UNIVERSITY OF TARTU
FACULTY OF SCIENCE AND TECHNOLOGY
Institute of Computer Science
Software Engineering Curriculum

Margus Haavala

Mobility Data Mining for Rural and Urban Map-Matching

Master's Thesis (30 ECTS)

Supervisor(s): Amnir Hadachi, PhD

Tartu 2016

Mobility Data Mining for Rural and Urban Map-Matching

Abstract:

The functionality of gathering spatio-temporal data has seen increasing usage in various applications and devices. The Global Positioning System (GPS) is a satellite navigation system which is mostly used for gathering location information. Map-matching is the procedure of matching trajectories from a sequence of raw GPS data points to the appropriate road networks. GPS data errors are one of the biggest problems and correcting them is a big challenge. The main goal of this thesis work is to build a data pipeline and visualization framework for turning raw GPS data to trajectories and correcting erroneous GPS points by new map-matching approach. For achieving the goal a new approach for trajectory pattern mining is introduced.

Keywords:

Map-matching, GPS trajectories, GPS, visualization, space-time cube, temporal spatial data, movement, routing

CERCS: P170

Liikumisandmete andmekaeve meetodite kasutamine kaardipunktide vastavusse seadmiseks

Lühikokkuvõte:

Ajaliste ja ruumiliste andmete kogumine on hoogustunud erinevates rakendustes ja seadmetes. Globaalne positsioneerimise süsteem (GPS) on kõige populaarsem viis asukohateabe saamiseks. Kaardipunktide vastavusse seadmine on kontseptsioon, mis püüab GPS andmeid trajektooris viia vastavusse reaalse teedevõrguga. GPS andmete suurim probleem tuleneb andmete mõõtmis- ja kogumise vigadest ja nende parandamine on suur väljakutse. Käesoleva lõputöö eesmärk on arendada andmete töötlusvoo ja visualiseerimise raamistik, et muuta GPS punktid loogilisteks trajektoorideks ja parandada vigaste GPS punktide asukohad. Selle eesmärgi saavutamiseks tutvustatakse uut lähenemist trajektooride mustrite leidmiseks.

Võtmesõnad:

GPS trajektoorid, GPS, visualiseerimine, ajaline-ruumiline andmestik, liikumine, marsruutimine

CERCS: P170

LIST OF FIGURES

Table of Contents

1 Introduction

1.1 General View and Motivation

The functionality of gathering spatiotemporal data has seen increasing usage in various applications and devices. The Global Positioning System (GPS) is a satellite navigation system, which is mostly used for gathering location information. Different Intelligent Transportation Systems require navigation and location information and it is important that gathered trajectory data could be matched to spatial road networks. Map-matching is the procedure of matching trajectories from a sequence of raw GPS data points to the appropriate road network segments. GPS data errors are one of the biggest problems and correcting them is a big challenge. There are multiple factors influencing GPS data accuracy: quality of GPS device, the position of satellites and the surrounding landscape. A good map-matching process needs to take into account the logic of the trajectory, high variance of road dynamics and the behaviour of the drivers. Analysing and mining the data from spatiotemporal aspect will be very helpful in the achievement of solving this problem. The current state of the art regarding map-matching problem has been moving towards multi-track map-matching, which considers multiple trajectories instead of only one. This improves the matching accuracy when the GPS sampling rate is low or sampling error is high. Applying extra data feeds to multiple trajectories has shown promising results.

1.2 Research Questions and Objectives

The main goal of this thesis work is to build a data pipeline and visualization framework for turning raw GPS data to trajectories and correcting erroneous GPS points by new map-matching approach.

Research questions are the following:

1. How to solve trajectory pattern extraction?
2. How similar trajectories could help with map-matching process?
3. How to validate trajectory correctness?

Based on the research questions the objectives for the thesis are:

- Designing a workflow for gathering and cleaning geospatial data for creating trajectories for map-matching purposes
- Extracting similarity patterns from trajectory data
- Applying extracted similar routes for map-matching process
- Using possible routing paths for validating trajectory paths
- Implementing a visual framework to aid implementation and understanding

1.3 Scope

The need for matching sensor-gathered location data to correct map positions has existed for a long time for different Intelligent Transportation System applications. But in recent years the rise of location-tracking capable devices and technology-based business models like ridesharing has created even more demand for precise and scalable solutions.

1.4 Contributions

As the outcome of the thesis a data pipeline for turning raw GPS data to trajectories and visualization framework will be created. A new approach for trajectory pattern mining is introduced based on GPS errors for map-matching purposes.

1.5 Road Map

In section 2 the state of the art in map-matching techniques and approaches will be introduced. An overview of related research and future directions is given. Section 3 presents applied methodologies and technical implementation details. Section 4 gives an overview of results and analysis outcomes. Conclusions and future work ideas are discussed in section 5.

2 State of the art

The following chapter will introduce the domain and theoretical contributions of related research.

2.1 Introduction

The increasing popularity of different sensors gathering large spatiotemporal datasets (e.g. GPS tracks from navigation devices or call detail records from mobile phones) has made possible to extract more useful and diverse knowledge from it, which can be used in Location Based Services (LBS). Some of the use cases of LBS are: fleet management, traffic monitoring, vehicle tracking, navigation, logistics and mobile commerce [1]. The main positioning technologies are Global Positioning System (GPS) and network-based (cellular or Wi-Fi networks). In this thesis the focus will be on GPS measurements.

2.1.1 GPS

Global Positioning System (GPS[1]) is capable of determining position on Earth. The system has at least 24 operational satellites orbiting the Earth at 20000 km. GPS requires 4 satellites and distances between them and target position for determining latitude and longitude. The moving-point trajectories are inherently imprecise, which is caused by the used measurement process and by the sampling approach. Measurements errors are caused by inaccurate GPS measurements. Sampling error is the uncertainty of the point movement between the times samples are taken. Current GPS technology allows determining position of a moving object with an accuracy of 2 meters. Determining the position every 2.5 seconds has the worst-case error of 50 meters as measurement error. In practice these could be even 200 meters [2]. The distance between two GPS points is mostly bigger than the true distance because of the GPS measurement errors [3].

GPS devices are capable of collecting data about current location, time, speed, heading and other attributes. Due to GPS errors the raw trajectories do not align correctly with real-world road networks.

2.1.2 Map-matching

Map matching is the procedure of matching trajectories from a sequence of raw GPS data points to the appropriate road network on the map. Figure 1 illustrates the map-matching task of finding the correct road segment from multiple options. The main map-matching approaches are:

- **Incremental**: trajectories will be constructed on arrival of new data. Can generate vehicle paths on the fly. Used in navigation, where sampling rate is high.
- **Global**: finds the globally optimal path after reading in complete trajectories. Focus on accuracy and robustness.

[1] http://www.gps.gov/

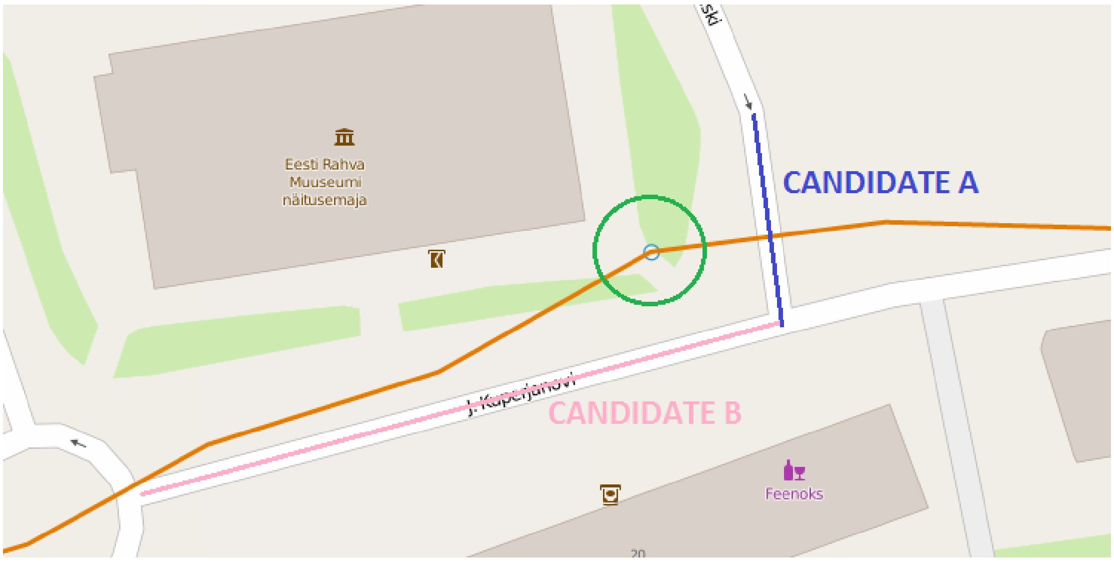


Figure 1. Map-matching task

2.2 Related Work

2.2.1 Map-matching

In the following section detailed overview of map-matching algorithms will be given.

Map-matching algorithms can be categorized into groups [4,5]:

- **Geometric map-matching** - only the shape of the road segments are considered and not the way these are connected. Like point-to-point (matching closest road network node to given GPS point), point-to-curve (matching closest road network segment to given GPS point), curve-to-curve.
- **Topological map-matching** - map-matching uses the connectivity, geometry and contiguity of the links.
- **Probabilistic map-matching** - such road segments are selected as candidates that intersect with the confidence region.
- **Advanced map-matching** - usually combines both topological and probabilistic approaches. Concepts like probabilistic theory, Kalman Filter, fuzzy logic

Purely geometric approaches are sensitive to measurement noise and sampling rate. A map-matching algorithm based on Hidden Markov Model (HMM) was developed focusing on improving accuracy of noisy and sparse GPS data. HMM models the connectivity of the road network and considers multiple path hypotheses simultaneously [6]. A multi-track map-matching algorithm called UrbMatch [7] was implemented for improving accuracy of map-matching by considering also additional urban area properties like high volume of road segments, diverse functionality of roads. The entire road network will be divided into multiple smaller sub-networks to speed up the map matching through concurrent execution [7]. Common trajectories have usually many regularity patterns. The usual trajectory consists mostly only small number of all possible connected k-segments on the road network. Multi-track map-matching algorithm was developed to recover regularity patterns. The process starts with initialization of all road segments in the map. Each trajectory sample point will be assigned to a segment (the optimization is determined by regularity, proximity and consistency between segments). And those selected segments will be stitched to a projected path. The problem can be divided to sub-problems, which allows analyzing a lot of data thanks to parallel work [8].

Different sampling rates require alternative approaches. For low sampling (one GPS point per 2 minutes or more) GPS data a global map-matching algorithm called ST-Matching [9] was implemented. It uses topological information about the road network and temporal/speed constraints of road segments. A candidate will be constructed based on spatio-temporal analysis and best path identified. It was based on the observations [9]:

- Correct paths are more likely to be direct than roundabout
- Correct paths are more likely to be inside the speed constraints of the road.

2.2.2 Trajectory Data Mining

The following chapter will give an overview of data mining techniques when dealing with trajectory data.
Spatial data mining tries to discover new and useful patterns from spatial datasets. Spatial objects have implicit relationships (e.g. intersections, overlapping) between objects [10].
In a systematic approach (figure 2) to describe the field of trajectory data mining different steps were defined [11]:

- Trajectory data requiring
- Trajectory data preprocessing: noise filtering, segmentation, compression and map-matching
- Trajectory indexing and retrieval
- Uncertainty in a trajectory
- Trajectory pattern mining
- Trajectory classification
- Anomalies detection

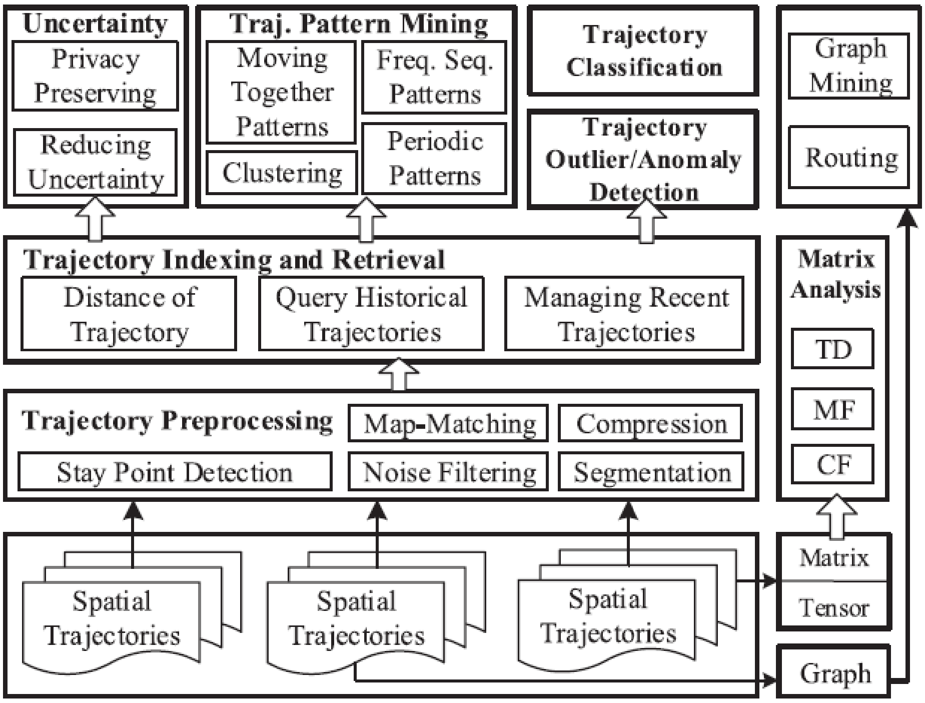


Figure 2. Paradigm of trajectory data mining [11]

Inside the system M-Atlas [12], which is a mobility data mining tool for storing, querying and mining trajectory data, basic mobility patterns were described [12]:

- **T-Flock** - representing a spatiotemporal coincidence of a group of moving points. It represents the common behavior of vehicles using the same routes during same time intervals
- **T-Cluster** - representing a group of similar trajectories. Multiple similarity functions are available.
- **T-Pattern** - representing trajectory segments that visit same regions with same sequence and with similar transition times.

2.2.3 Visualization

Using right visual analytics tools helps humans in understanding mobility patterns. Purely visual analytics methods are not enough for complex and large data. The combination of interactive visualisation tools and data manipulation tasks are considered essential [13]. When trying to visualize both spatial and temporal aspects of the trajectories, then space-time cube is often used as representation view (figure 3). It represents geospatial attributes and time in a view of three dimensions. It is also considered to be important to support additional filters (e.g. time filter) to apply restictions on dataset when the amount of data is making it unreadable [14].

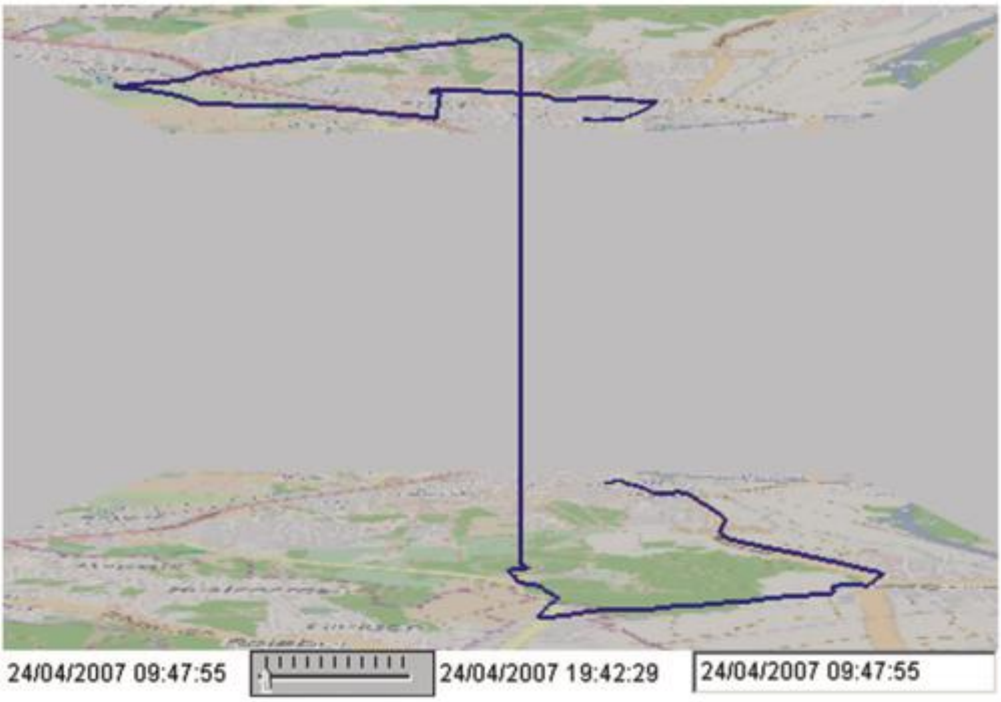


Figure 3. Space-time cube for representing the trajectory of a car [14]

2.3 Conclusion

Spatiotemporal datasets are gathered by more sensors than ever and used in different Location Based Services. GPS is the most popular solution for gathering location information. GPS errors are caused by inaccurate measurements, urban canyon effect or sampling errors. Map-matching is the procedure of matching trajectories from a sequence of raw GPS data points to the appropriate road network on the map. The main categories of map-matching algorithms are: geometric, topological, probabilistic and advanced. From trajectory data different knowledge can be extracted by mining the data. Trajectory data mining consists of multiple steps and different visualization options for combining temporal and spatial data are used for better understanding of the underlying data.

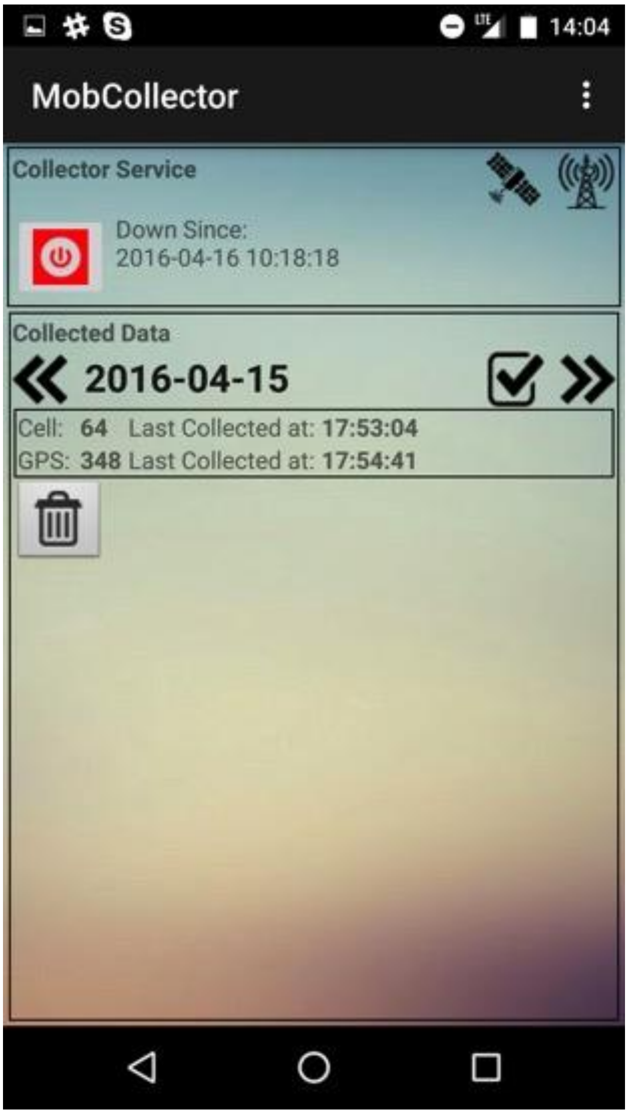
3 Methodology

The following chapter will describe applied methodology and technical implementation details.

3.1 Introduction

3.1.1 GPS Data

The input dataset consists of GPS points collected from March 2015 to April 2016 by 13 different devices. The total number of collected points is 601829. For data collection a tool called *MobCollector* (figure 4) is used, which was developed by the Distributed System Group[2] from University of Tartu. It is an Android[3] application, which is capable of keeping track of GPS positions with configurable sampling rate.



Down Since:
2016-04-16 10:18:18

2016-04-15

Cell: 64 Last Collected at: 17:53:04
GPS: 348 Last Collected at: 17:54:41

Figure 4. MobCollector application

[2] http://ds.cs.ut.ee/
[3] https://www.android.com/

The input CSV has the following structure:

```
imei;t_ascii;ev_type;lat;lon;speed;acc;heading
1;2015-03-21 14:50:54.0;5;58.37712602;26.72358884;1.41421353817;8.0;266.8999938
```

A subset of imported attributes is used. The attributes with significance in the context of the methodology are the following:

- **imei** - stands for International Mobile Station Equipment Identity. The value is anonymized in the system.
- **t_ascii** - date and time value in local time zone of the device.
- **lat** - latitude is the part of geographical coordinate representing north-south position of the Earth. Is in EPSG:4326 projection[4].
- **lon** - longitude is the part of geographical coordinate representing east-west position of the Earth. Is in EPSG:4326 projection.
- **speed** - speed value returned by GPS sensor. This attribute is not used in further analyze phases, because only 15% of GPS points have it defined.

The collected GPS points have various sampling rates (figure 5) and 75% of measurements have sampling rate less than 6 seconds.

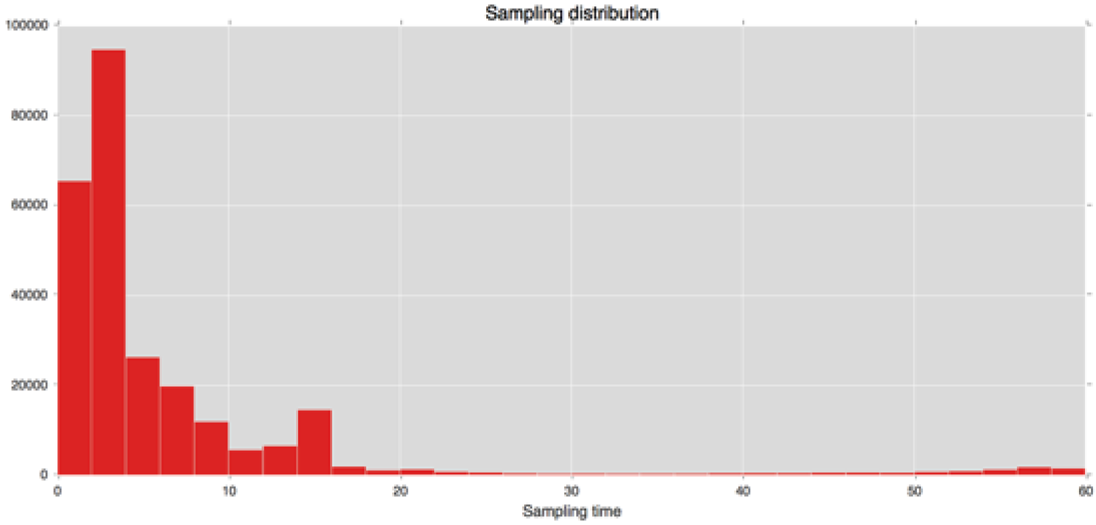


Figure 5. GPS points sampling time distribution

3.1.2 Geographic Information System (GIS) Data

As the source of geospatial data OpenStreetMap[5] data is exported. OpenStreetMap is probably the most popular example from the field of Volunteered Geographic Information (VGI). It is user-generated spatial data. Although there have not been any studies yet about the data quality of OpenStreetMap in Estonia, but in other areas the studies have concluded that the data is fairly accurate [15]. For the proposed methodology the road network (figure 6) is exported from OpenStreetMap. The tool called osm2pqsql[6] is used for the data import. The main elements of OpenStreetMap data are:

- **Nodes** - represents a specific point on the earth's surface defined by its id, latitude and longitude values.

[4] http://spatialreference.org/ref/epsg/wgs-84/
[5] https://www.openstreetmap.org/
[6] http://wiki.openstreetmap.org/wiki/Osm2pgsql

- **Ways** - represent linear features (e.g. roads) and area boundaries. It’s an ordered list of nodes. Ways must share a node if they intersect at the same altitude.
- **Relations** - defines the relationship between two or more data elements. E.g. route relation, which connects roads to highway.

All the elements can have tags, which describe the element. A tag consists of key-value pair. The key describes the feature type (e.g. *highway*, *maxpeed*, *name*) and the value specifies the type.

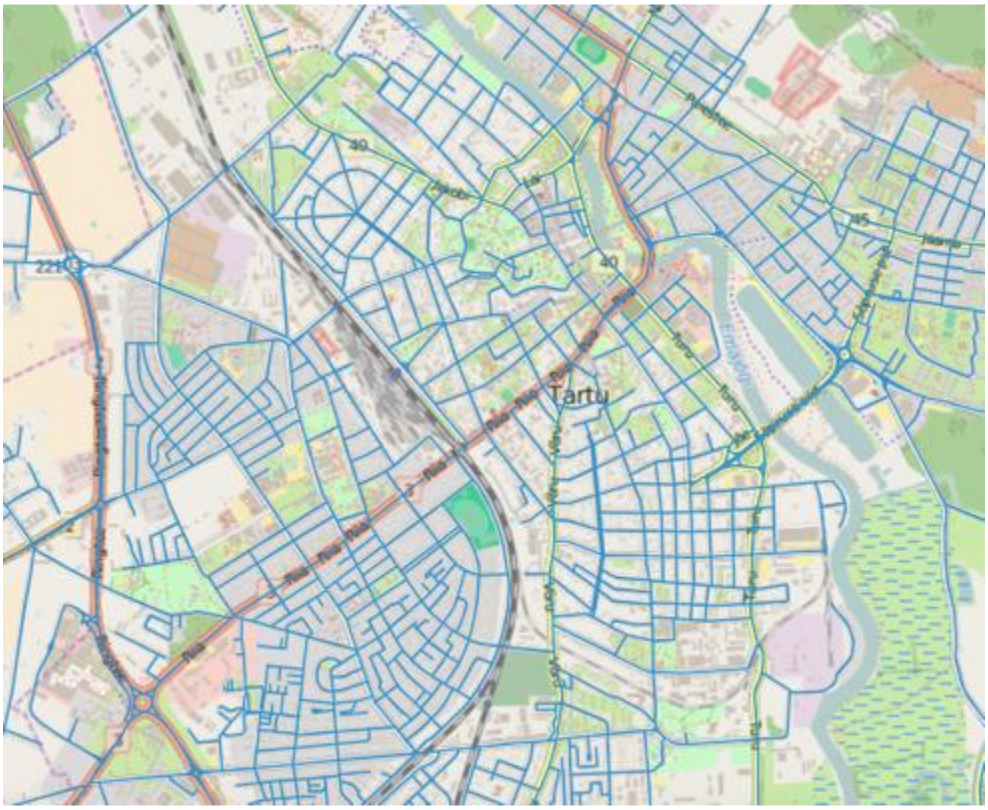


Figure 6. Road network from OpenStreetMap

3.2 Problem Statement

The process of mapping raw GPS points to road segments involves multiple steps. After GPS data import, points must be validated and outliers filtered out. Individual points must be grouped by different characteristics to trajectories and classified by movement type. Map-matching involves two steps: finding the most probabilistic road segment and fixing the point location by moving it to the selected road segment. The applied methodology performs the preprocessing steps from raw GPS data to vehicle trajectories, locates possible matching road segments and fixes the GPS measurement errors by re-locating the erroneous points.

3.3 System Design and Architecture

For data import and manipulation mostly Ruby[7] (with library pg[8]), Python[9] (with libraries pandas[10], matplotlib[11], psycopg[12]) and Bash[13] are used. PostgreSQL[14] is used as the data-

[7] https://www.ruby-lang.org/en/

[8] https://rubygems.org/gems/pg/

base engine. PostGIS[15] is the extension for PostgreSQL that provides support for spatial data types and spatial operations. All the spatial data in the system is converted to Geometry[16] data type and spatial indexes added, which improves significantly the performance of working with spatial data. Geometry types are stored in "EPSG:4326" projection. PgRouting[17] is the extension for PostgreSQL that provides routing options directly in SQL. Visualization application is built on top of Node.js[18] (with libraries express[19], pg-promise[20]) for the backend and in the frontend Angular.js[21], OpenLayers[22], d3.js[23] and Bootstrap[24] are used.

3.4 Adopted Methodology

The methodology consists of three main building blocks (figure 7). In the preprocessing stage raw GPS data is read from CSV files and stored in spatial database. Pre-filtering will find outliers; in the segmentation phase GPS points are grouped to trajectories and later classified by movement type. During trajectory data mining phase a grid system based on GPS errors is constructed, which is used to find per trajectory the intersecting grid cells. Based on the intersections similar trajectories will be found. For map-matching it's important to determine the matching road segment and later fixing the location of GPS points with measurement errors. For finding the best candidate road segment for each GPS point matching grid cells are used and the order of road segments is checked for correctness. For validation the confidence of correct matching is found based on trajectory similarity patterns and possible routes between origin-destination pairs.

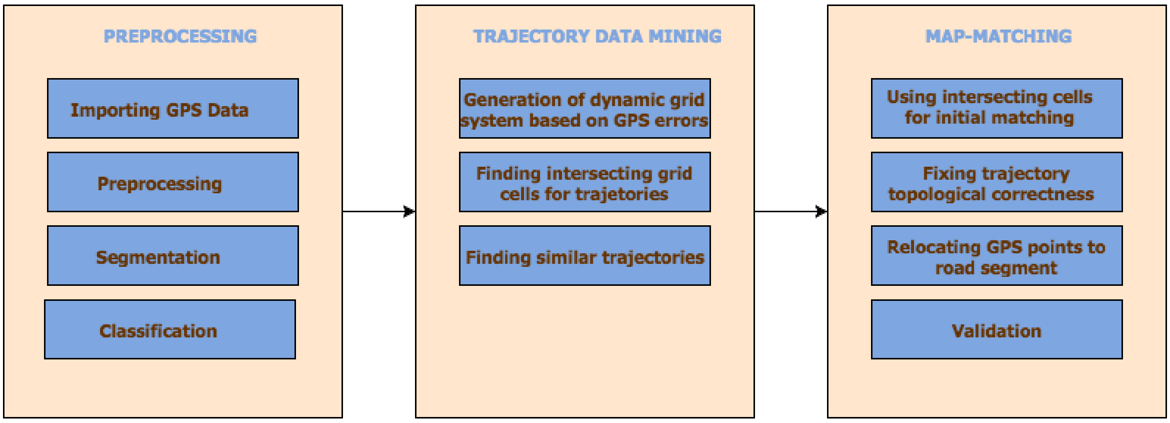

PREPROCESSING
Importing GPS Data
Preprocessing
Segmentation
Classification
TRAJECTORY DATA MINING
Generation of dynamic grid system based on GPS errors
Finding intersecting grid cells for trajetories
Finding similar trajectories
MAP-MATCHING
Using intersecting cells for initial matching
Fixing trajectory topological correctness
Relocating GPS points to road segment
Validation

Figure 7. Methodology overview

[9] https://www.python.org/
[10] http://pandas.pydata.org/
[11] http://matplotlib.org/
[12] http://initd.org/psycopg/
[13] https://www.gnu.org/software/bash/
[14] http://www.postgresql.org/
[15] http://postgis.net/
[16] http://postgis.net/docs/geometry.html
[17] http://pgrouting.org/
[18] https://nodejs.org/en/
[19] http://expressjs.com/
[20] https://github.com/vitaly-t/pg-promise
[21] https://angularjs.org/
[22] http://openlayers.org/
[23] https://d3js.org/
[24] http://getbootstrap.com/

3.4.1 Data Pre-processing

Geospatial filtering will be performed to flag all points that are outside the region of interest. The selected bounding box is defined as

$$ST\_BBOX = \{ (lat_{min}, lon_{min}), (lat_{min}, lon_{max}), (lat_{max}, lon_{max}), (lat_{max}, lon_{min}) \}.$$

All the GPS points will be enriched with additional attributes: the distance to next position in meters and the time to next position in seconds. And all the points are flagged with the information if they are on top of the building and on top of the road network.

3.4.2 Trajectory Segmentation

Trajectory segmentation is the process of splitting a trajectory into smaller parts called segments. Points inside each segment share some movement characteristic (e.g. speed, spatial closeness)[16]. These segments will be called trips.

Trajectory segmentation is implemented in multiple steps. The first step is to split the global trajectories per unique user by time interval defined as *ST_TIME_THRESHOLD_FOR_CUT*. If the time between two consecutive GPS points is longer than the threshold value, those points will be classified as trip end and start points accordingly.

Next steps are splitting the existing trips to multiple trips by identifying a change in the spatiotemporal aspect of the trajectory. Stop points cause the most common spatiotemporal change. For that the Spatio-Temporal Kernel Window (STKW) statistics value is calculated for each point [17]. All points per trip are ordered by timestamp and for each point in both directions the number of points were counted having distance smaller than *ST_SEGMENTATION_POINT_THRESHOLD* (figure 8). When the differences of the sum to both directions differ significantly, it indicates that the point is a stop point and the trips are split.

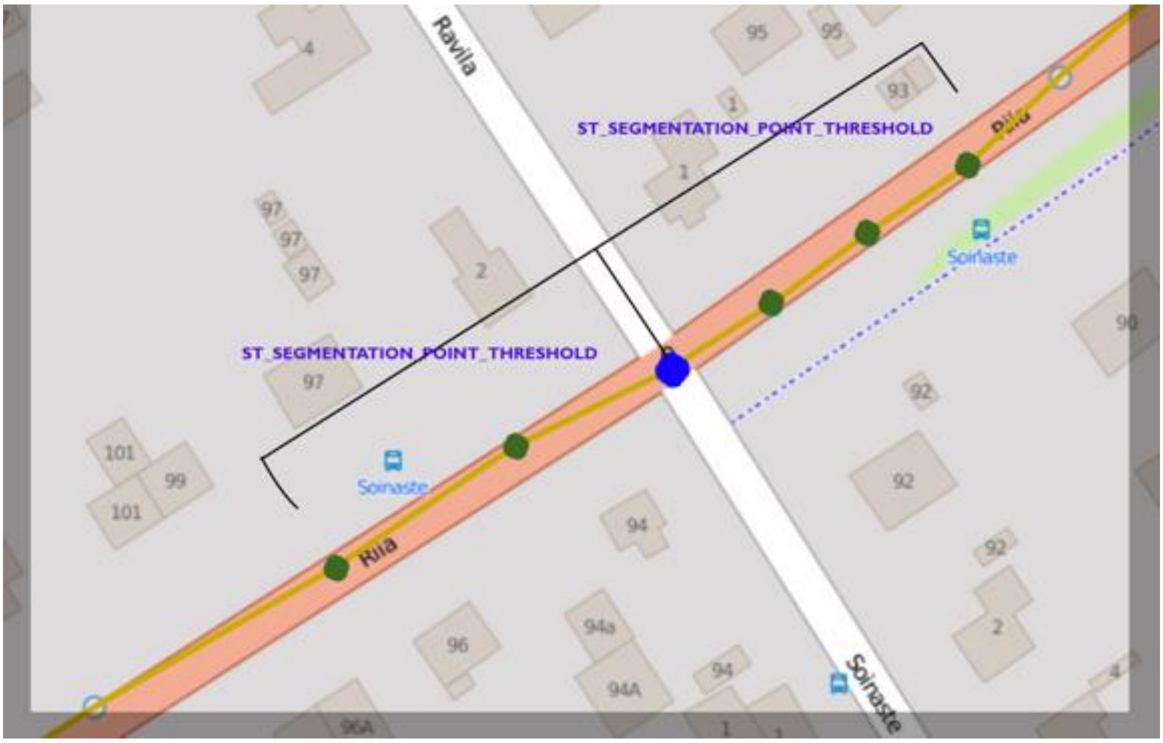

ST_SEGMENTATION_POINT_THRESHOLD
ST_SEGMENTATION_POINT_THRESHOLD
Ravila
Riia
Soinaste
Soinaste

Figure 8. Spatio-Temporal Kernel Window (STKW) principle [17]

3.4.3 Trajectory Classification

The trajectory movement classes in our context are vehicle and pedestrian movements. For each trajectory median speed is calculated as it handles outliers better compared to average speed [17]. If the median speed is higher than the value *ST_DRIVING_SPEED_THRESHOLD*, trip will be classified as '*driving*' type. Otherwise the trip will be classified as '*walking'*.

3.4.4 Dynamic Road Segments Grid System

In geospatial analysis often grid-based layers are used for indexing and for clustering. The most common grid cell creation patterns are rectangles and hexagonal figures. In the context of using cells for road segments the problem with rectangles and hexagonal is that they are noisy. Depending on the situation it can be that some cells have many intersecting roads and in some cases one road segment is in many cells.

We are introducing an alternative grid creation approach, which takes into account possible GPS errors and builds cells for every road network segment. The process iterates through all road segments and finds all GPS points around the segment inside the defined *ST_SEGMENT_BUFFER_THRESHOLD* threshold value (figure 9). Only those GPS points are taken into account that are part of some trajectory with type '*driving*'.

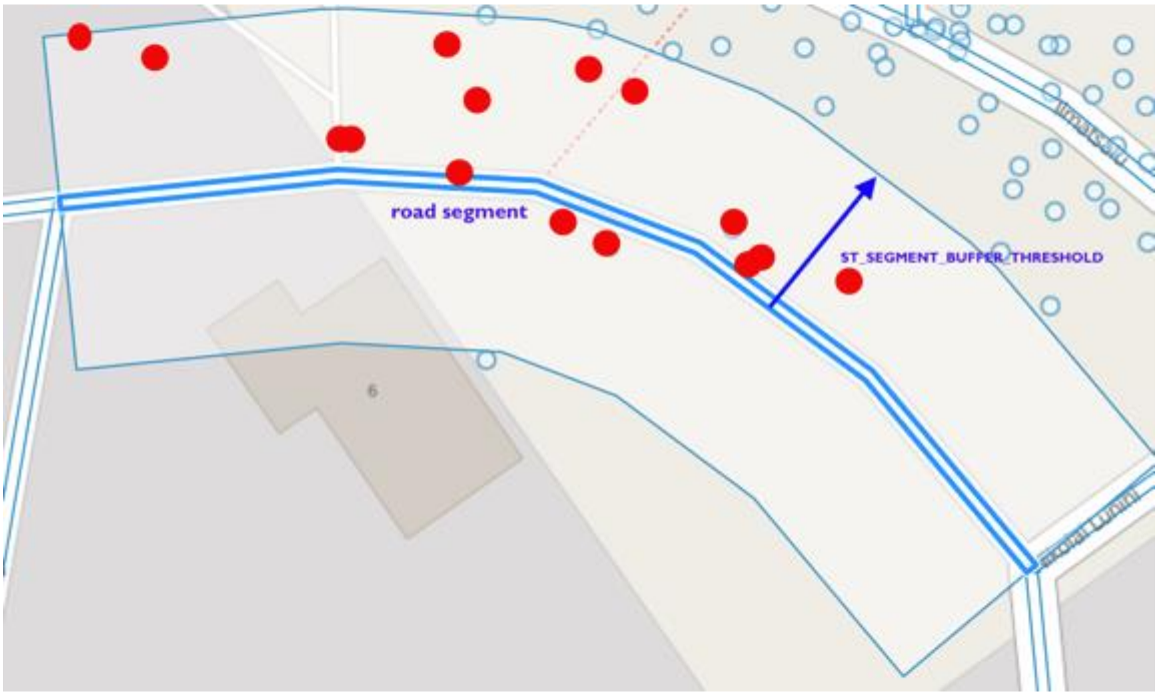


Figure 9. GPS points selection principle per road segment

The length of the cell will be the length of the road network segment. The width of the cell will be calculated with the following formula (figure 10):

$ST\_GRID\_WIDTH = max\ (distance\ (POINTS\_IN\_BUFFER)) + \varepsilon$,

where

$\varepsilon = avg\ (distance\ (POINTS\_IN\_BUFFER))$,

$POINTS\_IN\_BUFFER = \{(lat_1,lon_1),\ \ldots,\ (lat_n,lon_n)\}$,

where (lat_i, lon_i) was inside *ST_SEGMENT_BUFFER_THRESHOLD*.

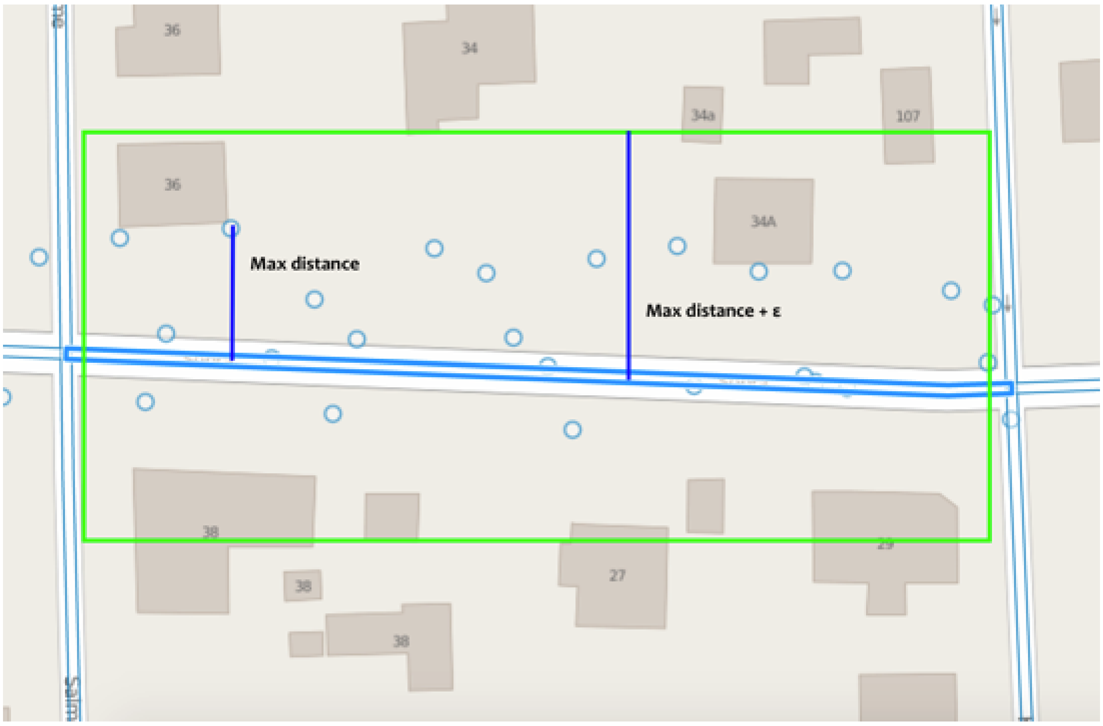


Figure 10. Dynamic grid cell generation principle

As the end result after iterating through all the segments and calculating grid cell sizes based on GPS errors, segments will have intersecting cells, which do intersect with each other (figure 11).

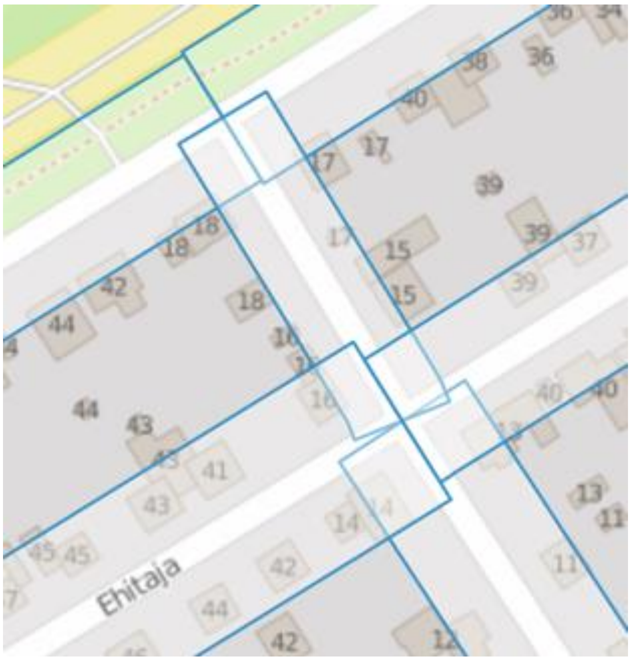


Figure 11. Step 1 of grid system creation

For the intersections of road segments round cell will be generated (figure 12). Overlapping cells with be cut off (figure 13). Additionally special cell for all the roundabouts are added (figure 14) and all overlapping cells cut off (figure 15).

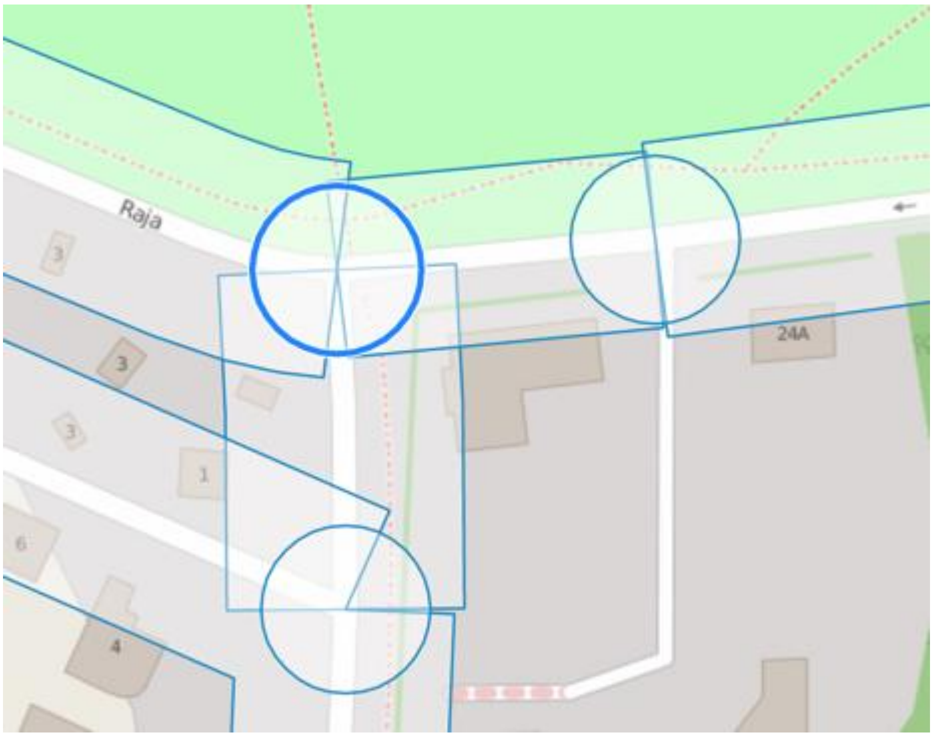


Figure 12. Step 2 of grid system creation: adding node buffers

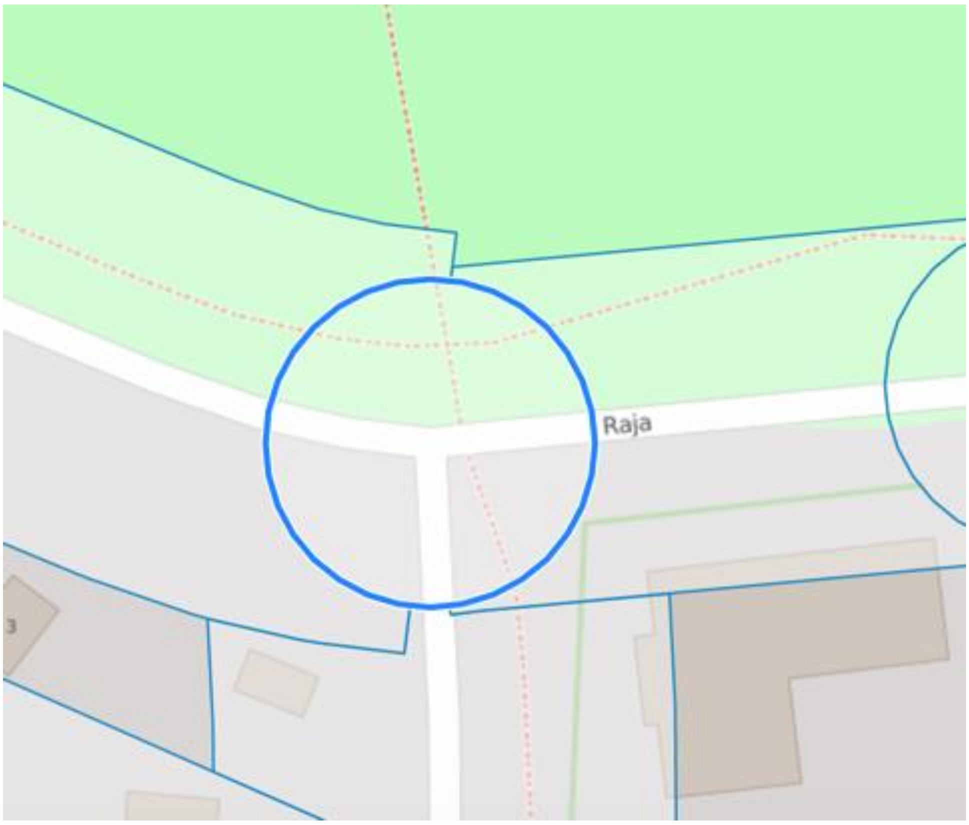


Figure 13. Step 3 of grid system creation: removing intersections

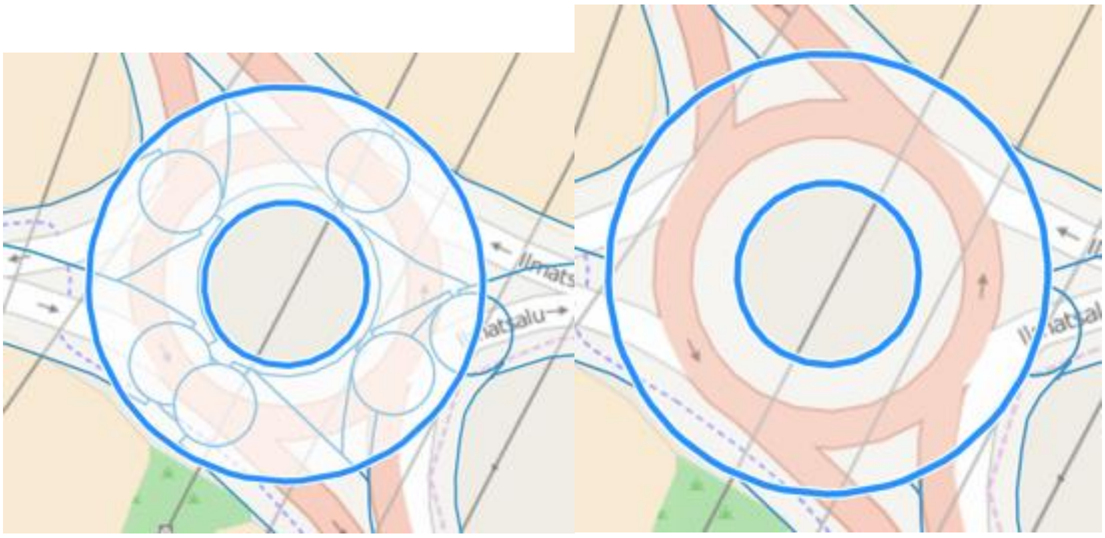


Figure 14. Step 4 of grid system creation: add cells for roundabout

3.4.5 Trajectory Similarities

The generated dynamic road segment grid is used for trajectory pattern extraction. For every trajectory point and trajectory line segment intersecting grid cells will be found. There can be only up to 1 cell per point, but up to multiple cells per line segment. Special attention will be given for trajectory start and end points. In case start and end points will not have corresponding grid cell (figure 15), the first and last intersecting grid cell will be set correspondingly as start and end grid cells.

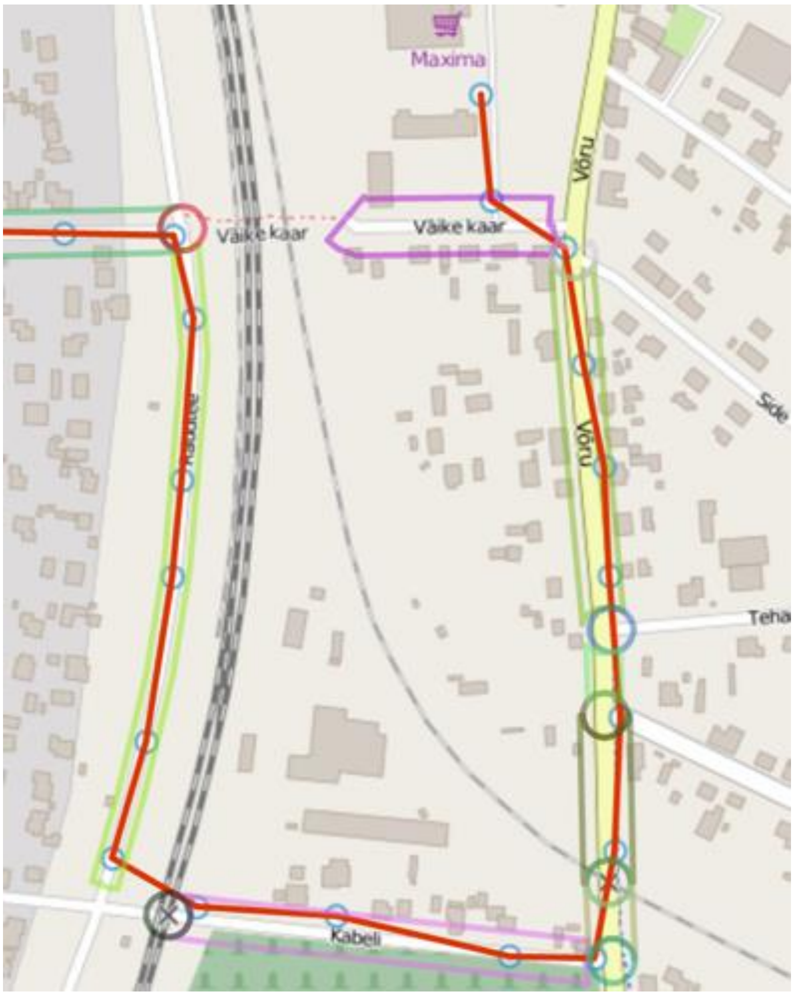


Figure 15. Trajectory start/end point is outside grid cell

After the intersecting cells have been found for all trajectories, similarity weights between the trajectories are calculated. The value is in the range 0 - 1 with the following semantics:

- **0** - trajectories have no common cells
- **1** - trajectories have all the same cells
- **0 < x < 1** - trajectories have some cells the same

3.4.6 Routing

pgRouting extends Postgres with routing functionalities. It supports multiple routing algorithms (e.g. Shortest Path Dijkstra, Shortest Path A*). In this thesis K-Shortest Path algorithm was applied. K-Shortest Path algorithm finds *k* shortest paths between two nodes [18]. We have defined a system-wide parameter *ST_K_SHORTEST_PATH* and in the next map-matching phase as many routes will be constructed.

3.4.7 Map-matching Process

For the map-matching process it is first required to determine the corresponding road segments for every point. After the most likely road segment is determined, fixing the GPS error to the road is performed.

In the previous steps for every GPS point corresponding road segment grid cell was found. There can be up to one cell, as the grid cells do not interleave with each other. Grid cells can have multiple types:

- **Road segment based** – polygon cell representing the segment of the road network. Is directly matched to a road segment (figure 16).
- **Road intersection based** – round cell representing the node of the road network. There can be multiple road segments inside the cell (figure 16).
- **Roundabouts based** – round cell with a hole representing roundabouts. There can be multiple road segments inside the cell.

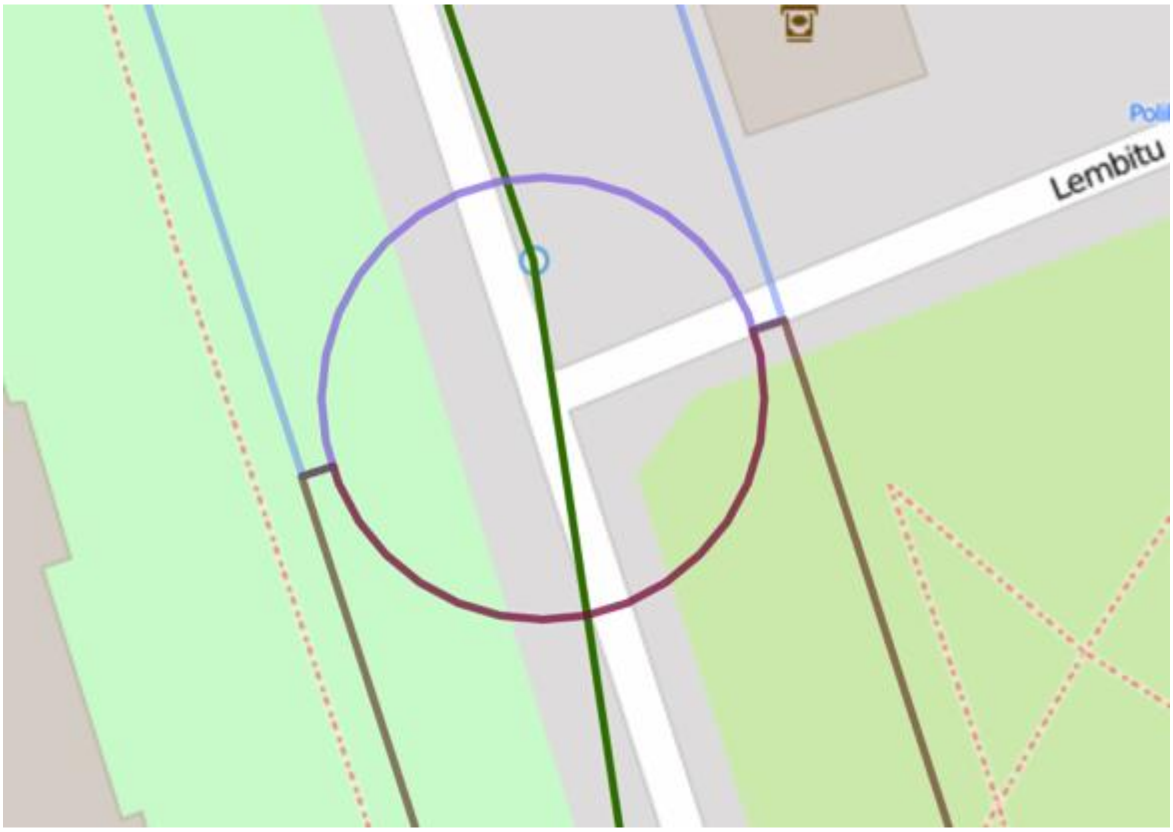


Figure 16. Different road segment cell types

Finding the corresponding road segment cell is an iterative algorithm, which starts from the first point of the trajectory. For each point the following rules are applied based on the grid cell type:

- GPS point is inside the intersection grid cell. The road segment is chosen between the two segments connected to node. Those segments are selected based on other points in the trajectory before and after the current point. Closest node is selected (figure 17). If it's a start or an end point, only one candidate segment will be found.
- GPS point is inside the road segment grid cell. In the first phase nothing else is checked, validation will be done later.
- GPS point does not have corresponding cell. If it's a start or an end point, the first point on the trajectory matching a cell will be chosen. For other points adjacent trajectory points will be checked for matching cells and closest one is selected.

- GPS point is inside other cell because of the measurement error (figure 18). The issue will be found and fix in another step.

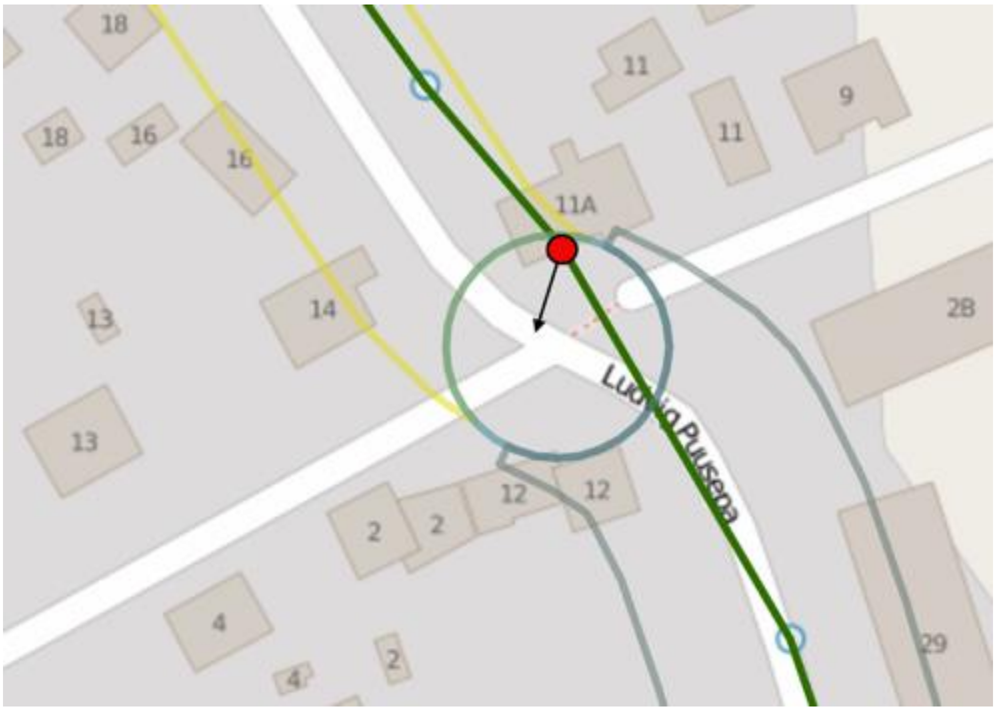


Figure 17. Point inside the node cell

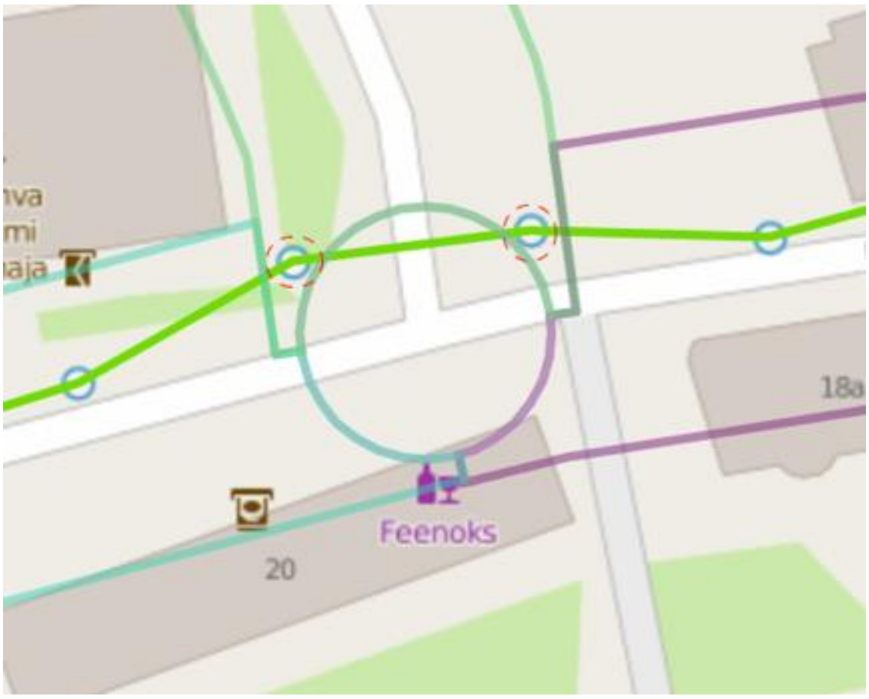


Figure 18. Points on wrong road segment

The second phase is validating the correctness of matched trajectories. For all origin-destination pairs *k* alternative routing paths are generated. Those generated paths will be mapped to the road segment grid and the corresponding list of cells is generated. Every trajectory grid cells will be compared to the generated path cells. In case the order of cell matches, it can be concluded that the GPS points belong to the road segments related to the corresponding grid cells. If no matching path will be found among the generated candidate paths, trajectory will be checked for topological correctness. In most cases the points near the intersection can be wrongly matched (figure 19). Based on the underlying road network rules and restrictions such errors will be checked. In case of invalid order of segments, alternative valid segments are chosen as candidates and based on trajectory and other matched points one valid segment is selected if possible. But there can be issues of wrong road network data, which makes matching not possible.

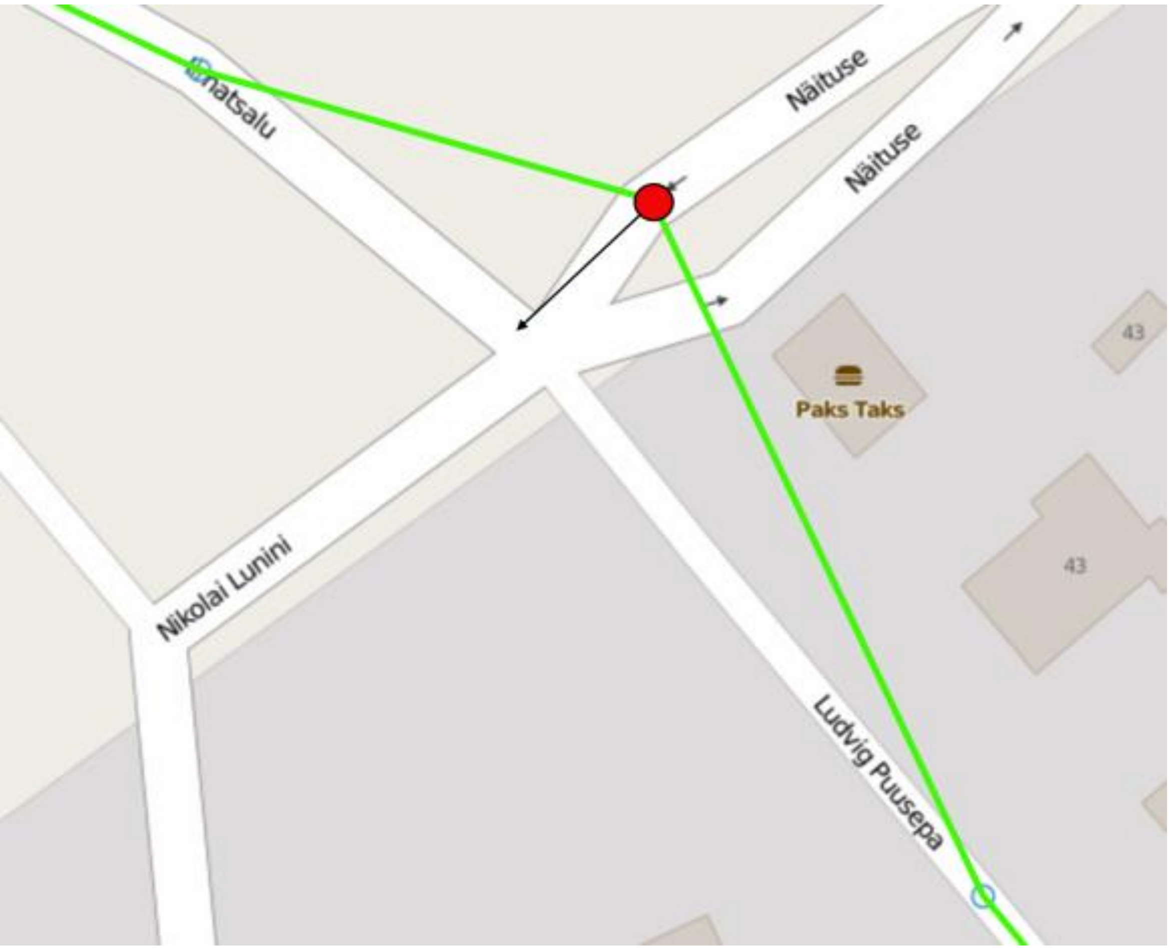


Figure 19. Wrongly matched GPS point

For all matched trajectories a confidence score is calculated, which is based on topological correctness and regularity determined by similar routes. It's a value, which should help decision-makers to find data errors and anomalies. For example if regular trajectories (trajectories that have other similar ones) do not have any possible matching routing paths, it may be cause by missing or wrong topological data. Confidence score is defined in the range 0.0 – 1.0 with the following semantics:

- **1.0** – matched trajectory is topologically correct and there exist other similar trajectories
- **0.0** – no points could be matched at all

- **0.0 < x < 1.0** – trajectory is partly matched correctly; trajectory is matched topologically correctly, but no similar trajectories exist; because of data errors correct trajectory could not be matched correctly to road network

After the best candidate for the road segment is selected the location of GPS points with errors can be corrected. All the points that are flagged with the information about being not on the road network, will get orthogonal projection to closest point on the selected road segment (figure 20).

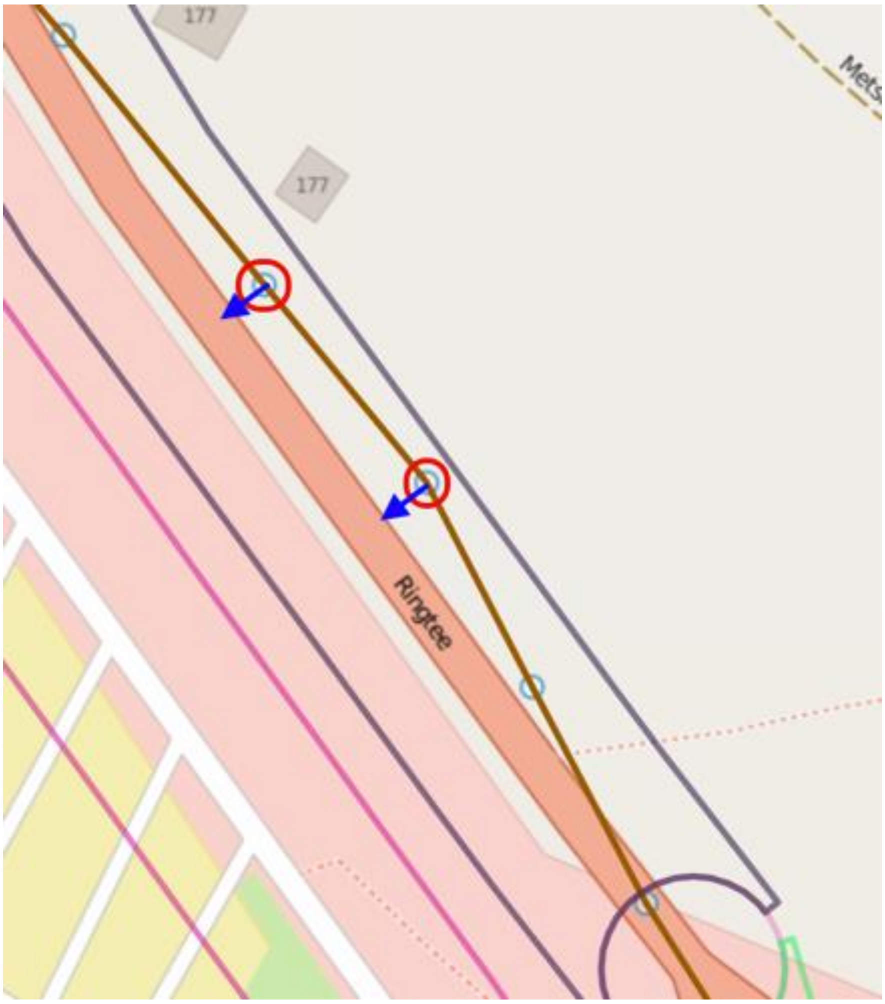


Figure 20. Orthogonal projection to closest point

3.5 Conclusion

In the preprocessing stages different known practices for managing raw GPS data are used. For global trajectories segmentation and classification processes are applied. We presented also our new approach in tilted dynamic grid method for creating GPS error zones based on the GPS error distribution observed from the collected data. The grid is used for finding similar trajectories in a precise and computationally effective way.

For all the trajectories *k* different routes will be generated as reference data. As those generated routes are topologically correct, it is highly probable that vehicle trajectories should follow same paths. The routing results will be used for fixing GPS errors on wrong road segments. For the measurement errors inside the correct road segment orthogonal projection to closest point approach is applied for fixing them.

4 Results and Analysis

In the following chapter the results of the chosen methodology are introduced.

4.1 Introduction

A web application was built to aid the methodology implementation process. The main goals for the functionality were the following: data search and filtering capabilities, visualization of raw GPS points and trajectories in both time and space dimensions, combination of different data layers, possibility to validate routing paths and provide visual comparisons for GPS errors and the fixes.

4.2 Trajectory Pre-processing

Preprocessing and data cleaning were the most time-consuming tasks in the applied data manipulation pipeline.

For the geospatial filtering the bounding box was defined as

$$ST\_BBOX = \{ (lat_{min}, lon_{min}), (lat_{min}, lon_{max}), (lat_{max}, lon_{max}), (lat_{max}, lon_{min}) \},$$

where lat_{min} = 58.3291, lat_{max} = 58.4233, lon_{min} = 26.5759 and lon_{max} = 26.8942. As the result 70% out of the initial 601829 GPS points were inside the target area (figure 21), points outside the defined area were not used. 19% of the points inside the spatial filter were on the road network.

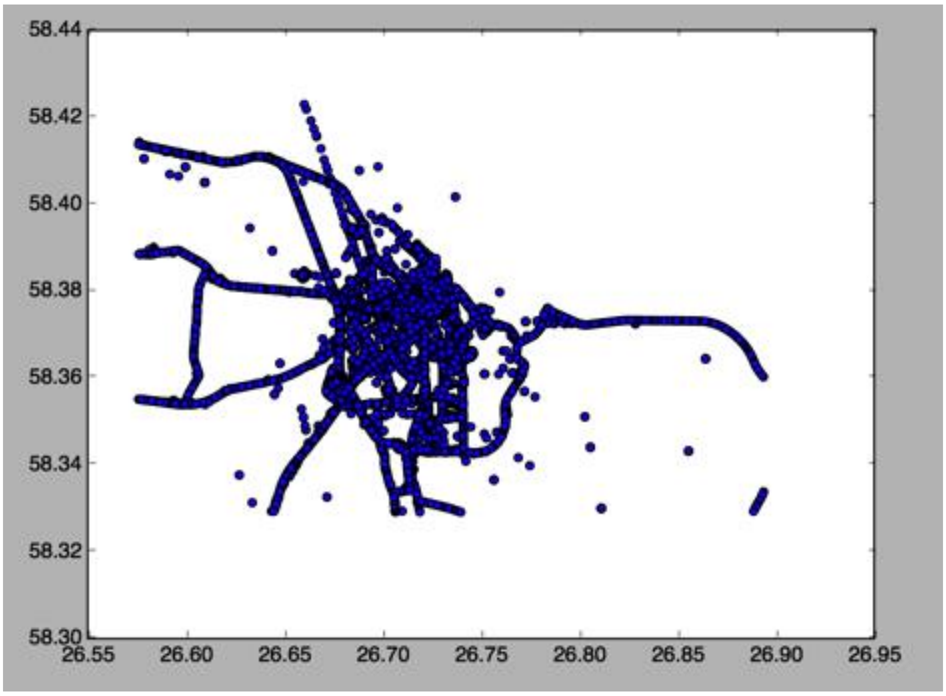


Figure 21. GPS data inside spatial filter

All data pipeline tasks were implemented as individual scripts developed in Python, Ruby or Bash. The implementation idea was to keep those scripts small and focused on concrete tasks. There were 13 different scripts that were representing some step in the data pipeline.

For the segmentation phase the following system parameters were used:

- *ST_TIME_THRESHOLD_FOR_CUT* - was set to 60 seconds. It defined the maximum allowed difference in time of sequential GPS points. As the outcome of time-based segmentation 3967 individual trips were generated.
- *ST_SEGMENTATION_POINT_THRESHOLD* - was set to 30 meters. Minimum number of points for a trip was set to 10. As the results there was 2164 trips generated.

Those extracted trips were classified to two types based on calculated mean speed. The results (figure 22):

- 532 ‘*driving*’-type trajectories
- 1632 ‘walking’-type trajectories

Only ‘*driving*’-type trajectories are used for further analysis.

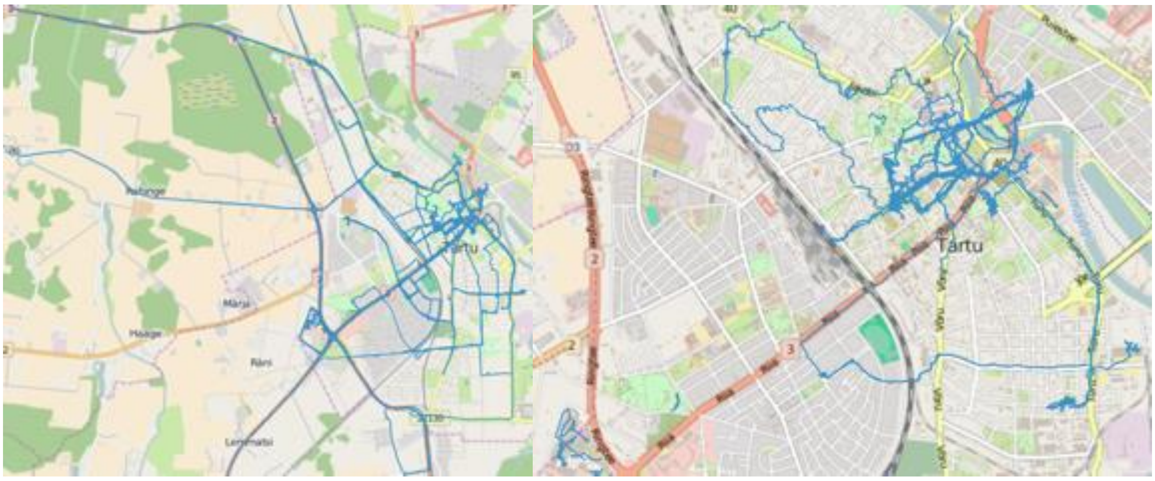
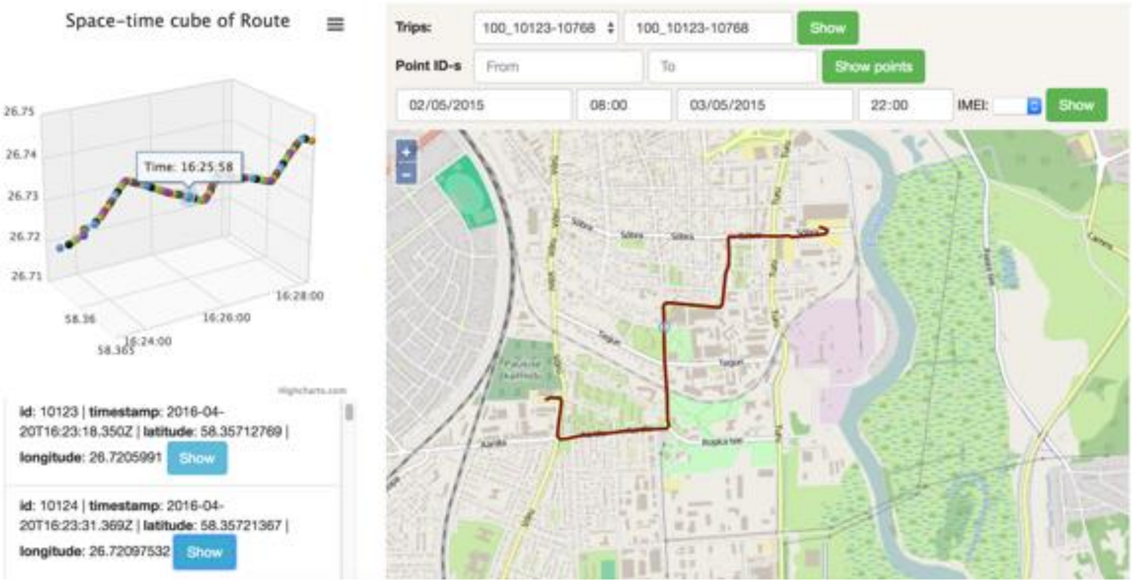


Figure 22. *driving*-type trajectories on the left;*walking*-type trajectories on the right

4.2.1 Visualization

The developed application has different visualization options for trajectories (figure 23). It’s possible to select existing trip from the list of all existing trips and display its spatial representation on the map and attributes in a list. Additionally there are options for filtering the global trajectory by timestamp and user to generate trajectories on the fly for visual check. Interactive 3-dimensional space-time cube allows visualizing the trajectory in space and in time.



Trips: 100_10123-10768 | 100_10123-10768

Point ID-s: From | To

02/05/2015 | 08:00 | 03/05/2015 | 22:00 | IMEI:

id: 10123 | **timestamp**: 2016-04-20T16:23:18.350Z | **latitude**: 58.35712769 | **longitude**: 26.7205991

id: 10124 | **timestamp**: 2016-04-20T16:23:31.369Z | **latitude**: 58.35721367 | **longitude**: 26.72097532

Figure 23. Trajectory visualization

With the aid of visualization options data errors were found. For some *imei* values multiple user trajectories were combined, which caused the following erratic back and forth jumping of the trajectory (figure 24). This data was extracted to multiple users and problem was solved.

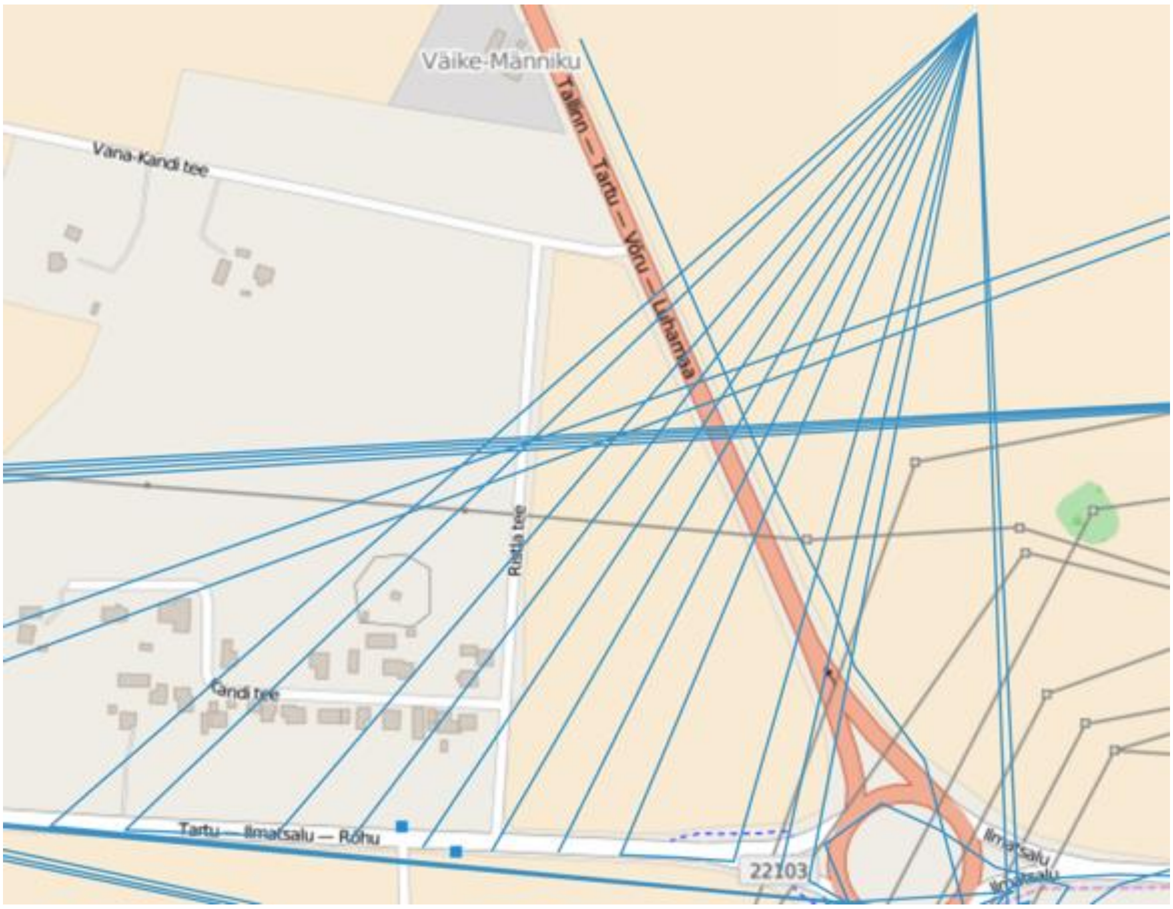


Figure 24. Errors in data source

4.3 Dynamic Road Segments Grid System Creation

The process started defining the system parameter *ST_SEGMENT_BUFFER_THRESHOLD*, which defines the threshold in meters for finding GPS points around every road segment. Only points that belong to a trajectory with type ‘*driving*’ are considered. In figure 25 it’s possible to see GPS error distribution for points distances from the road segment center and in figure 26 the distribution of point distances from the road segment per segment average. As the result 3951 grid cells were created (figure 27).

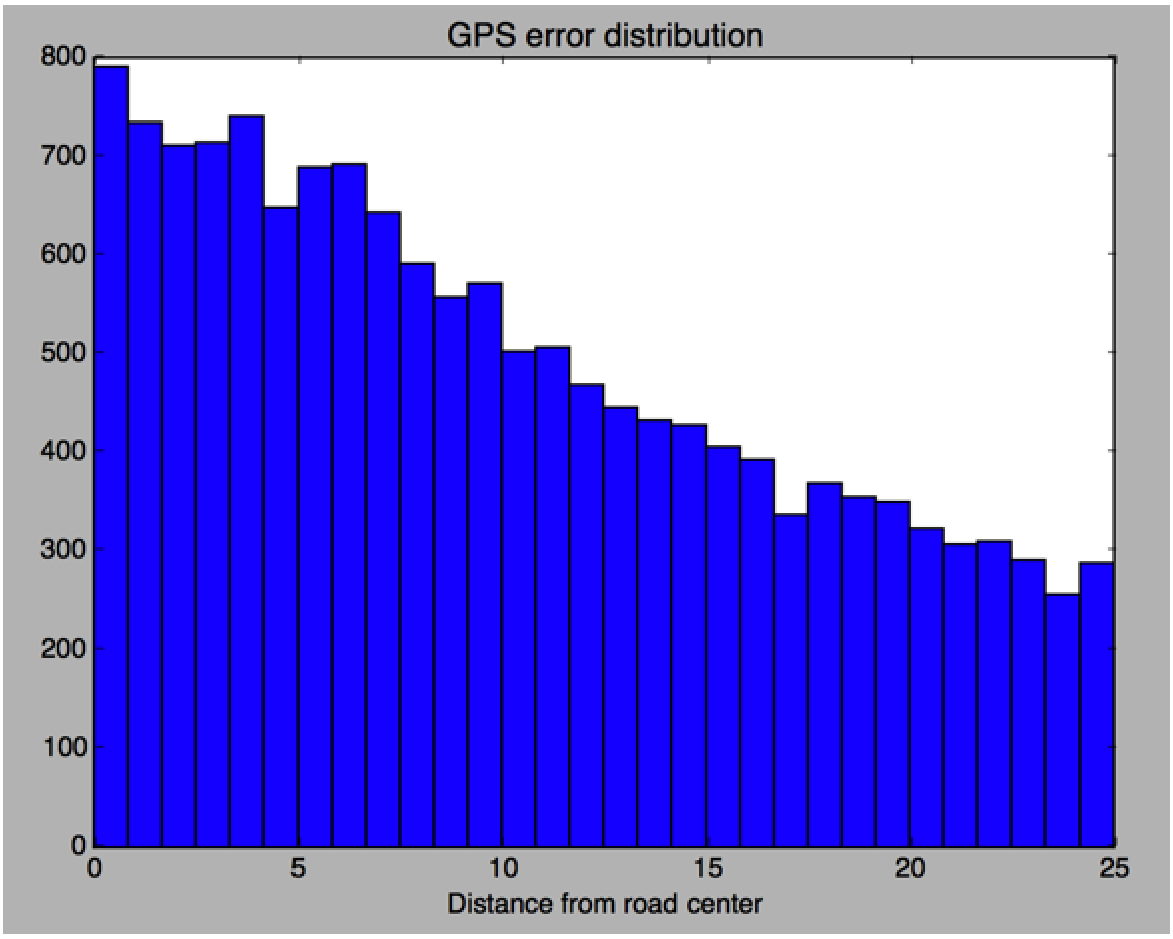


Figure 25. GPS error distribution by point

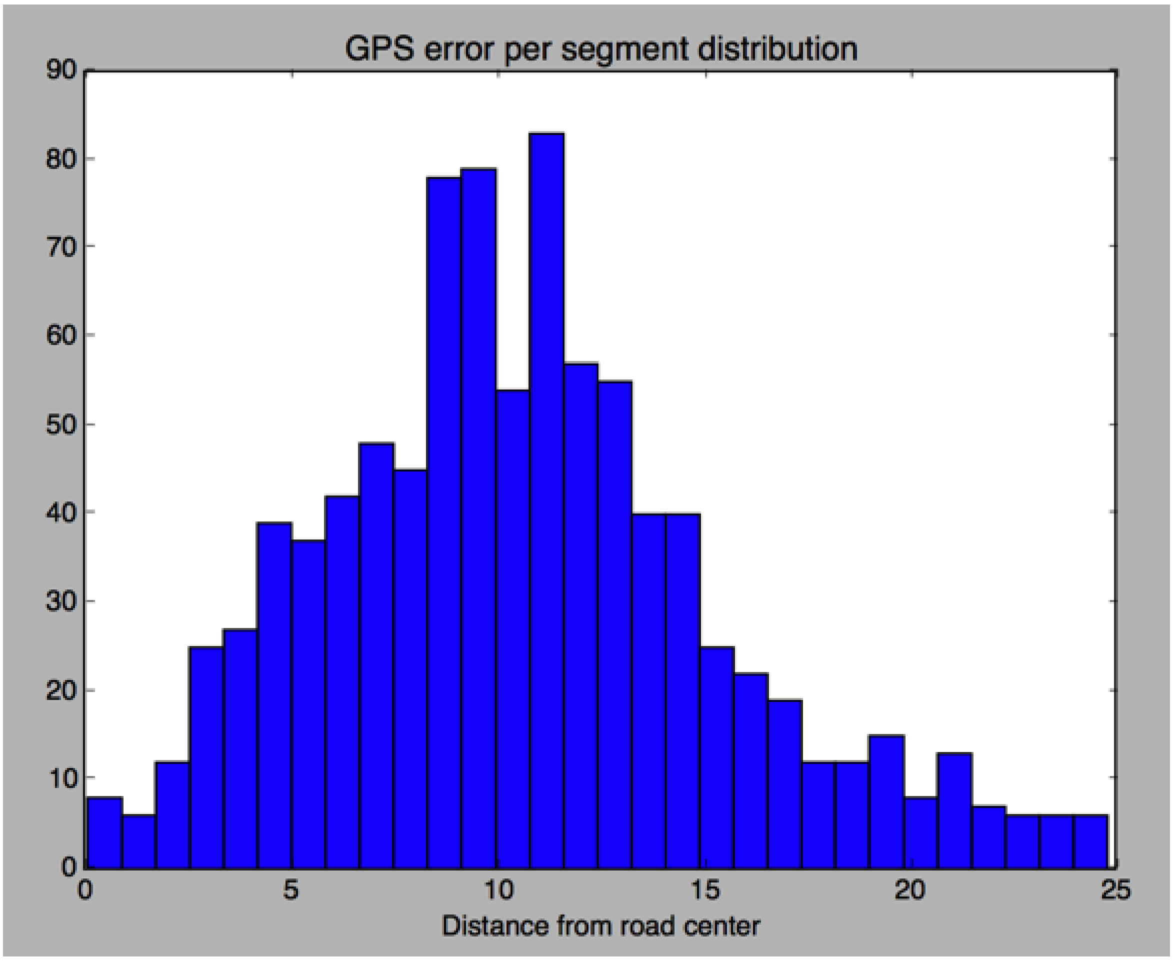


Figure 26. GPS error distribution by segment average

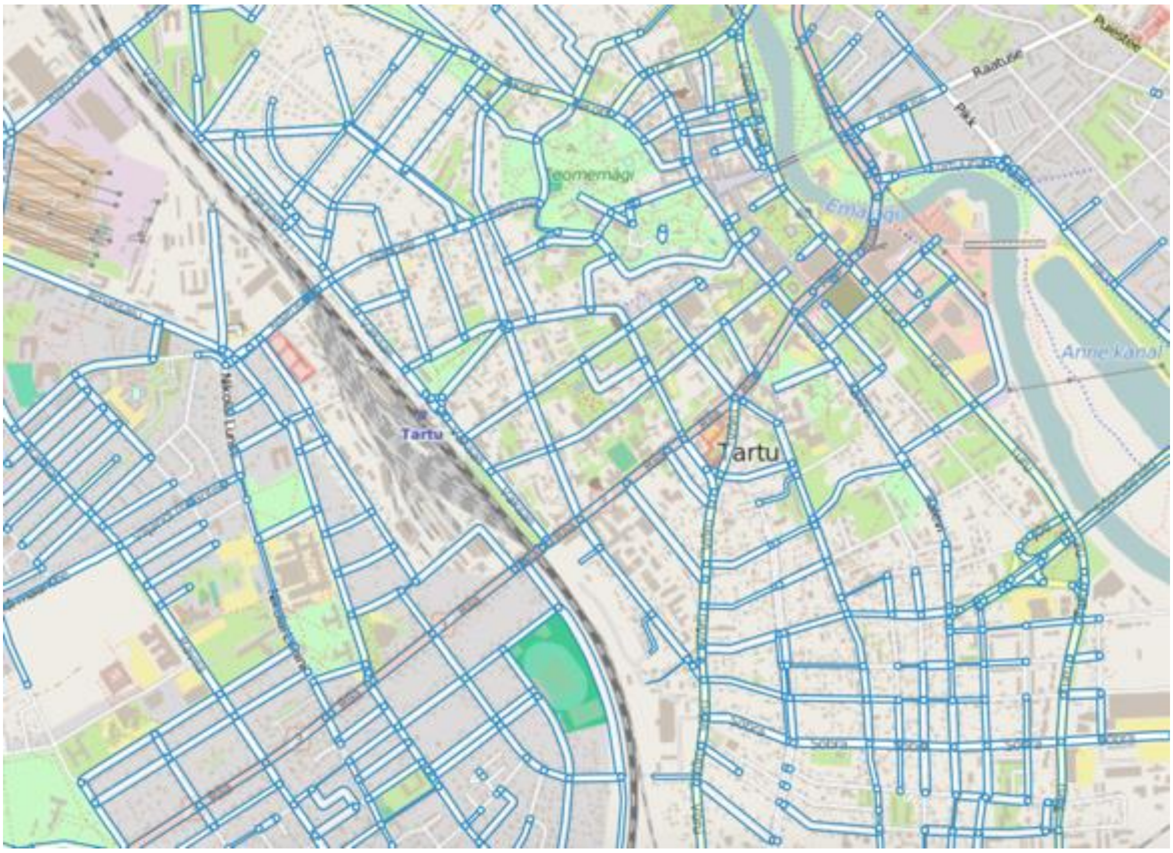


Figure 27. Generated road segments grid system

4.4 Trajectory Similarities

For all the trips with type '*driving*' the following connected grid cells were found:

- For every point in the trajectory corresponding grid cell was found if there existed such
- For every trajectory segment the list of corresponding grid cells were found if there existed such.

If start and end points or start and segments of a trajectory didn't have any corresponding grid cells, the first intersecting cell for a point or a trajectory segment was selected as the start or end grid cell accordingly. In figure 28 it's visible 2 similar trajectories.

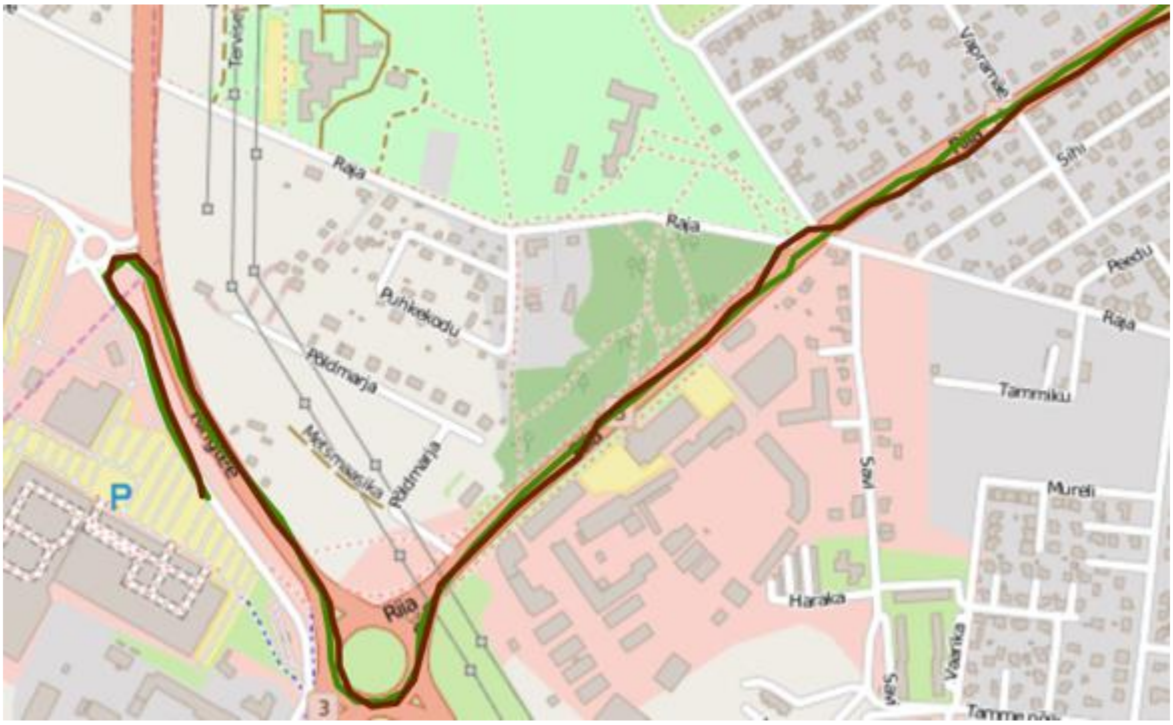


Figure 28. Example of two similar trajectories

4.5 Map-matching

For all the trajectories matching road segments were found and re-locating the coordinates of the point to the corresponding road segment was performed (figure 29). For the validation phase route correctness was checked. For every origin-destination pair in trajectories *ST_K_SHORTEST_PATH* candidate paths were found. *ST_K_SHORTEST_PATH* was set to 15. The process started finding the closest road segments for trajectory start and end points. Based on the road segment a node was found. The node that is closest is chosen if the road segment is not one-way street and the segments reverse cost is too high. In such case target node of the segment was taken. When no global paths were found, trajectories were checked segment by segment for the correctness and when possible alternative road segment was found. After validation phase moving the GPS points to matched road segments was performed again.

To give a score of the probability of matched trajectory being correct, confidence score was calculated for all trajectories.

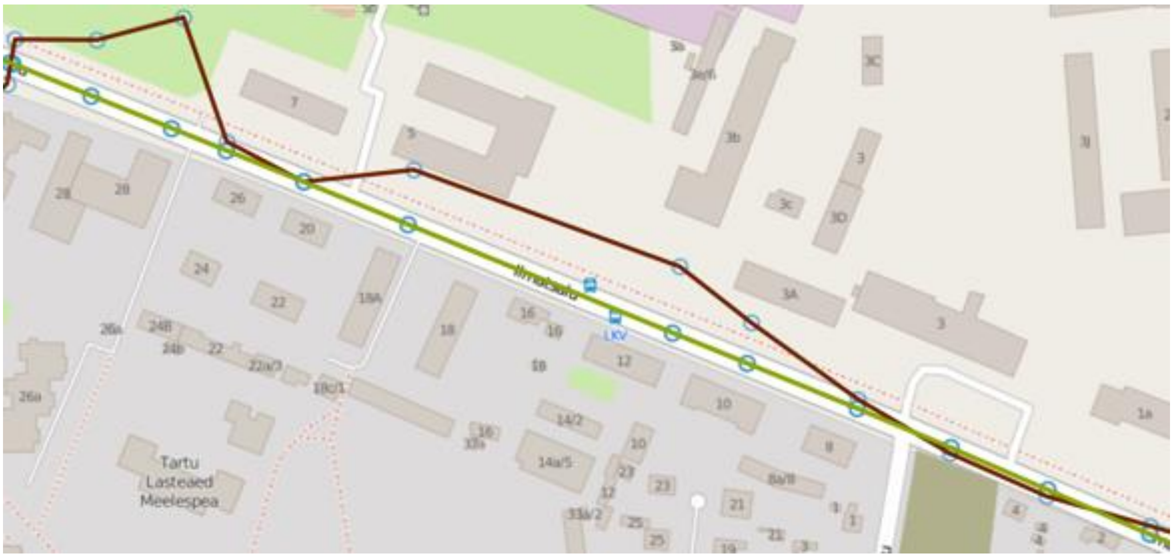


Figure 29. Comparison of original (in red) and matched (in green) trajectories

For validating the correctness of the results reference trajectories were selected for which the trajectories were known. There does not exist any good automated verification process for map-matching results. And the result of matched GPS points was compared to those known paths. The characterics of reference trajectories are listed in Table 1 and in Table 2 the outcome of map-matching is described.

Table 1. Characterics of reference trajectories

Description	**Value**
Number of trajectories	15
Number of GPS points	976
Number of *driving* trajectories	15
Average points per trajectory	66
Minimum points per trajectory	12
Maximum points per trajectory	140

Table 2. Results of map-matching reference trajectories

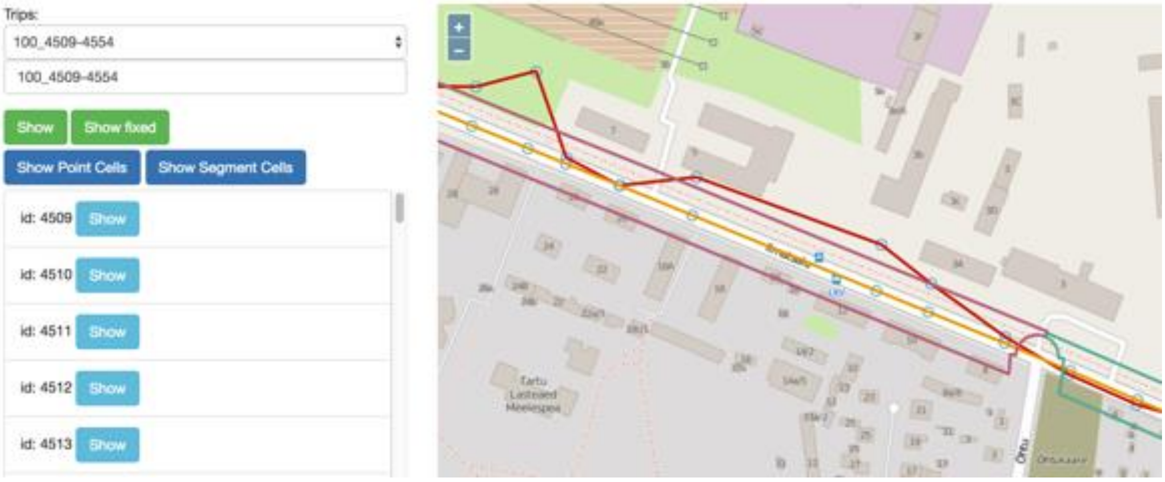
Description	**Value**
Percentage of correctly segmented trajectories	100%
Percentage of correctly classified trajectories	100%

Percentage of trajectories matched correctly for all points	47%
Percentage of trajectories matched partly	53%
Percentage of correctly matched GPS points	98.5%

The pre-processing steps were successfull of creating individual trajectories and classifying those by movement type. When comparing the map-matching results than among all the points 98.5% were matched correctly, out of the trajectories 47% were matched correctly for all points.

4.5.1 Visualization

For visual aid a view (figure 30) was added to the application, which displays original trajectory, shows corresponding grid segments and allows showing also the fixed trajectory in the same view.

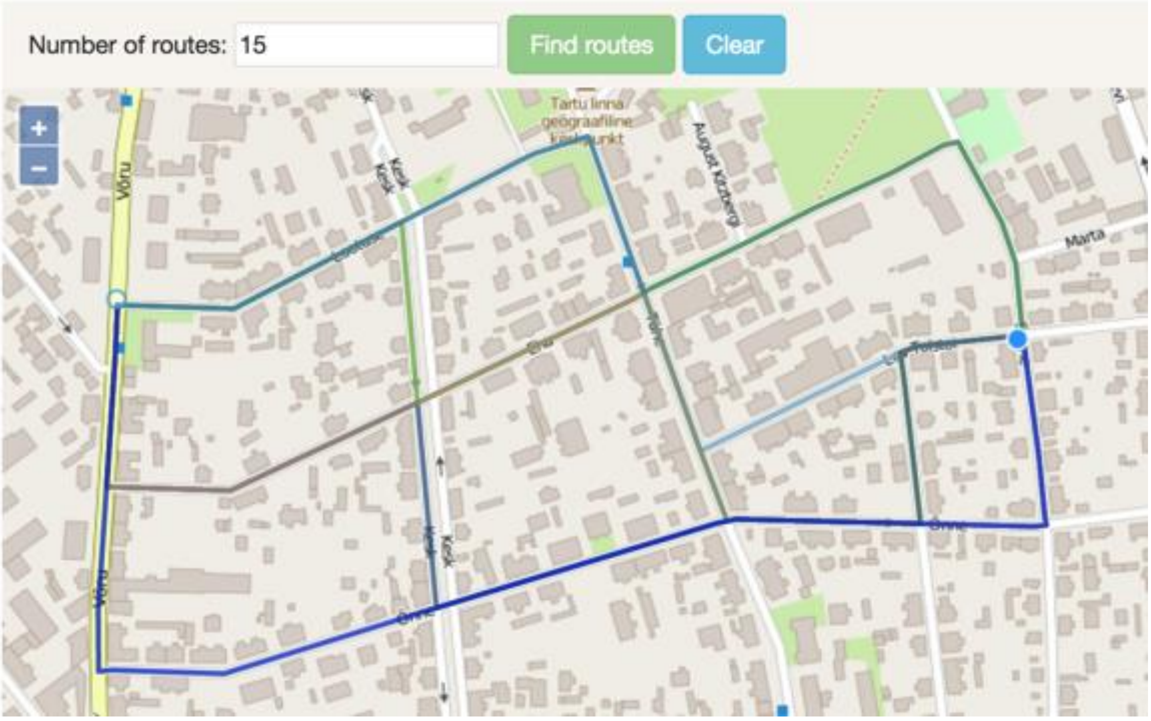


Trips: 100_4509-4554

100_4509-4554

Figure 30. View for comparing matched trajectories

In the visualization application it's possible to define any two points on the map and calculate *k* routes between them and visualize them all (figure 31).



Number of routes: 15

Figure 31. K-shortest routes between two points

4.6 Conclusion

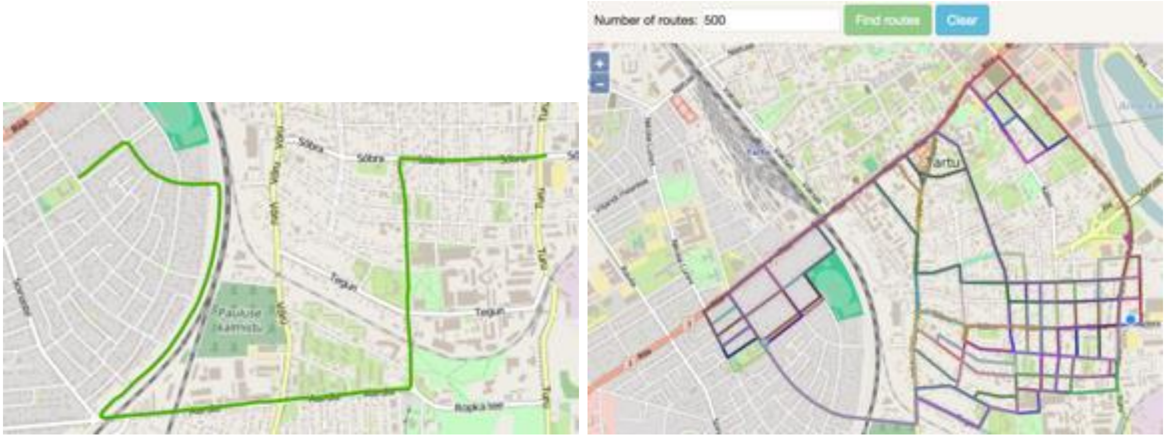
The applied methodology introduced a data pipeline for turning raw GPS points into trajectories by removing outliers, extracting global trajectories to segments and classifying found segment by movement type. The new grid system based on GPS errors was used for matching points to road segments. For every trajectory a confidence score was given based on its topological correctness and regularity.

For validation 15 reference trajectories were selected for which the real trajectory was known. And the outcome of the map-matching was compared to those paths. The results:

- Those 15 trajectories had 976 combined GPS points
- Average points per trajectory was 66, minimum was 12 and maximum 140
- 47% of trajectories were correctly matched for all points
- 53% of trajectories were matched partly, out of those 37.5% had 2 wrongly matched point and 62.5 had only 1 wrongly matched point
- From all points 98.5% were correctly matched

Some of the shortcomings were found:

- Errors in underlying GIS data. The road network layer from OpenStreetMap had multiple roads with wrong restrictions, which complicated fixing the topological correctness of the trajectory. It is an issue with all map-matching algorithms. For that the confidence score was introduced to find out more unlikely trajectories in some other way.
- For more unlikely and long trajectories finding alternative routes for origin-destination pairs does not provide hoped results (figure 32). It’s necessary to apply routing partially.



Number of routes: 500

Figure 32. Left - original trajectory; right - 500 alternative routes for origin-destination pair

5 Conclusions and Future Perspectives

5.1 Conclusion

With the applied data pipeline and visualization application we have shown how raw GPS data can be turned into trajectories and matched to the underlying road network. Here are some of the conclusion based on the methodology and results:

- Dynamic grid system based on GPS measurement errors provides good way of matching points to road network.
- Bigger dataset with more regular trajectories would be beneficial for pattern extraction
- Better underlying geospatial data provides better results. Extracted trajectory regularity patterns could help make correction to road network.

5.2 Future Perspectives

For the future the implemented framework could be made more scalable as it was not the main goal of this thesis work. To be able to support massive amount of spatiotemporal data for map-matching restructuring some parts of the application would be needed. Fortunately the existing data pipeline is made of multiple small scripts that could be distributed to multiple nodes and later results be merged.

To support matching pedestrian trajectories would need better underlying GIS data, because at the moment the pedestrian road network data available is largely missing.

The visualization application could be further developed to be part of the decision making process for some domain problem solving that require to work with trajectory data.

The applied methodology would need some enhancements to work with more parse data (e.g. sampling rate is 2-5 minutes), as some of low-powered devices can’t afford storing location too often.

References

[1] Soora Rasouli, Harry Timmermans "Mobile Technologies for Activity-Travel Data Collection and Analysis". IGI Global, Chapter 10, 2014.

[2] Dieter Pfoser, Christian S. Jensen “Capturing the Uncertainty of Moving-Object Representations”. In Proceedings of the 6th international Symposium on Advances in Spatial Databases, 111-132, 1999.

[3] Peter Ranachera, Richard Brunauerb, Wolfgang Trutschnigc, Stefan Van der Spekd, Siegfried Reichb “Why GPS makes distances bigger than they are”. International Journal of Geographical Information Science, 30 (2), 316–333, 2016.

[4] Mohammed A. Quddusa, Washington Y. Ochiengb, Robert B. Nolandb "Current map-matching algorithms for transport applications: State-of-the art and future research directions". Transportation Research Part C: Emerging Technologies. Volume 15, Issue 5, Pages 312–328, 2007.

[5] Mohammed A. Quddus "Map Matching Algorithms for Intelligent Transport Systems”. Handbook of Research on Geoinformatics, 2009.

[6] Paul Newson, John Krumm "Hidden Markov Map Matching Through Noise and Sparseness". 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2009.

[7] Josh Jia-Ching Ying, Bo-Nian Shi, Kun-Chan Lan and Vincent S. Tseng “Spatial-temporal Mining for Urban Map-Matching”. UrbComp The 3rd international workshop on Urban Computing, 2014.

[8] Yang Li, Qixing Huang, Michael Kerber “Large-Scale Joint Map Matching of GPS Traces”. SIGSPATIAL Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2013.

[9] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, Yan Huang “Map-Matching for Low-Sampling-Rate GPS Trajectories”. ACM SIGSPATIAL GIS, 2009.

[10] Shashi Shekhar, Pusheng Zhang, Yan Huang, Ranga Raju Vatsavai "Trends in Spatial Data Mining”. AAAI/MIT Press, 2003.

[11] Yu Zheng “Trajectory Data Mining: An Overview”. ACM Transactions on Intelligent Systems and Technology (TIST), v.6 n.3, 2015.

[12] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, Fabio Pirelli, Chiara Renso, Salvatore Rinzivillo, Roberto Trasarti “Unveiling the complexity of human mobility by querying and mining massive trajectory data”. The VLDB Journal, 2011.

[13] Gennady Andrienko, Natalia Andrienko, Stefan Wrobel “Visual Analytics Tools for Analysis of Movement Data”. SIGKDD Explorations, 9, 38-46, 2007.

[14] Gennadi Andrienko, Natalia Andrienko Peter Bak, Daniel Keim, Stefan Wrobel “Visual Analytics of Movement”. Springer, 2013.

[15] Mordecai Haklay "How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets”. Environment and Planning B: Planning and Design, 37 (4), 682-703, 2010.

[16] Maike Buchin , Anne Driemel , Marc van Kreveld , Vera Sacristán “An algorithmic framework for segmenting trajectories based on spatio-temporal criteria”. Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010.

[17] Katarzyna Siła-Nowickaab, Jan Vandrolc, Taylor Oshand, Jed A. Longa, Urška Demšara, A. Stewart Fotheringhamd “Analysis of human mobility patterns from GPS trajectories and contextual information". International Journal of Geographical Information Science, 30, 881-906, 2016.

[18] David Eppstein “Finding the k shortest paths”. In: Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS’94), 154–165, 1994.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, Margus Haavala,

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

Mobility Data Mining for Rural and Urban Map-Matching, supervised by Dr. Amnir Hadachi.

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, **25.05.2016**