



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

## Masterthesis

# Large-Scale Accessibility Analysis Using OpenStreetMap Data

vorgelegt von

Martin Poppinga

Fakultät für Mathematik, Informatik und Naturwissenschaften

Fachbereich Informatik

Arbeitsbereich Wissenschaftliches Rechnen

Studiengang: M.Sc. Informatik

Matrikelnummer: 6318650

Erstgutachter: Prof. Dr. Thomas Ludwig

Zweitgutachter: Dr. Julian Kunkel

Betreuer: Dr. Julian Kunkel

Hamburg, June 6, 2018





## Abstract

By using accessibility analyses, the quality of coverage of institutions can be determined, giving knowledge how many persons can be reached within a given time. Such analyses have a variety of use-cases in the public, private and commercial sector. Existing approaches showed precise results only for smaller regions or were at the expense of precision in a small-scale context. Additionally, the used datasets and software frameworks are often proprietary, which hinders reproducibility in research.

In this thesis, it is shown that large-scale accessibility analyses are possible using open data while retaining a high precision. Also, the data cleaning and preparation is presented. Further is demonstrated that obtained information from different analyses can be combined for more complex analyses and can be used for solving location optimization problems. For evaluation, different studies for Germany are sketched, and the outcomes are shown. Examples are the coverage of fire stations, hospitals, stroke units or charging stations for electric cars. The results show that it is possible to achieve a more precise resolution as existing approaches, enabling analyses using small-scaled aspects while maintaining country-sized expansion and acceptable computation times. This advancement enables, for example, further studies in spatial and urban research.

## Zusammenfassung

Mittels Erreichbarkeitsanalysen kann die regionale Abdeckungsqualität von Einrichtungen bestimmt werden. Dies ermöglicht es zu bestimmen, wie viele Menschen innerhalb einer bestimmten Zeit erreicht werden können. Relevant ist dies für verschiedenste Anwendungsbereiche im öffentlichen, privaten als auch im gewerblichen Sektor. Existierende Ansätze zeigten präzise Ergebnisse nur für kleinere Regionen oder gingen zulasten der Präzision im kleinräumigen Kontext. Zusätzlich kommt es oft zum Einsatz von proprietären Datensätzen oder Software, wodurch eine wissenschaftliche Reproduzierbarkeit eingeschränkt wird.

In dieser Arbeit wird gezeigt, dass es mittels offener Daten möglich ist großräumige Erreichbarkeitsanalysen zu erstellen, welche eine hohe Präzision behalten. Zusätzlich wird die Datenaufbereitung gezeigt und weiterführenden Analysen, wie das Kombinieren von verschiedenen Erreichbarkeitsanalysen und Standortoptimierungen durchgeführt. Zur Evaluation wurden verschiedene Szenarien für Deutschland bearbeitet, unter anderem die Abdeckung der Feuerwehren, Krankenhäuser, Schlaganfallversorgungsmöglichkeiten oder Ladestationen für Elektroautos. Die Ergebnisse zeigen, dass es möglich ist eine höhere Präzision als existierende Ansätze zu erreichen, um kleinräumliche Aspekte zu berücksichtigen und es dennoch zu ermöglichen die Rechenzeiten, während der Durchführung von Analysen mit landesweitem Umfang, in einem akzeptablen Rahmen zu halten. Diese Verbesserungen ermöglichen weitere Studien in der Raum- und Stadtforschung.



# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Motivation . . . . .	9
1.2	Goals . . . . .	11
1.3	Structure of this Thesis . . . . .	12
<b>2</b>	<b>Background</b>	<b>13</b>
2.1	Navigation and Routing . . . . .	13
2.1.1	Graph . . . . .	13
2.1.2	Overlay Graph . . . . .	13
2.1.3	Planar Graph . . . . .	14
2.1.4	Finding the Shortest Path . . . . .	14
2.1.5	Accessibility Analysis . . . . .	15
2.1.6	Isochrones . . . . .	15
2.1.7	Voronoi Diagrams . . . . .	15
2.1.8	Heavy Traffic . . . . .	15
2.2	OpenStreetMap . . . . .	16
2.2.1	Project Overview . . . . .	16
2.2.2	Limitations and Extent of the Dataset . . . . .	16
2.2.3	Data Quality . . . . .	17
2.2.4	Data Format . . . . .	17
2.2.5	Other Maps . . . . .	19
2.3	Political Borders . . . . .	19
2.4	Population Data . . . . .	19
2.5	Humanitarian Statistical Datasets . . . . .	20
2.6	Geo Data Processing and Analysis . . . . .	20
2.6.1	Geographic Information Systems . . . . .	20
2.6.2	Storing Geospatial Data . . . . .	20
2.6.3	Spatial Index . . . . .	21
2.7	Emergency Services . . . . .	21
2.7.1	Special Rights / German Sonder- und Wegerecht . . . . .	21
2.7.2	Reaction Times . . . . .	21
2.7.3	Fire Department Demand Reports . . . . .	22
<b>3</b>	<b>Related Work</b>	<b>23</b>
3.1	Accessibility Analysis . . . . .	23
3.1.1	Sampling . . . . .	23
3.1.2	Isochrones . . . . .	24

3.1.3	Further Approaches . . . . .	26
3.2	Population Data . . . . .	26
3.3	Routable Street Network . . . . .	26
3.4	Estimating Movement Speeds . . . . .	27
3.5	Programming Frameworks and APIs . . . . .	27
3.5.1	Commercial APIs . . . . .	27
3.5.2	Programming Libraries . . . . .	28
3.5.3	pgRouting . . . . .	28
3.5.4	GIS Software . . . . .	28
3.5.5	Open Source Navigation Software . . . . .	28
3.5.6	Comparison . . . . .	29
<b>4</b>	<b>Exploration</b>	<b>31</b>
4.1	Population Datasets . . . . .	31
4.2	Quality of POI in OSM . . . . .	34
4.2.1	Comparing to Official Sources . . . . .	34
4.2.2	Comparing to Google Maps . . . . .	34
4.3	Visualizing Results . . . . .	35
<b>5</b>	<b>Design</b>	<b>39</b>
5.1	Overview . . . . .	39
5.1.1	Workflow . . . . .	40
5.2	Analyzed Use-Cases . . . . .	40
5.2.1	Service Coverage . . . . .	40
5.2.2	Quality Map: Individual Combination of Analyses . . . . .	43
5.2.3	Location Optimization: Determining a Good Settlement Point to Increase Coverage . . . . .	44
5.3	Data Handling . . . . .	45
5.4	Data Preparation . . . . .	46
5.4.1	Points of Interest . . . . .	48
5.4.2	Political Boundaries . . . . .	48
5.4.3	Movement Models . . . . .	49
5.4.4	Refining Population Data . . . . .	51
5.5	Accessibility Analysis - Navigation Task . . . . .	53
5.5.1	Complete Analysis . . . . .	53
5.5.2	Fast Analysis . . . . .	54
5.5.3	Parallelization . . . . .	54
5.5.4	Alternative Models . . . . .	55
5.6	Analyzing the Results . . . . .	55
5.6.1	Population Coverage . . . . .	55
5.6.2	Highway Coverage . . . . .	55
5.6.3	Statistics per POI . . . . .	56
5.6.4	Combining Different Scenarios . . . . .	56
5.6.5	Comparing Driving Time . . . . .	57

5.6.6	Isochrones . . . . .	57
<b>6</b>	<b>Implementation</b>	<b>59</b>
6.1	Techniques . . . . .	59
6.2	Data Preparation . . . . .	59
6.2.1	Street Network . . . . .	60
6.2.2	Points of Interest . . . . .	60
6.2.3	Population . . . . .	61
6.2.4	Administrative Boundaries . . . . .	61
6.3	Computation . . . . .	62
6.4	Evaluation . . . . .	63
6.4.1	Isochrones . . . . .	63
6.4.2	Plotting . . . . .	64
6.4.3	Statistical Evaluation . . . . .	64
<b>7</b>	<b>Results</b>	<b>65</b>
7.1	General Evaluation . . . . .	65
7.1.1	Comparing to Navigation Software . . . . .	65
7.1.2	Quality of the Street Network . . . . .	66
7.1.3	Population Downscaling . . . . .	66
7.2	Scenario A: Service Coverage . . . . .	68
7.2.1	Fire Station Coverage in Germany . . . . .	68
7.2.2	Fire Station Coverage in Northern Germany . . . . .	74
7.2.3	Hospital Coverage . . . . .	77
7.2.4	Electric Car Charging . . . . .	77
7.2.5	Stroke Units . . . . .	80
7.2.6	Pharmacies . . . . .	81
7.2.7	Discussion and Conclusions . . . . .	81
7.3	Scenario B: Combined Quality Map . . . . .	83
7.3.1	Holiday/Leisure Quality . . . . .	83
7.3.2	Living Quality . . . . .	85
7.3.3	Discussion and Conclusions . . . . .	87
7.4	Scenario C: Location optimization . . . . .	87
7.4.1	Logistics Center . . . . .	87
7.4.2	Stroke Service Vehicle . . . . .	90
7.4.3	Discussion and Conclusions . . . . .	91
<b>8</b>	<b>Conclusion</b>	<b>93</b>
8.1	Summary . . . . .	93
8.2	Remarks . . . . .	94
8.3	Further Work . . . . .	94



# 1 Introduction

*In Section 1.1, a general introduction in this work is given. Section 1.2 gives an impression of the goals of this thesis and Section 1.3 gives an overview of the structure of the following chapters.*

## 1.1 Motivation

How many people and which places can I reach in 15 minutes? Where is the best place to settle down, or to be reached by as many people as possible? These and similar questions are highly relevant for retail and logistics as well as in emergency situations. Delayed arrival of emergency services could even cause fatalities. Accessibility analyses are an essential tool for location planning and studies in spatial and urban research.

Investigating if the essential services sufficiently cover the areas of interest, as well as looking at possible optimizations, is crucial for a wide range of fields: Proper placement of facilities and short transportation times have a significant impact on profits for commercial services. So are fast and optimized logistics routes to transport wares to customers or other locations of the cooperation a vital aspect of business planning. A good settlement place has an impact on the living quality, for example, preferring short commuting to public services like schools. Furthermore, individuals looking for a new home can improve their decision, and local authorities can improve the overall living quality in their communities. Moreover, in emergency situations, it is indispensable to the health and lives of the persons concerned to reach them quickly. It is possible to optimize this transportation times and identify problems as well as finding new opportunities using different data sources and techniques.

For the safety of the population, one example of an important aspect is the service coverage of fire departments. In the case of an emergency, it is in most cases not relevant which fire department is reacting, as long as the alarmed forces can reach the place of emergency in the shortest possible time. The latest tolerable arrival time is also often determined by officials and needed to ensure the safety of the population. Unmanned fire departments or barely reachable areas are a risk. In times of urbanization and rural exodus, especially voluntary fire departments, as they are common in Germany, are facing the problem that it gets increasingly challenging manning the stations and to preserve the needed coverage. As a result, more often forces from neighboring departments are required supporting the local communities.

To determine if constraints are fulfilled or to find new solutions, analysis capability is required. Analyses for cases like this, where short reaction times are crucial or desirable, allow on one side to decide where to place new service locations to enable needed coverage or on the other side to show which sites are dispensable for maintaining sufficient coverage. Furthermore, it would be possible to create precise fields of responsibility, where institutions can see how much of the population is covered by which station. Inhabitants can see which location is reachable fastest. Gained information from these analyses allows optimization in many ways. Currently, stating an analysis is in many cases just a vague approximation. Even if precise information is crucial in a humanitarian or financial point of view, accurate models are often missing. Different methods exist to measure the coverage of a service. In many cases, only the linear distance or similar simplistic models are used to define the catchment area. While easy to compute, these do not take geographical or human-made obstacles into account. Rivers or railway tracks can increase driving times a lot. Therefore, the topology of the transportation network is crucial for accurate estimations. Other existing approaches, which are only approximating coverage areas, are missing a sufficient detail level, too, which distorts the results.

Finding the optimal way from a point **A** to a point **B** can be solved by navigation systems or other routing services, with improving results. However, for analysis of service quality, it is in most cases more relevant to evaluate the accessibility to a set of places, like the entire population of a region or country. An analysis should be possible for large-scaled areas like whole countries or continents, allowing to compare service quality across regions. While existing approaches reach this goal by reducing the precision of their results and approximating the quality on a fine scale, this vagueness prevents fine-grained local analyses.

For companies, by combining several analyses, multiple aspects can be taken into concern, like supply chains, competitors, and customer reachability. This knowledge can be attractive for the service offering side, finding optimal locations for new facilities, as well as for the customers or other parts of the population. Further, families could search for a new apartment based on a set of requirements, like the driving distance to work, the walking distance to elementary school and the number of nearby playgrounds. Facilities could be placed where currently a low service coverage exists, and the transportation network allows access to as many persons as possible, optimizing overall coverage.

To reach the full potential of accessibility analyses, the workflow must be capable of maintaining a high precision on a small scale, while analyzing a large-scale scenario. Different approaches exist for performing accessibility analyses but have their flaws in either precision or scalability. Other important aspects are the customizability and the reproducibility. High costs for tools and datasets are a hindrance to research as well as public services and smaller businesses. Using open tools and data sources enables reproducible and highly customizable analyses combining multiple datasets into one analysis. This allows further and more precise studies in urban and spatial research.



## 1.2 Goals

The goal of this thesis is to enable and explore accessibility analyses using heterogeneous data sets on any-sized areas giving information on a street or building precise resolution. It shall be shown how to gain knowledge from open data sources and using it to investigate follow-up questions. Finally, different scenarios are sketched and explored, representing a group of questions which could be answered.

For achieving this goal, the following methodological issues are addressed:

- Data extraction and cleaning. Preparing the used datasets for later usage in accessibility analyses
- Performing an accessibility analysis on a country-sized (e.g. Germany) datasets
- Analyzing coverage and statistics of several thousand points of interest (POI) at once
- Maintaining a precise (i.e. single streets) resolution of the results
- Combining different analyses to create individual ratings of regions and places
- Using these techniques to propose and evaluate additional locations for optimized coverage

Different transportation models are sketched and prototyped to allow a more precise approximation depending on the scenario. It also allows comparing different ways of transportation. The here sketched models include walking, cycling, driving and driving of emergency vehicles with privileges.

Furthermore, all tools and data sources used in this thesis are open or under free-to-use licenses. The primary data source is the *OpenStreetMap* project, an open map and database for various kinds of geospatial data. This way, results can be used or refined without fees for the data sources or the frameworks. Beside from being costly, several non-open datasets restrict publishing results or forbid local processing at all. Using open data gives community services and individuals the opportunity to use or reproduce this information. Using a self-designed tool-chain allows to investigate the different parts of the process better and provides more flexibility, while creating and evaluating results and intermediate results, over existing frameworks. Also, no external Application Programming Interface (API) is used while performing an analysis, so no API rate-limitations and related interruptions are of concern, allowing to create large-scale analyses. The street network is in this work enriched with population data to determine the number of inhabitants which can be reached.

For evaluation, different scenarios are sketched. Coverage for fire stations, hospitals, and other POI are computed for Germany. Also, location optimization results and combined analyses are presented.

## 1.3 Structure of this Thesis

In Chapter 2, the used techniques and datasets are presented and their possibilities and limitations described. Following, in Chapter 3, existing and alternative approaches are described. In Chapter 4, the different datasets will be presented and discussed. Chapter 5 gives a general overview of the proposed workflow and sketches some scenarios which will be part of the evaluation. It also describes the different parts and approaches of the toolchain, while Chapter 6 gives information about challenging aspects of the implementation. In Chapter 7 the results of the analyses are presented and evaluated. Chapter 8 concludes and discusses the presented approaches.

*In the next Chapter, background information to used techniques and datasets is presented.*

## 2 Background

*In this Chapter, an introduction to the basics, datasets, and approaches is given. At first in Section 2.1, fundamentals of graph routing and navigation are shown. Following, in Section 2.2, the OpenStreetMap project and its dataset is described. It gives a brief overview which data exists and is used in research. In Sections 2.3 to 2.5, more in this work used datasets are introduced. In Section 2.6, some fundamental techniques of processing geospatial data are briefly described which were further used in this work. Finally, an overview of particular characteristics of emergency services is given in Section 2.7.*

### 2.1 Navigation and Routing

For analyzing geographic data and performing navigation tasks, the data needs to be represented as a graph. The graph contains the spatial position of the streets and other information like for example the *transportation time*, the time needed from one position to the other.

#### 2.1.1 Graph

A graph is an abstract data structure of objects and connections between them. The objects in a graph  $G = (V, E)$  are called *vertices*  $V$  and the connections *edges*  $E$ . In graph theory, diverse types of graphs are described. The edges and therefore the graph can either be directed if the movement of an edge is limited to one direction or be undirected if the direction of the edge is not of concern. In a weighted graph, the edges have a weight, representing for example transitions costs. Weighted directed graphs are often called networks [dL13].

#### 2.1.2 Overlay Graph

An overlay graph describes a graph with a reduced complexity which approximates the original graph [HSW09]. This can increase the performance of search tasks. Also, it can give an overview of range if smaller streets are not of concern and do not need to be computed. Instead, a whole area can be marked as reached. Multiple levels of overlay graphs can be combined providing a hierarchy, allowing to decrease the number of edges which are to be considered while performing a shortest-path query.

### 2.1.3 Planar Graph

A planar graph defines a graph which can be embedded in a plane without intersecting edges aside of the vertices [DSSW09]. This property is exploited by various algorithms in the spatial context. In a road network, bridges and tunnels are violating this property in contrast to junctions which are represented by a vertex.

### 2.1.4 Finding the Shortest Path

Routing is an abstract task to find the best path between two points in an environment, for example, in a road network. A discrete environment is usually defined by different positions and movement costs between them. Continuous environments are discretized and can be approximated in a graph. Such an environment is then represented by vertices and weighted edges. The definition of the best path is dependent on the given scenario. The scenario defines which weights the different edges have, in the context of a road network it could be any combination of time, distance, tolls or similar metrics.

Different shortest path algorithms exist for solving a routing problem. The search can either be informed, if a heuristic of the distance to the goal is given, or uninformed if no such heuristic is used. The main difference between different techniques is the way the search area is expanded. The algorithms perform differently regarding the completeness, needed memory, time consumption as well as other metrics. Also, the number of possible start- and endpoints has to be taken into account.

The Dijkstra algorithm [Dij59] works on directed and weighted graphs. It guarantees to find the shortest path in all graphs not containing negative edge costs. It is a greedy algorithm, as it always expands to the unvisited node with the cheapest path. For this, it stores the paths costs to every visited node. It takes the vertex  $u$  with the minimum travel cost and expands it with the following transitions costs to all its successors. Because the node with minimum cost is selected each time, and negative weights are not allowed (for optimal paths), the path to  $u$  is optimal as all paths to  $u$ , which are now explorable, have at least the current cost of  $u$ . The Dijkstra algorithm performs an uninformed search in contrast to, for example, the A-star heuristics [DSSW09]. A-star optimizes this approach as the heuristics value is also taken into concern.

If the successor relation is stored together with the minimal cost, the shortest paths can be reconstructed. Otherwise, only the minimal costs are available.

As the task of finding the minimal node in large networks is time-consuming, it can be optimized using a minimum-heap (priority queue), a self-organizing data structure for extracting the minimal element in  $O(\log n)$ . It improves the runtime performance but needs more memory than a list.

### 2.1.5 Accessibility Analysis

Accessibility describes the quality of being able to be reached in a geospatial network. Different approaches are possible to perform such an analysis. Given one or more start points, the time to reach a group of points or an area is analyzed. In the following, these start or goal points which are used in analytics are called *points of interest (POI)*. An area is covered if at least one POI can reach the inhabitants in the defined time.

The most straightforward approach is to use the linear distance from a set of POI. While trivial to compute, it has its limitations regarding topological effects. Using a street network or other transportation topologies typically increase the quality of the results. Different strategies are possible which are given a closer look in Section 3.1.

### 2.1.6 Isochrones

A common approach to illustrate the accessibility is to obtain the area which is reachable in a given time. Often this is displayed with isolines, similar to showing the altitude of the area but representing the borders of the same needed time. Such isochrones are usually an approximation as for better visibility the areas between streets are enclosed, and the complexity of the polygons is reduced. Different approaches how to create isochrones are described in Section 3.1.2.

### 2.1.7 Voronoi Diagrams

An often-used technique to display the area of responsibility, the area for which a specific POI is the best choice, is the Voronoi diagram [Aur91]. It displays the areas with the shortest linear distance to each POI. It creates a set of polygons, defining the area in which the given POI has the shortest linear distance.

### 2.1.8 Heavy Traffic

Depending on the region and the time, the traffic situation has a small or big impact on the movement time. Different approaches exist to estimate the delay comparing to an uninfluenced model. In most cases, a model is created per street for each point of time within a week. The difficulty is to obtain the current traffic situation to fit the model. Some vendors of navigation systems, like *Waze*, or *Google Maps*, use the current movement speeds of their customers. Others rely on information provided by the *Traffic Message Channel (TMC)*. As this data is not available for all streets and smaller alternative routes usually cannot handle the traffic throughput of a highway, most systems display this information but use the estimated time to pick alternative high-capacity routes which show less traffic-related delay.

## 2.2 OpenStreetMap

### 2.2.1 Project Overview

The *OpenStreetMap* project is an open database for geospatial data with its primary focus on creating a free-to-use map. Starting 2004 with only a street network, it is now a database for different kinds of geospatial information [LWZ09]. It is now the largest project of *Volunteered Geographic Information (VGI)* as the data on OpenStreetMap is mostly crowd-sourced or provided by institutions under a free license. Similar to other crowd-sourced projects like the *Wikipedia* it has reached a high density of information as users can add information around their current location to the OSM database.

Crowd-sourcing allows committing changes that occur in the real world like new or closed streets, within minutes to the database. It also allows creating maps for regions which had no official data. This newly created information is then available worldwide within a few minutes. Everyone can add changes to this dataset, using a variety of editors and tools. So far over one million persons have contributed to the map<sup>1</sup>. The database does not only contain authoritative paths. It tries to represent the as-it-is state, displaying, for example, usable but unofficial trails or not officially mapped buildings. Humanitarian organizations are often using the project in case of catastrophes due to these benefits.

One of the main benefits compared to other maps or geospatial databases is, besides the high level of detail, the free license<sup>2</sup>. The *Open Database License*, (ODbL), allows noncommercial and commercial applications to use this data freely and even republish it while mentioning the data source. This permission enables the wide usage of the OSM dataset, for example in navigation systems or games without paying license fees. The whole dataset can be accessed by different APIs or downloaded and processed locally. Due to its free character and volume, it is also heavily used in research. Like other crowd-sourced projects, the OSM stores the complete editing history, so changes are always traceable and unwanted changes can be undone.

### 2.2.2 Limitations and Extent of the Dataset

The degree of detail and accuracy in the OSM dataset varies. Densely populated, wealthy regions like Germany are usually mapped accurately while rural areas are lacking details. However, even in third-world countries, cities and larger streets are mapped, allowing basic analyses. Even inside countries like Germany, the degree of detail is differing [dL13], urban regions show details up to the position of streetlights and benches, while in rural areas sometimes information is missing or outdated. While streets and borders are mostly existing, pedestrian paths are in many cases incomplete. However, they are also often nonexistent in other sources. For example, in official sources are publicly accessible but privately created paths often missing, which are partly available in the OSM dataset as locals can submit them.

---

<sup>1</sup><https://blog.openstreetmap.org/2018/03/18/1-million-map-contributors/>

<sup>2</sup><https://www.openstreetmap.org/copyright>

The dataset is offering different kinds of information. Most important is the street and path network, containing additional information, like the legal maximum speed or the kind of surface. Several features can be used to determine the estimated movement speed, for more details see Section 5.4.3. Also, traffic lights or turn restrictions are included. Objects like buildings, stores or institutions are also widely mapped, including information like the opening hours or type of buildings and areas. Also, the public transportation network is included, in parts with further information, like the operation times.

There exist different other projects focusing on spatial data, using the OSM dataset together with own databases or by storing this information directly in the OSM dataset. One example is the *OpenFireMap*<sup>3</sup> which is dedicated to showing fire stations and hydrants from the OSM database. The OSM dataset is much larger than the data which, for example, the tiling server at <http://www.osm.org> is displaying. Meta information or relational data is stored and can be used when accessing the data directly.

However, the database does not store or contain real-time data, like traffic information or closed train crossings. Noteworthy is, that there is a difference between allowed speed and the actual speed, which can be slower, or faster depending on other factors. In some parts, references to other data sources like to Traffic Message Channel (TMC) are present; this way the OSM dataset can be further enriched.

### 2.2.3 Data Quality

Despite an uneven distribution of detail, the data is suitable for routing tasks and analysis. However, as everyone can edit the data, errors like wrongly connected streets or misspelled tags are possible, so it depends on the field of research if the data is stable enough [dL13]. Measurement of data quality is difficult as the ground truth is hard to define, but several studies exist, with varying results depending on investigated region and year. Neumeier [Neu14b] stated 2014, the data quality is good. The amount data has even grown since then. Neis et al. showed that “the OSM dataset for Germany can be considered ‘complete’ in relative comparison to a commercial dataset” [NZZ11]. Barron et al. [BNZ14] also investigated the data quality and created a workflow to obtain a quality measurement.

As the number of details, like traffic lights or speed bumps, is growing, movement speed approximations can be more accurate. The same is true for the number of public amenities or shops which are inserted in the dataset. In Section 4.2 the completeness of POI will be further addressed.

### 2.2.4 Data Format

OpenStreetMap uses three different data objects to store information. An example is given in Listing 2.1. The most basic datatype is the node, containing the spatial position and additional information. Several basic objects, like a tree or a speed bump, are

---

<sup>3</sup><http://www.openfiremap.org/>

mapped using a node. The second one is the way, a list of nodes in a fixed order. It is for example used to define a street or the outline of a building. If the first point is also the last it can represent an area. Ways can also contain meta information as well as the nodes in the way. The third type is the relation. These are objects including any set of objects of the three types and holds objects which are related to each other. One example is the list of different stations of a bus line or the different parts of an administrative boundary.

In contrast to other often used maps, like *Google Maps* or *Bing Maps*, a dump of the whole dataset can be downloaded by everyone, without charges, in different data formats. Usually, so-called *extracts* are downloaded, containing only a subset. These files are filtered spatial, by type of object or have already some other form of pre-processing applied. The complete dump is structured in a single large *planet.osm*<sup>4</sup> weekly updated XML file. The file size is as of May 2018 about 66GB in compressed XML. The OSM-extracts for specific countries can be gained from several other data providers, as everyone is allowed to modify and republish the data. The data is due to its size usually compressed in the *Protocol Buffers* format<sup>5</sup> or in otherwise compressed XML.

---

<sup>4</sup><https://planet.osm.org/>

<sup>5</sup><https://developers.google.com/protocol-buffers/>

```
1 <osm version="0.6" [...]>
2   <node id="1619962907" [...] lat="53.5978840"
      ↪ lon="9.9315196">
3     <tag k="button_operated" v="yes"/>
4     <tag k="crossing" v="traffic_signals"/>
5     <tag k="highway" v="crossing"/>
6     <tag k="tactile_paving" v="no"/>
7   </node>
8   <way id="30077179" [...]>
9     <nd ref="1619962909"/>
10    <nd ref="1619962907"/>
11    [...]
12    <tag k="highway" v="tertiary"/>
13    <tag k="maxspeed" v="50"/>
14    <tag k="name" v="Vogt-Koelln-Strasse"/>
15    <tag k="surface" v="asphalt"/>
16  </way>
17 </osm>
```

Listing 2.1: OpenStreetMap Example Data Structure



The data can be used for populating other databases or can be processed directly, using different programming languages or frameworks. Desktop frameworks often have problems to process the whole *planet.osm* file due to its size. So usually extracts are processed. Tools like *Osmosis*<sup>6</sup> can be used to create extracts.

### 2.2.5 Other Maps

Other platforms for maps are existing, for example *Google Maps* or *Bing Maps*. These are popular for everyday usage, providing a rich set of additional features like high-resolution satellite imagery. The data of most of these alternative maps are licensed from other vendors, or they have their individual crowd-sourcing interfaces. Several systems for mobile phones are using maps based on the OpenStreetMap dataset.

## 2.3 Political Borders

While OSM has information of administrative boundaries, which can be used, the data is hierarchically stored, which leads to some problems while processing, because a single boundary level often does not produce a complete areal coverage. Also, which levels are used in the dataset is differing between different countries. For example, the city of Hamburg is marked as a federal state in Germany, which is correct, but is not holding the annotation for a district-independent city. Alternatively, the postal code areas can be used which provide complete coverage in a finer resolution. In Germany, most administrative borders are labeled with the *Regionalschlüssel*, an official twelve-digit number which denotes for example states, districts and cities.

Alternative data sources are available. For example, the *European Union (EU)* has created the *NUTS (Nomenclature des unités territoriales statistiques)* dataset<sup>7</sup>, which gives for the member states of the EU borders in different hierarchical levels. The data is collected from the different national authorities. This data has a less spatial resolution but remains sufficient for statistical purposes.

## 2.4 Population Data

Population datasets contain information about the population density in different regions of the world. For analyses, it is often relevant to identify how many persons are living in a particular place instead of using only the size of the covered area or the number of streets. Population data is often structured in a grid of a specific resolution, providing the number of persons which are approximately living in this area.

---

<sup>6</sup><https://github.com/openstreetmap/osmosis>

<sup>7</sup><http://ec.europa.eu/eurostat/de/web/nuts>

Various datasets exist, but precise data is often not available or very expensive. The data quality varies from dataset to dataset and also on different regions. For more insights into the data quality see Section 4.1. In Section 5.4.4, it is described how the data can be refined using OpenStreetMap data.

Most population data is obtained from official census data of the different authorities or data from other official sources like the *United Nations*. The information is often refined using satellite images, searching for buildings or using related techniques [LST17].

## 2.5 Humanitarian Statistical Datasets

Different data sources exist, giving information about demographics or other humanitarian data. Most of these datasets are not publicly available in sufficient resolution or precision. In Germany, federal institutions like the *Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR)* give access to basic demographic data like the average age per district. Using the *Regionalschlüssel*, the values can be assigned to the districts which are extracted from the OSM dataset.

## 2.6 Geo Data Processing and Analysis

### 2.6.1 Geographic Information Systems

The term *geographic information system (GIS)* refers to software which allows to show or process geospatial data. Various aspects, like data manipulation, analysis and presentation are often in its focus. Two of the most prominent frameworks with graphical user interfaces are the commercially available *ArcGIS*<sup>8</sup> by *ESRI* and the free *QGIS*<sup>9</sup> by the *Open Source Geospatial Foundation*<sup>10</sup>. Each framework provides several further tools which can be used in their software ecosystem.

### 2.6.2 Storing Geospatial Data

Different approaches for storing large amounts of spatial data are existing. While the XML export format of OpenStreetMap is simple to use, it lacks regarding speed if only a subset of the data is needed, as the whole dataset needs to be read. Applications often make use of a *PostGIS* database<sup>11</sup>. *PostGIS* is a relational database based on the *PostgreSQL* database. It is a project of the *Open Source Geospatial Foundation*. It implements different open standards for geographic data and is compatible with many other systems. Also, other spatial databases exist like *geomesa*<sup>12</sup> for systems in

---

<sup>8</sup><https://www.arcgis.com>

<sup>9</sup><https://www.qgis.org>

<sup>10</sup><https://www.osgeo.org>

<sup>11</sup><https://postgis.net>

<sup>12</sup><http://www.geomesa.org/>

the Big Data ecosystem. For data export of smaller or medium-sized datasets often *GeoJSON* [BDD<sup>+</sup>16] or *shapefiles* [ESR98] are used. These datatypes can be read by different programs and frameworks.

### 2.6.3 Spatial Index

Crucial for fast access to spatial data is a suitable data structure which allows access to the information related to a given position. Different methods exist. In most cases, a tree structure is used to store data.

For example, the *R-tree* [BKSS90] allows fast access and filtering of multidimensional data. Splitting the data into blocks of a fixed size, or more dynamic the quad-tree, which splits the data, if a threshold is reached, in four sub-areas, are alternative approaches. In contrast to them, the R-tree can select dynamic sizes of its leaves. This technique allows fast access even if the data is unevenly distributed. Either a smaller set of data can be returned, or the path to the leaf is shorter. The R-tree is for example used in *PostGIS* or in geospatial programming libraries like *geopandas*<sup>13</sup>.

## 2.7 Emergency Services

### 2.7.1 Special Rights / German Sonder- und Wegerecht

Emergency services have in most countries privileges in emergency situations, in this work we have a look at the situation in Germany. Other countries assumable have similar laws. In Germany, services like police, fire departments or ambulances have privileges in the case of urgency to fulfill their duty. §35 StVO (Sonderrecht) allows to ignore traffic laws like traffic lights or speed limits, and §38 StVO (Wegerecht) allows to instruct other road users to clear the way. These privileges allow fast movement in an emergency regardless of the traffic situation. Still, heavy traffic slows down the speed of the vehicles as often other road user need time to clear the road. In Germany, vehicles on the highway are required to create a corridor for emergency vehicles in heavy traffic even before the movement comes to a full stop. If not performed correctly, the required time to the place of accident can increase dramatically.

Steinvoord [SBB12] investigated the speed of emergency services using GPS. The results showed different speeds for different types of streets and different times of the day. These results allow approximating the movement speed of emergency services better. Rauch and Rauh [RR16] described the movement speeds of emergency vehicles, too.

### 2.7.2 Reaction Times

In Germany most fire departments are voluntary. Usually, only larger cities have a professional fire brigade (“Berufsfeuerwehr”), which is full-time manned. In parts of Germany also voluntary stations exist which are permanent manned [Heg18]. Also, plant

---

<sup>13</sup><http://geopandas.org>

fire brigades exist. Unmanned departments have an extended reaction time until the first vehicles can start from the station, as the firefighters need time to reach the station. The needed time until the first vehicle leaves the station is dependent on the station. Different metrics of quality are required by law, dependent on the region [KK98]. In some cases, only the needed driving time is of concern, in other cases also the reporting time is taken into account.

### 2.7.3 Fire Department Demand Reports

In many cases simple models like the linear distance are used, for example, to create the fire station demand reports (“Feuerwehrbedarfsplanung”)<sup>14</sup>. A report investigating the demand for the fire brigades and estimating if a region can theoretical fulfill their demanded coverage of fires stations. In Schleswig-Holstein, for example, a simple radius of 2.9km(rural) /1.9km(urban) is given which is used to determine if or if not the requested times can be fulfilled.

## Summary

*In this Chapter, the fundamentals of navigation tasks were described, as well as some general words to accessibility analyses. The OpenStreetMap project and its open dataset were described. The general processing of geospatial data was briefly described together with some words to alternative data sources. Finally, some insights into emergency services were given. In the next Chapter, more insights on research and work in this field are given.*

---

<sup>14</sup><https://www.lfs-sh.de/BSBP/Gemeinde/GeoLage.php?ab=1>

## 3 Related Work

*This Chapter gives an overview of existing solutions and related work in research. Section 3.1 gives an overview of existing approaches and solutions creating accessibility analyses and related techniques. In Section 3.2, approaches for refining population data are shown and in Sections 3.3 and 3.4 for creating routable street networks. Finally, in Section 3.5, a brief look at existing frameworks and commercial vendors for these analyses is given.*

### 3.1 Accessibility Analysis

#### 3.1.1 Sampling

An often-used approach is to either select a random set of points or apply points in a pattern around the POI. The needed transportation times from the POI to each point are computed locally or fetched from an online routing API. Assuming a reasonable time estimation by the routing algorithm, the estimation of the selected points is good. However, the times for the surrounding area are not computed, and it is assumed that the whole area is reachable within the same amount of time. The benefit is that existing routing engines can be used, and the computation time is reduced if using a low resolution of points. It is possible to align the centroids of population grids with these points to estimate population coverage.

- Neumeier [Neu16, Neu15, Neu14a, Neu13] investigated the distribution of ambulant nursing services and local food suppliers in Germany. For performing a regional distance analysis, he determined the distance for each centroid of a grid cell to the closest POI. He used the Euclidean distance to determine the closest three POI and performed in the following for these points the routing tasks. The routing was implemented with a *PostGIS* database and the *pgRouting* extension with a fixed movement speed of 50 km/h. For population estimation, they were using the non-public *EWZ250* dataset.
- Jamtsho and Corner [JC14] used a similar approach. They determined the distance from health facilities to population clusters. The population estimation was based on census data and settlement areas of land-use maps. Instead of using road networks for the computation of the shortest path, information of land use and slopes were used.

- At the end of 2017, the project report *Regionale Erreichbarkeitsanalysen* (regional accessibility analyses) by Peter et al. for the metropolitan region Hamburg was published [PG17]. In this project, accessibility analyses for Hamburg and surrounding areas were performed. As the street network, the OpenStreetMap (OSM) dataset was used as well as for parts of the POI. For estimating car movement times, additionally, to the data provided by OSM, a traffic model was used. Movement times were limited to 60 minutes. The goal of the project was to provide the local communities this information to assist in urban planning. They investigated the needed movement times from the center of population grid cells to the given POI, using the *Census 2011* dataset.

### 3.1.2 Isochrones

Isochrones are a way of displaying the reached area within a given time limit. (See Section 2.1.6)

#### Analysis

Instead of sampling, it is possible to determine the area of an isochrone to perform an analysis. In this case, a polygon representing the reached area is created. The transportation times inside the polygon are not quantified. This allows determining a reached population share by intersecting the polygons with the population data. However, this way no information like average times is available.

- In *Regionale Erreichbarkeitsanalysen* [PG17] this strategy was applied to determine the reachability of multiple POI at once. They called these *cumulated indicators*.
- Seip et al. [SBK16] investigated the coverage of ambulances in the federal state of Mecklenburg-Vorpommern, Germany. They came to the result that only around 70% of the population is reachable within 10 minutes by an ambulance. They used the API from the commercially available *ArcGIS Online*, *ArcGIS* with OSM data as well as the free accessible OpenStreetMap based *OpenRouteService* API (see Section 3.5). Population coverage was computed with areal coverage and census data. For each POI, a request was made, and the resulting areas were combined. They also stated that the polygons by the *OpenRouteService* API are not usable for coverage analysis, as they are “too large”. The OSM data and the proprietary data showed similar results as both were using the same polygon creation technique. As only the covered area is used, the result is strongly dependent on the computation of the polygon itself, for example, the way unreachable regions inside the polygon (“holes”) or single streets are handled.
- Rauch and Rauh [RR16] investigated the accessibility of stroke units in Germany. They used OpenStreetMap for routing information and ArcGIS to obtain the isochrones. With the area of the isochrones, the accessed number of persons was determined.

## Creating the Polygon

Different techniques are available for computing these polygons, all of them are approximations of the exact result. A precomputed overlay graph containing only primary roads, or a triangulation can be used to reduce the complexity. Polygons have a trade-off between simplicity, correctness and computation time. The fastest approach is to calculate the convex hull. However, much information is lost there.

- A general approach is to compute the alpha-shapes [EKS83], an approach which is also used in other fields as it determines polygon which is representing a set of points. Unfortunately, this approach has two main flaws. It is computational complex on large sets of points. Also, the  $\alpha$  value has to be estimated. For more details see Section 6.4.1.
- An algorithm by Duckham et al. [DKWG08] also uses triangulation. In this case, the longest exterior edge of the triangulation is removed repeatably until a given threshold is reached, creating a concave hull without holes.
- Jarvis [Jar73] algorithm for a convex hull selects in a first step the minimum point on the x-axis and then looks for the point with the smallest polar angle. A modification of this algorithm as presented by Marciuska and Gamper [MG10] uses the smallest angle at connected vertices in a graph. This way a concave hull can be created. This approach fails if crossings of edges in the street graph exists which are not connected, like bridges or tunnels. This can lead to involuntary loops and create smaller areas as intended. For this approach, a planar graph is mandatory. In contrast, for the routing task, this would be disadvantageous. Postgres *pgRouting* can create nodes whenever two edges intersect<sup>1</sup>, this has also to be taken into concern if using the same graph for routing tasks. The authors stated that further processing is needed if the results consist of unconnected subgraphs.
- This approach can be modified to look at the  $k$  closest points and only take them into account instead of the links of the graph [MS06], this reduces the complexity of the created polygon but loses details. This approach has a lower time-complexity but cannot handle holes in the reached area. Also, it can lead to unconnected areas at paths like highways, with a small number of junctions.
- Another approach is to divide the street network in a set of reached and unreached vertices. Using the edges in the border between the two sets a polygon can be extracted [BBG<sup>+</sup>16a, BBG<sup>+</sup>16b]. Unfortunately, this algorithm reaches his extends, if at one section of the graph no unreachable nodes exist for example, on large lakes or at the coast. Here further pre- or post-processing steps are necessary.

---

<sup>1</sup>[http://docs.pgRouting.org/2.0/en/src/common/doc/functions/node\\_network.html](http://docs.pgRouting.org/2.0/en/src/common/doc/functions/node_network.html)

- Another way is to construct a grid and use their centroids as points to build the polygon. These techniques are used by services like *mapbox*<sup>2</sup> to enable fast displaying of results in a web browser<sup>3</sup>.

### 3.1.3 Further Approaches

- In the work of Glander et al. [GKD10], a model of public transport is used to visualize reachability in Berlin, but they relied on linear distance for the accessible area around of train stations again.
- Also, approaches performing a complete search exists. Hollburg et al. [HSV12] investigated walking accessibility of touristic attractions in the city of Potsdam, based on OpenStreetMap data, creating high-resolution results but using a simple model on a city-wide scale. It appears that such precise approach for accessibility analyses was in research so far not used on a larger scale.

## 3.2 Population Data

For accessibility analyses, population data is needed to determine the number of accessed persons. Here, a fine resolution is desirable. For disaggregation or downscaling of population data, a variety of approaches exist, in many cases using aerial imagery. Scholz et al. [SAM13] used land-cover data and the GEOSTAT 1B [Eur14] population grid. Rauch and Rauh [RR16] used local census data and refined it with building data from OpenStreetMap. By using the size of the buildings, the assumed number of inhabitants per building was computed. Langford [Lan13] discusses different approaches to create more precise population density maps, using information about buildings, bus-stops or streets. Bakillah et al. [BLM<sup>+</sup>14] used a similar method but were focusing on POI like educational buildings. For more details see Section 4.1.

## 3.3 Routable Street Network

OpenStreetMap provides a large amount of data about the street network. The OpenStreetMap dataset is in the first place not directly usable for routing, as the data structure assembles no graph. Different tools exist to convert this data in a graph. The primary task is to split the used ways at each intersection to create a routable format. Another task is to separate the needed elements from the unwanted data. For example, when designing a model for a car, paths must be discarded which are not drivable, like footways or service roads. Some streets are tagged as not publicly accessible therefore not directly usable for routing purposes.

---

<sup>2</sup><https://www.mapbox.com/>

<sup>3</sup><https://github.com/mapbox/mapbox-isochrone>



The tool *osm2pgrouting*<sup>4</sup> can import OSM data into a *PostGIS* database. *ArcGIS OSM Editor*<sup>5</sup> allows the import of the OSM dataset into the *ArcGIS* system. The *OpenSourceRoutingMachine*<sup>6</sup> also has an importing tool for using the OSM dataset. All investigated tools provide configuration files for defining the driving models, default speeds and similar metrics to determine transportation times.

## 3.4 Estimating Movement Speeds

An accurate estimation of driving times has a significant impact on the overall results. While some approaches use a strongly simplified estimate, like a fixed movement speed of 60 km/h, navigation services for standard routing tasks provide a more complex model. These models have on the one side the purpose to give an appropriate time estimation and on the other hand to select the best suitable path for the given profile. In this sense, for bicycles, a slightly longer route on a designated bicycle track is typically preferred over the motorway by the algorithm. Or, for example, turning right preferred over turning left. These optimizations increase the quality of the proposed route, but the assumed driving times are often just based on the type of street as well as the allowed maximum speed and not a physical model of motion. Some vendors have their own statistical datasets giving an estimation how fast the average movement is on this street.

Open source software gives an insight which speeds are typically assumed. The importing tool *osm2pgrouting*, for example, uses the plain speed limits and if no speed limits are given a default value<sup>7</sup>. *Graphhopper*<sup>8</sup> uses a similar approach, where the maximum speed is limited if a bad surface is tagged, additionally. Similar techniques are used by *OSRM*<sup>9</sup>.

## 3.5 Programming Frameworks and APIs

### 3.5.1 Commercial APIs

Different vendors for driving time analysis exist, providing, for example, driving time isochrones. Vendors like *route360*<sup>10</sup>, *mapbox*<sup>11</sup> or *OpenRouteService*<sup>12</sup>[NZ04] giving limited access to a free API. Since 2018, at the *OpenRouteService* also a population measurement of the isochrones is given, based on population grids and the area of the isochrone.

---

<sup>4</sup><https://github.com/pgRouting/osm2pgrouting>

<sup>5</sup><https://github.com/Esri/arcgis-osm-editor>

<sup>6</sup><http://project-osrm.org/>

<sup>7</sup>[https://github.com/pgRouting/osm2pgrouting/blob/master/mapconfig\\_for\\_cars.xml](https://github.com/pgRouting/osm2pgrouting/blob/master/mapconfig_for_cars.xml)

<sup>8</sup><https://github.com/graphhopper/graphhopper/blob/master/core/src/main/java/com/graphhopper/routing/util/CarFlagEncoder.java>

<sup>9</sup><https://github.com/Project-OSRM/osrm-backend/blob/master/profiles/car.lua>

<sup>10</sup><https://www.route360.net/>

<sup>11</sup><https://www.mapbox.com/>

<sup>12</sup><https://openrouteservice.org/>

Some tools like *Graphhopper*<sup>13</sup> are partial open source but require paid access to the API for more complex tasks like isochrones.

### 3.5.2 Programming Libraries

*UrbanSim* developed several python libraries for urban data science. Most similar to the approach developed in this thesis, is *UrbanAccess* [BW17]. It enables accessibility analysis with public transportation data and a pedestrian network based on OSM.

*OSMnx* [Boe17] is another library enabling downloading and processing of OSM data. Also, simple accessibility analyses are possible. Unfortunately, these libraries reach their limits if large maps need to be processed.

### 3.5.3 pgRouting

*pgRouting* is an open source extension to the *PostGIS* database for routing tasks. It can obtain all accessed nodes in a given driving distance. Querying and analyses are done with SQL statements. Therefore, it can be accessed by a wide range of other software and frameworks.

### 3.5.4 GIS Software

*ESRI*<sup>14</sup> provides additional to their *ArcGIS Online* environment also *ArcGIS Desktop*, a commercial platform as well, which allows a wide range of tasks. Using this platform, it is possible to perform accessibility analysis and obtain the isochrones. Also, a delayed start of different POI is possible.

*QGIS* is also providing a variety of tools. For example, an extension to include the *OpenRouteService* or using other APIs of vendors of isochrone services. It is well integrated into other open source software for geospatial data processing and analyses.

### 3.5.5 Open Source Navigation Software

The *Open Source Routing Machine* is the routing engine used by [www.osm.org](http://www.osm.org). It does not provide the computation of isochrones but classic routing tasks. It gives a good and complex example of drive time estimation based on OSM data<sup>15</sup>. Software like this can also be installed in a local instance, allowing a high number of requests [HR16]. Also, further routing engines are existing and providing complex navigation tasks like *Graphhopper*.

---

<sup>13</sup><https://www.graphhopper.com>

<sup>14</sup><https://www.esri.com>

<sup>15</sup><https://github.com/Project-OSRM/osrm-backend/blob/master/docs/profiles.md>

Name	Free	OpenSource	Routing	Isochrones	Cust. models	Population
ORS	o	✓	✓	✓	local	✓
Graphhopper	o	o	✓	✓	local	x
Route360	o	x	✓	✓	x	x
OSRM	✓	✓	✓	x	local	x
Mapbox	o	x	✓	✓	x	x
Google Maps	o	x	✓	x	x	x
iso4app	o	x	x	✓	o	x

Table 3.1: Routing API solutions (extraction)

### 3.5.6 Comparison

Table 3.1 shows some properties of several routing services. The *OpenRouteService (ORS)* as well as the *Open Source Routing Engine* are open source, *Graphhopper* in most parts, too. They allow individual driving models if running a local instance. All of them allow free access to at least some features. Most require a paid account for more complex tasks. In Section 4.3, a more detailed view is given to the isochrone creation by some of these services.

## Summary

*In this Chapter, the related work and frameworks were presented. Existing approaches for accessibility analyses often use sampling or the area of isochrones. These approaches have their flaws in precision or scalability. Isochrones can be created using different algorithms. Population grids can be disaggregated to increase the dataset quality. Different frameworks and API exist for routing tasks. In the next Chapter, a closer look at the datasets and the preprocessing techniques is given.*



## 4 Exploration

*This Chapter gives a look at the different datasets and their preparation. In Section 4.1, a look at population datasets is given and how the data was refined and prepared for this work. In Section 4.2 a look on the quality of POI is given. Section 4.3 gives insights how to visualize obtained results effectively.*

### 4.1 Population Datasets

**Methodology** For gaining gridded population data, several datasets were investigated. As no dataset can be defined as the truth, it is difficult to determine the best dataset objectively. For evaluation, the different datasets were loaded into *QGIS* and given a closer look for inconsistencies. Figure 4.1 shows the different datasets for the city of Hamburg, Germany.

#### Datasets

- The *European population grid from the GEOSTAT 1B project* [Eur14] is containing information for Europe. The *Gridded Population of the World (GPWv4)* [Cen17] has worldwide coverage. Both have a spatial resolution of about  $1\text{km}^2$  grid cell size, containing the number of persons approximately living in this area. Both datasets used official census data as a source and refined it with other sources. The evaluation showed that the GEOSTAT dataset provided more qualitative data in Europe (see Figure 4.1c). This dataset showed realistic estimations regarding places like lakes or industrial areas, showing no or only a few residents.
- The GPWv4 dataset often displayed an unrealistic high number of residents in these places, while showing too few residents in the inner cities. In the case of Hamburg, it exists a constant number of residents on the whole area according to the GPWv4 dataset. The GEOSTAT dataset shows a significantly higher number of residents in the inner city and a lower number for example in the industrial areas in the south. In the United States, the GPWv4 dataset was not showing these flaws and seemed more qualitative.
- For other regions like Africa, Asia or South America the *WorldPop*<sup>1</sup> dataset seemed suitable but was not further evaluated in this work.

---

<sup>1</sup><http://www.worldpop.org.uk>

- Another alternative is the *GHS population grid* [ECCU] dataset, it has a fine resolution, but seems to have misclassified industrial areas as residential areas (see Figure 4.1b). The industrial harbor areas in the south of Hamburg were shown falsely as densely populated as the inner city.
- Fine resolution population data is available for Germany, based on the *Census 2011*<sup>2</sup>. This data is as well as the GEOSTAT dataset INSPIRE-compatible<sup>3</sup>. It has a more precise resolution but shows some outliers in the data. For example, in several cases an unrealistic dense populated area in the center of towns given (see Figure 4.1e). Therefore, the data was not further used.

**Conclusion** For further processing, the GEOSTAT dataset was used because European and German data was used for the following work. To increase the spatial resolution, downscaling can be used; different algorithmic approaches are available. While often satellite imagery is used, in this work, OpenStreetMap data was taken. Figure 4.1d shows a part of the, in this work refined, dataset. Investigation of the OSM dataset shows that at least in Europe in most cities either residential areas or the buildings itself are mapped, often both. In rural areas, frequently only the residential areas are included. While building outlines are often precisely traced, other information like the type of building or the number of levels is sparse. Only in well-mapped areas, the type of buildings is described. For the methods of refinement of the population dataset see Section 5.4.4, for results see Section 7.1.3.

---

<sup>2</sup><https://www.zensus2011.de>

<sup>3</sup>INSPIRE - Infrastructure for Spatial Information in the European Community - An approach to creating a shared data structure in Europe for geospatial data

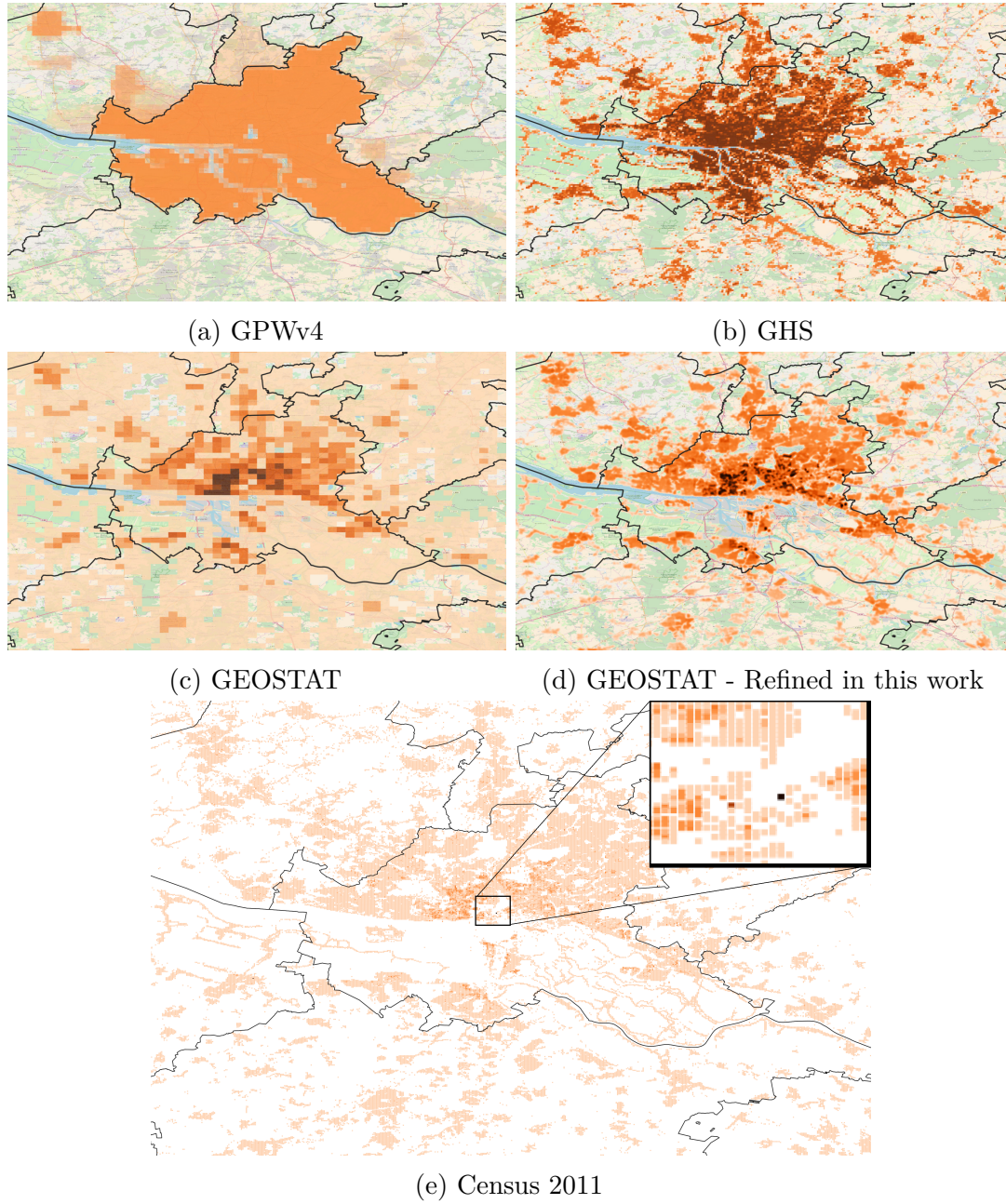


Figure 4.1: Population of Hamburg, according to the investigated population grids.  
*(darker = more populated / different scales)*

## 4.2 Quality of POI in OSM

Essential for the quality of the results is the completeness of the POI as starting points of the analysis. If a company or similar users are performing such an analysis, a complete list of POI often exists. If no such list is present, other sources must be used.

In this work primarily the data from the OSM dataset is used. In several cases, one POI is tagged as multiple entities, if the POI consists of several buildings. For this sake POI, which were too close to each other, were discarded. In many cases the POI from OSM were sufficient, but in other cases, probably many POI were missing. While official buildings were widely mapped, local shops like bakeries or hair stylists seemed incomplete in some regions.

### 4.2.1 Comparing to Official Sources

It is difficult to determine a completeness measurement as other data source to compare against can in most cases also not guarantee completeness and up-to-dateness. Also, many corporations are protecting their data and prevent access to the whole dataset. In other cases, datasets exist which seem official, but are only an extract of the OSM dataset.

One example is the coverage of electric car charging stations, in Germany, such stations need to be registered at the *Bundesnetzagentur (Federal Network Agency)* if build after March 2016<sup>4</sup>. To mention is, that in these cases the presence of an official list may have influenced the data quality in the OSM dataset. However, both datasets do not seem complete as they each contain roughly half of the POI of the other dataset. So for the final evaluation, both datasets will be merged.

### 4.2.2 Comparing to Google Maps

Another approach is to look at the data from *Google Maps* as this is one of the most used digital maps. As the dataset cannot be downloaded and the free API is restricted the evaluation restricts to a smaller area. The *Google Places API* is searched for fire stations in northern Germany. Obtaining a list of stations which could be compared to the location of the stations in the OSM dataset. The fetched data was hardly comparable, as the *Google Maps API* showed several false positives. The API often returns the center of the city or a street in the city instead of the correct position, probably if the system has no precise address. In Section 7.2.2 is shown, that the OSM dataset is not complete for fire stations, but contains most of the POI. *Google Maps* could be used to add stations in suspicious areas manually. An automatic extraction is discouraged or should at least be verified.

While the quality of larger chains seemed sufficient in OSM as well in *Google Maps*, it is more difficult to obtain the quality for stores or restaurants which are not part of a chain. While often attributes like the name is set, tags like the kind of cuisine are missing

---

<sup>4</sup>[https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen\\_Institutionen/HandelundVertrieb/Ladesaeulenkarte/Ladesaeulenkarte\\_node.html](https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_Institutionen/HandelundVertrieb/Ladesaeulenkarte/Ladesaeulenkarte_node.html)



in the OSM data or are not provided in other sources like *Google Maps*. Automated guessing if POI from different datasets are matching is not easily possible. Providers like *Google Maps* are assumable obtaining information by the local *Commercial Registry* or by crawling websites, which can lead to mixing the addresses of the shop and the private address of the owner, or falsely classified locations. So, it is possible that a fire department orchestra is tagged as a fire station.

## 4.3 Visualizing Results

In related work, analyses were performed using isochrones obtained by web APIs. As the area is dependent on the algorithm and the offering service, the services providing a web API were given a closer look.

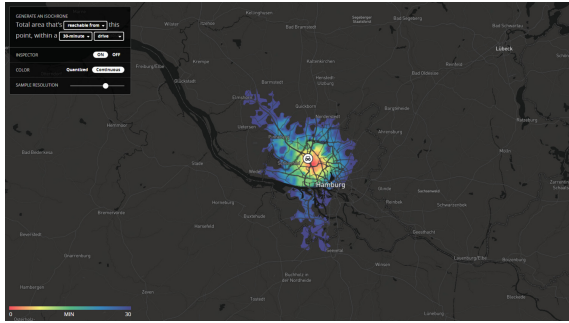
**Methodology** As in Section 2.1.6 described, different approaches exist to create isochrones, creating different polygons. Some online services allow generating isochrones for demonstration purposes. Investigating their results can give insights into the techniques they used. Aside from the polygon creation, the different services make different assumptions about movement speed. For evaluation, some services which allow free isochrone creation were investigated. Services which provide 30-minute isochrones are compared in Figure 4.2 using the same starting point in Hamburg.

### Services

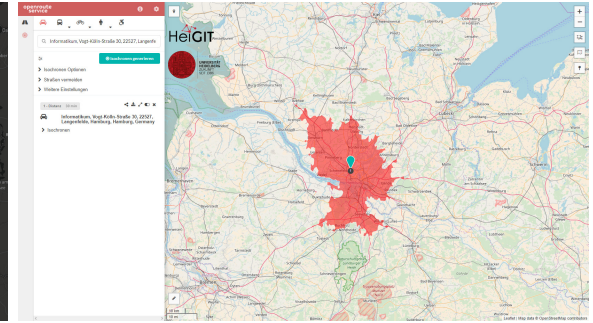
- *Mapbox* is creating the polygon on the client side while the API returns the estimated driving times to a gridded set of points with their matrix API (see Figure 4.2a). For creating the polygons, they rely on a JavaScript implementation of the Conrec algorithm [Bou87]. This leads to partly inaccurate information but enables displaying holes in the isochrone data, representing areas which are not as good reachable as the surrounding areas. Also, this approach is fast and does not need as much computation time as other server-side approaches.
- *TravelTimePlatform*<sup>5</sup> seems to use a similar approach like *Mapbox*, but is directly returning the polygon (see Figure 4.2c).
- *OpenRouteService* [NZ08] is a platform performing different tasks on OpenStreetMap data. Their *Accessibility Analysis Service* [NDZ07] returns directly the polygons (see Figure 4.2b) representing the different intervals in the isochrone. In their original implementation, they used an approach with a sparse mesh point cloud and sliced it to represent the different times. In the current implementation, they seem to use a sparse point cloud and the concave hull algorithm by Duckham (see Section 3.1.2). Resulting is a single polygon without holes.

---

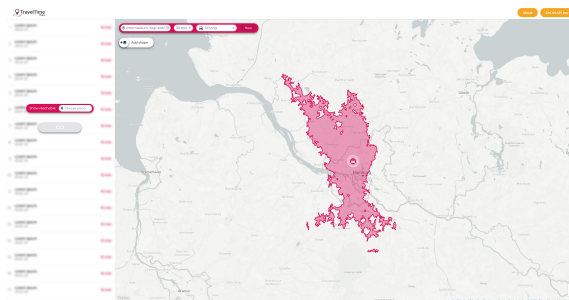
<sup>5</sup><https://app.traveltimeplatform.com/>



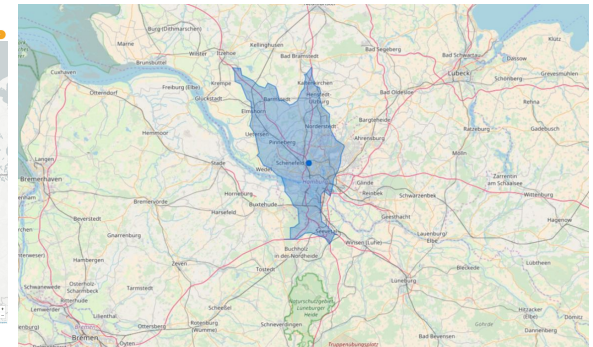
(a) Mapbox



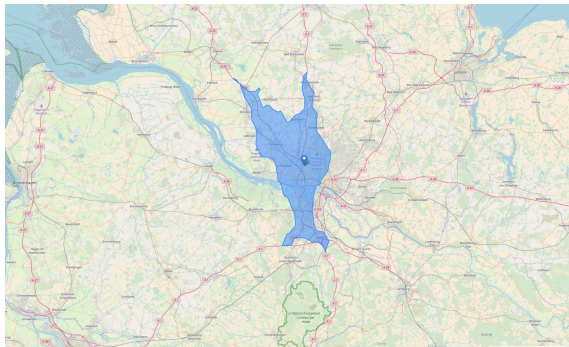
(b) OpenRouteService



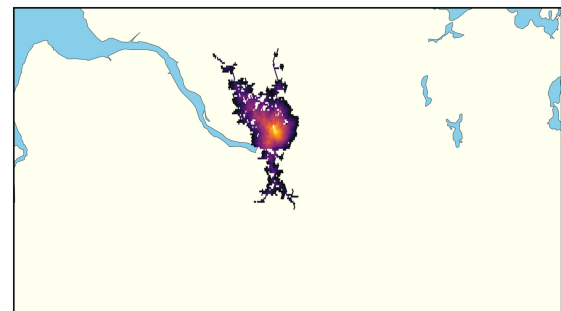
(c) TravelTimePlatform



(d) Iso4App



(e) This work (alpha shapes)



(f) This work (hexbin)

Figure 4.2: Different 30-minutes isochrones

- *Iso4app*<sup>6</sup> allows some basic configuration on creating the isochrones, like the maximum speed. They also directly return a polygon which can be obtained as GeoJSON file or displayed in the browser. The polygons are similar to the *OpenRouteService* and let assume that a similar approach for creating the polygon was used.

**Techniques** For displaying the results in a web browser, different frameworks are available. The services reduce the complexity of the polygon further to allow a smooth presentation in a web browser. If the polygon is too complex, the loading times are too high. Frameworks like *leaflet* allow displaying polygons from *GeoJSON* files. An example from this work is given in Figure 4.2e.

Also using tiles is possible, pre-rendered images which are only loaded for the given zoom level and position. Because many images have to be generated and cached, this is, in general, not suitable for web services providing custom computations in real time. If the number of analyses is limited, using map tiles with maps created by, for example, *matplotlib* or *gnuplot* are showing satisfactory performance. Also, libraries for direct export from Python exist. But *bokeh* and *mpleaflet* show performance flaws on large result sets.

For in-detail analyses or exact displaying of results, polygons are often not precise enough. Advantageous can be displaying the streets or buildings with its corresponding results. This can be applied to tiling servers as the complexity for displaying this information, vectorized and unfiltered, is too high.

**Conclusion** The shape of the polygon is dependent on the used algorithm, they all only approximate the result. Polygons are not well suitable for precise analyses, but for giving a fast overview. If using a sampling approach, the *Conrec* approach, otherwise a *Jarvis* or *Duckham* based approach shows good results. The alpha-shapes approach, which is used in this work, would show comparable results but has its benefits, as it can create holes in the polygon. The alpha-shapes approach seems not to be used on the web API, probably because the computation time is too high.

Creating *GeoJSON* files with only the needed degree of details and displaying via *leaflet* showed the best results for fast result displaying in a web browser, *Iso4app* also seems to use this approach. For general plotting, libraries as *matplotlib* can be used, creating scatter or hexbin plots (Figure 4.2f).

## Summary

*In this Chapter, different population datasets were evaluated and described in more detail. The GEOSTAT dataset showed the best results. The quality of POI was further evaluated. Investigating different web services, the creation methods of isochrones were discussed. In the next Chapter, the design of the created workflow is described.*

---

<sup>6</sup><https://www.iso4app.net/>



# 5 Design

*In this Chapter, the created tool-chain is described. In Section 5.1, a short overview is given. In Section 5.2, different strategies are presented, and it is shown which single tasks have to be solved to achieve the goals of this thesis. Following, in Sections 5.3 to 5.5, the different components of the tool-chain are described in more detail.*

## 5.1 Overview

Based on the goals of this thesis, there are several requirements for the software:

**Accuracy** To perform analyses, using small scaled aspects and receiving a high accuracy, a high resolution of the results is necessary. The goal is to allow to obtain results for individual streets or buildings. On top of a high resolution of the results, the navigation task needs to be done on a proper transportation model. Depending on the scenario it may be necessary that different POI are modeled with different delayed starting times. For accurate estimation of transportation times, a movement model needs to be developed determining the time which is required to move from one place to another.

**Performance** Performing an independent navigation task for each node in the street network would need too much computation time, and by using sampling, the results may become unrepresentative on a smaller scale. Approaches in related work try to accomplish this by using the area of isochrones. While being fast, this approach lacks precision and, if obtaining the isochrones from a web API, for large analyses the rate-limits of the APIs may become a problem. Relying on an API shows further predicaments, as lacking customizability or scalability.

**Flexibility** The movement model should be adaptable to the particular scenario, as unique characteristics apply. For special cases like emergency services or agricultural vehicles, designated transport models need to be specified. For further analyses, results, based on different sets of POI and movement models, need to be combinable.

**Extensibility** Using the obtained data, more knowledge can be obtained by further processing of the results. Allowing to extend the workflow of the accessibility analyses, for example, allowing to combine different results or perform optimization tasks.

### 5.1.1 Workflow

The workflow designed within this thesis, allows a high degree of customization and result quality. For each experiment, the following steps were individually adjusted and executed:

- Exploring which data sets are needed and can be used for this experiment
- Preparing data sets for a sufficient spatial resolution
- Performing navigation tasks and storing value for each node
- Scaling this approach, so a high number of POI can be used by spatial portioning the tasks
- Visualizing and analyzing the results

Based on the results, further challenges can be addressed, like the combination of different analyses as well as location optimization. As it turned out during the evaluation, most steps and analyses in this thesis can be processed on a conventional desktop PC, even for analyzing country-sized maps and assumable up to continental size. As many modules are scalable, it is possible to reduce computation time by using a large-scale setup like computation clusters.

In the following section, the different use-cases are described, and an overview which tasks need to be performed is given. After that, the single tasks are given a more detailed description.

## 5.2 Analyzed Use-Cases

To give an overview of the possibilities and needed tasks to perform, different use-cases are sketched, and their workflow described. Results and parameters for the here outlined use-cases will be presented in Chapter 7.

A general overview of the explored experiments is given in Figure 5.1. Using an accessibility analysis on the prepared data, the *service coverage* can be determined. This data can be further used for *statistical analyses*. Alternatively, several analyses can be combined to create individual *quality maps*, which again can be used to determine coverage statistics. This data can also be used to find *optimal locations* for new facilities.

### 5.2.1 Service Coverage

**Description** The first presented workflow is to determine the coverage of a service like fire departments or stores of a specific chain. To determine the coverage, in many cases it is not relevant which exact service location is the closest one. For most purposes, the different POI are substitutable. For example, which particular fire department reaches the place of emergency is irrelevant; in this case, we assume they are equivalent to maintain coverage. For now, we ignore capacities, assuming that the capacity of the location corresponds to the size of the population in their catchment area. The two cases,

driving from a set of places to the population and driving to the POI, are describing the identical problem if taking care of the driving direction in, for example, one-way roads. An overview of the performed tasks to obtain service coverage is shown in Figure 5.2. The pre- and post-processing and tasks mentioned in the figure are described in the following and given a detailed view in the corresponding section. Note that several steps are optional and depend on the type of analysis performed.

**Data Preparation** The data has to be prepared, as described in Section 5.4, most of the data can be stored and used for different analyses. In a first example application, we are focusing on the coverage of emergency services like *fire departments*. The network graph is created depending on the needed transportation model. In this case, with fire trucks in mind, trying to approximate their unique movement characteristics. For details, see Section 5.4.3.

As far as possible, the obtained stations are differentiated between voluntary and professional departments, giving voluntary departments a larger starting delay due to additional required time for the firefighters to arrive at the station. In Germany, most stations are voluntary, so if no further information is given when obtaining the locations, it is assumable that the given station is voluntary. Usually, only larger cities have permanently manned fire departments.

**Analyses Computation** Performing an accessibility analysis determines the estimated driving time to reach each node in the network from any fire station. For each node, the needed times from the  $n$  fastest origins of the navigation task are stored. See Section 5.5, for insights of performing the analyses and Section 7.2 for results.

**Population Coverage** This step allows determining the number of people which are covered by the POI and can be used for various subsequent analyses. In this case, a population dataset has to be consulted. The data is refined by using downscaling (see

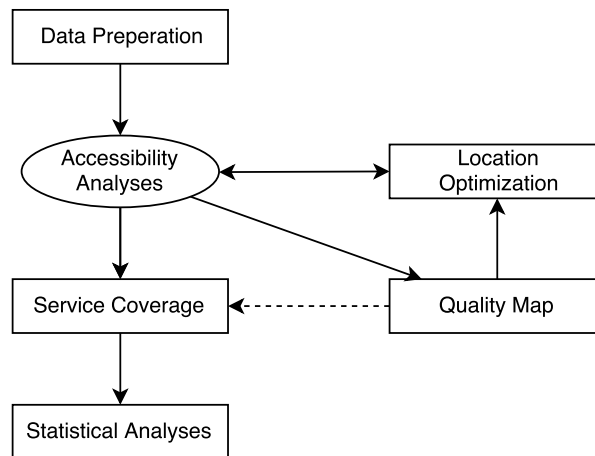


Figure 5.1: Overview of the presented scenarios

Section 5.4.4). Each node in the network is assigned an estimated number of inhabitants. This can be used to create more representative maps as the results can be weighted per person. One example is using a hexagonal bin plot<sup>1</sup>. For more details see Section 5.6.1.

**Regional Coverage** This step allows determining the share of the population which is covered for each district or other administrative boundaries. For later evaluation, each vertex in the network is assigned to the corresponding district (see Section 5.4.2), for example, the local communities, states or postal code areas. The average value or the percentage of coverage per region can be shown when combining this data with the results. Alternatively, drive time polygons created by previous runs can be used.

**Highway Coverage** This step allows determining the length of the highway which is covered by the investigated set of POI. If the coverage of the highway needs to be determined, a dataset containing all nodes of the graph related to highways is used. While looking on each edge, its lengths and the path costs to both vertices, coverage can be determined. In contrast to other approaches, like using the area of a polygon, this allows more precise results as highways are often lacking access from nearby streets and turning points, increasing the needed driving time. This way, both directions can be investigated separately. Highways are of particular interest for emergency services and services focusing on passenger traffic. For more details see Section 5.6.2.

**Result per POI** This step provides statistics per individual POI. For this, if an independent navigation task for each POI needs to be performed. (If using an optimized approach this is not possible, see Section 5.5.2). The number of accessed persons, in the

<sup>1</sup>[https://matplotlib.org/examples/pylab\\_examples/hexbin\\_demo.html](https://matplotlib.org/examples/pylab_examples/hexbin_demo.html)

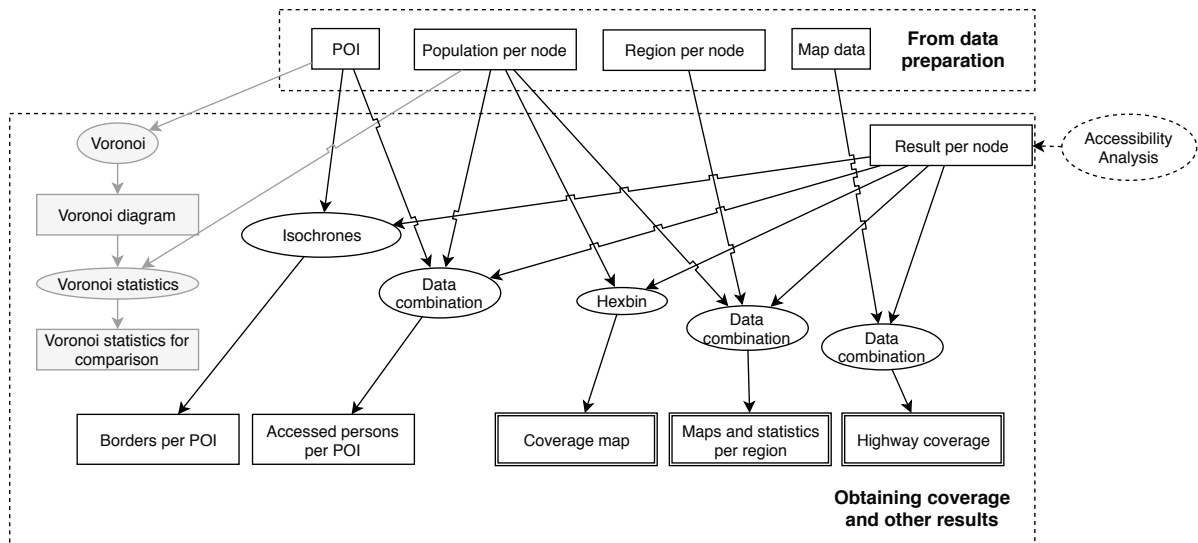


Figure 5.2: Workflow for determining service coverage



given time and per POI, can be determined as well as a polygon enclosing the reached area or the average time (see Section 5.6.3). Additionally, as every time the origin of the fastest connection is determined, it can be visualized which POI is the closest to each node in the graph. This allows more precise areas of responsibility comparing to Voronoi diagrams, which can be computed for comparison.

## 5.2.2 Quality Map: Individual Combination of Analyses

**Description** Determining the best place to live or to settle down is dependent on a variety of factors. For example, the access to certain POI or in some cases the absence of these. This is true for individuals, looking for a new home or a place for holidays, as well as for store owners determining the best place to locate a new store location. For now, we are focusing on individuals. While some general assumptions could be made to determine the estimated living quality at a given place, most factors depend on an individual ranking. In this scenario, we want to visualize a map that can be parameterized by users, giving soft and hard conditions concerning reachability of POI. An overview is given in Figure 5.3. For this, we are combining (see Section 5.6.4) different results from Section 5.2.1. For some example results, see Section 7.3.

**Computation** Different analyses are defined, giving a maximum time range and a weight describing how much it should be respected in the final result. Also, it is possible to demand the absence of a POI. The determined score gives an abstract quality measurement for each node in the graph.

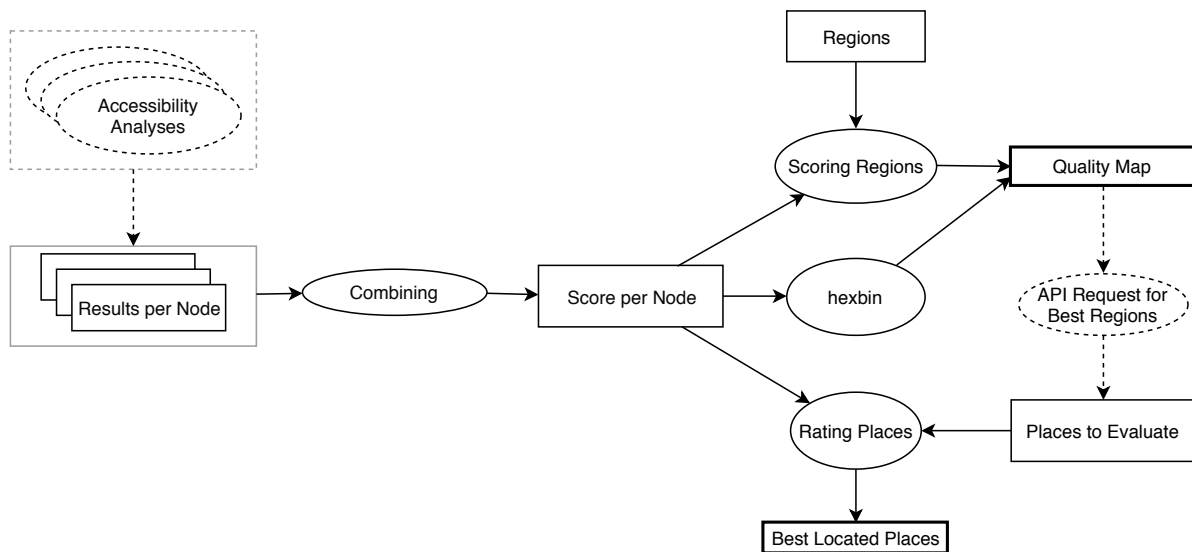


Figure 5.3: Creating a *Quality Map*

If a result to a set of POI already exists, a cached result can be used and is only needed to be check for the given constraints, while results for individual POI, like the location of the employer, is computed. This allows fast access to results so that it could be used for end-user applications. Plotting this data on a map the average score is shown and can be determined for each region, as well as administrative boundaries.

**Further Analyses** With this measurement of quality, for example, candidates in the real estate market can be selected. By using an API, available offers can be obtained. For evaluation, we are again using POI from the OSM dataset. If accessing the places to evaluate by an API, probably not all offers are required to obtain. In an intermediate step, a search request can be performed on each region which reaches a given threshold. Alternatively, the center of each region can be used, and these results evaluated manually. As each place to evaluate can be assigned to a node in the graph and therefore to a score from the quality map, it is also possible to determine the scores for a set of places to evaluate.

### 5.2.3 Location Optimization: Determining a Good Settlement Point to Increase Coverage

**Description** While individuals may be pleased if the set of criteria are fulfilled, for store owners or service providing facilities, it is also crucial to pick a location which has many persons in the catchment area. If other store locations already exist, it is possible to determine which new location is most efficiently increasing the overall coverage. Plain selecting the worst covered areas would not necessarily create the best location as this area may be hardly inhabited. However, it can give a good first estimation.

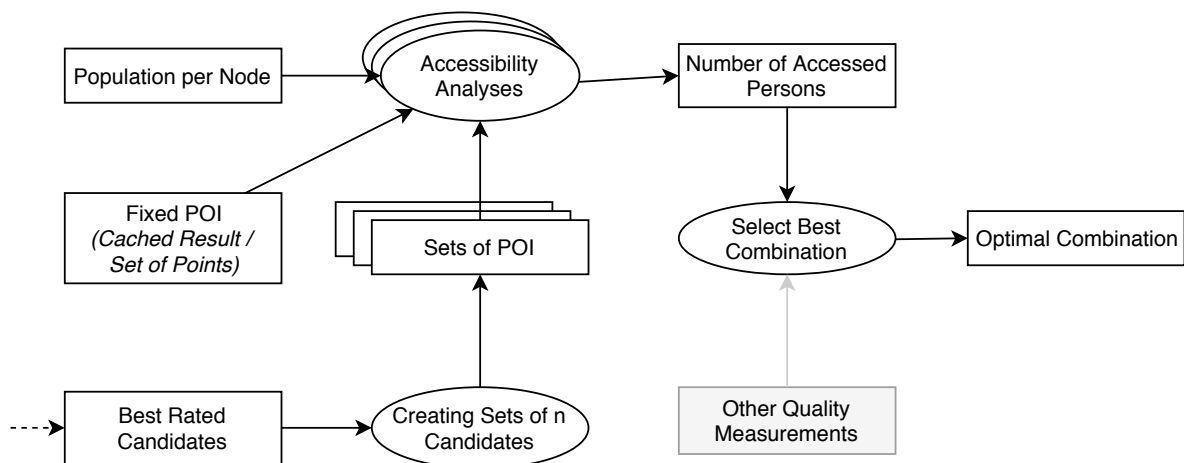


Figure 5.4: Optimization of facility placement

**Computation** An overview of the workflow is given in Figure 5.4. In a first step, a quality map like in Section 5.2.2 is created. By extracting the candidates with the best rating, possible locations are selected. After this, the best-rated candidates are grouped into sets of POI which are analyzed together. If only one new location is required, the sets would only contain one POI each. In a next step, by using a coverage analysis like in Section 5.2.1, the accessed number of persons in their catchment area is determined. The set of new POI can be combined with already existing locations to estimate the coverage improvement. Aside from the accessed number of persons in a given time or the average needed time, individual characteristics can be applied. If already a large number of locations of the store exist, a pre-computed result can be used and updated with a single or multiple possible locations and the increase of coverage determined. This allows fast iterations to compare different solutions finding the best. Using a large setup, with this strategy the best store locations out of thousands of candidates can be selected, allowing optimized access to customers and logistics.

**Remarks** In this work, a simple optimization approach was used, by brute-forcing all sets of possible POI were evaluated. This approach benefits from reducing the number of candidates when removing locations in well-covered areas or based on other metrics in the first evaluation step. For optimal results, a broad set of candidates should be used, as selecting the worst covered point may not increase the overall coverage if the area is not inhabited. However, using large candidate sets can become unpractical. For these cases, optimization techniques that further decrease the number of candidates, using the linear distance, or other measurements like the presence of competitors should be applied. Aside from the number of accessed persons, other ratings, like the score from the quality map or the real estate rates can be used to select the best-suited locations in the second step.

## 5.3 Data Handling

**Architecture** The software is split into different modules which can be launched together or separately. Using standalone parts allows fast debugging and development. The results of these modules are used for further processing. Some intermediate results need to be newly computed for each run, while other parts can be cached as long as the map data is not altered. Different possibilities exist for storing the datasets, intermediate data and results. Spatial databases or files are valid alternatives.

**Spatial Databases** Often used are spatial databases, systems for handling spatial data built on top of other database systems. One of the main benefits of spatial databases is that data can be accessed precisely for the requested area. This enables to obtain data which is needed for local computation. If the main memory is limited, the benefit is that data can be precisely loaded once required, preventing to access unnecessarily large areas. Also, databases like *PostGIS* guarantee characteristic properties of relational databases, which are essential for productive instances, when the software, for example,

is interrupted while writing data. However, especially while storing, maintaining the database indexes and validity is causing an overhead of space and computation time. If the database is located on a dedicated system, loading times can be a bottleneck, too. For presenting the data or analyses of smaller subsets, a database has its benefits as it allows fast access to a small set of the data or can efficiently perform other index accelerated tasks. For large-scaled distributed setups, systems like *GeoMesa*<sup>2</sup> could be used. Another drawback of databases is their often non-trivial deployment process and requirements on the runtime environment.

**Files** A benefit of using a simpler approach using plain files on the file system is that it avoids the bottleneck of a database system like *PostGIS* which slows down especially while storing (intermediate) results. Furthermore, data can be processed on different compute nodes using a distributed or shared file-system without the need of maintaining a consistent distributed database or big data system. For faster processing, the input data, as well as large results, can be stored in multiple files, one file per region and loaded on demand in the memory. This enables parallelization and reduced memory consumption at runtime. If it is necessary to obtain the closest elements to a given spatial coordinate, a spatial index can be used as in a spatial database; if the data is needed for multiple analyses, this increased the performance noteworthy.

**Decision** Due to the flexibility of files, parallelism and simplicity the data was stored in files instead of a database. Furthermore, this reduced additional efforts for maintaining a database and enabled fast prototyping and debugging.

As some intermediate results need to be stored because they are too large to be in memory on desktop systems, the overhead for saving this data in a database would be too high, especially as the intermediate results are discarded after processing. On the other side, for persistent data and fast result presentation, *PostGIS* would give an appropriate alternative.

Using spatially separated files as well as spatial indexes, fast access using files was possible without additional overhead.

## 5.4 Data Preparation

The primary dataset is an extract of the OpenStreetMap database. The data preparation is sketched in Figure 5.5. This data is, due to its size, in a first step split into different smaller files to speed up later steps. Extracting and splitting up the data is done with the tool *Osmosis*<sup>3</sup>. An extract containing only the potential POI is created. Further, an extract with only street data is split into different areas to reduce processing time when creating the graph. This also allows to parallel process this dataset. After splitting up the dataset with *Osmosis* the following tasks can process the four smaller datasets (POI, streets, borders, residential areas).

---

<sup>2</sup><http://www.geomesa.org/>

<sup>3</sup><https://wiki.openstreetmap.org/wiki/Osmosis>

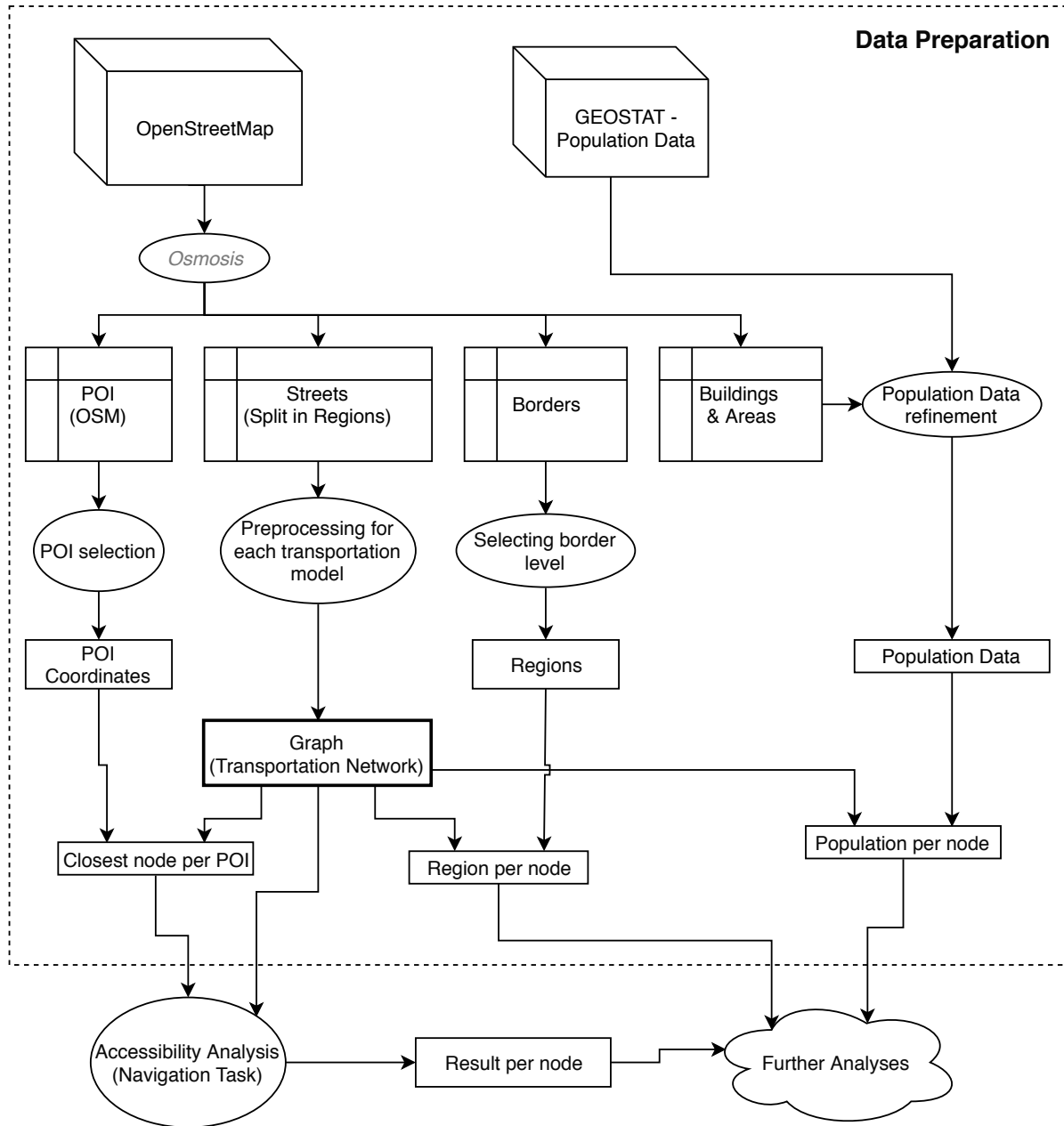


Figure 5.5: Steps for data preparation

### 5.4.1 Points of Interest

Points of interest (POI), like supermarkets or fire stations, are defining the starting points of the accessibility analyses.

**Obtaining the POI** Positional data for POI can be accessed from different data sources. If official sources are available, it is preferable to use them. However, also the OpenStreetMap dataset gives an excellent opportunity to obtain a broad set of objects. The quality of this data is crucial. For notes on the quality, see Section 4.2. If no coordinates but addresses are given, it is possible to obtain coordinates using a geocoding service. Also available are open POI databases for navigation systems. However, it is difficult to estimate how accurate this data is. More information on the implementation is given in Section 6.2.2.

The OSM data is searched for given aspects to obtain the points of interest. Depending on the objects to investigate the amenity, shop and the name field are used. The range of objects can be general, like obtaining all supermarkets, or be very specific if combining with information like the name, or their opening hours. Additionally, based on their tags, start delays for each POI can be set. Giving the opportunity to treat each POI independently.

**Graph Assignment** After obtaining the objects, the closest node in the street graph is selected and stored as a starting point for the navigation task. The path to the next node in the network is not represented, as it is hard to define the real-world path from the building to the street network. When no parking lot is needed, like for delivery services or emergency services, the time to the nearest position on the street is sufficient. Further, if designated parking areas exist, they are in most cases close to the POI.

When comparing the accessibility analysis to other techniques using the linear distance, either the original position or the position in the street network can be used.

### 5.4.2 Political Boundaries

Different data can be used for obtaining borders (see Section 2.3). As in the OpenStreetMap dataset borders are included, these can be used for structuring the results. Administrative borders are represented as a set of ways in a relation which contains relevant information like the name or the border level. The border level defines the hierarchical structure of borders. Alternatively, the postal code districts can be used. All polygons with an outer tag are creating the corresponding multi-polygon, the ones with an inner-tag are creating the holes, like independent cities inside of another district.

For this work mostly, the data from the OpenStreetMap dataset with border level 6 was used, this data is mostly equivalent to the *NUTS-3* data. The federal states of Hamburg and Berlin were manually added, as they occur as level 4 borders in the dataset. This data can also be obtained from other sources, as any set of polygons in a standardized format can be processed. It is also possible to use polygons created by an earlier run.

After the borders were extracted, for each polygon the contained nodes of the street network are selected. With these nodes the information of population and the navigation task result is accessible.

### 5.4.3 Movement Models

#### Approach

Different scenarios can use different movement models. A model describes which streets (edges in the graph) can be used and how fast one can move along the edges.

Aside from the topology, the weights of the network graph have to be estimated. For each connection, the weight is the estimated time to move on this edge. The model for estimating the time can be simple or complex, up to a simulation with a realistic physical model as, for example, used in traffic simulations. In this work, we reduce the complexity and use more simple measurements, as often used for navigation systems as described below.

All computations for determining the weight of the edges are done while extracting the graph data from the OSM data. This saves computation time and needed memory while performing the accessibility analysis. It has to be done only once per model, or when the street network needs to be updated. Alternatively, all aspects can be stored, and the weights are computed at runtime, this would allow dynamic customization of the transportation model. OpenStreetMap provides several data fields which can be used for estimation. Most important is the given speed limit and the kind of road. The parameters were chosen similar as in already existing open source routing solutions. Other aspects like traffic signals are also used, depending on the given scenario. In the following, the different models are briefly described.

#### Standard Car Model

The standard car model is designed for car usage of the population, driving to given POI. It shall equal the average speed of regular cars. The assumed speed depends mostly on the speed limit. Also, it creates a short delay passing a light signal. For a first estimation, the values similar as used in the default *Graphhopper* configuration are used<sup>4</sup>. Especially in areas where no speed limit is set, as on German highways, time estimations can be different to the real driving speed of an individual. Also, not all ways in the OSM dataset are drivable. Various kinds of ways which are not part of the street network had to be discarded. However, some cases like parking aisles were crucial keeping, as they showed in some parts the only connection between two other kinds of streets. Removing these ways could lead to cases where the navigation task stops after a few meters as no further streets were reachable. An approach to fix this could be to discard small subgraphs which are not connected to the rest of the graph. However, in this case, small islands

---

<sup>4</sup><https://github.com/graphhopper/graphhopper/blob/master/core/src/main/java/com/graphhopper/routing/util/CarFlagEncoder.java>

or other separated areas need to be handled. To create a robust street network, many edge-cases need to be addressed as data in the dataset in some cases was incomplete or showed small errors.

In this model, the current traffic situation is not considered; instead, the given speeds and delay at traffic lights are assuming a situation with moderate traffic. In Section 7.1.1, the estimated drive times of the model is compared to estimates produced by the *OpenSourceRoutingMachine*, as well as *Google Maps*.

## Emergency Services Model

Emergency services have other physical aspects as regular cars. Firetrucks or ambulances are heavier but have special traffic rights. Thus, the maximum speed is reduced particularly for ambulances carrying a patient while making a turn or passing speedbumps must be considered. As special vehicles have the right to cross red lights, traffic signals are not taken into concern. This typically results in higher average speeds, for example, in living streets.

For this work, a model based on measured speeds is used (see Section 2.7). Each type of street was given average speed. As while measuring only the type of road was stored, delays concerning turning or traffic are already respected in this data. Speeds in specific cases like pedestrian areas or passing one-way streets in the opposite direction are difficult to estimate. In this work, a speed of 10 km/h is assumed for these special cases. Other access restrictions for cars were ignored, assuming that streets are drivable under special rights. The goal was to estimate an average needed time, depending on the scenario worst-case runtime could also be of interest, with for example reduced speed on highways or waiting times at train crossings.

## Walking Model

The walking model contains more paths, but no highways or similar inaccessible types of roads. The needed time is strongly dependent on the individual walking speed and the time to cross streets. As appropriate data concerning street crossings is not available, a constant walking speed of 5 km/h is assumed, as this seems to be the preferred average walking speed [BBHK06].

## Public Transport

As in most cases, no time schedules for public transportation are available in the OSM dataset, using this source, the transportation times can only be assumed vaguely. In a more advanced step, data using the API of the corresponding public transportation network could be used. However, this requires an individual solution for each local transportation provider. For this work, public transportation was not taken into concern due to its high complexity. It is of interest for further work.



## Model Storage

In any case, each way from the OSM dataset is split at each node. Additionally, segments which are only relevant for maintaining the geometry, like curves, can be combined. The latter approach only stores the position of junctions where separate ways meet. The needed times are aggregated. This saves space and computation time but decreases the precision of POI and population, which are then projected to the nearest network node.

The graph is stored in separate files depending on their spatial position. This way, other modules can access the data selectively for a given region. The data is split into information about the *nodes*, containing *id* and *coordinates*, and the edges containing two *node-ids* as well as the *weight*. Additionally, all ways of a given particular kind, for example, highways, can be stored separately, together with their length. By providing the lengths of the streets, it is possible to determine the coverage of the streets itself, as otherwise the information of the distance would be lost.

### 5.4.4 Refining Population Data

**Overview** The available population datasets had no sufficient spatial resolution for the wanted degree of analysis, since for every node an assumed number of inhabitants needs to be determined. The idea was to refine the data with the information about buildings and residential areas which is provided by the OpenStreetMap dataset. An overview is given in Figure 5.6. For the population in Europe, the *GEOSTAT* dataset was used, as it showed the best data quality of the investigated datasets (see Section 4.1).

**Residential Areas** All residential buildings and buildings with no specific tags as well as all residential areas are extracted from the OSM dataset. This results in a massive set of polygons where we assume people are living. As in rural regions often buildings are missing and in other regions, only the buildings were mapped, and no residential areas are specified, these data sources were combined. Polygons which are inside of other polygons, like buildings in a residential area were ignored to reduce the number of objects for succeeding usage. Areas which are neither mapped as a building or residential area are assumed uninhabited.

**Disaggregation** For each cell in the population grid, now every person in this area is set to a randomized location and checked if it is inside a polygon using spatial indexes for the bounding box and *raycasting* for exact determination. This approach is time-consuming but has the benefit that each person has, in the following, a discrete position. An alternative approach is to use the size of the polygon to obtain information on how many persons probably live in this area or building. While this approach is faster to compute, on large polygons it is difficult to match persons to the network nodes. Alternatively, to reduce the computation time only every tenth person can be sampled, decreasing the dataset size and computation time.

By using this sampling approach, the population is assigned to the houses or residential areas, which gives a high spatial resolution, while it is just a model of the distribution, it allows a more precise assignment to single streets. In cases where no polygons exist or points were not hitting the polygons in a given number of tries, the data is sampled randomly. In this worst case, if in a region no polygons are set, the population data is equally distributed over the area of the grid, which equals the data quality of the original population data.

**Remarks** This approach has its limits, as missing tags can falsely classify an industrial building as a residential place. Also, with a missing number of stories of a building, a skyscraper in a region of mostly smaller buildings can be inaccurate. With growing data quality this data could be used, but also an approach using only the given outlines improves the population data. By using a finer population grid, the error can be minimized. Example results can be seen in Section 7.1.3.

**Graph Assignment** Further, the newly set persons are mapped to the nearest node in the network graph. As exits and entrances to residential buildings are rarely set, this data is not used. As the population is mapped to the nodes, the persons are assumed to be on the nearest corresponding junction or curve of the street. The resolution could be improved by adding vertices in the graph. However, this would increase the complexity of the navigation task for only a small benefit, as population positions are just vaguely approximated.

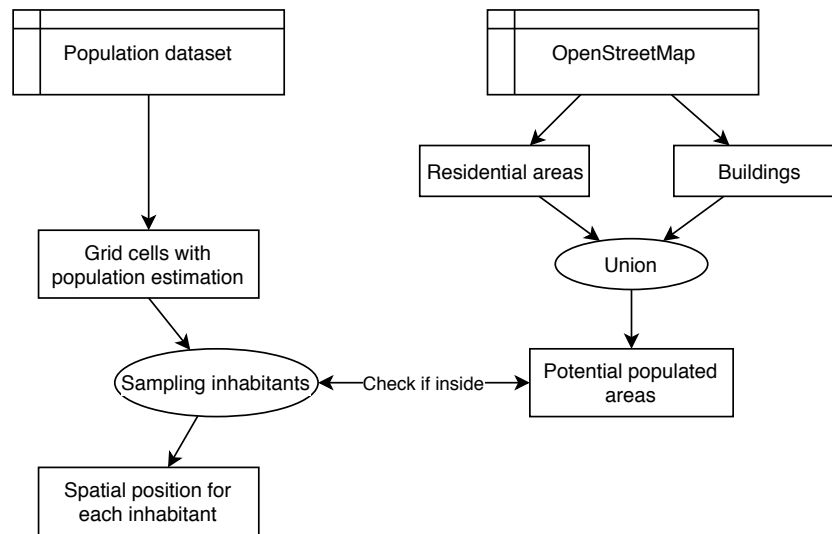


Figure 5.6: Steps for population disaggregation

## 5.5 Accessibility Analysis - Navigation Task

### 5.5.1 Complete Analysis

**Overview** For the navigation task of the accessibility analysis, different approaches as shown in Section 3.1 are possible. In contrast to other approaches, for each point of interest, one single navigation task is started and the needed time for every node in the graph is stored. This leads to more precise results as using single navigation task for each grid-cell or working with polygons.

**Navigation Task** The algorithm needs to compute from the starting point, the graph distance (transportation time) to each other point in the network. The Dijkstra algorithm [Dij59] allows this uninformed search. For details of the implementation see Section 6.3. As selecting the minimum distance object is a central feature in this Dijkstra's algorithm a minimum heap can reduce the computation time significantly. Limitations, like the absence of negative weights in the network in the Dijkstra algorithm, are not of concern, as no negative times are possible in this context.

A heuristic extension as the  $a^*$  algorithm is not used, as the movement time to every reachable point is needed. For systems like navigation systems looking for a single route, this would cause a significant speedup. The same is true for other optimization techniques which reduce the number of nodes and edges in the graph. Techniques using overlay graphs or other hierarchical methods can lead to suboptimal results and forbid obtaining exact results for areas which would be skipped.

Usually, a shortest-path query is stopped if the best path between start and goal point is found. For Dijkstra, the termination criterium is the first expansion to the goal node. In this work, the needed times of all nodes are of interest. Therefore, the termination criterion is not reaching a given point, but a virtually driven time limit, or if no further nodes are in the graph to explore. As the algorithm is performing a wide search, the subgraph of reached nodes is growing uniformly concerning the transportation time, and the task can be stopped if the first node outside of the virtual driven time limit is reached. For each node in the graph, the best route and its corresponding needed time are found.

For each point of interest, an individual time can be set, to define an individual starting delay. This is dependent on the scenario, to set for example reaction times of different emergency services. In this case, the node in the graph, which represents the POI, is initialized with the delay when starting the computation, which allows other POI with less delay to have an extended coverage area.

**Storing Results** As the navigation task is processed for each POI, when storing the results, the transportation time for each point can be compared to existing results and updated if a better result was achieved. Additionally, for each starting point, the reached subgraph can be stored, split into different time ranges, for further analyses. The routes are not stored, as they are of no particular interest in this context.

When combining the results of different POI, not only the best but the second best and more results can be collected. A configurable number  $n$  defines for each vertex how many best results (from different POI) are stored and respectively updated. This can be useful for evaluation of emergency situations where the forces of several stations are needed, or if it is of interest that a given number of equal POI are reachable in a given time. For an example, see Section 7.2.

### 5.5.2 Fast Analysis

If only the overall result is needed, the fastest time for each node, multiple starting points can be set when initializing the algorithm. In this case from all POI, the search area will be expanded simultaneously, stopping if reaching the exploration frontier from another POI. It equals a connection between sources with a connection weight of zero, which would be expanded in the first steps. This computation of multiple POI allows much faster processing and needs less memory if several POI are close to each other as all POI can be handled in one task, see Section 6.3 for implementation. Due to the early stopping when reaching an already, by another source point, explored area, the reached subgraph of a POI is not complete and plotting individual isochrones per POI is restricted to the area where the given POI is the fastest one.

### 5.5.3 Parallelization

As the investigated area is large, parallelization and modularization were necessary to reduce time consumption. As the whole network graph is divided into different areas, only a fraction of it is loaded into memory per process. For each area which contains at least one POI, the network graph is loaded, together with areas in a given radius around this tile. The radius depends on the estimated area which is needed to perform the routing tasks. This can be done static when initiating the process or dynamic while runtime to reduce memory consumption. While dynamic loading, if the search reaches a node outside of the loaded area, the graph would be extended, and the search continued. Otherwise, the graph has no knowledge about the nodes in the other area and cannot expand these nodes. In this work the simpler static approach was used, preventing loading of new parts for the graph at runtime, since memory consumption was not an issue and a sufficiently large area could be loaded at process initialization.

If each process computed the results for their POI, the data is written again to disk for each region. If there is already data for this region from another process, the data is read and compared to the own results before writing. As this needs write-locks, it can be only processed if this feature is supported. If this is not the case for example on a widely distributed setup, the data had to be stored separately and merged in a second step. By using this map-reduce-like approach, the computation can be spread to as many computation nodes as regions with POI are present.

### 5.5.4 Alternative Models

Alternatively, to the approach using the transportation time, other methods have been added for comparison. Voronoi polygons can be computed for comparison, using the original positions, as no access to the street network is necessary. For each Voronoi polygon, the number of sampled persons is found using a point in polygon check. Also, for each point of interest, all persons in a given linear distance can be selected as well as the average shortest linear distance to a POI. Determining if a person is in range, has to be repeated for each set of POI, this task is time-consuming. Both alternative techniques have no direct information about the needed time to the POI.

## 5.6 Analyzing the Results

When the results of the navigation task are available, a variety of further analyses can be performed which are needed for the different use-cases presented in Section 5.2.

### 5.6.1 Population Coverage

For each district to investigate, the set of contained nodes is stored. To achieve this, for each area all nodes, where at least one person is assigned to and are inside the polygons bounding box, are selected and then further filtered with a point in polygon check. This has only to be applied once per network as this mapping can be cached. Per node, the number of inhabitants and the result are directly accessible.

Different analyses can be performed. For example, it is possible to see how large the share of the population is, which is reachable within a given time. Alternatively, it is possible to compute the average result for the entire population or, for example, of the 95<sup>th</sup> percentile. Using the population data allows increasing the weight of densely populated areas in contrast to uninhabited areas.

These districts can then be presented on a map using a color scheme. Alternatively, a *hexbin plot* or other gridded schemes can be used. These allow a finer resolution but does not represent administrative borders. These plots are also weighted by the population data, to prevent that areas with more nodes in the graph are more weighted than areas with fewer nodes but more population.

In this work, the total population at their homes was determined. If further datasets would be available, it could be possible to filter the population on aspects like the age of the population. Alternatively, other metrics can be used while creating the population dataset, for example estimating the workplaces.

### 5.6.2 Highway Coverage

For some scenarios, the population coverage gives less to no information about service quality, as it only gives the potential number of people reached home. For traffic accidents or services for travelers, this metrics only have a small benefit. Therefore, the coverage of the highways is computed, as this type of streets is mainly used for wide area traveling

and are often hard to access, comparing to inner-city streets. Other types of streets can be investigated accordingly. As distances are lost in the first precomputing step, since only the needed time is stored, in a second step the distances for highways were stored while building the network separately. After computing the accessibility as before on the complete street network, the highway coverage is calculated. For this, every as highway classified streets and used nodes are extracted and combined with the information about the distances between the nodes. As the needed time is stored for each node, an additional step must be made to get the covered distance. In the data model, the speed between two nodes is constant. Therefore, it is possible to get the portion of the way which is reached in time, in the case one node was reached and the other not. It is also possible to obtain average times and other metrics, depending on the question to investigate.

Alternatively, computed areas from isochrones (see Section 5.6.6) could be used, but are lacking precision here as the driving direction is not distinguishable in the result. As in most cases, it is not possible to change the driving direction without a highway-exit, it is worthwhile to investigate the different lanes separately. This is also true for nearby or crossing streets and bridges. Using isochrones, this could lead to falsely classified reachable streets.

### **5.6.3 Statistics per POI**

Aside from results for each district, information about each POI can be obtained. For example, the number of accessed persons in a given time, the average driving time or the covered area. Also, the area of responsibility, in the sense of this POI has the shortest travel time, can be defined as well their extended area of responsibility in the case of the drop out of other POI.

### **5.6.4 Combining Different Scenarios**

To obtain a quality measurement for a given place or area, it can be relevant to combine different scenarios. This can be of interest to people or corporations looking for fitting places to buy or rent property. It is also relevant for more complex analysis.

The workflow allows combining different models and analysis. Apart from the scenario, there can be the maximum movement time specified as well as the weighting for the overall result. Several runs are started with early stopping when the time limit is reached. Therefore, different movement models can be mixed. For example, driving and walking can be combined in one run. This is true for all specifications which are possible set for separate runs. After each separate run, the result is stored and later combined using the specified weighting, creating a rated set of nodes or persons which are reachable within the given time constraints.

To obtain the score  $S$  for each node  $n$ , the value of the results  $r_{ni}$  in relation to their maximum value  $t_i$  of the  $I$  separate tasks are multiplied by the given factor  $w_i$  to create a weighting. Alternative and more complex formulas can be used.

$$S_n = \sum_{i=0}^I \left( \left( 1 - \frac{r_{ni}}{t_i} \right) * w_i \right)$$

### 5.6.5 Comparing Driving Time

For comparing the assumed transportation times, a utility was written which compares the assumed times to other services. For this, random POI and nodes of the network were selected. These pairs were given to different APIs which were performing standard routing tasks with time estimation. The exact parameters of the request are dependent on the called API. This obtained times can then be compared to the assumed times in this work. For results, see Section 7.1.1.

### 5.6.6 Isochrones

**Overview** For displaying the results, it is useful to create polygons, as displaying the nodes and edges are on the one hand problematic in sparse regions and on the other hand, with several million objects hardly displayable. By using polygons, the area between the streets is also shown as covered, creating a simpler view. While these polygons are often used for analysis, it gives this the risk that information is lost while approximating the result with the polygon. In this work, the polygons are only used for displaying the results; all analysis is done directly on the results.

In this work, the alpha-shapes algorithm was used. The shapes are only one way to visualize the results and practical if for example coverage areas are needed to compute. This data heavily depends on the choice of the alpha. For most scenarios, the time within the street network is more valuable as shops and points of interest are mostly close to streets or ways. However, with this data also areal statistics could be approximated in the case this information is needed.

**Computation** For generating the alpha shapes, Delaunay triangulation is used. This approach needs in contrast to other approaches like the convex hull more computation time but gives better results. This step is performed for computing the area of responsibility for each point of interest, as well as for the computed isochrones. This task is parallelized per POI to reduce the computation time. These computed polygons are then stored as *GeoJSON* for interactive displaying in a web browser. As the number of polygons can be too high for showing in a browser, alternatively they can be plotted, or polygon simplification is needed.

**Alternatives** Alternative approaches can be used, as described in Section 3.1.2 having their pros and cons. Alpha-shapes which are created on a complete set of nodes are most likely not practical for web services, as they need too much computation time if working on this large sets of nodes.

For plotting, also hexagonal binning plots are used. In this case, the map is divided into hexagons, and a weighted average value for all nodes in the hexagon is displayed, this enables an isochrone-like way of displaying. See Section 4.3 for different ways of visualizing the results.

## Summary

*In this Chapter, an overview of the design of the workflow was given. First, the different use-cases were sketched, giving an overview of the possibilities of the presented work. The three sketched use-cases were determining the coverage, combining different analyses to a quality map and performing a localization optimization. Determining the coverage allows estimating the services quality of given sets of POI. A combination of different coverage results allows more detailed analyses using several aspects and movement models. Further, the optimization approach allows to reduce the number of candidates as well as determining candidate quality. In the following, the data handling and preparation were described. In data preparation, it was shown how to obtain and pre-process the data from the different data sources. Examples are the creation of the movement model based on the OpenStreetMap dataset and the refinement of the population data. The design of the navigation task was described. Based on the Dijkstra algorithm the approach allows obtaining the needed movement time to all nodes in the street network. An improved approach is possible setting several starting points, allowing faster processing. The disaggregated population dataset in combination with the movement model allows more precise results as existing approaches. Also, the workflow for additional tasks, like identifying the population coverage or tasks for evaluation, was shown. In the next Chapter, the focus is set on the implementation details and challenges.*



# 6 Implementation

*In this Chapter, the implementation of the workflow is described. It starts with Section 6.1 for a general overview of the used techniques. In Section 6.2 the challenges of obtaining and processing the data are described. Following, in Section 6.3 some examples of the core elements of the analyses are given. Finally, in Section 6.4 insights to further analyses are presented.*

## 6.1 Techniques

Python3 was used as programming language. Python is widely spread in the research community. It allows on the one side fast prototyping due to its easy understandable syntax. On the other hand, Python has many fast and feature-rich libraries for various kinds of tasks. Using this language enables the use of these libraries, which speeds up development as well enables improved runtime speeds. For example, different algorithms are already implemented, APIs to different data sources are available, and the multiprocessing modules allow fast development of local parallelization on multiple cores. Python, as well as most used libraries, is platform independent so the implementation can run on Windows as well on Linux distributions. In many cases, the implementation is storing intermediate results in an easily readable format to ease debugging. For productive instances, aside from this thesis, further optimization would probably lead to a decreased runtime.

Each scenario was described in a settings file, determining information like the required model, a maximum number of simultaneous processes or the spatial range for each evaluation.

## 6.2 Data Preparation

First approaches in this work were using existing Python libraries like *geopandas* and *shapely*. These modules contain feature-rich implementations for computing and storing. Unfortunately, first tests showed that the runtime increased significantly compared to build-in data types. For this reason, in most cases, Python's built-in data types were used, and datatypes or complex functions of external feature-rich modules were discarded.

### 6.2.1 Street Network

**Data Format** The data is loaded from extracts or the *planet.osm* file. The file is split into smaller chunks and filtered for ways which are tagged with “highway”, determining all streets and paths. The splitting is done by the tool *Osmosis*. Several instances have been started, allowing to create the regions in parallel. It can be either used the XML based .osm format or the disk-space saving *ProtoBuf*<sup>1</sup> format. Since *ProtoBuf* was slightly slower in first tests, the .osm file type was chosen. In the case i/o-speed or hard drive size is of concern, the *ProtoBuf* or a compressed file would assumably result to faster processing times if the task is not computation bound. Splitting a large dataset is very time-consuming, taking several hours for the German map.

**Movement Models** As several different movement models should be supported, the parameters for converting the OSM data to a graph are defined using JSON settings files. This way can default values for each model and type of street be set or specific paths excluded.

**Error Handling** A variety of data source errors must be addressed, mostly due to the range of different tags and different possibilities how to store an entity in the OSM database. Problems were caused by unconnected streets, for example on parking lots, or one-way roads which are leading to restricted areas. If these ways are selected for starting the navigation task, it could not leave the limited area. However, also wrong tags for speed limits, for example, must be addressed. A slow counter direction traversing of one-way roads has been added, too slow to be included for valid routing but to skip wrongly tagged paths of a few meters. For example, driveways for parking garages caused problems as the interior was not defined and the car was ‘stuck’ at the starting point of the navigation task.

**Graph Storage** The obtained data is stored into CSV files, again split into regions for later parallelization. This allows the import through other tools for network analysis. The coordinates of all used nodes are stored in separate files. To reduce the number of nodes, in a first iteration it was count how often each node was used. If a node was only used once in an OSM object, it shows that no turn opportunity is given. While extracting the information these parts of the way can then be merged, aggregating the times of the individual parts, leaving only junctions per way.

### 6.2.2 Points of Interest

**Datasets** The process of obtaining the POI, the starting points to investigate, is split into two parts. The first part is to obtain the coordinates of the entities. In this work, in most cases, the POI were extracted from the OSM dataset. Preprocessing the dataset by discarding not needed information, improves the processing needed speed to extract the

---

<sup>1</sup><https://github.com/google/protobuf>

POI. First, the file is filtered by *Osmosis*, removing for example streets and borders. In an intermediate step, the file is read, and the content stored in an internal format, this allows fast queries for the POI without the overhead of reading the OSM file. Nodes, ways or relations representing the examined entities are filtered by their tags. If a way or relation represents the location, a single node will be selected. If two points are close to each other, the second one read will be discarded, as it is probably a second building or object of the same entity, or different data sources were merged. Alternatively, all other sources are usable, where a set of coordinates can be obtained.

**Assignment in the Graph** In the second step, the points are matched to the closest node of the street network. This step has to be done separately for each street model as the used streets can differ. For this matching, a spatial index (R-tree) with all nodes in the graph is used. This index allows a significant speedup as no longer to all nodes in the subgraph the distance must be computed.

### 6.2.3 Population

**Datasets** The extraction of the population grid data differs, depending on the used dataset as various data formats are existing. The GEOSTAT dataset uses a shapefile record dataset in the EPSG:3035 coordinate system which had to be transformed into the WGS84 coordinate system. The GPW population dataset uses an annotated TIFF which had to be transformed from image coordinates to the WGS84 system. For Python, libraries exist for converting this data, however, depending on the input format different libraries are needed to read and transform the data sources.

**Refinement** For the downscaling, residential areas and buildings from the OSM dataset were used. Polygons are stored in the OSM dataset as *ways* or *relations*.

A *relation* is, in this case, a combination of multiple *ways*. To obtain the corresponding polygons, the *ways* have to be combined. If all *nodes* of a building are inside of a larger residential area, the objects can be dismissed. For the downscaling of the population dataset, many points have to be randomly set and checked if they are in any polygon. This task achieves a better speed if again using a spatial index and first checking the bounding box before performing the point-in-polygon check.

**Assignment in the Graph** Using a spatial index improves the performance significant if looking in the second step for the closest node in the graph. This step is repeated for all sampled points, each representing one person. This way all nodes obtain the number of inhabitants which are probably living closest to the node.

### 6.2.4 Administrative Boundaries

As for the administrative boundaries also the OSM dataset is used, the procedure is first similar to extracting polygons of the residential areas. After transforming the polygons to a generic format, they are stored. Also, each node in the bounding box, using a spatial

index, is tested if its inside the polygon and not inside a hole of the polygon. This data is also stored for later usage. Storing the data is done in the JSON format for easy debugging or using the pickle module for more space and time efficient storing.

For displaying the results, the value can be averaged, or otherwise processed per region and is then stored in a *GeoJSON* format for display in a browser and plotting.

## 6.3 Computation

**Partitioning** The points of interest are loaded and separated by their spatial position. The graph data of the needed area is completely loaded into the main memory and stored in Python dictionaries, allowing fast access to all connected edges for each node.

**Search** Listing 6.1 shows the central part of the used search algorithm. An implementation of the Dijkstra algorithm using heaps was used, based on a proposed implementation by Eppstein<sup>2</sup>. The concept is described in Section 5.5.

**Storing the Results** To create statistics about each point of interest each point is computed separately. In virtual drive time intervals like every 15 minutes, the reached subgraph is stored to a file for this specific POI. This allows investigating questions

---

<sup>2</sup><https://mail.python.org/pipermail/python-dev/2003-April/035013.html>

```
1 D = {} # dictionary of final distances
2 P = {} # dictionary of predecessors
3
4 Q = [(0, None, node)] # heap of (est.dist., pred., vert)
5 heapq.heapify(Q)
6
7 while Q: # While the search frontier could be expanded
8     dist, pred, v = heapq.heappop(Q)
9     if dist > max_time: # early stopping
10         break
11     if v in D:
12         continue # tuple already reached, ignore
13     D[v] = dist
14     P[v] = pred
15     for w in connections_edges[v]:
16         newd = D[v][0] + connections_weights[v, w]
17         heapq.heappush(Q, (newd, v, w))
```

Listing 6.1: The heap-based Dijkstra algorithm

regarding specifying time borders. For this, on every expansion step, it is checked if the time border would be crossed, when `newd` is larger than the next border. If this is true, the current value of `D` is copied and stored separately.

If the computation is complete, because the search frontier could not be further expanded or the `maxtime` criteria has been reached, the data in `D` is the final result for this POI. The gained values in the distance dictionary `D` are stored and, after completing, written to disk for further processing. For the overall result, looking for the best time of all POI, before storing to disk, the best result for each node is selected from different tasks and then saved. It is also stored for each node in the graph, which POI is for this node the one with the fastest connection. This data gives, for example, information about the size of its area of responsibility.

**Multiple Starting Points** If multiple POI with overlapping areas are present, the subgraph is computed several times. If it is not necessary to know how many persons an individual POI can reach in a given time, rather than the overall result, multiple start points in the navigation task can be set simultaneously. For this, not one but all POI of this region are initializing the heap `Q` (Line 4). In this case, also the corresponding POI had to be stored in the heap `Q` and added while expanding.

**Parallelization** For parallelization, the *Python multiprocessing* library is used. For each area with goals and areas in a given radius, a process out of a processing pool is required. Write locks are globally available so each process can lock the needed files for storing their results.

## 6.4 Evaluation

### 6.4.1 Isochrones

The computation of alpha-shapes is quite complex as for larger polygons a large set of nodes have to be used. First simple approaches to thin out the data were not successful. For example, only saving the nodes where the time border was crossed results in incomplete polygons if the map ends before reaching the time at one place, for example at the coast. Other ways of sampling showed a decrease in output quality. Overall, alpha-shapes show the problem of having to guess the correct value *alpha*. Additionally, all results are just approximations. Therefore, the concave hulls are only used for displaying the results. All further analyses are done on the result set, which shows precise results.

While the Delaunay triangulation is part of different libraries, alpha-shapes are not included. Each isochrone can be processed in parallel, allowing a good speedup on large setups. Alpha-shapes are, all in all, creating well-formed results, if choosing the correct alpha value, following the classical alpha shapes algorithm, by removing long edges from a Delaunay triangulation.

Also, other approaches were implemented, this way for each analysis the algorithm could be changed. Polygons were created for each of the defined borders, as well as for the area where the POI is the fastest reaching. These areas were often overlapping, as the polygon creating using alpha-shapes is only an approximation.

These polygons were stored as GeoJSON objects and can be displayed by libraries like leaflet in a web browser.

### 6.4.2 Plotting

To create the maps and graphics in this thesis in most cases the Python library *matplotlib* was used. Together with the *basemap* extension, it is possible to plot geospatial data in a projection of choice.

### 6.4.3 Statistical Evaluation

For statistical evaluation, the *pandas* module was used, allowing to import and analyze the results by the accessibility analyses and the following steps. So, it is possible to compare different results for each district, depending on what was investigated.

## Summary

*In this Chapter, some insights of the implementation were given. Techniques for the data set preparation were shown, for example obtaining the movement model. Further insights into the routing algorithm were given. Also, some implementation details for evaluation tasks were shown. In the next Chapter, the results of the evaluation are presented.*

# 7 Results

In this Chapter, results are presented. The results are structured in different scenarios as described in Section 5.2 and processed in the described workflow. It starts with evaluation results of the used datasets in Section 7.1. Section 7.2 shows for the coverage workflow various experiments. Section 7.3 combines different results in the example of settlement and holiday locations. Section 7.4 shows two example results of the optimization workflow.

## 7.1 General Evaluation

### 7.1.1 Comparing to Navigation Software

To evaluate if the movement model is based on realistic values, the achieved movement times of the accessibility analyses were compared to other navigation systems. For this purpose, a random set of results was taken and given to an accessible API. For comparison the *Open Source Routing Machine (OSRM)* as well as the *Google Directions API* were used. Due to rate limitations of this services and the high number of experiments conducted, no full comparison is possible. However, using a subset allows estimating if results of the here used navigation tasks are comparable to the time estimation of other, broadly used, services.

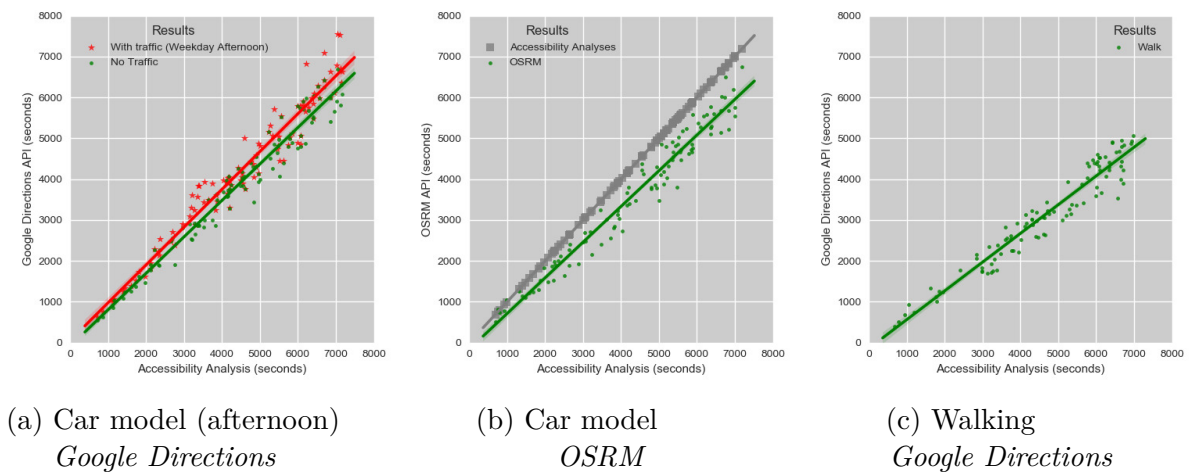


Figure 7.1: Comparing results with routing APIs

**Driving Model** Figures 7.1a and 7.1b compare the standard car movement model used in this thesis to other services. One point represents one randomly selected result. Results close to the diagonal axis, are showing similar times in the here used model and the API result. The times of the model are in many cases close to the results of the *Google Directions API* including traffic-related delay.

These results show that the here used model can give a useful approximation of the movement times. The needed times by the *OSRM*, are a little bit lower, here no traffic estimation was given. It is difficult to define what the 'correct' result would be, but the results show that the estimated values for the accessibility analyses are close enough to the estimation by other navigation systems, even using a simple model as in this thesis. As no real-time traffic data is used in this work, in heavy traffic situations would a system like *Google Maps* assumable show more realistic results. Other differences between the models may be caused by differing assumptions of the vehicles speed or other differences in the movement model.

**Walking Model** Figure 7.1c shows the walking model by *Google Directions*. The movement speed is remarkable faster as in the model used in this work. This high walking speed was a surprising result, as it indicates that this model is assuming a significantly higher average walking speed of 5 km/h.

### 7.1.2 Quality of the Street Network

To investigate if the whole street network obtained from OSM is connected, a navigation task for all fire stations in Germany was performed (see Section 7.2.1), and the accessed number of population evaluated. If a part of the population is not reachable by this analysis, it would indicate that the street network is in parts not connected and persons cannot be reached.

The evaluation showed that 79,870,094 persons were reached from the sampled population of 80,111,071 persons. This equals 99.7%. While on some smaller islands no fire station is available, a significant share of the missing population is due to errors in the street graph. As this error is relatively small, it will not further be addressed in this thesis, but model optimization should be considered in future work.

### 7.1.3 Population Downscaling

For population estimation the gridded dataset was disaggregated as described in Section 5.4.4. Figure 7.2 shows examples of the generated population positions. Each blue marker represents one person. For Germany, over 80,000,000 persons were sampled this way. It is visible that in areas where no people live, for example, lakes, parks or agricultural fields no persons are sampled in contrast to residential areas and buildings.



While the population estimation neither represents the truth nor is a perfect estimation, it gives a better estimation of the population distribution as in the original population dataset. This approach allows a more realistic assignment of inhabitants to the street network. This way, for example, fewer residents are assigned to parks or connection roads.

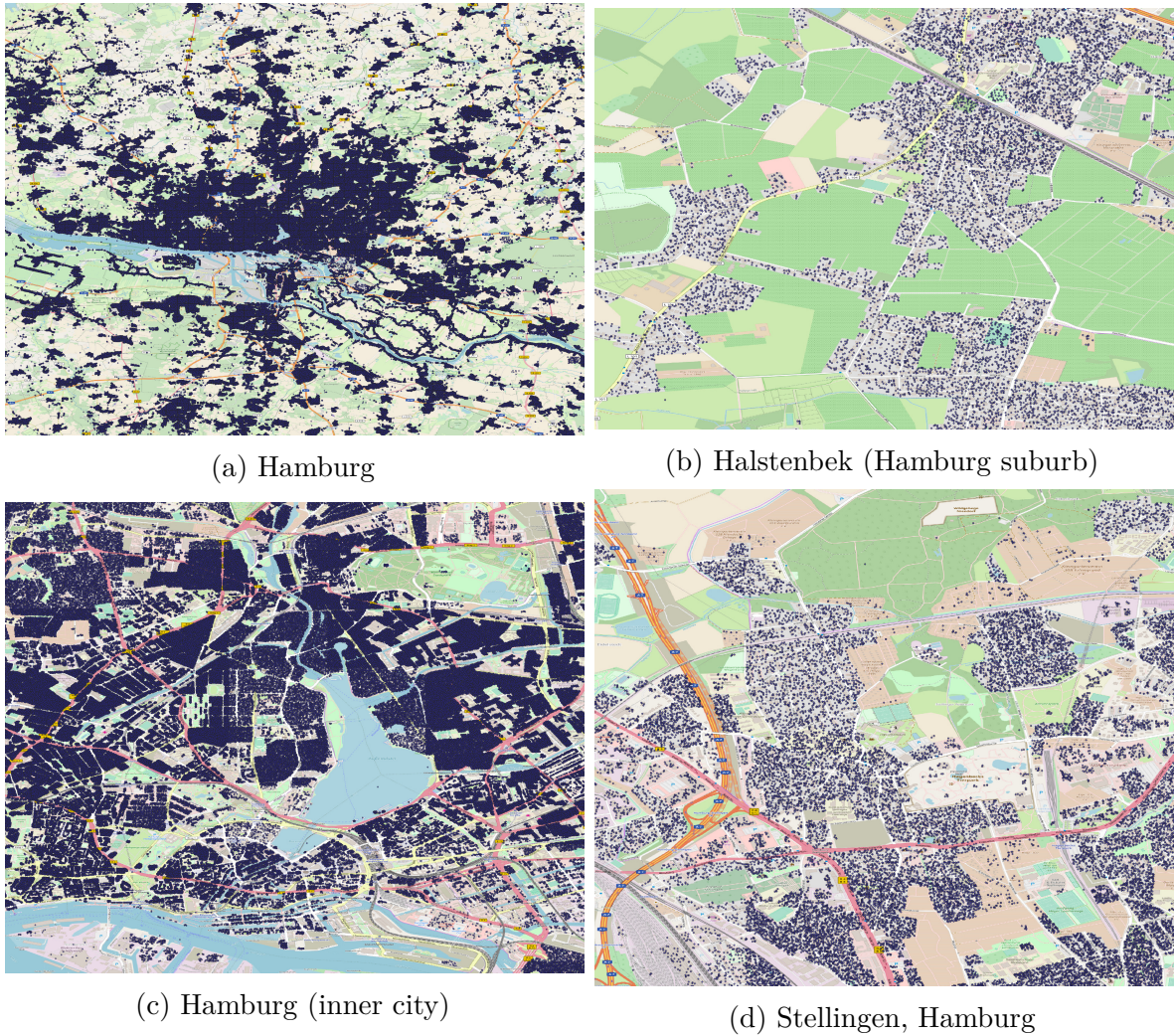


Figure 7.2: Results of population data downscaling - Each blue marker represents one person

## 7.2 Scenario A: Service Coverage

### 7.2.1 Fire Station Coverage in Germany

<i>Street data:</i>	OSM - Germany (18.03.2018)
<i>Population data:</i>	GEOSTAT 1 km <sup>2</sup> - 2011 - refined
<i>Model:</i>	OSM - refined for emergency services(see Section 5.4.3)
<i>Model Size:</i>	Nodes: 44,049,151 - Edges: 90,971,894
<i>POI:</i>	OSM (18.03.2018) 92 (professional)/22,288 (voluntary)
<i>Start delay:</i>	1 min (professional) / 4 min (voluntary)

Table 7.1: Fire station coverage in Germany — Data sources and settings

**Setup** In this first test, the estimated service coverage of fire stations in Germany is investigated. The used data sources are shown in Table 7.1. For an insight into the processing workflow, see Section 5.2.1. For this experiment, a full analysis was performed to obtain data for each individual POI additionally. This increased the execution time of the search task to about 50 minutes on the test system (48 core CPU - AMD Opteron 6344) compared to about 7 minutes of the fast analyses. Over 22,000 fire stations are listed in Germany in the OpenStreetMap dataset. 92 Fire stations were recognizable in the dataset as professional fire stations. We assume that stations of the professional forces need one minute from the alarm until the first vehicles can leave the station, and assume four minutes for the forces of voluntary stations. The movement model is representing driving under special rights (see Section 5.4.3). In Figure 7.3 all found fire stations in Germany are shown.

**Overall Results** The overall result is shown in Figure 7.4. It displays the average time, till the first fire station can reach their goal. The results are weighted by the number of inhabitants at each node in the graph. Two kinds of maps for presenting this result were used. The first one (see Figure 7.4a) uses hexagonal areas of a fixed size. This kind of plotting allows a high resolution for the maps. The white areas determine uninhabited areas or are outside of the map for this simulation. Areas which are hardly covered by the fire stations are clearly visible, even though maybe only a few inhabitants live there. Figure 7.4b on the other hand shows the average needed times per district or independent city in Germany.

Both kinds of maps are used in this work. It is noticeable that the average time is differing between regions, Figure 7.4a shows areas with high average times in the North-East of Germany. However, since the average times per district (Figure 7.4b) show no remarkable high average times, it gives a hint that in most cases the areas with inadequate coverage are not densely populated. All districts and independent cities show an average time of 9 minutes or less.

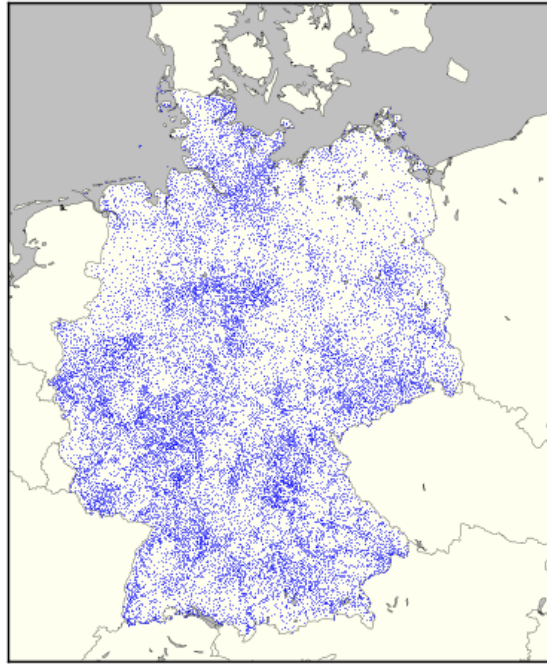


Figure 7.3: Fire departments in Germany, each blue marker represents a fire station

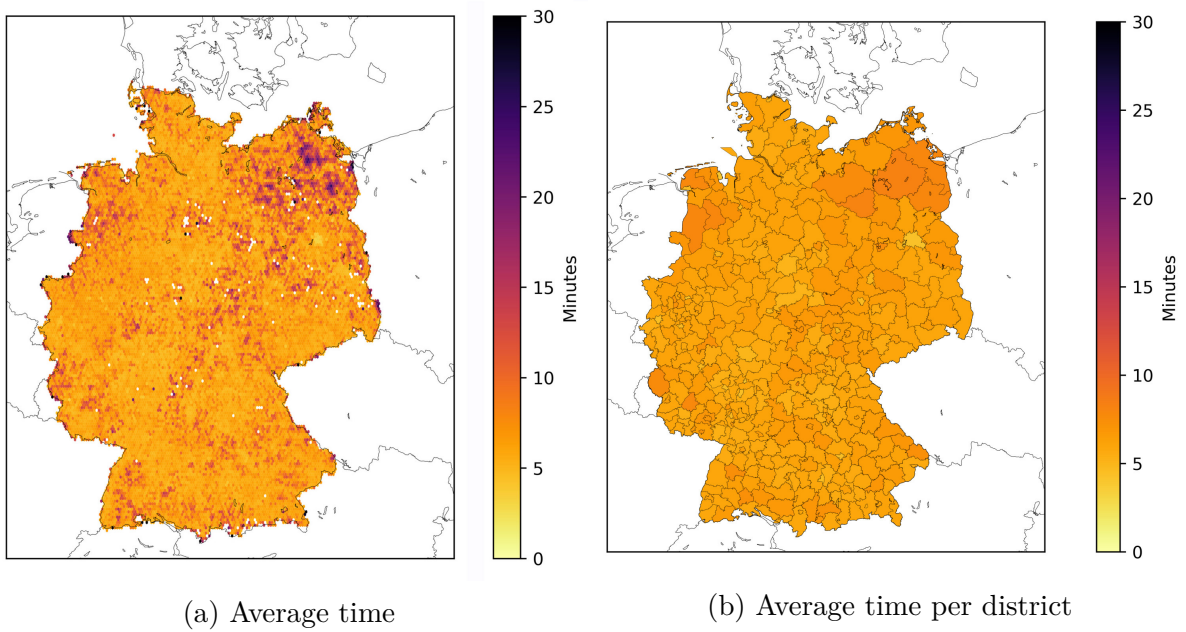


Figure 7.4: Fire station coverage in Germany



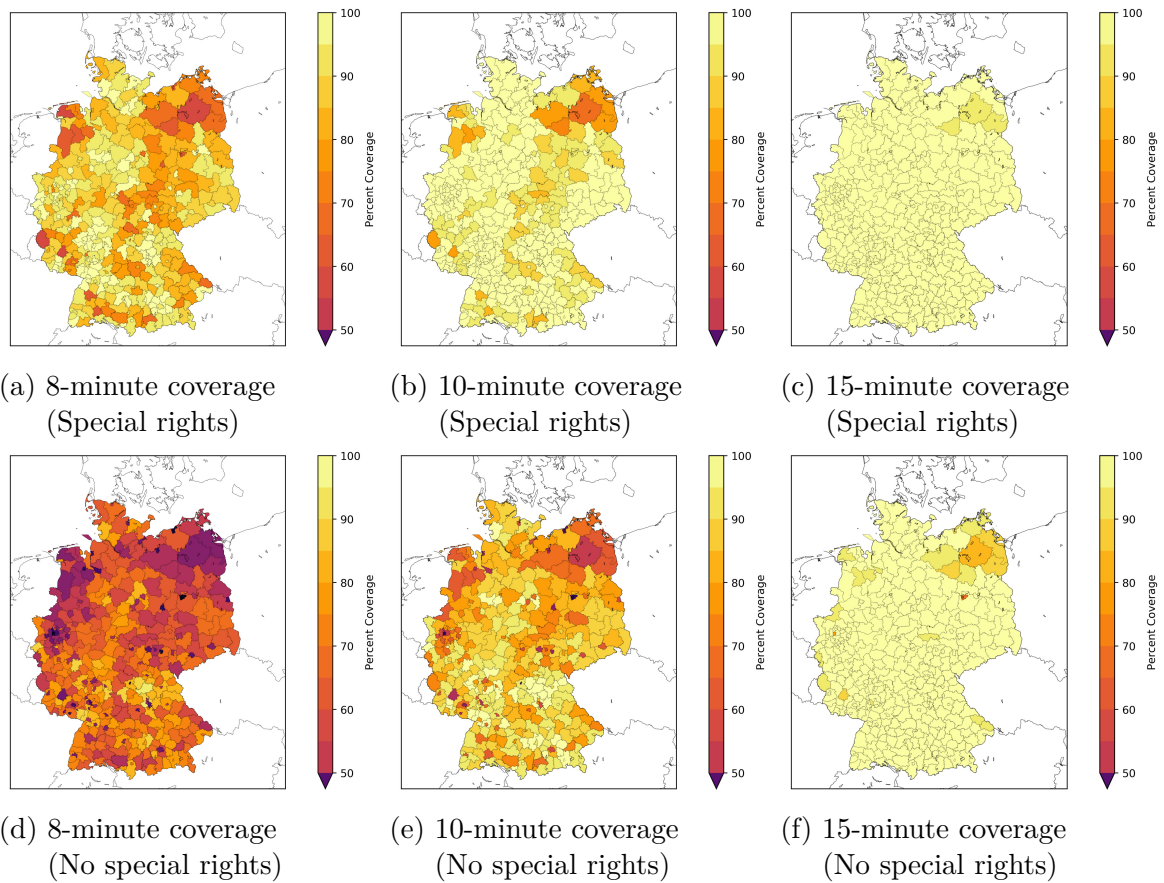


Figure 7.5: Percent of coverage per inhabitant in given minutes per district in Germany, with and without special privileges - Fire stations

**Population Coverage** Figure 7.5 shows the percentage of the population in each district which is reachable in 8, 10 or 15 minutes. The results are presented for the normal car movement model, as well as, the dedicated model with privileged rights. The results of the simulation determined that urban areas are in most cases well covered, while rural areas are performing unsatisfactorily. While the city of Hamburg is able to cover 97% of the population in 8 minutes, in *Landkreis Mecklenburgische Seenplatte* only 56% of the population is covered in this time. However, also in the legal requirements, this difference is present. Every federal state has differing laws, and some even distinguish between rural and urban areas or use other measurements. It is visible that without privileges a significantly smaller share of the population would be reached. In Table 7.2, the best and worst covered districts are shown.

**Additional Stations** While it seems that under optimal conditions all districts can fulfill their requirements, the needed arrival times of the second and third fire brigade were evaluated. Figure 7.6 shows the needed times of the first, second and third arriving fire brigade. In case of large emergencies, if several emergency cases happen at once or in other cases, when the closest station is not available, the needed time of the second or third arriving fire brigade is of concern. For example, the Eastern Frisian islands only have one fire station per island and no street connection to the mainland so that no other stations could reach the destination. Unfortunately, no data was available for the size or capacity of each fire station, as smaller stations often have only one brigade other stations have several brigades and can handle more than one emergency. The existing model could be adapted for this case: If this information would be available, one start point for each fire brigade in a station could be set, and be treated as two independent stations. Dedicated models for specific times of the day or specific kinds of emergencies could this way created, as it is possible that a station may handle one fire and one traffic incident but not two fires at the same time.

**Areas of Responsibility** Figure 7.7 compares the number of persons, which are in the catchment area for each station, between the shortest linear distance and the fastest driving time. In the Voronoi model, the persons who have the closest linear distance for each POI are determined, while for the here compared accessibility analyses results the fastest driving time is used. The results show that the measurements are similar but have in some cases differences. For example, volunteer fire stations may are on locations

District	Avg. time	% 10-minute coverage
Wilhelmshaven	3.9 minutes	99.9%
Offenbach am Main	4.0 minutes	99.7%
...	...	...
Landkreis Prignitz	8.2 minutes	86.8%
Landkreis Mecklenburgische Seenplatte	8.3 minutes	67.1%

Table 7.2: Best and worst results - fire stations

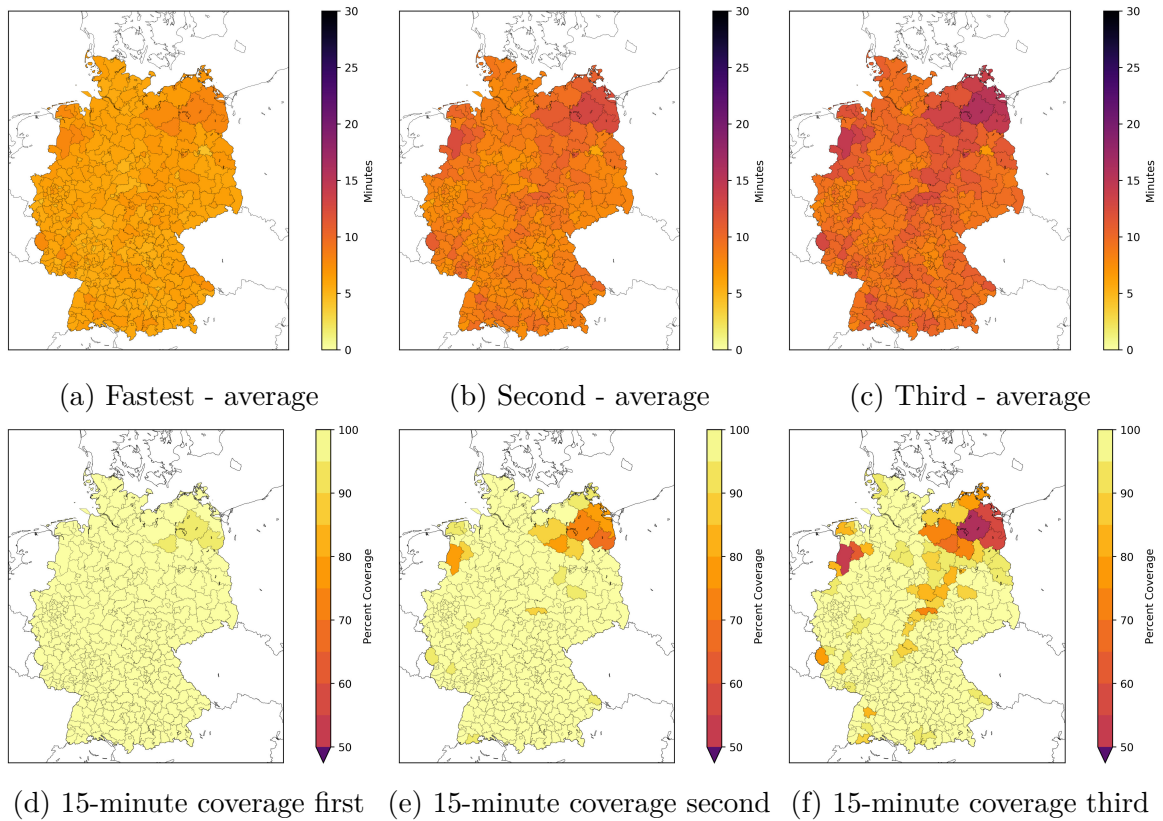


Figure 7.6: Fire station coverage, unavailability of fastest station

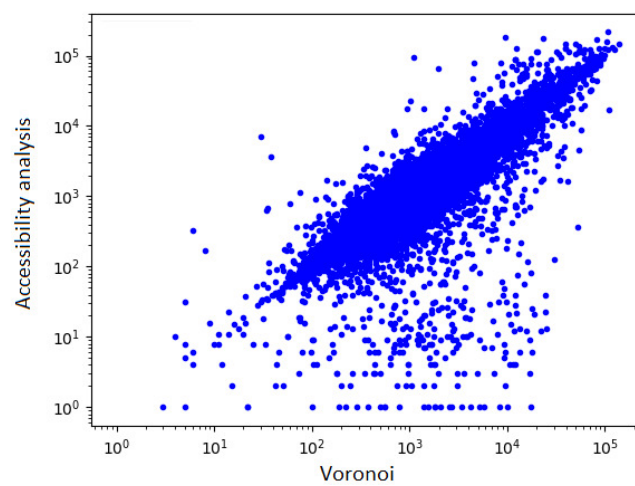


Figure 7.7: Comparing Voronoi to accessibility analyses - Persons per station (log. scale)

which are reached by a professional fire brigade even before the voluntary forces would be able to leave their station. In these cases, no or only a few persons are in their catchment area according to the driving time model. Also, stations in industrial regions may show suspicious low population coverage. Further, some data errors in the OSM dataset were present, so that different buildings were tagged as independent fire stations. While in the Voronoi diagram the area would be split, in the accessibility analyses it is possible that the navigation task for one of the POI stops at the first junctions as the other POI already covered this part.

**Highway Coverage** As another example, in Figure 7.8, the results for the highway coverage are presented, this is of special interest as emergency services are needed here as well, but often the highway traverses rural areas. Furthermore, the highway can only be entered at designated points, and turns are in most cases not possible. As shown in Figure 7.8b, in many cases only one lane is accessible, till the next opportunity to turn. These times do not take into account that the road may be blocked, for example, if no emergency corridor on the highway was cleared, which can delay the arrival of the emergency services drastically. The results show that highways are significant less covered than residential areas.

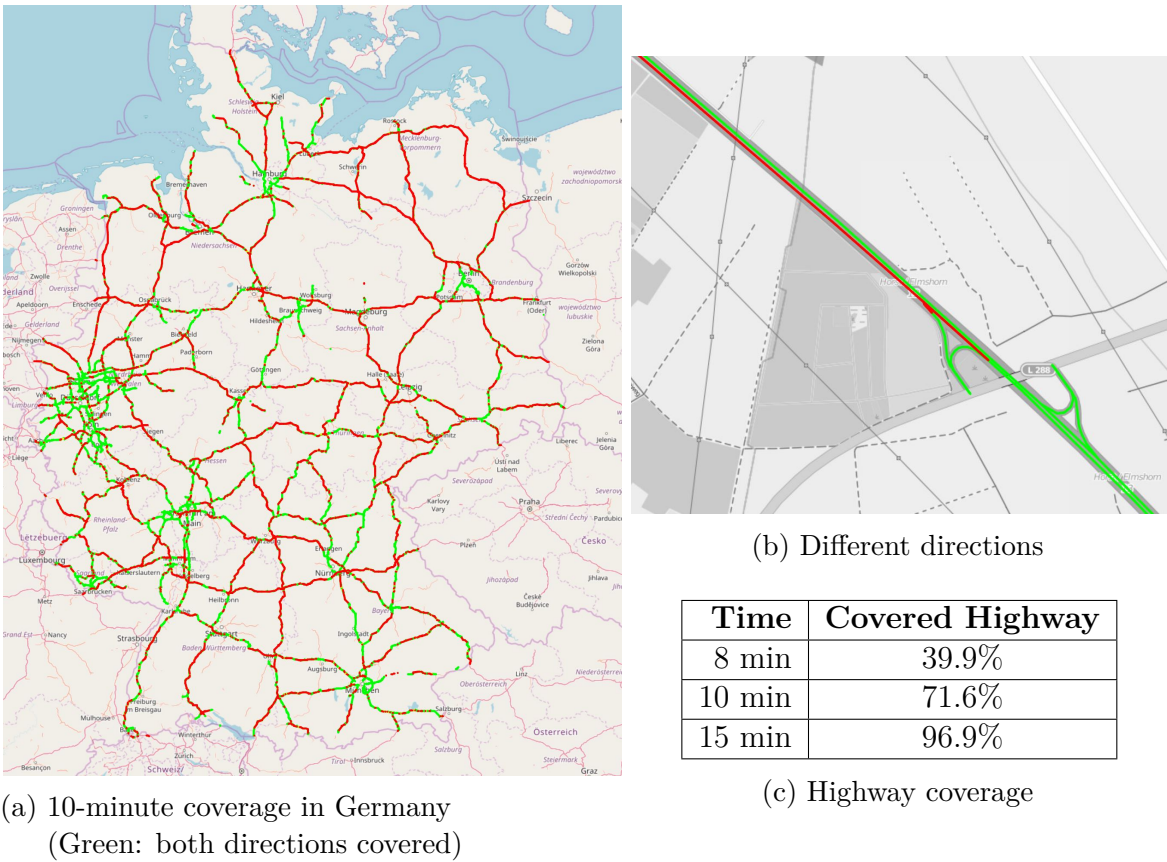


Figure 7.8: Highway coverage in Germany of fire stations

**Linear Distance** In Figure 7.9, the difference between the movement time and the linear distance is shown. In this case, the results from the fire station analyses were used per district. Each point represents the average values of one district or district independent city. Depending on the given region the difference is more or less significant. In general, the two measurements are linear matching. Several areas showed significant lower movement times. These were in most cases districts with professional fire brigades, which have a lower start delay.

This shows that the movement model has its advantages especially if modeling such start delays. However, a deviation is also visible in other districts, even if the average values of many stations are used.

**Remarks** In the dataset only 92 professional fire stations were found, so assumable several stations were misclassified as voluntary stations, this can lead to imprecise results. In Germany, about 100 cities have a professional fire brigade, with a total of 324 stations [Heg18]. Additionally, there exist full-time manned voluntary stations which were not determined. Further, the time till the first vehicles can leave the station is dependent on further factors like the time or the day of the week. The results can be improved if corresponding data is available. Also, not necessarily the fastest station is alarmed, other aspects as the available tools and kinds of vehicles are taken into account. For these cases, dedicated analyses can be performed.

## 7.2.2 Fire Station Coverage in Northern Germany

**Setup** For evaluation, we have a closer look at northern Germany. With the example of the federal state of Schleswig-Holstein, Germany, we try to find regions which are not sufficiently covered and would benefit from new stations. The exact locations for new

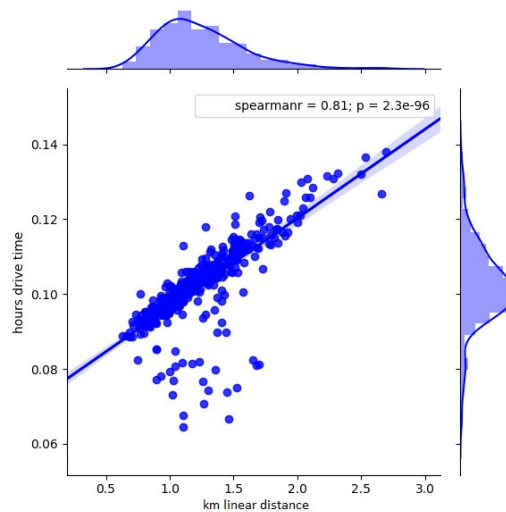


Figure 7.9: Fire stations: Comparing movement model to linear distance



buildings could be determined by the local communities. These results could also be automatically used for location optimization (see Section 7.4). In Schleswig-Holstein, a 10-minute coverage is requested including the alarm time. Assuming two minutes for alarming, we investigate the 8-minute coverage and have a look if the requested times can be fulfilled according to the here created model. We investigated every postal code area, allowing a finer resolution as using the districts.

**Areas of Responsibility** Figure 7.10 compares the areas with the shortest linear distance and fastest driving time. In Figure 7.10a the areas are shown for each station which can be reached fastest, representing the optimal areas of responsibility, according to this model. While for some stations similar areas are presented, others differ a lot. For example, professional fire brigades cover a larger area, as they can act faster. In Figure 7.10b the Voronoi diagram cannot represent these differences. If data is available how fast the single stations can be manned, these areas by the accessibility analyses can be improved even further.

**Results and Review** Figure 7.11 gives an overview of the results. By having a closer look at the worst covered districts, the problem was identified as missing stations in the dataset. To investigate the impact of missing stations, in a second attempt, nine more fire stations have been added manually from other sources as far as information was found. This second evaluation showed different results as shown in Figure 7.11d. The areas which changed compared to the first set of POI is shown in Figure 7.11e.

For the quality of the analyses, in a small-scale evaluation, the data quality of the POI is crucial. It shows that the dataset is not complete, but the workflow can be used to determine missing POI. If the source dataset guarantees a sufficient quality, it can find places with insufficient coverage. Otherwise, results must be verified.

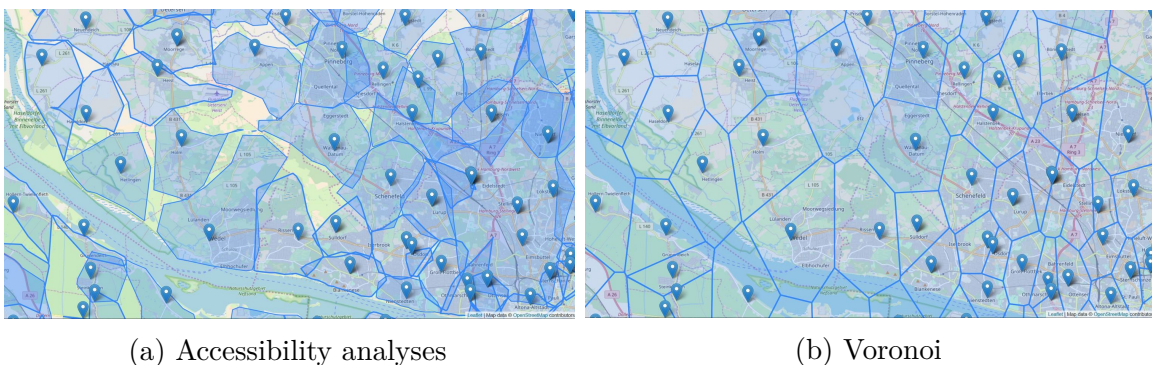
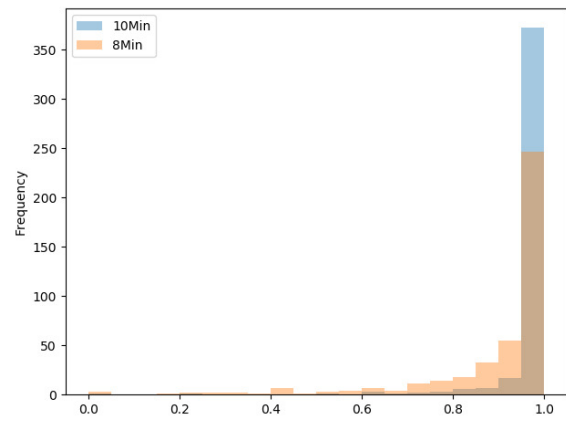


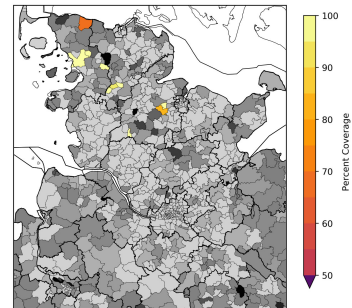
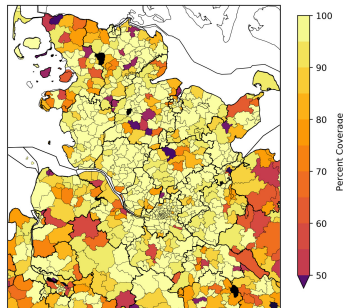
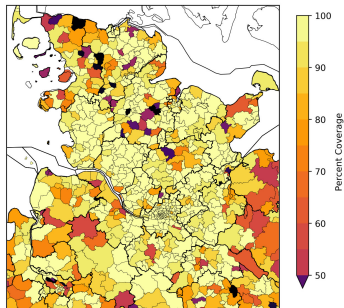
Figure 7.10: Areas of responsibility - fire stations

Postal Code	% Covered 1st run	% Covered 2nd run
25926	00.0%	67.4%
24259	04.0%	82.6%
25850	05.4%	100.0%
24872	06.7%	100.0%
24899	08.1%	91.2%
25575	17.1%	100.0%
23566	18.3%	18.3%
24598	24.6%	24.6%



(a) Worst covered areas (8 Min.) – Impact of missing stations in the dataset

(b) Histogram of areas reaching given coverage



(c) 8-minute coverage 1st run

(d) 8-minute coverage 2nd run

(e) 8-minute coverage – Diff

Figure 7.11: Fire station coverage in Schleswig-Holstein, Germany

### 7.2.3 Hospital Coverage

<i>Street data:</i>	OSM (18.03.2018)
<i>Population data:</i>	GEOSTAT 1 km <sup>2</sup> - refined
<i>Model:</i>	OSM - refined for emergency services
<i>POI:</i>	OSM (18.03.2018) 3,006 entries (tag: “hospital”)

Table 7.3: Hospital coverage — Data sources and settings

**Overall Results** Giving a second example, Figure 7.12 gives the coverage of hospitals. In this case, the needed time from the place of accident to the hospital is shown. In this case, a scale from 0 to 60 minutes was used.

The coverage in cities is good, while in rural regions often long transportation times are necessary. Several islands are not reachable at all, as no hospital is present. In emergency situations, the transportation with a helicopter is possible, to move patients faster as using an ambulance. Figures 7.13a and 7.13b show the fastest reachable and nearest hospitals in northern Germany.

**Remarks** OpenStreetMap distinguishes between hospitals and clinics. Latter is used for smaller facilities or if only specific branches of medicine are covered. It is from this dataset not reconcilable if a hospital can provide sufficient service for the particular case of emergency. By using other datasets, hospitals could be selected which are, for example, capable of handling brain injuries.

### 7.2.4 Electric Car Charging

<i>Street data:</i>	OSM (18.03.2018)
<i>Population data:</i>	GEOSTAT 1 km <sup>2</sup> - refined
<i>Model:</i>	OSM - drive distance
<i>POI:</i>	OSM and <i>Bundesnetzagentur</i> (see Section 4.2)

Table 7.4: Electric car charging — Data sources and settings

**Setup** As an example of a different transportation model for coverage analyses, the coverage of electric car charging stations is investigated and shown in Figure 7.14. Instead of the needed transportation time, the needed driving distance is used.

**Results** The results showed that in most regions a charging station is available within less than 25 km. Additional to residential charging opportunities, highways show a good coverage as many charging stations are located at rest stops. As charging stations are often needed while driving long distances, it is a good sign that highways are well covered.

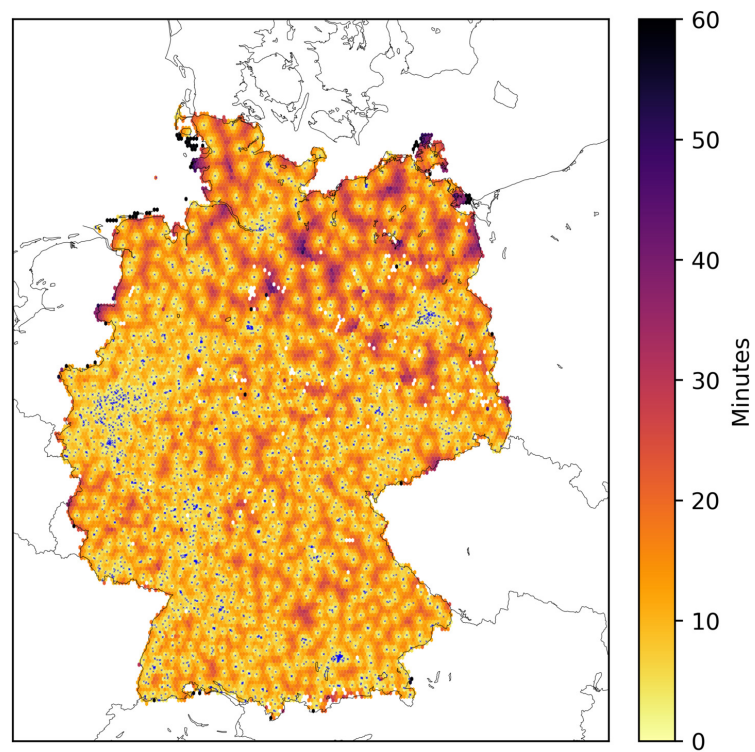
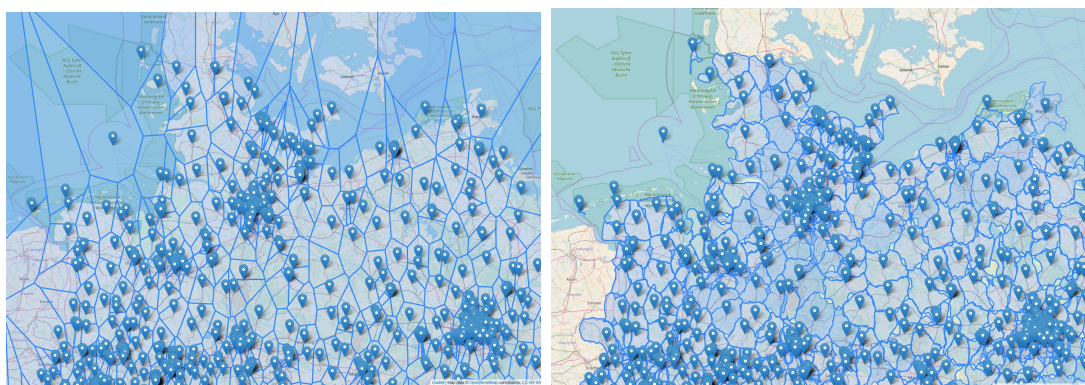


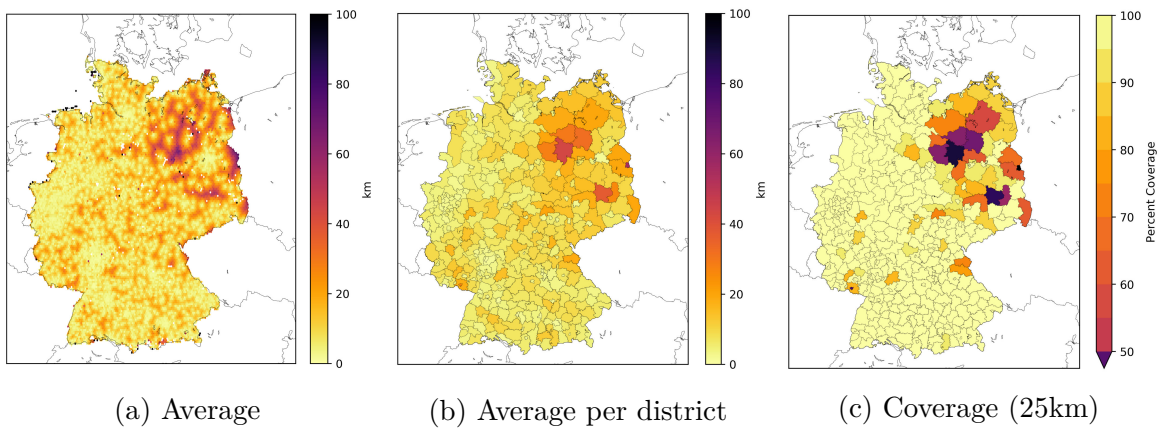
Figure 7.12: Needed time to the closest hospital



(a) Voronoi

(b) Accessibility analysis

Figure 7.13: Hospitals: Optimal areas of responsibility in northern Germany



Max. Distance	% of Highway
10km	66.1%
25km	96.3%
50km	99.8%

(e) Coverage

(d) Highway Coverage 25 km

Figure 7.14: Coverage of electric car charging stations in Germany



## 7.2.5 Stroke Units

<i>Street data:</i>	OSM (18.03.2018)
<i>Population data:</i>	GEOSTAT 1 km <sup>2</sup> - refined
<i>Model:</i>	OSM - refined for emergency services
<i>POI:</i>	German Stroke Society (11.5.2018)

Table 7.5: Stroke units — Data sources and settings

**Setup** For comparison, the results by Rauch und Rauh [RR16] shall be reproduced. For this, the situation in Germany with over 300 stroke units has been investigated. For this purpose, a list of all certified stroke units in Germany was obtained by the *German Stroke Society*<sup>1</sup>.

**Result** The analysis showed that 68,216,867 of 80,111,151 (85.2%) persons were reached within 30 minutes in Germany. An overview is given in Figure 7.15. Rauch and Rauh came to a similar result of 87% coverage with their areal approach, using 279 stroke units (number of units as of the end of 2015).

However, they described *Landkreis Ludwigslust-Parchim* with a coverage of less than 6 percent. This result could not be verified, in this work, it reached coverage of 27.3%, while other districts showed no coverage at all (*Dessau-Roßlau* and *Landkreis Lüchow-Dannenburg*). One reason for the highly different results could be a change in the position of the certified stroke units. Other differences in the results could be caused by the difference in methodology.

<sup>1</sup><http://www.dsg-info.de/stroke-units/stroke-units-uebersicht.html>

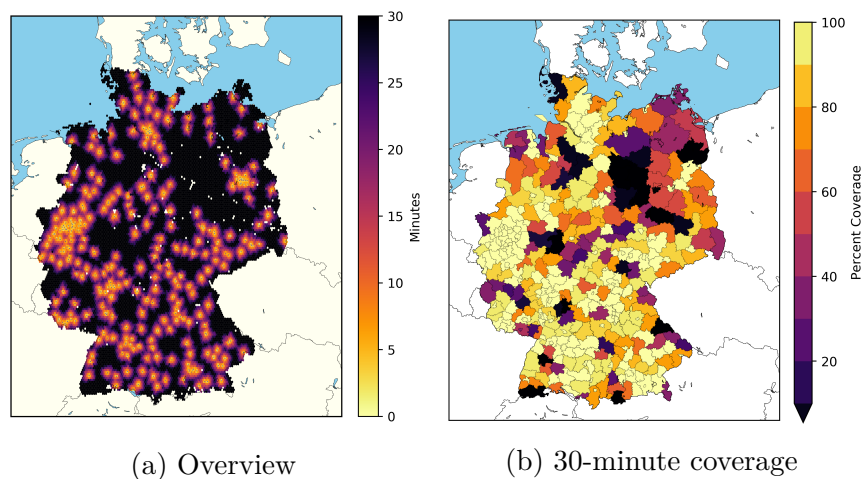


Figure 7.15: Coverage of stroke units

## 7.2.6 Pharmacies

<i>Street data:</i>	OSM (18.03.2018)
<i>Population data:</i>	GEOSTAT 1 km <sup>2</sup> - refined
<i>Model:</i>	OSM - Car Standard (see Section 5.4.3)
<i>POI:</i>	OSM (18.03.2018) 18,782 POI

Table 7.6: Pharmacies — Data sources and settings

**Setup** In this Section, the results of pharmacy coverage are compared to the results by Peter and Gertz [PG17].

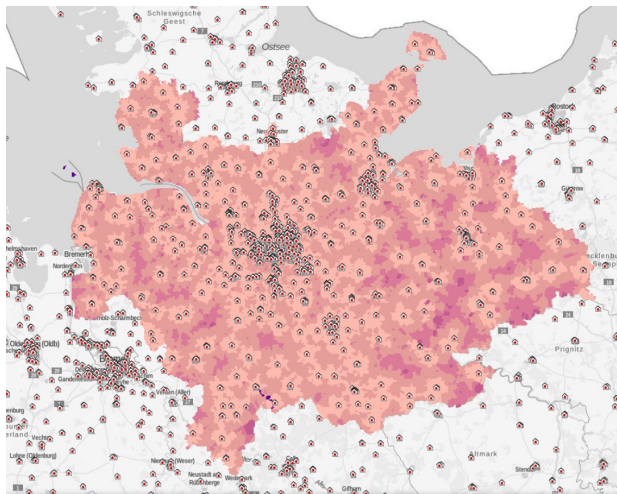
**Results** Figures 7.16a and 7.16b show an overview of the coverage by Peter and Gertz and this work, it is visible that the results are comparable. A look at detail-level (Figure 7.16c) compares the degree of detail. The gridded fields represent the work by Peter. The points are the result of this work. Here, each node in the graph has its individual value, in contrast to approaches where only the center of grid cells was investigated, this allows finer resolutions and to consider, for example, the driving directions as well as single streets. This way the accessibility of different streets inside the grid cell is taken into concern. As each of the nodes is assigned with the assumed number of inhabitants, also the number of inhabitants can be determined in this resolution.

**Performance** The computation time is reasonable for analyses. In this example, the navigation task itself (without data preparation or further regional analyses) took about 25 Minutes for over 18,000 POI in Germany on a standard desktop PC (4-core CPU - Intel i7-4790k). Running on a sparse graph, using only the junctions as nodes, decreases the time to about 7 Minutes. The time can further be decreased by running on a more powerful system. On the other side, the needed time and memory increases, if the maximum transportation time (in this case only 30 minutes) is higher and therefore more parts of the network have to be loaded for each region. The time for data preparation and further analyses can take several hours, depending on the scenario.

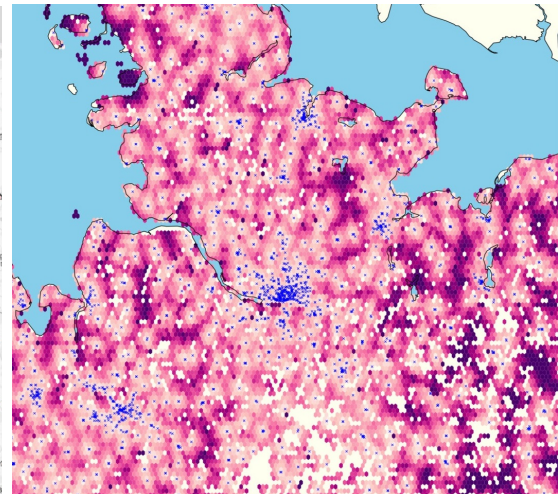
## 7.2.7 Discussion and Conclusions

The different scenarios showed good results obtaining the coverage of services, for large sets of POI as well on a small scale with a high resolution. While the needed computation time is currently too long for web applications, the time requirements are manageable for performing analyses.

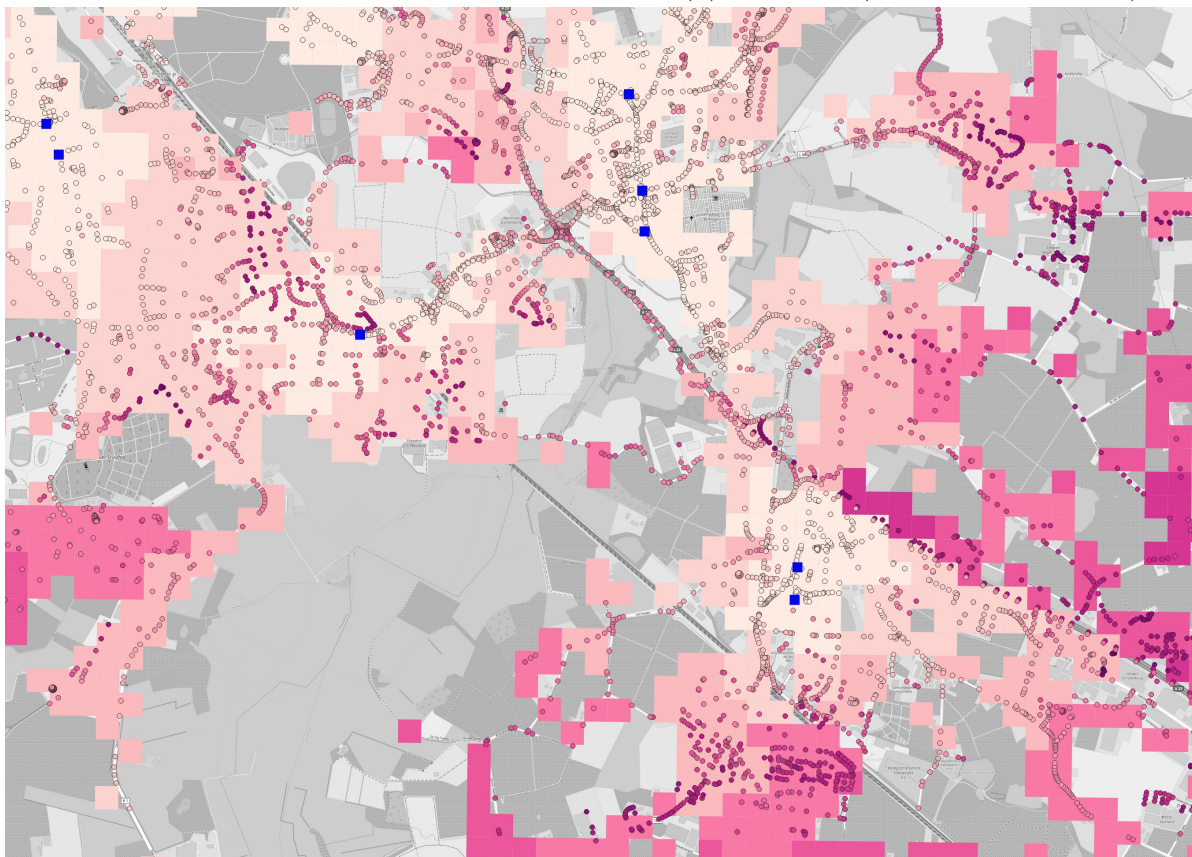
To mention is that in this simulation always the optimal POI was selected and always the fastest route is taken. If the areas of responsibility are not optimal, this had an impact on the coverage. However, this approach could be used to create better areas of responsibility. The computed times are assumed averaged times, there exist cases where



(a) Overview ([PG17])



(b) Overview (accessibility analysis)



(c) Comparing on detail-level (*Pinneberg*)  
Squares by Peter [PG17], Points by this work

Figure 7.16: Driving time to the next pharmacy – brighter color: faster reachable

Map: *geoportal Hamburg* [http://geoportal.metropolregion.hamburg.de/mrh\\_erreichbarkeitsanalysen/](http://geoportal.metropolregion.hamburg.de/mrh_erreichbarkeitsanalysen/), OpenStreetMap



the needed time can be significantly higher, for example, if a railway crossing gate or a traffic jam on the highway is blocking the way. For these cases, a worst-case runtime could be created by adding these obstacles into the movement model.

The quality of results depends on the completeness of the POI. Nevertheless, if the dataset is complete, precise results can be obtained. These can be used for large-scale analyses, determining coverage across different districts or countries. Also, it can be used for local communities to recognize problems in small-scale scenarios. If altering the map by removing single streets, the change in coverage can be shown if streets are blocked due to construction work or similar.

## 7.3 Scenario B: Combined Quality Map

*In this scenario, we are creating a map displaying the quality of regions using the needed time to a set of different POI as described in Section 5.2.2.*

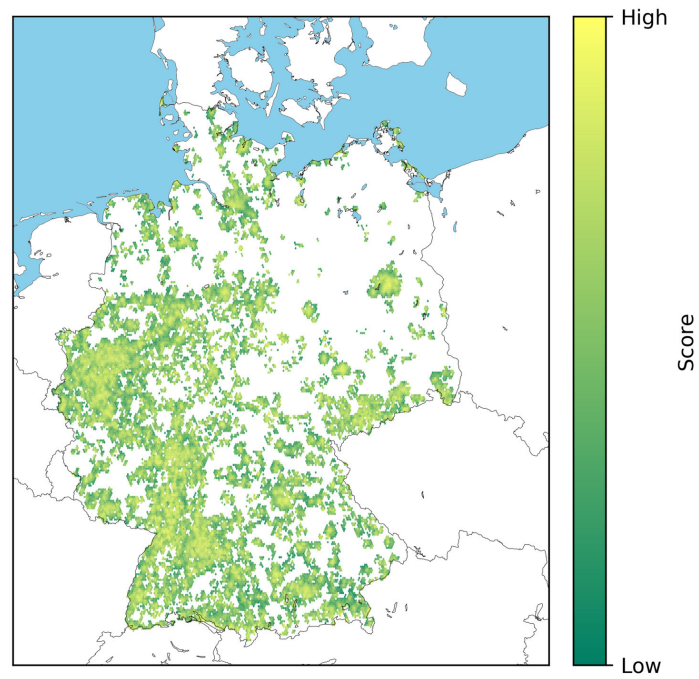
### 7.3.1 Holiday/Leisure Quality

**Setup** For this scenario, we create a set of requirements as an example of personal interest and requests for a holiday site as presented in Table 7.7. Figure 7.18 is showing the separate coverage maps which were combined. The score is computed as shown in Section 5.6.4, combining the values from the different analyses for each node.

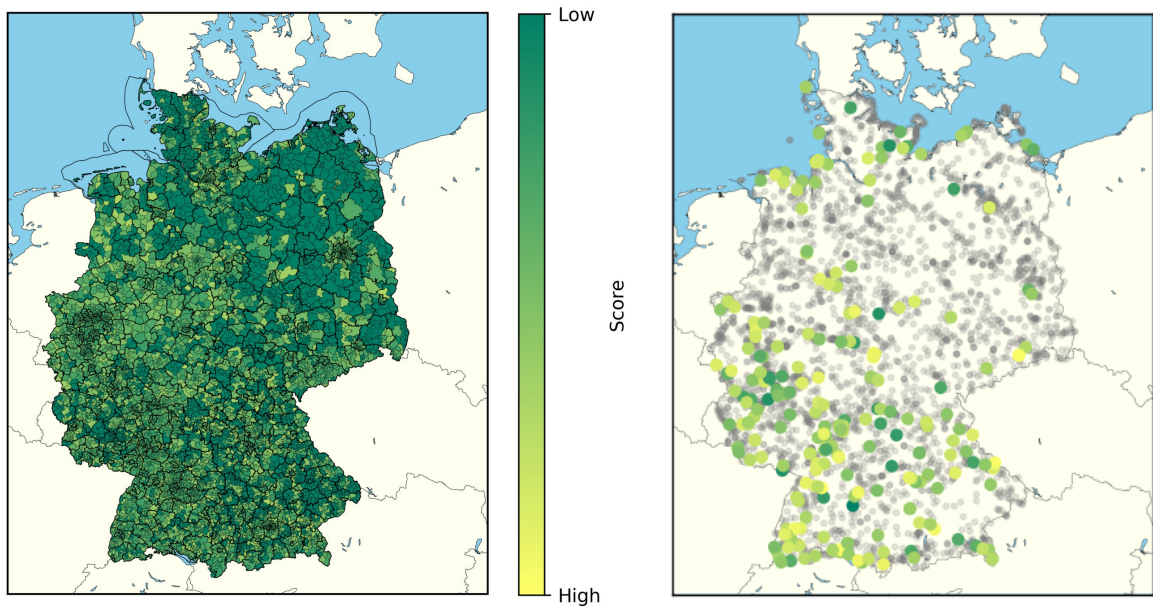
**Results** The map in Figure 7.17a shows a score for different regions. A higher score represents a better coverage of the requested services. Such map can be used to determine regions of interest if looking for a settlement place, holiday locations or other cases where a location based on different parameters is needed. All places are shown and rated which allow fulfilling the requests for this example in 60 minutes. Areas which are failing in fulfilling at least one request are not shown on the map. Each node in the graph has its individual score, therefore also average data for regions can be obtained. In this case, it represents an average holiday facility quality. This data can also be averaged for each postal code area, as shown in Figure 7.17b.

POI	Model	relative weight
Two museums	Car	0.3
Waterpark	Bike	2.0
Bakery	Walk	1.0
Miniature Golf	Bike	0.4
Zoo	Car	1.0

Table 7.7: Requests for the holiday location



(a) Quality map



(b) Average score per postal code area

(c) Score of camping sites

Figure 7.17: Region based quality map - holiday/leisure

**Evaluation of Places** In a second step, we are comparing the scores to a list of camping sites, resulting in a score of individual best locations. For this, the results for nodes which are closest to the places are extracted, as shown in Figure 7.17c. A gray marker denotes a camping site which does not fulfill at least one requirement, matching camping sites can be ranked by their score.

### 7.3.2 Living Quality

In a second case, we search for a living place with specific properties. The parameters are as shown in Figure 7.19a. For example, it is required to reach the next bakery within 15 minutes walking. In this case, the result benefits from the high precision of the model. Other aspects, as the post office, are required to be reached within 30 minutes by car but may be needed only a few times. Therefore, the weighting is low, and the result does not influence the overall score as much as, for example, the train station. In Figure 7.19b, an overview of the scores is given. In this case, especially requirements like the close train station exclude many rural areas. However, various smaller towns seem to fulfill the requirements.

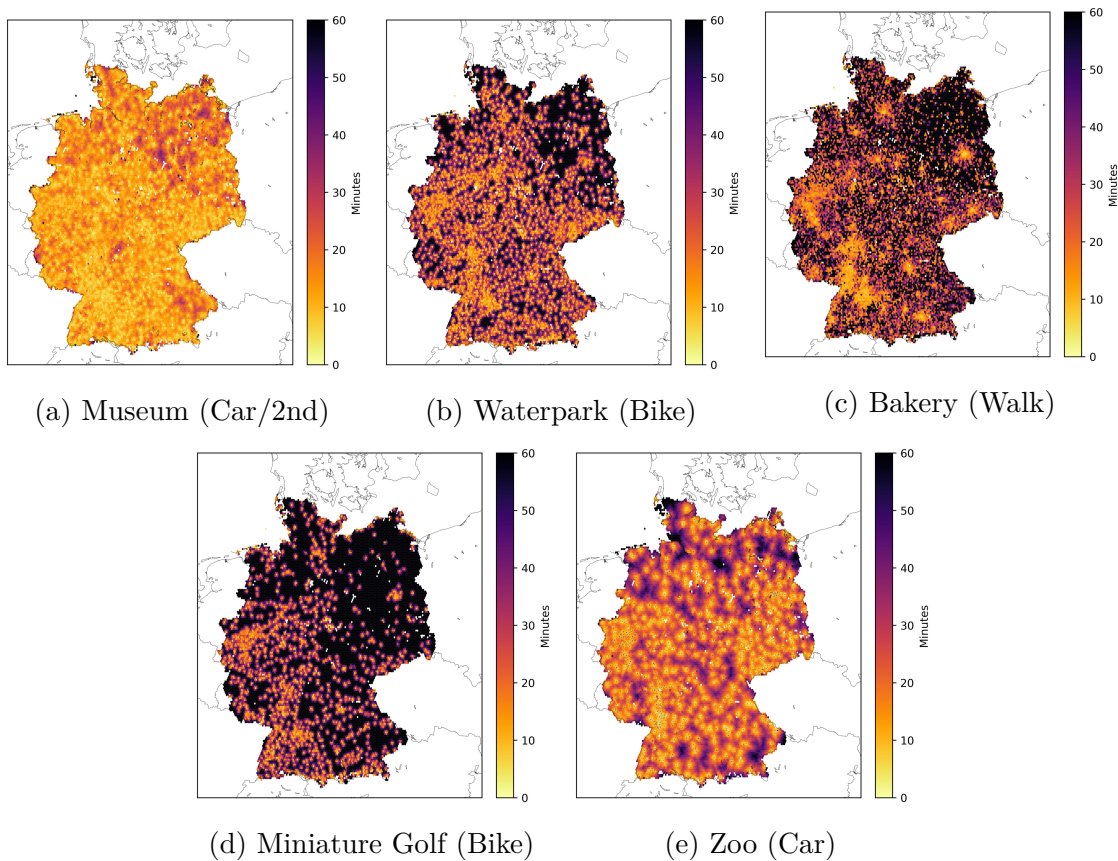
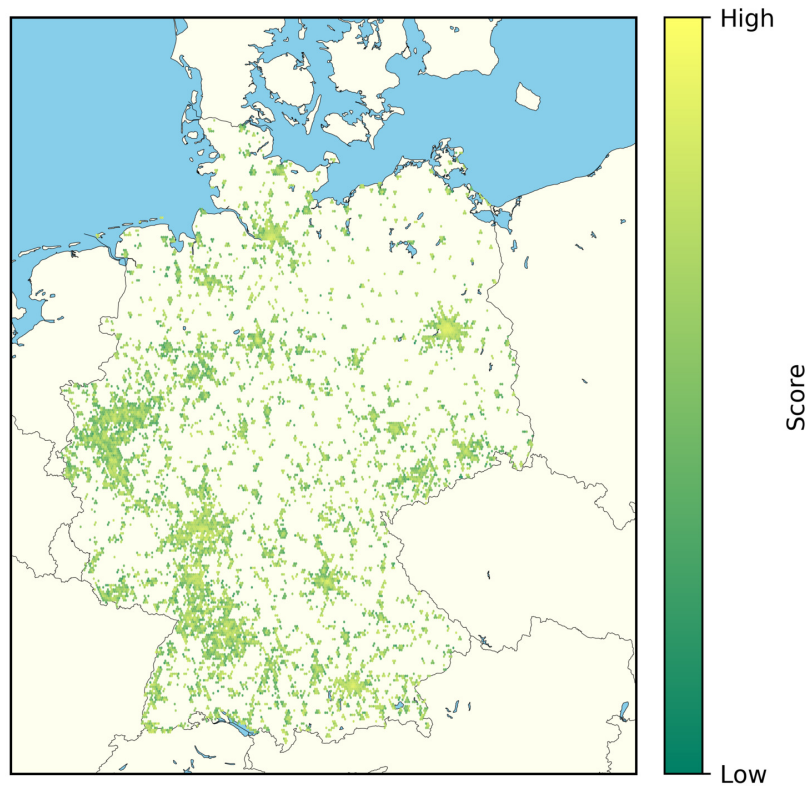


Figure 7.18: Separate maps for the different POI

POI	Model	weight	max time(hours)
Kindergarten	Walk	0.6	0.5
Train station	Bike	2	0.5
Bakery	Walk	1	0.25
Post office	Car	0.4	0.5
Supermarket	Car	0.5	1
Cinema	Car	1	1

(a) Parameters



(b) Overview living quality

Figure 7.19: Aggregated Living Quality Score

### 7.3.3 Discussion and Conclusions

The gained information could be used for various kinds of ranking of locations, like filtering real estate offers or choosing building grounds. This data can be used to estimate an overall living quality, merging several factors. Combining different transportation models allows adapting the used transportation methods for the individual set of POI. Also, it is possible to specify POI with an over regional coverage and others with a local impact so that maintaining coverage is possible either with some high capacity POI or several small ones.

In contrast to existing approaches, which combine a set of different POI, not only the number of reachable POI is determined, but also the needed time can be used and weighted. This way, more precise analyses are possible. Also, different movement models can be compared as needed. For the population estimation, the nodes in the standard car movement model are used, creating a more fine resolution as alternative approaches with grid cells.

Using this approach, it is also possible to create statistics about the connection between sets of POI. For example, what is the percentage of *LIDL discount supermarkets* where a *ALDI discount supermarkets* is reachable within 3 Minutes? (58.5%)

## 7.4 Scenario C: Location optimization

*In this scenario, it is shown that that accessibility analysis can also be used for localization optimization as described in Section 5.2.3*

### 7.4.1 Logistics Center

**Setup** In a fictional application, we are looking for a new logistics center, assuming that already five locations close too larger cities in Germany are existing. In this work, no external API for determining candidates for real estate or building ground is used, as the terms and conditions of the API holders may forbid storing information and publishing results based on their data. Instead, we are selecting 500 random locations for demonstration purposes. These represent building areas or already existing locations which could be extended for logistics purposes. First, we evaluate how many persons are covered by the existing locations within 90 Minutes as shown in Figure 7.20. The number of reached persons is *28,309,279*. The goal is to find a new location which maximizes the number of reached persons. Also, a set of other constraints should be fulfilled.

**Candidate Selection** To find the best candidates, we create a quality map as shown before. In this example, two constraints are used, the areas should be close to a highway entry, to allow easy transportation with semi-trucks. Also, the new location should be not too close to existing locations, this way candidates, which are, presumably, not increasing the accessed population, are discarded. The resulting map is shown in Figure 7.21a. Using this map, we select candidates for further evaluation (shown in Figure 7.21b).

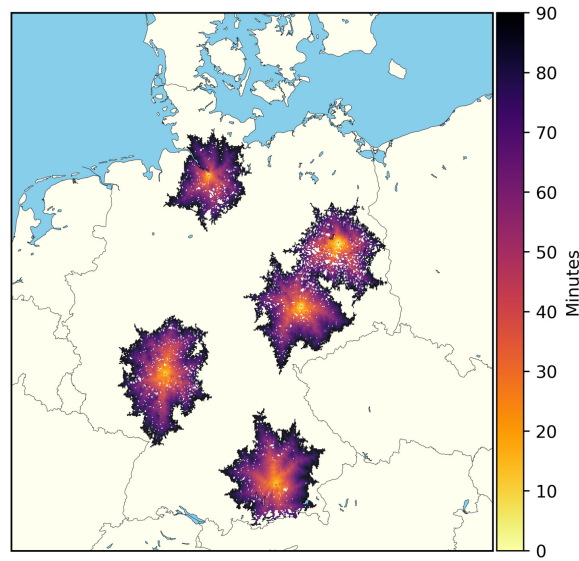
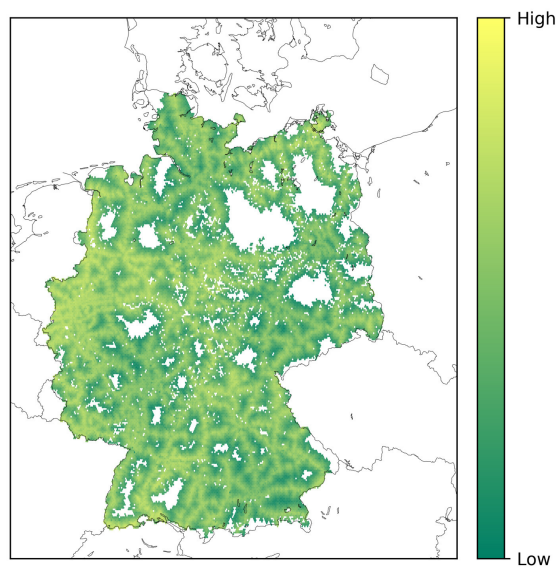
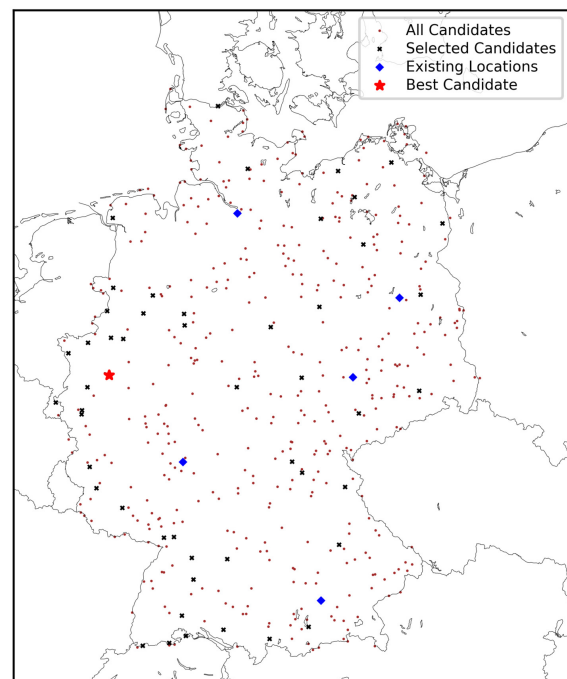


Figure 7.20: 90-minute isochrones of existing locations



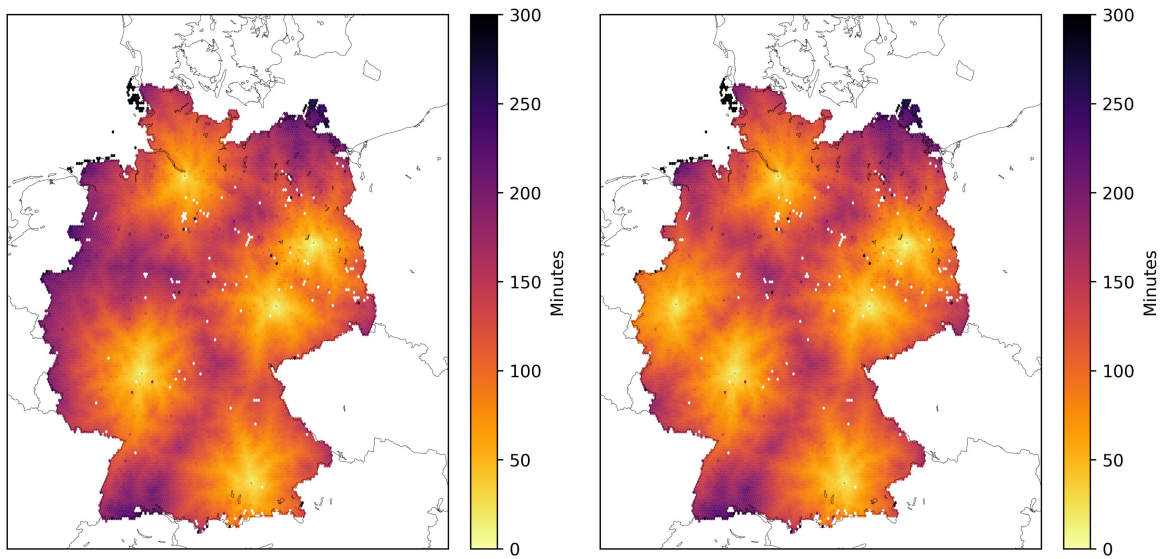
(a) Quality map - Scores for candidate selection



(b) Candidate selection

Figure 7.21: Logistic center optimization – finding candidates for further evaluation





(a) Accessibility – existing locations  
(28,309,279 persons reached in 90 minutes)

(b) Accessibility – including new location  
(42,357,065 persons reached in 90 minutes)

Persons reached by all POI	Position of new location (longitude, latitude)
42,357,065	7.061507, 51.342227
41,262,911	6.595874, 51.180353
40,798,258	7.10446, 51.843415
39,600,254	7.369327, 51.828023
39,376,994	6.608656, 51.777258

(c) Best results for optimization

Figure 7.22: Optimization Results

**Optimized Coverage** For each of the 50 selected candidate locations, we are performing an accessibility analysis and are comparing the total number of inhabitants reached by all locations in 90 minutes. The results of the best candidates are shown in Figure 7.22.

## 7.4.2 Stroke Service Vehicle

**Setup** In this second example, we try to find an optimal location for stroke service vehicles (“*Stroke-Einsatz-Mobil*”)<sup>2</sup> in Berlin, Germany. The fire brigade in Berlin currently has three special vehicles for mobile stroke care. As an example, we try to find the optimal location for a fourth vehicle, allowing to reach as many persons as possible within 10 minutes. As candidates, the 150 professional and voluntary fire stations are selected, which are fastest reachable from the center of Berlin.

**Results** 1,995,210 persons are reachable within 10 minutes using the current locations. As an optimal solution, the *Freiwillige Feuerwehr Prenzlauer Berg* was found, enabling the accessibility of a total of 2,693,689 persons in 10 minutes (second best solution was *Berliner Feuerwache Mitte*). The resulting isochrones are shown in Figure 7.23.

**Performance** On a conventional desktop PC, the evaluation of one set of four locations, the three existing locations and the candidate, needed a computation time of about 6 seconds. Since in the current brute-force strategy, all 150 combinations need to be computed. The total evaluation time was about 15 minutes for candidate evaluation.

<sup>2</sup><https://www.berliner-feuerwehr.de/technik/fahrzeuge/rettungsdienstfahrzeuge/stroke-einsatz-mobil/>

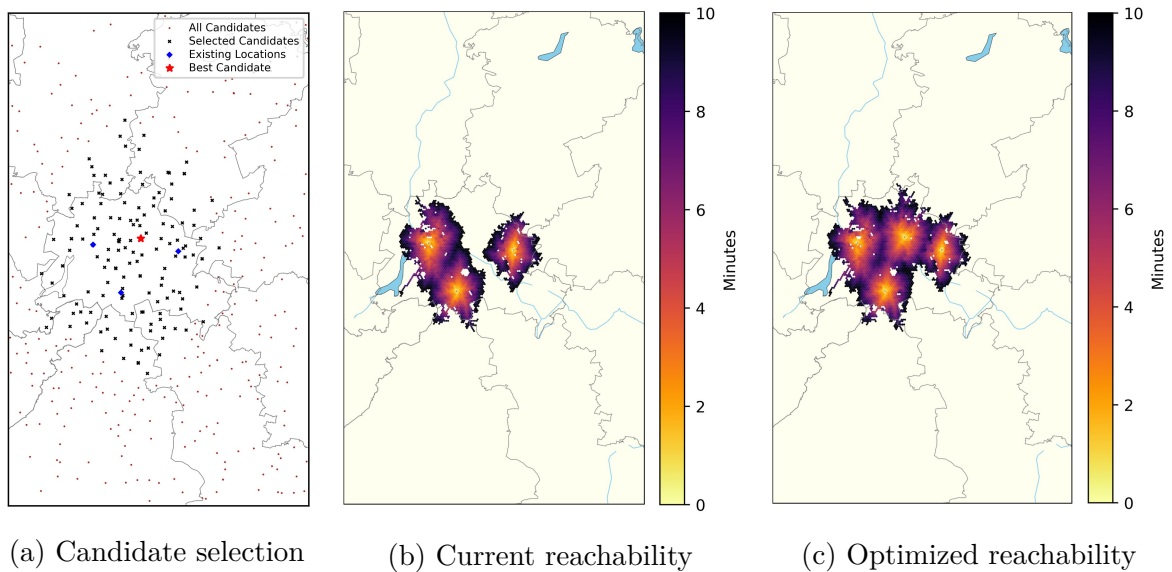


Figure 7.23: Optimization: Stroke Units Berlin - 10-minute reachability



### 7.4.3 Discussion and Conclusions

The localization optimization showed, in the here presented version, already satisfying results. In the candidate selection, as well as in the final evaluation, other metrics can be added to further improve the model like the costs for building grounds. Creating the quality map combines the purpose of fulfilling and ranking according to soft or hard constraints, as well decreasing the number of candidates which are close to existing locations. In this case, it has to be taken care that not too many candidates are discarded before final analysis. The results show that it is possible to find new locations. It was shown that this workflow can be used for reducing the number of candidates as well as, in a second step, measuring the quality of the locations. Its full potential is reached if the evaluation is extended by other datasets, as the choice of an optimal location does not only depend on the presented measurements.

For evaluating a large set of candidates, or while finding the optimum for placing multiple POI, powerful systems are needed, since an evaluation on large areas as Germany or whole Europe may require several minutes. For largely scaled analyses also the complexity of the graph could be reduced.

### Summary

*In this Chapter, the results of the thesis were presented and discussed. In the first section, the movement model was compared to navigation APIs, and examples of the population downscaling were given. In the following, the different use-cases were presented. The service coverage was evaluated for fire stations, showing that the coverage is heterogeneous in Germany but seems to fulfill the time limits requested by law. Other examples were given. Various aspects were demonstrated while investigating further the accessibility of hospitals, pharmacies, car charging stations and more. Results like the coverage of highways were shown as well as the creation of areas of responsibility. In the following, the combination of multiple analyses was shown on the example of holiday and living quality. Further, results for optimization problems were shown. With the example of logistics centers in Germany and stroke treatment in Berlin, it was demonstrated that well-suited places could be found.*

*It could be shown that the presented workflow is effective to address various subjects with increased precision, that previously had been addressed in individual solutions. Moreover is shown, that combinational analysis can be done to a level that has not been conducted before. Localization optimization shows a use-case where coverage and combinational analyses can be combined.*

*In the next Chapter, the overall workflow will be discussed, and a brief look at future work is given.*



# 8 Conclusion

## 8.1 Summary

In Chapters 2 and 3 the fundamentals were described, and an overview was given which approaches were used in related work. Different population datasets were investigated and compared. The GEOSTAT dataset was used and downscaled using buildings, and residential areas from the OSM dataset. All of data sources and software used in this work are publicly available and free of charge to use. Strategies for evaluation quality of POI data from the OSM dataset were sketched. Further, approaches for result presentation were tested and discussed.

Three use-cases were built, to demonstrate the possibilities of accessibility analyses. Determining service quality and coverage, combining different analyses and performing location optimization. A framework was built to fulfill the goals of this thesis:

- **Precise Resolution** - The results are provided on a per-node basis by using the Dijkstra algorithm and storing the path costs for each node in the graph
- **Continental Sized Analyses** - Splitting the area, allows to create fine-resolution analyses on a continental size, as only the region around each POI is loaded. Different areas can be processed in parallel
- **High Number of POI** - By splitting up the task, and setting multiple starting points in the search algorithm, it is possible to perform the analyses with several thousand POI
- **Maintainability** - The workflow has low requirements for the system, as no database or clustering system is required.

Separate movement models were created to estimate the needed times for different scenarios. Several experiments were performed to evaluate the presented approaches.

- **Coverage** - Different scenarios were evaluated. Determining accessibility of emergency services and other sets of POI. Results were compared to existing approaches.
- **Quality maps** - On two examples the combination of different analyses with distinct movement models was presented.
- **Optimization** - For localization optimization was shown that it is possible to use the analyses for candidate reduction as well as for determining the increase of coverage for each candidate.

## 8.2 Remarks

For performing an accessibility analysis, the fastest transportation time from a set of POI to many points must be computed. On the one hand, for precise analyses, a high resolution is necessary, as the complete street network needs to be resolved. On the other hand, for large-scale analyses, the computation time needs to remain manageable. Only when both goals can be achieved, the full potential of accessibility analyses is revealed, since local low-range aspects can be handled as well as over-regional aspects. Existing results could be verified and improved regarding resolution and customization.

The presented work shows that accessibility analyses are possible on open geospatial data by the OpenStreetMap dataset. A high resolution of the results on a street-precise level allowed eliminating outlines. For example, by using a gridding approach, it could be possible to pick a non-representative street for the cell, like a highway or a living street. Further datasets were preprocessed, to enrich the analyses. By using downscaling, the resolution of population grids was improved, allowing to handle the high precision of the results. By this new distribution of population, the estimated population for each node in the graph could be estimated. Existing approaches were using the population grids directly with their centroids or were intersecting isochrones with them.

The evaluation showed that results are similar but more precise as existing approaches, if given a sufficient quality of the POI dataset. By using the presented workflow, it is possible to perform an accessibility analysis on a country-sized dataset (Germany) with more than 25,000 POI while maintaining a high resolution.

The accessibility analyses workflow can also be used for more complex tasks, the combination of different models and sets of POI, as well localization optimization. A variety of analyses can be performed, providing crucial knowledge for local authorities, companies as well as other institutions and individuals.

## 8.3 Further Work

The following tasks and topics are of interest for further work:

- Computations are currently too slow using the full graph for cases like a web service. Several optimization techniques exist for navigation tasks and data processing. Evaluating which can be applied additionally, could further decrease the runtime. For example, techniques like hierarchical routing graphs or dynamic loading of the required areas. If only large-scaled aspects are investigated, it is possible to use a simplified graph, using only junctions and discarding residential streets. This way the complexity of the graph and therefore the required computation time would be reduced. Also, the usage of *PostGIS* for storing persistent data can be further evaluated.
- We are confident that this approach also works on continental-sized datasets. For this, further data preprocessing needs to be done to investigate differences in POI and the street network between different countries.

- In further work creating a public transport movement model is of particular interest as this is an essential kind of movement, especially in urban areas. This model would also allow comparison between car, bike and public transport reachability. As public transport networks are multi-modal, it is more complex and denotes an own field of research.
- Similar is true for traffic simulation, which has a substantial impact, for example, on commuter traffic. With increasing data quality, the movement models can be further developed, and the results can be improved. Other approaches to simulate the traffic or obtaining real-time data help to create a more realistic transportation time estimation.
- The refined population dataset showed good results. If a qualitative dataset with a smaller grid size is used, the precision would be further improved. These datasets determine the homes of the inhabitants. These do not equal the place where the population is at a usual working day. Industrial areas seem unpopulated while many persons are working there and for example, emergency services need to cover these places as well.
- Using the here presented workflow, a variety of new studies in the field of transportation, urban and spatial research is possible.



# Bibliography

- [Aur91] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 9 1991.
- [BBG<sup>+</sup>16a] Moritz Baum, Thomas Bläsius, Andreas Gemsa, Ignaz Rutter, and Franziska Wegner. Computing Minimum-Link Separating Polygons in Practice. *EuroCG*, 2016.
- [BBG<sup>+</sup>16b] Moritz Baum, Thomas Bläsius, Andreas Gemsa, Ignaz Rutter, and Franziska Wegner. Scalable Isocontour Visualization in Road Networks via Minimum-Link Paths. 2016.
- [BBHK06] Raymond C. Browning, Emily A. Baker, Jessica A. Herron, and Rodger Kram. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of Applied Physiology*, 100(2):390–398, 2 2006.
- [BDD<sup>+</sup>16] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub. The GeoJSON Format. Technical report, 8 2016.
- [BKSS90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R\*-tree: an efficient and robust access method for points and rectangles. *ACM SIGMOD Record*, 19(2):322–331, 1990.
- [BLM<sup>+</sup>14] Mohamed Bakillah, Steve Liang, Amin Mobasher, Jamal Jokar Arsanjani, and Alexander Zipf. Fine-resolution population mapping using OpenStreetMap points-of-interest. *International Journal of Geographical Information Science*, 28(9):1940–1963, 9 2014.
- [BNZ14] Christopher Barron, Pascal Neis, and Alexander Zipf. A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, 18(6):877–895, 12 2014.
- [Boe17] Geoff Boeing. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 9 2017.
- [Bou87] Paul Bourke. CONREC A Contouring Subroutine. 1987.
- [BW17] Samuel D. Blanchard and Paul Waddell. UrbanAccess. *Transportation Research Record: Journal of the Transportation Research Board*, 2653:35–44, 1 2017.

- [Cen17] Center for International Earth Science Information Network (CIESIN) Columbia University. Documentation for the Gridded Population of the World, Version 4 (GPWv4), Revision 10 Data Sets. Technical report, Center for International Earth Science Information Network (CIESIN) Columbia University, 2017.
- [Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 12 1959.
- [DKWG08] Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224–3236, 10 2008.
- [dL13] Norbert de Lange. *Geoinformatik in Theorie und Praxis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 3. edition, 2013.
- [DSSW09] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Engineering route planning algorithms. In *Algorithmics of Large and Complex Networks*, volume 5515 LNCS, pages 117–139. Springer, Berlin, Heidelberg, 2009.
- [ECCU] Joint Research Centre (JRC) European Commission and Center for International Earth Science Information Network CIESIN Columbia University. GHS population grid, derived from GPW4, multitemporal (1975, 1990, 2000, 2015).
- [EKS83] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 7 1983.
- [ESR98] ESRI. ESRI Shapefile Technical Description, 1998.
- [Eur14] European Forum for Geostatistics. Geostat 1b Final Report. Technical report, 2014.
- [GKD10] Tassilo Glander, Martin Kramer, and Jürgen Döllner. Erreichbarkeitskarten zur Visualisierung der Mobilitätsqualität im ÖPNV. *Kartographische Nachrichten*, 60(3):137–142, 2010.
- [Heg18] Jan-Erik Hegemann. So funktioniert Feuerwehr in Deutschland, 2 2018.
- [HR16] Stephan Huber and Cristoph Rust. Calculate Travel Time and Distance with OpenStreetMap Data using the Open Source Routing Machine (OSRM). *The Stata Journal*, 16(2):1–8, 1 2016.
- [HSV12] Henning Hollburg, Christoph Sinn, and Patrick Volland. Hier bin ich—was kann ich erreichen? Webbasierte, interaktive Erreichbarkeitsanalyse touristischer Ziele der Stadt Potsdam. In *Angewandte Geoinformatik 2012*, pages 317–322, 2012.



- [HSW09] Martin Holzer, Frank Schulz, and Dorothea Wagner. Engineering multi-level overlay graphs for shortest-path queries. *Journal of Experimental Algorithmics*, 13:2.5, 2 2009.
- [Jar73] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2(1):18–21, 3 1973.
- [JC14] S Jamtsho and R J Corner. Evaluation of spatial accessibility to primary healthcare using GIS. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-2:79–86, 2014.
- [KK98] Bernhard Koch and Beate Kuschinsky. Handbuch Rettungswesen - Die Hilfsfrist im Rettungsdienst in der präklinischen Notfallversorgung als Grundlage der rettungsdienstlichen Konzeption. *Handbuch des Rettungswesens*, pages 1–12, 1998.
- [Lan13] Mitchel Langford. An Evaluation of Small Area Population Estimation Techniques Using Open Access Ancillary Data. *Geographical Analysis*, 2013.
- [LST17] Christopher T. Lloyd, Alessandro Sorichetta, and Andrew J. Tatem. High resolution global gridded data for use in population studies. *Scientific Data*, 4:170001, 1 2017.
- [LWZ09] Jürgen Lerner, Dorothea Wagner, and Katharina Anna Zweig. *Algorithmics of large and complex networks. Design, analysis, and simulation*. Springer Berlin Heidelberg, 2009.
- [MG10] Sarunas Marciuska and Johann Gamper. Determining objects within isochrones in spatial network databases. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6295 LNCS, pages 392–405, 2010.
- [MS06] Adriano Moreira and Maribel Yasmina Santos. Concave Hull: A k-Nearest Neighbours Approach for The Computation of The Region Occupied By A Set of Points. *Proceedings of the 2nd International Conference on Computer Graphics Theory and Applications (GRAPP 2007), Barcelona, Spain*, pages 61–68, 2006.
- [NDZ07] P Neis, L Dietze, and A Zipf. A Web Accessibility Analysis Service based on the OpenLS Route Service. In *AGILE 2007. International Conference on Geographic Information Science of the Association of Geographic Information Laboratories for Europe*, pages 1–12, 2007.
- [Neu13] Stefan Neumeier. Modellierung der Erreichbarkeit öffentlicher Apotheken : Untersuchung zum regionalen Versorgungsgrad mit Dienstleistungen. 2013.

- [Neu14a] Stefan Neumeier. Modellierung der Erreichbarkeit von Straßentankstellen: Untersuchung zum regionalen Versorgungsgrad mit Dienstleistungen der Grundversorgung. 2014.
- [Neu14b] Stefan Neumeier. Open Source in accessibility analyses. *OSGeo Journal*, 13(1):90–102, 4 2014.
- [Neu15] Stefan Neumeier. Street petrol station shops as an alternative to missing local food suppliers - Contribution to the German discourse on 'Daseinsvorsorge': A German view. *Applied Geography*, 60:150–164, 2015.
- [Neu16] Stefan Neumeier. Regional Distribution of Ambulant Nursing Services in Germany. A GIS Accessibility Analysis in the German Social Code Book XI · Raster-based GIS-accessibility analysis. *Raumforschung und Raumordnung*, 74:339–359, 2016.
- [NZ04] Pascal Neis and Alexander Zipf. LBS\_2.0- Realisierung von Location Based Services mit user-generated, collaborative erhobenen freien Geodata. *Information Systems*, page 8, 2004.
- [NZ08] Pascal Neis and Alexander Zipf. Zur kopplung von opensource, opens und openstreetmaps in openrouteservice. org. *AGIT 2008 - Symposium für angewandte Geoinformatik*, 2008.
- [NZZ11] Pascal Neis, Dennis Zielstra, and Alexander Zipf. The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011. *Future Internet*, 4(4):1–21, 12 2011.
- [PG17] Marcus Peter and Carsten Gertz. Leitprojekt Regionale Erreichbarkeitsanalysen Abschlussbericht und Erreichbarkeitsatlas, 2017.
- [RR16] Sebastian Rauch and Jürgen Rauh. Verfahren der GIS-Modellierung von Erreichbarkeiten für Schlaganfallversorgungszentren. *Raumforschung und Raumordnung*, 74(74):437–450, 10 2016.
- [SAM13] Johannes Scholz, Michael Andorfer, and Manfred Mittlboeck. Spatial accuracy evaluation of population density grid disaggregations with corine landcover. In *Geographic Information Science at the Heart of Europe. Lecture Notes in Geoinformation and Cartography*, volume 2013-Janua, pages 267–283. Springer, Cham, 2013.
- [SBB12] Mark Steinvoord, Peter Berger, and Daniel Batlle. *GIS-gestützte Analyse von Fahrgeschwindigkeiten unter Sonder- und Wegerecht*. PhD thesis, Hochschule für Angewandte Wissenschaften Hamburg, 2012.

- [SBK16] C. Seip, R. Bill, and A. Kinskofer. Notfallversorgung der Bevölkerung in Mecklenburg-Vorpommern: Erreichbarkeitsanalysen auf Grundlage von OpenStreetMap und ArcGIS Online. *Notfall und Rettungsmedizin*, 19(2):99–107, 2016.



# List of Figures

4.1	Population of Hamburg . . . . .	33
a	GPWv4 . . . . .	33
b	GHS . . . . .	33
c	GEOSTAT . . . . .	33
d	GEOSTAT - Refined in this work . . . . .	33
e	Census 2011 . . . . .	33
4.2	Different 30-minutes isochrones . . . . .	36
a	Mapbox . . . . .	36
b	OpenRouteService . . . . .	36
c	TravelTimePlatform . . . . .	36
d	Iso4App . . . . .	36
e	This work (alpha shapes) . . . . .	36
f	This work (hexbin) . . . . .	36
5.1	Overview of the presented scenarios . . . . .	41
5.2	Workflow for determining service coverage . . . . .	42
5.3	Creating a <i>Quality Map</i> . . . . .	43
5.4	Optimization of facility placement . . . . .	44
5.5	Steps for data preparation . . . . .	47
5.6	Steps for population disaggregation . . . . .	52
7.1	Comparing results with routing APIs . . . . .	65
a	Car model (afternoon) <i>Google Directions</i> . . . . .	65
b	Car model <i>OSRM</i> . . . . .	65
c	Walking <i>Google Directions</i> . . . . .	65
7.2	Population data downscaling . . . . .	67
a	Hamburg . . . . .	67
b	Halstenbek (Hamburg suburb) . . . . .	67
c	Hamburg (inner city) . . . . .	67
d	Stellingen, Hamburg . . . . .	67
7.3	Fire departments in Germany, each blue marker represents a fire station .	69
7.4	Fire station coverage in Germany . . . . .	69
a	Average time . . . . .	69
b	Average time per district . . . . .	69
7.5	Fire station coverage with and without special privileges . . . . .	70
a	8-minute coverage (Special rights) . . . . .	70
b	10-minute coverage (Special rights) . . . . .	70

c	15-minute coverage (Special rights) . . . . .	70
d	8-minute coverage (No special rights) . . . . .	70
e	10-minute coverage (No special rights) . . . . .	70
f	15-minute coverage (No special rights) . . . . .	70
7.6	Fire station coverage, unavailability of fastest station . . . . .	72
a	Fastest - average . . . . .	72
b	Second - average . . . . .	72
c	Third - average . . . . .	72
d	15-minute coverage first . . . . .	72
e	15-minute coverage second . . . . .	72
f	15-minute coverage third . . . . .	72
7.7	Comparing Voronoi to accessibility analyses - Persons per station (log. scale) . . . . .	72
7.8	Highway coverage in Germany of fire stations . . . . .	73
a	10-minute coverage in Germany . . . . .	73
b	Different directions . . . . .	73
c	Highway coverage . . . . .	73
7.9	Fire stations: Comparing movement model to linear distance . . . . .	74
7.10	Areas of responsibility - fire stations . . . . .	75
a	Accessibility analyses . . . . .	75
b	Voronoi . . . . .	75
7.11	Fire station coverage in Schleswig-Holstein, Germany . . . . .	76
a	Worst covered areas (8 Min.) – Impact of missing stations in the dataset . . . . .	76
b	Histogram of areas reaching given coverage . . . . .	76
c	8-minute coverage 1st run . . . . .	76
d	8-minute coverage 2nd run . . . . .	76
e	8-minute coverage – Diff . . . . .	76
7.12	Needed time to the closest hospital . . . . .	78
7.13	Hospitals: Optimal areas of responsibility in northern Germany . . . . .	78
a	Voronoi . . . . .	78
b	Accessibility analysis . . . . .	78
7.14	Coverage of electric car charging stations in Germany . . . . .	79
a	Average . . . . .	79
b	Average per district . . . . .	79
c	Coverage (25km) . . . . .	79
d	Highway Coverage 25 km . . . . .	79
e	Coverage . . . . .	79
7.15	Coverage of stroke units . . . . .	80
a	Overview . . . . .	80
b	30-minute coverage . . . . .	80
7.16	Driving time to the next pharmacy . . . . .	82
a	Overview ([PG17]) . . . . .	82
b	Overview (accessibility analysis) . . . . .	82

c	Comparing on detail-level ( <i>Pinneberg</i> )	
	Squares by Peter [PG17], Points by this work . . . . .	82
7.17	Region based quality map - holiday/leisure . . . . .	84
a	Quality map . . . . .	84
b	Average score per postal code area . . . . .	84
c	Score of camping sites . . . . .	84
7.18	Separate maps for the different POI . . . . .	85
a	Museum (Car/2nd) . . . . .	85
b	Waterpark (Bike) . . . . .	85
c	Bakery (Walk) . . . . .	85
d	Miniature Golf (Bike) . . . . .	85
e	Zoo (Car) . . . . .	85
7.19	Aggregated Living Quality Score . . . . .	86
a	Parameters . . . . .	86
b	Overview living quality . . . . .	86
7.20	90-minute isochrones of existing locations . . . . .	88
7.21	Logistic center optimization – finding candidates for further evaluation . . . . .	88
a	Quality Map . . . . .	88
b	Candidate selection . . . . .	88
7.22	Optimization Results . . . . .	89
a	Accessibility – existing locations . . . . .	89
b	Accessibility – including new location . . . . .	89
c	Best results for optimization . . . . .	89
7.23	Optimization: Stroke Units Berlin - 10-minute reachability . . . . .	90
a	Candidate selection . . . . .	90
b	Current reachability . . . . .	90
c	Optimized reachability . . . . .	90





# List of Tables

3.1	Routing API solutions (extraction) . . . . .	29
7.1	Fire station coverage in Germany — Data sources and settings . . . . .	68
7.2	Best and worst results - fire stations . . . . .	71
7.3	Hospital coverage — Data sources and settings . . . . .	77
7.4	Electric car charging — Data sources and settings . . . . .	77
7.5	Stroke units — Data sources and settings . . . . .	80
7.6	Pharmacies — Data sources and settings . . . . .	81
7.7	Requests for the holiday location . . . . .	83



## Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Studiengang M.Sc. Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

---

Ort, Datum

---

Unterschrift

## Veröffentlichung

Ich bin damit einverstanden, dass meine Arbeit in den Bestand der Bibliothek des Fachbereichs Informatik eingestellt wird.

---

Ort, Datum

---

Unterschrift