AI6128 URBAN COMPUTING

PROJECT 2: TRAJECTORY AND ROAD NETWORK DATA ANALYSIS

Chang Fuh Yong
School of Computer Science and Engineering
Nanyang Technological University
Singapore
ch0113ng@e.ntu.edu.sg

Brandon Chua Shaojie
School of Computer Science and Engineering
Nanyang Technological University
Singapore
bran0026@e.ntu.edu.sg

Cheong Kok Yuet
School of Computer Science and Engineering
Nanyang Technological University
Singapore
kcheong012@e.ntu.edu.sg

Introduction

There is a saying “Every road leads to Rome”, however in today’s context, which is mainly the modern Urban City environment, due to traffic congestion, road accidents etc, this could not be true.
The study of trajectory analysis has become paramount important especially for companies like taxi dispatchers such as Uber and also the city road planning institutes.

With the ubiquitous GPS signals produced by taxis, the amount of data that can be used for “Map Matching” [1] can be very beneficial and precious to the research in the area of traffic trajectory analysis. However, not all the GPS signals are “cleaned”. Due to various reasons, the noisy signal [2] needs to be pre-processed, filtered and followed by processed and analysed.

This project consists of 5 tasks and 1 optional Task. Task 1 is to download the two sets of data, namely the trajectory and road network data. Once the datasets are prepared, visualizations for the raw GPS points from the trajectory dataset should be plotted onto the road network. The first ten trips will serve as a basis to represent the trajectory dataset. Task2 is to visualize the raw GPS points from the trajectory dataset and plot them onto the road network. Task 3 is to perform map matching to map the raw GPS points to the road segments. Fast Map Matching (FMM) [3] is recommended for this task. Task 4 is to visualise and analyse the results of the “Map matching “result. Task 5 requires us to derive and implement the extraction of the top five most traversed and longest average travelling time road segments and visualize them on graphs. The final optional task wants us to identify less satisfactory map matching cases and suggest improvement and implementation.

Task 1 Data Preparation

Road network is a system of interconnecting roads, highways between two points. The road network within Porto city’s administrative boundary in Portugal from OpenStreetMap is represented by notes and edges, as show in Figure 1-1.

Figure 1-1: Porto, Portugal, Administrative Boundary

For Task 1, taxi trajectory data from [Kaggle’s ECML/ PKDD 15: Taxi Trajectory Prediction] train.csv is used. Each row in the .csv file details one trip as follows:

- TRIP ID: String is a unique identifier for each trip.
- CALL TYPE: Char identifies the way used to demand this service, can be ‘A’, ‘B’ or ‘C’.
- ORIGIN CALL: Integer is a unique identifier for each phone number used to demand service(s).
- ORIGIN STAND: Integer is a unique identifier for the taxi stand.
- TAXI ID: Integer is a unique identifier for the taxi driver for that trip.
- TIMESTAMP: Integer is a Unix Timestamp (in seconds). It identifies the trip's start time.
- DAYTYPE: Char identifies the daytype of the trip's start, can be ‘A’, ‘B’ or ‘C’.
- MISSING DATA: Boolean is FALSE when GPS data stream is complete and TRUE whenever one (or more) locations are missing
- POLYLINE: String contains a list of GPS coordinates (i.e. WGS84 format).

For this Task 1, first 1000 trajectories contained in the file will be used for the remaining road network analysis. This subset will be referred as ‘train-1000’ dataset.

Data Analysis

All Trajectories

Train.csv contains the following attributes:

- 1,710,669 taxi trajectories
- Call Type: 364,770 trips dispatched from the central. 817,881 trips directly from a specific stand. 528,019 other trip types.

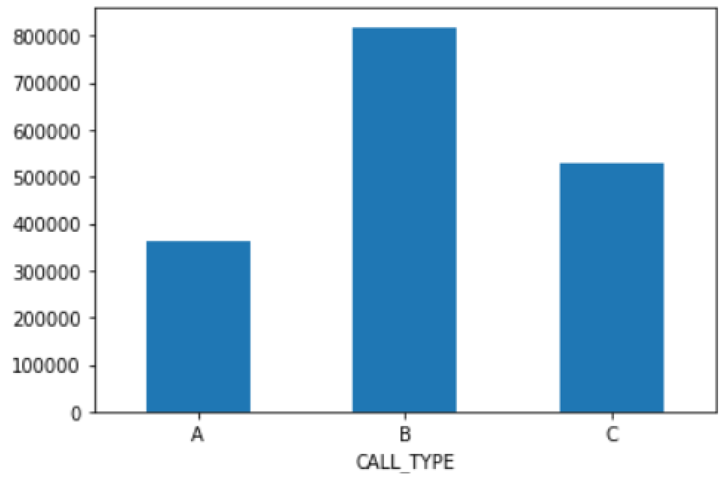

Figure 1-2: Call Type for All Trajectory

- 57,105 phone calls for taxis. Note the first caller (ID: 2002.0) and the last caller (ID: 63882.0) of the 57,105 callers are frequent clients. It is likely that the two callers are business entities.
- 63 taxi stands as originating point

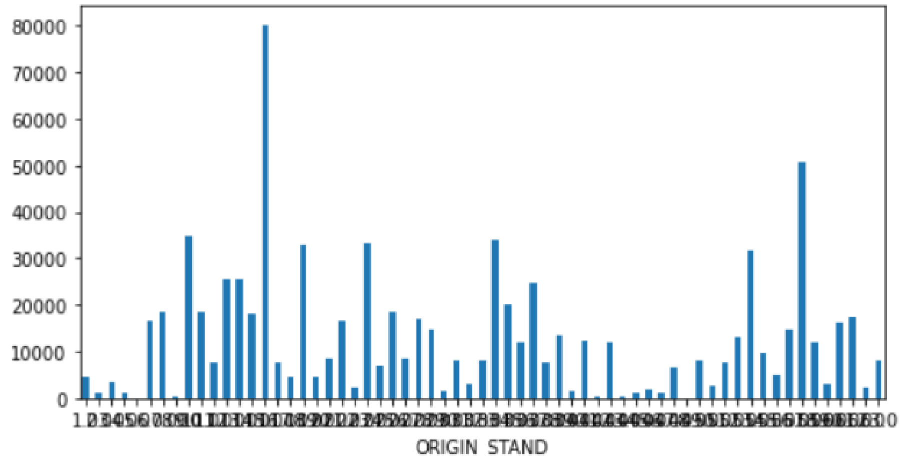

Figure 1-3: Origin Stand for All Trajectory

- 448 taxis operating

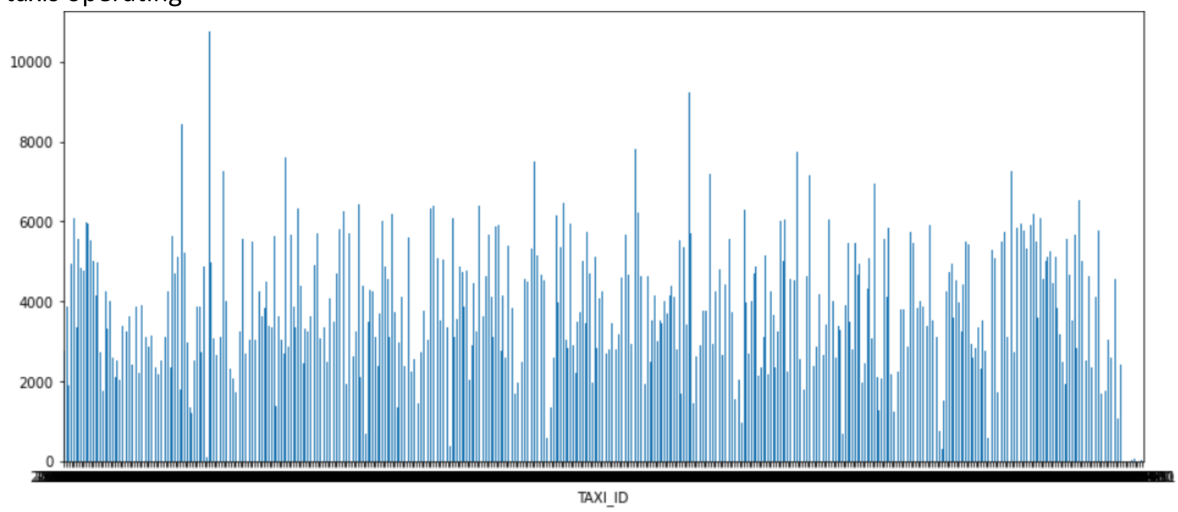

Figure 1-4: Taxis for All Trajectory

- 1,710,660 GPS data steam is complete. 10 GPS data steam has one or more locations missing.

1000 Trajectories

Train.csv contains the following attributes:

- Call Type: 267 trips dispatched from the central. 434 trips directly from a specific stand. 299 other trip types.

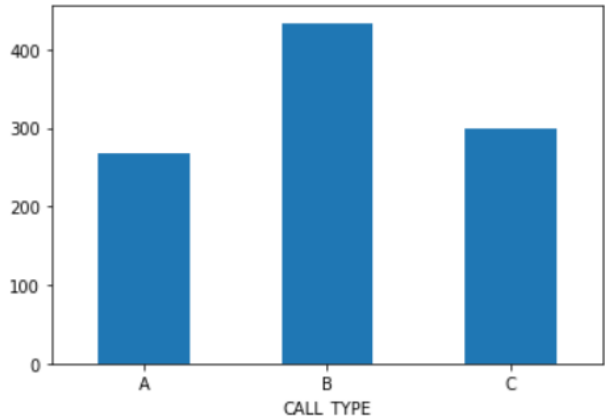

Figure 1-5: Call Type for 1000 Trajectory

- 203 phone calls for taxis. Note the first caller (ID: 2002.0) and the last caller (ID: 63882.0) of the 203 callers are frequent clients. It is likely that the two callers are business entities. This observation is the same for analysis on all trajectories.

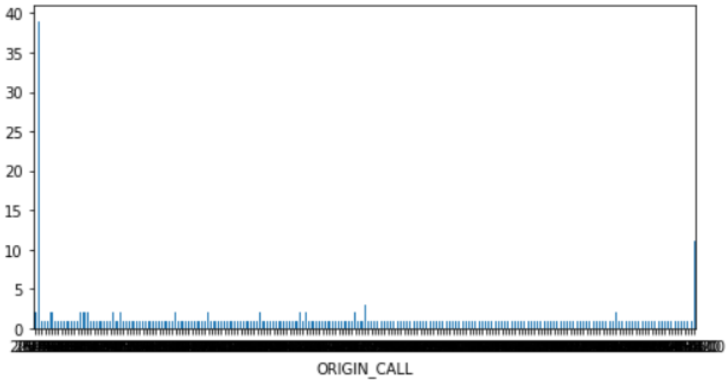

Figure 1-6: Origin Call for 1000 Trajectory

- 63 taxi stands as originating point

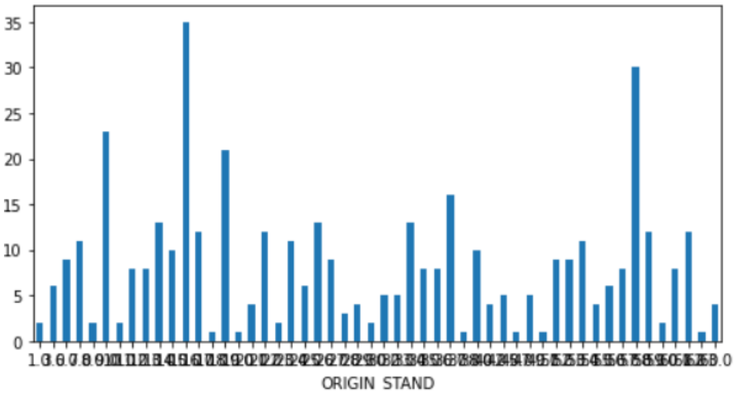

Figure 1-7: Origin Stand for 1000 Trajectory

- 309 taxis operating

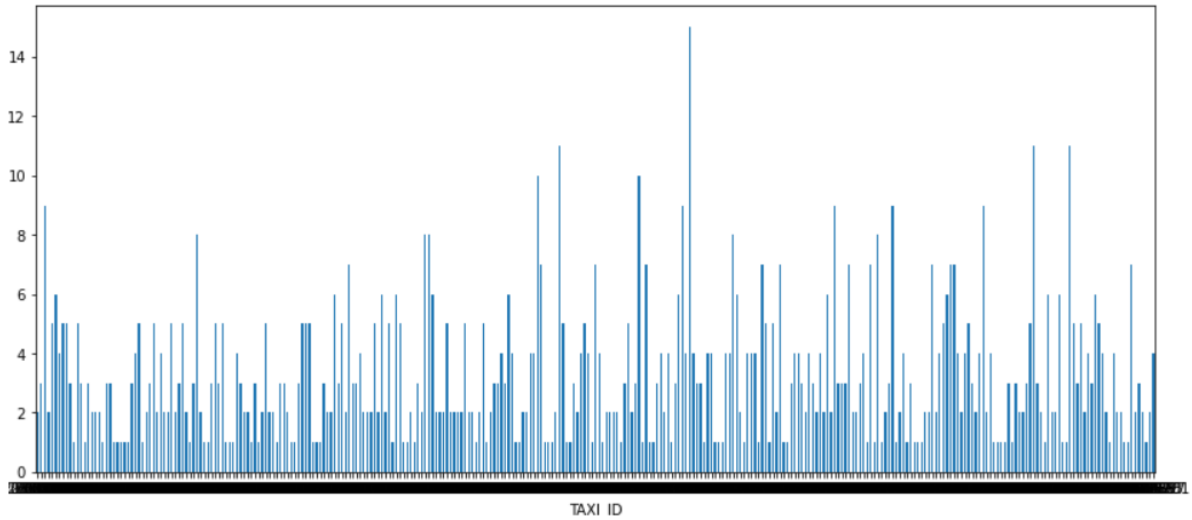

Figure 1-8: Taxis for 1000 Trajectory

- All 1000 GPS data steam is complete, no missing locations.
- Row, trajectory 763 is an empty row, with no data.

Task 2 GPS Point Visualization

Each GPS point consisting of latitude and longitude for each of the 10 trajectories is plotted onto the road network of Porto. Figure 2-1 shows the 10 trajectories plotted in 1 map. Figure 2-2 to 2-6 shows the GPS points of each trajectory.

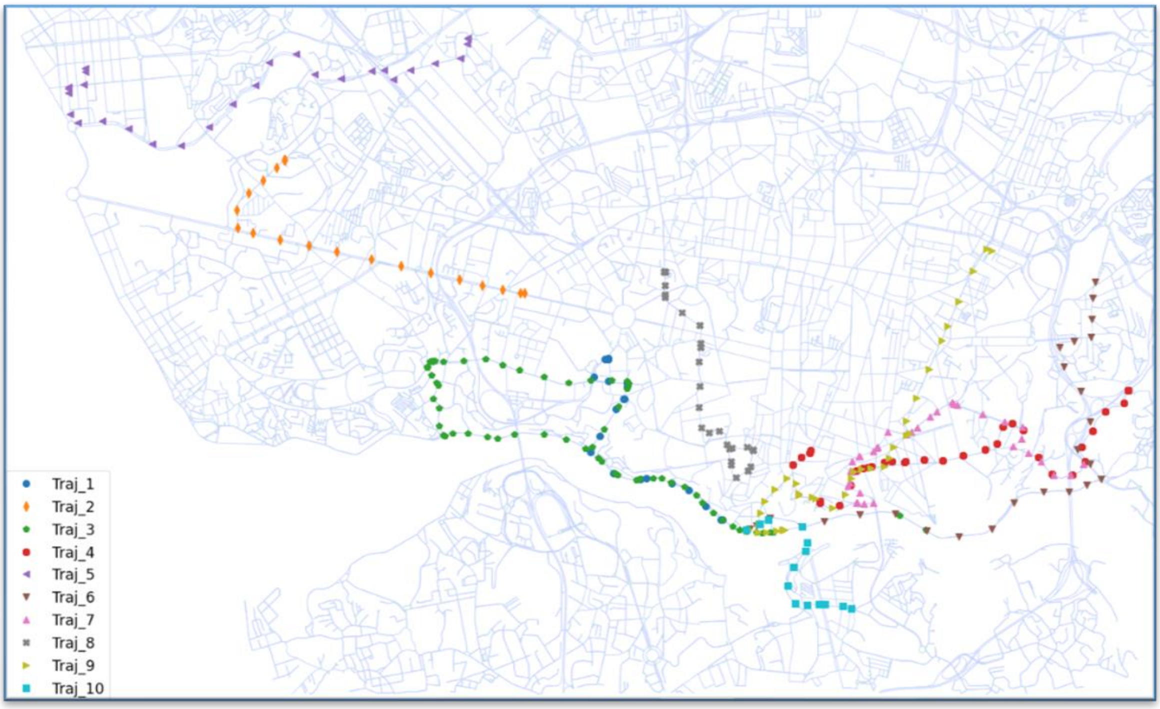

Figure 2-1: GPS Plot for 1st 10 Trajectory

Figure 2-2: GPS Plot for 1st & 2nd Trajectory

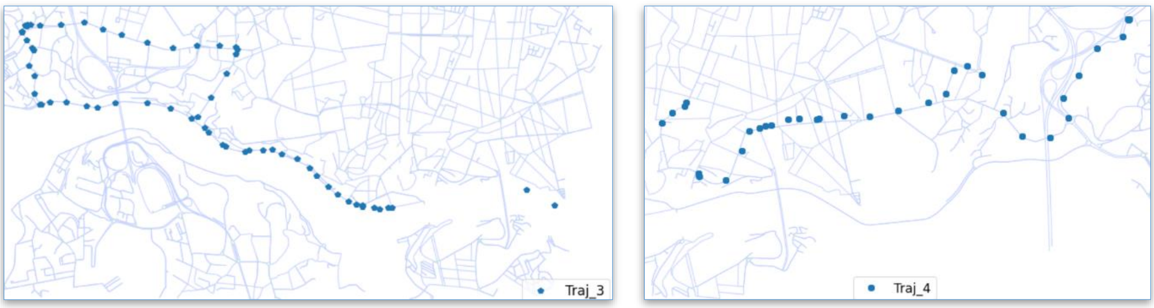

Figure 2-3: GPS Plot for 3rd & 4th Trajectory

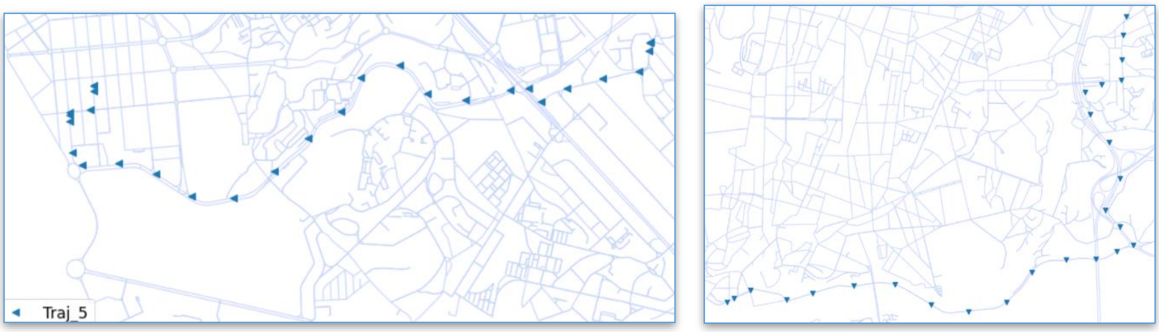

Figure 2-4: GPS Plot for 5th & 6th Trajectory

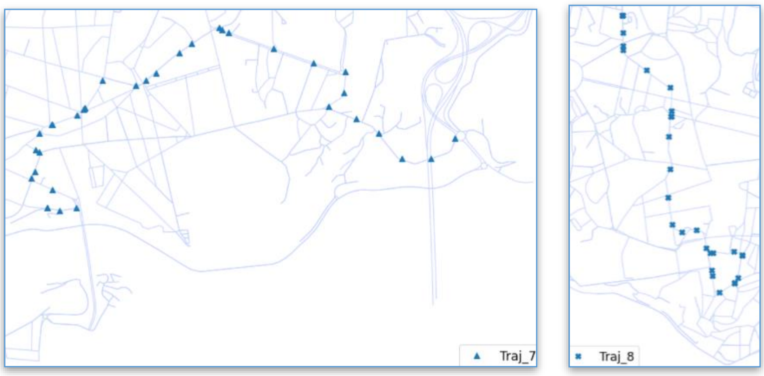

Figure 2-5: GPS Plot for 7th & 8th Trajectory

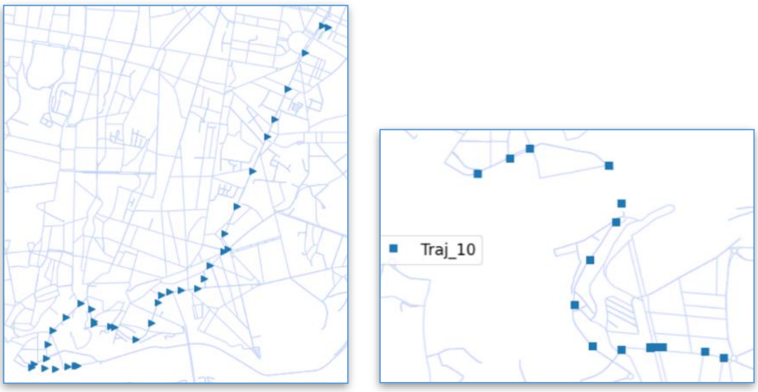

Figure 2-6: GPS Plot for 9th & 10th Trajectory

Task 3 Map Matching

The trajectory data is matched to the road network using Fast Map Matching (FMM) [3], an open-source map matching framework in C++ and Python. Task 3 is run in a cygwin environment. The output is a CSV file containing all the mapped routes of all 1000 trajectories in the train-1000 dataset. The CSV file is named (matchMaps.csv) and is stored in the **Task3_mappedRoutes** folder.

Task 4 Route Visualization

Visualize the routes that are mapped from the trajectories of the first 10 trips on the road network.

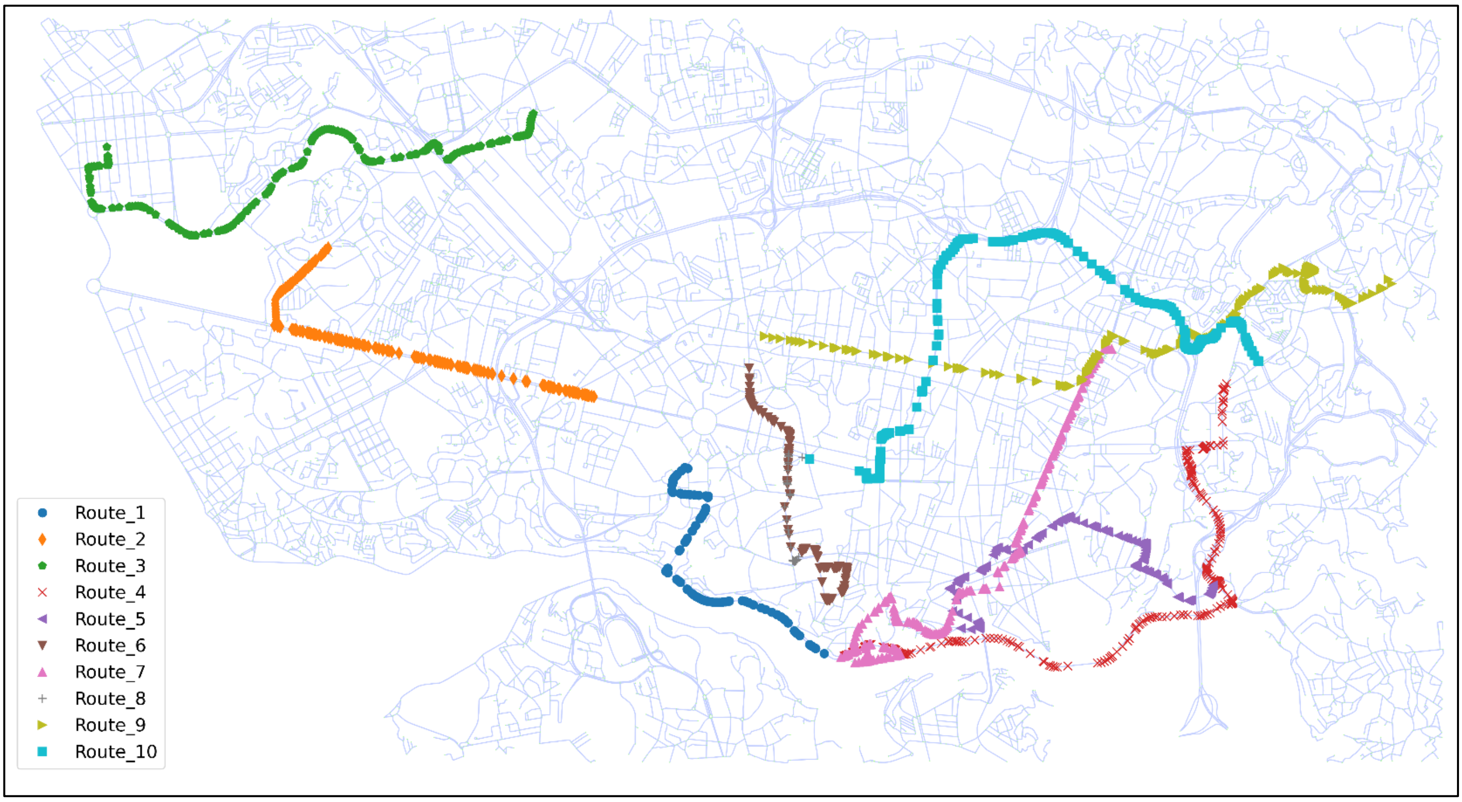

Figure 4-0

Figure 4-0 shows the first 10 mapped routes plotted together in one figure. One thing to note is that Route_6 overlaps with Route_8, making Route_8 harder to be seen. Figures 4-1 to 4-10 shows the first ten mapped routes separately.

Figure 4-1

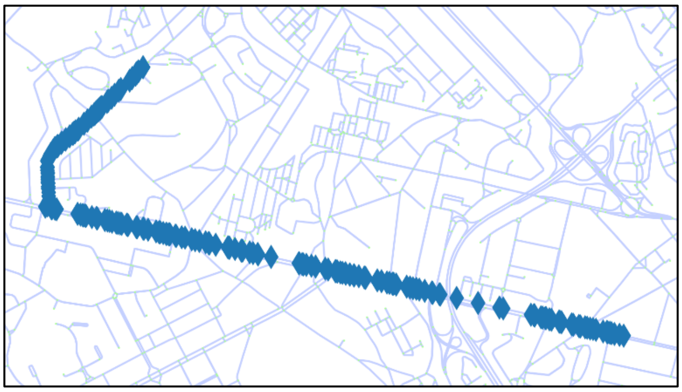

Figure 4-2

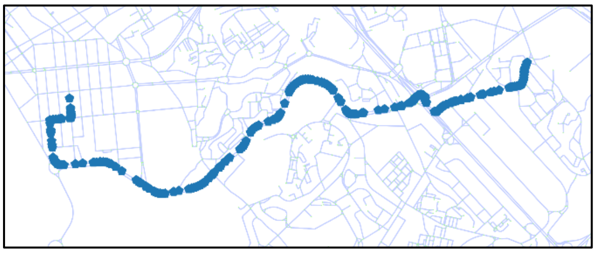
Figure 4-3


Figure 4-4

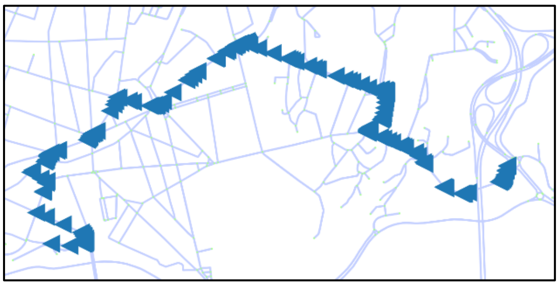
Figure 4-5


Figure 4-6

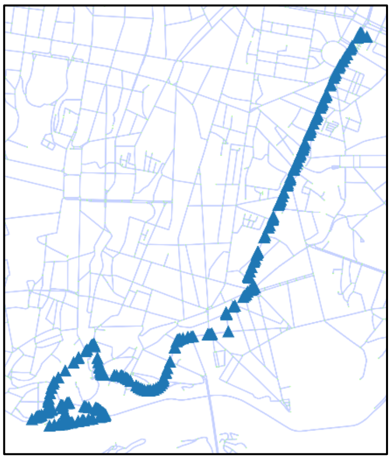
Figure 4-7

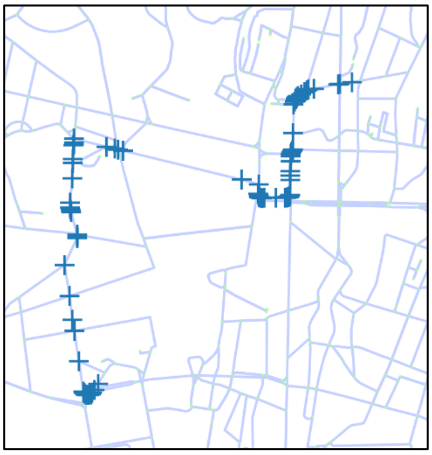
Figure 4-8


Figure 4-9

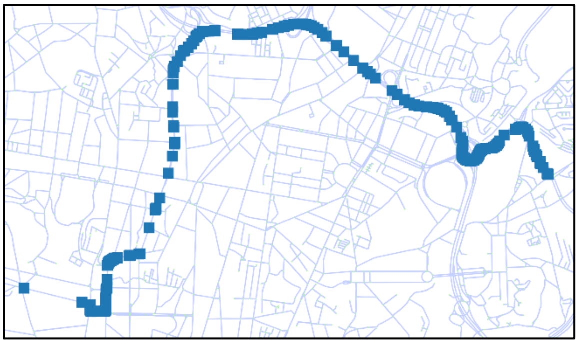
Figure 4-10

Task 5 Route Analysis

5a) 5 most traversed road segments

Figure 5-1 shows the 5 most traversed road segments in one plot. One thing to note is that these 5 road segments lead into and away from the Porto airport.

5b) 5 road segments with largest average travelling time

Figure 5-2 shows the top 5 road segments with largest travelling time in one plot while Figures 5-3 to 5-7 shows them plotted separately.

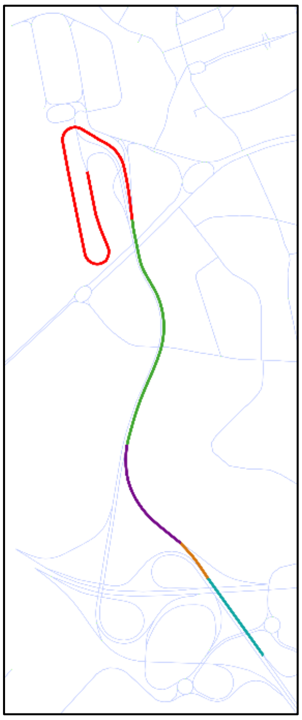

Figure 5-1

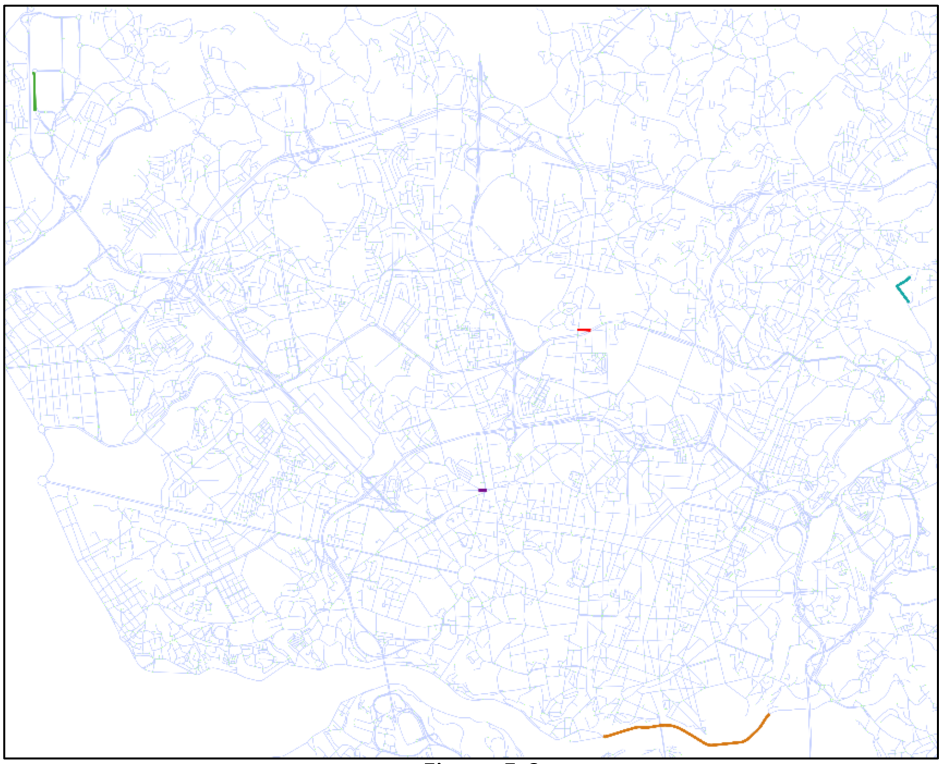

Figure 5-2

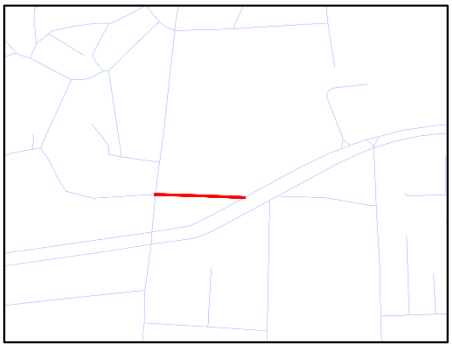

Figure 5-3, Rua do Carriçal

Figure 5-4, Avenida Doutor Óscar Lopes

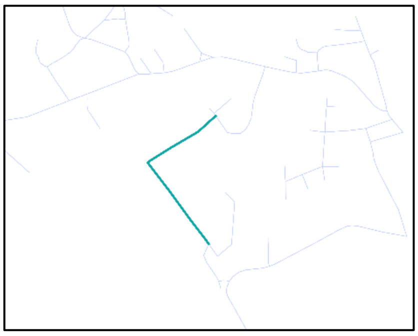

Figure 5-5, (No Name)

Figure 5-6, Avenida Gustavo Eiffel, Avenida de Paiva Couceiro

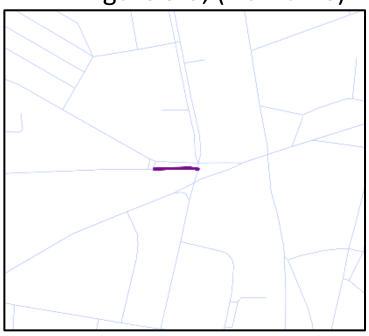

Figure 5-7, Rua da Prelada

Task 6 Outliers Analysis

Figure 6-1 to 6-10 shows the superimposed plots of GPS points, mapped nodes and mapped segments for each trajectory. Plots of three different methods for trajectory 1, 2, 5, 6, 7, 8, 9 matches almost perfectly. However Figure 6-3, 6-4, 6-10 for trajectory 3, 4 and 10 shows that GPS and Mapped Nodes matches perfectly but the Mapped Segments is not correct and is actually very far. As FMM has managed to map most of the segments correctly using Hidden Markov Model, further research is required to investigate the reason behind the huge discrepancies between the mapped segments and the GPS plots and mapped nodes.

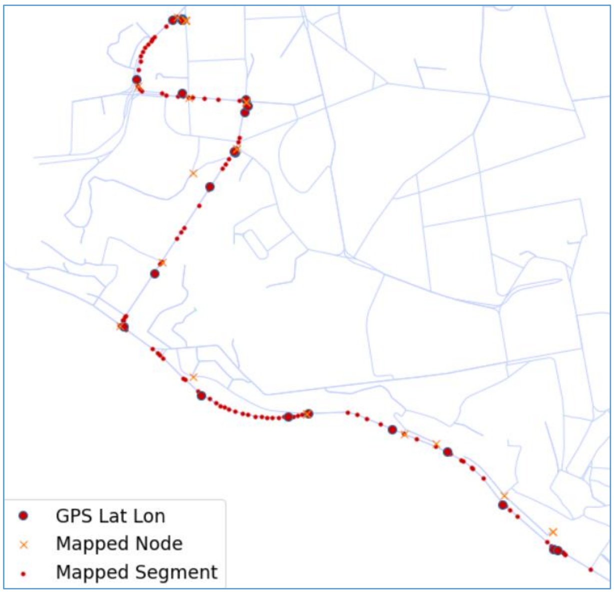

Figure 6-1: Combined Plots for Trajectory 1

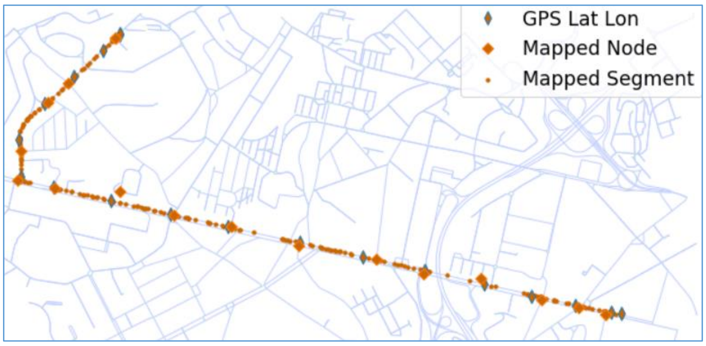

Figure 6-2: Combined Plots for Trajectory 2

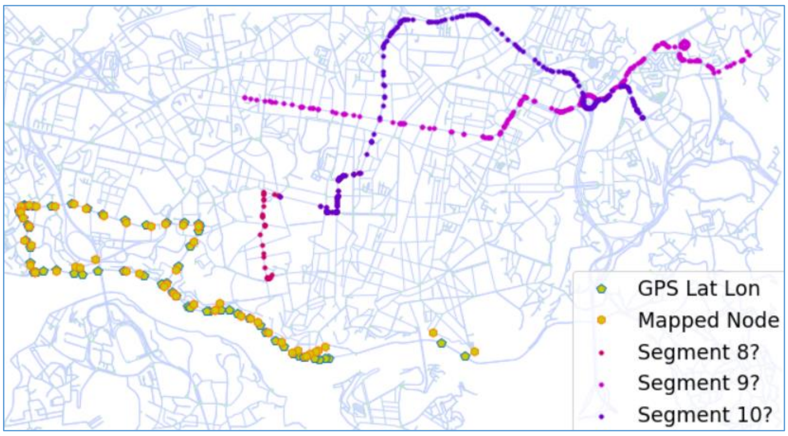

Figure 6-3: Combined Plots for Trajectory 3

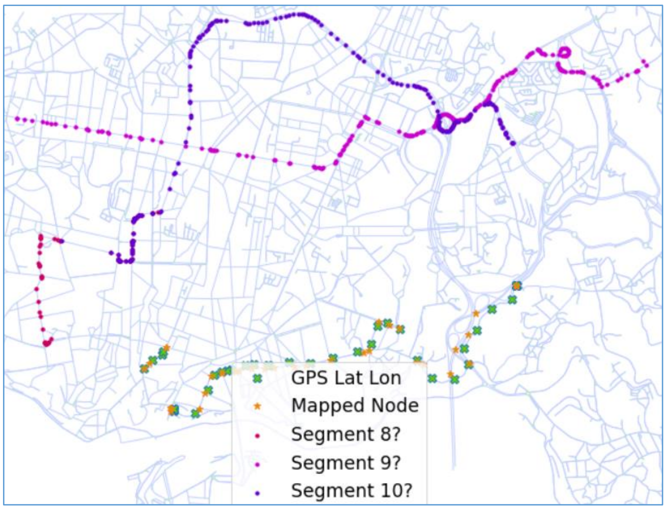

Figure 6-4: Combined Plots for Trajectory 4

Figure 6-5: Combined Plots for Trajectory 5

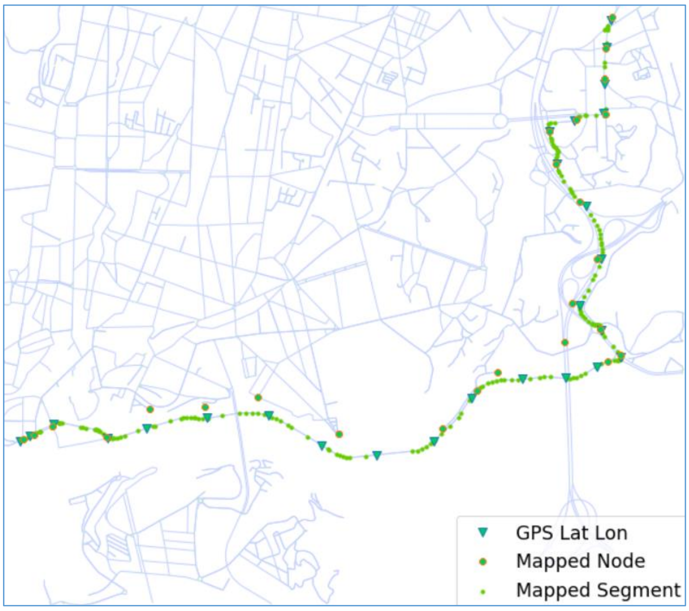

Figure 6-6: Combined Plots for Trajectory 6

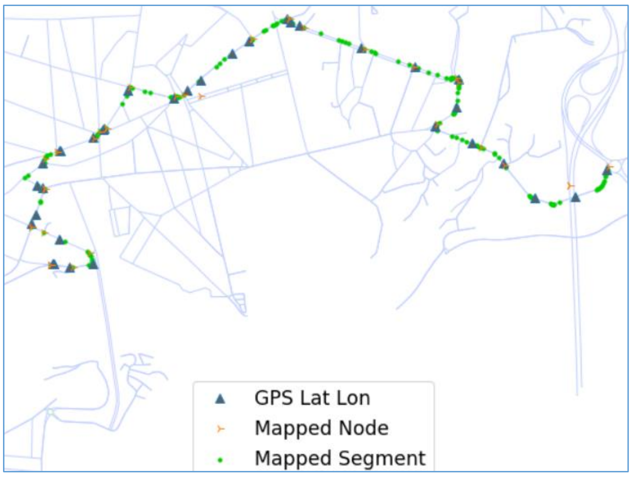

Figure 6-7: Combined Plots for Trajectory 7

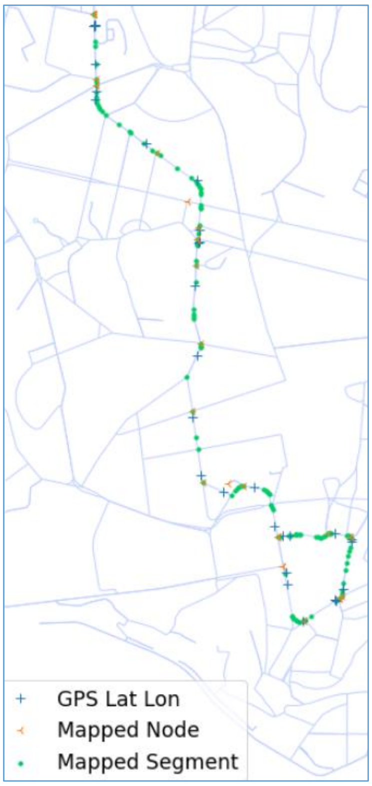

Figure 6-8: Combined Plots for Trajectory 8

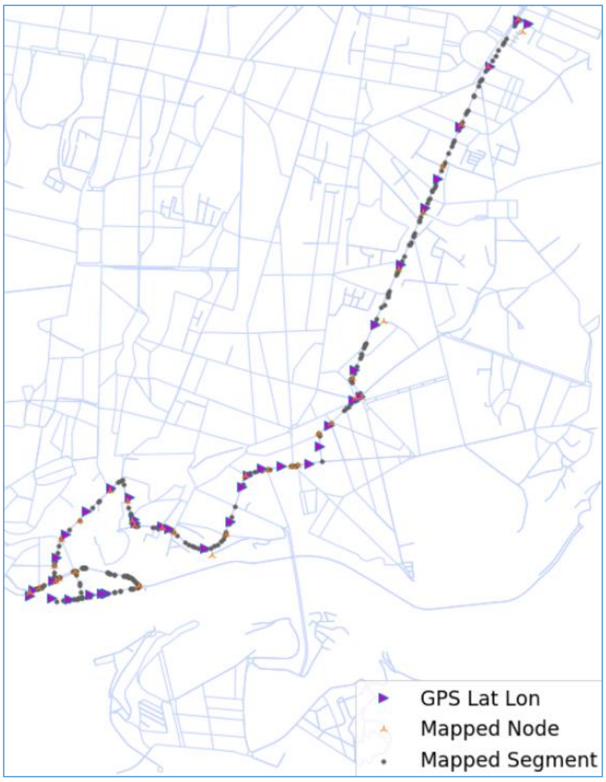

Figure 6-9: Combined Plots for Trajectory 9

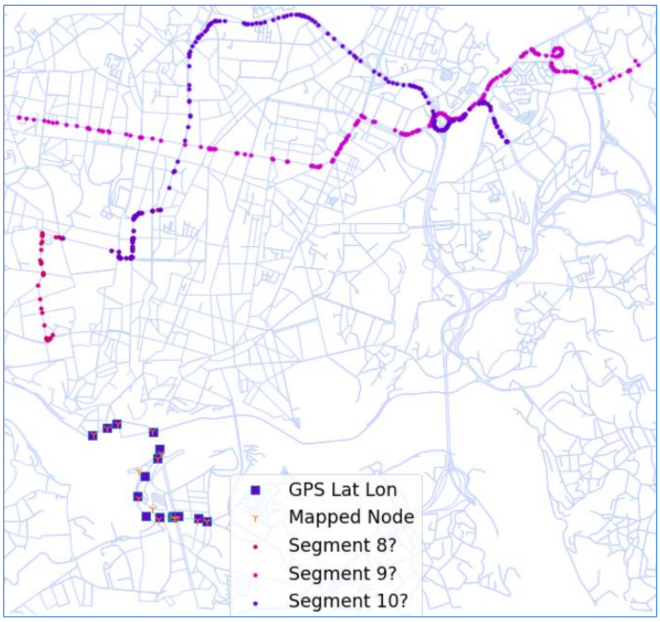

Figure 6-10: Combined Plots for Trajectory 10

Conclusion

In summary, the tracking of the taxi or vehicle position can be supported by a map-matching algorithm. In this project we make use of various state of art algorithms and tools that help us in processing the GPS signals, visualising the results, and plotting them onto the maps. We then do the route visualisation and route analysis

We realised that in order to get a good “Map matching” the quality of GPS signal is important. This is related to task 1 whereby the noisy GPS is filtered off first prior feeding in for analysis. Secondly the quality of map itself will affect the outcome of the Map matching as the say go,” Garbage in, garbage out”

In terms of tools, we found that OSMNX is a very useful and required short learning curve as it is an open-source package using python library that comes with simple application programming. The aesthetic of the graph is, however, can be complemented by other graph visualisation software such as folium.

The Map matching tool that we used in this project is Fast map matching open-source tool. This tool has the advantage of speed. Its unique selling point is its capability to handle a large amount of GPS data in a short time. This criterion is much sought after in terms of real time processing of GPS signal for map matching. The disadvantage of this tool is that the setting up of the tool is not smooth. The python version is quite old at 2.7 (Presently it is 3.10.0) and OSMNX that can couple with is 0.16.2. (The latest version is 1.1.2)

In terms of route accuracy, HMM model might suffer from “junction-decision matching” issue [4]. According to the author, the mismatch caused by the HMM model could amount to 13.8 %. The “Junction decision domain and hidden Markov model” claimed to improve the normal traditional map matching method which constitute a good candidate for future study.

References

1. Jensen C., Tradišauskas N. (2009) Map Matching. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_215

2. Zhong Zheng, Soora Rasouli, Harry Timmermans, Evaluating the Accuracy of GPS-based Taxi Trajectory Records, Procedia Environmental Sciences, Volume 22,2014, Pages 186-198,

3. Yang, Can & Gidófalvi, Győző. (2017). Fast map matching, an algorithm integrating hidden Markov model with precomputation. International Journal of Geographical Information Science. 1-24. 10.1080/13658816.2017.1400548.

4. Qi H, Di X, Li J (2019) Map-matching algorithm based on the junction decision domain and the hidden Markov model. PLoS ONE 14(5): e0216476. https://doi.org/10.1371/journal.pone.0216476