

UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Careelika Liisi Kuik
**Platform for Trajectory Reconstruction Using
Sparsely Sampled GPS Data**
Bachelor's Thesis (9 ECTS)

Supervisor(s): Amnir Hadachi, PhD

Platform for Trajectory Reconstruction Using Sparsely Sampled GPS Data

Abstract:

As the devices having GPS receivers are becoming more common, collecting and analysing GPS data is increasing as well. The analysed data can be used to give the user information about travelled trajectories.

Finding the exact travelled real trajectory can be difficult to achieve as the GPS signals may be sparse or missing in some areas. Using the shortest possible path between the start and destination does not always lead to a correct result. One way to improve the accuracy of determining the correct trajectory is to consider the travel time of the user as well.

The aim of this thesis is to propose a platform for trajectory reconstruction using sparsely sampled GPS data. Additionally, it aims to propose a method that uses the temporal aspect of GPS data for trajectory reconstruction.

As a result of this thesis work a platform for trajectory reconstruction is created and the method for trajectory reconstruction is proposed. The platform enables to reconstruct the real trajectory travelled by the user. Instead of using only shortest path search the decision making algorithm uses the real travel time of the user to find more probable trajectory travelled by the user.

The provided platform enables to analyse and visualise GPS data and reconstruct the trajectory more accurately than it would have been performed without including the temporal aspect of GPS data.

Using more accurate method to reconstruct trajectories using GPS data makes finding the real trajectory more precise. Hence, the quality of the information given to the user improves.

Keywords:

GPS data, trajectory reconstruction

CERCS: P 170

Platvorm trajektoori rekonstrueerimiseks hajusalt esitatud GPS-andmeid kasutades

Lühikokkuvõte:

GPS vastuvõtjat sisaldavate seadmete populariseerumisega kaasneb ka GPS-andmete kogumise ja analüüsimise suurenemine. Analüüsitud andmeid saab kasutada kasutajale läbitud teekonna kohta informatsiooni andmiseks.

Läbitud teekonna täpne leidmine võib osutuda keerukaks, sest GPS-signaalid võivad mõnes piirkonnas olla hajusad või puudulikud. Algus- ja lõpp-punkti vahel kõige lühema võimaliku teekonna kasutamine ei vii alati õige tulemuseni. Üks viis parandada õige teekonna leimise täpsust on arvesse võtta ka kasutaja poolt teekonna läbimiseks kulunud aega.

Selle bakalaurusetöö eesmärk on välja pakkuda platvorm teekonna rekonstrueerimiseks kasutades hajusalt esitatud GPS-andmeid. Lisaks pakutakse välja meetod, mis kasutab trajektoori rekonstrueerimisel GPS-andmete ajalist aspekti.

Selle bakalaureusetöö tulemusena luuakse platvorm teekonna rekonstrueerimiseks ja pakutakse välja meetod teekonna rekonstrueerimiseks. Platvorm võimaldab rekonstrueerida kasutaja poolt läbitud teekonna. Selle asemel, et kasutada ainult lühima tee otsimist, võtab otsust tegev algoritm arvesse ka kasutaja poolt teekonna läbimiseks kulunud aega, et leida tõenäolisem tee.

Platvorm võimaldab analüüsida ja visualiseerida GPS-andmeid ja rekonstrueerida trajektoori täpsemini kui ainuüksi kõige lühema võimaliku teekonna kasutamisel ajalist aspekti arvesse võtmata.

Täpsema meetodi kasutamine tee rekonstrueerimisel kasutades GPS-andmeid muudab õige trajektoori leidmise täpsemaks. Tänu sellele paraneb kasutajale antava informatsiooni kvaliteet.

Võtmesõnad:

GPS-andmed, trajektoori rekonstruktsioon

CERCS: P 170

Table of Contents

Table of Contents	4
List of Abbreviations	5
1. Introduction	6
1.1 General view	6
1.2 Objectives.....	7
1.3 Contributions.....	7
1.4 Road Map.....	7
2. Background and Related Work.....	8
2.1 Introduction	8
2.2 Key Terms – Definitions	8
2.3 Related work	8
2.3.1 Collecting GPS Data.....	8
2.3.2 Pathfinding implementations	8
2.3.3 Trajectory Reconstruction	9
2.4 Conclusion.....	9
3. System Architecture	10
3.1 Introduction	10
3.2 The General Overview of the Architecture	10
3.3 The Source Data for Road Network.....	10
3.3.1 OSM Data Files	11
3.4 Database and Related Tools	12
3.5 Algorithm for Trajectory Reconstruction.....	12
3.6 Visualisation Process.....	13
4. Testing, Results and Analyses	15
4.1 Introduction	15
4.2 Extracting the OSM Data	15
4.3 Visualising the Data	15
4.4 Trajectory Reconstruction	17
4.4.1 Test Strategy	18
4.4.2 Test Results.....	18
5. Conclusion and Future Work.....	22
References.....	23
Appendix.....	25
I. License	25

List of Abbreviations

ITS	Intelligent Transportation Systems
OSM	OpenStreetMap
SQL	Structured Query Language
GPS	Global Positioning System
XML	Extensible Markup Language
API	Application Programming Interface
URL	Uniform Resource Locator
LTS	Long Term Support
CSV	Comma Separated Values
PL/SQL	Procedural Language/Structured Query Language
GB	Gigabyte

1. Introduction

1.1 General view

With the ever increasing digitalisation of everyday life also road maps are being digitalised and provided with additional data concerning the roads. Besides, the start and destination points of the roads as well as data like the speed limit on the road exist in the data set of the digital map. Next to it the freely accessible Global Positioning System (GPS) [1] is being more and more frequently included to the devices that people use in their everyday life. The increase of the production of these kind of devices together with their incremental usage creates a possibility to gather and a massive amount of GPS data. One of the ways to illustrate the use of this data is by creating a digital map on the top of web application such as OpenStreetMap (OSM) [2]. OSM is an open source project that is built by a community of mappers around the world. The contributors add data about roads, buildings, sightseeings and other objects of interest. The collected GPS data can be used to analyse the traffic on the roads and to improve the existing data. As the GPS receiver works instantly, it is possible to give its user real-time information using GPS data. The GPS receiver positions its user in two-dimensional space, enabling coordinates returned from one of the GPS satellites. The consequent signals gathered by the device, can be used to reconstruct the trajectory travelled by the user. The knowledge about the movement of the user holding a device that has a GPS receiver can be used to improve the trueness of the information given to the user. For instance, if the data is used to match the GPS readings to a digital map, it is important to specify the correct trajectory to give the user accurate information about its movement on the map. This can in some cases be difficult as the information about the user's location is not received constantly, rather after some time interval. Furthermore, the characteristics of the surroundings of the user may worsen the signal and make their frequency even sparser. For example, in urban areas obstacles like trees, constructions and signals provided by other devices prevent the device from sending and receiving the signals. Thus the information about location of the user on each moment is not received and the potential trajectory travelled by the user during this time interval has to be reconstructed using estimation. One more step in the process of trajectory reconstruction is mapping the coordinates of the user to the digital map. Further, depending on the length of the particular interval, the selection of the potential path travelled by the user can either be easier or harder. Generally, the longer is the time interval between the two consecutive signals, the more difficult is the decision making to choose the accurate path from the set of all possible paths that the user could have travelled within the given time and in this current location. To make the decision making more precise, the additional information from the data of the consecutive signals combined with the records provided by the digital map can be used. One of the aspects to take into account in order to improve the decision making for trajectory reconstruction is the temporal aspect. As the information about the speed limits on the roads of the digital map is freely available and in some cases already mapped to the digital map's records, it is possible to use this data together with the data received from the device to compute the estimated distance travelled by the user. This computed parameter allows decreasing the amount of possible candidate paths. Nevertheless, the given speed limits are rather theoretical and do not always depict the real situation in the traffic. For this reason the usage average speed computed from the data helps to make a better decision based on more probable speed chosen by the user than the maximum speed limit allowed in the given road.

1.2 Objectives

The main purpose of this thesis is to provide a platform for trajectory reconstruction using sparsely sampled GPS data. In this thesis, an alternative method to find the user's real travelled path by emphasising on the temporal aspect of data, is provided.

To achieve the main objective of the thesis, there is a need to use and create a digital map for the purpose of GPS data manipulation and representation. Moreover, for creating and finding the right path between the sparse GPS data a routing algorithm is needed. This latter has to be also evaluated and enhanced by integrating the temporal aspect.

Finally, for illustration and evaluation of the final output a visualization tool is a necessity in this work.

1.3 Contributions

The contributions of this thesis can be resumed as follows:

- Creation of a platform for trajectory reconstruction and visualization using sparsely sampled GPS data. The platform uses QGIS as host environment, stores data in PostgreSQL database and performs trajectory reconstruction using an algorithm implemented in Java.
- Implementation of modified algorithm for pathfinding in order to find all possible path which are used later by the main algorithm to extract the estimate trajectory.
- Creation of an algorithm that takes into account the temporal aspect and estimate the correct trajectory using sparse data.

As a result the method finds the real trajectory travelled by the user more accurately than by simply applying the shortest path algorithm.

1.4 Road Map

This thesis begins with an introductory chapter 1. Then, the second chapter of this thesis describes the state-of-art of the current topic and the related works that have been performed until the current moment. The topics covered are associated to collecting GPS data, pathfinding implementations and trajectory reconstruction.

The third chapter consists of the description of system architecture. More precisely, it describes the source data for road network representation, the database and the selection of the related tools. Lastly, the visualisation process and the selection of its tools are mentioned.

The fourth chapter describes and presents the results of implementing the designed system architecture and how the extraction of the OSM data is described. Next, the implementation of the visualisation process is described. Lastly, trajectory reconstruction is described, test strategy and test results are presented and analysed.

The final, fifth chapter makes a conclusion of the thesis work and states the perspectives for future work.

2. Background and Related Work

2.1 Introduction

This chapter covers state of the art related to the thesis topic. Therefore. The chapter is divided into four sections. Firstly, the key definitions used in the context of this work are provided. Secondly, the related work concerning the collection of the data is described. Thirdly, related work regarding the pathfinding algorithms are described. As there are many more pathfinding algorithms than described in this section, only the ones that are most similar to the algorithm proposed in this thesis are introduced. Finally, the related work concerning trajectory reconstruction is described.

2.2 Key Terms – Definitions

Node: is defined as a pair of x and y coordinates on planar graph and is used to provide localization on the map.

Way: is defined as an edge that connects two nodes on planar graph and is used to depict a road segment in form of polyline on the map.

Trajectory or path: is defined as a collection of one or more ways in a sequential order starting with the source node and ending with the target node on planar graph. It is used to depict directed sequence of ways on a map.

2.3 Related work

2.3.1 Collecting GPS Data

To begin with, GPS data can be obtained in different ways. The most common is to use the data provided by the device's GPS receiver. This works in case the object under inspection is for example a mobile phone that has a GPS receiver. However, in traffic management not in all cases are the objects under inspection devices with GPS receivers. For example in case of vehicles the presence of GPS receiver cannot be assumed. In this case alternative ways like using toll readers [4] is possible. This however may result in inaccurate measurement as the sensors used for measurement are located not on the mainline but on the ramps. Secondly, collecting the data using a GPS receiver consumes the device's battery. This can result uncomfortable for the user as the battery life of the device decreases noticeably. To avoid this downside, the necessary data can be collected in a sparsely manner to reduce the energy consumption [5]. On the other hand, using this data collection method demands a fully-optimized design process like coordinating phone sensor types, sensing circles, etc. This results far more complex than the previous method. Also, this method does not provide more accurate data than the previous one. To sum up, collecting data without the use of the device's GPS receiver results in increased inaccuracy and taking into account that GPS receiver itself provides inaccurate in positioning anyway, it is difficult to provide correct results based on the data collected this way. For this reason, in this dissertation a solution to reduce this bad effect is proposed.

2.3.2 Pathfinding implementations

There are several ways to implement pathfinding for road networks. Vavpetič et al. [6] introduces two methods for Pathfinder algorithm for sparse data. One of them would be to use Binary Pathfinder, where initial matrix is generated, and the paths between nodes are improved with each round. The second one, a modification of Dijkstra's algorithm [7], uses adjacency list with neighbouring nodes. The previously mentioned algorithms are reasonable

to use mainly in case of large amount of data nodes and are unreasonably complicated for smaller datasets. Besides, their usage is neither grounded nor efficient in case of small data sets. Furthermore, to improve the probability of choosing the correct path, additional parameters should be introduced. Doing so would decrease the efficiency even more and thus writing alternative algorithm for pathfinding were used in this thesis.

2.3.3 Trajectory Reconstruction

Wang et al. [8] propose trajectory reconstruction model using point-based speed data collected by existing Intelligent Transportation Systems (ITS). Additionally, it uses smoothing scheme that makes changes of speed less abrupt. Nevertheless, the suitable data for this model is not always available – it is difficult to find datasets containing point-based data and route-based data simultaneously. Furthermore, bi-level probability method can be used to reconstruct the trajectories [9]. This method uses shortest path algorithm and produces reasonable routes on complicated urban road network. Still, it does not always choose the most probable path. First of the above mentioned trajectory reconstruction models is restricted to GPS data in specific format and requires datasets containing point-based data and route-based data simultaneously. The method proposed in this thesis uses only point-based GPS data. In addition, not always the shortest path is used, but the travel time is taken into account as well.

2.4 Conclusion

There is a number of works related to the current topic, but all of them have downsides that can be surmounted. It is clear that there is a lot work done in this topic. However, the case studies and testing environment vary from paper to another. For this reason, the proposed approaches can have some limitation depending on the defined hypothesis or the nature of data itself. As a result, the method described in the current thesis aims to propose a solution that uses GPS data as the source data.

3. System Architecture

3.1 Introduction

In this chapter the system design and architecture of the platform proposed in this thesis is described. The chapter is divided into subsections according to the tools used in the creation of the platform and according to the steps passed to create the platform. In each subsection a small introduction to the tool is given, and also the reason for selecting this particular tool, the purpose and usage of this tool in the platform.

3.2 The General Overview of the Architecture

The general schema of the system architecture is shown of figure 1. The architecture of the system consists of three main parts which are PostgreSQL database, the program with the algorithm and QGIS environment.

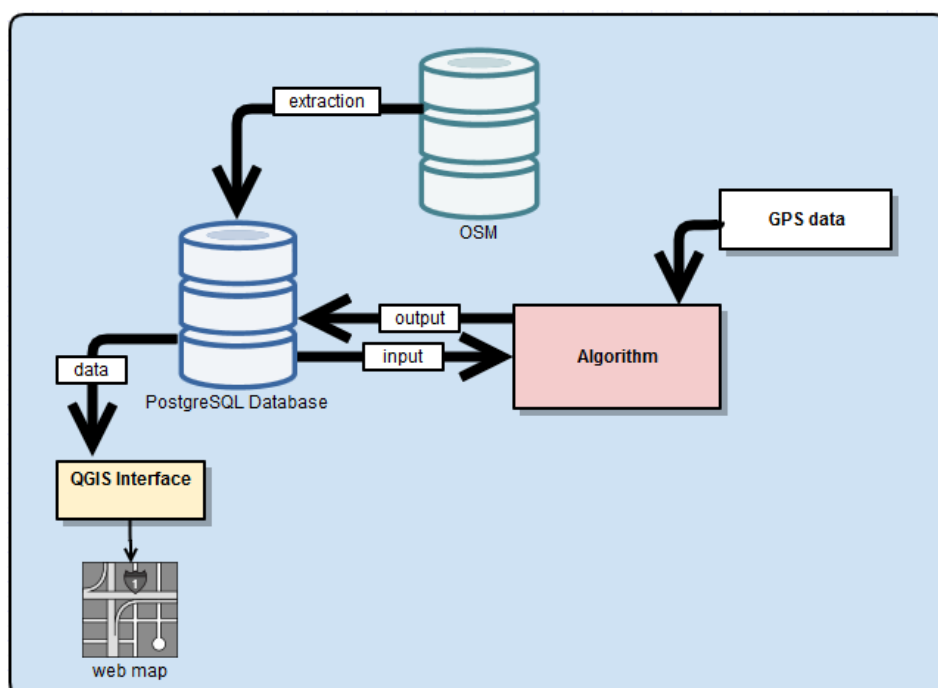


Figure 1. The system architecture.

3.3 The Source Data for Road Network

To begin with, the purpose of this platform was to provide tools for trajectory reconstruction and visualisation. It was necessary to decide right at the beginning the format of the geographic data files to be used. This latter is important, because the type of the extension of the digital maps has a direct influence on the tools to be used for GIS manipulation and programming the platform features. For this platform OSM provided road network maps were decided to use. OSM provides open source data and is being updated by volunteer collaborators. It has quite high coverage in comparison to real road networks, although quite a big number of the roads are still missing even in more popular regions. However, OSM digital maps have the advantage of being freely usable for any purpose and this fact makes its usage fast and clear. Other option would have been to use data provided by Google Maps [10]. Google Maps provides data with

high coverage regarding the real road network and is being updated frequently by the team of Google maps itself as well as by volunteer collaborators. Nevertheless, Google Maps has the disadvantage of having strongly regulated usage permissions. Thus, taking into account the possible further development of the created platform, the usage of open source data appeared to be a more reasonable choice.

3.3.1 OSM Data Files

As OSM provides free and open data, the contents of its database are freely accessible and downloadable for anybody. As the data provided by OSM is massive even in compressed form, exceeding 30 GB [11], and difficult to work with at once, it is reasonable to extract a smaller area for exporting. In this context, by extracting the data cutting the data from the area of one's choosing as well as pulling out the specific features that are wanted from the chosen area, is meant [11]. Exporting the data is defined as converting OSM data from its native XML format into a format that is convenient [11]. There are a number of tools that offer the option of selecting a precise area of OSM data and exporting it. The structure of a regular OSM file can be seen on figure 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" copyright="OpenStreetMap and contributions">
  <file>
    <infor str="The main road system in Hanoi. It includes 20 zones. " />
    <infor str="Author: Tuan Nam Nguyen&#10;Date: January 2015" />
  </file>
  <bounds minlat="20.911" minlon="105.685" maxlat="21.152" maxlon="105.953" />
  <node id="-28855" lat="21.0107278" lon="105.8373101">
    <tag k="intersection" v="yes" />
  </node>
  <node id="-29888" lat="21.0227201" lon="105.8575671">
    <tag k="highway" v="traffic_signals" />
  </node>
  <way id="-36069">
    <nd ref="-28855" />
    <nd ref="-29888" />
    <nd ref="-29889" />
    <tag k="oneway" v="yes" />
    <tag k="highway" v="primary_link" />
    <tag k="name" v="Khuất Duy Tiến" />
    <tag k="tm_connector" v="yes" />
  </way>
  <way id="-36364">
    <nd ref="-29778" />
    <nd ref="-28687" />
    <tag k="oneway" v="yes" />
    <tag k="highway" v="secondary" />
    <tag k="width" v="16" />
    <tag k="name" v="Trần Nhân Tông" />
  </way>
</osm>
</xml>
```

Figure 2. The structure of OSM file [12].

OSM file format is based on XML file format that enables data exchange between different software tools. The file contains data primitives like nodes, ways and relations. The set of

features contained in each OSM file can vary, as the features to be contained in the file can be chosen in the process of extracting the data. In the creation of this platform, the OSM file contains the three major topological data structure features: nodes, ways and relations. There is a number of possible ways to export the OSM data from OSM database. However, if it is important to export the data of a particular area, it can happen that the selected tool does not enable exporting the data of the selected area. For small amounts of data the usage of OSM API's *map* request or downloading via JOSM is suggested [13]. JOSM is an extensible open source editor for OSM. Among other features, it enables exporting and editing OSM data [14]. For larger amounts of data the usage of tools like Osmosis or Overpass API is suggested [15]. The Overpass API is a read-only API that acts as a database over the web. The data can be obtained by sending a query to the API. As a result, the query selected parts of the OSM map data are returned to the client as a data set [14]. The constructed query obtains the data set that corresponds to the area specified. The area has to be specified as a set of parameters for the URL. The parameters to be specified are the minimum longitude, the minimum latitude, the maximum longitude and the maximum latitude of the area to be downloaded.

3.4 Database and Related Tools

To efficiently use the data obtained with the OSM data file, a database with suitable extensions is needed. As PgRouting seemed to fill the conditions for the further work on road networks data, it was selected for this work. PgRouting is an extension to PostGIS that provides geospatial routing functionality. It helps to create data and attributes that can be modified by many clients like QGIS [16]. PostGIS is a spatial database extender for PostgreSQL which adds support for geographic objects and allows to run location queries in SQL [17]. PostgreSQL is an open source object-relational database system [18]. The workshop page of PgRouting [19] demonstrated using PostgreSQL with various tools, among which the usage of PostgreSQL and PostGIS are suggested. This page provides a good set of suggestions, however many of the suggested solutions are outdated or do not perform expected results with the current configuration, although the tool is created for PostgreSQL taking into account its properties. Not all versions of PostgreSQL support the named extensions. These extensions are supported for example by PostgreSQL 9.3 and 9.4. Unfortunately in Ubuntu 14.04 that was used to create the platform, PostgreSQL has missing features that prevent the usage of these tools. Hence, PostgreSQL 9.4 was selected for the platform.

For adding pgRouting functions to database pgAdmin III is used. PgAdmin III is an open source administration and development platform for PostgreSQL [20]. Using PgAdmin makes adding extensions to the database easier as well as provides a good visualization for the tables stored in the database and their contents. As OSM data cannot be just exported to the database another tool, called Osm2pgrouting is used to export the data. Osm2pgrouting is a tool that comes with pgRouting and enables easy exporting of the OSM data to pgRouting database [21].

3.5 Algorithm for Trajectory Reconstruction

The algorithm for trajectory reconstruction takes both start location and destination location as an input. The input data are GPS locations with the following information: latitude, longitude, timestamps, and headings. The timestamps are used to compute the real travel time TT_r spent to go from start location to destination location:

$$TT_r = |t_e - t_s|$$

Where, t_s is the timestamp at the start location and t_e is the timestamp at the destination location. Next, start and destination nodes are used in pathfinding algorithm to find the candidate trajectories. After finding all the candidates we use the observed speed collected from

real world data for each candidate trajectories. Then, estimated travel time TT_c of each candidate trajectories is computed using the average speed observed on each trajectory. The next step is to associate a weight w to the candidate trajectories and it is defined as follows:

$$w = \frac{TT_c}{TT_r}$$

where TT_c is the travel time of the candidate trajectory and TT_r is the travel time of the real trajectory. As a result of the comparison, the best weights is the weight of the estimated trajectory that is given as an output. In case the weights for different candidate trajectories happen to be the same, the candidate trajectory with shorter distance is chosen.

3.6 Visualisation Process

As the next step, the trajectories are visualised on a web map. The visualization is performed using QGIS (version 2.8.3). QGIS is a free and open source application that enables to create, edit, analyse, visualise and publish geospatial information [22]. The data points from the real data are added to QGIS from the CSV file containing the real data.

For exporting the QGIS project to a web map a Python plugin called qgis2web is used. It enables to generate a web map from the current QGIS project without requiring server-side software. There are two sorts of web maps to choose from while exporting the project. One of them is Leaflet which is generated using QGIS2Leaflet plugin. Leaflet is an open source JavaScript library that enables to create interactive maps [23]. However, it is deprecated [24] and does not function properly in the version of QGIS used for this platform. Another one is OpenLayers 3. OpenLayers 3 is an open source JavaScript library that has many mapping related features, among them creating interactive web maps [25]. In the process of exporting the QGIS project the layers and labels as well as other features to be displayed on the created web map can be specified. Moreover, additional features like background map layer can be added. There are a number of possible background maps offered in by the prompt, however only small amount of them actually work. An example of the exporting prompt window is shown on the figure 3.

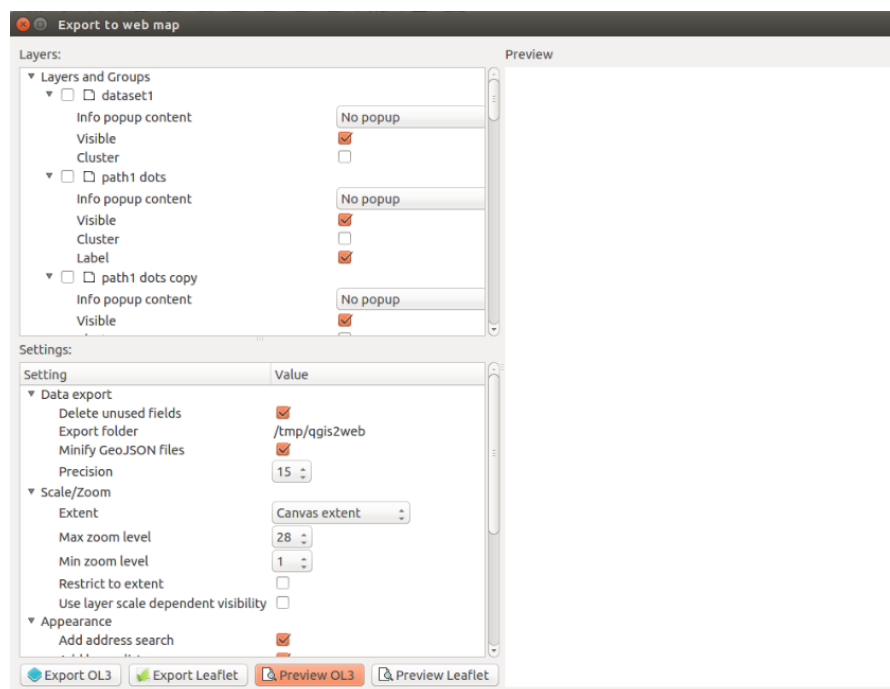


Figure 3. An example of the web map exporting prompt window.

The exported web map can be viewed in any of the most common web browsers as well as integrated with other software. As all the necessary data is added to the generated output folder, no additional sources but the generated folder are necessary to display the generated web map.

The data is visualized on the road network retrieved from the database table that contains all the possible ways. In addition the nodes with the corresponding labels are added as another layer. The road network and nodes layers are shown on figure 4.

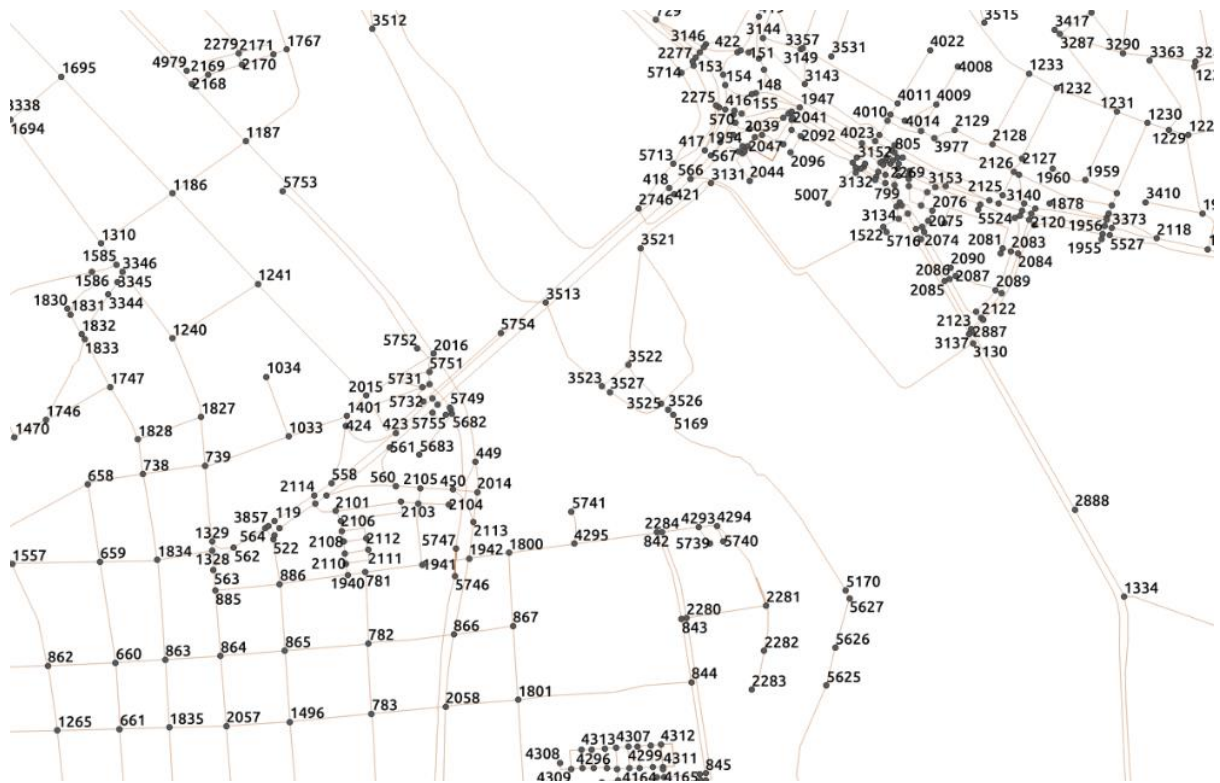


Figure 4. Road network layer with nodes and node labels layers.

There are the way segments obtained from the database on this layer as well as the conjunction nodes for the ways.

4. Testing, Results and Analyses

4.1 Introduction

In this chapter the testing, results and analyses of the platform proposed in this thesis is described. The chapter is divided into subsections according to the tools used in the creation of the platform and according to the steps passed to create the platform. In each subsection the usage of this tool in the platform, the problems faced during the implementation are described and the results are presented. The platform is tested under 64 bit Linux Ubuntu 14.04 LTS operating system.

4.2 Extracting the OSM Data

To manage the data provided by OSM and storing the test results, a database was created. Before creating the database, the tools mentioned in system architecture were installed. With PostgreSQL previously installed, the packages for PgRouting were installed with the command shown on figure 5.

```
sudo apt-get install postgresql-9.4-pgrouting
```

Figure 5. Commands used to install PgRouting packages.

Although PgRouting and PostGIS extensions are supported by the PostgreSQL version used in this work, some features of PostGIS still could not be used as the necessary features just did not appear in the menu where they should have appeared after the installation of the extension. The database to store the OSM data was created using the commands shown on figure 6.

```
createdb -U dbuser routingdb  
psql -U dbuser -d routingdb -c "CREATE EXTENSION postgis;"  
psql -U dbuser -d routingdb -c "CREATE EXTENSION pgrouting;"
```

Figure 6. The commands used for creating the database.

To import OSM data to the database, Osm2pgrouting tool was used from command line with the command line with the command shown on figure 7.

```
osm2pgrouting -file "./map.osm" \  
- conf "/usr/share/osm2pgrouting/mapconfig.xml" \  
- dbname routingdb \  
- user postgres \  
- host localhost \  
- clean
```

Figure 7. The command used for exporting OSM data to the database.

In this command the OSM data file and database parameters are given to the tool as an input. As a result, the database is populated with the tables created from the OSM data and additionally the necessary attributes to perform shortest path queries, are added.

4.3 Visualising the Data

The visualisation was performed using QGIS. As the estimated trajectory was stored in the database as a separate table, visualising the trajectory could be done by importing the

corresponding table to QGIS project. To visualise the ground truth trajectory, the CSV file entries were visualised on the map. As the GPS locations are not precise and often have erroneous locations, matching the GPS data with map appeared to have difficulties. As map-matching was not the purpose of this thesis, the problem was solved intuitively matching the unprecise locations to the closest way segment on the map. An example of the GPS data and corresponding trajectory is shown on figure 12.

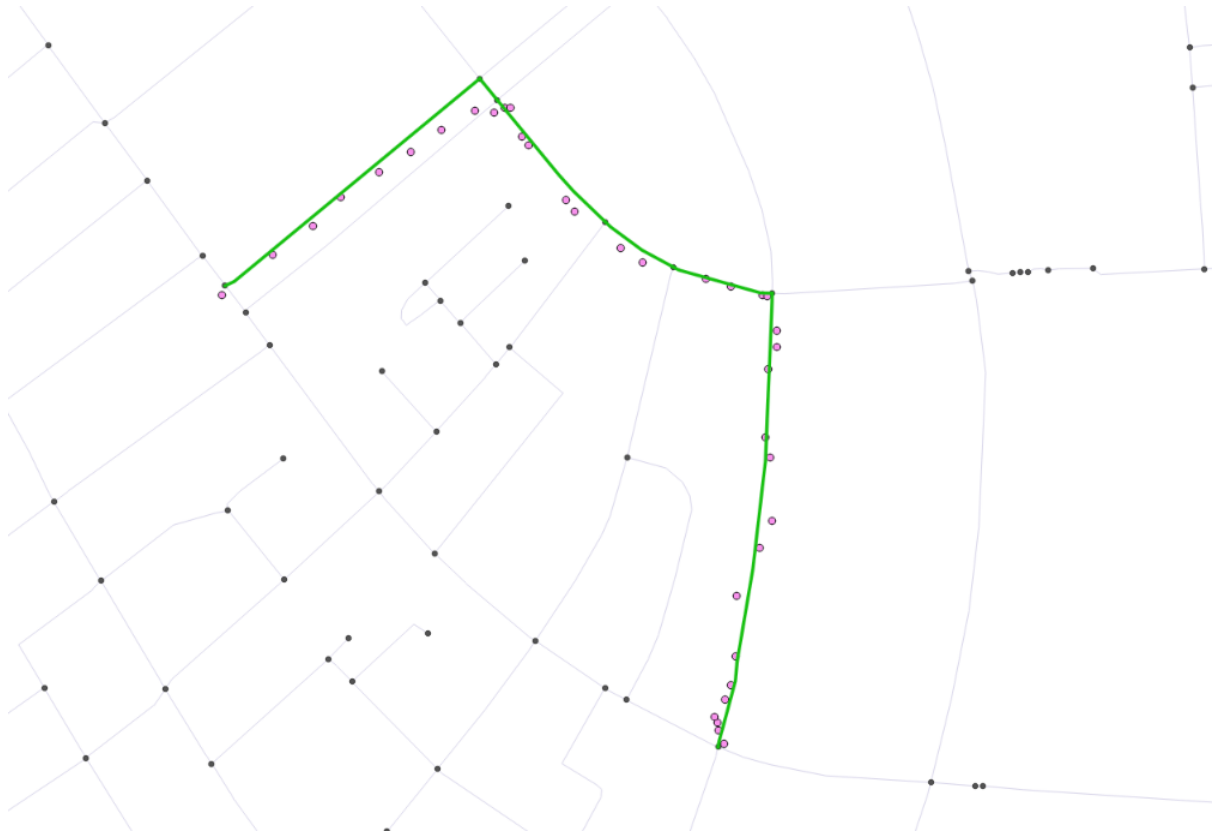


Figure 12. An example of the GPS data and corresponding trajectory visualised. The pink dots represent the GPS data and the green line marks the way segments that correspond to the data.

As a base layer a layer with all the ways in the database is added. Furthermore, the map composed by the layers was exported to an interactive web map visible in browser using the plugin qgis2web. More precisely the template using OpenLayers3 JavaScript library was selected. On figure 13 an example map with the ground truth trajectory and estimated trajectory is shown.

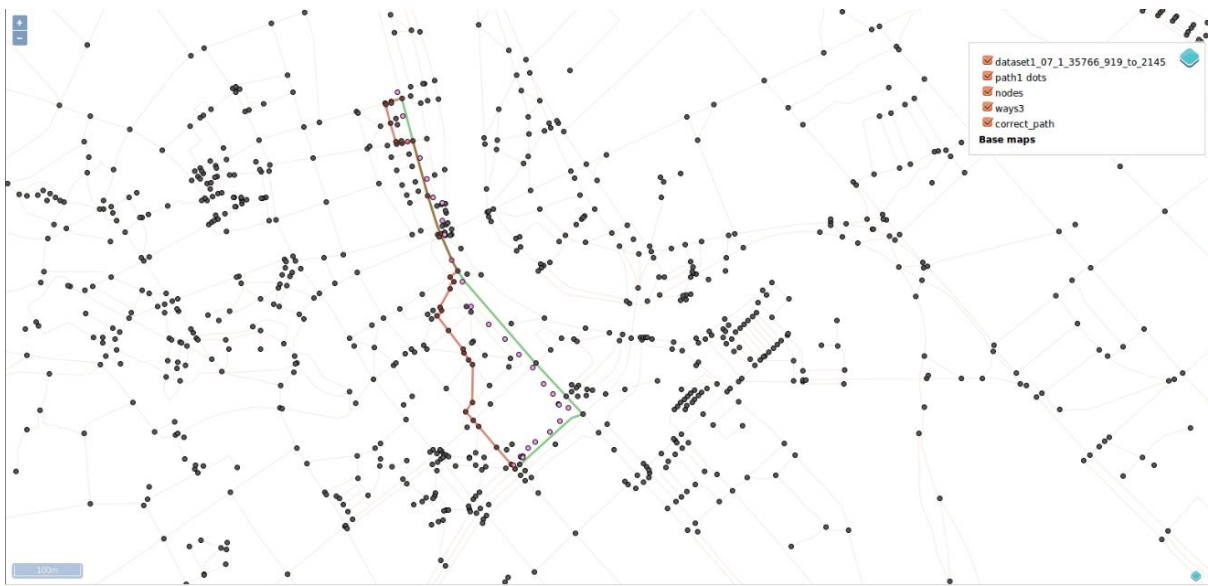


Figure 13. An example web map visualized. There is a legend added to the upper right corner of the map.

The map is usable with different browsers and the layers added to the map can be viewed either separately or all at the time.

4.4 Trajectory Reconstruction

For testing real GPS data is used. The data is located in CSV files. Figure 9 shows an example CSV file content.

```
1;2015-03-21 15:20:16.0;5;58.40442718;26.67450439;27.5;37.0;338.799987793
1;2015-03-21 15:20:22.0;5;58.4058653;26.67348991;28.0;20.0;339.299987793
1;2015-03-21 15:20:30.0;5;58.40767944;26.67213849;27.5;43.0;338.299987793
1;2015-03-21 15:20:34.0;5;58.40863141;26.67147437;28.25;25.0;339.299987793
1;2015-03-21 15:20:41.0;5;58.41024559;26.67024502;28.0;26.0;338.0
1;2015-03-21 15:20:52.0;5;58.41277789;26.6686336;28.25;63.0;339.200012207
1;2015-03-21 15:21:04.0;5;58.41555058;26.6664538;27.5;44.0;337.200012207
```

Figure 9. The example of the format of a CSV file used as the input of the algorithm.

On the first line the names of the data fields given below are stated. The file contains one entry on each line. Every entry consists of the data related to one GPS receiver signal. Each entry has speed, location of the measurement in form of longitude and latitude, timestamp and some other fields that are not used by the algorithm. As different GPS data files have slightly different fields, the functions for reading in the data and processing it have to be slightly different as well. In some example files time is given in milliseconds, in some in in ISO 8601:1988 format [26] (yyyy-MM-dd HH:mm:ss.SSS) and in some cases in both formats. Another problem faced while testing was the differences between the data units. For example time can be given in form of milliseconds while the length of the segments is computed in meters and the speed is given in meters per second. As OSM is an open source free data, the data is not being updated systematically and hence is not complete. There are a significant number of roads that exist in the real road network but are not present in OSM data. Furthermore, many streets are marked as one-way streets even though they are actually two-way streets. For more accurate test results the missing way of the streets marked as one-way street was added.

4.4.1 Test Strategy

For testing 22 trajectories were extracted from different raw GPS data files. The trajectories were selected randomly, trajectories of different length were chosen. The number of way segments in the chosen trajectories was between 3 and 17. The start and destination locations for each trajectory were extracted from the data. The observed average speed for every trajectory was computed as well. As the GPS data most often has measuring errors, the selection of the start and destination locations was corrected intuitively, if did not correspond to the real trajectory. From the ground truth trajectories real travel time was extracted. The start and destination locations and real travel time were given to the program as an input. As a result estimated trajectory was returned by the program. Finally, the ground truth trajectory was compared to the estimated trajectory.

4.4.2 Test Results

Testing showed that on almost 50% of the test cases the correct trajectory was found. The average accuracy of the proposed algorithm was about 64.6%. For more details, the percentages of the correct segments found by the algorithm are shown in the table 1.

Table 1. The test results.

No of Segments	No of Correct Segments Found	Correctness Percentage
7	7	100.00%
6	6	100.00%
5	5	100.00%
3	3	100.00%
5	5	100.00%
7	7	100.00%
3	3	100.00%
5	5	100.00%
4	4	100.00%
7	7	100.00%
12	9	75.00%
16	6	37.50%
15	4	26.70%
13	4	30.80%
7	3	42.90%
8	6	75.00%
15	8	53.30%
9	3	33.30%
16	0	0.00%
10	3	30.00%
17	3	17.60%
8	0	0.00%

One of the cases where the algorithm found the correct trajectory, is shown on figure 10.



Figure 10. The test case with all correct way segments found. The pink dots represent the GPS data and the green line marks the way segments that correspond to the data. The red line marks the trajectory found by the algorithm.

Next, the observation of the test cases where the correct trajectory was not found was done. Instead of comparing only whether the complete trajectory was found correctly, it can be estimated how many segments of the found trajectory corresponded to the ground truth trajectory. As can be seen from the table, in such case the minimum percentage of correctness was 0% and the maximum percentage of correctness was 75%. In two cases any of the segments were found correctly. The ground truth trajectory and estimated trajectory of one of these cases is shown in figure 11.

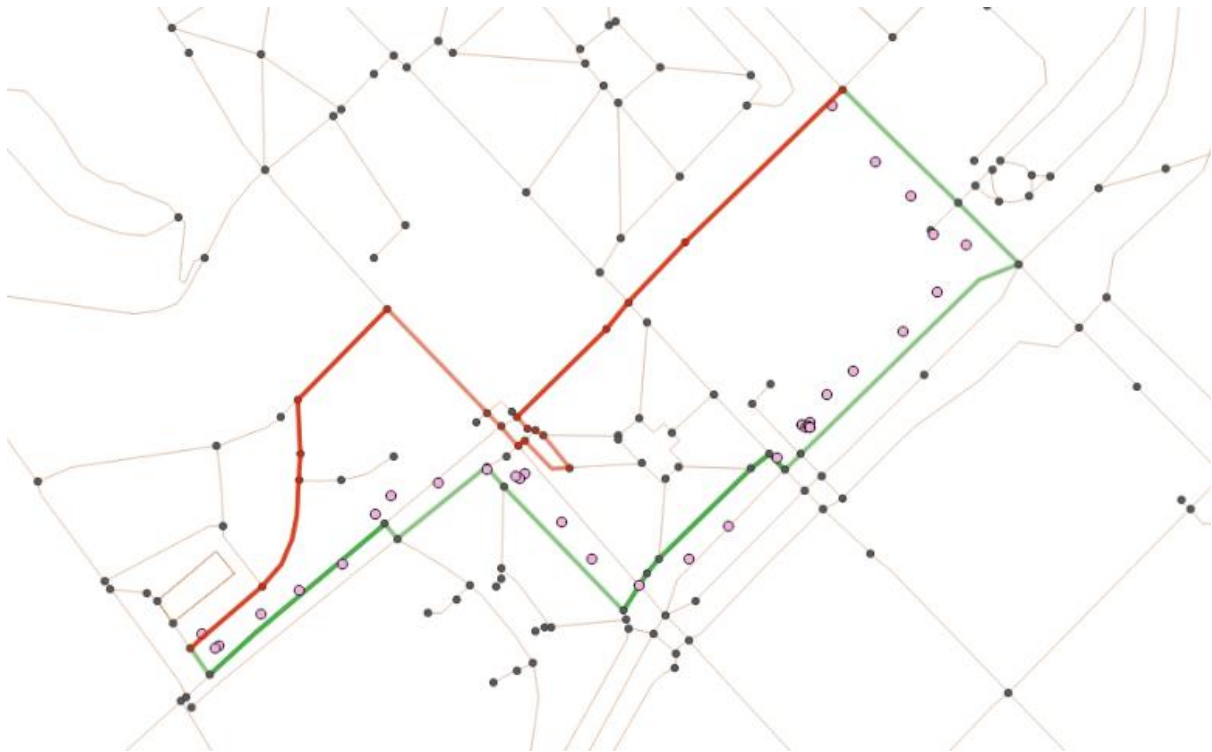


Figure 11. The test case with 0% of correct way segments found. The pink dots represent the GPS data and the green line marks the way segments that correspond to the data. The red line marks the trajectory found by the algorithm.

One of the cases where 75% of the way segments were found correctly is shown in figure 11.



Figure 11. The test case with 75% of correct way segments found. The pink dots represent the GPS data and the green line marks the way segments that correspond to the data. The red line marks the trajectory found by the algorithm.

Furthermore, as can be seen from the table the percentage of correct way segments found do not depend completely on the total number of the segments in the trajectory, although a connection can be seen. The proposed algorithm is using the temporal aspect of GPS data and therefore finds more accurate results than it would have found with using only shortest path search. The figure 12 shows an example of a test case where the algorithm returns a more accurate trajectory than it would have returned when the temporal aspect of the GPS data would not have been taken into account.

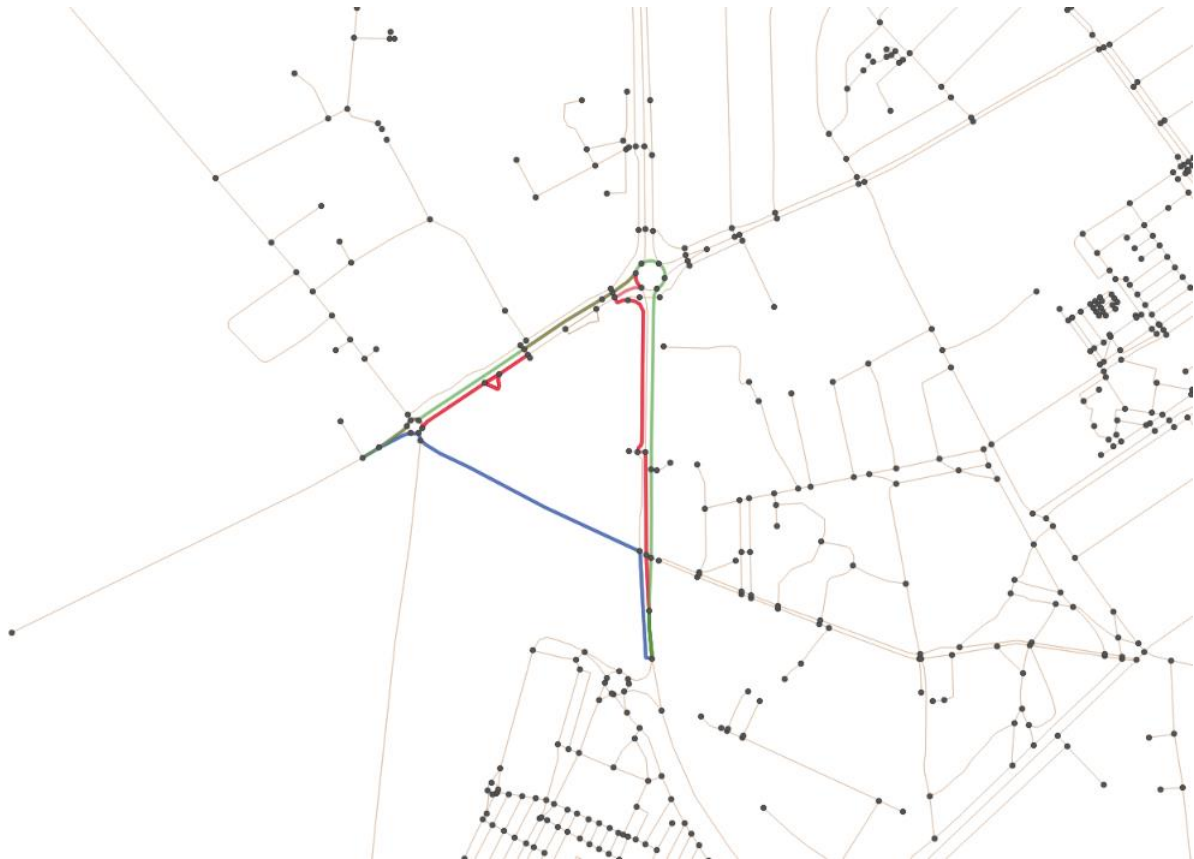


Figure 12. An example of test case where the algorithm finds more correct trajectory than the shortest path. The green line marks the way segments that correspond to the data. The red line marks the trajectory found by the algorithm. The blue line marks the trajectory with the least distance possible.

As can be seen from the figure 12, the trajectory with the shortest distance between given start and destination differs immensely from the real trajectory travelled by the user. The trajectory returned by the proposed algorithm returns trajectory that widely corresponds to the real trajectory.

5. Conclusion and Future Work

The usage of the devices with GPS receiver is constantly increasing and the obtained GPS data is analysed and more frequently used for different purposes. Hence, there is a need to improve the tools, techniques and methods to perform these processes as well. Better methods improve the accuracy of the information given to the user and lead to more favourable user experience.

The aim of this thesis was to create a platform for trajectory reconstruction using sparsely sampled GPS data. Additionally, a method for trajectory reconstruction was proposed. The method applies decision making based on the temporal aspect of the provided GPS data.

In this thesis work suitable platform architecture was designed. Firstly, appropriate source data was selected and processed. Furthermore, the tools that meet the objectives of this thesis work the best and fit with the used data, were found and integrated. In addition, the algorithm for trajectory reconstruction was added. Real GPS data was used to test the proposed method. As a result the method was able to find the real trajectory travelled by the user more accurately than it would have been when only the shortest path would have been found and the temporal aspect would not have been considered. All in all, the aim of this thesis was met and a platform that is capable to analyse the data, reconstruct the probable real trajectory and visualise it on an interactive web map was developed.

However, there still remain improvements that could be done. In future work map-matching could be either implemented or some of the libraries enabling its usage could be integrated to this platform. Additionally, instead of using Java to implement the trajectory reconstruction, the algorithm could be written in PL/SQL. This would help to make it less memory and time consuming. Another improvement would be to create a plugin for QGIS that would automate a significant amount of the work needed to be done.

References

- [1] Wikipedia webpage: https://en.wikipedia.org/wiki/Global_Positioning_System (12.05.2016)
- [2] OpenStreetMap Foundation: <https://www.openstreetmap.org/> (12.05.2016)
- [3] Hart, P. E., Nilsson, N. J., Raphael, B. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths" IEEE Transactions of Systems Science and Cybernetics, Vol. SSC-4, No. 2 July 1968. (1968)
- [4] Yildirimoglu M., Ozbay K. "Comparison of Two Novel Travel Time Estimation Techniques Based on Probe Vehicle Data: Kriging versus Non-linear Programming Based Approaches" Transportation Research Board 91th Annual Meeting, (2012)
- [5] Shin, D., Aliaga, D., Tunçer, B., Müller Arisona, S., Kim, S., Zünd, D., Schmitt, G. "Urban sensing: Using smartphones for transportation mode classification" Computers, Environment and Urban Systems, Volume 53, September 2015, Pages 76–86, (2015)
- [6] Vavpetič, A. "An implementation of the Pathfinder algorithm for sparse networks and its application on text networks" Vladimir Batagelj Faculty of Mathematics and Physics, Department of Mathematics, Vid Podpečan Jožef Stefan Institute Ljubljana
- [7] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs" Numerische Mathematik 1: 269–271. (1959)
- [8] Wang, H., Ni, D. "Trajectory Reconstruction for Travel Time Estimation" Journal of Intelligent Transportation Systems: Technology, Planning, and Operations Volume 12, Issue 3, (2008)
- [9] Li, J., Xie, L.-H., Lai, X.-J., Research Journal of Applied Sciences, Engineering and Technology 6(12): 2153-2158, (2013)
- [10] Google Maps / Google Earth: <https://www.google.com/maps/> (12.05.2016)
- [11] OpenStreetMap Foundation: <http://learnosm.org/en/osm-data/data-overview/> (12.05.2016)
- [12] Nguyen, T. N. webpage: https://nguyentuannam.files.wordpress.com/2015/08/code_file_osm.png?w=869 (12.05.2016)
- [13] OpenStreetMap Foundation: http://wiki.openstreetmap.org/wiki/Downloading_data (12.05.2016)
- [14] JOSM webpage: <https://josm.openstreetmap.de/> (12.05.2016)
- [15] Overpass API: http://wiki.openstreetmap.org/wiki/Overpass_API (12.05.2016)
- [16] PgRouting webpage: <http://pgrouting.org/> (12.05.2016)
- [17] PostGIS webpage: <http://postgis.net/> (12.05.2016)
- [18] PostgreSQL webpage: <http://www.postgresql.org/about/> (12.05.2016)
- [19] PgRouting Workshop Manual: <http://workshop.pgrouting.org> (12.05.2016)
- [20] PgAdmin webpage: <http://www.pgadmin.org/> (12.05.2016)
- [21] PgRouting webpage: <http://pgrouting.org/docs/tools/osm2pgrouting.html> (12.05.2016)
- [22] QGIS webpage: <http://qgis.org/en/site/> (12.05.2016)

- [23] Leaflet webpage: <http://leafletjs.com/> (12.05.2016)
- [24] QGIS Python Plugins Repository: <https://plugins.qgis.org/plugins/qgis2leaf/> (12.05.2016)
- [25] OpenLayers webpage: <http://openlayers.org/> (12.05.2016)
- [26] W3C from Reuters Limited document: <https://www.w3.org/TR/NOTE-datetime> (12.05.2016)

Appendix

I. License

Non-exclusive licence to reproduce thesis and make thesis public

I, Careelika Liisi Kuik,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
 - 1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Platform for Trajectory Reconstruction Using Sparsely Sampled GPS Data,

supervised by Amnir Hadachi, PhD

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 5/16/16