

INAUGURAL - DISSERTATION
zur Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Johannes Lauer, M.Sc.
aus Wadern

Tag der mündlichen Prüfung:
17. Juli 2017

THEMA

Extraction of routing relevant geodata
using telemetry sensor data
of agricultural vehicles

Gutachter: Prof. Dr. Alexander Zipf
Ruprecht-Karls-Universität Heidelberg

Prof. Dr.-Ing. Stefan Böttinger
Universität Hohenheim

Acknowledgements

First of all I want to thank my parents for their support that allows me to follow my ideas and visions. Without the practical work on my parents farm it would not be possible to combine this practical knowledge with the theoretical academic work in such a project.

I would like to thank my supervisor Prof. Dr. Alexander Zipf, who gave me the opportunity to write this thesis in his group and for the freedom to work on this topic. Additionally, I also want to thank Prof. Dr.-Ing. Stefan Böttinger for taking over the role of a co-supervisor and his feedback from the perspective of agricultural engineering.

Thanks a lot to my friends that brought the right compensation through several bike and climbing trips. Especially Hannah for many cooking sessions and her proof reading just before her first of seven summits. Further thanks to Barbara and Kristen for proof reading and some hints from the view of a native speaker and to Matthias for his comments from a technical perspective.

I want to thank the farmers, who gave their permission to make use of their data. Especially Helmut Thurmeier and Michael Kett who also gave their support for the field-work and the Woeda Agrar GmbH for providing their field reference data. Thanks to the CLAAS E-Systems team (André Kluge, Markus Kuhlmann, Yaron Engel and Stephan Weller) for the great collaboration and the open minded brain storming sessions within the TeleAgro+ project. Furthermore, I dedicate my thanks to the Ministry of Food and Agriculture BMEL for the financial support and the project execution organization BLE for the uncomplicated project management of the TeleAgro+ project.

I would like to thank the students (Ludwig Richter, Hans Schell, Sen Sun, Achim Fürsich and Markus Döring) and colleagues (Harsha Vemu and Tim Ellersiek) within the TeleAgro+ Project which always gave their best and enriched the project with their work. Many of them got in touch with agricultural technology in this depth for the first time and the work in this project opened their mind for a broader context.

Big thanks to my colleagues of the GIScience and LiDAR research group. In particular Bernhard Höfle for his scientific support and the fruitful methodological discussions, Lukas Loos for proof reading and technical out of the box discussions. I am keeping my fingers crossed for your way to the PhD finish line. Thanks to Tobias Toernros for his great support on R programming and thanks to Adam Rousell for proof reading and his hints as a native speaker. Many thanks to the whole GIScience group that came from Bonn University to Heidelberg University and managed the move and the setup of a fast growing working group which was not always easy.

Also thanks to my colleagues from HERE where I started my new job. Especially Matthias Bobzien for proof reading and his help with \LaTeX , Arvind from the HERE Geocoder team for spell checking from a native speakers perspective and in general to the whole team for fruitful discussions and comments through their perspective on this work.

Special thanks to the OpenStreetMap community for their great work on an amazing geographical data set. I hope the success story of volunteered geographic information will proceed and many applications will benefit from crowd sourced data.

And last but not least to my new born nephew and godson Jonas Emil. You gave the final motivation to finish the thesis!

Kurzfassung

Die Mechanisierung im Agrarsektor ist innerhalb der letzten hundert Jahre stetig gewachsen. Innerhalb der letzten Jahrzehnte hielt die Informationstechnologie Einzug in diesen Sektor und Daten sind mittlerweile ein entscheidender Faktor für die Prozessoptimierung. Zur Planung und Durchführung von Logistikprozessen und Erntekampagnen sind digitale Straßenkarten und Feldgrenzen essentiell. Sie sind Voraussetzung, um bei fehlender Ortskenntnis der Fahrer und Planer die großen Maschinen zum Feld zu führen und diese Prozesse zu verbessern. Mit der Verfügbarkeit von Fahrzeugtelemetrie, den Bewegungsdaten einzelner Maschinen, entstehen neue Möglichkeiten zur Generierung der benötigten Daten. Die Gewinnung geographischer Daten aus Bewegungsdaten ist einer der Schwerpunkte dieser Arbeit. Zu Beginn werden Verfahren zur Bereinigung der Eingangsdaten untersucht und weiter Vorprozessierungsschritte durchgeführt. Über Klassifikationsverfahren werden die Bewegungsdaten zu verschiedenen Arbeitsmodi zugeordnet. Basierend auf diesen Daten werden Algorithmen zur Generierung geographischer Features, wie Feldgrenzen analysiert und verbessert. Um die Qualität der erzeugten Geometrien sicher zu stellen, wird die Jaccard-Distanz als Metrik eingeführt. Mit den klassifizierten Straßen-Messpunkten werden mit verschiedenen Algorithmen ländliche Straßen- und Wegenetze erzeugt und deren Resultate gegenübergestellt. In einem dritten Schritt wird die Nutzbarkeit von Volunteered Geographic Information (VGI) für die Nutzbarkeit zur Routenplanung für landwirtschaftliche Fahrzeuge betrachtet. Da die Wegführung für landwirtschaftliche Maschinen nicht an der Feldgrenze endet, werden zudem Methoden zur Berechnung von feldinternen Graphen untersucht. Die einzelnen Komponenten stellen somit die erforderlichen Daten und Dienste für den Use-Case landwirtschaftliche Routenplanung innerhalb eines Frameworks bereit. Der abschließend vorgestellte webbasierte Routingdienst demonstriert das Zusammenspiel aller Komponenten und ermöglicht eine lückenlose Routenplanung von Hof zu Feld und innerhalb des Feldes.

Abstract

The mechanization of processes in agriculture is growing within the last hundred years. Within the last decades, the information technology in this sector constantly grew and data is one of the key factors for process optimization. For planning and execution of logistic processes and harvest campaigns, road maps and field geometries are essential to guide the large vehicles to their work places and to optimize harvest chains. Through nowadays available telemetry data of these vehicles, a new data source generates new opportunities. Mining geographic data from these movement data, that can improve agricultural work processes is one of the main objectives of this thesis. As a first step, data cleaning processes, and further preprocessing steps are shown. With classification algorithms, the continuous movement data will be separated into different work processes. Based on this data, algorithms to generate geographic features, such as field boundaries have been analyzed and improved. As quality metric to compare the results, the Jaccard-Distance has been established. With the classified road representing measurements, the rural road networks were created and the results of different algorithmic approaches have been compared. The usability of volunteered geographic information to route the heterogeneous set of agricultural vehicles is shown in a third step. Due to the fact, that routes for e.g. harvesters are not ending at the field boundary, solutions for infield route graph generation have been given. The presented components provide the content and the services within a framework structure. The concluding prototype, a web based routing system demonstrates the interaction of all components and provides a consecutive routing from farm to field and within the field.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Introduction and Motivation | 1 |
| 1.2 | State of the Art | 5 |
| 1.3 | Research Questions | 7 |
| 1.3.1 | How can agricultural movement data be classified to have distinct input data for field and road generation? | 7 |
| 1.3.2 | How can agricultural field boundaries be extracted from movement trajectories? | 7 |
| 1.3.3 | How can an infield routing graph be extracted from field geometries? | 7 |
| 1.3.4 | How can a road network be extracted from agricultural telemetry data? | 7 |
| 1.3.5 | Can OpenStreetMap data as VGI be a source for vehicle specific route planning in agriculture? | 7 |
| 2 | Used Datasets and TeleAgro+ Framework | 9 |
| 2.1 | Agricultural Telematics Data | 9 |
| 2.1.1 | Telematics in Agriculture | 9 |
| 2.1.2 | CLAAS Telematics | 10 |
| 2.2 | OpenStreetMap | 14 |
| 2.2.1 | Project Description | 14 |
| 2.3 | TeleAgro+ Framework | 16 |
| 2.3.1 | Framework Overview | 16 |
| 2.3.2 | Processes and Services | 17 |
| 2.3.3 | Trajectory Processor | 17 |
| 3 | Data Preprocessing | 21 |
| 3.1 | Data cleaning | 21 |
| 3.2 | Map Matching | 23 |
| 3.2.1 | Method | 23 |
| 3.2.2 | Map Matching Results | 24 |
| 3.3 | Average Speeds for Road Types | 26 |
| 3.4 | Extended Attribute Calculation | 27 |
| 3.5 | Measurement Classification | 27 |
| 3.5.1 | Reference Data | 30 |
| 3.5.2 | Classification Algorithm - KNN | 31 |
| 3.5.3 | Classification Algorithm - Random Forest | 32 |
| 3.5.4 | Post Classification | 43 |
| 3.5.5 | Track Segmentation | 44 |

| | | |
|----------|--|-----------|
| 3.6 | Estimating Main Working Direction | 44 |
| 3.6.1 | Statistical Calculation of the Main Working Direction | 45 |
| 3.7 | Discussion | 47 |
| 4 | Generation of Road Network using agricultural Telemetry Data - Geometric refinement | 49 |
| 4.1 | State of the Art | 49 |
| 4.2 | Road generation Algorithms in Detail | 53 |
| 4.2.1 | Edelkamp and Schroedl - cluster based | 53 |
| 4.2.2 | Cao and Krumm - trace merging | 56 |
| 4.2.3 | Morris et al. - Graph Reduction | 58 |
| 4.2.4 | Ahmed and Wenk - Network generation using Fréchet distance | 61 |
| 4.3 | Application on Agricultural Telemetry Data | 62 |
| 4.3.1 | Edelkamp and Schroedl | 62 |
| 4.3.2 | Morris et al. | 63 |
| 4.3.3 | Ahmed and Wenk | 63 |
| 4.3.4 | Cao and Krumm | 64 |
| 4.3.5 | Comparing methods | 65 |
| 4.3.6 | Algorithmic evaluation | 68 |
| 4.4 | Summary and future directions | 68 |
| 5 | Field Boundary Computation | 70 |
| 5.1 | Introduction and Motivation | 70 |
| 5.2 | Field Boundary Computation | 71 |
| 5.2.1 | State of the Art | 71 |
| 5.2.2 | Definition of Field Boundary | 72 |
| 5.3 | Field Reference Data | 73 |
| 5.4 | Computation Methods | 75 |
| 5.4.1 | Alpha-Shapes | 75 |
| 5.4.2 | Raster approach | 76 |
| 5.4.3 | Blow Shrink | 77 |
| 5.4.4 | Postprocessing - Optimization | 82 |
| 5.4.5 | Comparison of the presented approaches | 83 |
| 5.4.6 | Conclusion | 88 |
| 5.5 | Field Gateway Computation | 90 |
| 5.5.1 | Geometric calculation of Field Gateways | 90 |
| 5.6 | Integration in the Processing Framework | 91 |
| 5.7 | Results | 92 |
| 6 | Agricultural Routeplanning | 94 |
| 6.1 | Introduction and State of the Art | 94 |
| 6.1.1 | Routable Road Data | 94 |
| 6.1.2 | Computation and Data Handling Complexity | 97 |
| 6.1.3 | Edge Weight Calculation | 97 |
| 6.1.4 | Infield Routing | 97 |
| 6.2 | VGI data for Agricultural Route Planning - OSM | 98 |
| 6.2.1 | OSM Road Types | 98 |
| 6.2.2 | Relevant OSM Road Attributes | 99 |

| | | |
|----------|--|------------|
| 6.3 | Route Calculation with Multiple Attributes | 99 |
| 6.3.1 | Memory efficient Graph Structure | 100 |
| 6.4 | Cost Function with Empirical Edge Weights from Real Data | 101 |
| 6.4.1 | Calculation of Edge Weights | 102 |
| 6.4.2 | Vehicle Data Integration | 104 |
| 6.5 | Infield Routing Graph generation | 104 |
| 6.5.1 | Rectangular Grid Method | 104 |
| 6.5.2 | Oriented Diamond Grid Method | 105 |
| 6.5.3 | Delaunay Method | 106 |
| 6.5.4 | Voronoi Method | 106 |
| 6.5.5 | Line of Sight Method - Visibility Graph | 107 |
| 6.5.6 | Straight Skeleton Method | 108 |
| 6.5.7 | Method Comparison | 108 |
| 6.6 | Integration into MARS Routing | 109 |
| 6.6.1 | OpenLS Extension for Routing heterogeneous Vehicles | 109 |
| 6.6.2 | Integration of Field Boundaries and Gateways | 112 |
| 6.7 | Comparison between driven and calculated routes | 112 |
| 6.8 | Conclusions | 114 |
| 7 | Conclusions | 117 |
| 7.1 | Summary | 117 |
| 7.2 | Discussion of the Methodology | 119 |
| 7.3 | Constraints of the presented Approach and resulting further Research Questions | 121 |
| 7.4 | Transferability on other Domains | 123 |
| 7.5 | Outlook | 123 |
| | List of Figures | 125 |
| | List of Tables | 129 |
| | References | 131 |

Chapter 1

Introduction

1.1 Introduction and Motivation

I grew up in the early 80ties on a small farm that produced several kind of crops, oil seeds and grassland. An average mid European part-time farm with grassland and cropland. The whole work focused on some weeks of the year, the harvest period. Peak workload came during the harvest period, where everyone had to participate. From cutting the grass over teddering, raking and baling the grass, to collecting and bringing the bales to the shed, the process took several days. Especially the removal of the yield from field needed several people. During these regular processes, almost all involved people were locals and had the local knowledge how to get to the fields. In case the machine driver was not aware of this, one involved person knew where tractor and bale trailer could pass onto the field. Most of the helpers had local knowledge and finding the correct routes to the fields was no problem. However, this was usually the most stressful period since most of the income depends on the quality of the yield. Especially in the mid Europe weather conditions resulting in time windows that are usually short and volatile. In recent years, the agricultural machines became larger and fields rearranged to make work processes more efficient. The yield time can be reduced and the dependency on weather conditions and harvesting windows decreased. The machine size and the technological inventions increase investments. This made large farms more profitable and lead to a structural change. For smaller farms, a big, expensive harvester is not cost efficient. The number of contractors offering harvest services increased and machinery rings (syndicates for machine sharing) offer bigger, more expensive machinery to their community, to increase the capacity utilization. A set of contractors now serve multiple clients and work on a large number of agricultural fields. The bigger machinery increase the requirement for passable roads. In many cases, foreign contractors do not have the needed up-to-date local knowledge required to adapt to the continuously changing agricultural roads and fields (e.g. changed field boundaries, new fields). A common way is to assists the machine driver and guide him to the fields. For small logistic chains this is also a feasible way nowadays. Regarding complex harvesting chains with several harvesters and transportation vehicles, the disposition of the campaign is very complicated and often needs a computational solution to optimize it. Finding the route to the field where the whole logistic chain can operate under time pressure in an optimum manner is one of the key factors for success. Regarding large agribusiness companies, this knowledge is essential to plan and manage field operations.

Digital data as essential input for computerization is one of the main accelerators during the last decades. Process automation and optimization in commercial and public or

private domains are dependent mainly on digital data and algorithms that extract valuable knowledge from this data. Most of the data has a strong correlation to location, which is pointed out in talks and publications (cf. Hahmann and Burghardt [2013]) including the vision of a “Digital Earth”, a theory put forth by the former Vice President of the United States Al Gore (cf. Gore [1998]) which is becoming reality and with that the relevance of digital geographic data increases.

The domain of agriculture has faced many challenges within the last decades. This sector has evolutionized mechanization and digitization in the last years. Due to the fact, that global population has increased and will continuously increase within the next decades (cf. figure 1.1), distribution and efficient food production and food security will be key factors of global society. Technical improvements in agricultural machinery, increase of efficiency in harvesting processes and logistics will be main factors managing upcoming demands. Treating sustainable resources responsibly is imperative to manage a growing world population and preserving the world for future generations.

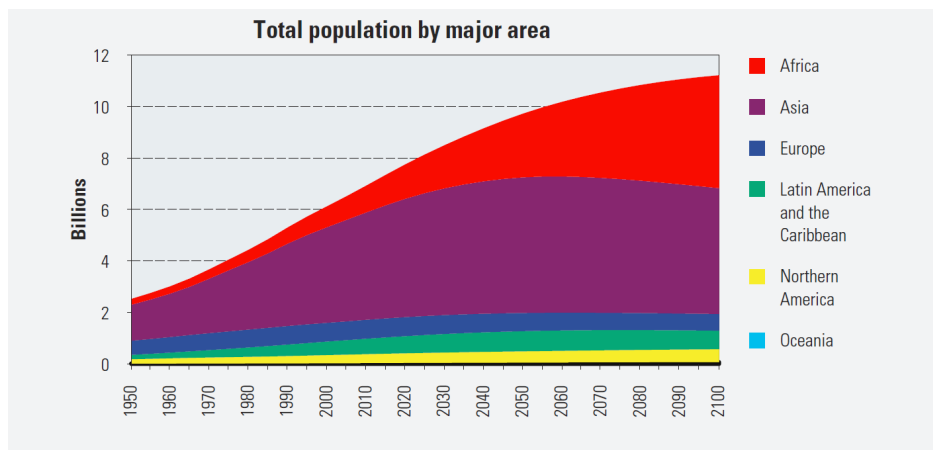


Figure 1.1: Total population by major area
(source: United Nations, 2015, https://esa.un.org/unpd/wpp/Publications/Files/World_Population_2015_Wallchart.pdf, (accessed 2016/07/16))

Accurate and up-to-date geospatial data is needed to address the challenges. This data will be the input for problem solving algorithms that have also being analyzed, extended or developed. Even for global commercial data providers such as HERE¹ or TOMTOM² with their high capacity of mappers, it is not possible to map the huge volume of tracks that are (in most cases) only accessible for agricultural vehicles. Continuous changes in these road networks and the need of further surface attributes make it impossible to collect data in conventional ways (e.g. through data imports, collection by mapping cars and manual mapping). Projects like OpenStreetMap³ established a new way of thinking about maps and geographic data. Through this crowdsourcing approach, missing geographic data or map errors can be detected by the community and not only by a limited group of professional mappers.

This approach is more cost efficient for current and created spatial map updates, especially in cases where smaller updates are spatially widely distributed. The community

¹<https://company.here.com/here/>, (accessed 2016/07/16)

²<http://tomtom.com/>, (accessed 2016/07/16)

³<http://www.openstreetmap.org>, (accessed 2016/07/16)

mapping approach has been adopted by commercial mapping companies in their products as a further data source (e.g. HERE MapCreator ⁴ or TOMTOM MapShare). However, this cannot solve the problem of mapping agriculturally relevant roads and geo features. Generating a digital earth needs much more data. An approach with only professional mappers, which is only based on manual work (e.g. surveying and digitizing data) will never be able to generate the digital earth and keep it up-to-date. Furthermore, the unequal distribution of human mappers tends to result in more current map updates in regions with larger population density (e.g. urban regions) while the larger and, for agricultural production, more important area is in rural regions with far less population and potential mappers. The current trend of people spending their life in urban areas and the decreasing rural population is also increasing the disparity in mapper distribution. Another reason for a lesser mapping activity is the (personal) value and the legal restrictions of the required geodata. Road data, such as geometries, surface type, incline and measures (e.g. width, clear height) are parameters that are useful for many applications. The mapping of this data has value for the mapper community itself, as these people are generally not from the agricultural domain. Route planning for sport activities, walks or emergency transports are obvious use cases that benefits from this data. In these cases, a cross domain requirement exists and a larger community will be available as potential data makers - a win-win situation. The more specific the geographic features, the less people are motivated to map it. Field boundaries are very important geographic features within the agricultural domain that only have a value for farmers, for contractors to plan the needed logistics, and for the service engineers that drive to the field for machine repairs. The government also has an interest in field boundaries to determine agricultural subsidies and structure. Field boundaries usually represent the farmed field area and are part of the operating record of each farm or agrarian company. The field boundary data can therefore underlay legal restrictions. Nonetheless, the availability of accurate digital field data will have a large impact on this domain and especially the single user in terms of productivity.

From a logistics perspective, navigation to the field is only one part. In a typical harvesting session (e.g. for a transfer meeting point within a harvesting process) the destination for the transport vehicle is located in the field. The other part is the guidance of the transport vehicle from the boundary to the meeting point in the safest and fastest way. From data perspective this needs an infield routing solution which should be connected to the surrounding road graph.

Smart phones and tablets are essential communication systems in the agricultural domain. Planning, disposition and urgent repair calls are only some use cases where these communication devices are indispensable. Through interfaces to machine bus, these devices can also be set as user interfaces for machine data. Due to the fact, that all of these smart devices are fully equipped with sensors (including a positioning sensor, such as GPS), they can be a further data source for location data. However, the heterogeneity of hardware and applications create huge efforts in terms of data pre-processing. In addition, there is no homogeneous infrastructure for data storing and data access.

In recent years, farming machinery has become technically advanced. Harvesters and tractors are equipped with a large number of sensors that measure machinery performance and give information on working processes. Guidance systems have also quickly developed from simple electronic guidance systems for parallel driving to modern automatic steering

⁴<http://mapcreator.here.com/>, (accessed 2016/07/16)

systems based on highly accurate positional data are now common tools in the agricultural sector. Telemetry systems collect data and store it in data bases used for further analysis. Figure 1.2 shows an example of a driver's cab: The modern cabin of a CLAAS AXION 900 tractor has terminals for parallel driving and sensor information on the right.



Figure 1.2: Cabin of a CLAAS AXION 900 (source: CLAAS - <http://www.claas-media-database.com>)

With the information of the data producer (e.g. machine type) and the distinct information of each machine, this data is a promising source for extraction of geographic features. The main shortcomings lie in the missing or not up-to-date field geometries and gateways where machines can pass from the road network onto the fields, road geometry refinement and the creation of a tagged routing graph for agricultural applications. As a precondition, a data classification step is necessary. This step is needed to provide distinct data for the specialized algorithms and use-cases. The previously mentioned legal problems and privacy issues are handled by the telemetry providers. They store the data according to the agreement with their clients. The overlap between the group of data producers and the group of data consumers is obviously large. Companies that collect data by tracking their vehicles later will have the benefit to use the extracted data for their farm management system or for routing their vehicles (fleet management) or the vehicles of authorized contractors. Therefore, domain related data will be produced in large parts from their future users. The overall challenge is the knowledge extraction from the collected data and the algorithmic computation of the needed data from this raw telemetry data.

Knowledge extraction from sensor data is one of the most recent challenges in the field of data mining. Due to the fact, that the proprietary sensor data is not available to a broad community (compared to GPS libraries, such as GPSies⁵ or GPS trajectories from OpenStreetMap⁶) a data analysis for these use cases has not been achieved yet.

⁵<http://www.gpsies.org>, (accessed 2016/07/16)

⁶<http://www.openstreetmap.org>, (accessed 2016/07/16)

The generation of data for routing purposes contains many different algorithms. To solve the distinct problems, it needs a pre-classified data set which filters the telemetry data at least for road and field data. This procedure is essential to make use of more specific algorithms that extract geographic features from the classified data and is the first step to investigate the problem of routing relevant data extraction from agricultural telemetry data. With this preceding step, a requirement tree will be spanned that handles the specific sub tasks. These tasks are handled separately and will finally be connected to complete the processing chain.

1.2 State of the Art

This chapter provides an overview on the current situation of data, systems and services in agriculture and similar domains that are related to this thesis. A methodological and algorithmic state of the art including the relevant publications will be given in the particular chapters.

Making a complex process computable, it needs deep knowledge about the process itself. A further requirement is the availability of data that is needed to compute the solutions. In the domain of agriculture, the machinery is equipped with a large number of sensors to give machine drivers an overview on the performance of a single machine. With the telemetry system it is possible to collect data from many machines in near real time that enables a direct extraction of knowledge. In the recent years, agricultural telemetry systems such as CLAAS Telematics⁷, John Deere JDLink⁸ and AGCommand⁹ enlarged their telemetry systems and has been build large collections of agricultural sensor and movement data. In the beginning, machine optimizations were in the focus of the manufacturers. Nowadays the spatiotemporal analysis of the data is coming into the focus. The collecting and analysis of movement data is driven by market and latest research. This is the case for many domains. In the domain of sports and leisure activities, companies like STRAVA and Garmin provide data analysis of trajectories. Personal health applications in smartphones collect movement and sensor data and give clients a feedback on their health and activity status. To achieve these analysis, spatiotemporal pattern analysis and classification methods are used. The objective and the main motivation is enhancement of knowledge on user behaviour.

In the domain of agriculture, the classification of working processes is mainly done manually (e.g. by setting the on-field/off-field mode using a switch on the vehicle). The classes of interest are mainly: Is the machine in a working mode or not and how much time is needed for the processing of single fields to calculate transparent billings for contractors clients. The recognition of these classes are mainly done by spatial intersections with field geometries (cf. Heizinger [2014]) or manual switching by machine driver (e.g. CLAAS Telematics working mode). From a data analysis perspective, the auto classification of the machinery movement data will be the first step to generate classified input data for further algorithms. Based on this foregoing classification step, the geographic feature extraction algorithms that consume the classified data are the next upcoming tasks.

The extraction of field geometries is done manually by farmers. By walking along the field boundaries with a GPS or digitizing them from aerial imagery. Large-scale

⁷<https://claas-telematics.com/>, (accessed 2016/10/22)

⁸<http://www.jdlink.com/>, (accessed 2016/10/22)

⁹<http://www.myagcommand.com/>, (accessed 2016/10/22)

field geometry extractions are done by aerial imagery classification and polygonization algorithms. To extract the field area that has been processed by the machine, the machine trajectories will provide a new data source. To route vehicles from road network on the fields, the knowledge of gateway points is essential.

The routing of vehicles within fields is a further task. To calculate the route for a transport or service vehicle from the field gateway point at the boundary to the harvester an infield route graph is needed. Comparable problems exist in the field of (autonomous) robotics (cf. Choset et al. [2005]), indoor routing (cf. Zlatanova et al. [2013]) or routing on squares (cf. Graser [2016]). Due to the fact, that these algorithms have not yet been used for agricultural fields, the research gaps have to be identified. The domain related requirements, such as field and machine related route graph orientation and the integration of gateway points, that connect the road network with the infield route graph, have to be integrated in the graph generation and routing algorithms.

The end to end routing of agricultural vehicles needs an attributed road network (including attributes, such as surface type or travel speed, to calculate edge weights). Existing commercial digital road networks do not meet these requirements. Data from HERE or TOMTOM lack the not public track road network. Companies such as logiball¹⁰ try to close this gap by using data from further sources (e.g. from companies that provide forestry data such as NavLog¹¹). This data is only available for Germany and it takes a huge effort and manual work to keep the data up-to-date. A worldwide data set for the usability in routing agricultural machinery is therefore needed. The only worldwide available vector data set with relevant scale is OpenStreetMap. It is based on geographic data which has been provided from volunteered mappers from all over the world. This data set will be evaluated if it meets the requirements for agricultural routing purposes.

Then, closing the white spots on the map is the missing part of the whole workflow. The research on methods for generating road networks from GPS trajectories has been started in the automotive industries in the end of the 1990ties. With the growing popularity of GPS receivers and the availability of tracking data, further methods have been developed in the domains of hiking and fleets tracking (e.g. taxis or delivery services). The research gap lies therefore in the analysis and refinement of methods that are able to extract a road network from the agricultural movement data. This can then be used to enrich and update existing road network data.

These components can be classified in:

- data preparation
- algorithms
- services

Most of the presented components will be integrated in a service oriented software architecture. Through this, it will be shown that a processing chain from data acquisition, via data processing to applications can be established. This demonstrator gives an example for an applicable real world scenario and shows the interaction between the individual components:

- The data preparation step with the fetching and storing.

¹⁰<https://www.logiball.de>, (accessed 2016/10/22)

¹¹<http://www.navlog.de>, (accessed 2016/10/22)

- The route service and the field service, that are connected via service interfaces.
- The different algorithms for road and field geometry computation, that feed the services and
- The routing demonstrator, that uses the data and provides the services as a prototype for end users.

1.3 Research Questions

Summing up the following research questions will be treated in this thesis.

1.3.1 How can agricultural movement data be classified to have distinct input data for field and road generation?

Agricultural movement data, as it is used in this thesis is produced from agricultural vehicles. Compared to cars and trucks, these vehicles drive on the public road network, but the usually greater part of their movement time, they drive on agricultural fields where they work. Hence, methods have to be investigated to classify the movement data and distinguish between different movement modes.

1.3.2 How can agricultural field boundaries be extracted from movement trajectories?

Identify methods and their limitations for automatic generation of field boundaries and field gateways to keep an up-to-date field record system and realize navigation from road network to fields.

1.3.3 How can an infield routing graph be extracted from field geometries?

Compared to the public road network, a complete route planning for agricultural vehicles requires not only a route to, but also a route within their working area (agricultural fields). Therefore, new methods for generating connected navigable paths within a field are needed to enable an efficient routing e.g. within a harvesting campaign.

1.3.4 How can a road network be extracted from agricultural telemetry data?

The extraction of a mainly rural road network is a further research question. Feasible methods have to be identified and parametrized. Based on a data set which represents the on road trajectories, methods for trajectory averaging and road network extraction have to be found and adapt on the agricultural data. The parametrization and the restrictions of each method have to be identified.

1.3.5 Can OpenStreetMap data as VGI be a source for vehicle specific route planning in agriculture?

The extracted data from the telemetry data will not represent the whole road network. Hence, a basic digital road data set is needed for routing without gaps. In the last years,

OpenStreetMap became to a proper data set that shows in different services its feasibility for routing. The usability for routing agricultural vehicles, the pros and cons will be enlightened here.

Chapter 2

Used Datasets and TeleAgro+ Framework

2.1 Agricultural Telematics Data

2.1.1 Telematics in Agriculture

Observation and analysis of processes is the key to improve them. Sensors and displays provide feedback to the user and thus an insight into the status of both the machine and processes. Having the information onboard of the machine is one part but analysis options are restricted to the single machine and to limited attention of the driver who is sometimes flooded with information on machine, accessory equipment and communication with e.g. further players of the logistic chain. An overview of the machinery as a whole and performance comparisons between similar machines enable better machine setups. Machine maintenance is easier and the availability of previous sensor values, readable from service simplifies fast and effective repair which is tremendous valuable especially while harvest period. With the integration of GPS, mainly initiated by parallel driving and guidance systems the on-board sensor data is enriched with locational information. An overview on agricultural telemetry systems is given in Andres [2009]. The extension of the mobile network and the establishing of faster mobile network standards like LTE promoted the distribution of telemetry systems. Besides several small companies that are providing smaller telemetry solutions, the three big agricultural machinery manufacturers AGCO, John Deere and CLAAS cover the main part of the telemetry market. The systems of the three global acting agricultural manufacturers underlie continuous development within the last years and the implementation of telemetry on vehicles is become default for superclass tractors like the CLAAS Xerion. John Deere's telemetry product JDLink comes from the construction part of the company. Therefore the functionality lies mainly in observing the machinery, primarily for product observation and maintenance. However a broader set of functions is planned for future. The AGCommand system from AGCO has its roots mainly on machine health and performance monitoring and in operation and efficiency monitoring. The software allows geo-fencing and visualization of the fleet. The CLAAS Telematics is one of the first telemetry systems for agriculture machinery and has its origins in monitoring harvester machinery. Starting with the AgroScout system (a fleet management software for agriculture), CLAAS set the focus more on performance analysis of harvesters. With the update in 2014, the CLAAS Telematics system is now available for tractors and machinery attachments such as balers. Including sensor data available

from ISOBUS and proprietary BUS systems the CLAAS Telematics is one of the most detailed systems available on the market. Figure 2.1 shows the desktop views of the three telemetry systems (CLAAS Telematics, AGCommand and JDLINK).



Figure 2.1: Agricultural Telematic Systems
 CLAAS Telematics (source: <https://www.claas.com>),
 AGCommand (source: <http://www.agcottechnologies.com>)
 and JDLINK (source: <http://www.deere.com>)

Besides these main actors surveying companies like Leica Geosystems (VirtualVista) and TopCon (Tierra) offer systems for machine telemetry. Another agricultural machinery company “Horsch” recently developed their own system [Baum and Rothmund, 2014] which is mainly used for planning and controlling machinery. The disadvantage of these systems is their boundedness to ISOBUS and the free offered data by agricultural machinery companies. This thesis primarily focuses on the CLAAS Telematics system which provides mainly data from CLAAS agricultural vehicles. Due to the fact that methods in this thesis are developed mainly using movement data, this processing can be adapted on every other telemetry system by using their interfaces.

2.1.2 CLAAS Telematics

History

The data used in this thesis is extracted from the proprietary CLAAS Telematics system. This agricultural telemetry system is the telemetry system of the CLAAS company and started in the year 2005 [Andres, 2009] as a system for remote performance surveying of agricultural vehicles. At the beginning it was implemented as special feature on high class combine harvesters. Later CLAAS implemented the Telematics on forage harvesters and some years ago the system is also available as add on for tractors such as the Xerion, Axion and Arion models. With the implementation of the ISOBUS standard, the CLAAS Telematics system is implemented as now also an additional feature on no-CLAAS tractors and transportation vehicles and can log the available data on these machines (although this data is less comprehensive than data available from CLAAS machines). The telematics feature will be available as a standard implementation on high class machines and as an extension for mid and small machinery in near future. An integration of further

CLAAS products (e.g. Scorpion) and the integration of implements (e.g. binders, manure distributors, sprayer) is also planned.

Process and Architecture

The CLAAS Telematics system consists of a centralized database architecture which collects data from machine clients. Each sensor, which is connected with the machine's CAN-BUS, sends its data to the BUS system where a further process takes the sensor values and stores them in a fixed frequency (15 s) on the CEBIS-board computer (for new harvesters and especially tractors this frequency will be higher). The sensor measurements will be packaged together with a timestamp and a location (from GPS) and sent in flexible intervals to a centralized data storage platform. The incoming data is stored in a database where further processing steps will be managed. In figure 2.2 a general overview on the general Telematics architecture and its components is given.

The CLAAS Telematics system consists of following main components [Andres, 2009]:

- Datalogger software
- Teleservice module (with integrated GSM modem and SIM card)
- communication server
- telematics database and webserver
- webinterface CLAAS-Telematics

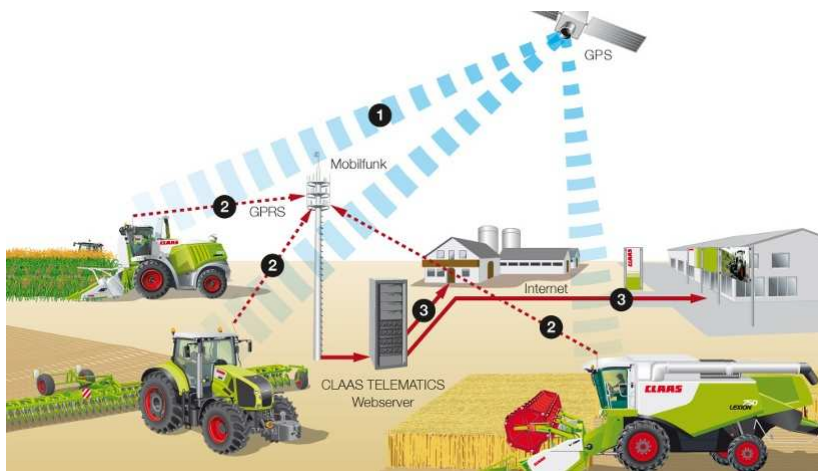


Figure 2.2: Architecture CLAAS Telematics (source: <https://www.claas.com>)

Machines, Sensors and Data

In this work, a subset of the available machinery from the CLAAS Telematics system data is used. The sample data set consists of Telematics data of eighteen machines for the last five years (2010-2015). The set of sample machines consists of combine harvesters, forage harvesters, CLAAS tractors and tractors from other manufacturers where the CLAAS Telematics has been implemented. In table 2.1 and figure 2.4 the machines and their provided data (respectively) are described in more detail.

The spatial extend of the data is limited on Central Europe, in particular on the countries Germany, France, Czech Republic and England (cf. figure 2.3).

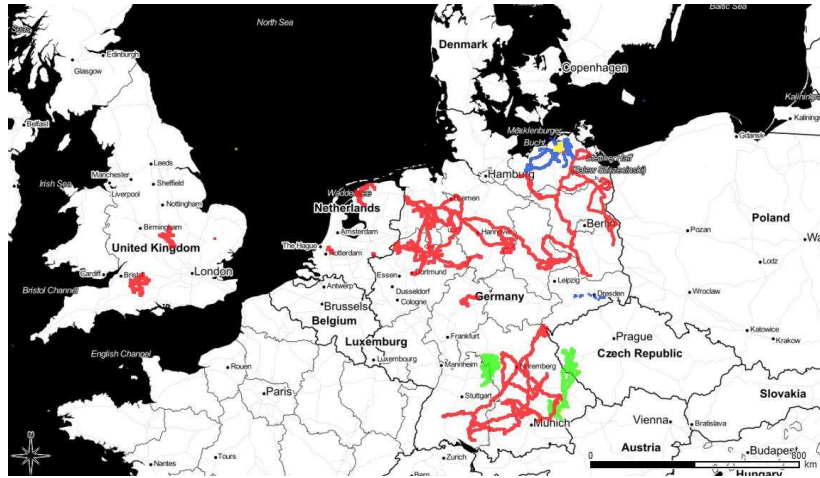


Figure 2.3: Overview on spatial distribution of the data set (machine types: forage harvester, combine harvester, tractor and tractors from other manufacturers, are grouped by colors) (Datasource basemap: ©OpenStreetMap contributors)

Figure 2.4 gives an overview on the available measurement values for the analyzed machines from the CLAAS Telematics system. The sample set contains data from different combine harvesters, forage harvesters, CLAAS tractors and tractors from other manufacturers. The figures also show the differences between machines regarding semantic similar measurement values. While the Axion tractors have two speedometers (radar and gearing), the other machines only have one speedometer (gearing). Therefore the derived “Schlupf” (slip) parameter is only available for the newer Axion tractors. Some of the Jaguar forage harvesters were built for a period with two engines. Hence the measurements for “Kraftstoffverbrauch” (fuel consumption), “Motordrehzahl” (engine rpm) and “MOTORauslastung” (engine capacity utilization) are collected separately for each engine. Two different namings “max. Anzahl Teilbreitenstufen” and “maximale Teilbreitenstufenzahl” (maximum section count) for the same value and (already revised for this table) inconsistent type setting for parsing the values (additional space characters etc.) reveals the necessity of a consistent dictionary for all values. The values for “Steuergerät XX Status” are not further explained, because of their dependency of the accessory equipment like sprayers, manure distributor etc. . It should also be considered that the measurement values of the machines used in this thesis is a subset of values. Machine owners have a bigger set of measurement values that also consists of harvest data, harvest good type and further data, but such data is subject to privacy regulations and therefore would not be part of this thesis.

As shown in figure 2.4 the available sensor data is different for each machine type. Within the type classes there are also differences between models and individual machine configurations. Hence the table presents a representative subset for machines available in the CLAAS Telematics system. This subset highlights the requirement for grouping the data at least by type which have equal sensor configurations installed and therefore comparable data (e.g. Axion 900, Lexion 600 ...). Some values seem to be comparable due to semantic equality of tags and different spellings of identifiers.

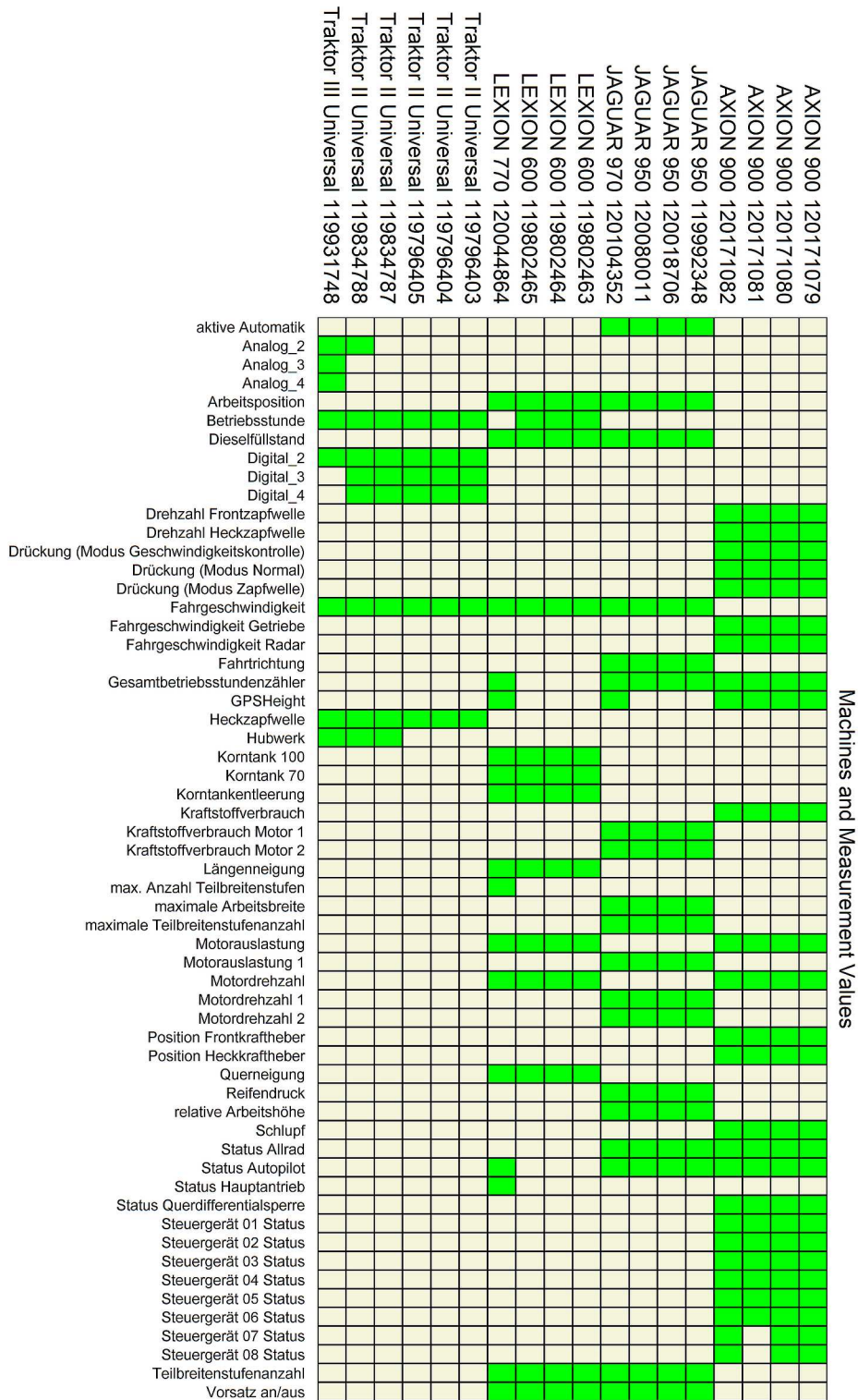


Figure 2.4: Measurement Values for Machines

2.2 OpenStreetMap

2.2.1 Project Description

OpenStreetMap (OSM) was invented by Steve Coast in 2004 with the idea to provide user generated, free available cartographic data [Ramm and Topf, 2010]. The infrastructure became ready in march 2006 to collect and map larger areas. It is an exemplary project for volunteered geographic information (VGI). The idea at the beginning was to collect GPS traces from users, show them within a web editor and let users digitize their tracks and add semantic data. The provision of satellite imagery from LandSat and later the more detailed yahoo maps imagery forced the digitizing of new geographic features and closed bigger gaps in the digitized data. With the continuous provision of services based on the freely available data (e.g. online maps, routing, geocoders), users were motivated to generate new data and the availability of OpenStreetMap data increased very quickly. The data structure is very simple as it consists only of nodes and ways (which are an ordered list of nodes) and key value pairs for each type that consists of semantic data [Ramm and Topf, 2010]. The scheme has not been changed apart from new additions like relations that represent sets of associated ways or nodes (cf. figure 2.5). The data is stored on a central server infrastructure and provided by web interfaces or file downloads from OpenStreetMap servers or mirrors.

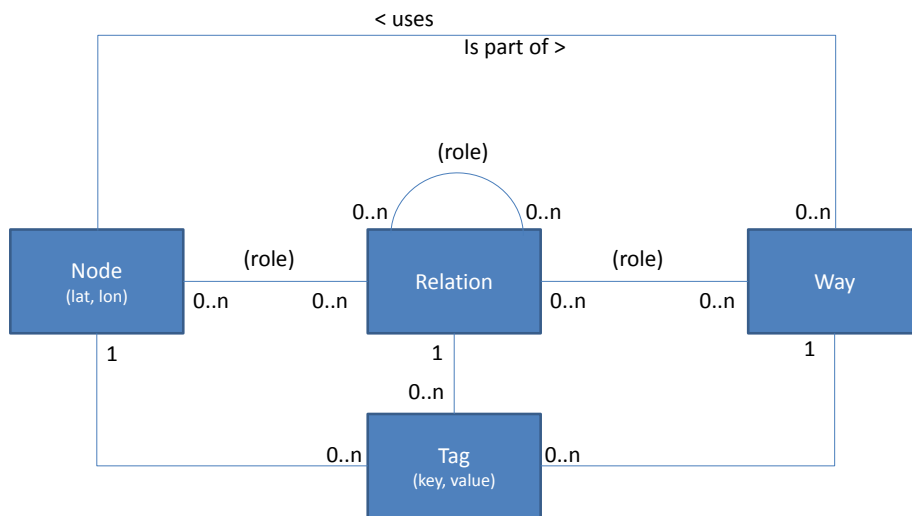


Figure 2.5: OSM Data Model (source: adapted from Ramm and Topf [2010])

From 2006 on the number of users rapidly grew (figure 2.6) and with that so did the quantity of digitized features (figure 2.7).

Based on the OpenStreetMap data a countless number of software and services have been implemented. Map renderers such as OSM WMS [Goetz et al., 2012], OpenMap-surfer (Rylow [2014]), Mapnik, domain related maps like “Reit- und Wanderkarte” (<http://www.wanderreitkarte.de/>) and 3D services such as W3DS [Schilling, 2012], etc. process

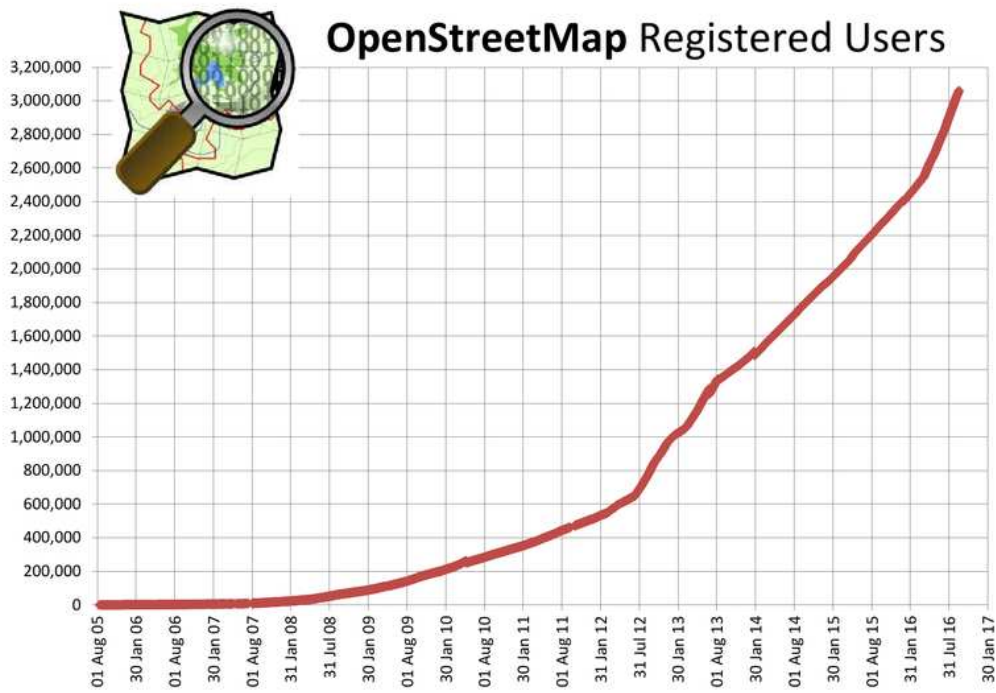


Figure 2.6: OSM user statistics (source: <http://wiki.openstreetmap.org/wiki/Stats>)

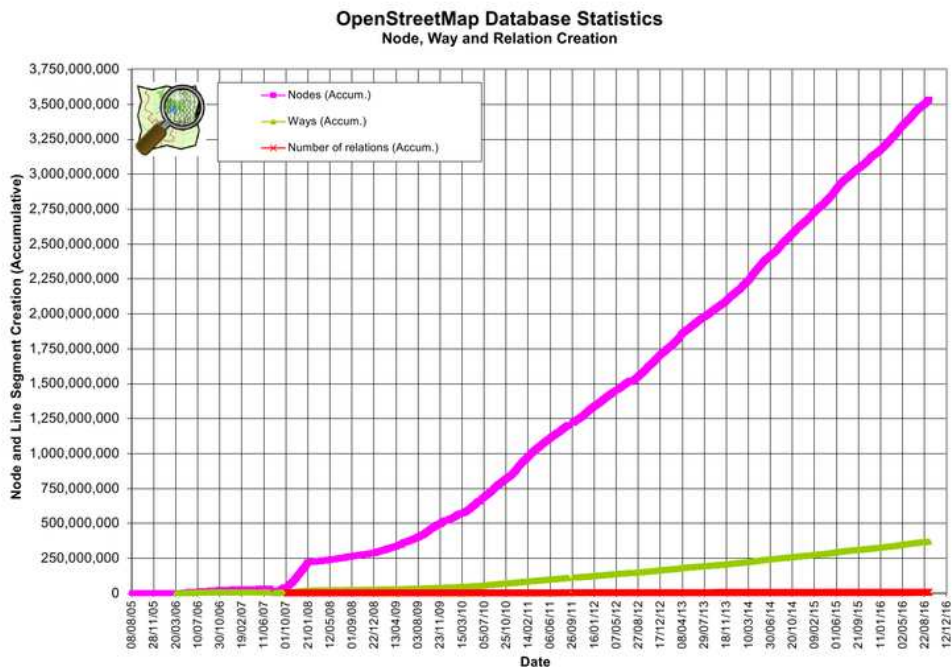


Figure 2.7: OSM data base statistics (source: <http://wiki.openstreetmap.org/wiki/Stats>)

the data mainly for visual presentation. POI services like Nominatim (<http://nominatim.openstreetmap.org/>), the OGC OpenLS directory service of GIScience Heidelberg or Wheelmap.org (<http://www.wheelmap.org/>) provide POI data via web interfaces and allows spatial requests on the POI database. Route services like OpenRouteService.org [Neis and Zipf, 2008], OSRM [Luxen and Vetter, 2011] or the Graphhopper project demonstrate operative routing systems using the free OpenStreetMap data. Regarding the quality of this VGI source, several works showed with comparisons to other data sets [Zielstra and Zipf, 2010, Haklay, 2010, Ludwig et al., 2011], through intrinsic analysis of the data [Baron and Neis, 2013] or further investigations [Arsanjani et al., 2013, Mooney et al., 2010, Graser et al., 2013, Ciepluch et al., 2011] the usefulness of the data set. Further investigations have been made by Canavosio-Zuzelski et al. [2013], Girres and Touya [2010]. In Haklay [2010] the positional accuracy of OSM road data has been measured using buffer methods and sample data was at average less than 9 m or better.

As OpenStreetMap is mainly based on crowdsourced, human generated content, mechanical edits through bots and automatic processing chains have to be handled carefully and well communicated with the OpenStreetMap community. For these kind of edits there exists a code of conduct which sets up rules for automatic data processing [OpenStreetMap, 2015]. Due to this conduct, the data generated in this work will not be directly integrated in the OpenStreetMap data set. However, services which are installed within the context of TeleAgro+ can provide the extracted data as layers for manual editing. This allows the community to check the data before integrating the data manually into the OpenStreetMap VGI data base.

Within chapter 6 further investigations on OpenStreetMap as base data for agricultural route planning will done. There will given a detailed description on the data, related to the application of routing agricultural vehicles.

2.3 TeleAgro+ Framework

This work is part of the research Project TeleAgro+ which was funded by the German Ministry of Food and Agriculture (BMEL) from 2011 to 2014. The research project is a collaborative project between the agricultural machinery company CLAAS E-Systems (former CLAAS Agrosystems) and the GIScience group of Heidelberg University. The project objectives are the extraction of knowledge from telemetry data of agricultural machines, the integration of geodata in the end user products, and a user friendly interface for data management and analysis of productive agricultural machinery. Furthermore a prototypical route planning service for agricultural vehicles has been implemented in the project.

2.3.1 Framework Overview

The TeleAgro+ framework consists of data storing and analysis functions. The architecture is shown in figure 2.8 and for the route service in detail in 6.6. The following chapters explain the used algorithms from data pre-processing 3 that is the basis for the later processing, the extraction of field geometries in chapter 5, the extraction of road geometries in 4 and the routing for the agricultural vehicles including algorithms for infield route computation in chapter 6 which brings al the components into a seamless workflow. The architecture builds mainly on services. Therefore extracted data is provided to users through web services and prototypical user interfaces and applications. This modular

structure with standardized interfaces allows a flexible framework with high potential for reusability and a partial integration of services in other systems.

2.3.2 Processes and Services

The components of the architecture showed in figure 2.8 will be described in the following subchapters.

Telematics Data Updater

The Telematics Data Updater is an essential service of the TeleAgro+ framework. It organizes the initial creation of the database structure and the initial data import. Using the REST interface provided by CLAAS Telematics the data is pulled via this interface and loaded into the local PostGIS instances. The DataRetriever also allows the updating of an existing data set by comparing the latest stored data with the latest available data from the Telematics system. This service also implements the latter in chapter 3 described filtering procedures and is responsible to get clean data into the data base.

Extended Measurement Processor

The *Extended Measurement Processor* is the calculation tool for extended measurements. This processor uses the spatial functions of the database and enriches the existing raw data with further attributes such as spatio-temporal neighborhood, speed and acceleration, . . .

Field Boundary Generator

The *Field Boundary Generator* is responsible for extracting fields from pre classified measurement data. It computes geometries for fields and stores them in the spatial database with additional attributes (e.g. harvest year and machine). The polygons are also simplified by algorithms produced by Ramer [1972], Douglas and Peucker [2011] to prune the data. With this simplification the JSON data for delivering the field features via the REST interface is small enough to realize a usable client-server communication. Furthermore the later used algorithms for infield route graph computation which are based on the field geometry and its shape points are running with improved speed and without noteworthy loss of detail. The *Field Boundary generator* also computes the field connection points that are used to connect the field geometry with the road network.

2.3.3 Trajectory Processor

The *Trajectory Processor* classifies the measurements using machine learning methods on training data. With the classified measurements it computes trajectories and corresponding line strings and stores them in the spatial data base.

RoadProcessor

The *Road Processor* implements several methods for road computation. In this processor, the algorithms for data import and track averaging are located. The data output is commonly made by files or the data base.

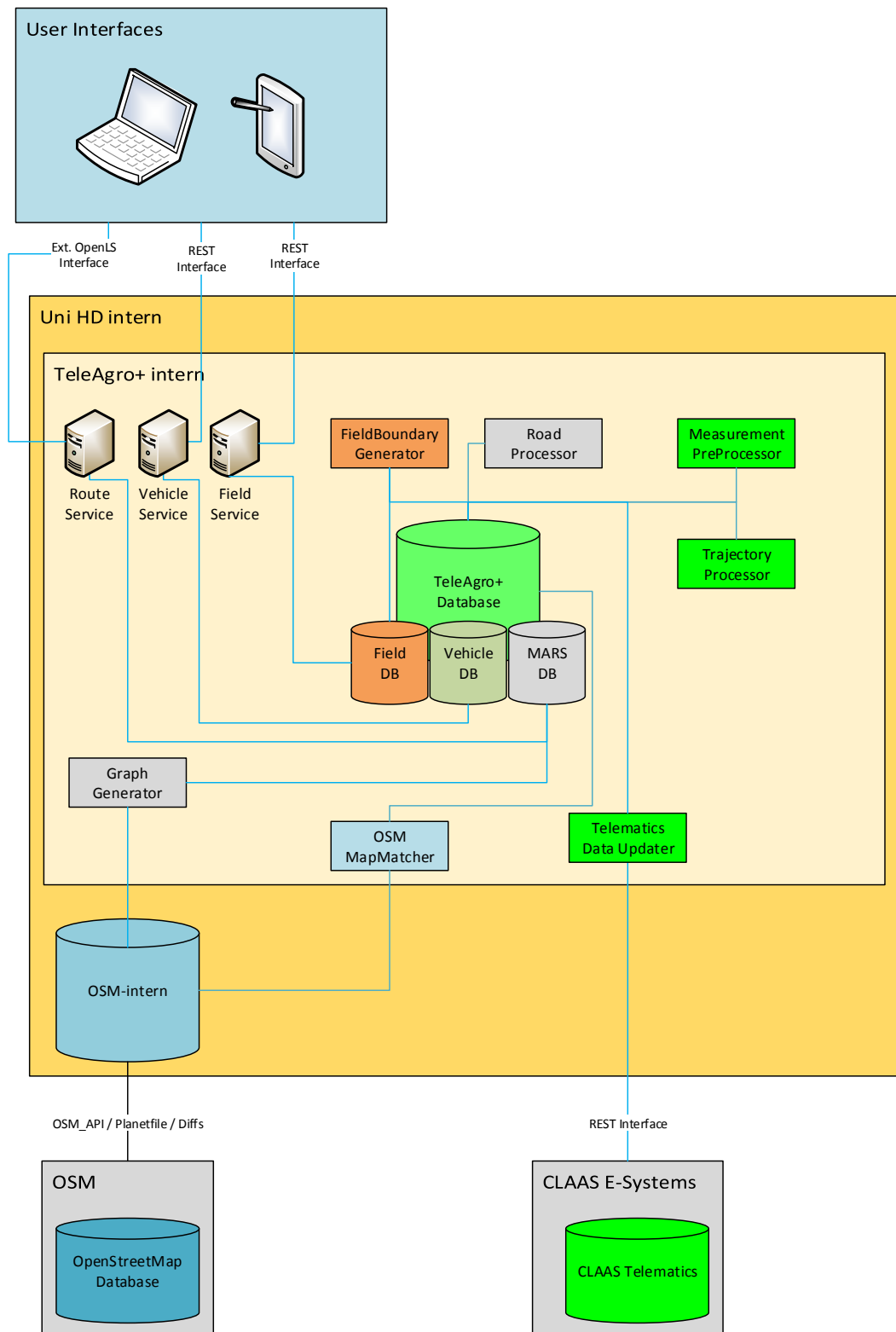


Figure 2.8: Architecture TeleAgro+

Graph Generator

The *Graph Generator* imports OSM data and builds a graph structure from the data. In this step the edge weights were also encoded and stored within files and the routing data base.

RouteService

The *MARS Route Service* is the *(M)ultit (A)ttribute (R)oute (S)ervice* that calculates routes on the generated routing graph using a set of multiple attributes and restrictions. The service interface is based on the OGC OpenLS specification [Mabrouk et al., 2005] which has been extended to enable the multiple attribute functionality.





VehicleService

The *Vehicle Service* provides the vehicle data of multiple vehicle configurations via a REST interface. The data sent as in a standardized JSON format through the web service interface. This service also returns average speed data for different highway types and attribute configurations for individual vehicles.

FieldService

The *Field Service* provides the output of the *FieldBoundaryGenerator* via a standardized REST interface. The fields are sent in (Geo)JSON format to the client. This service also sends the field connection points which are needed to connect the fields to the road network. It also enables spatial and attribute related requests (e.g. fields within a bounding box, fields of a specified harvest year).

Table 2.1: Analyzed machine types from CLAAS Telematics
(source of imagery: <https://claas-telematics.com/>)

| | |
|---|--|
| <p>forage harvester</p>  | <p>CLAAS JAGUAR</p> <p>The forage harvester Jaguar has been built since 1973. The 950 and 970 models which contribute data for this work have been built from 2007. The 950 has a 16 cyl. 372 kW machine and can be equipped with 3.0 or 3.5 m pick-up or for corn harvest with a 7.5 or 6 m corn picker-husker. Other equipment from foreign manufacturers is also possible. The Jaguar machines have been constructed for a period of time with two engines with the aim to save fuel on road and add the power of a second engine for foraging on field. This has been canceled due to machinery problems and higher effort in maintenance. The forage harvesters are primarily used for chopping grassland, whole crop silage and corn chopping.</p> |
| <p>combine harvester</p>  | <p>CLAAS LEXION</p> <p>The CLAAS Lexion has been built from mid 1990. The analyzed machines in this thesis are from the 600 series which have a working width up to 12.0 m, an engine power of 390 kW and a grain tank of 12 to storing weight. Also the Lexion 770 from the upper class that has been built from 2010 on with similar measures. Both types can be equipped with TerraTrac technology for more efficient driving. The Lexions harvest primary grain like wheat or rye, but also reps and corn.</p> |
| <p>tractor</p>  | <p>CLAAS AXION</p> <p>The Axion tractors are quite new in the CLAAS machinery programme. With the takeover of the tractor division of Renault, CLAAS started to build their own tractors. The data subset in this thesis is produced by mainly testing machines of the 900 series which have been built from 2011 onwards. The machines have engines with 235 - 302 kW and represents the upper tractor class beside the CLAAS Xerion which was developed as a functional large tank. The tractors are used for almost all field work like soil cultivation (tillage, ploughing), seeding, fertilizing, and spraying. Further they are used as transport vehicles in biomass logistic chain to transport the harvest e.g. from field to a bio gas plant or the grains to agricultural traders (mainly for short distances less than 30 km).</p> |
| <p>other manufacturers</p>  | <p>External brands</p> <p>Additional data from external brands are in the classes "Traktor II Universal" and "Traktor III Universal". These machines have only a semi integrated CLAAS Telematics on board which means, that the set of available sensor data is very sparse and their description is less detailed (e.g. "Steuergeraet XX Status"). The used data set includes Fendt 936 Vario (the flagship of AGCO-Fendt), 722 and 718, a John Deere 6190 R and two CASE tractors, a CASE Puma and a CASE caterpillar.</p> |

Chapter 3

Data Preprocessing

This chapter explains, evaluates and extends algorithms for pre-processing the agricultural telemetry measurements. While the first paragraphs describe the required data cleaning steps to eliminate corrupt and obvious wrong data, the further sub-chapters describe methods for matching the measurements with existing map data, the calculation and derivation of (movement-) attributes, and the classification methods to group the measurements into different classes. These classes are the basic input data for the road and field boundary extraction processes in the later chapters.

3.1 Data cleaning

As the used data is from a productive telemetry system, it relies on continual development. Hardware (sensors and terminals) as well as software and its interfaces are subject to continual updates. Data is sometimes deficient and erroneous, but some errors are only noticeable after further data processing. In this section the main errors which hamper the following data analysis will be shown and methods for correction and elimination of faulty data will be given. Some of the error types are already described in Lauer et al. [2014]. The following part will give a broader view on the errors and their correction or elimination. It has to be said that this error handling is done in a generic way to get a basic data set for the subsequent processing steps. Within the later, more specialized processing stages further individualized and method dependent data cleaning and separation processes are needed. This individual processing will be explained in the specialized sections.

General errors can be classified as positional errors, range errors, duplicate errors, logical errors and type errors. Positional errors originated from wrong GPS measurements. They can be also range or logical errors. Hence GPS measurements are affected by several sources of errors that have an influence on the geometrical accuracy. The positional errors can only be estimated by taking the statistical measurements (e.g. DOP - dilution of precision) usually provided by the hardware sensor and the standardized interface into account. These errors are usually hard to detect and the DOP values (from GPS sensor) are not provided by the CLAAS Telematics interface. Due to the fact that GPS usually has a positional error which is less than 10 m, and the conditions for position measurement with GPS on agricultural vehicles in open areas are very good, the positional errors would not be handled directly. This positional error can also be a benefit for some algorithms which make use of the distributed values and resulting trajectory crossings.

The other part of positional errors are mainly originating from hardware and software

interfaces. Examples are zero coordinate N0.0, E0.0, switched lat/lon values or numbers out of range (latitudes $\notin [-90, 90]$ and longitudes $\notin [-180, 180]$). These errors can be produced by GPS hardware or during the transference and importing of the data into the CLAAS Telematics database. A bug in the outgoing interface is also possible as the database and the interface software sometimes reorganize the data before sending. A further logical check has been implemented to eliminate obvious wrong values (e.g. with unrealistic speeds or alternating positions in figure 3.1).



Figure 3.1: Error - Alternating positions (Datasource basemap (orthophoto): Bayerische Vermessungsverwaltung – www.geodaten.bayern.de)

The last type of errors are naming and typing errors. The constantly growing and evolving CLAAS Telematics system brings some inconsistency. Attribute naming and typing were not standardized and as a result semantically similar attributes were named differently. Different typing (upper/lower case) is also a problem for further processing and has to be handled by less sensitive string handling. The usage of non common characters (e.g. German umlaut) in combination with wrong string encoding will also affect the further processing steps. The strings therefore were filtered and encoded in UTF-8 before importing into the TeleAgro+ database. Also preceding or trailing space characters were eliminated. As far as possible semantically equal and comparable attributes were named equal while importing to make them comparable within the analysis steps.

The main error types are listed in 3.1. After the detection and cleaning steps, the main

Table 3.1: Error classification - Telemetry data

| Error Type | Description | Example |
|------------|------------------------------------|---|
| positional | wrong positions | measurement within the forest next to the road or within a river |
| logical | hopping of successive positions | unrealistic speeds (e.g. harvester drives with 150 km/h, cf. 3.1) |
| duplicate | duplicate measurements | identical measurements due to duplicate data fetching |
| range | data is not within a defined range | latitudes $\notin [-90, 90]$ and longitudes $\notin [-180, 180]$ |
| type | wrong encoding or description | “ue” instead of “ü” |

errors are fixed and the data set is prepared for following analysis.

3.2 Map Matching

To connect the measurements with existing map data a matching process is needed. There exists many approaches to match GPS data to an existing road network. Quddus et al. [2007] classifies matching algorithms into geometric, topological, probabilistic and more advanced algorithms that use more refined concepts like Kalman filters or particle filters amongst others. The Hidden Markov Model (HMM) approach of Newson and Krumm [2009] is one of the more recent and most promising approaches. Problems with the HMM approach lie in the scalability (computation time will unduly increase with dense point intervals and high detailed map data) and its sensitivity to the data. The project *enviroCar* (Bröring et al. [2015]) uses a simple “points-to-line”-snapping approach to match the vehicle data on the OSM roads. This approach is not feasible because of the possible alternating matches and therefore wrong matching results. The alternating matches can be caused by bad GPS signals and wrong positions. The approach that is used in this thesis is realized by point to line matching (with a given threshold) and the looking forward and backward to neighbor measurements. A sudden road change from measurement to its successor and back is quite unrealistic and the fact that vehicles usually not change a road without physical connection the measurement series is analyzed on runs and the matching underlies a smoothing like method to calculate the correct road match. The algorithm represents therefore the tree classes of geometrical (intersection with road buffer), topological (which values have predecessor and successor values) and probabilistic (which road segment has most hits). This algorithm fits the requirements for the following processing and the visual analysis confirmed a reasonable matching.

3.2.1 Method

The position for each single measurement is given by GPS. Their positional accuracy is therefore determined by the onboard GPS sensor. For matching the vehicle positions to existing map data, map matching algorithms are needed. Using a naive geometric buffering gives first results on matched measurements. Through its simplicity, this approach is not able to handle wrong matches with e.g. parallel ways and also causes problems with crossroads and produces multiple matches.

As shown in figure 3.2 (left) using the simple map matching algorithm by using a buffer on each road geometry, all the points P1...P5 are matched with road 1, but P3 is also intersected with the buffer geometry on road 2. Due to this additional overlap, P3 has two matches. To handle these wrong map matches, especially the hopping from one road on its neighbor and the multiple matches, a more precise algorithm is needed. The algorithm has to consider the movement of the vehicles and the fact that changes of highways from measurement to measurement are not common. Therefore long runs of matched measurements are more probable than short runs with many alternations. The points P1...P5 are now matched to Road 1 because of the longer chain where P3 is part of figure 3.2 (right). The result is a more precise road geometry matching for each measurement.

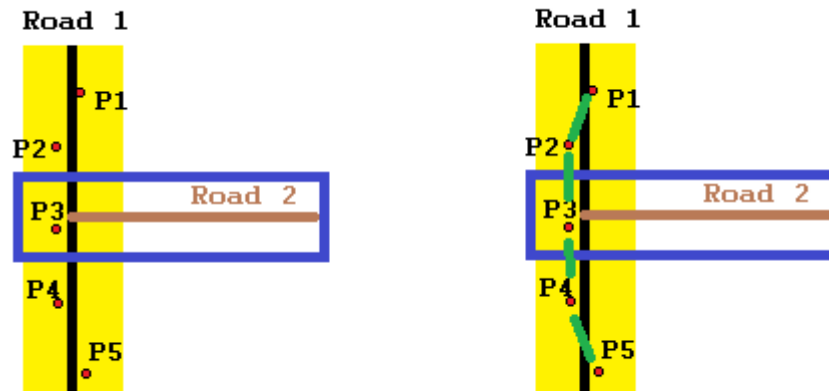


Figure 3.2: Mapmatching process

3.2.2 Map Matching Results

The map matching for all measurements was computed using the previously described algorithm. The input map data is from OpenStreetMap. A mirrored original database is used for an up-to-date OpenStreetMap data base server instance. The workflow for this service is described in Goetz et al. [2012]. The map matching is done once after updating the CLAAS Telematics data. The OpenStreetMap Data is therefore from the same actuality as the latest telemetry data. Figure 3.3 gives an overview of matched OSM-highway classes. 2.9 mil measurements are matched to the street network of OpenStreetMap. The main part (98.6%) of all matched measurements is matched on the classes secondary, track, residential, unclassified, tertiary, service and primary. To avoid misunderstandings, the OpenStreetMap-tag “unclassified” stands for connecting roads between villages.

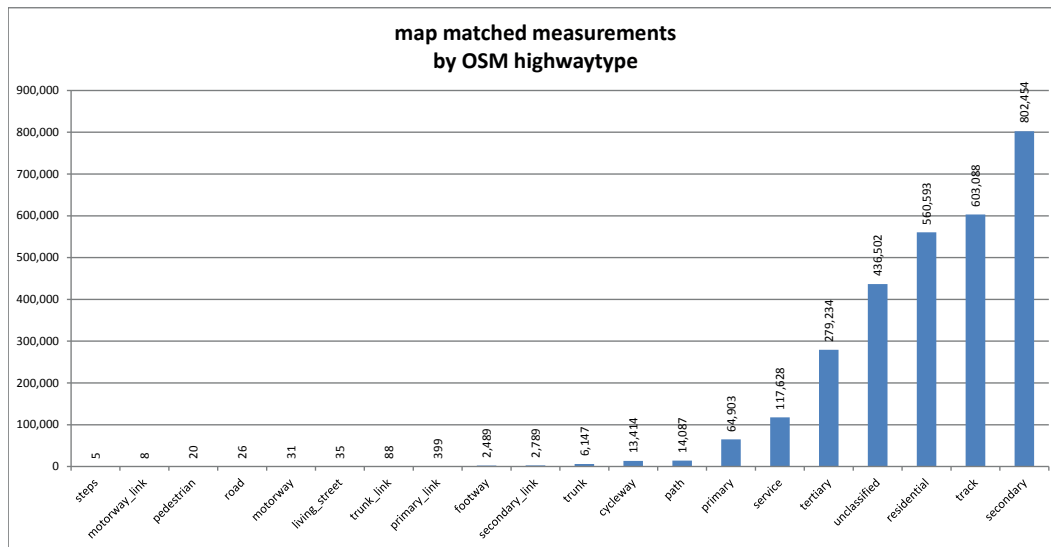


Figure 3.3: Mapmatching statistics

The wrong matches of the extended buffer method could be decreased and the method is able to produce tangible results for the later processing steps. However mainly two kinds of errors generally occur which could not be eliminated. Examples are shown in following maps. One reason for a wrong classification is the wrong geometric matching. This is the case, if two streets are parallel and very close to each other (e.g. a track or a path beside a primary road) as can be seen in figure 3.4. The algorithm is not able to differentiate between the two parallel roads (highway=path or highway=primary). If the only criteria is the nearest road, the measurements will be alternating their match due to GPS inaccuracy. This is one of the main challenges of map matching. The chosen approach, a window based matching solves this problem sufficiently.

The other error is a wrong tagged highway-type in map data (figure 3.5). The figure shows a matched way with the attribute *highway=path* which is described in the Wiki as “highway=path is a generic path, either multi-use or unspecified usage, open to all non-motorized vehicles.”¹. This is an error that is independent from the matching and can only be solved by revising the map data.

Semantic correction is also a possible solution to get better results, as some highway types cannot be driven on by the vehicles. This is mainly the case for single trails or stairs. Due to the very low error level and its only nominal influence on further processing, this correction is not made in this work. A positive side effect of this method is the chance to reveal errors in the OpenStreetMap data.

¹<http://wiki.openstreetmap.org/wiki/DE:Tag:highway=path>, (accessed 2016/10/22)



Figure 3.4: Matched path vs. primary

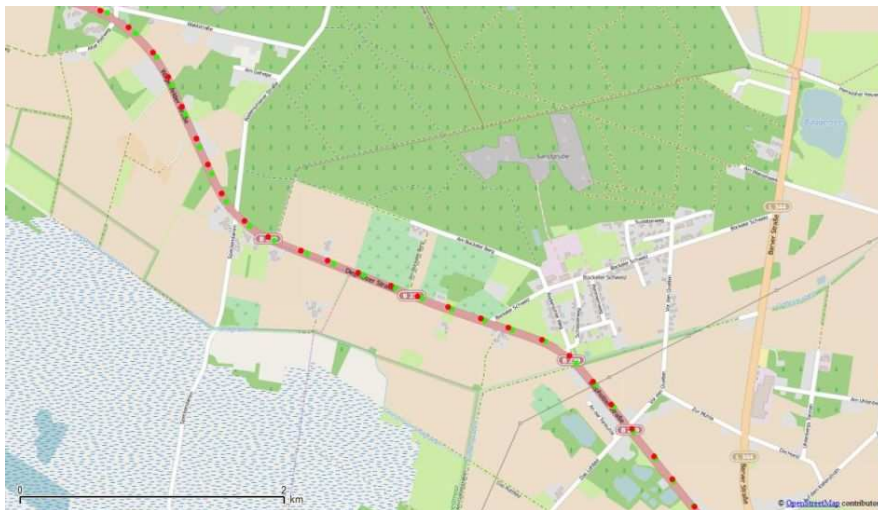


Figure 3.5: Matched path

3.3 Average Speeds for Road Types

A further pre-processing step is the calculation of average speeds for every combination of used highway-, surface-, smoothness- and track-types. These statistics are used for later route calculation and time estimation. The matched measurements and the calculated GPS-speeds of each measurement are averaged for each combination of driven roads. The results are then stored in a vehicle table. This table will be the data source for the *VehicleService* which provides the vehicle data for the vehicle specific routing. The average speeds were calculated for every existing matching combination of OpenStreetMap tags (highway, tracktype, surface and smoothness). Missing combinations were filled by the *VehicleService* with default speeds of the machines. Figure 3.6 shows the average speeds of each machine on each matched highway type.

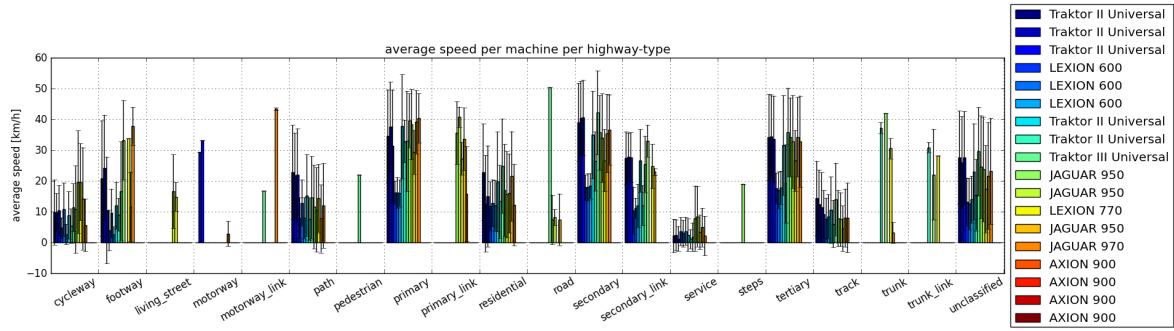


Figure 3.6: Average speeds per Highwaytype

The figure shows the typical speeds of the agricultural vehicles which are depending on the road types. On broad roads the machine driver usually accelerates to the maximum speed and holds that as long as possible. This was the impression of several field campaigns and also represents own experience during several years of practical farming. The average speed on these road types are in the region of the machines maximum speed. Road types like service or track are usually smaller roads with only one lane and sometimes not paved surfaces where higher speeds with large machines are not possible. The short distance between crossings and the small curvatures need often accelerations and decrease the maximum and the average speeds.

3.4 Extended Attribute Calculation

Analyzing movements requires attributes that describe the movement and the spatiotemporal neighborhood of individual measurements. Hence, extended measurement attributes are computed to extend the feature space.

Table 3.2 shows the calculated attributes for each measurement. The attributes distance, speed and acceleration are typical values for movement description. The azimuth is the angle that is spanned by the connection of two successional measurements and the north direction. As the coordinates are polar coordinates, *atan2* has to be used to differentiate between the different quadrants. The curvature describes the angle that two successional measurements with their azimuth angles will form. The spatiotemporal neighborhood of a measurement is the sum of neighbor values within a given time τ and a maximum euclidean distance δ . The thresholds of τ and δ were restricted to plausible values that also depend on the use-case. For τ the values 0.25h, 0.5h, 1.0h, 2.0h and for δ the distances of 50m, 100m and 200m were chosen and the numbers of neighbor elements were computed for each combination of τ and δ . These values are no typical movement parameters, but they give a number for the vicinity of a sample.

3.5 Measurement Classification

The telemetry data represents the complete spatiotemporal movement of the agricultural vehicles. Further processing needs a differentiation of trajectory segments depending on the analysis objectives. Due to the primary objectives (extraction of field boundaries and road data set), the raw data has to be classified to at least the classes field and road. Simple approaches are using spatial intersection and map matching but this is only

Table 3.2: Formula for extended attribute calculation

| parameter | formula | unit |
|-----------------------------|--|---------------------|
| distance | $d(S_n, S_{n-1}) = S_n.p - S_{n-1}.p $ | [m] |
| time | $t(S_n, S_{n+1}) = S_n.t - S_{n-1}.t$ | [s] |
| speed | $speed(S_n) = \frac{d(S_n, S_{n-1})}{t(S_n, S_{n-1})}$ | [m/s] |
| acceleration | $acc(S_n) = \frac{speed(S_{n+1}) - speed(S_n)}{time(S_n, S_{n+1})}$ | [m/s ²] |
| azimuth | $azimuth(S_n) = atan2(y_{S_{n+1}} - y_{S_n}, x_{S_{n+1}} - x_{S_n})$ | [°] |
| curvature | $curvature(S_n) = azimuth(S_{n-1}) - azimuth(S_n) $ | [°] |
| spatiotemporal neighborhood | $stn(S_n) = S_i \in S_O d(S_i, S_A) < \delta \wedge t(S_i, S_A) < \tau$ | [#elements] |
| | <p>with S_{n-1} as chronological predecessor and S_{n+1} as chronological successor of S_n S_O as the set of all samples for an object O δ as distance threshold τ as temporal threshold</p> | |

feasible when feature data of fields and roads are given (chicken-and-egg problem). To fulfill the aim, a process to classify the movement data has to be established. Preliminary work of measurement classification is mainly done in movement analysis where different transportation modes have to be distinguished.

Dodge et al. [2009] present a heuristic method for trajectory segmentation. Within three steps (data preparation, global descriptors computation, and local feature extraction) they demonstrated how to classify trajectories by transportation modes (motorcycle, car, bicycle, pedestrian) using unlabeled movement trajectories. Further approaches classifying movement modes were performed by Bolbol et al. [2012] who classified GPS trajectories in transportation modes of car, walking, cycle, underground, train and bus using a SVM classifier and attributes like average speed, acceleration, distance and time. Their data set was collected by 81 participants during a two weeks data collection phase. Biljecki et al. [2013] classified GPS using fuzzy concepts from expert systems to distinguish between 10 transportation modes. They ran their algorithm on a 17-million point data set collected in the Netherlands and Europe and get an accuracy of 91.6 percent determined with the comparison of the classified results with manual classified reference data.

In the context of agricultural vehicles Jensen and Bochtis [2013] presented an approach for recognition of operation modes of combines and transport units. Their aim is the generation of methods for automatic recognition of operating modes of all machinery types involved in grain harvest, based on raw GNSS trajectories. Their approach is mainly based on geometric algorithms that recognize if the machine is harvesting or not by moving on non harvested parts on a field. Unloading operations were detected by a very low or zero speed difference between the unloading machine and the transport unit and a maximum distance threshold of both vehicles. Further preliminary work is done by Grisso et al.

[2002] who used DGPS and a yield monitor to derive statistics for five example fields aiming the analysis of field efficiency determination. Their work is therefore focused on infield operations while the main focus in our work lies in the general classification of trajectories including the road and on farm parts. Hence the extracted field trajectories of our work can deliver an input data set for this work.

On some of the probed machines there are installed switches for working and not working mode. Further data is given by a proprietary algorithm that distinguish the working mode for every machine. Figure 3.7 shows the stored working mode classification from the original CLAAS Telematics data. Several roads are classified in “*Working Mode = on*” and also a greater amount of measurement points within the field area are set as “*Working Mode = off*”. Through visual analysis from expert perspective, these measurements are wrong classified. While the “*Working Mode = on*” on a road is obviously wrong, the decision if the “*Working Mode = off*” within a field boundary is correct can only be made by detailed analysis of the driven trajectory. Transit- or transportation maneuvers can also take place on field areas. However, this data contains too many wrong classified measurements for later processing steps as none of these methods is reliable due to either human mistakes or algorithmic failures.

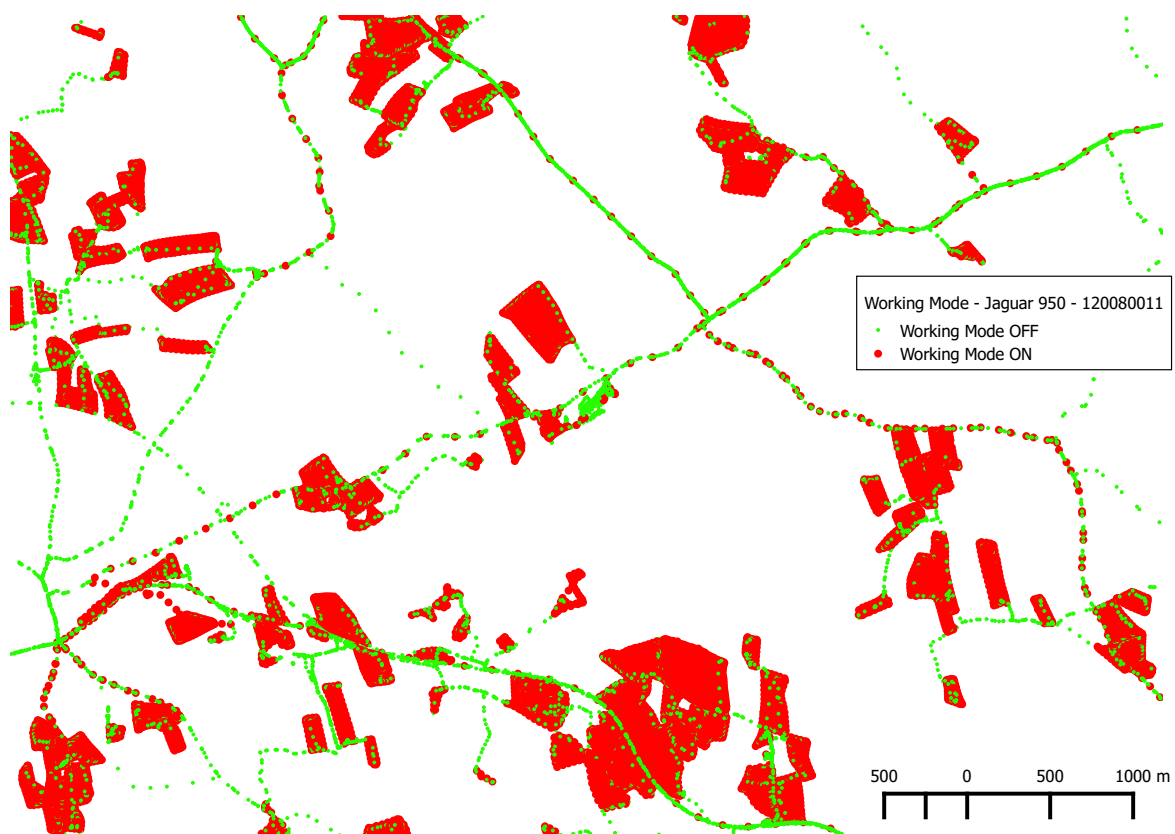


Figure 3.7: Working Mode for Jaguar 950 forage harvester

For a more general overview figure 3.8 shows the processing chain with focus on field and road computation that follow the preprocessing steps from this chapter and make use of its generated data.

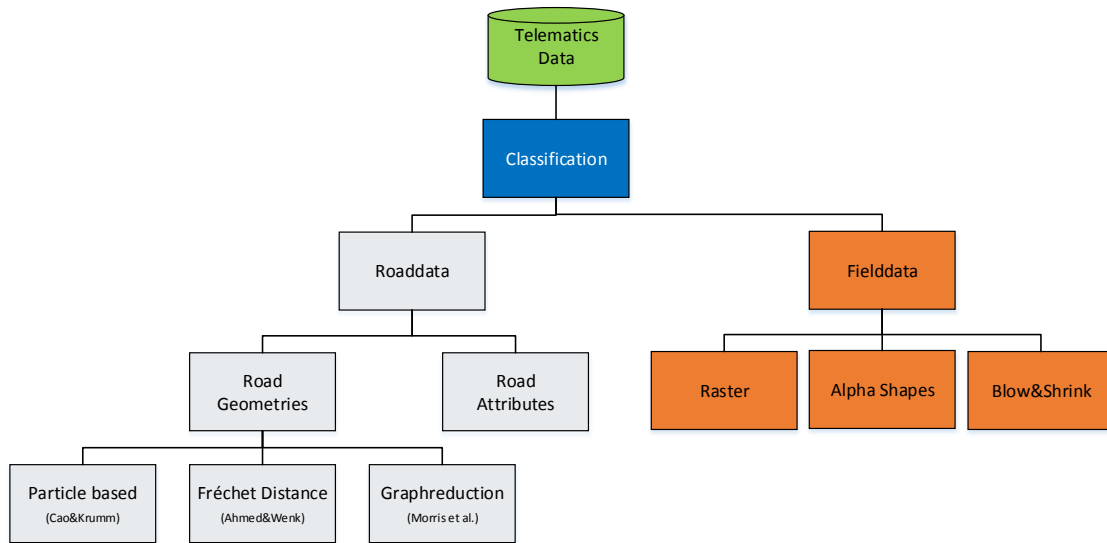


Figure 3.8: Workflow - Dataprocessing workflow with algorithms for road and field computation

3.5.1 Reference Data

Generating correct reference data is crucial for good classification results. Hence a preliminary step for digitizing of reference data is needed. This work is done manually, which means additional and time consuming work. Another option would be to generate this reference data by data collection on the machine e.g. as a tagging app for the driver. As the second option is desirable but not available for the CLAAS Telematics data, a practical solution for digitizing the ground truth is needed. The manual classification is done visually using a base map (e.g. satellite photos or maps). The work is done by experienced mappers who are aware of driving maneuvers and agricultural processes on road and on field. This guarantees a qualitatively good reference data set.

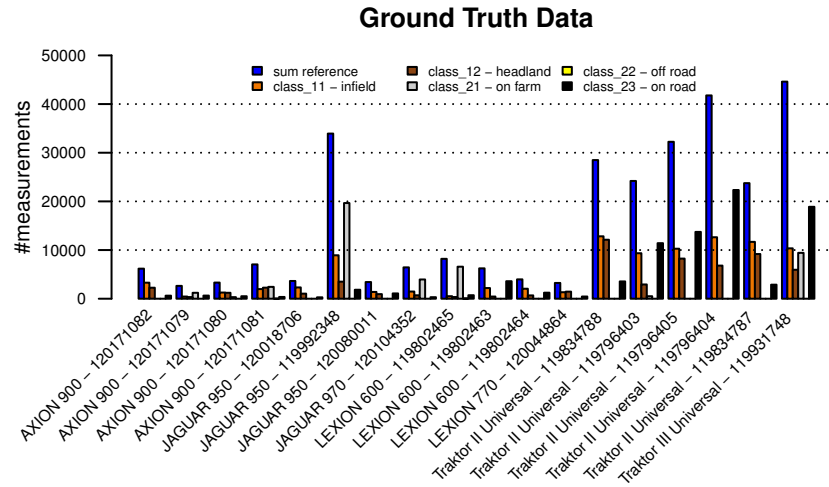


Figure 3.9: Ground truth distribution for all classes

Figure 3.9 shows the numbers of digitized ground truth measurements for each class, separated by machines. Due to the unequal distribution of ground truth data, some remarks are needed to explain the ground truth data. The off-road class is underrepresented for nearly all machines. Tagging of off-road samples is very difficult and hardly possible in very dense point and trajectory clouds. Therefore a filtering step before tagging the measurements is needed to extract these measurements properly. In this work, this work intensive step is omitted because this problem would not affect the further processing as the focus lies mainly on the road-field distinction. In the manual digitizing process the different working modes and therefore differing movement types are considered as far as possible. A generally good reference data set is the result. It has also to be considered that a further integration of machines would also bring additional work on the mapping of reference data for them. Thus, the efforts for generating this data should be as low as possible but also as high as needed to produce a representative data set for each machine.

3.5.2 Classification Algorithm - KNN

To distinguish between road and field data, a supervised classification algorithm is used. Lauer et al. [2014] showed a KNN based approach to classify road, field and off-road measurements for a single machine within a defined area. The KNN classification is a parameter free method for probability density estimation [Altman, 1992]. Based on this estimator the algorithm determines the class for a chosen sample considering the k-nearest neighbors. In this first approach, the classification is limited to the three classes of infield, road and off-road (which represents movement mainly between fields on “not road like structures” or open space). The training set digitized for this approach is a spatially and temporally limited data subset where all measurement points are manually classified in the three classes.

3.5.3 Classification Algorithm - Random Forest

The Random Forest classification algorithm was firstly named by Breiman [2001]. In this publication he developed an approach for a statistical motivated decision tree model. Using the complete reference data, the random forest algorithm takes a random, fixed size subset of the data and builds one of k decision trees. This is done k -times which builds a random forest of decision trees. The complete reference data is finally used to build the forest. To calculate the classification error, the complete reference data will be classified by the random forest classifier. Although this method is common practice, it has also be considered that there is a bias due to using partly the same data for training and evaluation. On the whole there is nevertheless a clear tendency given by the evaluation. Regarding the classifier's configuration, a reliable number of trees (the k) has to be found. In this thesis, the convergence of class prediction error is used to derive a value for a proper number of trees.

In the context of this thesis, two implementations of the random forest algorithm have been focused on. For the analysis the R randomForest implementation is used. As R brings a lot of various plotting functionality and implemented statistical analysis, this is a suitable way for the analysis, even if this breaks the workflow of the independent Java framework. The second implementation is the java library Java-ML [Abeel et al., 2009] which also provides an implementation of the Breiman [2001] RandomForest algorithm. This is fully integrated in the workflow, but has less functionality to generate statistical plots.

The importance measures for the classification attributes were generated with the R randomForest package. The mean decrease accuracy is calculated from the OOB (out-of-bag data) as follows: "For each tree, the prediction error on the out-of-bag portion of the data is recorded (error rate for classification, MSE for regression). Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences. If the standard deviation of the differences is equal to 0 for a variable, the division is not done (but the average is almost always equal to 0 in that case)." [Liaw and Wiener, 2002].

The data is classified in three independent runs:

1. Using the whole reference data of all machines
2. Using the reference data separately for each machine type
3. Using the reference data separately for each machine

With this splitting, the differences of classifications (whether they are present and how) needs to be investigated and if a whole reference data set for all machines would be feasible to classify the data of all agricultural machine types (combine harvester types, forage harvester types and tractor types). To compare the three classifications, the indicators for attribute importance and error convergence are produced by the out-of-box error are used.

Classify all machines together

The first attempt is the classification of the whole reference data in one step for all machines.

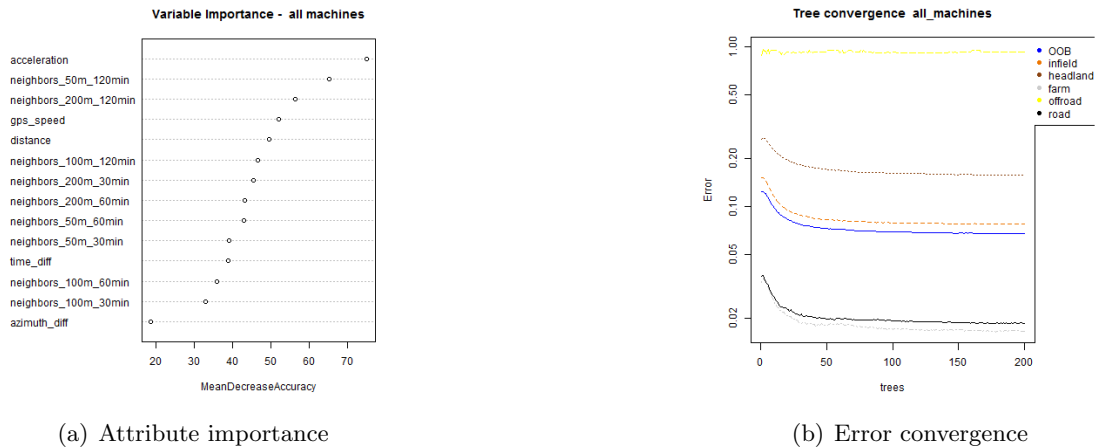
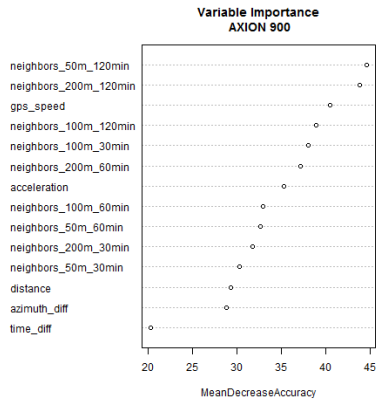


Figure 3.10: randomForest results - all machines

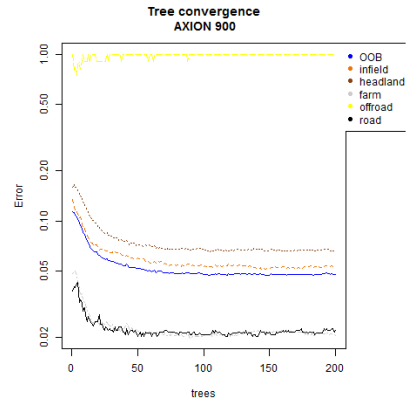
The variable importance for the whole data set in figure 3.10(a) shows the highest ranked attributes “acceleration”, “neighbours_50m_120min” and “neighbours200m_120min” as the most influential attributes. The “time_diff” value (which usually has a constant value and changes only at longer stops when the machines are turned off) has a very low influence on the classification. Also “neighbours_100m_60min”, “neighbours_100m_30min” and the “azimuth_diff” have a low influence on the classification. Apart from the off-road class error which is constantly approaching 100% the other classes converge on an acceptable accuracy of less than 20% . The number of trees that are needed before no big raise of accuracy is expected lies around 100 trees and decreases rapidly from 1 to 50 trees. The class accuracy of the road class (with a value of less than 2%) is very good and gives therefore good results for later processing steps (see figure 3.10(b)).

Classify separately for each machine type

In the second attempt, the data is separated by machine type. The machine type is defined in this context as a specific model and type of a manufacturer (e.g. CLAAS AXION 900). It is expected that errors are machine dependent and a separation into machine types will give a more specific insight on possible relevant attributes.

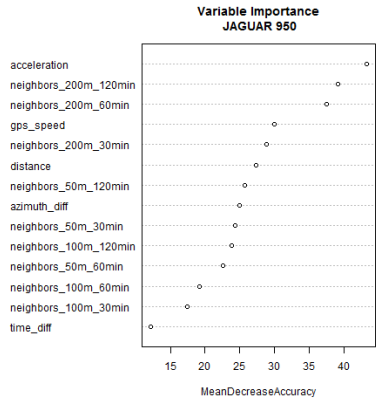


(a) Attribute importance

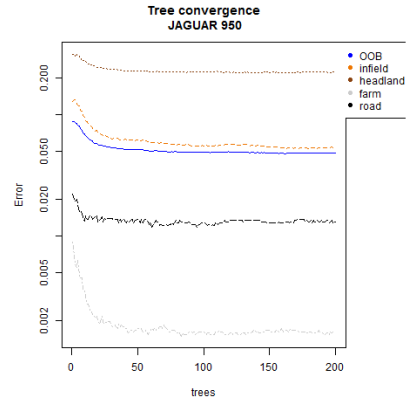


(b) Error convergence

Figure 3.11: randomForest results - Axion 900

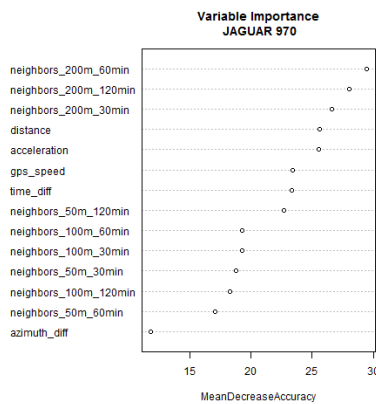


(a) Attribute importance

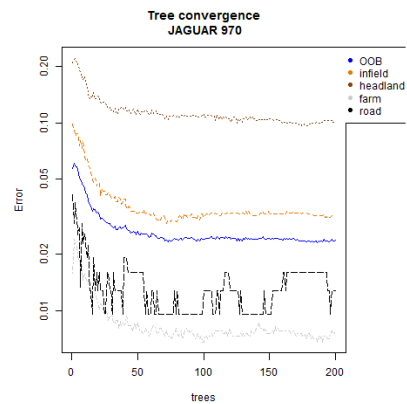


(b) Error convergence

Figure 3.12: randomForest results - Jaguar 950



(a) Attribute importance



(b) Error convergence

Figure 3.13: randomForest results - Jaguar 970

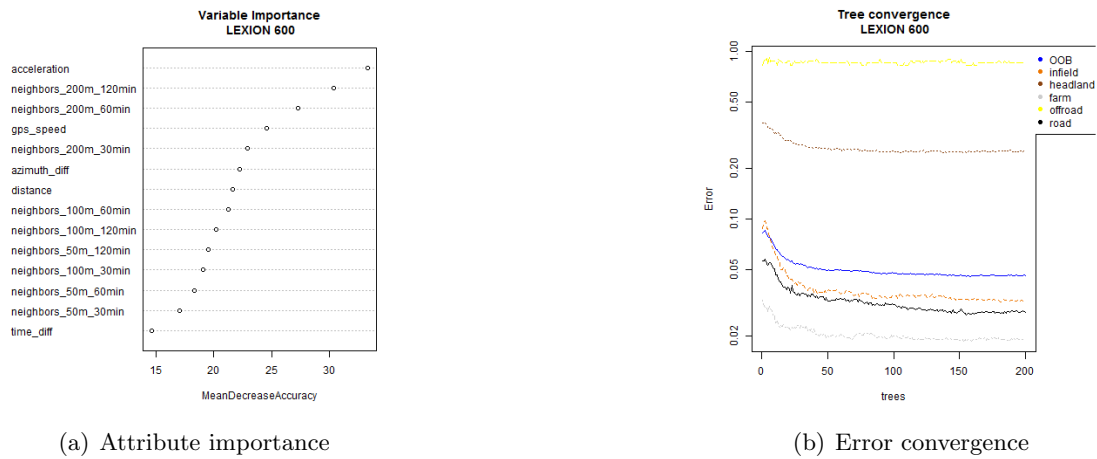


Figure 3.14: randomForest results - Lexion 600

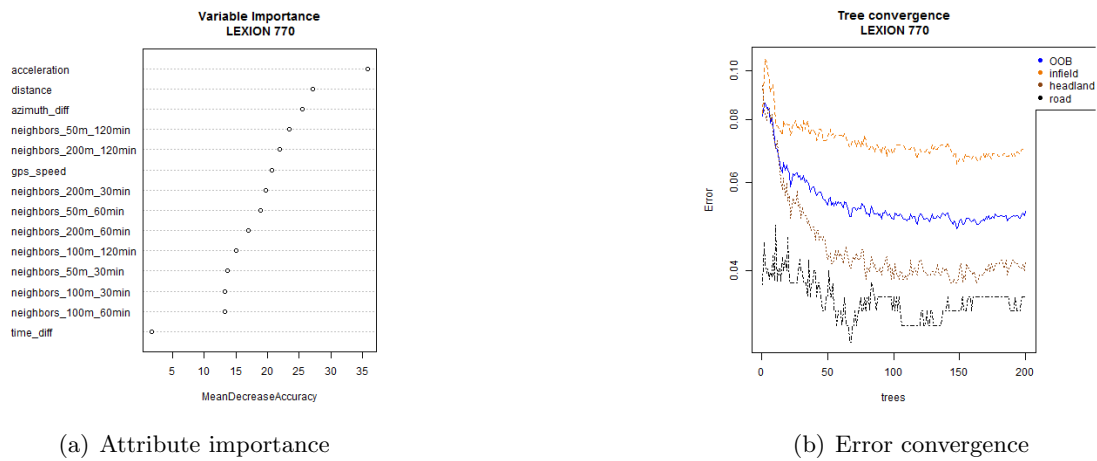


Figure 3.15: randomForest results - Lexion 770

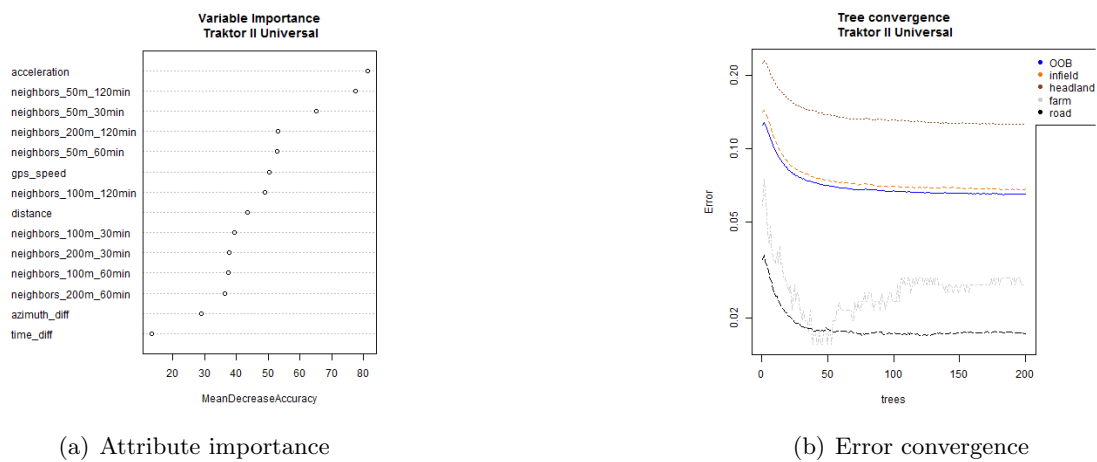


Figure 3.16: randomForest results - Traktor II Universal

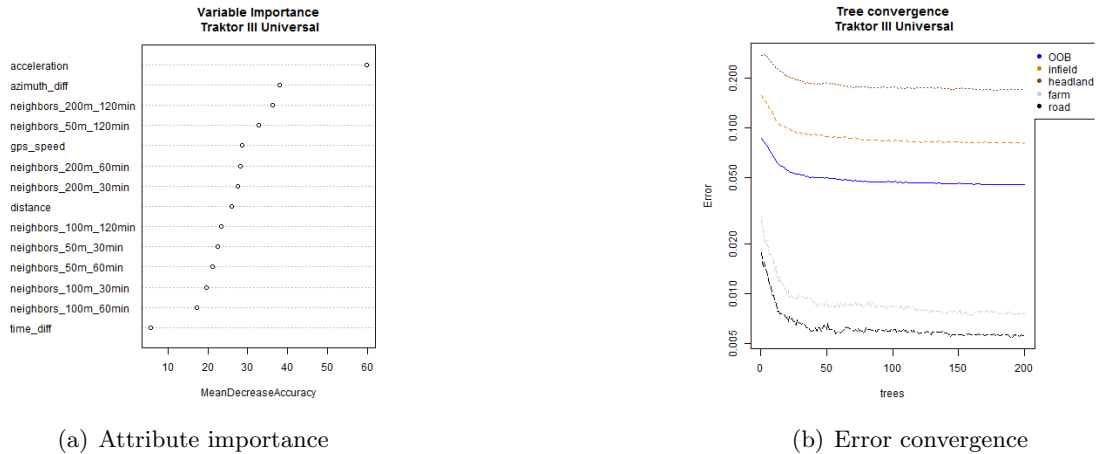
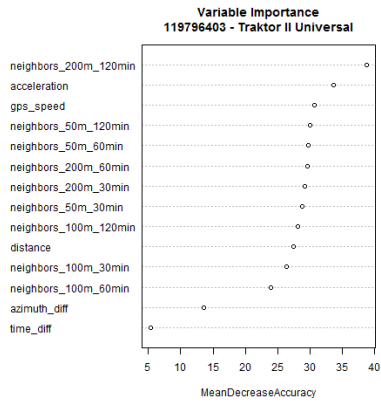


Figure 3.17: randomForest results - Traktor II Universal

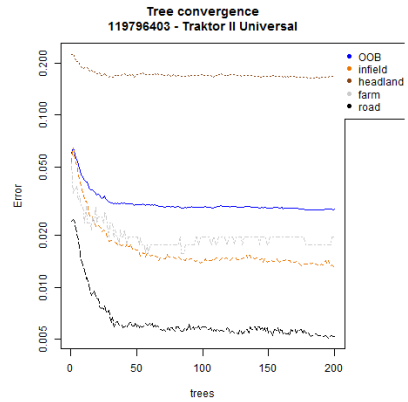
Reference data for the off-road class only exist for the “AXION 900” and the “LEXION 600” machines and the classification errors are high, independent from the number of used trees. This shows a not well separated class which was also expected regarding the low number of reference measures and the (also for experts) not easily separable classes. The “farm” class is only available for the machine types “JAGUAR 950”, “JAGUAR 970” and “LEXION 600”. Its error decreases rapidly and converges to less than 2%. Disregarding the “LEXION 600” class which has very high errors for headland, the classification errors are less than 20% and for most classes also less than 5%. As shown in the attribute importance figures, the most important attribute for classification is the “acceleration” value which is within the top ranked three attributes, usually as first ranked. “gps-speed” and spatiotemporal neighbourhood are also important attributes for classification. Less important are “time.diff” and several spatiotemporal measurements depending on the machine type.

Classify separately for each individual machine

The last and most specific attempt is the usage of separate data for each individual machine. As manual data classification for test- and training-data is very time consuming, this is the most work intensive approach. The advantage of this approach is the availability of a machine specific reference data set.

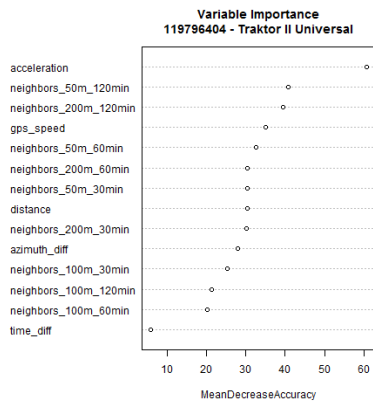


(a) Attribute importance

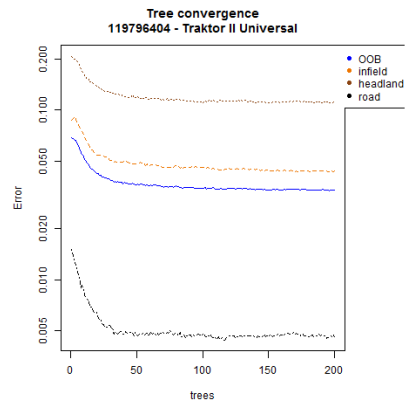


(b) Error convergence

Figure 3.18: randomForest results - 119796403 - Traktor II Universal

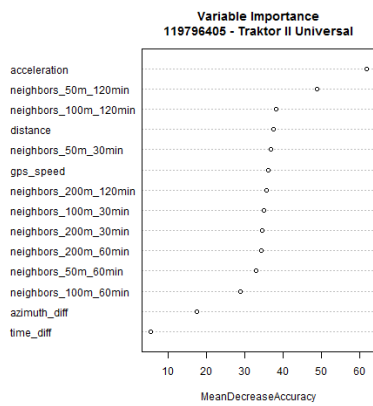


(a) Attribute importance

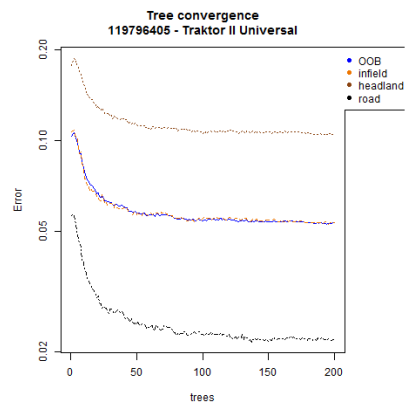


(b) Error convergence

Figure 3.19: randomForest results - 119796404 - Traktor II Universal

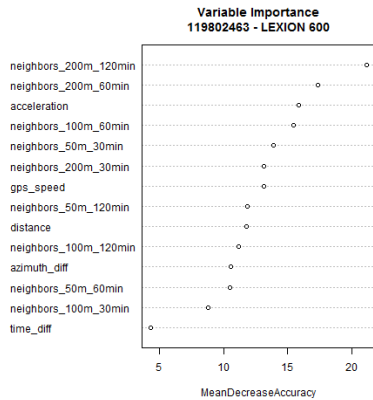


(a) Attribute importance

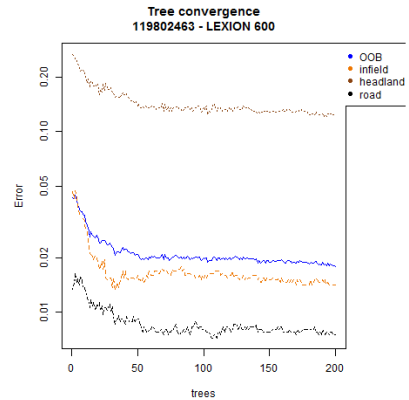


(b) Error convergence

Figure 3.20: randomForest results - 119796405 - Traktor II Universal

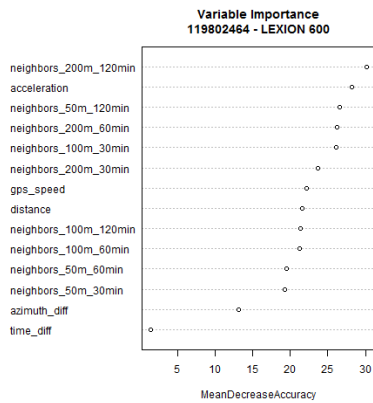


(a) Attribute importance

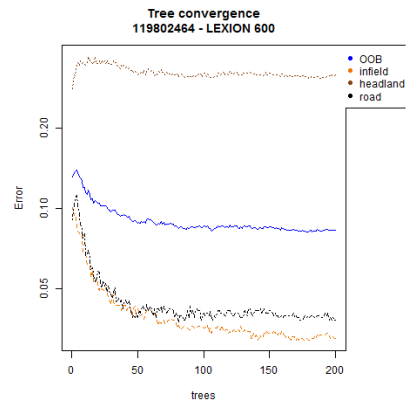


(b) Error convergence

Figure 3.21: randomForest results - 119802463 - Lexion 600

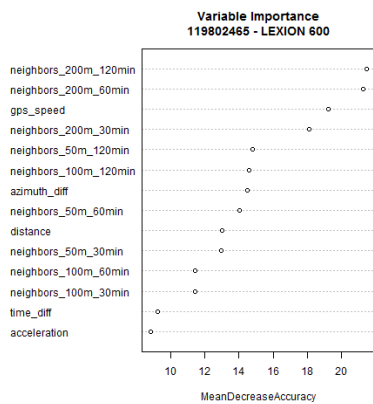


(a) Attribute importance

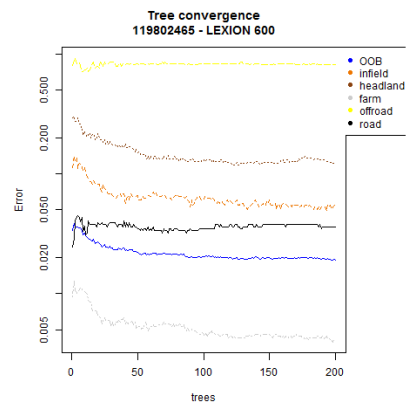


(b) Error convergence

Figure 3.22: randomForest results - 119802464 - Lexion 600

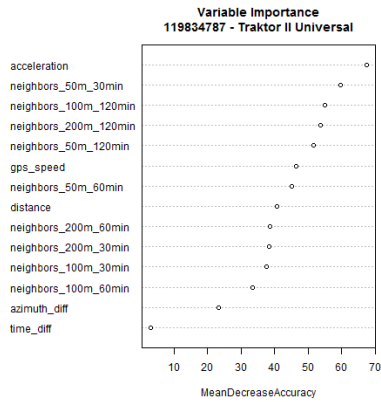


(a) Attribute importance

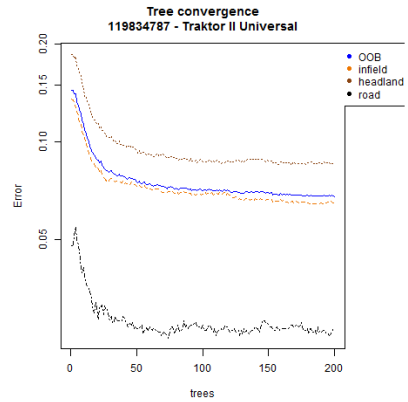


(b) Error convergence

Figure 3.23: randomForest results - 119802465 - Lexion 600

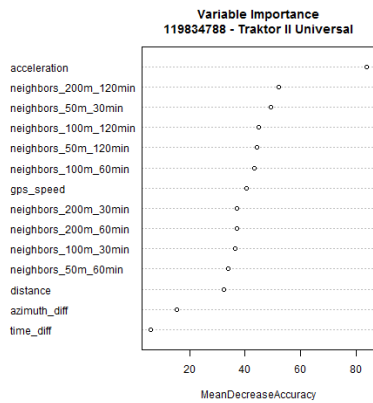


(a) Attribute importance

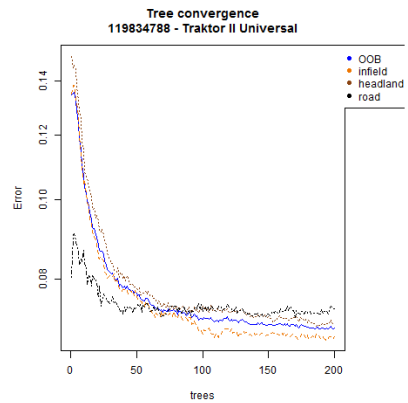


(b) Error convergence

Figure 3.24: randomForest results - 119834787 - Traktor II Universal

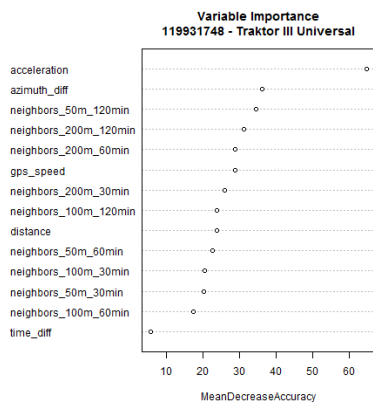


(a) Attribute importance

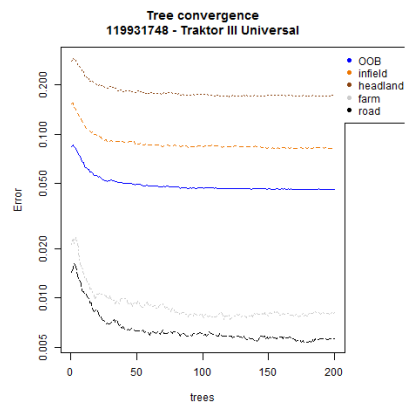


(b) Error convergence

Figure 3.25: randomForest results - 119834788 - Traktor II Universal

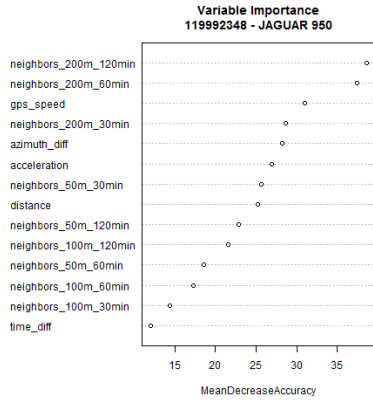


(a) Attribute importance

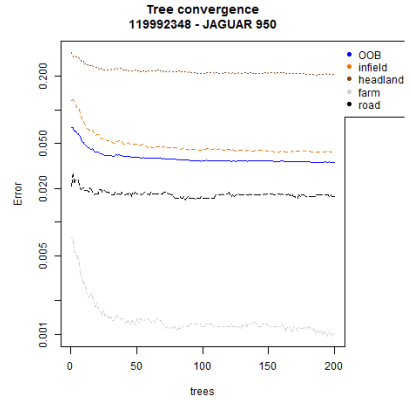


(b) Error convergence

Figure 3.26: randomForest results - 119931748 - Traktor III Universal

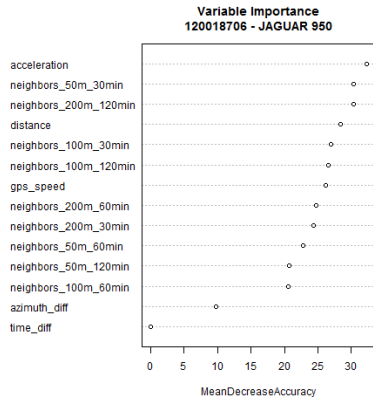


(a) Attribute importance

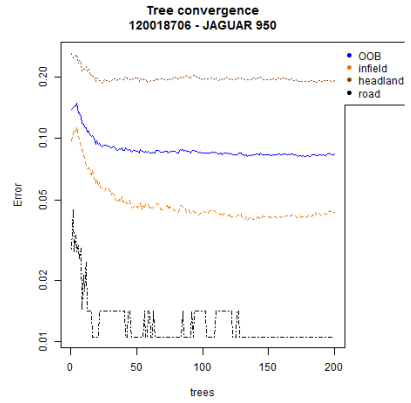


(b) Error convergence

Figure 3.27: randomForest results - 119992348 - Jaguar 950

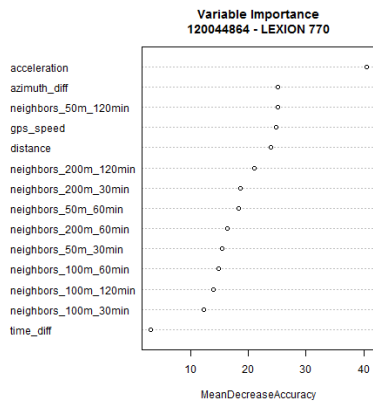


(a) Attribute importance

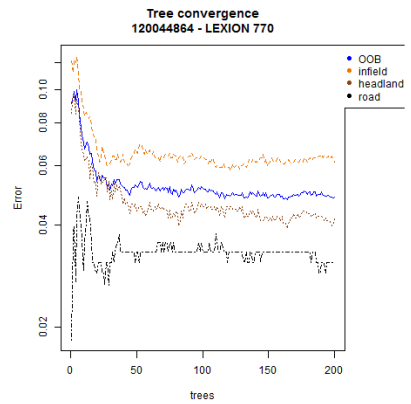


(b) Error convergence

Figure 3.28: randomForest results - 120018706 - Jaguar 950



(a) Attribute importance



(b) Error convergence

Figure 3.29: randomForest results - 120044864 - Lexion 770

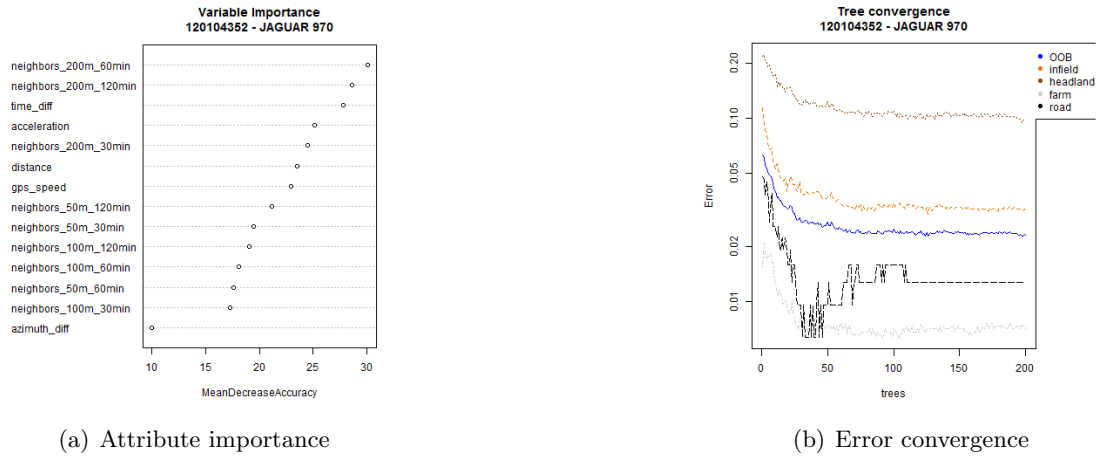


Figure 3.30: randomForest results - 120104352 - Jaguar 970

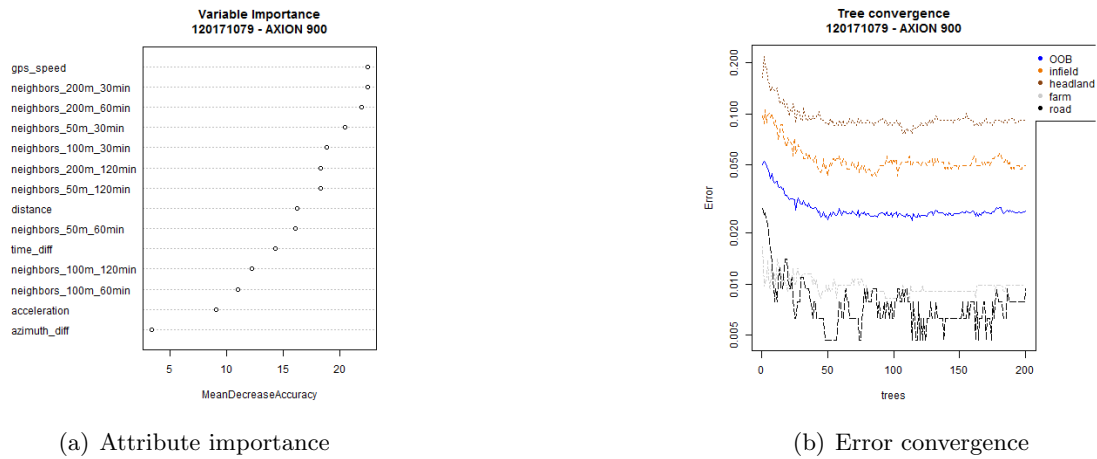


Figure 3.31: randomForest results - 120171079 - Axion 900

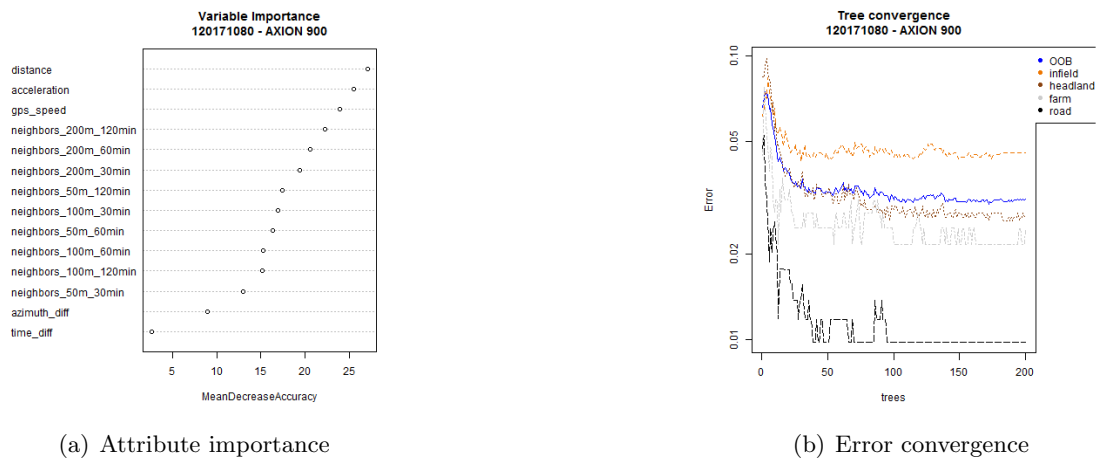


Figure 3.32: randomForest results - 120171080 - Axion 900

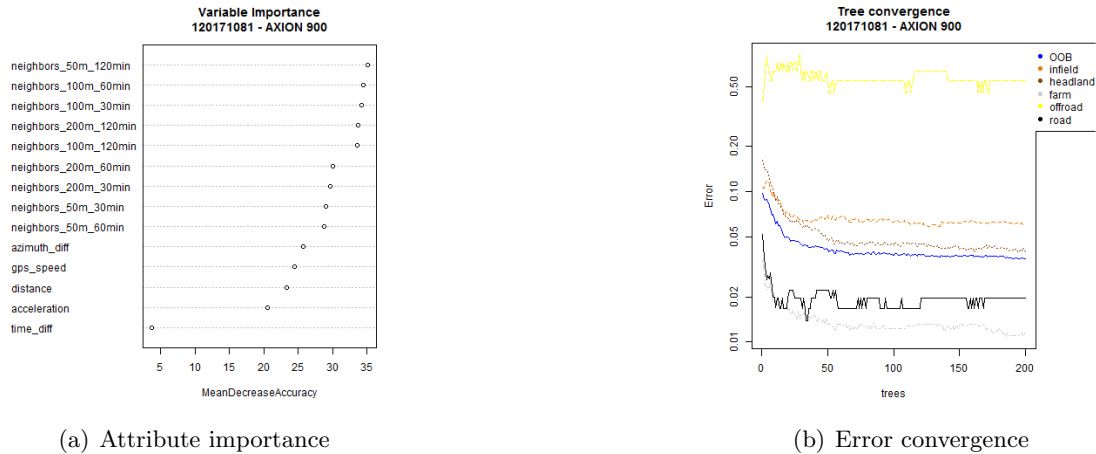


Figure 3.33: randomForest results - 120171081 - Axion 900

The individual classification shows the variable importance and classification accuracy machine specific. Aside from the two “JAGUAR 950” machines and one of the “AXION 900” the “time_diff” attribute is not an essential attribute for classification. For most of the machines, the “acceleration” attribute is one of the most important measures. Outliers are the “AXION 900” tractors and one of the “LEXION 600” harvesters where this value is ranked in the last positions in importance ranking.

Comparison of the Classifications

The three presented approaches for processing show that a differentiation of individual machines could improve the results. As the collection of individual reference data is very work intense this step should only be done for testing and validation purposes. For most machines, the classification results are proper or have a sufficient quality for later processing steps. Differences between the three approaches exist and show that the more individual the reference data, the better the classification. The variable importance shows only small jumps in all three approaches which seems to be no real breaks. Hence there are no attributes that are really unimportant for the classification step in general. The variable importance is machine and machine type dependent. Therefore attributes that are unimportant for one machine can be important for the classification of another.

The influence of the reference data must not be concealed. This step is one of the most influential tasks in this process. Getting a well representative reference data set is surely a very influential if not the most influential part of the classification. With the Random Forest approach, a statistic based and popular method for data classification is used which is transparent for users and later analyses. The results of the classification are visibly better than those from the KNN approach [Lauer et al., 2014], although these two approaches are not easy to compare since the KNN approach has a smaller set of target classes and is only based on data from one machine.

To get more reliable reference data, a tagging step should be implemented on the machine while driving. This can be realized as a tagging application for some trips where drivers can indicate which work steps they are doing. This effort is only needed to test the classifier and correct it if the classification results are too defective to get good processing results. With this step, the machine learning approach should improve the results and

avoids the classification step done by experts using a GIS software.

3.5.4 Post Classification

The objective of the classification algorithm is the generation of trajectory segments with continuous modes. Due to the classification of individual measurements, suddenly changed labels within longer series of equal working mode are not realistic and should be considered as outliers. Especially the beginning of trajectories is tracked by this wrong classifications as values for speed etc. are not computable due to missing predecessors. Therefore, a post processing filter technique is needed to filter these outliers and bring them to a suitable value. In the classification of raster data sets this technique is denoted by sieve [Exelis VIS, 2015].

Listing 3.1: Filtering separate classified machine data by 3-window filter to eliminate outliers

```

SELECT m.m_id,
       CASE
         WHEN Lead(mea.mea_predict_r_rf_200_sep_machines)
              OVER (
                ORDER BY m.ma_id, m.date) =
              Lag(mea.mea_predict_r_rf_200_sep_machines)
              OVER (
                ORDER BY m.ma_id, m.date)
         AND Lag(mea.mea_predict_r_rf_200_sep_machines)
              OVER (
                ORDER BY m.ma_id, m.date) <>
              mea.mea_predict_r_rf_200_sep_machines THEN
              Lag(mea.mea_predict_r_rf_200_sep_machines)
              OVER (
                ORDER BY m.ma_id, m.date)
         ELSE mea.mea_predict_r_rf_200_sep_machines
       END
INTO   mea_predict_r_rf_200_sep_machines_filtered
FROM   measurements_ext_attr mea,
       measurements m
WHERE  m.m_id = mea.measurementtype_m_id
       AND mea.generic_attributes IS NOT NULL

```

Listing 3.1 shows the SQL-statement for filtering the classified *measurements*. A window size of three measurements is used for data filtering and correcting the current value with the surrounding two values if the successor and predecessor are equal and the current value is not equal to one of them. This eliminates outliers that otherwise have a negative effect on following trajectory segmentation (e.g. unrealistic interruptions).

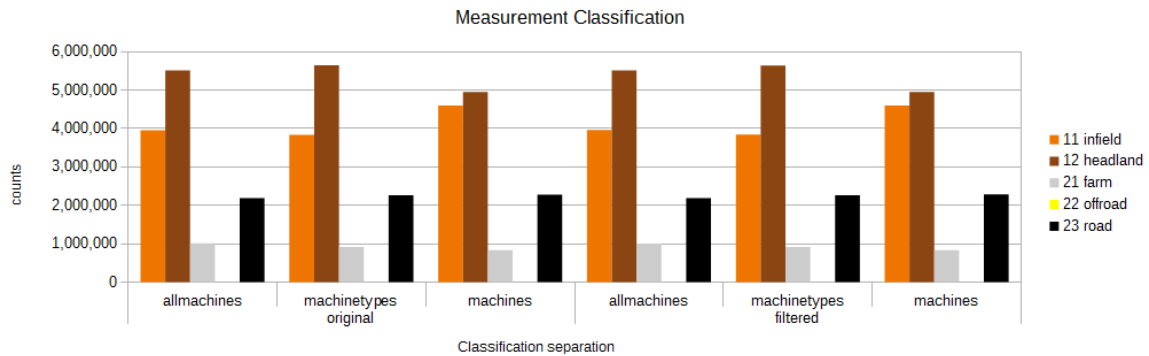


Figure 3.34: Classified measurements before and after sieve filtering

The filtering prevents a fragmentation resulting from outliers and brings the advantage of continuously connected track segments (filtering eliminates single outliers) and data problems that will be produced by trying to generate line strings from single points in the following step.

3.5.5 Track Segmentation

The measurements are classified by using the previously explained classification algorithm. In a following step, the measurements are then grouped by machine and ordered by timestamp. This order generates a list with succeeding measurements of each machine. From this list, a set of tracks is generated by splitting the succeeding measurements on positions where a predefined time threshold is exceeded. This list is then separated in *Tracks* which represents a continuous trajectory without big time gaps (the split criteria is a gap of more than 31 s which means, that for a tracking frequency of $1/15$ Hz a new track will be started when two chronological following measurements are missing). This processing is done by the *TrackExtractor*. The next processing step is done by the *TrackSegmenter* where each *Track* is split in equal classified runs of measurements. Each measurement in this list is then checked in order and from each run of equally labeled measurements a new *track segment* is generated. The architecture of the track generation and track segmentation allows the segmentation of tracks in multiple ways. Hence a *Track*-object consists of at least a complete list of measurements representing the whole track (the original, not classified) and (if classified) of *TrackSegments* of one or more classifications. Figure 3.35 shows two sample *Tracks* and the corresponding classified *TrackSegments*.

3.6 Estimating Main Working Direction

The main working direction of an agricultural field is the average direction of the field, usually the azimuth angle of its longer side. This parameter is mainly used for the GIS analysis of tillage (farmers in the European Union are legally obligated to organize their field work in an erosion debilitating way). This commitment was made in context of the Common Agricultural Policy (CAP) of the European Union in terms of the Cross Compliance direction [Council of European Union, 2009] and converted into national laws. Also, for generating infield route graphs, this parameter is beneficial for computation of practically feasible and realistic in-field directions (cf. 6.1.4). In Bochtis et al. [2010] the

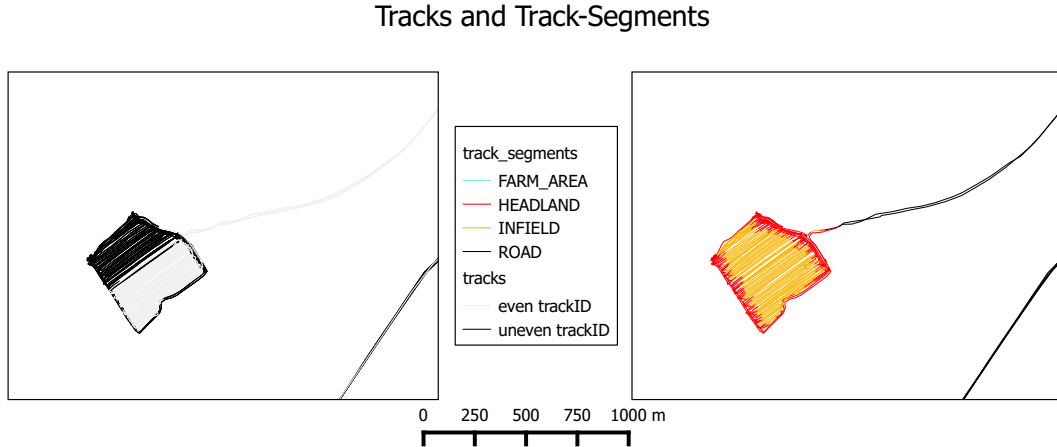


Figure 3.35: Sample tracks and their track segments

main working direction is also an important parameter to compute a harvesting strategy for machines on agricultural fields. In this section an algorithm for calculating the main working direction on the driven trajectories on field is presented. This differs from the method of computing the main working direction from a given field boundary. The advantage is that with the trajectory data we have the knowledge of the driving processes on the field and can therefore compute a main working direction that represents the direction that is physically driven. Especially for operations that rely on existing machine tracks, this strategy is a benefit compared to methods that do not consider driven trajectories.

3.6.1 Statistical Calculation of the Main Working Direction

The main working direction is estimated with three methods:

1. **Average**

$$avg(x) = \frac{1}{n} \sum_{i=1}^n x_i$$

2. **Median**

$$median(x) = \begin{cases} x_{\frac{n+1}{2}} & n \text{ odd} \\ \frac{1}{2}(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}) & n \text{ even} \end{cases}$$

3. **Histogram peak - Modus**

$$maxHist() = max(m_i)$$

with $n = \sum_{i=1}^k m_i$ (n =#observations, k =#bins, m_i =#measurements in i -th bin)

with $k = \frac{\max(x) - \min(x)}{h}$ and $h = \text{width}(\text{bin})$



(a) Rectangular Boundary



(b) Complicated Boundary

Figure 3.36: Main Working Direction (red arrow) for rectangular fields and for fields with a more complex boundary

For each method, the measurement points that are used to calculate the field boundary are used as input data. The azimuth data is then normalized to $[0;180]$ to get one value for the direction (instead of forward and backward run). The average is calculated for the set of azimuth values of the measurements. In a second method the median is calculated for this set of measurements also using the compressed range $[0;180]$. The third method, a

further approach using the histogram for azimuth values is implemented. The histogram is computed defining a resolution of 1° in a range $[0^\circ - 180^\circ]$. Using a window function, the global maximum is calculated. For efficient working processes, the fields are commonly oriented in a way, that allows long, parallel lanes. The global maximum is therefore a more reliable estimation for the main working direction as the result is mainly influenced by these multiple long parallel trajectories within the field (figure 3.36(a)). This approach will also return a plausible value for complex field geometries with exceptional headland zones (figure 3.36(b), at least for one part of the field. Due to the fact that these fields are not the standard and usually farmers tried to organize the fields in rectangular boundaries this approach fits on the requirements for following processes and gives a reasonable value for a main working direction. However, to calculate more reliable values for these more complex fields, their geometries have to be split into sub-fields and the modus has to be calculated for each sub-field. As these fields are only exceptional cases, the focus lies on usual field boundaries where this approach fits well. A further advantage of this approach is the filtering of the primary peak disregarding smaller peaks or accumulations of values in a broad section that can have large influences on the median and average values. For the main working direction only one value is needed and this has to be the one with the most counts.

3.7 Discussion

The methods that are used in this chapter surely have impacts on further processing steps. A discussion on these possible impacts is therefore essential. The presented map matching approach is feasible and fits the requirements on matching the used telemetry data. To avoid wrong classifications (e.g. parallel track to arterial road) a more sophisticated approach like a Hidden Markov Model based map matching and the integration of semantics could produce better matching results and thus improve the further processing. However this would increase the complexity and the computation time. The applied approach therefore is an agreement on complexity and computational effort.

The extended attribute calculation uses different formulae to calculate the movement attributes. These methods are highly influenced by the quality of the GPS signal and the resulting positional accuracy. Also the timestamps and rounding errors have an influence on the computation of the extended attributes. Computing the spatiotemporal neighborhood is very time consuming and this method should be improved by using more efficient index structures to get faster results for big data sets. However, the presented methods are usable and working with the test data set in a productive manner.

Calculating the bearing change could be improved using denser measurements. Especially for curvature calculations in headland areas the time increments of 15 s are not feasible for curvature detection. Integrating denser measurements will cause an adaption of the window size to recognize turning sequences in headland area.

The classification of measurements is an inevitable pre-processing step required for further data processing. The results of the different classifications show small improvements separating the data by machine type or doing the classification for each machine individually. The individual processing increases the costs due to the requirement of collecting individual training data. The calculation of different spatiotemporal attributes are important but it is assumed that the number and granularity of these factors depends on the region and the structure of road network and the size and geometry of the agricultural fields. The learning approaches obtain good results that fulfill the requirements for further

processing. It is also shown that the calculation of attributes is based only on movement data and its derived values. The proprietary measurement values that are quite heterogeneous (machine types, sensors, manufacturers), erroneous naming and not standardized sensor measurements constrained the usage of this data.

During the research on measurement classification and attribute calculation several beneficial side results have been generated. Especially the estimation of a main working direction will be an important attribute that can be estimated from vehicle trajectories and will improve the later calculation of infield routing graphs. The handling of complex field structures containing inner field structures with appropriate, different main working directions could achieve further improvements. Due to the fact, that these complicated field structures are exceptional cases, the handling of these cases have to be investigated in a separate work.

Chapter 4

Generation of Road Network using agricultural Telemetry Data - Geometric refinement

The first step to generate road data is the generation of an initial geometry and the construction of a graph structure which enables the data for routing. Especially for rural areas where digital road data is not permanently maintained and countries with many white spots on their digital map, the initial automatic generation of geometries is an important step to a more complete digital road network. In areas where this data already exist, a refinement can be useful, since road structure and data is continuously changing. This chapter provides an overview on the state of the art of road network generation and refinement algorithms and their usability for extracting a rural road network from movement data of agricultural vehicles. A subset of different methods will then be parametrized and the different characteristics of the algorithms using the agricultural telemetry data will be shown. Road geometries and network topology will then be the basis for further attribute refinement (e.g. road surface, road type, number of lanes, street names, ...). This work is not part of this thesis but it should be pointed out that further investigations have been made in Lauer et al. [2013, 2011].

4.1 State of the Art

At the beginning, this task has been strictly in the hand of land surveyors, usually governmental institutions that are mainly responsible for official maps and thereby road data. The majority of early approaches based on detecting linestring features in remote sensing images. Fischler et al. [1981] were one of the first that developed an approach for delineating roads and "line-like" features in low-resolution aerial imagery using raster algorithms. Recent research approaches like Predoehl et al. [2013] used a statistical model to infer recreational trails from aerial images. Further approaches that used orthophotos were Fortier et al. [2014] and Hu et al. [2007]. The main problem with these approaches lies in the unavailability of local information. Especially roads in forest areas are covered with canopies what makes it impossible to detect them in aerial imagery.

With the availability of accurate position data from GPS, researchers started using GPS measurements and trajectories to generate and update maps. Rogers et al. [1999] from the former Daimler Chrysler Research and Technology Center presented one of the

early approaches that mined trajectories of DGPS equipped cars to augment road models. In their work, they collected 44 position traces along a 15 km section on a Californian Interstate Highway. The frequency of collected positions was 2 Hz, which means a very high spatial density for common car velocities. This work also reveals the accuracy of the used NavTech, Inc. road data set by comparing the road center lines and a resulting average position error of about 7 m. Their averaging method calculates nearest points on driven trajectory for each point in origin road feature (which base points are set every 10 m, partly interpolated to get a more dense structure). The resulting new map point p is the average of the origin map point m and the trajectory point n weighted by m_σ and n_σ .

Morris [2002] combined aerial imagery with GPS data to derive roads from aerial imagery and correct the data using GPS tracks. Another work of Morris et al. [2004] is focused on recreational trails for hiking and cycling in off-road terrain, where off-road means small paths and single trails that are preferably used by hikers and bikers. They collect data coming from common GPS loggers that exports GPX format and build a graph by intersecting the tracks to a plain graph. To prune the numbers of trails for one path they processed the graph with three graph reduction algorithms: Parallel reduction, serial reduction and face reduction. Their reduction heuristic is organized as a loop with a face reduction surrounded by a combination of one parallel and one serial reduction process. This loop will determine if no more reduction process is possible. The result is a plain graph representing all used paths. As reductions are parametrized by distance thresholds, optimized thresholds for input data are essentially for a successful processing and a reliable result. The algorithms that have been used in Morris et al. [2004] are implemented in the proprietary Topofusion software.

Edelkamp and Schrödl [2003] originally described a k-means based approach for navigable road map generation from GPS traces. Schroedl et al. [2004] presented a workflow for map refinement based on GPS traces. They gave a list of possible applications for improved and more detailed digital maps like adaptive cruise control, lane departure warning, lane-level navigation and dynamic lane closure warning. Most of these applications are available nowadays in transport vehicles, luxury class and medium-class vehicles by using real time environment detection by vehicle sensors, lesser high accurate digital map as these maps are not available in the required quality. The main contributions of their article consists of three parts: (1) a spatial clustering algorithm that infers the connectivity structure of a map from scratch to avoid the requirement of an initial input map, (2) a lane clustering algorithm to handle lane splits and merges and (3) an algorithm to inferring detailed intersection models of roads. They split the so called traces (trajectory generated by motion of a single vehicle) into subsection that correspond to road segments (graph edges as a connection from node to node). Further, they consider the intersections not as points (as it is common for most commercial maps), but as a structured region for a more proper modelling (e.g. to represent turn restrictions). As data Schroedl et al. [2004] used collected DGPS traces which are optionally aggregated with wheel speeds, accelerometers etc. using a Kalman filter [Kalman, 1960] as described in Harvey [1990] for smoothing. After preprocessing the measurements (smoothing and filtering noise) they partitioned the traces into segments by matching them to a base map. Using spline fitting, a reference line as approximation for all measured points of this segment and as reference line for lanes is calculated. The splines are defined by a variable number of control points which is determined by the second derivative of the spline at constant intervals. The average of these values is defined as the curvature error. The aim is to find a good trade-off between

best fit and good curvature that represents a typical road geometry. The number of lanes is identified by the perpendicular offsets of sample points from the calculated centerline. In the last step, the intersection geometry and the lane transitions between adjacent segment boundaries are refined. For this, they calculate attraction forces of each endpoint in the intersection region which is defined by the concept of snake [Kass et al., 1988]. This approach is a refinement of the Edelkamp and Schrödl [2003] publication with the additional focus on refinement the finding and placement of road intersections and modelling individual lanes and the transitions between them. Furthermore, they used a spine-fitting technique to compute the geometry of the final turn-lane for an intersection. Brüntrup et al. [2005] provide a generic approach for map generation, which is especially handling unknown terrains. They filtered and partitioned the GPS traces input data. Afterwards, they applied a graph based clustering algorithm and updated the graph incrementally. By following the newly added track they determine at each node if it belongs to an existing edge (using a distance within search for each vertex). If the new trace and the existing one are similar (matching the configured thresholds), the new track is matched. If the current vertex of the new trajectory does not fulfill the matching criteria, the new track is split and connected to the graph on the latest matching vertex. Additionally, the graph edges are added with travel time information. To allow parallel processing, the input data is divided into tiles. After processing the graph is stored within a database and extractions of the database is the basis for a routing software. Their experiment consists of 107 traces which represents a set of 3,075 km recorded roads within a time of 40 hrs.

Roth [2008] developed a two step algorithm: In the first step track parts that represents the same paths are identified, the second step fuses identifiable parts to a single path considering the Gaussian distribution of the GPS measurements. He used a data set of 200,000 measurements in South-East Germany (region of Nuremberg) that represents approximately 6,000 km driving distance.

Cao and Krumm [2009] used a particle physics approach that describes attraction forces of each track point to its neighbours and spring forces to its origin position. Afterwards, they created a graph structure. For this, they selected points for linear pairing and build a topology which is feasible for routing purposes. As their figures visually demonstrate, the algorithms work well with the example data set of 20 million GPS points from campus shuttles and the results are quite promising.

A completely different approach is followed by Zhang et al. [2010]. With a skeleton operator they computed a road network using land parcel geometries. Their algorithm is divided into three parts: shape composition, skeleton approximation and topology decomposition. This is, compared to the other ones, an indirect approach where the road network is derived assuming its characteristic geometric structure. Compared to the analysis of GPS traces, this approach lacks missing real movement data. It also needs a regular updated cadastral data set which can be expensive and is usually not available at every place. Their test area is the city of Barcelona which represents a city with a typical old European town core. The newer parts are organized in rectangular block systems. Their straight skeleton algorithm method fits well for the rectangular road network of the newer parts. The errors are mainly located at the fine granular curvy structured part of the old town. As they have not given any information about their used ground truth data set or the source data, this approach is not comparable with the others. However, it outlines a different method to get road networks from geographical data sets.

Chazal et al. [2011] presented a trajectory smoothing approach using average window filtering of GPS tracks. Their focus lies on smoothing the GPS trajectories and their results

are comparable with the clarification step of Cao and Krumm [2009] or the principal curve of Hastie and Stuetzle [1989]. However, the generation of a topological road network is not part of their publication. Since the noisiness of GPS data is a problem of most of the graph generation algorithms, they give one more possible method to get a more clear data set. Compared to the approach of Cao and Krumm [2009], their more simple approach works for left-hand and right-hand drive systems but therefore they are not able to detect opposite lanes of a road. A positive side effect is, that their generated road center line will be more clear in case of simple (not lane related) map data generation.

Ahmed and Wenk [2012] present an approach based on partial matching of trajectories to the graph. They used minimum link path for complexity reduction of the reconstructed graph. As one of few they also give quality guarantees and show experiments based on synthetic and real data. The matching task in this work is solved by a new variant of partial Fréchet distance.

With the CrowdAtlas project, Wang et al. [2013] built up a whole framework which updates existing map data. They identified the lack of manual created digital road maps and combined latest established algorithms to the CrowdAtlas framework. The client part records GPS traces of users. These traces are collected by a server software which starts the map updating process. The collected GPS traces are matched by using the improved offline Viterbi map matching of Wei et al. [2012]. After a sufficient number of matched traces the map inference algorithm automatically updates the map. This is the recent and most complete framework that is integrating map generation in a applicable software. This work is basically comparable with the workflow that is implemented within this thesis.

A more recent approach in the domain of pedestrian routing is the publication of Kasemsuppakorn and Karimi [2013]. They took self tracked pedestrian GPS trajectories and GPS data from OpenStreetMap. With a three step processing (preprocessing, significant point filtering, pedestrian network construction), a navigable network for pedestrians was constructed.

Their work is distinct different from other approaches as they detect road intersections by spatial analysis of conflict points and determines the circle boundary as well as the traffic rules within the intersection region. They generate a simplified routable road network based on road intersections and one- or two-way streets without the need to refer to existing road networks. They used methods from Cao and Krumm [2009] and integrated them into a workflow. The aims of this work are partly the same like Wang et al. [2013] but this work does not show a complete framework hence it is more focused on the algorithm development to distinguish different parts of a road (intersection vs. road segment).

Table 4.1: Algorithms categories (source: Ahmed et al. [2014])

| Algorithm | Point Clustering | Incremental Track Insertion | Intersection Linking |
|-------------------------------|------------------|-----------------------------|----------------------|
| Ahmed and Wenk [2012] | | ✓ | |
| Biagioni and Eriksson [2012] | ✓ | | |
| Cao and Krumm [2009] | | ✓ | |
| Davies et al. [2006] | ✓ | | |
| Edelkamp and Schrödl [2003] | ✓ | | |
| Ge et al. [2011] | ✓ | | |
| Karagiorgou and Pfoser [2012] | | | ✓ |

Some of these approaches are classified and evaluated in the works of Ahmed et al.

[2014] and Biagioni and Eriksson [2012]. Biagioni and Eriksson [2012] implemented a subset of algorithms, described their characteristics and identified remaining challenges. They classified the algorithms by a set of criteria. Ahmed et al. [2014] took further approaches in their review and identified three algorithm categories: Point clustering, incremental track insertion and intersection linking cf. table 4.1. In this work, the table of Biagioni and Eriksson [2012] is extended with further and recently published approaches to give an updated state-of-the-art for road generation from geographic data (especially from track logs).

None of the existing approaches use data from agricultural vehicles. Most of the approaches are using data from cars or transport vehicles on public road network. Some of them use pedestrian data or bicycle trajectories. Therefore, the road network data is limited to the roads that are driven by the investigated moving objects. Hence, road data in rural areas is very sparse, especially on non public roads that are used mainly by agricultural vehicles and updating processes are very expensive. These specific data have to be investigated and methods have to be improved to close this gap.

4.2 Road generation Algorithms in Detail

Many of the algorithms in table 4.2 base on density computations that are realized through rasterizing the data. For higher resolutions, this technique allows only small coverages. Due to the fact, that most of the example data used in these publications have a small bounding box, the application of the algorithms is possible. Enlarging the regions raises the memory consumption tremendously and the algorithms therefore do not scale. The agricultural telemetry data is mainly collected in rural areas. The bounding box of the regions is quite huge and the data density is more heterogeneous. This fact will not allow or extremely limit the application of raster based algorithms on the data. Parallelization approaches could be a solution to solve the requirement of large hardware resources. A spatial parallelization needs the handling of spatial boundaries to guarantee topological correctness. Vector oriented approaches are more flexible and scale with the area size.

In this chapter, the chosen algorithms which are used for road geometry calculation are described in detail. The Algorithms have four different views on trajectories: Cluster based (k-means), trace merge based, graph based and distance based (Fréchet distance).

4.2.1 Edelkamp and Schroedl - cluster based

Edelkamp and Schroedl presented an early approach for map refinement and initial map generation based on clustering trajectories. They made use of the Bentley-Ottmann algorithm [Bentley and Ottmann, 1979], a sweep-line algorithm, to calculate the intersection points of the GPS trajectories. The result of the algorithm is a planar graph. The original algorithm, which is used in computational geometry for calculating undirected graphs has been extended to produce a directed graph structure. The edge direction has been determined by using the timestamps of the GPS trajectory points. For the following map generation process, they proposed two alternatives:

1. The refinement of an existing map, which limits the approach to a refinement of existing road structure and depends on the accuracy of the used base map.
2. The initial creation of a map, which allows to detect new roads. This approach limits the road creation to the geometry and the direction while the usage of an existing

Table 4.2: Algorithms categories (source: adapted from Biagioni and Eriksson [2012])

| Paper | Class | Data | Ground Truth | Evaluation Method | Features |
|----------------------------------|-----------------------------|---|--|---|---|
| Edelkamp and Schrödl [2003] | k-means | 250 synthetically perturbed traces | Generated from GPS traces | Lane error vs. amount of data | Lane finding |
| Schroedl et al. [2004] | k-means | 250 synthetically perturbed traces | Generated from GPS traces | Lane error vs. amount of data | Intersection geometry |
| Morris et al. [2004] | Graph reduction | GPS hiking traces | Aerial Images | Eyeball vs. ground truth | Implemented in commercial Software (TopoFusion) |
| Brüntrup et al. [2005] | Graph Clustering | 107 traces (3075 km, 40 h) in Germany | Conventional map | Eyeball vs. ground truth | Parallel processing |
| Davies et al. [2006] | KDE | 1 million GPS points | UK ordnance survey | Eyeball vs. ground truth | na |
| Worrall and Nebot [2007] | k-means | Traces from mining vehicles | None | Compact vs. raw | Compact representation |
| Guo et al. [2007] | k-means | Synthetic GPS traces | None | Relative error vs. amount of data | na |
| Chen and Cheng [2008] | KDE | Traces from automobiles | Google Earth | Eyeball vs. ground truth | na |
| Roth [2008] | Probabilistic Track Fusion | 200,000 GPS points, 6,000 km, Nuremberg | None | Eyeball | Integrating GPS precision |
| Niehöfer et al. [2009] | Trace merge | 7 traces | Google Maps | Eyeball vs. ground truth, relative error vs. amount of data | Edge classification |
| Cao and Krumm [2009] | Trace merge | 20 million GPS points from campus shuttles | Bing Maps | Eyeball vs. ground truth, route query vs. Bing Maps | GPS trace clarification |
| Shi et al. [2009] | KDE | Massive amounts of GPS traces | Google Earth | Eyeball vs. ground truth | na |
| Jang et al. [2010] | k-means | GPS traces | Naver maps | Eyeball vs. ground truth | na |
| Zhang et al. [2010] | Skeleton | Land parcel data of Barcelona | Not specified | Spatial Decomposition Algorithm | na |
| Agamennoni et al. [2011] | k-means | 5 days or 15 open mine vehicle GPS traces | None | Eyeball vs. Davies et al. and Schroedl et al. | Principal road path |
| Chazal et al. [2011] | Moving average | Synthetic data, Moscow data from OpenStreetMap, 7145 taxicab traces from a major city | CloudMade maps | Fréchet distance and Eyeball vs. ground truth | na |
| Ahmed and Wenk [2012] | Fréchet Distance | Taxi cab data, Berlin | None | Eyeball and integrated thresholds | Integrated quality analysis |
| Wang et al. [2013] | Principal curve | Beijing taxi data, 4,351,977 samples, 10 s interval | Google Earth, OpenStreetMap, Beidu Map | Eyeball vs. ground truth, checking self driven ways on map | Implemented framework and infrastructure |
| Kasemsuppakorn and Karimi [2013] | Significant point filtering | 60 self collected walking traces, OpenStreetMap GPS-tracks | Self collected ground truth by digitize from high resolution imagery and D-GPS surveying | RMS of geometries, Eyeball | Focused on pedestrian trajectories |
| Wang et al. [2014] | Trace merge | 10,000 GPS traces, Huaibei | Google Maps | Eyeball vs. Google Maps, routing | Road intersections |

map can enrich the derived road segments with further attributes, such as road type and speed limit.

The implemented approach used for telemetry data is based on the second alternative, in terms of comparability with the other approaches and the lack of road data in rural areas. With a preceding filtering step, outliers (e.g. unrealistic positional jumps through GPS problems) are deleted and traces are resampled to equalize ingoing data (high and low sampled trajectories). The next step is the *road segment clustering* which consists of three parts.

1. The *cluster seed location* identifies sample points of different traces that belong to the same road. Whenever a new trajectory should be added to the set of trajectories, two values for the already clustered trajectories (the averages) and the incoming new trajectory segments will be calculated: The minimum distance to the next trajectory and the heading. If the new trajectory is within the given thresholds, the trajectory will be added to the set and a new mean value for heading position of the cluster center will be calculated. This iterative process is similar to the *k*-means algorithm of Macqueen [1967]. Whenever a new trajectory is added, the cluster center will change its position and it is checked if the points in the set are still inside the given thresholds. If they are outside the threshold, they will be deleted from the set. This process continues until no more points can be added. The initial cluster seeds are generated repeatedly until each trace point has at least one seed within a given distance threshold. To not miss any intersection, this threshold is suggested to be e.g. 50 m. With a simple greedy strategy each trace will be followed and new cluster seeds will be placed at regular intervals if needed (cf. figure 4.1).

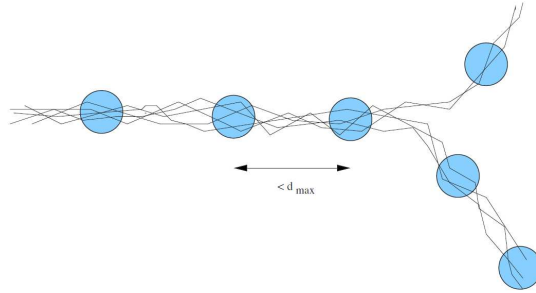


Figure 4.1: Example of traces with cluster seeds (source: Edelkamp and Schrödl [2003])

2. The following *segment merging* step merges the cluster centers that belong to the same street. The cluster centers will belong to the same road if cluster centers C_1 and C_2 fulfill following assumptions: 1) C_1 precedes C_2 (which implicates that all traces that belong to C_1 will pass C_2) and 2) all traces belonging to C_2 originate from C_1 . If and only if these two criteria are complied, the clusters are merged to a *segment*. The beginning C_1 and and cluster C_n are named as the boundary clusters of the segment (see figure 4.2).
3. In the last step, the *segment intersection identification*, road intersections are identified by a *snake* method (borrowed by image processing). The snake model is a contour model which fits a set of (noisy) sample points. Edelkamp and Schroedl

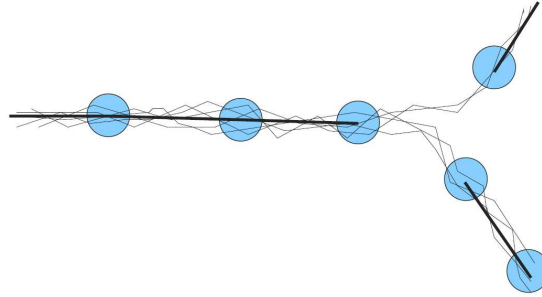


Figure 4.2: Merged cluster segments (thick black lines) (source: Edelkamp and Schrödl [2003])

uses a simple star shape approach to fit the intersection. An example is given in figure 4.3 where the dotted lines represent the intersection.

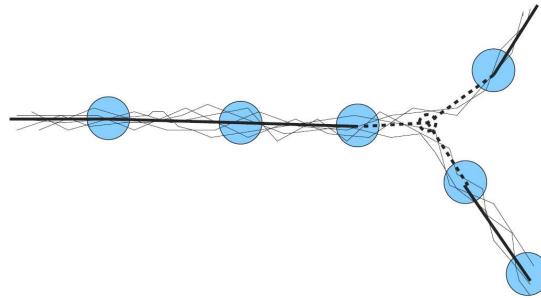


Figure 4.3: Traces, segments, and intersection contour model (dotted) (source: Edelkamp and Schrödl [2003])

The previous steps identified similar trajectories representing a road and show the calculation of intersections. In a last step, Edelkamp and Schroedel refine the individual segments using a spline based approach to generate a proper road centerline. For more details on their used spline fitting, it will be referred to their publication where they describe the derivation of spline parameters and give error assumptions for their different models. Since the agricultural telemetry data is mainly collected in rural areas where roads consists of usually equal less than two lanes (one for each direction), the lane derivation will be excluded in this work.

4.2.2 Cao and Krumm - trace merging

The approach of Cao and Krumm [2009] mainly consists of two parts. The first part is the clarification step for the usually noisy GPS points. The second part is the building of the graph.

Clarification

The first step is the clarification step. Since GPS measurements have a positional error and the resulting trajectories therefore represent the “real”, driven trajectory with a variance, it needs a processing step that clarifies the measurements. Cao and Krumm [2009] use

the model of particles where they calculate two kind of forces between the measurement points. One is the attraction force: Measurements of neighbor trajectories attract the current handled measurement point. The opposite is the spring force: Each measurement has the endeavor to rest on its current position. The spring force pulls the measurement back in direction of this position. Figure 4.4 shows the two forces and how they are calculated. The new position is then calculated as a translation where these forces are balanced. This is done for all measurement points of all trajectories. Since the new positions generates new attraction forces (due to the changed neighborhood), this process is done for multiple iterations.

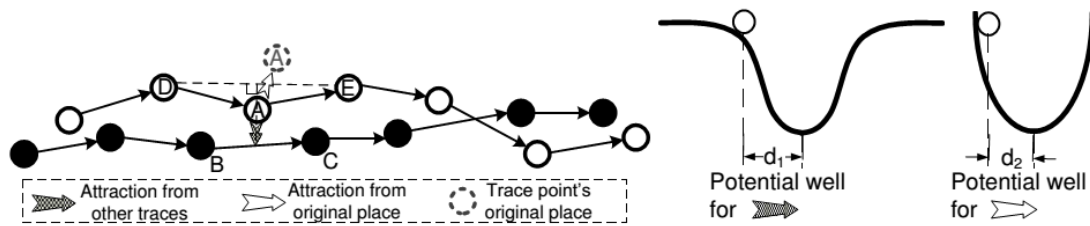


Figure 4.4: Forces on measurement points (source: Cao and Krumm [2009])

Graph Generation

The previously clarified point set is the base for the graph generation. Their graph generation algorithm works incrementally (track by track). For each new track in the list, all belonging nodes are ordered by time stamp. In the next step, a decision will be made for each node whether it has to be added to the existing graph or not. The node will be merged to the graph if it exceeded a difference threshold (Figure 4.5).

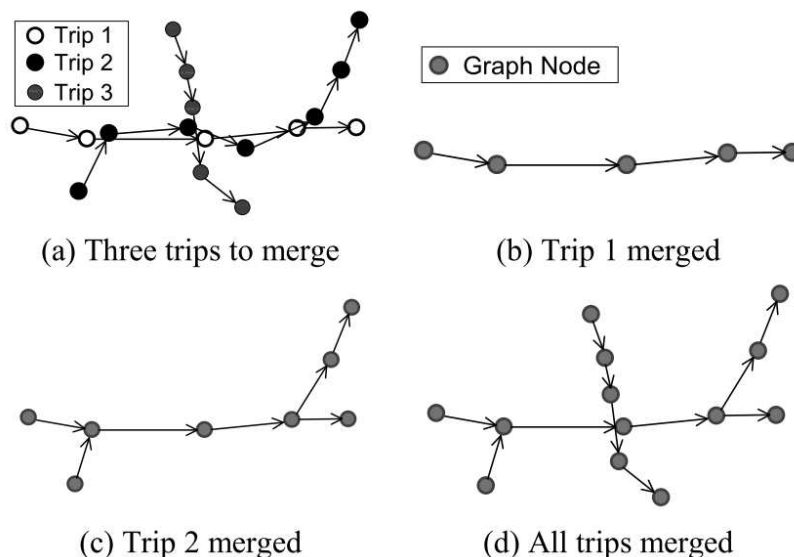


Figure 4.5: Graph generation algorithm (source: Cao and Krumm [2009]) - (a) The input, in terms of 3 trips. (b) The graph G after processing trip 1. (c) The graph G after processing trip 2. (d) The graph G after processing trip 3

This algorithm depends on the sorting of the ingoing track list. Changes within the resulting graph will be marginally, if the preceding clarification step returns a clear set of close line strings and the shape points of each line string are within a small distance to the neighbor trajectory. The first requirement can be fulfilled with a good clarification processing. The second condition depends on the input data. In case of the used telemetry data, the shape points are distributed randomly. Therefore, the first chosen trajectory has a large influence on curve representation. If the crest is not represented by a shape point, the curve will be generalized and the curve radius will be flattened. To reduce this, a possible pre-processing step could be interpolation of the sparse sampled ingoing trajectory shape points to increase the spatial density. This will increase the geometric road fitting with the payment of longer computation time. In the presented analysis, the original algorithm without this possible extension is shown.

4.2.3 Morris et al. - Graph Reduction

Morris et al. [2004] used a graph based geometric approach. The theory builds on the fact that due to GPS inaccuracy and different driving maneuvers the recorded trajectories of a street will intersect each other. The method makes use of this fact to generate a graph structure. To generate this topology, this first intersection step is realized by a sweep line approach which identifies the intersection nodes for all trajectories. In the following step, the meshes have to be identified and a planar graph is generated from the intersected trajectories. Figure 4.6 shows the processing steps from GPS trajectories via graph generation through intersecting trajectories to the final reduced graph.

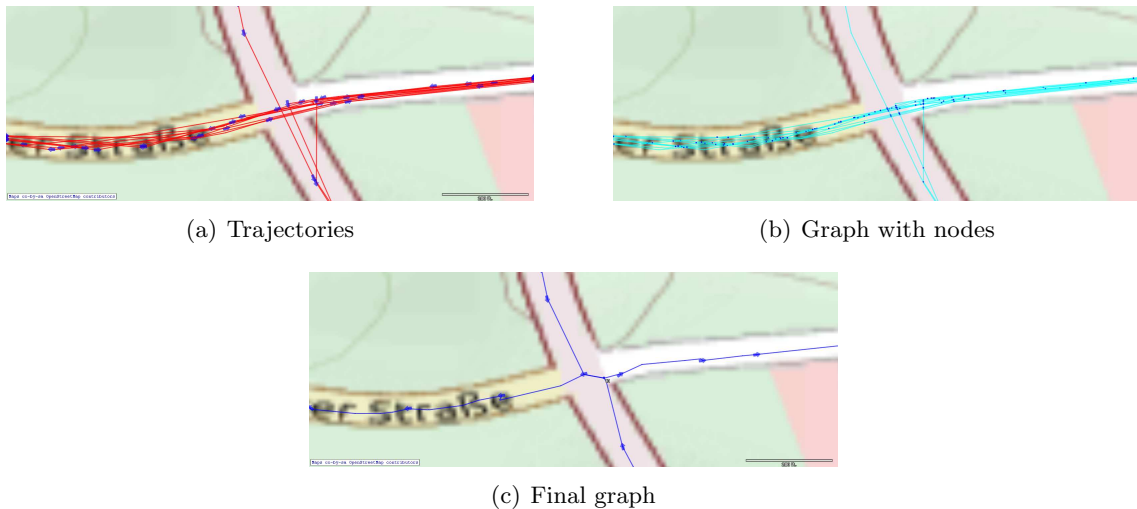


Figure 4.6: Graph reduction - from trajectories to final graph (Datasource basemap: ©OpenStreetMap contributors)

After this first computation, a set of graph reduction algorithms is used to shrink the faces which should finally result in a single edge for one street. Morris et al. provide three kinds of reduction algorithms that are explained in detail in the following sub sections.

Parallel reduction

Parallel reduction requires two parallel edges that are connected with the same nodes at their ends (figure 4.7). The parallelism is measured by calculating the Hausdorff distance [Alt and Guibas, 1999] of the two edges. Only if the calculated Hausdorff distance is lower than a given threshold (in their publication the threshold is chosen between 20 and 60 m, depending on the data quality), the two edges will be reduced to one. It is evident that a smaller threshold (less than GPS accuracy) will not reduce segments that are obviously representing the same street. A threshold which is larger than the minimum distance between two parallel streets will unintentionally result in merging two roads into one. The threshold is therefore depending on the road structure and on the positional accuracy of the measured trajectory points. If two parallel edges are identified, the reduction process starts.

Given:

Edge A with start node a_{start} and end node a_{end}

Edge B with start node b_{start} and end node b_{end}

Condition: A and B are parallel if the two conditions are fulfilled

$$1) (a_{start} = b_{start}) \wedge (a_{end} = b_{end}) \vee (a_{start} = b_{end}) \wedge (a_{end} = b_{start})$$

$$2) H(A, B) < rThresh \text{ with } H(A, B) = \max(h(A, B), h(B, A))$$

$$h(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\}$$

In: LineString A, LineString B $d(a_i, b_i) :=$ orthographic projection of a_i on (b_i, b_{i+1})

Reduction step:

The polyline for the single edge used to replace the parallel edges is determined as follows: Let the two parallel polylines being reduced be polylines A and B. Assume polyline A contains more points than polyline B (if this is not the case, reverse them) and let m be the number of points in A. The closest point in polyline B to each point in A is found. This produces m pairs of points, where points in B can appear more than once, while points in A appear only once. The geometric average of each of these pairs is computed and assembled into the resulting polyline. This polyline represents the average of A and B and has as much information as possible (since there are more points in A). Since A and B always share exactly two points in common, the average polyline will also share these same two points. Figure 4.7 gives an example of a parallel reduction. The resulting averaged polyline will then connect the two common nodes.

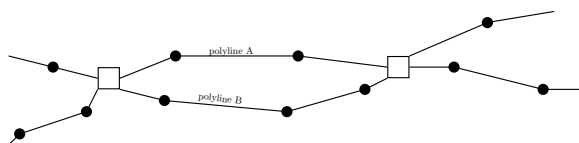


Figure 4.7: Parallel reduction (changed after source: Morris et al. [2004])

Face reduction

The face reduction is a more general case of parallel reduction. In a first step, the two most far-out points with a degree of four (two ingoing and two outgoing edges) of the face will divide it in two parts. These parallel parts get reduced using the parallel reduction algorithm. Previously connected edges will be connected to the resulting edges by linking the shifted vertices to the averaged line with the outgoing edges. Figure 4.8 shows the identified crossings, shape points (black) and nodes *a* (left) and *b* (right) which are the far most points of the face.

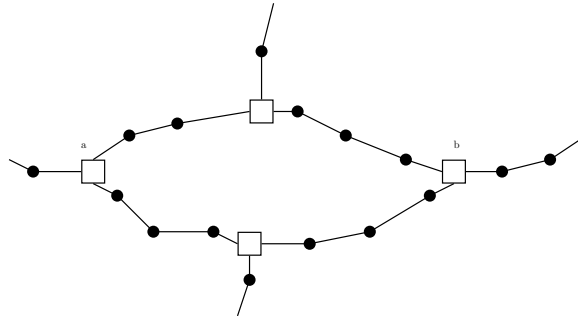


Figure 4.8: Face reduction (changed after source: Morris et al. [2004])

Serial reduction

Serial reduction deletes vertices in between nodes that are connected only with two edges. The method concatenates the edges and integrates the connecting vertex in the geometry as a shape point (figure 4.9). The algorithm re-establishes the condition of vertices only at trail intersection. Therefore, this algorithm will prune the graph and the result would be better structured road data.

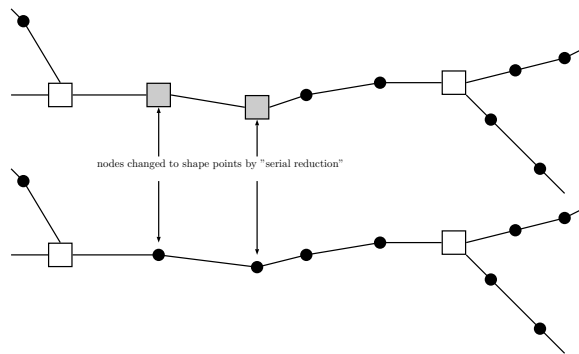


Figure 4.9: Serial reduction - converting nodes to shape points

The workflow

Morris et al. [2004] used above three types of graph reduction algorithms to reduce the starting graph to a feasible road network data set. They pointed out, that the order, in that the three types of algorithms are used has no effect on the workflow performance.

Further the order can have a theoretical influence on the result. Hence the synthetic examples where this would be the case, do not exist in real data set, the order will not have an appreciable effect on the result.

4.2.4 Ahmed and Wenk - Network generation using Fréchet distance

The incremental approach of Ahmed and Wenk [2012] uses partial matching of new trajectories to an existing graph. Starting without a graph takes the first trajectories as initial state and grows a street network from these. This partial matching is done by a new variant of partial Fréchet distance [Buchin et al., 2009].

Ahmed and Wenk modelled the road network as an embedded, undirected graph. With an initial base they define a precision parameter $\varepsilon > 0$ which gives a error bound for the positional error of each trajectory. The trajectories are modelled as piecewise linear curves (which is a common way for modelling trajectories). The similarity of two curves is measured by the Fréchet distance [Alt and Godau, 1995]. For two planar curves $f, g : [0, 1] \rightarrow \mathbb{R}^2$, the Fréchet distance δ_f is defined as:

$$\delta_F(f, g) = \inf_{\alpha, \beta: [0,1] \rightarrow [0,1]} \max_{t \in [0,1]} \|f(\alpha(t)) - g(\beta(t))\| \quad (4.1)$$

where α, β range over continuous and non-decreasing reparametrizations, and $\|\cdot\|$ denotes the Euclidean norm. The well-separability of streets is defined by Chen et al. [2010]:

Definition:

A point p on G is α -good if $B(p, \alpha\varepsilon) \cap G$ is a $1 - ball$ that intersects the boundary of $B(p, \alpha\varepsilon)$ in two points. A point p is α -bad if it is not α -good. A curve β is α -good if all points on β are α -good.

To identify clusters of similar sub-curves, the concept of free space and free space surface is used. The matched parts of the newly added trajectory will get rejected. The parts of the added trajectory that do not match the existing graph will be added. This step could effort the placement of new shape points and the splitting of existing edges. It is also the only part of the algorithm that changes the existing graph geometry which is derived from the trajectory geometries. Figure 4.10 shows the trajectory merging process for one increment.

The setting of ε , which is mainly depending on GPS data and road network structure, is the only variable in this algorithm. Therefore a good ε is one that is small enough to keep on the existing roads and large enough to avoid connected road fragments.

Compared to other algorithms, this approach is not moving the existing shape points. The algorithm also is not calculating average positions from the trajectories. From the first trajectory as origin, the algorithm has to make a decision for each of the following trajectories that is added to the graph what shape points will extend the graph and which already exist in the graph. This is one of the main differences comparing the output of the applied algorithms. Another limiting factor is the resulting graph which consist only edges with two geometric points. The cleaning algorithm, mentioned in Ahmed et al. [2014], is not implemented because it has no influence on the resulting geometry and will only have effect on the graph complexity. This is also the reason why Ahmed et al. [2014] excluded this last step from their processing workflow.

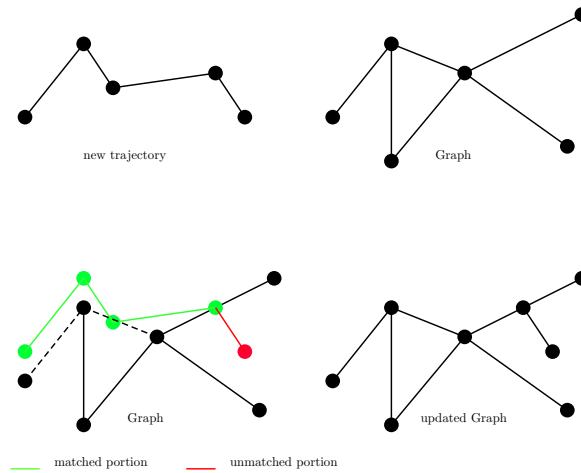


Figure 4.10: Graph merging - integration of a new trajectory (sketches modified from presentation of Ahmed and Wenk [2012])

4.3 Application on Agricultural Telemetry Data

To run the presented algorithms on agricultural telemetry data, several data processing steps have to be performed. Based on the general preprocessing steps presented in chapter 3, the data has to be brought into the right format to feed the presented algorithms. The algorithm of Morris et al. [2004] consumes the trajectory data which has to be exported in GPX format. The other algorithms consumes trips from ASCII-files where each trip has to be stored in a separate file. The Ahmed and Wenk [2012] algorithm additionally needs a metric, projected coordinate system to calculate the correct distances and intersections. Therefore the existing geographic lat/lon coordinates have to be transformed into a projected Cartesian coordinate system. The parametrization is set by the default parameters, if provided, or by visual checks on the results. The following subchapters will give more details on parametrization of each applied algorithm. Besides the requirements set by the telemetry data and its domain and region specific characteristic will be discussed.

4.3.1 Edelkamp and Schroedl

The algorithm of Edelkamp and Schroedl [Schroedl et al., 2004] needs a set of parameters which are given by default in their publication.

1. The cluster seed interval (for the initial setting of cluster seeds)
2. The bearing difference limit
3. The intra-cluster distance limit

For the agricultural telemetry data the parameters are chosen from the given empirical determined intervals from Edelkamp and Schroedl as follows:

1. The cluster seed interval is set to 50 m
2. The bearing difference limit is set to 45°
3. The intra cluster distance limit is set to 20 m

4.3.2 Morris et al.

The graph reduction algorithms are parametrized in two ways:

1. *rThresh* - The threshold value for application of reduction for two parallel edges. This threshold is the limit for to the Hausdorff distance $H(A, B)$ between edges A and B .
2. *cThresh* - Defines the length of a spur to be a salient trail.

The sequence of the algorithm application (parallel reduction, face reduction and serial reduction) has no greater impact on the result set as described in Morris et al. [2004].

For the agricultural telemetry data, these parameters are set for two runs:

1. $rThresh = 110$, $cThresh = 60$
2. $rThresh = 150$, $cThresh = 100$

These values are chosen higher than typical values given in Morris et al. [2004]. This is mainly due to the larger segments of the trajectories (shape points have a temporal distance of 15 s). The large values for the *rThresh* will guarantee a melting of nearby trajectories and the *cThresh*-value mainly eliminates the not desired appendices.

4.3.3 Ahmed and Wenk

As one of the most recent presented approaches, the algorithm of Ahmend and Wenk [Ahmed and Wenk, 2012] needs further investigation on parametrization. Although the parametrization algorithm is only set by one parameter, the ε threshold which defines the threshold for matching. This parameter depends mainly on two attributes (as already described in 4.2.4): 1) the quality of the GPS points and 2) the structure of the road network.

To define a proper ε -value, several test runs with different values for ε were performed. Results are shown in figure 4.11. Especially in regions where roads are crossing (e.g. Y-junctions, T-junctions), the influence of a good ε is obvious. While small ε values results in skeletonized structures due to multiple connections to neighbor road segments, higher ε values generalize the road network. The latter fact results in pruned junctions where the central junction node is shifted and in a melting of parallel roads. Melting is typical for agricultural roads and parallel higher classified public roads that are usually have only a small distance to each other. In regions where the predominant field structure is small (where field edges are smaller than the ε threshold) the algorithm will also merge the enclosing roads.

As a proper value for the given telemetry data, an ε -value of 70, has been visually identified using the following restrictions:

- the higher ε , the wider the crossing areas
- the smaller ε , the worse the network connectivity
- the smaller ε , the more fragmented and skeletonized the resulting road network

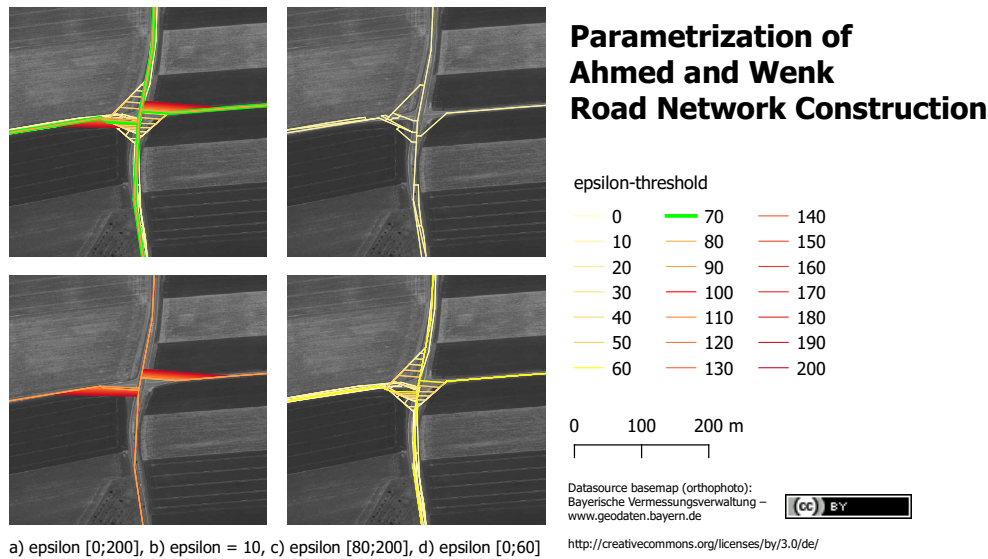


Figure 4.11: Parametrization of Ahmed and Wenk - Road Construction (a) and b) are the upper figures, c) and d) the lower ones)

4.3.4 Cao and Krumm

The method of Cao and Krumm, as formerly described, consists of two parts. In a first step, positions get aligned by a clarification step. This step is parametrized by the attraction force σ and the spring force constant that gives a value for the force which holds the point on its old position. The values are set by default as:

- spring force constant $D = 0.005$
- $\sigma_{attraction\ force} = 5.0$

The algorithm of Cao and Krumm has no termination criteria. A logical termination criteria is the iteration when spring force and attraction force will be in balance and the points will not be translated for a longer distance as the set threshold.

It has been observed that the clarification step has limitations in curve structures. Especially for narrow curves (e.g. hairpin curves), the points tend to merge the curve. For longer curves, the clarification step tends to straighten the curve. In typical US American cities with rectangular road networks (e.g. presented samples from Microsoft cars), these problems will not appear.

The second step performs the graph generation. Although the GPS points are placed in better positions through the clarification step, the resulting road depends strongly on the basic trajectory. This is the main disadvantage of the graph generation. Cao and Krumm made use of the trajectories and how they were driven. On junctions, the outgoing roads will be connected on the base trajectory. The main issue is the long distance between the trajectory shape points which cut curves and shorten linkages (cf. figure 4.12). A previous densification of shape points could improve the curve fitting. To not increase the computation effort, densification should be made after clarification.

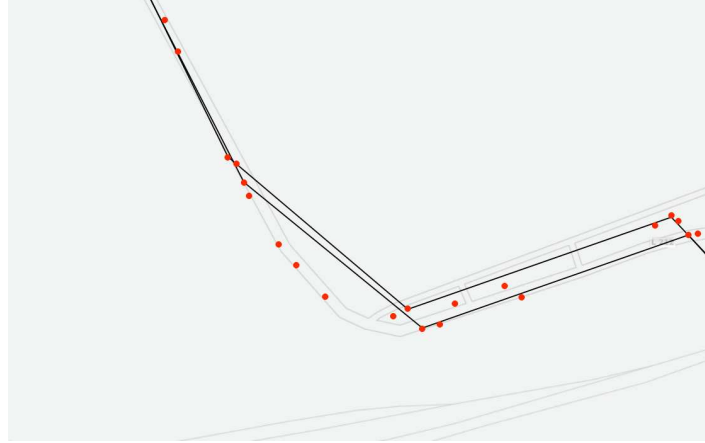


Figure 4.12: Problems with curve structures and graph generation through low point density

4.3.5 Comparing methods

Comparing road generation algorithms, different approaches exist. Biagioni and Eriksson [2012] and Ahmed et al. [2014] compared some of the presented algorithms. Biagioni and Eriksson [2012] introduced a quantitative, statistical methods for map comparison. In a first step they put holes in fixed intervals from a starting point on the graph of the reference map. In the second step, they put marbles also in fixed intervals on the extracted road network. In the final step, the algorithm then computes the matching between holes and marbles and an accuracy assessment is the result.

$$spurious = \frac{spurious_marbles}{spurious_marbles + matched_marbles}$$

$$missing = \frac{empty_holes}{empty_holes + matched_holes}$$

With these values they calculated the *F-score*:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 2 \cdot \frac{(1 - spurious)(1 - missing)}{(1 - spurious) + (1 - missing)}$$

This method works well for their used data set where most of the road are existing in both sets (generated data and reference map). For the agricultural telemetry data, the coverage is more heterogeneous. Therefore a visual comparison a useful approach to get an overview on the quality of the produced graphs.

Visual comparison

Figure 4.13 shows a rectangular road network within a rural area. The raw trajectories in the upper left picture show a clear road network. The upper horizontal road has a small curve where set of linestrings are not completely on the road. The crossings, especially the T-crossing in the upper right, also show the influence of the larger recording interval and due to this the shortened graph edges. The horizontal road in the lower part of the trajectory plot is cut before it reaches the T-crossing. This topological problem can be the cause of classification result or a dead end road which is blocked.

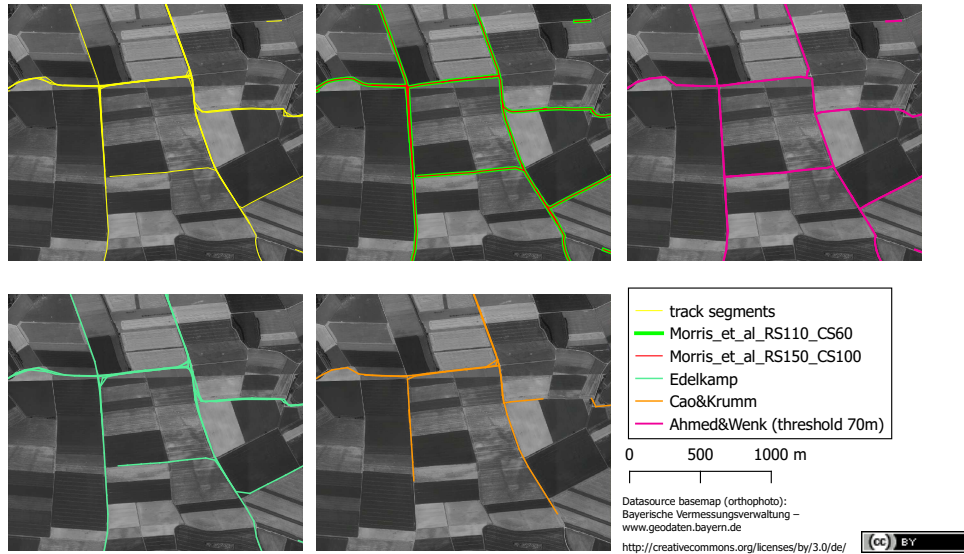


Figure 4.13: Comparison of Map Generation Algorithms in Baden-Wuerttemberg (example: rectangular road network): trajectories, Morris et al. [2004], Ahmed et al. [2014], Edelkamp and Schrödl [2003] and Cao and Krumm [2009]

The algorithm of Morris et al. (green and red trajectories in the upper middle figure) gives a good representation of the road network. Both configurations give similar results. The algorithm clearly computes the crossings and, compared to the aerial imagery, the graph fits very well to the underlying orthophoto. The cut trajectory segment in the lower part of the plot has not been snapped to north-south road which represents the situation in the first image. The curve structure in the upper left corner is well computed and the outliers have been averaged.

The method of Ahmed and Wenk also returns a clear road network. The curve structure in the upper left is well formed and the connections of the edges are proper in general. This algorithm snapped the lower horizontal road to the north-south road which is not represented in the first picture with the trajectory segments. The extraction of crossings has shortcuts. This is mainly the cause for the extraction of t-crossings. Due to the fact that this algorithm is not averaging the trajectories, the shortcuts directly result from the raw trajectories.

The methods of Edelkamp and Schroedl and from Cao and Krumm are not usable to extract a clean road network. The Edelkamp and Schroedl method generates a large family of linestrings for one road. The crossings are also represented through a large set of linestrings and turns are resulting in parallel edges (cf. the right part of the image). This algorithm also does not snap the road in the lower part to the north-south road. Furthermore, the dead end has been shortened through the algorithm.

The Cao and Krumm computation results in a more clear road network. The curve structure in the upper left corner is better represented than in the Edelkamp approach which shows the influence of the clarification step. The algorithm is the only one that produces obvious gaps within the road network. The dead end trajectory in the lower part is completely cut off. The west-east roads in the right part of the figure have a larger gap or the road is missed completely. Some of the T-crossings are well computed while others

have more connections (upper right crossing) or the crossing is placed on the wrong point (upper left crossing) where the roads divide like open scissors.

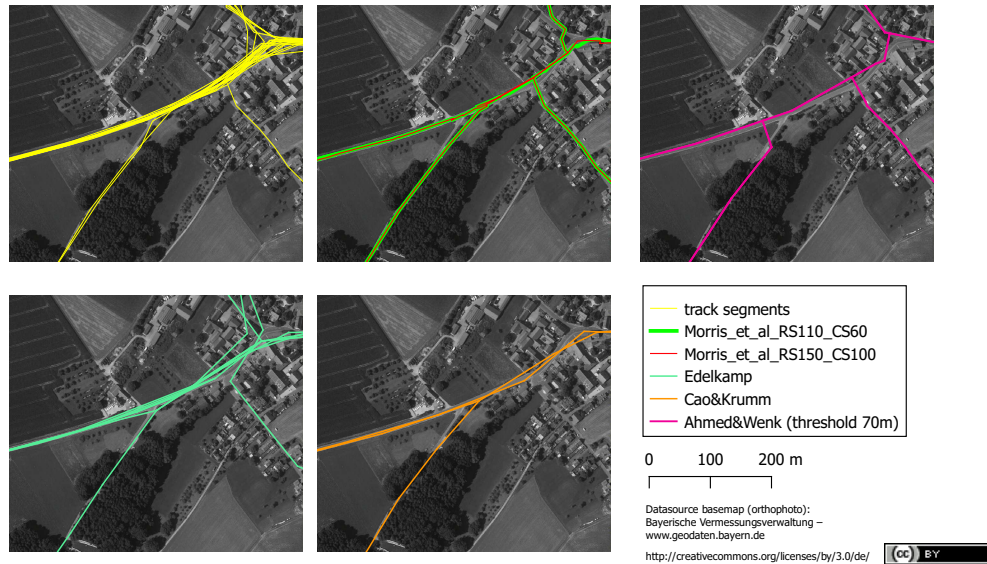


Figure 4.14: Comparison of Map Generation Algorithms in Baden-Wuerttemberg (example: crossing roads): trajectories, Morris et al. [2004], Ahmed et al. [2014], Edelkamp and Schrödl [2003] and Cao and Krumm [2009]

The second set of figures shows an example for a Y-crossing. The data set is located in a village area where road structure is more curvy and the positions are influenced by signal reflections on the surrounded buildings. The trajectory segment plot in the upper left shows a heavy traffic road from west to north east and two connected smaller roads with less trajectories.

The result of the Morris et al. algorithm represents the Y-crossing well since it fits nearly on the orthophoto. The large set of trajectories and the therefore multiple averaging steps result in a more troubled, sawtooth like structure of the middle road segments. The connections of the belonging smaller roads are well integrated.

The algorithm of Ahmed and Wenk calculates a smooth set of edges. The heavy traffic road is fitting well at the beginning but from the Y-crossing to the north-west part, the resulting edges shorten the curve and are placed some meters next to the road in the aerial image. The T-crossing is therefore moved too far to the north. Beside that, the crossing is correct and the edges are connected in the right way. The Y-crossing has been cut through the parametrization of the algorithm. The set threshold connects the smaller road in an early state to the heavy traffic road. The original Y-crossing is therefore cut and the Y-geometry is not well computed.

The Edelkamp and Schroedl approach also shows in this part of the map that the algorithm produces too many resulting edges. While the smaller road coming from the south-west part is computed visually correct, the other roads are represented through too many edges and the topology is not correct represented.

The algorithm of Cao and Krumm shows better results in this case. Although the previously identified issues are also present in this plot (missing roads and shortened curves), the algorithm computes a clean set of edges. The integrated lane computation

leads to two resulting edges for the heavy traffic road which are not well separated and crossing each other. The Y-crossing is well extracted and is after the Morris et al. result the best extraction of this part.

4.3.6 Algorithmic evaluation

The early Edelkamp and Schroedl approach produces the worst results and the extracted road network is not applicable for rendering. However, a proper route calculation can be made due to the fact, that the resulting edges are connected and therefore a topology is given.

The algorithm of Cao and Krumm does not have a break condition in the original implementation. This has been applied by adding a further condition to the clarification step, which terminates the clarification if a measurement point is not changing its position beyond a given threshold. The Cao and Krumm approach is also the most time consuming algorithm as the clarification step runs several times on the data and each measurement has to check its nearest neighbors.

The approach of Ahmed and Wenk is well scalable. The runtime is independent from the spatial extend of the data. As shown in the previous chapter, the algorithm has problems at road crossings and parallel roads. A break condition is not given. Therefore, finding algorithmic a good threshold has to be solved to compute good results for different regions and input data. Ahmed and Wenk provided a first approach for this in their recent publication [Ahmed et al., 2015].

Morris et al. implemented their approach for calculating hiking trails. In this use case, the detection of lanes and driving directions is not needed in general. If this information is needed, further steps should be integrated (e.g. separate calculation for trajectories with different direction). The generation of dead end roads is sometimes not desired, but the result of longer, not intersecting end parts of trajectories. With the original algorithm, these unreal dead end roads can only be removed by setting the threshold for dead end roads, which can also causes a deletion of probably needed dead ends.

4.4 Summary and future directions

The previous chapters gave an overview on selected approaches for road computation from movement trajectories. A subset of methods has been run with the agricultural telemetry data and adjustments for their parameters have been made. The results have then been visually analyzed and the method and data specific problems have been explained.

The performance of the used methods was fast enough to handle at least meaningful subsets of the telemetry data. The implementation of Ahmend and Wenk was the most scalable approach. This was also reasoned by the implementation language and the used software design. While the Ahmend and Wenk algorithm has been implemented in Java, the other algorithms are coded in Python (with an in-memory processing) or within a proprietary software package which sometimes causes memory issues. An algorithmic optimization would be helpful and would strongly improve the usability of the algorithms. Due to the fact that these free available implementations are only used as prototypes for research, this has to be solved when transferring the algorithms in a productive environment. Regarding the standard test data that is used for most of the algorithms in their origin publications, the algorithms have only been tested with fleet data from cars or public transport in an urban environment. Additionally, the typical American rectangular

urban road networks have a strong influence on the results. With the input of agricultural movement data within a rural German region, the results of the algorithms show potential for improvements and limits of the individual methods.

Improvements for future directions can be made manifold. Further postprocessing steps (e.g. smoothening and generalization) can be fast methods to increase the quality of the resulting road geometries (especially for the Morris et al. method, which sometimes tends to generate rough geometries for dense data). Due to the fact that agricultural vehicles are equipped with RTK (for precision farming and steering systems), the positional accuracy for the trajectories will increase in the foreseeable future. The scalability of the methods with the increasing amount of data will also be very important for the near future. Approaches for storing the immense amount of trajectory data in new data structures and distributed computing in combination with divide and conquer strategies will allow the analysis of large trajectory data sets. Through further input data, e.g. attributes of the local road network a better parametrization can be made. Approaches for measuring the straight or curve structure of a road network (cf. “How straight or bendy are the roads?”¹) can help to parametrize the algorithms. Pervasiveness of telemetry systems and usage of smartphones for documentation and billing will increase the available movement data. In combination with further initiatives of standardization (e.g. ISOBUS) and growing companies that combine different data sources for analysis (e.g. 365FarmNet²), the available data and its accessibility will vastly increase. The detection of road attributes from movement data such as curvature, number of lanes, crossings, parallelism, road types and road surface in addition with the knowledge of the vehicle attributes (e.g. measures and speeds) can also be useful information for parametrization of the road network extraction.

¹<http://www.technomancy.org/openstreetmap/bendy-roads/>, (Accessed 2016/06/05)

²<https://www.365farmnet.com/>, (Accessed 2016/06/05)

Chapter 5

Field Boundary Computation

5.1 Introduction and Motivation

The main part of farming takes place on fields. From soil working via seeding, fertilization and plant protection to harvesting, all of these processes are located on fields. Documentation of these processes and optimization will take a large part of time in agricultural management. Field specific analytics of harvest mass and application maps need accurate field boundaries. Resource planning (e.g. fertilizer and seeds) and contractor billing need exact knowledge of field area and its geometry to plan routes, predict time and provide transparent bills. One of the exemplary use-cases in this thesis is the farm to field routing. For this, knowledge of field geometries and gateways to reach the field ground from public road network is highly relevant. Examples for applications are the exact planning and guidance of logistic chains, service vehicles and harvesting machinery. The availability of digitized field geometries depends on the use case. Due to the fact that field surveying and geodata are governmental issues, digital parcel data is almost only available at governmental surveying offices. Most of the governments e.g. members of the European Union have made their geodata accessible through Inspire [Eu, 2007]. But it is also obvious that the surveyed parcel data is not equal to farmed field area. Not farmed areas such as wetlands, banks, trees and hedges and further artificial structures such as power poles or wind generators are not excluded from this area. Another governmental data source (limited to the states of the European Union) is data from IACS (Integrated Administration and Control System) [Council of European Union, 2009]. IACS (in German InVeKoS - Integriertes Verwaltungs- und Kontrollsystem) is a system for documentation of agricultural processes to control direct payment support schemes and is also used to fulfill the requirements under the cross-compliance agreement. Farmers are requested to digitize their field geometries and send them (with additional information) to the funding administration (e.g. agriculture or environmental ministry) [Krause, 2006]. These field boundaries are not constrained to represent the working boundaries (e.g. the harvested area) of the field. Hence, the digitalization base is an up-to-date orthophoto and other geodata such as neighbor fields and parcels. This system is only available in the European Union. Summarized, there is no officially available data set that fulfills the requirements of field boundaries for billings and routing objectives. Till now, billings are mainly related to the official parcel size or on area counts of the agricultural machine, which is also not calibrated, and the influence of overlaps (the area counter calculates the processed area by multiply the working width with the driven distance by segments). Therefore, the production of up-to-date field geometries that represents the harvested or processed area is

necessary. Another important requirement for routing use case is the availability of transit points, that allow a routing from road network into fields. At this time, these points do not exist in any available data set.

In this chapter an approach for automatic generation of field boundaries using agricultural machinery movement data is given. The focus lies on field boundaries for routing from farm to field and infield. Hence, a further step towards finding the transition points from road network into field is carried out. It will be shown that generated field boundaries are also a feasible basis for field record systems, subsidies (e.g. InVeKoS), documentation and strategy planning for future harvest campaigns. The derived field geometries are also a base for precision farming tasks.

5.2 Field Boundary Computation

Computing the boundary of an area is part of many application domains. Therefore several approaches and methods exist. These are mainly related to underlying and available data and the requirements that processing results have to fulfill. In the next subsection an overview on domain independent and domain related methodology will be given to specify the state of the art and research gaps for field boundary computation.

5.2.1 State of the Art

Polygonization of point clouds is a common task in 2D and 3D computer graphics. Hence, this is the main field of research where solutions for this problems are addressed. Due to the fact, that field boundary extraction is mainly a geometric algorithm issue (creation of planes from point clouds or trajectories), methods for computation of polygonal geometry structures from point clouds are considered within this section.

An overview on computational methods is given by Berger et al. [2014] where they focused on surface reconstruction from point clouds. They categorized existing algorithms by methods, point cloud artifacts, input requirements, shape class and the reconstruction output. In their work, they deal with point clouds acquired through 3D scanners. These approaches handle one more dimension and the problems have one more dimension in complexity compared to the 2D field boundary generation. However, these methods are reducible on two dimensions and need to be considered for field boundary creation in 2D. More general methods are α -shapes by Edelsbrunner et al. [1983] or convex hull algorithms by Preparata and Hong [1977]. Computing a convex hull is a fast approach to approximate polygonal structures (far better than bounding boxes) but it will give only an outer boundary of the point cloud and ignore concave structures. This is only feasible for rectangular or natural convex field polygons. More complex structures like α -shapes have a higher computation complexity but they are also able to detect holes and concave structures. Limits are mainly in handling of outliers that have a strong impact on the resulting shape, especially at margins. Duckham et al. [2008] developed the χ -shape approach that uses the Delaunay-triangulation of the input point data. They reshape the boundary of the Delaunay triangulation and with a threshold parameter l , they delete the longest triangle edge which represents the boundary of the convex hull. The generated shape lies then within the convex hull and gives usually a more detailed polygonal representation of the point set. However, this approach only handles simple polygons and lacks in the detection of holes. Regarding the agricultural field boundaries, a final step for approaches like the α -shape or the convex-hull algorithm has to be added.

The resulting boundary has to be finally enlarged by the working width to get the real field boundary. A very simple approach is the blow-shrink algorithm which is originated from raster calculation. Bartelme [1989] explains this algorithm for raster data which originally targets on the filtering or generalization of e.g. classification errors in raster data sets.

An efficient and accurate computation of field geometries for farmland will make the digital boundaries available that are needed to improve processes for administration and field processing. The currently used methods are not able to fulfill these criteria. In this chapter, the state-of-the-art methods are implemented and parametrized. The α -shape method and the blow-shrink method are used for a agricultural field-computation for the first time. It will be shown, that these algorithms will improve the quality of digital agricultural field boundaries. Furthermore, a parametrization approach for the grid method will be given and the dependencies on the used data set will be explained.

5.2.2 Definition of Field Boundary

As the term field boundary is not clearly defined, commonly used definitions are given. To clarify the term in context of this work it is also specified what field boundary stands for in this thesis.

InVeKoS Classification Schema in Germany

The InVeKoS classification schema defines field boundaries for its usage in the context of agricultural subsidies. A definition of the different boundary types is given in the Federal Law Gazette of Germany of 2004 [Bundesgesetzblatt, 2004].

1. **Feldblock**

A contiguous agricultural area with permanent bounds that is farmed by one or more farmers with one or more crops completely or partly disused.

2. **Schlag**

A contiguous agricultural area of one farmer with one crop completely or partly disused.

3. **Feldstück**

A contiguous agricultural area of one farmer with one or more crops completely or partly disused.

4. **Flurstück**

An area bounded by land register.

These definitions are mainly related to the InVeKoS context and fulfill its requirements. While the first three definitions include also semantic attributes like owner and crop, the fourth definition only takes the area into account which is bounded by the land register. This means that the surveyed and officially correct boundaries for fields commonly differ from the machined parts in case of not drivable areas, wetlands, scarps or even rearranged parcels. In the context of this thesis, the practical usage of field boundaries in context of agricultural routing is more relevant. Therefore, the term field boundary will be defined for this context with more relevance on the application.

Definition of field boundary in context of this thesis

The InVeKoS schema defines a field boundary in the context of agricultural subsidies which does not fulfill the requirements of a field boundary representation for routing purposes. Due to the fact, that there is not even a uniform definition of a field boundary within the German borders, an own definition that fulfills the requirements of agricultural routing is given.

Definition (Field Boundary). *A field boundary is the contiguous agricultural area of one or more owners with one crop that is completely farmed within one harvest year. The geometrical representation of the field boundary is a polygon or polygon with holes, where the holes represent not farmed regions. The gateways from road network on the agricultural fields are named field connection points or field gateways. They represent possible gateways for machines to drive from the road network on the agricultural area.*

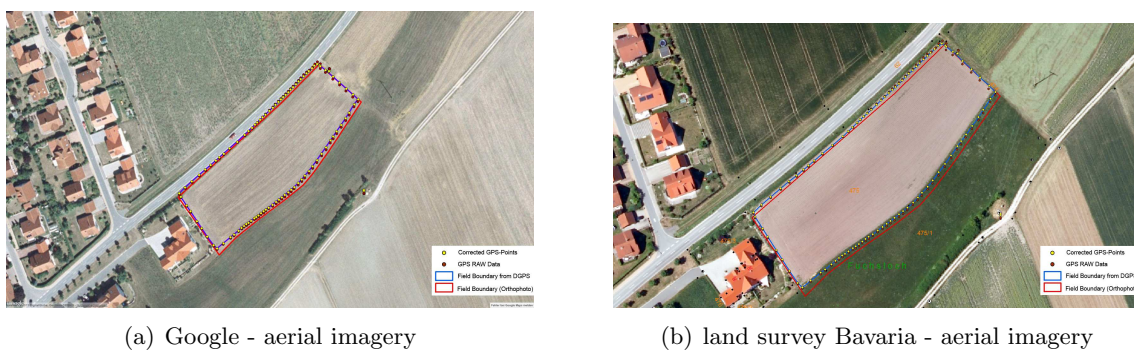
5.3 Field Reference Data

Generating accurate reference data for agricultural fields is a very time consuming task. Although farmers usually have their fields as digitized field boundaries in context of InVeKoS or within their electronic field record system, there is no information about the accuracy of the boundaries. Many farmers generate these boundaries by digitizing based on aerial imagery. Alternatively, they use handheld GPS devices with less accurate sensors to survey the field boundaries. Therefore, a survey of reference fields to have very accurate data for comparisons is needed.

The accurate survey of field reference data is done during a field survey. The fields are measured using a *Magellan ProMark3* DGPS with a rover and a reference station. The adjustment was done in post processing mode. Using the stop & go method the rover has been placed for 15s on a fixed position and a shape point for the field boundary was stored.

In figure 5.2 the four with DGPS surveyed field boundaries are shown. The fields are chosen by their different structure and size. The field in the upper left has an irregular convex structure and a power pole within. The one in the lower right is a small, more rectangular field and both, in the lower left and upper right, have convex boundaries. The field in the lower left is partly surrounded by higher trees which can influence the positional accuracy of the on board GPS device. These highly accurate measured field boundaries represent a typical set of different field boundaries.

Figures 5.1(b) and 5.1(a) show the differences between a digitized field boundary on different aerial imagery, the raw GPS points and the corrected GPS points after the DGPS post-processing step. They clarify the variation of the different boundaries and the reasons for these inaccuracies. It is obvious that these variations will result in an inaccurate field boundary and accordingly to negative effects on every calculation that is based on field boundaries (e.g. billings). For our purposes the reference field boundary has to be as accurate as possible. Therefore, a limited set of accurate surveyed geometries has been prepared.



(a) Google - aerial imagery

(b) land survey Bavaria - aerial imagery

Figure 5.1: Google and land survey administration Bavaria aerial imagery, DGPS points and field boundary

Reference Field Boundaries



Figure 5.2: Reference fields (Datasource basemap: ©OpenStreetMap contributors)

5.4 Computation Methods

Several approaches to extract a polygon from point cloud were evaluated on the field data set. In the following sub chapters, the different methods are explained in detail and (dis)advantages of each will be carved out.

5.4.1 Alpha-Shapes

The firstly in Edelsbrunner et al. [1983] mentioned α -shape approach is used for many applications from OCR [Packer et al., 2011] to building boundary extraction where Shen et al. [2011] introduced “A new algorithm of building boundary extraction based on LIDAR data”. The α -shape is a generalization of the concept of the convex hull which is an α -shape with $\alpha = \infty$. By setting $\alpha = 0$ the result is the point set itself.

Calculating the α -shape

The α -shape has been created using the PostGIS/pgRouting α -shape function. The function is a wrapper function based on CGAL [The CGAL Project, 2013]. An early stable version of pgrouting (where the α -shape function is integrated) does not allow to specify the α -value. Instead of this, an optimizer for the α -value is integrated. To get a proper α -value the α -shape needs to satisfy the following two properties:

1. the number of solid components is one (this parameter is set by pgRouting)
2. all data points are either on the boundary or in the interior of the regularized version of the α -shape.

If no such value is found, the iterator points to the first element with α -value such that the α -shape satisfies the second property (adapted from http://doc.cgal.org/latest/Alpha_shapes_2/classCGAL_1_1Alpha_shape_2.html).

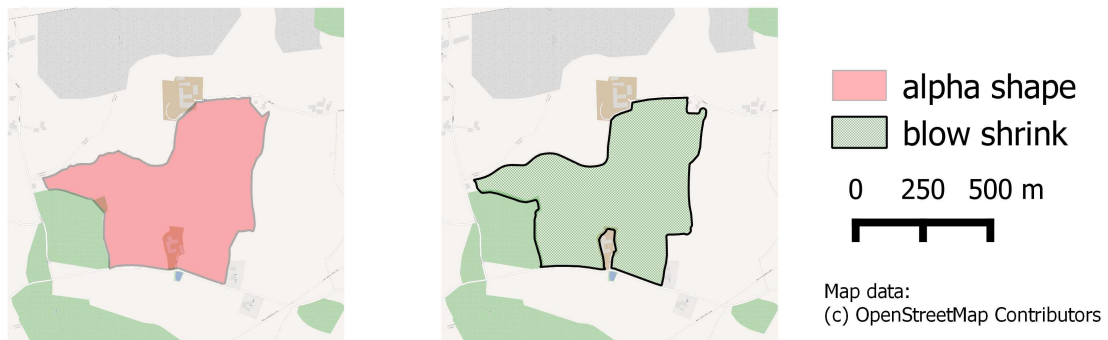
The used stable version of pgrouting (2.0) does not support α -shapes with holes. For most of the field boundaries this restriction would have no impacts. Regarding the fields with hedges or areas with high soil moisture inside (which restricts these areas for farming) the applied algorithm would only extract the outer boundary. This effect also appears when e.g. farm buildings are enclosed by a field (see figure 5.3).

The α -shape method calculates a boundary using the tracking points of a machine. The tracking point represents a location that is usually a central position on the machine. Hence a buffering step to enlarge the shape including the half working width of the machine is needed. This step will be done after the α -shape algorithm using the resulting geometry.

Optimizations for the algorithm are the integration of the specific setting of the α -value and the extension to handle multi-polygons. Extracting holes using this approach is bounded by the scanning frequency of the telemetry system. The available 15s data limits the hole detection to bigger holes (such as moist parts that are not drivable or larger areas of trees or hedges). Small areas like power poles or small cellphone towers are not detectable by this approach and this scanning frequency. For this detection the point cloud needs a minimum density.

The previously mentioned missing handling of polygons with holes and the missing integration of the α -parameter is included in a newer version of pgRouting which is only available in beta status during this thesis. Using the 2.1.0-beta version of pgRouting, the α -value can be specified by the user and enables further analysis and optimization for field boundary generation.

Field Boundary Generation - Alpha-Shape v.s. Blow-Shrink

Figure 5.3: Field Boundary - α -shape vs. Blow-Shrink

5.4.2 Raster approach

Kortenbruck and Griepentrog [2014] presented a raster based approach to generate field boundaries from agricultural movement data. They transform the machine trajectory (in their data the GPX-track of the agricultural vehicle) into an IO-matrix. The pixel size of the matrix is variable and based on the working width of the machine or the attachment. With considering the spatial neighborhood of a pixel they filtered data that is tracked on road network (relevant field pixel needs a minimum number of neighbor pixels). Areas that resulted from a defined threshold of pixels were also deleted (to eliminate crossroads and gateways). They used a not further specified algorithm from image processing to extract the boundaries of the leftover areas. To filter wrong extracted polygons within the farm area (that mainly result from manoeuvring), all polygons within a specified threshold around the farm were excluded.

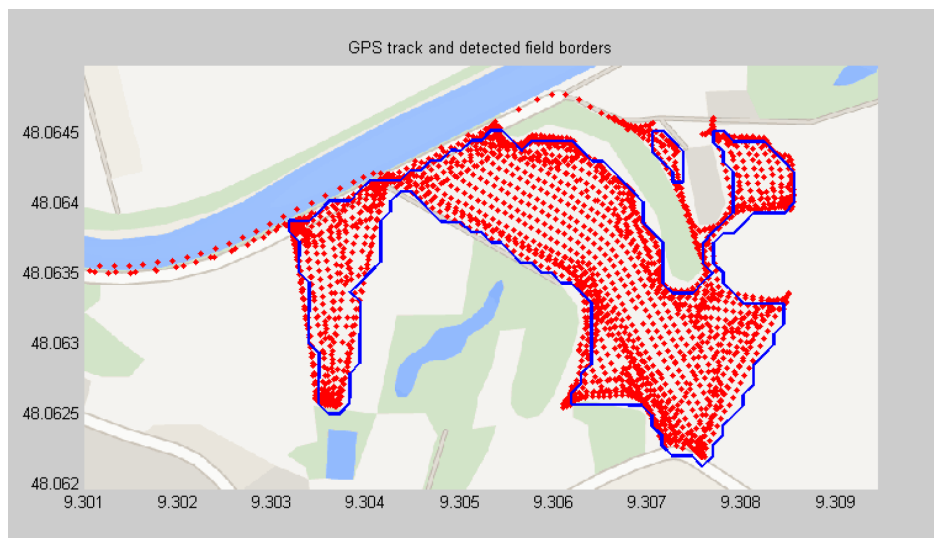


Figure 5.4: Field boundary computation - raster based approach (source: Kortenbruck and Griepentrog [2014])

Figure 5.4 shows the resulting field boundary (blue) after the processing of the GPS points (red). This figure also shows problems of the approach like separating of neighbor fields and the rough boundary through polygonization of the raster cells.

Adaption on TeleAgro+ Data

The implementation of the raster approach on the TeleAgro+ data needs some data adaption. In the original approach, parameters and thresholds are not specified precisely. Also the measurement frequency is not mentioned. Therefore the implemented approach is extended and a modified version of the primary presented approach of Kortensbrück and Griepentrog [2014] has been implemented. More details on the parametrization are given in the comparison sub-chapter where the parameters are set for raster computation approach.

Calculated field boundary - Raster based approach (Raster width 10m, 15m, 20m and 25m)



Figure 5.5: Field boundary computation for TeleAgro+ data - raster based approach (map data: ©OpenStreetMap contributors)

Figure 5.5 shows different values for raster width using the TeleAgro+ data of a forage harvester. The red boundary is a precise measured reference field boundary.

5.4.3 Blow Shrink

Method and Parametrization

The blow-shrink algorithm makes use of the fact that fields are usually larger and broader than road structures. Blowing geographic features and union with neighbour polygons will grow them, a larger (as the first blow) shrinking process or negative buffer will decrease all structures and eliminate small structures and growth structures will left. A final blow step including an estimated working width will grow the borders to the final extend. This process originates from raster processing where pixels grow by extending them to their surrounding neighbor pixels and shrink by deleting the boundary pixels. Having vector data sets like points, line strings and polygons, the blow-shrink algorithm can be realized using a buffer approach, blow existing geometries by adding a buffer and shrinking them using a negative buffer.

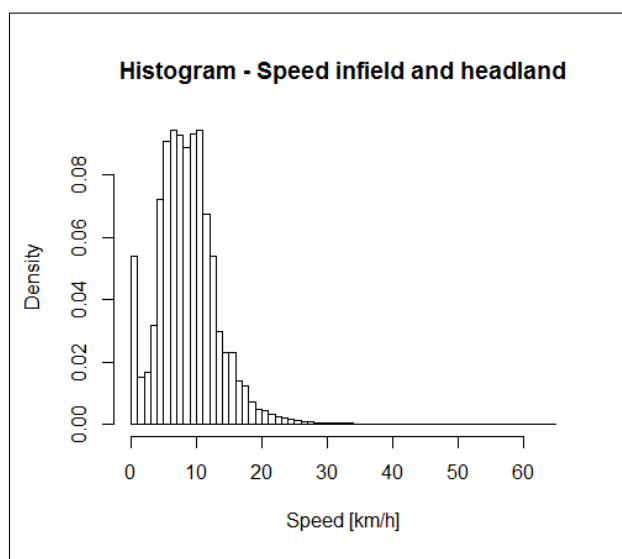


Figure 5.6: Speed statistics for infield and headland

The approach and its results mainly depend on the input data set (points or line strings) and the blow/shrink parameters. With the temporal interval of 15 s between the measurements, the agricultural vehicle makes between 0 m and 167 m at maximum speed. The analysis of the infield and headland measurements gives an average speed of 6.5 km/h which means a covered distance of 27 m. If the zero speed values are let out, the average speed is 8.7 km/h which means a covered distance of 36 m. These numbers give the reason for the different needed parameters in point blow-shrink and blow-shrink of the trajectories. Furthermore they give a rule of thumb for the parameter estimation. The histogram in figure 5.6 gives an overview on the distribution of speed values infield and in headland area for all machines.

Figure 5.7 illustrates the differences between using a point or a line string data set as input. Using a point data set, the buffer size should be larger, especially if the distance between two successive points is higher than the chosen buffer value. This delimitates the point based approach to be only feasible, if the input data is dense enough. The results of the point based approach also produce unnaturally rough borders which can cause problems and needs a post processing step to generalize or smooth the field boundary. Using the trajectory line strings as set of connected measurement points, the boundaries are more straight as they are derived from linear interpolated trajectory points. The distances between two trajectory points are short enough to assume the movement as straight or nearly straight from one point to its successor. Due to the fact that the measurement frequency is determined by the data set (5-15 s machine dependent) this variable is fixed for further processing. The remaining variables are the values for blowing and shrinking. these values depend on the GPS accuracy (the positional accuracy), the working width of the machine or the attachment and the overlapping between two lanes.

Blow-Shrink Comparison Measurement-Points vs. Tracksegments

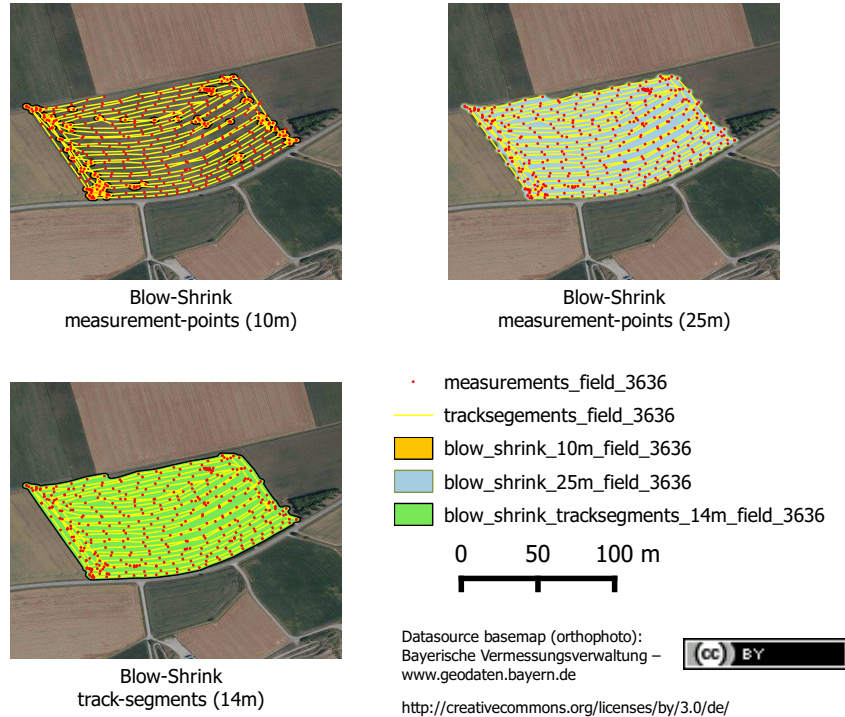


Figure 5.7: Field boundary Blow-Shrink - Comparison measurement based vs. tracksegment based

The parametrization of the blow-shrink contains three variables (blow, shrink and final blow) where the first two depend on each other. The starting blow-shrink step is used to union the geometries and for deletion of small artifacts and filtering larger structures. After the blow step, a spatial union of the created overlapping geometries is done. The following shrinking (negative buffer) with a minimal larger value than the preceding blowing will erase the small polygons that are not belonging to a field. Hence the first blow-shrink parameters are dependent and set to an equal or minimum different value (to compensate numerical inaccuracies). As the trajectory is produced by the GPS which position is measured in the machine center, a last blow step with a value of 50% of the working width will increase the polygon to the natural extend. The influencing factors for the estimation of the three parameters are:

- GPS accuracy
- logging frequency
- driving speed
- working width
- working width overlap
- field structure (complex, lots of curves, large/small)

Most of these factors can be formalized into equations for the different buffer types.

For the trajectory blow-shrink the blow-shrink parameter are then:

$$\begin{aligned} blow_1 &= GPS_{acc} + 0.5 \cdot WorkingWidth - WorkingWidth_{overlap} \\ shrink_1 &= blow_1 + \varepsilon \\ blow_2 &= 0.5 \cdot WorkingWidth \end{aligned}$$

Due to the fact that the measurement points are connected through their temporal relation, the chance to overlap parallel buffered lanes is higher than using only the individual measurements. This leads to the fact that the influence of logging frequency and vehicle speed increase if they cause a larger distance to preceding and successive measurement than the distance to the parallel trajectory (working width overlap). For the point based blow-shrink the gaps between the points in driving direction should be closed. With the estimated average speed of 8.6km/h it needs a logging frequency of more than 12 Hz, better 15 Hz or 20 Hz which means a measurement every 5, 4 or 3 seconds. At typical working widths, the measurements will then be distributed in driving direction and orthogonal to driving direction (parallel lines) equally. The blow-shrink parameter for the measurement based blow-shrink are then:

$$blow_1 = GPS_{acc} + \begin{cases} 0.5 \cdot WW - WW_{overlap}, & \text{if } 0.5 \cdot WW - WW_{overlap} > avg(distance(m_n, m_{n+1})) \\ avg(distance(m_n, m_{n+1})), & \text{otherwise} \end{cases}$$

with $WW := WorkingWith$ and $m = measurement$

Computing the Working Width

An essential attribute for the later parametrization (especially for the (blow-shrink approach) is the working width of the vehicle. The working width is the width of the machined area in driving direction which is in most cases similar to the width of the front attachment or the implement. Only at finishing sequences where the machine is cutting less than the working width is a threshold between e.g. the cutter width and the real cut crop area which is commonly termed as part width. A different case is the situation when the machine is not covering the complete ground. This is the case e.g. with a pick-up implement that is mainly used for grassland harvesting. In this setup the work steps of mowing will cover the whole area. The following rotary rakes will merge the harvest to smaller lanes which are picked-up afterwards by the forage harvester. With aiming a good parametrization of the blow shrink algorithm by estimating the working width using the trajectories, the working width will be generalized as the average distance between parallel trajectories on field. With this method, a field specific working width can be calculated and statistics for individual machines/attachments and fields can be generated. The presented algorithm makes use of the fact, that usually the field area (apart from headland) is only passed once by working step and the driven trajectories are parallel.

Working width computation for sample field



Figure 5.8: Computing the working width for one field (1. estimate mean working direction, 2. build orthogonal line string that is placed in field center and cut by outer trajectories, 3. compute intersection points and divide line string length by #intersection points)

Although the presented method needs pre-calculated field boundaries for identifying the field related trajectories, the field boundaries have no direct influence on the working width calculation. The working width is calculated solely on the trajectories and their intersection with a cropped line string orthogonal to the main working direction of the field. The steps are described in figure 5.8 in detail.

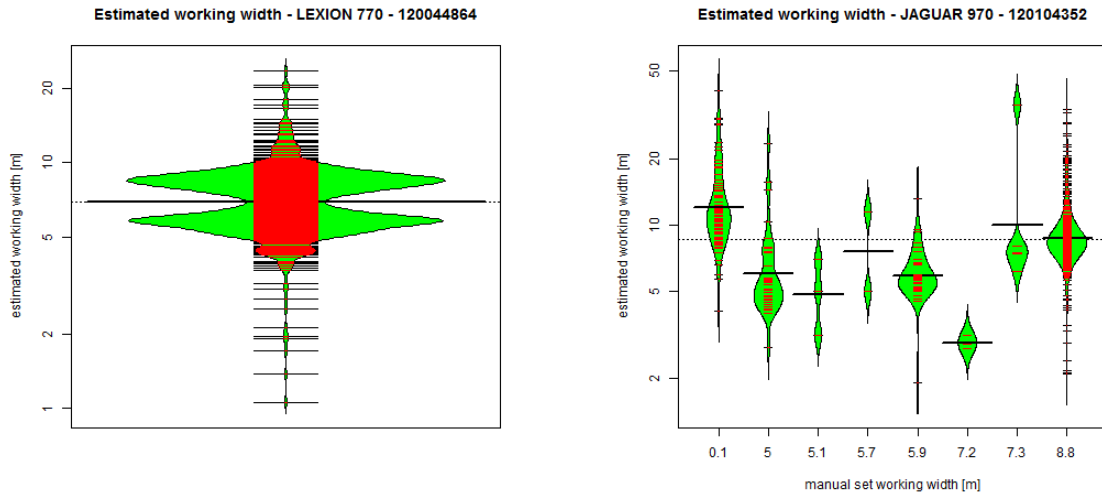
The estimated working width is then calculated as:

EstimatedWorkingWidth =

$$\frac{\sum \text{length}(\text{orthogonalMultiLineString})}{\#I(\text{TrackSegments}_{\text{field}}, \text{orthogonalMultiLineString}) \#D(\text{orthogonalMultiLineString})}$$

, with I=Intersections and D=Divisions

Figure 5.9(b) and 5.9(a) show the estimations of working width for a Jaguar 970 forage harvester and a Lexion 770 combine harvester. While the Jaguar 970 has manual data for the working width in its telemetry data, the Lexion 770 does not have manual tagged measurements. The beanplot [Kampstra, 2008] of the Lexion 770 shows two distinct peaks that denote two different front attachments. The Jaguar 970 has manually added “ground truth” data. As formerly described, the lower manual set working width represents the pick-up attachment, which has a virtual working width, that is much larger than the attachment width. Due to the less accurate driving of the rotary tedder, the swather and the multiple working processes, this estimation underlies a larger variance. The higher manual classified working widths represents e.g. “Direct Disc” headers for whole plant harvesting or “Orbis” header for maize harvesting. In these cases, the estimations give plausible results.



(a) Estimating machine working width (Lexion 770)

(b) Estimating machine working width (Jaguar 970)

Figure 5.9: Estimated working width for Jaguar 970 forage harvester and Lexion 770 combine harvester

5.4.4 Postprocessing - Optimization

OpenStreetMap Data for Boundary Refinement along Roads

Usually fields are partly enclosed by roads which are not part of the agricultural field. Due to the different geometry operations and e.g. inaccuracies or a final buffering step, the computed field boundary can overlap road structures which will not happen in reality. Overlapping field boundaries can also cause a merging of two or more naturally divided fields through geometric operations. Using a buffered road network and cut the buffered road features from the processed field geometries will split them and let them approximate more to their real extent. The data used for this step is already available from the map matching process of section 3.2. Due to the GPS inaccuracy the buffer size for matching has to be chosen wider than the real road to catch the measurements. For this step, the buffer size has to be reduced to a normal road width to cut this area from the generated polygons and split formerly overlapped and merged fields into separate fields. The reduction parameter depends on the former set matching buffer width and should be reduced to a plausible value of five to seven meters which is a feasible value for most roads.

Sieve Filtering

The geometric processing generates several smaller artifacts. Especially cutting the road network or smaller measurement agglomerations will produce not connected snippets (e.g. islands) which cannot be seen as field boundaries. Therefore a sieve step is established after cutting out the road structure as a final filtering step. As a rule of thumb, a threshold of $1,000m^2$ can be used. Nowadays where crop fields and green land are mainly managed by large size agricultural machinery this threshold is plausible and can be rather set higher in future. For processing field structures (e.g. speciality crops), which data is not used in this thesis this threshold value is feasible but can cause the lost of smaller fields.

Geometry generalization

The resulting field boundary geometries usually consist of a number of shape points originated by the foregoing processing. The bulk of these shape points are an over specification and include a large overhead for server-client transfer e.g. in the demo application. Furthermore, a generalisation will improve the field structure eliminating small humps that are not realistic for processed field areas and usually result of the processing algorithms. Hence a generalization algorithm is applied on the results of the field boundary computation algorithm. A widely used algorithm is the generalization algorithm of Ramer [1972] and Douglas and Peucker [1973]. The Douglas-Ramer-Peucker algorithm for polygons is used to generalize the resulting polygon from the field boundary computation algorithm. The simplified polygons are then stored additionally in the table and used for further processing (e.g. field connection points and the field service).

5.4.5 Comparison of the presented approaches

To compute the correctness and accuracy of the individual methods their results are compared to a reference data set. As a high accurate ground truth data set of four fields is surveyed, these four fields will be used for the computation of a similarity value. As a first step, the computed field boundaries were compared visually. This shows similarities and differences between the individual approaches. For automatic processing, a more authoritative approach has to be established. Therefore in the second subsection a measure for the similarity of sets, the Jaccard index will be introduced.

Visual Comparison

The visual comparison of the computed field geometries gives an overview on the performance of the different methods and parametrization for each of the reference fields. For the smaller, nearly rectangular field in figure 5.10, the α -shape and the blow-shrink algorithm both generate a small corner in the northern end while the eastern edge of the reference field is not covered by the computed boundaries. This is mainly caused by the classified measurement points where the first point of the classified trajectory lies outside the reference boundary. The buffered α -shape fits better due to the added working width. Especially for simple geometric structures like the first reference field, the generalized blow-shrink approach fits better to the reference boundary. The visualization of the different parametrized grid (10 m, 15 m, 20 m and 25 m) clearly demonstrates the problems with this approach. While the small grid sizes extremely underestimate the field polygon (the individual grid cells have no intersection with each other and cannot be merged), the larger grid sizes produce an unnatural zig-zag boundary. Only the 25 m grid approach produces enough overlap and merges the grid cells to a simple polygon without holes.

Figure 5.11 shows a typical mid-size agricultural farmland with straight edges and edges along a meandered watercourse bounded by hedges and a small forest. The α -shape and blown α -shape perform well and represent the meander structure on the northern edge. The blow-shrink overestimates the field boundary. This could be the result of GPS-inaccuracies next to the hedge structure that can have negative influences on the measured positions. The blow-shrink approach also tends to generalize which appears mainly in closer curve structures. The simplified approach eliminates the meander and in this case, produces worse results than the raw blow-shrink approach. The grid approach performs similar to the first field. While smaller grid sizes are not able to merge the

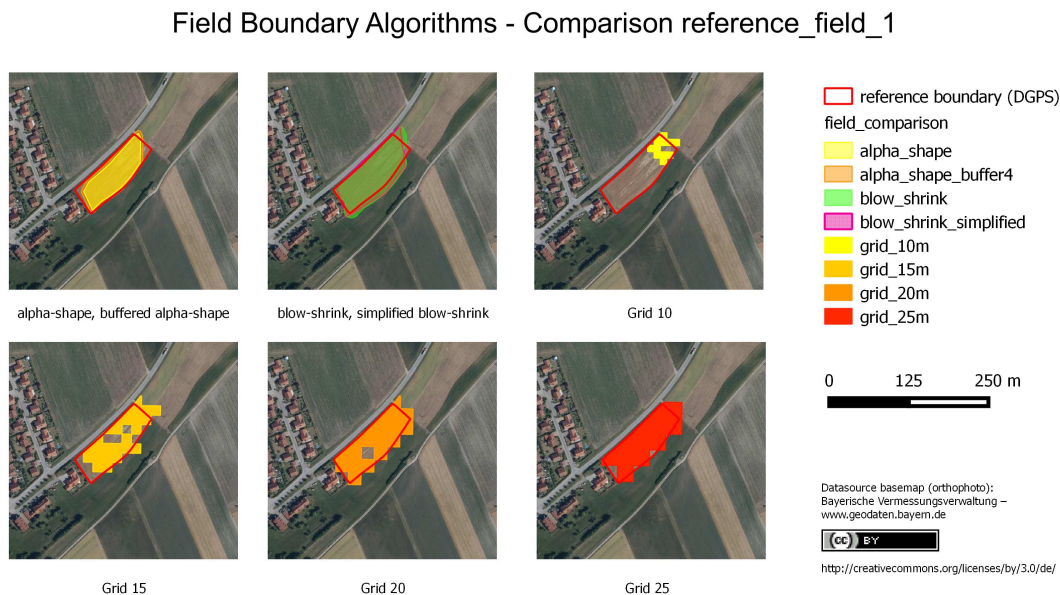


Figure 5.10: Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field.01

grids, bigger sizes will produce a continuous polygon. The 25 m grid approach only has one hole where no 25 m grid cell is intersected with a measurement point. The straight edge in the western part is represented very well due to the fact that this edge is nearly north-south oriented. The meander structure in the north cannot clearly be reproduced since the meanders and grid size are not fitting. Consulting the Nyquist-Shannon sampling theorem [Shannon, 1998], the cell size of the grid should be less than 0.5 of the meander wave length to reproduce the meander structure. This fact collides with the minimum cell size which is needed to merge the cells.

The computation of the third reference field boundary, figure 5.12, shows the problems that can occur due to wrong classified measurement values. The values in the north-western part of the field boundary are classified incorrect. The α -shape takes these measurements into account and spans a large edge structure to the outlier. This can be prevented calculating better α -values that exclude these points. As the other methods also suffer from these wrong classified measurements, a better classification result from the foregoing pre-processing step will improve all methods. However, the impact on the α -shape is much bigger than on the other approaches. The blow-shrink method only generates a small distortion wedge along the wrong classified driven path. The intersection with the road network clearly cuts the grid cells on two bounding edges which improves the results of the grid based approaches tremendously. However the small grid sizes are not able to reconstruct the field boundary.

The fourth field in figure 5.13 represents the most complicated boundary of the reference fields. The not rectangular shape has a small alcove in the mid north and a power pole that is also mapped in the reference data. The small stand of the power pole is not recognized by any of the applied methods. While the α -shape does not recognize the alcove, the blow-shrink approach without simplification shows a small alcove in this place.

Field Boundary Algorithms - Comparison reference_field_2

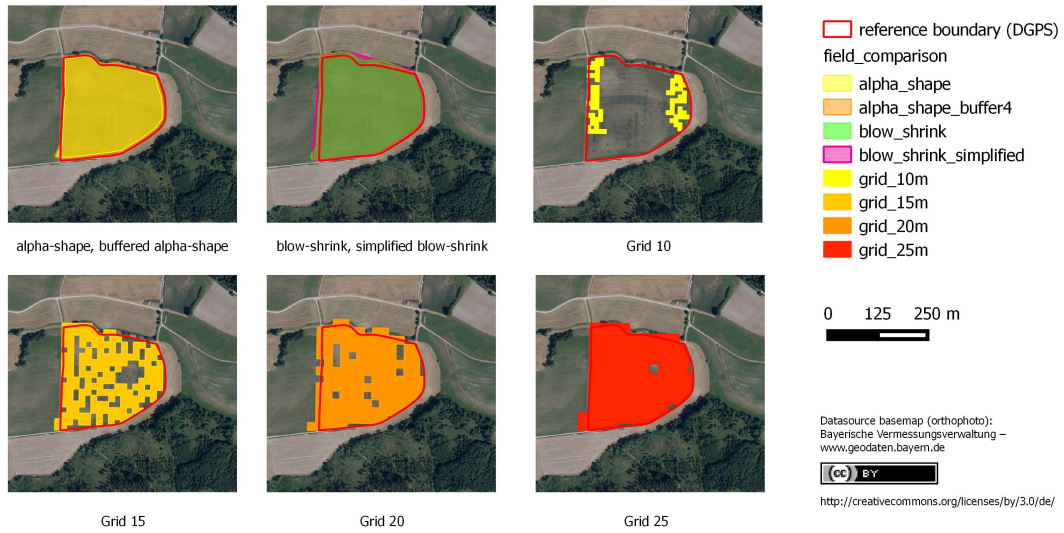


Figure 5.11: Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field_02

Field Boundary Algorithms - Comparison reference_field_3



Figure 5.12: Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field_03

The clear extraction of the alcove is not possible with the blow shrink approach. This results from the size of the buffers needed to merge themselves being as large as the alcove width. The only approach that represents the alcove is the 15 m grid approach which is caused by the sampling theorem. The 15 m grid size produces a continuous polygon and the cropped edges by the buffered road network give well fitting boundaries, but the grid size is not large enough to prevent the holes where no measurement intersects with a grid cell.

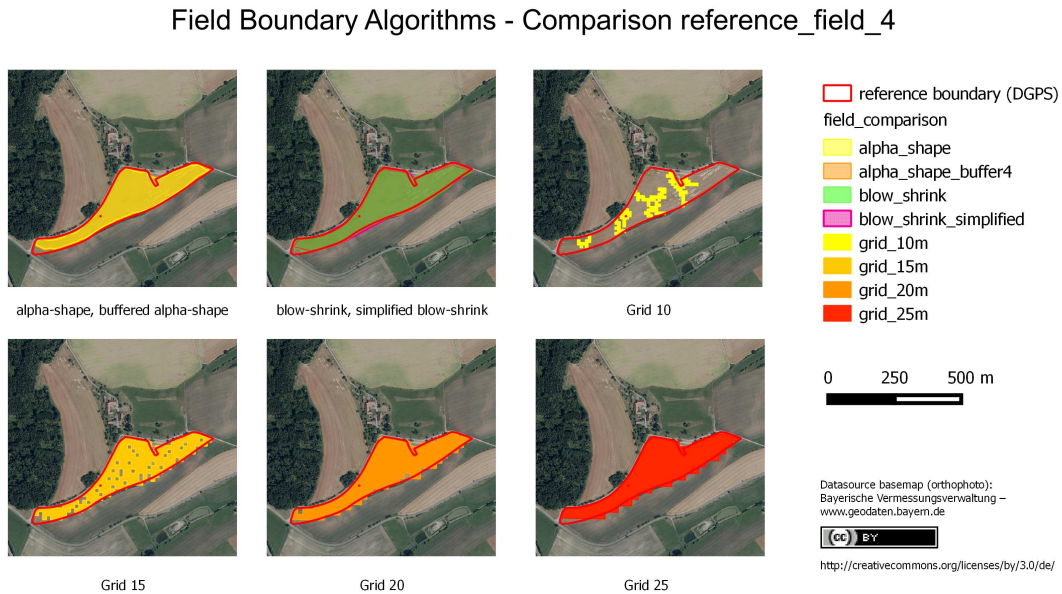


Figure 5.13: Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field_04

Jaccard Index

Due to the fact that a visual comparison of the applied approaches is not applicable in software, an index including the requirements of good field boundaries has to be established. A couple of indexes and similarity measures for geometries exist and each one has advantages and disadvantages, depending on the application. Simple measures can be the area, the outline or the number of shape points. Distance metrics, such as Hausdorff or Fréchet distance, depend on the distribution of shape points and aim mainly the matching of the outer polygon boundary.

The requirements for field boundary comparison are mainly the handling of different boundary granularity and the focus on area and shape. Algorithms for shape fitting often use translation, rotation and scale as variables. The computed geometries are equally oriented and placed. The computed boundaries are based on the same measurements. Therefore, the scale has only a small influence on the produced results. Only small variations caused by generalization and enlarging of the final geometry can have an influence on the results. These requirements are fulfilled by the Jaccard index, firstly introduced by Paul Jaccard [Jaccard, 1912]. He used this approach of the quotient of intersection and

union of two sets to compare the similarity and diversity of sample sets of species:

$$\left(\frac{\text{Number of species common to the two districts}}{\text{Total number of species in the two districts}} \right).$$

Since the used set operations are also basic GIS operations, this approach can easily be adopted on the field boundary geometries. The advantage of the Jaccard index is the fact that it includes the shape and the area of the polygons. The geometries can be seen as sets and geometric functions can be interpreted as set operations. Means: the bigger and uncomplex the field boundary (as it is in most cases for agricultural fields), the more robust is the metric. Jaccard defines his index for the sets $A, B \neq \emptyset$ as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The Jaccard index lies in the range of $[0; 1]$ with 0 meaning completely disjointed geometries and 1 meaning identical geometries. In our case a value of 1 is only possible, if the computed geometry is 100% identical to the reference data which is usually a not reachable due to different computation methods. However, this index gives a good value for the similarity of the geometries. Table 5.1 shows the pros and cons for the application of the Jaccard index to compare the estimated field boundaries with the reference data.

Table 5.1: Jaccard index - pros and cons

| pro | con |
|---|---|
| <ul style="list-style-type: none"> • simple measurement (only geometry shape as input data) • measurement for similarity of geographical objects • includes position, shape and area • represents the disparities of different shapes • fast and easy to compute | <ul style="list-style-type: none"> • no differentiation between error types (less shape points, shape point precision, shape precision e.g. holes etc.) • no distinction between over- and under-estimation • no tangible information for parameter optimization of the field boundary computation methods |

Comparison using the Jaccard Index

The Jaccard index has been applied to the four reference field boundaries. In figure 5.14, the index for all computation methods is visualized in a bar plot. The assumptions from the visual comparison are mirrored in the measures of the Jaccard index. The 10 m grid is not usable for this measurement density for any of the field geometries. Also the 15 m grid gave too low values for the index in three cases. Field 3 is the outlier because of the wrong classified measurements outside the field boundary where the α -shape algorithm spans a large edge outside the reference field boundary. The blow-shrink approach is the most stable approach for all four reference fields. However, the α -shape with the additional buffer performs very well, if the measurements are classified properly and a good estimation for the α -value exists.

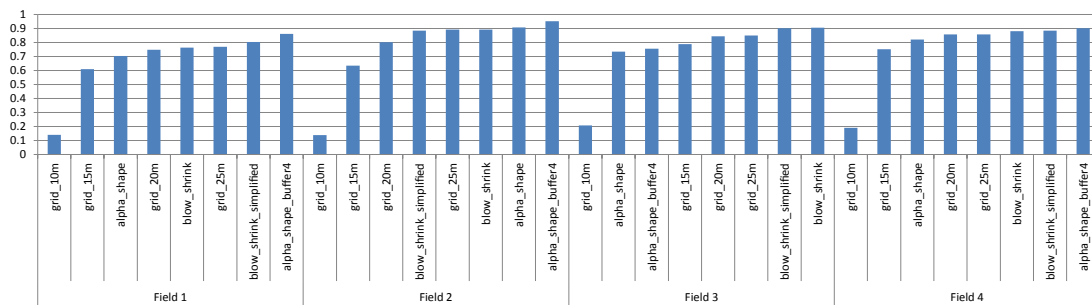


Figure 5.14: Comparison of the different approaches using the Jaccard Index

5.4.6 Conclusion

Table 5.2: Field boundary algorithms - pros and cons

| pro | con |
|---|---|
| Grid Method | |
| <ul style="list-style-type: none"> • simple method • good for rectangular, grid oriented fields • cutting the square edged polygon using buffered narrow road segments will improve the boundaries | <ul style="list-style-type: none"> • produces not natural, square-edged boundaries • nearby fields are not easy to distinguish • quality depends on grid size and measurement density |
| α-shape Method | |
| <ul style="list-style-type: none"> • only one parameter and the final buffer • popular method for surface estimation from point sets | <ul style="list-style-type: none"> • not robust to outliers • needs good estimation of α-value • works only on the point cloud (not aware of the driven trajectory) |
| Blow-Shrink Method | |
| <ul style="list-style-type: none"> • inherits trajectory \rightarrow represents close to the real driven trajectory • easy approach with basic GIS methods • robust to extreme outliers | <ul style="list-style-type: none"> • needs parameter optimisation (blow-shrink parameters) • produces large amount of shape points (need for simplification) • needs trajectory data (not only the point measurements) |

The previous analysis shows that a computation of field boundaries using agricultural telemetry data gives suitable results. The results were analyzed visually and index based. Table 5.2 summarizes the pros and cons for each approach and its requirements and restrictions. The classification of measurement points has a large influence on the field boundary computation. Especially the blow-shrink and the α -shape method are very sensitive to wrong classified points that lie outside the real field boundary. The blow-shrink approach can compensate this using the shrink where structures with a small width (less than the shrink value) will be eliminated. This is e.g. the cause for single drives off the field. The grid based approaches are very sensitive to the spatial density of the points. For higher point densities (higher measurement frequency and close parallel lanes), the grid approach with smaller grid size is assumed to give better results. However, the serrated pattern at the boundary will still exist. In regions where the field is bounded by the road geometry, this can be handled by refining the field boundary through cutting of a buffered bounded road segment. In regions where the field is not bounded by a road, these serrated patterns can be eliminated with generalization or smoothening algorithms. This processing is needed to compute a realistic shape. All presented approaches need a pre-classified point cloud. Some of the fields were machined with more than one machine within different time windows. This makes the detection of coherent *TrackSegments* hardly possible and causes wrong computed or fractional field boundaries. Complex field geometries with internal structures (hedges, scarps, single trees, utility poles, groves, stony or moistness areas) that cannot be machined can only be detected if the structures size is large enough compared to the measurement point density.

Having a proper classified set of trajectories, the blow-shrink method will give feasible results and provides realistic field boundaries. The approach is robust on extreme outliers. With a simplification of the shape geometry, this method is able to extract ready-to-use field geometries (e.g. for a field record system).

However, each approach has its advantages and disadvantages (cf. 5.2). Rule of thumb measures were given for all of the presented approaches with feasible results. A further parametrization e.g. through machine and process parameters such as the working process and therefore in detail driving speed, logging frequency, working width and the driving maneuvers could help to improve the results.

The results of the algorithms to compute field boundaries are very close to the high precise measured reference data and therefore a possible and time saving approach to actualize the farm internal field record system and e.g. field polygons that have to be digitized in the context of the Integrated Administration and Control System (IACS). The usability for the show case agricultural routing will be shown in chapter 6. The presented approaches are integrated in the context of field boundary computation for agriculture.

Since the algorithms requires trajectories and movement data, further applications in other domains are thinkable. Due to the popular and prevalent tracking of persons (e.g. through GPS devices and smartphones) and vehicles, a large number of accessible tracking data exists. Possible extensions would be the analysis of GPS traces of people that move on squares, open areas, park areas or cars that drive on large parking areas. The detection of these polygonal structures can be a next step to use the presented algorithms in a more general context.

5.5 Field Gateway Computation

Field boundaries for documentation and analysis are only one piece for a successful farm to field routing. In topographical complex regions it is not possible to drive from road to field at each part of the field boundary. Usually, the edges are preferred for the beginning of field work, if they are accessible. In this part a computation method for calculating field gateway points is introduced. This method is mainly based on learning from history: previous used gateways can be used for similar machines. These points are comparable with doors of a building which represent the access points for the indoor area.

5.5.1 Geometric calculation of Field Gateways

Due to the availability of the previously computed field geometries and the machine trajectories, a first naive approach is the geometric intersection of trajectories and field polygons. The intersection points are then potential gateways. This strategy tends to result in many field gateway points e.g. if the machine driver performs u-turns out of field instead in the headland area. This will be improved by taking only these trajectories into account which do not exceed a maximum number of intersections with the field polygon. The parameter (*MAX_INTERSECTIONS*) can be specified by the user and is set to 3 by default. Figure 5.15 illustrates the intersection points and the subsequent filtered field gateways. This will give a feasible result set of field gateway points.

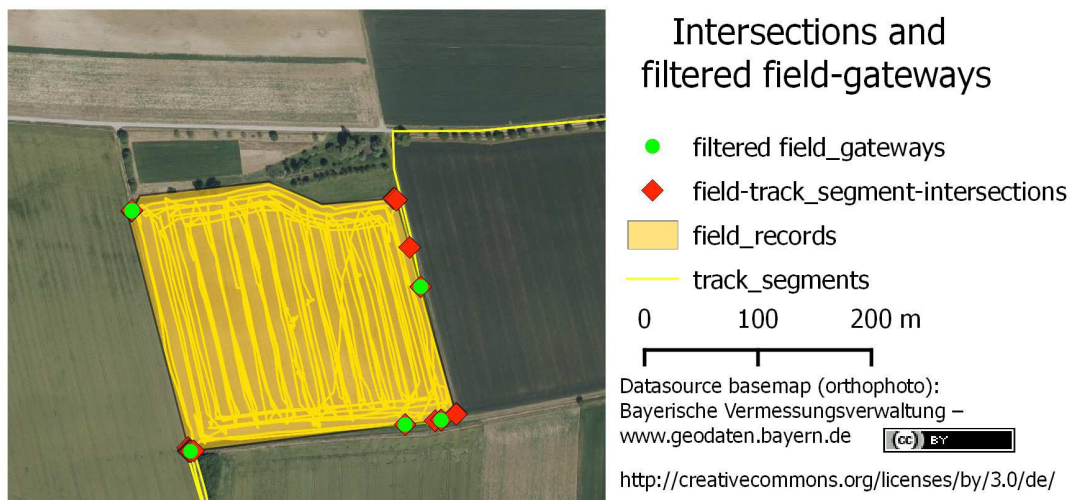


Figure 5.15: Field gateways filtering

As shown, the results are reasonable and ready to use. However, improvements could be achieved with higher complexity and further analysis. Promising strategies that are given by expert analysis of driving behaviour or visual analytics of the trajectories and field boundaries could be:

- step through trajectory and accept only intersections with minimum number of pre- and post-measurements on street/field

- clustering gateway points by clustering algorithms (e.g. peak detection on lines string or clustering algorithms like DBSCAN Ester et al. [1996])
- classification of field boundaries (drivable culvert, complete accessible edge, ditch, embankments or slopes)
- using further data, e.g. road and land use data to classify gateways by adjacent contextual information

5.6 Integration in the Processing Framework

The field geometries are calculated for all available machines and stored within the TeleAgro+ data store. The fields and their belonging field gateways are published by a field service which is an essential part of the whole TeleAgro+ system. The service is implemented as a RESTful web service that provides its data as JSON code for the TeleAgro+ web client. The API allows request by harvest year, fieldID, machine, owner and bounding box. These requests are also integrated in the web interface on client side 5.16.

REST-requests for the field service:

- by ownerID
`http://<IP-address>:<port>/VehicleService/rest/fieldservice/field_record_list/owner/9002`
- by fieldID
`http://<IP-address>:<port>/VehicleService/rest/fieldservice/field_record/2928`
- by bounding box (filtered by harvest year, owner)
`http://<IP-address>:<port>/VehicleService/rest/fieldservice/field_records_by_bbox/10.228632843796685/49.64144035194935/10.234356677834624/49.64371565773744/owner/9001/2013`
- get available harvest years for ownerID
`http://<IP-address>:<port>/VehicleService/rest/fieldservice/field_harvest_years/owner/9001`

The result is either a single field or a list of fields that fulfill the set restrictions. The request for available harvest years is used for filtering the requested fields from user interface to prevent traffic intense responses. The JSON code in listing 5.1 shows a response from a single field request. The responding JSON code is used for displaying requested fields and field gateways as shown in figure 5.16 and for building an XML-request for the MARS-route service in chapter 6.

Listing 5.1: JSON response on field request from field service

```

1 {
2   "farmNo":0,
3   "fieldNo":0,
4   "gateways":[
5     {
6       "id":20395,
7       "point":{"type":"Point","coordinates":[10.229337931112767, 49.64207638785487]}
8     },
9     {
10      "id":20396,
11      "point":{"type":"Point","coordinates":[10.229375353974094, 49.64201664233769]}

```

```

12     },
13     {
14         "id":20397,
15         "point":{"type":"Point","coordinates":[10.229330275626406, 49.642095074852556]}
16     }
17 ],
18 "generatedBy":"FieldBoundaryExtractor",
19 "harvestYear":2013,
20 "id":2928,
21 "passableAreas":[],
22 "simplifiedFieldBoundaries":{
23     "type":"Polygon",
24     "coordinates":[
25         [
26             [10.229211655885914, 49.64238462494925],
27             [10.232458705047277, 49.643034789655104],
28             [10.232869000886101, 49.642949384811324],
29             [10.23362408193011, 49.643141384766324],
30             [10.233777865745571, 49.642960258550126],
31             [10.23373641224658, 49.642734820598825],
32             [10.229363225262018, 49.64201464498365],
33             [10.229211655885914, 49.64238462494925]
34         ]
35     ]
36 },
37 "subFieldNo":0
38 }

```

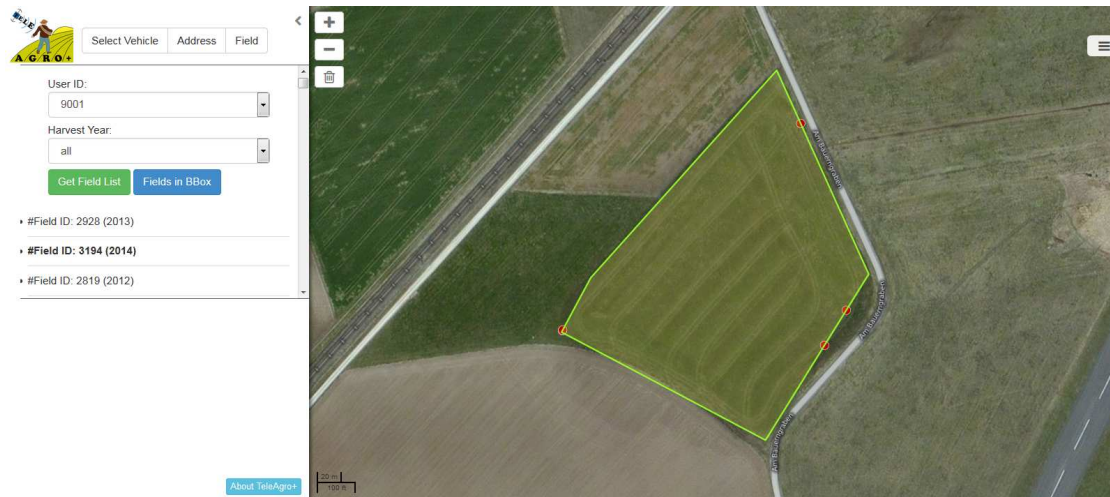


Figure 5.16: Field request in the TeleAgro+ GUI

5.7 Results

In this chapter approaches for field boundary computation on agricultural telemetry data are presented. A distinct overview on field boundary definition has been given beforehand. Different methods for geometrical boundary generation have been analyzed and their assets and drawbacks were discussed. Own approaches have been compared to published methods from literature. The only known approach that handles this task in the domain of agriculture is the published grid-approach, but neither parameters nor results for larger data sets are given. With the telemetry data as input, an experimental parametrization for this approach has been made to get good and reliable results. The cost intense generation of reference data limits the comparison of the analyzed methods to only a small subset of fields. However, a larger reference data set will give more insights into the characteristic of the different methods and will lead to a better parametrization. The results have a good fitness for use and can be integrated in the agricultural route planner that is the demon-

strator software for the analyzed methods. Regarding the field gateways this analysis is the first known that extracts field gateway points from real driven trajectories. Using this extracted information can be a benefit and an important factor for further planning of track networks and maintenance. This becomes important since the vehicles generating these trajectories are usually heavy weight/big size vehicles that have an obvious impact on the track network condition and road surface. With the working width estimator a new algorithm for the estimation of real working width is presented. This algorithm is able to calculate the working width which has an impact as parameter on different methods for field boundary computation. Furthermore, the computed values can be compared with the theoretic possible working width of the machine or the attachment. The estimated value can be an indicator for the earning power of of RTK guidance systems or can be used to compare individual machine drivers. It can also be used as input parameters to improve field harvesting or field work strategies in general as presented in Bochtis et al. [2010].

Chapter 6

Agricultural Routeplanning

6.1 Introduction and State of the Art

Routing is one of the most common applications using digital geographic data sets. The engineering of shortest paths on graphs, like Dijkstra [1959] improved vehicle route planning and became a useful tool improving way finding and logistics. In the year 2000 the former president of the United States, Bill Clinton, decided to turn off the *Selective Availability* of GPS [D’Roza and Bilchev, 2003], a degradation of the public GPS signals which implies a variance of the position accuracy of about 100 m (instead of 10 to 15 m without this noise signal). This far better positional accuracy enables real time navigation and the development of car and truck technology received a tremendous boost [BMVI, 2004]. With this, navigation and route planning became common in cars and public road network vehicles. Especially through increasing traffic, growing road networks, new vehicle sensors and the extreme rising transportation sector [European Commission EC - Directorate-General for Mobility and Transport, 2012], the requirements for navigation and route planning became more specific. Regarding the primary sector, the previously mentioned improvements had also a large influence on agricultural business processes. As the GPS on agricultural vehicles is mainly used on fields (e.g. parallel driving, precision farming), the installed hardware, the terminals and communication devices are nowadays also used for routing and planning logistics chains. This usage extends the requirements on data sources (maps), data structures and routing algorithms. Domain specific use cases such as infield routing indicate new algorithms and data sets. This enables interruption free routing for trucks from the road network on fields to e.g. a harvester. Through the availability of vehicular telemetry data, the usage of real time measurements and driven routes seems to reason. These subsidiary challenges are enlightened separately in the following subchapters.

6.1.1 Routable Road Data

Route calculation requires a routing algorithm, a proper data structure and as important it is fully dependent on the geospatial data input to achieve high quality results. Still, today the quality of geospatial base data for routing lacks in completeness regarding geometric road network and its attribution. Proper edge weights to determine realistic time of arrival are not available due to missing data. Furthermore, road networks are not static and underlie permanent changes such as the construction of new roads or the improvement or degradation of road parts due to different road management. Especially the lower graded

road network like pedestrian paths, bicycle trails and the very important for forestry as well as for agricultural logistics tracks and forest ways are underlying a constant change. Tracks in particular have a large influence on agricultural route planning because of their origin as planned transport roads [Bont, 2012], Stöcker et al. [2004]. The commercial mapping company Here shows in an online viewer that for Germany the changes in length of the street network are greater than seven percent within one year [Navteq, 2012]. To know where and when road network changes occur requires an up-to-date road network data set, which is very cost intensive to maintain for large areas. Having limited resources, generally areas with high frequency are prioritized for map updating compared to rarely used small tracks. However, in particular these small roads are of major interest for local routing in rural areas. And they are the last miles to get from the public road network onto the fields where the biomass is growing.

Within the last years Volunteered Geographic Information (VGI) as a new data source for geodata has become increasingly used as up-to-date source [Goodchild, 2007]. The meaning of VGI is to collect geographic data from volunteers. Everybody can act as a sensor and commit her or his impressions to big data collections. The benefit is like Aristoteles (384 BC to 322 BC), the Greek philosopher, described: “The whole is more than the sum of its parts”. In contrast to former common ways of mapping, precise maps of a few experts, the crowdsourcing approach benefits from the mass of individuals. This new way of earning geographic data is one of the most promising innovations of the last years. Research activities by Roick et al. [2012], Neis et al. [2011], Neis and Zipf [2012], Zielstra and Zipf [2010] in data quality analysis showed that the first doubts on the quality of user generated content were not confirmed. With primarily practical show cases, Neis and Zipf [2008], Müller et al. [2010] demonstrate that, especially within the VGI world, fitness for use is the forceful argument for using geodata made by volunteers. The publication of Heipke [2010] reviews the existing scientific work on crowdsourcing data and gives a brief overview on widespread usage of crowdsourced data. The most prominent example of crowdsourcing geoinformation is the OpenStreetMap (OSM) project with more than 2.2 Mio registered users (12/2015) [Openstreetmap, 2015]. Based on OSM data several studies investigate how to prepare the OSM data structure as route graph for further routing applications [Luxen and Vetter, 2011, Müller et al., 2010]. As one of the first approaches Neis and Zipf [2008] demonstrated the fitness for use of crowdsourced map data for routing. Especially the wheelchair routing in Müller et al. [2010] demonstrates how data collected from volunteers can help to solve sophisticated routing problems. Compared to usual car routing problems, routing wheelchairs needs more differentiated edge attributes (like surface tags, incline and road curb heights) and weight functions for calculating an accessible way for disabled people. Hartmann [2013] provides a service that provides processed OSM data sets for Garmin GPS devices that also implemented several basic routing functions for bicycles. In particular the quality of the crowd sourced data sets is one of the most discussed topics and has already been assessed in several articles [Ciepluch et al., 2011, Neis et al., 2011, Haklay, 2008, Kounadi, 2009]. Also the OSMMatrix tool [Roick et al., 2012] shows user actions and feature distribution that give an indication of quality. They summarized that the distribution of data quality is not uniform. The quality and the completeness of data correlate with the number of editors in a region. Especially in rural areas where not that many mappers work actively on the map, the data actuality and quality is behind the urban areas. Regarding fitness for routing, the quality of the street network including the topological correctness is sufficient to derive a proper routing graph and to calculate routes [Neis and Zipf, 2008]. They implemented a full functional

online service for route calculation based on OpenStreetMap Data. This service is publicly available and well established in the community. Press releases and many news group articles show that the service is usable and returns plausible and feasible routes¹.

Hence, first demonstrators for using VGI in routing applications exist. However, there is still a gap between leisure use cases (e.g. hiking, biking) and professional or commercial use cases like the transportation of biomass on rural road networks. Therefore, one objective in this chapter is the analysis of fitness for use for routing agricultural vehicles on crowdsourced road data.

For calculating routes on the public road network, which is nearly homogeneous for vehicles (like cars) regarding edge weights with a very small range of attributes or restrictions, many operational algorithms and tools already exist [Eksioglu et al., 2009]. This can be seen by the high number of route planning web sites as well as the established add-ons of navigation systems in cars and also the common smart phone route planning applications. Several research groups are working on improving exact route planning by using time dependent routing. Haghani and Jung [2005] present a genetic algorithm to give a nearly optimal solution for the NP-hard pick-up or delivery vehicle routing problem with soft time windows for vehicles with different capacities. Calculating fuel-efficient routes is the topic of Ganti et al. [2010] within the GreenGPS project. They developed and implemented a model that predicts fuel consumption on calculated individual vehicle type parameters and further input variables such as vehicle mass, elevation, frictional forces and acceleration. With this model and empirical data from some heterogeneous streetcars, they predict a “gallons per mile”-attribute that represents the edge weight for routing. The objective is the calculation of the route with the highest fuel-saving. Most of the work is focused on heterogeneous vehicles in sense of capacity (different capacities for freight transportation). Especially in the operations research domain this problem is very common and used for logistics optimizations [Chao, 2002]. However, calculating exact routes for very heterogeneous vehicles is still a problem. The knowledge about the road network as well as the information about the vehicles are needed and both should be integrated into equations to get a numeric value as edge weight for shortest path computation like done in Ganti et al. [2010]. Base data sets from companies like TomTom or Here are only applicable for routing on public road network. Restrictions like maximum weight or clear height partly exist but maintenance is expensive and technically extensive. These restrictions are mainly mapped by using signposts. Attributes, such as real width of the road or the drivable part do not exist for most of the road data. Hence it is not recommendable to assess these variables by time consuming field surveys and automatized ways are needed.

Another aspect is the field of real time routing. Calculating routes by regarding the actual traffic situation and modifying the routes in time is relevant for urban areas. Approaches that consult traffic sensors or get traffic flow information by mobiles [Fawcett and Robinson, 2000] are state of the art. Standardized interfaces for providing sensor data like OGC SOS [Bröring et al., 2012] gave the opportunity for extensions of routing by using e.g. traffic data [Mayer, 2009]. Companies like Google or Here implemented this kind of traffic into their applications to provide time dependent routing. These applications and the experience in time dependent routing made by customers confirm the demand for integration of real data into routing and navigation applications. As the routing for agricultural vehicles is less influenced by time dependent traffic situations (agricultural lo-

¹cf. <http://openrouteservice.org/contact.html#services>, (accessed 2016/07/16)

gistics usually happens not in typical urban areas with daily rush hours and traffic jams), the research in this work will not handle the integration of public available traffic data. The focus will be on the individual analysis of driving patterns and travel times on the heterogeneous rural road network. This will have an impact on the predictions made for planned routes (travel time and route geometry).

6.1.2 Computation and Data Handling Complexity

Route planning with complex edge weights requires a closer look at their data structure in terms of computational handling and memory consumption. Compared to a common route graph the multi attributed graph has to store a variety of attributes, also with different data types such as distance, road type, surface attributes, maximum speeds, dimensions, etc. [Graf et al., 2011, Luxen and Vetter, 2011, Lee et al., 2008]. Precomputed graphs that are commonly used for fast routing engines such as contraction hierarchies [Geisberger et al., 2008] are not feasible. Due to the large combination complexity of vehicle restrictions and route attributes, a precomputation of all combinations is not possible. Hence, a data model that considers a multitude of different attributes is needed.

6.1.3 Edge Weight Calculation

Beside the way of route planning for cars, specialized route planners have been developed for other domains. Most prominently bike and pedestrian routing systems have to be mentioned (<http://www.bbbike.de>, (accessed 2016/07/16), [Fu and Hochmair, 2009]), as well as routing applications for wheelchairs [Kasemsuppakorn and Karimi, 2009], in forestry [Stöcker et al., 2004] and in agriculture [Wörz et al., 2013]. All of them implemented basic restrictions and routing functions, like shortest or fastest route or functions for general vehicle classes (e.g. car, truck, bike and pedestrian) which is mainly due to missing input data on multiple attributes describing the road networks with rich details beyond the geometry. These approaches did not consider the variety of vehicles and their attributes and calculated routes just for very generic vehicles. Precise driving time estimations were done i.e. in the domain of logistics [Haghani and Jung, 2005] but hardly used as weights for route planning especially within the domain of agricultural routing. A very close work is the analysis of urban GPS traces by Brundell-freij and Ericsson [2005]. They used GPS traces for calculating statistics for street types, driver attributes and further variables. But the integration step into a routing service is missing there.

6.1.4 Infield Routing

Routing on (public) road network is only one part of agricultural routing. Collecting the harvest good from a harvester or switching the trailer to target a gap less harvesting process during forage harvesting are typical examples of scenarios where a precise route description from the road to the field boundaries and within the field area is required. While road networks are mainly represented as line strings, the agricultural fields are represented as polygonal areas. Bringing these two types to a seamless graph structure, algorithms for graph calculation and transit points are necessary. Similar problems are described in the field of indoor routing or routing in free spaces for pedestrians [Goetz, 2012, Isert et al., 2013]. Further domain specific approaches are described in Ali et al. [2009] where the theoretical optimization of the harvesting process is in focus. In this work, they used simple grid structures to generate the infield graph. Hameed et al. [2010] evaluated 15

different field shapes and generated a fieldwork pattern for each field. A split algorithm that divides the field area and recursive that is feasible for real time usage to generate a working path for the agricultural machine is presented in Oksanen and Visala [2007]. Zhou et al. [2014] show a further approach where they consider obstacles within the field and optimize the working path by dividing the field into blocks. The generation of optimal path is made by find the optimal block traversal sequence, formulated as TSP (Traveling Salesman Problem) which is solved by the ACO (ant colony) algorithmic approach [Dorigo and Gambardella, 1997].

Although these methods generate graph or graph like structures, their main focus lies on optimizing the harvesting process for the harvesting machine. Also the seamless routing from the road network into the field is not part of these more harvesting process focused publications. Therefore, the adaptability of infield graph generating algorithms for the agricultural use-case has to be investigated.

Methods for infield graph generation are described with advantages and disadvantages for the agricultural routing use-case in the infield routing subsection 6.5. Additionally, a new method is presented, which takes the requirements for agricultural routing into account. To enable a seamless routing, a further method for calculating gateways will be given.

To sum up, the related works for this article exhibits manifold research lacks with respect to routing of agricultural vehicles in areas with low order road networks. First, there is a lack of input data to build routing graphs accounting for multiple road attributes to derive edge weights for different types of vehicles. A road data set that fulfills the requirements for agricultural routing is needed. In particular this data set should be up-to-date and thus requires a straightforward mechanism of (self) maintenance. Second, a new data structure is needed to handle computation complexity of on-demand edge weight computation from multiple attributes for single edges in order to derive vehicle specific edge weights. A new data structure has to be developed. Third, this input data set and data structure should integrate and make beneficial use of empirical data from agricultural telemetry systems to attribute routing graph edges. Algorithms and mechanisms for linking routing graph and telemetry data are needed. And at least, to route the last mile on the agricultural field, methods to generate an appropriate infield routing graph are presented and evaluated.

6.2 VGI data for Agricultural Route Planning - OSM

The OpenStreetMap project is described in general already in chapter 2. In this section a more route planning specific analysis is given with a particular focus on relevant attributes for agricultural routing and their occurrence in the OpenStreetMap VGI project.

6.2.1 OSM Road Types

This section describes the OpenStreetMap highway- and track-types which are relevant for agricultural routing. The highway-types are classified in a large number of classes where only the relevant ones are used for the routing implementation. The track-types are categorized in five grades from *grade1* with a paved or heavily compacted surface to *grade5* that represents e.g. a path on grassland without any paved or compacted surface. The used highway- and track-types are listed in table 6.1.

6.2.2 Relevant OSM Road Attributes

A further routing relevant element (especially for tracks) is the description of the surface. In OpenStreetMap, this attribute is mainly described by two tags: *surface*, which describes the type of surface and *smoothness*, which describes the quality or the roughness of a road. The surface type consists of a large number of individual attributes where only the frequently tagged and agricultural relevant are taken into account for the routing. The smoothness tags are based on a quality scale that grade the road from *impassable* where wheeled vehicles are not able to drive and *excellent* which offers optimal conditions for in-liners or skateboards. As numerical attributes, measures like maximum width, maximum weight, speed and height are stored within the edge weight. The used surface attributes and the measure tags are listed in table 6.1.

Table 6.1: Used OpenStreetMap tags

| Highway types | Surface types | Track types |
|------------------------------------|-----------------------|-------------------------|
| motorway(_link) | unpaved | grade1 |
| trunk(_link) | compacted | grade2 |
| primary(_link) | gravel | grade3 |
| secondary(_link) | pavelstone | grade4 |
| tertiary | cobblestone | grade5 |
| unclassified | cobblestone:flattened | Smoothness Types |
| road | paving_stones | excellent |
| residential | paving_stones:30 | good |
| living_street | paving_stones:20 | intermediate |
| service | grass_paver | bad |
| track | asphalt | very bad |
| pedestrian | concrete | horribly |
| raceway | metal | very horrible |
| bus_guideway | wood | impassable |
| path | ice_road | Measures |
| cycleway | ground | axle_load |
| footway | earth | maxwidth |
| bridleway | mud | maxweight |
| byway | grass | incline |
| steps | sand | maxspeed |
| | dirt | maxheight |
| | paved | lanes |

6.3 Route Calculation with Multiple Attributes

Due to the heterogeneity of road data and its attributes on the one hand and the large amount of different vehicle configurations on the other hand, it is obvious that this combination will result in a complex route planning algorithm. The run time of the shortest path algorithm depends on the complexity of the underlying road network and on the number of comparisons that the algorithm has to compute within each step of checking

an edge for its weight. Therefore two problems have to be solved:

- 1 Designing a memory efficient data structure for storing the large amount of graph and attribute data and
- 2 Filtering and classifying relevant data for agricultural routing out of the OpenStreetMap data.

6.3.1 Memory efficient Graph Structure

Storing a graph with a large set of attributes will increase the memory consumption tremendously. Hence, a compressed storage of edge related attributes will decrease the needed storage and cause a slim graph without losing the needed information for multiple attribute routing. The implementation of the route graph is realized in Java, but the described optimizations will have the same or similar effects using other programming languages (as the used data types consume similar memory). The comparisons are explained theoretically. For a practical impression, the graphs of exemplary regions are compared with the two storing methods.

Table 6.2 shows the memory consumption of the different attributes. It shows the substantial saving of memory using a preceding classification and storage as bit chains.

Table 6.2: Memory Consumption of compressed and uncompressed Graph

| Attribute | Uncompressed Graph | | Compressed Graph | |
|-------------|--------------------|-------------|------------------|--|
| | Data type | #bit (Java) | #bit | Description |
| Length | Double | 64 | 16 | Metrical precision 1 m |
| Road type | String | 448 (288) | 6 | 27 classes (OSM highway types) |
| Max. height | Double | 64 | 8 | Metrical precision 0.1 m |
| Max. width | Double | 64 | 8 | Metrical precision 0.1 m |
| Track type | String | 448 (288) | 4 | 6 classes (grade1-5, default) |
| Smoothness | String | 448 (288) | 3 | 9 classes (incl. default class) |
| Surface | String | 448 (288) | 6 | 23 classes (incl. default class) |
| Max. weight | Double | 64 | 6 | Typical max. weight restrictions classified from European road classification and OpenStreetMap attributes |
| Axle load | Double | 64 | 6 | 18 classes (derived from merged European road classifications) |
| Max. speed | Double | 64 | 5 | Typical max. speeds classes derived from European streets in km/h |
| Incline | Double | 64 | 4 | 16 classes |
| # Lanes | Int | 32 | 3 | Number of lanes |

The memory usage of the Java String Objects is platform dependent. Häubl et al. [2008] pointed out that the minimum size of a Java string object is 36 bytes. Coffey [2011] provides an easy approach to calculate the individual approximately size of a Java string object by using the formula:

$$\text{Minimum String memory usage [bytes]} = 8 * (\text{int}) \left(\frac{(((\text{nochars}) * 2) + 45)}{8} \right)$$

That means for the attributes *Highway*-, *Track*-, *Smoothness*- and *Surface*-types an average minimum memory of 56 bytes (by formula of Coffey [2011]) is needed (cf. table 6.3).

Table 6.3: Add caption

| Name | Avg(Tag-length) | Sum [bytes] Coffey [2011] | Min [bytes] Häubl et al. [2008] |
|------------|-----------------|------------------------------|------------------------------------|
| Highway | 8.4 | 56 | 36 |
| Track | 5.666 | 56 | 36 |
| Smoothness | 7.888 | 56 | 36 |
| Surface | 8.217 | 56 | 36 |

Table 6.4 shows different graph implementations and their theoretical size. The graph size itself cannot be reduced, since the number of edges is given through the different roads and the nodes that represent the connection points. This topological structure only consists of an e.g. adjacency list that represents the connections. The most promising part regarding the reduction of graph size is the compression of the edge weight. Due to the fact, that for multi-attribute routing the edge weight consists of a large number of attributes and therefore of a large number of primitive data types that are very costly. An individual routing for the heterogeneous vehicles would not allow a pre-computation of edge weights, as the attributes would have a different influence for different vehicles. The only chance to save memory without notable information reduction is the classification and compression of the edge weights. The classification of the attributes (showed in table 6.2) and the storing using a bit array will reduce the memory tremendously. Compared with the uncompressed graph structure, the theoretical value 96 % or 95 % of memory (depending on the string memory calculation) will be saved (cf. table 6.4). The conversion of the weights can be done in a very fast way using look-up tables and simple calculations. Therefore the computation time is comparable to the version with more specialized, memory intense data types.

6.4 Cost Function with Empirical Edge Weights from Real Data

In order to calculate edge weights, an optimizing target is obligatory. Naive approaches are shortest and fastest route. The shortest route is simply calculated by using the geometrical edge length as weight. To calculate the fastest route, a velocity model is essential to calculate the time that a vehicle needs to travel the whole edge. Usually, this will be realized as specific velocity model for different highway types and vehicle types. The large number of attributes enables a complex optimization of a shortest path calculation by considering multiple factors. The attributes allow for example the preference of paved roads

Table 6.4: Graph size for Test Regions (data source OpenStreetMap: Heidelberg, Baden-Württemberg: 11-2013, Study site 02-2014)

| Area | # edges | Memory edge weight [MByte] (Coffey [2011]) | Memory edge weight [MByte] (Häubli et al. [2008]) | Memory compressed edge weight [MByte] |
|-------------------|-----------|--|---|---------------------------------------|
| Study site | 508,000 | 137.589 | 98.831 | 4.542 |
| Heidelberg | 24,000 | 6.500 | 4.669 | 0.215 |
| Baden-Württemberg | 2,370,000 | 641.899 | 461.082 | 21.189 |

or the degradation of roads with a rough surfaces or a low smoothness-type. These calculations and weight functions are mainly based on hypotheses which weight the influencing attributes. Considering the available telemetry data, an empirical approach is presented that is able to fulfill the requirements of more realistic routes. These approaches and the workflow of the empirical edge weight calculation is described in detail in the following sub chapters.

6.4.1 Calculation of Edge Weights

With the available telemetry data and with the map matched measurements in section 3.2, a computation of edge and vehicle individual weights is possible. The average driven speeds for every combination of highway-, track-, surface- and smoothness-type combination is made for each individual vehicle and generalized for every vehicle type. The result is a vehicle and vehicle type specific speed profile for every driven attribute combination.

Based on the matching, the average speeds for each vehicle and every road type combination are calculated. By using the relevant OpenStreetMap classification of highway types, track types, surface types and smoothness types we get a four dimensional matrix for each vehicle with different average speeds for every combination.

The averaging is processed in two tiers. First, the average speeds for every individual machine is calculated. That represents the regional influences and the driving style. With this, an exact route prediction for each individual machine is possible. Second, for new vehicles that are not measured by the telemetry system, the average speed of all vehicles of the same type and configuration is calculated. Uncared-for the driver style and the region where the machine is located. This allows an average prediction for this machine type without having real measurements, only using the existing data of similar machines. A sample data set is given in table 6.5.

Within the routing architecture the user is able to request the general speed for a vehicle type as well as the individual speed for one specific vehicle and starts routing with these attributes. The client fills the requested parameters pulled from the vehicle service into the more detailed *< ExtendedRoutePreference >* attribute of the extended OpenLS Standard [Mabrouk et al., 2005] and builds an extended route request for the MARS (*(M)ulti(A)tttribute(R)oute(S)ervice*) routing engine.

Table 6.5: Average speeds for *Jaguar 950* forage harvester (table reduced – some sub class values are omitted)

| avg-speed [km/h] | #measurements | highwaytype | tracktype | surface | smoothness |
|------------------|---------------|----------------|-----------|----------|------------|
| 14.4446807 | 285 | Cycleway | | | |
| : | : | : | : | : | : |
| 6.836541667 | 48 | Path | | | |
| : | : | : | : | : | : |
| 38.05867119 | 2357 | Primary | | | |
| : | : | : | : | : | : |
| 20.20037853 | 1733 | Residential | | | |
| : | : | : | : | : | : |
| 19.014 | 1 | Road | | | |
| 34.87330214 | 9383 | Secondary | | | |
| 39.94418212 | 604 | Secondary | | Asphalt | |
| 23.14911111 | 54 | Secondary | | Asphalt | Good |
| 34.57616 | 25 | Secondary | | Paved | |
| 30.79133333 | 3 | Secondary Link | | | |
| 5.15016129 | 1054 | Service | | | |
| : | : | : | : | : | : |
| 32.77981576 | 16457 | Tertiary | | | |
| : | : | : | : | : | : |
| 8.690729663 | 4868 | Track | | | |
| : | : | : | : | : | : |
| 9.191655172 | 1305 | Track | Grade 1 | | |
| 6.264727273 | 22 | Track | Grade 1 | Asphalt | |
| 30.59836364 | 33 | Track | Grade 1 | Concrete | |
| 8.157356046 | 2671 | Track | Grade 2 | | |
| 5.552853933 | 89 | Track | Grade 2 | Concrete | |
| 5.848875 | 32 | Track | Grade 2 | Earth | |
| 5.655593256 | 949 | Track | Grade 2 | Gravel | |
| 6.540298851 | 87 | Track | Grade 2 | Paved | |
| 9.266330709 | 381 | Track | Grade 2 | Unpaved | |
| 5.443401373 | 3204 | Track | Grade 3 | | |
| 3.11 | 17 | Track | Grade 3 | Grass | |
| 7.52084058 | 207 | Track | Grade 3 | Gravel | |
| 6.033188406 | 69 | Track | Grade 3 | Ground | |
| 15.68950617 | 324 | Track | Grade 3 | Paved | |
| 5.012376087 | 920 | Track | Grade 4 | | |
| 4.430212766 | 47 | Track | Grade 4 | | Very Bad |
| 9.393142857 | 14 | Track | Grade 4 | Asphalt | |
| 32.11 | 2 | Track | Grade 4 | Grass | |
| 4.494826087 | 138 | Track | Grade 4 | Gravel | |
| 6.504833333 | 12 | Track | Grade 4 | Ground | |
| 5.428928571 | 28 | Track | Grade 5 | | |
| 3.416163265 | 245 | Track | Grade 5 | Grass | |
| 1.326888889 | 18 | Track | Grade 5 | Ground | |
| 3.603530435 | 115 | Trunk | | | |
| 22.86066667 | 6 | Trunk Link | | | |
| 20.25423516 | 16261 | Unclassified | | | |
| : | : | : | : | : | : |

6.4.2 Vehicle Data Integration

The for each individual vehicle and for every aggregated vehicle group calculated speeds are stored within a database table. This database is requested by a *VehicleService* that provides the speeds for each vehicle via REST-interface. The front end prototype is able to request the needed vehicle parameters and generates the request for the *MARS-RouteService* including the `< ExtendedRoutePreference >`. The service can be requested using following request paths:

- by vehicleID (for a vehicle type and specific configuration)
`http://<IP-address>:<port>/VehicleService/rest/vehicleservice/vehicle/{v_id}`
- by vehicleID and machineID (for a specific machine)
`http://<IP-address>:<port>/VehicleService/rest/vehicleservice/vehicle/{v_id}/{ma_id}`
- basic list of vehicles (including vehicle type and manufacturer)
`http://<IP-address>:<port>/VehicleService/rest/vehicleservice/vehicle_id_list`
- complete vehicle list (including complete vehicle information and average speeds)
`http://<IP-address>:<port>/VehicleService/rest/vehicleservice/vehicle_list`

6.5 Infield Routing Graph generation

The graph generation for polygons is a typical problem for indoor route planning and pedestrian routing in open space. Straight forward approaches are grid generation and routing on polygon edges. Visibility graphs that calculate the direct connection of the graph edges, triangulation and generalization methods are further approaches with usually higher computation complexity but less edge complexity. The methods presented in this sub section were integrated in the MARS routing system. Due to constantly changing field conditions, the focus lies on the dynamic computation of the infield graph within each service request. This allows the implementation of changing drivable areas for the route graph generation which is needed during working processes such as harvesting or seeding. The generated route graph will then be integrated in the road network graph and enables seamless routing from the road network to the destination in the field.

6.5.1 Rectangular Grid Method

This approach is based on the indoor grid graph-based model of Li et al. [2010]. The applied approach is simplified in the way that the step length parameter is set to a proper value for agricultural vehicle navigation (e.g. 10m). Further in this first approach only rectangular connections were used. Therefore the grid creation is very simple and the created edges were spatially intersected with the field geometry that only edges completely within the field polygon were used for routing. The grid is computed independently from the field orientation oriented in a rectangular way in north-south/east-west. Limitations of this approach are the mentioned orientation, the for machinery usually not applicable 90°

turns, the missing connections from field gateways to infield graph and most notably missing connections infield when the polygon structure does not allow a rectangular connected set of edges (e.g. tube structures). The number of edges depends directly on polygon size and extend. Therefore the calculation time using this approach and the following shortest path calculation grow with the polygon area.

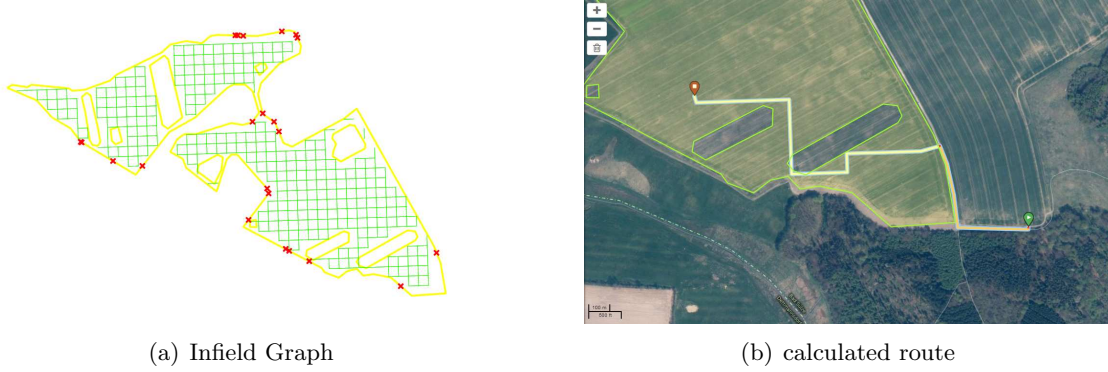


Figure 6.1: Infield routing using a grid graph

6.5.2 Oriented Diamond Grid Method

This approach is an extended grid method from Li et al. [2010]. At first a grid with 45° connections (instead of 90° rectangular connections) is created. This represents a more realistic and smooth turn for machines. The resulting grid consists then of diamonds. Afterwards the grid will be rotated that the orientation of the parallel lines is in main working direction. This is realized by a translation and rotation operation. To guarantee that the rotated grid will cover the whole field geometry the initial grid is build using the extended envelope of the field. In the final step, only these edges will be saved where both nodes lie within the field polygon.

The steps for this rotation are as follows:

1. Translate the coordinate system to the rotation point (M1)
2. Rotate around the point of origin of the new coordinate system (M2)
3. Translate the coordinate back to the original system (M3)

This leads to the following equation (with (x_1, y_1) as rotation point):

$$M = M1 * M2 * M3 = \begin{pmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.1)$$

The 45° turns are more applicable and realistic to route then using only 90° turns. The possible graph interruptions within tube structures are decreased but still remain.



Figure 6.2: Infield routing using a diamond graph

6.5.3 Delaunay Method

The Delaunay triangulation [Delaunay, 1934] is also an option for generating a topology within a field boundary. Using this approach, it is possible to integrate the destination point into the graph calculation which will improve the correct way finding. The resulting grid is much smaller than the grid approaches but limitations regarding the tube effect still remain. Further this method does not support orientations (like main working direction). Close edges with long straight boundaries will have an essential affect on generating a routing graph. These tubes will cut off the graph and a route through the polygon would not be possible. The triangulation method also does not include the main working direction. Therefore a specific directional routing is not realizable with this approach.



Figure 6.3: Infield routing using a Delaunay graph

6.5.4 Voronoi Method

The Voronoi diagram [Voronoi, 1908] is the dual graph of the Delaunay triangulation with the same points. Like the Delaunay method, it does not concern the driven paths or the main working direction of a field. Close edges with long straight boundaries will break the Voronoi structure and lead to a less applicable graph structure with many turns instead of a straight edge. Integrating the field boundary as further edges will improve the resulting graph, but the tunnel effects from long nearly parallel edges will still be problematic for feasible graph calculation.



Figure 6.4: Infield routing using a Voronoi graph

6.5.5 Line of Sight Method - Visibility Graph

The visibility graph is first mentioned in Lozano-Pérez and Wesley [1979]. They point out that in the case of motion in the plane with arbitrary polygonal objects, the shortest collision-free path connecting any two accessible points has the property that it is composed of straight lines joining the origin to the destination via a possibly empty sequence of vertices of obstacles. This approach is widely used in robotics [De Berg et al., 2008]. Also the routing of pedestrians through building structures [Goetz, 2012] is an application for the visibility graph. This means for the infield graph generation: Taking the n shape-points of the boundary polygon, the field connection points and the destination point a set of ray from each point to each other point as edge set is calculated and tested for intersection with the field polygon. Every edge which is within the polygon will be part of the infield routing graph. This method strongly depends on the number of shape points. The complexity for its calculation is n^2 . This method will give the shortest euclidean path from the field connection point to the destination using the shape points as nodes. Hence the destination point affects the graph calculation directly, this approach will not be a solution for static cases (e.g. precomputation of infield graphs). Also for complex geometries the costs for computing the graph and the shortest path increase rapidly. Compared to building structures, field structures can be more complex due to size and irregular boundaries and obstacles.

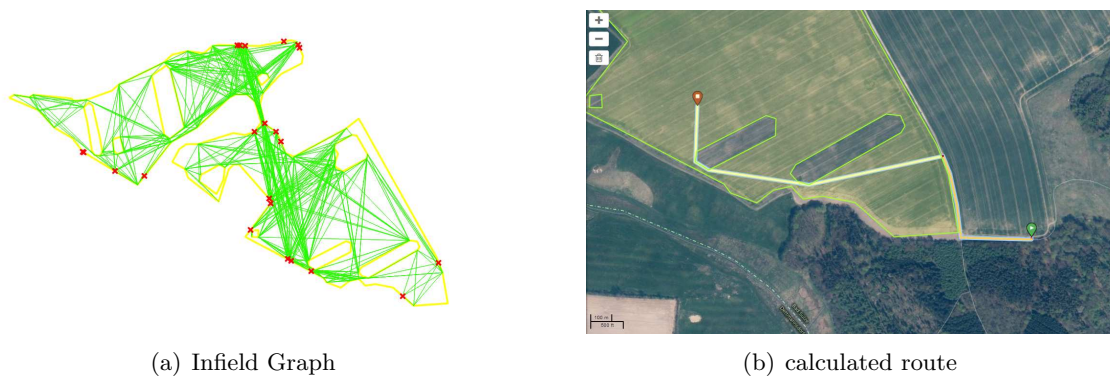


Figure 6.5: Infield routing using a visibility graph

6.5.6 Straight Skeleton Method

The straight skeleton algorithm is a computational geometry algorithm that has its origins in generalization of polygons. It represents a polygon by its skeleton. The straight skeleton can be seen similar to a medial axis of a polygon but with only straight segments instead of parabolic curves. The first definition of a straight skeleton for simple polygons is given by [Aichholzer et al., 1995]. Felkel and Obdrzalek [1998] show an algorithm for a straight skeleton implementation that computes a straight skeleton for a polygon in $O(nm + n \log n)$ time. Although this algorithm is discussed later on, there exist basic implementations that are feasible for computing an infield graph structure for field polygons. A common problem of the straight skeleton computation for the infield route graph is the fact, that the skeleton is derived by angles between polygon edges. Therefore, field connection points on straight edges are not considered in the original algorithm. Also the consideration of a main working direction will be only integrated indirectly as the straight skeleton algorithm produces longer edges in direction of the longest line within the polygon. These are approximately parallel to the longer polygon borders that are usually representing the main working direction.

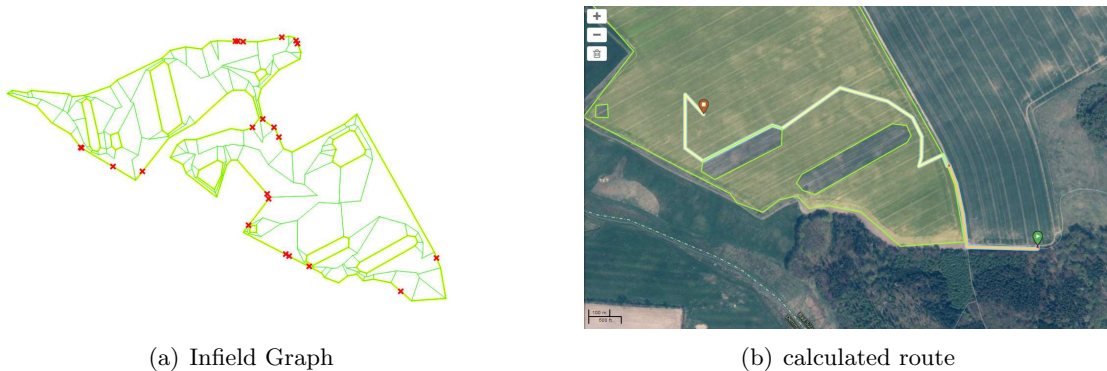


Figure 6.6: Infield routing using a straight skeleton graph

6.5.7 Method Comparison

The presented approaches follow different strategies. To compare them, a tabular comparison is made 6.6, regarding different attributes.

The attributes that specify the different approaches are explained as follows:

- **Scalability:** Factors that will increase the computational effort like polygon size, polygon complexity (complex boundary or simple polygon), number of edges and shape points and the number and complexity of holes.
- **Computation complexity:** The computation complexity for graph calculation with a given polygon and gateway points.
- **Considering agricultural requirements:**
 - Considering connection points or field gateways
 - Avoiding zig-zag routing
 - Prefer routing on paths in main working direction

- Targeting a destination point in field
- **Method specific issues:** Issues that are not handled by the above points

6.6 Integration into MARS Routing

The data for routing agricultural vehicles and the presented algorithms for infield route graph calculation are integrated into the MARS routing service. This prototype service is set up as a web service with an extended (but backward compatible) standardized OGC OpenLS interface. An example for the extension of the limited OpenLS interfaces to enable multiple attribute routing and a second extension that integrates fields as polygons with gateway points for routing destination will be given in this sub section. The route service architecture with its components and the integration in the overall context is showed in figure 6.7 .

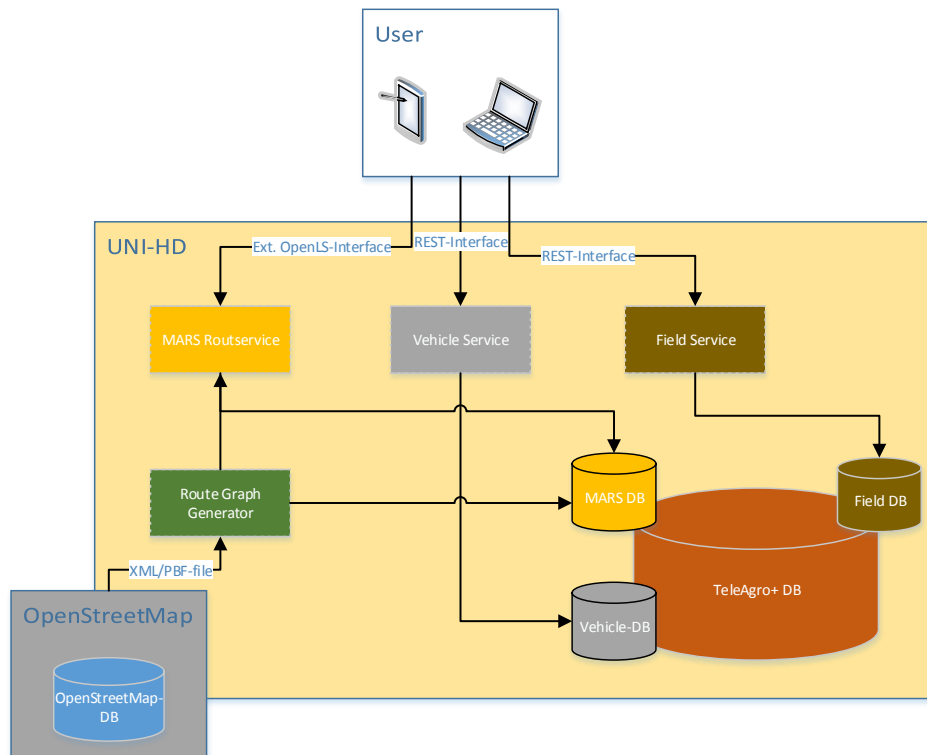


Figure 6.7: Service Architecture: Route-Service, Field-Service and Vehicle-Service

6.6.1 OpenLS Extension for Routing heterogeneous Vehicles

The standardized OGC OpenLS interface [Mabrouk et al., 2005] exists since 2005 through the collaboration of different involved parties. The interface is designed generic and, like other interfaces for spatial data populated by the Open Geospatial Consortium (OGC).

Table 6.6: Comparison of infield routing graph computation methods

| method | scalability | computation complexity | considering agricultural requirements |
|--------------------------------|--|---|---|
| Rectangular Grid | <ul style="list-style-type: none"> • polygon size • polygon structure | $\mathcal{O}(n^2) + \textit{overlapping} - \textit{delete}$ | <ul style="list-style-type: none"> • can be interrupted in small connecting tube structures |
| Oriented Diamond Graph | <ul style="list-style-type: none"> • polygon size • polygon structure | $\mathcal{O}(n^2) + \textit{rotation} + \textit{overlapping} - \textit{delete}$ | <ul style="list-style-type: none"> • routing in working direction • shortcuts to neighbor paths in 45° angle • less zig-zag-routing |
| Delaunay Graph | <ul style="list-style-type: none"> • polygon shape points | $\mathcal{O}(n \log n)$, (n = #points) | <ul style="list-style-type: none"> • concerns gateway points • concerns destination point • thin edge density in open space |
| Voronoi Graph | <ul style="list-style-type: none"> • polygon shape points • large but equal distributed polygons in open space | $\mathcal{O}(n \log n)$, (n = #points) | <ul style="list-style-type: none"> • concerns gateway points • concerns destination point |
| Visibility Graph | <ul style="list-style-type: none"> • polygon shape points | $\mathcal{O}(n^2)$ | <ul style="list-style-type: none"> • concerns gateway points • concerns destination point (but also needs it for proper calculation) • only straight lines within open space • can increase route length through detours • various edge density within the polygon |
| Straight Skeleton Graph | <ul style="list-style-type: none"> • polygon shape points • polygon structure | $\mathcal{O}(n^3 \log n)$ | <ul style="list-style-type: none"> • not concerning connection points (if they are located on a straight boundary line) • not concerning destination point |

The common routing purposes are represented in the interface specification. The service interfaces are initially implemented in XML. This interface does not consider multiple attributes and vehicle types. The specification only allows a single tag to set a `<RoutePreference>`. This tag is settable e.g. with *fastest*, *shortest*, *pedestrian*, *car*. However, this single tag is not able to fulfill the requirements of multiple attributed vehicle information integration. Therefore, an extension without tackling the original interfaces is needed. To maintain backward compatibility, the presented solution is the extension of the service interface with a multiple attribute enabled `RoutePreference`. The proposal is an extended specification with an `<ExtendedRoutePreference>` tag that consist all information on the routable vehicle. This `<ExtendedRoutePreference>` consists of the empirically computed speeds and the restrictions that limit the routing of the vehicle. Listing 6.1 shows the `<ExtendedRoutePreference>` part of an example for a possible multiple attribute routing.

Listing 6.1: Extended RoutePreferences for OpenLS Specification

```

<xls:RoutePreference>Fastest</xls:RoutePreference>
  <xls:ExtendedRoutePreference>
    <xls:width unit="m">2.60</xls:width>
    <xls:height unit="m">3.10</xls:height>
    <xls:weight unit="kg">6500</xls:weight>
    <xls:axleload unit="kg">3500</xls:axleload>
    <xls:maxspeed unit="km/h">40</xls:maxspeed>
    <xls:oneway>>false</xls:oneway>
    <xls:onewayBike>>false</xls:onewayBike>
    <xls:highwayTypes>
      <xls:highwayType>primary</xls:highwayType>
      ...
      <xls:highwayType>track</xls:highwayType>
    </xls:highwayTypes>
    <xls:trackTypes>
      <xls:trackType>grade1</xls:trackType>
    </xls:trackTypes>
    <xls:surfaceTypes>
      <xls:surfaceType>unpaved</xls:surfaceType>
    </xls:surfaceTypes>
    <xls:smoothnessTypes>
      <xls:smoothnessType>excellent</xls:smoothnessType>
    </xls:smoothnessTypes>
    <xls:roadSpeeds>
      <xls:roadSpeed>
        <xls:highwayType>primary</xls:highwayType>
        <xls:trackType>grade1</xls:trackType>
        <xls:defaultSpeed unit="km/h">40</xls:defaultSpeed>
      </xls:roadSpeed>
      <xls:roadSpeed>
        <xls:highwayType>primary</xls:highwayType>
        <xls:smoothnessType>good</xls:smoothnessType>
        <xls:surfaceType>asphalt</xls:surfaceType>
        <xls:avgSpeed unit="km/h">37.5</xls:avgSpeed>
      </xls:roadSpeed>
      ...
    </xls:roadSpeeds>
  </xls:ExtendedRoutePreference>

```

6.6.2 Integration of Field Boundaries and Gateways

The integration of polygonal destinations is the second lack that is not supported by the origin OpenLS specification. The proposal is the integration of an optional `< xls : FieldRoutingPreference >` section. The integration of the field boundary in the service interface allows the in-time calculation of the route using the most recent data (e.g. from field service). This is also enabled by the architectural design of the framework where single processes are implemented as services with specified interfaces. Therefore it is possible to compute an infield route graph which concerns e.g. the last known or predicted field boundaries. An example is the permanently changing drivable field area during the harvesting of a field. The algorithmic computation of the infield routing graph for the destination field is integrated into the MARS route service. The alternative approaches for infield route graph generations are changeable by setup the service on server side. An example of the extension for a route request with entry point(s) and a polygonal field geometry is given in 6.2.

Listing 6.2: Field Routing Preferences for OpenLS Specification

```

<xls:FieldRoutingPreference>
  <xls:EntryPoint>
    <xls:Position>
      <gml:Point srsName="EPSG:4326">
        <gml:pos>12.427874285469484 48.51023139457032 </gml:pos>
      </gml:Point>
    </xls:Position>
  </xls:EntryPoint>
  <xls:PassableFieldArea>
    <gml:Polygon srsName="EPSG:4326">
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:coordinates>
            12.427836051186633,48.51024710697691
            12.42829140517879,48.51156634048899
            12.42909671557971,48.511451882459525
            12.42880981357967,48.50984693860695
            12.427836051186633,48.51024710697691
          </gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </xls:PassableFieldArea>
</xls:FieldRoutingPreference>

```

6.7 Comparison between driven and calculated routes

To evaluate the calculated routes a comparison tool has been developed that extracts a subset of continuous trajectory segments from the telemetry data. Origin and destination of these trajectories were used as input parameter for the route services (MARS and OpenRouteService), that calculate the routes (geometry and travel time) for further analysis. The trajectory segments were cut by two conditions:

1. $\forall m_i \in M \exists m_{i+1} | (t(m_i) - t(m_{i+1})) < 16s$, with m_i as single measurement of one machine, $t(m_i)$ as time stamp of m_i and M as a set of all available measurements

2. The offered working mode parameter is switched from/to fieldwork. Afterwards the resulting trajectories were filtered by length and spanning bounding box to exclude short snippets and scattered measurements from machine halts.

After these steps the evaluation set consists of 160 real driven trajectories. The routes were calculated with the MARS routing engine using the available vehicle parameters and restrictions. For comparison, the routes from start to destination were also calculated by the OpenRouteService [Neis and Zipf, 2008] with the preference “fastest” route which is defined with a velocity profile for cars. As shown in figure 6.8 there is a strong correlation between the driven route length and the calculated routes. The differences between the driven trajectory and the calculated routes show that in nearly all cases, the calculated routes by the MARS routing were shorter or the same length as the routes chosen by the driver. The longer routes that are calculated by the OpenRouteService engine are mainly because of the limited road data for cars. Due to the fact that the “fastest” route option is only defined for cars, the used road network is limited to public accessible road network. Figure 6.9 shows the predicted time, calculated by the two route services and the time that the drivers needed to get from origin to destination. Differences are mainly based on different routes. Especially in the case of the OpenRouteService the differences are caused by the overestimated speed profiles for agricultural vehicles (the speed profile is set for common cars per default).

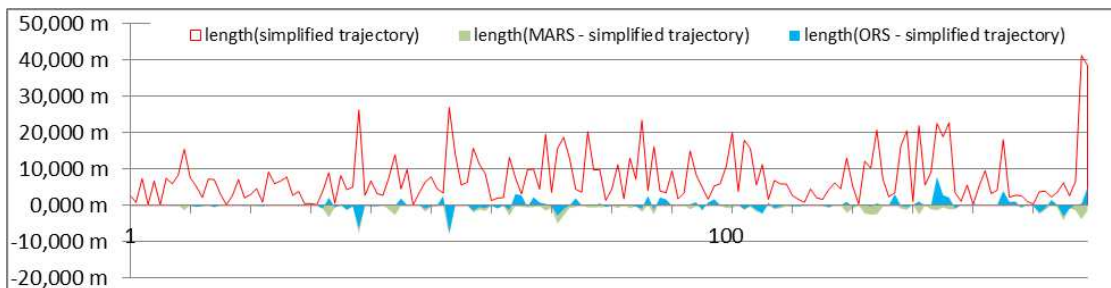


Figure 6.8: Comparison between calculated routes (ORS, MARS) and real driven trajectory - random order

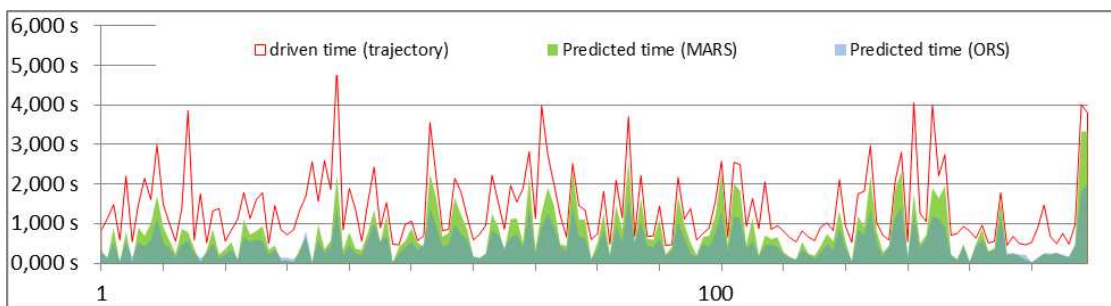


Figure 6.9: Comparison between real driving time and the predicted time from the route services (ORS and MARS) - random order

Figure 6.10(a) shows an example of a calculated route that is completely different from the driven one. While the drivers choice was to take the main roads to reach the destination, the routing algorithm suggested a route on tracks. Figure 6.10(b) gives an

example of a wrong driven route by the driver. The dead end part in the western part and the u-turn are indicators for a wrong direction. The calculated route takes a completely different direction in the northern part and goes directly to the destination. An example of three routes (driven, ORS, MARS) is shown by figure 6.11. The driver takes a route in the south while the MARS router suggests a route that is more direct, from east to west. The OpenRouteService calculates a route that prefers the higher level road types, that is mainly caused by the implemented speed profiles.



Figure 6.10: Different routes (left) and lost way v.s. correct calculation (right)
(map data: ©OpenStreetMap contributors)



Figure 6.11: Driven route v.s. OpenRouteService v.s. MARS
(map data: ©OpenStreetMap contributors)

6.8 Conclusions

This chapter presented solutions for each of the four research gaps. A data source for routing agricultural vehicles with its manifold attributes has been analyzed if it fulfills the requirements for this kind of route planning. The very flexible set of attributes and the “power of the crowd” that enlarges the free available OpenStreetMap data set over

the last years demonstrate the large potential of this data. However the quality assurance is at this time only made by the mappers themselves. There exist algorithms and tools within the set of editing tools that reduce editing errors during the mapping session and support the mappers which reduce the number of unmeant errors. However the error source through different opinions and democratic mapping will still be part of volunteered human generated data. There is no explicit quality guarantee for the free data set. The upcoming popularity of the data set and by this, the growing use within commercial applications² will presumably bridge the gaps of concerns. The vast set of attributes is not available in any other data set and the opportunity to easily update and maintain the data by customers is only in the beginning within more bounded products of established commercial mapping companies³. The various sets of attributes and the integration of general describing attributes for geographic data (e.g. surface and measurement-tags) make the data usable for a large set of applications and generate a domain comprehensive benefit. Agricultural mappers edit geographic data that is also needed by emergency, disaster response units, land use planning and tourists. Automatic quality assurance processes and continuous data quality analysis together with a close communication with the crowd and the close cooperation of application, data and the crowd will be the keys for a successful story in the future.

As second part, the challenges for the integration of a heterogeneous data set to route heterogeneous vehicles are analyzed. Through the opportunity to integrate the multiple attributes for an adapted in time route calculation, the presented approach is generic and provides a solution for a wide field of applications not only restricted to the demonstrated use case of agricultural routing. The provided approach classifies the data to a restricted but representative set of attributes that is not limiting the practical use cases. That is realised amongst others with the considering of existing classifications (e.g. for weight limits). The technical representation of the edge weights as a *bit array* decreases the needed memory for the graph structure. Further improvements, based on this approach, could be made by integrating compression methods that will further decrease the needed memory with the commitment of computation time.

The analysis of the available telemetry data enables an individual vehicle- and a more general vehicle type specific routing. The connection of the movement data through map matching with the road data and the calculation of average speeds for each road class (including attributes) and vehicle gives the input for the implemented route service. The usage of real movement data for route predictions is coming more into the focus of traffic planning companies. This is mainly caused by the availability of movement data (e.g. through smart phone applications) and the availability of on-board sensor in cars and trucks. In the specific use case of agricultural routing, the shown approach is limited to the available data. With the underrepresentation of several road classes, the influence of outliers is too large and average speeds are not representative for the vehicle and the road type. However the prediction of the routes including the empirical edge weight can improve the predicted routes and thereby increase the efficiency of a harvesting campaign or a logistic chain. Further improvements can be made by analyzing the spatial correlations and include further attributes into the analysis process. Candidates are the integration of incline, the vehicle motorization and the load weight. Furthermore, the structure of the road network (curviness, crossings and connectivity) can be influential factors for the average speed calculation.

²e.g. <https://www.mapbox.com/>, (accessed 2016/07/16)

³e.g. <http://mapcreator.here.com/>, (accessed 2016/07/16)

In the fourth part, different approaches for infield route graph generation are analyzed on their applicability for calculation of infield routes. Some of the approaches are used for routing pedestrians in open space which is a similar problem but without the additional restrictions given by the routing of agricultural vehicles on fields. Therefore most approaches are able to calculate a feasible route from the road to a point within the field. Especially the simple grid based approach lacks a realistic guidance. The more complex approaches generate connected edges which are not drivable by farm vehicles. To integrate the requirement of routing in the field working direction, it needs the improvement that is only fulfilled by the new approach of the oriented diamond grid, which is introduced for the first time. Through the availability of the telemetry data and the previous classification step, the mean working direction has been calculated and can be integrated in the grid computation process. Through this, a route graph strongly dependent on the real world can be generated. With this, the harvest transporting vehicles are able to make use of the machine track directions and avoid to cross these tracks which usually causes less comfortable driving, higher soil compaction and higher risk of bog down. The connections between the parallel edges in working direction are made by 45° edges which guarantee a more smooth driving from edge to edge. The concept of oriented graph structure can also be used for other approaches if these are able to integrate a parameter for orientation in their edge calculation. The transferability of the presented approach to other domains strongly depends on the requirements. All of these approaches can generate a connected set of edges (e.g. for pedestrian routing). Use cases where additional requirements are needed, in case of structured surface, inclines or slopes can be served by an oriented graph where edges are aligned with the orientation of the surface structure or related to the slope. Through the introduction of field gateway points, a seamless routing from road to field can be realized for all of the presented infield graph algorithms. A future improvement of the field gateway points will be a vehicle dependent classification of the points. For further improvements the classification of field boundaries as line string geometries to enable a point independent crossing of the field boundary depending on the target point within the field.

The presented approaches are integrated in the MARS route service which is implemented as a demonstrator route service. The OpenLS interfaces of the route service are extended to integrate the additional functionality. The service oriented architecture with open service interfaces is easily expandable. Through the modularization, single services are exchangeable and allow specific improvements of the whole framework. The realization of applications like forestry-, pedestrian-, wheelchair-, emergency- or bicycle routing is in most cases straight forward and the framework offers the functionality also for these domains.

Chapter 7

Conclusions

7.1 Summary

This work represents a workflow for processing agricultural telemetry data (cf. figure 2.8 and figure 3.8)). It starts with methods for preprocessing real world data in chapter 3, proceeds with new methods and algorithms for geographic data extraction of road- (chapter 4) and field geometries (chapter 5) and ends-up with the routing use-case in chapter 6. Within these parts a set of existing, extended and fundamentally new methods and algorithms for each research question have been analyzed. The conclusion of each chapter will be set in the broader context and connected with the research questions from section 1.3.

Each chapter provide methods and algorithms as input for services. The services are then integrated as part of the whole data processing chain within the service oriented architecture. The data preprocessing step is following principles of foregoing and ongoing research in the domain of movement analysis and data mining. The analysis of real object movement data from a commercial product comes along with challenges in terms of system evolution, data heterogeneity and data defects. These challenges that come with real world data have to be considered in the processing steps. Especially due to the fact that the output data of the used algorithms can be very sensitive on corrupt data. Many of the errors are only visible during the work with this data, which makes it even more complicated. Some of them are also not known beforehand and will be seen earliest during the processing or by checking the results.

In chapter 3 the data preprocessing is described. Several steps that handle the data errors and outliers are explained. The exclusion of obviously inaccurate measures, originated from GPS hops or the internal data storage process, the null values in the raw data originating from GPS hardware (e.g. when GPS hardware has a delay in positioning) is the main needed step for a clean data set. The subsequent data classification step to distinguish between different work states is then able to generate the needed data sets for the road- and field extraction algorithms in chapters 4 and 5. The first approaches to classify data have been done in Lauer et al. [2014] using a knn-classifier. The presented approach in chapter 3 extends the classes (e.g. with a headland class) and uses another classification approach (Random Forest). Using measures such as spatio-temporal density, speed and acceleration (difference) the presented classifier is able to detect infield and road data. The infield data can be further classified into headland and infield working. The classification accuracy is represented within the figures in subsection 3.5.3. Another advantage of the presented Random Forest approach is the integrated calculation of variable importance that shows the impact of each single variable on the classification. Three tiered

data classification demonstrates the impact of the machine type and the single machines on the classification process. It has been shown in chapter 3, that general classification is possible, but a finer treatment will improve the classification results. To handle each machine separately, it has to be guaranteed that a representative set of training data for each single machine is available. Additionally, a solution for the extraction of a mean working direction has been presented. While average and median method are affected by the field geometry and the headland processing, the histogram method reduced this influence and gives proper results which are used for infield routing graph generation. With these steps, this chapter answered the first question on classification of agricultural movement data and with the presented methods, the data preparation ensures the input data for the later chapters.

Chapter 5 makes use of the infield data. As methods for field boundary computation, following algorithms are developed, extended, parametrized and analyzed: α -shapes, a raster method and the blow-shrink algorithm. These methods were confirmed in a first step by visual analysis. A Jaccard-Index based quality assessment then enables the computational verification of the methods. The developed algorithm for working width detection gives proper results for the reference machines. However, a quality measure can only be made for this subset of machines due to the missing or not appropriate parameter settings by the machine driver for the other machines within the telemetry data subset (e.g. the working width is not necessarily equals to the attachment width). Nevertheless, it is expected that other machines will produce similar results. The working width parameter can then refine the blow shrink parametrization. The provided method enables the computation of up-to-date field geometries based on agricultural telemetry data and therefore answers the research question on the extraction of field geometries for field boundaries. Furthermore it gives the content for the field service component and the input data for infield routing algorithms.

Subsection 6.1.4 in chapter 6 presents different approaches for routing graph generation within polygonal geometries. Deriving the field access points from telemetry data using the intersection points of trajectories and field polygons gives a set of truly drivable gateways. In case of multiple gateways for a field boundary, gateway points that are spatially close to each other will be aggregated to a single gateway point. Besides common approaches for routing graphs within polygons, an extended grid approach, the rotated diamond grid approach is presented. This new approach uses the mean working direction, which has been calculated by the algorithm from section 3.6 to rotate the grid. Additionally, 45° outgoing edges from the main lines connect these. This newly oriented diamond grid approach computes a routing graph that involves the requirements for agricultural routing. Using this algorithm, vehicles are guided mainly on working direction that prevents machine driver from inconvenient line crossings. Compared to other approaches which precompute graphs for open spaces, the just-in-time calculation of the infield route graph for the target field during each route request keeps the in memory data low. Due to the fact, that the conditions for drivable sections of a field underly a constant change (e.g. during the harvest) and a crossing of other fields than the destination field is not typical, the in time calculation and integration of the up-to-date infield graph fulfills the requirements for the agricultural domain. These findings and the resulting algorithms are the answers for the research question on infield graph generation.

Computing road geometry information to close the gap of an existing base data set gives a solution for the research question on road geometry generation. Chapter 4 evaluates algorithms for road network generation from GPS trajectories. For each presented

algorithm the influence of the parametrization for the agricultural telemetry input data has been given. Each of the presented approaches is able to provide a road network. The best results in case of the agricultural telemetry data could be achieved by using the Fréchet distance based approach. This algorithm is able to scale the presented use case. The threshold for the telemetry data is the only variable that has to be set. The estimation of this threshold is done by visual check. With the hereby derived threshold, the algorithm is able to provide a feasible road network geometry. This geometry is then used as a new data layer for visual comparison and data integration. The presented approaches show the possibilities but also the limits of an automatic extraction of a road network geometry using the agricultural telemetry data set.

The closing research question is treated in chapter 6. At the beginning, a description of the OpenStreetMap data structure and road attributes is given. The extraction of a routing ready road data set for agricultural vehicles has been given. Furthermore, a route service for agricultural vehicles based on OSM has been implemented as a demo case. The attribute diversity of OpenStreetMap as a geographic data set, generated by volunteers, provides verifiably feasible data set for routing the agricultural vehicles. With the aggregated velocities for each driven road attribute combination, extracted from the telemetry data, the route service shows route estimations close to reality.

7.2 Discussion of the Methodology

The thesis consists of a set of methods within a couple of different research areas. Within each chapter different methods to solve the initially defined research questions are analyzed. Although it has been given a state of the art of the different research areas and an overview of existing solutions, a subset of methods had to be selected for further investigation. Ancillary methods that have not been analyzed in this thesis are also worthy of study and can hold potential for further improvements. In this subsection, a summary of these not prioritized methods will be given with potential ways of improvement. Due to the fact, that this work has been realized during a longer period of time, there were upcoming new methods that could not be considered, but have potential for further improvements. Furthermore, the continuous development of the telemetry services, the data, the agricultural machinery and processes modify the requirements and raise new possibilities.

In section 3.2 a straight forward approach for map matching has been presented. For the given use case, the method provides quite reasonable results. The implementation of more sophisticated methods could improve the matching and increase the reliability (e.g. prevent wrong matching on access roads or parallel roads). Variants of Hidden-Markov-Model map matching done by Newson and Krumm [2009] and the additional integration of semantic information, such as road and vehicle attributes can improve map matching and subsequent analyzes. A post processing step to increase the positional accuracy of GPS positions could also help. However, this usually needs the raw GPS data including additional measures, such as number of used satellites and dilution of precision parameters (HDOP, VDOP). Because the number of agricultural vehicles with RTK steering systems is increasing, the positional accuracy of telemetry measurements for new integrated RTK enabled vehicles will improve in the future.

Regarding classification algorithms, mainly two methods have been used with the agricultural telemetry data. The results of the knn-algorithm are presented in Lauer et al. [2014]. The Random Forest algorithm is described in section 3. These methods and their results are transparent and the influences of the input parameters can be investigated

through integrated measures such as variable importance. Classification using Neural Networks with a larger set of attributes could potentially increase classification accuracy. An individual test and training set for each machine with the integration of further, machine specific attributes could be beneficial in case of classification. On the other hand, it needs an expensive effort to collect this training data. Additionally this separation can restrict a possible portability.

The methods to compute the field geometries from chapter 5 return proper results. However, the presented approaches make only use of the telemetry data set. Improvements are likely if a combination of further machine trajectories of different work processes (e.g. seeding, fertilizing) within one harvesting year on one field would be considered. With this, also the overlapping polygons, that are results from discontinuous working or multiple machinery on one field, can be aggregated. Further clump and sieve techniques can smooth the data, which eliminate outliers on the one hand. On the other hand, detection of small geometry holes that represent obstacles such as trees, power pylons, large rocks or wet and muddy places is more complicated. Concave hull functions with later infield refinement such as cutting holes could be another promising approach that has not been investigated in this section. The geometry extraction of the holes is likely the most complex part and needs a deep investigation. In this case, the accuracy assessment should be improved through a larger number of various reference fields to increase the robustness of the algorithms.

The extraction or refinement of a road data set using movement data of vehicles in chapter 4 has been improved within recent years. There exist several approaches to handle this question. Most of the algorithms are related to the input data and are therefore highly data sensitive. To process other kind of movement data, they need to be set up properly. Due to the fact that many of them are not publicly available or need a large investigation on the data preparation, the focus lies on a subset of algorithms, that are well described. Further algorithms such as the principal curve approach used in Wang et al. [2013] could improve results. However these methods need further investigation on the algorithmic and interface implementation to integrate them into the workflow. The set of raster based approaches have been limited to smaller data sets and due to the fact that they are using mainly higher point densities and smaller spatial extension, most of them are not applicable for the agricultural telemetry data in wide rural areas to extract usable results.

Most commercial application related developments have been made in the research field of route planning (cf. chapter 6). Route planning is not a new topic. Since (publicly) available road data exists, there has been a lot of work done by companies that provide market ready solutions for their customers. The algorithms that are used for the route planning demonstrator are mainly standard shortest path algorithms. The data has been optimized at the graph generation level, but there is still potential for optimization in the shortest path algorithm. During the thesis, the most promising approach developed is the Graphhopper project that provides a configurable open source component for route planning with OpenStreetMap data. The developments within the context of the MARS routing had also an impact on the OpenRouteService that provides a large number of vehicle configurations such as heavy vehicles or agricultural vehicles. At the time of the realization of the MARS prototype, the OpenLS interface has been one promising approach to provide an open and well documented interface. Although this interface has benefits in terms of structure and documentation, it has several limits that result in the provided extension. Furthermore, the XML implementation of such an interface is not up-to-date anymore and from the technologically point of view, a JSON implementation would be far more up-to-date. However, a simplified interface will not have a big influence on the

service and the algorithms itself but it will simplify the ability to connect further services.

The workflow is embedded in a component based software. The algorithmic results are integrated in the corresponding services which are accessible via standardized web service interfaces. From end user perspective, this allows a comprehensive view on the process from data acquisition, data cleaning and preparation, algorithmic data and knowledge extraction to end user services that use this data. From the developer perspective, each component is replaceable. This increases the flexibility and the independent replaceability of each component. The usage of standardized web service interfaces and the semantically structured components allows the integration of further algorithms to improve each component of the whole workflow separately. Each component also works as a stand alone process, if the needed input data is provided in the corresponding format.

7.3 Constraints of the presented Approach and resulting further Research Questions

The workflow and its single components show a reliable first approach on the analysis of agricultural telemetry data for the improvement of routing relevant geographic data. The data set used is larger than the example data sets that are used in most of the related research approaches. The presented approach demonstrates, that a relational database can handle this amount of data well. However, one should keep in mind that the used data is only a subset of the whole telemetry database and the size of movement data is growing rapidly. Intelligent storage of movement data from several thousand machines with a growing number of sensors and higher sampled measures will be an inevitable requirement for the near future. Storing this data and making it searchable is therefore a key requirement to get the required attributes used by a machine learning approach. Divide and conquer, parallelized algorithms and intelligent indexing are current strategies to manage this. Recent development on “Big Data” frameworks, such as Apache Spark¹, promises to manage these challenges in near future.

Since only movement parameters are used for classification, the presented algorithms are generic, in that they can easily be used for other telemetry systems, if their output can be provided in the correct structure for the used interface. A further matter is privacy. For many of the algorithms, the data needs to be stored in a central place. In case of privacy, the accessibility has therefore to be regulated and it has to be discussed which parts of the data can be accessible for whom. To ensure privacy, the data can be made only accessible for its provider. Using data of a set of providers can improve the overall results of the data mining algorithms. Especially the completeness of road networks that are used by many agricultural companies could then be improved. Such conditions should be clarified with the contractors and farmers who provide the data. If the system can be opened to other domains, it would be beneficial for each single domain. The more user and data providers will give data into the processes, the higher the likelihood to improve the resulting output data. For regions with sparse data, the integration of movement data from other domains could close the gap in case of road network generation. Privacy should therefore be discussed beforehand.

Methods and reference data to validate the generated data set could be extended. In case of field boundary generation, validation is done by visual comparison and algorithmically using the Jaccard distance. To realize this comparison, a set of ground truth

¹<http://spark.apache.org>, (accessed 2017/01/07)

field boundaries is needed. For future work, the reference data for the validation should be extended to have a larger variance and heterogeneity of field boundaries. As another approach, the integration of aerial imagery and the usage of land surface classification to extract the field boundaries from digital imagery and compare them with results from the extraction algorithms of the telemetry data should be discussed. Regarding the road network generation, a set of validation approaches could be established. On the one hand, a crowdsourced validation approach will be helpful. Recent methods such as MapSwipe² can be very helpful to identify new roads from aerial or satellite imagery. The local knowledge of the farmers can be collected by OpenStreetMap, where they are able to edit the spatial features. However, this will only be sufficient for technically oriented farmers. Hence, more usable and autonomous approaches should be developed. Utilities that are quite easy to use and do not need high level of initial training. Visual interpretation of imagery and fast feedback generation should be the main objectives. The data integration can be made in an easy way using the provided APIs. The usage of multiple data sources will be the key to build a good data set. The blending of knowledge from multiple sources will help to validate the data and improve its completeness.

The generation of the infield route graphs is based on the preexisting field geometry. Another approach, that is not further explained in this work could be the analysis of driven infield routes. From these driven geometries, an infield graph representing the most driven routes can be extracted. This can be done by using the methods of the route geometry generation from chapter 4. The implemented route service shows the interaction of each single component and provides an overview on the resulting data. The routing algorithm fit the requirements for agricultural routing and is able to handle multiple constraints. However, scalability has to be tested. For a practical usage on the field, a navigation system would benefit farmers more. This is out of scope for this work. To bring the functionality into the market, this step will be one of the most important objectives. With this, further research questions, such as visual presentation, real time data analysis and real time data manipulation will emerge. The revision of the presented methods to satisfy this requirement will also be a future research question.

Another gap that has not been filled in this thesis is the enrichment of attributes for the generated road geometries. The extraction of the road geometry is the first part that is needed for routing. The connection of roads is needed for building the topology that represents the graph. The last part is then the enrichment of the geometries with attributes. This will then be part of the edge weight that is used by most of the routing algorithms to compute the fastest, shortest, most economic or scenic route. The extraction of these attributes is a broad field of research. Some of the attributes can be extracted directly from the static machine data, such as minimum width, minimum clear height and minimum loading capacity. Road type and road surface, which are useful attributes for routing, have to be extracted with other methods. Lauer et al. [2011, 2013] used smartphone sensors to learn road surface attributes. In John et al. [2016], they extracted slope and elevation of roads from GPS trajectories. At this time, the used data sets of the agricultural telemetry system did not provide the temporal data density and the accelerometer data to extract road surface attributes. However, the set of sensors and the data on the machines is still growing. Furthermore, nearly all drivers have a smartphone and a tablet PC on board. These devices can also be used to collect the needed data and enriching the extendable workflow.

²<http://www.mapswipe.org>, (accessed 2016/07/31)

The data that has been used is located in mid Europe. Therefore, an extension to the whole world will be another next step. From data perspective, this step is more than obvious. The telemetry system that is used here has clients from all over the world and consists therefore of machinery data from every continent. The different cultures can cause diverse movement patterns. Especially rice fields or irrigated farmland will show other, completely different driving patterns, than the typical European field structure. The size of machinery differs accordingly to the size of the fields and the field structure. These constraints have to be considered when extending the workflow to new regions.

7.4 Transferability on other Domains

This work is focused on the domain of agriculture. The input data from an agricultural telemetry system, the data classification, the created field boundaries and road networks for agricultural vehicles and the routing is mainly related to the challenges within the agricultural domain. However, the extracted data and most of the algorithms can also be used by other domains. The generated road network can be used by a very broad user group. A detailed digital road network with low level roads, such as tracks can be interesting for bicyclists or hikers. Furthermore the domain of forestry can also profit from a more detailed road network. Emergency and disaster response scenarios are further use cases that can be improved by enriched data. Also for winter services, such as snow clearing, a proper digital road network is very useful. A precise road geometry is essential for the snow clearing, when the snow is quite high and the road boundary is not visible.

The computed field boundaries can be used to verify land use data, in case there is no conflict with the farmers privacy (field boundaries are part of the operational data of a farm business). The organization of work processes and the process land consolidation can be improved with new and more accurate field geometries. The new sight on field boundaries and the real driven field gateways can be used for planning and organization of the rural road network. With the knowledge about frequently used roads, their roadways can be improved. Especially with the growing agricultural heavy goods traffic and the large sized vehicles a selective upgrading of the rural road network is needed [Ebke et al., 2012].

The presented methods for infield route graph generation can be used for further applications in the field of indoor routing or routing within open spaces, such as market places or public squares. Due to the fact that these domains have other preferences on the routes, the graph generation method that fits to the use-case has to be chosen. The analysis of available GPS trajectories of pedestrians with the presented methods can be helpful to generate the route graphs and the geographic features.

The extraction of spatiotemporal attributes can improve the classification of movement trajectories from other domains. In case of pedestrian data, the detection of squares such as market places or open space in front of a building could be realized by the classification of the single measurements. Extracting the geometries of these places is more complicated due to the fact, that the pedestrian movement is rather arbitrarily and shows other patterns than the more structured infield trajectories of agricultural vehicles.

7.5 Outlook

With the pervasiveness of agricultural telemetry systems, the requirement for connecting services of different manufacturers will increase. The linking of agricultural machinery

is ongoing and projects such as marion³ show the future of agricultural machinery and information technology. Activities to provide telemetry data via standardized service infrastructure will further homogenize data structures and processes. Furthermore, companies such as 365farmnet⁴ clearly show that the future way of farming will be more cooperative. Linked (open) data and the design of easy accessible interfaces will be the key forces to pave the way to cooperativeness. Driven by this up-to-date technology, the collaboration of machinery companies with other involved industries and farmers will be strengthened. This will set further requirements on handling big data. Data mining and knowledge extraction will gain further importance and the impact on productivity of new and powerful methods will increase.

Due to the fact that parts of the extracted geodata are also beneficial for other domains, the possible collaboration is not limited to the agricultural domain. Data collaborations with sports tracking companies, whose users are also using agricultural relevant roads (e.g. cycling or running) can be established to dense the data with their GPS trajectories. With a larger amount of data providers, also low traffic roads could be detected. The methodology for geographic feature extraction is also applicable in other use cases, as already mentioned. Especially the integration of a set of knowledge sources and the connection of services from movement data, web crawled and crowdsourced data as well as static and dynamic sensor data promise broad changes and improvements for location based services and processes.

The technical hurdles will be tackled, as latest developments in “Big Data”-research show. Social, ethical and therefore privacy and legal challenges have to be discussed and solutions should be found as soon as possible, or latest when the technology reached the broad society. With this, an overall improvement can be achieved through technology and social awareness of positive and less positive consequences of location based technology and geographic data.

³<http://www.projekt-marion.de>, (accessed 2016/07/31)

⁴<https://www.365farmnet.com/>, (accessed 2016/07/31)

List of Figures

| | | |
|------|---|----|
| 1.1 | Total population by major area (source: United Nations, 2015, http://www.claas-mediadatabase.com , (accessed 2016/07/16)) | 2 |
| 1.2 | Cabin of a CLAAS AXION 900 (source: CLAAS - http://www.claas-mediadatabase.com) | 4 |
| 2.1 | Agricultural Telematic Systems CLAAS Telematics (source: https://www.claas.com), AGCommand (source: http://www.agcototechnologies.com) and JDLink (source: http://www.deere.com) | 10 |
| 2.2 | Architecture CLAAS Telematics (source: https://www.claas.com) | 11 |
| 2.3 | Overview on spatial distribution of the data set (machine types: forage harvester, combine harvester, tractor and tractors from other manufacturers, are grouped by colors) (Datasource basemap: ©OpenStreetMap contributors) | 12 |
| 2.4 | Measurement Values for Machines | 13 |
| 2.5 | OSM Data Model (source: adapted from Ramm and Topf [2010]) | 14 |
| 2.6 | OSM user statistics (source: http://wiki.openstreetmap.org/wiki/Stats) | 15 |
| 2.7 | OSM data base statistics (source: http://wiki.openstreetmap.org/wiki/Stats) | 15 |
| 2.8 | Architecture TeleAgro+ | 18 |
| 3.1 | Error - Alternating positions (Datasource basemap (orthophoto): Bayerische Vermessungsverwaltung – www.geodaten.bayern.de) | 22 |
| 3.2 | Mapmatching process | 24 |
| 3.3 | Mapmatching statistics | 25 |
| 3.4 | Matched path vs. primary | 26 |
| 3.5 | Matched path | 26 |
| 3.6 | Average speeds per Highwaytype | 27 |
| 3.7 | Working Mode for Jaguar 950 forage harvester | 29 |
| 3.8 | Workflow - Dataprocessing workflow with algorithms for road and field computation | 30 |
| 3.9 | Ground truth distribution for all classes | 31 |
| 3.10 | randomForest results - all machines | 33 |
| 3.11 | randomForest results - Axion 900 | 34 |
| 3.12 | randomForest results - Jaguar 950 | 34 |
| 3.13 | randomForest results - Jaguar 970 | 34 |
| 3.14 | randomForest results - Lexion 600 | 35 |

| | | |
|------|---|----|
| 3.15 | randomForest results - Lexion 770 | 35 |
| 3.16 | randomForest results - Traktor II Universal | 35 |
| 3.17 | randomForest results - Traktor II Universal | 36 |
| 3.18 | randomForest results - 119796403 - Traktor II Universal | 37 |
| 3.19 | randomForest results - 119796404 - Traktor II Universal | 37 |
| 3.20 | randomForest results - 119796405 - Traktor II Universal | 37 |
| 3.21 | randomForest results - 119802463 - Lexion 600 | 38 |
| 3.22 | randomForest results - 119802464 - Lexion 600 | 38 |
| 3.23 | randomForest results - 119802465 - Lexion 600 | 38 |
| 3.24 | randomForest results - 119834787 - Traktor II Universal | 39 |
| 3.25 | randomForest results - 119834788 - Traktor II Universal | 39 |
| 3.26 | randomForest results - 119931748 - Traktor III Universal | 39 |
| 3.27 | randomForest results - 119992348 - Jaguar 950 | 40 |
| 3.28 | randomForest results - 120018706 - Jaguar 950 | 40 |
| 3.29 | randomForest results - 120044864 - Lexion 770 | 40 |
| 3.30 | randomForest results - 120104352 - Jaguar 970 | 41 |
| 3.31 | randomForest results - 120171079 - Axion 900 | 41 |
| 3.32 | randomForest results - 120171080 - Axion 900 | 41 |
| 3.33 | randomForest results - 120171081 - Axion 900 | 42 |
| 3.34 | Classified measurements before and after sieve filtering | 44 |
| 3.35 | Sample tracks and their track segments | 45 |
| 3.36 | Main Working Direction (red arrow) for rectangular fields and for fields with a more complex boundary | 46 |
| 4.1 | Example of traces with cluster seeds (source: Edelkamp and Schrödl [2003]) | 55 |
| 4.2 | Merged cluster segments (thick black lines) (source: Edelkamp and Schrödl [2003]) | 56 |
| 4.3 | Traces, segments, and intersection contour model (dotted) (source: Edelkamp and Schrödl [2003]) | 56 |
| 4.4 | Forces on measurement points (source: Cao and Krumm [2009]) | 57 |
| 4.5 | Graph generation algorithm (source: Cao and Krumm [2009]) - (a) The input, in terms of 3 trips. (b) The graph G after processing trip 1. (c) The graph G after processing trip 2. (d) The graph G after processing trip 3 . . | 57 |
| 4.6 | Graph reduction - from trajectories to final graph (Datasource basemap: ©OpenStreetMap contributors) | 58 |
| 4.7 | Parallel reduction (changed after source: Morris et al. [2004]) | 59 |
| 4.8 | Face reduction (changed after source: Morris et al. [2004]) | 60 |
| 4.9 | Serial reduction - converting nodes to shape points | 60 |
| 4.10 | Graph merging - integration of a new trajectory (sketches modified from presentation of Ahmed and Wenk [2012]) | 62 |
| 4.11 | Parametrization of Ahmed and Wenk - Road Construction (a) and b) are the upper figures, c) and d) the lower ones) | 64 |
| 4.12 | Problems with curve structures and graph generation through low point density | 65 |
| 4.13 | Comparison of Map Generation Algorithms in Baden-Wuerttemberg (exam- ple: rectangular road network): trajectories, Morris et al. [2004], Ahmed et al. [2014], Edelkamp and Schrödl [2003] and Cao and Krumm [2009] . . . | 66 |

| | | |
|------|--|-----|
| 4.14 | Comparison of Map Generation Algorithms in Baden-Wuerttemberg (example: crossing roads): trajectories, Morris et al. [2004], Ahmed et al. [2014], Edelkamp and Schrödl [2003] and Cao and Krumm [2009] | 67 |
| 5.1 | Google and land survey administration Bavaria aerial imagery, DGPS points and field boundary | 74 |
| 5.2 | Reference fields (Datasource basemap: ©OpenStreetMap contributors) | 74 |
| 5.3 | Field Boundary - α -shape vs. Blow-Shrink | 76 |
| 5.4 | Field boundary computation - raster based approach (source: Kortenbruck and Griepentrog [2014]) | 76 |
| 5.5 | Field boundary computation for TeleAgro+ data - raster based approach (map data: ©OpenStreetMap contributors) | 77 |
| 5.6 | Speed statistics for infield and headland | 78 |
| 5.7 | Field boundary Blow-Shrink - Comparison measurement based vs. track-segment based | 79 |
| 5.8 | Computing the working width for one field (1. estimate mean working direction, 2. build orthogonal line string that is placed in field center and cut by outer trajectories, 3. compute intersection points and divide line string length by #intersection points) | 81 |
| 5.9 | Estimated working width for Jaguar 970 forage harvester and Lexion 770 combine harvester | 82 |
| 5.10 | Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field_01 | 84 |
| 5.11 | Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field_02 | 85 |
| 5.12 | Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field_03 | 85 |
| 5.13 | Field boundary computation for TeleAgro+ data - α -shape, blow-shrink and grid based approach for reference field_04 | 86 |
| 5.14 | Comparison of the different approaches using the Jaccard Index | 88 |
| 5.15 | Field gateways filtering | 90 |
| 5.16 | Field request in the TeleAgro+ GUI | 92 |
| 6.1 | Infield routing using a grid graph | 105 |
| 6.2 | Infield routing using a diamond graph | 106 |
| 6.3 | Infield routing using a Delaunay graph | 106 |
| 6.4 | Infield routing using a Voronoi graph | 107 |
| 6.5 | Infield routing using a visibility graph | 107 |
| 6.6 | Infield routing using a straight skeleton graph | 108 |
| 6.7 | Service Architecture: Route-Service, Field-Service and Vehicle-Service | 109 |
| 6.8 | Comparison between calculated routes (ORS, MARS) and real driven trajectory - random order | 113 |
| 6.9 | Comparison between real driving time and the predicted time from the route services (ORS and MARS) - random order | 113 |
| 6.10 | Different routes (left) and lost way v.s. correct calculation (right) (map data: ©OpenStreetMap contributors) | 114 |
| 6.11 | Driven route v.s. OpenRouteService v.s. MARS (map data: ©OpenStreetMap contributors) | 114 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Analyzed machine types from CLAAS Telematics (source of imagery: https://claas-telematics.com/) | 20 |
| 3.1 | Error classification - Telemetry data | 23 |
| 3.2 | Formula for extended attribute calculation | 28 |
| 4.1 | Algorithms categories (source: Ahmed et al. [2014]) | 52 |
| 4.2 | Algorithms categories (source: adapted from Biagioni and Eriksson [2012]) . | 54 |
| 5.1 | Jaccard index - pros and cons | 87 |
| 5.2 | Field boundary algorithms - pros and cons | 88 |
| 6.1 | Used OpenStreetMap tags | 99 |
| 6.2 | Memory Consumption of compressed and uncompressed Graph | 100 |
| 6.3 | Add caption | 101 |
| 6.4 | Graph size for Test Regions (data source OpenStreetMap: Heidelberg, Baden-Württemberg: 11-2013, Study site 02-2014) | 102 |
| 6.5 | Average speeds for <i>Jaguar 950</i> forage harvester (table reduced – some sub class values are omitted) | 103 |
| 6.6 | Comparison of infield routing graph computation methods | 110 |

Bibliography

- T. Abeel, Y. Van de Peer, Y. Saeys, Y. de Peer, and Y. Saeys. Java-ML: A machine learning library. *The Journal of Machine Learning Research*, 10:931–934, dec 2009. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1577069.1577103>.
- G. Agamennoni, J. I. Nieto, and E. M. Nebot. Robust Inference of Principal Road Paths for Intelligent Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):298–308, mar 2011. ISSN 1524-9050. doi: 10.1109/TITS.2010.2069097. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5567159>.
- M. Ahmed and C. Wenk. Constructing street networks from GPS trajectories. In L. Epstein and P. Ferragina, editors, *ESA'12 Proceedings of the 20th Annual European conference on Algorithms*, volume 7501 of *Lecture Notes in Computer Science*, pages 60–71, Berlin, Heidelberg, sep 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33089-6. doi: 10.1007/978-3-642-33090-2. URL <http://dl.acm.org/citation.cfm?id=2404160.2404167>.
- M. Ahmed, S. Karagiorgou, D. Pfoser, and C. Wenk. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica*, 19(3):601–632, dec 2014. ISSN 1384-6175. doi: 10.1007/s10707-014-0222-6. URL <http://link.springer.com/10.1007/s10707-014-0222-6>.
- M. Ahmed, B. T. Fasy, M. Gibson, and C. Wenk. Choosing thresholds for density-based map construction algorithms. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '15*, pages 1–10, New York, New York, USA, 2015. ACM Press. ISBN 9781450339674. doi: 10.1145/2820783.2820810. URL <http://dl.acm.org/citation.cfm?doid=2820783.2820810>.
- O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A novel type of skeleton for polygons. *Journal of Universal Computer Science*, 1(12):752–761, 1995. doi: 10.3217/jucs-001-12-0752. URL http://link.springer.com/chapter/10.1007/978-3-642-80350-5_{_}65.
- O. Ali, B. Verlinden, and D. Van Oudheusden. Infield logistics planning for crop-harvesting operations. *Engineering Optimization*, 41(2):183–197, feb 2009. ISSN 0305-215X. doi: 10.1080/03052150802406540. URL <http://dx.doi.org/10.1080/03052150802406540> <https://hal.archives-ouvertes.fr/hal-00545360>.
- H. Alt and M. Godau. COMPUTING THE FRÉCHET DISTANCE BETWEEN TWO POLYGONAL CURVES. *International Journal of Computational Geometry & Applications*, 05(01n02):75–91, mar 1995. ISSN 0218-1959. doi: 10.1142/S0218195995000064. URL <http://www.worldscientific.com/doi/abs/10.1142/S0218195995000064>.

- H. Alt and L. Guibas. Discrete Geometric Shapes: Matching, Interpolation, and Approximation. *Handbook of computational geometry*, pages 1–34, 1999. doi: DOI:10.1016/B978-044482537-7/50004-8. URL <http://books.google.com/books?hl=en&lr=&id=uZdAqAWB3BcC&oi=fnd&pg=PA121&dq=Discrete+Geometric+Shapes:+Matching,+Interpolation,+and+Approximation&ots=lEx{ }DDwLMF&sig=xPC67HoYTYvLgSKz-t-E7XBJUbg>.
- N. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. ISSN 0003-1305. doi: 10.1080/00031305.1992.10475879.
- S. Andres. *Implementierung und Kosten-Nutzen-Analyse automatischer Datenerfassungssysteme in russischen Agrarholdings*. PhD thesis, 2009.
- J. J. Arsanjani, C. Barron, M. Bakillah, and M. Helbich. Assessing the Quality of Open-StreetMap Contributors together with their Contributions. In *Proceedings of the AGILE*, 2013. ISBN 978-3-319-00615-4.
- C. Barron and P. Neis. iOSMAlyzer - ein Werkzeug für intrinsische OSM Qualitätsuntersuchungen. In *AGIT 2013, Symposium Angewandte Geoinformatik, Salzburg*, pages 1–10, Salzburg, Austria, 2013.
- N. Bartelme. *GIS Technologie*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989. ISBN 978-3-540-50410-8. doi: 10.1007/978-3-662-07494-7. URL <http://link.springer.com/10.1007/978-3-662-07494-7>.
- D. Baum and M. Rothmund. HORSCH Telemetriesystem. In *IT-Standards in der Agrar- und Ernährungswirtschaft - Referate der 34. GIL Jahrestagung*, pages 1–4, Bonn, 2014. Gesellschaft für Informatik.
- J. Bentley and T. Ottmann. Algorithms for Reporting and Counting Geometric Intersections. *IEEE Transactions on Computers*, C-28(9):643–647, sep 1979. ISSN 0018-9340. doi: 10.1109/TC.1979.1675432. URL <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1675432>.
- M. Berger, P. Alliez, A. Tagliasacchi, L. M. Seversky, C. T. Silva, J. A. Levine, and A. Sharf. State of the Art in Surface Reconstruction from Point Clouds. *Eurographics STAR (Proc. of EG'14)*, 2014.
- J. Biagioni and J. Eriksson. Inferring Road Maps from Global Positioning System Traces. *Transportation Research Record: Journal of the Transportation Research Board*, 2291(-1):61–71, dec 2012. ISSN 0361-1981. doi: 10.3141/2291-08. URL <http://trb.metapress.com/openurl.asp?genre=article&id=doi:10.3141/2291-08>.
- F. Biljecki, H. Ledoux, and P. van Oosterom. Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, 27(2):385–407, feb 2013. ISSN 1365-8816. doi: 10.1080/13658816.2012.692791. URL <http://www.tandfonline.com/doi/abs/10.1080/13658816.2012.692791>.
- BMVI. *Transport telematics : developments and success stories in Germany*. Federal Ministry of Transport Building and Housing, Berlin, 2004. URL <http://www.worldcat.org/title/transport-telematics-developments-and-success-stories-in-germany/oclc/058734762>.

- D. D. Bochtis, C. G. Sørensen, and S. G. Vougioukas. Path planning for in-field navigation-aiding of service units. *Computers and Electronics in Agriculture*, 74:80–90, 2010. ISSN 01681699. doi: 10.1016/j.compag.2010.06.008.
- A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *Computers, Environment and Urban Systems*, 36(6):526–537, nov 2012. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2012.06.001. URL <http://www.sciencedirect.com/science/article/pii/S0198971512000543>.
- L. G. Bont. *Spatially explicit optimization of forest harvest and transportation system layout under steep slope conditions*. PhD thesis, ETH Zuerich, 2012.
- L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL <http://link.springer.com/article/10.1023/A%7D53A1010933404324>.
- A. Bröring, C. Stasch, J. Echterhoff, A. B. North, and C. S. Ifgi. Open Geospatial Consortium OGC ® Sensor Observation Service Interface Standard. 2012. URL <http://www.opengeospatial.org/standards/sos>.
- A. Bröring, A. Remke, C. Stasch, C. Autermann, M. Rieke, and J. Möllers. enviroCar: A Citizen Science Platform for Analyzing and Mapping Crowd-Sourced Car Sensor Data. *Transactions in GIS*, 19(3):362–376, jun 2015. ISSN 13611682. doi: 10.1111/tgis.12155. URL <http://doi.wiley.com/10.1111/tgis.12155>.
- K. Brundell-freij and E. Ericsson. Influence of street characteristics, driver category and car performance on urban driving patterns. *Technology*, 10:213–229, 2005. doi: 10.1016/j.trd.2005.01.001.
- R. Brüntrup, S. Edelkamp, S. Jabbar, and B. Scholz. Incremental map generation with GPS traces. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pages 413–418. IEEE, 2005. ISBN 0-7803-9215-9. doi: 10.1109/ITSC.2005.1520084. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1520084>.
- K. Buchin, M. Buchin, and Y. Wang. Exact algorithms for partial curve matching via the Fréchet distance. pages 645–654, jan 2009. URL <http://dl.acm.org/citation.cfm?id=1496770.1496841>.
- Bundesgesetzblatt. Verordnung über die Durchführung von Stützungsregelungen und gemeinsamen Regeln für Direktzahlungen nach der Verordnung (EG) Nr. 1782/2003 im Rahmen des Integrierten Verwaltungs- und Kontrollsystems sowie zur Änderung der Kartoffelstärkeprämienverordnung, 2004. URL <http://www.bgbl.de/xaver/bgbl/media/44CA9D57CD32B3FCB80ECF7371240DAD/bgbl104s3194{ }61090.pdf>.
- R. Canavosio-Zuzelski, P. Agouris, and P. Doucette. A Photogrammetric Approach for Assessing Positional Accuracy of OpenStreetMap© Roads. *ISPRS International Journal of Geo-Information*, 2(2):276–301, apr 2013. ISSN 2220-9964. doi: 10.3390/ijgi2020276. URL <http://www.mdpi.com/2220-9964/2/2/276/htm>.
- L. Cao and J. Krumm. From GPS traces to a routable road map. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information*

- Systems - GIS '09*, page 3, New York, New York, USA, 2009. ACM Press. ISBN 9781605586496. doi: 10.1145/1653771.1653776. URL <http://portal.acm.org/citation.cfm?doid=1653771.1653776>.
- I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29(1):33–51, jan 2002. ISSN 03050548. doi: 10.1016/S0305-0548(00)00056-3. URL [http://dx.doi.org/10.1016/S0305-0548\(00\)00056-3](http://dx.doi.org/10.1016/S0305-0548(00)00056-3).
- F. Chazal, D. Chen, L. Guibas, X. Jiang, and C. Sommer. Data-driven trajectory smoothing. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11*, page 251, New York, New York, USA, nov 2011. ACM Press. ISBN 9781450310314. doi: 10.1145/2093973.2094007. URL <http://dl.acm.org/citation.cfm?id=2093973.2094007>.
- C. Chen and Y. Cheng. Roads Digital Map Generation with Multi-track GPS Data. In *2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing*, volume 1, pages 508–511. IEEE, dec 2008. ISBN 978-0-7695-3563-0. doi: 10.1109/ETTandGRS.2008.70. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5070207>.
- D. Chen, L. J. Guibas, J. Hershberger, and J. Sun. Road Network Reconstruction for Organizing Paths. *Science*, pages 1309–1320, jan 2010. URL <http://dl.acm.org/citation.cfm?id=1873601.1873706>.
- H. M. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of robot motion : theory, algorithms, and implementation*. MIT Press, 2005. ISBN 9780262033275.
- B. Ciepluch, P. Mooney, and A. C. Winstanley. Building Generic Quality Indicators for OpenStreetMap. In *19th annual GIS Research UK (GISRUK)*, Portsmouth, 2011.
- N. Coffey. Memory usage of Java Strings and string-related objects, 2011. URL http://www.javamex.com/tutorials/memory/string_{_}memory_{_}usage.shtml.
- Council of European Union. Council regulation ({EU}) no 73/2009, 2009. URL <http://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:32009R0073>.
- J. Davies, a.R. Beresford, and a. Hopper. Scalable, Distributed, Real-Time Map Generation. *IEEE Pervasive Computing*, 5(4):47–54, oct 2006. ISSN 1536-1268. doi: 10.1109/MPRV.2006.83. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1717365>.
- M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*, volume 17. 2008. ISBN 3540779736. doi: 10.2307/3620533. URL <http://www.google.com/books?id=tkyG8W2163YC>.
- B. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, (6):793 – 800, 1934.
- E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271, dec 1959. ISSN 0029-599X. doi: 10.1007/BF01386390. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.165.7577>.

- S. Dodge, R. Weibel, and E. Forootan. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33(6):419–434, nov 2009. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2009.07.008. URL <http://www.sciencedirect.com/science/article/pii/S0198971509000556>.
- M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997. ISSN 1089778X. doi: 10.1109/4235.585892.
- D. H. Douglas and T. K. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, oct 1973. ISSN 0317-7173. doi: 10.3138/FM57-6770-U75U-7727. URL <http://www.utpjournals.press/doi/abs/10.3138/FM57-6770-U75U-7727>.
- D. H. Douglas and T. K. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. In *Classics in Cartography: Reflections on Influential Articles from Cartographica*, pages 15–28. 2011. ISBN 9780470681749. doi: 10.1002/9780470669488.ch2.
- T. D’Roza and G. Bilchev. An Overview of Location-Based Services. *BT Technology Journal*, 21(1):20–27, 2003. ISSN 1573-1995. doi: 10.1023/A:1022491825047. URL <http://link.springer.com/article/10.1023/A:1022491825047>.
- M. Duckham, L. Kulik, M. Worboys, and A. Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224–3236, oct 2008. ISSN 00313203. doi: 10.1016/j.patcog.2008.03.023. URL <http://www.sciencedirect.com/science/article/pii/S0031320308001180>.
- K. Ebke, T. Mitschang, and J. Omert, editors. *Wege in die Zukunft!? Neue Anforderungen an ländliche Infrastrukturen*, volume 9. Vorstand der Deutschen Landeskulturgeellschaft, Müncheberg, 2012. ISBN 1614-5240.
- S. Edelkamp and S. Schrödl. Route planning and map inference with global positioning traces. pages 128–151, jan 2003. URL <http://dl.acm.org/citation.cfm?id=865449.865460>.
- 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, jul 1983. ISSN 0018-9448. doi: 10.1109/TIT.1983.1056714. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1056714>.
- B. Eksioglu, A. V. Vural, and A. Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, nov 2009. ISSN 03608352. doi: 10.1016/j.cie.2009.05.009. URL <http://dx.doi.org/10.1016/j.cie.2009.05.009>.
- M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996. ISBN 1577350049. doi: 10.1.1.71.1980. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.2930>.

- Eu. Directive 2007/2/EC of the European Parliament and of the council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). *Official Journal of the European Union*, 50(January 2006):1–14, 2007. ISSN 17252555. URL <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF>.
- European Commission EC - Directorate-General for Mobility and Transport. *Road Transport - A change of Gear*. Belgium, 2012. ISBN 978-92-79-22827-8. doi: 10.2832/65952. URL http://ec.europa.eu/transport/road/doc/broch-road-transport{_.}en.pdf.
- Exelis VIS. ENVI software documentation, 2015. URL <http://www.exelisvis.com/docs/SievingClasses.html>.
- J. Fawcett and P. Robinson. Adaptive Routing for Road Traffic. *IEEE Computer Graphics and Applications*, 20(3):46–53, 2000. ISSN 02721716. doi: 10.1109/38.844372. URL http://ieeexplore.ieee.org/xpls/abs{_.}all.jsp?arnumber=844372.
- P. Felkel and S. Obdrzalek. Straight Skeleton Implementation. In *Proceedings of Spring Conference on Computer Graphics*, Budmerice, Slovakia, 1998. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.7175>.
- M. Fischler, J. Tenenbaum, and H. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15(3):201–223, mar 1981. ISSN 0146664X. doi: 10.1016/0146-664X(81)90056-3. URL <http://www.sciencedirect.com/science/article/pii/0146664X81900563>.
- M.-F. A. Fortier, D. Ziou, C. Armenakis, and S. Wang. Automated Correction and Updating of Road Databases from High-Resolution Imagery. *Canadian Journal of Remote Sensing*, 27(1):76–89, jul 2014. ISSN 0703-8992. doi: 10.1080/07038992.2001.10854922. URL <http://www.tandfonline.com/doi/abs/10.1080/07038992.2001.10854922>.
- J. Fu and H. H. Hochmair. Web Based Bicycle Trip Planning for Broward County , Florida. In *Esri International User Conference*, pages 1–12, San Diego, CA, 2009.
- R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. GreenGPS. In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, page 151, New York, New York, USA, jun 2010. ACM Press. ISBN 9781605589855. doi: 10.1145/1814433.1814450. URL <http://dl.acm.org/citation.cfm?id=1814433.1814450>.
- X. Ge, I. I. Safa, M. Belkin, and Y. Wang. Data skeletonization via Reeb graphs. In *Advances in Neural Information Processing Systems*, pages 837–845, 2011.
- R. Geisberger, P. Sanders, D. Schultes, and D. Delling. Contraction hierarchies: faster and simpler hierarchical routing in road networks. pages 319–333, may 2008. URL <http://dl.acm.org/citation.cfm?id=1788888.1788912>.
- J.-F. Girres and G. Touya. Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS*, 14(4):435–459, aug 2010. ISSN 13611682. doi: 10.1111/j.1467-9671.2010.01203.x. URL <http://doi.wiley.com/10.1111/j.1467-9671.2010.01203.x>.

- M. Goetz. Using Crowdsourced Indoor Geodata for the Creation of a Three-Dimensional Indoor Routing Web Application. *Future Internet*, 4(4):575–591, jun 2012. ISSN 1999-5903. doi: 10.3390/fi4020575. URL <http://www.mdpi.com/1999-5903/4/2/575/htm>.
- M. Goetz, J. Lauer, and M. Auer. An Algorithm Based Methodology for the Creation of a Regularly Updated Global Online Map Derived From Volunteered Geographic Information. In *The Fourth International Conference on Advanced Geographic Information Systems, Applications and Services*, number c, pages 50–58, 2012. ISBN 9781612081786.
- M. F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, nov 2007. ISSN 0343-2521. doi: 10.1007/s10708-007-9111-y. URL <http://link.springer.com/10.1007/s10708-007-9111-y>.
- A. Gore. The Digital Earth: Understanding our planet in the 21st Century. *Australian surveyor*, 43(2):89–91, 1998. ISSN 0005-0326. doi: 10.1080/00050326.1998.10441850.
- F. Graf, M. Renz, H.-p. Kriegel, and M. Schubert. MARiO: Multi Attribute Routing in Open Street Map. In *Proceedings of the 12th International Symposium on Spatial and Temporal Databases (SSTD)*, Minneapolis, MN, USA, 2011.
- A. Graser. Integrating Open Spaces into OpenStreetMap Routing Graphs for Realistic Crossing Behaviour in Pedestrian Navigation. In *GI Forum - Journal for Geographic Information Science*, volume 1, pages 12–29, Salzburg, Austria, 2016. doi: 10.1553/giscience2016.
- A. Graser, M. Straub, and M. Dragaschnig. Towards an Open Source Analysis Toolbox for Street Network Comparison: Indicators, Tools and Results of a Comparison of OSM and the Official Austrian Reference Graph. *Transactions in GIS*, pages n/a–n/a, oct 2013. ISSN 13611682. doi: 10.1111/tgis.12061. URL <http://doi.wiley.com/10.1111/tgis.12061>.
- R. D. Grisso, P. J. Jasa, and D. E. Rolofson. Analysis of traffic patterns and yield monitor data for field efficiency determination. *Applied Engineering in Agriculture*, 18(2):171–178, 2002.
- T. Guo, K. Iwamura, and M. Koga. Towards high accuracy road maps generation from massive GPS Traces data. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 667–670. IEEE, 2007. ISBN 978-1-4244-1211-2. doi: 10.1109/IGARSS.2007.4422884. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4422884>.
- A. Haghani and S. Jung. A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research*, 32(11):2959–2986, nov 2005. ISSN 03050548. doi: 10.1016/j.cor.2004.04.013. URL <http://dx.doi.org/10.1016/j.cor.2004.04.013>.
- S. Hahmann and D. Burghardt. How much information is geospatially referenced? Networks and cognition. *International Journal of Geographical Information Science*, 27(6):1171–1189, 2013. ISSN 1365-8816. doi: 10.1080/13658816.2012.743664. URL <http://www.tandfonline.com/doi/abs/10.1080/13658816.2012.743664>.
- M. Haklay. How good is OpenStreetMap information? A comparative study of OpenStreetMap and Ordnance Survey datasets for London and the rest of England. 2008. URL <http://www.citeulike.org/user/rfels/article/9929531>.

- M. Haklay. How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets. *Environment and Planning B: Planning and Design*, 37(4):682–703, 2010. ISSN 02658135. doi: 10.1068/b35097.
- I. Hameed, D. Bochtis, C. Sørensen, and M. Nøremark. Automated generation of guidance lines for operational field planning. *Biosystems Engineering*, 107(4):294–306, dec 2010. ISSN 15375110. doi: 10.1016/j.biosystemseng.2010.09.001. URL <http://linkinghub.elsevier.com/retrieve/pii/S1537511010001947>.
- F. Hartmann. Openmtbmap.org – Mountainbike and Hiking Maps based on Openstreetmap, 2013. URL <http://openmtbmap.org/>.
- A. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1990. ISBN 0521321964.
- T. Hastie and W. Stuetzle. Principal Curves. *Journal of the American Statistical Association*, 84(406):502–516, mar 1989. URL <http://amstat.tandfonline.com/doi/abs/10.1080/01621459.1989.10478797#.U2D1yldwUtI>.
- C. Häubl, C. Wimmer, and H. Mössenböck. Optimized strings for the Java HotSpot#8482; virtual machine. In *Proceedings of the 6th international symposium on Principles and practice of programming in Java - PPPJ '08*, page 105, New York, New York, USA, sep 2008. ACM Press. ISBN 9781605582238. doi: 10.1145/1411732.1411747. URL <http://dl.acm.org/citation.cfm?id=1411732.1411747>.
- C. Heipke. Crowdsourcing geospatial data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):550–557, nov 2010. ISSN 09242716. doi: 10.1016/j.isprsjprs.2010.06.005. URL <http://linkinghub.elsevier.com/retrieve/pii/S0924271610000602>.
- V. J. Heizinger. *Algorithmische Analyse von Prozessketten in der Agrarlogistik*. PhD thesis, TU München, 2014.
- J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka. Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4144–4157, dec 2007. ISSN 0196-2892. doi: 10.1109/TGRS.2007.906107. URL <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4378557>.
- C. Isert, D. Gusenbauer, and K. Klasing. Verfahren zum Verbessern der Positionierung eines Fußgängers unter Verarbeitung einer Fußgänger-Route, nov 2013. URL <http://www.google.com/patents/DE102012207610A1?cl=de&hl=de>.
- P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2): 37–50, 1912. ISSN 1469-8137. doi: 10.1111/j.1469-8137.1912.tb05611.x. URL <http://onlinelibrary.wiley.com/doi/10.1111/j.1469-8137.1912.tb05611.x/abstract%5Cnpapers2://publication/uuid/14166BB3-DA8C-4A1E-8885-2297ABCAE024>.
- S. Jang, T. Kim, and E. Lee. Map generation system with lightweight GPS trace data. pages 1489–1493, feb 2010. URL <http://dl.acm.org/citation.cfm?id=1833006.1833120>.

- M. A. F. Jensen and D. Bochtis. Automatic Recognition of Operation Modes of Combines and Transport Units Based on GNSS Trajectories. In *4th IFAC Conference on Modelling and Control in Agriculture, Horticulture and Post Harvest Industry*, volume 4, pages 213–218, Espoo, Finland, aug 2013. ISBN 978-3-902823-44-1. doi: 10.3182/20130828-2-SF-3019.00059. URL <http://www.ifac-paperonline.net/Detailed/61845.html>.
- S. John, S. Hahmann, A. Rousell, M.-O. Löwner, and A. Zipf. Deriving incline values for street networks from voluntarily collected GPS traces. *Cartography and Geographic Information Science*, 2016.
- R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, (82 (Series D)):35 – 45, 1960. URL <http://www.citeulike.org/user/alexv/article/347166>.
- P. Kampstra. Beanplot: A Boxplot Alternative for Visual Comparison of Distributions. *Journal of Statistical Software*, 28(code snippet 1):1–9, 2008. ISSN 15487660. URL <http://www.jstatsoft.org/v28/c01/paper>.
- S. Karagiorgou and D. Pfoser. On vehicle tracking data-based road network generation. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, page 89, New York, New York, USA, nov 2012. ACM Press. ISBN 9781450316910. doi: 10.1145/2424321.2424334. URL <http://dl.acm.org/citation.cfm?doid=2424321.2424334>.
- P. Kasemsuppakorn and H. A. Karimi. Personalised routing for wheelchair navigation. *Journal of Location Based Services*, 3(1):24–54, mar 2009. ISSN 1748-9725. doi: 10.1080/17489720902837936. URL <http://www.tandfonline.com/doi/abs/10.1080/17489720902837936>.
- P. Kasemsuppakorn and H. A. Karimi. A pedestrian network construction algorithm based on multiple GPS traces. *TRANSPORTATION RESEARCH PART C*, 26:285–300, 2013. ISSN 0968-090X. doi: 10.1016/j.trc.2012.09.007. URL <http://dx.doi.org/10.1016/j.trc.2012.09.007>.
- M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, jan 1988. ISSN 0920-5691. doi: 10.1007/BF00133570. URL <http://link.springer.com/10.1007/BF00133570>.
- D. Kortenbruck and H. W. Griepentrog. Ermittlung von Einsatzprofilen durch automatisierte Arbeitszeitanalyse an Landmaschinen: Universität Hohenheim. In *VDI-MEG Tagung Land. Technik*, pages 227–235. VDI-Verlag, Düsseldorf, 2014.
- O. Kounadi. *Assessing the quality of OpenStreetMap data*. PhD thesis, University College of London, 2009.
- A. Krause. *Einführung eines GIS für die Landwirtschaftsverwaltungen der BRD auf Grundlage EU-rechtlicher und nationaler Verordnungen: Unter besonderer Berücksichtigung des Bundeslandes Mecklenburg-Vorpommern: Amazon.de: Arno Krause, Martin Kappas: Bücher*. ibidem, 2006. ISBN 978-3898217385.
- J. Lauer, A. Jochem, and A. Zipf. Straßenzustandsermittlung durch Klassifikation mobiler Sensordaten von Smartphones. In *8. GI/KuVS-Fachgespräch "Ortsbezogene Anwendungen und Dienste"*, number i, 2011.

- J. Lauer, N. Billen, and A. Zipf. Processing crowd sourced sensor data. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science - IWCTS '13*, pages 43–48, New York, New York, USA, 2013. ACM Press. ISBN 9781450325271. doi: 10.1145/2533828.2533839. URL <http://dl.acm.org/citation.cfm?doid=2533828.2533839>.
- J. Lauer, L. Richter, T. Ellersiek, and A. Zipf. TeleAgro+: analysis framework for agricultural telematics data. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on Computational Transportation Science - IWCTS '14*, pages 47–53, New York, New York, USA, nov 2014. ACM Press. ISBN 9781450331388. doi: 10.1145/2674918.2674925. URL <http://dl.acm.org/citation.cfm?id=2674918.2674925>.
- J. G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *Proceedings - International Conference on Data Engineering*, pages 140–149, 2008. ISBN 9781424418374. doi: 10.1109/ICDE.2008.4497422.
- X. Li, C. Claramunt, and C. Ray. A grid graph-based model for the analysis of 2D indoor spaces. *Computers, Environment and Urban Systems*, 34(6):532–540, nov 2010. ISSN 01989715. doi: 10.1016/j.compenvurbsys.2010.07.006. URL <http://www.sciencedirect.com/science/article/pii/S0198971510000797>.
- A. Liaw and M. Wiener. Classification and Regression by randomForest. *R News*, 2(3): 18–22, 2002. URL <http://cran.r-project.org/doc/Rnews/>.
- T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, oct 1979. ISSN 00010782. doi: 10.1145/359156.359164. URL <http://dl.acm.org/citation.cfm?id=359156.359164>.
- I. Ludwig, A. Voss, and M. Krause-Traudes. A comparison of the street networks of Navteq and OSM in Germany. *Lecture Notes in Geoinformation and Cartography*, 1, 2011. doi: 10.1007/978-3-642-19789-5_4. URL <http://www.citeulike.org/user/rfels/article/9928887>.
- D. Luxen and C. Vetter. Real-time routing with OpenStreetMap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11*, page 513, New York, New York, USA, nov 2011. ACM Press. ISBN 9781450310314. doi: 10.1145/2093973.2094062. URL <http://dl.acm.org/citation.cfm?id=2093973.2094062>.
- M. Mabrouk, T. Bychowski, J. Williams, H. Niedzwiadek, Y. Bishr, J.-F. Gaillet, N. Crisp, W. Wilbrink, M. Horhammer, and G. Roy. OpenGIS Location Services (OpenLS): Core Services, OpenGIS® Implementation Specification, 2005. URL http://portal.opengeospatial.org/files/?artifact_id=3839&version=1.
- J. Macqueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967. ISSN 00970433. doi: citeulike-article-id:6083430.
- C. U. o. B. Mayer. Verkehrsinformationen in Geodateninfrastrukturen - Ein Sensorbasierter Ansatz. *GIScience*, (02):31–40, 2009. URL <http://www.wichmann-verlag.de/gis-fachzeitschriften/artikelarchiv/2009/gis-science-ausgabe->

- 02-2009/verkehrsinformationen-in-geodateninfrastrukturen-ein-sensor-basierter-ansatz.html.
- P. Mooney, P. Corcoran, and A. C. Winstanley. Towards quality metrics for OpenStreetMap. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*, page 514, 2010. doi: 10.1145/1869790.1869875. URL <http://portal.acm.org/citation.cfm?doid=1869790.1869875>.
- S. Morris. Introduction Pathway Extraction using Snakes with GPS Initialization. In *Proc. 2nd International Conference on Geographic Information Science*, 2002. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.111.4890>.
- S. Morris, A. Morris, and K. Barnard. Digital Trail Libraries. In *Proceedings of the 2004 joint ACM/IEEE conference on Digital libraries - JCDL '04*, page 63, New York, New York, USA, jun 2004. ACM Press. ISBN 1581138326. doi: 10.1145/996350.996367. URL <http://dl.acm.org/citation.cfm?id=996350.996367>.
- A. Müller, P. Neis, M. Auer, and A. Zipf. Ein Routenplaner für Rollstuhlfahrer auf der Basis von Einführung in die Thematik Motivation und Zielsetzung Stand der Technik. In *Angewandte Geoinformatik 2010*, pages 1–4, Salzburg, Austria, 2010.
- Navteq. See how the map changes within one and a half years, 2012. URL <http://mapchanges.navigation.com/>.
- P. Neis and A. Zipf. Zur Kopplung von OpenSource , OpenLS und OpenStreetMaps in OpenRouteService . org. In *AGIT 2008, Symposium Angewandte Geoinformatik, Salzburg*, Salzburg, 2008.
- P. Neis and A. Zipf. Analyzing the Contributor Activity of a Volunteered Geographic Information Project — The Case of OpenStreetMap. *ISPRS International Journal of Geo-Information*, 1(3):146–165, jul 2012. ISSN 2220-9964. doi: 10.3390/ijgi1020146. URL <http://www.mdpi.com/2220-9964/1/2/146/http://www.mdpi.com/2220-9964/1/2/146>.
- P. Neis, D. Zielstra, and A. Zipf. The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007-2011. *Future Internet*, 4(4):1–21, dec 2011. ISSN 1999-5903. doi: 10.3390/fi4010001. URL <http://www.mdpi.com/1999-5903/4/1/1>.
- P. Newson and J. Krumm. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, page 336, New York, New York, USA, nov 2009. ACM Press. ISBN 9781605586496. doi: 10.1145/1653771.1653818. URL <http://dl.acm.org/citation.cfm?id=1653771.1653818>.
- B. Niehöfer, R. Burda, C. Wietfeld, F. Bauer, and O. Lueert. GPS Community Map Generation for Enhanced Routing Methods Based on Trace-Collection by Mobile Phones. In *2009 First International Conference on Advances in Satellite and Space Communications*, pages 156–161. IEEE, jul 2009. doi: 10.1109/SPACOMM.2009.31. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5194603>.
- T. Oksanen and A. Visala. Path Planning Algorithms for Agricultural Machines, jul 2007. ISSN 1682-1130. URL <http://ecommons.cornell.edu/handle/1813/10604>.

- OpenStreetMap. OpenStreetMap - Automated Edits code of conduct, 2015. URL http://wiki.openstreetmap.org/wiki/Automated_{_}Edits_{_}code_{_}of_{_}conduct.
- Openstreetmap. Stats – OpenStreetMap Wiki, 2015. URL <http://wiki.openstreetmap.org/wiki/Stats>.
- E. Packer, A. Tzadok, and V. Kluzner. alpha-Shape Based Classification with Applications to Optical Character Recognition. *2011 International Conference on Document Analysis and Recognition*, pages 344–348, 2011. ISSN 1520-5363. doi: 10.1109/ICDAR.2011.77.
- A. Predoehl, S. Morris, and K. Barnard. A Statistical Model for Recreational Trails in Aerial Images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 337–344. IEEE, jun 2013. ISBN 978-0-7695-4989-7. doi: 10.1109/CVPR.2013.50. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6618894>.
- F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, feb 1977. ISSN 00010782. doi: 10.1145/359423.359430. URL <http://portal.acm.org/citation.cfm?doid=359423.359430> <http://dl.acm.org/citation.cfm?id=359423.359430>.
- M. A. Quddus, W. Y. Ochieng, and R. B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, oct 2007. ISSN 0968090X. doi: 10.1016/j.trc.2007.05.002. URL <http://www.sciencedirect.com/science/article/pii/S0968090X07000265>.
- U. Ramer. An iterative procedure for the polygonal approximation of plane curves, 1972. ISSN 0146664X.
- F. Ramm and J. Topf. *OpenStreetMap - Die freie Weltkarte nutzen und mitgestalten*. Lehmanns Media, Berlin, 2010.
- S. Rogers, P. Langley, and C. Wilson. Mining GPS data to augment road models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '99*, pages 104–113, New York, New York, USA, aug 1999. ACM Press. ISBN 1581131437. doi: 10.1145/312129.312208. URL <http://dl.acm.org/citation.cfm?id=312129.312208>.
- O. Roick, L. Loos, and A. Zipf. A Technical Framework for Visualizing Spatio-temporal Quality Metrics of Volunteered Geographic Information. In *Geoinformatik 2012*, Braunschweig, 2012.
- J. Roth. Extracting line string features from GPS logs. In *5. GI/ITG KuVS Fachgespräch "Ortsbezogene Anwendungen und Dienste"*, Schriftenreihe der Georg-Simon-Ohm-Hochschule Nürnberg, number 42, Nürnberg, 2008. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.3562>.
- M. Rylow. OpenMapSurfer, 2014. URL <http://openmapsurfer.uni-hd.de>.
- A. Schilling. Scene Graph Based Approach for Interoperable Virtual Globes. *International Journal of 3-D Information Modeling*, 1(2):46–68, 2012. ISSN 2156-1710. doi: 10.4018/ij3dim.2012040104. URL <http://www.igi-global.com/article/scene-graph-based-approach-interoperable/66864>.

- S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson. Mining GPS Traces for Map Refinement. *Data Mining and Knowledge Discovery*, 9(1):59–87, jul 2004. ISSN 1384-5810. doi: 10.1023/B:DAMI.0000026904.74892.89. URL <http://dl.acm.org/citation.cfm?id=989095.989112>.
- C. E. Shannon. Communication in the presence of noise. *Proceedings of the IEEE*, 86(2): 447–457, 1998. ISSN 00189219. doi: 10.1109/JPROC.1998.659497.
- W. Shen, J. Zhang, and F. Yuan. A new algorithm of building boundary extraction based on LIDAR data. In *2011 19th International Conference on Geoinformatics*, pages 1–4. IEEE, jun 2011. ISBN 978-1-61284-849-5. doi: 10.1109/GeoInformatics.2011.5981049. URL <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5981049>.
- W. Shi, S. Shen, and Y. Liu. Automatic generation of road network map from massive GPS vehicle trajectories. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 48–53. IEEE, oct 2009. ISBN 9781424455218. doi: 10.1109/ITSC.2009.5309871. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5309871>.
- M. Stöcker, U. Raape, C. von Itzenblitz, B. Hauck, and C. V. Itzenblitz. Navigation abseits öffentlicher Strassen - Projektvorhaben zur deutschlandweiten Erfassung von Waldwegen zur Optimierung der forstlichen Logistikkette — agriXchange, 2004. URL <http://www.agriXchange.org/results/literature/navigation-abseits-?ffentlicher-strassen-projektvorhaben-zur-deutschlandweiten-er>.
- The CGAL Project. *{CGAL} User and Reference Manual*. CGAL Editorial Board, 4.2 edition, 2013. URL <http://doc.cgal.org/4.2/CGAL.CGAL/html/packages.html>.
- G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik (Crelle's Journal)*, 1908(133), 1908. ISSN 0075-4102. doi: 10.1515/crll.1908.133.97. URL <http://www.degruyter.com/view/j/crll.1908.issue-133/crll.1908.133.97/crll.1908.133.97.xml>.
- J. Wang, X. Rui, X. Song, X. Tan, C. Wang, and V. Raghavan. A novel approach for generating routable road maps from vehicle GPS traces. *International Journal of Geographical Information Science*, (August):1–23, aug 2014. ISSN 1365-8816. doi: 10.1080/13658816.2014.944527. URL <http://dx.doi.org/10.1080/13658816.2014.944527> <http://www.tandfonline.com/doi/abs/10.1080/13658816.2014.944527>.
- Y. Wang, X. Liu, H. Wei, G. Forman, C. Chen, and Y. Zhu. CrowdAtlas. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services - MobiSys '13*, page 27, New York, New York, USA, jun 2013. ACM Press. ISBN 9781450316729. doi: 10.1145/2462456.2464441. URL <http://dl.acm.org/citation.cfm?id=2462456.2464441>.
- H. Wei, Y. Wang, G. Forman, Y. Zhu, and H. Guan. Fast Viterbi map matching with tunable weight functions. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*, page 613, New York, New York, USA, nov 2012. ACM Press. ISBN 9781450316910. doi: 10.1145/2424321.2424430. URL <http://dl.acm.org/citation.cfm?id=2424321.2424430>.

- S. Worrall and E. Nebot. Automated process for generating digitised maps through GPS data compression. In *Australasian Conference on Robotics and Automation*, 2007.
- S. Wörz, V. Heizinger, H. Bernhardt, and C.-f. Gaese. Routenplanung für landwirtschaftliche Fahrzeuge. In *Referate der 33. GIL-Jahrestagung in Potsdam 2013 - Massendatenmanagement in der Agrar- und Ernährungswirtschaft*, pages 3–6, Potsdam, 2013. M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen. URL [http://www.gil-net.de/publikationen{ }autoren.php?id=31{&}band=25](http://www.gil-net.de/publikationen/{_}autoren.php?id=31{&}band=25).
- J. Zhang, Y. Zhu, J. Krisp, and L. Meng. Derivation of road network from land parcels. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*, page 418, New York, New York, USA, nov 2010. ACM Press. ISBN 9781450304283. doi: 10.1145/1869790.1869851. URL <http://dl.acm.org/citation.cfm?id=1869790.1869851>.
- K. Zhou, A. Leck Jensen, C. Sørensen, P. Busato, and D. Bothtis. Agricultural operations planning in fields with multiple obstacle areas. *Computers and Electronics in Agriculture*, 109:12–22, nov 2014. ISSN 01681699. doi: 10.1016/j.compag.2014.08.013. URL <http://www.sciencedirect.com/science/article/pii/S0168169914002129>.
- D. Zielstra and A. Zipf. Quantitative Studies on the Data Quality of OpenStreetMap in Germany. 2010.
- S. Zlatanova, G. Sithole, M. Nakagawa, and Q. Zhu. Problems In Indoor Mapping and Modelling. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4/W4(December):63–68, nov 2013. ISSN 2194-9034. doi: 10.5194/isprsarchives-XL-4-W4-63-2013. URL <http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-4-W4/63/2013/>.

Eidesstattliche Versicherung

Eidesstattliche Versicherung gemäß §8 der Promotionsordnung der
Naturwissenschaftlich-Mathematischen Gesamtfakultät der Universität Heidelberg

1. Bei der eingereichten Dissertation zu dem Thema “Extraction of routing relevant geodata using telemetry sensor data of agricultural vehicles” handelt es sich um meine eigenständig erbrachte Leistung.
2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.
3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.
4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.
5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt. Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Heidelberg, 10.05.2017

Johannes Lauer