

Freie Universität Berlin
Fachbereich Mathematik und Informatik
Lehrstuhl für Künstliche Intelligenz



MASTERARBEIT DER INFORMATIK

Schwarm-basiertes Mapping für autonome Autos

von *Nicolai Sebastian Steinke*

Gutachter:

Prof. Dr. Daniel Göring

Betreuer:

Prof. Dr. Daniel Göring und Fritz Ulbrich

Zweitgutachter:

Prof. Dr. Raúl Rojas

6. September 2017

Zusammenfassung

Fahrzeuge mit immer größerem Automatisierungsgrad befinden sich auf dem Vormarsch, jedoch ist die derzeitige Verkehrsplanung noch vollständig auf menschliche Fahrer ausgerichtet.

In naher Zukunft werden moderne Fahrzeuge immer mehr Fahraufgaben in dem von menschlichen Fahrern dominierten Straßenverkehr übernehmen. Das kooperierende Fahren mit anderen Verkehrsteilnehmern stellt hierbei (teil-)autonome Fahrzeuge vor große Herausforderungen, denn viele Autofahrer folgen den Verkehrsregeln nicht mit der strikten Genauigkeit eines Computers.

Dies offenbart sich deutlich bei der Betrachtung von Kreuzungen. Hier finden die oftmals spärlichen Spurmarkierungen häufig nur wenig Beachtung. Eine Steigerung dazu sind große Kreuzungen und Kreisverkehre, bei denen gänzlich auf Spurmarkierungen verzichtet wurde.

In diesen Szenarien müssen sich autonome Fahrzeuge nach dem Fahrverhalten der anderen Verkehrsteilnehmer richten, um sicher und für andere Fahrer vorhersehbar zu agieren. Schwarmbasierte Navigationstechniken können die Entwicklung eines solchen Fahrverhaltens unterstützen.

Im Rahmen dieser Masterarbeit wurde ein Verfahren entwickelt, das eine Schwarm-Kartographie von Straßenabschnitten, insbesondere Kreuzungen und Kreisverkehren, erlaubt. Bei diesem schwarmbasierten Mapping werden die Verkehrsbewegungen anderer Fahrzeuge erfasst und in einer Karte gespeichert, die eine effiziente Auswertung der Daten erlaubt.

Es wird gezeigt, dass sich mit realen Sensoraufzeichnungen des autonomen Forschungsfahrzeugs „MadeInGermany“ präzise Karten aus den Fahrtbewegungen der anderen Fahrzeuge erstellen lassen.

Diese Karten werden einer umfassenden Genauigkeitsanalyse unterzogen. Zudem wird demonstriert, dass sie sich zur Pfadplanung, zur Erkennung von Abweichungen im Verkehrsfluss und zur Vorhersage der Intention anderer Verkehrsteilnehmer eignen.

Eidesstattliche Erklärung

Ich versichere, die Masterarbeit selbstständig und lediglich unter Benutzung der angegebenen Quellen und Hilfsmittel verfasst zu haben. Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Berlin, den 6. September 2017

Nicolai Sebastian Steinke

Inhaltsverzeichnis

Abbildungsverzeichnis	3
Tabellenverzeichnis	4
1 Einleitung	5
1.1 Motivation	5
1.2 Zielsetzung	6
1.3 Struktur dieser Arbeit	6
1.4 Projektbeschreibung	7
1.4.1 AutoNOMOS Labs	7
1.4.2 SWARMS: Schwarmintelligenz für Autonome Fahrzeuge	10
2 Grundlagen	12
2.1 Autonomes Fahren	12
2.2 Schwarmintelligenz	12
2.3 Stand der Forschung	13
2.3.1 Schwarmverhalten in der Robotik	13
2.3.2 Schwarmverhalten im Straßenverkehr	14
2.3.3 Schwarmverhalten autonomer Straßenfahrzeuge	14
3 Implementierung	16
3.1 Verwendete Software	16
3.1.1 Robot Operating System	16
3.1.2 Octomap Framework	16
3.1.3 Octomap-Server	17
3.1.4 Octomap-Saver	18
3.1.5 Octovis	18
3.1.6 Point Cloud Library (PCL)	19
3.1.7 QGIS und GeographicLib	20
3.2 Architektur	21
3.3 Datenerfassung	21
3.4 Datenspeicherung in der Octomap	23
3.5 Visualisierung in Octovis	23
3.6 Verwendung als Grundlage zur Pfadplanung in ROS	27
4 Experimente und Auswertung	29
4.1 Experiment mit Echtweltdaten	29
4.1.1 Untersuchung der Schwarmkarte	30
4.1.2 Vergleich der Umlaufanzahl	31
4.2 Auswertung der Genauigkeit der Karten	32
4.2.1 Methodik	33

4.2.2	Ergebnisse	34
4.3	Evaluation weiterer Nutzungsszenarien	37
4.3.1	Evaluation der Verwendung zur Abweichungserkennung	37
4.3.2	Evaluation der Verwendung zur Intentionsvorhersage	39
5	Zusammenfassung und Ausblick	41
5.1	Zusammenfassung	41
5.2	Ausblick	41
5.2.1	Erweiterung der Karten mit Fahrradfahrern	42
5.2.2	Erweiterung der Karten mit zusätzlichen Merkmalen	42
5.2.3	Car2Car- und Car2I-Kommunikation	42
6	Anhang	45
6.1	Schwarmkarten aus Taxi-Trajektorien	45
	Literaturverzeichnis	49

Abbildungsverzeichnis

1	DARPA Urban Challenge 2007 Teilnehmerfahrzeug „Spirit of Berlin“ des AutoNOMOS Labs-Projekts. Bild: AutoNOMOS Labs	8
2	Forschungsfahrzeug „MadeInGermany“ des AutoNOMOS Labs-Projekts. Bilder: AutoNOMOS Labs	9
3	Das autonome, elektrisch betriebene Forschungsfahrzeug e-Instein. Bild: AutoNOMOS Labs	10
4	ROS-Systeme lassen sich als Graphen darstellen.	16
5	Darstellung eines beispielhaften Octrees. Freie Knoten sind grau, belegte schwarz. Links befindet sich eine räumliche Darstellung, rechts die Baumstruktur. Bild: [19]	17
6	Octovis bei der Darstellung einer Schwarmkarte des Breitenbachplatzes in Berlin	19
7	QGIS bei der Darstellung der Trajektorien beobachteter Fahrzeuge. Satellitenbild: ©2017, Google, DigitalGlobe, GeoBasis-DE/BKG, GeoContent . . .	20
8	Schematische Darstellung der Software-Module und Werkzeuge	21
9	Die acht verwendeten Richtungsklassen.	22
10	Schwarmkartendarstellung von Octovis	24
11	Die Farbeinteilung der acht verwendeten Richtungsklassen für die Kartendarstellung in Octovis.	26
12	Gradientenmodus für Schwarmdaten in Octovis	26
13	Ein geplanter Weg (grün) auf einer aus einer Schwarmkarte erstellten Costmap.	28
14	Fahrtstrecke des Messfahrzeugs. Straßenkarte: [29]	29
15	Verschiedene Darstellungen der Schwarmkarte Englerallee-Schorlemerallee-Breitenbachplatz in Berlin (0,5m Auflösung). Straßenkarte: [29]	30
16	Schwarmkarten des Breitenbachplatzes und der Kreuzung Schorlemerallee-Englerallee in Berlin (Auflösung 0,5m)	32
17	Referenzdaten (rot). Satellitenbild: ©2017, Google, DigitalGlobe, GeoBasis-DE/BKG, GeoContent	33
18	Ergebnisse der Abweichungserkennung. Satellitenbilder: ©2017, Google, DigitalGlobe, GeoBasis-DE/BKG, GeoContent	38
19	Ergebnisse der Intentionsvorhersage. Satellitenbild: ©2017, Google, DigitalGlobe, GeoBasis-DE/BKG, GeoContent	39
20	Die Porto Schwarmkarte in Octovis.	45
21	Die Porto-Schwarmkarte in Octovis in der Gradientendarstellung.	46
22	Autobahnkreuz der Porto-Schwarmkarte	47
23	Die Porto-Schwarmkarte (türkis) über der Straßenkarte von OpenStreetMap im Vergleich mit der Schwarmkarte in Octovis. Karte: [29]	48

Tabellenverzeichnis

1	Genauigkeitsanalyse verschiedener Auflösungen der Schwarmkarte Englerallee-Schorlemerallee-Breitenbachplatz.	34
2	Binäre Genauigkeitsanalysen verschiedener Auflösungen der Schwarmkarte Englerallee-Schorlemerallee-Breitenbachplatz.	35
3	Binäre Genauigkeitsanalysen der Schwarmkarten der Kreuzung Schorlemerallee-Englerallee und des Breitenbachplatzes.	36

1 Einleitung

1.1 Motivation

In den letzten Jahren haben sämtliche führende Autohersteller teil- oder gar vollautonome Autos in Aussicht gestellt [11] [16] [21] [43] [45]. Mit jedem höheren Grad der Automatisierung des Fahrens müssen die Fahrzeuge im Stande sein, immer größere Herausforderungen des Straßenverkehrs ohne menschliche Hilfe zu bewältigen [5] [34].

Die Hauptaufgabe von Fahrzeugen im Straßenverkehr ist der Transport von Menschen und Gütern von einem Ursprungsort zu einem vor Fahrtantritt bekanntem Ziel. Ein vollautonomes Fahrzeug müsste deshalb in der Lage sein diese Transportaufgabe ohne menschliches Eingreifen zu lösen. Für die Lösung dieser Aufgabe muss das autonome Fahrzeug einen fahrbaren Weg von seiner Position zu dem Zielort berechnen und anschließend sicher durch den Straßenverkehr dorthin navigieren.

Daraus folgt, dass die Berechnung des Weges und die sichere Navigation im Straßenverkehr von großer Bedeutung für vollautonome Fahrzeuge sind. Für die Berechnung des Weges werden bei heutigen, autonomen Fahrzeugen häufig zuvor erfasste Straßenkarten genutzt [6]. Für die sichere Navigation im Straßenverkehr wird oft auf sogenannte „Controller-Module“ zurückgegriffen, die Beschleunigungs- und Bremsvorgänge oder Lenkbewegungen steuern [17].

Der Straßenverkehr ist jedoch ein hochkomplexes, dynamisches System und statische Karten können den resultierenden Anforderungen nur sehr begrenzt gerecht werden: Straßen werden vorübergehend und unvorhersehbar gesperrt, Witterungsbedingungen verlangen das Fahren abseits von Spurmarkierungen, und die menschliche Fahrweise entspricht häufig nicht vollständig der ursprünglichen Intention der Straßenplaner.

Die steigende Population voll- und teilautonomer Fahrzeuge im Straßenverkehr führt durch die Durchmischung von menschlichen und künstlichen Fahrern in einer Vielzahl von Verkehrssituationen zu einer kontinuierlichen Mensch-Maschine-Interaktion. Die Gesamtheit einer Anzahl von autonomen und menschlichen gesteuerten Fahrzeugen, die sich zeitgleich an einer Kreuzung befinden, lässt sich als Schwarm auffassen, der die sichere und schnelle Überquerungen der Kreuzung der Schwarmteilnehmer zum Ziel hat. Hier können Schwarm-Algorithmen den autonomen Fahrzeugen helfen ein natürliches und für den menschlichen Fahrer vorhersehbares Fahrverhalten zu produzieren.

Während Schwarmverhalten im Tierreich häufiger zur Inspiration von Algorithmen herangezogen werden (z. B. Ameisenkolonien-Optimierungsverfahren), sind menschliche Schwärme seltener Ausgangspunkt einer Algorithmus-Neuentwicklung. Dies hat sich jedoch in den letzten Jahren geändert, da sich beispielsweise der Straßenverkehr mit seiner großen Menge an komplexen Interaktionen der Verkehrsteilnehmer nicht auf traditionelle Weise modellieren lässt [3].

Algorithmen, die dem Verhalten von Tierschwärmen nachempfunden sind, wie die bei Reynolds [32] beschriebenen, lassen sich nicht ohne größere Modifikationen auf den Straßenverkehr übertragen. Einer der Gründe dafür ist, dass die Fortbewegung menschlicher Schwärme häufig durch Regeln gehemmt ist, welche die natürliche Schwarmbewegung einschränken. Insbesondere der Straßenverkehr ist durch die strenge Reglementierung eine große Herausforderung für die Analyse und Berechnung von Schwarmverhalten. Des Weiteren ist ein Automobil in der Regel ein nicht-holonomes Fahrzeug d. h., es kann sich nicht zu jeder Zeit in alle Richtungen frei bewegen. Auch diesen Umstand muss ein Navigationsalgorithmus berücksichtigen, der ein Schwarmverhalten simulieren soll.

Es ist daher besonders schwierig ein künstliches, dezentrales und reaktives Schwarmverhalten zu entwickeln, das zu jeder Zeit sämtlichen Verkehrsregeln Folge leistet und dennoch natürlich wirkt. Die Erstellung schwarmbasierter Karten von Verkehrskreuzungen könnte hierbei einen entscheidenden Beitrag zur Vereinfachung des Problems leisten, weil diese Karten helfen durch die Verarbeitung vergangener, ähnlicher Situationen das Verhalten anderer Verkehrsteilnehmer vorherzusagen und die eigene Wegplanung an die Fahr-Trajektorien anderer Fahrzeuge anzupassen.

1.2 Zielsetzung

Im Rahmen dieser Arbeit soll ein System entwickelt werden, das eine Umgebungskarte aus der sensorisch erfassten Bewegung anderer Verkehrsteilnehmer (insbesondere Kraftfahrzeuge) erzeugt.

Die erstellten Karten sollen hinsichtlich ihrer Genauigkeit und Abdeckung analysiert werden. Außerdem soll ihre Eignung untersucht werden, Abnormalitäten im Verkehrsfluss festzustellen und die Intention anderer (menschlich gesteuerter) Fahrzeuge vorherzusagen.

Bei dieser Evaluation wird auf die unterstützende Leistung der schwarmbasierten Karten hinsichtlich urbaner Verkehrsszenarien Wert gelegt, insbesondere bezüglich stark befahrener, mehrspuriger Kreuzungen.

1.3 Struktur dieser Arbeit

Nachdem in den vorhergehenden Abschnitten die Motivation des Ansatzes und die Zielsetzung dieser Arbeit erläutert wurden, wird im nächsten Abschnitt eine Einführung in die mit dieser Masterarbeit verbundenen Forschungsprojekte *AutoNOMOS Labs* und *SWARMS* und der Forschungsfahrzeuge gegeben.

Das folgende 2. Kapitel beschreibt die verschiedenen Automatisierungsgrade des autonomen Fahrens und gibt eine kurze Einführung in Schwärme und die damit verbundenen Begriffe. Daran anschließend wird der aktuelle Forschungsstand in Bezug auf Schwärme in der Robotik, im Straßenverkehr und in autonomen Straßenfahrzeugen präsentiert.

Das 3. Kapitel „Implementierung“ beschreibt die für die Implementierung und Evaluierung verwendete Software, insbesondere das *Robot Operating System*, welches die Grundlage für viele autonome Roboter bildet, und das *Octomap-Framework*, das Kernbestandteil der Kartenimplementierung ist.

Die weiteren Abschnitte des Kapitels erläutern die Architektur der Implementierung, die Implementierung der Datenerfassung und Vorverarbeitung, der Datenspeicherung und Verwaltung, und der Visualisierung. Den letzten Abschnitt des Kapitels bildet die Beschreibung der Modifikationen für die Verwendung als Pfadplanungsgrundlage.

Im darauffolgenden Kapitel 4 werden die durchgeführten Ergebnisse präsentiert und ausgewertet. Nach einer Beschreibung der verwendeten Daten werden visuelle Auffälligkeiten der aus den Daten gewonnenen Karte dargelegt. Danach werden Karten dargestellt und visuell analysiert, die mit verschiedenen großen Teilmengen der Daten erzeugt wurden.

Der daran anschließende Abschnitt erklärt die Methodik und zeigt die Ergebnisse der Genauigkeitsanalyse für die beschriebenen Karten. Den Abschluss des Kapitels bilden die Evaluationen der Abweichungserkennung und Intentionsvorhersage, die auf Basis der erstellten Karten prototypisch umgesetzt wurden.

Der diese Arbeit abschließende Abschnitt 5 bietet eine Zusammenfassung dieses Dokuments und zieht ein kurzes, einschätzendes Fazit über die gewonnenen Ergebnisse und Erkenntnisse. Zusätzlich wird ein Ausblick gewagt, was mit dem System noch möglich ist und welche Verbesserungen anzuraten wären.

1.4 Projektbeschreibung

1.4.1 AutoNOMOS Labs

Das *AutoNOMOS Labs-Projekt* ist ein Forschungsprojekt bezüglich autonomer Fahrzeuge der Freien Universität Berlin innerhalb des *Dahlem Center of Machine Learning and Robotics*.

Seit 1998 wurden an der Freien Universität Berlin kleinere autonome Roboter gebaut (Durchmesser von 18cm bzw. 50cm). Mit dieser langjährigen Erfahrung entschied das spätere AutoNOMOS Labs-Team im Jahr 2006 größere Roboter zu bauen.

Es wurde schließlich der Plan gefasst einen auf Handgas umgerüsteten Dodge Caravan zu einem autonomen Auto zu entwickeln [15]. Nach nur neun Monaten Entwicklungszeit erfolgte die gemeinschaftliche Teilnahme mit Forschern und Studenten der *Rice University* und der *Fraunhofer Society* an der *DARPA Urban Challenge 2007* [13].

Die DARPA Urban Challenge 2007 war ein Wettbewerb autonomer Fahrzeuge, dessen Aufgabenstellung an den Herausforderungen autonomer Autos orientiert war. Es galt eine Strecke von 60 Meilen autonom in einem Stadtumfeld zurückzulegen. Dabei mussten alle Verkehrsregeln beachtet, Hindernissen ausgewichen und, falls das Fahrzeug in eine Sackgasse geriet,



Abbildung 1: DARPA Urban Challenge 2007 Teilnehmerfahrzeug „Spirit of Berlin“ des AutoNOMOS Labs-Projekts. Bild: AutoNOMOS Labs

Wendemanöver absolviert werden. Zusätzlich mussten die 60 Meilen innerhalb eines Zeitlimits gefahren werden [14].

Trotz der kurzen Entwicklungszeit konnte das Team das Halbfinale in diesem Wettbewerb erreichen und setzt seine Forschung an autonomen Fahrzeugen bis heute fort. [14]

Das Forschungsfahrzeug „MadeInGermany“ der Freien Universität Berlin, welches auf einem Volkswagen Passat Variant 3c basiert, ist der Nachfolger des „Spirit of Berlin“, das für die DARPA Urban Challenge 2007 entwickelt wurde (siehe Abb. 1).

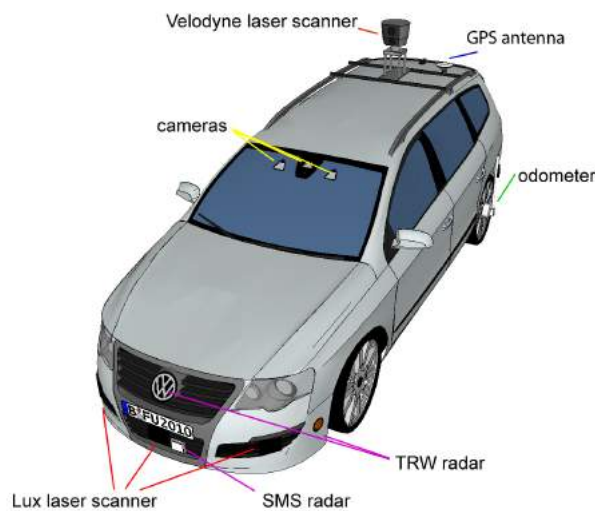
Im Gegensatz zum „Spirit of Berlin“ ist der „MadeInGermany“ explizit auf autonomes Fahren ausgerichtet, denn die „Volkswagen Research Garage“ integrierte eine große Anzahl moderner Sensoren, wie LIDAR, RADAR und Kameras direkt in das Fahrzeug (siehe Abbildung 2) [2].

„MadeInGermany“ absolvierte im September 2011 die Strecke vom Internationalen Congress Centrum zum Brandenburger Tor durch die Berliner Innenstadt mehrfach störungsfrei unter realen Verkehrsbedingungen in beiden Richtungen. Zu der etwa 20 Kilometer langen Strecke gehören unter anderem die großen Kreisverkehre „Großer Stern“ und Ernst-Reuter-Platz und 46 Ampeln [12].

Die schematische Abbildung 2 (b) des „MadeInGermany“ zeigt die verschiedenen verbauten Sensoren. Besonders relevant für die Erkennung anderer Fahrzeuge sind die sechs „Lux“-Laserscanner.



(a) „MadeInGermany“ auf dem ehemaligen Flughafen Berlin Tempelhof



(b) Sensorausstattung des „MadeInGermany“

Abbildung 2: Forschungsfahrzeug „MadeInGermany“ des AutoNOMOS Labs-Projekts. Bilder: AutoNOMOS Labs

Diese sind rund um das Fahrzeug angebracht (Abbildung 2 b zeigt nur die vorderen) und ermöglichen eine 360°-Erkennung von Fahrzeugen und anderen Hindernissen. Die Daten dieser Laserscanner bilden die Grundlage der im Rahmen dieser Arbeit erzeugten Karten.

Zusätzlich zum „MadeInGermany“ verfügt AutoNOMOS Labs noch über „e-Instein“ (Abb. 3). Bei „e-Instein“ handelt es sich um einen modifizierten Mitsubishi i-MiEV, der ähnlich wie das Fahrzeug „MadeInGermany“ über LIDAR, RADAR, Kameras und GPS-Sensorik verfügt [1].



Abbildung 3: Das autonome, elektrisch betriebene Forschungsfahrzeug e-Instein. Bild: Auto-NOMOS Labs

Der Hauptzweck des Forschungsfahrzeugs „e-Instein“ ist die Bereitstellung einer Plattform zur Erforschung autonomer Elektrofahrzeuge. Durch den elektrischen Antrieb und die kompakten Abmessungen des Fahrzeugs ist es außerdem besonders für die Erforschung von Navigation und Lokalisierung innerhalb von Gebäuden, wie z. B. Parkhäusern geeignet [1].

1.4.2 SWARMS: Schwarmintelligenz für Autonome Fahrzeuge

Das SWARMS-Projekt ist ein Forschungsprojekt bezüglich Schwarmintelligenz autonomer Fahrzeuge. Das erklärte Ziel dieses Projekts ist die Erforschung von kooperativen Verkehrsverhalten autonomer und nicht autonomer Fahrzeuge in einem Schwarmverbund. Das Projekt ist ein am 1. Januar 2016 gestartetes, 4-jähriges Drittmittelprojekt und wird durch die Deutsche Forschungsgemeinschaft (DFG) als Teilprojekt des Projekts „SPP 1835: Kooperativ interagierende Automobile“ finanziell unterstützt [7].

Es wird insbesondere auf die Erkennung der Intentionen, Kommunikation und Kooperation von autonomen und menschlichen Verkehrsteilnehmern Wert gelegt. Hierbei sollen verschiedene Ansätze zur Mensch-Maschine-Kommunikation zwischen computer- und manuell gesteuerten Fahrzeugen getestet werden. [8].

Es soll auch eine Erweiterung der Navigation autonomer Fahrzeuge erforscht werden, die es diesen Fahrzeugen ermöglicht sich einem bestehenden Verkehrsfluss anzupassen [8].

Zusätzlich sollen Möglichkeiten der Erstellung von Straßenkarten nach nur einmaligem Befahren, evaluiert werden [8].

Das in dieser Masterarbeit vorgestellte Verfahren zur Kartographierung der Bewegungen anderer Verkehrsteilnehmer könnte eine Grundlage zur Erforschung, Vorhersage und Nutzung dieser kooperativen Verkehrsverhalten bilden.

Das Kartenformat unterstützt den Kartenaustausch und die Kartenaktualisierung zur Laufzeit, sodass mehrere Fahrzeuge gemeinsam eine Karte bearbeiten können. Die Karten könnten die Datengrundlage für kooperative Navigationsalgorithmen für (teil-)autonome Fahrzeuge bilden und dabei helfen Straßenkarten nach nur einmaligem Befahren der Strecke zu erzeugen.

2 Grundlagen

2.1 Autonomes Fahren

Autonome, selbstfahrende oder fahrerlose Fahrzeuge gelten als Zukunft der Mobilität. Um rechtliche Unsicherheiten bei der Zulassung von Fahrzeugen mit immer höherem Automatisierungsgrad zu vermeiden, haben Standardisierungsorganisation ein System entwickelt, das den Automatisierungsgrad von Fahrzeugen in fünf [5] bzw. sechs [34] Stufen einteilt. Wobei die niedrigste Stufe unassistiertes Fahren bezeichnet und die höchste automatisiertes Fahren bei dem keine Eingriffe des Fahrers mehr nötig sind, um die Fahrzeugsicherheit in allen Situationen aufrecht zu erhalten. Die sechs Stufen der Automatisierung des Fahrens nach dem Standard der Organisation *SAE International* sind [34]:

- 0 Keinerlei Automatisierung, manuelles Fahren.
- 1 Assistiertes Fahren. Das Fahrzeug übernimmt entweder die Quer- oder Längsführung bei vollständiger Überwachung durch den Fahrer.
- 2 Partielle Automatisierung. Das Fahrzeug übernimmt die Quer- und Längsführung bei vollständiger Überwachung durch den Fahrer.
- 3 Bedingte Automatisierung. Das Fahrzeug fährt autonom ohne Überwachung durch den Fahrer. Dieser muss nur nach Aufforderung eingreifen können.
- 4 Hohe Automatisierung. Das Fahrzeug fährt autonom ohne Überwachung durch den Fahrer. Auch in dem Fall, dass dieser nach Aufforderung nicht eingreift, entsteht keine Gefährdung.
- 5 Vollständige Automatisierung. Das Fahrzeug ist in der Lage jede Verkehrssituation autonom zu lösen, die für einen menschlicher Fahrer lösbar ist.

Die fünf Stufen der Bundesanstalt für Straßenwesen [5] lassen sich den aufgezählten Stufen 0 bis 4 zuordnen. Für die Stufe 5 gibt es dort keine Entsprechung.

In dieser Masterarbeit werden ausschließlich die Stufen verwendet, die hier nach dem Standard der *SAE International* [34] aufgezählt wurden.

2.2 Schwarmintelligenz

Kooperatives Verhalten in einem Schwarmverband, bei dem eine größere Anzahl Individuen ohne zentrale Koordination auf ein gemeinsames Ziel kooperativ hinarbeiten, ist ein bekanntes Phänomen aus der Tierwelt. Insbesondere hochorganisierte Insektenvölker, wie Termiten, Ameisen und Bienen, stoßen bei Forschern auf großes Interesse [20] [22].

Solche Schwärme finden sich nicht nur bei Insekten, sondern auch bei Fischen, Säugetieren und beim Menschen. Der Mensch als soziales Lebewesen nutzt seit jeher kooperative Verhaltensweisen, um Ziele zu erreichen, die für ein einzelnes Individuum unerreichbar sind.

In Zeiten größerer Vernetzung können diese Verhaltensweisen nicht nur zur Erreichung physischer Ziele genutzt werden, wie bei der gemeinschaftlichen Jagd, sondern auch für immaterielle Zwecke, wie die Sammlung von Wissen oder die Vorhersage der Ergebnisse demokratischer Wahlen [24].

Hierbei spielt der Begriff *Schwarmintelligenz* eine Rolle, den J. Krause et al. [24] folgendermaßen definiert: Zwei oder mehr Individuen sammeln unabhängig voneinander Informationen, die durch soziale Interaktion verarbeitet wird und eine Lösung für ein kognitives Problem ergibt, welche für ein einzelnes Individuum nicht erreichbar ist.

In der Informatik existieren einige Algorithmen, die von natürlichen Schwärmen inspiriert sind. Als bekannte Beispiele gelten das Ameisenkolonien-Optimierungsverfahren, das dem Verhalten von Ameisen nachempfunden ist [10] und das Partikel-Schwarm-Optimierungsverfahren, welches das Sozialverhalten von Menschen nachzuahmen versucht [23].

Ameisen kommunizieren über Pheromone, die sie auf ihrem Weg zurücklassen. Die in dieser Arbeit vorgestellten schwarmbasierten Karten werden aus den Bewegungsdaten anderer Fahrzeuge erstellt. Diese *Schwarmkarten* sind von dem Verhalten der Ameisen inspiriert und zeichnen die Bewegungen anderer Fahrzeuge auf als würden sie Pheromone auf ihrem Weg zurücklassen.

2.3 Stand der Forschung

2.3.1 Schwarmverhalten in der Robotik

Schwärme von Robotern sind ein relativ neues Feld intensiver Forschung. Häufig wird für dieses Forschungsfeld der Begriff *Schwarmrobotik* (engl. *swarm robotics*) verwendet. Die Definitionskriterien dieses Begriffs sind nach Şahin [35]:

- Die Roboter sind autonom.
- Die Roboter sind in größerer Anzahl vorhanden, wobei auch kleinere Gruppen möglich sind, sofern der Forschungsansatz auch auf größere Gruppen skalierbar ist.
- Die Roboter sind in wenige, homogene Gruppen eingeteilt. Homogen bezeichnet hierbei, dass die einzelnen Roboter eine Gruppe unter gleichen Bedingungen identisches Verhalten zeigen.
- Die Roboter sind relativ ineffizient bei der Ausführung der gestellten Aufgabe. Beispielsweise das gemeinsame Bewegen eines Objekts, welches zu schwer für einen einzelnen Roboter des Schwarms wäre. Dieses Kriterium beschränkt aber ausdrücklich nicht die Hardware- oder Softwarekomplexität der Roboter, sondern bezieht sich auf die Ineffizienz eines einzelnen Roboters relativ gesehen zu der Aufgabenstellung.
- Die Roboter verfügen nur über lokale Sensor- und Kommunikationsmöglichkeiten. Falls sie doch Zugriff auf globale Kommunikationskanäle haben, sollten diese nicht für eine globale Kommunikation zwischen den Robotern genutzt werden.

Bezüglich dieser Definitionskriterien stellt Şahin jedoch klar, dass es sich nicht um absolute Kriterien im Sinne einer Checkliste handelt, sondern eher um eine Messlatte, die anzeigt, in welchem Umfang der Begriff Schwarmrobotik auf diese Forschung zutrifft [35].

2.3.2 Schwarmverhalten im Straßenverkehr

Die Betrachtung des Straßenverkehrs als Schwarm ist in der Verkehrsforschung weit verbreitet [30][36][44].

Eine schwarmbasierte Simulation des Straßenverkehrs wurde erfolgreich eingesetzt, um Ampelphasen an stark befahrenen Verkehrskreuzungen zu optimieren [30]. Ein von Bienen-schwärmen inspiriertes Routing-Verfahren bringt teilweise bessere Ergebnisse als andere aktuelle Routing-Algorithmen [36] und Ulbrich et al. [44] hat gezeigt, dass sich Fahrttrajektorien anderer PKWs zur Pfadplanung autonomer Autos eignen.

2.3.3 Schwarmverhalten autonomer Straßenfahrzeuge

Konvois von Straßenfahrzeugen sind alltäglich. Bei dichtem Verkehr ist es nötig, die eigene Fahrweise an die vorausfahrenden Fahrzeuge anzupassen. Moderne PKWs verfügen über eine dynamische Abstandsregelung (*Autonomous Cruise Control* (ACC)), die die Längsregelung des Fahrzeuges übernimmt (es handelt sich hierbei um ein Stufe 1 System, siehe 2.1) und an die Geschwindigkeit des vorausfahrenden Fahrzeugs anpasst.

Konsequenterweise wird auch am sogenannten *Platooning* von (teil-) autonomen Fahrzeugen geforscht. Ein *Platoon* (deutsch: Zug) von Fahrzeugen bezeichnet eine Kette autonomer Fahrzeuge, deren Zwischenabstände durch automatische Quer- und Längsregelung (Stufe 2, siehe 2.1) möglichst konstant gehalten werden sollen [38].

Eine praktische Anwendung eines solchen Systems wäre ein Zug von Lastkraftwagen (LKW) mit einem menschlich gesteuerten LKW an der Spitze und mehreren autonom folgenden dahinter. Eine Technologie, die dies ermöglicht, scheint greifbar [4].

Ein solcher Platoon lässt sich durchaus schon als ein Beispiel für Schwarmrobotik im Straßenverkehr heranziehen (siehe 2.3.1).

Systeme wie ACC eignen sich besonders für Autobahnen und Landstraßen. Das Schwarmverhalten autonomer Autos im städtischen Individualverkehr stellt die Forschung jedoch vor sehr große Herausforderungen, denn hier müssen die autonomen Fahrzeuge eng mit menschlichen Fahrern kooperieren. Menschliches Verhalten lässt sich nicht immer vorher-sagen und der Straßenverkehr als dynamisches System basiert stark auf der gegenseitigen Rücksichtnahme der einzelnen Teilnehmer und setzt kooperatives Verhalten voraus (vergleiche §1 StVO).

In einigen Situationen ist es beispielsweise nötig die Spuren anders zu befahren als vorgesehen oder auf die eigene Vorfahrt zu verzichten, um andere nicht zu gefährden. Diese Situationen treten im dichten Stadtverkehr wesentlich häufiger auf als auf der Landstraße oder der Autobahn.

Autonome Fahrzeuge müssen, um diese Anforderungen zu erfüllen, auf ein lokales und dynamisches Planungsverhalten zurückgreifen können, das situativ reagieren kann.

Ein solches Planungsverhalten ließe sich mit einem schwarmbasierten Pfadplaner verwirklichen, wie den durch Ulbrich et. al. entwickelten, der durch das Clustern von Trajektorien anderer Verkehrsteilnehmer eine empfohlene Schwarm-Trajektorie erstellt [44].

Dieser Ansatz unterscheidet sich von dem in dieser Arbeit vorgestellten dadurch, dass keine gerasterte Schwarmkarte erstellt wird, die zu freiem Planen und weiteren Analysen genutzt werden kann, sondern nur einzelne Trajektorien gespeichert und geclustert werden.

Aus den im Rahmen dieser Masterarbeit generierten Karten können zwar wieder Trajektorien gewonnen, aber auch andere Analysen mit den Daten durchgeführt werden, wie bspw. eine Intentionserkennung von anderen Verkehrsteilnehmern oder eine Abweichungserkennung des Verkehrsflusses. Damit haben die Schwarmkarten mehrere mögliche Anwendungsfälle, die über einfache Pfadplanung hinausgehen.

Ein autonomes Fahrzeug mit einem schwarmbasierten Pfadplaner in einem Verkehrsschwarm menschlich gesteuerter Fahrzeuge ist jedoch ein Grenzfall von Schwarmrobotik (siehe 2.3.1), da ein menschlich gesteuertes Fahrzeug kein autonomer Roboter ist.

Dennoch sind die einzelnen Fahrzeuge voneinander unabhängig gesteuert. Auch die anderen Punkte der Schwarmrobotik-Definition lassen sich in gewissem Maße auf diesen Fall anwenden. Insbesondere dann, wenn sich die Population autonomer Fahrzeuge in Zukunft erhöht, sodass Schwarmrobotik ein anwendbarer Begriff ist.

3 Implementierung

3.1 Verwendete Software

3.1.1 Robot Operating System

Das *Robot Operating System (ROS)* ist ein unter einer BSD-Lizenz publizierte Zusammenstellung von Bibliotheken und Werkzeugen, die die Softwareentwicklung für Roboterplattformen vereinfachen soll [27].

Die Entwicklung von ROS begann unter dem Namen *STAIR (Stanford AI Robot)* im Jahr 2007. Ab 2008 bis 2013 übernahm die Firma Willow Garage die Entwicklung [28]. Im Februar 2013 wurde die Aufgabe der Fortführung von ROS an die *Open Source Robot Foundation* übertragen [46]. Die *Open Source Robot Foundation (OSRF)* ist eine unabhängige, gemeinnützige Organisation, die sich durch Spenden finanziert. Das erklärte Ziel der OSRF ist es die Entwicklung, Verteilung und Akzeptanz von quelloffener Software in der Roboterforschung, -lehre und Produktentwicklung zu fördern.

ROS setzt auf ein System von Modulen, die durch das *ROS-Publisher-Subscriber-System* miteinander kommunizieren. Die Gesamtheit eines ROS-Systems lässt sich als Graph darstellen, bei dem die Knoten laufende Module sind und Publisher-Subscriber-Beziehungen durch Kanten dargestellt werden (siehe Abb. 4). Die Module sind als Ellipsen dargestellt und Publisher-Subscriber-Verbindungen durch Pfeile, die durch ein Rechteck unterbrochen sind, welches den Namen des *ROS-Topics* enthält. Ein ROS-Topic bezeichnet den Nachrichtenkanal, durch den Publisher und Subscriber miteinander kommunizieren. Bei der in dieser Arbeit verwen-

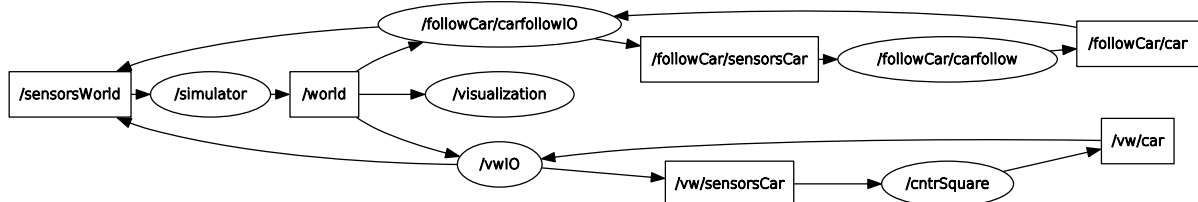


Abbildung 4: ROS-Systeme lassen sich als Graphen darstellen.

deten ROS-Version handelte es sich um „Kinetic Kame“, welche im Mai 2016 veröffentlicht wurde [41].

3.1.2 Octomap Framework

Für die Verwaltung der erfassten Bewegungsdaten wurde ein performantes, erweiterbares, Grid-Mapping-Framework benötigt, das dreidimensionale Karten unterstützt und eine effiziente Serialisierung der Kartendaten ermöglicht.

In einer Evaluation stellte sich das *Octomap*-Framework als das beste Framework heraus, um diesen Anforderungen gerecht zu werden. Das Octomap-Framework ist ein effizientes, probabilistisches 3D-Mapping-Framework, welches von Kai M. Wurm und Armin Hornung an der Universität Freiburg entwickelt wurde. Das Framework wurde unter einer BSD-Lizenz unter <http://octomap.github.com> veröffentlicht [19], verfügt über eine vollständige ROS-Integration und wird aktiv entwickelt. [18]

Das Octomap-Framework speichert die Daten in einer *Octree*-Baumstruktur. Ein Octree ist eine hierarchische Datenstruktur zur räumlichen Unterteilung des dreidimensionalen Raumes. Jeder Knoten (bei Octrees auch *Voxel* genannt) einer Octree-Struktur repräsentiert den Raum innerhalb eines kubischen Volumens (siehe Abbildung 5). Dieses Volumen wird rekursiv in acht Untervolumen unterteilt, bis eine feste Mindestvoxelgröße erreicht ist. Diese Mindestvoxelgröße ist die Auflösung eines Octrees [19].

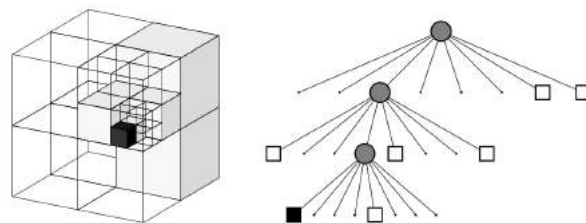


Abbildung 5: Darstellung eines beispielhaften Octrees. Freie Knoten sind grau, belegte schwarz. Links befindet sich eine räumliche Darstellung, rechts die Baumstruktur. Bild: [19]

Jeder existierende Knoten kann drei Zustände annehmen: unbelegt (*Free*), belegt (*Occupied*) und unbekannt (*Unmapped*).

Ferner bietet die Bibliothek Methoden zur komprimierten Speicherung des Octrees in einem Binärformat an, sodass der Speicherverbrauch minimal bleibt.

Für die Verwaltung von Schwarmkarten wurde die Bibliothek um einen neuen Knoten-Typ erweitert. Dieser ermöglicht es, die Anzahl der erfassten Fahrzeuge sowie deren Richtungs- und Geschwindigkeitswerte zu speichern. Weitere Details befinden sich im Abschnitt 3.4.

3.1.3 Octomap-Server

Der Octomap-Server ist die Serverkomponente der ROS-Integration des Octomap-Frameworks. Die Aufgabe dieser Komponente besteht darin, die Octomap zentral zu verwalten und anderen ROS-Modulen zugänglich zu machen.

Die Octomap ist für andere ROS-Nodes durch diesen Server abrufbar und der Server bietet die Möglichkeit neue Daten in die Octomap einzufügen.

Außerdem bietet der Server anderen Modulen eine projizierte 2D-Karte der verwalteten Octomap an. Diese kann als Costmap für 2D-Pfadplaner verwendet werden (Abschnitt 3.6).

Obwohl der Octomap-Server Teil des Octomap-Frameworks ist, musste er angepasst werden, um kompatibel mit dem verwendeten Datenformat (Geschwindigkeit und Richtung der beobachteten Fahrzeuge) zu sein. Es musste sowohl das neu entwickelte PCL-Austauschformat (siehe Abschnitt 3.1.6), als auch die neue Octree-Struktur (siehe 3.4) in den Server eingepflegt werden.

Des Weiteren erwartet der Octomap-Server in der Standardimplementierung als Dateneingabe Punktwolken, die von einem Laserscanner, Radar-System oder einer Stereokamera generiert wurden. Deshalb markiert der Server alle Punkte, die auf einer Linie zwischen dem Zentrum der Punktwolke und einem als belegt gemessenen Endpunkt liegen, als unbelegt. Dabei werden dort eventuell vorhandene Daten überschrieben.

Für die Verwaltung von schwarmbasierten Karten eignet sich dieses Verhalten nicht und musste angepasst werden, sodass keine Knoten mehr als unbelegt markiert werden.

3.1.4 Octomap-Saver

Der Octomap-Saver ist ein Kommandozeilenwerkzeug, das es auf komfortable Weise ermöglicht Octomaps auf dem Dateisystem zu speichern.

Die Octomaps können komprimiert abgelegt werden, sodass sie nur sehr geringe Mengen Speicherplatz verbrauchen.

3.1.5 Octovis

Octovis ist ein Darstellungswerkzeug für Octomaps. Es basiert auf dem Qt-Framework und verwendet für die Kartenvisualisierung die Bibliothek *libQGLViewer*, die eine einfache Verwendung von hardwarebeschleunigter Darstellung von OpenGL in Qt-Applikationen ermöglicht. Im Gegensatz zur Octolib-Bibliothek steht Octovis, bedingt durch die Integration der libQGL-Viewer-Bibliothek, unter der GNU-GPL-Lizenz (Version 2). [47]

Octovis verfügt über vier Darstellungsmodi:

- *Standard*. Dieser Modus stellt die Octomap in einem türkisen Farbton dar.
- *High map (color coded)*. Dieser Modus beschränkt sich auf die Verwendung mit Octrees mit Farbinformationen (sogenannten *ColorOcTrees*) und kann die enthaltene Farbinformation darstellen.
- *Printout (black/white)*. Hierbei handelt es sich um einen Modus für die druckfreundliche Darstellung von Octrees, der belegte Knoten schwarz und unbekannte weiß darstellt.
- *Semantic Coloring*. Dieser Modus ist für die Verwendung mit mehreren unterschiedlichen Octrees gedacht, die in unterschiedlichen Farben dargestellt werden.

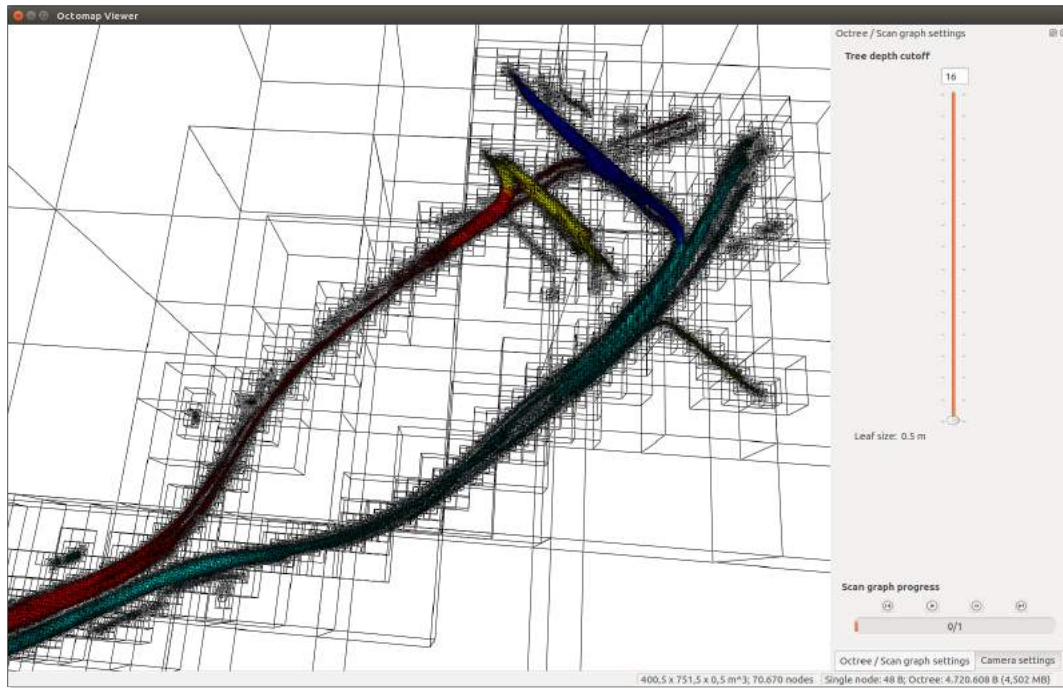


Abbildung 6: Octovis bei der Darstellung einer Schwarmkarte des Breitenbachplatzes in Berlin

Im Rahmen dieser Arbeit wurde Octovis um zwei weitere Modi erweitert, die die Arbeit mit schwarmbasierten Karten erleichtern sollen (siehe Abschnitt 3.5).

Des Weiteren lässt sich der Octree in größeren Auflösungen darstellen und die Baumstruktur einblenden. Auch rudimentäre Bearbeitungsmöglichkeiten sind vorhanden.

3.1.6 Point Cloud Library (PCL)

Die Point Cloud Library (PCL) ist eine umfangreiche, quelloffene Software-Bibliothek für die Verarbeitung zwei- und dreidimensionaler Bilder und Punktwolken. PCL wird unter einer BSD-Lizenz herausgegeben und die Entwicklung von einer großen Anzahl Unternehmen und Institutionen unterstützt [33].

PCL ist vollständig in ROS integriert und das Austauschformat der Wahl für Punktwolken in ROS.

In der Regel entstehen Punktwolken durch Messungen mit RADAR-, LIDAR-Scannern oder Stereokameras. Aber auch für die Bewegungsdaten anderer Fahrzeuge ist dies ein passendes Format, denn auch hier variiert die Anzahl der Datenpunkte und wir erhalten durch die Zusammenfassung aller beobachteten Fahrzeuge eine Punktwolke (siehe 3.3).

PCL ist das durch den Octomap-Server (siehe 3.1.3) vorgegebene Austauschformat. Über die „cloud_in“-ROS-Subscriber-Schnittstelle lassen sich PCL-Punktwolken einfach und effizient in die durch den Octomap-Server verwaltete Octomap einlesen.

Das durch PCL vorgegebene Format lässt sich leicht erweitern und anpassen, sodass die Konvertierung der Beobachtungsdaten in PCL-Punktwolken nur geringe Mengen Quelltext erfordert.

Die Details des verwendeten PCL-Formats sind im Abschnitt 3.3 erläutert.

3.1.7 QGIS und GeographicLib

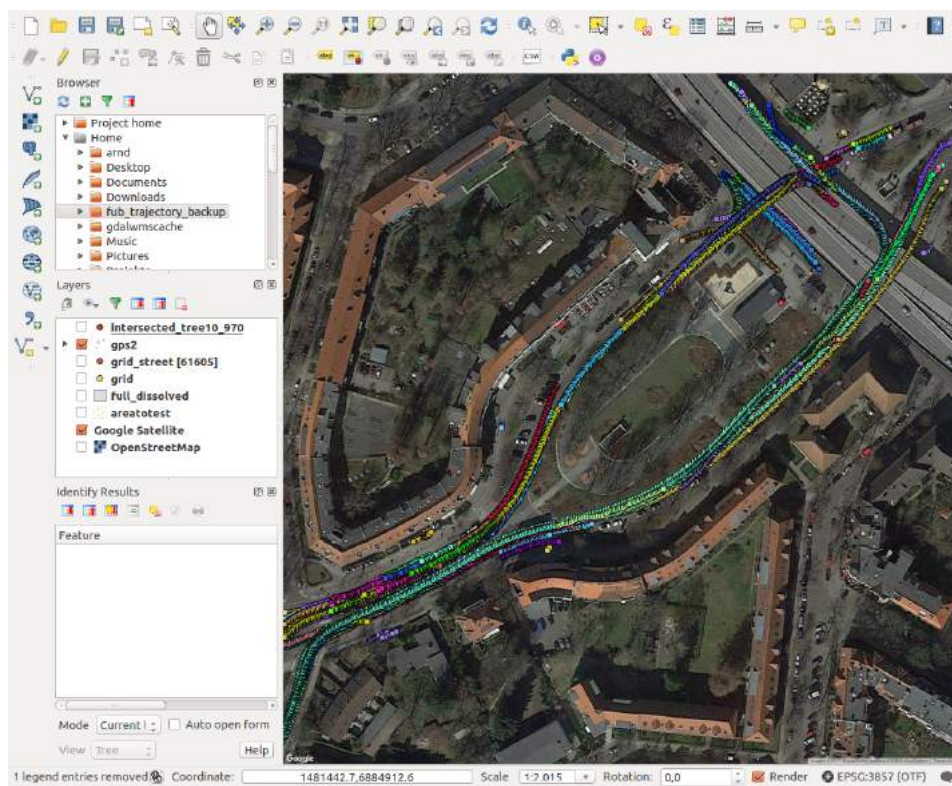


Abbildung 7: QGIS bei der Darstellung der Trajektorien beobachteter Fahrzeuge. Satellitenbild: ©2017, Google, DigitalGlobe, GeoBasis-DE/BKG, GeoContent

QGIS ist ein geographisches Informationssystem (GIS), welches die Arbeit mit geographischen Daten unterstützt und unter der GNU General Public License veröffentlicht wird [40].

Im Rahmen dieser Masterarbeit wurde es hauptsächlich zur Visualisierung und Genauigkeitsanalyse (Abschnitt 4.2) in der Version 2.8 verwendet.

QGIS kann Koordinaten verschiedenster Standards darstellen und bietet vielfältige Analyse-möglichkeiten (siehe Abb. 7). Unter anderem bietet QGIS eine Plugin-Schnittstelle und eine Python-Integration. Die Möglichkeit Python-Skripte in einem QGIS-Kontext mit Zugriff auf die geöffneten geographischen Daten auszuführen, macht es besonders zur automatischen Kartenverifizierung geeignet.

Die unter der MIT-Lizenz stehende [31] und in Version 1.45 verwendete *GeographicLib* ist eine Python-Implementierung der geodätischen Routinen der gleichnamigen C++-Bibliothek. Diese ermöglichen die Umrechnung von lokalen, rechtwinkligen Koordinaten in globale Polarkoordinaten auf dem Erd-Geoid.

Die GeographicLib bietet diese Berechnung für das vom GPS verwendeten geodätischen Referenzsystem *World Geodetic System 1984 (WGS84)* an, welches QGIS verarbeiten kann, sodass es mit dieser Softwarekombination möglich ist Schwarmkarten mit hoher Genauigkeit mit QGIS auszuwerten.

3.2 Architektur

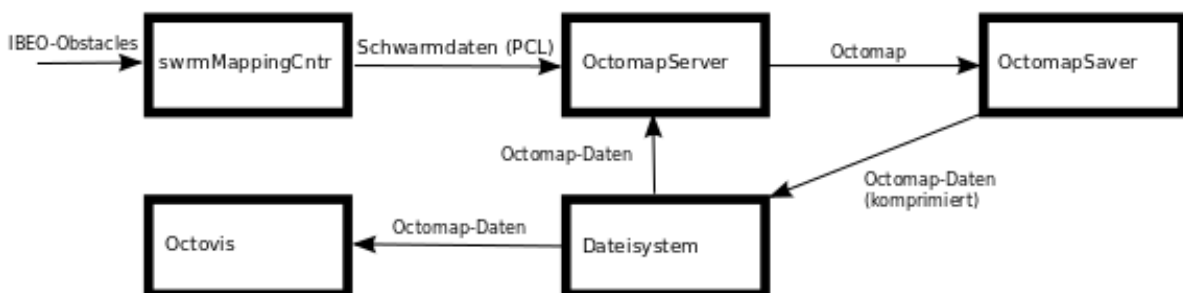


Abbildung 8: Schematische Darstellung der Software-Module und Werkzeuge

Die Implementierung ist in vier Teile unterteilt (siehe Abbildung 8):

- **swrmMappingCntr**: Datenerfassung und Vorverarbeitung (siehe 3.3)
- **OctomapServer**: Verwaltet die Octomap (siehe 3.1.3)
- **OctomapSaver**: Kommandozeilenwerkzeug zum Speichern einer Octomap (siehe 3.1.4)
- **Octovis**: Nachbearbeitung und Visualisierung (siehe 3.1.5)

3.3 Datenerfassung

Die für die Erfassung anderer Verkehrsteilnehmer verwendeten Daten werden durch die IBEO-Laserscanner erfasst und bereitgestellt. Neben der Abmessungen des Fahrzeug wird

auch der Fahrzeugtyp (LKW, PKW, Zweirad oder unbekannt), Bewegungsrichtung und Geschwindigkeit ermittelt.

Die durch die Laserscanner erfassten Daten werden zunächst gefiltert, denn derzeit werden nur Personen- und Lastkraftwagen in der Schwarmkarte unterstützt. Die Klassifizierung des Fahrzeugtyps wird durch die Firmware des IBEO-Lux vorgenommen. Bei einer zu geringen Klassifikationssicherheit wird die Messung verworfen.

Außerdem werden zu weit entfernte Beobachtungen (über 50 Meter vom Messfahrzeug entfernt) und wiederholte Beobachtungen des selben Fahrzeuges an nahezu der gleichen Position verworfen.

Die Bewegungsdaten der anderen Fahrzeuge liegen im lokalen Koordinatensystem des Messfahrzeuges vor und müssen zur Feststellung der globalen Position und Bewegungsrichtung in globale Koordinaten überführt werden. Hierbei spielt die Genauigkeit der Selbstlokalisierung des messenden Fahrzeugs eine besonders große Rolle, da sie damit ausschlaggebend für die Qualität der resultierenden Karte ist.

Anschließend wird die globale Bewegungsrichtung diskretisiert, sodass sie zu einer von acht Richtungsklassen wird. Die acht Richtungsklassen bilden Winkelsegmente von $\frac{\pi}{4}$ zwischen 0 bis 2π im Bogenmaß, siehe schematische Abbildung 9.

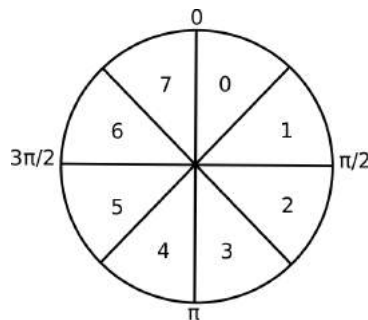


Abbildung 9: Die acht verwendeten Richtungsklassen.

Die Bewegungsgeschwindigkeit der anderen Fahrzeuge wird durch die IBEO-Sensoren als dreidimensionaler Vektor erfasst. Da in der Schwarmkarte bereits die Bewegungsrichtung in Form der Richtungsklasse gespeichert ist, wird nur die Länge des Geschwindigkeitsvektors für die Geschwindigkeit gespeichert.

Die Daten werden nach der Filterung und Vorverarbeitung in PCL-Nachrichten transformiert und anschließend über einen ROS-Publisher an den Octomap-Server gesendet.

Für die Datenübermittlung vom Controller zum Octomap-Server wird auf die Point Cloud Library (PCL 3.1.6) zurückgegriffen. Jede Beobachtung wird basierend auf der durch die Laserscanner erfassten Größe gerastert, sodass ein Fahrzeug mehreren Punkten in der Punktwolke entspricht.

Dafür wird die von den Laserscannern erfasste Breite auf ein Minimum von 1,5m und ein Maximum von 2,6m begrenzt, da es gelegentlich zu Fehlerkennungen mit unrealistischen Werten kommt. Die Länge wird auf 3m festgesetzt, da es bei der Länge häufig zu Messfehlern der Laserscanner kommt. Insbesondere wenn direkt hinter einem anderen Fahrzeug gefahren wird, können die IBEO-Laserscanner die Länge nicht korrekt ermitteln.

Die Gesamtheit der Beobachtungen zu einem Zeitpunkt wird zu einer einzelnen Punktwolke zusammengefasst und an den Octomap-Server übermittelt.

Der Octomap-Server ist auch in der Lage Octomaps als Dateien von einem Dateisystem zu lesen. Zum Speichern der Octomaps kommt das Konsolenwerkzeug *OctomapSaver* zum Einsatz (siehe 3.1.4).

3.4 Datenspeicherung in der Octomap

Die von dem Controller empfangenden Daten werden unverändert in der Octomap gespeichert. Jeder Knoten der Octomap speichert die Durchschnittsgeschwindigkeit der Beobachtungen und die Anzahl der Beobachtungen je Richtungsklasse.

Der Octomap-Server hält die Karte dauerhaft im Speicher vor und erweitert sie mit neu empfangenen Daten. Die Schwarm-Octomap nutzt nur zwei der drei möglichen Knotenzustände (nur *belegt* und *unbekannt*, der Zustand *frei* wird nicht verwendet).

Falls eine neue Beobachtung an einem Knoten eingefügt werden soll, der noch keine Beobachtung enthält, wird der Knoten auf *belegt* gesetzt und die Beobachtung dort eingefügt. Im Falle, dass eine Beobachtung an einem Knoten eingefügt werden soll, der schon eine Beobachtung enthält, werden diese zusammengeführt: Es wird die Anzahl der Beobachtungen pro Richtungsklasse aktualisiert und für die Geschwindigkeit wird der Mittelwert gebildet.

Durch diese effiziente Speicherung ist sichergestellt, dass der Speicherverbrauch für die gesamte Karte moderat bleibt. Der Speicherverbrauch je Knoten beträgt 48 Byte und für eine ganze Karte mit 160.000 Knoten etwa 10 MiB. Dies ist pro Knoten gerechnet größer als 48 Byte, da die äußeren Knoten einen leicht erhöhten Speicherbedarf haben [19].

Zudem ist die Speicherbelegung bei Einfügeoperationen erhöht, da das Entpacken der PCL-Punktwolke zusätzlichen Speicher benötigt. Dies sollte bei der Einfügung von sehr großen Punktwolken berücksichtigt werden.

3.5 Visualisierung in Octovis

Octovis ist das Werkzeug für die Visualisierung und Bearbeitung von Octomaps. Jedoch ist der Funktionsumfang überschaubar, wodurch einige Erweiterungen für den effizienten Umgang mit Schwarmkarten nötig wurden.

Es wurden zwei eigene Renderer für Schwarmkarten implementiert. Der erste umfasst Transparenz und Farbdarstellung abhängig von der Anzahl und Richtung der erfassten Fahrzeuge an jedem Knoten, der zweite orientiert sich an den Gradienten in den Beobachtungsdaten.

Zusätzlich wurde Octovis um OpenGL-Texture-Batch-Rendering erweitert, sodass sich die einzelnen Octomap-Knoten mit Texturen versehen lassen und sich diese, abhängig von den Schwarmdaten, über die Texture-Matrix manipulieren lassen.

Dieses Feature wird im Schwarmkarten-Renderer dafür verwendet um durch Pfeile die dominierende Fahrtrichtung jedes Knotens darzustellen (siehe Abbildung 10).

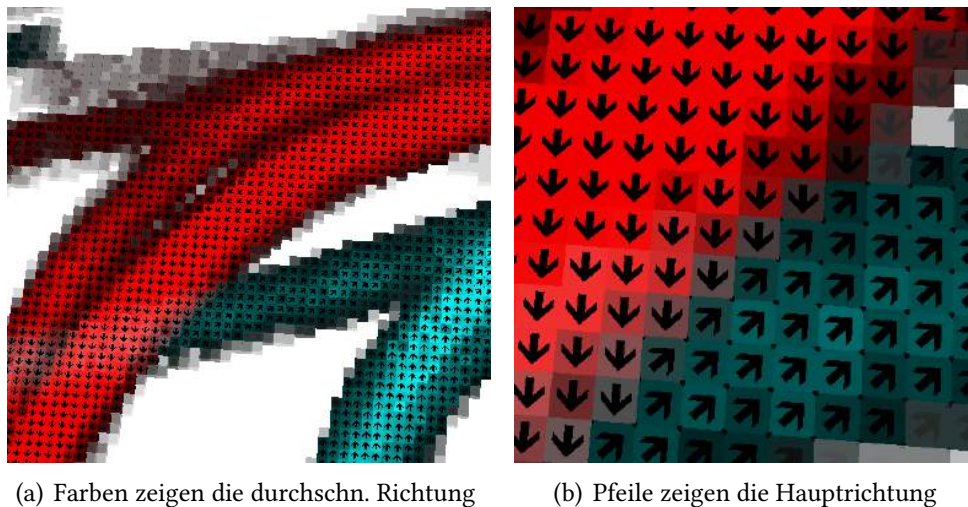


Abbildung 10: Schwarmkartendarstellung von Octovis

Des Weiteren wurde die Erweiterung mit einer Selektionsmöglichkeit via „*Color Coded Selection*“ implementiert. Es lassen sich nun einzelne Knoten selektieren, um die in diesem Knoten gespeicherten Daten (Richtung, Geschwindigkeit und Anzahl Fahrzeuge) abzufragen.

Bei „*Color Coded Selection*“ wird jedes selektierbare Element in einer eigenen einzigartigen Farbe dargestellt. Das resultierende Farbselektionsbild wird nicht angezeigt, sondern in einer Texture gespeichert, die über die gleiche Auflösung wie das OpenGL-Fenster verfügt. Anhand der Klick-Koordinaten der Selektion kann aus dieser Textur die Selektionsfarbe des Objekts an dieser Stelle abgefragt werden. Mit der Selektionsfarbe kann die Beziehung zum ursprünglich selektierten Objekt wieder eindeutig hergestellt werden. [25, S. 365–366]

Die Verwendung von „*Color Coded Selection*“ wurde nötig, da die Octrees durch ein effizientes Batch-Rendering per Vertex-Arrays in Octovis dargestellt werden. Diese Darstellungsmethode ist effizienter als die einzelne Vertexübermittlung, macht es aber unmöglich einen eindeutigen Identifizierer für jedes einzelne Objekt zu vergeben, da diese alle in einem Schritt dargestellt werden [25, S. 15].

Ohne eindeutigen Identifizierer oder Namen lässt sich das selektierte Objekt nicht mehr zuordnen, deshalb wird in diesem Fall häufig auf die „Color Selection“-Technik zurückgegriffen.

Zu Auswertungszwecken der schwarmbasierten Octrees wurde, insbesondere für das Zusammenspiel mit QGIS (siehe Abschnitt 3.1.7), eine Exportmöglichkeit dieser in CSV-Dateien („Comma Separated Values“) nötig. Dies wurde in Octovis implementiert und ermöglicht eine vollständige Text-Serialisierung einer Schwarmkarte.

Die Darstellung des Schwarmkarten-Renderers bietet einen Überblick über die Daten der Karte. Die Transparenz ist abhängig von der Anzahl Beobachtungen an der betreffenden Stelle und die Farbe abhängig von dem Mittelwert der Winkelklassen. Die Gleichung 1 und folgende werden zur Berechnung der Farbwerte verwendet. Hierbei gibt $W(k, i)$ die Anzahl der Beobachtungen des Knotens k mit der Richtungsklasse i zurück. Damit beschreibt $S(k)$ die summierten Beobachtungen aller Richtungsklassen eines Knotens k und $\bar{S}$ steht für den arithmetischen Mittelwert von $S(k)$ für alle Knoten k des aktuellen Octrees.

$$S(k) = \sum_{i=1}^8 W(k, i) \quad (1)$$

$$O(k) = \frac{S(k)}{\bar{S}} \cdot 128 \quad (2)$$

$$R(k) = \frac{W(k, 2) + W(k, 3) + W(k, 4) + W(k, 5)}{\bar{S}} \cdot 85 \quad (3)$$

$$G(k) = \frac{W(k, 0) + W(k, 1) + W(k, 2) + W(k, 3)}{\bar{S}} \cdot 85 \quad (4)$$

$$B(k) = \frac{W(k, 6) + W(k, 7) + W(k, 0) + W(k, 1)}{\bar{S}} \cdot 85 \quad (5)$$

$$(6)$$

Das Ergebnis von $O(k)$ wird für die Deckkraft verwendet, die Ergebnisse von $R(k)$, $B(k)$ und $G(k)$ für den Rot-, Blau- und Grünwert. Alle Werte werden auf den gültigen Bereich von 0 bis 255 limitiert.

Die Formeln erscheinen nicht offensichtlich, deshalb folgt eine kurze Erklärung:

Zunächst Gleichung 2, die die Deckkraft angibt. Die Division durch das arithmetische Mittel führt dazu, dass unterdurchschnittliche Werte abgeschwächt werden, während überdurchschnittliche verstärkt werden. Der Faktor 128 führt dazu, dass Knoten, die über genau die Anzahl Beobachtungen verfügen, die dem arithmetischen Mittel entsprechen, mit etwa der Hälfte der maximale Deckkraft von 255 dargestellt werden.

Die Gleichungen für die Farbwerte (Nummer 3, 4 und 5) arbeiten nach einem ähnlichen Prinzip wie die Gleichung für die Deckkraft. Allerdings ist der Faktor reduziert, um neutralere Farben zu erhalten und es werden nur Teile der Werte der Richtungsklassen summiert. Durch

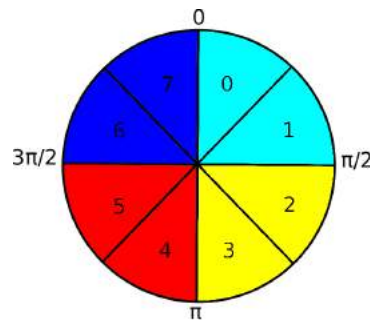


Abbildung 11: Die Farbeinteilung der acht verwendeten Richtungsklassen für die Kartendarstellung in Octovis.

die additive Farbmischung mit dem Grünkanal entstehen die Farben der Klassen 0 bis 3 (siehe Abb. 11).

Für weitere Analysemöglichkeiten, insbesondere hinsichtlich einer Evaluation, ob eine Spurerkennung mittels Schwarmkarten möglich ist, wurde ein Gradienten-Darstellungs-Modus erstellt. Dieser orientiert sich an dem Gradienten der Beobachtungsanzahl und ermöglicht es die relative Veränderung der Anzahl der Beobachtungen im Vergleich zu den Nachbarknoten innerhalb der Daten zu erkennen (Abbildung 12). Hierbei wird ein Knoten abhängig von der



Abbildung 12: Gradientenmodus für Schwarmdaten in Octovis

relativen Abnahme von Beobachtungen verglichen mit seinen Nachbarn dunkel oder hell dargestellt: Je größer die Abnahme, umso dunkler der Knoten. Die Berechnungsformel des Grauwertes ist als Formel 7 angegeben.

Hierbei steht $N(k, i)$ für den Nachbarknoten von k in Richtung i (von 0 bis 8 im Uhrzeigersinn, 0 ist Norden).

$$D(k) = \sum_{i=1}^8 (S(k) - S(N(k, i))) \cdot \frac{-\bar{S}}{S(k)} \quad (7)$$

Der Rückgabewert von $D(k)$ wird zwischen 0 und 255 begrenzt und als Grauwert zur Darstellung verwendet.

Die Berechnung des Grauwertes führt dazu, dass Stellen an denen die Beobachtungen rapide abnehmen, besonders dunkel dargestellt werden. Hierdurch lassen sich tatsächlich gefahrene Fahrspuren und Fahrbahngrenzen besonders gut erkennen (siehe Abb. 12).

3.6 Verwendung als Grundlage zur Pfadplanung in ROS

Die gewonnenen Schwarmkarten lassen sich auch zur Pfadplanung nutzen, sodass nur an Stellen gefahren wird, an denen bereits Fahrzeuge beobachtet wurden.

Der Octomap-Server stellt über einen ROS-Publisher sogenannte *Costmaps* zur Verfügung, die direkt zur Pfadplanung genutzt werden können. Eine Costmap ist eine zweidimensionale, gerasterte Karte, die für jede Zelle Bewegungskosten enthält. Umso höher die Kosten, desto schlechter lässt sich hier fahren. Ein Hindernis schlägt sich in der Costmap mit unendlichen Kosten nieder und macht diesen Teil damit unpassierbar.

Die Verwendung der Bewegungsdaten der anderen Fahrzeuge macht die Erzeugung einer Costmap möglich, die dort wo andere Fahrzeuge schneller fahren können, geringere Kosten aufweist als an Stellen an denen sich andere Fahrzeuge nur sehr langsam bewegen. Dort wo keine anderen Fahrzeuge beobachtet werden, werden die Kosten auf unendlich gesetzt, da dort eventuell keine befahrbare Straße existiert (angenommen der Pfadplaner verfügt über keine Straßenkarten).

Damit lässt sich, kombiniert mit einer einfachen Kollisionserkennung, ein gutes Schwarmverhalten erzeugen. Da die Karten Richtungs- und Geschwindigkeitsinformationen enthalten, kann sich das Fahrzeug den anderen Fahrzeugen bezüglich Fahrweise und Geschwindigkeit anpassen und so im Schwarm „mitschwimmen“.

Da die existierende Implementierung der Costmap-Berechnung belegte Knoten in der Octomap als unpassierbar markierte, musste die Berechnung im Octomap-Server angepasst werden. Um die Costmaps zur Navigation auf Schwarmkarten zu nutzen, ist eine gegenteilige Markierung sinnvoll, da anzunehmen ist, dass die Stellen passierbar sind, an denen bereits andere Fahrzeuge unterwegs waren.

Deshalb arbeitet die verbesserte Implementierung umgekehrt: Es werden niedrige Kosten für Felder angesetzt, für die ein belegter Knoten in der Octomap existiert und unendliche Kosten für Felder ohne entsprechenden Knoten in der Octomap.

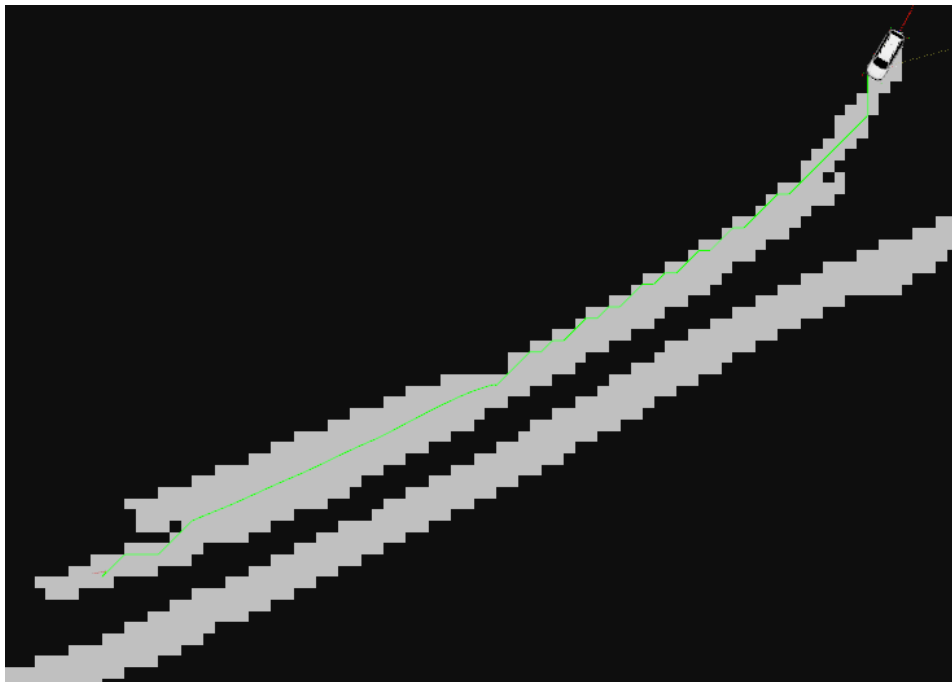


Abbildung 13: Ein geplanter Weg (grün) auf einer aus einer Schwarmkarte erstellten Costmap.

Abbildung 13 zeigt einen für Demonstrationszwecke auf einer Costmap einer Schwarmkarte geplanten Weg. Dafür wurde das „Move_Base“-Modul von ROS für die Pfadplanung verwendet und für die Costmap mit dem modifizierten „projected_map“-ROS-Publisher des Octomap-Servers verbunden.

4 Experimente und Auswertung

4.1 Experiment mit Echtweltdaten

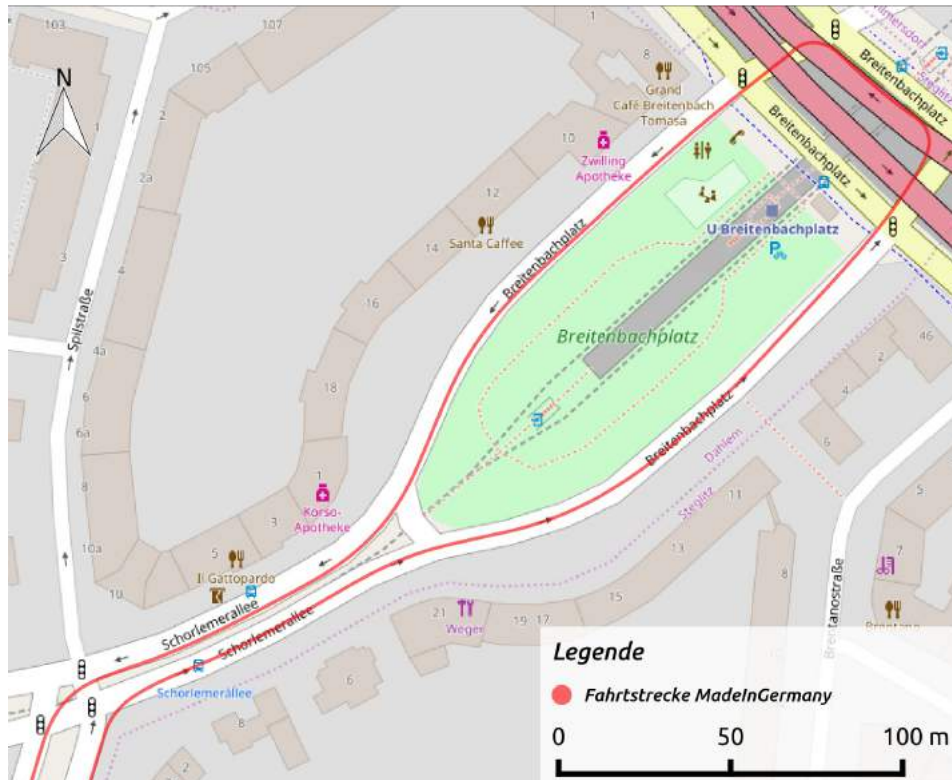


Abbildung 14: Fahrtstrecke des Messfahrzeugs. Straßenkarte: [29]

Für die Evaluation der schwarmbasierten Karten wurden reale Sensordaten einer wiederholten Fahrt des „MadeInGermany“ über die Englerallee und den Breitenbachplatz verwendet (siehe 14). Die Aufnahme der Sensordaten fand bei leichtem bis mäßigem Verkehrsaufkommen an einem Freitag um 13:46 Uhr statt. Dabei wurde die Strecke innerhalb von 16 Minuten insgesamt viermal zurückgelegt.

Die Auswertung der Daten konzentriert sich auf beide Richtungen des Bereichs von der Kreuzung Schorlemerallee-Englerallee, die sich in der süd-westlichen Ecke der Karte 14 befindet, über die Schorlemerallee bis zum Breitenbachplatz, der sich bis zur nord-östlichen Ecke der Karte 14 erstreckt. Es wird zunächst der gesamte Bereich als Einheit und anschließend die Kreuzung Schorlemerallee-Englerallee und der Breitenbachplatz einzeln analysiert.

Das Wendemanöver, das etwa einhundert Meter südlich der Kreuzung Schorlemerallee-Englerallee auf der Englerallee durchgeführt wurde, ist nicht auf der Karte 14 verzeichnet.

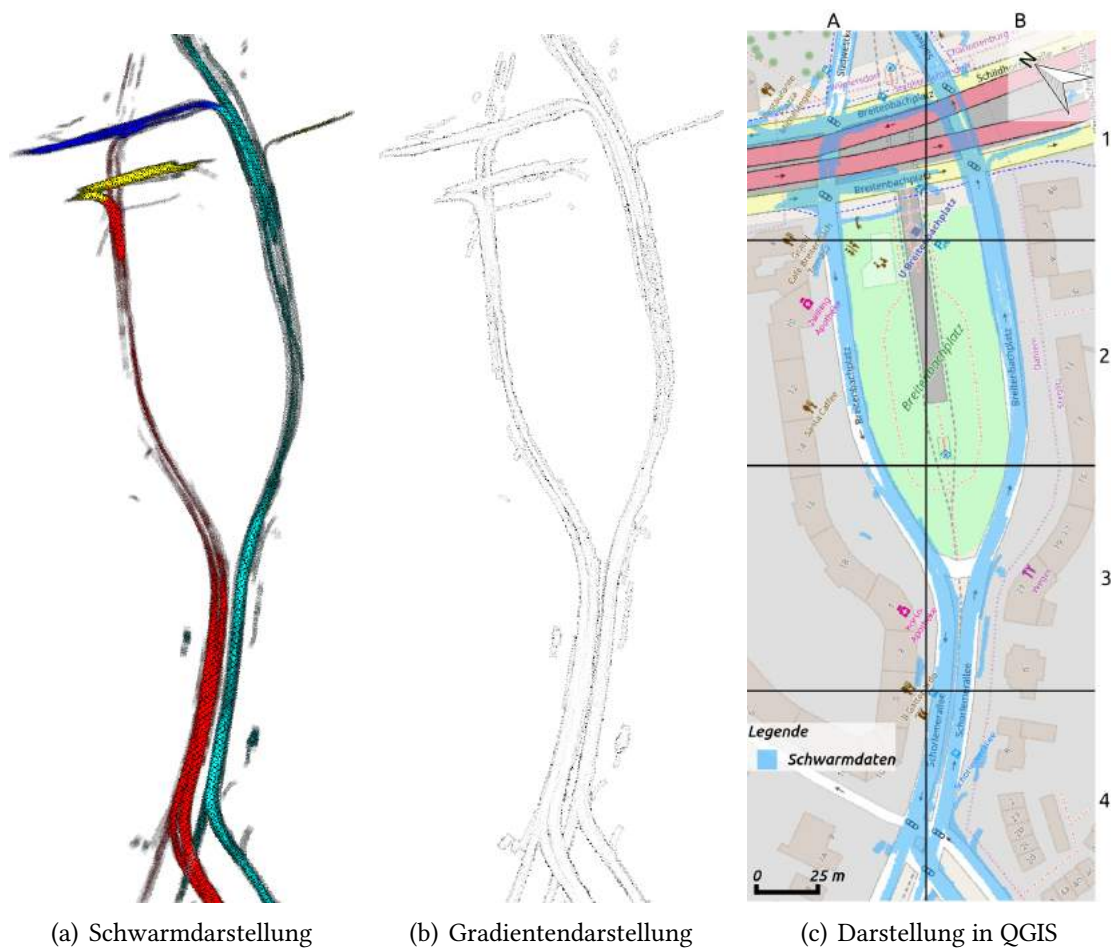


Abbildung 15: Verschiedene Darstellungen der Schwarmkarte Englerallee-Schorlemerallee-Breitenbachplatz in Berlin (0,5m Auflösung). Straßenkarte: [29]

Die Daten des Wendemanövers wurden allerdings auch nicht verwendet, da sie sich außerhalb des Ausschnitts 17 befanden.

4.1.1 Untersuchung der Schwarmkarte

Abbildung 15 zeigt die Schwarmkarte, die aus den Daten der Messfahrt (14) entstanden ist.

In den Feldern A4 und B4 befindet sich die Kreuzung Englerallee-Schorlemerallee, in den Feldern A4, B3 und B4 die Schorlemerallee, welche im Feld B3 in den Breitenbachplatz übergeht.

Folgende Beobachtungen lassen sich aufgrund der Kartendarstellung machen:

1. Während der gesamten Messfahrt ist kein Fahrzeug in die Spilstraße (Karte: A4) an der Kreuzung Englerallee-Schorlemerallee eingebogen, deshalb gibt es diese Straße in der Schwarmkarte nicht (vergleiche Straßenkarte 14 mit Schwarmkarte 15). Auch ist kein Fahrzeug von der Englerallee links in die Schorlemerallee abgebogen, deshalb findet sich keine Verbindung von der Englerallee in nördlicher Richtung zu der westwärtigen Schorlemerallee.
2. Der Breitenbachplatz in den Feldern B2 und B3 ist nur relativ schwach gezeichnet. Hier konnten nur wenige andere Fahrzeuge erkannt werden, weil an dieser Stelle die Fahrtgeschwindigkeit hoch ist und keine Ampel die Aufenthaltszeit des Messfahrzeugs erhöht hat.
3. Der Breitenbachplatz ist unterbrochen (siehe Abb. 15 Feld B1). Dies ist darauf zurückzuführen, dass das Messfahrzeug diesen Teil der Strecke nur gekreuzt hat. Die Ampelschaltung verhindert, dass das Messfahrzeug auf andere Fahrzeuge auf dieser Straße treffen kann.
4. Im Feld A1 gibt es eine prominente Fehlerkennung, die sich bis weit in den Breitenbachplatz hineinzieht. An dieser Stelle befindet sich ein Fußgängerweg, möglicherweise wurde fälschlicherweise ein Rollstuhl oder ein Kinderwagen als Kraftfahrzeug klassifiziert.
5. Die Felder A1 und A2 zeigen, dass die Nutzung einer Spur der zweispurigen Straße abrupt aufhört und erst später wieder aufgenommen wird. Dies liegt darin begründet, dass sich dort ein Parkplatz befindet. Ein quer zur Fahrbahn stehendes Fahrzeug, welches ein Parkmanöver vollzieht, ist in der Schwarmkarte erkennbar.
6. Die Gradientenkarte zeigt insbesondere im Bereich der Kreuzungen das Vorhandensein mehrerer Spuren an, aber an Stellen mit geringer Beobachtungsdichte neigt sie zu starkem Rauschen.

4.1.2 Vergleich der Umlaufanzahl

Durch das mehrfache Zurücklegen der Strecke lassen sich Karten mit einer unterschiedlich hohen Dichte an beobachteten Fahrzeugen erstellen, wie die Abbildung 16 zeigt. In diesem Abschnitt wird untersucht, ob Karten, die nach nur einmaligem oder zweimaligem Befahren der Messstrecke entstanden sind, genug Daten für die sinnvolle Verwendung enthalten.

Dafür werden die Abschnitte des Breitenbachplatzes und der Kreuzung Schorlemerallee-Englerallee (Bild 16) visuell analysiert und miteinander verglichen. Die Befahrungen der Kreuzung Schorlemerallee-Englerallee (Bild 16 e-g) erfolgten jeweils einmal von der Englerallee in die Schorlemerallee und umgekehrt.

Durch die größere Anzahl Beobachtungen steigt die Abdeckung der Karten an, aber bereits nach einer Umrundung wird eine nahezu geschlossene Karte gewonnen. Es ist deutlich zu erkennen, dass die Kreuzung Schorlemerallee-Englerallee bereits nach zweimaligem Befahren

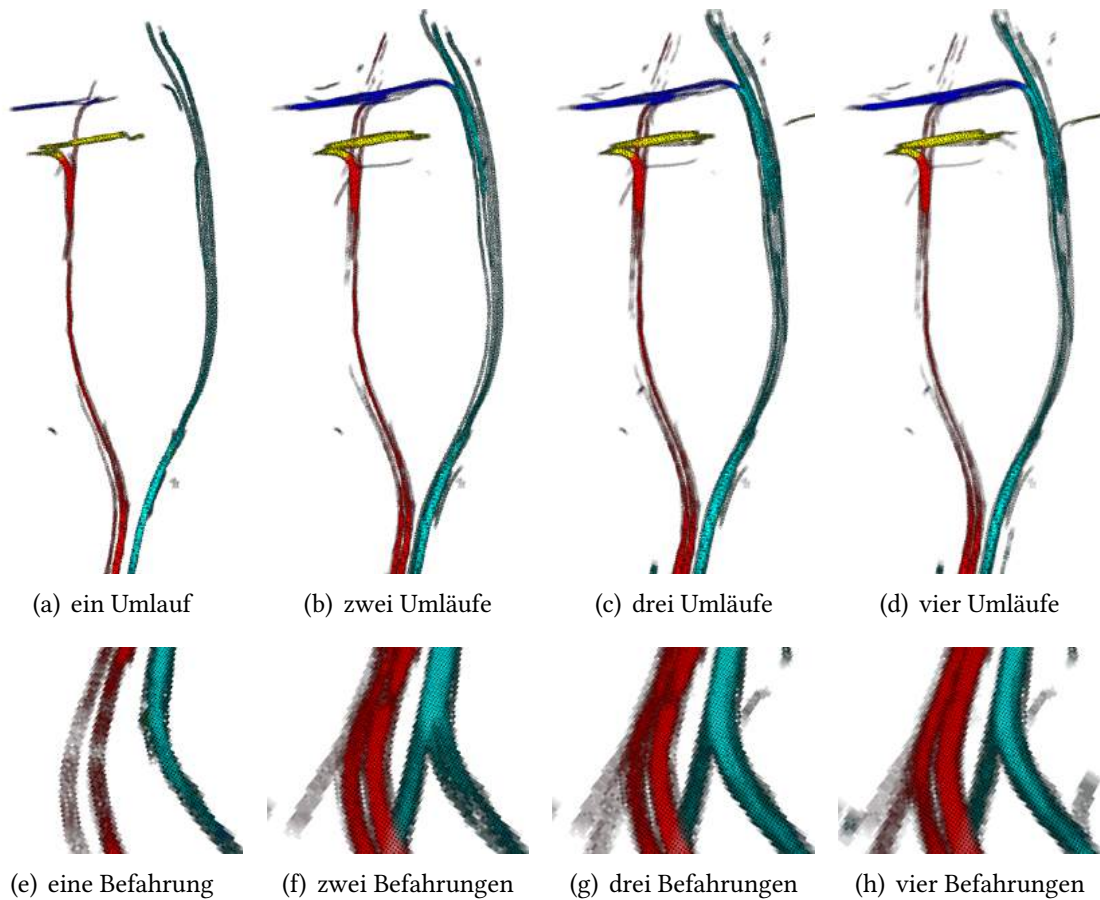


Abbildung 16: Schwarmkarten des Breitenbachplatzes und der Kreuzung Schorlemerallee-Englerallee in Berlin (Auflösung 0,5m)

von beiden Seiten eine qualitativ hochwertige Karte produziert, da durch die Wartezeiten an den Ampeln eine große Anzahl anderer Fahrzeuge erfasst werden können. Der Breitenbachplatz benötigt jedoch mehrere Durchfahrten, da die Geschwindigkeit hier deutlich höher ist und erheblich weniger andere Fahrzeuge beobachtet werden können.

Eine detaillierte Auswertung der Auswirkungen der Befahrungsanzahl wird im folgenden Abschnitt 4.2 durchgeführt.

4.2 Auswertung der Genauigkeit der Karten

Die erstellten Karten lassen sich nur dann zu Planungszwecken nutzen, wenn sie ein entsprechendes Maß von Genauigkeit haben.

Die Genauigkeit der Schwarmkarten wird in diesem Abschnitt mit einfachen Metriken analysiert. Die verwendeten Karten wurden allesamt aus den Daten, der im vorherigen Abschnitt 4.1



Abbildung 17: Referenzdaten (rot). Satellitenbild: ©2017, Google, DigitalGlobe, GeoBasis-DE/BKG, GeoContent

beschriebenen Messfahrt, erstellt. Abbildung 14 zeigt die Fahrtstrecke.

4.2.1 Methodik

Die Genauigkeitsanalyse wurde mit Hilfe von QGIS (siehe 3.1.7) durchgeführt. Dazu mussten die Schwarmkarten in QGIS importiert werden. Um dies zu erreichen, mussten die Daten zunächst in Octovis zu einer CSV-Datei exportiert werden. Anschließend wurde ein Python-Skript verwendet, das unter Verwendung der *Geographiclib* (siehe 3.1.7) die Koordinaten der Knoten von dem lokalen Kartenkoordinatensystem in das weltweite WGS84-Koordinatensystem überführt.

Um die WGS84-Koordinaten der einzelnen Knoten der Karte zu erhalten, wird der Abstand und der Winkel zu Referenz-Koordinaten im Koordinatensystem der Karte bestimmt, deren WGS84-Koordinaten bekannt sind.

Um die Referenz-Koordinaten zu ermitteln, lässt sich der hochpräzise GPS-Empfänger des „MadeInGermany“ verwenden. Dieser liefert dessen Position in WGS84-Koordinaten und da die Position des „MadeInGermany“ auch in Kartenkoordinaten bekannt ist, können diese

Tabelle 1: Genauigkeitsanalyse verschiedener Auflösungen der Schwarmkarte Englerallee-Schorlemerallee-Breitenbachplatz.

Auflösung	Positiv		Falsch positiv		Total
	Anzahl	%	Anzahl	%	Anzahl
1,0m	50972	95,2	2541	4,8	53513
0,5m	395267	94,0	25260	6,0	420527
0,33m	1096381	94,3	66533	5,7	1162914
0,25m	1901917	94,1	119985	5,9	2021902

als Referenz-Koordinaten genutzt werden, um die Beziehung von Kartenkoordinaten und WGS84-Koordinaten herzustellen.

Mit diesen drei Daten, der WGS84-Koordinaten von einem Vergleichspunkt in der Karte und Abstand und Richtung des anderen Punkts, dessen WGS84-Koordinaten bestimmt werden sollen, können die Koordinaten berechnet werden.

Durch diese Umrechnung von lokalen in globale Koordinaten können die Schwarmkarten in QGIS importiert und analysiert werden.

Für die Genauigkeitsanalyse wurde eine exakte Straßenkarte der Messstrecke benötigt. Diese wurde auf Basis von hochauflösenden Satellitenaufnahmen ermittelt und umfasst vollständig den befahrbaren Straßenbereich, aber ohne Fahrradwege und Parkflächen (siehe Abbildung 17).

4.2.2 Ergebnisse

Die Tabelle 1 vergleicht verschiedene Auflösungen miteinander. Hierfür wurde jede Beobachtung einzeln gezählt und als *positiv* gewertet, falls sie innerhalb oder auf dem Rand der Referenzdaten liegt, anderenfalls als *falsch positiv*. Die Spalte „Total“ listet die Gesamtanzahl an Beobachtung auf, diese steigt durch die Rasterung der Fahrzeuge mit zunehmender Auflösung an. Der Anteil an positiven Beobachtungen ist der Präzisionswert einer Schwarmkarte. Ein niedriger Präzisionswert deutet auf viele Fehlerkennungen hin, wodurch die Karte unbrauchbar wird.

Die größte Auflösung erreicht den besten Wert für die Präzision, da die Beobachtungen tendenziell kleiner gerastert wurden (es wird ab- statt aufgerundet). Die Präzisionsanalyse prüft nur, ob das Zentrum eines Knotens innerhalb der Grenzen der Straße liegt. Damit ist es möglich, dass ein als positiv gewerteter Knoten bei 1m Auflösung bis zu 70 cm über den Rand der Straße hinausragt, denn die halbe Diagonale einer solchen Knotenzelle beträgt $\frac{d}{2} = \frac{1m \cdot \sqrt{2}}{2} = 0,707m$.

Ebenso bemerkenswert ist der bessere Präzisionswert der Karte mit 33cm Auflösung. Dieser ist mit einer veränderten Rasterung der Fahrzeugbeobachtungen zu erklären:

Tabelle 2: Binäre Genauigkeitsanalysen verschiedener Auflösungen der Schwarmkarte Englerallee-Schorlemerallee-Breitenbachplatz.

Grenzwert	Precision %			Recall %			F1 %		
	1	2	4	1	2	4	1	2	4
Auflösung:									
1,0m	87,4	89,8	95,2	75,4	69,3	52,4	81,0	78,2	67,6
0,5m	84,4	86,3	89,6	81,8	79,6	70,5	83,1	82,8	78,9
0,33m	85,2	86,5	89,0	80,2	78,2	70,6	82,6	82,1	78,7
0,25m	84,4	85,8	88,4	81,5	78,7	70,3	82,9	82,1	78,3

Wie in Abschnitt 3.3 erläutert beträgt die Standardlänge für Beobachtungen 3 Meter. Dies gilt jedoch nicht für die Auflösung von 33cm, da die 3 Meter sich nicht ohne Rest durch 0,33m teilen lassen, sodass gerundet werden muss und die Standardlänge zu 2,97m wird.

Der Unterschied, den diese 3cm für die Präzision machen ist erstaunlich, jedoch heben die negativen Auswirkungen für den Recall die positiven für die Präzision mehr als auf (siehe Tabelle 2).

Die Präzisionswerte der 0,5m und 0,25m Auflösungen sind in Tabelle 1 nahezu identisch, dies trifft jedoch nicht auf die binäre Analyse zu.

Bei der Analyse in Tabelle 2, die die Werte für Precision, Recall und deren harmonisches Mittel $F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ zeigt, handelt es sich um mehrere binäre Analysen. Bei einer binären Analyse wird nur auf Knotenebene geprüft und nicht auf Beobachtungsebene, um damit die Vollständigkeit der Karte zu überprüfen.

Die Tabelle 2 vergleicht verschiedene Auflösungen mit verschiedenen binären Auswertungen miteinander. Die binären Auswertungen unterscheiden sich in dem gewählten Grenzwert von Beobachtungen ab welchem der Knoten als belegt gewertet wurde.

Der *Recall* wurde ermittelt, indem die Anzahl der auf der Straße liegenden Knoten berechnet wurde. Dies wurde erreicht, indem eine vollständige Octomap (d. h. alle Knoten auf der Straßenebene sind als belegt markiert) mit der gleichen Auflösung und dem gleichen Koordinatenursprung und Ausrichtung wie die zu prüfende Karte generiert wurde. Anschließend wurden alle nicht innerhalb der Referenz-Straßendaten liegenden Knoten gefiltert. Die Gesamtanzahl Knoten dieser gefilterten Referenzkarte entspricht einem Recall von 100%.

Der Recall zeigt die Abdeckung des Straßenbereichs durch die Schwarmkarte an. Ein zu niedriger Recall-Wert führt zu unterbrochenen Straßenzügen in der Schwarmkarte und macht diese damit unbrauchbar.

Die Tabelle 2 zeigt deutlich, dass die Präzision mit einem höheren binären Grenzwert zunimmt, da Randbeobachtungen gefiltert werden, aber der Recall gleichzeitig stark abnimmt.

Tabelle 3: Binäre Genauigkeitsanalysen der Schwarmkarten der Kreuzung Schorlemerallee-Englerallee und des Breitenbachplatzes.

Befahrung	Precision %		Recall %		F1 %	
	Br.-pl.	Kreuzung	Br.-pl.	Kreuzung	Br.-pl.	Kreuzung
1	91,2	64,8	46,4	16,0	61,5	25,6
2	88,6	83,8	70,6	51,9	78,6	64,1
3	89,2	83,9	78,7	77,7	83,6	80,0
4	88,6	83,9	80,2	78,2	84,1	81,0
5		81,5		78,6		80,0
6		81,1		84,2		82,6
7		80,2		84,6		82,3
8		78,8		84,6		81,6

In Bezug auf die Präzision der Auflösungen ist die gleiche Beobachtung zu machen wie in Tabelle 1, aber der Recall zeichnet ein anderes Bild: Die mit 0,5m aufgelöste Karte zeigt die besten Recall-Werte und auch der F1-Wert ist hier am höchsten. Insbesondere bei höheren Grenzwerten zeigt sich, dass die 0,5m-Auflösung die beste Wahl zu sein scheint, denn bei dieser Auflösung fallen noch genügend Beobachtungen in jeden Knoten, sodass der Recall bei höheren Grenzwerten geringer abnimmt als bei feineren Auflösungen. Die 1m-Auflösung hingegen hat einen sehr geringen Recall. Auch dies lässt sich mit der tendenziell zu kleinen Rasterung der Beobachtungen erklären.

Die besten Werte erreicht die Auflösung von 0,5m. Diese wird in der Tabelle 3 für die Kartenausschnitte der Kreuzung Schorlemerallee-Englerallee und des Breitenbachplatzes detailliert analysiert. Die Tabelle vergleicht dabei die Werte für Precision, Recall und F1 der verschiedenen Befahrungen. Dabei wird für jede Umfahrung des Breitenbachplatzes, die Kreuzung Schorlemerallee-Englerallee zweimal befahren: einmal von jeder Seite. Deshalb sind für den Breitenbachplatz nur bei den ersten vier Befahrungen Werte vorhanden. Die Ergebnisse in Tabelle 3 zeigen, dass die Präzision nach drei Befahrungen des Breitenbachplatzes und der Kreuzung Schorlemerallee-Englerallee ihren Höchstwert erreicht. Auch die F1-Summe scheint danach nicht mehr nennenswert zu steigen, was in der Erkenntnis resultiert, dass für binäre Karten eine dreifache Befahrung pro Richtung ausreicht. Der Recall steigt zwar noch etwas, jedoch steigt die Abdeckung der Straße nach drei Befahrungen hauptsächlich in Randbereichen, sodass dieser den Verlust der Präzision nicht aufwiegt.

Deshalb scheint die Verwendung eines Grenzwertes von mindestens zwei bis drei Beobachtungen für Karten, die durch viele Befahrungen entstanden sind, empfehlenswert.

Weiterhin bemerkenswert ist, dass die erste Befahrung der Kreuzung Schorlemerallee-Englerallee ein sehr schwaches Ergebnis mit einem geringen Präzisionswert liefert. Dies ist anscheinend auf eine temporär leicht verschobene Lokalisierung zurückzuführen (siehe Bild 16).

4.3 Evaluation weiterer Nutzungsszenarien

Dieser Abschnitt behandelt die Evaluation einiger Nutzungsszenarien der Schwarmkarten. Verwendet wurde zu diesem Zweck die Karte 15 mit 0,5m Auflösung, da diese die besten Ergebnisse in der Genauigkeitsanalyse erreichte.

Eine vollständige Implementierung und Evaluation dieser Szenarien würde den Rahmen dieser Masterarbeit sprengen, deshalb wird nur eine prototypische Implementierung vorgestellt und evaluiert.

Die Prototypen sollen die generelle Machbarkeit der Nutzung für die Szenarien demonstrieren und zu einer möglichen Erweiterung anregen.

4.3.1 Evaluation der Verwendung zur Abweichungserkennung

Die Schwarmkarten lassen sich auch dazu verwenden, um Abweichungen im Verkehrsfluss zu erkennen und somit früh Hindernisse auf der Straße und sogar Gefahrensituationen zu erkennen.

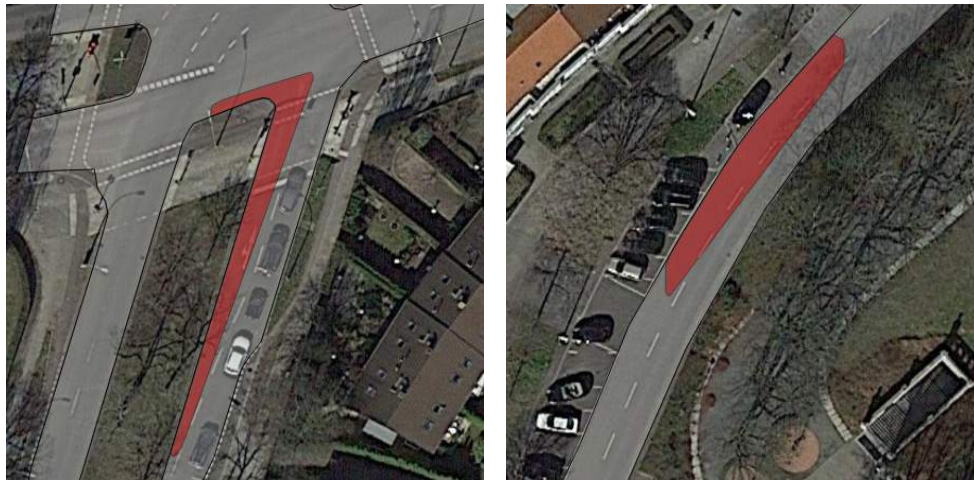
Um die Tauglichkeit der schwarmbasierten Karten für diese Aufgabe zu demonstrieren, wurde eine einfache Heuristik entwickelt und mit Hilfe von QGIS und Python umgesetzt. Die Implementierung setzt prinzipbedingt exakte Straßenkarten voraus.

Die Funktionsweise der Heuristik ist folgendermaßen:

1. Zunächst werden die Stellen der Straße isoliert, an denen keine Beobachtungen vorliegen. Dies kann erreicht werden, indem die symmetrische Differenz der zu überprüfenden mit einer vollständigen Karte (d. h. alle Knoten, die auf der Straße liegen, sind als belegt markiert) gebildet wird.
2. Die resultierende Karte zeigt an Stellen an denen Hindernisse zu erwarten sind, größere Cluster von belegten Knoten. Um diese zu isolieren, werden sie nach Größe gefiltert: Zu kleine (unter 2m Länge und Breite) deuten auf Rauschen hin und zu große (die sich auf die ganze Straßenbreite erstrecken) auf wenig befahrene Seitenstraßen.

Die Ergebnisse der Anwendung der Heuristik auf die Schwarmkarte (Abbildung 15) sind in der Darstellung 18 abgebildet. Es sind durch die Filterung (Schritt 2 der Heuristik) nur zwei Cluster übriggeblieben. Für diese Evaluation wurde die Filterung auf Basis des Flächeninhalts der Cluster durchgeführt: Es wurden Cluster mit einer Fläche, die geringer war als $60m^2$ oder größer als $150m^2$ gefiltert.

Der Cluster der Abbildung 18 Bild a) erstreckt sich über die gesamte Spur, die von der Englerallee geradeaus auf die Spilstraße oder links auf die Schorlemerallee führt. Wie bereits in Abschnitt 4.1.1, Beobachtung 1 festgestellt, ist kein Fahrzeug während der gesamten Messfahrt diesen Weg gefahren. Aufgrund dessen wurde die entsprechende Spur nicht genutzt.



(a) Unbenutzte Spur der Kreuzung Englerallee-Schorlemerallee

(b) Unbenutzte Spur vor Parkplätzen

Abbildung 18: Ergebnisse der Abweichungserkennung. Satellitenbilder: ©2017, Google, DigitalGlobe, GeoBasis-DE/BKG, GeoContent

Letztlich handelt es sich hierbei aber nicht um eine Abweichung im normalen Verkehrsfluss, deshalb ist dieser Cluster als falsch positiv zu erachten. Dies ist auf die Filterung zurückzuführen, die ausschließlich auf der Fläche der erkannten Bereiche basiert. Eine bessere Heuristik müsste spurgenaue filtern und Cluster-Kandidaten entfernen, die sich über alle Geradeaus- oder Abbiegespuren für eine bestimmte Straße erstrecken, da diese wahrscheinlich nur wegen der geringen Fahrzeugdichte der Zielstraße unbefahren sind und nicht wegen eines außergewöhnlichen Umstands.

Der Cluster b) derselben Abbildung erstreckt sich über eine der zwei Spuren des Breitenbachplatzes. Der Cluster korreliert mit Beobachtung 5 des Abschnitts 4.1.1 und zeigt tatsächlich eine Abweichung des Fahrverhaltens menschlicher Fahrer von dem, welches bei normaler Nutzung zu erwarten wäre. Während der vier Umrundungen der Messfahrt wurde nicht ein einziges Fahrzeug auf diesem Streckenabschnitt beobachtet, deshalb ist die Wahrscheinlichkeit, dass es sich um eine zufällige Abweichung handelt, sehr gering.

Es ist davon auszugehen, dass die anderen Fahrer auf die linke Spur ausweichen, um nicht von ein- oder ausparkenden Fahrzeugen ausgebremst zu werden. Zwar ist in der Schwarmkarte ein parkendes Fahrzeug erfasst, aber da auch während der anderen Umrundungen dieser Bereich von keinem Fahrzeug befahren wurde, liegt der Schluss nahe, dass die Fahrer diese Stelle präventiv meiden, um nicht von parkenden Fahrzeugen aufgehalten zu werden.



Abbildung 19: Ergebnisse der Intensionsvorhersage. Satellitenbild: ©2017, Google, Digital-Globe, GeoBasis-DE/BKG, GeoContent

4.3.2 Evaluation der Verwendung zur Intensionsvorhersage

Ein dritter, wichtiger Verwendungszweck der Schwarmkarten ist die Intensionsvorhersage. Kooperatives Verhalten setzt das Wissen der Intention der anderen Schwarmteilnehmer voraus, jedoch ist es besonders für Computer schwierig die Intention von menschlichen Fahrern jederzeit zu berechnen.

Hier könnten Schwarmkarten von großer Hilfe sein. Durch die aufgezeichneten Trajektorien anderer Fahrer, könnte die Intention für jedes Fahrzeug mit minimalem Rechenaufwand und vergleichsweise großer Zuverlässigkeit bestimmt werden.

Auch hierfür wurde für eine Machbarkeitsstudie eine naive Heuristik entwickelt, welche versucht aufgrund der Positionsdaten eines Fahrzeugs seine zukünftige Bewegung vorherzusagen, indem sie mögliche Trajektorien aus einer Schwarmkarte generiert. Die Generierung der Trajektorien erfolgt durch das Verfolgen der in den Knoten gespeicherten Richtungen der Karte.

Die folgende Aufzählung gibt die Funktionsweise der Heuristik kurz wieder:

1. Die Positionsdaten des Fahrzeugs, dessen Intention vorhergesagt werden soll, werden verwendet, um den entsprechenden Knoten in der Karte aufzulösen. Dieser Knoten ist der Startpunkt der vorhergesagten Trajektorie.
2. Es wird die Hauptbewegungsrichtung des ermittelten Knotens berechnet, d. h. die Richtungsklasse mit den meisten Beobachtungen.
3. Es wird der Nachbarknoten, der in der berechneten Richtung liegt ermittelt und der Trajektorie hinzugefügt. Falls die gewünschte Trajektorienlänge noch nicht erreicht ist, wird mit Punkt 2 fortgefahren.

Die Evaluierung wurde mit einer Schwarmkarte mit 0,5m Auflösung, die die Daten von zwei Umrundungen des Breitenbachplatzes enthielt (siehe Abb. 16 b) durchgeführt.

Die Heuristik wurde anschließend auf fünf zufällige Fahrzeugbeobachtungen aus den anderen zwei Umrundungen des Breitenbachplatzes angewendet, um die Trajektorien der Fahrzeuge für die nächsten 20 Knoten auf der Schwarmkarte vorherzusagen. Dies entspricht einer Distanz von mindestens 10 Metern bei einer Auflösung von 0,5m.

Die Ergebnisse der Heuristik sind in Abbildung 19 dargestellt. Die vorhergesagten Trajektorie sind durchaus akkurat, aber auf Kreuzungen stößt die naive Heuristik an ihre Grenzen, wie an den zwei nördlichsten Trajektorien zu erkennen ist.

Dies ist darin begründet, dass die Hauptbewegungsrichtung zwischen einzelnen Knoten stark variieren kann, da auf der Kreuzung viele Fahrzeugbeobachtungen in völlig verschiedenen Richtungen gemacht werden. Eine bessere Heuristik müsste hier mehrere mögliche Trajektorien generieren und mit Wahrscheinlichkeitswerten auf Basis der Kartendaten versehen.

Eine weitere Erkenntnis ist, dass die Trajektorien für Fahrzeuge, die nur ihre Geradeausfahrt fortsetzen, um etwa einen Meter von der beobachteten Trajektorie abweichen. Der Grund dafür ist wahrscheinlich die Diskretisierung der gespeicherten Winkel in die acht Richtungsklassen, welche für die Intentionsvorhersage zu grob zu sein scheint. Hier wäre es anzuraten die Karten für die Intentionsvorhersage mit einer höheren Winkelauflösung zu generieren, um genauere Vorhersagen zu erreichen.

Die von Norden aus dritte Trajektorie sagt den Abbiegevorgang des Fahrzeugs auf der Abbiegespur korrekt voraus, führt aber zur mittleren anstatt zur linken Spur. Die für die Intentionsvorhersage verwendete Karte enthält keine Daten eines Fahrzeuges, welches an dieser Stelle auf die linke Spur abbog, deshalb ist die vorhergesagte Trajektorie an dieser Stelle ungenau. Das erste Fahrzeug mit dieser Trajektorie wird erst bei der dritten Umrundung des Breitenbachplatzes beobachtet (vergleiche Abb. 16 b und c).

5 Zusammenfassung und Ausblick

5.1 Zusammenfassung

Im Rahmen dieser Masterarbeit wurde die Motivation von schwarmbasierten Karten dargestellt. Es wurde das AutoNOMOS Labs-Projekt und das SWARMS-Projekt in Kapitel 1 vorgestellt.

In dem folgenden Kapitel 2 wurde eine Einführung in die Schwarmintelligenz gegeben und der Forschungsstand in Bezug auf Schwarmverhalten in der Robotik, im Straßenverkehr und autonomer Straßenfahrzeuge dargestellt.

Das Kapitel 3 stellte Software und Frameworks vor, die in der anschließend beschriebenen Implementierung genutzt wurden, um ein System zur Kartographierung von Kreuzungen und Straßenzügen auf Basis von Beobachtungen anderer Fahrzeuge zu erstellen.

Es wurde ebenso auf die Erweiterung des Programms Octovis eingegangen, die nötig waren, um informative Visualisierungen der Schwarmkarten zu erhalten.

Außerdem wurde demonstriert, dass die Schwarmkarten zur Wegplanung in ROS geeignet sind.

In Kapitel 4 wurde gezeigt, dass sich allein durch die Beobachtung anderer Verkehrsteilnehmer präzise Karten erstellen lassen. Dies wurde durch die Auswertung von Schwarmkarten, die von realen Sensordaten des „MadeInGermany“-Forschungsfahrzeugs gewonnen wurden, aufgezeigt. Die erstellten Karten wurden einer intensiven Genauigkeitsanalyse unter Zuhilfenahme eines Geoinformationssystems unterzogen. Dabei wurden verschiedene Auflösungen, Grenzwerte für binäre Karten und die Entwicklung der Karten über mehrere Befahrungen einzelner Straßenabschnitte hinweg, miteinander verglichen und analysiert.

Weiterhin wurden mehrere Nutzungsszenarien für diese Karten vorgestellt, prototypenhaft implementiert und die Ergebnisse ausgewertet.

Durch die Entwicklung des Schwarmkartierungssystems wurde ein Grundstein für die Verwertung der durch die Beobachtung anderer Fahrzeuge anfallenden Daten gelegt. Der Einsatz der generierten Karten, um Abweichungen im Verkehrsfluss oder die Intention anderer Fahrer zu erkennen, würde das autonome Fahren voranbringen und die Implementierung von kooperativen Fahrverhalten vereinfachen.

5.2 Ausblick

Wie im vorhergehenden Abschnitt erwähnt, konnte im Rahmen dieser Arbeit nur ein Grundstein eines adaptiven Schwarmkartographierungssystems gelegt werden, das die denkbaren Möglichkeiten kaum ausnutzt.

In diesem Abschnitt werden einige naheliegende Erweiterungen erläutert, die die Schwarmkarten zu einem mächtigen Werkzeug der autonomen Navigation machen könnten.

5.2.1 Erweiterung der Karten mit Fahrradfahrern

In vielen Innenstädten ist das Fahrrad ein beliebtes Verkehrsmittel, jedoch kommt es leider immer wieder zu teils schweren Unfällen zwischen PKWs bzw. LKWs und Radfahrern. Insbesondere Kreuzungen und Einmündungen, bei denen Fahrradfahrer die Fahrbahn der Autos kreuzen müssen, stellen einen Gefahrenschwerpunkt dar und führen oft zu Unfällen mit abbiegenden Autos [37]. Deshalb könnte eine Erweiterung der Karten mit Fahrradfahrern autonomen Fahrzeugen helfen Situationen mit Fahrradfahrern in ihrer Fahrtplanung zu berücksichtigen.

Die IBEO-Laserscanner sind bereits in der Lage Zweiräder zu erfassen und zu klassifizieren, sodass nur die Integration in die Schwarmkarte geleistet werden müsste. Dadurch könnten Schwarmkarten bürgersteigführende und straßenbasierte Fahrradwege ausweisen und sogar eine Intentionsvorhersage für Fahrradfahrer anbieten.

5.2.2 Erweiterung der Karten mit zusätzlichen Merkmalen

Die Karten speichern derzeit nur Bewegungsinformationen anderer Fahrzeuge, jedoch sind sie prinzipiell geeignet Informationen jeglicher Art zu speichern und zu verwalten.

Eine Erweiterung der Karten mit zusätzlichen Informationen, wie Gefahrenstellen, Baustellen, Staus oder freien Parkplätze wäre möglich und sinnvoll.

Durch die Abweichungsanalyse lassen sich bereits jetzt problematische Stellen im Straßenverkehr erkennen und umplanen. Zusätzliche Informationen, die über weitere Sensorik oder über andere Informationskanäle, wie z. B. Verkehrsfunk, erfasst werden, könnten in die Karte integriert werden. Dies wäre insbesondere dann sinnvoll, wenn die Karte mit anderen Fahrzeugen geteilt wird (siehe nächsten Abschnitt 5.2.3).

5.2.3 Car2Car- und Car2I-Kommunikation

Durch die immer stärkere Computerisierung und Vernetzung von Fahrzeugen wären die technischen Möglichkeiten zum Austausch von Schwarmkarten zwischen mehreren Fahrzeugen gegeben. In diesem Zusammenhang bezeichnet die *Car2Car*- oder *Vehicle2Vehicle*-Kommunikation (kurz *V2V*) den Datenaustausch zwischen Fahrzeugen und die *Car2Infrastructure*- oder *Vehicle2Infrastructure*-Kommunikation (kurz *Car2I* und *V2I*) den Austausch von Daten zwischen Fahrzeugen und anderen Objekten der Verkehrsinfrastruktur, wie z. B. Ampeln, Bahnübergängen, oder auch Servern, die über das Internet erreichbar sind.

Folglich könnten durch *Car2Car*- und *Car2I*-Kommunikation die Karten zwischen Fahrzeugen und auch anderen Objekten, wie LTE-Basisstationen und zentralen Servern im Internet ausgetauscht und aktuell gehalten werden.

Damit könnten sich Autos innerhalb von Bruchteilen einer Sekunde gegenseitig vor Gefahren, wie nicht abgesicherten Unfallstellen, Falschfahrern oder Tieren an oder auf der Fahrbahn warnen. Einsatzfahrzeuge von Polizei und Feuerwehr könnten ihr Kommen, lange bevor die Sirenen hörbar sind, ankündigen; der Winterdienst könnte bereits geräumte Straßenzüge markieren; die Müllabfuhr, Kräne und Schwertransporte könnten andere Fahrzeuge über die durch sie blockierte Straße warnen, bevor diese in die Straße eingebogen sind und auch freie Parkplätze könnten in die Karte aufgenommen werden, sodass sich die Parkplatzsuche auf ein Minimum verkürzt.

Diese Aufzählung umfasst nur einige der vielen weiteren Möglichkeiten zur Nutzung dieses dynamischen, aktualisierbaren Kartenformats.

Derzeit wird die Entwicklung von Kartenformaten für autonome Autos mit ähnlichem Funktionsumfang auch von einigen privaten Firmen verfolgt; unter anderem HERE [26] und TomTom [42]. Hierbei könnte eine Standardisierung eines Austauschprotokolls helfen, um die Interoperabilität der verschiedenen Kartensysteme zu gewährleisten.

Es existieren bereits einige Standards bezüglich der Car2Car- und Car2I-Kommunikation, wie der Standard *IEEE 802.11p*, welcher auch als *Wireless Access for Vehicular Environments (WAVE)* bezeichnet wird und *SAE J2735*, der hauptsächlich für die Car2Car-Kommunikation verwendet wird [9, S. 14-19]. Weitere Standards, die auf dem Mobilfunknetz basieren und keine zusätzliche Infrastruktur benötigen sollen, befinden sich in der aktiven Entwicklung [9, S. 23-33].

In Bezug auf diese Arbeit ist besonders der EU-Standard *EN 302 637-2* hervorzuheben. Dieser Standard spezifiziert den *Cooperative Awareness Basic Service*, der das Nachrichtenformat *Cooperative Awareness Message (CAM)* definiert. Diese CAMs sollen per Car2Car- und Car2I-Kommunikation ausgetauscht werden und die jeweils anderen Kommunikationspartner über eigene Position, Geschwindigkeit und weitere Fahrdynamiken informieren [9, S. 32]. Die über das CAM-Protokoll empfangenen Informationen ließen sich direkt in die Schwarmkarten einlesen.

Der EU-Standard *EN 302 637-3* definiert den *Decentralized Environmental Notification (DEN) Basic Service*, welcher die *Decentralized Environmental Notification Message (DENM)* beinhaltet, die über Gefahrensituationen und ungewöhnlichen Verkehrssituationen warnen soll [9, S. 32]. Auch die, durch diese Nachrichten verbreiteten, Informationen würden sich sinnvoll in die Schwarmkarten einfügen lassen.

Da diese ohnehin das Potential haben abnormales Verkehrsverhalten zu erkennen (siehe Abschnitt 4.3.1), könnten die Daten der DENMs mit den Ergebnissen der eigenen Heuristik korreliert werden.

Mit einer zunehmenden Anzahl teil- und vollautonomer Fahrzeuge auf im Straßenverkehr gewinnt die sichere, kooperative Navigation dieser, im Verbund mit manuell gefahrenen Fahrzeugen, immer größere Bedeutung. Dynamische Karten, wie die in dieser Arbeit vorgestellten Schwarmkarten, werden unabdingbar, um den entstehenden Anforderungen gerecht zu werden.

Car2Car- und Car2I-Kommunikation könnten die Schwarmkarten mit zusätzlichen Informationen anreichern und dadurch zu einer unverzichtbaren Datengrundlage für die effiziente Pfadplanung autonomer Autos machen.

Die hier vorgestellten Standards für die Fahrzeugkommunikation können ohne großen Aufwand in die Schwarmkarten integriert werden, um dieses Ziel zu erreichen.

6 Anhang

6.1 Schwarmkarten aus Taxi-Trajektorien



Abbildung 20: Die Porto Schwarmkarte in Octovis.

Zur Evaluation des Octomap-Schwarmkartenformats war es nötig die Skalierbarkeit mit einer großen Menge Beobachtungsdaten zu untersuchen. Zu diesem Zweck wurde ein Trajektoriendatensatz aller Taxifahrten der 442 Taxis in der portugiesischen Stadt Porto zwischen dem 1. Juli 2013 und dem 30. Juni 2014 verwendet, der im Rahmen des *ECML/PKDD 15: Taxi Trajectory Prediction (I)*-Wettbewerbs auf der Website *kaggle.com* veröffentlicht wurde. Dieser Wettbewerb wurde im Rahmen der *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2015* organisiert und am 1. Juli 2015 beendet [39].

Die Trajektorien in dem Datensatz enthalten in regelmäßigen Abständen die Positionsdaten der Taxis im WGS84-Format, sowie die Start- und Zielposition. Allerdings sind nicht alle Trajektorien vollständig. Die Erfassung der Koordinaten wurde durch die in den Taxis installierten mobilen Datenterminals erreicht, die über einen GPS-Empfänger verfügen [39].

Diese Empfänger erreichen nicht die hohe Genauigkeit, wie der des „MadeInGermany“, aber für eine straßengenaue Lokalisierung sind sie ausreichend.



Abbildung 21: Die Porto-Schwarmkarte in Octovis in der Gradientendarstellung.

Die Daten wurden so gefiltert, dass nur Punkte innerhalb eines 10km Radius um den Stadtkern akzeptiert wurden, um eine zu große Ausdehnung der Karte zu verhindern. Für die Auflösung der Karten wurde 3m gewählt, um der Ungenauigkeit der GPS-Empfänger gerecht zu werden.

Den Trajektorien fehlen Richtungs- und Geschwindigkeitsinformationen. Deshalb mussten die Richtungsinformationen aus jeweils zwei aufeinanderfolgenden Punkten berechnet werden. Da die Punkte der Trajektorien teilweise relativ weit auseinanderliegen, ist die dadurch gewonnene Richtungsinformation recht ungenau, jedoch lassen sich zumindest einige Einbahnstraßen in der erstellten Karte erkennen (siehe Karte 23). Der jeweils letzte Punkt einer Trajektorie wird verworfen, da auf eine Extrapolation der Richtungsinformation für den letzten Punkt verzichtet wurde.

Wegen der fehlenden Geschwindigkeitsinformationen wurde die Geschwindigkeit in der Schwarmkarte auf eine Konstante gesetzt.

Abbildung 20 zeigt einen Ausschnitt aus der erstellten Schwarmkarte, welche über mehr als vier Millionen Knoten verfügt, aber weniger als 300 MiB Arbeitsspeicher benötigt.

Die Darstellung 21 zeigt die Porto-Schwarmkarte im Gradientenmodus. In diesem Modus sind die einzelnen Straßen besonders gut zu erkennen, wie beispielsweise die Hauptstraße

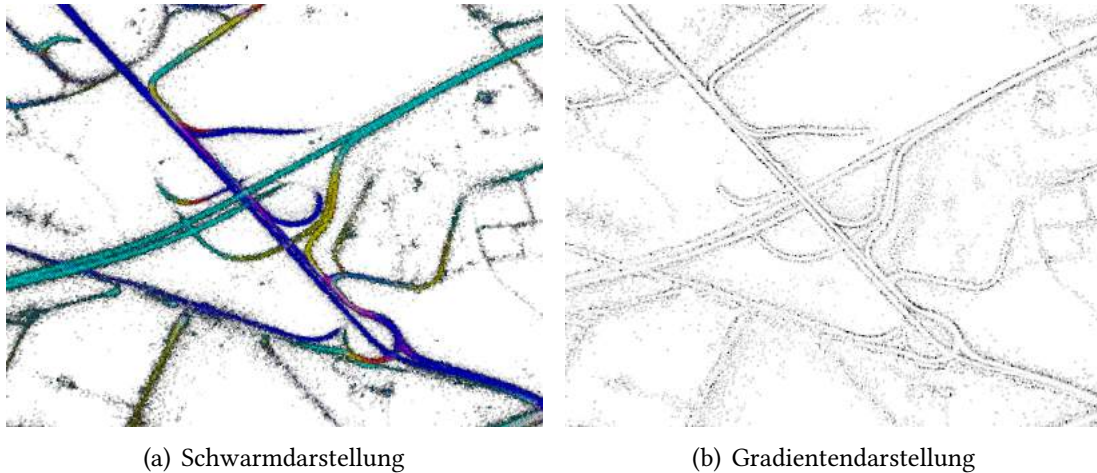


Abbildung 22: Autobahnkreuz der Porto-Schwarmkarte

„Av. da Boavista“ und der zentrale Kreisverkehr „Rotunda da Boavista“. Auch die Küste und der Fluss „Douro“ zeichnet sich deutlich in der Abbildung ab.

Die Gegenüberstellung 22 zeigt ein Autobahnkreuz der Porto-Schwarmkarte, dass durch die verschiedenen Darstellungsmöglichkeiten von Octovis dargestellt wurde. Teilweise lassen sich sogar die einzelnen Richtungsspuren erkennen, aber es gibt auch immer wieder Unterbrechungen der Straße, möglicherweise wegen Unterführungen, die den GPS-Empfang verhindern.

Die Karte 23 zeigt die Daten der Schwarmkarte über der Straßenkarte von OpenStreetMap im Vergleich mit der Schwarmkartendarstellung in Octovis. Die Darstellung von Octovis ist durch die dreidimensionale Darstellungsweise leicht verzerrt, dennoch lassen sich die einzelnen Straßen zuordnen.

Die Übersichtskarte vermittelt die Ausmaße der Schwarmkarte, die fast ganz Porto abdeckt. Zur besseren Orientierung sind an den Rändern die WGS84-Koordinaten eingeblendet.

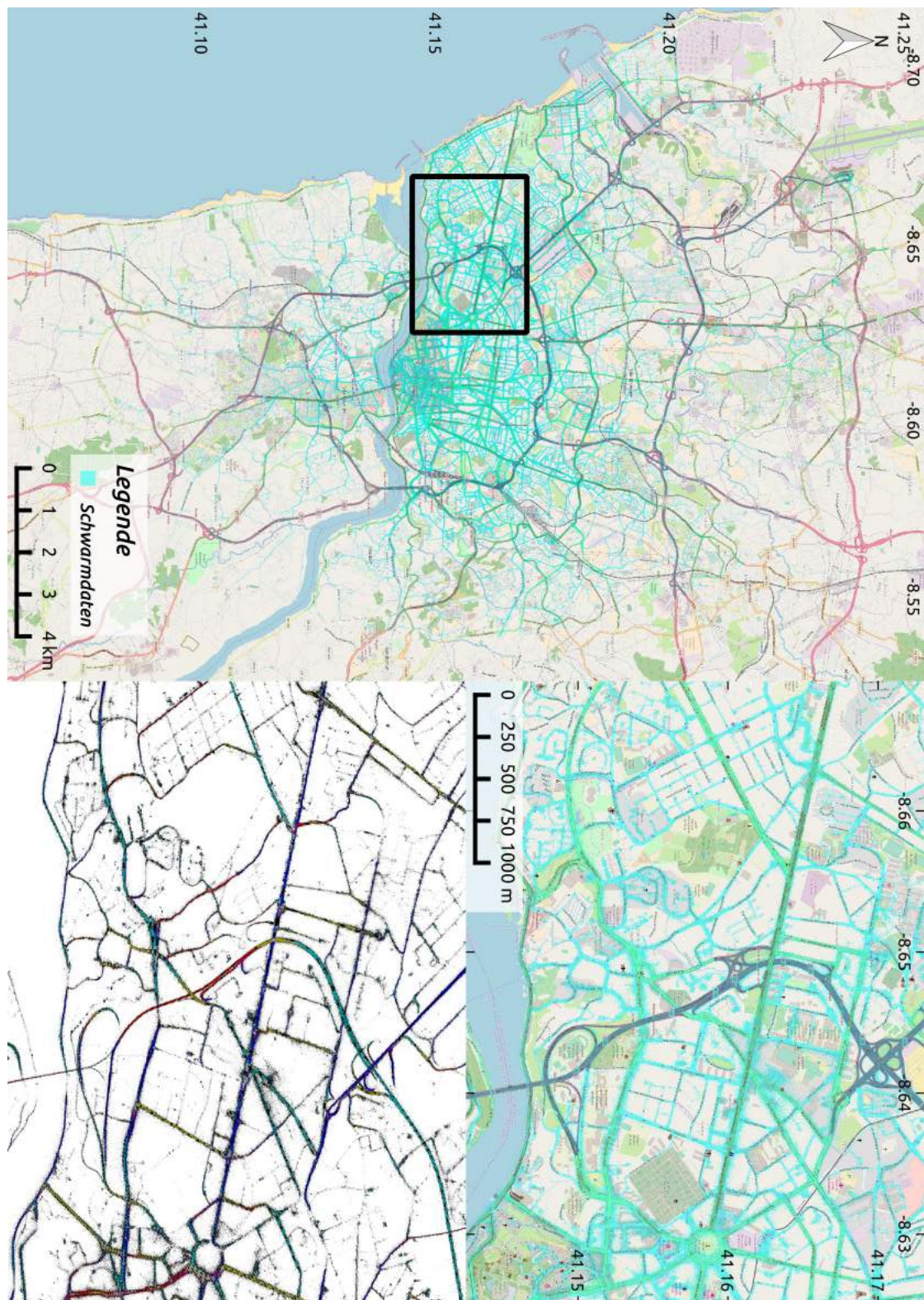


Abbildung 23: Die Porto-Schwarmkarte (türkis) über der Straßenkarte von OpenStreetMap im Vergleich mit der Schwarmkarte in Octovis. Karte: [29]

Literatur

- [1] AutoNOMOS Labs. e-instein | autonomos labs. <http://autonomos-labs.com/vehicles/e-instein/>, 2015. [Online; abgerufen 23. August 2017].
- [2] AutoNOMOS Labs. Madeingermany | autonomos labs. <http://autonomos-labs.com/vehicles/made-in-germany/>, 2015. [Online; abgerufen 7. August 2017].
- [3] N. Bellomo and F. Brezzi. Traffic, crowds, and swarms. *Mathematical Models and Methods in Applied Sciences*, 18(supp01):1145–1148, 2008.
- [4] R. Bishop, D. Bevy, J. Switkes, and L. Park. Results of initial test and evaluation of a driver-assistive truck platooning prototype. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 208–213. IEEE, 2014.
- [5] Bundesanstalt für Straßenwesen. Rechtsfolgen zunehmender fahrzeugautomatisierung. http://www.bast.de/DE/Publikationen/Foko/Downloads/2012-11.pdf?__blob=publicationFile, November 2012. [Online; abgerufen 31. Juli 2017].
- [6] P. Czerwionka. A three dimensional map format for autonomous vehicles. Master’s thesis, Freie Universität Berlin, 2014.
- [7] Die Deutsche Forschungsgemeinschaft. Dfg - gepris - spp 1835: Kooperativ interagierende automobile. <http://gepris.dfg.de/gepris/projekt/255645177>, 2015. [Online; abgerufen 23. August 2017].
- [8] Die Deutsche Forschungsgemeinschaft. Dfg - gepris - swarms: Schwarmintelligenz für autonome fahrzeuge. <http://gepris.dfg.de/gepris/projekt/273356841>, 2015. [Online; abgerufen 23. August 2017].
- [9] G. Dimitrakopoulos and G. Bravos. *Current Technologies in Vehicular Communication*. Springer, 2017.
- [10] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.
- [11] Ford Motor Company. Autonomous 2021. <https://corporate.ford.com/innovation/autonomous-2021.html>, August 2016. [Online; abgerufen 31. Juli 2017].
- [12] Freie Universität Berlin. Mission brandenburger tor erfüllt. http://www.fu-berlin.de/presse/informationen/fup/2011/fup_11_291/index.html, September 2011. [Online; abgerufen 1. August 2017].
- [13] Freie Universität Berlin, Institut für Informatik. Häufig gestellte fragen zum autonomos-projekt. http://autonomos.inf.fu-berlin.de/sites/default/files/AutoNOMOS_FAQs.pdf, 2011. [Online; abgerufen 1. August 2017].
- [14] Freie Universität Berlin, Institut für Informatik. Darpa urban challenge | autonomos labs. <http://autonomos-labs.com/projects/past-projects/darpa-urban-challenge/>, 2015. [Online; abgerufen 2. August 2017].

- [15] Freie Universität Berlin, Institut für Informatik. History | autonomos labs. <http://autonomos-labs.com/history/>, 2015. [Online; abgerufen 2. August 2017].
- [16] General Motors Co. Gm produces first round of self-driving chevrolet bolt ev test vehicles. <http://media.gm.com/media/us/en/gm/home.detail.html/content/Pages/news/us/en/2017/jun/0613-boltev.html>, June 2017. [Online; abgerufen 31. Juli 2017].
- [17] D. Göhring. Controller Architecture for the Autonomous Cars: MadeInGermany and e-Instein. Technical report, Freie Universität Berlin, Fachbereich Informatik, November 2012.
- [18] A. Hornung. octomap - ros wiki. <http://wiki.ros.org/octomap>, 2014. [Online; abgerufen 4. September 2017].
- [19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at <http://octomap.github.com>.
- [20] Z.-Y. Huang and G. E. Robinson. honey bee colonies. *Information processing in social insects*, page 165, 1999.
- [21] HYUNDAI MOTOR AMERICA. Hyundai motor company introduces new autonomous ioniq concept at automobility los angeles. <http://www.hyundainews.com/us/en/media/pressreleases/46598/hyundai-motor-company-introduces-new-autonomous-ioniq-concept-at-automobility-los-angeles4>, November 2016. [Online; abgerufen 31. Juli 2017].
- [22] L Keller. Queen lifespan and colony characteristics in ants and termites. *Insectes Sociaux*, 45(3):235–246, 1998.
- [23] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.
- [24] J. Krause, G. D. Ruxton, and S. Krause. Swarm intelligence in animals and humans. *Trends in Ecology & Evolution*, 25(1):28 – 34, 2010.
- [25] T. McReynolds and D. Blythe. *Advanced graphics programming using OpenGL*. Elsevier, 2005.
- [26] G. Miller. Autonomous cars will require a totally new kind of map. <https://www.wired.com/2014/12/nokia-here-autonomous-car-maps/>, December 2014. [Online; abgerufen 4. September 2017].
- [27] Open Source Robotics Foundation. About ros. <http://www.ros.org/about-ros/>. [Online; abgerufen 7. August 2017].
- [28] Open Source Robotics Foundation. History. <http://www.ros.org/history/>. [Online; abgerufen 7. August 2017].
- [29] OpenStreetMap contributors. Map data retrieved from <https://www.openstreetmap.org>. <https://www.openstreetmap.org>, 2017.

- [30] J. Penner, R. Hoar, and C. Jacob. Swarm-based traffic simulation with evolutionary traffic light adaption. In *Proceedings of the IASTED International Conference on Applied Simulation and Modelling (ASM)*, Crete, Greece, 2002. ACTA Press.
- [31] Python Software Foundation. geographiclib 1.45 : Python package index. <https://pypi.python.org/pypi/geographiclib/1.45>, 2017. [Online; abgerufen 4. September 2017].
- [32] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, 21(4):25–34, 1987.
- [33] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [34] SAE International. (r) taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. http://standards.sae.org/j3016_201609/, September 2016. [Online; abgerufen 31. Juli 2017].
- [35] E. Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pages 10–20. Springer, 2004.
- [36] S. Senge. *Ein Bienen-inspiriertes Schwarmintelligenz-Verfahren zum Routing im Straßenverkehr*. PhD thesis, Universität Oldenburg, 2014.
- [37] Stab des Polizeipräsidenten Berlin, PPr St II 4. Verkehrssicherheitslage 2016. https://www.berlin.de/polizei/_assets/aufgaben/anlagen-verkehrssicherheit/verkehrssicherheitslage2016.pdf, 2016. [Online; abgerufen 30. August 2017].
- [38] D. V. A. H. G. Swaroop. String stability of interconnected systems: An application to platooning in automated highway systems. *California Partners for Advanced Transit and Highways (PATH)*, 1997.
- [39] The Organization of ECML/PKDD 2015. Ecml/pkdd 15: Taxi trajectory prediction (i). <https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i>, 2015. [Online; abgerufen 31. August 2017].
- [40] The QGIS Projekt. Discover qgis. <http://www.qgis.org/en/site/about/index.html>, 2017. [Online; abgerufen 26. August 2017].
- [41] The ROS Team. Ros kinetic kame released. <http://www.ros.org/news/2016/05/ros-kinetic-kame-released.html>, September 2016. [Online; abgerufen 2. August 2017].
- [42] TomTom N.V. Tomtom hd map. <https://automotive.tomtom.com/wordpress/wp-content/uploads/2017/01/HD-Map-Product-Info-Sheet-improved-1.pdf>, 2017. [Online; abgerufen 28. Juli 2017].
- [43] Toyota Motor Corporation. Highway teammate. http://www.toyota-global.com/pages/contents/innovation/intelligent_transport_systems/world_congress/2015bordeaux/pdf/Highway_Teammate_A0.pdf, October 2015. [Online; abgerufen 31. Juli 2017].

- [44] F. Ulbrich, S. Rotter, D. Goehring, and R. Rojas. Extracting path graphs from vehicle trajectories. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 1260–1264. IEEE, 2016.
- [45] Volkswagen AG. Individual mobility redefined: Autonomous driving at the touch of a button. https://www.volkswagenag.com/en/news/2017/03/Autonomous_driving.html, March 2017. [Online; abgerufen 31. Juli 2017].
- [46] Willow Garage. Open source robotics foundation. <http://www.willowgarage.com/blog/2012/04/16/open-source-robotics-foundation>, April 2012. [Online; abgerufen 7. August 2017].
- [47] K. M. Wurm and A. Hornung. Octomap - 3d occupancy mapping. <http://octomap.github.io/>. [Online; abgerufen 4. September 2017].