that drivers travelled and compare the routes chosen by drivers in the simulation. The secondary objective was to output the inferred speed of the vehicle across the inferred path as well as indicate the confidence or probability of the inferred path.

The immediate use case for this algorithm was to analyse the routes of waste collection vehicles and infer on which parts of the routes were the vehicle collecting waste versus deadheading. This will allow other research projects to assign waste collection data to specific areas to analyse waste generation. This could also provide valuable information to assist in optimising waste collection routes and improve efficiency and effectiveness of waste collection. Ghiani et al. (2015) describes how they use GPS data to not only estimate the travel and service times of vehicle but also automatically classify whether certain services has been provided on a certain day which aids in reporting and management of waste collecting services.

This study produced the following artefacts:

- map-matching algorithm written in Java that maps a list of GPS points (containing longitude, latitude and time of observation) with a MATSim road network object;

- benchmark datasets of random routes travelled on the Cape Town [MATSim](#) road network and using self-generated [GPS](#) trajectories at different frequencies for each trajectory;
- Java algorithms for generating experimental datasets of:
 - [MATSim](#) grid networks;
 - routes, as a list of links from a road network, between a number of either random or predefined points; and
 - [GPS](#) trajectories generated from a route on a road network;
- Java algorithms to calculate:
 - accuracy of a route compared to the true path according to four different measurements;
 - analysing the inferred speed of the matched route on the network and comparing it against the free speed of the matched network links;
- analysis of the relationship between accuracy and efficiency of the algorithm at different parameters; and
- the preferred parameters for analysing routes on the Cape Town road network.

1.3 Research methodology

Since this study was not aimed at improving existing map-matching algorithms or developing novel ways of implementing map-matching, the research methodology was focussed on researching existing map-matching algorithms and converting the most applicable algorithm into a Java program that uses [MATSim](#) objects. The second part of the study was to analyse the performance of the algorithm on benchmark datasets generated and then to analyse real-world data sets from municipal waste collection vehicles in the City of Cape Town.

The dissertation starts with a thorough literature review in Chapter 2 of different methods used to map [GPS](#) trajectories to road networks. Literature divides the different methods based on the type of [GPS](#) trajectories to map and what the purpose of the map-matching is. [GPS](#) trajectories are either high or low-frequency trajectories, and the purpose of mapping is either real-time mapping, for [GPS](#) guidance, or post-mapping, for route analysis [Li et al. \(2013\)](#). At the end of Chapter 2, the different accuracy measurements used to assess the algorithm performance in previous studies are explored, and the chosen measurements used in this study defined.

In Chapter 3, the map-matching method most applicable to this study is analysed and its implementation in Java, with the use of [MATSim](#) objects, is explained in detail. Some aspects of the algorithm had to be slightly adjusted for this Java-specific implementation.

To test the impact of various factors on the performance of the algorithm, both in terms of accuracy and efficiency, Chapter 4 reviews the results of various controlled experiments with the algorithm. Experiments range from simple grid networks with fixed trajectories and paths to data simulated on a real-world road network at various frequencies. The effectiveness and efficiency of the algorithm is tested using different parameters to identify the optimum setup for doing analyses on the real-world network

In Chapter 4 the aim is not only to assess the accuracy of the algorithm but also the efficiency, as running the algorithm on a dataset similar to that used in [Joubert &](#)

[Axhausen \(2011\)](#), can take significant time to execute, around one hour for a route with four thousand [GPS](#) points.

Chapter 5 ends with a brief analysis of the algorithm on real-world data and a review of the results generated.